

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

Факультет компьютерных наук
Кафедра технологий обработки и защиты информации

Курсовой проект
По дисциплине: Технологии программирования

Веб-приложение «Cowër»

09.03.04 Программная инженерия
Информационные системы и сетевые технологии

Зав. Кафедрой _____ д. ф.-м. н., профессор С. Д. Махортов
Обучающийся _____ П. В. Печенкин, 3 курс
Обучающийся _____ Т. А. Улезько, 3 курс
Обучающийся _____ Е. И. Сидорова, 3 курс
Руководитель _____ В.С. Тарасов, ст. преподаватель __.__.20__.

Воронеж 2024

СОДЕРЖАНИЕ

ПЕРЕЧЕНЬ СОКРАЩЕНИЙ И ОБОЗНАЧЕНИЙ	3
ВВЕДЕНИЕ	6
1 Анализ предметной области	7
1.1 WORKKODE.....	7
1.2 GrowUp.....	8
1.3 Ясная поляна.....	8
1.4 Итог анализа	9
2 Постановка задачи.....	10
2.1 Требования к функциональной части	10
2.2 Требования к программному обеспечению веб-сервиса	11
2.3 Общие требования к оформлению и верстке страниц	12
3 Графическое описание работы системы	13
3.1 Диаграммы прецедентов.....	13
3.2 Диаграмма развертывания.....	15
3.3 Диаграммы последовательности	16
4 Реализация.....	19
4.1 Реализация базы данных.....	19
4.2 Реализация серверной части приложения	20
4.3 Реализация клиентской части приложения	23
5 Тестирование	30
ЗАКЛЮЧЕНИЕ	31
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	32

ПЕРЕЧЕНЬ СОКРАЩЕНИЙ И ОБОЗНАЧЕНИЙ

Перечень терминов, используемых в курсовом проекте, представлен в таблице 1.

Таблица 1 - Термины и сокращения

Термин	Определение термина
Авторизация	Предоставление определённому лицу прав на выполнение определённых действий, а также процесс проверки (подтверждения) данных прав при попытке выполнения этих действий
Авторизованный пользователь	Пользователь, прошедший процесс авторизации
Администратор	Человек, имеющий доступ к расширенному функционалу веб-сервиса
Веб-сайт	Набор связанных по смыслу страниц, которые находятся в сети Интернет. На них размещается информация, текстовая или графическая
Веб-сервис	Идентифицируемая уникальным веб-адресом (URL-адресом) программная система со стандартизированными интерфейсами, а также HTML-документ сайта, отображаемый браузером пользователя
Коворкинг	Офисное пространство, в котором человек может арендовать рабочее место
Модальное окно	Такое окно, которое появляется поверх родительского окна
Неавторизованный пользователь	Пользователь, не прошедший процесс авторизации

Термин	Определение термина
Пользователь	Лицо, которое использует систему для решения стоящих перед ним задач
Профиль	Учетная запись пользователя в веб-приложении, вход в которую осуществляется с помощью логина, номера телефона или электронной почты и пароля. В учетной записи содержится информация о пользователе
Регистрация	Раздел веб-приложения, предназначенный для создания профиля пользователя, в котором человек указывает свои данные и придумывает пароль
СУБД	Система управления базами данных. Комплекс программ, позволяющих создать базу данных (БД) и манипулировать данными
Фреймворк	Заготовка, готовая модель в программировании для быстрой разработки, на основе которой можно дописать собственный код. Он задает структуру, определяет правила и предоставляет необходимый набор инструментов для создания проекта
Фрилансер	Человек, занимающийся удалённым способом заработка без трудоустройства в штат
Фронтенд	Пользовательский интерфейс компьютера или любого устройства
Чекаут	Страница или набор страниц, которые покупатели видят, когда переходят к оплате покупки

Термин	Определение термина
C#	Высокоуровневый язык программирования общего назначения
CSS	Формальный язык, служащий для описания оформления внешнего вида документа, созданного с использованием языка разметки (HTML, XHTML, XML)
HTML	Стандартизированный язык гипертекстовой разметки документов для просмотра веб-страниц в браузере
JavaScript	Язык программирования, который используют разработчики для создания интерактивных веб-страниц
PostgreSQL	Свободная объектно-реляционная система управления базами данных
React	JavaScript-библиотека для создания пользовательских интерфейсов
REST	Архитектурный стиль взаимодействия компонентов распределенного приложения в сети
TypeScript	Язык программирования, позиционируемый как средство разработки веб-приложений, расширяющее возможности JavaScript

ВВЕДЕНИЕ

С ростом популярности удаленной работы в современном мире увеличивается популярность коворкингов [1]. В связи с этим возникает потребность в решениях для бронирования рабочих мест в коворкингах онлайн.

В процессе разработки за основу были взяты уже имеющиеся решения, одной из задач стало сохранение преимуществ этих решений и исправление недостатков.

Целью данного курсового проекта является разработка веб-приложения «Cowër», предоставляющего пользователям возможность выбирать рабочие места на карте коворкинга и бронировать их, оплатить бронь онлайн, студентам и волонтерам получать скидки на бронирование.

1 Анализ предметной области

Анализ существующих решений будет проводиться на основе сервисов, перечисленных в таблице 2.

Таблица 2 - Примеры существующих решений

WORKKODE	https://wkode.co/
GrowUp	https://growup-coworking.ru/
Ясная поляна	https://coworkingspb.ru/coworking#openpaces

1.1 WORKKODE

Веб сервис для аренды рабочих мест и деловых пространств на день, месяц и год.

Преимущества:

- перечисление характеристик рабочих мест;
- наличие информации о коворкинге на главной странице.

Недостатки:

- отсутствие карты коворкинга с выбором рабочего места на ней;
- отсутствие скидок для студентов;
- отсутствие информации о возможном времени бронирования на сайте;
- отсутствие возможности забронировать место меньше, чем на день;
- отсутствие возможности осуществить и оплатить бронь на сайте.

1.2 GrowUp

Веб сервис для аренды рабочих мест в коворкинге на час, день, 10 дней и месяц.

Преимущества:

- возможность забронировать рабочее место на час;
- наличие возможности заказать организацию мероприятия.

Недостатки:

- отсутствие карты коворкинга с выбором рабочего места на ней;
- отсутствие скидок для студентов;
- отсутствие информации о возможном времени бронирования на сайте;
- отсутствие подробной информации о рабочих местах;
- отсутствие возможности осуществить и оплатить бронь на сайте.

1.3 Ясная поляна

Веб сервис для аренды рабочих мест в коворкинге на час, день и месяц.

Преимущества:

- наличие подробной информации о каждой зоне коворкинга;
- возможность забронировать рабочее место на час;
- наличие скидок для студентов.

Недостатки:

- отсутствие карты коворкинга с выбором рабочего места на ней;
- отсутствие информации о возможном времени бронирования на сайте;
- отсутствие возможности осуществить и оплатить бронь на сайте.

1.4 Итог анализа

В процессе анализа предметной области было установлено, что при разработке приложения следует придерживаться следующих аспектов:

- наличие карты коворкинга с выбором рабочего места на ней;
- наличие скидок для студентов и волонтеров;
- возможность бронирования рабочего места на сайте;
- возможность оплаты брони на сайте;
- наличие информации о рабочих местах;
- наличие информации о возможном времени бронирования на сайте.

2 Постановка задачи

Целью данного курсового проекта является разработка веб-сервиса «Cowër», предназначенного для бронирования рабочих мест в коворкинге.

2.1 Требования к функциональной части

Приложение должно содержать следующие бизнес-роли пользователей:

- неавторизованный пользователь;
- авторизованный пользователь;
- администратор.

Для неавторизованного пользователя должны предоставляться следующие возможности:

- регистрация и авторизация на сервисе;
- просмотр карты коворкинга по этажам;
- выбор даты бронирования;
- выбор времени начала и окончания бронирования;
- выбор рабочего места на карте и просмотр описания этого места;
- восстановление пароля.

Для авторизованного пользователя должны предоставляться следующие возможности:

- просмотр карты коворкинга по этажам;
- выбор рабочего места на карте и просмотра описания этого места;
- выбор даты бронирования;
- выбор времени начала и окончания бронирования;
- бронирование выбранного места;
- оплата заказа;
- просмотр заказов, созданных пользователем;

- отмена заказа, при условии, что время, на которое совершена бронь еще не наступило;

- просмотр и редактирование имени, фамилии, номера телефона в личном кабинете.

Для администратора должны предоставляться следующие возможности:

- просмотр карты коворкинга;

- добавление, перемещение, удаление рабочих мест на карте коворкинга по этажам;

- изменение описания рабочего места;

- изменение цены на рабочее место за час;

- просмотр, добавление и отмена заказов пользователей.

2.2 Требования к программному обеспечению веб-сервиса

Для реализации серверной части приложения будут использоваться следующие средства:

- язык программирования C# версии 12;

- фреймворк ASP.NET Core версии 8.0;

- СУБД PostgreSQL версии 16.2.

Для реализации клиентской части приложения будут использоваться следующие средства:

- HTML версии 5;

- CSS версии 3;

- язык программирования TypeScript версии 5.2.2;

- библиотека React 18.2.0.

2.3 Общие требования к оформлению и верстке страниц

Сайт должен быть оформлен в одной цветовой палитре с использованием ограниченного набора шрифтов. У страниц сайта должен быть единый стиль.

Основные цвета веб-сайта:

- бежевый используется как цвет фона веб-сайта;
- розовый используется как цвет верхней панели и рабочих мест на карте;
- бордовый используется как цвет основного текста и иконок.

Основные шрифты веб-сайта:

- Saira;
- Montserrat;

В оформлении приложения должны присутствовать разработанные название и логотип.

Необходимо корректное и одинаковое отображение страниц сайта в следующих браузерах:

- Google Chrome версии 122.0.6261 и новее;
- Mozilla Firefox версии 123.0 и новее;
- Opera версии 108.0.5067 и новее;
- Яндекс Браузер версии 23.9.0 и новее.

Необходимо корректное отображение интерфейса на персональных компьютерах с операционными системами Windows версии 7 и выше, с разрешением окна браузера от 1280 x 720 пкс.

3 Графическое описание работы системы

В данном разделе содержится графическое описание работы системы.

