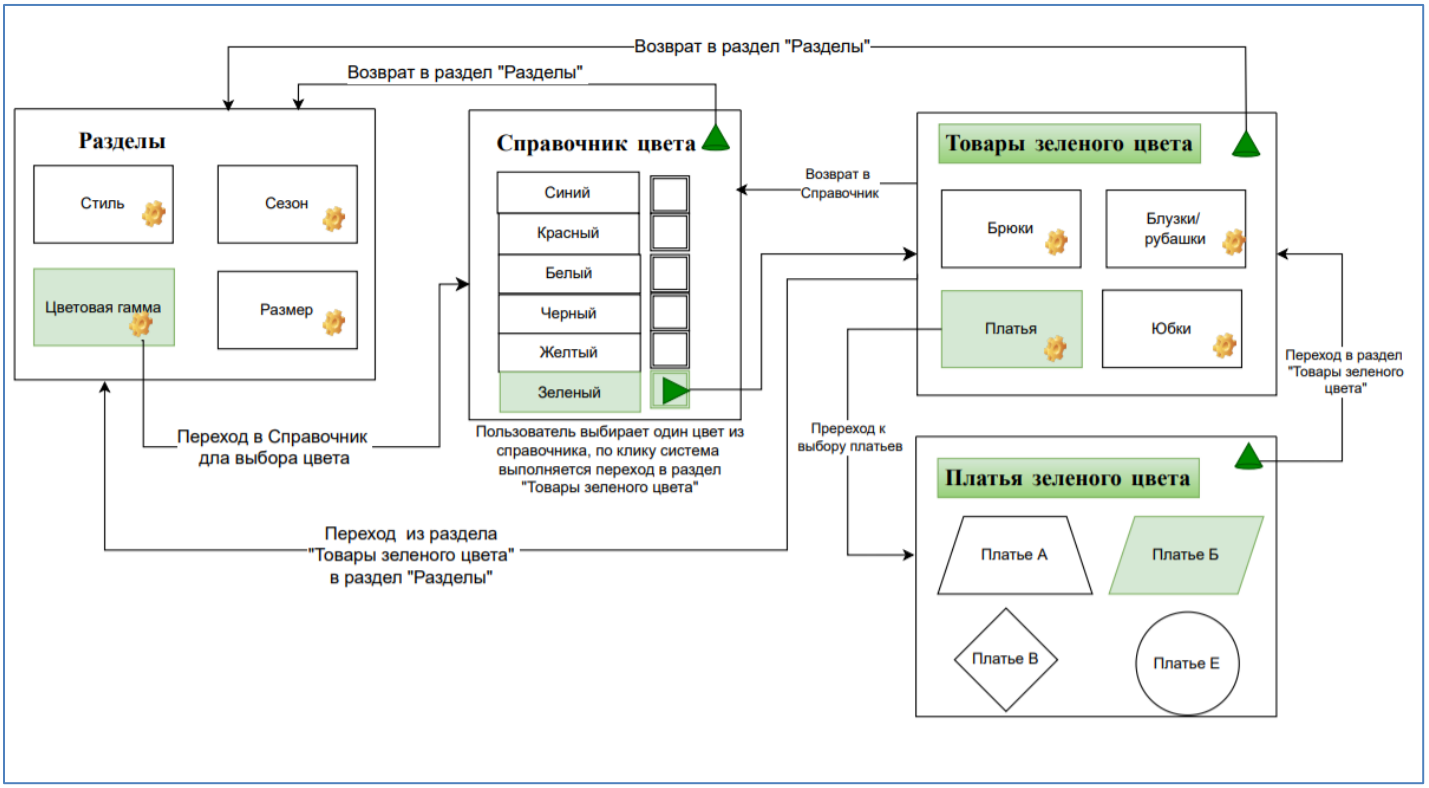
1. Название продукта -
   Интернет - магазин

2. User story

Я как пользователь Интернет-магазина, хочу увидеть товары заданного цвета, чтобы быстрее найти нужный товар. (Одежда)

