

Техническое задание на разработку мобильного приложения Lamoda

Выполнила: Мухаметьянова Наргиза Азаматовна

Содержание

Техническое задание на разработку	1
<i>Бизнес-требования</i>	<i>1</i>
1. Название продукта.....	1
2. User Story:	1
3. Макеты	1
4. Use Case.....	2
5. BPMN	3
<i>Функциональные требования</i>	<i>4</i>
1. Архитектура	4
2. Модель данных	4
3. ER-диграмма.....	7
4. Sequence diagram	8
5. REST – табличный вид.....	9
6. Swagger.....	10
<i>Критерии приемки и Нефункциональные требования.....</i>	<i>13</i>
1. Критерии приемки	13
2. Нефункциональные требования	15

Техническое задание на разработку

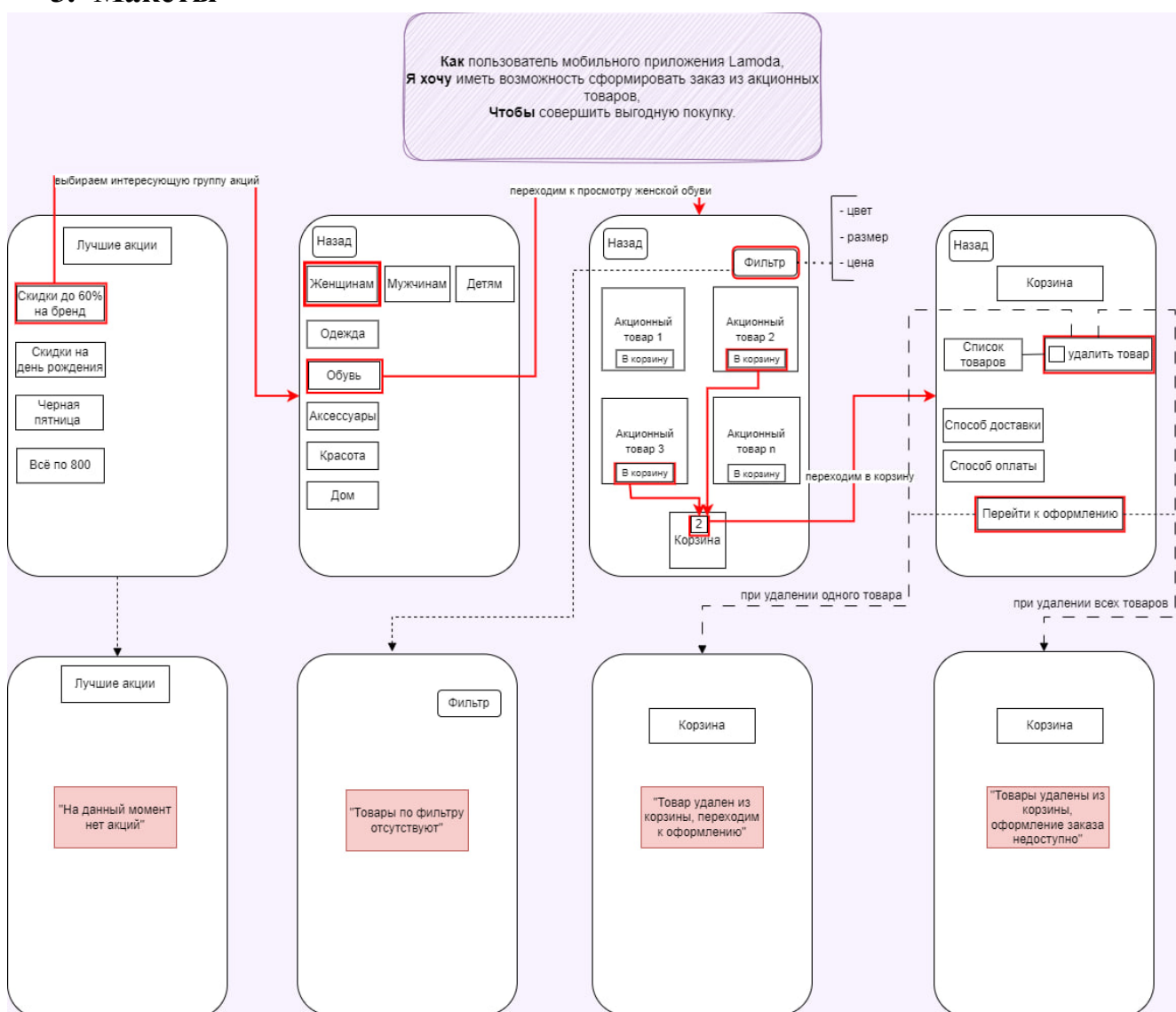
Бизнес-требования

1. Название продукта – приложение интернет-магазина Lamoda

2. User Story:

Как пользователь мобильного приложения Lamoda,
Я хочу иметь возможность сформировать заказ из акционных товаров,
Чтобы совершить выгодную покупку.

3. Макеты



4. Use Case

Заголовок	Формирование заказа из акционных товаров
Акторы	Покупатель
Предусловие	Покупатель зарегистрирован и авторизован в приложении Lamoda и находится на странице "Лучших акций"
Ограничения	Акционные товары разделены на категории "Женщины", "Мужчины", "Дети", подкатегории "Одежда", "Обувь", "Аксессуары", "Красота", "Дом". Возможно выбрать только один способ оплаты и способ доставки. Фильтр товаров осуществляется по: цвету, размеру, цене.
Триггер	Покупатель заходит в раздел "Лучшие акции"
