**Федеральное государственное автономное образовательное**

**учреждение высшего образования**

**«ОМСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**

**Кафедра математических методов и  
информационных технологий в экономике**

|  |  |
| --- | --- |
|  | **ИТОГОВЫЙ ПРОЕКТ** |
| по дисциплине | Методы и средства проектирования информационных систем и технологий в медиаиндустрии |
|  | **Пояснительная записка** |

Выполнила ст. гр. \_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_С. Ю. Кузина

"\_\_\_\_"\_\_\_\_\_\_\_\_\_\_\_20\_\_г.

Принял

\_\_\_\_\_\_\_\_\_О.Н. Канева

"\_\_\_\_"\_\_\_\_\_\_\_\_\_\_\_20\_\_ г.

**Омск 2024**

ВВЕДЕНИЕ

Цель разработки данного приложения — создание простой и удобной CRM-системы для управления клиентскими данными. Приложение ориентировано на малые и средние компании, которым необходим инструмент для хранения, редактирования и анализа информации о клиентах. В основе системы — минималистичный интерфейс и функциональность, позволяющие пользователю легко управлять клиентской базой.

Идея приложения заключается в предоставлении базового инструмента для работы с клиентами, который включает такие функции, как добавление, обновление и удаление данных, импорт и экспорт клиентской информации, а также поиск и фильтрацию. Это позволяет компаниям поддерживать актуальность данных и быстро находить необходимую информацию.

Актуальность разработки обусловлена растущей потребностью в цифровых инструментах для организации работы с клиентами. Малый бизнес часто сталкивается с отсутствием подходящих CRM-систем, которые бы соответствовали их потребностям: были достаточно простыми, не требовали значительных затрат и при этом оставались функциональными. Данное приложение закрывает эту нишу, предоставляя решение, которое сочетает в себе основные функции CRM-системы и простоту использования.

**1 Техническое задание**

**1.1 Основные и технические требования**

**1.1.1 Бизнес-требования**

1. Приложение должно предоставлять минимально необходимый функционал для работы с клиентской базой (добавление, удаление, редактирование, поиск и фильтрация клиентов).

2. Возможность работы без подключения к интернету.

3. Простой процесс импорта и экспорта данных в формате CSV для взаимодействия с другими системами.

4. Низкая стоимость внедрения и минимальные требования к обучению пользователей.

**1.1.2 Пользовательские требования**

1. Простой и интуитивно понятный интерфейс для работы с клиентами.

2. Удобный поиск и фильтрация клиентов по заданным критериям.

3. Возможность импорта данных из CSV без сложной настройки.

4. Экспорт клиентской базы без идентификаторов для удобного использования в других системах.

**1.1.3 Функциональные требования**

1. Добавление, обновление и удаление информации о клиентах.

2. Импорт данных из файлов CSV с обработкой возможных ошибок (например, пустых строк или дублирующихся записей).

3. Экспорт клиентской базы в формат CSV с возможностью настройки, какие данные экспортировать.

4. Функции фильтрации и поиска клиентов по имени, фамилии или другим полям.

5. Возможность отображения прогресса во время импорта данных через прогресс-бар.

6. Раздел "Об авторе" для отображения информации о разработчике.

**1.1.4 Нефункциональные требования**

1. Высокая производительность при работе с клиентской базой, содержащей до 10 000 записей.

2. Надежность и сохранность данных при сбоях приложения.

3. Минимальные задержки при выполнении операций с базой данных.

**1.1.5 Требования к интерфейсу**

1. Использование графического интерфейса на базе PyQt5.

2. Размещение основных кнопок управления (добавление, обновление, удаление, экспорт, импорт) в одном блоке для удобного доступа.

3. Четкая и простая структура, состоящая из таблицы клиентов и вспомогательных элементов управления.

**1.1.6 Требования к безопасности**

1. Ограничение прав записи данных в базу для предотвращения случайного удаления всей информации.

2. Проверка формата входных данных (например, формат CSV при импорте).

3. Возможность восстановления данных из резервной копии при сбое.

**1.1.7 Платформы**

1. Приложение должно работать на операционных системах:

- Windows (7 и выше),

- Linux (современные дистрибутивы с поддержкой Python 3.8+),

- macOS (10.13 и выше).

