

МИНОБРНАУКИ РОССИИ  
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ  
УЧРЕЖДЕНИЕ  
ВЫСШЕГО ОБРАЗОВАНИЯ  
“ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ”

Факультет компьютерных наук

Кафедра программирования и информационных технологий

Мобильное приложение «Приложение для организации совместных поездок,  
включающее профили пользователей, систему отзывов и расчет стоимости  
поездки TripShare»

Курсовой проект  
по дисциплине  
Технологии программирования

09.03.04 Программная инженерия

6 семестр 2023/2024 учебного года

Зав. кафедрой	_____	С. Д. Махортов, д.ф.-м.н., профессор
Обучающийся	_____	А. А. Тарарыков, ст. 3 курса оч. отд.
Обучающийся	_____	Е. А. Рогачев, ст. 3 курса оч. отд.
Обучающийся	_____	В. Е. Платон, ст. 3 курса оч. отд.
Руководитель	_____	В.С. Тарасов, ст. преподаватель
_____.____.20____		

Воронеж 2024

## СОДЕРЖАНИЕ

ВВЕДЕНИЕ .....	4
1 Постановка задачи.....	5
2 Требования к разрабатываемой системе.....	6
2.1 Функциональные требования.....	6
2.1.1 Для неавторизованных пользователей.....	6
2.1.2 Для незарегистрированных пользователей .....	7
2.1.3 Для зарегистрированных пользователей .....	7
2.1.4 Для пассажиров .....	8
2.1.5 Для водителей.....	9
2.2 Технические требования.....	10
2.3 Требования к интерфейсу .....	10
2.4 Задачи, решаемые в процессе разработки .....	11
3 Анализ предметной области .....	12
3.1 Обзор аналогов .....	12
3.1.1 BlaBlaCar .....	12
3.1.2 UberPool.....	13
3.1.3 CarpoolWorld.....	15
3.1.4 Едем.рф.....	16
3.1.5 Автостоп.....	18
3.2 Моделирование системы .....	18
3.2.1 Диаграмма состояний .....	18
3.2.2 Диаграмма последовательности .....	19
3.3 Аналитика веб-приложения .....	31
3.4 Границы проекта .....	31
4 Реализация.....	32
4.1 Средства реализации.....	32
4.2 Реализация базы данных.....	33

4.3 Реализация серверной части приложения .....	35
4.4 Реализация клиентской части приложения .....	39
4.4.1 Общая информация .....	39
4.4.2 Графический интерфейс .....	39
5 Тестирование .....	46
5.1 Процесс тестирования .....	46
5.2 Результаты тестирования .....	46
Заключение .....	47

## **ВВЕДЕНИЕ**

Современное общество сталкивается с проблемами транспортных пробок, загрязнения окружающей среды и высокими затратами на транспорт. Для решения этих проблем было разработано приложение для организации совместных поездок, которое призвано упростить процесс планирования и использования транспорта. Это приложение предоставляет пользователю возможность поиска совместных поездок, создания профилей пользователей, системы отзывов и рейтингов, а также расчета стоимости поездки.

## **1 Постановка задачи**

Целью данной работы является разработка приложения для организации совместных поездок с целью упрощения процесса планирования и использования транспорта. На основе вышеописанных назначения и целей приложения выделяются следующие основные задачи:

- Разработка функционала для пользователей;
- Разработка архитектуры системы;
- Выбор технологий и интеграция внешних интерфейсов;
- Оформление и верстка страниц;
- Система администрирования;
- Тестирование и обеспечение качества;
- Документирование и поддержка;
- Подготовка документации для разработчиков и пользователей;
- Обеспечение поддержки и обновлений приложения;

## **2 Требования к разрабатываемой системе**

### **2.1 Функциональные требования**

#### **2.1.1 Для неавторизованных пользователей**

- обеспечение возможности перехода на страницу авторизации после нажатия на кнопку "Забронировать" при просмотре деталей поездки. После успешной авторизации или отмены авторизации пользователь будет возвращен на страницу с деталями поездки;
- обеспечение возможности перехода на страницу авторизации после нажатия на кнопку "Создать поездку". После успешной авторизации или отмены авторизации пользователь будет возвращен на главный экран;
- обеспечение возможности перехода на страницу авторизации после нажатия на кнопку "Запланированные поездки". После успешной авторизации или отмены авторизации пользователь будет перенаправлен на главную страницу;
- обеспечение возможности перехода на страницу авторизации после нажатия на кнопку "Чаты". После успешной авторизации или отмены авторизации пользователь будет возвращен на главный экран;
- обеспечение возможности перехода на страницу авторизации после нажатия на кнопку "Профиль". После успешной авторизации или отмены авторизации пользователь будет возвращен на главный экран;
- обеспечение возможности ввода номера телефона в отведенное для этого поле с маской и нажатия на кнопку "Продолжить" после перехода на страницу авторизации.

### **2.1.2 Для незарегистрированных пользователей**

То же самое, что и для неавторизованных пользователей, с добавлением следующих дополнительных задач:

- обеспечение возможности перехода на страницу регистрации после ввода номера телефона на странице авторизации и нажатия на кнопку "Продолжить";
- обеспечение возможности указания имени и загрузки аватарки при регистрации;
- обеспечение возможности создания пароля при регистрации.
- обеспечение возможности принятия условий пользовательского соглашения при регистрации;
- обеспечение возможности нажатия на кнопку "Зарегистрироваться" для завершения процесса регистрации;
- после успешной регистрации обеспечение возможности автоматического выполнения процесса аутентификации, чтобы пользователь стал авторизованным пользователем и имел доступ к функционалу, доступному только авторизованным пользователям;
- предоставление возможности для зарегистрированных пользователей использовать свой аккаунт для доступа ко всем функциям приложения без необходимости повторной регистрации.

### **2.1.3 Для зарегистрированных пользователей**

То же самое, что и для неавторизованных пользователей, с добавлением следующих дополнительных задач:

- обеспечение возможности перехода на страницу ввода пароля после ввода номера телефона на странице авторизации и нажатия на кнопку "Продолжить";
- обеспечение возможности ввода пароля на странице ввода пароля;

- обеспечение возможности нажатия на кнопку "Войти" после верного ввода пароля для завершения процесса аутентификации;
- после успешной аутентификации обеспечение возможности получения доступа ко всем функциям, доступным авторизованным пользователям;
- предоставление возможности для авторизованных пользователей использовать свой аккаунт для доступа ко всем функциям приложения без необходимости повторной аутентификации.

#### **2.1.4 Для пассажиров**

- обеспечение возможности поиска поездок с учетом следующих параметров: дата отправления, диапазон времени отправления, адрес отправления, адрес прибытия, количество пассажиров;
- обеспечение возможности отображения списка найденных поездок с информацией о стоимости, времени отправления и прибытия, времени поездки, адресе отправления и прибытия, а также аватаре, имени, рейтинге водителя, проценте совместимости музыкальных жанров, расстоянии до начальной и конечной точки, количестве пассажиров и иконках удобств;
- обеспечение возможности выбора сортировки найденных поездок по различным критериям;
- обеспечение возможности просмотра деталей выбранной поездки, включая день недели и дату, время отправления и прибытия, адреса отправления и прибытия, стоимость, марку, модель, цвет и год выпуска автомобиля водителя, а также список пассажиров и кнопку связи с водителем;
- обеспечение возможности открытия чата с водителем после нажатия на кнопку связи с ним;
- обеспечение возможности перехода на страницу профиля



- пользователя после нажатия на виджет с краткой информацией о пассажире;
- обеспечение возможности перехода на страницу проверки деталей поездки после нажатия на кнопку "Забронировать", где будет отображена информация о поездке и оповещение о том, что бронирование будет подтверждено после одобрения водителя;
- обеспечение возможности перехода на страницу с запланированными поездками после нажатия на кнопку "Отправить запрос".

### **2.1.5 Для водителей**

- обеспечение возможности перехода на страницу создания поездки после нажатия на кнопку "Создать поездку";
- обеспечение возможности ввода адреса отправления и адреса прибытия на странице создания поездки;
- обеспечение возможности выбора маршрута поездки;
- обеспечение возможности добавления адресов дополнительных остановок для взятия дополнительных пассажиров;
- обеспечение возможности указания даты и времени отправления;
- обеспечение возможности указания количества попутчиков, которые могут расположиться на заднем сиденье, и сколько попутчиков можно взять в дорогу;
- обеспечение возможности выбора дополнительных удобств с помощью чекбоксов;
- обеспечение возможности указания цены за место;
- обеспечение возможности выбора автомобиля, ранее добавленного в профиль, на котором будет совершаться поездка;
- обеспечение возможности добавления описания поездки;

- обеспечение возможности нажатия на кнопку "Создать поездку" для завершения процесса создания поездки;
- после нажатия на кнопку "Создать поездку" обеспечение возможности перехода на страницу с запланированными поездками.

## **2.2 Технические требования**

Программный продукт должен обеспечить:

- алгоритм поиска, учитывающего местоположение и предпочтения пользователей;
- фильтрацию результатов поиска на основе критериев, таких как направление движения, время отправления и прибытия;
- добавление, изменение и удаление пользователями поездок;
- интерфейс для управления временем отправления и прибытия, а также другими деталями поездки;
- возможность для составления маршрута и расчета времени в пути;
- возможность добавления остановок или изменения маршрута в процессе планирования;
- систему рейтинга и отзывов для пользователей;
- алгоритм анализа поведения пользователей для выявления подозрительной активности;
- алгоритм для расчета стоимости поездки на основе расстояния, времени и других факторов;

## **2.3 Требования к интерфейсу**

Интерфейс должен иметь единую цветовую гамму и стиль для всех экранов. Все надписи должны быть легко читаемыми, а элементы управления

- выполненными в едином стиле и размере, выделяясь на фоне содержимого экранов.

Интерфейс должен содержать только необходимую пользователю информацию, расположенную в соответствующих местах приложения. Основные элементы управления должны быть хорошо заметны.

## **2.4 Задачи, решаемые в процессе разработки**

Перед проектом были поставлены следующие задачи:

- анализ предметной области;
- анализ аналогов;
- написание технического задания;
- описание разрабатываемой системы UML диаграммами;
- разработка БД;
- реализация ролей;
- реализация функциональных возможностей ролей;
- разработка функциональных возможностей сайта;
- создание макета дизайна и его реализация;
- реализация интерфейса;
- проведение тестирования;
- описание процесса разработки и результата.

## 3 Анализ предметной области

### 3.1 Обзор аналогов

#### 3.1.1 BlaBlaCar

BlaBlaCar – сервис, который помогает автомобилистам найти попутчиков, а пассажирам — найти подходящую поездку.

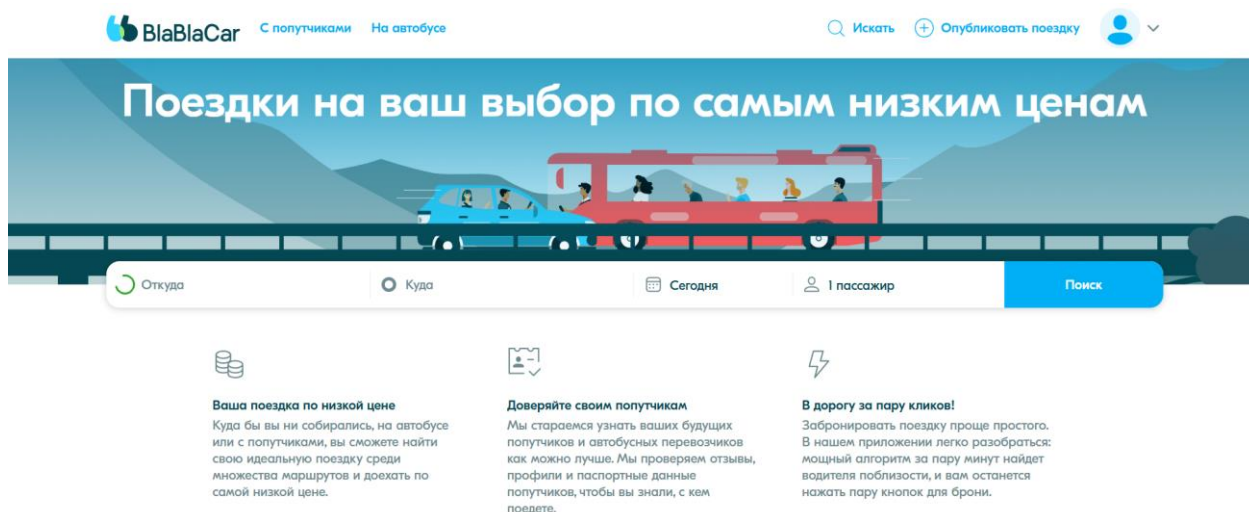



Рисунок 1 - Экран поиска поездок BlaBlaCar

Преимущества: Обширная пользовательская база и широкое международное покрытие обеспечивают множество доступных маршрутов и высокую степень удобства.


# Чт, 30 мая

 Поездка уже началась.

08:30  
6 ч 10

воронеж центр

Воронеж


 До места, откуда вы хотите выехать: 6 км

>

14:40

Выставка достижений народного хозяйства (ВДНХ)

Москва

 До места, куда вы хотите приехать: 8 км

>

Итого за 1 пассажира **1 750,00 ₽**

Владимир

★ 5/5 - 2 отзыва

 >



Профиль подтвержден



Редко отменяет поездки

Рисунок 2 - Экран деталей поездки BlaBlaCar

Недостатки: Высокая стоимость услуг в некоторых регионах и неравномерное качество услуг могут отталкивать потенциальных пользователей.

### 3.1.2 UberPool

UberPool – сервис совместных поездок UberPool, где водитель подбирает и высаживает каждого пассажира там, где тому удобно.

