

User stories

Критичный сценарий:

Я как клиент

Хочу авторизоваться

Тогда я смогу пользоваться системой

Я как клиент

Хочу получить список товаров

Тогда я смогу найти нужный мне

Я как клиент

Хочу искать товары по параметрам

Тогда я смогу найти нужный мне быстрее

Я как клиент

Хочу посмотреть детальное описание товара

Тогда я смогу найти нужный мне

Я как клиент

Хочу добавить/удалить товар в корзину

Тогда я смогу сформировать заказ

Я как клиент

Хочу получить список товаров в корзине

Тогда я смогу его проанализировать

Я как клиент

Хочу получить варианты доставки

Тогда я смогу выбрать подходящий

Я как клиент

Хочу указать адрес доставки

Тогда я смогу указать удобный для меня адрес

Я как клиент

Хочу оформить заказ

Тогда я смогу получить товар

Я как клиент

Хочу оплатить заказ

Тогда мой заказ будет окончательно оформлен

Я как клиент

Хочу получить подтверждение заказа на email

Тогда я буду знать, что мой заказ оформлен

Системные действия

Клиент авторизовывается

Клиент получает список товаров

Клиент получает товары по параметрам

Клиент получает детальное описание товара

Клиент добавляет/удаляет товар в корзину

Клиент получает список товаров в корзине

Клиент получает варианты доставки

Клиент выбирает вариант доставки

Клиент указывает адрес доставки

Клиент оформляет заказ

Клиент оплачивает заказ

Приложение резервирует товар на складе

Приложение отправляет подтверждение заказа на email Клиента

Приложение передает заказ в службу доставки

Модель на основе ООП

Основные сущности:

Клиент
Товар
Корзина Заказ
Доставка / Служба доставки
Оплата / Платежный сервис
Склад
Подтверждение на email



Сервисы:

ClientService? UserManagement? CustomerManagement
OrderManagement
OfferingCatalog
WarehouseManagement
PaymentService
DeliveryService
NotificationService

Функциональная модель



Сервисы:

OfferingSearchService

OrderManagement

ReservationService

PaymentService

DeliveryService

NotificationService

Сервисы

Дальше работаем с моделью на основе ООП

Сервисы:

CustomerManagement

OrderManagement

OfferingCatalog

WarehouseManagement

PaymentService

DeliveryService

NotificationService

Системные действия в виде API

Клиент

GET /api/v1/customer/{id}

Заказ

GET /api/v1/orders/{id}

POST /api/v1/orders

POST /api/v1/orders/{id}/items

DELETE /api/v1/orders/{id}/items

PUT /api/v1/orders/{id}

Товар

GET /api/v1/offerings/search?query=FENDER CD-60SCE

GET /api/v1/offerings/{id}

Склад

POST /api/v1/reserve

Оплата

GET /api/v1/paymentMethods

POST /api/v1/pay

Доставка

GET /api/v1/deliveryOptions

CustomerManagement

Запросы:

	Request	Response
Получить информацию о клиенте	GET /api/v1/customer/{customerId}	{ "id":"123", "email":"alex@gmail.com", "firstName":"Alex", "lastName":"Xela", "defaultAddress":"Voronezh, Lenina 1", "defaultPaymentMethod":"Credit Card" }

Команды:

События:

Зависимости:

Вопросы:

OrderManagement

Запросы:

	Request	Response
Получить информацию о заказе	GET /api/v1/orders/{orderId}	{ "id":"123", "name":"Order#123", "totalPrice":555.22, "status":"In progress", "deliveryAddress":"Voronezh, Lenina 1", "paymentMethod": "CreditCard", "items":[{ "id":"123", "offeringId":"123", "price":222.22 }] }

Команды:

	Request	Response
Создать заказ	POST /api/v1/orders	{ "id":"123", "name":"Order#123", "totalPrice":0, "status":"In progress", "deliveryAddress":"Voronezh, Lenina 1", "paymentMethod": "Credit Card", "items":[] }
Добавить товар	POST /api/v1/orders/{orderId}/items { "offeringId":"123" }	{ ..., "items":[{ "id":"123", "offeringId":"123", "price":222.22 }] }
Удалить товар	DELETE /api/v1/orders/{orderId}/items/{itemId}	
Обновить адрес доставки	PUT /api/v1/orders/{id} { "deliveryAddress":"Voronezh, Lenina 1" }	{ "id":"123", "name":"Order#123", "totalPrice":0, "status":"In progress", }

	}	"deliveryAddress": "Voronezh, Lenina 1", "paymentMethod": "Credit Card", "items": [] }
Обновить метод оплаты	PUT /api/v1/orders/{id} { "paymentMethod": "Credit Card" }	{ "id": "123", "name": "Order#123", "totalPrice": 0, "status": "In progress", "deliveryAddress": "Voronezh, Lenina 1", "paymentMethod": "Credit Card", "items": [] }
Обновить статус заказа. (оформить, отменить)	PUT /api/v1/orders/{id} { "status": "Submitted" }	{ "id": "123", "name": "Order#123", "totalPrice": 0, "status": "Submitted", "deliveryAddress": "Voronezh, Lenina 1", "paymentMethod": "Credit Card", "items": [] }

События:

Смена статуса заказа: Оформление, Отмена
--

Зависимости:

CustomerManagement	Получение адреса доставки и метода оплаты по умолчанию
OfferingCatalog	Получение деталей о товаре. Цена товара

Вопросы:

Добавить более явные методы для смены статуса заказа?

Нужно ли ходить отдельно за дефолтными значениями в юзер менеджмент? — Нет. Пусть приходят при инициализации ордера

Если заказ бросили? Удалять? — Нет. Пусть хранится история пока что

OfferingCatalog

Запросы:

	Request	Response
Получить товар	GET /api/v1/offerings/{id}	{ "id": "123", "name": "Offering Name", "price": 222.22, "description": "Some Text", "available": 12 }
Поиск товара по параметрам	GET /api/v1/offerings/search ?query=OfferingName123	[{ "id": "123", "name": "Offering Name", "price": 222.22, "description": "Some Text" }, { ... }]

Команды:

События:

Зависимости:

WarehouseManagement	Слушает события на изменения количества доступных товаров
---------------------	---

Вопросы:

Поиск через эластик?
Проверять наличие товара на складе?

WarehouseManagement

Запросы:

Команды:

События:

Изменение количества доступных товаров
--

Зависимости:

OrderManagement	Слушает события изменения статуса заказа. Во время оформления заказа резервирует товар на складе
-----------------	--

Вопросы:

PaymentService

Запросы:

	Request	Response
Получение доступных методов оплаты	GET /api/v1/paymentMethods	[{ "id": "123", "name": "Credit Card", "description": "Some Text" }]

Команды:

События:

Заказ оплачен

Зависимости:

Вопросы:

Как именно выглядит способ оплаты? Это типы? Кредитка? Что и где хранится? Типы

DeliveryService

Запросы:

	Request	Response
Получение доступных методов доставки	GET /api/v1/deliveryOptions	[{ "id":"123", "name":"Courier", "description":"Some Text" }]

Команды:

События:

Зависимости:

PaymentService	Слушает статус оплаты. После оплаты передает заказ в доставку.
OrderManagement	Получает детальную информацию по заказу. Слушает изменение статуса заказа. После оплаты передает заказ в доставку
CustomerManagement	Получает детальную информацию по заказчику

Вопросы:

Какие типы доставки? Надо ли определить список? — Пока что только доставка домой курьером. В теории: Самовывоз

NotificationService

Запросы:

Команды:

События:

Зависимости:

PaymentService	Слушает изменение статуса заказа. Когда оплачен — отправляет оповещение
OrderManagement	Получает детальную информацию по заказу Слушает изменение статуса заказа. Когда оплачен – отправляет оповещение
CustomerManagement	Получает детальную информацию по заказчику. Как минимум его email

Вопросы:

Схема взаимодействия сервисов