Кто	*Потребность*	*Цель*
Я как пользователь интернет-магазина	***хочу увидеть товары заданного цвета***	***чтобы быстрее найти нужный товар***

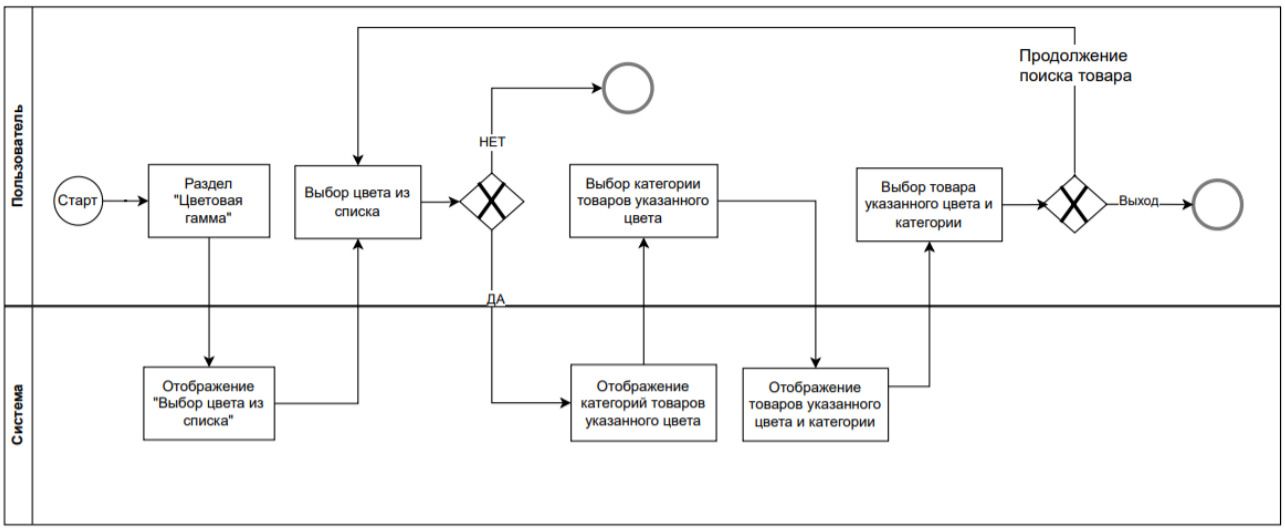
3. Макеты



4. Use case

Заголовок	Поиск одежды по цвету
Акторы	Пользователь интернет-магазина
Предусловие	Пользователь авторизован и идентифицирован на сайте интернет-магазина, находится в разделе "Разделы"
Ограничения	1) Ограничение к выбору цвета (фиксирован перечень цветов) 2) Ограничение к выбору видов одежды (фиксирован перечень видов одежды) 3) Можно выбрать для поиска только один цвет
Триггер	Пользователь переходит в раздел "Цветовая гамма"
Основной сценарий	1. Система отображает список цветов, доступных к выбору (экран 2), 2. Пользователь выбирает нужный цвет из списка и переходит в раздел товаров указанного цвета, 3. Система отображает категории товаров (экран 3), 4. Пользователь выбирает нужную категорию, 5. Система отображает товары выбранной категории (экран 4), **Критерий успеха:** Пользователь нашел нужный товар
Альтернативный сценарий	
Исключительный сценарий	3А Нет нужной категории товаров 5А В данной категории нет товаров

