

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ**

**ВЫСШЕГО ОБРАЗОВАНИЯ**

**«РОССИЙСКАЯ АКАДЕМИЯ НАРОДНОГО ХОЗЯЙСТВА И ГОСУДАРСТВЕННОЙ**

**СЛУЖБЫ ПРИ ПРЕЗИДЕHTE РОССИЙСКОЙ Ф ЕДЕРАЦИИ»**

**КОЛЛЕДЖ МНОГОУРОВНЕВОГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ**

**КУРСОВОЙ ПРОЕКТ**

**по дисциплине: МДК 01.01 Разработка программных модулей**

**Тема: Разработка программного модуля**

**системы** **«Нарушители правил дорожного движения»**

**Специальность 09.02.07 «Информационные системы и программирование»**

|  |  |  |
| --- | --- | --- |
| **Выполнил студент(ка) группы 311ИС-22** | **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_** | **Б. С. Абанькин** |
| **Руководитель** | **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_** | **Н.И. Кручинкина** |

**Москва 2024**

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ**

**ВЫСШЕГО ОБРАЗОВАНИЯ**

**«РОССИЙСКАЯ АКАДЕМИЯ НАРОДНОГО ХОЗЯЙСТВА И ГОСУДАРСТВЕННОЙ**

**СЛУЖБЫ ПРИ ПРЕЗИДЕHTE РОССИЙСКОЙ Ф ЕДЕРАЦИИ»**

**КОЛЛЕДЖ МНОГОУРОВНЕВОГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ**

****

**УТВЕРЖДАЮ**

**Зам. директора КМПО**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_С.Ф. Гасанов**

**«\_\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2024 г.**

**ЗАДАНИЕ НА КУРСОВОЙ ПРОЕКТ**

**По дисциплине: МДК 01.01 РАЗРАБОТКА ПРОГРАММНЫХ МОДУЛЕЙ**

**Специальность 09.02.07 «Информационные системы и программирование»**

**Студент группы 311ИС-22 Богдан Абанькин**

**ТЕМА: «Разработка программного модуля системы «Нарушители правил дорожного движения»»**

Дата выдачи задания «\_\_\_\_\_» \_\_\_\_\_\_\_\_\_ 2024 г.

Срок сдачи проекта «\_\_\_\_\_» \_\_\_\_\_\_\_\_\_\_ 2024 г.

**Москва 2024**

**Перечень вопросов, подлежащих разработке:**

Введение

1. АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ

1.1 Предпроектное обследование

1.2

2. ПРОЕКТИРОВАНИЕ И РАЗРАБОТКА МОДУЛЯ

2.1 Эскизное проектирование

2.1.1

2.1.2

2.2 Проектирование базы данных

2.3 Техническое проектирование

2.3.1 Разработка архитектуры

2.3.2 Защита доступа к данным

3. ОТЛАДКА И ТЕСТИРОВАНИЕ МОДУЛЯ

3.1

3.2.

Заключение

**Исходные данные:** *исходные данные студент собирает самостоятельно*.

**Перечень разрабатываемых материалов для визуализации:** электронная презентация, содержащая в том числе:

1. Блок-схемы алгоритмов программного модуля
2. Use-case диаграмма.
3. ER-диаграмма базы данных.
4. Архитектура программного модуля
5. Экранные формы интерфейса.

Задание выдал:

Руководитель курсового проекта Н.И. Кручинкина

Задание принял к исполнению Б.С. Абанькин

*Рассмотрено*

на заседании предметно-цикловой комиссии

информационных технологий и системного

администрирования

Протокол № \_\_\_*от «\_\_» \_\_\_\_\_\_\_\_\_* 2024 г.

Председатель ПЦК \_\_\_\_\_\_\_\_\_\_\_\_\_

**СОДЕРЖАНИЕ**

[Введение 3](#_Toc185190165)

[1. АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ 6](#_Toc185190166)

1.1 Понятие базы данных и СУБД 6

1.2 Современные программные решения для хранения и обработки данных пользователя 7

[2. ПРОЕКТИРОВАНИЕ И РАЗРАБОТКА МОДУЛЯ 9](#_Toc185190167)

[2.1 Проектирование базы данных 9](#_Toc185190168)

[2.2 Разработка и описание базы данных 9](#_Toc185190169)

[2.3 Разработка программного модуля на языке Python 13](#_Toc185190171)

[3. ПРАКТИЧЕСКАЯ РЕАЛИЗАЦИЯ ПРОГРАММНОГО МОДУЛЯ 17](#_Toc185190172)

3.1 Описание среды разработки 17

3.2 Реализация пользовательского интерфейса 18

3.3 Тестирование модуля 21

[ЗАКЛЮЧЕНИЕ 22](#_Toc185190173)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 24](#_Toc185190174)

[ПРИЛОЖЕНИЕ 1 26](#_Toc185190175)

[ПРИЛОЖЕНИЕ 2 29](#_Toc185190176)

ВВЕДЕНИЕ

С ростом численности населения городов увеличивается не только количество транспортных средств, но и число пешеходов, активно участвующих в дорожном движении. Это ведет к возрастанию плотности транспортных потоков и, как следствие, увеличению количества нарушений правил дорожного движения. Такие нарушения не только создают угрозу безопасности на дорогах, но и становятся причиной значительных социальных и экономических потерь, включая ущерб жизни и здоровью граждан, материальные убытки и нагрузку на инфраструктуру. Существующие методы фиксации и анализа нарушений зачастую оказываются недостаточно оперативными и эффективными для обработки огромных массивов данных, что требует внедрения автоматизированных решений.

Одной из ключевых мер предотвращения нарушений является наложение штрафов на виновных лиц. Однако процесс выявления и обработки таких случаев требует значительных временных затрат и трудозатрат, особенно в условиях больших городов. В связи с этим возникает потребность в разработке специализированных программных средств, способных ускорить и упростить этот процесс. Современные информационные технологии предоставляют широкие возможности для автоматизации таких задач, что делает их незаменимыми в работе с большими объемами данных.

Следует отметить, что разработка программных решений для учета нарушений играет важную роль не только с точки зрения повышения эффективности контроля, но и в образовательной сфере. Такие проекты позволяют студентам приобрести навыки, необходимые выпускнику по специальности «Информационные системы и программирование». Создание программного модуля, описанного в данном курсовом проекте, позволяет продемонстрировать на практике этапы разработки программного обеспечения, начиная с анализа требований и заканчивая тестированием готового продукта.

Целью данного курсового проекта является разработка программного модуля системы "Нарушители правил дорожного движения", предназначенного для хранения и обработки информации о нарушениях. Основной задачей является создание функционального инструмента, который позволит сократить временные затраты на обработку данных, повысить точность фиксации нарушений и упростить взаимодействие с соответствующими службами.

Для достижения поставленной цели предполагается решение следующих задач:

1. анализ существующих методов и программных средств;
2. проектирование структуры и функциональных возможностей программного модуля;
3. реализация модуля с использованием языка программирования Python и его стандартного модуля tkinter для реализации пользовательского интерфейса;
4. тестирование и оценка эффективности разработанного решения.

Объектом исследования является процесс управления нарушениями ПДД, включая учет нарушителей, регистрацию событий, назначение штрафов и контроль их оплаты.

Предметом исследования является разработка программного модуля, который позволит автоматизировать данные процессы и упростить взаимодействие между сотрудниками дорожной полиции и органами управления дорожным движением.

