

МИНОБРНАУКИ РОССИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ
«ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»
(ФГБОУ ВО «ВГУ»)

Факультет компьютерных наук
Кафедра информационных систем

Веб приложение для покупки комплектующих для ПК "YourPC"

Курсовая работа

09.03.02 Информационные системы и технологии

Информационные системы и сетевые технологии

6 семестр 2022/2023 учебного года

Зав. кафедрой	_____	к. т. н., доцент Д.Н. Борисов
Обучающийся	_____	ст. 3 курса оч. отд. В. С. Железной
Обучающийся	_____	ст. 3 курса оч. отд. А. Н. Костромин
Обучающийся	_____	ст. 3 курса оч. отд. П. А. Комиссаров
Руководитель	_____	ст. преп. В.С. Тарасов

Содержание

1 Терминология.....	4
2 Введение.....	7
2.1 Обзор сферы веб сайтов для комплектующих ПК.....	7
2.2 Значение и актуальность приложений для обмена рецептами.....	7
2.3 Цели и задачи работы.....	7
3 Разработка пользовательского интерфейса.....	8
3.1 Дизайн интерфейса.....	8
3.2 UX/UI тестирование.....	8
4 Обзор предметной области.....	10
4.1 История развития и состояние рынка кулинарных приложений.....	10
4.2 Анализ конкурентов и их функциональности.....	10
4.3 Преимущества и недостатки существующих решений.....	10
5 Техническое задание.....	11
5.1 Обзор общих требований к сайту и его функциональных характеристик.....	11
5.2 Описание архитектуры и использование технологий (React, REST API, MongoDB).....	11
5.2.1 Технологический обзор: React Native, REST API.....	12
5.2.2 Безопасность мобильных приложений.....	13
6 Функционал приложения.....	15
6.1 Описание основных функций приложения: регистрация, добавление в избранное, отзывы....	15
6.3 Описание мер по обеспечению безопасности и защиты данных.....	16
7 Перспективы развития приложения.....	17
7.1 Анализ пользовательской активности и возможности сбора обратной связи.....	17
7 Навигация по сайту.....	18
7.1 Основное навигационное меню.....	18
9.2 Дополнительное навигационное меню.....	18
8 Описание страниц сайта.....	19
8.1 Описание статических страниц.....	19
8.2 Описание динамических страниц.....	19
9 Контент и наполнение сайта.....	21
10 Разработка.....	21
11 Вывод.....	27
ПРИЛОЖЕНИЕ.....	28

1 Терминология

Проект, система – разрабатываемое веб-приложение.

Личный кабинет клиента – веб-страница, где отражаются персональная информация пользователя, введенная на этапе регистрации и/или отредактированная позже, а также история заказов данного пользователя, переход к каталогу и переход к странице редактирования личных данных.

Личный кабинет администратора – веб-страница, где отражаются список заявок на аренду машины и список принятых заявок, информация о клиентах, а также возможность назначить на заказ-доставку водителя курьера.

Администратор – пользователь, обслуживающий веб-приложение, осуществляющий связь с клиентами и данные которого были заранее занесены в систему.

Front–end – клиентская часть приложения. Отвечает за получение информации с программно–аппаратной части и отображение ее на устройстве пользователя.

Сервер, серверная часть – компьютер, обслуживающий другие компьютеры (клиентов) и предоставляющий им свои ресурсы для выполнения определенных задач.

Клиентская сторона – компьютер, использующий ресурсы сервера и предоставляющий пользователю возможность взаимодействия с системой.

Back–end – программно–аппаратная часть веб-приложения. Отвечает за функционирование внутренней части веб-приложения.

GitHub – веб-сервис для хостинга IT–проектов и их совместной разработки.

Фреймворк – программное обеспечение, облегчающее разработку и объединение разных компонентов большого программного проекта.

Пользователь (Клиент) – человек, пользующийся функционалом

веб-приложения.

REST API (REST) – стиль архитектуры программного обеспечения для построения масштабируемых веб-приложений.

Система управления базами данных (СУБД) – это набор программ, которые управляют структурой БД и контролируют доступ к данным, хранящимся в БД.

React – JavaScript-библиотека для создания пользовательских интерфейсов.

HTML – стандартизированный язык разметки для просмотра веб-страниц в браузере.

TypeScript — язык программирования, представленный Microsoft в 2012 году и позиционируемый как средство разработки веб-приложений, расширяющее возможности JavaScript.

MongoDB — документо ориентированная система управления базами данных, не требующая описания схемы таблиц. Считается одним из классических примеров NoSQL-систем, использует JSON-подобные документы и схему базы данных.

Mongoose — это библиотека объектно-ориентированного программирования JavaScript, которая создает соединение между MongoDB и средой выполнения JavaScript Node.js.

Express — фреймворк web-приложений для Node.js, реализованный как свободное и открытое программное обеспечение под лицензией MIT. Он спроектирован для создания веб-приложений и API.

JSON Web Token — это открытый стандарт для создания токенов доступа, основанный на формате JSON. Как правило, используется для передачи данных для аутентификации в клиент-серверных приложениях.

Bcrypt — адаптивная криптографическая хеш-функция формирования ключа, используемая для защищенного хранения паролей.

Feature-Sliced Design (FSD) — это архитектурная методология для

проектирования frontend-приложений.

Vite – это инструмент сборки, созданный для обеспечения быстрого и бережливого (lean) процесса разработки современных веб-проектов.

React — JavaScript-библиотека с открытым исходным кодом для разработки пользовательских интерфейсов.

React Router DOM — это модуль узла, который предназначен для маршрутизации в веб-приложениях.

Redux — это база данных в программе. В этой базе хранится состояние — то есть данные приложения. Эта база отвечает только за данные, поэтому она никак не связана с браузером, DOM и фронтом в целом.

2 Введение

2.1 Обзор сферы веб сайтов для комплектующих ПК.

Сфера веб-сайтов для комплектующих ПК является одной из самых важных и популярных среди тех, кто интересуется сборкой и апгрейдом компьютеров. На таких сайтах можно найти огромное количество информации о различных комплектующих: процессорах, видеокартах, оперативной памяти, жестких дисках, блоках питания и так далее.