5. BPMN



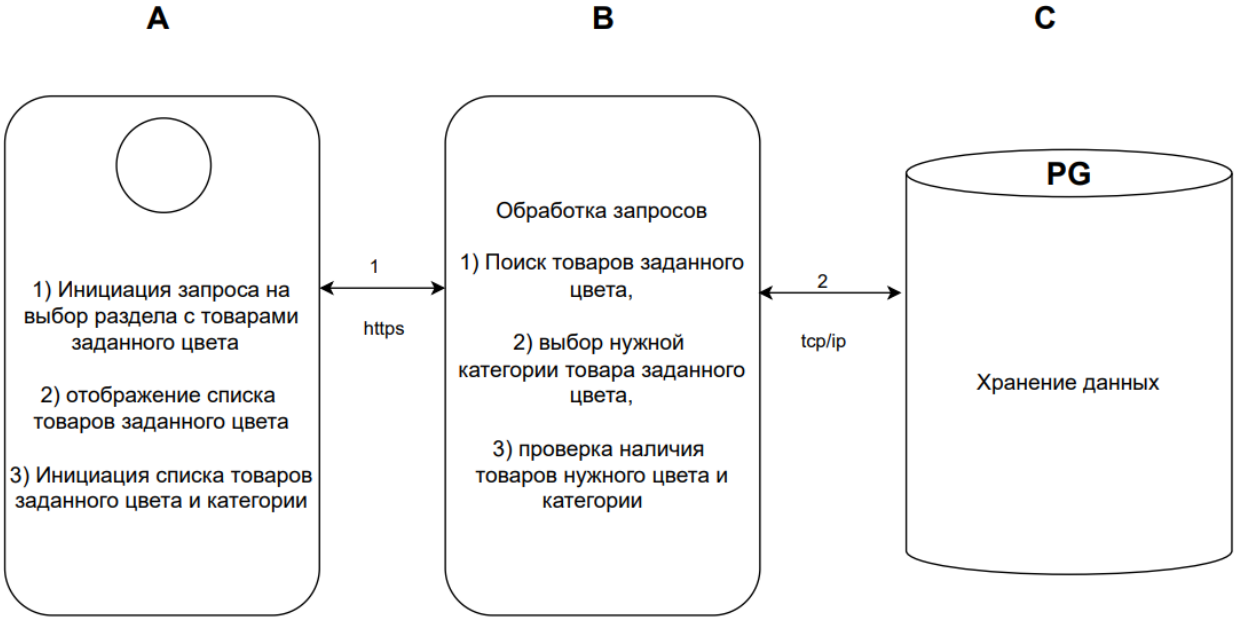
6. Архитектура

Фронтенд – это схематичный веб или мобильное приложение системы. Также фронтенд называют клиентом

Бэкенд – это внутренний сервер системы проекта. Сервер обращается к базе данных с полученными данными от клиента (фронтенда)

База данных – это хранилище данных, с которыми будет работать сервер

Мой вариант



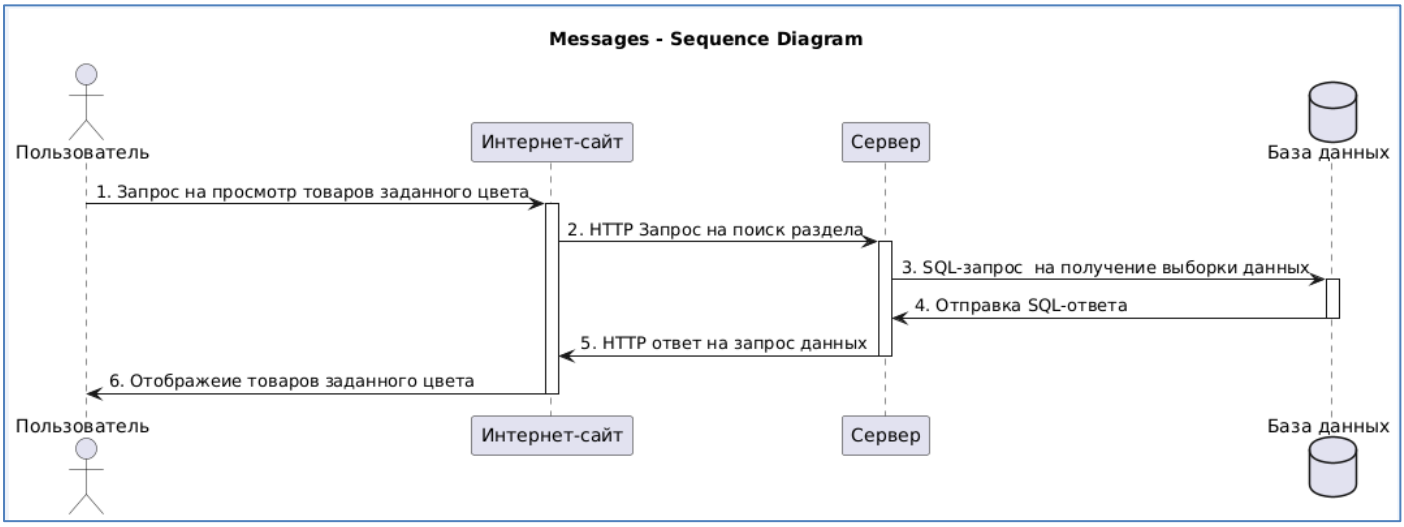
A - Интернет-сайт, Frontend
B - Сервер, Backend
C - Реляционная База данных, PostgreSQL

