

Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Сибирский государственный университет геосистем и технологий»
(СГУГиТ)

ИНТЕРЭКСПО ГЕО-СИБИРЬ

XXII Международный научный конгресс

Сборник материалов в 9 т.

Т. 9

Международная научная конференция

**«ПЕРСПЕКТИВНЫЕ РЕШЕНИЯ В СФЕРЕ
ИНФОРМАЦИОННО-КОММУНИКАЦИОННЫХ
ТЕХНОЛОГИЙ И КИБЕРБЕЗОПАСНОСТИ»**

№ 2

Новосибирск
СГУГиТ
2026

Ответственные за выпуск:

Кандидат технических наук, директор Научно-исследовательского института
информационно-коммуникационных технологий, ведущий научный сотрудник
Национального исследовательского ядерного университета «МИФИ», г. Москва
Е. А. Басыня

Кандидат технических наук, доцент, заведующая кафедрой прикладной информатики
и информационных систем, СГУГиТ, г. Новосибирск
Т. Ю. Бугакова

Кандидат технических наук, доцент кафедры прикладной информатики и информационных
систем, СГУГиТ, г. Новосибирск
С. Ю. Кацко

С26 Интерэкспо ГЕО-Сибирь. XXII Международный научный конгресс,
18–22 мая 2026 г., Новосибирск : сборник материалов в 9 т. Т. 9: Междуна-
родная научная конференция «Перспективные решения в сфере информаци-
онно-коммуникационных технологий и кибербезопасности». – Новоси-
бирск : СГУГиТ, 2026. № 2. – 251 с. – ISSN 2618-981X. – Текст : непосред-
ственный.

DOI 10.33764/2618-981X-2026-9-2

В сборнике опубликованы материалы XXII Международного научного конгресса
«Интерэкспо ГЕО-Сибирь», представленные на Международной научной конференции
«Перспективные решения в сфере информационно-коммуникационных технологий и ки-
бербезопасности».

Печатается по решению редакционно-издательского совета СГУГиТ

Материалы публикуются в авторской редакции

Издание может содержать сведения об иностранных агентах

Л. Э. Мамонова^{1✉}, А. А. Шарпов¹

Проектирование и разработка пользовательского интерфейса веб-ресурса управления научными конференциями с применением фреймворка Vue.js и принципов UX/UI

¹Сибирский государственный университет геосистем и технологий,
г. Новосибирск, Российская Федерация
e-mail: mamonova_lyudmila@list.ru

Аннотация. В статье представлены основные результаты проектирования и разработки пользовательского интерфейса веб-ресурса для управления научными конференциями. Выполнен анализ предметной области, в котором рассмотрены основные процессы организации конференций, выделены роли пользователей и выявлены недостатки существующих аналогов. Для создания интерфейса использованы фреймворк Vue.js 3, среда разработки Vite, а также инструменты Figma для прототипирования. Веб-ресурс предоставляет пользователям возможность просматривать конференции, регистрироваться, подавать заявки на участие и отслеживать их статус, а организаторам управлять заявками и данными о мероприятии. Разработанный интерфейс отличается удобством и современным визуальным стилем.

Ключевые слова: веб-ресурс, управление конференциями, пользовательский интерфейс, UX/UI, Vue.js, прототипирование, юзабилити-тестирование

L. E. Mamonova^{1✉}, A. A. Sharapov¹

Design and development of a user interface for a scientific conference management web resource using the Vue.js framework and UX/UI principles

¹Siberian State University of Geosystems and Technologies, Novosibirsk, Russian Federation
e-mail: pochta@mail.ru

Abstract. The article presents the main results of the design and development of a user interface for a scientific conference management web resource. The analysis of the subject area is performed, in which the main processes of conference organization are considered, user roles are identified, and the shortcomings of existing analogues are identified. The Vue.js 3 framework, Vite development environment, and Figma prototyping tools were used to create the interface. The web resource provides users with the ability to view conferences, register, submit applications and track their status, and organizers to manage applications and event data.

Keywords: web resource, conference management, user interface, UX/UI, Vue.js, prototyping, usability testing

Введение

Разработка веб-ресурсов в настоящее время связана не только с созданием страниц, но и с продумыванием удобного взаимодействия пользователя с системой [1]. Особенно это заметно в тех случаях, когда сайт используется не просто для просмотра информации, а для выполнения конкретных действий. К таким системам можно отнести веб-ресурс для управления научными конференциями.

В нём пользователь должен не только прочитать сведения о мероприятии, но и зарегистрироваться, подать заявку, просмотреть её статус, а организатор обязан обработать поступившие данные и управлять конференцией.

Научные конференции обычно требуют достаточно большой организационной работы. Нужно заранее опубликовать информацию о мероприятии, указать сроки проведения, направления работы, требования к заявкам и материалам. После этого участники отправляют свои данные, выбирают конференцию, подают заявку и ждут решения. Организатор, в свою очередь, должен просматривать список участников, проверять заявки, менять их статусы и при необходимости редактировать информацию о мероприятии. Если все эти действия выполняются через почту, таблицы и отдельные документы, процесс становится менее удобным [2]. Часть данных может дублироваться, обновляться с задержкой или теряться среди других сообщений.

Исследование предметной области

Управление научной конференцией включает несколько этапов: анонсирование мероприятия, приём заявок от участников, проверку поступивших материалов, рецензирование, формирование программы и информирование авторов о результатах [3]. Если эти процессы выполняются разрозненно - через электронную почту, таблицы и отдельные документы - возрастает риск потери данных, задержек и дублирования информации. Особенно остро эта проблема проявляется при росте числа участников и появлении нескольких секций или параллельных мероприятий.

В предметной области выделяются четыре основные роли пользователей. Участник просматривает открытую информацию о конференциях и регистрироваться. Секретарь просматривает входящие заявки. Организатор управляет секретарями и формирует итоговую программу. Администратор контролирует пользователей и общие данные системы [1]. Каждая из этих ролей требует своего набора функций и своего представления данных, поэтому интерфейс должен адаптироваться под конкретного пользователя.

Анализ существующих решений (EasyChair, ConfTool, а также российских платформ Science Index и Leader-ID) показывает, что большинство из них либо имеют устаревший и перегруженный интерфейс, либо не автоматизируют полный цикл управления конференцией [4]. Это обосновывает необходимость разработки веб-ресурса, который объединит все ключевые процессы в одном месте, будет понятен пользователям с разным уровнем подготовки и обеспечит удобную работу как с компьютера, так и с мобильных устройств.

Методы и технологии

При проектировании и разработке пользовательского интерфейса веб-ресурса для управления научными конференциями применялся комплекс методов, ориентированных на конечного пользователя. В основе подхода лежат принципы человеко-центрированного проектирования (Human-Centered Design), согласно которым структура и функциональность интерфейса определяются реальными

задачами и сценариями работы пользователей [5]. На этапе анализа были выделены четыре основные роли (участник, секретарь организатор, администратор), для каждой из которых построены сценарии использования и карты пользовательских путей. Для проектирования информационной архитектуры использовался метод построения карты сайта, обеспечивающий глубину навигации не более трёх кликов до любой целевой страницы [1]. Визуальное проектирование и прототипирование выполнены в Figma, что позволило последовательно пройти этапы от низкодетализированных схем до кликабельных макетов, включая согласование цветовой схемы и типографики.

Для реализации клиентской части веб-ресурса выбран фреймворк Vue.js 3 с использованием Composition API [6]. Выбор обусловлен рядом факторов: компонентный подход позволяет разбить интерфейс на переиспользуемые блоки (карточки конференций, формы заявок, таблицы, модальные окна); реактивность автоматически обновляет интерфейс при изменении данных без дополнительного кода; встроенная маршрутизация (Vue Router) упрощает настройку навигации и привязку прав доступа к ролям пользователей. Для управления состоянием приложения используется Pinia, обеспечивающая централизованное хранение данных о пользователях, конференциях и заявках. Сборка проекта выполняется через Vite, что даёт высокую скорость разработки и горячую перезагрузку модулей.

Стилизация интерфейса реализована на кастомном CSS без использования внешних UI-библиотек. Это позволило получить полный контроль над дизайном, обеспечить точное соответствие утверждённым макетам и минимизировать объём загружаемого кода.

Результаты

В результате работы был создан пользовательский интерфейс веб-ресурса для управления научными конференциями, в котором реализованы следующие функции (рис. 1–3):

- просмотр списка конференций с фильтрацией по статусу (активные/архивные) и датам;
- регистрация и авторизация пользователей с проверкой уникальности email;
- подача заявки на конференцию (выбор конференции, заполнение ФИО, email, темы доклада и секции);
- личный кабинет участника с отображением поданных заявок и их статусов («на рассмотрении», «принята», «отклонена»);
- личный кабинет организатора со списком заявок и возможностью изменять их статус;
- панель администратора для управления пользователями (блокировка, смена роли);
- адаптивная вёрстка, корректно отображающаяся на экранах от 320 до 1920 пикселей [2].



Рис. 1. Интерфейс главной страницы со списком конференций

РЕГИСТРАЦИЯ НА КОНФЕРЕНЦИЮ

Выберите секцию
ИНТЕР ЭКСПО ГЕО-Сибирь 2025

Выберите подсекцию
Выберите подсекцию

Введите название доклада
Роль цифровых карт в исследовании городской среды

Описание доклада
Краткое описание работы и ключевые тезисы выступления

Прикрепить файл с отчетом о докладе
report_geo_siberia_2025.docx

АВТОР

Фамилия
Иванова

Имя
Мария

Отчество
Александровна

Организация (полное наименование)
Организация (полное наименование)

Организация (краткое наименование)
Организация (краткое наименование)

Кафедра/отделение/лаборатория
Кафедра/отделение/лаборатория

Рис. 2. Интерфейс формы подачи заявки

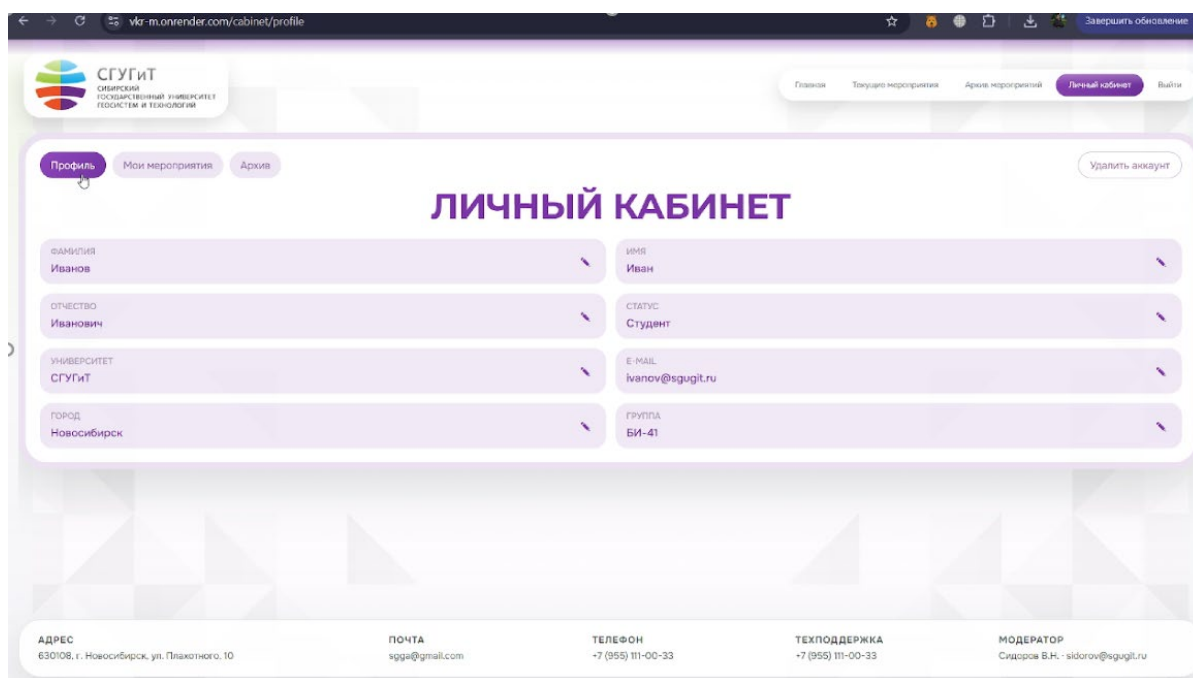


Рис. 3. Интерфейс авторизованного аккаунта

Заключение

В результате выполнения работы был спроектирован и разработан пользовательский интерфейс веб-ресурса для управления научными конференциями. Интерфейс поддерживает четыре роли пользователей (участник, секретарь, организатор, администратор), позволяет просматривать конференции, регистрироваться, подавать заявки и отслеживать их статус. Разработанный ресурс может использоваться как основа для дальнейшего создания полноценной системы сопровождения научных конференций. Проект может быть расширен за счёт подключения серверной части, базы данных, системы уведомлений и загрузки файлов. Практическая значимость работы подтверждена актом внедрения на кафедре информационных систем.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Круг С. Веб-дизайн: не заставляйте меня думать. СПб.: Символ-Плюс, 2010. -232 с.
2. Норман Д. Дизайн привычных вещей. - М.: Манн, Иванов и Фербер, 2012. - 384 с.
3. ISO 9241-210:2019. Ergonomics of human-system interaction - Part 210: Human-centred design for interactive systems. - Geneva: ISO, 2019.
4. Нильсен Я., Лоранжер Х. Web-дизайн: удобство использования веб-сайтов. - М.: Вильямс, 2009. - 368 с
5. Brooke J. SUS: A Quick and Dirty Usability Scale // Usability Evaluation in Industry. - London: Taylor & Francis, 1996. - P. 189–194.
6. Vue.js Documentation. - Режим доступа: <https://vuejs.org>.

© Л. Э. Мамонова, А. А. Шаранов, 2026

И. В. Махорин¹✉

Метод адаптивной оконной фильтрации при расчете квантильного порога в задачах обнаружения инсайдерской активности

¹Национальный исследовательский ядерный университет «МИФИ»,
г. Москва, Российская Федерация
e-mail: ilyamakhorin@mail.ru

Аннотация. В статье рассматривается задача обнаружения инсайдерской активности в корпоративных вычислительных сетях в условиях нестационарности поведенческих данных. Показано, что использование квантильных порогов на скользящем окне фиксированного размера приводит к снижению устойчивости методов обнаружения аномалий при изменении характеристик входного потока. Предложен метод адаптивной оконной фильтрации, в котором размер окна определяется динамически на основе локальной вариативности и интенсивности данных. Представлена математическая формализация метода и приведено вероятностное обоснование его устойчивости. Анализ показывает, что предложенный подход позволяет повысить устойчивость оценки квантильного порога и улучшить адаптацию к изменяющимся условиям. Метод ориентирован на потоковую обработку данных и может быть применен в системах UEBA и других задачах кибербезопасности.

Ключевые слова: обнаружение инсайдерских угроз, квантильный порог, адаптивное окно, поведенческий анализ, UEBA, кибербезопасность

I. V. Makhorin¹✉

Adaptive window filtering method for quantile threshold estimation in insider threat detection

¹National Research Nuclear University «MEPhI», Moscow, Russian Federation
e-mail: ilyamakhorin@mail.ru

Abstract. This paper addresses the problem of insider threat detection in corporate information systems under conditions of non-stationary behavioral data. It is shown that the use of quantile thresholds based on a fixed-size sliding window leads to reduced robustness of anomaly detection methods when the characteristics of the input data stream change. An adaptive window filtering method is proposed, in which the window size is dynamically adjusted based on local variability and event intensity. A mathematical formulation of the method is presented, along with a probabilistic justification of its robustness. The analysis shows that the proposed approach improves the stability of quantile threshold estimation and enhances adaptation to changing conditions. The method is designed for streaming data processing and can be applied in UEBA systems and other cybersecurity tasks.

Keywords: insider threat detection, quantile threshold, adaptive window, behavioral analysis, UEBA, cybersecurity

Введение

В современных корпоративных вычислительных сетях задача обнаружения инсайдерских угроз приобретает особую актуальность в связи с ростом объемов

обрабатываемых данных и усложнением поведенческих паттернов пользователей. В отличие от внешних атак, инсайдерская активность характеризуется наличием легитимных прав доступа, что существенно затрудняет ее выявление традиционными средствами информационной безопасности. В этой связи все большее распространение получают методы поведенческого анализа и обнаружения аномалий, лежащие в основе систем класса User and Entity Behavior Analytics (UEBA) [1–3]. Современные исследования также демонстрируют активное применение методов машинного обучения и глубокого обучения для анализа поведения пользователей и выявления скрытых признаков инсайдерской активности [4]. При этом отдельное внимание уделяется развитию UEBA-моделей, ориентированных на выявление аномалий в корпоративных вычислительных сетях [5].

Одним из распространенных подходов к обнаружению аномалий является использование статистических порогов, в частности квантильных оценок распределения наблюдаемых признаков [6, 7]. Пусть задан временной ряд наблюдений, отражающий интенсивность или характеристики активности пользователя. Тогда аномальные события могут определяться как значения, превышающие некоторый квантильный порог, вычисляемый на основе исторических данных. Однако на практике такие методы сталкиваются с существенными ограничениями [8, 9].

Во-первых, поведение пользователей в корпоративной сети является нестационарным: распределение признаков может изменяться во времени под влиянием рабочих процессов, сезонных факторов и внешних условий [10, 11]. Во-вторых, интенсивность событий может значительно варьироваться, что приводит к нестабильности оценок квантилей при использовании фиксированного окна наблюдения. В результате возникают ситуации, когда порог либо избыточно чувствителен (что увеличивает число ложноположительных срабатываний), либо, наоборот, теряет способность выявлять аномалии. Исходя из этого, возникает необходимость в разработке методов, способных адаптироваться к изменяющимся характеристикам входных данных и обеспечивать устойчивую оценку пороговых значений. Одним из перспективных направлений является использование адаптивных оконных методов, в которых параметры окна наблюдения (размер, положение) изменяются в зависимости от текущих свойств временного ряда [12–14].

В данной работе предлагается метод адаптивной оконной фильтрации при расчете квантильного порога, направленный на повышение устойчивости и точности обнаружения аномальной активности. В отличие от классических методов, основанных на использовании фиксированного окна наблюдения, в предлагаемом подходе размер окна определяется динамически на основе локальных статистических характеристик временного ряда. Это позволяет учитывать нестационарность процесса и адаптировать оценку квантильного порога к текущим условиям функционирования системы.

Методы и материалы

В рамках настоящего исследования рассматривается стохастический процесс наблюдений:

$$X_{t=1}^T, \quad (1)$$

где $X_t \in R$ представляет собой значение признака, характеризующего активность пользователя или системы в момент времени t . Предполагается, что процесс является локально нестационарным, то есть распределение случайной величины X_t задается семейством функций распределения $F_t(x)$, изменяющихся во времени. Для обнаружения аномальной активности используется квантильный подход. Квантиль уровня $\alpha \in (0,1)$ определяется как:

$$q_\alpha(t) = \inf \inf \{x \in R: F_t(x) \geq \alpha\}, \quad (2)$$

и служит пороговым значением, превышение которого интерпретируется как потенциальная аномалия. Поскольку истинное распределение $F_t(x)$ неизвестно, используется его эмпирическая оценка, построенная по выборке наблюдений. В классическом подходе применяется скользящее окно фиксированного размера, однако в условиях нестационарности такой подход приводит к компромиссу между смещенностью и дисперсией оценки. Для устранения данного ограничения в работе предлагается метод адаптивной оконной фильтрации, в рамках которого размер окна наблюдения изменяется во времени в зависимости от локальных характеристик процесса.

Пусть окно наблюдения переменной длины w_t :

$$W_t = X_{t-w_t+1}, \dots, X_t. \quad (3)$$

Тогда эмпирическая функция распределения определяется как:

$$\hat{F}_t(x) = \frac{1}{w_t} \sum_{i=t-w_t+1}^t 1\{X_i \leq x\} \quad (4)$$

а оценка квантиля принимает вид:

$$\widehat{q}_\alpha(t) = \inf \{x: \hat{F}_t(x) \geq \alpha\}. \quad (5)$$

Ключевым элементом предлагаемого метода является механизм адаптации размера окна. Размер окна определяется как функция локальной вариативности и интенсивности потока событий:

$$w_t = w_0 \cdot (1 + \beta \sigma_t)^{-1} \cdot (1 + \gamma \lambda_t)^{-1}, \quad (6)$$

где w_0 – базовый размер окна, β и γ – параметры чувствительности.

Локальная вариативность σ_t оценивается через стандартное отклонение значений в текущем окне, что позволяет учитывать степень разброса данных. Ин-

тенсивность потока событий λ_t определяется как среднее число наблюдений, удовлетворяющих заданному условию активности, в фиксированном интервале времени. Тем самым, при увеличении вариативности или плотности событий размер окна автоматически уменьшается, обеспечивая быструю реакцию на изменения, а при стабильном поведении увеличивается, снижая влияние случайных флуктуаций. Аномальная активность фиксируется при превышении текущим наблюдением адаптивного квантильного порога:

$$X_t > \hat{q}_\alpha(t) . \quad (7)$$

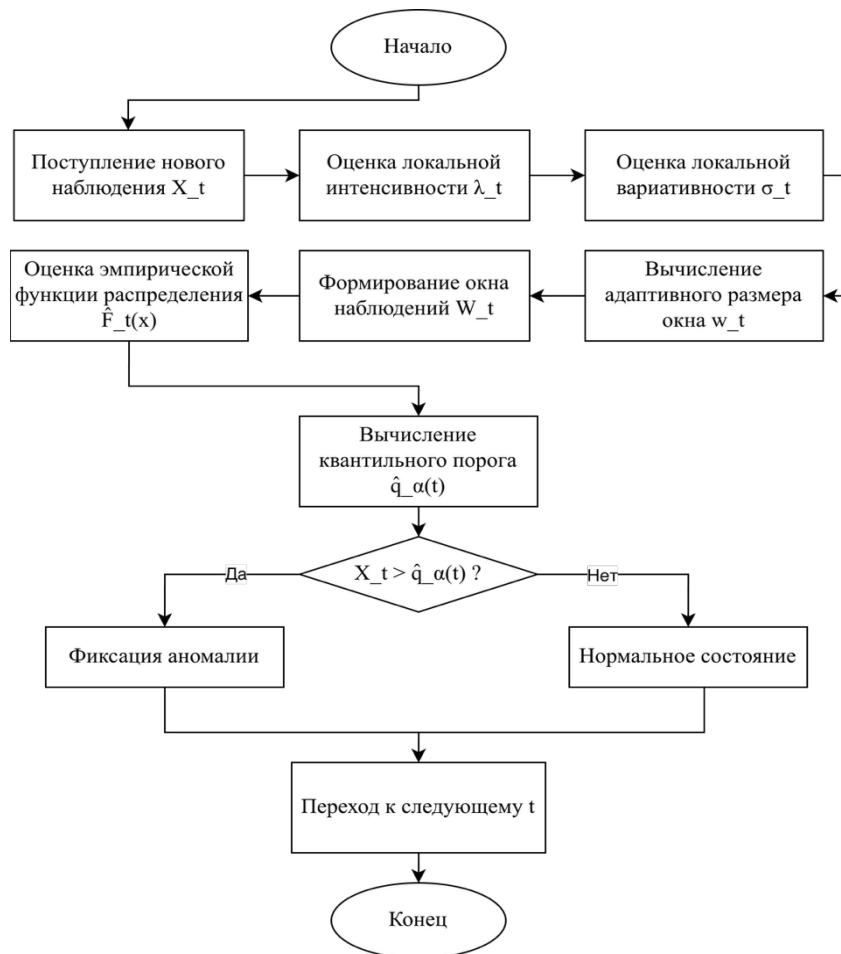


Рис. 1. Блок-схема метода адаптивной оконной фильтрации при расчете квантильного порога

Реализация метода осуществляется в виде последовательной обработки временного ряда, в рамках которого на каждом шаге производится оценка локальных характеристик, пересчет размера окна и обновление квантильного порога. Предложенный подход ориентирован на использование в потоковых системах анализа событий и не требует хранения полной истории наблюдений, что обеспечивает его вычислительную эффективность.

Визуализация метода представлена на рис. 1. На каждом шаге обработки входного потока данных производится оценка локальных характеристик временного ряда, на основе которых динамически определяется размер окна наблюдения. Далее формируется выборка для оценки эмпирической функции распределения и вычисляется квантильный порог. Принятие решения об аномалии осуществляется путем сравнения текущего значения с адаптивным порогом. Такой подход обеспечивает онлайн-обработку данных и позволяет оперативно реагировать на изменения в поведении системы.

Вероятностное обоснование устойчивости метода

Для анализа свойств предложенного метода рассмотрим ошибку оценки квантильного порога, определяемую как:

$$\varepsilon_t = |\widehat{q_\alpha}(t) - q_\alpha(t)|. \quad (8)$$

В классическом случае фиксированного окна w при предположении стационарности процесса выполняется сходимость оценки квантиля по вероятности. Однако в условиях локальной нестационарности возникает дополнительная ошибка, связанная с изменением распределения во времени. Суммарная ошибка может быть представлена в виде:

$$E[\varepsilon_t^2] = Bias^2(w_t) + Var(w_t), \quad (9)$$

где первая составляющая определяется скоростью изменения распределения $F_t(x)$, а вторая – объемом выборки.

При использовании фиксированного окна уменьшение дисперсии достигается за счет увеличения w , однако это приводит к росту смещенности в условиях нестационарности. В предложенном методе данный компромисс разрешается за счет адаптивного выбора размера окна.

Предположим, что семейство распределений $F_t(x)$ удовлетворяет условию Липшица:

$$|F_t(x) - F_{t-1}(x)| \leq L, \quad (10)$$

где L – константа, характеризующая скорость изменения процесса. Тогда оптимальный размер окна, минимизирующий среднеквадратичную ошибку, зависит от локальной скорости изменения распределения.

В предлагаемом методе эта зависимость аппроксимируется через локальную вариативность σ_t и интенсивность λ_t , что позволяет адаптивно приближаться к оптимальному значению w_t . В результате достигается снижение ошибки оценки квантиля по сравнению с фиксированным окном:

$$E[\varepsilon_t^2]_{adaptive} \leq E[\varepsilon_t^2]_{fixed}. \quad (11)$$

Таким образом, предложенный метод обеспечивает более устойчивую оценку порогового значения в условиях нестационарности данных.

Обсуждение

Проведенный анализ показал, что использование адаптивной оконной фильтрации при оценке квантильного порога обеспечивает более устойчивые результаты по сравнению с классическим подходом, основанным на фиксированном размере окна. Ключевое преимущество метода заключается в способности учитывать локальные изменения характеристик временного ряда, что особенно важно в условиях нестационарности поведенческих данных.

Анализ механизма адаптации демонстрирует, что при увеличении вариативности и интенсивности событий размер окна уменьшается, обеспечивая более оперативную реакцию на изменения распределения. В условиях стабильного поведения, напротив, окно увеличивается, что снижает влияние случайных флуктуаций. Таким образом, достигается баланс между чувствительностью метода и устойчивостью оценки квантильного порога.

Сравнение с классическим подходом показывает, что использование фиксированного окна может приводить к запаздыванию адаптации порогового значения и увеличению числа ложноположительных срабатываний, тогда как предложенный метод обеспечивает более гибкую реакцию на изменения данных. В то же время эффективность метода зависит от выбора параметров адаптации и способа оценки локальных характеристик временного ряда. Кроме того, практическая эффективность подобных методов во многом определяется качеством исходных данных, полнотой журналирования событий и архитектурой системы мониторинга, обеспечивающей непрерывный сбор и обработку телеметрии [15–17].

Заключение

В работе предложен метод адаптивной оконной фильтрации при расчёте квантильного порога для задач обнаружения инсайдерской активности. Показано, что предложенный подход позволяет повысить устойчивость оценки и улучшить адаптацию к изменяющимся условиям по сравнению с методами с фиксированным окном.

Практическая значимость метода заключается в возможности его применения в системах анализа поведения пользователей и сущностей (UEBA), а также в других задачах обнаружения аномалий в потоковых данных.

В качестве направлений дальнейших исследований целесообразно рассмотреть методы автоматической настройки параметров адаптации, а также расширение предложенного подхода на многомерные признаки.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Kantchelian A., Neo C., Stevens R. Facade: High-precision insider threat detection using deep contextual anomaly detection //arXiv. – 2024. – С. 1–15.
2. Kamatchi K., Ramesh S. Insights into user behavioral-based insider threat detection: A systematic review //International Journal of Information Security. – 2025. – С. 1–20.

3. Parimalla S. Hunting the invisible: Harnessing UEBA to unmask insider threats //IntechOpen. – 2025. – С. 1–20.
4. Ofori H. K., Mensah K. Machine learning and deep learning for insider threat detection: A bibliometric review //Symmetry. – 2025. – Т. 17. – №. 10. – С. 1–20.
5. Song Y., Chen J. Insider threat detection based on UEBA models //Lecture Notes in Computer Science. – 2024. – С. 210–225.
6. Ibraheem I. O., Morufat A. T., Segbefia S. K. Detecting malicious insider threats through anomaly-based user behaviour analytics in enterprise networks //Southern African Journal of Security. – 2025. – С. 1–23.
7. Koli L., Kalra S., Thakur R. AI-driven insider risk management with adaptive scoring and behavioral analytics //arXiv. – 2025. – С. 1–18.
8. Feng W., Liu Y., Zhang X. Multi-granularity user anomalous behavior detection //Applied Sciences. – 2024. – Т. 15. – №. 1. – С. 128–140.
9. Ahmad W., Khan M. Leveraging machine learning for internal user behaviour analytics in cybersecurity //International Journal of Information Technology and Information Systems. – 2025. – С. 1–15.
10. Cain J., Beadles H. Mask-based window-level insider threat detection for campaign discovery //arXiv. – 2026. – С. 1–12.
11. Tian T., Li H., Zhou X. Insider threat detection for specific threat scenarios //Journal of Big Data. – 2025. – С. 1–18.
12. A-UEBA: An anomaly heterogeneity learning framework for user and entity behavior analytics //Proceedings of ACM Conference. – 2026. – С. 1–10.
13. Ye X., Zhang Y., Wang Z. Insider threat detection using hybrid deep learning models //Scientific Reports. – 2025. – Т. 15. – С. 1–12.
14. Huang Z., Tang X., Li H. TabSec: A collaborative framework for insider threat detection //arXiv. – 2024. – С. 1–14.
15. Muñoz D. A., Homoliak I. Dataset generation for insider threat detection systems //Computers & Security. – 2025. – С. 1–14.
16. Басыня Е. А., Худяков Д. С. Комплексный мониторинг информационной инфраструктуры предприятия //Защита информации. Инсайд. – 2020. – №. 6. – С. 28–33.
17. Стрельцов А. С., Французова Г. А., Басыня Е. А. Разработка системы сбора, обработки и анализа событий информационной инфраструктуры предприятия //Системы анализа и обработки данных. – 2023. – №. 1 (89). – С. 101–113.

© И. В. Махорин, 2026

Г. М. Милькович¹✉, И. А. Дубров¹, А. А. Шарапов¹

Создание интерактивной образовательной веб-платформы проектирования и программирования робототехнических систем с применением искусственного интеллекта

¹Сибирский государственный университет геосистем и технологий,
г. Новосибирск, Российская Федерация
e-mail: milkovichgleb@mail.ru

Аннотация. В статье представлены основные результаты проектирования и разработки комплексной образовательной веб-платформы для дистанционного изучения робототехники. Выполнен анализ предметной области, в котором выявлен дефицит отечественных программных решений для виртуального моделирования электрических цепей. Для создания серверной архитектуры и системы управления обучением применялся фреймворк NestJS и СУБД PostgreSQL, клиентский интерфейс спроектирован на базе библиотеки React и среды Next.js. В платформу интегрированы модули симуляции микроконтроллеров Arduino и автоматизированной проверки исходного кода на языке C++. Веб-платформа предоставляет пользователям интерактивное рабочее пространство для конструирования схем, преподаватели получают функционал для создания собственных уникальных курсов и автоматическую проверку практических заданий. Готовый продукт исключает зависимость от зарубежных аналогов и экономит бюджет учебных заведений на закупку физических компонентов.

Ключевые слова: образовательная веб-платформа, микроконтроллеры Arduino, робототехника, дистанционное обучение, симуляция схемотехники, автоматизированная проверка кода

G. M. Milkovich¹✉, I. A. Dubrov¹, A. A. Sharapov¹

Development of an interactive educational web platform for the design and programming of robotic systems using artificial intelligence

¹Siberian State University of Geosystems and Technologies, Novosibirsk, Russian Federation
e-mail: milkovichgleb@mail.ru

Abstract. The article presents the main results of the design and development of a comprehensive educational web platform for distance learning in robotics. An analysis of the subject area was conducted, which revealed a shortage of domestic software solutions for the virtual simulation of electrical circuits. The NestJS framework and the PostgreSQL DBMS were used to create the server architecture and learning management system, the client interface was designed using the React library and the Next.js environment. Modules for simulating Arduino microcontrollers and for automated verification of source code in C++ are integrated into the platform. The web platform provides users with an interactive workspace for designing circuits, instructors gain the ability to create their own unique courses and automatically grade practical assignments. The finished product eliminates dependence on foreign counterparts and saves educational institutions money on the purchase of physical components.

Keywords: educational web platform, Arduino microcontrollers, robotics, distance learning, circuit simulation, automated code verification

Введение

Подготовка современных инженерных кадров неразрывно связана с освоением микроконтроллерной техники, традиционной базой для практических занятий служат платы семейства Arduino. Изучение робототехники требует постоянного взаимодействия с физическими компонентами, однако использование реальных деталей приводит к регулярным проблемам. Компоненты изнашиваются, а контакты перегорают из-за неверного подключения. Закупка и поддержание материальной базы требует значительного финансирования учебных классов [1].

Развитие технологий дистанционного образования выявило недостаток гибкости классического образовательного процесса [2]. Проводить полноценные лабораторные работы удаленно без специализированных сред практически невозможно [3]. Рынок предлагает ряд зарубежных симуляторов, таких как Tinkercad и Wokwi, но использование данных сервисов в российских образовательных учреждениях сопряжено с критическими рисками: блокировки доступа и невозможность официальной оплаты подписок делают учебный процесс нестабильным [4]. К тому же, большинство аналогов имеют закрытый исходный код и не позволяют интегрировать в свои платформы систему управления обучением (СУО) для контроля успеваемости обучающихся.

Целью данного исследования является проектирование и разработка отечественной образовательной веб-платформы. Создаваемый ресурс должен объединить теоретическую базу, виртуальную сборку электрических схем и модуль автоматизированной проверки кода. Разработка такого продукта позволит школам и вузам выстраивать учебный процесс автономно, исключая риск внешних технических сбоев.

Методы и материалы

Базовым каркасом выступила клиент-серверная архитектура, вычислительная нагрузка на устройства обучающихся была минимизирована [5]. Серверная часть базируется на фреймворке NestJS, вся логика разбита на независимые модули: отдельно обрабатываются HTTP-запросы, отдельно контролируются пользовательские сессии. Информация хранится в реляционной СУБД PostgreSQL, структура учебных курсов жестко зафиксирована. Рутинные запросы к таблицам автоматизированы с помощью инструмента Prisma ORM [6].

Клиентская сторона спроектирована на базе React и Next.js, перенос рендеринга на сторону сервера дал существенный прирост в скорости – браузер принимает уже готовый HTML-код. Нестабильная работа на слабых учебных компьютерах исключена. Холст с электрическими цепями генерируется средствами React Flow, глобальные состояния интерфейса надежно удерживаются менеджерам Zustand. Структура взаимодействия программных узлов представлена на схеме (рис. 1).

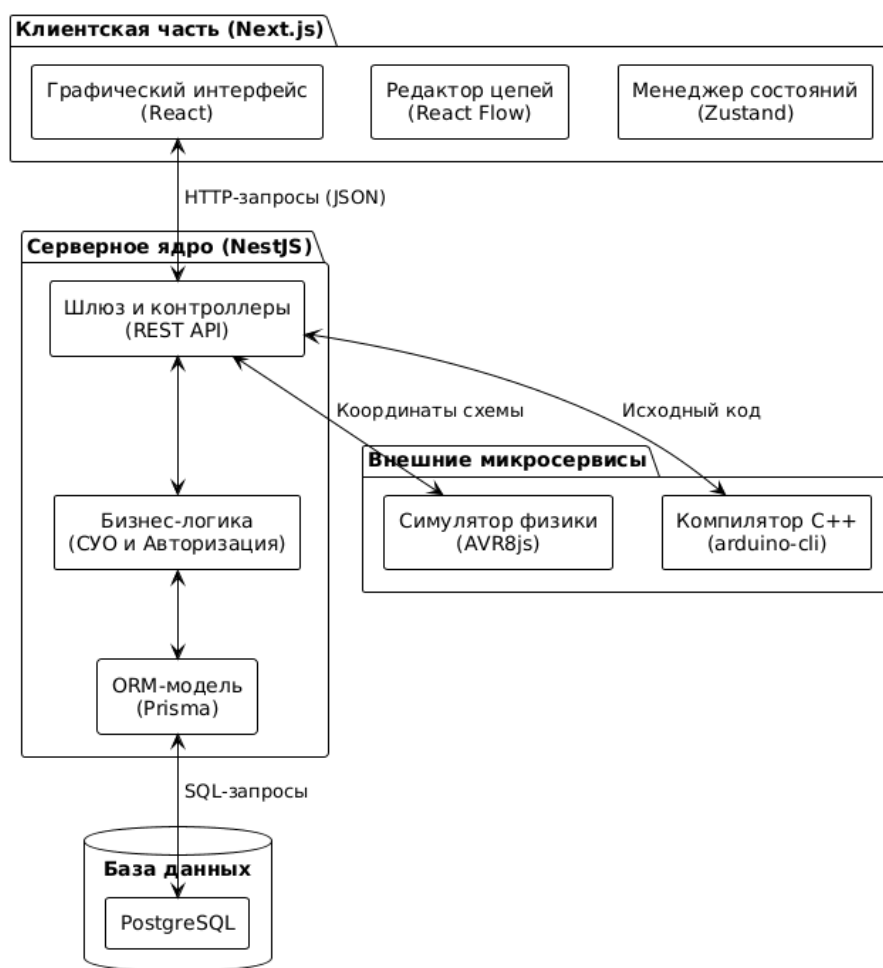


Рис. 1. Архитектура программного взаимодействия компонентов платформы

Для работы с виртуальными платами развернут независимый микросервис, физика процессов внутри цепей рассчитывается программным движком AVR8js. Задача по компиляции C++ скриптов передана консольной утилите Arduino-Cli. Связующим звеном для сетевого обмена выступает архитектура REST API, вся информация пересылается исключительно в текстовом формате JSON.

Результаты

В результате проделанной работы был создан комплексный программный продукт, получивший название «Voltrix». Платформа разделяет пользователей на три основные ролевые группы: гости, обучающиеся и преподаватели, каждая группа наделена строгим уровнем прав доступа [7].

Для обучающихся реализовано интерактивное рабочее пространство с разделенным экраном браузера на функциональные зоны. Пользователь имеет доступ к библиотеке электронных компонентов: микроконтроллеры, макетные платы, датчики, сервоприводы и т. п. Детали переносятся на центральный холст методом визуального перетаскивания Drag-and-Drop. При переключении вкладки с выбора деталей на «Код», пользователю предоставляется встроенный редактор программ-

ного кода с терминалом последовательного порта. Интерфейс рабочего пространства симулятора представлен на рис. 2.

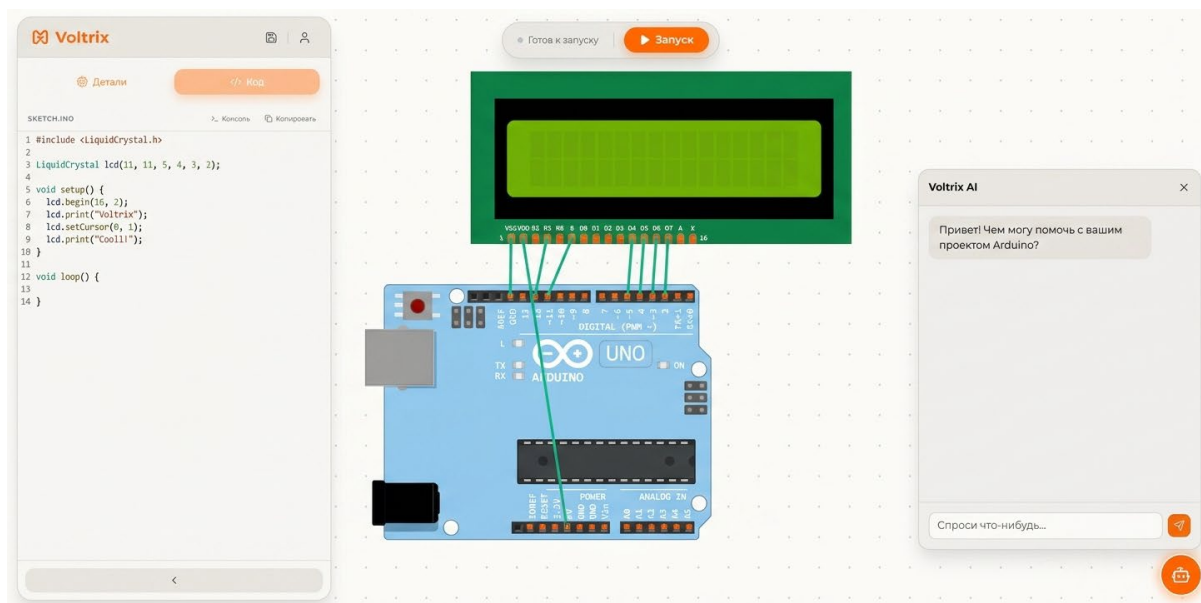


Рис. 2. Интерфейс рабочего пространства симулятора

Для преподавателей спроектирован модуль конструирования курсов (рис. 3). Система управления обучением позволяет формировать структуру образовательной дисциплины, учитель создает теоретические шаги и практические задания. Условия успешного прохождения задачи кодируются в строгие правила формата JSON, при запуске симуляции алгоритм автопроверки считывает этот блок и сверяет его с текущей цепью обучающегося.

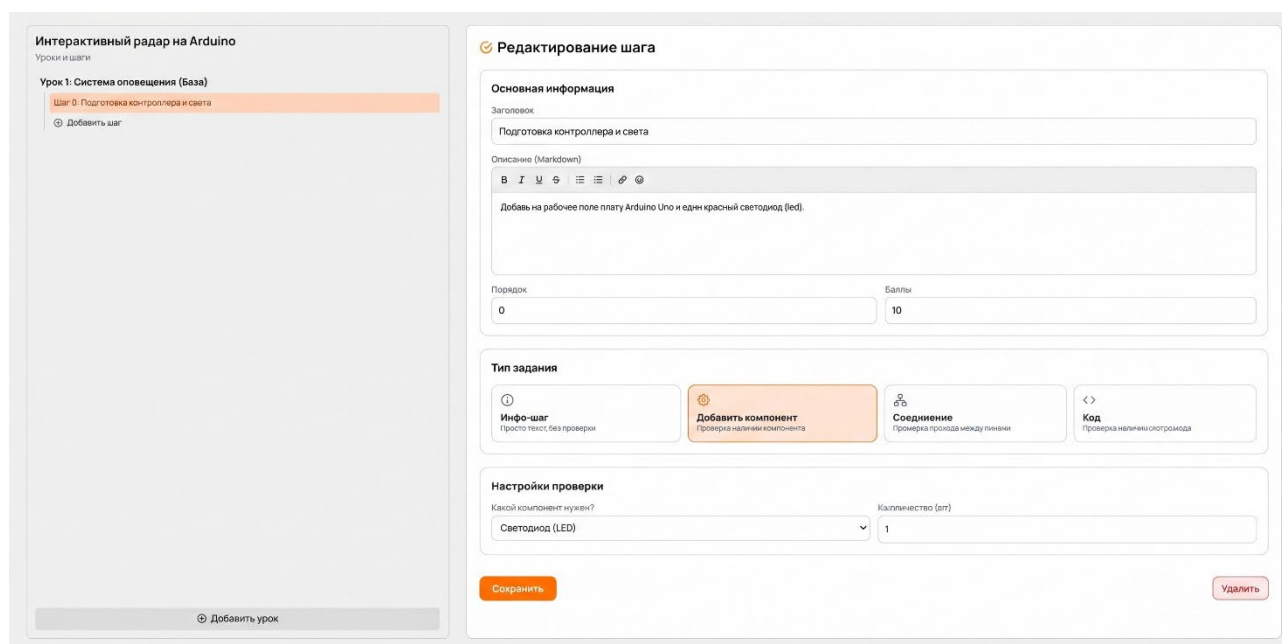


Рис. 3. Панель настройки практического задания (шага)

Система автоматизирует рутинные процессы. Клиентский скрипт собирает координаты расставленных компонентов, связи проводов и текст программы в единый массив. Сформированный пакет маршрутизируется в модуль компиляции, в случае неверного подключения контактов платформа мгновенно выводит уведомление об ошибке. При корректной сборке сервер обновляет статистику в базе данных и автоматически начисляет баллы студенту.

Обсуждение

Анализ конкурентоспособности созданного продукта подтверждает его высокую эффективность. Популярные зарубежные сервисы предоставляют качественную визуализацию, однако их функционал ограничивается форматом виртуальной песочницы. Образовательным учреждениям приходится использовать дополнительные платформы для ведения успеваемости обучающихся.

Разработанная веб-платформа решает данную проблему комплексно. Теория, сборка схем и программирование происходят в одном окне браузера. Внедрение функции автоматизированной проверки кода избавляет преподавателей от необходимости визуально анализировать каждую студенческую работу. Время на проверку домашних заданий сокращается кратно. Сравнительная характеристика существующих решений приведена в табл. 1.

Таблица 1. Сравнение образовательных веб-платформ

Критерий оценки	Зарубежные симуляторы	Текстовые системы	Разработанная платформа
Наличие встроенной СУО	Нет	Да	Да
Виртуальная сборка цепей	Да	Нет	Да
Автоматическая проверка	Нет	Частично	Да
Риск блокировок и санкций	Высокий	Отсутствует	Отсутствует

Для оценки экономической целесообразности внедрения разработанной платформы применяется математическая модель расчета сокращения финансовых издержек учебного заведения. Вычисление коэффициента экономической эффективности (K_{eff}) рассчитывается по формуле:

$$K_{eff} = \frac{n \cdot (C_{hw} + C_{fail})}{C_{soft}}, \quad (1)$$

где n – количество оборудуемых учебных мест;

C_{hw} – базовая стоимость физического набора микроконтроллера и датчиков;

C_{fail} – средние ежегодные затраты на замену сломанных компонентов;

C_{soft} – стоимость годовой лицензии на использование веб-платформы.

Проведенные расчеты показывают, что при увеличении числа студентов значение коэффициента многократно возрастает, закупка физического оборудования требует линейного увеличения бюджета. Внедрение программной симуляции сводит издержки на амортизацию и замену сгоревших компонентов к нулю.

Дополнительным коммерческим и академическим преимуществом является независимость от санкционных рисков, так как серверная архитектура базируется на отечественных облачных мощностях. Разработанный продукт позволяет полностью отказаться от закупок физических микроконтроллеров и расходных радиодеталей, что ведет к значительной экономии государственного бюджета.

Заключение

Проектирование и разработка образовательной веб-платформы успешно завершены. Итоговый продукт объединяет клиентские интерфейсы, серверную архитектуру, базу данных и независимый модуль симуляции микроконтроллеров. Технология автоматизированной проверки кода доказала свою отказоустойчивость в ходе тестирования.

Созданный программный комплекс полностью удовлетворяет потребности преподавателей и студентов в качественном инструменте для дистанционного изучения робототехники. Платформа готова к внедрению в реальный учебный процесс государственных школ, технопарков и высших учебных заведений.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Зайцева, С. А. Развитие образовательной робототехники: проблемы и перспективы / С. А. Зайцева, В. В. Иванов, В. С. Киселев, А. Ф. Зубаков // Образование и наука. – 2022. – Т. 24, № 2. – С. 84–115 (дата обращения: 19.11.2025). – Текст : непосредственный.
2. Пьянова, Н. В. Перспективы использования дистанционного обучения в образовательной робототехнике / Н. В. Пьянова, А. И. Попова, Е. А. Алекса // Ученые записки Орловского государственного университета. Серия: Гуманитарные и социальные науки. – 2022. – С. 263–266 (дата обращения: 14.12.2025). – Текст : непосредственный.
3. Едренкина, М. В. Возможности применения дистанционных образовательных технологий в обучении конструированию в условиях дополнительного образования / М. В. Едренкина // Кронос. – 2021. – № 1 (51). – С. 21–23. – Текст : непосредственный.
4. Сатлер, О. Н. Использование цифровых инструментов Arduino и Tinkercad в условиях дистанционного формата обучения / О. Н. Сатлер, О. В. Капралов // Проблемы современного педагогического образования. – 2025. – Текст : непосредственный.
5. Что такое клиент-серверная архитектура // Work Systems :[блог]. – URL: <https://work-systems.ru/blog/chto-takoe-klient-servernaya-arhitektura> (дата обращения: 04.02.2026). – Текст : электронный.
6. Обзор Prisma ORM: как забыть об SQL и сосредоточиться на данных // Хабр : [информационный портал]. – URL: <https://habr.com/ru/companies/macloud/articles/554952/> (дата обращения: 07.02.2026).
7. Литченко, С. П. Сравнение ПО для моделирования схем / С. П. Литченко, Д. В. Тарасов // Вестник науки. – 2026. – Текст : непосредственный.

© Г. М. Милькович, И. А. Дубров, А. А. Шаранов, 2026

В. В. Михайлюк¹✉, Е. Ю. Воронкин¹

Разработка Python-фреймворка для автоматизации релизных процессов с интеграцией в CI/CD-систему Jenkins

¹Сибирский государственный университет геосистем и технологий,
г. Новосибирск, Российская Федерация
e-mail: vitia.mikhailiuk@mail.ru

Аннотация. В статье рассматривается процесс разработки Python-фреймворка для автоматизации релизных процессов с интеграцией в систему непрерывной интеграции и доставки Jenkins. Актуальность работы обусловлена необходимостью вынесения сложной бизнес-логики релизных конвейеров за пределы конкретных CI/CD-оркестраторов и обеспечения ее единообразного исполнения как на локальных компьютерах разработчиков, так и в распределенных сборочных кластерах. В качестве технологической основы выбран язык Python в сочетании с библиотеками Click, Pydantic, Jinja2, Cookiecutter, Loguru и rich. Спроектирована многоуровневая архитектура с разграничением ответственности между слоем командного интерфейса, ядром и подсистемами пайплайнов, плагинов и логирования. Реализованы динамическая загрузка пользовательского кода посредством importlib, типизированные модели параметров и секретов, интеграция с HashiCorp Vault для управления секретами и передача данных наблюдаемости в OpenSearch. Корректность работы подтверждена комплексом из 214 автоматизированных тестов с покрытием исходного кода 80 %. Полученное программное решение пригодно для практического применения и последующего расширения за счет плагиновой архитектуры.

Ключевые слова: python, автоматизация, релизный процесс, ci/cd, jenkins, фреймворк, консольное приложение, click, pydantic, jinja2, cookiecutter, loguru, плагиновая архитектура, hashicorp vault, opensearch

V. V. Mikhaylyuk¹✉, E. Yu. Voronkin¹

Development of a python framework for release process automation with integration into the Jenkins CI/CD system

¹Siberian State University of Geosystems and Technologies, Novosibirsk, Russian Federation
e-mail: vitia.mikhailiuk@mail.ru

Abstract. The article discusses the process of developing a Python framework for release process automation with integration into the Jenkins continuous integration and delivery system. The relevance of the work is determined by the need to decouple complex business logic of release pipelines from specific CI/CD orchestrators and to ensure its uniform execution both on developers' local machines and in distributed build clusters. Python was chosen as the technological basis together with the Click, Pydantic, Jinja2, Cookiecutter, Loguru and rich libraries. A multi-level architecture has been designed with a clear separation of responsibilities between the command-line interface layer, the core, and the pipelines, plugins and logging subsystems. Dynamic loading of user code via importlib, typed models of parameters and secrets, integration with HashiCorp Vault for secret management, and transmission of observability data to OpenSearch have been implemented. Correctness is confirmed by a suite of 214 automated tests with 80 % source code coverage. The resulting software solution is suitable for practical application and further extension through its plugin architecture.

Keywords: python, automation, release process, ci/cd, jenkins, framework, command-line application, click, pydantic, jinja2, cookiecutter, loguru, plugin architecture, hashicorp vault, opensearch

Введение

Современный цикл выпуска программных продуктов неразрывно связан с практиками непрерывной интеграции и доставки и предполагает оркестрацию десятков внешних систем – трекеров задач, баз знаний, репозиторий артефактов, систем статического анализа и корпоративных мессенджеров [1]. На практике реализация таких межинтеграционных связей часто осуществляется стихийно посредством разрозненных сценариев на Bash, PowerShell или встроенных DSL-языках CI-серверов, что порождает высокую связность кода, отсутствие единых механизмов обработки ошибок, риски утечки конфиденциальных данных в журналы и невозможность локального запуска бизнес-логики для отладки.

В качестве альтернативы данному подходу рассматривается вынесение сложной бизнес-логики из конфигурации CI/CD в самостоятельные консольные приложения на языках общего назначения, в первую очередь на Python [2]. Такой подход позволяет применять к инфраструктурным сценариям классические практики программной инженерии: модульное тестирование [3], объектно-ориентированное проектирование и строгий статический анализ.

Целью настоящей работы является разработка Python-фреймворка для автоматизации релизных процессов с интеграцией в CI/CD-систему Jenkins, обеспечивающего независимое и безопасное выполнение интеграционных сценариев как на локальных компьютерах разработчиков, так и в распределенных сборочных кластерах.

Для достижения поставленной цели выполнены следующие задачи:

- провести анализ существующих методов и средств создания автоматизационных сценариев и сравнительное рассмотрение представленных на рынке решений;
- обосновать технологический стек реализации;
- проектировать функциональную структуру фреймворка и построить комплекс UML-моделей;
- реализовать ядро системы и подсистемы пайплайнов, плагинов и логирования, а также интерфейс командной строки;
- разработать комплекс автоматизированных тестов и провести количественную оценку покрытия исходного кода.

Методы и материалы

Перед началом программной реализации фреймворка спроектирована его функциональная структура и построен комплекс UML-моделей, охватывающих систему на структурном и поведенческом уровнях [4].

Диаграмма прецедентов формализовала функциональные требования к системе с позиции единственного актора – разработчика – и зафиксировала пять групп сценариев использования: управление проектом, взаимодействие с пай-

пайнами, работу с конфигурацией, расширение через плагины и контроль выполнения (рис. 1).

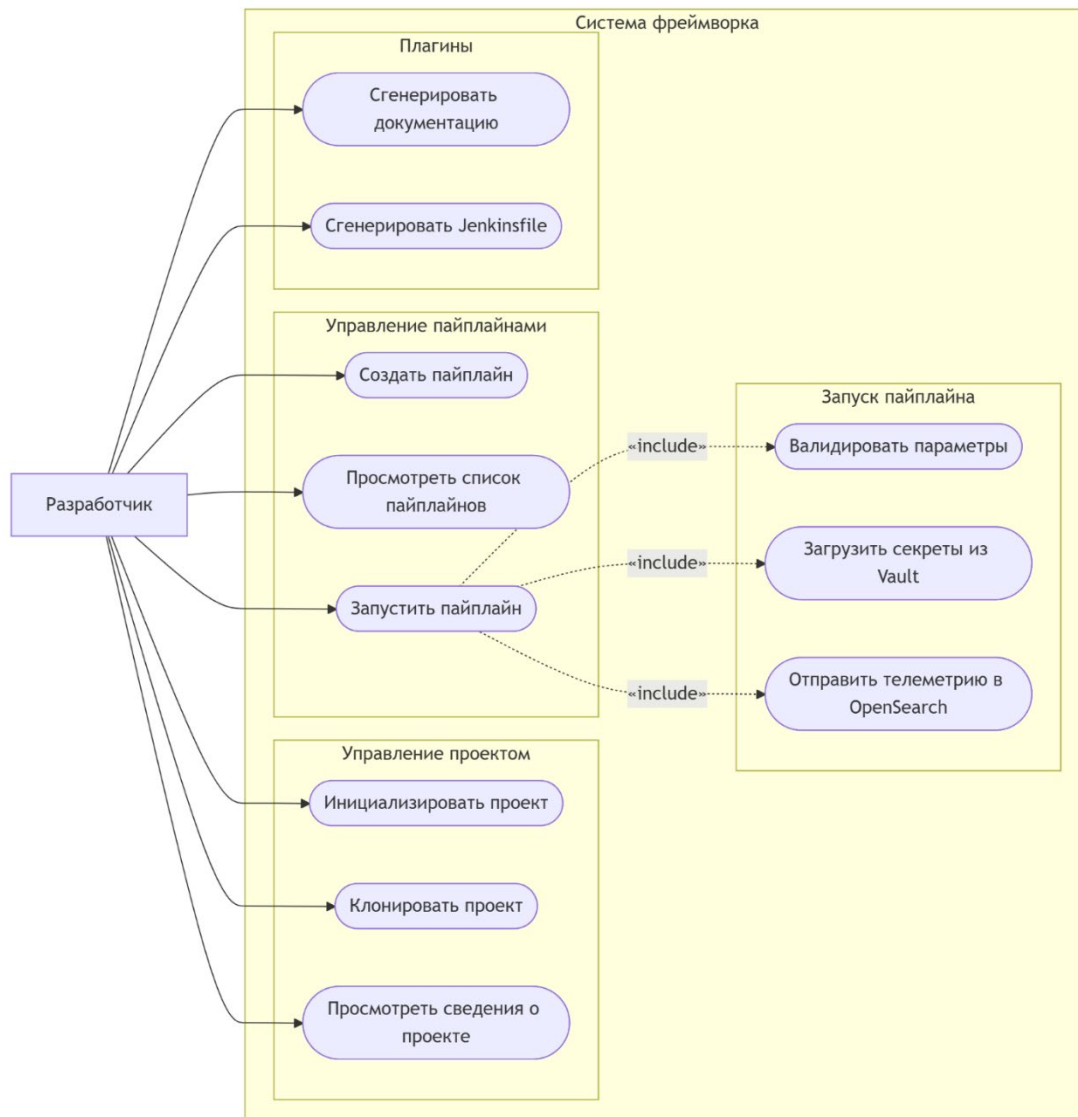


Рис. 1. Диаграмма прецедентов фреймворка

Диаграмма классов описала объектную модель ядра, сгруппировав ключевые классы по функциональному назначению – управление выполнением, передача данных и валидация – и зафиксировав отношения агрегации, композиции, наследования и зависимости (рис. 2).

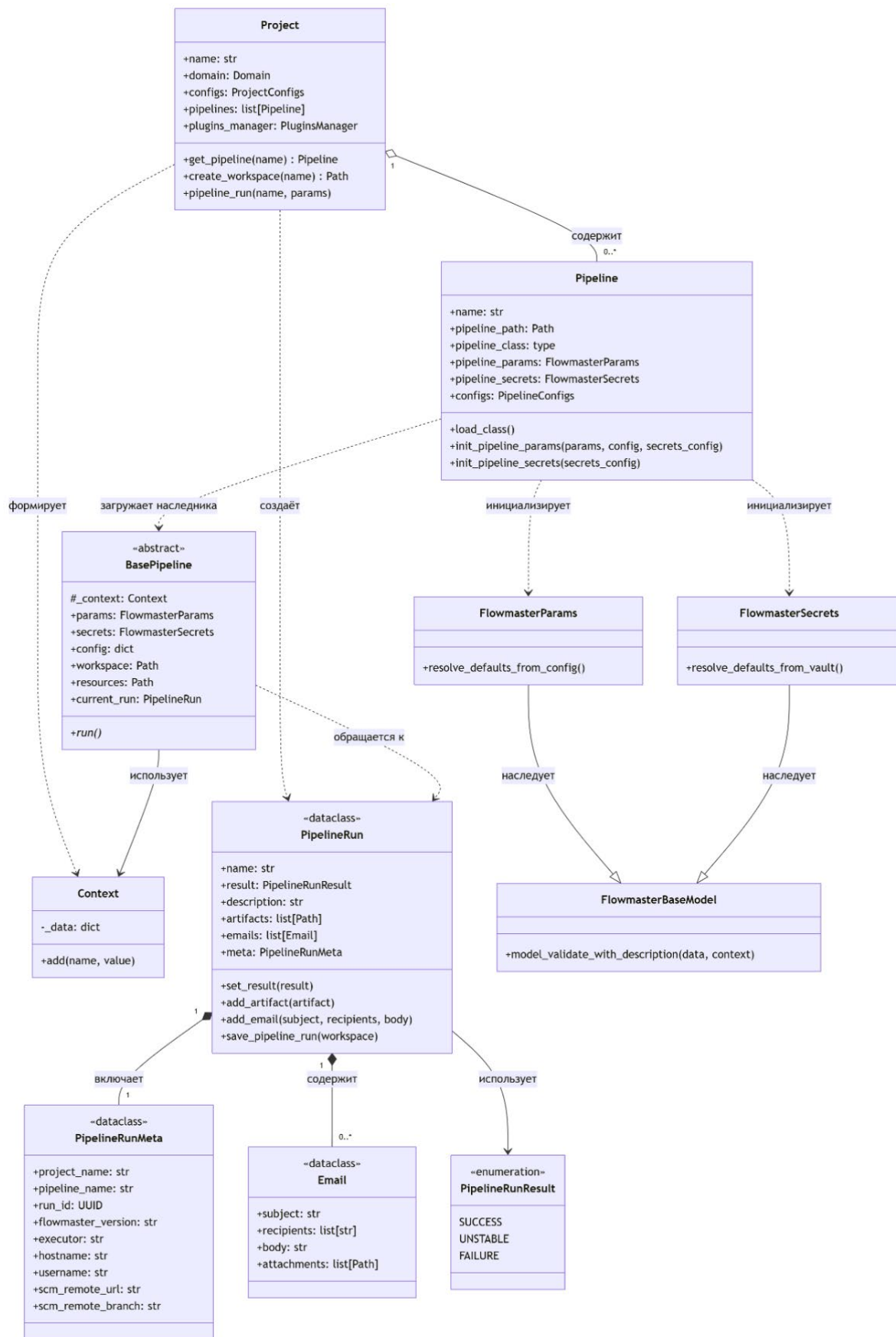


Рис. 2. Диаграмма классов фреймворка

Диаграмма последовательностей смоделировала временной порядок взаимодействия компонентов в ходе типового запуска пайплайна, отразив инициализацию проекта, подготовку запуска, динамическую загрузку пайплайна, валидацию параметров, разрешение секретов, формирование контекста исполнения и сохранение результатов (рис. 3).

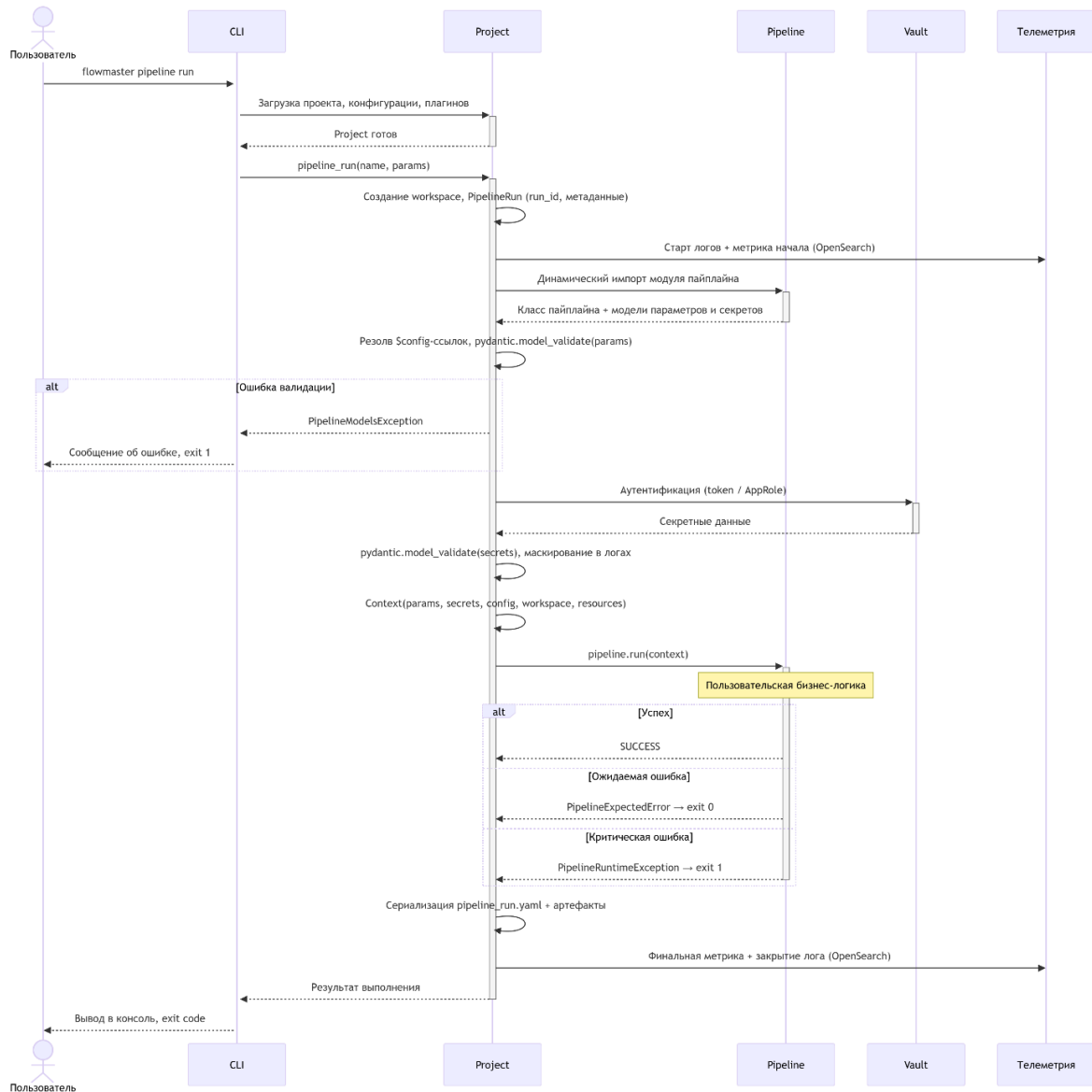


Рис. 3. Диаграмма последовательностей выполнения пайплайна

В качестве серверной части и точки входа выбран язык программирования Python с богатой объектно-ориентированной моделью и развитой системой типизации. Интерфейс командной строки реализован средствами библиотеки Click, обеспечивающей маршрутизацию команд через декораторы и автоматическую генерацию справочной документации. Валидация параметров пайплайнов и секретов выполняется на основе типизированных моделей библиотеки Pydantic [5]. Для генерации конфигурационных файлов и шаблонов документации применя-

ется шаблонизатор Jinja2; для разворачивания типовой структуры новых проектов и пайплайнов используется библиотека Cookiecutter. Подсистема журналирования построена на основе библиотеки Loguru с поддержкой потокобезопасной записи через очередь, форматированный табличный вывод в терминал реализован средствами библиотеки rich.

Хранение и доступ к секретам осуществляются через HashiCorp Vault [6], что обеспечивает централизованное управление учетными данными и их изоляцию от исходного кода пайплайнов. Журналы и метрики выполнения направляются в систему централизованной телеметрии на базе OpenSearch [7] и используются для последующего анализа и построения дашбордов.

Результаты

В соответствии со спроектированной архитектурой реализованы все предусмотренные подсистемы фреймворка.

Ядро системы оформлено в пакете flowmaster.core. Класс Project выступает единой точкой оркестрации и инкапсулирует жизненный цикл запуска пайплайна. Подсистема конфигурации реализует доменно-специфичную загрузку параметров с разделением на базовый и доменный уровни, а иерархия исключений обеспечивает дифференцированную обработку ошибок различных категорий. Вспомогательные утилиты – рекурсивное слияние словарей, безопасный доступ к вложенным ключам и контекстный менеджер смены рабочей директории – формируют переиспользуемые примитивы для всех последующих подсистем.

Подсистема пайплайнов построена вокруг базового класса BasePipeline и метаклассового механизма копирования атрибутов вложенного класса Meta. Динамическая загрузка пользовательского кода выполняется через стандартный модуль importlib [8], что позволяет отделить пользовательский код от ядра фреймворка. Параметры и секреты пайплайна описываются типизированными моделями FlowmasterParams и FlowmasterSecrets с автоматическим разрешением ссылок на значения из конфигурации и из HashiCorp Vault. Класс Context формирует окружение исполнения, а классы PipelineRun и PipelineRunMeta обеспечивают трассируемое сохранение результатов запуска в файле pipeline_run.yaml.

Подсистема логирования реализована в пакете flowmaster.logging на базе библиотеки Loguru. Глобальный фильтр маскирования конфиденциальных данных реализован с применением паттерна Singleton [9] и гарантирует обработку секретов во всех выходных потоках независимо от точки журналирования. Передача журналов во внешнюю систему агрегации выполняется через очередь и фоновый поток, что выносит сетевое взаимодействие за пределы основного потока исполнения и сохраняет работоспособность пайплайна при временной недоступности OpenSearch.

Подсистема плагинов основана на разделении контракта Plugin/CliPlugin и менеджера PluginsManager, выполняющего динамическую загрузку классов плагинов по путям, указанным в конфигурации. Такое решение реализует принцип открытости/закрытости: добавление новой интеграции сводится к созданию класса-наследника CliPlugin и регистрации его в конфигурации без изменений в

ядре фреймворка. На основе данного контракта разработаны два штатных плагина – генератор документации пайплайнов средствами Jinja2 и интеграционный плагин Jenkins, обеспечивающий формирование Jenkinsfile и общей разделяемой библиотеки.

Слой командного интерфейса реализован в пакете flowmaster.cli на основе библиотеки Click и объединяет все точки взаимодействия пользователя с фреймворком. Группа команд project отвечает за управление каркасом автоматизационного проекта, группа pipeline – за работу со списком пайплайнов, их инициализацию из шаблона и запуск с передачей параметров в формате JSON. Декоратор @pass_obj_with_project выполняет ленивую инициализацию объекта Project и пробрасывает его в обработчики команд. Команды плагинов встраиваются в общее дерево CLI наряду со штатными группами, что обеспечивает единообразный пользовательский опыт. Пример интерфейса представлен на рисунках 5, 6, 7, 8.

```
(flowmaster-py3.12)
vitia@viktor-pc MINGW64 /e/Development
• $ mkdir test-project
(flowmaster-py3.12)
vitia@viktor-pc MINGW64 /e/Development
• $ cd test-project
(flowmaster-py3.12)
vitia@viktor-pc MINGW64 /e/Development/test-project
• $ flowmaster project init -n test-project
2026-04-30 21:56:05 | INFO | Генерация нового проекта test-project в E:\Development\test-project.
2026-04-30 21:56:05 | SUCCESS | ✔ Проект test-project успешно инициализирован.
(flowmaster-py3.12)
vitia@viktor-pc MINGW64 /e/Development/test-project
• $ flowmaster project current
2026-04-30 21:56:13 | INFO | Информация о проекте test-project
```

Параметр	Значение
name	test-project
description	Описание проекта test-project
created_at	30-04-2026
default_domain	test
plugins	flowmaster.plugins.jenkins, flowmaster.plugins.docs

```
(flowmaster-py3.12)
vitia@viktor-pc MINGW64 /e/Development/test-project
○ $
```

Рис. 5. Пример работы команд интерфейса фреймворка

```
(flowmaster-py3.12)
vitia@viktor-pc MINGW64 /e/Development/test-project
• $ flowmaster pipeline init -n test_pipeline
2026-04-30 22:00:36 | SUCCESS | ✔ Пайплайн test_pipeline успешно инициализирован.
(flowmaster-py3.12)
vitia@viktor-pc MINGW64 /e/Development/test-project
• $ flowmaster pipeline run -n test_pipeline
2026-04-30 22:00:42 | INFO | Инициализация параметров пайплайна 'test_pipeline'.
2026-04-30 22:00:42 | SUCCESS | ✔ Инициализация параметров пайплайна 'test_pipeline' завершена успешно.
2026-04-30 22:00:42 | INFO | Инициализация секретов пайплайна 'test_pipeline'.
2026-04-30 22:00:42 | SUCCESS | ✔ Инициализация секретов пайплайна 'test_pipeline' завершена успешно.
2026-04-30 22:00:42 | INFO | Запуск пайплайна 'test_pipeline'.
2026-04-30 22:00:42 | INFO | Тестовый запуск.
2026-04-30 22:00:42 | INFO | Завершена работа пайплайна 'test_pipeline'.
(flowmaster-py3.12)
vitia@viktor-pc MINGW64 /e/Development/test-project
○ $
```

Рис. 6. Пример работы команд интерфейса фреймворка

```

vitia@viktor-pc MINGW64 /e/Development/test-project
● $ flowmaster pipeline run -n test_pipeline
2026-05-01 00:07:23 | INFO | Инициализация параметров пайплайна 'test_pipeline'.
2026-05-01 00:07:23 | SUCCESS | ✓ Инициализация параметров пайплайна 'test_pipeline' завершена успешно.
2026-05-01 00:07:23 | INFO | Инициализация секретов пайплайна 'test_pipeline'.
2026-05-01 00:07:23 | SUCCESS | ✓ Инициализация секретов пайплайна 'test_pipeline' завершена успешно.
2026-05-01 00:07:23 | INFO | Запуск пайплайна 'test_pipeline'.
2026-05-01 00:07:23 | INFO | Pydantic-модель с тестовым секретом из HashiCorp Vault: test_key='***HIDDEN***'
2026-05-01 00:07:23 | INFO | Завершена работа пайплайна 'test_pipeline'.

```

Рис. 7. Пример работы команд интерфейса фреймворка

```

vitia@viktor-pc MINGW64 /e/Development/test-project (master)
● $ flowmaster jenkins init
2026-05-01 01:14:23 | INFO | Генерация flowmaster-shared-library
2026-05-01 01:14:23 | SUCCESS | ✓ Генерация flowmaster-shared-library завершена успешно!
(flowmaster-py3.12)
vitia@viktor-pc MINGW64 /e/Development/test-project (master)
● $ flowmaster jenkins generate pipeline -n test_pipeline -j test_instance
2026-05-01 01:14:25 | SUCCESS | ✓ Jenkinsfile для пайплайна test_pipeline успешно обновлен.

```

Рис. 8. Пример работы команд интерфейса фреймворка

Корректность реализации подтверждена разработанным комплексом автоматизированных тестов на базе фреймворка `pytest`. Набор включает 214 тестов, распределенных по двадцати одному модулю, и сочетает модульное тестирование изолированных единиц поведения с функциональными end-to-end сценариями, выполняющими реальные CLI-команды в изолированной временной директории. Изоляция тестового окружения достигается стандартными фикстурами `pytest` и подменой внешних зависимостей средствами модуля `unittest.mock`. Автоматизированный прогон завершается без отказов; измеренное средствами `coverage.py` покрытие исходного кода составляет 80 % от 1621 исполняемой инструкции фреймворка, при этом подсистемы ядра, моделей параметров и секретов, логирования и плагиновых контрактов покрыты полностью.

Заключение

В статье рассмотрен процесс разработки Python-фреймворка для автоматизации релизных процессов с интеграцией в CI/CD-систему Jenkins. На основании проведенного анализа существующих подходов обоснована необходимость вынесения бизнес-логики релизных конвейеров за пределы конкретных оркестраторов в самостоятельное программное решение. Спроектирована многоуровневая архитектура с разграничением ответственности между слоем командного интерфейса, ядром и подсистемами пайплайнов, плагинов и логирования; согласованность принятых решений зафиксирована комплексом UML-моделей.

Реализация фреймворка обеспечила динамическую загрузку пользовательского кода, типизированную валидацию параметров и секретов, централизованное управление секретами через HashiCorp Vault, асинхронную передачу журналов в OpenSearch и расширяемость через плагиновый контракт. Корректность работы подтверждена прогоном 214 автоматизированных тестов без отказов и достигнутым покрытием исходного кода в 80 %.

Полученное программное решение пригодно к практическому применению в качестве унифицированного инструмента описания и исполнения многокомпонентных автоматизационных сценариев, обеспечивающего единообразный запуск как в локальном окружении разработчика, так и в кластере Jenkins. Перспективы дальнейшего развития связаны с расширением набора штатных плагинов для распространенных CI/CD-платформ и трекеров задач, реализацией параллельного исполнения нескольких пайплайнов, подключением альтернативных систем управления секретами и телеметрии, а также с развитием инструментальной поддержки разработки пользовательских пайплайнов.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Хамбл Д., Фарли Д. Непрерывное развертывание ПО: автоматизация процессов сборки, тестирования и внедрения новых версий программ / Пер. с англ. А. Г. Сысолюка. – М.: Вильямс, 2011. – 432 с. – ISBN 978-5-8459-1739-3.
2. Гифт Н., Берман К., Деза А., Георгиу Г. Python и DevOps: ключ к автоматизации Linux / Пер. с англ. – СПб.: Питер, 2022. – ISBN 978-5-4461-2929-4.
3. Майерс Г., Баджетт Т., Сандлер К. Искусство тестирования программ. 3-е изд. / Пер. с англ. – М.: Вильямс, 2012. – 272 с. – ISBN 978-5-8459-1796-6.
4. Фаулер М. UML. Основы: краткое руководство по стандартному языку объектного моделирования. 3-е изд. / Пер. с англ. – СПб.: Символ-Плюс, 2005. – 192 с. – ISBN 978-5-93286-060-1.
5. Pydantic. Официальная документация. – URL: <https://docs.pydantic.dev/> (дата обращения: 08.04.2026).
6. HashiCorp Vault. Официальная документация. – URL: <https://developer.hashicorp.com/vault/docs> (дата обращения: 16.04.2026).
7. OpenSearch. Официальная документация. – URL: <https://opensearch.org/docs/latest> (дата обращения: 16.04.2026).
8. Python Software Foundation. Документация стандартной библиотеки Python 3: модуль importlib. – URL: <https://docs.python.org/3/library/importlib.html> (дата обращения: 17.04.2026).
9. Гамма Э., Хелм Р., Джонсон Р., Влиссидес Д. Паттерны объектно-ориентированного проектирования / Пер. с англ. – СПб.: Питер, 2021. – 448 с. – ISBN 978-5-4461-1595-2.

© В. В. Михайлюк, Е. Ю. Воронкин, 2026

М. А. Мищенко^{1✉}, Н. С. Головачев¹

Разработка симулятора настройки сборочного конвейера для обучения программированию и автоматизации на виртуальной модели производственного участка

¹Сибирский государственный университет геосистем и технологий,
г. Новосибирск, Российская Федерация
e-mail: mmisha2702@gmail.com

Аннотация. В работе представлены результаты разработки учебного приложения «Автоматика», предназначенного для начального освоения программирования производственных манипуляторов на виртуальной модели производственного участка. Актуальность темы обусловлена высокой стоимостью реального робототехнического оборудования, требованиями к безопасности и ограниченной доступностью практических занятий для школьников. Цель работы состоит в создании доступного симулятора, позволяющего в безопасной среде изучать базовые операции автоматизации и логику управления манипулятором. В ходе исследования выполнен анализ существующих образовательных и инженерных симуляторов, сформулированы требования к пользовательскому сценарию, разработана структура уровней и реализована программная система на Unity с поддержкой пользовательских Lua-скриптов. В результате создан прототип, включающий пять учебных уровней, консоль программирования, встроенную справку, журнал сообщений и задания с конвейером. Разработанное решение может использоваться для профориентации и вводного обучения школьников.

Ключевые слова: учебный симулятор, промышленный манипулятор, программирование, автоматизация, Unity, Lua

M. A. Mishchenko^{1✉}, N. S. Golovachev¹

Development of an assembly line setup simulator for programming and automation training on a virtual model of a production site

¹Siberian State University of Geosystems and Technologies, Novosibirsk, Russian Federation
e-mail: mmisha2702@gmail.com

Abstract. The paper presents the results of developing the Automatica training application intended for introductory learning of industrial manipulator programming on a virtual model of a production area. The relevance of the study is determined by the high cost of real robotic equipment, safety requirements, and limited access to hands-on practice for school students. The aim was to create an accessible simulator that allows learners to study basic automation operations and manipulator control logic in a safe digital environment. The study included an analysis of existing educational and engineering simulators, formulation of user-scenario requirements, design of level structure, and implementation of a software system in Unity with support for user Lua scripts. The resulting prototype includes five training levels, a programming console, built-in reference materials, a message log, and conveyor-based tasks. The developed solution can be used for career guidance and introductory training in industrial automation.

Keywords: training simulator, industrial manipulator, programming, automation, Unity, Lua

Введение

Автоматизация производственных процессов и использование промышленных манипуляторов становятся базовой частью современной промышленности. Вместе с этим растет потребность в раннем знакомстве обучающихся с принципами программирования оборудования и логикой автоматизированных операций. Исследования по инженерному образованию показывают, что цифровые симуляторы позволяют организовать безопасную, повторяемую и сравнительно недорогую практику, особенно в тех случаях, когда работа с реальным оборудованием требует значительных затрат или связана с риском [1–3].

Для школьного обучения проблема особенно заметна. Реальные манипуляторы и учебные роботизированные ячейки дороги в закупке, требуют обслуживания и квалифицированного сопровождения, а ошибки начинающих пользователей способны привести к поломкам или опасным ситуациям. При этом современные виртуальные и смешанные среды обучения подтверждают, что даже сложные понятия кинематики, планирования траекторий и взаимодействия с оборудованием могут быть поданы в более наглядной форме [4, 5].

Целью работы является разработка симулятора настройки сборочного конвейера для обучения школьников базовым навыкам программирования производственного манипулятора. Для достижения цели были поставлены следующие задачи: проанализировать существующие аналоги; сформулировать требования к учебному сценарию; спроектировать структуру уровней и пользовательский интерфейс; реализовать программное управление манипулятором на основе текстовых скриптов; выполнить предварительное тестирование прототипа.

Исследование предметной области

Для определения функциональных требований к приложению были рассмотрены как образовательные, так и профессиональные решения в области робототехнического моделирования. К числу образовательных продуктов относятся Promobot M Edu, RoboSim и RoboMind [6–8]. Они ориентированы на обучение основам программирования и робототехники, однако используют различный подход к вовлечению пользователя: от практики на реальном оборудовании до работы в полностью виртуальной среде. Профессиональные платформы Gazebo и CoppeliaSim [9, 10] обеспечивают более широкий набор средств для моделирования, но рассчитаны прежде всего на инженерную и исследовательскую деятельность.

Наиболее близкими к тематике разрабатываемого приложения являются Promobot M Edu и RoboSim. Первый продукт ценен высокой связью с реальным оборудованием и позволяет формировать представление о физических ограничениях промышленного робота. Второй обеспечивает более доступный вход в обучение, так как не требует наличия роботизированной установки и дает возможность быстро переходить к выполнению учебных сценариев на обычном персональном компьютере. RoboMind, в свою очередь, полезен как пример удачной подачи алгоритмического мышления для начинающих пользователей, но слабее

связан именно с тематикой промышленной автоматизации и работы манипулятора на производственном участке.

