МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

**Федеральное государственное бюджетное образовательное учреждение**

**высшего образования**

**«Вятский государственный университет»**

**Колледж ВятГУ**

**КУРСОВОЙ ПРОЕКТ**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| По междисциплинарному курсу | | | | | МДК 05.01 Проектирование и дизайн | | | |
| информационных систем | | | |
| Тема | Проектирование информационной системы для сети цветочных магазинов | | | | | | | |
|  | | | | | | | |
| Студента (ки) | | | | Цветковой Кристины Андреевны | | | | |
|  | | | *ФИО (полностью в родительном падеже)* | | | | | |
| Курс | | 3 | | | Форма обучения | очная | |  |
|  | | *(арабской цифрой)* | | |  | | *(очная, заочная)* | |
| Основная профессиональная образовательная программа по специальности | | | | | | | | |
| 09.02.07 Информационные системы и программирование | | | | | | | | |
| (код и наименование специальности без кавычек) | | | | | | | | |

Руководитель курсового проекта \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/ Жукова М.Н.

(подпись) (фамилия, инициалы)

Оценка \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

*(прописью, без сокращений)*

Киров, 2024

**РЕФЕРАТ**

Цветкова К.А. ПРОЕКТИРОВАНИЕ ИНФОРМАЦИОННОЙ СИСТЕМЫ ДЛЯ СЕТИ ЦВЕТОЧНЫХ МАГАЗИНОВ: ТПЖА 09.02.07 738314 ПЗ: Курс. проект / Колледж ВятГУ; рук. Жукова М.Н. – Киров, 2024. ПЗ 42 с., 21 рис., 2 табл., 7 источников.

СЕТЬ ЦВЕТОЧНЫХ МАГАЗИНОВ, АДМИНИСТРАТОР, ПРОДАВЦЫ, ПРОДАЖИ, ПОСТАВКИ.

Объектом исследования является система для сети цветочных магазинов.

Предметом исследования является оптимизация процессов сети цветочных магазинов.

Цель работы – проектирование и разработка информационной системы для сети цветочных магазинов, а также подготовка необходимой документации.

В процессе работы были проведены следующие исследования: 1) обзор предметной области, обзор аналогов и сравнительный анализ; 2) написание самого технического задания, состоящего из: назначения разработки, функциональных характеристик, требований к надёжности, условий эксплуатации и требований к составу и параметрам технических средств; 3) описание решения и концепции; 4) архитектура решения; 5) разработка схем бизнес-процессов с их описанием; 6) разработка схем алгоритмов и кода на естественном языке; 7) проектирование прототипа пользовательского интерфейса с описанием.

Элементами научного новшества полученных результатов является информационная система.

Областью возможного практического применения являются: сети цветочных магазинов.

Технико-экономическая и социальная значимость: автоматизация процессов управления деятельностью сети цветочных магазинов.

**СОДЕРЖАНИЕ**

Изм.

Лист

№ докум.

Подпись

Дата

Лист

2

ТПЖА 09.02.07 738314 ПЗ

Разраб.

Цветкова К.А.

Провер.

Жукова М.Н.

Реценз.

Н. Контр.

Утверд.

Проектирование информационной системы для сети цветочных магазинов

Лит.

Листов

36

Колледж ВятГУ

ИСПк-303-52-00

[ВВЕДЕНИЕ 3](#_Toc192462145)

[1 АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ 4](#_Toc192462146)

[1.1. Описание предметной области 4](#_Toc192462147)

[1.2. Глоссарий 4](#_Toc192462148)

[1.3. Обзор аналогов 5](#_Toc192462149)

[1.3.1. Аналог 1 – «Флория» 5](#_Toc192462150)

[1.3.2. Аналог 2 – «FloraPOS» 6](#_Toc192462151)

[1.3.3. Аналог 3 – «GBS.Market» 6](#_Toc192462152)

[1.4. Требования 8](#_Toc192462153)

[1.4.1. Нефункциональные требования 8](#_Toc192462154)

[1.4.2. Функциональные требования 8](#_Toc192462155)

[2 ТЕХНИЧЕСКОЕ ЗАДАНИЕ 10](#_Toc192462156)

[3 ПРОЕКТИРОВАНИЕ ИНФОРМАЦИОННОЙ СИСТЕМЫ 11](#_Toc192462157)

[3.1 Моделирование информационной системы 11](#_Toc192462158)

[3.1.1. Разработка IDEF0 11](#_Toc192462159)

[3.1.2. Разработка DFD 13](#_Toc192462160)

[3.1.3. Разработка диаграммы вариантов использования 14](#_Toc192462161)

[3.1.4. Разработка базы данных 16](#_Toc192462162)

[4 РЕАЛИЗАЦИЯ ИНФОРМАЦИОННОЙ СИСТЕМЫ 25](#_Toc192462163)

[ЗАКЛЮЧЕНИЕ 35](#_Toc192462164)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 36](#_Toc192462165)

[ПРИЛОЖЕНИЕ А 37](#_Toc192462166)

[ПРИЛОЖЕНИЕ Б 38](#_Toc192462167)

24.12.24

# ВВЕДЕНИЕ

Курсовой проект проходил на базе Колледжа ФГБОУ ВО «Вятский государственный университет» в период с 05.09.2024 г. по 25.12.2024 г.

В документе рассматриваются следующие разделы:

1. Анализ предметной области.
2. Техническое задание.
3. Реализация информационной системы.
4. Заключение.
5. Список использованных источников.
6. Приложения.

Анализ предметной области содержит описание изучаемой темы, глоссарий, концепция, обзор аналогов, их преимущества и недостатки, на основе которых были рассмотрены аспекты, которые были реализованы в проекте.

В разделе «Техническое задание» описывается понятие технического задания, его значение и роль в процессе разработки.

Разделы «Проектирование» и «Реализация информационной системы» включает в себя этапы создания информационной системы, включая моделирование её работы, разработку прототипов и интерфейса, а также создание базы данных, необходимой для функционирования системы.

В «Заключении» содержится вывод о проделанной работе и обобщение полученных результатов.

Список использованных источников содержит в себе перечисление всех использованных при написании курсового проекта источников информации.

В «Приложениях» находится дополнительные материалы проекта, такие как ТЗ и исходный код программы.

# АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ

## Описание предметной области

На сегодняшний день информационные системы для цветочных магазинов играют ключевую роль в автоматизации всех аспектов бизнеса. Основными задачами системы являются управление каталогом товаров, обработка заказов клиентов, аналитика продаж, обеспечение удобного интерфейса для пользователей. Система ориентирована на магазины электроники, интернет-магазины и дистрибьюторов.

Система предоставляет возможности для упрощения и ускорения процессов, связанных с управлением запасами, продажами и поставками. Они позволяют улучшить качество обслуживания клиентов и повысить конкурентоспособность бизнеса. Это позволяет не только оптимизировать внутренние процессы, но и способствует улучшению финансовых результатов.

## Глоссарий

Информационная система (ИС) – совокупность программного и аппаратного обеспечения для обработки данных.

Администратор – сотрудник, управляющий всеми аспектами работы системы.

База данных – это организованная структура для хранения, управления и манипуляции данными.

Интерфейс программы – набор инструментов, который позволяет пользователю взаимодействовать с программой.

Информационная система – система обработки информации совместно с соответствующими организационными ресурсами, которая обеспечивает и распространяет информацию.

Модуль – независимая часть программы или системы, выполняющая конкретную функцию и способная работать отдельно от других частей.

Продавец – сотрудник, ответственный за взаимодействие с клиентами и обработку заказов.

## Обзор аналогов

В данном разделе приведен обзор аналогов, рассматриваемой предметной области.

### Аналог 1 – «Флория»

«Программа Программное обеспечение для автоматизации работы цветочных магазинов. Интерфейс «Флория» представлен на рисунке 1.

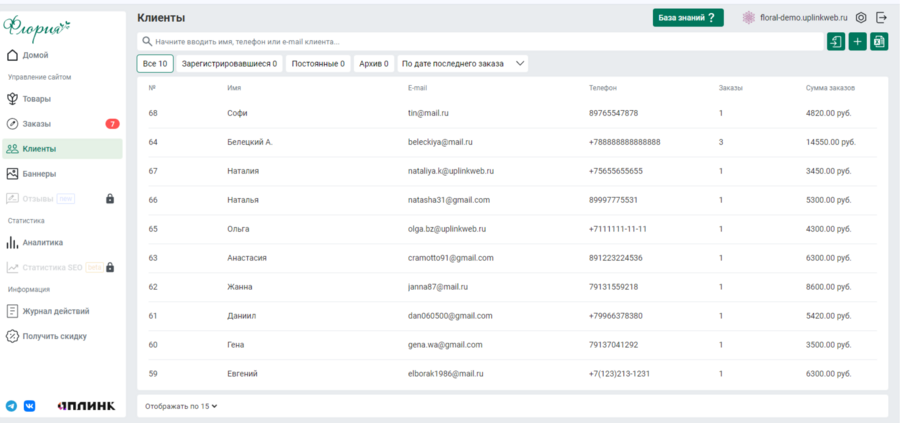


Рисунок 1 – Интерфейс «Флория»

Приложение предлагает:

* предоставление отчетности;
* управление заказами;
* удобный интуитивно понятный интерфейс.

Недостатками являются:

* ограниченный функционал в бесплатной версии;
* стоимость, которая может быть высокой для небольших магазинов или сетей;
* возможность сбоев программы при нагрузке.

### Аналог 2 – «FloraPOS»

Информационная система для управления небольшими цветочными магазинами, специализированная на кассовых операциях и управлении товарами. Интерфейс «FloraPOS» представлен на рисунке 2.



Рисунок 2 – Интерфейс «FloraPOS»

Достоинства:

* доступная цена;
* возможность поиска товаров;
* интуитивно понятный интерфейс.

Недостатки данного приложения:

* отсутствие возможности управления заказами;
* отсутствие возможности добавления, редактирования характеристик товаров;
* отсутствие возможности делать отчет;
* недостаточно оптимизировано для крупных магазинов или сетей.

### Аналог 3 – «GBS.Market»

«GBS.Market» используется продавцами для осуществления покупок.

Интерфейс программы представлен на рисунке 3.

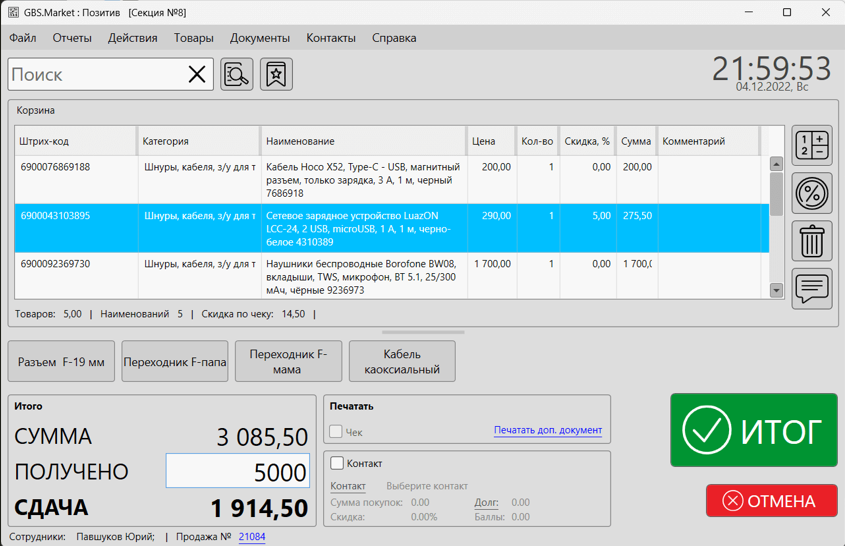


Рисунок 3 – Интерфейс «GBS.Market»

Обладает следующими преимуществами:

* возможность создания отчета;
* возможность добавления товара и редактирование;
* доступная цена.

Недостатки:

* низкая производительность, что может спровоцировать медленную работу или зависание;
* присутствует сложность при освоении программы, что может занять некоторое время;
* интуитивно непонятный интерфейс;
* отсутствие возможности управления заказами;

В результате обзора аналогов обнаружилось мало доступных ИС в публичном доступе. Проанализировав их, были найдены недостатки.