Практическая значимость проекта заключается в возможности его применения для учета нарушений правил дорожного движения в локальных и ведомственных системах, а также возможности его последующей доработки для интеграции с государственными автоматизированными системами. Разработанное решение может быть использовано как основа для создания более масштабных систем управления дорожной безопасностью, включая онлайн-сервисы и мобильные приложения. Настоящий проект охватывает этапы проектирования базы данных, реализации программного модуля и создания интуитивно понятного интерфейса, что позволяет эффективно решать задачи учета и обработки данных о нарушениях, а также демонстрирует возможности современной разработки информационных систем.

1. АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ
   1. Понятие базы данных и СУБД

База данных (БД) – организованная структура данных, хранящихся в компьютерной системе, которая позволяет эффективно управлять большими объемами информации, обеспечивая их хранение, извлечение, обновление и удаление. Базы данных используются в различных областях, включая бизнес, науку, медицину и образование, для управления данными, такими как записи о клиентах, финансовые транзакции, научные данные и многое другое. Основная цель базы данных – обеспечить надежное и эффективное управление данными, минимизируя избыточность и обеспечивая целостность информации.

Система управления базами данных (СУБД) представляют собою программное обеспечение, которое предоставляет интерфейс для взаимодействия с базой данных. СУБД позволяет пользователям и приложениям выполнять операции над данными, такие как создание, чтение, обновление и удаление (CRUD-операции). Основные функции СУБД включают управление данными, обеспечение их целостности и безопасности.

СУБД могут быть реляционными, нереляционные, документно-ориентированными и другими, в зависимости от типа данных и требований к их управлению.

Реляционные СУБД, такие как MySQL, PostgreSQL, Oracle используют таблицы для хранения данных и SQL (Structured Query Language) для выполнения операций над ними. Реляционные СУБД обеспечивают высокую степень структурированности и целостности данных, а также поддерживают сложные запросы.

Нереляционные СУБД, такие как Redis, Apache Cassandra используются чтобы упростить масштабируемость, увеличить производительность.

Документно - ориентированные СУБД, такие как MongoDB и CouchDB, хранят данные в виде документов, обычно в формате JSON или BSON. Эти СУБД хорошо подходят для работы с неструктурированными или полуструктурированными данными, такими как текстовые документы, веб-страницы и конфигурационные файлы. Они обеспечивают гибкость и масштабируемость, что делает их популярными для веб-приложений и больших данных.

СУБД обеспечивает безопасность данных, включая аутентификацию и авторизацию пользователей, шифрование данных и управление доступом, что позволяет защитить данные от несанкционированного доступа и обеспечить их конфиденциальность. СУБД также поддерживают резервное копирование и восстановление данных, что позволяет защитить данные от потерь и обеспечить их восстановление в случае сбоя.

1.2 Современные программные решения для хранения и обработки данных пользователя

На сегодняшний день данные пользователя стали важным ресурсом, используемым во всех сферах деятельности - от бизнеса до науки и образования. Доступные на сегодняшний день программные решения для хранения и обработки данных предлагают широкий спектр инструментов, обеспечивающих надежность, безопасность, удобство работы и масштабируемость. Ключевым аспектом современных программных систем для хранения данных является их способность работать с различными типами данных. И все-таки, реляционные базы данных продолжают оставаться популярными для хранения структурированных данных благодаря своей надежности и поддержке сложных запросов. Подобные системы обеспечивают высокую степень целостности данных и поддерживают транзакции, что делает их идеальными для приложений, требующих строгой согласованности данных.

Однако с ростом объема неструктурированных данных, таких как текстовые документы, изображения и видео, возникла необходимость в новых типах баз данных. Документно-ориентированные базы данных, такие как MongoDB и CouchDB, стали популярными для хранения и обработки таких данных, которые используют формат JSON или BSON для хранения данных, что обеспечивает гибкость и масштабируемость. Они позволяют легко добавлять новые поля и изменять структуру данных без необходимости изменения схемы базы данных, что делает их идеальными для веб-приложений и больших данных.

Современные программные решения для хранения данных включают облачные базы данных, такие как Amazon RDS, Google Cloud SQL и Microsoft Azure SQL Database. Эти системы предоставляют масштабируемость, высокую доступность и управляемость, что делает их идеальными для приложений, требующих высокой производительности и надежности. Облачные базы данных позволяют легко масштабировать ресурсы в зависимости от потребностей приложения, обеспечивая при этом высокую доступность и отказоустойчивость.

Безопасность данных является критически важным аспектом современных программных решений для хранения и обработки данных. Эти системы предоставляют различные механизмы для обеспечения безопасности данных, включая аутентификацию и авторизацию пользователей, шифрование данных и управление доступом, что позволяет защитить данные от несанкционированного доступа и обеспечить их конфиденциальность. Кроме того, современные СУБД поддерживают резервное копирование и восстановление данных, что позволяет защитить данные от потерь и обеспечить их восстановление в случае сбоя.

# ПРОЕКТИРОВАНИЕ И РАЗРАБОТКА МОДУЛЯ

## 2.1 Проектирование базы данных

Прежде чем начать разработку UI (user interface, пользовательский интерфейс) и программного модуля необходимо создать базу данных о нарушителях ПДД для систематизации и хранения информации. Рассмотрим основные функции для разрабатываемой системы, которые помогут спроектировать структуру базы данных. Предположим, что система должна хранить информацию о нарушителях (пусть это будут водители и пешеходы), их нарушениях, событиях и штрафах (статус оплаты, дата оплаты). Также включим возможность поиска нарушителей по имени, однако можно реализовать и функцию поиска по типу нарушения, статусу оплаты и другим параметрам. Кроме того, будет полезно если, система будет поддерживать CRUD-операции: добавление, просмотр, редактирование и удаления записей.

2.2 Разработка и описание базы данных

База данных была разработана с использованием среды SQLite для Windows, которая была выбрана за свою легковесность, широкую совместимость и простоту в использовании. SQLite, будучи встроенной базой данных, идеально подходит для локальных приложений, где требуются надежные и быстро обрабатываемые хранилища данных.

Процесс проектирования начался с определения основных требований к системе. База данных будет содержать четыре основные сущности: нарушители, нарушения, штрафы и события. Рисунок 1 демонстрирует ER-диаграмму базы данных traffic\_violations.db для учета нарушителей и нарушений ПДД. Из рисунка можно заметить, что структура таблиц спроектирована для минимизации избыточности данных и соответствия третьей нормальной форме (3NF).

