

Министерство науки и высшего образования РФ
Федеральное государственное автономное
образовательное учреждение высшего образования
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт космических и информационных технологий
Кафедра вычислительной техники

УТВЕРЖДАЮ
Заведующий кафедрой
_____ О.В. Непомнящий
« ___ » _____ 2024 г.

БАКАЛАВРСКАЯ РАБОТА

090301 Информатика и вычислительная техника

Разработка системы дропшипинга для интернет-магазина

Руководитель	_____	_____	доцент, канд. техн. наук	М.С. Медведев
	<i>подпись</i>	<i>дата</i>	<i>должность, ученая степень</i>	
Выпускник	_____	_____		Л.В. Колесник
	<i>подпись</i>	<i>дата</i>		
Нормоконтролёр	_____	_____	доцент, канд. техн. наук	М.С. Медведев
	<i>подпись</i>	<i>дата</i>	<i>должность, ученая степень</i>	

Красноярск 2024

Министерство науки и высшего образования РФ
Федеральное государственное автономное
образовательное учреждение высшего образования
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт космических и информационных технологий

Кафедра вычислительной техники

УТВЕРЖДАЮ

Заведующий кафедрой

_____ О.В. Непомнящий

« ___ » _____ 2023 г.

**ЗАДАНИЕ
НА ВЫПУСКНУЮ КВАЛИФИКАЦИОННУЮ РАБОТУ
в форме бакалаврской работы**

Красноярск 2023

Студенту Колеснику Леониду Викторовичу
фамилия, имя, отчество

Группа КИ20-07Б Направление (специальность) 09.03.01
номер код

Информатика и вычислительная техника
полное наименование

Тема выпускной квалификационной работы: Разработка системы
дроппинга для интернет-магазина

Утверждена приказом по университету № _____ от _____

Руководитель ВКР: М.С. Медведев, кандидат технических наук, доцент
ВТ ИКИТ
инициалы, фамилия, учёная степень, должность, место работы

Исходные данные для ВКР:

1) задание на ВКР;

2) Swagger;

3) интернет-магазин fireboxclub.com.

Перечень разделов ВКР:

1) анализ предметной области;

2) этапы разработки системы дроппинга;

3) реализация системы дроппинга;

Перечень графического материала: Презентация в формате Microsoft
PowerPoint

Руководитель ВКР _____
подпись М.С. Медведев
инициалы, фамилия

Задание принял к исполнению _____
Подпись, инициалы и фамилия Л.В. Колесник

« » _____ 2023 г.

РЕФЕРАТ

Выпускная квалификационная работа по теме «Разработка системы дропшипинга для интернет-магазина» содержит 46 страниц текстового документа, 24 изображение, 1 таблицу и 10 использованных источников.

ИНТЕРНЕТ-МАГАЗИН, SWAGGER, PHP.

Цель работы: разработка API для интернет-платформы, предназначенной для дропшипперов, позволяющей автоматизировать процесс запуска собственного интернет-магазина с функцией дропшипинга.

При выполнении данной работы был произведен обзор предметной области, задания на выпускную квалификационную работу, изучены существующие аналоги и сформированы требования.

Объект работы – API для интернет-магазина.

Задачи:

- выполнить проектирование разрабатываемого API сайта;
- осуществить выбор программных средств разработки API сайта;
- выполнить программную реализацию API сайта;
- проанализировать полученные результаты работы.

СОДЕРЖАНИЕ

Введение.....	3
1 Анализ задания на ВКР.....	4
1.1 Существующие аналоги	4
1.1.1 Outmaxshop.ru.....	4
1.1.2 Nazya.com	5
1.1.3 Markethot.ru.....	6
1.2 Вывод по анализу аналогов.....	7
1.3 Требования к разрабатываемому модулю	8
1.4 Сравнительный анализ средств разработки.....	8
1.4.1 PHP.....	9
1.4.2 Swagger.....	9
1.5 Выводы по главе.....	10
2 Проектирование.....	11
2.1 Диаграммы последовательности	11
2.1.1 Прецедент «Запрос токена»	11
2.1.2 Прецедент «Расчёт стоимости доставки».....	12
2.1.3 Прецедент «Получить список товаров»	13
2.1.4 Прецедент «Получить список цен товаров»	14
2.1.5 Прецедент «Получение информация о товаре».....	15
2.1.6 Прецедент «Получение информации о размерах товара».....	16
2.2 Структурная схема системы	18
2.3 Вывод по главе	19
3 Разработка методов API.....	20
3.1 Swagger.....	20
3.1.1 Метод для аутентификации и получения токена	20
3.1.2 Метод для расчёта стоимости доставки	22
3.1.3 Метод для получения списка товаров	27
3.1.4 Метод для получения списка цен товаров	31
3.1.5 Метод для получения информации о товаре	34
3.1.6 Метод для получения информации о размерах товара.....	37
3.2 Безопасность.....	40
3.3 PHP	41

3.4 Вывод по главе	44
Заключение	45
Список использованных источников	46

ВВЕДЕНИЕ

Дропшипперы – это люди или компании, которые продают товары через интернет, но не хранят их на складе. Вместо этого они передают заказы поставщикам, которые напрямую отправляют товары покупателям. Дропшипперы зарабатывают на разнице между ценой, по которой они продают товар, и ценой, по которой закупают его у поставщика, без необходимости заниматься инвентаризацией или доставкой. Развитие электронной коммерции и модели бизнеса дропшиппинга становятся все более значимыми в современном мире. Однако для многих потенциальных предпринимателей, особенно для тех, кто хочет начать бизнес на основе дропшиппинга, остается сложным создание и запуск собственного интернет-магазина. Тема, связанная с разработкой платформы для дропшипперов, позволяющей быстро и эффективно запустить персонализированный интернет-магазин, обладает высокой актуальностью. Такой проект отвечает запросам рынка, предоставляя инструменты для автоматизации процесса запуска бизнеса в сфере электронной коммерции. Это сокращает временные и финансовые затраты предпринимателей, делает доступ к онлайн-торговле более доступным и удобным, что в свою очередь способствует развитию предпринимательства и электронной коммерции в целом.

Цель работы – разработка API для интернет-платформы, предназначенной для дропшипперов, позволяющей автоматизировать процесс запуска собственного интернет-магазина с функцией дропшиппинга. Основной упор делается на создание логики, передачи дропшипперам данных от главного интернет-магазина к их собственному интернет-магазину. Целью является обеспечение предпринимателей возможностью начать и управлять своим онлайн-бизнесом на основе дропшиппинга с минимальными техническими и временными затратами, способствуя их успешному вхождению в сферу электронной коммерции.

1 Анализ задания на ВКР

Перед началом разработки API для системы дропшипинга необходимо провести анализ уже существующих решений данной задачи, а также определить требования к будущей системе. API (Application Programming Interface) - это набор определенных правил и соглашений, которые определяют, как различные компоненты программного обеспечения могут взаимодействовать друг с другом. API определяет, какие функции и процедуры могут быть вызваны, как передавать данные и какие форматы данных использовать при этом взаимодействии.

1.1 Существующие аналоги

Существует множество интернет-магазинов, предлагающие функцию дропшипинга. Рассмотрим к примеру outmaxshop.ru, nazya.com и markethot.ru.

1.1.1 Outmaxshop.ru

Страница для оставления заявки на дропшипинг интернет-магазина outmaxshop.ru показана на рисунке 1:

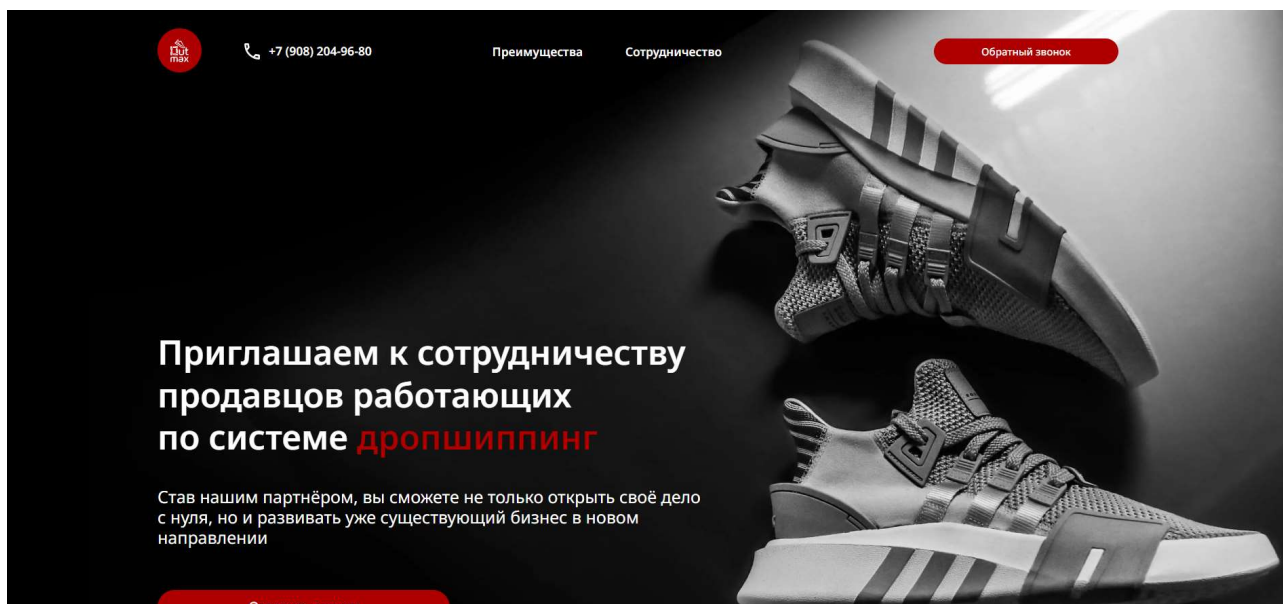


Рисунок 1 — Страница для оставления заявки на дропшипинг

На сайте легко найти страницу, чтобы оставить заявку. Преимущества данного интернет-магазина:

- возможность покупать товар от одной единицы;
- большой ассортимент товара, представленный репликами известных спортивных брендов;
- еженедельно обновляемый модельный ряд, реальные фотографии товара, актуальные цены и остаток;
- персональный менеджер, который будет курировать заказы и помогать в процессе работы;
- свои цены. Продавец сам устанавливает наценку на товар.

