

Содержание

Содержание	1
Введение	3
1 Постановка задачи	4
1.1 Требования к разрабатываемой системе	4
1.1.1 Функциональные требования	4
1.1.2 Нефункциональные требования	4
1.2 Задачи, решаемые в процессе разработки	5
2 Анализ предметной области	6
2.1 Глоссарий	6
2.2 Обзор аналогов	6
2.2.1 Aviasales	7
2.2.2 EasyJet	7
2.2.3 OneTwoTrip	8
3 Диаграммы	10
3.1 Диаграмма прецедентов (Use-case diagram)	10
3.2 Диаграмма классов (Class diagram)	11
3.3 Диаграмма активности (Activity diagram)	12
3.4 Диаграмма последовательности (Sequence diagram)	13
3.5 Диаграмма развёртывания (Deployment diagram)	14
3.6 Диаграмма сотрудничества (Collaboration diagram)	15
3.7 Диаграмма состояний (Statechart diagram)	17
4 Реализация	19
4.1 Средства реализации	19
4.2 Реализация backend	20

4.3 Реализация frontend	22
4.4 Навигация по приложению.....	23
5 Тестирование.....	30
5.1 UI-тестирование.....	30
5.2 Дымовое тестирование.....	31
5.3 Usability-тестирование	32
Заключение.....	34
Список использованной литературы	35

Введение

В настоящее время сложно представить мир без передвижений на большие расстояния, в век цифровых технологий человек привык экономить свое время, поэтому, когда необходимо спланировать поездку на отдых или по работе, люди предпочитают покупать авиабилеты, бронировать отели через Интернет. Существует большое множество информационных систем, помогающих выбрать и купить билет на рейс.

Данный проект направлен на создание веб-приложения с возможностью просматривать рейсы, покупать авиабилеты. Разработка подобной системы весьма актуальна на данный момент. В современном мире самолеты являются не только самым быстрым видом транспорта, но и самым безопасным, в связи с этим авиаперелеты пользуются весьма высокой популярностью.

1 Постановка задачи

Целью данного проекта является разработка веб-приложения, позволяющего оформлять покупку билетов авиакомпании с возможностью просмотра краткой информации о направлениях (странах) и мероприятиях, которые будут происходить в соответствующей стране.

1.1 Требования к разрабатываемой системе

1.1.1 Функциональные требования

К разрабатываемому сайту выдвигаются следующие функциональные требования:

- возможность просматривать список направлений;
- возможность выбирать направление и просматривать информацию о нем и о его мероприятиях;
- возможность ознакомиться со списком самолетов, осуществляющих авиаперевозку пассажиров компании;
- возможность регистрироваться, авторизоваться в системе;
- возможность оформлять и покупать авиабилет;
- возможность использования промокода на скидку при покупке билета;
- возможность ознакомиться с информацией о компании;
- возможность ознакомиться с контактными данными для связи;
- возможность просматривать историю своих заказов.

1.1.2 Нефункциональные требования

К разрабатываемому сайту выдвигаются следующие технические требования:

- регистрация и авторизация должна реализовываться с использованием личных данных пользователя;
- данные пользователей после регистрации должны храниться и идентифицироваться системой авторизации.

1.2 Задачи, решаемые в процессе разработки

В процессе разработки сайта будут решаться следующие задачи:

- сбор необходимой информации и предпроектное обследование;
- анализ предметной области, анализ системы и построение структуры требований, ведущих к решению поставленных задач и целей;
- построение модели программы, описание спецификаций данных, определение связей между сущностями, разработка модели БД;
- построение рабочего проекта, состоящего из написания кода, отладки и корректировки кода программы;
- проведение тестирования и доработка информационного программного обеспечения.

2 Анализ предметной области

2.1 Глоссарий

Направление – страна для осуществления перелета. Данное обозначение используется на сайте с целью привлечения внимания, поскольку элемент направления содержит не только информацию о рейсах, но и краткую информацию о стране и мероприятиях, которые в ней будут проводиться.

Промокод – состоящая из букв и/или цифр совокупность символов, дающая право на приобретение товара или услуги на специальных условиях.

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

Привилегированный пользователь – пользователь, прошедший авторизацию и имеющий расширенные возможности при работе с сайтом (выбор дополнительных направлений) и привилегии от авиакомпании (услуги осуществляются вне сайта, например: индивидуальный трансфер от аэропорта до самолета). Особым доступом обладают пользователи, которые превысили определенное число перелетов в рамках данной компании. В «Личном кабинете» отображается статус для данной группы пользователей.

2.2 Обзор аналогов

Прежде, чем начинать разработку сайта по продаже авиабилетов, необходимо проанализировать уже имеющиеся платформы для понимания их плюсов и минусов. После сбора информации можно переходить к этапу разработки с учетом сделанных выводов по изученным аналогам.

2.2.1 Aviasales

Aviasales – мета поисковик авиабилетов, существующий с 2007 года [1].

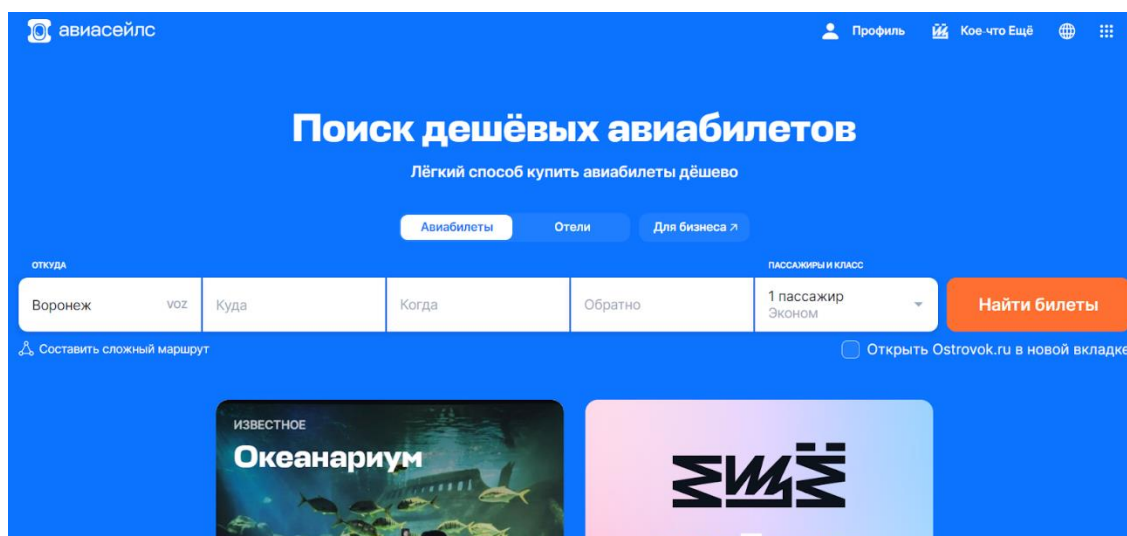


Рисунок 1 - Главная страница сайта «Aviasales»

Преимущества сайта:

- автоматическое составление маршрутов с пересадками;
- есть функция «подписка на цену», которая оповещает об изменении цены авиабилета на выбранную дату отправления;
- есть «тёмная тема».

Недостатки сайта:

- отсутствует возможность покупки авиабилета напрямую.

2.2.2 EasyJet

EasyJet – сайт британской лоу-кост-авиакомпании, основанной в 1995 году [2].

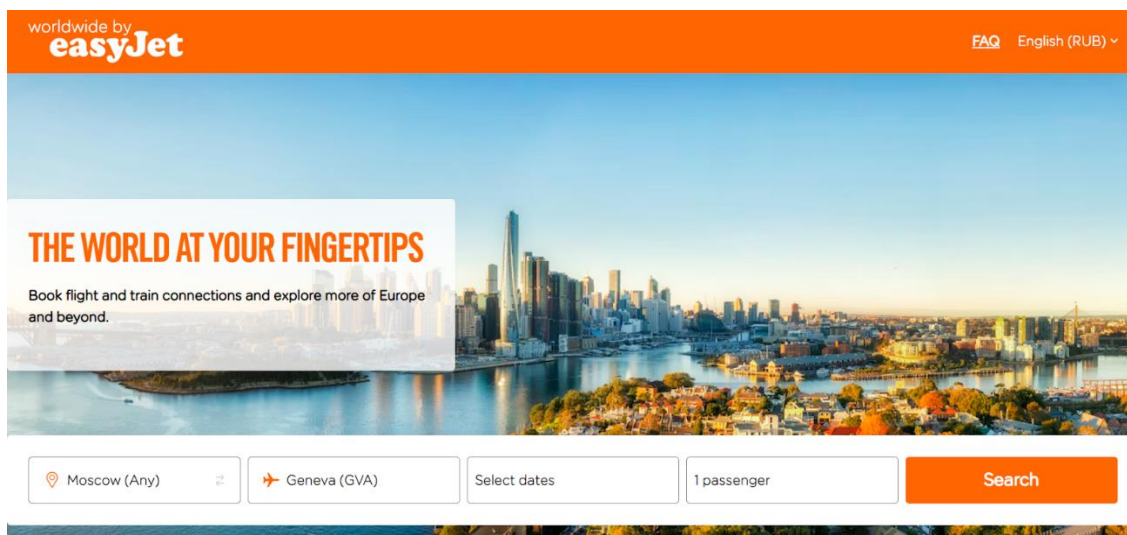


Рисунок 2 - Главная страница сайта «EasyJet»

Преимущества сайта:

- автоматическое составление сложных маршрутов совместно с авиакомпаниями-партнёрами;
- яркий и понятный интерфейс.

Недостатки сайта:

- отсутствует русскоязычная версия сайта.

2.2.3 OneTwoTrip

OneTwoTrip – онлайн-сервис для организации путешествий, позволяет найти и забронировать номер в отеле, купить ж/д, авиа и автобусные билеты, арендовать автомобиль, заказать экскурсию или приобрести тур [3].

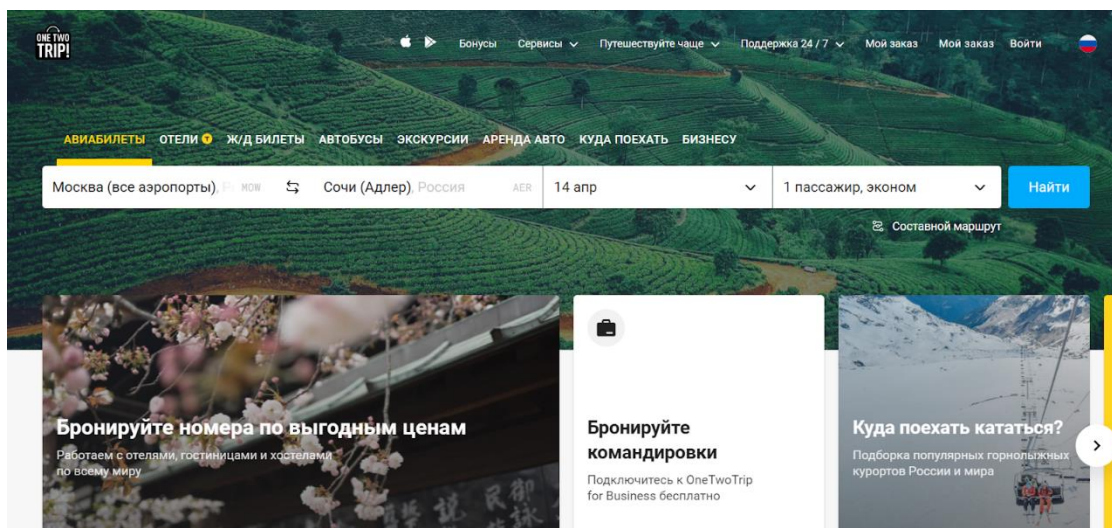


Рисунок 3 - Главная страница сайта «OneTwoTrip»

Преимущества сайта:

- автоматическое составление маршрутов с пересадками;
- можно купить авиабилет, не переходя на сторонние сайты.