1 - протокол HTTPS

7. Диаграмма последовательности

Диаграмма последовательности — это тип UML-диаграммы, который используется для моделирования взаимодействий между объектами в системе. Она отображает, в каком порядке и как взаимодействуют между собой различные объекты или компоненты во времени. Основное назначение — показать, как запросы и ответы передаются между элементами системы.

Мой вариант:

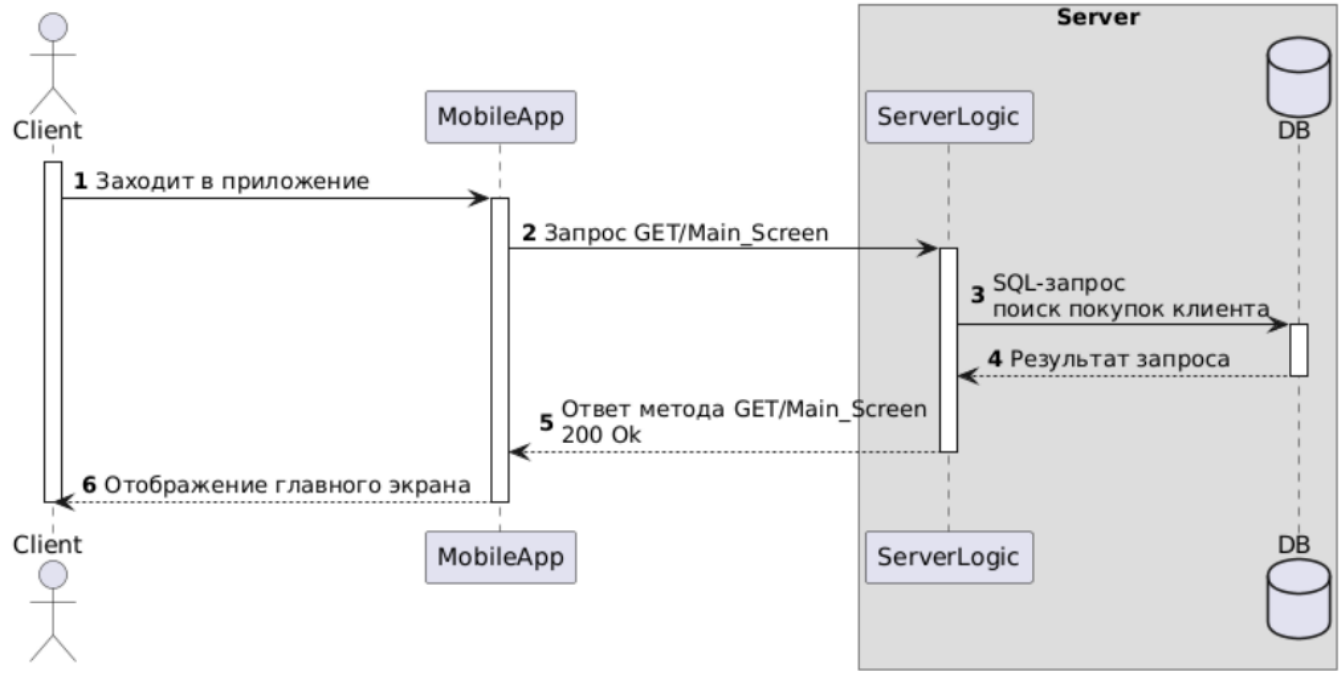


Номер	Описание
1	Запрос на просмотр товаров заданного цвета
2	HTTP - запрос на поиск нужного раздела
3	SQL - запрос на формирование выборки данных (товаров заданного цвета)
4	Отправка SQL - ответа
5	HTTP – ответ на запрос данных
6	Отображение товаров заданного цвеьа

