

Домашнее задание по дисциплине
“Разработка интернет-приложений”

Выполнил

студент группы ИУ5-55Б _____/Тазенков И.Д./

Москва, 2022 г.

1. Введение

Цель:

Создать интернет-приложение – онлайн магазин для продажи табачных изделий.

Задачи:

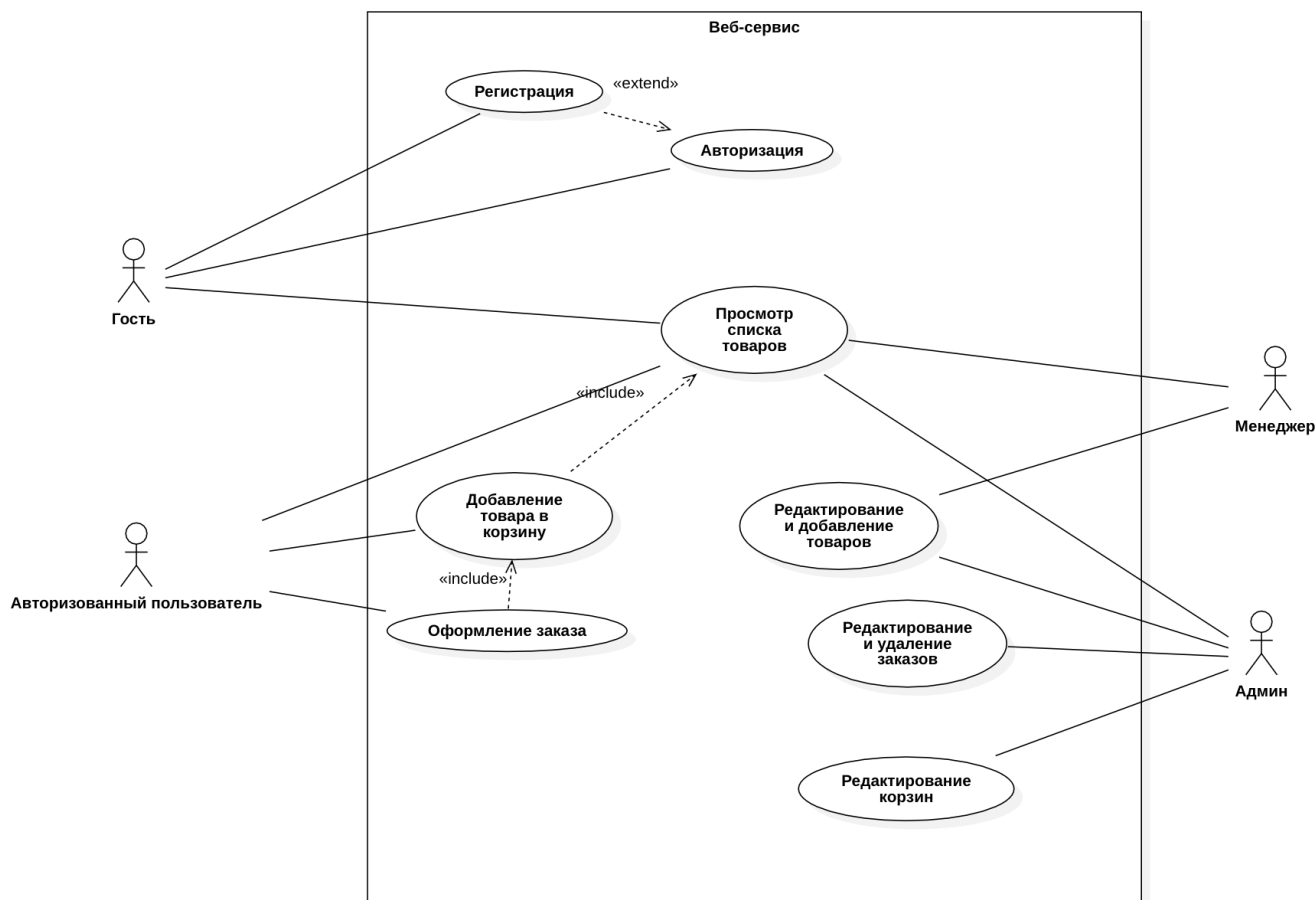
Реализовать интернет-приложение, включающее в себя веб-сервис и пользовательский интерфейс в виде сайта.

