**<<ТАШКЕНТСКИЙ УНИВЕРСИТЕТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ ИМЕНИ Мухаммада Аль-Хорезмий>>**

**Ургенчский филиал**

Факультет: Телекоммуникационные технологии

Кафедра: Программный инжиниринг

**Курсовая работа**

**на тему:**

**<<Car service (Программа для авто сервисов) проектирование>>**

**Автор:** студенты группы\_\_\_\_943-20\_\_\_\_\_\_\_

очной формы обучения \_\_\_\_\_\_\_\_\_\_\_

Ф.И.О. Сабиров Жамшид Пулатович  
Кучкаров Азамат Игоревич   
Подпись\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Руководитель:**

Ф.И.О. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
Подпись\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Ургенч – 2022

Содержание

* Введение \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_3
* Цель курсовой работы \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_3
* Соглашение о терминах \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_3
* Общие сведения продукта \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_4
* Цели и задачи данного проекта \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_6
* UML диаграммы проекта \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_6
* UI/UX дизайн \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_8
* Краткое ТЗ (тех. задание) \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_12
* Тесты \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_16
* Заключение \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_19
* **Введение**

Наше поколение живет в таком времени, в котором почти все люди владеют той или иной информацией. Поэтому роль информационных технологий огромна в жизни каждого из нас. И, порой невозможно представить нашу жизнь без этих технологий.

На сегодняшний день, с помощью всех современных устройств наша жизнь стала намного проще, а главное – удобнее! Ведь всего пару десятков лет назад человечество и мечтать не могло о том, что может позволить себе сейчас. Я считаю, что с появлением ИТ жизнь каждого поменялась в положительную сторону. Хотя бы потому что меньше вреда наносится окружающей среде, так как вся информация теперь хранится на всевозможных электронных носителях.  
И хотел бы внести свою долю в развитие определенной отрасли в жизни человека.

* **Цель курсовой работы**

Цель курсовой работы проектировать программный продукт для владельцев авто сервисов, позволяющий им контролировать и управлять своим бизнесом в не зависимости от место нахождения (владельца).

## Соглашение о терминах

## Пользователи в системе имеют три роли:

## - Product Owner (P.Owner)– Владелец всей системы, а не авто сервиса

## Имеет все привилегии.

## - Business Owner (B.Owner) – Владелец какого-либо сервиса

## Имеет все привилегии, кроме:

## - отправки писем владельцам автосервисов

## - просмотра других автосервисов кроме своего

## - Manager – Работник этого сервиса, который может добавлять заказы, добавлять жалобы и редактировать предоставляемые услуги сервиса.

## Домен – символьное имя идентифицирующее адрес сайта(сервиса).

## Endpoint - (в переводе с англ. — конечная точка) представляет собой некий шлюз, который соединяет серверные процессы приложения с внешним интерфейсом. Простыми словами, это адрес, на который отправляются сообщения.

## ТЗ – техническое задание для разработчика(ов).

## JWT - тот открытый стандарт (RFC 7519) для создания токенов доступа, основанный на формате JSON.

* **Общие сведения продукта**

## Наименование системы: Car service (Авто сервис)

## Видение продукта:

## Данный продукт будет помогать B.Owner контролировать свой бизнес вне зависимости места нахождения B.Owner. Знать точную информацию о персонале сервиса, количестве клиентов, качестве облуживания, суточного (недельного и месячного) оборота сервиса. Тем самым даст возможность увеличивать количество филиалов без страха и с уверенностью зная, что его бизнес работает только на него.

## Функциональность продукта

## Данный продукт позволит B.Owner добавлять ограниченное кол-во сервисов, персонал для каждого сервиса, Manager сервиса для контролирования обслуживающего персонала и поддержки обратной связи с посетителями и клиентами сервиса.

## Целевая аудитория

## Целевая аудитория данного проекта бизнесмены, имеющие свои сервисы по предоставлению услуг автолюбителям (мойки, мастерские, детейлинги, тюнинг ателье, и т.д.) которые хотят контролировать и развивать свой бизнес по всей стране и не только.