3.1 Диаграммы прецедентов

Далее будут приведены диаграммы Use-Case (диаграммы прецедентов) для разных ролей пользователей. Эти диаграммы демонстрируют различные сценарии, возникающие при использовании сервиса.

На рисунке 1 представлена Use-case диаграмма для неавторизованного пользователя.

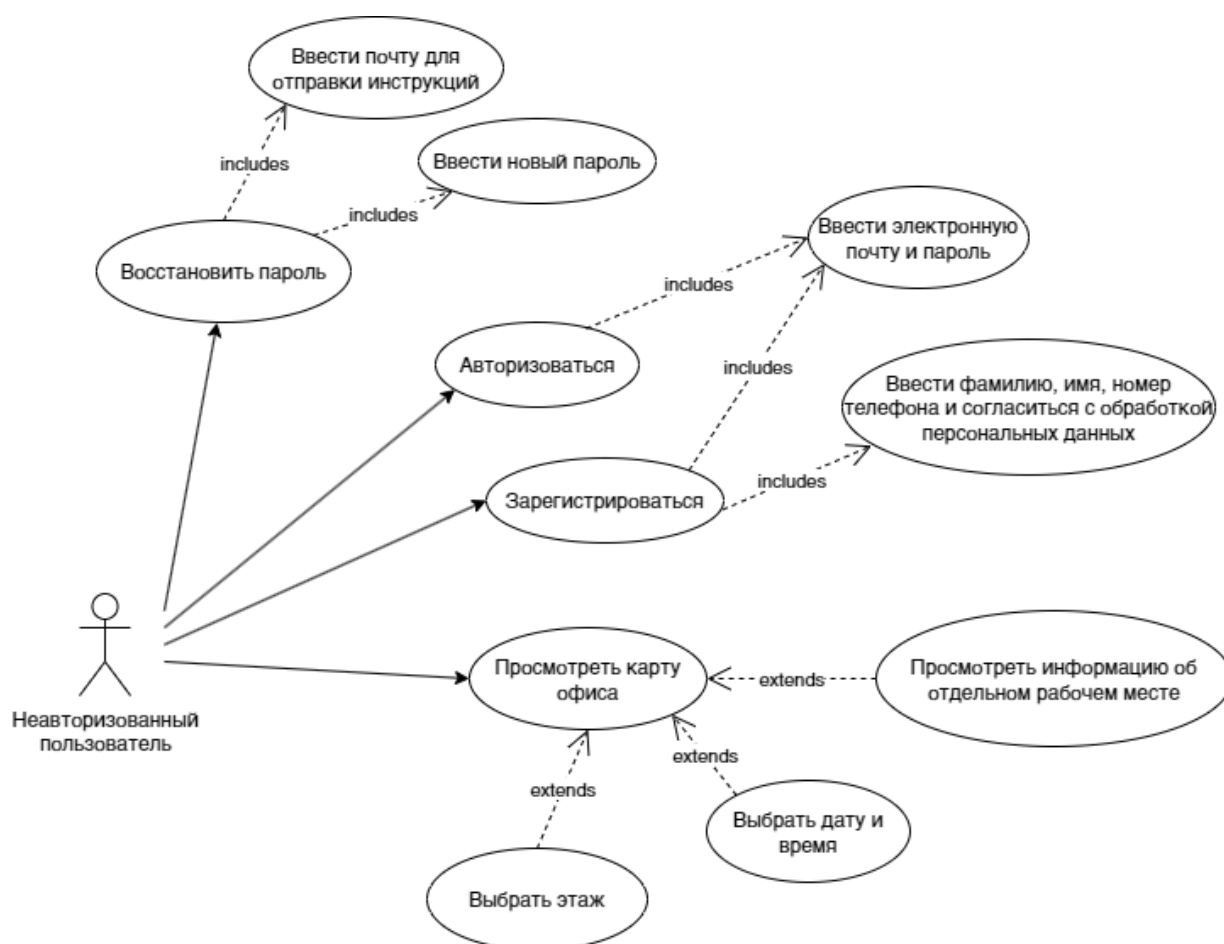


Рисунок 1 - Use-case диаграмма для неавторизованного пользователя

На рисунке 2 представлена Use-case диаграмма для авторизованного пользователя.

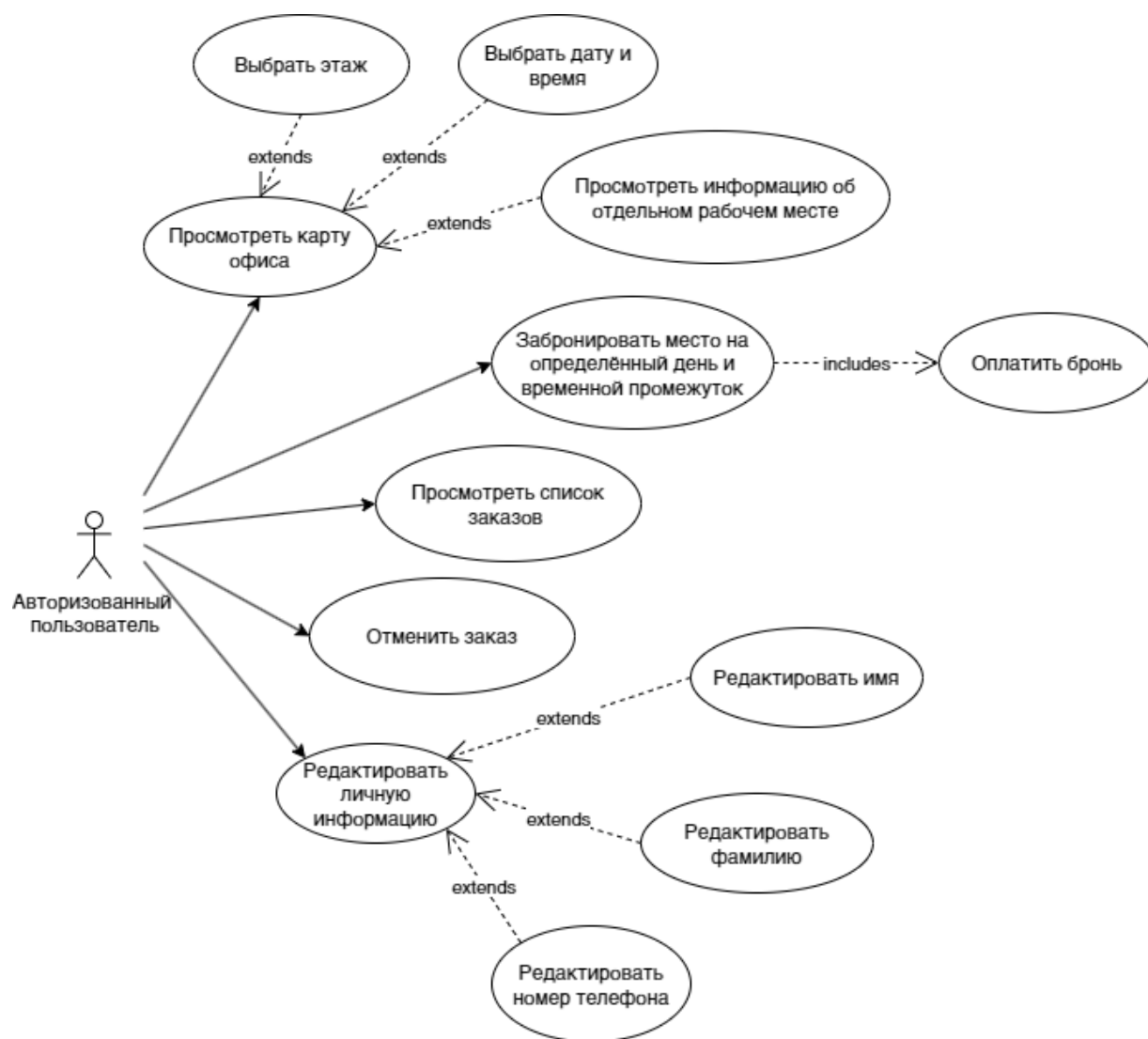


Рисунок 2 - Use-case диаграмма для авторизованного пользователя
На рисунке 3 представлена Use-case диаграмма для администратора.