Актуальность:

В наше время люди стали ленивыми, и никто не хочет стоять в очереди в табачном магазине. Данное интернет-приложение решает эту проблему: достаточно оформить заказ на сайте и подойти к магазину, в котором весь заказ уже будет собран. На месте нужно будет только его оплатить.

2. Бизнес-процесс

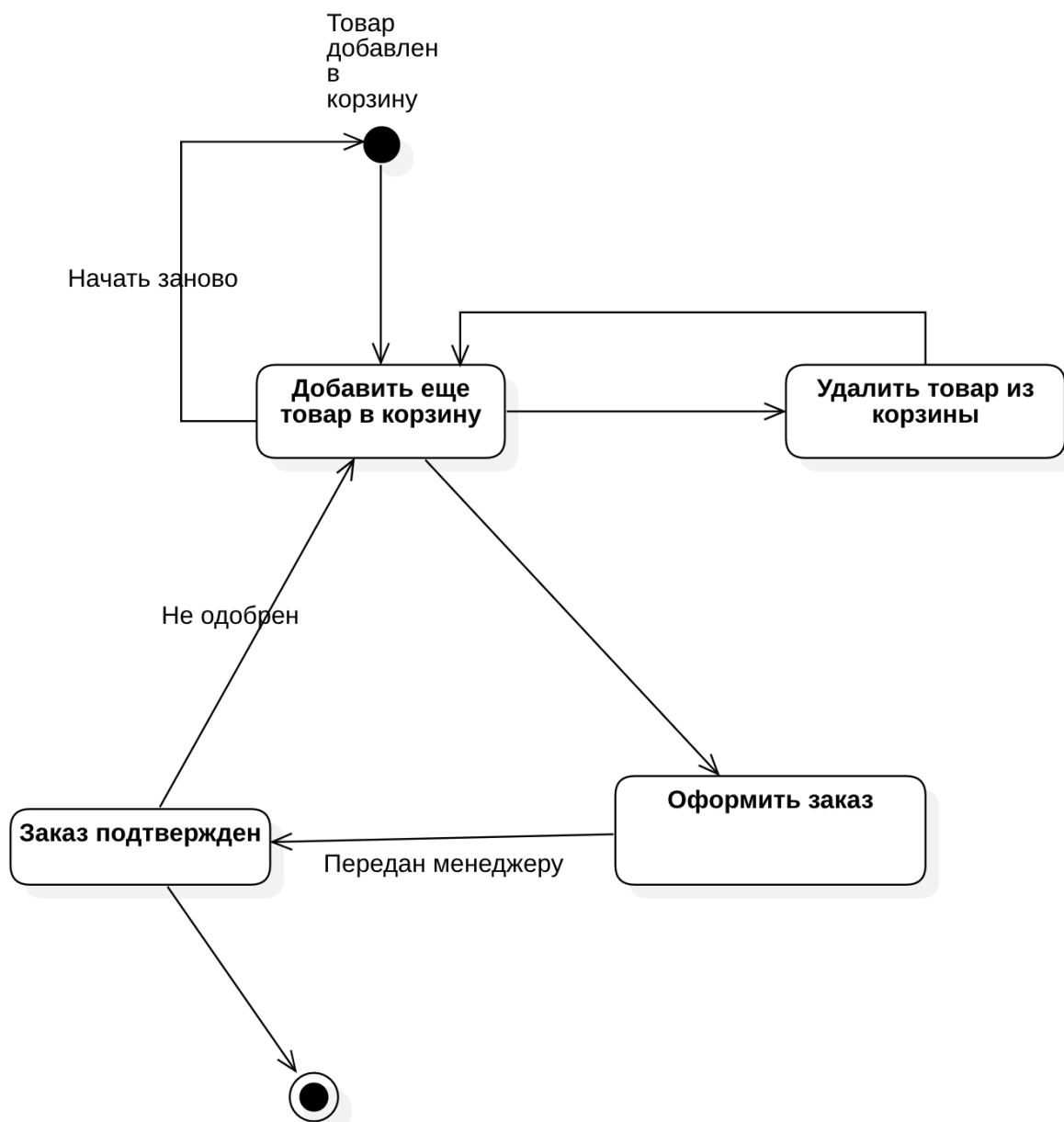
Диаграмма прецендентов:



Всего есть 4 вида пользователей:

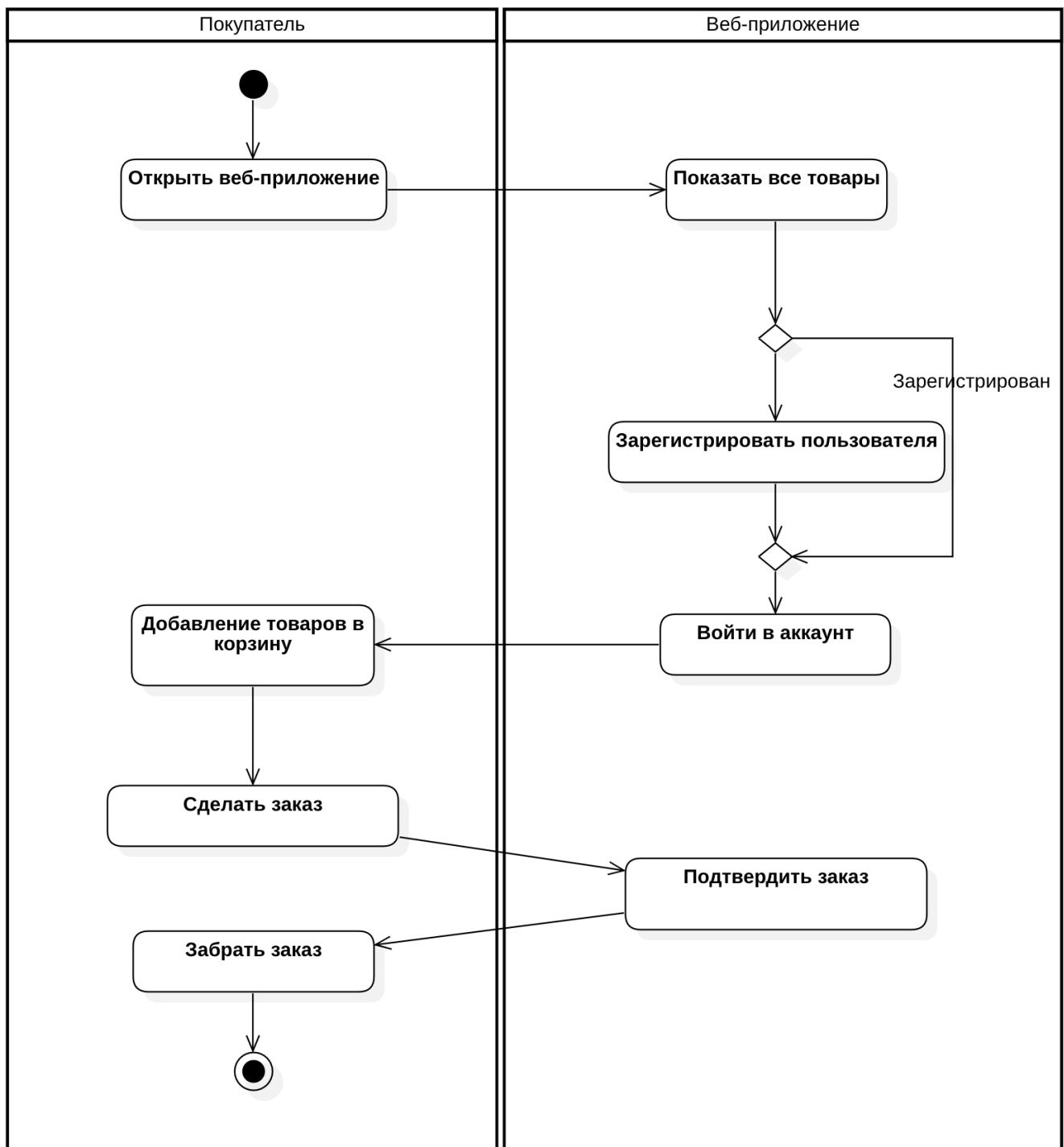
- Гость (анонимный пользователь) – может зарегистрироваться, войти в аккаунт либо просмотреть продукцию
- Авторизованный пользователь – может также просматривать товары, добавлять их в корзину, оформлять заказ
- Менеджер – может также просматривать товары, их редактировать и добавлять новую продукцию
- Админ – может удалять товары, редактировать и удалять заказы пользователей, редактировать корзины пользователей