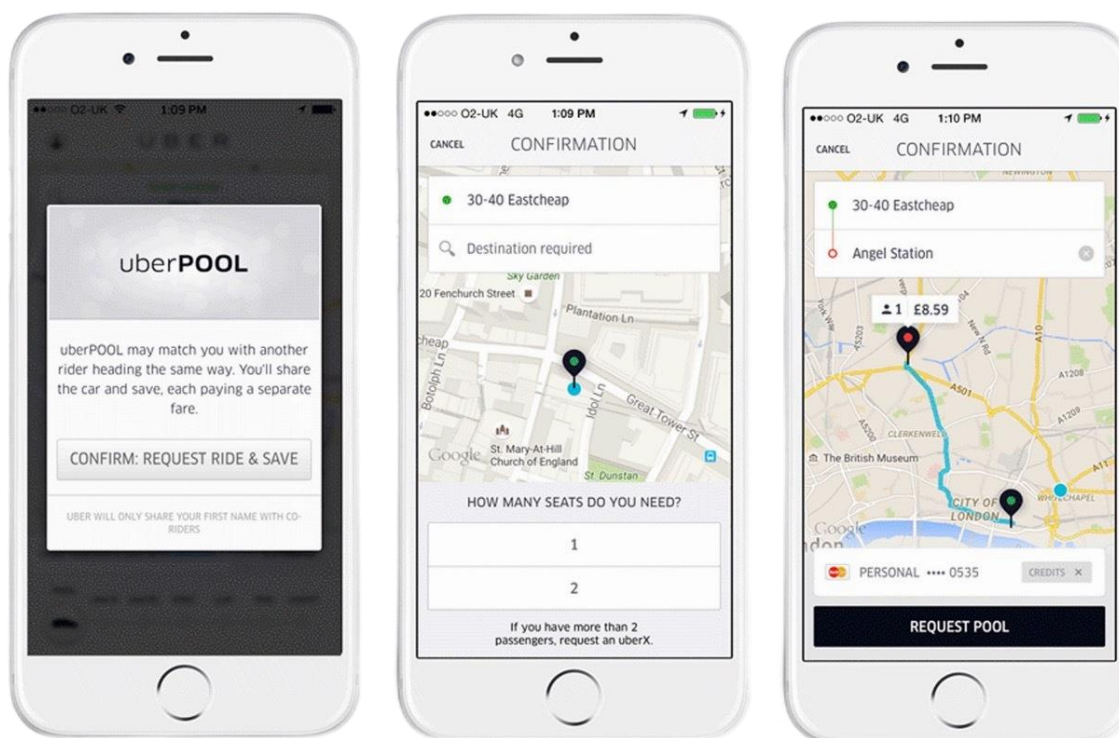


Рисунок 3 - Экраны поиска и построение маршрутов UberPool

Преимущества: Высокий уровень надежности и узнаваемость бренда Uber способствуют легкому взаимодействию с системой и доверию пользователей.

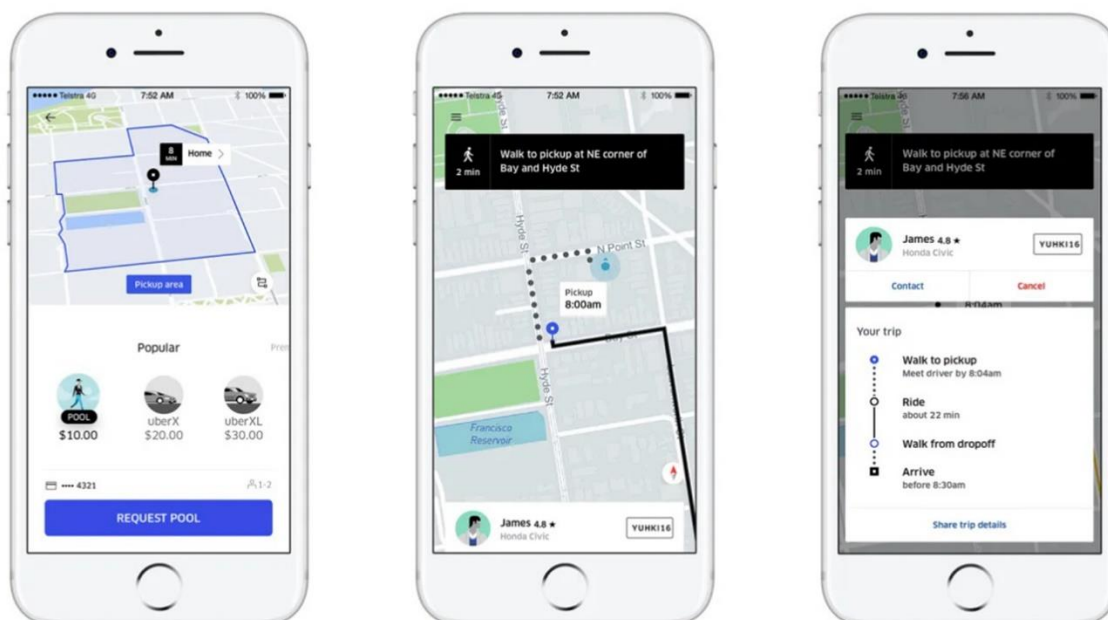


Рисунок 4 - Экраны процесса поездки UberPool

Недостатки: Ограниченная доступность и высокая цена из-за динамического ценообразования могут сделать услугу недоступной для широкого круга пользователей. Сервис недоступен на территории Российской Федерации.

### 3.1.3 CarpoolWorld

CarpoolWorld – предоставляет бесплатные глобальные услуги по подбору маршрутов для частных лиц, а также платные системы совместного использования автомобилей для организаций.

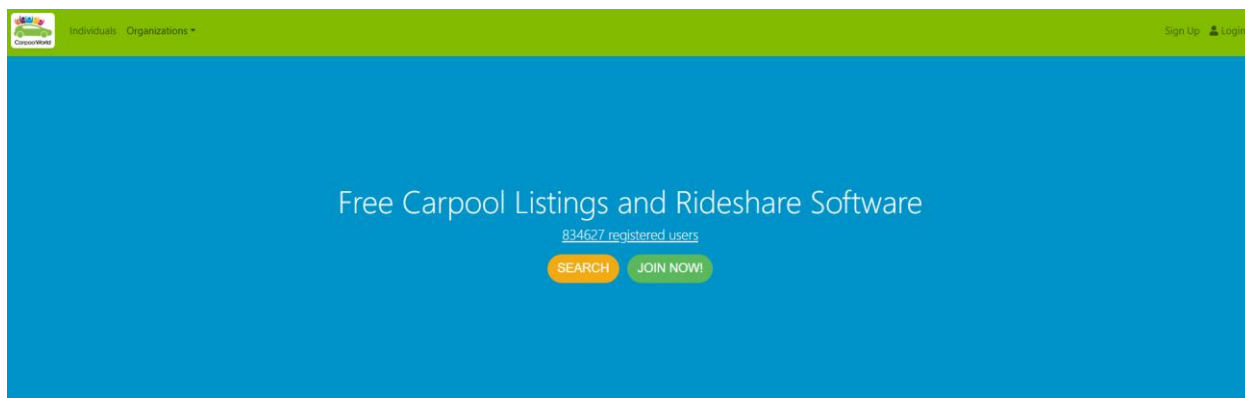


Рисунок 5 - Главный экран CarpoolWorld

Преимущества: Фокус на экологичных поездках и отсутствие платы за использование делают эту платформу привлекательной для экологически осознанных пользователей.

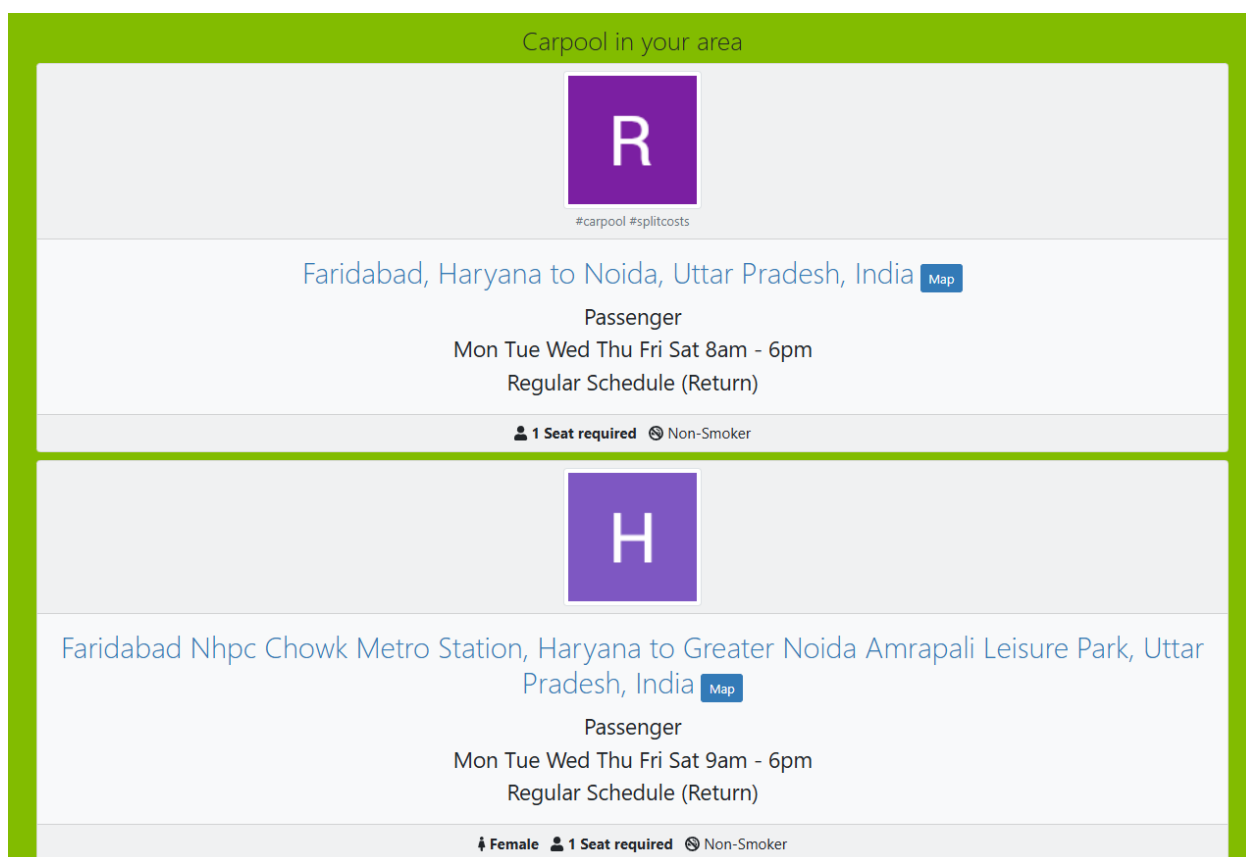


Рисунок 6 - Экран поиска поездок CarpoolWorld

Недостатки: Меньшая пользовательская база и ограниченный функционал приложения могут уменьшить его привлекательность. Сервис недоступен на территории Российской Федерации.

### 3.1.4 Едем.рф

Едем.рф – это российская платформа для поиска и бронирования попутчиков для поездок на автомобилях.



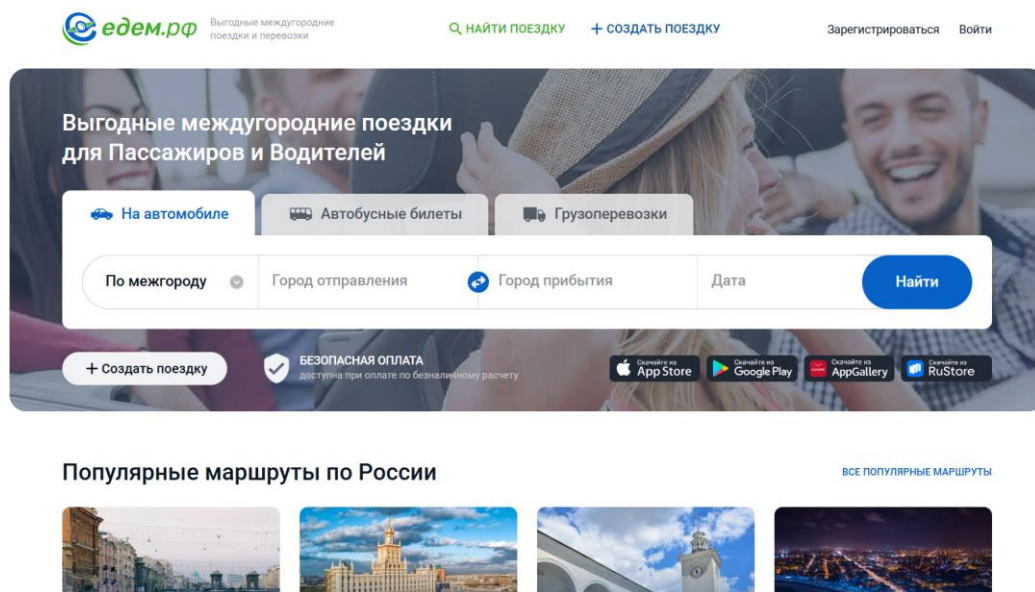


Рисунок 7 - Главный экран Едем.рф

Преимущества: Сервис предоставляет возможность пользователям совершать совместные поездки, которые будут снижать расходы и выбросы в атмосферу вредных веществ. Сервис прост в использовании, имеет отзывы и верификацию. Сервис позволяет обрести новые знакомства и предоставляет удобный выбор маршрутов.