Недостатки сайта:

- сервис является посредником при покупке авиабилета, по этой причине цена на билет больше, чем у самой авиакомпании.

После изучения аналогов разрабатываемого проекта можно сделать вывод, что ни один из них не является оптимальным решением поставленной задачи. Тем не менее большое количество различных сервисов, позволяющих приобретать авиабилеты, дает понять, что у людей есть потребность в этом. В связи с этим разработка проекта системы покупки электронных авиабилетов «SkyLink Airlines» является актуальной, потому необходимо будет учесть недостатки аналогов и не допустить их.

3 Диаграммы

3.1 Диаграмма прецедентов (Use-case diagram)

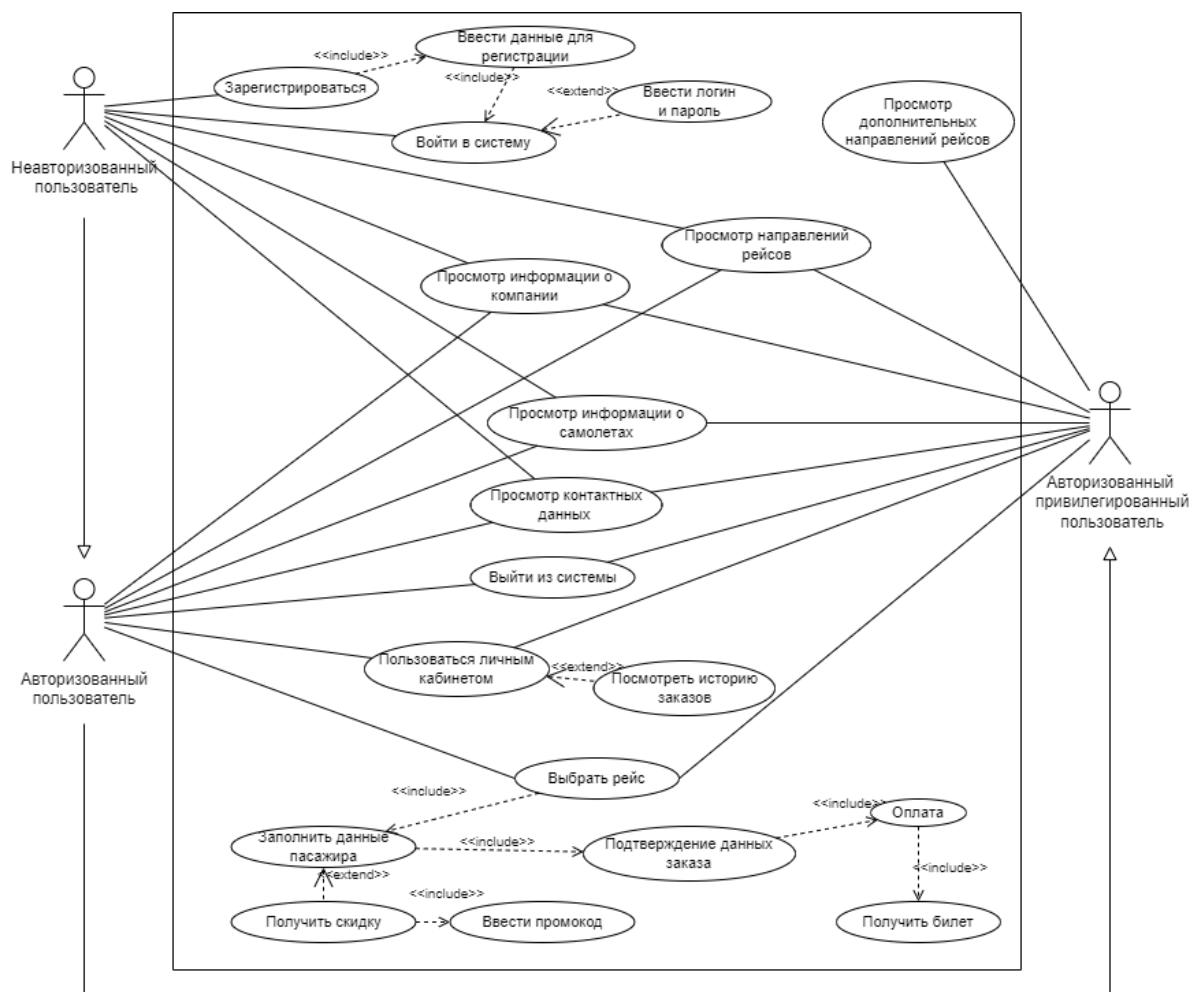


Рисунок 4 - Use-case diagram

Данная диаграмма отражает пользователей системы и их функции.

Неавторизованный пользователь – пользователь сайта, не зашедший в авторизованную зону системы и не имеющий полного доступа к функциям сайта.

Неавторизованный пользователь может:

- ознакомиться со списком направлений в соответствующем селекторе;
- выбирать направление и просматривать информацию о нем и о его мероприятиях;
- зарегистрироваться в системе;
- войти в систему.

Авторизованный пользователь — пользователь сайт, зарегистрированный или авторизованный в системе и имеющий полный доступ к функциям сайта.

Помимо функций неавторизованного пользователя, авторизованный может:

- просматривать историю своих заказов;
- выбирать и покупать билет на рейс;
- выйти из системы.

Авторизованный привилегированный пользователь — авторизованный пользователь сайта, имеющий расширенные возможности при работе с сайтом и привилегии от авиакомпаний.

Помимо функций авторизованного пользователя, авторизованный привилегированный пользователь может:

- просматривать дополнительные направления рейсов.

3.2 Диаграмма классов (Class diagram)

На данной диаграмме отображены классы системы, атрибуты с типами данных. Помимо этого, продемонстрировано взаимодействие между классами посредством связей.

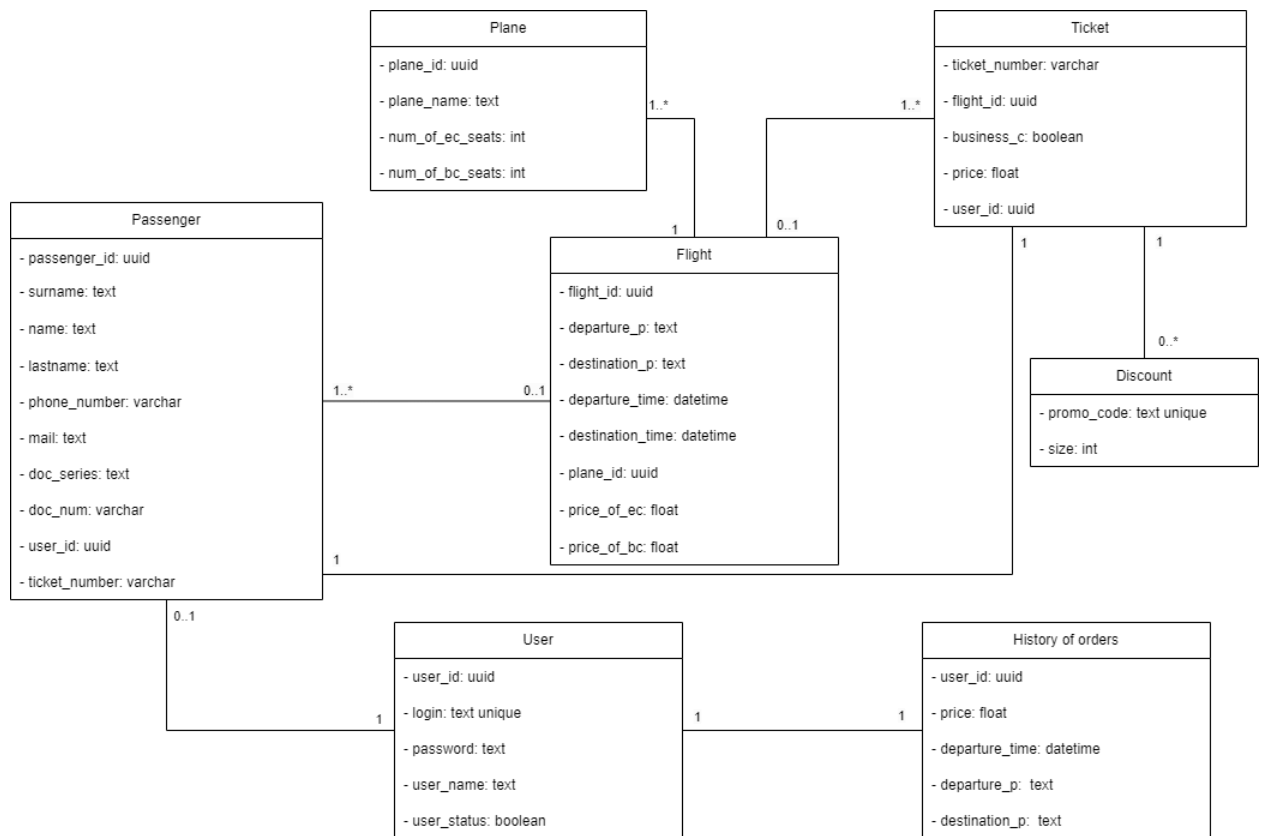


Рисунок 5 - Class diagram

3.3 Диаграмма активности (Activity diagram)

На данной диаграмме отображены возможные действия пользователя и их последовательность. Диаграмма показывает, что пользователь, находясь в неавторизованной зоне системы не может оформлять билет.

