

Министерство образования Республики Беларусь  
Учреждение образования

«БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ  
ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ»

Факультет компьютерных систем и сетей

Кафедра электронных вычислительных машин

Дисциплина: Жизненный цикл разработки программного обеспечения

Лабораторная работа  
ЖИЗНЕННЫЙ ЦИКЛ РАЗРАБОТКИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

Выполнил:

Сапега Г.В.

Проверил:

Внук О.М.

МИНСК 2025

## СОДЕРЖАНИЕ

|   |    |
|---|----|
| ВВЕДЕНИЕ.....   | 3  |
| 1 ТРЕБОВАНИЯ ПОЛЬЗОВАТЕЛЯ.....                            | 4  |
| 1.1 Программные интерфейсы.....                           | 4  |
| 1.2 Интерфейс пользователя .....                          | 4  |
| 1.3 Характеристики пользователей .....                    | 5  |
| 1.4 Предположения и зависимости .....                     | 5  |
| 2 СИСТЕМНЫЕ ТРЕБОВАНИЯ.....                               | 6  |
| 2.1 Функциональные требования .....                       | 6  |
| 2.2 Нефункциональные требования .....                     | 6  |
| 3 ГЛОССАРИЙ.....  | 7  |
| 4 ДИАГРАММА КЛАССОВ.....                                  | 8  |
| 5 АРХИТЕКТУРА ВЗАИМОДЕЙСТВИЯ С СИСТЕМОЙ .....             | 9  |
| 5.1 Диаграмма вариантов использования .....               | 9  |
| 5.2 Диаграммы активностей.....                            | 10 |
| 5.3 Диаграммы состояний .....                             | 11 |
| 5.4 Диаграммы последовательности .....                    | 13 |
| 5.5 Диаграмма компонентов .....                           | 14 |
| 5.6 Диаграмма развертывания .....                         | 14 |
| 6 ТЕСТИРОВАНИЕ УЧЕБНОГО ПРОЕКТА .....                     | 16 |
| 6.1 Введение (Introduction).....                          | 16 |
| 6.2 Объект тестирования (Test Items).....                 | 16 |
| 6.3 Риски (Risk Issues).....                              | 16 |
| 6.4 Аспекты тестирования (Features to be Tested).....     | 17 |
| 6.5 Подходы к тестированию (Test Approach) .....          | 17 |
| 6.6 Представление результатов (Pass / Fail Criteria)..... | 17 |
| 6.7 Тестовые сценарии (Test Cases).....                   | 18 |
| ЗАКЛЮЧЕНИЕ .....  | 19 |

## ВВЕДЕНИЕ

В условиях повсеместной цифровизации и роста популярности онлайн-шопинга существует устойчивая рыночная потребность в надежных, масштабируемых и удобных платформах для электронной коммерции. Название проекта «JavaInternetShop». Данный электронный магазин реализует продажу комплектующих для персонального компьютера

Назначение системы — создание надежной, масштабируемой и удобной платформы для электронной коммерции, ориентированной на продажу комплектующих для персональных компьютеров. Система позволяет пользователям просматривать товары, добавлять их в корзину, оформлять заказы, авторизовываться и искать продукты. Для администраторов предусмотрена панель управления.

Бизнес-цели:

- Удовлетворение рыночного спроса на онлайн-покупки комплектующих для ПК в условиях цифровизации.
- Обеспечение удобства для пользователей возрастом 16-40 лет из крупных городов, интересующихся сборкой ПК.
- Масштабируемость для роста ассортимента и посещаемости.

# 1 ТРЕБОВАНИЯ ПОЛЬЗОВАТЕЛЯ

## 1.1 Программные интерфейсы

Для разработки серверной части приложения используется Java Spring Boot, для клиентской – JavaScript, фреймворк React, с использованием библиотеки MaterialUI. Взаимодействие осуществляется через REST API, с авторизацией на базе JWT. База данных – реляционная СУБД MySQL.