Диаграмма состояний корзины:



Сначала товар добавляется в корзину. Далее можно еще добавлять товары в корзину. Также пользователь может удалять товары из корзины. Когда корзина сформирована, пользователь оформляет заказ. Менеджер получает информацию о заказе, может его одобрить или нет. Если заказ не одобрен, то пользователь либо удаляет товары из корзины, либо добавляет еще товары. И финальное состояние – когда заказ одобрен.

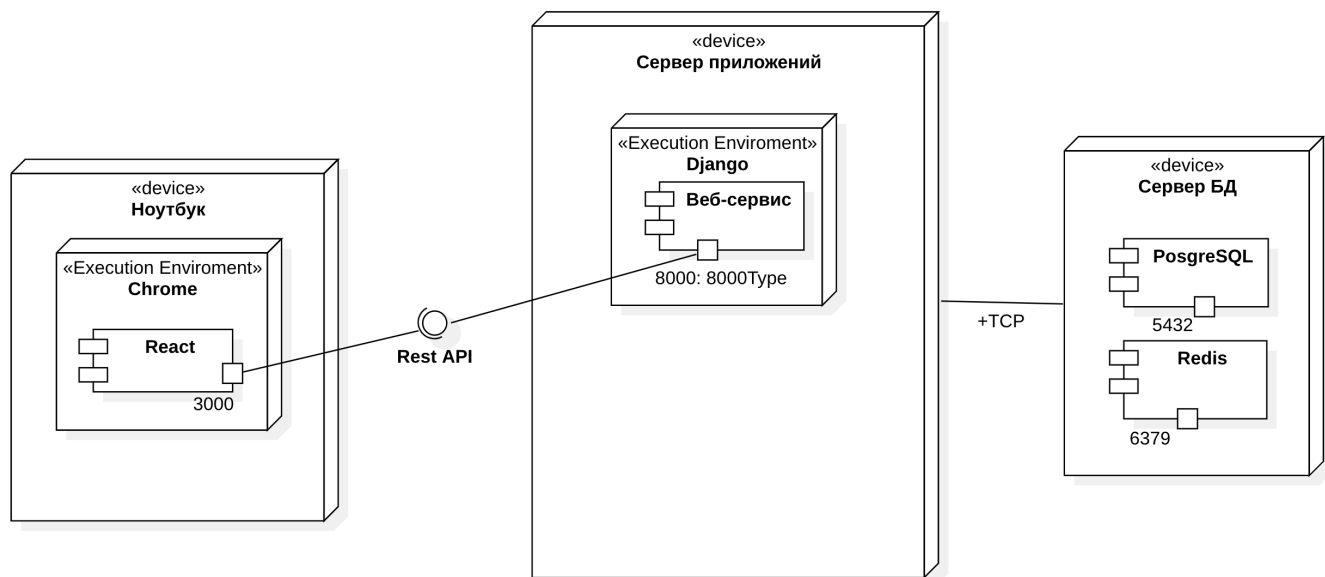
Диаграмма деятельности:



После открытия веб-приложения пользователю показываются все товары. Далее пользователь может зарегистрироваться либо войти в свой аккаунт. Когда пользователь авторизован, он может добавлять товары в корзину и оформлять заказ. Веб-приложение, а именно менеджер этот заказ подтверждает. После чего пользователь может забрать свой заказ в магазине.

