

МИНОБРНАУКИ РОССИИ  
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ  
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
«ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»  
(ФГБОУ ВО «ВГУ»)

Факультет Компьютерных наук  
Кафедра программирования и информационных технологий

Техническое задание  
на разработку веб-сайта  
«Автоответчик для почты и других чатов»

Исполнители

\_\_\_\_\_ Д. Е. Баранник  
\_\_\_\_\_ Н. Е. Флейшман  
\_\_\_\_\_ И. П. Сидоренко  
\_\_\_\_\_ В. А. Гончаров  
\_\_\_\_\_ И. И. Давыдов  
\_\_\_\_\_ В. А. Кузнецов

Заказчик

\_\_\_\_\_ В. С. Тарасов

Воронеж 2025

## Оглавление

1	Термины, используемые в техническом задании .....	4
2	Общие сведения .....	6
2.1	Наименование системы .....	6
2.2	Наименование исполнителя и заказчика приложения .....	6
2.2.1	Наименование заказчика .....	6
2.2.2	Наименование исполнителя .....	6
2.3	Перечень документов, на основании которых создается приложение .....	7
2.4	Состав и содержание работ по созданию приложения .....	7
2.5	Порядок оформления и предъявления заказчику результатов работ по созданию приложения .....	8
2.6	Цели и назначение создания web-сайта .....	8
3	Анализ конкурентов .....	10
3.1	Angry.Space .....	10
3.2	Раст .....	11
4	Требования к приложению и программному обеспечению .....	11
4.1	Требования к базе данных .....	11
4.2	Требования к архитектуре .....	12
4.2.1	Требования к средствам реализации .....	12
4.3	Требования к защите информации .....	13
4.4	Требования по патентной части .....	14
4.5	Требования к интеграциям .....	14
4.6	Требования для внедрения чата на сайт компании .....	14
5	Функциональные требования .....	15
5.1	Функциональные требования неавторизованного пользователя .....	15
5.2	Функциональные требования авторизованного пользователя .....	15
5.3	Функциональные требования оператора .....	16
5.4	Функциональные требования администратора .....	16
6	Нефункциональные требования .....	17
6.1	Макет приложения .....	18

6.1.1	Макет экрана авторизации.....	18
6.1.2	Макет экрана регистрации.....	19
6.1.3	Макет экрана информационной страницы.....	20
6.1.4	Макет экрана выбора тарифа подписки .....	21
6.1.5	Макет экрана интеграций .....	22
6.1.6	Макет экрана диалога.....	23
6.1.7	Макет экрана шаблонных ответов .....	24
6.1.8	Макет экрана сделки .....	25
6.1.9	Макет экрана статистики .....	26
7	Планы на дальнейшее развитие проекта .....	27
8	Источники разработки .....	28
	Приложение.....	29

## 1 Термины, используемые в техническом задании

Термин	Определение термина
Авторизация	Предоставление определённого 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	Человек, имеющий доступ к расширенному функционалу веб-сервиса, имеющий знания о формате приема статей
Неавторизованный пользователь	Пользователь, который не прошел процесс авторизации
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
Профиль (в веб-приложении)	Персональная запись пользователя, где хранится информация, необходимая для взаимодействия с ресурсом.
Сделка	Это запись о взаимодействии с клиентом, включающая этапы, сумму и другие детали процесса.

<b>Термин</b>	<b>Определение термина</b>
СУБД	Комплекс программ, позволяющих создать базу данных и манипулировать данными (вставлять, обновлять, удалять и выбирать).
Фреймворк	Заготовка, готовая модель в программировании для быстрой разработки, на основе которой можно дописать собственный код
Чат	Средство обмена различной информацией по компьютерной сети в режиме реального времени, а также программное обеспечение, позволяющее организовывать такое общение
API	Программный интерфейс, то есть описание способов взаимодействия одной компьютерной программы с другими программами
Front-end	Презентационная часть информационной или программной системы, ее пользовательский интерфейс и связанные с ним компоненты
PostgreSQL	Свободная объектно-реляционная система управления базами данных
REST	Архитектурный стиль взаимодействия компонентов распределенного приложения в сети
Back-end	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 Общие сведения**

### **2.1 Наименование системы**

Полное наименование приложения: «Автоответчик для почты и других чатов».

Краткое наименование: «DialogX».

### **2.2 Наименование исполнителя и заказчика приложения**

#### **2.2.1 Наименование заказчика**

Заказчик: Старший преподаватель Тарасов Вячеслав Сергеевич. Воронежский Государственный Университет, Факультет компьютерных наук, кафедра Программирования и Информационных Технологий.

#### **2.2.2 Наименование исполнителя**

Разработчик: команда «4» группы «10».

Состав команды разработчика:

- Флейшман Никита Евгеньевич (team lead, backend разработчик);
- Баранник Данил Евгеньевич (аналитик, технический писатель, product-менеджер);
- Сидоренко Игорь Павлович (backend разработчик).
- Гончаров Вадим Александрович (дизайнер).
- Давыдов Игорь Игоревич (frontend-разработчик).
- Кузнецов Валерий Александрович (тестировщик).

## **2.3 Перечень документов, на основании которых создается приложение**

Данное веб-приложение будет создаваться на основании следующих документов:

- закона РФ от 07.02.1992 N 2300–1 (ред. от 11.06.2021) "О защите прав потребителей";
- федерального закона "О персональных данных" от 27.07.2006 N 152-ФЗ.

## **2.4 Состав и содержание работ по созданию приложения**

Состав и содержание работ по созданию приложения включают в себя следующие этапы:

- Сбор необходимой информации, постановка целей, задач системы, которые в будущем должны быть реализованы 16.02.25 – 01.03.25;
- Анализ предметной области, анализ конкурентов и построение структуры требований, ведущих к решению поставленных задач и целей 01.03.25 – 16.03.25;
- Построение модели программы, описание спецификаций данных, определение связей между сущностями, разработка модели БД 16.03.25 – 01.04.25;
- Разработка рабочего проекта, состоящего из написания программного кода, отладки и корректировки кода программы 01.04.25 – 16.05.25;
- Проведение тестирования программного обеспечения 16.05.25 – 01.06.25.

## **2.5 Порядок оформления и предъявления заказчику результатов работ по созданию приложения**

Предварительные отчёты по работе будет проводиться во время рубежных аттестаций:

- 1 аттестация (конец марта 2025) – создан репозиторий проекта на GitHub, распределены задачи проекта в таск-трекер Jira, создан проект с общей логикой системы, предоставлены промежуточные результаты по курсовому проекту и готовое техническое задание;
- 2 аттестация (конец апреля 2025) – написана основополагающая часть программного кода приложения, которая содержит большинство требуемого функционала, реализована БД и ее взаимодействие с сервером, проведена отладка и доработка кода, проведено тестирование по работе системы;
- 3 аттестация (конец мая 2025) – разработан курсовой проект, выполнены завершающие работы по доработке приложения, предоставлена готовая система.

## **2.6 Цели и назначение создания web-сайта**

Цель проекта заключается в создании системы автоответчика, которая поможет пользователям автоматизировать и упростить коммуникации с клиентами, а заказчику – повысить эффективность и доходность бизнеса, улучшить клиентский сервис и привлечь новых пользователей и инвесторов.

Web-сайт позволяет решать следующие задачи:

- выполнять редактирование данных своего аккаунта после авторизации или регистрации;
- добавлять, редактировать, удалять подписку администратором;



- возможность оператора просматривать диалоги и отвечать на сообщения клиентов;
- возможность оператора и администратора создавать, просматривать редактировать сделку;
- возможность администратора редактировать, продлить, аннулировать подписку;
- возможность администратора просматривать, добавлять, редактировать и удалять интеграции;
- возможность администратора добавлять, удалять операторов.

### 3 Анализ конкурентов

В современном мире бизнесу важно быстро и эффективно общаться с клиентами. Автоответчики помогают автоматизировать процесс, сокращая время ожидания ответа и улучшая качество обслуживания. Рынок систем автоответчиков предлагает множество решений для автоматизации клиентской поддержки и управления входящими запросами.

В этом многообразии выделяются два проекта, которые занимают значительно место среди автоответчиков – Pact и Angry.Space.

#### 3.1 Angry.Space

Angry.Space – система управления клиентскими обращениями из соцсетей и мессенджеров, которая объединяет потоки из нескольких социальных сетей, улучшает процесс обработки обращений, поступающих из этих каналов.

Преимущества:

- Широкий список поддерживаемых соцсетей.
- Хорошие аналитические инструменты для контроля работы операторов.
- Возможность блокировать сотрудников, не теряя историю переписки.

Недостатки:

- Ограниченное время хранения данных (всего 3 месяца в базовом тарифе).
- Нет интеграции с CRM, что усложняет работу с клиентской базой.
- Отсутствие AI ответов.
- Отсутствие возможности подключить email.

### **3.2 Ract**

Ract – агрегатор мессенджеров и соцсетей для бизнеса. Позволяет объединить несколько каналов связи в едином окне и наладить связь с клиентами через CRM-систему удобным для них способом.

Преимущества:

- Встроенная интеграция с AmoCRM и Bitrix24.
- Поддержка WhatsApp Business API, что важно для компаний.
- Можно заранее создавать шаблоны ответов.

Недостатки:

- Дорогая подписка (от 990 Р/мес + минимальный депозит 5,000 Р).
- Тестовый период всего 3 дня, что не позволяет полноценно протестировать сервис.
- Сложная структура тарифов, где каждая дополнительная функция увеличивает стоимость.
- Отсутствие возможности подключить email и AI-ответы.

## **4 Требования к приложению и программному обеспечению**

### **4.1 Требования к базе данных**

База данных будет реализована с помощью PostgreSQL, кэширование будет реализовано с помощью Redis.

## 4.2 Требования к архитектуре

Приложение должно быть реализовано с применением архитектуры, соответствующей модели Клиент-Серверного взаимодействия на основе REST API.

### 4.2.1 Требования к средствам реализации

Для реализации серверной части сайта будут использоваться следующие средства:

- язык программирования Java (предлагает сочетание производительности, надёжности и богатой экосистемы инструментов и библиотек);
- Spring Boot (позволяет работать со встроенными модулями, которые легко интегрируются в приложения, добавляя функциональность и ускоряя разработку);
- Spring Security (предназначен для обеспечения защищенной аутентификации и авторизации в Java-приложениях);
- СУБД PostgreSQL (предоставляет расширенные возможности SQL, надежность и хорошую документацию);
- Docker (позволяет ускорить разработку, тестирование и развертывание приложения).
- В качестве AI-модели будет использоваться Qwen 2.5

Для реализации клиентской части приложения будут использоваться:

- язык разметки HTML (позволяет создавать и структурировать веб-страницы);

- язык стилей CSS (определяет, как именно должны отображаться элементы HTML, включая макет, цвета, шрифты и другие визуальные аспекты);
- язык программирования JavaScript (с его помощью сайты делают интерактивными: добавляют всплывающие окна, анимацию, кнопки лайков и формы для отправки информации);
- Tailwind CSS (CSS-фреймворк с открытым исходным кодом для быстрого создания и настройки приложений без написания пользовательских CSS).
- Vite (инструмент для настройки среды разработки и сборки проекта).
- React (библиотека для создания пользовательских интерфейсов).
- React Router (библиотека для навигации между разными частями веб-приложения, созданного на React, она позволяет менять содержимое страницы без перезагрузки браузера)

Для ведения документации:

- Swagger (поскольку позволяет делать спецификации для Rest API).

### **4.3 Требования к защите информации**

- обеспечение авторизации и аутентификации пользователей;
- использование механизмов защиты от SQL-инъекций;
- использование протокола передачи данных HTTPS.

#### **4.4 Требования по патентной части**

Приложение должно быть разработано в соответствии с законодательством об авторских правах и патентах, без нарушения соответствующих лицензий. В случае нарушения данного требования, полную ответственность за нарушение несет сторона, ответственная за разработку приложения.

#### **4.5 Требования к интеграциям**

Приложение должно поддерживать интеграцию со следующими сервисами:

- Email (пользователь должен предоставить адрес электронной почты, пароль для IMAP-доступа);
- Telegram (пользователь должен предоставить токен бота (bot token), имя бота (bot username));
- ВКонтакте (VK) (пользователь должен предоставить токен сообщества (access token), название сообщества);
- WhatsApp (Business API) (пользователь должен предоставить токен (API token), номер телефона).

#### **4.6 Требования для внедрения чата на сайт компании**

- Для внедрения виджета чата на сторонний сайт нашей системой генерируется уникальный API-ключ, который должен быть передан администратору сайта;
- Интеграция должна работать через Rest API для возможности обмена сообщениями в реальном времени.

## **5 Функциональные требования**

Приложение должно поддерживать функционал для различных пользователей:

- неавторизованный пользователь;
- авторизованный пользователь;
- оператор;
- администратор.

### **5.1 Функциональные требования неавторизованного пользователя**

Перечень функций:

- авторизация в системе;
- регистрация в системе;
- просмотр ознакомительной информации о сайте.