## 1.2 Интерфейс пользователя

Mock-up сайта представлен на рисунке 1.1

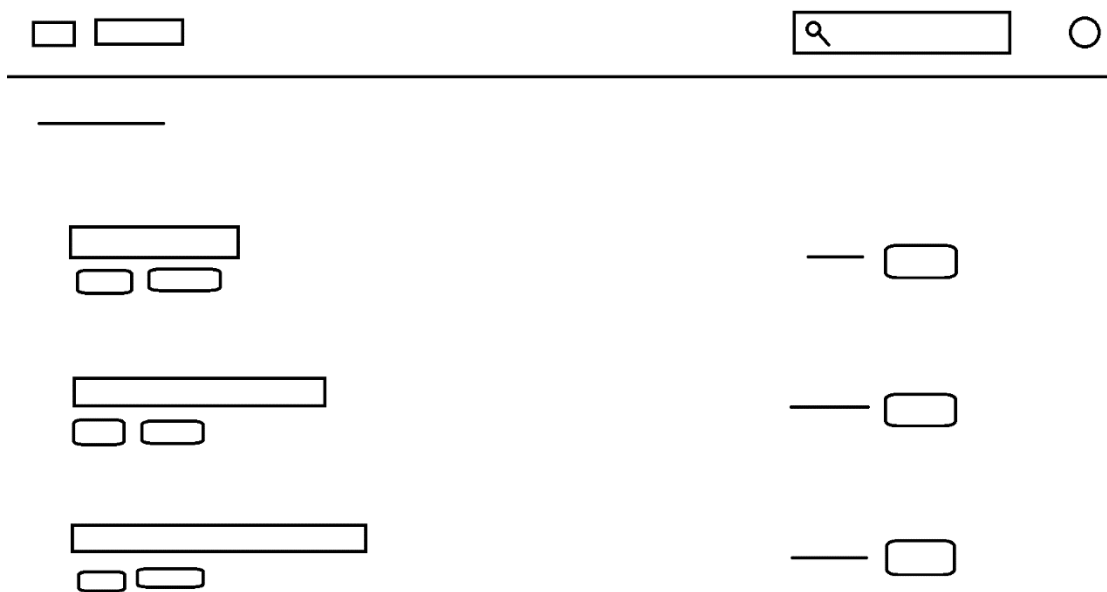


Рисунок 1.1 – Mock-up сайта

Mock-up страницы авторизации представлен на рисунке 1.2

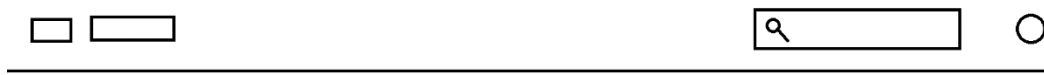


Рисунок 1.2 – Mock-up страницы авторизации.

### 1.3 Характеристики пользователей

Целевая аудитория: Пользователи из густонаселенных городов, возрастом 16-40 лет, интересующиеся ручной сборкой персонального компьютера. Уровень: Средний (знание базовых IT-терминов), ожидают удобного интерфейса.

Администраторы: Персонал магазина (менеджеры, модераторы), с доступом к панели для управления контентом.

### 1.4 Предположения и зависимости

- Возраст аудитории: 16–40 лет;
- Перечень категорий: процессоры, видеокарты, материнские платы, память, накопители и т.д;
- Зависимости: доступ к интернету, браузер (Chrome, Firefox), интеграция с платежными системами;
- Ограничения: нет мобильной версии на старте, фокус на веб.

## **2 СИСТЕМНЫЕ ТРЕБОВАНИЯ**

### **2.1 Функциональные требования**

- Возможность просматривать характеристики товара;
- Возможность добавления товара в корзину;
- Возможность оформления заказа;
- Авторизация;
- Масштабируемость количества категорий товаров (базы данных);
- Поиск товаров по имени.

### **2.2 Нефункциональные требования**

#### **2.2.1 Атрибуты качества:**

- Продвинутая авторизация (с использованием JWT);
- Базовая безопасность приложения (защита от SQL-инъекций, XSS);
- Админ-панель для различных категорий персонала.

Дополнительно:

- Производительность;
- Масштабируемость;
- Надежность;
- Удобство: Интуитивный UI на базе MaterialUI.

