

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

Факультет Компьютерных наук

Кафедра информационных технологий управления

Сервис бронирования услуг красоты и здоровья «Fast
booking»

Курсовая работа по дисциплине
«Технологии программирования»

09.03.02 Информационные системы и технологии
Информационные системы и технологии в управлении
предприятием

Зав. кафедрой _____ д.т.н., профессор М.Г. Матвеев _____. 20__

Обучающийся _____ А.Г. Самсонова, 3 курс, д/о

Обучающийся _____ Е.И. Маслова, 3 курс, д/о

Обучающийся _____ К.Р. Денисов, 3 курс, д/о

Руководитель _____ В.С. Тарасов, ст. преподаватель

Руководитель _____ В.А. Ушаков, преподаватель

Воронеж 2024

Содержание

Содержание	2
Введение	4
1 Постановка задачи	6
1.1 Требования к разрабатываемой системе.....	6
1.1.1 Функциональные требования.....	6
1.1.2 Нефункциональные требования.....	7
1.2 Требования к архитектуре	7
1.3 Задачи, решаемые в процессе разработки	8
2 Анализ предметной области.....	9
2.1 Терминология (гlossарий предметной области).....	9
2.2 Обзор аналогов.....	10
2.2.1 Fresha	10
2.2.2 Clickerbook	13
2.3 Диаграммы, отражающие функциональность веб-приложения.....	17
2.3.1 Диаграмма прецедентов (Use case)	17
2.3.2 Диаграмма последовательности (Sequence diagram).....	19
2.3.3 Диаграмма состояний (Statechart diagram)	20
2.3.4 Диаграмма активностей (Activity diagram).....	21
2.3.5 Диаграмма объектов (Object diagram).....	21
2.3.6 Диаграмма классов (Class diagram).....	22
2.3.7 Диаграмма сотрудничества (Collaboration diagram).....	22
2.3.8 Диаграмма развёртывания (Deployment diagram)	24
2.3.9 ER-диаграмма.....	25
3 Реализация	26

3.1 Средства реализации.....	26
3.1.1 Средства реализации серверной части приложения	26
3.1.2 Средства реализации клиентской части приложения	27
3.2 Реализация серверной (backend) части приложения	28
3.3 Реализация клиентской (frontend) части приложения.....	30
3.4 Навигация по приложению.....	32
3.4.1 Для неавторизованного пользователя.....	32
3.4.2 Для авторизованного пользователя	35
3.4.3 Для администратора	38
4 Тестирование	42
4.1 Дымовое тестирование	42
4.2 Тестирование пользовательского интерфейса.....	44
Заключение	46
Список использованной литературы.....	48

Введение

В настоящее время Интернет-технологии стали неотъемлемой частью современного мира, ведь именно они позволяют получать доступ к большому количеству информации, не выходя из дома. В связи с этим все больше компаний и индивидуальных предпринимателей начинают использовать Интернет для продажи своих товаров и услуг. В этом контексте онлайн-сервисы бронирования играют важную роль, предоставляя пользователям возможность ознакомиться с ассортиментом возможных услуг, и предоставляя детальную информацию о существующих свободных местах для записи, а также об уникальных характеристиках услуг и их ценах.

Одной из наименее популярных областей, где используются онлайн-сервисы бронирования, является сфера красоты и здоровья. Сервисы бронирования помогают людям ознакомиться с существующими услугами и выбрать нужные на данный момент клиенту. К сожалению, на сегодняшний день существует мало сервисов бронирования услуг красоты и здоровья, что заставляет клиентов звонить для записи, что возможно только в рабочие часы, а не в любой удобный момент. Сайты салонов красоты и спортивных залов обеспечивают привлекательную визуальную презентацию помещений и самих услуг, что может заинтересовать клиента и способствует принятию решения о записи, но предпринимателям зачастую сложно найти сервис, который позволил бы посредством рекламы на стороннем сайте обеспечивать себе более высокий поток клиентов.

Время – очень ценный ресурс в современном мире, а для его экономии необходимо выполнение некоторых задач в нерабочее время. Онлайн-сервисы бронирования услуг красоты и здоровья, несомненно, помогают в этом деле, они важны для людей, у которых нет возможности позвонить с рабочего места, чтобы записаться, например, в парикмахерскую.

В данной курсовой работе был реализован сервис бронирования услуг красоты и здоровья «Fast booking», который будет предоставлять пользователям подробную информацию об услугах и экономить время из-за отсутствия долгих звонков для записи.

1 Постановка задачи

Данный проект предназначен для просмотра доступных в Воронеже центров красоты и здоровья (фитнес – центры, спа, салоны красоты) и бронирования их услуг в режиме онлайн, что позволяет клиентам не задумываться о рабочих часах центра и личном общении с администратором.

1.1 Требования к разрабатываемой системе

1.1.1 Функциональные требования

К разрабатываемому приложению выдвигаются следующие функциональные требования:

- Осуществление просмотра центров по категориям (салоны красоты, спа, фитнес-центры);
- Получение информации о предоставляемых услугах;
- Просмотр рейтинга и отзывов об интересующем пользователя центре;
- Возможность просмотра отзывов других пользователей.
- Добавление выбранных центров в список «Избранное» авторизованным пользователем;
- Удаление из списка «Избранное» выбранного центра авторизованным пользователем;
- Запись в выбранный центр на желаемую услугу авторизованным пользователем;
- Возможность оставить свой отзыв и поставить оценку конкретному центру авторизованным пользователем;
- Добавление и удаление центров в список сайта администратором;
- Редактирование информации о конкретном центре и его услугах администратором.

1.1.2 Нефункциональные требования

К разрабатываемому приложению выдвигаются следующие нефункциональные требования:

- Приложение должно отвечать на запросы пользователей в течение нескольких секунд;
- Приложение должно обладать интерфейсом, выполненным в едином стиле со всем необходимым набором функций, чтобы с ним могли работать пользователи различных возрастных и культурных групп;
- Приложение должно использовать современные технологии и инструменты разработки.

1.2 Требования к архитектуре

Список требований к архитектуре:

- Приложение должно быть построено с использованием протоколов HTTP;
- Для хранения информации необходимо использовать реляционную базу данных;
- Клиентская часть приложения должна быть написана с использованием технологий frontend разработки, таких как HTML, CSS, JavaScript с фреймворком React. Выбор этого фреймворка объясняется тем, что он обладает простым синтаксисом, позволяет обновлять только те элементы, которые требуют изменений и использовать повторно уже существующие элементы;
- Серверная часть приложения должна быть написана с использованием технологий backend разработки, таких как Java и Spring на основе архитектурного паттерна MVC. Выбор этого фреймворка объясняется тем, что он позволяет управлять данными

как на этапе разработки, так и на стадии публикации. Также Spring предоставляет возможность использовать админ-панель в приватном режиме для внутреннего управления данными, обеспечивая защиту конфиденциальности.

1.3 Задачи, решаемые в процессе разработки

Процесс организации данного веб-приложения построен на основе гибкой методологии Kanban.

В процессе разработки сервиса бронирования услуг красоты и здоровья будут решаться следующие задачи:

- Анализ предметной области: необходимо изучить особенности работы сервиса бронирования услуг красоты и здоровья;
- Проектирование базы данных: на основе полученных требований необходимо разработать структуру базы данных, которая будет использоваться в приложении;
- Разработка серверной части приложения: на этом этапе необходимо разработать серверную часть приложения, которая будет отвечать за обработку запросов клиента и взаимодействие с базой данных. Для этого используется фреймворк Spring;
- Разработка клиентской части приложения: клиентская часть приложения должна быть написана с использованием современных технологий frontend разработки, таких как HTML, CSS, JavaScript;
- Тестирование и отладка: на этом этапе необходимо провести тестирование и отладку приложения, чтобы убедиться, что оно соответствует требованиям, определенным в начале проекта.

2 Анализ предметной области

2.1 Терминология (гlossарий предметной области)

Веб-приложение - клиент-серверное приложение, в котором клиент взаимодействует с веб-сервером при помощи браузера.

Фреймворк - программная платформа, определяющая структуру программной системы; программное обеспечение, облегчающее разработку и объединение разных компонентов большого программного проекта.

Клиент (клиентская сторона) - сайт, который предоставляет пользователю возможность взаимодействовать со всей системой.

Сервер (серверная часть) - компьютер, обслуживающий другие устройства (клиентов) и предоставляющий им свои ресурсы для выполнения определенных задач.

MVC – схема разделения данных приложения и управляющей логики на три отдельных компонента: модель, представление и контроллер - таким образом, что модификация каждого компонента может осуществляться независимо.

Backend - логика работы сайта, внутренняя часть продукта, которая находится на сервере и скрыта от пользователя.

Frontend - презентационная часть информационной или программной системы, ее пользовательский интерфейс и связанные с ним компоненты.

CSS - формальный язык описания внешнего вида документа, написанного с использованием языка разметки (HTML, XHTML, XML).

HTML - стандартизированный язык гипертекстовой разметки веб-страниц в браузере.

GitHub - веб-сервис для хостинга IT-проектов и их совместной разработки.

JavaScript - язык программирования высокого уровня, который используется для написания frontend- и backend-частей сайтов, а также мобильных приложений.

PostgreSQL - реляционная база данных с открытым кодом.

React — JavaScript-библиотека для создания пользовательских интерфейсов.

2.2 Обзор аналогов

2.2.1 Fresha

Сервис бронирования услуг в области красоты и здоровья «Fresha» предоставляет возможность размещения рекламы для малого и большого бизнеса. В список доступных для бронирования центров входят как сетевые места, так и небольшие заведения, которые только начинают путь в бизнесе beauty-услуг.[1]

