

И. А. Зыков^{1✉}, Е. Ю. Воронкин¹

Разработка веб-приложения для автоматизации клиентских коммуникаций с поддержкой интеграции во внешние сервисы

¹Сибирский государственный университет геосистем и технологий,
г. Новосибирск, Российская Федерация
e-mail: heler.2004@gmail.com

Аннотация. В статье представлены основные результаты разработки веб-приложения для автоматизации клиентских коммуникаций с поддержкой интеграции во внешние сервисы. Рассмотрена проблема фрагментарного хранения клиентских контактов, задач и договорённостей в мессенджерах, таблицах и заметках. Для решения данной проблемы была разработана CRM-платформа, объединяющая веб-интерфейс, базу данных, Telegram-бота, VK-бота и модуль автоматических напоминаний. В качестве основного стека технологий использованы Python, FastAPI, Jinja2, SQLAlchemy, SQLite, Aiogram, VKBottle и APScheduler. Разработанное приложение позволяет создавать и редактировать лидов, изменять статусы обработки, добавлять задачи с дедлайнами, получать напоминания о просроченных задачах, а также добавлять лидов через внешние сервисы на примере Telegram и VK.

Ключевые слова: CRM-система, веб-приложение, клиентские коммуникации, FastAPI, Telegram-бот, VK-бот, SQLite, автоматизация

I. A. Zykov^{1✉}, E. Yu. Voronkin¹

Development of a web application for automating customer communications with support for integration into external services

¹Siberian State University of Geosystems and Technologies, Novosibirsk, Russian Federation
e-mail: heler.2004@gmail.com

Abstract. The article presents the main results of the development of a web application for automating customer communications with support for integration into external services. The problem of fragmented storage of customer contacts, tasks and agreements in messengers, spreadsheets and notes is considered. To solve this problem, a CRM platform was developed, combining a web interface, a database, a Telegram bot, a VK bot and an automatic reminder module. Python, FastAPI, Jinja2, SQLAlchemy, SQLite, Aiogram, VKBottle and APScheduler were used as the main technology stack. The developed application allows users to create and edit leads, change processing statuses, add tasks with deadlines, receive reminders about overdue tasks, and add leads through external services using Telegram and VK as examples.

Keywords: CRM system, web application, customer communications, FastAPI, Telegram bot, VK bot, SQLite, automation

Введение

В современных условиях значительная часть взаимодействия с потенциальными клиентами осуществляется через цифровые каналы связи: мессенджеры, социальные сети, электронную почту и веб-формы. Такой формат коммуникации удобен для быстрого контакта, однако при отсутствии централизованной информационной

системы приводит к фрагментации данных. Контакты клиентов, договорённости, заметки, задачи и сроки выполнения могут храниться в разных чатах, личных сообщениях, таблицах или текстовых документах. В результате повышается риск потери информации, дублирования записей и пропуска важных действий по клиенту.

Для организаций и специалистов, работающих с входящими заявками, особенно важны оперативная фиксация лида, контроль его текущего статуса и своевременное выполнение последующих задач. Использование только таблиц или заметок не обеспечивает достаточного уровня связности данных, а внедрение крупных CRM-систем может быть избыточным для небольшой компании. В связи с этим актуальной становится разработка прикладной CRM-платформы, ориентированной на базовые сценарии работы с клиентскими коммуникациями и поддерживающей интеграцию с внешними сервисами.

Целью работы является разработка веб-приложения для автоматизации клиентских коммуникаций с поддержкой интеграции во внешние сервисы. В рамках проекта была реализована CRM-платформа, позволяющая вести список лидов, работать с карточками клиентов, назначать задачи, контролировать дедлайны и получать напоминания. В качестве внешних сервисов были рассмотрены мессенджеры Telegram и VK, через которые пользователь может быстро добавить нового лида в систему.

Исследование предметной области

CRM-системы используются для организации работы с клиентами, фиксации обращений, ведения истории взаимодействий, контроля статусов и повышения управляемости процессов продаж и сопровождения [1]. Основная задача таких систем заключается в том, чтобы объединить разрозненные сведения о клиентах в едином информационном пространстве. Это особенно важно в тех случаях, когда коммуникация ведётся через несколько каналов и менеджеру необходимо быстро понимать, на каком этапе находится каждый клиент.