### 3 ГЛОССАРИЙ

- Товар (Product): элемент ассортимента, с характеристиками (название, цена, описание).
- Корзина (Cart): временное хранилище выбранных товаров пользователя.
- Заказ (Order): финализированная покупка с деталями доставки.
- Пользователь (User): зарегистрированный клиент, с ролями (гость, авторизованный, администратор, продавец).
- Категория (Category): группа товаров (например, "Процессоры").
- Авторизация (Authentication): процесс входа с использованием JWT.
- Поиск (Search): функция фильтрации товаров по имени и параметрам.
- Админ-панель (Admin Panel): интерфейс для управления системой.

## 4 ДИАГРАММА КЛАССОВ

Упрощенная диаграмма классов представлена на рисунке 4.1.

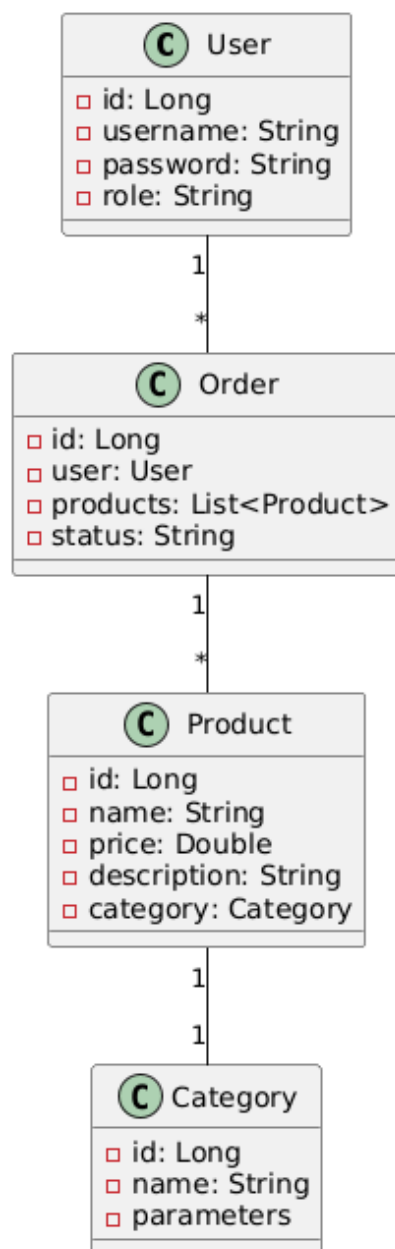


Рисунок 4.1 – Диаграмма классов.



## **5 АРХИТЕКТУРА ВЗАИМОДЕЙСТВИЯ С СИСТЕМОЙ**

### **5.1 Диаграмма вариантов использования**

Акторы: Гость, Авторизованный пользователь, Продавец, Доставщик, Администратор.

Варианты использования:

#### **5.1.1 Гость:**

- получить главную страницу;
- искать товары;
- просматривать информацию о товарах;
- регистрироваться или авторизовываться.

#### **5.1.2 Авторизованный пользователь:**

- просматривать свой личный кабинет;
- писать отзывы на товары;
- добавлять товары в корзину;
- оформлять заказы;
- просматривать свои заказы.

#### **5.1.3 Продавец:**

- добавлять товары;
- изменять информацию о товаре;
- удалять товары;
- просматривать заказы.

#### **5.1.4 Доставщик:**

- просматривать заказы;
- изменять состояние заказов.

#### **5.1.5 Администратор:**

- просматривать список пользователей;
- менять права пользователей;