2.2 Значение и актуальность приложений для обмена рецептами.

Сфера веб-сайтов для комплектующих ПК является важным ресурсом для всех, кто следит за развитием компьютерной индустрии, стремится собрать оптимальную конфигурацию ПК или просто хочет быть в курсе последних новостей и тенденций в мире вычислительной техники.

2.3 Цели и задачи работы.

Основная цель данной курсовой работы - разработка веб сайта "YourPC", которое предоставляет платформу для поиска и покупки комплектующих. Этот сайт должен стать удобным инструментом для пользователей, заинтересованных в данной сфере, позволяя им не только находить комплектующие, но и покупать новые, а также оставлять отзывы.

Для достижения этой цели были поставлены следующие задачи:

- _ Исследование существующего рынка и анализ конкурентов.
- _ Определение функциональных требований и технических спецификаций на основе потребностей целевой аудитории.
- _ Проектирование архитектуры и интерфейса пользователя.
- _ Реализация основных функций приложения.

3 Разработка пользовательского интерфейса

3.1 Дизайн интерфейса

В разработке сайта "YourPC" особое внимание уделяется дизайну интерфейса, который является ключевым фактором в обеспечении удобства и привлекательности приложения для пользователя. Процесс создания дизайна включает несколько этапов:

- **Исследование пользователей и сбор требований:** Первый этап заключается в изучении целевой аудитории и выявлении основных требований к функциональности и дизайну сайта. Это позволяет определить ключевые элементы интерфейса и их оптимальную организацию для обеспечения удобного пользовательского опыта.
- **Разработка макетов:** создаются детализированные макеты интерфейса с использованием инструментов дизайна, например Figma. В этом процессе уточняются цветовая схема, шрифты и расположение элементов управления.
- **Адаптивный дизайн:** учитывается необходимость адаптации интерфейса под различные размеры экранов и ориентации устройств.

3.2 UX/UI тестирование

Тестирование пользовательского интерфейса (UI) и пользовательского опыта (UX) критически важно для определения соответствия разработанного продукта ожиданиям и потребностям конечных пользователей. Тестирование проводится на нескольких этапах разработки:

— **Юзабилити-тестирование:** оценивает удобство и интуитивность интерфейса. Тесты могут проводиться в лабораторных условиях с участием реальных пользователей. Пользователи выполняют типичные задачи, позволяя идентифицировать любые проблемы с навигацией, доступностью функций или пониманием контента.

Результаты этих тестов влияют на дизайн, помогая дизайнерам и разработчикам улучшить продукт перед его окончательным запуском.

4 Обзор предметной области

4.1 История развития и состояние рынка кулинарных приложений.

Рынок интернет-магазинов комплектующих для ПК начал своё развитие вместе с развитием самого интернета и онлайн-торговли в конце 20 века и начале 21 века. В это время многие компании, предлагающие комплектующие для ПК, стали создавать онлайн-присутствие для удобства покупателей. Это позволило потребителям найти и купить комплектующие для своих ПК, не выходя из дома.

4.2 Анализ конкурентов и их функциональности

На рынке существует несколько популярных приложений, которые можно считать конкурентов "YourPC", например, "ДНС", "Ситилинк". Эти сайты предлагают пользователям возможность просматривать товары, добавлять их в избранное, а также делиться отзывами.

4.3 Преимущества и недостатки существующих решений

Одним из основных преимуществ существующих приложений является их способность предоставлять пользователю обширный выбор. Однако, существующие решения часто страдают от перегруженности интерфейса и недостаточного количества информации по данным товарам.

"YourPC" стремится устранить эти недостатки, предлагая удобный, интерфейс и увеличения количества информации.

5 Техническое задание

5.1 Обзор общих требований к сайту и его функциональных характеристик.

Согласно техническому заданию, сайт "YourPC" должен предоставлять пользователям платформу поиска и покупки комплектующих ПК. Приложение позволит пользователям сохранять в избранное, сравнивать, просматривать, покупать, а также оставлять отзывы. Он также должен поддерживать функции регистрации и входа для различения между анонимными и зарегистрированными пользователями.

Основные функции приложения включают:

- **Регистрация и аутентификация:** позволяет пользователям создавать учетные записи и входить в систему для доступа к персональным и дополнительным функциям.
- **Интерактивные функции:** возможность оценивать товары, добавлять их в избранное.
- **Личный профиль:** просмотр и редактирование личной информации и смена пароля.

5.2 Описание архитектуры и использование технологий (React, REST API, MongoDB).

Приложение "YumYard" будет использовать модернизированную клиент-серверную архитектуру с использованием следующих технологий:

Frontend (клиентская часть):

- **React Native:** использование React Native позволит разрабатывать нативные мобильные приложения для iOS и Android с

использованием JavaScript, что существенно ускоряет разработку и поддержку приложений на обеих платформах.

- **Взаимодействие с сервером:** коммуникация между клиентом и сервером будет осуществляться через REST API, что позволяет поддерживать стандарты и упрощает интеграцию с другими системами и сервисами.

Безопасность и защита данных:

- **Аутентификация и авторизация:** безопасность будет обеспечиваться с помощью системы аутентификации и авторизации, которые контролируют доступ к функционалу приложения.

В целом, выбор этих технологий направлен на создание надежного, безопасного и легко масштабируемого приложения, которое может быть адаптировано к изменяющимся требованиям и условиям рынка.

5.2.1 Технологический обзор: React Native, REST API

React Native

- **Архитектура:** React Native позволяет разработчикам использовать React для разработки нативных сайтов. Код TypeScript выполняется в отдельном потоке, обеспечивая взаимодействие с нативным API через асинхронный мост.
- **Принципы работы:** React Native использует компонентный подход, что позволяет создавать переиспользуемые блоки интерфейса, что улучшает эффективность разработки и поддерживаемость кода.

REST API

- **Архитектура:** REST (Representational State Transfer) — это стиль архитектуры программного обеспечения для систем, основанных на сети, включая веб. RESTful API позволяет взаимодействовать с веб-ресурсами через HTTP с использованием методов, таких как GET, POST, PUT и DELETE.
- **Принципы работы:** REST API взаимодействует с веб-приложениями, отправляя запросы на сервер и получая ответы. Эти API должны быть без состояний, что означает, что каждый запрос должен содержать всю необходимую информацию для его обработки.

