

**Колледж космического машиностроения и технологий**

**КУРСОВОЙ ПРОЕКТ**

по дисциплине МДК.02.02. Технология разработки и защиты баз данных

**РАЗРАБОТКА СИСТЕМЫ УПРАВЛЕНИЯ ЦВЕТОЧНЫМ МАГАЗИНОМ «РОМАШКА»**

Пояснительная записка

КП.09.02.03.23.13ПЗ

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Обучающийся группы | | П1-19 | Новоселова В.А. | |
| Руководитель курсового проекта | | | Стрельников С.Д. | |
| Результат защиты |  | | |  |

г. Королёв, 2023 г.

**ОГЛАВЛЕНИЕ**

[ВВЕДЕНИЕ 4](#_Toc131697157)

[1. ТЕОРЕТИЧЕСКАЯ ЧАСТЬ 5](#_Toc131697158)

[1.1. Описание предметной области 5](#_Toc131697159)

[1.2. Описание существующей разработки 5](#_Toc131697160)

[2. ПРОЕКТНАЯ ЧАСТЬ 8](#_Toc131697161)

[2.1. Обоснование инструментов разработки 8](#_Toc131697162)

[2.2. Проектирование сценария работы проекта 11](#_Toc131697163)

[2.3. Разработка диаграммы классов 12](#_Toc131697164)

[2.4. Разработка ER-диаграммы баз данных 16](#_Toc131697165)

[2.5. Описание модулей проекта 18](#_Toc131697166)

[2.6. Тестирование проекта 19](#_Toc131697167)

[2.7. Руководство оператора 23](#_Toc131697168)

[ЗАКЛЮЧЕНИЕ 30](#_Toc131697169)

[Список используемых источников 31](#_Toc131697170)

# **ВВЕДЕНИЕ**

Тема курсового проекта – «Разработка системы управления цветочным магазином «Ромашка»».

Актуальность данного курсового проекта заключается в том, что он позволяет сократить время, используемое на учет цветочной продукции и запись заказов клиентов.

Целью данного курсового проекта является разработка клиентской части десктопного приложения, которое будет взаимодействовать с базой данных.

В первой части будет рассмотрена предметная область и существующие продукты по данной теме.

Во второй части будут описаны инструменты и модули, которые были разработаны и описано руководство оператора.

В заключительной части будет и заключение по курсовому проекту.

# **ТЕОРЕТИЧЕСКАЯ ЧАСТЬ**

## **Описание предметной области**

Целью данной работы является разработка десктопного приложения для электронного учета цветочной продукции, которая имеется на складе, а также ведения заказов клиентов. В данной ситуации предлагается следующее решение поставленной задачи:

Десктопное приложение в функционал которого входит:

* Возможность записи заказов клиентов;
* Управление базой данных через визуальный интерфейс;
* Фильтр, определяющий нужное количество закупаемых цветов, для выполнения заказов;
* Составление продаваемых цветочных букетов;
* Подсчет прибыли.

## **Описание существующей разработки**

В этом разделе рассмотрены некоторые уже имеющиеся программы по данной тематике.

Практически все из них предоставляют похожий функционал:

* Управление базой данных;
* Запись заказов клиентов;
* База данных цветов.

Примеры программ:

* Florapoint

Florapoint – программный комплекс учета продаж в цветочном бизнесе: CRM, склад, аналитика. Визуальный интерфейс показан на рисунке 1.

Преимущества:

* Удобство в использовании;
* Есть пробная бесплатная версия.

Недостатки:

* Полный функционал программы открывается при помощи платной ежемесячной подписке;
* Платные интеграции;
* Низкая оптимизации скорость работы программы.

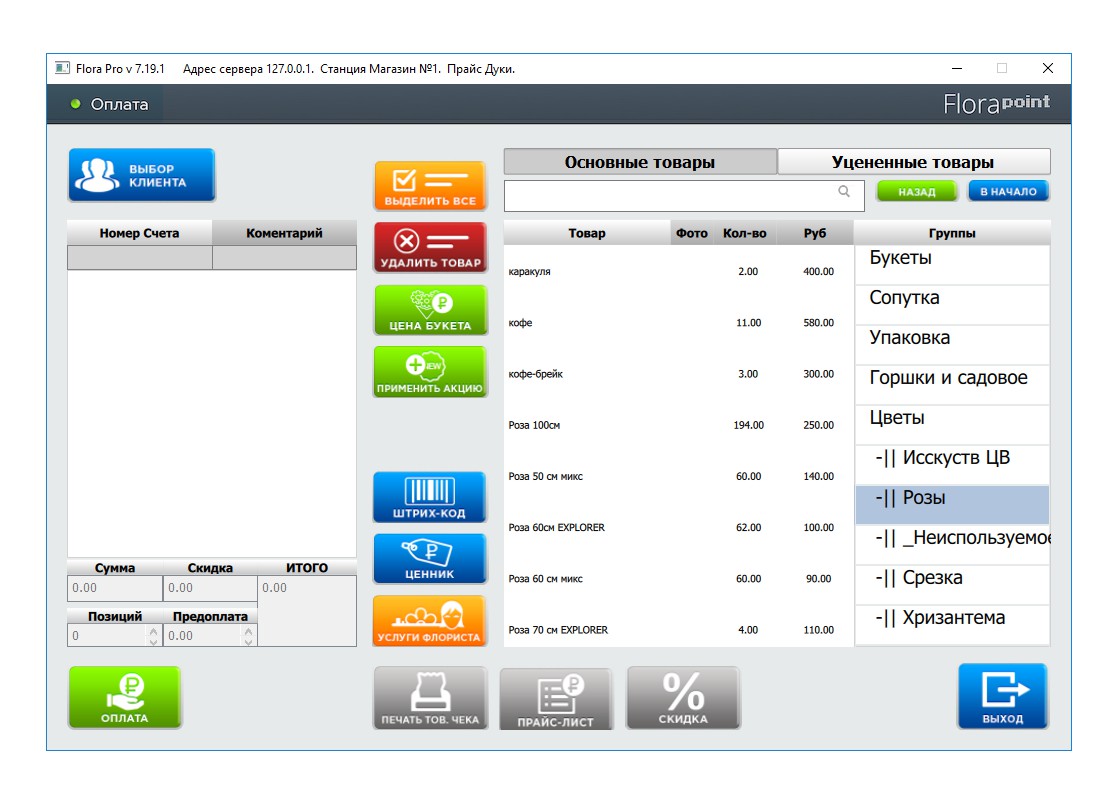


Рисунок 1. Интерфейс программы «Florapoint»

* Posiflora

Posiflora – программа позволяет автоматизировать работу цветочных салонов, интернет‑магазинов, цветочных супермаркетов, маленьких цветочных магазинов и сетей. Вы сможете управлять клиентской базой и системой лояльности, вести складской учёт, контролировать доставку, снизить расходы и повысить продажи. Визуальный интерфейс показан на рисунке 2.

Преимущества:

* Удобство в использовании;
* Есть стандартная бесплатная версия.