Диаграмма вариантов использования представлена на рисунке 5.1

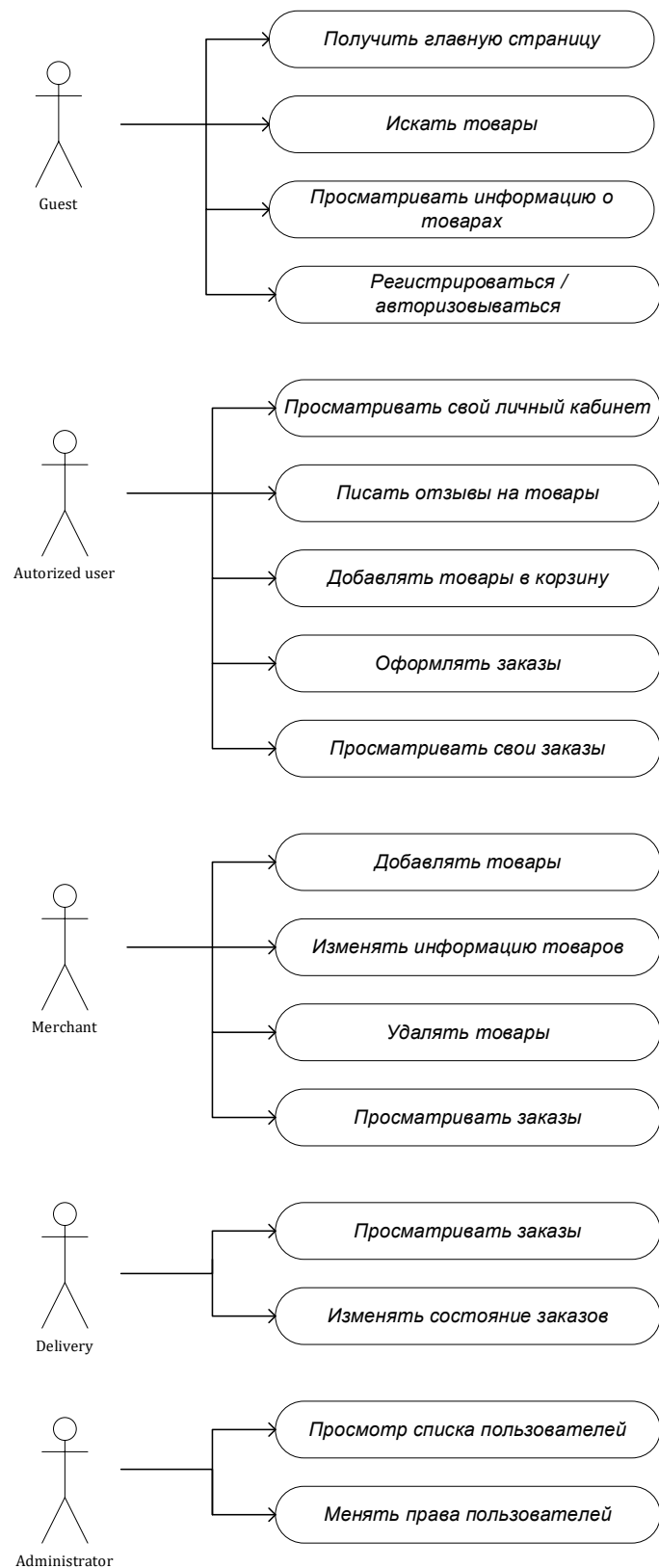


Рисунок 5.1 – Диаграмма вариантов использования.