Изображение выглядит как текст, снимок экрана, диаграмма, Шрифт

Автоматически созданное описание

Рисунок 1 – IDEF1X диаграмма проектируемой БД

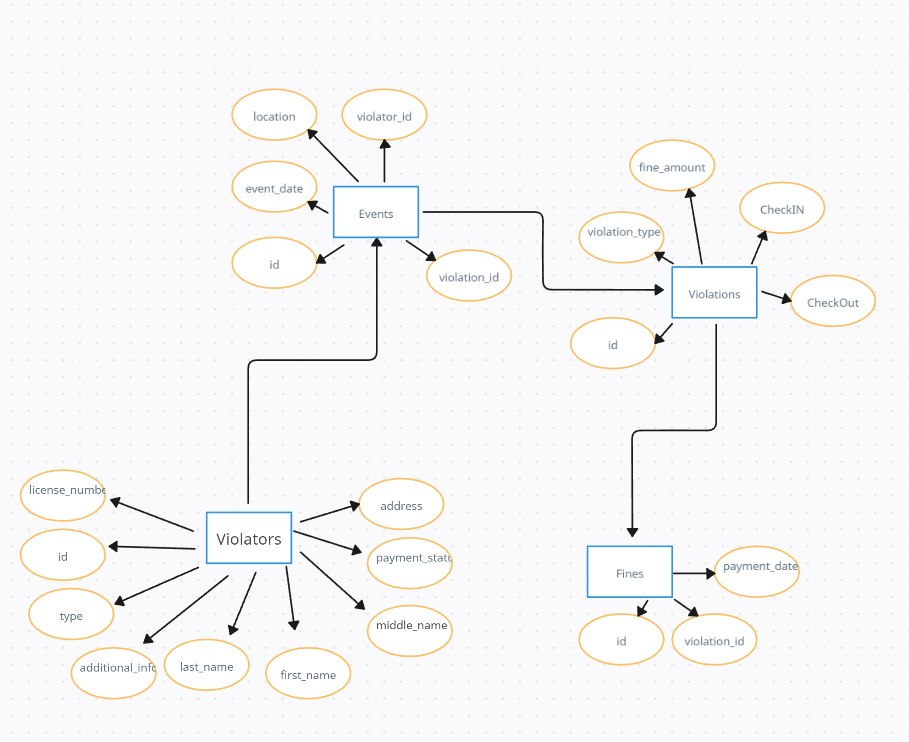
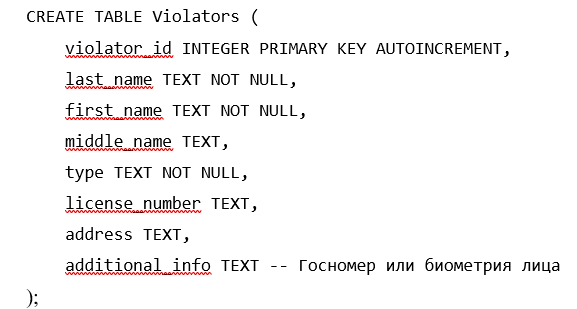
****

Рисунок 2 – ER-диаграмма проектируемой БД

Вначале была создана таблица Violators, которая хранит информацию о нарушителях. Поля last\_name, first\_name и middle\_name используются для хранения фамилии, имени и отчества. В таблицу включены поля для типа нарушителя, номера водительского удостоверения, адреса и дополнительной информации. Для создания таблицы Violators использовался следующий SQL-запрос:



Следующим шагом было создание таблицы Violations (нарушения), которая будет хранить типы нарушений и соответствующие суммы штрафов:

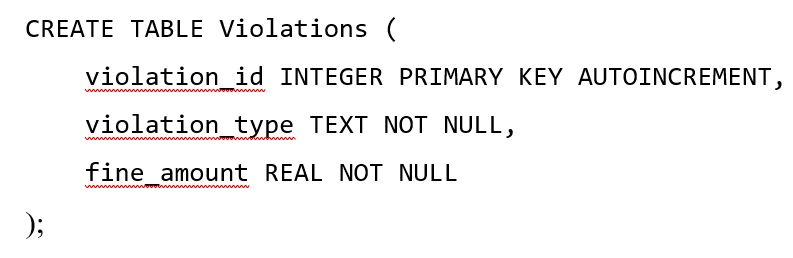
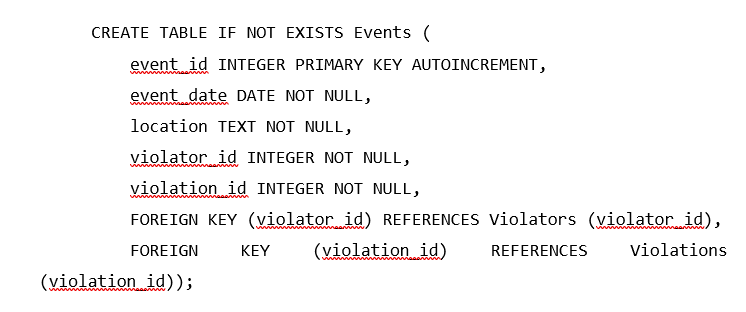


Таблица Fines содержит информацию о штрафах, включая идентификатор нарушения, статус оплаты и дату оплаты.

И наконец, таблица Events связывает информацию о нарушителе, нарушении и месте события, также фиксируя дату события:



Все таблицы в базе данных взаимосвязаны для управления информацией о нарушителях ПДД, их нарушениях, штрафах и событиях. Так, таблица Violators содержит информацию о нарушителях, включая их фамилию, имя, отчество, тип (например, водитель или пешеход), адрес и дополнительные данные. Таблица Violations описывает типы нарушений и соответствующие суммы штрафов. Связь между нарушителями и типами нарушений реализована через таблицу Events, где фиксируются события нарушений с указанием даты, места, идентификатора нарушителя (violator\_id) и идентификатора нарушения (violation\_id). Fines хранит данные о штрафах, включая их статус оплаты и дату, и связана с таблицей Violations через поле violation\_id, что позволяет отслеживать платежи по конкретным нарушениям. Таким образом, таблицы связаны через ключи violator\_id и violation\_id, создавая целостную модель данных для учёта всех аспектов нарушений ПДД.

Созданная база данных обеспечивает структурированный подход к управлению данными о нарушителях ПДД и готова к интеграции в программный модуль и позволяет выполнять как поставленные задачи, такие как, например, поиск нарушителей, так и аналитические запросы, которые реализуются внутри модуля.

Таблица 1 - Объекты предметной области

|  |  |  |
| --- | --- | --- |
| № | Наименование объекта | Краткое описание |
| 1 | Нарушители (Violators) | Хранит, информацию о нарушителях |
| 2 | Нарушения (Violations) | Описание нарушений и хранение информации о сумме нарушения |
| 3 | События (Events) | Хранит информацию о месте и времени |
| 4 | Штрафы (Fines) | Хранит информацию о статусе платежа и дате платежа |

После того как мы описали объекты нашей предметной области, рассмотрим атрибуты, присущие каждому из них.

Таблица 2 - Сущность «Нарушители»

|  |  |  |
| --- | --- | --- |
| Атрибут | Тип | Описание |
| violator\_id | INT PRIMARY KEY | Уникальный идентификатор нарушителей |
| last\_name | TEXT NOT NULL | Имя нарушителей |
| first\_name | TEXT NOT NULL | Фамилия нарушителей |
| middle\_name | TEXT NOT NULL | Отчество нарушителей |
| type | TEXT NOT NULL | Тип нарушителя (водитель, пешеход) |
| license\_number | TEXT | Номер лицензии |
| address | TEXT NOT NULL | Адрес нарушителей |
| additional\_info | TEXT | Доп. Информация о нарушителях (номер машины, лицо) |

Таблица 3 - Сущность «Нарушения»

|  |  |  |
| --- | --- | --- |
| Атрибут | Тип | Описание |
| violation\_id | INT PRIMARY KEY | Уникальный идентификатор нарушений |
| violation\_type | TEXT NOT NULL | Тип нарушения |
| fine\_amount | REAL NOT NULL | Сумма штрафа за нарушение |

Таблица 4 - Сущность «События»

|  |  |  |
| --- | --- | --- |
| Атрибут | Тип | Описание |
| event\_id | INT PRIMARY KEY | Уникальный идентификатор событий |
| event\_date | DATE NOT NULL | Дата нарушения |
| location | TEXT NOT NULL | Место нарушения |
| violator\_id | INT NOT NULL | Уникальный идентификатор нарушителей |
| violation­\_id | INT NOT NULL | Уникальный идентификатор нарушений |

Таблица 5 - Сущность «Штрафы»

|  |  |  |
| --- | --- | --- |
| Атрибут | Тип | Описание |
| fine\_id | INT PRIMARY KEY | Уникальный идентификатор штрафов |
| violation­\_id | INT NOT NULL | Уникальный идентификатор нарушений |
| payment\_status | TEXT NOT NULL | Статус платежа (оплачен, не оплачен) |
| payment\_date | DATE | Дата платежа |

2.3 Разработка программного модуля на языке Python

Для реализации разрабатываемого программного модуля системы "Нарушители правил дорожного движения" использована PyCharm – интегрированную среду разработки на языке Python.

