

### **Анализ ТЗ: функционал «Корзина» интернет-магазина**

1. Пользователь может добавить в корзину от 1 до 10 единиц одного товара.
2. Пользователь может изменить количество каждого товара в корзине не менее, чем до 1-го.

*Для удаления товара из корзины используется отдельная кнопка.*

3. В корзине может находиться не более 5 различных товаров.
4. Суммарное количество всех товаров в корзине не может превышать 20 штук.
5. Товары в корзине могут быть разные.
6. При попытке добавить товар, превышающий лимиты, система показывает сообщение:

*"Лимит корзины превышен".*

7. Цена на продукт фиксируется на момент добавления в корзину и не меняется.
8. На странице корзины отображается список товаров, их количество, цена за единицу и общая стоимость позиции.

9. Если пользователь уменьшает количество товара до 0, товар удаляется из корзины.

10. В корзине может быть реклама других продуктов.

11. Реклама товаров в корзине должна быть каждый будний день по утрам и вечерам.

13. Если цена на товар изменилась в каталоге, система должна автоматически обновить ее в корзине у всех пользователей.

*Мнение автора: честно говоря, это отвратительное ТЗ, которое проще переписать нормально, чем пытаться исправить*

### **1. Выявленные логические противоречия и недочёты**

#### **1.1 Логические противоречия**

**Противоречие №1: удаление товара из корзины (п.2 и п.9)**

- В п.2 указано, что количество товара нельзя уменьшить ниже 1, а удаление производится отдельной кнопкой.
- В п.9 сказано, что при уменьшении количества до 0 товар удаляется автоматически.

**Проблема:**

Требования описывают два взаимоисключающих сценария. Непонятно, допустимо ли значение 0 и какой механизм удаления является корректным.

Противоречие №2: логика работы с ценой (п.7 и п.13)

- П.7 утверждает, что цена фиксируется на момент добавления в корзину.
- П.13 требует автоматического обновления цены при изменении в каталоге.

Проблема:

Цена не может быть одновременно зафиксированной и автоматически обновляемой.

Это критичное противоречие, влияющее на расчёт стоимости заказа.

## 1.2 Недочёты и неясности требований

### а. Неопределённость при работе с лимитами (п.1, п.3, п.4)

Задано несколько ограничений:

- до 10 единиц одного товара,
- до 5 различных товаров,
- не более 20 единиц всего.

Проблема:

Не описано поведение системы, если превышает несколько лимитов одновременно, а также приоритет этих ограничений.

### б. Общее сообщение об ошибке (п.6)

Используется единый текст: «Лимит корзины превышен».

Проблема:

Пользователь не понимает, какое именно ограничение нарушено. Это ухудшает UX и усложняет тестирование граничных сценариев.

### с. Избыточное требование (п.5)

«Товары в корзине могут быть разные».

Проблема:

Требование не добавляет новой информации и дублирует п.3.

d. Неопределённость требований к рекламе (п.10, п.11)

Не указано:

- точное время «утра» и «вечера»,
- часовой пояс,
- поведение рекламы в другое время.

Проблема:

Требование невозможно корректно реализовать и протестировать без уточнений.

### 1.3 Формальные ошибки

- а. Отсутствует пункт 12.

Нарушена последовательность нумерации.

## 2. Исправленная и непротиворечивая версия ТЗ

1. Пользователь может добавить в корзину от 1 до 10 единиц одного товара.
2. В корзине может находиться не более 5 различных товаров.
3. Суммарное количество всех товаров в корзине не может превышать 20 единиц.
4. Пользователь может изменять количество товара в корзине в диапазоне от 1 до максимально допустимого значения.
5. Для удаления товара из корзины используется кнопка «Удалить».

6. При попытке добавить или изменить количество товара с нарушением ограничений операция блокируется и показывается сообщение об ошибке.
7. Система отображает сообщение с указанием конкретного нарушения:
- превышен лимит одного товара;
  - превышен лимит различных товаров;
  - превышен лимит общего количества.
8. Цена товара фиксируется на момент добавления в корзину и не изменяется до оформления заказа.
9. В корзине отображаются:
- название товара;
  - количество;
  - цена за единицу;
  - общая стоимость позиции;
  - итоговая стоимость корзины.
10. В корзине может отображаться рекламный блок с другими товарами.
11. Реклама отображается по будним дням с 08:00 до 11:00 и с 18:00 до 21:00 по местному времени пользователя.
12. Рекламные блоки не влияют на расчёт стоимости корзины.

### **3. Уточняющие вопросы к продукт-менеджеру / бизнес-заказчику**

#### **1. По цене**

Чем обусловлены требования о фиксации цены в корзине (п.8)

## 2. По рекламе

«утро» и «вечер» определяются по времени пользователя?

Должна ли реклама полностью скрываться вне указанных интервалов?

Чем обусловлены временные ограничения рекламы (п.11)?

## 3. По поведению корзины

Сохраняется ли корзина между сессиями и устройствами?

Что делать, если товар стал недоступен после добавления в корзину?