Недостатки:

* Полный функционал программы открывается при помощи платной ежемесячной подписке;
* В определенные дни у программы может быть сбой.

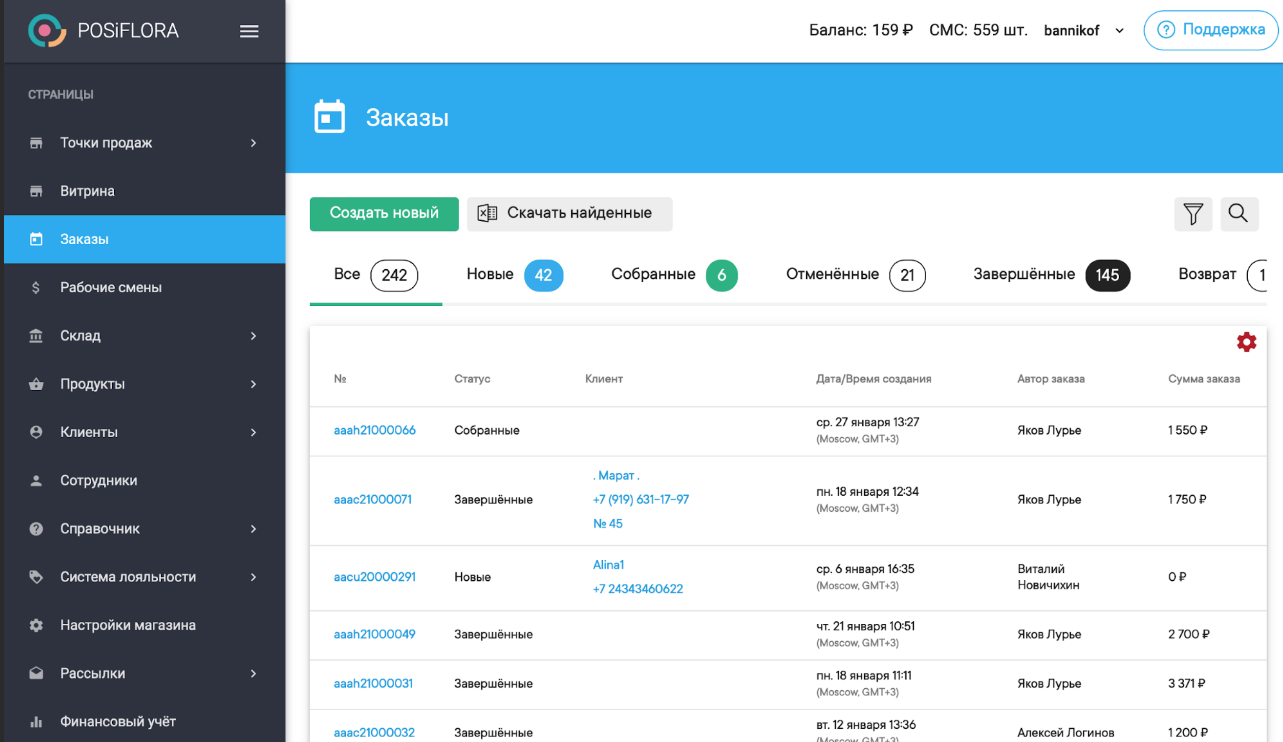


Рисунок 2. Интерфейс программы «Posiflora» 1

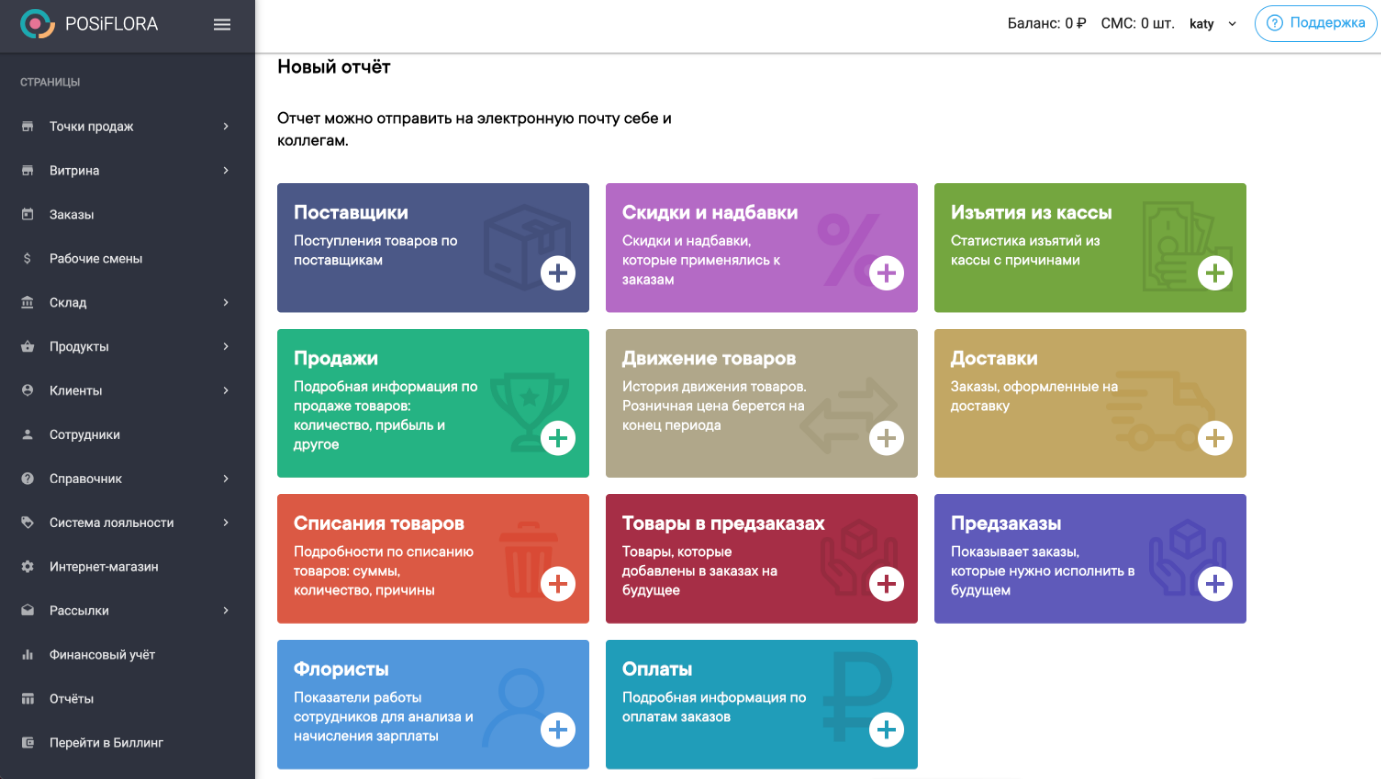


Рисунок 3. Интерфейс программы «Posiflora» 2

# **ПРОЕКТНАЯ ЧАСТЬ**

## **Обоснование инструментов разработки**

Для создания проекта, основным инструментом был выбран PyCharm: the Python IDE, потому что он является одним из лучших инструментов для разработки на Python. В качестве СУБД был выбран SQLite. В качестве модуля для создания GUI приложений в Python был выбран PyQt5, который является одним из наиболее популярных.