С методической точки зрения образовательные аналоги демонстрируют несколько сильных сторон, которые представляют интерес для дальнейшего развития «Автоматики». У Promobot M Edu важным преимуществом является реалистичность учебного контекста, однако зависимость от физического оборудования ограничивает массовое применение такого решения в школьной среде. RoboSim выигрывает за счет доступности, понятного входа и ориентации на учебный процесс, но в большей степени охватывает робототехнику в целом, а не узкий сценарий программирования производственного манипулятора. RoboMind удачно использует постепенное усложнение и низкий порог входа, что особенно важно для школьников; именно эти качества целесообразно усиливать и в рассматриваемом проекте через короткие задания, шаблоны решений и быструю обратную связь по ошибкам.

Профессиональные симуляторы Gazebo и CoppeliaSim показывают другой полюс развития подобных систем. Их сильными сторонами являются поддержка сложной физики, гибкая работа с моделями роботов, возможность расширения под исследовательские задачи и сценарии цифровых двойников. Вместе с тем для целевой аудитории школьников такие платформы оказываются слишком сложными из-за насыщенного интерфейса, большого числа настроек и необходимости знания дополнительных инструментов. Тем не менее отдельные их достоинства могут быть интегрированы в «Автоматику» на следующем этапе: визуализация траектории манипулятора, более подробный анализ ошибок, редактор сценариев и автоматическая проверка решений.

На основе сравнительного анализа были сформулированы ключевые требования к проекту: низкий порог входа, ограниченный и понятный набор команд, ориентация на один тип манипулятора, постепенное усложнение задач, работа на персональном компьютере без специализированного оборудования и наличие встроенной справки. Таким образом, концепция приложения определяется не стремлением заменить профессиональный инженерный симулятор, а задачей создать доступную учебную среду для раннего погружения в промышленную автоматизацию.

Материалы и методы

Программная реализация выполнена в среде Unity 6 [11]. Для построения интерфейса использованы стандартные средства UGUI и пакет TextMeshPro [12]. В качестве языка пользовательских сценариев выбран Lua, хорошо подходящий для компактных учебных скриптов [13]. Выполнение скриптов внутри приложения обеспечивается интерпретатором MoonSharp [14]. Для анимации интерфейсных элементов применен DOTween [15], а для части асинхронных операций пользовательского интерфейса и сценической логики - UniTask [16].

Приложение построено по сценовой модели и включает главное меню и пять учебных уровней. Общая структура проекта продемонстрирована на рис.1. Каждая сцена содержит виртуальный производственный участок с манипулято-

ром, набором объектов, зонами контроля и условиями завершения задания. Для объектов предусмотрены свойства типа и качества, что позволяет переходить от простых задач переноса к заданиям по отбору деталей. Управление манипулятором выполняется через консоль, в которой пользователь вводит текст программы и запускает ее выполнение.

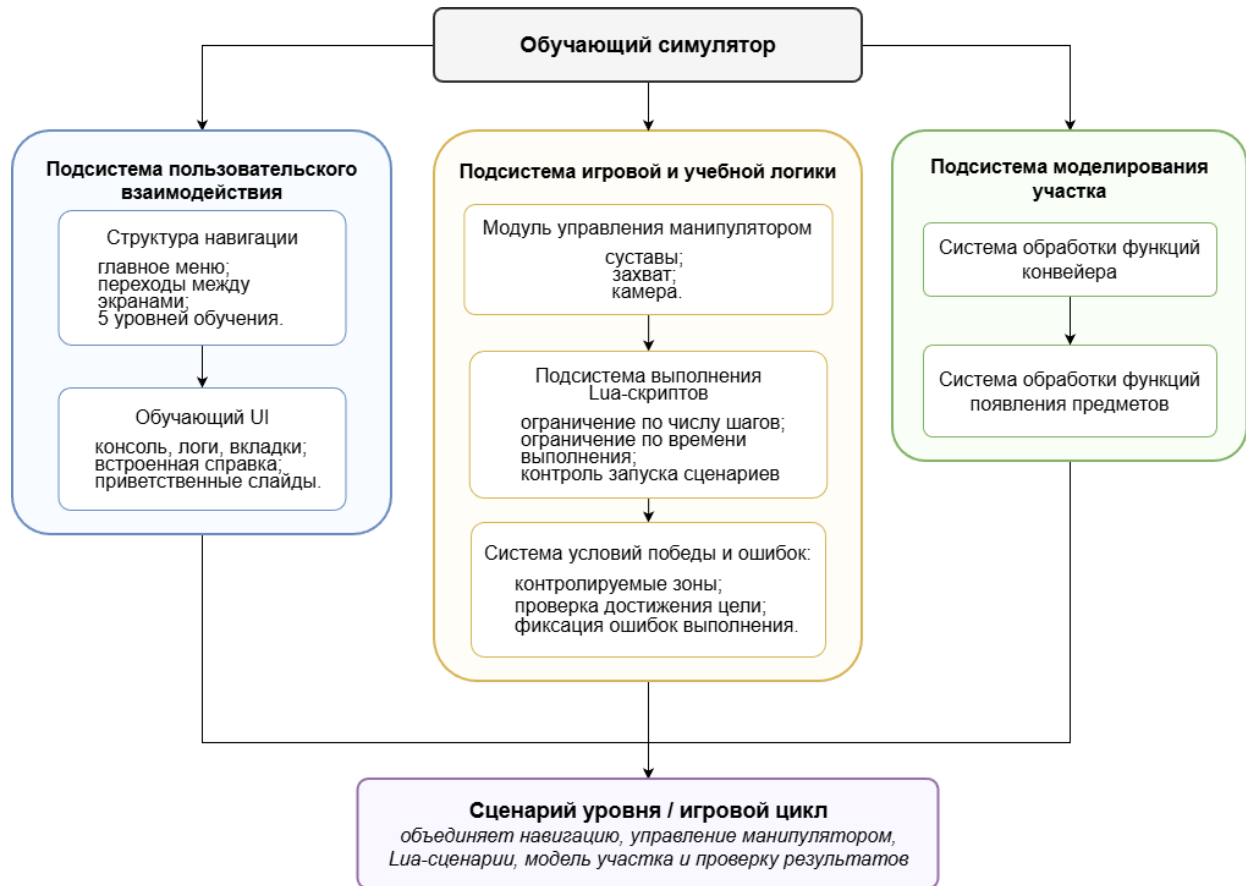


Рис. 1. Структура системы проекта

Учебный сценарий был спроектирован вокруг набора базовых команд: поворот суставов на заданный угол, захват и отпускание объекта, проверка состояния захвата, а также задание белых и черных списков по типам и качеству деталей. Такой подход позволяет сосредоточить внимание не на сложной математической модели, а на логике действий, последовательности операций и обработке ограничений. Отдельное внимание уделено встроенной справке, журналу сообщений и визуальному контролю выполнения задания, поскольку эти элементы напрямую влияют на понятность обучения для начинающего пользователя.

Проверка прототипа проводилась методом ручного тестирования на каждой сцене с контролем прохождения уровней, устойчивости выполнения скриптов и корректности интерфейсных элементов. Оценивались успешность выполнения учебных сценариев, корректность обработки ошибочных команд, удобство работы с консолью и справочными окнами, а также целостность визуального интерфейса на разных уровнях.

Результаты

В результате работы создан учебный прототип «Автоматика», ориентированный на школьников, делающих первые шаги в сфере промышленной автоматизации. Обучение построено по принципу поэтапного усложнения. На первом уровне пользователь переносит один куб с одного ящика на другой. На втором уровне количество объектов увеличивается до двух. На третьем уровне добавляются различие типов и требование соблюдать заданный порядок укладки. Четвертый уровень вводит в сценарий конвейер, а пятый - отбор объектов по качественным признакам.

Важным результатом стала реализация текстовой консоли программирования, объединенной со встроенной справкой и журналом сообщений. Пользователь может выбирать манипулятор, вводить скрипт, запускать его и получать диагностические сообщения при ошибках. В справочном окне представлены доступные команды, ограничения и примеры применения. Такой формат сближает приложение с реальной практикой программирования оборудования, но при этом сохраняет учебную понятность интерфейса.

Кроме консоли, в приложении реализованы вспомогательные механики, повышающие удобство обучения: переключение между несколькими камерами, панель задания, перезагрузка уровня и визуальный контроль состояния сцены. На поздних уровнях пользователь может применять фильтрацию деталей с помощью белых и черных списков по типам и качеству. Это расширяет учебный сценарий от простого перемещения объектов до принятия логических решений внутри программы.

Интерфейс симулятора показан на рис. 2. На изображении видны виртуальный производственный участок, окно написания программы и встроенная справка по доступным командам. Такое совмещение пространства сцены и инструментов программирования позволяет учащемуся сразу соотносить написанный код с поведением манипулятора и результатом на конвейере.

По итогам предварительного тестирования были выявлены и устранены отдельные недочеты интерфейса и сценарной логики. В частности, был увеличен допустимый интервал работы скрипта, устранены ошибки прокрутки текстовых областей и переработано оформление элементов управления. Исправления позволили повысить устойчивость прототипа и целостность восприятия интерфейса при прохождении уровней.

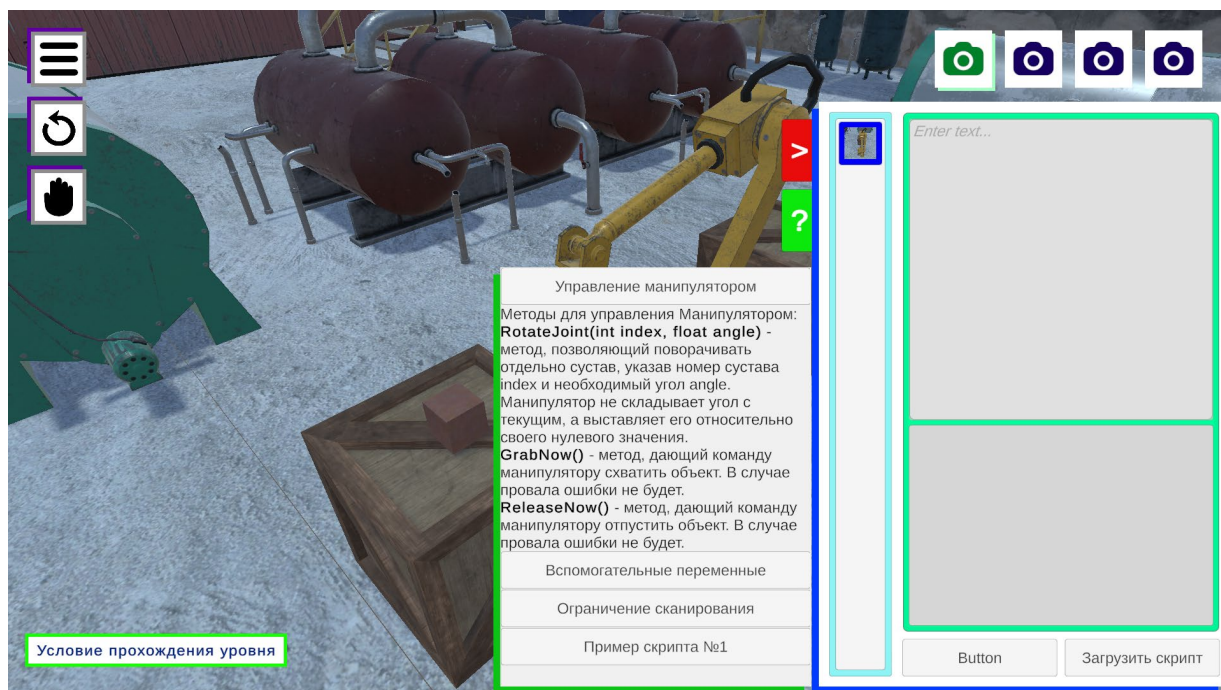


Рис. 2. Интерфейс симулятора «Автоматика»

Обсуждение

Полученный результат занимает промежуточное положение между простыми средами для начального знакомства с программированием и сложными профессиональными симуляторами. В отличие от решений, ориентированных на работу с реальным учебным оборудованием, разработанный прототип не требует закупки роботизированной установки и снижает риск повреждения техники. По сравнению с профессиональными платформами Gazebo и CoppeliaSim проект сознательно ограничивает функциональность и уровень физической детализации, поскольку его основная задача заключается не в точном моделировании производственной ячейки, а в раннем погружении в логику программирования манипулятора.

Одновременно проект имеет и ограничения. На текущем этапе не реализованы расширенные средства анализа результатов, редактор пользовательских уровней, автоматическая проверка решений и высокоточная кинематическая или динамическая модель оборудования. Перспективными направлениями развития являются расширение библиотеки заданий, добавление сценариев с несколькими манипуляторами, внедрение системы оценки действий учащегося и интеграция с более подробной моделью производственного процесса.

Заключение

Разработанный учебный симулятор позволяет в безопасной и доступной форме познакомить школьников с базовыми принципами программирования производственных манипуляторов и автоматизации операций на конвейере. Реализованный прототип содержит последовательный учебный сценарий, средства написания и отладки скриптов, а также набор заданий, постепенно усложняю-

ших логику взаимодействия с оборудованием. Практическая значимость работы заключается в возможности использования приложения для профориентации, демонстрационных занятий и вводного обучения без необходимости применения дорогостоящего реального оборудования.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Negahban, A. Simulation in engineering education: The transition from physical experimentation to digital immersive simulated environments // *Simulation*. 2024. Vol. 100, No. 7. DOI: 10.1177/00375497241229757.
2. Strojny, P., Duzmanska-Misiarczyk, N. Measuring the effectiveness of virtual training: A systematic review // *Computers & Education: X Reality*. 2023. Vol. 2. Art. 100006. DOI: 10.1016/j.cexr.2022.100006.
3. Shamsuzzoha, A., Toshev, R., Vu Tuan, V., Kankaanpaa, T., Helo, P. Digital factory-virtual reality environments for industrial training and maintenance // *Interactive Learning Environments*. 2021. Vol. 29, No. 8. P. 1339-1362. DOI: 10.1080/10494820.2019.1628072.
4. Nathanael, D. et al. Development and evaluation of a virtual training environment for on-line robot programming // *International Journal of Industrial Ergonomics*. 2016. Vol. 53. P. 274-283. DOI: 10.1016/j.ergon.2016.02.004.
5. Tarng, W., Wu, Y.-J., Ye, L.-Y. et al. Application of Virtual Reality in Developing the Digital Twin for an Integrated Robot Learning System // *Electronics*. 2024. Vol. 13, No. 14. Art. 2848. DOI: 10.3390/electronics13142848.
6. Promobot M Edu. URL: <https://promo-bot.ai/promobot-m-edu/> (дата обращения: 13.05.2026).
7. Робосим. URL: <https://robosim.io/> (дата обращения: 13.05.2026).
8. RoboMind. URL: <https://www.robomind.net/> (дата обращения: 13.05.2026).
9. Gazebo. Getting Started. URL: <https://gazebo.org/docs/latest/getstarted/> (дата обращения: 13.05.2026).
10. CoppeliaSim. URL: <https://www.coppeliarobotics.com/> (дата обращения: 13.05.2026).
11. Unity User Manual 6000.0. URL: <https://docs.unity3d.com/6000.0/Documentation/Manual/> (дата обращения: 13.05.2026).
12. TextMeshPro package documentation. URL: <https://docs.unity3d.com/Packages/com.unity.textmeshpro@3.2/manual/index.html> (дата обращения: 13.05.2026).
13. Lua 5.4 Reference Manual. URL: <https://www.lua.org/manual/5.4/> (дата обращения: 13.05.2026).
14. MoonSharp - A Lua interpreter written entirely in C#. URL: <https://www.moonsharp.org/> (дата обращения: 13.05.2026).
15. Demigiant. DOTween. URL: <https://github.com/Demigiant/dotween> (дата обращения: 13.05.2026).
16. Cysharp. UniTask. URL: <https://github.com/Cysharp/UniTask> (дата обращения: 13.05.2026).

© М. А. Мищенко, Н. С. Головачев, 2026

Д. Д. Модестова¹✉, Е. Ю. Воронкин¹

Модернизация веб-ресурса для учебно-научного центра «Планетарий» СГУГиТ

¹Сибирский государственный университет геосистем и технологий,
г. Новосибирск, Российская Федерация
e-mail: modes08@mail.ru

Аннотация. В статье рассматривается процесс модернизации веб-ресурса учебно-научного центра «Планетарий» СГУГиТ. Актуальность работы обусловлена необходимостью обеспечить актуальное информирование посетителей о мероприятиях и услугах центра, а также организовать удобное администрирование контента. Проектирование интерфейса было выполнено в Figma, вёрстка проходила в Visual Studio Code, в качестве технологической основы выбран WordPress. Реализованы разделы афиши и событий, структурированное представление информации о центре, галерея, а также сервисные сценарии, такие как привязанная к записям афиш модальная форма «Купить билет» и форма обратной связи на странице «Контакты», которые отправляют данные на электронную почту планетария. Особое внимание уделено интеграции пользовательского интерфейса с системой управления содержимым, тестированию ключевых сценариев и повышению качества представления ресурса в сети. Модернизированный веб-ресурс развёрнут и готов к эксплуатации; работа демонстрирует практическое применение современных CMS-технологий в образовательной и культурно-просветительской среде вуза.

Ключевые слова: веб-программирование, модернизация веб-ресурсов, CMS, PHP, HTML, CSS, Javascript, Figma, Visual Studio Code, учебно-научный центр, планетарий

D. D. Modestova¹✉, E. Yu. Voronkin¹

Modernization of the web resource for the educational and scientific center "Planetarium" of SSUGT

¹Siberian State University of Geosystems and Technologies, Novosibirsk, Russian Federation
e-mail: modes08@mail.ru

Abstract. The article considers the process of modernizing the web resource of the Educational and Scientific Center "Planetarium" of SSUGT. The relevance of the work is due to the need to provide up-to-date information to visitors about the center's events and services, as well as to organize convenient content administration. The interface was designed in Figma, the layout was carried out in Visual Studio Code, and WordPress was chosen as the technological basis. The sections of the poster and events, structured presentation of information about the center, a gallery, as well as service scenarios such as a modal form "Buy a ticket" linked to the poster records and a feedback form on the "Contacts" page, which send data to the planetarium's email, have been implemented. Special attention is paid to the integration of the user interface with the content management system, testing of key scenarios, and improving the quality of the resource's presentation on the web. The modernized web resource is deployed and ready for operation; the work demonstrates the practical application of modern CMS technologies in the educational and cultural-educational environment of the university.

Keywords: web programming, web resource modernization, CMS, PHP, HTML, CSS, Javascript, Figma, Visual Studio Code, educational and scientific center, planetarium

Введение

Современные образовательные учреждения, в том числе технические вузы, нуждаются в эффективных цифровых инструментах для информирования общественности и сопровождения учебно-научной и культурно-просветительской деятельности. Учебно-научные центры выполняют важную роль в популяризации знаний, организации мероприятий и взаимодействии с посетителями. Учебно-научный центр «Планетарий» СГУГиТ не является исключением: для него веб-ресурс служит основным каналом распространения информации об афише, событиях, правилах посещения и контактах, а также средством приёма обращений и заявок от посетителей. Однако на практике существующее представление центра в сети не всегда обеспечивает актуальность контента, удобное администрирование и полноценную реализацию сервисных функций.

На данный момент основная информация о деятельности планетария размещается на сайте, созданном на платформе Tilda. Такой подход удобен для быстрого запуска витринного ресурса, но ограничивает гибкость сопровождения: контент обновляется нерегулярно, структура сайта не ориентирована на системное ведение афиши и событий, отсутствуют целевые механизмы заявки на участие в мероприятии с привязкой к конкретной записи и полноценная форма обратной связи. В связи с этим возникла необходимость в модернизации веб-ресурса с переходом к управляемой платформе, ориентированной на нужды центра и его администраторов.

Целью настоящей работы является модернизация веб-ресурса учебно-научного центра «Планетарий» СГУГиТ с использованием системы управления содержимым WordPress. Для достижения цели были выполнены следующие задачи:

- анализ текущего состояния веб-ресурса и требований центра «Планетарий» к информации и сервисным функциям;
- проектирование структуры сайта и технической концепции;
- реализация функционального прототипа веб-ресурса;
- интегрирование пользовательского интерфейса с системой управления содержимым в административной части;
- проведение тестирования и оценка достижения цели.

Методы и материалы

Перед началом модернизации веб-ресурса были выполнены анализ предметной области [1], сравнение с существующим аналогом на платформе Tilda и формирование технического задания. На этапе проектирования созданы основные диаграммы, позволяющие определить требуемые возможности модернизируемого веб-ресурса и взаимодействие его частей.

Для определения информационной структуры ресурса и состава разделов построена схема структуры сайта [2] с иерархией подразделов и типов контента (рис. 1).

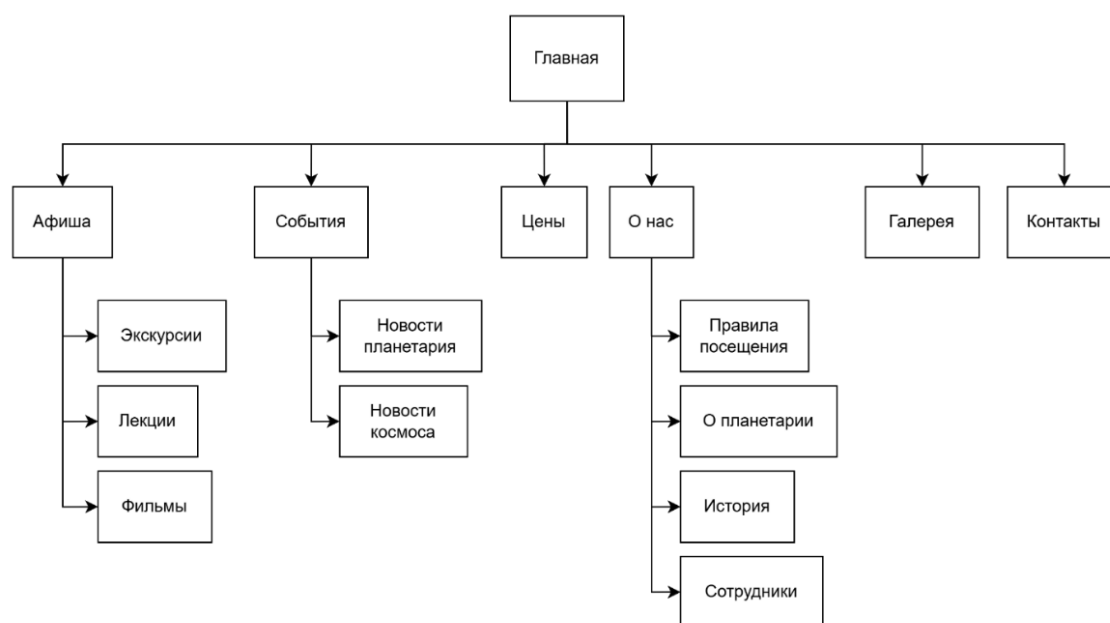


Рис. 1. Схема структуры веб-ресурса

Диаграмма прецедентов [3] построена для описания сценариев взаимодействия посетителя и администратора с системой, а также функции административной части (рис. 2)

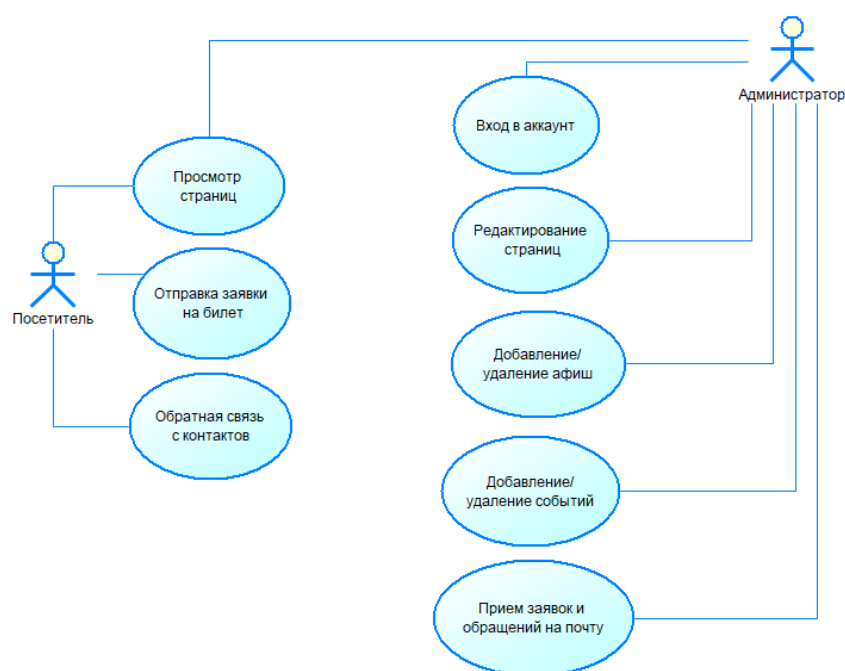


Рис. 2. Диаграмма прецедентов веб-ресурса

Затем разработана диаграмма последовательностей [4] для сценария «Покупка билета», отражающая цепочку взаимодействия пользователя со средой. (рис. 3).

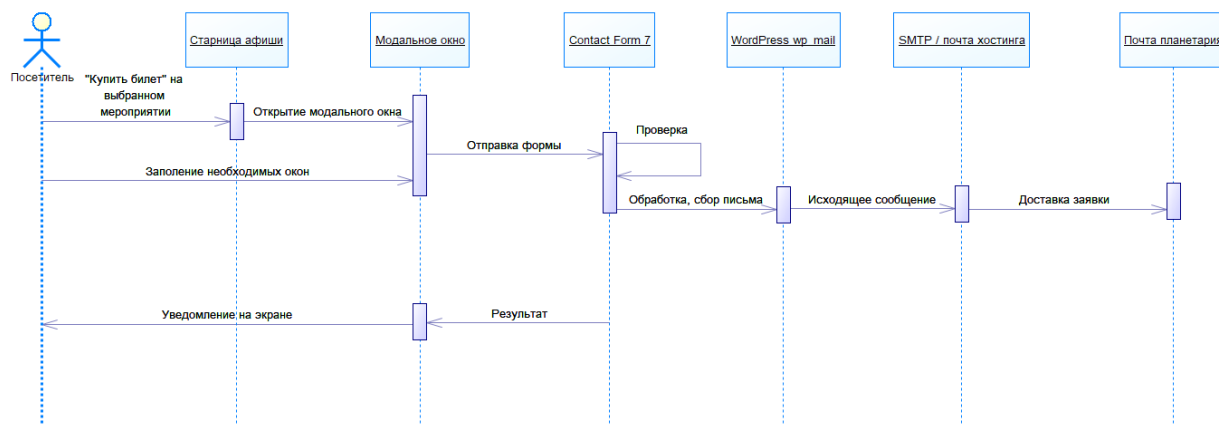


Рис. 3. Диаграмма последовательностей веб-ресурса

Перед реализацией программной части в среде Figma переработан дизайн шаблонов страниц веб-ресурса (рис. 4).

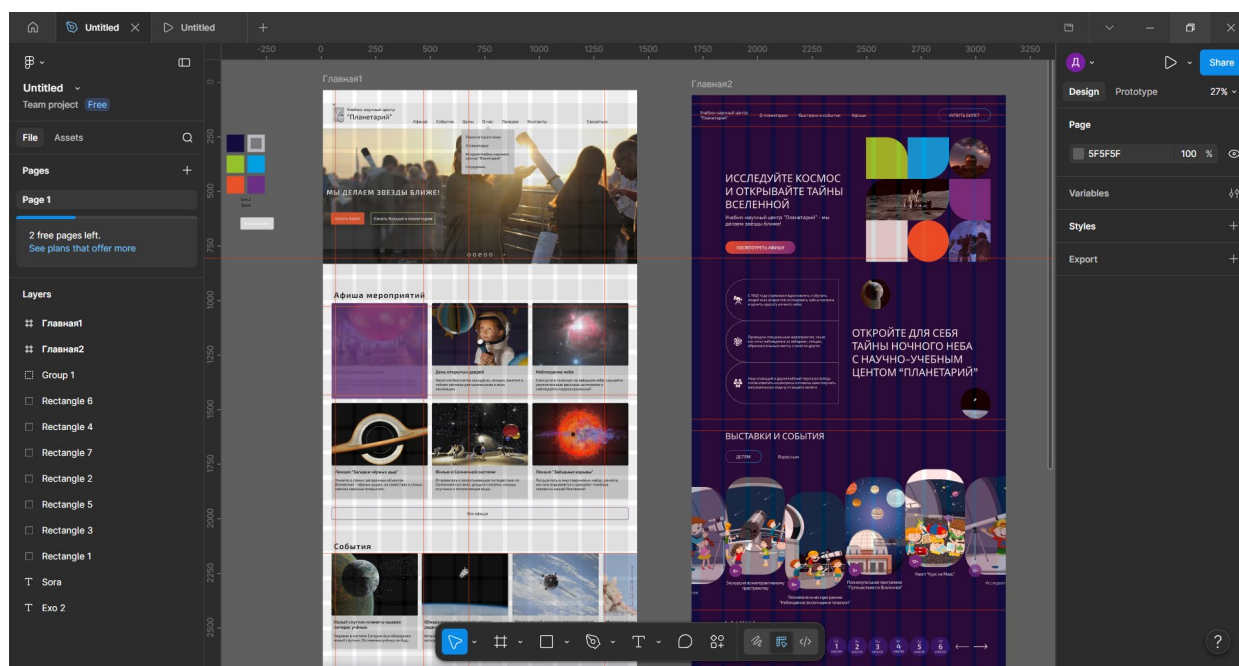


Рис. 4. Работа с дизайнами шаблона веб-ресурса

Далее определён стек разработки и сопровождения. В качестве платформы выбрана система управления содержимым WordPress [5]. Использование административной панели CMS позволяет сотрудникам центра управлять контентом

публичной части, исключая необходимость ручного редактирования HTML-файлов [6].

Интерфейс реализован на базе индивидуальной темы WordPress с применением HTML, CSS, JavaScript и шаблонов PHP. Доработка кода выполнялась в Visual Studio Code. Структурирование полей записей (афиша, события, сотрудники и др.) обеспечено плагином Advanced Custom Fields PRO, а для форм «Купить билет» и обратной связи используется Contact Form 7 с настройкой шаблонов писем и адресата. В число дополнительных плагинов входят Yoast SEO, Cypylitera, Classic Editor, ARI Fancy Lightbox и Theme Editor.

Результаты

На основе проведённого анализа предметной области, сравнения с существующим веб-ресурсом на платформе Tilda, проектирования структуры сайта и диаграмм прецедентов была разработана индивидуальная тема WordPress, включающая шаблоны PHP, таблицы стилей CSS и сценарии JavaScript. Технологическую основу составили CMS WordPress, СУБД MySQL, плагины Advanced Custom Fields PRO и Contact Form 7, а также средства проектирования интерфейса Figma и среда разработки Visual Studio Code.

Контент хранится в базе данных MySQL в стандартной для WordPress модели. Публикации и страницы создаются и редактируются через административную панель, а дополнительные атрибуты афиши и событий задаются с помощью плагина Advanced Custom Fields PRO и сохраняются вместе с записями в базе данных. Этот плагин позволяет структурировать контент путём задания пользовательских полей, что обеспечивает единообразие данных при выводе в шаблонах темы. В административной панели отображается форма редактирования записи афиши с заполненными полями и статусом публикации (рис. 5).

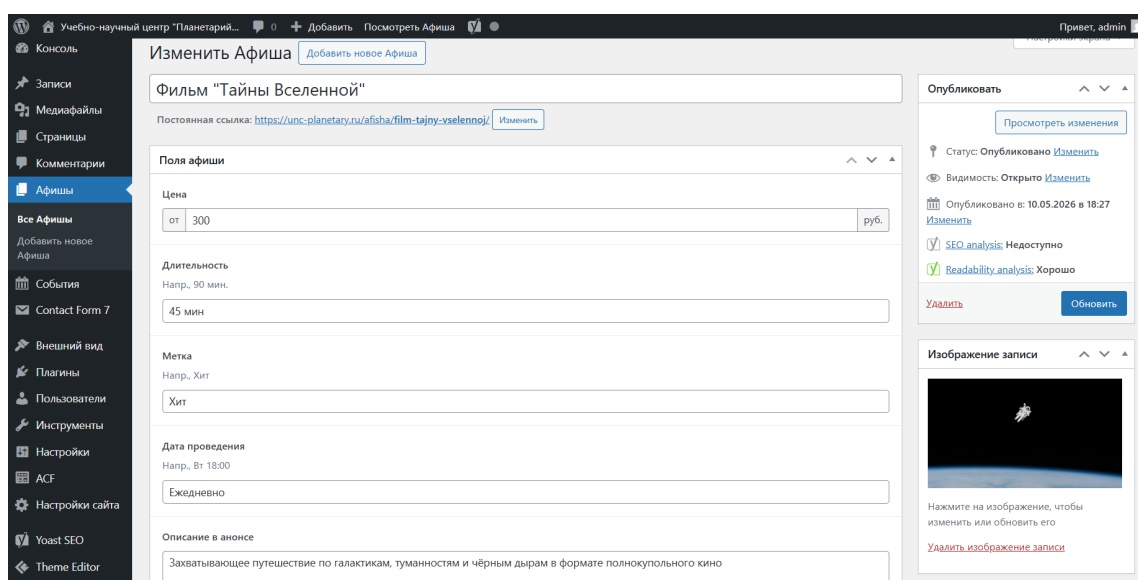


Рис. 5. Пример созданной записи афиши

Публичные разделы «Афиша», «События», «О центре», «Галерея» и «Контакты» формируются шаблонами темы, а контент подставляется из базы данных при обращении посетителя к странице. Пример отображения карточки мероприятия с актуальными данными из CMS приведён на рис. 6.

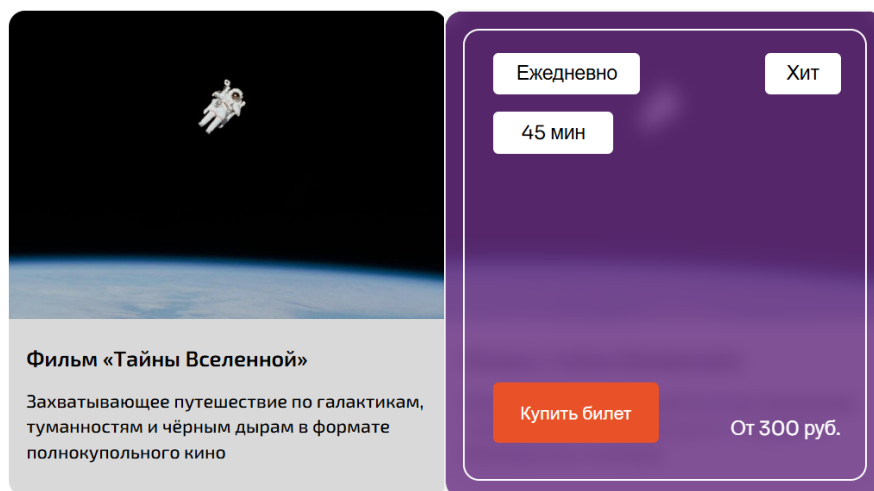


Рис. 6. Пример опубликованной записи афиши

Для оформления заявки на участие в мероприятии реализован сценарий «Купить билет». При нажатии кнопки у выбранной записи афиши открывается модальное окно с полями имени, электронной почты и комментария. В заявку автоматически передаётся контекст выбранного мероприятия. Обработка формы выполняется плагином Contact Form 7 с последующей отправкой письма на корпоративную почту планетария через механизм `wp_mail` и почтовый канал хостинга. На рисунке 7 показано модальное окно заявки и подтверждение успешной отправки.

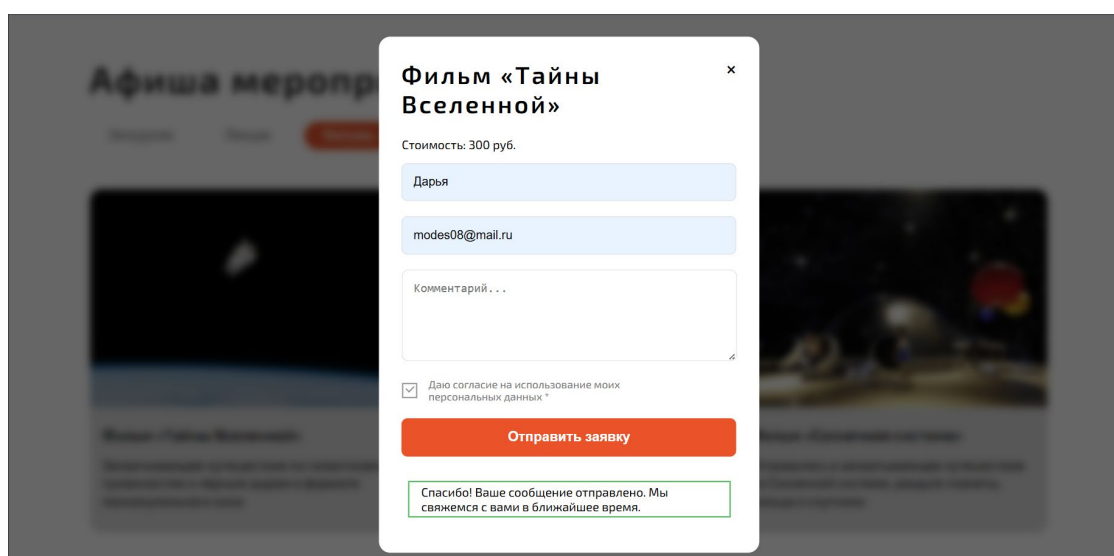


Рис. 7. Пример успешно отправленной заявки

Для просмотра изображений в разделе «Галерея» используется плагин ARI Fancy Lightbox, который обеспечивает увеличение фотографий во всплывающем окне без перехода на отдельную страницу.

Администрирование веб-ресурса осуществляется через стандартную панель WordPress (wp-admin). Это позволяет сотрудникам центра создавать и редактировать афиши и события, обновлять тексты страниц, а также управлять медиа-файлами и настройками отображения без привлечения разработчика.

Отдельно проверена форма обратной связи на странице «Контакты», включающая поля имени, e-mail, темы и сообщения, а также доставка писем на электронный ящик планетария (рис. 8).

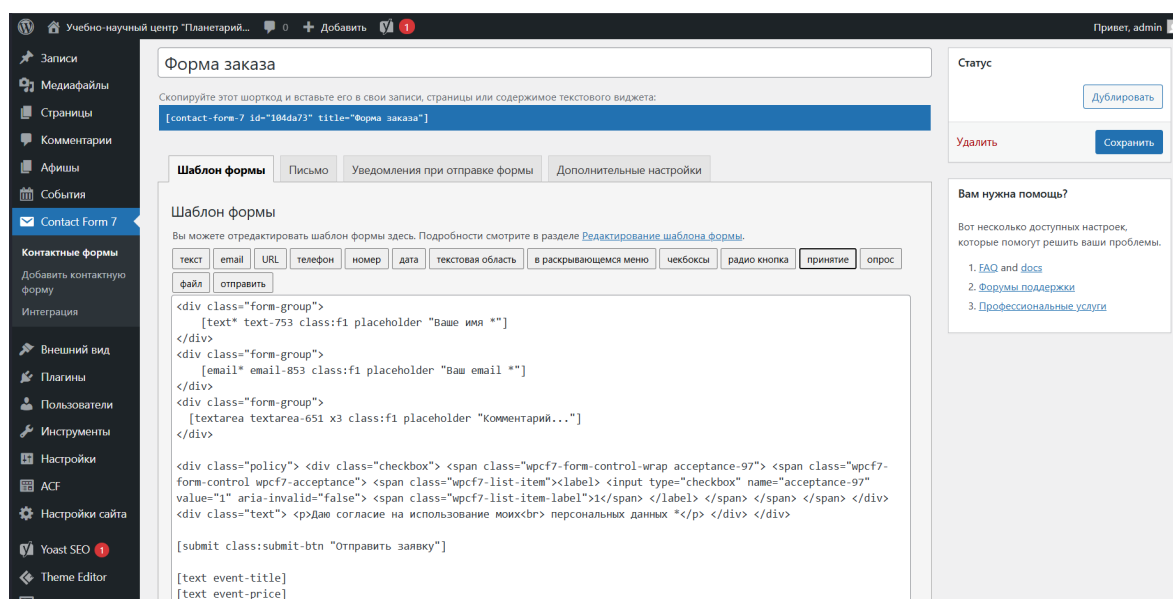


Рис. 8. Пример стандартного шаблона формы заявки заказа

По результатам тестирования подтверждена работоспособность ключевых сценариев в браузерах Google Chrome и Yandex Browser, а также корректное отображение страниц на мобильных и десктопных разрешениях экрана. Реализация адаптивной вёрстки выполнена в соответствии с принципами отзывчивого веб-дизайна [7]. Модернизированный веб-ресурс размещён на хостинге Timeweb и готов к самостоятельному администрированию сотрудниками центра «Планетарий».

Заключение

В статье рассмотрен процесс модернизации веб-ресурса учебно-научного центра «Планетарий» СГУГиТ. Целью работы стало обновление существующего представления центра в сети для обеспечения актуального информирования посетителей о мероприятиях и услугах, а также организации удобного администрирования контента.

В отличие от предыдущего решения на платформе Tilda, которое ограничивало гибкость сопровождения и не предоставляло целевых механизмов для при-

ёма заявок, в ходе модернизации была внедрена полноценная система управления содержимым с структурированным отображением событий и интегрированными сервисными сценариями.

В качестве технологической основы модернизации выбрана CMS WordPress с использованием индивидуальной темы, разработанной на HTML, CSS, JavaScript и PHP, а также хостинг Timeweb и СУБД MySQL. В ходе работы спроектированы обновлённая структура сайта и диаграммы прецедентов. Разделы «Афиша», «События», «Галерея» и «Контакты» адаптированы под новую платформу, кроме того, добавлены модальная форма «Купить билет» с привязкой к конкретному мероприятию и форма обратной связи.

Итог: обновлённый веб-ресурс обеспечивает актуальное и структурированное информирование посетителей, упрощает администрирование для сотрудников центра за счёт ролевой модели WordPress и готов к самостоятельной эксплуатации. Выполненная модернизация демонстрирует практическое применение современных CMS-технологий для преобразования образовательных и культурно-просветительских веб-ресурсов вуза, а также способствует повышению качества представления центра «Планетарий» в сети.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Шешко А. С. Проектирование дизайн пространства сайта для Томского планетария // Электронный архив ТПУ: <http://earchive.tpu.ru/handle/11683/74453> (дата обращения: 15.05.2026).
2. Структура сайта: схемы, советы по разработке, примеры // Генератор продаж URL: <https://sales-generator.ru/blog/struktura-sayta/> (дата обращения: 15.05.2026).
3. Диаграмма Прецедентов (англ. Use Case Diagram) // Хабр URL: <https://habr.com/ru/articles/907804/> (дата обращения: 15.05.2026).
4. Диаграмма последовательности (sequence-диаграмма) // Википедия URL: <https://habr.com/ru/articles/814769/> (дата обращения: 15.05.2026).
5. Пономаренко К. О., Свириденко А. А., Никитин П. В. Преимущества движка WordPress в качестве системы современного управления интерактивным обучением // Международный научный журнал "Инновационная наука". - 2015. - № 11. (дата обращения: 15.05.2026).
6. Преобразование HTML-сайта в WordPress // WordPress URL: <https://wordpress.com/ru/support/import/convert-an-html-site-to-wordpress/> (дата обращения: 15.05.2026).
7. Маркотт И. Отзывчивый веб-дизайн = Responsive Web Design. - Москва : Манн, Иванов и Фербер, 2012. - 31-82 с.

© Д. Д. Модестова, Е. Ю. Воронкин, 2026

Р. В. Морозов^{1✉}, П. Ю. Бугаков¹

Сравнительный анализ методов обмена сообщениями в распределенной системе мониторинга сетевых узлов

¹Сибирский государственный университет геосистем и технологий,
г. Новосибирск, Российская Федерация
e-mail: me@craftersmine.ru

Аннотация. В работе рассматриваются и анализируются подходы к организации обмена сообщениями между компонентами в распределенных системах мониторинга. Проведено сравнение транспортных протоколов TCP и UDP, прикладного протокола HTTP в рамках архитектурного стиля REST, а также протокола WebSocket с точки зрения их применимости в системах, требующих обмена данными в режиме, близком к реальному времени. Особое внимание уделено таким характеристикам, как задержка передачи данных, надежность доставки, поддержка двустороннего обмена, требования к вычислительным ресурсам и особенностям безопасности. На основе проведенного анализа обоснован выбор WebSocket в качестве основного механизма обмена сообщениями между координатором и агентами, а также использование HTTP для односторонней передачи данных от агентов к координатору. Результаты исследования демонстрируют, что комбинированное использование WebSocket и HTTP позволяет обеспечить баланс между производительностью, масштабируемостью и сложностью реализации распределенной системы мониторинга.

Ключевые слова: сети, обмен сообщениями, протоколы связи, tcp, udp, http, websockets

R. V. Morozov^{1✉}, P. Yu. Bugakov¹

Comparative analysis of messaging methods in a distributed system for monitoring network nodes

¹Siberian State University of Geosystems and Technologies, Novosibirsk, Russian Federation
e-mail: me@craftersmine.ru

Abstract. The work examines and analyzes approaches to organizing message exchange between components in distributed monitoring systems. A comparison is conducted of the TCP and UDP transport protocols, the HTTP application protocol within the REST architectural style, and the WebSocket protocol in terms of their applicability in systems requiring data exchange in near real-time conditions. Special attention is given to characteristics such as data transmission latency, delivery reliability, support for bidirectional communication, computational resource requirements, and security aspects. Based on the analysis performed, WebSocket is justified as the primary mechanism for message exchange between the coordinator and agents, while HTTP is used for unidirectional data transmission from agents to the coordinator. The results of the study demonstrate that the combined use of WebSocket and HTTP provides a balance between performance, scalability, and implementation complexity in a distributed monitoring system.

Keywords: networks, message exchange, communication protocols, tcp, udp, http, websockets

Введение

В современных вычислительных и сетевых системах существует широкий спектр способов организации обмена данными между узлами, от передачи сырых

байтовых потоков до использования высокоуровневых проприетарных протоколов прикладного уровня. Особенно критичную роль выбор механизма обмена играет в системах мониторинга и управления, где требуется своевременный обмен информацией о событиях, метриках производительности, состоянии узлов, а также команд управления и сценариев реагирования на инциденты, связанные с непредвиденной остановкой работающих служб, ошибками в их работе, остановкой рабочих станций или серверов.

Главная особенность таких систем – распределенная архитектура, в которой есть отдельный координатор и множество агентов, работающих на управляемых узлах (рис. 1).

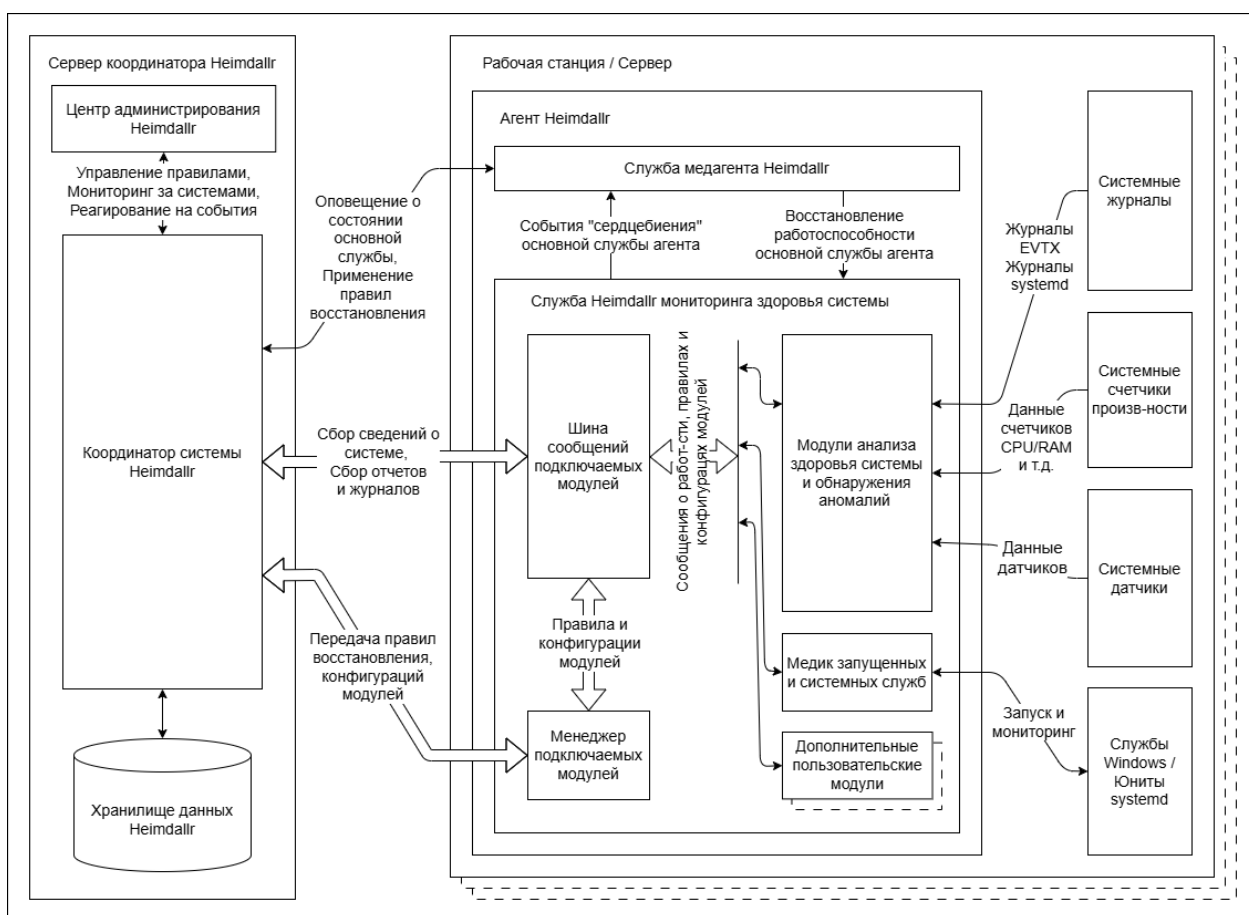


Рис. 1. Концептуальная схема архитектуры

Взаимодействие между этими компонентами должно соответствовать ряду важных требований: обеспечивать низкие задержки при передачи данных, поддерживать двусторонний обмен сообщениями, сохранять устойчивость к сбоям и эффективно использовать вычислительные ресурсы при сериализации и десериализации данных. Не менее важным является и высокий уровень защиты передаваемой информации.

Отдельное внимание уделяется безопасности, поскольку агенты обычно обладают расширенными привилегиями в системе. Ошибки или ненадежная реали-

зация механизма обмена сообщениями могут привести к компрометации всей инфраструктуры и вызвать серьезные последствия.

С учетом большого количества существующих сетевых протоколов и архитектурных подходов возникает задача выбора наиболее подходящего способа организации взаимодействия между координатором и агентами. Такой выбор должен опираться на сравнительный анализ различных технологий с точки зрения производительности, надежности, сложности реализации и уровня безопасности.

Способы сетевого взаимодействия

В рамках исследования были рассмотрены четыре распространенных подхода к организации сетевого взаимодействия: транспортные протоколы TCP и UDP, прикладной протокол HTTP, используемый в архитектуре REST, а также протокол WebSocket. Эти технологии относятся к разным уровням сетевой модели и обладают собственными особенностями, поэтому для выбора подходящего решения требуется их сравнительный анализ [1, 2].

TCP (Transmission Control Protocol) представляет собой транспортный протокол, функционирующий поверх IP и обеспечивающий надежную передачу данных между узлами сети. Для этого в протоколе предусмотрены процедура установления логического соединения, проверка целостности передаваемых данных, подтверждение факта доставки и повторная отправка утерянных сегментов. Благодаря перечисленным механизмам TCP гарантирует корректную и упорядоченную доставку информации. Вместе с тем такая надежность достигается ценой дополнительных задержек: обработка данных становится возможной только после приема всех сегментов в правильной последовательности. В системах, где требуется обмен данными в реальном времени, данное обстоятельство может служить существенным ограничением [3, 4].

Кроме того, TCP является потоковым протоколом и не задает формат передаваемых данных. Поэтому разработчику приходится самостоятельно реализовывать протокол прикладного уровня, который определял бы структуру сообщений, правила их интерпретации и способы сериализации. Это усложняет реализацию, но одновременно предоставляет гибкость при выборе формата обмена данными в соответствии с конкретной задачей.

После установления соединения TCP обеспечивает двусторонний обмен данными между клиентом и сервером. Упрощенная схема установления соединения и процесса передачи данных с использованием TCP приведена на рис. 2.

UDP (User Datagram Protocol) представляет собой транспортный протокол, функционирующий поверх IP и, в отличие от TCP, не обеспечивающий установление соединения, гарантии доставки, упорядоченности и повторной передачи данных. Передача осуществляется в виде независимых датаграмм, которые могут быть потеряны, дублированы или доставлены вне исходного порядка.

Отсутствие механизмов контроля надежности и упорядоченности снижает накладные расходы и задержки передачи, что делает UDP предпочтительным для приложений реального времени, включая системы потоковой передачи данных и

сетевые игры. Однако обеспечение целостности, порядка доставки и обработки потерь при необходимости должно реализовываться на уровне приложения, что увеличивает сложность разработки [5].

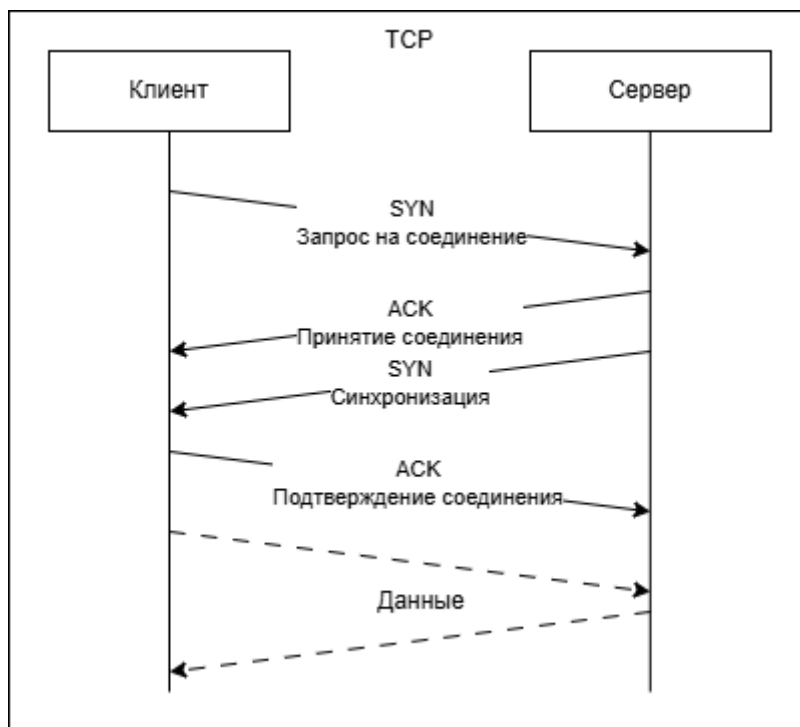


Рис. 2. Диаграмма установления соединения и передачи данных используя протокол TCP

UDP не накладывает ограничений на структуру передаваемых данных, что требует использования протокола прикладного уровня для их интеграции. Обобщенная схема передачи данных с использованием UDP представлена на рис. 3.

HTTP (Hypertext Transfer Protocol) представляет собой протокол прикладного уровня, широко используемый для взаимодействия компонентов распределенных систем по модели «клиент-сервер». Архитектурный стиль REST (Representational State Transfer) определяет принципы организации такого взаимодействия, включая ресурсно-ориентированный подход, унифицированный интерфейс и отсутствие сохранения состояния между запросами.

В рамках архитектуры REST обмен данными выполняется с применением текстовых форматов, например (JSON) JavaScript Object Notation или расширяемого языка разметки XML (Extensible Markup Language). Передача двоичных данных возможна как в непосредственном виде, в частности с использованием типа MIME «application/octet-stream», так и посредством их инкапсуляции в текст, например при помощи кодировки Base64. Протокол HTTP поддерживает механизмы кодирования содержимого, предназначенные для сжатия данных, к числу которых относятся алгоритмы deflate, gzip и brotli [6, 7].

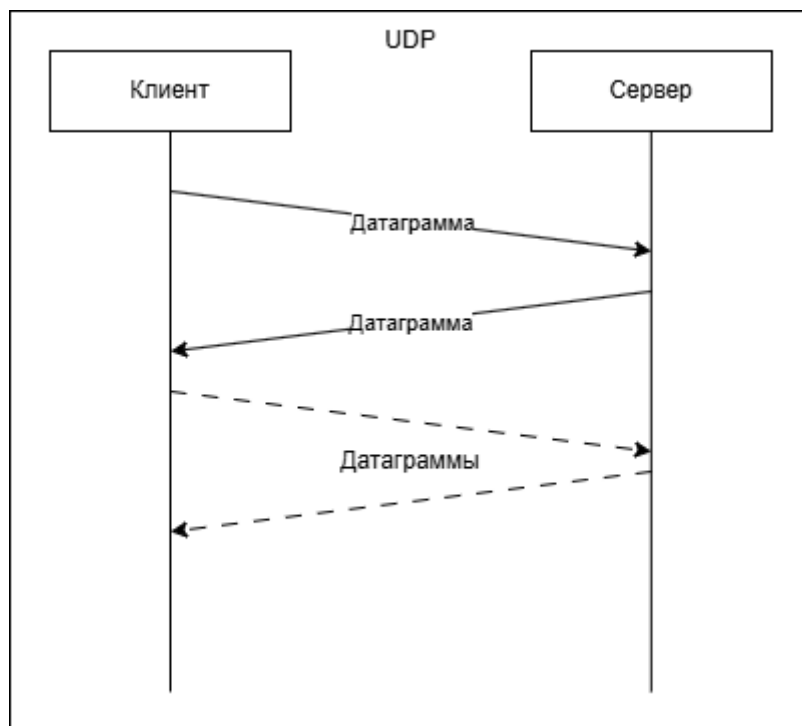


Рис. 3. Схема передачи данных с применением протокола UDP

Структура запросов и ответов в протоколе HTTP, включая методы, заголовки и коды состояния, задается явным образом, что заметно облегчает разработку как клиентской, так и серверной частей системы.

Для обеспечения защиты передаваемых данных применяется расширение HTTP, известное как HTTPS и базирующееся на протоколе TLS. Данное расширение выполняет шифрование канала связи и аутентификацию соединения. Использование готового решения позволяет не разрабатывать собственные криптографические механизмы, одновременно оставляя возможность дополнительной защиты на других уровнях при необходимости.

Существенным ограничением данного подхода в контексте рассматриваемой системы является необходимость наличия HTTP-сервера на стороне узла, предоставляющего данные. В случае двустороннего взаимодействия между координатором и агентами это может потребовать развертывания серверных компонентов на обеих сторонах. Наличие открытых сетевых портов и веб-сервисов на стороне агентов потенциально увеличивает поверхность атаки и предъявляет повышенные требования к обеспечению безопасности. Обобщенная схема выполнения HTTP/HTTPS-запроса в рамках REST-подхода представлена на рис. 4.

WebSocket представляет собой протокол прикладного уровня, обеспечивающий полно-дуплексный обмен данными между клиентом и сервером поверх одного установленного TCP-соединения. В отличие от HTTP, где взаимодействие строится по модели «запрос-ответ», WebSocket после первоначального HTTP-рукопожатия устанавливает постоянное соединение. Это позволяет обеим сторо-

нам обмениваться сообщениями независимо друг от друга и поддерживать двустороннюю связь в реальном времени [8].

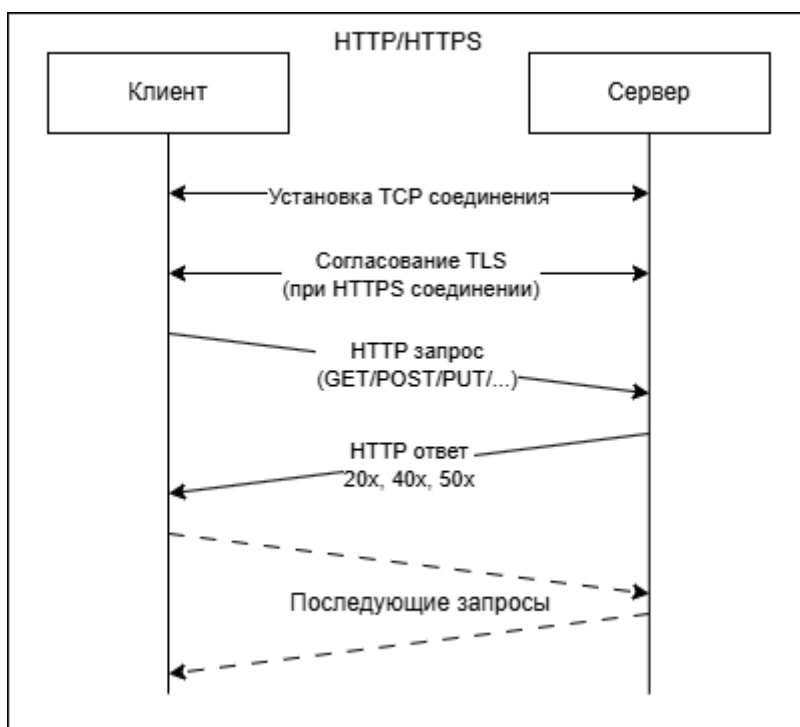


Рис. 4. Диаграмма выполнения HTTP-REST запроса

Протокол поддерживает передачу как текстовых, так и двоичных данных, благодаря чему можно использовать различные форматы сериализации, включая JSON, XML, и собственные бинарные протоколы. Это дает больше гибкости при проектировании системы обмена сообщениями и позволяет оптимизировать передачу данных с учетом требований к производительности и объему сетевого трафика.

Защищенная версия протокола – WSS- использует TLS-шифрование по аналогии с HTTPS, обеспечивая конфиденциальность и целостность передаваемых данных. За счет постоянного соединения и отсутствия необходимости заново устанавливать соединение для каждого сообщения, WebSocket обеспечивает низкие задержки передачи данных. Благодаря этому протокол хорошо подходит для систем, которым требуется взаимодействие, близкое к реальному времени [9].

WebSocket позволяет организовать эффективный двусторонний обмен данными без необходимости постоянного опроса сервера (polling). Однако для работы протокола требуется поддерживать активные соединения, что при большом количестве агентов может увеличивать нагрузку на сервер. Обобщенная схема обмена сообщениями с использованием WebSocket представлена на рис. 5.

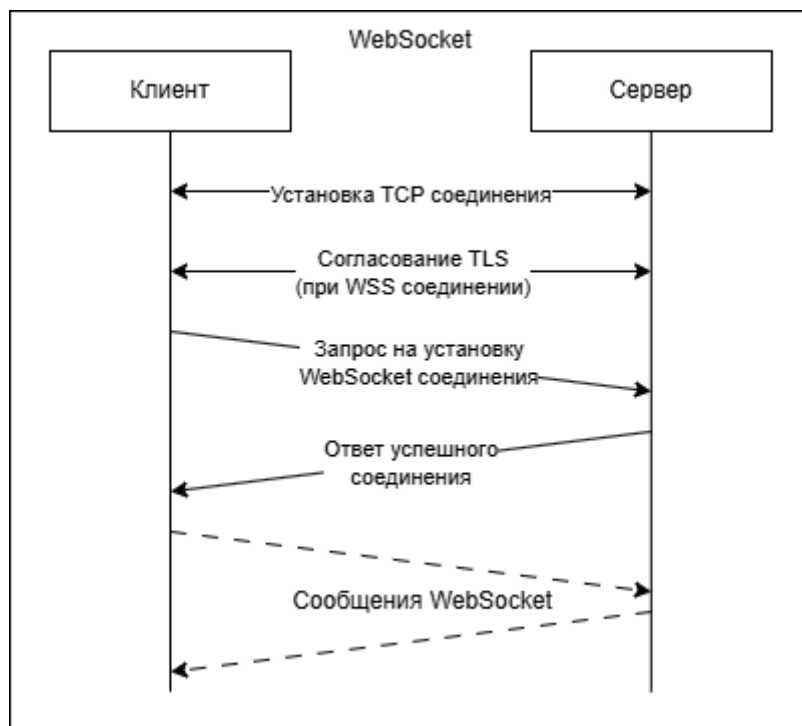


Рис. 5. Передача сообщений с использованием протокола WebSocket

Результаты сравнительного анализа

Сравнительный анализ протоколов TCP, UDP, HTTP (REST) и WebSocket позволяет выделить их ключевые различия по ряду характеристик, важных при проектировании распределенных систем: надежность доставки, задержки, модель взаимодействия, накладные расходы, сложность реализации и уровень безопасности.

Протокол TCP обеспечивает высокую надежность благодаря таким механизмам, как подтверждение доставки, контроль целостности и повторная передача данных. Это делает его пригодным для задач, в которых критически важны полнота и корректность информации. Вместе с тем указанная надежность сопровождается дополнительными задержками, возникающими как на этапе установления соединения, так и в процессе ожидания всех сегментов перед началом обработки данных. В отличие от TCP, протокол UDP снижает задержки за счет отсутствия механизмов гарантированной доставки, что делает его более эффективным для сценариев, работающих в реальном времени. Однако при использовании UDP логику обеспечения надежности приходится реализовывать на прикладном уровне. В обоих случаях разработчику зачастую требуется самостоятельно проектировать прикладные механизмы обмена сообщениями, что увеличивает сложность разработки.

Протокол HTTP в рамках REST-архитектуры, будучи протоколом прикладного уровня, предоставляет стандартизированный способ взаимодействия между компонентами системы. К его ключевым достоинствам относятся простота интеграции, широкая поддержка различными платформами и встроенные механизмы обеспечения безопасности. Тем не менее модель взаимодействия «запрос-ответ»,

а также необходимость установления соединения для каждого запроса приводят к росту накладных расходов и задержек. Данное обстоятельство ограничивает применение REST в сценариях, требующих интенсивного двустороннего обмена данными.

WebSocket сочетает в себе надежность TCP при передаче данных и возможность постоянного двустороннего обмена сообщениями. После установления соединения он обеспечивает низкие задержки и эффективную передачу данных без необходимости повторного создания соединения. Это делает его особенно подходящим для систем, где требуется обмен информацией в реальном времени. В то же время необходимость поддержания большого числа открытых соединений может приводить к дополнительной нагрузке на серверную сторону.

Сводное сравнение рассмотренных протоколов приведено в табл. 1.

Таблица 1. Сравнение функционала рассмотренных протоколов

Характеристика	TCP	UDP	HTTP (REST)	WebSocket
Уровень модели OSI	Транспортный	Транспортный	Прикладной	Прикладной
Надежность	Высокая	Отсутствует	Высокая (через TCP)	Высокая (через TCP)
Установление соединения	Требуется	Не требуется	Требуется при каждом запросе	Требуется однократно
Модель взаимодействия	Поток (Stream)	Датаграммы (Datagram)	Запрос-Ответ (Request-Response)	Сообщения (Messages)
Накладные расходы подключения	Средние	Минимальные	Высокие	Низкие после установления соединения
Сложность реализации передачи сообщений	Средняя	Высокая	Низкая	Низкая
Безопасность	Требуется дополнительных механизмов	Требуется дополнительных механизмов	HTTPS (TLS)	WSS (TLS)

Для получения практических данных о задержках было разработано приложение для тестирования, позволяющее оценить производительность каждого протокола по времени передачи одного пакета. В рамках тестирования использовались пакеты различных размеров – от 1 байта до 32768 байт (с шагом по степеням двойки). Данные передавались между компьютером и удаленным сервером в локальной сети.

Полученные данные о времени передачи данных приведено в табл. 2.

По получившимся данным был построен график зависимости размера пакета к длительности его передачи по локальной сети (рис. 6). Протокол HTTP не

был включен в данный график, так как задержки передачи данных при использовании данного протокола многократно превышали ожидаемые результаты.

Таблица 2. Время передачи данных по разным протоколам в локальной сети

Размер пакета (Б)	WebSocket (мс)	UDP (мс)	TCP (мс)	HTTP (мс)
1 (прогрев)	1,078	0,167	0,066	199,904
1	0,028	0,080	0,046	3,636
2	0,020	0,037	0,052	3,327
4	0,032	0,054	0,033	4,616
8	0,024	0,031	0,036	2,436
16	0,028	0,047	0,052	3,143
32	0,036	0,082	0,067	3,252
64	0,025	0,051	0,042	18,054
128	0,032	0,042	0,036	6,356
256	0,044	0,067	0,031	3,306
512	0,028	0,062	0,037	5,764
1024	0,064	0,074	0,038	3,686
2048	0,093	0,034	0,072	3,542
4096	0,120	0,086	0,037	3,540
8192	0,787	0,071	0,039	8,411
16384	0,516	0,076	0,075	3,800
32768	1,040	0,120	0,026	3,442
Общее время	3,995	1,181	0,785	280,215
Минимальное	0,020	0,031	0,026	2,436
Максимальное	1,078	0,167	0,075	199,904
Среднее	0,235	0,069	0,046	16,483

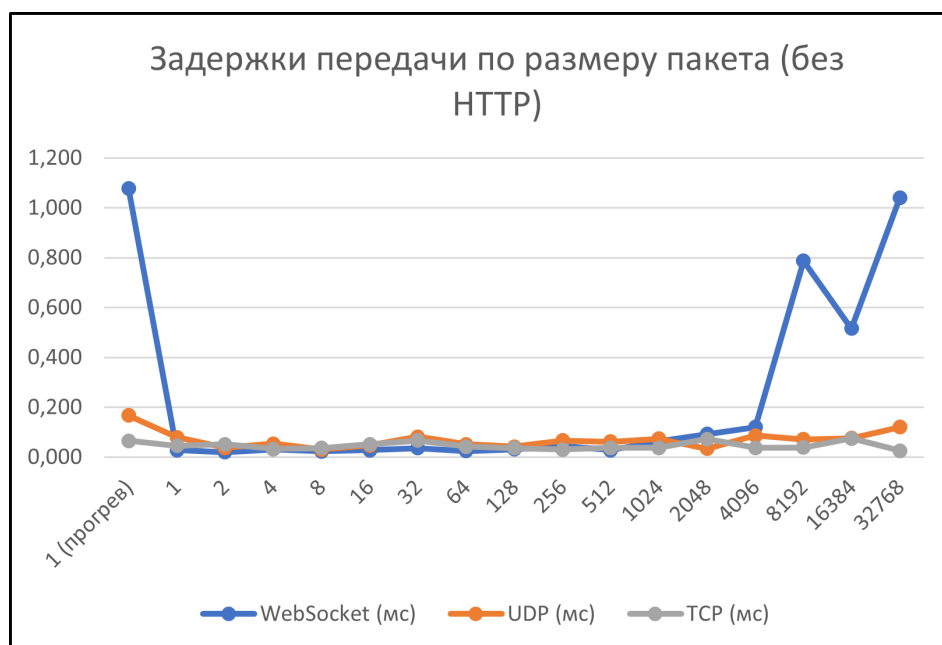


Рис. 6. График отношения размера пакета к длительности передачи пакета

По данным графика, производительность WebSocket сопоставима с TCP и UDP, за исключением пакетов крупнее 4 КБ – их передача занимает заметно больше времени, но все равно в разы быстрее, чем при использовании HTTP.

По результатам анализа можно сделать вывод, что выбор протокола сетевого взаимодействия должен определяться требованиями конкретной системы. Для задач, где критична надежность передачи, целесообразно использовать TCP или протоколы прикладного уровня, основанные на нем (HTTP, WebSocket). В сценариях, требующих минимальных задержек и допускающих потери данных, предпочтение следует оставлять UDP. Для построения масштабируемых веб-сервисов, оптимальным является HTTP (REST), в то время как для систем с интенсивным двусторонним обменом в реальном времени, наиболее подходящим и эффективным решением является WebSocket.

Заключение

В ходе исследования были рассмотрены основные подходы к организации обмена сообщениями в распределенных системах мониторинга между координатором и агентами, включая транспортные протоколы TCP и UDP, прикладной протокол HTTP с архитектурным стилем REST, а также протокол WebSocket. Проведенный сравнительный анализ показал, что рассмотренные решения существенно различаются по таким характеристикам, как задержка передачи данных, надежность доставки, требования к вычислительным ресурсам и сложность реализации. В рамках выбранного подхода протокол WebSocket обеспечивает эффективную организацию двустороннего обмена данными благодаря поддержанию постоянного соединения и снижению накладных расходов, связанных с повторным установлением сеансов связи. Указанные свойства делают его пригодным для передачи событий мониторинга и управляющих сообщений в режиме, близком к реальному времени.

Дополнительно в архитектуре системы обосновано использование протокола HTTP в одностороннем режиме передачи данных от агентов к координатору для вспомогательных задач, к числу которых относится периодическая отправка отчетной информации. Такой подход позволяет снизить нагрузку на постоянное WebSocket-соединение и разделить каналы передачи данных в соответствии с их назначением. Следовательно, совместное применение протоколов WebSocket и HTTP позволяет достичь баланса между производительностью, масштабируемостью и сложностью разработки. Дополнительным преимуществом выступает повышение гибкости архитектуры, что особенно важно для распределенных систем мониторинга.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Таненбаум Э., Уэзеролл Д. Компьютерные сети. - 5-е изд. - СПб.: Питер, 2012. - 960 с.
2. Куроуз Дж., Росс К. Компьютерные сети: нисходящий подход. - 6-е изд. - М.: Вильямс, 2016. - 912 с.
3. Стивенс У. Р. TCP/IP Illustrated. Том 1. Протоколы. - М.: Вильямс, 2003. - 576 с.
4. Постел Дж. Transmission Control Protocol (TCP) (RFC 793). - IETF, 1981. - URL: <https://datatracker.ietf.org/doc/html/rfc793> (дата обращения: 09.04.2026).

5. Постел Дж. User Datagram Protocol (UDP) (RFC 768). - IETF, 1980. - URL: <https://datatracker.ietf.org/doc/html/rfc768> (дата обращения: 09.04.2026).
6. Филдинг Р. Т., и др. Hypertext Transfer Protocol (HTTP/1.1: Semantics and Content) (RFC 9110). - IETF, 2022. - URL: <https://datatracker.ietf.org/doc/html/rfc9110> (дата обращения: 09.04.2026).
7. Ричардсон Л., Руби С. RESTful Web Services. - Sebastopol: O'Reilly Media, 2007. – 448 с.
8. Бугаков А. WebSocket: как это работает и зачем он нужен. - Хабр, 2018. - URL: <https://habr.com/ru/articles/346306/> (дата обращения: 09.04.2026).
9. Фетт И., Мельников А. The WebSocket Protocol (RFC 6455). - IETF, 2011. - URL: <https://datatracker.ietf.org/doc/html/rfc6455> (дата обращения: 09.04.2026).

© *Р. В. Морозов, П. Ю. Бугаков, 2026*

Д. А. Остриков¹✉

Метод адаптивного контроля привилегированного доступа на основе мультипликативной агрегации рисков

¹Национальный исследовательский ядерный университет «МИФИ»,
г. Москва, Российская Федерация
e-mail: denis.ostrikov.01@mail.ru

Аннотация. В статье предложен метод адаптивного контроля привилегированного доступа к серверам Linux-инфраструктуры, основанный на количественной оценке рисков. Разработана мультипликативная модель агрегации пяти контекстных факторов запроса доступа: истории аномалий пользователя, критичности запрашиваемых ролей, подсети источника, геолокации и времени запроса. Модель обладает свойством вето: единственный критический фактор приводит к максимальной оценке риска независимо от остальных. Проведено сравнение трёх методов агрегации на публичном датасете из 31 млн записей аутентификации; мультипликативная модель показала AUC-ROC = 0,883, сопоставимый с взвешенной суммой (0,885), при наличии теоретически обоснованных преимуществ. На основе интегральной оценки риска система динамически определяет параметры выдаваемого SSH-сертификата: срок действия по экспоненциальной функции, уровень верификации по ступенчатой функции с двумя порогами и режим записи сессии. Метод интегрирован с подсистемой PAM операционной системы Linux. Реализован программный прототип, валидированный на виртуальном стенде. Показано сокращение срока действия аутентификационных данных на 64–82% по сравнению со статическими политиками. Симуляция работы 50 пользователей на протяжении 7 дней подтвердила корректное разделение профилей и адаптацию системы к изменениям поведения через механизм обратной связи.

Ключевые слова: адаптивный контроль доступа, оценка рисков, Linux-инфраструктура, SSH-сертификаты, PAM, Zero Trust

D. A. Ostrikov¹✉

Adaptive privileged access control method based on multiplicative risk aggregation

¹National Research Nuclear University MEPhI, Moscow, Russian Federation
e-mail: denis.ostrikov.01@mail.ru

Abstract. The paper proposes a method for adaptive privileged access control in Linux infrastructure based on quantitative risk assessment. A multiplicative aggregation model combining five contextual request factors is developed: user anomaly history, requested role criticality, source subnet, geolocation, and request time. The model features a veto property where a single critical factor forces maximum risk regardless of others. Three aggregation methods are compared on a public dataset of 31 million authentication records; the multiplicative model achieved AUC-ROC of 0.883, comparable to the weighted sum (0.885), while offering theoretically grounded advantages. Based on the integral risk score, the system dynamically determines SSH certificate parameters: validity period via an exponential function, verification level via a step function with two thresholds, and session recording mode. The method integrates with the Linux PAM subsystem. A software prototype has been implemented and validated on a virtual testbed, demonstrating 64–82% reduction in authentication creden-

tial validity period compared to static policies. A 7-day simulation with 50 users confirmed correct profile separation and behavioral adaptation through a feedback mechanism.

Keywords: adaptive access control, risk assessment, Linux infrastructure, SSH certificates, PAM, Zero Trust

Введение

Управление привилегированным доступом к серверам корпоративной инфраструктуры остаётся одной из ключевых задач информационной безопасности. Компрометация привилегированной учётной записи позволяет злоумышленнику выполнять произвольные команды на серверах, извлекать данные и перемещаться по инфраструктуре. Согласно отчёту Verizon DBIR 2024, злоупотребление привилегиями присутствует в 82% успешных инцидентов [1], а среднее время обнаружения компрометации составляет 207 дней [2].

Традиционный подход к управлению доступом основан на статических политиках: пользователю назначается фиксированный набор ролей, а параметры сеанса – срок действия удостоверяющих данных, уровень верификации, режим аудита – одинаковы для всех запросов. Такой подход не учитывает контекст: подключение в рабочее время из корпоративной сети и подключение ночью с неизвестного адреса оцениваются одинаково. Архитектура Zero Trust [3, 13] требует непрерывной верификации каждого запроса, однако на практике её реализация сводится к фиксированным многофакторным проверкам без количественной оценки риска.

Исследования в области интеллектуальной поддержки принятия решений в информационной инфраструктуре [4, 5] и методология интеллектуально-адаптивного управления [6] показали эффективность адаптивных подходов к обеспечению безопасности. Модели риск-ориентированного контроля доступа [7, 8] позволяют принимать градуированные решения, однако используют взвешенную сумму факторов [7, 15], которая допускает компенсацию критического фактора благоприятными значениями остальных. Кроме того, существующие решения либо ориентированы на специфические домены [9, 14], либо не имеют практической реализации, интегрированной с механизмами операционной системы [10].

Проблема заключается в том, что существующие методы агрегации рисков не гарантируют, что единственный критический фактор – например, запрос привилегированной роли – приведёт к максимальному уровню защиты. Кроме того, отсутствуют готовые решения, связывающие количественную оценку риска с параметрами SSH-сертификатов через стандартные механизмы операционной системы Linux.

Постановка задачи

Целью исследования является повышение безопасности Linux-инфраструктуры за счёт разработки методов динамического управления привилегированным доступом на основе оценки рисков. Под динамическим управлением понимается адаптивное определение параметров каждого сеанса доступа – требуе-

мого уровня верификации, срока действия удостоверяющих данных и детальности аудита – в зависимости от контекста запроса.

Для достижения цели необходимо решить следующие задачи: разработать математическую модель многофакторной агрегации рисков, обеспечивающую свойство вето – невозможность компенсации критического фактора; разработать метод динамического определения параметров сеанса доступа на основе вычисленной оценки; обеспечить интеграцию метода с подсистемой РАМ операционной системы Linux; провести экспериментальное сравнение с альтернативными подходами к агрегации.

Новизна предлагаемого решения заключается в применении мультипликативной модели агрегации со свойством вето для задачи контроля привилегированного доступа и её интеграции с подсистемой РАМ через стандартные SSH-сертификаты, что обеспечивает практическую реализуемость без модификации существующей инфраструктуры.

Предлагаемое решение

Модель агрегации рисков. Для каждого запроса привилегированного доступа система вычисляет интегральную оценку риска $R \in [0, 1]$ на основе пяти контекстных факторов, каждый из которых нормализуется в оценку $r_i \in [0, 1]$. Агрегация выполняется по мультипликативной формуле:

$$R=1 - \prod_{i=1}^d (1 - r_i)^{w_i} \quad (1)$$

где $w_i > 0$ – вес i -го фактора. Принципиальное отличие от взвешенной суммы состоит в свойстве вето: если хотя бы один $r_i = 1$, то $R = 1$ независимо от остальных факторов. Запрос с критической ролью не может быть «компенсирован» низкими значениями остальных факторов, что соответствует требованиям модели безопасности.

Контекстные факторы сгруппированы по категориям NIST SP 800-162 [11]: субъект (r_1 – агрегированная оценка аномальности прошлых сессий с экспоненциальным затуханием), объект (r_2 – максимальная критичность запрашиваемых ролей), окружение (r_3 – принадлежность IP-адреса к зоне доверия; r_4 – типичность геолокации; r_5 – принадлежность времени к рабочему периоду). Выбор факторов обоснован экспериментально: на датасете RBA (31 млн записей) [12] подсеть и страна источника показали наибольшую взаимную информацию с меткой атаки ($I = 0,127$ и $I = 0,094$), а на датасете LANL Cyber1 (1,05 млрд событий) подтверждена информативность временного фактора в корпоративной среде.

Адаптивные параметры сеанса. На основе R определяются четыре параметра выдаваемого SSH-сертификата. Срок действия задаётся экспоненциальной убывающей функцией:

$$T(R)=T_{min}+(T_{max} - T_{min}) \cdot e^{-k \cdot R} \quad (2)$$

где $T_{\min} = 1$ ч, $T_{\max} = 12$ ч, $k = 3$. Уровень верификации определяется ступенчатой функцией с двумя порогами: $\theta_1 = 0,10$ (пароль + ОТР $\rightarrow$ аппаратный ключ) и $\theta_2 = 0,65$ (аппаратный ключ $\rightarrow$ аппаратный ключ с одобрением администратора). Режим записи сессии и уведомление администратора определяются аналогичными пороговыми функциями.

Интеграция с операционной системой. Прототип состоит из трёх компонентов: сервер сертификатов, модуль аутентификации и клиент командной строки. Модуль аутентификации вызывается подсистемой PAM при каждом SSH-подключении. Адаптивные параметры (срок действия, режим записи, уведомление) передаются через поле расширений стандартного сертификата OpenSSH, что обеспечивает совместимость с существующей инфраструктурой без модификации серверного программного обеспечения.

Экспериментальная методика

Для обоснования выбора мультипликативной модели проведено сравнение с двумя альтернативами: взвешенной суммой с логистической функцией и двухуровневой агрегацией по категориям NIST. Веса каждого метода определены из данных методом лог-линеаризации. Оценка выполнена на тестовой выборке из 400 тыс. записей датасета RBA по метрике AUC-ROC; статистическая значимость различий проверена тестом Делонга.

