Министерство образования Республики Беларусь

Учреждение образования

БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Факультет компьютерных систем и сетей
Кафедра информатики
Дисциплины:
Технологии разработки программного обеспечения
и Технологии обработки финансовой информации.

Техническое задание

“Сервис по оказанию логистических услуг «Пакетик»”

Выполнила:
студентка
гр. 953501
Гресик М. М.

Минск 2022

СОДЕРЖАНИЕ

1. Введение

1.1. Назначение проекта

Платформа предоставляет возможность заказа, выбора и оплаты доставок посредством веб-приложения. Список текущих доставок и заказов хранится в личном кабинете пользователя.

Основная задача сервиса предоставить пользователям удобный, интуитивно-понятный интерфейс и возможность легко и быстро оформлять доставку.

1.2. Предполагаемая аудитория

Целевой аудиторией будут являться компании и лица заинтересованные в быстрой и удобной доставке, а так же лица заинтересованные в подработке курьером.

2. Общее описание

2.1. Функциональность продукта

Функции:

- Регистрация пользователя
- Авторизация пользователя
- Создание заказов на доставку
- Удаление заказов
- Просмотр текущих заказов и доставок
- Просмотр истории заказов и доставок
- Просмотр баланса и истории транзакций
- Настройка аккаунта пользователя

2.2. Классы и характеристики пользователей

Веб-приложение будет иметь два вида пользователей - заказчик, который будет иметь возможность регистрироваться, входить в систему, создавать заказы на доставку, просматривать историю заказов, редактирование своего аккаунта; курьер будет иметь возможность регистрироваться, входить в систему, просматривать текущие заказы на доставку, просматривать историю доставок, редактирование своего аккаунта.

3. Функциональность системы

3.1. Функциональный блок Back-End

3.1.1. Описание и функциональные требования

В качестве средств разработки бэкенда используется веб-фреймворк Django REST Framework, СУБД PostgreSQL и утилита развертывания окружений Docker с установленным docker-compose для поддержки нескольких контейнеров в кластере. Тип архитектуры бэкенд-приложения — микросервисная.

Реализация микросервисной архитектуры бэкенд-приложения (API) будет основана на запуске нескольких изолированных контейнеров под каждый из микросервисов (API, database, ...). Сила микросервисной архитектуры состоит в том, что любой из компонентов, как правило, легко заменяется/удаляется/добавляется в кластер сервисов, то есть если до конца не ясна полная архитектура сервиса или же будет расширятся со временем — это отличный выбор. Недостатками являются трудоёмкость анализирования и интеграционного тестирования таких систем. Также важно понимать, что если один из сервисов в кластере перестанет работать, то это не должно повлиять на работоспособность остальных частей. В качестве бэкенд-приложения выступает один из сервисов кластера, API. Если он перестанет работать, значит и сама платформа тоже.