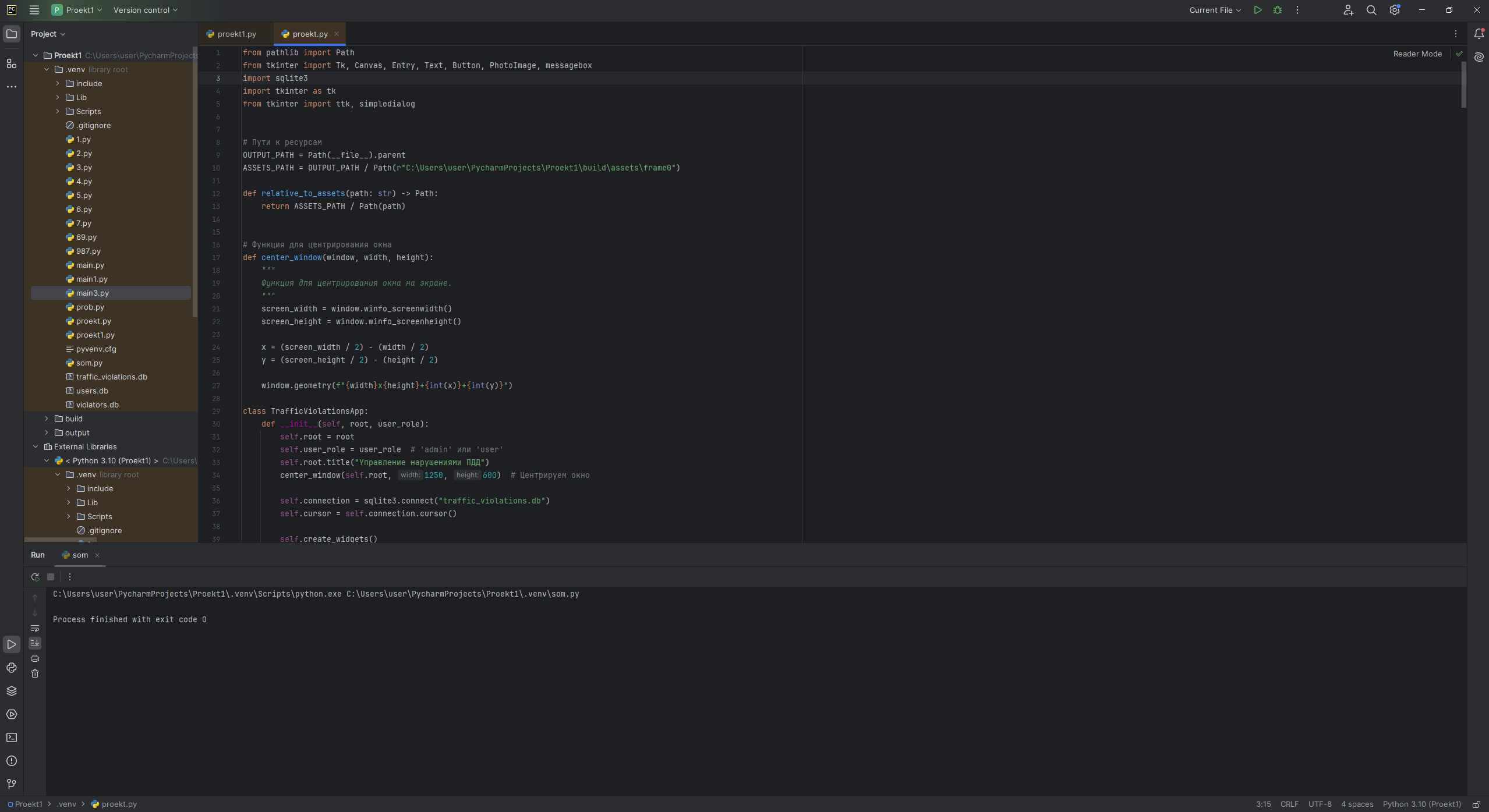
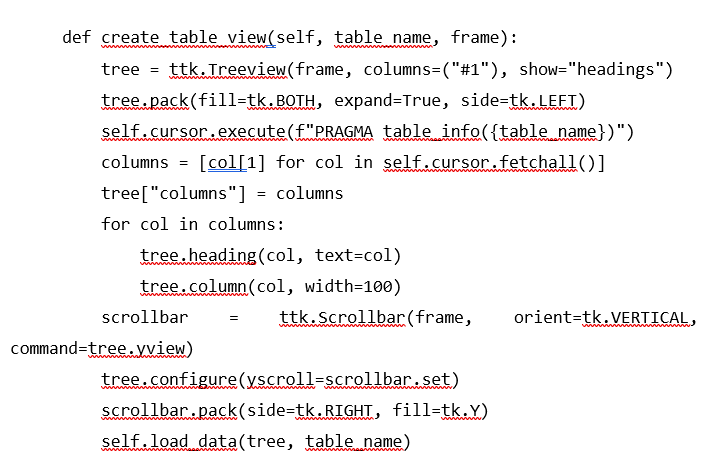
****

Рисунок 3 - Среда разработки PyCharm

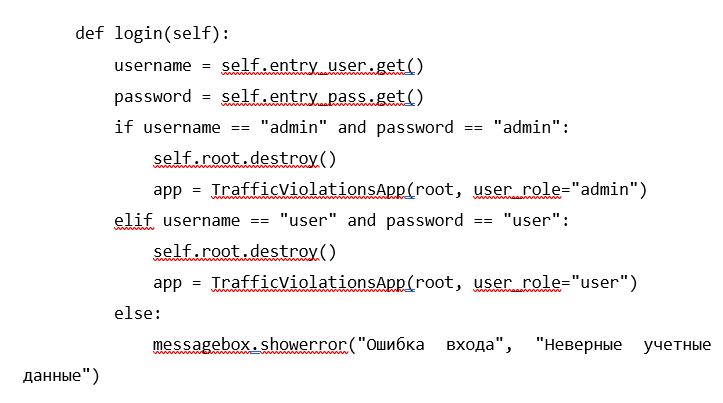
В рамках разработки была реализована система с двумя ролями пользователей: администратор и пользователь, каждая из которых имеет доступ к определенному набору функций.

Реализовано динамическое обновление данных: при выполнении операций изменения данные в таблицах автоматически обновляются.