Прототип валидирован на виртуальном стенде с реальной операционной системой. Проведены три серии экспериментов: оценка адаптивности срока действия сертификата для 500 сессий с различным профилем риска; нагрузочное тестирование при 1, 10 и 50 параллельных клиентах; 7-дневная симуляция с 50 пользователями трёх поведенческих профилей (нормальный, необычный, вредоносный). Вредоносные пользователи генерировали аномальные оценки сессий, поступающие в фактор r_1 через механизм обратной связи.

Результаты

Сравнение методов агрегации. Результаты сравнения трёх методов на датасете RBA представлены в табл. 1. Все три метода показали сопоставимые значения AUC-ROC; различие между взвешенной суммой и мультипликативной моделью составило 0,002 (тест Делонга: $p = 0,002$). Мультипликативная модель выбрана как единственная, обладающая свойством вето.

Таблица 1. Сравнение методов агрегации рисков

Метод	AUC-ROC	Recall	Свойство вето
Взвешенная сумма	0,885	0,894	Нет
Мультипликативная	0,883	0,901	Да
Двухуровневая	0,881	0,894	Нет

Адаптивность срока действия. В табл. 2 показано, как предложенный метод дифференцирует срок действия сертификата в зависимости от профиля риска. При статической политике все пользователи получают одинаковый срок (ти-

пично 12–24 ч); предложенный метод сокращает его от 5,78 ч для низкого риска до 1,55 ч для критического. Среднее сокращение относительно фиксированного срока 12 ч составляет 64%.

Таблица 2. Зависимость срока действия сертификата от уровня риска

Профиль запроса	Средний R	Срок действия, ч
Низкий риск (непривилегированная роль, доверенная подсеть)	0,28	5,78
Умеренный риск (привилегии на промежуточную среду)	0,44	3,48
Высокий риск (привилегированная роль, внешняя подсеть)	0,76	2,05
Критический риск (аномальная история, внешняя подсеть)	0,99	1,55

Нагрузочное тестирование (табл. 3) показало, что латентность принятия решения остаётся приемлемой даже при значительной параллельной нагрузке: при 50 одновременных клиентах 99-й перцентиль составляет 95 мс при пропускной способности 1210 запросов в секунду.

Таблица 3. Латентность принятия решения

Параллельных клиентов	p_{99} , мс	Запросов/с
1	9	129
10	22	647
50	95	1210

Симуляция. Результаты 7-дневной симуляции (табл. 4) демонстрируют, что система корректно дифференцирует три класса пользователей исключительно на основе контекстных факторов. Нормальные пользователи получают минимальные ограничения, вредоносные – максимальные. Особенно показателен механизм обратной связи: средний R вредоносных пользователей вырос с 0,989 в первый день до 1,000 к третьему дню за счёт накопления аномальных оценок в факторе r_1 , что свидетельствует об адаптации системы к устойчивым угрозам.

Таблица 4. Результаты 7-дневной симуляции

Профиль	Средний R	Верификация	Срок, ч	Запись, %
Нормальный (n=840)	0,278	100% L1	5,78	0
Необычный (n=98)	0,642	67% L1, 33% L2	2,70	67
Вредоносный (n=84)	0,998	100% L2	1,55	100

Обсуждение

Предложенный метод отличается от существующих подходов к риск-ориентированному контролю доступа [7, 8] в двух аспектах. Во-первых, мультипликативная агрегация обеспечивает свойство вето, отсутствующее у методов на основе взвешенной суммы [7]. Сравнительный эксперимент на 31 млн записей подтвердил, что это свойство достигается без потери качества классификации (AUC-ROC 0,883 против 0,885). Во-вторых, метод интегрирован с подсистемой РАМ через стандартные SSH-сертификаты, что отличает его от теоретических моделей [8, 10], не имеющих практической реализации.

Результаты симуляции показывают, что доминирование фактора критичности ролей обеспечивает чёткое разделение классов пользователей, что соответствует принципу наименьших привилегий [3] и рекомендациям по интеллектуально-адаптивному управлению инфраструктурой [6]. Механизм обратной связи через r_1 позволяет системе адаптироваться к устойчивым угрозам без переобучения модели.

Ограничением метода является зависимость качества от экспертного назначения значений факторов. В дальнейших исследованиях планируется автоматическое определение значений из данных, а также интеграция нейросетевой модели обнаружения аномалий в последовательностях привилегированных команд.

Заключение

В работе предложен метод адаптивного контроля привилегированного доступа, основанный на мультипликативной агрегации пяти контекстных факторов риска. Метод обеспечивает свойство вето, отсутствующее у альтернативных подходов на основе взвешенной суммы, при сопоставимом качестве классификации (AUC-ROC 0,883 против 0,885 на датасете из 31 млн записей). На основе интегральной оценки риска система динамически определяет срок действия SSH-сертификата, уровень верификации и режим записи сессии. Реализованный прототип интегрирован с подсистемой РАМ Linux и валидирован на виртуальном стенде: показано сокращение срока действия удостоверяющих данных на 64% относительно статической политики, а 7-дневная симуляция подтвердила адаптацию системы к изменениям поведенческого профиля пользователей.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Verizon. 2024 Data Breach Investigations Report [Электронный ресурс]. – URL: <https://www.verizon.com/business/resources/reports/dbir/> (дата обращения: 15.04.2026).
2. IBM. X-Force Threat Intelligence Index 2024 [Электронный ресурс]. – URL: <https://www.ibm.com/reports/threat-intelligence> (дата обращения: 15.04.2026).
3. Rose S., Borchert O., Mitchell S., Connelly S. Zero Trust Architecture : NIST SP 800-207. – National Institute of Standards and Technology, 2020. – 59 p.
4. Басыня Е. А. Интеллектуальная поддержка при принятии технических решений в информационной инфраструктуре предприятия // Защита информации. Инсайд. – 2020. – № 4. – С. 68–73.
5. Басыня Е. А., Худяков Д. С. Комплексный мониторинг информационной инфраструктуры предприятия // Защита информации. Инсайд. – 2020. – № 6. – С. 28–33.

6. Басыня Е. А. Комплексная методология интеллектуально-адаптивного управления информационной инфраструктурой предприятия // Защита информации. Инсайд. – 2021. – № 5 (101). – С. 16–25.
7. Kandala S., Sandhu R., Bhamidipati V. An Attribute Based Framework for Risk-Adaptive Access Control Models // Proceedings of the 6th International Conference on Availability, Reliability and Security. – IEEE, 2011. – P. 236–241.
8. Chen L., Crampton J. Risk-Aware Role-Based Access Control // Proceedings of the 7th International Workshop on Security and Trust Management. – Springer, 2012. – P. 140–156.
9. Atlam H. F., Wills G. B. IoT Security, Privacy, Safety and Ethics // Digital Twin Technologies and Smart Cities. – Springer, 2020. – P. 123–149.
10. Магомедов Ш. Г. Модели и методы интеллектуально-адаптивного управления доступом в образовательных вычислительных сервисах : дис. ... д-ра техн. наук. – М., 2024. – 312 с.
11. Hu V. C. et al. Guide to Attribute Based Access Control (ABAC) Definition and Considerations : NIST SP 800-162. – National Institute of Standards and Technology, 2014. – 54 p.
12. Wiefeling S., Lo Iacono L., Dürmuth M. Is This Really You? An Empirical Study on Risk-Based Authentication Applied in the Wild // IFIP Annual Conference on Data and Applications Security and Privacy. – Springer, 2019. – P. 134–152.
13. Syed N. F. et al. Zero Trust Architecture (ZTA): A Comprehensive Survey // IEEE Access. – 2022. – Vol. 10. – P. 57143–57179.
14. Ramezanpour K., Jagannath J. Intelligent Zero Trust Architecture for 5G/6G Networks // IEEE Communications Standards Magazine. – 2022. – Vol. 6. – No. 4. – P. 60–67.
15. Xiao S., Ye Y., Kanwal N., Neue T., Lee B. SoK: Context and Risk Aware Access Control for Zero Trust Systems // Security and Communication Networks. – 2022. – Vol. 2022. – P. 1–14.

© Д. А. Остриков, 2026

Н. М. Острянин¹✉, Е. Ю. Воронкин¹

Разработка веб-сервиса для автоматизации и анализа продаж на основе ABC и XYZ-анализа для малого бизнеса

¹Сибирский государственный университет геосистем и технологий,
г. Новосибирск, Российская Федерация
e-mail: nikitaostryanin2018@gmail.com

Аннотация. В статье рассматривается проблема автоматизации товарного учёта и аналитики продаж на малых предприятиях, лишённых доступа к дорогостоящим ERP-системам. Выявлен «информационный разрыв» между оперативным вводом сделок и формированием управленческой отчётности: традиционные электронные таблицы не обеспечивают ни ссылочной целостности данных, ни автоматического пересчёта аналитических матриц. Предложен архитектурный подход на базе стека React, Node.js и PostgreSQL, реализующий одностраничное клиентское приложение (SPA) с REST API-сервером и реляционным хранилищем. Разработана платформа Dashly, объединяющая модули регистрации транзакций, импорта исторических данных из CSV и динамического расчёта матриц ABC/XYZ. ABC-классификация распределяет номенклатуру по вкладу в совокупную выручку (пропорция 80/15/5 %), XYZ-анализ оценивает стабильность спроса через коэффициент вариации (пороговые значения 10 % и 25 %). Описаны алгоритмы серверной агрегации данных PostgreSQL, типобезопасного доступа через Drizzle ORM и валидации входящих запросов посредством библиотеки Zod. Результаты апробации подтверждают высокую эффективность предложенного решения: пиковая загрузка процессора не превысила 1 %, потребление памяти стабилизировалось на уровне 500 МБ, а время формирования аналитического отчёта сократилось с нескольких часов ручного труда до долей секунды. Разработанные методы могут быть масштабированы для внедрения предиктивной аналитики и мультиканальных интеграций.

Ключевые слова: веб-сервис, автоматизация учёта, ABC-анализ, XYZ-анализ, React, Node.js, PostgreSQL, REST API, управление ассортиментом

N. M. Ostryanin¹✉, E. Yu. Voronkin¹

Development of a web service for sales automation and analysis based on ABC and XYZ analysis for small business

¹Siberian State University of Geosystems and Technologies, Novosibirsk, Russian Federation
e-mail: nikitaostryanin2018@gmail.com

Abstract. The article deals with the problem of automating inventory management and sales analytics in small enterprises that lack access to expensive ERP systems. An 'information gap' was identified between operational transaction entry and management reporting: traditional spreadsheets provide neither referential data integrity nor automatic recalculation of analytical matrices. An architectural approach based on the React, Node.js, and PostgreSQL stack is proposed, implementing a single-page client application (SPA) with a REST API server and a relational database. The Dashly platform was developed, combining modules for transaction registration, historical data import from CSV files, and dynamic calculation of ABC/XYZ matrices. ABC classification distributes the product range by contribution to total revenue (80/15/5% ratio), while XYZ analysis evaluates demand stability through the coefficient of variation (threshold values of 10% and 25%). The algorithms of server-side data aggregation in PostgreSQL, type-safe access through Drizzle ORM, and validation of incoming

requests using the Zod library are described. The test results confirm the high efficiency of the proposed solution: peak CPU load did not exceed 1%, memory consumption stabilized at 500 MB, and the time to generate an analytical report was reduced from several hours of manual labor to fractions of a second. The developed methods can be scaled to implement predictive analytics and multichannel integrations.

Keywords: web service, accounting automation, ABC analysis, XYZ analysis, React, Node.js, PostgreSQL, REST API, assortment management

Введение

Слабая интеграция между системами повседневного учёта и модулями глубокой аналитики является существенным препятствием для эффективного функционирования малых предприятий. Зачастую сотрудники вынуждены фиксировать сделки в децентрализованных электронных таблицах или на бумажных носителях, а формирование управленческой отчётности происходит обособленно. Товарный анализ либо полностью игнорируется, либо требует значительных ручных трудозатрат [1].

Отсутствие единого информационного пространства провоцирует расхождения в данных: предпринимателю крайне сложно определить, какие позиции приносят реальный доход, а какие «замораживают» оборотные средства в избыточных запасах. При расширении ассортимента в базах данных накапливаются дублирующие записи, а вычисление метрик нестабильности спроса (XYZ-анализ) становится слишком ресурсоёмким, поскольку любое изменение первичных данных требует повторного ручного пересчёта [2].

Целью данного исследования является разработка и практическая реализация специализированного веб-сервиса Dashly, адаптированного под потребности малого бизнеса и обеспечивающего единый контур регистрации продаж, ведения справочников и автоматического построения аналитических матриц ABC/XYZ без участия аналитика.

Методы и материалы

Для реализации поставленной цели применялись методы системного анализа предметной области, реляционного проектирования баз данных и объектно-ориентированного программирования. Технологической основой решения выбран стек, широко применяемый в современной веб-разработке: библиотека React [3] для клиентского интерфейса, среда Node.js с фреймворком Express [4] для серверной логики и СУБД PostgreSQL [5] для надёжного хранения данных [6].

Архитектура системы строго подчинена стандартам разработки современных веб-платформ: сочетание одностраничного клиентского приложения (SPA) и серверного REST API [7]. Клиентская часть перехватывает действия пользователя и формирует сетевые запросы к бэкенд-узлу, который выступает единственным доверенным посредником при взаимодействии с базой данных через Drizzle ORM [8].

Фундаментом сервиса служит реляционная схема базы данных. Ключевыми таблицами выступают справочник номенклатуры (products) и реестр транзакций (sales). Связи между сущностями строятся через внешние ключи, что гаранти-

рует ссылочную целостность на уровне СУБД. Drizzle ORM транслирует TypeScript-описания таблиц непосредственно в DDL-команды PostgreSQL, исключая ручное написание миграций [9].

Событийная модель обмена между клиентом и сервером опирается на классический цикл «запрос – ответ» протокола HTTP. Браузер формирует REST-запрос к конкретному эндпоинту – например, POST /api/sales для регистрации продажи или GET /api/analytics/abc-xyz для получения матрицы. Сервер валидирует входящие данные посредством схем Zod [10] и возвращает ответ в формате JSON [11].

ABC-классификация опирается на принцип Парето: товары ранжируются по убыванию вклада в совокупную выручку, затем вычисляется накопительная доля. Группа А формирует первые 80 % результата, В – следующие 15 %, С – оставшиеся 5 % [12]. XYZ-анализ классифицирует позиции по стабильности спроса: для каждого товара рассчитывается коэффициент вариации ($CV = \sigma / \mu \times 100 \%$). При $CV \leq 10 \%$ товар относится к классу X (стабильный спрос), при $10 \% < CV \leq 25 \%$ – к Y (умеренные колебания), при $CV > 25 \%$ – к Z (нерегулярный спрос) [13].

Комбинированная матрица ABC/XYZ формирует девять пересекающихся зон: от стратегически важных AX до проблемных CZ. Вся математика выполняется на стороне сервера, что полностью исключает зависимость от вычислительных возможностей клиентского устройства. Для оценки эффективности предложенного решения применялось нагрузочное тестирование на наборах от 100 до 10 000 записей о продажах.

Сравнение предложенного подхода с существующими альтернативами проведено по критериям стоимости владения, скорости развёртывания, масштабируемости и степени интерактивности (табл. 1).

Таблица 1. Сравнение основных подходов к организации учёта продаж

Технология	Достоинства	Недостатки
Электронные таблицы (Excel)	Нет затрат на сервер, простота освоения	Высокий процент ошибок, медленная работа с большими массивами
Монолитные ERP-системы	Максимальная функциональность и надёжность	Высокая стоимость лицензий и внедрения
Веб-сервис (React / Node.js)	Быстрый старт, интерактивность, масштабируемость	Требует грамотного проектирования API и архитектуры

Результаты

Практическая реализация разработанной архитектуры позволила создать полнофункциональный веб-сервис Dashly, охватывающий весь контур от первичного ввода данных до визуализации аналитической матрицы. При поступлении транзакции серверный модуль Node.js выполняет валидацию входных данных, сохраняет запись в PostgreSQL через Drizzle ORM и инвалидирует клиент-

ский кэш React Query, обеспечивая актуальность аналитических представлений без ручного обновления страницы.

Маршрутизация запросов реализована следующим образом. Обращение к эндпоинту GET /api/analytics/abc инициирует на сервере группирующий SQL-запрос для суммирования выручки по каждому идентификатору товара, сортировку по убыванию финансового вклада, вычисление накопительной доли и разбиение по категориям согласно пропорции 80–15–5. Запрос GET /api/analytics/xyz дополнительно группирует продажи по месячным интервалам, рассчитывает среднее значение и стандартное отклонение для каждого товара и определяет коэффициент вариации. Итоговая матрица объединяет оба фильтра и возвращается клиенту в формате JSON [11].

Функция массового импорта обеспечивает загрузку исторических архивов в формате CSV. Алгоритм осуществляет построчное чтение файла с потоковой обработкой (stream-based parsing), верифицирует названия колонок и проверяет корректность сумм и форматов дат посредством схем Zod. Некорректные строки отклоняются с формированием подробного отчёта об ошибках. Валидные записи сохраняются в PostgreSQL в рамках единой транзакции: если хотя бы одна строка не проходит валидацию, откатывается весь пакет, предотвращая частичную загрузку данных.

Клиентский интерфейс матрицы ABC/XYZ реализован в виде интерактивной сетки 3×3, где каждая ячейка окрашивается в зависимости от статуса: стратегически важные товары AX выделяются зелёным цветом, проблемные CZ – красным. Пользователь взаимодействует с интуитивно понятным графиком, при этом данные запрашиваются через REST API и кэшируются посредством React Query, что соответствует рекомендациям по проектированию отзывчивых интерфейсов.

Комплексное тестирование охватило все подсистемы платформы. Строгой проверке подверглась надёжность защиты REST-маршрутов: неавторизованные запросы к приватным эндпоинтам отклоняются с кодом 401, сессионная кука корректно выдаётся при успешной аутентификации. Результаты основных сценариев тестирования приведены в табл. 2.

Таблица 2. Основные сценарии тестирования платформы Dashly

Проверяемый сценарий	Ожидаемый результат	Фактический результат
Аутентификация и защита маршрутов	Выдача сессионной куки, блокировка без неё	Выполнено
Импорт повреждённого CSV	Валидатор Zod отклоняет файл с сообщением об ошибке	Выполнено
Алгоритм ABC-классификации	Корректное ранжирование по пропорции 80/15/5 %	Выполнено
Алгоритм XYZ-классификации	Верный расчёт коэффициента вариации (CV)	Выполнено

Нагрузочное тестирование продемонстрировало высокую эффективность выбранной архитектуры. Пиковые показатели загрузки центрального процессора Node.js не превысили 1 %, а расход оперативной памяти стабилизировался на уровне 500 МБ. Данный факт свидетельствует о том, что веб-сервис готов к промышленной эксплуатации даже на бюджетных виртуальных серверах (VPS) с минимальными тарифными планами, что критически важно для малого бизнеса с ограниченным ИТ-бюджетом.

Результаты сравнительного анализа эффективности до и после внедрения платформы Dashly приведены в табл. 3.

Таблица 3. Сравнительный анализ показателей до и после внедрения веб-сервиса

Параметр оценки	До внедрения Dashly	После внедрения Dashly
Формирование ABC/XYZ-отчётов	Многочасовой ручной расчёт в таблицах	Доли секунды в браузере (автоматически)
Загрузка исторических данных	Долгое ручное копирование строк	Моментальный импорт CSV с валидацией
Вероятность ошибки ввода	Критически высокая	Сведена к минимуму (валидация Zod)
Трудозатраты аналитика	Высокие	Нулевые на этапе расчёта матриц
Перспективы масштабирования	Практически отсутствуют	Неограниченные благодаря модульной архитектуре

Обсуждение

Полученные в ходе тестирования данные подтверждают состоятельность выбранной архитектуры. Сравнение предложенного метода с монолитными ERP-системами и электронными таблицами выявило существенное преимущество в части доступности и скорости развёртывания. Отказ от тяжёловесных монолитных систем в пользу легковесного REST API с реляционной СУБД снизил порог входа для малого бизнеса, сохранив при этом возможности промышленных аналитических систем [14, 15].

Проблема обработки некорректных данных при импорте CSV была решена внедрением строгой валидации на уровне схем Zod и транзакционной модели PostgreSQL, что соответствует лучшим практикам разработки надёжных серверных приложений [16]. Переход на потоковую обработку CSV-файлов (stream-based parsing) позволил полностью исключить блокировку основного потока выполнения Node.js при загрузке объёмных архивов.

Основным ограничением предложенного метода является необходимость первоначальной настройки серверной инфраструктуры (VPS, PostgreSQL, переменные окружения). Однако затраты на развёртываниекратно ниже стоимости лицензий и поддержки корпоративных ERP-решений, а модульная архитектура

обеспечивает горизонтальное масштабирование без переработки ядра системы [17].

Заключение

В рамках проведённого исследования разработана, протестирована и введена в эксплуатацию платформа Dashly – специализированный веб-сервис для автоматизации учёта продаж и расчёта аналитических матриц ABC/XYZ для малого бизнеса. Успешно решена проблема «информационного разрыва» между оперативным вводом транзакций и формированием управленческой аналитики. Разработанный серверный модуль на Node.js обеспечил сквозную автоматизацию расчёта классификационных матриц, полностью исключив ручной труд аналитика.

Результаты нагрузочного тестирования подтверждают многократное повышение эффективности: время формирования ABC/XYZ-отчёта сокращено с нескольких часов до долей секунды, пиковая загрузка CPU не превысила 1 %, что позволяет эксплуатировать сервис на бюджетном VPS. Реализованная валидация входящих данных через Zod и транзакционная модель PostgreSQL обеспечивают целостность данных при массовом импорте CSV-архивов.

Дальнейшие исследования будут направлены на расширение пула аналитических моделей: внедрение RFM-анализа для сегментации клиентской базы, создание механизма автоматических уведомлений при критическом смещении товара между группами матрицы, а также интеграцию алгоритмов машинного обучения для предиктивной аналитики спроса.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. СТО СМК СГУГиТ 8.5–39–2025. Государственная итоговая аттестация выпускников СГУГиТ. Структура и правила оформления. – Новосибирск : СГУГиТ, 2025. – 67 с.
2. Панина У. Е., Попова Е. В. ABC/XYZ-анализ как способ анализа товарооборота розничного предприятия. – URL: <https://cyberleninka.ru/article/n/abc-xyz-analiz-kak-sposob-analiza-tovarooborota-rozничного-predpriyatiya> (дата обращения: 25.05.2026).
3. Документация React. – URL: <https://react.dev/> (дата обращения: 15.04.2026).
4. Документация Express.js. – URL: <https://expressjs.com/> (дата обращения: 15.04.2026).
5. Руководство по PostgreSQL. – URL: <https://www.postgresql.org/docs/> (дата обращения: 15.04.2026).
6. Фрэнкс Б. Укрощение больших данных: как извлекать знания из массивов информации с помощью глубокой аналитики. – Москва : Манн, Иванов и Фербер, 2014.
7. Филдинг Р. Архитектурные стили и дизайн сетевых программных архитектур / Р. Филдинг // Information and Computer Science: сайт. – URL: https://ics.uci.edu/~fielding/pubs/dissertation/rest_arch_style.htm (дата обращения: 15.04.2026). – Текст : электронный.
8. Документация Drizzle ORM. – URL: <https://orm.drizzle.team/> (дата обращения: 15.04.2026).
9. Документация TypeScript. – URL: <https://www.typescriptlang.org/> (дата обращения: 15.04.2026).
10. Документация Zod. – URL: <https://zod.dev/> (дата обращения: 15.04.2026).

11. Брей Т. Формат обмена данными JavaScript Object Notation (JSON). RFC 8259 / Т. Брей // RFC Editor : сайт. – URL: <https://www.rfc-editor.org/rfc/rfc8259> (дата обращения: 15.04.2026).
12. Бузмакова Е. Е. Методы классификации запасов предприятия // КиберЛенинка. – URL: <https://cyberleninka.ru/article/n/metody-klassifikatsii-zapasov-predpriyatiya> (дата обращения: 25.05.2026).
13. Деревянко А. А., Рябова Т. В. Сущность и направления использования ABC-анализа // КиберЛенинка. – URL: <https://cyberleninka.ru/article/n/suschnost-i-napravleniya-ispolzovaniya-avs-analiza> (дата обращения: 25.05.2026).
14. Фаулер М. Шаблоны корпоративных приложений / М. Фаулер. – Москва : Вильямс, 2016.
15. Ричардсон К. Микросервисы. Паттерны разработки и рефакторинга / К. Ричардсон. – Санкт-Петербург : Питер, 2019.
16. Киченко Л. П., Шалимов В. Совершенствование системы управления запасами на предприятиях оптовой торговли // КиберЛенинка. – URL: <https://cyberleninka.ru/article/n/sovershenstvovanie-sistemy-upravleniya-zapasami-na-predpriyatiyah-optovoy-torgovli> (дата обращения: 25.05.2026).
17. Клеппман М. Высоконагруженные приложения. Программирование, масштабирование, поддержка / М. Клеппман. – Санкт-Петербург : Питер, 2018. – 640 с.
18. Durski K., Murlewski J., Makowski D., Sakowicz B. Warehouse Management System in Ruby on Rails Framework on Cloud Computing Architecture // Радиоэлектроника и информатика. – 2011. – № 4. – URL: <https://cyberleninka.ru/article/n/warehouse-management-system-in-ruby-on-rails-framework-on-cloud-computing-architecture> (дата обращения: 25.05.2026).
19. Документация Node.js. – URL: <https://nodejs.org/ru/docs/> (дата обращения: 15.04.2026).
20. Официальная документация Tailwind CSS. – URL: <https://tailwindcss.com/docs> (дата обращения: 15.04.2026).

© Н. М. Острянин, Е. Ю. Воронкин, 2026

А. С. Панов¹✉, С. Ю. Кацко¹

Разработка веб-приложения для управления личными задачами с системой анализа и рекомендаций по повышению продуктивности

¹Сибирский государственный университет геосистем и технологий,
г. Новосибирск, Российская Федерация
e-mail: vip.zip.rar@mail.ru

Аннотация. В статье представлены результаты создания веб-приложения для персонального управления задачами, включающего модуль статистики и генерации рекомендаций. Проведен анализ существующих методик (GTD, Pomodoro, матрица Эйзенхауэра) и программных аналогов (Todoist, TickTick, Microsoft To Do), выявлен дефицит полностью бесплатных локальных решений с аналитикой. Предложена модульная архитектура на чистом JavaScript с хранением данных в LocalStorage. Реализованы CRUD-операции, фильтрация, сортировка, визуализация динамики (Chart.js) и продукционная модель рекомендаций (4 правила: низкий процент выполнения, просроченные высокоприоритетные задачи, вечерняя активность, отставание по категориям). Проведено тестирование (Jest, ручное, Lighthouse): производительность 98 баллов, лучшие практики 100. Приложение не требует регистрации, работает полностью на стороне клиента.

Ключевые слова: веб-приложение, управление задачами, продуктивность, LocalStorage, JavaScript, продукционная модель, визуализация данных, тестирование

A. S. Panov¹✉, S. Yu. Katsko¹

Development of a web application for managing personal tasks with a system of analysis and recommendations for increasing productivity

¹Siberian State University of Geosystems and Technologies, Novosibirsk, Russian Federation
e-mail: vip.zip.rar@mail.ru

Abstract. The article presents the results of developing a web application for personal task management that includes a statistics module and a recommendation generator. An analysis of existing methodologies (GTD, Pomodoro, Eisenhower matrix) and software analogues (Todoist, TickTick, Microsoft To Do) was carried out, revealing a shortage of completely free local solutions with analytics. A modular architecture using pure JavaScript with LocalStorage data storage is proposed. CRUD operations, filtering, sorting, dynamics visualisation (Chart.js) and a production rule-based recommendation model (4 rules: low completion rate, overdue high-priority tasks, evening activity, lagging categories) are implemented. Testing (Jest, manual, Lighthouse) shows performance of 98 points and best practices of 100. The application requires no registration and runs entirely on the client side.

Keywords: web application, task management, productivity, LocalStorage, JavaScript, production model, data visualisation, testing

Введение

В условиях информационной перегрузки и многозадачности управление личными делами становится критически важным. Бумажные списки уступают

место цифровым инструментам, однако многие из них либо требуют платной подписки для доступа к аналитике, либо хранят данные на серверах, что вызывает опасения о конфиденциальности. Существует потребность в бесплатном, полностью локальном инструменте, сочетающем планирование задач с обратной связью по продуктивности и персонализированными советами.

Цель работы – разработка веб-приложения для управления личными задачами с модулем анализа и системой рекомендаций. Задачи: анализ предметной области и аналогов, проектирование архитектуры и пользовательского интерфейса, реализация на JavaScript с LocalStorage, алгоритмизация рекомендаций на производственных правилах, тестирование.

Исследование предметной области

Исследования показывают, что методики GTD (сбор, обработка, организация, обзор, выполнение) [1], Pomodoro (25-минутные интервалы) [2] и матрица Эйзенхауэра [3] могут быть эффективно поддержаны программными средствами. Из GTD вытекают требования к единому хранилищу задач, категоризации и инструментам обзора (фильтрация, сортировка). Метод Pomodoro обосновывает включение рекомендации по интервальной работе при низком проценте выполнения. Матрица Эйзенхауэра в упрощенной трехуровневой проекции (низкий, средний, высокий приоритет) служит основой для визуальной индикации и анализа просроченных задач.

Проведенный обзор существующих решений Todoist, TickTick, Microsoft To Do, Trello, Яндекс Трекер [4–8] выявил, что аналитические функции либо платные, либо отсутствуют, а хранение данных везде серверное с обязательной регистрацией. Ни один из продуктов не предоставляет персонализированных рекомендаций на основе поведения пользователя. Это обосновывает актуальность разработки полностью локального бесплатного приложения с аналитикой и рекомендательным модулем.

Методы и технологии

При разработке применялась клиентская архитектура (Single Page Application) без серверной части. Основной язык – JavaScript (ES6+) [9], среда разработки – Visual Studio Code [10], тестирование – Jest [11] и Chrome DevTools [12], визуализация данных – библиотека Chart.js.

Хранение данных реализовано через LocalStorage. Сериализация в JSON, ключ «tasks». Структура задачи: id, title, description, category, priority (low/medium/high), dueDate, isCompleted, createdAt, completedAt. Обрабатывается ситуация переполнения квоты (5 МБ) и недоступности хранилища.

Модульная архитектура включает пять компонентов:

- TaskManager – CRUD, фильтрация, сортировка.
- StorageService – чтение/запись LocalStorage.
- RenderService – динамическое обновление DOM.
- AnalyticsService – расчет процента выполнения, просроченных, распределения по категориям, динамики по дням.

– RecommendationEngine – генерация советов на основе правил.

Правила (пороги обоснованы литературой по тайм-менеджменту):

1. Если процент выполнения за последние 7 дней $< 40\%$ → предложить метод Pomodoro.

2. Если невыполненных задач с высоким приоритетом и просроченным сроком > 2 → пересмотреть сроки или делегировать.

3. Если доля завершённых после 18:00 или до 6:00 $> 60\%$ → планировать важное на утро.

4. Если по какой-либо категории процент выполнения $< 30\%$ (при наличии не менее 3 задач) → уделить внимание этой категории.

Функция generateRecommendations() вызывается после каждого изменения данных, возвращает не более трех советов.

Результаты

Разработано веб-приложение, доступное в браузере без установки. Интерфейс (рис. 1) имеет два режима: «Задачи» и «Статистика». В режиме задач реализованы:

- создание / редактирование / удаление задач через модальное окно;
- визуальная индикация приоритетов (зеленый – низкий, желтый – средний, красный – высокий);
- фильтрация по категории (Работа, Учеба, Дом, Личное) и приоритету;
- сортировка по дате создания, сроку или приоритету;
- отметка выполнения с фиксацией времени.

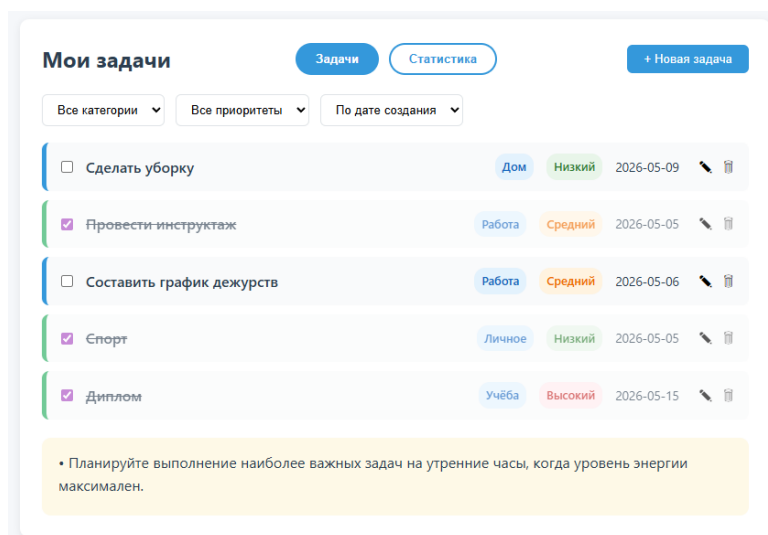


Рис. 1. Главный экран приложения

Режим статистики (рис. 2) отображает:

- общее количество задач, процент выполнения, число просроченных;
- линейный график динамики созданных и выполненных задач по дням (за неделю, месяц или все время);

– круговую диаграмму распределения задач по категориям.

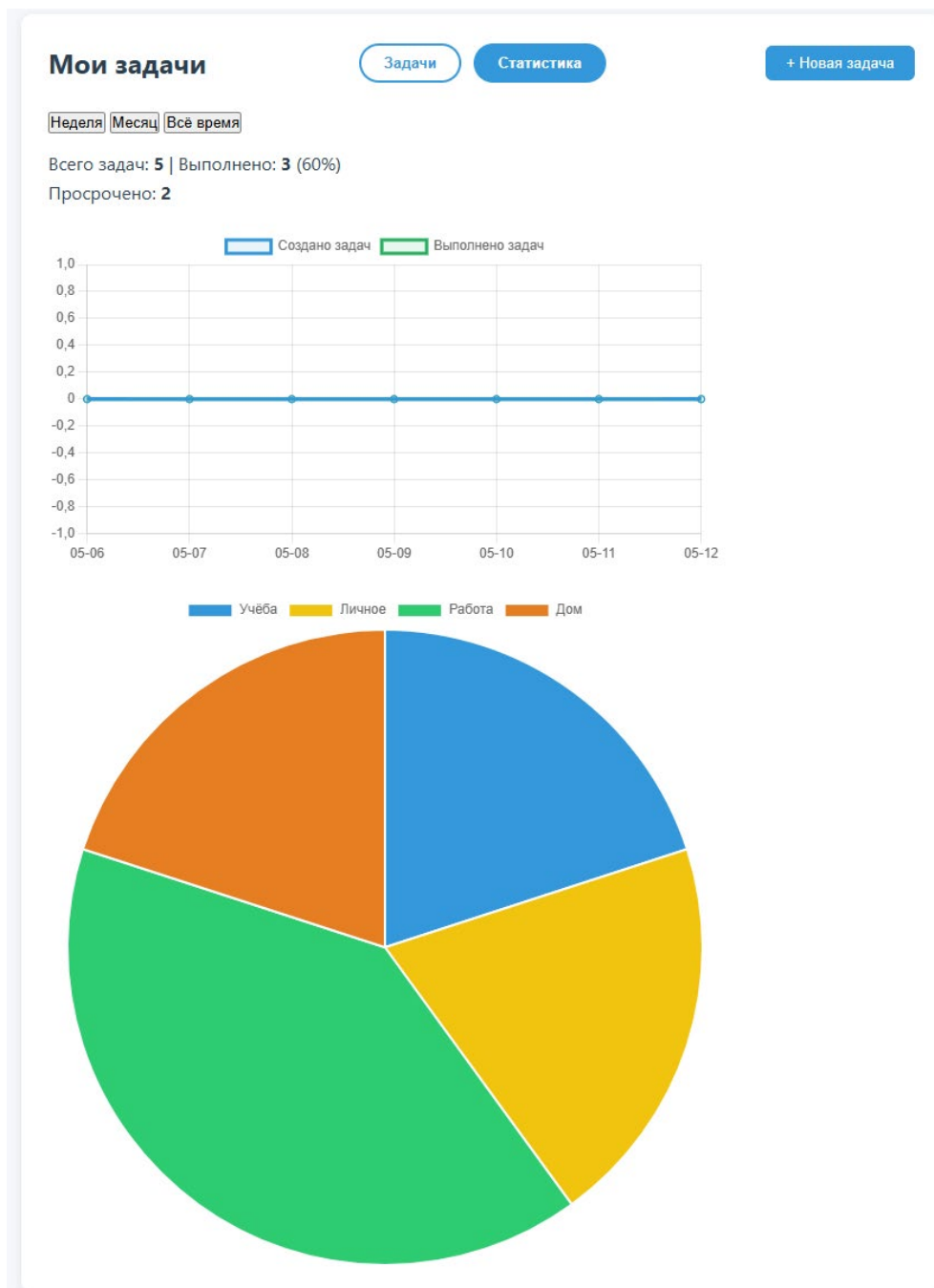


Рис. 2. Страница статистики

Блок рекомендаций динамически обновляется под списком задач. Примеры срабатывания правил: при падении недельной продуктивности ниже 40 % выводится совет использовать Pomodoro; при накоплении трех просроченных высокоприоритетных задач – рекомендация пересмотреть планы.

Модульное тестирование (Jest [11]) подтвердило корректность функций фильтрации, сортировки, расчета статистики и логики рекомендаций. Ручное функциональное тестирование охватило все пользовательские сценарии: создание, ре-

дактирование, удаление, отметку выполнения, сохранность после перезагрузки, адаптивность (320–1920 пикселей) с использованием браузерных инструментов разработчика [12, 13]. Аудит Lighthouse [14] дал следующие результаты: производительность – 98, доступность – 75, лучшие практики – 100, SEO – 90 (рис. 3). Недостатки доступности связаны с отсутствием текстовых меток у символьных кнопок и могут быть устранены в следующих версиях.

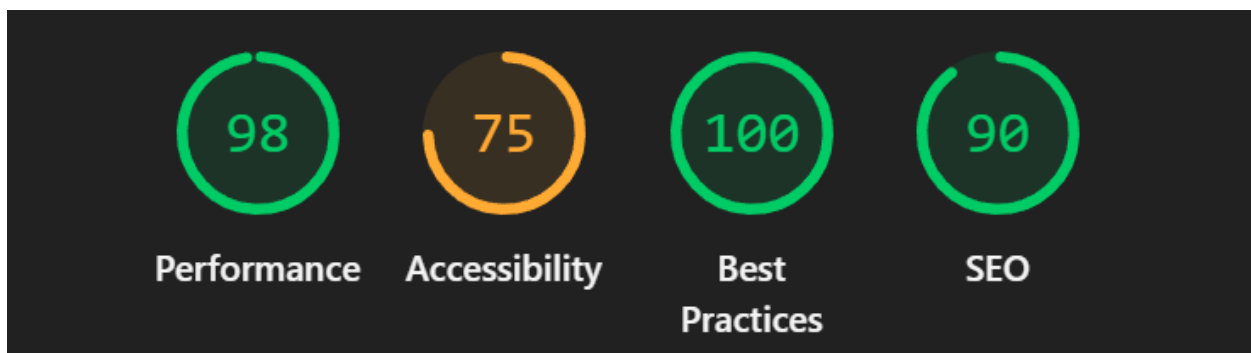


Рис. 3. Результаты Lighthouse

Заключение

В ходе работы разработано клиентское веб-приложение для управления личными задачами с системой анализа и рекомендаций. Основные достижения:

- проведен анализ предметной области и методик продуктивности;
- предложена модульная архитектура на JavaScript с хранилищем LocalStorage;
- реализованы CRUD, фильтрация, сортировка, визуализация статистики (Chart.js);
- создана производственная модель генерации рекомендаций (4 правила);
- выполнено тестирование (Jest, ручное, Lighthouse), подтвердившее высокую производительность и соответствие требованиям.

Приложение может использоваться студентами и офисными сотрудниками для повышения личной эффективности. Перспективы развития: добавление синхронизации между устройствами (серверный слой), расширение набора производственных правил, улучшение доступности интерфейса.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Аллен Д. Как привести дела в порядок: искусство продуктивности без стресса / Д. Аллен ; пер. с англ. Ю. Константиновой. – М. : Манн, Иванов и Фербер, 2015. – 416 с.
2. Чирилло Ф. Метод Pomodoro: управление временем / Ф. Чирилло ; пер. с англ. – М. : Эксмо, 2022. – 224 с.
3. Кови С. 7 навыков высокоэффективных людей / С. Кови ; пер. с англ. – 18 е изд. – М. : Альпина Паблишер, 2023. – 396 с.
4. Todoist: приложение для управления задачами. – URL: <https://todoist.com/ru> (дата обращения: 16.04.2026). – Текст : электронный.
5. TickTick: To Do List & Calendar. – URL: <https://ticktick.com/> (дата обращения: 16.04.2026). – Текст : электронный.

6. Microsoft To Do. – URL: <https://to-do.office.com/> (дата обращения: 16.04.2026). – Текст : электронный.
7. Trello: Управление заданиями. – URL: <https://trello.com/ru/use-cases/task-management> (дата обращения: 16.04.2026). – Текст : электронный.
8. Яндекс Трекер. – URL: <https://tracker.yandex.ru/> (дата обращения: 16.04.2026). – Текст : электронный.
9. Флэнаган Д. JavaScript. Полное руководство / Д. Флэнаган ; пер. с англ. А. А. Слинкина. – 7 е изд. – СПб. : Диалектика, 2021. – 720 с.
10. Visual Studio Code. – URL: <https://code.visualstudio.com/> (дата обращения: 21.04.2026). – Текст : электронный.
11. Jest: Delightful JavaScript Testing Framework. – URL: <https://jestjs.io/> (дата обращения: 21.04.2026). – Текст : электронный.
12. Chrome DevTools: отладка JavaScript // Chrome for Developers. – URL: <https://developer.chrome.google.cn/docs/devtools/javascript> (дата обращения: 21.04.2026). – Текст : электронный.
13. Общие сведения о средствах разработки // Microsoft Edge Developer documentation. – URL: <https://learn.microsoft.com/ru-ru/microsoft-edge/devtools/overview> (дата обращения: 21.04.2026). – Текст : электронный.
14. Lighthouse: Оптимизируйте свой сайт // Chrome for Developers. – URL: <https://developer.chrome.google.cn/docs/devtools/lighthouse> (дата обращения: 21.04.2026). – Текст : электронный.

© А. С. Панов, С. Ю. Кацко, 2026

С. Е. Пелих¹✉, А. А. Шаранов¹

Разработка веб-приложения для поддержки образовательного процесса по робототехнике в центрах дополнительного образования

¹Сибирский государственный университет геосистем и технологий,
г. Новосибирск, Российская Федерация
e-mail: lana.pelikh.sveta.pelikh@mail.ru

Аннотация. В статье представлены основные результаты разработки веб-приложения ArLab для поддержки образовательного процесса по робототехнике в центрах дополнительного образования. Выполнен анализ предметной области, в котором рассмотрены особенности преподавания основ робототехники, а также ограничения существующих программных средств обучения. Для создания веб-приложения были использованы стандартизированные веб-технологии: HTML5, CSS3 и JavaScript. Веб-приложение предоставляет пользователю возможность визуального конструирования электронных схем посредством перетаскивания компонентов, создания соединений между контактами, написания управляющего кода для микроконтроллеров Arduino, а также автоматической проверки корректности собранных схем и программного кода с мгновенной обратной связью.

Ключевые слова: веб-приложение, образовательная платформа, робототехника, программирование микроконтроллеров, визуальный конструктор схем, Arduino

S. E. Pelikh¹✉, A. A. Sharapov¹

Development of a web application to support the educational process in robotics at additional education centers

¹Siberian State University of Geosystems and Technologies, Novosibirsk, Russian Federation
e-mail: lana.pelikh.sveta.pelikh@mail.ru

Abstract. The article presents the main results of the development of the ArLab web application to support the educational process in robotics at additional education centers. The analysis of the subject area is performed, which examines the features of teaching robotics fundamentals to students aged 10–14 years, as well as the limitations of existing educational software. Standardized web technologies were used to create the web application: HTML5, CSS3, and JavaScript. The web application provides the user with the ability to visually construct electronic circuits by dragging and dropping components, creating connections between contacts, writing control code for Arduino microcontrollers, as well as automatically checking the correctness of assembled circuits and program code with instant feedback.

Keywords: web application, educational platform, robotics, microcontroller programming, visual circuit designer, Arduino

Введение

Обучение основам робототехники и программированию микроконтроллеров в учреждениях дополнительного образования представляет собой специфический педагогический процесс, обусловленный возрастными, психологиче-

скими и организационными особенностями целевой аудитории. Основной контингент обучающихся составляют дети и подростки в возрасте от десяти до четырнадцати лет, для которых характерно преобладание наглядно-образного мышления над абстрактно-логическим. В данном возрастном периоде учащиеся демонстрируют высокую восприимчивость к визуальной информации, однако сталкиваются со значительными трудностями при усвоении формализованных программных конструкций и фундаментальных физических законов, описывающих работу электрических цепей.

Традиционная методика преподавания, базирующаяся на непосредственном взаимодействии с физическими учебными наборами на базе микроконтроллеров Arduino, сопряжена с рядом методических и ресурсных ограничений. Практическая работа с аппаратным обеспечением требует существенных материальных затрат, сопровождается высоким риском повреждения оборудования вследствие ошибочных подключений и не обеспечивает мгновенной визуальной обратной связи [1]. Столкновение с первыми техническими неудачами часто провоцирует фрустрацию и снижение внутренней мотивации, из-за чего значительная часть учащихся прекращает регулярные занятия на начальном этапе.

В последние годы стремительное развитие технологий веб-приложений открывает многообещающие возможности для решения этих проблем и позволяет создать безопасную цифровую среду для экспериментирования с электронными схемами и программным кодом без риска повреждения оборудования. Целью данной работы является разработка веб-приложения ArLab, объединяющего визуальный конструктор схем и редактор кода в единой интерактивной среде.

Исследование предметной области

Современный рынок образовательных технологий предлагает ряд программных решений для моделирования электронных схем и написания кода. Платформа Tinkercad Circuits, разработанная компанией Autodesk, представляет собой облачный сервис, предоставляющий возможности визуального конструирования электронных схем. Однако интерфейс характеризуется избыточной функциональностью, не адаптированной для пошагового освоения материала.

Платформа Wokwi представляет собой специализированный онлайн-симулятор для разработки и тестирования проектов на базе Arduino и ESP32, обеспечивающий возможность совместной работы над проектами. Сервис поддерживает загрузку внешних библиотек и эмуляцию сетевых соединений, однако функционирует по подписной модели с ограниченным набором компонентов в бесплатной версии и не предоставляет адаптированных учебных материалов для начинающих [2].

Официальная интегрированная среда разработки Arduino IDE представляет собой профессиональный инструмент, отличающийся строгой синтаксической требовательностью, отсутствием визуальной симуляции электрических цепей и высоким порогом вхождения для начинающих, поскольку сообщения об ошибках компиляции сформулированы в технической терминологии [3].

Существующие программные комплексы не реализуют комплексной интеграции визуального редактора схем и текстового редактора кода в едином рабочем пространстве, не содержат адаптивных механизмов диагностики ошибок с пояснениями, доступными для восприятия подростками, и не учитывают психофизиологические особенности учащихся десяти–четырнадцати лет. Отсутствие специализированных инструментов, соответствующих педагогическим задачам центров дополнительного образования, обуславливает необходимость разработки отечественного веб-приложения, ориентированного на упрощение процесса обучения и снижение когнитивной нагрузки при первом знакомстве с инженерными дисциплинами [4].

Методы и технологии

Разработку веб-приложения ArLab было решено вести с применением стандартизированных веб-технологий: HTML5, CSS3 и JavaScript (ECMAScript 6+). Выбор данной конфигурации обусловлен необходимостью обеспечения кроссплатформенной доступности, исключения требований к установке клиентского программного обеспечения и соответствия современным стандартам W3C [5].

Язык гипертекстовой разметки HTML5 формирует семантическую структуру пользовательского интерфейса, обеспечивая логическое разделение рабочих зон: центральной области сборки схем, библиотеки компонентов, редактора кода и панели управления заданиями.

Каскадные таблицы стилей CSS3 отвечают за визуальное оформление, адаптивную вёрстку и управление состоянием интерфейсных компонентов. Применение гибких сеточных структур (Flexbox) и медиавыражений позволяет сохранять корректное отображение приложения на устройствах с различными разрешениями экранов и плотностью пикселей.

Язык программирования JavaScript выступает в качестве основного инструмента реализации клиентской логики, обработки пользовательских событий, динамического управления DOM-деревом и выполнения алгоритмов проверки электрических соединений. Отказ от применения тяжёлых фреймворков (React, Angular, Vue) в пользу нативного JavaScript продиктован необходимостью обеспечения прозрачности архитектуры, минимизации объёма загружаемых ресурсов и соответствием образовательной задаче демонстрации фундаментальных принципов взаимодействия клиентской части веб-приложения с пользовательским интерфейсом [6].

Для проектирования пользовательского интерфейса была использована среда визуального проектирования Figma, которая позволила сформировать интерактивные прототипы, согласовать дизайн-решения на этапе предпроектной подготовки и провести предварительную оценку юзабилити до начала программной реализации.

Таким образом, для разработки веб-приложения был использован следующий технологический стек:

- HTML5 для семантической разметки интерфейса;
- CSS3 с применением адаптивной вёрстки и CSS-анимаций;

- JavaScript (ES6+) для реализации клиентской логики и алгоритмов валидации;
- Figma для проектирования макетов и прототипирования пользовательского опыта.

Результаты

В результате работы было создано веб-приложение ArLab, в котором реализованы следующие функции:

- ввод переключение между режимом визуального конструирования схем и режимом написания программного кода;
- перетаскивание компонентов (Arduino Uno, светодиоды, резисторы, сервоприводы) из библиотеки в рабочую область посредством обработки событий указательного устройства с расчётом смещения относительно границ контейнера;
- создание электрических соединений между контактными точками компонентов и пинами микроконтроллера с автоматическим расчётом длины и угла наклона проводов с использованием тригонометрических функций; динамическое обновление геометрии соединительных линий при перемещении компонентов в реальном времени без полной перерисовки DOM-дерева;
- ввод управляющего программного кода для микроконтроллеров Arduino в текстовом редакторе с моноширинным шрифтом;
- автоматическая проверка корректности собранной электрической цепи путём сопоставления фактического графа соединений с эталонным шаблоном учебного задания;
- валидация написанного программного кода на наличие обязательных ключевых слов и синтаксических конструкций с генерацией детализированных диагностических сообщений;
- визуальная обратная связь при успешной проверке схемы в виде анимации свечения светодиодов с применением CSS-фильтров и ключевых кадров;
- удаление компонентов и проводов из рабочей области в специальном режиме с подтверждением действия;

Хранение состояния приложения (позиции компонентов, установленные соединения, введённый код) осуществляется в локальных структурах данных на стороне клиента, что обеспечивает мгновенный доступ и модификацию без сетевых задержек. Алгоритм проверки схемы основан на построении ориентированного графа, где вершинами являются контактные точки компонентов, а рёбрами – установленные проводники. При инициализации соединения система фиксирует идентификаторы исходной и целевой точек, их семантические атрибуты и геометрические параметры.

Результаты функционального тестирования подтвердили стабильность работы интерфейса, корректность алгоритмов валидации и высокое удобство взаимодействия. Выявленные в ходе опытной эксплуатации погрешности позицио-

нирования контактных узлов были устранены путём перехода к процентной системе координат. На рис. 1 представлен интерфейс приложения.

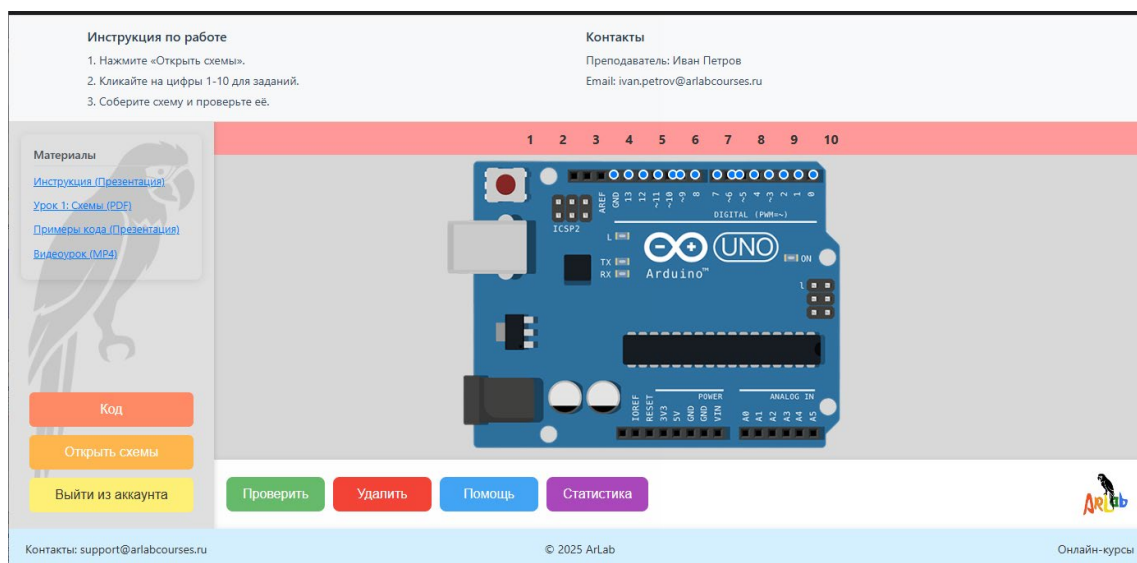


Рис. 1. Интерфейс веб-приложения ArLab

Заключение

В результате выполнения работы было создано веб-приложение ArLab для визуального моделирования электронных схем и обучения программированию микроконтроллеров, адаптированное для учащихся 10–14 лет в системе дополнительного образования. Приложение объединяет конструктор схем и редактор кода в единой интерактивной среде, обеспечивает мгновенную обратную связь и работает без установки дополнительного программного обеспечения.

Тестирование в условиях, имитирующих реальный образовательный процесс, подтвердило стабильность функционирования интерфейса, корректность алгоритмов валидации и высокое удобство взаимодействия. Разработанное решение устраняет материальные и организационные барьеры при изучении микроэлектроники, обеспечивает безопасную цифровую среду для экспериментирования и формирует устойчивую познавательную мотивацию к дальнейшему освоению инженерных дисциплин.

Практическая значимость определяется технической готовностью приложения к интеграции в учебные программы центров дополнительного образования, соответствием требованиям национального проекта «Цифровая экономика» и стратегическим задачам импортозамещения в сфере образовательных технологий. Перспективными направлениями дальнейшего развития являются расширение библиотеки виртуальных компонентов, интеграция модуля совместного проектирования и внедрение подсистемы аналитики успеваемости обучающихся.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Федеральный государственный образовательный стандарт основного общего образования. Приказ Минпросвещения России от 31.05.2021 № 287. – Москва, 2021.

2. Смирнов, А. В. Методика преподавания робототехники в школе / А. В. Смирнов. – Санкт-Петербург : БХВ-Петербург, 2019. – 352 с. – ISBN 978-5-9775-4120-8.
3. Фримен, Э. Изучаем программирование на JavaScript / Э. Фримен, Э. Робсон. – Москва : Вильямс, 2020. – 704 с. – ISBN 978-5-8459-2158-3.
4. Duckett, J. HTML and CSS: Design and Build Websites / J. Duckett. – Indianapolis : Wiley, 2011. – 490 p. – ISBN 978-1-118-00818-8.
5. Мартино, М. Arduino для начинающих / М. Мартино ; пер. с англ. – Москва : ДМК Пресс, 2018. – 256 с. – ISBN 978-5-97060-552-4.
6. Nielsen, J. Usability Engineering / J. Nielsen. – San Francisco : Morgan Kaufmann, 1993. – 362 p. – ISBN 978-0-12-518406-4.

© С. Е. Пелих, А. А. Шаранов, 2026

В. В. Пеньков^{1✉}, Е. Ю. Воронкин¹

Разработка веб-модуля для настройки ролей и разграничения доступа пользователей в рамках внутренней административной системы организации с применением технологий React и Node.js

¹Сибирский государственный университет геосистем и технологий,
г. Новосибирск, Российская Федерация
e-mail: penkov.voldemor@yandex.ru

Аннотация. В статье представлены результаты разработки веб-модуля для настройки ролей и разграничения доступа пользователей во внутренней административной системе организации. Рассмотрена проблема централизованного управления учетными записями, группами ролей и правами доступа. Для решения данной задачи выбрана ролевая модель разграничения доступа RBAC, позволяющая назначать разрешения пользователям не напрямую, а через группы ролей. Описаны основные требования к веб-модулю, архитектура клиентской и серверной частей, структура базы данных и механизм проверки прав доступа. Клиентская часть приложения реализована с использованием React и TypeScript, серверная часть – на Node.js и Express.js, в качестве системы управления базами данных использована PostgreSQL. Авторизация пользователей выполняется с применением JWT-токенов, а пароли хранятся в хешированном виде. В результате разработки создан веб-модуль, обеспечивающий управление пользователями, группами ролей, разрешениями, авторизацией и проверкой доступа к защищенным API-методам.

Ключевые слова: веб-модуль, разграничение доступа, RBAC, React, Node.js, PostgreSQL

V.V. Penkov^{1✉}, E. Y. Voronkin¹

Development of a web Module for configuring roles and user access control in an organization's internal administrative system using React and Node.js

¹Siberian State University of Geosystems and Technologies, Novosibirsk, Russian Federation
e-mail: penkov.voldemor@yandex.ru

Abstract. The article presents the results of developing a web module for configuring roles and controlling user access in an internal administrative system of an organization. The problem of centralized management of user accounts, role groups and access rights is considered. The role-based access control model, RBAC, was selected to solve this problem, as it allows permissions to be assigned to users through role groups rather than directly. The article describes the main requirements for the web module, the architecture of the client and server parts, the database structure and the access rights verification mechanism. The client part of the application was implemented using React and TypeScript, while the server part was developed using Node.js and Express.js. PostgreSQL was used as the database management system. User authorization is performed using JWT tokens, and passwords are stored in hashed form. As a result, a web module was developed that provides user management, role group management, permission configuration, authorization and access control for protected API methods.

Keywords: web module, access control, RBAC, React, Node.js, PostgreSQL

Введение

Современные корпоративные информационные системы включают большое количество пользователей, выполняющих разные функции в рамках рабочих процессов организации. Одним сотрудникам требуется только просмотр данных, другим – возможность создания и редактирования записей, а администраторам – управление учетными записями, группами ролей и настройками системы. В таких условиях важной задачей становится разграничение доступа к функциям приложения и данным.

При отсутствии централизованного механизма управления правами администрирование пользователей усложняется. Права могут назначаться вручную для каждой учетной записи, что повышает риск ошибок, дублирования настроек и случайной выдачи лишних полномочий. Кроме того, недостаточный контроль доступа может привести к несанкционированному изменению данных или выполнению действий, которые не должны быть доступны конкретному пользователю.

Одним из распространенных подходов к решению данной задачи является ролевая модель разграничения доступа RBAC. В данной модели разрешения назначаются не каждому пользователю отдельно, а роли или группе ролей. Пользователь включается в одну или несколько групп и получает набор прав, связанных с этими группами [1, 2]. Такой подход позволяет централизованно управлять доступом и изменять полномочия сразу для нескольких пользователей.

Целью работы является разработка веб-модуля для настройки ролей и разграничения доступа пользователей в рамках внутренней административной системы организации. Для достижения данной цели необходимо определить требования к модулю, спроектировать архитектуру приложения и базу данных, реализовать клиентскую и серверную части, а также проверить корректность основных сценариев работы.

Методы и материалы

Для разработки веб-модуля была выбрана клиент-серверная архитектура. Клиентская часть отвечает за отображение пользовательского интерфейса, работу с таблицами, формами, диалоговыми окнами и панелями детальной информации. Серверная часть принимает запросы от клиента, выполняет авторизацию, проверяет права доступа, обрабатывает данные и взаимодействует с базой данных.

В качестве основы для разграничения доступа использована модель RBAC. Ее применение позволяет разделить пользователей, группы ролей и разрешения. Пользователь получает доступ к функциям системы через назначенные ему группы ролей, а каждая группа содержит определенный набор разрешений. Такой подход удобен для административных систем, поскольку позволяет управлять правами централизованно и не настраивать доступ вручную для каждой учетной записи [3, 4].

Клиентская часть веб-модуля разработана с использованием React и TypeScript. React позволяет строить интерфейс из отдельных компонентов, что удобно для стра-

ниц пользователей и групп ролей, где повторяются таблицы, формы, кнопки, элементы поиска, пагинация и панели деталей [1, 5]. TypeScript применяется для статической типизации данных, что снижает вероятность ошибок при работе с API и формами [5, 6].

Для сборки клиентского приложения используется Vite. Данный инструмент обеспечивает быстрый запуск проекта в режиме разработки и оптимизированную сборку приложения [7]. Для оформления интерфейса применяются Tailwind CSS и shadcn/ui. Tailwind CSS позволяет описывать оформление через утилитарные классы, а shadcn/ui предоставляет набор локальных компонентов, которые можно адаптировать под требования административного интерфейса [8, 9]. Для управления формами и валидацией используется React Hook Form [10].

Серверная часть разработана на Node.js с использованием Express.js. Node.js позволяет выполнять JavaScript-код на сервере, а Express.js предоставляет средства для описания маршрутов, подключения middleware и обработки HTTP-запросов [11–14]. Взаимодействие между клиентской и серверной частями организовано через REST API.

Для авторизации пользователей используется JSON Web Token. После успешного входа сервер формирует JWT-токен, который клиентская часть использует при последующих запросах к защищенным API-методам [15]. Пароли пользователей не хранятся в открытом виде. Для их защиты применяется хеширование с использованием bcrypt [16]. Такой подход повышает безопасность учетных данных и снижает риск их раскрытия при несанкционированном доступе к базе данных.

В качестве системы управления базами данных выбрана PostgreSQL. Данная СУБД подходит для хранения структурированных данных и реализации связей между пользователями, группами ролей и разрешениями [17]. Для взаимодействия серверной части с PostgreSQL используется node-postgres, позволяющий выполнять SQL-запросы из Node.js-приложения [18].

Структура базы данных включает пять основных таблиц: users, role_groups, permissions, user_role_groups и role_group_permissions. Таблица users хранит учетные записи пользователей. Таблица role_groups содержит группы ролей. Таблица permissions хранит системные разрешения. Таблицы user_role_groups и role_group_permissions используются для реализации связей «многие ко многим» между пользователями и группами ролей, а также между группами ролей и разрешениями.

Проверка доступа выполняется на серверной стороне. При обращении к защищенному API-методу сервер проверяет JWT-токен, определяет текущего пользователя, получает его группы ролей и итоговый набор разрешений. Если необходимое разрешение присутствует, операция выполняется. Если разрешение отсутствует, сервер возвращает ошибку доступа. Такой подход соответствует рекомендациям по защите API и серверной проверке авторизации [19, 20].

Результаты

В результате разработки был создан веб-модуль, включающий клиентскую и серверную части, базу данных и механизм разграничения доступа. Модуль предназначен для управления пользователями, группами ролей и правами доступа во внутренней административной системе.

На клиентской стороне реализованы основные страницы приложения: страница авторизации, страница пользователей и страница групп ролей. Страница авторизации используется для входа в систему. После успешной авторизации пользователь получает доступ к административному интерфейсу. Если токен отсутствует или недействителен, приложение перенаправляет пользователя на страницу входа.

Страница пользователей содержит таблицу учетных записей, поиск, сортировку, постраничную навигацию и действия над строками. Через интерфейс можно создать нового пользователя, отредактировать данные существующего пользователя, изменить статус активности, назначить группы ролей и удалить учетную запись. Для просмотра подробной информации используется боковая панель, позволяющая не покидать страницу со списком.

Страница групп ролей реализована по аналогичному принципу. Она позволяет просматривать список групп, создавать новые группы, редактировать их название, системный код, описание и набор разрешений. Через назначение разрешений группам формируется итоговый набор прав пользователей, входящих в эти группы.

В формах создания и редактирования реализована валидация данных. Проверяются обязательные поля, формат электронной почты, пароль при создании пользователя, а также корректность выбранных групп и разрешений. Ошибки отображаются рядом с соответствующими полями формы, что упрощает исправление данных пользователем.

Для работы с большим количеством записей реализована постраничная пагинация. Клиентская часть загружает только записи выбранной страницы, а общее количество записей получает от серверной части. Это снижает нагрузку на приложение и делает работу таблиц более стабильной при увеличении количества пользователей и групп ролей.

Отдельное внимание уделено обработке ошибок и пользовательских состояний. При загрузке данных отображаются индикаторы или скелетон-элементы. Если список пуст, пользователь видит поясняющее сообщение. При ошибке загрузки отображается текст ошибки и возможность повторить запрос. Для критичных действий, таких как удаление пользователя или группы ролей, используется окно подтверждения.

На серверной стороне реализованы API-методы авторизации, получения данных текущего пользователя, управления пользователями, управления группами ролей и получения списка разрешений. Серверная часть выполняет проверку токена, хеширование паролей, валидацию входных данных, обработку ошибок и проверку прав доступа к защищенным операциям.

Для первичной подготовки базы данных используется скрипт инициализации. Он добавляет базовые разрешения, группу администраторов и тестового пользователя с полным набором прав. Это позволяет быстро проверить основные сценарии: вход в систему, управление пользователями, настройку групп ролей и контроль доступа.

Тестирование веб-модуля проводилось на уровне API и пользовательского интерфейса. Для проверки серверной части использовался Postman [21]. Проверялись успешная авторизация, ошибки входа, запросы без токена, запросы с некорректным токеном, выполнение операций без необходимых прав, создание и редактирование пользователей, назначение групп ролей, создание групп и назначение разрешений.

Основные сценарии тестирования включали проверку авторизации, защиты закрытых маршрутов, операций создания, редактирования и удаления пользователей, назначения групп ролей, управления разрешениями, поиска, пагинации, отображения ошибок и переключения светлой и темной темы. По результатам тестирования критических ошибок, препятствующих использованию модуля, выявлено не было.

88

Diplom Admin

Панель управления

🔍 Пользователи

👤 Группы

🌙 Тема

admin

Пользователи

Учётные записи каталога: поиск, просмотр доступа и управление активностью учётных записей.

Создать пользователя

Всего пользователей

54

Учётные записи в каталоге

Активные

54

Могут входить в систему

Администраторы

2

Участники группы администраторов

Поиск

Статус

Сортировка

Логин, email или имя...

Все статусы

Пользователь

Обновить

Пользователь	Email	Группы	Статус	Обновлен	Действия
admin System Admin	admin@example.com	Administrators	Активен	25.04.2026, 13:46:47	...
manager Team Manager	manager@example.com	Managers	Активен	22.04.2026, 17:51:35	...
user_1 Пользователь 1	user_1@test.ru	Administrators	Активен	24.04.2026, 19:47:42	...
user_10 Пользователь 10	user_10@test.ru	Managers Доп Группа 20 Группа 21 Группа 22 Группа 23	Активен	25.04.2026, 14:21:03	...
user_11 Пользователь 11	user_11@test.ru	—	Активен	24.04.2026, 19:23:41	...
user_12 Пользователь 12	user_12@test.ru	—	Активен	24.04.2026, 19:23:41	...
user_13 Пользователь 13	user_13@test.ru	—	Активен	24.04.2026, 19:23:41	...
user_14 Пользователь 14	user_14@test.ru	—	Активен	24.04.2026, 19:23:41	...
user_15 Пользователь 15	user_15@test.ru	—	Активен	24.04.2026, 19:23:41	...
user_16 Пользователь 16	user_16@test.ru	—	Активен	24.04.2026, 19:23:41	...

Рис. 1. Интерфейс таблицы

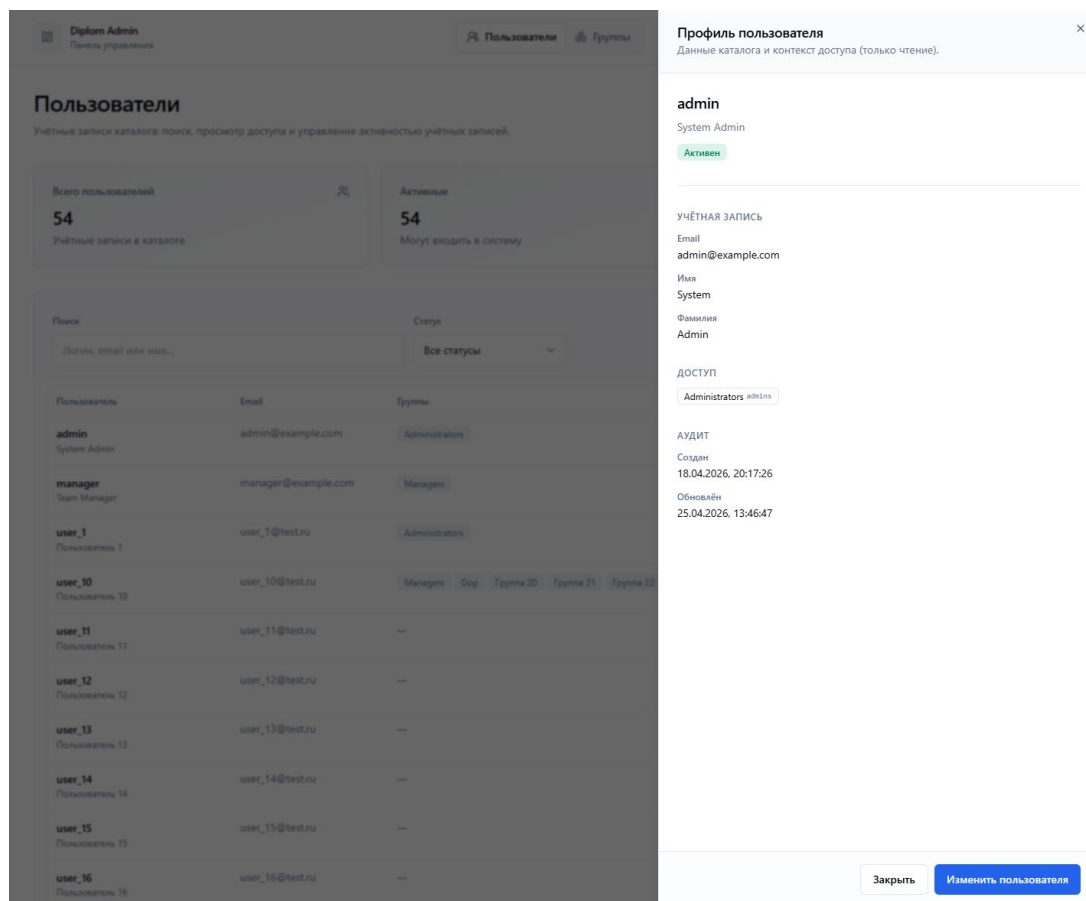


Рис. 2. Интерфейс модального окна информации

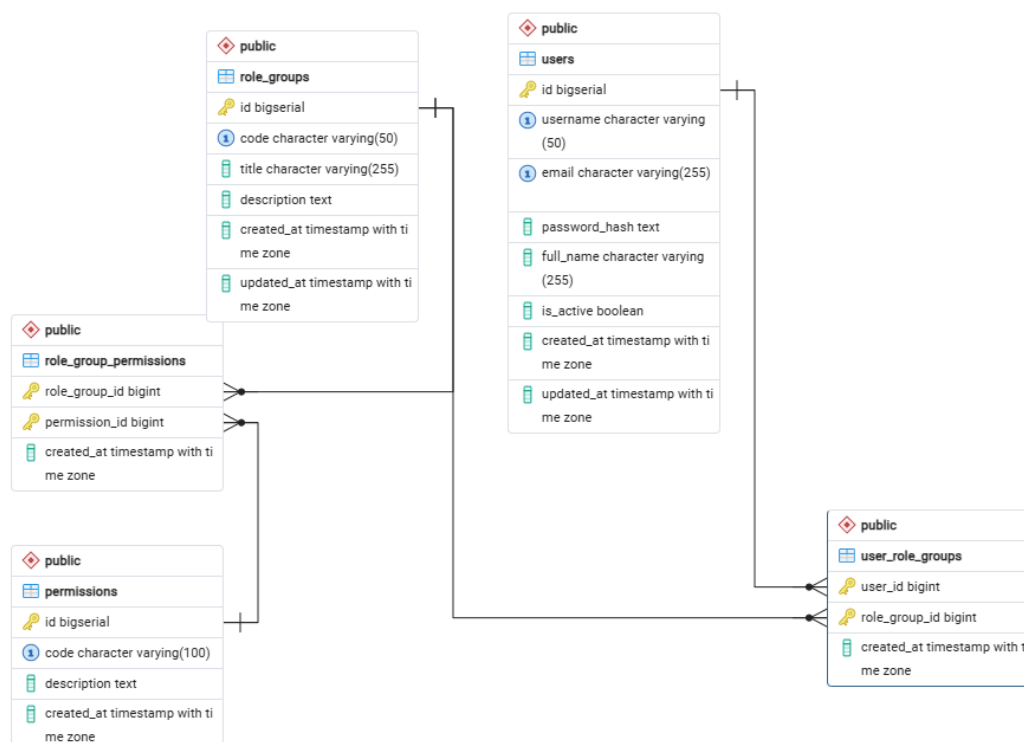


Рис. 3. Диаграмма базы данных

Обсуждение

Разработанный веб-модуль решает задачу централизованного управления доступом во внутренней административной системе. Использование RBAC позволяет избежать прямого назначения прав каждому пользователю и упростить сопровождение системы. Администратор может изменить состав разрешений у группы ролей, после чего изменения будут применены ко всем пользователям, входящим в эту группу.

По сравнению с прямым назначением прав отдельным учетным записям, ролевая модель делает структуру доступа более понятной. Пользователи объединяются в группы в соответствии с выполняемыми обязанностями, а права задаются на уровне этих групп. Такой подход снижает вероятность ошибок и упрощает добавление новых пользователей.

В то же время ролевая модель имеет ограничения. Если в системе требуется учитывать большое количество динамических условий, например подразделение пользователя, время запроса, состояние ресурса или дополнительные атрибуты объекта, может потребоваться более гибкая модель доступа. В таких случаях может применяться ABAC, однако ее реализация сложнее и требует большего количества правил проверки [19].

В разработанном модуле проверка прав выполняется как на клиентской, так и на серверной стороне. Клиентская часть использует информацию о правах для отображения или блокировки элементов интерфейса. Это улучшает пользовательский опыт, так как пользователь видит только доступные действия. Однако основная защита реализована на серверной стороне, поскольку клиентский интерфейс не может считаться надежным механизмом безопасности. Даже если пользователь попытается отправить запрос напрямую к API, сервер дополнительно проверит наличие необходимого разрешения.

Отдельным преимуществом разработанного решения является разделение клиентской и серверной частей. Такое архитектурное решение позволяет независимо развивать интерфейс и backend-логику. Например, изменение внешнего вида таблиц или форм не требует изменения базы данных, а изменение правил проверки прав доступа может быть выполнено на серверной стороне без полной переработки интерфейса.

Практическое применение разработанного модуля возможно во внутренних административных системах организаций. Он может использоваться как отдельный модуль управления доступом или как часть более крупной корпоративной информационной системы. В дальнейшем решение может быть расширено за счет добавления журнала действий пользователей, аудита изменений, восстановления пароля, интеграции с внешними сервисами авторизации и более детального анализа назначенных прав.

Заключение

В ходе работы был разработан веб-модуль для настройки ролей и разграничения доступа пользователей во внутренней административной системе органи-

зации. Модуль обеспечивает управление учетными записями, группами ролей и разрешениями, а также выполняет авторизацию и проверку прав доступа к защищенным API-методам.

Для клиентской части использованы React, TypeScript, Vite, Tailwind CSS, shadcn/ui и React Hook Form. Серверная часть реализована на Node.js и Express.js. Для хранения данных применена PostgreSQL. Авторизация пользователей выполнена с использованием JWT-токенов, а пароли защищены с помощью bcrypt.

В результате разработки создана структура базы данных, соответствующая ролевой модели разграничения доступа. Пользователи могут состоять в нескольких группах ролей, а группы могут содержать несколько разрешений. Такая структура позволяет гибко управлять полномочиями пользователей и расширять систему без существенной переработки базы данных.

Проведенное тестирование подтвердило корректность основных сценариев работы веб-модуля: авторизации, защиты маршрутов, управления пользователями, назначения групп ролей, управления группами, назначения разрешений, поиска, пагинации и обработки ошибок. Разработанное решение может быть использовано во внутренних административных системах для повышения удобства администрирования и безопасности управления доступом.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Sandhu, R. The NIST Model for Role-Based Access Control: Towards a Unified Standard / R. Sandhu, D. Ferraiolo, R. Kuhn // National Institute of Standards and Technology : сайт. – URL: https://tsapps.nist.gov/publication/get_pdf.cfm?pub_id=916402 (дата обращения: 03.05.2026). – Текст : электронный.
2. Ferraiolo, D. F. Role-Based Access Controls / D. F. Ferraiolo, D. R. Kuhn // National Institute of Standards and Technology : сайт. – URL: <https://csrc.nist.gov/files/pubs/conference/1992/10/13/rolebased-access-controls/final/docs/ferraiolo-kuhn-92.pdf> (дата обращения: 03.05.2026). – Текст : электронный.
3. React. Документация : официальный сайт. – URL: <https://react.dev> (дата обращения: 03.05.2026). – Текст : электронный.
4. React Reference Overview // React : официальный сайт. – URL: <https://react.dev/reference/react> (дата обращения: 03.05.2026). – Текст : электронный.
5. TypeScript. Документация : официальный сайт. – URL: <https://www.typescriptlang.org/docs/> (дата обращения: 03.05.2026). – Текст : электронный.
6. TypeScript Handbook // TypeScript : официальный сайт. – URL: <https://www.typescriptlang.org/docs/handbook/intro.html> (дата обращения: 03.05.2026). – Текст : электронный.
7. Vite. Документация : официальный сайт. – URL: <https://vite.dev/guide/> (дата обращения: 03.05.2026). – Текст : электронный.
8. Tailwind CSS. Styling with utility classes // Tailwind CSS : официальный сайт. – URL: <https://tailwindcss.com/docs/styling-with-utility-classes> (дата обращения: 03.05.2026). – Текст : электронный.
9. shadcn/ui. Introduction // shadcn/ui : официальный сайт. – URL: <https://ui.shadcn.com/docs> (дата обращения: 03.05.2026). – Текст : электронный.
10. React Hook Form. Get Started // React Hook Form : официальный сайт. – URL: <https://react-hook-form.com/get-started> (дата обращения: 03.05.2026). – Текст : электронный.
11. Node.js. Документация : официальный сайт. – URL: <https://nodejs.org> (дата обращения: 03.05.2026). – Текст : электронный.

12. Express. Документация : официальный сайт. – URL: <https://expressjs.com> (дата обращения: 03.05.2026). – Текст : электронный.
13. Express.js. Routing // Express : официальный сайт. – URL: <https://expressjs.com/en/guide/routing.html> (дата обращения: 03.05.2026). – Текст : электронный.
14. Express.js. Using middleware // Express : официальный сайт. – URL: <https://expressjs.com/en/guide/using-middleware.html> (дата обращения: 03.05.2026). – Текст : электронный.
15. JSON Web Token Introduction // JWT.io : сайт. – URL: <https://jwt.io/introduction> (дата обращения: 03.05.2026). – Текст : электронный.
16. bcrypt // npm : сайт. – URL: <https://www.npmjs.com/package/bcrypt> (дата обращения: 03.05.2026). – Текст : электронный.
17. PostgreSQL. Документация : официальный сайт. – URL: <https://www.postgresql.org/docs/> (дата обращения: 03.05.2026). – Текст : электронный.
18. node-postgres. Документация : сайт. – URL: <https://node-postgres.com/> (дата обращения: 03.05.2026). – Текст : электронный.
19. Authorization Cheat Sheet // OWASP Cheat Sheet Series : сайт. – URL: https://cheatsheetseries.owasp.org/cheatsheets/Authorization_Cheat_Sheet.html (дата обращения: 03.05.2026). – Текст : электронный.
20. REST Security Cheat Sheet // OWASP Cheat Sheet Series : сайт. – URL: https://cheatsheetseries.owasp.org/cheatsheets/REST_Security_Cheat_Sheet.html (дата обращения: 03.05.2026). – Текст : электронный.
21. Postman documentation overview // Postman Docs : официальный сайт. – URL: <https://learning.postman.com/docs/introduction/overview/> (дата обращения: 03.05.2026). – Текст : электронный.
22. Прохоренок, Н. А. JavaScript и Node.js для веб-разработчиков / Н. А. Прохоренок, В. А. Дронов. – Санкт-Петербург : БХВ-Петербург, 2022. – 768 с.
23. Фленов, М. В. JavaScript. Подробное руководство / М. В. Фленов. – Санкт-Петербург : БХВ-Петербург, 2020. – 720 с.
24. Кирсанов, Д. И. CSS: каскадные таблицы стилей / Д. И. Кирсанов. – Санкт-Петербург : БХВ-Петербург, 2020. – 352 с.
25. Built-in React Hooks // React : официальный сайт. – URL: <https://react.dev/reference/react/hooks> (дата обращения: 03.05.2026). – Текст : электронный.
26. Introduction to Node.js // Node.js : официальный сайт. – URL: <https://nodejs.org/learn/getting-started/introduction-to-nodejs> (дата обращения: 03.05.2026). – Текст : электронный.
27. React Router. Документация : официальный сайт. – URL: <https://reactrouter.com/> (дата обращения: 03.05.2026). – Текст : электронный.
28. dotenv // npm : сайт. – URL: <https://www.npmjs.com/package/dotenv> (дата обращения: 03.05.2026). – Текст : электронный.
29. Cross-Origin Resource Sharing (CORS) // MDN Web Docs : сайт. – URL: <https://developer.mozilla.org/en-US/docs/Web/HTTP/Guides/CORS> (дата обращения: 03.05.2026). – Текст : электронный.
30. cors // npm : сайт. – URL: <https://www.npmjs.com/package/cors> (дата обращения: 03.05.2026). – Текст : электронный.
31. Zod. Документация : официальный сайт. – URL: <https://zod.dev/> (дата обращения: 03.05.2026). – Текст : электронный.
32. WebStorm : официальный сайт. – URL: <https://www.jetbrains.com/webstorm/> (дата обращения: 03.05.2026). – Текст : электронный.
33. Authentication Cheat Sheet // OWASP Cheat Sheet Series : сайт. – URL: https://cheatsheetseries.owasp.org/cheatsheets/Authentication_Cheat_Sheet.html (дата обращения: 03.05.2026). – Текст : электронный.

34. Session Management Cheat Sheet // OWASP Cheat Sheet Series : сайт. – URL: https://cheatsheetseries.owasp.org/cheatsheets/Session_Management_Cheat_Sheet.html (дата обращения: 03.05.2026). – Текст : электронный.

35. Password Storage Cheat Sheet // OWASP Cheat Sheet Series : сайт. – URL: https://cheatsheetseries.owasp.org/cheatsheets/Password_Storage_Cheat_Sheet.html (дата обращения: 03.05.2026). – Текст : электронный..