5.2.2 Безопасность мобильных приложений

Шифрование

- **Описание:** шифрование — это процесс кодирования информации таким образом, что только уполномоченные стороны могут ее дешифровать. В мобильных приложениях данные пользователя, отправляемые на сервер или хранимые локально, должны быть зашифрованы для предотвращения несанкционированного доступа.
- **Технологии:** используются такие стандарты, как AES (Advanced Encryption Standard) для шифрования данных и SSL/TLS для защиты данных, передаваемых по сети.

Аутентификация и авторизация

- **Описание:** аутентификация проверяет, является ли пользователь тем, за кого он себя выдает, в то время как авторизация определяет, к каким ресурсам у пользователя есть доступ.

Защита от уязвимостей

- **Описание:** сайты подвержены различным уязвимостям, таким как SQL-инъекции, XSS (межсайтовый скриптинг), и межсайтовая подделка запросов (CSRF).
- **Меры предосторожности:** регулярное обновление зависимостей, использование проверенных библиотек, проведение пентестирования и код-ревью, а также обучение разработчиков основам безопасности.

6 Функционал приложения

6.1 Описание основных функций приложения: регистрация, добавление в избранное, отзывы.

Регистрация и аутентификация: пользователи могут создавать свои учетные записи в приложении "YourPC", предоставляя базовую информацию, такую как имя пользователя, электронная почта и пароль. Аутентификация будет обеспечиваться через стандартные механизмы входа с подтверждением по электронной почте для усиления безопасности учетных записей.

Добавление и редактирование рецептов: зарегистрированные пользователи могут добавлять товары в избранное. Пользователи также могут редактировать и удалять выбранные товары.

Комментирование и оценки: зарегистрированные пользователи могут оставлять комментарии к товарам и оценивать их по пятибалльной шкале.

6.2 Описание интерфейса пользователя и навигации по приложению

Главная страница: отображает популярные товары, а также предоставляет доступ к функции каталога. Главная страница служит входной точкой для исследования контента сайта.

Каталог: включает в себя элементы для быстрого доступа к главной странице, избранному, заказам, корзине и личному профилю. Для неавторизованных пользователей предоставляется ограниченный доступ с возможностью только просматривать товары и их описание.

Страница товара: показывает полную информацию о характеристиках товара, включая оценки пользователей.

6.3 Описание мер по обеспечению безопасности и защиты данных

Защита данных: все данные пользователей и рецепты хранятся в защищенной базе данных. Доступ к данным строго регулируется и ограничен соответствующими правами доступа.

Аутентификация и сессии: приложение использует защищенные методы аутентификации и управления сессиями, включая HTTPS для шифрования данных во время передачи. Сессии пользователя защищены токенами, которые предотвращают перехват и подделку сессий.

CSRF и XSS защита: Веб-интерфейс защищен от атак типа CSRF (Cross-Site Request Forgery) и XSS (Cross-Site Scripting), используя современные методы защиты, включенные в фреймворки Django и React.

Эти меры по обеспечению безопасности и защиты данных позволяют пользователю чувствовать себя уверенно, используя приложение для обмена кулинарными рецептами, и способствуют созданию доверительной и безопасной среды для всех пользователей.

7 Перспективы развития приложения

7.1 Анализ пользовательской активности и возможности сбора обратной связи.

Для постоянного улучшения качества и функциональности приложения "YourPC" предусмотрено внедрение системы аналитики, которая позволит отслеживать активность пользователей: частоту использования приложения, наиболее популярные рецепты, активность в комментариях и рейтингах. Эта информация будет использоваться для адаптации контента и функциональных возможностей под интересы и предпочтения пользователей.

7.2 Оценка потенциальных направлений монетизации сайта.

Монетизация приложения "YourPC" может быть реализована через несколько каналов:

- **Реклама:** внедрение таргетированной рекламы.
- **Премиум подписки:** введение платных подписок, предоставляющих доступ к статьям о мире комплектующих.
- **Партнерские программы:** сотрудничество с производителями для предложения специальных акций и скидок пользователям приложения и продвижения в популярных товарах.

Эти направления не только помогут в финансировании дальнейшего развития и поддержки приложения, но и усилят его привлекательность для пользователей, предлагая им дополнительную ценность.

7 Навигация по сайту

7.1 Основное навигационное меню

Логотип закреплен в верхнем левом углу. Основное меню навигации на экранах расположено в верхней части экрана (в шапке). Оно позволяет переходить на страницы:

- Главная страница;
- Каталог;
- Личный кабинет;
- Мои заказы (все заказы, если пользователь является администратором);
- Корзина;
- Избранное;
- Сравнение.

9.2 Дополнительное навигационное меню.

Логотип закреплен в нижнем левом углу. Дополнительное меню навигации на экранах расположено в нижней части экрана. Оно позволяет переходить на страницы компании в соцсетях.

8 Описание страниц сайта

8.1 Описание статических страниц

Страница «Каталог товаров» включает в себя:

- Список разделов.

Страница «Карточка товара» включает в себя:

- Подробную информацию о товаре.

Страница «Главная страница» включает в себя:

- Список популярных товаров разбитых по разделам.

Страница «Личный кабинет» Клиента и Администратора включает в себя:

- Блок с личными данными;
- Возможность редактирования личных данных.

8.2 Описание динамических страниц

Страница «Выбранный раздел» включает в себя:

- Перечень товаров, соответствующих выбранному разделу.

Страница «Авторизация» включает в себя:

- Форму ввода данных для авторизации;
- Кнопка “забыли пароль?”.

Страница «Регистрация» включает в себя:

- Форму ввода данных для авторизации.

Страница «Восстановление пароля» включает в себя:

- Форму ввода данных для восстановления пароля.

Страница «Выбранный раздел» включает в себя:

- Перечень товаров, соответствующих выбранному разделу.

Страница «Мои заказы» включает в себя:

- Информацию об оформленных заказах.

