Изображение выглядит как графическая вставка, Графика, мультфильм

Автоматически созданное описание

**КУРСОВОЙ ПРОЕКТ**

**Тема: Разработка программного модуля системы «Обувной магазин».**

**Специальность 09.02.07 Информационные системы и программирование**

|  |  |  |
| --- | --- | --- |
| **Выполнила студентка группы 307ИС-22** | **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_** | **Д. Р. Симаченко** |
| **Руководитель** | **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_** | **О. А. Калашникова** |

**Москва 2024**

Оглавление

[**Введение** 2](#_Toc185569119)

[**1 АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ** 3](#_Toc185569120)

[**1.1 Предпроектное обследование** 3](#_Toc185569121)

[**1.2 Обзор и анализ существующих программных решений** 4](#_Toc185569122)

[**1.3 Постановка задач к программному продукту** 5](#_Toc185569123)

[**2 ПРОЕКТИРОВАНИЕ И РАЗРАБОТКА МОДУЛЯ** 8](#_Toc185569124)

[**2.1 Эскизное проектирование** 8](#_Toc185569125)

[**2.1.1 Выбор архитектуры** 9](#_Toc185569126)

[**2.1.2 Выбор инструментальных средств реализации** 11](#_Toc185569127)

[**2.2 Проектирование базы данных** 12](#_Toc185569128)

[**2.3 Написание программного кода** 15](#_Toc185569129)

[**3 ОТЛАДКА И ТЕСТИРОВАНИЕ МОДУЛЯ** 17](#_Toc185569130)

[**3.1 Тестирование программного продукта.** 17](#_Toc185569131)

[**3.1.1 Реализация пользовательского интерфейса** 18](#_Toc185569132)

[**3.2 Отладка программного продукта.** 19](#_Toc185569133)

[**Заключение** 20](#_Toc185569134)

# Введение

Всегда актуальной и важной является тема создания эффективных и современных систем управления в розничной торговле. В современном мире сфера торговли обувью занимает значительное место, предлагая широкий ассортимент товаров и услуг. В рамках этого курсового проекта рассматривается создание веб-приложения для управления магазином обуви, с фокусом на эффективное управление товарным ассортиментом, заказами и взаимодействием с клиентами.

Целью данного проекта является разработка программного комплекса, который обеспечит автоматизацию основных процессов управления магазином обуви, начиная от учета товара и заканчивая обработкой заказов и ведением истории покупок. Включение современных технологий разработки позволит создать удобный и функциональный интерфейс для сотрудников магазина и улучшить опыт покупателей.

Проект предполагает использование инструментов веб-разработки, баз данных и методов программирования для создания эффективной платформы, позволяющей управлять продажами, организовывать обработку заказов, а также анализировать предпочтения покупателей для оптимизации бизнес-процессов. Результаты данного исследования и разработки имеют целью не только повысить эффективность работы магазина обуви, но и обеспечить лучший сервис для клиентов в условиях современной розничной торговли.

Таким образом, разработка приложения для магазина обуви позволяет не только улучшить пользовательский опыт и упростить работу для сотрудников магазина, а также внесет значительный вклад в развитие индустрии розничной торговли, позволяя даже неопытным пользователям с легкостью ознакомиться с отраслью интернет-магазинов и упрощая пользование уже существующим клиентам.

# 1 АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ

## 1.1 Предпроектное обследование

Информационная система «Магазин обуви» – это система, предназначенная для управления и отображения информации об обуви, ее приобретения и администрирования магазина. Основные функции системы включают:

Изучение обуви. Пользователи могут просматривать обувь в наличии, ее название, стоимость и приложенные фотографии.

Покупка обуви. Пользователи могут приобретать выбранную обувь, которая в дальнейшем будет в их списке покупок.

Администрирование. Администраторы управляют моделями наличии, добавляют новую, редактируют или удаляют существующие.

Роли в системе:

Пользователь

- Просмотр обуви: Пользователь может изучить всю обувь в наличии, найти конкретную модель, воспользовавшись поиском по названию.

- Приобретение обуви: Пользователь выбирает понравившуюся модель и покупает ее.

- Личный кабинет: Пользователь может просматривать купленную им обувь.

2. Администратор

- Добавление новых моделей обуви: Администратор добавляет новую модель, набирая название, стоимость и фотографии модели.

- Редактирование существующих моделей обуви: Администратор выбирает существующую модель, и редактирует необходимый ему параметр (Название, стоимость, фотографии).

- Удаление модели обуви: По какой-либо причине Администратор может удалить существующую позицию обуви.

Основные функции системы:

1. Просмотр обуви

- Общий список. Отображение всех моделей обуви.

- Поиск конкретной модели. Возможность поиска обуви по названию или цене.

- Детали. Просмотр сведений об обуви, ее внешний вид, цена.