### **5.2 Функциональные требования авторизованного пользователя**

Данный функционал доступен авторизованным ролям.

Перечень функций:

- выход из системы;
- просмотр страницы профиля;
- редактирование данных профиля;
- просмотр страницы, содержащей информацию о подписке и тарифах;
- оформление подписки.

### **5.3 Функциональные требования оператора**

Имеет весь функционал авторизованного пользователя, а также:

- создать/изменить сделку;
- просмотреть чаты и ответить на сообщения

### **5.4 Функциональные требования администратора**

Имеет весь функционал от оператора, а также:

- просматривать/создавать/удалять/изменять интеграции;
- просматривать/оформлять/аннулировать/изменять подписку;
- добавлять и удалять операторов.



## 6 Нефункциональные требования

Web-сайт должен выполнять следующие нефункциональные требования:

- производительность: Сайт должен обеспечивать быструю загрузку и отзывчивость интерфейса (при стабильном интернет-соединении), даже при большой активности пользователей, а также масштабироваться для работы с большим количеством пользователей;
- безопасность: Данные пользователей должны быть защищены от несанкционированного доступа, будет использоваться защищенный тип соединения;
- надежность: Сайт должен быть стабильным и надежным, минимизируя возможность сбоев и потери данных;
- 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);
- документация и поддержка: Приложение должно предоставлять структурированную документацию, охватывающую аспекты приложения, включая функциональность, API, компоненты системы и бизнес-процессы. Документация должна быть полной, ясной и актуальной, с примерами использования и иллюстрациями, чтобы обеспечить легкость понимания как пользователями, так и разработчиками.

## 6.1 Макет приложения

### 6.1.1 Макет экрана авторизации

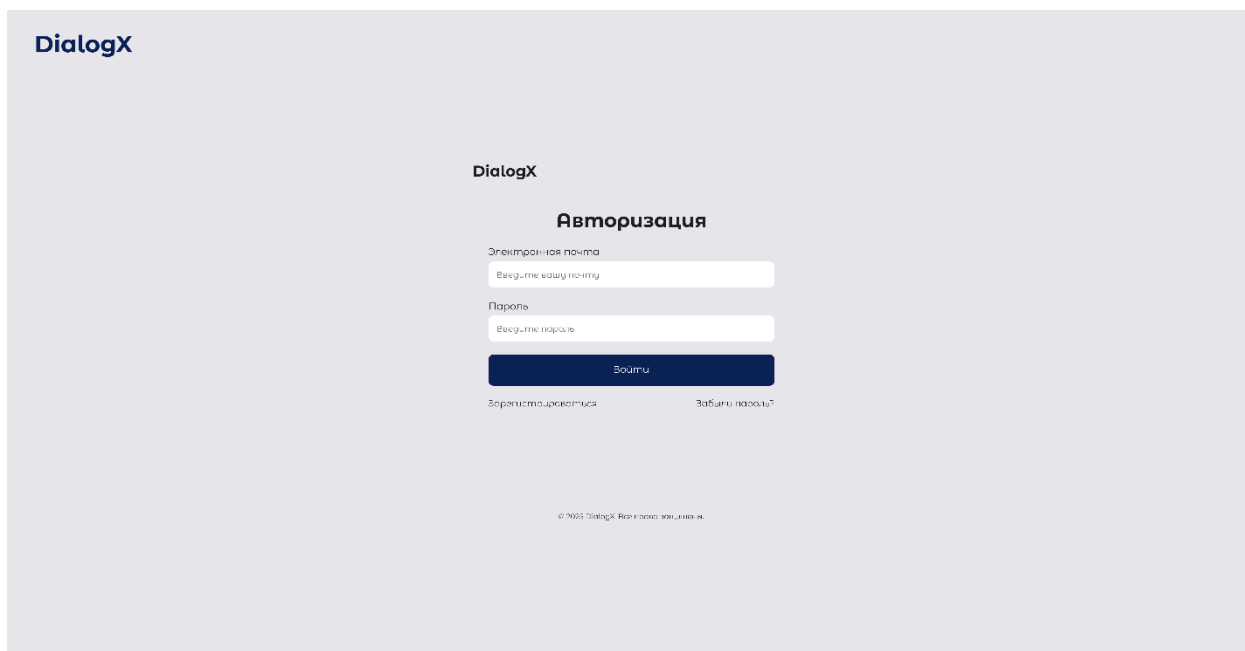


Рисунок 1 - Макет экрана авторизации

На этом экране отображаются поля с вводом адреса электронной почты и пароля от аккаунта, кнопка для входа, кнопка для регистрации и кнопка для восстановления пароля. При вводе некорректных данных пользователю будет выведено сообщение об ошибке, указывающее на отсутствие профиля с такими данными.

## 6.1.2 Макет экрана регистрации

Рисунок 2 - Макет экрана регистрации

На данном экране находятся поля для ввода почты и пароля вместе с кнопкой для регистрации. При вводе некорректного формата данных пользователю будет выведено сообщение об ошибке, уведомляющее о некорректно введенных данных.

## 6.1.3 Макет экрана информационной страницы

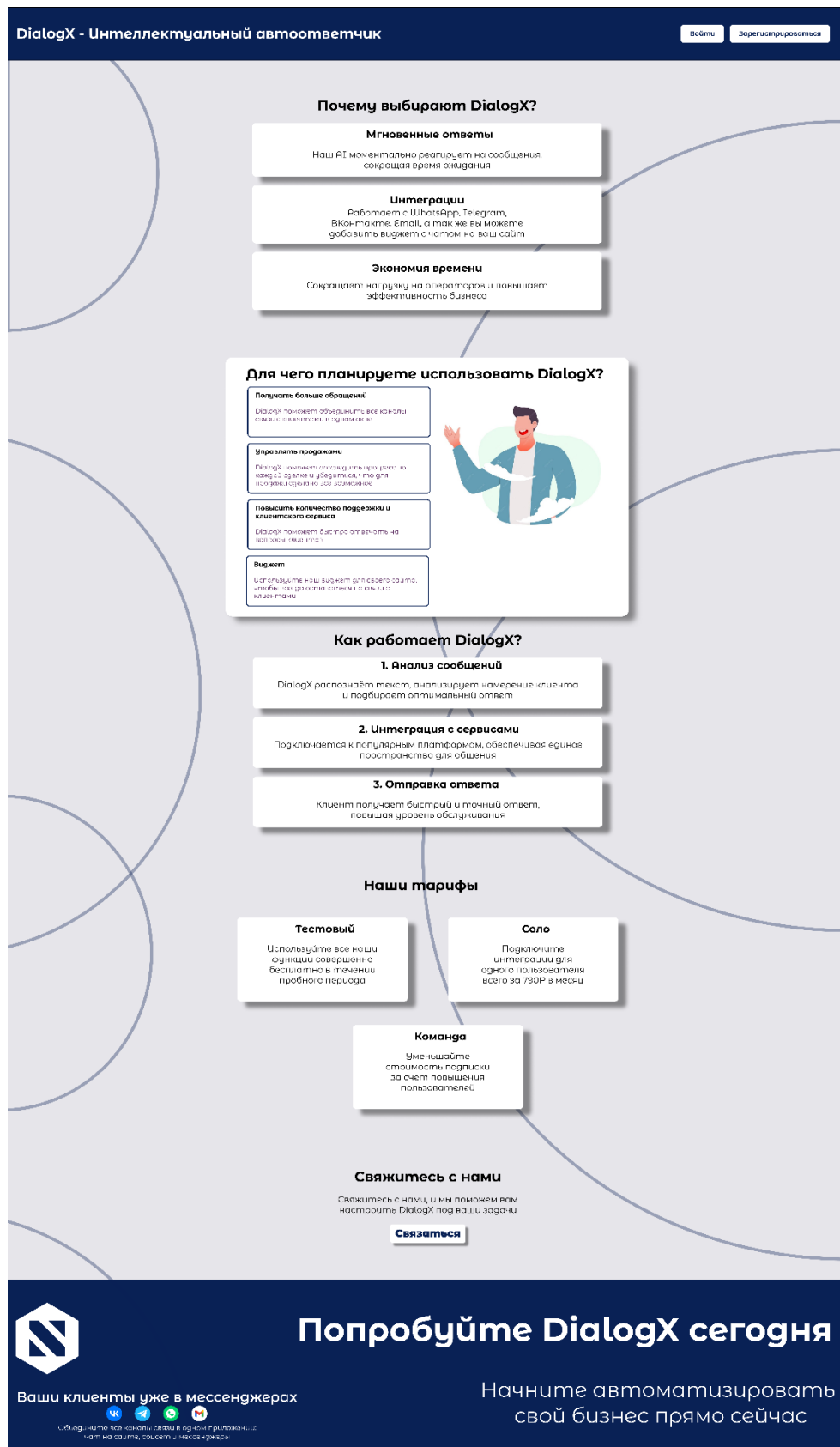


Рисунок 3 - Макет экрана информационной страницы

На данном экране пользователь может ознакомиться с возможностями сайта, преимуществами, принципами работы, а также информацией о тарифах. На экране присутствуют кнопки для авторизации и регистрации, нажимая на которые пользователь попадает на страницу авторизации и регистрации соответственно.

#### 6.1.4 Макет экрана выбора тарифа подписки

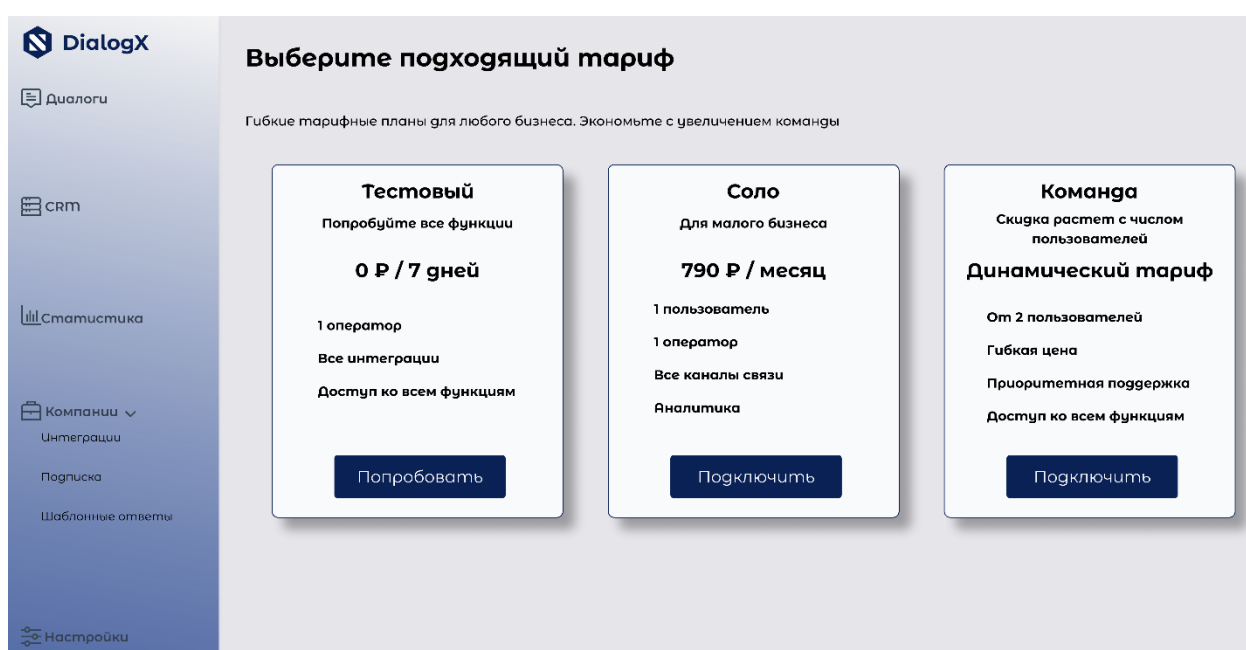


Рисунок 4 - Макет экрана выбора тарифа подписки

На данном экране пользователь может ознакомиться с доступными тарифными планами, их стоимостью, преимуществами и условиями. Под платными тарифами находится кнопка “Оформить”, а также под пробным бесплатным тарифом находится кнопка “Попробовать” по которым происходит оформление подписки. Данная страница помогает пользователю сравнить тарифы, выбрать тариф и перейти к оформлению подписки.