Страница «Все заказы» включает в себя:

- Информацию об оформленных заказах.

Страница «Корзина» включает в себя:

- Список товаров выбранных к заказу;
- Кнопка “Оформить заказ”.

Страница «Избранное» включает в себя:

- Информацию о товарах выделенных в избранное.

Страница «Сравнение» включает в себя:

- Характеристики товаров, отправленных в сравнение.

9 Контент и наполнение сайта

Основным наполнением сайта служит сущность «Каталог», благодаря которой пользователь имеет возможность просматривать перечень групп товаров, представленных у компании и дополнительную информацию о выбранном товаре.

10 Разработка

10.1 Архитектура проекта

— Frontend часть

Проект разделен на frontend и backend части, каждая из которых организована с использованием современных подходов и инструментов для обеспечения удобства разработки, поддержки и масштабирования.

Фронтенд-часть:

- shared: содержит общие компоненты, утилиты и стили, которые могут использоваться в различных частях приложения.
- entities: включает бизнес-логические сущности, такие как модели данных и сервисы для работы с ними.
- features: реализует отдельные функциональные возможности приложения.
- widgets: содержит виджеты, которые могут быть переиспользованы на разных страницах приложения.
- pages: хранит страницы приложения, каждая из которых представляет собой отдельный маршрут.

—app: главный каталог приложения, включает конфигурационные файлы и корневой компонент (Рис. 1).

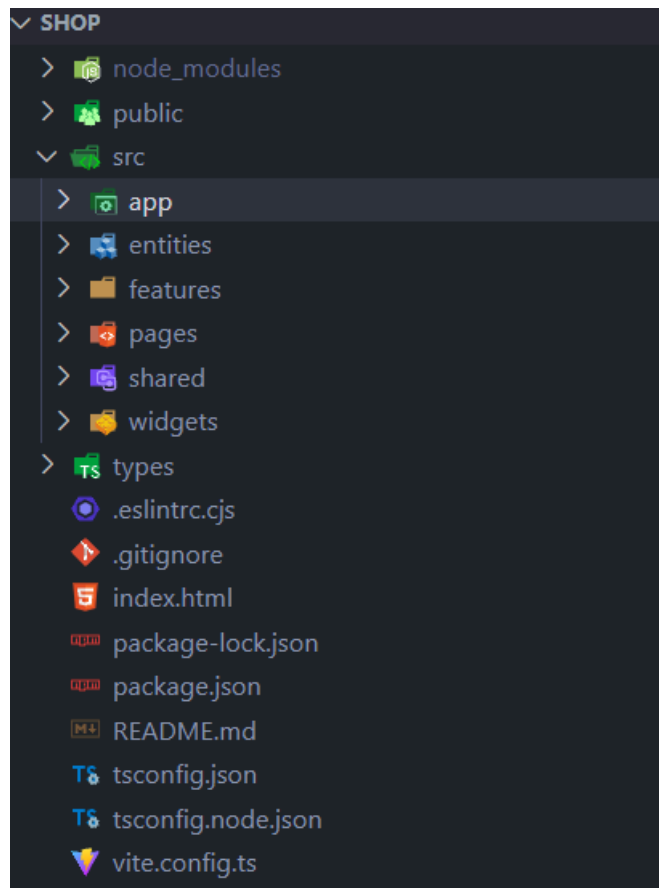


Рисунок 1 – Файловая структура Frontend

— Backend часть

Бэкенд-часть (Рис. 2):

- config: содержит конфигурационные файлы и параметры приложения.
- controllers: включает контроллеры, которые обрабатывают запросы и отправляют ответы.
- core: содержит основные функциональные модули, такие как

подключение к базе данных.

—dtos: включает объекты передачи данных (DTO), используемые для обмена данными между клиентом и сервером.

—middleware: содержит промежуточное ПО для обработки запросов.

—models: включает схемы и модели данных для работы с MongoDB.

—routes: содержит маршруты для обработки различных запросов.

—services: включает бизнес-логику и сервисы для обработки данных.

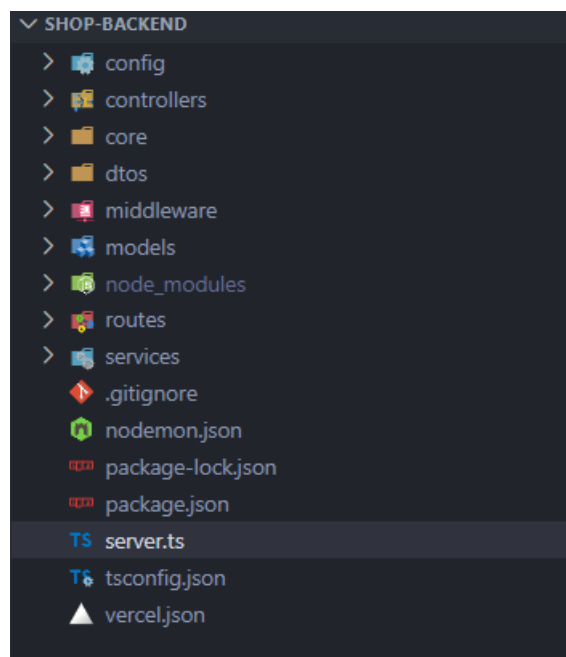


Рисунок 2 – Файловая структура Backend

Главный файл бэкенда (server.ts) показан на Рис.3.

10.2 Обоснование выбора стека технологий для Frontend и Backend частей

Frontend:

- React: используется для создания пользовательского интерфейса благодаря его компонентному подходу и виртуальному DOM, что обеспечивает высокую производительность.
- TypeScript: добавляет статическую типизацию в JavaScript, что улучшает качество кода и упрощает отладку.
- Redux: используется для управления состоянием приложения, что особенно важно для сложных интерфейсов с множеством взаимодействий.
- Material UI: библиотека компонентов для React, которая позволяет быстро создавать стильные и удобные пользовательские интерфейсы.
- Vite: инструмент для быстрой сборки проектов, который обеспечивает высокую скорость разработки.

