

МИНОБРНАУКИ РОССИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

Факультет Компьютерных наук
Кафедра программирования и информационных технологий

Автоответчик для почты и мессенджеров «DialogX» с использованием ИИ

Курсовая работа
Направление: 09.03.04. Программная инженерия

Зав. Кафедрой _____ д. ф.-м. н, доцент С.Д. Махортов
Руководитель _____ ст. преподаватель В.С. Тарасов
Руководитель практики _____ ассистент, Г.В. Прядченко
Обучающийся _____ Н. Е. Флейшман, 3 курс, д/о
Обучающийся _____ Д. Е. Баранник, 3 курс, д/о
Обучающийся _____ И. П. Сидоренко, 3 курс, д/о
Обучающийся _____ В. А. Гончаров, 3 курс, д/о
Обучающийся _____ И. И. Давыдов, 3 курс, д/о
Обучающийся _____ В. А. Кузнецов, 3 курс, д/о

Воронеж 2025

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	4
1 Постановка задачи.....	5
1.1 Требования к разрабатываемой системе	5
1.1.1 Функциональные требования	5
1.1.2 Нефункциональные требования	6
2 Анализ предметной области	8
2.1 Терминология (гlossарий) предметной области	8
2.2 Обзор аналогов	10
2.2.1 i2CRM.....	10
2.2.2 Angry.Space.....	12
2.2.3 Pact.....	14
2.2.4 Quickley	16
2.2.5 Вывод на основе анализа конкурентов.....	18
3 Реализация.....	19
3.1 Средства реализации.....	19
3.2 Разработка frontend (клиентской части веб-приложения)	21
3.3 Разработка backend (серверной части веб-приложения).....	23
3.4 Основная функциональность сайта.....	25
3.4.1 Для неавторизованных пользователей	25
3.4.2 Для авторизованных пользователей	29
3.4.3 Для оператора.....	31
3.4.4 Для администратора	34
4 Тестирование	38
4.1 UI-тестирование	38
4.2 Тестирование API.....	40
4.3 Модульное (Unit) тестирование серверной части	41

5 Анализ пользовательского интерфейса	44
ЗАКЛЮЧЕНИЕ	46
ПРИЛОЖЕНИЕ А	48

ВВЕДЕНИЕ

В современном мире коммуникация между бизнесом и клиентом все чаще осуществляется через онлайн-каналы, а именно – мессенджеры, электронную почту, чаты на сайтах. На смену традиционным методам общения с клиентом приходят многофункциональные веб-платформы, способные автоматически принимать и обрабатывать входящие сообщения. Такие решения позволяют компаниям оперативно взаимодействовать с клиентами, не терять потенциальные обращения и поддерживать высокий уровень сервиса даже в нерабочее время.

С каждым годом возрастает потребность в автоматизации общения с клиентами, особенно со стороны малого и среднего бизнеса. Ручная обработка сообщений требует значительных трудовых и финансовых затрат, порой приводит к потере клиентов из-за долгого ожидания ответа. В таких условиях растет потребность в технологиях, которые позволяют централизовать коммуникации, снижать нагрузку на операторов, уменьшать время ожидания ответа.

Система автоответчика становятся важным инструментом оптимизации процессов в бизнесе. Она объединяет обращения из разных источников, предоставляет единый интерфейс для операторов, позволяет настраивать автоматические ответы, а также вести статистику. Особенно востребованными становятся решения, поддерживающие интеграции с наиболее популярными сервисами, такими как: WhatsApp, Telegram, VK, email.

Цель данной курсовой работы – разработка веб-приложения, представляющего собой автоответчик для почты и мессенджеров. Система будет предоставлять пользователям возможность подключения и управления интеграциями, настройки шаблонных ответов, просмотра диалогов, статистики, создания сделок, а также использования AI-ответов.

1 Постановка задачи

Целью данной курсовой работы является разработка веб-приложения «DialogX» – автоответчика для мессенджеров и почты, предназначенного для автоматизации общения с пользователями. Разработка ориентирована как на нужды малого и среднего бизнеса, так и частных пользователей. Основными задачами являются: сокращение времени отклика, снижение нагрузки на операторов и повышение эффективности взаимодействия с клиентами.

Сайт решает ряд следующих задач:

- регистрация и авторизация пользователей;
- подключение и настройка интеграций с email, Telegram, WhatsApp, VK;
- добавление виджета диалога нашего автоответчика на сторонний сайт;
- добавление, изменение и удаление шаблонных ответов;
- ведение диалогов с клиентами из разных каналов через единую платформу;
- создание сделок и изменение их статусов;
- сбор и отображение статистики по мессенджерам и операторам;
- управление подпиской на сервис.

Приложение предоставляет функционал для следующих категорий пользователей: неавторизованных пользователей, авторизованных, операторов и администраторов.

1.1 Требования к разрабатываемой системе

1.1.1 Функциональные требования

К разрабатываемому веб-приложению предъявляются функциональные требования, которые различаются в зависимости от роли пользователя.

Для неавторизованного пользователя выделяются следующие функциональные требования:

- регистрация и авторизация в системе;
- смена пароля от учетной записи;
- просмотр ознакомительной информации о сайте.

Для авторизованного пользователя выделяются следующие функциональные требования:

- редактирование и просмотр данных своего профиля;
- просмотр информации о подписке и доступных тарифных планах;
- оформление подписки;
- выход из учетной записи.

Для оператора поддерживается весь функционал авторизованного пользователя, а также:

- просмотр списка диалогов;
- чтение и отправка сообщений клиентам;
- создание и редактирование сделок по диалогам;
- доступ к статистике.

Для администратора поддерживается весь функционал оператора, а также:

- управление подпиской (оформление, изменение);
- добавление и удаление операторов;
- добавление, редактирование и удаление шаблонных ответов;
- подключение интеграций с мессенджерами (Telegram, VK, WhatsApp);
- подключение интеграций с электронной почтой (@yandex.ru, @gmail.com);
- подключение интеграции виджета диалога на сторонний сайт.

1.1.2 Нефункциональные требования

Web-сайт должен выполнять следующие нефункциональные требования:

- производительность: основные страницы загружаются за 1 – 5 секунды при стабильном интернете (от 50мб/с), при нагрузке до 100 пользователей приложение должно оставаться работоспособным;
- безопасность: данные пользователей должны быть защищены от несанкционированного доступа, будет использоваться защищенный тип соединения HTTPS, а также Spring Security;
- масштабируемость: система должна быть горизонтально масштабируема для работы с большим числом пользователей, с использованием контейнеризации (Docker), а также обеспечивать возможность добавления нового функционала с минимальным изменением существующего кода;
- надежность: приложение должно стабильно работать в типовых условиях, ошибки фиксируются в логах, возможен автоматический перезапуск через Docker;
- совместимость с различными браузерами: сайт должен быть доступен для использования на большинстве популярных браузеров (Яндекс, Гугл, Опера);
- адаптивный дизайн: интерфейс сайта должен корректно отображаться при масштабировании от 75% до 150% (уровень масштабирования в браузере).

2 Анализ предметной области

2.1 Терминология (гlossарий) предметной области

Авторизация – предоставление определённому лицу прав на выполнение определённых действий; а также процесс проверки (подтверждения) данных прав при попытке выполнения этих действий.

Авторизованный пользователь – пользователь, который прошел процесс авторизации.

Администратор – пользователь с максимальными правами доступа, который управляет подписками, шаблонными ответами, интеграциями и операторами.

Неавторизованный пользователь – пользователь, который не прошел процесс авторизации.

Оператор – пользователь, взаимодействующий с клиентами через чат, обрабатывающий диалоги и создающий сделки.

Интеграция – подключение внешних сервисов (мессенджеров, почтовых систем) для автоматизации обработки входящих сообщений и управления коммуникации с клиентами, передача данных происходит через Rest API

Пользователь – лицо, которое использует действующую систему для выполнения конкретной функции.

Рабочая область – это основной пользовательский интерфейс внутри веб-приложения, в котором пользователь (например, оператор или администратор) выполняет ключевые действия: обработку диалогов, работу с CRM, просмотр статистики, настройку интеграций и управление шаблонами, рабочая область отображается после авторизации.

Сделка – запись о взаимодействии с клиентом, включающая этапы, сумму и другие детали процесса.

СУБД – комплекс программ, позволяющих создать базу данных и манипулировать данными (вставлять, обновлять, удалять и выбирать).

Фреймворк – набор инструментов, компонентов и методов, которые облегчают разработку программного обеспечения.

Чат – средство обмена различной информацией по компьютерной сети в режиме реального времени, а также программное обеспечение, позволяющее организовывать такое общение.

Шаблонный ответ – заранее подготовленное сообщение, используемое для ответа на сообщения.

AI-ответ – автоматически сгенерированный ответ на основе искусственного интеллекта, предназначенный для ответа на сообщения.

API – программный интерфейс, то есть описание способов взаимодействия одной компьютерной программы с другими программами.

Frontend – часть веб-приложения, которая обеспечивает отображение информации и взаимодействие с пользователем.

PostgreSQL – свободная объектно-реляционная система управления базами данных.

REST – архитектурный стиль взаимодействия компонентов распределенного приложения в сети.

Backend – внутренняя часть сайта или приложения, которая находится на сервере и отвечает за бизнес-логику, обработку данных, и взаимодействие с базами данных или другими внешними системами.

2.2 Обзор аналогов

Прежде чем приступить к разработке веб-приложения автоответчика, было проведено исследование существующих решений в данной области. Цель анализа – выявить преимущества и недостатки конкурентов, понять их функциональные особенности. И на основе выделенных преимуществ и недостатков переходить к разработке собственного приложения.

2.2.1 i2CRM

Сервис i2CRM представляет собой мощную платформу для интеграции мессенджеров и CRM-систем. Он предлагает централизованное управление переписками с клиентами из различных каналов, включая WhatsApp, Telegram, VK и Instagram, с последующей отправкой данных в CRM.

На практике система позволяет сохранять всю историю общения, настраивать автоматические сообщения и использовать шаблоны для ускорения коммуникации. Поддерживаются интеграции с amoCRM, Битрикс24.

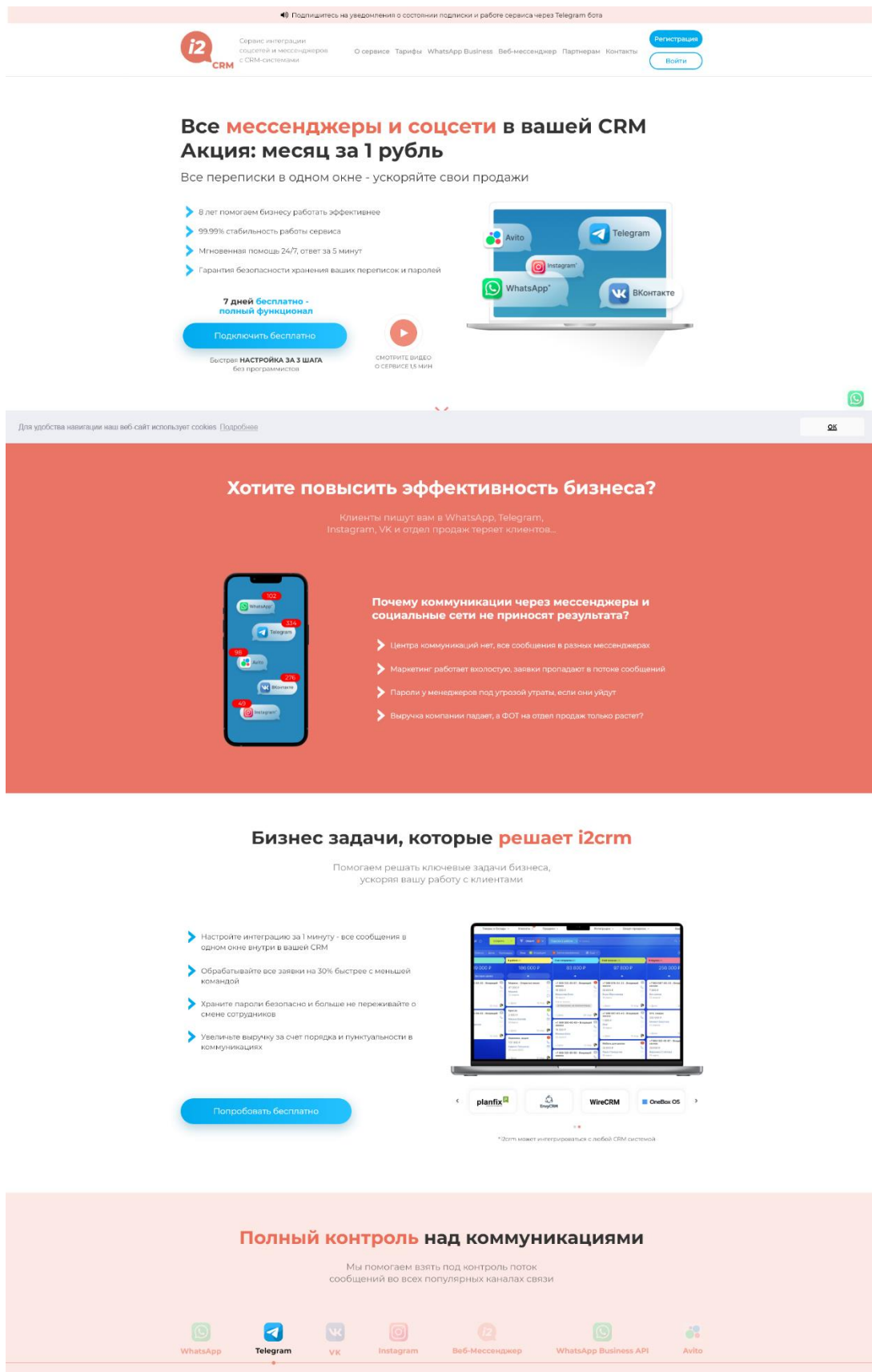


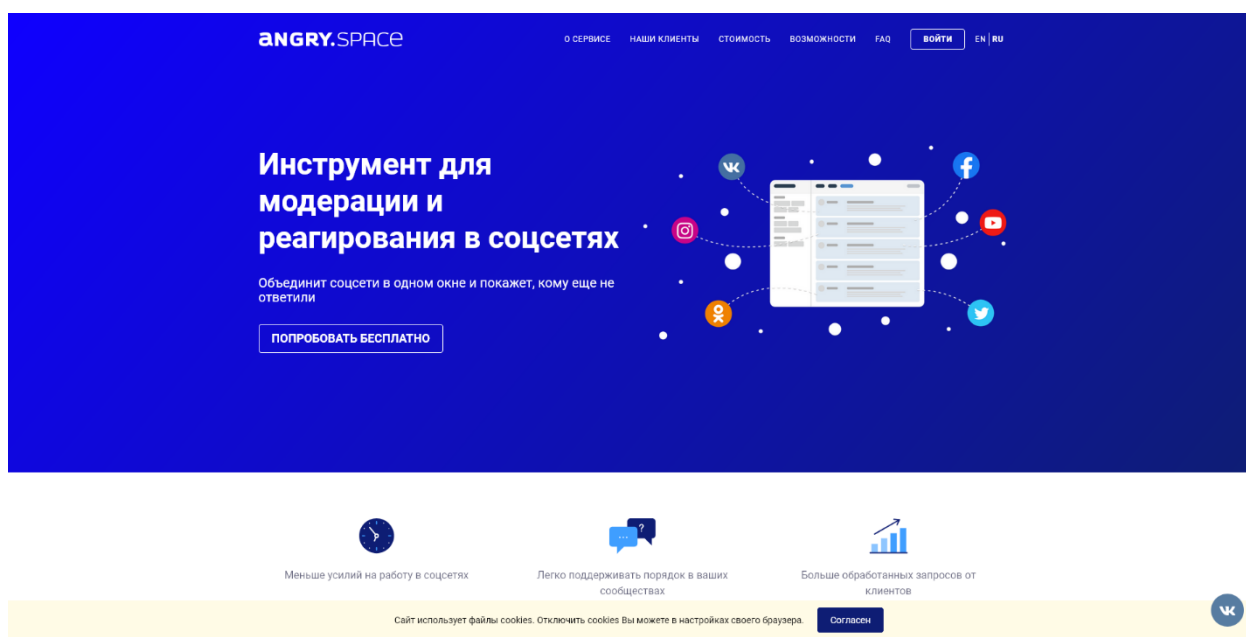
Рисунок 1 - Главная страница сайта «i2CRM»

Преимуществами сервиса являются высокая скорость доставки сообщений в CRM, гибкость в настройке интеграций и поддержка корпоративного доступа для нескольких сотрудников. Однако к недостаткам можно отнести довольно высокую стоимость подписки (от 990 Р в месяц) и отсутствие полноценной поддержки AI-ответов и электронной почты как канала связи.

2.2.2 Angry.Space

Платформа Angry.Space фокусируется на агрегации комментариев и личных сообщений из социальных сетей (VK, Instagram, Telegram и др.) в единый интерфейс. Она ориентирована в первую очередь на модерацию, реакцию на пользовательские запросы и отслеживание активности.

Сервис предоставляет удобную навигацию по диалогам, возможность настроить шаблоны ответов и фильтрацию сообщений по источникам. Присутствует базовая аналитика, а также возможность подключения нескольких сотрудников.



Основные возможности сервиса

Все возможности

Мониторинг комментариев и упоминаний

Смотрите все комментарии к своим постам и @упоминаниям

Отправляйте ответы из сервиса

Удаляйте комментарии

Отвечайте с помощью картинок, гифок, эмодзи

Ищите по тексту комментариев

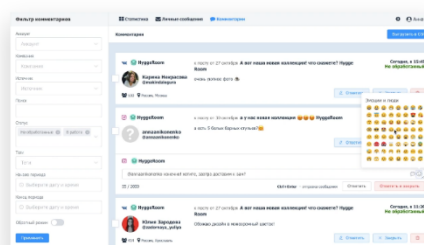
Мониторинг личных сообщений

Уведомления

Работа с пользователями

Статистика и аналитика

Контроль



Сфокусируйтесь на общении с клиентами

Для работы со всеми сообществами вам понадобится только одна вкладка - с Angry.Space. Вы ответите на вопросы клиентов быстрее, не тратя время на лишние переключения.

"С Angry.Space мы обрабатываем в 9 раз больше обращений из соцсетей, потому что сервис автоматизирует всю рутину."