Язык Python был выбран по следующим факторам:

* Понятность кода – простое зрительное восприятие программ, написанных на этом языке.
* Наличие мощного фреймворка для разработки клиентских приложений;
* Наличие подробной документации.

Сравнение Python с другими языками программирования представлено в таблице 1.

Таблица 1. сравнения языков программирования

|  |  |  |  |
| --- | --- | --- | --- |
| **Критерий** | **C++** | **С#** | **Python** |
| Читабельность | 5 | 7 | 8 |
| Простота | 4 | 6 | 8 |
| GUI | 9 | 8 | 7 |
| Скорость разработки | 5 | 6 | 8 |
| Знание языка | 5 | 3 | 8 |
| Итого баллов | 28 | 30 | 39 |

Причины, по которым был выбран данный инструмент и СУБД:

**PyCharm**

PyCharm – это интегрированная среда разработки (IDE) для языка программирования Python, разработанная компанией JetBrains. Она предоставляет широкий набор инструментов для разработки, отладки, тестирования и управления проектами на Python. PyCharm поддерживает множество популярных фреймворков, таких как Django, Flask, PyQt5 и другие.

Включает в себя:

* Редактор кода;
* Инструменты рефакторинга кода;
* Интеграцию с системой контроля версий Git.

PyCharm делает разработку максимально продуктивной благодаря функциям автодополнения и анализа кода, мгновенной подсветке ошибок и быстрым исправлениям. Автоматические рефакторинги помогают эффективно редактировать код, а удобная навигация позволяет мгновенно перемещаться по проекту.

Возможности форматирования и настройки стиля кода помогают писать понятный код, который легко поддерживать.

**SQLite**

SQLite - это легковесная реляционная система управления базами данных, которая не требует отдельного сервера и может быть использована в качестве встроенной базы данных в приложениях. SQLite может быть использована для хранения небольших объемов данных, таких как настройки приложения, и для создания более сложных приложений, таких как браузеры и мобильные приложения.

**Преимущества SQLite**

1. Легковесность: SQLite является очень легкой базой данных, которая может быть использована в качестве встроенной базы данных в приложениях.
2. Нет необходимости в отдельном сервере: SQLite не требует отдельного сервера для работы, что делает его очень удобным для использования в небольших приложениях.
3. Транзакционность: SQLite поддерживает транзакции, что позволяет обеспечить целостность данных и защиту от ошибок.
4. Индексирование: SQLite поддерживает индексирование, что позволяет быстро находить нужные данные.
5. Ограничения целостности: SQLite поддерживает ограничения целостности, что позволяет обеспечить правильное хранение данных и защиту от ошибок.
6. Бесплатность: SQLite является бесплатной базой данных, что делает его доступным для использования в любых приложениях.

**PyQt5**

PyQt 5 - это библиотека для создания графических интерфейсов пользователя (GUI) на языке Python. Она предоставляет доступ к богатому набору компонентов GUI, таких как кнопки, поля ввода, меню и т.д., а также к мощным инструментам для создания пользовательских интерфейсов.

Основные возможности PyQt 5:

1. Создание графических интерфейсов пользователя любой сложности.
2. Поддержка различных виджетов и элементов управления.
3. Возможность создания многопоточных приложений.
4. Поддержка работы с базами данных.
5. Возможность создания кроссплатформенных приложений, которые работают на различных операционных системах.

В целом, PyQt 5 предоставляет мощный и гибкий инструментарий для создания пользовательских интерфейсов на Python и позволяет разработчикам создавать высококачественные приложения с минимальными затратами времени и усилий.

Одним из основных преимуществ PyQt 5 является его кроссплатформенность. Это означает, что приложения, созданные с помощью PyQt 5, могут работать на различных операционных системах, таких как Windows, macOS и Linux. Также PyQt 5 предоставляет широкий набор виджетов и элементов управления, что позволяет создавать пользовательские интерфейсы любой сложности.

Кроме того, PyQt 5 имеет мощные инструменты для работы с базами данных и интеграцию с другими библиотеками Python, такими как NumPy и Pandas. Это позволяет разработчикам создавать приложения с расширенными возможностями обработки данных и визуализации.

Наконец, PyQt 5 обладает гибкой архитектурой и возможностью создания многопоточных приложений, что позволяет эффективно использовать ресурсы компьютера и улучшить производительность приложения.

## **Проектирование сценария работы проекта**

Сценарий работы проекта - это документ, который описывает последовательность действий, необходимых для выполнения проекта. Он включает в себя информацию о целях проекта, ресурсах, необходимых для его выполнения, временных рамках, задачах и ответственных за их выполнение, рисках и способах их управления.

Сценарий работы проекта помогает команде проекта понимать, какие шаги должны быть предприняты для успешного завершения проекта и какие риски могут возникнуть по ходу выполнения проекта.

В данном разделе приведен сценарий использования программы пользователем приложения (Рисунок 4).