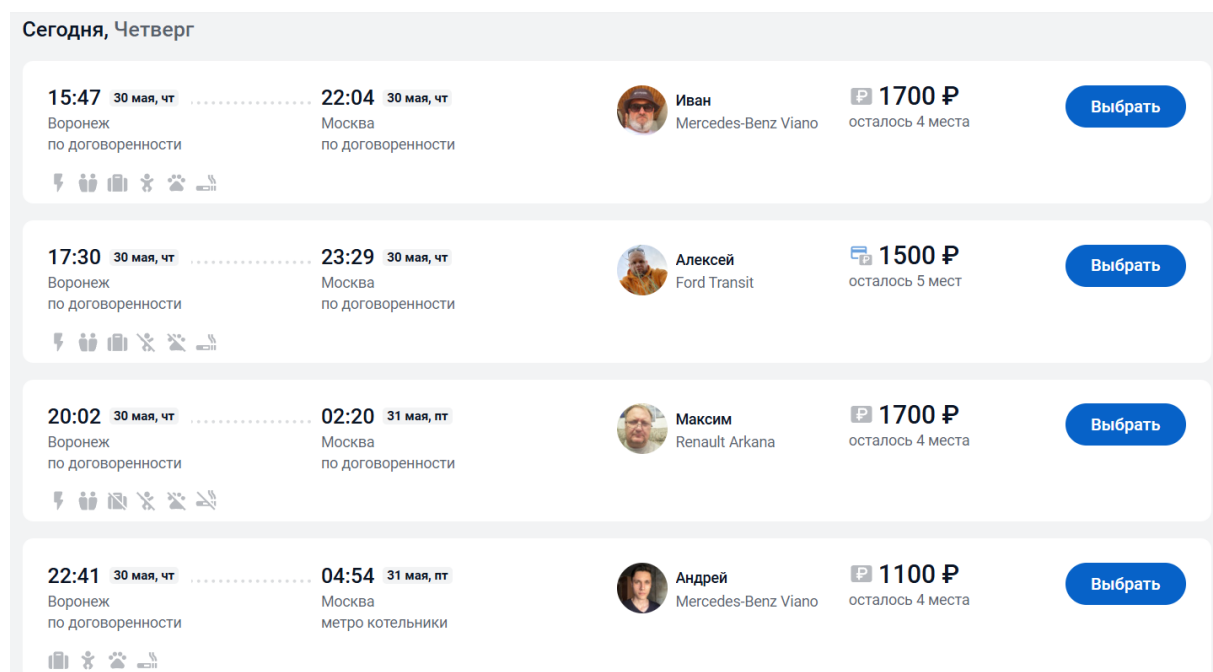


Рисунок 8 - Экран поиска поездок Едем.рф

Недостатки: Небольшое количество пользователей, риск мошенничества несмотря на верификацию, ограниченность маршрутов, недостаточное количество информации об автомобилях и возможные конфликты по дополнительным условиям поездки.

### **3.1.5 Автостоп**

Преимущества: Отсутствие необходимости оплаты за транспорт, возможность завести новые знакомства и обмена опытом с попутчиками, возможность обнаружить новые места и приключения, а также свобода выбора маршрута.

Недостатки: Потенциальные риски для личной безопасности, нет гарантии, что водитель найдётся быстро, ограниченный контроль над временем и маршрутом, а также возможные неудобства в пути, такие как длительное ожидание.

## **3.2 Моделирование системы**

### **3.2.1 Диаграмма состояний**

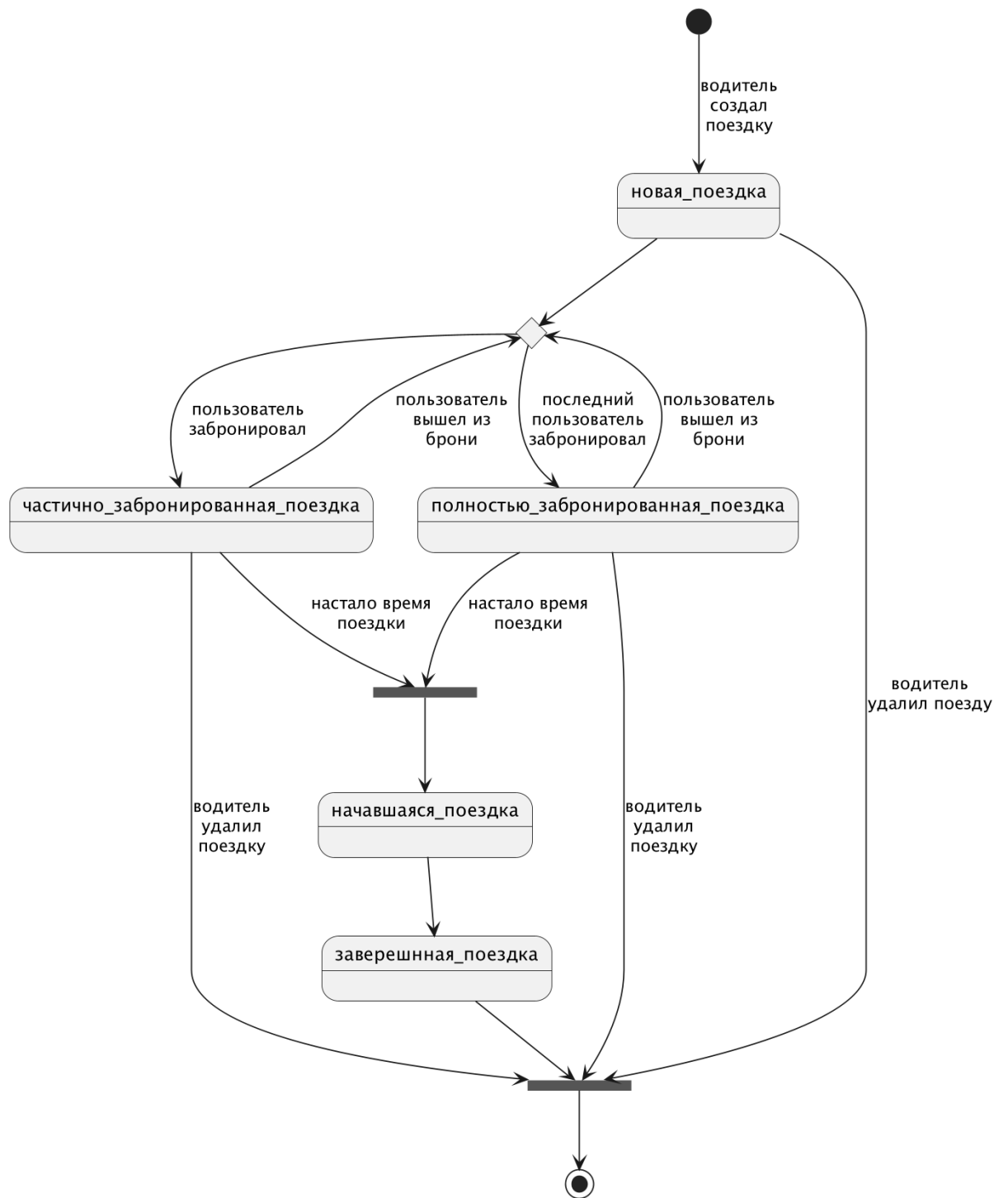


Рисунок 9 - Диаграмма состояния поездки

### 3.2.2 Диаграмма последовательности

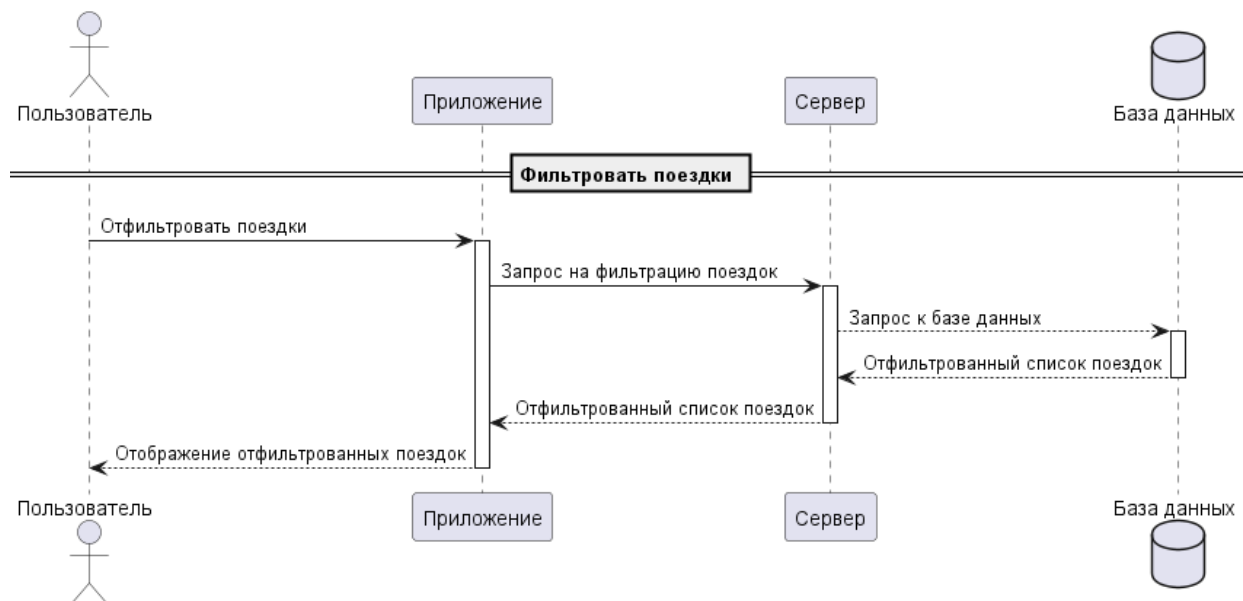


Рисунок 10 - Диаграмма воздействия для авторизованного пользователя  
(часть 1 из 17)

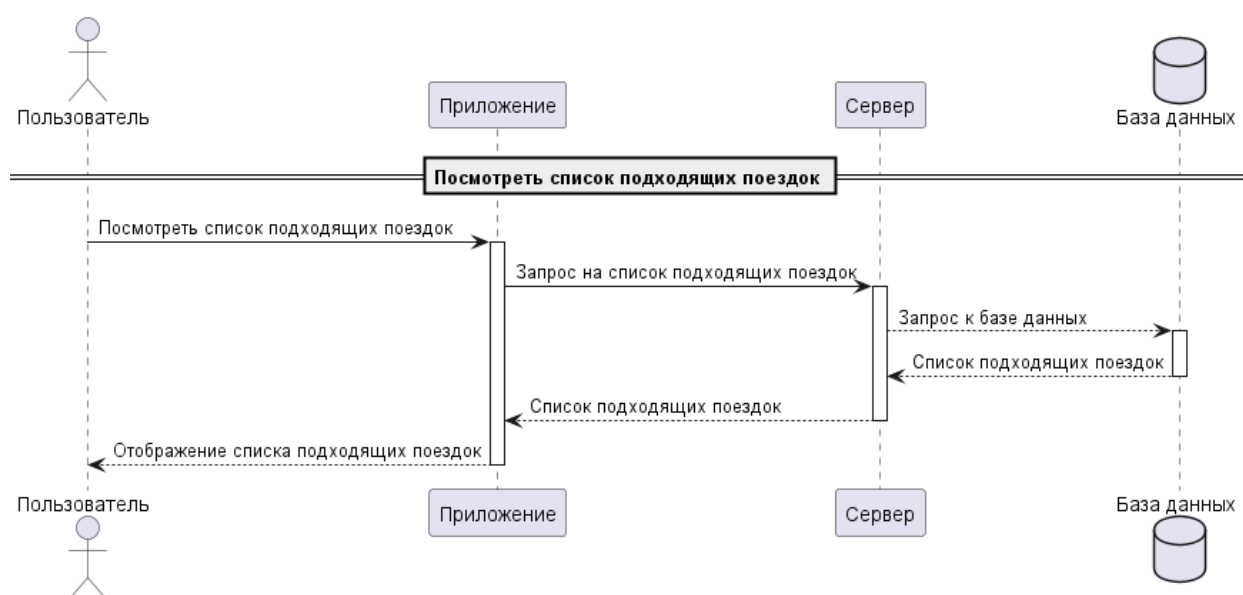


Рисунок 11 - Диаграмма воздействия для авторизованного пользователя  
(часть 2 из 17)

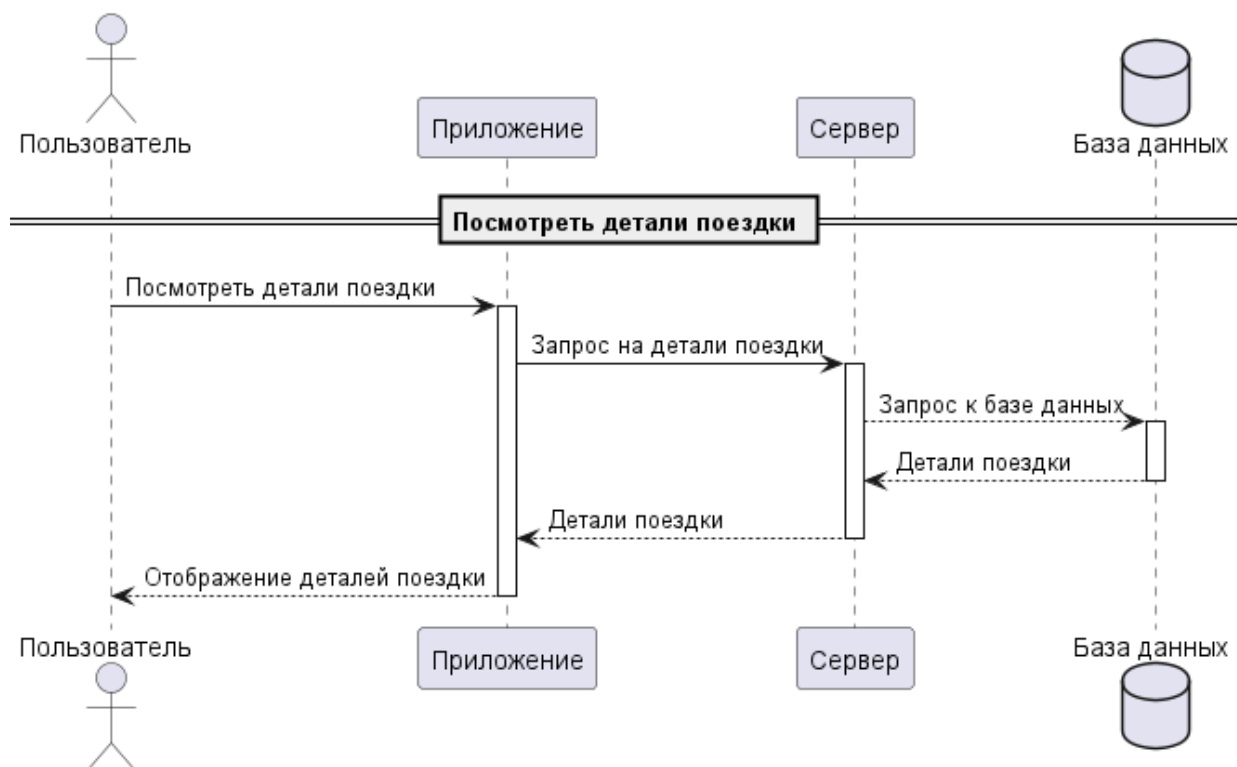


Рисунок 12 - Диаграмма воздействия для авторизованного пользователя (часть 3 из 17)