Наталья Цвентарная, Account Group Head Manager, CatzWolf

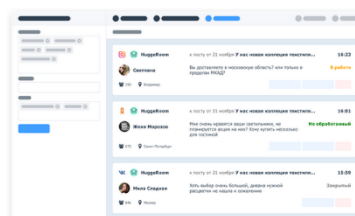


Рисунок 2 - Главная страница сайта «Angry.Space»

К преимуществам можно отнести простоту использования и качественную интеграцию с соцсетями. Однако платформа не предлагает расширенных инструментов CRM, а электронная почта и AI-ответы не поддерживаются, что ограничивает её функциональность в сравнении с многоканальными решениями.

2.2.3 Pact

Сервис Pact ориентирован на мультиканальное взаимодействие с клиентами. Он поддерживает множество каналов, включая WhatsApp, Telegram, VK, Instagram и Facebook Messenger. Pact предоставляет инструменты для автоматизации общения, ведения сделок, а также интеграцию с amoCRM и Битрикс24.

Интерфейс позволяет отслеживать переписку, назначать ответственных операторов, создавать воронки продаж и использовать автоответы. Также доступны функции рассылок и интеграция с чат-ботами.

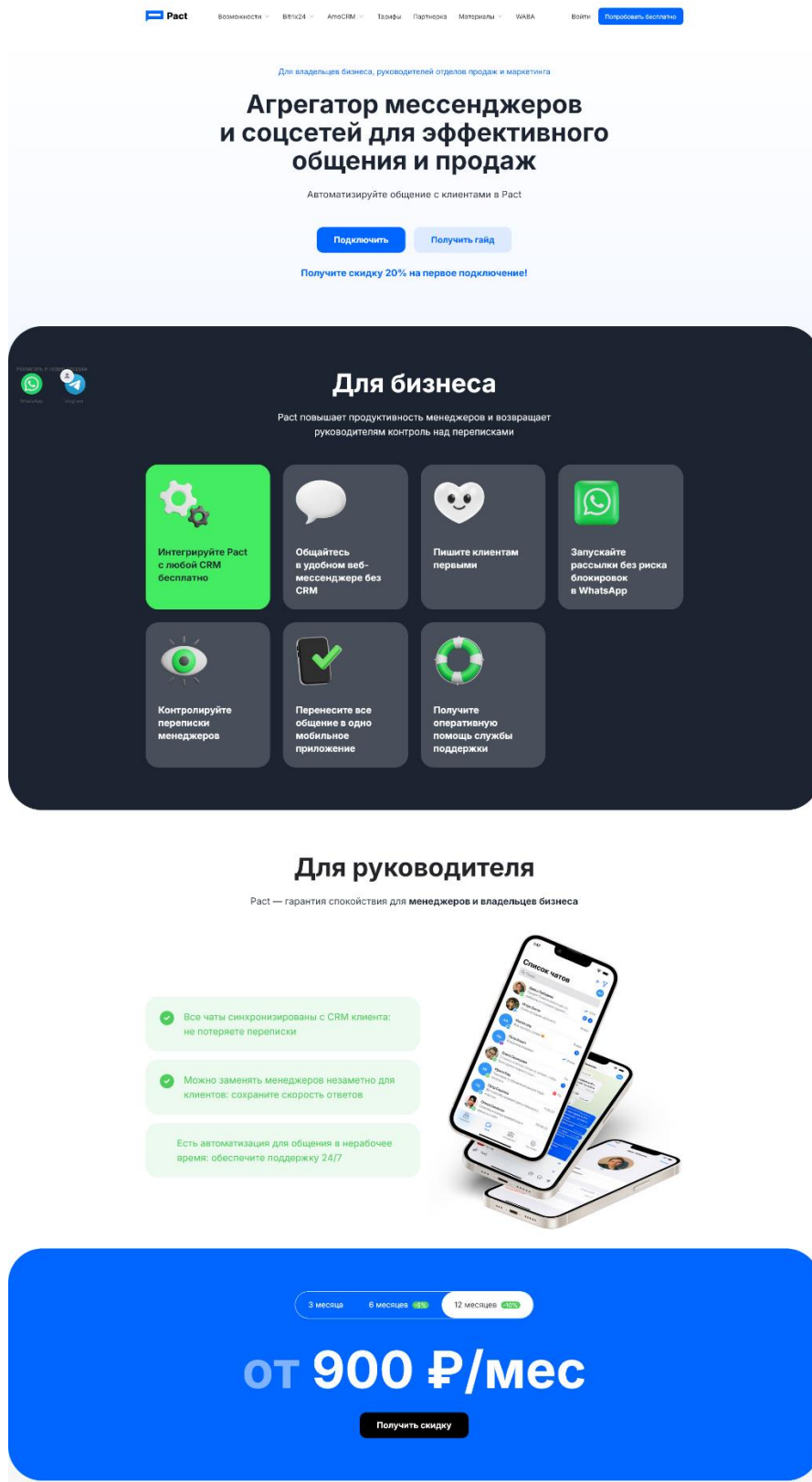


Рисунок 3 - Главная страница сайта «Раст»

Главным преимуществом Rast является широкий охват каналов и гибкость в настройке. К минусам можно отнести высокую стоимость (от 990 Р в месяц), необходимость внесения депозита для WhatsApp Business API, а также отсутствие интеграции с email и поддержки AI-функций.

2.2.4 Quickley

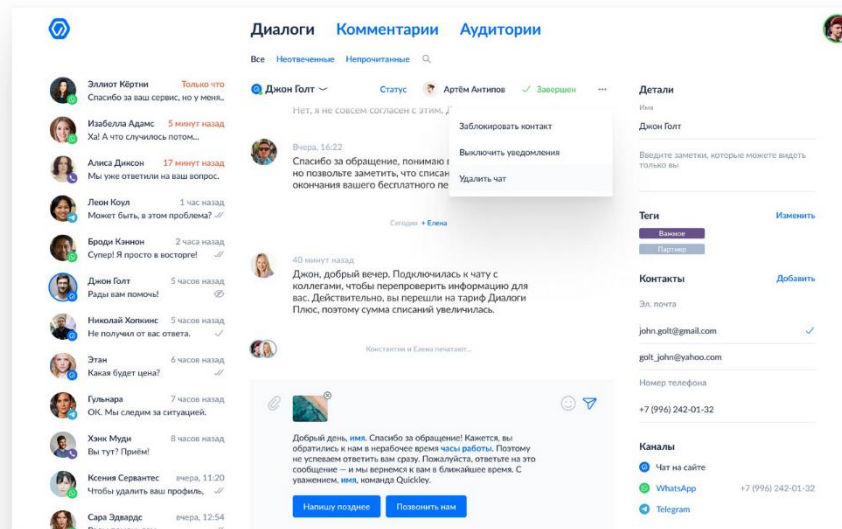
Платформа Quickley создана с расчётом на малый бизнес и предлагает простое в использовании решение для обработки сообщений из мессенджеров. Интерфейс ориентирован на быструю работу операторов: есть доступ к шаблонам, чат-ботам, базовым отчётам и таймерам.

Quickley поддерживает WhatsApp, Telegram, Viber и VK, а также предоставляет бесплатную версию с ограниченным функционалом. Основные возможности включают настройку автоматических ответов, сценариев диалога и режимов работы.

Quickley — единый бизнес-чат для ваших соцсетей и мессенджеров

Отвечайте на все сообщения клиентов в одном месте и всегда оставайтесь на связи со своей аудиторией

Попробовать бесплатно



Все входящие сообщения в одном месте

Получайте все сообщения из онлайн-чата, WhatsApp, групп ВКонтакте, ботов Telegram и Viber в одном удобном месте



Продажи

С помощью Quickley компании любого размера быстрее и эффективнее обрабатывают заявки и повышают конверсию своих сайтов.



Поддержка

Отвечайте на вопросы и претензии пользователей из всех каналов связи, тем самым повышая лояльность аудитории и увеличивая повторные покупки.



Автоматизация

Создавайте zero-code скрипты, которые автоматизируют рутинную коммуникацию с клиентами и экономят время и деньги вашей компании.

Рисунок 4 - Главная страница сайта «Quickley»

К достоинствам сервиса можно отнести невысокую стоимость (от 590 Р в месяц), простоту внедрения и поддержку нескольких пользователей в рамках одной команды. К недостаткам — отсутствие AI-ответов, ограниченные функции CRM и базовый уровень аналитики.

2.2.5 Вывод на основе анализа конкурентов

Анализ существующих решений показывает, что большинство сервисов ориентированы либо на интеграцию с CRM, либо на работу исключительно с социальными сетями. Лишь немногие платформы обеспечивают поддержку email, полноценную аналитику и AI ответы.

При разработке проекта «DialogX» необходимо учесть выявленные недостатки и реализовать:

- поддержку наиболее популярных мессенджеров (VK, Telegram, WhatsApp) и электронной почты;
- поддержка AI для автоответов;
- наличие CRM модуля;
- встроенные шаблоны для диалогов;
- тарифную модель, выгодную для малого бизнеса.

Это позволит создать конкурентоспособный продукт, сочетающий функциональность, удобство и доступность.

3 Реализация

3.1 Средства реализации

Разрабатываемое веб-приложение «DialogX» реализуется на клиент-серверной архитектуре. Система разделяется на две основные части: серверную (backend) часть, реализующую бизнес-логику, и клиентскую (frontend), предоставляющую пользовательский интерфейс. Обмен данными между этими компонентами осуществляется через REST API, с применением формата JSON.

Для командной разработки использовалось ветвление Git-flow, предполагающие наличие основной ветки main, а также временных веток для разработки отдельных функций. Такой подход упрощает управление версиями и интеграцию кода.

Для разработки серверной части приложения выбраны следующие технологии:

- Java – основной язык программирования backend-части, обеспечивает высокую производительность, стабильность при масштабировании и широкую поддержку библиотек для корпоративной разработки;
- Spring Boot – фреймворк для создания REST-приложений, обеспечивает быструю конфигурацию, управление зависимостями и встроенную поддержку модулей (Web, JPA, Security и др.), что ускоряет процесс разработки;
- Spring Security – фреймворк для настройки аутентификации и авторизации, в проекте используется для реализации защищённого входа пользователей с помощью JWT-токенов;
- PostgreSQL – реляционная база данных, используемая для хранения информации о пользователях, диалогах, подписках, шаблонах ответов и сделках, обеспечивает надёжность хранения и поддержку сложных SQL-запросов;

- Docker – применяется для контейнеризации серверной части, упрощает развёртывание приложения и его переносимость между средами разработки;
- Qwen 2.5 – AI-модель, используемая для генерации интеллектуальных автоответов на входящие сообщения, интегрируется через внутренний сервис обработки запросов и используется в связке с шаблонами.

Для разработки клиентской части приложения выбраны следующие технологии:

- HTML – язык разметки, используемый для построения структуры веб-страниц, включая размещение элементов интерфейса: форм, кнопок, заголовков и таблиц;
- CSS – язык стилей, применяемый для визуального оформления элементов, обеспечивает адаптивность интерфейса, настройку цветовой схемы и расположения компонентов;
- JavaScript – основной язык программирования фронтенда, реализующий логику взаимодействия пользователя с приложением: отправка форм, переключение вкладок, обновление данных без перезагрузки страницы;
- Tailwind CSS – фреймворк, позволяющий быстро настраивать внешний вид компонентов с помощью встроенных классов без необходимости писать пользовательские CSS-файлы;
- Vite – современный инструмент для сборки и локальной разработки, обеспечивает быструю загрузку, поддержку модульной структуры и горячую перезагрузку при изменении кода;
- React – библиотека для создания компонентных пользовательских интерфейсов, применяется для построения одностраничного приложения с разделением логики по функциональным блокам;
- React Router – библиотека для организации навигации внутри приложения, позволяет переключаться между страницами

(авторизация, диалоги, настройки) без полной перезагрузки, сохраняя состояние интерфейса.

3.2 Разработка frontend (клиентской части веб-приложения)

Клиентская часть приложения была написана на языке JavaScript с использованием фреймворка React. Для отправки запросов с клиентской части приложения на серверную часть используется стандартный инструмент для работы с HTTP в JavaScript – fetch. Он предоставляет интерфейс для выполнения HTTP-запросов прямо из браузера.

Проект имеет следующую структуру:

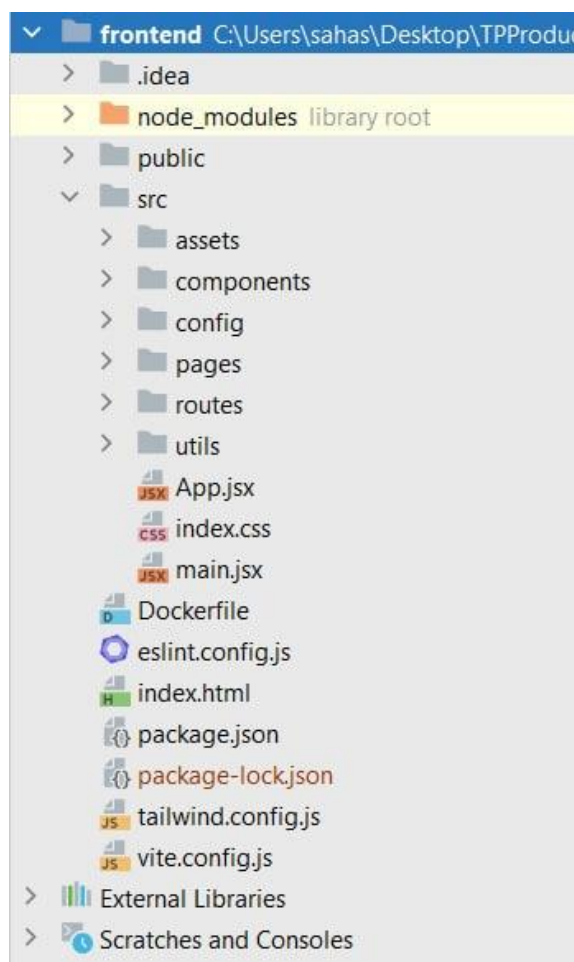


Рисунок 5 - Структура клиентской части приложения

Для создания модульной и переиспользуемой архитектуры, приложение было разбито на компоненты. Каждый компонент отвечает за конкретную функциональность или отображение определенной части пользовательского интерфейса:

- модуль «config» содержит конфигурационный файл с эндпоинтами для работы с API;
- модуль «utils» содержит файлы, связанные с управлением контекстом и роутингом;
- модуль «pages» содержит компоненты, представляющие отдельные страницы приложения;
- модуль «routing» содержит файлы, связанные с маршрутизацией, такие как вспомогательные функции и константы маршрутов;
- файл «main.jsx» является точкой входа в приложение, он служит главным файлом, с которого начинается выполнение приложения;
- файл «index.html» основной HTML-файл, который, задаёт базовую разметку страницы и подключает скрипт main.jsx как модуль;
- файл «App.jsx» является основным компонентом React, в который встраиваются остальные компоненты;
- файл «vite.config» является конфигурационным файлом, определяет настройки разработки, сборки приложения;
- файл «eslint.config.js» является конфигурационным файлом для настройки статического анализа JavaScript кода.

3.3 Разработка backend (серверной части веб-приложения)

Серверная часть приложения была написана на языке Java с использованием Spring Framework и СУБД PostgreSQL.

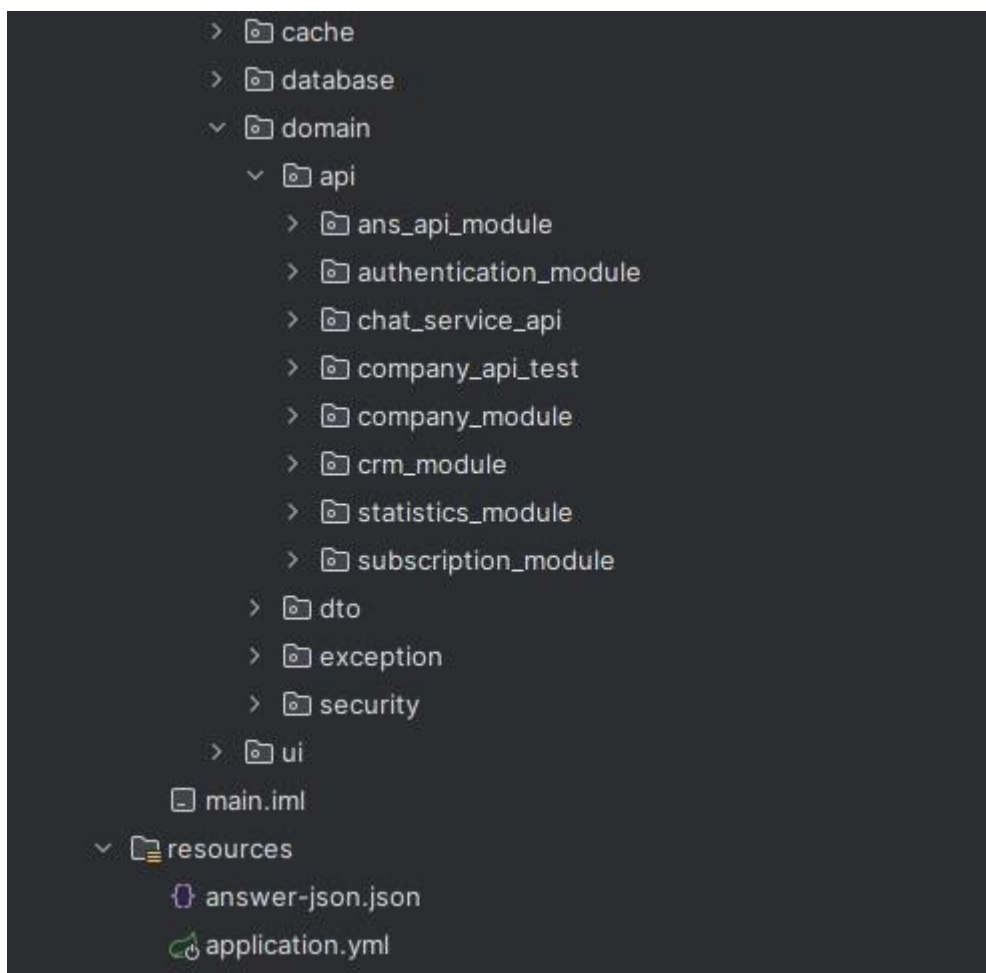


Рисунок 6 - Структура серверной части приложения

Проект имеет следующую структуру:

- пакет «cache» содержит конфигурационные файлы для работы с Redis, Spring Data Redis предоставляет инструменты для кэширования и быстрого получения необходимой информации;
- пакет «database» содержит в себе классы, которые описывают данные таблиц базы данных в виде объектов с помощью ORM, также в пакете

- содержатся репозитории, позволяющие получать необходимые данные из БД с помощью SQL запросов;
- пакет «api» внутри содержит пакеты модулей, отвечающих за бизнес-логику, каждый модуль имеет контроллеры для взаимодействия клиентской части приложения с сервером, сервисы – классы, реализующие бизнес-логику проекта, классы ошибок, которые могут возникнуть;
 - пакет «ans_api_module» – пакет модуля, который отвечает за предоставление пользователям шаблонных ответов, предварительно улучшенных с помощью внешнего AI-сервиса, модуль получает сырой текст шаблонного ответа, отправляет его на обработку (коррекцию, генерацию вариантов и т.д.) в отдельный контейнер с нейросетью, а затем возвращает или доставляет обогащенный ответ, также включает логику взаимодействия с AI-сервисом и обработку возможных ошибок, обеспечивая отказоустойчивость на сервере;
 - пакет «authentication_module» – пакет модуля, который отвечает за авторизацию и аутентификацию пользователя, модуль проверяет и выдает JWT токены которые используются для авторизации пользователя, предоставляет контроллеры для регистрации, доступа и управления аккаунтом пользователя;
 - пакет «chat_service_api» – пакет модуля, который отвечает за работу чата, он реализует логику работы чата для операторов;
 - пакет «company_module» – пакет модуля, который отвечает за компанию, он предоставляет контроллеры для настройки компании, добавления и выдачи прав участникам компании;
 - пакет «crm_module» – пакет модуля, который отвечает за работу CRM, он позволяет участникам компании создавать сделки с клиентами, отслеживать этапы их выполнения, переносить выполненные сделки в архив;

- пакет «statistics_module» – пакет модуля, который отвечает за сбор и отображения статистики, модуль позволяет получить статистику работы операторов;
- пакет «subscription_module» – пакет модуля, который отвечает за подписку, модуль позволяет пользователю оформить или продлить подписку, а также высчитать ее стоимость;
- пакет «dto» содержит в себе модели запросов и ответов клиенту;
- пакет «exception» содержит в себе классы, отвечающие за глобальную обработку ошибок, в пакете каждого модуля содержится набор ошибок, которые могут произойти во время работы модуля, ошибки перехватываются обработчиком, который передает сообщение об ошибке и http статус ответа;
- пакет «security» содержит в себе классы, которые ограничивают доступ пользователя к некоторым запросам в соответствии с правами, которые он имеет.