Основной сценарий	<ol style="list-style-type: none"> 1. Система отображает форму лучших акций (экран 1) 2. Покупатель выбирает интересующую группу акций 3. Система отображает категории и подкатегории акционных товаров (экран 2) 4. Покупатель выбирает нужные категории и подкатегории 5. Система отображает список акционных товаров с возможностью фильтровать товары (экран 3) 6. Покупатель пользуется фильтром 7. Система отобразила товары согласно заполненному фильтру покупателя 8. Покупатель добавляет интересующий товар в корзину 9. Система отображает счетчик корзины 10. Покупатель переходит в корзину 11. Система отображает список товаров в корзине (экран 4) 12. Система отображает покупателю возможность удалить товар 13. Система отображает покупателю возможность выбрать способ доставки 14. Система отображает покупателю возможность выбрать способ оплаты 15. Покупатель выбирает подходящие условия заказа и переходит к оформлению заказа 16. Система сохранила заказ и одобрила переход к оформлению заказа <p>Критерий успеха: Покупатель оформил заказ, в системе создан заказ покупателя</p>
Альтернативный сценарий 1	<p>1b. Система сообщает, что акций нет (экран 5)</p> <p>Результат: Просмотр акционных товаров не состоялся</p>
Альтернативный сценарий 2	<p>7b. Товары по запрошенному фильтру не найдены (экран 6)</p> <p>Результат: Просмотр отфильтрованных акционных товаров не состоялся</p>
Альтернативный сценарий 3	<p>12b. Покупатель удалил один из товаров корзины и продолжает оформление заказа</p> <p>Результат: Система подтверждает переход к оформлению заказа</p>
Исключительный сценарий	<p>12c. Покупатель удалил все товары из корзины (экран 7)</p> <p>Результат: Система заблокировала возможность оформить заказ</p>