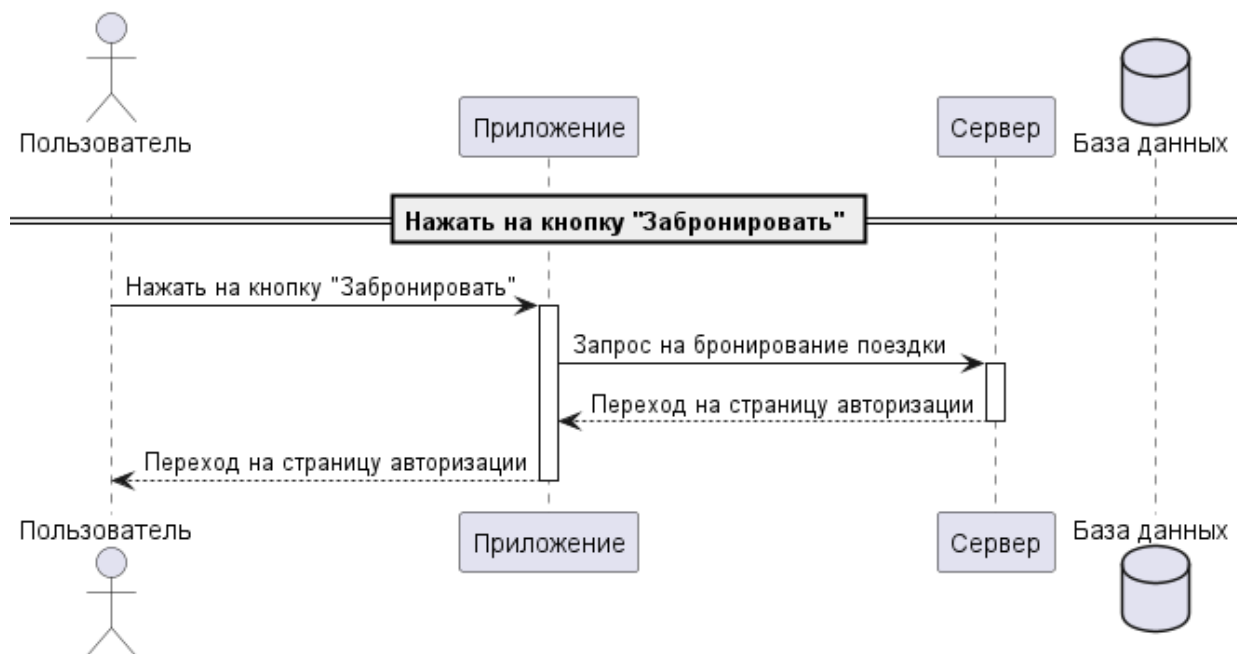


Рисунок 13 - Диаграмма воздействия для авторизованного пользователя (часть 4 из 17)

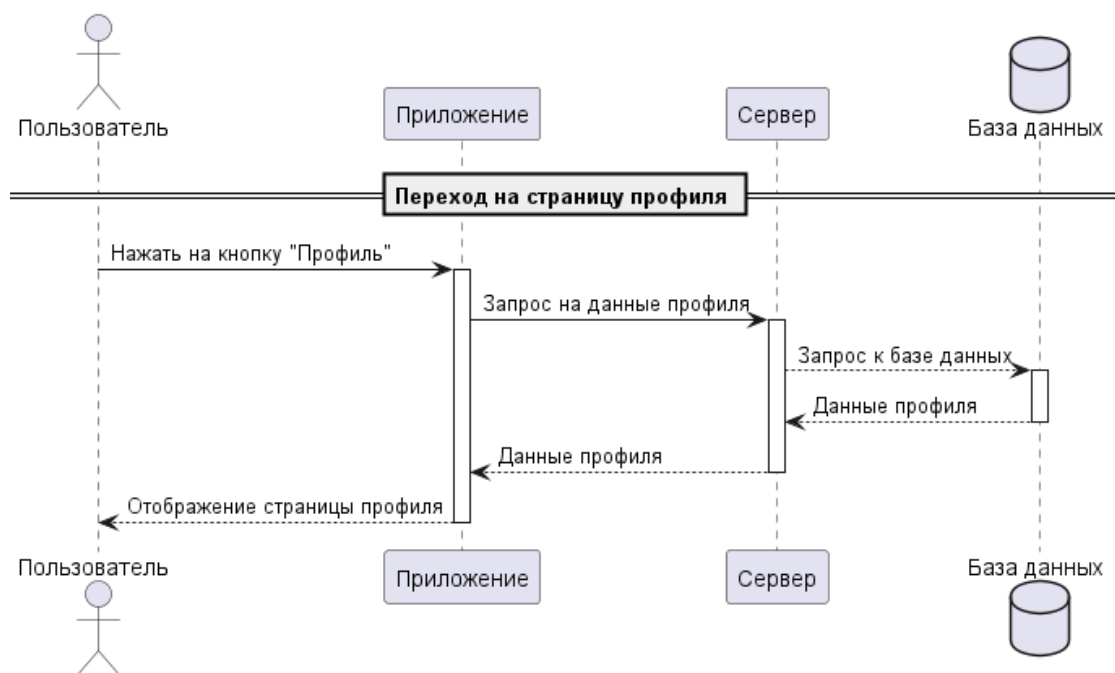


Рисунок 14 - Диаграмма воздействия для авторизованного пользователя  
(часть 5 из 17)

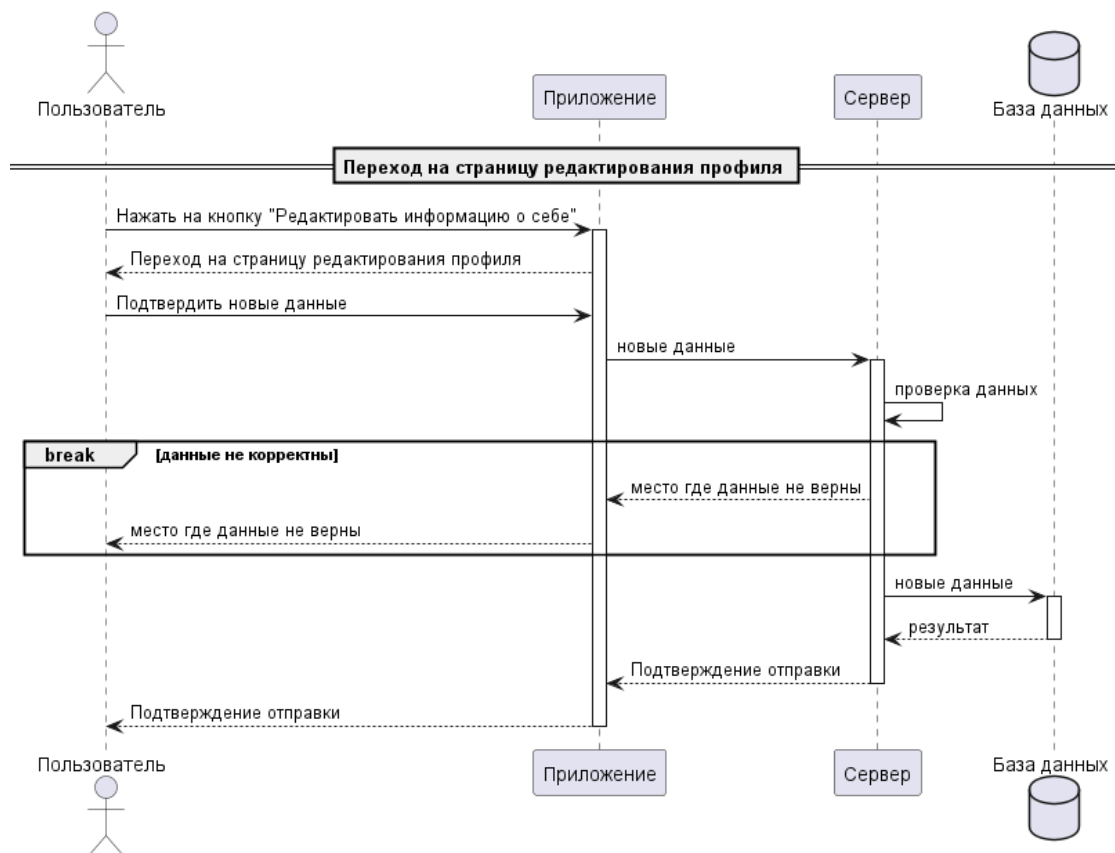


Рисунок 15 - Диаграмма воздействия для авторизованного пользователя  
(часть 6 из 17)

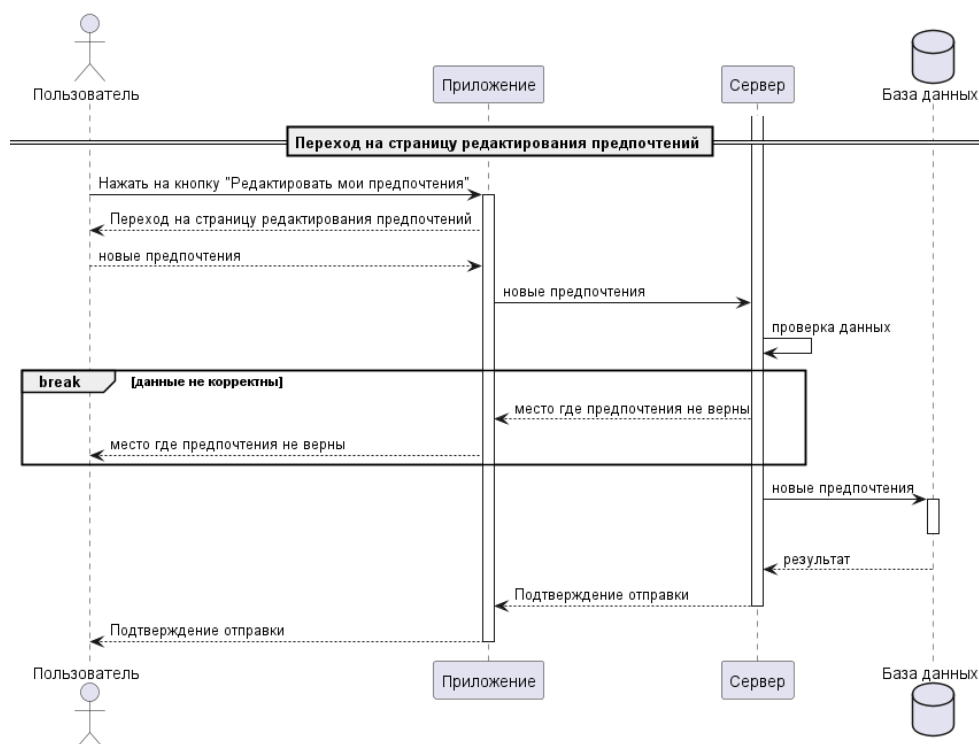


Рисунок 16 - Диаграмма воздействия для авторизованного пользователя  
(часть 7 из 17)

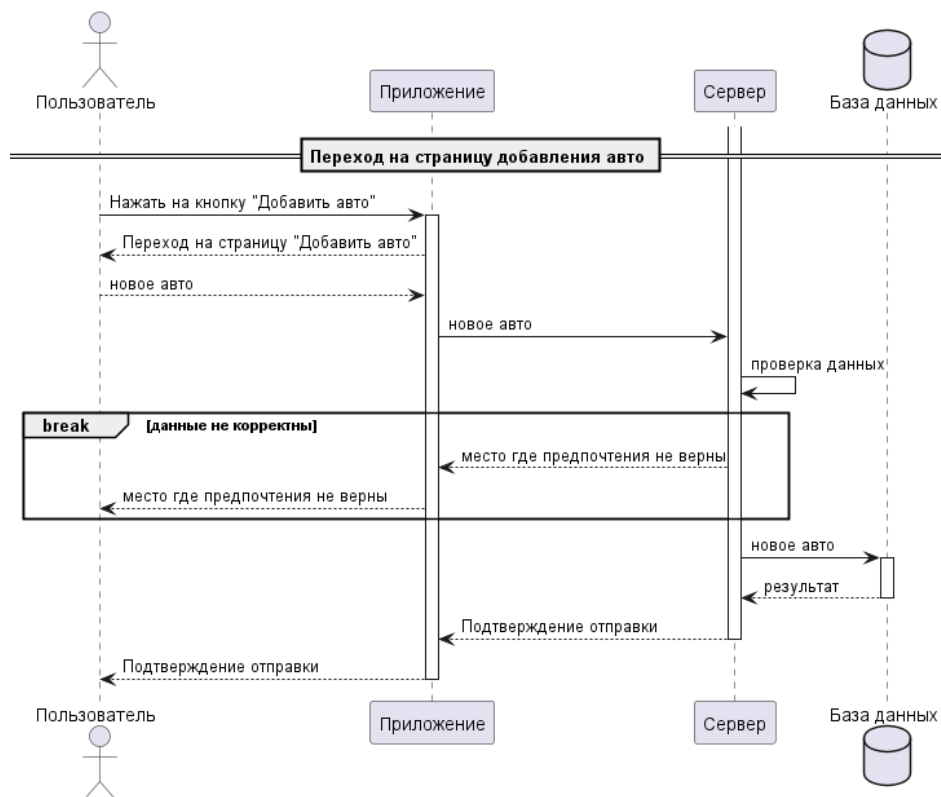


Рисунок 17 - Диаграмма воздействия для авторизованного пользователя  
(часть 8 из 17)