Для развертывания приложения был использован облачный сервер (VPS) от TimeWeb Cloud. Для разворачивания был использован Docker – инструмент для контейнеризации приложений. На сервере работают шесть контейнеров:

- frontend – для доступа к клиентской части сервиса;
- backend – отвечающий за бизнес-логику;
- nginx – для проксирования запросов на сервер;
- ai-service – для работы с нейросетью;
- postgres – для работы с базой данных;
- redis – для кэширования данных.

Для описания спецификации API использовался Swagger

3.4 Основная функциональность сайта

3.4.1 Для неавторизованных пользователей

При открытии сайта пользователь попадает на приветственную страницу, где представлена основная информация о нашем сайте DialogX – интеллектуальном автоответчике для бизнеса. На этой странице описаны основные преимущества сервиса, такие как мгновенные AI-ответы, интеграции с WhatsApp, Telegram, VK, email. Так же описано, как наше приложение позволяет экономить времени, благодаря автоматизации общения. Кроме того, пользователь может ознакомиться с тарифами и назначением платформы. В верхней части сайта доступны кнопки «Войти» и «Зарегистрироваться».

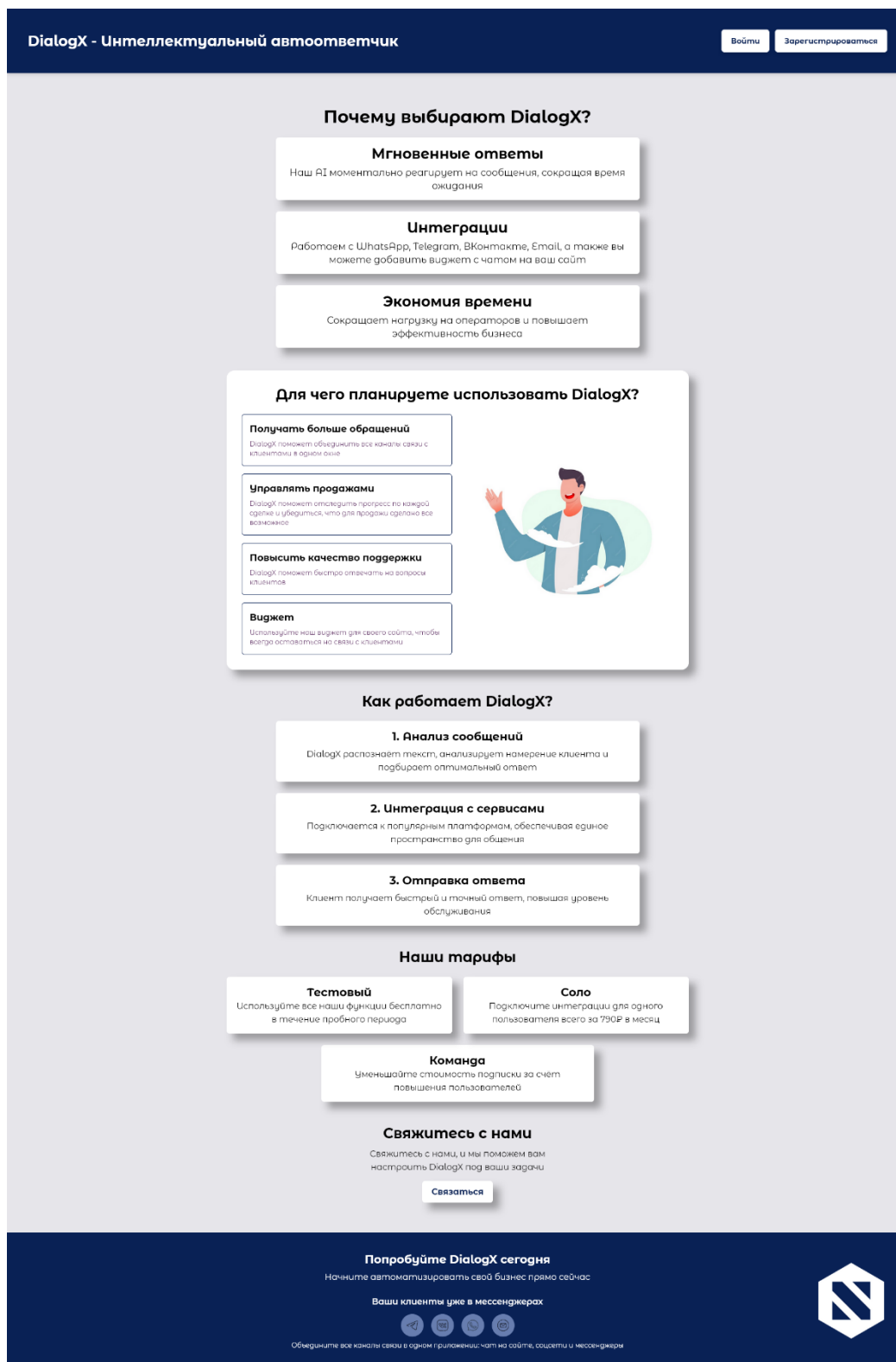


Рисунок 7 - Главная страница

При нажатии на кнопку «Зарегистрироваться» пользователь переходит на страницу регистрации, где ему предлагается ввести адрес электронной

почты и пароль. Интерфейс минималистичен, что облегчает процесс регистрации.

Рисунок 8 - Страница регистрации

Если пользователь уже имеет аккаунт, он может перейти на страницу авторизации, где необходимо ввести адрес электронной почты и пароль. В случае, если доступ к аккаунту утрачен, доступна функция восстановления пароля. Также с этой страницы можно вернуться к регистрации, если пользователь ещё не создавал учётную запись.

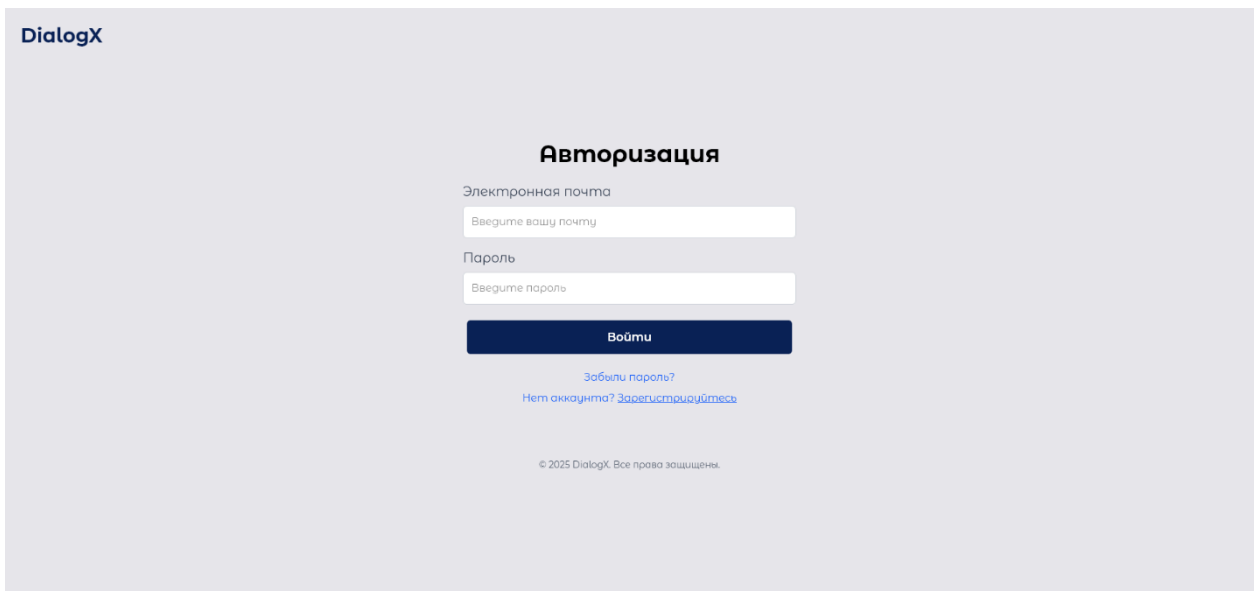


Рисунок 9 - Страница авторизации

После успешной авторизации пользователь получает доступ к личному кабинету и функциональности, соответствующей его роли.

3.4.2 Для авторизованных пользователей

После авторизации пользователю необходимо оформить подписку, чтобы получить доступ к функционалу приложения.

Страница подписки доступна после авторизации в системе. Она позволяет выбрать и оформить один из тарифных планов, а также управлять действующей подпиской.

Если пользователь не имеет оформленной подписки, то на странице отображается перечень доступных тарифов. Каждый тариф содержит краткое описание, стоимость и доступный функционал. Пользователь может выбрать между тремя вариантами:

- тестовый — бесплатный доступ ко всем функциям на срок 7 дней, подходит для первичного знакомства с платформой;

- соло — подписка на одного пользователя, предлагается выбрать срок действия подписки, включает одного оператора, поддержку всех каналов связи и доступ к аналитике;
- команда — тариф для двух и более пользователей, цена рассчитывается динамически в зависимости от количества участников и длительности подписки (пример: 1501 Р за 2 пользователей на 1 месяц).

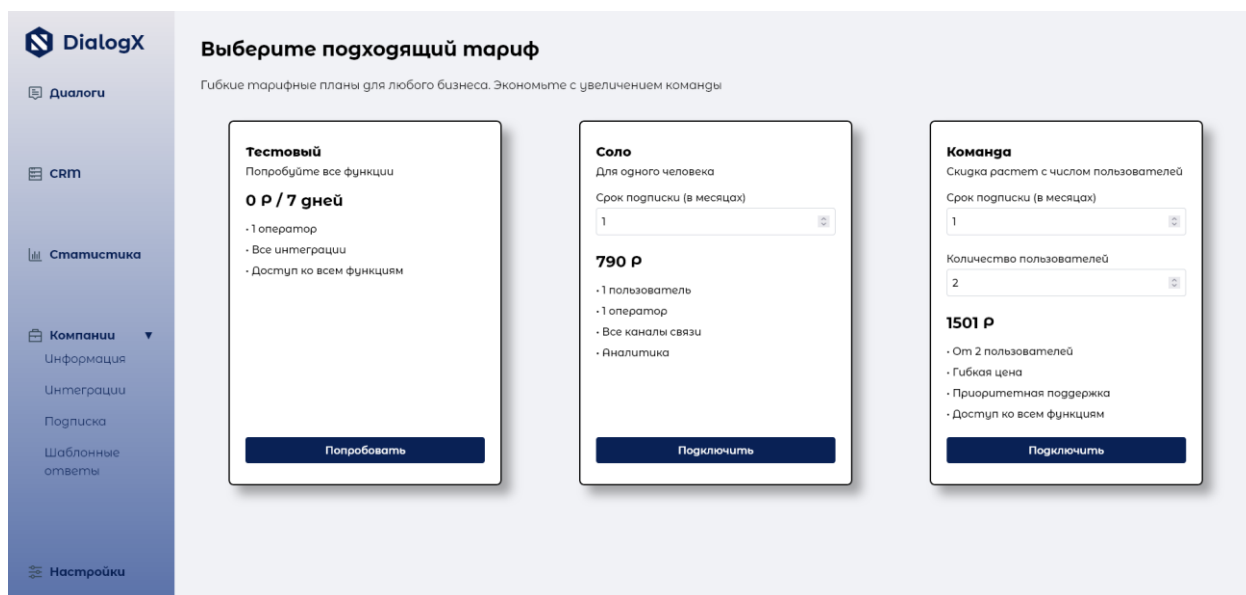


Рисунок 10 - Страница подписки

После выбора тарифа пользователь нажимает кнопку «Подключить» или «Попробовать», в зависимости от типа тарифа.

После оформления подписки текущий пользователь получает роль администратора, на странице отображается информация о количестве пользователей в подписке, сроке действия (дата окончания подписки). Информация на странице оформленной подписки доступна только администратору.

Также предоставлены дополнительные функции продление текущей подписки и добавление новых пользователей (с динамическим перерасчётом стоимости).

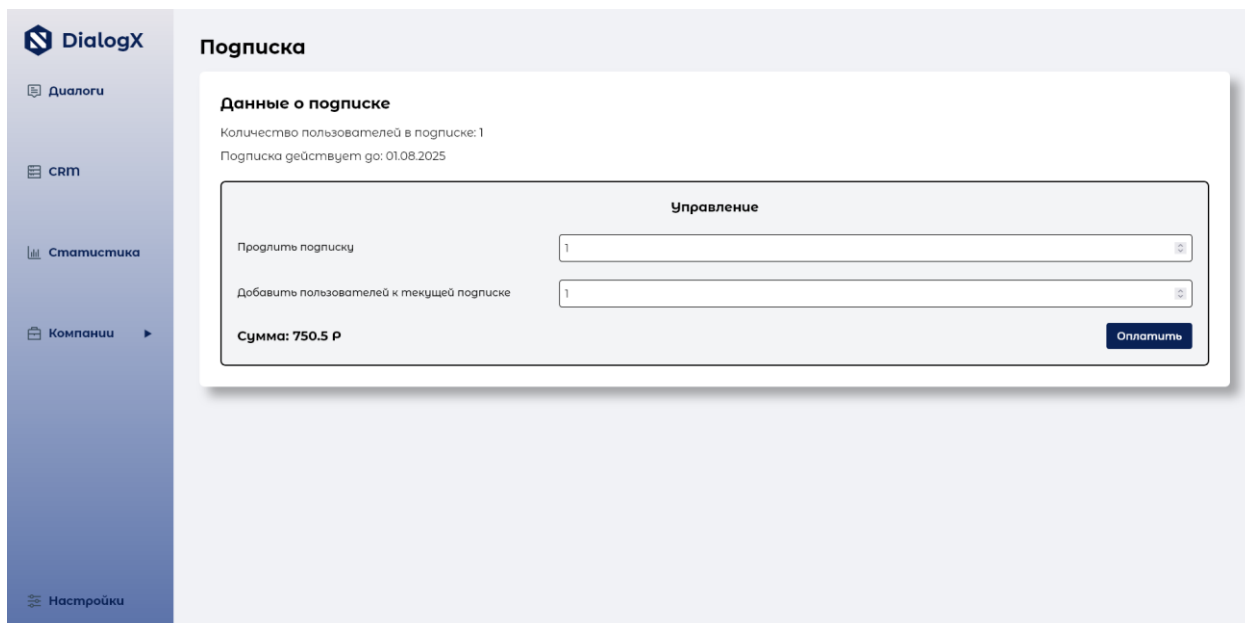


Рисунок 11 - Страница оформленной подписки

Таким образом, пользователь может в любое время изменить параметры подписки и продлить её срок.

3.4.3 Для оператора

Пользователь с ролью оператора получает доступ к ключевому функционалу приложения, необходимому для ежедневной работы с обращениями клиентов.

Основной рабочий инструмент оператора – страница «Диалоги», где отображаются все обращения клиентов, поступившие через мессенджеры или чат-виджет.