## 6.1.5 Макет экрана интеграций

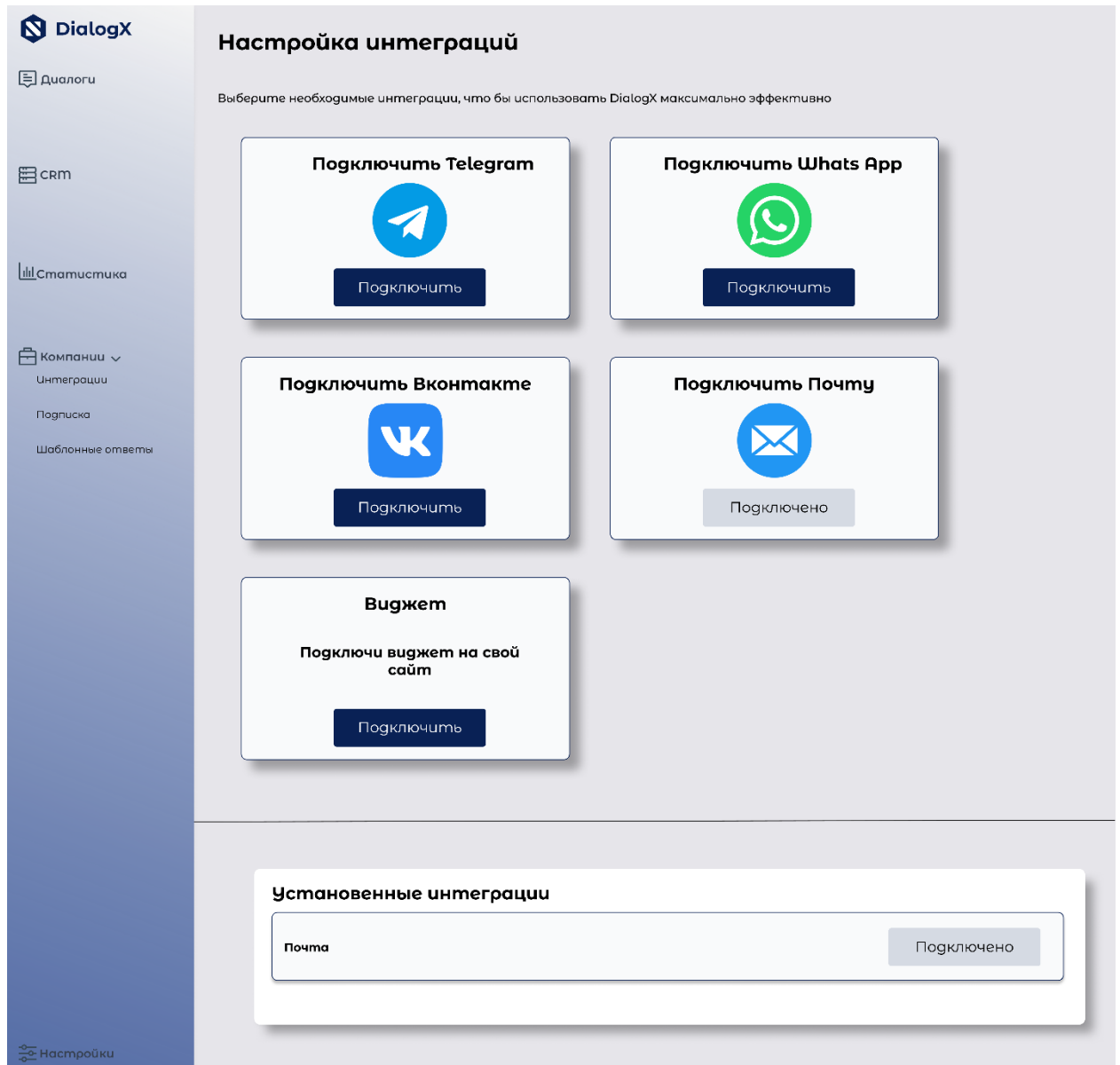


Рисунок 5 - Макет экрана интеграций

На данном экране пользователь может подключить интеграции с мессенджерами, почтой, а также виджет чата на свой сайт. Каждая интеграция представлена в виде блока с кнопкой подключения. В разделе установленных интеграций отображаются уже подключенные сервисы (в данном случае- “Почта”).

## 6.1.6 Макет экрана диалога

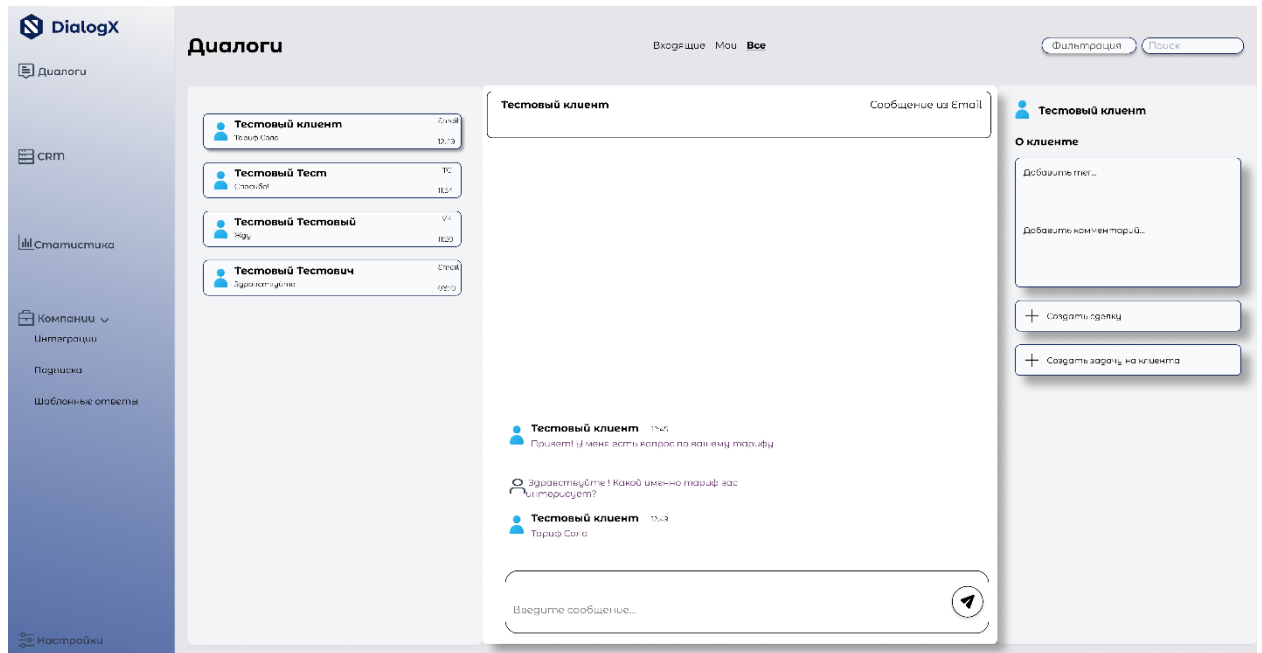


Рисунок 6 - Макет экрана диалога

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

## 6.1.7 Макет экрана шаблонных ответов

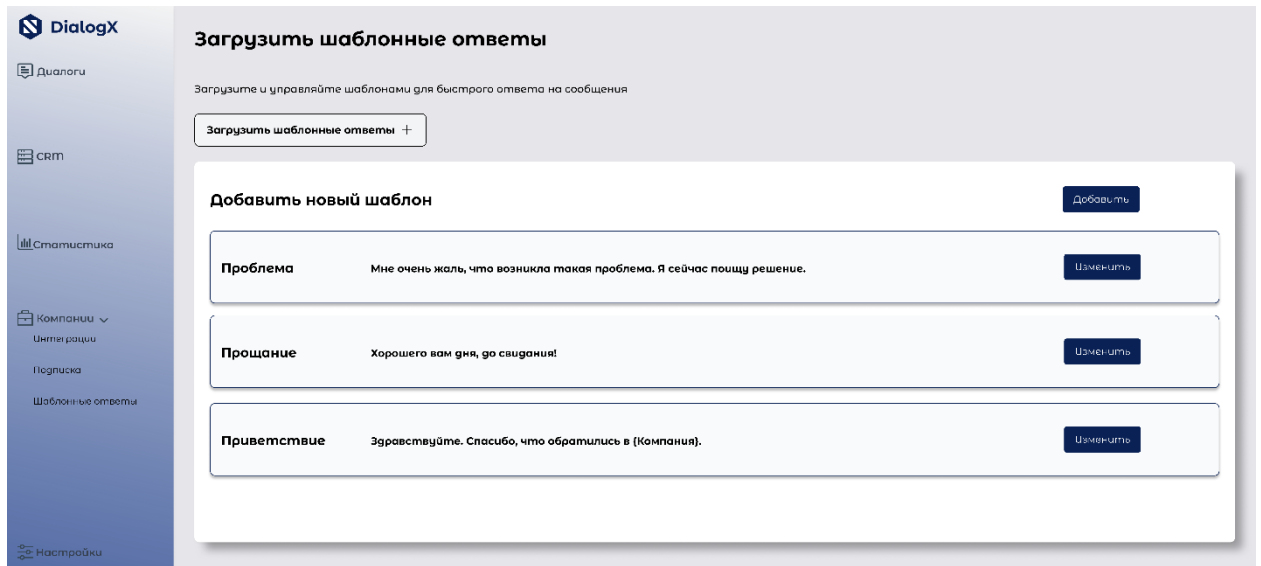


Рисунок 7 - Макет экрана шаблонных ответов

На данном экране можно создавать, загружать, изменять, удалять шаблонные ответы. Блок “Добавить новый шаблон” позволяет добавить новые варианты шаблонных ответов. Блок управления шаблонными ответами находится ниже блока добавления нового шаблона и содержит имеющиеся шаблонные ответы, а также кнопку “Изменить” для изменения шаблона.



## 6.1.8 Макет экрана сделки

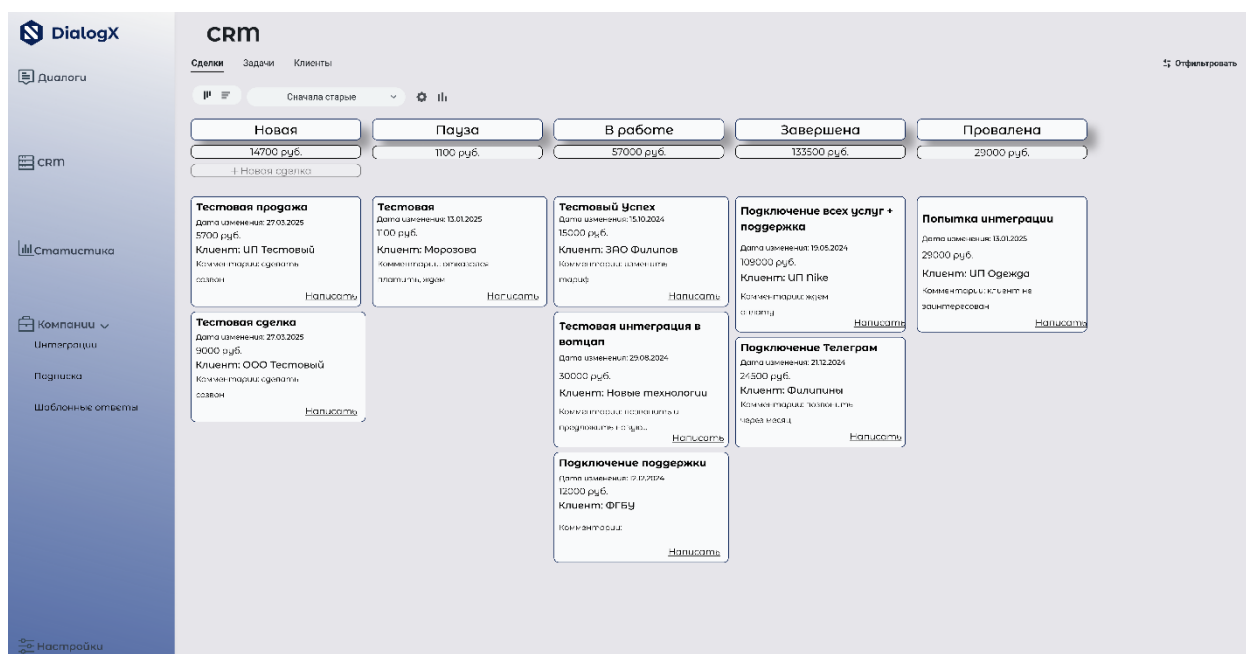


Рисунок 8 - Макет экрана сделки

На данном экране отображены новые, приостановленные, активные, завершенные и проваленные сделки, которые организованы в блоки с информацией о каждой сделке. В верхней части страницы есть сортировка, которая позволяет отсортировать сделки по дате (например, “Сначала старые”). Также присутствует кнопка “+Новая сделка”, по которой можно создать новую сделку. У каждой сделки есть название, дата создания, сумма сделки, информация о клиенте, а также можно оставить комментарий, который будет содержать дополнительную информацию о сделке.

