

Министерство науки и высшего образования Российской Федерации
федеральное государственное бюджетное образовательное учреждение высшего
образования
«Российский экономический университет имени Г.В. Плеханова»
Московский приборостроительный техникум

Дипломный проект (работа)

На тему: Разработка программного комплекса контроля аварийных
ситуаций устройств (на примере АО «Корпорация Комета»).

ИГОШЕВА РОСТИСЛАВА ВАДИМОВИЧА

Студент 4 курса группы П50-4-21

по специальности 09.02.07 «Информационные системы и программирование»
для присвоения квалификации: программист
Форма обучения: очная

Руководитель: _____ / Комаров Андрей Алексеевич /
(подпись)
«_____» _____ 2025 г.

Студент (-ка): _____ / Игошев Ростислав Вадимович /
(подпись)
«_____» _____ 2025 г.

Допущен к защите
Приказ от «_____» _____ 2025 г. № _____

2025

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	4
1. ОБЩАЯ ЧАСТЬ	5
1.1. Цель разработки	5
2. СПЕЦИАЛЬНАЯ ЧАСТЬ	7
2.1. Постановка задачи	7
2.1.1. Входные данные предметной области	7
2.1.2. Требования к проекту	8
2.2. Внешняя спецификация	10
2.2.1. Описание задачи	10
2.2.2. Входные и выходные данные	10
2.2.3. Методы	12
2.2.4. Тесты	17
2.2.5. Контроль целостности данных	17
2.3. Проектирование	19
2.3.1. Схема архитектуры приложения	19
2.3.2. Логическая схема данных	20
2.3.3. Физическая схема данных	21
2.3.4. Структурная схема	22
2.3.5. Функциональная схема	24
2.3.6. Диаграмма классов	25
2.3.7. Схема тестирования	25
2.3.8. Схема пользовательского интерфейса	32
2.4. Результат работы программы	35
3. ТЕХНОЛОГИЧЕСКАЯ ЧАСТЬ	37
3.1. Инструментальные средства	37
3.2. Отладка программы	38
3.3. Защитное программирование	38
3.4. Характеристики программы	39
ЗАКЛЮЧЕНИЕ	40
СПИСОК ИСПОЛЬЗУЕМЫХ МАТЕРИАЛОВ	42
ПРИЛОЖЕНИЕ А ТЕКСТ ПРОГРАММЫ	
ПРИЛОЖЕНИЕ Б СЦЕНАРИЙ И РЕЗУЛЬТАТЫ ТЕСТОВЫХ ИСПЫТАНИЙ	

ПРИЛОЖЕНИЕ В РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ

ПРИЛОЖЕНИЕ Г СКРИПТ БАЗЫ ДАННЫХ

ПРИЛОЖЕНИЕ Д БИЗНЕС ПРОЦЕССЫ

ВВЕДЕНИЕ

С развитием технологий и увеличением числа электронных устройств в промышленных и производственных процессах, возникла необходимость в эффективном отслеживании их показателей. Эти устройства часто используются для автоматизации и управления различными системами, и их надежная работа является критически важной для бесперебойного функционирования предприятий. Однако, несмотря на наличие различных инструментов и программ для мониторинга, многие организации сталкиваются с трудностями в централизованном управлении и анализе данных с таких устройств.

В связи с этим в данной дипломной работе разработан программный комплекс контроля аварийных ситуаций устройств (на примере АО «Корпорация Комета»). Данная система предоставляет функционал для сбора, анализа и визуализации данных с устройств, используемых на предприятиях. Она будет включать в себя удобный интерфейс для отслеживания устройств, возможность отображения устройств при выходе показателей за допустимые пределы, что позволит пользователям легко отслеживать состояние своих устройств и принимать своевременные меры для их оптимальной работы.

1. ОБЩАЯ ЧАСТЬ

1.1. Цель разработки

Целью разработки является обеспечение централизованного мониторинга и управления электронными устройствами на предприятии.

1.2. Средства разработки

Программные средства, используемые для проектирования, разработки и тестирования приложения представлены в таблице 1.

Таблица 1 – Программные средства

№	Тип средства	Название средства	Назначение
1	2	3	4
1	Операционные системы	Astra Linux 1.6	Организация взаимодействия программ и пользователя
2	Операционные системы	Windows 10	Организация взаимодействия программ и пользователя
3	Операционные системы	Android 13	Организация взаимодействия программ и пользователя
4	Среда разработки	Qt Creator 4.15	Среда запуска
5	Среда разработки	Android Studio 17.0.9	Среда запуска
6	Сервер базы данных	PostgreSQL 9.4	Хранение данных
7	Графическая оболочка	pgAdmin 3	Разработка базы данных и функционала работы с ней

В качестве средств вычислительной техники при разработке ПО использовался персональный компьютер, ноутбук и телефон. Характеристики представлены в Таблице 2.

Таблица 2 - Технические характеристики ПК при разработке ПО

№	Тип оборудования	Наименование оборудования
1	2	3
Для разработки		
1	Процессор	Intel Core Duo e7500
2	Оперативная память	2 GB
4	Тип видеокарты	Встроенная
5	Видеокарта	AMD Radeon Graphics
6	Конфигурация накопителей	SSD

№	Тип оборудования	Наименование оборудования
1	2	3
7	Общий объем накопителей	256 ГБ
8	Компьютерная мышь	Logitech M100R
9	Клавиатура	Любая рабочая клавиатура с английской и русской раскладкой
10	Операционная система	Astra Linux
Для разработки		
1	Процессор	AMD Ryzen 3 3250U
2	Оперативная память	16 GB
4	Тип видеокарты	Встроенная
5	Видеокарта	AMD Radeon Graphics
6	Конфигурация накопителей	SSD
7	Общий объем накопителей	256 ГБ
8	Компьютерная мышь	Qumo Techno
9	Клавиатура	Любая рабочая клавиатура с английской и русской раскладкой
10	Операционная система	Windows 10
Для тестирования		
1	Процессор	AMD Ryzen 3 3250U
2	Оперативная память	16 GB
4	Тип видеокарты	Встроенная
5	Видеокарта	AMD Radeon Graphics
6	Конфигурация накопителей	SSD
7	Общий объем накопителей	256 ГБ
8	Компьютерная мышь	Qumo Techno
9	Клавиатура	Любая рабочая клавиатура с английской и русской раскладкой
10	Операционная система	Windows 10

2. СПЕЦИАЛЬНАЯ ЧАСТЬ

2.1. Постановка задачи

Разработать программный комплекс «DeviceMonitor» который представляет из себя комплекс из двух Desktop-приложений и одного мобильного контроля аварийных ситуаций устройств.

2.1.1. Входные данные предметной области

Входными данными программного комплекса «DeviceMonitor» являются следующие данные:

- Добавление пользователя (Логин, пароль, роль);
- Добавление устройства (Название, напряжение, минимальное напряжение, максимальное напряжение, ток, максимальный ток, температура, максимальная температура, емкость, название, расположение, включено ли);
- Добавление расположения (Название);
- Фильтрация устройств (Минимальное значение напряжения, минимальное значение тока, минимальное значение);
- Сетевые данные для подключения к серверу (IP-адрес, порт)

Выходные данные предметной области

- Пользователи (Логин, роль, дата регистрации);
- Устройства (Название, напряжение, минимальное напряжение, максимальное напряжение, ток, максимальный ток, температура, максимальная температура, емкость, название, расположение, включено ли, дата получения данных);
- Расположения (Название);
- История аварийных случаев (Дата аварии, копия данных устройства в момент аварии: Название, напряжение, минимальное напряжение, максимальное напряжение, ток, максимальный ток,

температура, максимальная температура, емкость, название, расположение, включено ли, дата получения данных);

2.1.2. Требования к проекту

Требования к программному комплексу «DeviceMonitor» предоставлены ниже:

- Многопользовательская поддержка – данные о пользователях и их ролях должны храниться в БД;
- В программном комплексе должно быть две роли: Администратор и Пользователь;
- Администратор может пользоваться админ-панелью и не может зайти в клиентскую версию программы;
- Администратор может запустить сервер, авторизоваться в админ-панели, подключить счетчик, отслеживать устройства, просматривать аварийные ситуации, просматривать историю аварийных ситуаций, добавлять и изменять устройства, добавлять и изменять пользователей, добавлять и изменять расположения;
- Пользователь может пользоваться клиентской версией программы и не может зайти в админ-панель;
- Пользователь в клиентской версии программы может подключиться к серверу, авторизоваться, отслеживать устройства, просматривать историю аварийных ситуаций, фильтровать устройства по изменяемым физическим показателям (ток, напряжение, температура), просматривать динамический график показателей устройств, сохранять данные об устройствах в формате json и csv;
- Приложение должно иметь простой и интуитивный интерфейс;
- Интерфейс и цветовая палитра приложения не должны ярко контрастировать и напрягать зрение;

- Приложение должно предоставлять возможность добавлять и отслеживать устройства;
- Вход в админ-панель должен начинаться с ввода логина и пароля от аккаунта;
- Приложение должно позволять формировать историю аварийных случаев с показателями и датами аварий;
- Серверное Desktop-приложение должно запускать сервер, принимающий подключения на получение данных об устройствах и других данных от клиентских версий приложения и отправляющий ответы в формате json-файла по протоколу TCP;
- Все входящие запросы и информация о подключениях к серверу должны быть отображены на основном экране в виде объемной текстовой строки;
- Серверное приложение должно предоставлять возможность использовать дополнительный режима работы приложения, в котором сервер по TCP подключается к серверу-отслеживателю как клиент, к которому подключены счетчики, для извлечения из него данных и записи их в БД;
- Клиентское приложение должно предоставлять возможность подключения к серверу по IP-адресу и порту приложения, после чего предоставлять возможность авторизации и дальнейшей работы программы, получая данные с подключенного сервера.
- Пароль должен храниться в виде хэша формата sha256;
- База данных должна быть приведена в 3 нормальную форму.

2.2. Внешняя спецификация

2.2.1. Описание задачи

Основной задачей является разработка программного решения, которое будет упрощать процесс взаимодействия между арендодателями, арендаторами и администраторами в сфере аренды недвижимости.

На этапе анализа были определены ключевые роли пользователей, такие как администратор, модератор и пользователь (арендодатель и арендатор), а также процессы, связанные с созданием, редактированием и управлением объявлениями, поиском и фильтрацией объектов недвижимости, и взаимодействием между сторонами.

Диаграммы прецедентов и модели AS-IS и TO-BE описаны в приложении Д «Бизнес процессы».

2.2.2. Входные и выходные данные

На таблице 3 изображены входные данные программного комплекса контроля аварийных ситуаций устройств.

Таблица 3 – Входные данные

Имя	Ограничения	Обязательное	Структура, формат	Форма ввода
1	2	3	4	5
Входные данные				
Логин	Максимальная длина строки 50 символов	Да	Строка	Текстовое поле
Пароль	Максимальная длина строки 50 символов	Да	Строка	Текстовое поле
Название расположения	Максимальная длина строки 50 символов	Да	Строка	Текстовое поле
Название устройства	Максимальная длина строки 70 символов	Да	Строка	Текстовое поле
Напряжение		Да	Число	Текстовое поле
Минимальное напряжение		Да	Число	Текстовое поле
Максимальное напряжение		Да	Число	Текстовое поле
Ток		Да	Число	Текстовое поле
Максимальный ток		Да	Число	Текстовое поле
Емкость		Да	Число	Текстовое поле
Температура		Да	Число	Текстовое поле
Максимальная температура		Да	Число	Текстовое поле
IP-адрес		Да	Строка	Текстовое поле
Порт		Да	Число	Текстовое поле

На таблице 4 изображены выходные данные программного комплекса.

Таблица 4 – Выходные данные

Имя	Ограничения	Обязательное	Структура, формат	Форма ввода
1	2	3	4	5
Выходные данные				
Логин	Максимальная длина строки 50 символов	Да	Строка	Текстовое поле
Пароль	Максимальная длина строки 50 символов	Да	Строка	Текстовое поле
Название расположения	Максимальная длина строки 50 символов	Да	Строка	Текстовое поле
Дата регистрации		Да	Строка	Текстовое поле
Дата аварийной ситуации		Да	Строка	Текстовое поле
Дата добавления данных устройства		Да	Строка	Текстовое поле
Название устройства	Максимальная длина строки 70 символов	Да	Строка	Текстовое поле
Напряжение		Да	Число	Текстовое поле
Минимальное напряжение		Да	Число	Текстовое поле
Максимальное напряжение		Да	Число	Текстовое поле
Ток		Да	Число	Текстовое поле
Максимальный ток		Да	Число	Текстовое поле
Емкость		Да	Число	Текстовое поле
Температура		Да	Число	Текстовое поле
Максимальная температура		Да	Число	Текстовое поле

2.2.3. Методы

Возможность объектов различных классов реагировать на одни и те же запросы с использованием общего интерфейса. Это позволяет

использовать один и тот же метод для разных объектов, что упрощает кодирование и повышает его гибкость.

Использована инкапсуляция, которая позволяет защитить данные от нежелательного доступа и изменений, а также облегчить поддержку кода и его повторное использование с помощью скрытия определённых данных программы от других данных. Также, использовался рефакторинг и оптимизация.

На рисунке 12 представлен пример без оптимизации. На рисунке представлен пример кода без оптимизации. Данный код содержит множество проблем, которые делают его небезопасным и неоптимизированным.

В коде отсутствует проверка типов данных при обработке значений JSON-объекта. Это может привести к ошибкам, если значения не соответствуют ожидаемому типу, например, если ожидается строка, но передано число. Такие ошибки могут вызвать аварийное завершение программы или некорректное выполнение логики.

Также, код не проверяет наличие ключа "checksum" перед его использованием. Если этот ключ отсутствует в JSON-объекте, программа может столкнуться с исключением, что сделает её уязвимой к сбоям.

Кроме того, используется неэффективный способ чтения данных из потока. Вместо использования стандартных методов чтения, код выполняет лишние операции, что замедляет его выполнение и увеличивает нагрузку на систему. Это особенно критично при работе с большими объемами данных, где производительность имеет первостепенное значение.

Также отсутствует проверка корректности контрольной суммы, что делает передачу данных ненадежной. Без этой проверки невозможно

гарантировать целостность данных, что может привести к использованию поврежденных или поддельных данных.

Код содержит избыточные преобразования данных, такие как преобразование JSON-объекта обратно в строку, что является лишней операцией и замедляет выполнение программы. Эти преобразования не только увеличивают время выполнения, но и потребляют дополнительные ресурсы памяти, что недопустимо в высоконагруженных системах.

Наконец, код использует неэффективный способ обработки ошибок, что также замедляет его выполнение. Вместо оптимальных алгоритмов обработки исключений используются методы, которые увеличивают сложность операций и снижают общую производительность программы.

Все эти проблемы делают код непригодным для использования в реальных приложениях, где требуется надежность, безопасность и высокая производительность.