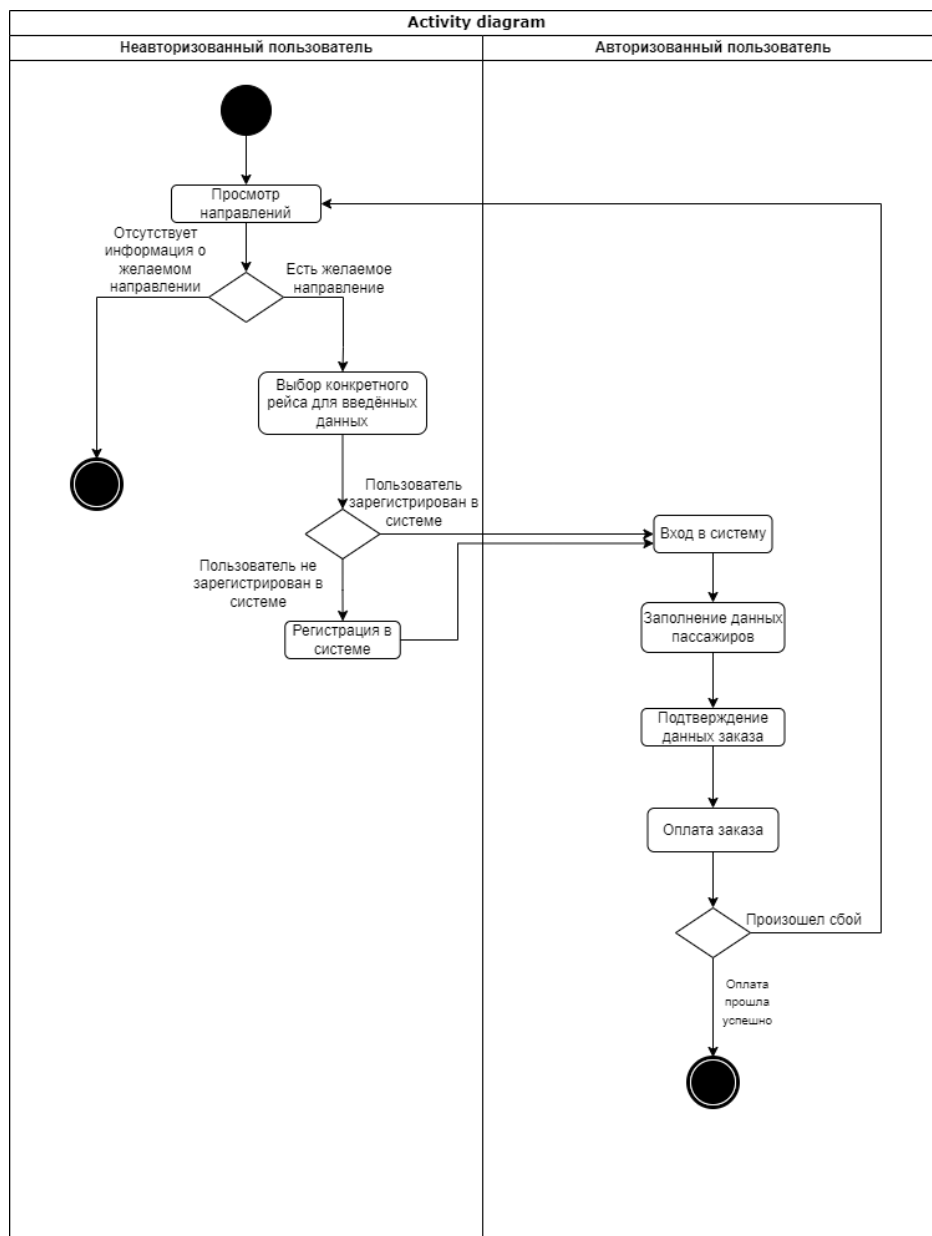


Рисунок 6 - Activity diagram

3.4 Диаграмма последовательности (Sequence diagram)

Данная диаграмма отражает взаимодействие различных частей системы между собой для выполнения функции, а также показывает последовательность действий, которые проводят к завершению этих функций. Целью диаграммы последовательности является документирование последовательности сообщений между объектами в представлении, основанном на времени.

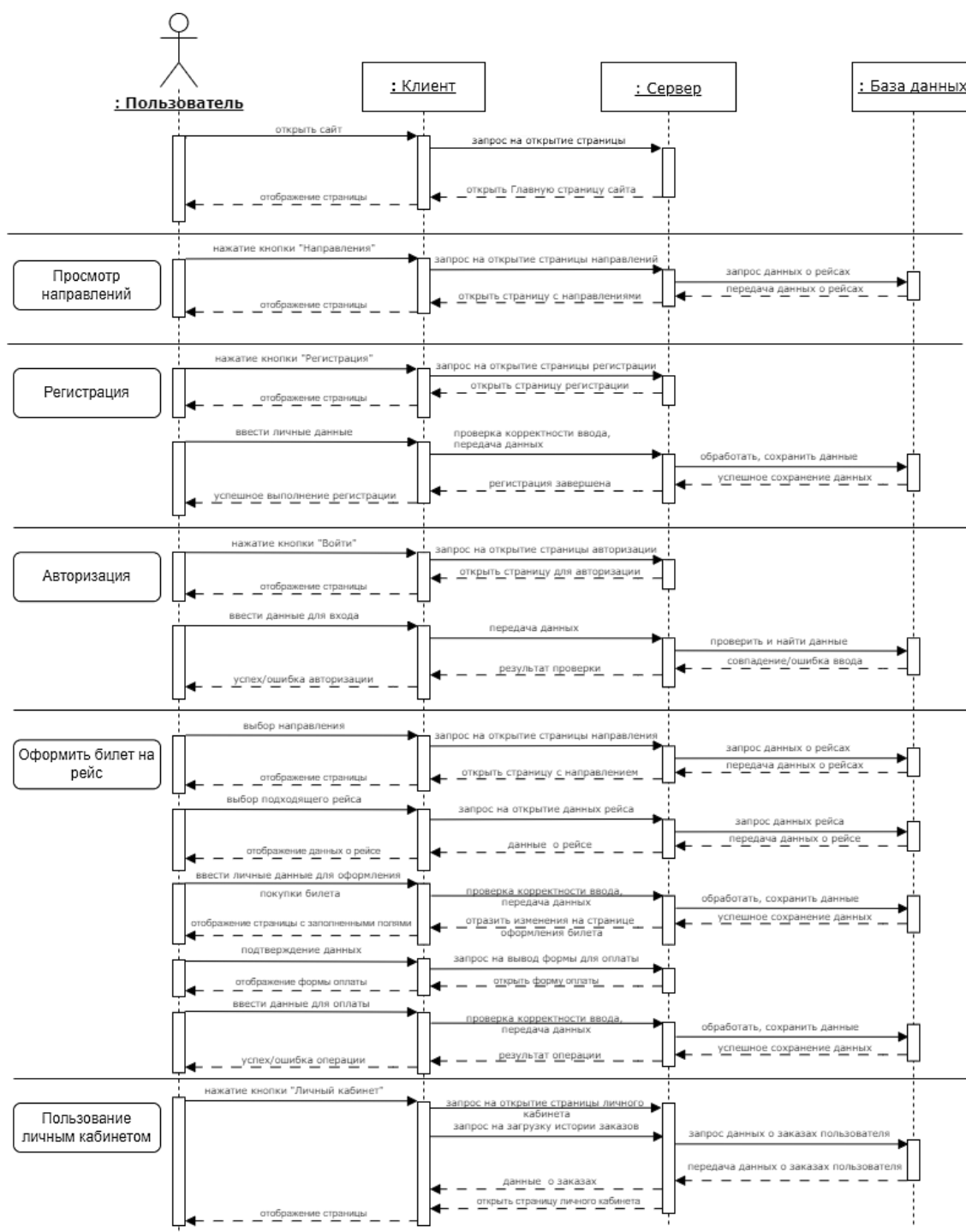


Рисунок 7 - Sequence diagram

3.5 Диаграмма развёртывания (Deployment diagram)

Данная диаграмма необходима для отражения программных компонентов, общей конфигурации и топологии распределенной программной системы.

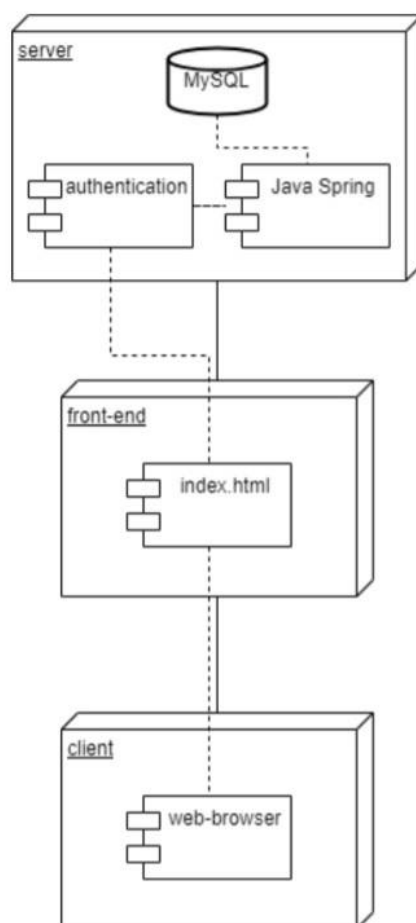


Рисунок 8 - Deployment diagram

3.6 Диаграмма сотрудничества (Collaboration diagram)

Данная диаграмма показывает динамическое взаимодействие объектов в системе. Отличительной особенностью является то, что она показывает объекты и их связь с другими объектами в системе, помимо того, как они взаимодействуют друг с другом.

