**1. Определение целей и требований**

*Цели разработки продукта:*

- Создание удобного и доступного онлайн-магазина для пользователей, позволяющей находить и заказывать велосипеды интересующей их марки, а также аксессуаров к ним.

- Обеспечение безопасности и конфиденциальности данных пользователей.

- Автоматизация процессов управления товаров и обработка заказов.

*Основные функциональные требования:*

- Регистрация и аутентификация пользователей.

- Поиск и фильтрация товаров по различным параметрам (бренд, марка, цена).

- Оформление и управление заказами.

- Уведомления о статусе заказа (обработка, отправка, готов к получению, доставлено).

- Интеграция с системами оплаты для обработки транзакций.

- Управление запасами и автоматическое обновление статусов товаров.

- Сбор статистики об популярных товарах

- Рекомендация популярных товаров пользователям.

*Основные бизнес-кейсы:*

1. Регистрация и вход пользователей.

2. Поиск велосипедов, аксессуаров и добавление их в корзину.

3. Оформление заказа и выбор способа оплаты.

4. Отслеживание статуса заказа пользователем.

5. Управление запасами для сотрудников склада магазина.

6. Обработка возвратов и отмен заказов.

7. Сбор статистики об популярных брендах и марках

8. Отображение популярных товаров

*Требования к производительности, масштабируемости и надежности:*

- Возможность горизонтального масштабирования микросервисов по мере роста числа пользователей.

- Обеспечение высокой доступности и автоматическое восстановление после сбоев.

- Мониторинг производительности для быстрого реагирования на инциденты и предотвращения сбоев.

**2. Архитектурное проектирование**

*Микросервисная архитектура*:

1. Сервис аутентификации:

- Обработка регистрации и входа пользователей.

2. Сервис управления товарами:

- Поиск, добавление, обновление и удаление велосипедов и аксессуаров, сбор статистики, отображение популярных товаров.

3. Сервис заказов:

- Оформление, обновление и отслеживание статусов заказов.

4. Платежный сервис:

- Обработка платежей через интеграцию с платежными системами.

5. Сервис уведомлений:

- Отправка уведомлений пользователям о статусах заказов.

6. Сервис управления запасами:

- Отслеживание наличия товара на складе и автоматическое обновление.

*Брокер сообщений:*

- Выбор брокера: RabbitMQ.

- Роли в системе:

- Асинхронная обработка событий (например, уведомления о статусе заказа).

- Коммуникация между микросервисами (например, уведомление сервиса уведомлений о новом заказе).

- Обработка событий в реальном времени для управления товарами на складе.

**3. Выбор технологий**

*Технологический стек для микросервисов*:

1. Сервис аутентификации:

- Язык: C#

- База данных: PostgreSQL

- Фреймворк: ASP.NET Core

2. Сервис управления товарами:

- Язык: C#

- База данных: PostgreSQL

- Фреймворк: ASP.NET Core

3. Сервис заказов:

- Язык: C#

- База данных: PostgreSQL

- Фреймворк: ASP.NET Core

4. Платежный сервис:

- Язык: C#

- База данных: PostgreSQL

- Фреймворк: ASP.NET Core

5. Сервис уведомлений:

- Язык: C#

- База данных: PostgreSQL

- Фреймворк: ASP.NET Core

6. Сервис управления запасами:

- Язык: C#

- База данных: PostgreSQL

- Фреймворк: ASP.NET Core

*Конфигурация брокера сообщений (RabbitMQ):*

- Настройки:

- Количество очередей для разных типов сообщений (заказы, уведомления).

- Хранение сообщений: до 5 дней.

- Обработка сообщений с использованием "fanout" и "direct" обменников для различных сценариев.

Эта структура проекта обеспечит функциональность, масштабируемость и надежность онлайн-магазина, а также позволит легко добавлять новые функции в будущем.