

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

Факультет компьютерных наук

Кафедра информационных технологий управления

Веб-приложения автопарка с  
возможностью онлайн бронирования и  
оплаты РМК

Курсовой проект  
по дисциплине

Технологии программирования

09.03.02 Информационные системы и технологии  
Информационные системы и технологии управления предприятием

6 семестр 2022/2023 учебного года

Зав. кафедрой	_____ д.т.н., профессор М. Г. Матвеев
Обучающийся	_____ ст. 3 курса оч. отд. П. А. Токарева
Обучающийся	_____ ст. 3 курса оч. отд. К. В. Смотров
Обучающийся	_____ ст. 3 курса оч. отд. М. А. Федосова
Руководитель	_____ К. В. Зенин, преподаватель
Руководитель	_____ В.С. Тарасов, ст. преподаватель _____.20__

Воронеж 2023

# Содержание

Содержание.....	2
Введение.....	4
1 Постановка задач.....	5
1.1 Требования к разрабатываемой системе.....	5
1.1.1 Функциональные требования.....	5
1.1.1.1 Для клиентов .....	6
1.1.1.2 Для водителей-курьеров .....	6
1.1.1.3 Для администраторов.....	7
1.1.2 Технические требования.....	8
1.2 Требования к интерфейсу.....	8
1.3 Задачи, решаемые в процессе разработки .....	8
2 Анализ предметной области .....	10
2.1 Терминология (гlossарий) предметной области .....	10
2.2 Обзор аналогов .....	13
2.2.1 Автопрокат car2you .....	13
2.2.2 Авто М4 .....	14
2.2.3 RentCars .....	15
2.3 Моделирование системы .....	18
2.3.1 Диаграмма в стиле методологии IDEF0.....	18
2.3.2 Диаграмма прецедентов.....	20
2.3.3 Диаграмма состояний.....	22
2.3.4 Диаграмма последовательности .....	24
2.3.5 Диаграмма активностей .....	27
2.3.6 Диаграмма классов.....	29
2.3.7 ER-диаграмма .....	30
2.3.8 Диаграмма развёртывания.....	30
2.3.9 Диаграмма сотрудничества .....	31
2.3.10 Диаграмма объектов.....	32
2.4 Аналитика веб-приложения .....	33

2.5 Границы проекта .....	35
3 Реализация.....	36
3.1 Средства реализации .....	36
3.2 Реализация серверной части веб-приложения .....	37
3.3 Реализация клиентской части веб-приложения .....	37
4 Тестирование .....	39
4.1 Ручное UI–тестирование.....	39
4.2 Юзабилити–тестирование .....	43
Заключение .....	47
Список используемых источников.....	48

## **Введение**

В настоящее время Интернет–технологии все более активно используются в различных сферах, в том числе и в автомобильной индустрии. Также стоит помнить, что в современном мире, где время – деньги, автомобиль является неотъемлемой частью в жизни многих людей. Благодаря автомобилю можно значительно сократить время на перемещение между точками назначения, улучшить комфортность поездки и иметь большую свободу при выборе маршрута и времени поездки. К сожалению, не каждый обладает своим собственным авто, и в этом случае на помощь приходят арендные автопарки. Однако процесс аренды автомобиля может быть не всегда удобным и быстрым, особенно в случае, когда нужно совершить бронирование и оплату в режиме онлайн.

Также стоит отметить, что сейчас большинство людей любят путешествовать по разным городам, добираясь туда на поезде или самолете, но иметь машину для перемещения по городу будет очень удобно.

В данной курсовой работе рассмотрена разработка веб–приложения автопарка с возможностью онлайн бронирования и оплаты. Основная цель проекта заключается в создании удобной и функциональной системы, которая позволит пользователям легко и быстро выбрать нужный автомобиль, забронировать его на определенный период времени и произвести оплату онлайн. Также проект позволяет клиенту самостоятельно забрать автомобиль из офиса, или оформить доставку автомобиля по необходимому адресу.

В работе будет рассмотрен процесс проектирования и разработки веб–приложения, включая выбор подходящих технологий и инструментов, создание базы данных, разработку интерфейса и реализацию основных функций приложения. Кроме того, будут рассмотрены вопросы безопасности и защиты данных пользователей.

## **1 Постановка задач**

Целью данного проекта является создание веб-приложения автопарка с возможностью онлайн бронирования и оплаты.

Основными задачами проекта являются реализации услуг аренды автомобилей и возможность оплаты по QR-коду.

Для достижения поставленных целей необходимо, во-первых, иметь представление о разрабатываемой системе, представленное необходимыми UML-диаграммами и разработанным дизайном веб-приложения, как в целом, так и в отдельных сценариях.

Во-вторых, необходимо реализовать базы данных, которые будут хранить информацию об автомобилях и клиентах автопарка, а также, осуществить возможность оплаты по QR-коду, посредством внедрения данной функции в систему и ее реализации благодаря “заглушке”.

Также, необходимо провести тестирование системы и ее аналитику.

### **1.1 Требования к разрабатываемой системе**

#### **1.1.1 Функциональные требования**

К разрабатываемому приложению выдвинуты следующие требования:

- разделение пользователей на: клиентов, администраторов и водителей-курьеров;
- обеспечение возможности отправки пользователем писем на почту администрации веб-приложения;
- обеспечение формы для аренды машины с возможностью доставки выбранной машины по указанному клиентом адресу;
- обеспечение возможности оплаты по QR-коду;
- обеспечение возможности авторизации пользователя;
- обеспечение возможности восстановления доступа к аккаунту пользователя.

#### **1.1.1.1 Для клиентов**

- Обеспечение возможности просматривать подробную информацию о выбранной машине;
- обеспечение возможности редактирования в личном кабинете информации о себе;
- обеспечение возможности выбирать способ оплаты аренды;
- обеспечение возможности просмотра истории заказов аренды;
- обеспечение возможности просматривать каталог компании;
- обеспечение возможности просмотра главной страницы веб-приложения;
- обеспечение формы для заказа на аренду с возможностью доставки выбранной машины по указанному клиентом адресу;
- обеспечение возможности авторизации;
- обеспечение возможности отправки писем на почту администрации веб-приложения;
- обеспечение возможности просмотра контактных данных компании;
- обеспечение возможности восстановления доступа к личному кабинету;
- обеспечение возможности просмотра условий аренды автомобиля.

#### **1.1.1.2 Для водителей-курьеров**

- Обеспечение возможности просматривать подробную информацию о конкретной машине;
- обеспечение возможности просматривать каталог компании;
- обеспечение возможности просмотра главной страницы веб-приложения;
- обеспечение возможности авторизации;
- обеспечение возможности отправки писем на почту администрации

- веб-приложения;
- обеспечение возможности просмотра контактных данных компании;
- обеспечение возможности просмотра условий аренды машины;
- обеспечение возможности просматривать список заказов-доставок для конкретного водителя-курьера.

#### **1.1.1.3 Для администраторов**

- Обеспечение возможности просматривать подробную информацию о выбранной машине;
- обеспечение возможности просматривать каталог компании;
- обеспечение возможности просмотра главной страницы веб-приложения;
- обеспечение формы для аренды машины с возможностью доставки выбранной машины по указанному клиентом адресу (данная процедура реализуется только в случае ошибок в заявках клиентов или отсутствия на данный момент в офисе компании выбранной пользователем машины для аренды);
- обеспечение возможности просмотра контактных данных компании;
- обеспечение возможности просмотра заявок на аренду машин от клиентов;
- обеспечение возможности просмотра условий аренды автомобиля;
- обеспечение возможности удалять конкретную заявку на аренду машины от пользователей;
- обеспечение возможности просматривать информацию о клиентах;
- обеспечение возможности добавлять машину в каталог;
- обеспечение возможности удалять машину из каталога;
- обеспечение возможности авторизации;
- обеспечение возможности назначить водителя-курьера на заказ-

доставку.

### **1.1.2 Технические требования**

Программный продукт должен обеспечить:

- авторизацию пользователей с использованием телефона и пароля;
- шифрование пароля при записи в БД;
- хранение данных в БД.

### **1.2 Требования к интерфейсу**

Интерфейс должен быть выполнен в единой для всех экранов цветовой гамме, едином стиле. Все надписи должны быть легко читаемы, все элементы управления должны быть выполнены в едином стиле, размере, должны выделяться на фоне содержимого экранов.

Интерфейс должен содержать только необходимую для пользователя информацию. Информация должна находиться в тех местах приложения, где она будет актуальна. Основные элементы управления должны быть заметны для пользователя.

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

Перед проектом были поставлены следующие задачи:

- анализ предметной области;
- анализ аналогов;
- написание технического задания;
- описание разрабатываемой системы UML диаграммами;
- создание доски Trello и репозитория GitHub;
- 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.

## **2 Анализ предметной области**

### **2.1 Терминология (гlossарий) предметной области**

Проект, система – разрабатываемое веб-приложение.

Личный кабинет клиента – веб-страница, где отражаются персональная информация пользователя, введенная на этапе регистрации и/или отредактированная позже, а также история заказов данного пользователя, переход к каталогу и переход к странице редактирования личных данных.

Личный кабинет администратора – веб-страница, где отражаются список заявок на аренду машины, информация о клиентах, а также возможность назначить на заказ-доставку водителя курьера.

Заказ–доставка – заказ на аренду машины с дополнительной услугой доставки машины по указанному клиентом адресу.

Личный кабинет водителя–курьера – веб-страница, где отражается список заказов–доставок для конкретного водителя–курьера.

Водитель–курьер – пользователь, статус которого был назначен администратором.

Администратор – пользователь, данные которого были заранее занесены в систему под отличительным доменом.

Пользователь (Клиент) – авторизованный в системе человек, пользующийся функционалом веб-приложения.

Front–end – клиентская часть приложения. Отвечает за получение информации с программно–аппаратной части и отображение ее на устройстве пользователя.

Клиентская сторона – компьютер, использующий ресурсы сервера и предоставляющий пользователю возможность взаимодействия с системой.

Back–end – программно–аппаратная часть приложения. Отвечает за функционирование внутренней части приложения.

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

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

Фреймворк – программное обеспечение, облегчающее разработку и объединение разных компонентов большого программного проекта.

REST API (REST) – стиль архитектуры программного обеспечения для построения масштабируемых веб–приложений.

Python – высокоуровневый язык программирования общего назначения с динамической строгой типизацией и автоматическим управлением памятью.

Django – это высокоуровневый Python веб–фреймворк, который позволяет быстро создавать веб–сайты.

Система управления базами данных (СУБД) – это набор программ, которые управляют структурой БД и контролируют доступ к данным, хранящимся в БД.

PostgreSQL – это реляционная база данных с открытым кодом.

JavaScript – это язык программирования, выполняющийся на стороне клиента и использующийся для взаимодействия с пользователем.

CSS – формальный язык описания внешнего вида веб–страницы, написанного с использованием языка разметки (HTML).

HTML – стандартизированный язык разметки для просмотра веб–страниц в браузере.

React – JavaScript–библиотека для создания пользовательских интерфейсов.

Продуктовая воронка – это маркетинговая модель, описывающая предполагаемое «путешествие» будущего покупателя от первого знакомства с предложением или товаром до реальной покупки.

Тестирование графического интерфейса пользователя или UI-тестирование – это процесс тестирования элементов управления в приложении, который помогает убедиться, что интерфейс соответствует ожидаемой производительности и функциональности. Задача UI-тестировщиков убедиться, что в функциях пользовательского интерфейса отсутствуют дефекты.

Юзабилити–тестирование – это метод оценки интерфейса со стороны удобства и эффективности его использования.

## 2.2 Обзор аналогов

### 2.2.1 Автопрокат car2you

Car2you[1] – сайт, где можно арендовать автомобиль как эконом класса, так и комфорт класса. Содержит инструкцию по аренде, ответы на часто задаваемые вопросы, каталог с разделением автомобилей по классам. Интерфейс сайта представлен в соответствии с рисунком 1, рисунком 2, рисунком 3.

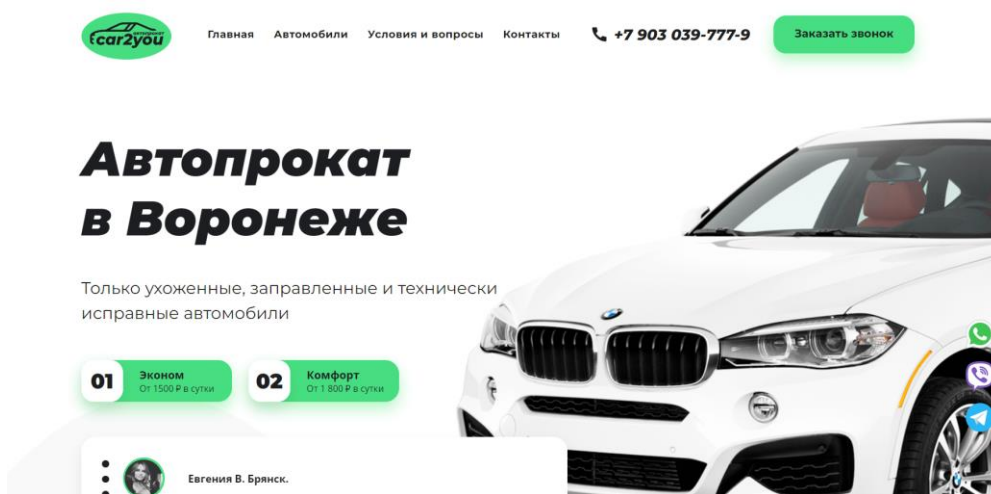


Рисунок 1 - Интерфейс сайта «car2you»

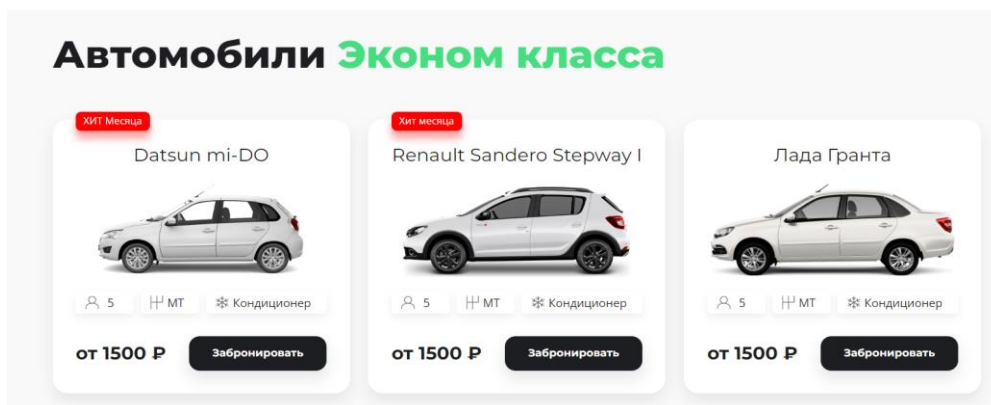


Рисунок 2 - Каталог эконом класса

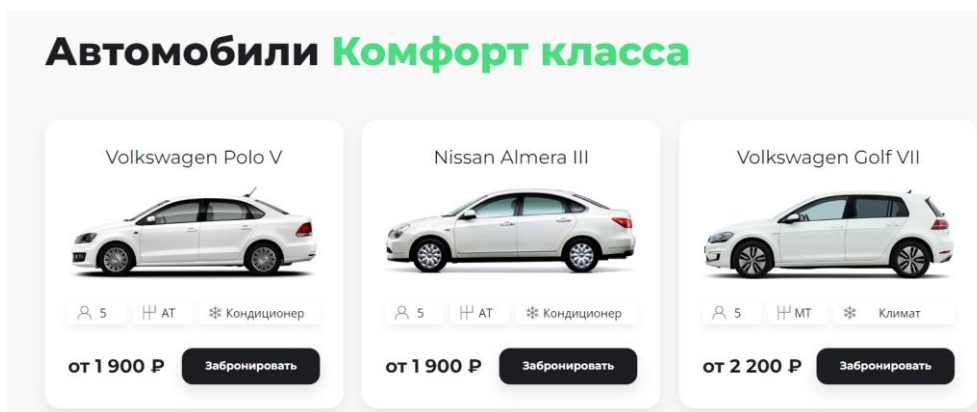


Рисунок 3 - Каталог комфорт класса

Недостатками сайта Car2you являются:

- отсутствует поддержка непосредственно на сайте;
- недостаточно информации о машинах.