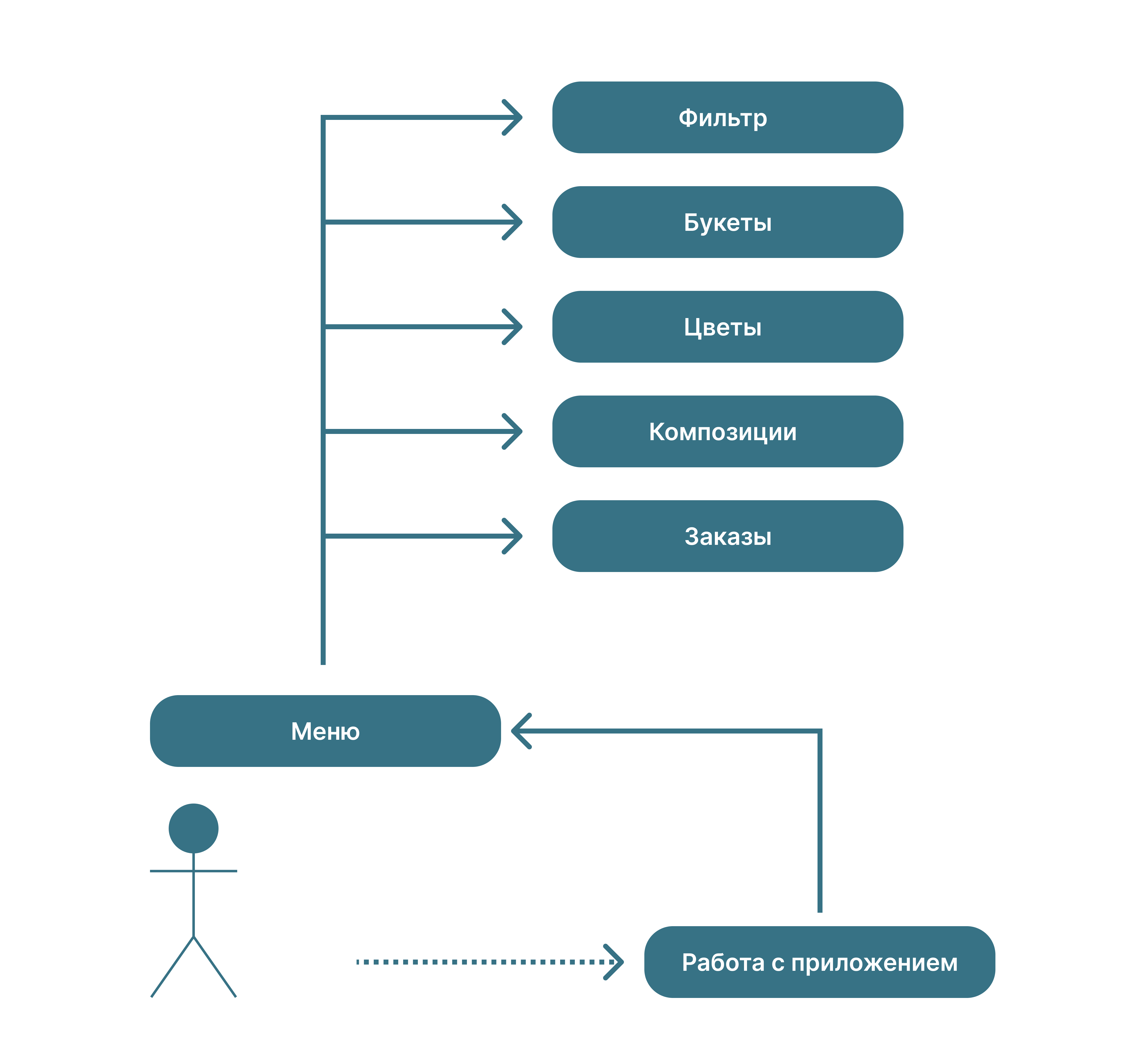


Рисунок 4. Сценарий использования

Пользователь после запуска приложения может выполнить следующие действия:

1. Управление заказами клиентов, добавление новых;
2. Управление букетами;
3. Добавление/удаление/редактирование цветов;
4. Составление цветочных композиций;
5. Фильтрация по определенным цветам и периоду.

При выходе из программы, она завершает свою работу.

## **Разработка диаграммы классов**

Диаграмма классов программы - это графическое представление классов и связей между ними в программе. Она позволяет лучше понять структуру программы и ее компоненты.

Для создания диаграммы классов необходимо определить классы, которые будут участвовать в программе, и связи между ними. Классы обычно представляют собой объекты, имеющие свои свойства и методы. Связи между классами могут быть различными: наследование, ассоциация, агрегация и композиция.

Для создания диаграммы классов можно использовать специальные инструменты для моделирования, такие как UML-редакторы. В этих инструментах уже есть готовые элементы для создания классов и связей между ними, что значительно упрощает процесс создания диаграммы.

После создания диаграммы классов необходимо проработать ее детали и убедиться, что все классы и связи заданы правильно и полностью отражают структуру программы. Диаграмма классов может использоваться в качестве руководства при разработке программного кода, а также для общения с другими разработчиками и заказчиками.

Создание диаграмм классов имеет ряд преимуществ:

1. Улучшение понимания структуры программы: диаграмма классов помогает лучше понять, какие классы используются в программе и как они связаны между собой;
2. Упрощение процесса разработки: диаграмма классов может использоваться в качестве руководства при написании кода, что упрощает процесс разработки;
3. Уменьшение количества ошибок: диаграмма классов позволяет выявить возможные ошибки в структуре программы на ранней стадии разработки, что уменьшает количество ошибок в итоговом продукте;
4. Ускорение коммуникации: диаграмма классов может использоваться для общения между разработчиками и заказчиками, что ускоряет процесс разработки и уменьшает количество недопониманий;
5. Улучшение переиспользуемости кода: диаграмма классов позволяет лучше понимать, какие классы могут быть переиспользованы в других проектах, что улучшает переиспользуемость кода;

Ниже представлены все классы, использующиеся в проекте, а также их отношения между собой (Рисунок 5).

1. Main: входная точка;
2. OrderSystem: класс содержит набор функций, импортируемых модулей, реализующих логику приложения, взаимодействие с БД;
   1. Load\_data\_composition - загрузка данных с таблицы composition из БД sql Db.db. Срабатывает при нажатии на кнопку «Загрузить» на вкладке «Композиции»;
   2. Load\_data\_flowers\_flavors - загрузка данных с таблицы flowers и flavors из БД sql Db.db. Срабатывает при нажатии на кнопку «Загрузить» на вкладке «Цветы» и «Букеты»;
   3. Add\_new\_row – добавление новой строки. Срабатывает при нажатии на кнопку «Добавить»;
   4. Delete\_row – удаление выбранной строки. Срабатывает при нажатии на кнопку «Удалить»;
   5. Save\_data\_flowers\_flavors – сохранение данных из таблицы flowers или flavors в таблицу sql. Срабатывает при нажатии на кнопку «Сохранить» на вкладке «Цветы» или «Букеты»;
   6. Load\_data\_order – загрузка данных с таблицы orders из БД sql Db.db. Срабатывает при нажатии на кнопку «Загрузить» на вкладке «Заказы»;
   7. List\_flowers – формирование всего списка цветов для добавления в comboBox на вкладке «Фильтр».
3. DateEdit: класс переопределяет родительский класс QDateEdit, для назначения кастомных стилей;
   1. Date\_edit\_style – переменная, хранящая в себе стили CSS для виджета QDateEdit.
4. ComboPickFlavor: класс переопределяет родительский класс QComboBox, для назначения кастомных стилей;
   1. Combo\_style – переменная, хранящая в себе стили CSS для виджета QComboBox.
5. Singleton: класс создает паттерн singleton, не позволяя создавать более 1 экземпляра приложения;
   1. Getinstance – функция-обертка, реализующая паттерн singleton.
6. UiMainWindow: класс инициализирует виджеты и GUI приложения, производит их первоначальную настройку.
   1. Setup\_ui – функция, создающая необходимые виджеты и стили CSS к ним.
7. Database: класс инициализирует базу данных, если она была не создана, наполняет ее шаблонными данными, содержит набор функцию для взаимодействия с БД.
   1. Db – путь к файлу БД;
   2. Connection – представляет соединение с БД через обращение к переменной «Db»;
   3. Tbl\_data – составляет и выполняет запрос на получение данных из таблицы, получаемой в аргументах;
   4. Save\_data – составляет и выполняет запрос на сохранение данных таблицы, получаемой в аргументах;
   5. Last\_id – возвращает номер последнего ID из таблицы, получаемой в аргументах;
   6. List\_flowers – составляет и выполняет запрос на взятие суммарного количество цветов по взятым параметрам;
   7. All\_money – получение итоговой прибыли;
   8. Get\_flowers – полученгие всего списка цветов.

