

Министерство цифрового развития, связи и массовых коммуникаций РФ
Уральский технический институт связи и информатики (филиал)
ФГБОУ ВО «Сибирский государственный университет
телекоммуникаций и информатики» в г. Екатеринбурге
(УрТИСИ СибГУТИ)



Уральский технический
институт связи
и информатики



Информационные
системы и
технологии

КАФЕДРА
Информационные системы и технологии
(ИСиТ)

Курсовой проект
по дисциплине «Сетевое программирование»
Разработка приложения для автоматизации оптового склада

Выполнил:

студент гр. ПЕ-126
Камков Д. А.

Проверил:

Преподаватель
Бурумбаев Д. И.

г. Екатеринбург, 2024

Министерство цифрового развития, связи и массовых коммуникаций РФ
Уральский технический институт связи и информатики (филиал)
ФГБОУ ВО «Сибирский государственный университет
телекоммуникаций и информатики» в г. Екатеринбурге
(УрТИСИ СибГУТИ)
Кафедра «Информационные системы и технологии»

ЗАДАНИЕ НА ВЫПОЛНЕНИЕ КУРСОВОГО ПРОЕКТА

По дисциплине _____ Сетевое программирование _____

Студент _____ Камков Дмитрий Анатольевич _____

Курс _____ 3 _____ Группа _____ ПЕ-126 _____

ИСХОДНЫЕ ДАННЫЕ

Разработка приложения для автоматизации работы оптового склада

ПЕРЕЧЕНЬ ВОПРОСОВ ДЛЯ ИССЛЕДОВАНИЯ

В курсовом проекте необходимо разработать приложение, которое будет автоматизировать бизнес-процесс автомастерской. Для этого необходимо:

- 1) анализ предметной области согласно теме курсового проекта;
- 2) разработать техническое задания на проектируемое программное обеспечение согласно ГОСТ;
- 3) выбрать стек-технологий, необходимый для разработки приложения;
- 4) спроектировать и разработать базу данных для разработки программного обеспечения;
- 5) разработать блок-схему работы приложения согласно ГОСТ;
- 6) разработать интерфейс программного продукта;
- 7) рассмотреть вопросы информационной безопасности при эксплуатации программного продукта.

Дата выдачи _____ Срок сдачи: _____

Допуск к защите _____ Оценка _____

Руководитель преподаватель _____ Бурумбаев Д. И. _____

Студент _____ Камков Д. А. _____

Отзыв руководителя от УрТИСИ

Содержание

Введение	4
1. Анализ предметной области	5
2. Разработка технического задания	7
3. Выбор используемых технологий	11
4. Проектирование и разработка БД	13
4.1. Анализ предметной области и информационных задач пользователей	13
4.2. Формирование схемы данных	14
4.3. Выбор СУБД и других программных средств	17
4.4. Описание групп пользователей и прав доступа	17
5. Разработка блок-схем	18
6. Разработка интерфейса ПО	21
6.1. Проектирование интерфейса	21
7. Информационная безопасность	26
Заключение	27
Библиография	28

					09.03.01.000021 П.552 ПЗ			
Изм.	Лист	№ докум.	Подпись	Дата				
Разраб.		Камков Д. А.			Разработка приложения для автоматизации работы оптового склада Пояснительная записка	Лит.	Лист	Листов
Пров.		Бурумбаев Д. И.					3	28
Реценз.						УрТИСИ СибГУТИ		
Н. Контр.		Бурумбаев Д. И.						
Утв.								

Введение

В современном мире, где веб-технологии играют ключевую роль в повседневной жизни, автоматизация процессов становится неотъемлемой частью различных отраслей бизнеса. В частности, оптовые склады, предоставляющие широкий спектр товаров, ощущают необходимость в оптимизации своей работы с помощью современных технологий.

Актуальность разработки приложения для автоматизации оптового склада ещё более подчёркивается указом Президента от 09.05.2017 года "О развитии цифрового оборудования"[1]. Этот документ явно указывает на важность внедрения цифровых технологий, включая программные продукты и веб-приложения, в различные отрасли экономики. Такое внедрение призвано повысить эффективность и конкурентоспособность предприятий. В контексте оптовых складов, создание приложения для автоматизации становится не только актуальным, но и стратегически важным шагом в направлении цифровизации и оптимизации бизнес-процессов.

Целью курсового проекта является создание приложения, которое позволит оптовым складам эффективно управлять товарами и процессами внутри предприятия. В основе этой автоматизации лежит система, включающая в себя роли администратора, рабочего и пользователя с возможностью распределения ролей внутри приложения.

С учетом растущего числа клиентов и необходимости обеспечить оперативную обработку товаров на складе, разработка приложения для оптимизации склада представляет собой актуальную задачу. Приложение предназначено прежде всего для внутреннего использования, облегчая процесс работы сотрудников и повышая эффективность бизнеса в целом.

Анализируя текущие тенденции в сфере сбыта товаров и учитывая потребности современного бизнеса в автоматизации и оптимизации процессов, разработка веб-приложения для оптовых складов является важным шагом в направлении совершенствования предоставляемых услуг и улучшения внутренних рабочих процессов.

Для достижения цели создания приложения для автоматизации оптового склада необходимо выполнить следующие задачи:

1) провести исследование среди сотрудников складов для выявления основных потребностей и требований к функционалу приложения. Это позволит точно определить основные функции и возможности, необходимые для оптимального управления заказами и процессами внутри предприятия;

2) на основе результатов анализа разработать дизайн и структуру приложения, определив функциональные модули и интерфейсные элементы. Важно учесть удобство использования приложения и его интуитивную понятность для пользователей;

3) реализовать основной функционал приложения, включая возможности создания, удаления товаров администратором, а также их изменение сотрудниками;

					09.03.01.000021 П.552 ПЗ	Лист
						4
Изм.	Лист	№ докум.	Подпись	Дата		

1. Анализ предметной области

Предприятия торговли товарами представляет собой значительный сегмент экономики, сфокусированный на контроле товаров внутри предприятия. Оптовые склады являются ключевыми игроками в этой отрасли, предоставляя широкий спектр услуг, начиная с торговли между юридическими лицами и заканчивая торговлей больших объемов между предприятиями-гигантами. Они обычно оснащены специализированным оборудованием и имеют квалифицированный персонал, обладающий необходимыми знаниями и навыками для проведения различных работ внутри этой сферы.

Существующие методы управления и обработки заказов на оптовых складах могут варьироваться в зависимости от их размера и организационной структуры. В некоторых небольших склада процессы могут осуществляться вручную с помощью бумажных журналов или записных книжек, в то время как в крупных предприятиях могут использоваться компьютерные программы или электронные системы учета. Однако, в большинстве случаев, наблюдаются определенные недостатки в этих методах, такие как неоптимальное использование времени, задержки в обработке товаров, риск ошибок и сложности в управлении информацией. Также может отсутствовать структурированная система управления пользовательской базой данных, что затрудняет оперативное реагирование на обработку товаров и координацию работы между сотрудниками. Эти проблемы могут оказывать негативное воздействие на эффективность и производительность склада в целом, и именно поэтому существует необходимость в разработке более современных и эффективных методов управления информацией внутри предприятия.

Одной из основных проблем, с которой сталкиваются оптовые склады, является неоптимальное управление процессами обработки товаров. В некоторых случаях товары могут теряться или оставаться необработанными из-за неструктурированных систем управления. Это может приводить к недовольству клиентов, потере доходов и даже потере репутации предприятия. Другими проблемами являются задержки в обработке товаров из-за неоптимального распределения ресурсов, недостаточная прозрачность и управление информацией, а также высокий уровень риска ошибок при ручной обработке заказов.

Для полноценного понимания ситуации необходимо тщательно исследовать потребности и требования как внутренних пользователей (сотрудников и администраторов склада), так и внешних пользователей. Это включает в себя изучение их рабочих процессов, потребностей в информации и инструментах, а также ожиданий от системы управления заказами. Например, сотрудники могут нуждаться в простом и интуитивно понятном интерфейсе для управления товарами, а клиенты могут ожидать быстрого доступа к. Исследование потребностей пользователей поможет определить ключевые функциональные требования к будущему веб-приложению и обеспечить его максимальную эффективность и удобство использования.

Разработка такого приложения способствует укреплению технологической самодостаточности страны и стимулирует рост отечественного IT-рынка. Кроме

					09.03.01.000021 П.552 ПЗ	Лист
						5
Изм.	Лист	№ докум.	Подпись	Дата		

того, использование отечественного программного обеспечения повышает безопасность и надежность информационных систем, так как обеспечивает больший контроль над данными. Это особенно важно для предприятий, работающих с конфиденциальными данными клиентов.

Важным аргументом также является экономическая выгода от использования отечественного ПО. Внедрение отечественных разработок позволяет сократить расходы на лицензионные платежи и обслуживание, что способствует снижению издержек для предприятий и повышению их конкурентоспособности на рынке.