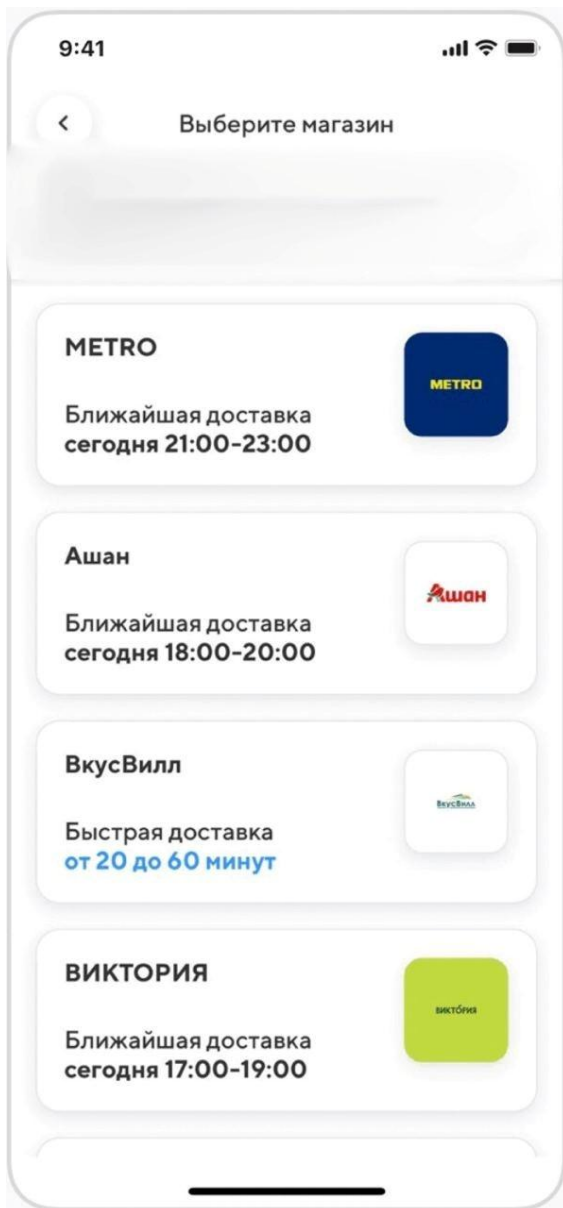
## 4. Системные

Чем обусловлены ограничения пользователя количеством разных товаров в корзине (п.2, п.3) и количеством конкретного товара (п.1)

## Задание 2: проектирование API

### Описание:

Интернет-магазин "Петрушка Зеленая" преуспевает, расширяется и в мобильном приложении решили создать новый экран, который будет отображать магазины партнеров (см. макеты ниже).



### Что нужно сделать:

1. Написать пример REST API запроса, который будет вызываться при переходе пользователя на данный экран.
2. Привести пример ответа этого REST API в соответствии с макетом. Формат - JSON. Учесть, что при клике на плашку магазина должен осуществляться переход по ссылке на внешний ресурс.

## 1. Пример REST API запроса

Метод: GET

Описание: Получение списка магазинов-партнёров для экрана выбора магазина.

### Endpoint

GET /api/v1/partners/stores

### Query-параметры

city\_id=123

lat=55.7558

lon=37.6173

Параметр	Тип	Описание
city_id	number	Город пользователя
lat / lon	number	Координаты для расчёта доставки

### HTTP-заголовки

Authorization: Bearer <access\_token>

Accept-Language: ru-RU

X-App-Version: 5.12.0

X-Platform: ios

## 2. Пример ответа REST API (JSON)

```
{
  "success": true,
  "data": {
    "city_id": 123,
    "generated_at": "2025-12-24T15:10:00+03:00",
    "stores": [
      {
        "id": "metro",
        "name": "METRO",
        "logo_url": "https://cdn.petrushka.ru/logos/metro.png",
        "external_url": "https://online.metro-cc.ru",

        "availability": {
          "is_available": true,
```

"reason": null

},

"delivery": {

"type": "scheduled",

"nearest\_delivery\_at": "2025-12-24T21:00:00+03:00",

"window": {

"from": "2025-12-24T21:00:00+03:00",

"to": "2025-12-24T23:00:00+03:00"