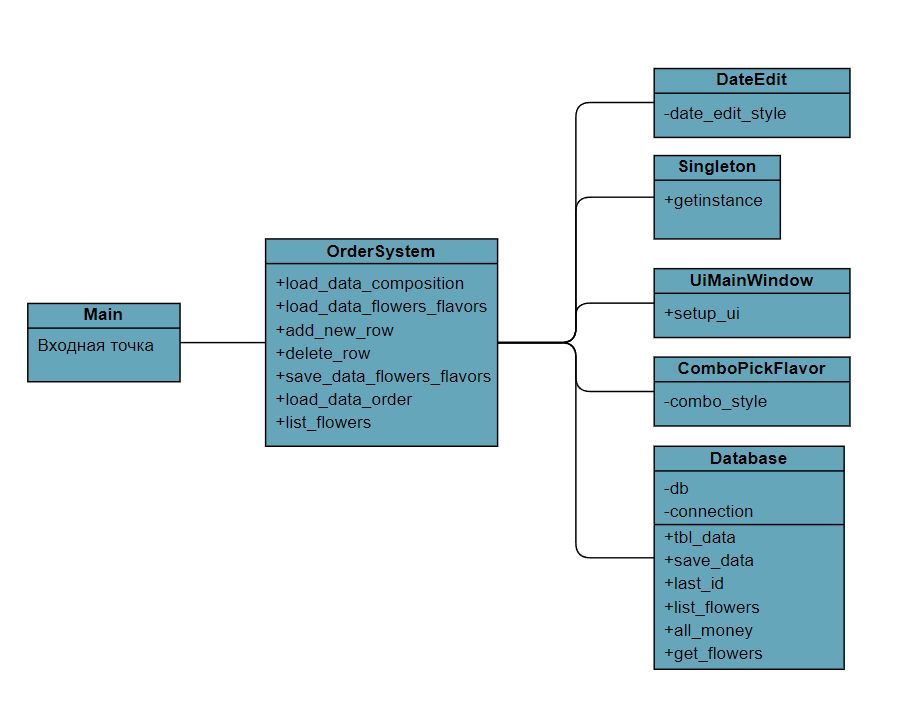


Рисунок 5. Диаграмма классов

## **Разработка ER-диаграммы баз данных**

ER-диаграммы баз данных - это графическое представление сущностей и связей между ними в базе данных. ER-диаграммы помогают описать структуру и отношения между объектами в базе данных. ER-диаграмма состоит из сущностей, атрибутов и связей. Сущность представляет объект в базе данных, атрибуты - характеристики этого объекта, а связи - отношения между объектами. ER-диаграммы являются важным инструментом для проектирования баз данных, позволяя описать их структуру и связи между объектами.

Разработка ER-диаграммы баз данных происходит в несколько этапов:

1. Определение сущностей - определяются объекты, которые будут храниться в базе данных. Например, для базы данных интернет-магазина это могут быть клиенты, товары, заказы и т.д;
2. Определение атрибутов - для каждой сущности определяются её характеристики. Например, для сущности "клиент" это могут быть имя, фамилия, адрес и т.д;
3. Определение связей - определяются отношения между сущностями. Например, один клиент может сделать несколько заказов, а один заказ может содержать несколько товаров;
4. Создание ER-диаграммы - на основе полученных данных создаётся графическое представление базы данных с помощью специальных символов и обозначений;
5. Тестирование и доработка - после создания ER-диаграммы она тестируется на соответствие требованиям и при необходимости дорабатывается;
6. Реализация базы данных - после разработки и утверждения ER-диаграммы начинается реализация базы данных с помощью СУБД.

В приложении для хранения всей нужной информации была создана база данных. На рисунке 6 представлена ER-модель Базы данных. Так же ниже описана структура таблиц и назначение полей.

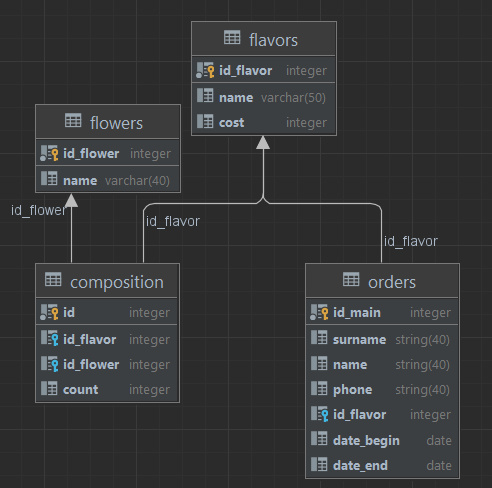


Рисунок 6. ER-модель

База данных состоит из 4 таблиц:

1. Flavors – таблица для хранения наименований букетов и их цены;
   1. Id\_flavor – первичный ключ;
   2. Name – наименование цветочного букета;
   3. Cost – цена букета.
2. Flowers – таблица для хранения наименований цветов;
   1. Id\_flower – первичный ключ;
   2. Name – наименование цветка.
3. Composition – таблица цветочных композиций;
   1. Id – первичный ключ;
   2. Id\_flavor – внешний ключ на таблицу «Flavors»;
   3. Id\_flower – внешний ключ на таблицу «Flowers»;
   4. Count – количество цветка.
4. Orders – таблица заказов клиентов.
   1. Id\_main – первичный ключ;
   2. Surname – фамилия клиента;
   3. Name – имя клиента;
   4. Phone – номер телефона книента;
   5. Id\_flavor – внешний ключ на таблицу «Flavors»;
   6. Date\_begin – дата оформления заказа;
   7. Date\_end – дата выдачи заказа.

## **Описание модулей проекта**

Помимо стартового модуля, в проект входят и другие модули с фрагментами, и интерфейсные модули.

Также в стартовый модуль входит код, который запускает саму программу при ее запуске. Полный код всех модулей находится в «Приложение 1».

async\_init() – данный метод запустится при старте программы, как входная точка в асинхронность;

load\_data() – этот метод вызывается для загрузки данных из БД в определенную таблицу;

add\_new\_row() – данный метод вызывается для добавления новой строки в определенную таблицу;

delete\_row() – данный метод вызывается для удаление строки в определенной таблице;

save\_data() – данный метод берёт сохраняет данные редактируемой таблицы.

## **Тестирование проекта**

В этом разделе будут продемонстрированы результаты тестирования программы.

**Тест. Запуск приложения**

Действия: запустить приложение (Рисунок 7).

Ожидаемый результат: отображение окна «Заказы».

Результат теста:



Рисунок 7. Главное меню «Заказы»

В этом разделе показано умение применять средства отладки. Так как я работаю в PyChram, отладчик уже встроен в программу.

Для проверки была искусственно создана ошибка в коде. Вот что показал отладчик (Рисунок 8,9).