Backend:

- Node.js: платформа для выполнения JavaScript на серверной стороне, которая позволяет создавать масштабируемые приложения.
- TypeScript: обеспечивает статическую типизацию и повышает надежность кода.
- Express: минималистичный веб-фреймворк для Node.js, который упрощает создание серверных приложений.

—MongoDB и Mongoose: MongoDB - это NoSQL база данных, а Mongoose предоставляет удобные средства для работы с ней в Node.js, включая схемы и валидацию данных.

Взаимодействие между клиентской и серверной частями

Клиентская и серверная части взаимодействуют через REST API. Клиент отправляет HTTP-запросы к серверу, который обрабатывает их и возвращает необходимые данные или результаты операций. Основные маршруты и обработчики запросов определены в каталоге routes на бэкенде и используются для выполнения CRUD операций над товарами, пользователями и заказами. Использование JSON для передачи данных обеспечивает удобство и стандартизированный формат обмена информацией между клиентом и сервером.

10.3 Реализация пользовательского интерфейса

— Главная страница

Главная страница интернет-магазина комплектующих для ПК предоставляет пользователю обзор популярных и рекомендованных товаров. Она включает следующие компоненты (Рис. 3):

—Header: верхняя панель с логотипом, навигационным меню и иконками для входа в профиль, корзины и поиска.

—Banner: рекламный баннер с текущими акциями и предложениями.

—ProductHighlights: секция с рекомендованными товарами и новинками.

—Footer: нижняя панель с дополнительной информацией о магазине, контактами и ссылками на социальные сети.

Material UI используется для создания компонентов интерфейса,

обеспечивая их современный внешний вид и удобство использования. Основные компоненты, используемые на главной странице:

- AppBar и Toolbar для верхней панели.
- Grid для размещения элементов в виде сетки.
- Card для отображения товаров с изображениями, названиями и ценами.
- Typography для стилизованного текста.

```
export const ProductsPage = () => {  
  const dispatch = useAppDispatch();  
  const products = useAppSelector(productsSelector);  
  const pathname = useLocation().pathname as EPathnameProducts;  
  
  useEffect(() => {  
    dispatch(getProductsAsync({ pathname }));  
  }, [pathname]);  
  
  return (  
    <Box sx={{ marginBottom: "30px" }}>  
      <ProductsListWidget title={getTitle(pathname)} items={products} />  
    </Box>  
  );  
};
```

Рисунок 3 – Главная страница

— Корзина и оформление заказа

Корзина отображает добавленные товары и общую сумму заказа. Пользователь может изменять количество товаров или удалять их из корзины.

11 Вывод

В ходе работы наша команда реализовала удобное веб приложение для покупки комплектующих