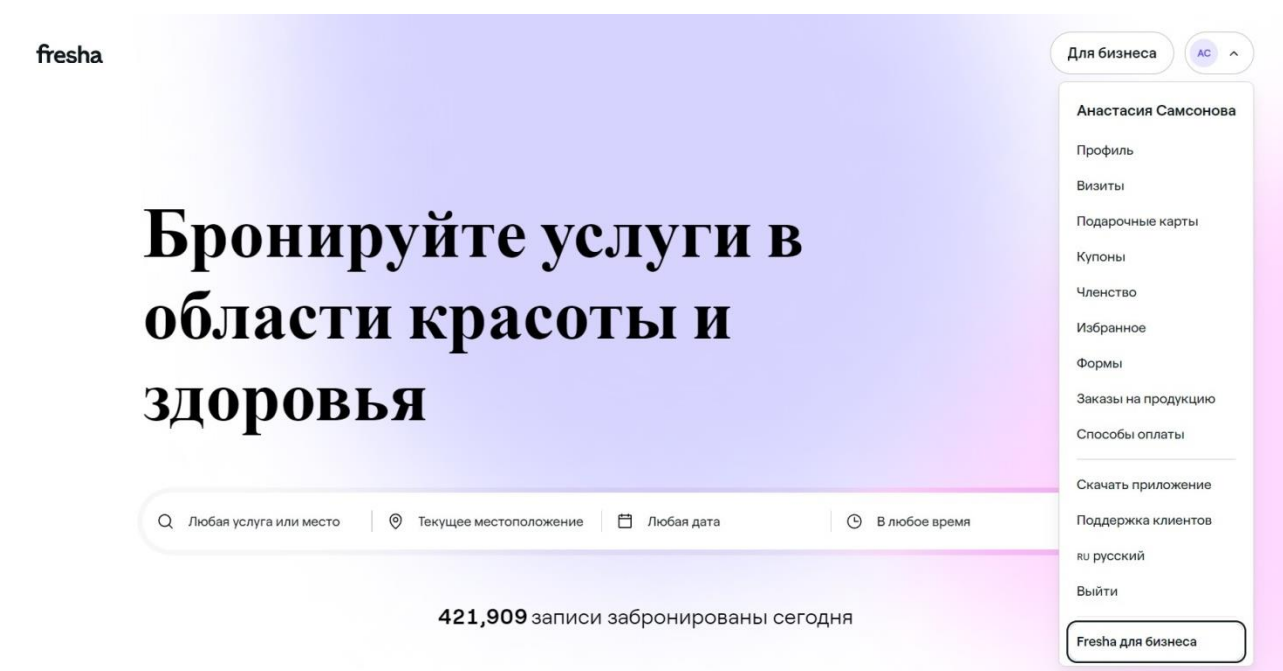


Рисунок 1 - Главный экран сайта Fresha

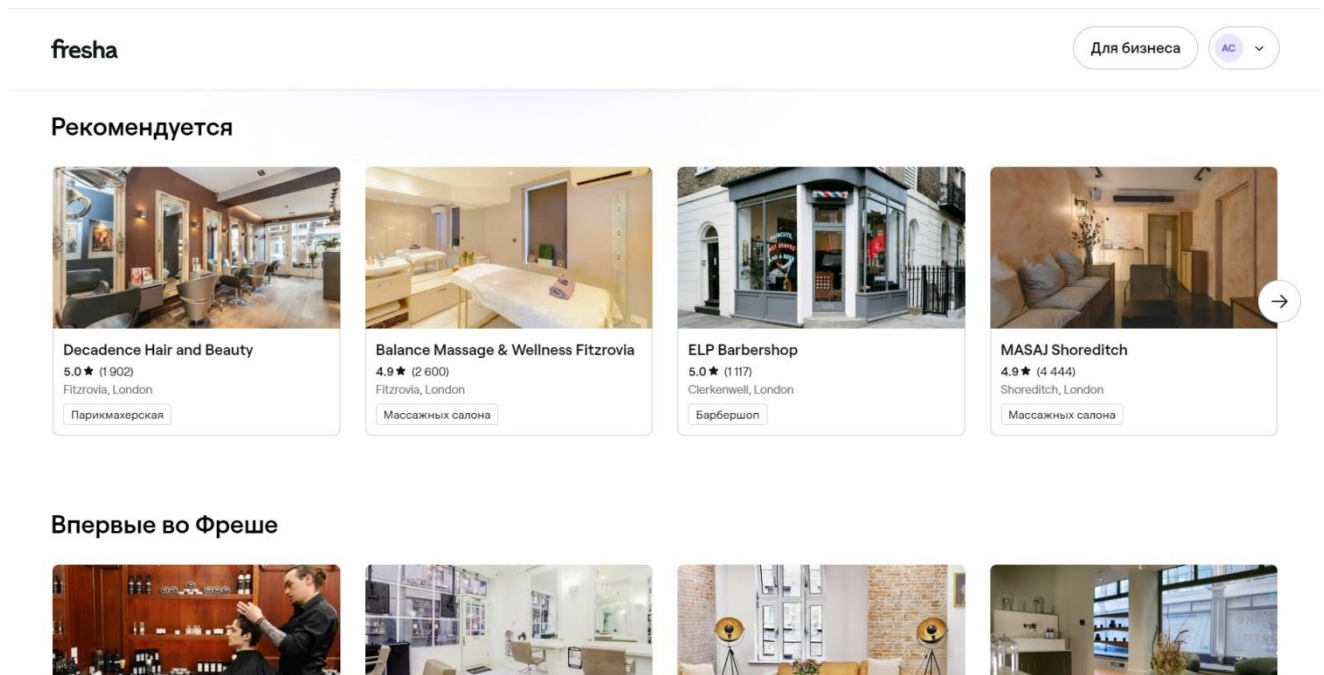


Рисунок 2 - Вторая часть главного экрана

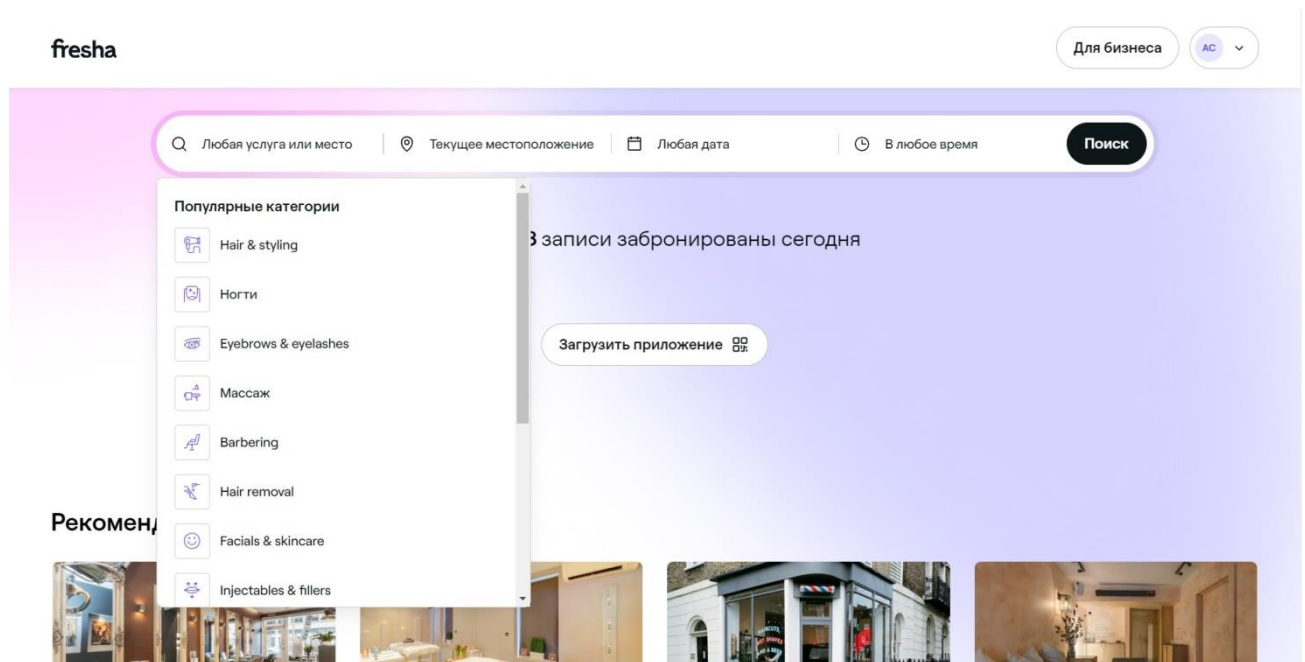


Рисунок 3 - Выбор услуги, города и времени

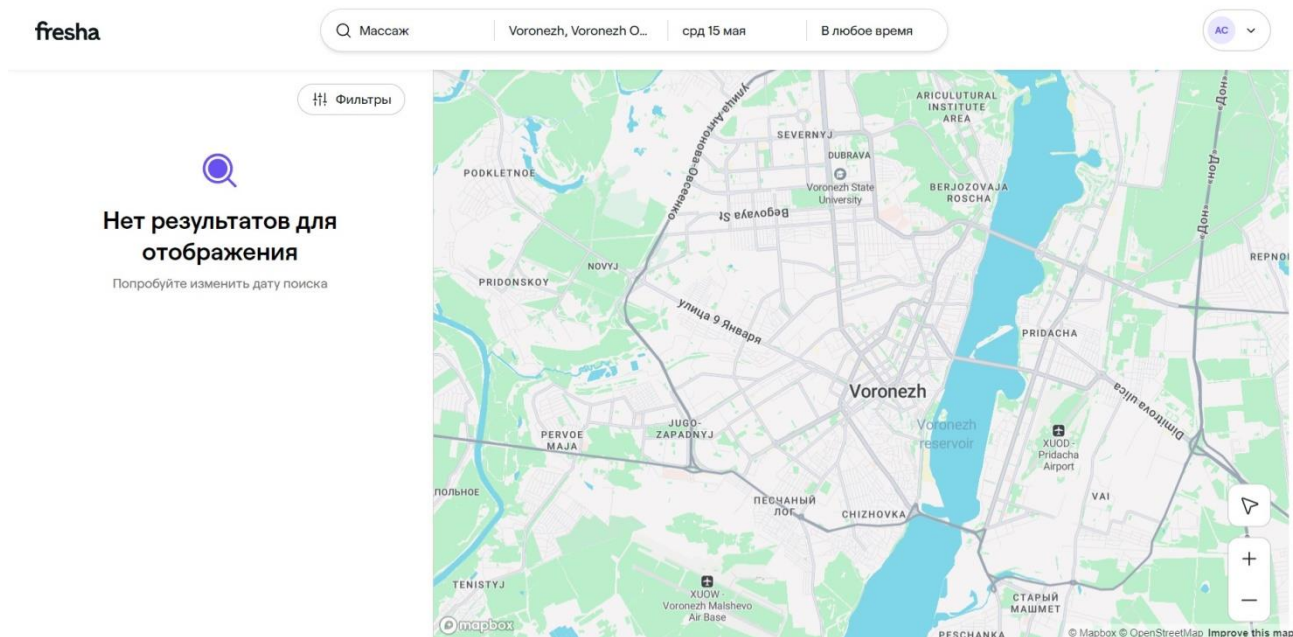


Рисунок 4 - Результаты запроса выбранной услуги

К достоинствам данного сервиса относятся:

- Большой выбор доступных центров;
- Удобный интерфейс для просмотра интересующих клиента мест;
- Возможность подать заявку на добавление в базу доступных центров, что в свою очередь работает как реклама;
- Возможность просмотра мастеров;
- Возможность просмотра рейтингов и отзывов.

К недостаткам данного сервиса можно отнести:

- Не работает в России;
- Добавление центра услуг красоты и здоровья в базу сайта является платным.

2.2.2 Clickerbook

«Clickerbook» – это система онлайн-бронирования 24/7 для индустрии красоты и фитнеса, которая предоставляет клиентам легкий доступ к услугам и скидкам в любом месте и в любое время, а также предоставляет салонам, фитнес-клубам и спа-центрам простой способ рекламировать свои услуги и управлять своими заказами.[2]

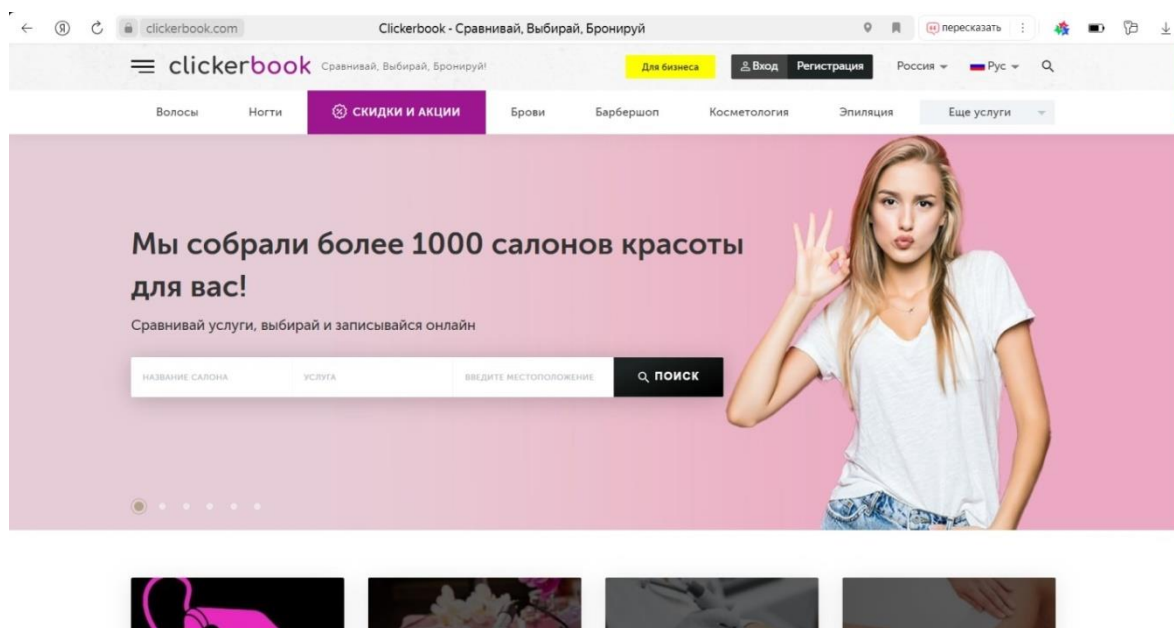


Рисунок 5 - Главный экран сайта Clickerbook

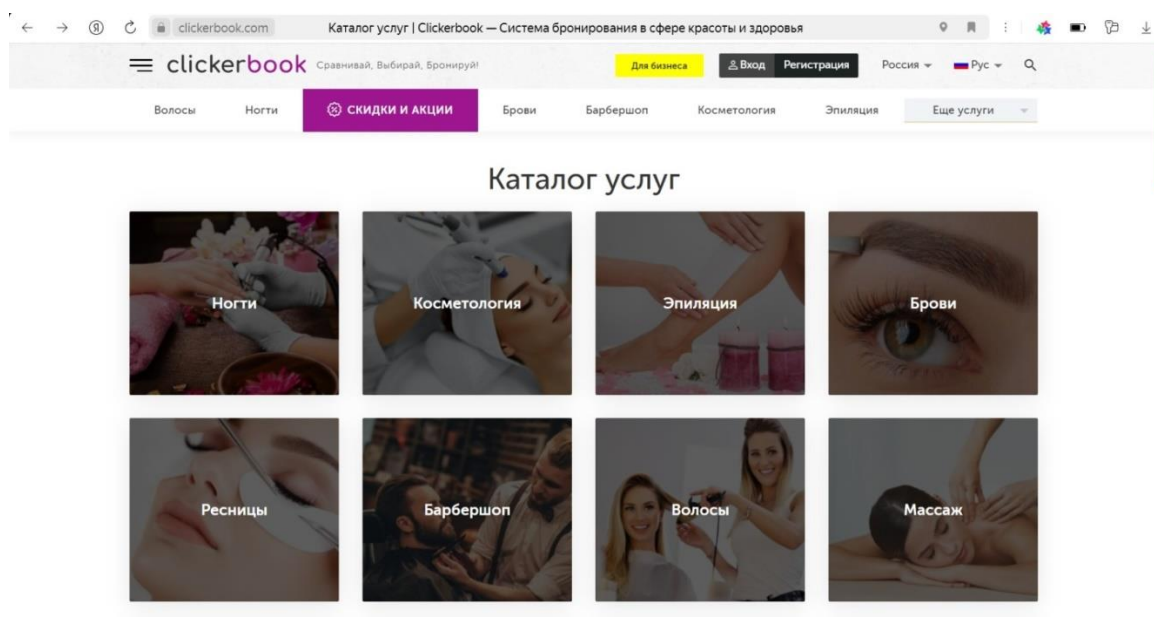


Рисунок 6 - Каталог услуг

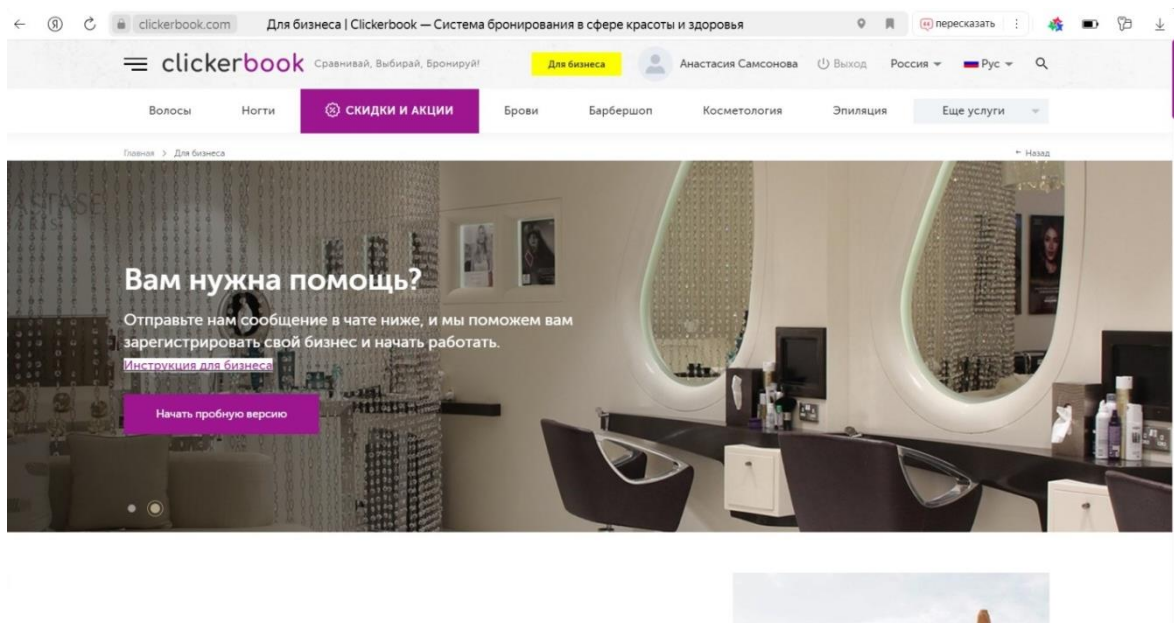


Рисунок 7 - Страница «Для бизнеса»