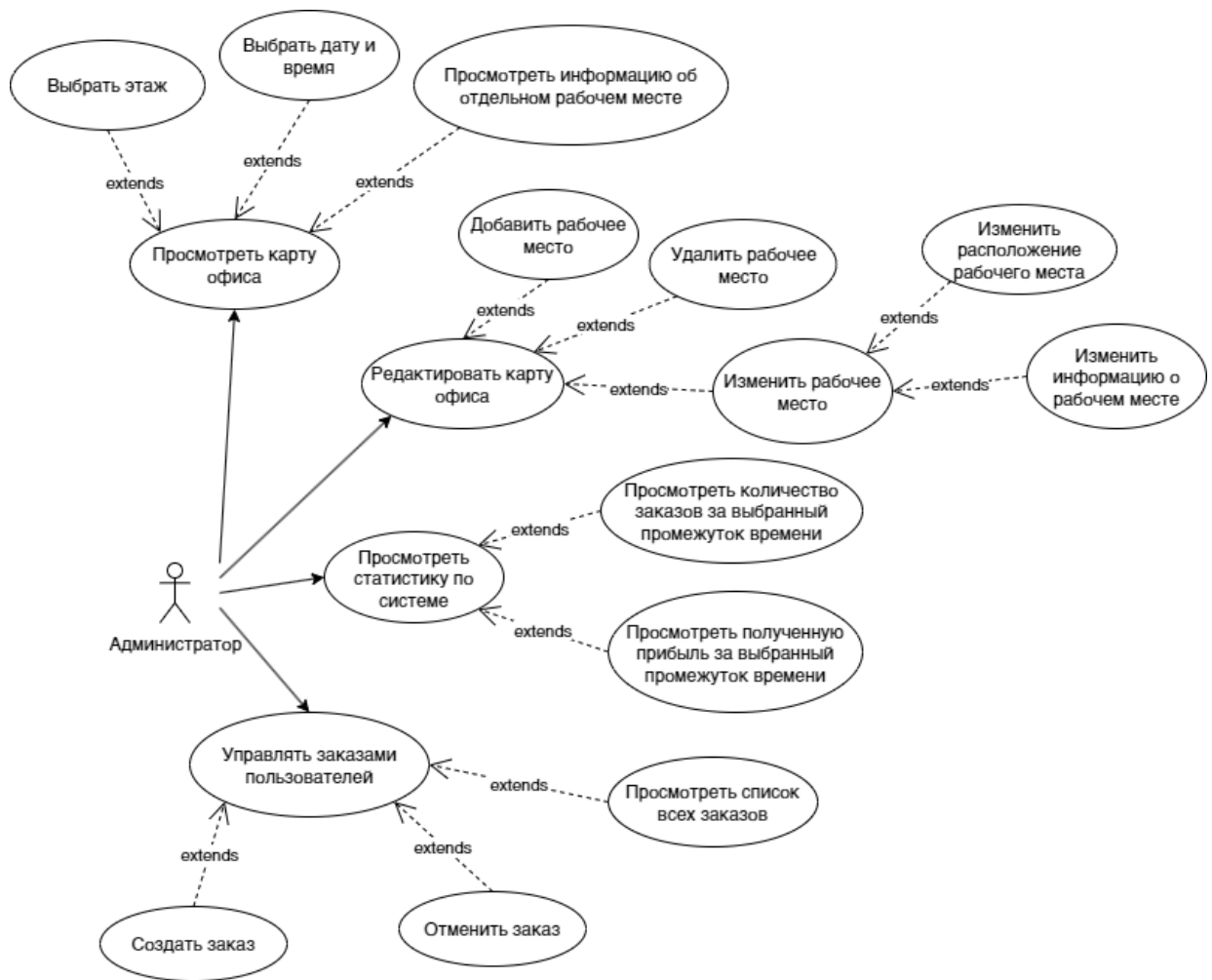


Рисунок 3 - Use-case диаграмма для администратора

3.2 Диаграмма развертывания

На рисунке 4 представлена диаграмма развёртывания. Она используется для моделирования архитектуры системы, ее компонентов. Она показывает структуру системы, то есть как ее компоненты взаимодействуют между собой.

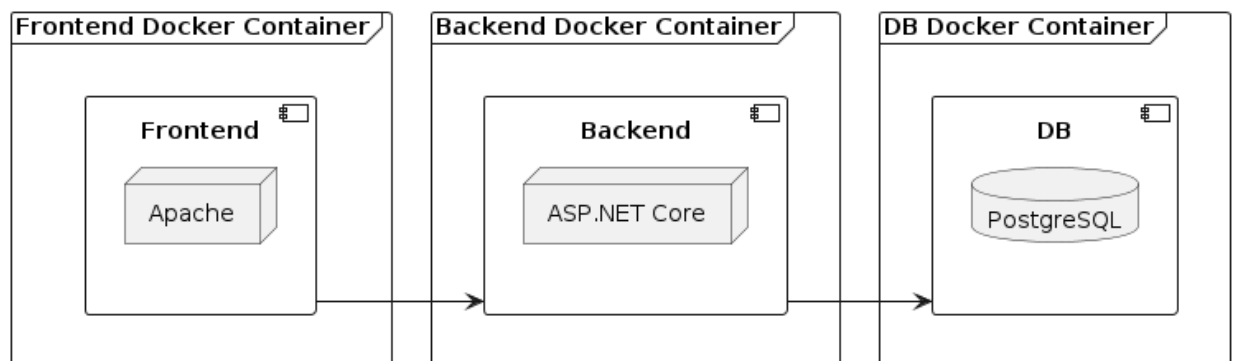


Рисунок 4 - Диаграмма развертывания

3.3 Диаграммы последовательности

На рисунках 5, 6, 7, 8, 9 представлены диаграммы последовательности, они позволяют описать последовательность происходящих событий при взаимодействии пользователя с системой для различных процессов.

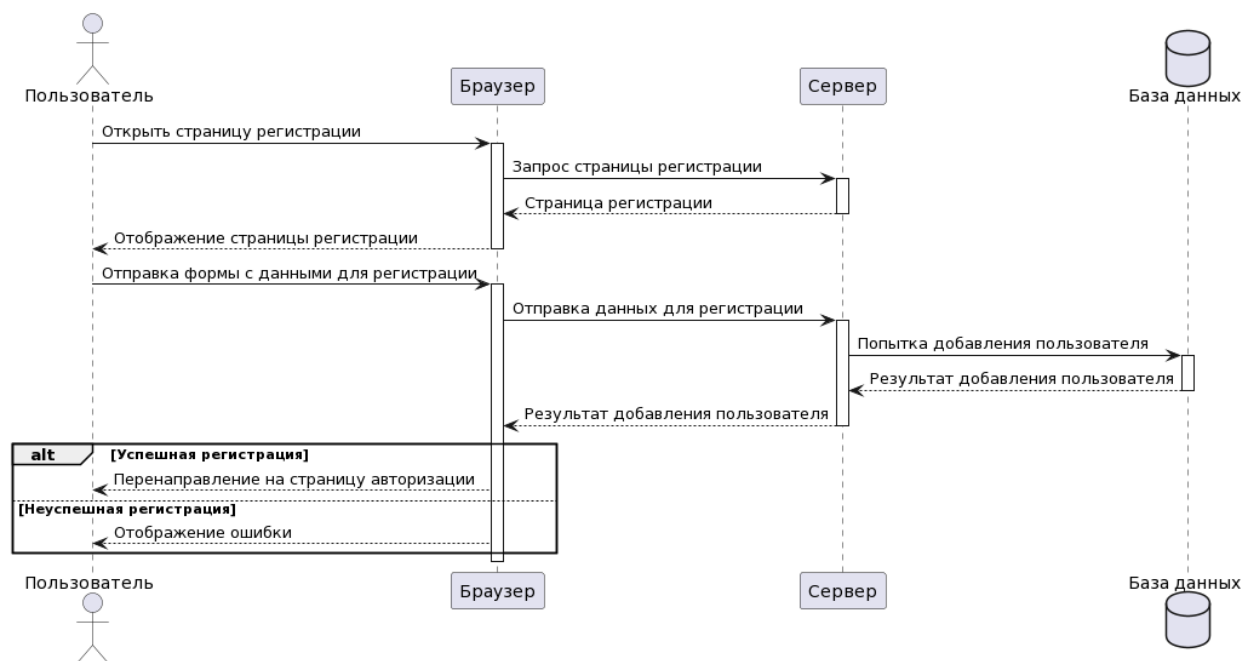


Рисунок 5 - Диаграмма последовательности для процесса регистрации

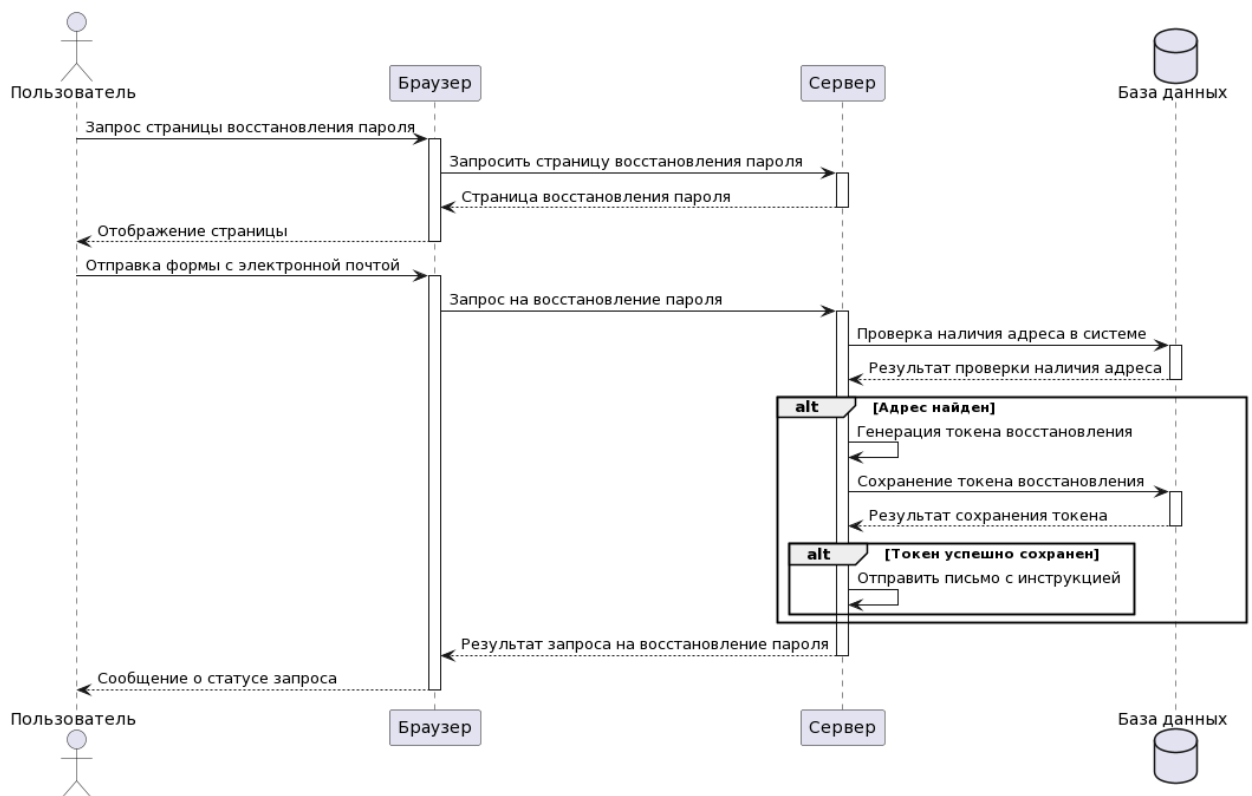


Рисунок 6 - Диаграмма последовательности для процесса запроса восстановления пароля

