

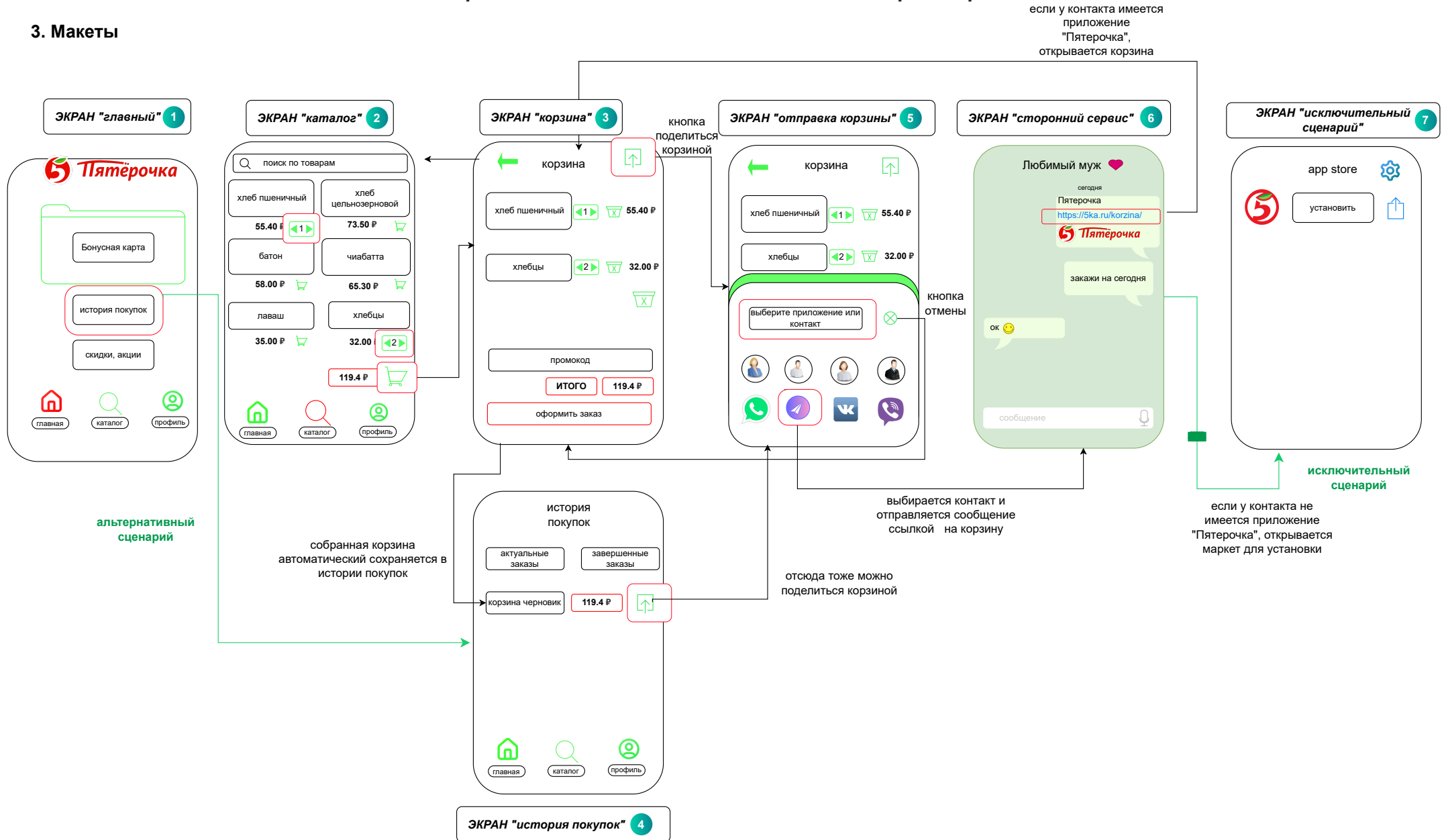
# Бизнес-требования

## 1. Название продукта - приложение "Пятерочка"

## 2. User story:

*Я как пользователь приложения доставки магазина "Пятерочка", хочу иметь возможность отправить собранную корзину другому пользователю приложения, чтобы поделиться со списком товаров в корзине.*

