**КУРСОВОЙ ПРОЕКТ**

**Тема: разработка программного модуля информационной системы, предназначенной для эффективного управления парком автомобилей**

**Специальность 09.02.07 Информационные системы и программирование**

|  |  |  |
| --- | --- | --- |
| **Выполнил студент(ка) группы 31ИС-21** | **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_** | **С. В. Савченко** |
| **Руководитель** | **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_** | **В.Ю. Назаров** |

**Москва 2023**

**Федеральное государственное бюджетное образовательное учреждение**

**высшего образования**

**«РОССИЙСКАЯ АКАДЕМИЯ НАРОДНОГО ХОЗЯЙСТВА И ГОСУДАРСТВЕННОЙ СЛУЖБЫ ПРИ**

**ПРЕЗИДЕНТЕ РОССИЙСКОЙ ФЕДЕРАЦИИ»**

КОЛЛЕДЖ МНОГОУРОВНЕВОГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ

**Задание**

**на курсовой проект**

Дисциплина: МДК.01.01 Разработка программных модулей

Тема: Разработка программного модуля «Система управления парком автомобилей»

Специальность: 09.02.07 Информационные системы и программирование

Группа: 31ИС-21

ФИО студента Савченко С. В.

ФИО руководителя Назаров В.Ю.

1. Проанализировать предметную область
2. Проанализировать готовые решения
3. Подготовить техническое задание
4. Подготовить план тестирования
5. Обосновать выбор инструментов и средств разработки
6. Описать реализацию технического задания
7. Выполнить тестирование

Задание выдано «\_\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2023 г.

Срок выполнения «\_\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2023 г.

Сроки защиты \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Преподаватель: \_\_\_\_\_\_\_\_\_\_\_\_\_\_

Задание получил: \_\_\_\_\_\_\_\_\_\_\_\_\_\_

ОГЛАВЛЕНИЕ

[ВВЕДЕНИЕ 4](#_Toc153894750)

[1. АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ 7](#_Toc153894751)

[1.1 Описание предметной области 7](#_Toc153894752)

[1.2 Обзор и анализ существующих программных решений 8](#_Toc153894753)

[2. ПРОЕКТИРОВАНИЕ СИСТЕМЫ 13](#_Toc153894754)

[2.1 Техническое задание 13](#_Toc153894755)

[2.2 Общее назначение системы 17](#_Toc153894756)

[3. РЕАЛИЗАЦИЯ ПРОЕКТА СИСТЕМЫ 20](#_Toc153894757)

[3.1 Описание кодом основных функций модуля Authorization 20](#_Toc153894758)

[3.2 Описание кодом основных функций модуля ChooseWindow 21](#_Toc153894759)

[3.3 Описание основных функций модуля Admin 22](#_Toc153894760)

[3.3.1 Описание кодом основных методов класса AdminDatabase 22](#_Toc153894761)

[3.3.2 Описание кодом основных методов класса AdminExcel 23](#_Toc153894762)

[3.4 Описание кодом основных функций модуля Driver 26](#_Toc153894763)

[3.5 Тестирование приложения 28](#_Toc153894764)

[3.5.1 Тестирование окна авторизации 28](#_Toc153894765)

[3.5.2 Тестирование интерфейса админа 29](#_Toc153894766)

[3.5.3 Тестирование интерфейса пользователя 30](#_Toc153894767)

[ЗАКЛЮЧЕНИЕ 31](#_Toc153894768)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 32](#_Toc153894769)

[ПРИЛОЖЕНИЯ 34](#_Toc153894770)

[Приложение 1. Добавление записи в таблицу 34](#_Toc153894771)

[Приложение 2. Удаление записи из таблицы 35](#_Toc153894772)

[Приложение 3. Редактирование существующей записи в таблице 36](#_Toc153894773)

[Приложение 4. План тестирования 37](#_Toc153894774)

# ВВЕДЕНИЕ

Эффективное управление парком автомобилей представляет собой ключевой аспект для различных транспортных компаний и предприятий. В условиях динамичного развития транспортной отрасли и необходимости эффективного использования автопарка, вопросы автоматизации и оптимизации управления транспортными средствами становятся крайне важными.

Создание подобной системы обладает высокой актуальностью в современном бизнес-контексте, поскольку предприятия и организации сталкиваются с рядом вызовов, такими как необходимость проведения анализов и составления отчётностей, оптимизация ресурсов, улучшение качества обслуживания техники и многие другие, которые можно эффективно решать с использованием такой системы.

Цель данной курсовой работы - разработка программного модуля информационной системы, предназначенной для эффективного управления парком автомобилей. Такая система должна обеспечивать комплексное взаимодействие с транспортными средствами, оптимизировать процессы логистики, обеспечивать техническое обслуживание и контроль за безопасностью, с учетом современных стандартов и требований, возможность создания различных отчётов,

Актуальность проблемы создания модуля информационной системы для управления автопарком обусловлена не только ростом объемов автомобильных перевозок, но и стремительным развитием технологий, позволяющих сделать управление транспортными ресурсами более эффективным и прозрачным. Компании, занимающиеся автотранспортной деятельностью, сталкиваются с необходимостью современного подхода к управлению автопарком, который включает в себя автоматизацию, аналитику и оптимизацию процессов.

К задачам курсовой работы по созданию модуля информационной системы для управления парком автомобилей относятся:

Анализ требований и функциональности:

* Изучение потребностей и требований заказчика к информационной системе.
* Составление списка функциональных требований к программному модулю.
* Анализ существующих аналогов и определение основных возможностей.

Проектирование системы:

* Разработка структуры базы данных для хранения информации о парке автомобилей.
* Проектирование пользовательского интерфейса для удобного взаимодействия с системой.
* Определение архитектуры программного модуля.

Разработка основного функционала:

* Создание модуля для регистрации и учета автомобилей в парке.
* Разработка функционала для управления состоянием автомобилей (ремонт, техобслуживание).
* Внедрение механизмов мониторинга за расходом топлива и техническим состоянием.

Безопасность и доступ:

* Внедрение системы аутентификации и авторизации пользователей.
* Обеспечение защиты данных и обеспечение конфиденциальности информации.

Тестирование:

* Разработка тестовых случаев для проверки функционала.

Документирование:

* Написание технической документации по разработанному программному модулю.
* Подготовка руководства пользователя для использования системы.

Оптимизация и улучшение производительности:

* Идентификация узких мест в работе системы и их оптимизация.

В данной курсовой работе объектом исследования будет являться управление парком автомобилей, а предметом исследования будут такие аспекты, как добавление, редактирование и удаление данных из базы данных, составление отчётностей, на основе имеющихся данных.

# АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ

* 1. Описание предметной области

Транспортные компании осуществляют перевозки различных грузов и/или пассажиров, оперируя автопарками, водителями, и маршрутами.

Информационная система транспортной компании предназначена для эффективного управления операционной деятельностью, отслеживания движения автотранспорта, анализа данных и принятия решений.

Основные компоненты предметной области – это:

* Водители: Регистрация и учет водителей, их личной и профессиональной информации, включая данные о водительских правах, медицинских осмотрах и прочее.
* Автопарк: Учет транспортных средств, их техническое обслуживание, топливо и технические характеристики.
* Маршруты: Планирование, создание и управление маршрутами, определение географических точек и расписания движения.
* Грузы и Пассажиры: Регистрация и отслеживание грузов, пассажиров, услуг и их перемещение через систему.
* Операционная деятельность: Отслеживание текущих перевозок, статусов, контроль исполнения заказов и учет времени в пути.

