

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

Факультет компьютерных наук

Кафедра информационных технологий управления

Веб-приложение 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 _____.20__
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

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	9
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	16
2.3 Моделирование системы.....	19
2.3.1 Диаграмма в стиле методологии IDEF0	19
2.3.2 Диаграмма прецедентов	22
2.3.3 Диаграмма состояний	23
2.3.4 Диаграмма последовательности.....	26
2.3.5 Диаграмма активностей.....	29
2.3.6 Диаграмма классов.....	31
2.3.7 ER-диаграмма.....	32
2.3.8 Диаграмма развёртывания.....	32

2.3.9 Диаграмма сотрудничества	33
2.3.10 Диаграмма объектов	34
2.4 Продуктовые воронки	35
2.5 Границы проекта.....	35
3 Реализация	37
3.1 Средства реализации	37
3.2 Реализация серверной части веб-приложения	37
3.3 Реализация клиентской части веб-приложения	38
4 Тестирование	39
4.1 Ручное UI-тестирование	39
4.2 Юзабилити тестирование	39
4.3 Тестирование сервера и его взаимосвязи с клиентом	39
ЗАКЛЮЧЕНИЕ	40
Список используемых источников.....	41

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, в первую очередь, 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, введенной на этапе регистрации;
- 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 одобрять/удалять конкретную заявку на аренду машины от пользователей;
- Обеспечение возможности просматривать информацию о клиентах;
- Обеспечение возможности просматривать уже одобренные заявки на аренду машин;
- 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 логина и 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. Сайт должен быть оптимизирован для изменения размеров экрана.

1.3 Задачи, решаемые в процессе разработки

Перед проектом были поставлены следующие задачи:

- Анализ предметной области;
- Анализ аналогов;
- Написание технического задания;
- Проектирование веб-сервиса средствами языка UML;
- Построение use case диаграмм;
- Создание доски Trello и репозитория GitHub;
- Определение используемой платформы;
- Подключение внешнего модуля для хранения данных;
- 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 и список принятых заявок, 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, данные которого были заранее занесены в систему под отличительным доменом.

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

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

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

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.

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

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

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

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 безопасные и поддерживаемые 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.

Хлебные крошки (навигационная цепочка) – элемент навигации сайта, показывающий путь в структуре ресурса от главной страницы к текущей, на которой в данный момент находится пользователь.

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 – 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 представлен на рисунке 1.

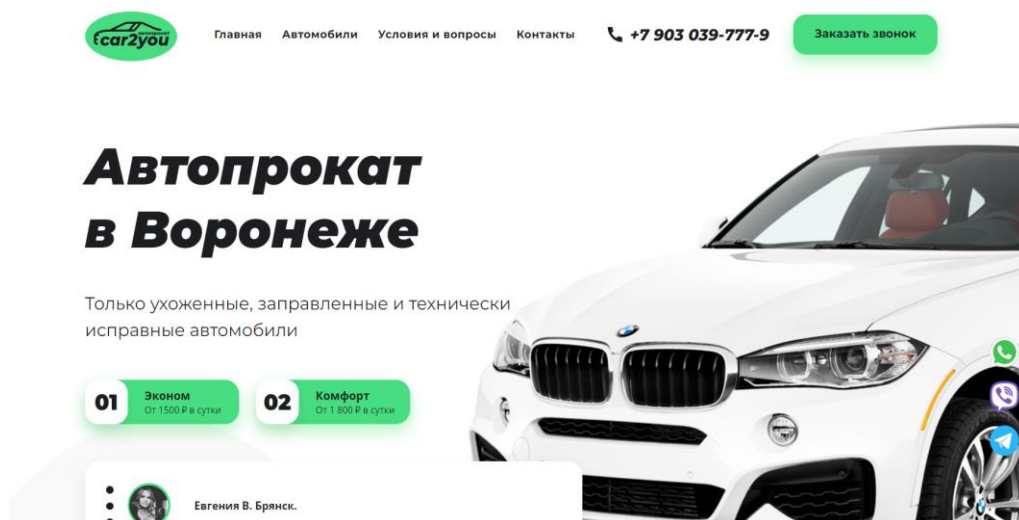


Рисунок 1 - Интерфейс сайта «car2you»

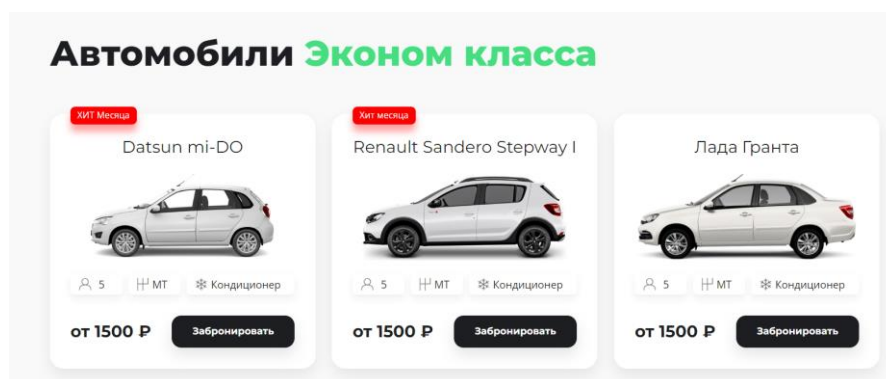


Рисунок 2 - Каталог эконом класса



Рисунок 3 - Каталог комфорт класса

Недостатками сайта Car2you являются:

- Шапка главной страницы неподвижна;
- Отсутствует поддержка непосредственно на сайте.

2.2.2 Авто М4

Автопрокат М4 – на данном сайте для аренды представлены модели от эконома до премиум класса, промышленный автотранспорт, а так же разнообразное промышленное оборудование.

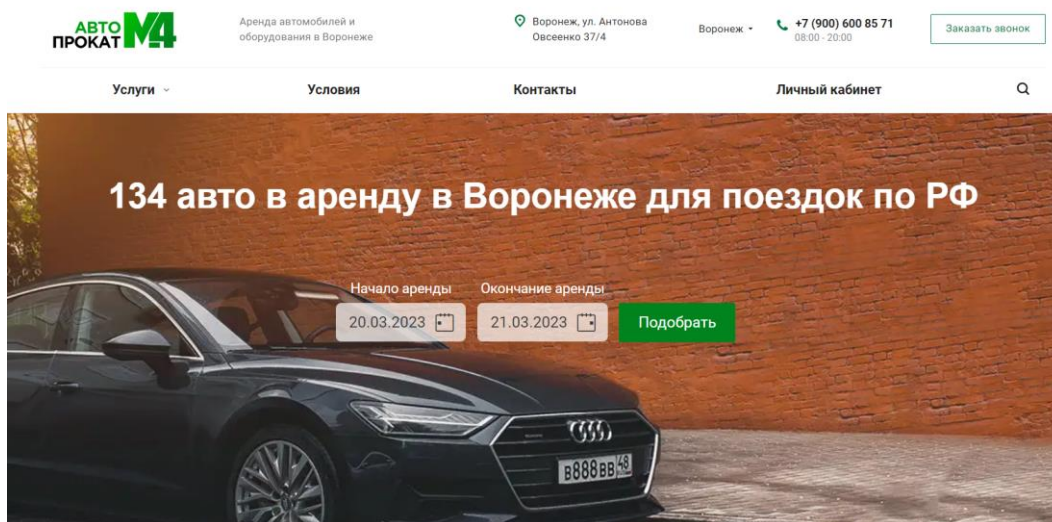


Рисунок 4 - Интерфейс сайта «Авто М4»

Из плюсов стоит отметить: большой выбор автомобилей (Рисунок 5), возможность изменить формат просмотра каталога (Рисунок 6).

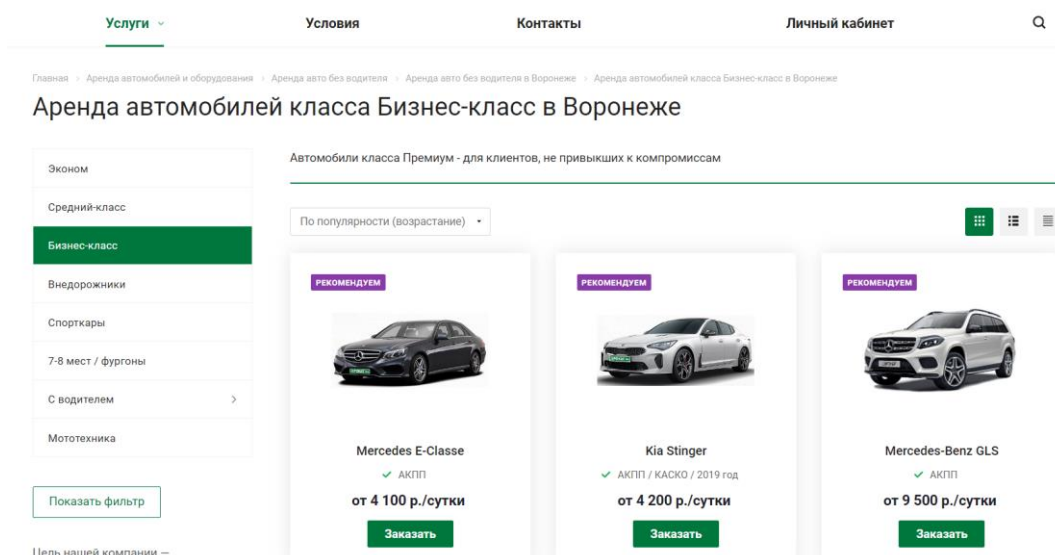


Рисунок 5 - Каталог «Авто М4»

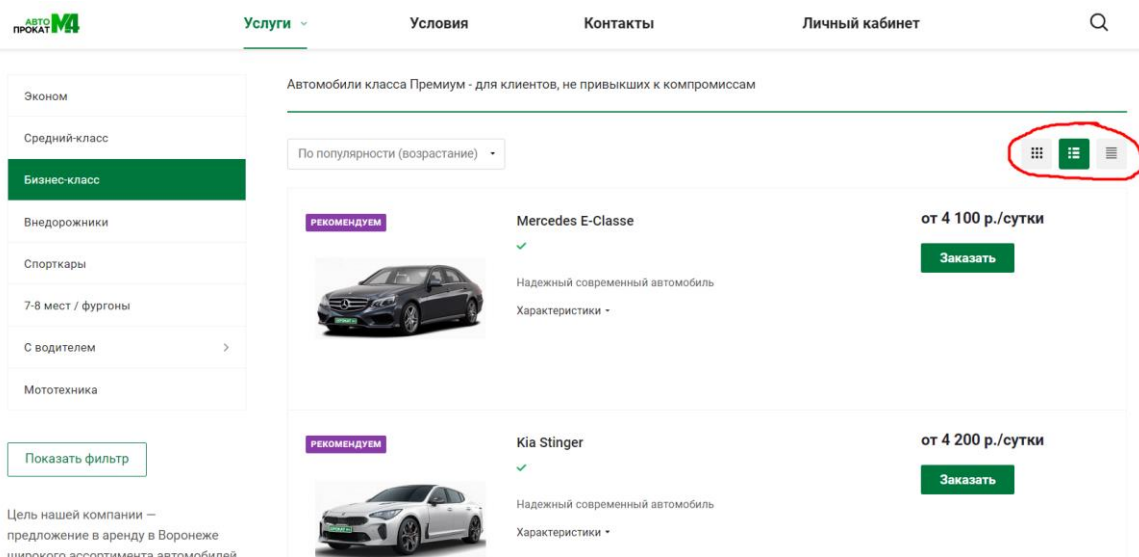


Рисунок 6 - Выбор просмотра страницы

Так же стоит отметить реализацию «хлебных крошек», данная функция помогает ориентироваться пользователю по сайту (Рисунок 7).