На практике можно выделить несколько распространённых подходов к ведению клиентских коммуникаций. Первый подход заключается в использовании таблиц, заметок и отдельных текстовых файлов. Его преимущество состоит в простоте начала работы и отсутствии необходимости внедрения специализированного программного обеспечения. Однако такой способ плохо масштабируется, не обеспечивает полноценного контроля задач и не позволяет удобно отслеживать статус каждого лида.

Второй подход связан с ведением коммуникации непосредственно в мессенджерах. Telegram, VK и другие сервисы позволяют быстро получать сообщения от клиентов и оперативно отвечать на них. Однако мессенджер сам по себе не является CRM-системой: в нём сложно структурировать лидов, назначать статусы, контролировать дедлайны и формировать общую картину работы с клиентской базой.

Третий подход заключается во внедрении готовых CRM-систем. Такие решения обладают широким функционалом, но могут быть избыточными для небольших проектов, учебных задач или MVP. Кроме того, их настройка часто

требует времени, обучения пользователей и адаптации бизнес-процессов под конкретную платформу.

С учётом рассмотренных ограничений было принято решение разработать собственную CRM-платформу в формате MVP. Такой подход позволяет реализовать только необходимые функции: создание лида, хранение контактных данных, изменение статуса, добавление задач, контроль дедлайнов и интеграцию с внешними сервисами. При этом система остаётся достаточно простой для пользователя и может быть расширена в дальнейшем.

Методы и технологии

Разработка CRM-платформы была выполнена с использованием языка программирования Python. Данный язык был выбран благодаря широкому набору библиотек для веб-разработки, работы с базами данных, создания ботов и планирования фоновых задач. Такой подход позволил реализовать основные компоненты системы в едином технологическом стеке.

В качестве веб-фреймворка был выбран FastAPI. Он используется для создания серверной части приложения, обработки HTTP-запросов, маршрутизации и взаимодействия с шаблонами интерфейса. FastAPI хорошо подходит для разработки прикладных веб-сервисов, так как поддерживает современный подход к описанию маршрутов и позволяет быстро создавать структуру серверного приложения [3].

Для формирования пользовательского интерфейса применялись Jinja2, HTML, CSS и JavaScript. Веб-интерфейс CRM предоставляет пользователю доступ к списку лидов, карточкам клиентов, формам создания и редактирования данных, а также к управлению задачами. Такой вариант реализации был выбран из-за простоты и достаточности для MVP-версии системы.

Для хранения данных использовалась база данных SQLite. Она подходит для локального прототипа, так как не требует отдельной серверной установки и позволяет быстро развернуть приложение. В базе данных хранятся сведения о пользователях, лидах и задачах. При дальнейшем развитии проекта возможен переход на PostgreSQL, что позволит повысить устойчивость и масштабируемость системы [4].

Работа с базой данных реализована через SQLAlchemy. Данная библиотека используется как ORM-инструмент, позволяющий описывать модели данных в виде классов Python и выполнять операции создания, чтения, обновления и удаления записей [5]. В проекте были выделены основные сущности: пользователь, лид и задача. Пользователь может иметь несколько лидов, а каждый лид может быть связан с несколькими задачами.

Для интеграции с Telegram был использован фреймворк Aiogram. Он позволяет создавать асинхронных Telegram-ботов и обрабатывать пользовательские команды, текстовые сообщения и контакты [6]. В рамках проекта Telegram-бот реализует сценарий быстрого добавления лида: пользователь вводит имя, username и заметку, после чего данные сохраняются в CRM.

Для интеграции с VK был использован VKBottle. Данный инструмент предназначен для создания ботов и работы с VK API [7]. VK-бот реализует похожий сценарий: пользователь запускает диалог, вводит данные лида, после чего запись сохраняется в общей базе данных и становится доступной в веб-интерфейсе CRM.

Для автоматической проверки просроченных задач применялся APScheduler. Планировщик периодически запускает фоновую задачу, которая проверяет дедлайны и отправляет пользователю напоминания в Telegram при наличии просроченных задач [8]. Это позволяет автоматизировать контроль сроков и снизить риск пропуска важных действий по клиенту.

Таким образом, для разработки CRM-платформы был использован следующий стек технологий:

- язык программирования Python;
- веб-фреймворк FastAPI;
- шаблонизатор Jinja2;
- HTML, CSS и JavaScript для пользовательского интерфейса;
- SQLite для хранения данных MVP-версии;
- SQLAlchemy для работы с базой данных;
- Aiogram для интеграции с Telegram Bot API;
- VKBottle для интеграции с VK API;
- APScheduler для планирования автоматических напоминаний.