Основные потребности бизнеса — это:

* Оптимизация маршрутов: Минимизация времени в пути и расходов на топливо.
* Планирование ресурсов: Равномерное распределение загрузки по водителям и автомобилям.
* Учет операций: Отслеживание движения транспорта, регистрация выполненных задач и загруженности автопарка.
* Безопасность и соблюдение законов: Ведение документации по водителям, автомобилям и выполненным маршрутам для соблюдения нормативов и требований.
  1. Обзор и анализ существующих программных решений

Обзор и анализ существующих программных решений в области информационных систем управления парком автомобилей транспортной компании выявляет разнообразие инструментов, предназначенных для повышения эффективности управления автопарком и оптимизации операционных процессов. Важно отметить, что выбор программного решения должен быть основан на конкретных потребностях и характеристиках транспортной компании. Стоит рассмотреть некоторые из существующих программных решений:

Fleet Complete:

Особенности: управление данными о водителях, машинах, рейсах и техническом обслуживании, генерация отчетов по эффективности водителей и техническому состоянию машин, предусматривает мобильную версию приложения.

Преимущества:

* Интегрированный подход к управлению всеми аспектами автопарка.
* Гибкая конфигурация и адаптация под различные типы автопарков.
* Эффективное управление техническим обслуживанием и ресурсами.

Недостатки:

* Возможно, избыточность функций, если не требуется GPS-отслеживание.

Далее представлен интерфейс Fleet Complete (Рисунок 1).

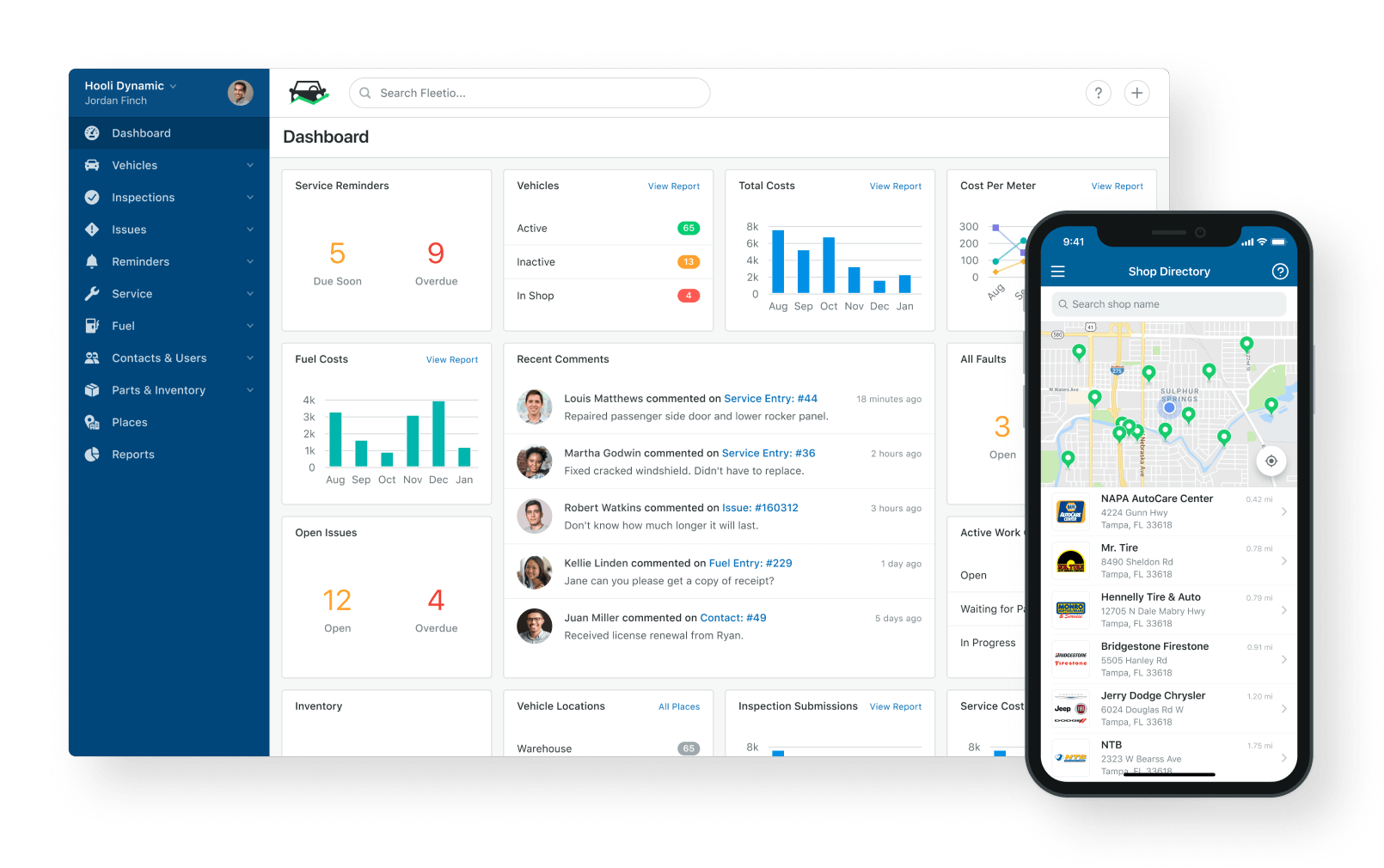


Рисунок 1 - Fleet Complete

Geotab:

Особенности: Geotab предоставляет комплексное решение для мониторинга и управления автопарком. Включает в себя GPS-отслеживание, датчики топлива, мониторинг вождения, отчеты о техническом состоянии и другие функции.

Преимущества:

* Обширные возможности аналитики, геозонирования, отслеживания топлива.

Недостатки:

* Высокие затраты на внедрение и обслуживание.

Далее представлен интерфейс Geotab (Рисунок 2).

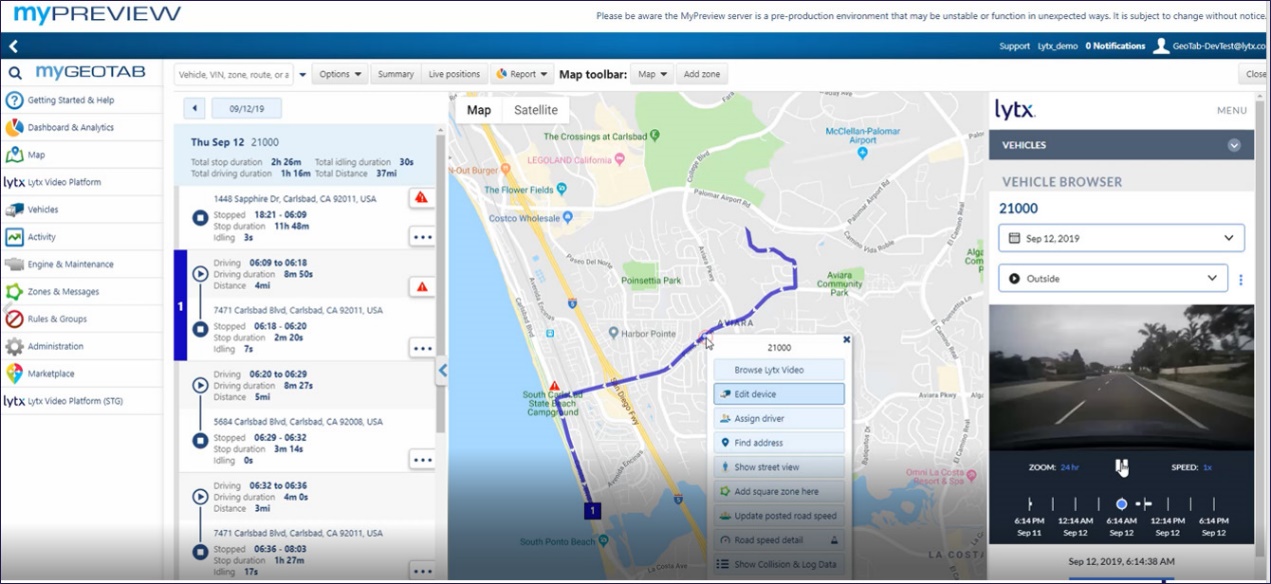


Рисунок 2 - Geotab

Verizon Connect:

Особенности: предоставляет решение для маршрутизации, мониторинга топливопотребления и анализа водительского стиля, предусматривает мобильную версию приложения.