Оптимизированный код демонстрирует хорошие практики обработки данных и обеспечения безопасности. Он корректно проверяет типы значений в JSON-объекте, что предотвращает ошибки при несоответствии типов. Код также включает проверку наличия ключа "checksum" и выполняет проверку контрольной суммы, что гарантирует целостность данных. Использование эффективных методов чтения данных и обработки исключений обеспечивает высокую производительность. Этот код надежен, безопасен и оптимизирован для работы с JSON-данными, что делает его подходящим для использования в реальных приложениях.

```

public String sendDataString(String msg) {
    long startTime = System.currentTimeMillis();

    if (msg == null) {
        Log.e(TAG, msg: "Входящее сообщение равно null.");
        return "ERROR_Некорректное сообщение.";
    }
    String msgCopy = msg.toString();
    if (msgCopy.length() == 0) {
        Log.e(TAG, msg: "Входящее сообщение пустое.");
        return "ERROR_Некорректное сообщение.";
    }

    if (error) {
        Log.e(TAG, msg: "Ошибка подключения к серверу выделена.");
        return "ERROR_Не удалось подключиться к серверу.";
    }

    try {
        String logMsg = "Отправка данных: " + msgCopy + " в " + System.currentTimeMillis();
        Log.d(TAG, logMsg);

        out.println(msgCopy);
        out.flush();
        Log.d(TAG, msg: "Данные успешно отправлены: " + msgCopy);
        Log.d(TAG, msg: "Ожидание ответа от сервера.");

        String jsonData = "";
        String line;
        try {
            while ((line = in.readLine()) != null) {
                if (line.isEmpty()) {
                    break;
                }
                jsonData += line;
                Log.d(TAG, msg: "Получена строка: " + line + " в " + System.currentTimeMillis());
                if (line.equals("}")) {
                    break;
                }
            }
        } catch (IOException e) {
            IOException logException = new IOException(e.getMessage());
            Log.e(TAG, msg: "Ошибка чтения данных: " + logException.toString());
            return "ERROR_Ошибка чтения данных.";
        }

        Log.d(TAG, msg: "Начало обработки JSON: " + jsonData);

        if (jsonData.length() == 0) {
            Log.e(TAG, msg: "Ответ от сервера пустой.");
            return "ERROR_Пустой ответ от сервера.";
        }

        JSONObject tempJson = new JSONObject(jsonData);
        String jsonString = tempJson.toString();
        JSONObject jsonResponse = new JSONObject(jsonString);

        String receivedChecksum = jsonResponse.getString("checksum");
        jsonResponse.remove("checksum");
        String dataStr = jsonResponse.getString("data");
        String calculatedChecksum = calculateChecksum(dataStr);

        String checksumComparison = receivedChecksum.equals(calculatedChecksum) ? "Совпадает" : "Не совпадает";
        Log.d(TAG, msg: "Контрольная сумма: " + checksumComparison);

        if (!receivedChecksum.equals(calculatedChecksum)) {
            Log.e(TAG, msg: "Обнаружено несоответствие контрольных сумм.");
            return "ERROR_Несоответствие контрольных сумм.";
        } else if (jsonResponse.getString("type").equals("error")) {
            Log.d(TAG, msg: "Получен тип ответа: error");
            return "ERROR_" + jsonResponse.getString("data");
        } else {
            Log.d(TAG, msg: "Получен тип ответа: data");
            return "DATA_" + jsonResponse.getString("data");
        }
    } catch (JSONException e) {
        String errorMsg = e.toString();
        Log.e(TAG, msg: "Ошибка обработки JSON: " + errorMsg);
        return "ERROR_Ошибка JSON.";
    } catch (Exception e) {
        String errorDetails = e.getMessage() != null ? e.getMessage() : "Неизвестная ошибка";
        Log.e(TAG, msg: "Неожиданная ошибка: " + errorDetails);
        return "ERROR_Неожиданная ошибка.";
    } finally {
        long endTime = System.currentTimeMillis();
        String timeLog = "Время выполнения: " + (endTime - startTime) + " мс";
        Log.d(TAG, timeLog);
    }
}

```

Рисунок 1 – Неоптимизированный код

На рисунке 13 представлен оптимизированный код функционала.

```

public String sendDataString(String msg) {
    long startTime = System.currentTimeMillis(); // Начало замера времени

    if (error) {
        return "ERROR_Не удалось подключиться к серверу.";
    }

    try {
        // Отправка данных
        out.println(msg);
        out.flush();
        Log.d(TAG, msg: "Данные отправлены: " + msg);
        Log.d(TAG, msg: "Ожидание завершено.");

        // Чтение ответа
        StringBuilder jsonBuilder = new StringBuilder();
        String line;

        try {
            while ((line = in.readLine()) != null && !line.isEmpty()) {
                jsonBuilder.append(line);
                Log.d(TAG, msg: "Полученные данные: " + line);
                if (line.equals("{}")) break;
            }
        } catch (IOException e) {
            Log.e(TAG, msg: "Ошибка чтения данных: " + e.getMessage());
            return "ERROR_Ошибка чтения данных.";
        }

        Log.d(TAG, msg: "Приступил к JSON");
        String jsonData = jsonBuilder.toString();
        Log.d(TAG, msg: "Полученный JSON: " + jsonData);

        if (jsonData.isEmpty()) {
            Log.e(TAG, msg: "Пустой ответ от сервера.");
            return "ERROR_Пустой ответ от сервера.";
        }

        JSONObject jsonResponse = new JSONObject(jsonData);

        // Проверка контрольной суммы
        String receivedChecksum = jsonResponse.getString( name: "checksum");
        jsonResponse.remove( name: "checksum");
        String dataStr = jsonResponse.getString( name: "data");
        String calculatedChecksum = calculateChecksum(dataStr);

        if (!receivedChecksum.equals(calculatedChecksum)) {
            Log.e(TAG, msg: "Несовпадение контрольных сумм.");

            // Проверка контрольной суммы
            String receivedChecksum = jsonResponse.getString( name: "checksum");
            jsonResponse.remove( name: "checksum");
            String dataStr = jsonResponse.getString( name: "data");
            String calculatedChecksum = calculateChecksum(dataStr);

            if (!receivedChecksum.equals(calculatedChecksum)) {
                Log.e(TAG, msg: "Несовпадение контрольных сумм.");
                return "ERROR_Несовпадение контрольных сумм.";
            } else if (jsonResponse.getString( name: "type").equals("error")) {
                return "ERROR_" + jsonResponse.getString( name: "data");
            } else {
                return "DATA_" + jsonResponse.getString( name: "data");
            }
        }
    } catch (JSONException e) {
        Log.e(TAG, msg: "Ошибка JSON: " + e.getMessage());
        return "ERROR_Ошибка JSON.";
    } catch (Exception e) {
        Log.e(TAG, msg: "Неожиданная ошибка: " + e.getMessage());
        return "ERROR_Неожиданная ошибка.";
    } finally {
        long endTime = System.currentTimeMillis(); // Конец замера времени
        Log.d(TAG, msg: "Время выполнения функции: " + (endTime - startTime) + " мс");
    }
}

```

Рисунок 2 – Оптимизированный код

На рисунке 14 представлено время исполнения неоптимизированного кода.

TcpClient	com.example.diplom_main	D	Время выполнения функции: 7 мс
TcpClient	com.example.diplom_main	D	Время выполнения функции: 116 мс
TcpClient	com.example.diplom_main	D	Время выполнения функции: 134 мс
TcpClient	com.example.diplom_main	D	Время выполнения функции: 127 мс

Рисунок 3 – Время исполнения неоптимизированного кода

На рисунке 15 представлено время исполнения оптимизированного кода:

TcpClient	com.example.diplom_main	D	Время выполнения функции: 15 мс
TcpClient	com.example.diplom_main	D	Время выполнения функции: 11 мс
TcpClient	com.example.diplom_main	D	Время выполнения функции: 11 мс
TcpClient	com.example.diplom_main	D	Время выполнения функции: 8 мс

Рисунок 4 – Время исполнения оптимизированного кода

2.2.4. Тесты

Тестирование приложения описано в ПРИЛОЖЕНИИ Б «Сценарий и результаты тестовых испытаний».

2.2.5. Контроль целостности данных

В таблице 5 изображен алгоритм контроля целостности данных приложения контроля аварийных ситуаций устройств.

Таблица 5 – Контроль целостности данных

№	Ситуация	Аномалия	Реакция	Примечание
1	2	3	4	5
1	Авторизация	Введён несуществующий логин и/или пароль и/или введены некорректные, согласно ограничениям, данные в соответствующие поля	Приложение обработает ответ базы данных и выведет пользователю диалоговое окно о несовпадении логина и пароля	Авторизация возможна только по существующим в БД данным
2	Добавление устройства	Не введено название устройства	Приложение сообщает о пустом поле, данные не добавляются	Данные при добавлении устройства не могут быть пустыми

№	Ситуация	Аномалия	Реакция	Примечание
1	2	3	4	5
		Введены некорректные, согласно ограничениям, данные в соответствующие поля	Приложение обработает ошибку с базы данных и выведет пользователю диалоговое окно с содержанием ошибки	Добавление устройства возможно только с валидными полями, соответствующим ограничениям полей
3	Добавление расположения	Не введено название расположения	Приложение сообщает о пустом поле, данные не добавляются	Данные при добавлении расположения не могут быть пустыми
		Введены некорректные, согласно ограничениям, данные в соответствующие поля	Приложение обработает ошибку с базы данных и выведет пользователю диалоговое окно с содержанием ошибки	Добавление расположения возможно только с валидными полями, соответствующим ограничениям полей
4	Подключение счетчика	Введены пустые или некорректные, данные в соответствующие поля	Приложение обработает ошибку подключения и выведет пользователю диалоговое окно с сообщением о неверных данных подключения	Данные при подключении счетчика должны быть корректными, и не пустыми, а на самом адресе должен быть запущен сервер
5	Подключение к серверу	Введены пустые или некорректные, данные в соответствующие поля	Приложение обработает ошибку подключения и выведет пользователю диалоговое окно с сообщением о неверных данных подключения	Данные при подключении счетчика должны быть корректными, и не пустыми, а на самом адресе должен быть запущен сервер
6	Фильтрация устройств	Введены пустые или	Приложение обработает	Данные при подключении

№	Ситуация	Аномалия	Реакция	Примечание
1	2	3	4	5
		некорректные, данные в соответствующие поля	ошибку данных и выведет пользователю диалоговое окно с сообщением о неверных данных подключения	счетчика должны быть корректными, и не пустыми, а на самом адресе должен быть запущен сервер

2.3. Проектирование

2.3.1. Схема архитектуры приложения

Компонент пользовательский интерфейс виден пользователю и взаимодействует с ним. В зависимости от действий пользователя осуществляются запросы на сервер на котором хранятся данные, который возвращает ответы на запрос в виде данных.

На Рисунке 16 представлена архитектурная схема приложения.

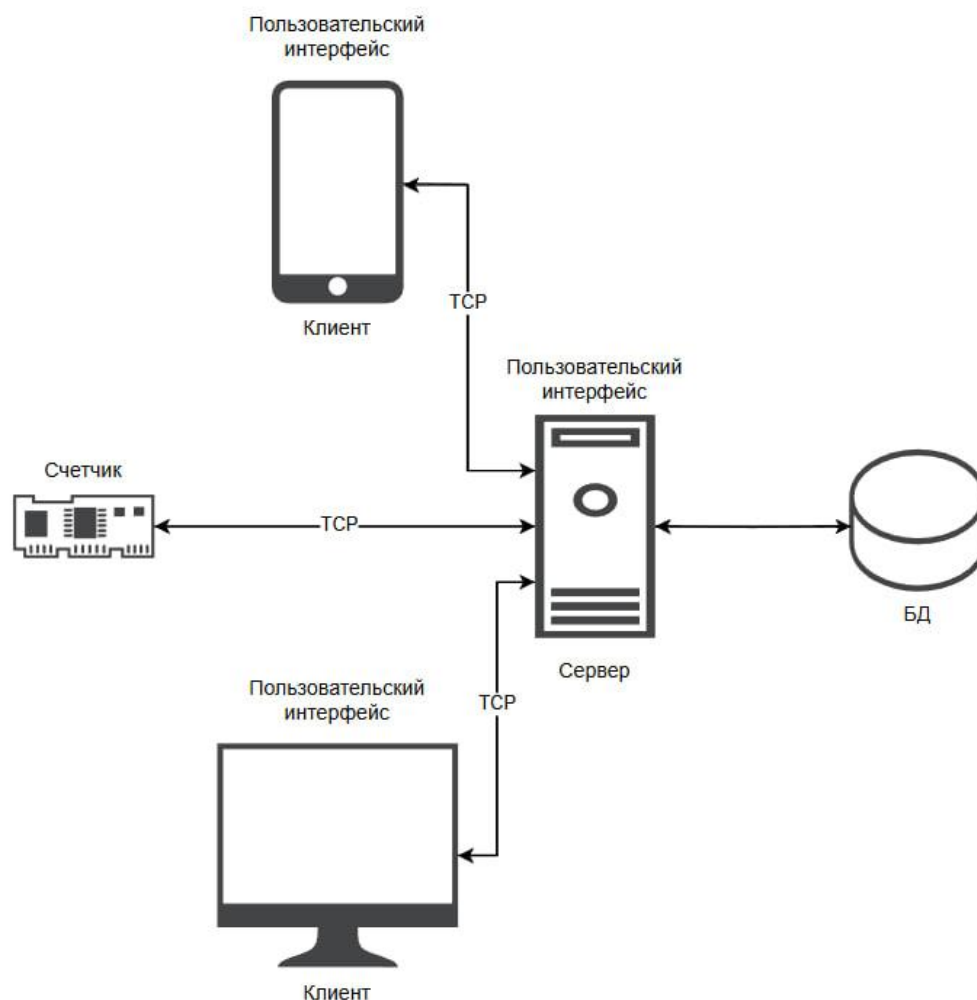


Рисунок 5 – Архитектурная схема приложения

2.3.2. Логическая схема данных

Разработанная логическая модель базы данных, представленная на Рисунке 17, послужила основой для реализации логики манипуляции данными в проектируемой базе данных.

В базе данных содержатся 5 сущностей: Расположения устройств, устройства, история аварийных случаев устройств, роль пользователя, пользователь.

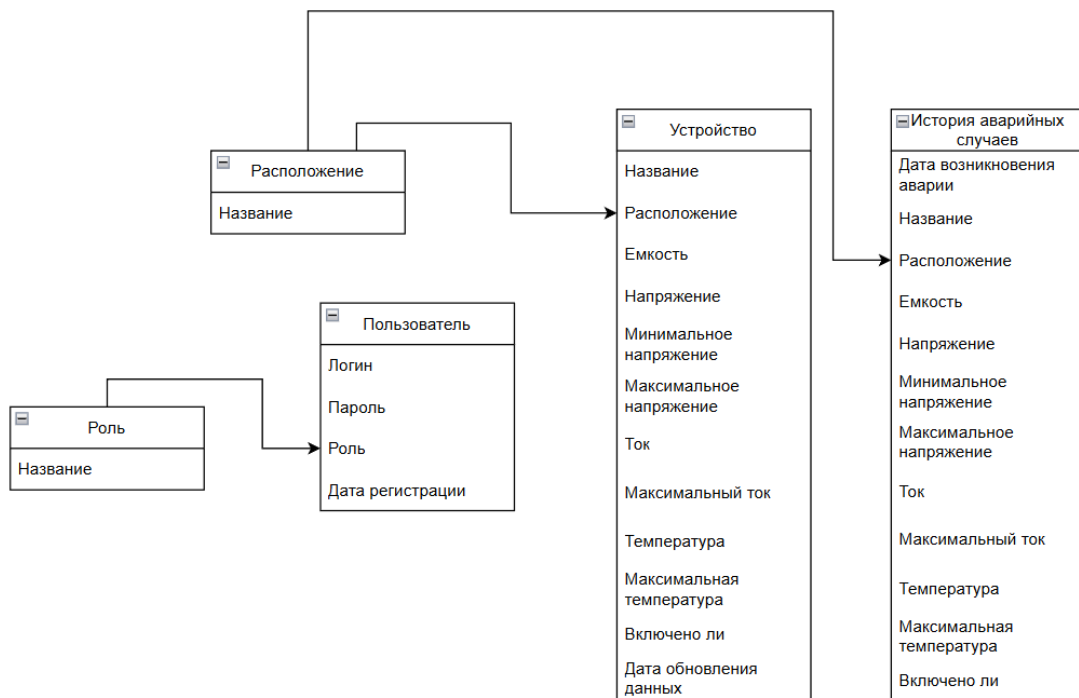


Рисунок 6 – Логическая схема данных

2.3.3. Физическая схема данных

На Рисунке 18 представлена модель базы данных с фактическими названиями и типами данных в схематическом виде.