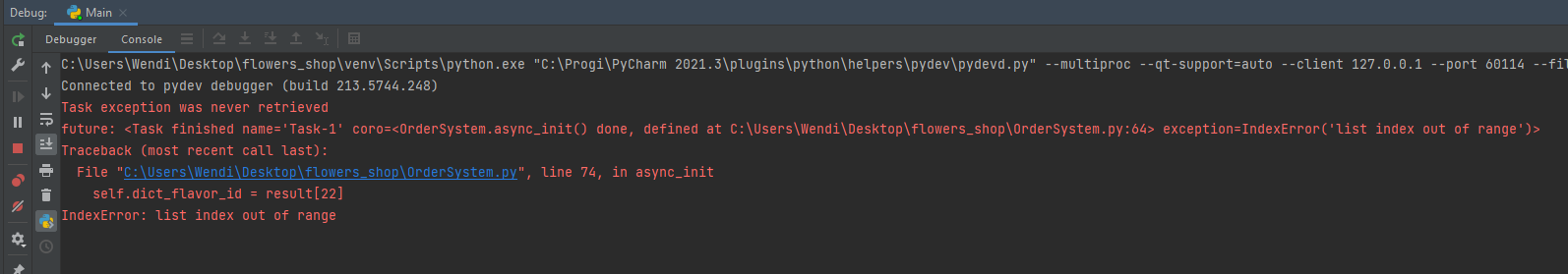


Рисунок 8. Ошибка в коде

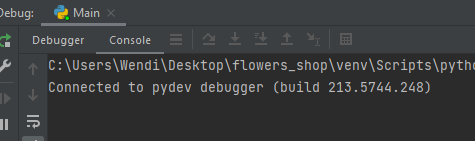


Рисунок 9. Исправление ошибки

**Тест. Срабатывание кнопки «Добавить»**

Действия: добавление новой строки в таблице, посредством кнопки «Добавить».

Ожидаемый результат: отображение новой пустой строки в таблице.

Результат теста:

Программа была запущена с помощью встроенного отладчика PyCharm – Debugger, чтобы отследить возможные ошибки. На вкладке «Заказы» была нажата кнопка «Добавить» (Рисунок 10).



Рисунок 10. Тест добавления новой строки

После этого была добавлена новая строка в таблицу. Следовательно, функция add\_new\_row\_orders сработала корректно (Рисунок 11).



Рисунок 11. Тест с добавленной строкой

В консоли отладчика не было выявлено никаких ошибок (Рисунок 12).

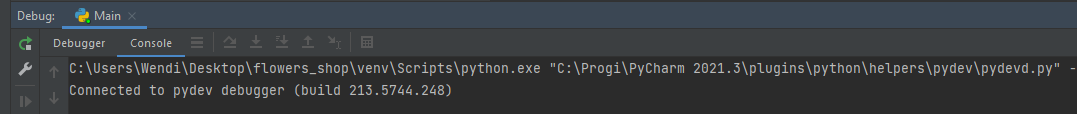


Рисунок 12. Добавление строки - консоль

**Тест. Отображение таблиц на вкладке «Композиции» из списка**

Действия: открыть список таблик на вкладке «Композиции», выбрать букет «Салют».

Ожидаемый результат: отображение выбранной таблицы с загруженными данными.

Результат теста:

Программа была запущена с помощью встроенного отладчика PyCharm – Debugger, чтобы отследить возможные ошибки. Далее, на вкладке «Композиции» был открыт список букетов (Рисунок 13).



Рисунок 13. Выбор таблицы на вкладке «Композиции»

После этого был выбран букет «Салют». Программа загрузила корректные данные в таблицу (Рисунок 14).



Рисунок 14. Загрузка таблицы букета «Салют»

В консоли отладчика не было выявлено никаких ошибок (Рисунок 15).

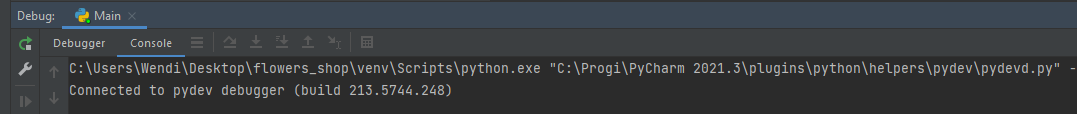


Рисунок 15. Добавление строки – консоль

## **Руководство оператора**

В данном программном документе приведено руководство оператора по применению и эксплуатации программы «Chamomile.exe».

В данном программном документе, в разделе «Назначение программы» указаны сведения о назначении программы и информация, достаточная для понимания функций программы и ее эксплуатации.

В разделе «Условия выполнения программы» указаны условия, необходимые для выполнения программы (минимальный состав аппаратных и программных средств и т.п.).

В данном программном документе, в разделе «Выполнение программы» указана последовательность действий оператора, обеспечивающих загрузку, запуск, выполнение и завершение программы, приведено описание функций, формата и возможных вариантов команд, с помощью которых оператор осуществляет загрузку и управляет выполнением программы, а также ответы программы на эти команды.

Оформление программного документа «Руководство оператора» произведено по требованиям ЕСПД (ГОСТ 19.101-77 [[1]](#footnote-1)1), ГОСТ 19.103-77 [[2]](#footnote-2)2), ГОСТ 19.104-78\* [[3]](#footnote-3)3), ГОСТ 19.105-78\* [[4]](#footnote-4)4), ГОСТ 19.106-78\* [[5]](#footnote-5)5), ГОСТ 19.505-79\* [[6]](#footnote-6)6), ГОСТ 19.604-78\* [[7]](#footnote-7)7)).

Далее приведено содержание разработанного документа «Руководство оператора» на спроектированное приложение.

1. **Назначение программы**
   1. **Функциональное назначение программы**

Специальное программное обеспечение «Сhamomile.exe» используется для управления цветочным магазином «Ромашка».

* 1. **Эксплуатационное назначение программы**

Специальное программное обеспечение «Сhamomile.exe» может эксплуатироваться в учебных заведениях.

* 1. **Состав функций**
     1. **Функция загрузки данных из БД.**

Эта функция позволяет загрузить данные с базы данных в выбранную таблицу.

* + 1. **Функция сохранения данных в БД.**

Эта функция позволяет пользователю сохранить данные в таблице после изменения данных в ней.

* + 1. **Функция добавления новой строки в таблицу.**

Эта функция позволяет добавить новую строку в выбранную таблицу.

* + 1. **Функция удаления строки в таблице.**

Эта функция позволяет удалить строку из выбранной таблице.

* + 1. **Функция фильтрации по выбранному цветку.**

Эта функция позволяет отфильтровать по выбранному цветку и увидеть в каких букетах он используется и в каком количестве.

1. **Условия выполнения программы**
   1. **Минимальный состав аппаратных средств**

* ОС: Windows 10 и выше;
* Процессор: 32-битные и 64-битные: ARM, MIPS, x86. Тактовая частота не ниже 1,8 ГГц;
* ОЗУ: 2 ГБ;
* Место на жестком диске: 10 МБ;
* Видеоадаптер: DirectX версии не ниже 9 с драйвером WDDM 1.0.;
* Дисплей: 720 x 1600.
  1. **Минимальный состав программных средств**

Дополнительные программные средства не требуются.