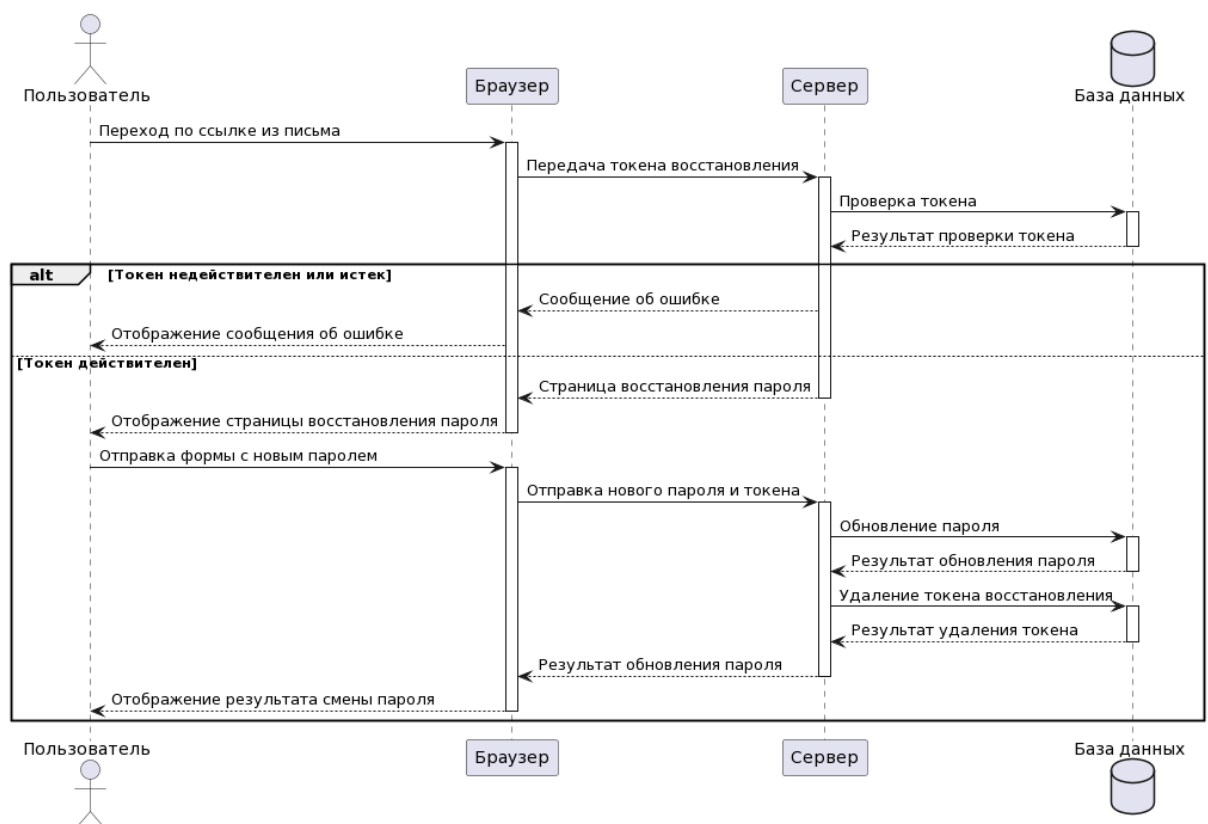


Рисунок 7 - Диаграмма последовательности для процесса восстановления пароля

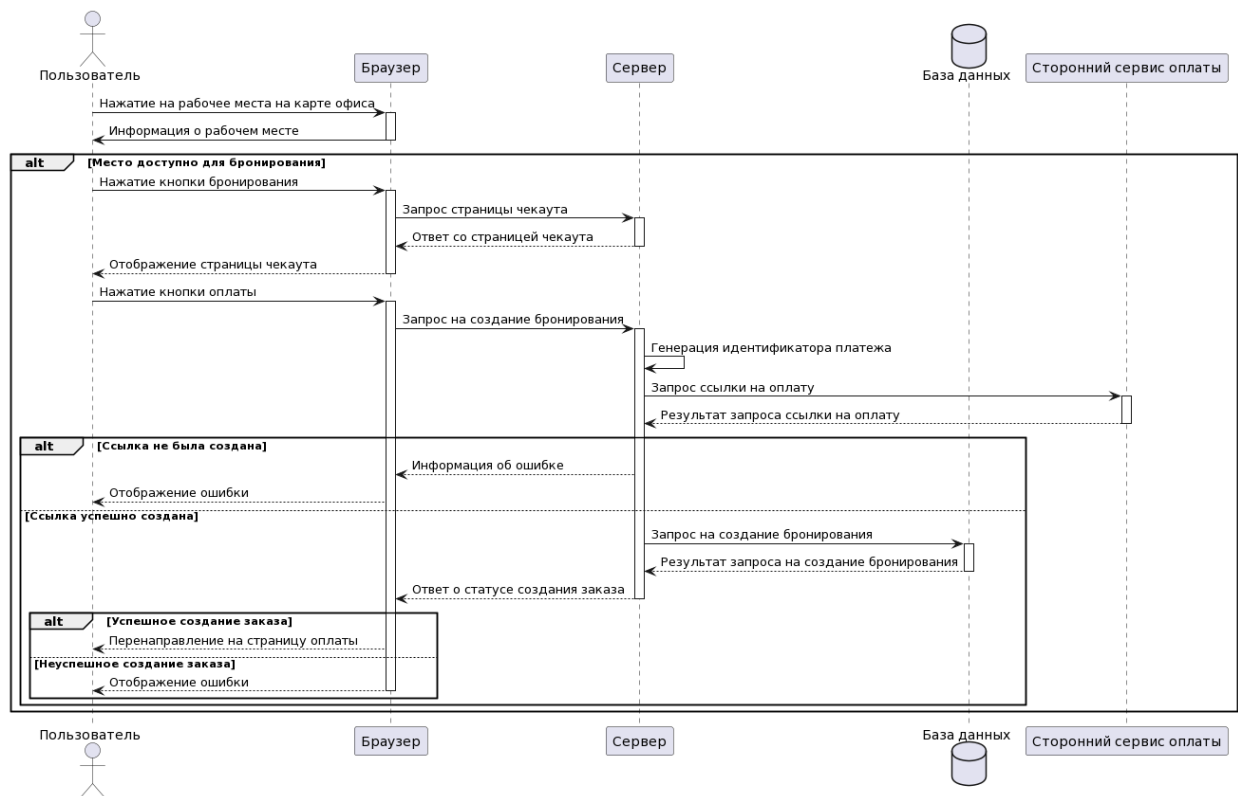


Рисунок 8 - Диаграмма последовательности для процесса создания бронирования

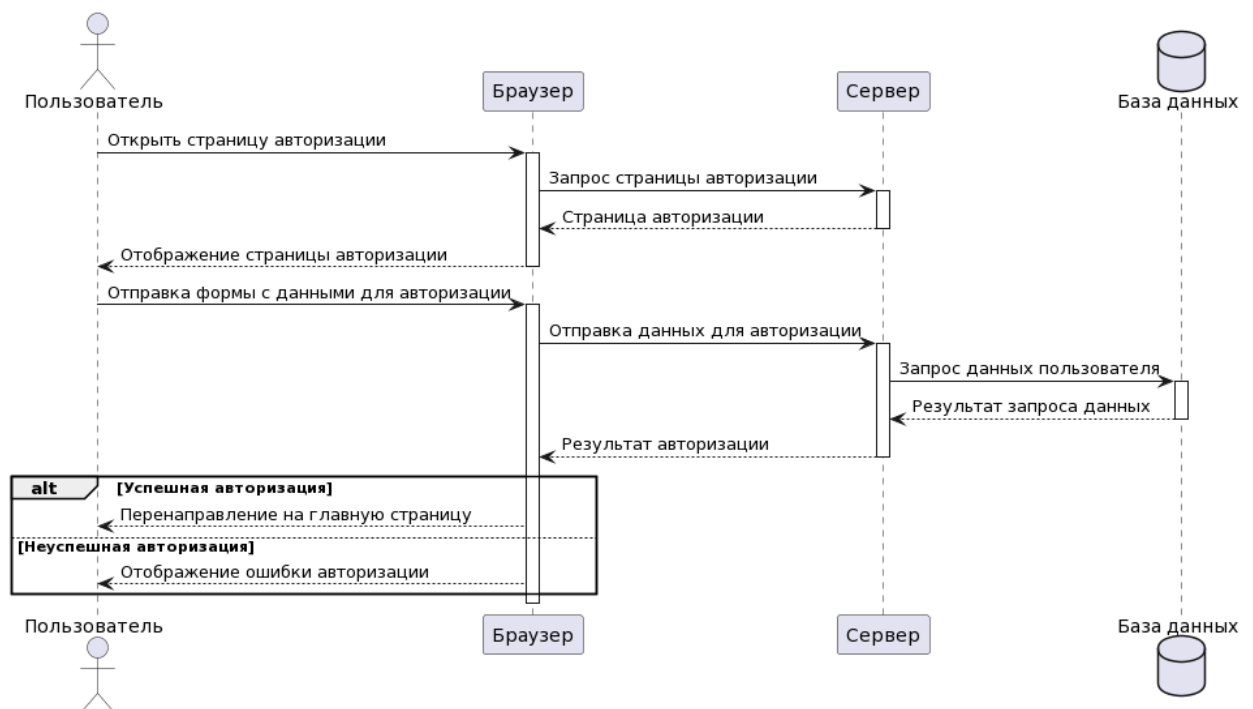


Рисунок 9 - Диаграмма последовательности для процесса авторизации

4 Реализация

Приложение должно быть реализовано с использованием трехуровневой клиент-серверной архитектуры, состоящей из клиента (веб-приложение), сервера и базы данных.

Для реализации серверной части были выбраны следующие средства:

- язык программирования C#;
- фреймворк ASP.NET Core;
- СУБД PostgreSQL.

Для реализации серверной части были выбраны следующие средства:

- HTML;
- CSS;
- язык программирования TypeScript;
- библиотека React.