3.1.2. Функциональная карта

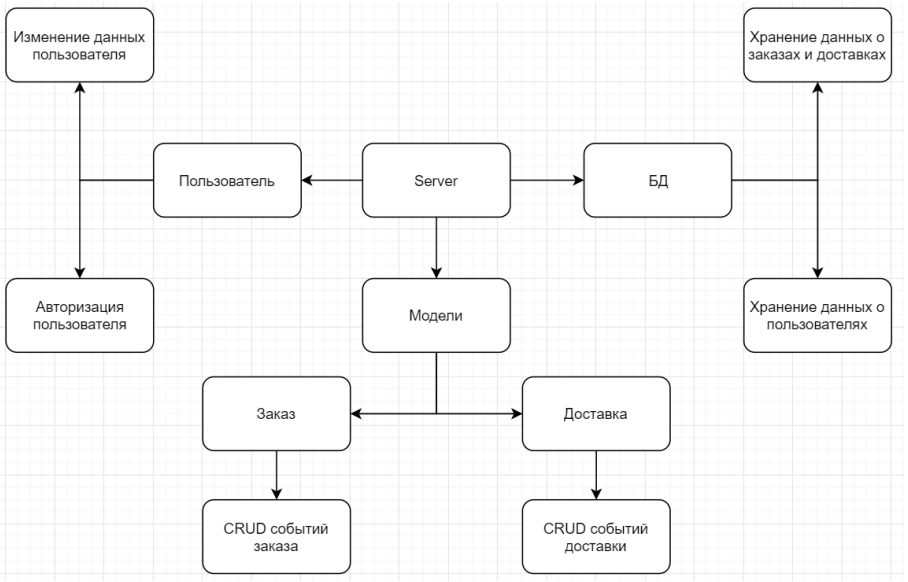

Рис. 1. Функциональная карта

3.1.3. Диаграмма использования

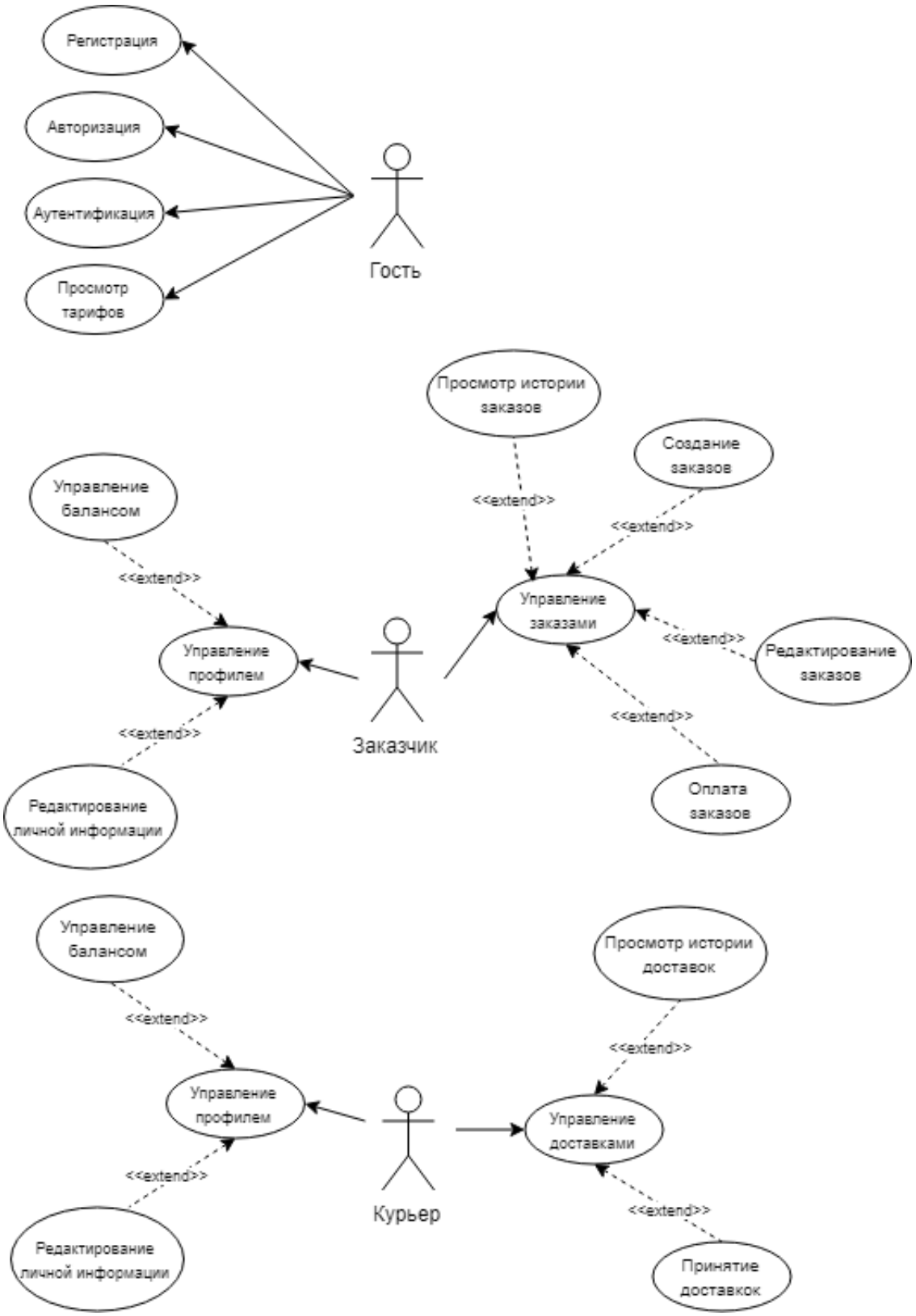

Рис. 2. Диаграмма вариантов использования

3.1.4. Функциональные требования

1. CRUD событий Заказа по запросу клиентской части от лица обычного пользователя.
2. CRUD событий Доставки по запросу клиентской части от лица курьера.
3. Просмотр собственной истории заказов и доставок от лица обычного пользователя.
4. Просмотр истории заказов и доставок всех пользователей от лица курьера.
5. Предоставление истории заказов/доставок за определённый период времени.

4. Нефункциональные требования

4.1. Требования к сохранности данных

Действия, применяемые для сохранности данных:

- Экранирование данных, приходящий с клиентской части, для избежания инъекций.
- Хеширование паролей и другой конфиденциальной информации
- Определенное максимальное количество запросов за определенный промежуток времени.
- Авторизация через систему JWT
- Сервис не должен хранить данные оплаты в чистом виде.

4.2. Используемые технологии

- Python
- Django / Django REST framework
- PostgreSQL
- Docker & Docker Compose