

Р. Ж. Гафиятов^{1✉}, Е. Ю. Воронкин¹

Разработка веб-приложения для организации отрядной деятельности и методической поддержки вожатых в детском оздоровительном лагере

¹Сибирский государственный университет геосистем и технологий,
г. Новосибирск, Российская Федерация
e-mail: gafrus06@mail.ru

Аннотация. В статье рассматривается процесс разработки веб-приложения для организации отрядной деятельности и методической поддержки вожатых в детском оздоровительном лагере. Актуальность работы обусловлена необходимостью создания единой цифровой платформы, объединяющей управление организационной информацией, методическими материалами и информационным сопровождением. В качестве технологической основы выбраны Java, Spring Boot, PostgreSQL, Redis, Apache Kafka и React. Разработанная система направлена на повышение эффективности работы вожатых и администрации лагеря, а также на улучшение информационного сопровождения родителей во время смены.

Ключевые слова: веб-программирование, Java, Spring Boot, PostgreSQL, Redis, HTML, CSS, JavaScript, React JS, Apache Kafka, Api Gateway, разработка веб-приложений, база данных

R. J. Gafiyatov^{1✉}, E. Yu. Voronkin¹

Development of a web application for organizing detachment activities and methodological support for counselors in a children's health camp

¹Siberian State University of Geosystems and Technologies, Novosibirsk, Russian Federation
e-mail: gafrus06@mail.ru

Abstract. This article examines the development of a web application for organizing team activities and providing methodological support to camp counselors at a children's health camp. The relevance of this work stems from the need to create a unified digital platform that integrates the management of organizational information, methodological materials, and information support. Java, Spring Boot, PostgreSQL, Redis, Apache Kafka, and React were selected as the technological foundation. The developed system is aimed at increasing the efficiency of camp counselors and administration, as well as improving information support for parents during camp.

Keywords: web programming, Java, Spring Boot, PostgreSQL, Redis, HTML, CSS, JavaScript, React JS, Apache Kafka, API Gateway, web application development, database

Введение

Лагерная смена представляет собой сложный социально-педагогический процесс, в котором участвуют администрация, вожатые, дети и родители. При этом значительная часть организационной работы на практике выполняется с использованием разрозненных инструментов: мессенджеров, бумажных планов, таблиц, локальных файлов и устных распоряжений. Такой подход усложняет координацию действий, снижает прозрачность управления сменой и повышает риск потери важной информации.

Особую роль в лагере выполняет вожатый, который ежедневно организует взаимодействие с детьми и участвует в реализации общего плана смены. Родители нуждаются в удобном канале получения сведений о жизни лагеря. Поэтому возникает необходимость в создании единой цифровой платформы, ориентированной на потребности детского оздоровительного лагеря.

Целью работы – разработка веб-приложения для организации отрядной деятельности и методической поддержки вожатых в детском оздоровительном лагере.

Методы и материалы

На этапе проектирования веб-приложения были использованы методы объектно-ориентированного анализа и UML-моделирования. Применение UML позволило формализовать структуру системы, определить основные роли пользователей и описать взаимодействие ключевых компонентов приложения [1].

В системе выделены четыре базовые роли: пользователь, родитель, вожатый и администратор. Дополнительно для роли вожатого предусмотрена внутренняя иерархия подролей: вожатый, старший вожатый и медик. Поэтому сначала была построена диаграмма иерархии акторов, отражающая наследование прав и различия между категориями пользователей (рис. 1).