Однако если вы оставите заявку на дропшипинг, вы не получите свой личный сайт, вам сдадут в аренду домен и готовый сайт, в котором вы не сможете ничего персонализировать.

1.1.2 Nazya.com

Страница для оставления заявки на дропшипинг интернет-магазина nazya.com показана на рисунке 2:

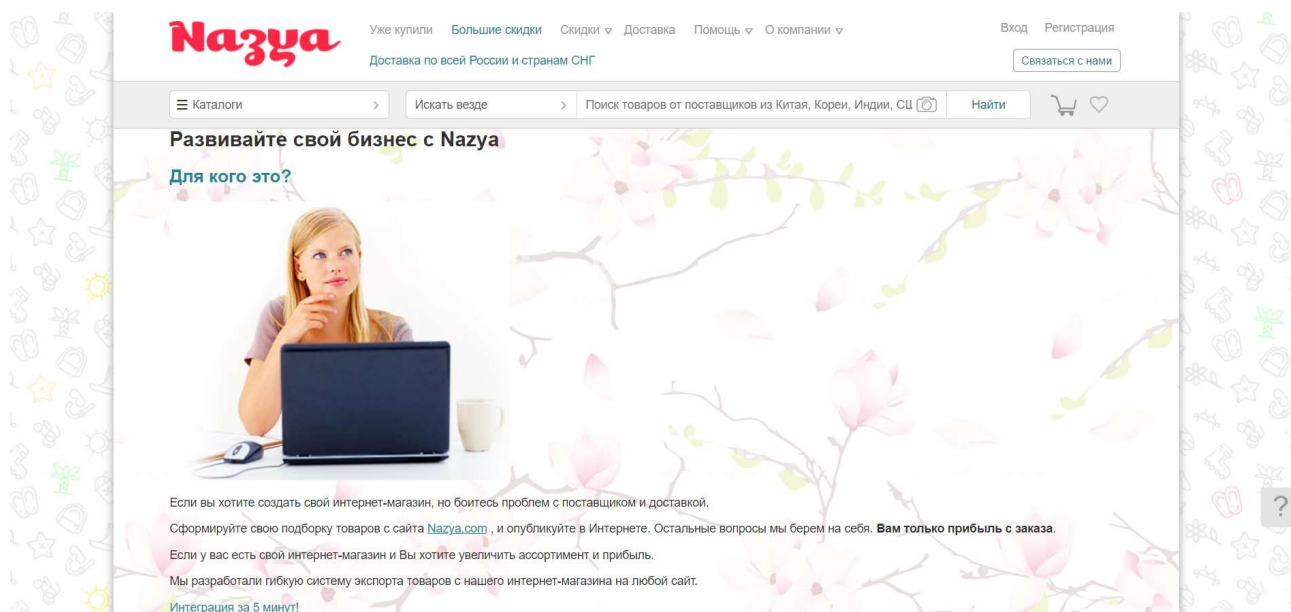


Рисунок 2 — Страница для оставления заявки на дропшипинг

В данном интернет-магазине функция дропшипинга есть, но она не реализована до конца, нужно в поисковой строке ввести путь чтоб попасть на эту страницу. Взаимодействие с сайтом строится путем создания собственного онлайн-ресурса с последующим размещением товаров.

1.1.3 Markethot.ru

Страница для оставления заявки на дропшипинг интернет-магазина markethot.ru показана на рисунке 3:

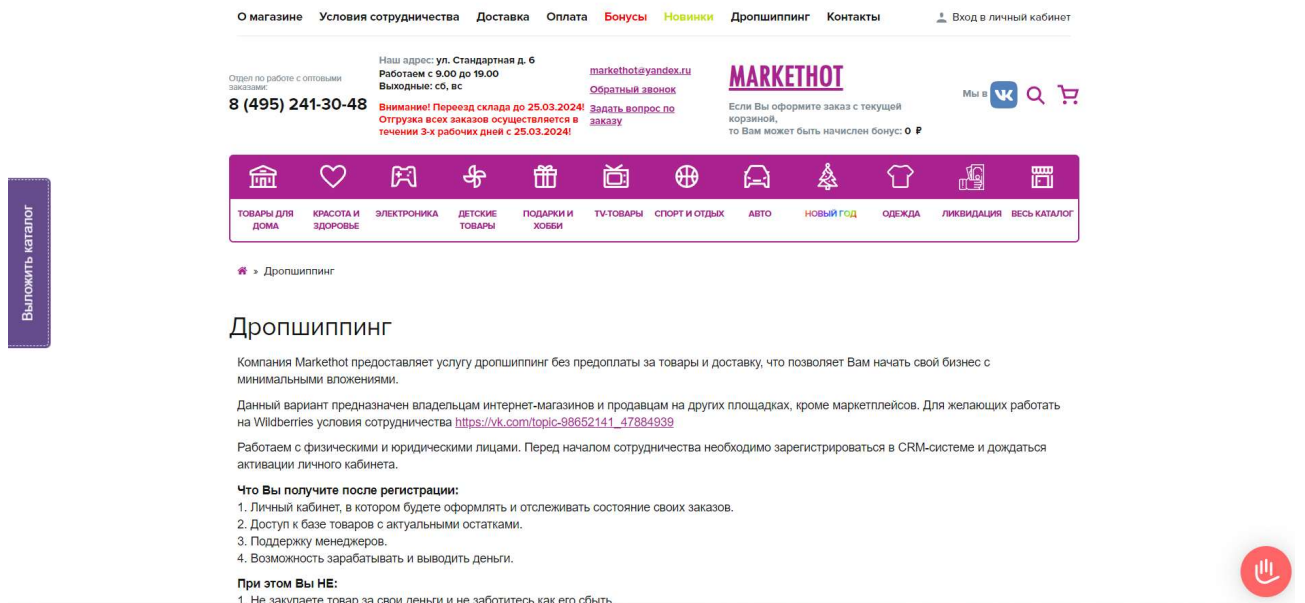


Рисунок 3 — Страница для оставления заявки на дропшипинг

Этот интернет-магазин предлагает схожую с первым примером схему дропшипинга. Вы оставляете заявку и вам дают в аренду домен и сайт, в котором нельзя ничего персонализировать. На этом сайте так же не сложно найти ссылку, чтобы перейти в окно и оставить заявку. Вы получаете доступ к личному кабинету в котором управляете всеми заявками на заказы.

1.2 Вывод по анализу аналогов

Рассмотрев несколько аналогов, была составлена таблица сравнения по следующим критериям (Таблица 1):

- возможность персонализировать сайт;
- получение своего собственного сайта;
- получение личного кабинета.

Таблица 1 — Критерии сравнения сервисов

Сервис	Возможность персонализировать сайт	Получение своего собственного сайта	Получение личного кабинета
Outmaxshop.ru	Нет	Нет	Да
Nazya.com	Нет	Нет	Нет
Markethot.ru	Нет	Нет	Да

В результате анализа аналогов было принято решение, что разрабатываемая система будет иметь следующие возможности:

- возможность дропшипперам персонализировать сайт;
- возможность дропшипперам получить свой собственный сайт;

1.3 Требования к разрабатываемому модулю

Целью данной бакалаврской работы является разработка API для системы дропшиппинга в интернет-магазине.

Система должна решать следующие задачи:

- 1) Авторизация пользователя как дропшиппер;
- 2) Расчёт стоимости доставки;
- 3) Получение списка товаров;
- 4) Получение списка цен товаров;
- 5) Получение информации о товаре;
- 6) Получение информации о размерах товаров.

1.4 Сравнительный анализ средств разработки

В качестве средств разработки были выбраны PHP и Swagger. Данный выбор обусловлен их широкой популярностью, гибкостью и функциональностью. Swagger для описания и создания логики API, а PHP для написания тестов для всех разрабатываемых методов API.

1.4.1 PHP

PHP (Hypertext Preprocessor) — это скриптовый язык программирования общего назначения, который широко применяется для разработки веб-приложений и динамических веб-сайтов. PHP имеет синтаксис, схожий с синтаксисом C, что облегчает его изучение и использование для тех, кто уже знаком с другими языками программирования. PHP обладает мощными функциями для работы с базами данных, обработки форм, создания изображений и многими другими аспектами веб-разработки. Одним из основных применений PHP является создание динамических веб-сайтов. Он может быть интегрирован непосредственно в HTML код, что позволяет вам создавать динамические страницы, обрабатывать данные форм, взаимодействовать с базами данных и многое другое.

1.4.2 Swagger

Swagger — это набор инструментов для разработчиков API.