Было решено разработать информационную систему для сети цветочных магазинов, которой могли бы пользоваться администратор и продавцы. Она должна обладать рядом преимуществ, обеспечивающих её эффективность и удобство в эксплуатации. Система должна иметь интуитивно понятный интерфейс, благодаря которому отсутствует потребность в локализации, должная иметь возможность создания отчетов.

## Требования

В результате проведения обзора аналогов, был составлен перечень требований к разрабатываемой системе.

### Нефункциональные требования

Система должна соответствовать следующим нефункциональным требованиям:

* иметь интуитивно понятный интерфейс;
* работать на операционных системах Windows;
* время полного запуска программы – не более 30 секунд;
* время отклика на запросы — не более 5 секунд.

### Функциональные требования

При использовании программы пользователи должны иметь возможность:

* управлять каталогом товаров;
* управлять магазинами;
* управлять продавцами;
* управлять поставщиками и поставками;
* отчетности;
* обработки заказов: оформление, расчет стоимости, создание документов продаж;
* обеспечения разграничения возможностей в зависимости от роли;
* обеспечения регистрации и авторизации;

Разграничение возможностей:

* администратор: может создавать учетные записи продавцам, может добавлять, редактировать и удалять сведения о: товарах, поставках, поставщиках, магазинах, продавцах. Имеет доступ к отчетности и документам о продажах.
* продавец: может просматривать каталог товаров, собирать и обрабатывать заказ, смотреть документы о продажах.

# ТЕХНИЧЕСКОЕ ЗАДАНИЕ

Техническое задание представляет собой документ, в котором определяются требования к результату работы, которые служат основой для его разработки, выполнения и контроля. В ТЗ отражаются цели и задачи проекта, функциональные и нефункциональные требования, технические параметры, сроки реализации, бюджет, а также иные данные, необходимые для успешного выполнения работы.

ГОСТы, используемые при подготовке ТЗ:

* ГОСТ 34.602-2020 «Информационные технологии. Комплекс стандартов на автоматизированные системы. Техническое задание на создание автоматизированной системы» — содержит требования к составу, содержанию и правилам оформления технического задания на создание автоматизированных систем.

Техническое задание на разработку информационной системы для центра детского творчества приведено в Приложении А.

# ПРОЕКТИРОВАНИЕ ИНФОРМАЦИОННОЙ СИСТЕМЫ

## Моделирование информационной системы

Моделирование информационной системы включает этапы: разработку методологии функционального моделирования IDEF0 для описания функций системы, создание диаграммы потоков данных (DFD), которая отображает движение информации между компонентами, и разработку диаграммы вариантов использования (Use Case), описывающую взаимодействие пользователей с системой.

### Разработка IDEF0

IDEF0 (Integrated Definition for Function Modeling) – это методология функционального моделирования, которая описывает функции, информацию и поведение системы, а также их взаимосвязи. Она использует формальный язык и графическую нотацию для визуализации функциональных аспектов, включая потоки данных и контрольные точки. IDEF0 позволяет отображать иерархические уровни процессов и понимать взаимодействие между функциональными элементами.

На рисунке 4 представлена контекстная диаграмма в нотации IDEF0, иллюстрирующая основной процесс разрабатываемой системы, включая основные входы, выходы, управляющие элементы и механизмы взаимодействия системы с внешними компонентами.



Рисунок 4 – Функциональная модель ИС в нотации IDEF0 (контекстная диаграмма)

Для получения большей информации о функциях системы была проведена декомпозиция, и также отображена с использованием нотации IDEF0 на рисунке 5.

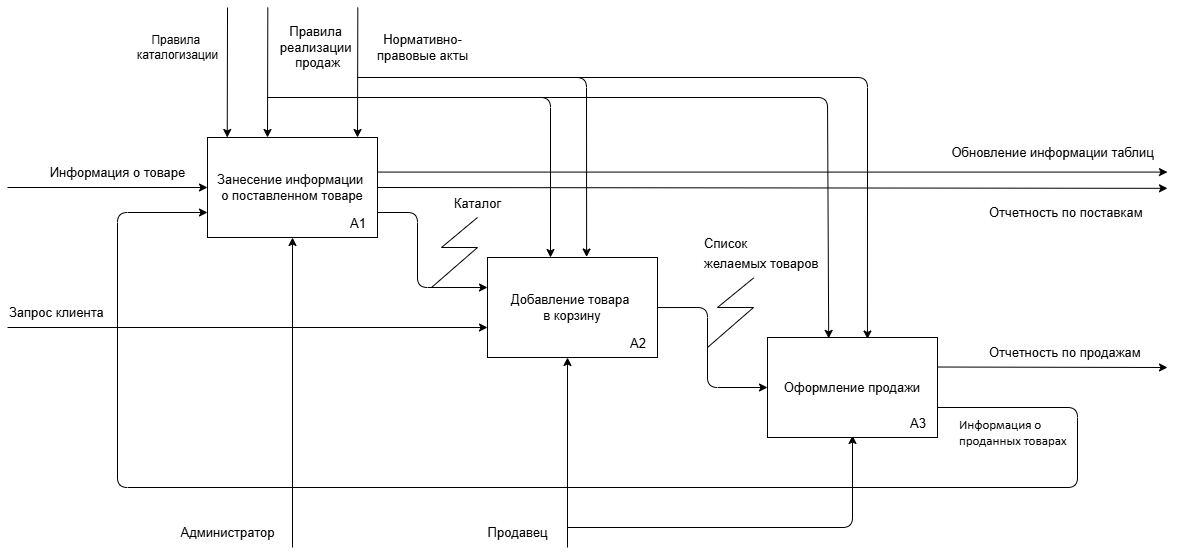


Рисунок 5 – Диаграмма декомпозиции функциональной модели ИС

На данной диаграмме можно выделить три основных процесса: «Занесение информации о поставленном товаре», «Добавление товара в корзину» и «Оформление продажи».

Занесение информации о поставленном товара начинается с поступающего товара. Оно регулируется нормативно-правовыми актами. На данном этапе заносится информация о товарах, поставленных в магазин. Результатом является обновление информации таблиц и отчетность о поставках.

После занесения информации о поставленных товарах и добавления товара в корзину начинается процесс оформления продажи. Результатом этого процесса является отчетность о продажах.

На основе отчетности о поставках и продажах, администратор сети формирует отчеты, которые позволяют оценить эффективность работы сети.

### Разработка DFD

DFD (Data Flow Diagram) – это графическое представление потоков данных в информационной системе. Он используется для моделирования и анализа потоков данных в информационных системах с целью описания взаимодействия между различными процессами, данными и внешними сущностями.

Диаграмма декомпозиции в нотации DFD для сети цветочных магазинов представлена на рисунке 6.

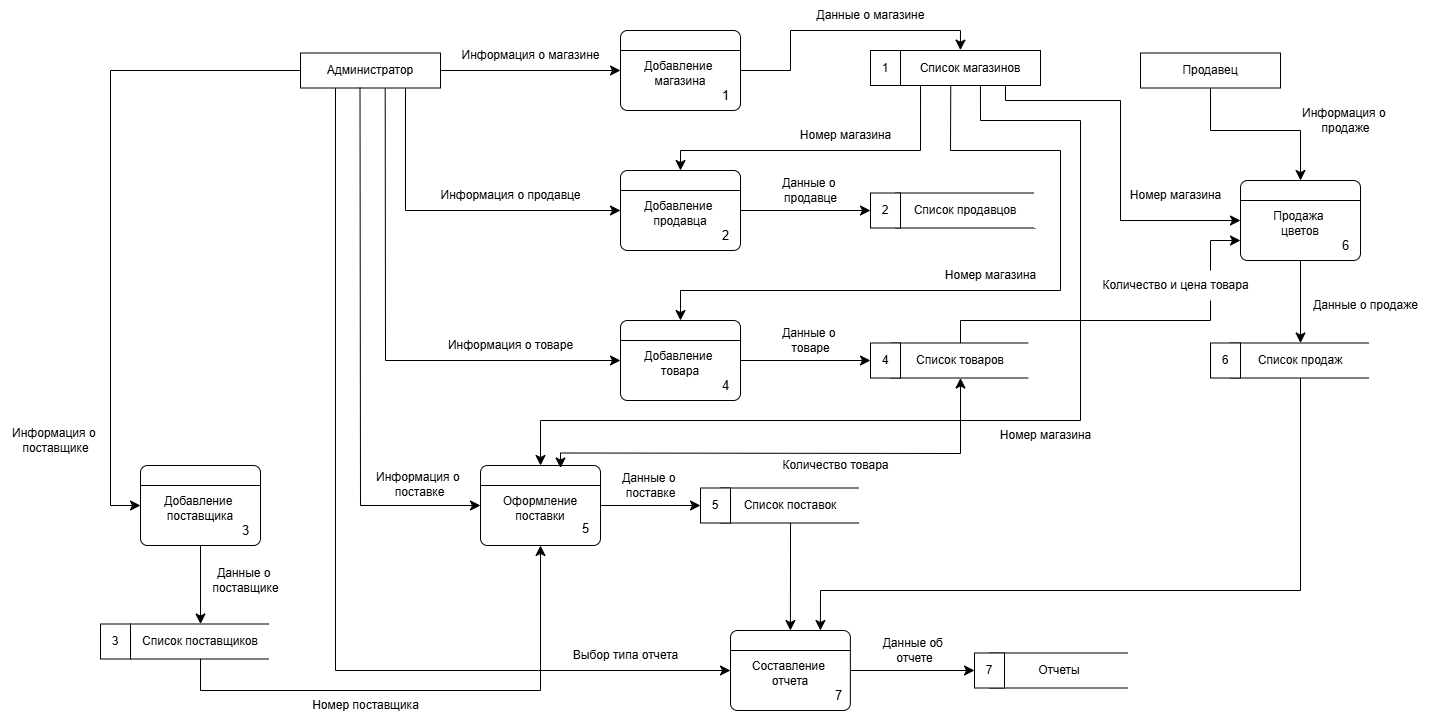


Рисунок 6 – Диаграмма в нотации DFD

### Разработка диаграммы вариантов использования

Основными действующими лицами являются акторы Администратор и Продавец.

В данной информационной системе Администратор может создавать учетные записи продавцам, может добавлять, редактировать и удалять сведения о: товарах, поставках, поставщиках, магазинах, продавцах. Имеет доступ к отчетности и документам о продажах.

Продавец, в свою очередь, может просматривать каталог товаров, собирать и обрабатывать заказ, смотреть документы о продажах.

Моделирование функциональности системы представлено с использованием Use Case-диаграммы на рисунке 7.