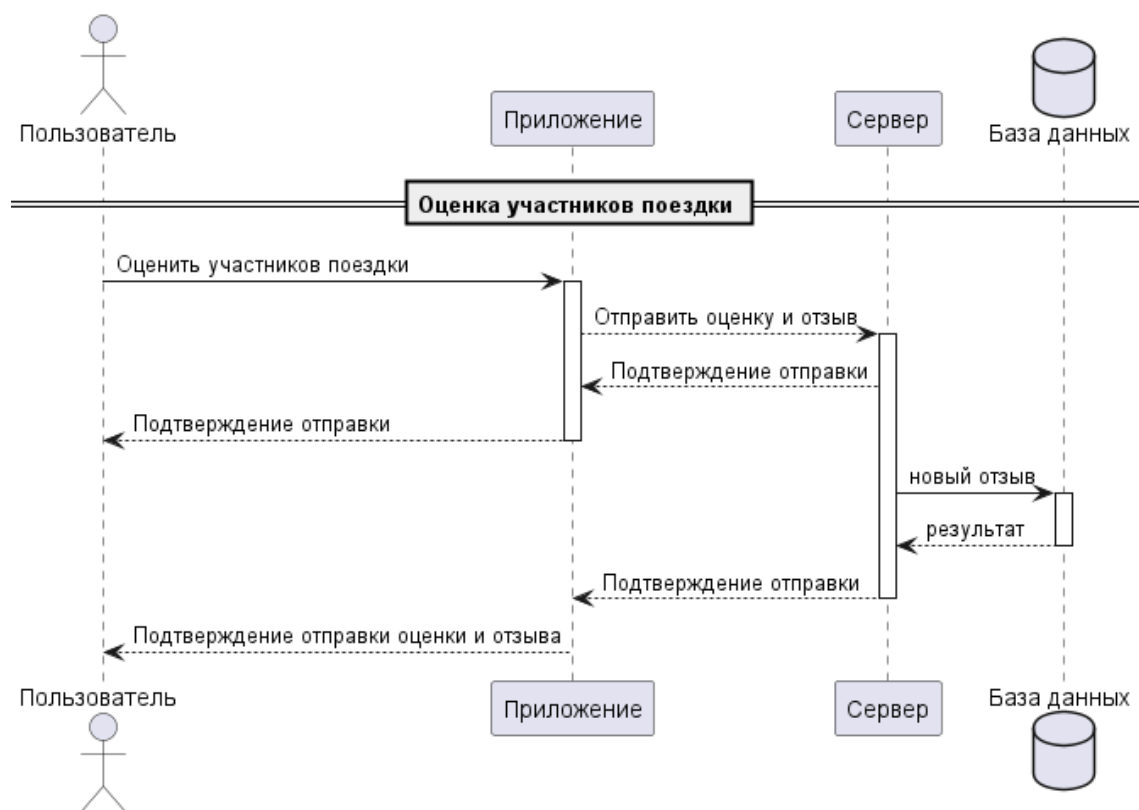


Рисунок 18 - Диаграмма воздействия для авторизованного пользователя (часть 9 из 17)

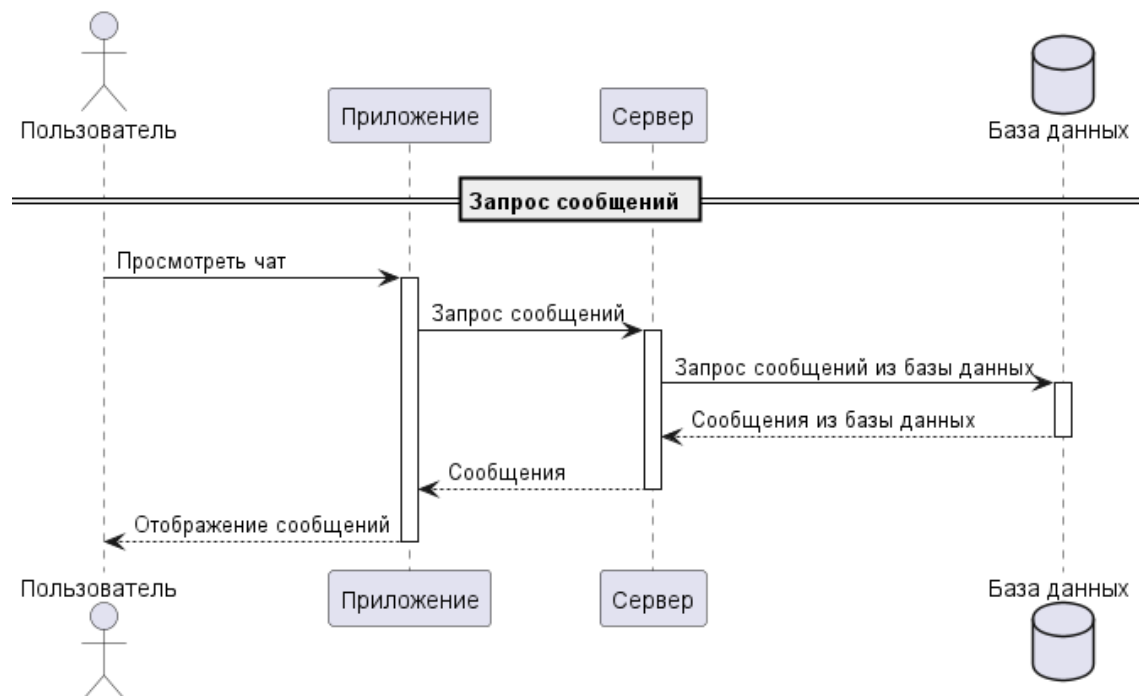


Рисунок 19 - Диаграмма воздействия для авторизованного пользователя (часть 10 из 17)





Рисунок 20 - Диаграмма воздействия для авторизованного пользователя  
(часть 11 из 17)

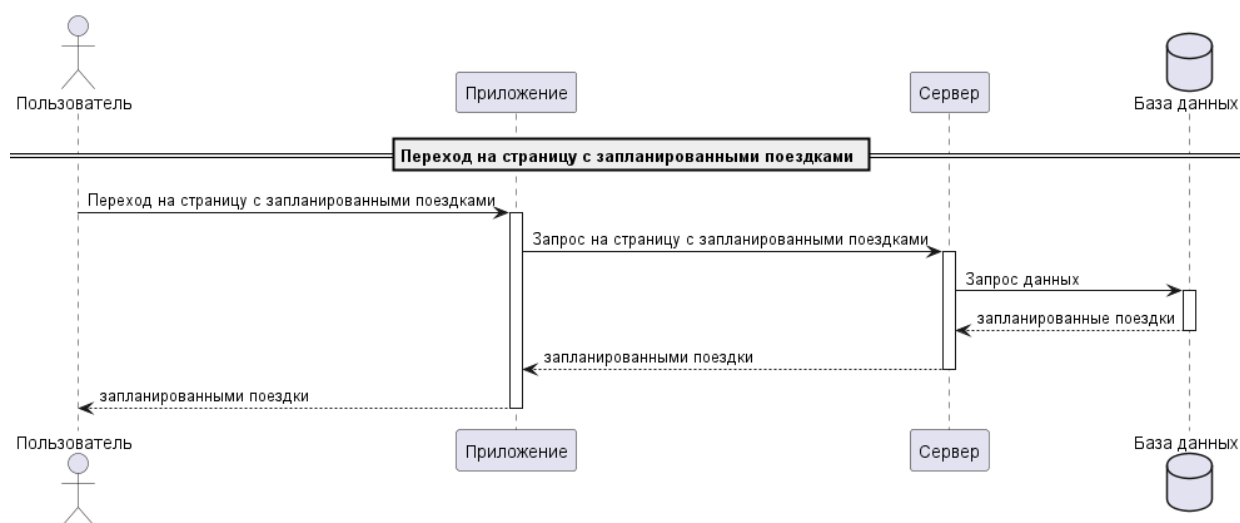


Рисунок 21 - Диаграмма воздействия для авторизованного пользователя  
(часть 12 из 17)

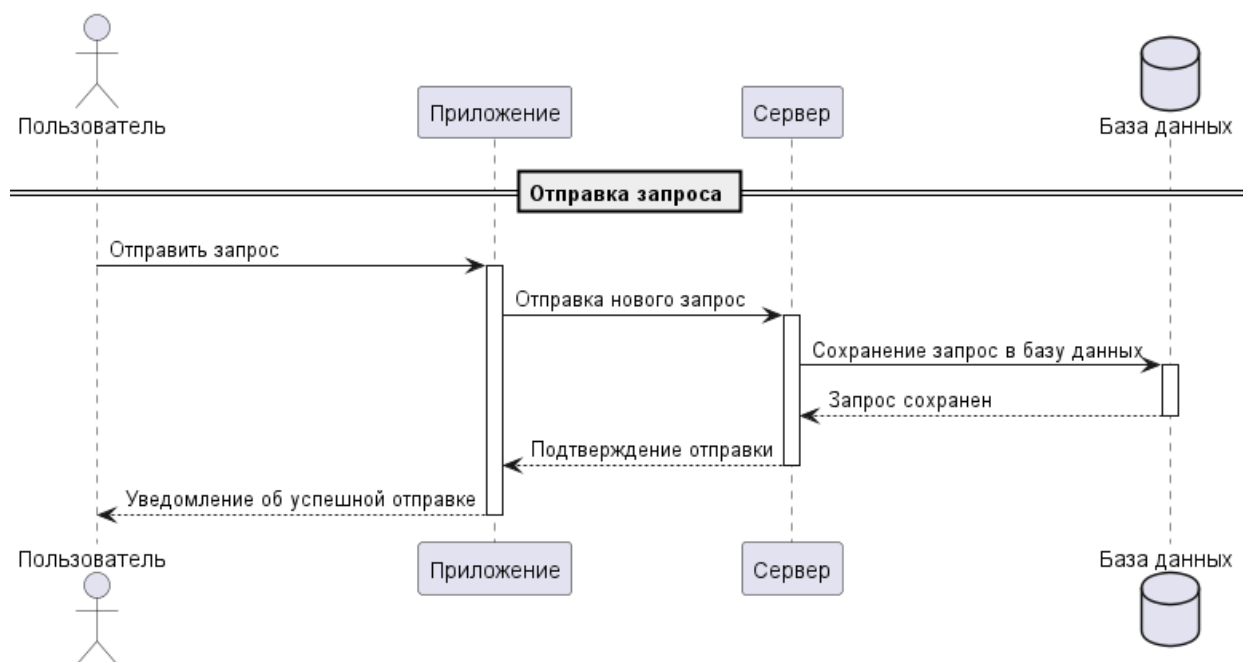


Рисунок 22 - Диаграмма воздействия для авторизованного пользователя (часть 13 из 17)

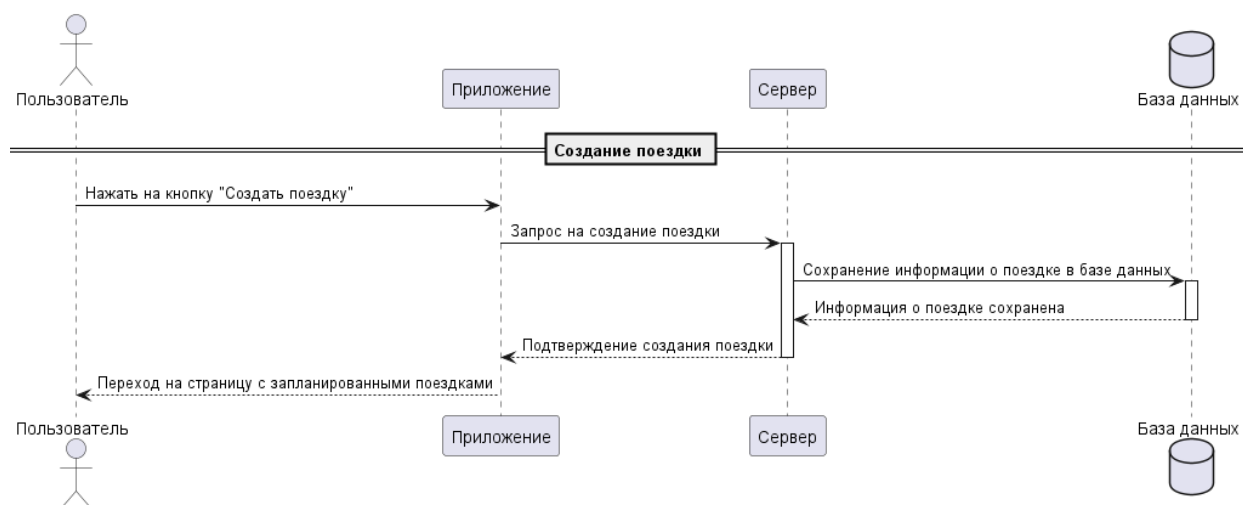


Рисунок 23 - Диаграмма воздействия для авторизованного пользователя (часть 14 из 17)

