

Задание 1. User Story

User Story:

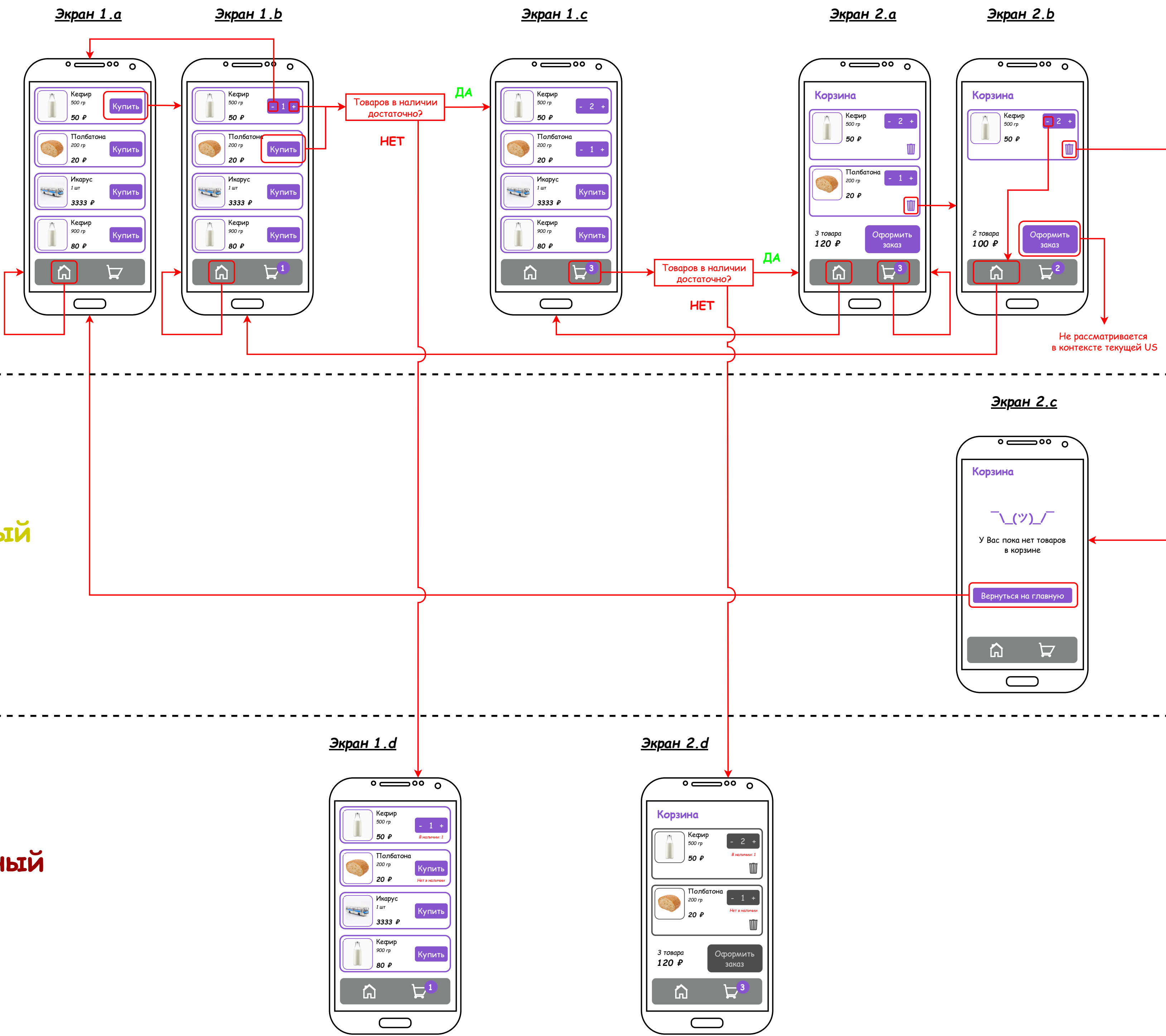
*Как пользователь приложения [МегаМаркет],
Я хочу иметь возможность управлять содержимым корзины,
Чтобы менять состав заказа в любое время до его оплаты.*