2. Поддержка запуска на устройствах с архитектурой x86 и x64.

**1.1.8 Требования к производительности**

1. Операции с базой данных не должны превышать 2 секунд при выполнении над 10 000 записями.

2. Приложение должно запускаться менее чем за 5 секунд на стандартном ПК с 4 ГБ оперативной памяти.

**1.1.9 Требования к совместимости**

1. Совместимость с Python 3.8+ и библиотеками PyQt5, SQLite3.

2. Поддержка импорта данных из стандартных CSV-файлов, созданных в Excel или Google Sheets.

3. Возможность экспортировать данные в формате, совместимом с другими CRM-системами или приложениями для работы с таблицами.

**1.2 Архитектура приложения**

Приложение основано на архитектурной модели Model-View-Controller (MVC), которая разделяет логику данных, представление и управление, упрощая поддержку и масштабирование системы.

Модуль данных (Model) отвечает за взаимодействие с базой данных SQLite. Он обеспечивает такие функции, как создание и инициализация базы данных, добавление, обновление, удаление записей, получение данных по запросам (например, для поиска или фильтрации), а также импорт и экспорт данных в формате CSV.

Модуль интерфейса (View) предназначен для отображения данных и взаимодействия с пользователем. Основные компоненты интерфейса включают таблицу для отображения списка клиентов, кнопки управления (добавление, удаление, экспорт, импорт), поля ввода для фильтрации и поиска, а также всплывающие окна для подтверждения действий, отображения ошибок или прогресса операций импорта.

Модуль управления (Controller) связывает пользовательский интерфейс и базу данных, обеспечивая обработку действий пользователя (например, нажатие кнопок). Он вызывает соответствующие методы модели для работы с данными и обновляет интерфейс после изменений.

Дополнительно в приложении реализованы функции импорта и экспорта данных в CSV, механизм отображения статуса операций с помощью окна прогресса и раздел "Об авторе", предоставляющий информацию о разработчике.

В основе базы данных SQLite находится одна таблица clients, содержащая поля: id (уникальный идентификатор клиента), first\_name (имя), last\_name (фамилия), email (электронная почта), phone (телефонный номер). Таблица создается с помощью следующего SQL-запроса:

CREATE TABLE IF NOT EXISTS clients (

id INTEGER PRIMARY KEY AUTOINCREMENT,

first\_name TEXT NOT NULL,

last\_name TEXT NOT NULL,

email TEXT,

phone TEXT,

);

Архитектура приложения предполагает четкое разделение между компонентами. Модуль данных (Model) хранит и управляет информацией в базе данных, предоставляя интерфейсы для выполнения операций. Модуль интерфейса (View) отвечает за отображение данных и обработку взаимодействий с пользователем. Модуль управления (Controller) связывает интерфейс и модель, обеспечивая выполнение пользовательских запросов и обновление данных в реальном времени.

Например, при добавлении нового клиента пользователь нажимает соответствующую кнопку. Controller вызывает метод модели для добавления записи в базу данных и затем обновляет таблицу интерфейса через View. При фильтрации клиентов текст, введенный пользователем в поле поиска, передается модели для выполнения фильтрации, после чего Controller обновляет интерфейс с отфильтрованными данными.

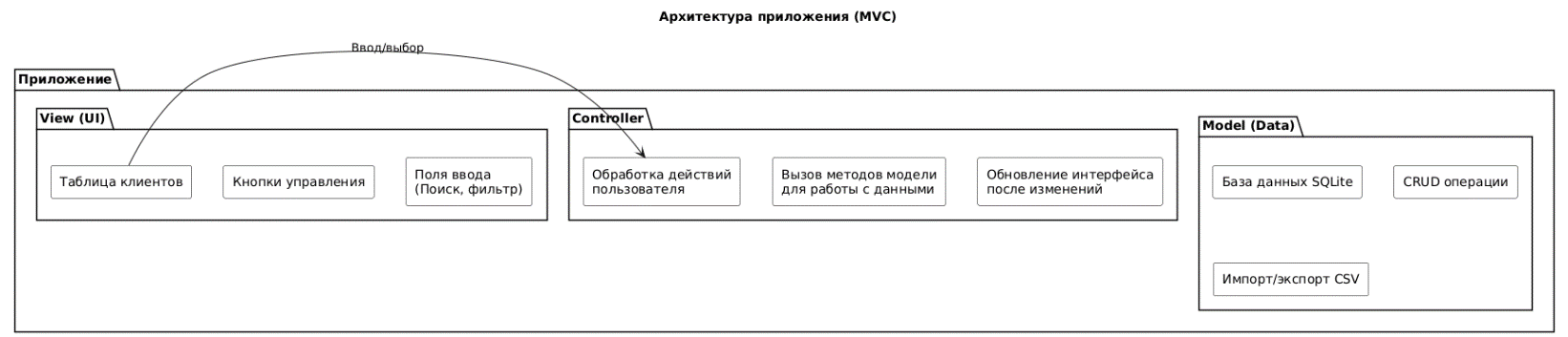
Архитектура MVC обеспечивает четкое разделение ответственности между модулями, что упрощает разработку, тестирование и дальнейшее масштабирование приложения.