4.1 Реализация базы данных

На рисунках 10 и 11 представлена ER-диаграмма используемой базы данных

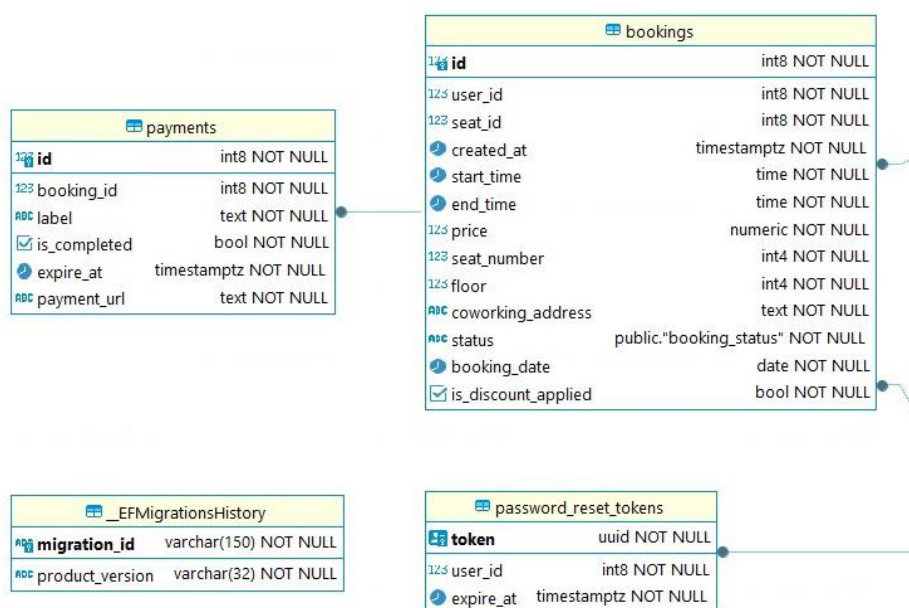


Рисунок 10 – ER-диаграмма, 1 часть

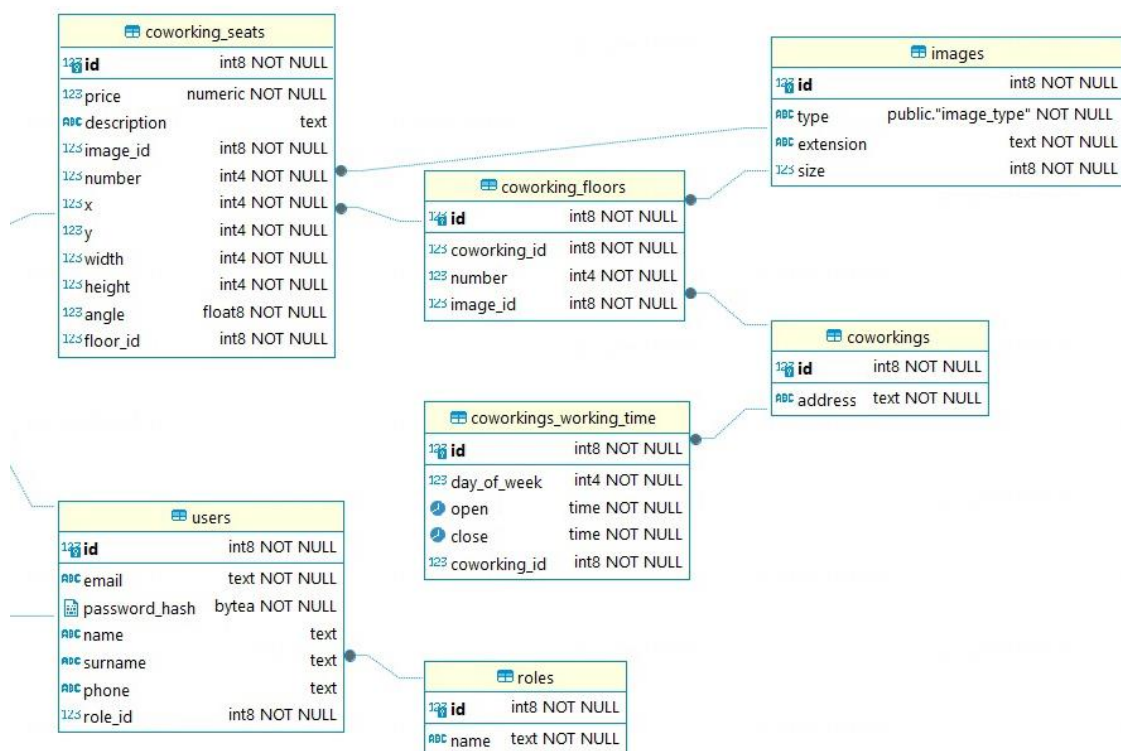


Рисунок 11 - ER-диаграмма, 2 часть

4.2 Реализация серверной части приложения

Серверная часть приложения состоит из следующих компонентов:

- Проект Cower.Domain, состоящий из доменных моделей, которые используются в нескольких слоях приложения (диаграмма классов доменной модели представлена на рисунке 12).
- Проект Cower.Data, являющийся слоем доступа к данным. Содержит в себе репозитории и Dal объекты (репозитории представлены на рисунках 13 и 14).
- Проект Cower.Service, являющийся слоем бизнес логики. В нем находятся сервисы, которые используются слой доступа к данным, и вызывается из слоя презентации (сервисы представлены на рисунках 15 и 16).

—Проект Cower.Web, являющийся слоем презентации. Внутри него находятся REST контроллеры, которые используются клиентской частью.

Настройки приложение берет из переменных среды при запуске.