Задание 2. Макеты

Основной
сценарий

Альтернативный
сценарий

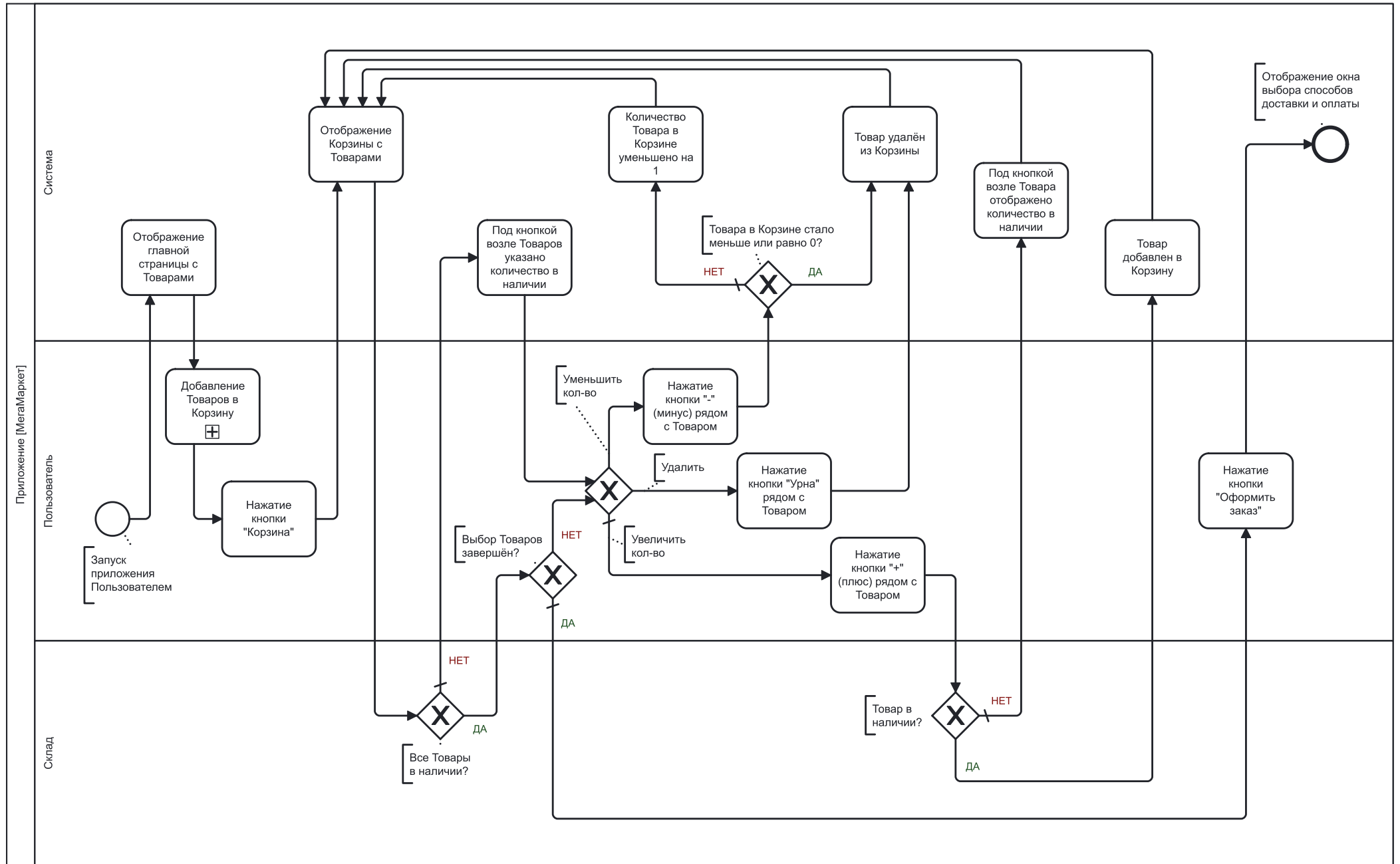
Исключительный
сценарий



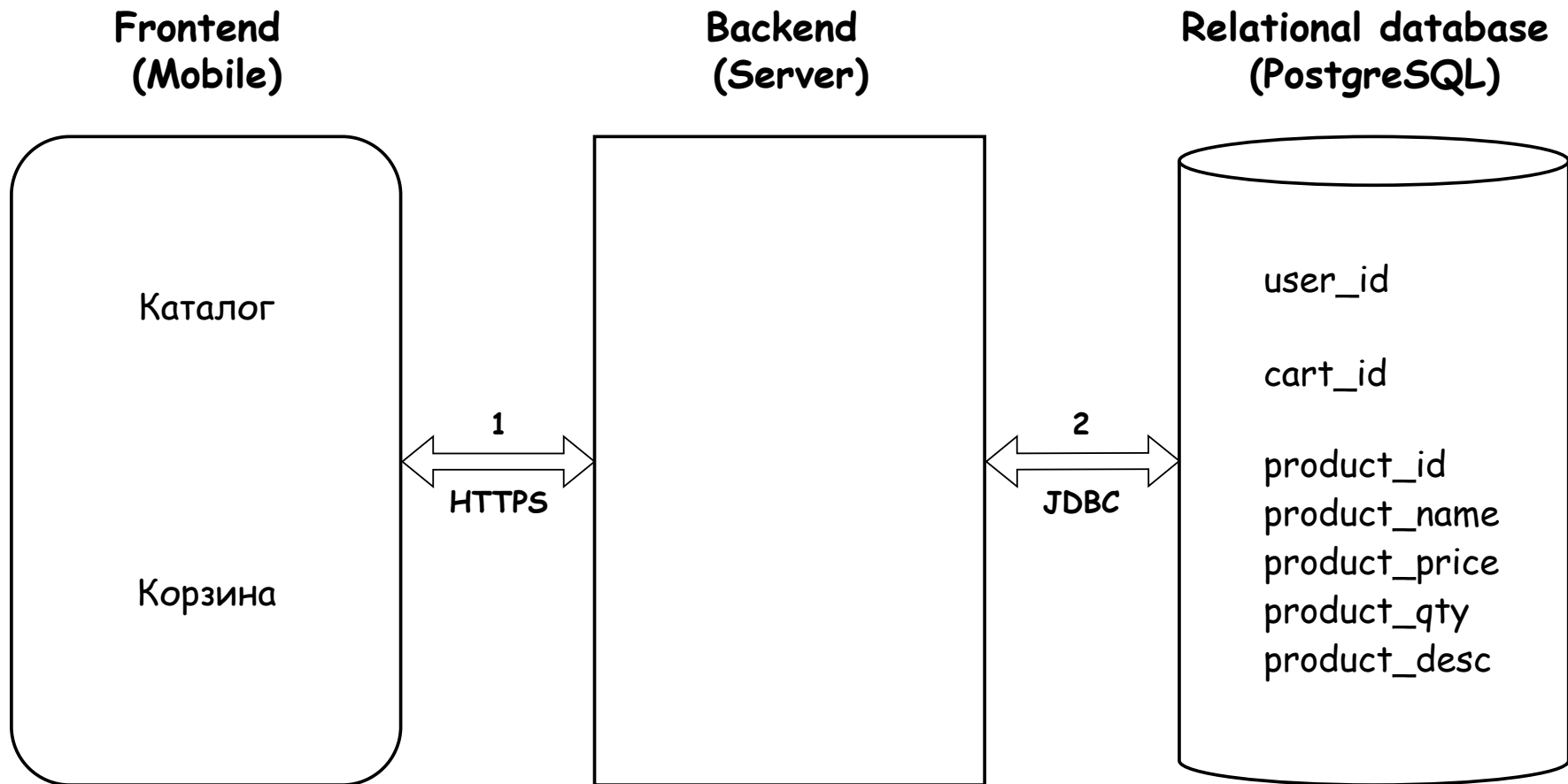
Задание 3. Use Case

| | |
|-----------------------------|--|
| Заголовок | Управление содержимым корзины |
| Акторы | Пользователь приложения [МегаМаркет] |
| Предусловие | Пользователь зарегистрирован и авторизован в приложении |
| Ограничения | Управление содержимым корзины возможно только при наличии выхода в интернет |
| Триггер | Пользователь запускает приложение [МегаМаркет] |
| Основной сценарий | <ol style="list-style-type: none"> 1. Система отображает домашнюю страницу с предлагаемыми Товарами (экран 1.а); 2. Пользователь нажимает кнопку "Купить" возле интересующего Товара; 3. Система проверяет наличие выбранного Товара и после успешной проверки отображает количество Товара, добавленного в Корзину (экран 1.б); 4. Пользователь через кнопку "Купить" добавляет другие Товары, либо с помощью кнопок "+" (увеличить) и "-" (уменьшить) возле Товара корректирует количество уже добавленного; 5. Система проверяет наличие выбранных Товаров и после успешной проверки обновляет количество Товара, находящегося в Корзине (экран 1.с); 6. Пользователь нажимает кнопку "Корзина"; 7. Система проверяет наличие выбранных Товаров и после успешной проверки отображает окно Корзины со списком находящихся в ней Товаров, где указывается количество и цена за единицу, а также итоговая сумма заказа в нижней части экрана и активная кнопка "Оформить заказ" (экран 2.а); 8. Пользователь с помощью кнопок "+" (увеличить) и "-" (уменьшить) возле Товара корректирует количество, а также через кнопку "Урна" рядом с Товаром удаляет ненужный Товар из Корзины; 9. Система проверяет наличие выбранных Товаров и после успешной проверки обновляет список и количество Товаров, находящихся в Корзине, а также удаляет Товар из Корзины, если его количество уменьшилось до 0 (экран 2.б); 10. Проверив список Товаров и итоговую сумму Пользователь нажимает кнопку "Оформить заказ"; 11. Система проверяет наличие выбранных Товаров и после успешной проверки открывает окно выбора способов доставки и оплаты (не рассматривается в контексте текущей US); <p>Критерий успеха: Пользователь добавляет и удаляет Товары в Корзине, а также корректирует их количество.</p> |