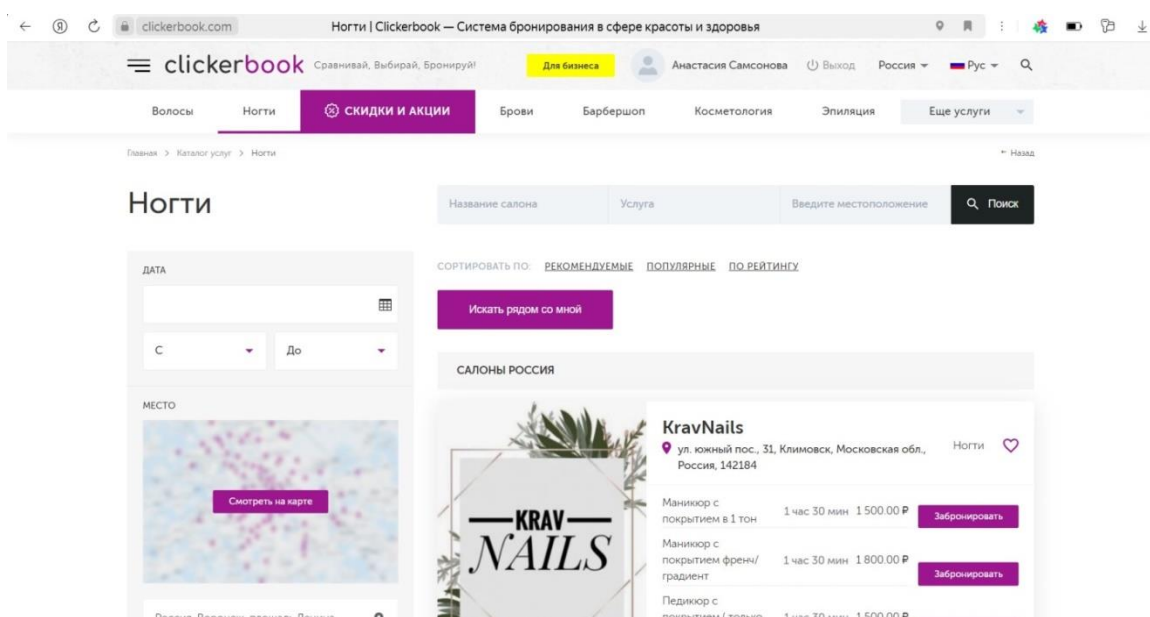


Рисунок 8 - Страница «Ногти» со списком салонов

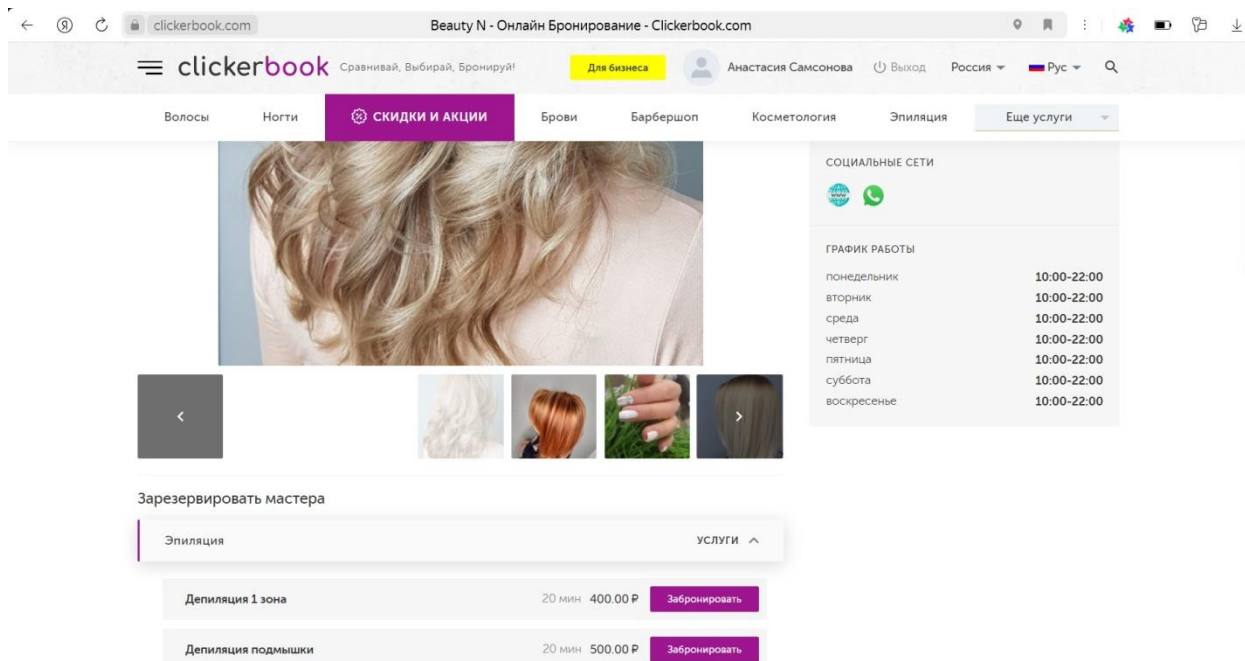


Рисунок 9 - Карточка салона

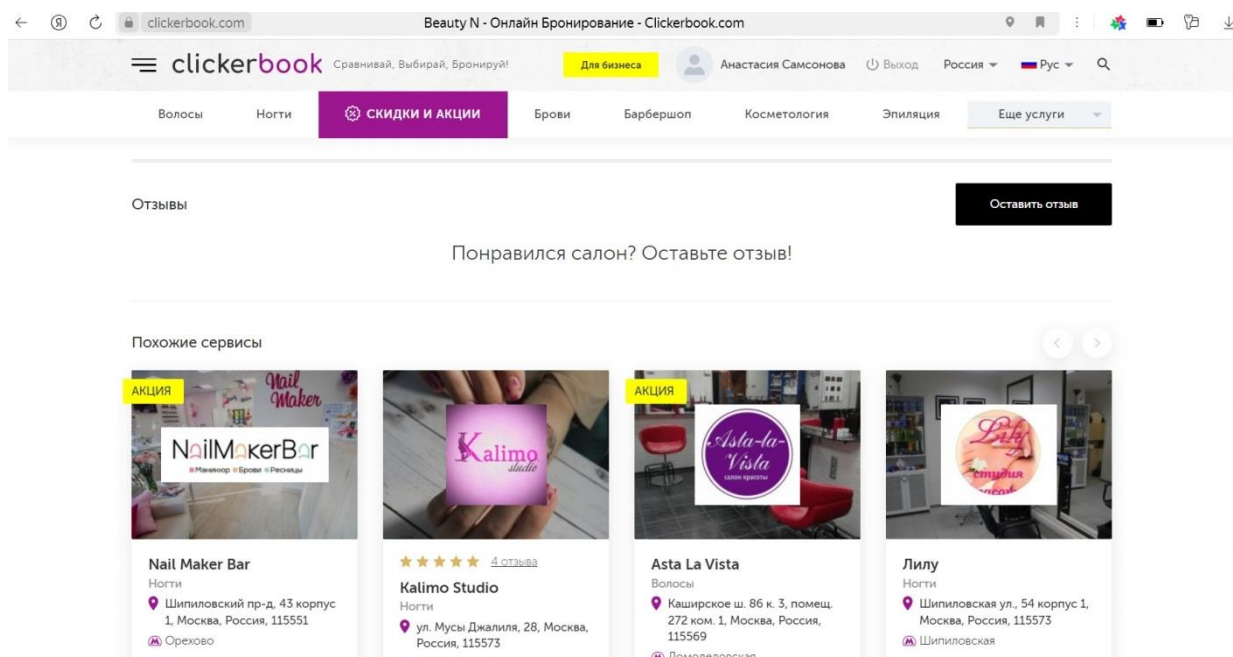


Рисунок 10 - Отзывы

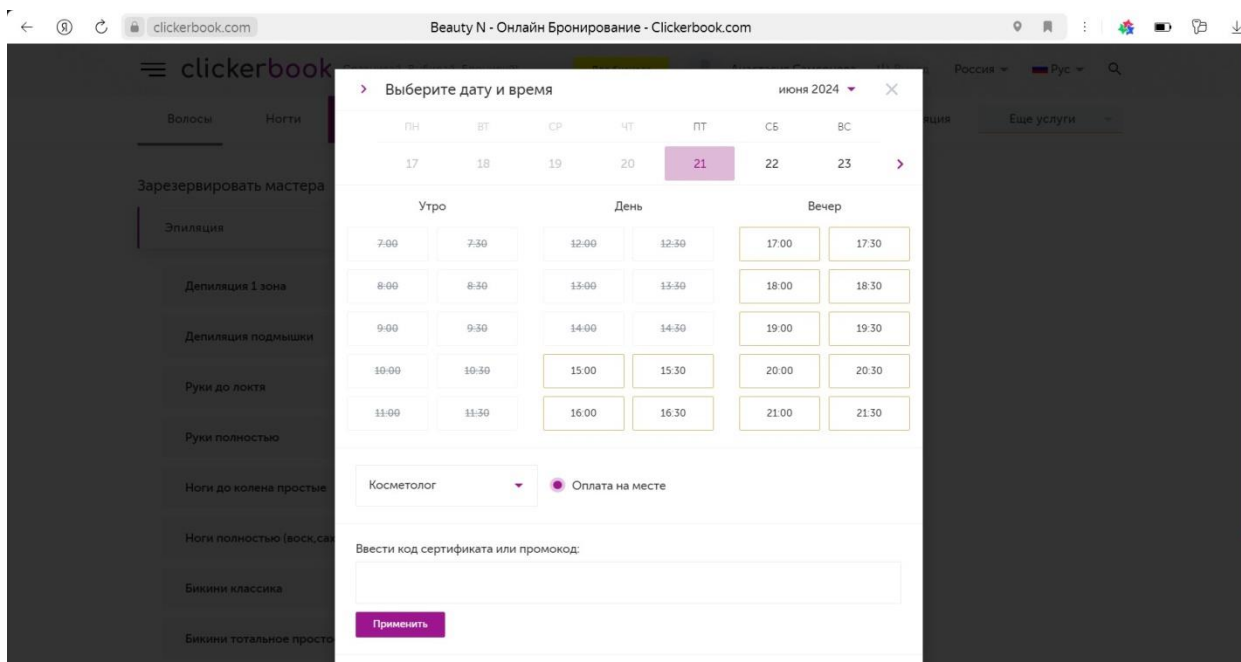


Рисунок 11 - Запись на услугу

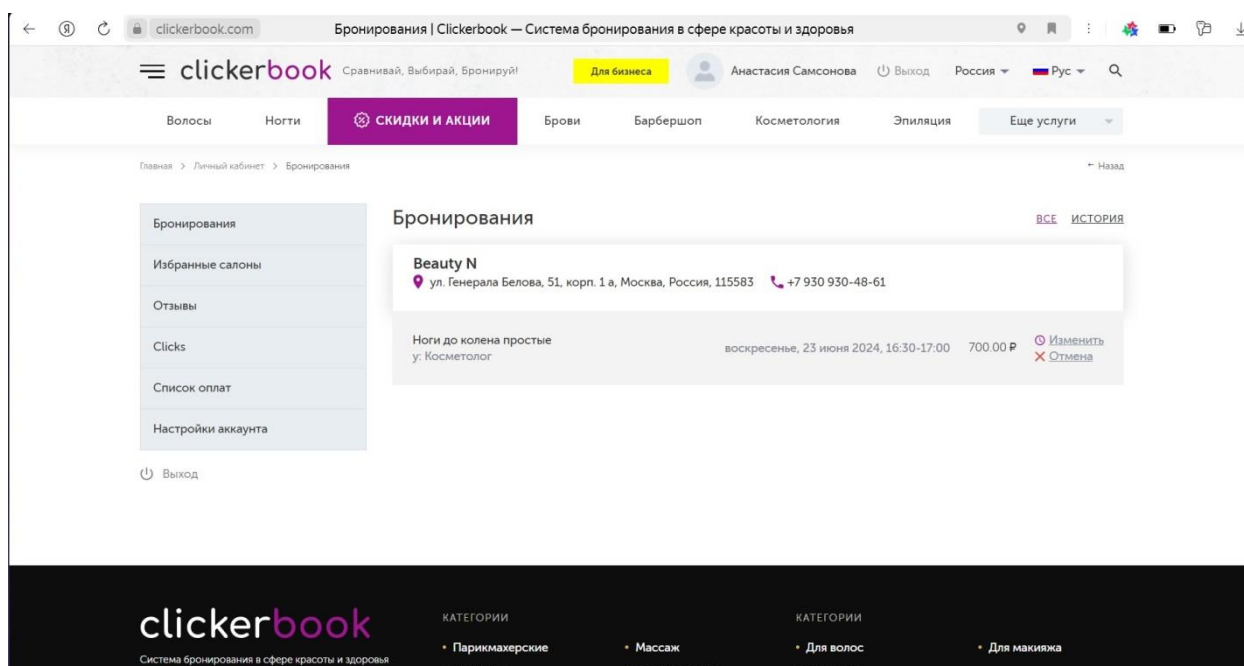


Рисунок 12 - Личный кабинет

К достоинствам данного сайта можно отнести:

- Система скидок и акций;
- Система накопления баллов для дальнейшей оплаты услуг;
- Большой выбор доступных центров;

- Возможность подать заявку на добавление в базу доступных центров, что в свою очередь работает как реклама;
- Работает в России;
- Возможность просмотра отзывов и рейтингов.

К недостаткам можно отнести:

- Работает только для Москвы и Московской области;
- Добавление центра услуг красоты и здоровья в базу сайта является платным;
- Отсутствие информации о мастерах.

2.3 Диаграммы, отражающие функциональность веб-приложения

2.3.1 Диаграмма прецедентов (Use case)

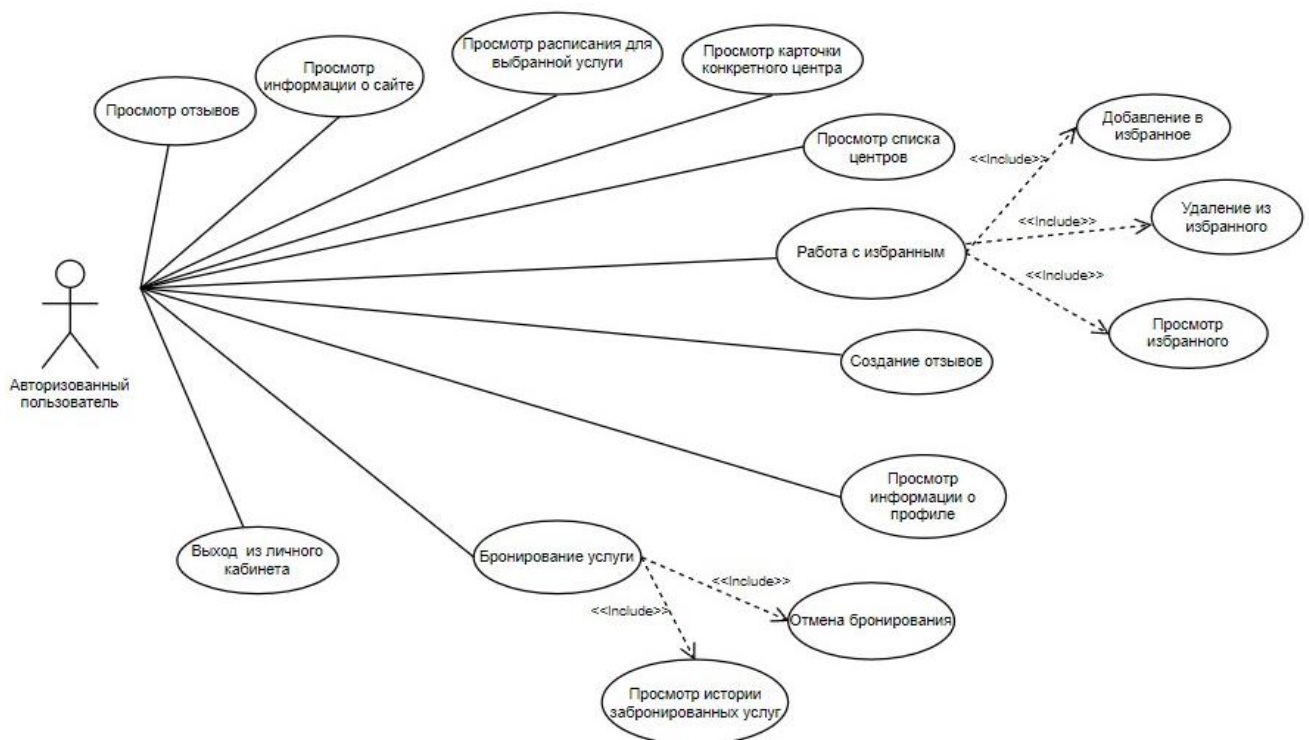


Рисунок 13 - Диаграмма прецедентов для авторизованного пользователя

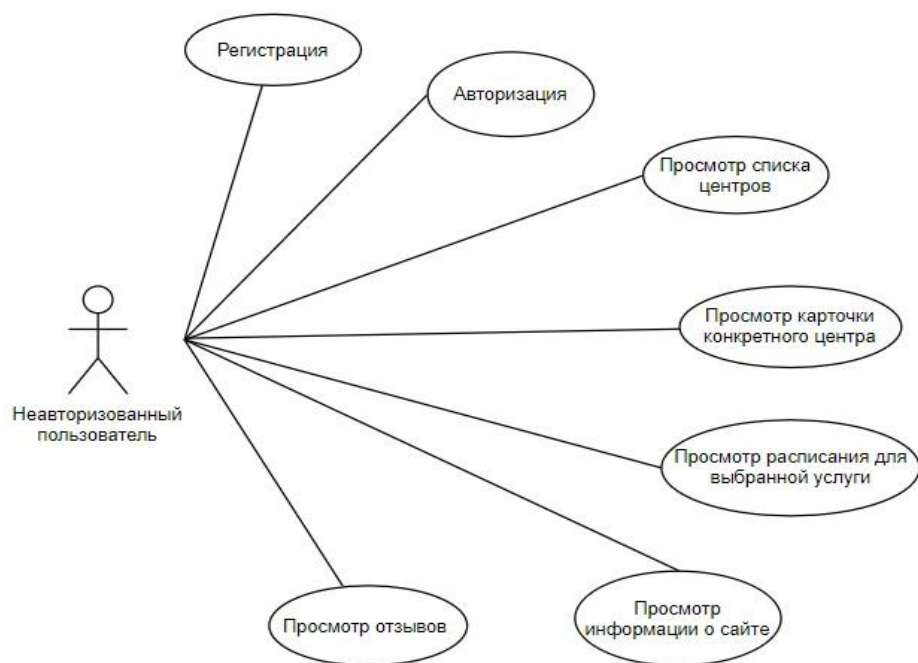


Рисунок 14 - Диаграмма прецедентов для неавторизованного пользователя

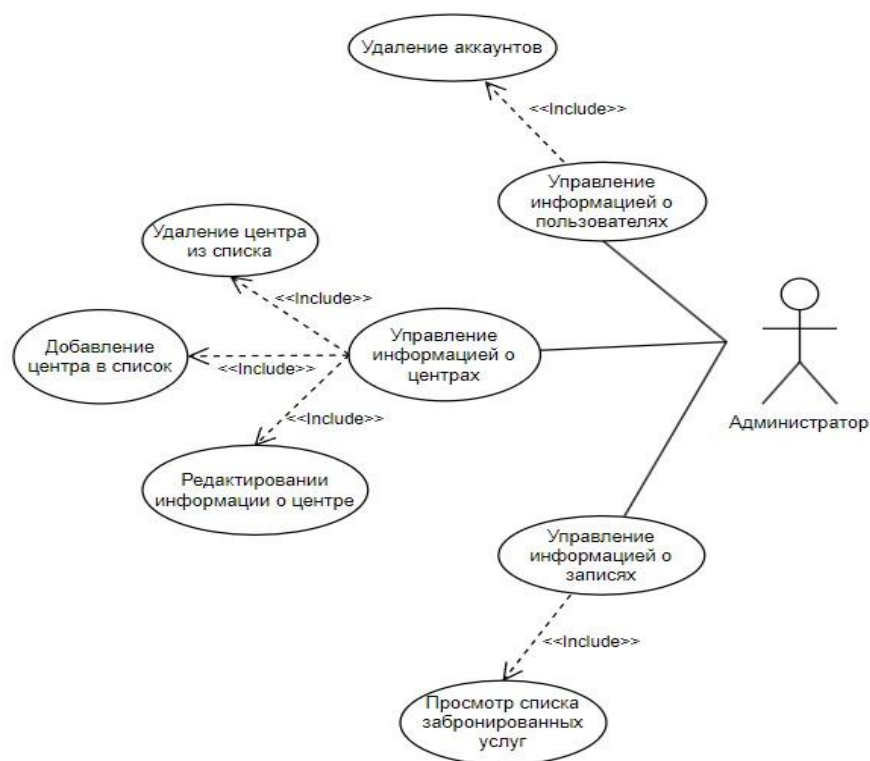


Рисунок 15 - Диаграмма прецедентов для администратора

2.3.2 Диаграмма последовательности (Sequence diagram)