Пример

№	Описание
Номер шага	Описание того, что происходит на шаге

Пример



№	Описание
1	Вход пользователя на главный экран приложения или нажатие на кнопку "Подобрано для вас" из экрана "Хиты продаж"
2	Запрос на отображение главного экрана
3	SQL-запрос на получение покупок по данному клиенту
4	Возврат данных, содержащих все покупки клиента
5	Успешный ответ метода GET/Main_Screen – 200 OK. В ответе содержатся данные о покупках клиента
6	Отображение главного экрана приложения со списком покупок пользователя

8. Модель данных

Модель данных – это представление данных, атрибутного состава сущностей, как сущности связаны друг с другом.

Шаблон – Мой вариант

Объект Product

Родительская сущность	Атрибут	Описание
Product		Товар
	Style	**Стиль** Возможные значения: 1)Классический стиль 2) Спортивный 3) Кэжуал
	Season	**Сезон**

		Возможные значения: • Зима • Лето • Межсезонье
	Category	**Категория товара** Возможные значения: 1) Брюки 2) Юбка 3) Блуза/ рубашка 4) Платье

Объект Size

Дочерняя сущность	Атрибут	Описание
Size		**Размер одежды**
	clothing size chart	**Сетка размеров одежды** Возможные значения • Российский размер • Европейский • джинсы
	Gender_size	**Для кого** **Возможные значения:** • Женщины • Мужчины • дети
	Size	**Размер одежды** Возможные значения XS, S, M, L, XL или 42, 44, 46, 48, 50 ...

Объект Colour

Дочерняя сущность	Атрибут	Описание
Colour		**Цвет**
	Blue	Синий
	Red	Красный
	White	Белый
	Green	Зеленый
	black	Черный
	green	Зеленый

Пример

Объект User

Родительская сущность	Атрибут	Описание
User	----	Объект пользователя, который имеет атрибуты и ссылки на другие объекты
	FootSize	Размер ноги
	Gender	Пол
	WorkingAdress	Рабочий адрес. Ссылка на объект рабочий адрес.
	WorkingPhone	Рабочий телефон. Ссылка на объект рабочий телефон.