| Альтернативный сценарий | <p>10а. Пользователь с помощью кнопок "Урна" или "-" возле Товара, соответственно, удаляет каждый Товар из Корзины или уменьшает его количество до 0;</p> <p>11а. Система сообщает, что Корзина пуста и отображает кнопку "Вернуться на главную" (экран 2.с);</p> <p>12а. Пользователь нажимает на одну из кнопок "Вернуться на главную" или "Дом";</p> <p>-- Переход к шагу 1 основного сценария</p> <p>Результат: Корзина пуста, Пользователю предложено вернуться на главную страницу для наполнения Корзины.</p> |
| Исключительный сценарий № 1 | <p>3б. Система проверяет наличие выбранного Товара и, в случае отсутствия добавленного количества, не увеличивает количество Товара в Корзине больше количества в наличии и отображает под кнопкой возле такого Товара надпись красного цвета "Нет в наличии" или "В наличии X", где X - количество Товара в наличии (экран 1.д);</p> <p>-- Переход к шагу 4 основного сценария</p> <p>Результат: Пользователь не может добавить Товара в Корзину больше, чем есть в наличии, и проинформирован об этом.</p> |
| Исключительный сценарий № 2 | <p>7с. Система проверяет наличие выбранных Товаров и, в случае отсутствия одного из них, меняет цвет кнопки возле отсутствующего Товара на серый и отображает под кнопкой надпись красного цвета "Нет в наличии" или "В наличии X", где X - количество Товара в наличии, а также кнопка "Оформить заказ" становится неактивной до удаления Пользователем отсутствующего Товара из Корзины или уменьшения количества такого Товара до количества X (экран 2.д);</p> <p>8с. Пользователь удаляет из Корзины отсутствующий Товар или уменьшает его количество до количества X и, при желании, может вернуться на домашнюю страницу через кнопку "Дом", чтобы выбрать другие Товары;</p> <p>-- Переход к шагу 7 основного сценария</p> <p>Результат: Пользователь, находясь в Корзине, не может нажать кнопку "Оформить заказ" до тех пор, пока в Корзине находится хотя бы один отсутствующий Товар.</p> |