## Масштаб проекта

## Проект подразумевается в первую очередь региональным (г. Ургенч).

## Функциональность системы;

## Вход/регистрация через JWT токен.

## Для просмотра, изменения, добавления данных своего автосервиса нужна обязательная авторизация

## Запросы на Endpoint’ы должны передаваться строго нужного типа и должны валидироваться в ином случае

## Будущий UI/UX интерфейс должен быть легко понятным и главное адаптируемым

## Не функциональные требования

## На этом этапе не нужно реализовывать оплату за сервис

## Просмотр главной страницы без авторизации

## Нет необходимости кеширования данных, данные должны быть актуальными на момент их просмотра.

## Цели и задачи данного проекта

## Повышение удобства управления своим бизнесом

## Расширения бизнеса

## Достижение конкурентных преимуществ перед подобными компаниями

## Обеспечение безопасности финансовой составляющей системы

## Повышение контроля и качества над выполненными работами

## Изучение и максимальное удовлетворение потребностей посетителей сервисов.

## Управления финансами

## Анализ и планирование дохода

* **UML диаграммы проекта**

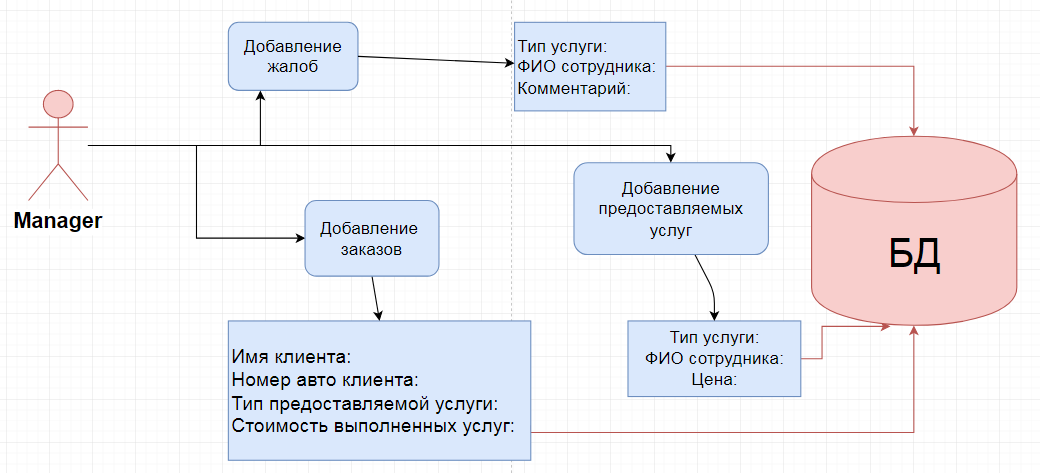


Рисунок 1: UML диаграмма 1.0

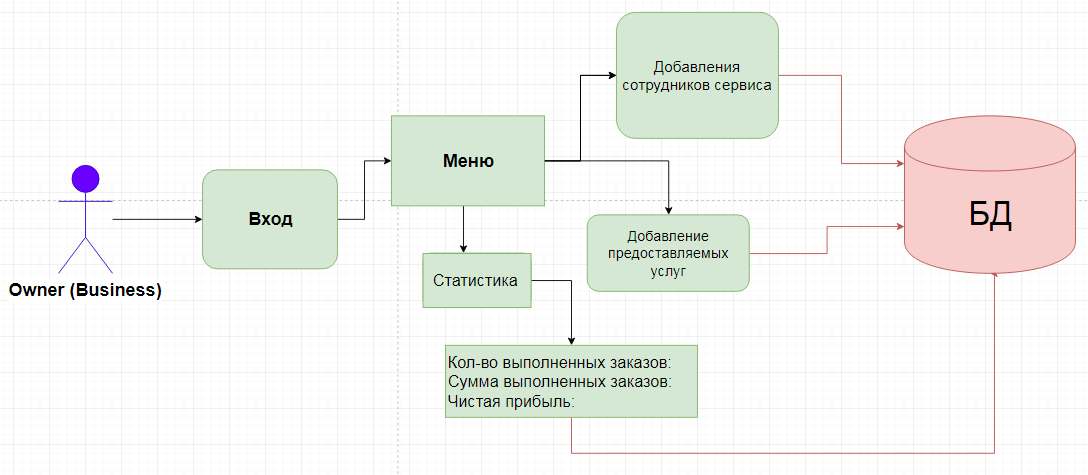


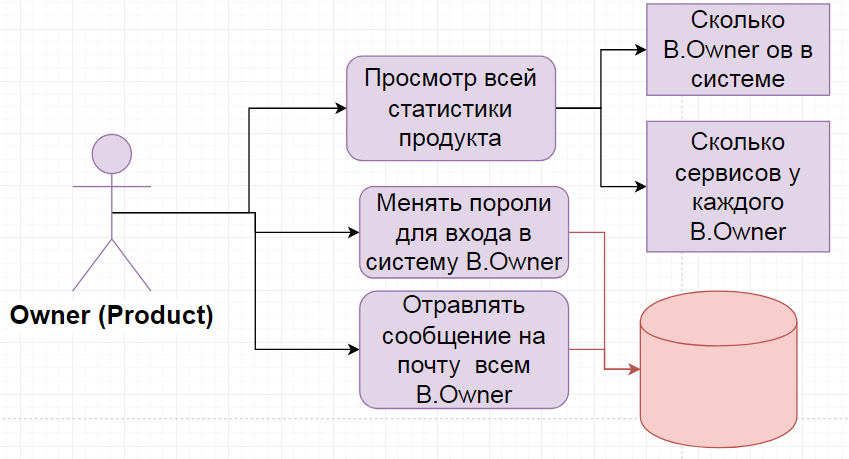
Рисунок 2: UML диаграмма 1.1  
  


Рисунок 3: UML диаграмма 1.3



Рисунок 4: Use case диаграмма

## UI/UX дизайн

## 

Рисунок 5: Главная страница / окно входа

Главная страница продукта выглядит очень простым для удобства пользования. Для входа в свой аккаунт требуется ввести логин и пароль пользователя, (**рисунок 3)**.