2. Приобретение обуви

- Выбор модели. Пользователь выбирает из перечня необходимую ему модель.

- Оплата. Пользователь покупает выбранную модель.

- Подтверждение. Пользователь подтверждает приобретение выбранной им модели.

3. Администрирование.

- Добавление товаров. Администратор добавляет новые модели обуви.

- Редактирование товаров. Администратор редактирует созданные товарные ячейки.

- Удаление товаров. Администратор может удалить созданные позиции.

## 1.2 Обзор и анализ существующих программных решений

Сайт neylonov.ru представляет собой сервис по заказу одежды и обуви. Он предоставляет пользователям информацию о моделях в наличии, проданных моделях, их стоимости и способы доставки.

Основные функции сайта.

- Выбор типа товара. Пользователи выбирают, какой именно товар им нужен, например верхняя одежда, обувь или аксессуары.

- Информация о товаре. Пользователь может изучить необходимый товар по его краткому описанию, характеристике и отзывам покупателей.

- Заказ товара. Пользователь выбирает нужный товар, добавляет его в корзину, выбирает способ оплаты и доставки.

- Информация об акциях. Пользователь при переходе на сайт первым делом видит всплывающие новости об распродажах, акциях.

- Сортировка товара. Пользователь может сортировать товар по убыванию, возрастанию цены и названию модели.

- Фильтрация товаров. Пользователь может отфильтровать тип товара, бренд, размер, цвет, сезон, цену.

- Подробная информация. Пользователь может подробно изучить все интересующие его аспекты, такие как информация о компании, доставка, обмен и возврат, как оформить заказ, таблица размеров, отзывы.

Сайт имеет удобный и интуитивно понятный интерфейс. Навигация по сайту проста, с быстрым доступом к основным разделам. Обширный и актуальный контент, включающий наличие товара, его описание и характеристики. Регулярные обновления новостей и рецензий.

Функциональность. Возможность онлайн-заказа и покупки товаров. Фильтрация и поиск по различным параметрам, что облегчает пользователям нахождение нужной информации. Интеграция с социальными сетями для обмена информацией и отзывами. Подписка на рассылку новостей и уведомлений о предстоящих релизах.

Сайт nikitaefremov.ru предоставляет информацию о лимитированных моделях обуви и аксессуаров. Основные функции сайта включают:

- Наличие моделей обуви. Пользователь может ознакомиться с моделями обуви в наличии.

- Фильтрация товаров по цене и названию.

- Актуальная информация. Пользователь может ознакомить с актуальными скидками, новыми поступлениями, подарочными сертификатами.

- Подробная информация. Пользователь может ознакомиться с брендом, деталями по доставке, возврату, оплате. Также пользователь может узнать ответы на частые вопросы, изучить отзывы, получить контактную информацию с поддержкой.

## 1.3 Постановка задач к программному продукту

Система «Магазин обуви» должна выполнять ряд функций, обеспечивающих удобство использования для пользователей и эффективность управления для администраторов. Ниже приведены основные функциональные требования к системе.

Функции для пользователей:

Просмотр моделей обуви: Пользователь должен иметь возможность просматривать все товары в наличии; Система должна предоставлять поиск моделей по названию, цене; Пользователь должен иметь возможность просматривать подробную информацию о каждой модели, включая название, стоимость, внешний вид той или иной модели.

Покупка обуви. Пользователь должен иметь возможность приобрести модель обуви онлайн; Система должна запросить подтверждение на операцию у Пользователя.

Поиск товарной позиции. Пользователь должен иметь возможность легко, быстро и удобно найти интересующий его товар; Система должна предоставить пользователю инструмент для поиска любо позиции товара из наличия.

Личный кабинет. Пользователь должен иметь возможность просматривать историю своих действий, а именно видеть, какую обувь он приобретал ранее; Система должна предоставлять пользователю полный доступ к его прошлым покупкам.

Функции для администраторов:

- Управление товарным рядом: Администратор должен иметь возможность добавления, редактирования и удаления товарных позиций; Система должна предоставлять инструменты для создания, редактирования и удаления товарной позиции.

- Поиск товарных позиций: Администратор должен иметь возможность быстро найти определенный товар для дальнейшего его редактирования или удаления; Система должна предоставить инструмент Администратору для облегчения выполнения его задачи.

Общие требования

Интерфейс и удобство использования: Система должна иметь интуитивно понятный и удобный интерфейс с возможностью видоизменения как для пользователей, так и для администраторов; Навигация по системе должна быть простой и логичной, обеспечивая быстрый доступ к основным функциям.

- Высокая эффективность и надежность: система должна демонстрировать высокую производительность и быстрый отклик, быть стабильной и способной к самостоятельной коррекции ошибок для бесперебойной работы.