3. Архитектура

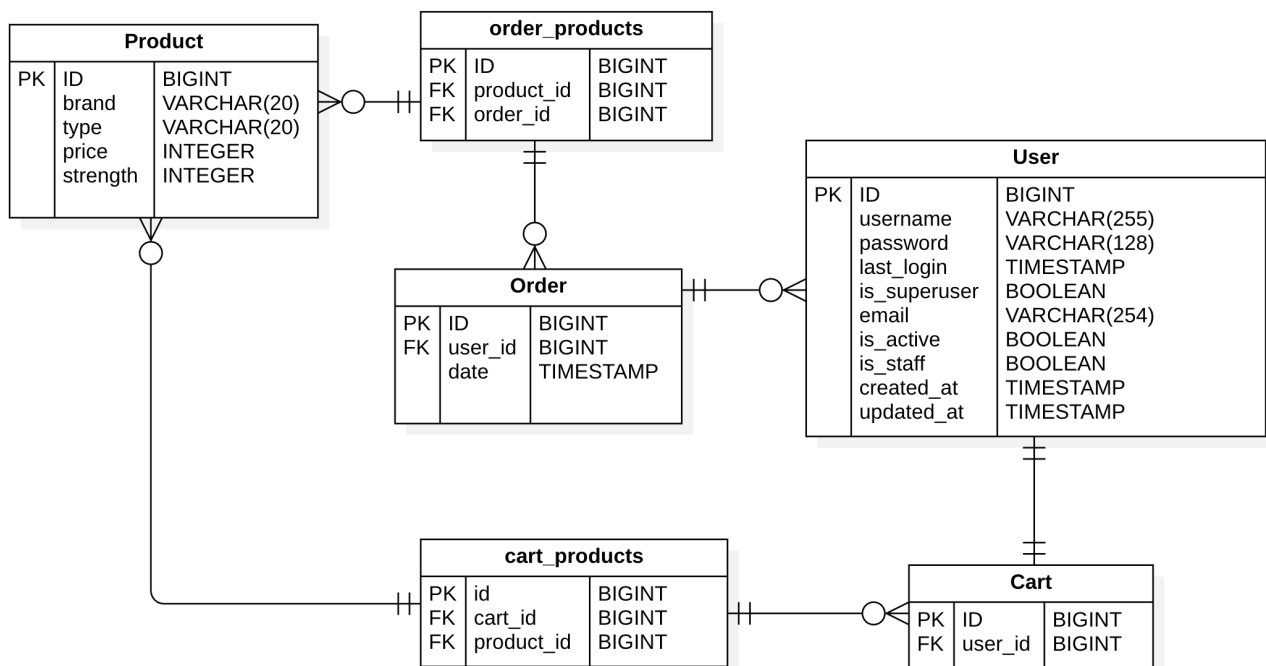
Диаграммы развертывания:



На сервере приложений лежит веб-сервис Django, который предоставляет порт 8000. На клиенте в браузере находится React-сервер, который обращается к веб-сервису на Django через порт 3000.

Веб-сервис с помощью протокола TCP обращается к серверу БД: PostgreSQL и Redis, предоставляющих 5432 и 6379 порты соответственно.

ER-диаграмма:



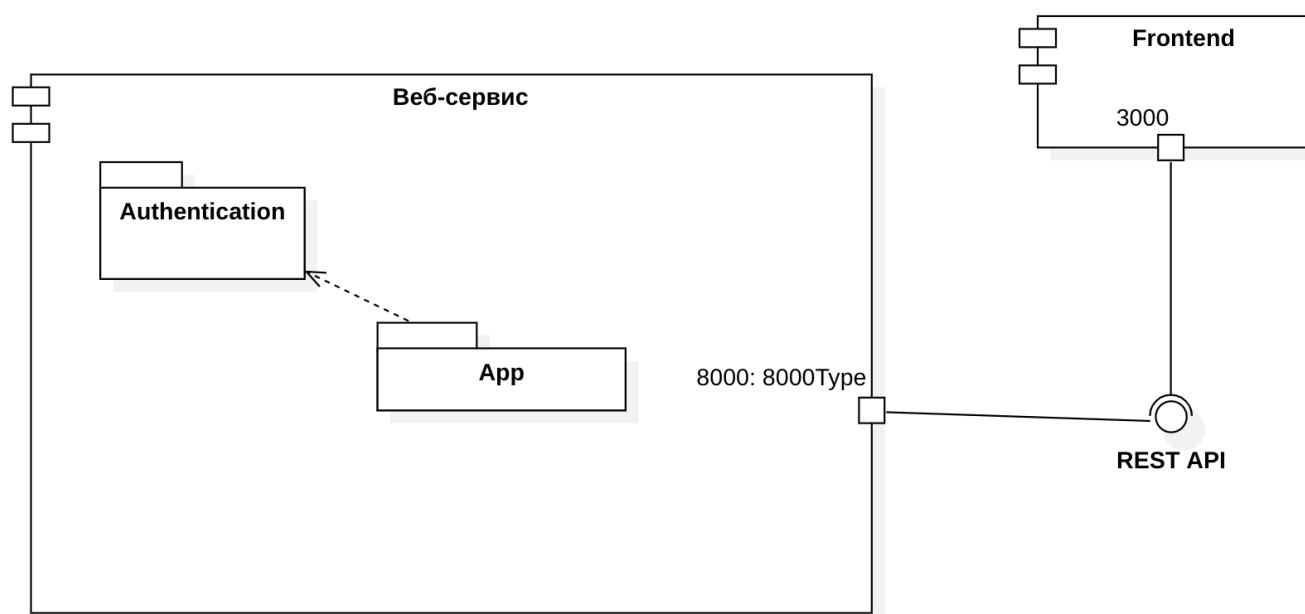
Сущность Product – товар для веб-сервиса, в данном случае – табачное изделие. В нем есть такие поля, как бренд, тип изделия, цена и никотиновая крепость.

Сущность Cart – пользовательская корзина, в ней хранятся товары пользователей через сущность посредник – cart_products. Так же в сущности корзины есть поле user_id – связь один-к-одному с сущностью пользователя – User.