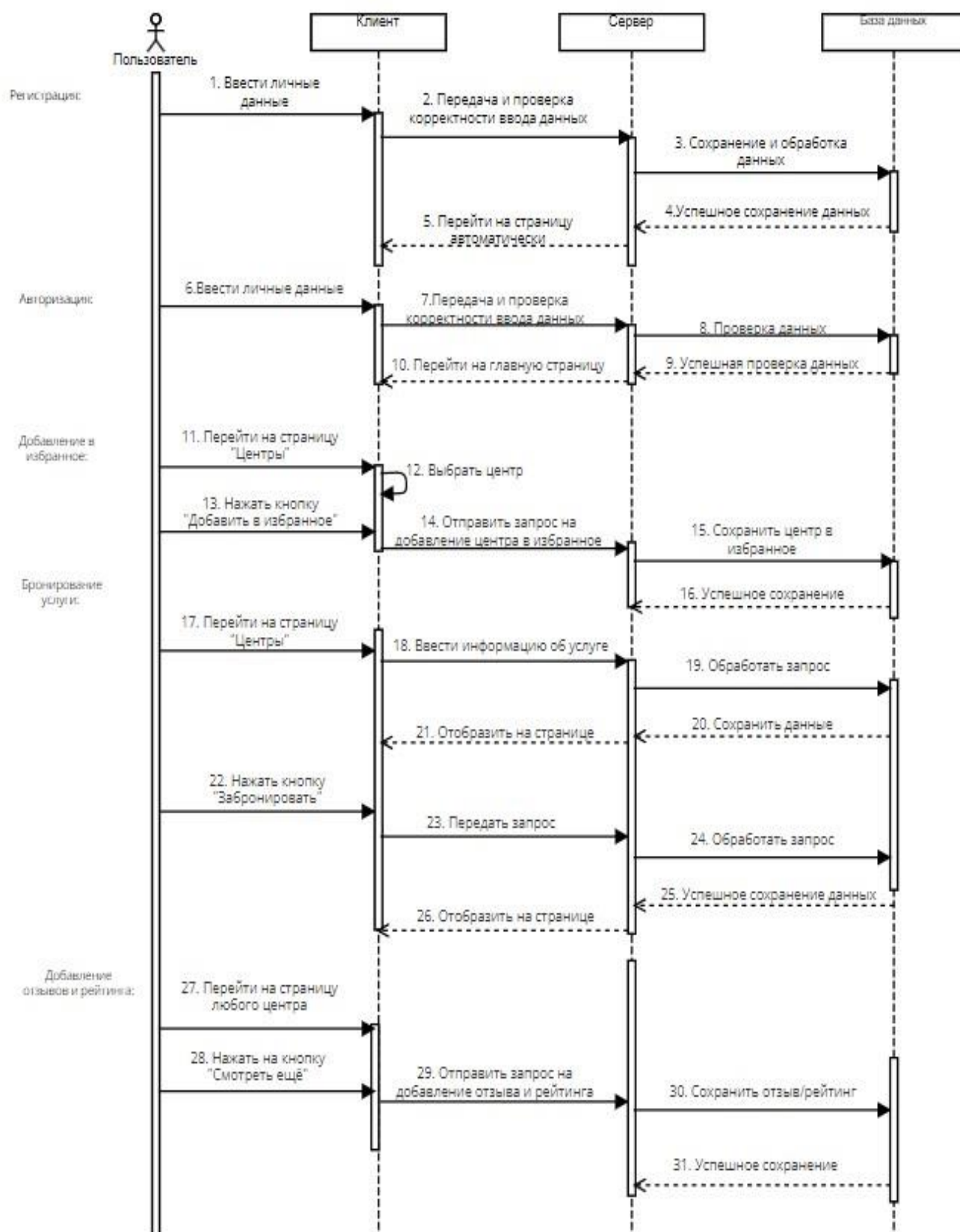


Рисунок 16 - Диаграмма последовательности

2.3.3 Диаграмма состояний (Statechart diagram)

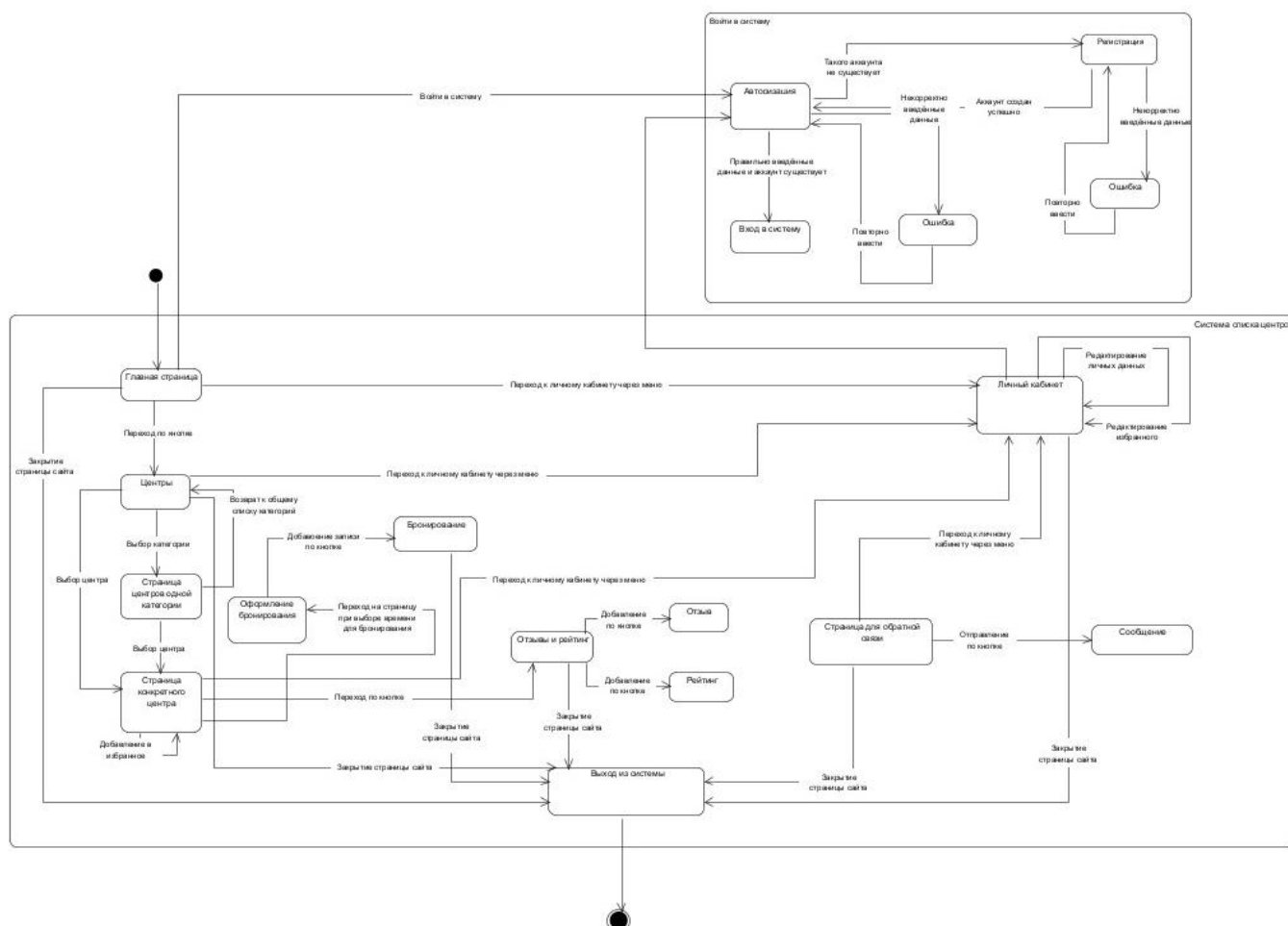


Рисунок 17 - Диаграмма состояний

2.3.4 Диаграмма активностей (Activity diagram)

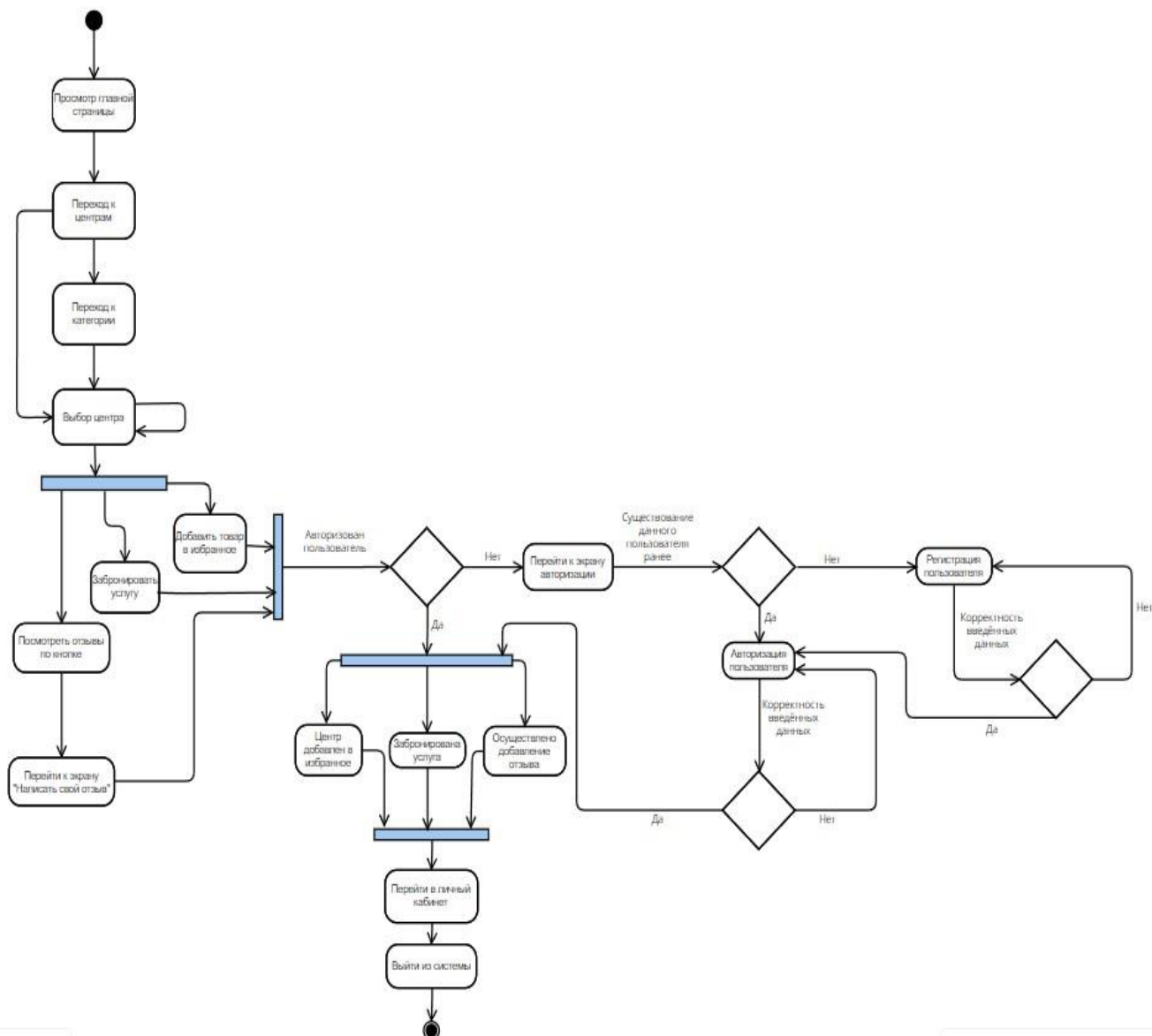


Рисунок 18 - Диаграмма активностей

2.3.5 Диаграмма объектов (Object diagram)

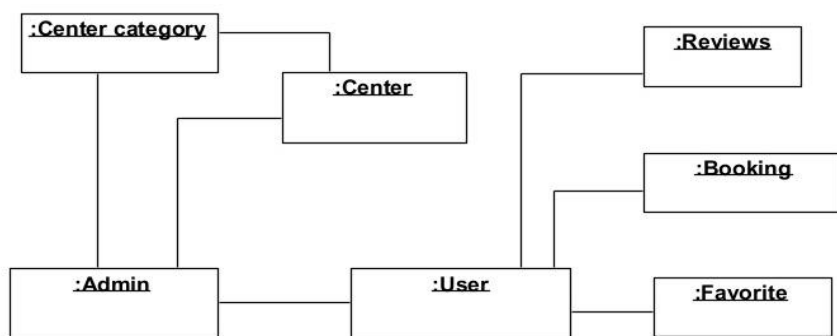


Рисунок 19 - Диаграмма объектов

2.3.6 Диаграмма классов (Class diagram)

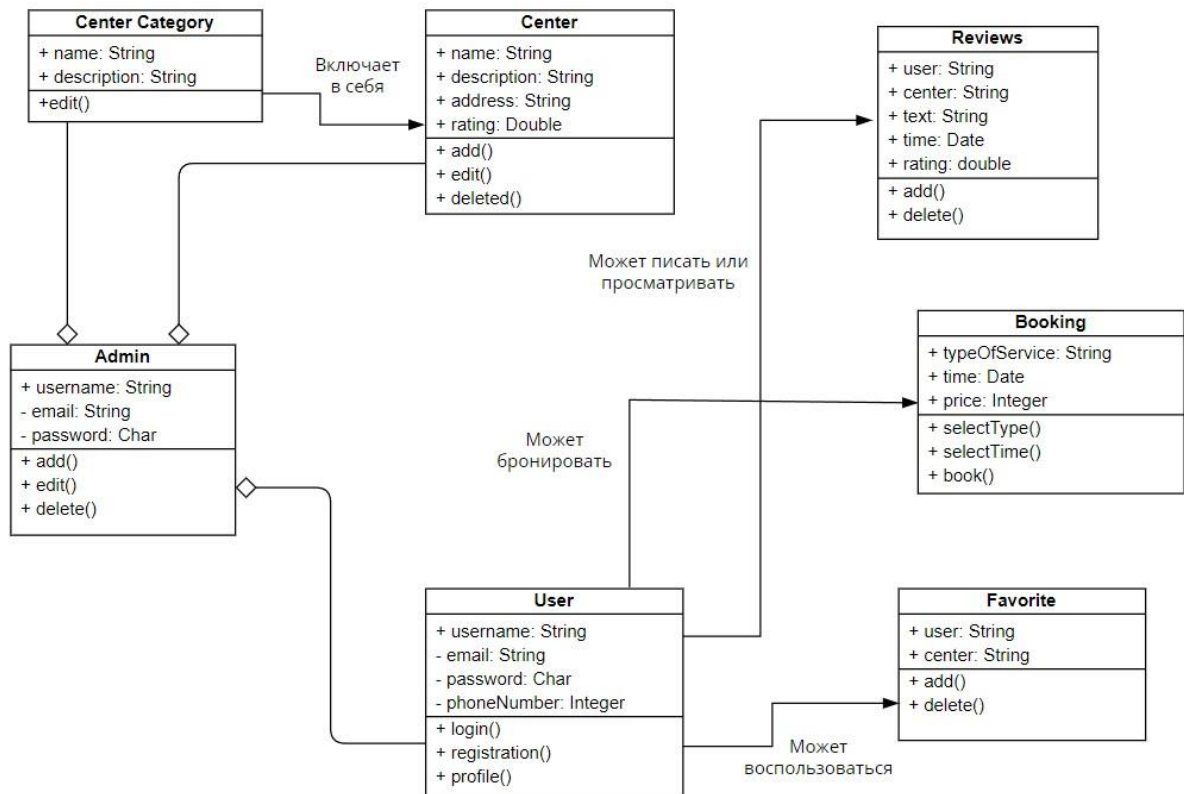


Рисунок 20 - Диаграмма классов

2.3.7 Диаграмма сотрудничества (Collaboration diagram)