## 5.2 Диаграммы активностей

**5.2.1 Оформление заказа.** Данная диаграмма активности представлена на рисунке 5.2.



Рисунок 5.2 – Оформление заказа

**5.2.2 Поиск товаров.** Данная диаграмма активности представлена на рисунке 5.3.



Рисунок 5.3 – Поиск товаров

### 5.3 Диаграммы состояний

**5.3.1 Корзина.** Диаграмма состояний корзины представлена на рисунке 5.4.

**5.3.2 Поиск товаров.** Диаграмма состояния поиска товаров представлена на рисунке 5.5.



Рисунок 5.4 – Корзина

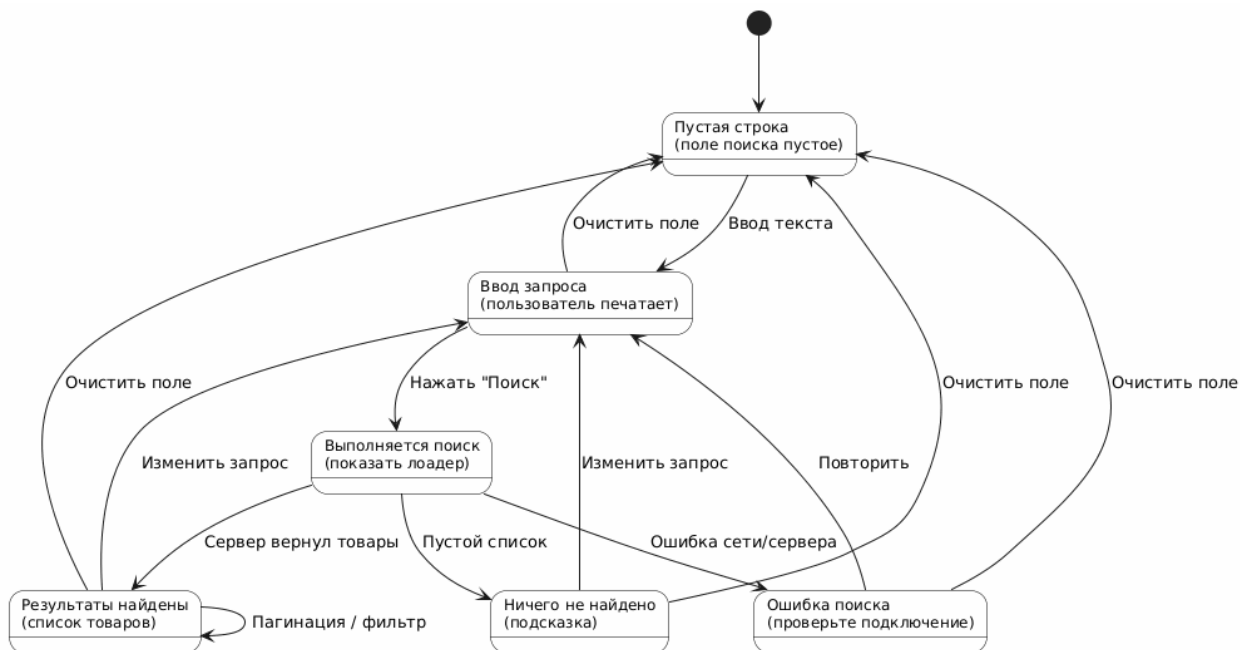


Рисунок 5.5 – Поиск

## 5.4 Диаграммы последовательности

**5.4.1 Оформление заказа.** Диаграмма последовательности оформления заказа представлена на рисунке 5.6.

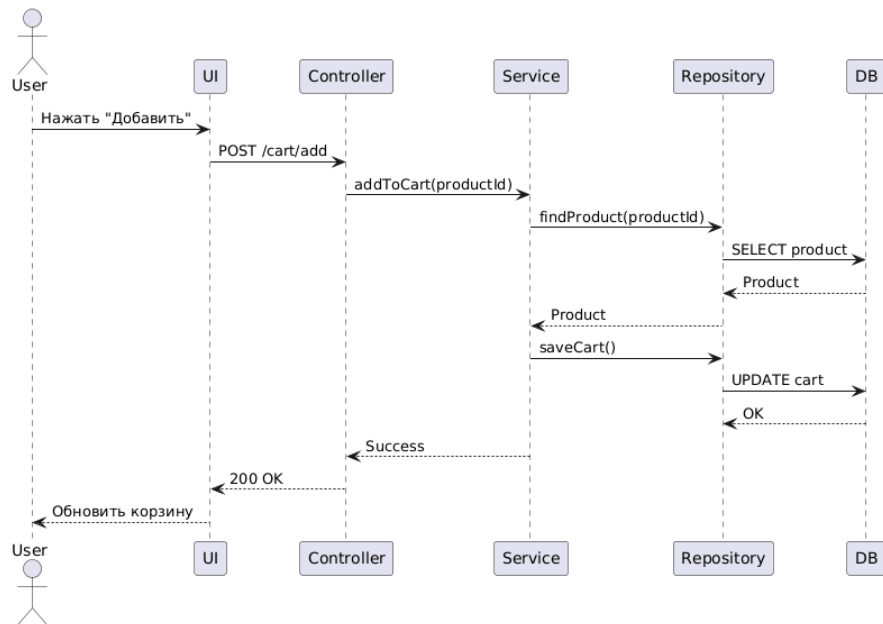


Рисунок 5.6 – Оформление заказа.