## 6.1.9 Макет экрана статистики

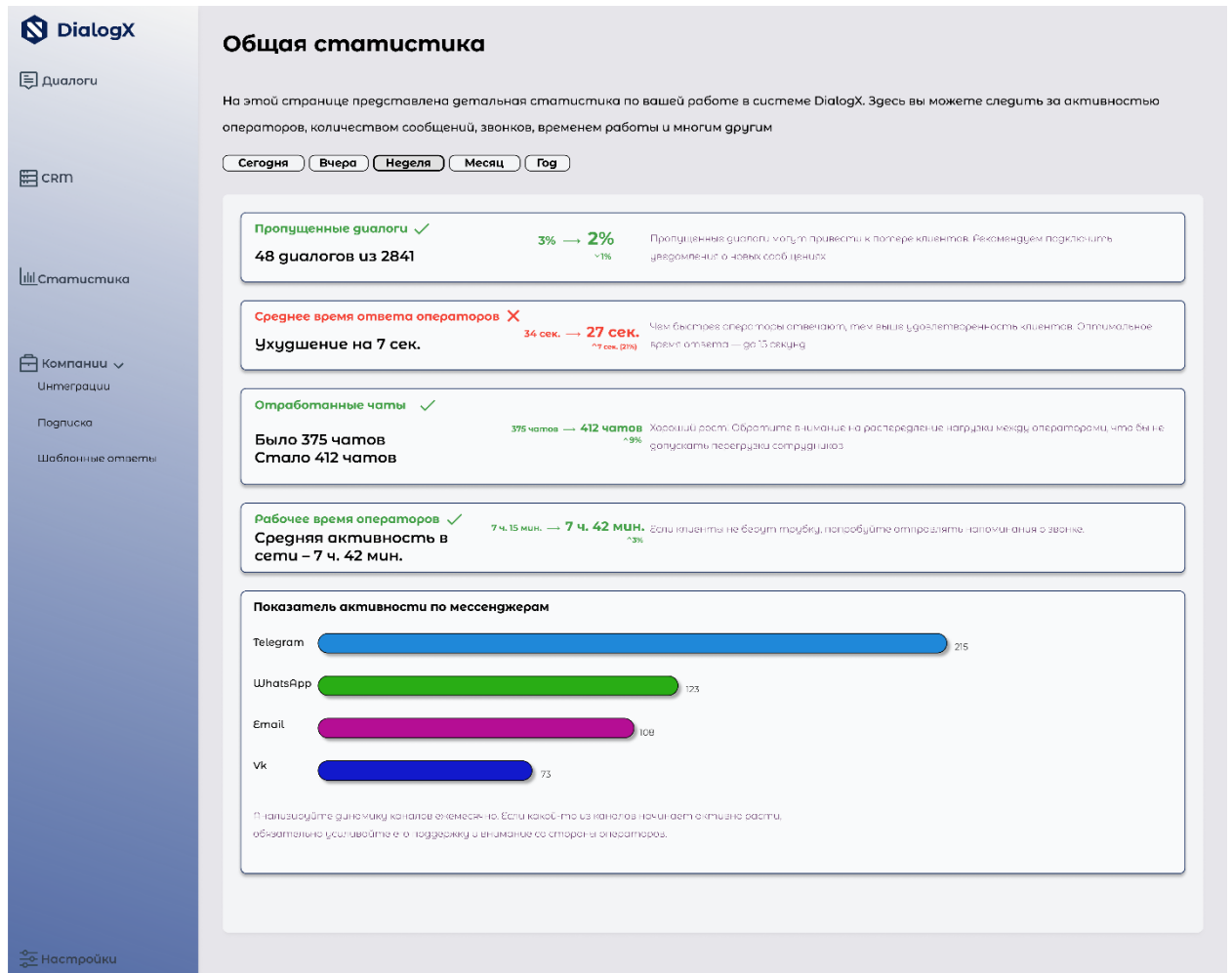


Рисунок 9 - Макет экрана статистики

На данном экране представлена статистика. Метрики, которые присутствуют на странице позволяют просматривать статистику по количеству пропущенных диалогов, мониторить среднее время ответа оператора, просматривать количество обработанных чатов, а также время работы оператора. В верхней части страницы присутствуют кнопки, которые позволяют выбрать период времени, за который будет отображена статистика (например, за неделю). В нижней части страницы находится диаграмма, которая отображает активность пользователей по конкретным мессенджерам.

## **7 Планы на дальнейшее развитие проекта**

После завершения основной разработки планируется разработать мобильное приложение, улучшить аналитику пользовательских запросов и внедрить систему предиктивных ответов.

## 8 Источники разработки

1. ГОСТ 34.602-89. Информационная технология. Комплекс стандартов на автоматизированные системы. Техническое задание на создание автоматизированной системы;
2. Pact (<https://www.pact.im/>);
3. Angry.Space (<https://angry.space/>).

## Приложение

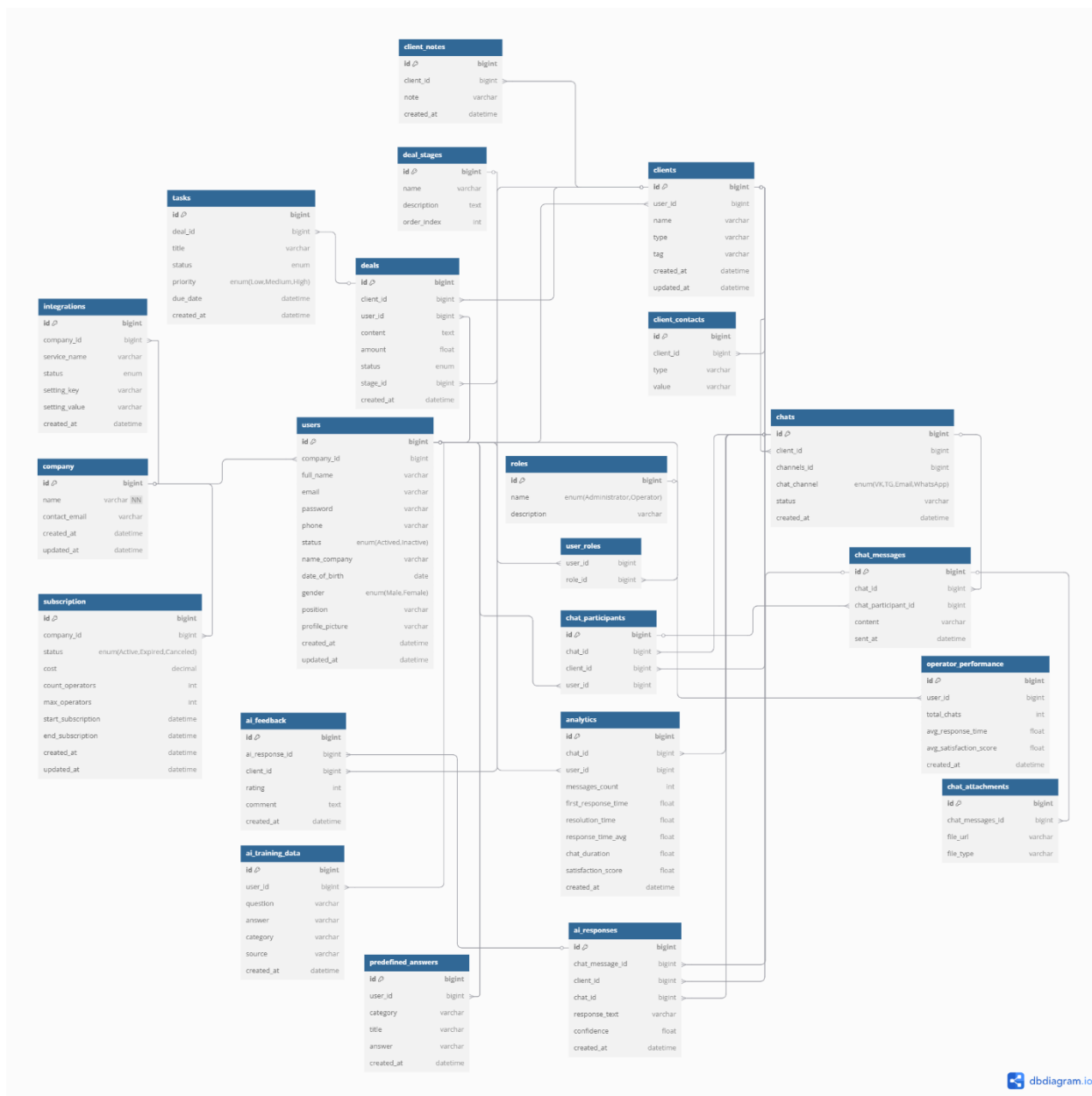


Рисунок 10 - ER-диаграмма

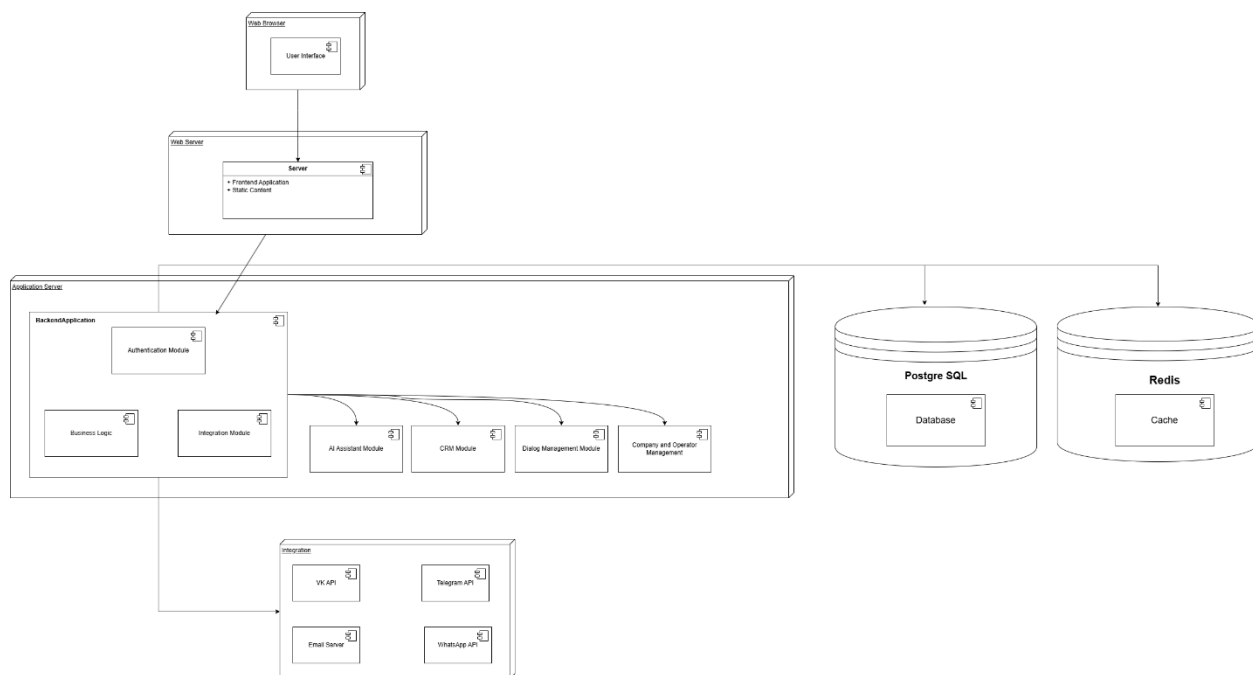


Рисунок 11 - Диаграмма развертывания

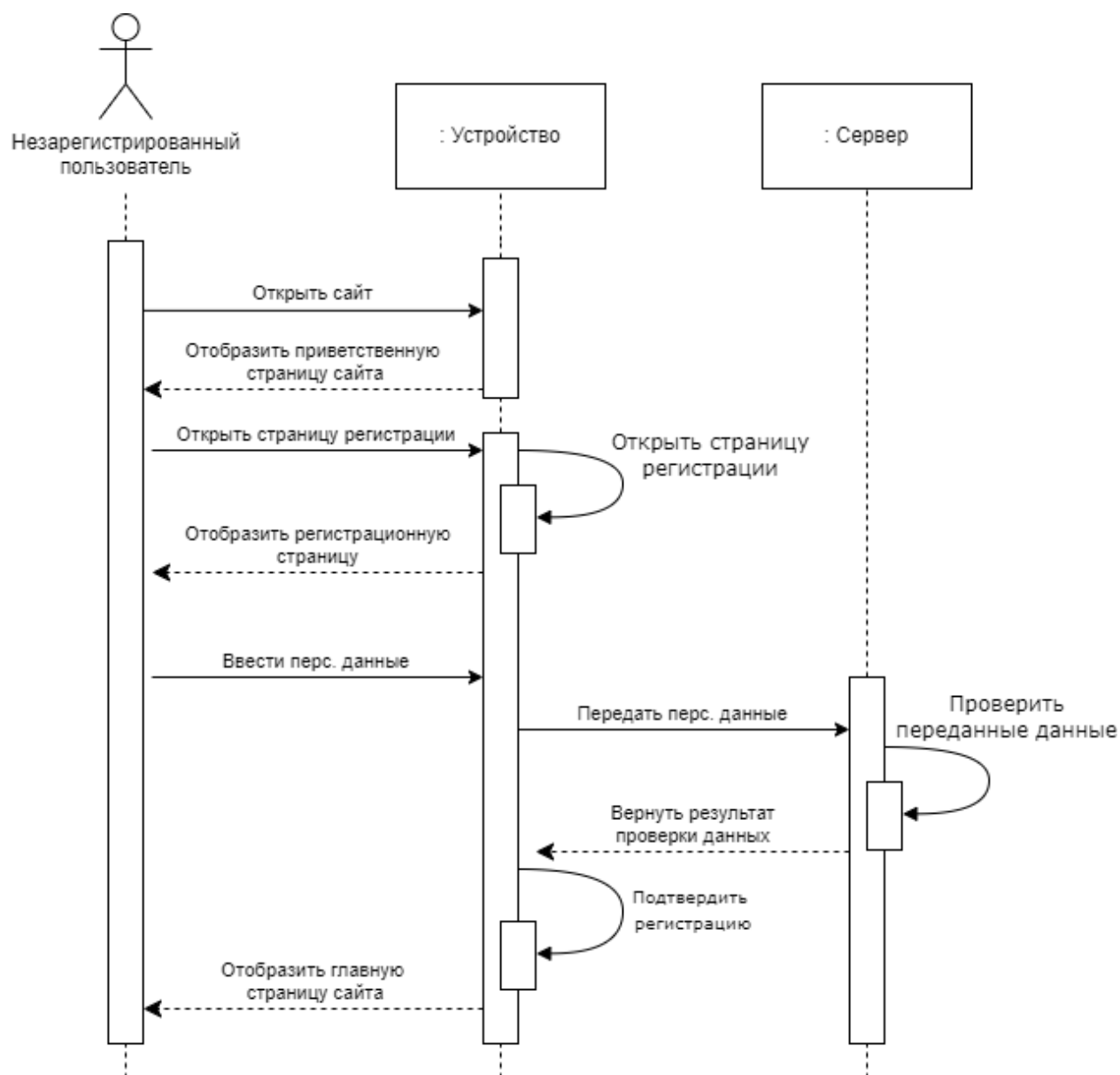


Рисунок 12 - Диаграмма последовательности (Незарегистрированный пользователь)