5. BPMN

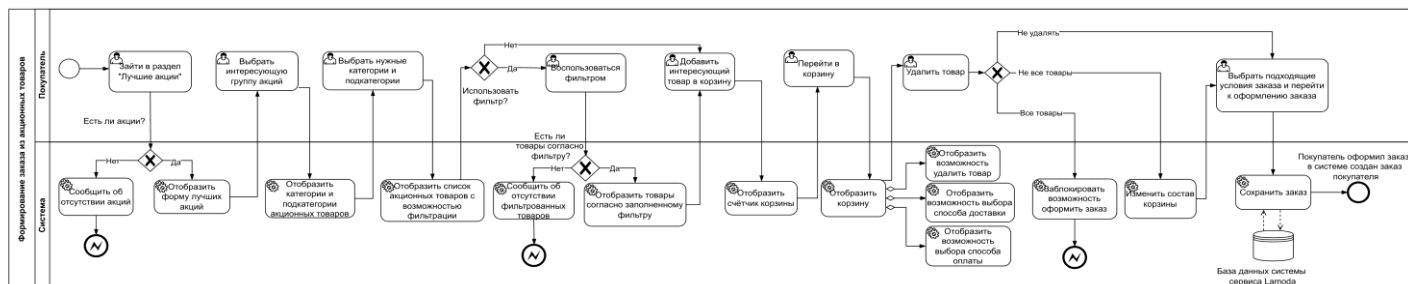


Рисунок 1 – полная модель BPMN

Более наглядная демонстрация модели BPMN, разделенная на 2 части:

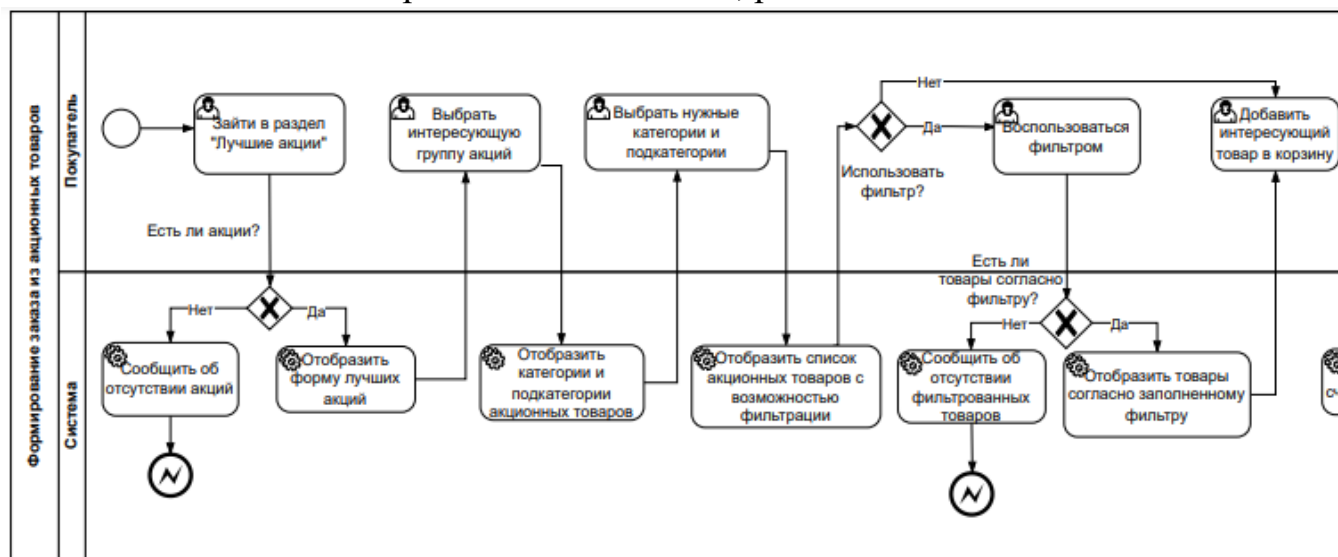


Рисунок 2 – Первая часть модели BPMN

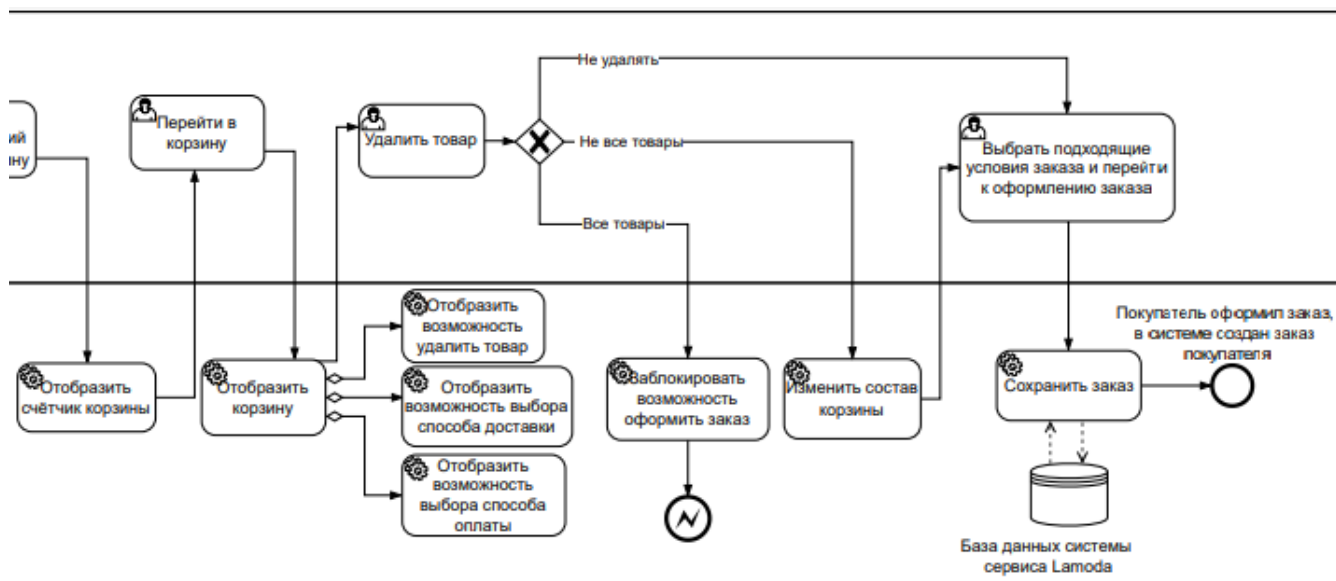


Рисунок 3 – Вторая часть модели BPMN

Функциональные требования

1. Архитектура

Формирование заказа из акционных товаров



A - frontend, мобильное приложение
B - бэкэнд/сервер
C - реляционная база данных PostgreSQL

1 - протокол взаимодействия https
2 - PostgreSQL протокол взаимодействия TCP/IP

2. Модель данных



Объект: «Покупатель» (родительская сущность)

Родительская сущность	Атрибут	Описание
buyers	-	Объект покупателя, который имеет атрибуты
	surname	Фамилия
	name	Имя
	telephone_num	Номер телефона
	email	Электронная почта
	personal_sale	Персональная скидка

Объект: «Товар» (родительская сущность)

Родительская сущность	Атрибут	Описание
products	-	Объект товаров, который имеет атрибуты и ссылки на другие объекты
	categories	Ссылка на объект «Категории» (родительская сущность)
	product_name	Название товара
	product_photo	Фото товара
	product_cost	Цена товара
	product_color	Цвет товара
	product_size	Размер товара

Объект: «Лучшие акции» (родительская сущность)

Родительская сущность	Атрибут	Описание
best_actions	-	Объект лучших акций, который имеет атрибуты и ссылки на другие объекты
	action_name	Название акции
	action_description	Описание акции
	action_start	Начало акции
	action_end	Конец акции