Автоматическая генерация документации: Swagger позволяет автоматически создавать подробную документацию API на основе аннотаций в исходном коде. Это существенно упрощает процесс документирования и поддержания актуальной информации о API.

Интерактивная документация: Swagger предоставляет интерактивную документацию, которая позволяет разработчикам и пользователям API изучать его функционал, отправлять запросы и просматривать ответы прямо в браузере. Это улучшает понимание и использование API.

Удобство тестирования API: Интерактивная документация Swagger позволяет быстро и удобно тестировать различные эндпоинты API, что помогает выявить и исправить ошибки еще на ранних этапах разработки.

Автоматическая валидация запросов и ответов: Swagger может автоматически валидировать отправляемые запросы и получаемые ответы в соответствии с описанием API, что обеспечивает более высокую надежность и качество работы приложения.

Интеграция с другими инструментами: Swagger легко интегрируется с другими инструментами разработки, такими как фреймворки для создания API.

1.5 Выводы по главе

В первой главе проведен анализ, направленный на изучение предметной области и определение необходимых технологий для разработки системы дропшипинга для интернет-магазина.

В первую очередь проведен анализ сайтов конкурентов, включающий изучение их функциональности. Этот анализ позволил выявить сильные и слабые стороны существующих решений, а также определить требования, которые должны быть учтены при разработке.

Так же был произведен выбор технологий для реализации. В качестве основных технологий были выбраны PHP и Swagger.

2 Проектирование

В качестве основного инструмента проектирования был выбран процесс проектирования ICONIX, как один из более распространённых фреймворков проектирования ПО. Мной были выделены ключевые прецеденты (не являющиеся тривиальными) и построены соответствующие диаграммы пригодности и последовательности, из которых затем синтезированы начальные варианты классов, представленных в диаграммах классов.

2.1 Диаграммы последовательности

Диаграммы последовательности дают возможность глубже рассмотреть взаимодействие подсистем. Поэтому проектирование системы начинается именно с них.

2.1.1 Прецедент «Запрос токена»

Токен – это небольшая строка, представленная в виде случайного набора символов, которая выдается пользователю для подтверждения его легитимности при обращении к ресурсам. Я буду использовать методы GET и POST для взаимодействия, метод GET используется для запроса данных с сервера, а POST для отправки данных на сервер.

На сайте дропшиппера, когда пользователь авторизовывается в свою учетную запись, вводя свой адрес электронной почты и пароль, отправляется GET запрос, сервер сверяется с базой данных и в случае совпадения, генерируется уникальный токен. Если электронной почты нет в базе данных, сервер возвращает ошибку 401 и сообщает, что пользователь не авторизован. Так же возможно, что аккаунт заблокирован, в этом случае сервер вернет ошибку 403 с сообщением о том, что аккаунт заблокирован.

На рисунке 4 представлена диаграмма последовательности прецедента «Запрос токена»

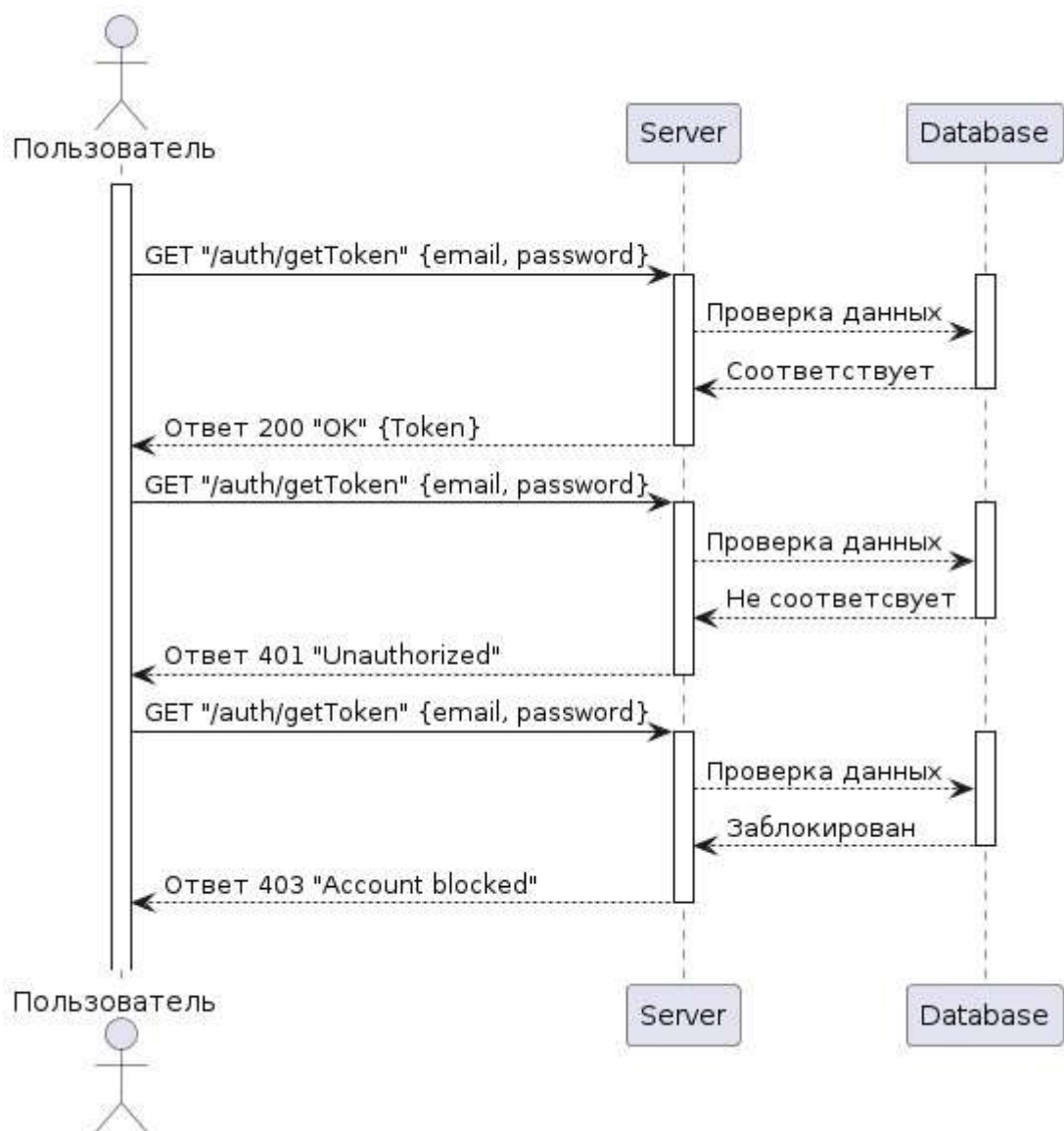


Рисунок 4 – Диаграмма последовательности прецедента «Запрос токена»

2.1.2 Прецедент «Расчёт стоимости доставки»

Когда пользователь на этапе оформления заказа товара, на сервер отправляется POST запрос. Запрос должен включать в себя информацию о месте куда надо доставить товар и информацию о самом товаре. Если пользователь не ввел все необходимые данные, сервер вернет ошибку 400 с сообщением что данные не верны. Если пользователь не авторизован, сервер вернет ошибку 401, сообщая о том, что пользователь не авторизован и если учетная запись заблокирована, вернется ошибка 403.

На рисунке 5 представлена диаграмма последовательности прецедента «Расчёт стоимости доставки»