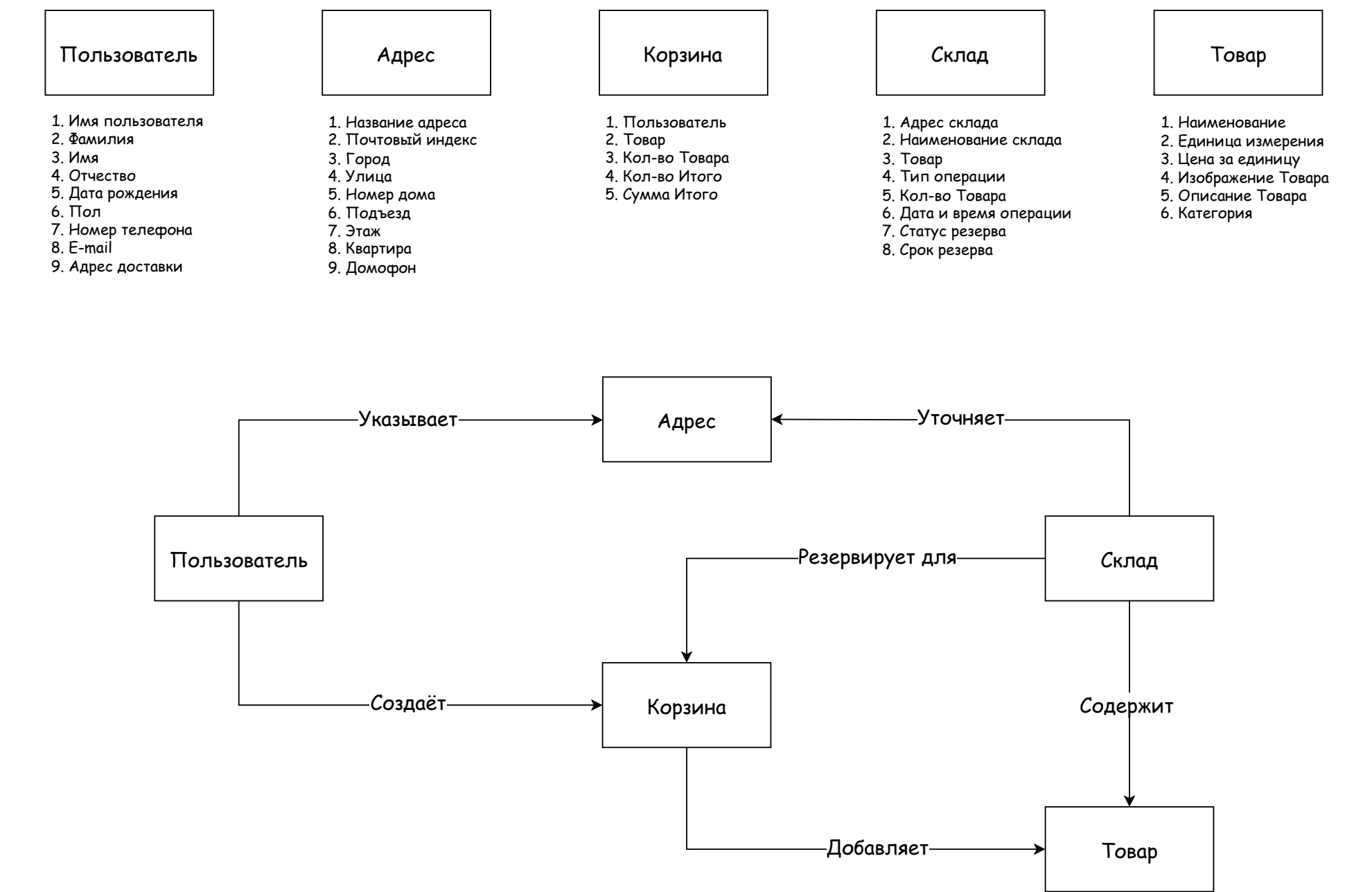
Задание 4. BPMN



Задание 5. Архитектура



Задание 6. Моделирование данных



Объект: Пользователь (родительская сущность)

| Атрибут | Описание |
|------------------|---------------------------|
| username | Имя пользователя |
| surname | Фамилия |
| name | Имя |
| patronymic | Отчество |
| birthdate | Дата рождения |
| gender | Пол |
| phone_number | Контактный номер телефона |
| email | Адрес электронной почты |
| shipping_address | Адрес (дочерняя сущность) |

Объект: Корзина (дочерняя сущность)

| Атрибут | Описание |
|------------|--------------------------------------|
| user | Пользователь (родительская сущность) |
| product | Товар (дочерняя сущность) |
| item_qty | Кол-во выбранного Товара в Корзине |
| total_qty | Общее кол-во Товаров в Корзине |
| total_cost | Общая стоимость Корзины |

Объект: Товар (дочерняя сущность)

| Атрибут | Описание |
|----------|-------------------------------|
| name | Наименование |
| unit | Единица измерения |
| price | Цена за единицу |
| photo | Изображение Товара |
| desc | Описание Товара |
| category | Группировка Товаров по смыслу |

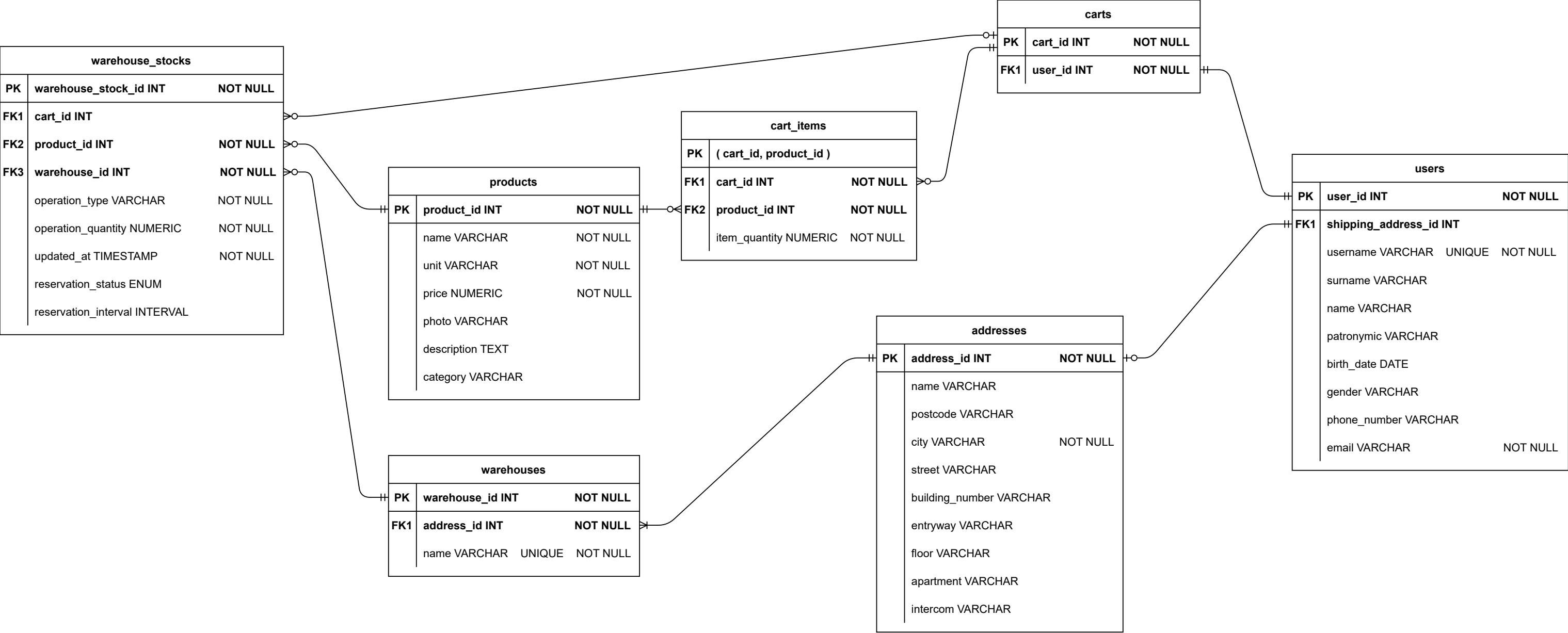
Объект: Адрес (дочерняя сущность)

| Атрибут | Описание |
|-----------------|-----------------|
| name | Название адреса |
| postcode | Почтовый индекс |
| city | Город |
| street | Улица |
| building_number | Номер дома |
| entryway | Подъезд |
| floor | Этаж |
| apartment | Квартира |
| intercom | Домофон |