### 2.2.2 Авто М4

Автопрокат М4[2] – на данном сайте для аренды представлены модели от эконома до премиум класса, промышленный автотранспорт, а так же разнообразное промышленное оборудование. Интерфейс сайта представлен в соответствии с рисунком 4.

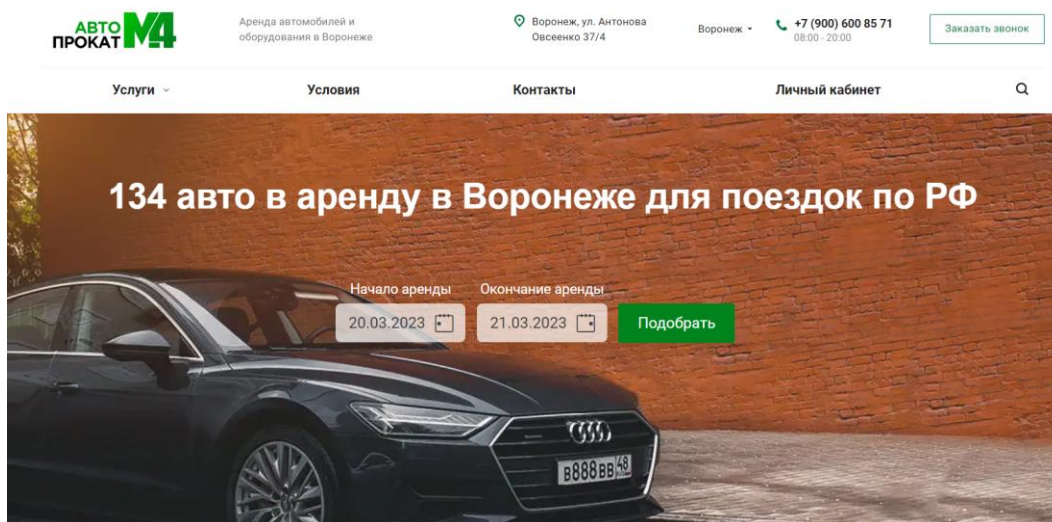


Рисунок 4 - Интерфейс сайта «Авто М4»

Из плюсов стоит отметить возможность изменить формат просмотра каталога в соответствии с рисунком 5.

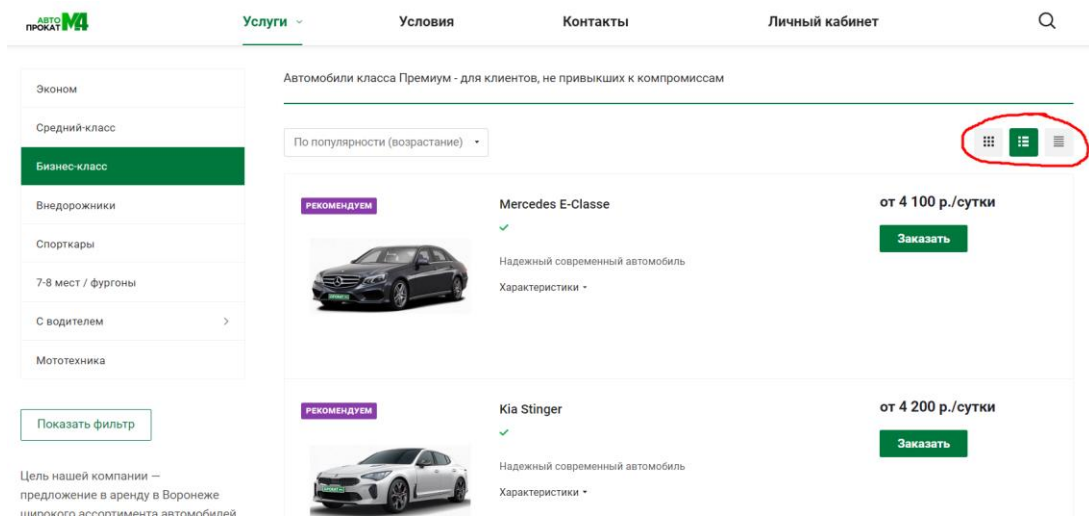


Рисунок 5 - Выбор просмотра страницы

Недостатками сайта Авто М4 являются:

- отсутствует поддержка именно на сайте;
- большой объём текста, главные моменты плохо выделены.

Пример вышесказанного приведен в соответствии с рисунком 6.

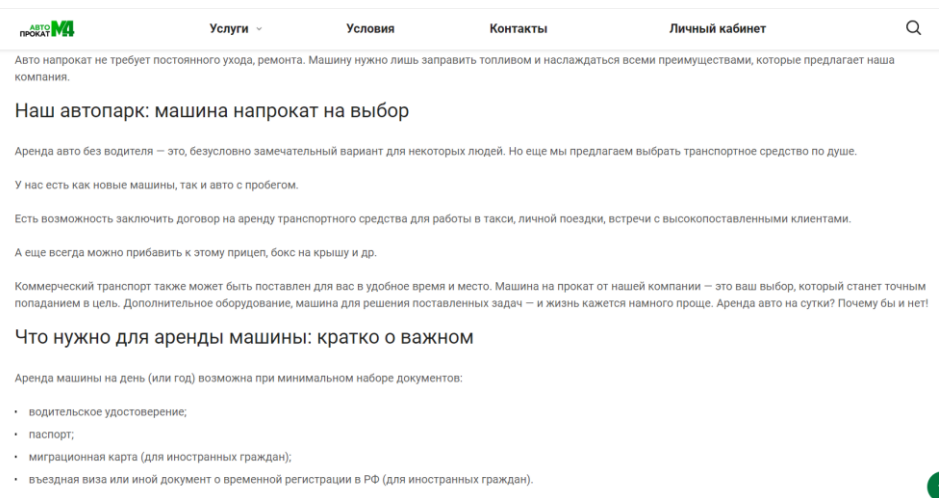


Рисунок 6 - Информация об условиях аренды

## 2.2.3 RentCars

RentCars[3] — аренда автомобилей от класса эконом, до класса внедорожник. Интерфейс сайта представлен в соответствии с рисунком 7.

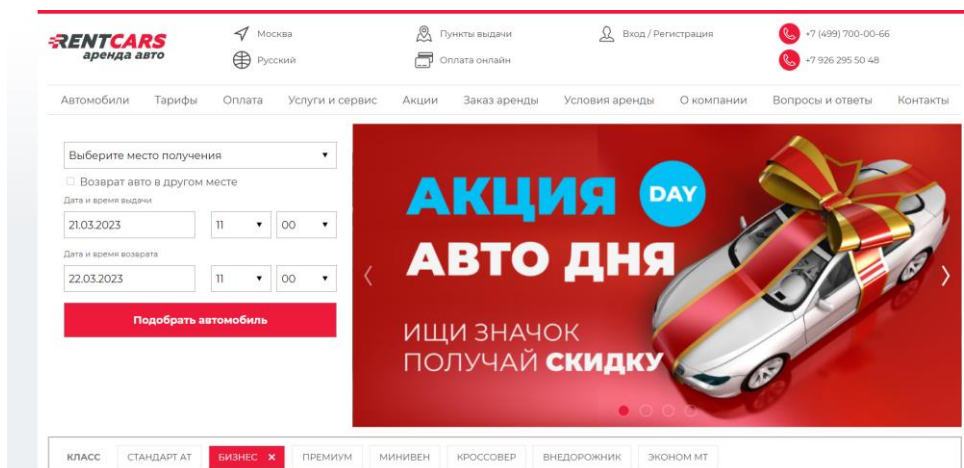


Рисунок 7 - Интерфейс главной страницы «RentCars»

На сайте представлен каталог с большим выбором как новых, так и машин с пробегом. В каталоге можно сортировать машины по классам, выбрав одну или несколько позиций – пример приведен в соответствии с рисунком 8.

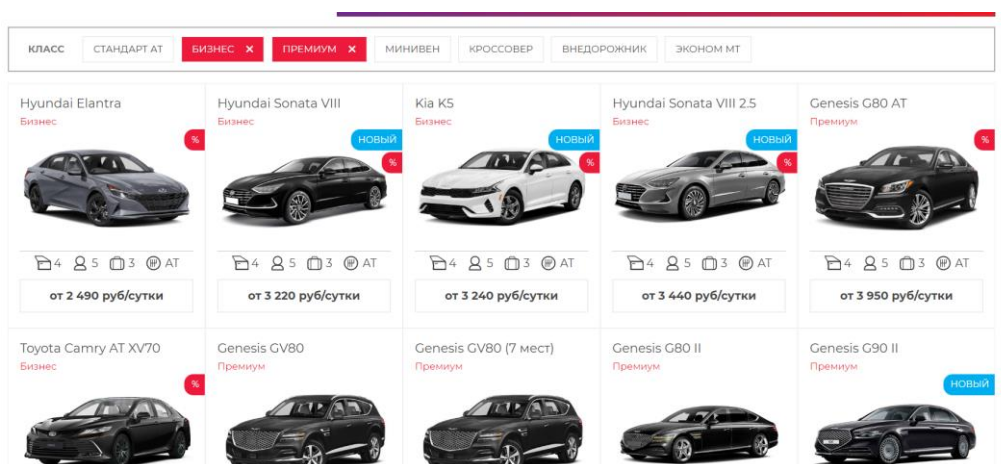


Рисунок 8 - Каталог «RentCars»

При выборе определённой модели открывается страница с подробным описанием машины, требованием к арендатору, и видео обзором – пример приведен в соответствии с рисунком 9 и рисунком 10.



Автомобили

Тарифы

Оплата

Услуги и сервис

Акции

Заказ аренды

Условия аренды

О компании

Вопросы и ответы

Контакты

Классы автомобилей

Автомобили класса Эконом

Автомобили класса Стандарт

Автомобили класса Бизнес



Автомобили класса Премиум

Автомобили класса Минивен

Автомобили класса Кроссовер

Автомобили класса Внедорожник


Аренда Chevrolet Tahoe RWD

Основные характеристики:

Число дверей:	4	Количество пассажиров:	7
Мест для багажа:	3	Подушек безопасности:	8
Кондиционер:	✓	Аудиосистема:	✓
Тип топлива:	AWD	Коробка передач:	AT

Цвета кузова в парке:



1-2 суток	3-6 суток	7-14 суток	15-30 суток	30+ суток
11 310 Р	11 310 Р	11 310 Р	11 310 Р	9 430 Р

Заказать

Все цены указаны в рублях, без НДС.

Внимание! Текущий тариф действует до 30 марта 2023 включительно. Для для подробного ознакомления с тарифами на другие периоды перейдите в раздел ТАРИФЫ и выберите интересующие вас даты и нажмите кнопку "смотреть все цены" напротив интересующего вас автомобиля.

Рисунок 9 - Страница автомобиля «RentCars»

Характеристики:

Объем двигателя:	5.3L
Салон:	Черная кожа
Мультимедиа:	Apple CarPlay и Android Auto
Мощность двигателя:	343 л.с.
Привод:	RWD (задний)
Объем бака:	98 литров
Год выпуска:	2022
Круиз-контроль:	Да
Парктроники:	передние и задние
Камера заднего вида:	Да
Количество мест:	6+1 (капитанские кресла)

Требования к арендатору:

- ✓ для класса Эконом, Стандарт возраст от 20 лет и стаж вождения не менее 1 года
- ✓ для остальных классов возраст от 23 лет и стаж вождения не менее 2-х лет

Документы для аренды:

- ✓ Паспорт
- ✓ Водительские права
- ✓ Именная банковская карта

✓ Депозит: 40 000 Р

Описание:

Chevrolet Tahoe - внедорожник, которому не нужно доказывать свой легендарный статус. Монументальный. Современный. Универсальный. Все это - Chevrolet Tahoe. Он идеально сочетает в себе узнаваемые брутальные черты, ошеломляющую мощь двигателя V8 и первоклассный комфорт просторного салона.

Обновление нашего автопарка 2022 года: Chevrolet Tahoe

автопарке 2022 года

Смотреть... Поделиться

Chevrolet Tahoe




Рисунок 10 - Характеристики автомобиля «RentCars»

Минусы сайта «RentCars»:

- отсутствие поддержки пользователей на сайте;
- мелкий шрифт и большой объем текста.