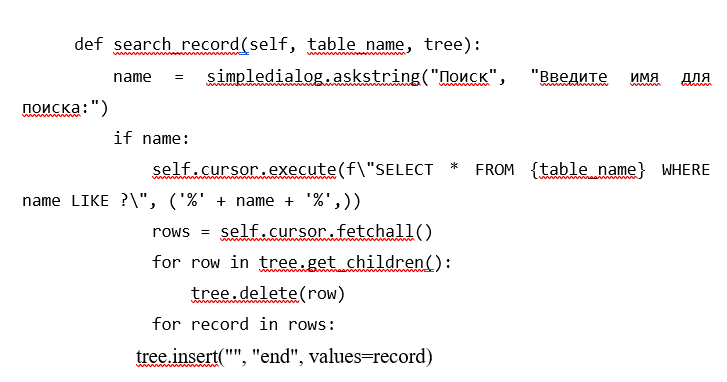
Для реализации удобного и интуитивного интерфейса была использована библиотека Tkinter, входящая в стандартный набор инструментов Python. Интерфейс включает: главное окно с вкладками для работы с таблицами базы данных, виджет Treeview, предназначенный для отображения записей таблиц в табличном формате, кнопки управления, позволяющие добавлять, редактировать, удалять записи, а также выполнять поиск:



Особенности интерфейса включают поддержку нескольких вкладок, каждая из которых соответствует отдельной таблице базы данных, что упрощает навигацию и работу с данными. При открытии вкладки данные из соответствующей таблицы автоматически загружаются и отображаются в табличном формате. Для каждой вкладки предусмотрены кнопки для выполнения операций добавления, редактирования и удаления записей. Для повышения безопасности и удобства работы реализована система авторизации, где пользователь вводит имя и пароль, после чего система определяет его роль. Администратор имеет полный доступ к функциям модуля, включая просмотр, добавление, редактирование и удаление записей, тогда как пользователь имеет доступ только к функции поиска записей по имени. Пример реализации авторизации:



Добавление записей реализовано через всплывающее окно, где пользователь заполняет все необходимые поля, после чего данные сохраняются в базу данных. Редактирование записей осуществляется путем загрузки данных выбранной записи в форму, где их можно изменить, а после сохранения изменения фиксируются в базе данных. Удаление записей выполняется после подтверждения пользователем. Поиск записей позволяет пользователю искать записи по имени, при этом для администратора результаты поиска отображаются в таблице, а для пользователей – в виде сообщения с деталями. Реализация поиска записи:



Разработанный программный модуль обеспечивает удобный и интуитивно понятный интерфейс для работы с данными о нарушениях правил дорожного движения. Благодаря использованию Tkinter модуль может быть легко доработан или адаптирован для новых задач. Применение разграничения прав доступа гарантирует безопасность данных и удобство использования для разных категорий пользователей.

# ТЕСТИРОВАНИЕ МОДУЛЯ

3.1 Описание среды разработки

Для разработки программного модуля системы "Нарушители правил дорожного движения были выбраны следующие программные инструменты:

1. Среда разработки: PyCharm — это полнофункциональная среда разработки (IDE) для языка программирования Python. Она разработана компанией JetBrains и предлагает широкий набор инструментов, помогающих программистам увеличить производительность и эффективность при разработке программного обеспечения на Python.
2. Язык программирования: Python — это интерпретируемый высокоуровневый язык программирования, который прост в изучении, при этом мощный и гибкий.
3. Библиотеки: Tkinter — это стандартная библиотека Python для создания графических интерфейсов. Она предоставляет простой и интуитивно понятный способ разработки GUI-приложений. Выбор Tkinter обусловлен его доступностью, простотой использования и достаточной функциональностью для реализации требований данной курсовой работы.
4. База данных: sqlite3 это встраиваемая реляционная база данных, которая предлагает мощные возможности хранения и управления данными в локальном файле. Она является частью стандартной библиотеки Python и позволяет разработчикам создавать, изменять и извлекать данные из базы данных SQLite3 с помощью языка SQL.
5. Figma – инструмент для проектирования пользовательского интерфейса программного модуля. Который позволил визуализировать структуру и дизайн приложения, определить расположение элементов интерфейса и обеспечить согласованность дизайна.

Выбранный инструментарий обеспечивает необходимый уровень функциональности для разработки консольного приложения, сохраняя простоту использования и обеспечивая эффективность в процессе работы.

3.2 Реализация пользовательского интерфейса

В процессе работы приложения реализовано автоматическое обновление данных в таблицах после каждого изменения, что исключает необходимость ручного обновления, что обеспечивает удобство работы и позволяет пользователю сразу видеть актуальную информацию.

При запуске приложения пользователь попадает в окно авторизации. Вход в систему осуществлялся с использованием двух учетных записей: администратора и пользователя. После ввода логина и пароля (пользователь заранее знает свой логин и пароль) пользователь попадает в окно Управление нарушениями ПДД. Интерфейс окон показан на рисунках 5-6.

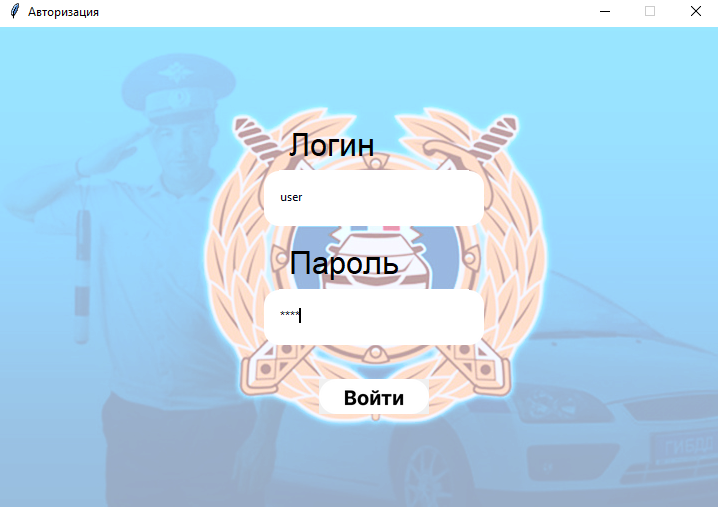
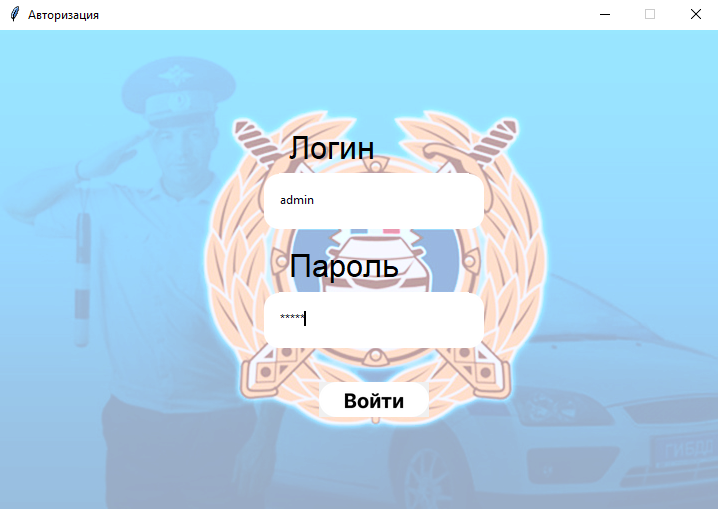
****

Рисунок 4 - Процесс авторизации (администратор или пользователь)