© В. В. Пеньков, Е. Ю. Воронкин, 2026

Е. А. Плехоткин^{1✉}, Е. Ю. Воронкин¹

Разработка веб-ресурса «Дневник хроник моих интересов» на языке программирования Python

¹Сибирский государственный университет геосистем и технологий,
г. Новосибирск, Российская Федерация
e-mail: loldalender@gmail.com

Аннотация. В статье рассматривается процесс разработки веб-ресурса для публикации отзывов на различные категории и получения персонализированных рекомендаций на основе профиля интересов. Актуальность работы обусловлена необходимостью создания единой цифровой платформы для предоставления пользователям мультикатегорийных рекомендаций на основе его отзывов. В качестве основного фреймворка был выбран Django framework с применением встроенной базы данных SQLite и интегрированием GigaChat API. При разработке был разработан функционал публикации отзывов, просмотра, редактирования, удаления оставленных отзывов и формирование списка рекомендаций. Было уделено особое внимание обеспечению безопасности, масштабируемости и предоставления пользователям удобного интерфейса. В статье продемонстрировано использование современных актуальных технологий.

Ключевые слова: веб-программирование, Django, Python, ORM, HTML, CSS, Javascript, API, разработка веб-ресурсов, база данных, SQLite

E. A. Plehotkin^{1✉}, E. Yu. Voronkin¹

Development of the web resource «Diary of chronicles of my interests» in the Python programming language

¹Siberian State University of Geosystems and Technologies, Novosibirsk, Russian Federation
e-mail: loldalender@gmail.com

Abstract. The article discusses the process of developing a web resource for publishing reviews on various categories and receiving personalized recommendations based on the interests profile. The relevance of the work is due to the need to create a single digital platform for providing users with multicategory recommendations based on their feedback. The Django framework was chosen as the main framework, using the built-in SQLite database and integrating the GigaChat API. During the development, the functionality of publishing reviews, viewing, editing, deleting left reviews and creating a list of recommendations was developed. Special attention was paid to ensuring security, scalability and providing users with a user-friendly interface. The article demonstrates the use of modern relevant technologies.

Keywords: web programming, Django, Python, ORM, HTML, CSS, Javascript, API, web resource development, Database, SQLite

Введение

В современном мире пользователи зачастую проводят свободное время, потребляя медиаконтент или посещая различные заведения или мероприятия для отдыха и хорошего времяпрепровождения. В связи с этим, пользователи заинтересованы в поиске новых объектов увлечения, которые придется им по душе.

Однако из-за переизбытка информации и перенасыщенности контента, пользователям становится все труднее найти то, что соответствует их вкусам и интересам.

Лучшим вариантом для поиска являются отзывы и рекомендации других пользователей. Но в данный момент они распределены на по разным платформам, которые зачастую ограничены одной категорией контента, в следствии чего пользователям приходится тратить много времени на поиск того, что могло бы их заинтересовать. Кроме того, многие ресурсы, при составлении рекомендаций, не учитывают интересы пользователей в смежных категориях, и, следовательно, не могут предоставить более точные рекомендации.

Целью работы является разработка веб-ресурса «Дневник хроник моих интересов» на языке программирования Python. Для достижения поставленной цели были выполнены следующие задачи:

- анализ предметной области;
- анализ существующих веб-ресурсов отзывов и выявление их недостатков;
- проектирование структуры базы данных и создание интуитивного интерфейса;
- реализация основного функционала с использованием фреймворка Django;
- интеграция GigaChat API для ИИ-анализа отзывов и составления рекомендаций;
- тестирование и исправление выявленных ошибок.

Методы и материалы

Первым делом, были разработаны ключевые диаграммы, чтобы определить структуру базы данных и работу системы рекомендаций.

Была разработана диаграмма прецедентов, для определения действий, которые пользователь будет совершать в системе [1] (рис. 1).

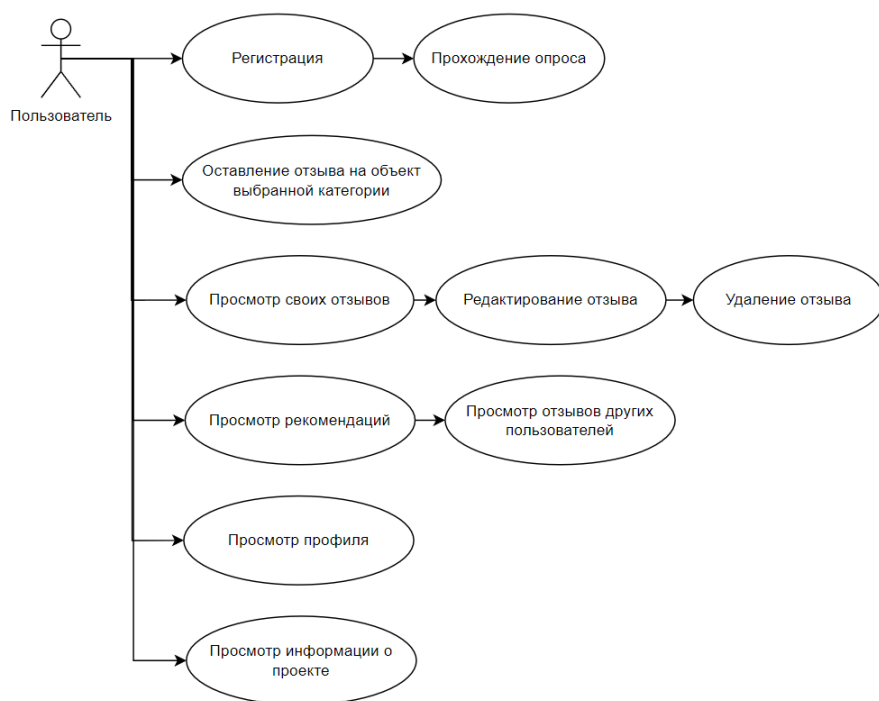


Рис. 1. Диаграмма прецедентов веб-ресурса

Следующей разработанной диаграммой стала диаграмма последовательности, которая демонстрирует процесс взаимодействия с веб-ресурсом от регистрации и оставления отзывов до получения пользователем персональных рекомендаций [2] (рис. 2).

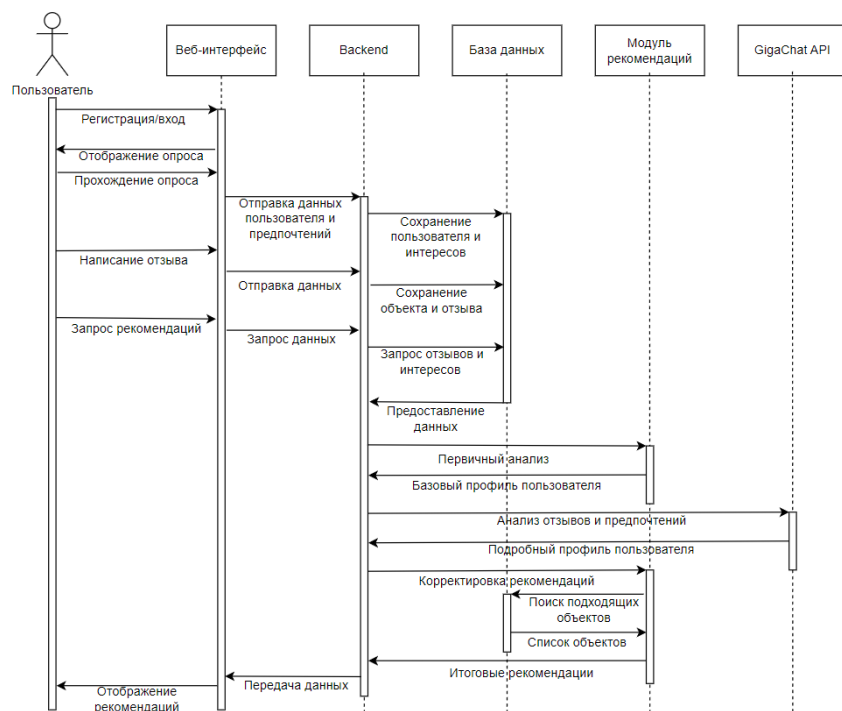


Рис. 2. Диаграмма последовательностей веб-ресурса

Для определения сущностей в базе данных, содержащейся в них информации и определения связей между сущностями была создана диаграмма связей сущностей [3] (рис. 3).

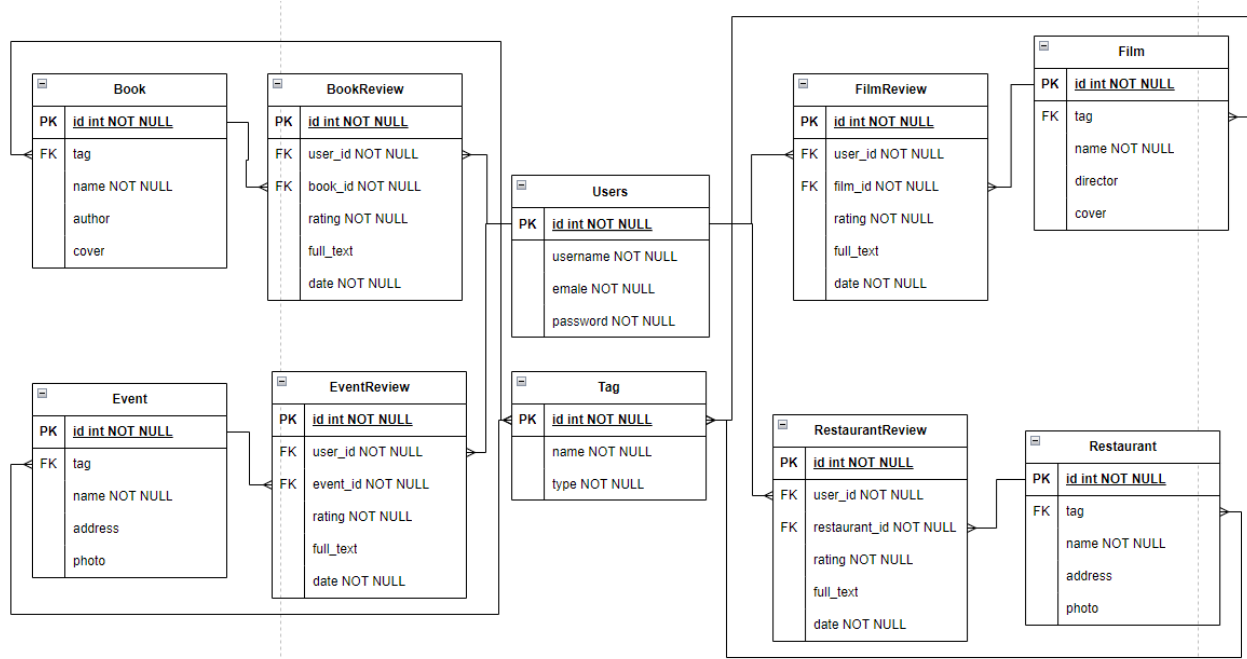


Рис. 3. Диаграмма отношения сущностей базы данных веб-ресурса

До начала разработки был создан макет интерфейса с помощью инструментов сервиса Figma (рис 4). Данный макет был использован как шаблон для верстки.

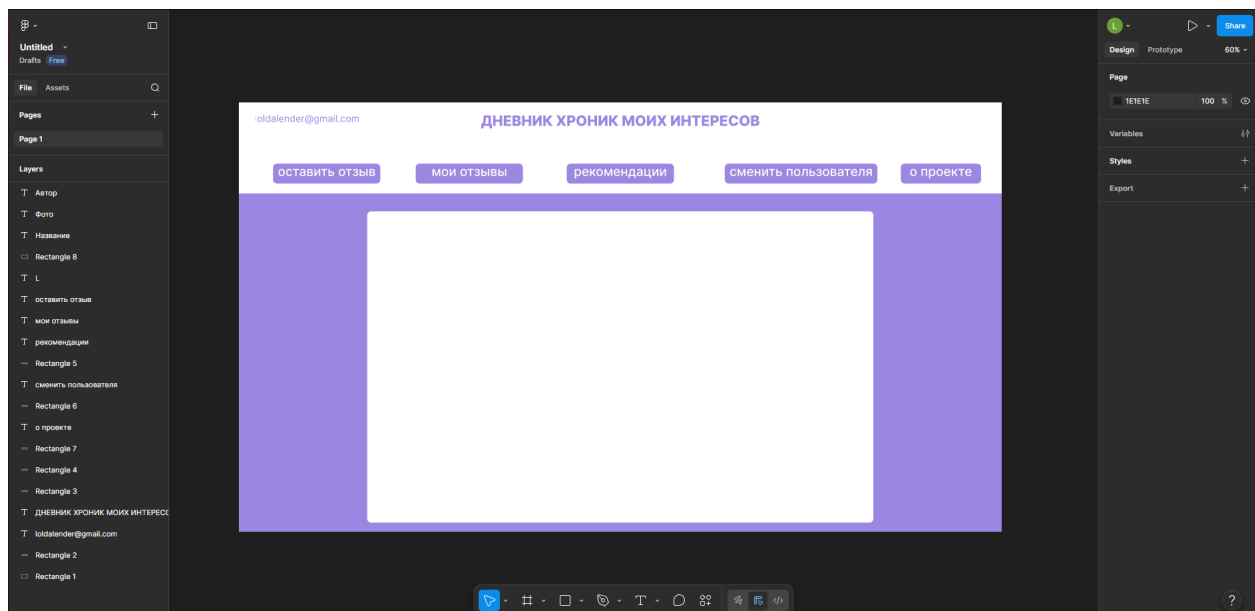


Рис. 4. Разработка дизайна шаблона веб-ресурса

Следующим шагом был выбран и обоснован стек технологий, к примеру фреймворк и СУБД.

Для разработки бэкэнд части был выбран фреймворк Django — распространенный веб-фреймворк с широким функционалом для разработки веб ресурсов. Для реализации интерфейса использовался метод MVT и HTML, CSS и JavaScript для фронтэнда.

Для хранения данных была выбрана встроенная СУБД SQLite, взаимодействие с которой осуществляется с помощью Django ORM [4].

Система рекомендаций реализована с интеграцией GigaChat API [5].

Результаты

Анализ, создание диаграмм для определения архитектуры и выбранный стек технологий (Django, SQLITE и GigaChat API) легли в основу реализации веб-ресурса.

Для взаимодействия с базой данных была написана функция home, которая является главным обработчиком, отвечающим за загрузку данных пользователя, получения списка отзывов, формирование рекомендаций и передачу информации в HTML-шаблон главной страницы.

Затем были созданы ORM модели сущностей базы данных и зарегистрированы в административной панели.

Среди моделей сущностей находятся модели объектов для каждой категории, модели отзывов на объекты для каждой категории и модель тегов, которая была вручную заполнена множеством тегов через административную панель (рис. 5).

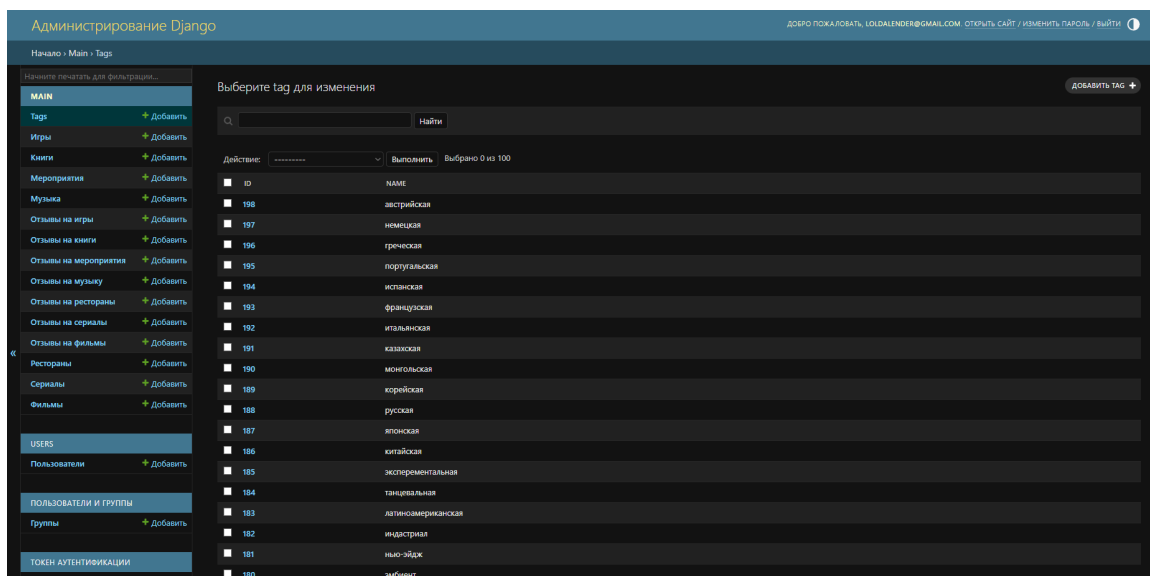


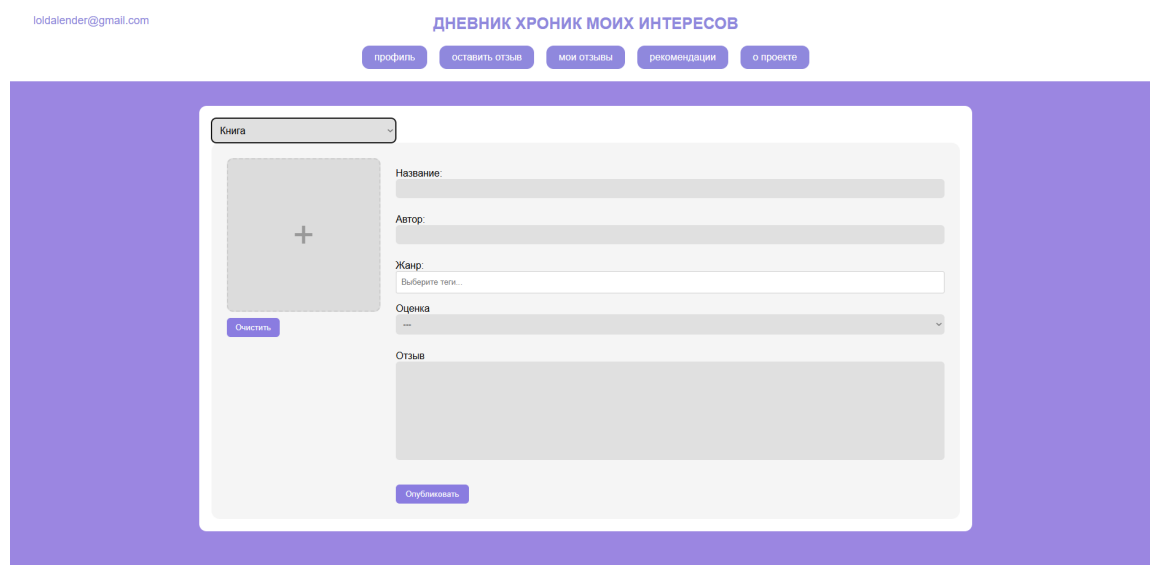
Рис. 5. Заполненная модель тегов в административной панели Django

Затем были разработаны формы ввода данных с использованием системы Django forms для централизованной проверки данных, генерации HTML-полей и

обработки загружаемых изображений. Формы были созданы для всех моделей объектов и отзывов.

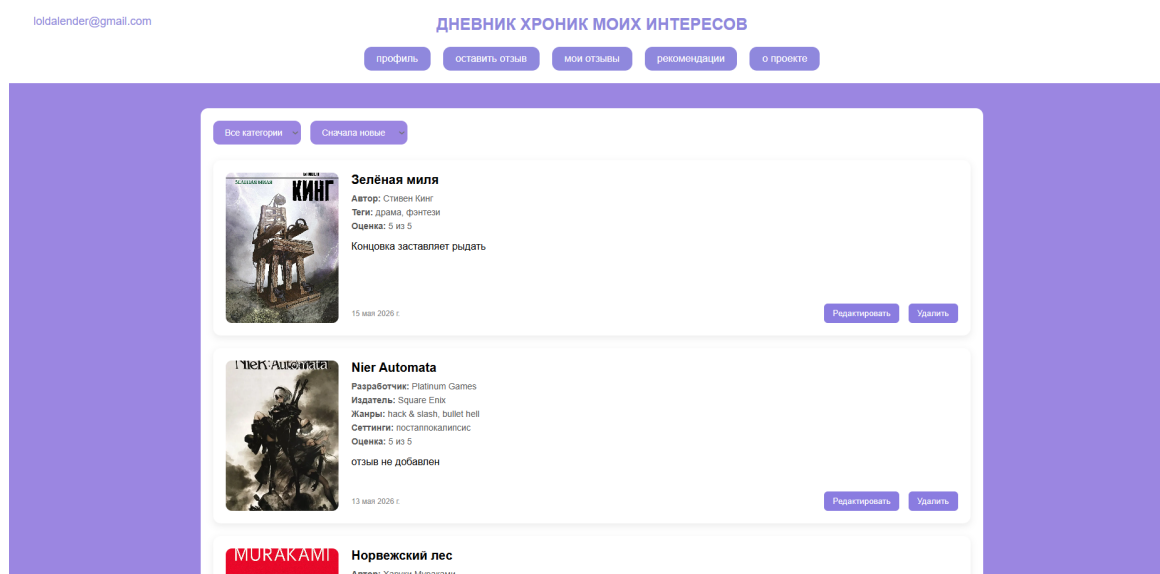
После форм был разработан обработчик создания отзывов `create_review`, который необходим для приема данных из HTML-форм и сохранения данных объекта и отзыва, которые пользователь введет, заполняя форму.

Для создания пользовательского интерфейса были разработаны HTML-шаблоны для формы написания отзыва (рис. 6), отображения списка оставленных отзывов с возможностью редактирования и удаления (рис. 7), и для отображения списка рекомендаций (рис. 8). Функциональная часть интерфейса разработана на языке JavaScript с использованием AJAX-запросов для создания динамично-обновляемого интерфейса.



The screenshot shows a web interface for a diary titled "ДНЕВНИК ХРОНИК МОИХ ИНТЕРЕСОВ". At the top, there are navigation buttons: "профиль", "оставить отзыв", "мои отзывы", "рекомендации", and "о проекте". Below these is a form for writing a review. The form has a dropdown menu for "Книга" (Book) with a plus icon. To the right of the dropdown are input fields for "Название:" (Title), "Автор:" (Author), and "Жанр:" (Genre). Below the genre field is a "Выберите теги..." (Choose tags...) dropdown. There is a "Оценка" (Rating) dropdown set to "5". Below the rating is a large text area for "Отзыв" (Review). At the bottom of the form is a "Опубликовать" (Publish) button. The background is a solid purple color.

Рис. 6. Демонстрация формы написания отзыва



The screenshot shows a web interface for a diary titled "ДНЕВНИК ХРОНИК МОИХ ИНТЕРЕСОВ". At the top, there are navigation buttons: "профиль", "оставить отзыв", "мои отзывы", "рекомендации", and "о проекте". Below these is a list of reviews. The list has two tabs: "Все категории" (All categories) and "Сначала новые" (Newest first). The first review is for "Зелёная миля" (The Green Mile) by Stephen King, with a rating of 5 out of 5. The second review is for "Nier Automata" by Platinum Games, with a rating of 5 out of 5. The third review is for "Норвежский лес" (Norwegian Forest) by Haruki Murakami. Each review entry includes a book cover, title, author, genre, rating, and a brief description. There are "Редактировать" (Edit) and "Удалить" (Delete) buttons for each review. The background is a solid purple color.

Рис. 7. Список составленных ранее отзывов

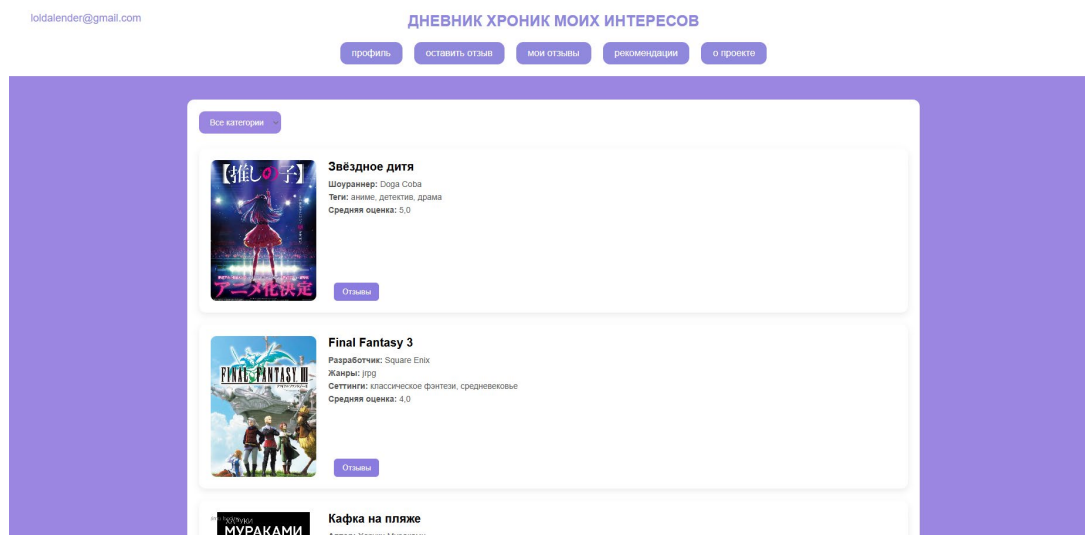


Рис. 8. Список рекомендаций

Для реализации системы рекомендаций был разработан отдельный модуль `recommendation.py`, отвечающий за анализ пользовательских предпочтений и подбор подходящих объектов. Основной функцией модуля является `get_user_recommendation(user)`, принимающая отзывы пользователя на объекты и возвращающая список рекомендаций.

Система рекомендаций работает следующим образом: из базы данных выбираются объекты, которым пользователь поставил высокую оценку. На основе этих данных система определяет предпочитаемые пользователем теги, жанры, сеттинги, авторов книг, режиссеров фильмов и т.д. Для фильтрации объектов используются ORM-запросы и выражения Q, позволяющие объединять несколько условий поиска. Из рекомендаций автоматически исключаются объекты, на которые пользователь уже оставлял отзывы. Дополнительно система рассчитывает среднюю пользовательскую оценку каждого объекта с помощью функции `Avg`.

Также для повышения качества рекомендаций дополнительно была реализована система ИИ-рекомендаций. Она анализирует не только оценки и теги объектов, но и текст отзывов пользователя. Основной функцией модуля является `get_ai_recommendations(user)`, которая получает все отзывы пользователя, формирует единый текстовый запрос и отправляет его в GigaChat по API.

В запросе передаются названия объектов, оценки, которые поставил пользователь и теги, на основе которых нейросеть анализирует предпочтения пользователя и формирует рекомендации среди всех объектов разных категорий, представленных в базе данных. Для взаимодействия с языковой моделью используется библиотека `gigachat`.

Заключение

В статье был рассмотрен процесс создания единого веб-ресурса для публикации мультикатегорийных отзывов и получения персональных рекомендаций. Це-

лью проекта стала разработка безопасного, удобного и масштабируемого веб-ресурса, предоставляющего пользователям возможность публиковать отзывы на объекты любимых категорий и получать наиболее точные рекомендации. В отличие от разработанного веб-ресурса – аналоги в виде IMDb или Отзовика являются узконаправленными, поддерживая лишь несколько категорий тематического контента или не предоставляют пользователям рекомендации.

Разработка велась на основе фреймворка Django с применением метода MVT, SQLite и GigaChat API для ИИ-анализа пользовательских отзывов и составления более точного списка рекомендаций. Были разработаны основные диаграммы (связей сущностей, прецедентов и последовательности), реализован функционал формы написания отзывов, списка оставленных отзывов и списка рекомендаций.

Итог: разработанный веб-ресурс предоставляет пользователям удобный сервис для поиска нового контента разных категорий через публикацию отзывов и получение персональных рекомендаций.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Использование диаграммы вариантов использования UML при проектировании программного обеспечения // Хабр URL: <https://habr.com/ru/articles/566218/> (дата обращения: 15.05.2024).
2. Диаграмма последовательности (sequence-диаграмма) // Хабр URL: <https://habr.com/ru/articles/814769/> (дата обращения: 15.05.2024).
3. ER-модель // Википедия URL: <https://ru.wikipedia.org/wiki/ER-модель> (дата обращения: 15.05.2024).
4. Django ORM для начинающих | Оптимизируем запросы // Хабр URL: <https://habr.com/ru/articles/503526/> (дата обращения 15.05.2026).
5. GigaChat API: Обзор | Документация для разработчиков // Sber Developer URL: <https://developers.sber.ru/docs/ru/gigachat/api/overview> (дата обращения 15.05.2026).

© Е. А. Плехоткин, Е. Ю. Воронкин, 2026

Д. И. Пономарев¹✉, Е. Ю. Воронкин¹

Разработка информационной среды учета товаров для компании-импортера на платформе «1С»

¹Сибирский государственный университет геосистем и технологий,
г. Новосибирск, Российская Федерация
e-mail: danil.ponomarev.99@mail.ru

Аннотация. В статье рассматривается процесс разработки информационной среды учета товаров для компании-импортера на базе технологической платформы «1С: Управление нашей фирмой». Актуальность работы обусловлена необходимостью автоматизации учета товаров на предприятиях малого бизнеса, осуществляющих внешнеэкономическую деятельность, где использование табличных форм Excel приводит к увеличению количества ошибок, усложнению процесса анализа себестоимости товаров компании и увеличению трудозатрат на производственные процессы. В ходе работы осуществлен анализ существующих технологических решений и информационных платформ для автоматизации учета импортных операций с товарами, спроектирована структура базы данных и реализованы ключевые бизнес-процессы учета импорта товаров. Также был разработан и внедрен внешний отчет «Анализ себестоимости импортной партии», позволяющий обеспечить распределение дополнительных расходов между товарами, что в свою очередь способствует получению более четких и точных данных о себестоимости продукции предприятия. Обеспечен удаленный доступ к разработанной информационной системе через веб-клиент с использованием Apache HTTP Server.

Ключевые слова: информационная среда, учет импорта товаров, автоматизация производственных процессов, технологическая платформа, удаленный доступ

D. I. Ponomarev¹✉, E. Yu. Voronkin¹

Development of an information environment for inventory accounting of an importing company using the 1C platform

¹Siberian State University of Geosystems and Technologies, Novosibirsk, Russian Federation
e-mail: danil.ponomarev.99@mail.ru

Abstract. The article discusses the process of developing an information environment for goods accounting for an importing company based on the platform "1C: Managing Our Company". The relevance of the work is due to the need to automate the accounting of import operations in small businesses, where the use of Excel spreadsheets leads to an increase in the number of errors, complicates cost analysis, and increases labor costs. In the course of the work, an analysis of existing solutions for automating import operations accounting was carried out, a database structure was designed, and the main business processes for accounting for the import of goods were implemented. An external report "Cost Analysis of an Import Batch" was also developed, which ensures the distribution of additional expenses among goods and allows obtaining more accurate data on product costs. Remote access to the system was also implemented via a web client using Apache HTTP Server.

Keywords: information environment, accounting for import of goods, automation of production processes, technological platform, remote access

Введение

Современное функционирование компаний, осуществляющих внешнеэкономическую деятельность, нуждается в грамотной автоматизации производственных процессов. Предприятия малого бизнеса особенно подвержены данной тенденции в связи с тем, что они осуществляют свою непосредственную деятельность в условиях ограниченности ресурсов, в том числе необходимых для разработки информационных баз и внедрения программных продуктов.

В исследуемой в рамках данной работы компании-импортере учет импорта товаров осуществляется путем использования электронных таблиц Excel. По мере роста номенклатуры и количества операций с товарами такой метод представляется менее удобным за счет ограниченного функционала, в связи с чем при выполнении сотрудниками компании повседневных производственных задач появляются ошибки при вводе массивов данных, производится дублирование информации, а также возникают сложности с анализом себестоимости товаров.

Целью настоящей работы является анализ деятельности компании по учету импорта товаров в целях автоматизации с дальнейшей разработкой и реализацией информационной среды на основе выбранной технологической платформы «1С» с конфигурацией «Управление нашей фирмой» [1]. При этом сам процесс автоматизации учета импортных товаров позволяет не только обеспечить соответствие законодательным требованиям, но и получать оперативную управленческую информацию, необходимую для принятия обоснованных решений [2].

Для достижения поставленной цели были поставлены и выполнены следующие задачи:

1. Анализ деятельности компании для осуществления процесса автоматизации учета импорта.
2. Внедрение подходящего программного продукта с учетом особенностей осуществления предприятием внешнеэкономической деятельности.
3. Проектирование архитектуры системы, разработка базы данных, реализация ключевых бизнес-процессов и обеспечение веб-доступа к приложению.

В качестве основного направления проводимого исследования была выбрана и обоснована платформа «1С». Внедрение информационной среды на основе данной платформы направлено прежде всего на повышение эффективности работы компании и создание базы для ее дальнейшего развития.

Методы и материалы

В целях наглядности демонстрирования структуры справочников была построена ER-диаграмма. С ее помощью можно более подробно проанализировать ключевые взаимосвязи в имеющейся базе данных (рис. 1).

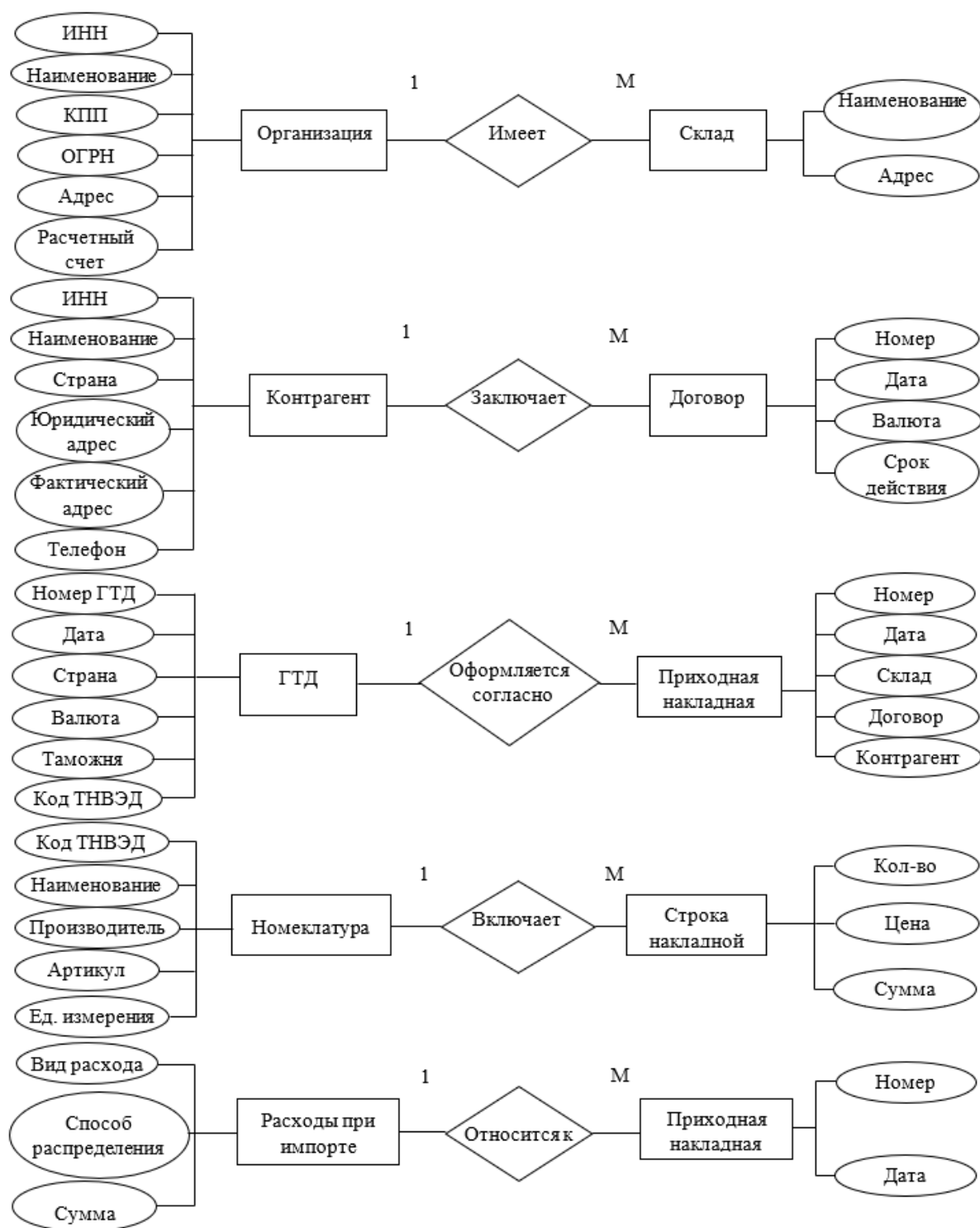


Рис. 1. Диаграмма ER

Затем была построена диаграмма прецедентов, которая отражает процесс взаимодействия пользователей с информационной системой. Были выделены три ключевые роли: администратор, генеральный директор и складской работник. Благодаря такому подходу была реализована упрощенная ролевая модель с разграничением прав доступа пользователей к системе (рис. 2).



Рис. 2. Диаграмма прецедентов

Для того, чтобы понять основы самого бизнес-процесса импорта товаров на малом предприятии, также была создана диаграмма классов, показывающая основные взаимосвязи между документацией на импортные товары. Ее использование позволило определить состав и механизмы формирования ключевых документов. В свою очередь указанные связи помогают отследить процесс автоматизации документарного учета импортных товаров в компании (рис. 3).

В процессе проектирования и разработки использовались в основном типовые механизмы конфигурации без внесения значительных изменений в основную структуру информационной системы. Указанный подход позволяет сохранить совместимость с последующими обновлениями внедряемой в деятельность малого предприятия технологической платформы и сократить трудозатраты на ее сопровождение в дальнейшем.

Для организации удаленного доступа к базе данных использовался веб-клиент технологической платформы «1С» [3], подключенный с веб-сервером Apache HTTP Server [4], а также дополнительно был настроен проброс портов и DDNS-доступ для подключения пользователей через интернет (рис. 4).



Рис. 3. Диаграмма классов

Статус: Мастер, Сеть, Безопасность, Приложение, Проброс портов

Путь: Приложение-Проброс портов

Включено ☒

Имя: 1C

Протокол: TCP

Начальный IP адрес WAN хоста:

Конечный IP адрес WAN хоста:

WAN соединение: p0

Начальный порт WAN: 8081

Конечный порт WAN: 8081

Включить MAC Mapping: ☐

IP адрес LAN хоста: 192.168.0.11

Начальный IP адрес LAN хоста: 80

Конечный IP адрес LAN хоста: 80

Изменить Отменить

Включено	Имя	Начальный IP адрес WAN хоста	Начальный порт WAN	Начальный IP адрес LAN хоста	Начальный порт LAN хоста	WAN соединение	Изменить	Удалить
✓	1C		8081	80	80	p0		
	TCP		8081	80	80	192.168.0.11		

© 2008-2015 ZTE Co Ltd. Все права защищены.

Рис. 4. Проброс портов на роутере

Результаты

В ходе выполнения работы была развернута информационная среда учета импортных товаров на базе платформы «1С» с конфигурацией «Управление

нашей фирмой». Данная конфигурация представляет программный продукт, предназначенный для полной автоматизации оперативного управленческого учета на предприятиях малого бизнеса [5]. Использование данной системы позволило повысить эффективность осуществления предприятием внешнеэкономической деятельности: реализовать централизованное хранение массива имеющихся данных о товарах, автоматизировать учет импортных операций, сократить количество ошибок, связанных с вводом и обработкой информации вручную.

Также были сформированы основные справочники системы, к которым относятся номенклатура, контрагенты и склады. Реализованы ключевые бизнес-процессы учета импортных операций: создание грузовой таможенной декларации, оформление приходных и расходных накладных, учет дополнительных расходов и контроль складских остатков предприятия.

Особенности осуществления компанией внешнеэкономической деятельности все же потребовали внесения некоторых доработок типовых отчетов, используемых в рассматриваемой конфигурации. В целях учета таких особенностей деятельности и повышения точности процесса анализа себестоимости был разработан внешний отчет «Анализ себестоимости импортной партии», который позволяет обеспечить более пропорциональное распределение дополнительных расходов между товарами в рамках каждой импортной партии (рис. 5). При этом у каждого отчета разработчик может предусмотреть несколько вариантов настроек, представляющих данные в виде диаграммы, таблицы или группировок, имеющих различный состав полей, итогов и т. п. [6]

← → Анализ себестоимости импортной партии

Сформировать Настройки... Разворачивать до

Введите слово для фильтра (название т... ?)

Отбор: Документ Равно "Приходная накладная 2 от 11.04.2026"

Документ	Количество	Сумма закупки	Расходы строки	Себестоимость
Номенклатура				
Приходная накладная 2 от 11.04.2026	67,00	3 467 089,00	7 500,00	3 474 589,00
Вал CAMSHAFT,4210087	1,00	404 839,00	875,75	405 714,75
Гильза цилиндра LINER-CYLIND,5238663	24,00	955 104,00	2 066,08	957 170,08
Клапан REGULATOR-TE,2485513	10,00	75 860,00	164,10	76 024,10
Кольцо уплотнительное SEAL RING-ME,1752582	8,00	206 488,00	446,67	206 934,67
Набор прокладок GASKET KT SF,2481394	6,00	17 370,00	37,57	17 407,57
Набор прокладок KIT GASKET,4678967	3,00	319 209,00	690,51	319 899,51
Набор прокладок KIT GASKET,4800204	1,00	173 895,00	376,17	174 271,17
Набор прокладок KIT-GASKET,3135020	1,00	94 879,00	205,24	95 084,24
Набор уплотнений KIT-GASKET,3136412	1,00	87 725,00	189,77	87 914,77
Набор уплотнений KIT-GASKET,3136413	1,00	40 442,00	87,48	40 529,48
Насос PUMP GP-PS-B,3736630	1,00	628 120,00	1 358,75	629 478,75
Палец PIN AS-CRTG,3585259	4,00	347 160,00	750,98	347 910,98
Подшипник BEARING-SLEE,4J2379	2,00	19 124,00	41,37	19 165,37
Пружина винтовая, на сжатие, д. 11.1мм.ст SPRING,9J5756	2,00	2 014,00	4,36	2 018,36
Цапфа TRUNNION	2,00	94 860,00	205,20	95 065,20
Итого	67,00	3 467 089,00	7 500,00	3 474 589,00

Рис. 5. Отчет «Анализ себестоимости импортной партии»

Также стоит отметить, что была произведена успешная публикация платформы «1С» на веб-сервере Apache HTTP Server, благодаря чему у пользователей появилась возможность подключения к базе данных с любого устройства по URL (рис. 6).

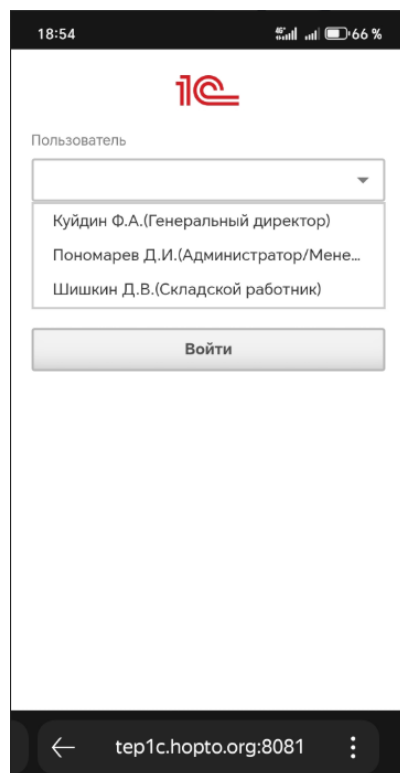


Рис. 6. Подключение к базе 1С через веб-клиент

Заключение

В данной статье рассматривался процесс разработки информационной среды учета товаров для компании-импортера на базе технологической платформы «1С» с конфигурацией «Управление нашей фирмой». Целью работы являлось создание наиболее удобной для повседневной деятельности компании информационной системы, учитывающей особенности осуществления внешнеэкономической деятельности, что в свою очередь обеспечило бы автоматизацию процессов учета импортных операций с товарами, снижение количества ошибок при вводе и обработке данных, также повысило эффективность работы предприятия и его сотрудников. В отличие от ранее используемого подхода с применением электронных таблиц Excel, разработанная информационная система позволяет осуществить хранение массивов информации о товарах, документации, складах и контрагентах в едином пространстве, что приводит к упрощению процесса анализа и обработки.

Также был реализован удаленный доступ к базе данных через веб-клиент с использованием Apache HTTP Server, что в свою очередь позволило обеспечить возможность работы с системой через браузер без установки дополнительного программного обеспечения на устройство пользователя.

Итогом работы стала разработка информационной среды, обеспечивающей учет импортных операций, упрощение обработки данных и повышение эффективности работы предприятия малого бизнеса с созданием условий для дальнейшей автоматизации производственных процессов.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Платформа 1С: Управление нашей фирмой: сайт // URL: <https://v8.1c.ru/small.biz/> (дата обращения: 12.05.2026).
2. Зайцев Н. В. Путь 1С-разработки. Не спеша, эффективно и правильно: учеб. пособие. – Санкт-Петербург: Прогресс книга, 2024. – 395 с.
3. Платформа 1С: Предприятие: сайт // URL: <https://v8.1c.ru/platforma/web-klient/> (дата обращения: 12.05.2026).
4. The Apache HTTP Server Project: сайт // URL: <https://httpd.apache.org/> (дата обращения: 12.05.2026).
5. Ильюков В. Д. Реальная автоматизация малого бизнеса. 1С: Управление нашей фирмой и 1С: Розница: учеб. пособие. – Москва: 1С-Паблишинг, 2024. – 754 с.
6. Хрусталева Е. Ю. Разработка сложных отчетов в «1С: Предприятии 8». Система компоновки данных: учеб. пособие. – Москва: 1С-Паблишинг, 2024. – 1012 с.

© Д. И. Пономарев, Е. Ю. Воронкин, 2026

В. С. Пустовит¹✉

Метод формирования виртуальных защищенных каналов связи с использованием российских криптографических алгоритмов

¹Национальный исследовательский ядерный университет «МИФИ»,
г. Москва, Российская Федерация
e-mail: VSPustovit@mephi.ru

Аннотация. В статье рассматривается метод построения виртуальных защищенных каналов связи (ВЗКС), полностью базирующийся на российских криптографических стандартах. Актуальность работы обусловлена законодательными требованиями по импортозамещению в области защиты информации и выявленными в ходе сравнительного анализа недостатками популярных решений (OpenVPN, WireGuard, L2TP/IPsec), связанными с использованием зарубежных криптографических алгоритмов. Целью исследования является разработка метода формирования ВЗКС, обеспечивающих конфиденциальность сетевого взаимодействия на базе стека TCP/IPv4 с применением российских криптографических алгоритмов. В основу предложенного метода положена адаптация паттерна IK фреймворка Noise Protocol, реализованная на алгоритмах VKO (ГОСТ Р 50.1.113-2016), шифрования «Кузнечик» в режиме MGM (ГОСТ Р 34.12/13-2018) и хэширования «Стрибог» (ГОСТ Р 34.11-2018). Предложенный метод позволяет достичь минимальной задержки установления соединения (1-RTT с возможностью 0-RTT шифрования данных). В работе представлена модульная архитектура децентрализованной системы с сервером для управления узлами и прямого P2P-туннелирования трафика. Результатом работы является формализованное описание криптографического рукопожатия на основе паттерна IK, структур пакетов взаимодействия и диаграммы состояний туннеля, что в совокупности образует законченную спецификацию метода и создаёт основу для практической реализации сертифицируемого средства защиты информации.

Ключевые слова: виртуальный защищенный канал связи, российские криптографические алгоритмы, децентрализованная архитектура

V. S. Pustovit¹✉

Establishing secure virtual communication channels method using russian cryptographic algorithms

¹National Research Nuclear University MEPhI, Moscow, Russian Federation
e-mail: VSPustovit@mephi.ru

Abstract. This article discusses a method for establishing virtual secure communication channels (VSCCs) based entirely on Russian cryptographic standards. The relevance of this work stems from legislative requirements for import substitution in the field of information security and the shortcomings of popular solutions (OpenVPN, WireGuard, L2TP/IPsec) identified during a comparative analysis, which are related to the use of foreign cryptographic algorithms. The aim of the research is to develop a method for creating VSCs that ensure the confidentiality of network interactions based on the TCP/IPv4 stack using Russian cryptographic algorithms. The proposed method is based on an adaptation of the IK pattern of the Noise Protocol framework, implemented using VKO algorithms (GOST R 50.1. 113-2016), the “Kuznechik” cipher in MGM mode (GOST R 34.12/13-2018), and the “Stribog” hash function (GOST R 34.11-2018). The proposed method allows for minimal connection establishment delay (1-RTT with the possibility of 0-RTT data encryption). The paper pre-

sents a modular architecture of a decentralized system with a server for node management and direct P2P traffic tunneling. The result of this work is a formalized description of a cryptographic handshake based on the IK pattern, interaction packet structures, and a tunnel state diagram, which together form a complete specification of the method and provide the foundation for the practical implementation of a certifiable information security solution.

Keywords: secure virtual communication channel, Russian cryptographic algorithms, decentralized architecture

Введение

Рост популярности удалённой и гибридной занятости, а также необходимость объединения территориально распределённых филиалов в единое информационное пространство сопровождаются увеличением объёмов данных, передаваемых через публичные сети. В этих условиях обеспечение конфиденциальности и целостности сетевого взаимодействия становится критически важной задачей, решаемой посредством виртуальных защищённых каналов связи (ВЗКС). Нормативно-правовая база Российской Федерации [1–7] (ФЗ № 187, ФЗ № 152, Приказ ФСБ № 66) прямо предписывает применение сертифицированных российских криптографических средств для защиты информации ограниченного доступа.

Широко распространённые технологии построения ВЗКС, такие как OpenVPN, WireGuard и L2TP/IPsec, обладают рядом принципиальных ограничений. Проведённый сравнительный анализ показал, что OpenVPN и L2TP/IPsec ориентированы на зарубежные криптографические примитивы (AES, RSA) и демонстрируют значительное падение пропускной способности (до 97%) [8–10]. WireGuard, несмотря на высокую производительность, использует фиксированный стек алгоритмов (ChaCha20, Curve25519) [11–13] и не поддерживает отечественные стандарты ГОСТ, что исключает его применение в аттестуемых системах. Кроме того, все три протокола обладают устойчивыми сигнатурами, позволяющими обнаруживать и блокировать их трафик с помощью средств глубокого анализа пакетов.

Существующие научные подходы к совершенствованию ВЗКС либо ориентированы на нейросетевую аутентификацию без строгого криптографического обоснования [14–16], либо опираются на аппаратно-зависимые квантовые технологии [17–19]. В патентных решениях (RU2820803C1, RU2838158C1) преобладают централизованные архитектуры с двойной инкапсуляцией, что приводит к избыточным задержкам и усложнению управления. Таким образом, актуальность настоящего исследования обусловлена отсутствием легковесного метода построения ВЗКС, одновременно удовлетворяющего требованиям российских регуляторов по применению ГОСТ, обладающего высокой пропускной способностью и устойчивого к пассивному обнаружению трафика.

Целью исследования является разработка метода построения виртуальных защищённых каналов связи на основе российских криптографических алгоритмов, обеспечивающей конфиденциальность сетевого взаимодействия при использовании стека протоколов TCP/IP версии 4.

Предлагаемое решение

Для обоснования необходимости разработки нового метода был проведён экспериментальный сравнительный анализ трёх наиболее распространённых технологий построения ВЗКС: OpenVPN (в конфигурации UDP/TLS), WireGuard и L2TP/IPsec. Исследование выполнялось в виртуальной сетевой лаборатории под управлением гипервизора Proxmox VE 9.0.3 на базе ОС Ubuntu Server 24.04. Измерения производились по 16 количественным и качественным критериям, включая пропускную способность, задержку установления соединения, накладные расходы на инкапсуляцию, а также поддержку отечественных криптоалгоритмов.

Проведённый сравнительный анализ существующих решений показал следующее. WireGuard демонстрирует наилучшую производительность (установка соединения <10 мс, падение скорости <20%) за счёт работы на уровне ядра, однако использует фиксированный набор зарубежных криптоалгоритмов и не соответствует требованиям ГОСТ, что исключает его сертификацию. OpenVPN, напротив, обладает гибкостью в выборе криптографии, но работает в пользовательском пространстве, что приводит к падению пропускной способности до 97% и высокой нагрузке на процессор [20]. L2TP/IPsec характеризуется наихудшими показателями из-за двойной инкапсуляции и опоры на устаревшие алгоритмы. Кроме того, трафик всех трёх протоколов устойчиво идентифицируется средствами DPI. Таким образом, ни одно из рассмотренных решений не сочетает соответствие российским криптографическим стандартам с высокой производительностью и устойчивостью к пассивному обнаружению.

В основу предлагаемого метода положена адаптация паттерна ИК фреймворка Noise Protocol Framework [21], обеспечивающая минимальное количество раундов взаимодействия при установлении защищённого соединения. Выбор паттерна ИК обусловлен возможностью выполнения рукопожатия за 1-RTT с опциональной поддержкой 0-RTT шифрования данных, что существенно сокращает задержку начала полезной передачи. Также предлагается отойти от традиционного клиент-серверного взаимодействия в пользу децентрализованной архитектуры, включающей сервер управления узлами и равноправные P2P-узлы, что повышает масштабируемость и отказоустойчивость системы.

Криптографическое ядро метода полностью опирается на российские стандарты. Генерация ключевых пар выполняется по ГОСТ Р 34.10-2018, согласование ключей — по протоколу VKO (ГОСТ Р 50.1.113-2016), симметричное шифрование — алгоритмом «Кузнечик» в режиме аутентифицированного шифрования MGM (ГОСТ Р 34.12/13-2018), хэширование и диверсификация ключей — функцией «Стрибог» (ГОСТ Р 34.11-2018) в составе HKDF.

Процедура формирования ВЗКС можно разбить на 7 фаз, начиная с регистрации узла, заканчивая передачей данных внутри. Данная процедура представлена в виде UML-диаграммы последовательности (рис. 1–3).

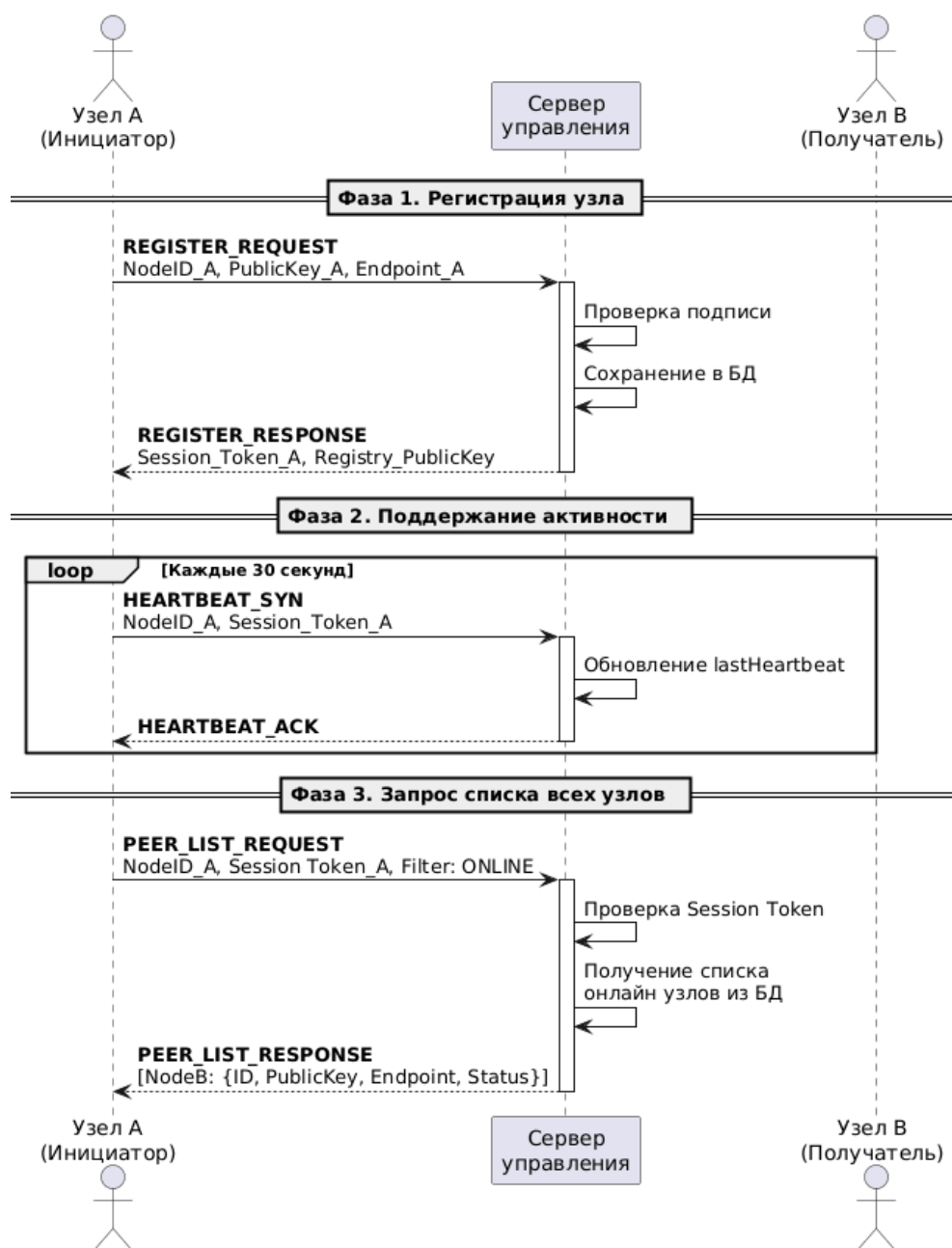


Рис. 1. UML-диаграмма последовательности формирования ВЗКС (фазы 1-3)

Прежде чем узел будет подключаться к другому узлу ему необходимо зарегистрировать себя в специальном сервере управления узлами. Его задача аутентифицировать каждый узел в сети и разрешать им соединяться прямыми P2P-соединениями.

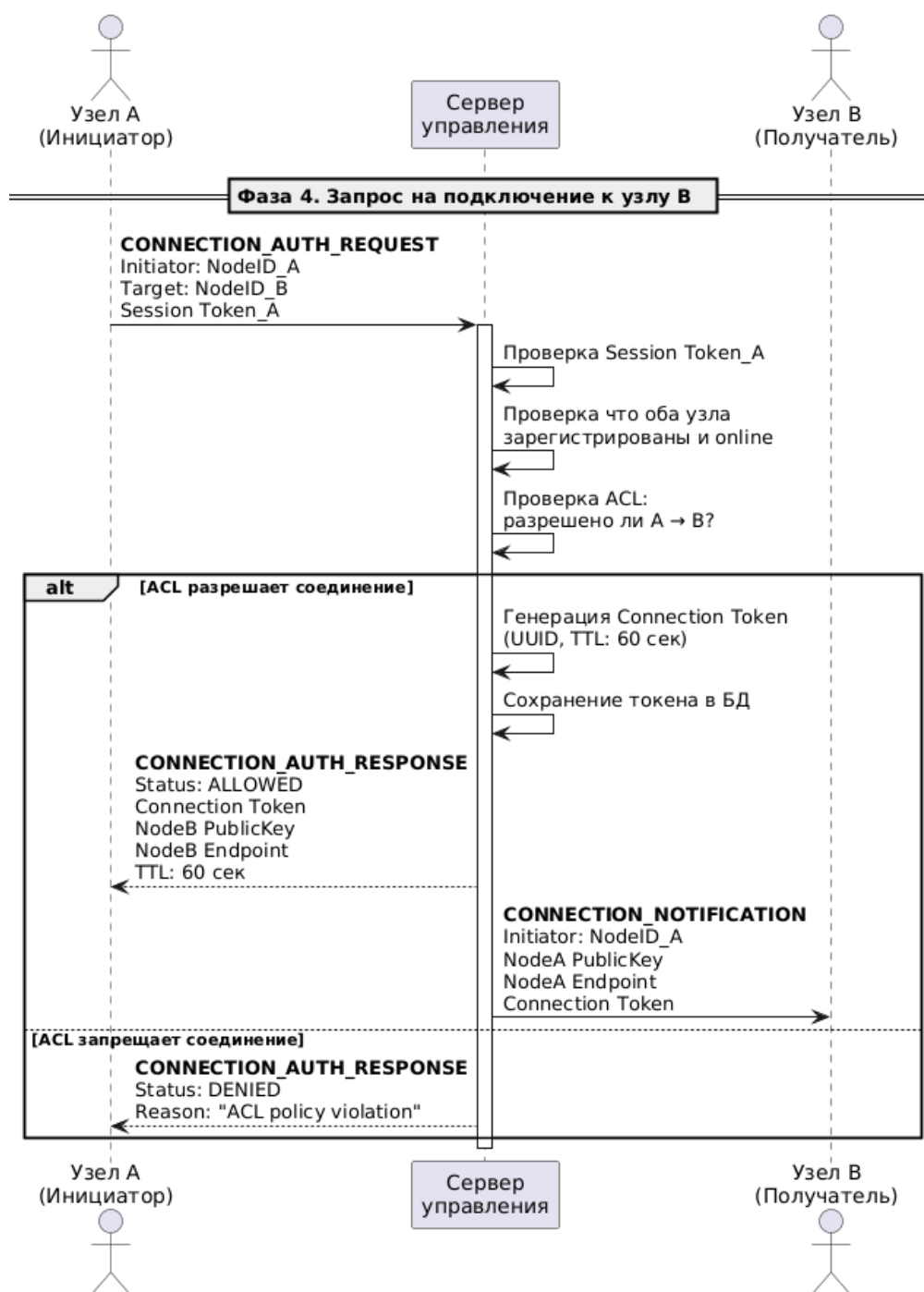


Рис. 2. UML-диаграмма последовательности формирования ВЗКС(фаза 4)

После регистрации и получения списка узлов в сети, инициатор может запросить доступ на подключение. Роль сервера управления разрешать и блокировать соединение двух узлов и при разрешении выдавать актуальную информацию о подключении (публичный статический ключ, адрес и порт подключения), а также уведомления второго узла, что к нему будут подключаться.

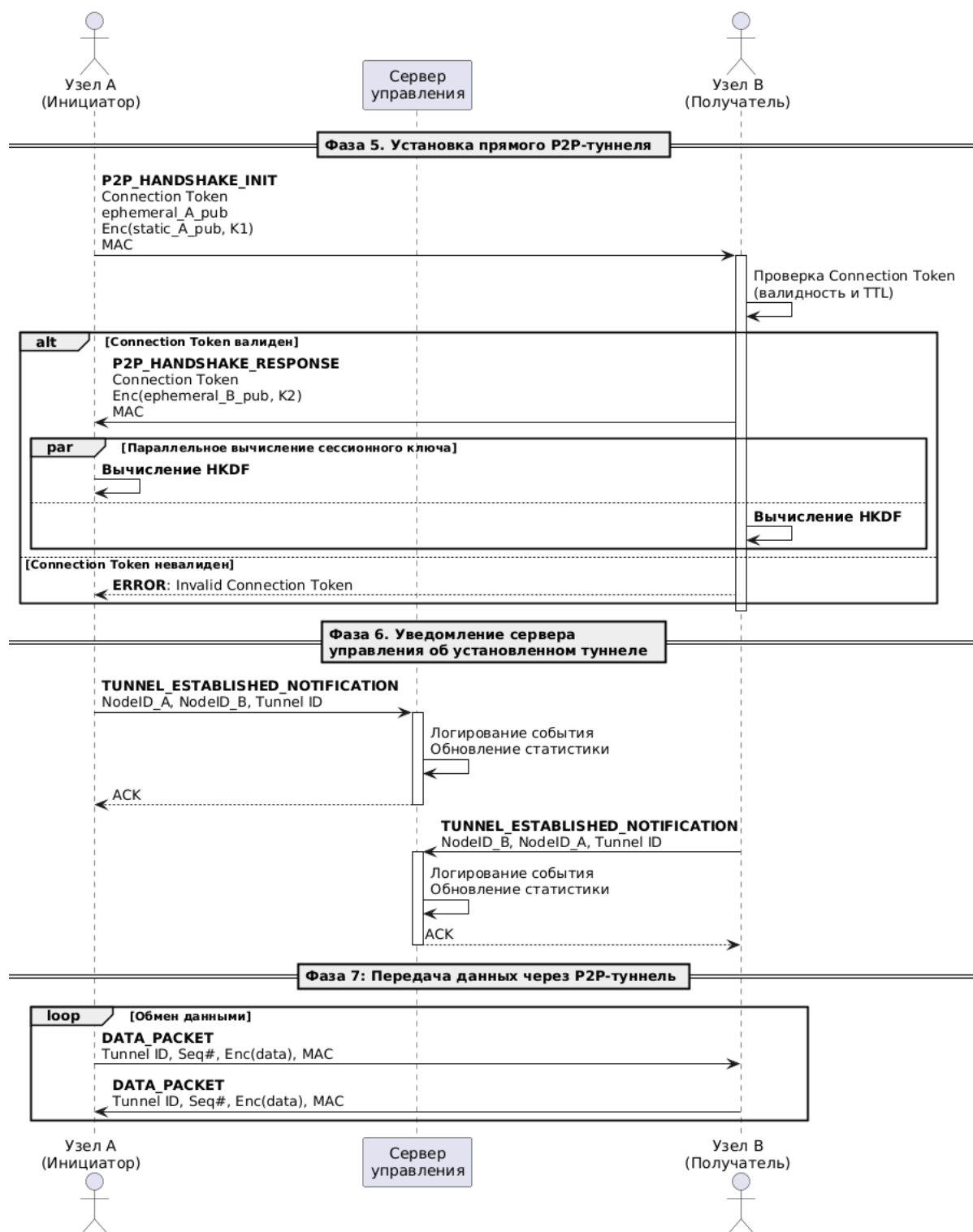


Рис. 3. UML-диаграмма последовательности формирования ВЗКС (фаза 5-7)

Если от сервера было разрешено соединение, то в разрешенное временное окно по ИК-паттерну Noise Protocol Framework происходит выработка сессионного ключа. Если все прошло успешно, оба узла сообщают серверу, что установлено соединение.

Заключение

В работе разработан метод формирования виртуальных защищённых каналов связи, основанный на адаптации паттерна ИК фреймворка Noise Protocol к российским криптографическим стандартам. Предложена формальная процедура установления защищённого соединения, включающая фазы регистрации узла, авторизации подключения, двухэтапного рукопожатия и защищённой передачи данных. Сформулировано описание криптографического ядра метода с использованием алгоритмов VKO для согласования ключей, шифра «Кузнечик» в режиме MGM для аутентифицированного шифрования и хэш-функции «Стрибог» для диверсификации ключей. Разработана процедура выработки сессионного ключа на основе четырёх комбинаций статических и эфемерных ключей, обеспечивающая свойство прямой секретности и завершающаяся за 1-RTT. Метод представлен в виде UML-диаграмм последовательности с детализацией по фазам и описанием структур передаваемых пакетов.

Метод ориентирован на построение отечественных средств защиты сетевого взаимодействия и может служить основой для программной реализации протокола с последующей сертификацией по требованиям ФСТЭК и ФСБ России.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Указ Президента Российской Федерации "О Стратегии национальной безопасности Российской Федерации" от 02.07.2021 № 400 // Официальный интернет-портал правовой информации. – 2021;
2. Указ Президента Российской Федерации "Об утверждении Доктрины информационной безопасности Российской Федерации" от 05.12.2016 № 646 // Официальный интернет-портал правовой информации. – 2016;
3. Указ Президента Российской Федерации "Об утверждении приоритетных направлений развития науки, технологий и техники в Российской Федерации и перечня критических технологий Российской Федерации" от 07.05.2018 № 899 // Официальный интернет-портал правовой информации. – 2018;
4. Указ Президента Российской Федерации "О мерах по обеспечению технологической независимости и безопасности критической информационной инфраструктуры Российской Федерации" от 30.03.2022 № 166 // Официальный интернет-портал правовой информации. – 2022;
5. Федеральный закон "О безопасности критической информационной инфраструктуры Российской Федерации" от 26.07.2017 № 187-ФЗ // Официальный интернет-портал правовой информации. – 2017 - Ст. 17-20;
6. Федеральный закон "О персональных данных" от 27.07.2006 № 152-ФЗ // Официальный интернет-портал правовой информации. - 2006 г. - Ст. 19;
7. Приказ ФСБ РФ "Об утверждении Положения о разработке, производстве, реализации и эксплуатации шифровальных (криптографических) средств защиты информации (Положение ПКЗ-2005)" от 09.02.2005 № 66 // Официальный интернет-портал правовой информации. - 2006 г. - с изм. и допол. в ред. от 12.04.2010;
8. IBM Cost of a Data Breach Report // IBM URL: <https://www.ibm.com/reports/data-breach>;

9. Verizon Data Breach Investigations Report // Verizon URL: <https://www.verizon.com/business/resources/reports/dbir>;
10. Data records breached worldwide // Statista URL: <https://www.statista.com/statistics/1307426/number-of-data-breaches-worldwide/>;
11. WireGuard: Next Generation Kernel Network Tunnel // WireGuard URL: <https://www.wireguard.com/papers/wireguard.pdf>;
12. A Cryptographic Analysis of the WireGuard Protocol // Cryptology ePrint Archive URL: <https://ia.cr/2018/080>;
13. Master, Alexander and Garman, Christina, "A WireGuard Exploration" (2021). CERIAS Technical Reports. Paper 1. <http://dx.doi.org/10.5703/1288284317610>;
14. Zaenchkovski, A., Lazarev, A., Tukaev, D., & Epifanov V. (2022). Intelligent information flow management system in innovative scientific and industrial clusters. International Journal of Parallel, Emergent and Distributed Systems, 37(3), 303–317. <https://doi.org/10.1080/17445760.2022.2060976>;
15. Artur Zaenchkovski, Alexey Lazarev. Development of an intelligent decision-making system to support scientific and industrial formations VPN connections. E3S Web Conf. 431 05034 (2023). DOI: 10.1051/e3sconf/202343105034;
16. Заенчковский А. Э., Лазарев А. И., Леднева О. В., Прокимнов Н. Н. Анализ и тестирование алгоритмов нейросетевой TCP/IP маршрутизации пакетов в частных виртуальных туннелях // Прикладная информатика. 2023. Т. 18. № 6. С. 98–108. DOI: 10.37791/2687-0649-2023-18-6-98-108;
17. Sajeed, S., Chaiwongkhot, P., Huang, A. et al. An approach for security evaluation and certification of a complete quantum communication system. Sci Rep 11, 5110 (2021). <https://doi.org/10.1038/s41598-021-84139-3>;
18. Математическая модель квантового генератора случайных чисел на основе флуктуации вакуума / А. А. Гайдаш, Р. К. Гончаров, А. В. Козубов, П. В. Яковлев // Вестник Санкт-Петербургского университета. Прикладная математика. Информатика. Процессы управления. – 2024. – Т. 20, № 2. – С. 136-153. – DOI 10.21638/spbu10.2024.202. – EDN KBUVNP;
19. Уязвимости систем квантовых коммуникаций в видимом диапазоне длин волн / Б. А. Наседкин, А. О. Исмагилов, В. В. Чистяков [и др.] // Невская фотоника-2023 : Всероссийская научная конференция с международным участием сборник научных трудов, Санкт-Петербург, 09–13 октября 2023 года. – Санкт-Петербург: Национальный исследовательский университет ИТМО, 2023. – С. 186. – EDN PSXUMZ;
20. Канатъев К.Н., Нурмагомедов М.Д., Гребенщиков А.А., Каравцев Н.А., Станолевич В.С. Передовые методы обнаружения и обфускации VPN-трафика: углубленный анализ OpenVPN и его уязвимостей в современную цифровую эпоху // ИННОВАЦИИ И ИНВЕСТИЦИИ. - 2023. - №10. - С. 211-214;
21. Д. Н. Колегов, Ю. Р. Халниязова, “Использование российских криптографических алгоритмов в протоколе безопасности сетевого уровня WireGuard”, ПДМ. Приложение, 2021, № 14, 81–84;

© В. С. Пустовит, 2026

А. О. Ревазова^{1✉}

Метод интеллектуально-адаптивной оценки и фильтрации уязвимостей в системах статического анализа безопасности программного обеспечения

¹Национальный исследовательский ядерный университет МИФИ,
г. Москва, Российская Федерация
e-mail: aanrevazova@gmail.com

Аннотация. Данная статья посвящена разработке метода интеллектуально-адаптивной оценки и фильтрации уязвимостей, выявляемых средствами статического анализа безопасности программного обеспечения. Актуальность работы обусловлена высокой долей ложноположительных срабатываний и избыточным объемом результатов анализа, затрудняющих выявление и устранение действительно критических дефектов. Разработанный подход основан на применении методов машинного обучения и контекстного анализа кода для оценки обнаруженных уязвимостей с учетом их вероятности эксплуатации, критичности и особенностей программной среды. Метод включает этапы извлечения набора признаков, интеллектуальной классификации уязвимостей и адаптивной фильтрации результатов. Предлагаемое решение обеспечивает снижение количества ложноположительных срабатываний, повышение точности ранжирования уязвимостей и оптимизацию процесса принятия решений при обеспечении безопасности ПО. Результаты экспериментальной оценки демонстрируют эффективность метода по сравнению с традиционными подходами к статическому анализу исходного кода.

Ключевые слова: статический анализ, уязвимости программного обеспечения, ложноположительные срабатывания, интеллектуально-адаптивная фильтрация, машинное обучение, контекстный анализ кода

A. O. Revazova^{1✉}

A method for intelligent adaptive vulnerability assessment and filtering in static software security analysis systems

¹National Research Nuclear University MEPhI, Moscow, Russian Federation
e-mail: aanrevazova@gmail.com

Abstract. The article is devoted to the development of a method for intelligent adaptive assessment and filtering of vulnerabilities identified by static software security analysis. The relevance of the work is due to the high proportion of false positives and the excessive volume of analysis results, which make it difficult to identify and eliminate truly critical defects. The developed approach is based on the use of machine learning methods and contextual code analysis to assess the discovered vulnerabilities, taking into account their likelihood of exploitation, criticality and features of the software environment. The method includes the stages of feature set extraction, intelligent vulnerability classification, and adaptive filtering of results. The proposed solution reduces the number of false positives, increases the accuracy of vulnerability ranking, and optimizes the decision-making process while ensuring software security. The results of the experimental evaluation demonstrate the effectiveness of the method in comparison with traditional approaches to static source code analysis.

Keywords: static analysis, software vulnerabilities, false positives, intelligent adaptive filtering, machine learning, contextual code analysis

Введение

Современные программные системы характеризуются высокой сложностью и масштабируемостью, что приводит к увеличению количества уязвимостей, выявляемых в них. Одним из ключевых инструментов обеспечения безопасности исходного программного кода является статический анализ кода (SAST), позволяющий выявлять уязвимости без выполнения программы.

Однако одной из главных проблем SAST является большое количество ложноположительных результатов (*false positives*), что существенно снижает эффективность применения этих инструментов на практике. Разработчики вынуждены тратить значительное время на анализ нерелевантных срабатываний, что замедляет процесс разработки и снижает доверие к инструментам анализа.

В связи с этим возникает необходимость разработки методов интеллектуальной фильтрации и приоритизации уязвимостей, позволяющих адаптироваться к контексту проекта и повышать точность анализа.

Статический анализ представляет собой метод исследования исходного кода без его исполнения с целью выявления ошибок и уязвимостей и способен выявлять значительную часть известных уязвимостей, однако его эффективность сильно зависит от типа уязвимости и используемого инструмента [1]. При этом точность анализа варьируется, а доля ложноположительных срабатываний может достигать критических значений.

Современные SAST-инструменты генерируют большое количество предупреждений, значительная часть которых оказывается нерелевантной [2]. Это создает дополнительную нагрузку на разработчиков и требует внедрения механизмов фильтрации.

Кроме того, исследования показывают, что комбинирование SAST с ручным тестированием повышает качество обнаружения уязвимостей, но увеличивает затраты времени и человеческих ресурсов [3].

Методы и материалы

Ложноположительные результаты являются одной из ключевых проблем статического анализа. Выделяется несколько подходов к обработке ложноположительных срабатываний, включая ручную фильтрацию, использование правил и применение методов машинного обучения [4]. Однако большинство существующих решений либо требуют значительных усилий со стороны разработчиков, либо они плохо масштабируются.

Таким образом, возникает необходимость создания адаптивных методов, способных учитывать специфику проекта и автоматически корректировать результаты анализа.

Современные исследования демонстрируют активное внедрение методов машинного обучения в процессы анализа безопасности. В рамках DevSecOps машинное обучение позволяет автоматизировать анализ и повышать точность обнаружения уязвимостей [5].

Работы в области AI для DevSecOps показывают, что модели могут использоваться для:

- 1) классификации уязвимостей;
- 2) ранжирования предупреждений;
- 3) выявления паттернов уязвимого кода;
- 4) снижения ложноположительных результатов.

Существующие исследования в области улучшения статического анализа исходного кода, в том числе с использованием методов машинного обучения [6–8] показывают значительное улучшение в сравнении с классическими средствами.

Предлагаемый метод интеллектуально-адаптивной фильтрации срабатываний основан на сочетании статического анализа и машинного обучения для автоматической фильтрации и приоритизации уязвимостей с добавлением искусственного интеллекта для формирования расширенного признакового набора. Предполагается, что благодаря комбинации классических средств статического анализа исходного кода с интеллектуально-адаптивными методами получится существенно снизить долю ложноположительных срабатываний анализаторов исходного кода. В свою очередь применение искусственного интеллекта для формирования признакового пространства может улучшить качество набора данных и его объективность.

Первым этапом является сбор данных. Он включает в себя результаты SAST-инструментов, в том числе предупреждения, типы уязвимостей и контекст кода.

На основе данных от инструментов статического анализа формируется признаковое пространство. Производится извлечение характеристик с последующим использованием в моделях.

Выбирается модель, на основе которой будет проводиться эксперимент и производится ее обучение.

Необходимо адаптировать модель машинного обучения, а также производить постоянное обновление модели на основе обратной связи от разработчиков.

Финальным этапом является ранжирование предупреждений и их приоритизация уязвимостей по степени риска.

По результатам работы средства на основе метода интеллектуально-адаптивного выявления ложноположительных срабатываний проводится сравнение с применением многокритериальной методики оценки результатов исследования.

Результаты

Ключевой особенностью метода является его адаптивность. В отличие от статических правил, данная модель учитывает особенности конкретного проекта, обучается на реальных данных и динамически обновляется.

Данная особенность позволяет существенно повысить точность анализа по сравнению с традиционными подходами.

Благодаря тому, что используется размеченный набор данных, включающий себя как истинно положительные, так и ложноположительные и ложноотрица-

тельные срабатывания, возможно корректно оценить работу предлагаемого метода.

Используемая модель машинного обучения на основе признакового пространства анализирует характеристики найденных предупреждений, после чего определяет, является ли данная уязвимость истинной или она относится к ложноположительным.

Модель использует исторические данные об уязвимостях, которые были ранее подтверждены или отвергнуты экспертом в ручном режиме и формирует набор признаков с качественными характеристиками. Благодаря такому подходу, отсутствует строгая привязка к конкретному языку программирования, архитектуре или особенностям конкретного проекта. Тестирование проводилось на языках программирования Python, C#, Java и др.

Эксперимент проводился с использованием виртуальной сетевой лаборатории, выстроенной в рамках методологии DevSecOps [9–11]. Обобщенная схема виртуального испытательного стенда представлена на рис. 2.

Предусматривается интеграция метода в конвейер DevSecOps для непрерывного мониторинга и анализа уязвимостей при добавлении нового кода:

- 1) на этапе CI/CD;
- 2) при автоматической проверке кода;
- 3) в системах контроля качества.

Для оценки эффективности метода используются следующие стандартные метрики машинного обучения: точность (precision) и полнота (recall).

Данные метрики позволяют в достаточной степени оценить эффективность разработанного метода в количественных характеристиках.

Эксперимент проводился на синтетических наборах данных, исходных кодах, размещенных в открытых репозиториях и фрагментах кода, сгенерированных нейронными сетями. Исследования показывают, что языковые модели способны генерировать текст близкий к оригинальному на котором можно проводить тестирование [11–13].

Благодаря использованию синтетических данных и использованию языковых моделей для генерации исходного кода удалось добиться формирования набора данных порядка 10^5 строк. Данный набор данных включал в себя классические Web уязвимости, уязвимости, связанные с сторонними компонентами, а также проблемы, связанные с качеством исходного кода.

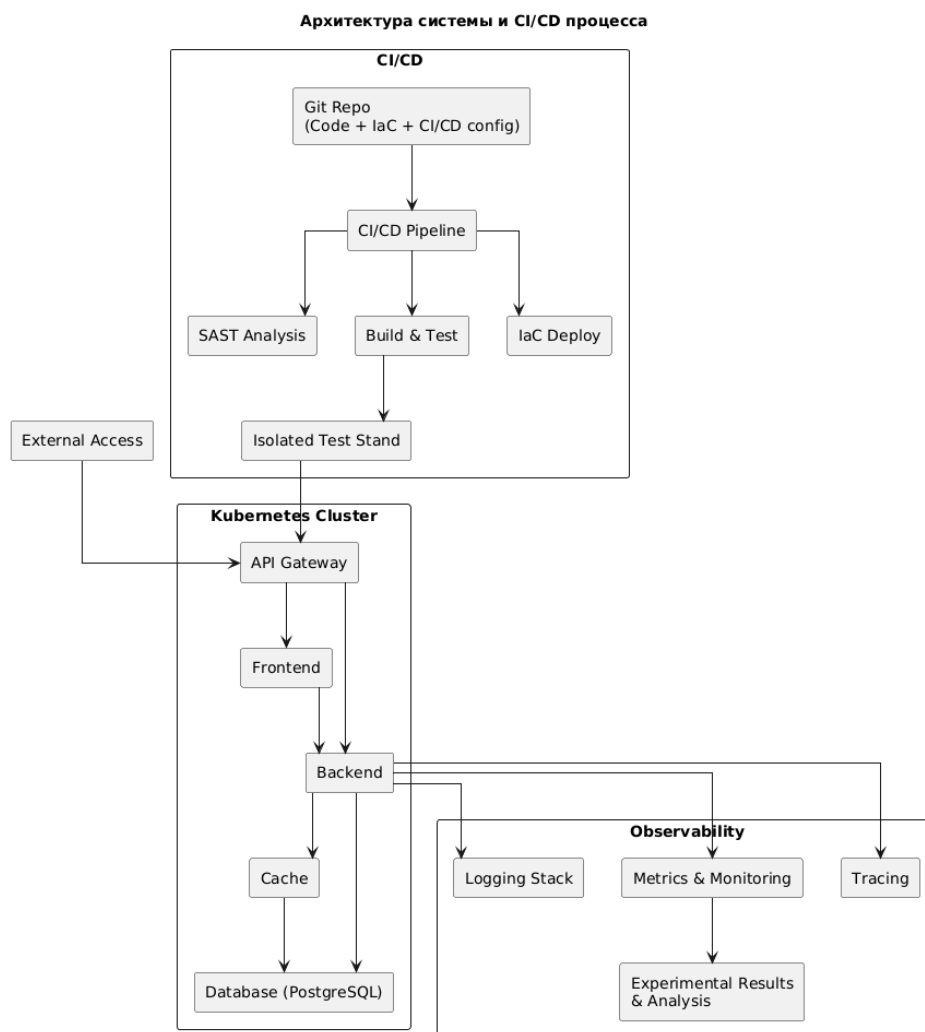


Рис. 2. Обобщенная схема виртуального испытательного стенда

В результате исследований на наборе, полученным от SAST инструментов, удалось снизить уровень ложноположительных срабатываний на 59,3% по сравнению с классическим статическим анализатором исходного кода.