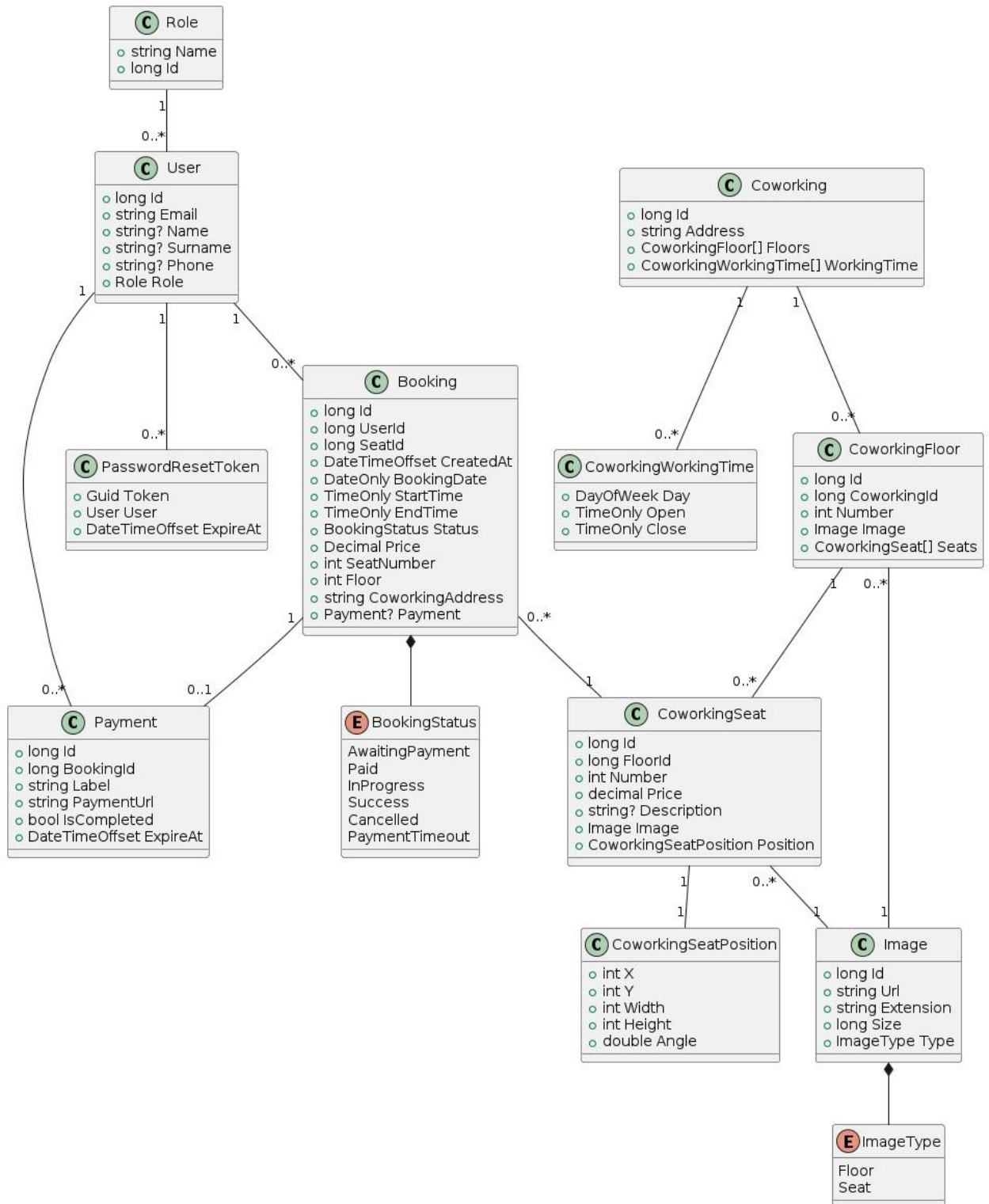


Рисунок 12 – Диаграмма классов доменной модели

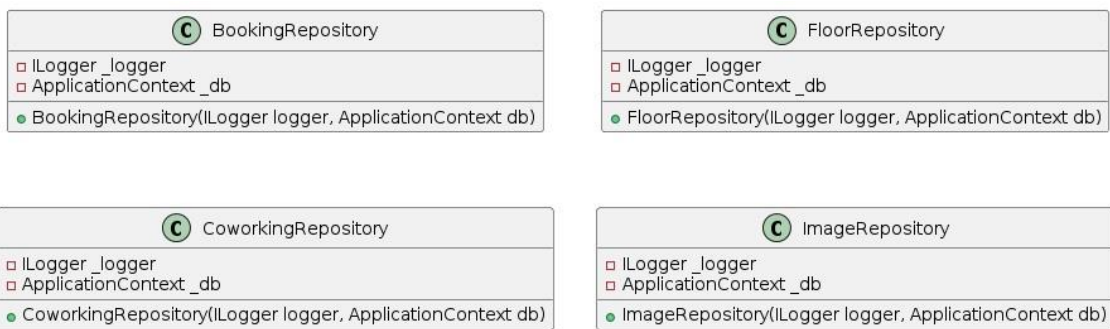


Рисунок 13 – Репозитории, часть 1

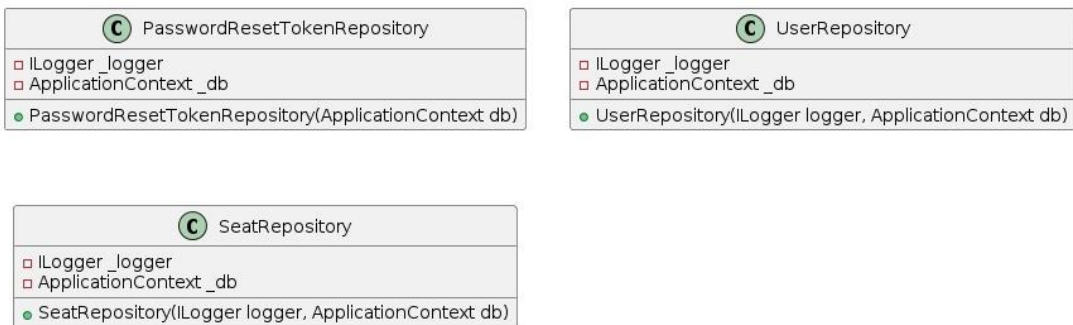


Рисунок 14 – Репозитории, часть 2

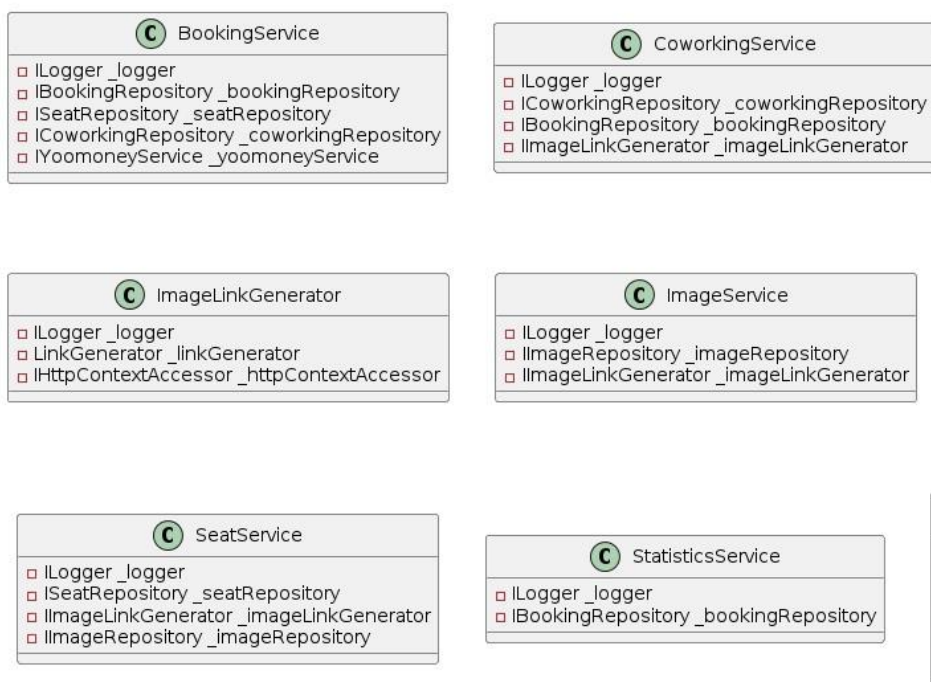


Рисунок 15 – Сервисы, часть 1

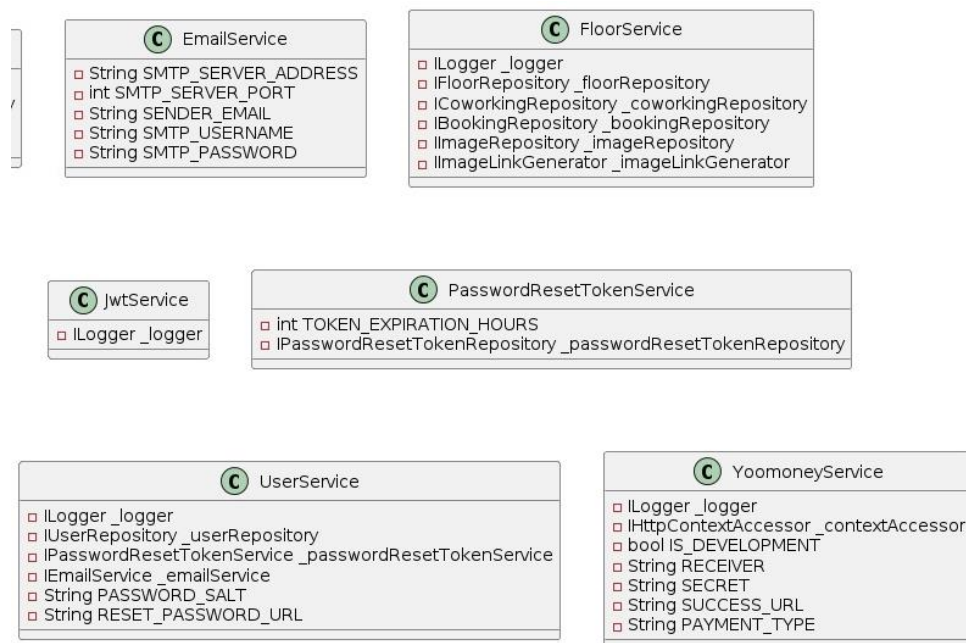


Рисунок 16 – Сервисы, часть 2

4.3 Реализация клиентской части приложения

В данном разделе содержится описание реализации клиентской части приложения.

4.3.1 Общая информация

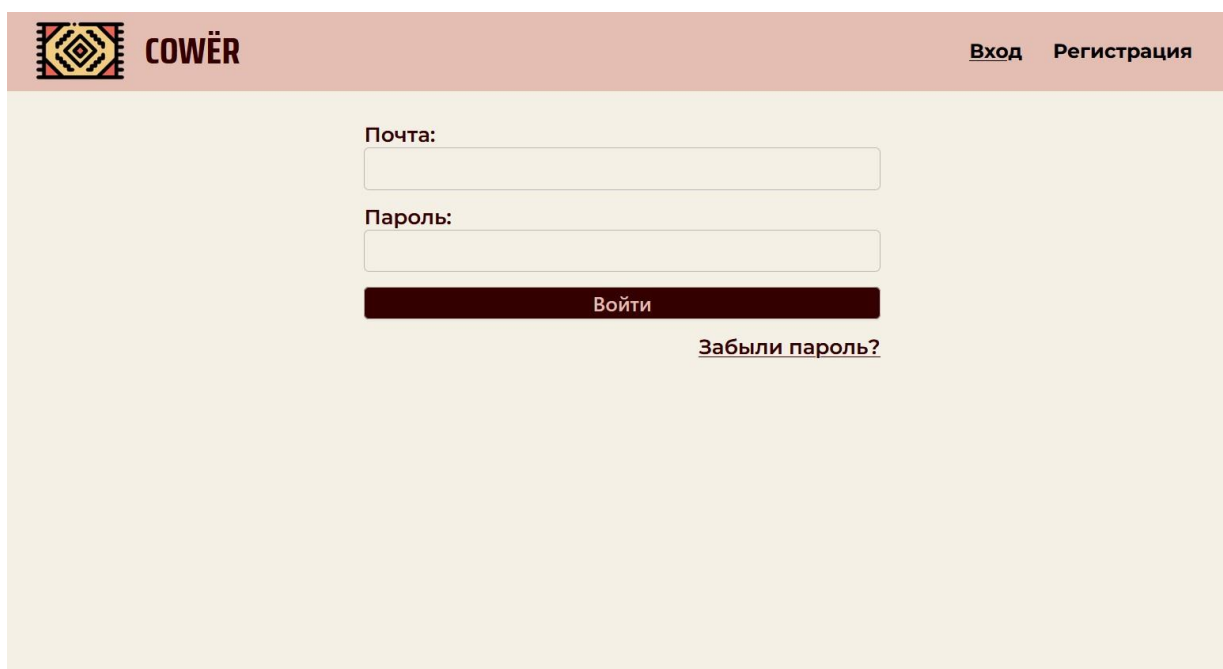
Клиентская часть приложения представляет собой веб-сайт, написанный при помощи языка TypeScript с использованием библиотеки ReactJS и инструмента сборки проекта Vite.js. Пользовательский интерфейс формируется из так называемых компонентов - отдельных логически обособленных частей, которые содержат в себе стили для отображения на странице и логику поведения (примеры компонентов: модальное окно, форма регистрации, навигационная панель). Для стилизации элементов интерфейса использовалась библиотека StyledComponents. Для управления состоянием приложения использовалась библиотека Redux. Для клиентской навигации в приложении использовалась библиотека React Router. Для анализа и

приведения javascript кода к единому стилю использовался программный инструмент ESLint.

4.3.2 Графический интерфейс

Далее будут представлены основные страницы веб-сайта и приведена информация о них.