Преимущества:

* Гибкие настройки.
* Многофункциональность.
* Возможности интеграции.

Недостатки:

* Высокие расходы на внедрение и обслуживание.

Далее представлен интерфейс Verizon Connect (Рисунок 3).



Рисунок 3 - Verizon Connect

Open-source решения:

Существуют также open-source решения, такие как Traccar, которые предоставляют базовые функции мониторинга автопарка и могут быть адаптированы под конкретные потребности компании.

Преимущества:

* Бесплатность.
* Возможность настройки под конкретные требования.

Недостатки:

* Ограниченные функциональные возможности по сравнению с коммерческими решениями.

Далее представлен интерфейс Traccar (Рисунок 4).

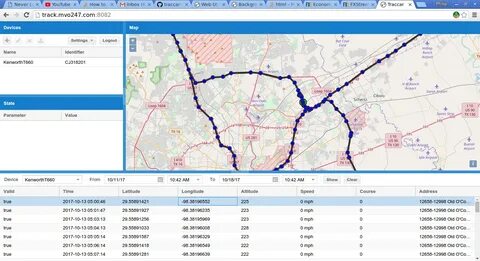


Рисунок 4 - Traccar

На основании анализа решений вывод о том, что включает разрабатываемый модуль.

Программный модуль для информационной системы для управления паком автомобилей должен включать в себя возможность управления данными о водителях, машинах, рейсах и техническом обслуживании, возможность генерации отчетов о проделанных маршрутах и техническому состоянию машин, медицинскому состоянию водителей.

# ПРОЕКТИРОВАНИЕ СИСТЕМЫ

* 1. Техническое задание

Целью разработки модуля информационной системы «Управление парком автомобилей» является создание графического приложения для управления данными в базе данных, а также самой базы данных для данного модуля информационной системы. Программа предоставляет пользователю удобный интерфейс для добавления, удаления и просмотра данных в таблицах базы данных.

Перечень требований к функциям, выполняемым системой – это:

Авторизация пользователя, к ней относятся:

* Создание 2 уровней допуска пользователя:
  + «admin» - имеет права на добавление, удаление и редактирование любых данных, создавать отчёты на основании этих данных;
  + «driver» – может просматривать свои личные данные и информацию о своих выездах;
* Возможность ввода логина пользователя при запуске приложения.
* Проверка учетных данных для доступа к приложению.
* В случае успешной авторизации, в зависимости от пользователя, открывается главное окно приложения для конкретного пользователя.

Главное окно:

* Пользователь «admin»:
  + Пользовательский интерфейс с кнопками "Добавить данные", "Удалить данные" и "Редактировать данные":
    - Окно добавления данных:
      * Форма для ввода данных согласно полям таблицы базы данных.
      * Кнопка "Добавить" для выполнения операции вставки данных в таблицу.
      * Кнопка "Отмена" для закрытия окна без добавления данных.
    - Окно удаления данных:
      * Форма для выбора записи для удаления по полю ID, уникальному для каждой таблицы
      * Кнопка "Удалить" для выполнения операции удаления данных из таблицы.
      * Кнопка "Отмена" для закрытия окна без удаления данных.
    - Окно редактирования данных:
      * Форма для редактирования данных согласно полям таблицы базы данных.
      * Кнопка "Подтвердить" для подтверждения операции редактирования данных.
      * Кнопка "Отмена" для закрытия окна без редактирования данных.
  + Таблица для отображения данных из выбранной таблицы базы данных.
    - Выбор таблицы для вывода данных
    - Кнопка «Вывести данные» для подтверждения вывода

Далее приведена диаграмма вариантов использования для требуемого программного модуля (Рисунок 5).

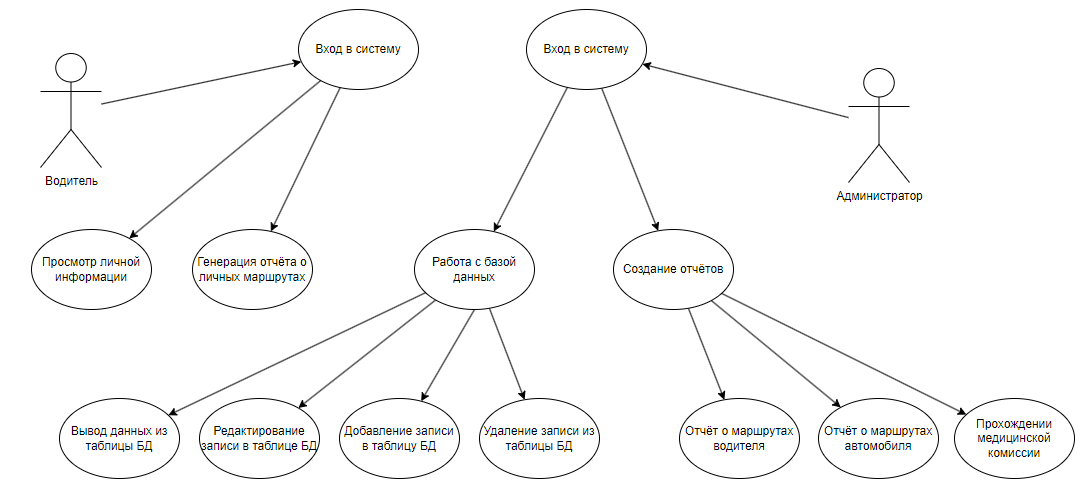


Рисунок 5 - Диаграмма вариантов использования

Интеграция с базой данных:

* Подключение к базе данных MySQL.
* Выполнение запросов SELECT, INSERT, DELETE и так далее для работы с данными.

Дополнительные функциональные требования (по желанию):

* Автоматическая подгонка размеров элементов интерфейса под размер окна.

Технические требования:

* Использование языка программирования Python.
* Использование библиотеки PyQt6 для создания графического интерфейса.
* Подключение к базе данных MySQL для выполнения запросов.
* Обработка ошибок при выполнении запросов к базе данных.
* Валидация вводимых данных в формах добавления и удаления.

Требования к базе данных:

* В базе данных программа создать следующие таблицы:
* Auto – таблица для хранения данных об автомобилях
* Driver – таблица для хранения данных о водителях
* Routs – таблица для хранения данных о маршрутах автотранспорта и водителей
* Technical service – таблица для хранения данных о техническом обслуживании автотранспорта

Далее приведена ER-диаграмма для требуемой базы данных (Рисунок 6).