На данном экране отображается история переписки с клиентом. Интерфейс включает список диалогов и активный чат. В правой части экрана мы можем добавить тэг, создать сделку, оставить комментарий. Снизу находится диалоговое окно, в котором оператор вручную может отправить сообщение

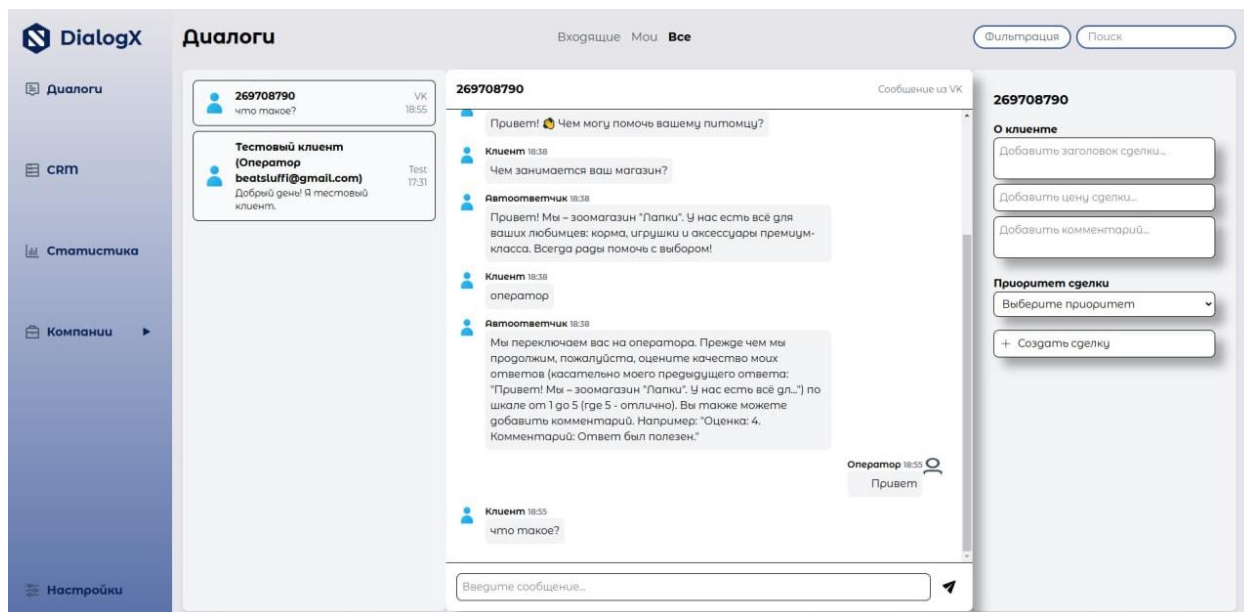


Рисунок 12 - Страница диалогов

Еще одним инструментом является страница CRM. Она используется для управления сделками, связанными с обращениями клиентов. Все сделки группируются по этапам:

- новая;
- пауза;
- в работе;
- завершена;
- провалена.

Оператор может добавлять сделки в архив, фильтровать их по статусу и контролировать денежные суммы на каждом этапе. Это позволяет не только отслеживать обращения, но и управлять воронкой продаж внутри системы.

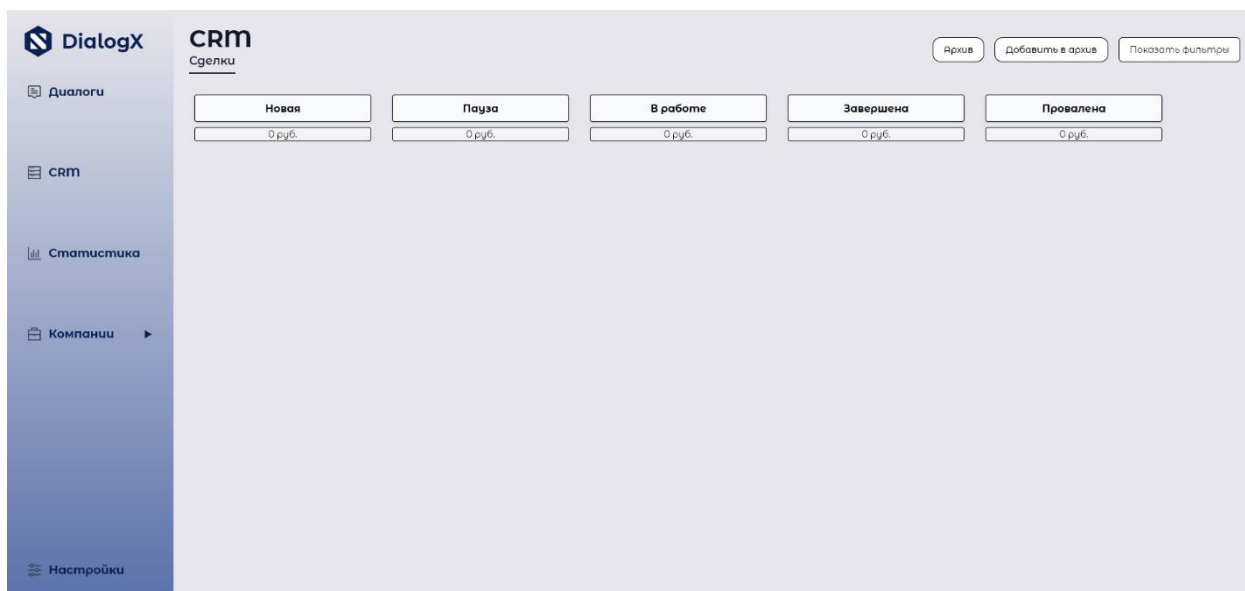


Рисунок 13 - Страница CRM

Раздел «Статистика» предоставляет оператору данные о его работе за выбранный период: день, неделю, месяц или год. В интерфейсе отображаются ключевые метрики: количество пропущенных диалогов, среднее время ответа, число обработанных чатов, длительность рабочего времени и распределение активности по каналам связи (Telegram, WhatsApp, Email, VK).

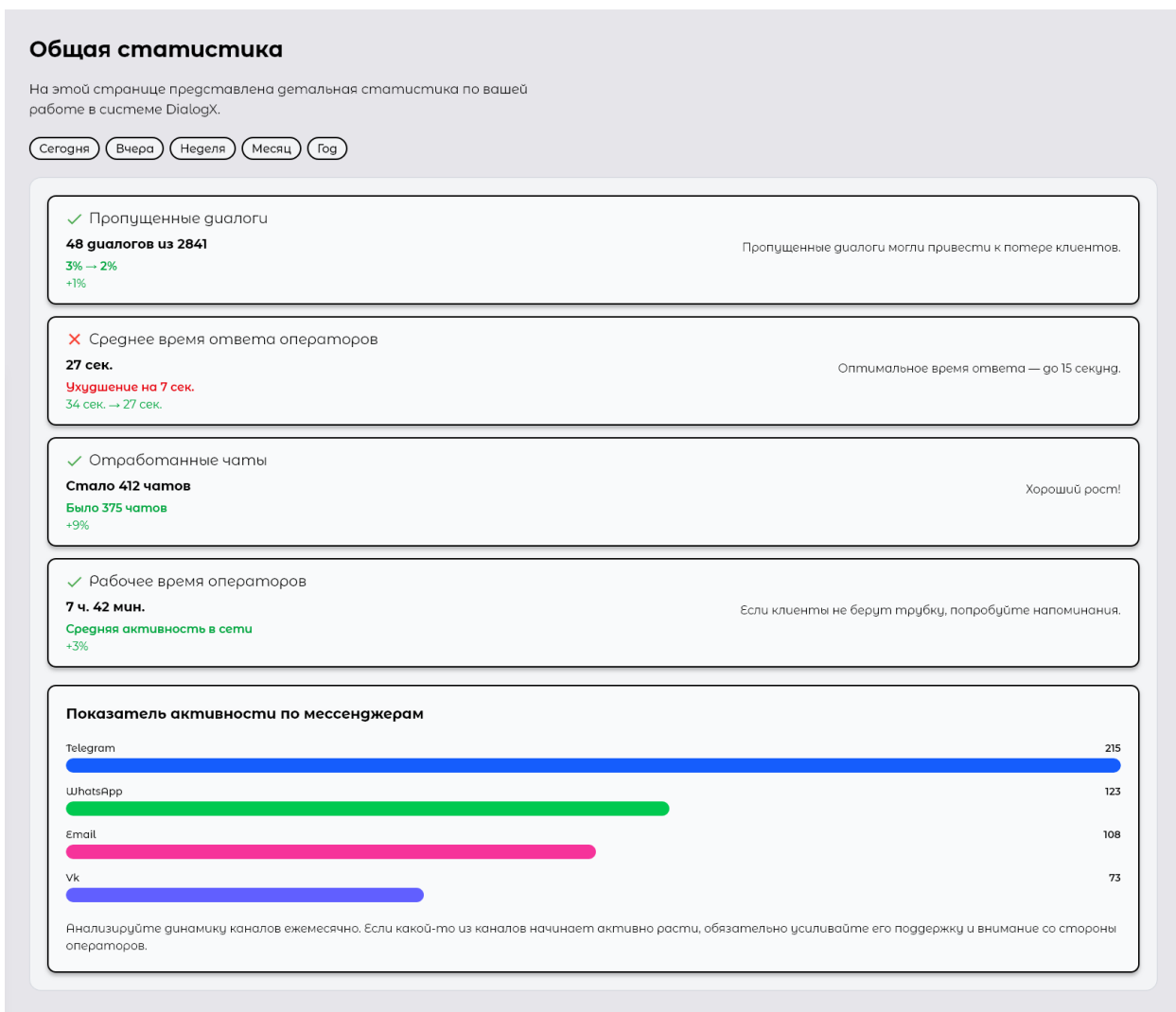


Рисунок 14 - Страница статистики

Благодаря визуальной аналитике оператор может отслеживать личную эффективность и реагировать на изменения.

3.4.4 Для администратора

Пользователь с ролью администратора имеет весь функционал от оператора, а также доступ к расширенным страницам интерфейса, предназначенным для управления компанией, подпиской, пользователями, интеграциями и шаблонами ответов. Ниже приведено описание ключевых страниц, доступных только администратору.

На странице «Информация» отображаются основные сведения о текущей организации, зарегистрированной в системе. Кроме того, администратор может добавлять новых сотрудников и изменять их роли. На данной странице администратор может ознакомиться со следующей информацией:

- название компании;
- краткое описание (если указано);
- список сотрудников с их ролями и адресами электронной почты.

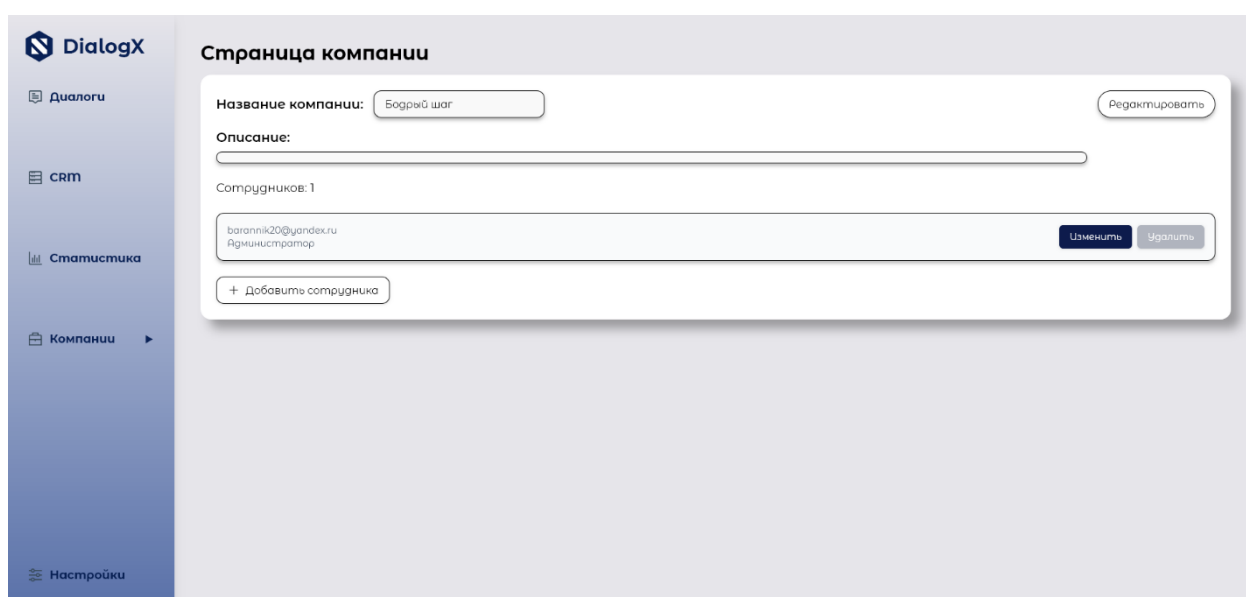


Рисунок 15 - Страница информации о компании

Следующим инструментом администратора являются интеграции. Раздел «Интеграции» предоставляет администратору возможность подключения внешних каналов связи, с которыми будет работать платформа. На странице есть возможность подключения:

- интеграции мессенджеров: Telegram, WhatsApp, ВКонтакте;
- интеграция электронной почты;
- установка чат-виджета на сторонний сайт.

Каждая из интеграций может быть активирована через кнопку «Подключить». Ниже отображается список уже установленных интеграций.

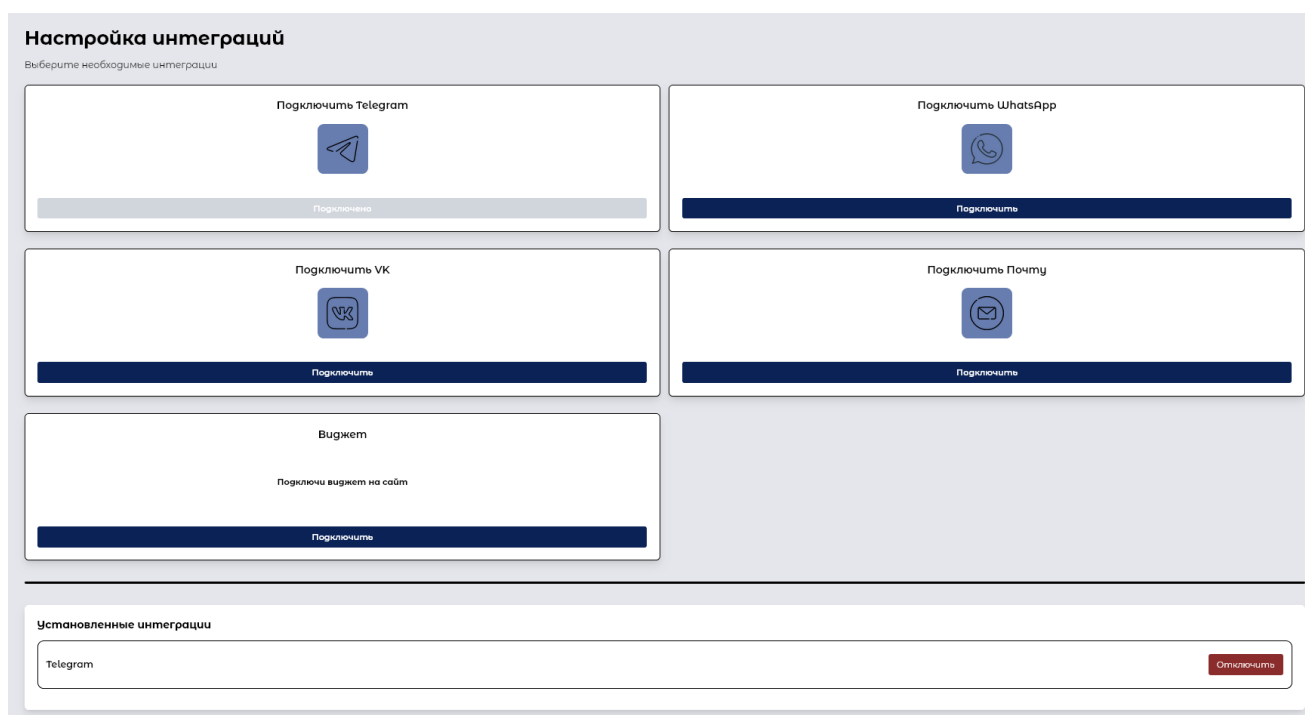


Рисунок 16 - Страница интеграций

На странице «Шаблонные ответы» администратор может управлять заранее подготовленными фразами и сообщениями, которые используются для автоматического ответа на входящие сообщения. Интерфейс включает следующие функции:

- загрузка и добавление новых шаблонов;
- редактирование существующих;
- удаление шаблонного ответа;
- возможность скачать файл с примером шаблонных ответов.

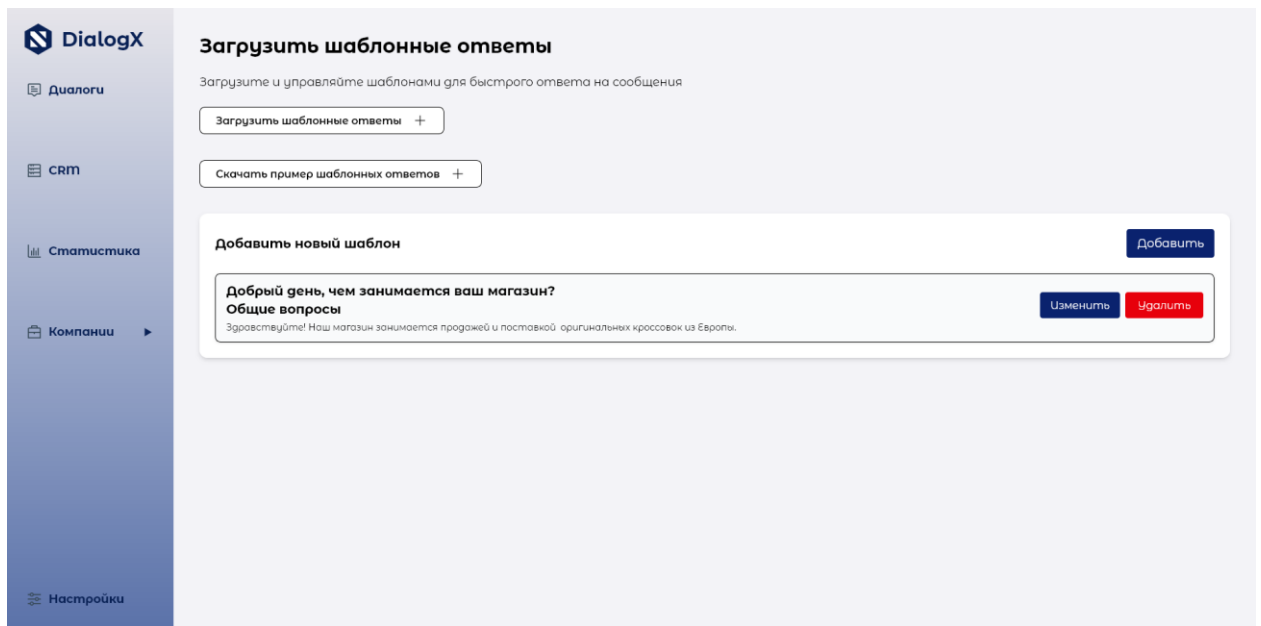


Рисунок 17 - Страница шаблонных ответов

Наличие страницы шаблонных ответов позволяет ускорить обработку повторяющихся запросов и снизить нагрузку на операторов.