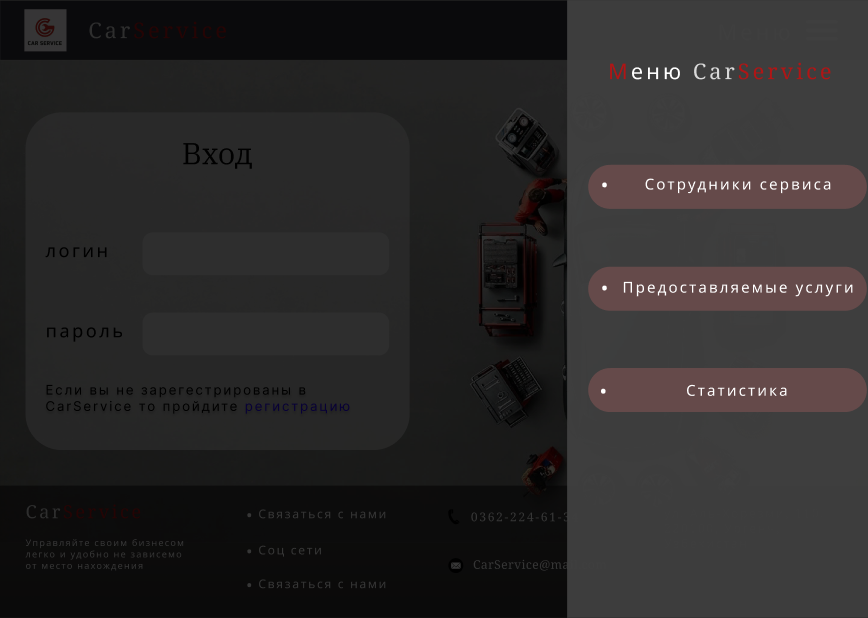


Рисунок 7: Окно меню

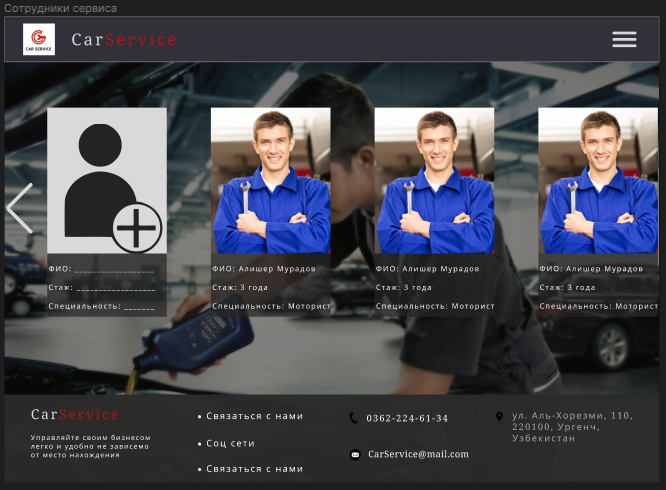


Рисунок 8: Окно добавления сотрудников

В разделе “сотрудники сервиса” можно добавлять и удалять сотрудников. Каждый сотрудник имеет пункты: **ФИО**, **сфера работы** и **опыт работы**. (**Рисунок 6**)

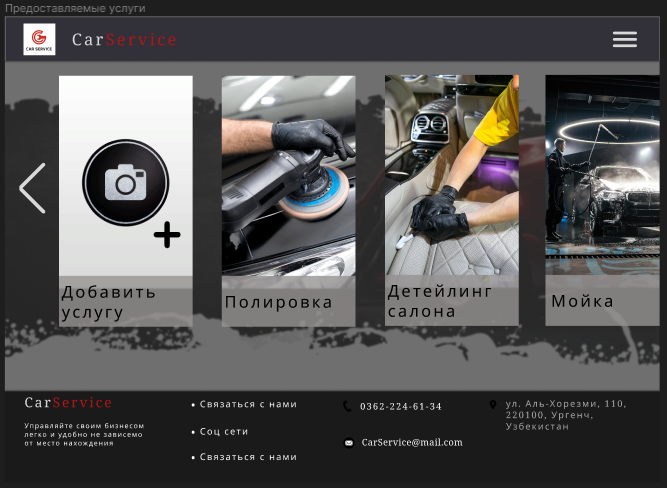


Рисунок 9: Окно редактирования предоставляемых услуг

