**Task 2. Описать микросервисную архитектуру Cайта онлайн магазина**

**Описание магазина**

1.       У магазина должен быть свой сайт, оформленный в цветах брендбука.

2.       Пользователь может видеть товары, сгруппированные по разделам (бакалея, молочные продукты, мясные продукты и прочее).

3.       Пользователь может видеть весь перечень товаров, участвующих в акции на текущий период. Пользователь может добавлять товары в корзину.

4.       Пользователь может писать отзывы на товары, которые он ранее приобретал.

5.       Пользователь может видеть сумму стоимости товаров, которые сложил в корзину с учётом скидки.

6.       Пользователь может ввести номер своей дисконтной карты при оформлении заказа и стоимость заказа будет автоматически пересчитана.

7.       Пользователь может указать адрес доставки или выбрать самовывоз при выборе способа доставки.

8.       Пользователь видит либо режим работы пункта выдачи, либо диапазоны времени, когда может приехать курьер заказом.

**Дополнительные фичи**

1.       Пользователь может оплатить товар через ЕРИП

2.       Пользователь может оплатить товар через Интернет-банк

**Обоснование выбора архитектуры**

**Высокая масштабируемость**

Приложение будет разбито на небольшие независимые компоненты (Микросервисы), каждый из которых выполняет конкретную функцию. Это позволит горизонтально масштабировать отдельные сервисы и гибко управлять ресурсами. Сервис корзины или оплаты будет масштабироваться независимо от сервиса каталога товаров, что особенно важно во время пиковых нагрузок (например, в период акций и праздников).

**Высокая производительность приложения**

Когда пользователи или запросы к приложению вырастут (как заложено в стратегии), добавление микросервисов будет реализовано путем размещения их на дополнительных серверах. Это позволит легко управлять нагрузкой и распределять трафик.

**Отказоустойчивость.**

Отказ одного сервиса не приводит к остановке системы в целом. Если одна функция выходит из строя, это не останавливает всю систему, а требует обслуживания конкретного сервиса. Например, если сервис отзывов временно недоступен, это не повлияет на возможность пользователей просматривать товары или оформлять заказы.

**Высокая скорость разработки и внедрения нового функционала**

Микросервисы позволят разработчикам работать над отдельными сервисами независимо друг от друга. На знакомство с приложением уходит пара дней: специалист получает задачу, быстро в нее погружается, делает продуктовую версию, тестирует, выпускает в релиз.

**Отсутствие ограничений в стеке технологий**

Микросервисы могут сочетать разные технологии и языки программирования. Можно использовать параллельно разные команды разработчиков.

**Удобство интеграции с внешними системами**

Микросервисы легко интегрируются с внешними API (например, платежными системами, службами доставки). Это позволяет быстро добавлять новые функции, такие как оплата через ЕРИП или интеграция с курьерскими службами (возможность быстрой смены или работы нескольких служб).

**Улучшенная производительность**

Микросервисы позволят распределить нагрузку между несколькими серверами, что повысит общую производительность системы. Например, сервис каталога товаров будет оптимизирован для быстрого поиска, а сервис оплаты — для обработки транзакций.

**Основные микросервисы:**

1. Сервис каталога товаров (Product Catalog Service)

Отвечает за хранение и отображение информации о товарах.

Группирует товары по категориям (бакалея, молочные продукты, мясные продукты и т.д.).

Предоставляет API для поиска товаров, фильтрации и получения деталей товара.

Интегрируется с сервисом акций для отображения товаров, участвующих в акциях.

2. Сервис корзины (Shopping Cart Service)

Управляет добавлением, удалением и изменением товаров в корзине пользователя.

Рассчитывает общую стоимость товаров с учётом скидок и акций.

Интегрируется с сервисом скидок и дисконтных карт для пересчёта стоимости.

3. Сервис акций и скидок (Promotion Service)

Управляет акциями, скидками и специальными предложениями.

Предоставляет информацию о текущих акциях для отображения на сайте.

Интегрируется с сервисом корзины для применения скидок.

4. Сервис отзывов (Review Service)

Позволяет пользователям оставлять отзывы на товары, которые они приобрели.

Хранит и отображает отзывы на страницах товаров.

Интегрируется с сервисом заказов для проверки, что пользователь действительно покупал товар.

5. Сервис заказов (Order Service)

Управляет процессом оформления заказов.

Обрабатывает выбор способа доставки (самовывоз или курьерская доставка).

Интегрируется с сервисом доставки для расчёта времени доставки.

6. Сервис доставки (Delivery Service)

Управляет информацией о пунктах выдачи и времени работы курьеров.

Предоставляет API для получения доступных временных диапазонов доставки.

Интегрируется с сервисом заказов для обновления статуса доставки.