Объект: Склад (родительская сущность)

| Атрибут | Описание |
|----------------------|-----------------------------|
| warehouse_address | Адрес (дочерняя сущность) |
| name | Наименование склада |
| cart | Корзина (дочерняя сущность) |
| product | Товар (дочерняя сущность) |
| operation_type | Тип операции |
| operation_qty | Количество Товара |
| updated_at | Дата и время операции |
| reservation_status | Статус резерва |
| reservation_interval | Срок резерва |

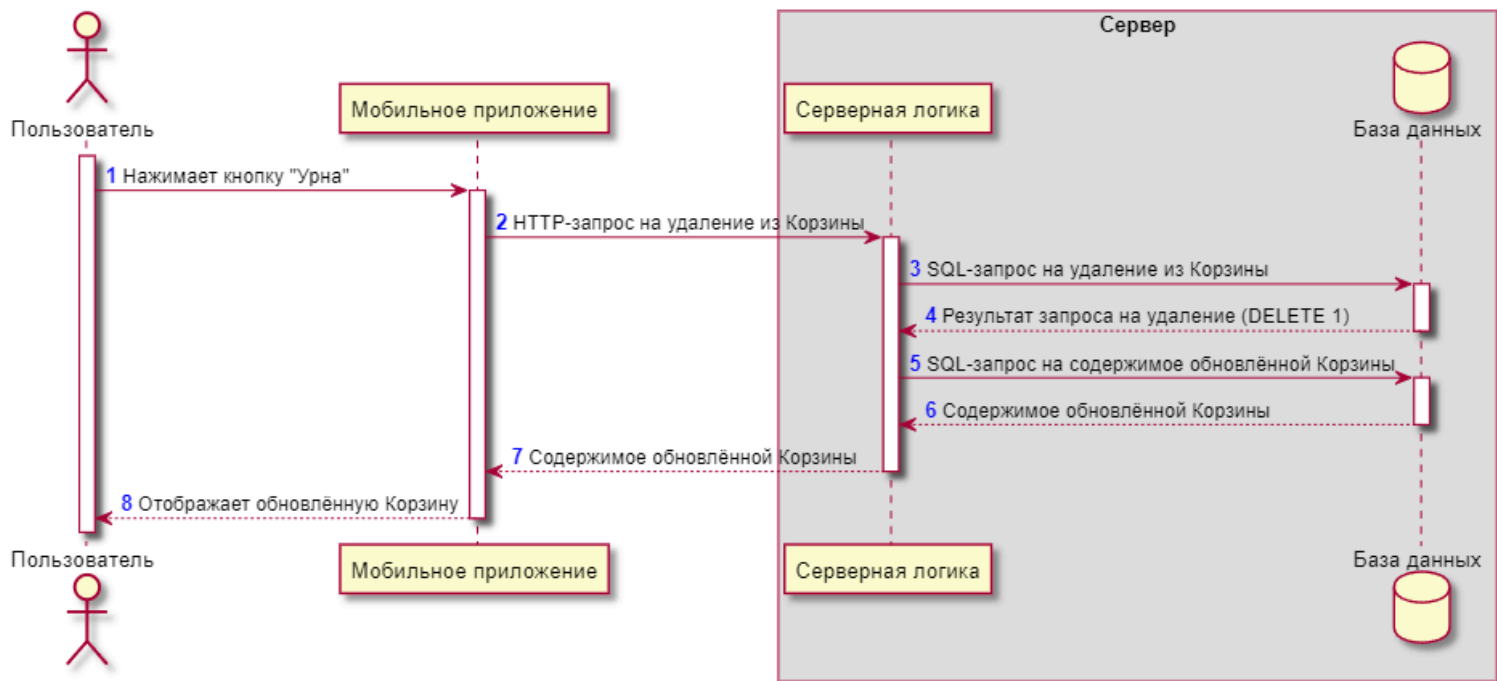
Задание 7.1. ER-диаграмма



Задание 7.2. SQL-запросы

Задание 8. Sequence-диаграмма

[МегаМаркет]: Удаление товара из корзины



| Шаг | Описание |
|-----|--|
| 1 | Пользователь, находясь на экране Корзины, решил удалить Товар из Корзины и нажимает кнопку "Урна" рядом с Товаром |
| 2 | Мобильное приложение отправляет HTTP-запрос Серверу на удаление Товара из Корзины |
| 3 | Сервер отправляет SQL-запрос в Базу данных на удаление записи из таблицы cart_items с последующей отменой резерва соответствующего Товара в таблице warehouse_stocks |
| 4 | База данных возвращает Серверу SQL-ответ с успешным удалением записи (DELETE 1) |
| 5 | Сервер отправляет SQL-запрос в Базу данных на содержимое обновлённой Корзины |
| 6 | База данных возвращает Серверу SQL-ответ с содержимым обновлённой Корзины |
| 7 | Сервер возвращает Мобильному приложению HTTP-ответ с содержимым обновлённой Корзины (HTTP 200 OK) |
| 8 | Мобильное приложение отображает Пользователю экран с содержимым обновлённой Корзины |

Задание 10. API

GET /v1/users/{userId}/cart

Получение содержимого Корзины пользователя

Request

| Название параметра | Местонахождение | Тип данных | Описание | Обязательность параметра |
|--------------------|-----------------|------------|----------------------------|--------------------------|
| userId | path | number | Уникальный ID Пользователя | Да |