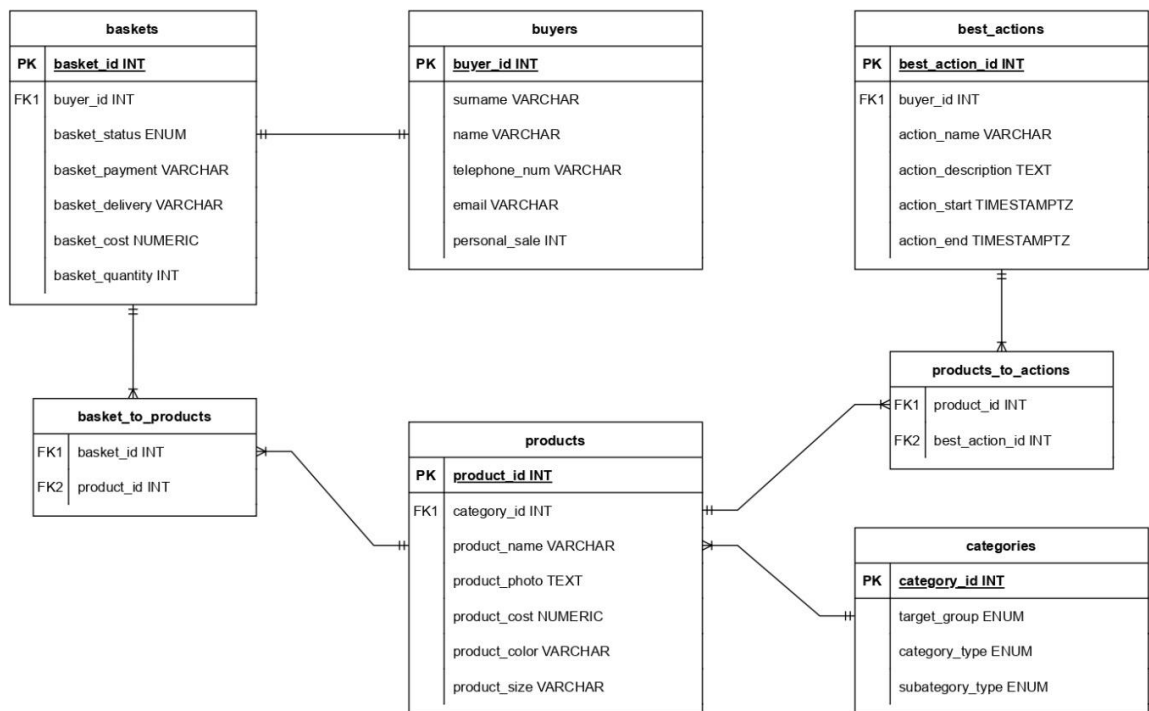
Объект: «Корзина»

Родительская сущность	Атрибут	Описание
baskets	-	Объект корзины, который имеет атрибуты и ссылки на другие объекты
	buyers	Ссылка на объект покупатель (родительская сущность)
	basket_status	Статус корзины
	basket_payment	Способ оплаты
	basket_delivery	Способ доставки
	basket_cost	Сумма заказа
	basket_quantity	Количество товаров

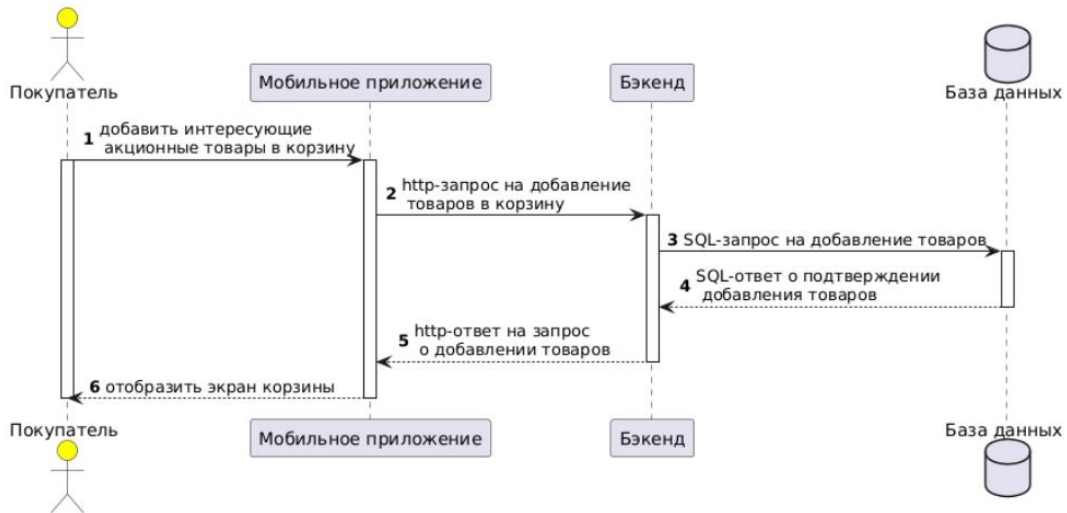
Объект: «Категории»