**5.4.2 Авторизация.** Диаграмма последовательности авторизации представлена на рисунке 5.7

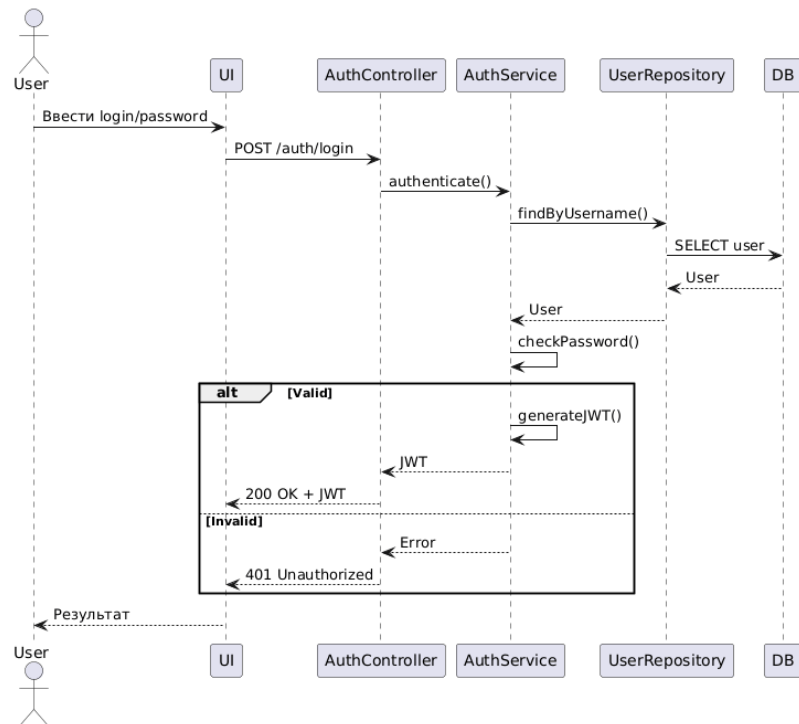


Рисунок 5.7 – Авторизация.

## 5.5 Диаграмма компонентов

Диаграмма компонентов представлена на рисунке 5.8.