Главная > Аренда автомобилей и оборудования > Аренда авто без водителя > Аренда авто без водителя в Воронеже > Аренда автомобилей кроссоверов и джипов в Воронеже

Рисунок 7 - Навигационная цепочка

Недостатками сайта Авто М4 являются:

— Отсутствует поддержка именно на сайте;

— Большой объём текста, главные моменты плохо выделены (Рисунок 8).

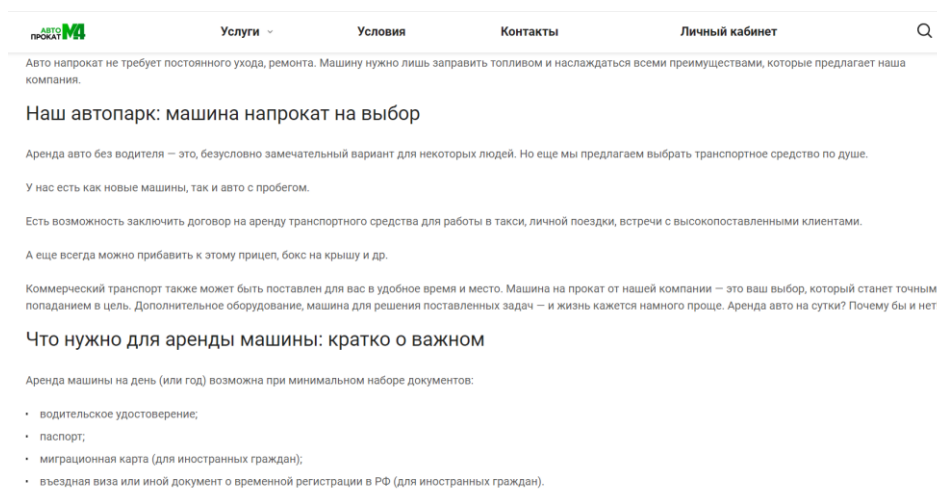


Рисунок 8 - Информация об условиях аренды

2.2.3 RentCars

RentCars — аренда автомобилей от класса эконом, до класса внедорожник. На главной странице представлен каталог с большим выбором как новых, так и машин с пробегом. В каталоге можно сортировать машины по классам, выбрав одну или несколько позиций (Рисунок 10). На данном сайте работает служба технической помощи на дорогах для клиентов. Так же стоит отметить большой плюс — это возможность перевести сайт на английский язык.

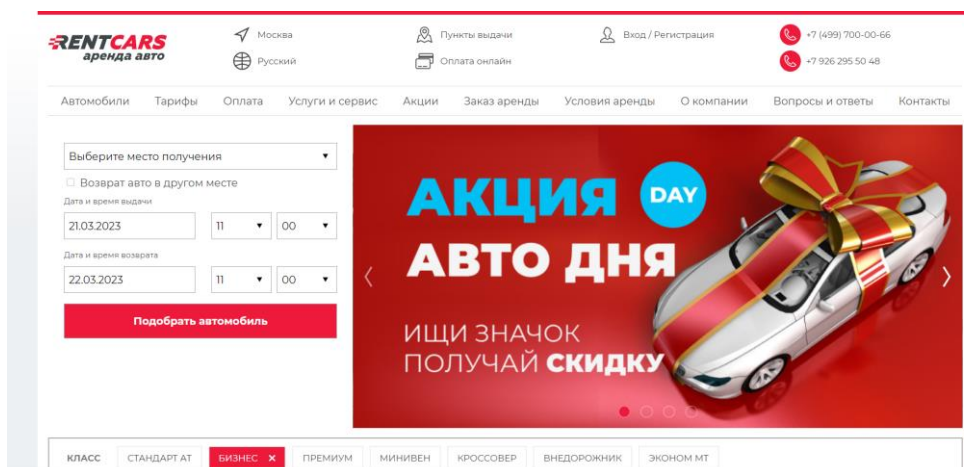


Рисунок 9 - Интерфейс главной страницы «RentCars»

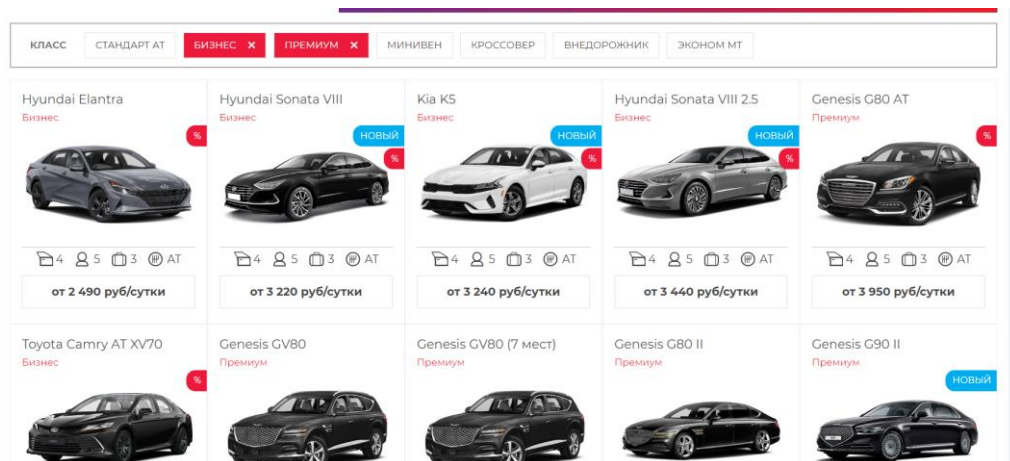


Рисунок 10 - Каталог «RentCars»

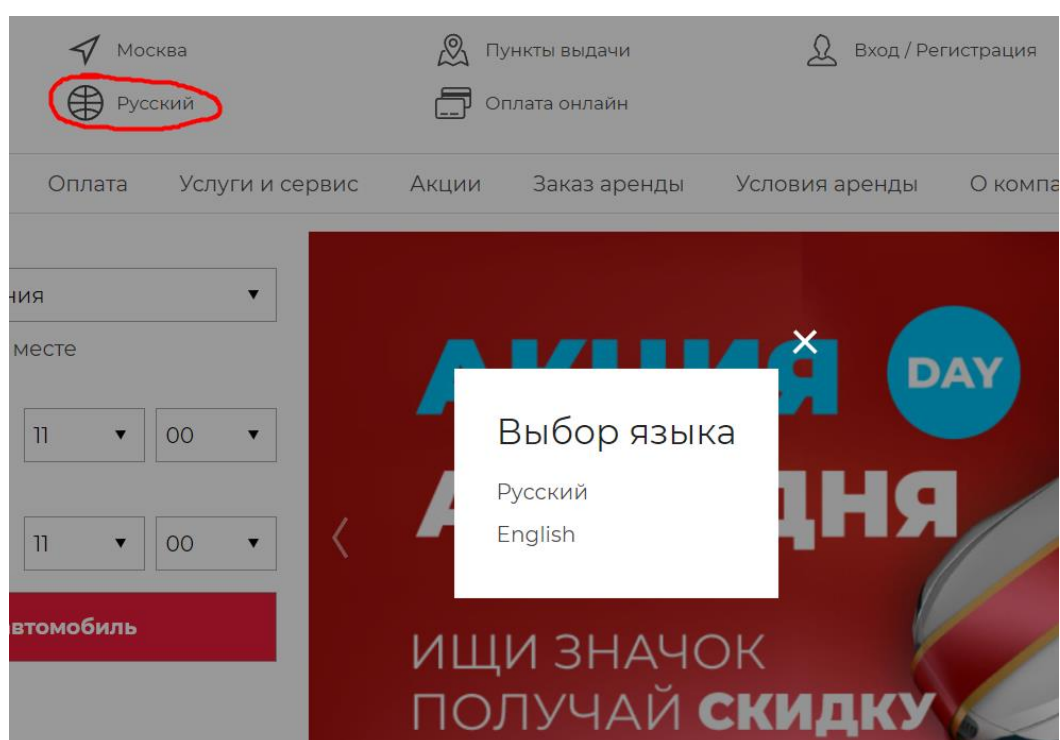


Рисунок 11 - Выбор языка «RentCars»

При выборе определённой модели открывается страница с подробным описанием машины, требованием к арендатору, и видео обзором.

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 12 - 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. Монуменальный. 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 13 - Характеристики автомобиля «RentCars»

Минусы сайта «RentCars»:

- Неподвижная шапка на всех страницах сайта;
- Отсутствие поддержки пользователей на сайте.