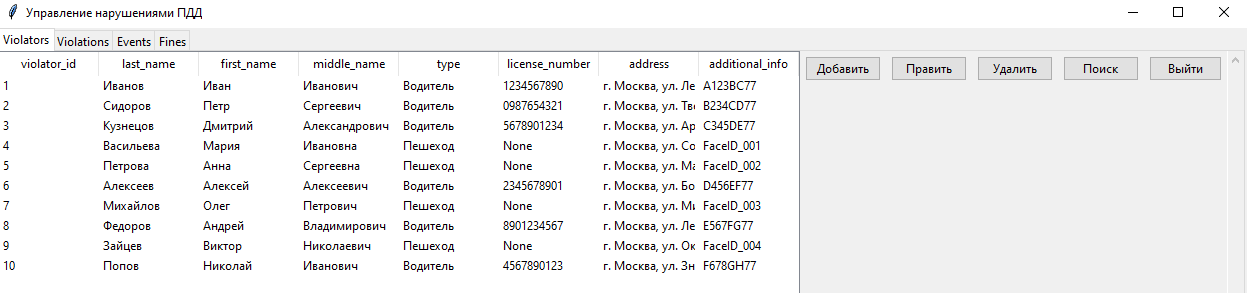


Рисунок 5 – Интерфейс администратора

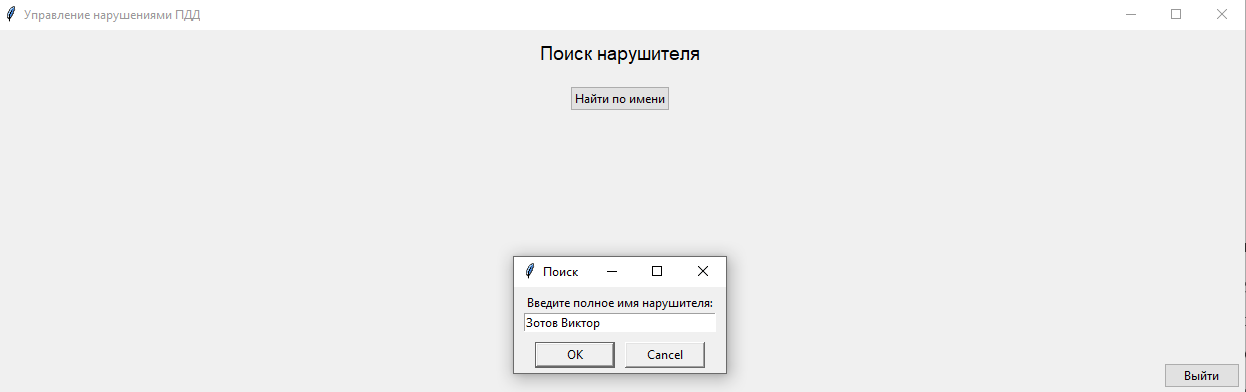
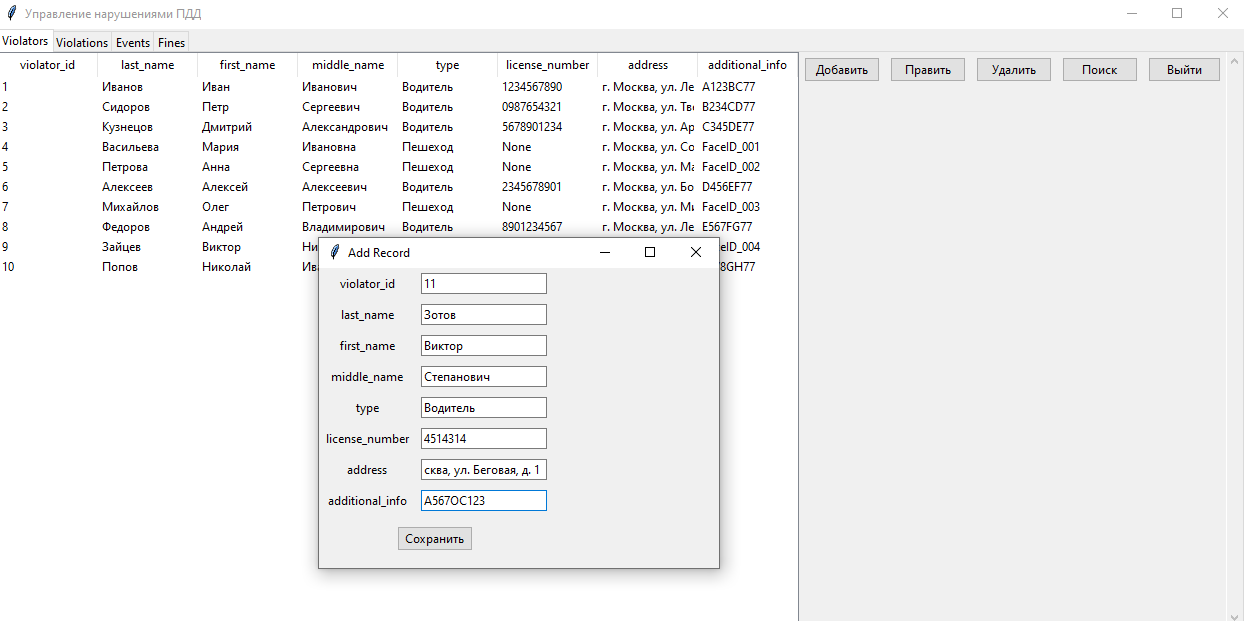
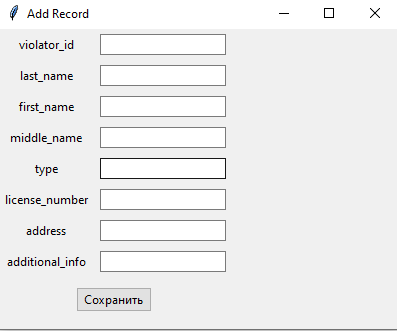


Рисунок 6 - Интерфейс пользователя

У администратора открывается интерфейс с доступом ко всем таблицам базы данных и возможностью выполнения всех операций. Если администратор кликнет на кнопку «добавить», то открывается окно добавления записи, где пользователь может записать следующие данные: идентификатор нарушителя, имя, фамилия, отчество, тип нарушителя (водитель, пешеход), номер водительского удостоверения, адрес место жительства, государственный номер автомобиля. (Рисунок 7)



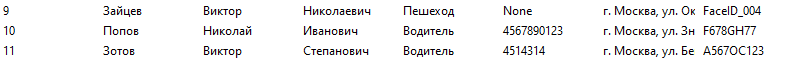


Рисунок 7 – Add Record (Добавление записи)

Если администратор кликнет на кнопку «Править» - он может внести изменения в данных. (Рисунок 8)

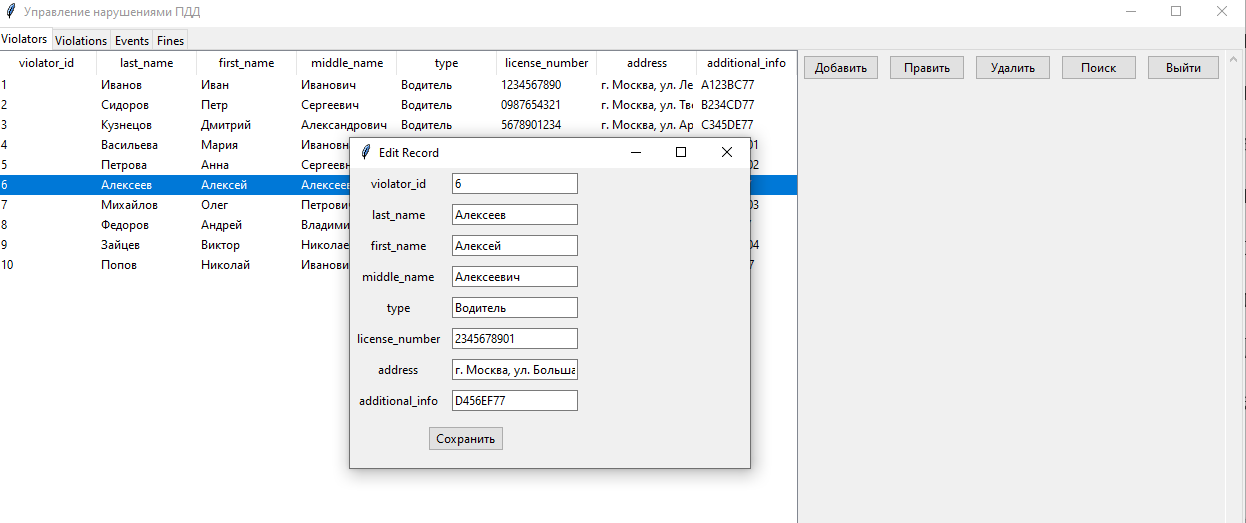


Рисунок 8 – Edit Record (Редактировать запись)