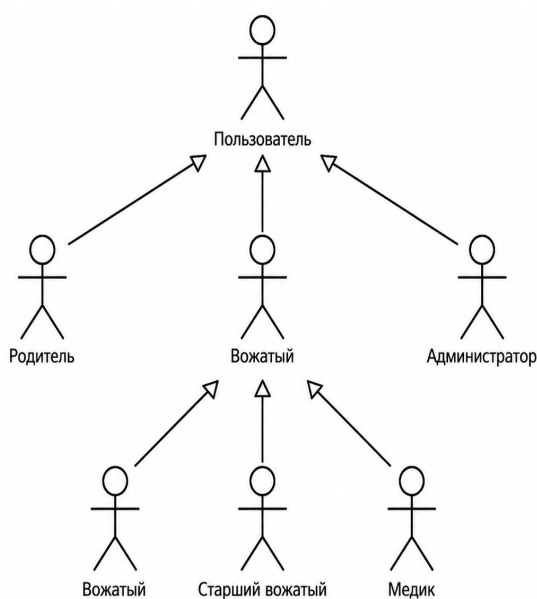


Рис. 1. Диаграмма иерархии

Архитектура веб-приложения построена на основе микросервисного подхода. Для отображения общей структуры системы была разработана схема функционирования веб-приложения, которая показывает взаимодействие клиентской части, API Gateway, микросервисов, баз данных и инфраструктурных компонентов. Такой подход позволяет обеспечить возможность дальнейшего масштабирования.

В качестве клиентской части используется React, который отвечает за отображение пользовательского интерфейса и взаимодействие пользователя с системой [3]. Все запросы от клиента поступают в API Gateway, реализованный на базе Spring Cloud Gateway. Шлюз является единой точкой входа в систему, выполняет маршрутизацию запросов к микросервисам, проверяет JWT-токены и передаёт пользовательские данные во внутренние сервисы [4].

Серверная часть приложения реализована в виде набора микросервисов. auth-service отвечает за регистрацию, аутентификацию и авторизацию пользователей. user-service хранит и обрабатывает профили пользователей. camp-service реализует основную бизнес-логику, связанную с лагерями, сменами, отрядами, детьми, заявками и назначениями сотрудников. news-feed-service отвечает за публикации, новости, медиафайлы и модерацию постов. file-storage-service обеспечивает загрузку и получение файлов через S3-совместимое хранилище. notification-service отвечает за уведомления пользователей.

В качестве основной системы хранения данных используется PostgreSQL. Каждый микросервис имеет собственную базу данных, что снижает связанность компонентов и позволяет независимо развивать отдельные части системы [5]. Для кэширования и проверки актуальности JWT-токенов используется Redis. Асинхронное взаимодействие между сервисами реализовано с помощью Apache Kafka, которая используется для передачи событий, например при регистрации пользователя или создании уведомлений [6].

Файлы и медиаобъекты хранятся в VK Cloud S3 хранилище, а в базе данных сохраняются только их метаданные: идентификатор, ссылка, тип, размер, владелец и связь с публикацией или другим объектом системы.

Разделение клиентской и серверной частей позволяет поддерживать независимость микросервисов, централизованную маршрутизацию запросов, событийное взаимодействие между сервисами и возможность дальнейшего масштабирования системы.

Результаты

В результате работы создано веб-приложение «Loom / FunKids» для организации отрядной деятельности и методической поддержки вожатых в детском оздоровительном лагере. Приложение реализует основные пользовательские сценарии для родителя, вожатого и администратора, а также обеспечивает работу с новостной лентой, карточками детей, заявками и отрядами.

В разработанном веб-приложении реализованы следующие функции:

- авторизация/регистрация пользователей с проверкой введённых данных и последующим переходом в систему;
- отображение новостной ленты лагеря с публикациями, медиафайлами и информацией для пользователей (рис. 2);
- управление отрядом, включая просмотр состава, работу с детьми и доступ к основным действиям вожатого (рис. 3);

- заполнение родителем формы заявки ребёнка на участие в смене с указанием персональных и контактных данных (рис. 4);
- просмотр карточки ребёнка с основной информацией, данными родителя и сведениями, необходимыми для работы сотрудников лагеря;
- разделение доступа к функциям приложения в зависимости от роли пользователя: родитель, вожатый или администратор;
- получение пользователями уведомлений о важных событиях в системе.