2.3 Моделирование системы

2.3.1 Диаграмма в стиле методологии IDEF0

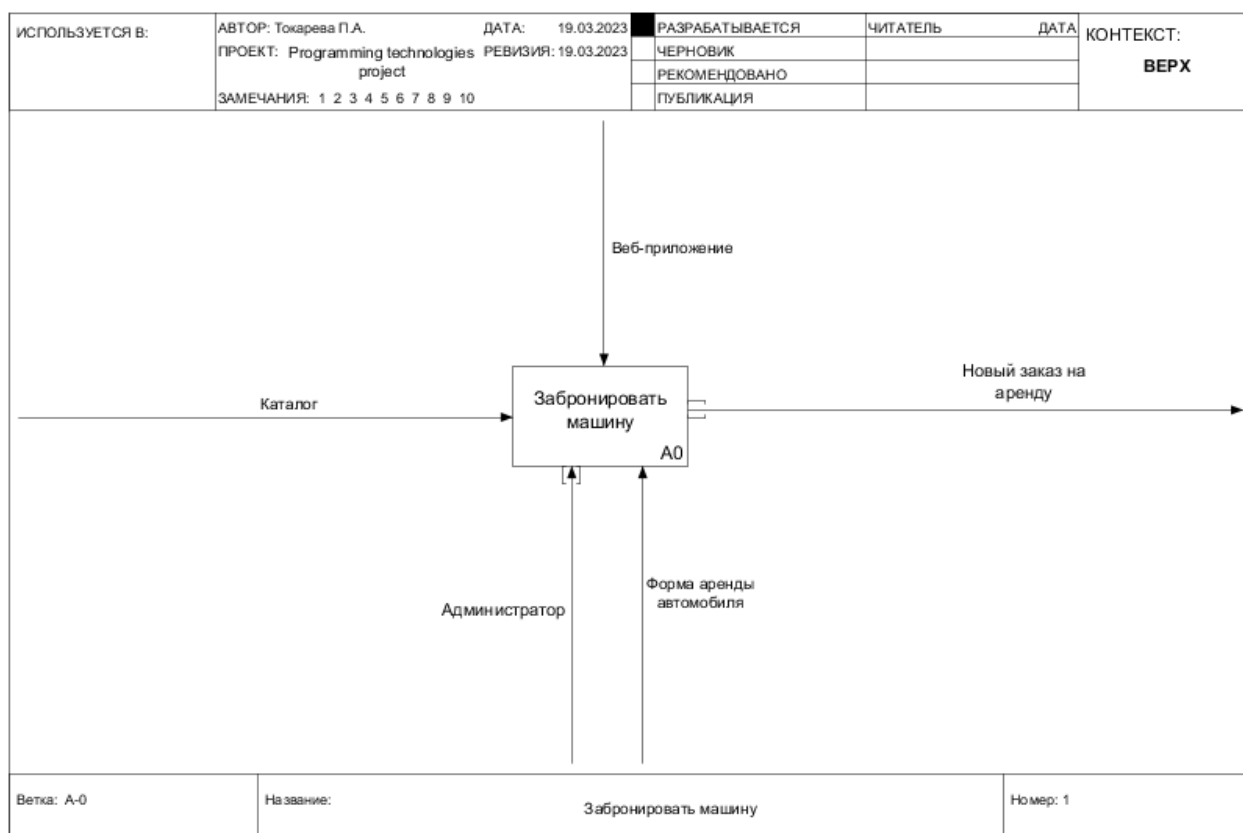


Рисунок 14 - Контекстная диаграмма со стороны Клиента

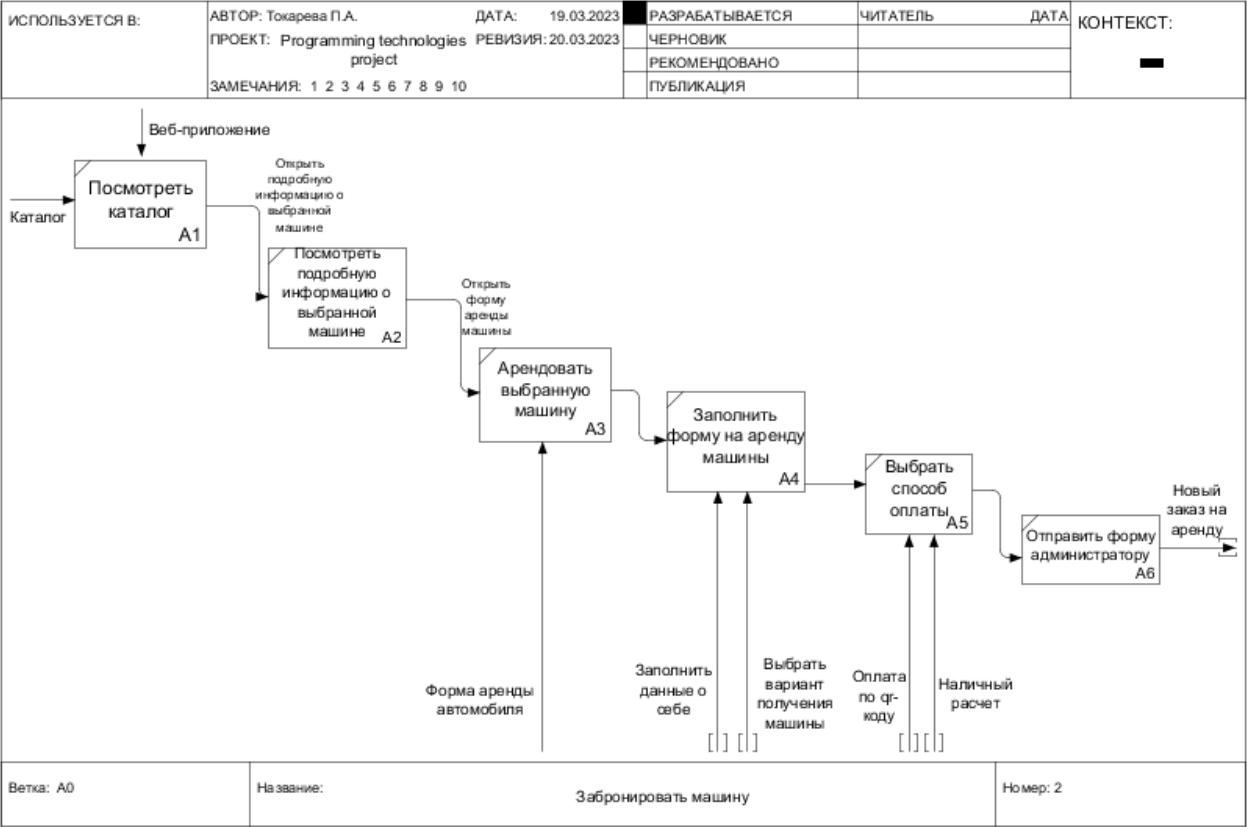


Рисунок 15 - Дочерняя диаграмма со стороны Клиента

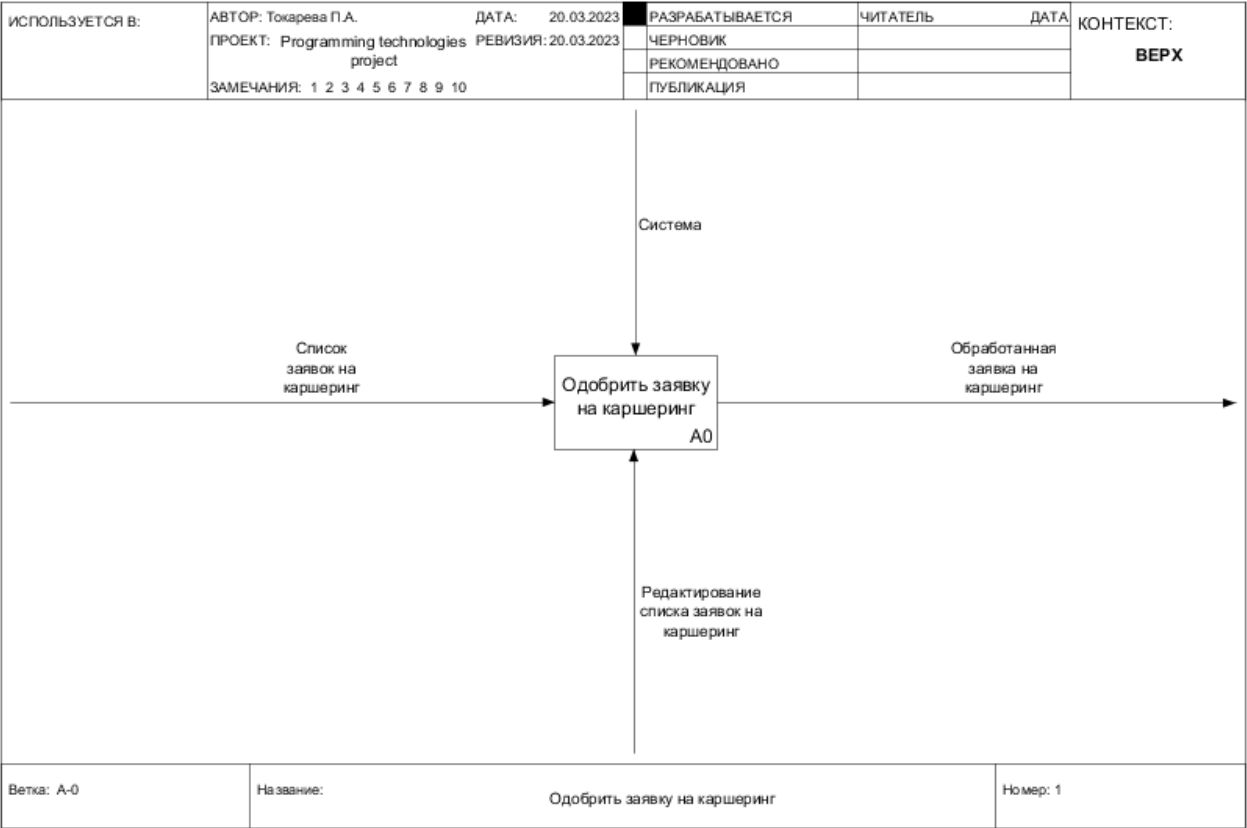


Рисунок 16 - Контекстная диаграмма со стороны Администратора

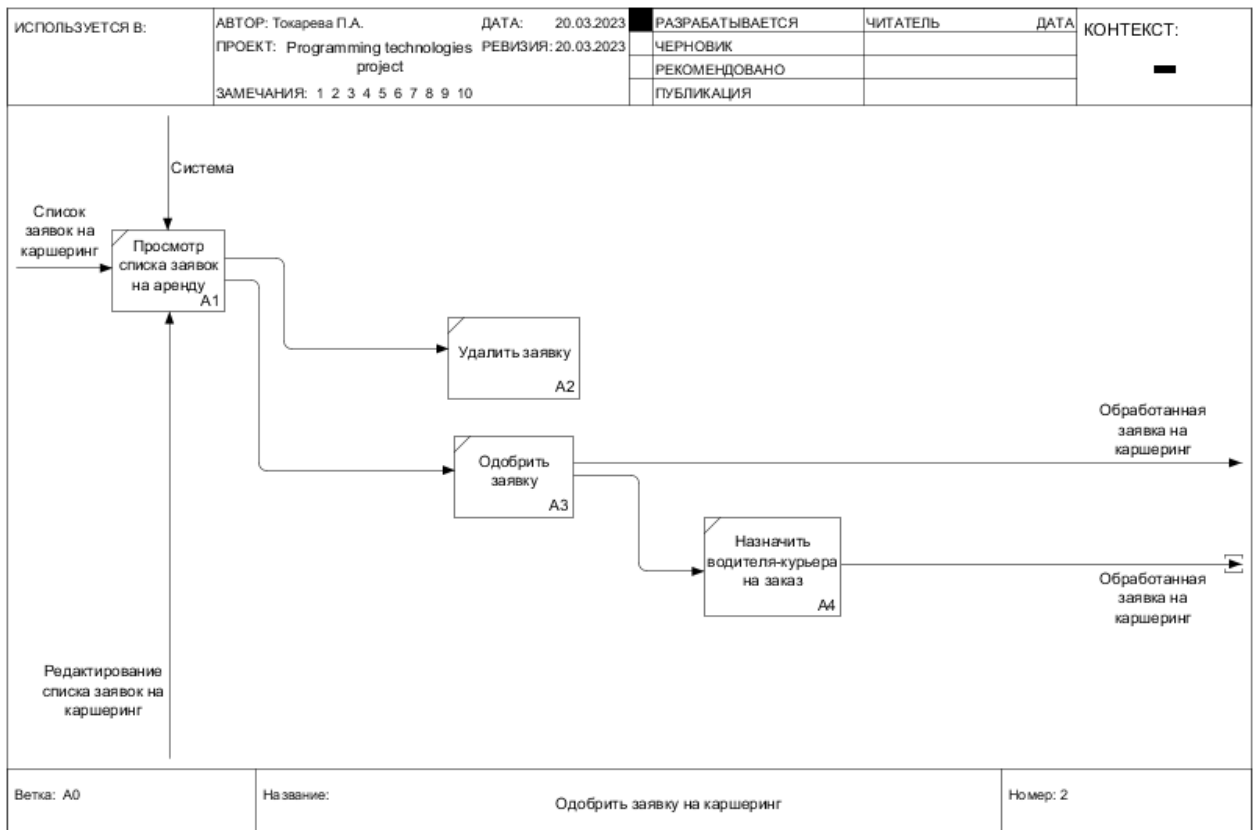


Рисунок 17 - Дочерняя диаграмма со стороны Администратора

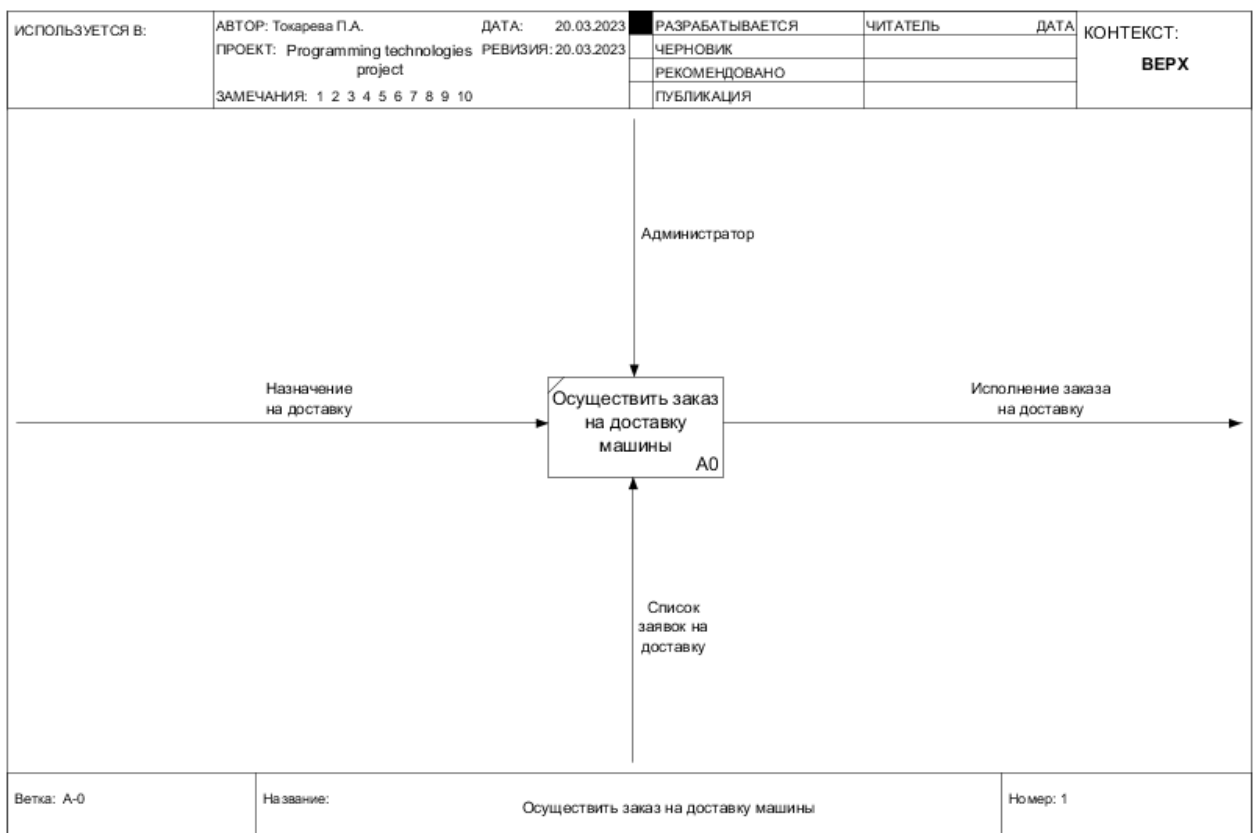