Несмотря на то, что среднее количество пропущенных уязвимостей достигало 14,7% от общего объема набора данных, при условии снижения общего уровня ложно положительных срабатываний и отсутствии среди пропущенных уязвимостей с уровнем «критичный», можно утверждать о применимости данной модели.

Заключение

В работе предложен метод интеллектуально-адаптивной оценки и фильтрации уязвимостей в системах статического анализа безопасности программного обеспечения. Показано, что предлагаемый метод позволяет повысить уровень выявленных ложноположительных срабатываний в системах статического анализа безопасности программного обеспечения на 59%.

Практическая значимость метода заключается в возможности его применения в существующие системы статического анализа кода, внедренных в процессы DevSecOps.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Goseva-Popstojanova K., Perhinschi A. On the capability of static code analysis to detect security vulnerabilities // *Information and Software Technology*. – 2015. – Vol. 68. – P. 18-33.
2. Aloraini B., Nagappan M., German D. M., Hayashi S., Higo Y. An empirical study of security warnings from static application security testing tools // *Journal of Systems and Software*. – 2019. – Vol. 158.
3. Ali E. A., Ammar F. M. SAST Tools and Manual Testing to Improve the Methodology of Vulnerability Detection in Web Applications // *The International Journal of Engineering & Information Technology (IJEIT)*. – 2024. – Vol. 12 №1. P.79-83
4. Alenezi M., et al. Software security static analysis false alerts handling approaches // *International Journal of Advanced Computer Science and Applications*. – 2021. – Vol. 12, № 11.
5. Fu M., Pasuksmit J., Tantithamthavorn C. AI for DevSecOps: A Landscape and Future Opportunities // *ACM Transactions on Software Engineering and Methodology*. – 2025. – Vol. 34. №4.
6. Hüther L., Sohr K., Berger B., Rothe H. Machine Learning for SAST: A Lightweight and Adaptable Approach // *Lecture Notes in Computer Science*. – 2024. – P. 89–108.
7. Zhu J., Li K., Chen S., Fan L. A Comprehensive Study on Static Application Security Testing (SAST) Tools for Android // *Journal of LaTeX class files*. – 2024. – Vol. 14. №8.
8. Chen J., Xiang H., Li L., Zhang Y. Utilizing Precise and Complete Code Context to Guide LLM in Automatic False Positive Mitigation // *arXiv preprint arXiv:2411.03079*. – 2024.
9. Басыня Е. А. Распределенная система сбора, обработки и анализа событий информационной безопасности сетевой инфраструктуры предприятия / Безопасность информационных технологий. 2018. Т. 25. № 4. С. 43-52.
10. Басыня Е. А. Обеспечение достоверности результатов научно-практических изысканий с применением программной инженерии = Ensuring the reliability of scientific research results using software engineering / Е. А. Басыня, Е. А. Малышев. – Текст : непосредственный // Защита информации. Инсайд = Zasita informacii. Inside. – 2023. – № 4 (112). – С. 14–21.
11. Armstrong N., Hartley T. Artificially Insecure: Examining GitHub Copilot’s AI-based Vulnerability Prevention System // *Proceedings of the 2025 Silicon Valley Cybersecurity Conference (SVCC)*. – 2025.
12. Al Madi N. How Readable is Model-generated Code? Examining Readability and Visual Inspection of GitHub Copilot // *Proceedings of the 37th IEEE/ACM International Conference on Automated Software Engineering (ASE)*. – 2022.
13. Tony C., Mutas M., Díaz Ferreyra N. E., Scandariato R. LLMSecEval: A Dataset of Natural Language Prompts for Security Evaluations // *Proceedings of the 2023 IEEE/ACM 20th International Conference on Mining Software Repositories (MSR)*. – 2023.

© А. О. Ревазова, 2026

Б. С. Сагды¹✉, Н. С. Головачев¹

Разработка интерактивного наглядного пособия «Освоение Венеры» для учебно-научного центра «Планетарий» СГУГиТ

¹Сибирский государственный университет геосистем и технологий,
г. Новосибирск, Российская Федерация
e-mail: sbilzek@inbox.ru

Аннотация. В статье представлены основные результаты разработки интерактивного наглядного пособия «Освоение Венеры» для учебно-научного центра «Планетарий» СГУГиТ. Выполнен анализ предметной области, в котором рассмотрено строение Солнечной системы, а также необходимость наглядного изучения космических объектов посредством интерактивных трехмерных приложений. Для создания приложения с 3D-моделью Солнечной системы, предоставляющего информацию о планетах, а также симуляцию полета искусственного спутника к Венере и возможность сборки спутника в режиме конструктора, были использованы среда разработки Unreal Engine, а также визуальное программирование Blueprints. Разработанное приложение предоставляет пользователю возможность увидеть часть Солнечной системы и как планеты вращаются вокруг Солнца и получать систематизированные справочные данные о планете, а также в интерактивном режиме собирать спутник из компонентов и наблюдать его траекторию полета к Венере с информацией ключевых этапов миссии.

Ключевые слова: наглядное интерактивное пособие, 3D-модель Солнечной системы, спутник

B. S. Sagdy¹✉, N. S. Golovachev¹

Development of an Interactive Visual Aid "Exploration of Venus" for the Educational and Scientific Center "Planetarium" of SSUGT

¹Siberian State University of Geosystems and Technologies, Novosibirsk, Russian Federation
e-mail: sbilzek@inbox.ru

Abstract. The article presents the main results of the development of an interactive visual aid "Exploration of Venus" for the Educational and Scientific Center "Planetarium" of SGUGiT. An analysis of the subject area was carried out, which examined the structure of the Solar System, as well as the need for visual study of space objects through interactive three-dimensional applications. To create an application with a 3D model of the Solar System that provides information about the planets, as well as a simulation of an artificial satellite flight to Venus and the ability to assemble a satellite in a constructor mode, the Unreal Engine development environment and Blueprints visual programming were used. The developed application allows the user to see a part of the Solar System and how the planets revolve around the Sun, to receive systematized reference data about the planets, as well as to interactively assemble a satellite from components and observe its flight trajectory to Venus with information on the key stages of the mission.

Keywords: interactive visual aid, 3D model of the Solar System, satellite

Введение

Пользующиеся учебные материалы по астрономии не могут показывать динамику космических процессов: движение планет по орбитам, вращение небес-

ных тел или полет космических аппаратов – это процессы, которые меняются во времени. Обычные средства обучения не могут передать эту сложность. Поэтому у учащихся пропадает интерес к теме космоса, быстро пропадает вовлеченность. Знания о строении Солнечной системы и принципах космических полетов у обучающихся остаются поверхностно изученными и искаженными. Отсутствие возможности взаимодействовать с материалом усложняет развитие пространственного мышления, снижает эффективность обучения.

Пособие «Освоение Венеры» позволяет пользователю исследовать объекты Солнечной системы, увидеть вращения планет и получать информацию о планетах. Пользователь участвовать в симуляции полета спутника к Венере и собрать космический аппарат «Магеллан» как конструктор. Повышается интерес к учебе, помогает глубже понять материал и развивает пространственное и системное мышление. Интерактивное пособие помогает каждому учащемуся выбрать удобный темп и тему изучения.

Исследование предметной области

Образование в области астрономии сейчас связано с изучением объектов и процессов. Процессы характеризуются пространственной протяженностью, большими временными масштабами и сложной динамикой движения.

Актуальной проблемой астрономического образования является несоответствие между сложностью изучения и ограниченными возможностями средств обучения. В исследовании S. Salimpour, M. Fitzgerald и R. Hollow пишется разрыв между содержанием учебных программ по астрономии и современными научными представлениями о Вселенной [1].

Для изучения темы космоса можно использовать статические учебники, которые дают фундаментальные знания и точные формулы, планетарии (2D/3D), которые обеспечивают красивую визуализацию движения планет, симуляторы полетов с возможностью сборки ракет и полной физикой движения, а также конструкторы спутников, которые помогают наглядно собрать космический аппарат.

На рынке представлены различные программные решения, предназначенные для визуализации движения планет, моделирования космических полетов и сборки космических аппаратов в режиме конструктора. Однако большинство таких решений являются готовыми закрытыми продуктами, ориентированными на массового пользователя и заранее заданные сценарии взаимодействия. Их затруднительно адаптировать под локальные условия проведения занятий: техническое оснащение учебных аудиторий, структуру конкретного учебного мероприятия, требования преподавателя, уровень подготовки обучающихся и тематическую направленность учебно-научного центра «Планетарий» СГУГиТ.

В связи с этим было принято решение разработать интерактивное наглядное пособие «Освоение Венеры», включающее визуализацию движения планет внутренней части Солнечной системы, демонстрацию траектории полета космического аппарата к Венере и интерактивную сборку спутника.

Разработка интерактивной составляющей

Для разработки интерактивного наглядного пособия «Освоение Венеры» для учебно-научного центра «Планетарий» СГУГиТ было решено использовать Unreal Engine 5. Этот движок выступает основной средой разработки: в нем происходит сборка сцены, настройка освещения и визуальных эффектов, реализация логики движения объектов (орбит и траекторий), создание интерактивных элементов (выбор объектов, отображение информации, пользовательский интерфейс) и финальная сборка приложения.

К преимуществам Unreal Engine 5 можно отнести следующие моменты:

- высокое качество трехмерной графики в реальном времени;
- физически достоверная симуляция благодаря встроенной системе Chaos Physics, которая позволяет моделировать траектории движения космических аппаратов;
- активное сообщество разработчиков и обширная документация, включая официальные обучающие материалы, бесплатные готовые ассеты и поддержку со стороны Epic Games;
- бесплатность для образовательных и некоммерческих проектов – лицензирование не требует отчислений до достижения определенного уровня дохода, что делает движок доступным для использования в учебных заведениях.

Существенным преимуществом Unreal Engine 5 при разработке интерактивного учебного приложения является поддержка системы Blueprints – визуальной среды программирования, предназначенной для создания логики поведения объектов без написания кода. Blueprint-логика строится на основе графа из взаимосвязанных узлов и фактически выступает графической оболочкой для работы с классами, объектами и компонентами Unreal Engine, которые реализованы на C++. При необходимости созданные визуальные сценарии можно расширять с помощью C++, что дает сочетание быстрой разработки и возможности дальнейшей доработки.

Для проекта «Освоение Венеры» такой подход позволяет эффективно реализовать все интерактивные функции: управление движением планет, отображение справочной информации, настройку траектории полета аппарата, сборку спутника и переходы между уровнями. Кроме того, визуальный характер Blueprints упрощает последующее сопровождение проекта. После того как базовая архитектура сформирована, отдельные сценарии можно исправлять, дополнять и адаптировать под новые учебные задачи даже пользователям с небольшим опытом программирования. Это делает разработку более гибкой и пригодной для дальнейшего расширения, например при добавлении новых планет, аппаратов, учебных модулей или интеграции с другими проектами, созданными в Unreal Engine [2].

Перед началом разработки пособия была создана диаграмма прецедентов, чтобы определить, как пользователи будут взаимодействовать с пособием [3] (рис. 1).

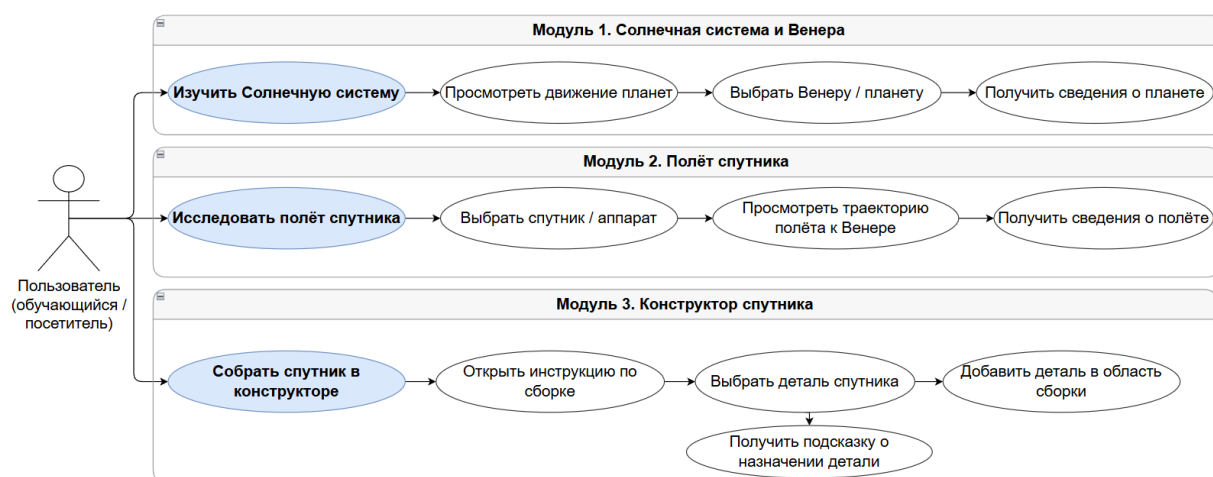


Рис. 1. Диаграмма прецедентов

Затем была создана диаграмма последовательностей, чтобы определить шаблон поведения взаимодействия пользователя и пособия [4] (рис. 2).

Уровень Солнечной системы мы создаем в редакторе UE через меню File-New Level-Empty Level. Уровень был сохранен под именем SolarSystem. Далее мы создаем наши планеты и Солнце. Используя текстуры в редакторе Material Editor создаем материал для каждой сферы [5] (рис. 3, 4).

Чтобы каждая планета вращались вокруг солнца в Blueprints добавляем событие постоянного вращения. Центром вращения были выбраны нулевые координаты и в зависимости расстояния от Солнца у каждой планеты были выбраны соответствующие расположения во вкладке Сведения. Чтобы спутник летал по определенному пути используем компонент Spline – кривая, проходящая через набор контрольных точек [6].

Для реализации учебного модуля сборки спутника была создана трехмерная модель космического аппарата «Магеллан». На этапе моделирования спутник был не просто представлен как единый объект, а предварительно разделен на отдельные функциональные элементы.

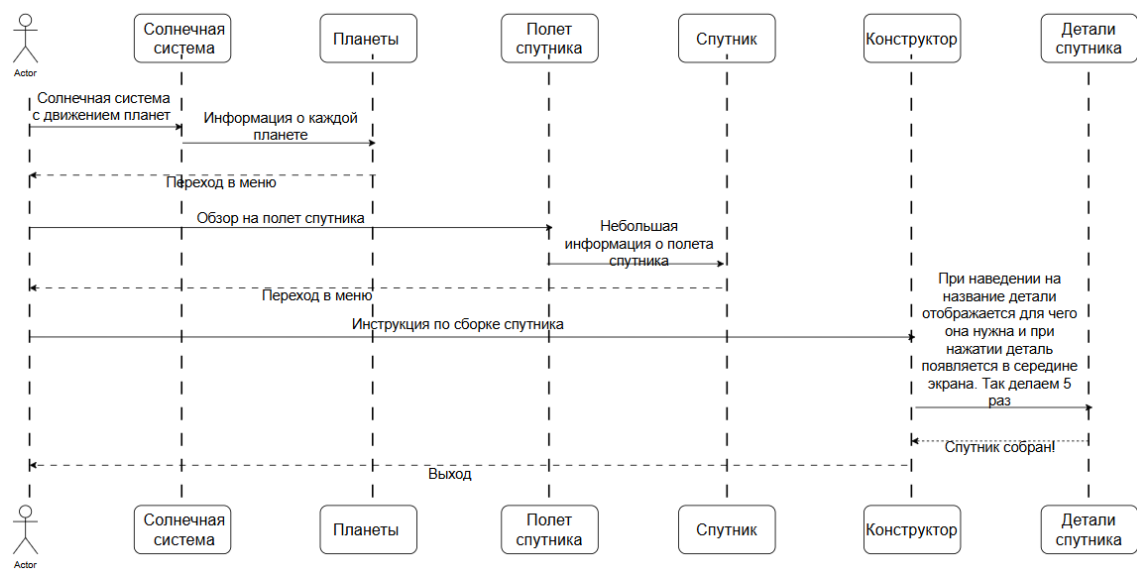


Рис. 2. Диаграмма последовательностей

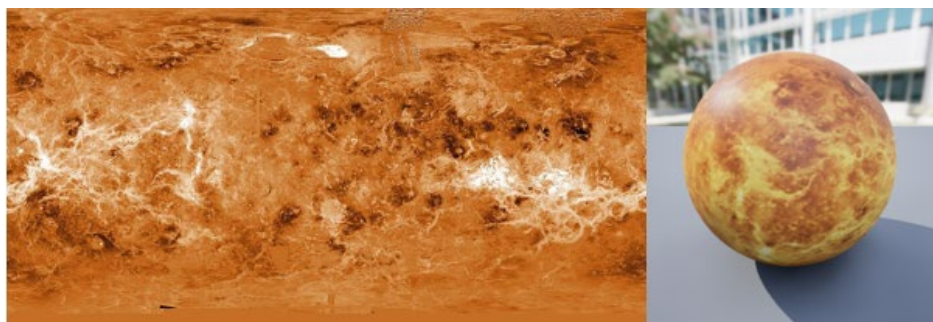


Рис. 3. Текстура для Венеры и модель планеты Венера

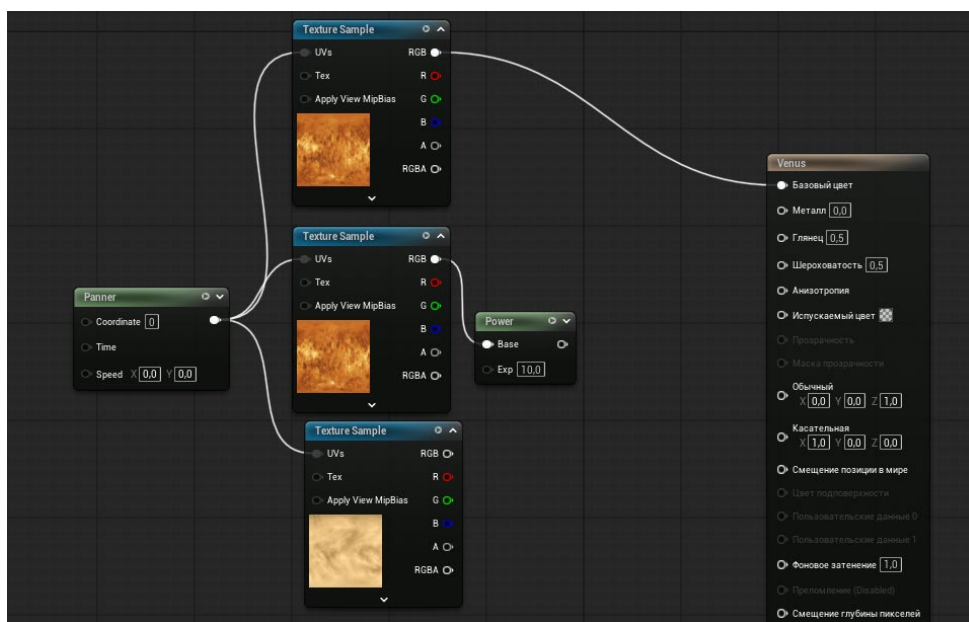


Рис. 4. Создание материала планеты Венера

В структуру модели были включены основной корпус, солнечные панели, модуль ракетного двигателя, антенна радара и связи, а также звездный сканер. Такой подход позволил не только визуализировать внешний вид аппарата, но и организовать интерактивное взаимодействие пользователя с каждой составной частью спутника.

Обучающемуся надо будет определить, какая деталь обеспечивает энерго-снабжение аппарата, какая деталь используется для передачи данных, какая деталь защищает внутренние приборы и электронику, а какая помогает ориентировать спутник в космическом пространстве. При выборе ответа элемент автоматически перемещается и устанавливается на свое место в конструкции спутника. Так продолжается до тех пор, пока вся модель космического аппарата не будет полностью собрана. Интерфейс модуля с разобранной моделью спутника представлен на рис. 5.

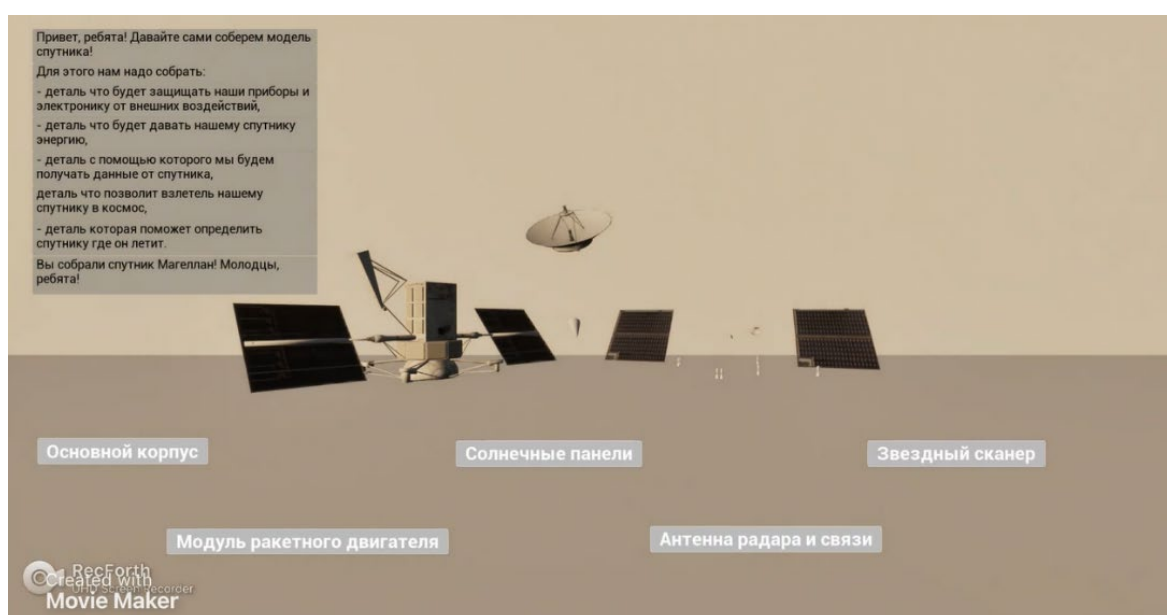


Рис. 5. Результат сборки спутника

Такие взаимодействия помогают объединить визуальное восприятие объекта и проверку знаний об устройстве и о его функциях. Вместо пассивного просмотра готовой модели, обучающийся участвует в процессе сборки космического аппарата. Это помогает лучше понять, как устроен космический аппарат, и запомнить назначение каждой детали. Пошаговая сборка делает обучение более наглядным и увлекательным, что важно при использовании пособия на демонстрационных и учебных занятиях со школьниками.

После завершения викторины пользователь получает полностью собранную модель спутника. Она может использоваться как итоговый наглядный результат изучения этого модуля. Собранный космический аппарат показывает взаимное расположение всех основных частей и помогает целостно представить конструк-

цию спутника как технической системы. Окончательный вид модели после завершения сборки представлен на рис. 6.



Рис. 6. Результат сборки спутника

В процессе работы над интерактивным наглядным пособием «Освоение Венеры» для учебно-научного центра «Планетарий» СГУГиТ, были реализованы следующие функции:

- базовая модель Солнечной системы с внутренними планетами: Солнце, Меркурий, Венера, Земля и Марс;
- визуализация движения планет и траектории спутника и алгоритм вывода справочного материала;
- сборная 3D модель аппарата для исследования Венеры – Магеллан.

Заключение

В результате выполнения работы было создано интерактивное наглядное пособие «Освоение Венеры» для учебно-научного центра «Планетарий» СГУГиТ. Оно включает в себя визуализацию Солнечной системы с вращающимися внутренними планетами, траекторию полета спутника и сборку спутника в режиме конструктора.

Разработанное пособие можно использовать как демонстрационный и учебный инструмент на занятиях, посвященных устройству Солнечной системы, изучению Венеры и принципам работы космических аппаратов.

В качестве перспектив дальнейшего развития проекта можно выделить расширение списка небесных тел и космических аппаратов, добавление новых учебных миссий, усложнение сценариев взаимодействия и включение дополнительных справочных материалов. Также возможно создание системы уровней, когда отдельные тематические модули будут объединяться в единое интерактивное об-

разовательное пространство. Такое пространство можно будет адаптировать под разные возрастные группы и учебные задачи.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Physical Review Physics Education Research. (2024) – Текст : электронный // Хабр // URL: <https://habr.com/ru/articles/234827/> (дата обращения: 12.05.2026).
2. Unreal Engine Documentation. – Текст : электронный // Epic Games Developer Documentation. – URL: <https://dev.epicgames.com/documentation/unreal-engine/> (дата обращения: 12.05.2026).
3. OMG Unified Modeling Language, Version 2.5.1. – Текст : электронный // Object Management Group. – 2017. – URL: <https://www.omg.org/spec/UML/2.5.1/About-UML/> (дата обращения: 12.05.2026).
4. Bell D. Explore the UML sequence diagram / D. Bell. – Текст : электронный // IBM Developer. – 2023. – URL: <https://developer.ibm.com/articles/the-sequence-diagram/> (дата обращения: 15.05.2026).
5. Solar Textures. – Текст : электронный // Solar System Scope. – URL: <https://www.solarsystemscope.com/textures/> (дата обращения: 15.05.2026).
6. Blueprint Spline Components Overview in Unreal Engine. – Текст : электронный // Epic Games Developer Documentation. – URL: <https://dev.epicgames.com/documentation/en-us/unreal-engine/blueprint-spline-components-overview-in-unreal-engine> (дата обращения: 15.05.2026).

© Б. С. Сагды, Н. С. Головачев, 2026

В. Ю. Сергеев¹✉, Д. Ю. Смирнов¹

Разработка программного обеспечения для определения уровня стресса сотрудников с применением технологий искусственного интеллекта

¹Сибирский государственный университет геосистем и технологий,
г. Новосибирск, Российская Федерация
e-mail: pokeronat@mail.ru

Аннотация. В статье представлены результаты разработки программного обеспечения для определения уровня стресса сотрудников с применением технологий искусственного интеллекта. Выполнен анализ предметной области, в рамках которого рассмотрены физиологические и психологические показатели, используемые для оценки уровня стресса, а также обоснована необходимость их совместного применения в системе мониторинга. Для практической реализации создана конфигурация в среде «1С:Предприятие 8.3», обеспечивающая ввод и хранение данных, а также реализован модуль анализа на языке python, формирующий итоговое заключение и перечень факторов риска. Полученные результаты демонстрируют возможность автоматизации процесса оценки и последующего использования системы для регулярного мониторинга состояния сотрудников.

Ключевые слова: уровень стресса, искусственный интеллект, 1С:Предприятие 8.3, python, мониторинг состояния сотрудников

V. Yu. Sergeev¹✉, D. Yu. Smirnov¹

Developing software to determine employee stress levels using artificial intelligence technologies

¹Siberian State University of Geosystems and Technologies, Novosibirsk, Russian Federation
e-mail: pokeronat@mail.ru

Abstract. This article presents the results of developing software for assessing employee stress levels using artificial intelligence technologies. A domain analysis was conducted, examining the physiological and psychological indicators used to assess stress levels and substantiating the need for their combined use in a monitoring system. For practical implementation, a configuration was created in the 1C:Enterprise 8.3 environment for data entry and storage, and a python analysis module was implemented to generate a final report and a list of risk factors. The obtained results demonstrate the feasibility of automating the assessment process and subsequently using the system for regular employee monitoring.

Keywords: stress level, artificial intelligence, 1C:Enterprise 8.3, python, employee health monitoring

Введение

Современная рабочая среда с высокой информационной и профессиональной нагрузкой способствует развитию психоэмоционального напряжения, влияющего на общее состояние здоровья и качество принимаемых рабочих решений. Раннее выявление симптомов стресса и выгорания позволяет своевременно реа-

лизовать профилактику и минимизировать негативные последствия как для сотрудника, так и для всего коллектива.

На практике оценка уровня стресса проводится редко и основывается на отдельных данных: медицинские показатели фиксируются отдельно от психологических методик. В связи с этим актуальной задачей является создание программного решения, совмещающего физиологические и психологические параметры и обеспечивающего автоматизированное формирование итогового заключения по состоянию сотрудника.

Исследование предметной области

Анализ уровня стресса относится к задачам мониторинга психоэмоционального состояния и предполагает учет как психологических, так и физиологических проявлений. Стресс представляет собой целостную реакцию организма на существенные факторы среды и проявляется как на уровне ощущений человека, так и в изменениях физиологических показателей. Физиологические метрики не являются строго специфичными для стресса и могут изменяться по ряду причин, поэтому их анализ должен учитывать общее состояние и сопутствующие условия [1].

Для повышения объективности оценки широко применяются стандартизированные психометрические методики. Одни методики направлены на выявление факторов рабочего стресса и характеристик организационной среды, другие на оценку текущего уровня тревоги и признаков профессионального выгорания. Применение поэтапного подхода позволяет переходить к более углубленным методикам только при наличии оснований, что снижает нагрузку на обследуемого сотрудника и повышает практическую эффективность мониторинга [2].

На российском рынке представлены решения для оценки психоэмоционального состояния, однако многие из них ограничиваются опросниками, слабо учитывают физиологические показатели и динамику, а также не обеспечивают удобной интеграции с корпоративными системами. Поэтому было принято решение разработать собственную систему, ориентированную на комплексный мониторинг, этапную проверку и формирование поясняемого заключения.

Методы и технологии

При разработке системы была выбрана модульная архитектура, предполагающая разделение подсистемы работы с данными и подсистемы анализа. Такой подход позволяет независимо развивать пользовательскую часть и аналитический модуль.

В качестве платформы для хранения данных и построения пользовательского интерфейса использована среда «1С:Предприятие 8.3» [3]. Данная платформа широко применяется для корпоративных информационных систем и предоставляет развитые средства проектирования структуры данных, настройки прав доступа и построения отчетности. Одним из преимуществ 1С является быстрая разработка прикладных систем на встроенном языке и создание пользовательских форм, ориентированных на реальные рабочие процессы.

Аналитический модуль реализован на языке python, который является одним из наиболее распространенных инструментов для задач анализа данных и машинного обучения. Язык python предоставляет широкий набор инструментов для обработки структурированных данных, построения моделей и их дальнейшего использования на новых входных данных. Кроме того, python удобен для экспериментальной отладки, что актуально при работе с данными смешанного типа [4, 5].

Для реализации модели машинного обучения использована библиотека tensorflow, которая позволяет создавать нейронные сети для задач табличной классификации и выполнять обучение с настройкой параметров. Использование keras позволяет удобно описывать модель и сохранять результаты обучения, обеспечивая их корректное воспроизведение при последующем применении.

Для подготовки данных и построения предобработки применялись библиотеки pandas и scikit-learn. Pandas обеспечивает удобную загрузку и преобразование таблиц, а scikit-learn предоставляет инструменты масштабирования признаков и разбиения данных на обучающую и контрольную выборки. Применение стандартных методов предобработки позволяет привести признаки к единому масштабу и обеспечить корректную работу модели на данных, поступающих из информационной системы.

Для интеграции между подсистемами был выбран обмен данными в формате JSON, что является универсальным способом передачи структурированных данных между приложениями. Формат JSON удобен тем, что поддерживается как средствами 1С, так и средствами python, позволяет передавать набор признаков в виде словаря «имя параметра - значение», а также упрощает диагностику ошибок при обмене. Результат анализа возвращается тем же способом и фиксируется в записи осмотра.

Таким образом, для разработки системы приложения был использован следующий стек технологий:

- «1С:Предприятие 8.3»;
- Python 3.10;
- TensorFlow/Keras;
- Scikit-learn;
- Pandas;
- Joblib;
- JSON.

Результаты

В результате выполненной работы была разработана подсистема, предназначенная для мониторинга и определения уровня стресса сотрудников на основе медицинских и психологических параметров с применением технологий искусственного интеллекта. В разработанном программном решении реализованы следующие функции:

- ведение справочника сотрудников с хранением персональных и медицинских сведений;

- фиксация результатов медицинских осмотров с внесением физиологических показателей;
- ввод результатов психологической оценки по этапной схеме (рис. 1);
- автоматический анализ данных и получение итогового заключения по уровню стресса;
- сохранение результатов анализа в документе осмотра;
- ведение истории наблюдений и формирование отчетов по текущему состоянию сотрудников, включая визуализацию распределения уровней стресса.

История здоровья Медицинский осмотр Сотрудники Отчеты ▾

← → ☆ Медицинский осмотр 000000015 от 12.05.2026 6:50:07

Провести и закрыть Записать Провести Анализ ИИ

Сотрудник: ▾

Рост:

Вес:

Сердечно-сосудистая система

Давление верхнее: Давление нижнее: Пульс:

Лабораторные исследования

Глюкоза: Холестерин:

Нервная система

Бессонница: ☐ Тремор: ☐

COPSOQ

Темп работы: Количественная нагрузка: Конфликт работа личная жизнь: Контроль на работе: Социальная поддержка:

STAI

STAI пройден: ☒ STAI баллы:

СВІ

СВІ пройден: ☐ СВІ баллы:

Заключение Искусственного интеллекта

Уровень стресса: ▾

Вероятность:

Факторы риска:

ТемпРаботы=70

Рис. 1. Интерфейс документа медицинский осмотр

На данный момент данные о медицинских осмотрах и результатах психологических методик сохраняются в информационной базе, что позволяет формировать историю наблюдений и оперативно получать актуальное заключение по каждому сотруднику. Автоматическое формирование результата и перечня факторов риска упрощает работу специалиста, обеспечивая быстрый анализ состояния и поддержку принятия решений при необходимости дополнительной проверки. Поэтапная схема обследования, реализованная в системе, позволяет рационально организовать процесс мониторинга, снижая объём лишних действий и концентрируя внимание на случаях, требующих уточнения.

Заключение

В результате выполнения работы была разработана подсистема мониторинга и определения уровня стресса сотрудников, обеспечивающая ввод и хра-

нение физиологических показателей и результатов психологических методик, а также автоматическое формирование итогового заключения с поясняющими факторами риска и рекомендациями по дальнейшей проверке. Реализованные функции направлены на упрощение регулярного наблюдения за состоянием сотрудников и повышение оперативности выявления случаев, требующих дополнительного внимания.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Михайлова, Е. В. Методы диагностики стресса в психологии и медицине / Е. В. Михайлова. – Санкт-Петербург: Питер, 2021. – 224 с.
2. Щербатых, Ю. В. Психология стресса и методы коррекции / Ю. В. Щербатых. – Санкт-Петербург: Питер, 2012. – 256 с.
3. Радченко, М. Г. 1С:Предприятие 8.3. Практическое пособие разработчика. Примеры и типовые приёмы / М. Г. Радченко, Е. Ю. Хрусталева. – Москва: 1С-Публишинг, 2019. – 964 с.
4. Рашка, С. Python и машинное обучение / С. Рашка, В. Мирджалили. – Москва: ДМК Пресс, 2020. – 848 с.
5. Николенко, С. Глубокое обучение. Погружение в мир нейронных сетей / С. Николенко, А. Кадури, Е. Архангельская. – Санкт-Петербург: Питер, 2018. – 480 с.

© В. Ю. Сергеев, Д. Ю. Смирнов, 2026

В. А. Сидун^{1✉}, С. Ю. Кацко¹

Разработка информационного веб-ресурса для ООО «ПСК “Стройкаркас”»

¹Сибирский государственный университет геосистем и технологий,
г. Новосибирск, Российская Федерация
e-mail: vs_sidun@mail.ru

Аннотация. В статье представлены основные результаты разработки информационного веб-ресурса для ООО «ПСК “Стройкаркас”», специализирующегося на изготовлении дверей и лестниц из натурального массива дерева. Выполнен анализ предметной области, в котором рассмотрены современные подходы к созданию веб-сайтов, включая использование конструкторов и разработку с нуля. Для создания веб-ресурса были использованы технологии HTML5, CSS3, JavaScript, а также библиотеки Swiper.js, IMask.js и сервис EmailJS. Разработанный веб-ресурс предоставляет пользователю круглосуточный доступ к каталогу продукции, портфолио реализованных проектов, а также отправить структурированную заявку на электронную почту предприятия.

Ключевые слова: веб-ресурс, адаптивный дизайн, HTML5, CSS3, JavaScript, EmailJS, Swiper.js

V. A. Sidun^{1✉}, S. Yu. Katsko¹

Development of an information web resource for LLC «PSK “Stroykarkas”»

¹Siberian State University of Geosystems and Technologies, Novosibirsk, Russian Federation
e-mail: vs_sidun@mail.ru

Abstract. This article presents the main results of the development of a web resource for PSK “Stroykarkas” LLC, a company specializing in the manufacture of doors and staircases made of solid wood. A domain analysis was conducted, examining modern approaches to website creation, including the use of website builders and development from scratch. To create the web resource, HTML5, CSS3, and JavaScript technologies were used, as well as the Swiper.js and IMask.js libraries and the EmailJS service. The developed web resource provides users with round-the-clock access to the product catalog and a portfolio of completed projects, as well as the ability to send a structured inquiry via the company’s email.

Keywords: web resource, responsive design, HTML5, CSS3, JavaScript, EmailJS, Swiper.js

Введение

Наличие официального веб-сайта является необходимым условием для привлечения клиентов и информирования о продукции предприятий малого и среднего бизнеса. Веб-ресурс обеспечивает демонстрацию портфолио, публикацию актуальной информации и прием заявок в автоматическом режиме [1].

На момент начала разработки у ООО «ПСК “Стройкаркас”» отсутствовал официальный веб-сайт. Это ограничивало его присутствие в сети Интернет и затрудняло удаленное знакомство заказчиков с ассортиментом продукции. Отсут-

ствие структурированного каталога и формы обратной связи приводило к тому, что прием заявок осуществлялся только по телефону или при личном визите в офис.

Актуальность работы обусловлена необходимостью создания корпоративного веб-ресурса с уникальным дизайном, адаптивным интерфейсом и высокой скоростью загрузки. В отличие от конструкторов сайтов [2], индивидуальная разработка позволяет реализовать специфику деревообрабатывающего производства без ограничений, накладываемых готовыми шаблонами. Цель работы – проектирование и программная реализация информационного веб-ресурса для ООО «ПСК «Стройкаркас»», обеспечивающего создание цифрового пространства предприятия и получение обратной связи от клиентов.

Методы и технологии

Разработка веб-ресурса выполнялась в соответствии с циклом создания веб-продукта, включающим проектирование структуры и дизайна, frontend-разработка, интеграция сторонних сервисов, тестирование и размещение на хостинге [1].

Для создания макетов выбрана программа Figma, представляющая собой бесплатное веб-приложение, работающее в браузере и поддерживающее одновременное редактирование. Инструмент обладает средствами создания адаптивных сеток, компонентов и прототипов с анимацией переходов [3].

На основе анализа деятельности предприятия разработана цветовая палитра, включающая шесть базовых оттенков: молочно-белый #F6F8F5 - основной фон; светло-бежевый #E8EDE6 - вторичные фоновые блоки; приглушенно-серый #D4DDD2 - разделители и границы; насыщенный темно-зеленый #1E2B1D- акцентные блоки и кнопки; оливковый #6B7F6A - интерактивные элементы; мягкий серо-зеленый #8A9E88 - подписи и дополнительный текст.

Логотип выполнен в векторном формате SVG. Графическое решение объединяет два основных направления деятельности предприятия – двери и лестницы. Для типографики подобраны два шрифтовых семейства: Raleway (заголовки) и Plus Jakarta Sans (основной текст).

На основе утвержденного дизайна создана карта навигации, включающая пять основных страниц: главная, «О нас», «Продукция», «Портфолио», «Контакты». Для каждой страницы разработан фрейм шириной 1440 пикселей с разметочной сеткой из 12 колонок. Сформирована библиотека локальных стилей и система мастер-компонентов для повторяющихся элементов (кнопки, карточки товара, теги) с вариантами состояний: статическое, при наведении курсора, активное.

Для реализации клиентской части выбраны технологии HTML5, CSS3 и ванильный JavaScript [4]. Отказ от использования фреймворков (React, Vue, Angular) обусловлен необходимостью минимизации объема загружаемого кода и времени до первой отрисовки. Сложная динамическая логика в проекте отсутствует, что делает применение фреймворков избыточным.

Слайдер отзывов реализован с использованием библиотеки Swiper.js версии 11 [5]. Библиотека обеспечивает адаптивную настройку количества отображаемых карточек (3 на десктопе, 2 на планшете, 1 на мобильном устройстве), под-

держку touch-жестов и плавную анимацию переключения. Инициализация слайдера выполняется после загрузки DOM с проверкой наличия соответствующего элемента на странице.

Для фильтрации каталога продукции на странице «Продукция» применен JavaScript-скрипт. Карточкам товаров присвоен атрибут `data-category`. При нажатии на кнопку фильтра скрипт перебирает все карточки и скрывает те, чья категория не совпадает с выбранной. Плавность анимации обеспечена CSS-переходами свойств `opacity` и `transform`.

Для отображения местоположения офиса предприятия использован Yandex Maps API версии 2.1 [6]. Карта инициализируется в отдельном файле `map-init.js`. Координаты центра установлены по адресу: г. Томск, ул. Ракетная, д. 4 (56.517063, 85.012453). Добавлена метка с балуном, отключена возможность масштабирования колесиком мыши.

Для отправки заявок на электронную почту предприятия выбран сервис EmailJS [7]. Решение не требует разработки собственного backend. Сервис подключается через SDK на frontend. EmailJS предоставляет бесплатный тариф на 200 писем в месяц, что покрывает потребности малого предприятия. На стороне сервиса настроен динамический шаблон письма, в который подставляются имя пользователя, телефон и текст сообщения.

Форма обратной связи включает три поля: «Ваше имя», «Телефон» и «Сообщение». Маска ввода телефонного номера реализована с помощью библиотеки `IMask.js` [8]. Маска автоматически форматирует вводимые цифры в шаблон `+7 (***) ***-**-**`. Перед отправкой выполняется валидация: проверка длины имени (не менее двух символов) и проверка завершенности маски телефона (свойство `masked.isComplete`). После успешной валидации кнопка отправки блокируется, данные передаются методом `emailjs.sendForm()`, форма очищается, и пользователь получает уведомление.

Адаптивность интерфейса реализована с помощью CSS-медиазапросов в файле `adaptive.css` [4]. Выделены три диапазона разрешений: до 768 пикселей (мобильные устройства), до 1024 пикселей (планшеты) и свыше 1440 пикселей (широкие мониторы). Для каждого диапазона изменяются следующие параметры: количество колонок сетки (с 12 до 4 или 8), направление компоновки блоков (с горизонтального на вертикальное), размеры шрифтов и отступов. На мобильных устройствах горизонтальная навигация заменяется кнопкой-бургером, декоративные изображения скрываются.

Для управления версиями кода использована система Git с удаленным репозиторием на платформе GitHub. Развертывание выполнено на платформе Vercel, связанной с репозиторием через OAuth-авторизацию. Vercel автоматически запускает сборку и публикацию обновлений при каждом изменении ветки `main`, выпускает TLS-сертификат и обеспечивает работу через глобальную CDN-сеть.

Результаты

В результате работы создан информационный веб-ресурс, доступный по адресу `dveri-iz-dereva.vercel.app`. Ресурс представляет собой многостраничное приложение (MPA), в котором реализованы следующие функции:

- отображение каталога продукции с фильтрацией по категориям (рис. 1);
- интерактивный слайдер отзывов с адаптивным количеством карточек (3/2/1);
- форма обратной связи с маской ввода телефона и валидацией (рис. 2, а);
- автоматическая отправка заявки на электронную почту предприятия через сервис EmailJS;
- интерактивная карта Yandex Maps с меткой офиса и отключенным масштабированием колесиком (рис. 2, б);
- адаптивная верстка под три диапазона разрешений (мобильные, планшеты, десктопы);
- мобильное меню-бургер с выдвигаемым оверлеем (рис. 3).

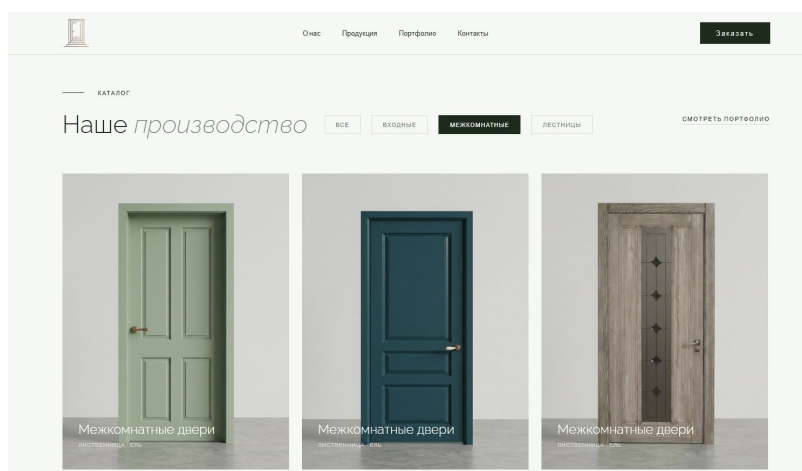


Рис. 1. Страница «Продукция»

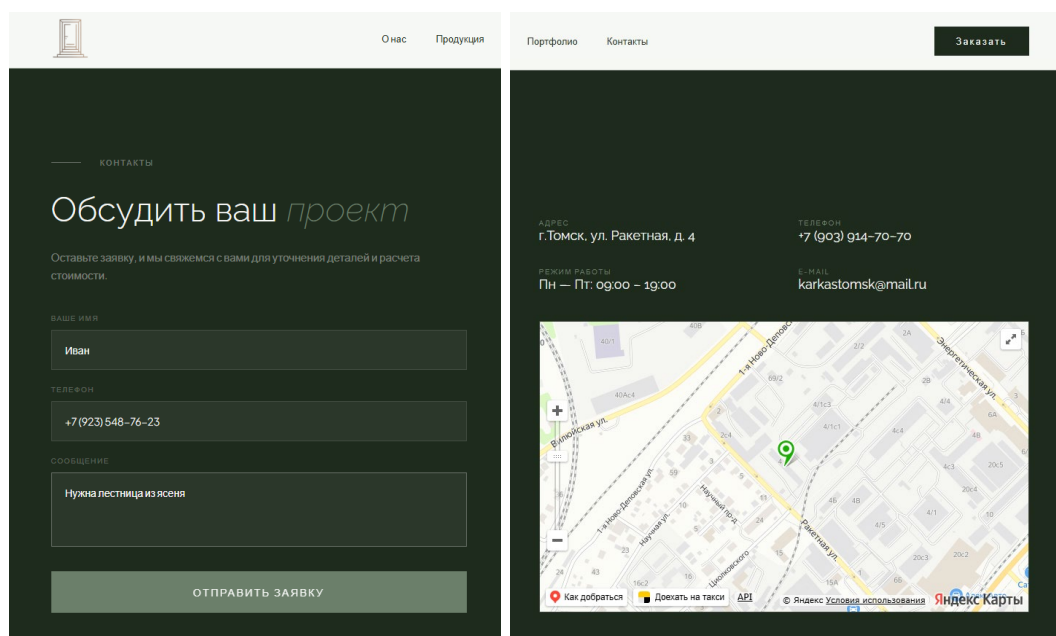


Рис. 2. Страница «Контакты»

а) форма обратной связи; б) интерактивная карта с меткой офиса

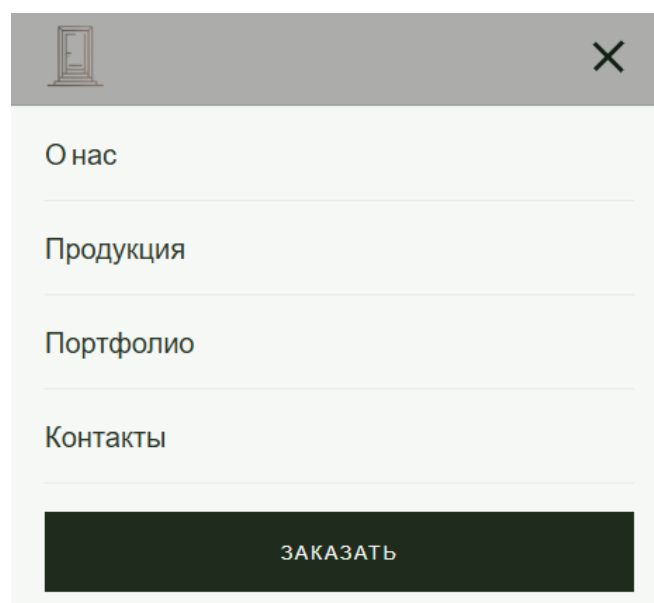


Рис. 3. Мобильный оверлей

Функциональное тестирование выявило три дефекта: неправильный порядок подключения скриптов Swiper.js (отображалась одна карточка вместо трех); отсутствие подключения файла adaptive.css на странице «О нас» (мобильное меню отображалось без стилей); ошибка в шаблоне EmailJS (поле «Сообщение» не попадало в письмо). Все дефекты устранены до публикации ресурса.

Кроссбраузерное тестирование выполнено в браузерах Google Chrome, Yandex Browser, Mozilla Firefox, Apple Safari и Microsoft Edge. Для каждого браузера проверялось отображение всех страниц, работа анимаций, корректность шрифтов и функционирование формы. Во всех браузерах ресурс отображается идентично, отклонений не выявлено.

Адаптивное тестирование выполнялось в двух режимах: эмуляция устройств в Chrome DevTools (разрешения 320, 375, 768, 1024, 1440 и 1920 пикселей) и проверка на реальных устройствах (смартфон 390 пикселей, ноутбук 1440 пикселей) (рис. 4, 5). Во всех случаях компоновка блоков соответствует адаптивным макетам, горизонтальная прокрутка отсутствует.

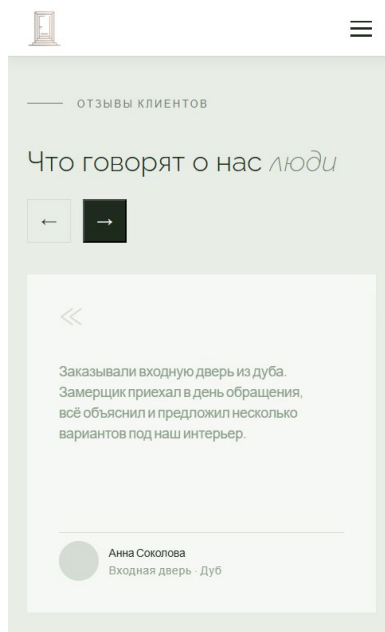


Рис. 4. Страница «О нас» мобильная версия (390px)

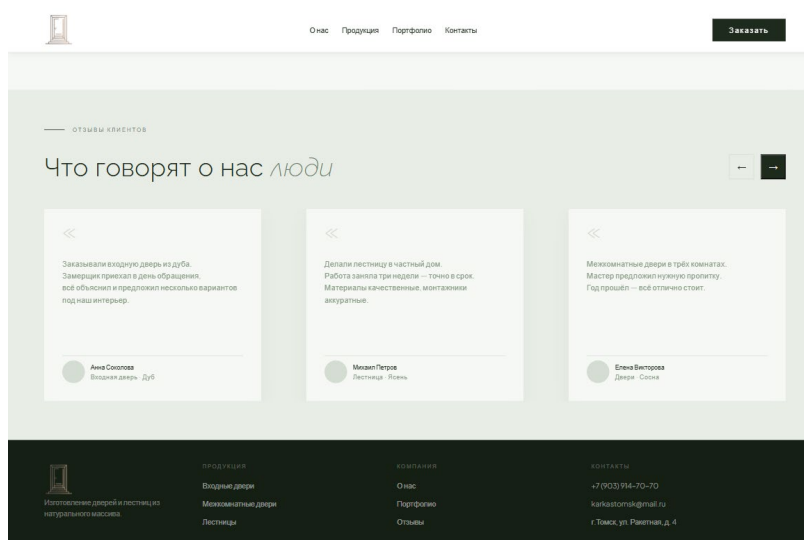


Рис. 5. Страница «О нас» десктопная версия (1440px)

Нагрузочное тестирование выполнено с помощью сервиса Google PageSpeed Insights. Для десктопной версии главной страницы получена оценка 96 баллов из 100 возможных (рис. 6, а), для мобильной версии – 71 балл (рис. 6, б). Высокие показатели обеспечены за счет оптимизации изображений, минимального количества сторонних скриптов и отсутствия избыточных CSS-правил.

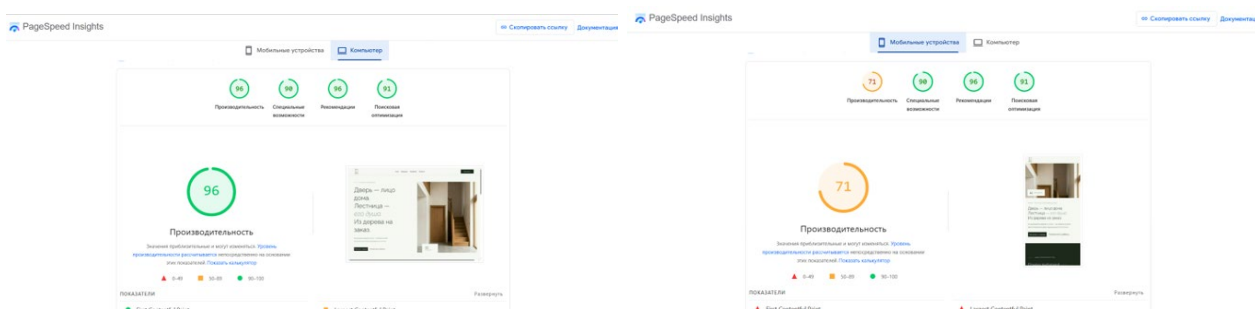


Рис. 6. Показатели Google PageSpeed Insights
а) для компьютеров; б) для мобильных устройств

Заключение

Технические требования реализованы в полном объеме. Результатом работы стал действующий информационный веб-ресурс, предназначенный для демонстрации каталога продукции, портфолио и приема заявок от клиентов.

Выбранная архитектура многостраничного приложения на нативном стеке HTML5, CSS3 и JavaScript продемонстрировала свою эффективность в ходе тестирования. Отказ от фреймворков обеспечил высокую скорость загрузки (0,9 с на десктопе, 1,4 с на мобильных устройствах) и независимость страниц.

На стороне клиента интерфейс реализует строгую валидацию вводимых данных. Для пользователя минимизируется необходимость ручного контроля формата номера телефона и заполнения обязательных полей. Данные функции реализуются библиотекой IMask.js и встроенной валидацией формы. Алгоритм отправки заявки использует сервис EmailJS, при этом процесс передачи данных осуществляется без разработки собственного backend.

Система продемонстрировала стабильность в ходе выполнения всех запланированных испытаний. Скорость работы соответствует установленным нормативам. Обеспечение базового уровня информационной безопасности реализовано посредством TLS-сертификата на стороне хостинга Vercel.

В перспективе приоритетными направлениями развития станут создание интерактивного калькулятора стоимости изделий, SEO-оптимизация для привлечения целевого трафика, интеграция с CRM-системами (Битрикс24, amoCRM) для автоматической фиксации заявок, а также внедрение капчи для защиты формы от спама.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Подчукаев, В. А. Теория информационных процессов и систем : учеб. пособие для вузов / В. А. Подчукаев. – Москва: Гардарики, 2007. – 207 с. – ISBN 5-8297-0297-5.
2. Конструкторы сайтов: минусы и плюсы. – Текст : электронный // RemOnline : блог. – 2023. – URL: <https://remonline.app/ru/blog/web-constructors> (дата обращения: 10.05.2026).
3. Figma: официальная документация компании. – Текст : электронный // Figma Inc. – 2016. – URL: <https://www.figma.com> (дата обращения: 10.04.2026).
4. HTML5, CSS3, JavaScript: документация разработчика MDN. – Текст : электронный // MDN Web Docs. – 2015. – URL: <https://developer.mozilla.org/ru/docs/Web> (дата обращения: 20.04.2026).

5. Swiper.js: официальное руководство по интеграции слайдеров. – Текст : электронный // Swiper. – 2014. – URL: <https://swiperjs.com> (дата обращения: 12.04.2026).
6. Yandex Maps API: справочное руководство разработчика. – Текст : электронный // Яндекс Технологии. – 2012. – URL: <https://yandex.ru/dev/maps/jsapi> (дата обращения: 15.04.2026).
7. EmailJS: официальная документация сервиса интеграции почты. – Текст : электронный // EmailJS. – 2019. – URL: <https://www.emailjs.com/docs> (дата обращения: 14.04.2026).
8. IMask.js: официальное руководство по валидации полей ввода. – Текст : электронный // IMask. – 2016. – URL: <https://imask.js.org> (дата обращения: 12.04.2026).

© В. А. Сидун, С. Ю. Кацко, 2026

Д. С. Строганов^{1✉}, С. Ю. Кацко¹

Мультиагентная система планирования группы БПЛА с нейросетевой оценкой конфигурации

¹Сибирский государственный университет геосистем и технологий,
г. Новосибирск, Российская Федерация
e-mail: stroganovdanila@outlook.com

Аннотация. В статье рассматривается мультиагентная система планирования полетов группы беспилотных летательных аппаратов (БПЛА), функционирующая на основе сети стационарных док-станций. Предложен подход, в котором агент планирования использует нейросетевую модель для оценки качества конфигурации группы дронов при выполнении задачи покрытия заданной территории. В отличие от классических методов многокритериальной оптимизации с фиксированной целевой функцией, оценка конфигурации осуществляется на основе обучаемой модели, учитывающей параметры зоны, характеристики дронов и контекст выполнения миссии. Описана архитектура системы, включающая агента планирования, агентов станций и агентов дронов, а также механизм распределения задач и формирования маршрутов. Рассмотрен сценарий применения системы для мониторинга аграрных угодий. Полученные результаты моделирования демонстрируют сокращение времени планирования и повышение полноты покрытия по сравнению с базовым подходом с линейной целевой функцией.

Ключевые слова: мультиагентные системы, БПЛА, планирование полетов, распределение задач, оценка конфигурации, многокритериальная оптимизация, нейросетевые модели, док-станции

D. S. Stroganov^{1✉}, S. Yu. Katsko¹

Multi-Agent system for UAV swarm planning with neural network-based configuration evaluation

¹Siberian State University of Geosystems and Technologies, Novosibirsk, Russian Federation
e-mail: stroganovdanila@outlook.com

Abstract. This paper presents a multi-agent system for planning the operation of a group of unmanned aerial vehicles (UAVs) based on a network of stationary docking stations. An approach is proposed in which a planning agent employs a neural network model to evaluate the quality of a UAV swarm configuration for covering a target area. In contrast to classical multi-criteria optimization methods with a fixed objective function, the configuration evaluation is performed using a learned model that takes into account the characteristics of the target area, UAV parameters, and mission context. The system architecture is described, including a planning agent, station agents, and UAV agents, as well as mechanisms for task allocation and route generation. A use case involving agricultural land monitoring is presented. Simulation results demonstrate a reduction in planning time and an improvement in coverage compared to a baseline approach using a linear objective function.

Keywords: multi-agent systems, UAV, flight planning, task allocation, configuration evaluation, multi-criteria optimization, neural network models, docking stations

Введение

В последнее десятилетие беспилотные летательные аппараты (БПЛА) стали одним из ключевых инструментов дистанционного мониторинга территорий.

Широкий спектр применений – от аграрного контроля состояния посевов до инспекции линейной инфраструктуры и экстренного реагирования на чрезвычайные ситуации – предъявляет все более высокие требования к автономности, гибкости и эффективности систем управления группами БПЛА. Традиционный подход, при котором оператор вручную управляет одним или несколькими дронами с мобильного устройства, имеет принципиальные ограничения: необходимость физического присутствия на месте, невозможность одновременного контроля нескольких аппаратов и крайне длительное время реакции при внезапном возникновении нештатной ситуации. Так, при обнаружении очага болезни посевов или начала пожара время развертывания ручной системы может составлять 30–60 минут, тогда как автономная система на основе сети док-станций обеспечивает взлет дрона в течение 15 секунд после поступления сигнала тревоги.

Новый подход предполагает развертывание стационарных доковых станций, оснащенных средствами связи (SIM-карта, спутниковый канал) и системой автоматической зарядки дронов. Каждая станция обслуживает несколько БПЛА, поддерживает их в постоянной готовности и может управлять ими по радиоканалу на расстоянии до 25 км [1]. При объединении нескольких таких станций образуется распределенная система, способная автономно покрывать территории площадью в сотни квадратных километров. Основной нерешенной задачей при таком подходе является эффективное распределение заданий между дронами с учетом их индивидуальных ресурсных ограничений: уровня заряда батарей, угла обзора камер, текущей высоты, а также необходимости избегать пересечений маршрутов между аппаратами разных станций. Ручное решение этой задачи в реальном времени практически невозможно, особенно при числе дронов более десяти.

Целью настоящей работы является разработка интеллектуальной мультиагентной системы, в которой агент планирования автоматически, без вмешательства оператора, определяет оптимальный состав группы БПЛА и строит маршруты для полного покрытия заданной территории с учетом всех ресурсных ограничений.

Краткий обзор существующих подходов

Задача распределенного планирования маршрутов группы БПЛА относится к классу задач многоагентного распределения ресурсов и является NP-трудной при большом числе аппаратов [2]. В литературе рассматриваются следующие основные классы методов.

Классические алгоритмы маршрутизации (A^* , алгоритм Дейкстры, методы ветвей и границ) гарантируют оптимальность решения, однако требуют полной и заранее известной информации о среде. Они плохо масштабируются с ростом числа агентов и не адаптируются к динамическим изменениям – например, к внезапной разрядке батареи одного из дронов в процессе выполнения миссии.

Эволюционные и роевые методы (генетические алгоритмы, метод роя частиц, алгоритм муравьиных колоний) обеспечивают приемлемое качество решений для задач большой размерности, однако требуют значительного вычисли-

тельного времени и тонкой настройки гиперпараметров для каждого конкретного сценария [3].

Мультиагентные системы (МАС) с фиксированными правилами взаимодействия обеспечивают высокую гибкость и масштабируемость: каждый агент действует автономно и обменивается сообщениями с соседними агентами [4, 5]. Однако фиксированные правила задаются разработчиком вручную и не улучшаются в процессе эксплуатации. Кроме того, в большинстве существующих систем целевая функция оптимизации имеет явный вид с заранее заданными весовыми коэффициентами, что требует ручной настройки под конкретный сценарий [6].

Методы обучения с подкреплением (Reinforcement Learning) позволяют агентам обучаться оптимальной политике управления в ходе взаимодействия со средой [7]. Однако их применение в группах БПЛА осложнено высокой размерностью пространства состояний и необходимостью длительного обучения перед развертыванием. Предлагаемый в настоящей работе подход занимает промежуточное положение: в рамках МАС-архитектуры агент планирования использует нейронную сеть, обученную на исторических данных реальных миссий, для замены ручной целевой функции. Это сочетает гибкость МАС с адаптивностью методов машинного обучения [8].

Постановка задачи

Рассмотрим формальную постановку задачи. Пусть на исследуемой территории расположено n док-станций $S = \{s_1, s_2, \dots, s_n\}$. К каждой станции s_i подключено множество БПЛА $D_i = \{d_{i1}, d_{i2}, \dots, d_{ij}\}$. Каждый дрон d_{ij} характеризуется вектором параметров состояния:

- b_{ij} – уровень заряда батареи (0–100 %);
- h_{ij} – рабочая высота полета;
- φ_{ij} – угол обзора камеры, определяющий ширину полосы покрытия;
- r_{ij} – техническая готовность (доступен / на зарядке / в ремонте).

Оператор задает целевую зону G в виде полигона на карте и, при необходимости, дедлайн выполнения миссии T_{max} . Требуется:

- выбрать подмножество $D^* \subseteq D$ из общего пула доступных дронов;
- построить для каждого дрона из D^* маршрут, обеспечивающий полное покрытие зоны G без слепых зон и без пересечений маршрутов соседних аппаратов;
- обеспечить выполнение ресурсных ограничений: заряд батареи каждого дрона должен быть достаточен для выполнения его подзадачи и возврата на станцию.

Задача является многокритериальной, поскольку оператор может преследовать конкурирующие цели: минимизировать количество задействованных дронов N (экономия ресурса), минимизировать общее время выполнения миссии T и максимизировать полноту покрытия C (долю площади зоны G , охваченной

маршрутами). В классическом подходе эти критерии агрегируются в единую взвешенную целевую функцию вида:

$$F = \min(\alpha \cdot N + \beta \cdot T + \gamma \cdot (1 - C)), \quad \alpha + \beta + \gamma = 1,$$

где α, β, γ – весовые коэффициенты, задаваемые оператором вручную перед каждой миссией. Основным недостатком такого подхода является необходимость экспертной настройки весов: неправильный выбор коэффициентов приводит к неоптимальным решениям. Кроме того, данная функция является линейной и не способна моделировать нелинейные взаимодействия между параметрами.

В предлагаемом подходе весовые коэффициенты и, более широко, оценка качества конфигурации заменяются нейронной сетью, обученной на исторических данных миссий, что устраняет оба указанных недостатка.

Описание предлагаемого подхода

Система построена по трехуровневой иерархической архитектуре МАС и включает три типа агентов (рис. 1) [4].

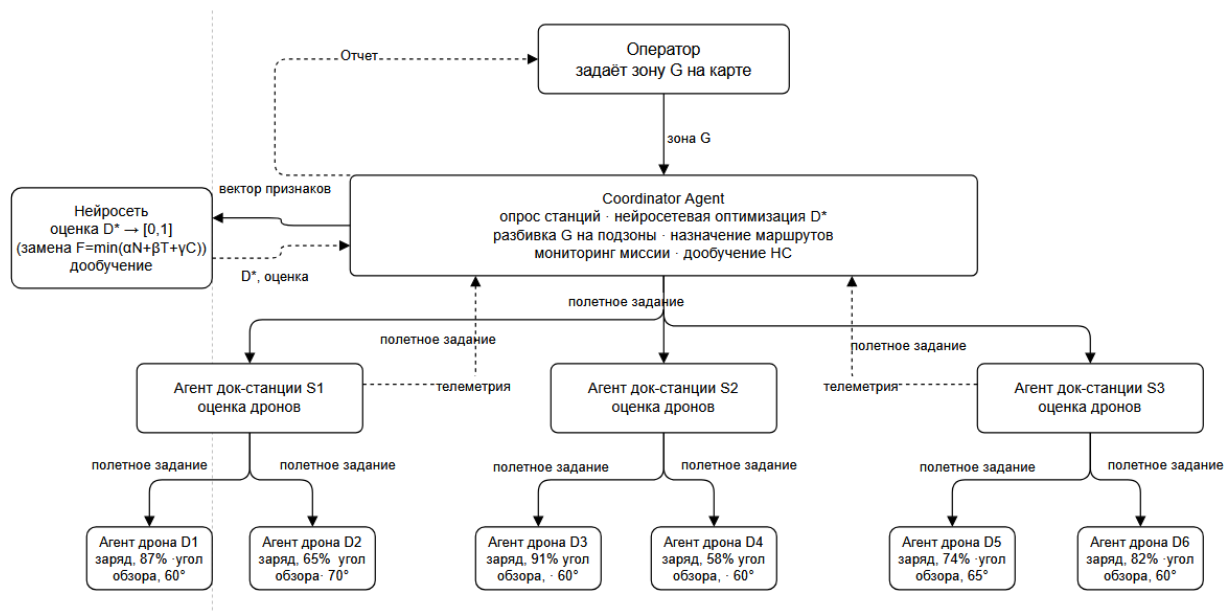


Рис. 1. Концептуальная модель системы

Агент планирования (Coordinator Agent) – единственный агент верхнего уровня. Он получает от оператора описание зоны G и вычисляет ее геометрические параметры (площадь, форму, протяженность). Затем рассылает запросы агентам всех станций, собирает информацию о доступных дронах и их параметрах, после чего с помощью нейронной сети формирует оптимальный план: выбирает состав группы D^* и назначает каждому дрону его подзону.

Агент станции (Station Agent) – по одному на каждую док-станцию. Агент станции непрерывно опрашивает подключенные к ней дроны, актуализирует их параметры состояния и отвечает на запросы агента планирования. При получении

нии задания от координатора агент станции транслирует его соответствующему агенту дрона.

Агент дрона (UAV Agent) – по одному на каждый БПЛА. Отвечает за автономное выполнение назначенного маршрута: удерживает заданную высоту и курс, непрерывно передает телеметрию и снимки на свою станцию, а при критическом снижении уровня заряда самостоятельно прерывает выполнение задания и возвращается на зарядку.

Ключевым элементом предлагаемого подхода является замена классической взвешенной целевой функции нейронной сетью (НС), обученной предсказывать качество произвольной конфигурации роя для заданной миссии. Входом НС служит вектор признаков, включающий:

- параметры зоны G : площадь, периметр, расстояние до ближайших станций;
- параметры предложенной конфигурации D^* : количество дронов N , их индивидуальные заряды $\{b_{ij}\}$, высоты $\{h_{ij}\}$ и углы обзора $\{\varphi_{ij}\}$;
- контекстные параметры: время суток, наличие дедлайна T_{max} .

Выходом НС является оценка качества конфигурации – скалярное значение в диапазоне $[0, 1]$, отражающее ожидаемую эффективность данного плана относительно всех трех критериев одновременно. Агент планирования перебирает несколько кандидатных конфигураций и выбирает ту, для которой НС предсказывает наибольшее качество. Существенным преимуществом такого подхода является способность НС улавливать нелинейные зависимости, недоступные линейной целевой функции. Например, в процессе обучения сеть может выявить, что при площади зоны свыше 50 га и наличии дедлайна конфигурация из двух дронов с зарядом 60 % статистически превосходит конфигурацию из одного дрона с зарядом 90 %, поскольку параллельное покрытие двух подзон оказывается быстрее и надежнее. Обучение НС осуществляется в два этапа. На первом этапе используются синтетические данные, генерируемые в среде симуляции: для случайных зон и случайных конфигураций дронов запускается физическая модель полета, по результатам которой вычисляются фактические значения N , T и C . На втором этапе сеть дообучается на реальных данных эксплуатации системы по мере накопления истории миссий.

После того как агент планирования определил состав группы D^* и назначил каждому дрону подзону, выполняется построение маршрутов внутри каждой подзоны. Используется модифицированный метод параллельных полос (lawnmower pattern): ширина полосы определяется индивидуально для каждого дрона исходя из его высоты и угла обзора камеры, что обеспечивает точное покрытие без перекрытий между соседними полосами. Для предотвращения пересечений маршрутов дронов разных станций применяется временная координация: агент планирования назначает каждому дрону временной слот начала работы в зоне пересечения подзон таким образом, чтобы аппараты находились в этой зоне в разное время.

Пример применения: мониторинг аграрных угодий

Рассмотрим следующий сценарий. Агрохозяйство площадью 320 га разделено на три участка неправильной формы. На территории расположены три док-станции: S1 (северо-запад), S2 (центр), S3 (юго-восток). К каждой станции подключены по четыре дрона с параметрами, представленными в таблице 1.

Таблица 1. Параметры дронов на момент начала миссии

Дрон	Станция	Заряд, %	Высота, м	Угол камеры, °	Статус
D1-1	S1	87	50	60	Готов
D1-2	S1	23	50	60	Зарядка
D1-3	S1	65	60	70	Готов
D2-1	S2	91	50	60	Готов
D2-2	S2	58	50	60	Готов
D3-1	S3	74	55	65	Готов
D3-2	S3	82	50	60	Готов

Далее, оператор рисует на карте интерфейса целевую зону, охватывающую все три участка хозяйства (суммарная площадь 320 га), и запускает планирование. Дедлайн не задается – система работает в режиме качественного покрытия. Агент планирования запрашивает агентов трех станций. Те опрашивают своих дронов и возвращают параметры состояния. Агент планирования исключает дрон D1-2 как недоступный (на зарядке) и формирует набор кандидатных конфигураций из оставшихся шести дронов. Нейронная сеть оценивает несколько конфигураций. Конфигурация из трех дронов {D1-1, D2-1, D3-2} с высоким зарядом получает оценку 0,71, однако НС, обученная на исторических данных, предсказывает более высокую эффективность (0,84) для конфигурации из четырех дронов {D1-3, D2-1, D2-2, D3-1}: несмотря на меньший средний заряд, они обеспечивают более равномерное географическое распределение покрытия и сокращают общее время миссии благодаря параллельной работе. Агент планирования принимает четырехдронную конфигурацию, разбивает зону на четыре подзоны с учетом ширины полосы каждого дрона (определяемой высотой и углом камеры) и назначает маршруты. Для участка пересечения подзон дронов D2-1 и D3-1 применяется временная координация: D3-1 начинает работу в зоне пересечения на 4 минуты позже D2-1.

Система завершает планирование за 2,3 секунды с момента запроса оператора. Дроны одновременно взлетают со своих станций и выполняют маршруты. Итоговые показатели миссии приведены в таблице 2.

Таблица 2. Сравнение подходов к планированию

Показатель	Ручное управление	МАС (линейная функция)	МАС (нейросеть)
Время планирования, с	~1200	4,1	2,3
Покрытие территории, %	~70	96	99
Время миссии, мин.	~180	52	41
Задействовано дронов, шт.	1	3	4
Участие оператора	Постоянное	Настройка весов	Только задание зоны

Полученные результаты демонстрируют, что предложенный подход обеспечивает высокую полноту покрытия территории (99 %) при сокращении времени выполнения миссии на 21 % по сравнению с линейной многокритериальной функцией и без какой-либо ручной настройки весовых коэффициентов.

Заключение

В работе представлена мультиагентная система распределенного планирования полетов группы БПЛА, функционирующая на основе сети стационарных док-станций. Предложена трехуровневая архитектура МАС (агент планирования – агент станции – агент дрона), в которой ключевым элементом является нейросетевой модуль агента планирования. Основной вклад работы состоит в применении нейросетевой модели для оценки конфигурации роя, позволяющей учитывать нелинейные взаимодействия между параметрами дронов, которые не моделируются линейной функцией. Проведенный вычислительный эксперимент на сценарии мониторинга аграрных угодий площадью 320 га подтвердил работоспособность предложенного подхода: покрытие территории составило 99 %, время выполнения миссии сократилось на 21 % по сравнению с базовым вариантом МАС, а участие оператора сведено к минимуму – указанию зоны мониторинга на карте. Направлениями дальнейших исследований являются интеграция модуля обнаружения препятствий в реальном времени, разработка механизма динамического перепланирования маршрутов при выходе одного из дронов из строя в процессе миссии, а также апробация системы на реальном аппаратном стенде.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. DJI Enterprise. DJI Dock 3 Product Specifications. – 2024. URL: <https://enterprise.dji.com/dock-3> (дата обращения: 15.04.2025).
2. Agatz N., Bouman P., Schmidt M. Optimization Approaches for the Traveling Salesman Problem with Drone // Transportation Science. – 2018. – Vol. 52. – №. 4. – P. 965–981.
3. Курейчик В. М., Кажаров А. А. Использование роевого интеллекта в решении NP-трудных задач // Известия ЮФУ. Технические науки. 2011. № 7 (120). С. 30-37.

4. Городецкий В. И. Самоорганизация и многоагентные системы. I. Модели многоагентной самоорганизации / В. И. Городецкий // Известия Российской академии наук. Теория и системы управления. - 2012. - № 2. - С. 92-120.
5. Dorri A., Kanhere S.S., Jurdak R. Multi-Agent Systems: A Survey // IEEE Access. – 2018. – Vol. 6. – P. 28573–28593.
6. Queralta J.P. et al. Collaborative Multi-Robot Search and Rescue: Planning, Coordination and Communication // Proceedings of the IEEE. – 2020. – Vol. 108. – №. 10. – P. 1572–1597.
7. Mnih V. et al. Human-level control through deep reinforcement learning // Nature. – 2015. – Vol. 518. – P. 529–533.
8. Кольчерин Д.В., Печеркин С.А. Применение мультиагентных систем, основанных на нейронных сетях // Проблемы современной науки и образования. 2017. № 23. С. 6-10.

© Д. С. Строганов, С. Ю. Кацко, 2026

А. Т. Ташиханов^{1✉}, С. Ю. Кацко¹

Разработка веб-приложения для автоматизации выдачи микрозаймов с функцией кредитного скоринга

¹Сибирский государственный университет геосистем и технологий,
г. Новосибирск, Российская Федерация
e-mail: masakatam@gmail.com

Аннотация. В статье представлены результаты разработки веб-приложения «MicroLoan», предназначенного для автоматизации процесса выдачи микрозаймов и оценки кредитоспособности клиентов. Выполнен анализ предметной области, выявлены недостатки существующих информационных систем для микрофинансовых организаций (высокая стоимость, закрытость алгоритмов скоринга). Обоснован выбор балльной линейной модели кредитного скоринга с пятью факторами (доход, показатель долговой нагрузки, кредитная история, тип занятости, возраст). Для реализации приложения выбраны технологии PHP 8.2, MySQL 5.7, HTML5/CSS3/JavaScript, среда разработки XAMPP. Спроектирована реляционная база данных, включающая таблицы users, clients, loan_applications, loans, payments, scoring_settings. Реализованы модули аутентификации, подачи заявки, расчёта скоринга, панели администратора и отправки уведомлений через сервис EmailJS. Проведённое тестирование (13 тест-кейсов) подтвердило соответствие функциональным и нефункциональным требованиям. Разработанное приложение может использоваться в учебном процессе и в малых микрофинансовых организациях.

Ключевые слова: микрозаймы, веб-приложение, кредитный скоринг, PHP, MySQL, автоматизация, балльная модель

A. T. Tashkhanov^{1✉}, S. Yu. Katsko¹

Development of a web application for microloan issuance automation with credit scoring function

¹Siberian State University of Geosystems and Technologies, Novosibirsk, Russian Federation
e-mail: masakatam@gmail.com

Abstract. The article presents the results of the development of the "MicroLoan" web application designed to automate the process of microloan issuance and creditworthiness assessment of clients. The analysis of the subject area revealed the disadvantages of existing information systems for microfinance organizations (high cost, closed scoring algorithms). The choice of a point-based linear credit scoring model with five factors (income, debt-to-income ratio, credit history, employment type, age) is justified. The technologies PHP 8.2, MySQL 5.7, HTML5/CSS3/JavaScript, and the XAMPP development environment were chosen for implementation. A relational database including tables users, clients, loan_applications, loans, payments, scoring_settings was designed. The modules of authentication, application submission, scoring calculation, administrator panel, and notification sending via the EmailJS service were implemented. The performed testing (13 test cases) confirmed compliance with functional and non-functional requirements. The developed application can be used in the educational process and in small microfinance organizations.

Keywords: microloans, web application, credit scoring, PHP, MySQL, automation, point model

Введение

Микрофинансовый сектор Российской Федерации демонстрирует устойчивый рост, однако обработка заявок на микрозаймы во многих организациях до сих пор выполняется вручную или с использованием дорогостоящих информационных систем, алгоритмы скоринга в которых часто закрыты и не поддаются адаптации [1, 2]. Существующие CRM-системы («Магус», «FintechOffice») имеют высокую стоимость лицензий, избыточный функционал и низкую гибкость настройки [3]. Это обуславливает актуальность разработки собственного веб-приложения, сочетающего простоту использования, прозрачность модели оценки рисков и низкую стоимость внедрения.

Целью работы является создание веб-приложения «MicroLoan», автоматизирующего процесс подачи заявок, расчёт кредитного скоринга и управление клиентской базой. В статье описываются архитектура, основные модули, результаты тестирования и оценка эффективности.

Исследование предметной области

Микрофинансовые организации (МФО) в Российской Федерации предоставляют микрозаймы физическим и юридическим лицам на сумму до 500 тыс. руб. и 1 млн руб. соответственно [1]. Деятельность МФО регулируется Федеральным законом № 151-ФЗ, а также требованиями Банка России по расчёту показателя долговой нагрузки (ПДН) и ограничению предельной задолженности [2].

Анализ существующих информационных систем, используемых в МФО, выявил их основные недостатки. Коммерческие CRM-системы («Магус», «FintechOffice») имеют высокую стоимость лицензий, закрытые алгоритмы скоринга и сложность адаптации под конкретные бизнес-процессы [3]. Универсальные системы («Битрикс24») требуют значительных доработок, так как не содержат встроенных финансовых алгоритмов.

Для оценки кредитоспособности заёмщиков в автоматизированных системах применяются скоринговые модели – линейные, логистическая регрессия, деревья решений [6]. Наиболее простой и интерпретируемой является балльная линейная модель, не требующая обучения на больших объёмах исторических данных и легко реализуемая на языке PHP.

Методы и технологии

Для реализации выбран стек технологий, обеспечивающий бесплатность, кроссплатформенность и высокую производительность: серверная часть – PHP 8.2 (с использованием PDO для работы с MySQL), клиентская часть – HTML5, CSS3, JavaScript, среда локальной разработки – XAMPP (Apache, MySQL, phpMyAdmin). Для отправки уведомлений о новых заявках использован сервис EmailJS, позволяющий отправлять письма напрямую из браузера без настройки SMTP-сервера [4].

Модель кредитного скоринга. Оценка кредитоспособности реализована на основе балльной линейной модели. Выделены пять факторов которые отобра-

жены в таблице 1, каждый оценивается по шкале от 0 до максимального балла. Итоговый балл вычисляется по формуле:

$$\text{Score} = \sum_{i=1}^5 w_i \cdot x_i,$$

где w_i – вес i -го фактора, x_i – количество баллов, начисляемых за соответствующий фактор. Максимальный балл – 100. Пороговые значения: ≥ 70 – «одобрить», 50–69 – «на рассмотрении», < 50 – «отказать».

Таблица 1. Факторы кредитоспособности и их веса

Фактор	Градация	Балл	Вес w_i
Ежемесячный доход	>50 000 руб.	30	0,30
	30 000–50 000 руб.	20	
	15 000–30 000 руб.	10	
	<15 000 руб.	0	
Показатель долговой нагрузки (ПДН)	<30%	25	0,25
	30%–50%	15	
	>50%	0	
Кредитная история	отличная	25	0,25
	хорошая	15	
	средняя	10	
	плохая	0	
Тип занятости	постоянная работа	15	0,15
	временная / самозанятый	8	
	безработный	0	
Возраст	25–55 лет	5	0,05
	18–24 или 56–65 лет	2	
	>65 лет	0	

База данных. Спроектирована реляционная база данных, включающая шесть таблиц: users (учётные записи), clients (данные заёмщиков), loan_applications (заявки), loans (выданные займы), payments (график платежей) и scoring_settings (настройки скоринга). Все таблицы нормализованы (3НФ), используются внешние ключи с правилами каскадного удаления, движок InnoDB, кодировка utf8mb4.

Результаты

Разработано веб-приложение «MicroLoan», включающее следующие модули:

Публичная форма подачи заявки – содержит поля для ввода личных, паспортных и финансовых данных, суммы и срока займа. Реализована клиентская и серверная валидация: сумма займа от 1000 до 50 000 руб., срок от 5 до 30 дней. После отправки данные сохраняются в БД, вызывается модуль скоринга, резуль-

тат отображается пользователю, и одновременно через EmailJS отправляется уведомление на почту организации.

Модуль кредитного скоринга – класс `ScoringModel` с методом `calculate()`, возвращающим балл и рекомендацию. Веса факторов и пороговые значения могут быть изменены администратором через таблицу `scoring_settings` без изменения кода.

Панель администратора – включает форму входа, страницу списка клиентов с CRUD-операциями и страницу обработки заявок. Все запросы к БД выполняются через подготовленные выражения PDO, что гарантирует защиту от SQL-инъекций. Вывод данных экранируется функцией `htmlspecialchars()` для предотвращения XSS-атак.

Тестирование. Проведено тестирование по 13 тест-кейсам, охватывающим регистрацию, подачу заявки, расчёт скоринга, изменение статуса заявки, отправку email, защиту от SQL-инъекций и XSS, адаптивность интерфейса. Все тест-кейсы выполнены успешно. Выявлен один не критичный дефект (отсутствие значения по умолчанию для типа занятости) – устранён. Нагрузочное тестирование в Apache JMeter (10 одновременных пользователей, 50 запросов) показало среднее время ответа 1,2 секунды, что соответствует требованию (менее 3 секунд).