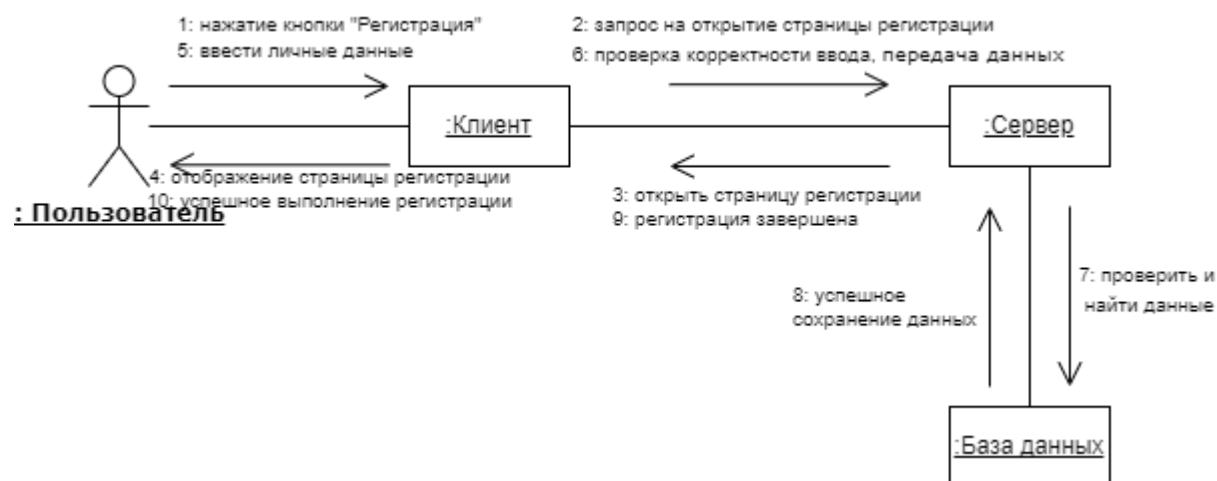


Рисунок 9 - Регистрация

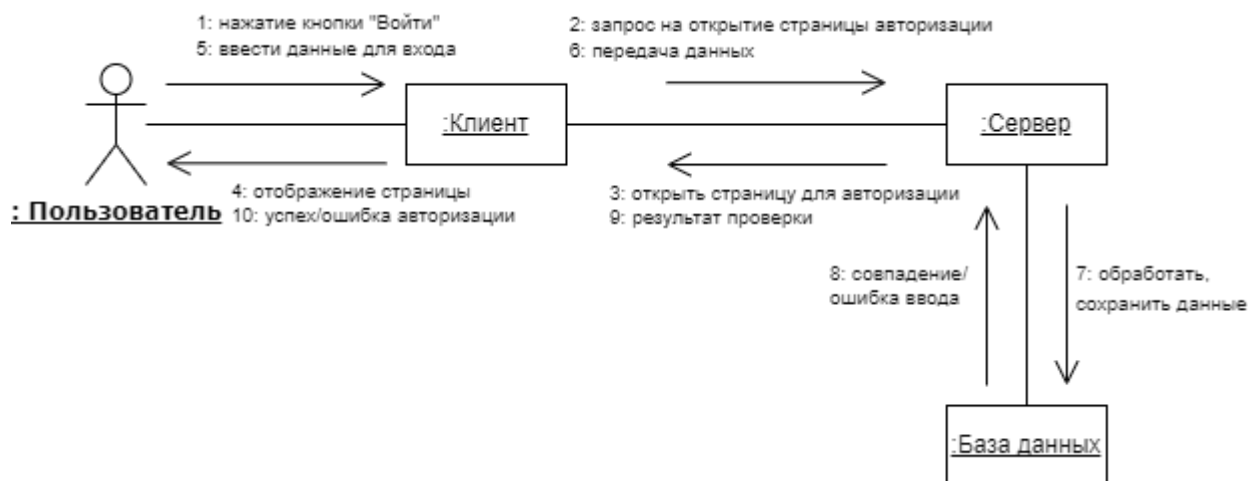


Рисунок 10 - Авторизация

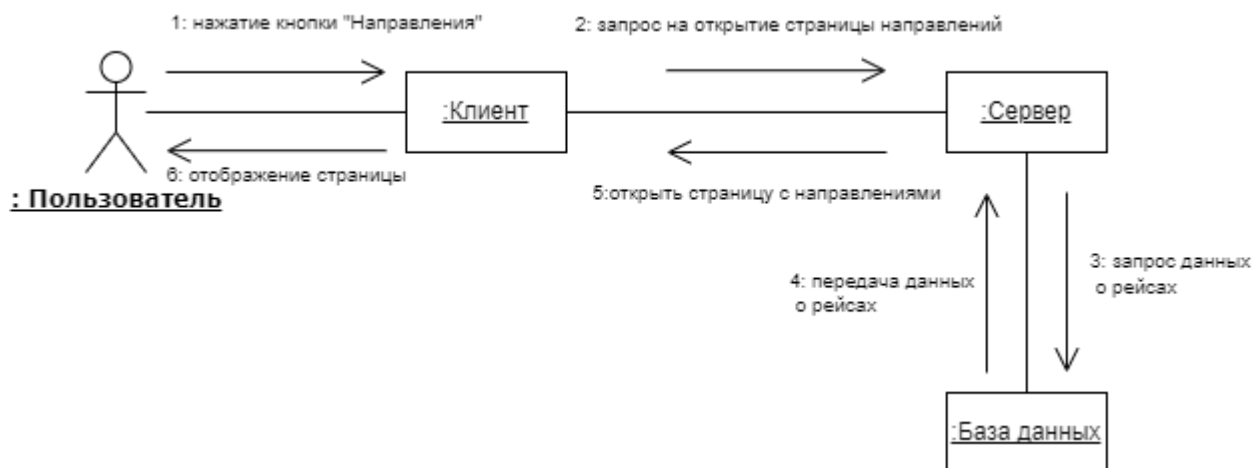


Рисунок 11 - Просмотр направлений

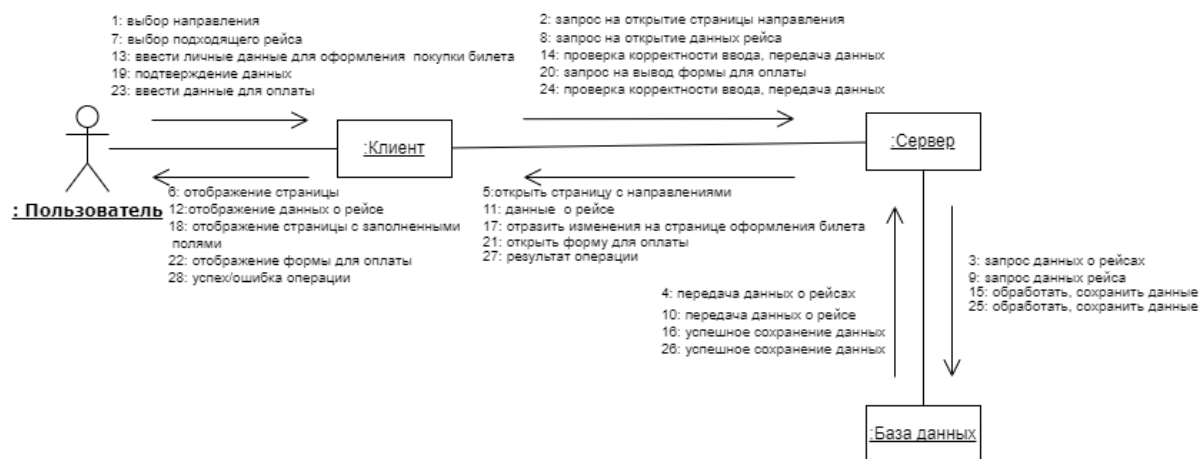


Рисунок 12 - Оформление и покупка билета

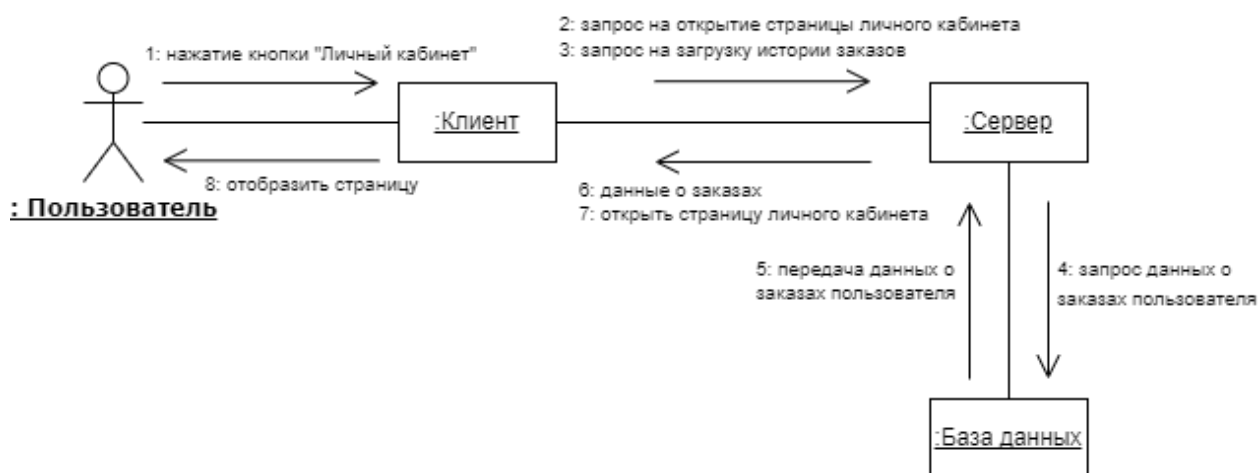


Рисунок 13 - Личный кабинет

3.7 Диаграмма состояний (Statechart diagram)

Диаграмма показывает, как объект переходит из одного состояния в другое. Диаграмма состояний полезна при моделировании жизненного цикла объекта. Характерной особенностью является то, что она описывает процесс изменения состояний только одного экземпляра определенного класса — одного объекта.

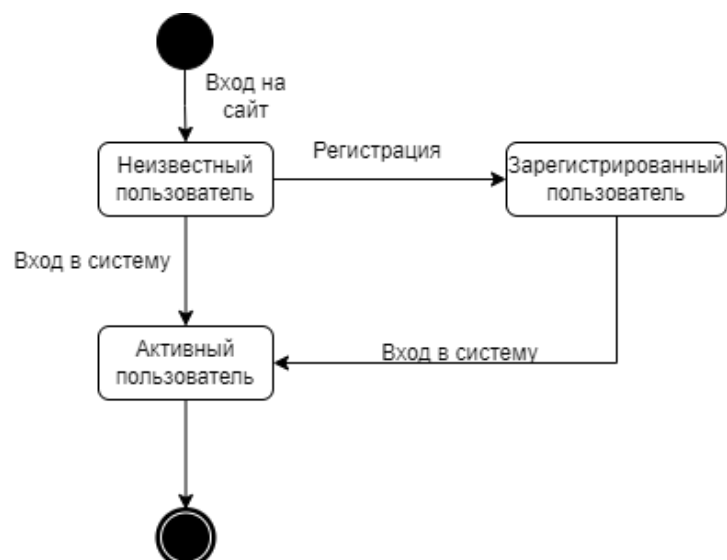


Рисунок 14 - Состояние авторизации

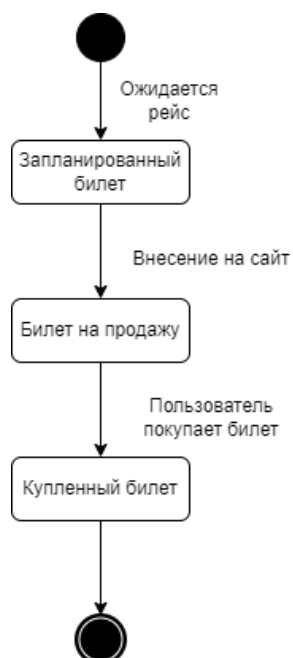


Рисунок 15 - Состояние билета

4 Реализация

4.1 Средства реализации

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

- Java — строго типизированный объектно-ориентированный язык программирования, характеризующийся платформенезависимостью. Он обладает следующими преимуществами: масштабируемость, безопасность, богатая стандартная библиотека, поддержка большого сообщества разработчиков [4].
- Spring Boot — фреймворк для разработки приложений на языке Java. Он используется для создания самостоятельных, готовых к работе приложений, которые легко масштабировать и поддерживать. Он обладает следующими преимуществами: ускорение разработки благодаря автоматической настройке и управлению зависимостями, встроенный контейнер сервлетов, интеграция с другими модулями и библиотеками Spring [5].
- MySQL — это система управления базами данных (СУБД) с открытым исходным кодом, которая также основана на языке SQL. Она обладает следующими преимуществами: расширяемость, надежность, поддержка транзакций, масштабируемость, гибкость модели данных [6].

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

- JavaScript — высокоуровневый интерпретируемый язык программирования, который применяется веб-разработке для создания интерактивных элементов на веб-страницах. Он обладает следующими преимуществами: широкая поддержка, гибкость, возможность создания динамических и интерактивных веб-страниц, доступ к различным библиотекам и фреймворкам.

- CSS — язык стилей, используемый для оформления веб-страниц. Он определяет внешний вид и расположение элементов на странице, включая цвета, шрифты, размеры, отступы, границы и другие стилевые свойства. Он обладает следующими преимуществами: возможность изменения внешнего вида всего сайта путем изменения одного файла CSS, улучшение доступности и поддержки мобильных устройств, возможность переиспользования стилей благодаря созданию классов и переопределения стилей для различных элементов.
- HTML — язык разметки, используемый для создания структуры и содержимого веб-страниц. 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 [7].