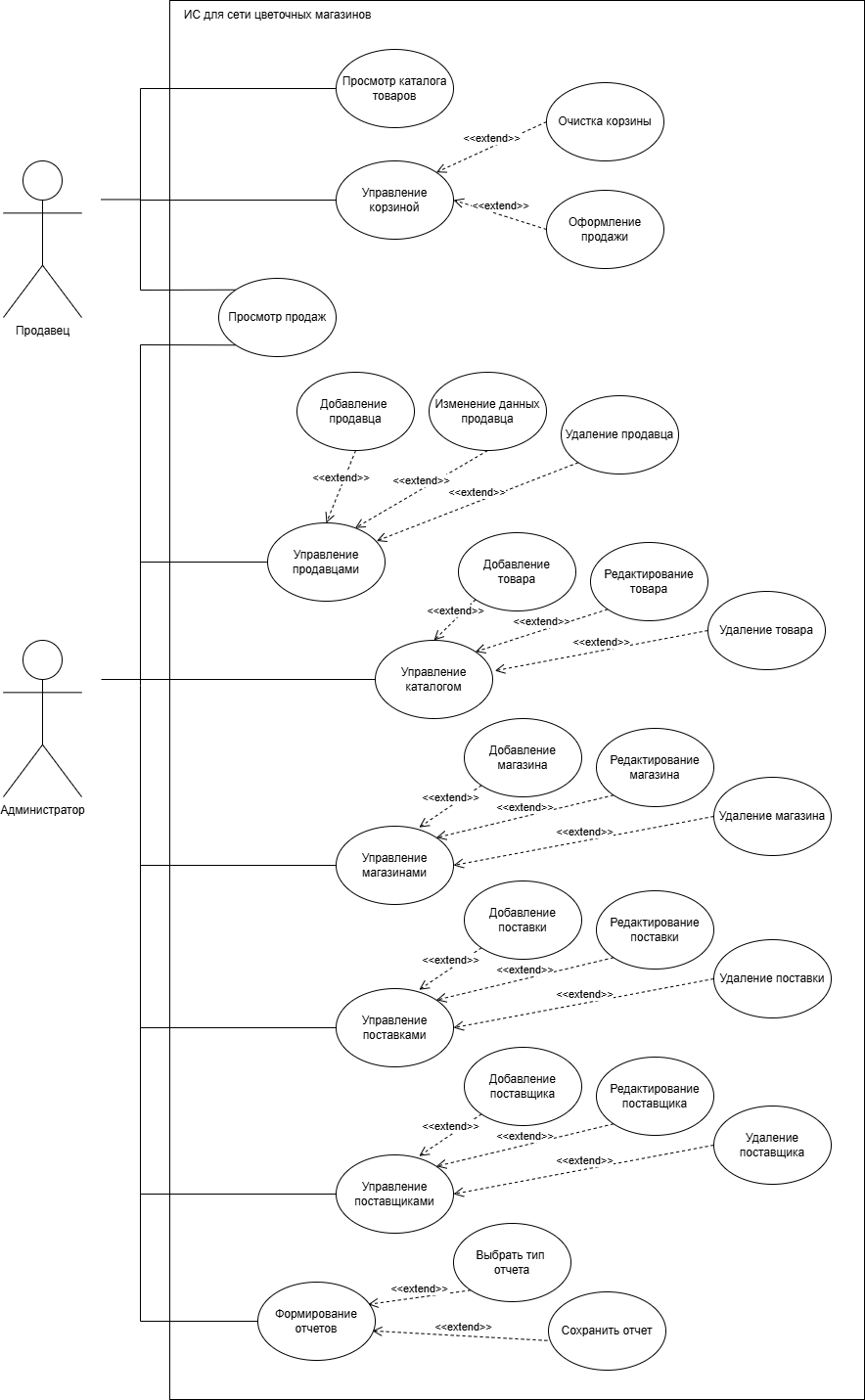


Рисунок 7 – Диаграмма прецедентов информационной системы

### Разработка базы данных

Перед реализацией базы данных для разрабатываемой системы были составлен перечень сущностей, а также их свойств. Данный перечень отображен в таблице 1.

Таблица 1 – Сущности базы данных

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Сущность | Свойство | Тип данных | Уникальность | Обязательно | Ограничения |
| cart | cart\_id | INT | Да | Да | Первичный ключ |
| tov\_id | INT | Нет | Да | Внешний ключ |
| quantity | INT | Нет | Да | >0 |
| created\_at | TIMESTAMP | Нет | Да | Дата |
| shop\_id | INT | Нет | Да | Внешний ключ |
| catalog | id\_tov | INT | Да | Да | Первичный ключ |
| name | VARCHAR | Нет | Да | 45 символов |
| shop | INT | Нет | Да | Внешний ключ |
| article | INT | Нет | Да | ≥ 0 |
| cena | DECIMAL | Нет | Да | Денежный формат |
| col | INT | Нет | Да | ≥ 0 |
| sale | sale\_id | INT | Да | Да | Первичный ключ |
| tov\_id | INT | Нет | Да | Внешний ключ |
| quantity | INT | Нет | Да | > 0 |
| cena | DECIMAL | Нет | Да | Денежный формат |

Продолжение таблицы 1

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | sale\_date | DATETIME | Нет | Да | Дата |
| shop\_id | INT | Нет | Да | Внешний ключ |
| seler\_id | INT | Нет | Да | Внешний ключ |
| seller | id\_pdv | INT | Да | Да | Первичный ключ |
| famio | VARCHAR | Нет | Да | 255 знаков |
| login | VARCHAR | Нет | Да | 100 знаков |
| password | VARCHAR | Нет | Да | 45 знаков |
| actor | VARCHAR | Нет | Да | 255 знаков |
| id\_mg | INT | Нет | Да | Внешний ключ |
| shop | id\_shopa | INT | Да | Да | Первичный ключ |
| name\_m | VARCHAR | Нет | Да | 45 знаков |
| adres\_m | VARCHAR | Нет | Да | 100 знаков |
| tel\_m | VARCHAR | Нет | Да | Формат номера телефона |
| dat\_sozd | DATE | Нет | Да | Дата |
| supplier | id\_post | INT | Да | Да | Первичный ключ |
| Org | VARCHAR | Нет | Да | 100 знаков |
| FIO | VARCHAR | Нет | Да | 100 знаков |
| tel | VARCHAR | Нет | Да | Формат номера телефона |
| supply | supply\_id | INT | Нет | Да | Внешний ключ |

Продолжение таблицы 1

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | tov\_id | INT | Нет | Да | Внешний ключ |
| quantity | INT | Нет | Да | ≥ 0 |
| cena | DECIMAL | Нет | Да | Денежный формат |
| shop\_id | INT | Нет | Да | Внешний ключ |
| supply\_date | DATETIME | Нет | Да | Дата |
| supply\_id | INT | Нет | Да | Внешний ключ |

Типы связей между сущностями:

* shop и catalog: один-ко-многим, один магазин (shop) может иметь несколько товаров в каталоге (catalog), но определенный товар привязан только к одному магазину;
* catalog и cart: один-ко-многим, один товар из каталога (catalog) может находиться в нескольких корзинах (cart), но каждая запись в корзине привязана к конкретному товару;
* shop и cart: один-ко-многим, в одном магазине (shop) может быть несколько корзин (cart), но каждая корзина принадлежит только одному магазину;
* catalog и sale: один-ко-многим, один товар из каталога (catalog) может быть продан несколько раз (sale), но каждая продажа привязана к конкретному товару;
* shop и sale: один-ко-многим, в одном магазине (shop) может быть много продаж (sale), но каждая продажа относится только к одному магазину;
* seller и sale: один-ко-многим, один продавец (seller) может совершать много продаж (sale), но каждая продажа привязана к одному продавцу;
* seller и shop: один-ко-многим, один магазин (shop) может иметь нескольких продавцов (seller), но каждый продавец работает только в одном магазине;
* supplier и supply: один-ко-многим, один поставщик (supplier) может делать много поставок (supply), но каждая поставка связана только с одним поставщиком;
* shop и supply: один-ко-многим, один магазин (shop) получает много поставок (supply), но каждая поставка относится только к одному магазину.

Для представления о структуре базы данных была разработана логическая модель в нотации IDEF1X. Данная модель представлена на рисунке 8.

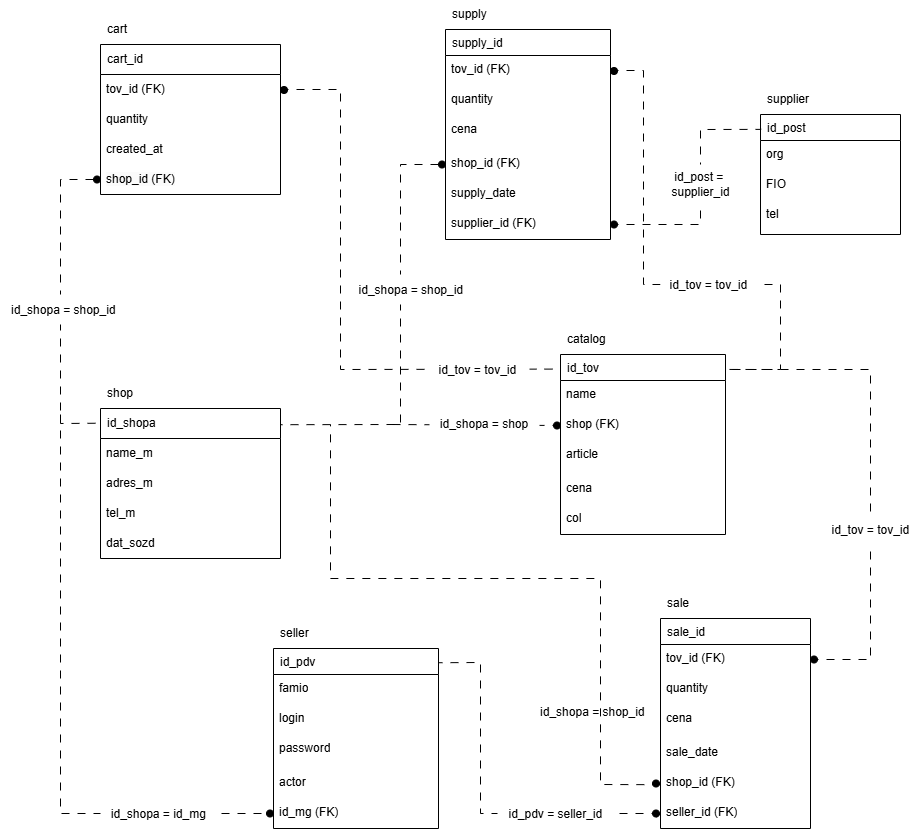


Рисунок 8 – Логическая модель базы данных

В ходе выполнения работы была выбрана СУБД MySQL. MySQL — это одна из самых популярных реляционных систем управления базами данных, которая часто используется благодаря своей простоте, высокой производительности и широкой поддержке. В контексте выбора подходящей СУБД для курсовой работы MySQL становится оптимальным решением, если учитывать ее особенности, функциональность и области применения.

Прежде всего, MySQL выделяется своей доступностью. Эта СУБД кроссплатформенна, что означает возможность ее развертывания как на Windows, так и на Linux или macOS. Это упрощает разработку и позволяет использовать одну и ту же базу данных на разных операционных системах.

Одним из главных преимуществ MySQL является простота ее освоения. Интуитивно понятная установка и настройка, а также огромный объем документации и руководств делают эту СУБД особенно привлекательной для начинающих. Даже пользователи с минимальным опытом работы с базами данных могут быстро освоить MySQL, настроить сервер и начать работать с данными.

В сравнении с другими популярными СУБД, такими как PostgreSQL, SQLite, MySQL предлагает наиболее сбалансированное сочетание простоты, производительности и функциональности. PostgreSQL, например, превосходит MySQL по функциональным возможностям и подходит для сложных аналитических задач, но ее настройка и освоение требуют больше времени и опыта. SQLite, несмотря на свою легковесность, не рассчитана на сложные или масштабные приложения.

Таким образом, MySQL становится логичным выбором благодаря своей универсальности, легкости использования и стабильности. Это СУБД, которая позволяет сосредоточиться на самой работе с данными, а не на сложной настройке или изучении специфических инструментов.

Описания полей созданных таблиц предоставлены в таблице 2.

Таблица 2 – Структура физической модели данных

|  |  |  |  |
| --- | --- | --- | --- |
| Таблица | Атрибут | Тип | Размер |
| cart | cart\_id | INT |  |
| tov\_id | INT |  |
| quantity | INT |  |
| created\_at | TIMESTAMP |  |
| shop\_id | INT |  |
| catalog | id\_tov | INT |  |
| name | VARCHAR | 45 |
| shop | INT |  |
| article | INT |  |
| cena | DECIMAL | 10, 2 |
| col | INT |  |
| sale | sale\_id | INT |  |
| tov\_id | INT |  |
| quantity | INT |  |
| cena | DECIMAL | 10, 2 |
| sale\_date | DATETIME |  |
| shop\_id | INT |  |
| seler\_id | INT |  |
| seller | id\_pdv | INT |  |

Продолжение таблицы 2

|  |  |  |  |
| --- | --- | --- | --- |
|  | famio | VARCHAR | 255 |
| login | VARCHAR | 45 |
| password | VARCHAR | 10 |
| actor | ENUM('Администратор', 'Продавец') |  |
| id\_mg | INT |  |
| shop | id\_shopa | INT |  |
| name\_m | VARCHAR | 45 |
| adres\_m | VARCHAR | 100 |
| tel\_m | VARCHAR | 17 |
| dat\_sozd | DATE |  |
| supplier | id\_post | INT |  |
| Org | VARCHAR | 100 |
| FIO | VARCHAR | 100 |
| tel | VARCHAR | 17 |
| supply | supply\_id | INT |  |
| tov\_id | INT |  |
| quantity | INT |  |
| cena | DECIMAL | 10, 2 |
| shop\_id | INT | 10 |

Продолжение таблицы 2

|  |  |  |  |
| --- | --- | --- | --- |
|  | supply\_date | DATETIME |  |
| supply\_id | INT |  |

А также разработана физическая модель базы данных в нотации IDEF1X, которая представлена на рисунке 9.