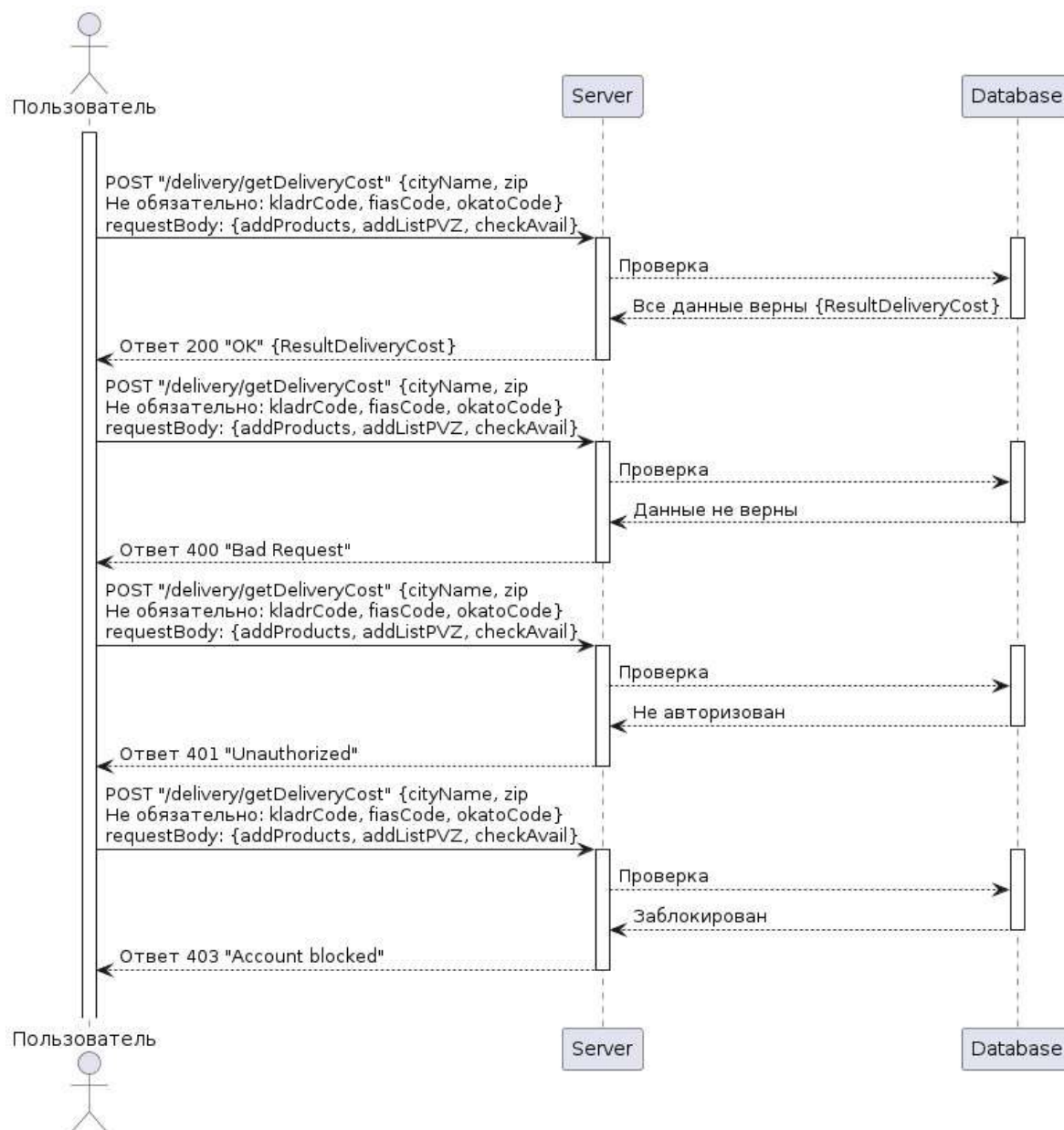


Рисунок 5 – Диаграмма последовательности прецедента «Расчёт стоимости доставки»

2.1.3 Прецедент «Получить список товаров»

Запрос GET для получения списка товаров делается зарегистрированным пользователем интернет-магазина. Пользователь может ввести какую страницу, и сколько товаров на странице отображать. При успешном запросе, сервер вернет ответ 200, и отобразит список товаров. В

случае если пользователь не авторизован, либо его аккаунт заблокирован, сервер вернет ошибку 401 и 403 соответственно.

На рисунке 6 представлена диаграмма последовательности для данного прецедента.

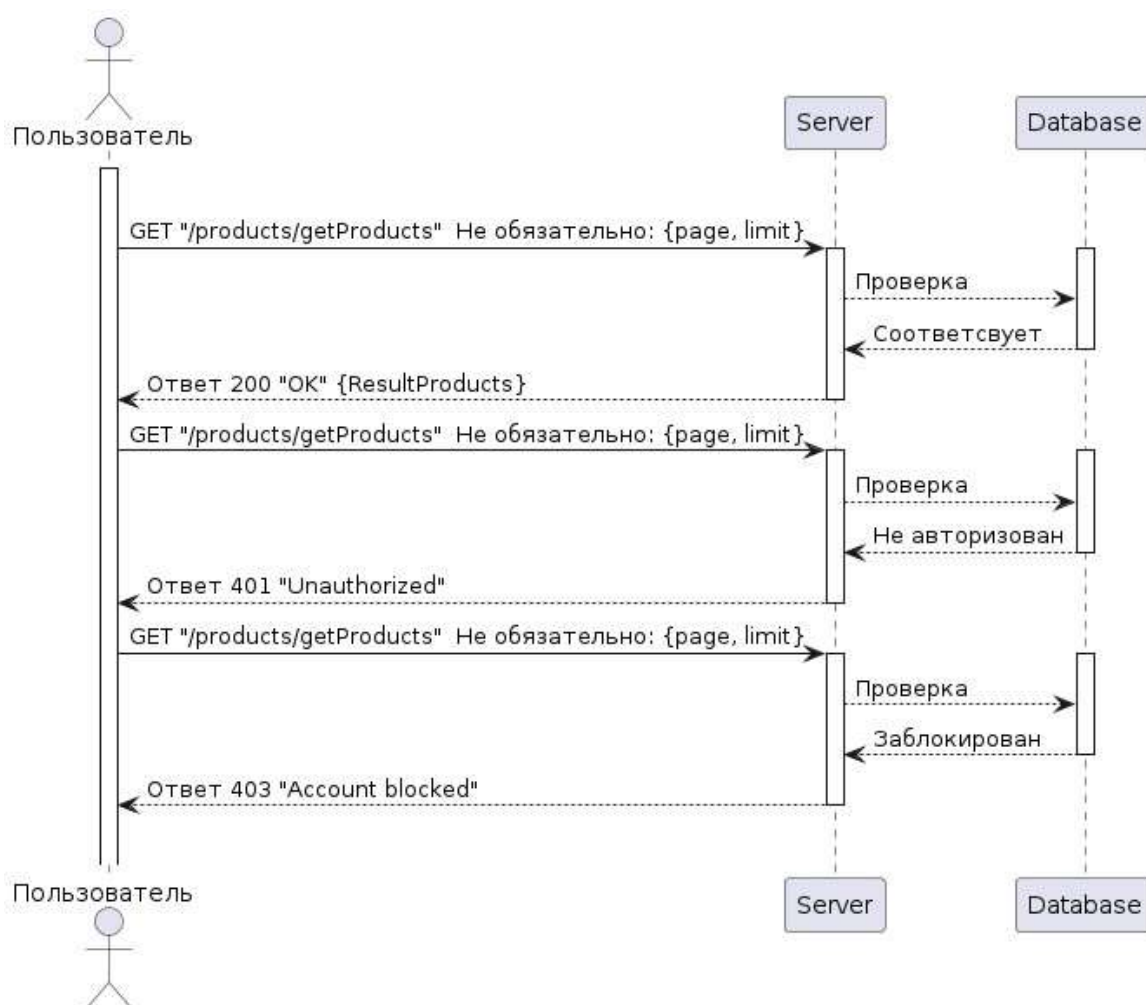


Рисунок 6 – Диаграмма последовательности прецедента «Получить список товаров»

2.1.4 Прецедент «Получить список цен товаров»

Запрос GET для получения списка цен товаров делается так же зарегистрированным пользователем интернет-магазина. Этот запрос отправляется вместе с запросом «Получить список товаров». При успешном запросе, сервер вернет ответ 200, и отобразит у каждого товара его цену. В

случае если пользователь не авторизован, либо его аккаунт заблокирован, сервер вернет ошибку 401 и 403 соответственно.

На рисунке 7 представлена диаграмма последовательности для этого прецедента.

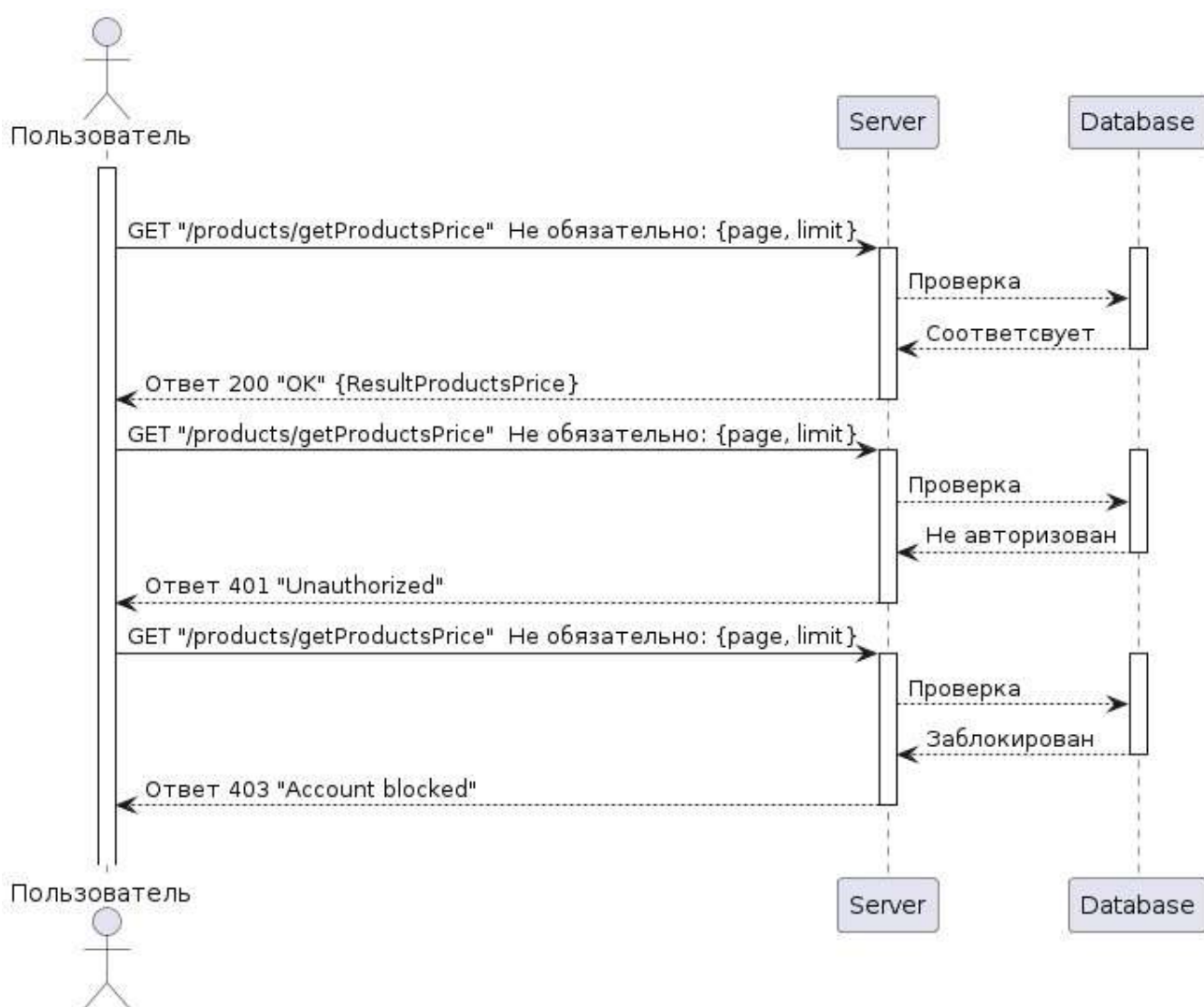


Рисунок 7 – Диаграмма последовательности прецедента «Получить список цен товаров»

2.1.5 Прецедент «Получение информация о товаре»

Если пользователь нажал на товар, на сервер отправляется GET запрос содержащий уникальный идентификационный номер товара. В случае успешного запроса, сервер возвращает пользователю информацию о товаре (название модели, другие картинки товара, размеры, описание и категория).