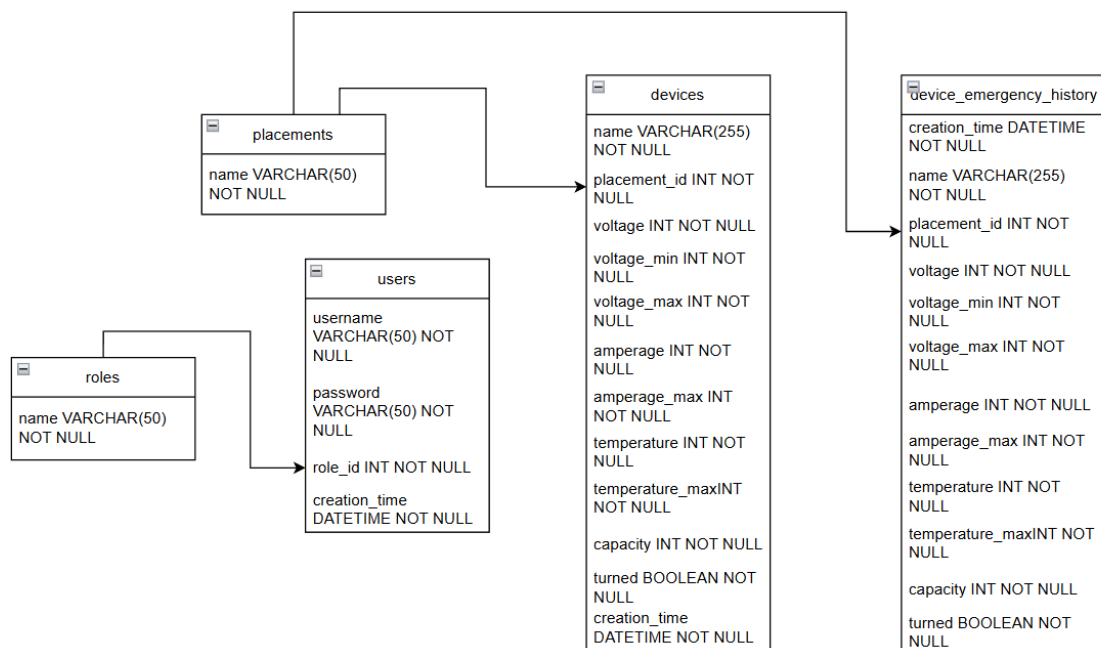


Рисунок 7 – Физическая схема данных

Словарь данных представлен в приложении Г «Скрипт базы данных».

2.3.4. Структурная схема

На Рисунке 19 представлен план структурной схемы Desktop-приложения серверной части, в которой графически показано взаимодействие всех компонентов и их переходы между друг другом.

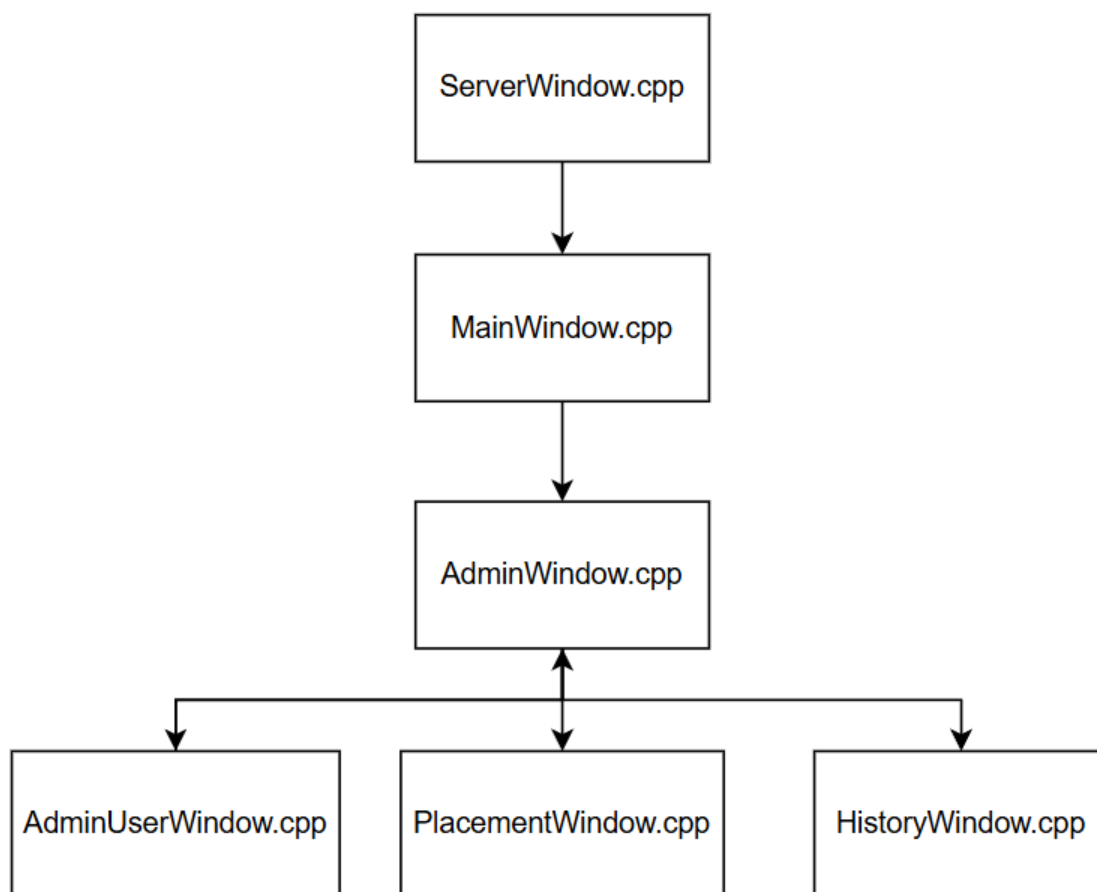


Рисунок 8 – Структурная схема серверной части

На Рисунке 20 представлен план структурной схемы Desktop-приложения клиентской части.

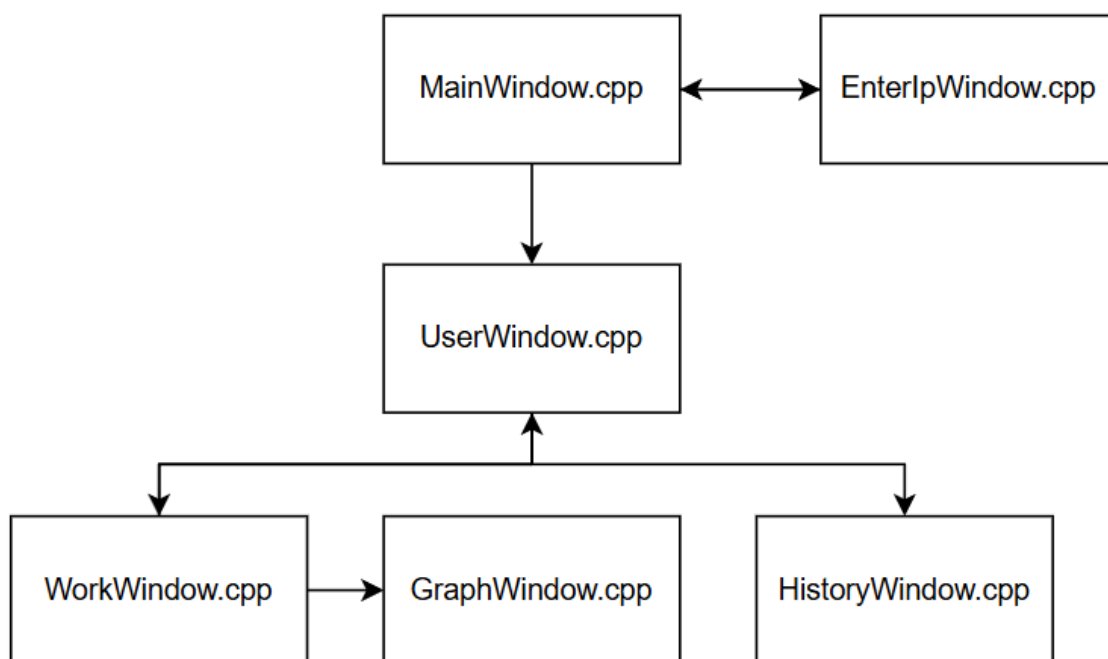


Рисунок 9 – Структурная схема Desktop-клиента

На Рисунке 21 представлен план структурной схемы Android-приложения клиентской части.

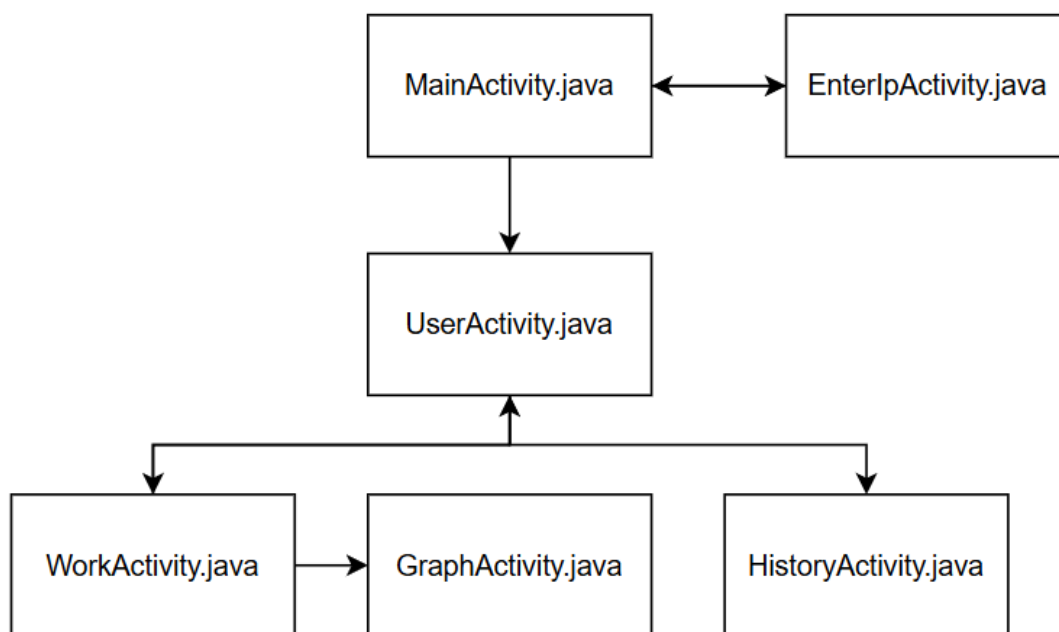


Рисунок 10 – Структурная схема Android-клиента

2.3.5. Функциональная схема

На Рисунке 22 представлена функциональная схема Desktop-приложения серверной части, на которой графически изображены возможности пользователя в процессе эксплуатации.

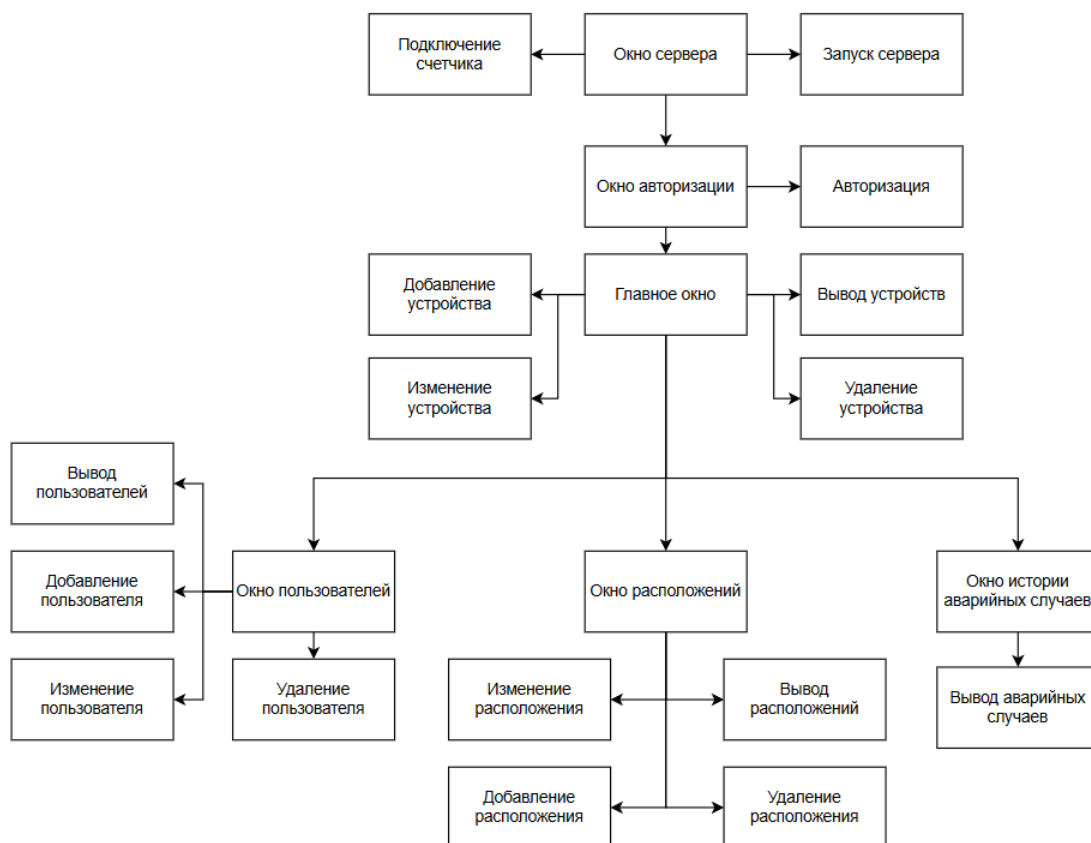


Рисунок 11 – Функциональная схема серверной части

На Рисунке 23 представлена функциональная схема приложений клиентской части, на которой графически изображены возможности пользователя в процессе эксплуатации.

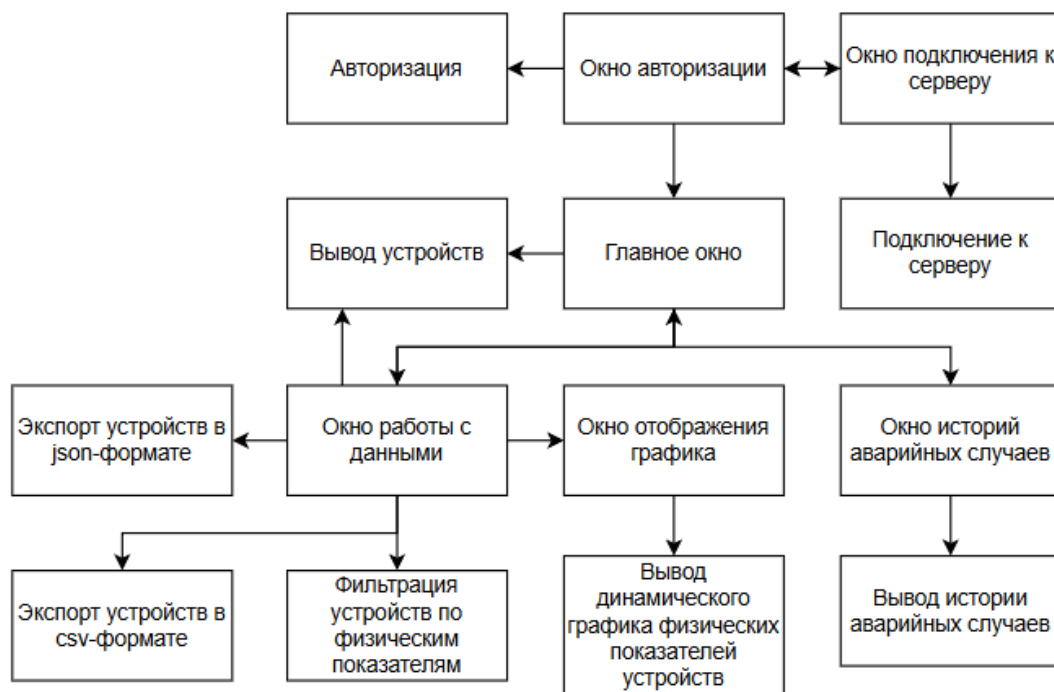


Рисунок 12 – Функциональная схема клиентской части

2.3.6. Диаграмма классов

На Рисунке 24 представлена диаграмма классов серверной части программного комплекса, на котором графически изображены модули, поля и атрибуты классов.