}

},

"display": {

"title": "Ближайшая доставка",

"subtitle": "сегодня 21:00–23:00",

"badge": null

},

"sort\_order": 1

},

{

"id": "auchan",

"name": "Ашан",

"logo\_url": "https://cdn.petrushka.ru/logos/auchan.png",

"external\_url": "https://www.auchan.ru",

"availability": {

"is\_available": true,

"reason": null

},

"delivery": {

"type": "scheduled",

"nearest\_delivery\_at": "2025-12-24T18:00:00+03:00",

"window": {

```
    "from": "2025-12-24T18:00:00+03:00",  
    "to": "2025-12-24T20:00:00+03:00"  
  }  
},
```

```
"display": {  
  "title": "Ближайшая доставка",  
  "subtitle": "сегодня 18:00–20:00",  
  "badge": null  
},
```

```
"sort_order": 2  
},  
{  
  "id": "vkusvill",  
  "name": "ВкусВилл",  
  "logo_url": "https://cdn.petrushka.ru/logos/vkusvill.png",  
  "external_url": "https://vkusvill.ru",
```

```
"availability": {  
  "is_available": true,  
  "reason": null  
},
```

```
"delivery": {  
  "type": "express",  
  "eta_minutes": {  
    "from": 20,  
    "to": 60  
  }  
},
```

```
"display": {  
  "title": "Быстрая доставка",  
  "subtitle": "от 20 до 60 минут",
```

```
    "badge": "FAST"
  },

  "sort_order": 3
},
{
  "id": "victoria",
  "name": "ВИКТОРИЯ",
  "logo_url": "https://cdn.petrushka.ru/logos/victoria.png",
  "external_url": "https://victoria-group.ru",

  "availability": {
    "is_available": true,
    "reason": null
  },

  "delivery": {
    "type": "scheduled",
    "nearest_delivery_at": "2025-12-24T17:00:00+03:00",
    "window": {
      "from": "2025-12-24T17:00:00+03:00",
      "to": "2025-12-24T19:00:00+03:00"
    }
  },

  "display": {
    "title": "Ближайшая доставка",
    "subtitle": "сегодня 17:00–19:00",
    "badge": null
  },

  "sort_order": 4
}
]
```

}

### Задание 3: архитектура

**Описание:**

Заказчик хочет, что в мобильное приложение интернет-магазина "Петрушка Зеленая" начали приходить пуши. Они могут быть разными: о том, что заказ слишком долго лежит без действий в корзине, об отмене заказа, рекламные рассылки и прочее. То есть нужен функционал, а какие пуши отправлять - точно найдется.

**Что нужно сделать:**

Построить верхнеуровневую архитектурную схему - как должна работать отправка PUSH уведомлений в данном приложении. Можно просто в виде блок схем. Считаем, что на бэкенде микросервисная архитектура. В данном задании рекомендуется в интернете изучить архитектуры подобных решений.  
Быть готовым на собеседовании обсудить эту схему.

**Решение**

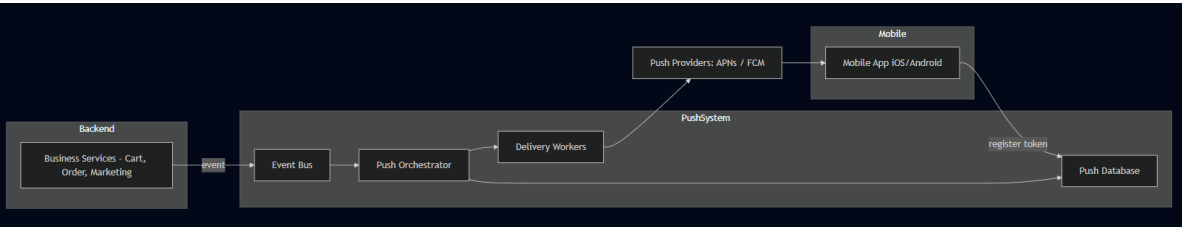


Рис.1 Архитектурная схема отправки PUSH-уведомлений

Название модуля	Что выполняет / за что отвечает
Mobile App (iOS/Android)	Регистрирует устройство для уведомлений, получает уведомления, отображает их пользователю
Push Providers (APNs / FCM)	Доставляет уведомления на устройство, обрабатывает TTL (время жизни) и ошибки доставки
Backend/ Business Services	Генерируют события, которые могут привести к отправке уведомления
Event Bus	Асинхронная шина событий: принимает события от сервисов, передаёт их в Push Orchestrator
Push Orchestrator	Координирует обработку событий, формирует уведомления, определяет получателей, применяет бизнес-правила
Push Database	Хранение идентификаторов устройств, истории уведомлений, информации о пользователях и статусах доставок
Delivery Workers	Берут сформированные уведомления и отправляют их через Push Providers; могут быть масштабируемыми

Таблица 1 — Описание модулей системы пуш-уведомлений

## Краткая логика работы системы

- Регистрация устройства:

Пользователь открывает приложение → мобильное приложение получает идентификатор устройства → отправляет его в Push Database.

- Генерация события:

Событие возникает в бизнес-сервисе (Cart / Order / Marketing) → отправляется в Event Bus.

- Обработка события:

Event Bus направляет событие в Push Orchestrator → оркестратор определяет, кто должен получить уведомление, применяет бизнес-правила (частота, тип уведомления, локализация), формирует текст/шаблон.

- Подготовка к отправке:

Orchestrator сохраняет информацию в Push Database (при необходимости) и передаёт уведомление в Delivery Workers.

- Доставка уведомления:

Delivery Worker отправляет уведомление через Push Provider (APNs / FCM) → Push Provider доставляет уведомление на устройство пользователя.

- История и аналитика:

Статус доставки может сохраняться в Push Database для аналитики и повторной отправки в случае ошибок.