Родительская сущность	Атрибут	Описание
categories	-	Объект категорий, который имеет атрибуты и ссылки на другие объекты
	target_group	Целевая группа (мужские, женские, детские)
	category_type	Основная категория (одежда, обувь, аксессуары, красота, дом)
	subcategory_type	Подкатегория (платья, блузки, джинсы, сумки, зонты, шарфы и т.д.)

3. ER-диграмма



4. Sequence diagram



№	Описание
1	Покупатель добавляет интересные акционные товары в корзину
2	Мобильное приложение отправляет http-запрос на добавление акционных товаров в корзину на бэкенд
3	Бэкенд формирует SQL-запрос для добавления товаров в корзину в базу данных
4	База данных возвращает SQL-ответ с подтверждением добавления товаров в корзину
5	Бэкенд передает http-ответ о добавлении товаров в корзину
6	Мобильное приложение отображает экран корзины с добавленными в корзину акционными товарами

5. REST – табличный вид

№1

Request

GET /baskets/{basket_id}

Получение стоимости корзины по идентификатору

Название параметра	Тип данных	Находится в	Описание	Обязательность параметра
basket_id	number	path	Уникальный идентификатор корзины	да

Response

Название параметра	Тип данных	Находится в	Описание	Обязательность параметра
basket_id	number	body	Уникальный идентификатор корзины	да
basket_cost	number	body	Стоимость корзины	да

№2

Request

POST /buyers

Создание покупателя в системе

Название параметра	Тип данных	Находится в	Описание	Обязательность параметра
surname	string	body	Фамилия покупателя	да
name	string	body	Имя покупателя	да
telephone_num	string	body	Телефон покупателя	да
email	string	body	Электронная почта покупателя	нет
personal_sale	number	body	Персональная скидка	нет

Response

Название параметра	Тип данных	Находится в	Описание	Обязательность параметра
code	number	body	Код ответа	да
description	string	body	Описание кода ответа	да
buyer_id	number	body	Уникальный идентификатор покупателя	да

surname	string	body	Фамилия покупателя	да
name	string	body	Имя покупателя	да
telephone_num	string	body	Телефон покупателя	да
email	string	body	Электронная почта покупателя	нет
personal_sale	number	body	Персональная скидка	нет

6. Swagger

Swagger Example - OpenAPI 3.0 1.0.0 OAS 3.0

Документирование REST-вызовов в Swagger

Servers

https://lamoda_demo.swagger.io/api/v3

basket Группа методов для работы с корзиной

GET /baskets/{basket_id} Получить стоимость корзины

Получение стоимости корзины по её идентификатору

Parameters

Try it out

Name	Description
basket_id * required	Уникальный идентификатор корзины
integer (path)	<input type="text" value="basket_id"/>

Responses

Code	Description	Links
200	OK <div>Media type: <input type="text" value="application/json"/></div> <div>Controls: Accept header.</div> <div>Example Value Schema</div> <pre>{ "basket_id": 4, "basket_cost": 1118.5 }</pre>	No links
400	Bad Request	No links

POST

/buyers

Создать покупателя

^

Создание нового покупателя в системе

Parameters

Try it out

No parameters

Request body

application/json

▼

Параметры для создания покупателя

Example Value

Schema

```
{
  "surname": "Власова",
  "name": "Белла",
  "telephone_num": "+7-999-321-32-21",
  "email": "bellavl@gmail.com",
  "personal_sale": 10.25
}
```