В разделе “**Предоставляемые услуги**” можно добавлять и редактировать виды услуг существующих в сервисе. (**рисунок 7**)

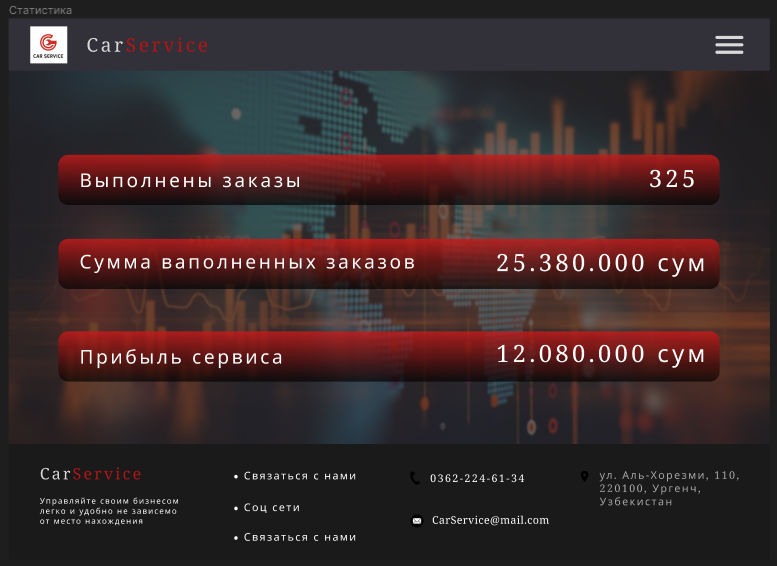


Рисунок 10: Окно статистика сервиса

В разделе “Статистика” можно наблюдать статистику сервиса, такие как: **Выполненные заказы**, **Сумма выполненных заказов** и **Прибыль сервиса. (рисунок 8)**

Ссылка на Figma UI/IX <https://www.figma.com/file/o1ZtgDOiqt5Q8SWeNz3mhc/Untitled?node-id=103%3A2&t=19DYq4JIY8oaAH1G-1>

