**ИНТЕРНЕТ-МАГАЗИН**

В качестве темы выпускной работы взял «Реализация интернет-магазина с использованием микросервисов»

Используемые технологии:

1. Java
2. Spring
3. RabbitMQ
4. MongoDB

В качестве web-интерфейса Postman. Если хватит времени, то какой-нибудь фреймворк на java.

Сервисы покрывают основной процесс работы интернет-магазина – продажа и доставка товаров

**Пользовательские сценарии**

**Поиск товара**

Есть клиент и интернет-магазин

И клиент вводит ключевые слова для запроса поиска

Тогда клиенту возвращается список товаров подходящих под данный запрос

И пользователь видит следующие данные о товаре:

1. Название
2. Цена
3. Количество на складе
4. Цвет
5. Производитель

**Просмотр товара**

Когда клиент выбирает товар

Тогда клиент переходит на страничку с его описанием:

1. Название
2. Цена
3. Количество на складе
4. Цвет
5. Производитель
6. Дополнительные параметры актуальные для данного вида товара

**Добавление товара в корзину**

Когда клиент нажимает на кнопку "В корзину"

Тогда товар добавляется в корзину

**Просмотр корзины**

Когда клиент переходит в корзину товаров

Тогда он видит список всех добавленных в корзину товаров, их цену и количество

**Оформление заказа новым пользователем**

Есть неавторизованный клиент и список товаров в корзине

И клиент нажимает на кнопку "оформить заказ"

Тогда пользователю предлагается зарегистрироваться

И заполнить контактные данные о себе

После перенаправляет на страницу оформления заказа

**Оформление заказа зарегистрированным пользователем**

Есть авторизованный клиент и список товаров в корзине

И клиент нажимает на кнопку "оформить заказ"

Тогда товары бронируются на складе на 15 минут

И формируется платежная квитанция

**Оплата товара в течение 15 минут**

Есть авторизованный клиент, оформленный заказ и платежная квитанция

И клиент нажимает оплатить

Тогда заказ резервируется на складе

И поступает заявка на доставку заказа в сервис доставки

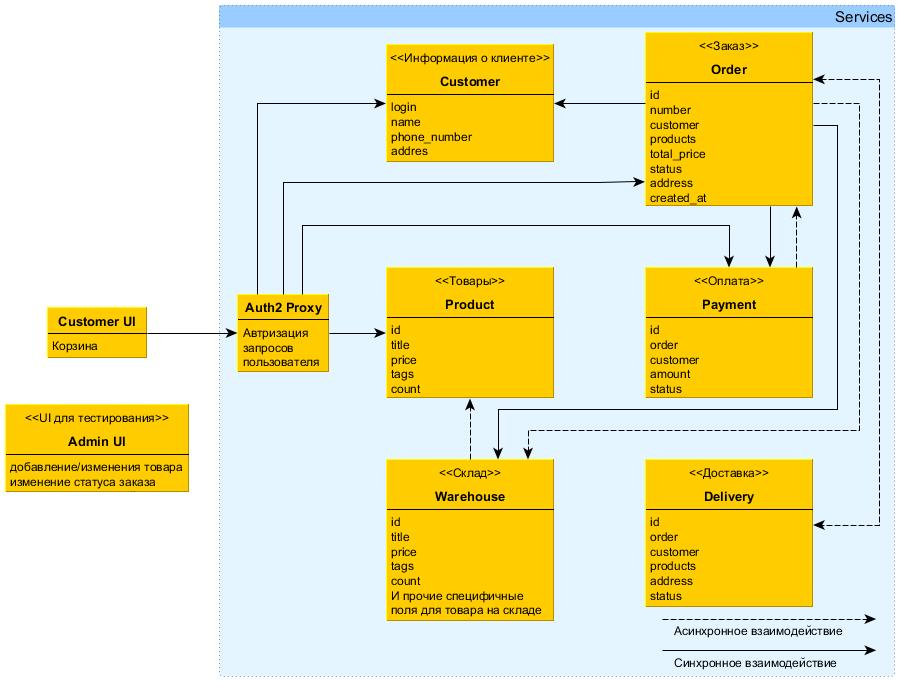
**Оплата товара через 15 минут**

Есть авторизованный клиент и неоплаченный заказ

И клиент нажимает на кнопку "оформить заказ"

Тогда смотри «**Оформление заказа зарегистрированным пользователем**»

**Общая схема взаимодействия сервисов**



**Customer UI**

UI для взаимодействия с интернет-магазинов. Кроме отправления запросов системе и вывода ответов, реализует логику управления корзиной пользователя.

**Admin UI**

UI для тестирования и эмуляция работы сотрудников магазина.

**Customer**

Сервис «Клиенты». Содержит необходимую информация для осуществления доставки.

Синхронное взаимодействие с сервисом:

1. Создание нового пользователя
2. Редактирование данных пользователя
3. Получение информации о пользователе

**Product**

Сервис «Товары». Содержит необходимые данные о товарах интернет-магазина для клиентов.

Синхронное взаимодействие с сервисом:

1. Поиск товаров по ключевым словам
2. Получение информации о товаре по id

Acинхронное взаимодействие с сервисом:

1. Добавление товара
2. Удаление товара
3. Обновление данных о товаре

Зависимости:

1. Потребление событий о товарах от сервиса Warehouse

**Warehouse**

Сервис «Склад товаров». Содержит полные данные о товарах интернет-магазина, такие как:

1. Название
2. Цена
3. Количество
4. Количество в резерве
5. Вес
6. Габариты

Синхронное взаимодействие с сервисом:

1. Добавление товара
2. Удаление товара
3. Обновление данных о товаре
4. Резервирование товара
5. Выдача товара

Асинхронное взаимодействие с сервисом:

1. Убрать товар из резерва

Публикует события:

1. Добавление товара
2. Удаление товара
3. Обновление данных о товаре

**Order**

Сервис «Заказы». Содержит информацию о товарах и статусе выполнения заказа клиента. Реализует паттерн «Сага» для процесса доставки купленных товаров пользователю, где является оркестратором.

Синхронное взаимодействие с сервисом:

1. Создать заказ
2. Удалить заказ
3. Получить активные заказы пользователя

Зависимости:

1. Отправка запроса «Зарезервировать товар» в Warehouse
2. Отправка сообщения «Убрать товар из резерва» в Warehouse
3. Отправка запроса «Создать оплату» в Payment
4. Потребление событий о статусе оплаты из Payment
5. Отправка сообщения «Создать доставку» в Delivery
6. Потребление событий о статусе доставки из Delivery

**Payment**

Сервис «Платежи». Содержит функционал для оплаты заказов.

Синхронное взаимодействие с сервисом:

1. Создать платеж
2. Оплатить

Публикует события:

1. Обновление статуса о платеже

**Delivery**

Сервис «Доставка». Сервис для отслеживания процесса доставки.

Асинхронное взаимодействие с сервисом:

1. Создать запрос на доставку

Публикует события:

1. Обновление статуса о доставке