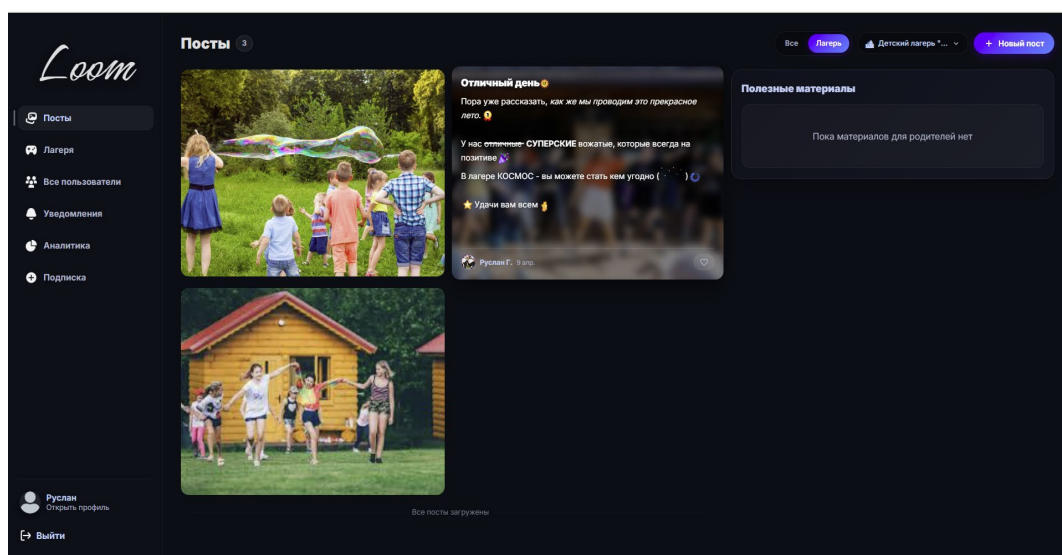


Рис. 2. Интерфейс страницы ленты новостей

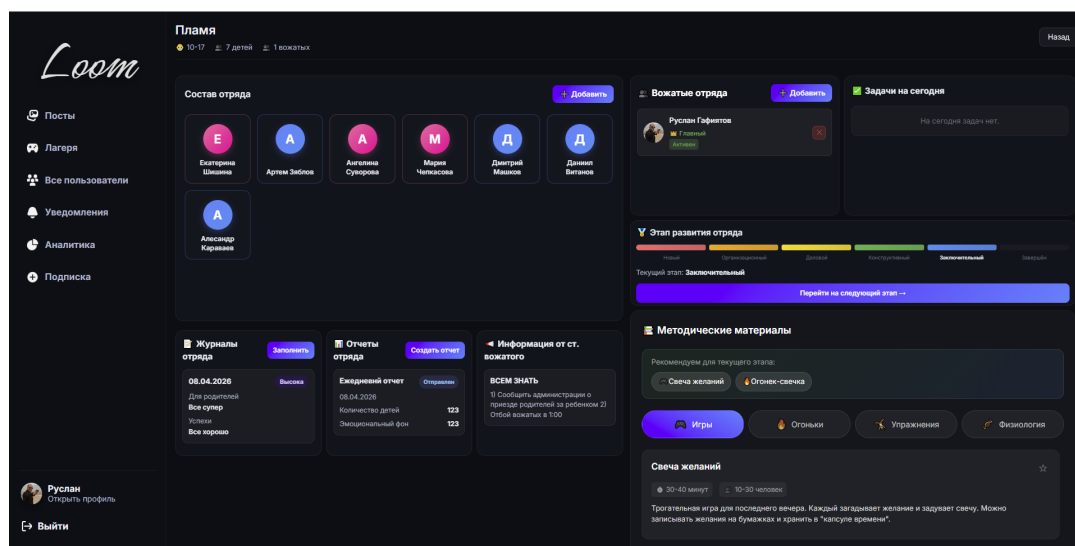


Рис. 3. Интерфейс страницы управления отрядом

Анкета ребёнка

Заполните информацию о ребёнке. Вожатый смены увидит анкету и примет решение.