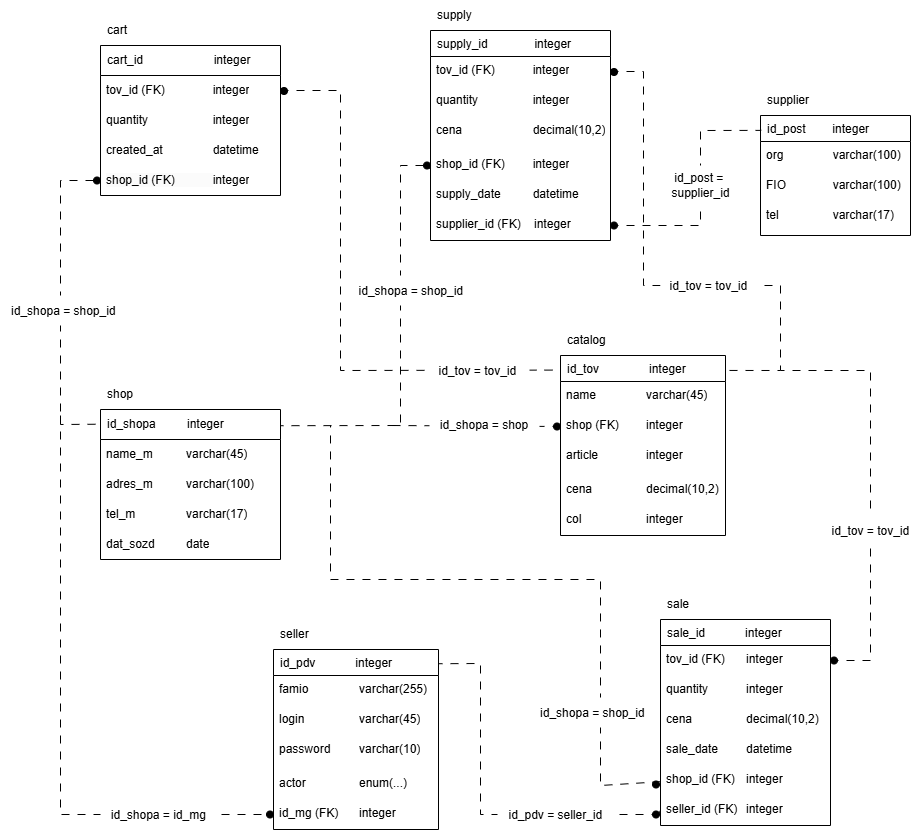


Рисунок 9 – Физическая модель базы данных

# РЕАЛИЗАЦИЯ ИНФОРМАЦИОННОЙ СИСТЕМЫ

В рамках проектирования информационной системы для сети цветочных магазинов были разработаны прототипы экранных форм интерфейса, которые представлены в Приложении А «Техническое задание».

Ниже представлены результаты пользовательского интерфейса приложения.

Входное окно представлено на рисунке 10.

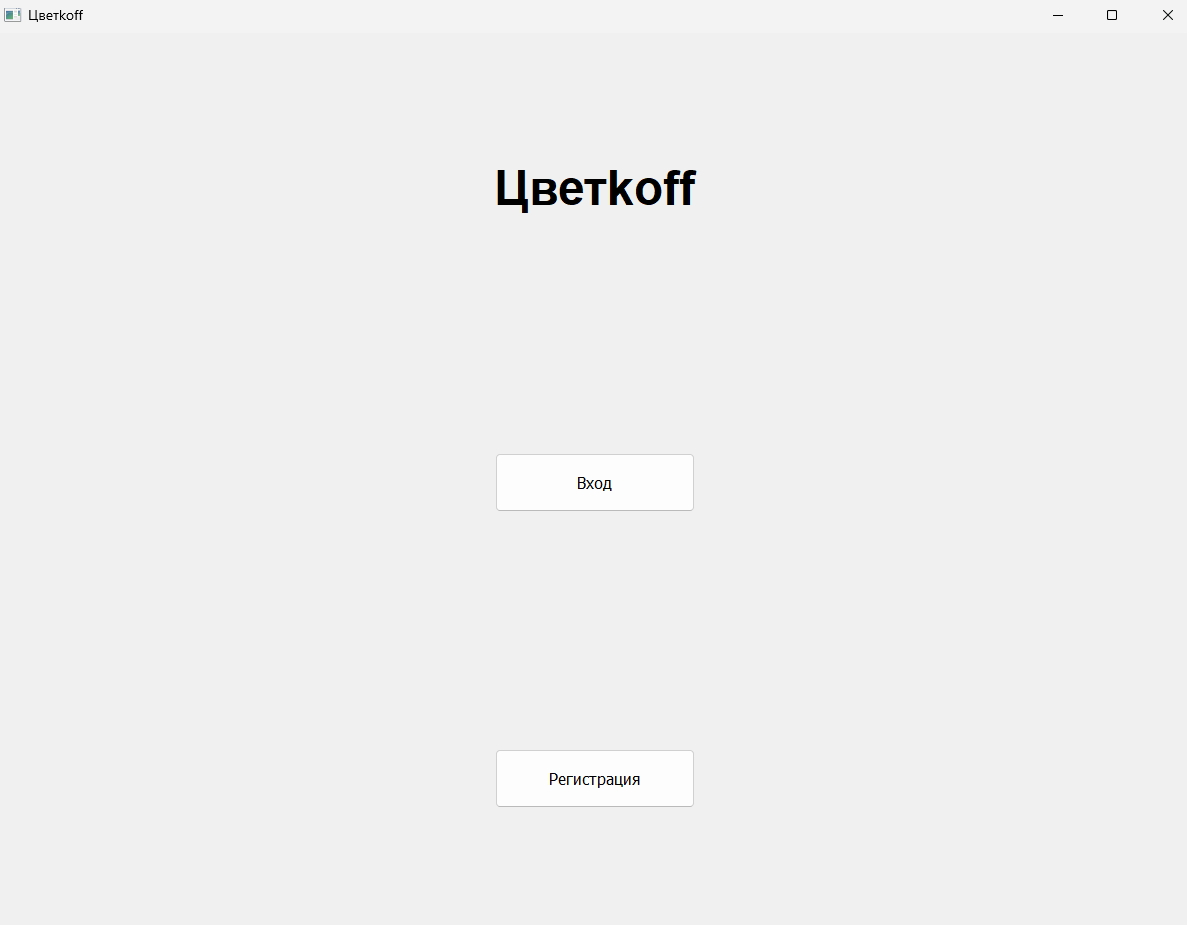


Рисунок 10 – Экранная форма входного окна

В данном окне предоставлен выбор войти в систему с использованием существующих логина и пароля или зарегистрироваться в системе, автоматически предоставленная роль будет «Продавец»

Окно регистрации представлено на рисунке 11.

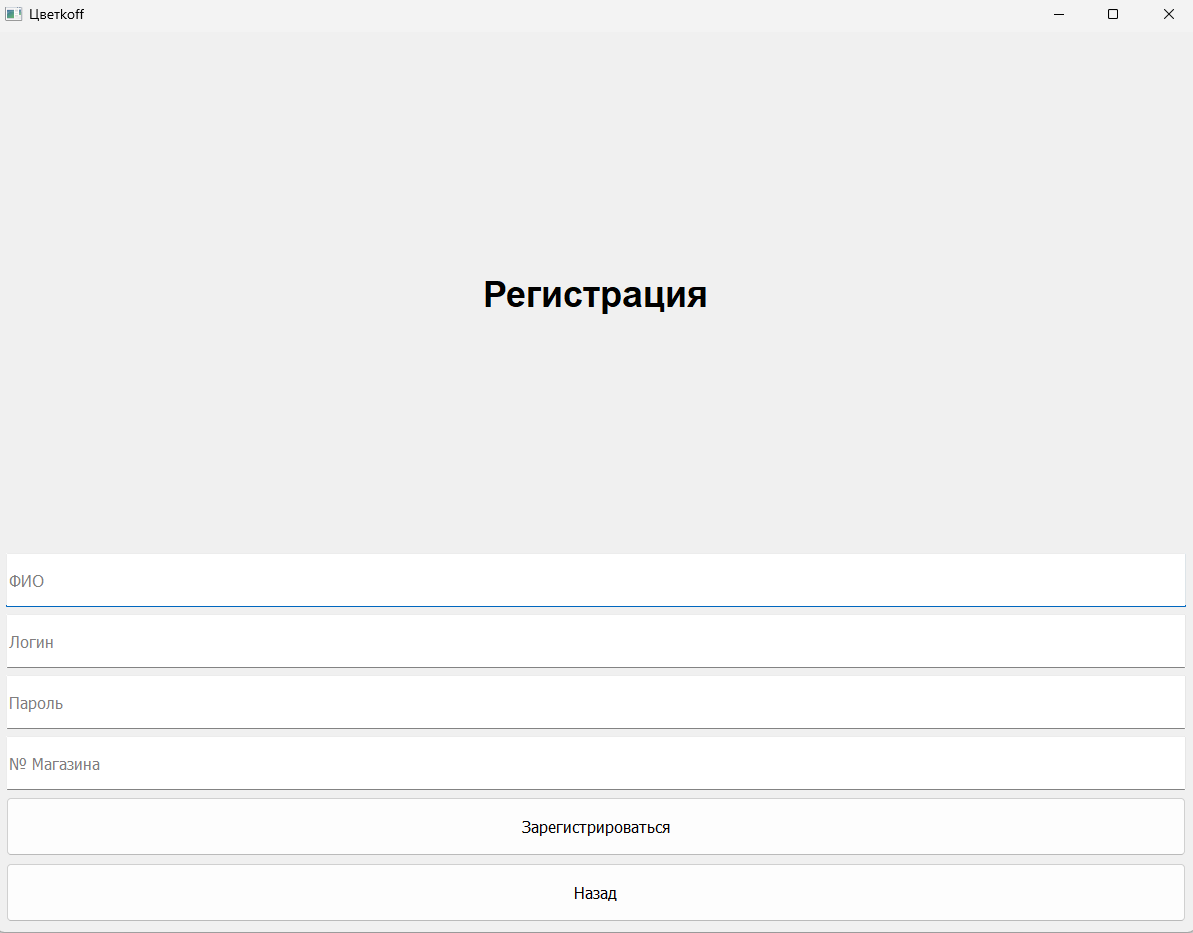


Рисунок 11 – Экранная форма регистрации

На данном этапе пользователю предлагается создать нового пользователя системы (в роли продавца) для дальнейшей работы с ИС.

При успешной регистрации пользователя, ему приписывается роль «продавца» в системе и его возвращают на входное окно для входа.

При нажатии кнопки «Вход» во входном окне пользователю открывается экранная форма входа в систему.

После ввода корректных логина и пароля пользователю открыта работа в системе в роли, прописанной в разделе «Продавцы».

Окно входа представлено на рисунке 12.