## 2.3 Моделирование системы

### 2.3.1 Диаграмма в стиле методологии IDEF0

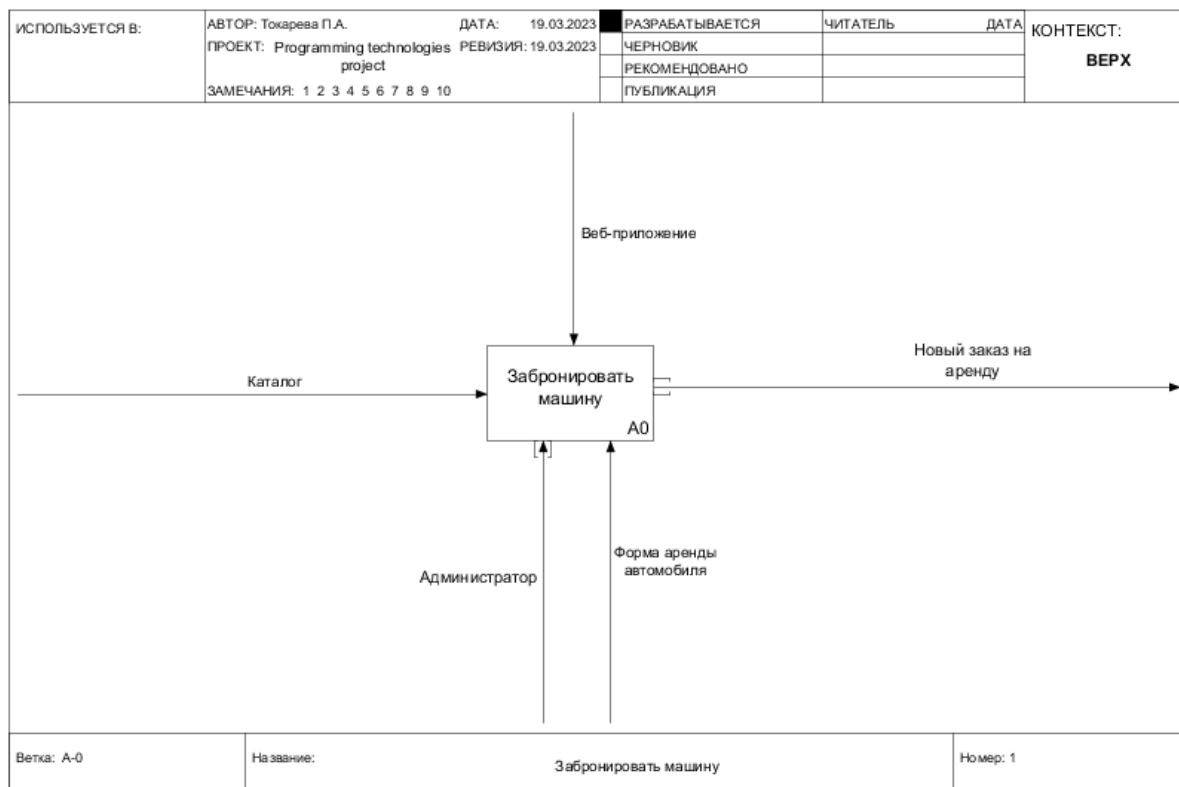


Рисунок 11 - Контекстная диаграмма со стороны Клиента

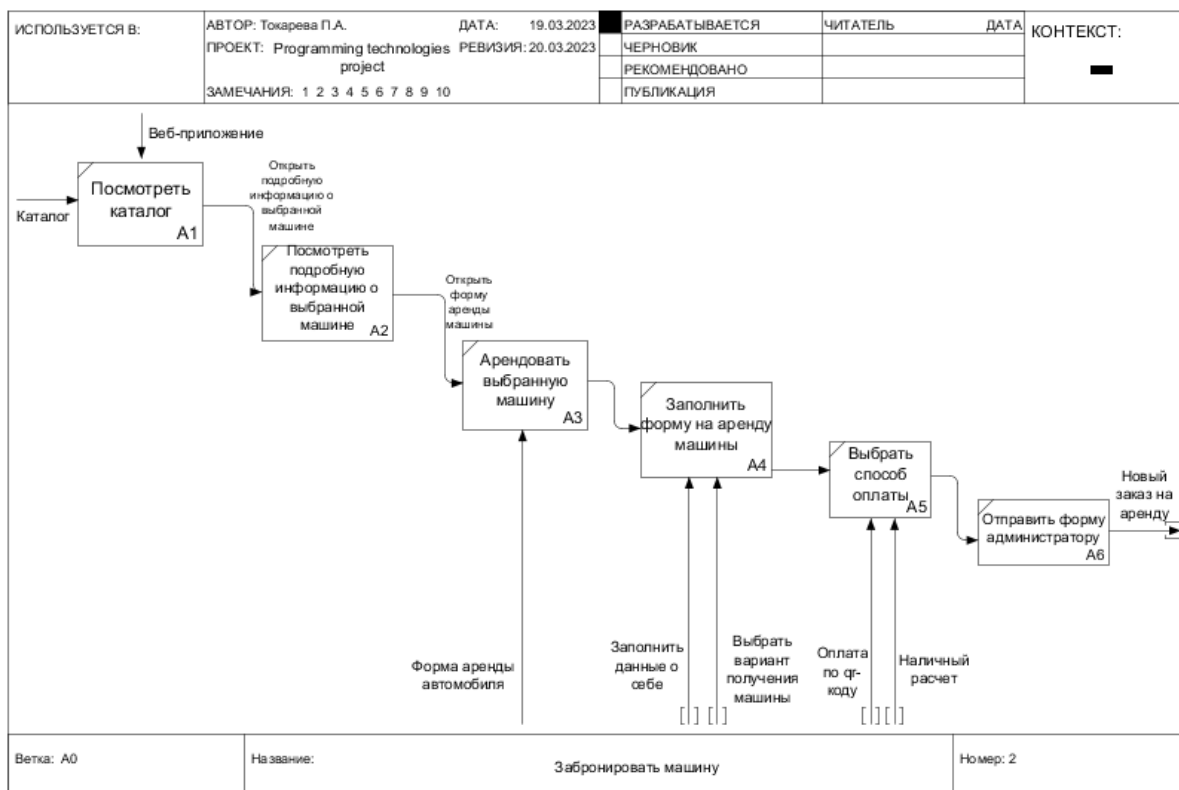


Рисунок 12 - Дочерняя диаграмма со стороны Клиента

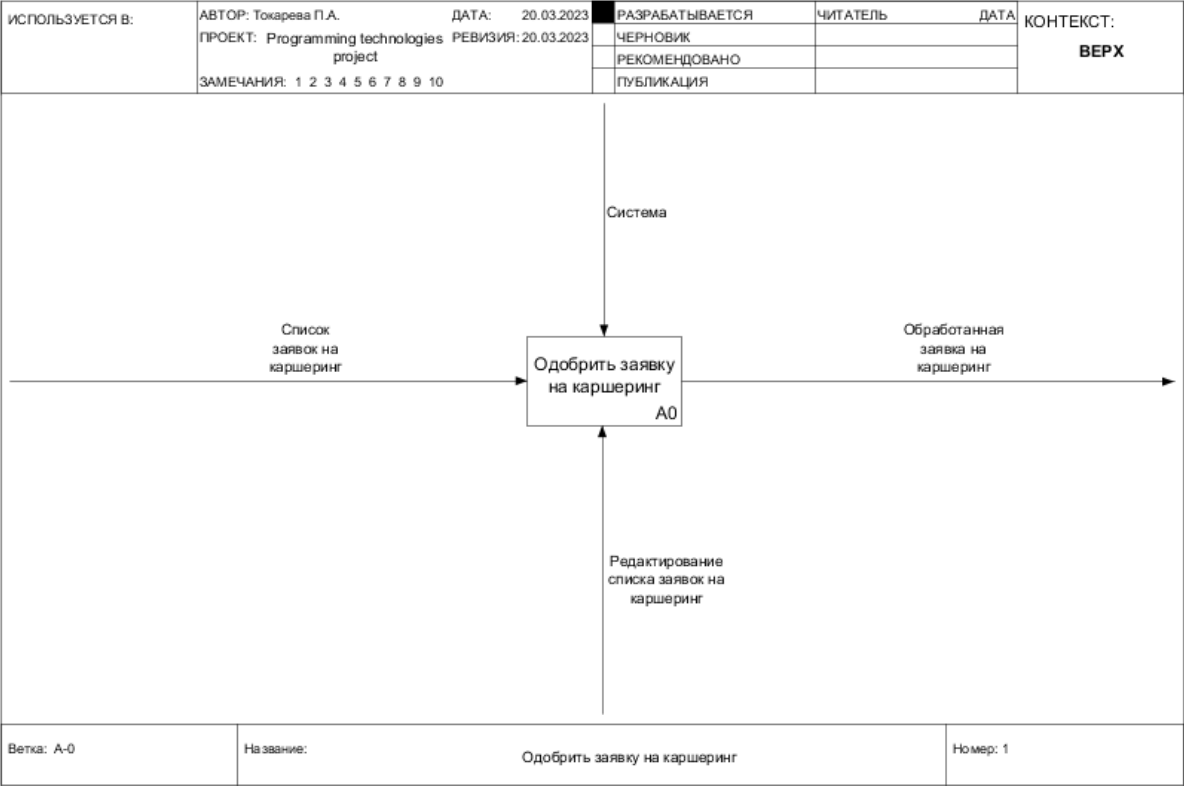


Рисунок 13 - Контекстная диаграмма со стороны Администратора

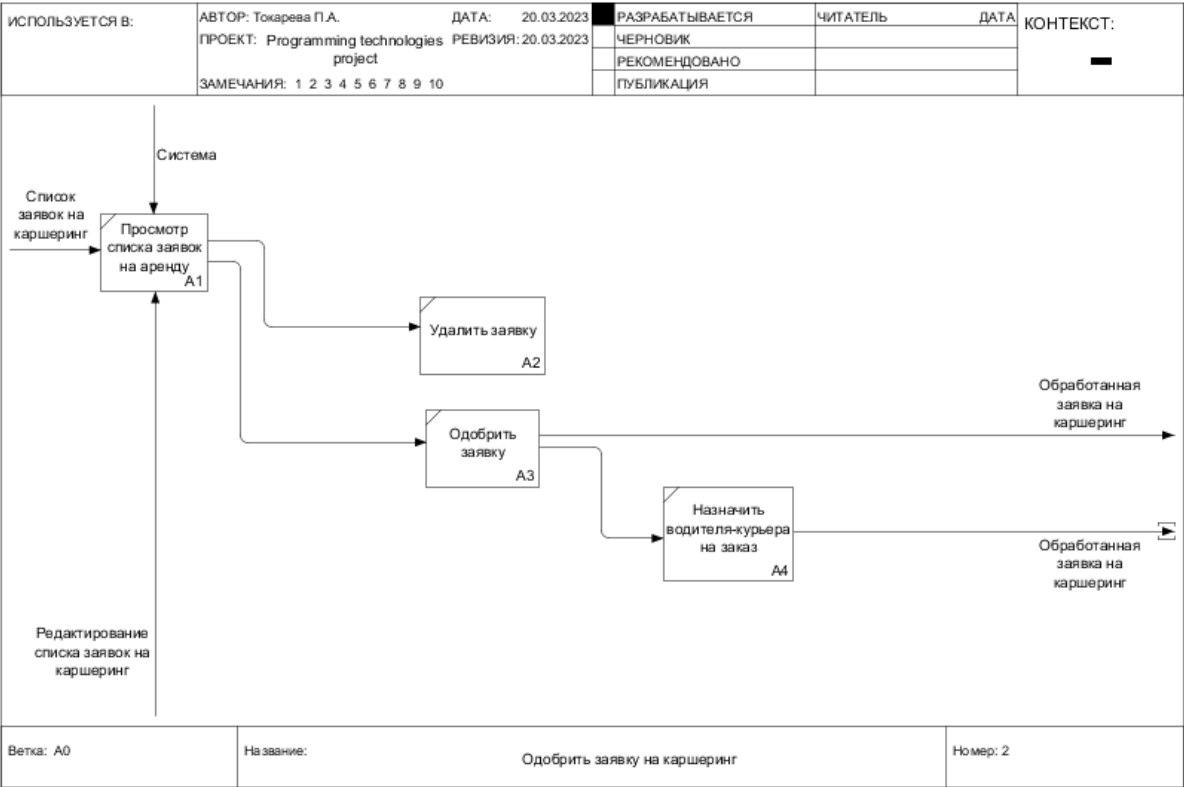


Рисунок 14 - Дочерняя диаграмма со стороны Администратора

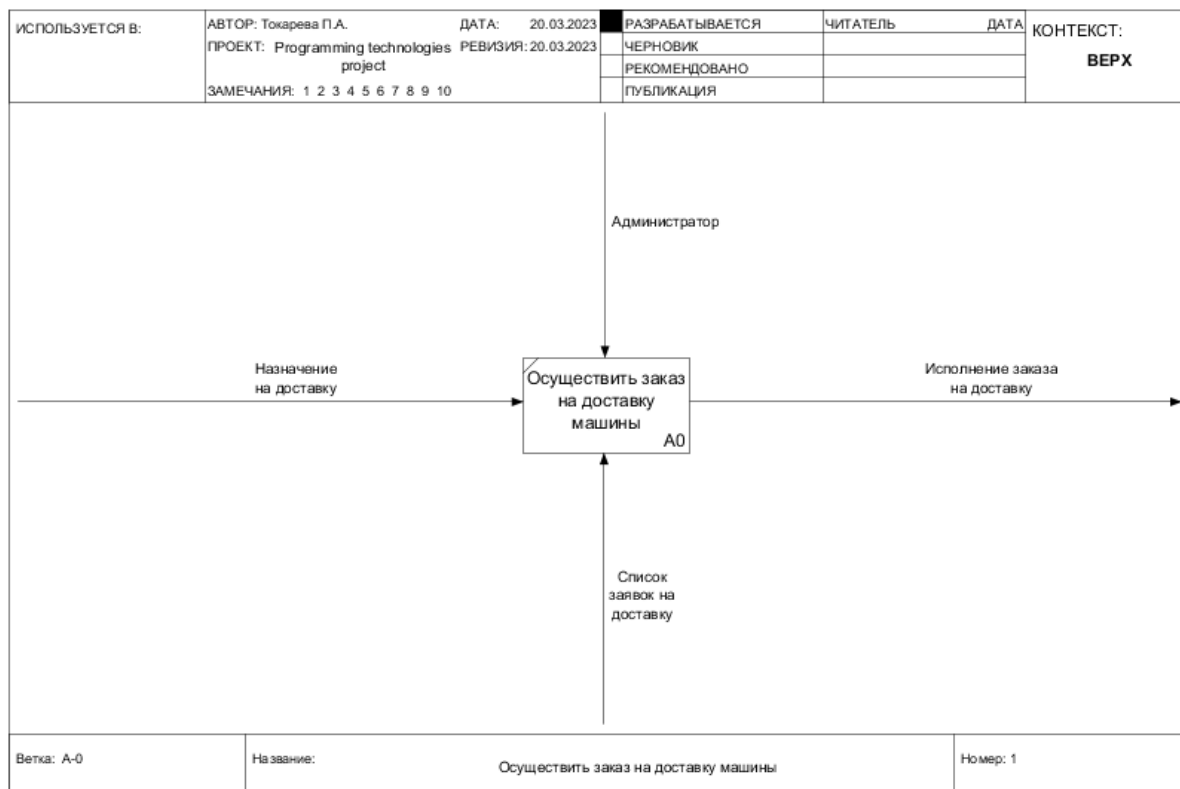


Рисунок 15 - Контекстная диаграмма со стороны Водителя-курьера

### 2.3.2 Диаграмма прецедентов

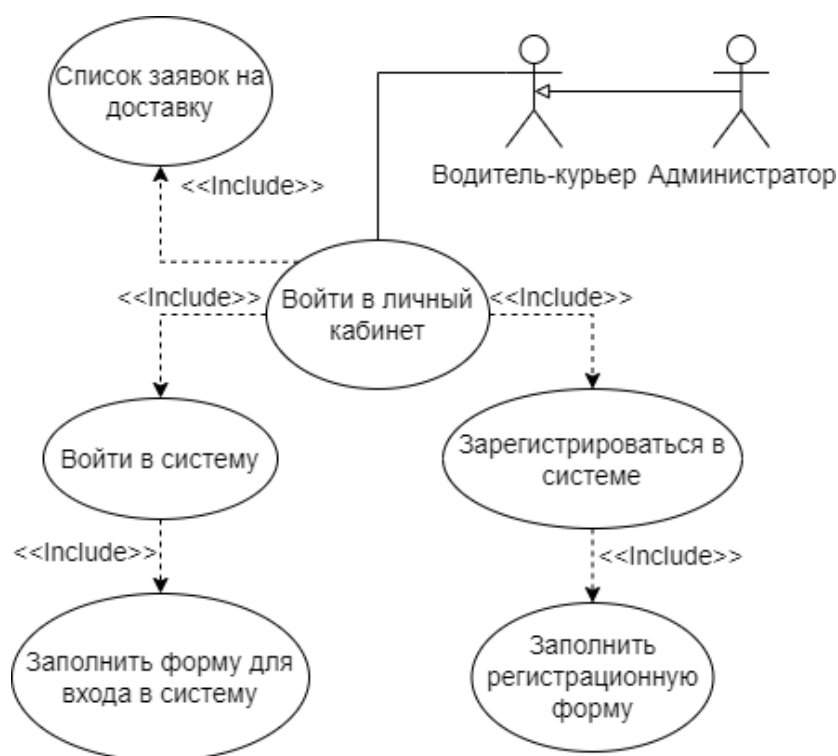


Рисунок 16 - Диаграмма прецедентов со стороны Водителя-курьера

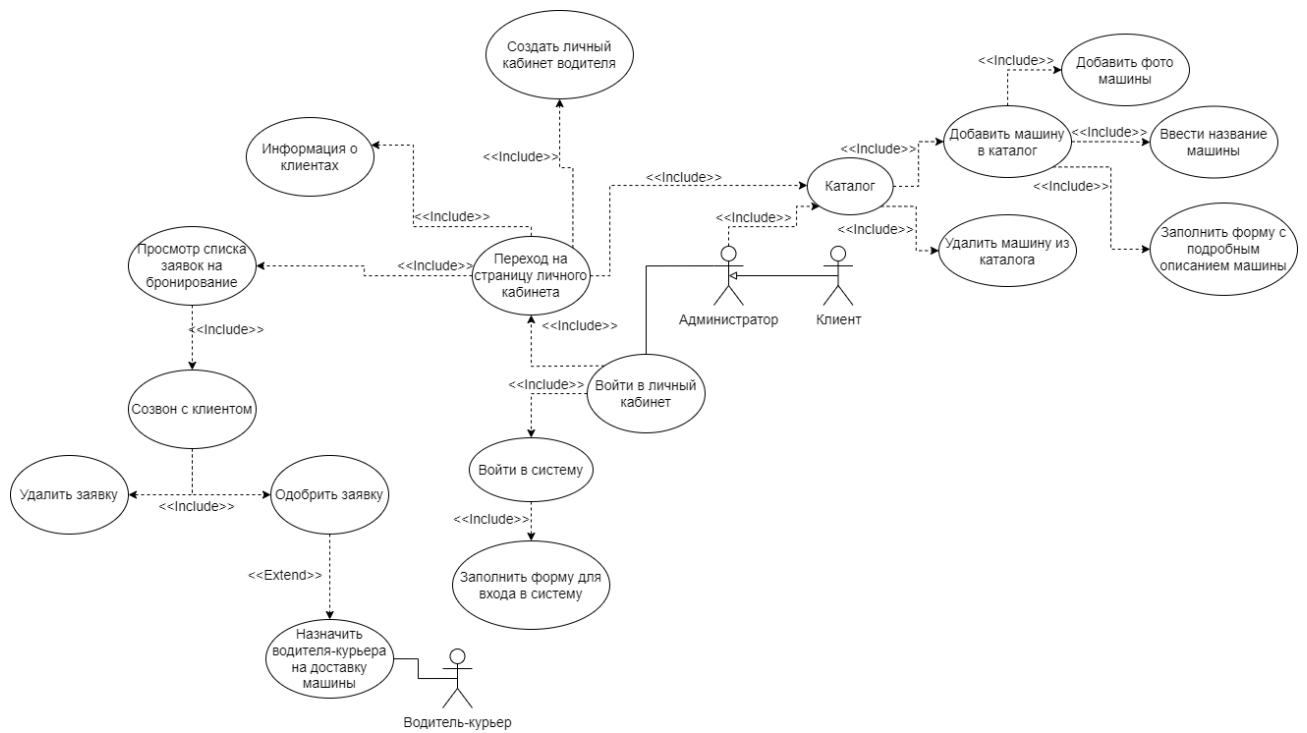


Рисунок 17 - Диаграмма прецедентов со стороны Администратора