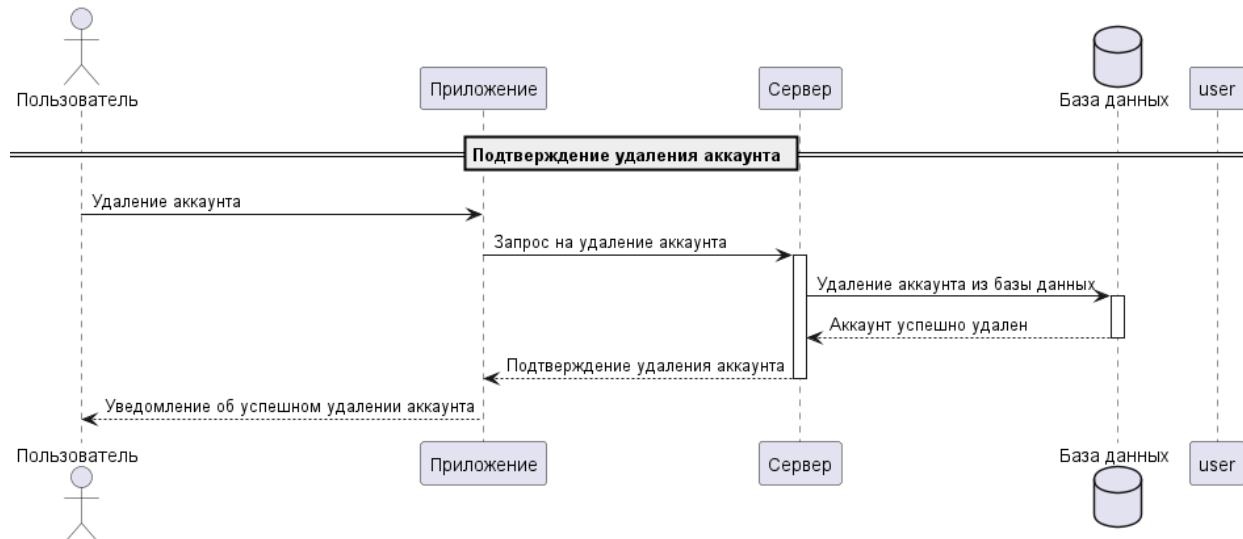


Рисунок 24 - Диаграмма воздействия для авторизованного пользователя  
(часть 15 из 17)

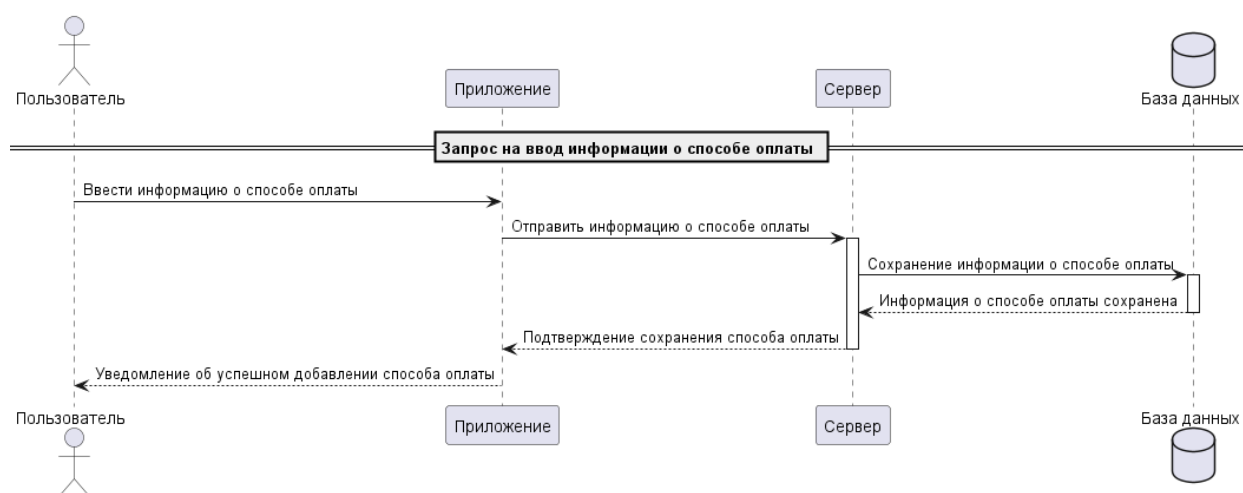


Рисунок 25 - Диаграмма воздействия для авторизованного пользователя  
(часть 16 из 17)

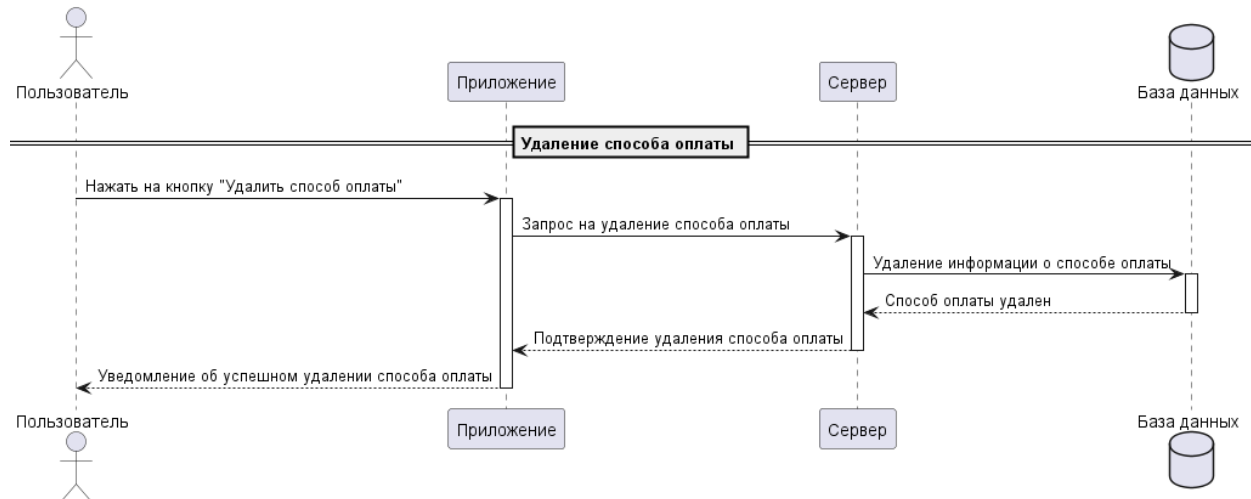


Рисунок 26 - Диаграмма воздействия для авторизованного пользователя  
(часть 17 из 17)

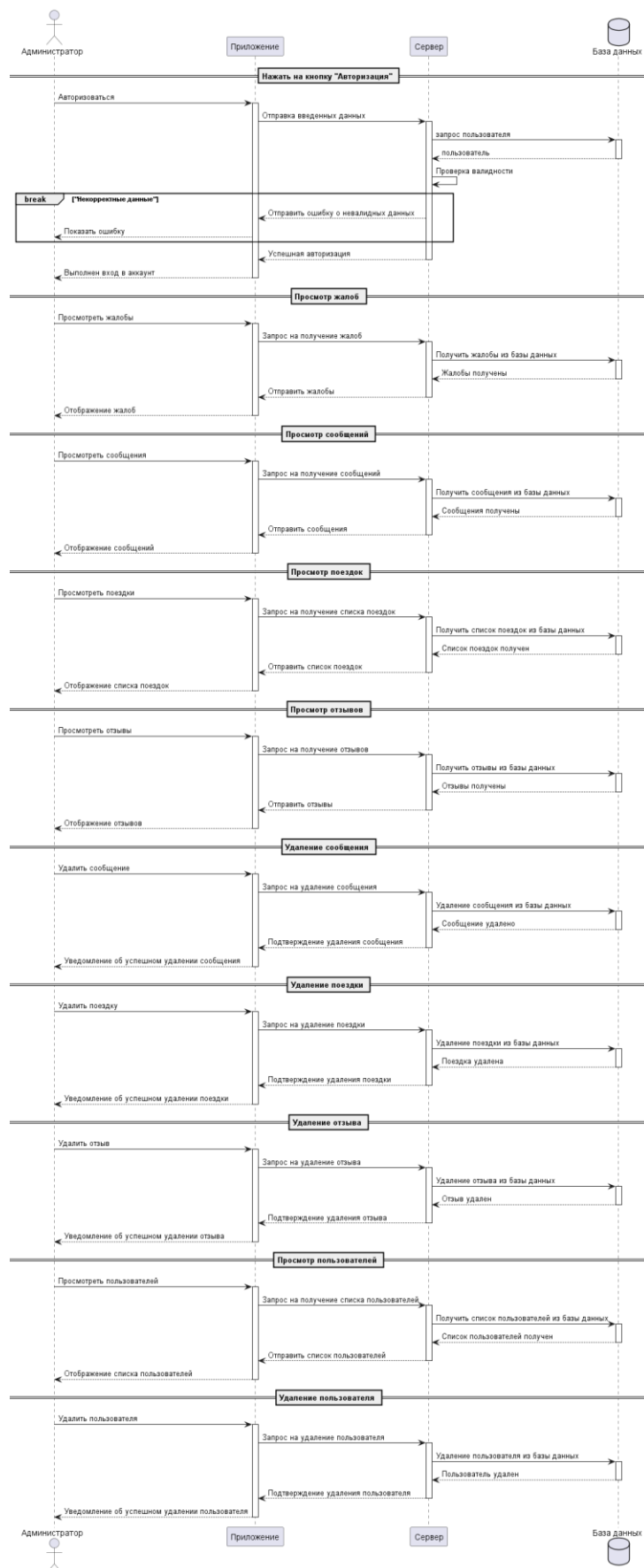


Рисунок 27 - Диаграмма воздействия для администратора

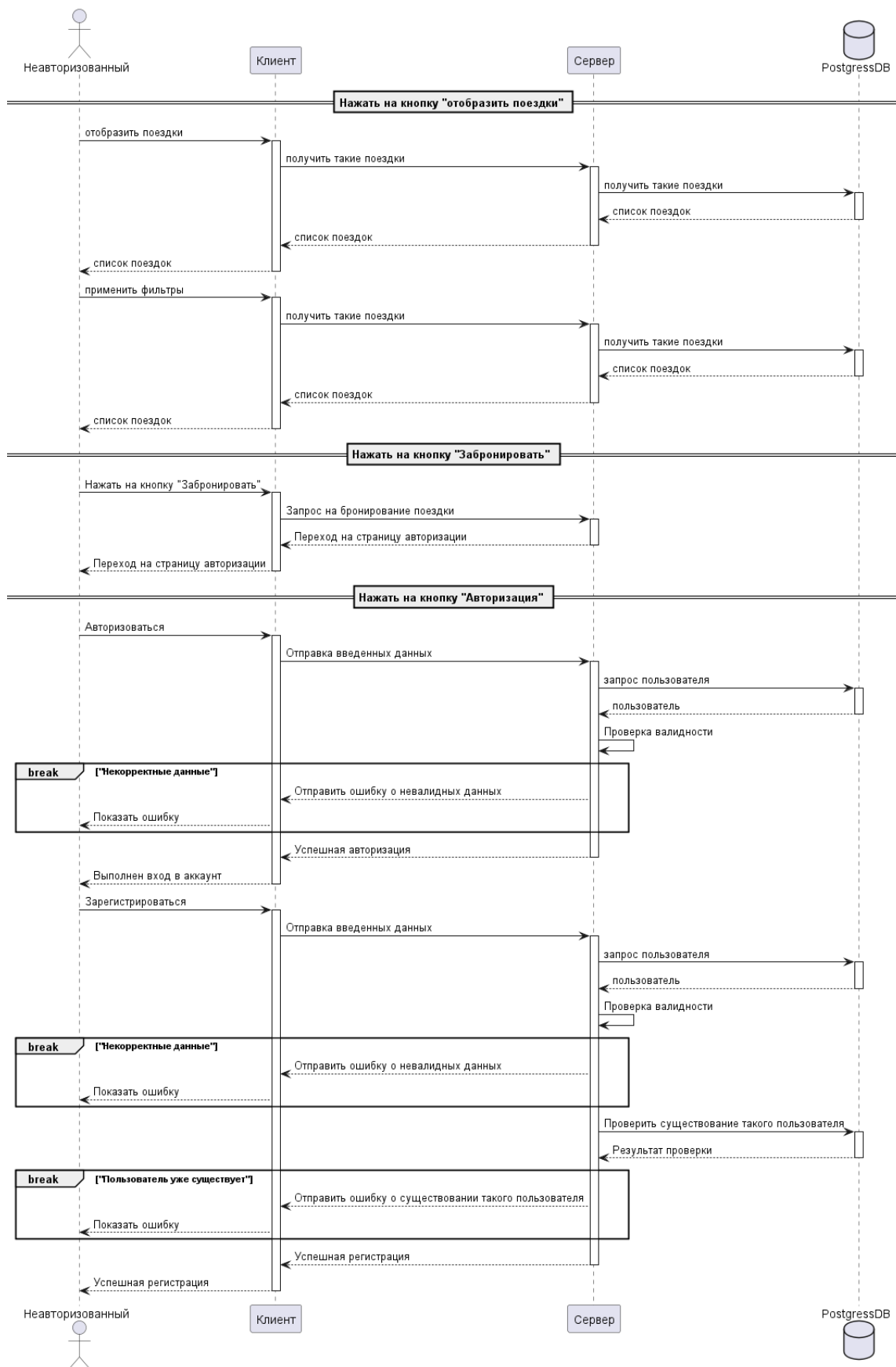


Рисунок 28 - Диаграмма воздействия для неавторизованного пользователя

### 3.3 Аналитика веб-приложения

В качестве системы для сбора данных об использовании приложения пользователями выступает сервис «Яндекс.Метрика», так как позволяет наиболее быстро и легко отслеживать метрики для мобильных приложений, обладает понятным интерфейсом и доступным руководством по использованию.

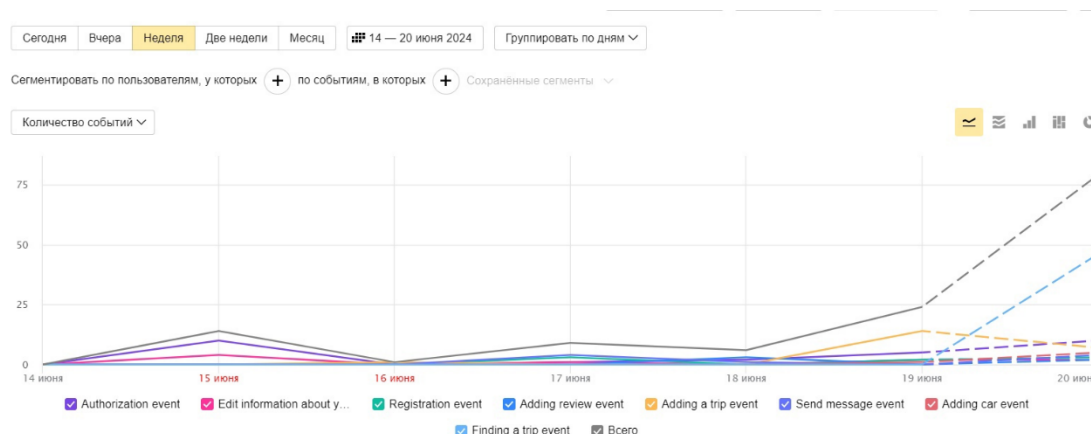


Рисунок 29 - События из Яндекс.Метрики

### 3.4 Границы проекта

Приложение должно предоставлять пользователям следующие возможности:

- поиск подходящих попутчиков для водителей и пассажиров;
- поиск подходящей поездки для пассажиров;
- управление графиком поездок;
- предварительное планирование поездки;
- оценка надежности и безопасности пользователей;
- расчет стоимости поездки;
- гарантированное получение оплаты за поездку водителем;
- гарантированный возврат средств пассажирам за несостоявшиеся поездки.