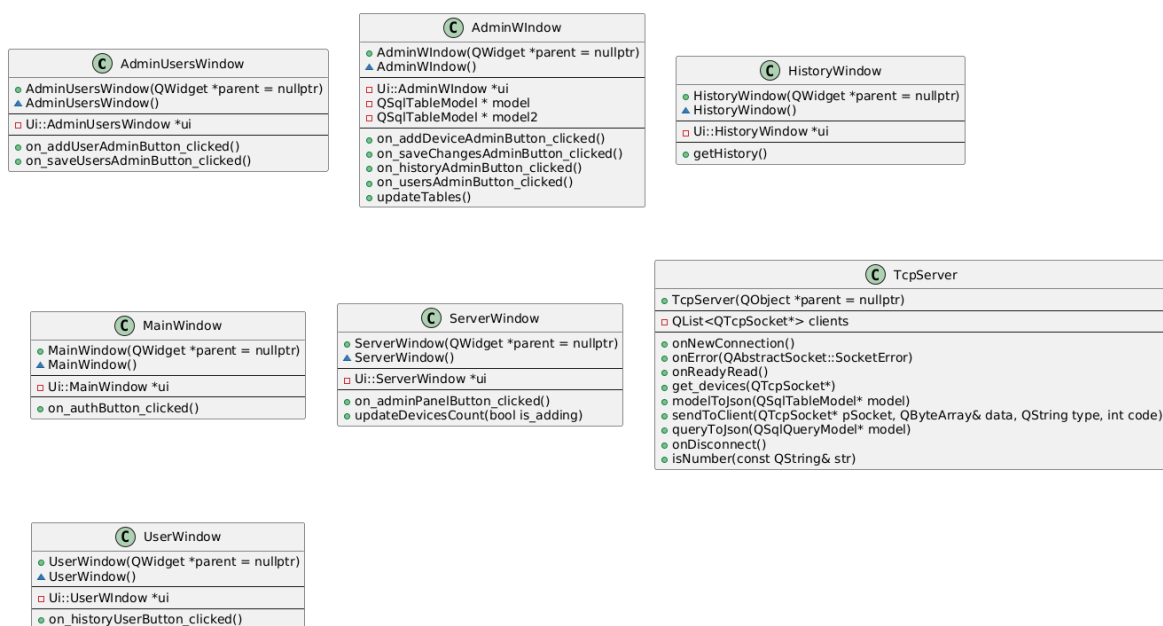


Рисунок 13 – Диаграмма классов серверной части

На Рисунке 25 представлена диаграмма классов клиентской Desktop части программного комплекса.

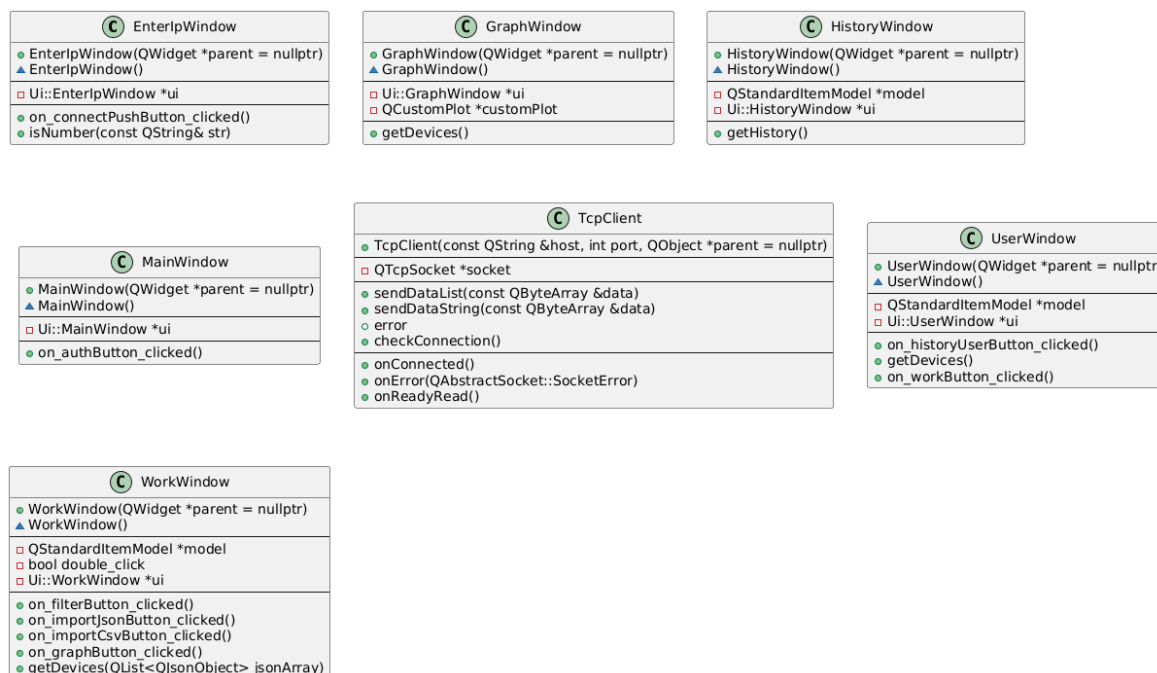


Рисунок 14 - Диаграмма классов клиентской Desktop части

На рисунке 26 изображена диаграмма классов клиентской Android части программного комплекса, которая, для удобства, разделена границами, по которым диаграмма будет разделена на фрагменты на рисунках 27-29. Фрагменты разделены границами в виде соответствующих цветов (Красный, синий, зеленый).

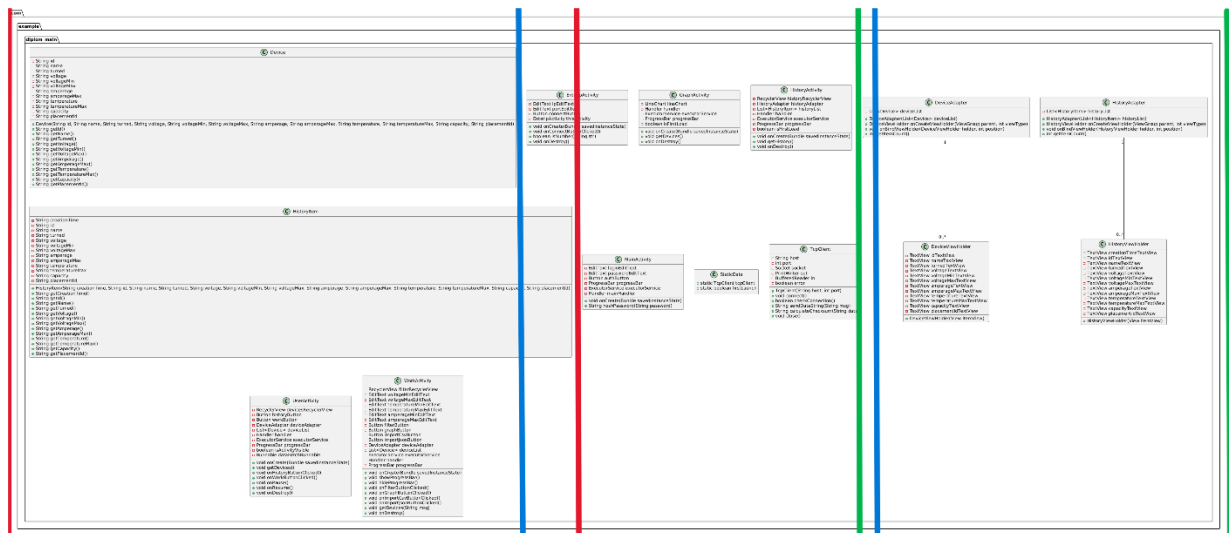


Рисунок 15 – Диаграмма классов клиентской Android части



Рисунок 16 – Фрагмент диаграммы классов клиентской Android части №1

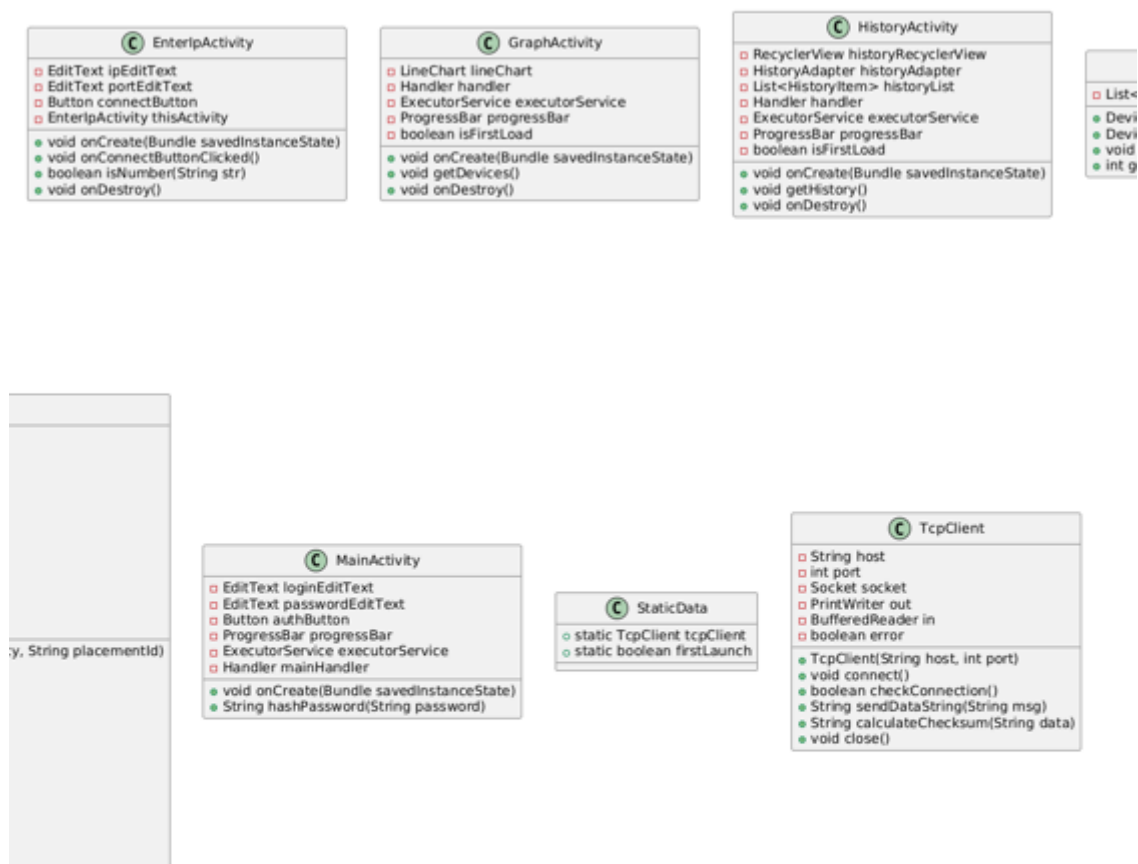


Рисунок 17 - Фрагмент диаграммы классов клиентской Android части №2

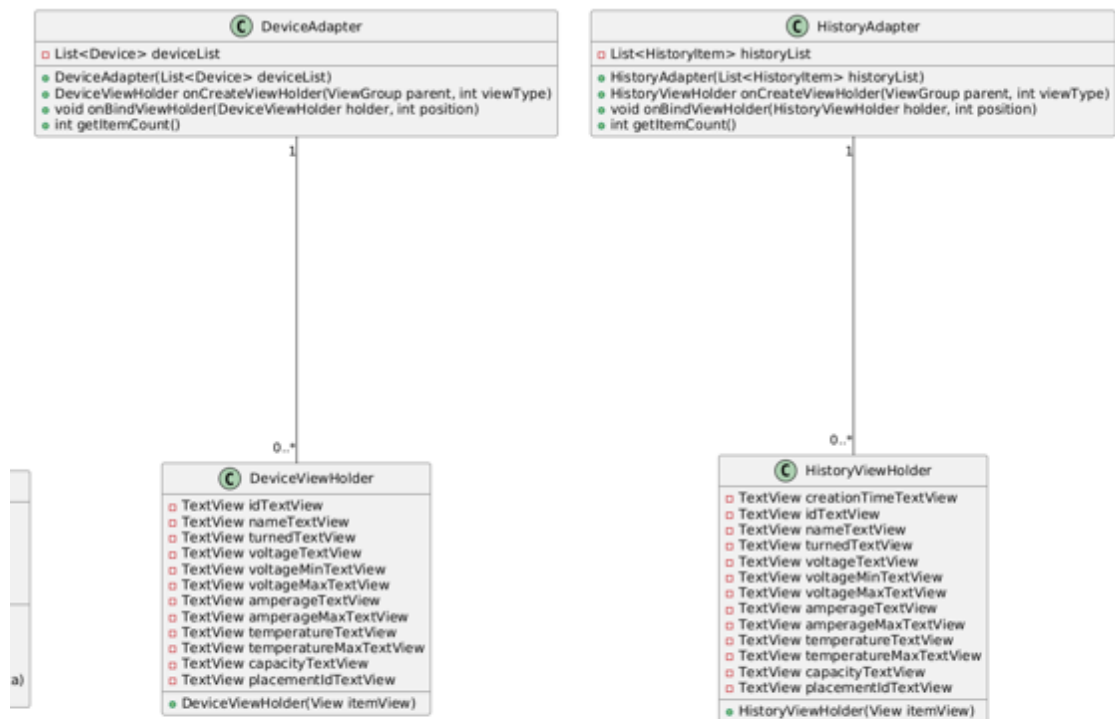


Рисунок 18 - Фрагмент диаграммы классов клиентской Android части №3

2.3.7. Схема тестирования

На Рисунке 30 представлена схема тестирования серверной части приложения, на котором графически изображены возможности приложения в процессе эксплуатации.