Предъявленные условия гарантируют разработку эффективной и удобной платформы "Магазин обуви", способной удовлетворить запросы как пользователей, так и администраторов, обеспечивая стабильную и интуитивно понятную работу для всех пользовательских категорий.

# 2 ПРОЕКТИРОВАНИЕ И РАЗРАБОТКА МОДУЛЯ

## 2.1 Эскизное проектирование

Система «Магазин обуви» предназначена для автоматизации процессов управления обувным магазином, предоставления актуальной информации о наличии моделей, а также для упрощения процесса приобретения товаров для пользователей. Основные цели и задачи системы включают:

Информирование пользователей

- Наличие моделей. Предоставление пользователям актуальной информации о наличии моделей обуви. Пользователи могут просматривать все модели, а также могут выбрать интересующую их модель или цену.

- Информация о моделях. Предоставление пользователю информации о товаре путем прикрепления фото товара и цены.

Упрощение процесса заказа обуви

- Онлайн приобретение. Возможность купить позицию товара, не выходя из дома, и в дальнейшем получить ее на пункте выдачи, доставкой или в магазине. Пользователю становится проще узнать о наличии товара, и исключается вариант потратить время впустую, поскольку программа не дает купить обувь, которой нет в наличии.

Управление магазином

- Добавление товара. Администратор вводит всю информацию о товаре, после чего она сразу становится доступна покупателям.

- Редактирование информации о товаре. Администратор может редактировать информацию о модели: добавить фото, изменить название, если в нем допущена ошибка, или изменить стоимость той или иной модели.

- Удаление товара. Администратор может удалить позицию товара, если по какой-то ошибке был введен несуществующий товар или товар был по ошибке введен дважды.

Улучшение пользовательского опыта

- История покупок. Пользователь может видеть приобретенные им позиции товаров.

Система "Магазин Обуви" разработана для создания удобного и эффективного инструмента управления магазином обуви и предоставления клиентам актуальной информации о моделях обуви. Она упрощает процесс выбора и покупки обуви, повышает качество обслуживания покупателей и обеспечивает владельцам магазина необходимые данные для анализа и оптимизации работы торговой точки.

## 2.1.1 Выбор архитектуры

Для разработки программы был выбран PyCharm. Эта среда разработки является одной из самых популярных для программирования благодаря своему удобству и широкому функционалу.

PyCharm — это среда программирования для языка Python, или IDE. Средами называют программы, в которых можно писать, запускать и отлаживать код, устанавливать новые расширения и дополнительные модули. Это мощный многофункциональный инструмент для разработчиков.

Ее разработала компания JetBrains, которая также сделала известную IDE для языка Java под названием IntelliJ IDEA. PyCharm пользуется более 50% разработчиков на Python: она удобная, понятная и многофункциональная.

PyCharm существует для нескольких операционных систем: Windows, Linux и macOS. Она поддерживает разные версии Python: и 2.x, и 3.x. Ее широкие возможности делают разработку на Python быстрее и эффективнее.

Языки программирования устроены так, что писать код на них можно где угодно, даже в «Блокноте». Главное — сохранить файл в нужном расширении, а потом запустить с помощью установленного интерпретатора или компилятора. Но большинство разработчиков все равно пользуются IDE. И вот почему:

Интеллектуальное завершение кода: PyCharm предлагает интеллектуальное завершение кода, которое предлагает фрагменты кода, имена переменных и функций по мере ввода, экономя ваше время и уменьшая количество ошибок.

Встроенный отладчик: Встроенный отладчик позволяет устанавливать точки останова, проверять переменные и пошагово выполнять код для выявления и устранения проблем.

Мощный рефакторинг: вы можете легко реорганизовать свой код с помощью PyCharm, сделав его более чистым и ремонтопригодным. Это может помочь вам переименовывать переменные, извлекать методы и многое другое.

Интеграция с системой контроля версий: PyCharm легко интегрируется с системами контроля версий, такими как Git, что упрощает отслеживание изменений и совместную работу с другими участниками над вашими проектами.

Анализ кода: PyCharm включает инструменты анализа кода, которые могут обнаруживать запахи кода и предлагать улучшения вашего кода, помогая вам писать более чистые и эффективные программы на Python.

Преимущества PyCharm

Повышение производительности: Интеллектуальные функции и мощные инструменты PyCharm значительно повышают производительность вашего программирования, позволяя вам писать лучший код за меньшее время.

Надежная экосистема: Он имеет обширную экосистему плагинов и расширений, которые можно настроить в соответствии с вашими конкретными потребностями и предпочтениями.

Поддержка сообщества: У PyCharm большое и активное сообщество, а это значит, что вы можете найти множество руководств, документации и поддержки онлайн.

Отличная документация и сообщество: JetBrains, компания, стоящая за PyCharm, предоставляет обширную документацию и ресурсы

Кроссплатформенность: PyCharm доступен для Windows, macOS и Linux, гарантируя, что вы можете использовать одну и ту же IDE в разных операционных системах без существенных различий в функциональности.