Кликнув на кнопку «Удалить» - администратор может удалить выбранного нарушителя. (Рисунок 9)

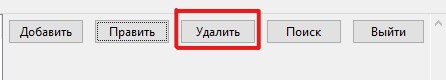


Рисунок 9 – кнопка «Удалить»

Кликнув на кнопку «Поиск» администратору откроется окно поиска, в которое он должен ввести имя для поиска. (Рисунок 10)

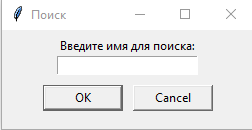


Рисунок 10 – окно «Поиск»

Кликнув на кнопку «Выйти», администратор, откроется окно выход с подтверждением его выбора, если администратор нажмет на кнопку, «да» он выйдет из окна Управление нарушителями ПДД. (Рисунок 11)

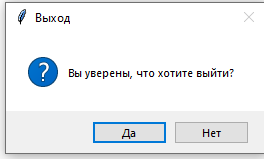


Рисунок 11 – окно «Выход»

У пользователя открывается интерфейс, ограниченный только функционалом поиска записей. Если пользователь кликнет на кнопку «Найти по имени» откроется окно «Поиска», где нужно ввести полное имя нарушителя для нахождения его данных. После чего пользователь нажимает на кнопку «ок» и открывается окно «Результаты поиска» в котором находятся данные нарушителя. (Рисунок 12)

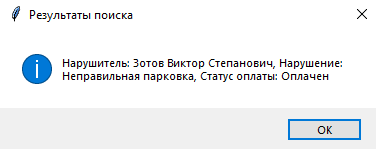
****

Рисунок 12 – окно «Результат поиска»

3.3 Тестирование

В ходе тестирования были протестированы случаи ввода некорректных данных, такие как неверный логин или пароль, а также попытка входа без указания данных, пустые поля при добавлении записи, выбор отсутствующего элемента для редактирования или удаления, а также поиск несуществующего имени. Во всех случаях система корректно идентифицировала ошибки, выводила информативные сообщения и предотвращала выполнение некорректных операций. (Рисунок 13)

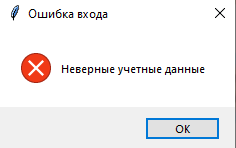


Рисунок 13 – Ошибка входа

Кроме того, предусмотрена обработка ошибок базы данных: если операция не удалась, программа выводит сообщение с подробным описанием проблемы, что помогает быстро найти и исправить причину. (Рисунок 14)

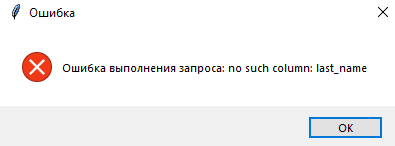


Рисунок 14 – Сообщение об ошибке

ЗАКЛЮЧЕНИЕ

Был успешно разработан программный модуль системы «Нарушители правил дорожного движения», который представляет собой интегрированное решение, сочетающее в себе графический интерфейс пользователя, базу данных и аналитические функции. Посредством программы на Python с использованием стандартной библиотеки tkinter, разработанный модуль предоставляет возможность для работы с информацией о нарушителях, включая добавление, редактирование, удаление и поиск записей. Благодаря реализации роли администратора и пользователя система обладает гибкостью и может быть адаптирована для различных категорий сотрудников. Администратору доступны все функции управления данными, тогда как пользователь ограничен функционалом поиска, что гарантирует безопасность и конфиденциальность данных.

Практическая значимость проекта заключается в возможности его использования как в учебных, так и в практических целях. Система может быть полезна для государственных структур, занимающихся контролем дорожного движения, а также для локальных ведомственных систем. Простота архитектуры и интерфейса программы позволяет масштабировать и дорабатывать ее для решения более сложных задач, таких как интеграция с национальными автоматизированными системами, создание мобильных приложений или добавление аналитических функций для оценки статистики нарушений.

Основной результат проекта – создание функционального программного обеспечения, способного повысить эффективность работы в области учета нарушений правил дорожного движения. Использование данной системы способствует упрощению процесса обработки данных, снижению трудозатрат и повышению точности фиксации нарушений. Будущие доработки могут включать интеграцию с внешними источниками данных, расширение аналитического функционала и использование современных технологий, таких как машинное обучение, для прогнозирования и предотвращения нарушений. Тем не менее, данный проект полностью соответствует целям и задачам курсовой работы, демонстрируя высокую практическую ценность и возможности применения современных информационных технологий в области обеспечения безопасности дорожного движения.

Выполнение данной работы позволило не только успешно реализовать поставленные задачи, но и приобрести ценный опыт, знания и компетенции в области разработки баз данных, интерфейсов и программирования. Навыки проектирования структур базы данных, создания интерфейса, построения запросов SQL, написание программного кода имеют важное значение для разработки приложений, ориентированных на работу с большими объемами данных и обработку информации, что делает этот опыт полезным для решения задач в различных областях. Полученные в настоящей работе данные результаты могут служить основой для дальнейшего профессионального роста и развития навыков в сфере информационных технологий и программирования, а разработанный программный модуль можно также рекомендовать для тестирования в области практического применения в соответствующих структурах с дальнейшей доработки и масштабирования.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

Нормативно - правовые источники

1. ГОСТ 7.32 - 2001. Международный стандарт. Система стандартов по информации, библиотечному и издательскому делу. Отчет о научно - исследовательской работе. Структура и правила оформления.

2. ГОСТ Р 7.0.5. - 2008. Национальный стандарт Российской Федерации. Система стандартов по информации, библиотечному и издательскому делу библиографическая ссылка.

3. ГОСТ 7.1 -2003. Система стандартов по информации, библиотечному и издательскому делу. Библиографическая запись. Библиографическое описание. Общие требования и правила составления.

4. ГОСТ 7.80 - 2000. Система стандартов по информации, библиотечному и издательскому делу. Библиографическая запись. Заголовок. Общие требования и правила составления.

Учебники, учебные пособия, статьи

1. Буйначев, С. К. Основы программирования на языке Python : учебное пособие / С. К. Буйначев, Н. Ю. Боклаг. – Екатеринбург : Изд-во Урал. ун-та, 2014. – 91 c.

2. Мамедли Р.Э. Системы управления базами данных: Учебное пособие. – Нижневартовск: Изд-во Нижневартовского государственного университета, 2021. – 214 с.

3. Прохоренок, Н. А. Python 3 и PyQt 5. Разработка приложений. — 2-е изд., перераб. и доп. /Н. А. Прохоренок, В. А. Дронов. — СПб.: БХВ-Петербург, 2018. — 832 с.

4. Федоров, Д. Ю. Программирование на языке высокого уровня Python : учеб.пособие для прикладного бакалавриата / Д. Ю. Федоров. — 2-е изд., перераб.и доп. — М. : Издательство Юрайт, 2019. — 161 с.

5. Швецов В.И., Визгунов А.Н., Мееров И.Б. Базы данных. Учебное пособие. Нижний Новгород: Изд-во ННГУ, 2004. - 217 с.