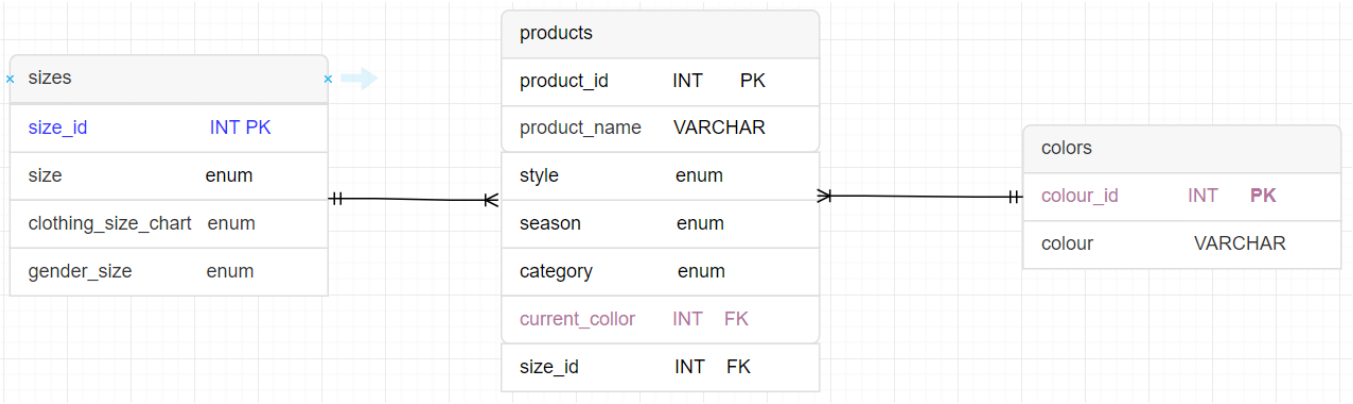
Объект WorkingAdress

Родительская сущность	Атрибут	Описание
WorkingAdress	----	Объект рабочего адреса.
	StreetName	Название улицы. Например, "Ленина".
	HomeIndex	Номер дома. Например, "14"
	PostIndex	Почтовый индекс. Например "644876"

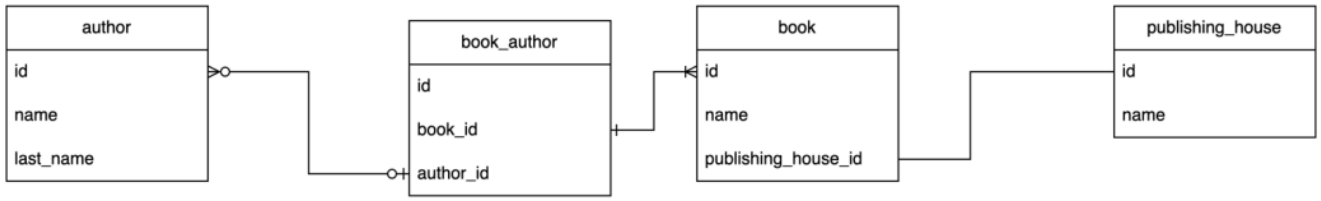
9. Диаграмма ERD

ERD-диаграмма — диаграмма, где показано, как разные «сущности» (люди, объекты, концепции и так далее) связаны между собой внутри системы.

Шаблон (мой вариант)



Пример



10. REST. Табличный вид

REST API подход использует HTTP-методы (GET, POST, PUT, DELETE и т.д.) для управления сущностями с уникальными URL.

Шаблон

Request

Название параметра	Тип данных	Находится в	Описание	Обязательность параметра

Response

Название параметра	Тип данных	Находится в	Описание	Обязательность параметра

Response code <200>

Мой вариант

Я как пользователь Интернет-магазина, хочу увидеть товары заданного цвета, чтобы быстрее найти нужный товар. (Одежда)

GET

Пользователь запрашивает платья зеленого цвета

get /products/product_name?category="dress"&colour="green"

Request Запрос платьев зеленого цвета

Название Query параметра	Принадлежность	Тип данных	Описание	Обязательность параметра
Category	quary	string	Категория товара - платье	нет
Colour	quary	String	Цвет – товара зеленый	нет

Response Сервер возвращает платья зеленого цвета

Название параметра			Принадлежность	Тип данных	Описание	Обязательность параметра
Arr_colour_category			body	Array of object	Массив платьев зеленого цвета	Да
		Product_id	body	int	Уникальный идентификатор продукта	Нет
		Category	body	string	Категория продукта	Нет
		Colour	body	string	Цвет продукта	Нет
		Product	body	String	Наименование продукта	нет

Запрос возвращает code 200

Пример

Get /v1/users/{userId}/purchases

Получение всех покупок определенного пользователя

Request

Название параметра	Тип данных	Находится в	Описание	Обязательность параметра
userId	int	path	Уникальный идентификатор пользователя	да

Response

Название параметра	Тип данных	Находится в	Описание	Обязательность параметра
userId	int	body	Уникальный идентификатор пользователя	да
purchases	array	body	Массив покупок	да
purchase	object	body	Объект покупки	нет
purchaseId	int	body	Уникальный идентификатор покупки	да
date	string	body	Дата покупки	да
shopId	int	body	Магазин, в котором совершили покупку	да