Рисунок 18 - Контекстная диаграмма со стороны Водителя-курьера

2.3.2 Диаграмма прецедентов

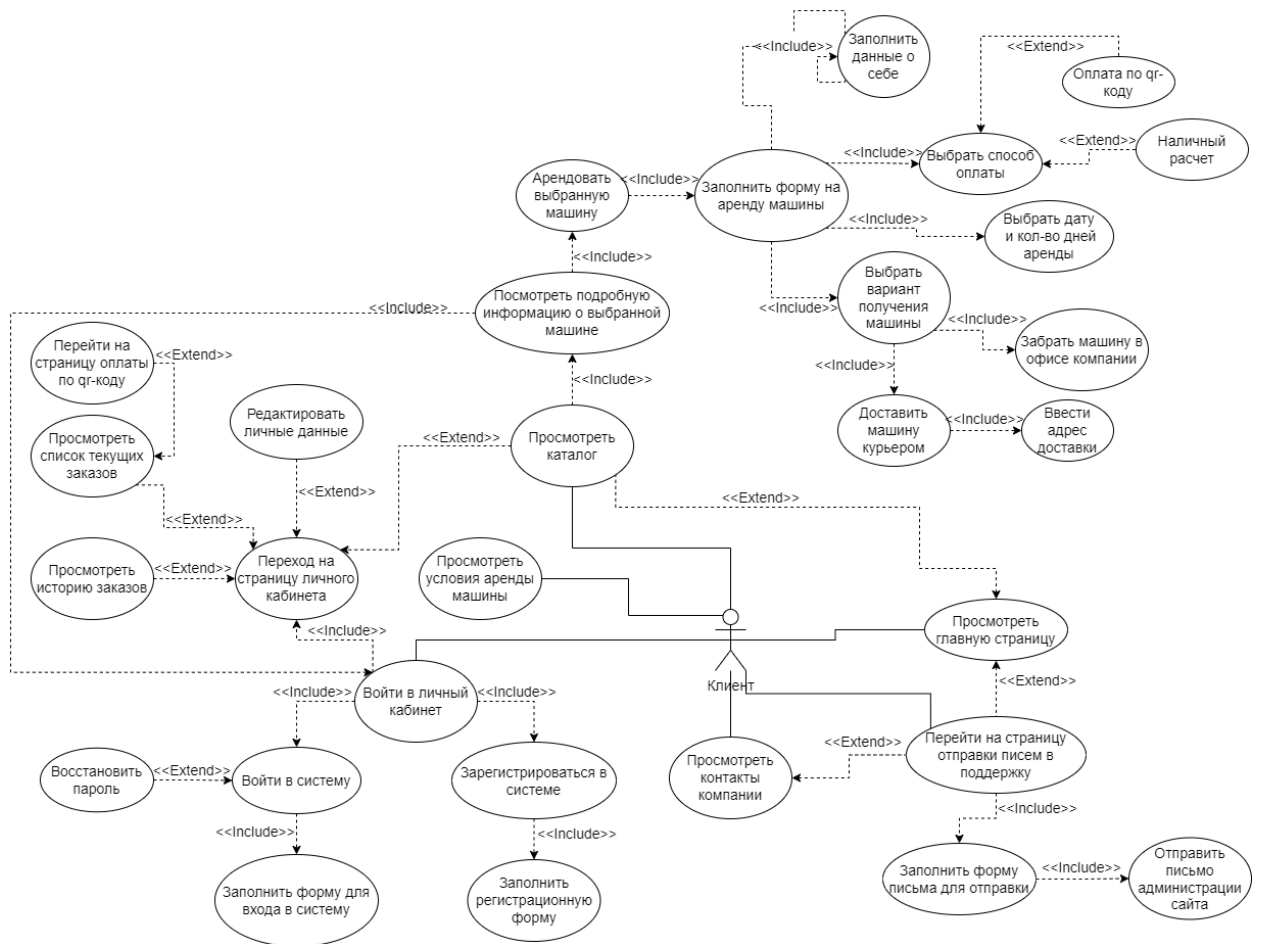


Рисунок 19 - Диаграмма прецедентов со стороны Клиента

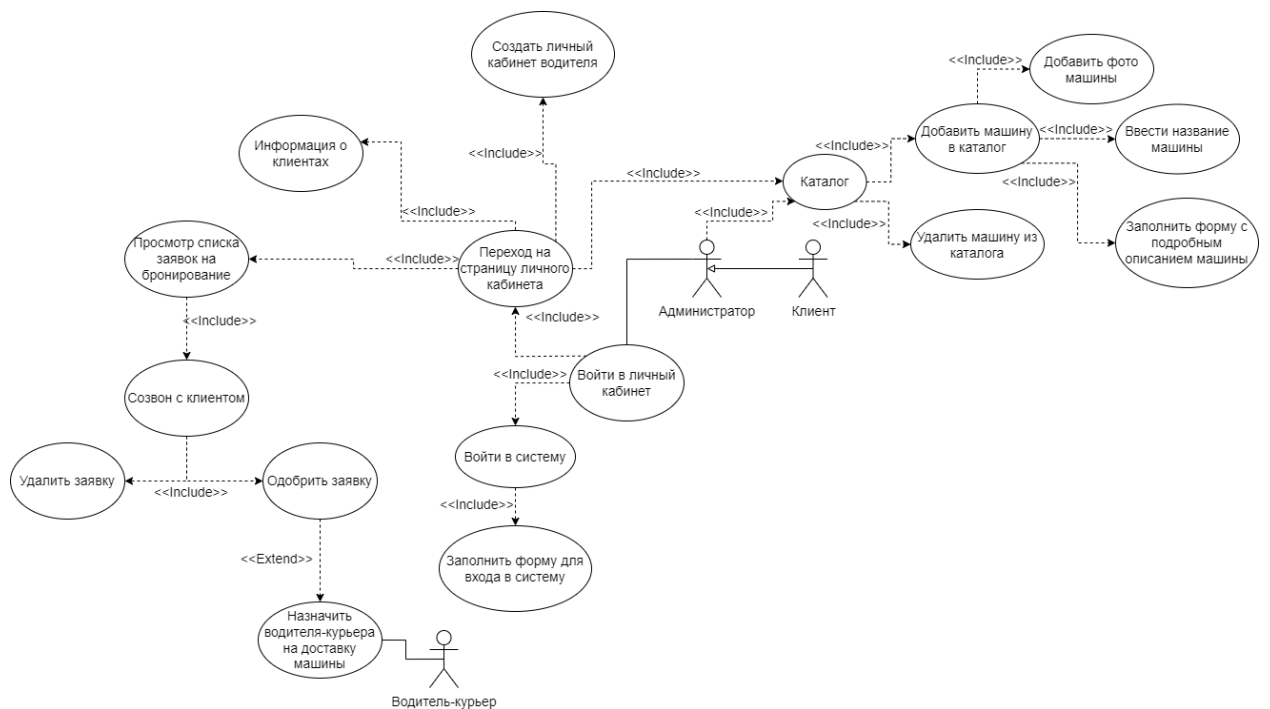


Рисунок 20 - Диаграмма прецедентов со стороны Администратора

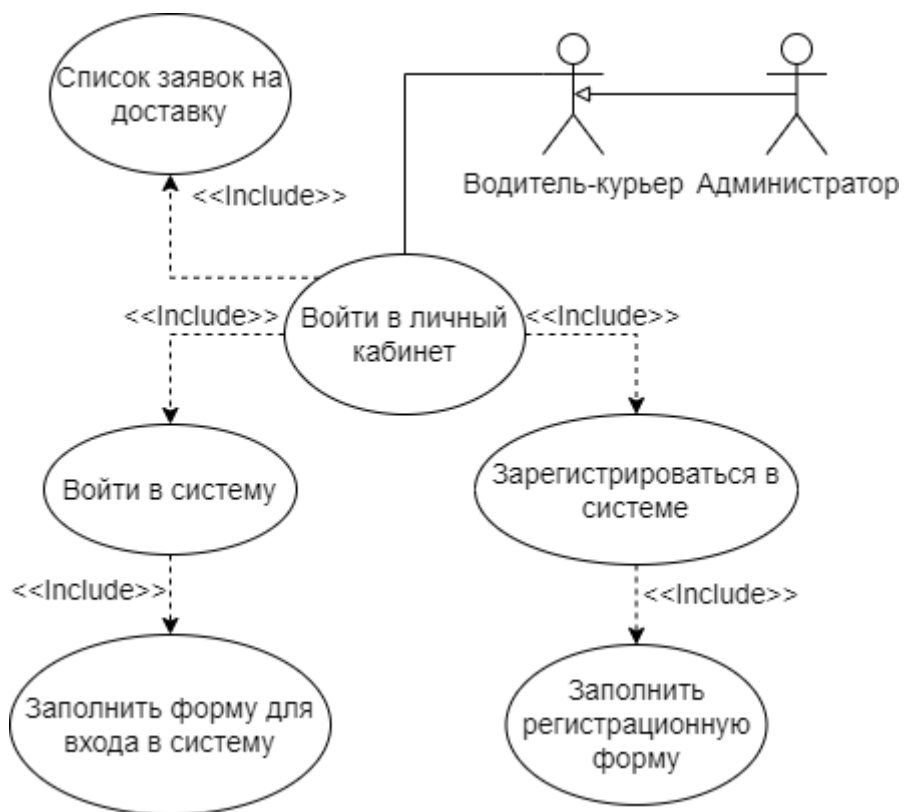


Рисунок 21 - Диаграмма прецедентов со стороны Водителя-курьера

2.3.3 Диаграмма состояний

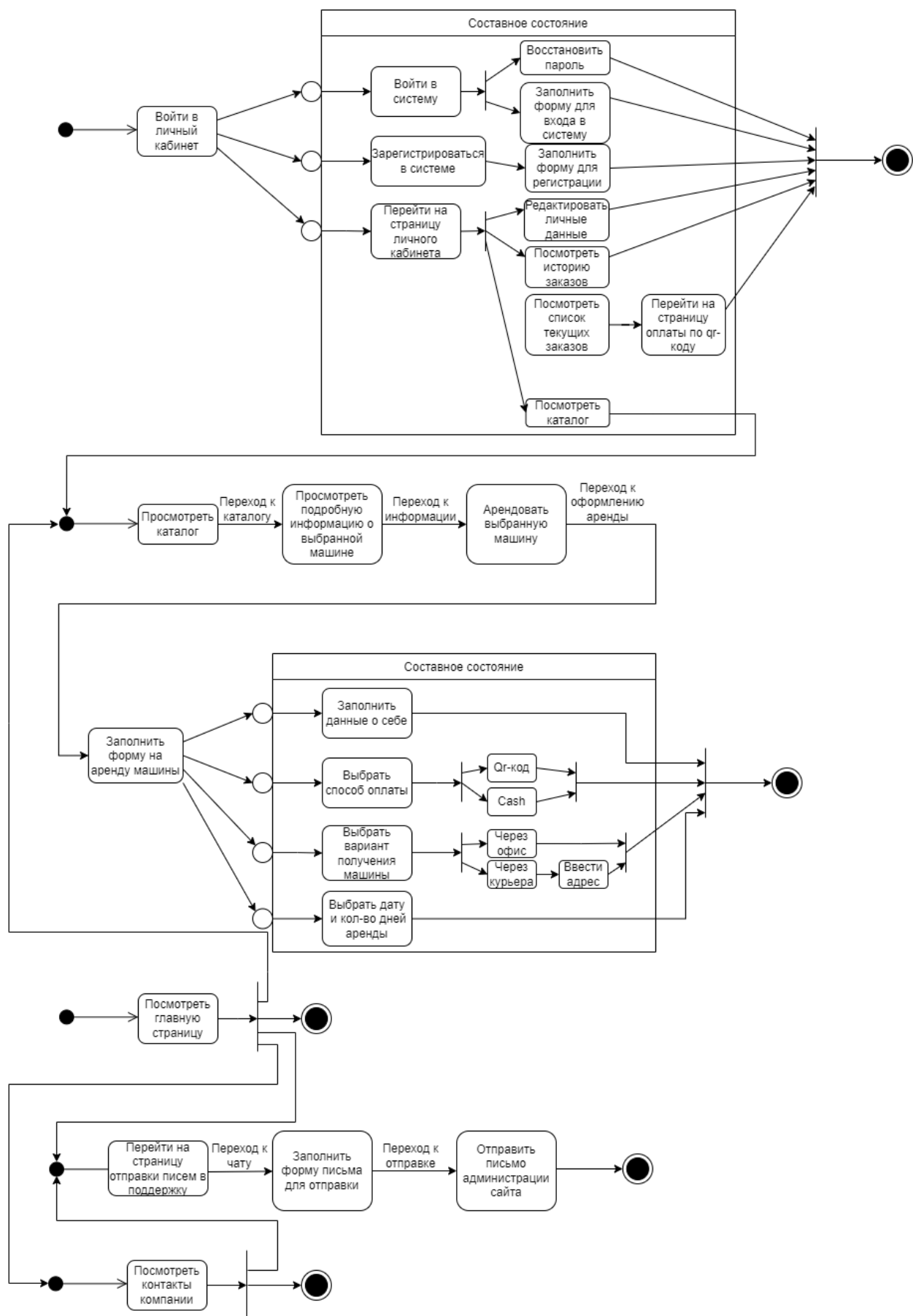


Рисунок 22 - Диаграмма состояний со стороны Клиента