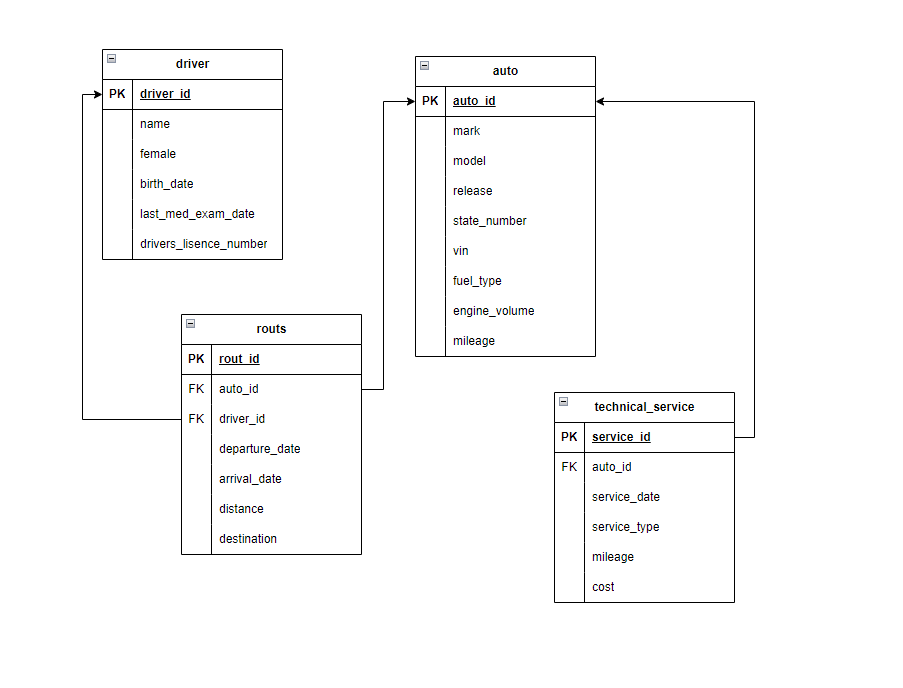


Рисунок 6 - ER-диаграмма

Далее приведена диаграмма классов для разрабатываемого программного модуля (Рисунок 7).

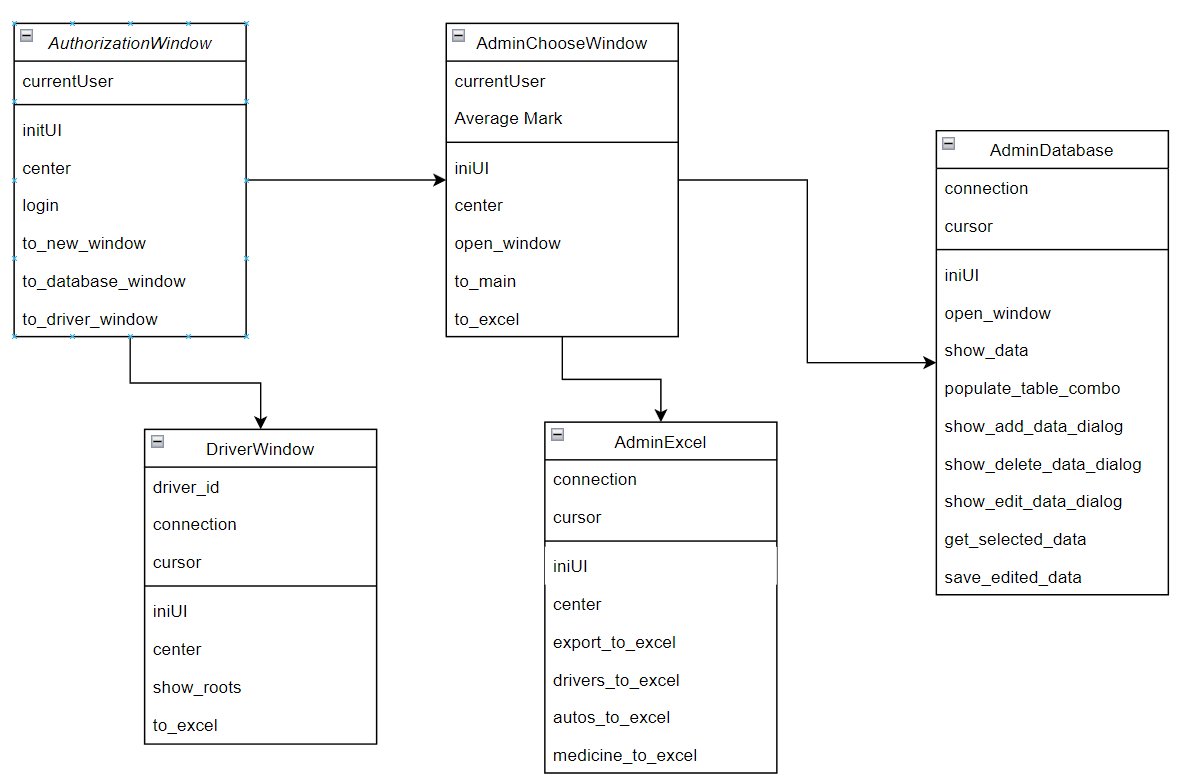


Рисунок 7 - Диаграмма классов

* 1. Общее назначение системы

Основной назначением разрабатываемого модуля является обеспечение комплексного контроля и анализа операций автопарка, повышение эффективности использования ресурсов и обеспечение безопасности и удобства для пользователей.

Ключевые функции системы:

Учет водителей:

* Добавление, редактирование и удаление информации о водителях.
* Возможность просмотра статистики и маршрутов каждого водителя.

Управление автомобилями:

* Регистрация и поддержка данных о каждом автомобиле в парке.
* Возможность добавления новых автомобилей и отслеживание их состояния.

Маршрутное планирование:

* Просмотр, добавление и редактирование данных о маршрутах.
* Мониторинг прохождения маршрутов, контроль расстояния.

Формирование отчетов:

* Создание отчетов по различным критериям, таким как пробег, расход топлива и статистика по водителям.
* Анализ данных для принятия решений и оптимизации работы автопарка.

Экспорт в Excel:

* Возможность экспорта данных о маршрутах и отчетов в формат Excel для удобного анализа и предоставления отчетности.

Система безопасности:

* Реализация механизма аутентификации и авторизации для защиты от несанкционированного доступа к данным.
* Обеспечение безопасной обработки и хранения конфиденциальной информации.

Удобный интерфейс:

* Создание интуитивно понятного и стильного пользовательского интерфейса для комфортного взаимодействия с системой.
* Легкое переключение между различными модулями и функционалом.

Ожидаемые результаты:

* Оптимизация управления ресурсами:
* Повышение эффективности использования автопарка.
* Уменьшение времени на планирование и мониторинг маршрутов.

Улучшение безопасности и контроля:

* Обеспечение надежной системы учета и контроля за водителями и автомобилями.
* Своевременное выявление и реагирование на нежелательные события.

Автоматизация процессов:

* Сокращение ручной работы при ведении данных и формировании отчетов.
* Улучшение точности и достоверности данных.

Удовлетворение потребностей пользователей:

* Обеспечение удобного и понятного интерфейса для различных категорий пользователей.
* Внедрение новых функций и улучшений в ответ на потребности и обратную связь пользователей.

# РЕАЛИЗАЦИЯ ПРОЕКТА СИСТЕМЫ

* 1. Описание кодом основных функций модуля Authorization

Модуль MainWindow предназначен для авторизации пользователя в системе. Данный модуль содержит ряд функций для выполнения этой задачи (Рисунок 8).