4 Тестирование

Чтобы проверить работоспособность приложения, было выполнено тестирование системы с использованием указанных типов тестов:

- тестирование пользовательского интерфейса (UI-тестирование);
- тестирование API;
- модульное (Unit) тестирование.

4.1 UI-тестирование

Тестирование пользовательского интерфейса (UI) – проверка внешнего вида и поведения элементов на страницах приложения. Цель такого тестирования – убедиться, что интерфейс работает корректно, понятен для пользователя и соответствует требованиям по дизайну и логике работы.

UI-тестирование позволяет проверить, насколько удобно пользователю взаимодействовать с системой: работают ли кнопки, корректно ли открываются страницы, правильно ли отображаются данные.

В таблице 1 приведены результаты проведенного UI-тестирования.

Таблица 1 - Результаты UI-тестирования

Сценарий теста	Ожидаемый результат	Результат
Нажатие на кнопку «Войти» в шапке сайта	Переход на страницу авторизации	Пройден
Нажатие на кнопку «Регистрация» в шапке сайта	Переход на страницу регистрации	Пройден
Нажатие на кнопку «Зарегистрируйтесь» на странице авторизации	Переход на страницу регистрации	Пройден

Регистрация аккаунта с новыми данными	Успешная регистрация и переход в рабочую область	Пройден
Авторизация по зарегистрированным данным	Успешный вход и переход в рабочую область	Пройден
Авторизация по несуществующим данным	Ошибка «Неверный email или пароль.»	Пройден
Нажатие на логотип на страницах регистрации и авторизации	Переход на главную страницу	Пройден
Нажатие на кнопку «Диалоги» в рабочей области	Переход в диалоги	Пройден
Нажатие на кнопку «CRM» в рабочей области	Переход в CRM	Пройден
Нажатие на кнопку «Архив» во вкладке CRM	Переход в архив	Пройден
Нажатие на кнопку «Показать фильтры» во вкладке CRM	Показ фильтров над таблицей сделок	Пройден
Нажатие на кнопку «Статистика» в рабочей области	Переход в статистику	Пройден
Нажатие на кнопку «Компании» в рабочей области	Открытие выпадающего меню	Пройден
Нажатие на кнопку «Информация» в выпадающем меню «Компания»	Переход на страницу компании	Пройден
Нажатие на кнопку «Интеграции» в выпадающем меню «Компания»	Переход на страницу настроек интеграций	Пройден
Нажатие на кнопку «Подписка» в выпадающем меню «Компания»	Переход на страницу с подписками	Пройден

Нажатие на кнопку «Шаблонные ответы» в выпадающем меню «Компания»	Переход на страницу редактирования шаблонных ответов	Пройден
Нажатие на кнопку «Настройки»	Переход на страницу редактирования профиля	Пройден
Заполнение графов при редактировании профиля	Обновление информации в профиле	Пройден
Допущение ошибок в заполнении графов при редактировании профиля	Вывод ошибки	Пройден
Смена пароля	Успешная смена пароля	Пройден
Нажатие на логотип в рабочей области	Переход на главную страницу	Пройден
Нажатие на значок профиля	Переход в рабочую область	Пройден
Нажатие на «Выйти»	Выход из аккаунта	Пройден

4.2 Тестирование API

В дополнение к автоматизированному тестированию контроллеров было выполнено ручное тестирование основных API-эндпоинтов с использованием инструмента Postman. Цель – проверить корректность работы ключевых пользовательских сценариев.

Проверялись следующие аспекты:

- правильность возвращаемых HTTP-статусов (200, 201, 400, 401, 403, 404, 500);
- соответствие структуры JSON-запросов и ответов;
- корректная обработка как валидных, так и ошибочных входных данных.

В процессе ручного тестирования были проверены основные пользовательские сценарии взаимодействия с API системы DialogX.

Сначала был выполнен запрос на регистрацию нового пользователя через эндпоинт POST /api/registration/register. Сервер успешно обработал запрос, вернув статус 200 OK и логическое значение true в теле ответа, что подтвердило успешную регистрацию.

Далее была проверена процедура подтверждения регистрации по коду с использованием запроса POST /api/registration/check-code. В ответ система вернула статус 201 Created и корректно установила access и refresh токены в cookies, что подтверждает успешную активацию учётной записи.

После этого был выполнен запрос на вход в систему через POST /api/auth/login. Аутентификация прошла без ошибок: в ответе возвращён статус 200 OK, а ранее установленные токены были успешно обновлены.

Следующим этапом проверялся доступ к защищённому ресурсу – списку пользователей (GET /api/users). После прохождения авторизации сервер корректно предоставил доступ и вернул массив с данными пользователей, что подтвердило правильную работу механизма авторизации.

В завершение была выполнена попытка получить доступ к тому же ресурсу без авторизации (без отправки cookies). В этом случае сервер вернул ожидаемый ответ с кодом 401 Unauthorized и сообщением об ошибке, указывающим на отсутствие доступа.

Результаты проверок соответствовали ожидаемому поведению системы.

4.3 Модульное (Unit) тестирование серверной части

Для проверки корректности функционирования отдельных компонентов серверной части приложения было проведено модульное тестирование. Основное внимание уделялось проверке бизнес-логики, правильной обработке

граничных условий, а также обработке исключительных ситуаций. Тестирование проводилось с использованием фреймворков JUnit 5 и Mockito.

Модуль искусственного интеллекта (`ans_api_module`) включал тестирование сервисов, связанных с генерацией и подбором автоматических ответов. В рамках этого модуля были протестированы:

- сервис поиска и ранжирования ответов (`AnswerSearchServiceImplTest`);
- сервис обработки и коррекции текстов (`TextProcessingServiceTest`);
- основная логика автоответчика (`AutoResponderServiceImplTest`);
- механизм пакетной загрузки и обработки файлов (`FileProcessorServiceImplTest`);
- операции с предопределёнными ответами (`PredefinedAnswerServiceImplTest`);
- сбор и анализ метрик пакетных задач (`JobMetricsServiceImplTest`).

В модуле аутентификации и авторизации (`authentication_module`) были протестированы контроллеры регистрации и входа в систему (`AuthControllerTest`, `RegistrationControllerTest`), а также сервисы, реализующие безопасность на основе Spring Security (`UserDetailsServiceImplTest`). Отдельное внимание уделялось обработке куков и JWT-токенов (`AuthCookieServiceImplTest`, `JWTUtilsServiceImplTest`), а также основным сервисам авторизации и управления ролями (`AuthServiceImplTest`, `RegistrationServiceImplTest`, `RoleServiceImplTest`).

Модуль Чатов и Сообщений (`chat_service_api`) охватывал тестирование интеграционных контроллеров, отвечающих за приём сообщений из внешних каналов (электронная почта, Telegram, VK, WhatsApp) – `IntegrationController*Test`. Кроме того, были проверены сервисы управления клиентами, автоматического распределения обращений между операторами, работы с уведомлениями и учётными данными пользователей (`ClientServiceImplTest`, `LeastBusyAssignmentServiceTest`, `NotificationServiceImplTest`, `UserServiceImplTest`).

В модуле управления компаниями (company_module) тестировались сервисы, реализующие бизнес-логику управления организациями, их участниками и параметрами – это CompanyMembersServiceImplTest, CompanyServiceImplTest и CompanySettingsServiceImplTest.

В рамках модуля подписок (subscription_module) были протестированы классы, отвечающие за расчёт стоимости и сопровождение подписок: SubscriptionPriceCalculateServiceImplTest и SubscriptionServiceImplTest.

Также отдельное тестирование проводилось для контроллеров пользовательского интерфейса на стороне сервера – в рамках модуля UI Controllers (Backend). Здесь проверялась корректность обработки запросов, связанных с отображением данных чатов, с помощью класса ChatUiControllerTest.

Проведённое тестирование позволило убедиться в стабильной работе ключевых компонентов системы и выявить ряд ошибок на этапе разработки, что способствовало улучшению надёжности финального продукта.

5 Анализ пользовательского интерфейса

Для оценки эффективности пользовательского интерфейса и основных сценариев взаимодействия с веб-приложением DialogX был использован сервис Яндекс Метрика. В нём были настроены цели, отражающие ключевые действия пользователей: регистрация и выбор тарифных планов с последующим оформлением подписки.



Рисунок 18 - Созданные цели в Яндекс Метрике

Начальным этапом взаимодействия пользователей является приветственная страница. Цель «Зашли на приветственную страницу» была достигнута 164 раза при 43 уникальных визитах, что составило 100% конверсии. Это подтверждает, что все посетители, пришедшие на сайт, успешно загружают главную страницу.

Далее, на страницу регистрации перешли 34 пользователя в рамках 27 визитов, при этом конверсия составила 62,8%, что указывает на высокий интерес к получению доступа к системе.

На самой странице регистрации 20 пользователей нажали на кнопку «Зарегистрироваться» (в 14 визитах, 32,6% конверсии), и 17 из них подтвердили регистрацию (в 13 визитах, 30,2% конверсии). Это говорит о том, что большинство пользователей, начавших регистрацию, успешно проходят весь процесс.

Анализ действий на этапе выбора подписки показал следующие результаты:

- подписка «Соло»: зафиксировано 12 нажатий в рамках 10 визитов, конверсия – 23,3%;
- подписка «Командная»: также 12 нажатий, 10 визитов, 23,3% конверсии;
- подписка «Тестовая»: 6 нажатий в 5 визитах, с наименьшей конверсией – 11,6%.

Полученные данные показывают, что пользователи активно проходят путь от ознакомления с платформой до регистрации и выбора тарифа. Наибольший интерес вызывают подписки «Соло» и «Командная», что подтверждает актуальность их функционала. Эти данные могут быть использованы для дальнейшей оптимизации пользовательского пути и маркетинговой стратегии.

ЗАКЛЮЧЕНИЕ

В ходе выполнения данной курсовой работы были реализованы ключевые задачи, связанные с разработкой веб-приложения автоответчика DialogX. Основной целью проекта являлось создание приложения, способного автоматически обрабатывать входящие обращения из различных каналов связи, включая мессенджеры и электронную почту.

Была спроектирована архитектура клиент-серверного взаимодействия, реализована функциональность для разных ролей пользователей (неавторизованных пользователей, операторов и администраторов), а также реализована модель монетизации с помощью подписки со скидочной системой. Отдельное внимание уделено страницам диалогов, CRM, настройке интеграций и шаблонных ответов.

Для реализации проекта использовались современные инструменты и технологии: Java, Spring Boot, PostgreSQL, Docker на серверной стороне, и React, Tailwind CSS, Vite на клиентской. Также была интегрирована AI-модель Qwen 2.5 для генерации ответов на основе информации о компании и шаблонных ответов.

Таким образом, веб-приложения DialogX является инструментом для автоматизации клиентского сервиса. Оно позволяет значительно сократить время ответа, снизить нагрузку на операторов и повысить качество обработки сообщений. В перспективе планируется расширение функционала, внедрение мобильной версии и развитие AI-модуля на основе пользовательского опыта и отзывов.

Список использованных источников

1. Агрегатор мессенджеров для бизнеса Pact [Электронный ресурс]. – Режим доступа: URL: <https://www.pact.im/> – заглавие с экрана (Дата обращения 26.05.2025)
2. Сервис интеграции мессенджеров i2crm [Электронный ресурс]. – Режим доступа: URL: <https://i2crm.ru> – заглавие с экрана (Дата обращения 26.05.2025)
3. Отслеживание комментариев и сообщений в соцсетях - Angry.Space [Электронный ресурс]. – Режим доступа: URL: <https://angry.space/> – заглавие с экрана (Дата обращения 26.05.2025)
4. Quickley – единый бизнес-чат для ваших соцсетей и мессенджеров [Электронный ресурс]. – Режим доступа: URL: <https://quickley.com/ru> – заглавие с экрана (Дата обращения 26.05.2025)
5. Документация Spring Framework [Электронный ресурс]. – Режим доступа: URL: <https://spring.io/projects/spring-framework#learn> – заглавие с экрана (Дата обращения 25.05.2025)
6. Документация Docker [Электронный ресурс]. – Режим доступа: URL: <https://docs.docker.com/> – заглавие с экрана (Дата обращения 28.05.2025)
7. Документация Яндекс Метрика [Электронный ресурс]. – Режим доступа: URL: <https://yandex.ru/support/metrika/ru/> – заглавие с экрана (Дата обращения 28.05.2025)