ПРИЛОЖЕНИЕ

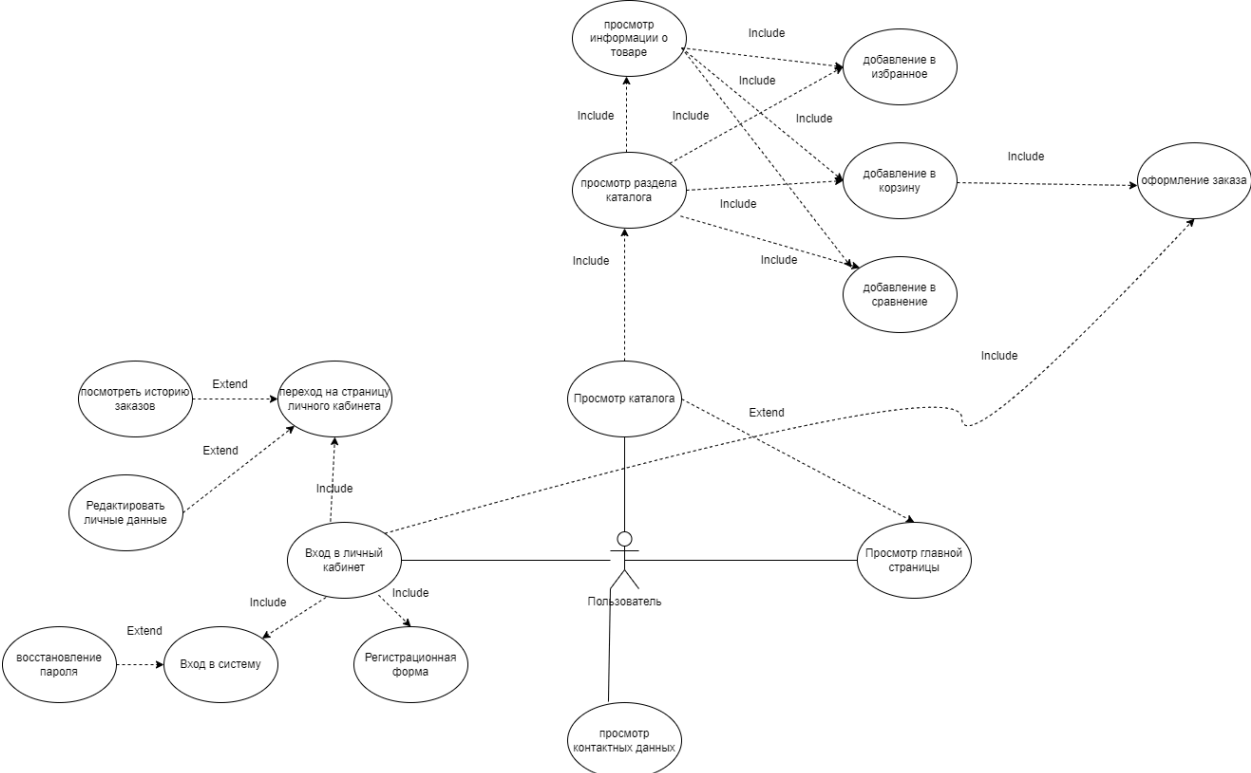


Рисунок 1 - Диаграмма Use-Case со стороны Клиента

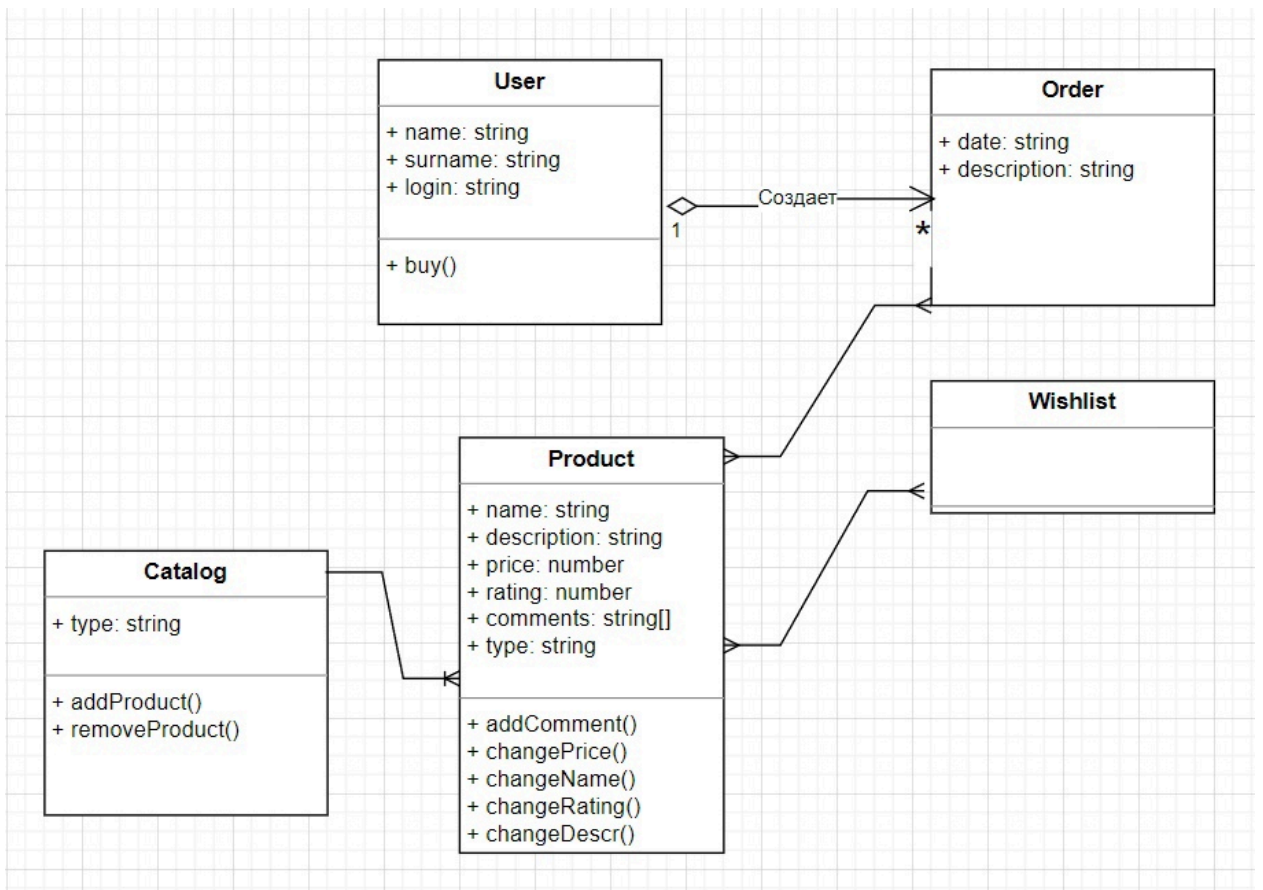