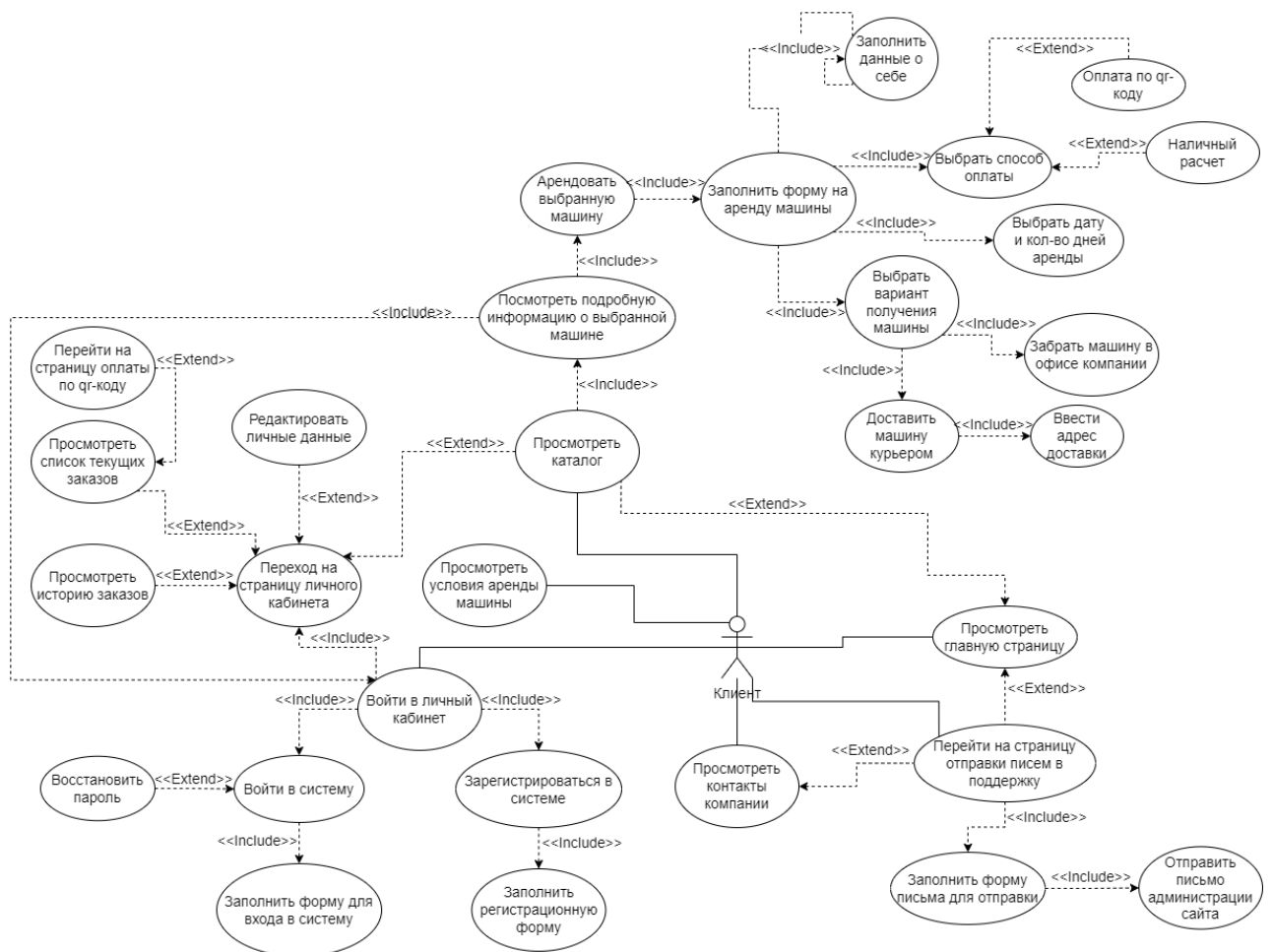


Рисунок 18 - Диаграмма прецедентов со стороны Водителя-курьера

### 2.3.3 Диаграмма состояний

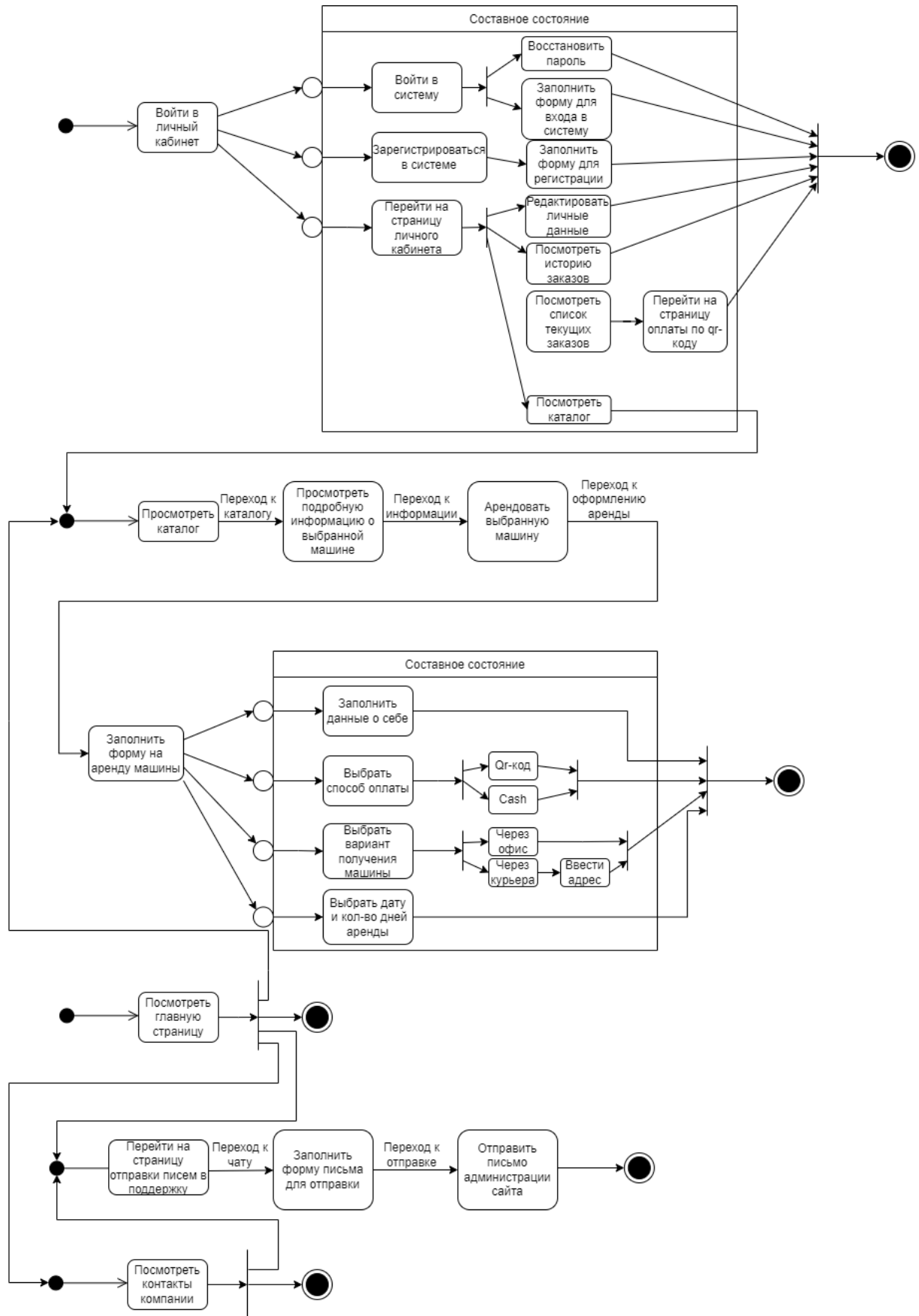


Рисунок 19 - Диаграмма состояний со стороны Клиента

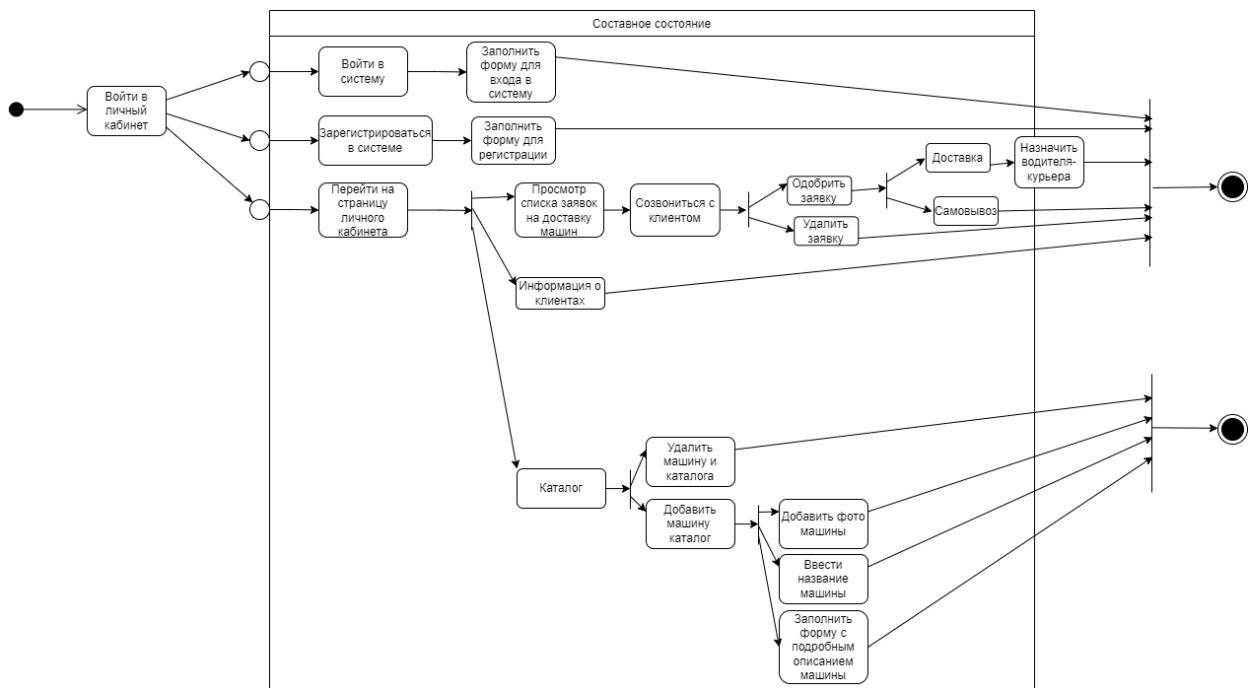


Рисунок 20 - Диаграмма состояний со стороны Администратора

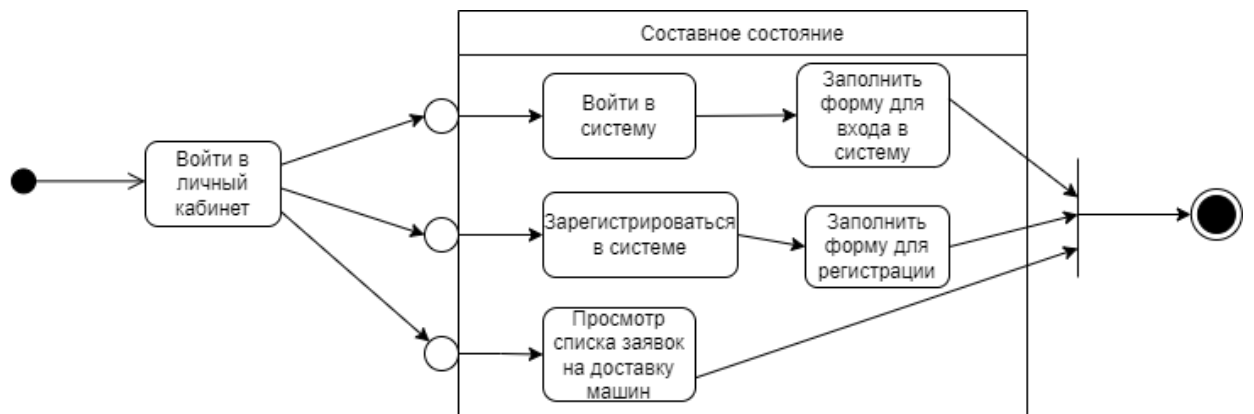


Рисунок 21 - Диаграмма состояний со стороны Водителя-курьера

## 2.3.4 Диаграмма последовательности

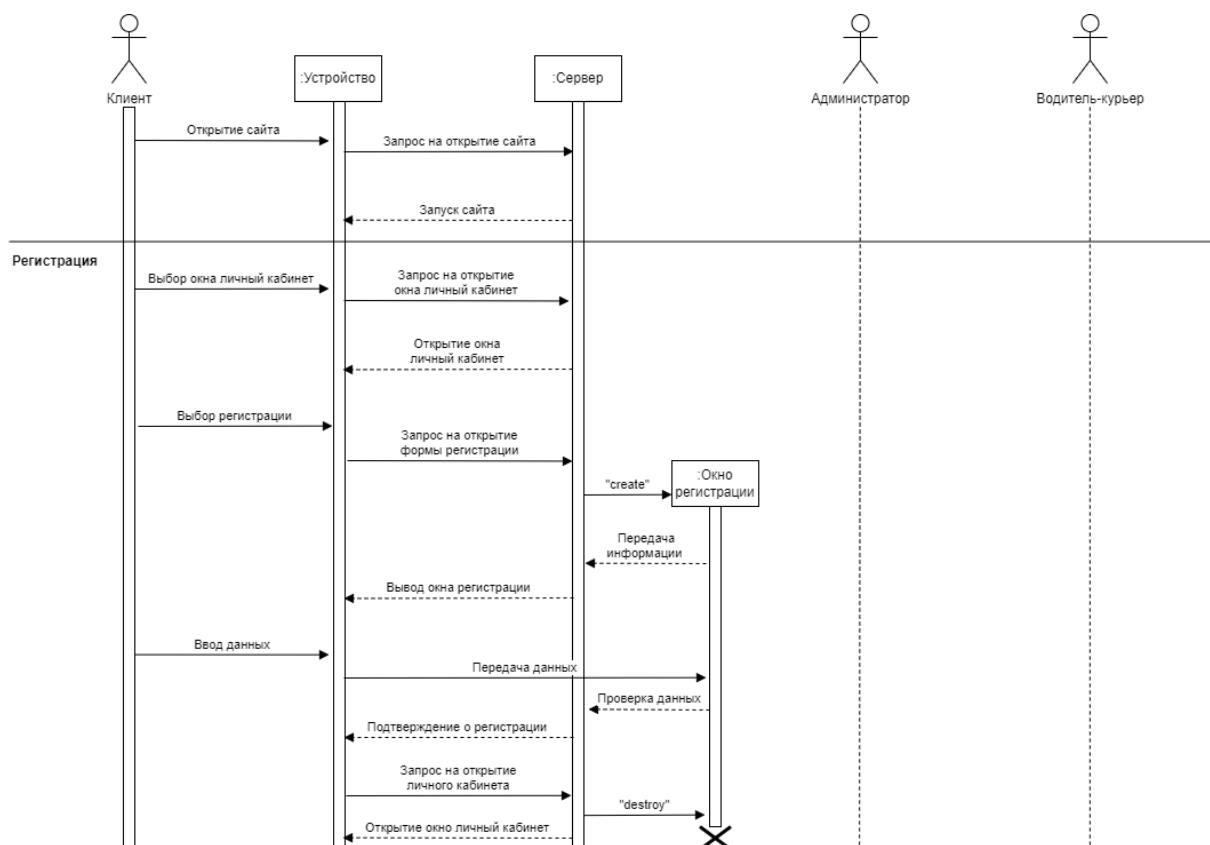


Рисунок 22 - Диаграмма последовательности. Регистрация

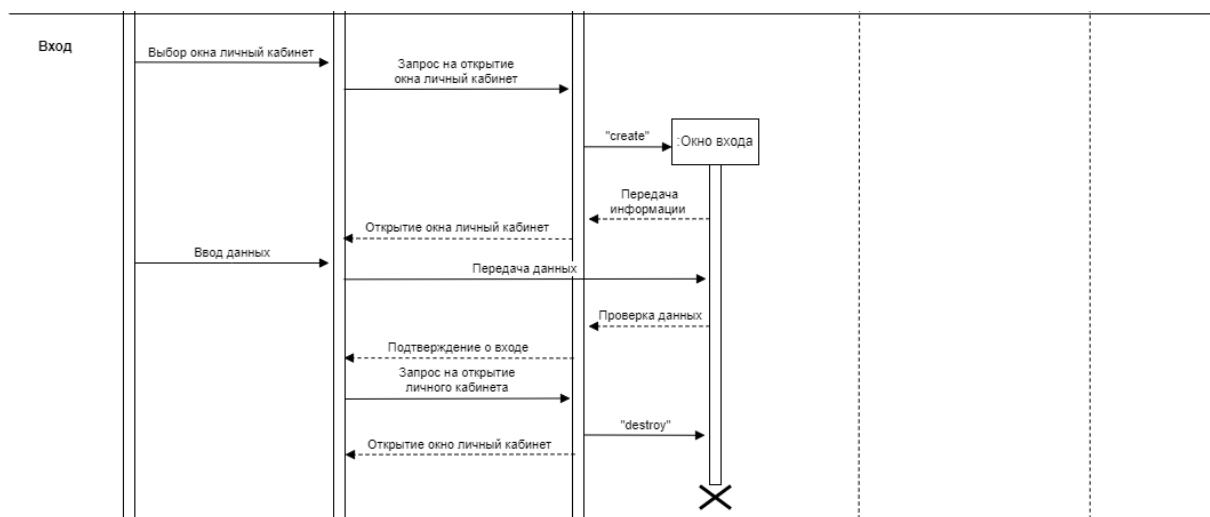


Рисунок 23 - Диаграмма последовательности. Вход



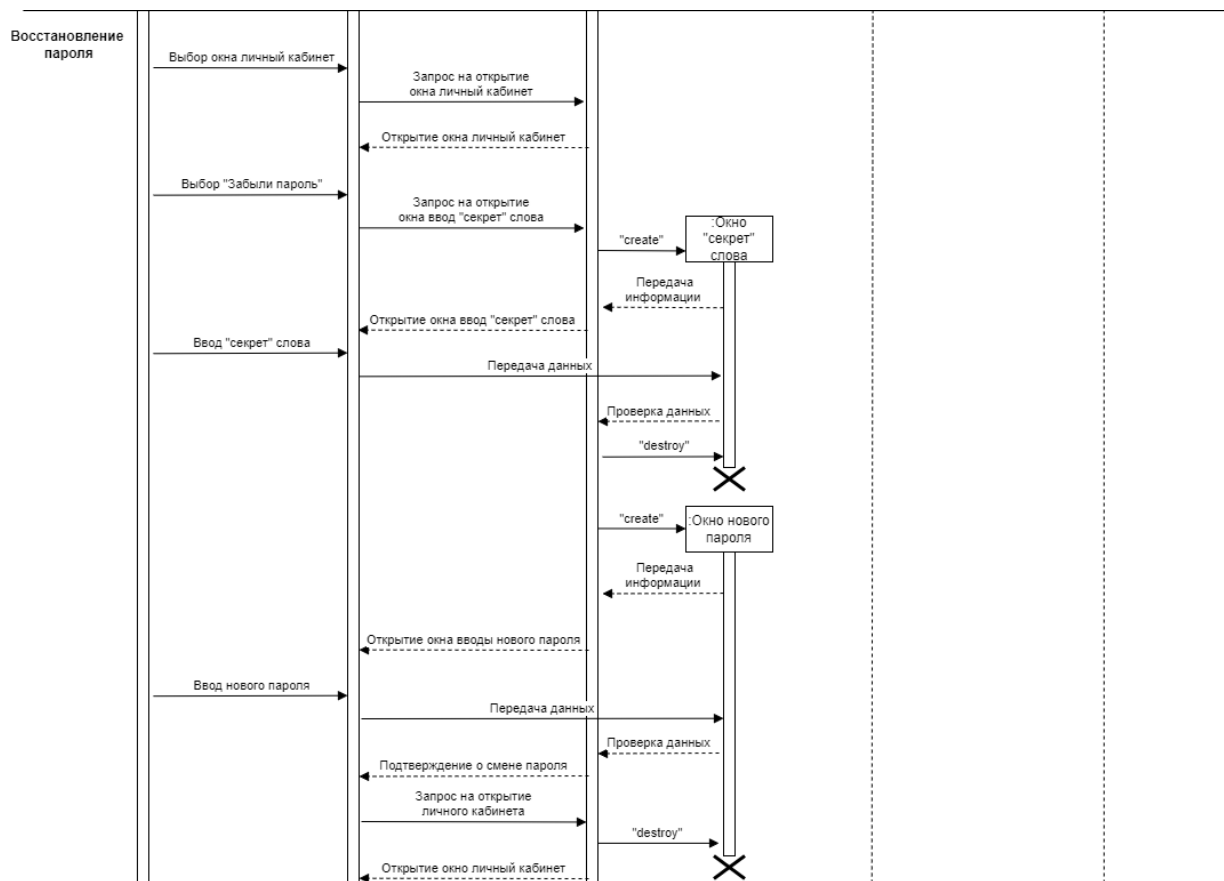


Рисунок 24 - Диаграмма последовательности. Восстановление пароля