Также следует отметить соответствие отечественного ПО законодательству, что позволяет предприятиям быть в соответствии с требованиями законодательства о защите данных и персональной информации.

					09.03.01.000021 П.552 ПЗ	Лист
						6
Изм.	Лист	№ докум.	Подпись	Дата		

2. Разработка технического задания

1) Введение

В современном мире оптовые склады играют важную роль в экономике. Оптимизация процессов управления товарами и информацией о них является ключевым аспектом повышения эффективности работы оптовых складов. Данное техническое задание разработано с целью создания веб-приложения для автоматизации работы оптовых складов.

2) Основания для разработки

Разработка веб-приложения обусловлена необходимостью современных инструментов для управления товарами и пользовательской базой данных. Учитывая растущую конкуренцию и требования рынка, внедрение автоматизированных систем является необходимым шагом для повышения конкурентоспособности.

3) Назначение разработки

Целью разработки веб-приложения для автоматизации является создание инструмента, который поможет складам оптимизировать процессы управления товарами и повысить эффективность работы внутри организации. Приложение будет разработано для внутреннего использования сотрудниками складов, а также будет содержать функционал для пользователей.

Основной целью приложения будет обеспечение удобного и эффективного инструмента управления товарами сотрудниками оптовых складов. Приложение позволит сотрудникам и администраторам быстро создавать новые товары в базе данных предприятия, а также управлять ими.

4) Требования к программе или программному изделию

Веб-приложение для оптовых складов должно соответствовать ряду функциональных, интерфейсных и безопасностных требований, чтобы обеспечить удобное и эффективное использование сотрудниками. Ниже перечислены основные требования к приложению:

После запуска программы пользователю отображается форма ввода логина и пароля.

В системе существует всего 4 вида пользователя — Администратор (ADMIN), Рабочий (WORKER), Пользователь (USER) и неавторизованный пользователь. Программа проверяет тип пользователя и открывает соответствующий интерфейс.

Для работника программа предоставляет следующие возможности:

- просмотр каталога товаров;
- поиск по имени в каталоге товаров;
- поиск по фильтрам бренда и типа каталога товаров;
- просмотр содержания определенного товара;
- изменение остатка на складе определенного товара;
- выход из учетной записи.

Для администратора программа предоставляет следующие возможности:

- просмотр каталога товаров;

					09.03.01.000021 П.552 ПЗ	Лист
						7
Изм.	Лист	№ докум.	Подпись	Дата		

- поиск по имени в каталоге товаров;
- поиск по фильтрам бренда и типа каталога товаров;
- просмотр содержания определенного товара;
- добавление нового товара;
- добавление нового типа товара;
- добавление нового бренда товара;
- удаление определенного товара;
- удаление определенного типа товара;
- удаление определенного бренда товара;
- установление роли (Администратор, Рабочий, Пользователь) для

определенного пользователя.

Для пользователя программа предоставляет следующие возможности:

- просмотр каталога товаров;
- поиск по имени в каталоге товаров;
- поиск по фильтрам бренда и типа каталога товаров;
- просмотр содержания определенного товара;
- добавление одного или нескольких товаров в корзину;
- просмотр корзины.

Для незарегистрированного пользователя программа предоставляет следующие возможности:

- просмотр каталога товаров;
- поиск по имени в каталоге товаров;
- поиск по фильтрам бренда и типа каталога товаров;
- просмотр содержания определенного товара;

4.1) Функциональные требования:

- приложение должно обеспечивать возможность создания новых товаров в базе предприятия с указанием необходимых параметров;
- сотрудники должны иметь возможность изменения количества товара на складе и количества минимальной продажи товара;
- сотрудники должны иметь возможность просматривать существующие товары;
- должна быть предусмотрена авторизация, для предотвращения несанкционированного доступа
- администраторы должны иметь возможность изменять роли пользователей.

4.2) Интерфейсные требования:

Приложение должно иметь простой и понятный интерфейс, обеспечивающий удобное взаимодействие с пользователем.

4.3) Безопасностные требования:

Приложение должно обеспечивать безопасную аутентификацию пользователей и управление доступом к функционалу на основе их ролей и прав.

4.4) Совместимость:

Программа должна быть совместимой с различными операционными системами и базами данных, используемыми на предприятии.

					09.03.01.000021 П.552 ПЗ	Лист
						8
Изм.	Лист	№ докум.	Подпись	Дата		

5) Техничко-экономические показатели

Веб-приложение для оптовых складов позволит оптимизировать процессы управления товарами на складе, уменьшить время на обработку информации о товарах и снизить вероятность ошибок, связанных с ручным вводом данных. Это приведет к увеличению производительности сотрудников.

В целом, веб-приложение для оптовых складов позволит снизить издержки на обработку товаров, повысить эффективность использования ресурсов и увеличить объемы продаж за счет улучшения качества обработки информации внутри предприятия. Это приведет к увеличению выручки и прибыли оптового склада в долгосрочной перспективе.

6) Стадии и этапы разработки

Разработка должна быть проведена в две стадии:

- 1) техническое задание;
- 2) технический (и рабочий) проекты;

На стадии «Техническое задание» должен быть выполнен этап разработки, согласования и утверждения настоящего технического задания.

На стадии «Технический (и рабочий) проект» должны быть выполнены перечисленные ниже этапы работ:

- разработка программы;
- разработка программной документации.

Содержание работ по этапам:

На этапе разработки технического задания должны быть выполнены перечисленные ниже работы:

- постановка задачи;
- определение и уточнение требований к техническим средствам;
- определение требований к программе;
- определение стадий, этапов и сроков разработки программы и документации на нее;
- согласование и утверждение технического задания.

На этапе разработки программы должна быть выполнена работа по программированию (кодированию) и отладке программы.

На этапе испытаний программы должны быть выполнены перечисленные ниже виды работ:

- разработка, согласование и утверждение порядка и методики испытаний;
- проведение приемо-сдаточных испытаний;
- корректировка программы и программной документации по результатам испытаний.

На этапе подготовки и передачи программы должна быть выполнена работа по подготовке и передаче программы и программной документации в эксплуатацию на объектах заказчика.

7) Порядок контроля и приемки

Приемосдаточные испытания программы должны проводиться согласно разработанной исполнителем и согласованной заказчиком «Программы и методики испытаний».

					09.03.01.000021 П.552 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		9

Ход проведения приемо-сдаточных испытаний заказчик и исполнитель документируют в протоколе испытаний.

На основании протокола испытаний исполнитель совместно с заказчиком подписывают акт приемки-сдачи программы в эксплуатацию.

					09.03.01.000021 П.552 ПЗ	Лист
						10
Изм.	Лист	№ докум.	Подпись	Дата		

3. Выбор используемых технологий

Были проанализированы современные технологии и инструменты для разработки веб-приложений, фокусируясь на фронтенд-части. В исследовании рассматривались языки программирования, такие как JavaScript, фреймворки для разработки, включая React, библиотеки для работы с запросами и состояниями, такие как Axios и MobX, а также инструменты маршрутизации, например, React Router Dom, и библиотеки стилизации, например, Bootstrap.

Проведён сравнительный анализ разных технологических стеков, учитывая их эффективность, удобство использования, степень поддержки и совместимость. Особое внимание уделялось выбору инструментов для создания высококачественных и эффективных веб-приложений.

Каждая технология была тщательно проанализирована с точки зрения ее преимуществ и недостатков в контексте конкретных требований проекта. Анализ включал оценку возможностей интеграции выбранных технологий с другими системами и платформами, гибкости разработки и расширяемости приложений, средств обеспечения безопасности данных и пользователей, а также уровня поддержки со стороны сообщества разработчиков. Особое внимание уделялось возможностям снижения времени и трудозатрат на разработку, а также повышению качества конечного продукта.

В качестве языка был выбран JavaScript:

1) широкая поддержка и кросс-браузерная совместимость.

JavaScript является единственным языком программирования, который поддерживается всеми современными веб-браузерами, что делает его универсальным инструментом для разработки фронтенда веб-приложений. Это обеспечивает кросс-браузерную совместимость и гарантирует, что приложения будут корректно отображаться и работать во всех браузерах

2) богатая экосистема библиотек и фреймворков.

JavaScript обладает обширной экосистемой, включающей популярные фреймворки и библиотеки, такие как React, Vue, Angular, Axios, MobX и другие. Эти инструменты упрощают процесс разработки, позволяют ускорить создание интерфейсов и обеспечить эффективное управление состоянием и данными приложения.

3) динамичность и интерактивность.

JavaScript предоставляет широкие возможности для создания динамичных и интерактивных веб-приложений. Это позволяет разрабатывать современные пользовательские интерфейсы с анимацией, обработкой событий и AJAX-запросами для улучшения пользовательского опыта.

4) поддержка современных стандартов.

JavaScript постоянно развивается и обновляется, следуя современным стандартам ECMA, что позволяет использовать новейшие возможности и синтаксис языка для улучшения качества кода и производительности приложений.

5) эффективность работы в браузере.