4.2 Реализация backend

Серверная (backend) часть приложения была написана на языке Java с использованием фреймворка Spring Boot. Spring Boot подразумевает разделение приложения на модули (прослойки), которые будут описаны далее. Каждый модуль организован в виде отдельного пакета. Структура приложения и иерархия пакетов модулей изображена на рисунке 16.

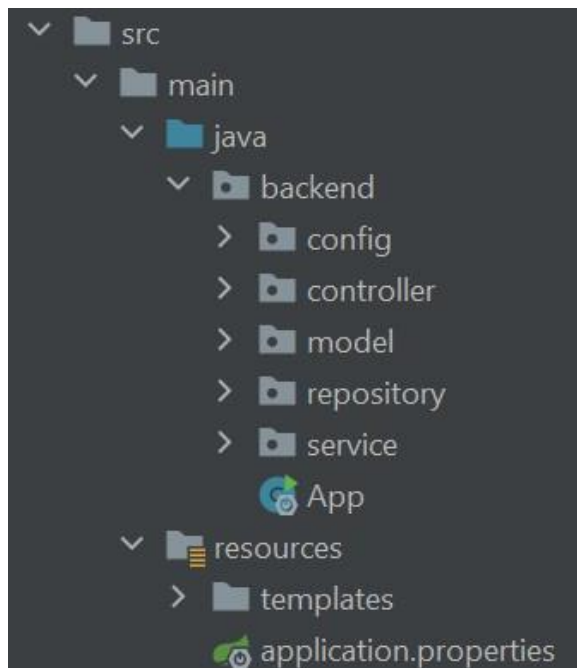


Рисунок 16 - Структура серверной части приложения

Приложение разделено на следующие модули:

- Модуль приложения (Application module). Это основной модуль, который содержит классы и настройки, связанные с запуском и конфигурацией приложения Spring Boot. В этом модуле находится главный класс приложения с аннотацией `@SpringBootApplication` и файл конфигурации `application.properties`, содержащий параметры подключения к базе данных. Для защиты базы данных эти параметры не объявляются явно в коде, а передаются в программу через переменные окружения.
- Модуль контроллеров (Controller Module). В этом модуле находятся классы контроллеров, отвечающие за обработку HTTP-запросов и взаимодействие с клиентом (браузером). Контроллеры содержат аннотации, такие как `@Controller` и `@RequestMapping`, для определения эндпоинтов и обработки запросов.
- Модуль конфигурации (Configuration Module). В этом модуле находятся классы конфигурации Spring, которые содержат настройки и бины, связанные с приложением. В случае разрабатываемого приложения в

модуле конфигурации два класса `MvcConfig` и `WebSecurityConfig`, в которых определены и настроены основные компоненты Spring Security для обеспечения безопасности серверной части приложения.

- Модуль моделей (Model Module). Этот модуль содержит классы моделей данных, которые отображают структуру таблиц в базе данных и представляют объекты бизнес-логики. Модели аннотированы аннотациями JPA, такими как `@Entity` и `@Table`, и содержат поля, геттеры и сеттеры.
- Модуль репозитория (Repository Module). В этом модуле находятся классы репозитория, отвечающие за взаимодействие с базой данных. Репозитории используют Spring Data JPA и содержат аннотацию `@Repository`. Они предоставляют методы для выполнения операций CRUD (создание, чтение, обновление, удаление) и других запросов к базе данных.
- Модуль сервисов (Service Module). Этот модуль содержит бизнес-логику приложения, которая обрабатывает запросы, полученные от контроллеров. Здесь находятся классы сервисов с аннотацией `@Service`, которые выполняют определенные операции и взаимодействуют с репозиториями.

4.3 Реализация frontend

Клиентская часть приложения (frontend) разработана на языке JavaScript. Для взаимодействия с серверной частью приложения на клиентской части используется библиотека `thymeleaf`, которая предоставляет интерфейс для выполнения HTTP-запросов из браузера.

Главной страницей сайта является HTML файл, который называется `index.html`, именно с него осуществляется навигация и переход на остальные страницы сайта.

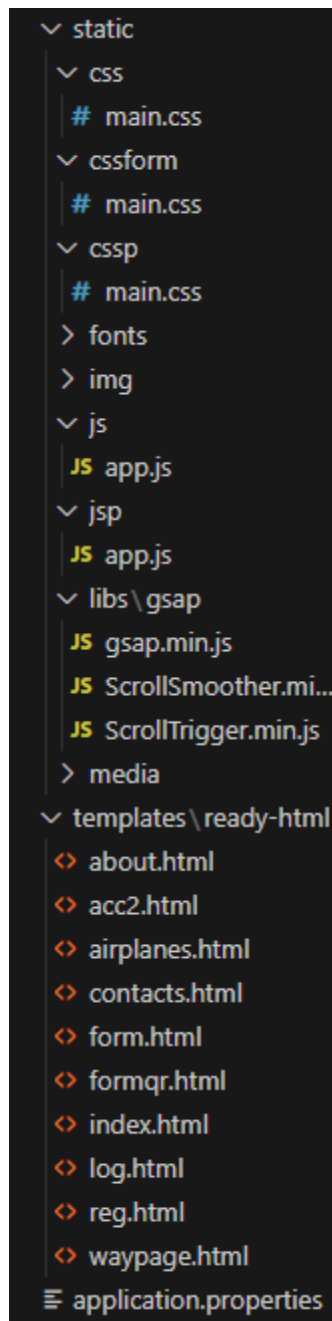


Рисунок 17 - Структура клиентской части приложения

Приложение имеет некоторые глобальные переменные, которые могут понадобиться в различных частях приложения, например, данные об авторизованном пользователе необходимы для оформления билета, отображения его истории заказов и т.д.

4.4 Навигация по приложению

Первое, что видит пользователь при входе на сайт, – главный экран. У пользователя есть выбор: войти в уже существующий аккаунт, пройти

регистрацию или продолжить пользоваться приложением без возможности использования расширенного функционала.



Рисунок 18 - Главная страница

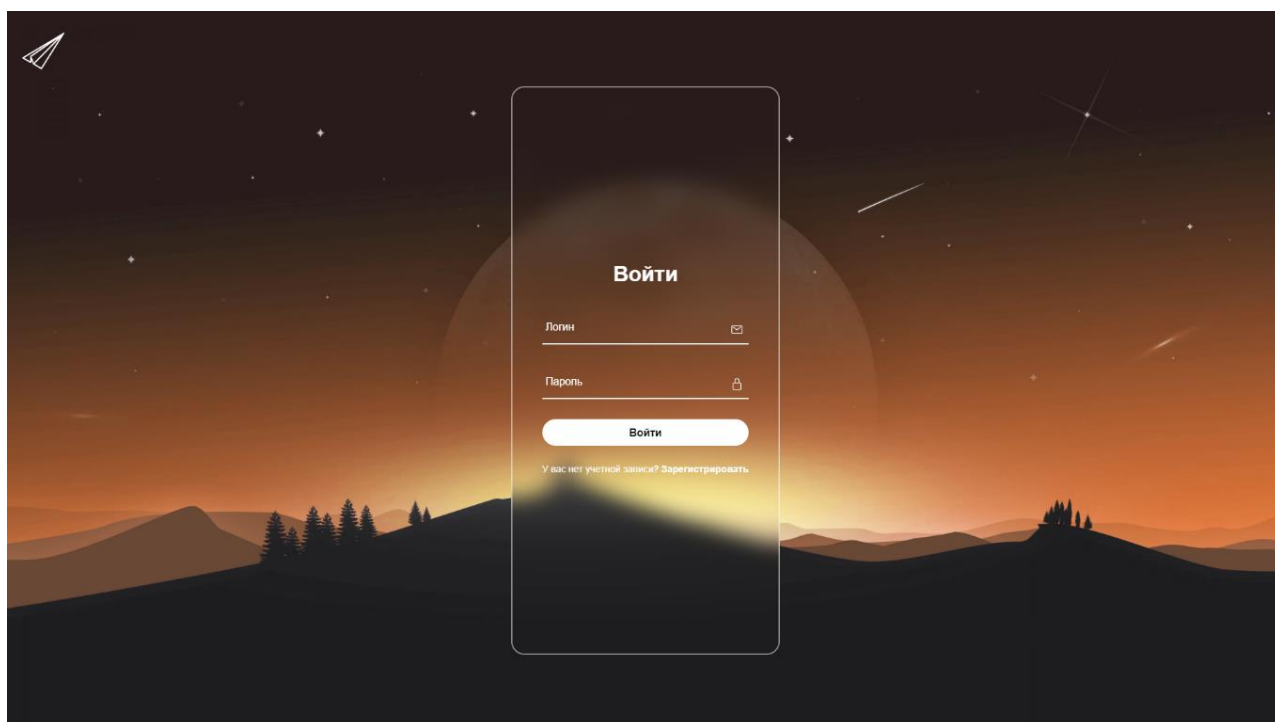


Рисунок 19 - Экран входа в существующий аккаунт

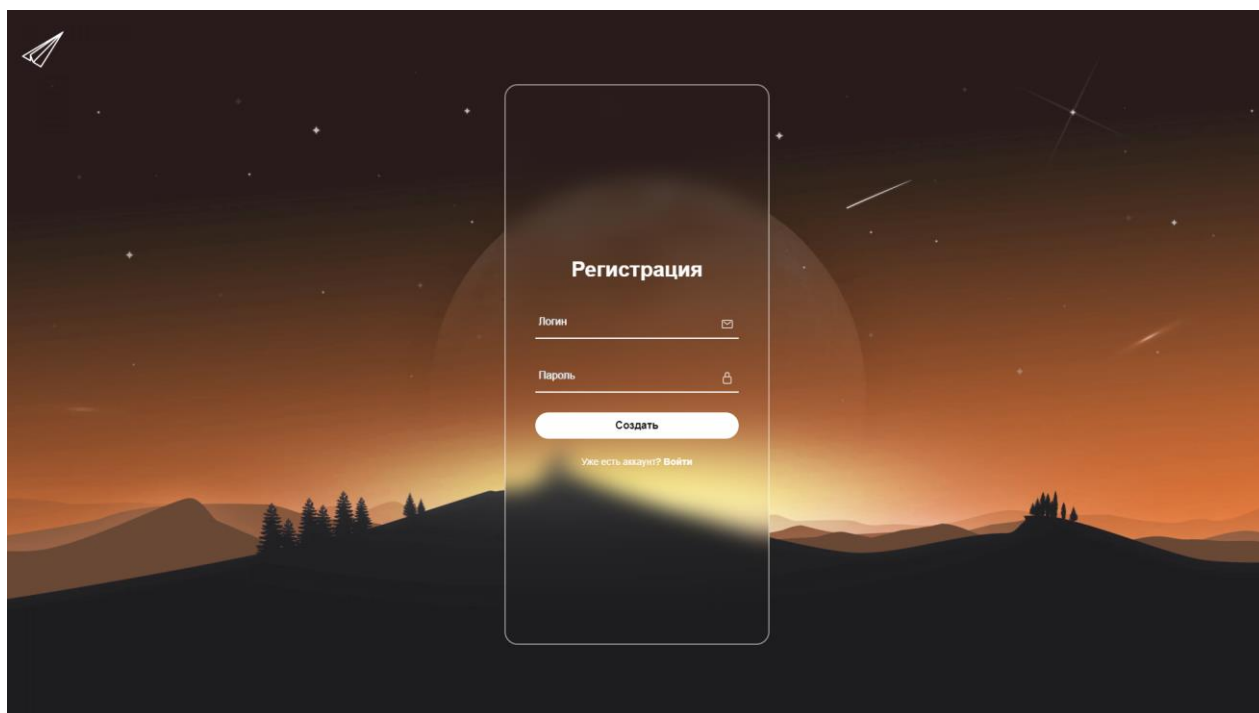


Рисунок 20 - Экран регистрации нового аккаунта

Также любой пользователь может просмотреть список направлений авиакомпании. Для привилегированного пользователя открывается возможность просмотра списка дополнительных направлений, которые включают дополнительные услуги для данной группы пользователей.