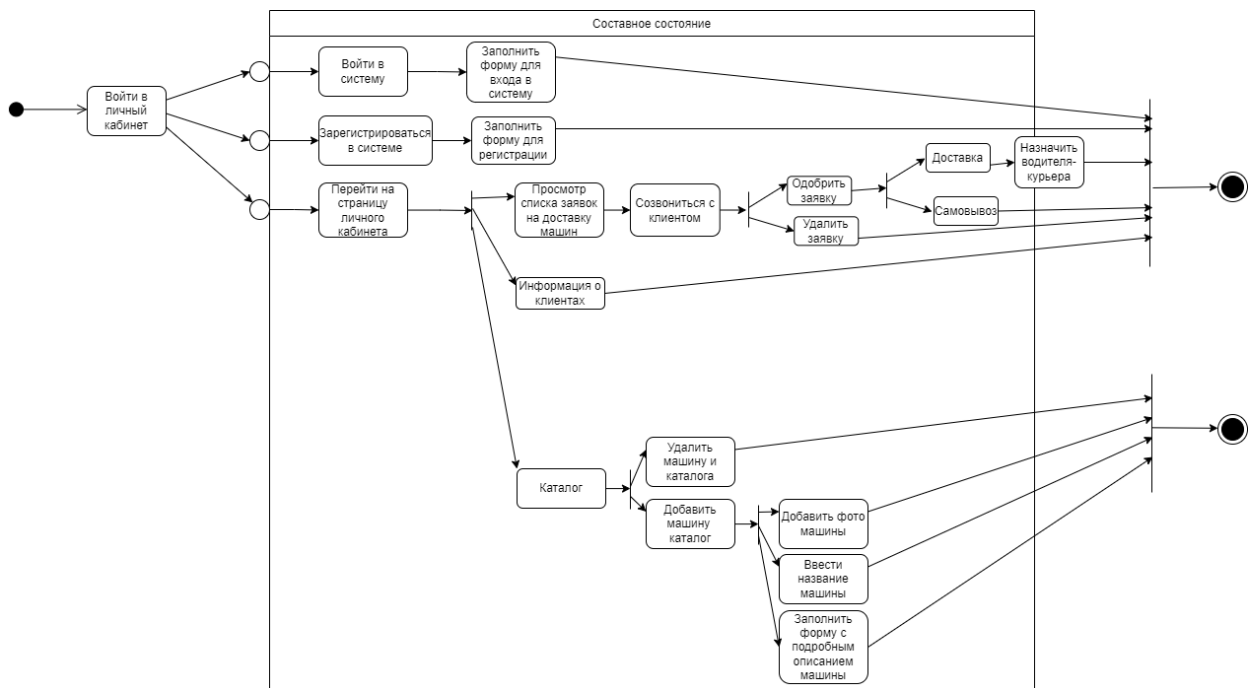


Рисунок 23 - Диаграмма состояний со стороны Администратора

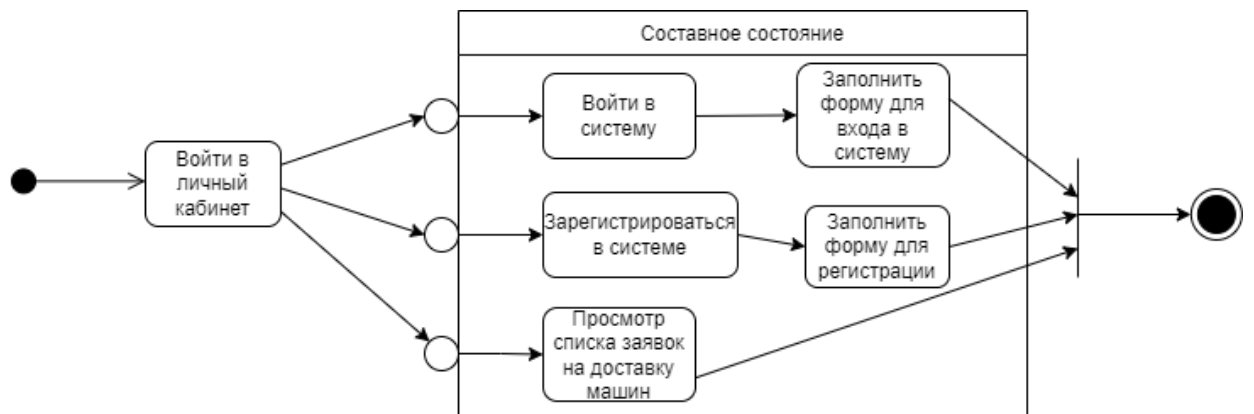


Рисунок 24 - Диаграмма состояний со стороны Водителя-курьера

2.3.4 Диаграмма последовательности

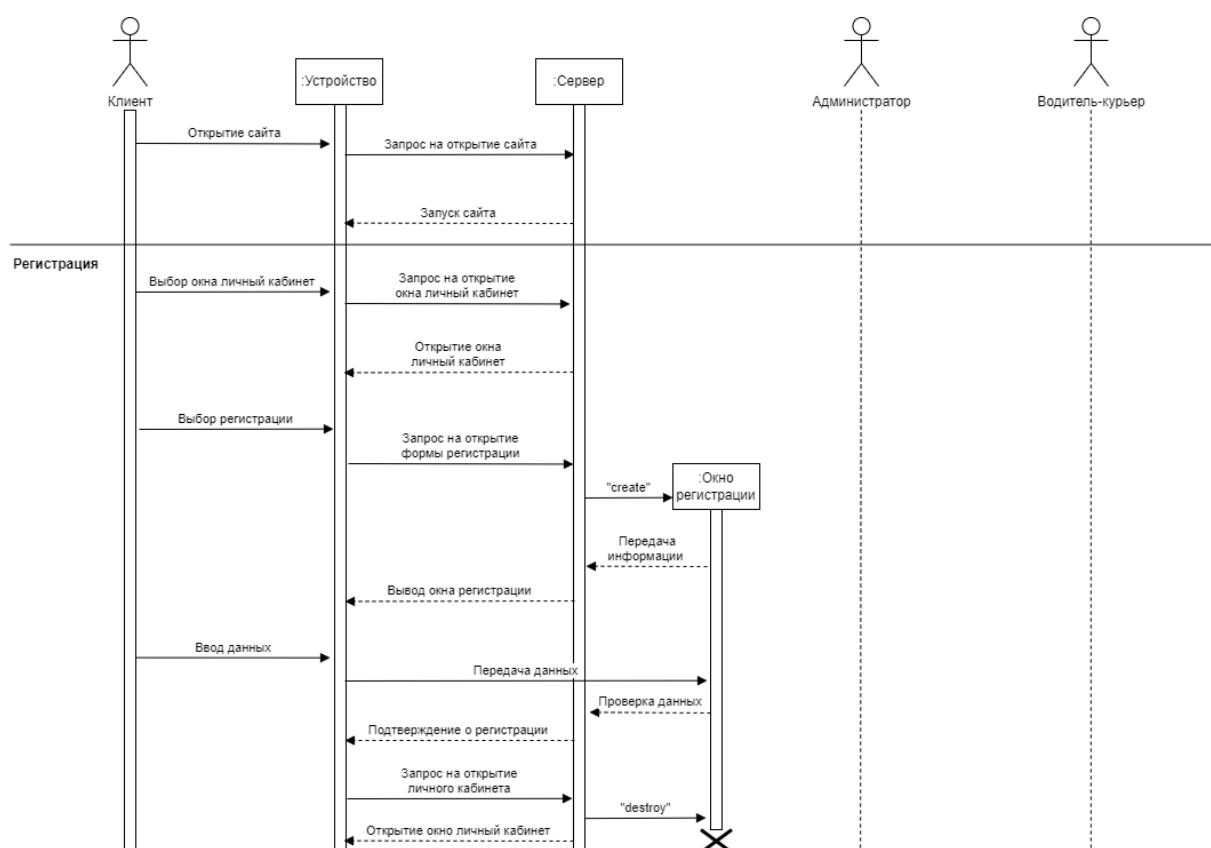


Рисунок 25 - Диаграмма последовательности. Регистрация

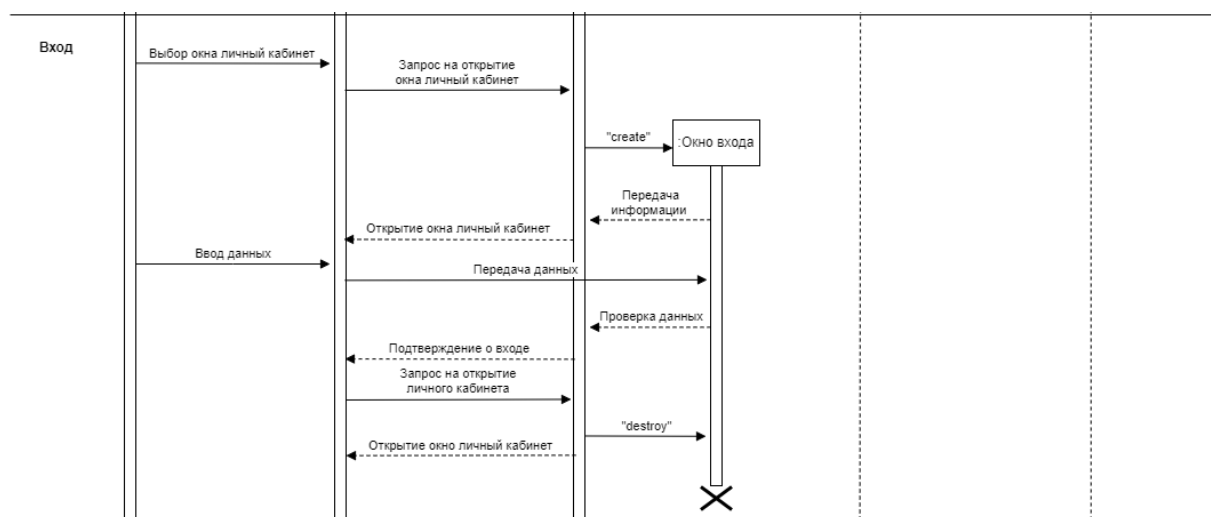


Рисунок 26 - Диаграмма последовательности. Вход

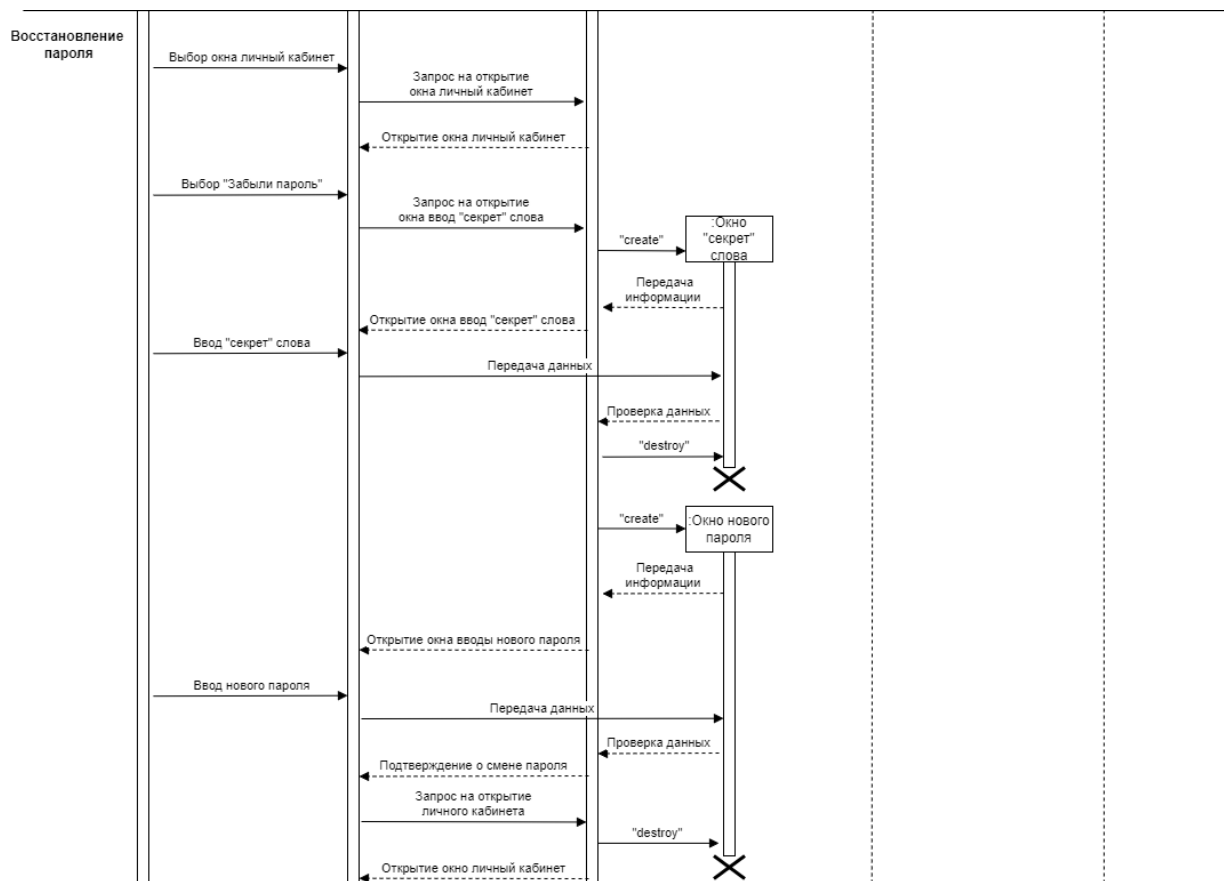


Рисунок 27 - Диаграмма последовательности. Восстановление пароля