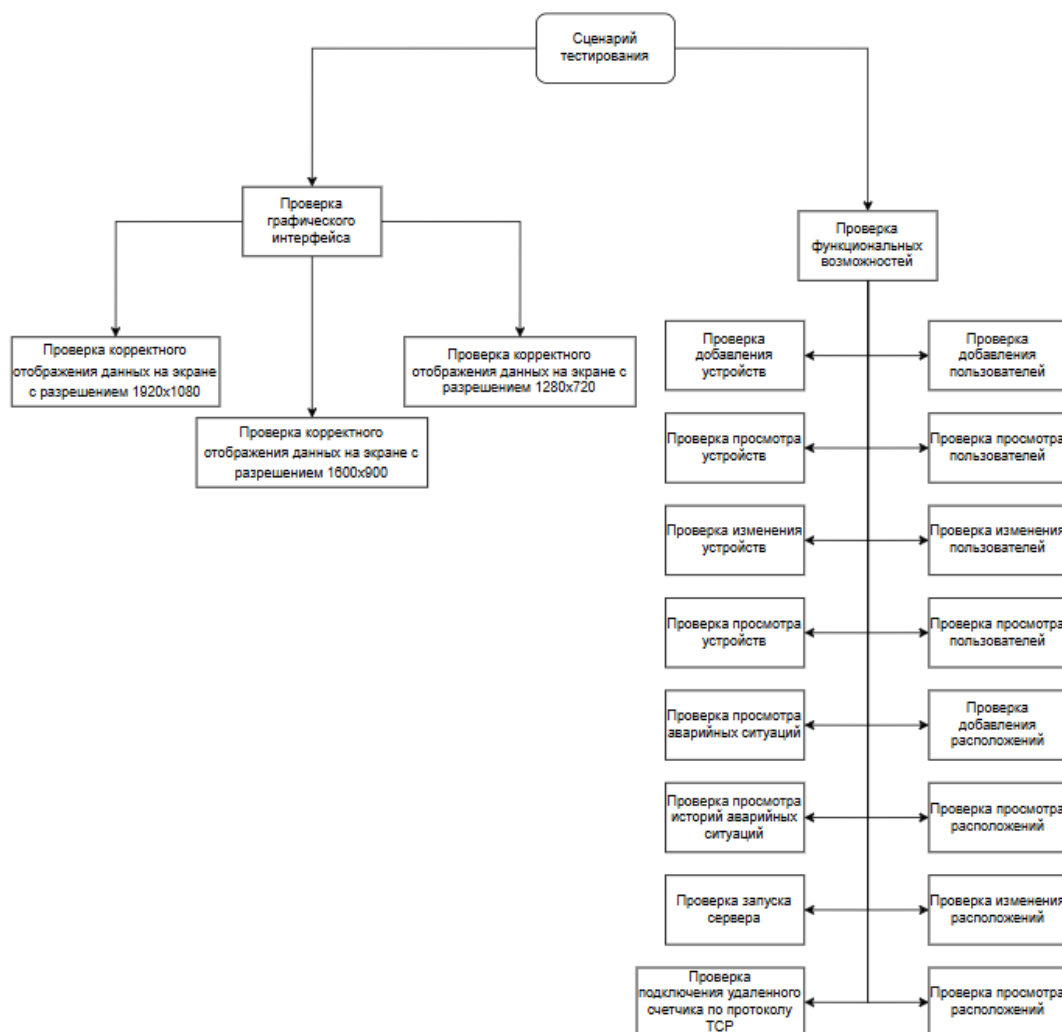


Рисунок 19 – Схема тестирования серверной части

На Рисунке 31 представлена схема тестирования клиентской части приложения.

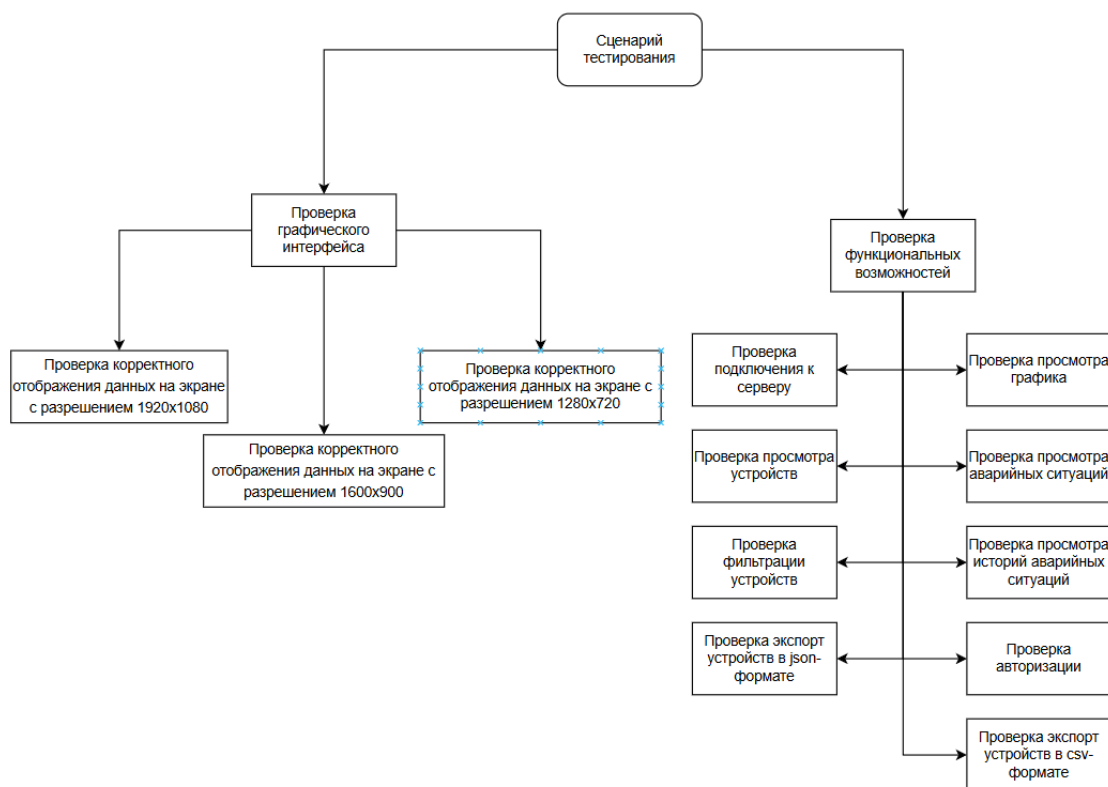


Рисунок 20 – Схема тестирования клиентской части

2.3.8. Схема пользовательского интерфейса

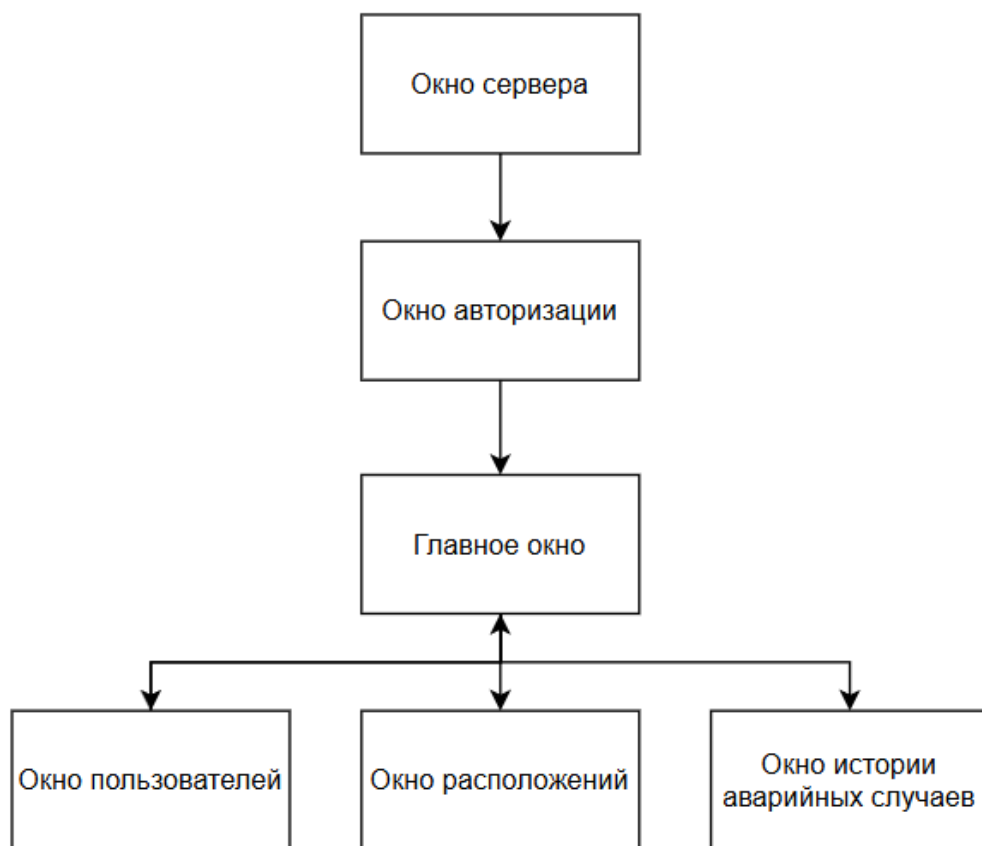


Рисунок 21 – Схема пользовательского интерфейса серверной части

Согласно схеме, на Рисунке 32, у пользователя есть возможность открыть страницу сервера, где есть возможность перейти на страницу авторизации, где необходимо ввести данные для входа в свой аккаунт. Авторизовавшись, откроется админ-панель в случае, если пользователь с введенными авторизационными данными обладает ролью «Администратор», где отображаются устройства, аварийные устройства, есть возможность перейти на страницу пользователей, страницу расположений, страницу историй аварийных случаев, а также есть возможность добавить устройство в БД или изменить его. Страница расположений содержит в себе список расположений устройств в БД и дает возможность его изменять, пополнять или удалять расположения из БД. Страница пользователей хранит тот же функционал, однако выводит

не всю информацию, а за исключением пароля, для безопасности данных. Перейдя на страницу истории аварийных случаев, выведутся все аварийные ситуации, когда-либо происходившие в процессе отслеживания.

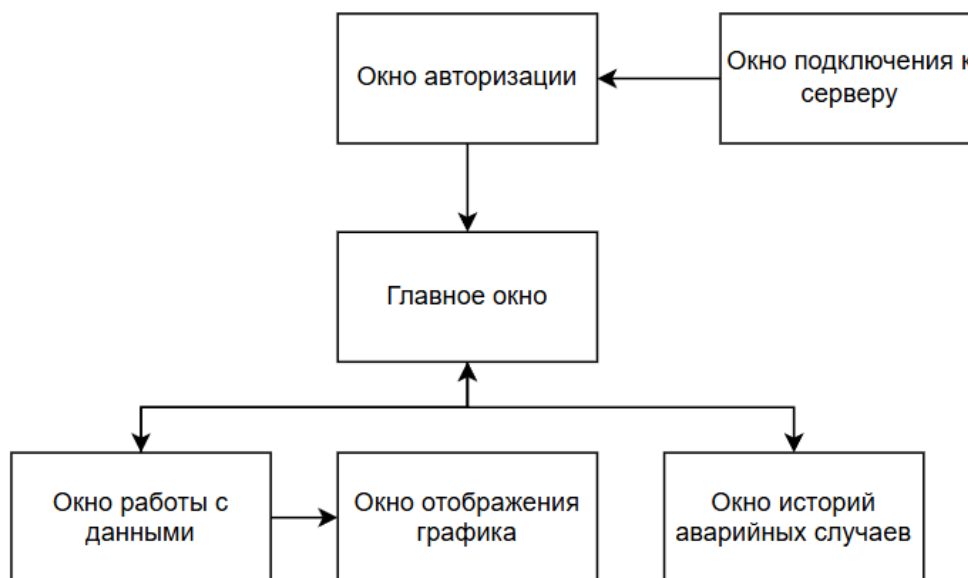


Рисунок 22 – Схема пользовательского интерфейса клиентской части

Согласно схеме, на Рисунке 33, у пользователя есть возможность открыть страницу подключения к серверу, где возможно ввести IP-адрес сервера и его порт, который отображается на начальном окне сервера (по умолчанию 1234). После успешного подключения откроется страница авторизации, где необходимо ввести авторизационные данные для допуска к дальнейшей программе. После авторизации, откроется главное окно, где отображаются устройства (в том числе и аварийные), а также кнопки перехода на окно истории аварийных случаев и перехода на окно работы с данными. Перейдя на окно истории аварийных случаев отобразится список, в котором будут указаны все когда либо случившиеся аварийные случаи с фиксацией значений и датой аварии. Перейдя на окно работы с данными, отобразятся данные устройств, а также поля для их

фильтрации по физическим показателям. Кроме того, на окне расположены кнопки для сохранения данных устройств в json и csv формате, а также кнопка перехода на окно графика. На этом окне отображается динамический график по физическим показателям устройств.

2.4. Результат работы программы

В результате поставленной задачи было разработано требуемое программное решение в виде программного комплекса из трех приложений.

На рисунке 34 представлен основной экран Desktop сервера. Окно авторизации, главное окно, окно пользователей, окно расположений, окно истории аварийных случаев, окно подключения к серверу, окно работы с данными, окно отображения графика описаны в ПРИЛОЖЕНИИ Г «Руководство пользователя».

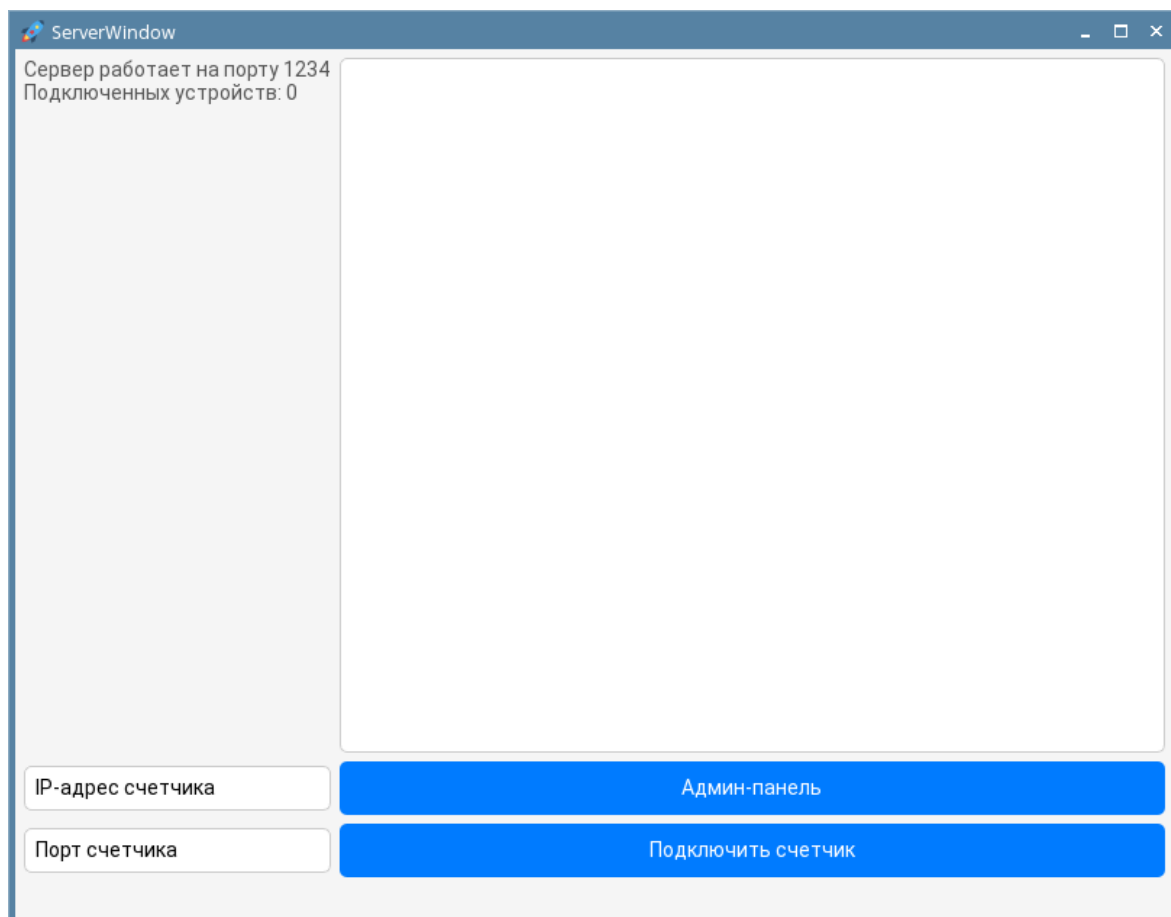


Рисунок 23 – Страница авторизации

3. ТЕХНОЛОГИЧЕСКАЯ ЧАСТЬ

3.1. Инструментальные средства

Для разработки программного комплекса контроля аварийных ситуаций устройств были использованы интегрированные средства разработки Qt Creator и Android Studio. Они предлагают мощные инструменты для написания, отладки и тестирования приложений на многих языках программирования и для разных платформ. Qt Creator и Android Studio поддерживает множество библиотек, что делает разработку более гибкой и продуктивной. Удобный интерфейс и встроенные средства отладки помогают разработчикам эффективно работать с кодом.