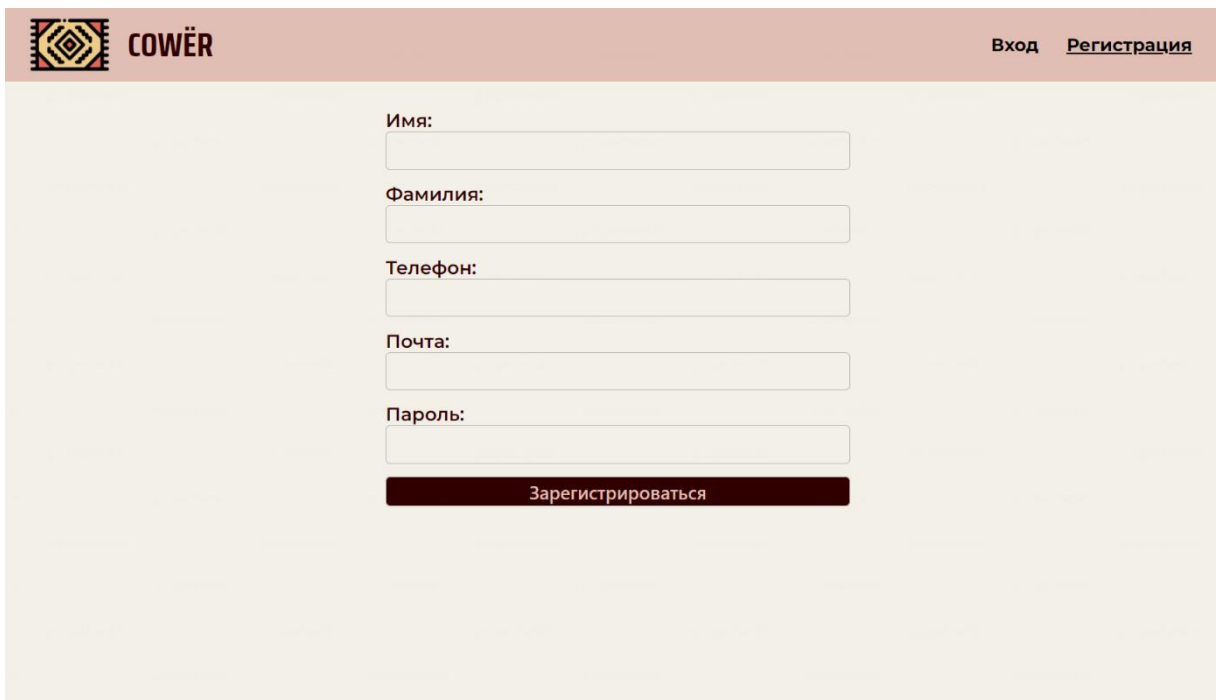
На рисунке 17 представлена страница авторизации. Пользователь может ввести почту и пароль, чтобы войти в аккаунт. При нажатии на кнопку «Регистрация» пользователь переходит на страницу регистрации.



The image shows a web page for user authentication. At the top, there is a red header bar. On the left side of the header is a logo consisting of a square with a geometric pattern, followed by the text "COWÉR". On the right side of the header are two links: "Вход" and "Регистрация". The main content area has a light beige background. In the center, there are two text input fields. The first is labeled "Почта:" and the second is labeled "Пароль:". Below these fields is a dark red button with the text "Войти". To the right of this button is a link "Забыли пароль?".

Рисунок 17 – Страница авторизации

На рисунке 18 представлена страница регистрации. Пользователь может ввести почту и пароль (обязательно), имя, фамилию, номер телефона (опционально) и нажать на кнопку «Зарегистрироваться», чтобы зарегистрироваться.



The image shows a registration form for a website. At the top left is a logo with a stylized 'C' and the word 'COWÉR'. At the top right are links for 'Вход' (Login) and 'Регистрация' (Registration). The form itself is centered and contains five input fields labeled 'Имя:', 'Фамилия:', 'Телефон:', 'Почта:', and 'Пароль:'. Below these fields is a dark red button with the text 'Зарегистрироваться' (Register).

Рисунок 18 – Страница регистрации

На рисунке 20 представлена страница личного кабинета. После нажатия на кнопку «Редактировать» пользователь может ввести имя, фамилию, номер телефона и электронную почту в соответствующие поля, чтобы изменить данные своего личного кабинета (рисунок 19). Также пользователь может нажать на кнопку «Сбросить пароль», чтобы изменить пароль от своего личного кабинета.

[На главную](#)

Личный кабинет

Редактировать

Имя:

Фамилия:

Телефон:

Почта:

Отмена Сохранить

Сбросить пароль

Рисунок 19 – Страница редактирования профиля

[На главную](#)

Личный кабинет

Редактировать

Имя:

Фамилия:

Телефон:

Почта:

Сбросить пароль

Рисунок 20 – Страница личного кабинета

На рисунке 21 представлена главная страница с картой офиса. Пользователь может выбрать адрес офиса, дату и время бронирования, а также нажать на рабочее место на карте офиса и посмотреть информацию о нем. При

нажатию на иконку в правом верхнем углу экрана пользователь может перейти на станицу личного кабинета, на страницу со списком заказов или выйти.

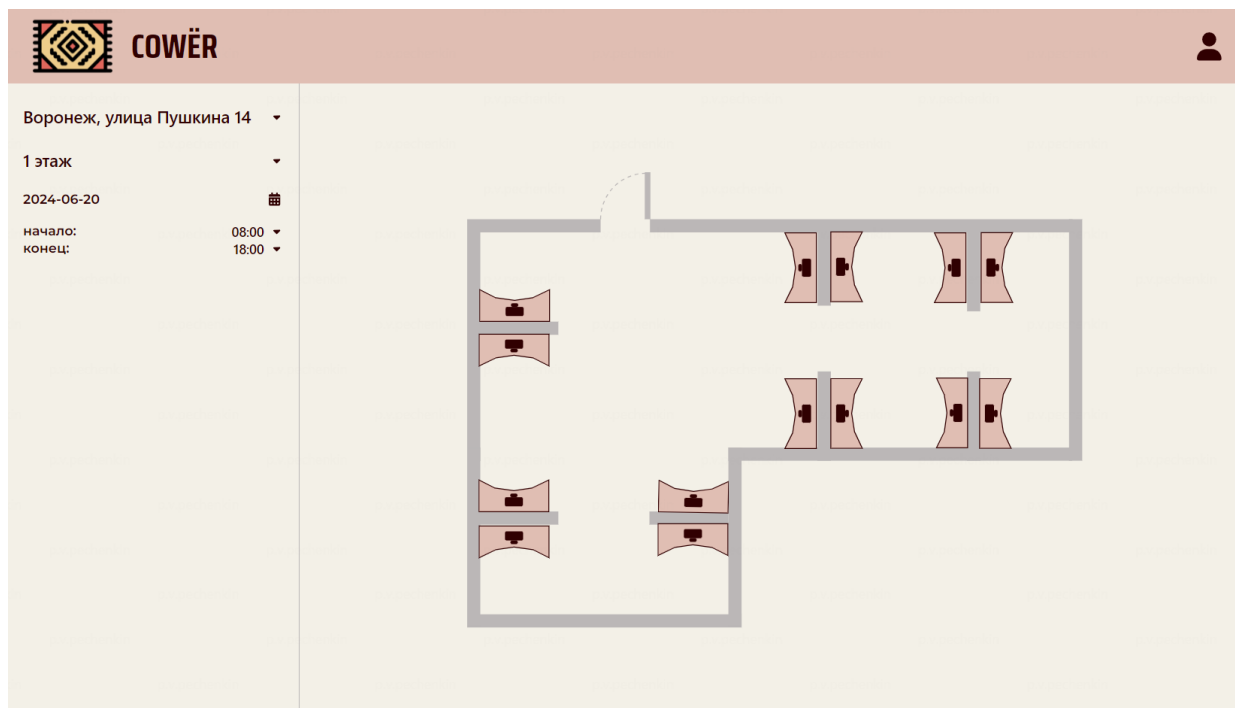


Рисунок 21 – Главная страница

На рисунке 22 представлена страница чекаута с оплатой, на которую пользователь переходит после нажатия на кнопку «Забронировать» на главной странице.

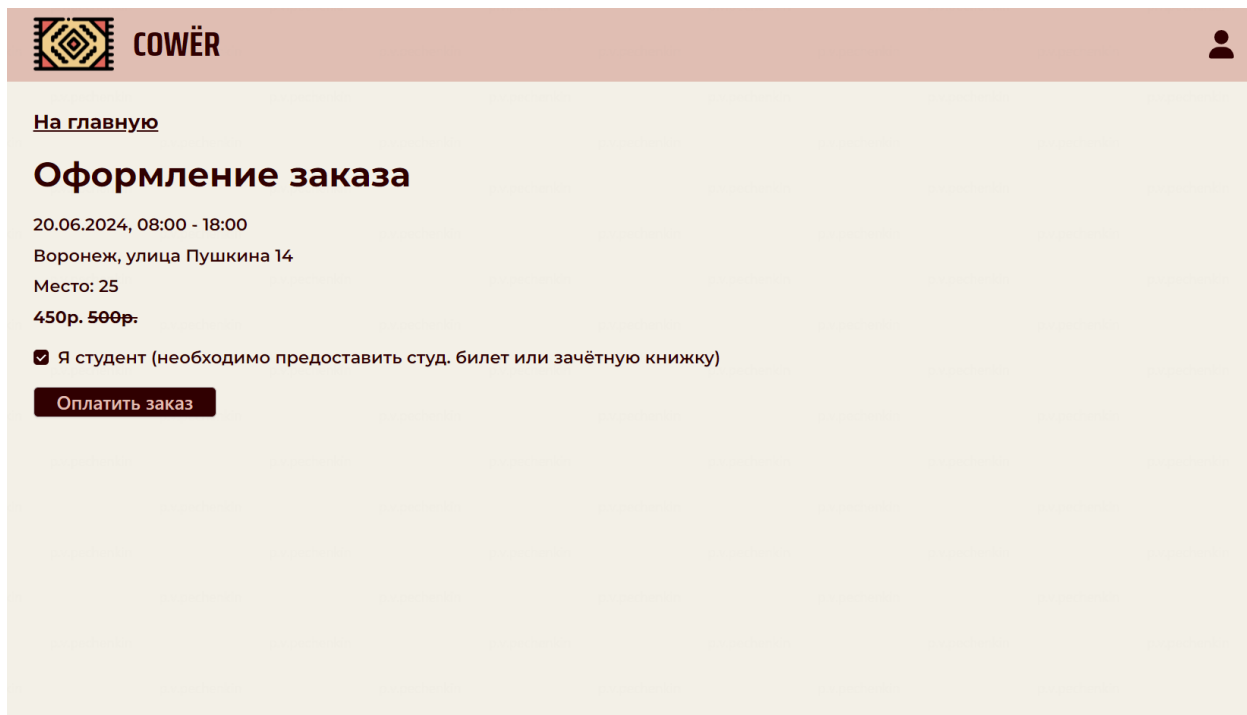


Рисунок 22 – Чекаут с оплатой

На рисунке 23 представлена страница со списком заказов. Пользователь может оплатить неоплаченный заказ или отменить заказ, если время, на которое совершена бронь, еще не наступило.