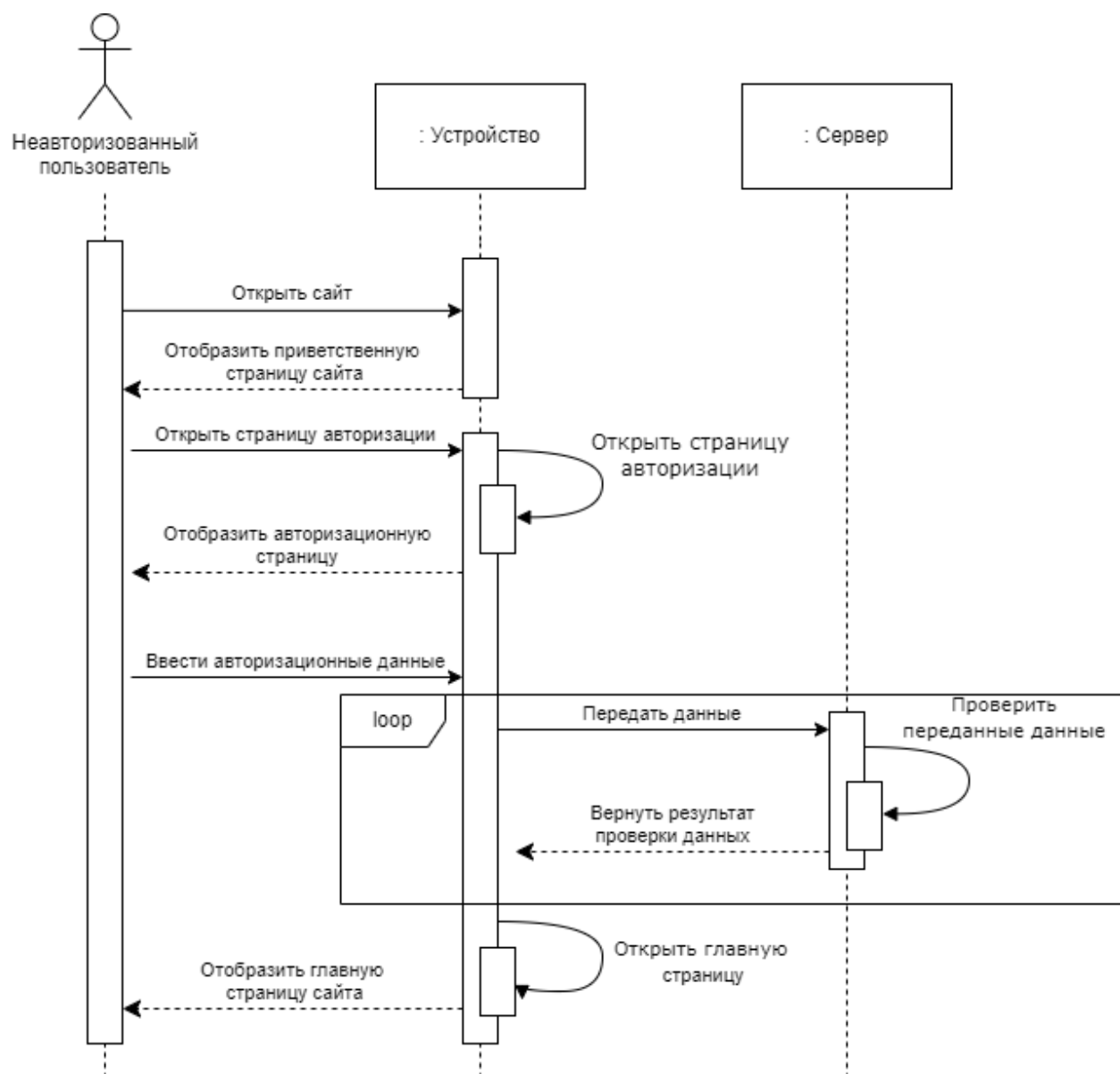


Рисунок 13 - Диаграмма последовательности (Неавторизованный пользователь)



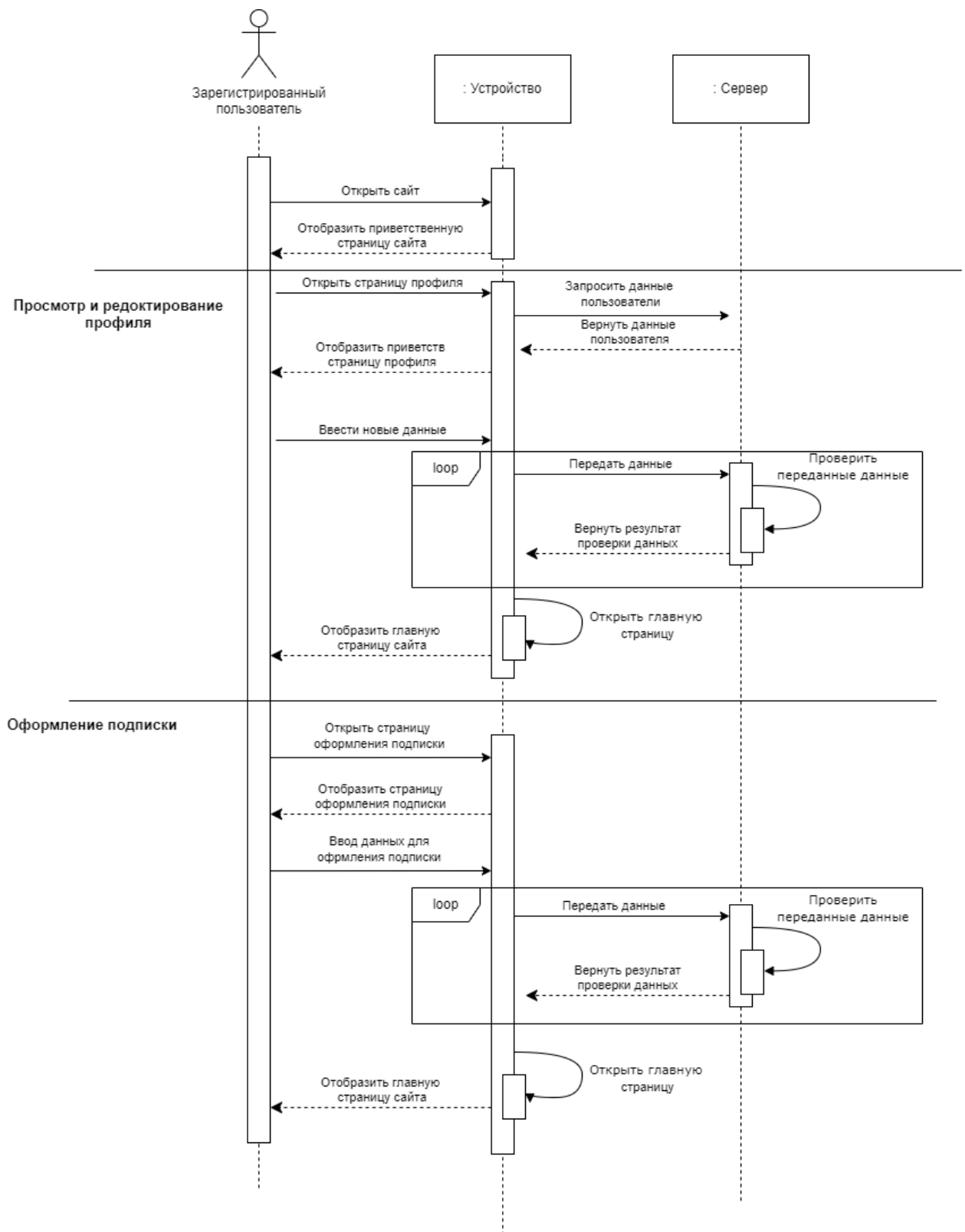


Рисунок 14 - Диаграмма последовательности (Авторизованный пользователь)

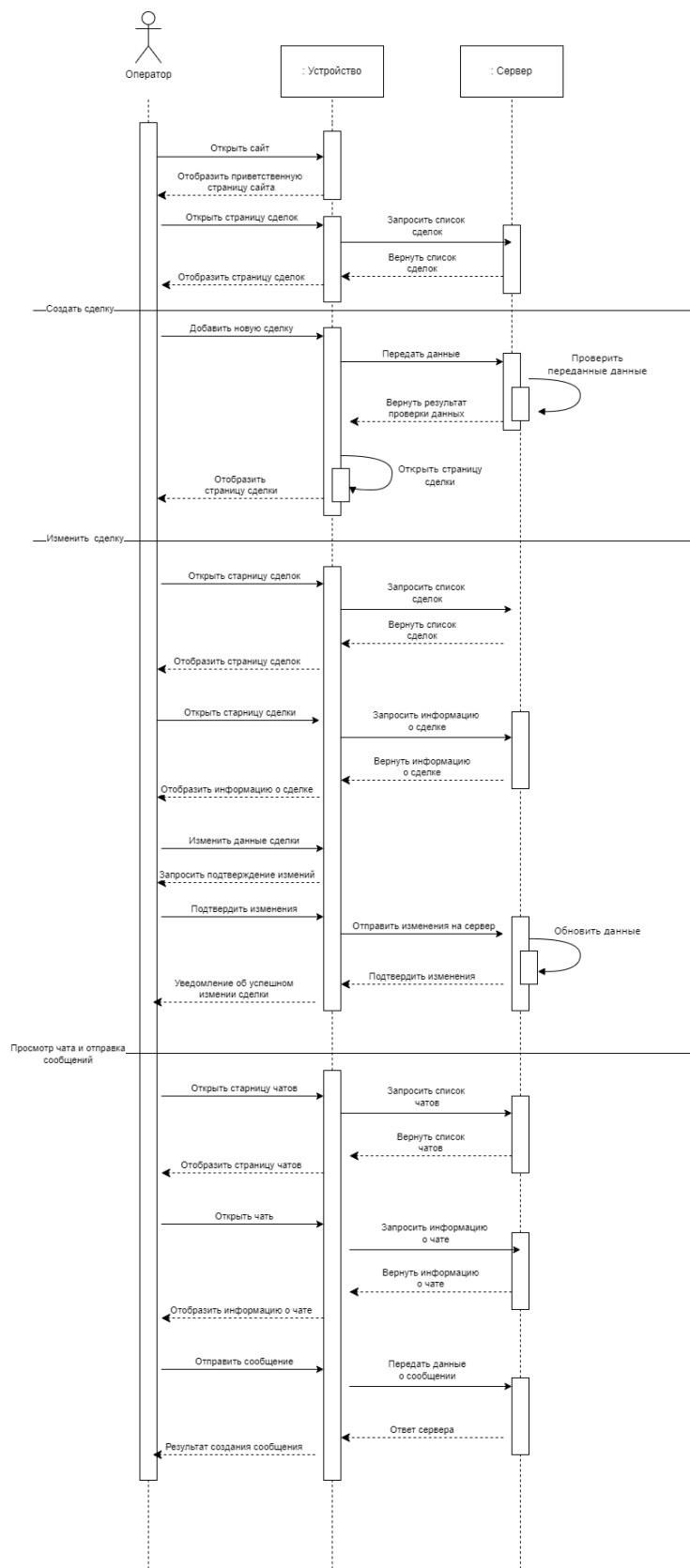


Рисунок 15 - Диаграмма последовательности (Оператор)