Response

| Название параметра | Местонахождение | Тип данных | Описание | Обязательность параметра |
|--------------------|-----------------|------------|-----------------------------|--------------------------|
| cartId | body | number | Уникальный ID Корзины | Да |
| cartItems | body | array | Массив Товаров | Да |
| cartItem | body | object | Объект Товара | Нет |
| productId | body | number | Уникальный ID Товара | Да |
| name | body | string | Наименование Товара | Да |
| unit | body | string | Единица измерения Товара | Да |
| price | body | number | Цена за единицу Товара | Да |
| photo | body | string | Изображение Товара | Нет |
| description | body | string | Описание Товара | Нет |
| category | body | string | Категория Товара | Нет |
| itemQuantity | body | number | Количество Товара в Корзине | Да |

Response code < 200 OK >

POST /v1/users/{userId}/cart/item

Добавление выбранного Товара в Корзину пользователя

Request

| Название параметра | Местонахождение | Тип данных | Описание | Обязательность параметра |
|--------------------|-----------------|------------|--------------------------------|--------------------------|
| userId | path | number | Уникальный ID Пользователя | Да |
| productId | body | number | Уникальный ID Товара | Да |
| itemQuantity | body | number | Количество добавляемого Товара | Да |

Response

| Название параметра | Местонахождение | Тип данных | Описание | Обязательность параметра |
|--------------------|-----------------|------------|-----------------------------------|--------------------------|
| message | body | string | Сообщение о результате выполнения | Да |

Response code < 201 Created >

Задание 11. Swagger

The image shows the Swagger Editor interface. On the left, a YAML file is being edited, defining an API for 'МегаМаркет API'. The file includes metadata like version (1.1.1) and a base URL, followed by two endpoints: a GET endpoint for retrieving a user's cart and a POST endpoint for adding items to the cart. The right panel displays the rendered API documentation. It features a title 'МегаМаркет API' with version and OAS version indicators, a description, a server selection dropdown, and a section titled 'Управление Корзиной' containing the two endpoints. Below this, the 'Schemas' section defines the 'Cart' and 'Error' objects with their respective fields and types.

```
1 openapi: 3.0.3
2 info:
3   title: МегаМаркет API
4   description: API для управления содержимым Корзины Пользователя
5   version: 1.1.1
6 servers:
7   - url: https://api.megamarket.com/v1
8
9 paths:
10  /users/{userId}/cart:
11    get:
12      tags:
13        - Управление Корзиной
14      summary: Получение содержимого Корзины Пользователя
15      description: Позволяет получить список Товаров в Корзине Пользователя
16      parameters:
17        - name: userId
18          in: path
19          description: Уникальный ID Пользователя
20          example: 111
21          required: true
22          schema:
23            type: number
24      responses:
25        '200':
26          description: OK
27          content:
28            application/json:
29              schema:
30                $ref: '#/components/schemas/Cart'
31        '400':
32          description: Некорректный запрос. Проверьте параметры запроса.
33          content:
34            application/json:
35              schema:
36                $ref: '#/components/schemas/Error'
37        '404':
38          description: Пользователь не найден.
39          content:
40            application/json:
41              schema:
42                $ref: '#/components/schemas/Error'
43        '500':
44          description: Внутренняя ошибка сервера. Попробуйте позже.
45          content:
46            application/json:
47              schema:
48                $ref: '#/components/schemas/Error'
49
50  /users/{userId}/cart/item:
51    post:
52      tags:
53        - Управление Корзиной
54      summary: Добавление Товара в Корзину
55      description: Позволяет добавить выбранный Товар в Корзину Пользователя
56      parameters:
57        - name: userId
58          in: path
59          description: Уникальный ID Пользователя
60          example: 111
61          required: true
62          schema:
63            type: number
64      requestBody:
65        description: Данные Товара для добавления в Корзину
66        required: true
67        content:
68          application/json:
69            schema:
70              type: object
71              properties:
```

МегаМаркет API 1.1.1 OAS 3.0

API для управления содержимым Корзины Пользователя

Servers

https://api.megamarket.com/v1

Управление Корзиной

GET /users/{userId}/cart Получение содержимого Корзины Пользователя

POST /users/{userId}/cart/item Добавление Товара в Корзину

Schemas

Cart {
 cartId*
 number example: 123
 Уникальный ID Корзины
 cartItems*
 [{
 productId*
 name*
 unit*
 price*
 photo
 description
 category
 itemQuantity*
 }]
}

Error {
 code*
 number example: 418
 Код ошибки
 message*
 string example: Я - чайник
 Сообщение об ошибке
}