В качестве системы управления базами данных была выбрана PostgreSQL — мощная объектно-реляционная база данных, которая обеспечивает высокую производительность и надежное хранение данных. PostgreSQL идеально подходит для средних и крупных приложений, благодаря своей масштабируемости, поддержке сложных запросов и расширенным возможностям управления данными. Это требует настройки отдельного сервера, но позволяет интегрировать базу данных в приложение с учетом высоких требований к производительности и безопасности.

C++ был выбран в качестве основного языка разработки благодаря своей мощной поддержке создания нативных приложений и гибкости в работе с графическими интерфейсами. Он поддерживает объектно-ориентированный подход и множество современных библиотек, таких как Qt, что делает его идеальным для разработки контроля аварийных ситуаций устройств. Java, как второй язык разработки был выбран для разработки мобильной части программного комплекса, поскольку Java

является основным языком программирования для Android с момента его запуска, и многие библиотеки и фреймворки, используемые в Android-разработке, написаны на Java, что делает его естественным выбором для создания приложений для мобильных платформ.

3.2. Отладка программы

Отладка — это важный этап разработки программного обеспечения, на котором выявляются и исправляются ошибки в коде. В рамках разработки Desktop-приложения контроля аварийных ситуаций устройств помогает обеспечить корректную работу всех функций приложения, таких как управление рецептами, личный профиль и взаимодействие с базой данных.

Qt Creator и Android Studio: интегрированные среды разработки, которые предоставляют мощные инструменты для отладки, включая точки останова, просмотр переменных и трассировку стека.

Логирование: использование встроенного логирования Qt, Logcat, Toast, а также библиотеки qDebug для записи информации о работе программы, что позволяет отслеживать выполнение кода и выявлять проблемные места.

3.3. Защитное программирование

На Рисунке 35 представлена обработка ошибок при авторизации пользователя в случае неверных данных или ошибки подключения.

```

void MainWindow::on_authButton_clicked()
{
    QString login = ui->login_tb->text();
    QString password = QCryptographicHash::hash(ui->password_tb->text().toUtf8(), QCryptographicHash::Sha256).toHex();
    QString str = "log_" + login + "_" + password;
    QString str_res = client->sendDataString(str.toUtf8());
    QList<QString> arr = str_res.split('_');

    if (arr[0] == "DATA" && arr[1] == "true"){
        UserWindow * w = new UserWindow();
        w->show();
        this->close();
    }
    else if (arr[0] == "DATA" && arr[1] == "false"){
        QMessageBox::information(this, "Не найден", "Введен неверный логин или пароль.");
    }
    else {
        QMessageBox::information(this, "Ошибка", arr[1]);
    }
}

```

Рисунок 24 – Обработка ошибок при авторизации

3.4. Характеристики программы

Характеристики программного комплекса представлены в приложении А «Текст программы» в таблице 1 «Модули».

ЗАКЛЮЧЕНИЕ

В рамках дипломной работы был разработан программный комплекс контроля аварийных ситуаций устройств, написанный на Qt C++ и Android Studio Java. В процессе разработки была реализована вся функциональность, предусмотренная техническим заданием. Выполнены ключевые этапы проектирования, включая разработку архитектуры приложения, создание моделей данных, диаграмм взаимодействия компонентов и описание входных и выходных данных.

На подготовительном этапе проведена настройка среды разработки Qt Creator и Android Studio, установка сервера базы данных PostgreSQL. Эти действия обеспечили надежное взаимодействие интерфейса приложения с базой данных, а также организовали хранение данных об аварийных ситуациях и профилях устройств.

В ходе разработки системы были реализованы функции мониторинга состояния устройств, отображения аварийных ситуаций, создания сервера и подключения к нему, управления учетными записями пользователей. Приложение создано с использованием языков программирования C++ и Java, а также фреймворков Qt и Android, что позволило разработать производительное, функциональное и удобное для пользователя приложение.

На этапе тестирования системы были проверены все реализованные функции, выявлены и устранены недочеты. Благодаря использованию методов защитного программирования система продемонстрировала стабильную работу и отсутствие критических ошибок.

Итоговый результат полностью соответствует поставленным задачам и создает возможности для дальнейшего развития системы.

Несмотря на достигнутые результаты, система обладает потенциалом для совершенствования. Среди направлений для дальнейшего улучшения можно выделить:

- Добавление системы автоматического уведомления о критических аварийных ситуациях.
- Реализацию функции удаленного управления устройствами.
- Адаптацию интерфейса под различные операционные системы.
- Внедрение автоматического формирования отчетов

Разработанный программный комплекс уже является эффективным инструментом для автоматизации процессов контроля аварийных ситуаций устройств и закладывает основу для дальнейшего расширения функциональности.

СПИСОК ИСПОЛЬЗУЕМЫХ МАТЕРИАЛОВ

1. Документация QT: сайт - <https://www.doc.qt.io/>. Дата обращения: 11.02.2025.
2. Документация QSql: сайт - <https://doc.qt.io/qt-5/qtsql-index.html>. Дата обращения: 15.02.25.
3. Начало работы с PostgreSQL: сайт - <https://www.postgresql.org/docs/current/tutorial-start.html>. Дата обращения: 16.02.2025.
4. Техническая документация устройства ЩП120: сайт - https://www.elpribor.ru/upload/iblock/ba0/RE_342_SHCHP_.pdf. Дата обращения: 20.02.2025.
5. Документация Android Studio: сайт - <https://developer.android.com/develop?hl=ru>. Дата обращения: 21.03.2025.
6. Документация Java: сайт - <https://docs.oracle.com/en/java/>. Дата обращения: 21.03.2025.
7. Документация TCP: сайт - <https://datatracker.ietf.org/doc/html/rfc793>. Дата обращения: 21.03.2025.
8. Написание сетевых приложений на Qt: сайт - <https://forum.qt.io/topic/134058/network-programming-with-qt>. Дата обращения: 21.03.2025
9. Разработка Android приложения в Android Studio: сайт - <https://developer.ibm.com/articles/os-android-networking/>. Дата обращения: 22.03.2025
10. Знакомство с QT: сайт - <https://www.youtube.com/watch?v=EkjaiDsiM-Q&pp=0gcJCdgAo7VqN5tD>. Дата обращения: 11.02.2025

ПРИЛОЖЕНИЕ А. ТЕКСТ ПРОГРАММЫ АННОТАЦИЯ

В данном программном документе приведён текст данные серверной части приложения контроля аварийных ситуаций устройств. В разделе «Текст программы» указано наименование программы, область применения программы, модули программы в виде таблицы с указанием описания и размера каждого модуля, код программы.

Наименование программы: документ содержит информацию о наименовании программы, что является ключевым идентификатором проекта.

Область применения программы: описана область, в которой предполагается использование программы. Это важно для понимания контекста её применения.

Код программы: представлен код программы, который выполняет функциональность, соответствующую поставленным задачам. Код разделен на соответствующие разделы, обеспечивая ясность и читаемость.

Данный документ служит ориентиром для разработчиков, предоставляя необходимую информацию о программе, её структуре и функциональности.

СОДЕРЖАНИЕ

<u>1.</u>	<u>ТЕКСТ ПРОГРАММЫ</u>	3
<u>1.1.</u>	<u>Наименование объекта</u>	3
<u>1.2.</u>	<u>Область применения объекта</u>	3
<u>1.3.</u>	<u>Модули</u>	3
<u>1.4.</u>	<u>Код программы</u>	5

1. ТЕКСТ ПРОГРАММЫ

1.1. Наименование объекта

Наименование - «DeviceMonitor».

1.2. Область применения объекта

Приложение «DeviceMonitor» предоставляет функционал для контроля аварийных ситуаций устройств. Приложение включает в себя удобный интерфейс для добавления устройств, возможность подключать счетчик по сетевому протоколу, а также запускать сервер.

1.3. Модули

В таблице 6 представлены модули программного комплекса контроля аварийных ситуаций устройств. В приложении представлено 7 модулей сервера, 8 модулей клиента Desktop и 12 модулей клиента Android с общим количеством кода – 4305.

Таблица 6 – Модули

№	Название	Описание	Количество строк
1	2	3	4
Модули Desktop Сервера			
1	ServerWindow	Основное окно, на котором содержится статус подключения, окно логирования, подключение к счетчику и кнопка открытия админ-панели	117
2	MainWindow	Окно авторизации админ-панели	191
3	AdminWindow	Основное окно админ-панели	265
4	AdminUserWindow	Окно CRUD-операций над пользователями	147
5	HistoryWindow	Окно вывода историй аварийных случаев устройств	91
6	TcpServer	Модуль реализующий запуск сервера и принимающий запросы используя TCP	212
7	main	Начальная функция программы,	57

		подключение к БД и запуск сервера	
Модули Desktop Клиента			
1	EnterIpWindow	Окно ввода IP-адреса и порта для подключения к серверу	146
2	GraphWindow	Окно графика	122
3	MainWindow	Окно авторизации	235
4	UserWindow	Основное окно	171
5	HistoryWindow	Окно вывода историй аварийных случаев устройств	131
6	TcpClient	Модуль реализующий подключение к серверу и отправляющий запросы используя TCP	156
7	WorkWindow	Модуль реализующий фильтрацию данных и их сохранение	272
8	main	Начальная функция программы, подключение к серверу	49
Модули Android Клиента			
1	Device	Модель устройства	48
2	DeviceAdapter	Класс для отображения устройств в RecyclerView	103
3	EnterIpActivity	Окно ввода IP-адреса и порта для подключения к серверу	77
4	GraphActivity	Окно графика	124
5	HistoryActivity	Окно вывода историй аварийных случаев устройств	106
6	HistoryAdapter	Класс для отображения истории аварийных случаев в RecyclerView	79
7	HistoryItem	Модель истории аварийных случаев	51
8	MainActivity	Окно авторизации	108
9	StaticData	Класс хранения статических данных, общих для всего приложения	8
10	TcpClient	Модуль реализующий подключение к серверу и отправляющий запросы используя TCP	137
11	UserActivity	Основное окно	122

12	WorkActivity	Модуль реализующий фильтрацию данных и их сохранение	181
----	--------------	--	-----

1.4. Код программы

Ниже представлен код модулей, разделенный строчками с названиями файлов:

adminuserswindow.cpp

```
#include "adminuserswindow.h"
#include "ui_adminuserswindow.h"
QSqlTableModel * model_3;
bool second_add_2 = false;
AdminUsersWindow::AdminUsersWindow(QWidget *parent) :
    QMainWindow(parent),
    ui(new Ui::AdminUsersWindow)
{
    ui->setupUi(this);
    model_3 = new QSqlTableModel(this);
    model_3->setTable("users");
    model_3->select();
    model_3->removeColumn(2);
    model_3->setEditStrategy(QSqlTableModel::OnManualSubmit);
    ui->usersAdminTable->setModel(model_3);
    connect(ui->saveUsersAdminButton, &QPushButton::clicked, this,
        &AdminUsersWindow::on_saveUsersAdminButton_clicked);
    connect(ui->addUserAdminButton, &QPushButton::clicked, this,
        &AdminUsersWindow::on_addUserAdminButton_clicked);
}
void AdminUsersWindow::on_saveUsersAdminButton_clicked() {
    model_3->submitAll();
}
void AdminUsersWindow::on_addUserAdminButton_clicked() {
    if (!second_add_2) {
        QSqlQuery query;
        second_add_2 = true;
        QString login = ui->loginAdminText->text();
        QString password = QCryptographicHash::hash(ui->passwordAdminText->text().toUtf8(),
            QCryptographicHash::Sha256).toHex();
        bool isAdmin = ui->addUserAdminButton->isChecked();
        QString str = "insert into users (username, password, role_id) "
            "values ('%1', '%2', '%3');";
        QString str1 = str.arg(login).arg(password).arg(isAdmin ? 1 : 2);
        if (!query.exec(str1)) {
            QMessageBox::information(this, "Ошибка", query.lastError().text());
        }
        else {
            model_3->submitAll();
        }
    }
    else {
        second_add_2 = false;
    }
}
AdminUsersWindow::~AdminUsersWindow()
```

```

{
    delete ui;
}

```

adminuserswindow.h

```

#ifndef ADMINUSERSWINDOW_H
#define ADMINUSERSWINDOW_H

#include <QMainWindow>
#include <QSqlQueryModel>
#include <QSqlQuery>
#include <QSqlTableModel>
#include <QMessageBox>
#include <QDebug>
#include <QSqlError>
#include <QCryptographicHash>

namespace Ui {
class AdminUsersWindow;
}

class AdminUsersWindow : public QMainWindow
{
    Q_OBJECT

public:
    explicit AdminUsersWindow(QWidget *parent = nullptr);
    ~AdminUsersWindow();
private slots:
    void on_addUserAdminButton_clicked();
    void on_saveUsersAdminButton_clicked();
private:
    Ui::AdminUsersWindow *ui;
};

#endif // ADMINUSERSWINDOW_H

```

adminuserswindow.ui

```

<?xml version="1.0" encoding="UTF-8"?>
<ui version="4.0">
<class>AdminUsersWindow</class>
<widget class="QMainWindow" name="AdminUsersWindow">
<property name="geometry">
<rect>
<x>0</x>
<y>0</y>
<width>800</width>
<height>600</height>
</rect>
</property>
<property name="windowTitle">
<string>Main Window</string>
</property>
<widget class="QWidget" name="centralwidget">
<layout class="QGridLayout" name="gridLayout">
<item row="0" column="0" colspan="4">

```



```

<widget class="QTableView" name="usersAdminTable"/>
</item>
<item row="1" column="0">
<widget class="QPushButton" name="saveUsersAdminButton">
<property name="text">
<string>Сохранить изменения</string>
</property>
</widget>
</item>
<item row="2" column="1">
<spacer name="horizontalSpacer">
<property name="orientation">
<enum>Qt::Horizontal</enum>
</property>
<property name="sizeHint" stdset="0">
<size>
<width>232</width>
<height>20</height>
</size>
</property>
</spacer>
</item>
<item row="2" column="2">
<widget class="QLineEdit" name="loginAdminText">
<property name="placeholderText">
<string>Логин</string>
</property>
</widget>
</item>
<item row="2" column="3">
<widget class="QCheckBox" name="isUserAdminButton">
<property name="text">
<string>Администратор</string>
</property>
</widget>
</item>
<item row="3" column="2">
<widget class="QLineEdit" name="passwordAdminText">
<property name="placeholderText">
<string>Пароль</string>
</property>
</widget>
</item>
<item row="3" column="3">
<widget class="QPushButton" name="addUserAdminButton">
<property name="text">
<string>Добавить аккаунт</string>
</property>
</widget>
</item>
</layout>
</widget>
<widget class="QMenuBar" name="menubar">
<property name="geometry">
<rect>
<x>0</x>
<y>0</y>
<width>800</width>
<height>29</height>
</rect>