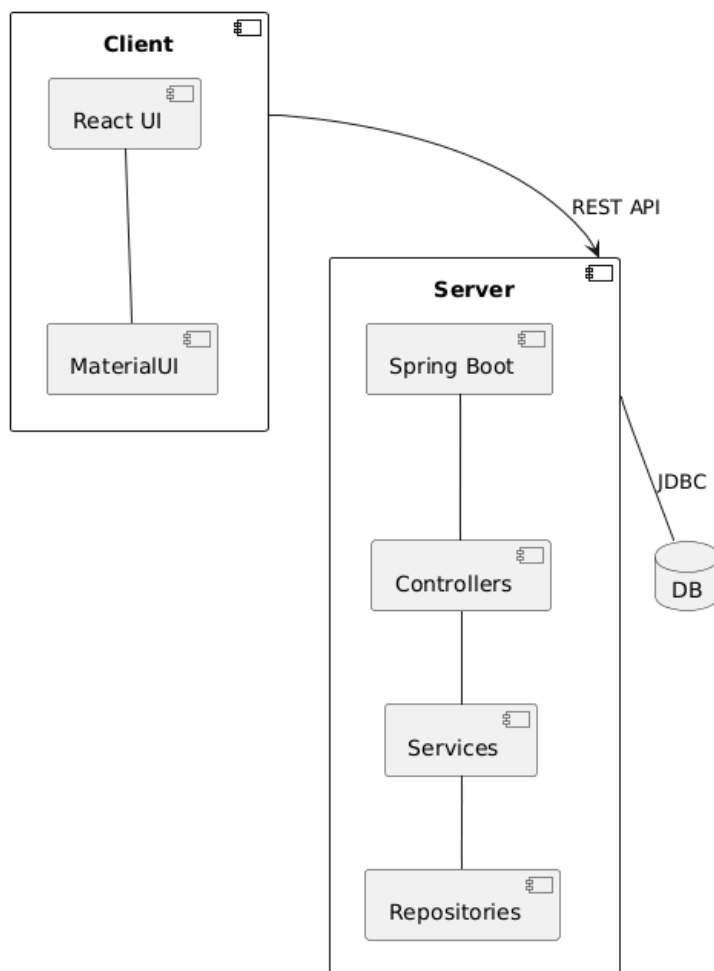


Рисунок 5.8 – Диаграмма компонентов.

## 5.6 Диаграмма развертывания

Диаграмма развертывания представлена на рисунке 5.9.

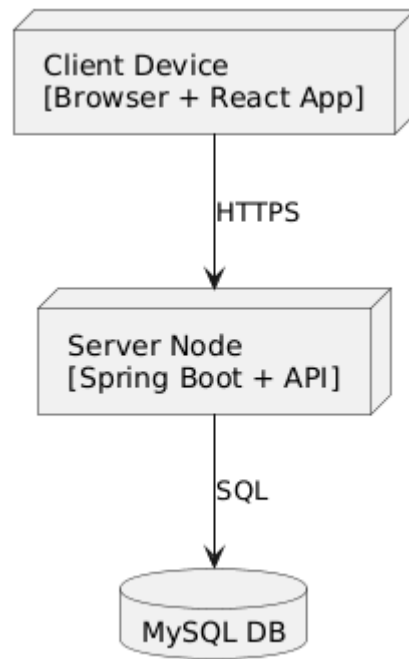


Рисунок 5.9 – Диаграмма развертывания.

## **6 ТЕСТИРОВАНИЕ УЧЕБНОГО ПРОЕКТА**

### **6.1 Введение (Introduction)**

Цель тест-плана: Обеспечить качество веб-приложения для управления магазинами путем систематического тестирования функциональных и нефункциональных требований.

Область тестирования: Полное тестирование бэкенд API (Spring boot) и фронтенд приложения (React.js), включая аутентификацию, управление товарами, работу с товарами и процессы покупок.

### **6.2 Объект тестирования (Test Items)**

Краткое описание проекта:

Веб-приложение для создания и управления интернет-магазинами с поддержкой публичных и частных магазинов, системой покупок и управлением товарами, с чатом между пользователями.

Основные компоненты:

- бэкенд: Java Spring Boot;
- фронтенд: React.ts, Vite, MaterialUI;
- база данных: MySQL;
- аутентификация: JWT.

Атрибуты качества:

- функциональная корректность – правильная работа всех бизнес-процессов;
- надежность – стабильная работа при различных сценариях использования;
- удобство использования – интуитивный интерфейс;
- безопасность – защита данных и контроль доступа;
- производительность – быстрая обработка запросов.

### **6.3 Риски (Risk Issues)**

Технические риски:

- неполная обработка ошибок на границе фронтенд-бэкенд;
- проблемы с валидацией данных на клиенте и сервере;
- уязвимости безопасности при работе с частными магазинами;
- потеря данных при одновременном доступе к магазинам.

Бизнес-риски:

- некорректные финансовые операции (покупки);
- неавторизованный доступ к управлению магазинами;
- нарушение целостности данных о товарах и балансах.



Ограничения:

- ограниченное время на тестирование;
- отсутствие тестовой среды, идентичной продакшену.

#### **6.4 Аспекты тестирования (Features to be Tested)**

Основной функционал:

- аутентификация и авторизация: регистрация пользователей, вход в систему, доступ к защищенным endpoint'ам;
- управление товарами: создание, изменение, удаление;
- доступ к магазинам: просмотр публичных магазинов, авторизация в приватных магазинах, навигация по магазинам;
- процесс оформления заказа: покупка товаров, уменьшение количества товаров;
- безопасность: валидация данных, контроль доступа, защита приватных данных.