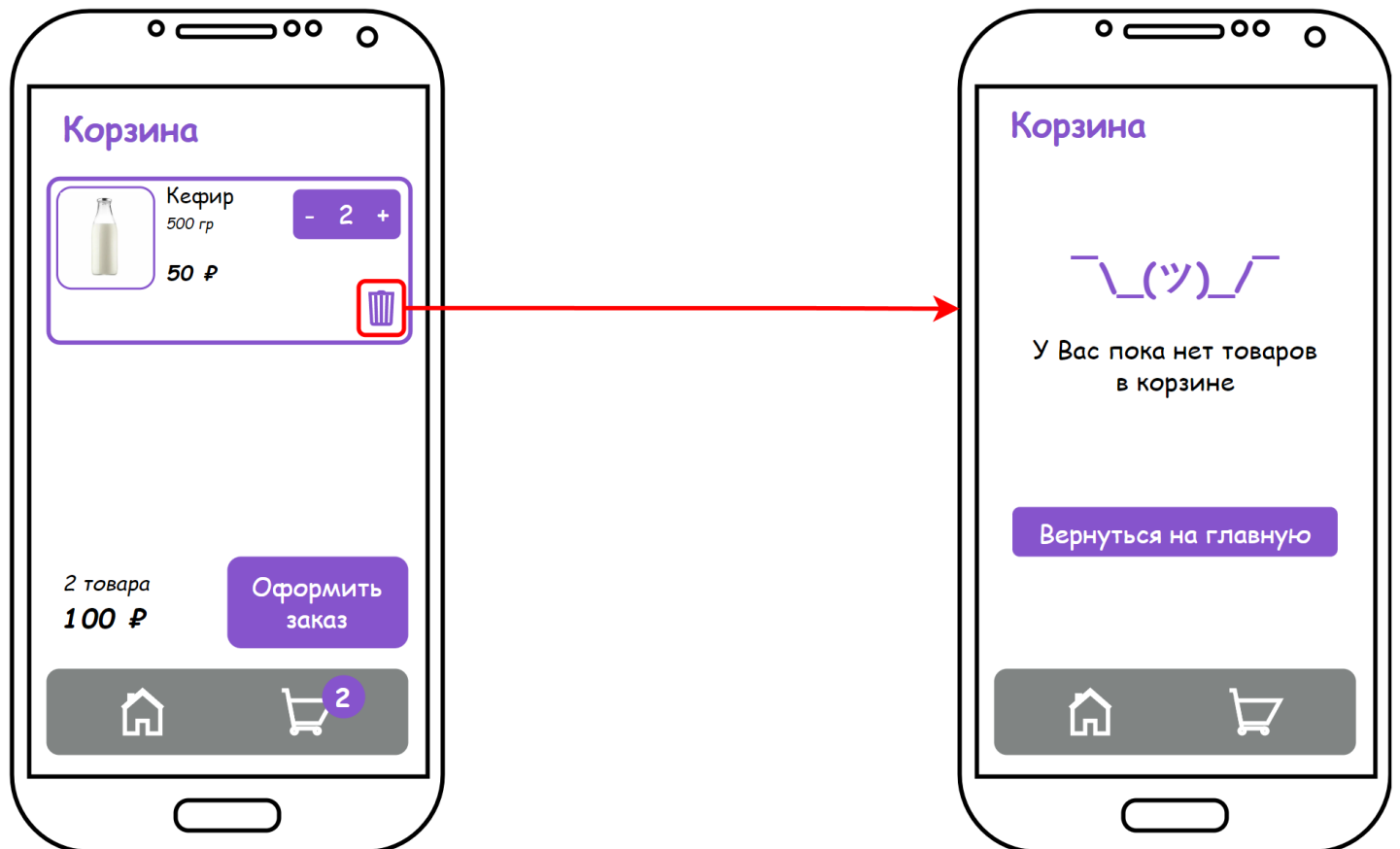
YAML-файл

Задание 12.1. Критерии приёмки

Функциональность: Управление содержимым Корзины Пользователя

Номер кейса: 1

Дано: Пользователь авторизован в приложении и находится на странице Корзины



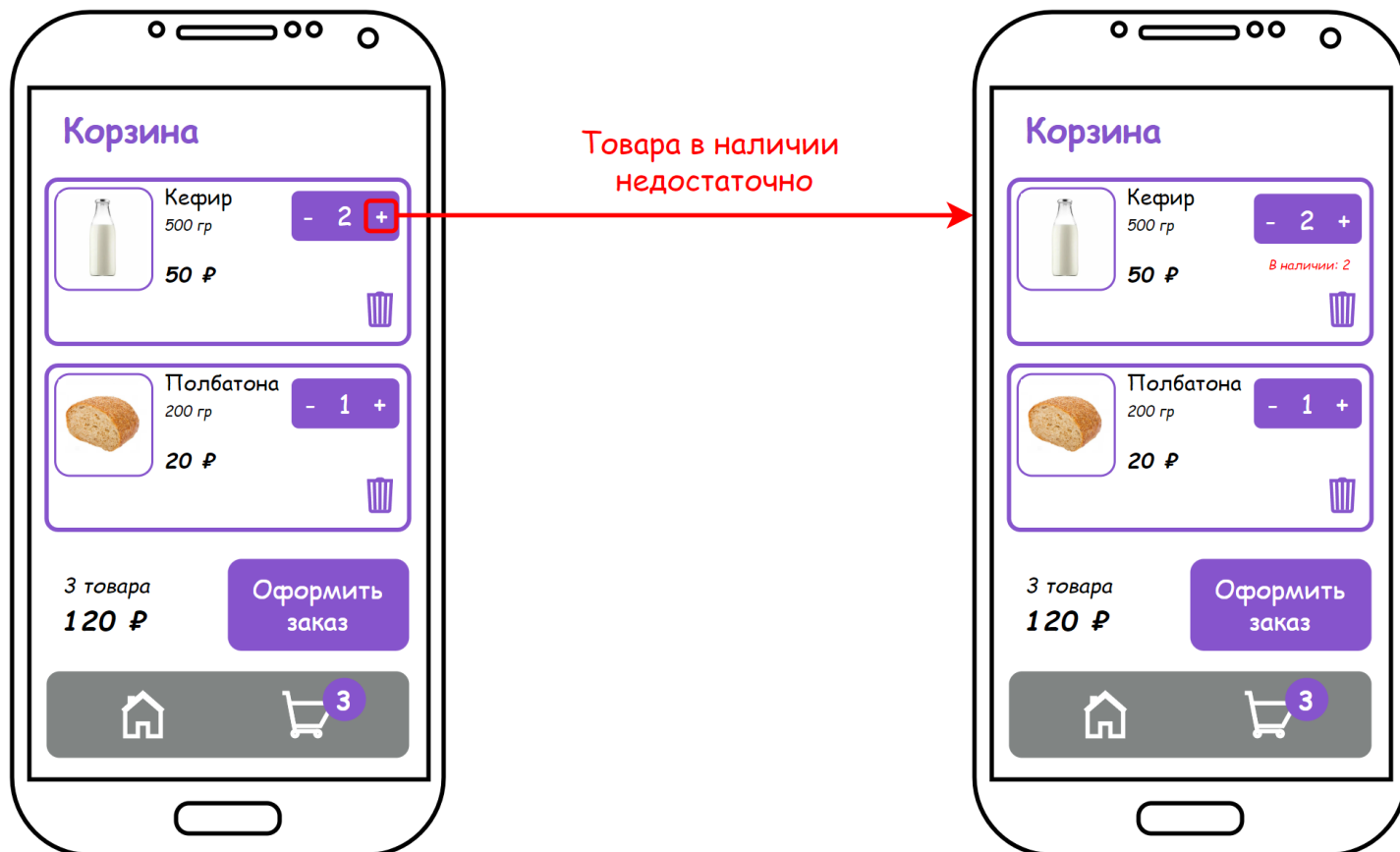
Когда: Пользователь нажимает кнопку "Урна" или "-" возле Товара, соответственно, удаляя каждый Товар из Корзины или уменьшая его количество до 0

