

## Отчет №2

### 1. Проектирование базы данных

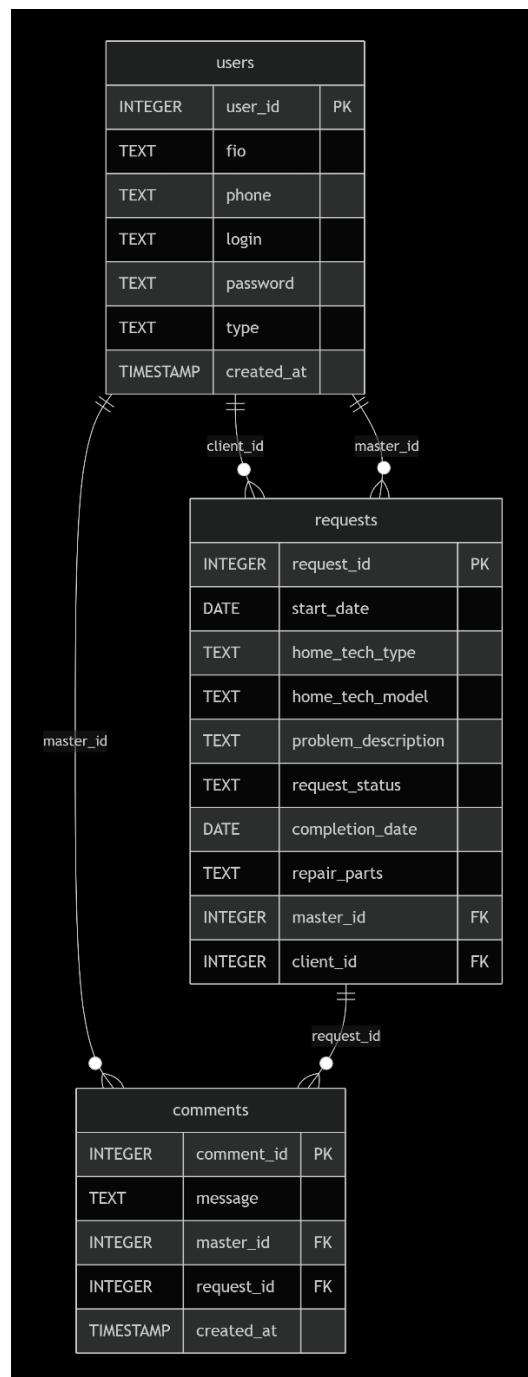
Разработана реляционная база данных в **SQLite** в соответствии с требованиями ЗНФ. База данных включает три основные сущности:

#### Схема базы данных (**SQLite**):

**Таблица users** - пользователи системы

**Таблица requests** - заявки на ремонт

**Таблица comments** - комментарии специалистов



Для SQLite реализован метод резервного копирования через файл «load\_data.py», запуск через «python load\_data.py» в консоли

## **2. Реализация ролей пользователей:**

Система реализует **5 основных ролей** с различными правами доступа:

### **Менеджер (полный доступ):**

- Просмотр, создание, редактирование всех заявок
- Назначение мастеров на заявки
- Управление пользователями
- Просмотр статистики
- Доступ ко всем функциям системы

### **Мастер (ограниченный доступ):**

- Просмотр назначенных заявок
- Изменение статуса заявок
- Добавление комментариев к заявкам
- Указание использованных запчастей
- Отметка о завершении работ

### **Оператор (работа с клиентами):**

- Создание новых заявок
- Редактирование информации о клиентах
- Просмотр статуса заявок
- Взаимодействие с клиентами

### **Заказчик (клиент системы):**

- Просмотр своих заявок
- Отслеживание статуса ремонта
- Получение уведомлений
- Доступ к QR-коду для оценки качества

### **Менеджер по качеству (новая роль):**

- Мониторинг сроков выполнения заявок
- Продление сроков с согласованием клиента

- Консультация мастеров при сложных случаях
- Анализ качества выполненных работ

### **3. Система аутентификации:**

- Простая проверка логина и пароля в базе данных
- Без использования JWT-токенов (статистическая архитектура)
- Сессионный подход с хранением состояния пользователя на клиенте

### **Защита паролей:**

- Пароли хранятся в открытом виде (для тестовой системы)
- В производственной версии рекомендуется использовать хеширование bcrypt