JavaScript оптимизирован для выполнения в браузере и предоставляет эффективные механизмы обработки запросов, событий и операций, что позволяет создавать быстро работающие и отзывчивые приложения

б) возможность работы на серверной части.

Кроме использования на клиентской части, JavaScript можно применять для серверной разработки с помощью платформы Node.js, что обеспечивает единый язык для всей разработки.

Были исследованы современные технологии и инструменты для разработки бэкенд-части веб-приложений. В исследовании рассматривались платформы для серверной разработки, такие как Node.js, базы данных, включая PostgreSQL, инструменты для управления базами данных, такие как pgAdmin, а также ORM-библиотеки для работы с базами данных, например, Sequelize. Помимо этого, анализу подвергались фреймворки для создания серверного API, такие как Express.

Был проведен сравнительный анализ разных стилей бэкенд-разработки, учитывая их эффективность, совместимость, удобство использования и степень поддержки. Особое внимание уделялось выбору инструментов для обеспечения высокой производительности, безопасности и легкости масштабирования веб-приложений.

Каждая из выбранных технологий была оценена с точки зрения их преимуществ и недостатков в контексте конкретных требований проекта. Это включало оценку возможностей интеграции, производительности, обеспечения безопасности данных и пользователей, а также уровня поддержки и доступности ресурсов от сообществ разработчиков.

					09.03.01.000021 П.552 ПЗ	Лист
						12
Изм.	Лист	№ докум.	Подпись	Дата		

4. Проектирование и разработка БД

4.1. Анализ предметной области и информационных задач пользователей

Целью веб-приложения является автоматизация работы оптового склада и обеспечение эффективного управления товарами. Приложение предназначено для использования сотрудниками склада, предоставляя небольшую функциональность для клиентов. Механики нуждаются в простом способе контроля информации о товарах, а также в возможности предоставления информации о товарах клиентам.

В рамках анализа выявляется, что для эффективного управления товарами необходимо реализовать гибкость изменения информации о товарах. Поэтому центральной сущностью является "товар" – единица внутри базы предприятия, создаваемая администратором и доступный для изменения рабочим. Каждый товар содержит информацию о количестве на складе, минимальном количестве для продажи, цене, а также содержит информацию о бренде и типе для удобного поиска по фильтрам.

Функциональность приложения:

- создание новых товаров администраторами;
- удаление товаров администраторами;
- создание новых типов администраторами;
- удаление типов администраторами;
- создание новых брендов администраторами;
- удаление брендов администраторами;
- изменение ролей пользователей администраторами;
- изменение доступных товаров для работников;
- просмотр доступных товаров пользователями;
- поиск по фильтрам доступных товаров;
- поиск по имени доступных товаров;
- добавление товаров в корзину;
- просмотр корзины.

В основе деятельности оптового склада лежит обработка товаров, поступающих на склад. Основными пользователями системы являются администраторы, рабочие и клиенты.

1) пользователи представляют собой обычных людей, желающих просмотреть доступный товар;

2) рабочие выполняют контроль над количеством товаров на складе, а также над количеством, минимальным для продажи;

3) администраторы управляют работой системы и контролируют добавление и удаление товаров, типов и брендов, также осуществляют контроль над ролями пользователей.

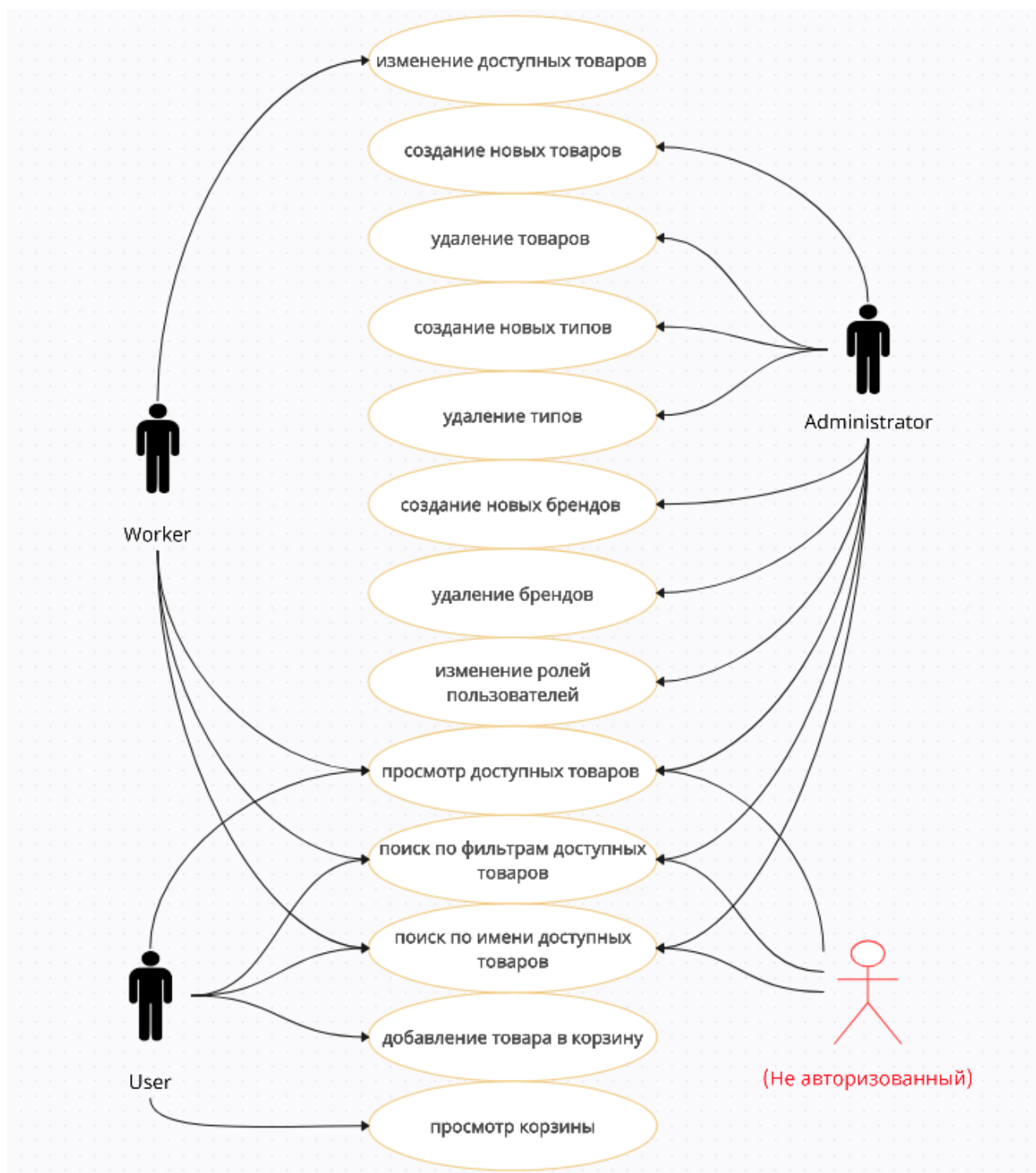


Рисунок 1 – Диаграмма вариантов использования

4.2. Формирование схемы данных

ER-диаграмма является важным инструментом в проектировании баз данных, поскольку она позволяет наглядно представить структуру данных и их

взаимосвязи. В контексте моей курсовой работы, ER-диаграмма необходима для того, чтобы я мог более глубоко понять связи и зависимости между различными сущностями в базе данных. Она поможет мне четко определить сущности (таблицы) и их атрибуты, а также установить связи между этими сущностями.

ER-диаграмма позволяет:

- 1) идентифицировать основные сущности и их атрибуты, что позволит мне более точно определить необходимую структуру базы данных;
- 2) установить связи между сущностями, что поможет мне понять логику хранения и связей данных в базе;
- 3) визуализировать сложные связи и зависимости между различными сущностями, что облегчит понимание структуры базы данных для меня и для других пользователей.