Недостатки PyCharm

- Ресурсоемкость: PyCharm может быть ресурсоемким, особенно профессиональная версия, которая может работать некорректно на старых или менее мощных компьютерах.

- Кривое обучение: Для начинающих обширные возможности PyCharm могут оказаться непосильными, и может потребоваться некоторое время, чтобы изучить и освоить все его возможности.

## 2.1.2 Выбор инструментальных средств реализации

При выборе инструментальных средств для реализации системы «Магазин обуви» я принял следующие решения:

Для написания программного кода своей программы я выбрал язык программирования Python. Одним из сильных аспектов Python является его обширная стандартная библиотека, содержащая модули для работы с различными аспектами, включая интернет-протоколы, файловую систему, обработку данных и многое другое. Благодаря своей платформенной независимости, Python успешно функционирует на различных операционных системах, включая Windows, macOS и Linux, что делает его универсальным выбором для программистов во всем мире.

Для графического интерфейса выбрал фреймворк PyQT. Выбор графического интерфейса PyQt может считаться одним из лучших вариантов для написания приложений по следующим причинам:

- Pythonic API: PyQt предлагает Pythonic API, что делает его привлекательным для разработчиков Python, так как позволяет им писать код более естественным образом.

- Широкие возможности: PyQt предоставляет широкий спектр функций и возможностей для создания разнообразных пользовательских интерфейсов. Это включает в себя виджеты, стилизацию, графику, события и т.д.

- Богатая документация и сообщество: Поддержка PyQt включает в себя обширную документацию, примеры кода, форумы поддержки и сообщества разработчиков, что облегчает изучение и решение проблем при разработке с использованием PyQt.

- Интеграция с другими библиотеками: PyQt хорошо интегрируется с другими популярными библиотеками Python, такими как NumPy, Pandas и Matplotlib, что упрощает создание приложений с графикой, обработкой данных и другими функциями.

3. База данных: SQLite

Выбор SQLite в роли реляционной базы данных обусловлен его легкостью интеграции с Python, простотой настройки и надежностью. Эта база данных позволяет хранить все данные приложения локально, что упрощает управление информацией и гарантирует эффективную работу, особенно при работе с небольшими объемами данных.

## 2.2 Проектирование базы данных

Я создал базу данных «real\_estate.db». В ней присутствуют следующие таблицы:

Users

Properties

Purchases

Purchase\_history

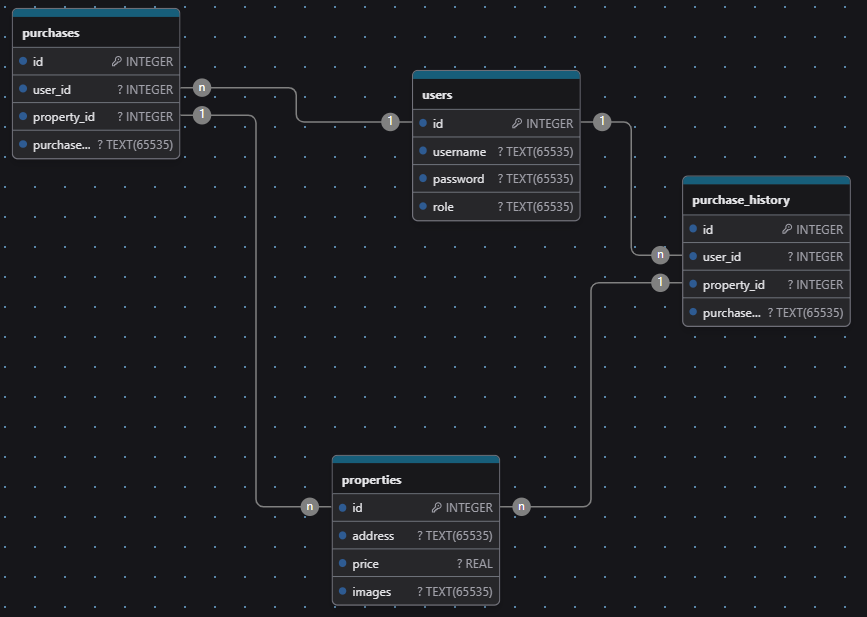


Рисунок 1 – ER-диаграмма базы данных

Структура Базы данных:

Таблица 1 — “users”

|  |  |  |
| --- | --- | --- |
| **Атрибут** | **Тип** | **Описание** |
| id | Integer primary key autoincrement | Первичный ключ пользователя |
| username | Text unique not null | Логин пользователя |
| password | Text not null | Пароль |
| role | Text not null | Роль |