Response code <200>

11. Swagger

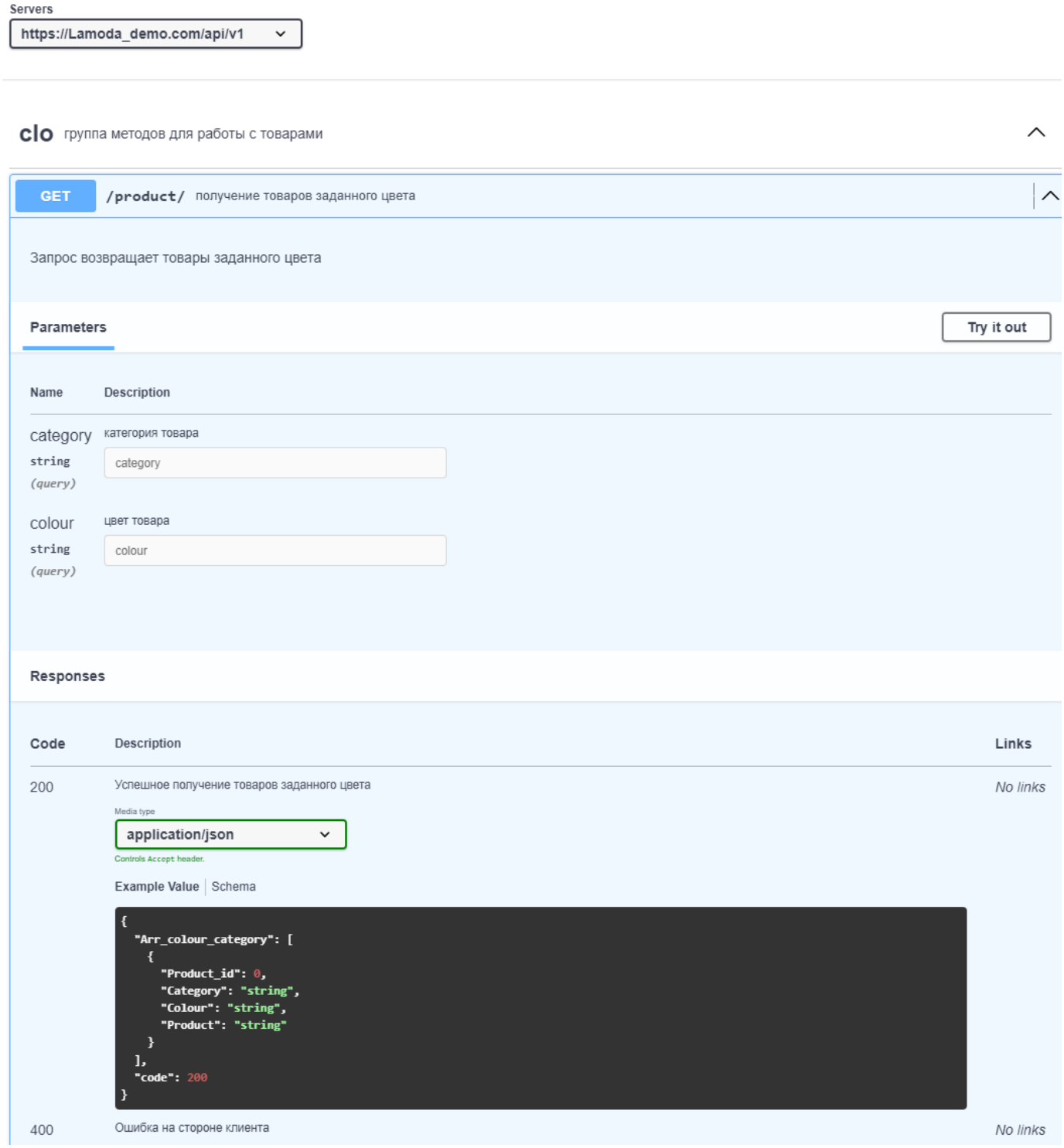
Swagger — это инструмент, который помогает аналитикам, разработчикам и тестировщикам создавать, документировать и проверять API.