Рисунок 2 - Диаграмма классов

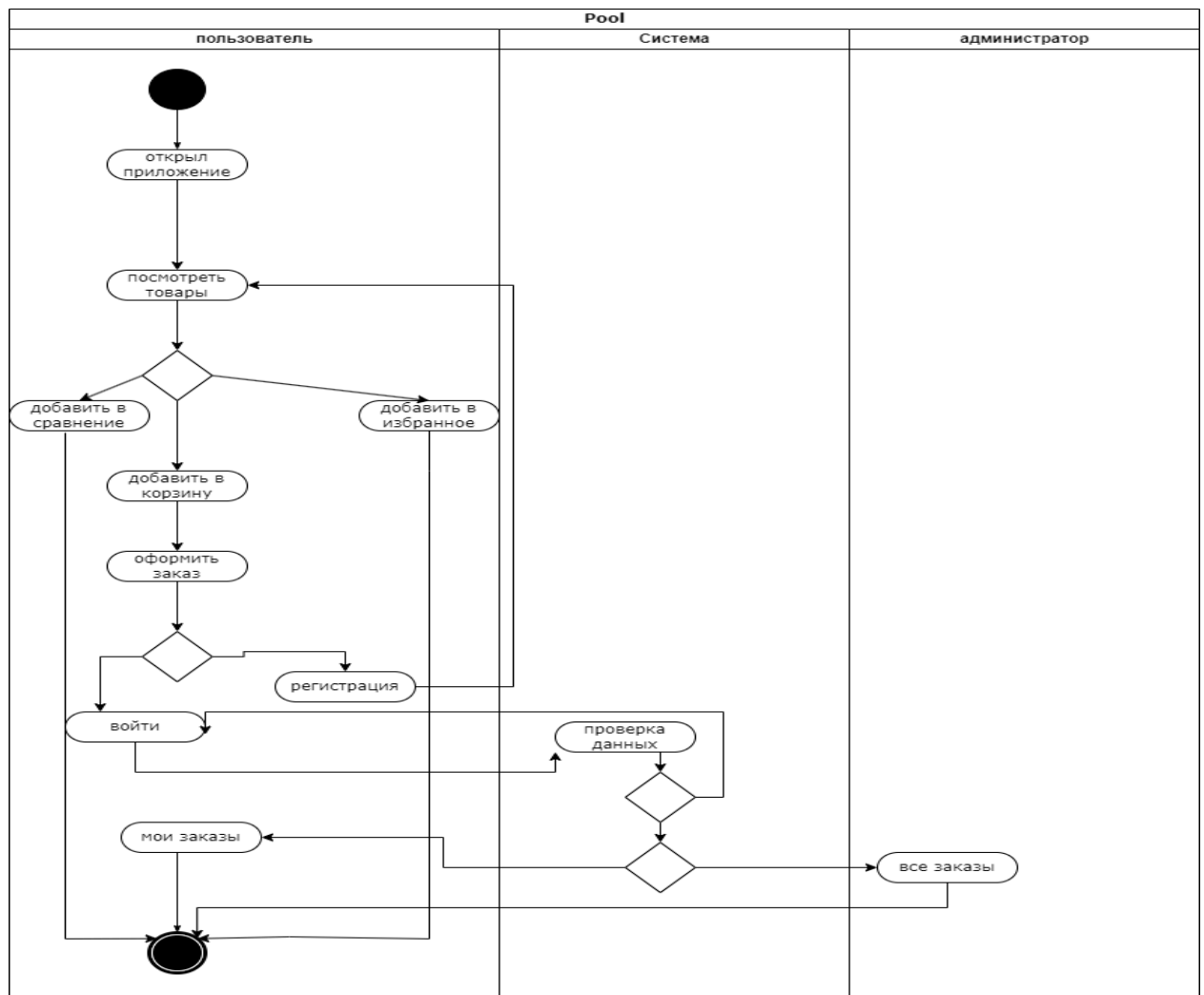


Рисунок 3 - Диаграмма активностей

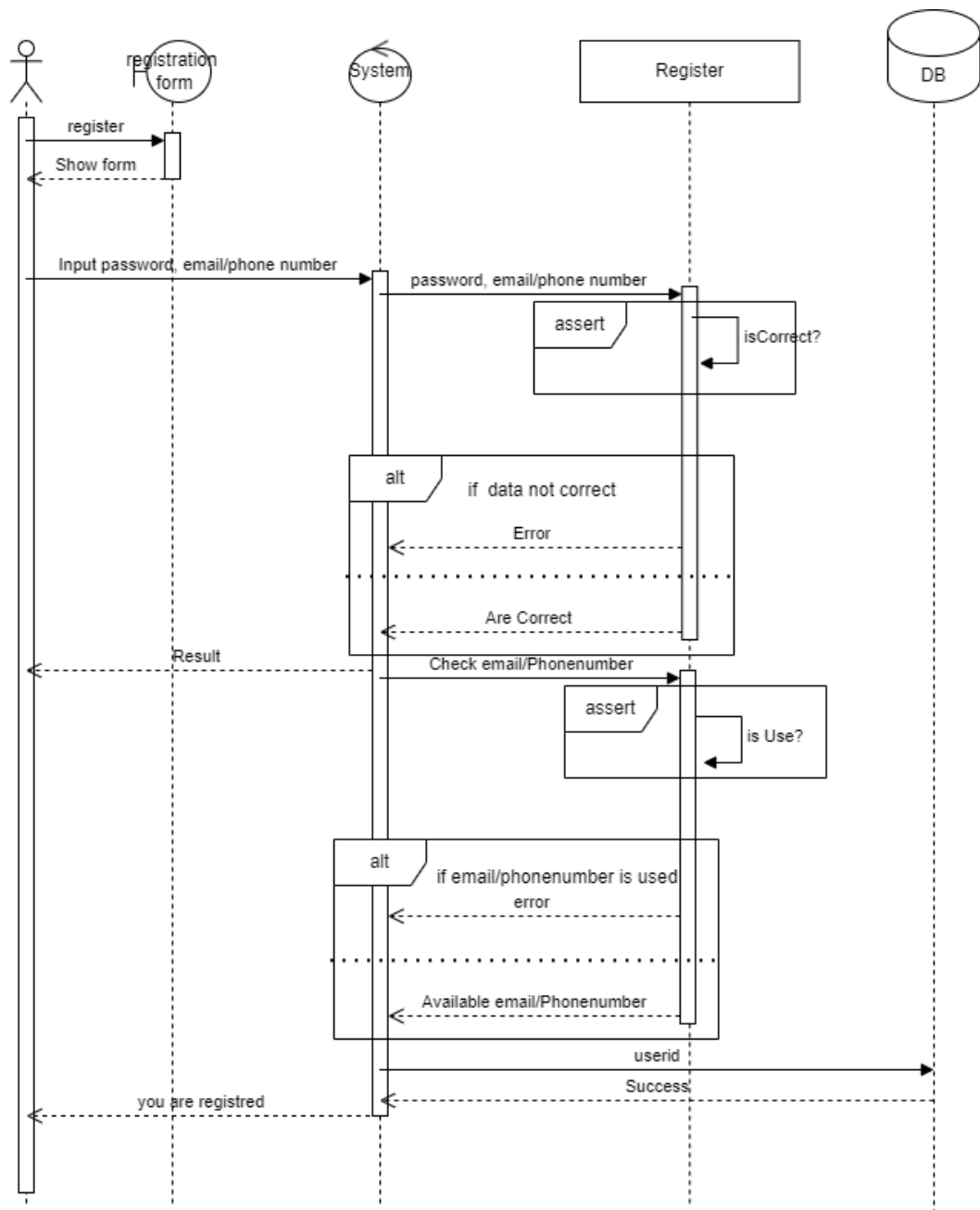


Рисунок 4 - Диаграмма последовательностей