ПРИЛОЖЕНИЕ А

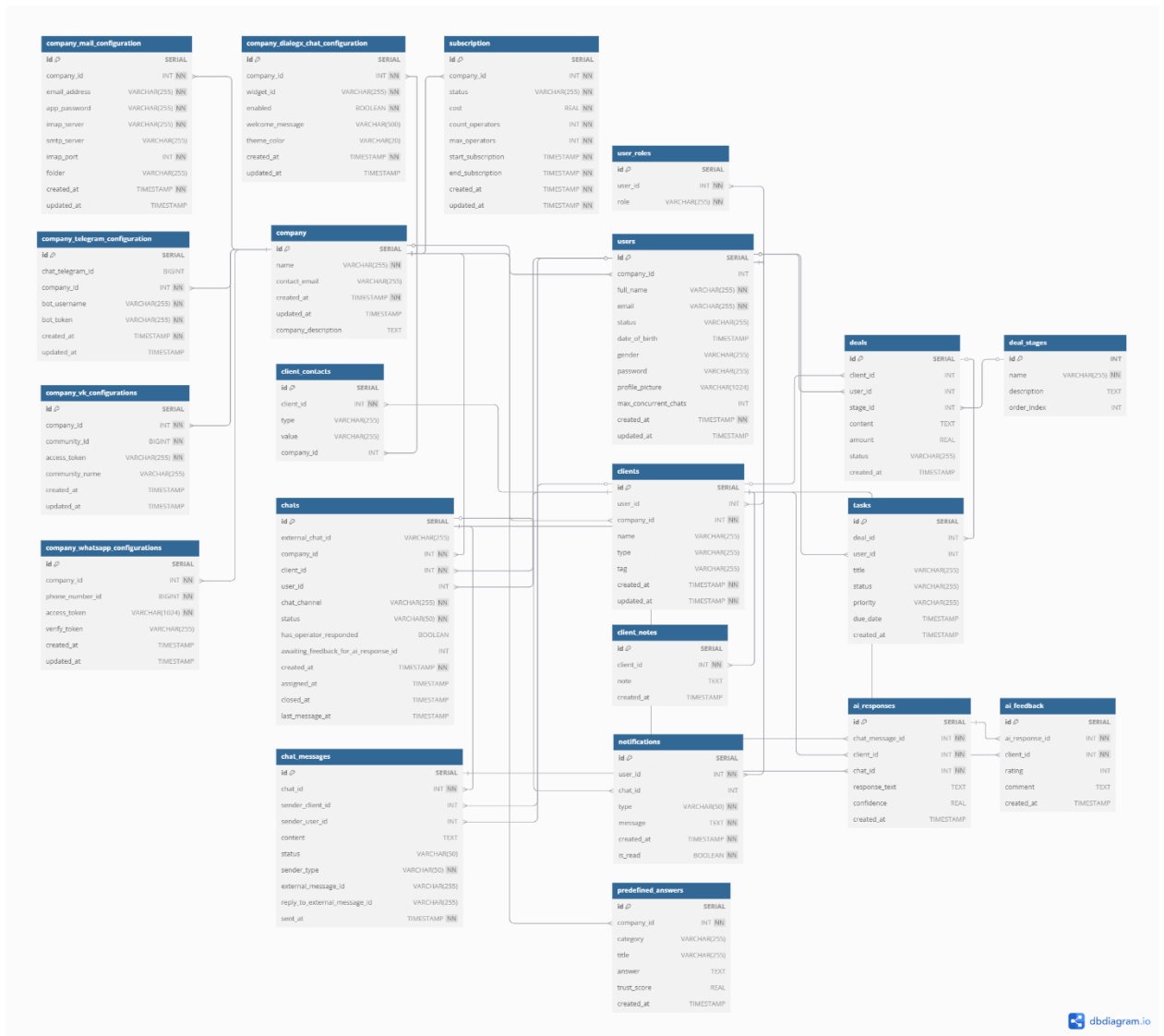


Рисунок А.1 — ER-диаграмма

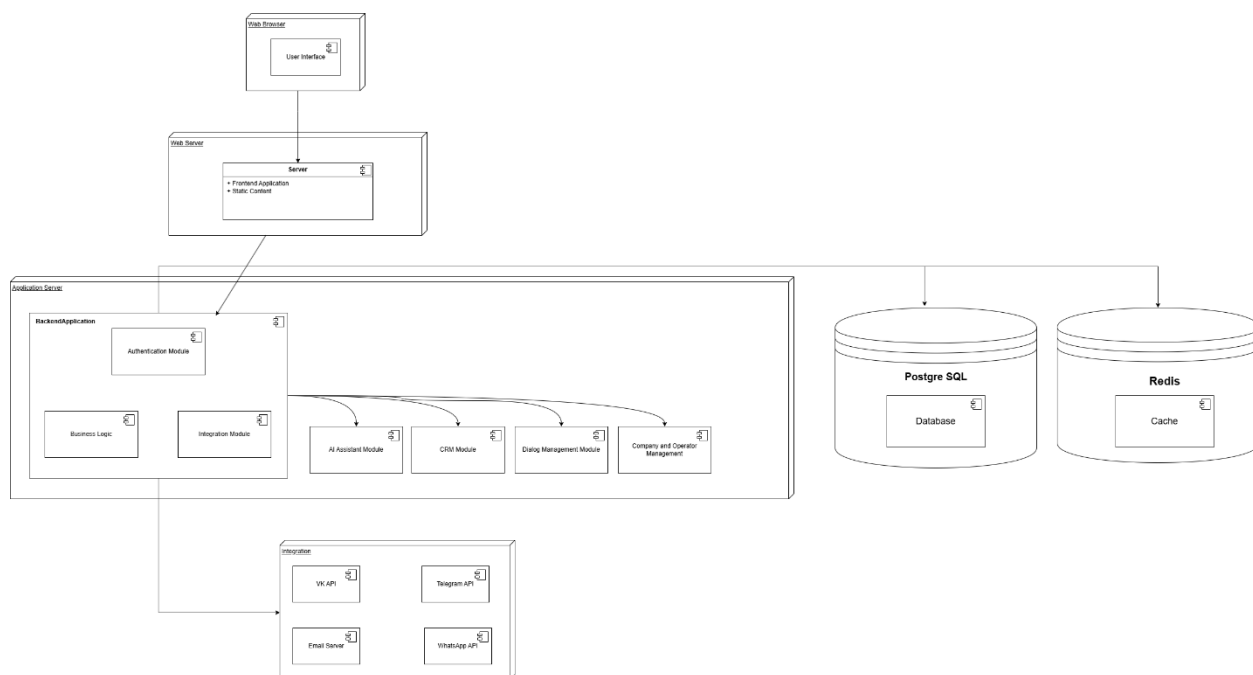


Рисунок А.2 — Диаграмма развертывания

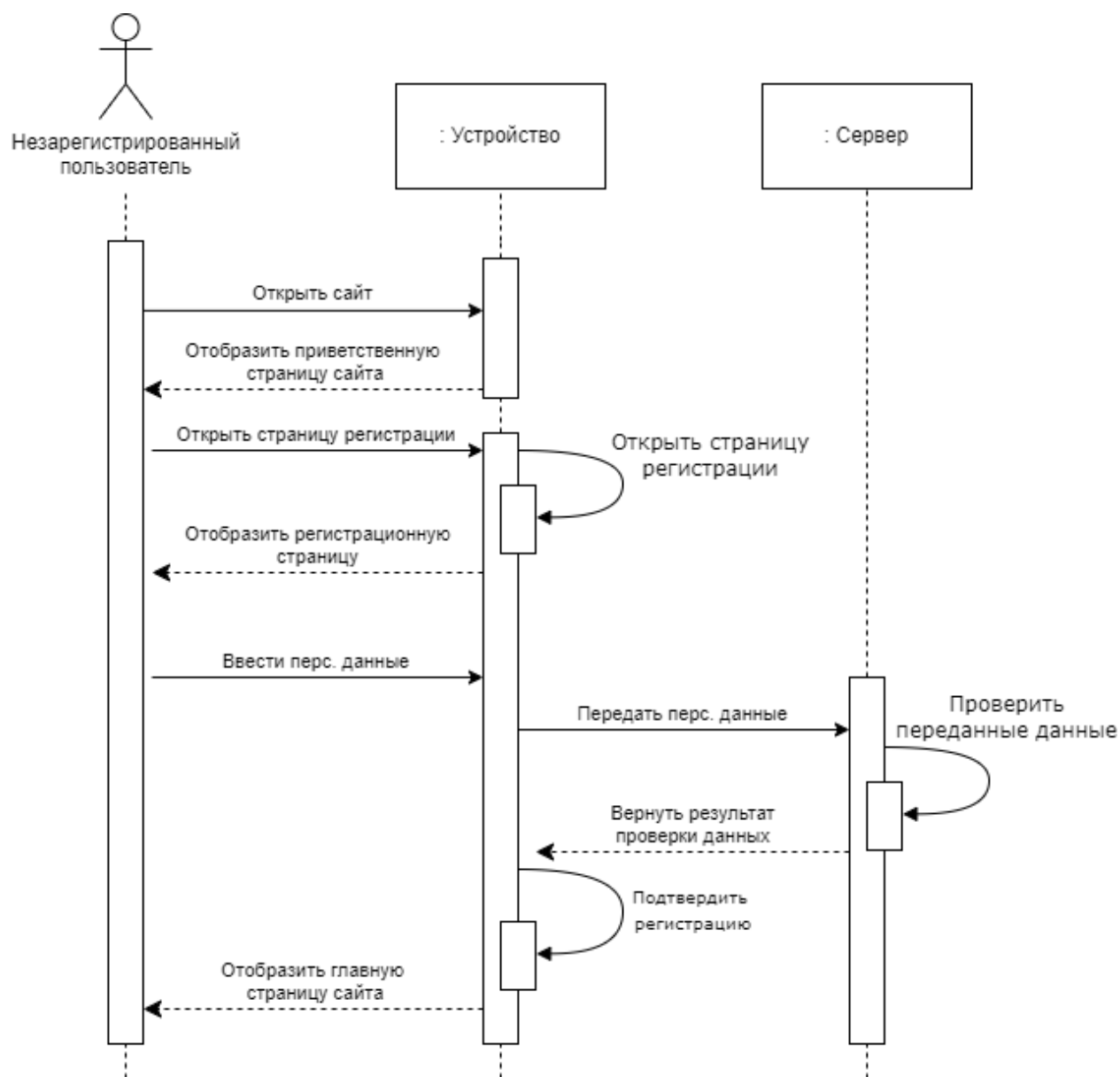


Рисунок А.3 — Диаграмма последовательности
(Незарегистрированный пользователь)

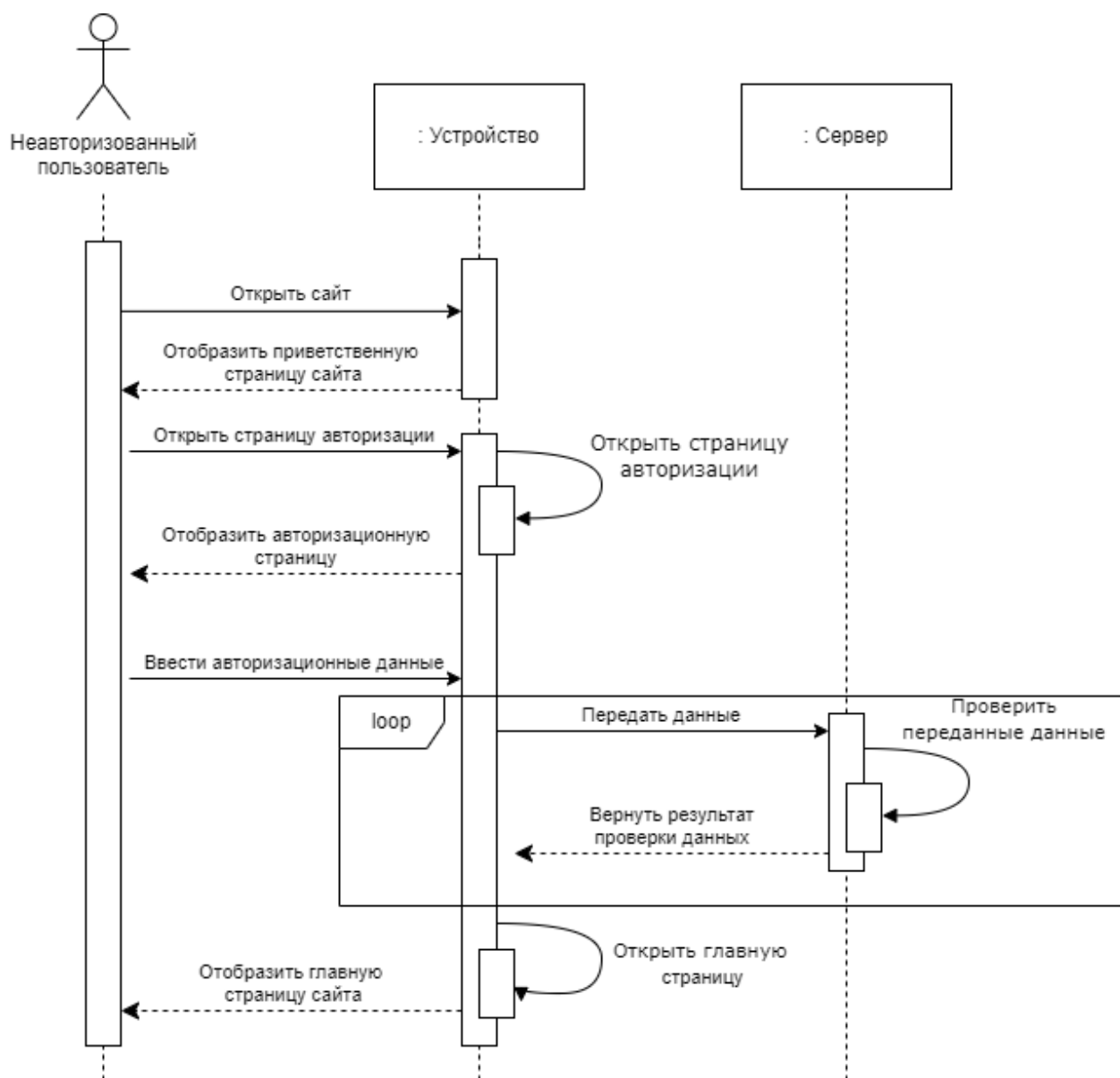


Рисунок А.4 — Диаграмма последовательности (Неавторизованный пользователь)

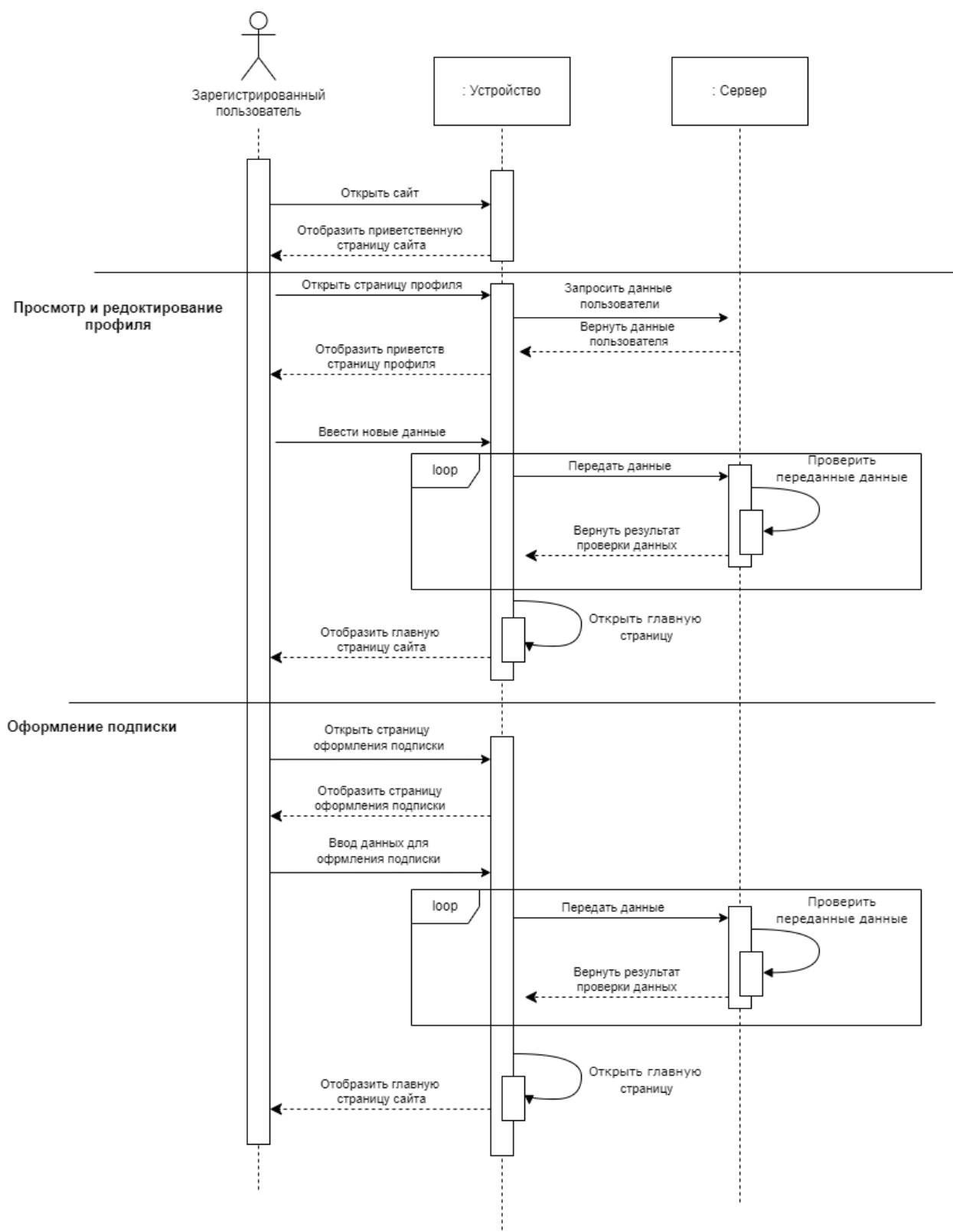


Рисунок А.5 — Диаграмма последовательности (Авторизованный пользователь)

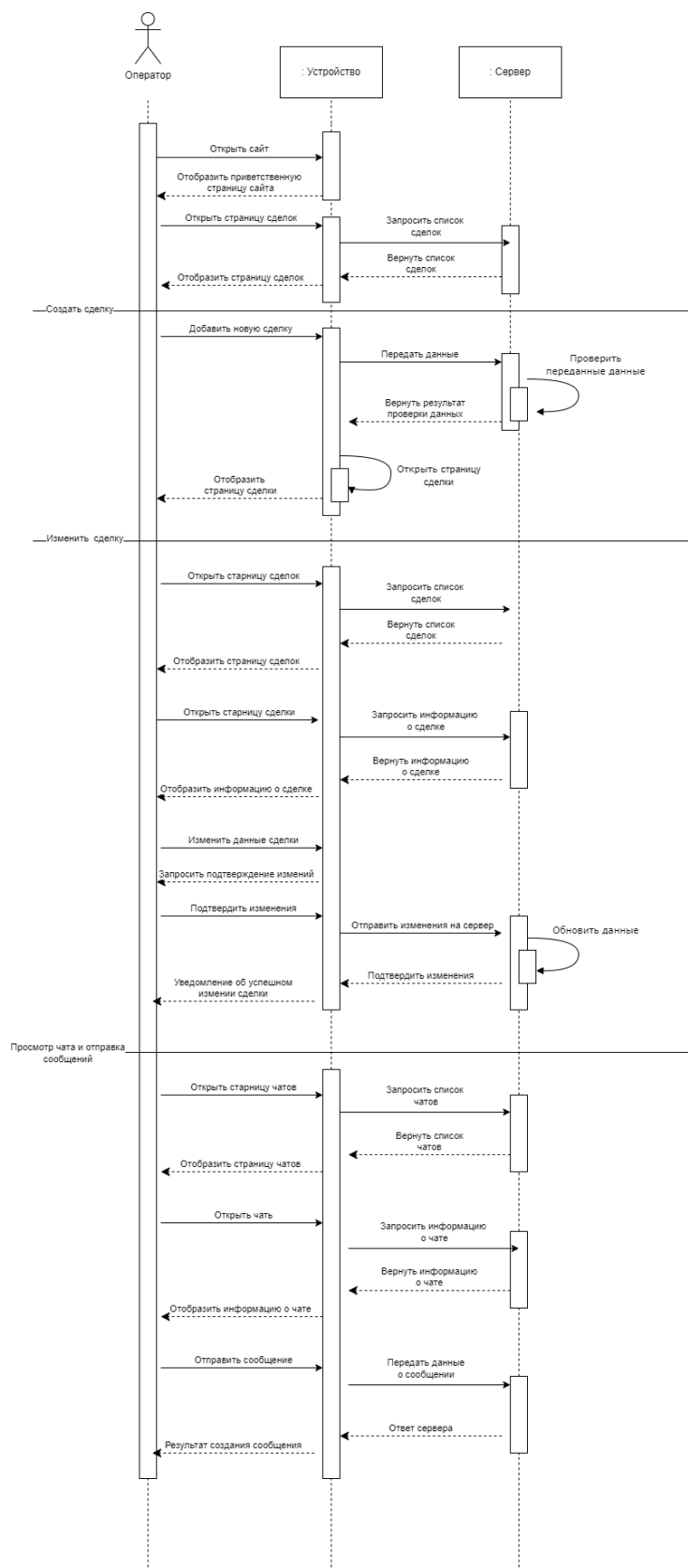


Рисунок А.6 — Диаграмма последовательности (Оператор)

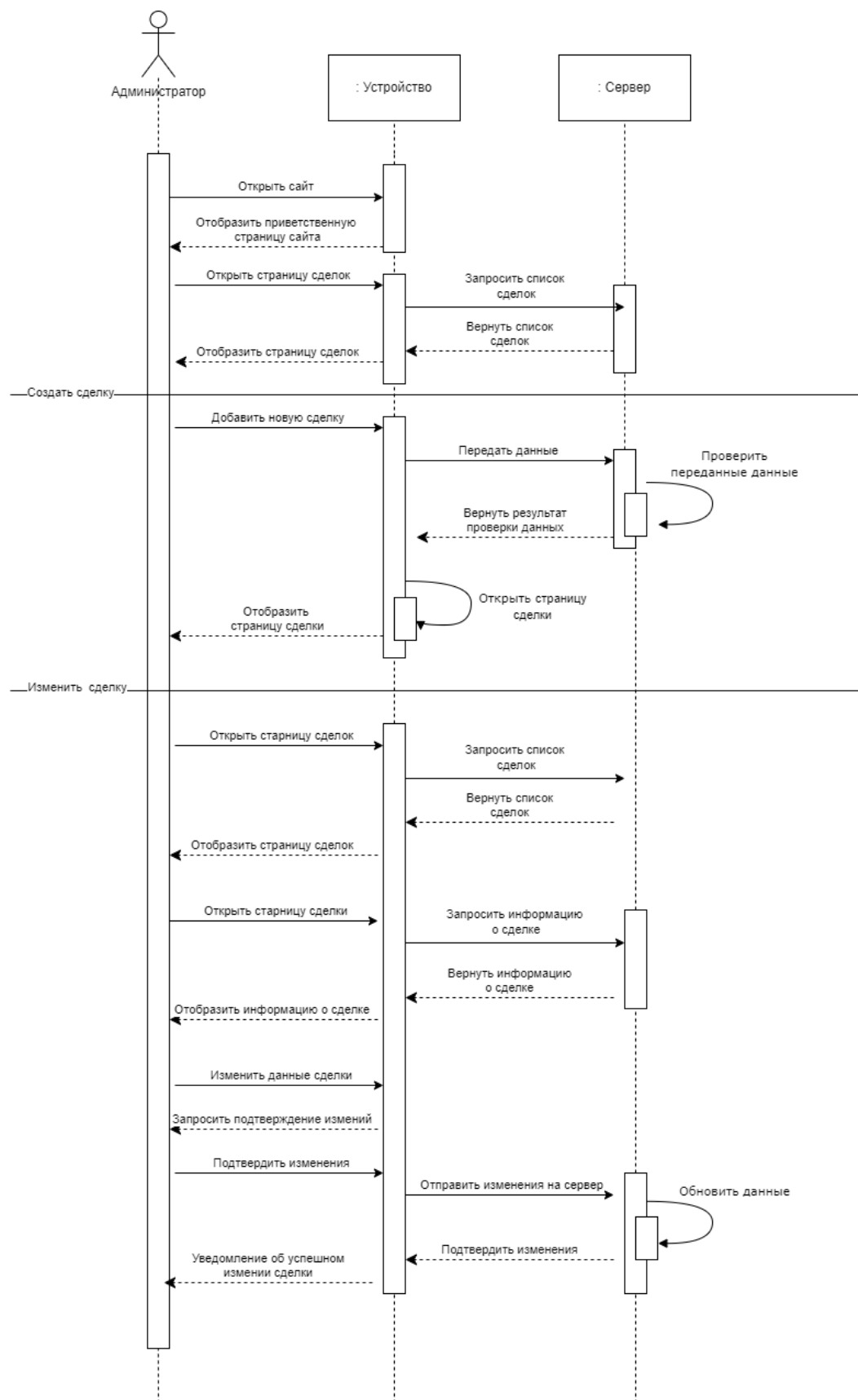


Рисунок А.7 — Диаграмма последовательности (Администратор, сделка)