* 1. **Требование к пользователю**

Конечный пользователь программы должен обладать практическими навыками работы с графическим пользовательским интерфейсом операционной системы.

1. **Выполнение программы**
   1. **Загрузка и запуск программы**

Запустите программу «Сhamomile.exe», после чего откроется окно пролога (Рисунок 16).



Рисунок 16. Окно приложения при запуске

* 1. **Выполнение программы**
     1. **Выполнение функции добавления и сохранения нового заказа**

Выберите кнопку «Добавить», после этого заполните все поля, кроме «ID» и «Цена букета» (после сохранения данные автоматически заполняться в этих ячейках) (Рисунок 17).



Рисунок 17. Окно «Заказы»

После того как мы заполним данные и нажмём кнопку «Сохранить», при корректном заполнении всех полей, данные будут сохранены в БД, нижняя строка статуса проинформирует об успешном сохранении данных (Рисунок 18).

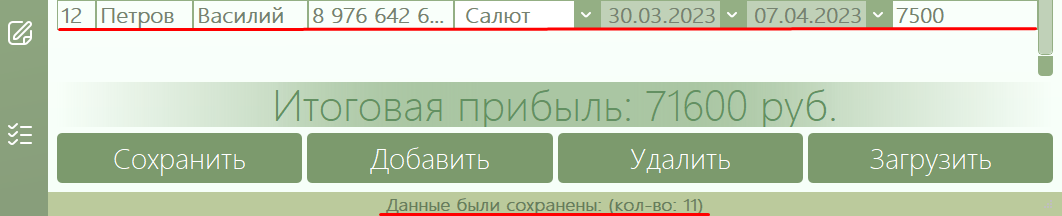


Рисунок 18. Сохранение данных

* + 1. **Выполнение функции удаления**

Перейдите на вкладку с нужной вам таблицей, после этого выберите строку для удаления (Рисунок 19).



Рисунок 19. Вкладка «Композиция»

После того, как вы выбрали нужную строку, нажмите на кнопку «Удалить». Строка будет удалена из визуальной таблицы и БД. Статусная строка в нижней части приложения проинформирует об успешном удалении (Рисунок 20).



Рисунок 20. Удаление строки

* + 1. **Выполнение функции загрузки данных**

Перейдите на вкладку с нужной вам таблицей, после этого измените любые ячейки (Рисунок 21, 22).

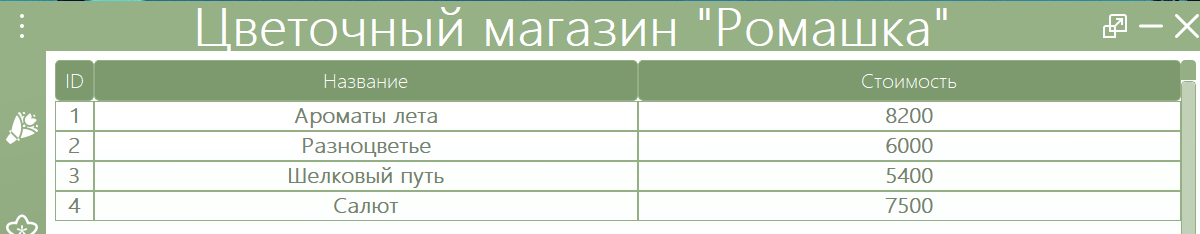


Рисунок 21. Таблица букетов до изменений

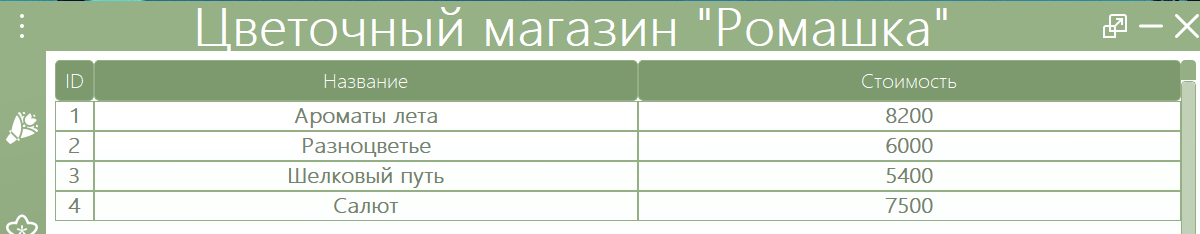


Рисунок 22. Таблица букетов после изменений

После изменения данных в таблице, без сохранения данных нажмите кнопку «Загрузить». Будут загружены данные, до несохраненных изменений, что видно на рисунке 23.

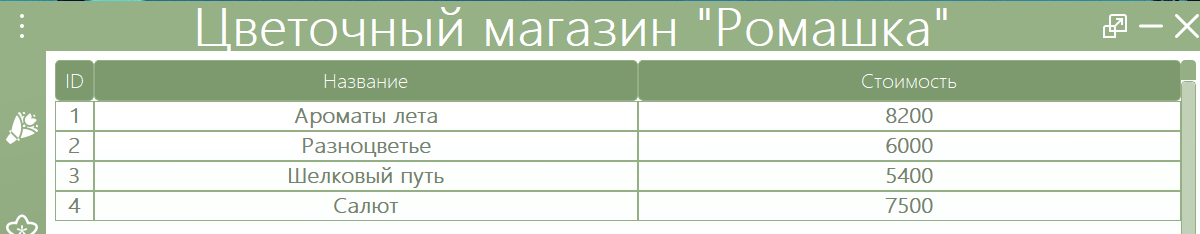


Рисунок 23. Загрузка старых данных без сохранения

* + 1. **Выполнение функции фильтрации**

Перейдите на вкладку «Фильтр». Далее выберите цветок и период, за который должна производиться фильтрация (Рисунок 24).