Схема базы данных

База данных SQLite содержит одну таблицу для хранения информации о клиентах.

|  |  |  |
| --- | --- | --- |
| Поле | Тип | Описание |
| Id | Integer | Уникальный идентификатор (первичный ключ). |
| First\_name | Text | Имя клиента |
| Last\_name | Text | Фамилия клиента |
| Email | Text | Электронная почта клиента |
| Phone | Text | Телефонный номер клиента |

Схема приложения



**1.3 Дизайн и пользовательский интерфейс**

Приложение имеет простой и функциональный дизайн, ориентированный на удобство использования. Внешний вид напоминает классические настольные приложения, где акцент сделан на доступности функций и минимальном времени на освоение.

Главное окно состоит из нескольких ключевых элементов. Центральная часть отведена под таблицу клиентов, которая отображает все записи из базы данных. Каждая строка таблицы представляет отдельного клиента, а столбцы содержат такие атрибуты, как имя, фамилия, email, телефон и адрес. Под таблицей расположены кнопки управления: "Добавить клиента", "Удалить клиента", "Обновить клиента", "Экспортировать таблицу CSV", "Импортировать таблицу CSV" и "Об авторе". В верхней части интерфейса расположены поля поиска и фильтрации, которые позволяют быстро находить нужные записи.

Дополнительно приложение включает модальные окна для выполнения отдельных действий. Окно добавления или обновления клиента содержит поля ввода для имени, фамилии, email, телефона и адреса, а также кнопки "Сохранить" и "Отмена". Для отображения длительных операций, таких как импорт CSV, используется окно прогресса с индикатором выполнения и кнопкой для отмены операции. Всплывающие окна уведомляют пользователей о результатах операций, таких как успешное завершение или возникновение ошибки.

Дизайн приложения придерживается светлой цветовой схемы с белым или серым фоном, синими или зелеными акцентами на кнопках и выделениях, а также черным текстом для читаемости. Используется шрифт Sans Serif (например, Arial или Roboto) с размерами текста, адаптированными для удобного чтения: 10–12 pt для таблиц и 14 pt для заголовков. Для кнопок добавлены простые минималистичные иконки, такие как плюс для добавления клиента или корзина для удаления.

Интерфейс напоминает приложения, такие как Microsoft Access или Google Sheets, с упором на простоту работы с таблицами. Также он следует традициям настольных приложений на PyQt5, используя панели инструментов и таблицы, что напоминает интерфейсы старых версий Microsoft Office.

**2 Реализация приложения**

**2.1 Используемые библиотеки**

В приложении используются несколько ключевых библиотек, каждая из которых играет важную роль в его функциональности и интерфейсе. Основная библиотека для разработки — PyQt5, предоставляющая инструменты для создания графического интерфейса пользователя. PyQt5 была выбрана за её богатый набор виджетов, широкий функционал и возможность создания приложений, работающих на различных операционных системах. Она также поддерживает современные интерфейсные элементы и отлично интегрируется с Python.

Для работы с базой данных используется sqlite3 — стандартная библиотека Python для взаимодействия с базами данных SQLite. SQLite является легковесным и встроенным решением, не требующим установки дополнительных серверов, что упрощает развертывание приложения. Выбор этой библиотеки обусловлен её надежностью, простотой использования и достаточной производительностью для управления данными небольших и средних объемов, таких как клиентские записи [1].