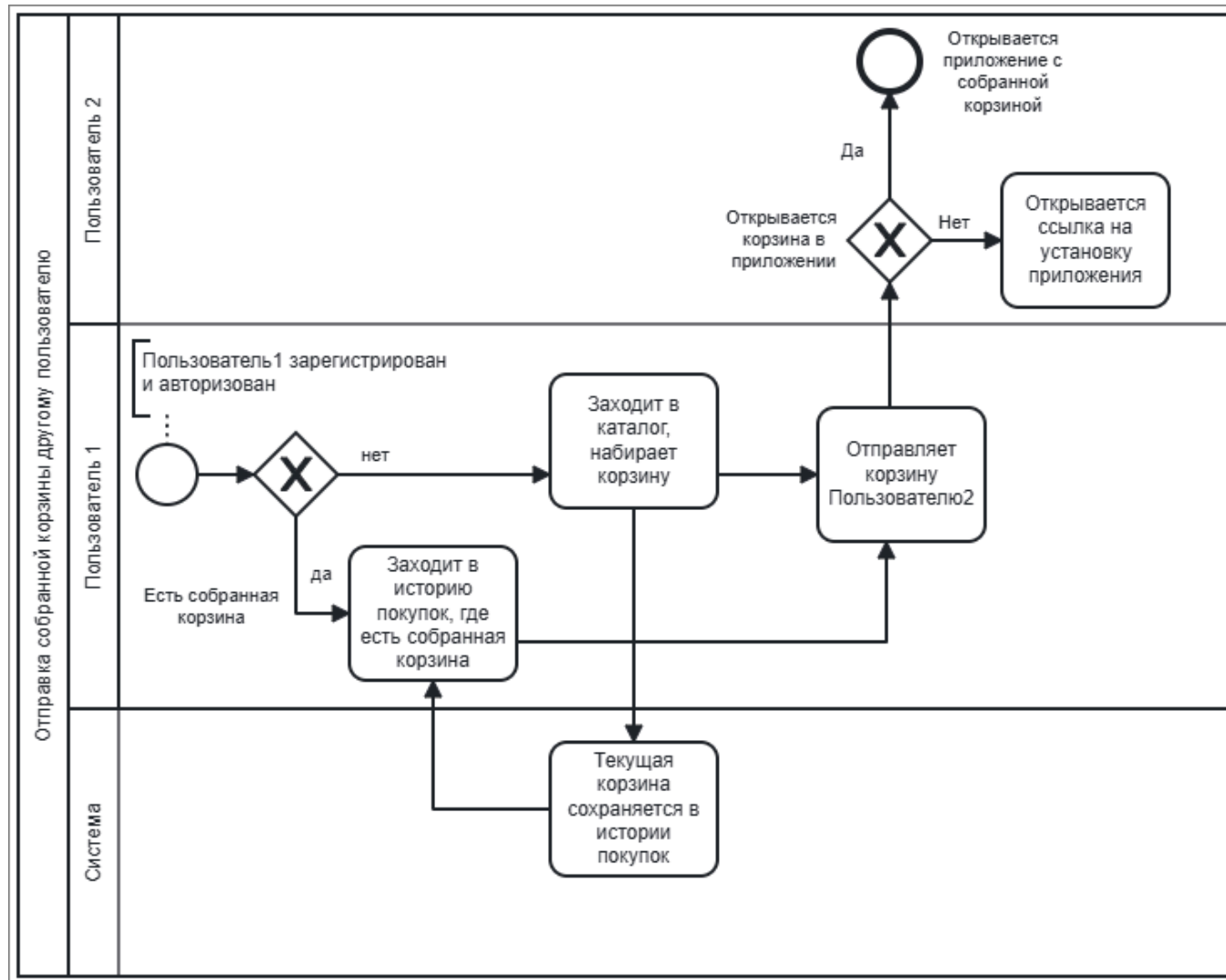
## 3. Макеты



#### 4. Use case

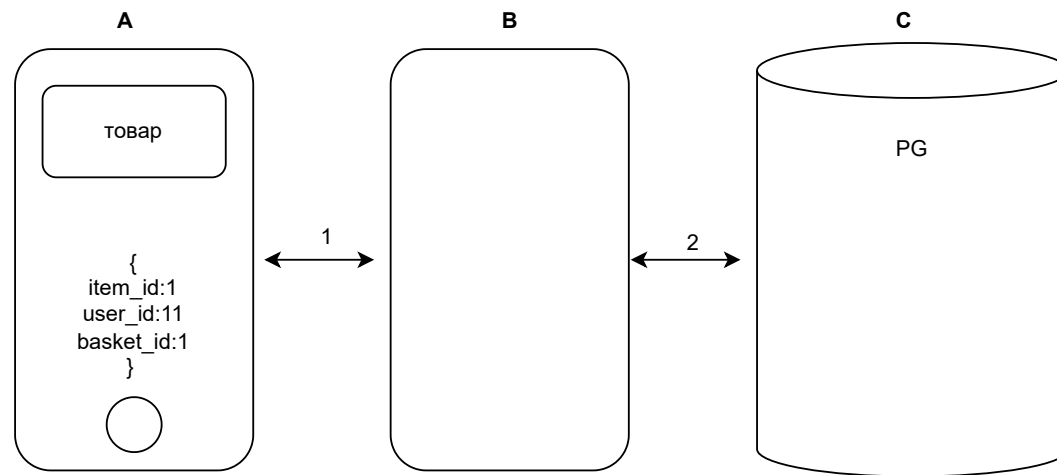
<b>Заголовок</b>	Отправка собранной корзины другому пользователю приложения
<b>Акторы</b>	Пользователь1 Пользователь2
<b>Предусловие 1</b>	Пользователь1 зарегистрирован и авторизован в приложении Пользователь2 зарегистрирован и авторизован в приложении
<b>Предусловие 2</b>	Пользователь1 зарегистрирован и авторизован в приложении
<b>Ограничения</b>	Наличие хотя бы одного товара в корзине Пользователя1
<b>Триггер</b>	Пользователь1 добавил в корзину хотя бы один товар
<b>Основной сценарий</b>	<p>Основной сценарий выполняется по предусловию 1</p> <ol style="list-style-type: none"><li>1. Система отображает каталог товаров (экран 2)</li><li>2. Пользователь1 выбирает товары и количество товаров</li><li>3. Система отображает экран с собранными товарами в корзине (экран 3)</li><li>4. Пользователь1 нажимает на кнопку "поделиться корзиной"</li><li>5. Система предлагает выбрать контакт и способы отправки корзины (экран 5)</li><li>6. Пользователь1 выбирает через какое приложение отправить корзину и с кем поделиться (пользователь2) и отправляет корзину сообщением в виде ссылки на корзину в приложении</li><li>7. В зависимости от выбранного приложения, открывается сторонний сервис, где создается сообщение с отправленной ссылкой для пользователя2 (экран 6)</li><li>8. У Пользователя2 успешно открылась корзина (экран3)</li></ol>
<b>Альтернативный сценарий</b>	<p>Альтернативный сценарий выполняется по предусловию 1</p> <p>4а. Пользователь1 заходит в историю покупок через главный экран и находит автосохраненную корзину (экран 1 - экран 4) -- Переход к шагу 4 основного сценария</p>
<b>Исключительный сценарий</b>	<p>Исключительный сценарий выполняется по предусловию 2</p> <p>8b. У Пользователя2 не открывается корзина по ссылке полученной от Пользователя1, так как у него не установлено данное приложение. 9b. В таком случае у Пользователя2 открывается по ссылке play market/app store для установки данного приложения (экран7)</p> <p><b>Результат:</b> Пользователь2 не смог успешно открыть список товаров полученной от Пользователя1, установка приложения поможет устранить данную проблему</p>