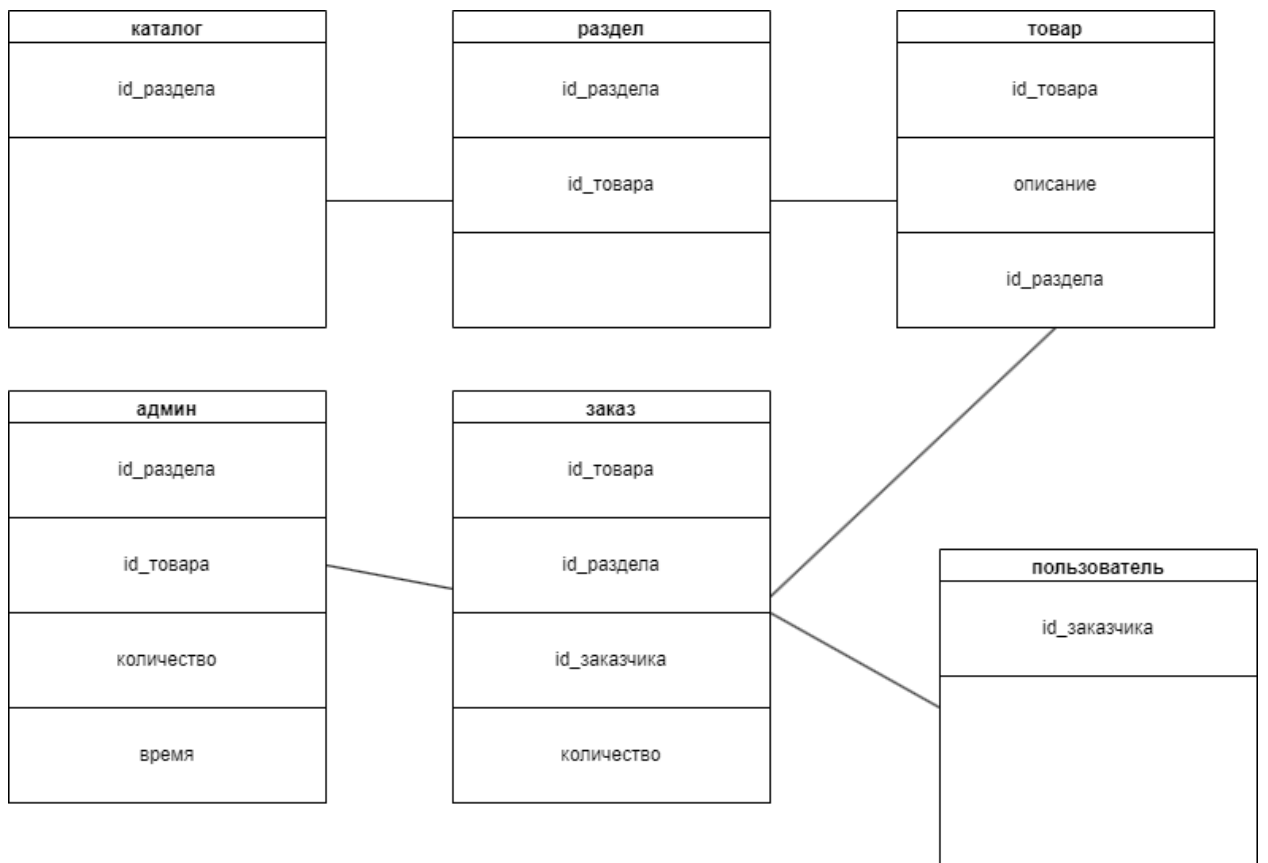


Рисунок 5 - ER-диаграмма

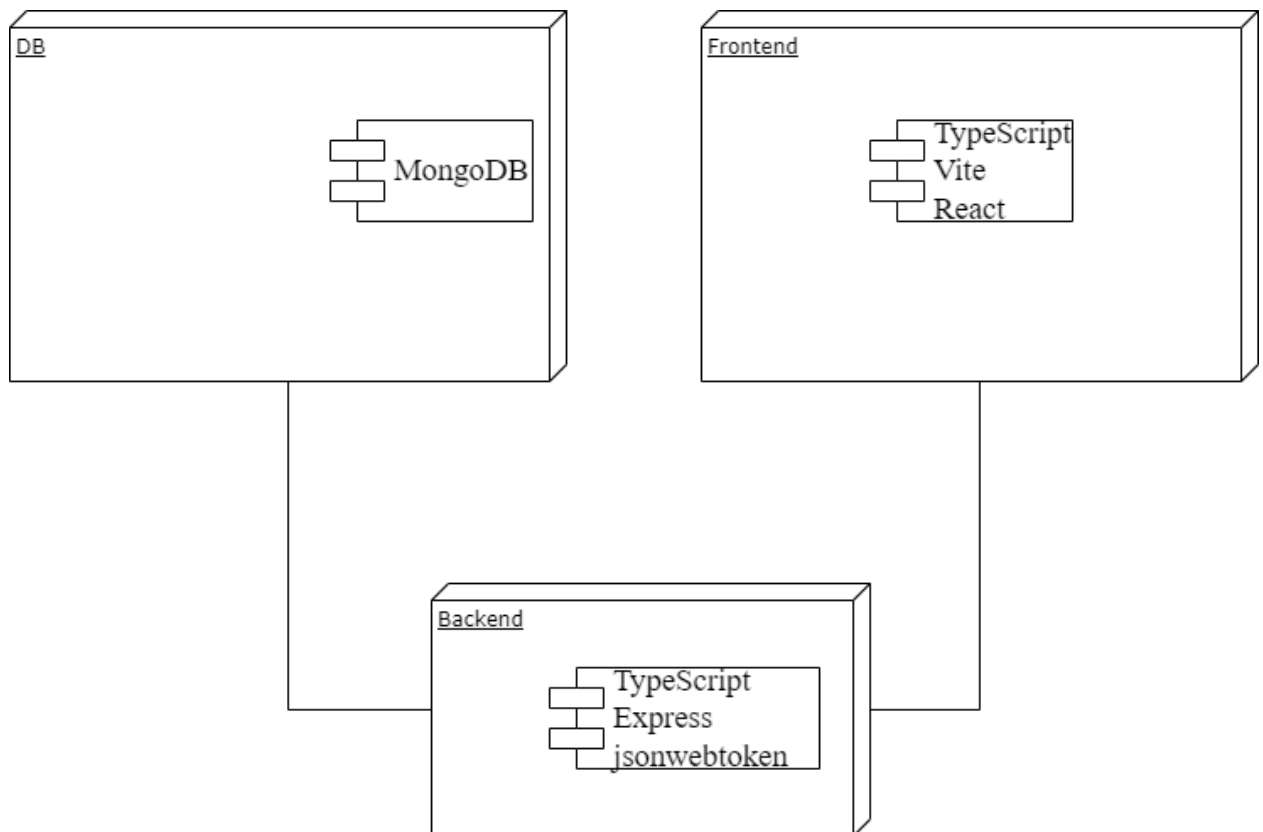


Рисунок 6 - Statechart диаграмма