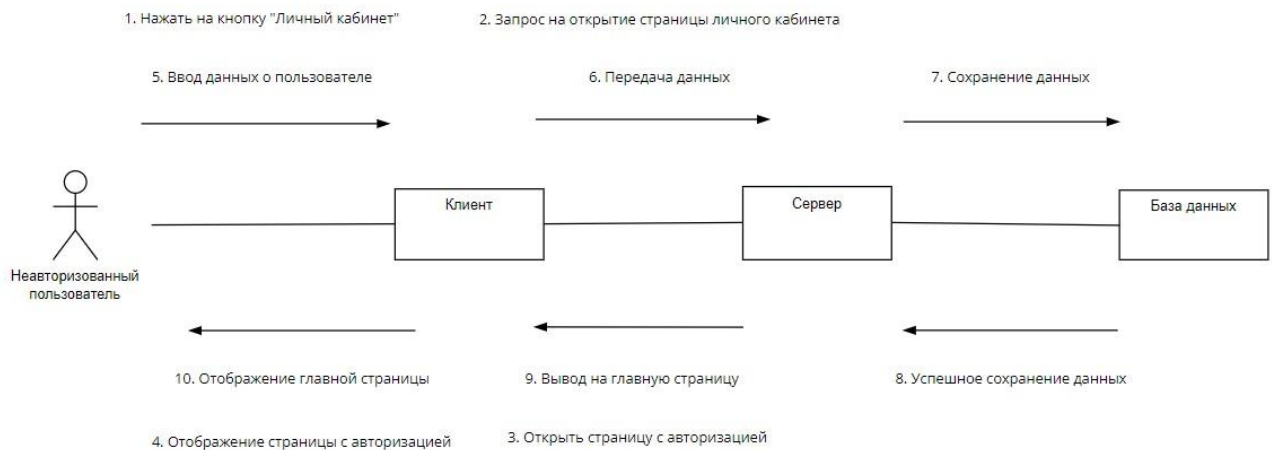


Рисунок 21 - Диаграмма сотрудничества при авторизации [3]



Рисунок 22 - Диаграмма сотрудничества при регистрации



Рисунок 23 - Диаграмма сотрудничества при добавлении в избранное



Рисунок 24 - Диаграмма сотрудничества при создании отзыва



Рисунок 25 - Диаграмма сотрудничества при бронировании

2.3.8 Диаграмма развёртывания (Deployment diagram)

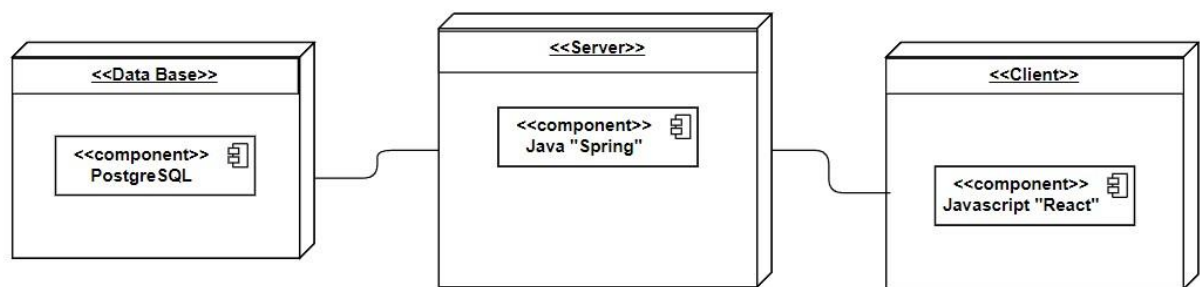


Рисунок 26 - Диаграмма развёртывания

2.3.9 ER-диаграмма

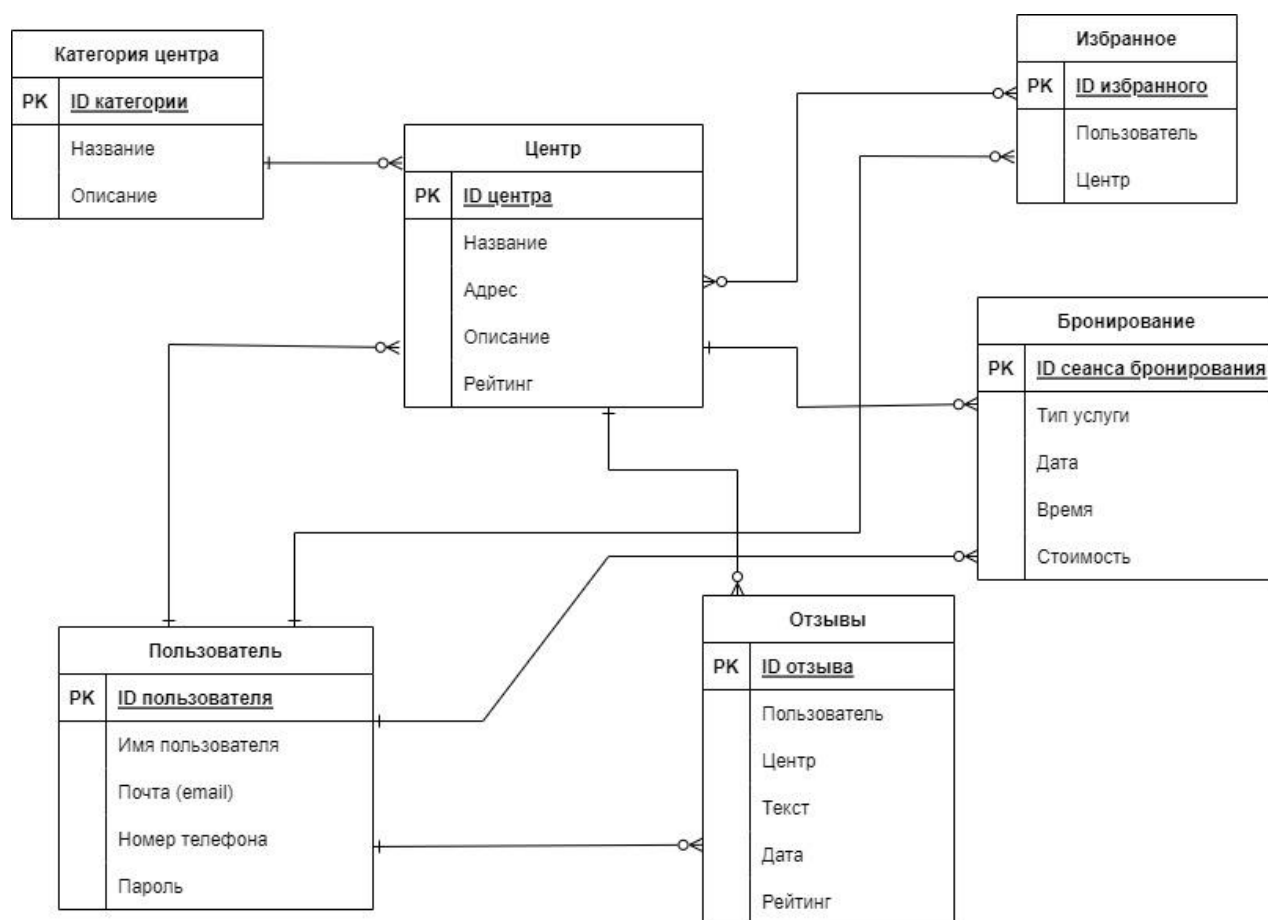


Рисунок 27 - ER-диаграмма

3 Реализация

3.1 Средства реализации

3.1.1 Средства реализации серверной части приложения

Для разработки серверной (backend) части приложения был выбран следующий стек технологий:

- Java — это многоплатформенный, объектно-ориентированный язык программирования, работающий на миллиардах устройств по всему миру. На нем работают приложения, операционные системы смартфонов, корпоративное программное обеспечение и многие известные программы. Данный язык программирования мультифункционален и независим. Код, написанный на языке Java может работать на любой платформе, поддерживающей этот язык программирования;
- Spring Boot - это популярный фреймворк для создания веб-приложений с использованием Java. Он представляет собой набор инструментов и библиотек для создания приложений корпоративного уровня. Spring Boot упрощает создание автономных приложений производственного уровня, которые можно легко развернуть и запустить с минимальной конфигурацией;
- PostgreSQL это объектно-реляционная система управления базами данных (СУБД), которая предоставляет мощные средства для хранения, организации и манипулирования данными. PostgreSQL является свободным и открытым программным обеспечением и использует SQL для работы с данными. Он предлагает расширяемость, высокую надежность, многофункциональность, поддержку геопространственных данных, JSON-данных и многое другое. PostgreSQL активно развивается и используется в крупных проектах по всему миру.

3.1.2 Средства реализации клиентской части приложения

Для разработки клиентской (frontend) части приложения был выбран следующий стек технологий:

- JavaScript — это один из наиболее популярных языков программирования, который используется для разработки веб-приложений, игр, мобильных приложений и других приложений. JavaScript является интерпретируемым языком скриптов, то есть он выполняется в среде браузера, что позволяет создавать динамические и интерактивные веб-сайты, а также управлять внешним поведением страницы. Помимо этого, он обладает следующими преимуществами: широкая поддержка, гибкость, доступ к различным библиотекам и фреймворкам;[4]
- React.js — это open-source JavaScript библиотека для построения пользовательских интерфейсов. React позволяет разрабатывать компоненты UI в виде функций или классов, которые могут быть многократно использованы, и которые просты в поддержке. React используется для создания одностраничных приложений (SPA), мобильных приложений, административных дашбордов и других веб-приложений. React реализует виртуальный DOM, который дает библиотеке многие преимущества перед другими фреймворками и библиотеками для разработки пользовательских интерфейсов, такие как скорость, универсальность, простота и удобство тестирования;[5]
- HTML (HyperText Markup Language) — это язык разметки, используемый для создания веб-страниц. С помощью HTML разработчики определяют структуру и содержание веб-страниц, позволяя браузеру правильно интерпретировать и отображать контент для пользователей. HTML использует теги для форматирования текста, вставки изображений, оформления

списков, аудио и видео, ссылок и многого другого. Он обладает следующими преимуществами: читаемость для разработчиков, возможность создания структурированного контента с использованием семантических элементов, доступность и адаптивности для различных устройств и браузеров. HTML является основным языком для создания веб-страниц и работает в сочетании с CSS для определения внешнего вида и JavaScript для добавления интерактивности;

— CSS (Cascading Style Sheets) — это язык стилей, используемый для задания внешнего вида веб-страниц. CSS позволяет разработчикам отделить визуальное представление веб-сайта от содержания, тем самым обеспечивая большую гибкость и управляемость веб-ресурсов. С помощью CSS можно задавать шрифты, цвета, расположение, размеры элементов, оформление фона и др. CSS является основным инструментом для создания визуального дизайна веб-сайтов и позволяет создавать уникальные дизайны и согласовывать внешний вид контента на всем сайте.

3.2 Реализация серверной (backend) части приложения

Серверная (backend) часть приложения была написана на языке Java с использованием фреймворка Spring Boot. Структура проекта организована таким образом, чтобы обеспечить удобное и логичное разделение компонентов по функциональным областям. Корневой каталог проекта называется `reservation_service`, а внутри него расположены следующие основные пакеты: `controller`, `dto`, `model`, `repository`, `service`, и `util.mapper`.

Кроме того, в проекте присутствуют ресурсы, такие как конфигурационные файлы, которые находятся в каталоге `resources`. Основные файлы включают `application.properties` для настройки приложения и `schema.sql` для определения схемы базы данных.

В корне проекта также находятся файлы для управления сборкой и зависимости проекта, такие как `pom.xml`, а также вспомогательные файлы, такие как `.gitignore`.

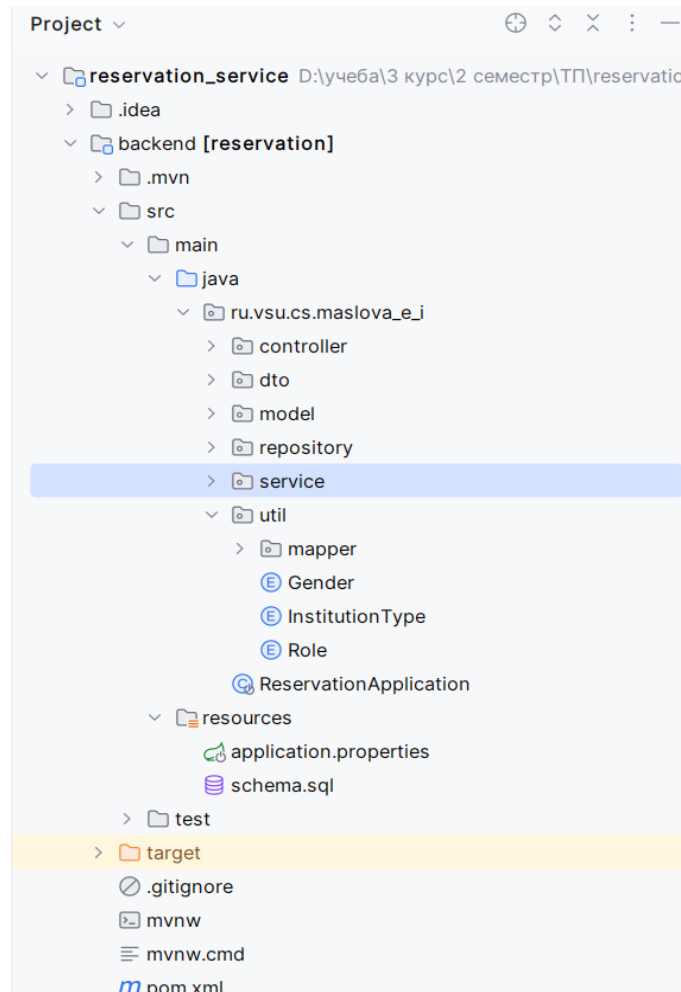


Рисунок 28 - Структура проекта

Пакет controller содержит контроллеры, которые обрабатывают HTTP-запросы от клиентов. Каждый контроллер содержит методы для обработки конкретных запросов, например, создание, получение списка или поиск услуг.

Пакет `dto` (Data Transfer Object) содержит классы, используемые для передачи данных между различными слоями приложения и для взаимодействия с внешними системами.

Пакет `model` включает модели данных, которые представляют сущности приложения и отображаются на таблицы базы данных.

Пакет `repository` содержит интерфейсы репозитория, которые обеспечивают взаимодействие с базой данных и реализуют методы для выполнения CRUD-операций.

Пакет `service` содержит бизнес-логику приложения, разделенную по функциональным областям. Каждый сервис-класс отвечает за операции, связанные с конкретной сущностью.

Пакет `util.mapper` включает мапперы, которые преобразуют данные между моделями и DTO.

Для описания спецификации API использовался Swagger. Для того чтобы интегрировать его в проект, были внесены изменения в код, включая добавление необходимой зависимости в файл `pom.xml`. Это позволяет автоматически генерировать и отображать документацию API на основе аннотаций, использованных в контроллерах.

3.3 Реализация клиентской (frontend) части приложения

Одной из особенностей разработки приложения на React является декомпозиция приложения на независимые компоненты. Каждый компонент отвечает за определенный функциональный блок приложения и представляет собой функцию, которая возвращает HTML-код. Кроме того, компоненты могут содержать переменные и другие функции, что позволяет дополнительно упростить код и повысить его читабельность. Приложение загружается на страницу по мере необходимости, что уменьшает время загрузки страницы и повышает производительность приложения в целом.

Основной HTML файл называется `index.html`, именно на него загружаются компоненты, которые будут выведены в браузере. В этом файле вызывается `index.js`, в котором находится точка входа React-приложения, и вызывается основной компонент `App.js`, который будет меняться в зависимости от действий пользователя. Все остальные компоненты вызываются по мере необходимости.

Папка `components` предназначена для хранения всех React-компонентов, которые используются для создания пользовательского интерфейса. React-компоненты могут быть представлены как отдельные файлы, каждый из которых отвечает за отдельный элемент интерфейса. Каждая страница приложения содержит файлы стилей и другие ресурсы, относящиеся к этой странице. Это обеспечивает логическую организацию файлов и делает их легкодоступными для разработчиков.

`Package.json`: Файл конфигурации проекта, содержащий информацию о проекте и его зависимостях.