```

```

</property>
</widget>
<widget class="QStatusBar" name="statusbar"/>
</widget>
<resources/>
<connections/>
</ui>

```

adminwindow.cpp

```

#include "adminwindow.h"
#include "ui_adminwindow.h"
#include "historywindow.h"
#include "adminuserswindow.h"

bool second_add = false;
AdminWIndow::AdminWIndow(QWidget *parent) :
    QMainWindow(parent),
    ui(new Ui::AdminWIndow)
{
    ui->setupUi(this);
    setWindowFlags(Qt::Dialog);
    connect(ui->addDeviceAdminButton, &QPushButton::clicked, this,
        &AdminWIndow::on_addDeviceAdminButton_clicked);
    connect(ui->saveChangesAdminButton, &QPushButton::clicked, this,
        &AdminWIndow::on_saveChangesAdminButton_clicked);
    model = new QSqlTableModel(this);
    model2 = new QSqlTableModel(this);
    updateTables();
    QTimer *timer = new QTimer(this);
    QObject::connect(timer, &QTimer::timeout, this, &AdminWIndow::updateTables);
    timer->start(15000);
}

void AdminWIndow::updateTables() {
    model->clear();
    model->setTable("devices");
    model->select();
    model->setEditStrategy(QSqlTableModel::OnManualSubmit);
    ui->devicesAdminTable->setModel(model);
    model2->clear();
    model2->setTable("devices");
    model2->setFilter("voltage > voltage_max or voltage < voltage_min or amperage > amperage_max or
temperature > temperature_max");
    model2->select();
    model2->setEditStrategy(QSqlTableModel::OnManualSubmit);
    ui->emergencyDeviceAdminTable->setModel(model2);
    ui->emergencyDeviceAdminTable->setEditTriggers(QAbstractItemView::NoEditTriggers);
    int a = ui->placementAdminCB->currentIndex();
    ui->placementAdminCB->clear();
    QSqlQuery query("select * from placements");
    while(query.next()) {
        int id = query.value(0).toInt();
        QString name = query.value(1).toString();
        ui->placementAdminCB->addItem(name, id);
    }
    ui->placementAdminCB->setCurrentIndex(a);
}

```

```

void AdminWIndow::on_addDeviceAdminButton_clicked(){
    if (!second_add){
        second_add = true;
        QString name = ui->NameAdminEdit->text();
        bool turned = ui->TurnAdminCB->isChecked();
        int voltage = ui->VoltageAdminEdit->text().toInt();
        int voltage_max = ui->maxVoltageAdminEdit->text().toInt();
        int voltage_min = ui->minVoltageAdminEdit->text().toInt();
        int amperage = ui->AmperageAdminEdit->text().toInt();
        int amperage_max = ui->maxAmperageAdminEdit->text().toInt();
        int temperature = ui->TempAdminEdit->text().toInt();
        int temperature_max = ui->maxTempAdminEdit->text().toInt();
        int capacity = ui->CapacityAdminEdit->text().toInt();
        int placement_id = ui->placementAdminCB->currentData().toInt();
        model->insertRows(0, 1);
        model->setData(model->index(0, 1), name);
        model->setData(model->index(0, 2), turned);
        model->setData(model->index(0, 3), voltage);
        model->setData(model->index(0, 4), voltage_max);
        model->setData(model->index(0, 5), voltage_min);
        model->setData(model->index(0, 6), amperage);
        model->setData(model->index(0, 7), amperage_max);
        model->setData(model->index(0, 8), temperature);
        model->setData(model->index(0, 9), temperature_max);
        model->setData(model->index(0, 10), capacity);
        model->setData(model->index(0, 11), placement_id);
        if (!model->submitAll()){
            QMessageBox::information(this, "Ошибка", "Ошибка выполнения запроса!");
        }
    }
    else {
        second_add = false;
    }
}

void AdminWIndow::on_saveChangesAdminButton_clicked(){
    model->submitAll();
}

void AdminWIndow::on_usersAdminButton_clicked(){
    AdminUsersWindow * w = new AdminUsersWindow(this);
    w->show();
}

void AdminWIndow::on_historyAdminButton_clicked(){
    HistoryWindow * w = new HistoryWindow(this);
    w->show();
}

AdminWIndow::~AdminWIndow()
{
    delete ui;
}

```

adminwindow.h

```

#ifndef ADMINWINDOW_H
#define ADMINWINDOW_H

#include <QMainWindow>
#include <QSqlQueryModel>
#include <QSqlQuery>

```

```

#include <QSqlTableModel>
#include <QMessageBox>
#include <QDebug>
#include <QTimer>

namespace Ui {
class AdminWindow;
}

class AdminWindow : public QMainWindow
{
    Q_OBJECT

public:
    explicit AdminWindow(QWidget *parent = nullptr);
    ~AdminWindow();
private slots:
    void on_addDeviceAdminButton_clicked();
    void on_saveChangesAdminButton_clicked();
    void on_historyAdminButton_clicked();
    void on_usersAdminButton_clicked();
    void updateTables();
private:
    Ui::AdminWindow *ui;
    QSqlTableModel * model;
    QSqlTableModel * model2;
};

#endif // ADMINWINDOW_H

```

```

adminwindow.ui

```

```

<?xml version="1.0" encoding="UTF-8"?>
<ui version="4.0">
<class>AdminWindow</class>
<widget class="QMainWindow" name="AdminWindow">
<property name="geometry">
<rect>
<x>0</x>
<y>0</y>
<width>800</width>
<height>600</height>
</rect>
</property>
<property name="windowTitle">
<string>Admin Window</string>
</property>
<widget class="QWidget" name="centralwidget">
<layout class="QGridLayout" name="gridLayout">
<item row="6" column="3">
<widget class="QLineEdit" name="AmperageAdminEdit">
<property name="text">
<string/>
</property>
<property name="placeholderText">
<string>Tok</string>
</property>
</widget>
</item>

```

```

<item row="6" column="1" colspan="2">
<widget class="QLineEdit" name="minVoltageAdminEdit">
<property name="placeholderText">
<string>Мин. Напряжение</string>
</property>
</widget>
</item>
<item row="7" column="1" colspan="2">
<widget class="QLineEdit" name="maxVoltageAdminEdit">
<property name="placeholderText">
<string>Макс. Напряжение</string>
</property>
</widget>
</item>
<item row="5" column="1" colspan="2">
<widget class="QLineEdit" name="VoltageAdminEdit">
<property name="placeholderText">
<string>Напряжение</string>
</property>
</widget>
</item>
<item row="7" column="0">
<widget class="QComboBox" name="placementAdminCB">
<property name="currentText">
<string/>
</property>
<property name="placeholderText">
<string>Помещение</string>
</property>
</widget>
</item>
<item row="3" column="0" colspan="6">
<widget class="QTableView" name="devicesAdminTable"/>
</item>
<item row="2" column="0" colspan="6">
<widget class="QTableView" name="emergencyDeviceAdminTable">
<property name="styleSheet">
<string notr="true">color: rgb(255, 0, 0);</string>
</property>
</widget>
</item>
<item row="1" column="3" colspan="2">
<widget class="QLabel" name="label_3">
<property name="text">
<string>Аварийные устройства</string>
</property>
</widget>
</item>
<item row="6" column="0">
<widget class="QLineEdit" name="NameAdminEdit">
<property name="placeholderText">
<string>Название</string>
</property>
</widget>
</item>
<item row="1" column="0" colspan="2">
<widget class="QLabel" name="label_2">
<property name="text">
<string>Все устройства</string>
</property>

```

```

</widget>
</item>
<item row="0" column="2" colspan="2">
<widget class="QLabel" name="label">
<property name="text">
<string>Панель администратора</string>
</property>
</widget>
</item>
<item row="5" column="0">
<widget class="QLineEdit" name="CapacityAdminEdit">
<property name="placeholderText">
<string>Емкость</string>
</property>
</widget>
</item>
<item row="5" column="3">
<widget class="QLineEdit" name="TempAdminEdit">
<property name="text">
<string/>
</property>
<property name="placeholderText">
<string>Температура</string>
</property>
</widget>
</item>
<item row="6" column="4" colspan="2">
<widget class="QCheckBox" name="TurnAdminCB">
<property name="text">
<string>Включено ли устройство</string>
</property>
</widget>
</item>
<item row="7" column="3">
<widget class="QLineEdit" name="maxAmperageAdminEdit">
<property name="text">
<string/>
</property>
<property name="placeholderText">
<string>Макс. Ток</string>
</property>
</widget>
</item>
<item row="5" column="4" colspan="2">
<widget class="QLineEdit" name="maxTempAdminEdit">
<property name="placeholderText">
<string>Макс. Температура</string>
</property>
</widget>
</item>
<item row="7" column="4" colspan="2">
<widget class="QPushButton" name="addDeviceAdminButton">
<property name="text">
<string>Добавить устройство</string>
</property>
</widget>
</item>
<item row="4" column="3">
<widget class="QPushButton" name="historyAdminButton">
<property name="text">

```

```

        <string>История аварийных случаев</string>
    </property>
</widget>
</item>
<item row="4" column="4" colspan="2">
    <widget class="QPushButton" name="saveChangesAdminButton">
        <property name="text">
            <string>Сохранить изменения</string>
        </property>
    </widget>
</item>
<item row="4" column="1" colspan="2">
    <widget class="QPushButton" name="usersAdminButton">
        <property name="text">
            <string>Пользователи</string>
        </property>
    </widget>
</item>
</layout>
</widget>
<widget class="QMenuBar" name="menubar">
    <property name="geometry">
        <rect>
            <x>0</x>
            <y>0</y>
            <width>800</width>
            <height>29</height>
        </rect>
    </property>
</widget>
<widget class="QStatusBar" name="statusbar"/>
</widget>
<resources/>
<connections/>
</ui>

```

historywindow.cpp

```

#include "historywindow.h"
#include "ui_historywindow.h"

```

```

 QSqlQueryModel *model = new QSqlQueryModel;
 HistoryWindow::HistoryWindow(QWidget *parent) :
     QMainWindow(parent),
     ui(new Ui::HistoryWindow)
 {
     ui->setupUi(this);
     getHistory();
     QTimer *timer = new QTimer(this);
     QObject::connect(timer, &QTimer::timeout, this, &HistoryWindow::getHistory);
     timer->start(5000);
 }
 void HistoryWindow::getHistory()
 {
     QSqlQuery query;
     query.prepare("select * from device_emergency_history JOIN devices on devices.id =
 device_emergency_history.device_id");
     if (!query.exec()){
         QMessageBox::information(this, "Ошибка", "Ошибка выполнения запроса!");
     }
 }

```

```

    }
    model->setQuery(query);
    ui->historyTable->setModel(model);
    ui->historyTable->setEditTriggers(QAbstractItemView::NoEditTriggers);
}

```

```

HistoryWindow::~HistoryWindow()
{
    delete ui;
}

```

historywindow.h

```

#ifndef HISTORYWINDOW_H
#define HISTORYWINDOW_H

```

```

#include <QMainWindow>
#include <QSqlQueryModel>
#include <QSqlQuery>
#include <QSqlTableModel>
#include <QMessageBox>
#include <QTimer>

```

```

namespace Ui {
class HistoryWindow;
}

```

```

class HistoryWindow : public QMainWindow
{
    Q_OBJECT

public:
    explicit HistoryWindow(QWidget *parent = nullptr);
    ~HistoryWindow();
private slots:
    void getHistory();
private:
    Ui::HistoryWindow *ui;
};

```

```

#endif // HISTORYWINDOW_H

```

historywindow.ui

```

<?xml version="1.0" encoding="UTF-8"?>
<ui version="4.0">
<class>HistoryWindow</class>
<widget class="QMainWindow" name="HistoryWindow">
<property name="geometry">
<rect>
<x>0</x>
<y>0</y>
<width>800</width>
<height>300</height>
</rect>
</property>
<property name="windowTitle">
<string>HistoryWindow</string>

```



```

</property>
<widget class="QWidget" name="centralwidget">
<layout class="QGridLayout" name="gridLayout">
<item row="0" column="0">
<widget class="QTableView" name="historyTable"/>
</item>
</layout>
</widget>
<widget class="QMenuBar" name="menubar">
<property name="geometry">
<rect>
<x>0</x>
<y>0</y>
<width>800</width>
<height>29</height>
</rect>
</property>
</widget>
<widget class="QStatusBar" name="statusbar"/>
</widget>
<resources/>
<connections/>
</ui>

```

main.cpp

```

#include "mainwindow.h"
#include "serverwindow.h"
#include <QApplication>
#include <QTableView>
#include "tcpserver.h"
TcpServer* server;
int* serverPort;

QString createConnection(){
    QSqlDatabase db = QSqlDatabase::addDatabase("QPSQL");
    db.setDatabaseName("microcontrollers_study");
    db.setUserName("admin");
    db.setHostName("localhost");
    db.setPassword("admin");
    if (!db.open()){
        qDebug() << "Ошибка подключения к БД: " << db.lastError();
        return db.lastError().text();
    }
    return "";
}

QString createServer(){
    server = new TcpServer();
    if (!server->listen(QHostAddress::Any, 1234)){
        return server->errorString();
    }
    qDebug() << "Сервер запущен на порте: " << server->serverPort();
    serverPort = new int(static_cast<int>(server->serverPort()));
    qDebug() << *serverPort;
    return "";
}

int main(int argc, char *argv[])

```

```

{
    QApplication a(argc, argv);
    QDir d;
    QFile file(d.absolutePath() + "../rostik_qt_admin/style.qss");
    if (file.open(QFile::ReadOnly | QFile::Text)){
        a.setStyleSheet(QLatin1String(file.readAll()));
        qDebug() << "Стили загружены.";
    }
    else{
        qDebug() << file.errorString();
        qDebug() << file.fileName();
    }
    QString conn = createConnection();
    if (conn != ""){
        QMessageBox::critical(nullptr, "Ошибка БД", "Не удалось подключиться к БД. " + conn);
        return -1;
    }
    conn = createServer();
    if (conn != ""){
        QMessageBox::critical(nullptr, "Ошибка сервера", "Не удалось запустить сервер. " + conn);
        return -1;
    }
    createServer();
    ServerWindow* w = new ServerWindow();
    w->show();
    return a.exec();
}

```

main.h

```

#ifndef MAIN_H
#define MAIN_H
#include <QTcpSocket>
#include <tcpserver.h>
#include <QDir>
extern TcpServer* server;
extern int* serverPort;
#endif // MAIN_H

```

mainwindow.cpp

```

#include "mainwindow.h"
#include "adminwindow.h"
#include "userwindow.h"
#include "ui_mainwindow.h"

MainWindow::MainWindow(QWidget *parent)
    : QMainWindow(parent)
    , ui(new Ui::MainWindow)
{
    ui->setupUi(this);
    connect(ui->auth_button, &QPushButton::clicked, this,
        &MainWindow::on_authButton_clicked);
}

MainWindow::~MainWindow()
{
}