## 5. BPMN



# Функциональные требования

## 1. Архитектура



**A** - мобильное приложение

**B** - сервер

**C** - реляционная БД - PG

**1** - протокол взаимодействия http/https

**2** - стандарт соединения между приложением и с СУБД - JDBC

2.Модель данных

Объект Пользователь1

Родительская сущность	Атрибут	Описание
user	name	Имя пользователя1
	last_name	Фамилия пользователя1
	email	электронная почта
	phone	номер телефона

Объект Товар

Родительская сущность	Атрибут	Описание
<i>product</i>	name_product	наименование товара
	description	описание товара
	price	цена товара

Объект История покупок

Дочерняя сущность	Атрибут	Описание
<i>shopping_history</i>	baskets	текущая сохраненная корзина
	orders	завершенные заказы

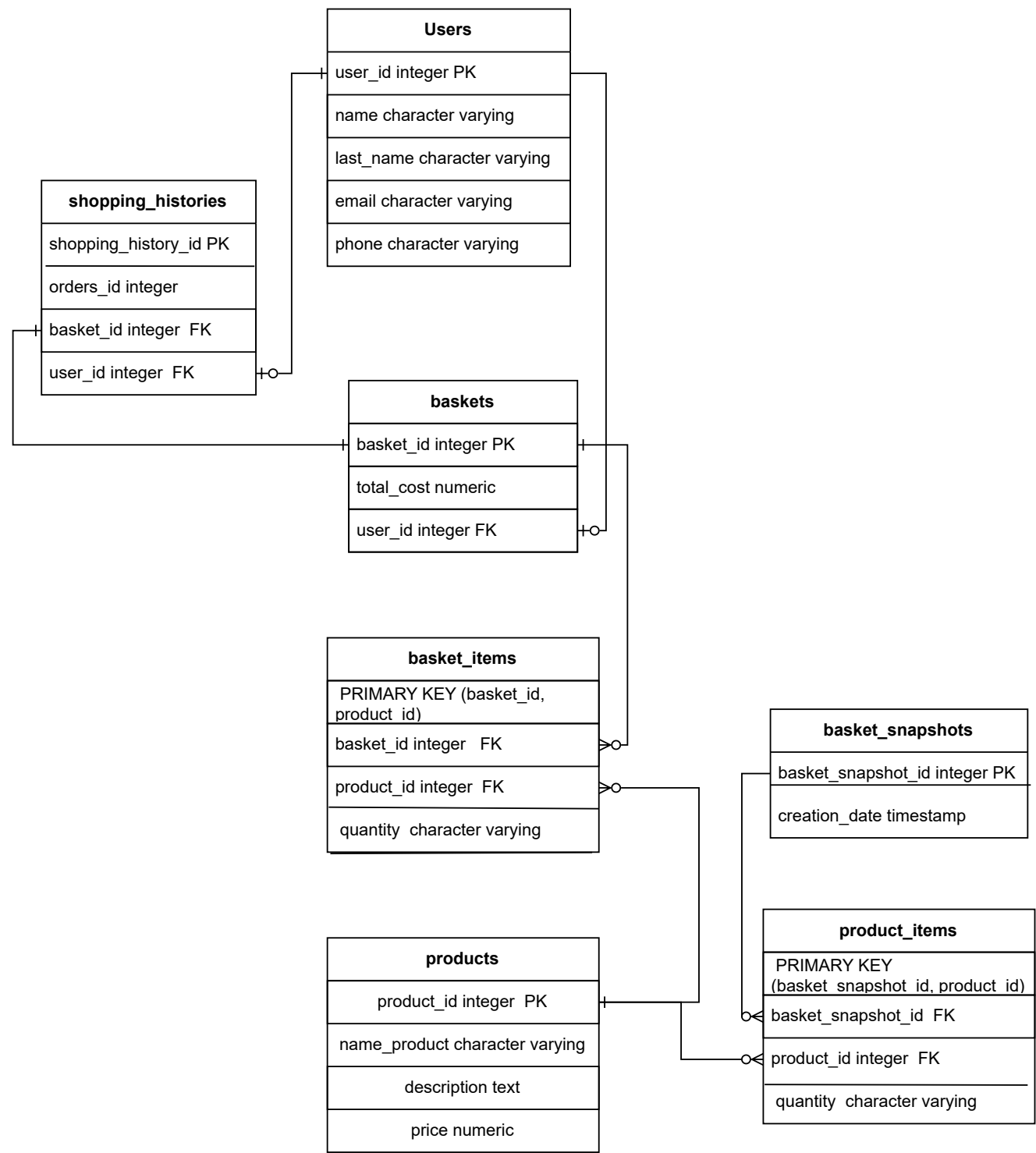
Объект Сообщение

Дочерняя сущность	Атрибут	Описание
<i>message</i>	sender	отправитель ссылки с корзиной
	recipient	получатель ссылки с корзиной
	text	ссылка на корзину (текст ссылки на корзину)

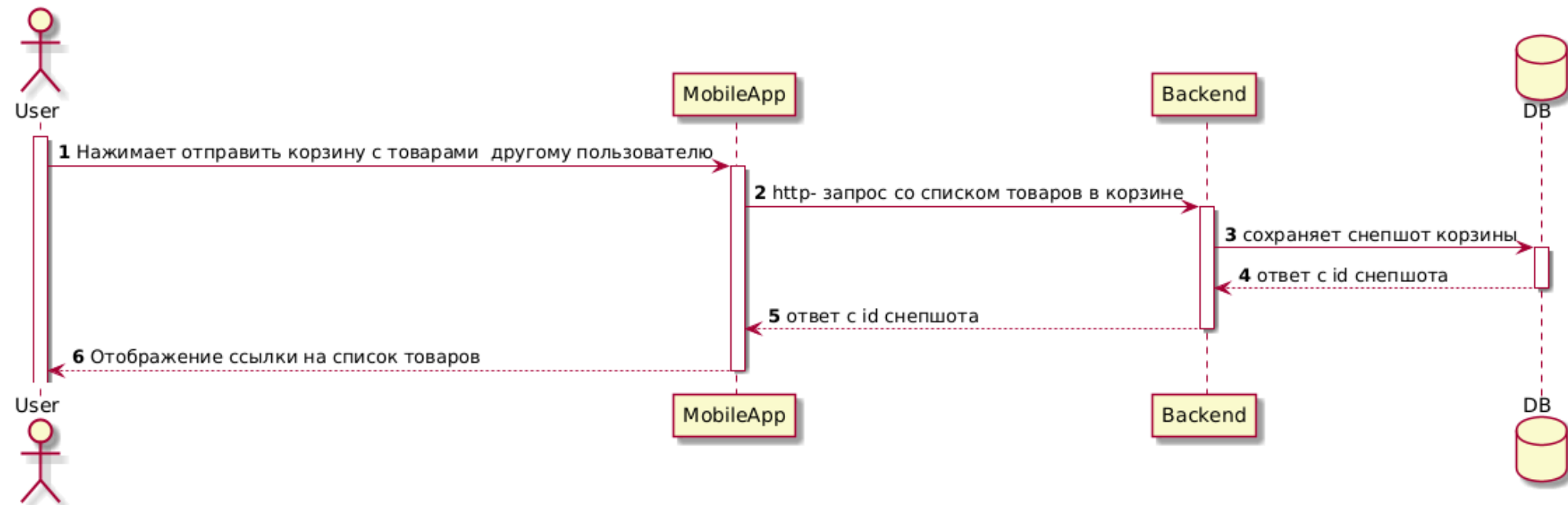
Объект Корзина

Дочерняя сущность	Атрибут	Описание
<i>basket</i>	quantity_totalproduct	общее количество товаров
	total_cost	общая стоимость
	link	ссылка на корзину
	products	ссылка на объект "товар"
	quantity_product	количество товара