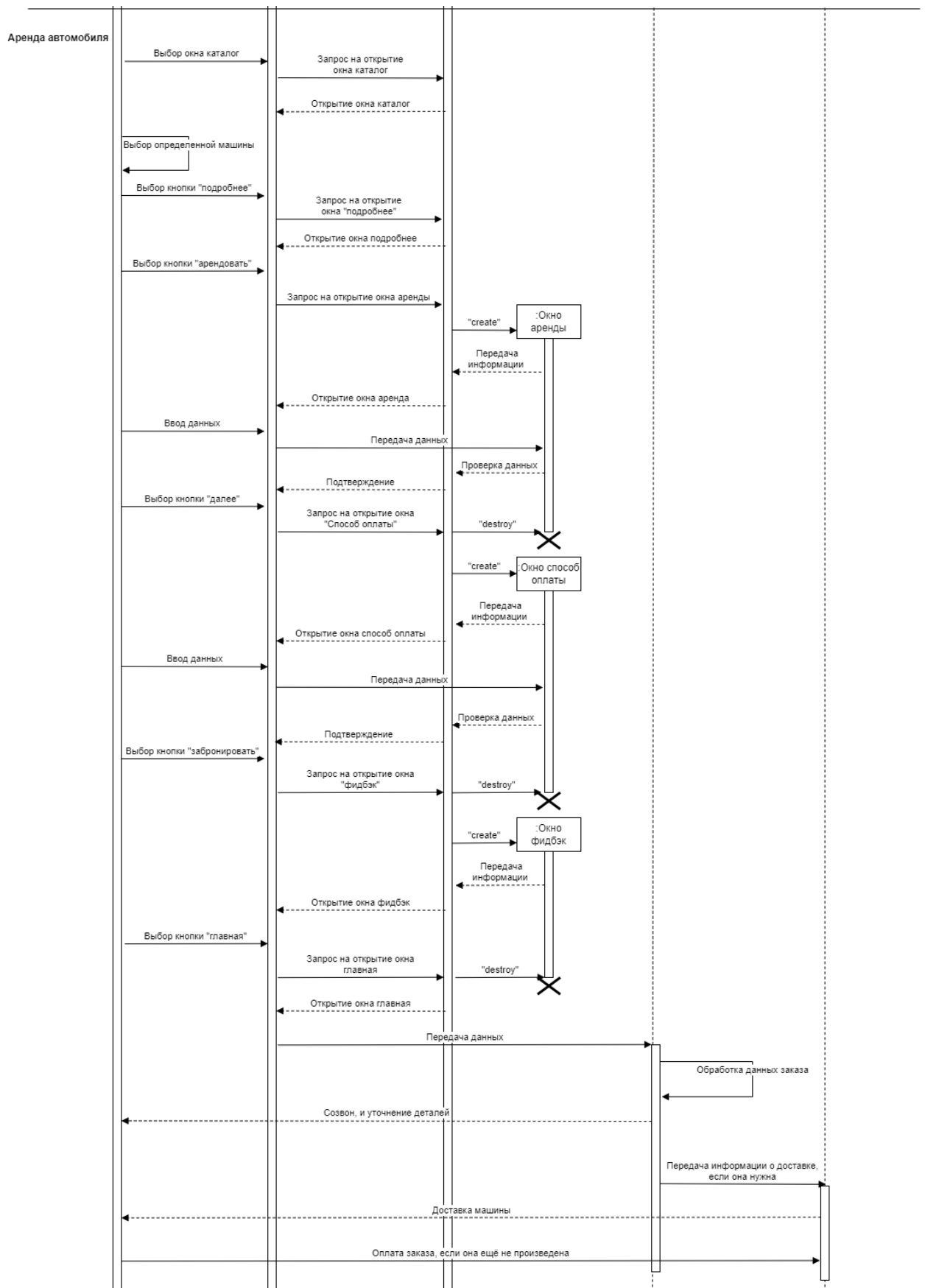


Рисунок 28 - Диаграмма последовательности. Аренда автомобиля

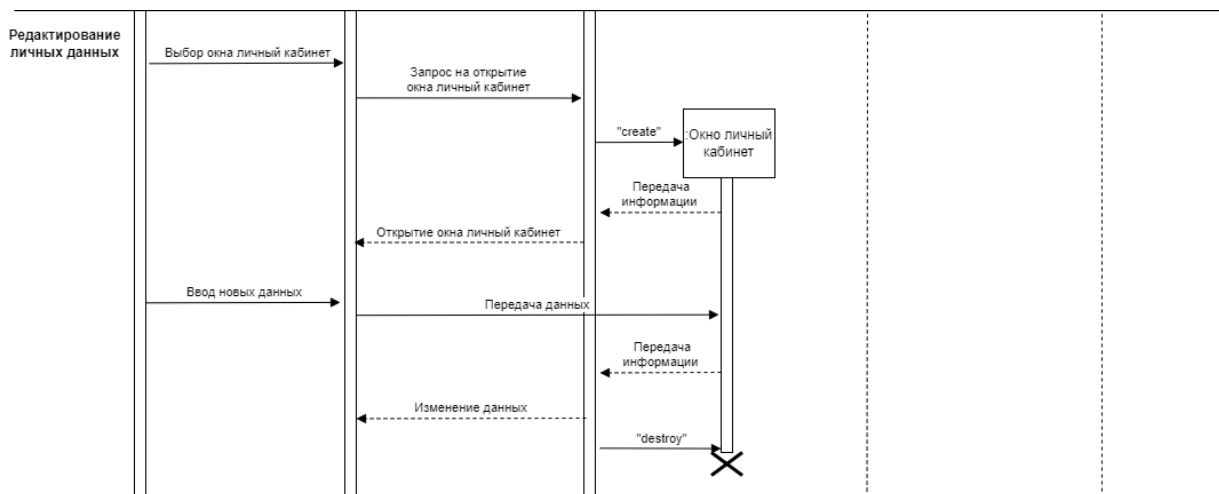


Рисунок 29 - Диаграмма последовательности. Редактирование личных данных

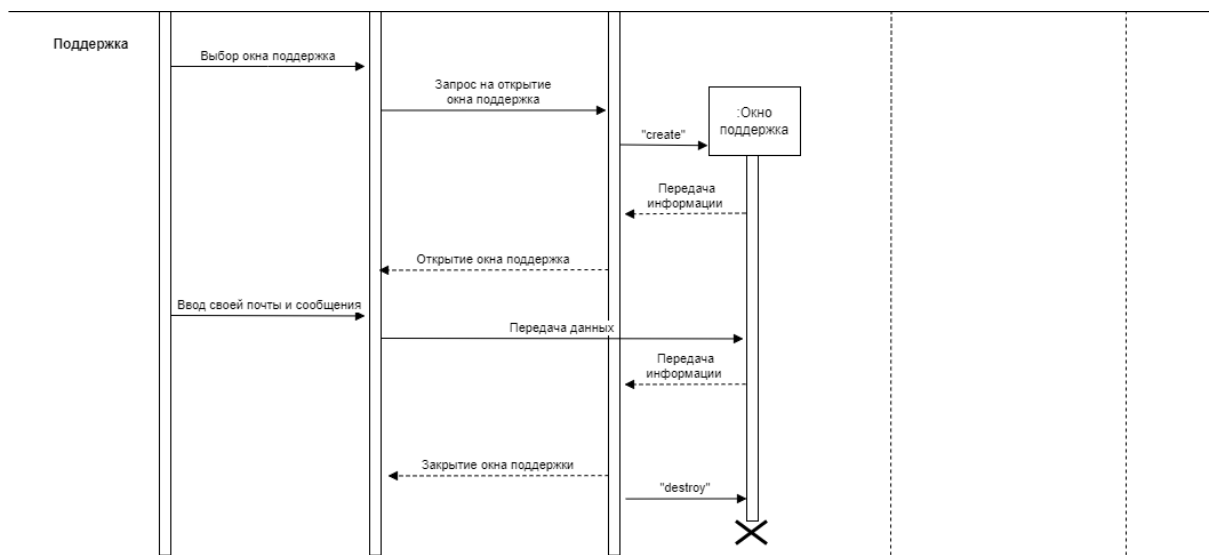


Рисунок 30 - Диаграмма последовательности. Поддержка

2.3.5 Диаграмма активностей

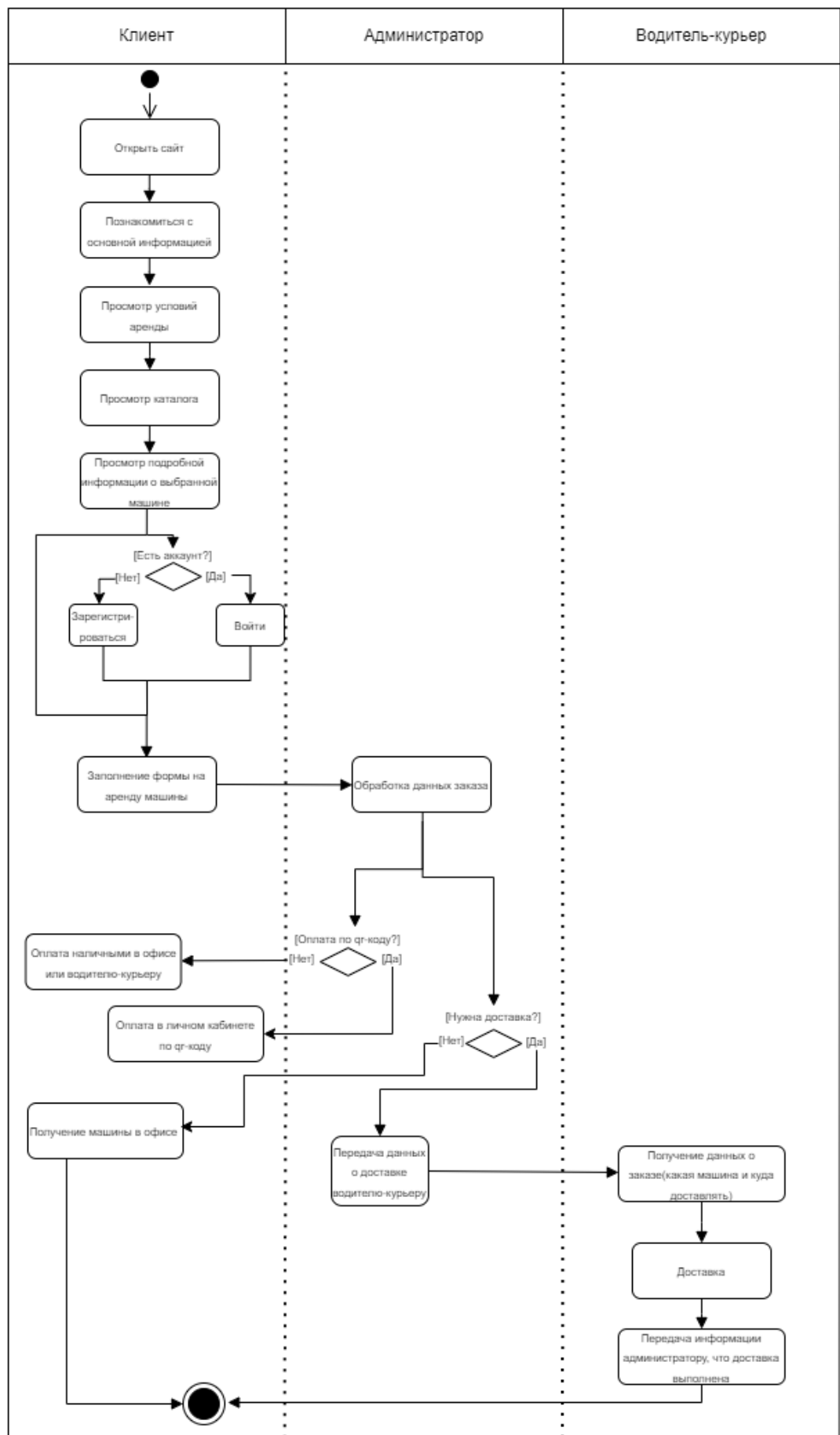


Рисунок 31 - Диаграмма активностей

2.3.6 Диаграмма классов

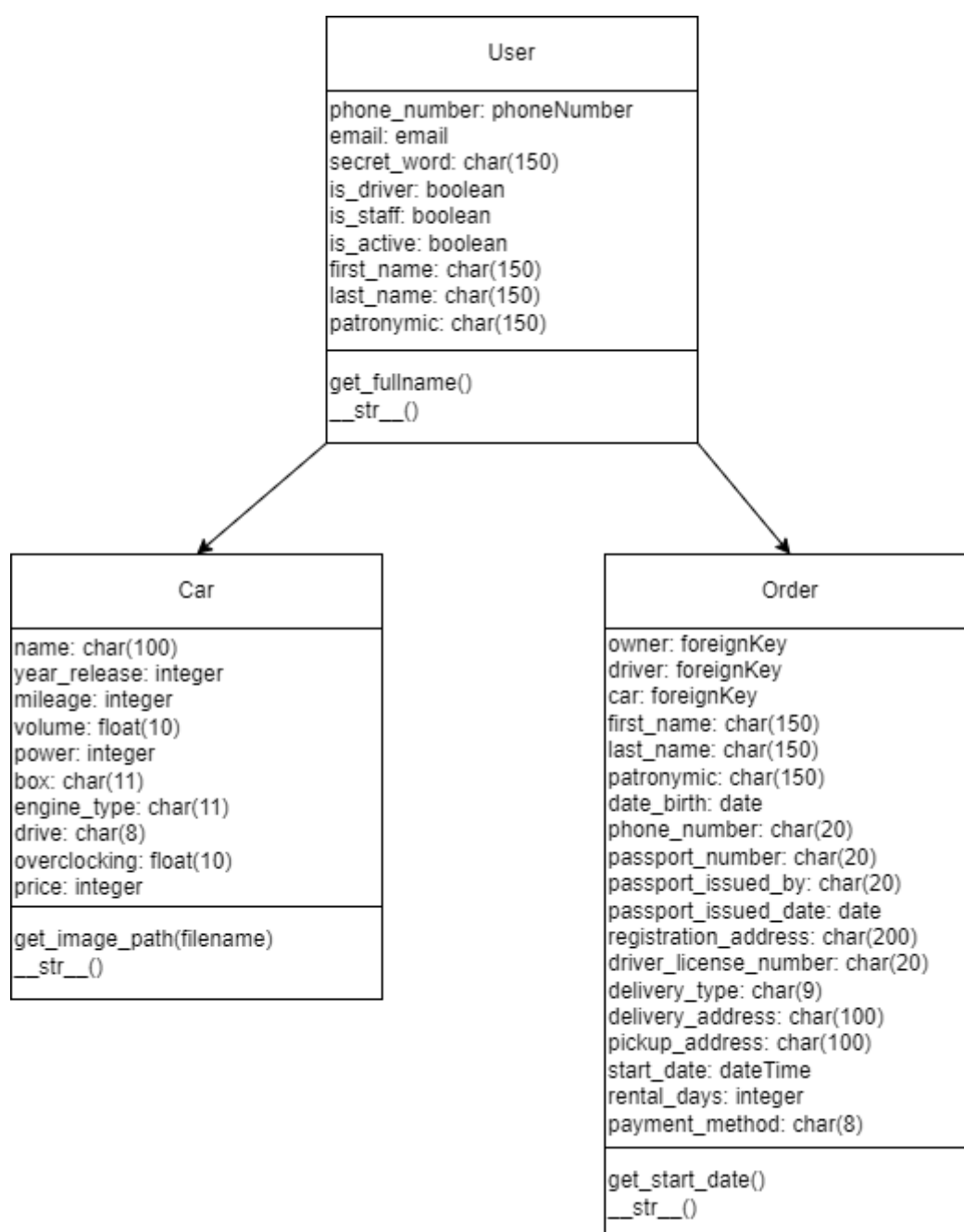


Рисунок 32 - Диаграмма классов

2.3.7 ER-диаграмма