Если запрос не успешен, как и в предыдущих, сервер уведомит об ошибке 401 или 403.

На рисунке 8 показана диаграмма последовательностей для прецедента «Получение информация о товаре»

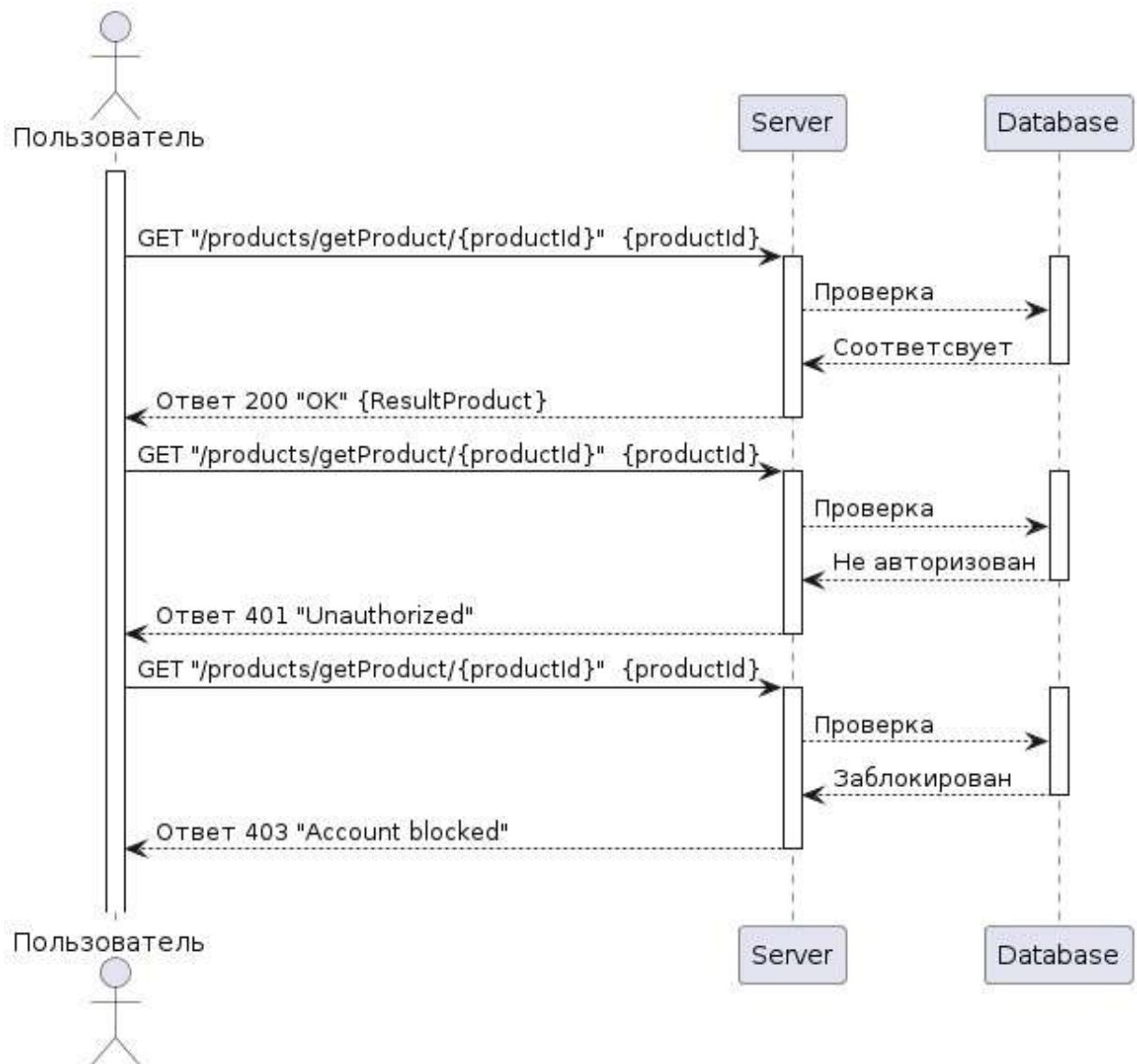


Рисунок 8 – Диаграмма последовательностей прецедента «Получение информация о товаре»

2.1.6 Прецедент «Получение информации о размерах товара»

Если пользователь нажал на размер товара, на сервер отправляется GET запрос содержащий уникальный идентификационный номер товара. В случае успешного запроса, сервер возвращает пользователю информацию о размерах

товара (сколько товаров данного размера осталось). Если запрос не успешен, сервер уведомит об ошибке 401 или 403.

На рисунке 9 представлена диаграмма последовательности прецедента «Получение информации о размерах товара»

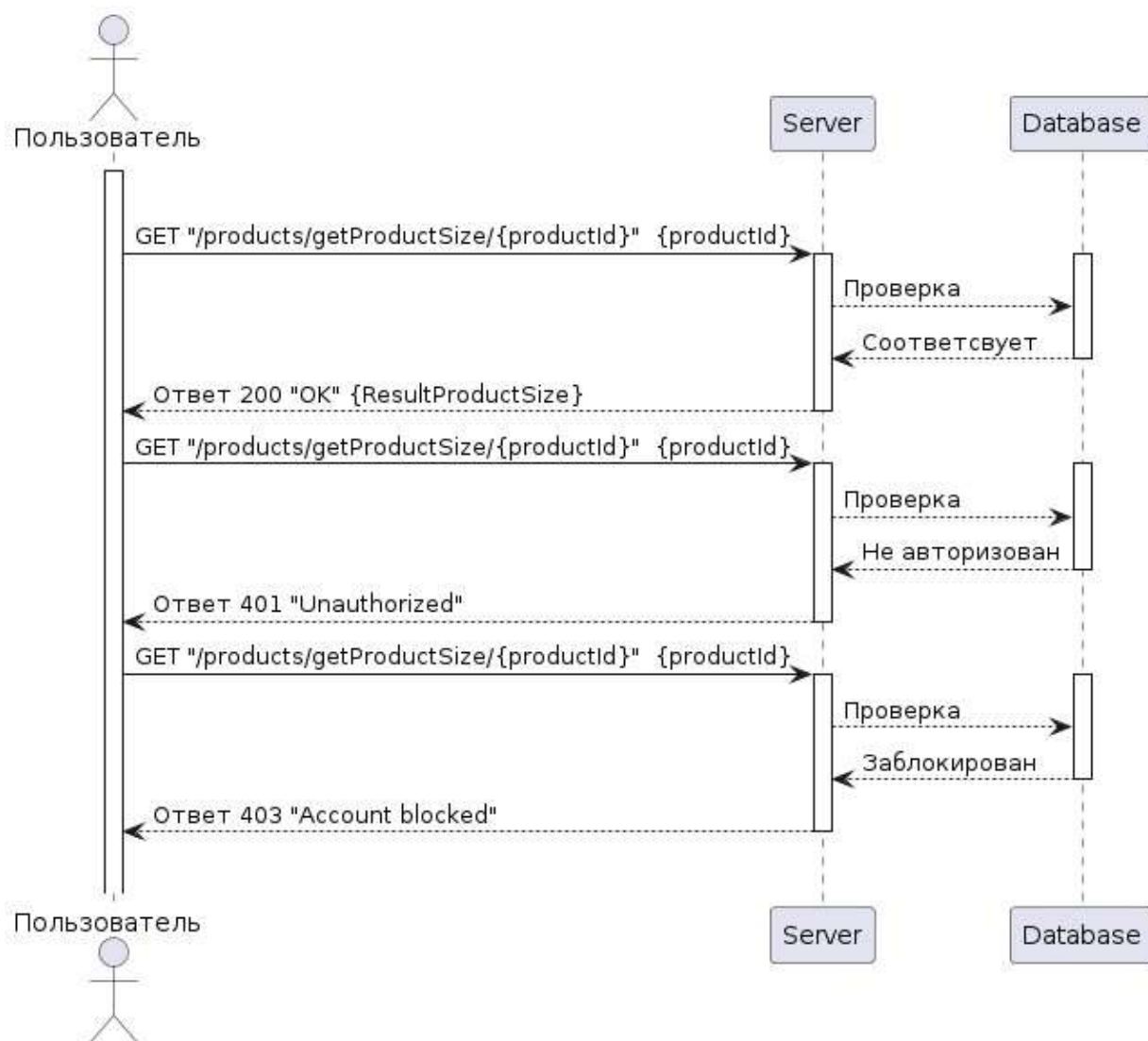


Рисунок 9 – Диаграмма последовательности прецедента «Получение информации о размерах товара»

2.2 Структурная схема системы



Рисунок 10 – Структурная схема системы

На рисунке 10 мы видим структурную схему системы. Первый блок представляет базу данных, которая является основным хранилищем информации для всей системы. Он взаимосвязан с двумя другими блоками: модулем обработки заказов и административной панелью. Эти связи представлены двунаправленными стрелками, что указывает на взаимное влияние и обмен данными между этими компонентами. Модуль обработки заказов, связан с блоком API, который служит для взаимодействия с внешними системами и сервисами. Это важное звено для обеспечения работы системы, так как через API осуществляется обмен данными и выполнение различных операций, связанных с заказами. Блок API связан с блоком сайт дропшиппера, который представляет из себя интернет-магазин, реализующий дропшиппинг.