3. ERD - диаграмма



4. Диаграмма последовательности



№	Описание
1	Пользователь нажимает отправить корзину с товарами другому пользователю
2	После нажатия кнопки, приложение отправляет запрос с содержанием корзины (со списком товаров)
3	Backend сохраняет текущую корзину как снимшот в базе данных
4	Ответ с id снимшота корзины
5	Backend отправляет в приложение ответ с id снимшота корзины
6	Приложение отображает экран с ссылкой на список товаров

5. REST. Табличный вид

POST/api/basketsnapshot

Создание снелшота корзины, чтобы поделиться им с другими пользователями

Request:

Название параметра		Тип параметра	Тип данных	Описание	Обязательность параметра
basketItems		body	array	массив с товарами корзине	ДА
	product	body	object	объект содержащий информацию о товарах	ДА
	productId	body	integer	id товара	ДА
	name	body	string	название товара	ДА
	price	body	number	цена товара	ДА
	quantity	body	integer	количество товара	ДА

Code:200

Сервер возвращает ответ об успешном создании снелшота

Response

Название параметра	Тип параметра	Тип данных	Описание	Обязательность параметра
message	body	string	сообщение об успешном выполнении операции	ДА
basketSnapshotId	body	integer	id снелшота	ДА



**GET/api/basketsnapshot/{basketSnapshotId}**

Запрос на получение id снелшота

**Request:**

Название параметра	Тип параметра	Тип данных	Описание	Обязательность параметра
basketSnapshotId	path	integer	id снелшота	ДА

**Response**

**Code:200**

Название параметра		Тип параметра	Тип данных	Описание	Обязательность параметра
creationDate		body	date	Дата и время создания снелшота	ДА
basketItems		body	array	массив с товарами корзине	ДА
	product	body	object	объект содержащий информацию о товарах	ДА
	productId	body	integer	id товара	ДА
	name	body	string	название товара	ДА
	price	body	number	цена товара	ДА
	quantity	body	integer	количество товара	ДА

# Swagger - OpenAPI 3.0

1.0.0OAS 3.0

Создание и получение снэпшота корзины

Servers

https://5ka.ru.swagger.io/api/v1

snapshotbasket

методы для работы с корзинами товаров, создание снимка корзины

POST

/basketsnapshot/

создание снимка корзины

создание снимка корзины, содержащего список товаров

Parameters

Try it out

No parameters

Request body

application/json

параметры метода для создания снимка корзины

Example Value

Schema

```
{  "basketItems": [    {      "productId": 1,      "name": "Milk",      "price": 79.99,      "quantity": 2    }  ]}
```

Активация Windows  
Чтобы активировать Windows, перейдите в раздел "Параметры".

## Responses

Code	Description	Links
200	успешное создание снелшота корзины	No links

Media type

application/json

Controls Accept header.

Example Value | Schema

```
{
  "code": 0,
  "message": "id снелшота создан",
  "basketSnapshotId": 456
}
```

GET

/basketsnapshot/{basketSnapshotId} получить снимок корзины по идентификатору



получить снимок корзины по идентификатору

## Parameters

Try it out

Name	Description
<b>basketSnapshotId</b> * required integer (path)	id снелшота

11

Insert text here

## Responses

Code	Description	Links
200	Success	No links

Media type

application/json

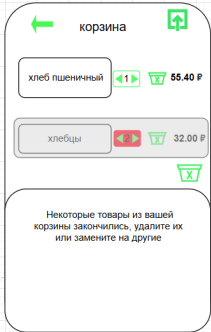
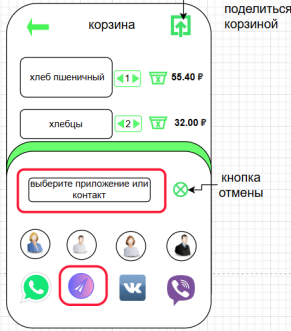
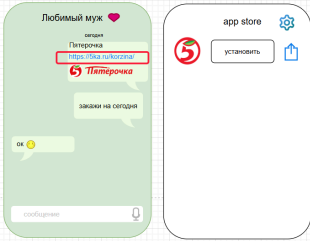
Controls Accept header.