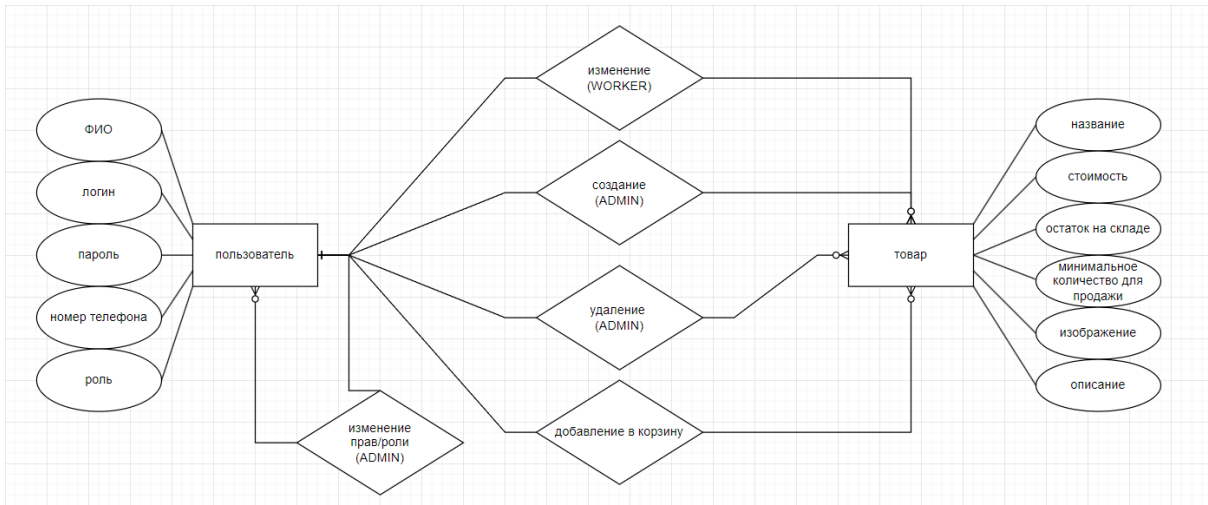


Рисунок 2 – ER-диаграмма

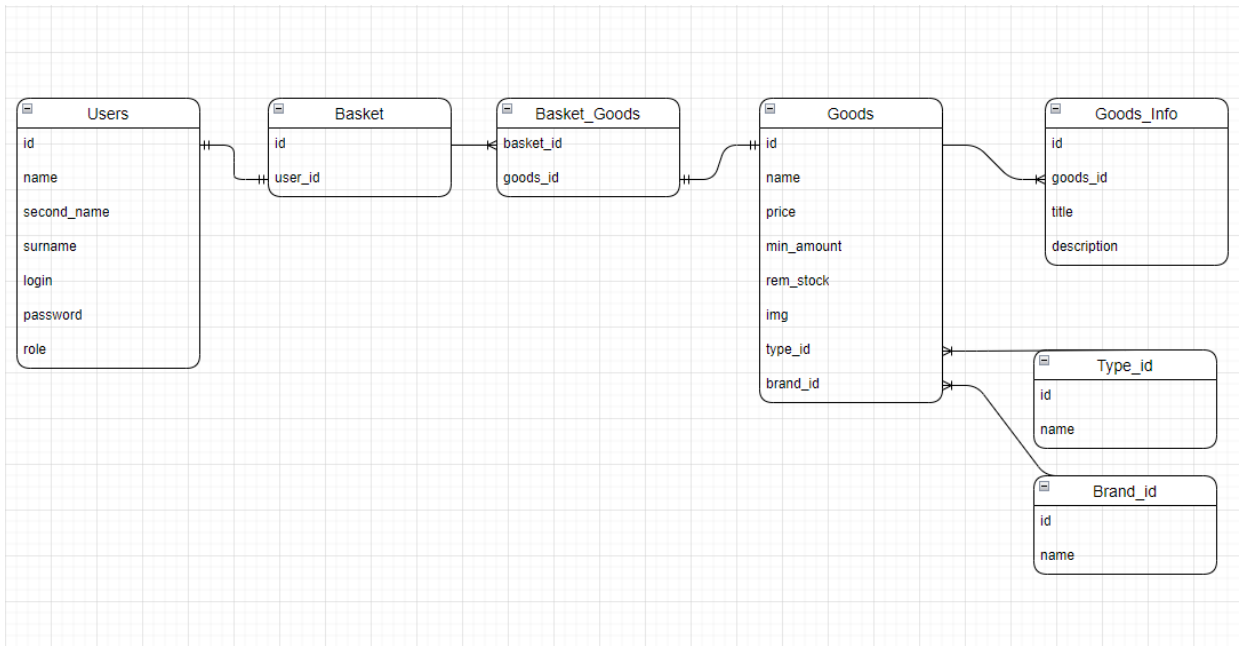


Рисунок 3 – Диаграмма базы данных

Добавление пользователей в базу происходит на странице регистрации. В случае, если пользователь с такими логином уже существует, система отправит уведомление на клиент.

Проводить добавление и удаление товаров, а также выдачу ролей может только администратор.

Проводить изменение товаров может только рабочий.

Для создания связи используется библиотека sequelize, которая связывает данные по id из таблиц:

```
Users.hasOne(Basket)
Basket.belongsTo(Users)

Basket.hasMany(BasketGoods)
BasketGoods.belongsTo(Basket)

Goods.hasMany(BasketGoods)
BasketGoods.belongsTo(Goods)

Type.hasMany(Goods)
Goods.belongsTo(Type)

Brand.hasMany(Goods)
Goods.belongsTo(Brand)

Goods.hasMany(GoodsInfo, {as: 'info'});
GoodsInfo.belongsTo(Goods)
```

Рисунок 4 – Связь между таблицами

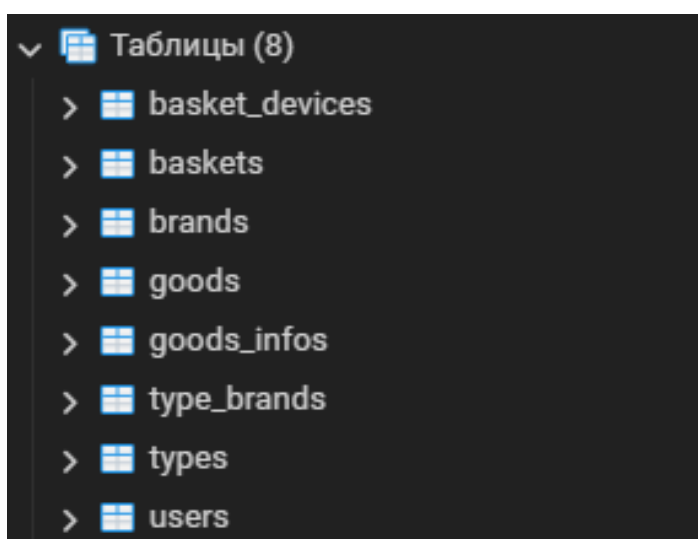


Рисунок 5 – Итоговый вид базы данных в СУБД

4.3. Выбор СУБД и других программных средств

Для реализации веб-приложения для оптового склада предлагается использовать реляционную базу данных. Учитывая, что приложение предназначено для внутреннего использования сотрудниками и не предполагает высокой нагрузки, реляционная база данных будет наиболее подходящим выбором.

Выбор СУБД может включать в себя такие варианты, как MySQL или PostgreSQL, в зависимости от конкретных потребностей и требований к производительности и масштабируемости. Обе эти СУБД обладают широкими возможностями и хорошо подходят для разработки приложений в небольших и средних компаниях.

4.4. Описание групп пользователей и прав доступа

В приложении можно выделить следующие группы пользователей и их права доступа:

- администратор. Может просматривать каталог товаров, осуществлять поиск по имени и фильтрации товаров, а также добавлять и удалять товары, бренды и типы, выдавать роли определенным пользователям;
- рабочий. Может просматривать каталог товаров, осуществлять поиск по имени и фильтрации товаров, а также изменять характеристики определенных товаров;
- пользователь. Может просматривать каталог товаров, осуществлять поиск по имени и фильтрации товаров, а также добавлять их в корзину;
- неавторизованный пользователь. Может просматривать каталог товаров, осуществлять поиск по имени и фильтрации товаров;

5. Разработка блок-схем

Для веб-приложения по автоматизации оптового склада были разработаны следующие блок-схемы:

Блок схема на рисунке 6 отображает процесс авторизации пользователя в системе.