## Краткое ТЗ (тех. задание)

## Таблицы в бд(желательно в PostgreSql) которые должны быть:

## Таблица WashCompany c полями: id, name, avatar(string), location(string)

## Таблица User c полями: id, name, login, password (логин уникальный), role Всего должно быть три роли: роль админа(admin), роль владельца(owner), роль менеджера(manager)

## Таблица Washer c полями: id, name, telephoneNumber(Int), stake(Int), image(string), isActive(Boolean)

## Таблица Service c полями: id, name, duration(Int), price(Int)

## Таблица Order с полями: id,set<<service>>, set<<washer>>, price(Int), carModel(string), carNumber(string), clientName(string), clientNumber(Int), isActive(Boolean), isCancelled (Boolean), date (Date) У одного ордера могут быть несколько сервисов и несколько мойщиков

## Таблицы соответствий: – Таблица (washCompanyId, userId) – Таблица (washCompanyId, orderId) – Таблица (washCompanyId, wahserId) – Таблица (washCompanyId, serviceId) – Таблица (washCompanyId, journalId) у одной washCompany могут быть несколько ордеров у одной washCompany могут быть несколько юзеров у одной washCompany могут быть несколько мойщиков у одной washCompany могут быть несколько услуг у одной washCompany могут быть несколько журналов

## Все Id должны быть primary key

## Эндпоинты которые должны быть: -api/signIn (Авторизация) для юзера -api/signUp (Регистрация) для юзера -api/signOut (Выход) для юзера

## //washCompany -api/washCompanyInsert - метод post для добавления washCompany -api/getWashCompanyId – метод post , который возвращает id WashCompany по паролю и логину юзера

## // orders -api/{washCompanyId}/orders?isActive….= …. &dateFrom=….&dateTo=…. &page = … – вернуть все активные(либо выполненные) ордеры по определенной мойке в указанные даты Пагинация по 10 элементов -api/order?id = … - вернуть все детали о конкретном заказе(order) - api/{washCompanyId}/insertOrder - метод post для добавления заказа(order) в таблицу -api/{washCompanyId}/updateOrder - метод post для изменения данных у заказа(order) по полям: carModel, carNumber, set<<admin>>, set<<service>>,price,isCancelled,isActive

## //washer -api/{washCompanyId}/washers?searchName=…. &page=... - вернуть всех мойщиков(washer) по определенной мойке , у которых имя совпадает с searchName Пагинация по 10 элементов -api/washer/?id=.... - вернуть все детали о мойщике -api/{washerId}/getOrders?dateFrom = …. &dateTo = …. & isActive=….&page = …. - Вернуть все активные(либо выполненные) заказы у определенного мойщика по выбранной дате Пагинация по 10 элементов -api/{washCompanyId}/insertWasher – метод post для добавления мойщика(washer) -api/{washCompanyId}/updateWasher – метод post для изменения данных у мойщика по всем полям

## //services -api/{washCompanyId}/services?searchName=…. &page=... - вернуть все услуги(service) по определенной мойке Пагинация по 10 элементов -api/service/?id=.... - вернуть все детали об услуге -api/{washCompanyId}/insertService – метод post для добавления услуги -api/{washCompanyId}/updateService – метод post для изменения данных услуги по всем полям

## //analytics -api/{washCompanyId}/analytics?dateFrom = …. &dateTo = …. - вернуть { “totalOrders”:Int (количество заказов у мойки за выбранный период времени), “toalWashers”:Int(общее количество мойщиков (без дубликатов) у заказов за выбранный период времени), “ordersSum”:Int(Общая прибыль за все заказы за выбранный период времени), “washersSum”:Int(Общая прибыль мойщиков за все заказы за выбранный период времени, считается как [сумма долей всех мойщиков \* 0.01 \* цена заказа ) }

## **(Только для юзера с ролью P.Owner)** //credentials -api/changePassword – метод пост для изменения пароля у юзера с ролью P.Owner. Передаваемые данные: { “oldPassword”:String, “newPassword”:String }

## //support -api/sendMessage – метод пост для отправки сообщения на конкретный gmail (поле toEmail) Передаваемые данные: { “contact”:String, “message”:String, “toEmail”:String }