Сущность User – пользователь. В этой сущности представлены такие поля как: имя пользователя, пароль, дата последнего входа в сервис, является ли пользователь администратором, является ли пользователь менеджером, почта, активен ли пользователь, даты создания и обновления аккаунта.

Так же есть сущность Order – заказ пользователя, в ней хранится дата заказа и связь один-к-одному с таблицей User. Также через сущность посредника order_products в таблице Order хранятся товары.

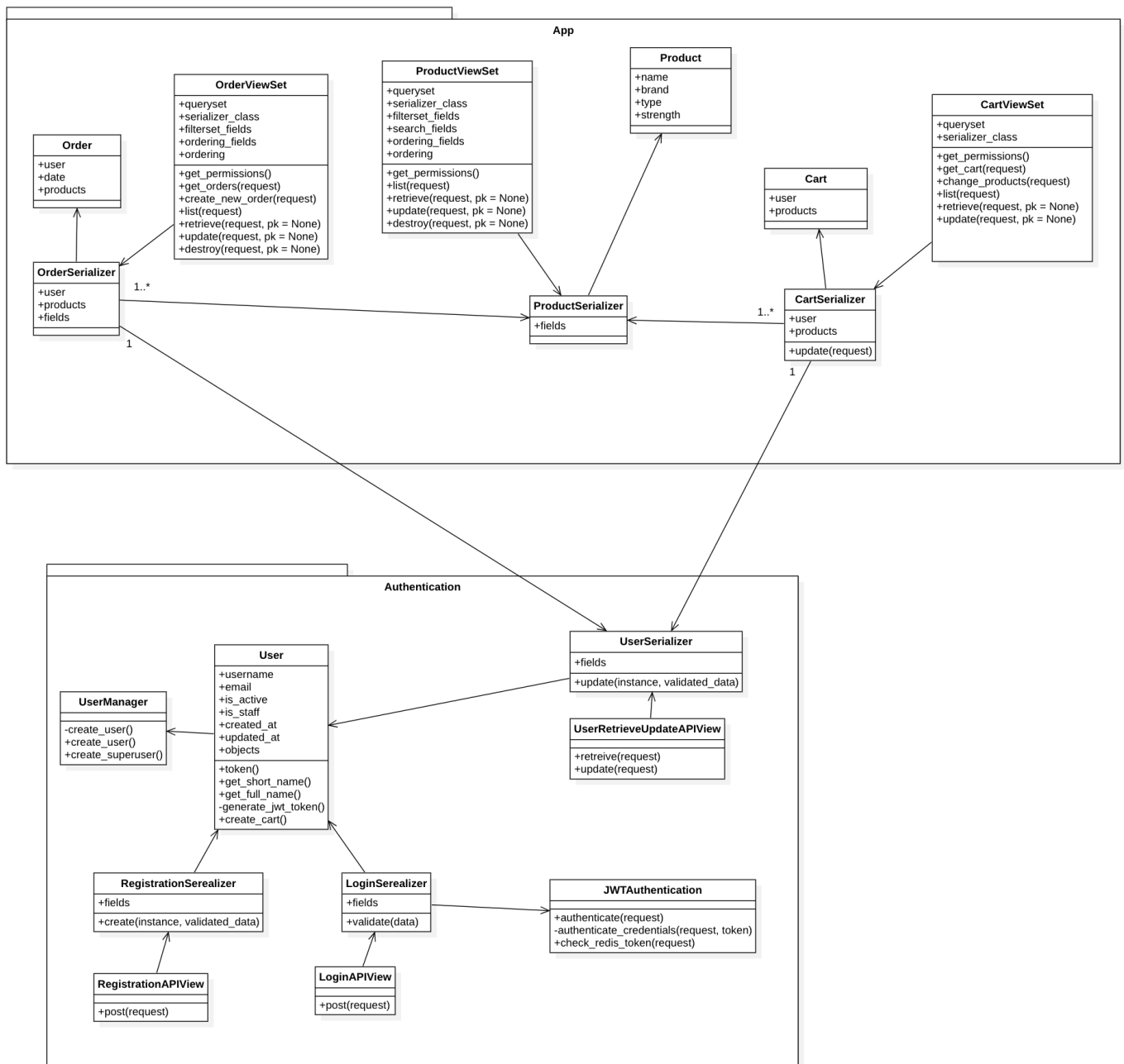
Диаграмма компонентов:



Интернет-приложение состоит из двух компонентов – веб-сервис и Frontend, которые передают данные через REST API протокол. Веб-сервис через 8000 порт, Frontend через 3000 порт.

Веб-сервис состоит из двух модулей: Authentication и App.

Диаграмма классов:



Модуль App состоит из классов-моделей из базы данных: Order, Product, Cart

У каждого из этих классов есть Serializer – класс, который сериализует и десериализует объекты. Также у каждого из этих классов есть свой ViewSet класс, который предоставляет API для запросов.

Модуль Authentication – модуль аутентификации пользователя.

Основным классом является User. Класс UserManager необходим для работы с Django ORM. UserSerializer – класс сериализатор объектов User.

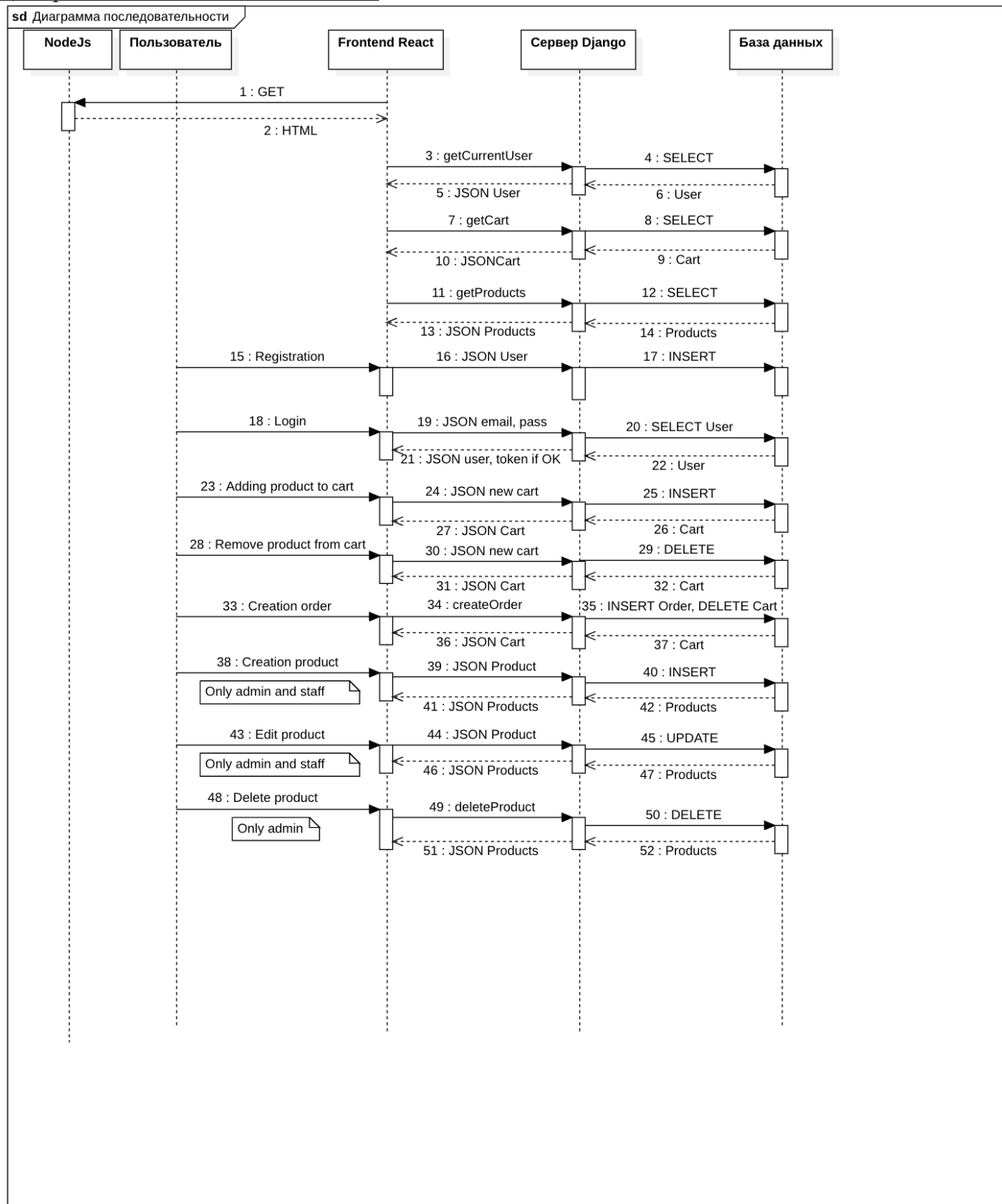
Также есть два других класса сериализатора объектов User – RegistrationSerializer и LoginSerializer. Они нужны для регистрации и авторизации пользователей.

Классы типа APIView предоставляют API для запросов.

Класс JWTAuthentication – класс аутентификации пользователя, используется в LoginSerializer для аутентификации пользователя по данным из запроса.

4. Алгоритмы

Диаграмма последовательности:



При запуске веб-приложения react запрашивает у сервера NodeJs HTML страницу, получая ее в ответ. Далее веб-приложение запрашивает у удаленного сервера с django на борту данные о текущем пользователе, его корзине и продуктах, чтобы отобразить эти данные не зависимо от того, авторизован ли пользователь, сервер делает SQL-запросы к базе данных, получает данные и отправляет их веб-приложению в формате

JSON. Пользователь может войти/зарегистрироваться, введя данные в форму, эти данные в формате JSON отправляются на удаленный сервер. Далее все действия сводятся к одному принципу, действие пользователя -> отправка запроса и данных в формате JSON веб-приложением на удаленный сервер, сервер делает SQL запросы к БД и возвращает данные все так же в формате JSON.

5. Описание интерфейса

Окна и их переназначение:

- Таблица всех товаров – показ всех товаров интернет-приложения, которые можно добавить в корзину.
- Информация о пользователе – просмотр и редактирование информации о пользователе.
- Корзина – просмотр товаров в корзине, а также их удаление из корзины.
- Добавление нового товара – это окно есть только у менеджеров и администраторов. Окно предназначено для заполнения информации о новом товаре и его публикации.
- Подробная информации о товаре – предназначено для просмотра подробной информации о товаре и добавления товара в корзину. Для менеджеров и администраторов в этом окне есть возможность отредактировать информацию о товаре.
- Авторизация и регистрация – предназначено для авторизации и регистрации пользователей.