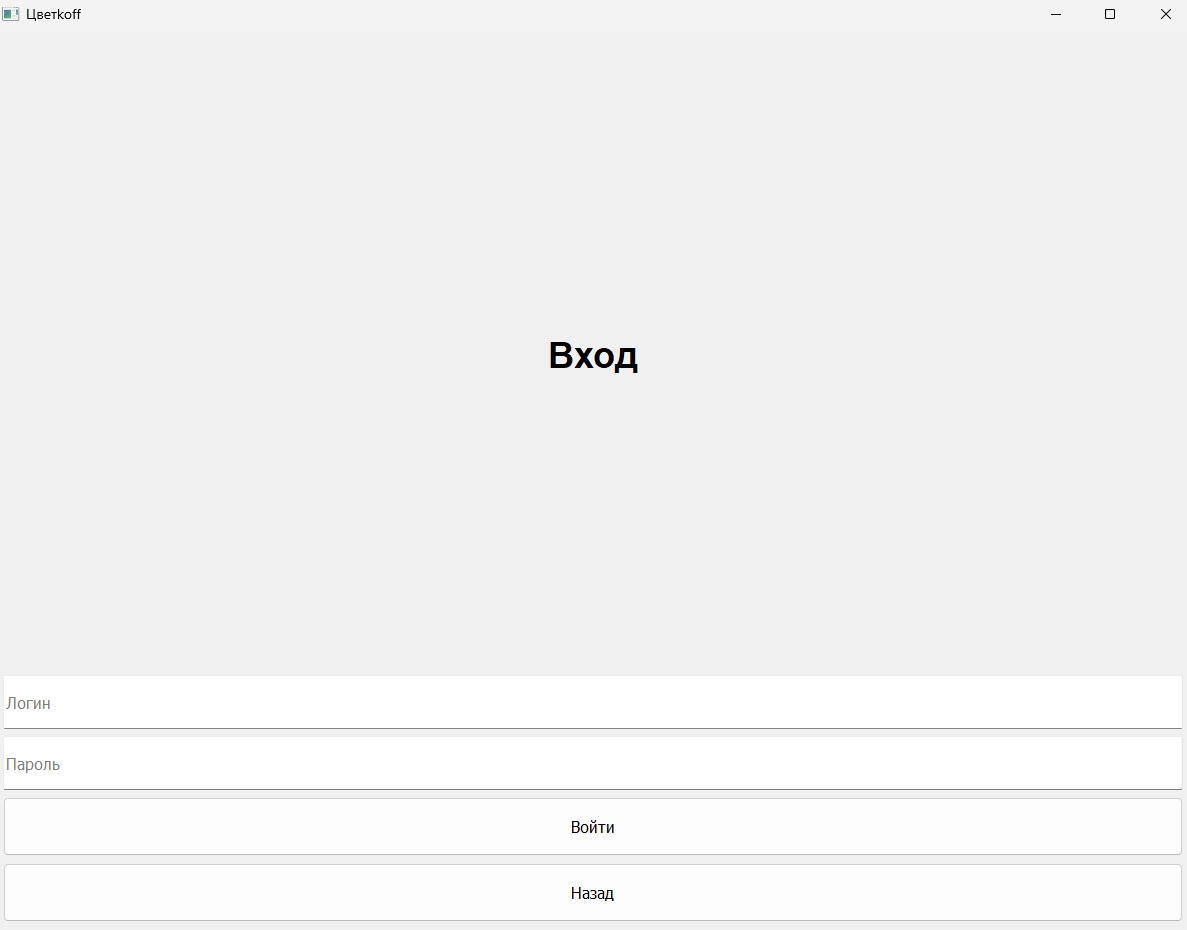


Рисунок 12 – Экранная форма окна входа

Экранная форма раздела «Каталог» – раздела ИС в котором отображаются все товары.

Окно раздела «Каталог» для администратора представлено на рисунке 13.

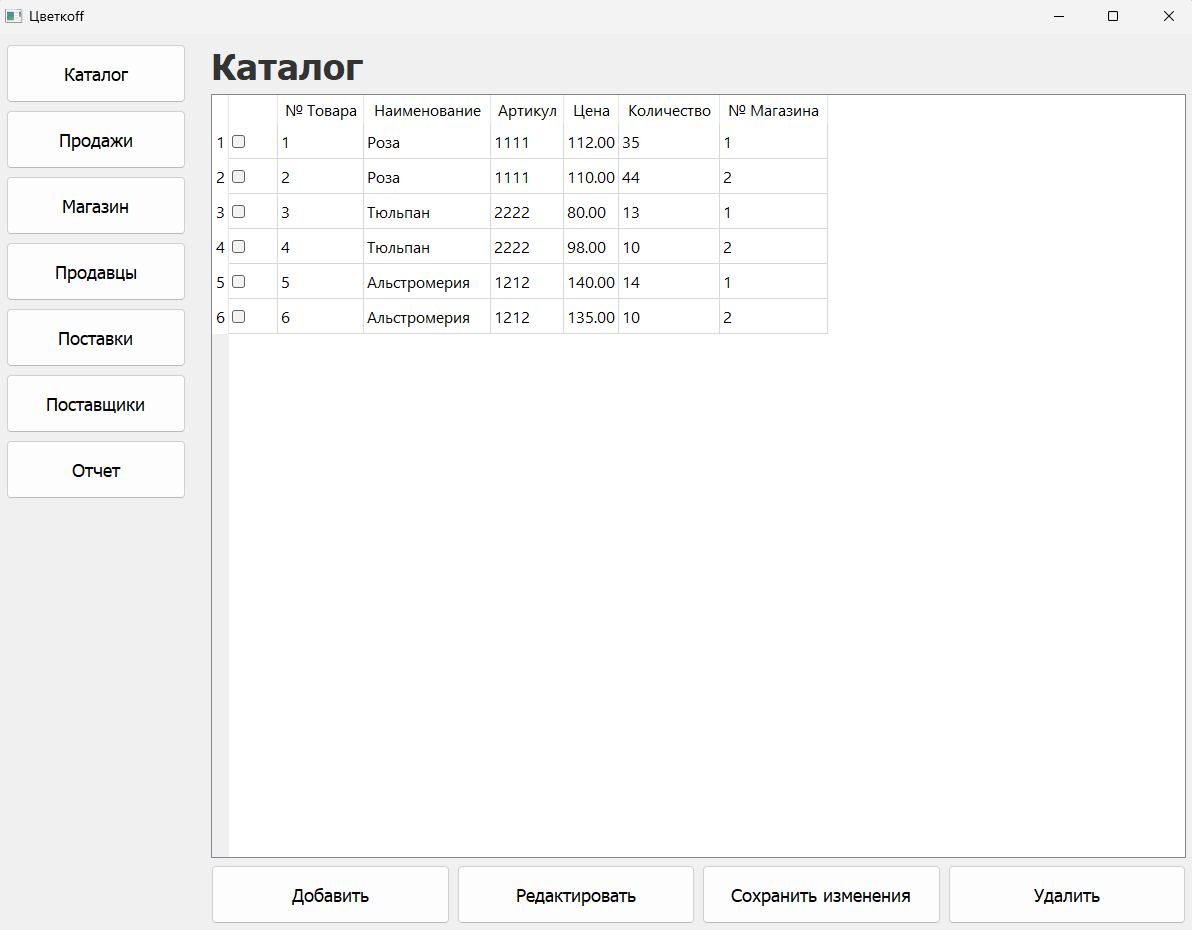


Рисунок 13 – Экранная форма раздела «Каталог» для администратора

На данной странице администратор может добавлять, редактировать и удалять выбранные товары. Продавец в свою очередь видит товары только своего магазина, где он работает и кнопка «В корзину».

Окно раздела «Каталог» для продавца представлено на рисунке 14.

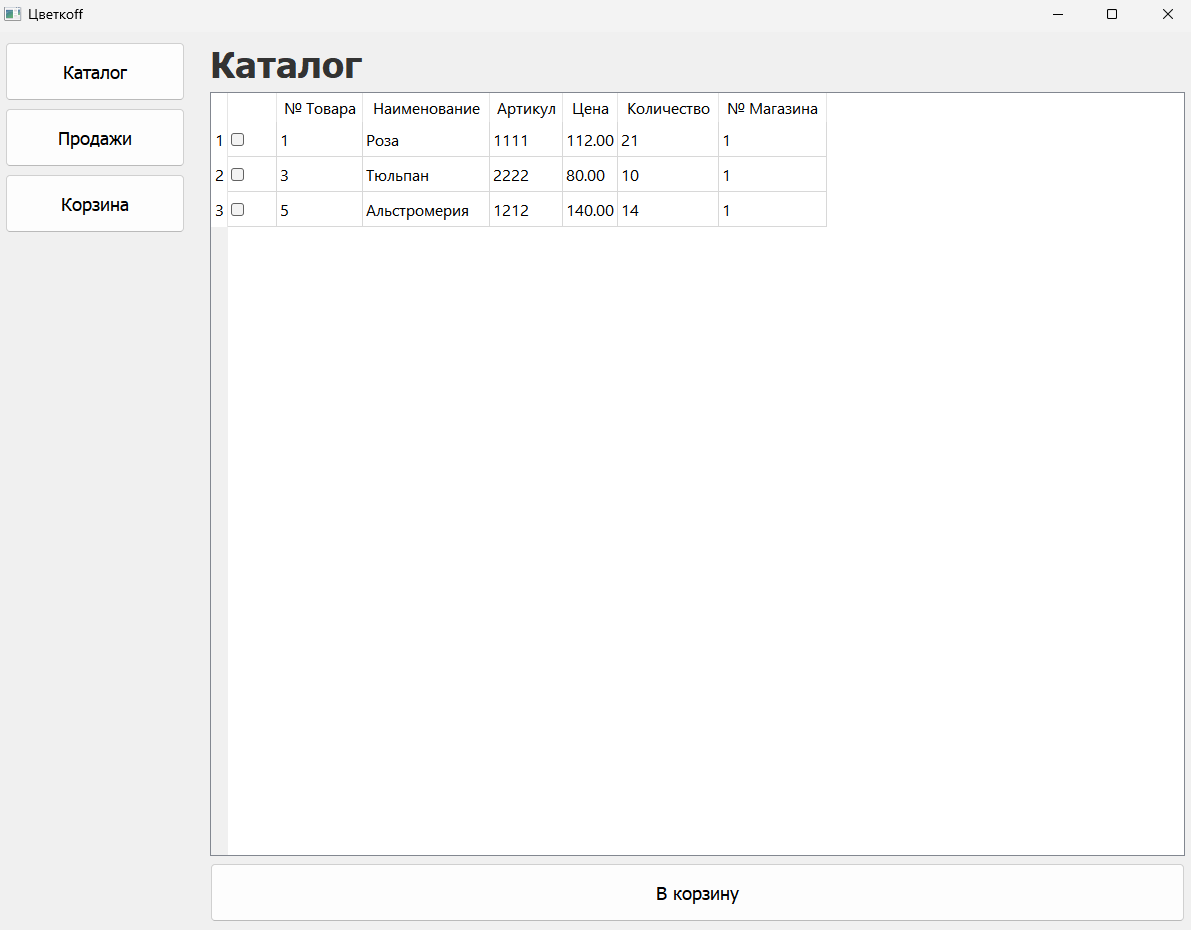


Рисунок 14 – Экранная форма раздела «Каталог» для продавца

Экранная форма раздела «Продажи» для администратора представлена на рисунке 15.

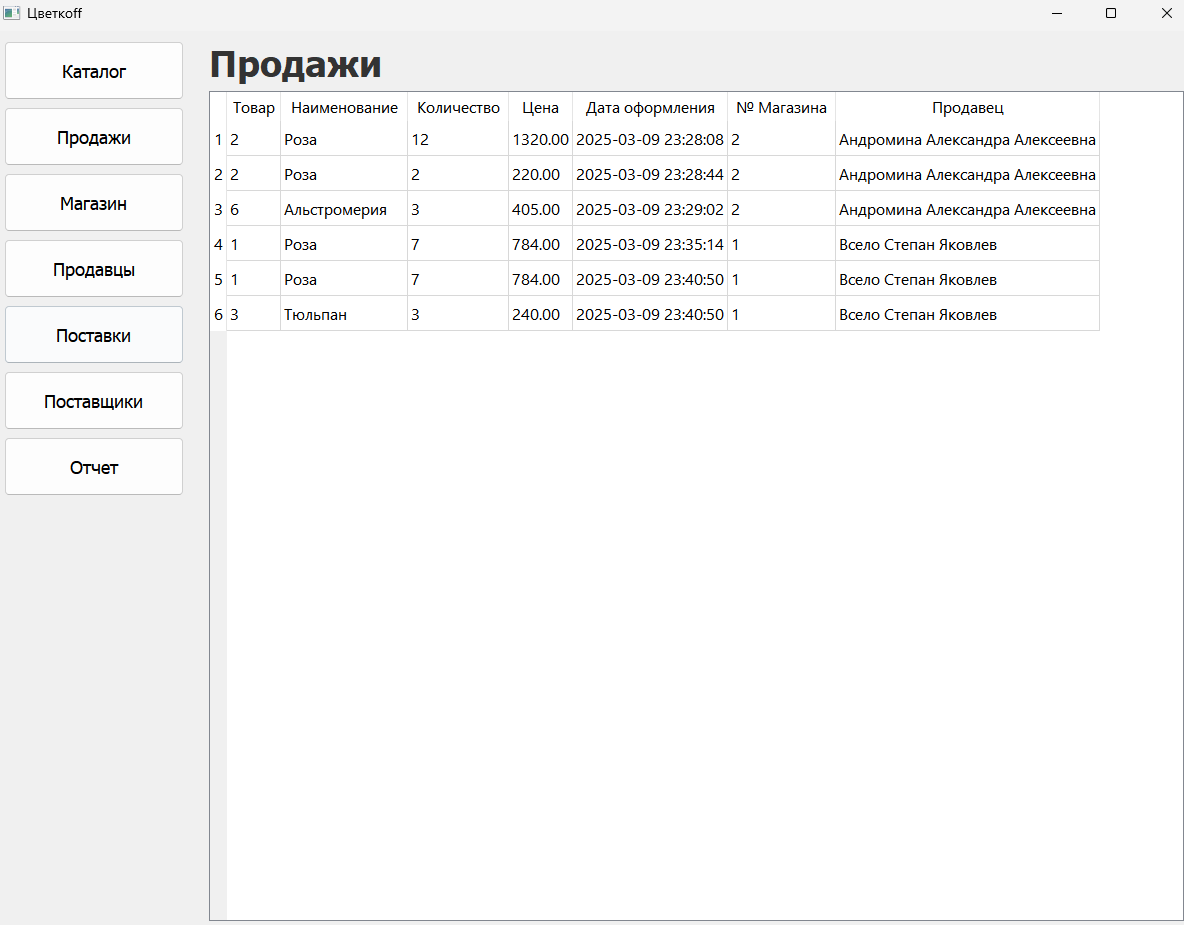


Рисунок 15 – Экранная форма раздела «Продажи» для администратора

Администратор может видеть продажи по всей сети магазинов. Продавец, в свою очередь, может смотреть список продаж только того магазина, в котором работает.