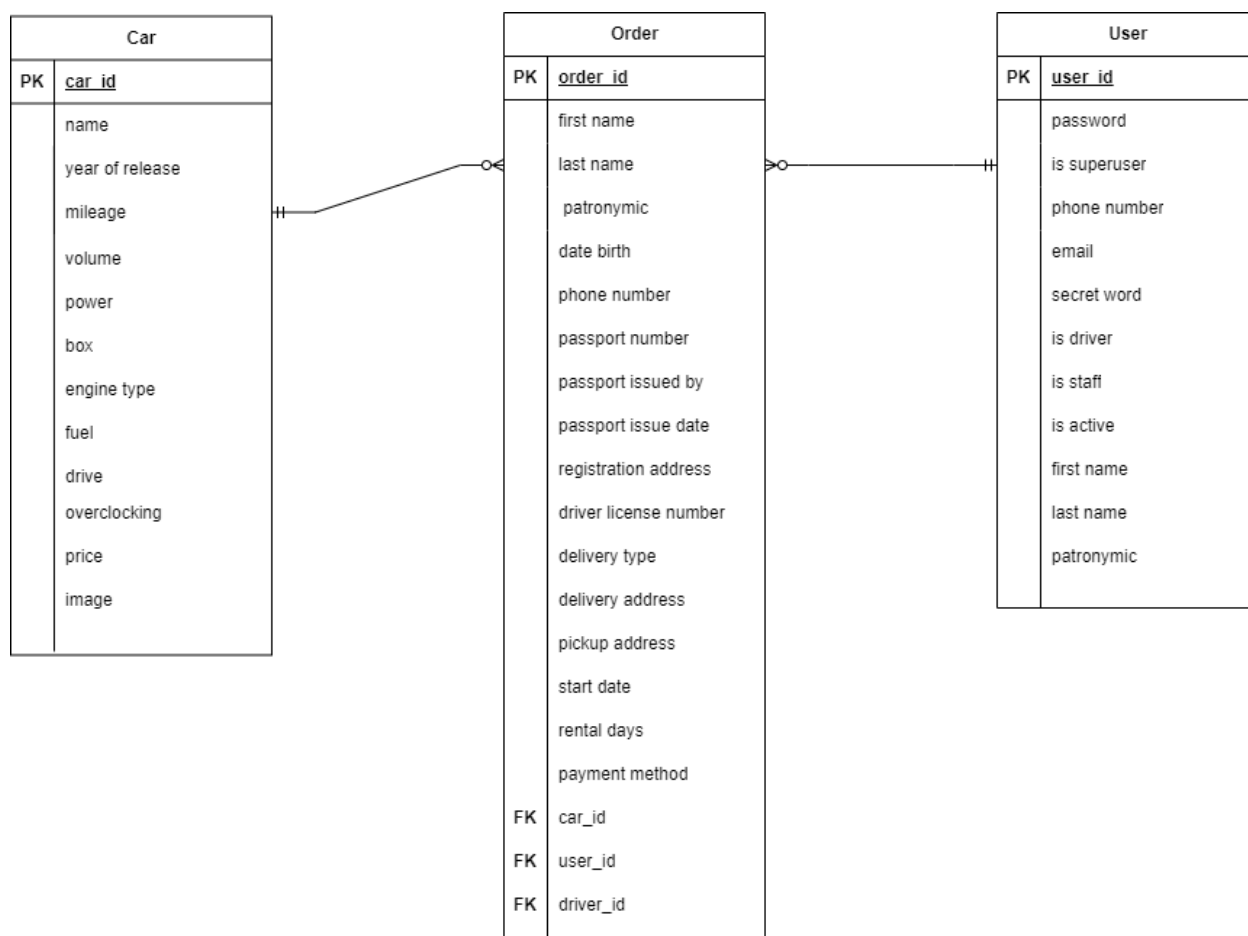


Рисунок 33 - ER-диаграмма

2.3.8 Диаграмма развёртывания

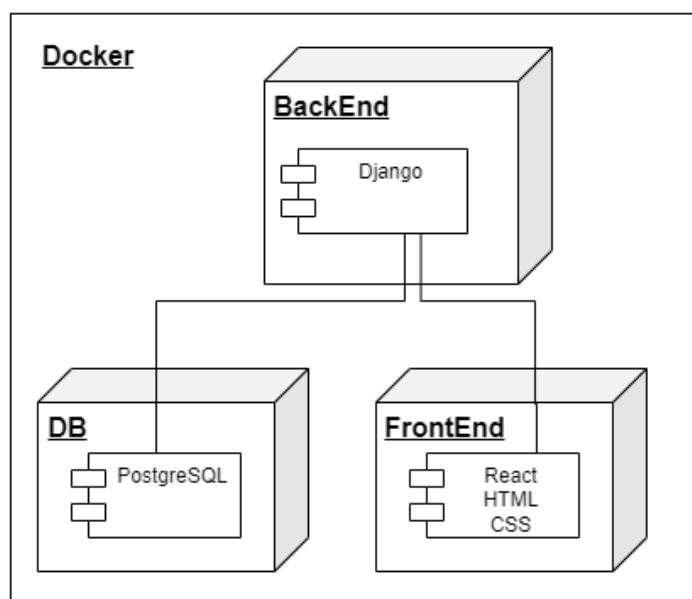


Рисунок 34 - Диаграмма развёртывания

2.3.9 Диаграмма сотрудничества

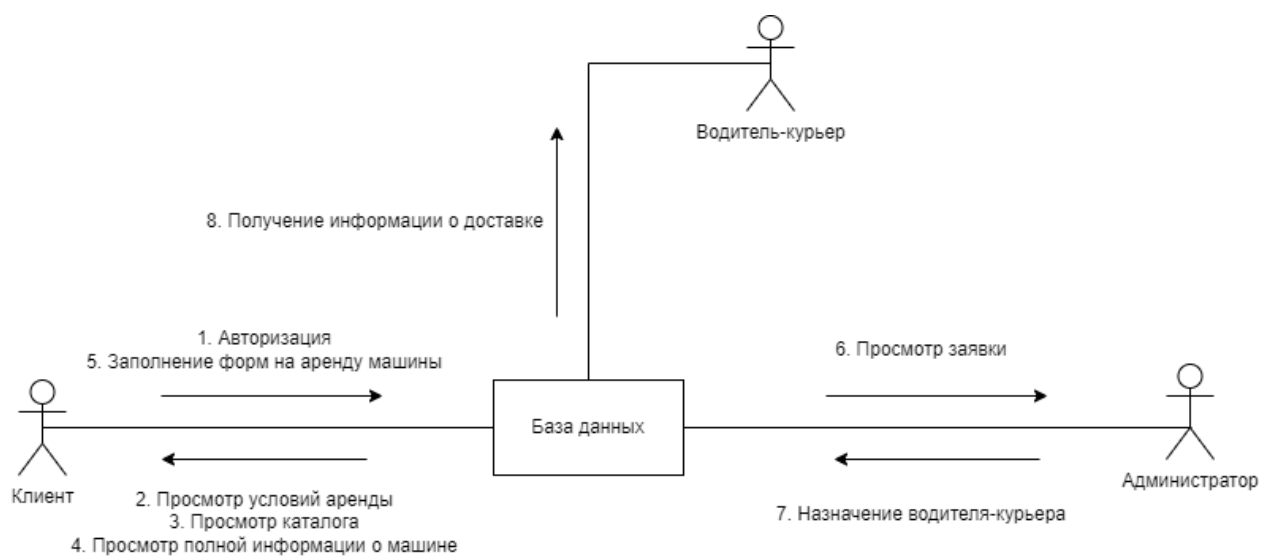


Рисунок 35 - Диаграмма сотрудничества

2.3.10 Диаграмма объектов

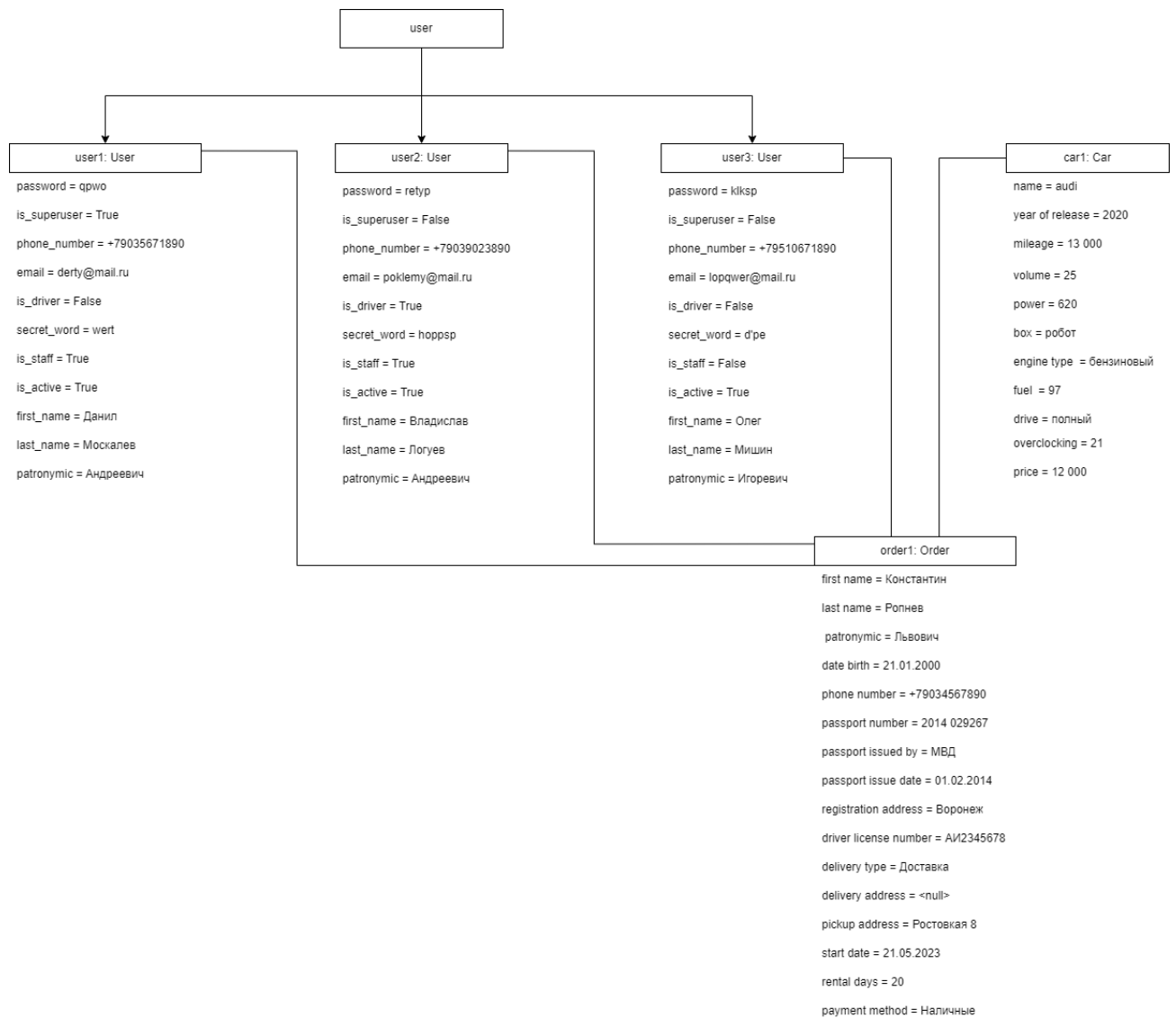


Рисунок 36 - Диаграмма объектов

2.4 Продуктовые воронки

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

Для реализации программно-аппаратной части были выбраны следующие технологии:

- Язык программирования Python;
- Фреймворк Django;
- СУБД PostgreSQL.

Для реализации клиентской части были выбраны следующие технологии:

- Язык программирования JavaScript;
- Язык стилей CSS;
- Фреймворк React;
- Язык разметки HTML.

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 `front-end`-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

4.2 Юзабилити тестирование

4.3 Тестирование сервера и его взаимосвязи с клиентом

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