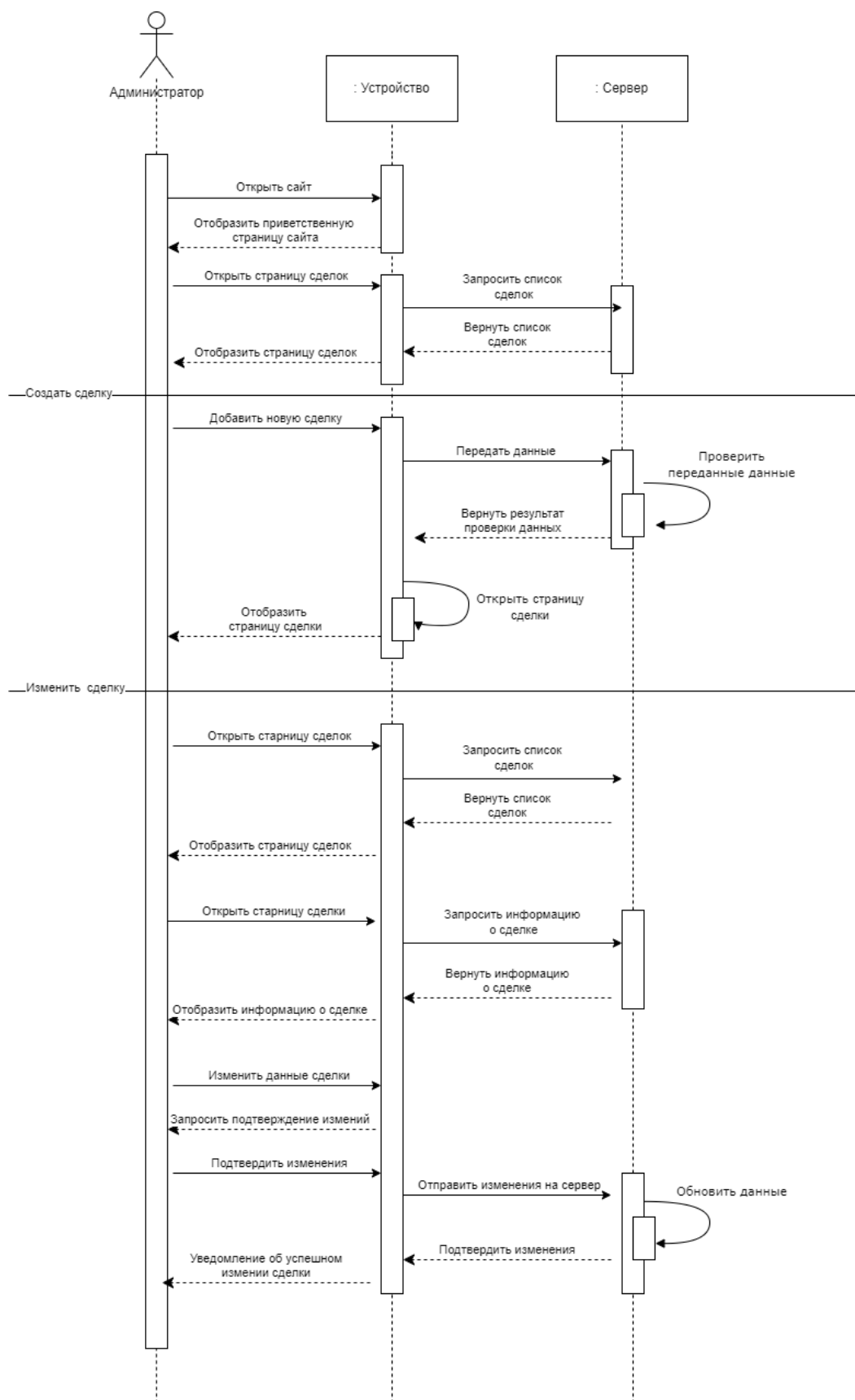


Рисунок 16 - Диаграмма последовательности (Администратор, сделка)

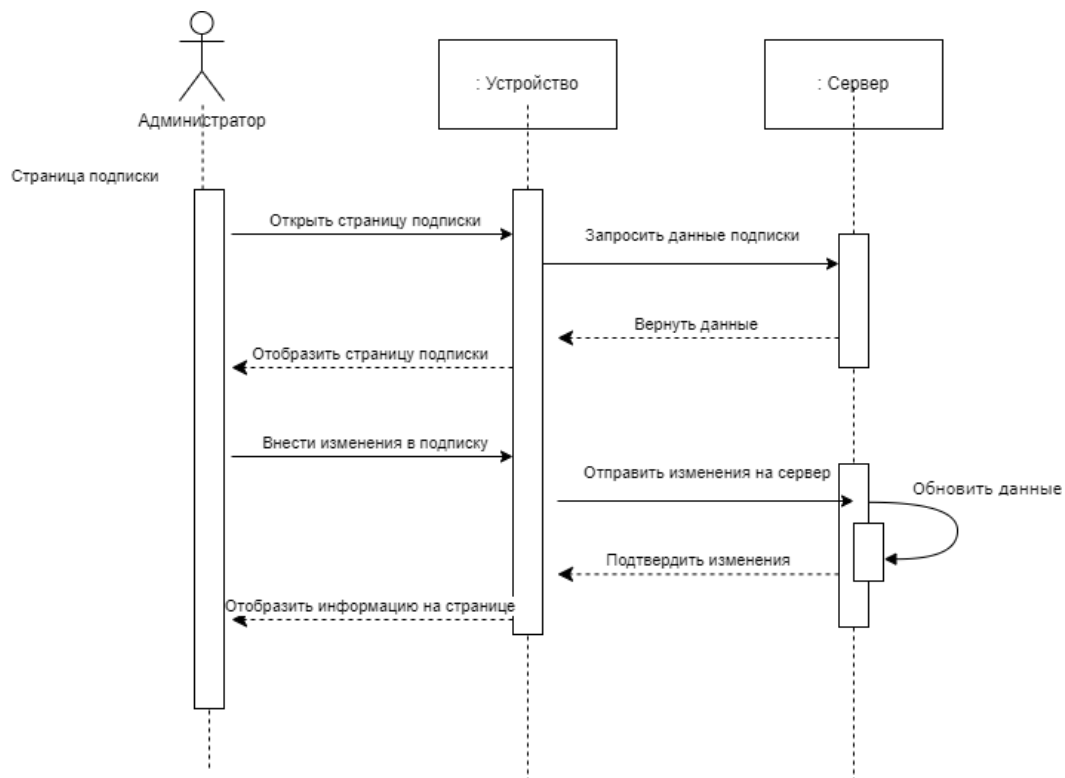


Рисунок 17 - Диаграмма последовательности (Администратор, подписка)

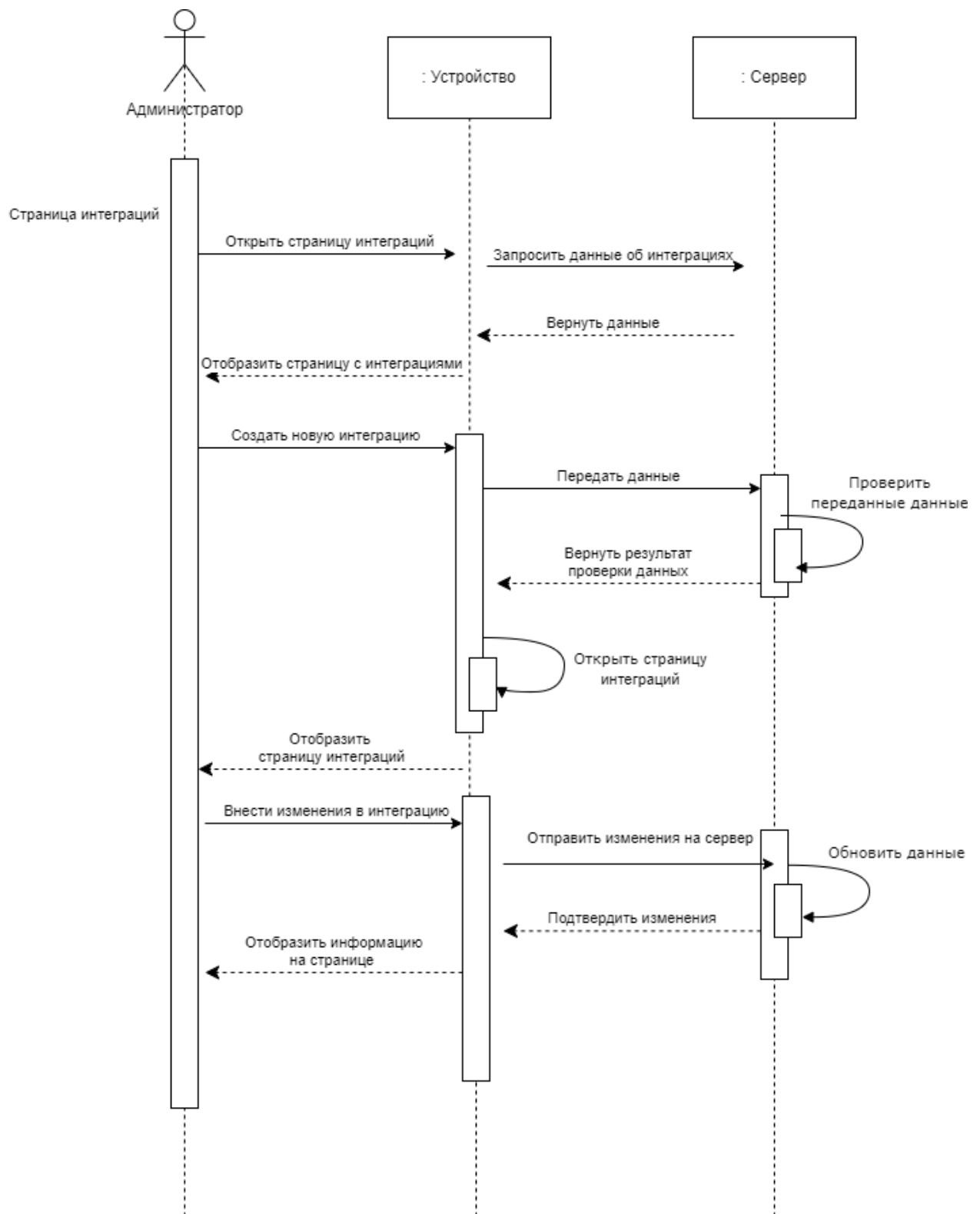


Рисунок 18 - Диаграмма последовательности (Администратор, интеграции)

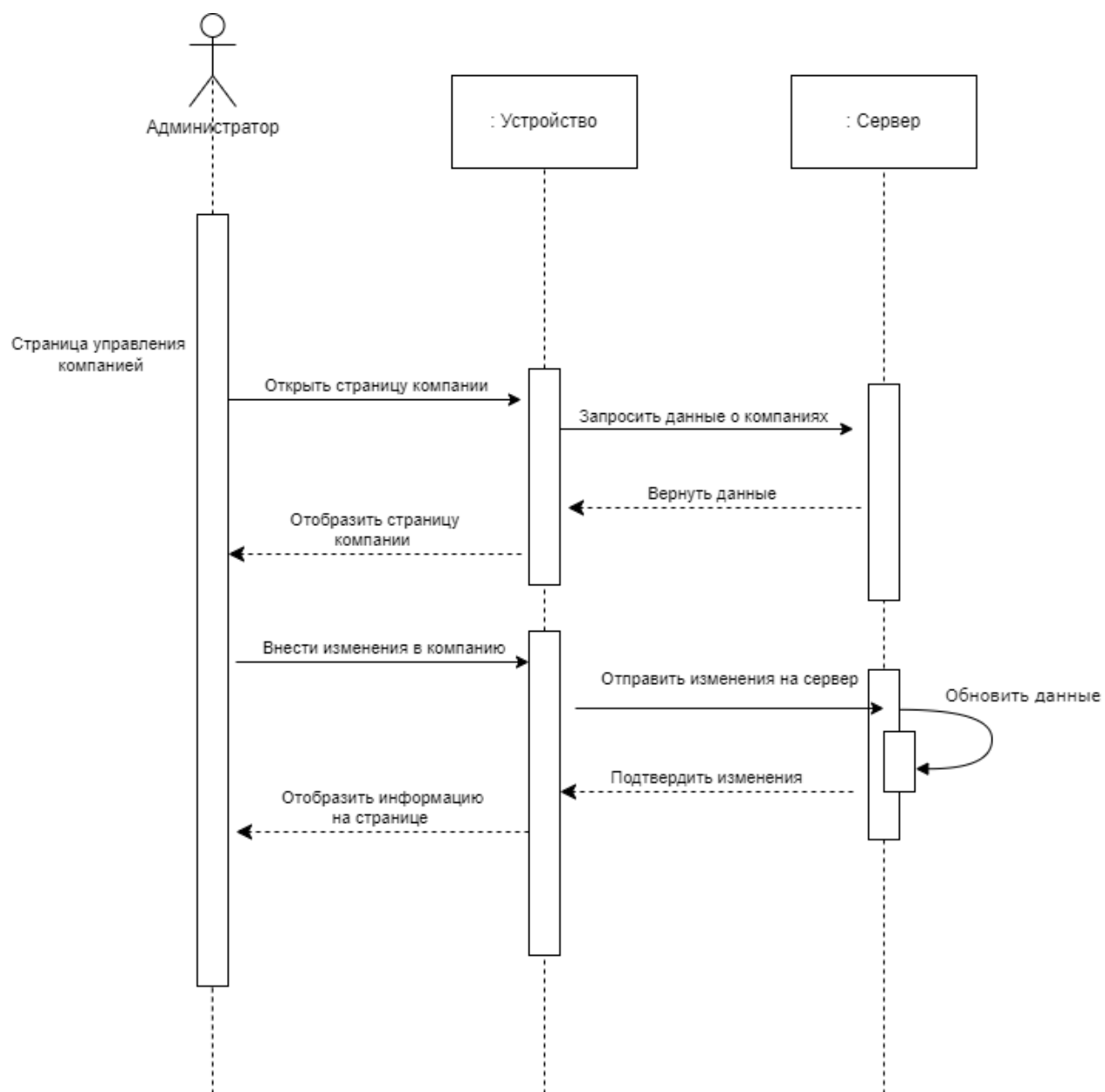


Рисунок 19 - Диаграмма последовательности (Администратор, компания)