## Тесты

## 

Рисунок 11: полный список возможный url для запросов

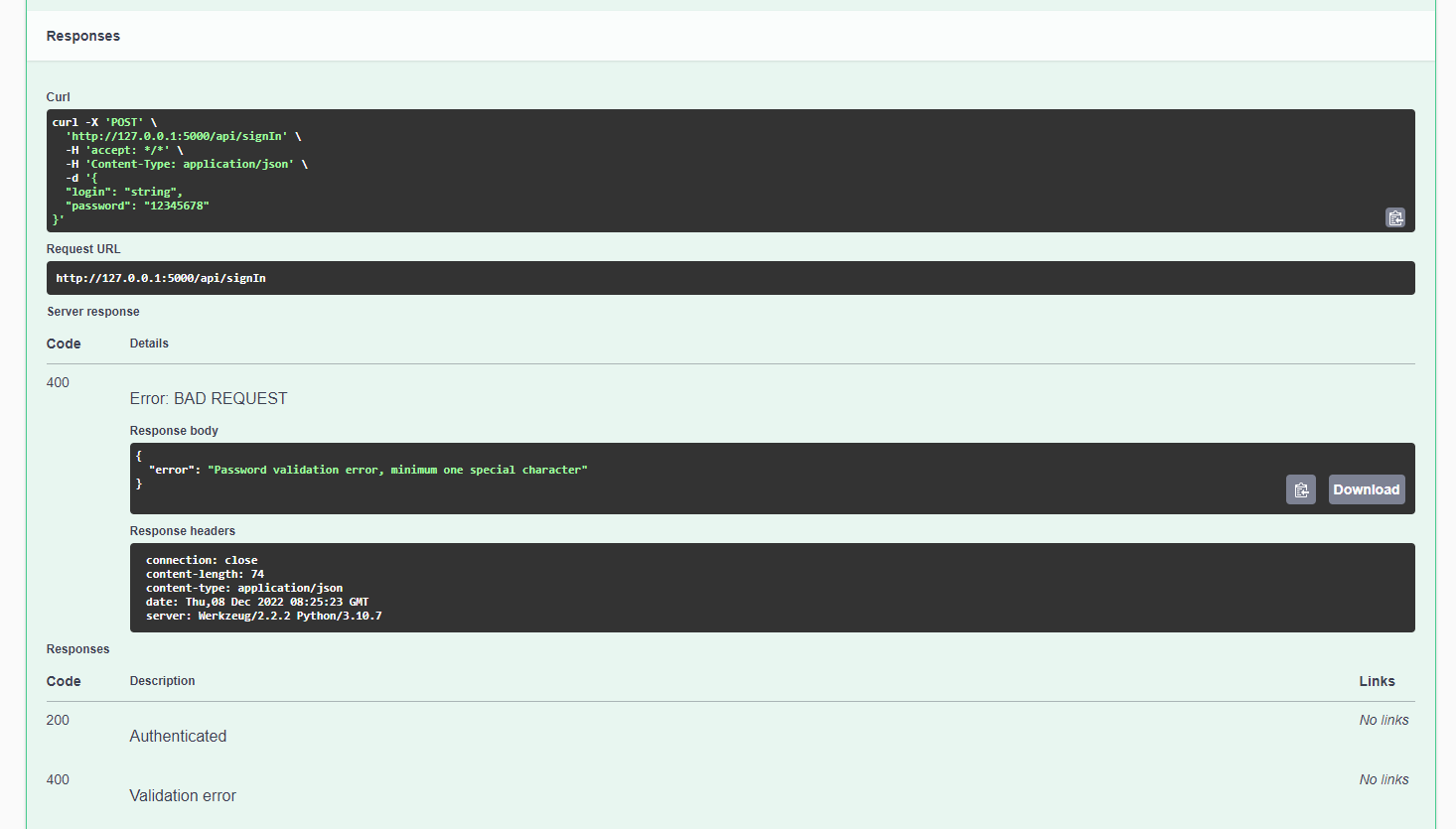


Рисунок 12: Ошибка при аутентификации с неверным паролем

## 

Рисунок 13: Регистрация пароль в виде hash(password + salt)