Заключение

В ходе работы разработано веб-приложение «MicroLoan», автоматизирующее процесс выдачи микрозаймов и оценки кредитоспособности клиентов на базе балльной скоринговой модели. Приложение реализовано на PHP 8.2 с использованием MySQL и современного веб-интерфейса. Результаты тестирования подтвердили корректность работы, производительность и безопасность. Дальнейшее развитие может включать интеграцию с API БКИ, внедрение электронной подписи и разработку мобильного клиента.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. О микрофинансовой деятельности и микрофинансовых организациях [Электронный ресурс] : Федеральный закон от 02.07.2010 № 151-ФЗ. – URL: <https://base.garant.ru/12180378/> (дата обращения: 10.05.2026).
2. О потребительском кредите (займе) [Электронный ресурс] : Федеральный закон от 21.12.2013 № 353-ФЗ. – URL: https://www.consultant.ru/document/cons_doc_LAW_155986/ (дата обращения: 10.05.2026).
3. Кокорев, А. С. Отечественные и зарубежные методики оценки кредитоспособности заемщика / А. С. Кокорев. – Москва : Научный консультант, 2021. – 130 с.
4. EmailJS Documentation [Электронный ресурс]. – URL: <https://www.emailjs.com/docs/> (дата обращения: 12.05.2026).
5. Дронов, В. А. PHP, MySQL, HTML5 и CSS. Разработка современных динамических веб-сайтов / В. А. Дронов. – Санкт-Петербург : БХВ-Петербург, 2021. – 672 с.
6. Первозванский, А. А. Эконометрика. Кредитный скоринг : учебное пособие / А. А. Первозванский. – Санкт-Петербург : Изд-во СПбГЭУ, 2018. – 112 с.

7. ГОСТ 34.601-90. Информационная технология. Комплекс стандартов на автоматизированные системы. Автоматизированные системы. Стадии создания. – Москва : Стандартинформ, 2020. – 8 с.

8. ГОСТ 19.101-77. Единая система программной документации. Виды программ и программных документов. – Москва : Стандартинформ, 2021. – 7 с.

© А. Т. Таиханов, С. Ю. Кацко, 2026

Н. С. Федотов¹✉, П. Ю. Бугаков¹

Концептуальная архитектура интеллектуального ассистента для работы с распределенными источниками данных

¹Сибирский государственный университет геосистем и технологий,
г. Новосибирск, Российская Федерация
e-mail: Fedotov.N.S@yandex.ru

Аннотация. В статье предложена концептуальная архитектура персонального интеллектуального ассистента, предназначенного для поиска и обработки информации в распределенной пользовательской среде. Актуальность исследования состоит в том, что сообщения, письма, вложения и документы пользователя хранятся в разных цифровых каналах, тогда как единый поисковый интерфейс доступа к ним, как правило, отсутствует. Цель работы состоит в разработке архитектуры, которая способна интерпретировать запрос, выделять его параметры, выбирать релевантные источники данных, обращаться к внешним API и возвращать результат в структурированном виде. Методологической основой стали публикации по мультиагентным системам и смежным темам, а также официальная документация API сервисов, применимых в роли источников данных и интеллектуальных компонентов. В результате предложена трехагентная архитектура. Практическая ценность работы состоит в пригодности данной схемы для последующего создания программного прототипа персонального интеллектуального ассистента.

Ключевые слова: мультиагентная система, интеллектуальный ассистент, обработка естественного языка, информационный поиск, распределенные источники данных

N. S. Fedotov¹✉, P. Yu. Bugakov¹

Conceptual architecture of an intelligent assistant for distributed data source integration

¹Siberian State University of Geosystems and Technologies, Novosibirsk, Russian Federation
e-mail: Fedotov.N.S@yandex.ru

Abstract. The article proposes a conceptual architecture for a personal intelligent assistant designed to search for and process information in a distributed user environment. The relevance of the research lies in the fact that user messages, emails, attachments, and documents are stored in various digital channels, while a unified search interface for accessing them is typically absent. The goal of the work is to develop an architecture that can interpret a query, extract its parameters, select relevant data sources, access external APIs, and return the result in a structured format. The methodology is based on publications on multi-agent systems and related topics, as well as the official documentation of APIs for services that can be used as data sources and intelligent components. As a result, a three-agent architecture has been proposed. The practical value of the work lies in the suitability of this framework for the subsequent development of a software prototype for a personal intelligent assistant.

Keywords: multi-agent system, intelligent assistant, natural language processing, information retrieval, distributed data sources

Введение

Пользователь при использовании различных сервисов все чаще сталкивается с тем, что история общения распределена между чатами, письма и вложения находятся в почтовых сервисах, а документы и рабочие материалы – в файловых хранилищах. Даже простой запрос на поиск нужного сообщения или вложения в такой ситуации превращается в последовательность ручных переходов между интерфейсами. В исследованиях по интеллектуальным системам это противоречие обычно снимается за счет агентной декомпозиции, когда сложная задача разделяется между специализированными ролями, а координация между ними становится самостоятельной функцией системы [1–3]. Однако одной поисковой подсистемы для такого класса задач недостаточно. Чтобы ассистент действительно работал с персональными данными, ему необходимо понимать намерение пользователя, удерживать контекст обращения, уточнять недостающие параметры, а затем выбирать корректный сценарий исполнения. Поэтому архитектурная база подобного решения естественным образом опирается на исследования по персональным когнитивным ассистентам, когнитивным диалоговым системам и современным архитектурам диалоговых агентов [4–7].

Не менее важен языковой слой. Для персонального поиска недостаточно буквально сопоставлять слова запроса и текста документов. Необходимо выделять сущности, учитывать временные ограничения, строить более содержательное представление пользовательского намерения и в ряде случаев применять вопросно-ответные механизмы над найденными данными [7–10].

Отдельного внимания заслуживает голосовой режим и мессенджерный контур. Публикации по речевым интерфейсам и прикладным вопросно-ответным системам в мессенджерах показывают, что итоговое качество определяется не только корректностью распознавания речи, но и тем, насколько удачно распознанный текст включается в дальнейшую логику поиска и исполнения [11, 12].

Целью работы является разработка концептуальной архитектуры персонального интеллектуального ассистента для работы с распределенными источниками данных по текстовым и голосовым запросам. В статье рассматривается не чат-бот в узком смысле, а архитектурная схема, позволяющая объединить личную историю сообщений, электронную почту и файловые хранилища в едином контуре поиска и обработки.

Методы и материалы

Работа носит концептуальный характер. Сначала выделяются пользовательские сценарии, затем определяются необходимые данные, после чего функции системы раскладываются по специализированным ролям. Такой подход удобен тем, что позволяет не смешивать понимание запроса, маршрутизацию сценария и собственно вызов внешних сервисов.

Для доступа к личной истории сообщений в Telegram в предлагаемом варианте используется клиентский контур Telegram API/TDLib. Это удобное решение, так как архитектура ориентирована именно на персональный поиск по сооб-

щениям пользователя. В документации TDLib явно поддерживается поиск сообщений по всем чатам и внутри отдельного чата, а значит, позволяет строить сценарии междialogового и контекстного поиска поверх пользовательской истории [13].

Почтовый и основные файловые контуры в статье опираются на сервисы Google. Gmail API предоставляет REST-доступ к почтовым ящикам, использует модель OAuth-scopes и поддерживает push-уведомления об изменениях ящика. Google Drive API, в свою очередь, поддерживает поиск файлов и папок, получение содержимого и поэтапную обработку изменений через changes.list. В совокупности это делает почту и файловое хранилище пригодными для событийно-ориентированного персонального ассистента [14, 15].

В качестве дополнительного файлового источника и альтернативного облачного контура может использоваться REST API Яндекс Диска, а для голосового режима – SpeechKit STT. Такой выбор важен не только с практической точки зрения, но и архитектурно: в системе появляется отдельный взаимозаменяемый слой доступа к файлам и отдельный взаимозаменяемый слой распознавания речи, что снижает жесткую зависимость от одного провайдера [16, 17].

Для интерпретации запроса, структурирования параметров и подключения инструментов могут использоваться модели OpenAI и Сбер GigaChat. Официальная документация обеих платформ поддерживает функциональные вызовы и подключение внешних инструментов, а OpenAI дополнительно предоставляет отдельный контур speech-to-text. Для концептуальной архитектуры это означает, что языковая интерпретация и выполнение внешних действий могут оставаться модульными и заменяемыми [18, 19].

В качестве промежуточного хранилища в работе предполагается реляционная база данных с JSON-полями и полнотекстовой индексацией. Полная копия пользовательских данных системе не требуется. Однако ей необходимы служебные таблицы для хранения нормализованных метаданных, найденных сущностей, идентификаторов объектов, статусов фоновых задач и журналов исполнения.

Результаты

В результате проведенного исследования сформирована архитектура мультиагентного интеллектуального ассистента, предназначенного для поиска и обработки информации из распределенных источников данных. Система включает три основных компонента: диалоговый агент, агент-оркестратор и агент исполнения. Общая схема взаимодействия компонентов представлена на рис. 1.



Рис. 1. Общая схема архитектуры интеллектуального ассистента для работы с распределенными источниками данных

Представленная схема показывает общий порядок прохождения пользовательского запроса через систему. На первом этапе запрос поступает через текстовый или голосовой интерфейс, после чего преобразуется в форму, пригодную для дальнейшей обработки. Далее запрос передается диалоговому агенту, который выделяет намерение пользователя и основные параметры задачи. Если данных достаточно, сформированная структура направляется агенту-оркестратору; если параметры неполные, система возвращает уточняющий вопрос.

Диалоговый агент реализует первичное взаимодействие с пользователем и выполняет преобразование естественно-языкового запроса в структурированное представление. Его схема приведена на рис. 2.

Диалоговый агент принимает текстовый запрос или результат распознавания голосовой команды. Затем он выполняет нормализацию текста, определяет намерение пользователя, извлекает сущности и фиксирует дополнительные параметры: источник данных, временной диапазон, тип объекта и ожидаемое действие. Например, запрос «найди вложение с договором из переписки за март» должен быть преобразован в структуру, где действие связано с поиском вложения, объектом является договор, а временным ограничением выступает март. Такой подход позволяет отделить пользовательскую формулировку от внутреннего формата задачи.

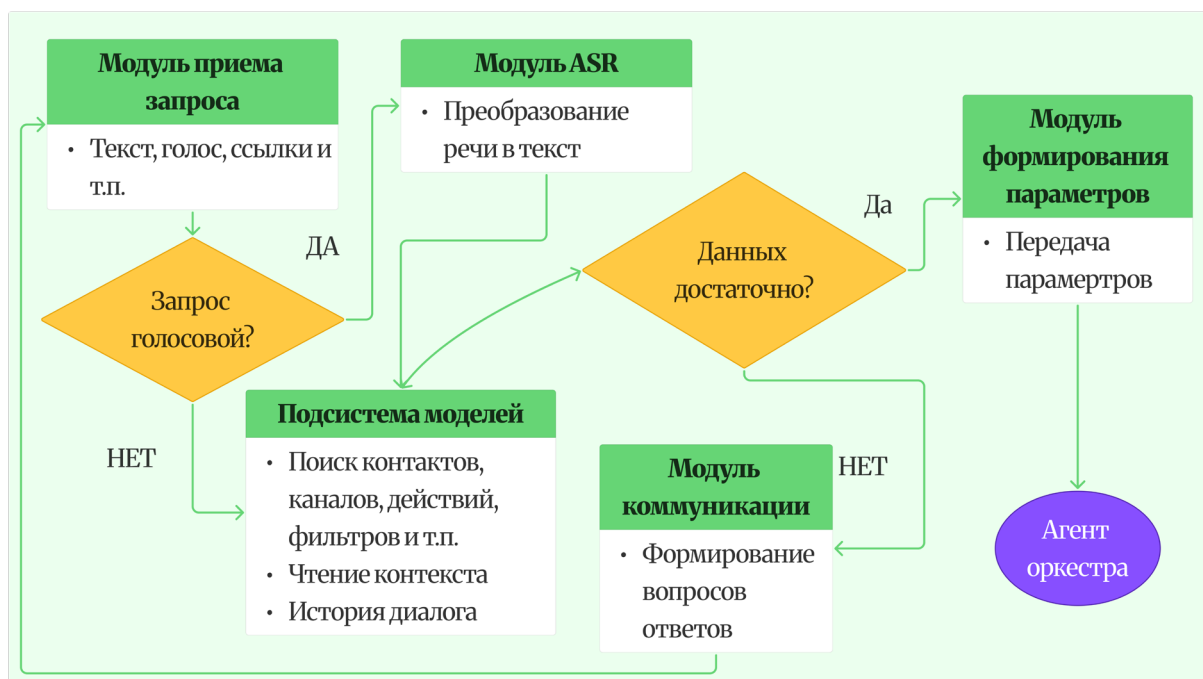


Рис. 2. Структура диалогового агента

Центральным компонентом архитектуры является агент-оркестратор, схема которого представлена на рис. 3.

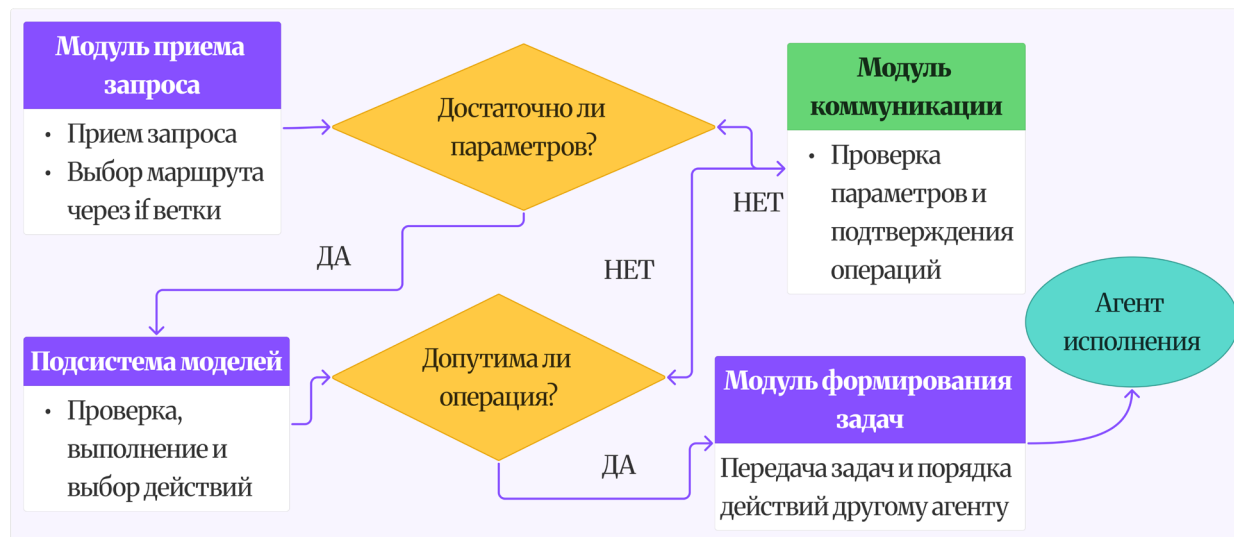


Рис. 3. Схема агента-оркестратора

Агент-оркестратор выполняет функции принятия решений и управления сценарием обработки. Он получает от диалогового агента формализованный запрос, проверяет полноту параметров, определяет допустимость операции и выбирает дальнейший маршрут выполнения. В зависимости от содержания запроса оркестратор может вернуть уточнение пользователю, обратиться к промежуточ-

ному хранилищу, передать задачу агенту исполнения или запустить фоновую обработку. Благодаря этому в системе разделяются понимание запроса и непосредственное выполнение действий.

Агент исполнения отвечает за обращение к внешним источникам данных и выполнение операций, выбранных оркестратором. Его структура показана на рис. 4.

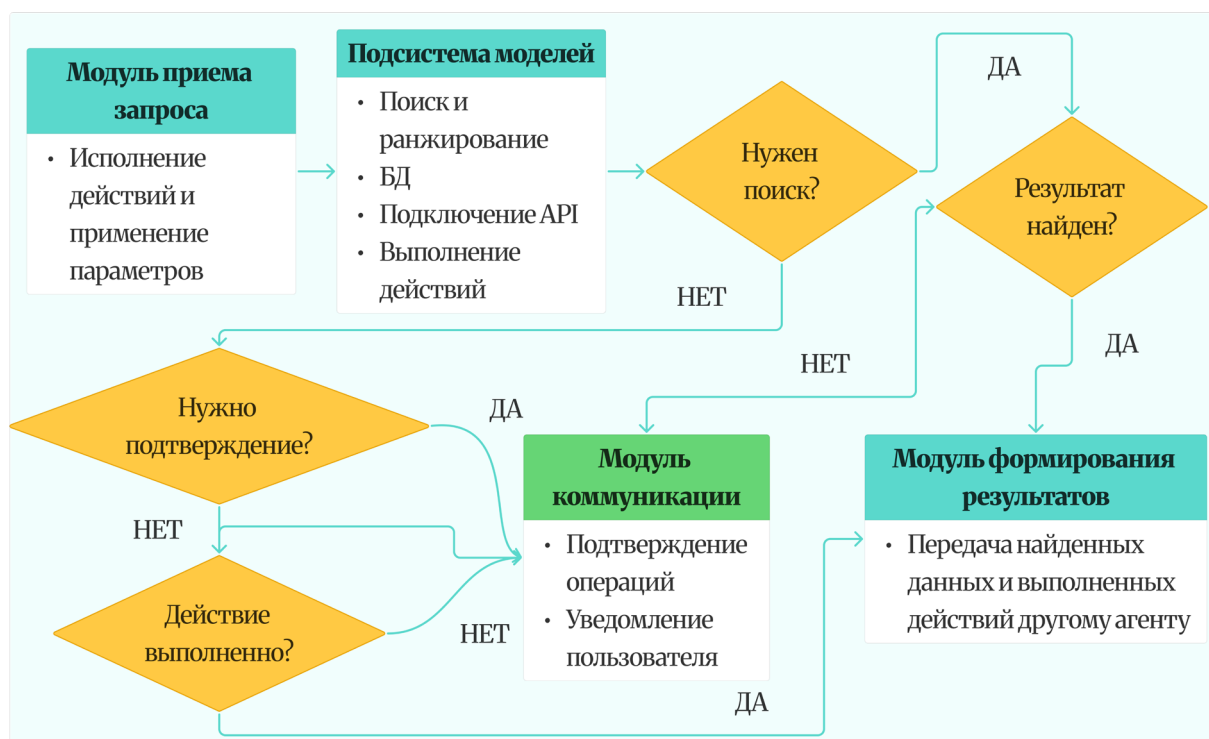


Рис. 4. Схема работы агента исполнения

Агент исполнения работает с почтовыми сервисами, мессенджерами, файловыми хранилищами, календарными сервисами и промежуточной базой данных. При поисковом сценарии он извлекает сообщения, письма, вложения или документы, нормализует полученные данные и возвращает результат в систему. При исполнительном сценарии агент может инициировать действие во внешнем сервисе, например создать событие в календаре, подготовить черновик документа или сохранить найденные сведения в хранилище. Для повышения устойчивости результаты операций и служебные метаданные могут сохраняться в базе данных.

В ходе проектирования были выделены основные пользовательские сценарии, которые отражены на диаграмме прецедентов (рис. 5).



Рис. 5. Диаграмма прецедентов интеллектуального ассистента

К основным сценариям относятся поиск сообщений по смысловому описанию, поиск вложений по контексту, поиск файлов в подключенных хранилищах, создание события в календаре, подготовка документа по образцу и запуск фоновой обработки. Отдельно учитывается сценарий уточнения запроса, поскольку пользователь не всегда указывает все параметры сразу. В этом случае система не завершает выполнение с ошибкой, а возвращает уточняющий вопрос через диалоговый агент.

Таким образом, предложенная архитектура объединяет пользовательский интерфейс, интеллектуальную обработку запроса, оркестрацию действий и доступ к распределенным источникам данных. Разделение функций между агентами делает систему расширяемой: при добавлении нового сервиса требуется изменить только слой исполнения и правила маршрутизации, не перестраивая всю архитектуру.

Обсуждение

Предложенная архитектура обладает несколькими важными достоинствами. Прежде всего, она уменьшает связанность между смысловой интерпретацией запроса и вызовом внешних сервисов. Это особенно важно в персональной цифровой среде, где почта, сообщения и файлы имеют разные модели доступа и разные механизмы синхронизации. Мультиагентная декомпозиция делает эту неоднородность не препятствием, а нормальным проектным условием.

Не менее важна модульность. Если в дальнейшем потребуется заменить речевой сервис, языковую модель или один из файловых коннекторов, такая замена не затронет всю систему. Благодаря этому архитектура хорошо подходит не только для быстрого прототипирования, но и для постепенного перехода от об-

лачных компонентов к локальным. Это особенно полезно в сценариях, где чувствительность данных ограничивает возможность внешней обработки.

При этом концепт имеет и ограничения. Во-первых, реальная реализация потребует аккуратной работы с правами доступа, токенами, журналами исполнения и политиками хранения данных. Во-вторых, качество ответа будет зависеть не только от языковой модели, но и от того, насколько корректно внутри системы описаны интенды, сущности, временные фильтры и условия полноты запроса. Поэтому основным направлением дальнейшей работы должна стать формализация межагентного контракта и экспериментальная проверка его устойчивости на реальных пользовательских сценариях.

Заключение

В статье разработана концептуальная архитектура персонального интеллектуального ассистента для работы с распределенными источниками данных по текстовым и голосовым запросам. Архитектура построена вокруг трех ролей – диалогового агента, агента-оркестратора и агента исполнения – и дополнена промежуточным хранилищем и фоновым контуром обработки.

Показано, что такой подход позволяет объединить в одном естественно-языковом интерфейсе поиск по личной истории сообщений, электронной почте и файловым хранилищам, а также поддерживать последующую интеллектуальную обработку результатов. Практическая значимость работы состоит в пригодности предложенной архитектуры для создания программного прототипа персонального ассистента. Дальнейшее развитие исследования целесообразно связать с реализацией прототипа, формализацией межагентного обмена и сравнительной оценкой облачных и локальных моделей в задачах персонального поиска.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Прикладные многоагентные системы группового управления / В. И. Городецкий, О. В. Карсаев, В. В. Самойлов, С. В. Серебряков // Искусственный интеллект и принятие решений. – 2009. – № 2. – С. 3-24. – EDN KWTRFV.
2. Рыбина, Г. В. Моделирование процессов взаимодействия интеллектуальных агентов в многоагентных системах / Г. В. Рыбина, С. С. Паронджанов // Искусственный интеллект и принятие решений. – 2008. – № 3. – С. 3-15. – EDN JYAFUL.
3. Леденева, Т. М. Многоагентные системы: модели агентов и постановки задач, обеспечивающие их характерные свойства (часть 1) / Т. М. Леденева, М. А. Принев // Вестник Воронежского государственного университета. Серия: Системный анализ и информационные технологии. – 2024. – № 3. – С. 50-73. – DOI 10.17308/sait/1995-5499/2024/3/50-73. – EDN GHJMUH.
4. Персональный когнитивный ассистент: концепция и принципы работы / И. В. Смирнов, А. И. Панов, А. А. Скрынник, Е. В. Чистова // Информатика и ее применения. – 2019. – Т. 13, № 3. – С. 105-113. – DOI 10.14357/19922264190315. – EDN QQSTUK.
5. Станкевич, Л. А. Когнитивные диалоговые системы / Л. А. Станкевич // XXI международная научно-техническая конференция "НЕЙРОИНФОРМАТИКА-2019": Сборник научных трудов, Москва, 07–11 октября 2019 года. Том Часть 1. – Москва: Московский физико-технический институт (национальный исследовательский университет), 2019. – С. 9-62. – EDN IRTDJD.

6. Новая архитектура диалогового агента для прикладной генерации диалогов / В. А. Рындюк, Я. С. Варакин, Е. А. Писаренко, Е. Н. Пронченко // Волновая электроника и информационные системы : Сборник статей XXIV Международной научной конференции, Санкт-Петербург, 31 мая – 04 2021 года. Том Часть 3. – Санкт-Петербург: Санкт-Петербургский государственный университет аэрокосмического приборостроения, 2021. – С. 169-174. – EDN OYUYPYPS.
7. Смирнов, И. В. Разноуровневая обработка естественного языка для интеллектуального поиска и анализа текстов / И. В. Смирнов // Искусственный интеллект и принятие решений. – 2023. – № 1. – С. 90-99. – DOI 10.14357/20718594230109. – EDN ALPUWY.
8. Максименко, О. А. Технологии семантического поиска информационных ресурсов / О. А. Максименко, А. П. Савченко // Общество и экономика знаний, управление капиталами: цифровая экономика знаний : Материалы XII Международная научно-практическая конференция, Краснодар, 27–28 мая 2022 года. – Краснодар: Кубанский государственный университет, 2022. – С. 510-515. – EDN QGRDXD.
9. Алексеев, А. А. Автоматическое извлечение сущностей на основе структуры новостного кластера / А. А. Алексеев, Н. В. Лукашевич // Искусственный интеллект и принятие решений. – 2011. – № 4. – С. 51-59. – EDN PXGUUZ.
10. Открытое извлечение информации из текстов. Часть III. Система вопросно-ответного поиска / Е. В. Чистова, Д. С. Ларионов, А. О. Шелманов [и др.] // Искусственный интеллект и принятие решений. – 2021. – № 4. – С. 35-49. – DOI 10.14357/20718594210204. – EDN LUQGSK.
11. Повышение эффективности речевого интерфейса с применением когнитивных и лингвистических знаний / М. П. Фархадов, Н. В. Петухова, С. В. Васьковский, М. Э. Фархадова // Управление большими системами: сборник трудов. – 2019. – № 81. – С. 90-112. – DOI 10.25728/ubs.2019.81.4. – EDN FVONJA.
12. Вопросно-ответная система для поддержки абитуриентов с использованием современных мессенджеров / Д. Р. Филонов, Д. Ю. Чалый, Д. М. Мурин [и др.] // Моделирование и анализ информационных систем. – 2018. – Т. 25, № 4(76). – С. 411-420. – DOI 10.18255/1818-1015-2018-4-411-420. – EDN XWQZHF.
13. Telegram APIs. URL: <https://core.telegram.org/> (дата обращения: 27.04.2026).
14. Gmail API overview: <https://developers.google.com/workspace/gmail/api/guides> (дата обращения: 27.04.2026).
15. Search for files and folders. URL: <https://developers.google.com/workspace/drive/api/guides/search-files> (дата обращения: 27.04.2026).
16. REST API Диска – Технологии Яндекса. URL: <https://yandex.ru/dev/disk/rest/> (дата обращения: 27.04.2026).
17. Speech recognition | AI Studio documentation. URL: <https://aistudio.yandex.ru/docs/en/speechkit/stt/> (дата обращения: 27.04.2026).
18. OpenAI API: Responses Overview. URL: <https://developers.openai.com/api/reference/responses/overview/> (дата обращения: 27.04.2026).
19. GigaChat API: Обзор. URL: <https://developers.sber.ru/docs/ru/gigachat/api/overview> (дата обращения: 27.04.2026).

© Н. С. Федотов, П. Ю. Бугаков, 2026

Н. С. Филина^{1✉}, Е. Ю. Воронкин¹

Разработка информационной системы для хранения и подготовки к печати документов демаркации российско-литовской государственной границы

¹Сибирский государственный университет геосистем и технологий,
г. Новосибирск, Российская Федерация
e-mail: filina.nelli050@mail.ru

Аннотация. В статье представлены основные результаты разработки информационной системы для хранения и подготовки к печати документов демаркации российско-литовской государственной границы. Рассмотрена проблема разрозненного хранения материалов в папках и на разных рабочих компьютерах, что затрудняет поиск документов, приводит к появлению дубликатов и усложняет подготовку итоговых комплектов к печати. Для решения данной проблемы была разработана настольная информационная система, объединяющая файловое хранилище, базу данных, механизмы поиска и фильтрации, работу с атласами документов и модуль формирования печатного PDF. В качестве основного стека технологий использованы C#, WPF, PostgreSQL, Entity Framework Core, QuestPDF, PdfSharp и механизм COM-взаимодействия с Microsoft Office. Разработанная система позволяет загружать документы в архив, контролировать наличие дубликатов, выполнять поиск и сортировку, собирать материалы в атласы и формировать итоговые комплекты в формате ZIP и PDF.

Ключевые слова: информационная система, электронный архив, документы демаркации, хранение документов, поиск документов, контроль дубликатов, атласы документов, подготовка к печати, PDF

N. S. Filina^{1✉}, E. Yu. Voronkin¹

Development of an information system for storing and preparing for printing documents of the russian-lithuanian state border demarcation

¹Siberian State University of Geosystems and Technologies, Novosibirsk, Russian Federation
e-mail: filina.nelli050@mail.ru

Abstract. The article presents the main results of developing an information system for storing and preparing for printing documents of the Russian-Lithuanian state border demarcation. The problem of fragmented storage of materials in folders and on different workstations is considered, which complicates document search, leads to duplicates and makes preparation of final document sets for printing more difficult. To solve this problem, a desktop information system was developed that combines file storage, a database, search and filtering mechanisms, document atlases and a printable PDF generation module. The main technology stack includes C#, WPF, PostgreSQL, Entity Framework Core, QuestPDF, PdfSharp and COM interaction with Microsoft Office. The developed system allows users to upload documents to the archive, control duplicates, search and sort materials, assemble them into atlases and generate final sets in ZIP and PDF formats.

Keywords: information system, electronic archive, demarcation documents, document storage, document search, duplicate control, document atlases, print preparation, PDF

Введение

Документы, связанные с демаркацией государственной границы, представляют собой разнородный архив материалов, включающий текстовые документы, акты, схемы, карты, таблицы, сканированные копии и графические файлы. На практике такие материалы часто хранятся в папках на рабочих компьютерах и в сетевых каталогах, что создает проблемы при повседневной работе. Поиск нужного документа занимает много времени, одинаковые файлы могут сохраняться в нескольких местах, а формирование подборок для печати выполняется вручную.

Особенно заметны эти трудности при работе с крупным архивом, где документы используются не по отдельности, а в виде тематических комплектов. В этом случае недостаточно просто хранить файлы в папках. Необходимо иметь возможность быстро находить материалы по их характеристикам, связывать документы между собой, исключать дубликаты и формировать упорядоченные подборки для дальнейшего использования.

Целью работы является разработка информационной системы для хранения и подготовки к печати документов демаркации российско-литовской государственной границы. В рамках проекта была реализована настольная система, позволяющая загружать документы в архив, хранить сведения о них в базе данных, выполнять поиск, фильтрацию и сортировку, формировать атласы документов и получать итоговые печатные комплекты.

Исследование предметной области

Архив документов демаркации государственной границы отличается от обычного набора электронных файлов тем, что в нем одновременно присутствуют материалы разных форматов и назначения. Это могут быть PDF-документы, изображения, сканы, документы Word и Excel. При этом один документ нередко используется совместно с другими материалами, то есть входит в состав общего комплекта.

Традиционный способ хранения в папках удобен только на самом простом уровне работы с архивом. Пока файлов немного, пользователь может ориентироваться в них вручную. Однако по мере увеличения объема архива возникают проблемы: документы дублируются, поиск замедляется, структура хранения усложняется, а подготовка комплектов для печати требует постоянной ручной работы.

Готовые универсальные системы электронного документооборота также не всегда подходят для подобных задач. Они часто содержат широкий набор функций, рассчитанных на общий документооборот, но не учитывают особенности работы именно с архивом документов демаркации. В рассматриваемой задаче важно не только хранение документов, но и возможность их объединения в атласы, изменения порядка, а также подготовки итогового печатного PDF.

С учетом перечисленных особенностей было принято решение разработать специализированную информационную систему, ориентированную именно на

задачи хранения, упорядочения и подготовки к печати документов демаркации государственной границы.

Методы и технологии

Разработка системы была выполнена с использованием языка программирования C#. Данный язык был выбран как удобное средство создания настольных приложений под Windows, обеспечивающее работу с файлами, базой данных и внутренней логикой программы [1, 2].

Для реализации пользовательского интерфейса использовалась технология WPF. Она позволяет создавать окна, формы, таблицы и элементы управления, необходимые для работы с электронным архивом. Такой подход был выбран, поскольку разрабатываемая система ориентирована на локальную работу в среде Windows и предполагает удобное табличное представление документов [3].

В качестве системы управления базой данных использовалась PostgreSQL. Она применялась для хранения метаданных документов, сведений о категориях, тегах, атласах и связях между сущностями. Работа с базой данных была организована через Entity Framework Core, что упростило реализацию операций чтения, добавления и изменения данных [4, 5].

Сами файлы документов сохраняются не в базе данных, а в отдельном файловом хранилище. Такое решение позволяет не перегружать базу крупными файлами и делает структуру системы более удобной для сопровождения.

Для формирования итогового PDF в проекте использовались библиотеки QuestPDF и PdfSharp. С их помощью была реализована сборка итогового документа, включающего титульный лист, оглавление и материалы атласа [6, 7]. Для включения в печатную сборку документов Word и Excel использовались возможности установленного Microsoft Office через COM-взаимодействие [8].

Таким образом, для разработки системы был использован следующий стек технологий:

- язык программирования C#;
- технология WPF для пользовательского интерфейса;
- PostgreSQL для хранения данных;
- Entity Framework Core для работы с базой данных;
- файловое хранилище для сохранения документов;
- QuestPDF и PdfSharp для формирования PDF;
- COM-взаимодействие с Microsoft Office для конвертации Word и Excel.

Результаты

В результате работы была разработана информационная система, обеспечивающая хранение и подготовку к печати документов демаркации российско-литовской государственной границы. Система реализована как настольное приложение, объединяющее интерфейс пользователя, базу данных, файловое хранилище и механизмы экспорта.

В структуре базы данных центральное место занимает сущность документа. Для каждого документа сохраняются его название, исходное и служебное имя

файла, хеш, размер, дата загрузки, описание, тип документа и другие характеристики. Дополнительно были реализованы сущности категорий, тегов, атласов и таблицы связей между ними. Это позволило работать с архивом не как с набором разрозненных файлов, а как с более структурированной системой.

В пользовательском интерфейсе реализованы следующие основные функции:

- загрузка документа в архив;
- автоматическая проверка на наличие дубликатов;
- просмотр списка документов;
- поиск, фильтрация и сортировка материалов;
- создание и редактирование атласов;
- изменение порядка документов внутри атласа;
- экспорт атласа в ZIP-архив;
- формирование единого печатного PDF.

Контроль дубликатов реализован на основе вычисления хеша файла. При загрузке документа система сравнивает его хеш со значениями, уже сохраненными в базе данных. Это позволяет выявлять повторяющиеся документы даже в тех случаях, когда они имеют разные имена файлов.

Одним из ключевых результатов стала реализация механизма атласов. Атлас в системе представляет собой упорядоченную подборку документов, собранную для решения конкретной задачи или для последующей печати. При этом документы не копируются повторно, а связываются с атласом через отдельную таблицу, в которой дополнительно хранится порядок их расположения.

Модуль подготовки к печати обеспечивает обработку документов разных форматов. Если в атлас входит PDF-файл, он включается в итоговую сборку без изменений. Изображения преобразуются в страницы PDF, а документы Word и Excel предварительно конвертируются в PDF. После этого все материалы объединяются в единый печатный документ.

Для оценки работоспособности системы было проведено функциональное, автоматическое и нагрузочное тестирование. Проверка показала, что загрузка документов, контроль дубликатов, поиск, фильтрация, создание атласов и экспорт работают корректно. Нагрузочное тестирование продемонстрировало устойчивость системы при увеличении количества документов в архиве.

Разработанное решение можно рассматривать как рабочую основу для дальнейшего использования и расширения. Система уже позволяет упорядочить архив документов, ускорить поиск материалов и сократить время подготовки итоговых комплектов к печати.

Архив документов Роскастра - v1.0						
АРХИВ ДОКУМЕНТОВ al UploadD По убыванию Сорес Загрузить Атласм Обновить Администратор						
КАТЕГОРИИ						
Все документы (15 документов)						
Все документы Название Файл Тип Размер Загрузил Дата загрузки						
искход данные.xlsx						
Тема 1.docx						
4_Содержание.docx						
3_Документы_Правительства_РФ_шапка_разд						
2_Предисловие.docx						
1_Титульный_лист.docx						
P1_кард1.png						
P1_кард2.png						
otchet.docx						
Отчет.jpg						
Карта Филина БИ-41.jpg						
Дневник_Филина.pdf						
Характеристика_Филина.pdf						
Аттестационный_лист_Филина.docx						
Отчет Техн_Филина.docx						

Рис. 1. Интерфейс информационной системы

Заключение

В результате выполнения работы была разработана информационная система для хранения и подготовки к печати документов демаркации российско-литовской государственной границы.

В рамках проекта были решены основные задачи: проведен анализ предметной области, определены требования к системе, спроектированы архитектура, база данных и пользовательский интерфейс, реализованы модули загрузки документов, контроля дубликатов, поиска, фильтрации, формирования атласов и экспорта итоговых материалов.

Практическая значимость работы заключается в том, что разработанная система может использоваться как основа для упорядочения архива документов и упрощения подготовки печатных комплектов. В дальнейшем развитие проекта может быть связано с расширением поиска, добавлением предпросмотра документов, развитием многопользовательского режима, разграничением прав доступа и дальнейшей адаптацией системы для практического внедрения.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Troelsen, A. Pro C# 10 with .NET 6 : Foundational Principles and Practices in Programming. – New York : Apress, 2022. – 1040 p.
2. Price, M. C# 10 and .NET 6 – Modern Cross-Platform Development. – Birmingham : Packt Publishing, 2021. – 822 p.
3. Петцольд, Ч. WPF: подробное руководство / пер. с англ. – Санкт-Петербург : Питер, 2019. – 880 с.

4. PostgreSQL: Documentation. – URL: <https://www.postgresql.org/docs/> (дата обращения: 14.05.2026). – Текст : электронный.
5. Microsoft Learn. Entity Framework Core. – URL: <https://learn.microsoft.com/ef/core/> (дата обращения: 14.05.2026). – Текст : электронный.
6. QuestPDF Documentation. – URL: <https://www.questpdf.com/> (дата обращения: 14.05.2026). – Текст : электронный.
7. PDFsharp & MigraDoc Foundation. – URL: <https://pdfsharp.net/> (дата обращения: 14.05.2026). – Текст : электронный.
8. Microsoft Learn. Office Primary Interop Assemblies. – URL: <https://learn.microsoft.com/office/client-developer/interop/> (дата обращения: 14.05.2026). – Текст : электронный.

© Н. С. Филина, Е. Ю. Воронкин, 2026

Р. Р. Фиткулина^{1✉}, Н. С. Головачев¹

Разработка серверного модуля для поддержки сетевых функций игры «BlushBound»

¹Сибирский государственный университет геосистем и технологий,
г. Новосибирск, Российская Федерация
e-mail: ruslanafit04@gmail.com

Аннотация. В статье описывается процесс реализации серверной части для игрового продукта «BlushBound», в котором объединены элементы визуальной новеллы и системы накопления уникальных наград. Реализация синхронизации прогресса пользователя, долговременного хранения информации о пользователях и внедрения сетевых. Рассмотрены архитектура серверного блока, схема БД, а также алгоритмы API для входа в систему, синхронизации достижений, управления инвентарем и механики получения уникальных наград. Отдельно рассмотрены вопросы сохранения целостности данных через транзакции и валидации запросов. Проведены ручные, автоматические и нагрузочные тесты. Итоги подтверждают устойчивую работу модуля при последовательных вызовах и корректную реализацию игровых сценариев. Результаты могут быть востребованы при создании инди-игр с клиент-серверной моделью и небольшим бюджетом.

Ключевые слова: серверный модуль, клиент-серверная архитектура, видеоигра, PHP, MySQL, Unity

R. R. Fitkulina^{1✉}, N. S. Golovachev¹

Development of a server module for supporting network functions in the game «BlushBound»

¹Siberian State University of Geosystems and Technologies, Novosibirsk, Russian Federation
e-mail: ruslanafit04@gmail.com

Abstract. The article describes the process of implementing the server-side component for the game project “BlushBound,” which combines elements of a visual novel and a unique reward collection system. The implementation includes user progress synchronization, long-term storage of user data, and the integration of network interaction features. The paper examines the architecture of the server module, the database schema, and the API algorithms for user authentication, achievement synchronization, inventory management, and unique reward mechanics. Particular attention is paid to maintaining data integrity through transactions and request validation. Manual, automated, and load testing were conducted. The results confirm the stable operation of the module under sequential requests and the correct implementation of gameplay scenarios. The obtained results may be useful in the development of indie games with a client-server architecture and a limited budget.

Keywords: server module, client-server architecture, video game, PHP, MySQL, Unity

Введение

В настоящее время игровые продукты все чаще отходят от модели исключительно локального хранения информации, переходя к использованию внешних серверных систем [1–3]. Даже те проекты, которые не задуманы как многополь-

зовательские, нуждаются в механизмах записи прогресса, отслеживания инвентаря и контроля внутриигровых ресурсов. Причина – как в необходимости защищать данные пользователей, так и в массовом распространении игр, построенных на коллекционировании предметов и персонажей [4, 5].

Для небольших команд разработчиков развертывание собственной серверной инфраструктуры – нетривиальная инженерная задача. Облачные сервисы «из коробки» ускоряют старт, но почти всегда сопряжены с регулярными платежами и мешают гибко менять внутреннюю логику [6, 7]. Поэтому при создании небольших игр сохраняет актуальность самостоятельная разработка серверных модулей, которые обеспечивают базовый сетевой функционал без привлечения платных внешних платформ [8, 9].

В данной работе речь идет о создании серверной составляющей для игры «BlushBound». Проект комбинирует признаки рогалика, визуального романа и системы случайного получения лута. Подобная структура подразумевает хранение больших объемов пользовательской информации: пройденные уровни, состояние аватаров, наполнение инвентаря, статистика сессий.

Ключевая цель исследования – реализовать такой серверный модуль, который через API-интерфейс соединяет клиентскую часть игры с базой данных [10]. Параллельно решались задачи по организации входа пользователей, обработке игровых вызовов и предотвращению разрушения целостности данных при параллельных операциях [5, 11].

Методы и материалы

На этапе предпроектного анализа перебирались разные варианты построения бэкенда: Cloud-решения Firebase, PlayFab и другие. Однако внедрение этих платформ ведет к росту эксплуатационных расходов и ограничивает возможность доработки игровой логики [7]. В итоге выбор остановлен на самостоятельной разработке серверного компонента с использованием PHP и MySQL [12, 13].

Бэкенд построен по классической схеме REST API [10]. Данные между клиентом и сервером передаются через HTTP-запросы, ответы формируются в JSON. Такое решение сильно упрощает сопряжение с клиентской частью Unity и отделяет логику клиента от деталей внутреннего устройства базы данных [14, 15].

Функционально серверная часть разделена на несколько крупных блоков: регистрация и аутентификация, загрузка и сохранение прогресса, управление инвентарем, проведение игровых транзакций, а также реализация механики случайной выдачи награды.

На рис. 1 представлена обобщенная архитектура взаимодействия пользователя, клиентского приложения на Unity и разработанного серверного модуля. Пользователь через клиентскую часть инициирует такие действия, как регистрация, авторизация, прохождение уровней, открытие предметов и управление инвентарем. Клиентское приложение, в свою очередь, формирует соответствующие запросы к серверному модулю: отправка боевых действий, запрос на открытие предметов, запрос инвентаря, экипировки, улучшений, профиля и заверше-

ния сессии. Серверный модуль обрабатывает каждый из этих запросов – от регистрации и авторизации до отдачи статических страниц, обработки боя, механики получения предметов, предоставления инвентаря, экипировки, улучшения, профиля и завершения сессии.

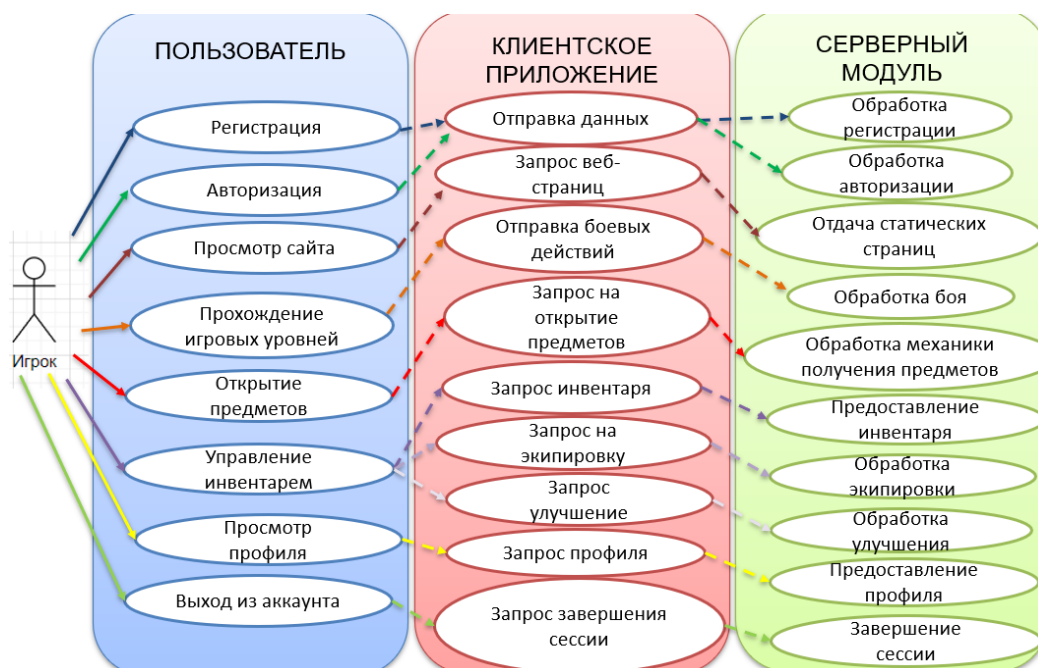


Рис. 1. Схема взаимодействия пользователя, клиентского приложения и серверного модуля

Каждый API-метод реализован в виде отдельного PHP-скрипта. На рис. 2 приведен сокращенный пример обработчика авторизации.

Данный фрагмент демонстрирует основные этапы аутентификации: получение учетных данных, подготовленный запрос к базе данных для предотвращения SQL-инъекций, верификацию пароля через `password_verify`, генерацию криптостойкого сессионного токена и его сохранение. Такой подход обеспечивает базовый уровень безопасности [5]. Пароли хранятся исключительно в хешированном виде, что исключает их компрометацию при возможной утечке базы данных [4].

Наибольшее внимание при кодировании было уделено операциям изменения игровой валюты и выдачи предметов. Такие действия обернуты в SQL-транзакции [11, 16]. Это гарантирует, что данные останутся непротиворечивыми даже при сбоях в ходе обработки запроса.

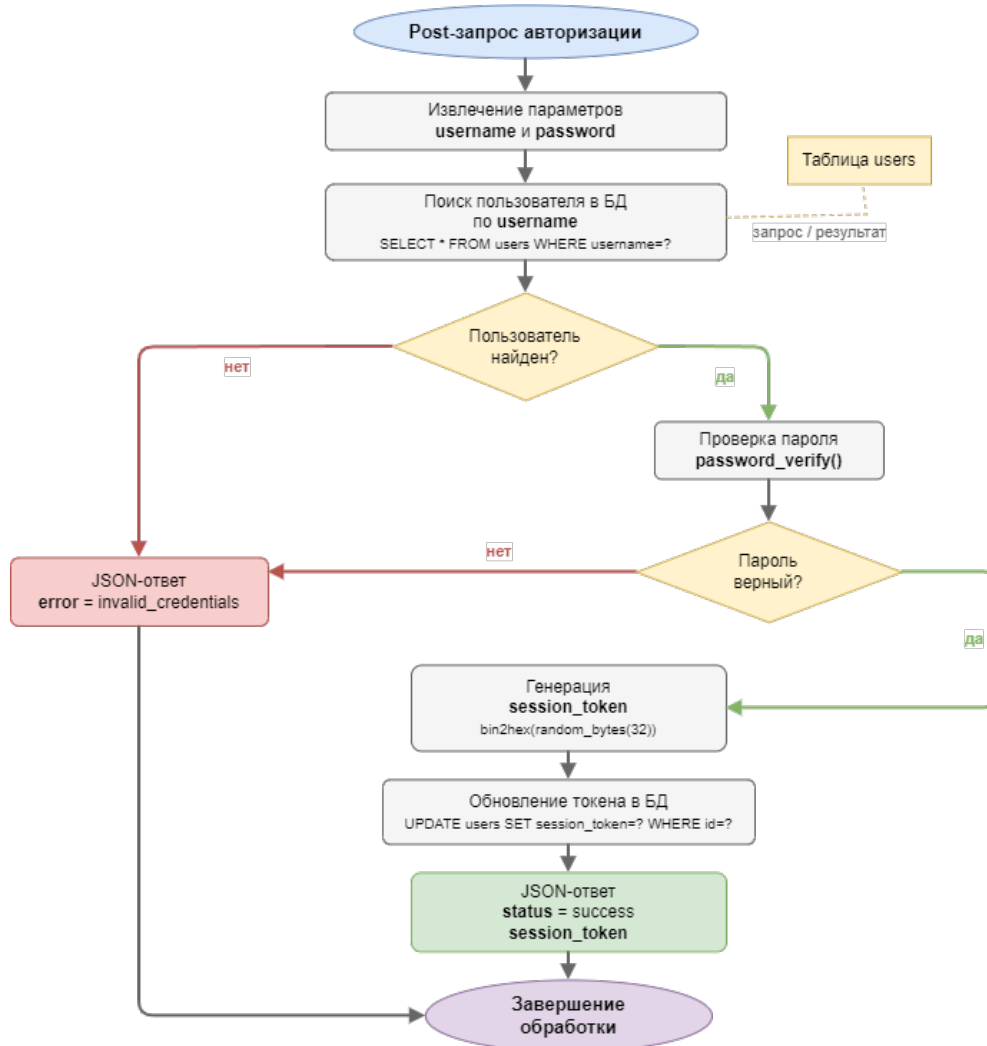


Рис. 2. Схема API-метода обработки авторизации

Результаты

Итогом работы стал полноценный серверный модуль, покрывающий все основные сетевые задачи игрового проекта «BlushBound». Серверная часть обеспечивает регистрацию и последующую авторизацию игроков, хранение и синхронизацию игрового прогресса, а также предоставление данных о содержимом инвентаря, и обработку событий, связанных с выдачей наград для развития персонажей пользователей.

В процессе тестирования серверный модуль корректно обработал последовательный набор запросов, поступавших от клиентской части, реализованной в среде разработки Unity [14, 17]. Проверке подверглись сценарии, отвечающие за создание учетных записей и повторный вход пользователей в систему, сохранение игрового процесса и получение информации из базы данных.

Особое внимание в процессе проверки было уделено механизму случайной выдачи предметов. Сервер самостоятельно производит вычисление результата, не привлекая к этой операции клиентскую сторону, после чего обновляет данные учетной записи игрока и направляет готовый ответ обратно клиенту [5].

Дополнительно оценивалась устойчивость API к некорректно сформированным вызовам. Сервер правильно обрабатывает ситуации, когда токен сессии передан неправильно, когда в запросе отсутствует обязательный параметр, а также когда значение игровой валюты указано в неверном формате.

Нагрузочное тестирование подтвердило, что при последовательной обработке серий запросов модуль функционирует стабильно, не допуская потери информации и не порождая критических сбоев в работе [18].

Обсуждение

Полученное решение демонстрирует потенциал простого стека технологий, для реализации серверной инфраструктуры, полностью покрывающую нужды инди-игры с элементами онлайн-взаимодействия [8, 12]. Для проектов с небольшим бюджетом это особенно критично, поскольку расходы на коммерческие облачные платформы растут вместе с увеличением числа игроков [7].

Плюс выбранного подхода – полный контроль над схемой БД и логикой обработки запросов [19]. В отличие от универсальных платформ, разработанный серверный модуль заточен именно под задачи «BlushBound» и содержит только необходимые функции.

При этом важно отметить, что текущая версия рассчитана на небольшую аудиторию. При многократном росте числа одновременных подключений потребуются доработки – добавление кэширования, балансировщиков нагрузки и асинхронной обработки [18, 20].

Заключение

В ходе выполнения работы создан серверный модуль для игрового приложения «BlushBound», который выполняет базовые сетевые функции, надежно хранит пользовательские данные, связывает клиент Unity с сервером через API, а также обеспечивает авторизацию пользователей, обработку запросов и синхронизацию игрового прогресса [10, 14].

Для хранения всей информации в разработанном решении используется реляционная база данных MySQL [16]. Все критически важные операции реализованы с применением механизма транзакций [11].

Проведенные тесты подтвердили корректность работы всех функций модуля и его устойчивость к ошибочным запросам [5, 18]. Полученные итоги работы демонстрируют, что полноценную серверную инфраструктуру для инди-проекта можно создать без использования сложных и дорогостоящих облачных платформ.

В числе ближайших направлений дальнейшего развития проекта можно выделить оптимизацию производительности, расширение аналитической подсистемы, а также внедрение инструментов масштабирования серверной части.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Newzoo. Global Games Market Report 2025. – Режим доступа: <https://newzoo.com/resources/trend-reports/newzoos-global-games-market-report-2025-free-version> (дата обращения: 12.05.2026).

2. Смирнов А. В. Безопасность веб-приложений / А. В. Смирнов. – Санкт-Петербург: Питер, 2021. – 288 с. – ISBN 978-5-4461-1500-6.
3. Атлас А. PHP для профессионалов / А. Атлас. – Санкт-Петербург: Питер, 2023. – 480 с. – ISBN 978-5-4461-1567-3.
4. Кузнецов М. В., Симдянов И. В. PHP. Практика разработки Web-сайтов / М. В. Кузнецов, И. В. Симдянов. – 3-е изд. – Санкт-Петербург: БХВ-Петербург, 2022. – 560 с. – ISBN 978-5-9775-6839-9.
5. Кручинин А. В. REST API. Создание и использование / А. В. Кручинин. – Москва: ДМК Пресс, 2021. – 304 с. – ISBN 978-5-97060-907-1.
6. Unity Documentation: руководство по WebRequest API. – Режим доступа: <https://docs.unity3d.com/Manual/UnityWebRequest.html> (дата обращения: 12.05.2026).
7. Unity. Web request high-level API reference. – Режим доступа: <https://docs.unity3d.com/6000.4/Documentation/Manual/web-request-hlapi.html> (дата обращения: 12.05.2026).
8. Демидов С. В. Unity и C#. Геймдев от идеи до реализации / С. В. Демидов. – Санкт-Петербург: Питер, 2023. – 928 с. – ISBN 978-5-4461-0715-5.
9. Бенкен Е. С. PHP, MySQL, XML: программирование для Интернета / Е. С. Бенкен. – 5-е изд., перераб. и доп. – Санкт-Петербург: БХВ-Петербург, 2022. – 672 с. – ISBN 978-5-9775-6812-2.
10. Duckett J. PHP & MySQL: серверная веб-разработка / Дж. Дакет; пер. с англ. А. В. Логвинова. – Москва: Эксмо, 2025. – 720 с. – ISBN 978-5-04-201405-9.
11. Зандстра М. PHP: объекты, шаблоны и методики программирования / М. Зандстра. – Москва: Вильямс, 2022. – 560 с. – ISBN 978-5-8459-1564-7.
12. Мартин Р. С. Чистая архитектура. Искусство разработки программного обеспечения / Р. С. Мартин. – Санкт-Петербург: Питер, 2022. – 352 с. – ISBN 978-5-4461-1627-0.
13. Грофф Дж., Вайнберг П., Оппель Э. SQL. Полное руководство / Дж. Грофф, П. Вайнберг, Э. Оппель. – 3-е изд. – Москва: Вильямс, 2021. – 1024 с. – ISBN 978-5-8459-1920-1.
14. Орлов С. А. Технологии разработки программного обеспечения / С. А. Орлов. – 6-е изд. – Санкт-Петербург: Питер, 2024. – 624 с. – ISBN 978-5-4461-1129-9.
15. Docker Documentation. – Режим доступа: <https://docs.docker.com/> (дата обращения: 12.05.2026).
16. Гамма Э., Хелм Р., Джонсон Р., Влиссидес Дж. Приемы объектно-ориентированного проектирования. Паттерны проектирования. – Санкт-Петербург: Питер, 2021. – 368 с. – ISBN 978-5-4461-1203-6.
17. MySQL: официальная документация. – Режим доступа: <https://dev.mysql.com/doc/refman/8.0/en/> (дата обращения: 12.05.2026).
18. Петров А. Н. Разработка игр на Unity / А. Н. Петров. – Москва: Эксмо, 2023. – 416 с. – ISBN 978-5-04-170421-6.
19. Татро К., Макинтайр П. Создаем динамические веб-сайты на PHP / К. Татро, П. Макинтайр. – Санкт-Петербург: Питер, 2021. – 544 с. – ISBN 978-5-4461-1509-9.
20. Хокинг Д. Unity в действии. Мультиплатформенная разработка на C# / Д. Хокинг. – Санкт-Петербург: Питер, 2020. – 352 с. – ISBN 978-5-4461-0816-9.

© Р. Р. Фиткулина, Н. С. Головачев, 2026

Ю. Н. Ханбекова^{1✉}

Метод обнаружения атак с предъявлением фотографий в биометрических системах

¹Национальный исследовательский ядерный университет «МИФИ»,
г. Москва, Российская Федерация
e-mail: yul2002@mail.ru

Аннотация. В статье описан метод обнаружения атак с использованием поддельных биометрических данных в задачах распознавания лиц. Актуальность заключается в том, что атаки предъявления являются одним из наиболее доступных и распространенных способов обхода биометрической аутентификации. Целью исследования является разработка метода, позволяющего повысить точность определения атаки в условиях плохого освещения на основе нейросетевой модели. Предложенный метод включает модуль предварительного улучшения освещения изображения, двухветвевую модель, в которой одна ветвь предназначена для извлечения признаков исходного изображения лица, а вторая – для извлечения текстурных признаков. Полученные признаки объединяются и подаются на вход блока бинарной классификации. В работе продемонстрировано, что совместное использование коррекции освещения, анализа признаков изображения и анализа текстурных особенностей позволяет повысить устойчивость метода к различным сценариям освещения и точность обнаружения атаки.

Ключевые слова: атаки предъявления, нейронная сеть, текстурные признаки, распознавание лиц, антиспуфинг, бинарная классификация

Yu. N. Khanbekova^{1✉}

Method for detecting photo presentation attacks in biometric systems

¹National Research Nuclear University MEPhI, Moscow, Russian Federation
e-mail: yul2002@mail.ru

Abstract. The article describes a method for detecting attacks involving spoofed biometric data in face recognition tasks. The relevance of the study lies in the fact that presentation attacks are among the most accessible and widespread ways of bypassing biometric authentication. The aim of the research is to develop a method that improves attack detection accuracy under poor lighting conditions based on a neural network model. The proposed method includes a module for preliminary image illumination enhancement and a dual-branch model, in which one branch is designed to extract features from the original facial image, while the other branch is intended to extract texture features. The extracted features are fused and fed into a binary classification block. The study demonstrates that the combined use of illumination correction, image feature analysis, and texture feature analysis makes it possible to improve the robustness of the method to various lighting scenarios and increase attack detection accuracy.

Keywords: presentation attacks, neural network, texture features, face recognition, anti-spoofing, binary classification

Введение

Современные информационные системы стали чаще внедрять использование биометрических методов идентификации личности, такие как отпечатки

пальцев, голос, лицо [1–3]. Одновременно с широким распространением биометрических технологий возникла новая угроза – спуфинг. Спуфинг – это попытка обмана системы путем предъявления поддельных данных. Появление данных атак стимулировало развитие отдельного направления – антиспуфинг [4].

В публикациях по теме антиспуфинга особое внимание уделяется применению методов глубокого обучения для определения визуальных признаков подделки изображения [1–5]. Несмотря на достигнутые результаты, остается ряд нерешенных вопросов: существующие методы не всегда обеспечивают достаточную точность распознавания фото атак при плохом освещении, когда часть информативных признаков скрыта. Многие подходы ориентированы на анализ общего изображения лица и не учитывают текстурные особенности, являющиеся метками поддельных данных.

Существенную значимость проблемы подтверждает нормативная база. Согласно ГОСТ Р 58624.1–2019 [6], наличие механизмов обнаружения атак на биометрическое предъявление является обязательным требованием при проектировании надежных биометрических систем [7–9]. Это подтверждает, что защита от спуфинг атак рассматривается не только как исследовательская задача, но и как необходимое условие практического внедрения биометрических решений. При этом антиспуфинг-модуль должен сохранять устойчивость к различным внешним факторам, таким как недостаточное освещение, изменение угла поворота головы, плохое качество камеры [10–13].

Целью работы является разработка метода обнаружения фото атак в условиях плохого освещения на основе нейросетевой модели, обеспечивающего повышение точности распознавания поддельных изображений лиц.

Предлагаемое решение

Предлагаемое решение основывается на совместном использовании предварительной коррекции освещения и двухветвевой нейросетевой архитектуры. Для надежного обнаружения атак предъявления недостаточно анализировать только общее изображение лица. Необходимо одновременно учитывать глобальные и текстурные признаки изображения. Глобальные признаки характеризуют визуальное представление лица. Текстурные артефакты могут содержать прямые признаки атаки: следы печати, муар, неестественную зернистость и др. Поэтому в разработанном методе используется архитектура, включающая две ветви обработки признаков: ветвь признаков изображения и ветвь текстурных признаков (рис. 1).

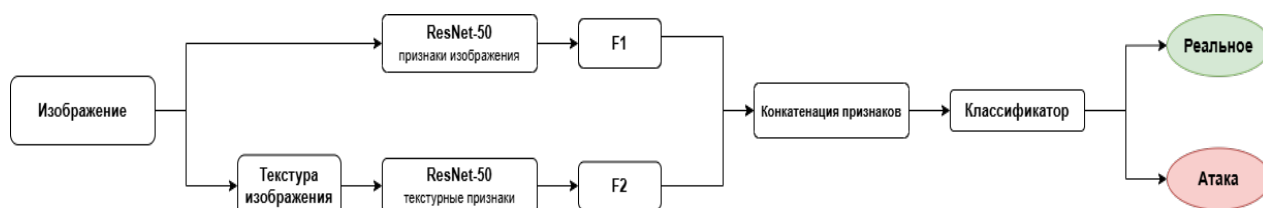


Рис. 1. Поточная диаграмма модели

Общий алгоритм работы метода включает несколько последовательных этапов. На первом этапе на вход системы подается изображение лица. Далее выполняется предварительная обработка изображения: нормализация размеров, приведение к заданному формату и коррекция освещения. После этого формируются два представления одного и того же входного кадра. Первое представляет собой улучшенное изображение лица, сохраняющее глобальные визуальные характеристики. Второе представление формируется на основе текстурного преобразования и предназначено для выделения локальных особенностей поверхности изображения. Далее каждое из этих представлений поступает в соответствующую ветвь нейронной сети, где происходит извлечение признаков. Полученные векторы признаков объединяются в единый вектор с помощью конкатенации. Объединенный вектор признаков подается в классификационный блок, который формирует итоговое решение об атаке (рис. 2).

Одной из составляющих предлагаемого решения является модуль предварительного улучшения освещения. Его назначение – это увеличение информативности признаков, значимых для задачи антиспуфинга. Отличительной чертой предлагаемого решения является сохранение локальных визуальных артефактов, указывающих на использование поддельного изображения. Разработанный модуль улучшения освещения ориентирован на селективную коррекцию темных участков при сохранении структуры изображения. На первом шаге оценивается распределение яркости входного изображения через L-канал пространства LAB и строится карта затемненных областей. Такая карта позволяет определить участки, в которых полезная информация скрыта из-за недостаточного освещения. На втором шаге изображение представляется как сочетание двух компонент: освещения и отражения, или текстуры. Дальнейшая коррекция применяется только к компоненте освещения, текстурная составляющая сохраняется исходной. На следующем этапе выполняется коррекция яркости. Коррекция выполняется как локально-переменная гамма-коррекция компоненты освещения. Для каждого пикселя строится карта гамма-параметра $\gamma(x,y)$: чем темнее участок, тем меньше γ и тем сильнее усиливается яркость. По сравнению с глобальным повышением яркости, здесь степень коррекции зависит от затемнения рассматриваемого участка. Чем ниже локальная освещенность, тем сильнее усиливается соответствующая область, но величина корректировки ограничивается так, чтобы не привести к пересвету и не разрушить естественное соотношение между соседними участками. Дополнительно применяется сдерживание слишком ярких областей, что позволяет уменьшить влияние пересветов и сделать распределение яркости более сбалансированным. После коррекции освещения итоговое изображение восстанавливается по Retinex-схеме.

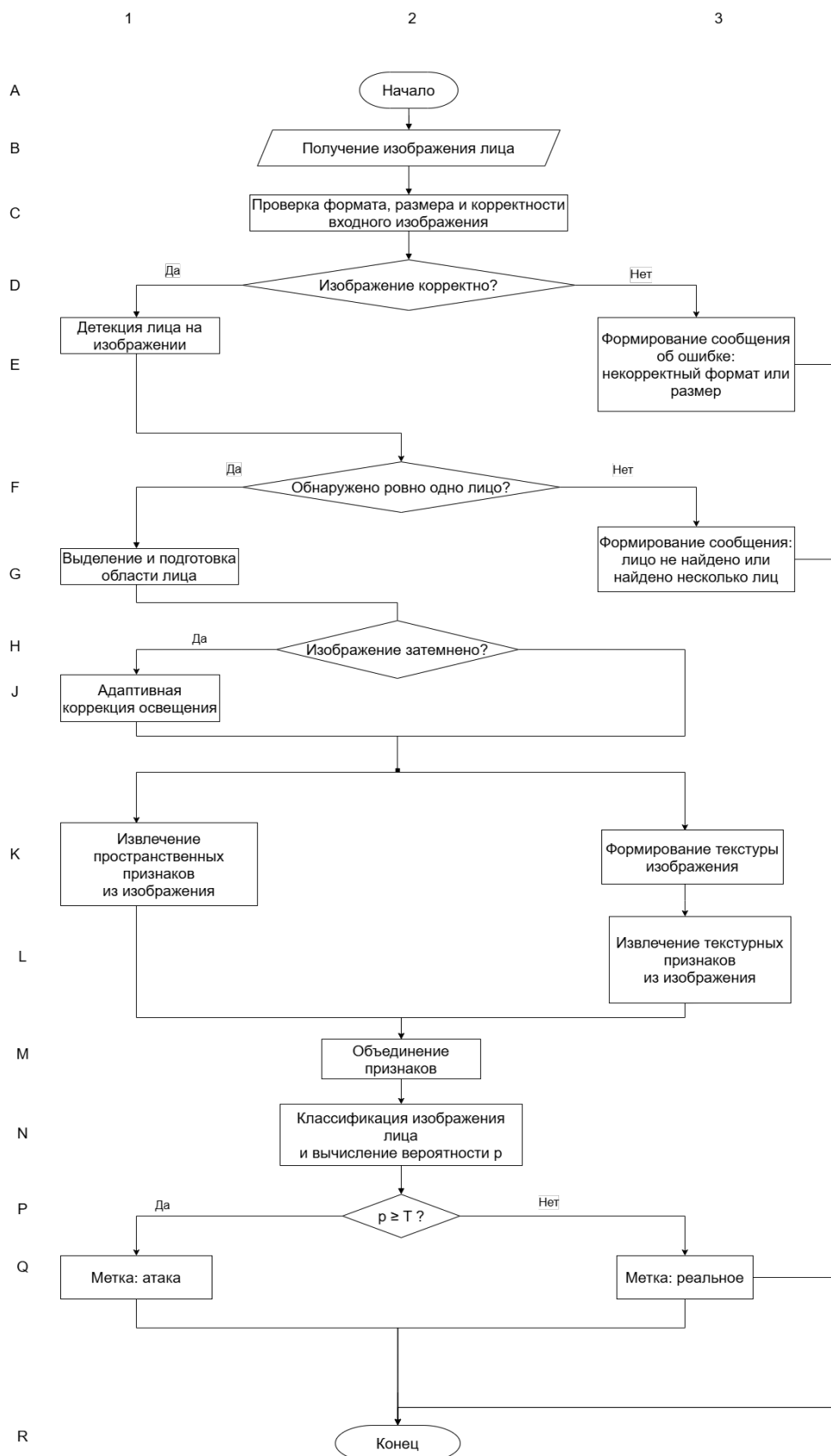


Рис. 2. Блок-схема алгоритма обнаружения атаки

Важным свойством модуля является то, что он не стремится удалить все блики и контрастные перепады, так как часть подобных эффектов может быть связана именно с признаками атаки.

Результаты

В ходе исследования было проведено тестирование разработанного метода обнаружения атак. В качестве базового варианта использовалась модель без модуля улучшения освещения и без разделения на две ветви признаков. Далее оценивалось влияние разработанного алгоритма улучшения освещения совместно с использованием двухветвевой архитектуры. В качестве основного показателя качества использовалась точность классификации. Результаты сравнительного эксперимента представлены в табл. 1.

Таблица 1. Сравнение точности моделей обнаружения атак

Вариант тестирования	Точность классификации, %
Базовая модель	88,3
Разработанный метод	98,5

Результаты тестирования подтвердили, что применение разработанного метода позволило повысить точность классификации с 88,3 % до 98,5 %. Количественное улучшение точности составило 10,2 %.

Заключение

В статье предложен метод обнаружения фото атак в биометрических системах распознавания лиц, основанный на использовании модуля улучшения освещения и двухветвевой нейросетевой архитектуры. Показано, что предварительная коррекция затемненных участков при сохранении локальных артефактов позволяет повысить информативность изображения для задачи антиспуфинга. Разработанный метод объединяет анализ глобальных признаков изображения и текстурных признаков. Предложенный подход может быть использован при создании модулей защиты биометрических систем, работающих в условиях нестабильного освещения.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Кудяева В.М., Федорова И.А., Ермаков А.В. Инновационные методы обнаружения спуфинг-атак на систему лицевой биометрии // Измерение, контроль, информатизация: Материалы XXIII Международной научно-технической конференции, Барнаул, 20 мая 2022 года / Под редакцией Л.И. Сучковой. – Барнаул: Алтайский государственный технический университет им. И.И. Ползунова – 2022. – С. 139-142.
2. Миронова Н.Г. Методы антиспуфинга в системах биометрической идентификации и верификации // Информационные технологии обеспечения комплексной безопасности в цифровом обществе: сборник материалов IV Всероссийской молодежной научно-практической конференции с международным участием, Уфа, 21–22 мая 2021 года. – Уфа: Башкирский государственный университет – 2021. – С. 43-51.

3. Минкин А.В., Каракеян А.С. Анализ методов face anti-spoofing для обеспечения безопасности биометрической аутентификации в условиях цифровизации // Журнал Современные наукоемкие технологии. – 2025. – № 9. С. 167-175.
4. К. Д. Русаков, А. А. Генов, С. Ш. Хиль Методика решения задачи антиспуфинга по ограниченному количеству фотографий // Программные продукты и системы. – 2020. – Т. 33 – №1. – С. 54-60.
5. Киба А.С., Фаворская М.Н. Обнаружение поддельных изображений лиц на основе глубокого обучения // Решетневские чтения: Материалы XXIV Международной научно-практической конференции, посвященной памяти генерального конструктора ракетно-космических систем академика М. Ф. Решетнева: в 2 частях, Красноярск, 10–13 ноября 2020 года. Том Часть 2. – Красноярск: Федеральное государственное бюджетное образовательное учреждение высшего образования "Сибирский государственный университет науки и технологий имени академика М.Ф. Решетнева" – 2020. – С. 403-404.
6. ГОСТ Р 58624.1–2019. Информационные технологии. Биометрия. Обнаружение атаки на биометрическое предъявление. Часть 1. Структура. – Москва: Стандартинформ, 2019. – 12 с.
7. Мудрич А. Б., Ежова К. В. Детектирование и распознавание лиц в системах видеонаблюдения // International Journal of Open Information Technologies. – 2024. – Т.12. – №3. – С. 82-84.
8. Частикова, В.А., Титова А.А., Войлова Д.О. Аналитический обзор методов идентификации личности на основе биометрических характеристик // Вестник Адыгейского государственного университета. Серия: Естественно-математические и технические науки. – 2022. – № 1(296). – С. 92-112. – DOI 10.53598/2410-3225-2022-1-296-92-112.
9. Е. А. Омеляненко, Е. А. Басыня, А. В. Сафронов Разработка системы контроля и управления доступом персонала в рудник на основе биометрической идентификации // Перспективное развитие науки, техники и технологий : Сборник научных статей VII-ой Международной научно-практической конференции, Курск, 13–14 октября 2017 года / Ответственный редактор А.А. Горохов. – Курск: Закрытое акционерное общество "Университетская книга", 2017. – С. 142-150.
10. Басыня, Е. А., А. В. Сафронов Разработка и исследование системы управления метаданными изображений // Актуальные проблемы электронного приборостроения АПЭП - 2016 : Труды XIII Международной научно-технической конференции. В 12-ти томах, Новосибирск, 03–06 октября 2016 года. Том Том 9. – Новосибирск: Новосибирский государственный технический университет, 2016. – С. 155-157.
11. Гусев Юрий Борисович Приватные биометрические криптосистемы: архитектура, математические основы и алгоритмы // Инновации и инвестиции. – 2025. – №2. – С. 424-429.
12. Денисова А. Ю., Федосеев В. А. Обнаружение атак на биометрическое предъявление на системы аутентификации лиц при помощи специальных устройств съемки // Компьютерная оптика. – 2022. – Т. 46. – №. 4. – С. 612-620.
13. Минкин А.В., Каракеян А.С. Анализ методов face anti-spoofing для обеспечения безопасности биометрической аутентификации в условиях цифровизации // Журнал Современные наукоемкие технологии. – 2025. – №. 9. – С. 167-175.

© Ю. Н. Ханбекова, 2026

А. А. Цыганков¹✉

Анализ проблем в области выявления аномалий в действиях пользователей информационных систем на основе поведенческой аналитики

¹Национальный исследовательский ядерный университет «МИФИ»,
г. Москва, Российская Федерация
e-mail: aatsygankov@mephi.ru

Аннотация. В статье рассматривается проблема выявления аномалий в действиях пользователей информационных систем с использованием методов поведенческой аналитики. Основная проблема исследований в данной области заключается в существующих ограничениях современных систем обнаружения инцидентов информационной безопасности, в частности высокая доля ложных срабатываний, дисбаланс данных и неспособность адаптации к изменениям в поведении. Целью работы является анализ проблем в области выявления аномалий в действиях пользователей информационных систем на основе поведенческой аналитики. В результате исследования определены фундаментальные проблемы существующих подходов, среди которых выделены: концептуальный сдвиг, острая нехватка качественных размеченных наборов данных, коллизии между сбором пользовательских метрик и правом на неприкосновенность частной жизни, а также проблема дисбаланса классов.

Ключевые слова: инсайдер, машинное обучение, кибербезопасность

A. A. Tsygankov¹✉

Analysis of problems in the field of detecting anomalies in user behavior within information systems based on behavioral analytics

¹National Research Nuclear University MEPhI, Moscow, Russian Federation
e-mail: aatsygankov@mephi.ru

Abstract. The article considers the problem of detecting anomalies in the actions of information system users using behavioral analytics methods. The main problem of research in this area lies in the existing limitations of modern information security incident detection systems, in particular, a high rate of false positives, data imbalance, and the inability to adapt to changes in behavior. The objective of this work is the analysis of problems in the field of detecting anomalies in the actions of information system users based on behavioral analytics. As a result of the study, fundamental problems of existing approaches were identified, among which the following are highlighted: concept drift, an acute shortage of high-quality labeled datasets, conflicts between the collection of user metrics and the right to privacy, as well as the problem of class imbalance.

Keywords: insider, machine learning, cybersecurity

Введение

Современные сетевые операционные системы ежедневно генерируют значительные объемы журналов событий, включающих сотни тысяч записей, отражающих различные аспекты пользовательской активности: обращения к ресурсам, процессы аутентификации и деаутентификации, а также иные события, ре-

гистрируемые в соответствии с установленными политиками информационной безопасности. Накопление и анализ таких данных создают предпосылки для построения поведенческих моделей пользователей.

При наличии достаточного объема и качества данных становится возможным формирование профиля нормального поведения пользователя, который может быть использован в задачах классификации с применением методов и алгоритмов машинного обучения. Сравнение текущей активности с эталонной моделью позволяет выявлять отклонения, интерпретируемые как аномалии. Такие аномалии могут свидетельствовать о потенциальных инцидентах информационной безопасности, включая несанкционированный доступ или компрометацию учётных записей.

В рамках данного исследования рассматриваются существующие подходы и решения в области обнаружения аномалий пользовательского поведения в информационных системах. Анализируются их ограничения, такие как высокая доля ложных срабатываний, сложность адаптации к изменяющемуся поведению пользователей, требования к объёму обучающих данных и вычислительным ресурсам. На основе проведенного анализа формулируются ключевые научные и практические вызовы, стоящие перед современными системами обнаружения аномалий, а также обосновываются направления дальнейших исследований, направленных на повышение точности, интерпретируемости и масштабируемости таких систем.

Целью исследования является анализ проблем в области выявления аномалий в действиях пользователей информационных систем на основе поведенческой аналитики. Для ее решения была проведена декомпозиция на атомарные подзадачи, а именно:

- 1) исследование предметной области;
- 2) моделирование предметной области;
- 3) анализ проблематики предметной области.

Теоретическая значимость исследования заключается в развитии подходов к анализу журналируемых действий пользователей информационных систем. Практическая значимость определяется возможностью применения исследуемых методов в системах обеспечения информационной безопасности с целью обеспечения информационной безопасности.

Исследование предметной области

UEBA (англ. User and Entity Behavior Analytics) представляет собой направление в области поведенческой кибербезопасности, ориентированное на выявление аномалий в активности пользователей и сущностей (устройств, сервисов, учетных записей) на основе анализа их поведенческих паттернов. Формально задача UEBA сводится к задаче обнаружения аномалий (anomaly detection) и, в отдельных случаях, к задаче классификации или ранжирования событий по степени риска.

В современной научной литературе доминируют методы машинного обучения (ML) и глубокого обучения (DL) [1, 2]. В работах используются методы

опорных векторов для построения границы нормального поведения; методы главных компонент для снижения размерности и выявления аномалий через реконструкционную ошибку; алгоритм изолированного леса, используя который аномалии изолируются быстрее нормальных точек [3]. С развитием отрасли нейросетевых моделей расширяются и методы, применяемые в кибербезопасности. Например, в исследовании [4] рассматривается использование GNN (англ. Graph Neural Network), применяемые для моделирования сложных взаимосвязей между пользователями и сущностями. Помимо этого, активно используются генеративно-состязательные сети (GAN) для классификации поведения пользователей и автоматического подбора гиперпараметров, что позволяет адаптироваться к реальным системам с сохранением высокой точности [5].

Одной из фундаментальных проблем UEBA-систем является нехватка размеченных данных, особенно для редких или ранее неизвестных атак. Для решения этой проблемы активно исследуются методы выявления аномалий, использующие минимальное количество данных для начала функционирования. В частности, разрабатываются новые методы FSL (англ. Few-shot learning) [6] и ZSL (англ. Zero-shot learning) [7]. Первые используются для обучения моделей при наличии малого числа размеченных примеров. Интеграция же методов ZSL необходима для определения ранее не встречаемых классов, что в рамках кибербезопасности является критически важной возможностью.

Существуют также классические методы обнаружений аномального поведения пользователей, в частности использование подходов, основанных на правилах (англ. rule-based), ручной анализ журналов событий. Очевидно, что второй полностью исключает возможность определения инцидента информационной безопасности в момент совершения атаки. Подход, основанный на правилах, эффективен при детектировании известных атак или нарушений политики безопасности [8], однако, данный метод строго ограничен скоростью внедрения новых правил и абсолютно не защищает от атак нового типа, что в условиях активного развития киберпреступности является одним из ключевых факторов, на которых основываются специалисты при выборе системы защиты.

Моделирование предметной области

Процесс взаимодействия пользователя с информационной системой может быть формализован в виде стохастического процесса. Поведение пользователя может описываться последовательностью сессий S , каждая из которых содержит многомерный вектор признаков $\{e_1, e_2, \dots, e_n\}$. Каждый из признаков e_i может содержать временные метки, сетевые запросы, транзакции и многое другое. На рис. 1 показана BPMN-диаграмма (англ. Business Process Model and Notation) процесса трансформации пользовательской активности в векторы признаков для выявления аномалий.

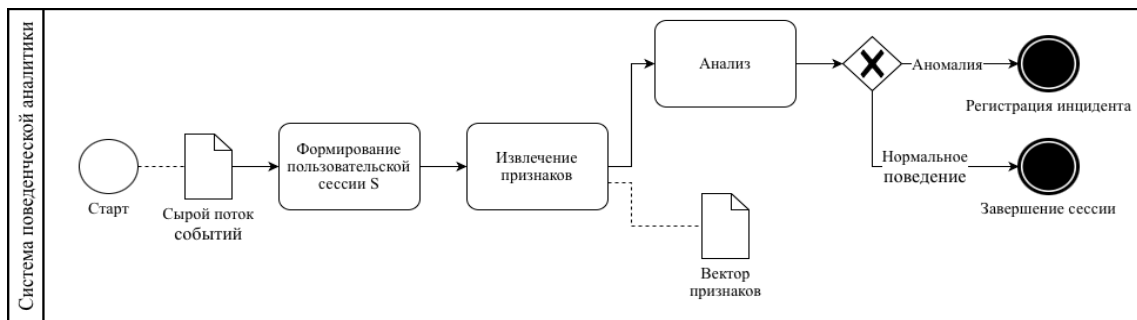


Рис. 1. BPMN-модель процесса трансформации пользовательской активности в векторы признаков для выявления аномалий

Изначально процесс инициализируется получением артефакта входных данных – сырого потока событий, который агрегирует сетевые запросы, процессы идентификации, временные метки и иные артефакты. На первом вычислительном этапе разрозненные события структурируются в рамках задачи формирования пользовательской сессии S . Далее следует этап извлечения признаков, результатом выполнения которого является многомерный вектор признаков $\{e_1, e_2, \dots, e_n\}$. Сформированный вектор передается в блок логического анализа, где модель оценивает степень отклонения текущего поведения от профиля нормального поведения пользователя. В случае, если выявленные отклонения трактуются алгоритмом как аномалия, процесс переходит по соответствующей ветви и инициирует терминальное событие — регистрацию инцидента информационной безопасности. В противном случае сессия завершается штатно.

Анализ проблематики предметной области

Важной проблемой в моделировании активности пользователя остается возможное образование концептуального сдвига (англ. *concept drift*) [9]. Он описывает непредсказуемые изменения в базовом распределении потоковых данных с течением времени. Например, смена формата работы, должности, переход к решению задач другого типа и многие другие перемены в жизни исследуемого пользователя могут изменить его поведение, что приведет к выявлению аномалии и, как следствие, ложноположительному срабатыванию.

Помимо этого, известной проблемой является дисбаланс данных (англ. *imbalanced data*). Например, в синтетическом датасете *CERT Insider Threat Dataset*, который был разработан научной группой Университета Карнеги – Меллона, на 1000 записей приходится всего 30 аномалий [10]. Проблема появляется именно тогда, когда экземпляры одного класса преобладают над экземплярами другого класса – в приведенном наборе данных, например, лишь 3% данных должны быть корректно определены как аномалии. В таких ситуациях модели зачастую снижают точность выявления аномалий и происходит смещение.