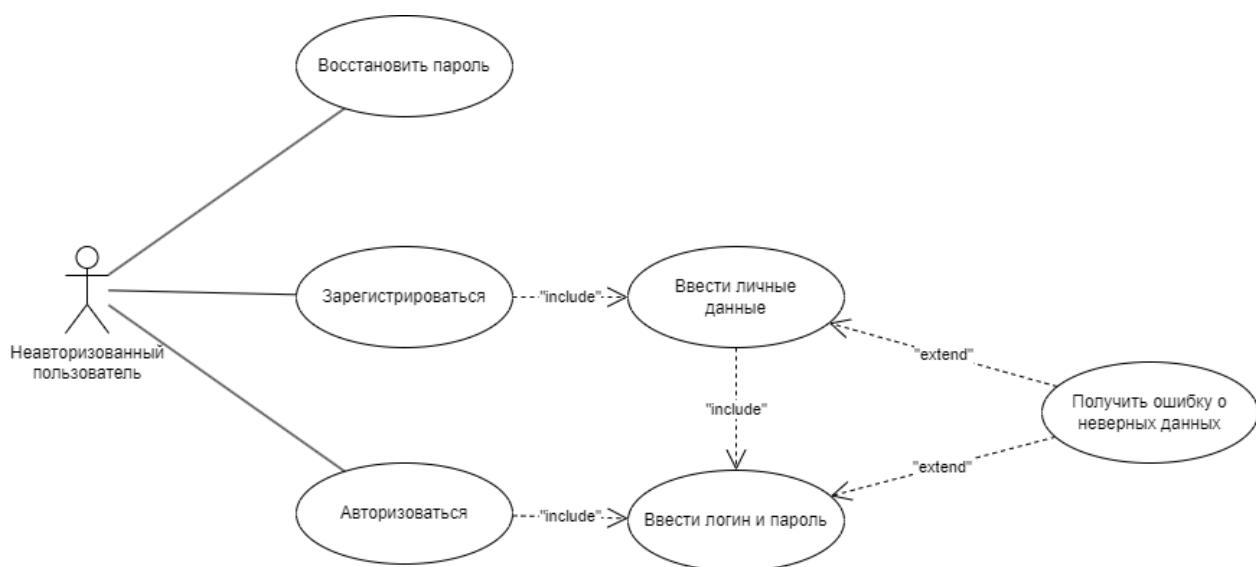


Рисунок 20 - Диаграмма прецедентов (Неавторизованный пользователь)

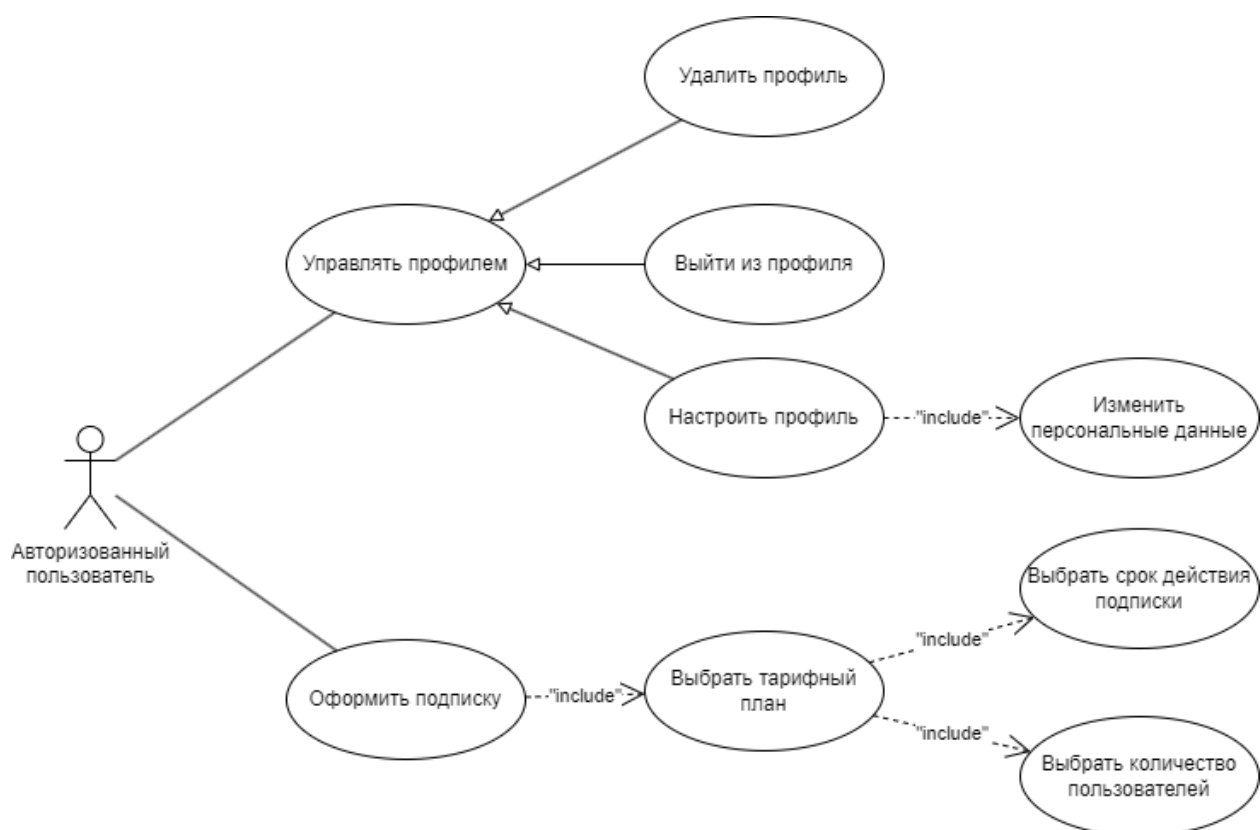


Рисунок 21 - Диаграмма прецедентов (Авторизованный пользователь)



Рисунок 22 - Диаграмма прецедентов (Оператор)

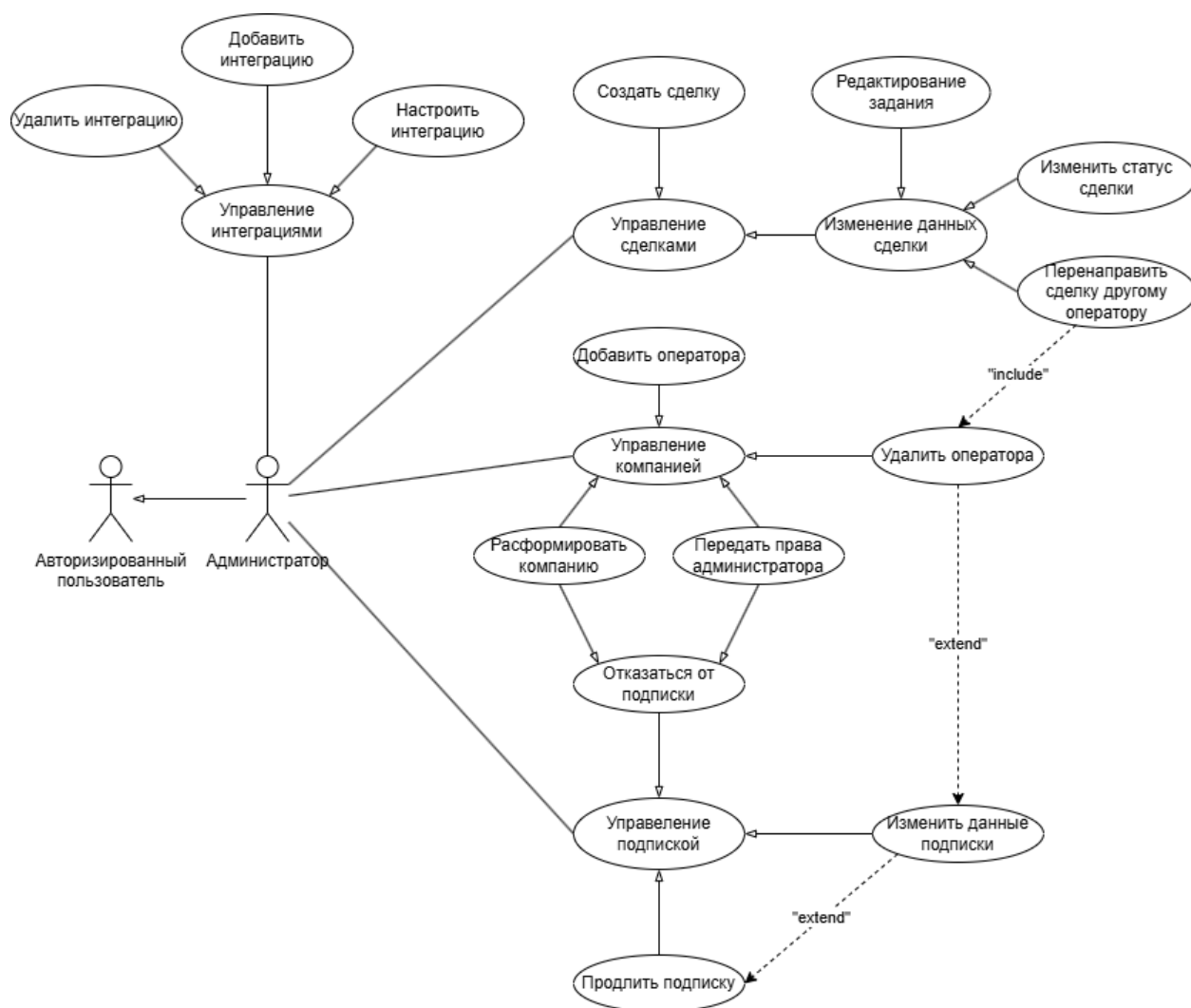


Рисунок 23 - Диаграмма прецедентов (Администратор)





Рисунок 24 - Диаграмма активности (Неавторизованный пользователь)