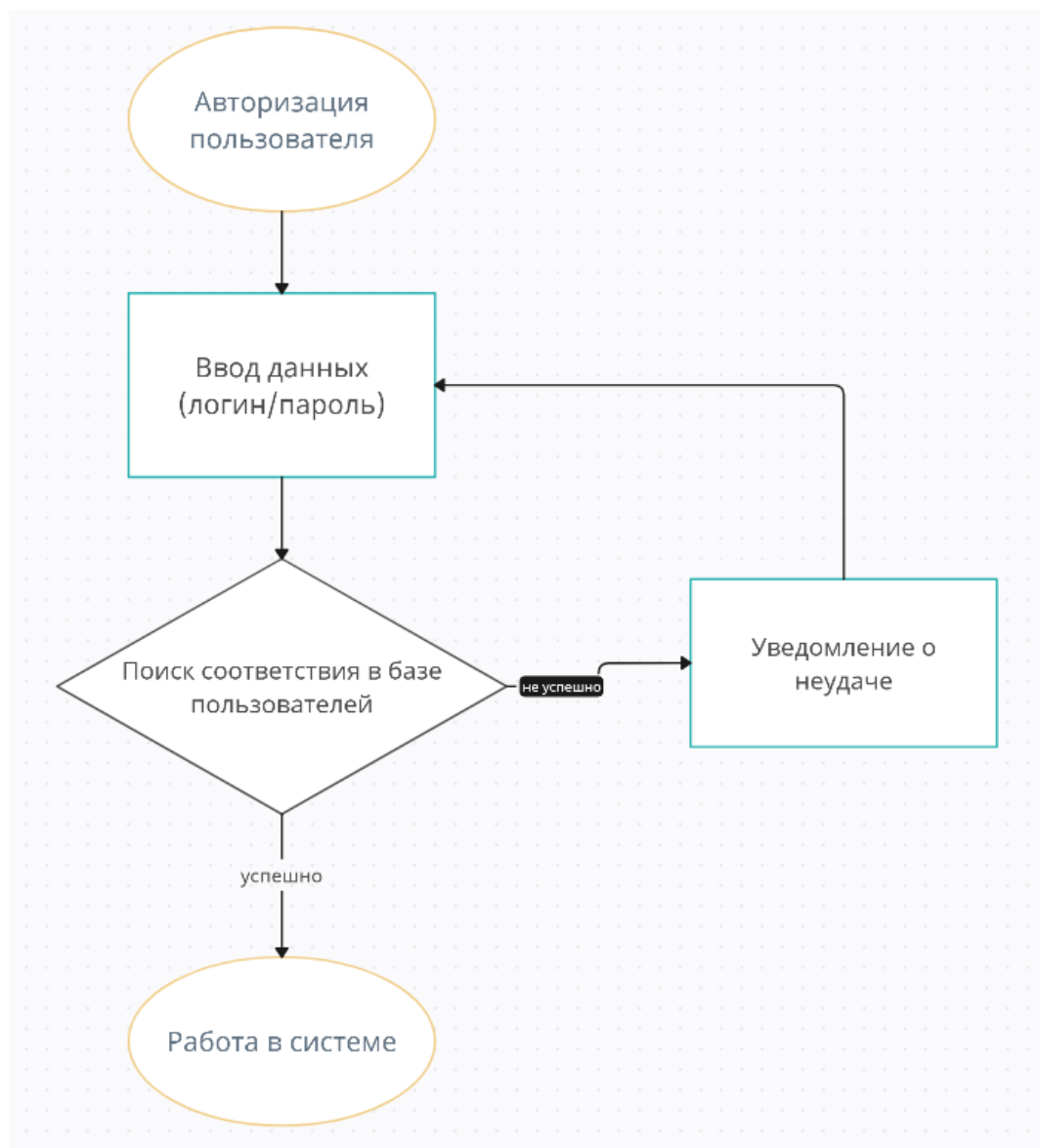


Рисунок 6 – Авторизация сотрудника в системе

Блок схема на рисунке 7 отображает процесс добавление товара в базу данных. Если введена не вся информация о товаре, то появится уведомление о том, что нужно заполнить все поля, в ином случае товар добавиться в базу данных.

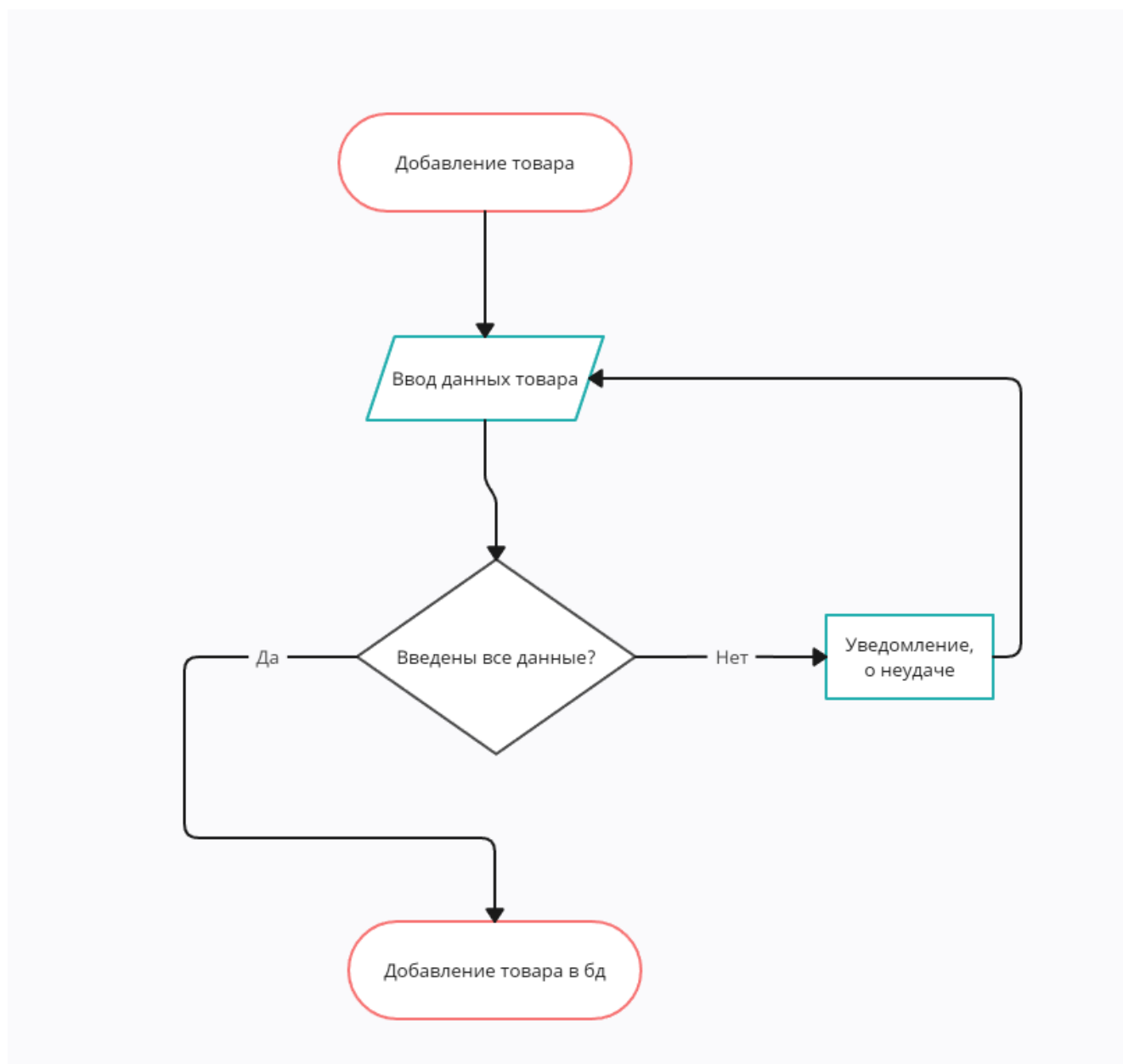


Рисунок 7 – Добавление запроса администратором

На блок-схеме, представленной на рисунке 8, отображён процесс удаления товара из базы данных, вначале сотрудник ищет товар, который он хочет удалить, после вводит количество товара для удаления, в конце сотрудник удаляет выбранное количества товара из базы данных.