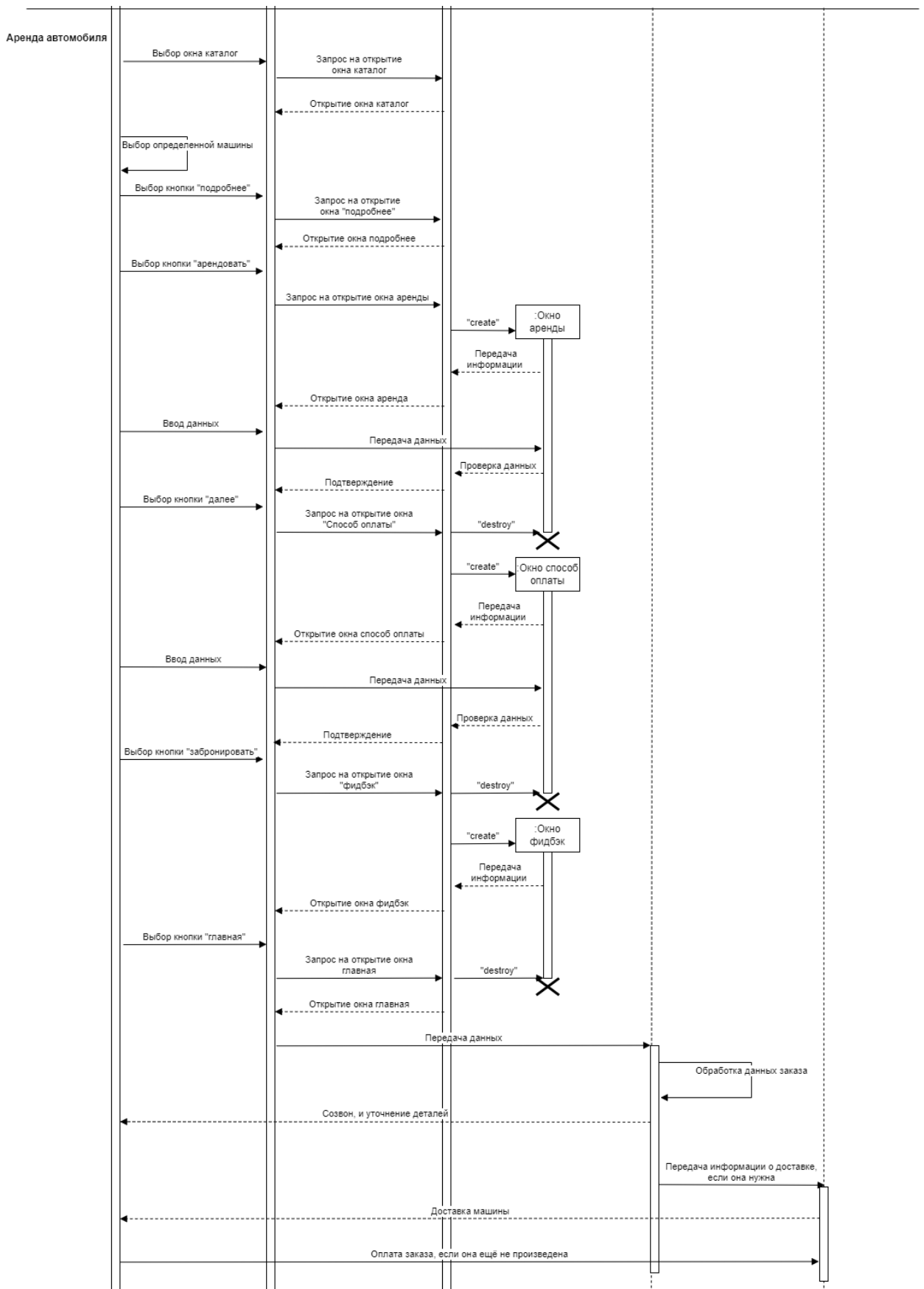


Рисунок 25 - Диаграмма последовательности. Аренда автомобиля

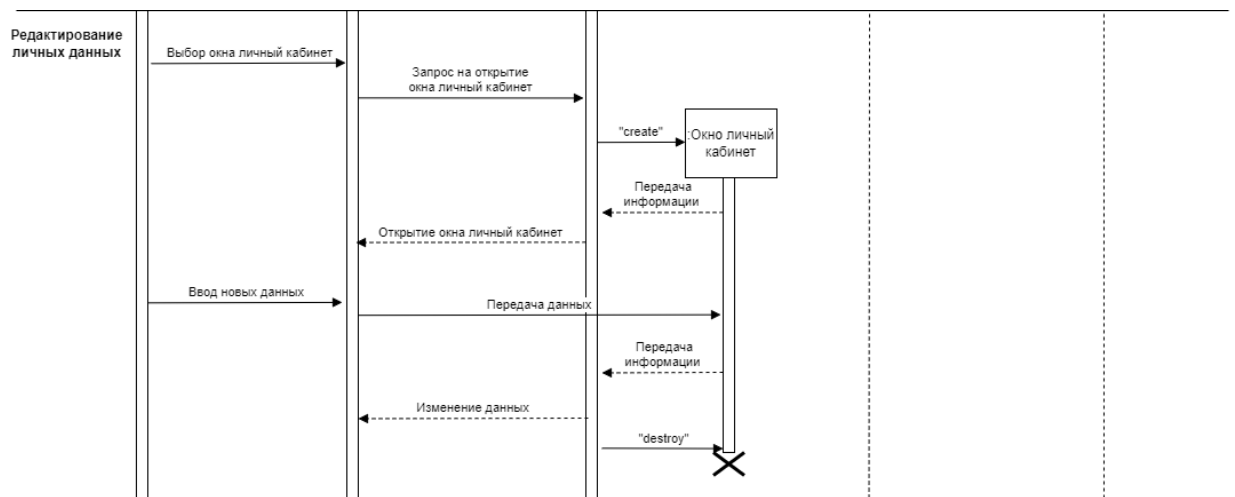


Рисунок 26 - Диаграмма последовательности. Редактирование личных данных

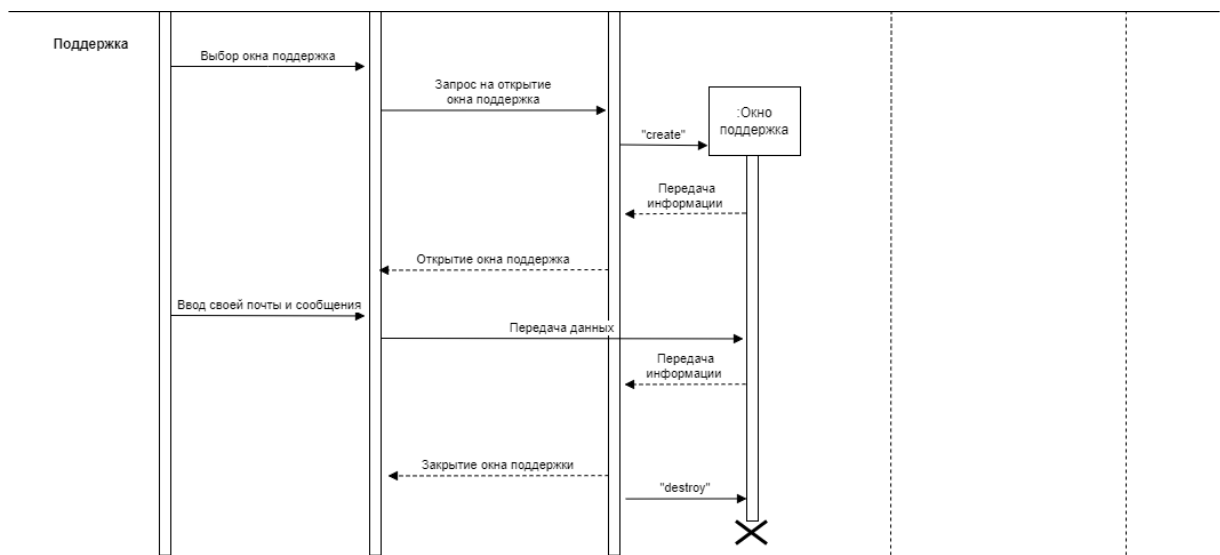


Рисунок 27 - Диаграмма последовательности. Поддержка

### 2.3.5 Диаграмма активностей

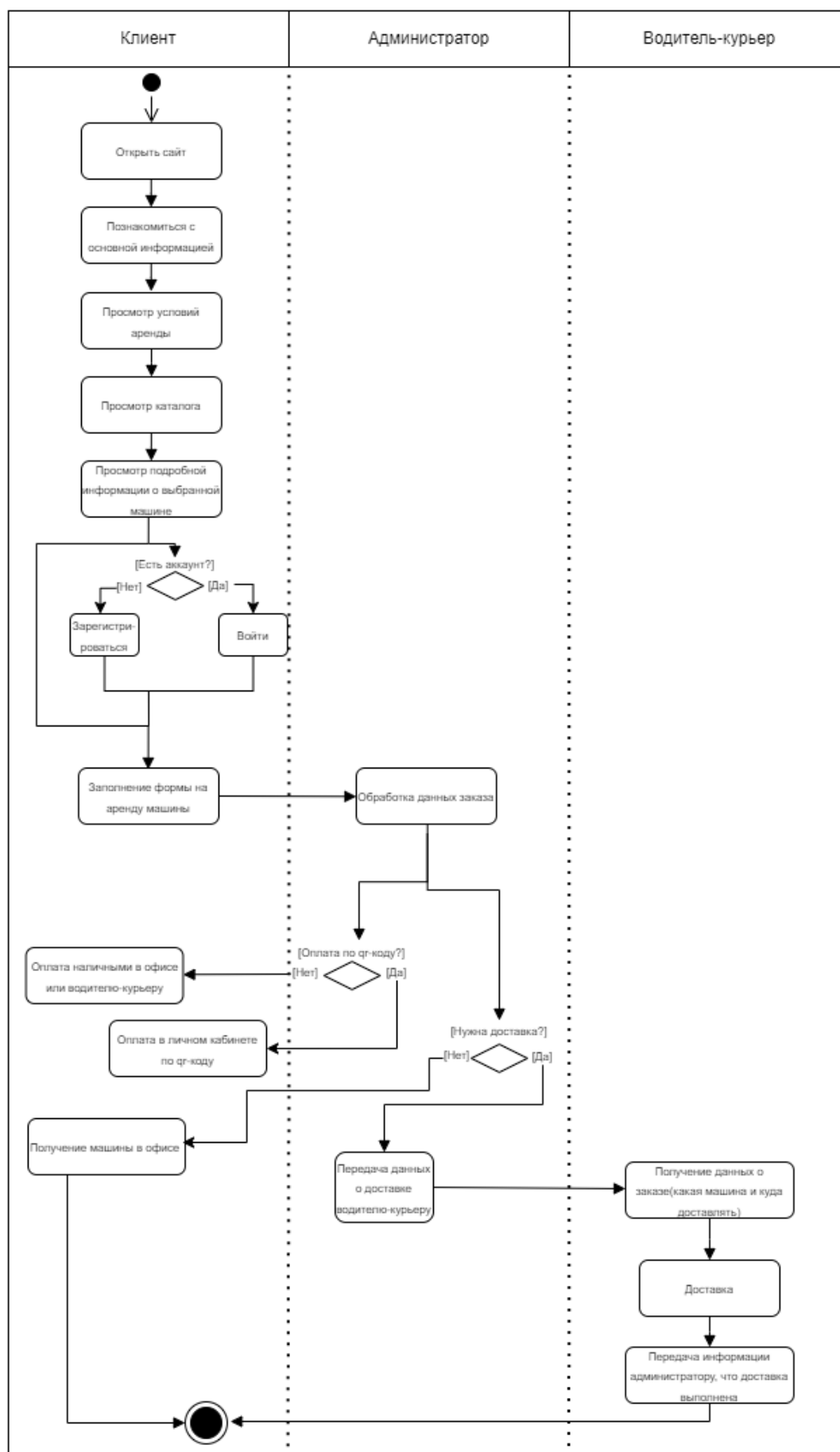


Рисунок 28 - Диаграмма активностей

### 2.3.6 Диаграмма классов

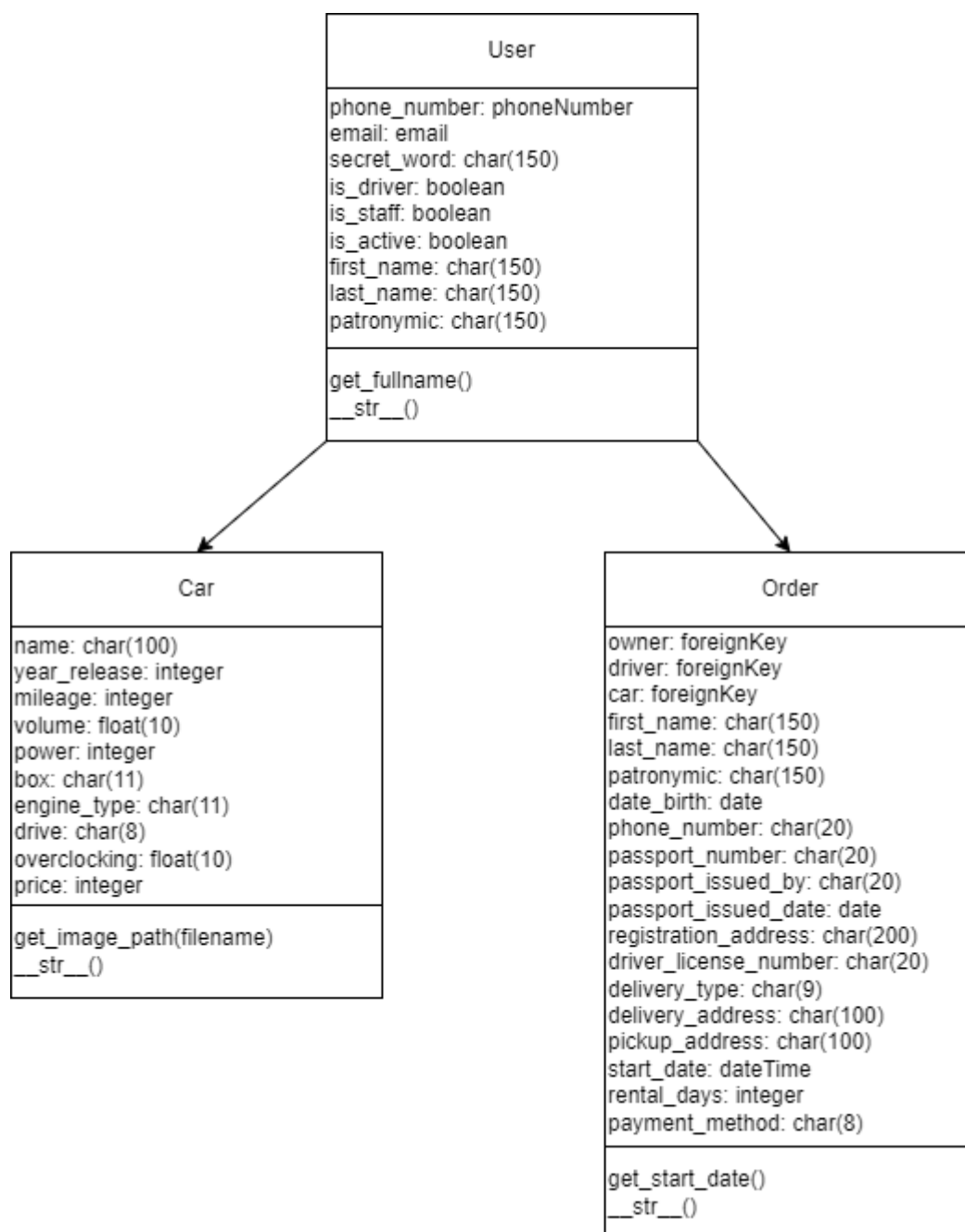


Рисунок 29 - Диаграмма классов

### 2.3.7 ER-диаграмма

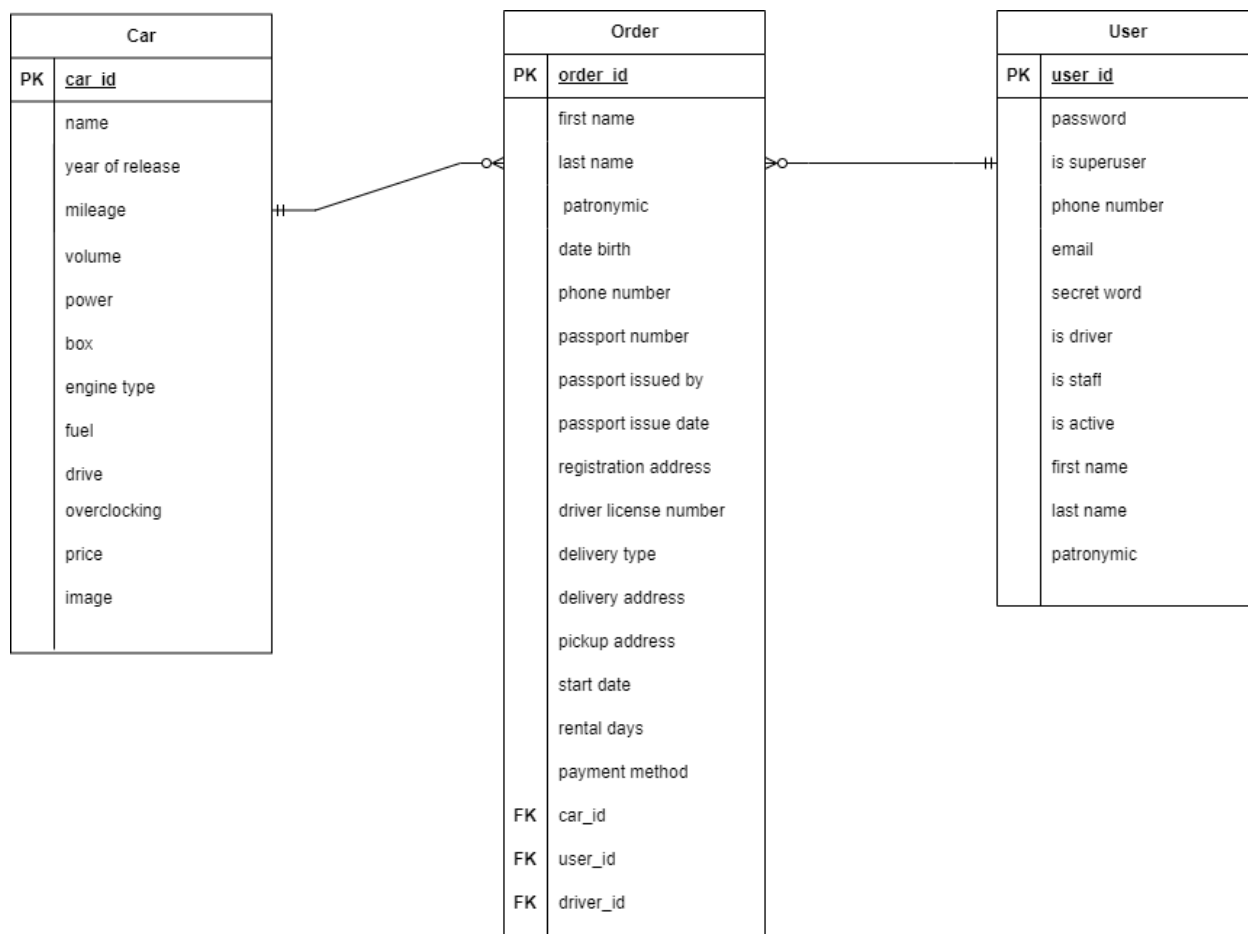


Рисунок 30 - ER-диаграмма

### 2.3.8 Диаграмма развёртывания

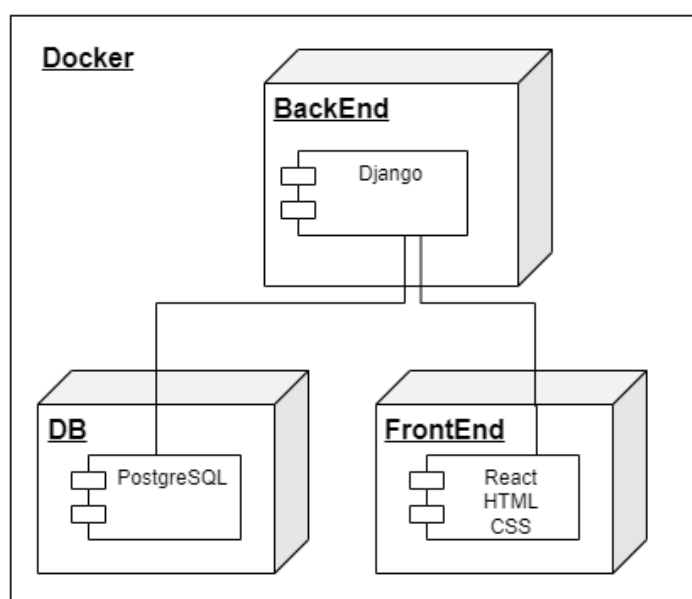


Рисунок 31 - Диаграмма развёртывания

### 2.3.9 Диаграмма сотрудничества

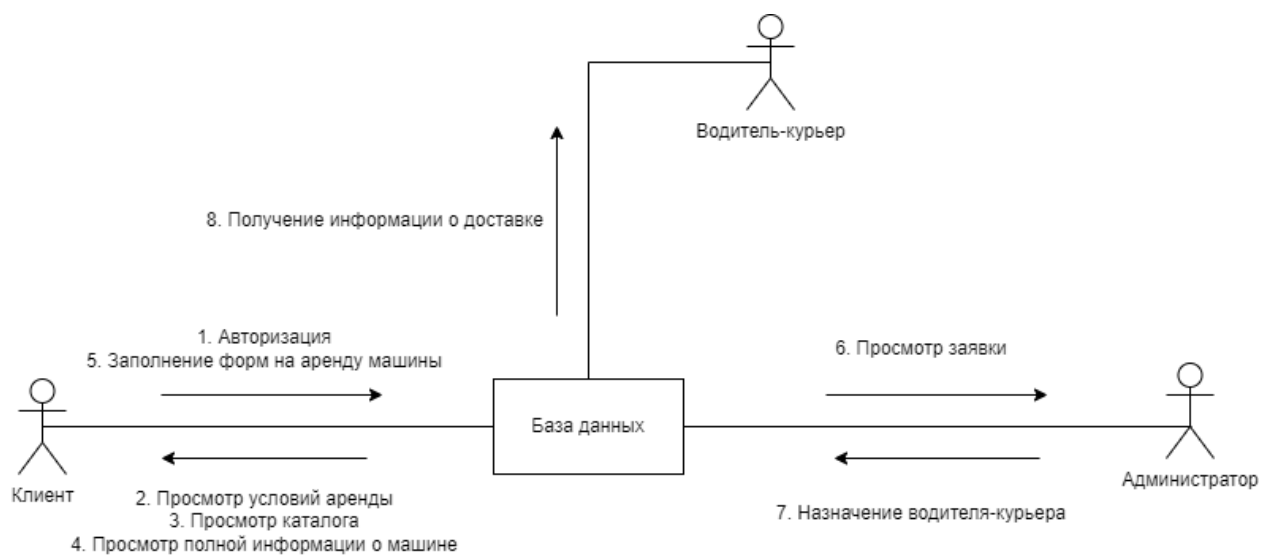