Для операций с данными в формате CSV используется библиотека csv, также входящая в стандартную библиотеку Python. Она обеспечивает удобный интерфейс для чтения и записи CSV-файлов, что делает её идеальной для функций импорта и экспорта данных. Использование стандартной библиотеки устраняет необходимость в сторонних зависимостях, повышая стабильность и предсказуемость работы приложения.

Таким образом, выбор библиотек был обусловлен их функциональностью, надежностью и совместимостью с основными требованиями приложения. Использование PyQt5 обеспечивает современный и удобный пользовательский интерфейс [2,3], sqlite3 предоставляет мощный инструмент для работы с базой данных, а csv добавляют поддержку необходимых функций импорта, экспорта, улучшая общее качество и удобство работы приложения.

**2.2 Описание реализации**

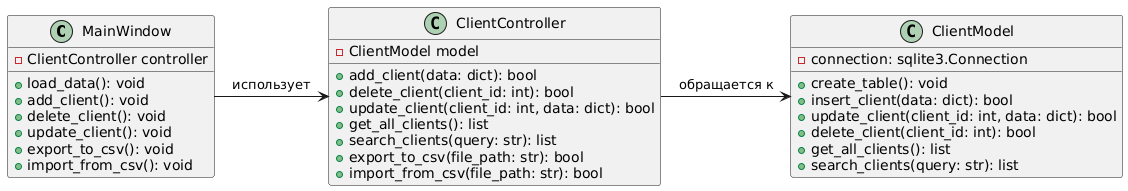
Приложение построено по архитектурной модели MVC (Model-View-Controller), что обеспечивает четкое разделение данных, логики и интерфейса. Основные компоненты реализации включают:

Model — отвечает за взаимодействие с базой данных (sqlite3), выполнение операций CRUD и обработку данных.

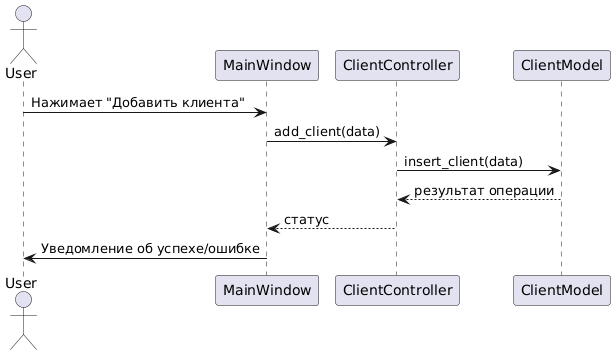
View — графический интерфейс на PyQt5, отображающий данные и предоставляющий пользователю элементы управления (таблицы, кнопки, формы).

Controller — связующее звено между Model и View. Управляет действиями пользователя, обращается к модели для выполнения операций и обновляет интерфейс.

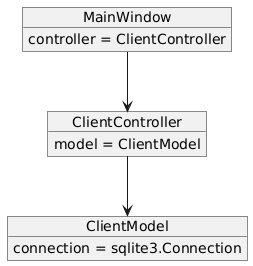
**Диаграмма классов**



**Диаграмма последовательности: добавление клиента**



**Диаграмма объектов**



**3 Руководство пользователя**

**3.1 Инструкция по установке и запуску**

**Шаг 1: Установите Python**

Перейдите на официальный сайт Python: https://www.python.org/.

Скачайте последнюю стабильную версию Python, совместимую с вашей операционной системой (Windows, macOS, Linux).

Во время установки:

Убедитесь, что вы отметили опцию "Add Python to PATH" (для Windows).

Завершите установку, следуя инструкциям установщика.

**Шаг 2: Установите необходимые библиотеки**

Откройте терминал (командную строку) вашей системы.

Убедитесь, что Python установлен, выполнив команду:

python --version

или

python3 --version

Установите менеджер пакетов pip (если он еще не установлен) командой:

python -m ensurepip --upgrade

Установите библиотеки, необходимые для работы приложения:

pip install pyqt5

**Шаг 3: Скачайте файлы приложения**

Перейдите в репозиторий или папку с исходным кодом приложения (например, GitHub, облачное хранилище).

Скачайте архив с исходным кодом или клонируйте репозиторий командой:

git clone <URL репозитория>

Распакуйте файлы в удобное место на вашем компьютере.