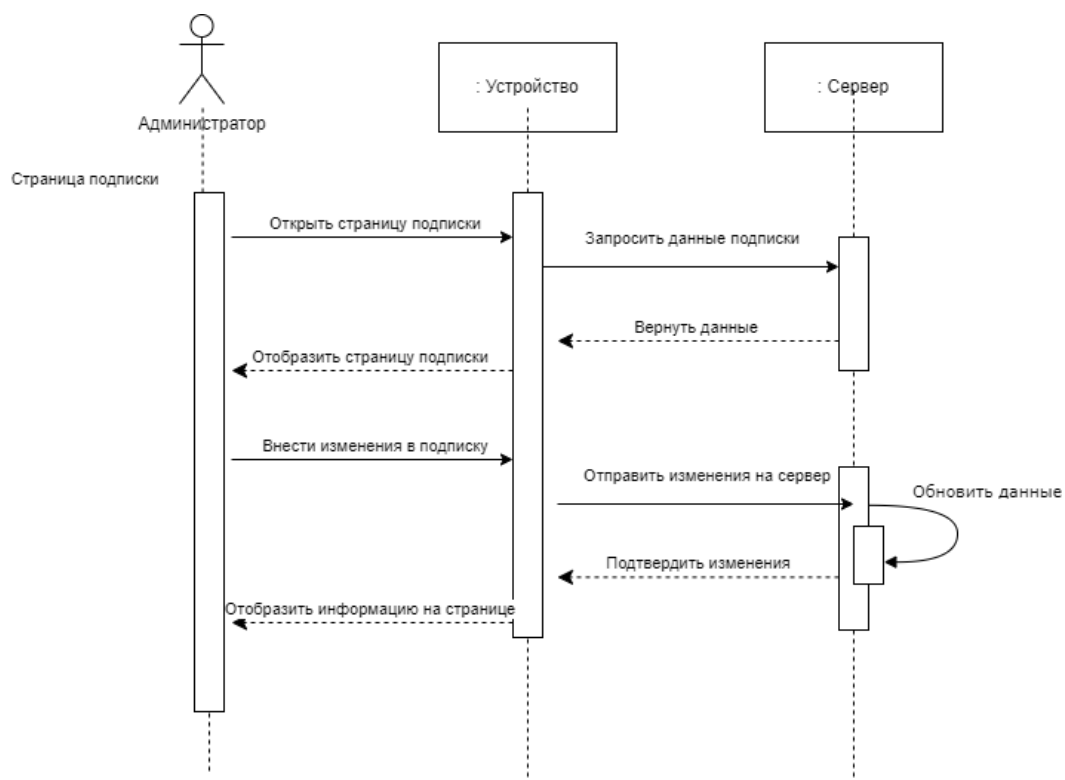


Рисунок А.8 — Диаграмма последовательности (Администратор, подписка)

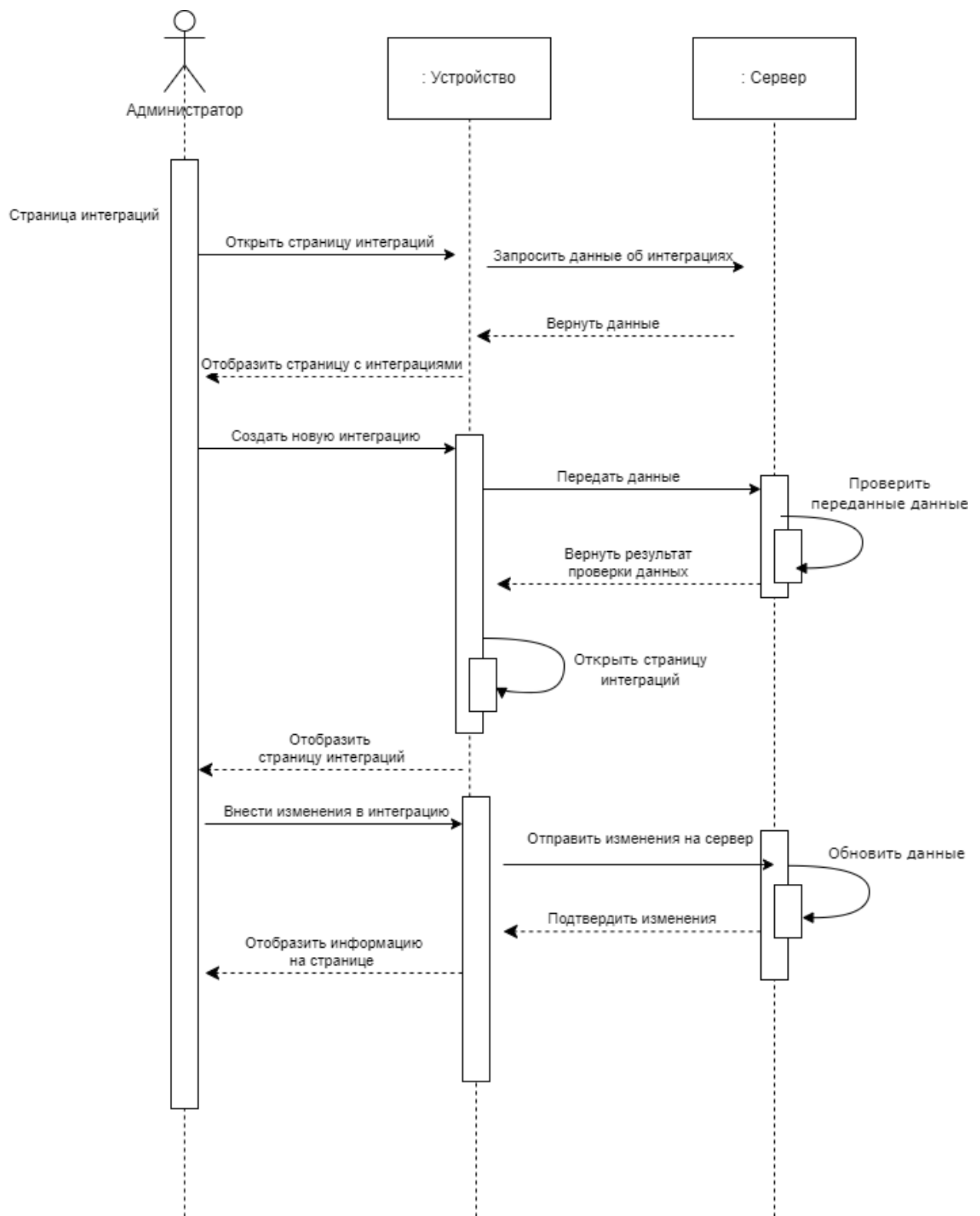


Рисунок А.9 — Диаграмма последовательности (Администратор, интеграции)

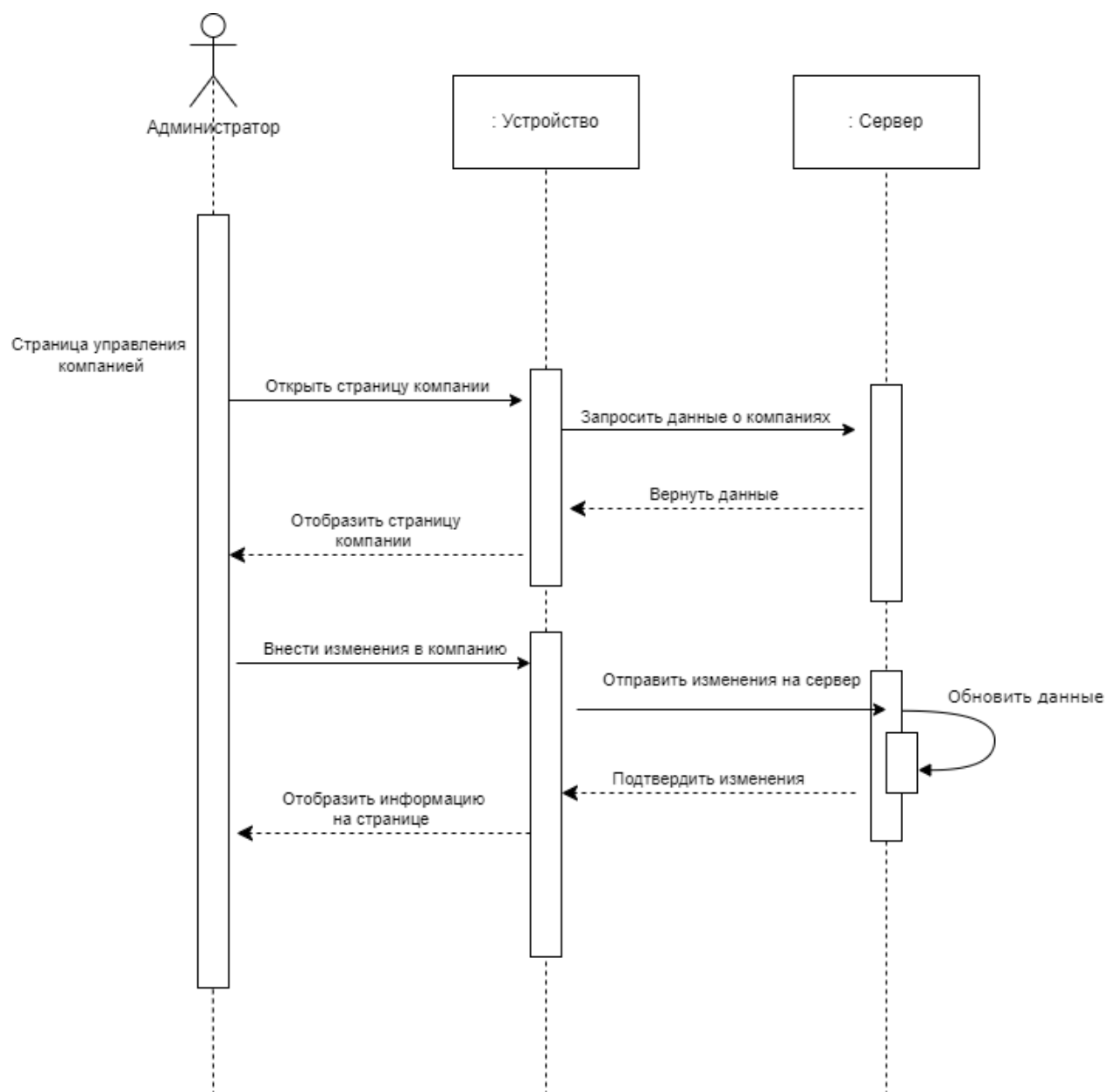


Рисунок А.10 — Диаграмма последовательности (Администратор, компания)

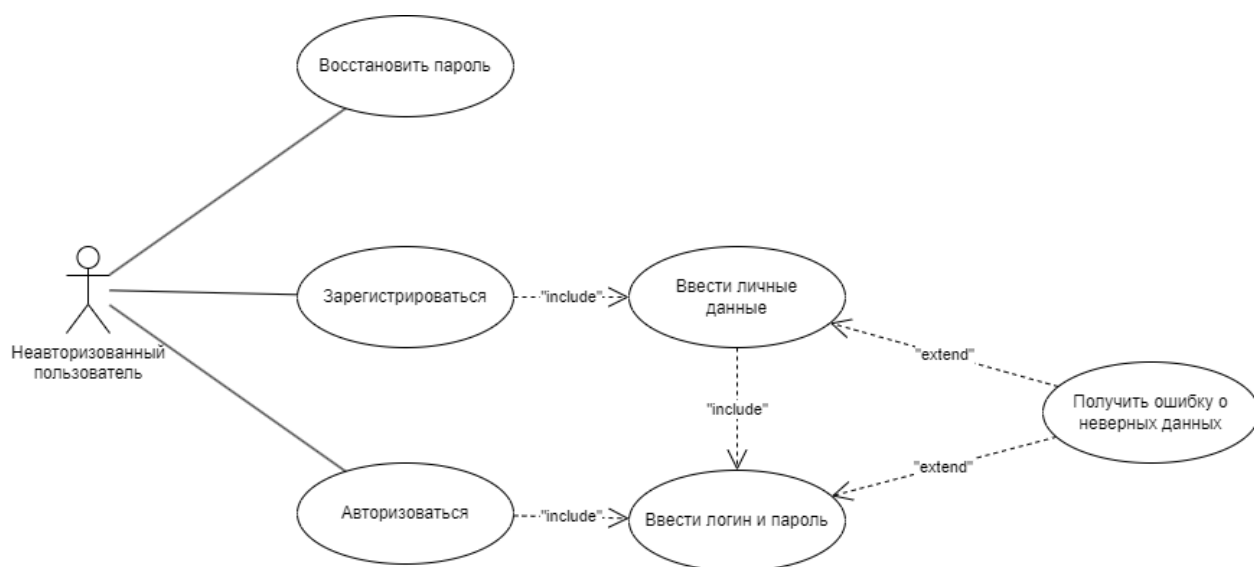


Рисунок А.11 — Диаграмма прецедентов (Неавторизованный пользователь)

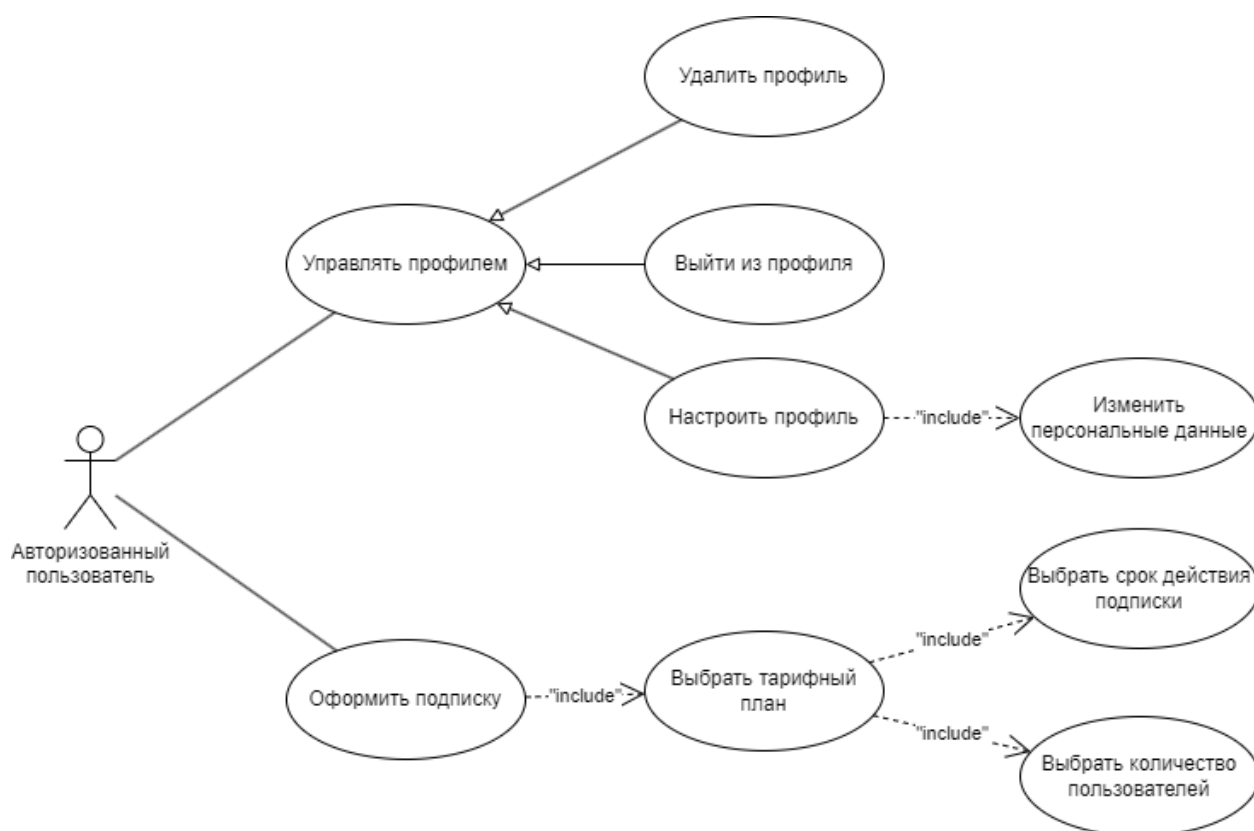


Рисунок А.12 — Диаграмма прецедентов (Авторизованный пользователь)



Рисунок А.13 — Диаграмма прецедентов (Оператор)