Онлайн - курсы

1. Добрый, добрый Python - обучающий курс от Сергея Балакирева

2. Инди - курс программирования на Python

3. Интерактивный тренажер по SQL

4. Python. Введение в программирование

Интернет - источники

1. SQLite: сайт. URL: <https://sqlite.org/> (30.11.2024)

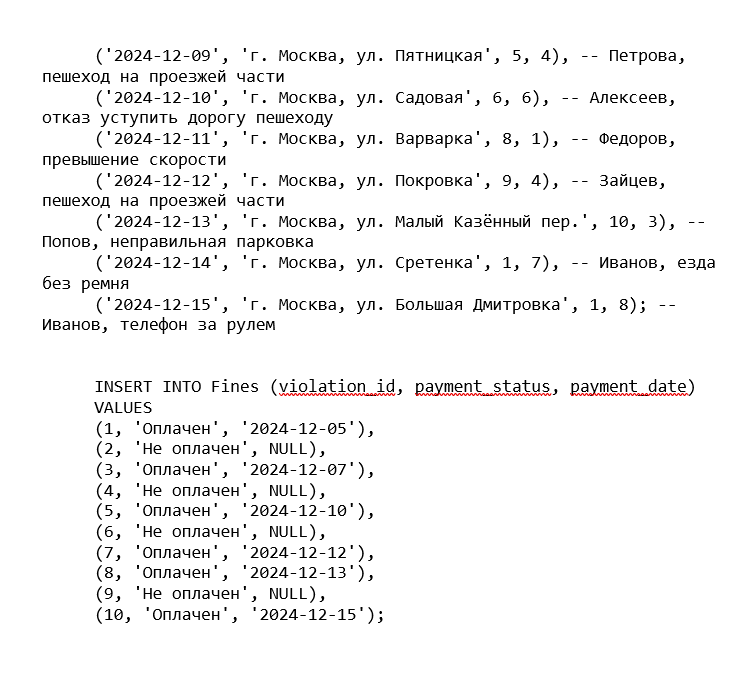
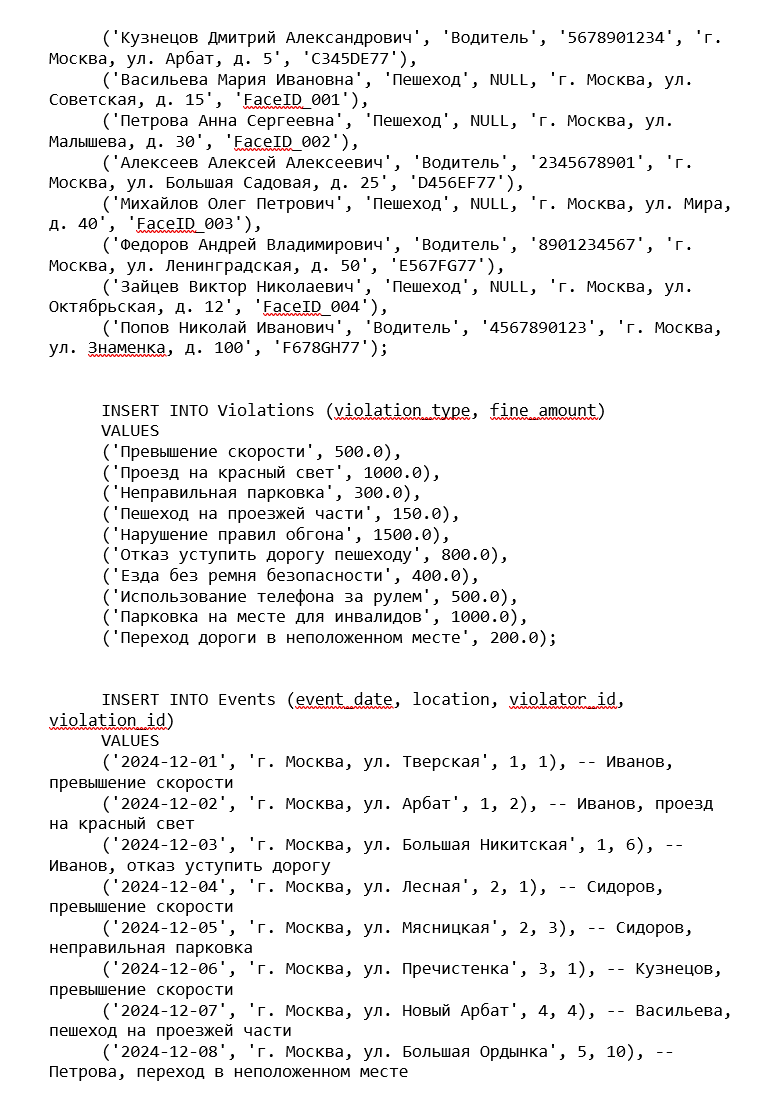
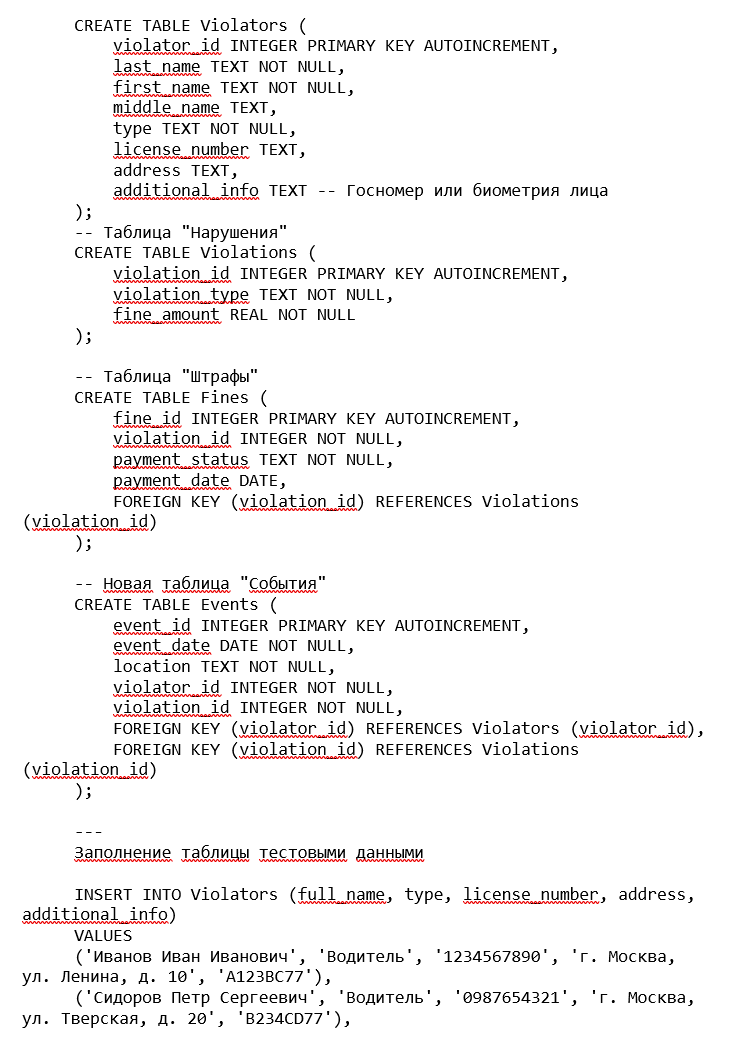
2. Habr.com: сайт.

URL: <https://habr.com/ru/companies/skillfactory/articles/599599/> (10.12.2024)

3. Роксис информационные системы: сайт. <https://www.roksis.ru/>

(17.12.2024)

ПРИЛОЖЕНИЕ 1



ПРИЛОЖЕНИЕ 2