**Шаг 4: Настройте базу данных**

В папке приложения должен находиться файл database.db (если он отсутствует, приложение создаст его автоматически при первом запуске).

Убедитесь, что у вас есть права записи в папке, где расположен файл.

**Шаг 5: Запустите приложение**

Перейдите в папку с исходным кодом приложения:

cd path/to/application

Запустите приложение с помощью команды:

python main.py

или

python3 main.py

**Шаг 6: Используйте приложение**

После запуска откроется главное окно программы. Вы можете начать добавлять, редактировать или удалять записи, а также использовать функции импорта/экспорта.

**Шаг 7: Решение возможных проблем**

Если приложение не запускается:

Убедитесь, что все зависимости установлены корректно.

Проверьте, что версия Python 3.6 или выше.

Если вы получаете ошибку о нехватке прав:

Убедитесь, что вы запускаете приложение из папки с правами записи.

Для macOS и Linux может понадобиться запуск приложения с использованием:

python3 main.py

вместо python main.py.

Поддерживаемые платформы

Windows 10 и выше.

macOS (с установленным Python).

Linux (любая современная дистрибуция, поддерживающая Python 3).

**Рекомендации**

Используйте виртуальное окружение для изоляции библиотек приложения:

python -m venv venv

source venv/bin/activate # Для macOS/Linux

venv\Scripts\activate # Для Windows

pip install -r requirements.txt

Это поможет избежать конфликтов с другими установленными пакетами.

**3.2 Описание функций и возможностей**

Приложение предоставляет пользователям множество удобных функций для управления клиентской базой данных. Основной функционал включает добавление, удаление, обновление информации о клиентах, поиск и фильтрацию, а также импорт и экспорт данных.

Добавление клиента позволяет пользователям вносить новых клиентов в базу данных. Форма включает поля для ввода имени, фамилии, email, телефона. Это упрощает процесс организации данных, предоставляя удобную структуру для их хранения. Например, менеджер может нажать кнопку "Добавить клиента", заполнить форму и сохранить данные, которые тут же появятся в таблице.

Функция удаления клиента помогает избавиться от устаревшей или ошибочной информации. Для этого достаточно выделить запись в таблице и нажать кнопку "Удалить клиента". После подтверждения действия данные удаляются из базы.

Функция обновления данных клиента позволяет редактировать существующие записи, чтобы поддерживать их актуальность. Например, если у клиента изменился номер телефона, менеджер может выделить его запись, нажать кнопку "Обновить клиента", изменить данные и сохранить их.