Для оформления продажи продавец должен выбрать в каталоге товары и добавить их в корзину, затем оформить продажу при помощи кнопки «Оплата».

Экранная форма раздела «Корзина», доступная только для продавца для осуществления продаж, представлена на рисунке 16.

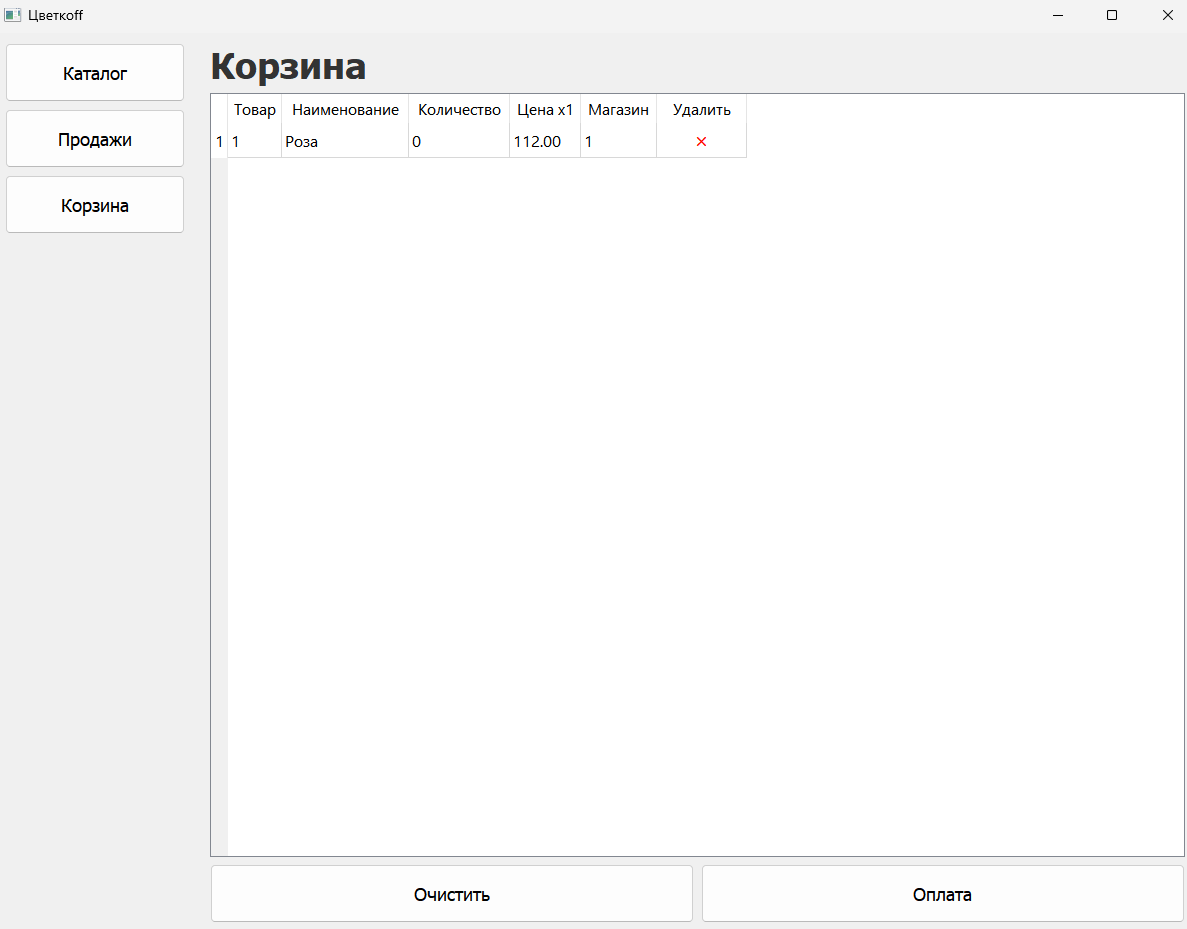


Рисунок 16 – Экранная форма раздела «Корзина»

Экранная форма раздела «Магазин» представлена на рисунке 17.

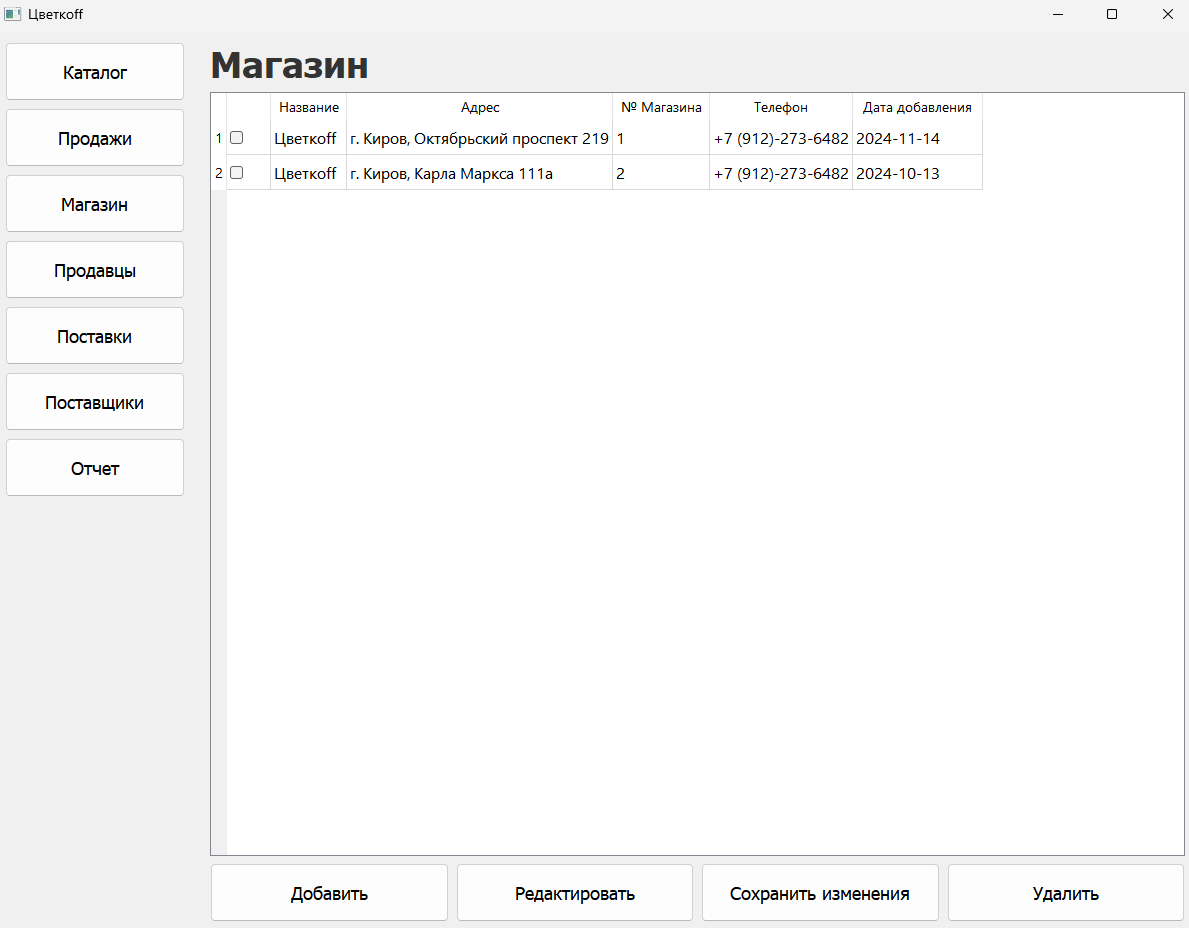


Рисунок 17 – Экранная форма раздела «Магазин»

На данной странице администратор может добавлять, редактировать и удалять выбранные магазины.

Для каждого магазина администратор может назначить своих продавцов, а также администратора, доверенное лицо, которое получит такой же полный доступ к системе.

Экранная форма раздела «Продавцы» представлена на рисунке 18.

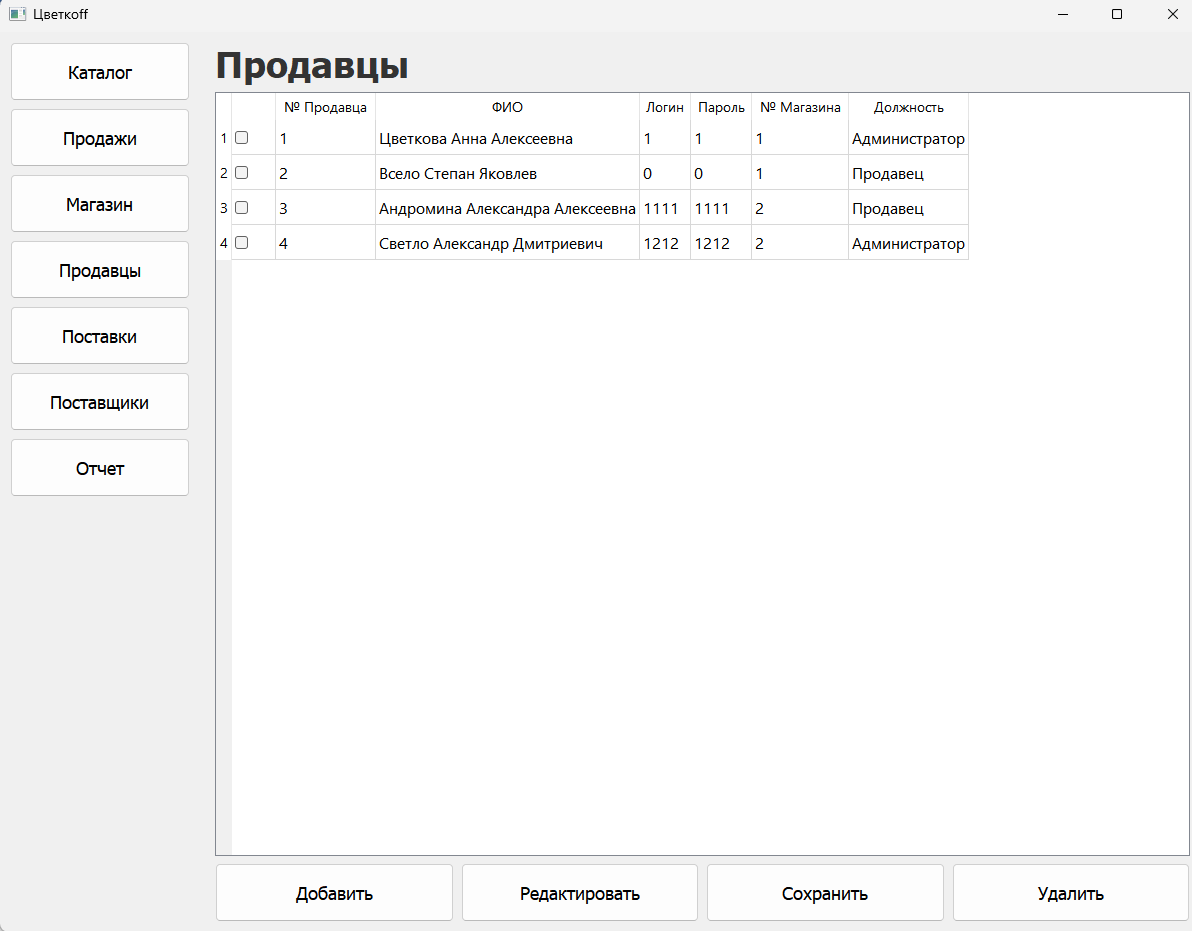


Рисунок 18 – Экранная форма раздела «Продавцы»

На данной странице администратор может добавлять, редактировать и удалять записи о продавцах. Назначать роль и магазин в системе.