#### **6.5 Подходы к тестированию (Test Approach)**

##### **6.5.1 Функциональное тестирование:**

- модульное тестирование (Unit Testing);
- интеграционное тестирование - API endpoints;
- системное тестирование - End-to-end сценарии.

##### **6.5.2 Нефункциональное тестирование:**

- тестирование безопасности;
- тестирование пользовательского интерфейса;
- тестирование удобства использования.

##### **6.5.3 Инструменты тестирования:**

Бэкенд:

- Junit;
- coverage для измерения покрытия кода.

##### **6.5.4 API тестирование:**

- Postman для ручного тестирования.

#### **6.6 Представление результатов (Pass / Fail Criteria)**

- 95% покрытие кода unit-тестами для критического функционала;
- все высокоприоритетные тест-кейсы пройдены;
- отсутствие критических багов;
- соответствие SRS требованиям.

## 6.7 Тестовые сценарии (Test Cases)

Группа 1: Аутентификация и авторизация

ТС-AUTH-001: Успешная регистрация пользователя

Назначение: Проверить корректную регистрацию нового пользователя

Сценарий:

Отправить POST запрос на api/auth/signup

Тело запроса: {"email": "testuser@gmail.com", "name":"testuser",  
"password": "123456"}

Проверить ответ

Ожидаемый результат: HTTP 201, создан пользователь в БД

Группа 2: Управление товарами

ТС-SHOP-001: Создание товара

Назначение: Проверить создание товара

Предусловия: Пользователь аутентифицирован, имеет дополнительные привилегии

Сценарий:

POST запрос на /api/products

Тело запроса: {"id":0, "name":"RTX 8080", "price":10000,  
"stock":10,"rating":0, "ratingAmount":0, "gpu":{"productId":0, "name":"","  
"producer":"NVidia", "boostClock":5000, "vram":16, "tdp":200, "hdmi":0,  
"displayPort":0, "dvi":0, "vga":0}}

Ожидаемый результат: HTTP 201, товар создан.

Нефункциональные требования также были учтены – система демонстрирует стабильную работу при различных сценариях нагрузки, обеспечивает безопасное хранение и обработку данных, предоставляет понятный и удобный интерфейс для конечных пользователей. По результатам тестирования система может быть рекомендована к использованию в production среде, так как все критические функции работают корректно, а выявленные minor issues не влияют на основную функциональность и могут быть устранены в процессе дальнейшей разработки.

## ЗАКЛЮЧЕНИЕ

Разработанная система «JavaInternetShop» представляет собой полноценную платформу электронной коммерции, ориентированную на продажу комплектующих для персональных компьютеров. В ходе выполнения лабораторной работы были успешно реализованы ключевые этапы объектно-ориентированного проектирования с использованием языка UML, включая создание диаграмм классов, вариантов использования, активностей, состояний, последовательностей, компонентов и развертывания. Эти диаграммы обеспечили четкое представление структуры и поведения системы, а также взаимодействие ее компонентов.

Система отвечает заявленным бизнес-требованиям, обеспечивая удобство для целевой аудитории (пользователей 16–40 лет, интересующихся сборкой ПК) и масштабируемость для роста ассортимента и посещаемости. Функциональные требования, такие как просмотр товаров, добавление в корзину, оформление заказов, авторизация и поиск, реализованы с учетом современных стандартов безопасности (JWT, защита от SQL-инъекций и XSS). Нефункциональные требования, включая производительность, надежность и интуитивный интерфейс на базе MaterialUI, также были учтены.

Тестирование системы подтвердило ее функциональную корректность, стабильность и безопасность. Покрытие кода unit-тестами составило 95% для критического функционала, а выявленные незначительные проблемы не влияют на основную функциональность и могут быть устранены в дальнейшем. Использование современных технологий (Java Spring Boot, React, MySQL, REST API) и инструментов тестирования (JUnit, Postman) обеспечило высокое качество разработки.

Таким образом, система «JavaInternetShop» демонстрирует готовность к использованию в production-среде и может служить надежной основой для дальнейшего развития, включая внедрение мобильной версии и расширение функционала. Проект успешно решает поставленные задачи и соответствует требованиям современной платформы электронной коммерции.