Рисунок 32 - Диаграмма сотрудничества

## 2.3.10 Диаграмма объектов

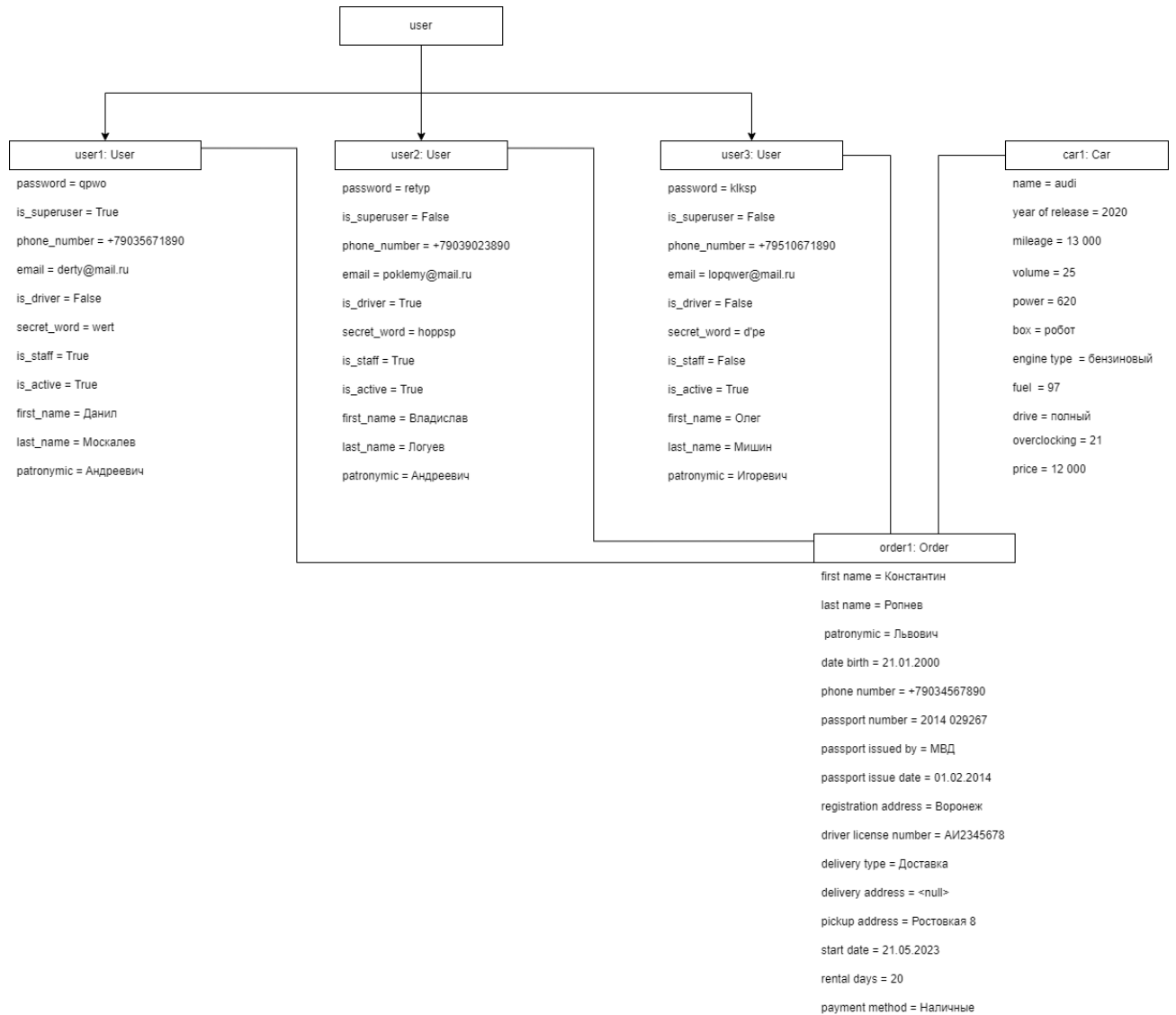


Рисунок 33 - Диаграмма объектов



## 2.4 Аналитика веб-приложения

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

На рисунке 34, рисунке 35, рисунке 36 показаны данные, собранные с сервиса «Яндекс.Метрика»:

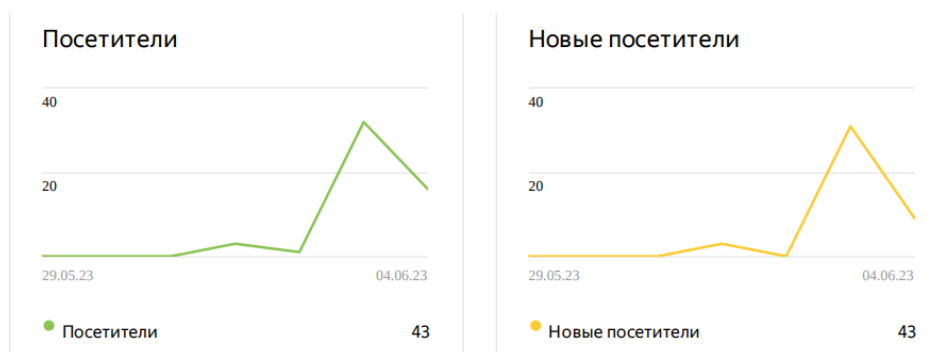


Рисунок 34 - Данные сервиса «Яндекс.Метрика»

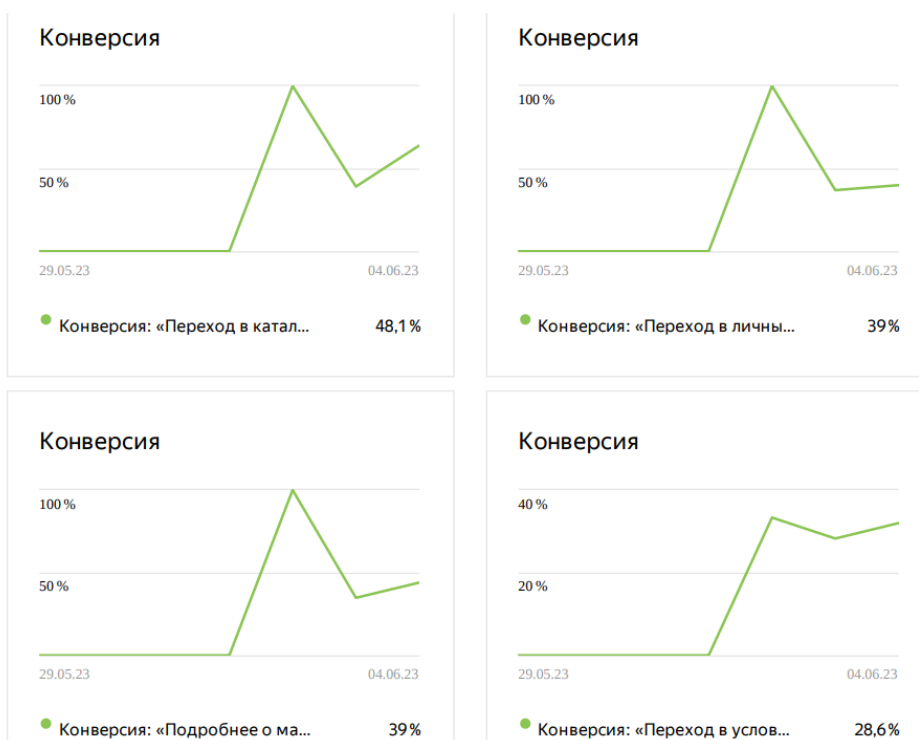


Рисунок 35 - Данные сервиса «Яндекс.Метрика»



Рисунок 36 - Данные сервиса «Яндекс.Метрика»

Была составлена воронка конверсии, которая позволяет увидеть использование пользователями основного сценария веб-приложения – аренда автомобиля. Результаты воронки конверсии приведены на рисунке 37.

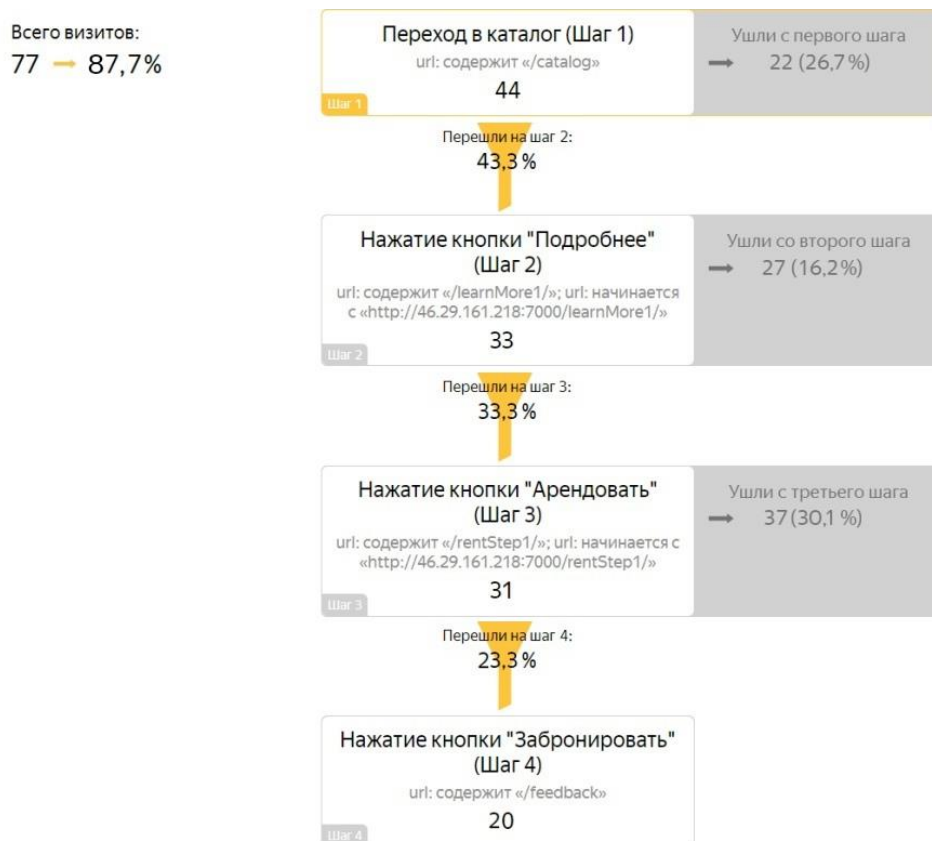


Рисунок 37 - Воронки конверсии

## 2.5 Границы проекта

Основными границами проекта являются:

- осуществление аренды автомобилей посредством заполнения и отправки заявок на аренду выбранной машины;
- осуществление всех необходимых уточнений или изменений, связанных с арендой автомобилей исключительно с администратором по телефонному звонку и только после оформления заявки на аренду;
- исключение возможности изменения условий аренды автомобилей;
- исключение возможности оплаты аренды машин иначе, чем с помощью наличной оплаты или благодаря QR-коду;
- исключение возможности расширения пунктов дополнительной информации о представленных в веб-приложении автомобилях;
- ограниченный бюджет на разработку и продвижение проекта;
- ограниченный выбор автомобилей в начальной стадии проекта;
- ограниченные сроки разработки и запуска проекта.