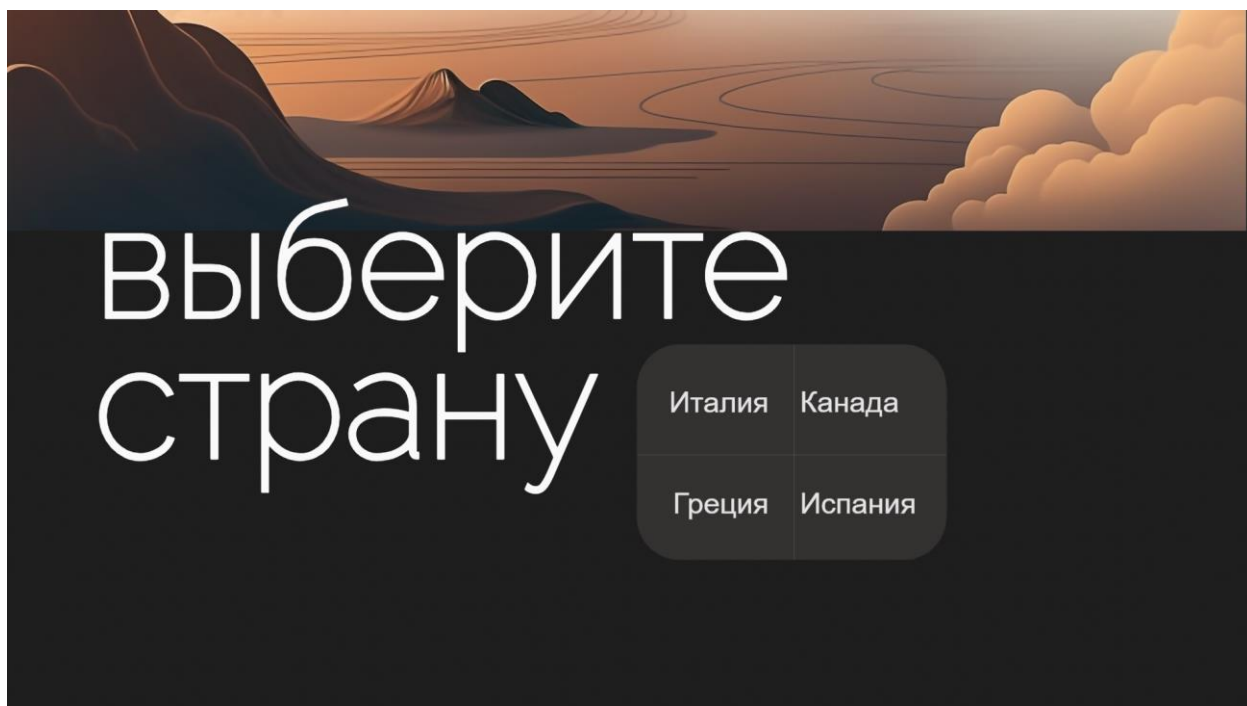


Рисунок 21 - Экран списка направлений для обычного пользователя

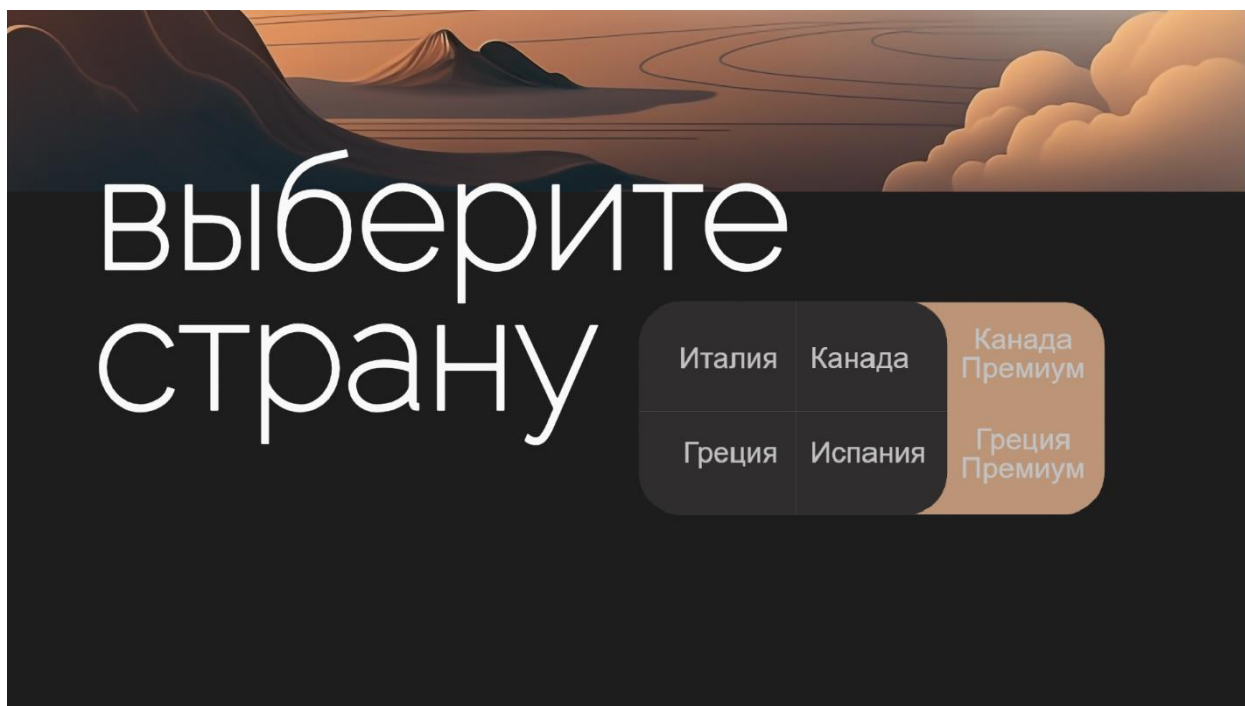


Рисунок 22 - Экран списка направлений для привилегированного пользователя

Пользователь может просмотреть дополнительную информацию о самолетах, осуществляющих перевозку, о компании, а также контактные данные.

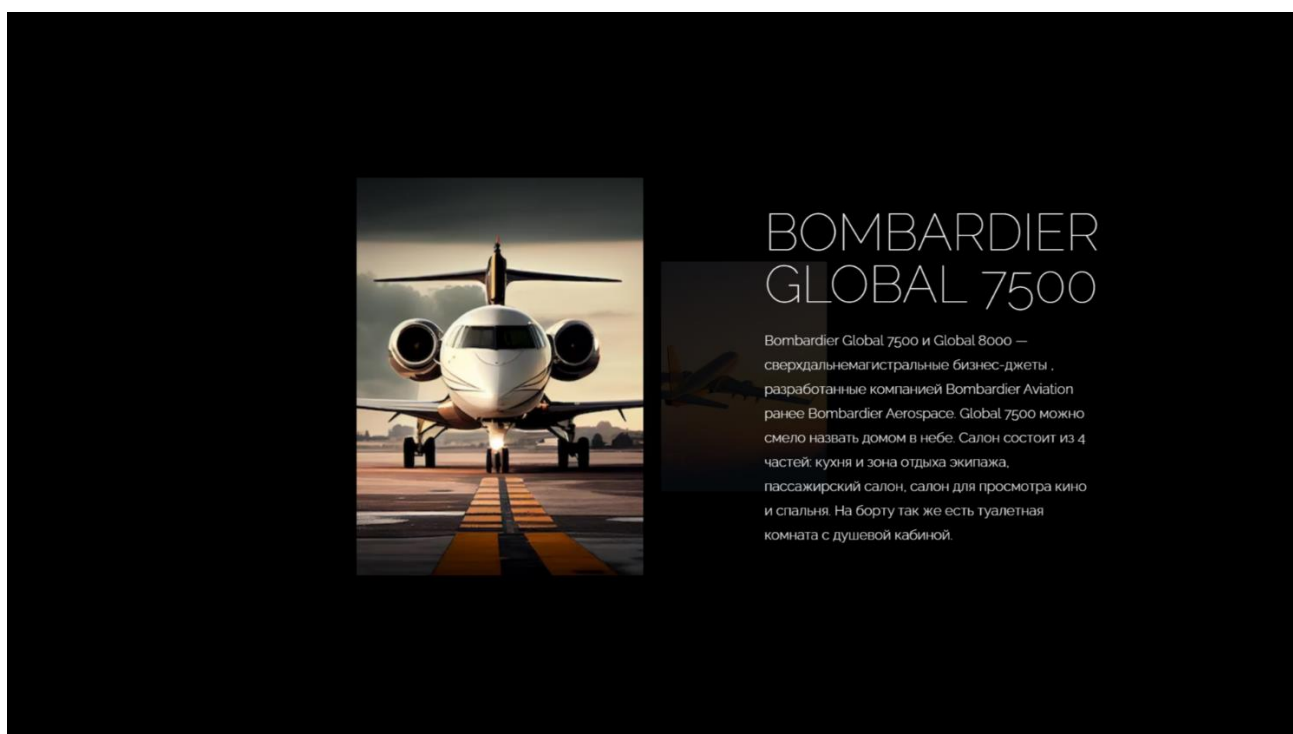


Рисунок 23 - Экран с дополнительной информацией о самолётах

Также пользователь может перейти на интересующее его направление, где при скроллинге страницы есть возможность ознакомиться с мероприятиями, которые будут происходить в соответствующей стране.