Так же на схеме мы видим, что клиент может взаимодействовать с сайтом дропшиппинга, собственно для того чтоб просматривать товары, и заказывать их себе. Мы видим, что администратор имеет доступ к администраторской панели, откуда он может управлять клиентами, а также обрабатывать заказы. Только администратор имеет доступ к управлению дропшипперами. Он может как зарегистрировать и добавить новых дропшипперов, давая им доступ к данным, так и удалить доступ и статус дропшиппера.

2.3 Вывод по главе

В данной главе были построены диаграммы последовательности для таких прецедентов как:

- запрос токена;
- расчёт стоимости доставки;
- получить список товаров;
- получить список цен товаров;
- получение информация о товаре;
- получение информации о размерах товара.

Так же была построена структурная схема.

3 Разработка методов API

3.1 Swagger

3.1.1 Метод для аутентификации и получения токена

Этот метод предназначен для аутентификации пользователей и получения токена доступа, который используется в дальнейшем для доступа к другим элементам системы.

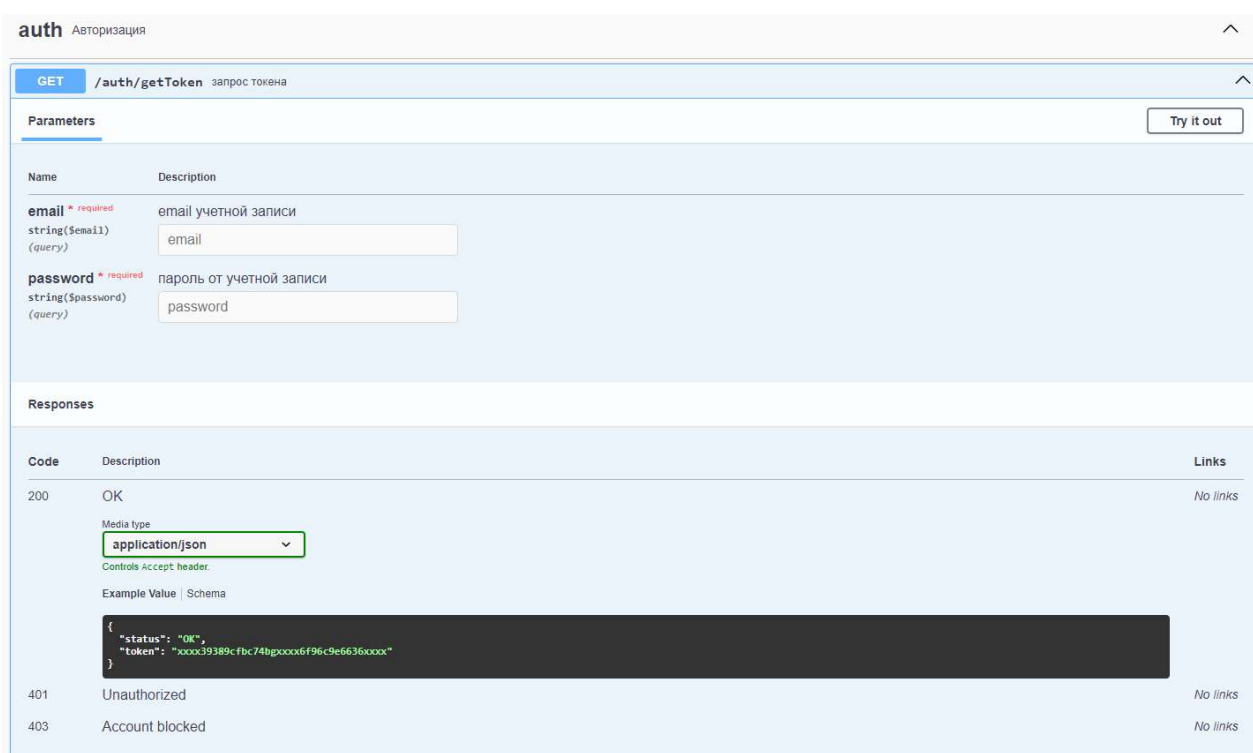


Рисунок 11 – Аутентификации и получения токена

На рисунке 11 мы видим, что используемый метод– GET, а конечная точка «/auth/getToken». Необходимые параметры — это адрес электронной почты, привязанный к учетной записи и пароль от этой учетной записи. Ответом на запрос могут быть:

- «200 OK» – успешный запрос. Токен доступа предоставлен в формате JSON, пример токена как на изображении:


```

    "description": "Unauthorized"
  },
  "403": {
    "description": "Account blocked"
  }
}
}
}

```

3.1.2 Метод для расчёта стоимости доставки

Этот метод нужен, чтобы рассчитать стоимость доставки исходя из получаемых данных.

The screenshot shows a REST client interface for a service named 'delivery' (Доставка). The endpoint is `POST /delivery/getDeliveryCost` with the description 'расчет стоимости доставки'. The 'Parameters' section lists the following fields:

Name	Description
<code>cityName</code> * required string (query)	Город, нас. пункт
<code>zip</code> * required integer (query)	Индекс нас. пункта
<code>kladrCode</code> integer (query)	Код КЛАДР нас. пункта
<code>fiasCode</code> string (query)	Код ФИАС нас. пункта
<code>okatoCode</code> integer (query)	Код ОКАТО нас. пункта

The 'Request body' section is set to `application/json` and shows the following JSON example:

```

{
  "addProducts": [
    {
      "sizeId": 117641,
      "qty": 2
    },
    {
      "sizeId": 17641,
      "qty": 1
    }
  ],
  "addListVZ": false,
  "checkVail": false
}

```

Рисунок 12 – Параметры для запроса расчета стоимости доставки

На рисунке 12, можно увидеть, что это метод POST, с конечной точкой «`delivery/getDeliveryCost`». Необходимые параметры:

- Город куда доставлять,
- индекс населенного пункта.

И дополнительные параметры, которые можно указать:

- код КЛАДР населенного пункта,
- код ФИАС,
- код ОКАТО.

Тело запроса содержит информацию о товаре и его наличии. Пример запроса как в изображении:

```
{
  "addProducts": [
    {
      "sizeId": 117641,
      "qty": 2
    },
    {
      "sizeId": 17641,
      "qty": 1
    }
  ],
  "addListPVZ": false,
  "checkAvail": false
}
```

Видим «sizeID», который передает идентификатор размера обуви, «qty» количество обуви, «addListPVZ» добавляет к выводу список пунктов выдачи заказов и «checkAvail» содержит информацию о наличии модели.

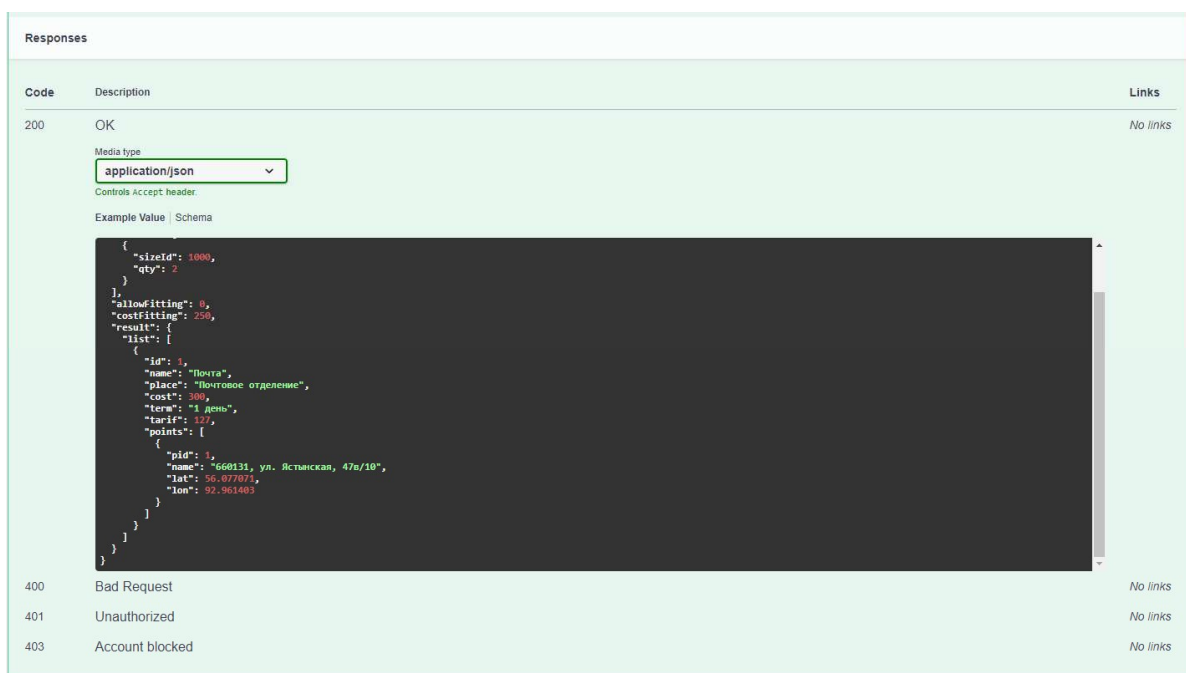


Рисунок 13 – Ответ на запрос расчета стоимости доставки

На рисунке 13 видны ответы сервера на запросы:

- «200 OK» - Успешный запрос. Возвращается информация о стоимости доставки. Пример формата ответа как на рисунке:

```
{
  "status": "OK",
  "req_time": 0,
}
```

```
    },
    "401": {
      "description": "Unauthorized"
    },
    "403": {
      "description": "Account blocked"
    }
  },
  "security": [
    {
      "bearerAuth": []
    }
  ]
},
},
```

3.1.3 Метод для получения списка товаров

Данный метод предназначен для получения списка товаров.

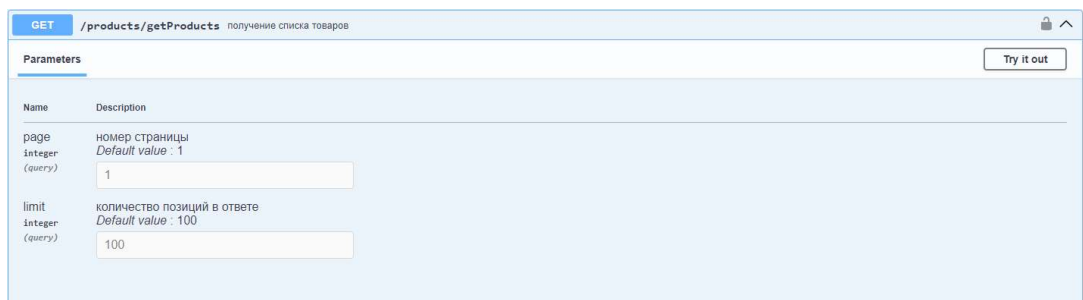


Рисунок 14 – Параметры для запроса получения списка товаров

На рисунке 14 видно, что это метод GET, с конечной точкой в «/products/getProducts». Параметры для данного метода: «page» - это параметр запроса, который определяет номер страницы списка товаров. Он необязательный и имеет значение по умолчанию 1. «limit» - это также параметр запроса, который указывает количество позиций (товаров) в ответе. Он также необязателен и имеет значение по умолчанию - 100.

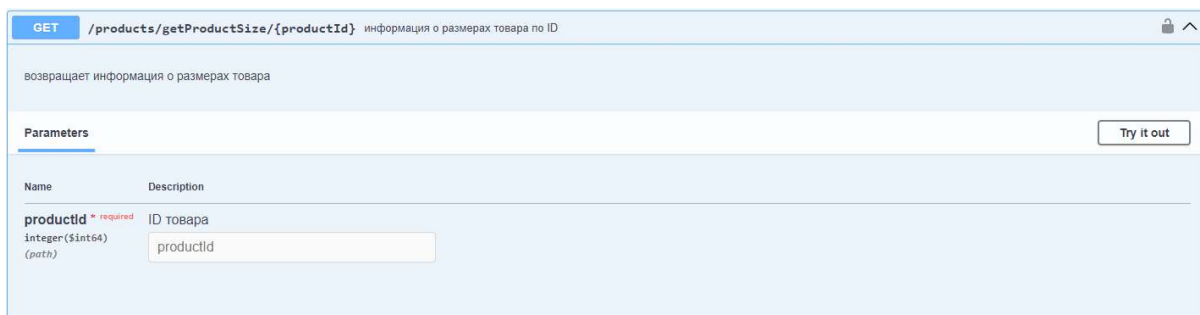


Рисунок 20 – Параметр для запроса получения информации о размерах товара

На рисунке 20 можно увидеть, что используемый метод GET, а конечная точка «/products/getProductSize/{productId}». Необходимый параметр: «productId» - параметр, который указывает на ID товара, информацию о размерах которого нужно получить.

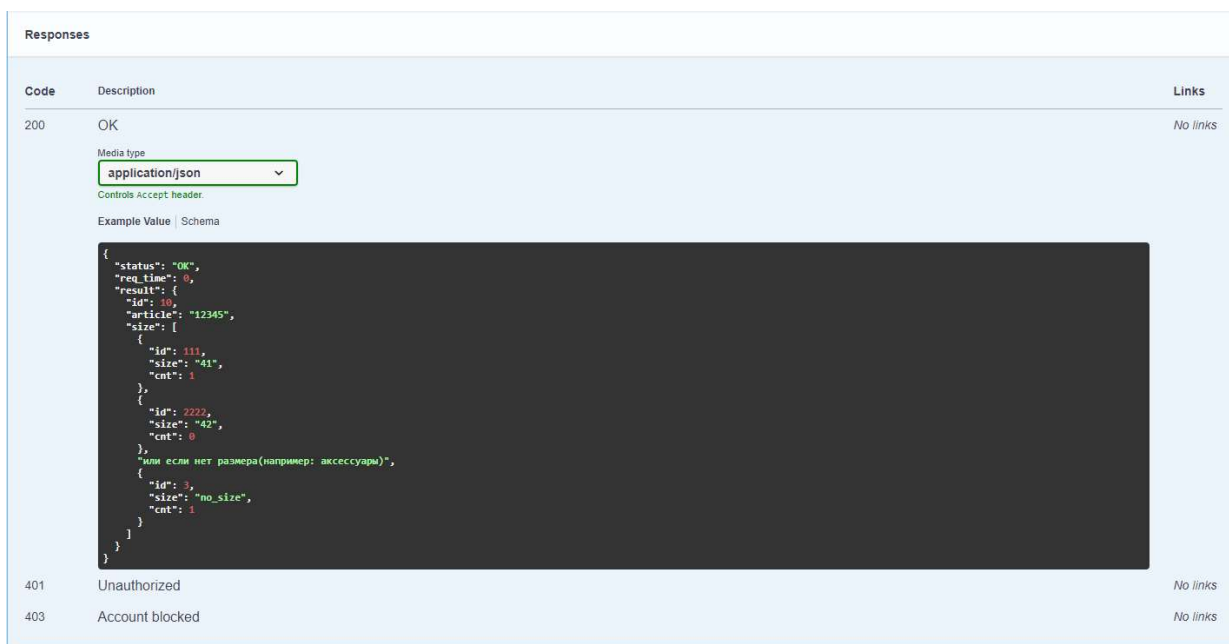


Рисунок 21 – Ответ на запрос получения информации о размерах товара

На рисунке 21 мы видим возможные ответы сервера:

- «200 OK» - в случае успешного выполнения запроса, возвращается статус "OK" и содержимое ответа представлено в формате JSON. Пример ответа из изображения:

Когда сервер получает запрос с заголовком Authorization, содержащим токен доступа, он проверяет этот токен для подтверждения прав доступа пользователя к запрашиваемому ресурсу или операции. Если токен действителен и предоставленные права доступа позволяют выполнить операцию, сервер обрабатывает запрос; в противном случае он возвращает ошибку доступа.

Таким образом, механизм bearerAuth обеспечивает безопасный способ защиты ресурсов сервера, требующих аутентификации и авторизации для доступа к ним.

3.3 РНР

С использованием РНР были написаны тесты для проверки работоспособности API запросов. На рисунках 22, 23 и 24 представлен код.

```
1 <?php
2
3 require 'vendor/autoload.php';
4
5 use GuzzleHttpClient;
6 use GuzzleHttpException\RequestException;
7
8 class ApiTest {
9     private $client;
10    private $token;
11
12    public function __construct() {
13        $this->client = new Client([
14            'base_uri' => 'https://api.fireboxclub.com',
15            'timeout' => 2.0,
16        ]);
17    }
18
19    private function authenticate($email, $password) {
20        try {
21            // Выполняем GET-запрос для получения токена
22            $response = $this->client->get('/auth/getToken', [
23                'query' => [
24                    'email' => $email,
25                    'password' => $password,
26                ]
27            ]);
28            $this->token = json_decode($response->getBody(), true)['token'];
29        } catch (RequestException $e) {
30            // Обработка ошибок, если запрос не удался
31            echo $e->getResponse() ? $e->getResponse()->getBody()->getContents() : $e->getMessage();
32        }
33    }
34
35    public function getDeliveryCost($cityName, $zip, $kladrCode = null, $fiasCode = null, $okatoCode = null, $body) {
36        try {
37            $response = $this->client->post('/delivery/getDeliveryCost', [
38                'query' => [
39                    'cityName' => $cityName,
40                    'zip' => $zip,
41                    'kladrCode' => $kladrCode,
42                    'fiasCode' => $fiasCode,
43                    'okatoCode' => $okatoCode,
44                ],
45                'json' => $body,
46                'headers' => [
47                    'Authorization' => 'Bearer ' . $this->token, // Используем токен для авторизации
48                ]
49            ]);
50            return json_decode($response->getBody(), true);
51        }
52    }
53 }
```