Следующий по списку раздел «Поставки», который позволяет восполнять запасы склада каждого магазина.

Экранная форма раздела «Поставки» представлена на рисунке 19.

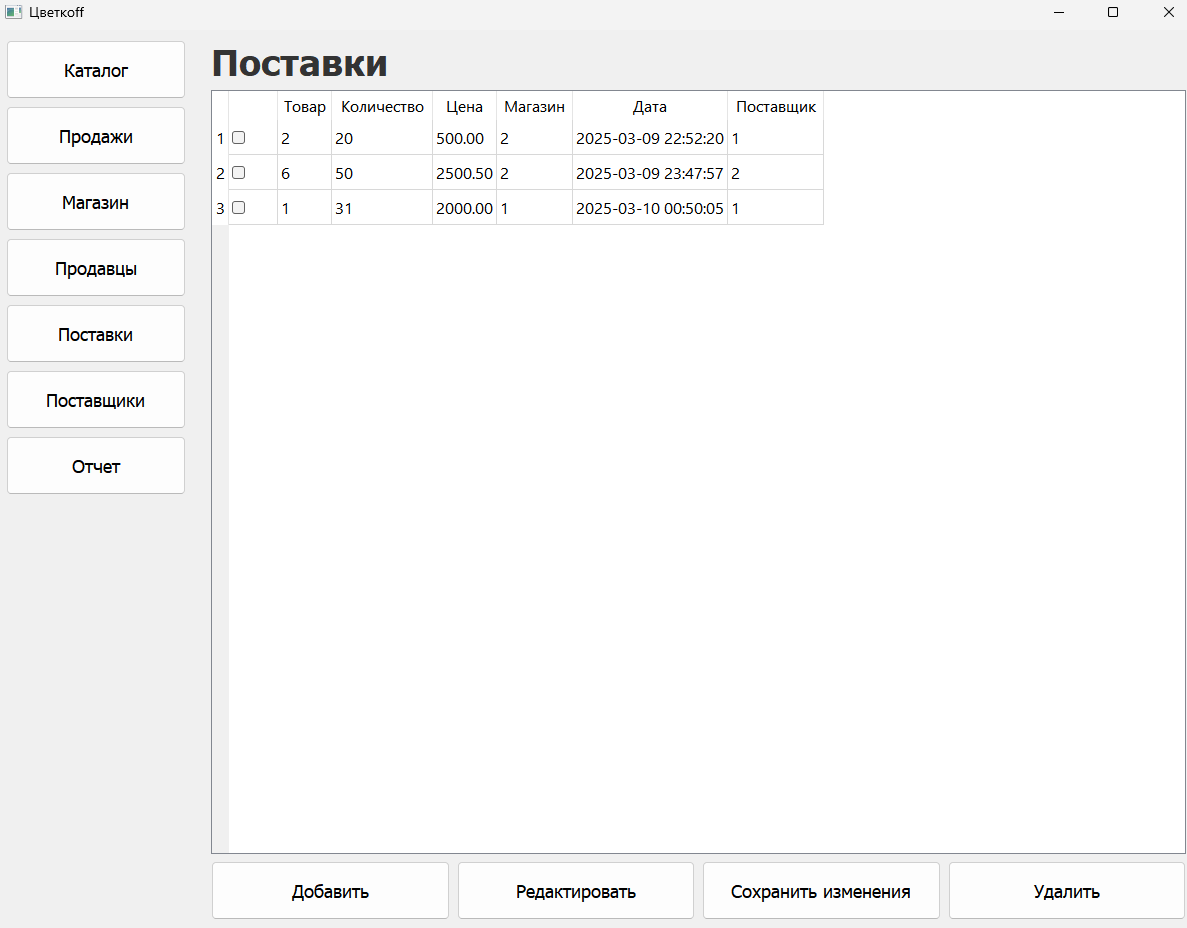


Рисунок 19 – Экранная форма раздела «Поставки»

Экранная форма раздела «Поставщики» представлена на рисунке 20.

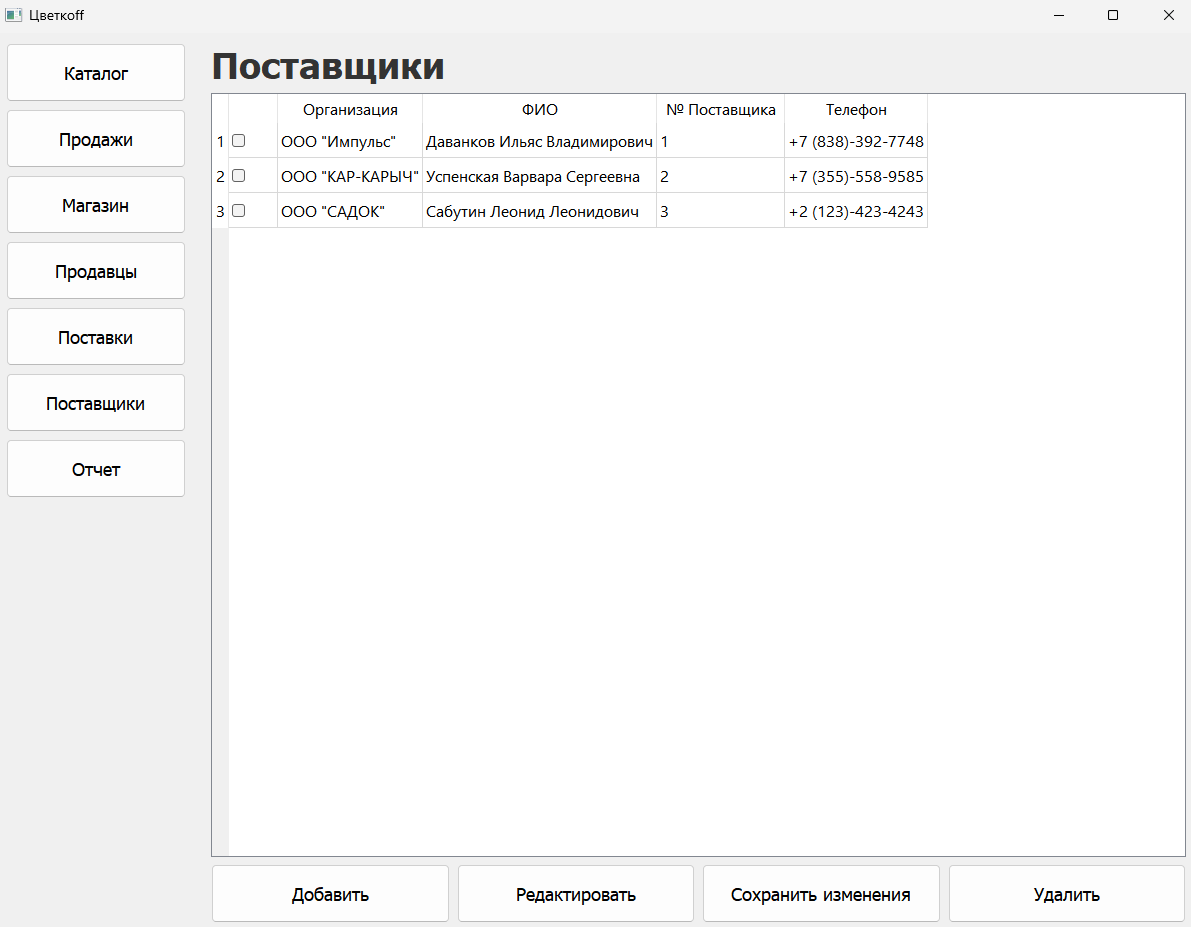


Рисунок 20 – Экранная форма раздела «Поставщики»

На данной странице администратор может добавлять, редактировать и удалять записи о поставщиках.

Далее по списку идет раздел «Отчет», в котором администратор может выбрать тип отчета, а именно доход или расход, посмотреть его и сохранить в формате PDF.

Экранная форма раздела «Отчет» представлена на рисунке 21.

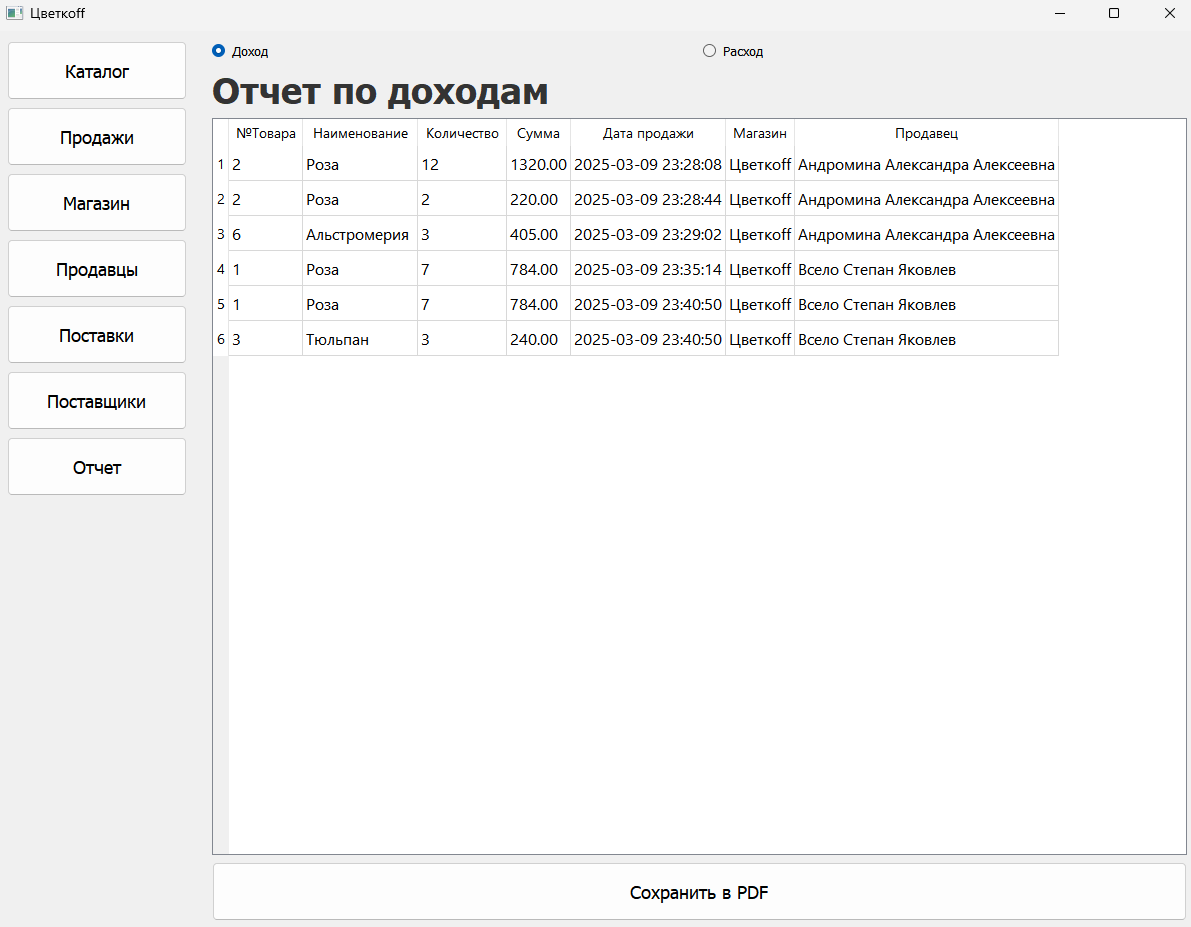


Рисунок 21 – Экранная форма раздела «Отчет»

# ЗАКЛЮЧЕНИЕ

В ходе выполнения курсового проекта была разработана и подробно описана информационная система, предназначенная для сети цветочных магазинов. Проект охватывает все этапы от проектирования до реализации, предоставляя интегрированный и автоматизированный подход к управлению всеми аспектами бизнеса.