На фоне основных сценариев и функциональных требований, можно сформулировать то, что точно выходит за основные границы проекта:

- разработка и производство автомобилей;
- продажа автомобилей;
- обслуживание и ремонт автомобилей;
- аренда другой техники, кроме автомобилей;
- разработка и продажа программного обеспечения для автомобильных компаний;
- разработка и продажа оборудования для автомобильных компаний;
- организация автосервисов и автомойки;
- продажа запчастей и аксессуаров для автомобилей;
- организация автоспортивных мероприятий и треков.

## 3 Реализация

### 3.1 Средства реализации

Веб-приложение имеет архитектуру, соответствующую шаблону клиент-серверного приложения и разделяется на back-end и front-end посредством REST API.

Данная архитектура веб-приложения соотносится с основными требованиями к проекту, а именно:

- разделение пользователей на: клиентов, администраторов и водителей-курьеров;
- отправка пользователем писем на почту администрации веб-приложения;
- обеспечение формы для аренды машины с возможностью доставки выбранной машины по указанному клиентом адресу;
- оплата по QR-коду;
- авторизация пользователя;
- восстановление доступа к аккаунту пользователя.

Для реализации программно-аппаратной части были выбраны следующие технологии:

- язык программирования Python[4];
- фреймворк Django[5];
- СУБД PostgreSQL[6].

Для реализации клиентской части были выбраны следующие технологии:

- язык программирования JavaScript[7];
- язык стилей CSS[8];
- фреймворк React[7];
- язык разметки HTML[9].

### **3.2 Реализация серверной части веб-приложения**

Для осуществления основных сценариев веб-приложения: аренды автомобиля и оплаты аренды, необходимы регистрация, авторизация и аутентификация.

При регистрации пользователь указывает заданные для ввода параметры. После регистрации пользователю выдается роль “Клиент”.

Все основные настройки приложения хранятся в модуле settings. Авторизация, аутентификация реализована посредством сторонних библиотек simple-JWT и Djoser.

Клиент отправляет запрос, а сервер получает его с помощью модулей APIView из Django Rest Framework, которые обрабатывают полученные данные при помощи serializer-ов и отправляют данные обратно на клиент.

Модуль models хранит модели, соответствующие базе данных, которые организуют взаимодействие с БД.

Также, для удобства работы с сервером подключен swagger, в котором видны все ссылки.

### **3.3 Реализация клиентской части веб-приложения**

Для реализации основных сценариев веб-приложения, клиентская часть разработки делится постранично. Каждая страница описывается языком программирования JavaScript, языком разметки HTML и благодаря использованию фреймворка React. За реализацию заранее осуществленного и утвержденного командой разработчиков дизайна используется язык стилей CSS.

Архитектура разработки была организована согласно бизнес-логике проекта на основании модульного подхода, по которому все компоненты и логика находятся рядом друг с другом, а благодаря модулю для работы с файлами и их загрузкой через файл index.js экспортируется наружу все, что разрешено использовать.

Все страницы веб-приложения были реализованы и представлены командой разработчиков в соответствии с заявленным дизайном и соответствующими правками, внесенными в процессе разработки системы.

## **4 Тестирование**

### **4.1 Ручное UI–тестирование**

UI–тестирование[10] – этап комплексного тестирования программного обеспечения

В выборе варианта UI–тестирования мы остановились на ручном, так как, данный вариант является наиболее эффективным, поскольку обеспечивает наилучшее покрытие тестами.

В работе были рассмотрены следующие аспекты данного тестирования:

- визуальные характеристики – проверка адаптивности;
- композиция – проверка корректного отображения компонентов на странице;
- взаимодействие – обрабатываются ли события так, как задумано: CTR , скроллинг, тестирование input- и readonly-полей;
- доступность – проверка корректного расположения компонентов на странице и возможность правильного взаимодействия с ними;
- пользовательские потоки – проверка полного прохода по сценарию.

Для каждого аспекта (кейса) тестирования были введены критерии успешности прохождения UI–тестирования для той или иной страницы веб-приложения:

- 0 – тестируемая страница полностью не прошла тестирование;
- 1 – тестируемая страница частично не прошла тестирование;
- 2 – тестируемая страница полностью прошла тестирование.

Тестирование веб–страниц для сценария основных функций веб-приложения приведено в соответствии с таблицей 1.

Таблица 1 -Тестирование основных функций веб-приложения

Кейс	Главная страница	Страница условий аренды	Страница поддержки
Визуальные характеристики	1	1	1
Композиция	2	2	2
Взаимодействие	2	2	2
Доступность	2	2	2
Пользовательские потоки	2	2	2

Тестирование веб–страниц для сценария оформления заявки на аренду автомобиля приведено в соответствии с таблицей 2 и таблицей 3, где:

- страница 1 – страница каталога;
- страница 2 – страница с подробной информацией о машине для неавторизованного пользователя;
- страница 3 – страница с подробной информацией о машине для авторизованного пользователя;
- страница 4 – страница аренды автомобиля 1;
- страница 5 – страница аренды автомобиля 2;
- страница 6 – фидбэк–страница.

Таблица 2 -Тестирование веб–страниц оформления заявки на аренду автомобиля

Кейс	Страница 1	Страница 2	Страница 3
Визуальные характеристики	1	1	1
Композиция	2	2	2
Взаимодействие	2	2	2
Доступность	2	2	2



Таблица 3 -Тестирование веб–страниц оформления заявки на аренду автомобиля

Кейс	Страница 4	Страница 5	Страница 5
Визуальные характеристики	1	1	1
Композиция	2	2	2
Взаимодействие	2	2	2
Доступность	2	2	2

Тестирование веб-страниц для сценария авторизации приведено в соответствии с таблицей 4 и таблицей 5, где:

- страница 1 – страница входа в ЛК;
- страница 2 – страница регистрации;
- страница 3 – страница ошибки входа в ЛК;
- страница 4 – страница сброса пароля;
- страница 5 – страница ввода нового пароля.

Таблица 4 -Тестирование веб-страниц авторизации

Кейс	Страница 1	Страница 2	Страница 3
Визуальные характеристики	1	1	1
Композиция	2	2	2
Взаимодействие	2	2	2
Доступность	2	2	2

Таблица 5 -Тестирование веб-страниц авторизации

Кейс	Страница 4	Страница 5
Визуальные характеристики	1	1
Композиция	2	2
Взаимодействие	2	2
Доступность	2	2

Тестирование веб–страниц для сценария личного кабинета клиента приведено в соответствии с таблицей 6 и таблицей 7, где:

- страница 1 – главная страница ЛК клиента;
- страница 2 – страница редактирования личных данных;
- страница 3 – страница истории заказов;
- страница 4 – страница текущего заказа;
- страница 5 – страница с QR-кодом;
- страница 6 – страница с “заглушкой”;
- страница 7 – страница с успешно прошедшей оплатой;
- страница 8 – страница с не успешно прошедшей оплатой.

Таблица 6 -Тестирование веб–страниц личного кабинета клиента

Кейс	Страница 1	Страница 2	Страница 3	Страница 4
Визуальные характеристики	1	1	1	1
Композиция	2	2	2	2
Взаимодействие	2	2	2	2
Доступность	2	2	2	2

Таблица 7 -Тестирование веб–страниц личного кабинета клиента

Кейс	Страница 5	Страница 6	Страница 7	Страница 8
Визуальные характеристики	1	1	1	1
Композиция	2	2	2	2
Взаимодействие	2	2	2	2
Доступность	2	2	2	2

Тестирование веб–страниц для сценария личного кабинета водителя-курьера приведено в соответствии с таблицей 8.

Таблица 8 -Тестирование веб–страниц личного кабинета водителя-курьера

Кейс	Главная страница ЛК водителя-курьера
Визуальные характеристики	1
Композиция	2
Взаимодействие	2
Доступность	2
Пользовательские потоки	2

Тестирование шапки веб–приложения приведено в соответствии с таблицей 9.

Таблица 9 -Тестирование шапки веб–приложения

Кейс	Шапка для неавторизованного пользователя	Шапка для авторизованного пользователя
Визуальные характеристики	1	1
Композиция	2	2
Взаимодействие	2	2
Доступность	2	2
Пользовательские потоки	2	2

## 4.2 Юзабилити–тестирование

Юзабилити–тестирование[11] является исследованием, выполняемым с целью определения, удобен ли некоторый искусственный объект для его предполагаемого применения.

Объектом тестирования является веб–приложение автопарка с возможностью онлайн бронирования и оплаты для получения информации о корректной работе приложения для пользователя:

- порога вхождения;
- прохождение сценариев;

- степени ориентированности в веб-приложении во время прохождения сценариев.

Первым сценарием для тестирования:

- сценарий 1 «Основные функции веб-приложения»;
- сценарий 2 «Авторизация»;
- сценарий 3 «Аренда автомобиля»;
- сценарий 4 «Оплата заказа»;
- страница 5 – страница с QR-кодом;
- страница 6 – страница с “заглушкой”;
- страница 7 – страница с успешно прошедшей оплатой.

Для каждой единицы информации были введены критерии успешности выполнения того или иного задания:

- 0 – респондент не справился с заданием;
- 1 – респондент справился с заданием со значительными затруднениями;
- 2 – респондент справился с заданием и перешел на следующий шаг.

Респонденты, отобранные для тестирования, были разделены по возрасту и полу. Тем самым в исследовании приняли участие:

- двое молодых людей возраста двадцати лет;
- двое молодых девушек возраста двадцати лет;
- двое мужчин с средним возрастом в 50 лет;
- двое женщин с средним возрастом в 50 лет.

Данное распределение для тестирования было выбрано с целью выявления преимуществ и недостатков в уже разработанном веб-приложении при работе с ним разных лиц пользователей.

Результаты юзабилити-тестирования переведены в соответствии с таблицей 10, таблицей 11 и таблицей 12.

Таблица 10 -Результаты юзабилити–тестирования

Как Вас зовут?	Ваш пол?	Сколько Вам лет?	Использовал и ли вы раньше приложения для аренды машин на какой-либо период времени?	Если Вы ответили утвердительно на предыдущий вопрос, оцените удовлетворенность использованием такого приложения:
Дмитрий	Мужской	18-25 лет	Нет	—
Данил	Мужской	18-25 лет	Нет	—
Глафира	Женский	18-25 лет	Да	0
Екатерина	Женский	18-25 лет	Нет	—
Алексей	Мужской	40+ лет	Нет	—
Николай	Мужской	40+ лет	Да	1
Светлана	Женский	40+ лет	Нет	—
Софья	Женский	40+ лет	Да	2

Таблица 11 -Результаты юзабилити–тестирования

Сценарий 1 «Основные функции веб-приложения»	Сценарий 2 «Авторизация»	Сценарий 3 «Аренда автомобиля»	Сценарий 4 «Оплата заказа»	Сценарий 5 «Пользование Личным Кабинетом»
2	2	2	2	2
2	2	2	2	2
2	2	2	2	2
2	2	2	2	2
2	2	2	2	2
2	2	1	1	1
2	2	2	2	2
2	2	2	2	2

Таблица 12 -Результаты юзабилити–тестирования

Как Вас зовут?	Оцените общее впечатление от работы с веб-приложением автопарка «РМК»:
Дмитрий	2
Данил	2
Глафира	2
Екатерина	2
Алексей	2
Николай	2
Светлана	2
Софья	2

## Заключение

В ходе выполнения курсового проекта командой было разработано веб-приложение автопарка с возможностью онлайн бронирования и оплаты, соответствующее поставленным перед проектом задачам.

В начале разработки был проведен анализ предметной области, определены основные требования к разрабатываемой системе, определены основные сценарии веб-приложения и пользовательские истории.

По результатам разработки проводился ряд тестов с целью проверки работоспособности системы.

В процессе работы были реализованы следующие задачи проекта:

- разделение пользователей на: клиентов, администраторов и водителей-курьеров;
- обеспечение возможности отправки пользователем писем на почту администрации веб-приложения;
- обеспечение формы для аренды машины с возможностью доставки выбранной машины по указанному клиентом адресу;
- обеспечение возможности оплаты по QR-коду;
- обеспечение возможности авторизации пользователя;
- обеспечение возможности восстановления доступа к аккаунту пользователя.

## Список используемых источников

1. Car2you [Электронный ресурс]. Режим доступа: <https://autoprokatru.ru/voronez/> - Заглавие с экрана. (Дата обращения: 26.03.2023).
2. M4 [Электронный ресурс]. Режим доступа: <https://воронеж.аренда-авто-м4.рф/> - Заглавие с экрана. (Дата обращения: 26.03.2023).
3. Rent Cars [Электронный ресурс]. Режим доступа: <https://rentcars.ru> - Заглавие с экрана. - (Дата обращения: 26.03.2023).
4. Python 3.9.16 documentation [Электронный ресурс]. – Режим доступа: <https://docs.python.org/3.9/>
5. Django: The web framework for perfectionists with deadlines [Электронный ресурс]. – Режим доступа: <https://docs.djangoproject.com/en/4.2/>.
6. PostgreSQL. Основы языка SQL: учеб. пособие / Е. П. Моргунов; под ред. Е. В. Рогова, П. В. Лузанова. — СПб.: БХВ-Петербург, 2018. — 336 с.
7. Документация JavaScript-библиотеки React.js [Электронный ресурс]. Режим доступа: <https://ru.react.js.org/docs/getting-started.html>
8. Документация Sass - метаязыка на основе CSS [Электронный ресурс]. Режим доступа: <https://sass-lang.com/documentation/>
9. Основы HTML [Электронный ресурс] Режим доступа: <https://html5book.ru/osnovy-html/>
10. Ручное UI-тестирование [Электронный ресурс]. Режим доступа: <https://ux-journal.ru/kak-provodit-ui-testirovanie-sravnenie-instrumentov.html>
11. Юзабилити-тестирование [Электронный ресурс]. Режим доступа: <https://lpgenerator.ru/blog/2016/07/28/yuzabiliti-testirovanie-poshagovaya-instrukciya-na-primere-yelp/>