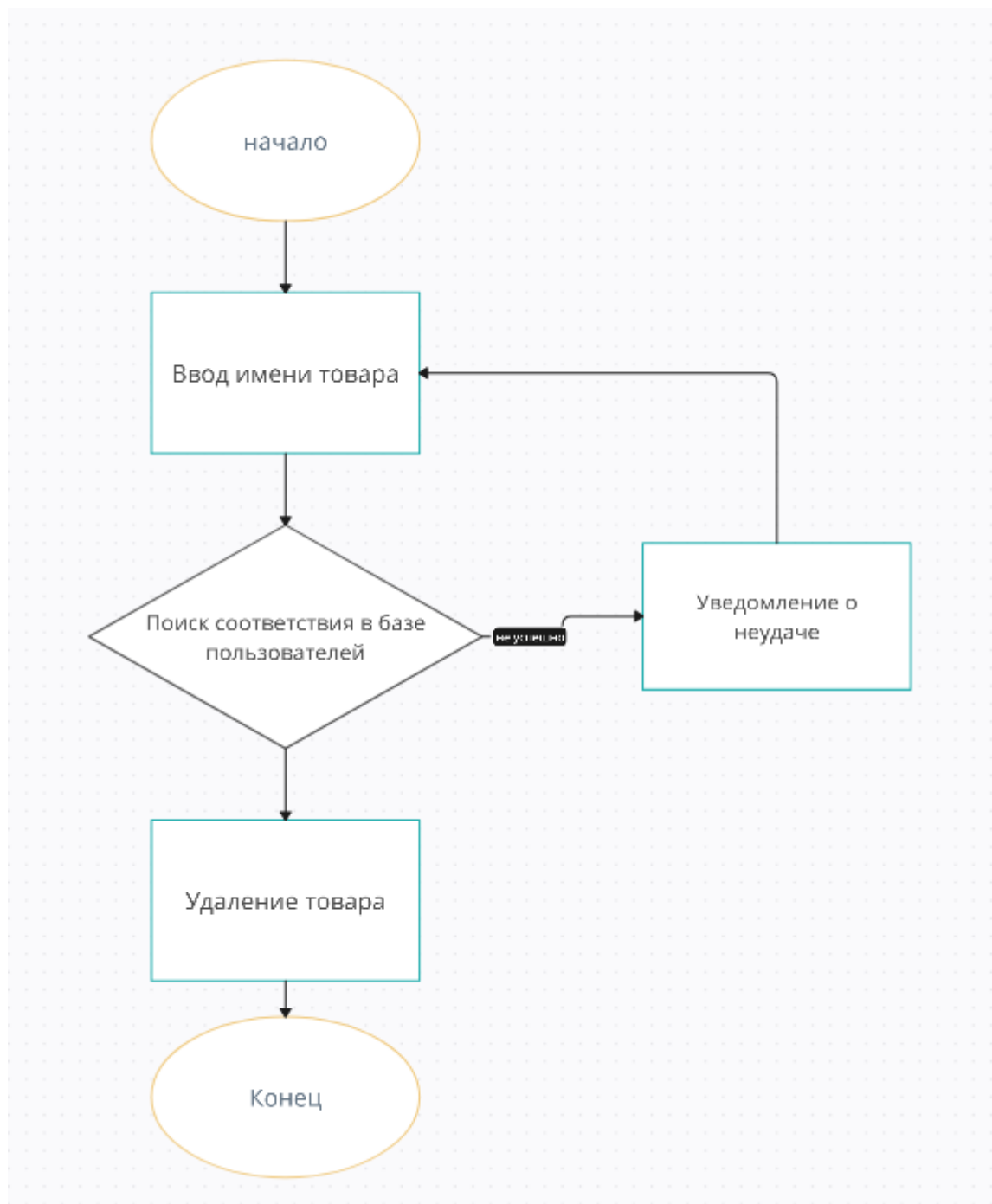


Рисунок 8 – Удаление товара из БД

Все блок-схемы отражают возможное взаимодействие с приложением.

6. Разработка интерфейса ПО

6.1. Проектирование интерфейса

Главное меню должно обеспечить доступ просмотру и поиску всех товаров. Самым лучшим вариантом, чтобы не нагружать дизайн – оставить общий список по середине с поиском сверху и навигационной панель. слева.

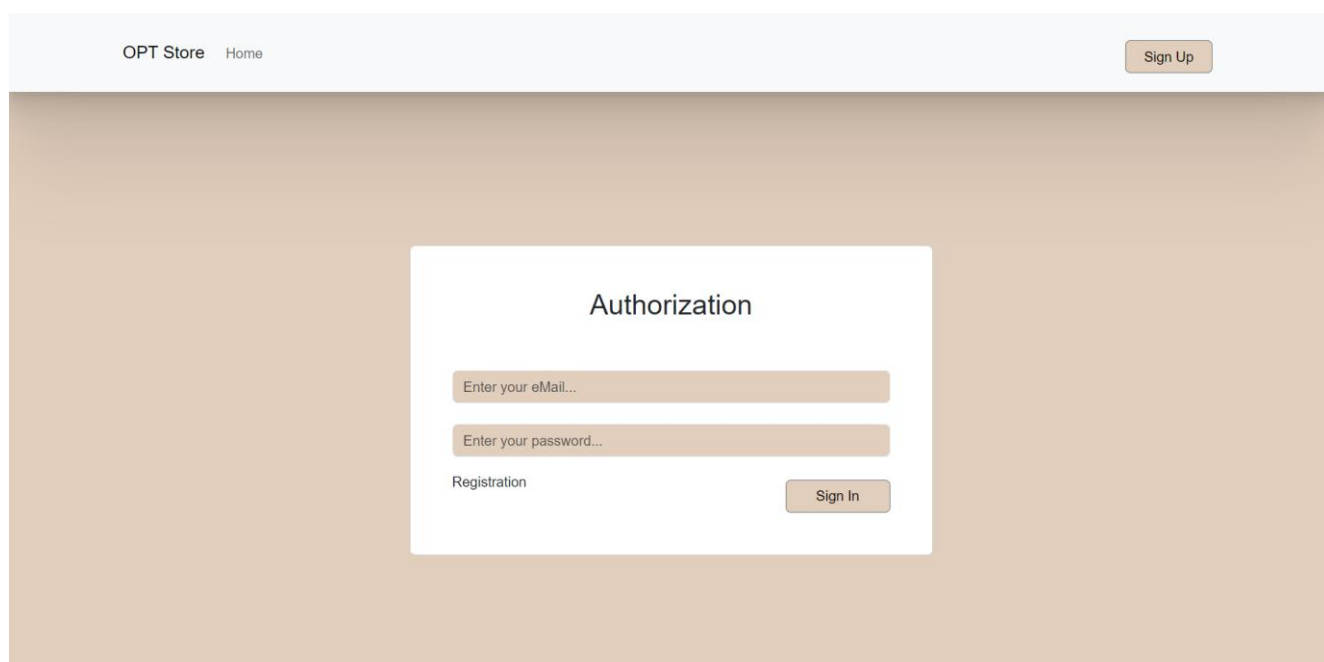


Рисунок 9 – Страница авторизации

					09.03.01.000021 П.552 ПЗ	Лист
						21
Изм.	Лист	№ докум.	Подпись	Дата		

OPT Store

Home

Sign Up

Registration

worker

.....

Enter your name...

Enter your second name...

Enter your surname...

Enter your phone number...

Sign In

Sign Up

Рисунок 10 – Страница регистрации

OPT Store

Home

Log Out


Navigation

--Select Type--


--Select Brand--

123


Enter name...