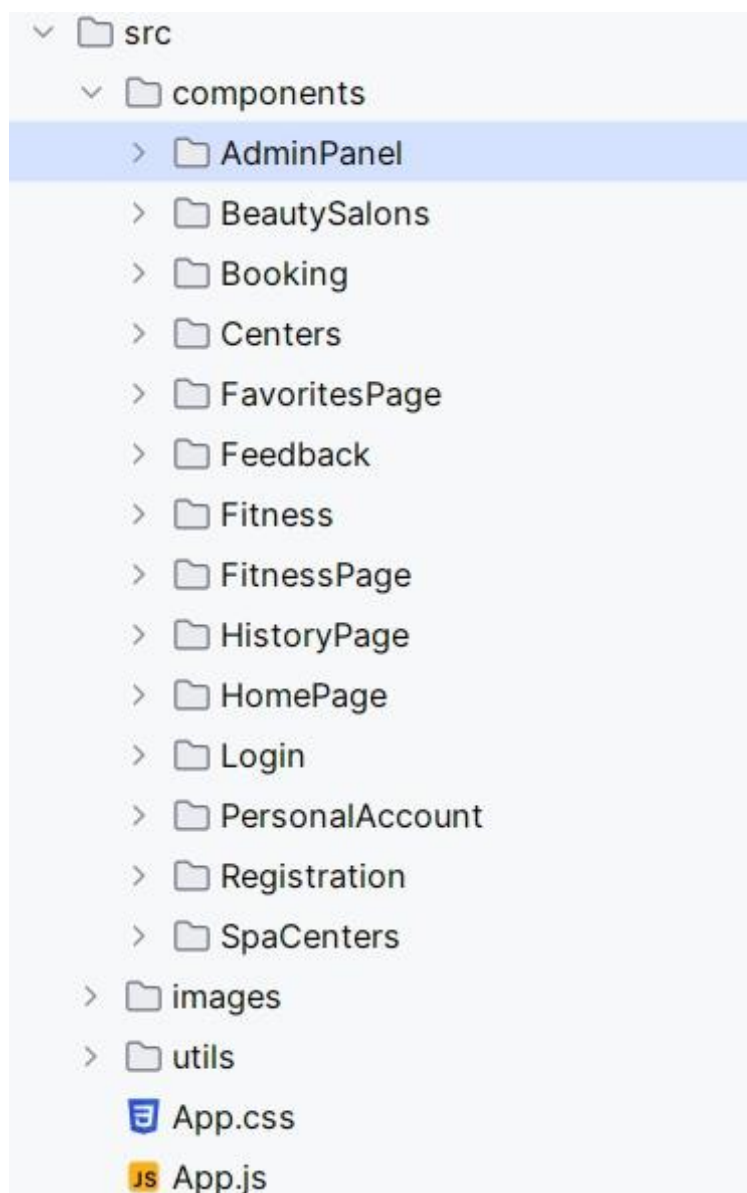


Рисунок 29 - Компоненты приложения

3.4 Навигация по приложению

3.4.1 Для неавторизованного пользователя

Первое, что видит пользователь — главный экран (Рисунок 30).

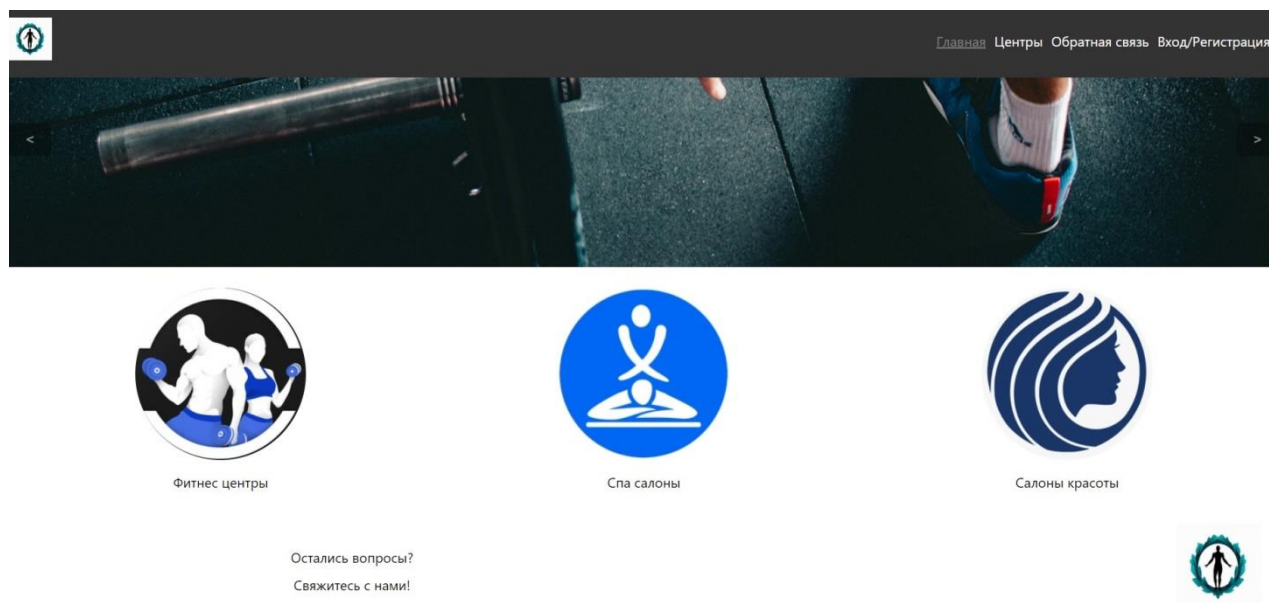
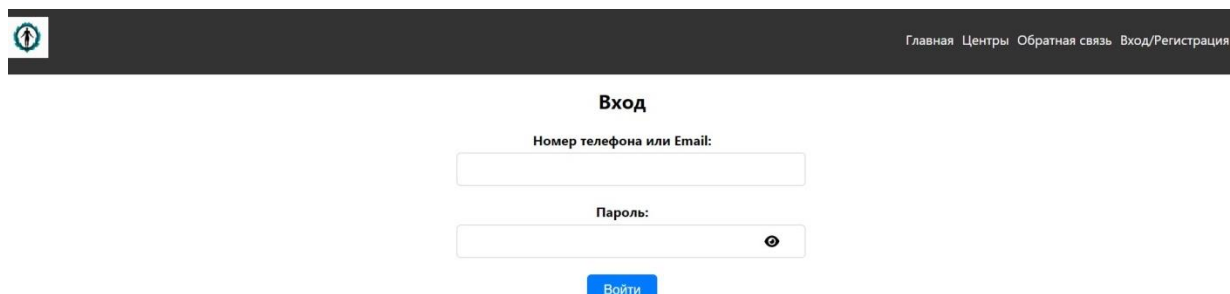


Рисунок 30 - Главный экран

Далее у пользователя возникает выбор: он может войти в личный кабинет при помощи авторизации (Рисунок 31), если нет аккаунта – при помощи регистрации (Рисунок 32) или продолжить пользоваться сайтом в режиме неавторизованного пользователя, но с ограниченным функционалом (без возможности бронирования услуг и добавления отзывов и заведений в избранное).



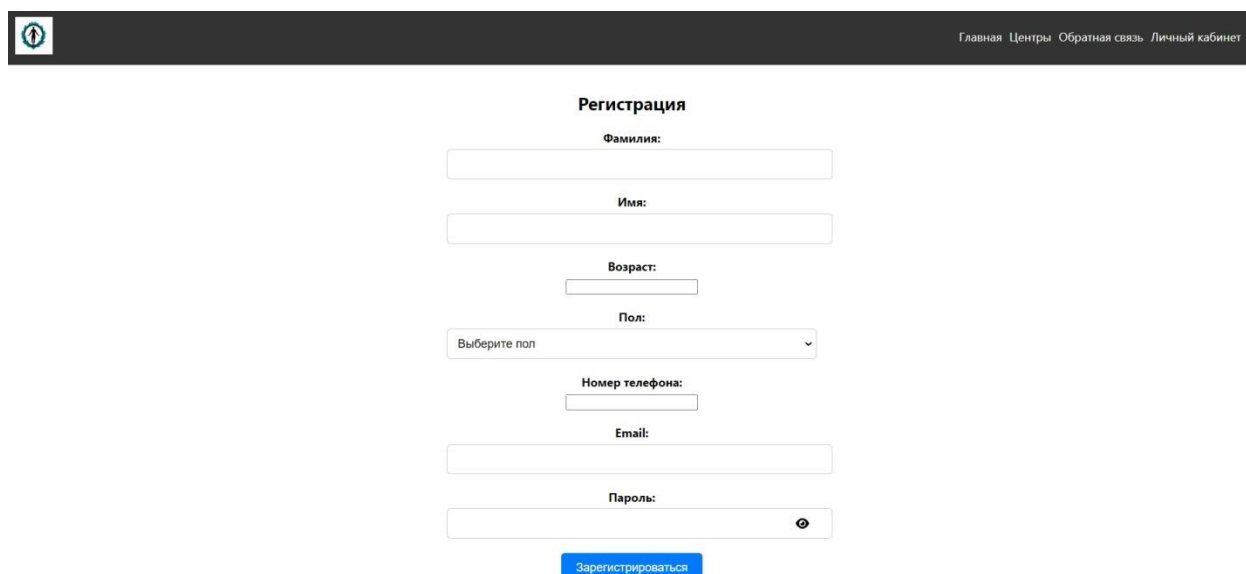
Вход

Номер телефона или Email:

Пароль:

Войти

Рисунок 31 - Экран авторизации



Регистрация

Фамилия:

Имя:

Возраст:

Пол:

Выберите пол

Номер телефона:

Email:

Пароль:

Зарегистрироваться

Рисунок 32 - Экран регистрации

Неавторизованному пользователю доступен просмотр центров красоты и здоровья по категориям (Рисунок 33) и списка центров по каждой категории отдельно (Рисунок 34).

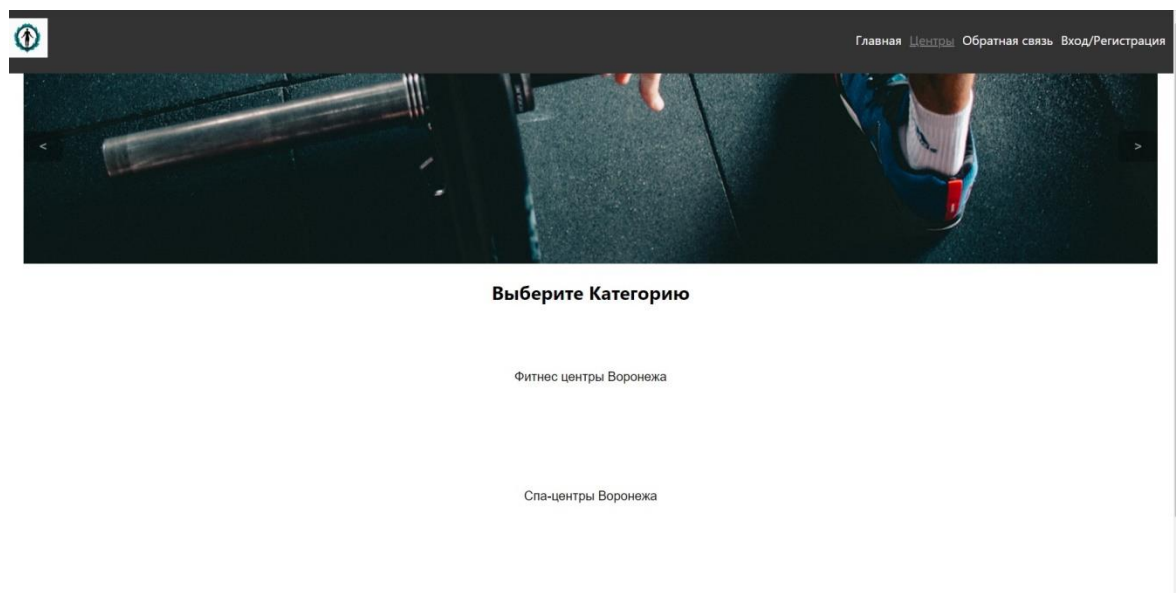


Рисунок 33 - Список категорий центров красоты и здоровья

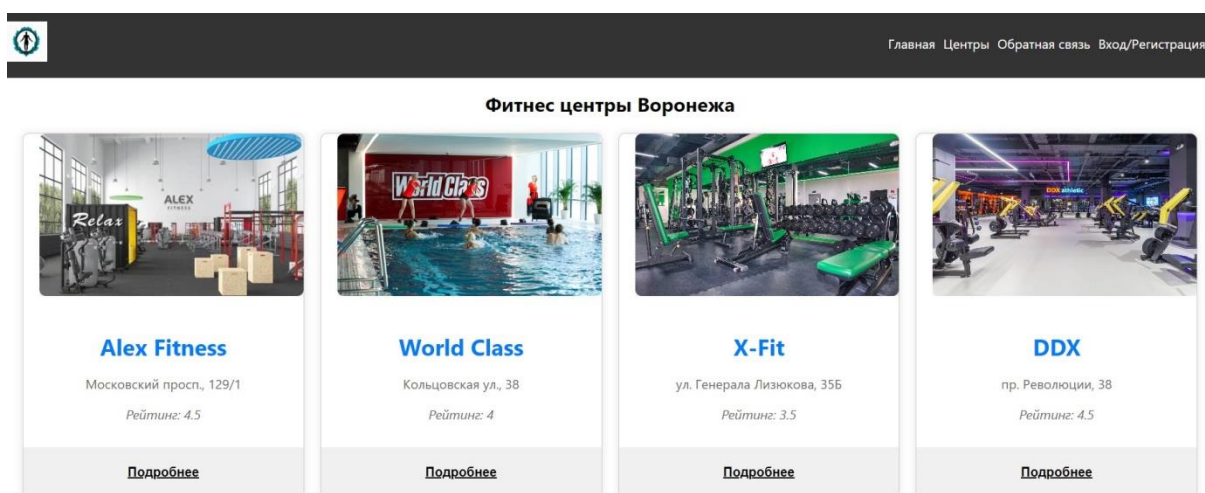


Рисунок 34 - Список центров в категории «Фитнес-центры»

Также неавторизованный пользователь может рассмотреть отдельные карточки центров с рейтингом и отзывами (Рисунок 35).

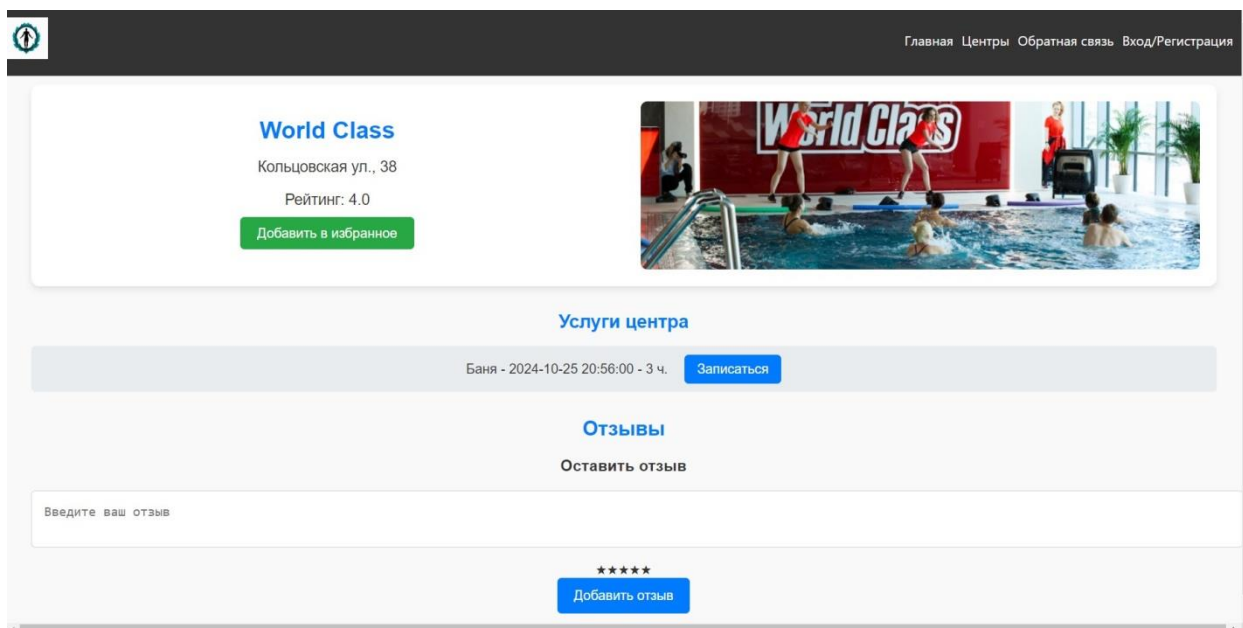


Рисунок 35 - Карточка фитнес-центра

Для неавторизованного пользователя также доступна страница «Обратная связь» (Рисунок 36).

Рисунок 37 - Экран «Обратная связь»

3.4.2 Для авторизованного пользователя

После авторизации пользователю становится доступной возможность добавлять в избранное конкретный товар. Для этого на странице каталога товаров необходимо нажать на кнопку «Добавить в избранное» (Рисунок 37).

