Проектирование системы (Техническое задание)

1. Общее назначение системы

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

1. Требования к функциям, выполняемым системой
   1. Авторизация пользователя:
      * Создать 3 уровня допуска пользователя:
        + «admin» - обладает неограниченными правами;
        + «engineer» - может просматривать все данные о транспортных средствах, их техническом состоянии;
        + «driver» + id водителя – 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.
   2. 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:
      * Возможность выбора записей для удаления.
      * Выбор записи для удаления производится по полю ID
      * 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
      * Пользователь «engineer»:
        + Таблица для отображения данных из ограниченного числа таблиц базы данных.
        + Выбор таблицы для вывода данных
        + Кнопка «Вывести данные» для подтверждения вывода
        + Пользовательский интерфейс с возможностью сбора информации для контроля за датами технического обслуживания у конкретных машин
      * Пользователь «driver»:
        + Пользовательский интерфейс c возможностью менять, добавлять, удалять или просматривать свои личные данные.
        + Пользовательский интерфейс c возможностью просмотра информации о своих выездах.
   3. Интеграция с базой данных:
      * Подключение к базе данных MySQL.
      * Выполнение запросов SELECT, INSERT, DELETE и так далее для работы с данными.
   4. Дополнительные функциональные требования (по желанию):
      * Автоматическая подгонка размеров элементов интерфейса под размер окна.
   5. 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.
2. Требования к базе данных:
   1. В базе данных программа создать следующие таблицы:
   2. Auto с полями: auto\_id, mark, model, release\_year, state\_number, VIN, fuel\_type, volume\_type, mileage.
   3. Driver с полями: driver\_id, name, female, birth\_date, last\_med\_exam\_date, driver’s\_license\_number.
   4. routs с полями rout\_id, auto\_id, driver\_id, departure\_date, arrival\_date, distance, destination.
   5. technical\_service; с полями: service\_id, auto\_id, service\_date, service\_type, mileage, cost.
   6. Связать таблицы базы данных по первичным ключам.