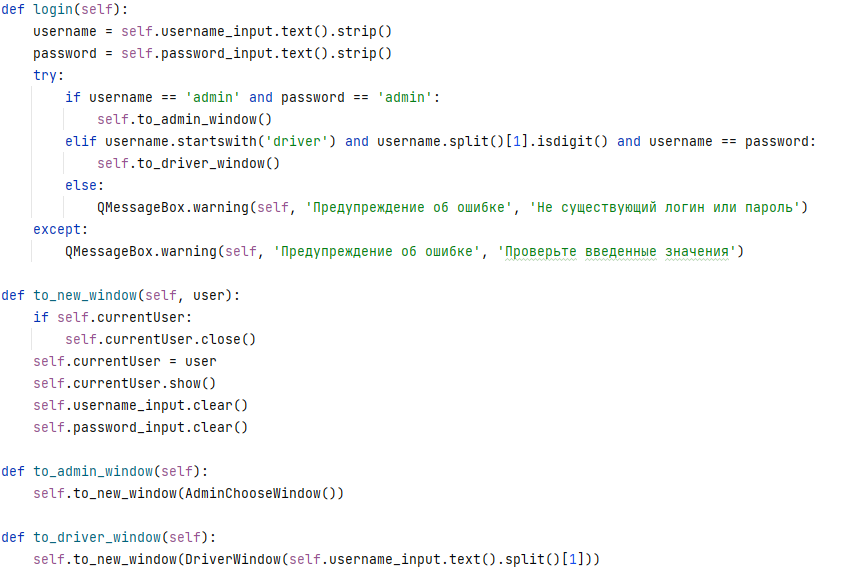


Рисунок 8 – Код функций для авторизации пользователя

Функция login организует принятие на вход логина и пароля, проверку их на корректность, при вводе некорректных или несуществующих данных пользователь получает уведомление о неправильных данных. В случае подтверждения данных, запустится одна из функций: to\_admin\_window или to\_driver\_window, в зависимости от введенного логина и пароля.

Переменная «currentUser», используемая в функции to\_new\_window служит для передачи ссылки на класс, который будет инициализирован в случае открытия того или иного окна.

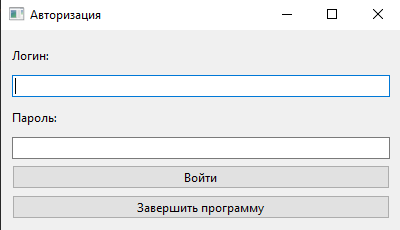


Рисунок 9 – Окно авторизации

* 1. Описание кодом основных функций модуля ChooseWindow

Модуль ChooseWindow предназначен для определения дальнейшего направления работы администратора: работа с базой данных или создание отчётов. Далее приведены функции, обеспечивающие выполнение требуемых задач (Рисунок 10).

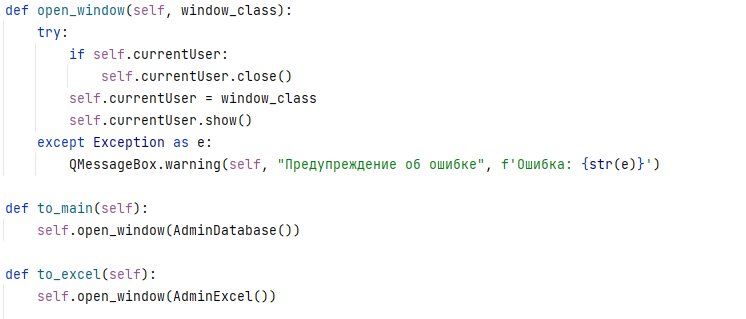
****

Рисунок 10 – Код функций окна для работы администратора

В модуле ChooseWindow переменная «currentUser» используется для передачи ссылки на класс, инициализируемый при запуске нового модуля, зависящего от выбора администратора.

* 1. Описание основных функций модуля Admin

Данный модуль предназначен для работы администратора в системе. Он содержит два основных класса: AdminDatabase и AdminExcel.

* + 1. Описание кодом основных методов класса AdminDatabase

При инициализации класса происходит подключение к базе данных для дальнейшей работы с ней (Рисунок 11).

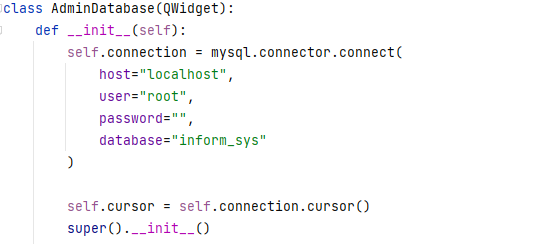


Рисунок 11 – AdminDatabase. Подключение к базе

Класс AdminDatabase содержит 4 основные функции для взаимодействия с базой данных:

* show\_data – предназначена для вывода информации из выбранной в специальном окне таблицы базы данных (Рисунок 12).
* show\_add\_data\_dialog – предназначена для добавления новой записи в выбранную таблицу базы данных (Приложение 1).
* show\_delete\_data\_dialog – предназначена для удаления записи из выбранной таблицы базы данных по ключевому полю «id» (Приложение 2). Полное название поля «id» индивидуально для каждой из таблиц базы данных.
* show\_edit\_data\_dialog – предназначена редактирования выделенной записи в выбранной таблице базы данных (Приложение 3).

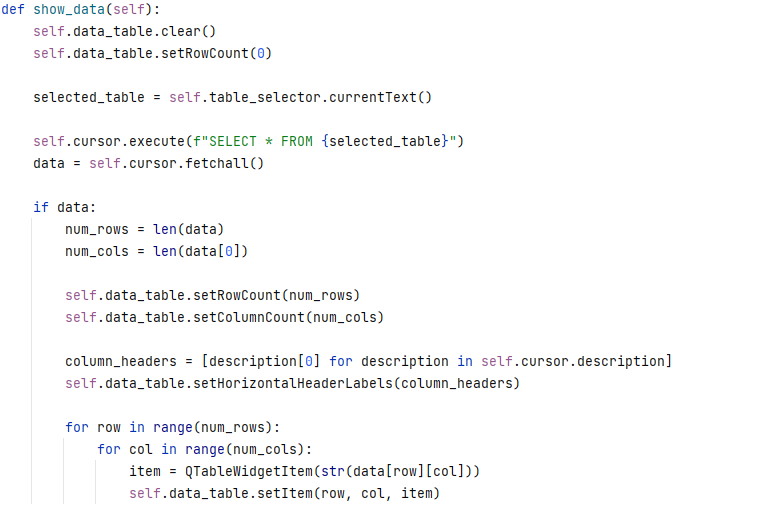


Рисунок 12 – Код функции show\_data

* + 1. Описание кодом основных методов класса AdminExcel

При инициализации класса происходит подключение к базе данных для дальнейшей работы с ней (Рисунок 13).

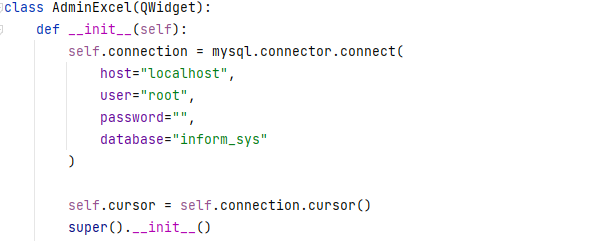


Рисунок 13 – AdminExcel. Подключение к базе

Класс AdminExcel содержит 4 основные функции формирования отчётов, основываясь на данных из таблиц базы данных:

* drivers\_to\_excel (Рисунок 14).



Рисунок 14 – Код функции drivers\_to\_excel

Функция предназначена выборки данных для составления отчёта о маршрутах одного конкретного водителя, а также для создания диалогового окна для выбора водителя для составления отчёта.

* autos\_to\_excel.

Код функции сильно идентичен коду функции drivers\_to\_excel (Рисунок 14). Меняются лишь заголовки таблиц, запрос в базу данных и название файла для сохранения.

Функция предназначена выборки данных для составления отчёта о маршрутах одного конкретного автомобиля, а также для создания диалогового окна для выбора авто для составления отчёта.

* medicine\_to\_excel (Рисунок 16).