7. Сервис оплаты (Payment Service)

Обрабатывает платежи через различные системы (ЕРИП, Интернет-банк и другие).

Интегрируется с сервисом заказов для подтверждения оплаты.

Поддерживает возвраты и обработку ошибок при оплате.

8. Сервис пользователей (User Service)

Управляет учётными записями пользователей.

Обрабатывает авторизацию и аутентификацию.

Хранит информацию о дисконтных картах и их привязке к пользователям.

9. Сервис дисконтных карт (Loyalty Service)

Управляет дисконтными картами и программами лояльности.

Применяет скидки при оформлении заказа, если пользователь ввёл номер дисконтной карты.

Интегрируется с сервисом корзины и заказов.

 10. Сервис авторизации (Auth Service)

Проверяет права доступа пользователя к определённым ресурсам (доступ к админ-панели).

Проверяет логин и пароль.

Генерация и валидация токенов (JWT).

**Сценарии пользователей в системе:**

Пользователь просматривает товары:

Пользователь заходит на сайт, выбирает категорию "Молочные продукты".

Сервис каталога товаров отображает список товаров.

Пользователь видит товары, участвующие в акции, благодаря интеграции с Сервисом акций.

Пользователь добавляет товары в корзину:

Пользователь выбирает товар и нажимает "Добавить в корзину".

Сервис корзины добавляет товар и рассчитывает общую стоимость с учётом скидок, запрашивая данные у Сервиса акций.

Пользователь оформляет заказ:

Пользователь переходит к оформлению заказа, выбирает способ доставки (самовывоз).

Сервис заказов взаимодействует с Сервисом доставки для отображения времени работы пункта выдачи.

Пользователь вводит номер дисконтной карты, и Сервис заказов запрашивает скидку у Сервиса дисконтных карт.

Пользователь оплачивает заказ:

Пользователь выбирает оплату через Интернет-банк.

Сервис оплаты обрабатывает платеж и уведомляет Сервис заказов об успешной оплате.

Пользователь получает подтверждение заказа.

Администратор добавляет новую акцию:

Администратор через админ-панель добавляет новую акцию в Сервис акций.

Сервис акций обновляет информацию, и товары с акцией отображаются на сайте.

Курьер проверяет заказы на доставку:

Курьер заходит в систему через мобильное приложение, интегрированное с Сервисом доставки.

Сервис доставки отображает список заказов, которые нужно доставить в текущий день.

Привет. Чекнула.

Смотри не нашла у тебя основною штуку - расчет самой стоимости товара(которые не на акции, а просто товаров).

Подумай ,куда ее лучше впихнуть.

Может сделать сервис ценообразования + склад товаров (а там пилить акции, скидки и расчет основной цены (включая ограничения по законодательству, например ограничение максимальной торговой надбавки и учет НДС и сами товары с остатками, тогда удобнее показывать на фронте весь каталог с актуальными ценами из одного МС) - это как вариант "на подумать"

И сервис дисконтных карт... Давай подумаем, может есть смысл склеить дисконтные карты с самим клиентами, т.е наличие дк это будет какой-то признак или характеристика, относящаяся к клиенту, ну это тоже как вариант "на подумать".

А так все ок. Домашка принята)

Привет. 1. По моему сценарию расчёт стоимости заказа делает Сервис корзины. А расчет стоимости товара я не предполагал, думая что в каталоге уже есть это стоимость, и ее не надо рассчитывать.Не знаю как правильно или как на практике (опыта нет). 2. Добавляю сервис авторизации (подсмотрел на лекции)

3. Сервис дисконтных карт выделил отдельно, только потому что хотел максимально разбить и достичь "атомарности". Если это не логично или плохая практика, то ок, склею с сервисом пользователей

Ага, смотри в рил жизни это происходит так.

Товар всегда приходит по ТТН(ТН), в ТТН(ТН) указана цена товара, потом столбец с НДС.

При приемке этого товара на остатки(склад) магазина происходит расчет окончательной стоимости товара для реализации:

цена из накладной + 20 (10)% НДС + торговая надбавка.

Торговая надбавка на определенные категории своя (колбаса до 30%, хлеб до15%, на овощи и фрукты белорусского происхождения она вообще зафиксирована в бел рублях, например капуста до 0,5 рубля, больше нельзя по законодательству).

И вот мы получили стоимость самого товара, а дальше прикручиваем акции, скидки и прочее.

Если решил делать это в каталоге, то ок, но я бы лучше это делала на складе, потому что когда товар приходит по новой стоимости - нам надо продавать часть товара по старой цене, а часть по новой , тогда происходит переоценка после продажи остатков по старой цене, а это можно увидеть, только зная остатки.