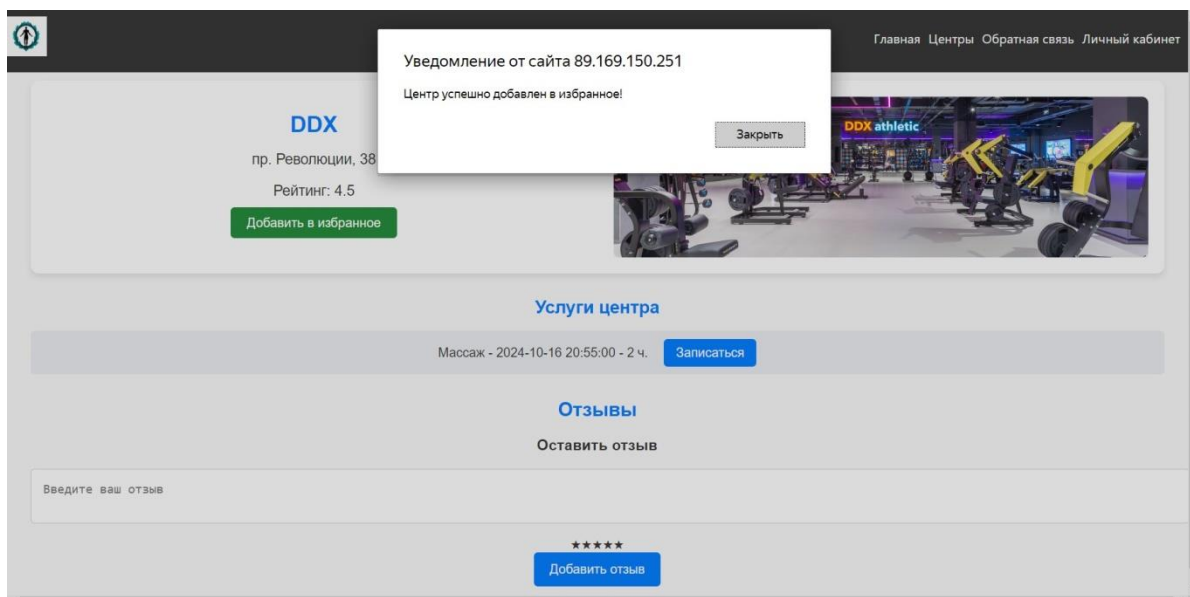


Рисунок 37 - Добавление центра в избранное

Список избранных центров, пользователь может посмотреть в своем личном кабинете (Рисунок 38).

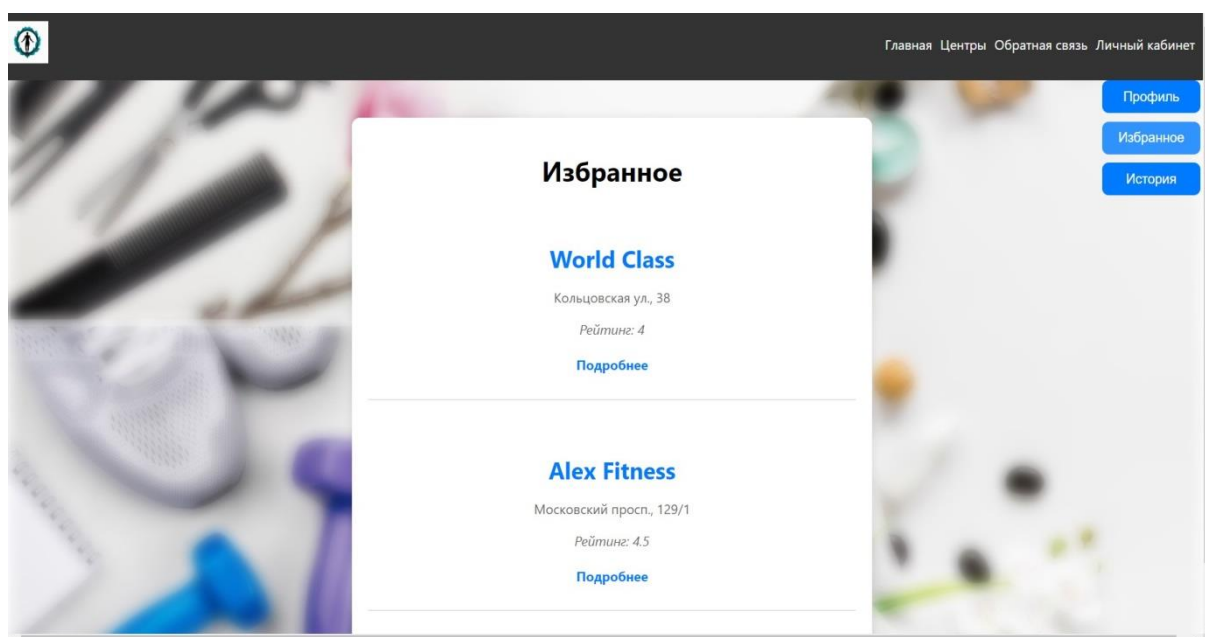


Рисунок 38 - Избранное

Также авторизованный пользователь может посмотреть информацию о себе, введенную при регистрации (Рисунок 39).

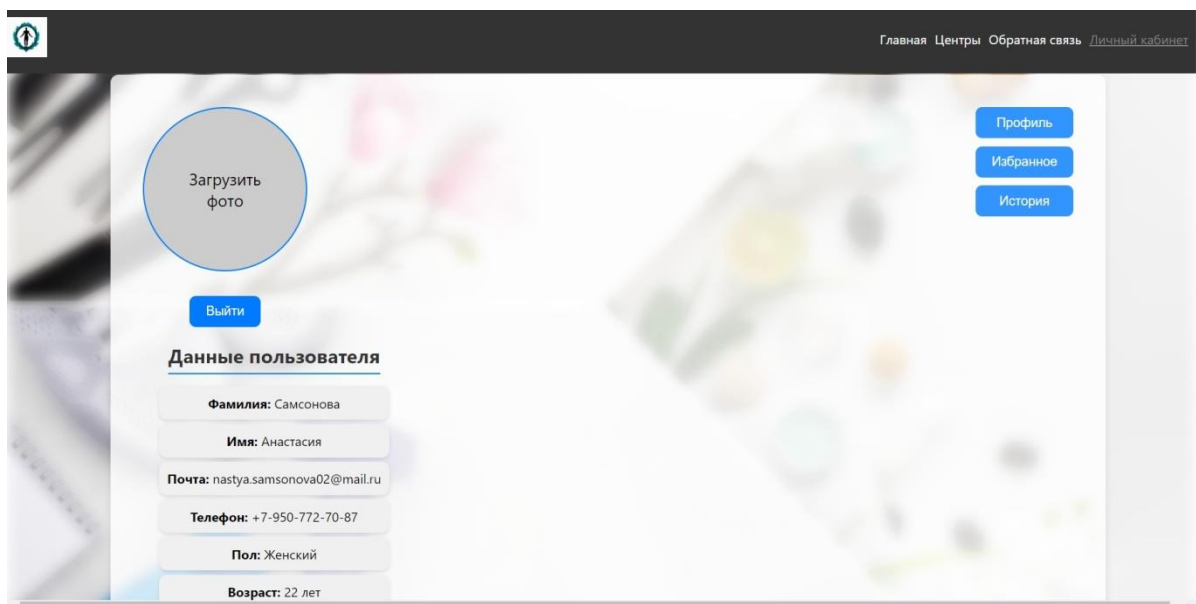


Рисунок 39 - Личный кабинет

Авторизованному пользователю становится доступно онлайн-бронирование (Рисунок 40).

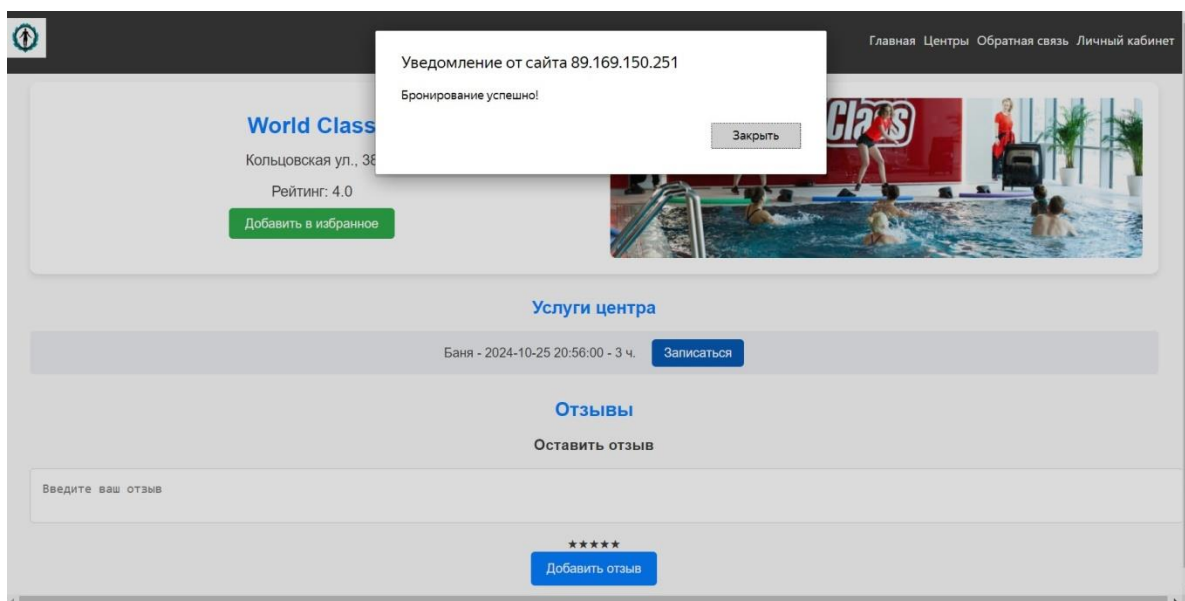


Рисунок 40 - Запись на выбранную услугу

Также авторизованный пользователь может просмотреть историю записей в личном кабинете (Рисунок 41).

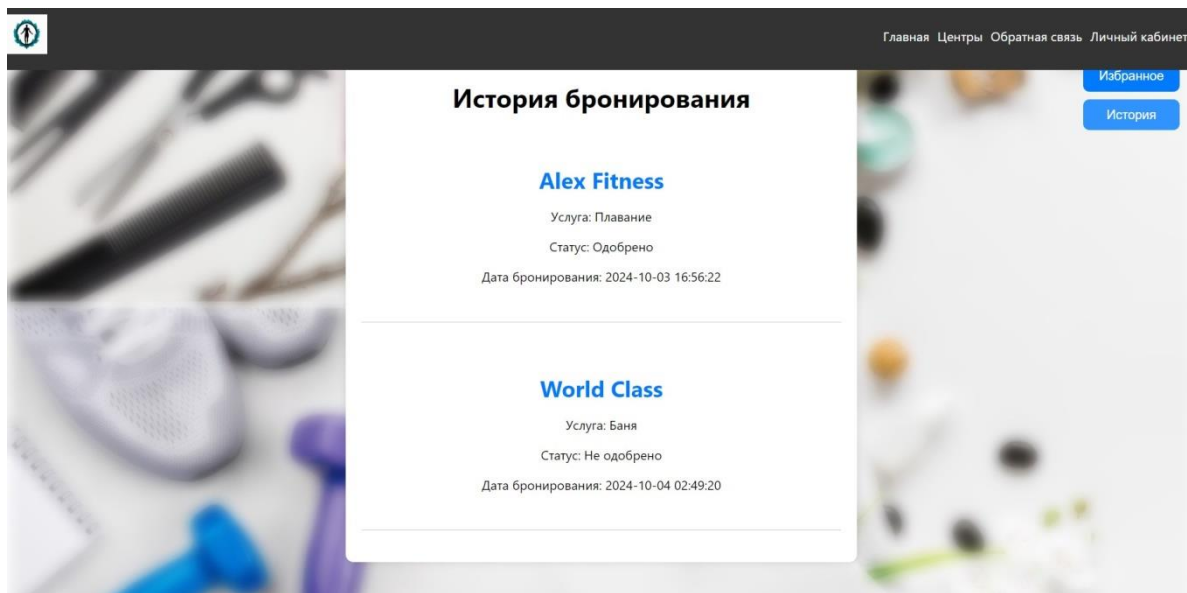


Рисунок 41 - История бронирования

И теперь пользователь может оставлять отзывы (Рисунок 42).

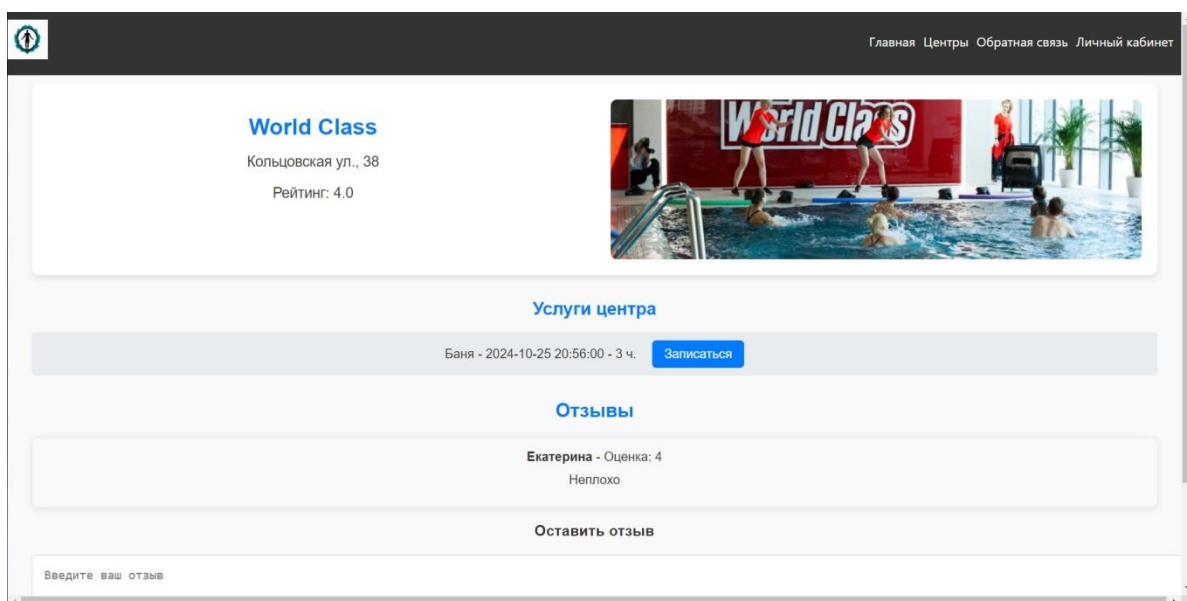


Рисунок 42 - Отзыв

3.4.3 Для администратора

Для администратора доступно редактирование заявок на бронирование (Рисунок 43), списка центров (Рисунок 44, 45), карточек центров (Рисунок 46) и списка пользователей (Рисунок 47).