Таблица 2 — “properties”

|  |  |  |
| --- | --- | --- |
| **Атрибут** | **Тип** | **Описание** |
| id | Integer primary key autoincrement | Первичный ключ модели |
| address | Text not null | Название модели |
| price | Real not null | Цена |
| images | TEXT | Изображение модели |

Таблица 3 — “purchases”

|  |  |  |
| --- | --- | --- |
| **Атрибут** | **Тип** | **Описание** |
| id | Integer primary key autoincrement | Первичный ключ покупки |
| user\_id | Integer not null | Внешний ключ пользователя |
| property\_id | Integer not null | Внешний ключ модели |
| purchase\_date | Text not null | Время покупки |

Таблица 4 — “purchase\_history”

|  |  |  |
| --- | --- | --- |
| **Атрибут** | **Тип** | **Описание** |
| id | Integer primary key autoincrement | Первичный ключ совершенной покупки |
| user\_id | Integer | Внешний ключ пользователя |
| property\_id | Integer not null | Внешний ключ модели |
| purchase\_date | Text not null | Время покупки |

Для доступа к функциям системы необходимо авторизоваться в диалоговом окне авторизации (Приложение 1, Рисунок 2)

Комбобокс позволяет выбрать, кто будет пользоваться системой (Пользователь/Администратор) (Приложение 1, Рисунок 3)

В случае успешной авторизации появится главное окно приложения (Приложение 1, Рисунок 4)

В случае ошибки Пользователь получает сообщение об ошибке и может снова совершить попытку входа в приложение неограниченное количество раз.

Пользователь может выбрать интересующую обувь, просмотреть ее фотографии (Приложение 1, Рисунок 5), а при покупке позиции, система запросит подтверждение покупки (Приложение 1, Рисунок 5.1). После приобретения на модели будет подпись «ПРОДАНО» для того, чтобы другие пользователи не приобретали купленную модель (Приложение 1, Рисунок 5.2).

Кроме того, у каждого Пользователя ведется учет его покупок, и, нажимая на кнопку «История покупок», Пользователь видит все купленные им товары (Приложение 1, Рисунок 5.3)

Авторизоваться можно как от имени Пользователя, так и от имени Администратора. Функционал меняется, в зависимости от роли.

Для Администратора доступна возможность добавлять, редактировать и удалять позиции товаров (Приложение 1, Рисунок 6)

Также, для обоих ролей доступен выбор темы (светлая/темная) для комфортного пользования системой (Приложение 1, Рисунок 7)

## 2.3 Написание программного кода

Данный код создает основное окно программы с информацией по товарам, регулировкой темы оформления, регулировкой длины и ширины ячеек с информацией из базы данных

Это конструктор класса MainWindow, который принимает два параметра: db, база данных, и user, представляющий идентификатор пользователя и его роль.



Вызывает конструктор родительского класса, в данном случае QMainWindow, чтобы корректно инициализировать основные компоненты оконного приложения.



Присваивает переменной self.db значение, переданное в параметре db, для дальнейшего использования в классе.



Разбивает кортеж user на две переменные self.user\_id и self.role, что представляет идентификатор пользователя и его роль.



Устанавливает заголовок главного окна на "Продажа Обуви".



Устанавливает флаг is\_dark\_theme в значение False, для определения темы интерфейса.



Создает вертикальный макет для управления компонентами внутри главного окна.



Создает кнопку theme\_button с пустым текстом, привязанную к главному окну self.



Устанавливает фиксированный размер для кнопки (40x40), чтобы сделать ее квадратной.



При нажатии на кнопку theme\_button вызывается метод toggle\_theme (для изменения темы интерфейса).



Добавляет кнопку theme\_button в макет layout с выравниванием кнопки справа в соответствии с флагом AlignRight



# 3 ОТЛАДКА И ТЕСТИРОВАНИЕ МОДУЛЯ

## 3.1 Тестирование программного продукта.

Тестирование начинаю с окна авторизации.

Если ввести неправильно логин и пароль, то вылезает окно с предупреждением:

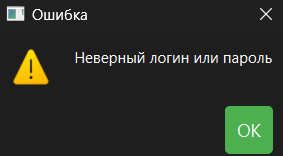


Рисунок 7 – Неверный ввод данных для входа

В случае неправильного или ошибочного нажатия по кнопке «Купить» программа оповещает об этом:

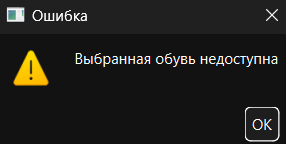


Рисунок 8 – Ошибочное нажатие по кнопке «Купить»

При запросе истории покупок без совершенных покупок программа оповещает об отсутствии покупок

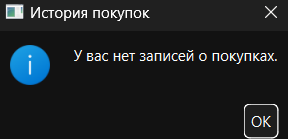


Рисунок 8 – Пустая история покупок

При отсутствии ввода данных по модели обуви программа оставляет цену по умолчанию 0, изображения оставляет пустым, а при пустом названии модели программа не создает ячейку.