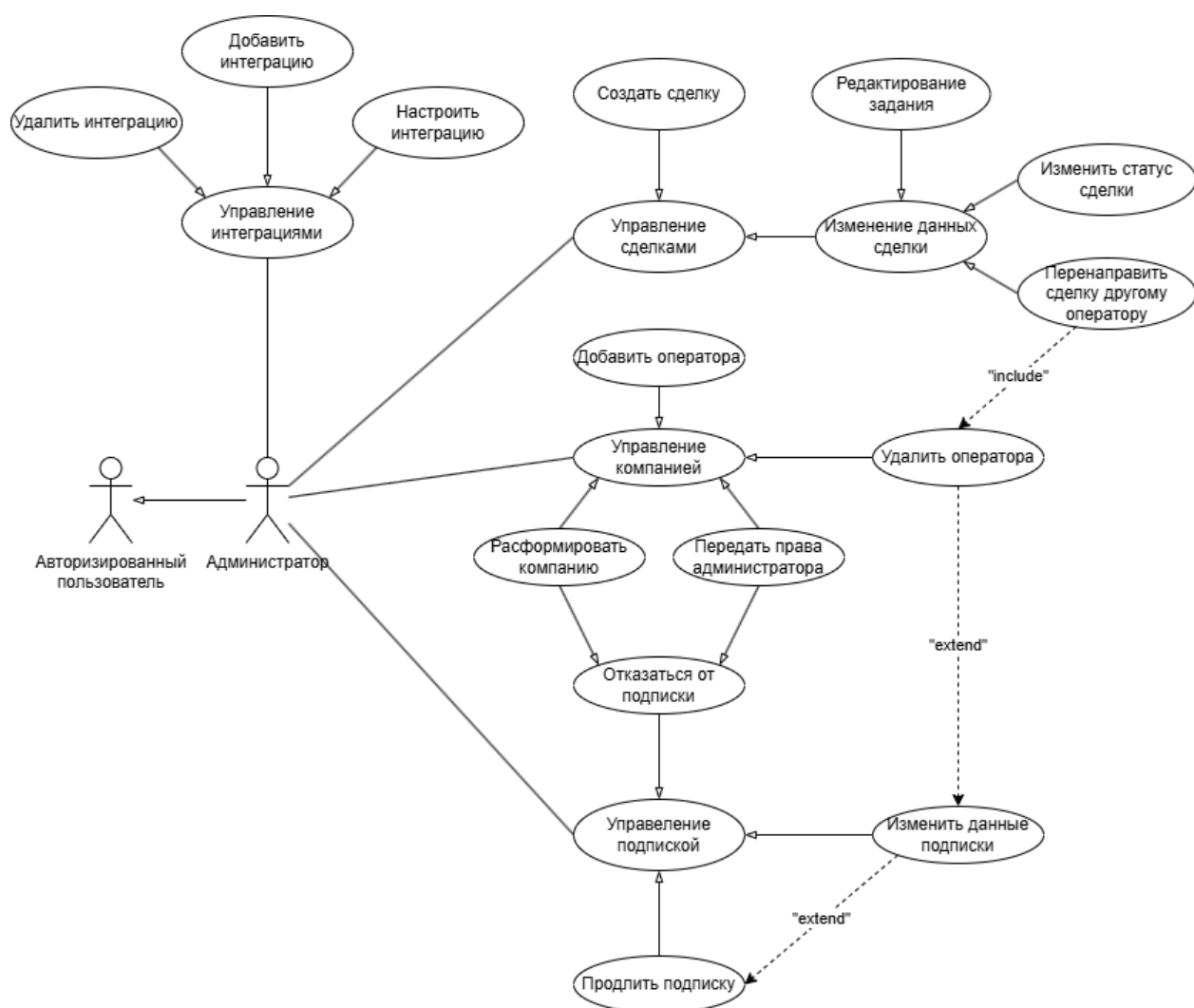


Рисунок А.14 — Диаграмма прецедентов (Администратор)



Рисунок А.15 — Диаграмма активности (Неавторизованный пользователь)

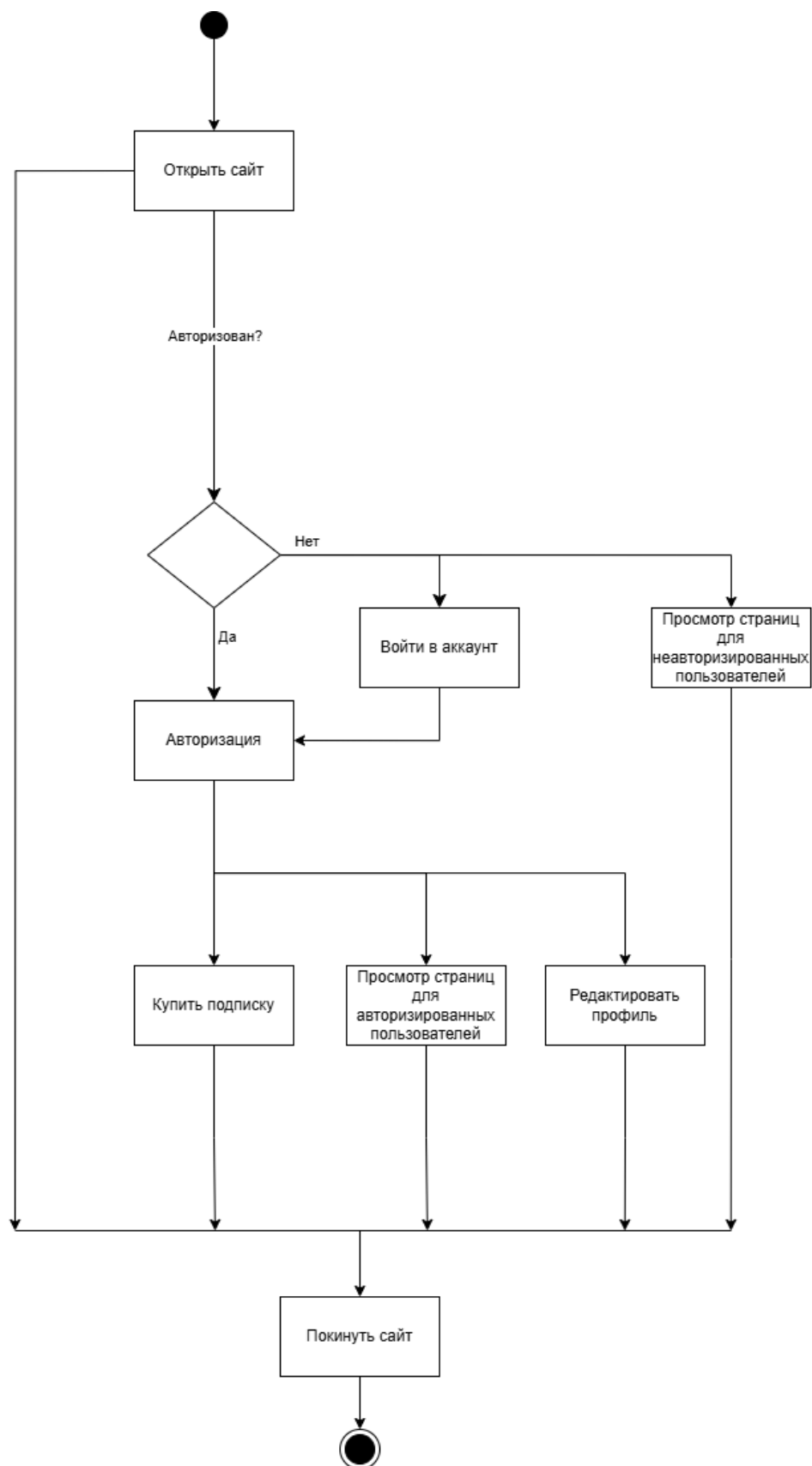


Рисунок А.16 — Диаграмма активности (Авторизованный пользователь)

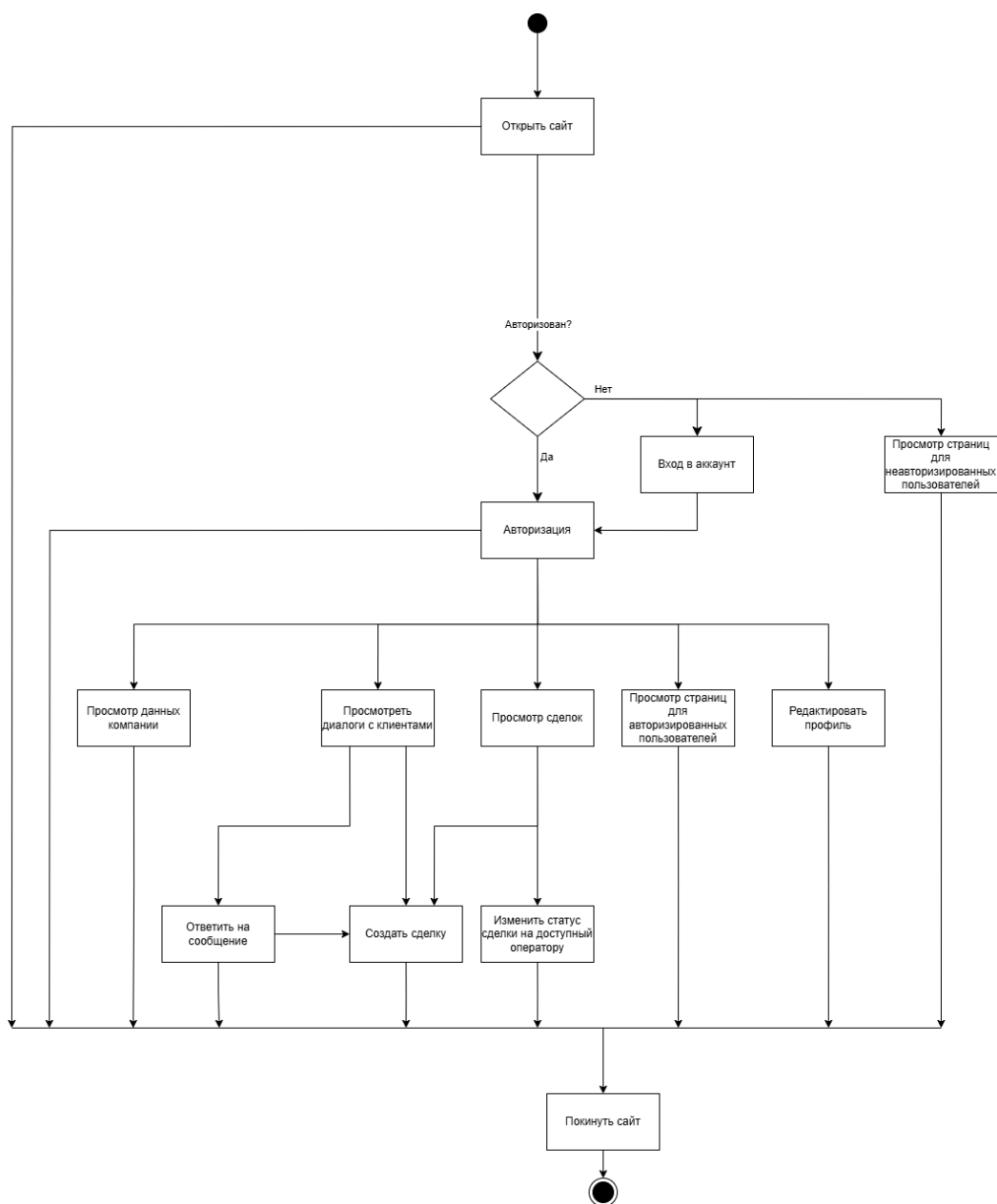


Рисунок А.17 — Диаграмма активности (Оператор)

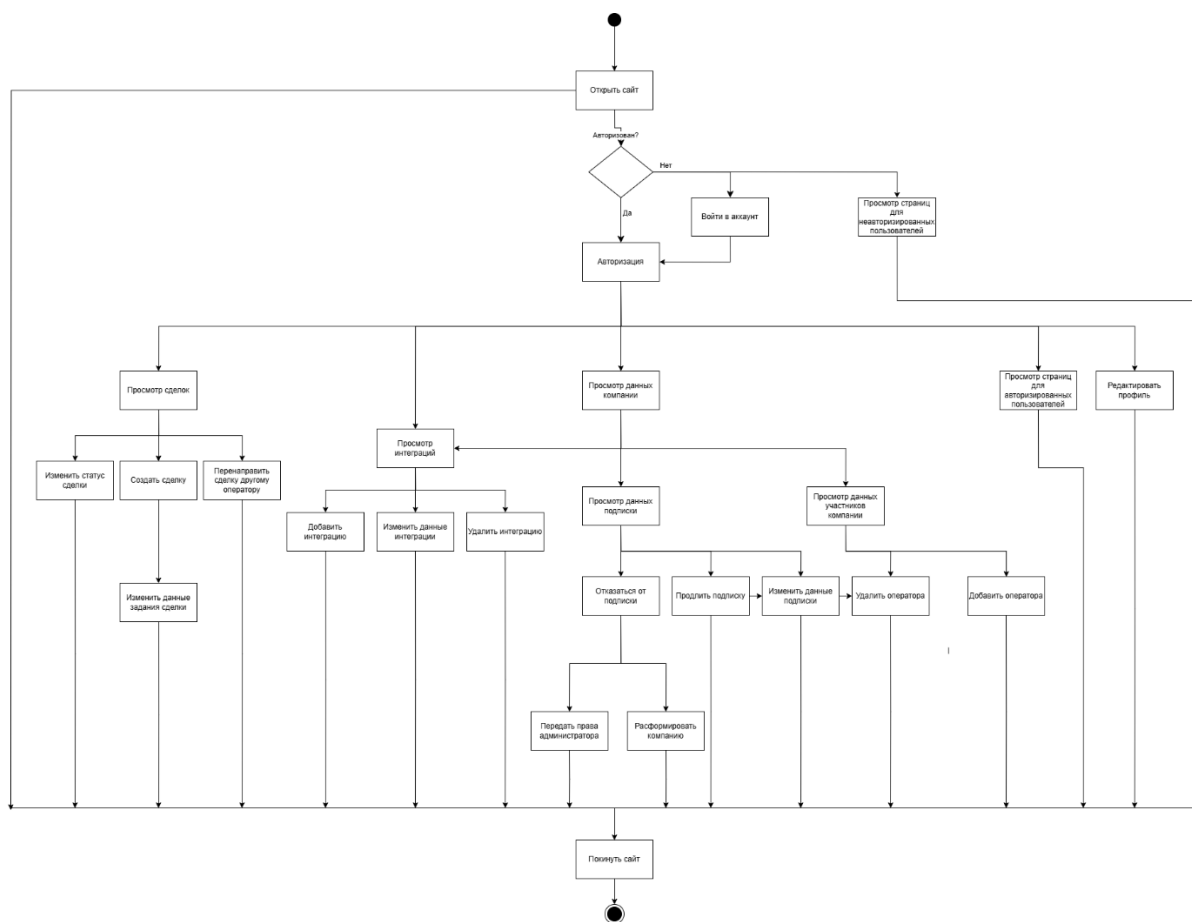


Рисунок А.18 —Диаграмма активности (Администратор)

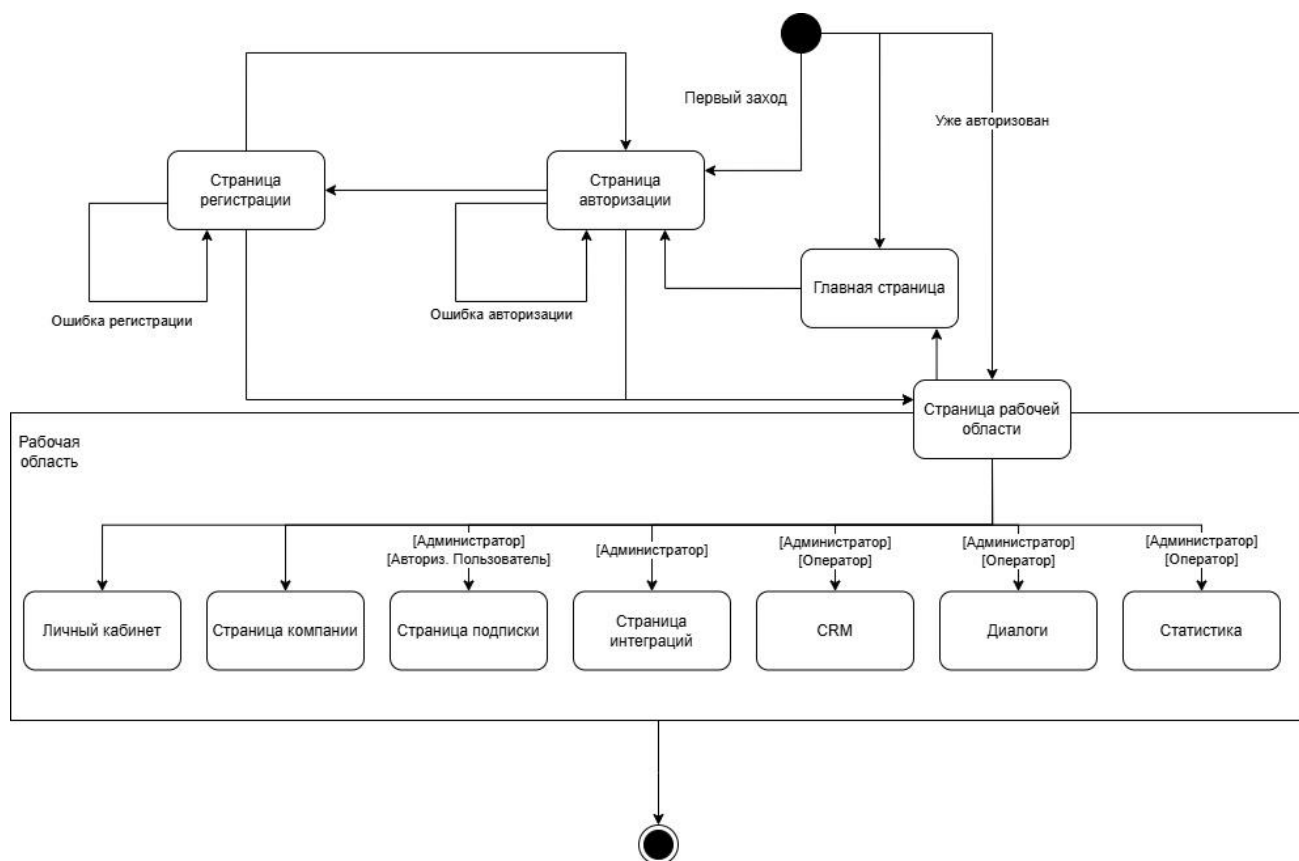


Рисунок А.19 — Диаграмма состояний