## 4 Реализация

### 4.1 Средства реализации

Приложение соответствует клиент-серверной архитектуре. Клиент отправляет запросы на серверную часть. REST API возвращает клиенту необходимые данные от сервера.

Реализована база данных для хранения личных данных пользователей, платежных данных их карт, логинов и паролей, а также данных о поездках.

Для реализации серверной части были выбраны средства:

- язык программирования Python 3.9.5;
- фреймворк FastAPI;
- библиотека SQLAlchemy для взаимодействия с базой данных;
- библиотека passlib для шифрования паролей данных карт пользователей;
- система PostgreSQL 14 для управления базами данных;
- ОС Ubuntu 22.04;
- система контроля версий Git 2.40.1.

Для реализации серверной части был выбран фреймворк FastAPI и язык Python, потому что они имеют ряд преимуществ:

- Благодаря асинхронным вызовам и эффективной архитектуре, FastAPI обеспечивает одну из самых высоких скоростей обработки запросов среди Python-фреймворков;
- Python — мощный и гибкий язык программирования, который предоставляет широкие возможности для разработки веб-приложений;
- Автоматическая генерация документации и поддержка аннотаций типов в FastAPI упрощают разработку и тестирование;
- Python имеет множество библиотек с готовыми решениями для обработки данных, изображений и других задач, что упрощает и ускоряет разработку.



Для реализации клиентской части были выбраны средства:

- Android Studio;
- язык программирования Kotlin 1.9.20;
- библиотека Retrofit 2.9.0 для работы с сетевыми запросами;
- система контроля версий Git 2.40.1.

Язык программирования Kotlin был выбран, потому что:

- Kotlin совместим с Java, что дает возможность использовать существующие Java-библиотеки и легко интегрироваться с Java-проектами;
- Kotlin является официальным языком программирования для разработки Android-приложений, поэтому он идеально подходит для приложений, которые должны быть оптимизированы под платформу Android и использовать все ее возможности;
- Kotlin предоставляет прямой доступ к Android SDK и нативным API, что позволяет повысить производительность приложения;
- Kotlin Multiplatform поддерживает кроссплатформенную разработку, что позволяет поддерживать несколько платформ.

## **4.2 Реализация базы данных**

ER-диаграмма определяет структуру базы данных, отображая сущности и их взаимосвязи.

Диаграмма отношений сущностей представлена на рисунке ниже.

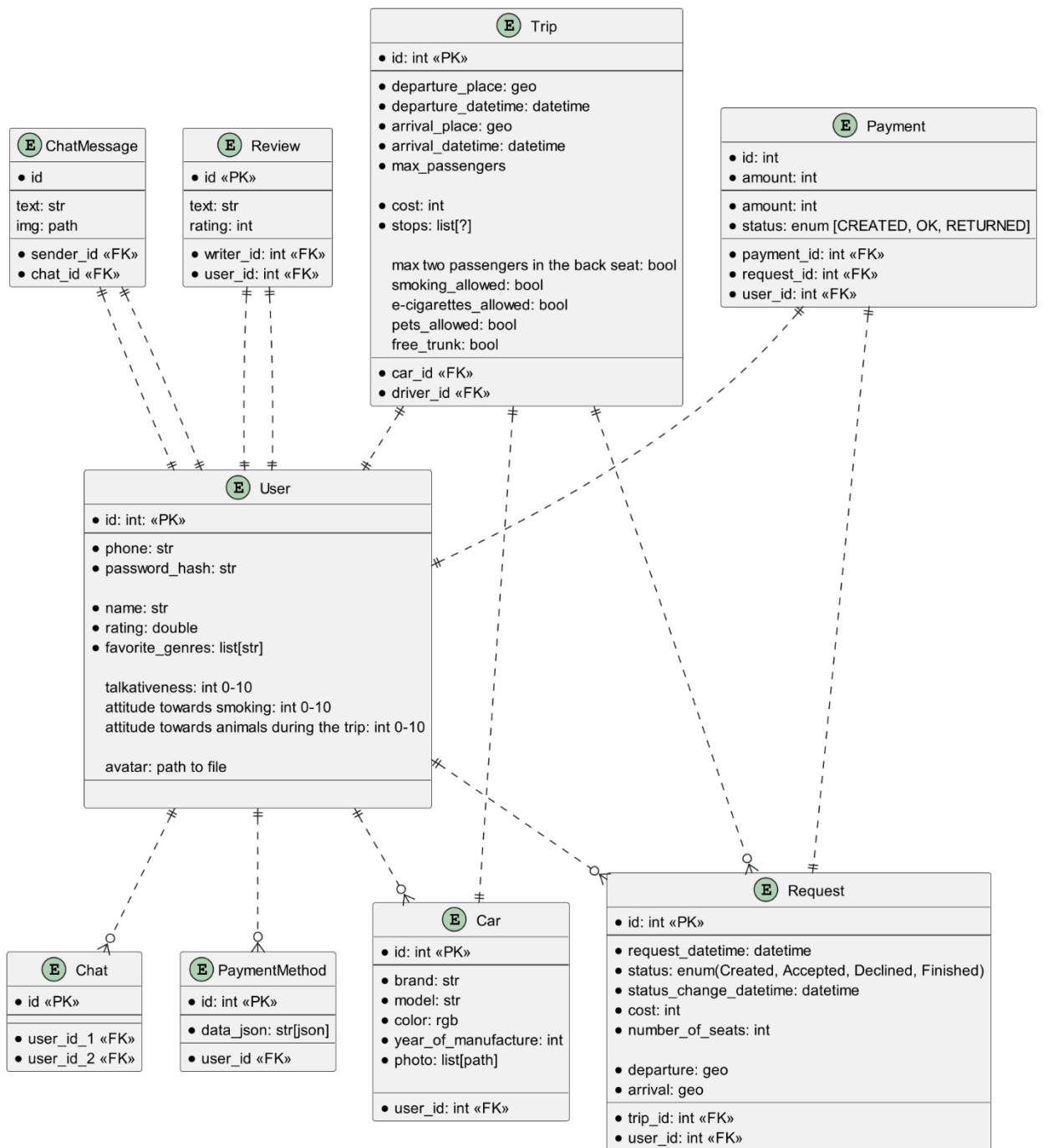


Рисунок 30 - ER диаграмма

База данных для нашего приложения содержит 9 таблиц.

User (Пользователь) - хранит информацию о пользователях системы, таких как имя, телефон, рейтинг и предпочтения.

Trip (Поездка) - содержит данные о поездках, включая места отправления и прибытия, время, стоимость и условия поездки.

Car (Автомобиль) - информация об автомобилях пользователей, таких как марка, модель и год выпуска.

Request (Запрос на поездку) - запросы пользователей на участие в поездках, включая статус запроса и количество мест.

Payment (Платеж) - данные о платежах за поездки, такие как сумма и статус платежа.

Review (Отзыв) - отзывы пользователей о поездках и водителях, включая оценку и комментарии.

ChatMessage (Сообщение в чате) - Сообщения, отправленные пользователями в чате.

Chat (Чат) - Информация о чатах между пользователями.

PaymentMethod (Способ оплаты) - Способы оплаты, привязанные к пользователям.

### **4.3 Реализация серверной части приложения**

Помимо базы данных, серверная часть состоит из следующих компонентов:

- .env\_ex - пример файла конфигурации среды, где хранятся переменные окружения;
- .gitignore - список файлов и директорий, которые должны игнорироваться системой контроля версий Git;
- Dockerfile - инструкции для сборки Docker-образа;
- README.md - описание проекта и инструкции по его использованию;
- alembic.ini - конфигурационный файл для Alembic, инструмента миграции базы данных;
- docker-compose.yaml - конфигурация для Docker Compose, используемая для управления многоконтейнерными приложениями Docker;

- requirements.txt - список зависимостей Python, необходимых для работы проекта;
- run.sh - скрипт для запуска приложения;
- split.py - скрипт, скорее всего, используемый для разделения данных или какой-либо другой обработки;
- .idea/ - директория с настройками проекта для IDE (например, PyCharm);
- alembic/ - директория для файлов миграций Alembic;
- alembic/env.py - основной скрипт для настройки среды миграции;
- alembic/script.py.mako - шаблон для генерации новых миграций;
- alembic/versions/ - директория с файлами миграций;
- app/ - основная директория приложения;
- app/\_\_init\_\_.py - инициализация пакета;
- app/config.py - конфигурация приложения;
- app/dependencies.py - зависимости приложения;
- app/main.py - главный скрипт приложения;
- app/database/database.py - настройка и подключение к базе данных;
- app/database/models.py - описание моделей базы данных;
- \_\_init\_\_.py - инициализация модуля мест;
- crud.py - функции для создания, чтения, обновления и удаления мест;
- router.py - определение маршрутов для операций с местами;
- schemas.py - схемы данных для запросов и ответов, связанных с местами;
- \_\_init\_\_.py - инициализация модуля запросов;
- controller.py - логика контроллеров для обработки запросов;
- crud.py - CRUD-операции для запросов;
- router.py - маршруты для операций с запросами;
- schemas.py - схемы данных для запросов и ответов;

- `calc/` - поддиректория для калькуляторов;
- `__init__.py` - инициализация калькуляторов;
- `cost_calc.py` - логика расчета стоимости;
- `dist_calc.py` - логика расчета дистанций;
- `validator_stops.py` - валидаторы остановок;
- `__init__.py` - инициализация модуля отзывов;
- `crud.py` - CRUD-операции для отзывов;
- `router.py` - маршруты для операций с отзывами;
- `schemas.py` - схемы данных для отзывов;
- `__init__.py` - инициализация модуля поездок;
- `comfrots_schemas.py` - схемы данных для удобств в поездках;
- `crud.py` - CRUD-операции для поездок;
- `router.py` - маршруты для операций с поездками;
- `schemas.py` - схемы данных для поездок;
- `stop_crud.py` - CRUD-операции для остановок в поездках;
- `__init__.py` - инициализация модуля пользователей;
- `crud.py` - CRUD-операции для пользователей;
- `router.py` - маршруты для операций с пользователями;
- `schemas.py` - схемы данных для пользователей;
- `security.py` - логика безопасности и аутентификации;
- `app/entities/enums.py` - определение перечислений;
- `app/entities/exceptions.py` - обработка исключений;
- `places` - операции с местами;
- `requests` - обработка запросов;
- `review` - управление отзывами;
- `trip` - операции с поездками;
- `user` - управление пользователями;
- `__init__.py` - инициализирует модуль предложений, указывая Python, что эта директория является пакетом;

- `controller.py` - содержит логику контроллеров для обработки и управления предложениями. Этот файл может включать функции для получения предложений на основе пользовательского ввода и другие вспомогательные методы;
- `router.py` - определяет маршруты для эндпоинтов предложений. Этот файл регистрирует маршруты, которые будут использоваться для взаимодействия с предложениями через HTTP-запросы;
- `yandex/` - поддиректория, в которой реализована интеграция с Yandex API для получения предложений и автодополнений;
- `__init__.py` - инициализирует подмодуль Yandex, указывая Python, что эта директория является частью пакета;
- `exceptions.py` - определяет исключения, специфичные для работы с Yandex API. Например, исключения могут быть выброшены при ошибках сети, неверных ответах API или других проблемах, связанных с Yandex;
- `geo_coder.py` - реализует функциональность геокодирования с использованием Yandex API. Геокодирование позволяет преобразовывать координаты в адреса и наоборот;
- `schemas.py` - определяет схемы данных для запросов и ответов, связанных с Yandex API. Схемы используются для валидации данных, получаемых и отправляемых через API;
- `suggest.py` - реализует функциональность предложений и автодополнений с использованием Yandex API. Этот файл может содержать методы для получения предложений на основе частичного ввода пользователя.

## **4.4 Реализация клиентской части приложения**

### **4.4.1 Общая информация**

Клиентская часть реализует пользовательский интерфейс, формирует запросы к серверу и обрабатывает ответы от него.

Клиентская часть построена на принципах многомодульной архитектуры и состоит из:

- `api` модуля, отвечающего за взаимодействие с серверной частью приложения и содержащего DTO классы и сервисы по работе с `api`;
- `data_store` модуля, содержащего классы, которые отвечают за локальное хранилище данных (на устройстве). Они предназначены для того, чтобы на устройстве запоминался токен авторизации пользователя и ему не приходилось вводить пароль при каждом входе в приложение;
- `models` модуля, содержащего классы объектов, используются для отрисовки интерфейса;
- `screens` модуля, содержащего реализацию всего пользовательского интерфейса (всех экранов приложения);
- `services` модуля, содержащего бизнес-логику клиентской части приложения.

### **4.4.2 Графический интерфейс**

Приложение включает в себя:

- Приветственный экран для первого запуска;
- Экран поиска поездок;
- Экран с результатом поиска поездок;
- Экран деталей поездки;

- Экран проверки деталей поездки;
- Экран создания поездки;
- Экран авторизации;
- Экран регистрации;
- Экран профиля;
- Экран редактирования информации о себе;
- Экран редактирования предпочтений;
- Экран профиля другого пользователя;
- Экран всех чатов;
- Экран чата;
- Экран подтверждения бронирования;
- Экран редактирования поездки;
- Экран всех запланированных поездок;
- Экран настроек аккаунта;
- Экран оплаты;
- Экран просмотра отзывов;
- Экран создания отзыва;
- Экран добавления автомобиля.
- Экран с информацией о приложении.