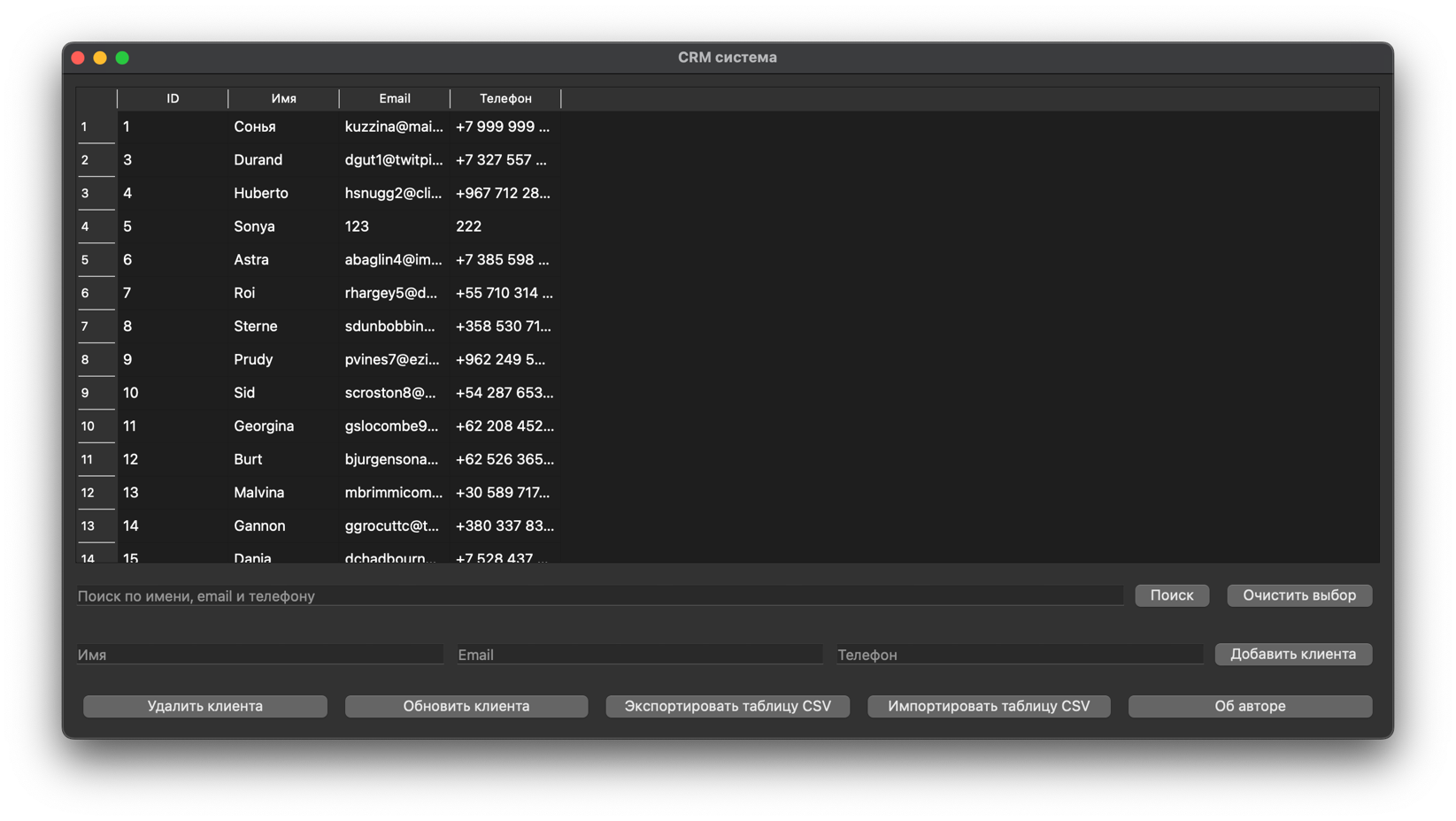
Для удобства работы с большими объемами информации предусмотрены поиск и фильтрация. Пользователь может быстро находить клиента, вводя его имя, email или телефон в строку поиска. Таблица тут же отобразит только подходящие записи, что существенно экономит время.

Импорт данных из CSV-файла облегчает добавление большого числа клиентов сразу. Пользователь выбирает файл в диалоговом окне, а прогресс бар показывает статус загрузки данных. Например, компания может загрузить в приложение клиентскую базу, хранящуюся в Excel, после её сохранения в формате CSV.

Экспорт данных в CSV позволяет выгружать клиентскую базу в файл для создания отчетов или использования в других системах. Пользователь выбирает место сохранения файла, и данные экспортируются в таблицу, исключая поле с идентификатором клиента.

Для отображения информации о разработчике предусмотрена кнопка "Об авторе", которая открывает всплывающее окно с контактными данными. Это полезно, если пользователю требуется поддержка или дополнительные консультации.

Все функции приложения сопровождаются интуитивно понятным интерфейсом, что делает его удобным даже для неопытных пользователей. От формы добавления клиента до строки поиска или прогресса импорта — каждый элемент спроектирован с целью улучшения пользовательского опыта.



ЗАКЛЮЧЕНИЕ

В рамках проекта было разработано приложение для управления клиентской базой данных с использованием PyQt5 и SQLite. Приложение включает функционал для добавления, редактирования, удаления клиентов, поиска и фильтрации данных, а также импорт и экспорт информации в формате CSV. Реализован удобный графический интерфейс, поддерживающий работу на различных платформах. Все задачи выполнены с учетом требований к производительности, безопасности и удобству использования, что обеспечивает стабильную и эффективную работу приложения.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1 Grayson, A. Python and SQLite. Apress, 2010

2 Саммерфилд, М. Rapid GUI Programming with Python and Qt: The Definitive Guide to PyQt Programming. - Pearson Education, 2007

3 Официальная документация PyQt5 // URL: https://riverbankcomputing.com/software/pyqt/intro (дата обращения: 07.12.2024).