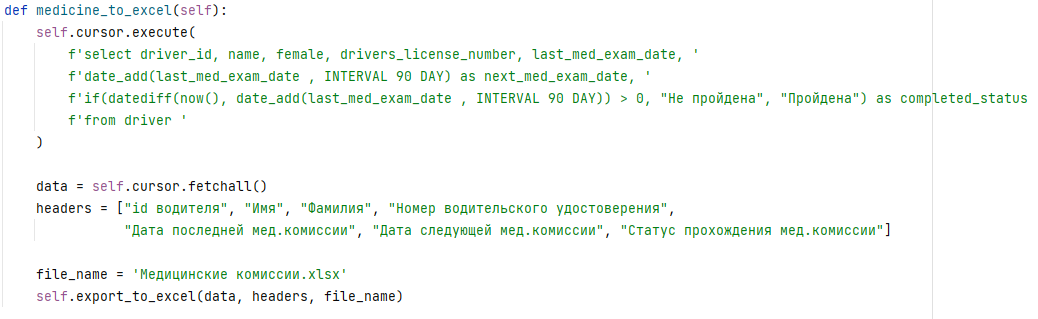


Рисунок 16 – Код функции medicine\_to\_excel

Функция предназначена для выборки данных для составления отчёта о прохождении мед. комиссии водителями транспортных средств, определения, была ли пройдена медкомиссия, установленная на запланированную дату.

* export\_to\_excel (Рисунок 17).



Рисунок 17 – Код функции export\_to\_excel

Функция является единым шаблоном для функций drivers\_to\_excel, autos\_to\_excel, medicine\_to\_excel, который принимает данные и экспортирует их в excel.

* 1. Описание кодом основных функций модуля Driver

Данный модуль предназначен для того, чтобы дать обычным пользователям, то есть водителям, возможность просматривать свои личные маршруты. При инициализации класса данного модуля производится подключение к базе данных (Рисунок 18). Так же функция \_\_init\_\_ принимает на вход параметр driver\_id, отмечающий за идентификацию пользователя в системе.

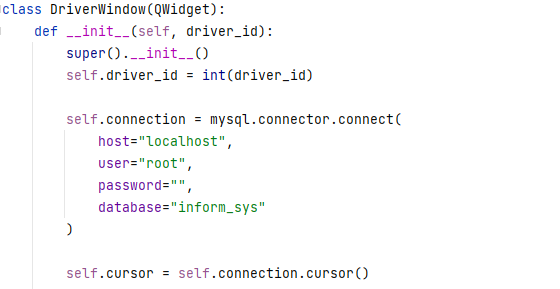


Рисунок 18 – DriverWindow. Подключение к базе

В классе DriverWindow существует функция для просмотра личной информации, которая автоматически выводится через диалоговое окно при входе пользователем в систему (Рисунок 19).

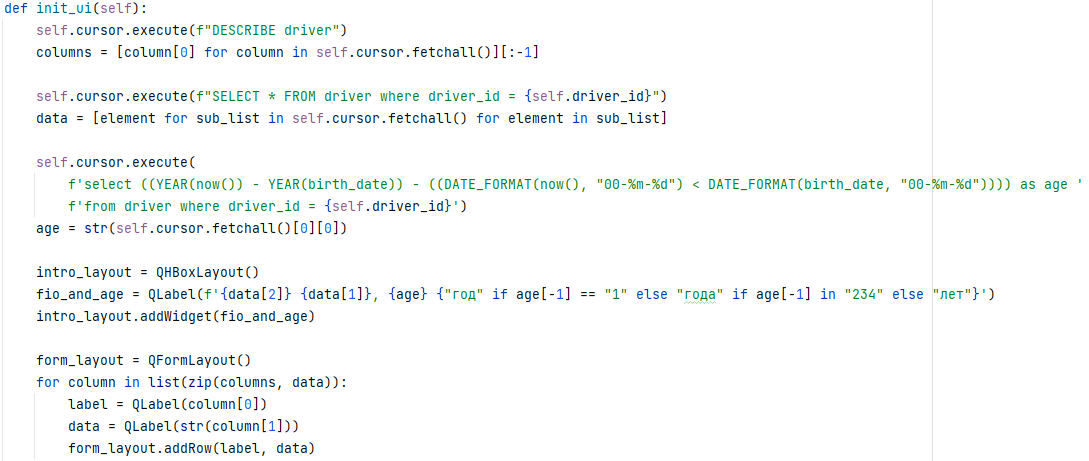


Рисунок 19 – Код функции init\_ui для вывода личной информации

Для просмотра пользователем своих рейсов существует функция show\_roots. Ниже представлена часть кода, отвечающая за вывод информации о личных маршрутах на экран (Рисунок 20).

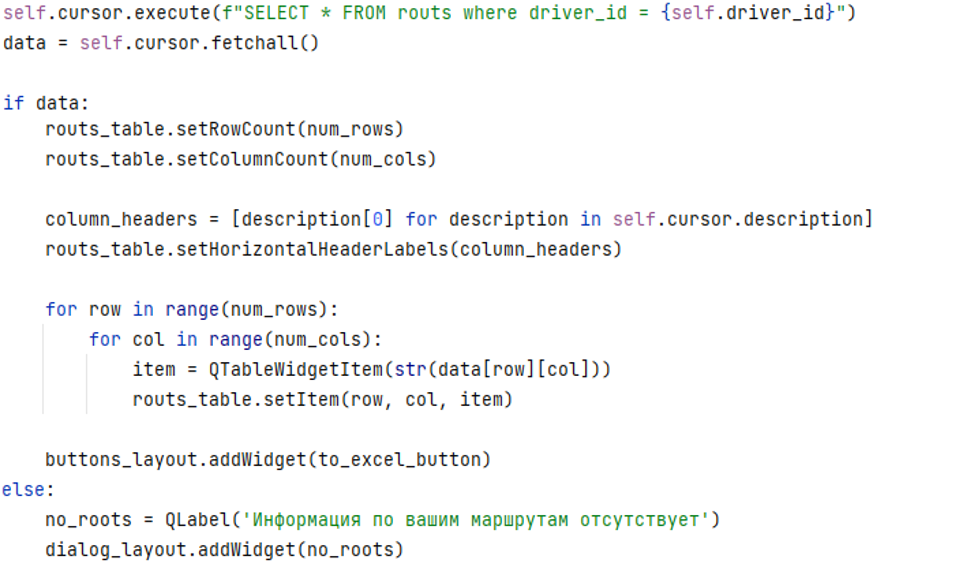


Рисунок 20 – Часть кода функции show\_roots

Для вывода информации о своих маршрутах в excel существует функция to\_excel (Рисунок 21).



Рисунок 21 – Код функции to\_excel

* 1. Тестирование приложения

Протестируем программу по плану тестирования (Приложение 4).

* 1. Тестирование окна авторизации

При вводе неверных данных пользователь должен увидеть уведомление о некорректном вводе (Рисунок 22).

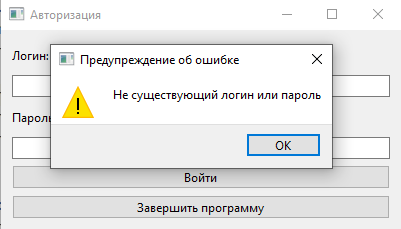


Рисунок 22 – Тестирование авторизации. Некорректные данные

Результатом является сообщение об ошибке и отчистка полей для ввода. При вводе верных данный администратора должно перенаправить на окно выбора действия (Рисунок 23).

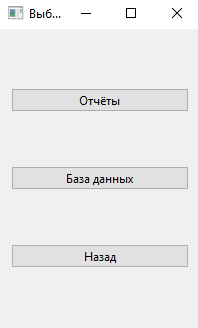


Рисунок 23 – Тестирование авторизации. Окно выбора действия

* 1. Тестирование интерфейса админа

В случае отсутствия сбоев, администратор сможет перейти на одно из двух окон (Рисунок 24) и (Рисунок 25).