Результаты

В результате работы было разработано веб-приложение для автоматизации клиентских коммуникаций с поддержкой интеграции во внешние сервисы. Система реализована как CRM-платформа, объединяющая веб-интерфейс, базу данных, Telegram-бота, VK-бота и механизм автоматических напоминаний.

В базе данных реализованы три основные сущности: users, leads и tasks. Сущность users хранит информацию о пользователях системы, включая идентификаторы Telegram и VK. Сущность leads содержит карточки лидов: имя, username, статус, заметку, дату создания и дату обновления. Сущность tasks используется для хранения задач, связанных с лидами: название задачи, срок выполнения, состояние выполнения и время последнего напоминания.

В веб-интерфейсе реализованы следующие функции:

- авторизация пользователя через внешние сервисы;
- просмотр списка лидов;
- создание нового лида;
- просмотр карточки лида;
- редактирование данных лида;
- изменение статуса обработки лида;
- создание задачи по конкретному лиду;
- отметка задачи как выполненной;
- удаление задач и лидов;
- просмотр информации о просроченных задачах.

Карточка лида является одним из основных элементов CRM. Она объединяет сведения о клиенте, текущий статус обработки, заметку и список связанных задач. Такой подход позволяет менеджеру быстро оценить текущее состояние работы с конкретным клиентом и понять, какое действие необходимо выполнить следующим.

Отдельное внимание было уделено интеграции с мессенджерами. Telegram-бот позволяет пользователю добавить лида без входа в веб-интерфейс. Для этого реализован пошаговый сценарий: пользователь вводит имя лида, username и заметку. После завершения сценария бот сохраняет запись в базу данных и отправляет пользователю сообщение с подтверждением и ссылкой на карточку лида в CRM.

VK-бот реализует аналогичный механизм добавления лида через сообщения сообщества. Пользователь может выбрать действие с помощью кнопок, ввести имя, ссылку или username, а также заметку. После этого информация сохраняется в общей базе данных. Это показывает возможность расширения CRM-платформы на разные внешние каналы коммуникации.

Модуль напоминаний предназначен для контроля просроченных задач. Система периодически проверяет задачи, срок выполнения которых уже прошёл, но которые ещё не были отмечены как выполненные. Если такие задачи обнаружены, пользователь получает уведомление в Telegram. В сообщении отображается список просроченных задач, связанный лид и информация о времени просрочки. Это помогает менеджеру своевременно возвращаться к клиентским действиям и снижает риск потери лида.

Для проверки работоспособности системы был разработан план тестирования. В качестве основного метода использовалось функциональное тестирование методом «чёрного ящика». Данный метод был выбран потому, что проверка выполнялась с позиции пользователя: осуществлялся ввод данных через веб-интерфейс и ботов, после чего ожидаемый результат сравнивался с фактическим. Были проверены сценарии авторизации, создания лида, редактирования карточки, изменения статуса, создания задачи, выполнения задачи и получения напоминания.

Результаты тестирования показали, что основные функции MVP-версии работают корректно. Создание лидов через веб-интерфейс, Telegram-бота и VK-бота приводит к сохранению данных в базе. Изменение статусов и редактирование карточек выполняются корректно. Задачи привязываются к соответствующим лидам, а просроченные задачи отображаются и используются модулем напоминаний.

Разработанная система является MVP-версией CRM-платформы. Она демонстрирует основной принцип автоматизации клиентских коммуникаций: данные о клиенте можно быстро зафиксировать через внешний сервис, а дальнейшая обработка выполняется в веб-интерфейсе CRM. Такой подход объединяет удобство мессенджеров и структурированность CRM-системы.