Далее будут описаны основные экраны приложения:



#### 4.4.2.1 Экран авторизации

Страница авторизации позволяет неавторизованным пользователям получить доступ к своему аккаунту или создать новый. Реализованный экран авторизации представлен на рисунке ниже.

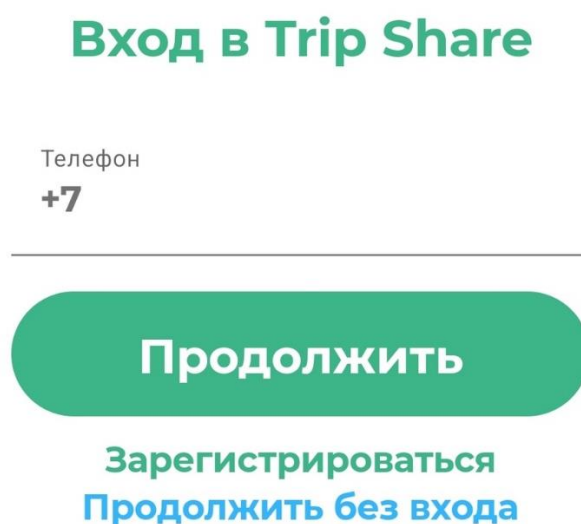


Рисунок 31 - Экран авторизации

#### 4.4.2.2 Экран профиля

Экран профиля позволяет авторизованным пользователям ознакомиться с информацией о себе и управлять своим аккаунтом. Реализованный экран профиля представлен на рисунке ниже.

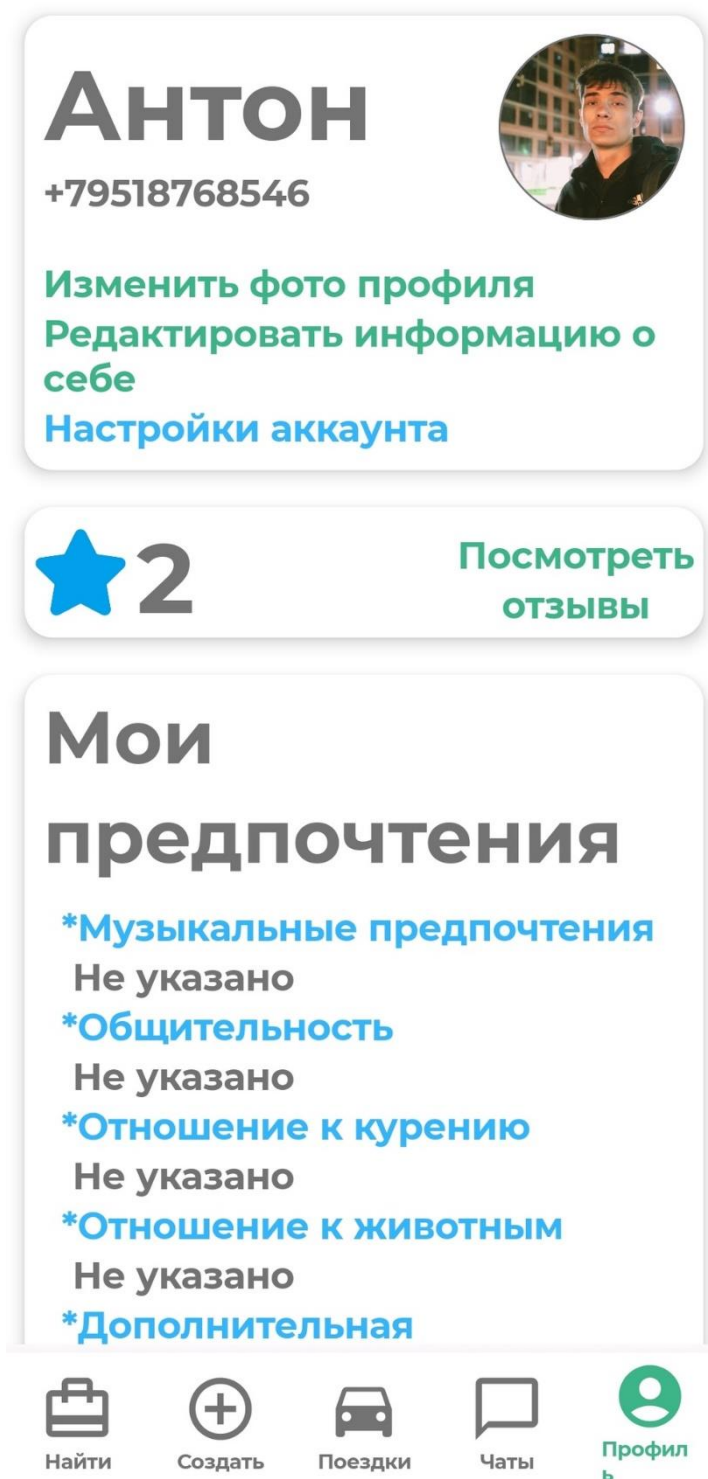
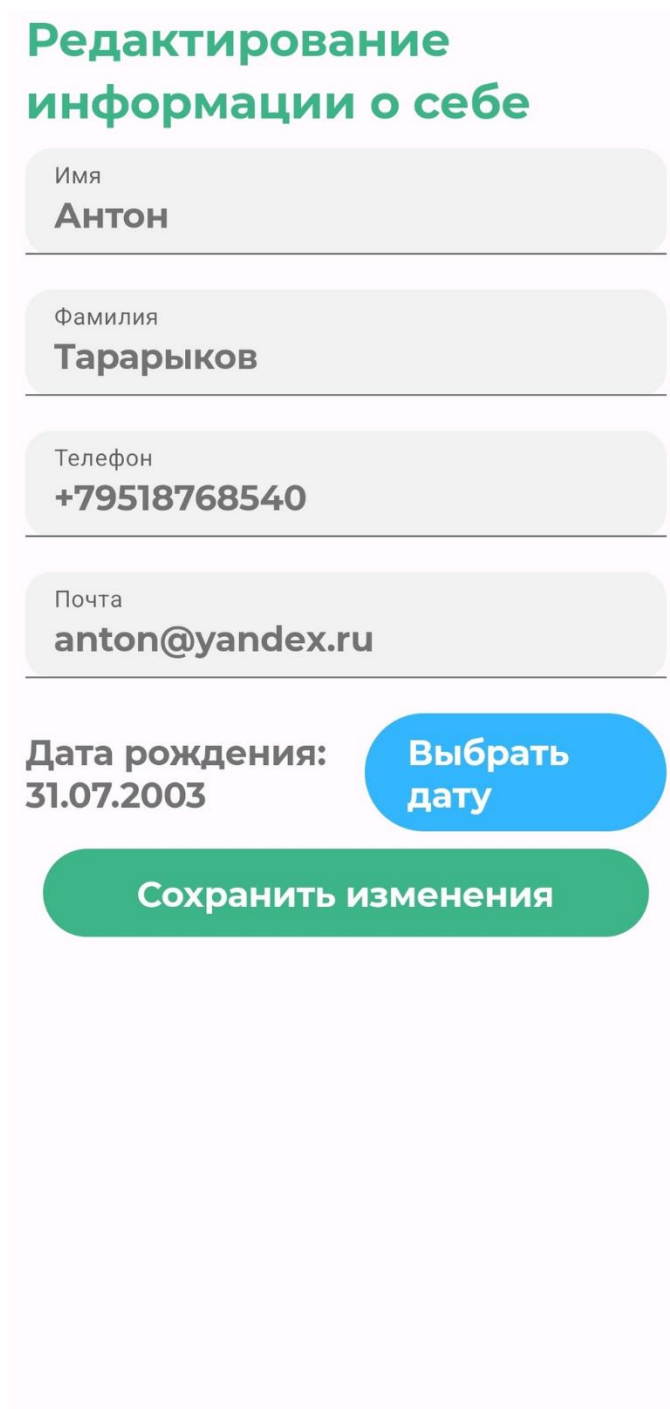


Рисунок 32 - Экран профиля

#### 4.4.2.3 Экран редактирования информации о себе

Страница редактирования информации о себе позволяет авторизованным пользователям обновить профиль, чтобы он всегда оставался актуальным. Реализованный экран редактирования информации о себе представлен на рисунке ниже.



**Редактирование информации о себе**

Имя  
**Антон**

Фамилия  
**Тарарыков**

Телефон  
**+79518768540**

Почта  
**anton@yandex.ru**

Дата рождения:  
**31.07.2003**

**Выбрать дату**

**Сохранить изменения**

Рисунок 33 - Экран редактирования информации о себе

#### 4.4.2.4 Экран редактирования предпочтений

Экран редактирования предпочтений позволяет авторизованным пользователям настроить профиль для наилучшего соответствия их предпочтениям и потребностям в поездках. Реализованный экран редактирования предпочтений представлен на рисунке ниже.

**Редактирование предпочтений**

Любимые музыкальные жанры

Степень разговорчивости (0-10)

Отношение к курению (0-10)

Отношение к животным (0-10)

Дополнительная информация

**Сохранить изменения**

Рисунок 34 - Экран редактирования предпочтений

#### 4.4.2.5 Экран добавления автомобиля

Экран добавления автомобиля позволяет авторизованным пользователям добавить информацию об автомобиле. Реализованный экран добавления автомобиля представлен на рисунке ниже.

**Добавление авто**

Добавить фото

Марка авто

Модель

Цвет

Год выпуска

**Добавить автомобиль**

Рисунок 35 - Экран добавления авто

## **5 Тестирование**

### **5.1 Процесс тестирования**

Процесс тестирования включал следующие этапы:

- Создание и настройка коллекций Postman: для каждого модуля приложения были созданы отдельные коллекции запросов, включающие все возможные сценарии использования API.
- Отправка запросов и анализ ответов: каждый запрос из коллекции отправлялся на сервер, а полученные ответы анализировались на соответствие ожидаемым результатам.
- Проверка обработки ошибок: тестировались различные сценарии с некорректными данными для проверки устойчивости сервера к ошибкам и его способности возвращать корректные коды ошибок.
- Тестирование на нагрузку: проверка стабильности работы сервера при большом количестве запросов за короткий промежуток времени.

### **5.2 Результаты тестирования**

В ходе тестирования были выявлены и исправлены следующие основные проблемы:

- Некорректная обработка некоторых запросов при неверных данных.
- Проблемы с авторизацией и сохранением сессий пользователей.
- Оптимизация производительности серверной части для улучшения времени отклика при высокой нагрузке.
- Все выявленные ошибки были устранены, что позволило добиться стабильной и корректной работы бекэнда приложения.

## **Заключение**

На основании проведенного анализа функциональных требований и поставленных задач, было разработано приложение для организации совместных поездок с использованием транспорта. Целью данного проекта является создание инструмента, который упростит процесс поиска попутчиков для пользователей.

Во время разработки данного приложения был реализован широкий спектр функционала, начиная от возможности авторизации и регистрации пользователей, и заканчивая системой управления поездками и чатами. Проект включал в себя разработку не только функциональных возможностей, но и архитектуры системы, выбор технологий, тестирование и обеспечение качества, а также поддержку приложения.

Использование подробной спецификации требований и их последующая реализация позволили создать удобное и эффективное приложение, которое соответствует потребностям пользователей в организации совместных поездок. Поддержка различных категорий пользователей, таких как авторизованные, незарегистрированные и зарегистрированные, обеспечит удобство использования приложения для всех его потенциальных пользователей.

Таким образом, разработка данного приложения позволила создать удобное и функциональное решение для организации совместных поездок, что способствует повышению комфорта и эффективности использования общественного транспорта.

## СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

- [1] ГОСТ 34.602-2020. ИНФОРМАЦИОННАЯ ТЕХНОЛОГИЯ. КОМПЛЕКС СТАНДАРТОВ НА АВТОМАТИЗИРОВАННЫЕ СИСТЕМЫ. ТЕХНИЧЕСКОЕ ЗАДАНИЕ НА СОЗДАНИЕ АВТОМАТИЗИРОВАННОЙ СИСТЕМЫ [В Интернете]. Доступно: <http://swrit.ru/doc/gost34/34.602-2020.pdf>
- [2] ФЗ "О персональных данных" от 27.07.2006 N 152-ФЗ [В Интернете]. Доступно: [https://www.consultant.ru/document/cons\\_doc\\_LAW\\_61801/](https://www.consultant.ru/document/cons_doc_LAW_61801/)
- [3] Wikipedia [Электронный ресурс]. Режим доступа: <https://clck.ru/3AxCQj> - Заглавие с экрана. (Дата обращения: 30.05.2024).
- [4] Едем.рф [Электронный ресурс]. Режим доступа: <https://clck.ru/3AxCiB> - Заглавие с экрана. (Дата обращения: 30.05.2024).
- [5] CarpoolWorld [Электронный ресурс]. Режим доступа: <https://www.carpoolworld.com/> - Заглавие с экрана. (Дата обращения: 30.05.2024).
- [6] UberPOOL [Электронный ресурс]. Режим доступа: <https://clck.ru/3AxDjk> - Заглавие с экрана. (Дата обращения: 30.05.2024).
- [7] BlaBlaCar [Электронный ресурс]. Режим доступа: <https://www.blablacar.ru/> - Заглавие с экрана. (Дата обращения: 30.05.2024).
- [8] Документация к Python 3.12.3 [Электронный ресурс]. Режим доступа: <https://docs.python.org/3/> - Заглавие с экрана. (Дата обращения: 30.05.2024).
- [8] Документация к Python 3.12.3 [Электронный ресурс]. Режим доступа: <https://docs.python.org/3/> - Заглавие с экрана. (Дата обращения: 30.05.2024).
- [9] Документация к FastAPI [Электронный ресурс]. Режим доступа: <https://fastapi.tiangolo.com/tutorial/> - Заглавие с экрана. (Дата обращения: 30.05.2024).