Responses

Code	Description	Links
201	<div>Created</div> <div>Media type</div> <div>application/json</div> <div>▼</div> <div>Control: Accept header.</div> <div><div>Example Value</div><div>Schema</div></div> <div><pre>{ "code": 201, "description": "Created", "buyer_id": 5, "surname": "Власова", "name": "Белла", "telephone_num": "+7-999-321-32-21", "email": "bellavl@gmail.com", "personal_sale": 10.25 }</pre></div>	No links
400	Bad Request	No links

Schemas

```
Basket {  
  basket_id*  
    integer($int32)  
    example: 4  
    Уникальный идентификатор корзины  
  
  basket_cost*  
    number($float)  
    example: 1110.5  
    Стоимость корзины  
  
}
```

```
Buyer {  
  surname*  
    string  
    example: Иванов  
    Фамилия покупателя  
  
  name*  
    string  
    example: Генна  
    Имя покупателя  
  
  telephone_num*  
    string  
    example: +7-999-321-32-21  
    Номер телефона покупателя  
  
  email  
    string  
    example: belilavi@gmail.com  
    Электронная почта покупателя  
  
  personal_sale  
    number($float)  
    example: 10.25  
    Персональная скидка покупателя  
  
}
```

```
Response {  
  code*  
    integer($int32)  
    example: 201  
    Код ответа  
  
  description*  
    string  
    example: Created  
    Описание ответа  
  
  buyer_id*  
    integer($int32)  
    example: 5  
    Уникальный идентификатор созданного покупателя  
  
  surname*  
    string  
    example: Иванов  
    Фамилия покупателя  
  
  name*  
    string  
    example: Генна  
    Имя покупателя  
  
  telephone_num*  
    string  
    example: +7-999-321-32-21  
    Номер телефона покупателя  
  
  email  
    string  
    example: belilavi@gmail.com  
    Электронная почта покупателя  
  
  personal_sale  
    number($float)  
    example: 10.25  
    Персональная скидка покупателя  
  
}
```

Критерии приемки и Нефункциональные требования

1. Критерии приемки

Критерии приемки — это формализованные требования, которые описывают, каким образом система должна работать, чтобы пользователь или заказчик признали функциональность выполненной. Критерии приемки состоят из кейсов "Дано - Когда – Тогда".

US: Как пользователь мобильного приложения Lamoda, Я хочу иметь возможность сформировать заказ из акционных товаров, Чтобы совершить выгодную покупку.

Функциональность: Управление содержимым корзины

Номер кейса: 1

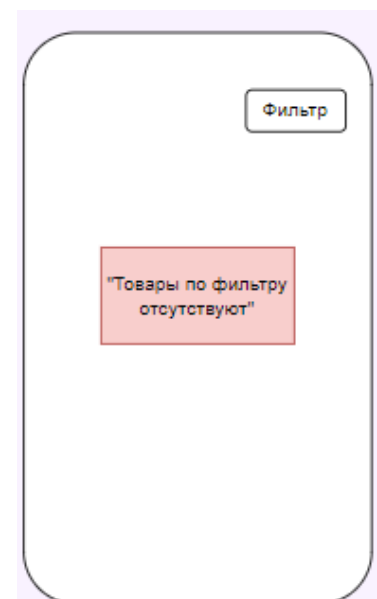
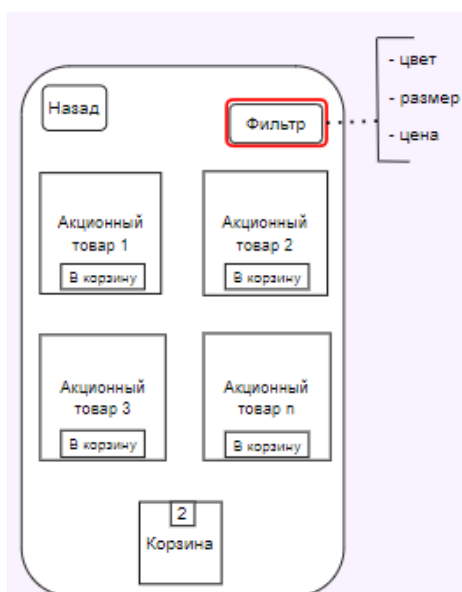
Дано: Покупатель зарегистрирован и авторизован в приложении Lamoda, находится на странице акционных товаров (экран 3)