Имя *

Фамилия *

Дата рождения *

Пол

Город

Кем приходиться

Аллергии

Медицинские особенности

Особые потребности

Рис. 4. Интерфейс формы заявки

Нагрузочное тестирование k6 проводилось в 4 режимах: 50 пользователей в течение 6 минут, 50 пользователей в течение 31 минуты, плавный рост до 200 пользователей за 14,5 минуты и резкий скачок до 200 пользователей за 5 минут. В первых двух сценариях система работала стабильно: доля ошибок составила 2,3% и 1,9%, среднее время ответа – 399 мс и 342 мс, p95 – 1401 мс и 1223 мс. При росте до 200 пользователей ошибки увеличились до 6,8%, а при резком скачке – до 14,2%, при этом p95 остался в допустимых пределах: 2100 мс и 2400 мс. Основные ошибки возникали в news-feed-service, так как при загрузке ленты сервис обращается к другим компонентам системы. Для решения проблемы были добавлены кэширование, постраничная загрузка, сокращение межсервисных вызовов и увеличенные таймауты для операций с медиа.

Заключение

В статье рассмотрен процесс разработки веб-приложения для организации отрядной деятельности и методической поддержки вожатых в детском оздоровительном лагере. Целью проекта стало создание единой цифровой платформы, позволяющей объединить управление лагерем, сменами, отрядами, новостями, методическими материалами и взаимодействием с родителями. В отличие от разрозненного использования мессенджеров, таблиц, бумажных планов и локальных файлов, разработанная система предоставляет централизованный доступ к основным данным и инструментам лагерной смены.

Разработка велась с использованием Java, Spring Boot, PostgreSQL, Redis, Apache Kafka и React. В ходе работы были спроектированы и реализованы ключевые функциональные модули: авторизация пользователей, управление профилями, лагерями, сменами и отрядами, новостная лента, хранение файлов и уведомления. Особое внимание было уделено разграничению прав доступа между пользователями, родителями, вожатыми и администраторами, а также поддержке ролей внутри лагеря.

Итогом работы стало веб-приложение, которое позволяет повысить эффективность работы вожатых и администрации, упростить организацию отрядной деятельности и улучшить информационное сопровождение родителей во время лагерной смены. Разработанная система может быть использована как основа для дальнейшего развития цифровой инфраструктуры детского оздоровительного лагеря.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Использование диаграммы вариантов использования UML при проектировании программного обеспечения // Хабр URL: <https://habr.com/ru/articles/566218/> (дата обращения: 15.05.2026).
2. Диаграмма последовательности (sequence-диаграмма) // Хабр URL: <https://habr.com/ru/articles/814769/> (дата обращения: 15.05.2026).
3. ER-модель // Википедия URL: <https://ru.wikipedia.org/wiki/ER-модель> (дата обращения: 15.05.2026).
4. Диаграмма компонентов // Википедия URL: https://ru.wikipedia.org/wiki/Диаграмма_компонентов (дата обращения: 15.05.2026).
5. PostgreSQL Documentation // The PostgreSQL Global Development Group : official documentation. – URL: <https://www.postgresql.org/docs/> – Дата обращения: 13.05.2026.
6. React Documentation // React : official documentation. – URL: <https://react.dev/> – Дата обращения: 13.05.2026.
7. Apache Kafka Documentation // Apache Software Foundation : official documentation. – URL: <https://kafka.apache.org/documentation/> – Дата обращения: 13.05.2026.
8. Spring Cloud Gateway Reference Documentation // Spring : official documentation. – URL: <https://docs.spring.io/spring-cloud-gateway/reference/index.html> – Дата обращения: 13.05.2026.

© Р. Ж. Гафиятов, Е. Ю. Воронкин, 2026