Фундаментальной проблемой во всей области детекции аномалий в действиях пользователей является исключительная редкость датасетов, на которых можно строить и обучать модели [11]. Организации, разрабатывающие собствен-

ные решения, зачастую не публикуют собственные журналы событий с размеченными данными по понятным причинам: такие данные содержат конфиденциальную информацию сотрудников и, конечно, могут использоваться злоумышленниками для получения чувствительной информации. В свою очередь синтетические датасеты, используемые в большинстве исследований, обладают недостатками любых синтетических данных: они в лучшем случае не полностью отражают реальность.

Строго говоря, проблемой является даже сам по себе анализ поведения пользователей, поскольку он зачастую вступает в прямое противоречие с законами о защите персональных данных. Это значит, что система детекции аномалий в пользовательском поведении должна не просто собирать максимальное количество требуемой ей для обучения информации, но и при этом не нарушать право сотрудников на неприкосновенность частной жизни.

Заключение

В ходе проведенного исследования был выполнен комплексный анализ проблем выявления аномалий в действиях пользователей информационных систем с использованием методов поведенческой аналитики. Установлено, что классические подходы, основанные на правилах и ручном анализе журналов, обладают критическими ограничениями, так как не способны выявлять новые типы кибератак в момент их совершения. Помимо этого, определены фундаментальные проблемы, существенно ограничивающие эффективность современных систем обнаружения аномалий, а именно:

- 1) естественные изменения в поведении сотрудников приводят к высокому уровню ложноположительных срабатываний;
- 2) преобладание записей нормального поведения над аномалиями приводит к смещению алгоритмов и снижению точности детекции;
- 3) отсутствие реальных (не синтетических) корпоративных датасетов делает невозможным качественное обучение моделей;
- 4) необходимость сбора максимального объема метрик для профилирования вступает в прямое противоречие с правом пользователей на неприкосновенность частной жизни и законами о защите персональных данных.

Основным направлением дальнейших исследований в данной области должна стать разработка и интеграция подходов, не требующих больших объемов обучающих выборок. В частности, развитие методов FSL и ZSL, что позволит системам обеспечения информационной безопасности эффективно классифицировать ранее неизвестные инциденты. Решение выявленных проблем позволит существенно повысить точность, интерпретируемость и масштабируемость систем поведенческой аналитики.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Mizanur M. et al. Machine learning-based anomaly detection for cyber threat prevention //Journal of Primeasia. – 2025. – Т. 6. – №. 1. – С. 1-8.

2. Mihailescu M. I. et al. Unveiling threats: Leveraging user behavior analysis for enhanced cybersecurity //2023 15th International Conference on Electronics, Computers and Artificial Intelligence (ECAI). – IEEE, 2023. – C. 01-06.
3. Lanuwabang L., Sarasu P. Detection of Anomalies Based on User Behavioral Information: A Survey //International Journal of Wireless and Microwave Technologies (IJWMT). – 2025. – T. 15. – №. 3. – C. 54-65.
4. Ares-Robledo F., Rifà-Pous H., Clarisó R. Graph neural networks for anomaly detection: a systematic review of dynamic temporal approaches //Artificial Intelligence Review. – 2026.
5. Xia X. et al. GAN-based anomaly detection: A review //Neurocomputing. – 2022. – T. 493. – C. 497-535.
6. Gamal M. et al. Few-shot learning for discovering anomalous behaviors in edge networks //Computers, Materials and Continua. – 2021. – T. 69. – №. 2. – C. 1823-1837.
7. Demirel D. Y., Sandikkaya M. T. Web based anomaly detection using zero-shot learning with CNN //IEEE Access. – 2023. – T. 11. – C. 91511-91525.
8. Alzaabi F. R., Mehmood A. A review of recent advances, challenges, and opportunities in malicious insider threat detection using machine learning methods //IEEE Access. – 2024. – T. 12. – C. 30907-30927.
9. Lu J. et al. Learning under concept drift: A review //IEEE transactions on knowledge and data engineering. – 2018. – T. 31. – №. 12. – C. 2346-2363.
10. Bin Sarhan B., Altwaijry N. Insider threat detection using machine learning approach //Applied Sciences. – 2022. – T. 13. – №. 1. – C. 259.
11. Manoharan P. et al. Insider threat detection using supervised machine learning algorithms //Telecommun. Syst. – 2024. – T. 87. – №. 4. – C. 899-915.

© A. A. Цыганков, 2026

И. В. Чугунков¹, М. М. Ровнягин¹, К. А. Федоров¹✉, К. В. Сысоев¹

Архитектура адаптивного планировщика в геораспределенном кластере

¹Национальный исследовательский ядерный университет «МИФИ»,
г. Москва, Российская Федерация
e-mail: klim.fedorov2001@yandex.ru

Аннотация. В работе рассматривается задача планирования размещения контейнеризированных приложений в геораспределенном кластере Kubernetes. Показано, что стандартный планировщик kube-scheduler, ориентированный на заявленные ресурсы и текущее состояние узлов, не учитывает в достаточной мере сетевую топологию, задержки межсервисного взаимодействия и временную динамику нагрузки, что в условиях геораспределения приводит к снижению эффективности использования ресурсов. Геораспределенный кластер формализован в виде четверки из множества узлов, множества Pod'ов, функции сетевого расстояния и матрицы интенсивностей взаимодействия Pod'ов; состояние кластера разделено на инфраструктурный и прикладной слои, различающиеся характерным временем изменения. На этой основе предложена архитектура адаптивного планировщика, в которой решение формируется несколькими специализированными контурами – первичного планирования, периодической адаптации и упреждающего управления, – а итоговое размещение принимает отдельный компонент-арбитр. Исполнение решения выполняют компоненты scheduler и descheduler через Kubernetes API; единым хранилищем состояния выступает PostgreSQL, источником показателей – VictoriaMetrics. Практическая значимость заключается в расширяемой архитектурной основе для построения планировщика, учитывающего специфику геораспределенных сред.

Ключевые слова: планирование, кластер, архитектура

I. V. Chugunkov¹, M. M. Rovniagin¹, K. A. Fedorov¹✉, K. V. Sysoev¹

Adaptive scheduler architecture in a geo-distributed cluster

¹National Research Nuclear University "MEPhI" Moscow, Russian Federation
e-mail: klim.fedorov2001@yandex.ru

Abstract. This paper addresses the problem of scheduling the placement of containerized applications in a geo-distributed Kubernetes cluster. It is shown that the standard kube-scheduler, which relies on declared resource requests and the current state of nodes, does not sufficiently account for network topology, inter-service communication latency, or the temporal dynamics of workloads, which leads to reduced resource utilization efficiency under geo-distribution usage scenario. The geo-distributed cluster is formalized as a quadruple consisting of a set of nodes, a set of Pods, a network distance function, and a matrix of pairwise Pod interaction intensities; the cluster state is decomposed into an infrastructure layer and an application layer, which differ in their characteristic time scales of change. Based on this formalization, an architecture of an adaptive scheduler is proposed, in which the decision is formed by several specialized loops – primary scheduling, periodic adaptation, and proactive control – while the final placement is produced by a separate arbiter component. The decision is executed by the scheduler and descheduler components through the Kubernetes API; PostgreSQL serves as the unified state store, and VictoriaMetrics as the source of metrics. The practical significance of the work lies in providing an extensible architectural foundation for building a scheduler that takes into account the specifics of geo-distributed environments.

Keywords: scheduling, cluster, architecture

Введение

В последние годы Kubernetes стал стандартной платформой оркестрации контейнеризированных приложений, однако по мере роста масштабов кластеров и усложнения микросервисных взаимодействий задача эффективного размещения Pod'ов становится существенно сложнее. В геораспределенном кластере качество планирования определяется не только доступностью CPU и RAM, но и сетевой топологией, задержками между узлами и интенсивностью обмена данными между сервисами. В то же время стандартные механизмы планирования Kubernetes ограничены в данных текущим состоянием узлов и статически заданных requests.

Вследствие этого возникает необходимость в адаптивном подходе к планированию, сочетающем реакцию на текущие события, периодическую корректировку размещения и упреждающее управление на основе прогноза нагрузки. Актуальность работы определяется потребностью в адаптивном планировщике и необходимостью проектирования архитектуры, способной учитывать специфику геораспределенной среды для кластеров различных размеров, и принимать решения на основе нескольких классов показателей. Целью работы является разработка архитектуры адаптивного планировщика для Kubernetes, обеспечивающего более эффективное использование ресурсов кластера и снижение топологической стоимости размещения.

Постановка задачи

Целью настоящей работы является разработка архитектуры адаптивного планировщика для оркестратора Kubernetes с учетом геораспределенной топологии сети. В случае плотного расположения узлов кластера (размещения внутри одного датацентра или одной серверной стойки) топологию взаимодействия Pod'ов можно не учитывать, поскольку задержками внутри сети кластера можно пренебречь [1–6]. Геораспределение узлов требует учета степени взаимодействия между Pod'ами, а также пропускную способность сети узлов, через которую будет осуществляться это взаимодействие [7, 8].

В таком случае геораспределенный кластер рассматривается как четверка:

$$C = (N, P, E, \Psi), \quad (1)$$

где $N = \{n_1, \dots, n_{|N|}\}$ – множество узлов, $P = \{p_1, \dots, p_{|P|}\}$ – множество Pod'ов, подлежащих размещению, $E: N \times N \rightarrow R_{\geq 0}$ – сетевая топология узлов: неотрицательная симметричная функция расстояния между узлами – латентность в мс, число шагов по топологическому дереву и т.п. $\Psi \in [0, 1]^{|P| \times |P|}$ – топология Pod'ов: нормированная матрица попарных интенсивностей взаимодействия. В общем случае не симметрична. Координата ψ_{ij} показывает, насколько сильно Pod p_i играет от близости к p_j , $\psi_{ii} = 0$.

В рамках одного цикла планирования удобно разделить четверку C на два слоя, различающиеся характерным масштабом времени изменений. Инфраструктурный слой C_{infra} :

$$C_{infra}(t) = (N(t), E(t)), \quad (2)$$

описывающий состав узлов кластера и сетевую топологию. Он меняется редко: при добавлении или выводе узлов: при реконфигурации сети. Для него характерно время порядка часов и суток. Прикладной слой C_{wl} :

$$C_{wl}(t) = (P(t), \Psi(t)), \quad (3)$$

описывает состав Pod'ов, подлежащих размещению, и интенсивность их попарного взаимодействия. Он изменяется динамично: при порождении и удалении Pod'ов; при изменении профиля обращений сервисов друг к другу. Он характеризуется временем порядка секунд и минут. Полное состояние кластера в момент t выражается как $C(t) = C_{infra}(t) \cup C_{wl}(t)$.

Формула (1) описывает срез кластера в момент принятия решения t_0 . Уже исходя из изменчивости кластера с течением времени, становится ясно, что планировщик в геораспределенном кластере должен опираться в решении не столько на статические ограничения, сколько быть адаптивным к изменениям как на инфраструктурном слое, так и на прикладном.

Каждому узлу $j \in \{1, \dots, |P|\}$ сопоставляется вектор емкости ресурсов:

$$r(n_j) = (r_j^{CPU}, r_j^{RAM}, r_j^{disk}, r_j^{disk-bw}, r_j^{net}, r_j^{GPU}, \dots) \in R_{\geq 0}^K, \quad (4)$$

где n_j – узел кластера, $r_j^{(k)}$ – значение емкости типа ресурса k для узла j , K – число типов ресурсов, учитываемых планировщиком. Различные контуры могут не учитывать все типы ресурсов, однако их набор должен единственным для всего планировщика. Каждому Pod'у i в момент его размещения сопоставляется вектор заявленных или, в случае повторного размещения, спрогнозированных потребностей:

$$d(p_i) = (d^{res}(p_i), \psi_i) \in R_{\geq 0}^K, \quad (5)$$

где $d^{res}(p_i) = (d^{(1)}(p_i), \dots, d^{(K)}(p_i)) \in R_{\geq 0}^K$ – ресурсный блок: вектор потребности Pod'а в ресурсах, $d^{(K)}(p_i)$ – значение потребности Pod'а p_i в ресурсе типа k . $\psi_i = (\psi_{i,1}, \dots, \psi_{i,|P|})$ – i -я строка матрицы Ψ : вектор интенсивностей взаимодействия Pod'а p_i с остальными Pod'ами – топологический блок потребностей Pod'а.

Размещение Pod'ов задается отображением:

$$\pi: P \rightarrow N, \quad (6)$$

Одновременно удовлетворяющим покоординатным ресурсным ограничениям:

$$u_j^{(k)}(\pi) = \frac{1}{r_j^{(k)}} \sum_{p: \pi(p)=n_j} d^{(k)}(p) \leq 1, \forall k \in \{1, \dots, K\}, \forall j \in \{1, \dots, |N|\}, \quad (7)$$

и минимизирующим топологическую стоимость:

$$T_{topo}(\pi) = \sum_{i=1}^{|P|} \sum_{j=1}^{|P|} \psi_{ji} E(\pi(p_i), \pi(p_j)), \quad (8)$$

где $u_j^{(k)}$ – утилизация ресурса типа k на узле n_j после применения размещения π , лежащая в диапазоне $[0, 1]$, где 0 – ресурс не востребован, 1 – используется полностью. $\sum_{p: \pi(p)=n_j} d^{(k)}(p)$ – суммарная потребность в ресурсе типа k всех Pod'ов размещенных на узле n_j .

Таким образом, допустимым размещением может называться то размещение, при котором $u_j^{(k)} \leq 1$, а недопустимым – $u_j^{(k)} > 1$.

Архитектура системы

Исходя из многофакторности задачи планирования размещения в геораспределенном кластере, в архитектуре планировщика предлагается предусмотреть несколько контуров планирования. Таким образом, каждый из них сможет решать узкую задачу, что позитивно скажется на их эффективности по отдельности. В то же время возможность принятия конечного решения по размещению будет возложена на отдельный сервис, который будет иметь доступ к решениям всех контуров и их прошлым решениям.

В качестве плюсов данного подхода можно отметить расширяемость системы планировщика – при необходимости учета новых показателей в размещении можно будет добавить новый контур, который будет учитывать их. Это могут быть, например, показатели о стоимости аренды узлов кластера, о связанном с кластером углеродном следе [9, 10]. Более того, необязательно ограничиваться показателями самого кластера – при необходимости контур может учитывать сколь угодно специфичные для сервисов кластера показатели, включая бизнес, QoS-метрики [11–13]. А также метрики, связанные с анализом киберинцидентов [14–17]. Так, планировщик будет способен закрыть потребности практически любой команды разработки.

На данный момент в планировщике планируется реализовать три контура: контур первичного планирования – реакция на событие создания пода/деплой-мента: решение принимается быстро, опираясь на текущие показатели утилизации узлов и политики scoring/filtering; контур периодической адаптации – решения принимаются для достижения сбалансированности в утилизации узлов и минимизации топологической стоимости $T_{topo}(\pi)$ с некоторой периодичностью, для сбора необходимых для анализа данных о Pod'ах и реальном состоянии узлов, и контур упреждающего управления – решения опираются на прогноз или

оценку будущего спроса/нагрузки для превентивного релевантного воздействия, что особенно релевантно при сезонности запросов.

Исходя из этого, можно представить планировщик в виде следующих компонентов (рис. 1).

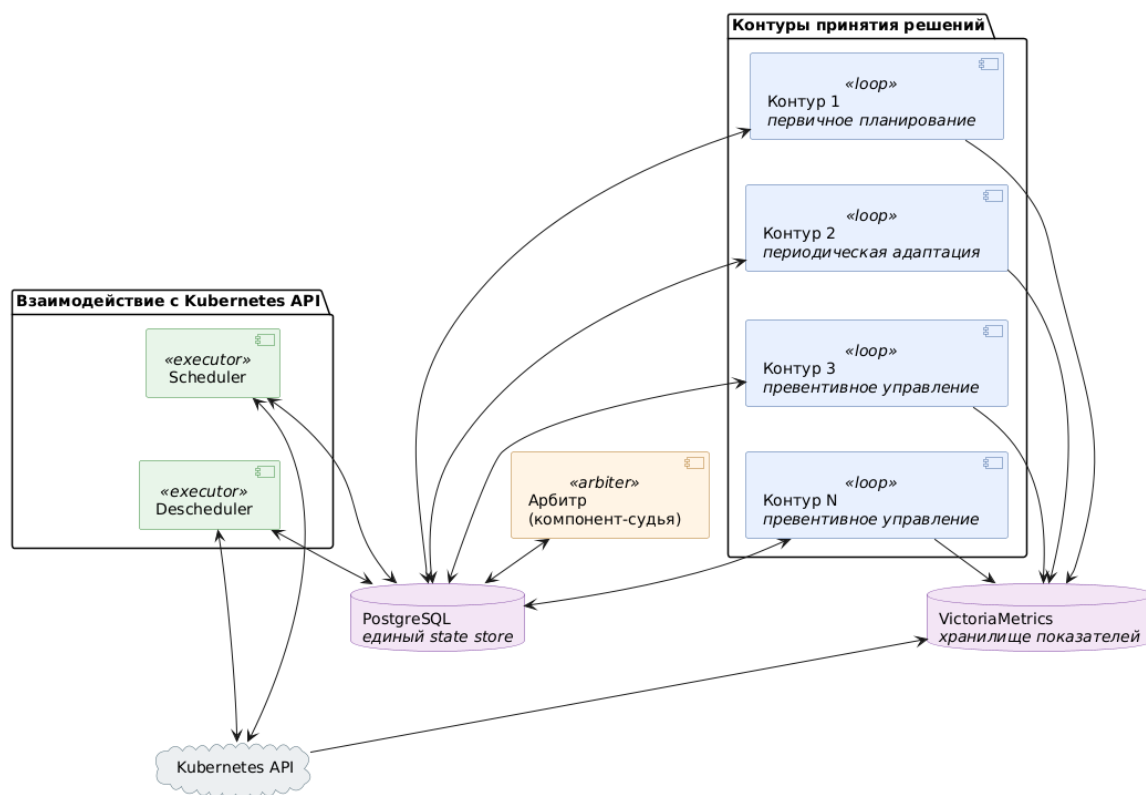


Рис. 1. Диаграмма компонентов планировщика

Компоненты, входящие в состав адаптивного планировщика, делятся на две группы: компоненты, благодаря которым формируется решение о размещении; и компоненты, реализующие размещение из решения, взаимодействуя с Kubernetes API. Отдельным компонентом выступает arbiter, принимающий итоговое решение. В качестве источника показателей для решений контуров выступает база данных временных рядов VictoriaMetrics. В нее записывают свои показатели сборщики, работающие внутри кластера Kubernetes.

Решения каждого контура в отдельности и итоговое размещение хранятся в реляционной базе данных PostgreSQL. Итоговое решение используют компоненты scheduler и descheduler. Они относятся ко второй группе. Scheduler привязывает Pod'ы к узлам, а descheduler отвязывает уже размещенные Pod'ы от несоответствующих новому решению компонента arbiter, инициализируя процесс их размещения внутри кластера заново. Необходимость компонента descheduler обусловлена тем, что Kubernetes не перемещает Pod'ы, которые уже перешли из состояния PENDING в состояние RUNNING.

Таким образом, один цикл планирования в диаграмме будет выглядеть следующим образом (рис. 2).

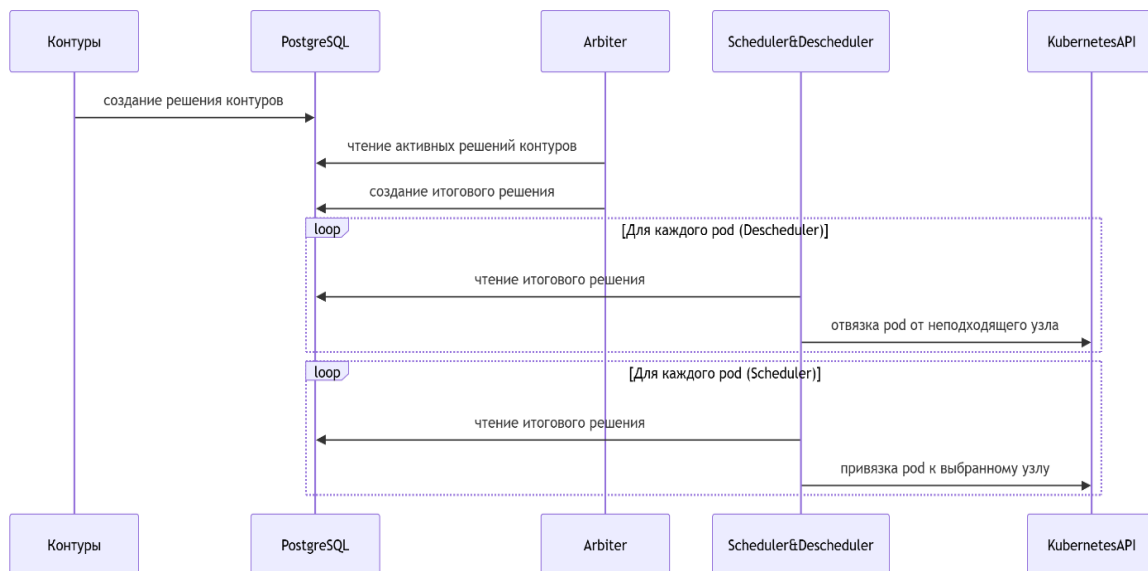


Рис. 2. Диаграмма последовательности цикла планирования

Сначала каждый контур будет принимать собственное решение о размещении и размещать его в БД. Затем arbiter, проанализировав его, создаст итоговое решение. Далее в действие вступает descheduler – вытесняя Pod’ы с неподходящих узлов. После того как таких Pod’ов не осталось scheduler назначает узел для неразмещенных Pod’ов.

Заключение

В работе предложена архитектура адаптивного планировщика для геораспределенного кластера Kubernetes, учитывающая как ресурсные ограничения узлов, так и топологию сетевого взаимодействия между Pod’ами. Показано, что в условиях геораспределения задача размещения не может эффективно решаться только на основе локальных метрик утилизации или статичных ограничений, поскольку существенное влияние на качество размещения оказывают сетевые задержки и интенсивность взаимодействия между сервисами, изменяющиеся с течением времени. Для учета этой специфики состояние кластера представлено в виде инфраструктурного и прикладного слоев, а процесс принятия решения организован как взаимодействие нескольких специализированных контуров планирования и отдельного компонента arbiter, формирующего итоговое решение.

Предложенная архитектура обеспечивает расширяемость и адаптивность системы: при появлении новых требований или метрик QoS можно добавлять новые контуры без переработки всей системы планирования. Разделение компонентов на формирующие решение и исполняющие его через Kubernetes API позволяет упростить развитие системы и повысить прозрачность процесса принятия решений. Практическая значимость работы заключается в формировании архитектурной основы для построения адаптивного планировщика, способного учитывать специфику геораспределенных вычислительных сред и динамику нагрузки.

Дальнейшие исследования целесообразно направить на разработку формальных правил арбитража между контурами, экспериментальную оценку качества размещения на реальных и синтетических нагрузках, а также на исследование устойчивости предложенного подхода при изменении топологии сети и профиля межсервисного взаимодействия.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Khaldoun S. et al. A survey of Kubernetes scheduling algorithms [J] //Journal of Cloud Computing. – 2023. – Т. 12. – №. 1.
2. Jakobović K. PROPOSAL FOR DEVELOPMENT OF A CUSTOM KUBERNETES OPERATOR: дис. – Algebra University, 2023.
3. Creane B., Gupta A. Kubernetes Security and Observability. – "O'Reilly Media, Inc.", 2021.
4. Carrión C. Kubernetes scheduling: Taxonomy, ongoing issues and challenges //ACM Computing Surveys. – 2022. – Т. 55. – №. 7. – С. 1-37.
5. Dakić V. et al. Optimizing Kubernetes Scheduling for Web Applications Using Machine Learning //Electronics. – 2025. – Т. 14. – №. 5. – С. 863.
6. Jian Z. et al. DRS: A deep reinforcement learning enhanced Kubernetes scheduler for microservice-based system //Software: Practice and Experience. – 2024. – Т. 54. – №. 10. – С. 2102-2126.
7. Ahmed A. M., Kashmola M. Y. Advances in Distributed Scheduling Algorithms: A Three-Layer Architecture Integrating Deep Reinforcement Learning and Energy Optimization (2023-2024) //Journal of Al-Qadisiyah for Computer Science and Mathematics. – 2025. – Т. 17. – №. 1. – С. 62–71-62–71.
8. Furnadzhiev R., Shopov M., Kakanakov N. Efficient Orchestration of Distributed Workloads in Multi-Region Kubernetes Cluster //Computers. – 2025. – Т. 14. – №. 4. – С. 114.
9. Lakshan S., Hussain S. A Review of AI-Driven Techniques for Cost Optimization in Kubernetes Environments //2025 International Research Conference on Smart Computing and Systems Engineering (SCSE). – IEEE, 2025. – С. 1-5.
10. Moriabadi M. Reducing carbon emissions in Kubernetes clusters through temporal and spatial workload management: дис. – Universitätsbibliothek der Universität Stuttgart, 2023.
11. Vogl R., Blank-Burian M. An Update on the Münster University Cloud-technical architecture and user adoption //Proceedings of EUNIS. – 2025. – Т. 105. – С. 126-135.
12. Moritz de Carvalho Neto P., Castro M., Siqueira F. Dynamic Load Balancing in Kubernetes Environments with Kubernetes Scheduling Extension (KSE) //Concurrency and Computation: Practice and Experience. – 2025. – Т. 37. – №. 3. – С. e8344.
13. Rejiba Z., Chamanara J. Custom scheduling in kubernetes: A survey on common problems and solution approaches //ACM Computing Surveys. – 2022. – Т. 55. – №. 7. – С. 1-37.
14. Ascensão P. et al. Assessing kubernetes distributions: A comparative study //2024 IEEE 22nd Mediterranean Electrotechnical Conference (MELECON). – IEEE, 2024. – С. 832-837.
15. БАСЫНЯ Е. А., САПЕГИН В. Ю. КИБЕРНЕТИКА И ИНФОРМАЦИОННАЯ БЕЗОПАСНОСТЬ" КИБ-2025 //Москва. – 2025. – Т. 3. – С. 130-131.
16. Стрельцов А. С., Французова Г. А., Басыня Е. А. Разработка системы сбора, обработки, анализа, идентификации корреляции событий информационной инфраструктуры предприятия //Системы анализа и обработки данных. – 2023. – №. 1 (89). – С. 101-113.
17. Басыня Е. А. Распределенная система сбора, обработки и анализа событий информационной безопасности сетевой инфраструктуры предприятия / Безопасность информационных технологий. 2018. Т. 25. № 4. С. 43-52.

© И. В. Чугунков, М. М. Ровнягин, К. А. Федоров, К. В. Сысоев, 2026

Д. Р. Чуркин^{1✉}, Е. Ю. Воронкин¹

Разработка веб-приложения «Аптечка+» для учета домашних лекарственных средств и контроля срока их годности

¹Сибирский государственный университет геосистем и технологий,
г. Новосибирск, Российская Федерация
e-mail: denis.churkin.afikei@mail.ru

Аннотация. В статье представлены результаты разработки веб-приложения «Аптечка+», предназначенного для учета домашних лекарственных средств и контроля срока их годности. Рассмотрена проблема бессистемного хранения информации о лекарствах в домашних условиях, при котором пользователь может не знать фактическое наличие препаратов, их количество и дату окончания срока годности. Выполнен анализ предметной области и существующих решений для ведения домашней аптечки и напоминаний о приеме лекарственных средств. Для реализации приложения была выбрана клиент-серверная архитектура. Клиентская часть разработана с использованием Vue 3, TypeScript, Vue Router, Pinia и Tailwind CSS. Серверная часть реализована на Node.js с применением Express. Для хранения данных использована база данных SQLite. В результате работы было создано MVP веб-приложения, включающее лендинг, регистрацию, авторизацию, защищенную часть системы и раздел управления лекарственными средствами. Приложение позволяет добавлять, редактировать, удалять и просматривать препараты, а также хранить сведения о дозировке, количестве, схеме приема и сроке годности.

Ключевые слова: веб-приложение, домашняя аптечка, учет лекарств, срок годности, Vue 3, Node.js

D. R. Churkin^{1✉}, E. Yu. Voronkin¹

Development of a web application for home medication accounting

¹Siberian State University of Geosystems and Technologies, Novosibirsk, Russian Federation
e-mail: denis.churkin.afikei@mail.ru

Abstract. The article presents the results of developing the “Aptechka+” web application designed for accounting home medicines and controlling their expiration dates. The problem of unsystematic storage of information about medicines at home is considered. In such conditions, the user may not know the actual availability of medicines, their quantity, and expiration dates. The subject area and existing solutions for maintaining a home first-aid kit and medication reminders are analyzed. A client-server architecture was chosen for the application implementation. The client side was developed using Vue 3, TypeScript, Vue Router, Pinia, and Tailwind CSS. The server side was implemented using Node.js and Express. SQLite was used for data storage. As a result, an MVP version of the web application was created, including a landing page, registration, authorization, a protected user area, and a medication management section. The application allows users to add, edit, delete, and view medicines, as well as store information about dosage, quantity, intake schedule, and expiration date.

Keywords: web application, home first-aid kit, medication accounting, expiration date, Vue 3, Node.js

Введение

В настоящее время цифровые технологии активно применяются в различных сферах повседневной жизни человека. Одним из направлений их использо-

вания является автоматизация хранения и обработки персональной информации. К такой информации можно отнести сведения о лекарственных средствах, находящихся в домашней аптечке.

Большинство пользователей хранит дома лекарственные препараты для оказания первой помощи, лечения сезонных заболеваний или регулярного применения по назначению врача. При этом учет лекарственных средств часто ведется бессистемно либо не ведется вовсе. Информация о наличии препаратов, их количестве, сроке годности и схеме приема может храниться в бумажном виде, в заметках или запоминаться пользователем самостоятельно.

Такой подход может приводить к различным проблемам. Пользователь может приобрести лекарственное средство повторно, не зная, что оно уже есть дома. Также возможна ситуация, при которой препарат находится в аптечке, но его срок годности уже истек. Кроме того, при большом количестве лекарств затрудняется быстрый поиск нужного препарата и контроль его остатка.

На рынке существуют решения для учета медикаментов и напоминаний о приеме препаратов. Часть таких приложений ориентирована на мобильные устройства, часть предназначена для контроля приема лекарств, а отдельные сервисы содержат избыточный функционал, который не всегда требуется обычному пользователю при ведении домашней аптечки [1–4]. В связи с этим актуальной задачей является разработка простого веб-приложения, позволяющего вести список домашних лекарственных средств и контролировать основные сведения о них.

Целью работы является разработка веб-приложения для учета домашних лекарственных средств и контроля срока их годности. Для достижения цели были поставлены следующие задачи: выполнить анализ предметной области, определить функциональные требования, выбрать технологии разработки, спроектировать структуру базы данных, реализовать клиентскую и серверную части приложения, а также провести тестирование основных пользовательских сценариев.

Исследование предметной области

Домашняя аптечка представляет собой набор лекарственных средств, которые используются пользователем для оказания первой помощи, лечения сезонных заболеваний, выполнения врачебных назначений или хранения препаратов постоянного применения. По мере увеличения количества лекарств возрастает сложность контроля их наличия, количества, срока годности и схемы приема.

Основными данными, которые необходимо учитывать при ведении домашней аптечки, являются наименование лекарственного средства, форма выпуска, дозировка, срок годности, количество препарата и схема приема. Дополнительно может потребоваться хранение признака напоминаний о приеме, чтобы в дальнейшем использовать эти данные для уведомлений пользователя.

При отсутствии систематизированного учета пользователь сталкивается с рядом затруднений. К ним относится невозможность быстро определить наличие нужного препарата, риск хранения и использования просроченных лекарственных средств, а также сложность контроля остатков. Кроме того, при регулярном

приеме медикаментов возникает необходимость напоминаний, позволяющих снизить вероятность пропуска приема.

В ходе анализа существующих решений были рассмотрены приложения, предназначенные для учета домашней аптечки, контроля сроков годности и напоминаний о приеме лекарств. Приложение «Домашняя аптечка» ориентировано на контроль сроков годности лекарств, составление графика приема и получение напоминаний [1]. Приложение «Аптечка. Учет медикаментов» позволяет создавать аптечки, добавлять препараты, искать нужные лекарства и получать уведомления [2]. Отдельную группу составляют решения MyTherapy и Medisafe, которые в большей степени ориентированы на напоминания о приеме лекарств и отслеживание назначений [3, 4].

Проведенный анализ показал, что существующие решения закрывают отдельные задачи учета лекарственных средств. Одни приложения ориентированы на ведение списка аптечки, другие – на напоминания о приеме препаратов, третьи – на предоставление справочной информации. При этом часть решений доступна преимущественно в мобильном формате или содержит функции, которые не всегда требуются обычному пользователю для учета домашней аптечки.

На основании анализа предметной области были определены основные требования к разрабатываемому веб-приложению. Система должна обеспечивать регистрацию и авторизацию пользователей, просмотр списка лекарственных средств, добавление, редактирование и удаление препаратов, поиск и фильтрацию записей, хранение информации о сроке годности и количестве, а также защиту данных пользователя.

Методы и технологии

При разработке веб-приложения была выбрана клиент-серверная архитектура. Такой подход позволяет разделить пользовательский интерфейс, серверную логику и хранение данных. Клиентская часть отвечает за отображение интерфейса и обработку действий пользователя, серверная часть – за обработку запросов, авторизацию и взаимодействие с базой данных.

Для реализации клиентской части был выбран фреймворк Vue 3 [5]. Он позволяет разрабатывать интерфейс на основе компонентного подхода. Такой подход упрощает сопровождение приложения, так как отдельные элементы интерфейса могут быть выделены в самостоятельные компоненты. В качестве языка программирования был использован TypeScript [6], который обеспечивает статическую типизацию и позволяет снизить вероятность ошибок на этапе разработки.

Для сборки клиентского приложения применялся Vite [7]. Маршрутизация между страницами реализована с помощью Vue Router [8]. Для управления состоянием приложения использовалась библиотека Pinia [9]. Она применялась для хранения данных авторизации, информации о пользователе и списка лекарственных средств. Для оформления интерфейса использовался Tailwind CSS [10], а для выполнения HTTP-запросов между клиентской и серверной частями – Axios [11].

Серверная часть приложения была реализована на платформе Node.js [12] с использованием фреймворка Express [13]. Express применялся для создания

REST API, описания маршрутов и обработки HTTP-запросов. Для проверки входных данных использовалась библиотека Zod [14]. Она позволила описать схемы данных для регистрации, авторизации и операций с лекарственными средствами.

Для авторизации пользователя был использован механизм JSON Web Token [15]. После успешного входа сервер формирует токен, который используется при последующих запросах к защищенным маршрутам. Для хранения данных была выбрана SQLite [16, 17]. Данная система управления базами данных подходит для MVP-версии приложения, так как не требует отдельного серверного процесса и позволяет хранить данные в одном файле.

В базе данных были предусмотрены две основные таблицы: `users` и `medications`. Таблица `users` предназначена для хранения учетных записей пользователей. В ней сохраняются идентификатор пользователя, имя, адрес электронной почты, хеш пароля, дата создания и дата обновления записи. Таблица `medications` предназначена для хранения информации о лекарственных средствах. В ней сохраняются идентификатор лекарства, идентификатор пользователя, название препарата, форма выпуска, дозировка, схема приема, количество, начальное количество, срок годности, признак включения напоминаний, дата создания и дата обновления записи.

Между пользователем и лекарственными средствами установлена связь «один ко многим». Один пользователь может иметь несколько записей о лекарствах, но каждая запись относится только к одному пользователю. Это позволяет формировать индивидуальный список препаратов для каждой учетной записи и ограничивать доступ к данным других пользователей.

Проект был организован в формате монорепоzitория. В рамках одной структуры проекта были размещены клиентская и серверная части приложения. Такой подход упрощает запуск, сборку и сопровождение проекта.

Результаты

В результате работы было разработано MVP веб-приложения «Аптечка+», предназначенное для ведения домашней аптечки. В приложении реализованы публичная и закрытая части. Публичная часть включает лендинг, страницу регистрации и страницу авторизации. Лендинг предназначен для знакомства пользователя с возможностями сервиса и перехода к регистрации или входу в систему.

Закрытая часть приложения доступна только авторизованному пользователю. При попытке перейти в закрытый раздел без авторизации пользователь перенаправляется на страницу входа. Для этого на клиентской стороне используются навигационные ограничения Vue Router, а на серверной стороне – проверка авторизации при обращении к защищенным маршрутам.

Основным разделом закрытой части является страница «Мои лекарства». В данном разделе пользователь может просматривать список добавленных препаратов, выполнять поиск по наименованию, добавлять новые лекарственные средства, редактировать существующие записи и удалять ненужные препараты.

Пример пользовательского интерфейса раздела «Мои лекарства» представлен на рис. 1.

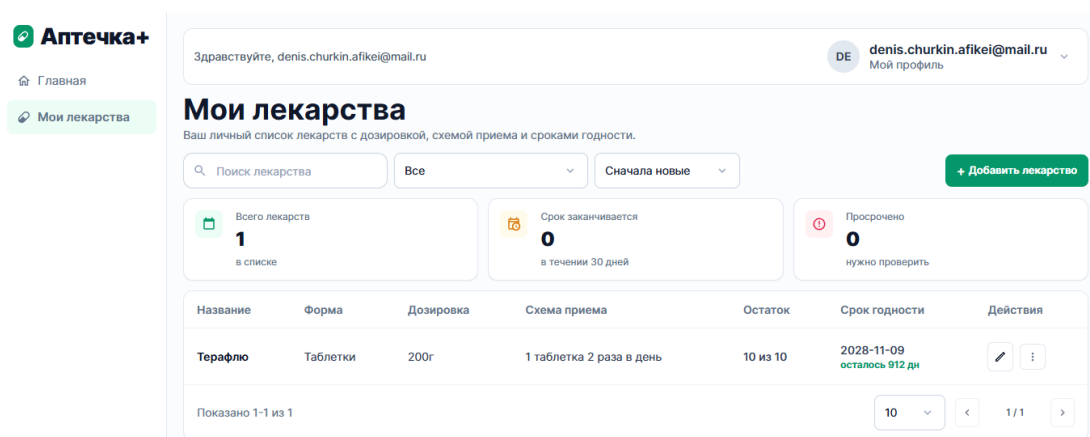


Рис. 1. Интерфейс раздела «Мои лекарства» веб-приложения «Аптечка+»

Для отображения лекарственных средств используется табличное представление. В таблице выводятся основные сведения о препаратах: наименование, форма выпуска, дозировка, схема приема, количество, срок годности и состояние признака напоминаний. Табличное представление позволяет компактно отображать данные и быстро находить необходимую информацию.

Добавление лекарственного средства реализовано через модальное окно с формой. Пользователь указывает название препарата, форму выпуска, дозировку, схему приема, количество, срок годности и признак включения напоминаний. После сохранения данные отправляются на серверную часть приложения и записываются в базу данных.

Редактирование лекарственного средства выполняется с использованием той же формы. При открытии формы поля заполняются текущими данными выбранного препарата. После изменения данных пользователь сохраняет запись, а обновленная информация отображается в списке лекарственных средств. Удаление препарата выполняется из таблицы. После подтверждения действия запись удаляется из базы данных и больше не отображается в интерфейсе.

Серверная часть предоставляет маршруты для регистрации, авторизации, получения текущего пользователя, выхода из учетной записи и управления лекарственными средствами. Основные маршруты API включают /auth/register, /auth/login, /auth/me, /auth/logout, а также маршруты с префиксом /medications, предназначенные для создания, получения, редактирования и удаления записей.

После реализации приложения было проведено тестирование основных пользовательских сценариев. Проверялись регистрация пользователя, авторизация, выход из учетной записи, защита закрытых маршрутов, добавление лекарственного средства, редактирование записи, удаление лекарства, поиск и отображение срока годности. Также проверялась корректность взаимодействия клиентской части с серверным API и сохранение данных в базе данных.

По результатам тестирования критических дефектов выявлено не было. Основные пользовательские сценарии выполнялись корректно, данные сохранялись в базе данных, а доступ к закрытым разделам ограничивался механизмом авторизации.

Заключение

В ходе работы было разработано веб-приложение «Аптечка+» для учета домашних лекарственных средств и контроля срока их годности. В процессе разработки был выполнен анализ предметной области, рассмотрены существующие решения, сформированы функциональные требования, спроектирована база данных и реализованы клиентская и серверная части приложения.

Разработанное приложение позволяет пользователю зарегистрироваться, авторизоваться, просматривать список лекарственных средств, добавлять новые препараты, редактировать и удалять записи, а также хранить сведения о количестве, дозировке, схеме приема и сроке годности. Реализованный функционал соответствует задачам MVP-версии и может быть использован как основа для дальнейшего развития системы.

Практическая значимость работы заключается в том, что приложение может применяться для организации домашней аптечки и упрощения контроля лекарственных средств. Использование такого сервиса позволяет снизить риск хранения просроченных препаратов и повысить удобство поиска информации о лекарствах.

В дальнейшем приложение может быть расширено за счет полноценной системы уведомлений, категорий лекарственных средств, истории приема, адаптивной мобильной версии, синхронизации данных между устройствами и перехода на более производительную серверную базу данных.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Домашняя аптечка : приложение. — URL: <https://www.rustore.ru/catalog/app/ru.application.homemedkit> (дата обращения: 03.05.2026). — Текст: электронный.
2. Аптечка. Учет медикаментов : приложение. — URL: <https://apps.apple.com/by/app/аптечка-учет-медикаментов/id1565373036> (дата обращения: 03.05.2026). — Текст: электронный.
3. MyTherapy : официальный сайт. — URL: <https://www.mytherapyapp.com> (дата обращения: 03.05.2026). — Текст : электронный.
4. Medisafe : официальный сайт. — URL: <https://www.medisafe.com> (дата обращения: 03.05.2026). — Текст : электронный.
5. Vue.js. Introduction : официальная документация. — URL: <https://vuejs.org/guide/introduction> (дата обращения: 03.05.2026). — Текст : электронный.
6. TypeScript Documentation : официальная документация. — URL: <https://www.typescriptlang.org/docs/> (дата обращения: 03.05.2026). — Текст : электронный.
7. Vite. Getting Started : официальная документация. — URL: <https://vite.dev/guide/>
8. (дата обращения: 03.05.2026). — Текст : электронный.
9. Vue Router. The official Router for Vue.js : официальная документация. — URL: <https://router.vuejs.org/> (дата обращения: 03.05.2026). — Текст : электронный.

10. Pinia. The intuitive store for Vue.js : официальная документация. – URL: <https://pinia.vuejs.org/> (дата обращения: 03.05.2026). – Текст : электронный.
11. Tailwind CSS Documentation : официальная документация. – URL: <https://tailwindcss.com/docs> (дата обращения: 03.05.2026). – Текст : электронный.
12. Axios. Getting Started : официальная документация. – URL: <https://axios-http.com/docs/intro> (дата обращения: 03.05.2026). – Текст : электронный.
13. Node.js Documentation : официальная документация. – URL: <https://nodejs.org/docs/latest/api/> (дата обращения: 03.05.2026). – Текст : электронный.
14. Express. Node.js web application framework : официальный сайт. – URL: <https://expressjs.com/> (дата обращения: 03.05.2026). – Текст : электронный.
15. Zod Documentation : официальная документация. – URL: <https://zod.dev/> (дата обращения: 03.05.2026). – Текст : электронный.
16. JSON Web Tokens. Introduction to JSON Web Tokens : официальный сайт. – URL: <https://jwt.io/introduction> (дата обращения: 03.05.2026). – Текст : электронный.
17. SQLite Documentation : официальная документация. – URL: <https://sqlite.org/docs.html> (дата обращения: 03.05.2026). – Текст : электронный.
18. SQLite. About SQLite : официальный сайт. – URL: <https://sqlite.org/about.html> (дата обращения: 03.05.2026). – Текст : электронный.
19. Флэнаган, Д. JavaScript. Подробное руководство. 7-е изд. – Санкт-Петербург : Питер, 2021. – 720 с.
20. Бэнкс, А. Изучаем JavaScript. Современные возможности языка / А. Бэнкс, Е. Порселло. – Санкт-Петербург : Питер, 2021. – 320 с.
21. Хэнчетт, Э. Vue.js в действии / Э. Хэнчетт, Б. Листуон. – Москва : ДМК Пресс, 2019. – 304 с.

© Д. Р. Чуркин, Е. Ю. Воронкин, 2026

Е. Ю. Шаранов^{1✉}, С. Ю. Кацко¹

Разработка игрового приложения с системой событий и управления ресурсами на платформе Godot

¹Сибирский государственный университет геосистем и технологий,
г. Новосибирск, Российская Федерация
e-mail: egor.sharapov90@mail.ru

Аннотация. В статье представлены основные результаты разработки игрового приложения в жанре стратегического менеджера отряда наемников. Выполнен анализ существующих игр-аналогов, выявлена недостаточная проработка роли «тылового» управляющего, принимающего решения без прямого участия в сражениях. Приложение реализовано на движке Godot 4 с использованием языка GDScript, пользовательского интерфейса на основе Control Nodes и формата JSON для хранения данных. Реализованы ключевые механики: подбор контрактов и наемников, система случайных событий с многовариантными решениями, динамический расчет исхода заданий, управление несколькими видами ресурсов, магазин снаряжения, а также сохранение и загрузка состояния. Архитектура приложения построена на data-driven принципах, что позволяет расширять контент без изменения программного кода.

Ключевые слова: игровое приложение, менеджер наемников, Godot Engine, GDScript, система событий, управление ресурсами, data-driven

E. Y. Sharapov^{1✉}, S. Yu. Katsko¹

Development of a game application with an event and resource management system on the Godot platform

¹Siberian State University of Geosystems and Technologies, Novosibirsk, Russian Federation
e-mail: egor.sharapov90@mail.ru

Abstract. The article presents the main results of the development of a game application in the genre of strategic mercenary manager. The analysis of existing analogue games is carried out, revealing an insufficient elaboration of the role of a "back-office" manager who makes decisions without direct participation in battles. The application is implemented on the Godot 4 engine using the GDScript language, a Control Nodes-based user interface and the JSON format for data storage. Key mechanics are implemented: selection of contracts and mercenaries, a system of random events with multi-choice decisions, dynamic calculation of mission outcomes, management of several types of resources, an equipment shop, as well as saving and loading the game state. The architecture is built on data-driven principles, allowing content to be expanded without changing the program code.

Keywords: game application, mercenary manager, Godot Engine, GDScript, event system, resource management

Введение

Современный рынок инди-игр насыщен тактическими менеджерами, в которых игрок управляет отрядом, экипирует бойцов и руководит ими в пошаговых сражениях. Однако за кадром остается фигура управляющего – того, кто находит контракты, распределяет людей по задачам и принимает стратегические реше-

ния, не спускаясь на поле боя. Игры, полностью сосредоточенные на этой «кабинетной» роли, практически отсутствуют, хотя спрос на неторопливый, аналитический геймплей, не требующий быстрой реакции и тактического микроменеджмента, стабильно растет.

Целью данной работы являлась разработка игрового приложения, моделирующего деятельность управляющего гильдией наемников в фэнтезийном мире. Приложение должно было предоставить игроку возможность выбирать контракты, назначать исполнителей, реагировать на случайные события и отслеживать изменения множества взаимосвязанных ресурсов, исключив при этом фазу прямого управления боем.

Исследование предметной области

Анализ рынка показал, что наиболее близкими к сформулированной концепции являются пять проектов, каждый из которых, однако, не реализует полноценную модель «чистого» управления.

Battle Brothers [1] – тактическая RPG с глубокой системой контрактов и найма, но игрок полностью контролирует каждый шаг отряда в сражении. Darkest Dungeon [2] делает акцент на психологическом состоянии героев, но также включает обязательные пошаговые бои. Our Adventurer Guild [3] комбинирует управление гильдией с тактическими боями. Gladiator Guild Manager [4] автоматизирует схватки, однако оставляет за игроком тактическую расстановку и вмешательство в ход боя. Blades for Hire [5] предлагает богатую событийную систему, но сохраняет элементы непрямого управления боем.

Все перечисленные игры, несмотря на наличие менеджерских механик, не позволяют игроку полностью устраниваться от сражений и сосредоточиться исключительно на стратегических решениях: выборе задания, назначении подходящего бойца и реакции на непредвиденные обстоятельства. Именно этот пробел послужил основанием для формулировки требований к разрабатываемому приложению.

Методы и технологии

В качестве среды разработки выбран игровой движок Godot Engine версии 4 [6]. Он предоставляет встроенную систему сцен и узлов, сигналы для организации взаимодействия, а также поддерживает кроссплатформенный экспорт. Языком программирования является GDScript – динамический, интегрированный в движок язык с простым синтаксисом, ориентированный на быструю разработку игровой логики.

Пользовательский интерфейс полностью построен на узлах Control (VBoxContainer, HBoxContainer, Button, Label, TabContainer и др.). Такой подход обеспечивает адаптивность под различные разрешения экрана и легкость модификации интерфейса без использования сторонних библиотек.

Все игровые данные – контракты, наемники, события, предметы – хранятся в виде структурированных массивов словарей (формат JSON). Для сохранения прогресса применяется механизм сериализации глобального состояния приложения в

JSON-файл стандартными средствами Godot (класс FileAccess). Это решение позволило отделить контент от кода (data-driven архитектура) и упростить расширение игры: добавление нового контракта или предмета сводится к редактированию соответствующего JSON-шаблона без изменения программных модулей.

Результаты

В результате работы создано игровое приложение «Менеджер наемников», в котором реализован ряд взаимосвязанных функций. Из JSON-шаблонов автоматически генерируется список доступных контрактов с детализацией по награде, сложности, риску, заказчику и связанной фракции. Пользователь может просматривать доступных наемников, каждый из которых обладает индивидуальными характеристиками (бой, ловкость, лояльность, здоровье, стресс, усталость), и назначать выбранного бойца на принятый контракт.

При завершении дня активируется система случайных событий: из подготовленного пула выбирается событие с тремя вариантами действий, каждый из которых влияет на шанс успеха, уровень стресса наемника и количество золота.

Исход контракта рассчитывается динамически по формуле, учитывающей характеристики назначенного бойца, риск задания и выбранный игроком модификатор (рис. 1). Приложение отслеживает несколько видов ресурсов: золото, репутацию, отношения с тремя фракциями, а также индивидуальные показатели здоровья, стресса и усталости каждого наемника (рис. 2).

Реализован магазин снаряжения, позволяющий приобретать предметы за золото и применять их к бойцам, изменяя их параметры. Предусмотрена система сохранения и загрузки состояния (золото, день, ростера, активные контракты, инвентарь) в JSON-файл.

Алгоритм разрешения контракта работает следующим образом. Базовый шанс успеха (50 %) модифицируется в зависимости от параметров назначенного наемника: если его боевой навык превышает числовую сложность контракта, добавляется бонус 15 %; ловкость не ниже 12 дает дополнительный бонус 5 %; высокий уровень стресса (≥ 70) накладывает штраф 20 %. Риск контракта снижает вероятность на 4 % за каждый пункт. Выбранное в событии действие добавляет итоговый модификатор (combat_mod).

Итоговая вероятность ограничивается диапазоном 10–95 %, после чего случайное число сравнивается с ней и определяется успех или провал. При успехе игрок получает золото и репутацию, при провале – штрафы и ранение бойца. Все изменения немедленно отражаются в интерфейсе (рис. 3).

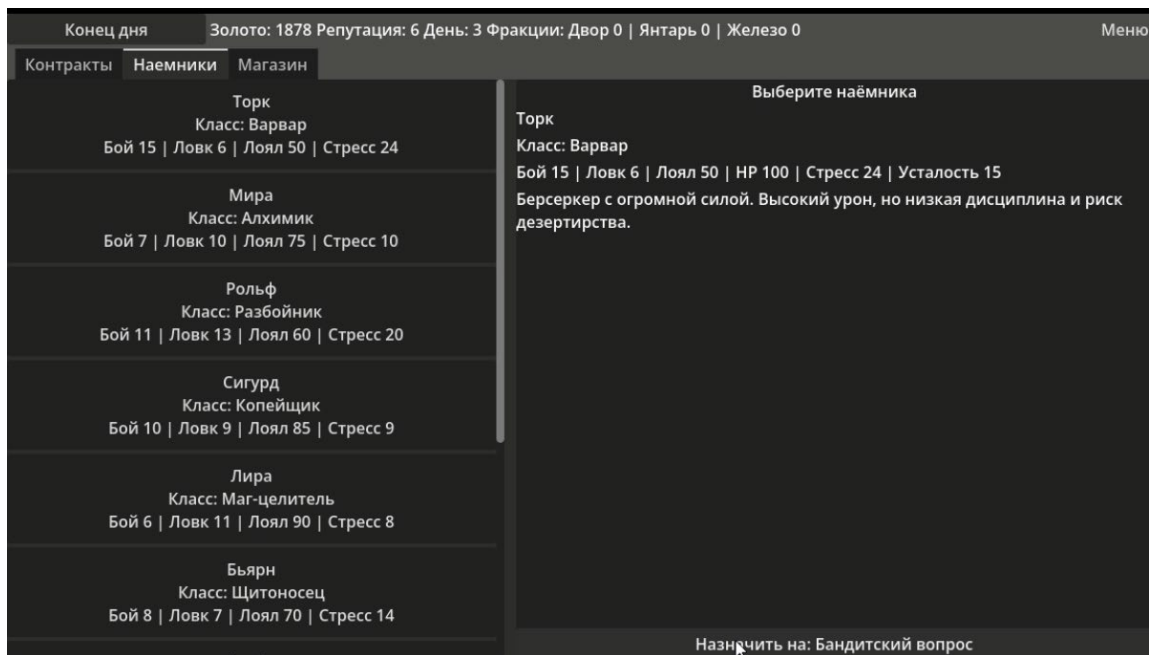


Рис. 1. Интерфейс игрового приложения

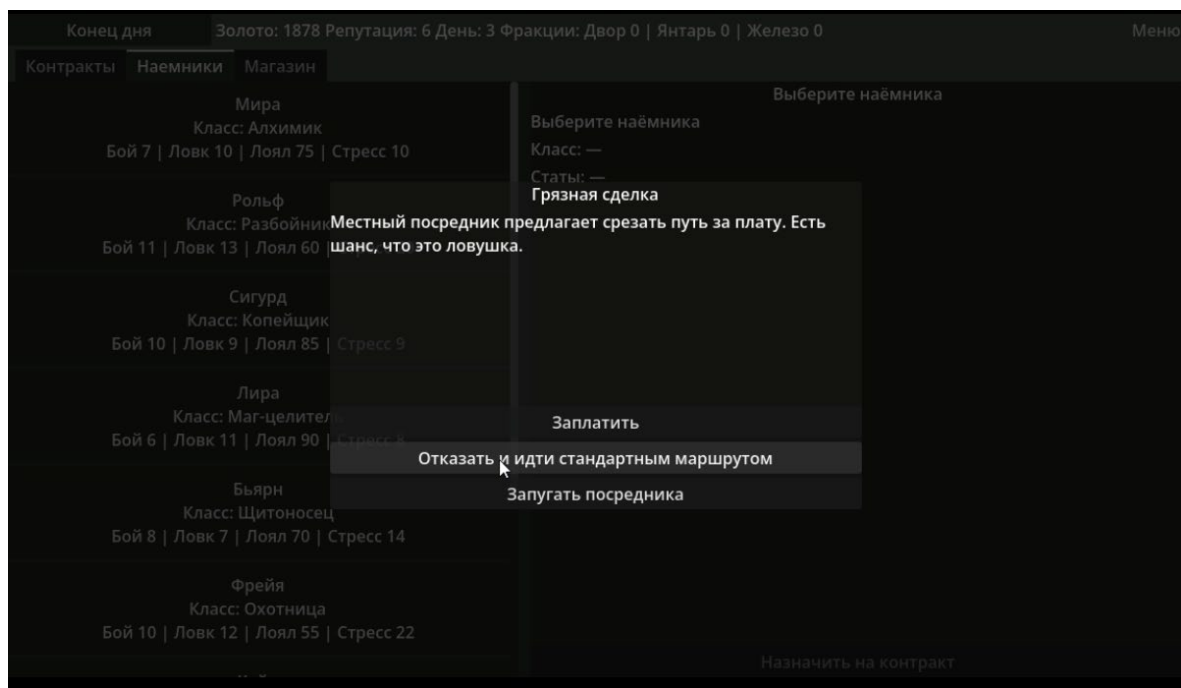


Рис. 2. Случайное событие

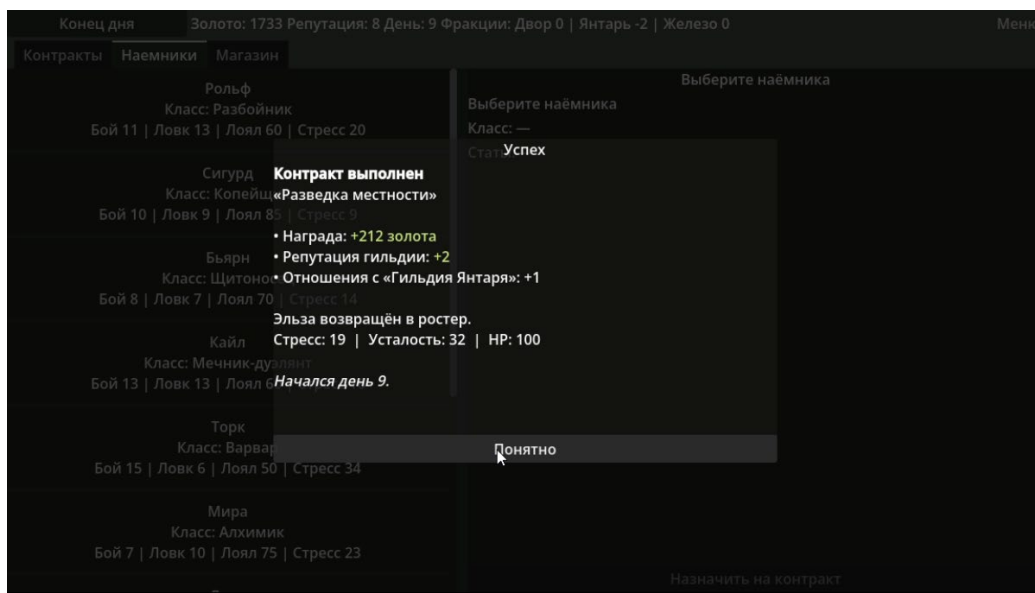


Рис. 3. Завершение задания

Заключение

Разработанное приложение заполняет нишу «кабинетного» управляющего, позволяя игроку сосредоточиться на стратегическом планировании и принятии решений без необходимости участвовать в тактических сражениях. Использование Godot 4 и GDScript обеспечило компактный, легковесный продукт, а data-driven подход на основе JSON-шаблонов дает возможность неограниченно наращивать контент и быстро проверять игровые гипотезы. Текущая версия функционирует как минимально жизнеспособный продукт (MVP), прошедший тестирование и подтвердивший интерес пользователей. Дальнейшее развитие предполагает расширение пула событий и контрактов, добавление графических ассетов и углубление экономической модели без перестройки архитектуры.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Battle Brothers. URL: https://store.steampowered.com/app/365360/Battle_Brothers/ (дата обращения: 15.05.2026).
2. Darkest Dungeon. URL: <https://www.darkestdungeon.com> (дата обращения: 15.05.2026).
3. Our Adventurer Guild. URL: https://store.steampowered.com/app/1764260/Our_Adventurer_Guild/ (дата обращения: 15.05.2026).
4. Gladiator Guild Manager. URL: https://store.steampowered.com/app/1041260/Gladiator_Guild_Manager/ (дата обращения: 15.05.2026).
5. Blades for Hire. URL: <https://play.google.com/store/apps/details?id=com.bladesforhire> (дата обращения: 15.05.2026).
6. Godot Engine Documentation. URL: <https://docs.godotengine.org/en/stable/> (дата обращения: 15.05.2026).
7. GDScript Reference. URL: https://docs.godotengine.org/en/stable/tutorials/scripting/gdscript/gdscript_basics.html (дата обращения: 15.05.2026).
8. JSON in Godot. URL: https://docs.godotengine.org/en/stable/classes/class_json.html (дата обращения: 15.05.2026).

© Е. Ю. Шапанов, С. Ю. Кацко, 2026

Т. С. Шевцова¹✉, Н. С. Головачев¹

Разработка программного средства анализа и проектирования озеленения жилых территорий на основе спутниковых данных и моделей застройки

¹Сибирский государственный университет геосистем и технологий,
г. Новосибирск, Российская Федерация
e-mail: shevchova004@gmail.com

Аннотация. В статье представлены основные результаты разработки программного средства для автоматизированного анализа и проектирования озеленения жилых территорий. Для реализации были использованы язык Python с библиотеками геоанализа (OpenCV, NumPy, Rasterio, GeoPandas, Shapely) и среда Unreal Engine 5 для визуализации. Программное средство состоит из двух аналитических модулей и модуля визуализации. Первый модуль по спутниковому снимку автоматически выделяет существующие деревья и формирует таблицу их координат и параметров. Второй модуль с учётом нормативных отступов, зон ограничений и инсоляции генерирует рекомендации по размещению новых деревьев. Третий модуль в Unreal Engine загружает полученные таблицы и отображает существующие и проектируемые деревья на трёхмерной сцене застройки. Приведены результаты тестирования, подтверждающие работоспособность и производительность разработанного решения.

Ключевые слова: озеленение жилых территорий, спутниковые данные, NDVI, Unreal Engine, Python, геоинформационные системы, визуализация, генерация рекомендаций

T. S. Shevtsova¹✉, N. S. Golovachev¹

Development of a software tool for analysis and design of residential area landscaping based on satellite data and building models

¹Siberian State University of Geosystems and Technologies, Novosibirsk, Russian Federation
e-mail: shevchova004@gmail.com

Abstract. The article presents the main results of the development of a software tool for the automated analysis and design of landscaping in residential areas. Python with geospatial analysis libraries (OpenCV, NumPy, Rasterio, GeoPandas, and Shapely) and Unreal Engine 5 for visualization were used for the implementation. The software consists of two analytical modules and a visualization module. The first module automatically detects existing trees from a satellite image and generates a table containing their coordinates and parameters. The second module generates recommendations for the placement of new trees, taking into account regulatory setback distances, restricted zones, and insolation conditions. The third module, implemented in Unreal Engine, imports the generated tables and visualizes existing and proposed trees within a three-dimensional urban development scene. Testing results confirming the functionality and performance of the developed solution are presented.

Keywords: residential landscaping, satellite data, NDVI, Unreal Engine, Python, geographic information systems, visualization, recommendation generation

Введение

Проблема системного озеленения жилых территорий становится всё более актуальной в условиях урбанизации. Зелёные насаждения улучшают микроклимат, снижают шум, очищают воздух и повышают комфорт проживания. Однако существующие подходы к проектированию озеленения часто либо трудоёмки (ручной обход территории, измерение каждого дерева), либо не обеспечивают наглядности (плоские схемы, таблицы). С развитием спутниковых технологий и систем трёхмерной визуализации появилась возможность автоматизировать процесс инвентаризации существующих деревьев и оптимизировать размещение новых, соблюдая нормативные требования и учитывая экологические факторы.

Цель работы – разработка программного средства, которое по спутниковым данным и модели застройки автоматически формирует таблицу существующих деревьев, генерирует рекомендации по посадке новых с учётом ограничений и предоставляет интерактивную 3D-визуализацию в Unreal Engine [1].

Исследование предметной области

Существующие методы проектирования озеленения можно разделить на три группы:

Традиционное ручное проектирование – выезд специалиста на местность, фотофиксация, нанесение деревьев на план. Недостатки: высокая трудоёмкость, субъективность, невозможность точного учёта инсоляции и затенения от зданий.

ГИС-системы (QGIS, ArcGIS) – позволяют обрабатывать спутниковые данные, вычислять вегетационные индексы, строить карты. Однако они не автоматизируют генерацию рекомендаций по посадке и не предоставляют встроенных средств фотореалистичной трёхмерной визуализации [2, 3].

Архитектурные визуализаторы (Twinmotion, Lumion) – дают качественную 3D-картинку, но растения расставляет пользователь вручную, без опоры на геоаналитику и нормативные ограничения.

Таким образом, существует пробел между аналитическими инструментами и средствами визуализации. Предлагаемое решение заполняет его, объединяя автоматическую обработку спутниковых снимков, генерацию рекомендаций по правилам и визуализацию в современном игровом движке.

Методы и технологии

Для реализации программного средства был выбран следующий стек технологий. Основным языком разработки аналитических модулей является Python [4]. Для обработки спутниковых снимков и выделения крон деревьев использовались библиотеки rasterio [5] (чтение GeoTIFF и геопривязка), numpy [6] (математические операции), opencv [7] (морфологическая фильтрация и поиск связанных компонент). Вычисление вегетационного индекса NDVI выполнялось по красному и ближнему инфракрасному каналам. Для работы с векторными данными (граница участка, зоны ограничений, существующие деревья) применялись библиотеки geopandas [8] и shapely [9] – они позволяют строить буферные зоны, выполнять операции вычитания полигонов и проверять вхождение точек.

Конфигурационные параметры (буферы, минимальные расстояния, шаг сетки) вынесены в файл формата YAML и загружаются через `ruyaml` [10]. Все таблицы сохраняются в формате CSV с использованием `pandas` [11].

Визуализация реализована в среде Unreal Engine 5.6. Выбор движка обусловлен фотореалистичным рендерингом, системой динамического освещения Lumen, а также встроенными механизмами уровней детализации (LOD), которые позволяют отображать тысячи деревьев без падения производительности. Загрузка CSV-файлов и расстановка 3D-моделей деревьев выполнены на языке визуального программирования Blueprints. Для каждой строки таблицы производятся проверка валидности данных, конвертация географических координат в мировые координаты сцены и создание актора (Spawn Actor) с соответствующей моделью. Модели деревьев (берёза, дуб, сосна, кустарник) созданы в Blender [12] и имеют по четыре уровня детализации: максимальная детализация на расстоянии до 30 метров, средняя – до 80 метров, низкая – до 200 метров, и спрайт (биллборд) – далее 200 метров. Подготовка геоданных (контуры зданий, дорог, зелёных зон) выполнена с помощью аддона BlenderGIS [12], который загружает информацию из открытого источника OpenStreetMap.

Для запуска аналитических модулей и вызова Unreal Engine разработан графический интерфейс на Tkinter [13], предоставляющий три кнопки: «Функция 1» – анализ существующих деревьев, «Функция 2» – генерация рекомендаций, «Функция 3» – открытие сцены в Unreal Engine. Все три модуля обмениваются данными через CSV-файлы, что обеспечивает их независимость и возможность модификации.

Обобщённая структура разработанного программного средства представлена на рис. 1. Схема отражает модульную организацию системы и последовательность обмена данными между аналитическими компонентами, пользовательским интерфейсом и средой визуализации Unreal Engine.

Работа программного средства начинается с запуска проекта пользователем и перехода в главное меню Unreal Engine. Из главного меню доступны три функциональных направления: анализ существующих древесных насаждений по спутниковым данным, генерация рекомендуемых точек размещения новых деревьев и открытие уровня с визуализацией результатов. Первый функциональный блок предназначен для загрузки исходного спутникового изображения и его последующей обработки в Python-модуле 1. В рамках данного этапа выполняются операции предварительной фильтрации изображения, выделения растительности, определения крон деревьев и формирования результирующей таблицы существующих деревьев. Таблица 1 содержит идентификатор объекта, координаты и тип или вид дерева, если такая классификация предусмотрена входными данными или результатами дешифрирования.

Второй функциональный блок реализует генерацию новых точек размещения древесных насаждений. Пользователь задаёт ограничения проектирования, включая минимальные отступы, пустые зоны, области инсоляции, запретные зоны и иные пространственные параметры. Указанные ограничения передаются в Python-модуль 2, который использует результаты анализа существующих дере-

вьев, данные о территории и заданные пользователем параметры. На выходе формируется таблица 2, содержащая рекомендуемые новые деревья с указанием идентификатора, координат, вида и ограничений, повлиявших на выбор каждой точки размещения.

Третий функциональный блок отвечает за визуальное представление результатов в 3D-сцене Unreal Engine. Модуль визуализации считывает данные из таблиц 1 и 2, выполняет преобразование координат в систему координат сцены и отображает соответствующие объекты в составе цифровой модели территории. В результате в единой 3D-сцене отображаются существующие деревья, рекомендуемые новые посадки, застройка и пространственные ограничения. Такая организация обеспечивает разделение задач между аналитическими модулями и модулем визуализации, упрощает отладку отдельных компонентов и позволяет независимо модифицировать алгоритмы обработки спутниковых данных, генерации рекомендаций и отображения объектов в Unreal Engine.

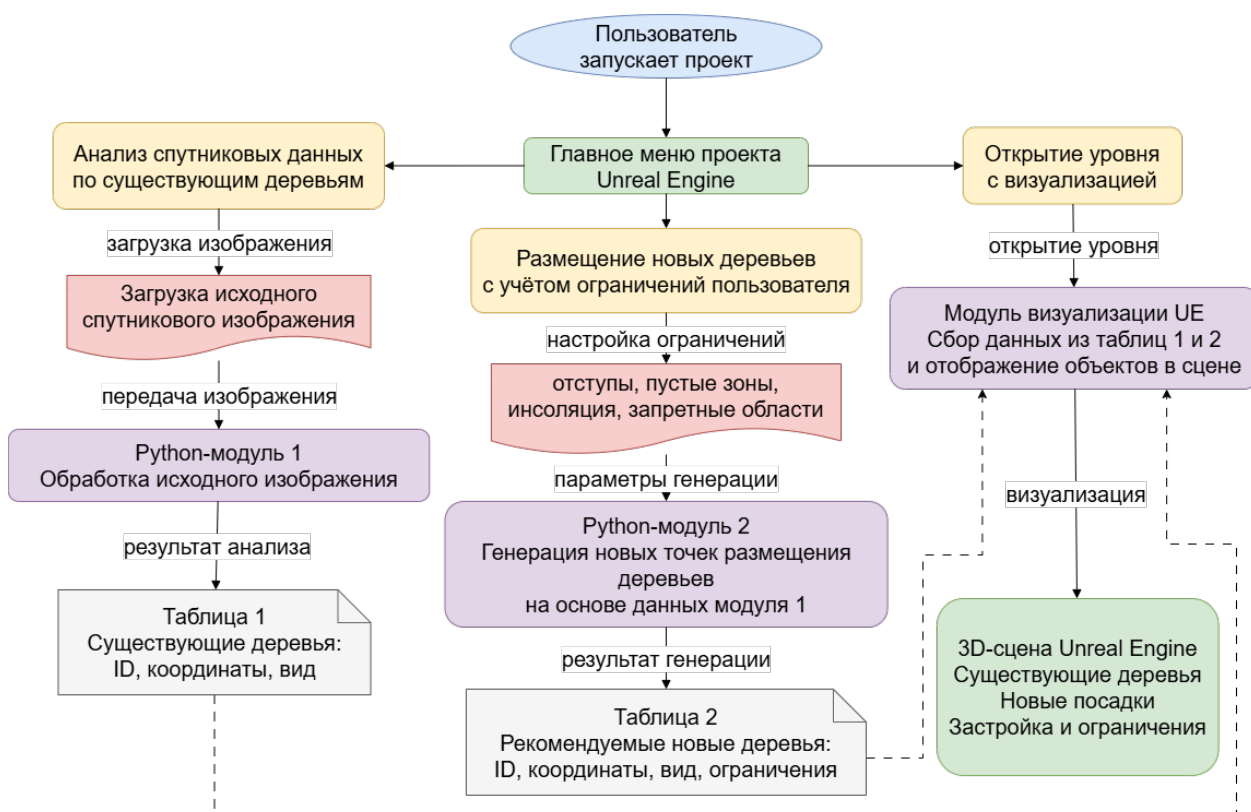


Рис. 1. Концептуальная схема функционирования системы

Результаты

В результате работы было создано программное средство, реализующее три основные функции. Первая функция – анализ спутниковых данных – по загруженному GeoTIFF вычисляет NDVI, выделяет кроны, формирует файл `existing_trees.csv` (рис. 2).

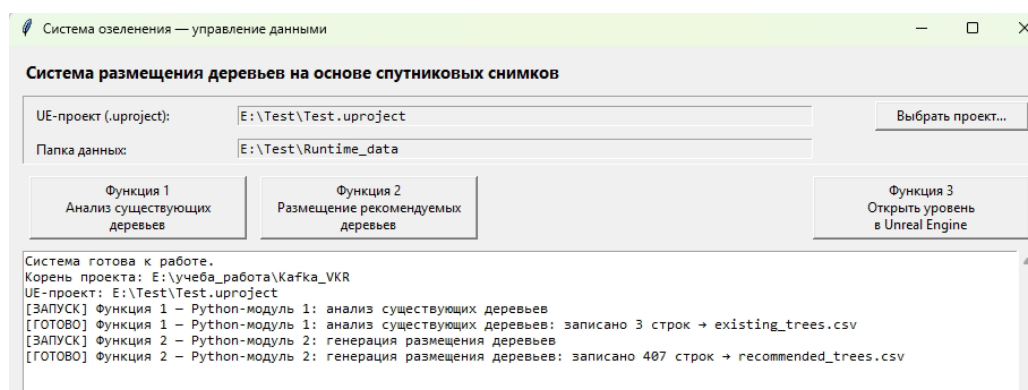


Рис. 2. Интерфейс мобильного приложения

На тестовом квартале площадью 0,5 км² обнаружено 187 существующих деревьев. Вторая функция – генерация рекомендаций – с учётом нормативов (отступ от зданий 5 м, от дорог 1 м, от существующих деревьев 4 м) и зон инсоляции сгенерировала 407 рекомендованных мест посадки. Все точки находятся внутри границы участка и не пересекают запретные зоны. Третья функция – визуализация в Unreal Engine – обеспечивает стабильную частоту кадров 58–72 FPS на конфигурации i5-12450H / RTX 3050 Ti / 16 ГБ при отображении 1000 деревьев; благодаря использованию LOD нагрузка на GPU снижена до 65 %.

Проведено тестирование: 10 модульных тестов (pytest), 3 интеграционных, 4 проверки валидации выходных данных – все 17 тестов пройдены. Выходные CSV-файлы полностью совместимы с Unreal Engine, координаты переведены в сантиметры (масштаб ×100). На данный момент все данные пользователя хранятся в памяти мобильного телефона. Использование таблиц для экспорта позволит пользователю упростить взаимодействие с лечащим врачом-эндокринологом, которому необходимо регулярно следить за состоянием пациента. Обучающие карточки позволят, в свою очередь, лучше разобраться как с функционалом мобильного приложения, так и, частично, с проблемными вопросами, с которым может столкнуться пользователь.

Заключение

Разработанное программное средство позволяет автоматизировать трудоёмкий процесс инвентаризации существующих деревьев и проектирования новых посадок. Оно объединяет возможности геоаналитики (NDVI, буферные зоны, сетка кандидатов) и фотореалистичной 3D-визуализации, доступной для неспециалистов. Решение может быть использовано ландшафтными архитекторами, студентами профильных направлений и муниципальными службами. В перспективе планируется добавить нейросетевое распознавание видов деревьев [14], интеграцию с BIM-моделями и поддержку виртуальной реальности.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Epic Games. Unreal Engine 5 Documentation. – URL: <https://docs.unrealengine.com> (дата обращения: 10.05.2026).

2. Коновалова, Н. В. ГИС в ландшафтном планировании / Н. В. Коновалова. – Новосибирск: СГУГиТ, 2019. – 180 с.
3. Jensen, J. R. Remote Sensing of the Environment: An Earth Resource Perspective / J. R. Jensen. – Pearson, 2016. – 656 p.
4. Python Software Foundation. Python Language Reference. – URL: <https://docs.python.org> (дата обращения: 10.05.2026).
5. Rasterio Documentation. – URL: <https://rasterio.readthedocs.io> (дата обращения: 10.05.2026).
6. NumPy Documentation. – URL: <https://numpy.org/doc> (дата обращения: 10.05.2026).
7. OpenCV. Image Processing. – URL: <https://opencv.org> (дата обращения: 10.05.2026).
8. GeoPandas Documentation. – URL: <https://geopandas.org> (дата обращения: 10.05.2026).
9. Shapely Documentation. – URL: <https://shapely.readthedocs.io> (дата обращения: 10.05.2026).
10. PyYAML Documentation. – URL: <https://pyyaml.org> (дата обращения: 10.05.2026).
11. Pandas Documentation. – URL: <https://pandas.pydata.org> (дата обращения: 10.05.2026).
12. Blender Foundation. BlenderGIS Add-on. – URL: <https://github.com/domlysz/BlenderGIS> (дата обращения: 10.05.2026).
13. Tkinter Documentation. Python standard library. – URL: <https://docs.python.org/3/library/tkinter.html> (дата обращения: 10.05.2026).
14. Рыбалкин, Д. А. Методы автоматизированного выделения древесной растительности по спутниковым снимкам / Д. А. Рыбалкин // Геопрофи. – 2024. – № 3. – С. 34–41.

© Т. С. Шевцова, Н. С. Головачев, 2026

В. В. Шнейдер^{1✉}, С. Ю. Кацко¹

Разработка рекомендательной информационной системы для поддержки абитуриентов при выборе специальности

¹Сибирский государственный университет геосистем и технологий,
г. Новосибирск, Российская Федерация
e-mail: Viktor.hs2004@mail.ru

Аннотация. В статье представлены основные результаты разработки рекомендательной информационной системы для поддержки абитуриентов при выборе образовательной программы. Выполнен анализ предметной области, в котором рассмотрены существующие сервисы для абитуриентов, а также выявлены их функциональные ограничения. Для создания системы были использованы фреймворк Spring Boot, язык программирования Java, клиентская часть на Vue.js и база данных PostgreSQL. Система предоставляет пользователю возможность заполнения профиля с баллами ЕГЭ и интересами, получения персонализированных рекомендаций с детализацией процента совпадения, интерактивного сравнения до пяти программ в одной таблице и просмотра карьерных перспектив с данными о зарплатах и работодателях через интеграцию с HeadHunter API.

Ключевые слова: рекомендательная система, абитуриент, образовательная программа, spring boot, vue.js, postgresql, headhunter API, JWT, swagger

V. V. Shneyder^{1✉}, S. Yu. Katsko¹

Development of a recommendation information system to support applicants in choosing a specialty

¹Siberian State University of Geosystems and Technologies, Novosibirsk, Russian Federation
e-mail: Viktor.hs2004@mail.ru

Abstract. The article presents the main results of the development of a recommendation information system to support applicants in choosing an educational program. The analysis of the subject area is performed, in which existing services for applicants are considered, as well as their functional limitations are identified. The Spring Boot framework, Java programming language, Vue.js client side and PostgreSQL database were used to create the system. The system provides the user with the ability to fill in a profile with Unified State Exam scores and interests, receive personalized recommendations with detailing of the match percentage, interactively compare up to five programs in one table and view career prospects with salary and employer data through integration with the HeadHunter API.

Keywords: recommendation system, applicant, educational program, spring boot, vue.js, postgresql, headhunter API, JWT, swagger

Введение

Выбор образовательной программы и высшего учебного заведения - один из ключевых этапов в жизни каждого абитуриента, определяющий его профессиональную траекторию на годы вперед. Современный рынок высшего образования в России представлен сотнями вузов и тысячами программ, различаю-

щихся по десяткам параметров: проходные баллы ЕГЭ, количество бюджетных мест, стоимость обучения, наличие общежития и военной кафедры, рейтинг учебного заведения. Данные об этих характеристиках распределены по множеству источников, что делает процесс их сбора и сопоставления крайне трудоемким [1]. В таких условиях абитуриенту все сложнее принять взвешенное решение, а риск ошибки при выборе направления подготовки существенно возрастает.

Для облегчения этой задачи были созданы специализированные онлайн-сервисы: Вузопедия, Поступи Онлайн, Табитуриент, государственный портал «Поступай правильно». Все они дают возможность искать программы по ЕГЭ, городу, стоимости, но на этом их возможности, по сути, заканчиваются [2–5]. Ни один из них не объясняет пользователю, почему ему предлагается именно эта программа, не дает удобной таблицы для сравнения выбранных вариантов между собой и не показывает, какие профессии ждут выпускника и какие зарплаты предлагают реальные компании. Таким образом, становится очевидной необходимость создания системы нового типа, объединяющей персонализированные рекомендации с прозрачной логикой, интерактивное сравнение программ и информацию о карьерных перспективах.

Исследование предметной области

Рынок сервисов для абитуриентов в России представлен несколькими крупными игроками. Вузопедия является одним из наиболее известных агрегаторов с карточным интерфейсом и базовой фильтрацией [2]. На платформе также присутствует профориентационный тест, однако его результаты не интегрированы в алгоритм подбора программ. Поступи Онлайн располагает крупнейшей базой вузов и предлагает калькулятор шансов, но не раскрывает принцип его работы [3]. Табитуриент представляет собой наиболее простой сервис с минимальным набором фильтров [4]. Государственный портал «Поступай правильно» использует официальные данные от вузов, но в плане персонализации не отличается от конкурентов [5].

Анализ показал, что все существующие сервисы работают как каталоги с фильтрами и не решают три ключевые задачи: не дают пользователю прозрачной детализации рекомендаций, не предлагают инструментов для многокритериального сравнения программ в едином интерфейсе и не предоставляют данных о карьерных перспективах после выпуска. Именно эти пробелы легли в основу требований к разрабатываемой системе.

Методы и технологии

Разработку рекомендательной системы было решено вести на базе фреймворка Spring Boot с использованием языка программирования Java. Spring Boot предоставляет встроенную поддержку множества компонентов, необходимых для реализации проекта: Spring Security для аутентификации, Spring Data JPA для работы с базой данных, Spring Web для создания REST API, а также встроенный планировщик задач [6].

К преимуществам Spring Boot можно отнести следующие моменты: модульная структура позволяет логически разделить код на контроллеры, сервисы и репозитории, что упрощает разработку и тестирование; обширная экосистема библиотек и плагинов расширяет функциональность; наличие подробной документации и активного сообщества снижает порог входа [6].

Для хранения данных была выбрана СУБД PostgreSQL, взаимодействие с которой осуществляется через Hibernate ORM. Hibernate позволяет описывать структуру данных в виде Java-классов с аннотациями, а создание таблиц и связей между ними возлагать на сам фреймворк, что ускоряет разработку и снижает вероятность ошибок при написании SQL-запросов вручную [7].

Клиентская часть реализована на фреймворке Vue.js 3 - прогрессивном JavaScript-фреймворке с удобной системой компонентов. Для оформления интерфейса использован Tailwind CSS, позволяющий быстро создавать адаптивный и визуально аккуратный интерфейс без написания кастомных стилей [8]. Взаимодействие с серверной частью осуществляется через библиотеку Axios.

Для защиты пользовательских данных реализована аутентификация с использованием JWT-токенов. Пароли хранятся в базе данных в хешированном виде с применением алгоритма BCrypt. Документирование API выполнено с помощью Swagger UI, который автоматически формирует интерактивную документацию на основе аннотаций в коде контроллеров.

Модуль карьерного трека интегрирован с HeadHunter API -открытым программным интерфейсом платформы hh.ru [9]. Синхронизация данных о зарплатах и работодателях реализована через механизм запланированных задач Spring Scheduled, что позволяет обновлять информацию автоматически без участия пользователя.