Hyper 3000



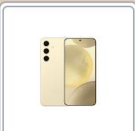
Hyper 5000




Asus Z




Asus Flop




GalaxyS24 Yel



GalaxyS24 Pur



iPhone 12 Pro Bl



LG Power

Рисунок 11 – Главная страница

					09.03.01.000021 П.552 ПЗ	Лист
						22
Изм.	Лист	№ докум.	Подпись	Дата		

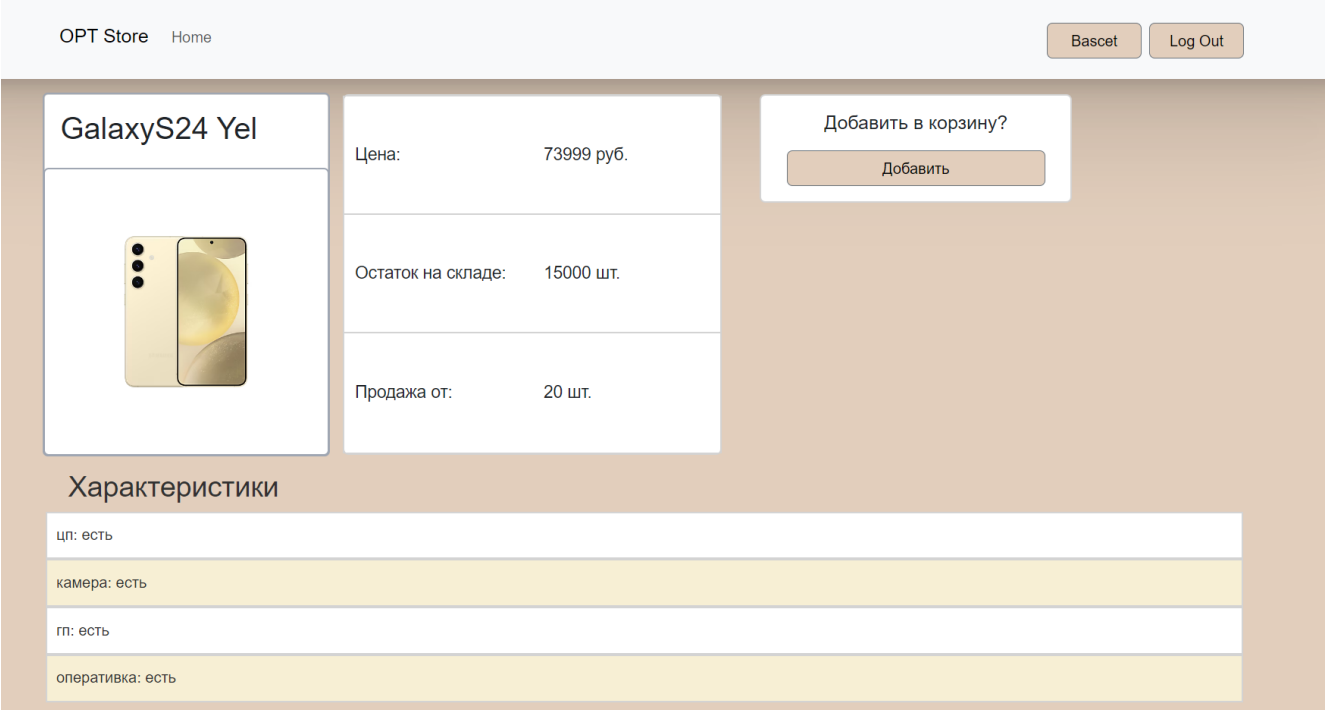


Рисунок 12 – Страница товара для пользователя

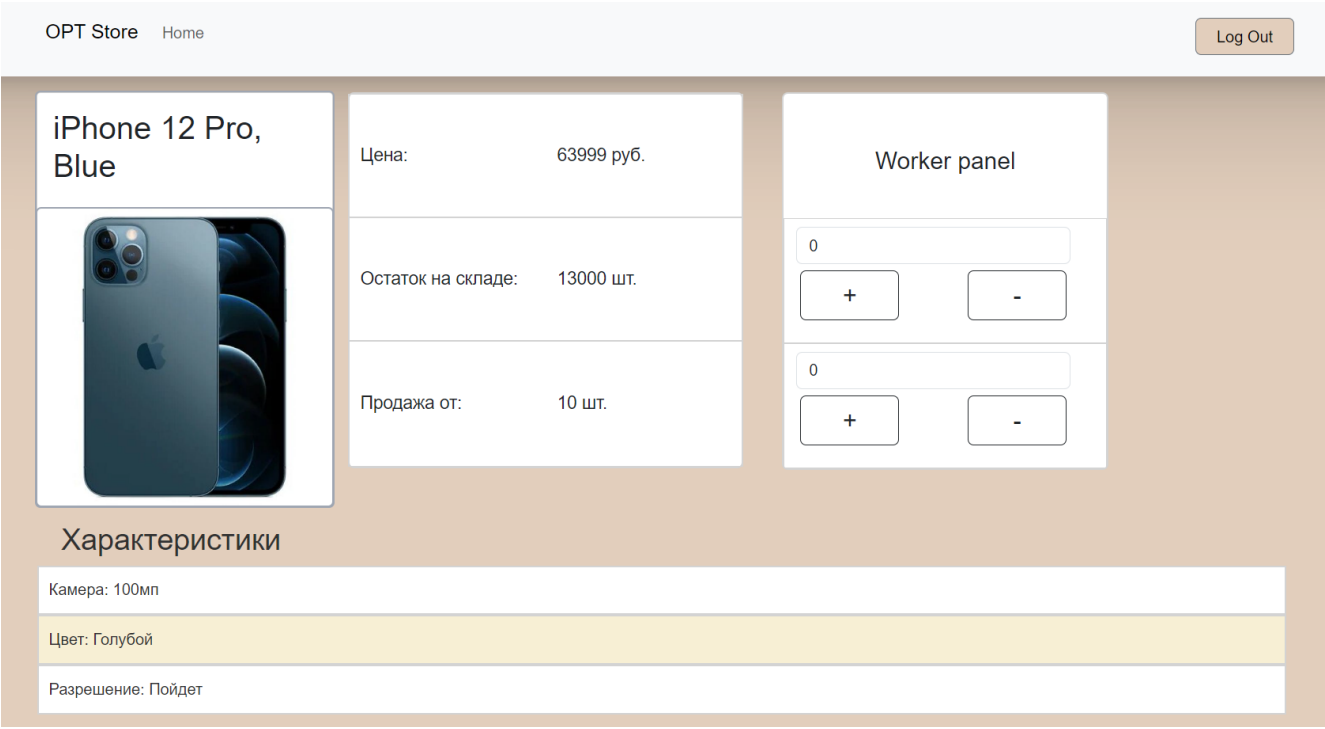


Рисунок 13 – Страница товара для рабочего

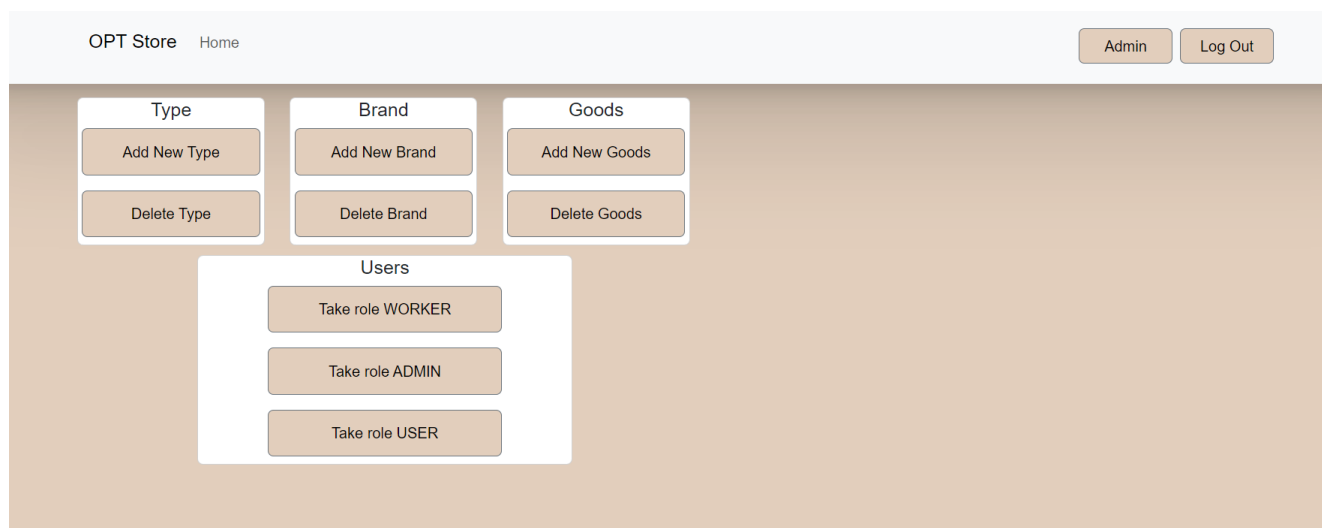


Рисунок 14 – Панель управления администратора

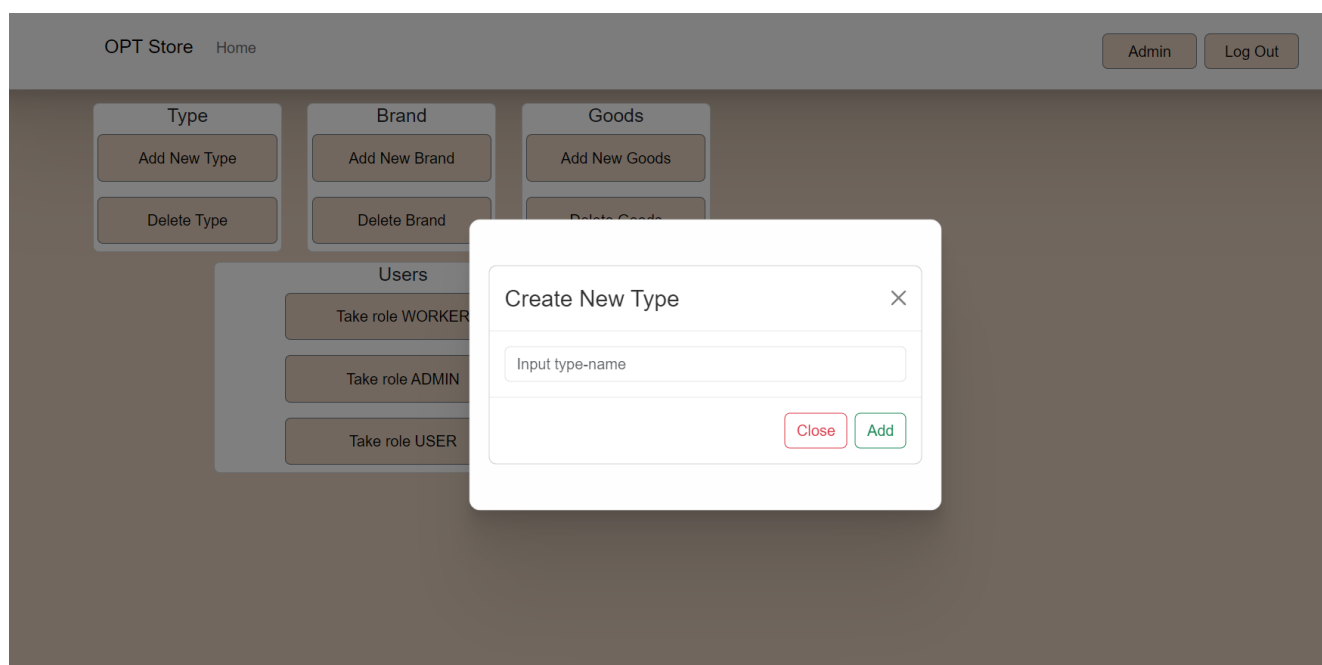


Рисунок 15 – Всплывающие окна панели администратора

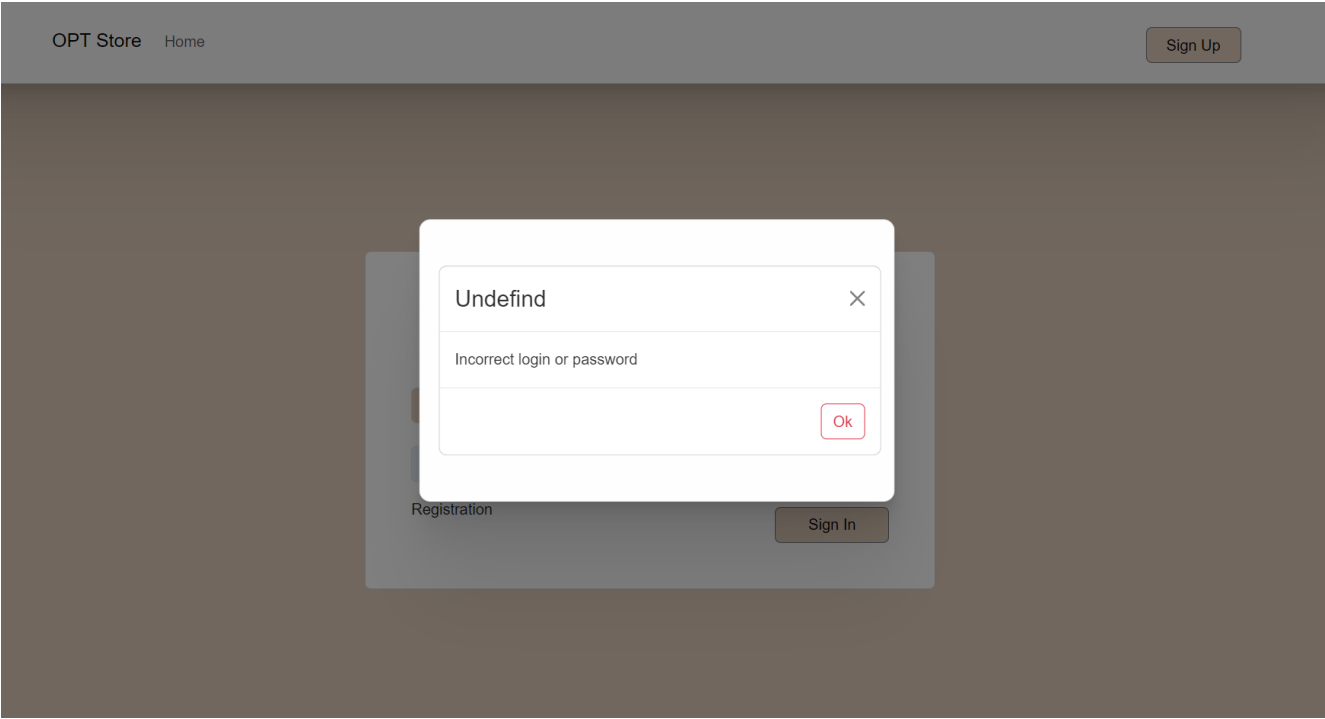


Рисунок 16 – Ошибка при входе

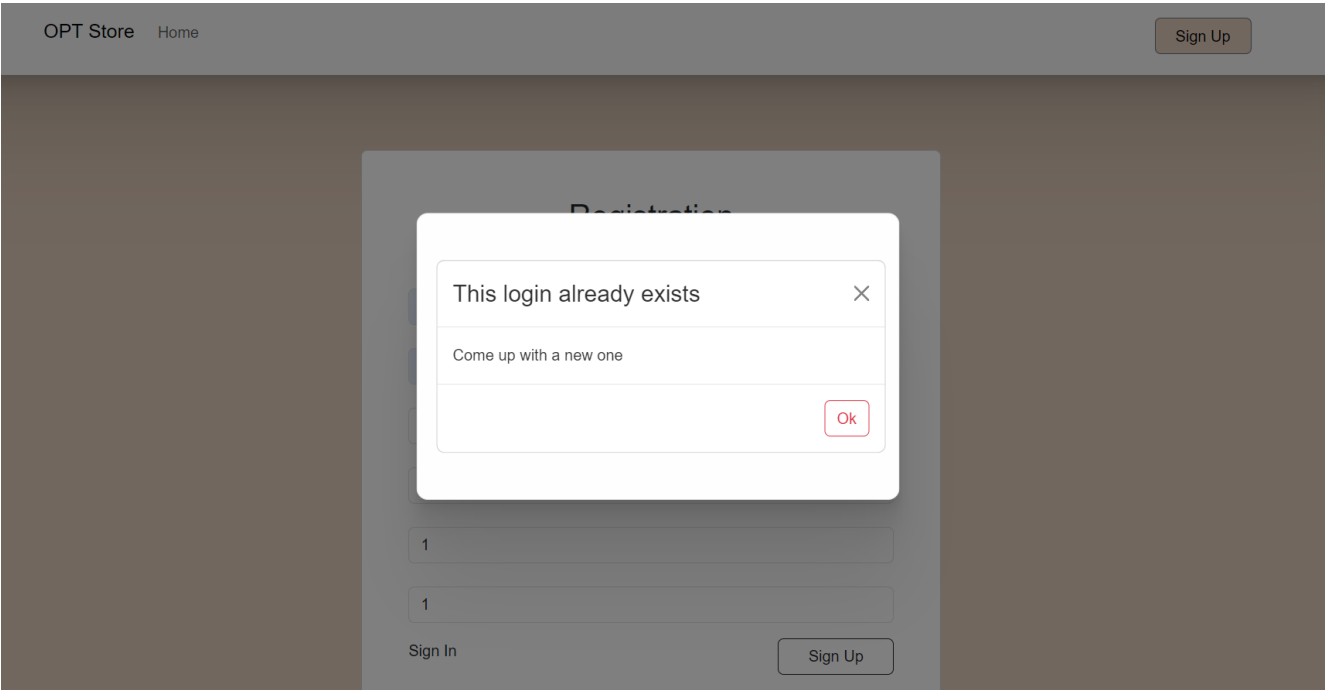


Рисунок 17 – Ошибка при регистрации

7. Информационная безопасность

Аутентификация и авторизация в приложении для оптового склада играют ключевую роль в обеспечении безопасности данных и доступа к функционалу. Для этого реализованы механизмы, обеспечивающие идентификацию пользователей и проверку их прав доступа.

Система аутентификации предполагает использование уникального логина и пароля для каждого пользователя. Эти учетные данные являются ключом к доступу в приложение и должны быть достаточно сильными, чтобы обеспечить защиту от несанкционированного доступа. Важным аспектом здесь является обучение пользователей правилам безопасности паролей и предоставление рекомендаций по созданию надежных комбинаций символов.

В качестве логина будет использоваться уникальный набор символов (логин) пользователя. А для проверки пользователя в базе данных используется код, представленный на рисунке 18.

```
async login (req, res, next) {  
  const {email, password} = req.body  
  const user = await Users.findOne({where: {email}})  
  if (!user) {  
    return next(ApiError.internal('Пользователь не найден'))  
  }  
  let comparePassword = bcrypt.compareSync(password, user.password)  
  if(!comparePassword){  
    return next(ApiError.internal('Указан неверный пароль'))  
  }  
  const token = generateJwt(user.id, user.email, user.role)  
  return res.json({token})  
}
```

Рисунок 18 – Проверка