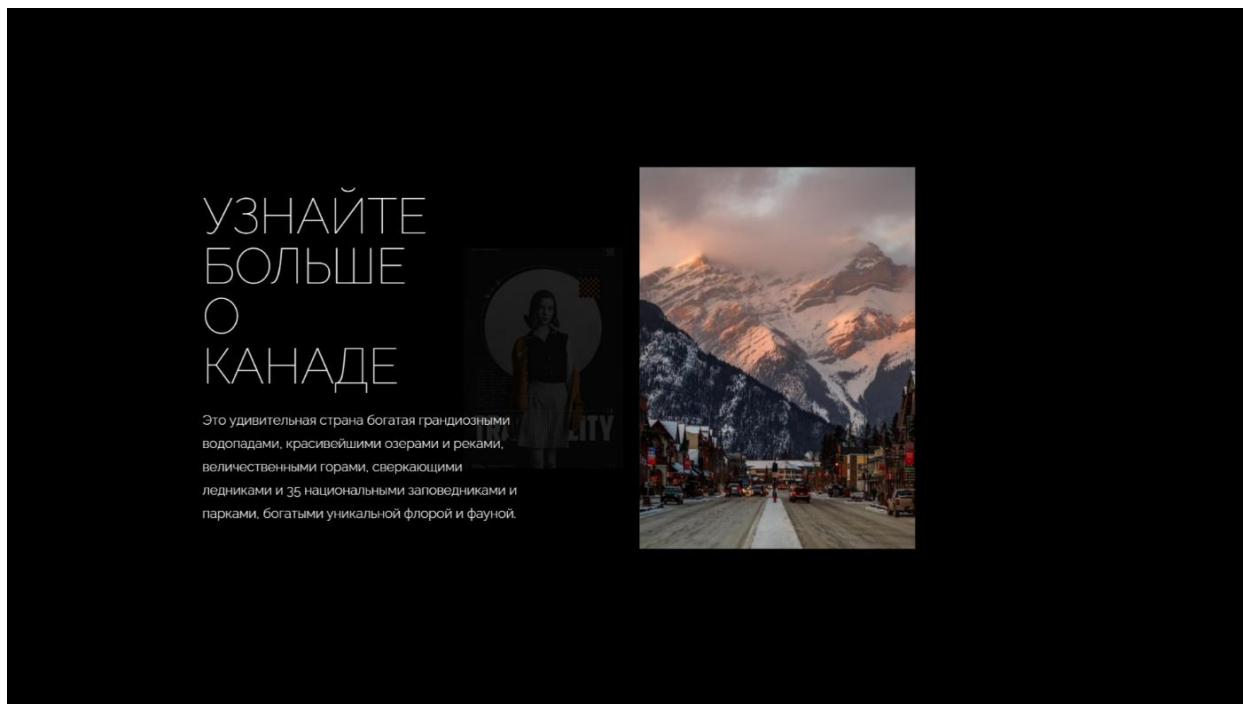


Рисунок 24 - Экран с дополнительной информацией о стране

Пользователь может оформить билет.

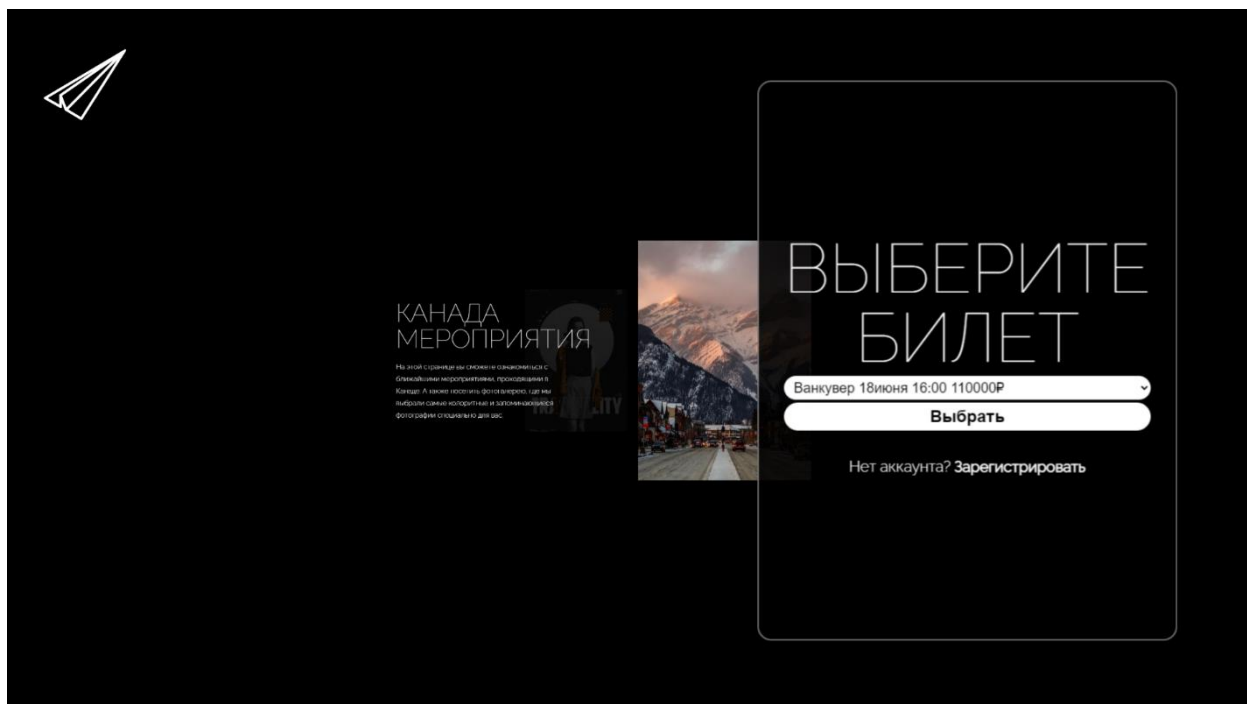
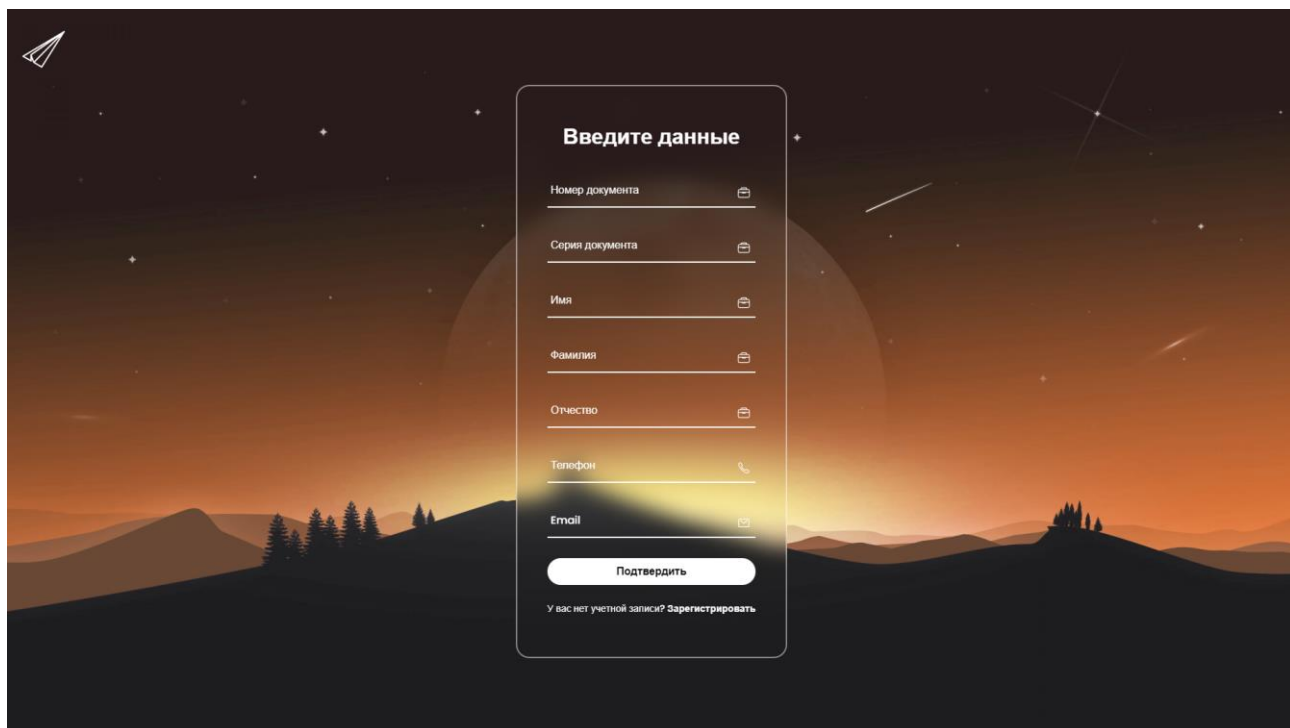




Рисунок 25 - Экран выбора авиабилета


После нажатия кнопки «Выбрать», пользователь переходит на страницу заполнения данных о пассажире.





Введите данные


Номер документа 


Серия документа 

Имя 

Фамилия 

Отчество 

Телефон 

Email 

Подтвердить

[У вас нет учетной записи? Зарегистрироваться](#)

Рисунок 26 - Экран ввода данных

Затем, нажав на кнопку «Подтвердить», пользователь переходит на страницу с оплатой авиабилета, где у него есть возможность применить промокод для получения скидки.

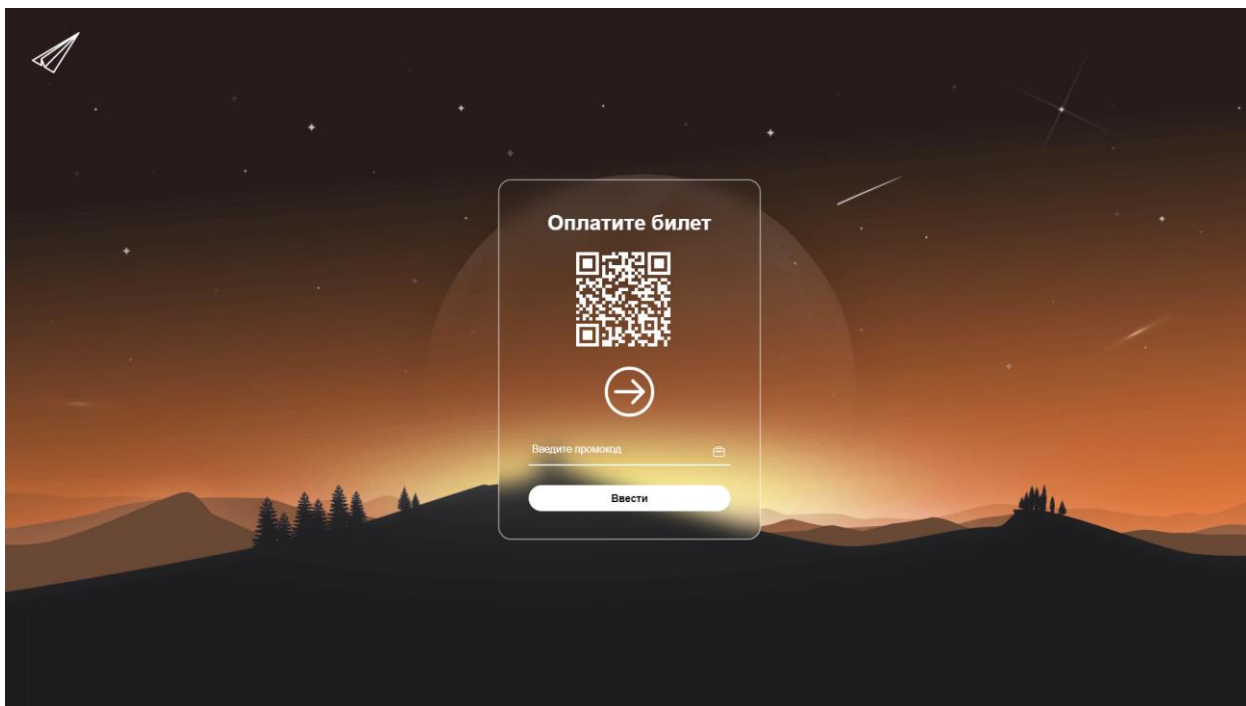


Рисунок 27 - Экран оплаты авиабилета

В личном кабинете пользователь имеет возможность ознакомиться с историей заказов.