Разработанная информационная система целенаправленно соответствует уникальным потребностям сети цветочных магазинов, обеспечивая точное управление товаром, продажами, поставками, продавцами и магазинами, поставщиками и отчетностью. В ходе проектирования были учтены особенности сферы.

Так же были получены навыки построение DFD диаграмм, Use Case диаграмм, логических моделей данных в нотации IDEF1X, физических моделей данных и диаграмм в нотации IDEF0.

# СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. ГОСТ 34.602–2020. Информационная технология. Комплекс стандартов на автоматизированные системы. Техническое задание на создание автоматизированной системы: национальный стандарт Российской Федерации: издание официальное: утвержден и введен в действие Приказом Федерального агентства по техническому регулированию и метрологии от 18 ноября 2020 г. № 850-ст: введен впервые: дата введения 2021-06-01. - Москва: Стандартинформ, 2020. - 32 с. - Текст непосредственный.
2. Что такое DFD (Диаграммы потоков данных) [Электронный ресурс] – URL: <https://trinion.org/blog/chto-takoe-dfd-diagrammy-potokov-dannykh> (дата обращения: 25.12.2024)
3. Лутц, М. Изучаем Python. Том 1. Основы программирования: учебное пособие / М. Лутц. - 5-е изд. - Москва: Бином, 2020. - 864 с. - ISBN 978-5-00101-154-5. - Текст непосредственный.
4. Саймон, Б. DFD: построение диаграмм потоков данных. Руководство / Б. Саймон. - Текст: электронный // Habr.com [сайт]. - 2023. - URL: https://habr.com/ru/articles/dfd (дата обращения: 22.12.2024).
5. Ким, Д. IDEF1X: нотация для проектирования баз данных: учебное пособие / Д. Ким. - Москва: Академия, 2019. - 240 с. - ISBN 978-5-4461-1495-0. - Текст непосредственный.
6. Документация по UML: официальная спецификация OMG [электронный ресурс]. - URL: https://www.omg.org/spec/UML/About-UML/ (дата обращения: 20.12.2024).
7. Документация MySQL [электронный ресурс]. — URL: <https://dev.mysql.com/doc/> (дата обращения: 12.11.2024).

# ПРИЛОЖЕНИЕ А

**ТЕХНИЧЕСКОЕ ЗАДАНИЕ**

# ПРИЛОЖЕНИЕ Б

**ИСХОДНЫЙ КОД**

**Код Entry.py**

import mysql.connector, pymysql  
from PyQt5.QtWidgets import (  
 QApplication, QMainWindow, QPushButton, QVBoxLayout, QHBoxLayout, QWidget, QLabel, QLineEdit, QMessageBox  
)  
from PyQt5.QtCore import Qt  
  
class MainWindow(QMainWindow):  
 def \_\_init\_\_(self):  
 super().\_\_init\_\_()  
 self.setWindowTitle('Цветkoff')  
 self.setMinimumSize(1200, 900)  
 self.resize(1200, 900)  
  
 self.actor = None *# Роль пользователя (продавец/администратор)* self.seller\_id = None *# ID продавца* self.setStyleSheet("""  
 QPushButton { height: 50px; font-size: 16px; }  
 QLabel { font-size: 52px; font-family: Arial; }  
 QLineEdit { font-size: 16px; height: 50px; }  
 """)  
 self.show\_initial\_window()  
  
 def show\_initial\_window(self):  
 *"""Создание начального окна с кнопками Вход и Регистрация"""* layout = QVBoxLayout()  
  
 title\_label = QLabel('Цветkoff')  
 title\_label.setAlignment(Qt.AlignCenter)  
 title\_label.setStyleSheet("font-size: 48px; font-weight: bold;")  
 layout.addWidget(title\_label)  
  
 self.login\_button = QPushButton('Вход')  
 self.login\_button.setMinimumSize(200, 50)  
 self.login\_button.clicked.connect(self.show\_login\_window)  
 layout.addWidget(self.login\_button, alignment=Qt.AlignCenter)  
  
 self.register\_button = QPushButton('Регистрация')  
 self.register\_button.setMinimumSize(200, 50)  
 self.register\_button.clicked.connect(self.show\_register\_window)  
 layout.addWidget(self.register\_button, alignment=Qt.AlignCenter)  
  
 central\_widget = QWidget()  
 central\_widget.setLayout(layout)  
 self.setCentralWidget(central\_widget)  
 self.show() *# Гарантируем, что окно обновится* def show\_login\_window(self):  
 *"""Окно входа с полями для логина и пароля"""* layout = QVBoxLayout()  
  
 login\_title = QLabel('Вход')  
 login\_title.setAlignment(Qt.AlignCenter)  
 login\_title.setStyleSheet("font-size: 36px; font-weight: bold;")  
 layout.addWidget(login\_title)  
  
 self.login\_input = QLineEdit()  
 self.login\_input.setPlaceholderText('Логин')  
 layout.addWidget(self.login\_input)  
  
 self.password\_input = QLineEdit()  
 self.password\_input.setPlaceholderText('Пароль')  
 self.password\_input.setEchoMode(QLineEdit.Password)  
 layout.addWidget(self.password\_input)  
  
 self.login\_action\_button = QPushButton('Войти')  
 self.login\_action\_button.clicked.connect(self.login)  
 layout.addWidget(self.login\_action\_button)  
  
 self.back\_button = QPushButton('Назад')  
 self.back\_button.clicked.connect(self.show\_initial\_window)  
 layout.addWidget(self.back\_button)  
  
 login\_widget = QWidget()  
 login\_widget.setLayout(layout)  
 self.setCentralWidget(login\_widget)  
  
 def show\_register\_window(self):  
 *"""Окно регистрации пользователя"""* layout = QVBoxLayout()  
  
 title\_label = QLabel('Регистрация')  
 title\_label.setAlignment(Qt.AlignCenter)  
 title\_label.setStyleSheet("font-size: 36px; font-weight: bold;")  
 layout.addWidget(title\_label)  
  
 self.name\_input = QLineEdit()  
 self.name\_input.setPlaceholderText('ФИО')  
 layout.addWidget(self.name\_input)  
  
 self.login\_input = QLineEdit()  
 self.login\_input.setPlaceholderText('Логин')  
 layout.addWidget(self.login\_input)  
  
 self.password\_input = QLineEdit()  
 self.password\_input.setPlaceholderText('Пароль')  
 self.password\_input.setEchoMode(QLineEdit.Password)  
 layout.addWidget(self.password\_input)  
  
 self.shop\_input = QLineEdit()  
 self.shop\_input.setPlaceholderText('№ Магазина')  
 layout.addWidget(self.shop\_input)  
  
 register\_button = QPushButton('Зарегистрироваться')  
 register\_button.clicked.connect(self.register\_user)  
 layout.addWidget(register\_button)  
  
 back\_button = QPushButton('Назад')  
 back\_button.clicked.connect(self.show\_initial\_window)  
 layout.addWidget(back\_button)  
  
 container = QWidget()  
 container.setLayout(layout)  
 self.setCentralWidget(container)  
  
 def register\_user(self):  
 *"""Регистрация пользователя в базе данных"""* name = self.name\_input.text().strip()  
 login = self.login\_input.text().strip()  
 password = self.password\_input.text().strip()  
 shop\_id = self.shop\_input.text().strip()  
 id\_pdv = self.\_input.text().strip()  
  
 *# Проверка на пустые поля* if not (name and login and password and shop\_id):  
 self.show\_message('Ошибка', 'Все поля обязательны для заполнения!')  
 return  
  
 *# Проверка длины строк* if len(name) > 255:  
 self.show\_message('Ошибка', 'ФИО должно быть не длиннее 255 символов.')  
 return  
 if len(login) > 45:  
 self.show\_message('Ошибка', 'Логин должен быть не длиннее 45 символов.')  
 return  
 if len(password) > 10:  
 self.show\_message('Ошибка', 'Пароль должен быть не длиннее 10 символов.')  
 return  
  
 *# Проверка, что shop\_id является целым числом* try:  
 shop\_id = int(shop\_id)  
 except ValueError:  
 self.show\_message('Ошибка', 'Номер магазина должен быть целым числом.')  
 return  
  
 try:  
 conn = mysql.connector.connect(  
 host='127.0.0.1',  
 user='root',  
 password='Cstud161620',  
 database='flowersh',  
 autocommit=True  
 )  
  
 cursor = conn.cursor()  
  
 *# Проверка уникальности логина (без учета регистра)* cursor.execute("SELECT 1 FROM seller WHERE LOWER(login) = LOWER(%s)", (login,))  
 if cursor.fetchone():  
 self.show\_message('Ошибка', 'Пользователь с таким логином уже существует!')  
 return  
  
 *# Проверка существования магазина* cursor.execute("SELECT 1 FROM shop WHERE id\_shopa = %s", (shop\_id,))  
 if not cursor.fetchone():  
 self.show\_message('Ошибка', 'Магазин с указанным номером не существует!')  
 return  
  
 *# Регистрация пользователя* cursor.execute("""  
 INSERT INTO seller (famio, login, password, id\_mg, actor)  
 VALUES (%s, %s, %s, %s, 'Продавец')  
 """, (name, login, password, shop\_id))  
  
 self.show\_message('Успех', 'Регистрация прошла успешно!')  
 self.show\_initial\_window() *# Возвращаем пользователя на экран входа* except mysql.connector.Error as err:  
 self.show\_message('Ошибка', f'Ошибка базы данных: {err}')  
  
 finally:  
 if cursor:  
 cursor.close()  
 if conn:  
 conn.close()  
  
 def login(self):  
 *"""Проверка логина и пароля с использованием базы данных MySQL"""* login = self.login\_input.text().strip()  
 password = self.password\_input.text().strip()  
  
 *# Проверка на пустые поля* if not login or not password:  
 self.show\_message("Ошибка", "Пожалуйста, введите логин и пароль.")  
 return  
  
 *# Проверка длины строк* if len(login) > 45:  
 self.show\_message("Ошибка", "Логин должен быть не длиннее 45 символов.")  
 return  
 if len(password) > 10:  
 self.show\_message("Ошибка", "Пароль должен быть не длиннее 10 символов.")  
 return  
  
 try:  
 *# Подключение к базе данных* conn = mysql.connector.connect(  
 host='127.0.0.1',  
 user='root',  
 password='Cstud161620',  
 database='flowersh'  
 )  
 cursor = conn.cursor()  
  
 *# Получаем данные о продавце, включая его id\_pdv* cursor.execute("SELECT id\_pdv, actor FROM seller WHERE login = %s AND password = %s", (login, password))  
 result = cursor.fetchone()  
  
 if result:  
 self.seller\_id = result[0] *# Сохраняем id\_pdv продавца* self.actor = result[1].lower() *# Сохраняем роль* self.close() *# Закрываем окно входа* else:  
 self.show\_message("Ошибка", "Неверные данные")  
  
 except mysql.connector.Error as err:  
 self.show\_message("Ошибка", f"Ошибка базы данных: {err}")  
  
 finally:  
 if 'cursor' in locals():  
 cursor.close()  
 if 'conn' in locals():  
 conn.close()  
  
 def show\_message(self, title, message):  
 *"""Показывает сообщение пользователю"""* msg\_box = QMessageBox()  
 msg\_box.setWindowTitle(title)  
 msg\_box.setText(message)  
 msg\_box.exec\_()  
  
 def closeEvent(self, event):  
 *"""Закрытие программы при закрытии окна"""* if self.actor is None: *# Если пользователь не вошел* QApplication.quit() *# Завершаем приложение* event.accept() *# Корректное завершение без ошибок*if \_\_name\_\_ == '\_\_main\_\_':  
 app = QApplication([])  
 window = MainWindow()  
 if window.actor is not None: *# Показываем главное окно только если роль определена* window.show()  
 app.exec\_()