Рисунок 9 – Попытка создания пустой ячейки

Если Администратор нажмет «Редактировать обувь» или «Удалить обувь», не выбрав модель, программа уведомит об этом

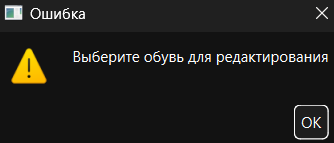


Рисунок 10 – Попытка редактирования не выбранной обуви

## 3.1.1 Реализация пользовательского интерфейса

Для доступа к функциям системы необходимо авторизоваться в диалоговом окне авторизации (Приложение 1, Рисунок 1)

Комбобокс позволяет выбрать, кто будет пользоваться системой (Пользователь/Администратор) (Приложение 1, Рисунок 2)

В случае успешной авторизации появится главное окно приложения (Приложение 1, Рисунок 3)

В случае ошибки Пользователь получает сообщение об ошибке и может снова совершить попытку входа в приложение неограниченное количество раз.

Пользователь может выбрать интересующую обувь, просмотреть ее фотографии (Приложение 1, Рисунок 4), а при покупке позиции, система запросит подтверждение покупки (Приложение 1, Рисунок 4.1). После приобретения на модели будет подпись «ПРОДАНО» для того, чтобы другие пользователи не приобретали купленную модель (Приложение 1, Рисунок 4.2).

Кроме того, у каждого Пользователя ведется учет его покупок, и, нажимая на кнопку «История покупок», Пользователь видит все купленные им товары (Приложение 1, Рисунок 4.3)

Авторизоваться можно как от имени Пользователя, так и от имени Администратора. Функционал меняется, в зависимости от роли.

Для Администратора доступна возможность добавлять, редактировать и удалять позиции товаров (Приложение 1, Рисунок 5)

Также, для обоих ролей доступен выбор темы (светлая/темная) для комфортного пользования системой (Приложение 1, Рисунок 6)

## 3.2 Отладка программного продукта.

Для корректной работы программы необходимо:

1. Вводить все данные соответственно требованиям программы
2. Не оставлять пустыми ячейки информации
3. Правильно записывать стоимость товаров
4. Загрузка медиафайлов только с расширениями .png, .jpg и jpeg
5. Своевременно очищать уже проданные позиции товаров

# Заключение

В ходе выполнения данной курсовой работы была разработана система "Магазин обуви", предназначенная для автоматизации управления ассортиментом обуви и предоставления актуальной информации покупателям. В процессе работы были решены следующие задачи: проведен анализ требований, спроектирована архитектура системы, реализованы основные компоненты программного модуля, проведено тестирование.

Разработанный программный модуль позволяет администраторам магазинов обуви эффективно управлять ассортиментом, а покупателям — получать актуальную информацию об товарных позициях в наличии. Это способствует улучшению обслуживания клиентов и оптимизации работы магазинов обуви.

Проведенное тестирование показало, что система обладает высокой надежностью и удобством использования. Разработанный модуль полностью соответствует ожиданиям и требованиям

Таким образом, цель курсовой работы достигнута, и разработанный программный модуль может быть рекомендован для внедрения в магазинах обуви с целью улучшения качества обслуживания и повышения эффективности их работы.СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

Нормативно-правовые источники:  
ГОСТ 7.32–2001. Межгосударственный стандарт. Система стандартов по информации, библиотечному и издательскому делу. Отчет о научно-исследовательской работе. Структура и правила оформления.  
ГОСТ Р 7.0.5–2008. Национальный стандарт Российской Федерации. Система стандартов по информации, библиотечному и издательскому делу библиографическая ссылка.  
ГОСТ 7.1–2003. Система стандартов по информации, библиотечному и издательскому делу. Библиографическая запись. Библиографическое описание. Общие требования и правила составления.  
ГОСТ 7.80–2000. Система стандартов по информации, библиотечному и издательскому делу. Библиографическая запись. Заголовок. Общие требования и правила составления.

Аллен, Г. Тейлор SQL для чайников / Аллен Г. Тейлор. - М.: Диалектика, Вильямс, 2015. - 416 c.

Бен, Форта SQL за 10 минут / Форта Бен. - М.: Диалектика / Вильямс, 2015. - 320 c.

Боклаг, Н. Ю. Основы программирования на языке Python / Н.Ю. Боклаг. - М.: Бибком, 2020. - 685 c.

Бьюли, Алан Изучаем SQL / Алан Бьюли. - М.: Символ-плюс,   
2014. - 450 c.

Бэрри, Пол Изучаем программирование на Python / Пол Бэрри. - М.: Эксмо, 2023. - 897 c.