Рисунок 22 – Скрипт РНР для аутентификации и получения стоимости доставки

```

51     } catch (RequestException $e) {
52         return $e->getResponse() ? $e->getResponse()->getBody()->getContents() : $e->getMessage();
53     }
54 }
55
56 public function getProducts($page = 1, $limit = 100) {
57     try {
58         $response = $this->client->get('/products/getProducts', [
59             'query' => [
60                 'page' => $page,
61                 'limit' => $limit,
62             ],
63             'headers' => [
64                 'Authorization' => 'Bearer ' . $this->token, // Используем токен для авторизации
65             ]
66         ]);
67         return json_decode($response->getBody(), true);
68     } catch (RequestException $e) {
69         return $e->getResponse() ? $e->getResponse()->getBody()->getContents() : $e->getMessage();
70     }
71 }
72
73 public function getProductsPrice($page = 1, $limit = 100) {
74     try {
75         $response = $this->client->get('/products/getProductsPrice', [
76             'query' => [
77                 'page' => $page,
78                 'limit' => $limit,
79             ],
80             'headers' => [
81                 'Authorization' => 'Bearer ' . $this->token, // Используем токен для авторизации
82             ]
83         ]);
84         return json_decode($response->getBody(), true);
85     } catch (RequestException $e) {
86         return $e->getResponse() ? $e->getResponse()->getBody()->getContents() : $e->getMessage();
87     }
88 }
89
90 public function getProduct($productId) {
91     try {
92         $response = $this->client->get('/products/getProduct/' . $productId, [
93             'headers' => [
94                 'Authorization' => 'Bearer ' . $this->token, // Используем токен для авторизации
95             ]
96         ]);
97         return json_decode($response->getBody(), true);

```

Рисунок 23 – Скрипт PHP для получения разной информации о продуктах

```

98     } catch (RequestException $e) {
99         return $e->getResponse() ? $e->getResponse()->getBody()->getContents() : $e->getMessage();
100     }
101 }
102
103 public function getProductSize($productId) {
104     try {
105         $response = $this->client->get('/products/getProductSize/' . $productId, [
106             'headers' => [
107                 'Authorization' => 'Bearer ' . $this->token, // Используем токен для авторизации
108             ]
109         ]);
110         return json_decode($response->getBody(), true);
111     } catch (RequestException $e) {
112         return $e->getResponse() ? $e->getResponse()->getBody()->getContents() : $e->getMessage();
113     }
114 }
115 }
116
117 $apiTest = new ApiTest();
118 $email = 'Leonid2002.lk@gmail.com';
119 $password = 'MyPass1s1';
120
121 $apiTest->authenticate($email, $password);
122
123 $deliveryCostResponse = $apiTest->getDeliveryCost('Красноярск', 660001, null, null, null, ['weight' => 5]);
124 echo "Стоимость доставки: ";
125 print_r($deliveryCostResponse);
126
127 $productsResponse = $apiTest->getProducts();
128 echo "Список товаров: ";
129 print_r($productsResponse);
130
131 $productsPriceResponse = $apiTest->getProductsPrice();
132 echo "Список цен товаров: ";
133 print_r($productsPriceResponse);
134
135 $productResponse = $apiTest->getProduct(1);
136 echo "Информация о товаре: ";
137 print_r($productResponse);
138
139 $productSizeResponse = $apiTest->getProductSize(1);
140 echo "Информация о размерах товара: ";
141 print_r($productSizeResponse);
142
143 >>

```

Рисунок 24 – Скрипт PHP для запуска тестов

На рисунке 22 функция «authenticate» выполняется GET-запрос для получения токена. Параметры «email» и «password» передаются в запрос. Ответ декодируется из формата JSON, и токен сохраняется в «\$token». Если запрос не удался, выводится сообщение об ошибке.

В функции «getDeliveryCost» выполняется проверка одноименного метода. Метод «getDeliveryCost» выполняет POST-запрос для получения стоимости доставки. Параметры запроса: «cityName», «zip», «kladrCode», «fiasCode», «okatoCode» передаются в запрос как параметры «query». Параметры, такие как вес посылки, передаются в теле запроса. «Authorization» использует токен, полученный при аутентификации, чтобы предоставить доступ к данным. Ответ декодируется из JSON и возвращается. Если запрос не удался, так же выводится сообщение об ошибке.

На рисунке 23 функция «getProducts» выполняет проверку метода «getProducts». Это GET-запрос для получения списка товаров. Параметры «page» и «limit» задают пагинацию результатов. «Authorization» как и в предыдущем использует токен, полученный при аутентификации. Ответ декодируется из JSON и возвращается. В случае если запрос не удался, выводится сообщение об ошибке.

Функция «getProductPrice» выполняет проверку аналогичного предыдущему метода, но запрашивает список цен товаров.

Метод «getProduct» проверяется в одноименной функции. Выполняется GET-запрос для получения информации о товаре по его уникальному идентификационному номеру. «productId» передается в URL. «Authorization» использует токен, полученный при аутентификации, для доступа к данным. Ответ декодируется из формата JSON и возвращается. Если запрос не удался, выводится сообщение об ошибке.

На рисунке 24 функция «getProductSize» выполняет проверку аналогичного предыдущему метода, но вместо общей информации о товаре запрашивается информация о доступных размерах товара.

В последнем блоке скрипта, показано как использовать класс «ApiTest»:

- создается экземпляр класса;
- выполняется аутентификация с использованием электронной почты и пароля;
- выполняются запросы к различным конечным точкам API с помощью методов класса.

3.4 Вывод по главе

В этой главе я разработал на Swagger API для интернет-магазина. Так же на PHP были написаны тесты для всех конечных точек API.

ЗАКЛЮЧЕНИЕ

Первая глава послужила введением в проект, был проведен анализ сопоставимых сайтов, имеющих функцию дропшипинга. Выявив сильные и слабые стороны существующих интернет-магазинов, были установлены необходимые требования для собственного проекта. Выбраны инструменты и технологии, необходимые для разработки API для интернет-магазина, включая PHP и Swagger. Эта глава заложила основу для последующих этапов разработки.

Вторая глава посвящена этапу проектирования API интернет-магазина. Созданию диаграмм последовательности и структурной схемы для системы. Были выявлены все конечные точки которые должны присутствовать в будущей системе. Так же были рассмотрены все варианты ответа сервера на разные ситуации.

Третья глава сместила акцент на реализацию API и проведении тестов.

В целом, в результате выполнения выпускной квалификационной работы я охватил основные аспекты разработки API для интернет-магазина, от анализа и выявления требований до проектирования и реализации, а также тестирования.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Интернет-магазин «Nazuа»: официальный сайт. – Красноярск, 2010. – URL: <https://nazuа.com/> (дата обращения: 29.04.2024).
2. Интернет-магазин «Outmaxshop»: официальный сайт. – Красноярск, 2012. – URL: <https://outmaxshop.ru/> (дата обращения: 29.04.2024).
3. Интернет-магазин «Markethot»: официальный сайт. – Москва, 2013. – URL: <https://markethot.ru/> (дата обращения: 29.04.2024).
4. Диаграммы PlantUML: редакция от 14 июня 2023 // Хабр : сайт-статейник. – URL: <https://habr.com/ru/companies/alfa/articles/740518/> (дата обращения: 30.04.2024).
5. Draw.io – онлайн приложение для рисования диаграмм [Электронный ресурс] – URL: <https://app.diagrams.net/> (дата обращения: 30.04.2024).
6. Php – Скриптовый язык общего назначения : редакция от 15 декабря 2021 // ХАБР : сайт-статейник. – URL: <https://habr.com/ru/hub/php> (дата обращения: 01.05.2024).
7. PlantUML – моделирование [Электронный ресурс] – URL: <http://www.plantuml.com/plantuml/uml/SyfFKj2rKt3CoKnELR1Io4ZDoSa70000> (дата обращения: 30.04.2024).
8. Swagger – набор инструментов для разработчиков API [Электронный ресурс] – URL: <https://www.amocrm.ru/> (дата обращения: 10.05.2024).
9. Что такое Swagger: редакция от 27 октября 2023 // Яндекс практикум: сайт-статейник. – URL: <https://practicum.yandex.ru/blog/что-такое-swagger/> (дата обращения: 05.05.2024).
10. ChatGPT - чат-бот с генеративным искусственным интеллектом [Электронный ресурс] – URL: <https://chatgpt.com/> (дата обращения: 10.05.2024)

Министерство науки и высшего образования РФ
Федеральное государственное автономное
образовательное учреждение высшего образования
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт космических и информационных технологий



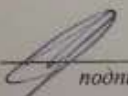
Кафедра вычислительной техники

УТВЕРЖДАЮ
Заведующий кафедрой
О.В. Непомнящий
«17» 06 2024 г.

БАКАЛАВРСКАЯ РАБОТА

090301 Информатика и вычислительная техника

Разработка системы дропшипинга для интернет-магазина

Руководитель	 подпись	10.06.24 дата	доцент, канд. техн. наук должность, ученая степень	М.С. Медведев
Выпускник	 подпись	17.06.24 дата		Л.В. Колесник
Нормоконтролёр	 подпись	10.06.24 дата	доцент, канд. техн. наук должность, ученая степень	М.С. Медведев

Красноярск 2024