Тогда: Система сообщает, что Корзина пуста, и отображает кнопку "Вернуться на главную"

Функциональность: Управление содержимым Корзины Пользователя

Номер кейса: 2

Дано: Пользователь авторизован в приложении и находится на странице Корзины



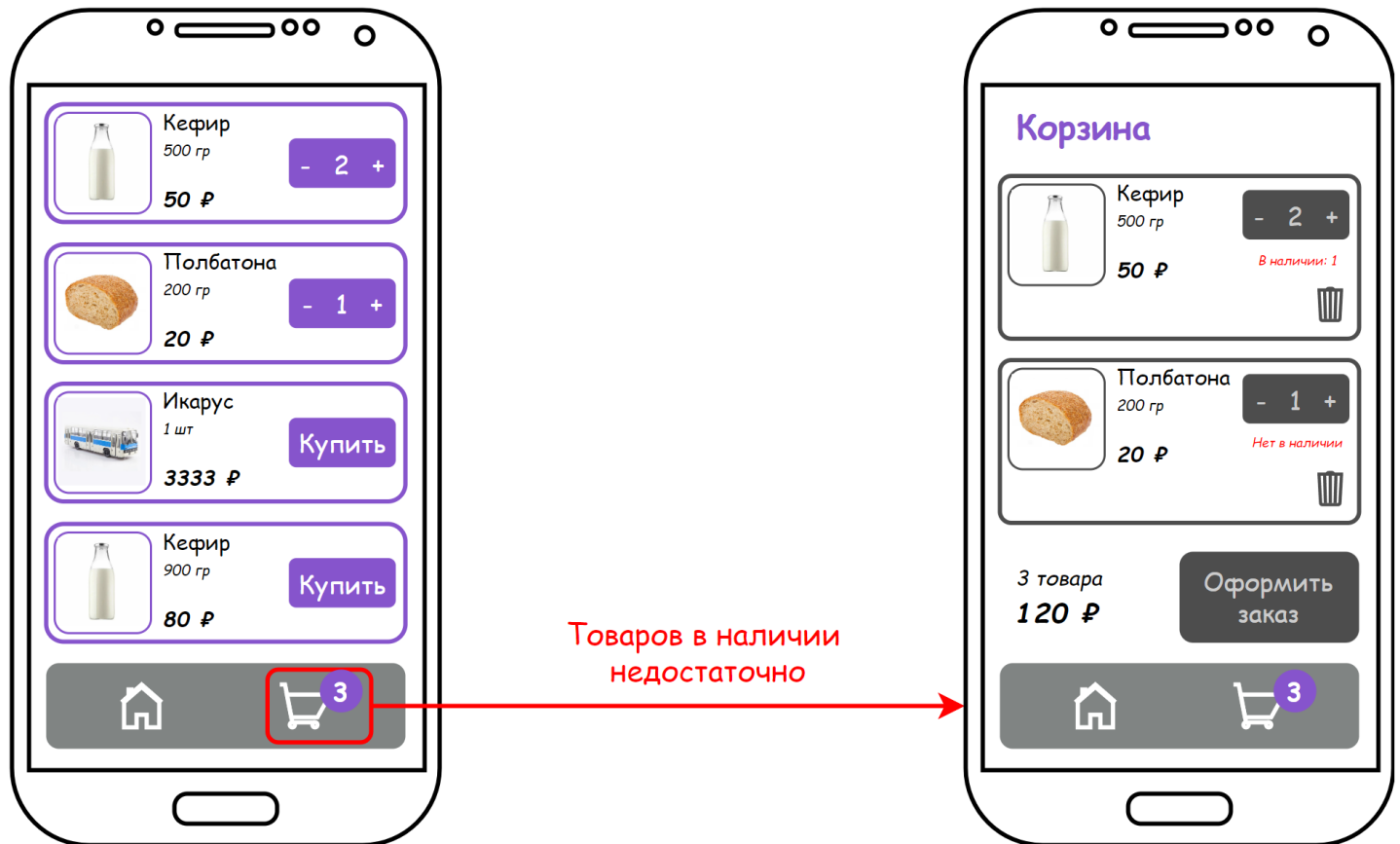
Когда: Пользователь нажимает кнопку "+" (увеличить) возле Товара, увеличивая количество выбранного Товара в Корзине

Тогда: Система проверяет наличие выбранного Товара и, в случае отсутствия добавленного количества, не увеличивает количество Товара в Корзине больше количества в наличии и отображает под кнопкой возле такого Товара надпись красного цвета "Нет в наличии" или "В наличии X", где X - количество Товара в наличии

Функциональность: Управление содержимым Корзины Пользователя

Номер кейса: 3

Дано: Пользователь авторизован в приложении и находится на странице Каталога



Когда: Пользователь нажимает кнопку "Корзина"

Тогда: Система проверяет наличие выбранных Товаров и, в случае отсутствия одного из них, меняет цвет кнопки возле отсутствующего Товара на серый и отображает под кнопкой надпись красного цвета "Нет в наличии" или "В наличии X", где X - количество Товара в наличии, а также кнопка "Оформить заказ" становится неактивной до удаления Пользователем отсутствующего Товара из Корзины или уменьшения количества такого Товара до количества X

Задание 12.2. Нефункциональные требования

Производительность

1. Страница Корзины должна открываться не более 2 секунд;
2. Запрос получения содержимого Корзины Пользователя `GET /users/{userId}/cart` должен выдерживать нагрузку 1 rps.

Надёжность

1. Система должна быть доступна 99% времени.

Безопасность

1. Личные данные Пользователей должны быть зашифрованы;
2. Пользовательские пароли хранятся в зашифрованном виде.

Удобство использования

1. Пользователь должен зарегистрироваться в системе не более, чем за 2 минуты без использования инструкции.

Масштабируемость

1. Система может поддерживать рост базы данных на 10% в год.

Спасибо за внимание!