Второй аспект — это ограничение доступа к конфиденциальным данным. Это достигается путем назначения ролей и привилегий каждому пользователю в системе в соответствии с его должностными обязанностями. Механизмы авторизации обеспечивают проверку прав доступа каждого пользователя к различным функциям и данным приложения.

Принцип наименьших привилегий гарантирует, что пользователи имеют доступ только к тем данным и функциям, которые необходимы для выполнения их работы. Это позволяет минимизировать риски утечки конфиденциальной информации и предотвращать несанкционированный доступ к данным.

Заключение

В результате выполнения курсового проекта было разработано приложение для автоматизации работы оптового склада. Для достижения целей были выполнены следующие задачи:

1) проведенное исследование позволило точно определить основные функциональные требования к разрабатываемому приложению. Полученные данные послужили основой для разработки приложения, учитывая потребности конечных пользователей;

2) на основе результатов анализа был разработан дизайн и структура веб-приложения, обеспечивающие удобство использования и интуитивную понятность для пользователей. Это позволит сотрудникам автомастерских эффективно управлять заказами и процессами внутри предприятия;

3) основной функционал приложения был успешно реализован, включая возможности создания и отслеживания заказов администратором, а также их прием и выполнение сотрудниками;

					09.03.01.000021 П.552 ПЗ	Лист
						27
Изм.	Лист	№ докум.	Подпись	Дата		

Библиография

1. Федеральный закон от 09.05.2017 № "О Стратегии развития информационного общества в Российской Федерации на 2017 - 2030 годы"
2. Годин В.В. Базы данных. Проектирование / В.В. Годин. – Москва : Юрайт, 2017. – 292 с.
3. Ёранссон Андерс Эффективное использование потоков в операционной системе / Андерс Ёранссон. – Москва : ДМК-Пресс, 2018. – 312 с. – ISBN 978-5-97060-168-6
4. Михайлов, Д. М. Защита мобильных телефонов от атак / Д. М. Михайлов, И. Ю. Жуков. – Москва : Фойлис, 2014. – 192 с. – ISBN 978-5-91860-010-8.
5. Сагаловский, В.А. Программирование / В.А. Сагаловский. – Петербург : БХВ, 2021. – 250 с.

					09.03.01.000021 П.552 ПЗ	Лист
						28
Изм.	Лист	№ докум.	Подпись	Дата		