## 

Рисунок 14: Получение заказов для определенной мойки с id = 1

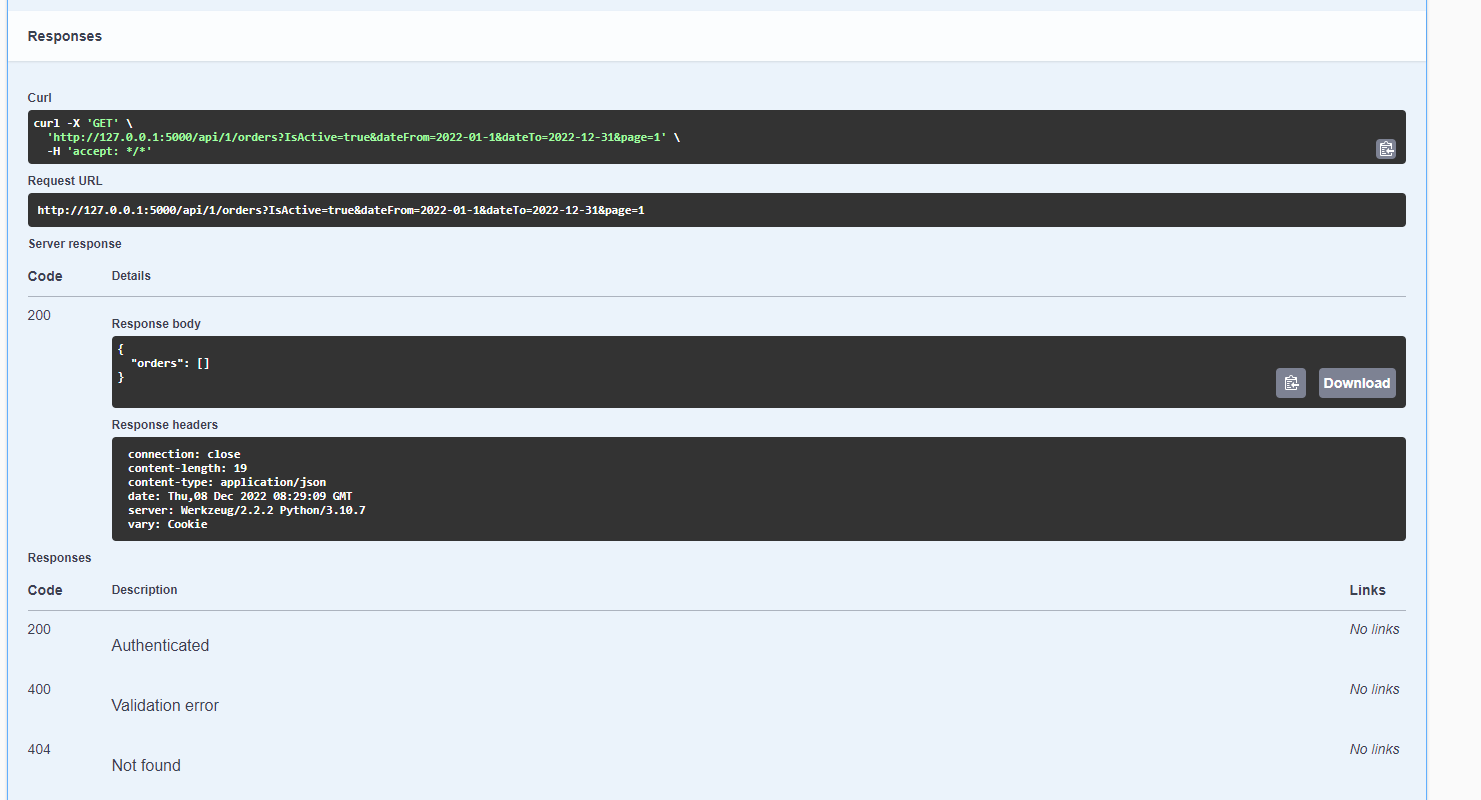


Рисунок 15: Ответе, пустой список - значит нет заказов

## Заключение

## В ходе выполнения данной курсовой работы был создан проект (Car service) программного продукта для отслеживания авто сервиса онлайн и получен опыт проектирования программных обеспечений. Также был спроектирован UI/UX дизайн на платформе Figma. Были применены знания по языку программирования Python и его фреймворков.