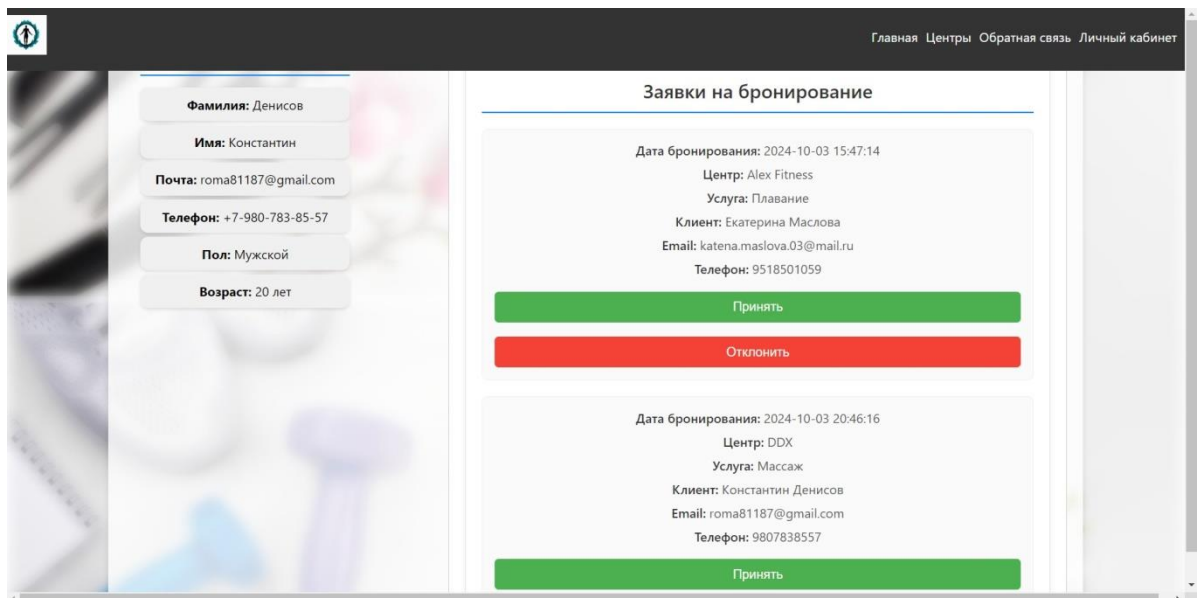


Рисунок 43 - Список заявок на бронирование

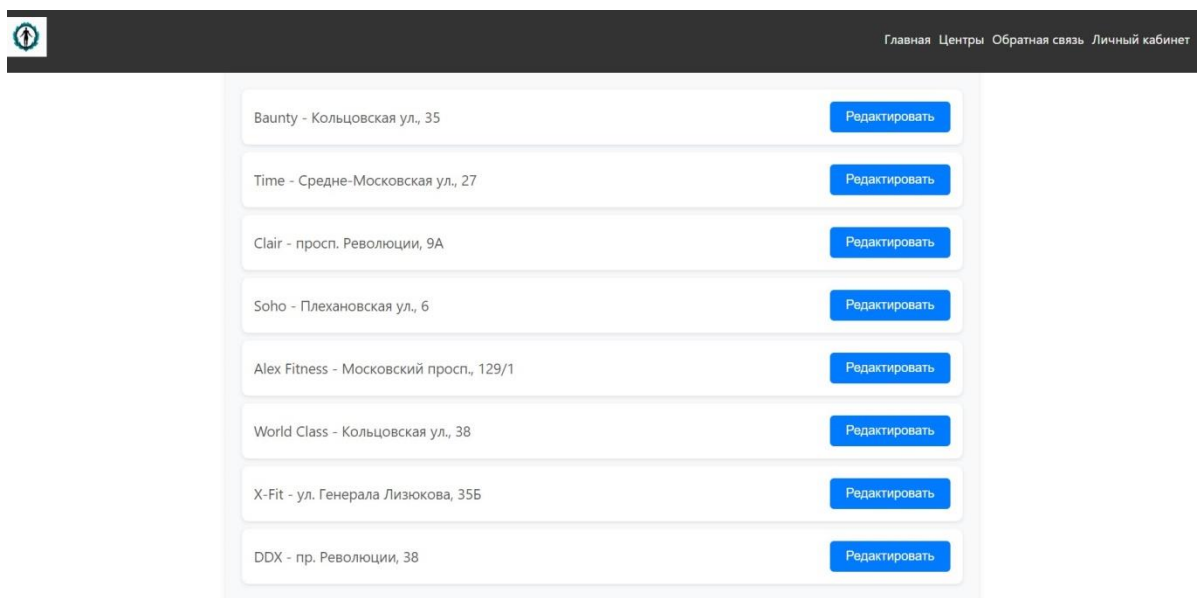
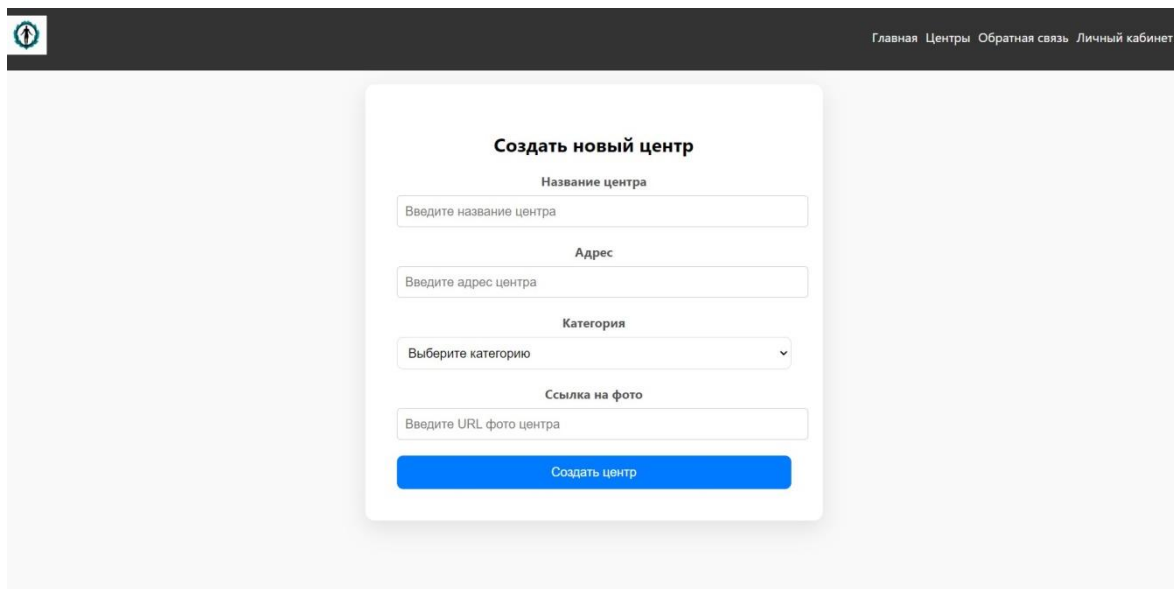
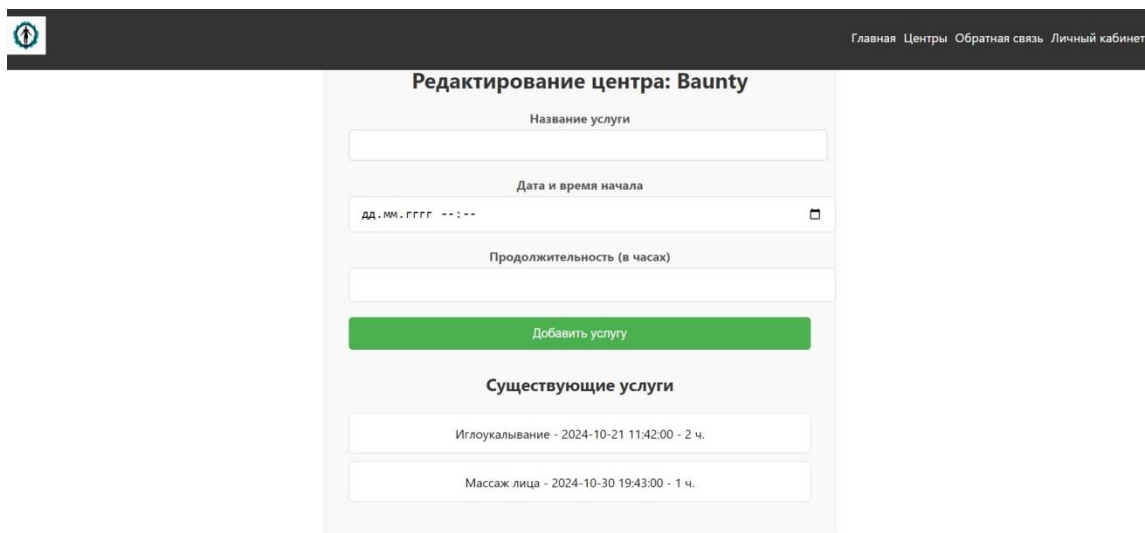


Рисунок 44 - Список центров для администратора



The screenshot shows a web interface with a dark header bar. On the left is a logo, and on the right are navigation links: Главная, Центры, Обратная связь, and Личный кабинет. The main content area features a white card titled 'Создать новый центр'. The card contains four input fields: 'Название центра' (with placeholder 'Введите название центра'), 'Адрес' (with placeholder 'Введите адрес центра'), 'Категория' (a dropdown menu with 'Выберите категорию'), and 'Ссылка на фото' (with placeholder 'Введите URL фото центра'). At the bottom of the card is a blue button labeled 'Создать центр'.

Рисунок 45 - Создание нового центра на сайте



The screenshot shows a web interface with a dark header bar. On the left is a logo, and on the right are navigation links: Главная, Центры, Обратная связь, and Личный кабинет. The main content area features a white card titled 'Редактирование центра: Baunty'. The card contains three input fields: 'Название услуги', 'Дата и время начала' (with a date-time picker showing 'ДД.ММ.ГГГГ --:--'), and 'Продолжительность (в часах)'. Below these fields is a green button labeled 'Добавить услугу'. Underneath the button is a section titled 'Существующие услуги' which contains two rows of service information: 'Иглоукальвание - 2024-10-21 11:42:00 - 2 ч.' and 'Массаж лица - 2024-10-30 19:43:00 - 1 ч.'.

Рисунок 46 - Редактирование карточки центра

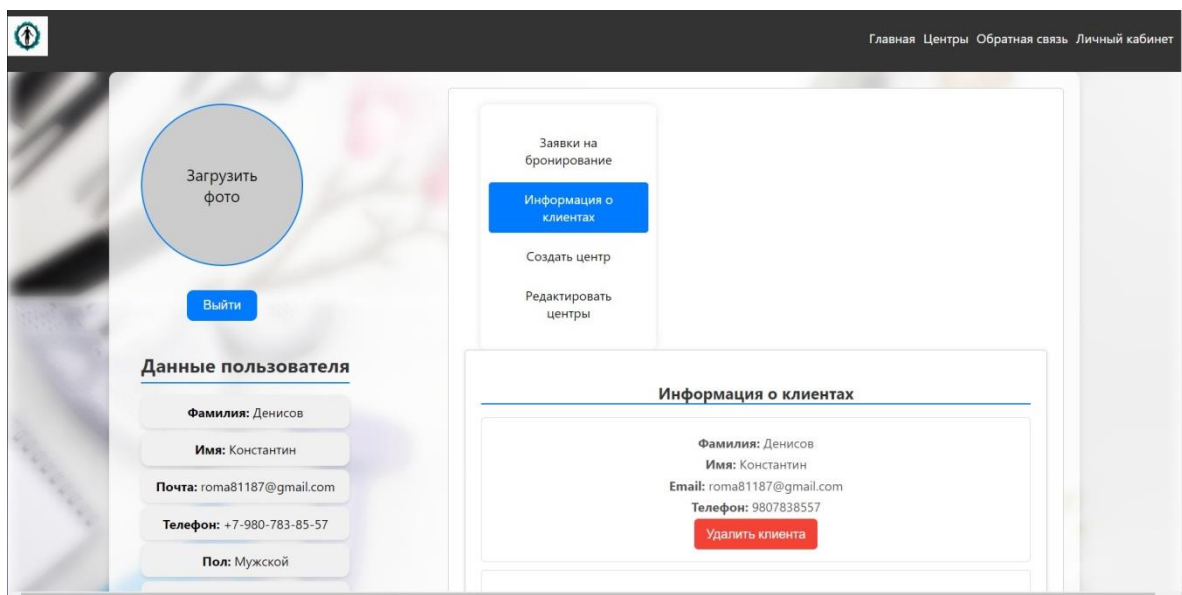


Рисунок 47 - Список пользователей

4 Тестирование

После реализации всех задач, был произведен запланированный набор тестов:

- Дымовое тестирование;
- Тестирование пользовательского интерфейса.

4.1 Дымовое тестирование

Дымовое тестирование — это тип тестирования, при котором производится кратковременная проверка основной функциональности системы, чтобы убедиться в отсутствии серьезных проблем, ошибок или сбоев.

Такой тип тестирования полезен, чтобы обнаружить крупные проблемы в функциональности системы до того, как будут проведены более подробные и насыщенные тесты. Если в процессе дымового тестирования найдены ошибки или проблемы, то это может стать причиной для проведения дополнительных и более детальных тестов, чтобы устранить недочеты и проблемы в работе программы.

В ходе дымового тестирования выполняются базовые операции или сценарии, которые предполагаются как наиболее важные и часто используемые пользователем. Далее представлены результаты дымового тестирования для основных сценариев неавторизованного, авторизованного пользователей и администратора (Таблица 1-3).

Таблица 1 - Результаты дымового тестирования для неавторизованного пользователя

Тестовый сценарий	Результат теста
Регистрация	Пройден
Авторизация	Пройден
Просмотр списка центров	Пройден
Просмотр карточки центра	Пройден

Осуществление обратной связи	Пройден
Просмотр отзывов	Пройден

Таблица 2 -Результаты дымового тестирования для авторизованного пользователя

Тестовый сценарий	Результат теста
Регистрация	Пройден
Авторизация	Пройден
Просмотр списка центров	Пройден
Просмотр карточки центра	Пройден
Осуществление обратной связи	Пройден
Добавление в избранное	Пройден
Получение информации о пользователе в личном кабинете	Пройден
Просмотр отзывов	Пройден
Добавление отзыва	Пройден
Осуществление записи на услугу	Пройден
Просмотр истории записей	Пройден

Таблица 3 - Результаты дымового тестирования для администратора

Тестовый сценарий	Результат теста
Авторизация	Пройден
Просмотр категорий центров	Пройден
Просмотр списка центров в категории	Пройден
Добавление центра	Пройден
Изменение данных о центре	Пройден
Просмотр отзывов	Пройден
Просмотр списка пользователей и их данных	Пройден

Удаление пользователя	Пройден
Просмотр заявок на бронирование	Пройден
Удаление заявок на бронирование	Пройден

4.2 Тестирование пользовательского интерфейса

Тестирование пользовательского интерфейса (GUI-тестирование) — это процесс тестирования элементов управления в приложении, который помогает убедиться, что интерфейс соответствует ожидаемой функциональности, включая проверку различных функций и элементов, таких как окна, диалоговые окна, кнопки, переключатели, выпадающие списки, формы, меню и т.д.

Задача проведения GUI-тестов — убедиться, что в функциях пользовательского интерфейса отсутствуют дефекты. В Таблице 4 представлена часть результатов тестирования пользовательского интерфейса. В ней отражены тестовые сценарии для неавторизованного пользователя.

Таблица 4 - Результаты GUI-тестирования

Тестовый сценарий	Ожидаемый результат	Статус теста
Нажатие на кнопку «Центры»	Переход на страницу с категориями центров	Пройден
Нажатие на категорию «Фитнес-центры»	Переход на страницу со списком центров, относящихся к данной категории	Пройден
Нажатие на категорию «Салоны красоты»	Переход на страницу со списком центров, относящихся к данной категории	Пройден
Нажатие на категорию «Спа-салоны»	Переход на страницу со списком центров, относящихся к данной категории	Пройден

Нажатие на кнопку «Главная»	Переход на главную страницу	Пройден
Нажатие на кнопку «Вход/регистрация»	Переход на страницу регистрации	Пройден
Нажатие на кнопку «Личный кабинет»	Переход на страницу личного кабинета	Пройден
Нажатие на кнопку «Добавить в избранное»	Добавление центра в избранное	Пройден
Нажатие на кнопку «Оставить отзыв»	Добавление отзыва на сайте	Пройден

Заключение

В ходе выполнения курсовой работы были выполнены все поставленные задачи. Было разработано веб-приложение для просмотра доступных в Воронеже центров красоты и здоровья (фитнес – центры, спа, салоны красоты) с возможностью добавления их в список избранного, добавления отзывов авторизованными пользователями и бронирования их услуг в режиме онлайн.

В начале разработки был проведен анализ предметной области, определены основные требования к разрабатываемой системе, определены основные сценарии веб-приложения. По результатам разработки проводился ряд тестов с целью проверки работоспособности системы.

Благодаря проекту были решены следующие задачи у пользователя:

- Осуществление просмотра центров по категориям (салоны красоты, спа, фитнес-центры);
- Получение информации о предоставляемых услугах;
- Просмотр рейтинга и отзывов об интересующем пользователя центре;
- Возможность просмотра и добавления отзывов;
- Добавление выбранных центров в список «Избранное»;
- Запись в выбранный центр на желаемую услугу;

И следующие задачи для администратора:

- Добавление и удаление центров в список сайта;
- Редактирование информации о конкретном центре и его услугах;
- Редактирование списка пользователей и их записей.

Таким образом, итоги разработки, проверенные в ходе тестирования, позволяют достигнуть поставленных заказчиком целей и решают сформулированные в начале разработки задачи.

Список использованной литературы

1. Fresha – мгновенно бронируйте услуги в салонах красоты и спа-центрах рядом с вами [Электронный ресурс]. – Режим доступа: <https://www.fresha.com/ru>. – Заглавие с экрана. – (Дата обращения: 10.04).
2. Clickerbook – Сравнивай, Выбирай, Бронируй [Электронный ресурс]. – Режим доступа: <https://clickerbook.com/ru>. – Заглавие с экрана. – (Дата обращения: 10.04.2024).
3. Диаграммы сотрудничества [Электронный ресурс]. — Режим доступа: <https://helpiks.org/9-53448.html>. — Заглавие с экрана. — (Дата обращения: 19.03.2024).
4. Что такое JavaScript? – Изучение веб-разработки | MDN [Электронный ресурс]. – Режим доступа: https://developer.mozilla.org/ru/docs/Learn/JavaScript/First_steps/What_is_JavaScript. – Заглавие с экрана. – (Дата обращения: 11.04.2024).
5. React – JavaScript-библиотека для создания пользовательских интерфейсов [Электронный ресурс]. — Режим доступа: <https://ru.react.js.org/?ref=dtf.ru>. — Заглавие с экрана. — (Дата обращения: 11.04.2024).