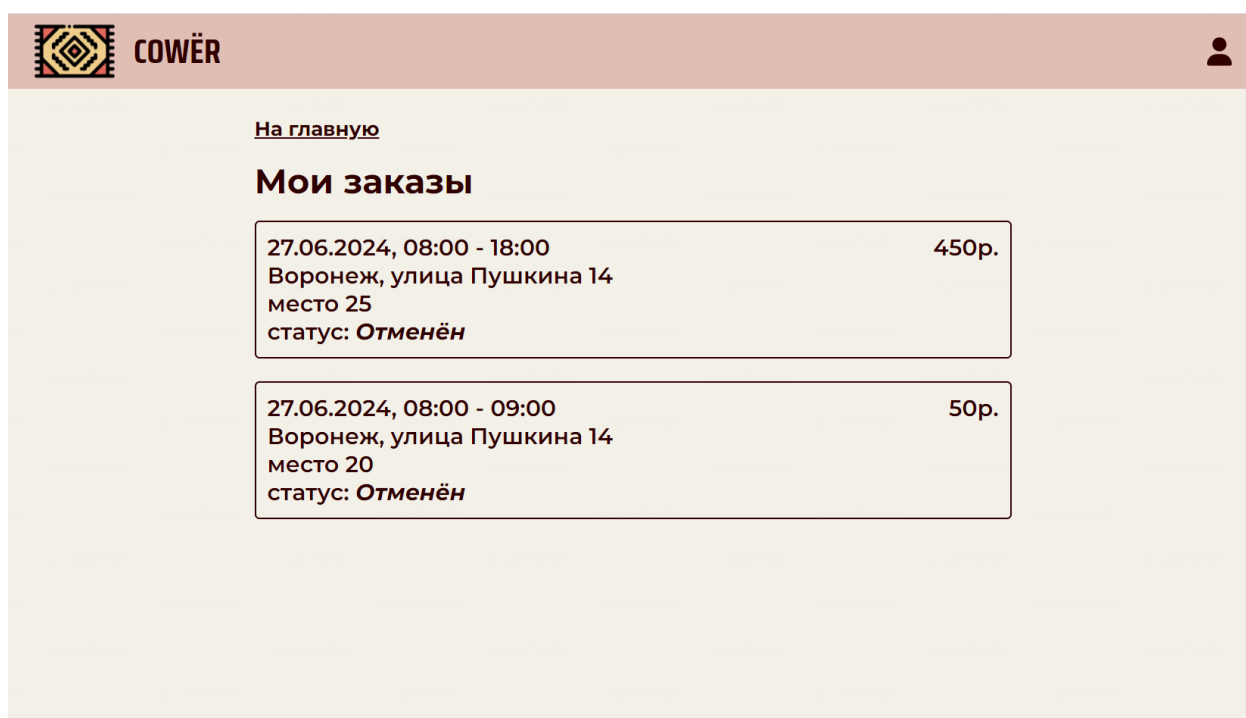


Рисунок 23 – Страница со списком заказов

На рисунке 24 представлена страница запроса на восстановление пароля. Пользователь может ввести электронную почту и нажать на кнопку «Отправить письмо на почту», после чего пользователю на электронную почту придёт письмо со ссылкой, перейдя по которой пользователь попадет на страницу восстановления пароля.

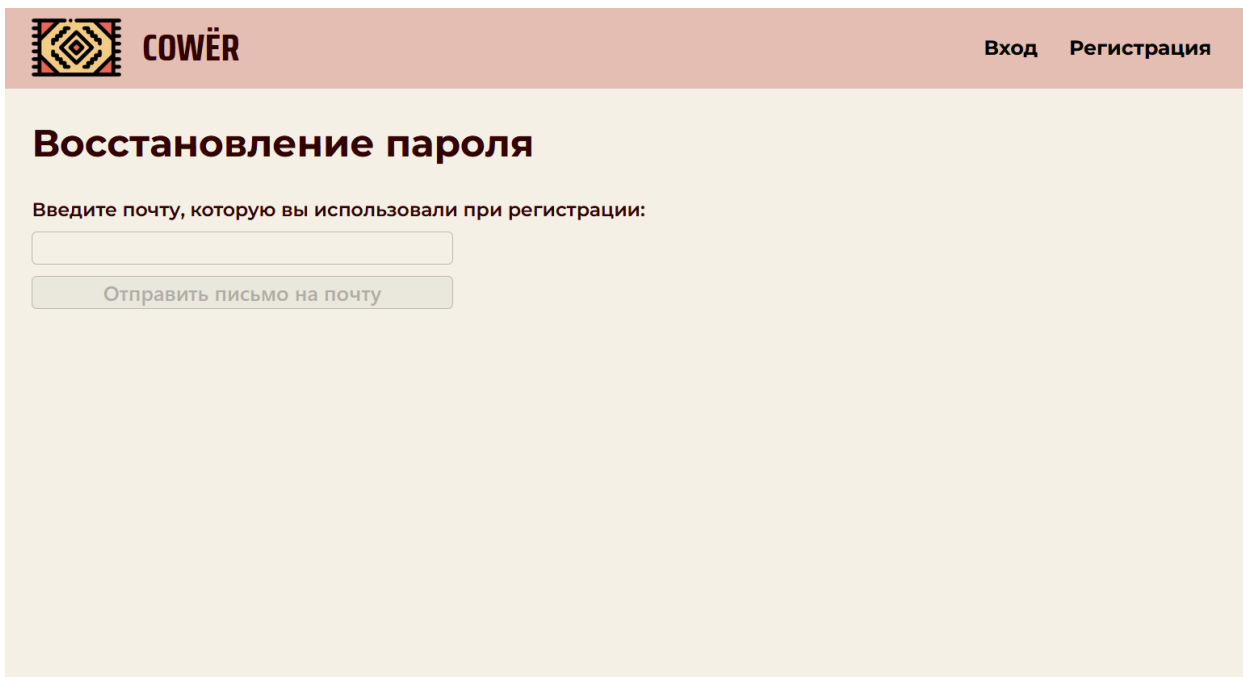


Рисунок 24 – Страница запроса на восстановление пароля

На рисунке 25 представлена страница администратора для редактирования планировки офиса. Администратор может добавить, удалить, переместить рабочее место, изменить его описание. Также администратор может добавлять и удалять коворкинги и этажи в коворкингах.

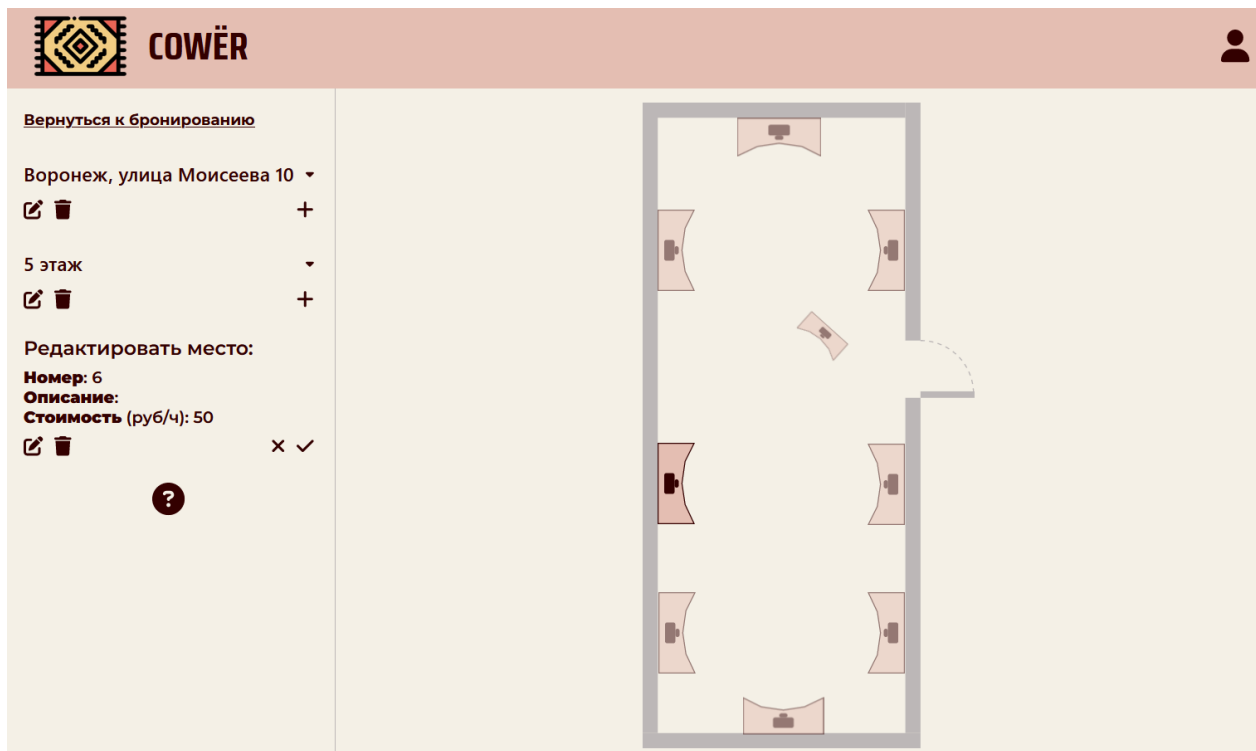


Рисунок 25 – Страница администратора для редактирования планировки
офиса

5 Тестирование

Перечень проводимых испытаний и их результаты приведены в документах «Программа и методика испытаний» и «Протокол проведения испытаний».

ЗАКЛЮЧЕНИЕ

В ходе реализации данного проекта были достигнуты поставленные задачи. Был разработан веб-сервис «Cowër», предназначенный для бронирования рабочих мест в коворкинге. В разработанном сервисе предусмотрена возможность просмотра карты офиса, выбора на ней рабочего места и последующего его бронирования.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Исследуем, как растет рынок коворкинга в России [Электронный ресурс]. – Режим доступа: https://bankspace.ru/blog/koworking_research (Дата обращения: 20.03.2024).
2. WORKKODE [Электронный ресурс] – Режим доступа: <https://wkode.co/> (Дата обращения: 29.04.2024).
3. GrowUp [Электронный ресурс] – Режим доступа: <https://growup-coworking.ru/> (Дата обращения: 20.03.2023).
4. Ясная поляна [Электронный ресурс] – Режим доступа: <https://coworkingspb.ru/coworking#openspaces> (Дата обращения: 20.03.2023).