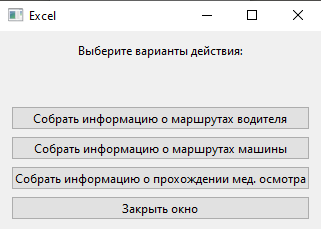


Рисунок 24 – Создание отчётов

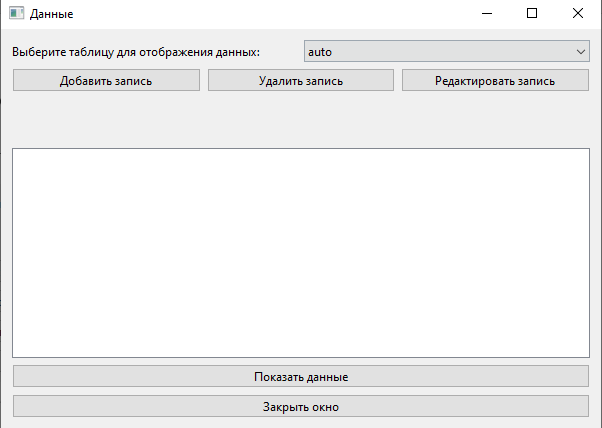


Рисунок 25 – Работа с базой данных

В случае введения верных данных пользователем, он попадет на окно с личной информацией и кнопкой, дающей возможность просмотреть свои маршруты (Рисунок 26).

* 1. Тестирование интерфейса пользователя

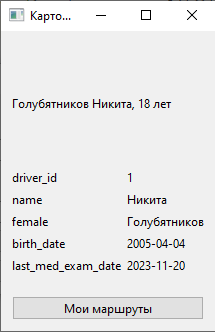


Рисунок 26 – Карточка водителя

При нажатии на кнопку «Мои маршруты» откроется окно с информацией о маршрутах пользователя (Рисунок 27).

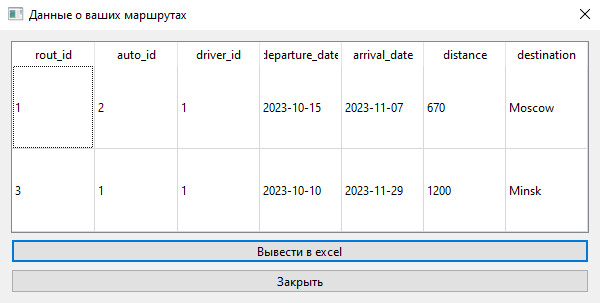


Рисунок 27 – Маршруты пользователя

При нажатии на кнопку «Вывести в excel» создастся excel файл с названием «Мои маршруты.xlsx», содержащий информацию с рисунка 27.

# ЗАКЛЮЧЕНИЕ

В рамках данной курсовой работы был создан программный модуль для управления парком автомобилей. В ходе разработки был осуществлен анализ требований и функциональности, связанных с управлением парком автомобилей, что позволило определить необходимые функции и особенности модуля.

Были разработаны и реализованы несколько ключевых функций, включая управление информацией об автомобилях, их состоянии, пробеге и техническом обслуживании. Модуль также предоставляет возможность просмотра статистических данных для качественного анализа работы парка автомобилей.

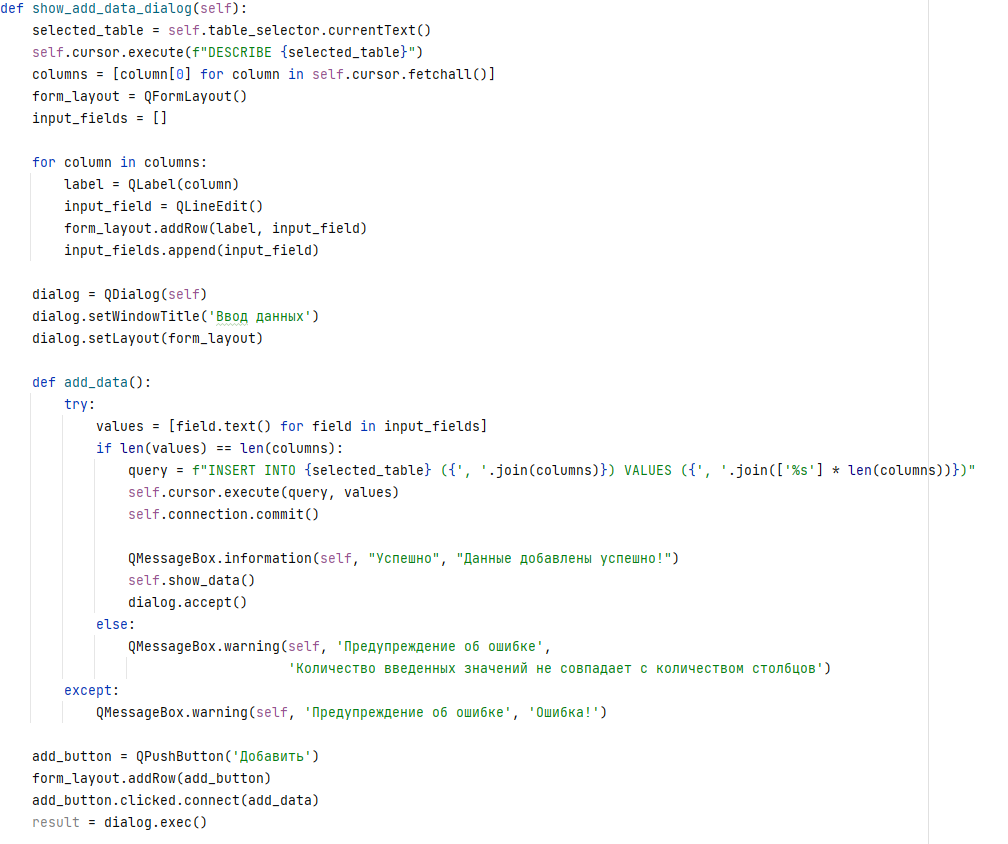
Модуль отличается интуитивно понятным интерфейсом и простотой использования, что позволяет сотрудникам эффективно работать с ним и улучшить общую производительность бизнеса. Он обеспечивает удобство, эффективность и надежность при работе с данными, а также способствует оптимизации бизнес-процессов, связанных с управлением автопарком. Он может быть использован в различных сферах, таких как логистика, транспортные компании и прокат автомобилей.

# СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Калитин С.В. Инструментальные средства информационных систем: учебное пособие / С.В. Калитин. – Москва: Издательство Солон-Пресс, 2021. – 104 с.
2. Хританков А. С., Полежаев В. А. Проектирование на UML. Сборник задач – Москва: Ridero, 2020. 214 c.
3. Голицына, О.Л. Информационные системы: Учебное пособие / О.Л. Голицына, Н.В. Максимов, И.И. Попов. - М.: Форум, 2016. - 352 c.
4. Панов А. И. Сетевое программирование на языке Python - М.: ДМК Пресс, 2017. - 246 с.
5. Томашевский В. А., Коновалов В. Г. Предметно-ориентированное прикладное программирование. - М.: Горячая Линия – Телеком, 2017. - 280 с.
6. Шалевич П. Загрузка данных и управление контентом в Интернете на PHP и MySQL. - М.: Трудовая Академия, 2018. - 384 с.
7. Оставько В. М., Карлунин А. И. Организация и технологии хранения данных в реляционных СУБД: Учебное пособие. - М.: БИНОМ. Лаборатория знаний, 2017. - 328 с.
8. Макконнелл С. Методы разработки программного обеспечения. - М.: КУДИЦ-Пресс, 2017. - 896 с.
9. Горишный, В. М. Основы разработки программного обеспечения: Учебное пособие. – Изд. 2-е, испр. и доп. – СПб.: БХВ-Петербург, 2018. – 400 с.
10. Имекаев Р. Задачи и упражнения по вычислительной математике: учебное пособие для вузов. – СПб.: БХВ-Петербург, 2019. – 352 с.
11. Нэфф Г. Самостоятельный разработчик: Осваиваем победный стиль программирования. – Диалектика, 2019. – 384 с.
12. Маршалл А., Коссет А. Как правильно использовать UML — М.: БИНОМ, Лаборатория знаний, 2017. 256 c.
13. Медведев И. В. Сравнение предметно-ориентированного подхода и объектно-ориентированного подхода в разработке программного обеспечения // Прикладная информатика. 2019. №2 (50). 218–223 с.
14. Муратов С. В., Агошков А. О. Организация интерфейсов с предметно-ориентированными языками // Прикладная информатика. 2019. №2 (50). 224
15. Кандидатов А. А. Онлайн-курс для обеспечения качества работы программных проектов // Прикладная информатика. 2019. №2 (50). 115–123 с.
16. Детков А. Е. Определение программных требований к интеллектуальной программной системе управления автотранспортом // Известия Саратовского университета. Новая серия. Серия Математика. Механика. Информатика. 2018. Т. 18. № 2. С. 193-198.
17. Краснов И. В., Муравьев С. В., Лысицин Ю. В. Проектирование систем распределенной обработки баз данных: монография. Волгоград: Изд-во ВолгГТУ, 2020. 95 с.
18. Румбахер Р., Эттингер У. Программирование на 99%. - М.: Драйзер-Медиа, 2017. - 600 с.
19. Егиазаров Ю. И., Хлебушкин А. А. Проектирование и реализация информационных систем на СУБД FoxPro: Учебное пособие. – СПб.: Политехника, 2008. – 256 с.
20. Каплун А. В., Ахиненко П. А. Основы проектирования инфокоммуникационных систем: учебное пособие. М.: ПИК, 2020. 161 с.

# ПРИЛОЖЕНИЯ

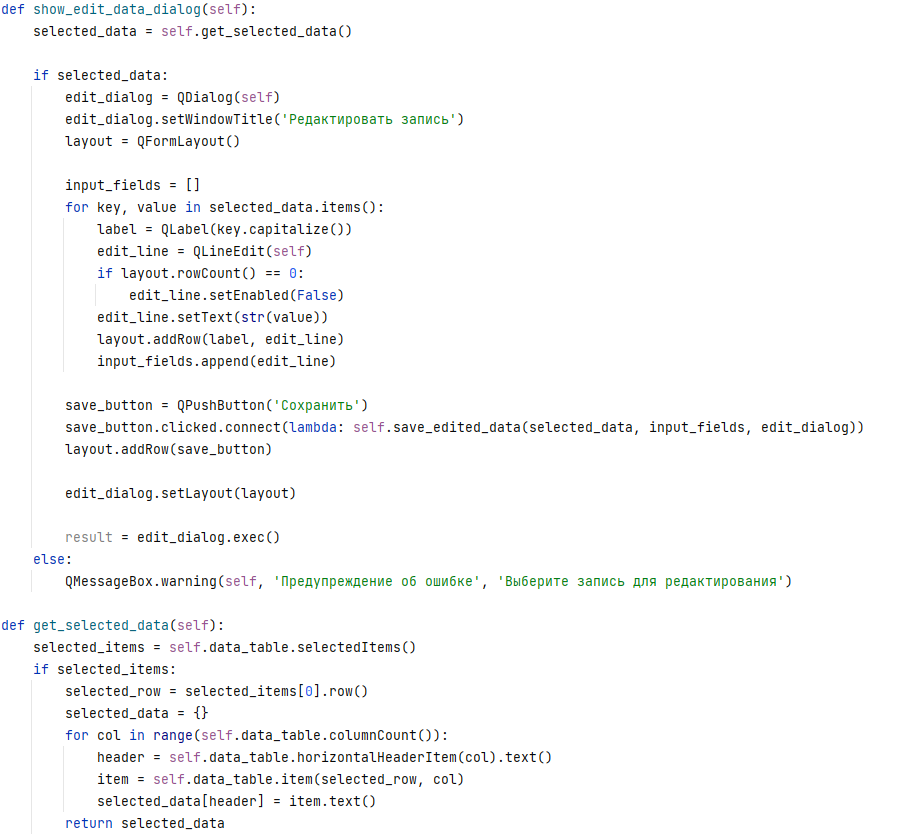
* 1. Добавление записи в таблицу

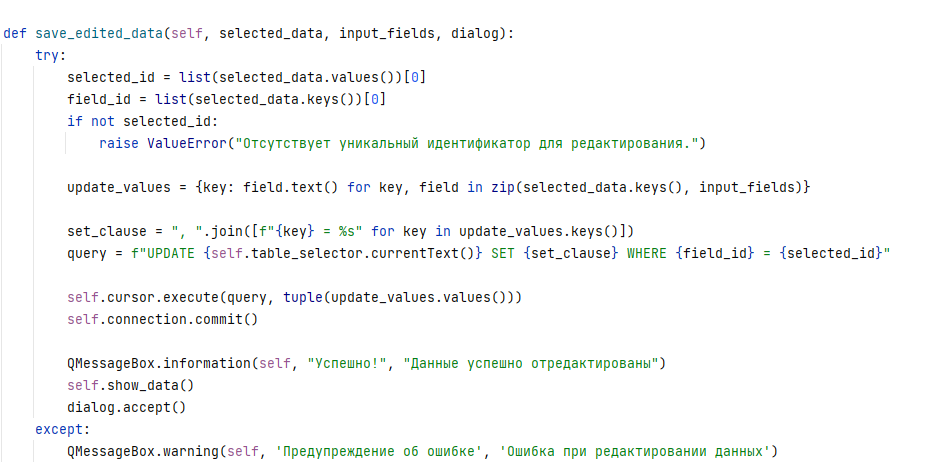


* 1. Удаление записи из таблицы



* 1. Редактирование существующей записи в таблице





* 1. План тестирования

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Наименование функциональности | Наименование поля | Тестовый набор | Результат тестирования |
| 1 | Авторизация | Логин и пароль | Ввод корректных данных | Вход в систему под введенными данными |
|  |  | Логин и пароль | Ввод неподходящих данных | Сообщение об ошибке: «Проверьте введенные значения  » |
| 2 | Добавление записи в базу данных | Ввод данных, в соответствии с наименованиями полей | Численное значение поля id | Сообщение: «Данные успешно добавлены» |
|  |  | Ввод данных, в соответствии с наименованиями полей | Ввод неподходящих данных | Сообщение об ошибке: «проверьте введенные данные» |
| 3 | Удаление записи из базы данных | Ключевое поле id, уникальное для каждой таблицы | Численное значение поля id | Сообщение: «Данные успешно удалены» |
|  |  | Ключевое поле id, уникальное для каждой таблицы | Ключевое поле id, не существующее в таблице | Сообщение об ошибке: «проверьте введенное значение поля id» |
| 4 | Редактирование существующей записи | Выделенная запись для редактирования | Ввод корректных данных, подходящих под наименование поля, кроме поля id(поле id редактировать нельзя) | Сообщение: «Данные успешно отредактированы» |
|  |  | Выделенная запись для редактирования | Не выделение ни одной из записей | Сообщение об ошибке: «проверьте введенное значение поля id» |