**При первом запуске приложение запрашивает данные для авторизации:**

**Фронтенд-разработчик:**

* Реализовать экран авторизации по макету. //добавить картинку макета в описание

**Бэкенд-разработчик:**

* Разработать API для аутентификации (проверка логина и пароля).
* Настроить связь с базой данных для хранения информации о пользователях.

**После авторизации от имени оператора программа отображает:**

**Фронтенд-разработчик:**

* Реализовать интерфейс для отображения информации о заправках на главном экране.
* Реализовать кнопку для добавления новой заправки.

**Бэкенд-разработчик:**

* Создать структуру базы данных для хранения информации о заправках.
* Настроить микросервис для передачи данных о заправках на фронтенд.

**Приложение должно показывать более подробные данные:**

**Дизайнер:**

* Разработать макет окна с подробной информацией о заправке (включая списки работников и ошибок).

**Фронтенд-разработчик:**

* Реализовать страницу для отображения подробной информации о заправке.
* Настроить возможность клика по заправке для открытия этой страницы.
* Реализовать механизм динамического отображения данных о заправке.

**Бэкенд-разработчик:**

* Реализовать API для получения списка работников и ошибок для конкретной заправки.
* Реализовать API для уведомления сотрудника (о чём? 25 -> 38 Суббота, 2 ноября).

**Программа должна предоставлять возможность отключения заправки:**

**Дизайнер:**

* Нарисовать кнопку «Отключить заправку» на экране заправки.

**Фронтенд-разработчик:**

* Реализовать функционал для отключения заправки через интерфейс приложения.

**Бэкенд-разработчик:**

* Настроить API для отключения заправки и изменения её статуса в базе данных.

**Программа должна предоставлять возможность ограничения потребления топлива заправкой:**

**Дизайнер:**

* Добавить интерфейс для настройки лимитов потребления топлива.

**Фронтенд-разработчик:**

* Реализовать функционал для установки и изменения лимитов.

**Бэкенд-разработчик:**

* Настроить API для сохранения ограничений на потребление топлива в базе данных.

**При введении неверных данных:**

**Дизайнер:**

* Разработать окно, которое будет показывать ошибку.

**Фронтенд-разработчик:**

* Реализовать механизм стирания данных при неверной аутентификации.

**Бэкенд-разработчик:**

* Настроить обработку ошибок при неверном вводе логина или пароля на сервере.

**Изменение цвета границы окна:**

**Фронтенд-разработчик:**

* Реализовать функционал изменения цвета границ окна в зависимости от состояния заправки.

**Приложение должно работать на автономном сервере:**

**DevOps-инженер:**

* Настроить архитектуру сервера (Трёхуровневая архитектура “Клиент-сервер”).

**Реализовать скроллинг заправок вниз:**

**Фронтенд-разработчик:**

* Реализовать скроллинг по списку заправок.

**Найти сервер:**

**Процессор: от 4 ядер**

**RAM: от 16 ГБ**

**Хранилище: от 1 ТБ**

**Сетевой интерфейс: от 1 Гбит/сек**

**ОС: Linux.**

**DevOps-инженер:**

* Обеспечить настройку серверной инфраструктуры в соответствии с минимальными требованиями (реализовать виртуализацию – рационально использовать вычислительные ресурсы; реализовать СУБД; реализовать эффективную передачу между клиентом и сервером – сервисы DNS/DHCP; …).

**Реализовать в безопасности:**

**DevOps-инженер:**

* Настроить резервирование серверных ресурсов (Дифференциальное резервное копирование. Хранение информации в головном офисе компании).
* Настроить систему обнаружения вторжений(DDoS-атак)

**Бэкенд-разработчик:**

* Реализовать шифрование данных с использованием шифрования и метода TDE(прозрачное шифрование данных)