Для чего нужен swagger системным аналитикам:

• Создание документации

• Тестирование API

Мой вариант



openapi: 3.0.3
info:
  title: Swagger - OpenAPI 3.0
  description: Документирование REST-вызовов в Swagger
  version: 1.0.0

servers:
  - url: https://Lamoda_demo.com/api/v1

tags:

```
  - name: clo
    description: группа методов для работы с товарами

paths:
    /product/:
      get:
        tags:
          - clo
        summary: получение товаров заданного цвета
        description: Запрос возвращает товары заданного цвета
        parameters:
        - name: category
          in: query
          description: категория товара
          required: false
          schema:
            type: string
        - name: colour
          in: query
          description: цвет товара
          required: false
          schema:
            type: string
        responses:
          '200':
            description: Успешное получение товаров заданного цвета
            content:
              application/json:
                schema:
                  $ref: '#/components/schemas/Product'

          '400':
            description:  Ошибка на стороне клиента
          '500':
            description:  Error

components:
   schemas:
     Product:
        type: object
        required:
          - Arr_colour_category
          - code
        properties:
          Arr_colour_category:
            type: array
            description: Массив одежды заданного цвета и категории
            items:
```

```
    properties:
      Product_id:
        type: integer
        description: Уникальный идентификатор товара
      Category:
        type: string
        description: Категория товара (рубашка,платье...)
      Colour:
        type: string
        description: Цвет товара (одежды)
      Product:
        type: string
        description: Конкретная единица одежды
code:
  type: integer
  example: 200
  description: код ответа
```

12. Критерии приемки

US: Я как пользователь Интернет-магазина, хочу увидеть товары заданного цвета, чтобы быстрее найти нужный товар. (Одежда).

Описание кейса

Критерии приемки состоят из кейсов "Дано - Когда – Тогда".

Функциональность: Пользователь просматривает товары заданного цвета для выбора и покупки

Успешный сценарий

Заголовок	Поиск одежды по цвету
Акторы	Пользователь интернет-магазина
Предусловие	Пользователь авторизован и идентифицирован на сайте интернет-магазина, находится в разделе "Разделы"
Ограничения	1) Ограничение к выбору цвета (фиксирован перечень цветов) 2) Ограничение к выбору видов одежды (фиксирован перечень видов одежды) 3) Можно выбрать для поиска только один цвет
Триггер	Пользователь переходит в раздел "Цветовая гамма"
Основной сценарий	1. Система отображает список цветов, доступных к выбору (экран 2), 2. Пользователь выбирает нужный цвет из списка и переходит в раздел товаров указанного цвета, 3. Система отображает категории товаров (экран 3), 4. Пользователь выбирает нужную категорию, 5. Система отображает товары выбранной категории (экран 4), **Критерий успеха:** Пользователь нашел нужный товар
Альтернативный сценарий	
Исключительный сценарий	3А Нет нужной категории товаров 5А В данной категории нет товаров

Номер кейса: 1.

Дано: Пользователь авторизован в Интернет-магазине, находится в разделе «Разделы/ Цветовая гамма»

Когда: **Пользователь** нажал на зеленый цвет;

Тогда: система выводит сообщение об ошибке: “Не найдено товаров зеленого цвета”

Номер кейса: 2.

Дано: пользователь на экране с категориями товаров заранее выбранного цвета. В категории «Платья» зеленых нет

Когда: пользователь нажал на категорию «Платья»
Тогда: система выводит сообщение об ошибке: “Не найдено товаров заданной категории и цвета”

Номер кейса: 3
Дано: **Пользователь** авторизован в Интернет-магазине, выбрал рубашку размера 50, находится в разделе «Разделы/ Размер»
Когда: Пользователь нажимает кнопку размера «50»,
Тогда: система выводит сообщение об ошибке: “Рубашки размера 50 временно отсутствуют в магазине”.

Требования надежности:
1. Система должна быть доступна 95% времени.

Требования производительности:
1. Страница просмотра запрошенных товаров должна открываться не более чем 3 секунды
2. Магазин должен выдерживать нагрузку в 1000 посетителей онлайн одновременно