Когда: Покупатель нажимает на кнопку «Корзина»

Тогда: Система отображает список товаров. Если товара нет в наличии, появляется надпись красного цвета «Нет в наличии».

Функциональность: Отображение товаров по фильтру

Номер кейса: 2



Дано: Покупатель находится на странице со списком акционных товаров, принадлежащими к выбранной категории и подкатегории, с возможностью фильтрации (экран 3)

Когда: Покупатель выбирает фильтры «цвет, размер, цена»

Тогда: Система отображает при отсутствии выбранной покупателем комбинации фильтров (например, цвет: красный, размер: 38, цена: до 5000) ошибку «Товары по фильтру отсутствуют» (экран 6)

Функциональность: Просмотр актуальных лучших акций

Номер кейса: 3

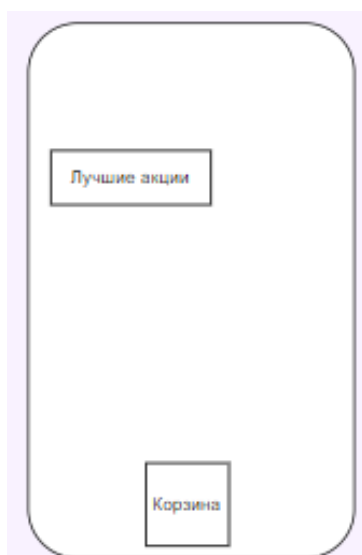


Рисунок 3 – Экран 0

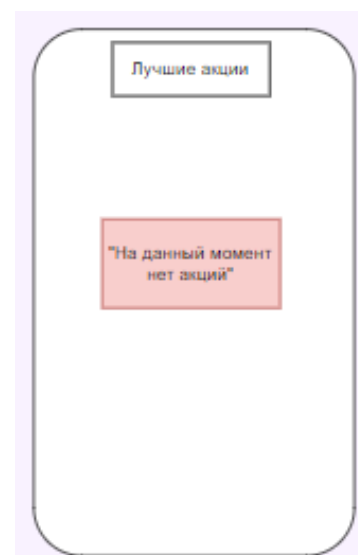


Рисунок 4 – Экран 5

Дано: Покупатель зарегистрирован и авторизован в приложении Lamoda, акций нет (экран 0)

Когда: Покупатель находится на странице «Лучшие акции»

Тогда: Система отображает ошибку «На данный момент нет акций» (экран 5)

2. Нефункциональные требования

НФТ относятся к атрибутам качества системы, которые определяют, как она работает, а не что она делает.

Требования производительности:

1. Загрузка страницы со списком акционных товаров, принадлежащих категории и подкатегории, должна занимать не более 2 секунд
2. Загрузка страницы корзины должна занимать не более 1,5 секунд
3. Запрос отображения корзины GET /buyers/buyer_id/baskets должен выдерживать нагрузку 1 gps

Требования надежности:

1. Система должна быть доступна 99% времени

Требования удобства использования:

1. Интерфейс приложения должен быть ясным и интуитивно понятным для покупателей
2. Процесс регистрации для покупателей должен занимать не более 3 минут

Требования безопасности:

1. Личные данные покупателей, передаваемые между клиентом и сервером, должны быть зашифрованы

Требования масштабируемости:

1. Архитектура приложения должна поддерживать масштабируемость для ежегодного роста новых зарегистрированных покупателей на 15%