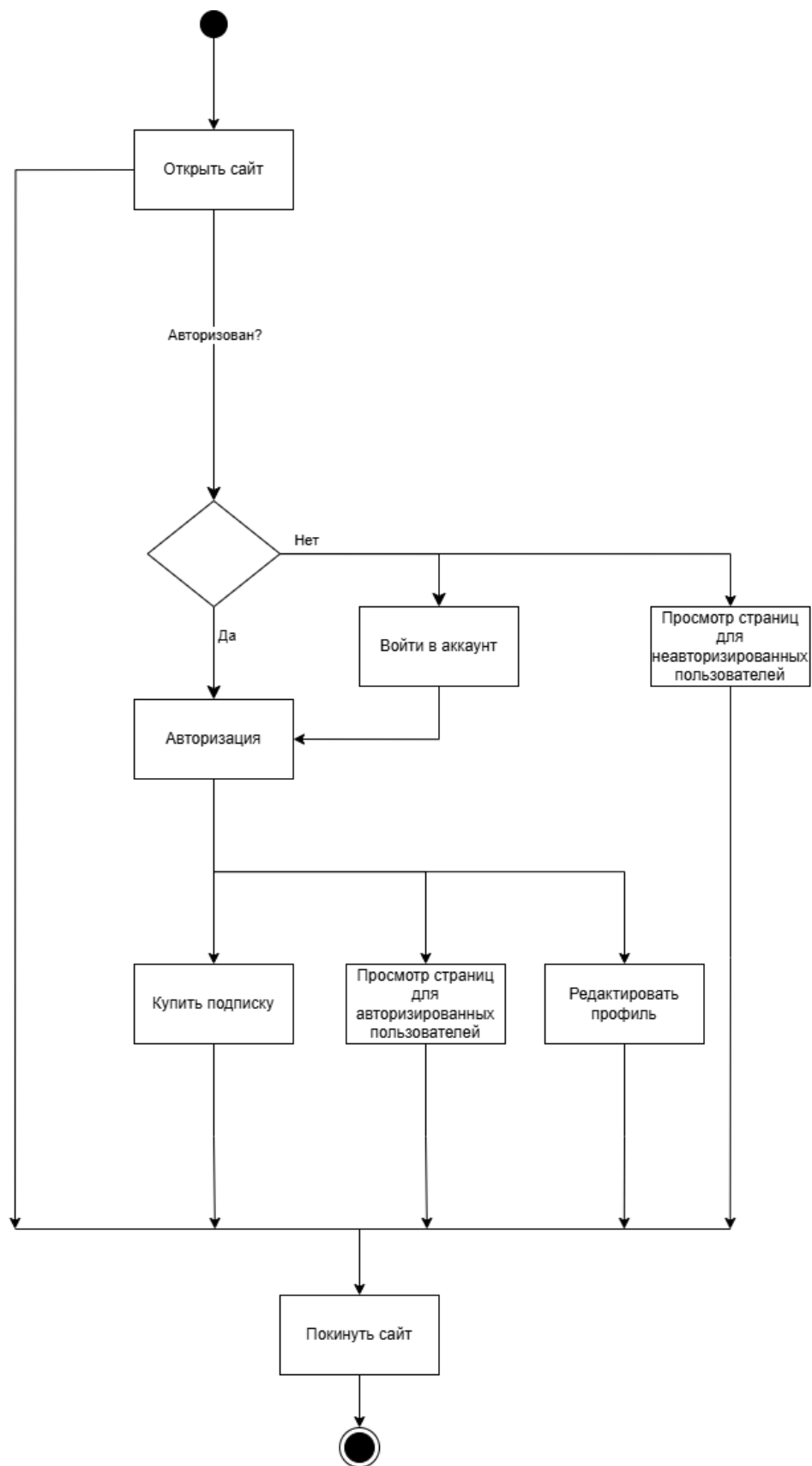


Рисунок 25 - Диаграмма активности (Авторизованный пользователь)

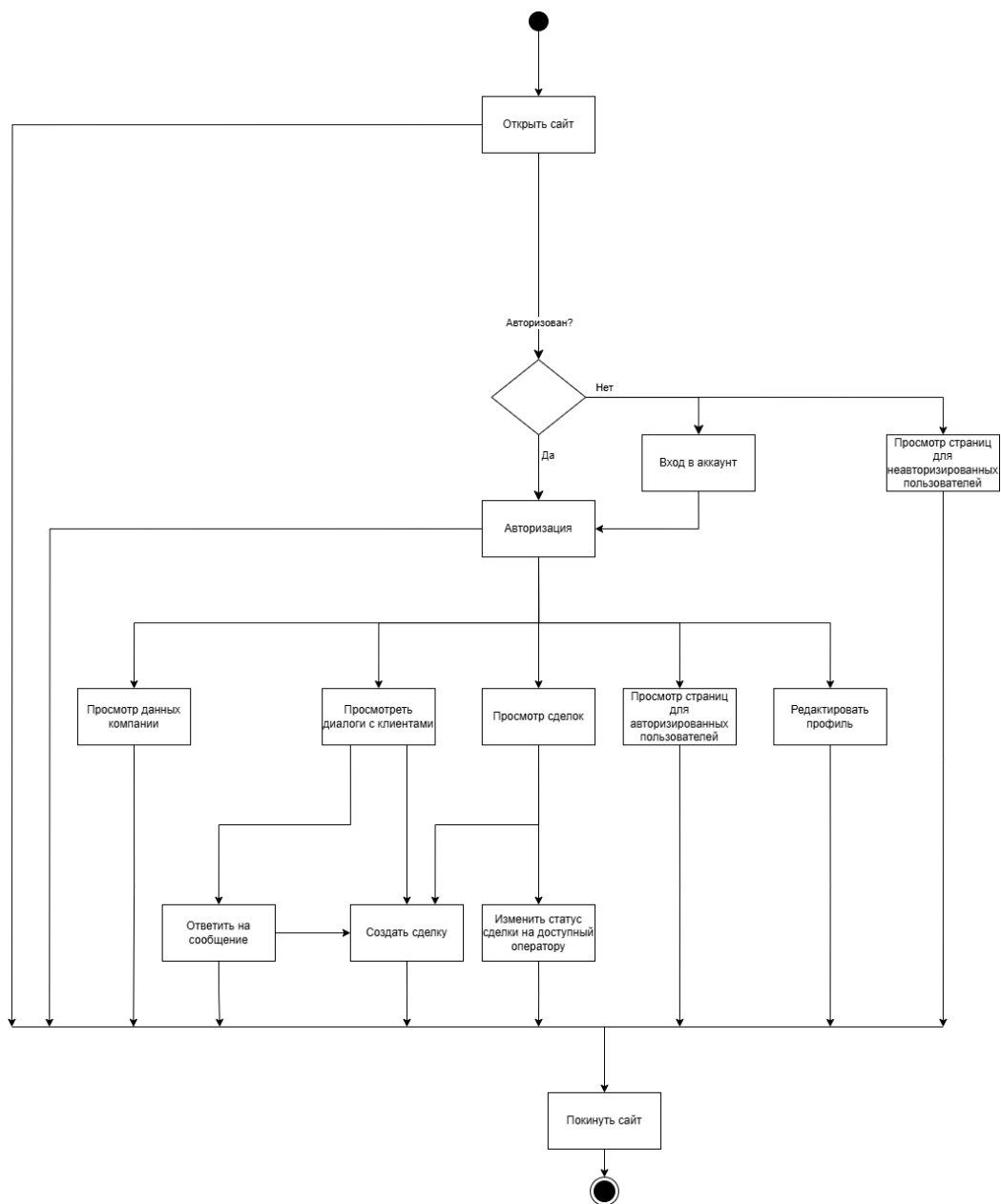


Рисунок 26 - Диаграмма активности (Оператор)

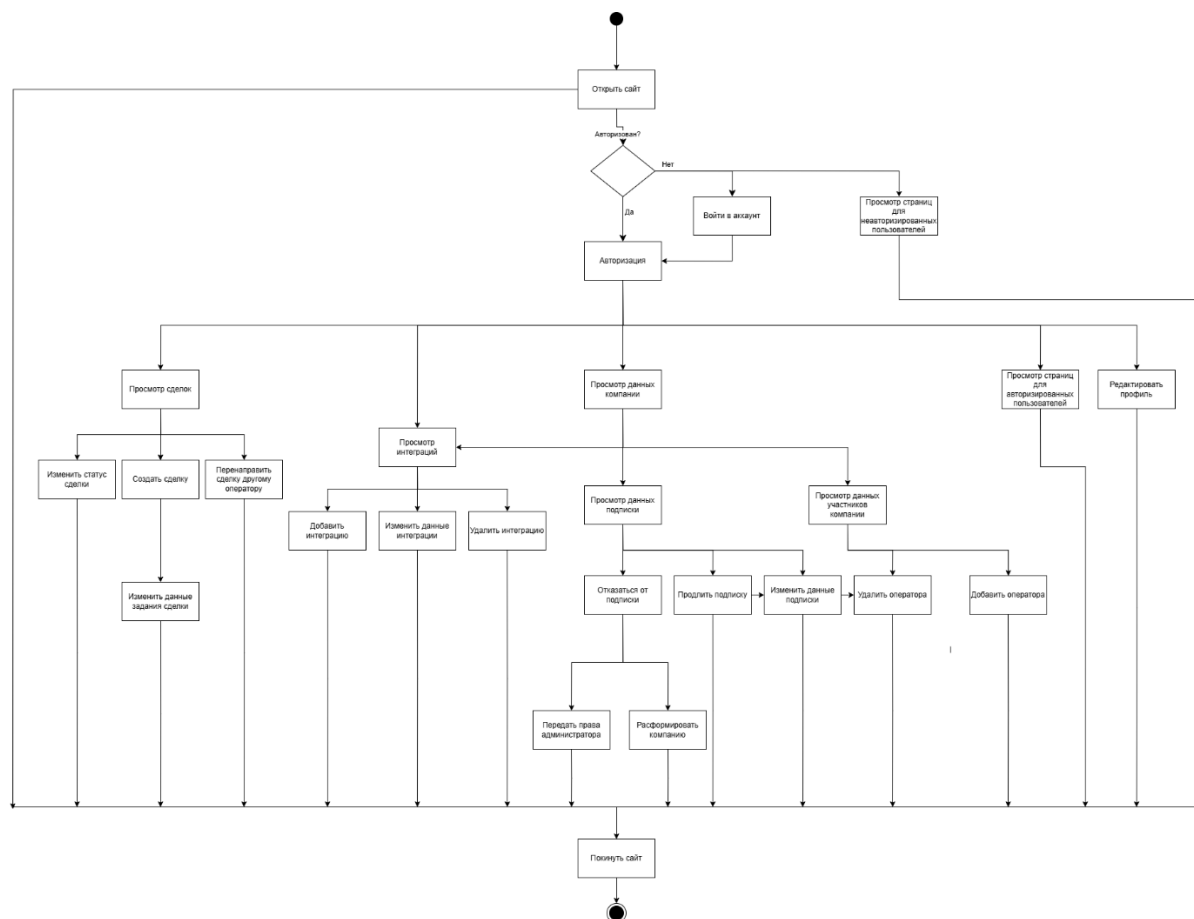


Рисунок 27 - Диаграмма активности (Администратор)

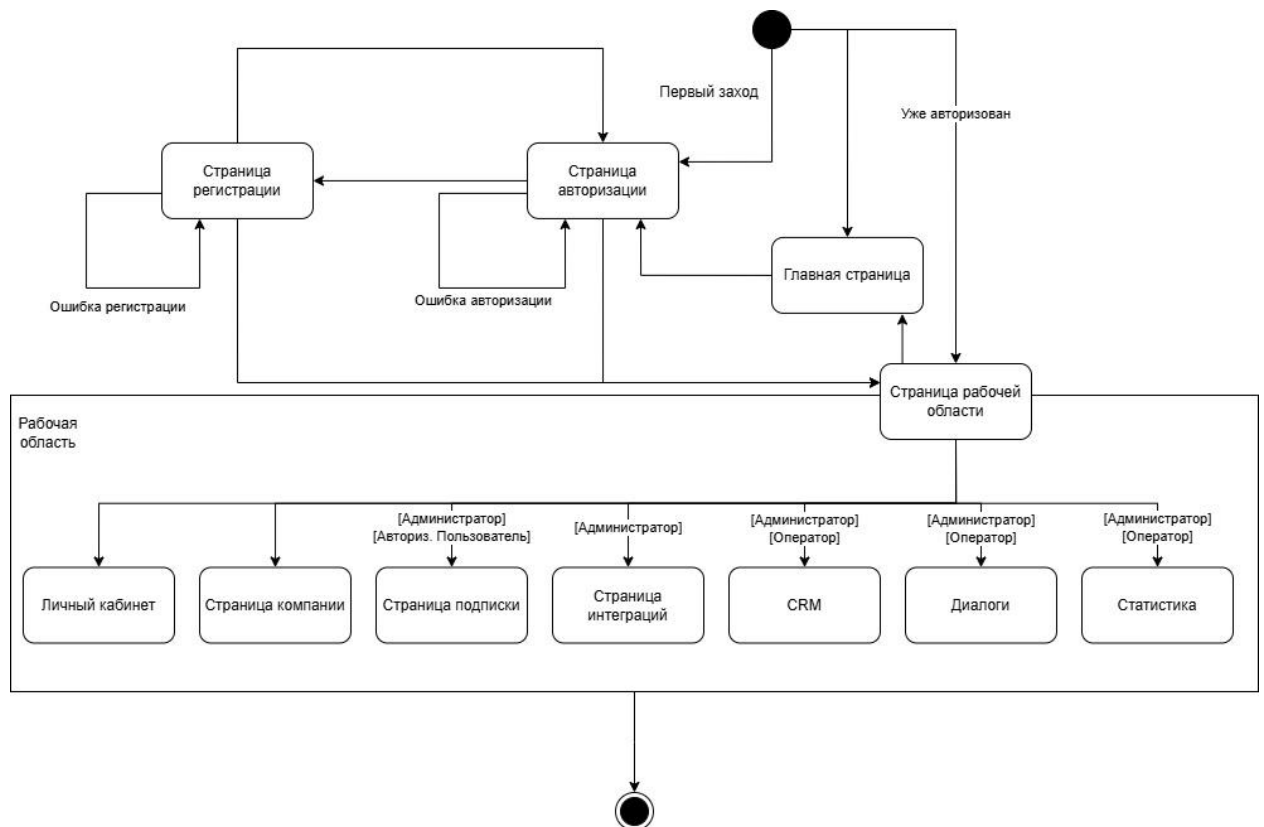


Рисунок 28 - Диаграмма состояний