Грабер, Мартин SQL для простых смертных / Мартин Грабер. - М.: ЛОРИ, 2014. - 378 c.

Гуриков, С.Р. Основы алгоритмизации и программирования на Python. Учебное пособие. Гриф МО РФ / С.Р. Гуриков. - М.: Инфра-М, Форум, 2018. - 160 c.

Дейт, К. Дж. SQL и реляционная теория. Как грамотно писать код на SQL / К.Дж. Дейт. - М.: Символ-плюс, 2016. - 0 c.  
Джеймс, Р. Грофф SQL. Полное руководство / Джеймс Р. Грофф, Пол Н. Вайнберг, Эндрю Дж. Оппель. - М.: Вильямс, 2014. - 960 c.

Дунаев, В. В. Базы данных. Язык SQL для студента / В.В. Дунаев. - М.: БХВ-Петербург, 2017. - 288 c.

<https://nikitaefremov.ru>

https://neylonov.ru

# Приложение 1 Описание интерфейса и кода

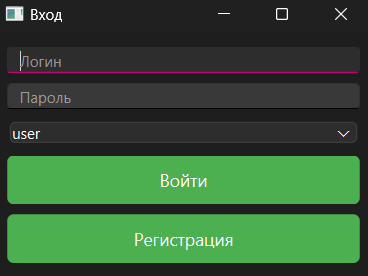