Example Value | Schema

```
{
  "creationDate": "2024-12-20T09:12Z",
  "basketItems": [
    {
      "productId": 12,
      "name": "Milk",
      "price": 79.99,
      "quantity": 2
    }
  ]
}
```

# Критерии приемки и нефункциональные требования

## 1.Критерии приемки

User story	Я как пользователь приложения доставки магазина "Пятерочка", хочу иметь возможность отправить собранную корзину другому пользователю приложения, чтобы поделиться списком товаров в корзине.	
Основной сценарий	<div>1. Система отображает каталог товаров (экран 2)</div> <div>2. Пользователь1 выбирает товары и количество товаров</div> <div>3. Система отображает экран с собранными товарами в корзине (экран 3)</div> <div>4. Пользователь1 нажимает на кнопку "поделиться корзиной"</div> <div>5. Система предлагает выбрать контакт и способы отправки корзины (экран 5)</div> <div>6. Пользователь1 выбирает через какое приложение отправить корзину и с кем поделиться (пользователь2) и отправляет корзину сообщением в виде ссылки на корзину в приложении</div> <div>7. В зависимости от выбранного приложения, открывается сторонний сервис, где создается сообщение с отправленной ссылкой для пользователя2 (экран 6)</div> <div>8. У Пользователя2 успешно открылась корзина (экран3)</div> <div>Критерий успеха: Пользователь2 получил список товаров в приложении от пользователя1</div>	
Кейс1	<div>Функциональность Отправка собранной корзины другому пользователю приложения</div> <div>Дано Пользователь находится на экране с сохраненной корзиной</div> <div>Когда Пользователь отправляет корзину со списком товаров,где есть товары, которые закончились</div> <div>Тогда Система выводит сообщение: Некоторые товары из вашей корзины закончились, удалите их или замените на другие</div>	
Кейс2	<div>Функциональность Отправка собранной корзины другому пользователю приложения</div> <div>Дано Пользователь находится на экране с сохраненной корзиной</div> <div>Когда Пользователь после того как нажал на кнопку поделиться корзиной, в появившемся окне нажимает на кнопку отменить отправку</div> <div>Тогда Корзина не отправляется, чтобы поделиться корзиной нужно повторно нажать кнопку поделиться корзиной</div>	
Кейс3	<div>Функциональность Отправка собранной корзины другому пользователю приложения</div> <div>Дано У Пользователя2 приложение "Пятерочка" не установлено</div> <div>Когда Пользователь2 открывает полученную ссылку</div> <div>Тогда По ссылке открывается маркет для установки приложения "Пятерочка"</div>	

## 2. Нефункциональные требования

### Требование надежности:

1. Система должна быть доступна 99% времени.

### Требование производительности:

1. Время отклика на запросы пользователей должно быть минимальным, чтобы обеспечить плавность взаимодействия с приложением.

### Требование совместимости:

1. Совместимость устройств : Мобильная версия приложения должна быть оптимизирована для работы на различных мобильных устройствах (iOS, Android).

### Требования безопасности:

1. Вся информация, передаваемая между клиентом и сервером, должна шифроваться для защиты содержимого сообщений.
2. Только авторизованные пользователи должны иметь доступ к функциям обмена списками товаров в корзине.

### Требования удобства использования:

1. Интерфейс приложения должен быть интуитивно понятным и легко осваиваемым пользователями.
2. Отправку корзины можно осуществить любым доступным приложением с необходимым функционалом, которое установлено у пользователя (приложения с мессенжерами типа WhatsApp, Telegram).

### Требование масштабируемости:

1. Архитектура приложения должна позволять масштабирование для поддержки растущего числа пользователей и увеличенной нагрузки.