Таким образом, для разработки системы был использован следующий стек технологий: язык программирования Java 17 [6], фреймворк Spring Boot 3 [6], СУБД PostgreSQL с Hibernate ORM [7], клиентская часть на Vue.js 3 с Tailwind CSS [8], JWT-аутентификация с BCrypt, документирование через Swagger UI, интеграция с HeadHunter API через Spring Scheduled [9].

Результаты

В результате работы была создана рекомендательная информационная система, в которой реализованы следующие функции: регистрация и авторизация пользователей с защитой через JWT-токены; заполнение профиля абитуриента с указанием баллов ЕГЭ, профессиональных интересов, предпочитаемых городов, формы обучения и бюджетных ограничений; получение персонализированных рекомендаций с отображением процента совпадения и детализацией по компонентам – баллы ЕГЭ, интересы, город, бюджет, рейтинг вуза (рис. 1); интерактивное сравнение до пяти программ в одной таблице с многоуровневой сортировкой по клику на заголовки и визуальными маркерами лучших значений (рис. 2); просмотр карьерного трека для каждой специальности с отображением профессий, средних зарплат, компаний-работодателей и уровня востребованности (рис. 3).

Алгоритм рекомендаций работает следующим образом: для каждой образовательной программы рассчитывается итоговый процент совпадения на основе многокритериальной оценки. Оценка складывается из шести компонентов - экзамены, интересы, проходной балл, бюджетные места, город и рейтинг вуза, -каждый из которых имеет собственный весовой коэффициент. Результатом является ранжированный список программ с полной детализацией, что позволяет пользователю понять, из чего сложилась оценка и что можно улучшить.

Модуль карьерного трека интегрирован с HeadHunter API [9]. Сервис DataSyncService по расписанию выполняет запросы к внешнему API для каждой профессиональной роли, получает данные о вакансиях, вычисляет средние заработные платы, определяет уровень востребованности и обновляет связи с компаниями-работодателями. Между запросами выдерживается пауза для соблюдения ограничений, установленных HeadHunter на частоту обращений [9].

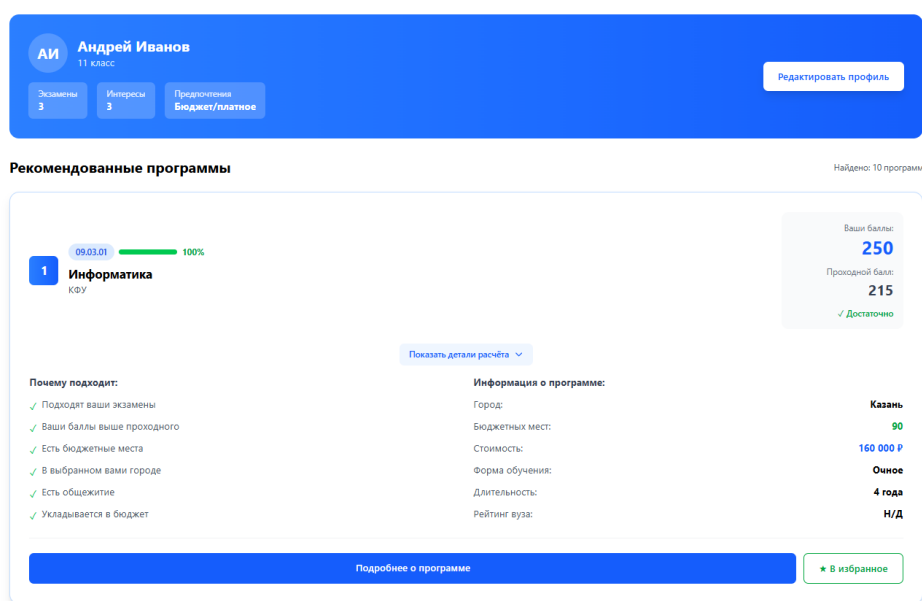


Рис. 1. Страница персонализированных рекомендаций с детализацией процента совпадения

Сравнение программ

Сравнивайте программы. Кликните на заголовки для многоуровневой сортировки.

→ К избранному

Сравниваются (6): УГАТУ... X МАИ... X ВГТУ... X ОмГУ... X ИТМО... X МГУ им. Ломоносо... X						Добавить
Параметр	09.03.01 Информатика УГАТУ Подробнее →	09.03.01 Информатика и вычислительная техника МАИ Подробнее →	09.03.01 Информатика ВГТУ Подробнее →	09.03.01 Информатика ОмГУ Подробнее →	09.03.01 Информатика и вычислительная техни ИТМО Подробнее →	
Город	Уфа	Москва	Воронеж	Омск	Санкт-Петербург	
Рейтинг вуза	Н/Д	Н/Д	Н/Д	Н/Д	Н/Д	
Проходной балл	190	225	185	180 ▼	260	
Бюджетных мест	65	50	60	80 ▲	80 ▲	
Стоимость/год	130 000 Р	200 000 Р	140 000 Р	125 000 Р ▼	290 000 Р	
Форма обучения	Очное	Очное	Очное	Очное	Очное	
Срок обучения	4 года	4 года	4 года	4 года	4 года	
Общежитие	✓ Есть	✓ Есть	✓ Есть	✓ Есть	✓ Есть	
Военная кафедра	X Нет	✓ Есть	X Нет	X Нет	X Нет	
Необходимые ЕГЭ	Математика Информатика	Математика Информатика	Математика Информатика	Математика Информатика	Математика Информатика	

Рис. 2. Страница интерактивного сравнения программ



Карьерный трек после выпуска

Высокая востребованность

Средняя ЗП: 190 тыс. Р

Профессии по специальности

Frontend-разработчик

Разработка пользовательских интерфейсов на React/Vue/Angular

Работодатели:

VK Avito Яндекс Ozon Tech

Востребованность: Высокая

170 тыс. Р
через 2 года

Вакансии на hh.ru →

Data Engineer

Построение ETL-пайплайнов, работа с большими данными, Spark, Hadoop

Работодатели:

Яндекс Ozon Tech Сбер

Востребованность: Высокая

210 тыс. Р
через 3 года

Вакансии на hh.ru →

Рис. 3. Страница карьерного трека с профессиями и зарплатами

Заключение

В результате выполнения работы была создана рекомендательная информационная система для поддержки абитуриентов при выборе специальности. Система позволяет пользователю заполнить профиль с данными о баллах ЕГЭ и интересах, получить персонализированные рекомендации с прозрачной детализацией, сравнить до пяти программ в интерактивной таблице и оценить карьерные перспективы с актуальными данными о зарплатах и работодателях. Все перечисленные функции направлены на то, чтобы помочь абитуриенту принять взвешенное и обоснованное решение при выборе образовательной программы.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Использование диаграммы вариантов использования UML при проектировании программного обеспечения. URL: <https://habr.com/ru/articles/566218/> (дата обращения: 15.05.2026).
2. Вузопедия - сервис для абитуриентов. URL: <https://vuzopedia.ru> (дата обращения: 15.05.2026).
3. Поступи Онлайн - сервис для абитуриентов. URL: <https://postupi.online> (дата обращения: 15.05.2026).
4. Табитуриент - сервис для абитуриентов. URL: <https://tabiturient.ru> (дата обращения: 15.05.2026).
5. Поступай правильно - государственный портал для абитуриентов. URL: <https://поступайправильно.рф> (дата обращения: 15.05.2026).
6. Spring Boot Documentation. URL: <https://spring.io/projects/spring-boot> (дата обращения: 15.05.2026).
7. Spring Data JPA Documentation. URL: <https://spring.io/projects/spring-data-jpa> (дата обращения: 15.05.2026).
8. Vue.js Documentation. URL: <https://vuejs.org/guide/introduction.html> (дата обращения: 15.05.2026).
9. HeadHunter API Documentation. URL: <https://dev.hh.ru> (дата обращения: 15.05.2026).

© В. В. Шнейдер, С. Ю. Кацко, 2026

В. В. Язовских¹✉, Р. В. Гришин¹

Веб-сервис транскрипции и анализа телефонных разговоров

¹Сибирский государственный университет геосистем и технологий,
г. Новосибирск, Российская Федерация
e-mail: YAzovskih-VV2022@sgugit.ru

Аннотация. В статье представлены результаты разработки веб-сервиса для автоматической транскрипции и интеллектуального анализа телефонных разговоров. Актуальность работы связана с высокой трудоемкостью ручной обработки аудиозаписей и потребностью в инструментах, позволяющих получать не только текстовую расшифровку разговора, но и интерпретируемый аналитический результат. В работе использованы методы анализа предметной области, проектирования реляционной базы данных, построения REST API и интеграции с внешними сервисами распознавания речи и языкового анализа. Разработанный сервис обеспечивает загрузку аудиофайлов, хранение данных, транскрипцию через Yandex SpeechKit, анализ транскрипта через Yandex GPT, работу с настраиваемыми критериями оценки и отображение результата пользователю. Полученное решение может быть использовано для автоматизации анализа телефонных переговоров в организациях, а также в учебных и исследовательских целях.

Ключевые слова: веб-сервис, транскрипция, телефонные разговоры, распознавание речи, интеллектуальный анализ, база данных

V. V. Yazovskikh¹✉, R. V. Grishin¹

Web service for transcription and analysis of telephone conversations

¹Siberian State University of Geosystems and Technologies, Novosibirsk, Russian Federation
e-mail: YAzovskih-VV2022@sgugit.ru

Abstract. The article presents the results of developing a web service for automatic transcription and intelligent analysis of telephone conversations. The relevance of the study is determined by the high labor intensity of manual audio processing and the need for tools that provide not only a text transcript but also an interpretable analytical result. The study applies methods of domain analysis, relational database design, REST API development, and integration with external speech recognition and language analysis services. The developed service supports audio file upload, data storage, transcription using Yandex SpeechKit, transcript analysis using Yandex GPT, work with customizable evaluation criteria, and presentation of the result to the user. The proposed solution can be used to automate the analysis of telephone conversations in organizations, as well as for educational and research purposes.

Keywords: web service, transcription, telephone conversations, speech recognition, intelligent analysis, database

Введение

Цифровизация прикладных процессов обработки данных приводит к активному использованию интеллектуальных методов в задачах, где ранее доминировал ручной анализ. Одним из наиболее сложных и одновременно практически значимых видов данных являются телефонные разговоры, широко применяемые

в продажах, клиентском сервисе и сопровождении пользователей. Такие разговоры содержат не только факт контакта, но и информацию о цели коммуникации, потребностях клиента, ходе диалога, достигнутых договоренностях и качестве работы сотрудника. В информационных системах, ориентированных на поддержку принятия решений, подобные сведения имеют высокую прикладную ценность [1, 2].

На практике обработка телефонных переговоров во многих случаях остается ручной. Прослушивание аудиозаписей, составление кратких выводов и оценка качества общения требуют значительных временных затрат и плохо масштабируются при росте объема звонков. Дополнительной проблемой является субъективность оценки: два специалиста могут по-разному интерпретировать одни и те же реплики, по-разному выделять смысловые акценты и формулировать рекомендации. В результате у организации отсутствует единый воспроизводимый механизм анализа аудиокоммуникаций.

Современные средства автоматического распознавания речи позволяют переводить аудиосигнал в текстовую форму, а языковые модели дают возможность интерпретировать содержание диалога, выделять роли участников и формировать структурированные аналитические выводы. Однако в реальных прикладных задачах востребовано не изолированное применение отдельных технологий, а их объединение в едином веб-сервисе, обеспечивающем полный цикл обработки: от загрузки аудиозаписи до получения итогового аналитического результата.

Целью исследования является разработка веб-сервиса, обеспечивающего автоматическую транскрибацию и интеллектуальный анализ телефонных разговоров. Для достижения поставленной цели были решены задачи анализа предметной области, выбора средств реализации, проектирования структуры данных и построения клиент-серверного решения, интегрированного с внешними сервисами распознавания речи и языкового анализа.

Методы и материалы

В основе разработанного решения лежит клиент-серверная архитектура, позволяющая разделить пользовательский интерфейс, серверную бизнес-логику и внешние сервисы обработки аудиоданных. Серверная часть реализована на языке Python, который широко применяется для интеграционных задач и построения веб-приложений благодаря развитой экосистеме библиотек и средств разработки [3, 4, 5, 18]. Для организации API выбран фреймворк FastAPI, обеспечивающий построение REST-маршрутов, типизацию запросов и удобную реализацию механизмов аутентификации [6, 7, 20].

Хранение структурированных данных организовано в реляционной базе PostgreSQL. Для взаимодействия с БД использована ORM SQLAlchemy, позволяющая описывать сущности предметной области в виде программных моделей и выполнять операции создания, чтения, изменения и удаления данных без ручного формирования SQL-запросов в основной части логики приложения [8, 9]. В базе данных предусмотрено хранение учетных записей пользователей, сведений

о звонках, сегментов транскрипта, критериев анализа, общих результатов анализа и детальных результатов оценки по каждому критерию.

Аудиофайлы не сохраняются непосредственно в базе данных, а размещаются во внешнем S3-совместимом объектном хранилище. Для этого используется сервис Yandex Object Storage и библиотека boto3, что позволяет централизованно хранить медиафайлы и передавать ссылки на них во внешние сервисы распознавания речи [14, 19]. Аутентификация реализована на основе токенов доступа JWT, что обеспечивает защищенный доступ к функциям системы и разграничение прав пользователя и администратора [7, 15].

Для транскрибации телефонных разговоров применяется сервис Yandex SpeechKit, позволяющий преобразовывать аудиозапись в текст и получать сегменты реплик, привязанные к условным спикерам [10, 11]. После завершения транскрибации текст разговора и активные критерии оценки передаются в сервис Yandex GPT, который возвращает структурированный аналитический результат: определение ролей участников, краткий вывод по разговору, договоренности, риски, следующий шаг, общую оценку и рекомендации [12, 13]. Пользовательский интерфейс реализован с применением стандартных веб-технологий HTML, CSS и JavaScript, а тестирование HTTP-запросов выполнялось в среде Postman [16, 17].

Результаты

В результате работы был разработан веб-сервис, обеспечивающий полный цикл обработки телефонного разговора: загрузку аудиофайла, сохранение ссылки на объект в хранилище, запуск транскрибации, формирование транскрипта, интеллектуальный анализ содержания разговора и отображение результатов пользователю. Архитектура сервиса приведена на рис. 1.

Логика обработки данных реализована последовательно. После загрузки файла создается запись о звонке и фиксируется текущий статус обработки. На следующем этапе система инициирует транскрибацию и сохраняет в базе данных текст реплик и временные границы сегментов. Затем выполняется интеллектуальный анализ транскрипта, в ходе которого учитываются активные критерии оценки, заданные администратором. Итог работы сервиса представляет собой не только расшифровку разговора, но и формализованный аналитический результат, пригодный для дальнейшего использования.

Ключевые функциональные возможности разработанного веб-сервиса представлены в табл. 1.

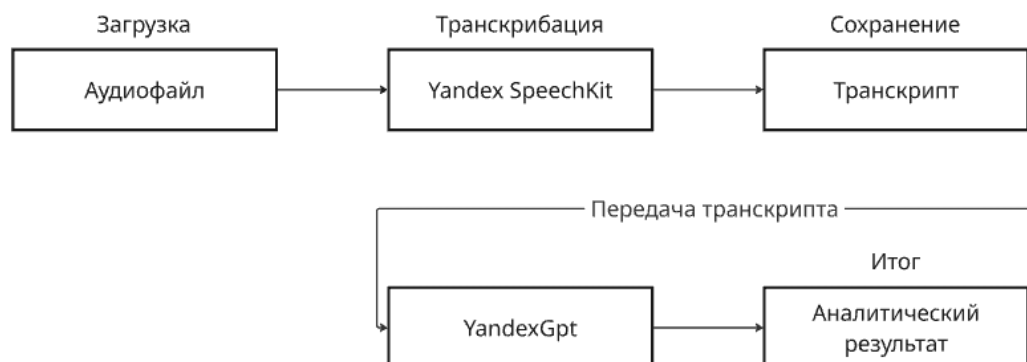


Рис. 1. Архитектура веб-сервиса транскрибации и анализа телефонных разговоров

В табл. 1 представлены основные функциональные модули системы и результаты их работы.

Таблица 1. Функциональные модули разработанного веб-сервиса

Модуль	Назначение	Входные данные	Результат
Загрузка звонка	Прием аудиофайла и создание записи о звонке	Файл, имя клиента, имя менеджера	Ссылка на объект и статус обработки
Транскрибация	Преобразование речи в текст	Ссылка на аудиофайл	Транскрипт и сегменты реплик
Интеллектуальный анализ	Интерпретация содержания разговора	Транскрипт и активные критерии	Роли, итог, риски, оценка, рекомендации
Управление критериями	Настройка параметров анализа	Название, описание, вес, активность	Изменяемый набор критериев
Пользовательский интерфейс	Отображение результатов и управление данными	Ответы API	Страницы звонков, анализа, профиля и критериев

Реализованный сервис включает модуль авторизации, модуль работы со звонками, модуль транскрибации, модуль интеллектуального анализа, модуль критериев оценки и модуль управления пользователями. После авторизации пользователь получает доступ к списку звонков, может загрузить новый аудиофайл, инициировать его обработку и просмотреть результат. Администратор, помимо этого, управляет учетными записями и составом критериев анализа.

Принципиальным результатом разработки является переход от хранения необработанных аудиозаписей к формированию интерпретируемой структуры данных.

Система сохраняет не только исходный файл и сплошной текст транскрипта, но и сегменты разговора, результаты анализа по каждому критерию, а также общий вывод по звонку. Это позволяет использовать сервис как инструмент накопления аналитических данных для последующей оценки качества коммуникации.

Обсуждение

Полученные результаты показывают, что объединение сервисов распознавания речи и языкового анализа в рамках единого веб-приложения позволяет перейти от фрагментарной обработки телефонных разговоров к последовательному и воспроизводимому сценарию анализа. В отличие от подхода, при котором пользователь получает только текстовый транскрипт, предложенное решение формирует структурированный результат, пригодный для контроля качества коммуникации и дальнейшей интерпретации.

Важным преимуществом разработанного сервиса является поддержка настраиваемых критериев анализа. Критерии не зафиксированы жестко в программном коде, а хранятся в базе данных и управляются через интерфейс администратора. Это позволяет адаптировать систему под разные прикладные сценарии без изменения серверной логики и делает решение более гибким по сравнению с системами, использующими неизменяемые правила оценки.

Вместе с тем качество аналитического вывода зависит от точности транскрибации исходного аудио и от структуры формируемого запроса к языковой модели. Перспективой дальнейшего развития работы является расширение набора критериев анализа, интеграция сервиса с CRM-платформами, использование дополнительных средств визуализации результатов и проведение сравнительного тестирования на более широком наборе аудиозаписей.

Заключение

В статье представлены результаты разработки веб-сервиса для автоматической транскрибации и интеллектуального анализа телефонных разговоров. Выполнен анализ предметной области, обоснован выбор клиент-серверного подхода, реализованы серверная и пользовательская части приложения, а также интеграция с Yandex SpeechKit, Yandex GPT и S3-совместимым объектным хранилищем.

Разработанный сервис обеспечивает получение транскрипта разговора и формирование аналитического результата, включающего определение ролей участников, общий итог, оценку по критериям и рекомендации. Практическая значимость работы заключается в возможности применения разработанного решения для автоматизации анализа телефонных переговоров в организациях, а также в учебных и исследовательских целях.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Подчукаев, В. А. Теория информационных процессов и систем : учеб. пособие для вузов / В. А. Подчукаев. – Москва : Гардарики, 2007. – 207 с.
2. Российская Федерация. Законы. Об информации, информационных технологиях и о защите информации : федер. закон N 149-ФЗ : принят Государственной Думой 8 июля 2006 г.

- : одобрен Советом Федерации 14 июля 2006 г. : послед. ред. // КонсультантПлюс : сайт. – URL: http://www.consultant.ru/document/cons_doc_LAW_61798/ (дата обращения: 26.05.2026).
3. Лутц, М. Изучаем Python. – Санкт-Петербург : Символ-Плюс, 2021.
 4. Рамальо, Л. Fluent Python. Ясное, лаконичное и эффективное программирование. – Санкт-Петербург : Питер, 2022.
 5. Лубанович, Б. Простой Python. Современный стиль программирования. – Санкт-Петербург : Питер, 2021.
 6. FastAPI Documentation. – URL: <https://fastapi.tiangolo.com/>.
 7. FastAPI | Security – OAuth2 with Password (and hashing), Bearer with JWT tokens. – URL: <https://fastapi.tiangolo.com/tutorial/security/oauth2-jwt/>.
 8. SQLAlchemy Documentation. – URL: <https://docs.sqlalchemy.org/>.
 9. PostgreSQL: Documentation. – URL: <https://www.postgresql.org/docs/>.
 10. Yandex SpeechKit | Документация. – URL: <https://yandex.cloud/ru/docs/speechkit/>.
 11. Yandex SpeechKit | Распознавание речи. – URL: <https://yandex.cloud/ru/docs/speechkit/quickstart/stt-quickstart-v1>.
 12. Yandex Foundation Models | Документация. – URL: <https://yandex.cloud/ru/docs/foundation-models/>.
 13. Yandex GPT API, совместимый с OpenAI. – URL: <https://yandex.cloud/ru/docs/foundation-models/concepts/openai-compatibility>.
 14. Yandex Object Storage | Документация. – URL: <https://yandex.cloud/ru/docs/storage/>.
 15. RFC 7519: JSON Web Token (JWT). – URL: <https://datatracker.ietf.org/doc/html/rfc7519>.
 16. MDN Web Docs. – URL: <https://developer.mozilla.org/ru/>.
 17. Postman Learning Center. – URL: <https://learning.postman.com/docs/getting-started/introduction/>.
 18. Python Documentation. – URL: <https://docs.python.org/3/>.
 19. boto3 Documentation. – URL: <https://boto3.amazonaws.com/v1/documentation/api/latest/index.html>.
 20. Pydantic Documentation. – URL: <https://docs.pydantic.dev/>.

© В. В. Язовских, Р. В. Гришин, 2026

Б. Ю. Якобсон^{1✉}

Алгоритм адаптивной фильтрации информационных потоков в корпоративных вычислительных сетях

¹Национальный исследовательский ядерный университет «МИФИ»,
г. Москва, Российская Федерация
e-mail: yazon16300@gmail.com

Аннотация. В работе рассмотрена задача адаптивной фильтрации сетевого трафика в корпоративных вычислительных сетях в условиях роста числа и сложности кибератак. Предложена гибридная двухуровневая архитектура системы обнаружения вторжений, объединяющая модуль сигнатурного анализа, реализованный на языке Go, и модуль поведенческого анализа на языке Python на основе алгоритма Isolation Forest. Для объединения результатов модулей разработан механизм Score Fusion с взвешенным усреднением оценок статистического и машинного анализа. Разработана методика тестирования системы на смешанном трафике с контролируемой долей семи классов атак, приближенная к условиям реальной эксплуатации. Проведено экспериментальное исследование эффективности на виртуализированном стенде из 48 виртуальных машин длительностью 8 часов. Получены следующие показатели: Precision=1,000, TPR=0,942, FPR=0,000, F1-мера=0,970. Проведен сравнительный анализ с учебным решением Simple Python IDS и промышленной системой Snort 3, показавший, что предложенное решение превосходит учебные аналоги и сопоставимо по ключевым метрикам с промышленными системами при существенно меньшем потреблении оперативной памяти.

Ключевые слова: адаптивная фильтрация, обнаружение вторжений, сигнатурный анализ, поведенческий анализ, Isolation Forest, Score Fusion

B. Y. Yakobson^{1✉}

Adaptive traffic filtering algorithm for corporate computer networks

¹National Research Nuclear University MEPhI, Moscow, Russian Federation
e-mail: yazon16300@gmail.com

Abstract. The paper addresses the task of adaptive traffic filtering in corporate computer networks under the growing number and complexity of cyberattacks. A hybrid two-level intrusion detection architecture is proposed, combining a signature analysis module implemented in Go and a behavioral analysis module in Python based on the Isolation Forest algorithm. A Score Fusion mechanism with weighted averaging of statistical and machine learning scores is developed to combine the outputs of the modules. A testing methodology on mixed traffic with a controlled share of seven attack classes is proposed, approximating real-world operating conditions. An experimental study was conducted on a virtualized testbed of 48 virtual machines for 8 hours. The following metrics were obtained: Precision=1.000, TPR=0.942, FPR=0.000, F1-score=0.970. A comparative analysis with the Simple Python IDS educational solution and the industrial Snort 3 system demonstrated that the proposed solution outperforms educational analogues and is comparable in key metrics to industrial systems at significantly lower RAM consumption.

Keywords: adaptive filtering, intrusion detection, signature analysis, behavioral analysis, Isolation Forest, Score Fusion

Введение

Рост масштабов и сложности корпоративных вычислительных сетей (КВС) сопровождается увеличением числа и разнообразия кибератак на инфраструктуру организаций. По данным Positive Technologies, в 2024–2025 гг. доля инцидентов с использованием программ-вымогателей в отдельных кварталах увеличивалась на 13% и более [1]. В этих условиях традиционные подходы к фильтрации трафика, основанные на статических правилах и классическом сигнатурном анализе, не обеспечивают своевременного реагирования на ранее неизвестные угрозы.

Альтернативой являются адаптивные методы, сочетающие сигнатурный и поведенческий анализ с применением машинного обучения [2, 3]. Коммерческие решения класса NGFW реализуют подобные подходы, однако требуют значительных вычислительных ресурсов [4, 5], что ограничивает их применение в организациях малого и среднего размера. В академических работах рассматриваются преимущественно либо изолированный сигнатурный анализ [6–8], либо отдельные методы поведенческого анализа [3, 9]; вопросы совместной обработки результатов статистического и ML-анализа, а также эмпирического подбора параметров такой интеграции проработаны недостаточно.

В настоящей работе предлагается гибридная двухуровневая архитектура системы обнаружения вторжений, в которой сигнатурный модуль реализован на языке Go для минимизации накладных расходов, а модуль поведенческого анализа на основе алгоритма Isolation Forest – на языке Python. Результаты модулей объединяются оригинальным механизмом Score Fusion с взвешенным усреднением оценок статистического и машинного анализа, что обеспечивает одновременную устойчивость к известным сигнатурам атак и к ранее неизвестным поведенческим аномалиям. Для оценки эффективности предложенного решения разработана методика тестирования на смешанном трафике с контролируемой долей семи классов атак (SQL-инъекции, XSS, SYN Flood, Port Scan, Host Scan, Path Traversal, Command Injection), приближенная к условиям реальной эксплуатации. Отдельное внимание уделено эмпирическому подбору порога бинарной классификации Score Fusion, обеспечивающего максимум F1-меры при сохранении нулевого уровня ложных срабатываний.

Актуальность исследования обусловлена необходимостью разработки алгоритма адаптивной фильтрации информационных потоков КВС на основе одновременного использования сигнатурных и поведенческих признаков, допускающего интеграцию в существующие средства защиты корпоративного сегмента. Предложенное решение ориентировано на защиту сетей малого и среднего размера, для которых развертывание промышленных систем класса Suricata или Snort 3 затруднено по соображениям стоимости и требуемой квалификации.

Целью исследования является разработка и экспериментальная оценка гибридного алгоритма адаптивной фильтрации информационных потоков в КВС, сочетающего сигнатурный и поведенческий анализ на основе алгоритма Isolation Forest.

Предлагаемый алгоритм

Предложенная система состоит из двух последовательно-параллельных модулей анализа и модуля объединения результатов. Сигнатурный модуль обеспечивает обнаружение известных угроз по базе регулярных выражений и паттернов (SQL-инъекции, XSS, флуд-атаки). Поведенческий модуль осуществляет анализ параметров трафика в скользящем временном окне и обнаружение статистических отклонений методами машинного обучения без учителя. Модуль Score Fusion выполняет взвешенное объединение оценок и принятие итогового решения. Общая схема обработки пакета представлена на рис. 1.

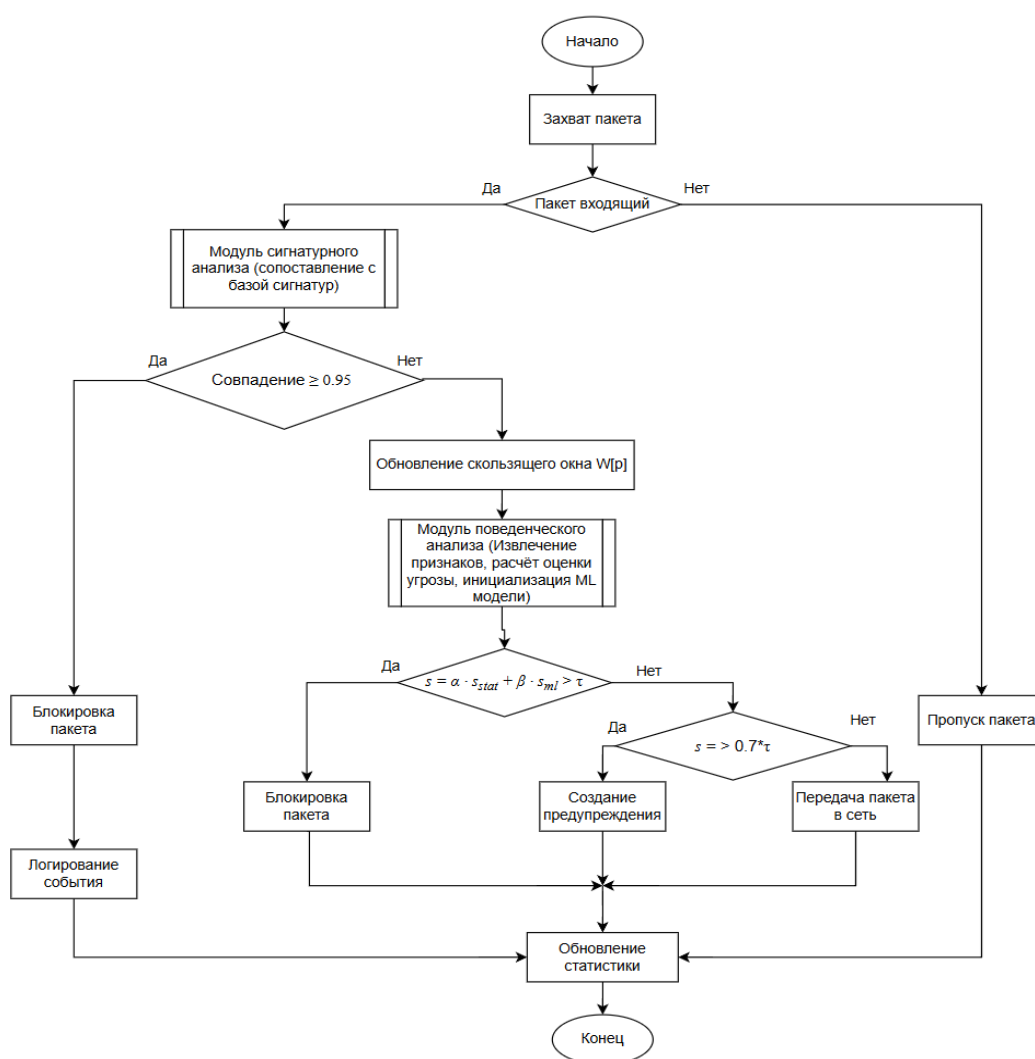


Рис. 1. Обобщенная схема алгоритма адаптивной фильтрации

В качестве модели машинного обучения выбран алгоритм Isolation Forest, обеспечивающий оценку аномальности за время $O(n \cdot \log n)$ без размеченной обучающей выборки [10]. Оценка аномальности вычисляется как

$$s(x, n) = 2^{-E(h(x))/c(n)}, \quad (1)$$

где $E(h(x))$ – математическое ожидание длины пути до изоляции наблюдения x в ансамбле случайных деревьев; $c(n)$ – нормализующий коэффициент для выборки объема n . Значения $s(x, n)$, близкие к 1, соответствуют аномалиям, близкие к 0 – нормальным наблюдениям.

Для каждого IP-адреса источника в скользящем временном окне длительностью 60 секунд извлекается вектор из 12 поведенческих признаков, описывающих активность узла: количество пакетов и байт в секунду, средний размер пакета, число уникальных портов и IP-адресов назначения, энтропия распределения портов, доля SYN-пакетов, среднее время между пакетами, энтропия содержимого и др.

Для объединения результатов необходимо, чтобы обе оценки находились в единой шкале. Сигнатурный модуль не возвращает бинарный ответ – вместо этого формируется вещественная оценка $s_{stat} \in [0, 1]$, вычисляемая как максимальный вес срабатывания сигнатуры с учетом степени совпадения (длины совпавшего паттерна, специфичности регулярного выражения) и дополнительных статистических признаков (частота срабатывания за последние 60 секунд, число уникальных правил, сработавших на данный источник). Модель машинного обучения по формуле (1) возвращает оценку $s_{ml} \in [0, 1]$ в той же шкале. Итоговая оценка угрозы формируется как взвешенное среднее двух нормализованных оценок:

$$s = \alpha \cdot s_{stat} + \beta \cdot s_{ml}, \quad \alpha + \beta = 1, \quad (2)$$

где α и β – весовые коэффициенты. В работе используются значения $\alpha = 0,4$, $\beta = 0,6$, полученные предварительной настройкой на обучающей подвыборке; больший вес модели машинного обучения обусловлен ее способностью выявлять распределенные атаки, слабо представленные в сигнатурной базе. Приведение обеих оценок к одному вещественному диапазону $[0, 1]$ обеспечивает корректность их взвешенного сложения. Решение о блокировке пакета принимается при $s \geq \tau$, где τ – порог, подбираемый эмпирически.

Для оценки качества классификации использовались стандартные метрики бинарной классификации: Precision (доля истинных срабатываний среди всех предсказанных как атакующие), Recall/TPR (доля обнаруженных атакующих источников среди всех реальных), FPR (доля ошибочно заблокированных легитимных источников) и F1-мера как гармоническое среднее Precision и Recall.

Результаты

Экспериментальное исследование предложенного алгоритма проведено на виртуализированном тестовом стенде, воспроизводящем типовую структуру КВС средней организации. Стенд включал 48 виртуальных машин в четырех VLAN-сегментах: 30 рабочих станций, 8 серверов приложений, 4 публичных сервиса в DMZ и 6 атакующих узлов под управлением Kali Linux. Центральный узел

защиты на Ubuntu Server 24.04 размещен на границе серверного сегмента в режиме прозрачного моста. Общая продолжительность эксперимента составила 8 часов непрерывной работы стенда, за которые было обработано порядка 4,82 млн пакетов, включающих 80,7% фоновое трафика и 19,3% атакующего. Обучение модели Isolation Forest выполнено на предварительно собранной выборке из 102 340 векторов поведенческих признаков, полученной на изолированной конфигурации стенда при отсутствии атакующей активности; среднее значение F1-меры на пятикратной стратифицированной кросс-валидации составило 0,91 со стандартным отклонением 0,02, что свидетельствует об отсутствии переобучения. Эксперимент проводился трижды с недельными интервалами для контроля воспроизводимости.

Таблица 1. Сравнение разработанного решения с Simple Python IDS и Snort 3

Параметр	Simple IDS	Разработанное решение	Snort 3
Архитектура	Монолит	Гибрид (2 модуля)	Модульная
Сигнатурный анализ	Да	Да	Да
ML-анализ	Нет	Да (Isolation Forest)	Нет
Пропускная способность, Мбит/с	60	620	940
TPR	0,70	0,942	0,951
FPR	0,12	0,000	0,014
Precision	0,65	1,000	0,968
F1-мера	0,67	0,970	0,960
Потребление RAM, МБ	120	248	512
Загрузка CPU-ядра, %	68	41	58

По результатам тестирования при оптимальном значении порога Score Fusion $\tau = 0,55$ система корректно классифицировала 278 атакующих источников из 295, при этом из 312 наблюдаемых источников фоновое трафика ни один не был ошибочно помечен как угроза. Итоговые значения метрик качества составили: Precision=1,00, TPR (Recall)=0,942, FPR=0, F1-мера=0,970. При консервативном значении порога $\tau = 0,60$, рекомендуемом для развертываний с повышенными требованиями к устойчивости, F1-мера снижается до 0,89 за счет увеличения доли пропусков атак с пограничными оценками угрозы. Для сравнения эффективности разработанного решения с существующими на том же стенде независимо развернуты учебный проект Simple Python IDS и промышленная система Snort 3 (версия 3.1.77) с набором правил Emerging Threats Open. Все три системы последовательно тестировались на одном и том же сгенерированном трафике. Результаты сравнения приведены в табл. 1.

Обсуждение

В отличие от учебного проекта Simple Python IDS, предложенный алгоритм использует двухуровневую схему принятия решения: быстрый сигнатурный этап блокирует трафик с явными признаками известных атак без накопления контекста, а поведенческий этап на базе Isolation Forest выявляет аномалии у источников с недостаточно выраженными сигнатурами. Такая схема позволяет достичь полного отсутствия ложных срабатываний ($FPR=0$) при сохранении высокой полноты обнаружения, что особенно важно при практической эксплуатации, где каждое ложное срабатывание требует ручного разбора оператором и увеличивает нагрузку на подразделение мониторинга.

Сравнение с промышленной системой Snort 3 показывает, что разработанное решение достигает сопоставимых значений F1-меры (0,970 против 0,960) и превосходит Snort 3 по Precision за счет поведенческого фильтра, отсеивающего нетипичные, но легитимные всплески активности. В то же время Snort 3 сохраняет преимущество по пропускной способности (940 против 620 Мбит/с) благодаря многолетней оптимизации движка и низкоуровневой реализации. Таким образом, предложенное решение не претендует на замену промышленных систем в высоконагруженных сегментах, однако оказывается конкурентоспособным в сетях малого и среднего размера при вдвое меньшем потреблении оперативной памяти.

Отдельного внимания заслуживает вопрос применимости алгоритма к шифрованному трафику, доля которого в современных КВС превышает половину. Из 12 используемых поведенческих признаков большинство (количество пакетов и байт в секунду, число уникальных портов и IP-адресов назначения, энтропия распределения портов, доля SYN-пакетов, среднее время между пакетами) извлекаются из заголовков сетевого и транспортного уровней и сохраняют информативность независимо от шифрования полезной нагрузки. Ограниченно применимой остается лишь энтропия содержимого пакетов, теряющая дискриминирующую способность на уже зашифрованных потоках. Это позволяет ожидать сохранения основных показателей качества обнаружения для шифрованного трафика, что составляет предмет отдельной экспериментальной проверки. Дополнительным направлением дальнейших исследований является анализ многоэтапных АРТ-сценариев с длинным временным горизонтом, выходящим за пределы текущего окна наблюдения в 60 секунд.

Заключение

В ходе исследования разработан гибридный алгоритм адаптивной фильтрации информационных потоков в корпоративных вычислительных сетях, объединяющий сигнатурный и поведенческий анализ через механизм Score Fusion с взвешенным усреднением. Предложенная методика тестирования на смешанном трафике с контролируемой долей семи классов атак и масштабный экспериментальный стенд из 48 виртуальных машин позволили получить показатели $Precision=1,000$, $TPR=0,942$, $FPR=0,000$ и $F1\text{-мера}=0,970$, сопоставимые с про-

мышленной системой Snort 3 при вдвое меньшем потреблении оперативной памяти.

В дальнейшем планируется исследование устойчивости алгоритма к шифрованному трафику и многоэтапным АРТ-атакам, реализация автоматической подстройки порога Score Fusion на основе онлайн-оценки распределения оценок угрозы, а также интеграция с промышленными SIEM-системами через стандартизованные интерфейсы.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Актуальные киберугрозы: III квартал 2025 года [Электронный ресурс]. – URL: <https://ar2025.ptsecurity.com/> (дата обращения: 15.05.2026).
2. Басыня Е. А., Равтович Ю. К. Разработка модуля системы обнаружения и предотвращения вторжений // Современные материалы, техника и технологии. – 2016. – № 2 (5). – С. 26–29.
3. Программная реализация интеллектуально-адаптивного управления // Вестник цифровых технологий. – 2021. – № 3. – С. 56–60.
4. Межсетевые экраны нового поколения: обзор технологий // Вестник информационных технологий. – 2020. – № 5 (3). – С. 34–40.
5. Подключение к Usergate DCFW [Электронный ресурс]. – URL: https://docs.usergate.com/podklyuchenie-k-usergate-dcfw_1601.html (дата обращения: 15.05.2026).
6. Алексеев А. В. Анализ систем обнаружения вторжений // Экономика и социум. – 2017. – № 6 (37). – С. 686–688.
7. Анчишкин А. П. Сравнительный анализ современных систем обнаружения вторжений // Colloquium-journal. – 2019. – № 26 (50). – С. 25–26.
8. Белова А. Л., Бородавкин Д. А. Сравнительный анализ систем обнаружения вторжений // Актуальные проблемы авиации и космонавтики. – 2016. – Т. 1. – С. 742–743.
9. Адаптивный метод фильтрации аномалий // Технологии информационной безопасности. – 2021. – № 2. – С. 45–49.
10. Liu F. T., Ting K. M., Zhou Z.-H. Isolation Forest // Proceedings of the 8th IEEE International Conference on Data Mining (ICDM '08). – Pisa, Italy : IEEE Computer Society, 2008. – P. 413–422. – DOI: 10.1109/ICDM.2008.17.

© Б. Ю. Якобсон, 2026

Е. Н. Яковлев¹✉, А. А. Шаронов¹

Разработка системы аналитики и визуализации данных для веб-платформы управления научными конференциями

¹Сибирский государственный университет геосистем и технологий,
г. Новосибирск, Российская Федерация
e-mail: jakovlev_egor@mail.ru

Аннотация. В статье предложена и обоснована распределенная микросервисная архитектура на базе асинхронной среды FastAPI и реактивного фреймворка Vue.js. Описан механизм внутридоменного сбора телеметрии, опирающийся на алгоритм генерации цифровых отпечатков и полностью исключающий зависимость от cookie-файлов. Детально рассмотрен подход к снижению вычислительной нагрузки на сервер приложений путем делегирования процессов агрегации логов на уровень реляционной СУБД PostgreSQL. Данная методика позволила реализовать мгновенный рендеринг тепловых карт активности и воронок конверсии без ущерба для основного бизнес-ядра. Приведенные метрики тестирования подтверждают высокую отказоустойчивость предложенного решения в условиях интенсивного конкурентного трафика.

Ключевые слова: микросервисная архитектура, фоновая телеметрия, асинхронные интерфейсы, агрегация на уровне СУБД

E. N. Yakovlev¹✉, A. A. Sharapov¹

Development of an analytics and data visualization system for a web-based scientific conference management platform

¹Siberian State University of Geosystems and Technologies, Novosibirsk, Russian Federation
e-mail: jakovlev_egor@mail.ru

Abstract. This article proposes and justifies a distributed microservice architecture based on the asynchronous FastAPI environment and the reactive Vue.js framework. It describes a mechanism for intra-domain telemetry collection that relies on a fingerprinting algorithm and completely eliminates the need for cookies. The approach to reducing the computational load on the application server by delegating log aggregation processes to the PostgreSQL relational DBMS is examined in detail. This method enabled the implementation of instant rendering of activity heatmaps and conversion funnels without compromising the core business logic. The testing metrics presented confirm the high fault tolerance of the proposed solution under conditions of intense competitive traffic.

Keywords: microservices architecture, background telemetry, asynchronous interfaces, aggregation at the DBMS level

Введение

Эволюция цифровых сред академического взаимодействия демонстрирует выраженный дисбаланс между функциональностью платформ приема статей и инструментами оценки их эффективности. Практика проведения научных мероприятий выявляет критическую уязвимость порталов при массовой загрузке отчетных материалов в преддверии дедлайнов. В такие моменты административ-

ному аппарату жизненно важно иметь доступ к объективной картине происходящего для оперативного масштабирования вычислительных мощностей.

Следствием этого становится необходимость проектирования и имплементации глубоко интегрированного, суверенного контура аналитики. Цель настоящего исследования заключается в создании независимой телеметрической подсистемы на основе микросервисной архитектуры и механизма бескуки-идентификации. Такая подсистема должна функционировать в скрытом режиме и обеспечить полный охват аудитории научного портала без снижения производительности его основного бизнес-ядра.

Исследование предметной области

Традиционный ландшафт университетских информационных систем представляет собой преимущественно монолитные решения. Их архитектура изначально не рассчитана на агрессивные флуктуации пользовательского трафика, когда подавляющий объем транзакций, связанных с выгрузкой объемных бинарных файлов (текстов докладов), концентрируется в узком временном коридоре.

Очевидным шагом к мониторингу подобных ситуаций кажется интеграция общедоступных метрических скриптов (коммерческих систем веб-аналитики). Однако глубокий анализ применимости таких решений в образовательном секторе выявляет два непреодолимых препятствия.

Прежде всего, отправка навигационных логов и метаданных исследователей в транснациональные ИТ-компании идёт вразрез с действующими в вузах правилами защиты конфиденциальной информации и политикой цифрового суверенитета. Далее, повсеместное внедрение пользователями клиентских расширений для фильтрации контента сводит на нет любые попытки объективного сбора статистики. Эвристические алгоритмы блокировщиков рекламы с высокой точностью идентифицируют и прерывают HTTP-соединения с доменами известных трекеров. Это искусственно занижает показатели вовлеченности на величину, порой превышающую сорок процентов [1, 2].

Следовательно, использование внешних облачных решений для аналитики научных платформ является нецелесообразным, что формирует четкий технический запрос на разработку децентрализованных, внутридоменных модулей телеметрии, не зависящих от сторонней инфраструктуры.

Методы и технологии

В основу архитектурного каркаса заложен принцип API Composition, предполагающий дробление экосистемы на узкоспециализированные домены [3, 4].

Основное бизнес-ядро, управляющее графами связей между конференциями, секциями и авторами, делегировано программной среде, ориентированной на строгую валидацию транзакций [5]. Параллельно с ним был развернут исследуемый аналитический узел. Для его конструирования использован инструмент FastAPI, чей выбор аргументируется не просто данью современным трендам, а строгой математической целесообразностью [6]. Реализация протокола ASGI позволяет данной части переводить операции ввода-вывода в разрешаю-

щий режим, что показывает возможность сервера поддерживать в оперативной памяти большое количество незавершённых соединений от браузеров без приостановки основного цикла обработки событий [7, 8]. Это, в свою очередь, обеспечивает непрерывность выполнения запросов.

Для обхода клиентских фильтров безопасности скрипт-перехватчик не подключался как внешний ресурс, а был скомпилирован нативно – прямо в ядро фронтенд-приложения Single Page Application. Из-за данного подхода эвристические анализаторы браузеров воспринимают его как часть стандартной логики сайта. Сигнал на отправку лога генерируется после того, как навигационный хук завершил свою работу, что отсекает ложные срабатывания и фиксирует только фактические переходы по страницам.

Главной сложностью подобного скрытого трекинга является необходимость различать пользователей, не нарушив их приватности. По той причине, что подход с применением файлов cookie ведет за собой серьезные юридические риски, от них пришлось отказаться. Идентификация построена на базе алгоритма локальной криптографической маркировки сессий [9].

Для обеспечения правильной работы метрики «Средняя продолжительность визита» и проблем, возникающих из-за длительного простоя открытой вкладки в браузере (вплоть до нескольких недель), в алгоритм был добавлен механизм пассивной инвалидации, который позволяет отсеивать незначимые данные и повышать точность статистики. При каждом действии скрипт сверяет разницу во времени с предыдущим кликом. Если она превышает определенное количество минут, то старый ключ стирается из памяти, а пользователь начинает считаться новым.

Для передачи пакетов на телеметрический узел используется браузерный API асинхронной диспетчеризации. Главное преимущество данного метода заключается в гарантированной доставке пакета в фоновом режиме, что позволяет минимизировать потери данных телеметрии в пограничных сценариях работы веб-приложения. На принимающей стороне происходит парсинг и обогащение логов. HTTP-заголовок, содержащий сигнатуру программного обеспечения клиента, проходит через лексический анализатор, который получает следующие данные: семейство ОС, ядро браузера и форм-фактор устройства, которые на следующем шаге передаются в реляционную таблицу [10].

Самым уязвимым местом аналитической системы является построение дашбордов по массивам из огромного количества записей. Выгрузка миллионов строк из базы данных в оперативную память Python-приложения приводит к отказу в обслуживании. Проблема решена смещением вычислений на уровень базы данных PostgreSQL [11]. Сложные SQL-запросы с использованием обобщенных табличных выражений и фильтров HAVING позволяют ядру СУБД самостоятельно отсеивать «сессии-отказы», в которых отсутствует активность пользователей на странице [12].

Таким образом, был использован следующий технологический стек:

– язык программирования Python, асинхронный фреймворк FastAPI;

- объектно-реляционная система управления базами данных PostgreSQL с логическим разделением контуров на уровне схем;
- JavaScript-фреймворк Vue.js 3 с использованием паттернов реактивного состояния, а также библиотека Chart.js для программного рендеринга дашбордов;
- протоколы межсервисного взаимодействия на базе Axios, маршрутизация пакетов Nginx, а также механизмы криптографической верификации на основе стандарта JSON Web Token.

Результаты

Результатом работы стал ввод в эксплуатацию независимого аналитического контура, интерфейсная часть которого интегрирована в рабочее пространство администратора конференций. Рендеринг визуальных компонентов базируется на аппаратном ускорении HTML5 Canvas, что снижает вероятность замедления при перерисовке сложных полигональных структур (рис. 1).

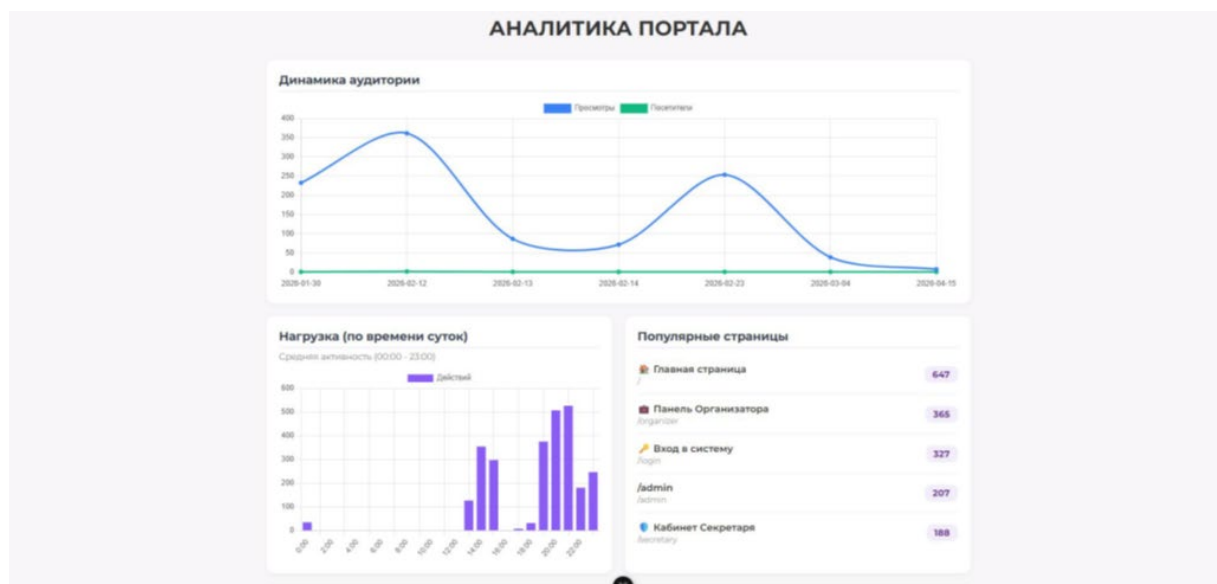


Рис. 1. Панель мониторинга пользовательской активности

Система динамически транслирует унифицированные идентификаторы ресурсов в паттерны, которые облегчают интерпретацию данных. Изменение временных фильтров на клиенте инициализирует пересчет SQL-матриц на сервере, возвращая легковесные JSON-ответы, содержащие итоговые массивы координат для осей графиков.

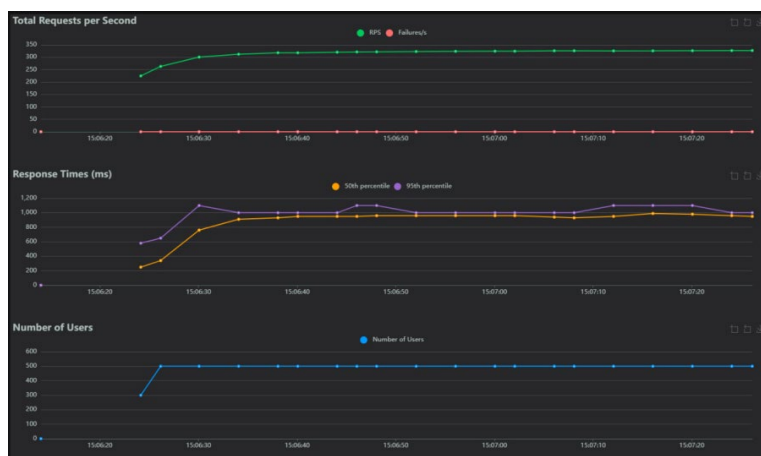
С целью верификации надежности спроектированного решения был инициирован цикл автоматизированного тестирования. Специфика проверок охватывала устойчивость лексического парсера к искаженным или злонамеренно модифицированным метаданным, а также валидацию результатов математической группировки. Результаты инструментального контроля приведены в табл. 1.

Таблица 1. Отчет инструментального анализа покрытия алгоритмов тестами

Компонент микросервиса	Объем кода	Пропуски	Уровень покрытия, %
Слой подключения к БД	22	0	100
Модуль бизнес-логики	69	1	99
Модели реляционных таблиц	17	0	100
Контроллер сценариев	22	0	100

Анализ табл. 1 показывает, что индекс покрытия исполняемой бизнес-логики достигает 99 %. Единственный зафиксированный пропуск относится к системному вызову процесса, инициирующего прослушивание портов операционной системы, что не подлежит проверке в рамках изолированных модульных тестов.

Помимо модульных проверок, для подтверждения заявленных архитектурных преимуществ асинхронного подхода (ASGI) было проведено нагрузочное стресс-тестирование модуля сбора телеметрии с использованием инструментария Locust. Эмулировался сценарий пиковой активности в момент массовой загрузки докладов, предполагающий наличие 500 конкурентных соединений, ежедневно генерирующих POST-запросы с метаданными. Метрики производительности представлены на рис. 2.

**Рис. 2.** Результаты стресс-тестирования

Анализ графиков (рис. 2) показывает высокую пропускную способность сервера, который без проблем держит нагрузку порядка 320-330 запросов в секунду. При этом важный показатель отказов остается на нуле, что показывает отсутствие потерь пакетов данных. Медианное время отклика при максимальной нагрузке стабилизируется в пределах секунды. Однако для пользователя эта задержка незаметна: поскольку браузер отправляет телеметрию в фоне, задержки в работе основной части портала отсутствуют.

Заключение

Итоги разработки подтверждают, что созданная система полностью удовлетворяет требованиям и справляется со всеми задачами. Внедрение продукта на

сервера учебного заведения закрывает вопрос цифрового суверенитета, так как вуз получает полный контроль над метриками. Перенос тяжелой комбинаторики и агрегационных процессов на сторону ядра PostgreSQL доказал свою эффективность, обеспечив быстрое формирование тепловых карт активности даже при экспоненциальном росте базы логов.

Исключение файлов cookie из архитектуры системы позволило устранить правовые риски, связанные с соблюдением строгих нормативных требований в отношении обработки персональных данных. Такой подход снижает нормативную нагрузку и соответствует принципам минимизации сбора персональных данных, закрепленным в действующем законодательстве (в том числе в Федеральном законе № 152). Созданный программный комплекс обладает запасом прочности и не подвержен деградации производительности в критические периоды завершения регистрационных кампаний. Перспективы развития заложенной архитектуры открывают пути для внедрения предиктивных математических моделей. Анализ собранных исторических массивов позволит в будущем проактивно прогнозировать моменты пиковых нагрузок и автоматизировать выделение аппаратных ресурсов серверов.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Алексеев С. Н. Методы защиты пользовательских данных в веб-аналитике // Информационная безопасность регионов. – 2022. – № 1. – С. 55–61.
2. Макаров С. В. Анализ влияния эвристических блокировщиков рекламы на достоверность веб-статистики // Вестник компьютерных и информационных технологий. – 2022. – № 2. – С. 18–25.
3. Ричардсон К. Микросервисы. Паттерны разработки и рефакторинга. – Санкт-Петербург : Питер, 2019. – 544 с.
4. Баранов И. В. Применение микросервисной архитектуры в распределённых информационных системах // Современные информационные технологии и ИТ-образование. – 2021. – Т. 17, № 3. – С. 601–610.
5. Костин Д. Ю. Использование REST-архитектуры при разработке веб-приложений // Молодой ученый. – 2021. – № 15. – С. 12–18.
6. Маржаков Д. А. Разработка высоконагруженных веб-приложений на Python // Информационные технологии. – 2021. – № 7. – С. 25–31.
7. Жуков Е. А. Асинхронное программирование в Python для веб-сервисов // Программные продукты и системы. – 2021. – № 4. – С. 715–722.
8. Гусев П. А. Анализ производительности асинхронных API-серверов // Вестник современных исследований. – 2023. – № 6. – С. 73–79.
9. Куракин А. С. Разработка системы сбора цифровых отпечатков браузера (Browser Fingerprinting) для идентификации пользователей // Программная инженерия. – 2020. – Т. 11, № 8. – С. 488–496.
10. Вьюгин В. В. Методы обработки и анализа больших данных в веб-системах // Научно-технический вестник информационных технологий, механики и оптики. – 2020. – Т. 20, № 6. – С. 912–919.
11. Гаврилов А. В. Методы оптимизации SQL-запросов в PostgreSQL // Системный администратор. – 2022. – № 5. – С. 44–49.
12. Моргунов Е. PostgreSQL. Основы оптимизации SQL-запросов. – Москва : ДМК Пресс, 2021. – 256 с.

© Е. Н. Яковлев, А. А. Шаранов, 2026

П. А. Яловой¹✉, Е. А. Зиновьев¹, И. Д. Шатц¹, Н. С. Косарев¹, А. А. Шаранов¹

Прототип автономного бюджетного датчика мониторинга деформаций бортов и отвальных массивов

¹Сибирский государственный университет геосистем и технологий,
г. Новосибирск, Российская Федерация
e-mail: pavelalovoj404@gmail.com

Аннотация. В статье рассматривается разработка прототипа малобюджетного датчика для геотехнического мониторинга отвальных массивов. Актуальность работы обусловлена высокой подвижностью техногенных отвалов и необходимостью своевременного выявления деформаций для предотвращения аварийных ситуаций. Цель исследования заключается в создании компактного измерительного комплекса, обеспечивающего непрерывный контроль углов наклона. Представлены результаты тестирования разработанного устройства, подтвердившие его работоспособность и точность измерений. Экспериментальная проверка показала стабильную работу системы передачи данных и возможность ее практического применения для мониторинга деформаций.

Ключевые слова: геотехнический мониторинг, отвальные массивы, датчик наклона

P. A. Yalovoi¹✉, E. A. Zinovev¹, I. D. Shatts¹, N. S. Kosarev¹, A. A. Sharapov¹

Prototype of a low-cost sensor for monitoring displacements of waste dumps

¹Siberian State University of Geosystems and Technologies, Novosibirsk, Russian Federation
e-mail: pavelalovoj404@gmail.com

Abstract. The article presents the development of a low-cost sensor prototype for geotechnical monitoring of waste dumps. The relevance of the study is due to the high mobility of technogenic dumps and the critical need for timely deformation detection to prevent emergency situations. The purpose of the research is to design a compact measuring system that ensures continuous monitoring of surface tilt angles. The paper presents testing results of the developed device, confirming its operational reliability and measurement accuracy. Experimental verification demonstrated stable data transmission and confirmed the feasibility of practical implementation for continuous deformation monitoring.

Keywords: geotechnical monitoring, waste dumps, tilt sensor

Введение

Отвальные массивы горных пород характеризуются высокой подвижностью, склонностью к оползневым процессам и деформациям под воздействием атмосферных осадков, динамических нагрузок и изменения гидрогеологических условий [1]. Раннее выявление предвестниковых смещений позволяет своевременно принимать меры по обеспечению безопасности персонала и техники, предотвращая техногенные катастрофы [2]. Существующие на рынке коммерческие предложения обеспечивают высокую точность измерений, однако их стоимость часто делает невозможным организацию плотной сети распределенных

наблюдений [3]. В связи с этим актуальной задачей является разработка мало-бюджетных, но функционально достаточных средств мониторинга.

Целью данной работы является разработка автономного бюджетного датчика мониторинга деформаций бортов и отвальных массивов на базе недорогих решений, имеющихся на рынке микроэлектроники.

Для решения поставленной цели необходимо было выполнить следующие задачи:

- разработать прототип устройства и программное обеспечение для его управления;
- разработать web-приложение для визуализации результатов измерений;
- выполнить оценку точности, получаемых устройством измерений, а также провести оценку автономности его работы.

Методы и материалы

Разработанный прототип представляет собой измерительно-передающий комплекс, базовой платформой которого служит микроконтроллер Arduino Nano. Выбор данной платформы обусловлен ее компактными размерами, низким энергопотреблением и достаточной производительностью для решения поставленных задач. В качестве первичного преобразователя используется модуль MPU6050, объединяющий трехосевой акселерометр и трехосевой гироскоп, в котором обработка данных осуществляется встроенным цифровым процессором движения (Digital Motion Processor, DMP), позволяющим выполнять фильтрацию шумов, компенсацию дрейфа гироскопа и расчет углов крена и тангажа автоматически [4]. Передача данных на удаленный сервер реализована с помощью GSM-модуля SIM800L, поддерживающего стандарты GPRS.

Питание датчика осуществляется от блока литий-ионных аккумуляторов типа 18650 через понижающий преобразователь MP1584EN, формирующий стабилизированное напряжение 4 В для питания микроконтроллера и периферийных устройств. Применение аккумуляторов обеспечивает автономную работу устройства в полевых условиях без необходимости подключения к стационарной сети электроснабжения (рис. 1).

Все компоненты смонтированы на односторонней печатной плате размером 70×90 мм. Плата помещена в защитный корпус, изготовленный методом FDM 3D-печати из ударопрочного пластика (PETG), что обеспечивает механическую защиту электроники в условиях открытой карьерной площадки. Крепление датчика к контролируемой поверхности осуществляется через металлическую резьбовую шпильку диаметром 12 мм, интегрированную в нижнюю часть корпуса (рис. 2).

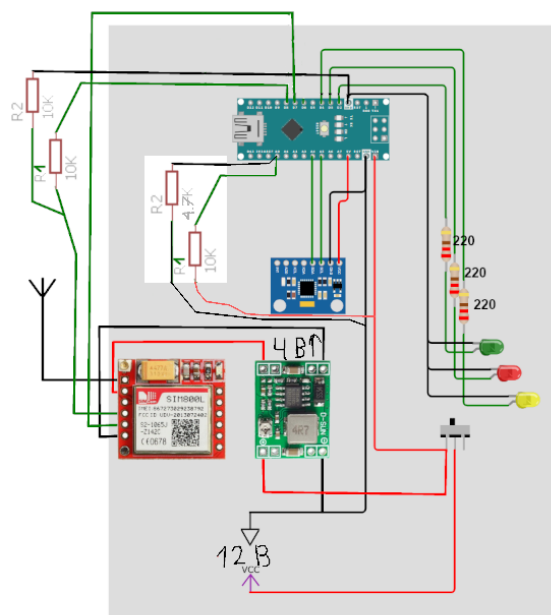


Рис. 1. Принципиальная схема автономного датчика



Рис. 2. Внешний вид корпуса

Сбор данных и их отправка происходят циклически. Микроконтроллер опрашивает MPU6050, формирует строку вида R: XX.XX, P: XX.XX (градусы с точностью до сотых), устанавливает GPRS-соединение и передает пакет методом GET на HTTP-сервер. После успешной передачи устройство переходит в режим ожидания для оптимизации сетевого трафика.

Для приема и визуализации данных развернуто веб-приложение, состоящее из технического сайта с HTTP протоколом и главного сайта, защищенного сертификатом. Сделано это из-за особенности GSM-модуля, который может сво-

бодно отправлять запросы на незащищенные шифрованием сервера. Прокси-домен состоит из двух файлов `proxy.php` и очереди `process\_queue.php`. Скрипт принимает HTTP-запросы, получает данные в виде строк с параметрами напряжения, углами крена и тангажа, а также температуры, и пересылает их на основной сайт. На главном сайте принимает сообщения с прокси файл `log.php`, сохраняя полученные значения в базе данных с временной меткой. Веб-интерфейс основного сайта отображает карту с местоположением датчиков и боковое меню с краткой сводкой по датчикам, закрепленных за организацией, в которой состоит пользователь. Просмотреть детально показатели датчика позволяет файл `sensor\_details.php` подгружая из базы данных текущие значения в виде карточек, строит интерактивные графики на базе библиотеки Chart.js и поддерживает фильтрацию по временным промежуткам (рис. 3, 4).

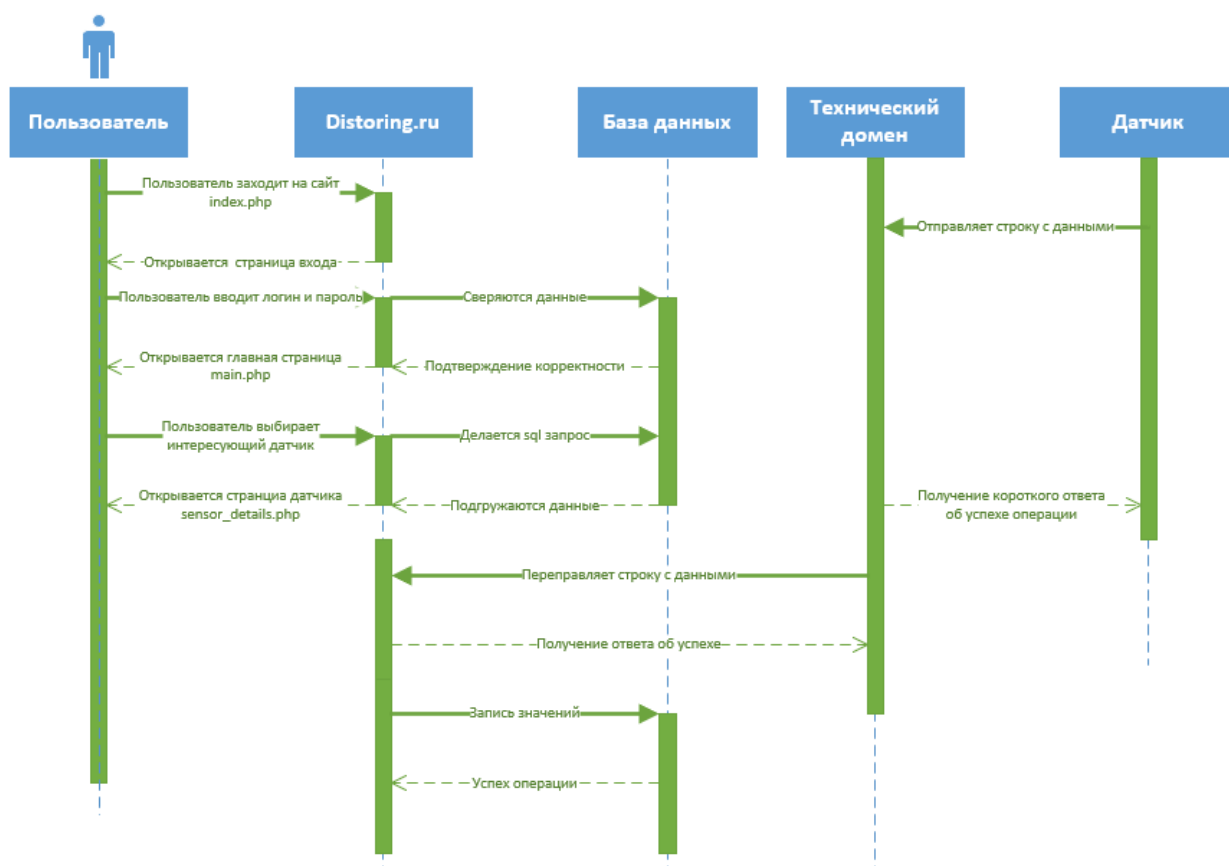


Рис. 3. Диаграмма последовательности системы

Приложение также поддерживает выбор часового пояса. Кроме этого реализована система аварийного оповещения: при отклонении измеренного угла от базового значения более чем на $1,5^\circ$ на дашборде автоматически появляется визуальное предупреждение, а точки на графике маркируются красным цветом. Архитектура приложения рассчитана на развертывание на бюджетных хостингах без использования тяжелых СУБД, что соответствует общей концепции экономической системы мониторинга.

На web-приложение для визуализации результатов мониторинга смещений отвальных массивов получено свидетельство о регистрации программы для ЭВМ [5].



Рис. 4. Страница значений датчика в веб-приложении

Результаты

Для оценки работоспособности разработанного датчика была проведена серия экспериментов, включающая проверку точности измерений углов наклона и стабильности передачи данных.

Во время оценки точности измерений проводилась серия статических измерений в разных положениях корпуса и сравнением с эталонным инклинометром. Тестирование системы показало точность измерения углов наклона в пределах $\pm 0,5^\circ$ при статическом положении.

Для тестирования стабильности передачи данных измерения с датчика отправлялись в течение продолжительного периода для оценки количества потерь пакетов и падений связи. В результате тестирования стабильность передачи данных составляет более 97% успешных операций, при общем количестве наблюдений в сутках более 8000.

Веб-приложение обрабатывало и записывало в базу данных все успешно полученные значения. Наблюдения отображаются в виде графиков, позволяя контролировать состояние объекта.

Заключение

В результате работы создан прототип автономного бюджетного датчика мониторинга смещений отвальных массивов. Аппаратная часть базируется на мик-

роконтроллере Arduino Nano, MEMS-датчике MPU6050 и GSM-модуле SIM800L, размещенных на компактной печатной плате 70×90 мм и заключенных в 3D-печатный корпус.

Разработанное веб-приложение позволяет осуществлять удаленный мониторинг, визуализацию данных и оперативное аварийное оповещение при превышении критических значений наклона.

Результаты тестирования подтвердили работоспособность системы и ее соответствие заявленным характеристикам:

- точность измерений углов наклона: $\pm 0.5^\circ$;
- надежность передачи данных: 97%;
- время автономной работы: до 100 дней (при интервале передачи 1 раз в 2 минуты).

Система способна обеспечить контроль сети датчиков, что позволит контролировать протяженные объекты, а за счет того, что плотность их установки будет выше, а также выявить движения на локальных участках на ранней стадии мониторинга.

Экономическая эффективность и простота развертывания делают разработанную систему перспективным инструментом для организации геотехнического мониторинга на предприятиях горнодобывающей промышленности.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Низаметдинов Ф.К., Барышников В.Д., Оралбай А.О. Оценка устойчивости бортов карьера Кентобе на основе разработанной цифровой геолого-геомеханической модели месторождения // Физико-технические проблемы разработки полезных ископаемых. 2022. № 6. С. 27-34. DOI: 10.15372/FTPRPI20220603.
2. Казанцева В.В., Ожигин Д.С., Косарев Н.С., Сатбергенова А.К., Ожигина С.Б. Разработка комплексной системы геотехнического мониторинга техногенных объектов на основе геопространственных данных // Записки Горного института. 2025. Т. 276. Вып. 1. С. 142-156. EDN OKPFFR.
3. Косарев Н. С. О необходимости создания отечественного программного обеспечения для автоматизированного геодезического мониторинга // Вестник СГУ-ГиТ. – 2025. – Т. 30, № 3. – С. 5–14. – DOI 10.33764/2411-1759-2025-30-3-5-14.
4. Буслаев А.Б., Кошелева Н.Н., Белокопытов С.С. Устройство управления модулями MPU6050 по шине I²C на основе микроконтроллера // Вестник Воронежского государственного технического университета. – Т. 17. № 2. – 2021. – С. 65 – 69. DOI: 10.36622/VSTU.2021.17.2.010.
5. Шарапов А.А., Косарев Н.С., Яловой П.А., Шатц И.Д., Зиновьев Е.А. Web-приложение для визуализации результатов мониторинга смещений отвальных массивов // Свидетельство о регистрации программы для ЭВМ RU 2026618472, 25.03.2026.

© П. А. Яловой, Е. А. Зиновьев, И. Д. Шатц, Н. С. Косарев, А. А. Шарапов, 2026

СОДЕРЖАНИЕ

1. Л. Э. Мамонова, А. А. Шарапов. Проектирование и разработка пользовательского интерфейса веб-ресурса управления научными конференциями с применением фреймворка Vue.js и принципов UX/UI.....	3
2. И. В. Махорин. Метод адаптивной оконной фильтрации при расчете квантильного порога в задачах обнаружения инсайдерской активности	8
3. Г. М. Милькович, И. А. Дубров, А. А. Шарапов. Создание интерактивной образовательной веб-платформы проектирования и программирования робототехнических систем с применением искусственного интеллекта	15
4. В. В. Михайлюк, Е. Ю. Воронкин. Разработка Python-фреймворка для автоматизации релизных процессов с интеграцией в CI/CD-систему Jenkins ...	21
5. М. А. Мищенко, Н. С. Головачев. Разработка симулятора настройки сборочного конвейера для обучения программированию и автоматизации на виртуальной модели производственного участка.....	30
6. Д. Д. Модестова, Е. Ю. Воронкин. Модернизация веб-ресурса для учебно-научного центра «Планетарий» СГУГиТ	37
7. Р. В. Морозов, П. Ю. Бугаков. Сравнительный анализ методов обмена сообщениями в распределенной системе мониторинга сетевых узлов.....	45
8. Д. А. Остриков. Метод адаптивного контроля привилегированного доступа на основе мультипликативной агрегации рисков.....	56
9. Н. М. Острянин, Е. Ю. Воронкин. Разработка веб-сервиса для автоматизации и анализа продаж на основе ABC и XYZ-анализа для малого бизнеса.....	63
10. А. С. Панов, С. Ю. Кацко. Разработка веб-приложения для управления личными задачами с системой анализа и рекомендаций по повышению продуктивности	70
11. С. Е. Пелих, А. А. Шарапов. Разработка веб-приложения для поддержки образовательного процесса по робототехнике в центрах дополнительного образования	76
12. В. В. Пеньков, Е. Ю. Воронкин. Разработка веб-модуля для настройки ролей и разграничения доступа пользователей в рамках внутренней административной системы организации с применением технологий React и Node.js.....	82
13. Е. А. Плехоткин, Е. Ю. Воронкин. Разработка веб-ресурса «Дневник хроник моих интересов» на языке программирования Python.....	92

14. Д. И. Пономарев, Е. Ю. Воронкин. Разработка информационной среды учета товаров для компании-импортера на платформе «1С».....	100
15. В. С. Пустовит. Метод формирования виртуальных защищенных каналов связи с использованием российских криптографических алгоритмов	108
16. А. О. Ревазова. Метод интеллектуально-адаптивной оценки и фильтрации уязвимостей в системах статического анализа безопасности программного обеспечения	116
17. Б. С. Сагды, Н. С. Головачев. Разработка интерактивного наглядного пособия «Освоение Венеры» для учебно-научного центра «Планетарий» СГУГиТ	122
18. В. Ю. Сергеев, Д. Ю. Смирнов. Разработка программного обеспечения для определения уровня стресса сотрудников с применением технологий искусственного интеллекта	130
19. В. А. Сидун, С. Ю. Кацко. Разработка информационного веб-ресурса для ООО «ПСК “Стройкаркас”».....	135
20. Д. С. Строганов, С. Ю. Кацко. Мультиагентная система планирования группы БПЛА с нейросетевой оценкой конфигурации.....	143
21. А. Т. Ташханов, С. Ю. Кацко. Разработка веб-приложения для автоматизации выдачи микрозаймов с функцией кредитного скоринга.....	151
22. Н. С. Федотов, П. Ю. Бугаков. Концептуальная архитектура интеллектуального ассистента для работы с распределенными источниками данных	156
23. Н. С. Филина, Е. Ю. Воронкин. Разработка информационной системы для хранения и подготовки к печати документов демаркации российско-литовской государственной границы	165
24. Р. Р. Фиткулина, Н. С. Головачев. Разработка серверного модуля для поддержки сетевых функций игры «BlushBound».....	171
25. Ю. Н. Ханбекова. Метод обнаружения атак с предъявлением фотографий в биометрических системах.....	177
26. А. А. Цыганков. Анализ проблем в области выявления аномалий в действиях пользователей информационных систем на основе поведенческой аналитики	183
27. И. В. Чугунков, М. М. Ровнягин, К. А. Федоров, К. В. Сысоев. Архитектура адаптивного планировщика в геораспределенном кластере	189
28. Д. Р. Чуркин, Е. Ю. Воронкин. Разработка веб-приложения «Аптечка+» для учета домашних лекарственных средств и контроля срока их годности ...	196
29. Е. Ю. Шарапов, С. Ю. Кацко. Разработка игрового приложения с системой событий и управления ресурсами на платформе Godot	203

30. Т. С. Шевцова, Н. С. Головачев. Разработка программного средства анализа и проектирования озеленения жилых территорий на основе спутниковых данных и моделей застройки	208
31. В. В. Шнейдер, С. Ю. Кацко. Разработка рекомендательной информационной системы для поддержки абитуриентов при выборе специальности.....	214
32. В. В. Язовских, Р. В. Гришин. Веб-сервис транскрибации и анализа телефонных разговоров	220
33. Б. Ю. Якобсон. Алгоритм адаптивной фильтрации информационных потоков в корпоративных вычислительных сетях.....	226
34. Е. Н. Яковлев, А. А. Шарапов. Разработка системы аналитики и визуализации данных для веб-платформы управления научными конференциями	233
35. П. А. Яловой, Е. А. Зиновьев, И. Д. Шатц, Н. С. Косарев, А. А. Шарапов. Прототип автономного бюджетного датчика мониторинга деформаций бортов и отвальных массивов	239

CONTENTS

1. L. E. Mamonova, A. A. Sharapov. Design and development of a user interface for a scientific conference management web resource using the Vue.js framework and UX/UI principles.....	3
2. I. V. Makhorin. Adaptive window filtering method for quantile threshold estimation in insider threat detection	8
3. G. M. Milkovich, I. A. Dubrov, A. A. Sharapov. Development of an interactive educational web platform for the design and programming of robotic systems using artificial intelligence	15
4. V. V. Mikhaylyuk, E. Yu. Voronkin. Development of a python framework for release process automation with integration into the Jenkins CI/CD system	21
5. M. A. Mishchenko, N. S. Golovachev. Development of an assembly line setup simulator for programming and automation training on a virtual model of a production site	30
6. D. D. Modestova, E. Yu. Voronkin. Modernization of the web resource for the educational and scientific center "Planetarium" of SSUGT	37
7. R. V. Morozov, P. Yu. Bugakov. Comparative analysis of messaging methods in a distributed system for monitoring network nodes	45
8. D. A. Ostrikov. Adaptive privileged access control method based on multiplicative risk aggregation.....	56
9. N. M. Ostryanin, E. Yu. Voronkin. Development of a web service for sales automation and analysis based on ABC and XYZ analysis for small business.....	63
10. A. S. Panov, S. Yu. Katsko. Development of a web application for managing personal tasks with a system of analysis and recommendations for increasing productivity.....	70
11. S. E. Pelih, A. A. Sharapov. Development of a web application to support the educational process in robotics at additional education centers.....	76
12. V.V. Penkov, E. Y. Voronkin. Development of a web Module for configuring roles and user access control in an organization's internal administrative system using React and Node.js	82
13. E. A. Plehotkin, E. Yu. Voronkin. Development of the web resource «Diary of chronicles of my interests» in the Python programming language	92
14. D. I. Ponomarev, E. Yu. Voronkin. Development of an information environment for inventory accounting of an importing company using the 1C platform.....	100

15. V. S. Pustovit. Establishing secure virtual communication channels method using russian cryptographic algorithms.....	108
16. A. O. Revazova. A method for intelligent adaptive vulnerability assessment and filtering in static software security analysis systems.....	116
17. B. S. Sagdy, N. S. Golovachev. Development of an Interactive Visual Aid "Exploration of Venus" for the Educational and Scientific Center "Planetarium" of SSUGT	122
18. V. Yu. Sergeev, D. Yu. Smirnov. Developing software to determine employee stress levels using artificial intelligence technologies.....	130
19. V. A. Sidun, S. Yu. Katsko. Development of an information web resource for LLC «PSK “Stroykarkas”».....	135
20. D. S. Stroganov, S. Yu. Katsko. Multi-Agent system for UAV swarm planning with neural network-based configuration evaluation.....	143
21. A. T. Tashkhanov, S. Yu. Katsko. Development of a web application for microloan issuance automation with credit scoring function.....	151
22. N. S. Fedotov, P. Yu. Bugakov. Conceptual architecture of an intelligent assistant for distributed data source integration	156
23. N. S. Filina, E. Yu. Voronkin. Development of an information system for storing and preparing for printing documents of the russian-lithuanian state border demarcation	165
24. R. R. Fitkulina, N. S. Golovachev. Development of a server module for supporting network functions in the game «BlushBound»	171
25. Yu. N. Khanbekova. Method for detecting photo presentation attacks in biometric systems	177
26. A. A. Tsygankov. Analysis of problems in the field of detecting anomalies in user behavior within information systems based on behavioral analytics.....	183
27. I. V. Chugunkov, M. M. Rovniagin, K. A. Fedorov, K. V. Sysoev. Adaptive scheduler architecture in a geo-distributed cluster.....	189
28. D. R. Churkin, E. Yu. Voronkin. Development of a web application for home medication accounting	196
29. E. Y. Sharapovr, S. Yu. Katsko. Development of a game application with an event and resource management system on the Godot platform.....	203
30. T. S. Shevtsova, N. S. Golovachev. Development of a software tool for analysis and design of residential area landscaping based on satellite data and building models	208
31. V. V. Shneyder, S. Yu. Katsko. Development of a recommendation information system to support applicants in choosing a specialty	214

32. V. V. Yazovskikh, R. V. Grishin. Web service for transcription and analysis of telephone conversations	220
33. B. Y. Yakobson. Adaptive traffic filtering algorithm for corporate computer networks	226
34. E. N. Yakovlev, A. A. Sharapov. Development of an analytics and data visualization system for a web-based scientific conference management platform	233
35. P. A. Yalovoi, E. A. Zinovev, I. D. Shatts, N. S. Kosarev, A. A. Sharapov. Prototype of a low-cost sensor for monitoring displacements of waste dumps.....	239

Научное издание

ИНТЕРЭКСПО ГЕО-СИБИРЬ

XXII Международный научный конгресс

Сборник материалов в 9 т.

Т. 9

Международная научная конференция

«ПЕРСПЕКТИВНЫЕ РЕШЕНИЯ В СФЕРЕ ИНФОРМАЦИОННО-КОММУНИКАЦИОННЫХ ТЕХНОЛОГИЙ И КИБЕРБЕЗОПАСНОСТИ»

№ 2

Материалы публикуются в авторской редакции

Компьютерная верстка *А. Бочарникова*

Изд. лиц. ЛР № 020461 от 04.03.1997.

Подписано в печать 19.08.2026. Формат 60 × 84 1/16.

Усл. печ. л. 14,58. Тираж 25 экз. Заказ 147.

Гигиеническое заключение

№ 54.НК.05.953.П.000147.12.02. от 10.12.2002.

Издательско-полиграфический центр СГУГиТ

630108, Новосибирск, ул. Плахотного, 10.

Отпечатано в издательско-полиграфическом центре СГУГиТ

630108, Новосибирск, ул. Плахотного, 8