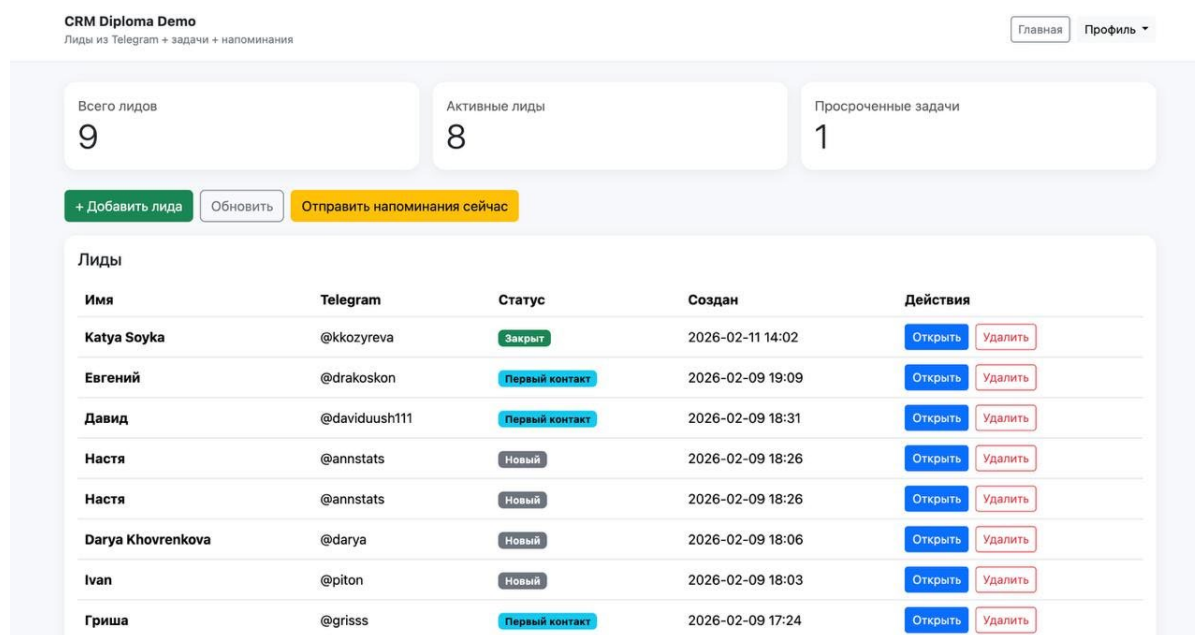


Рис. 1. Интерфейс приложения

Заключение

В результате выполнения работы было разработано веб-приложение для автоматизации клиентских коммуникаций с поддержкой интеграции во внешние сервисы. Реализованная CRM-платформа позволяет создавать и редактировать лидов, изменять статусы обработки, назначать задачи, контролировать дедлайны и получать напоминания о просроченных действиях.

В рамках проекта были решены основные задачи: проведён анализ предметной области, определены требования к системе, спроектирована структура данных, реализован веб-интерфейс CRM, разработаны Telegram-бот и VK-бот, создан модуль автоматических напоминаний и проведено тестирование основных пользовательских сценариев.

Практическая значимость работы заключается в том, что разработанная система может использоваться как основа для дальнейшего развития CRM-платформы. В перспективе возможно расширение функциональности за счёт перехода с SQLite на PostgreSQL, добавления ролей пользователей, реализации истории коммуникаций, аналитики по лидам, резервного копирования и повышения безопасности. Также одним из направлений дальнейшего развития является интеграция с мессенджером MAX, что позволит расширить количество внешних каналов взаимодействия с клиентами.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Buttle, F. Customer Relationship Management: Concepts and Technologies / F. Buttle, S. Maklan. – 4th ed. – London : Routledge, 2019. – 444 p.
2. Sommerville, I. Software Engineering / I. Sommerville. – 10th ed. – Boston : Pearson, 2015. – 816 p.

3. FastAPI : official documentation. – URL: <https://fastapi.tiangolo.com/> (дата обращения: 13.05.2026). – Текст : электронный.
4. SQLite Documentation. – URL: <https://sqlite.org/docs.html> (дата обращения: 13.05.2026). – Текст : электронный.
5. SQLAlchemy Documentation. – URL: <https://docs.sqlalchemy.org/> (дата обращения: 13.05.2026). – Текст : электронный.
6. Aiogram documentation. – URL: <https://docs.aiogram.dev/> (дата обращения: 13.05.2026). – Текст : электронный.
7. VKBottle Documentation. – URL: <https://vkbottle.readthedocs.io/ru/latest/> (дата обращения: 13.05.2026). – Текст : электронный.
8. Advanced Python Scheduler : documentation. – URL: <https://apscheduler.readthedocs.io/> (дата обращения: 13.05.2026). – Текст : электронный.

© И. А. Зыков, Е. Ю. Воронкин, 2026