Рисунок 1

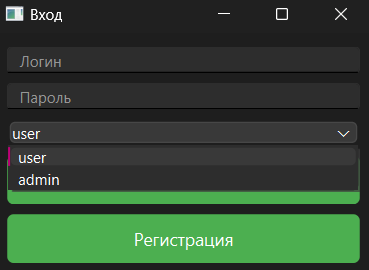


Рисунок 2 – выбор роли Пользователя

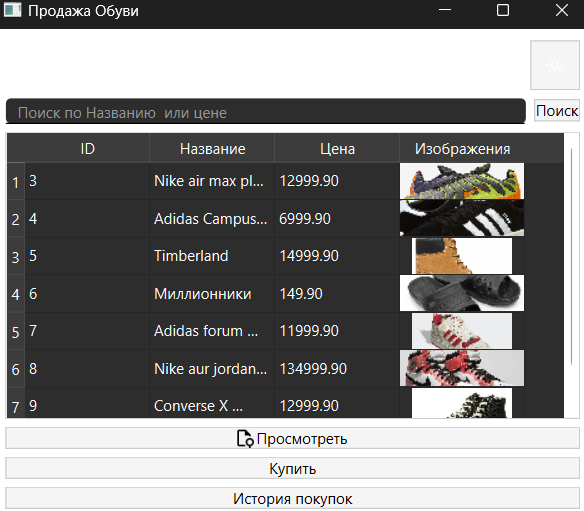


Рисунок 3 – Главное окно при входе под Пользователем

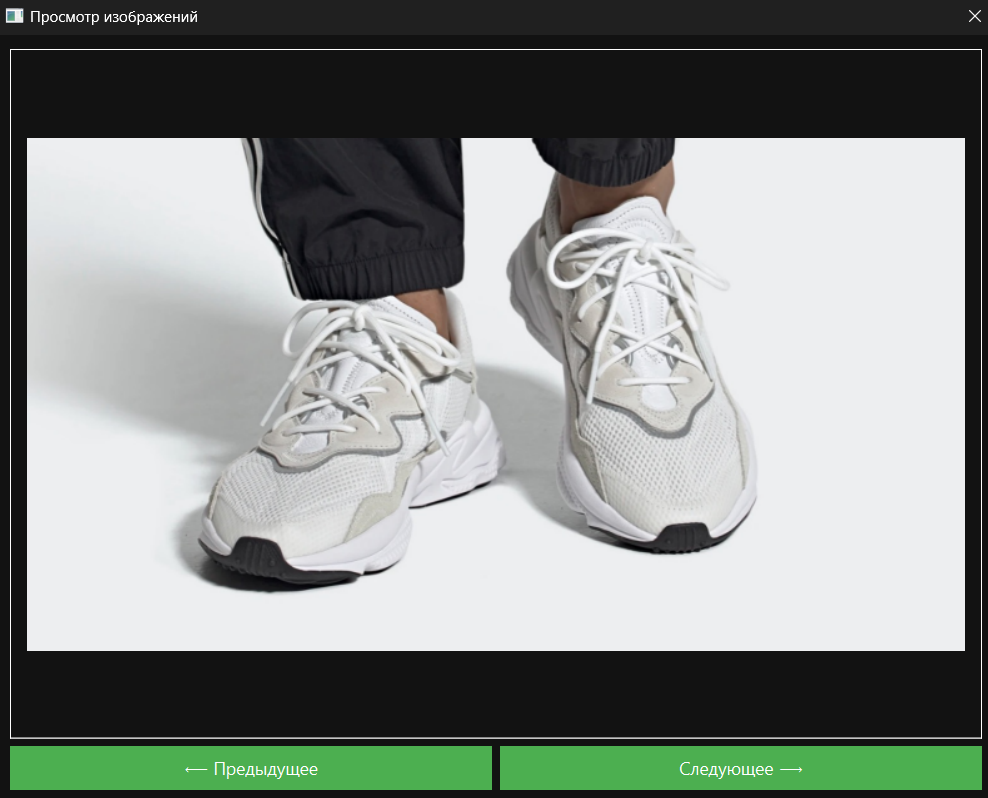


Рисунок 4 – Просмотр изображения модели

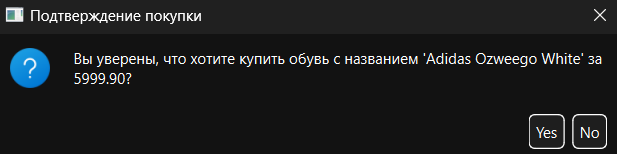


Рисунок 4.1 – Подтверждение покупки выбранной модели



Рисунок 4.2 – Метка «ПРОДАНО» на приобретенной модели

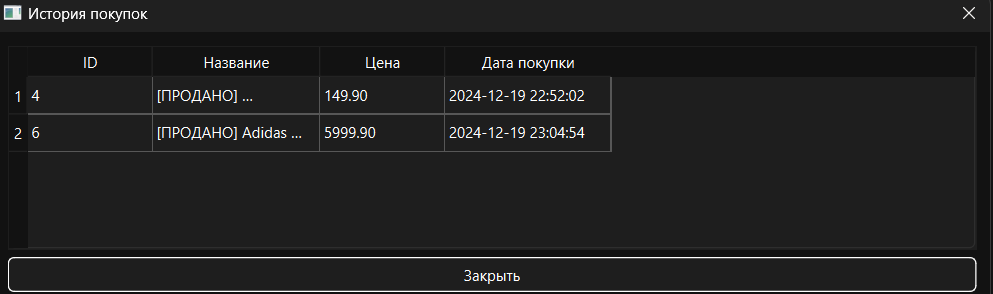


Рисунок 4.3 – История покупок Пользователя

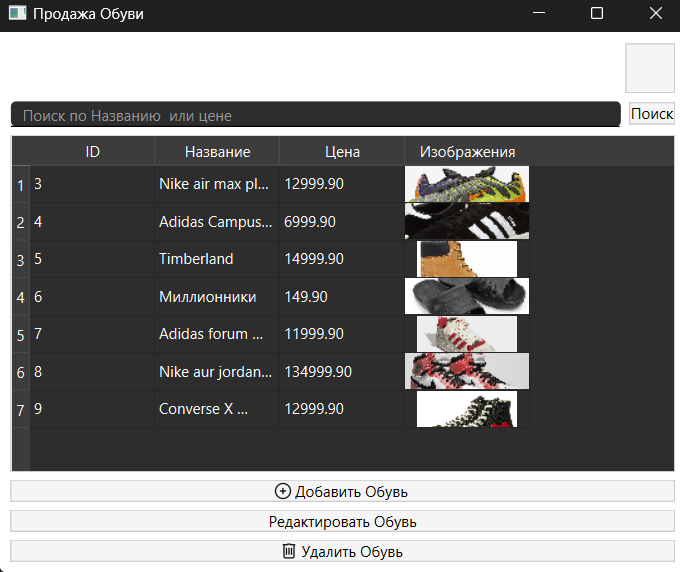


Рисунок 5 – Главное окно при входе под Администратором

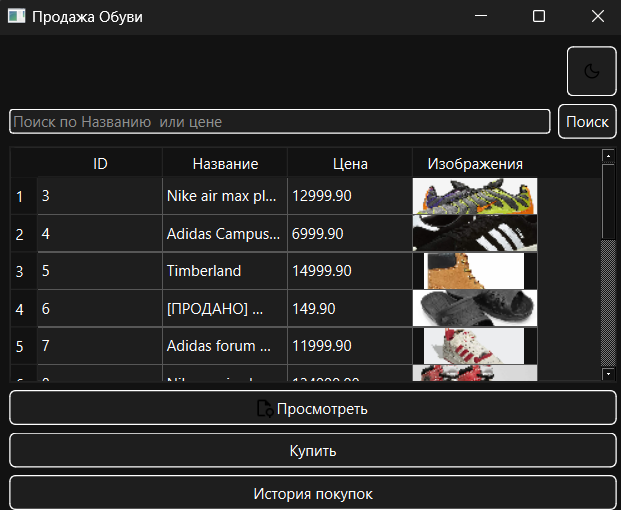


Рисунок 6 – Вид приложения с включенной темной темой