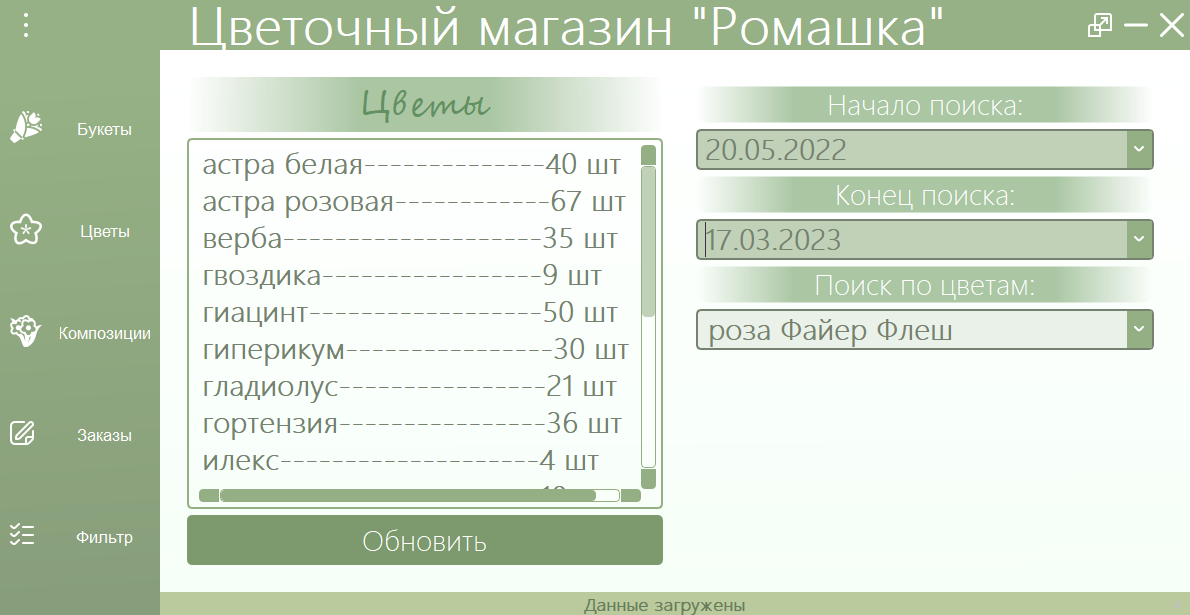


Рисунок 24. Вкладка «Фильтр»

После этого нажмите кнопку «Обновить». В списке будут показаны букеты, в которых используется выбранный цветок и количество цветков, необходимое для дальнейшей закупки, чтобы выполнить все действующие заказы из таблицы на вкладе «Заказы» (Рисунок 25).

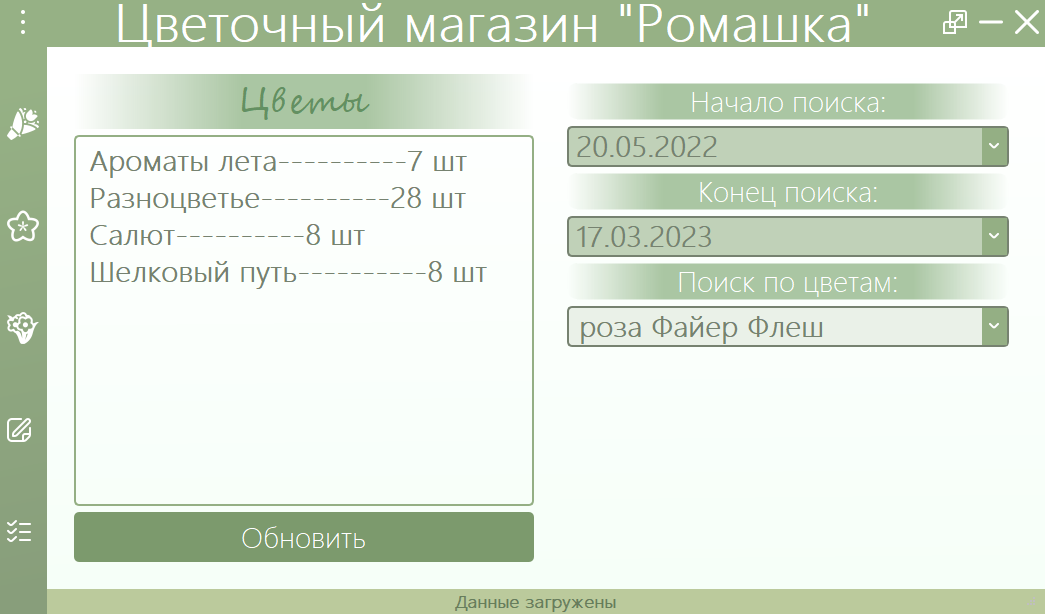


Рисунок 25. Отфильтрованные данные

* 1. **Завершение работы программы**

Чтобы завершить работу программы необходимо нажать на «крестик» в правой верхней части экрана.

# **ЗАКЛЮЧЕНИЕ**

В результате выполнения курсового проекта была написана программа «chamomile.exe» для упрощения и удобства учета цветочной продукции на складе и оформления заказов клиентов в цветочном магазине «Ромашка».

В ходе работы были проанализированы: предметная область, существующие разработки, получены навыки по созданию десктопных приложений и работе с БД.

Также планируется продолжение работы над данным проектом с целью расширения возможностей и удобства приложения для пользователей.

# **Список используемых источников**

1. Бейдер Д. Чистый Python. Тонкости программирования для профи / пер. с англ. - М.: ООО Издательство «Питер», 2022. – 288 с.
2. Доусон М. Программируем на Python / пер. с англ. - М.: ООО Издательство «Питер», 2022. – 416 с.: ил.
3. Лутц, М. Изучаем Python, 5-е издание / пер. с англ. СПБ.: ООО “Диалектика”, 2019. — 832 с.: ил. — Парал, тит. англ.
4. Любанович Б. Простой Python. Современный стиль программирования / пер. с англ. - М.: ООО Издательство «Питер», 2021. – 592 с.
5. Маккинни У. Python и анализ данных: Первичная обработка данных с применением pandas, NumPy и Jupiter / пер. с англ. А. А. Слинкина. 3-е изд. – М.: МК Пресс, 2023. – 536 с.
6. Свейгарт Э. Автоматизация рутинных задач с помощью Python: практическое руководство для начинающих. 2-е изд / пер. с англ. - М.: ООО «И.Д. Вильямс», 2021. – 592 с.: ил. — Парал, тит. англ.
7. Интернет-ресурс «Pythonist.ru» [Электронный ресурс] URL: <https://pythonist-ru.turbopages.org/pythonist.ru/s/rukovodstvo-po-pyqt5/> (дата обращения: 02.02.2023).
8. Интернет-ресурс «WebforMyself» [Электронный ресурс] URL: <https://webformyself.com/rukovodstvo-po-pyqt-python-gui-designer/> (дата обращения: 15.01.2023).

**Приложение 1.**

Находится в файле «Приложение 1.docx».

1. 1) ГОСТ 19.101-77 ЕСПД. Виды программ и программных документов [↑](#footnote-ref-1)
2. 2) ГОСТ 19.103-77 ЕСПД. Обозначение программ и программных документов [↑](#footnote-ref-2)
3. 3) ГОСТ 19.104-78\* ЕСПД. Основные надписи [↑](#footnote-ref-3)
4. 4) ГОСТ 19.105-78\* ЕСПД. Общие требования к программным документам [↑](#footnote-ref-4)
5. 5) ГОСТ 19.106-78\* ЕСПД. Общие требования к программным документам, выполненным печатным способом [↑](#footnote-ref-5)
6. 6) ГОСТ 19.505-79\* ЕСПД. Руководство оператора. Требования к содержанию и оформлению [↑](#footnote-ref-6)
7. 7) ГОСТ 19.604-78\* ЕСПД. Правила внесения изменений в программные документы, выполненные печатным способом [↑](#footnote-ref-7)