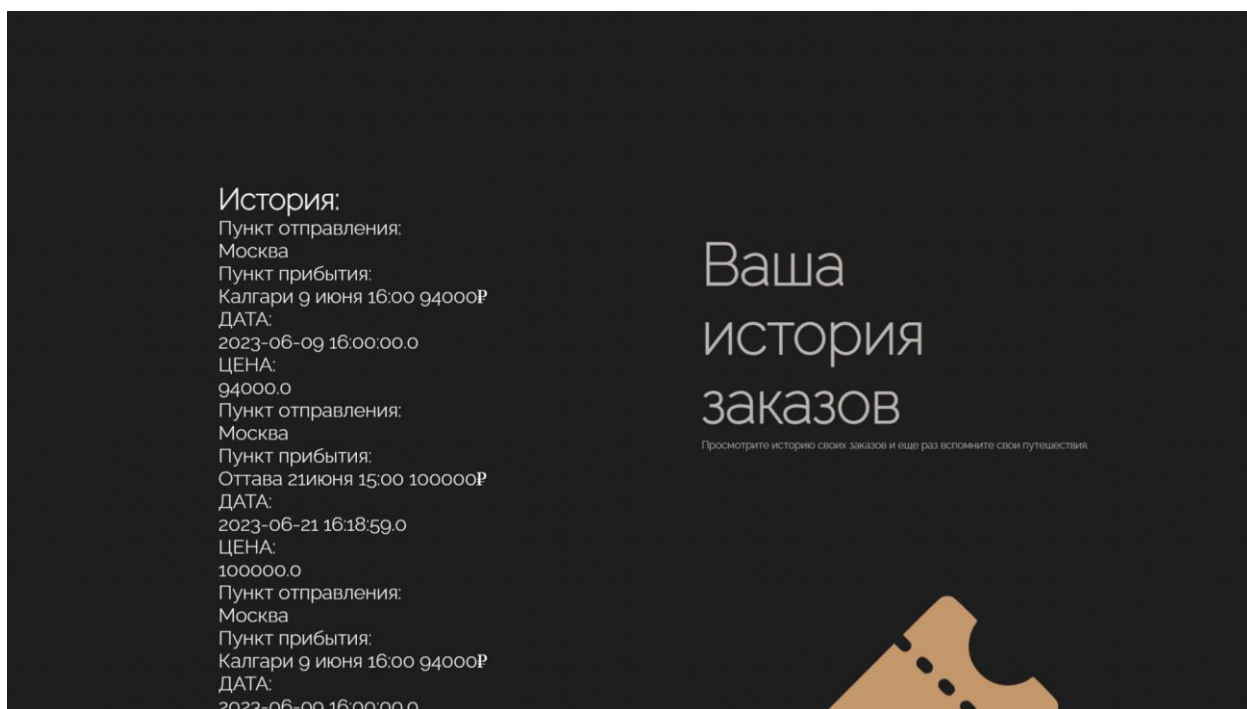


Рисунок 28 - Экран с историей заказов

5 Тестирование

Для проверки работоспособности разработанного приложения было проведено тестирование системы с использованием следующего типа тестов:

- UI-тестирование (тестирование пользовательского интерфейса);
- Smoke testing (дымовое тестирование);
- Usability-тестирование.

5.1 UI-тестирование

Тестирование пользовательского интерфейса (UI-тестирование) – это процесс проверки графического интерфейса приложения или веб-сайта на соответствие требованиям, удобство использования, визуальное оформление и правильное функционирование элементов пользовательского интерфейса.

Основная цель UI-тестирования заключается в убеждении, что пользовательский интерфейс выглядит и работает так, как ожидается, и что пользователи могут легко взаимодействовать с приложением или веб-сайтом [8].

В таблице 1 продемонстрированы результаты тестирования пользовательского интерфейса для неавторизованного пользователя.

Таблица 1 - Результаты UI-тестирования для неавторизованного пользователя

Тестовый сценарий	Ожидаемый результат	Статус теста
Нажатие на одну из стран на главной странице	Переход на страницу с выбранной страной	Пройден
Нажатие на кнопку «Вернуться на главную страницу»	Переход на главную страницу	Пройден
Нажатие на кнопку «Самолёты»	Переход на страницу с самолётами	Пройден
Нажатие на кнопку «О компании»	Переход на страницу с информацией о компании	Пройден

Нажатие на кнопку «Контакты»	Переход на страницу с контактными данными	Пройден
Нажатие на кнопку «Регистрация»	Переход на страницу с регистрацией	Пройден
Нажатие на кнопку «Войти»	Переход на страницу с авторизацией	Пройден

5.2 Дымовое тестирование

Дымовое тестирование (smoke testing) – это одна из методик функционального тестирования программного обеспечения, направленная на быструю проверку основных функций или основных компонентов приложения с целью выявления критических проблем.

Целью дымового тестирования является убедиться, что основные функциональности приложения работоспособны, а другие компоненты системы могут быть интегрированы и протестированы более подробно в дальнейшем [9].

В таблице 2 продемонстрированы результаты дымового тестирования для сценариев пользователя.

Таблица 2 - Результаты дымового тестирования для пользователя

Тестовый сценарий	Результат теста
Регистрация	Пройден
Авторизация	Пройден
Просмотр направлений	Пройден
Оформление билета	Пройден
Применение промокода	Пройден
Оплата билета	Пройден
Просмотр истории заказов	Пройден

5.3 Usability-тестирование

Usability-тестирование – это процесс оценки и измерения, насколько легко и удобно пользователи могут взаимодействовать с приложением, веб-сайтом или продуктом, а также насколько эффективно они могут достигать своих целей при использовании данного продукта [10].

В таблице 3 продемонстрированы результаты usability-тестирования для 4 респондентов, перед которыми был поставлен ряд задач.

Оценка успешности выполнения задания будет оцениваться из ряда трех показателей:

- в случае успешного выполнения задачи – хорошо;
- в случае если у респондента возникли проблемы, но он смог выполнить задачу – удовлетворительно;
- в случае не выполнения задачи – плохо.

Таблица 3 - Результаты usability-тестирования

Задание	Респондент 1	Респондент 2	Респондент 3	Респондент 4
Регистрация	хорошо	хорошо	хорошо	хорошо
Авторизация	хорошо	хорошо	хорошо	удовлетвор ительно
Просмотр направлений	хорошо	хорошо	хорошо	хорошо
Просмотр информации о самолётах	хорошо	удовлетвор ительно	хорошо	удовлетвор ительно
Просмотр информации о компании	хорошо	хорошо	хорошо	хорошо

Просмотр контактной информации	хорошо	хорошо	хорошо	удовлетворительно
Оформление билета	хорошо	удовлетворительно	удовлетворительно	хорошо
Применение промокода	хорошо	хорошо	хорошо	хорошо
Оплата билета	хорошо	удовлетворительно	хорошо	удовлетворительно
Просмотр истории заказов	хорошо	хорошо	удовлетворительно	хорошо

После проведённого опроса респондентам было предложено оставить отзыв о сайте (Рисунок 29).

Оставьте отзыв о функциональности сайта.
 Что понравилось, что не понравилось, что добавить или удалить на сайте?

4 ответа

удалить часть картинок со странами

Хороший сайт с необычной визуализацией. Интересная задумка с мероприятиями

В целом, данный сайт понравился своим удобством в использовании и понятным функционалом. Не очень понравилось, что при просмотре информации о самолетах и контактных данных происходит резкий переход на следующие экраны, после просмотра которых пропадает кнопочка "Вернуться назад".

Кнопка "Вернуться на главную страницу" на страницах с информацией активна не с начала появления, а лишь при определенном "приближении" к ней.

Рисунок 29 - Отзывы о работе сайта

Респонденты справились со всеми поставленными задачами. Полученные результаты являются возможными путями дальнейшего развития и доработки проекта.

Заключение

В ходе данной работы были выполнены все поставленные задачи. Было разработано веб-приложение для покупки авиабилетов. Благодаря нему пользователи могут онлайн:

- ознакомиться с направлениями рейсов;
- просмотреть информацию о мероприятиях в выбранной стране;
- оформить и купить билет;
- воспользоваться промокодом;
- просмотреть историю своих заказов.

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

Список использованной литературы

1. Aviasales [Электронный ресурс]. — Режим доступа: <https://www.aviasales.ru/>. — Заглавие с экрана. — (Дата обращения: 11.03.2023).
2. EasyJet [Электронный ресурс]. — Режим доступа: <https://worldwide.easyjet.com/>. — Заглавие с экрана. — (Дата обращения: 11.03.2023).
3. OneTwoTrip [Электронный ресурс]. — Режим доступа: <https://www.onetwotrip.com/>. — Заглавие с экрана. — (Дата обращения: 11.03.2023).
4. Backend разработка [Электронный ресурс]. — Режим доступа: <https://mobile-erp.ru/backend-razrabotka/>. — Заглавие с экрана. — (Дата обращения: 16.05.2023).
5. Введение в Spring Boot [Электронный ресурс]. — Режим доступа: <https://habr.com/ru/articles/435144/>. — Заглавие с экрана. — (Дата обращения: 22.05.2023).
6. Что такое MySQL [Электронный ресурс]. — Режим доступа: <https://timeweb.com/ru/community/articles/chto-takoe-mysql-server>. — Заглавие с экрана. — (Дата обращения: 21.05.2023).
7. Фронтенд [Электронный ресурс]. — Режим доступа: <https://www.gorkilib.ru/events/frontend>. — Заглавие с экрана. — (Дата обращения: 18.05.2023).
8. Как проводить UI-тестирование мобильных и верстки + сравнений инструментов автоматизации [Электронный ресурс]. — Режим доступа: <https://ux-journal.ru/kak-provodit-ui-testirovanie-sravnenieinstrumentov.html#testcase-example>. — Заглавие с экрана. — (Дата обращения: 25.05.2023).

9. Smoke-тестирование [Электронный ресурс]. — Режим доступа: <https://blog.skillfactory.ru/glossary/smoke-test/>. — Заглавие с экрана. — (Дата обращения: 25.05.2023).
10. Юзабилити-тестирование: пошаговая инструкция на примере Yelp [Электронный ресурс]. — Режим доступа: <https://lpgenerator.ru/blog/2016/07/28/yuzabiliti-testirovanie-poshagovayainstrukciya-na-primere-yelp/>. — Заглавие с экрана. — (Дата обращения: 25.05.2023).