```

```

        delete ui;
    }
    void MainWindow::on_authButton_clicked() {
        QString login = ui->login_tb->text();
        QString password = QCryptographicHash::hash(ui->password_tb->text().toUtf8(),
        QCryptographicHash::Sha256).toHex();
        QSqlQuery query;
        query.prepare("SELECT * FROM users WHERE username = " + login + " AND password = " +
        password + "");
        //query.prepare("SELECT * FROM users WHERE username = :username AND password = :password");
        //query.bindValue(":username", login);
        //query.bindValue(":password", password);
        if (!query.exec()) {
            QMessageBox::critical(this, "Ошибка", query.lastError().text() + " " + query.executedQuery());
        }
        else {
            if (query.size() > 0) {
                query.next();
                int a = query.value(4).toInt();
                if (a == 1) {
                    //QMessageBox::information(this, "Успех", "Вы вошли в систему как администратор!");
                    AdminWIndow * w = new AdminWIndow(this);
                    w->show();
                    this->close();
                }
                else {
                    QMessageBox::information(this, "Успех", "Вы вошли в систему, однако ваша роль не
                    совпадает с необходимой. Попробуйте использовать пользовательское приложение.");
                }
            }
            else {
                QMessageBox::information(this, "Не найдено", "Введен неверный логин или пароль.");
            }
        }
    }
}

```

mainwindow.h

```

#ifndef MAINWINDOW_H
#define MAINWINDOW_H

#include <QMainWindow>
#include <QtSql>
#include <QDebug>
#include <QMessageBox>

QT_BEGIN_NAMESPACE
namespace Ui { class MainWindow; }
QT_END_NAMESPACE

class MainWindow : public QMainWindow
{
    Q_OBJECT

public:
    MainWindow(QWidget *parent = nullptr);
    ~MainWindow();
private slots:
    void on_authButton_clicked();

```

```
private:
    Ui::MainWindow *ui;
};
#endif // MAINWINDOW_H
```

```
mainwindow.ui
```

```
<?xml version="1.0" encoding="UTF-8"?>
<ui version="4.0">
<class>MainWindow</class>
<widget class="QMainWindow" name="MainWindow">
<property name="geometry">
<rect>
<x>0</x>
<y>0</y>
<width>657</width>
<height>432</height>
</rect>
</property>
<property name="windowTitle">
<string>Main Window</string>
</property>
<widget class="QWidget" name="centralwidget">
<layout class="QGridLayout" name="gridLayout">
<item row="0" column="2">
<spacer name="verticalSpacer_3">
<property name="orientation">
<enum>Qt::Vertical</enum>
</property>
<property name="sizeHint" stdset="0">
<size>
<width>20</width>
<height>42</height>
</size>
</property>
</spacer>
</item>
<item row="1" column="0">
<spacer name="horizontalSpacer_7">
<property name="orientation">
<enum>Qt::Horizontal</enum>
</property>
<property name="sizeHint" stdset="0">
<size>
<width>49</width>
<height>20</height>
</size>
</property>
</spacer>
</item>
<item row="1" column="1" colspan="3">
<widget class="QLabel" name="label">
<property name="font">
<font>
<pointsize>10</pointsize>
</font>
</property>
<property name="text">
```

```

    <string>Добро пожаловать в программу администрирования электронных приборов!</string>
  </property>
</widget>
</item>
<item row="1" column="4">
  <spacer name="horizontalSpacer_8">
    <property name="orientation">
      <enum>Qt::Horizontal</enum>
    </property>
    <property name="sizeHint" stdset="0">
      <size>
        <width>49</width>
        <height>20</height>
      </size>
    </property>
  </spacer>
</item>
<item row="2" column="2">
  <spacer name="verticalSpacer_2">
    <property name="orientation">
      <enum>Qt::Vertical</enum>
    </property>
    <property name="sizeHint" stdset="0">
      <size>
        <width>20</width>
        <height>42</height>
      </size>
    </property>
  </spacer>
</item>
<item row="3" column="1">
  <spacer name="horizontalSpacer_3">
    <property name="orientation">
      <enum>Qt::Horizontal</enum>
    </property>
    <property name="sizeHint" stdset="0">
      <size>
        <width>137</width>
        <height>20</height>
      </size>
    </property>
  </spacer>
</item>
<item row="3" column="2">
  <widget class="QLabel" name="label_2">
    <property name="text">
      <string>Введите данные для авторизации:</string>
    </property>
  </widget>
</item>
<item row="3" column="3">
  <spacer name="horizontalSpacer_4">
    <property name="orientation">
      <enum>Qt::Horizontal</enum>
    </property>
    <property name="sizeHint" stdset="0">
      <size>
        <width>137</width>
        <height>20</height>
      </size>
    </property>
  </spacer>
</item>

```

```

</property>
</spacer>
</item>
<item row="4" column="1">
<spacer name="horizontalSpacer">
<property name="orientation">
<enum>Qt::Horizontal</enum>
</property>
<property name="sizeHint" stdset="0">
<size>
<width>137</width>
<height>20</height>
</size>
</property>
</spacer>
</item>
<item row="4" column="2">
<widget class="QLineEdit" name="login_tb">
<property name="placeholderText">
<string>Логин</string>
</property>
</widget>
</item>
<item row="4" column="3">
<spacer name="horizontalSpacer_5">
<property name="orientation">
<enum>Qt::Horizontal</enum>
</property>
<property name="sizeHint" stdset="0">
<size>
<width>137</width>
<height>20</height>
</size>
</property>
</spacer>
</item>
<item row="5" column="1">
<spacer name="horizontalSpacer_2">
<property name="orientation">
<enum>Qt::Horizontal</enum>
</property>
<property name="sizeHint" stdset="0">
<size>
<width>137</width>
<height>20</height>
</size>
</property>
</spacer>
</item>
<item row="5" column="2">
<widget class="QLineEdit" name="password_tb">
<property name="echoMode">
<enum>QLineEdit::Password</enum>
</property>
<property name="placeholderText">
<string>Пароль</string>
</property>
</widget>
</item>
<item row="5" column="3">

```

```

<spacer name="horizontalSpacer_6">
  <property name="orientation">
    <enum>Qt::Horizontal</enum>
  </property>
  <property name="sizeHint" stdset="0">
    <size>
      <width>137</width>
      <height>20</height>
    </size>
  </property>
</spacer>
</item>
<item row="6" column="2">
  <widget class="QPushButton" name="auth_button">
    <property name="text">
      <string>Войти</string>
    </property>
  </widget>
</item>
<item row="7" column="2">
  <spacer name="verticalSpacer_4">
    <property name="orientation">
      <enum>Qt::Vertical</enum>
    </property>
    <property name="sizeHint" stdset="0">
      <size>
        <width>20</width>
        <height>42</height>
      </size>
    </property>
  </spacer>
</item>
<item row="8" column="2">
  <spacer name="verticalSpacer">
    <property name="orientation">
      <enum>Qt::Vertical</enum>
    </property>
    <property name="sizeHint" stdset="0">
      <size>
        <width>20</width>
        <height>42</height>
      </size>
    </property>
  </spacer>
</item>
</layout>
</widget>
<widget class="QMenuBar" name="menubar">
  <property name="geometry">
    <rect>
      <x>0</x>
      <y>0</y>
      <width>657</width>
      <height>29</height>
    </rect>
  </property>
</widget>
<widget class="QStatusBar" name="statusbar"/>
</widget>
</resources/>

```

```
</connections/>
</ui>
```

```
rostik_qt_admin.pro
```

```
QT += core gui sql network
```

```
greaterThan(QT_MAJOR_VERSION, 4): QT += widgets
```

```
CONFIG += c++11
```

```
# You can make your code fail to compile if it uses deprecated APIs.
```

```
# In order to do so, uncomment the following line.
```

```
#DEFINES += QT_DISABLE_DEPRECATED_BEFORE=0x060000 # disables all the APIs deprecated
before Qt 6.0.0
```

```
SOURCES += \
    adminuserswindow.cpp \
    adminwindow.cpp \
    historywindow.cpp \
    main.cpp \
    mainwindow.cpp \
    serverwindow.cpp \
    tcpserver.cpp
```

```
HEADERS += \
    adminuserswindow.h \
    adminwindow.h \
    historywindow.h \
    main.h \
    mainwindow.h \
    serverwindow.h \
    tcpserver.h
```

```
FORMS += \
    adminuserswindow.ui \
    adminwindow.ui \
    historywindow.ui \
    mainwindow.ui \
    serverwindow.ui
```

```
# Default rules for deployment.
```

```
qnx: target.path = /tmp/${TARGET}/bin
```

```
else: unix:!android: target.path = /opt/${TARGET}/bin
```

```
!isEmpty(target.path): INSTALLS += target
```

```
rostik_qt_admin.pro.user
```

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<!DOCTYPE QtCreatorProject>
```

```
<!-- Written by QtCreator 6.0.2, 2024-11-11T09:55:05. -->
```

```
<qtcreator>
```

```
<data>
```

```
<variable>EnvironmentId</variable>
```

```
<value type="QByteArray">{eefd286a-512b-4646-a68b-0e52c907dc78}</value>
```

```
</data>
```

```
<data>
```

```
<variable>ProjectExplorer.Project.ActiveTarget</variable>
```



```

<value type="int">0</value>
</data>
<data>
<variable>ProjectExplorer.Project.EditorSettings</variable>
<valuemap type="QVariantMap">
<value type="bool" key="EditorConfiguration.AutoIndent">true</value>
<value type="bool" key="EditorConfiguration.AutoSpacesForTabs">false</value>
<value type="bool" key="EditorConfiguration.CamelCaseNavigation">true</value>
<valuemap type="QVariantMap" key="EditorConfiguration.CodeStyle.0">
<value type="QString" key="language">Cpp</value>
<valuemap type="QVariantMap" key="value">
<value type="QByteArray" key="CurrentPreferences">CppGlobal</value>
</valuemap>
</valuemap>
<valuemap type="QVariantMap" key="EditorConfiguration.CodeStyle.1">
<value type="QString" key="language">QmlJS</value>
<valuemap type="QVariantMap" key="value">
<value type="QByteArray" key="CurrentPreferences">QmlJSGlobal</value>
</valuemap>
</valuemap>
<value type="int" key="EditorConfiguration.CodeStyle.Count">2</value>
<value type="QByteArray" key="EditorConfiguration.Codec">UTF-8</value>
<value type="bool" key="EditorConfiguration.ConstrainToolTips">false</value>
<value type="int" key="EditorConfiguration.IndentSize">4</value>
<value type="bool" key="EditorConfiguration.KeyboardToolTips">false</value>
<value type="int" key="EditorConfiguration.MarginColumn">80</value>
<value type="bool" key="EditorConfiguration.MouseHiding">true</value>
<value type="bool" key="EditorConfiguration.MouseNavigation">true</value>
<value type="int" key="EditorConfiguration.PaddingMode">1</value>
<value type="bool" key="EditorConfiguration.PreferSingleLineComments">false</value>
<value type="bool" key="EditorConfiguration.ScrollWheelZooming">true</value>
<value type="bool" key="EditorConfiguration.ShowMargin">false</value>
<value type="int" key="EditorConfiguration.SmartBackspaceBehavior">0</value>
<value type="bool" key="EditorConfiguration.SmartSelectionChanging">true</value>
<value type="bool" key="EditorConfiguration.SpacesForTabs">true</value>
<value type="int" key="EditorConfiguration.TabKeyBehavior">0</value>
<value type="int" key="EditorConfiguration.TabSize">8</value>
<value type="bool" key="EditorConfiguration.UseGlobal">true</value>
<value type="bool" key="EditorConfiguration.UseIndenter">false</value>
<value type="int" key="EditorConfiguration.Utf8BomBehavior">1</value>
<value type="bool" key="EditorConfiguration.addFinalNewLine">true</value>
<value type="bool" key="EditorConfiguration.cleanIndentation">true</value>
<value type="bool" key="EditorConfiguration.cleanWhitespace">true</value>
<value type="QString" key="EditorConfiguration.ignoreFileTypes">*.md, *.MD, Makefile</value>
<value type="bool" key="EditorConfiguration.inEntireDocument">false</value>
<value type="bool" key="EditorConfiguration.skipTrailingWhitespace">true</value>
</valuemap>
</data>
<data>
<variable>ProjectExplorer.Project.PluginSettings</variable>
<valuemap type="QVariantMap">
<valuemap type="QVariantMap" key="AutoTest.ActiveFrameworks">
<value type="bool" key="AutoTest.Framework.Boost">true</value>
<value type="bool" key="AutoTest.Framework.CTest">false</value>
<value type="bool" key="AutoTest.Framework.Catch">true</value>
<value type="bool" key="AutoTest.Framework.GTest">true</value>
<value type="bool" key="AutoTest.Framework.QtQuickTest">true</value>
<value type="bool" key="AutoTest.Framework.QtTest">true</value>
</valuemap>
<valuemap type="QVariantMap" key="AutoTest.CheckStates"/>

```

```

<value type="int" key="AutoTest.RunAfterBuild">0</value>
<value type="bool" key="AutoTest.UseGlobal">true</value>
<valuelist type="QVariantList" key="ClangCodeModel.CustomCommandLineKey"/>
<value type="bool" key="ClangCodeModel.UseGlobalConfig">true</value>
<value type="QString" key="ClangCodeModel.WarningConfigId">Builtin.BuildSystem</value>
<valuemap type="QVariantMap" key="ClangTools">
  <value type="bool" key="ClangTools.AnalyzeOpenFiles">true</value>
  <value type="bool" key="ClangTools.BuildBeforeAnalysis">true</value>
  <value type="QString" key="ClangTools.DiagnosticConfig">Builtin.DefaultTidyAndClazy</value>
  <value type="int" key="ClangTools.ParallelJobs">1</value>
  <valuelist type="QVariantList" key="ClangTools.SelectedDirs"/>
  <valuelist type="QVariantList" key="ClangTools.SelectedFiles"/>
  <valuelist type="QVariantList" key="ClangTools.SuppressedDiagnostics"/>
  <value type="bool" key="ClangTools.UseGlobalSettings">true</value>
</valuemap>
</valuemap>
</data>
<data>
  <variable>ProjectExplorer.Project.Target.0</variable>
  <valuemap type="QVariantMap">
    <value type="QString" key="DeviceType">Desktop</value>
    <value type="QString" key="ProjectExplorer.ProjectConfiguration.DefaultDisplayName">qt5</value>
    <value type="QString" key="ProjectExplorer.ProjectConfiguration.DisplayName">qt5</value>
    <value type="QString" key="ProjectExplorer.ProjectConfiguration.Id">{d84de845-86d7-4330-b435-516f5fdd6885}</value>
    <value type="int" key="ProjectExplorer.Target.ActiveBuildConfiguration">0</value>
    <value type="int" key="ProjectExplorer.Target.ActiveDeployConfiguration">0</value>
    <value type="int" key="ProjectExplorer.Target.ActiveRunConfiguration">0</value>
    <valuemap type="QVariantMap" key="ProjectExplorer.Target.BuildConfiguration.0">
      <value type="int" key="EnableQmlDebugging">0</value>
      <value type="QString" key="ProjectExplorer.BuildConfiguration.BuildDirectory">/home/pouser/build-rostik_qt_admin-qt5-Debug</value>
      <value type="QString"
key="ProjectExplorer.BuildConfiguration.BuildDirectory.shadowDir">/home/pouser/build-rostik_qt_admin-qt5-Debug</value>
      <valuemap type="QVariantMap" key="ProjectExplorer.BuildConfiguration.BuildStepList.0">
        <valuemap type="QVariantMap" key="ProjectExplorer.BuildStepList.Step.0">
          <value type="bool" key="ProjectExplorer.BuildStep.Enabled">true</value>
          <value type="QString"
key="ProjectExplorer.ProjectConfiguration.Id">QtProjectManager.QMakeBuildStep</value>
          <value type="bool" key="QtProjectManager.QMakeBuildStep.QMakeForced">>false</value>
          <valuelist type="QVariantList" key="QtProjectManager.QMakeBuildStep.SelectedAbis"/>
        </valuemap>
        <valuemap type="QVariantMap" key="ProjectExplorer.BuildStepList.Step.1">
          <value type="bool" key="ProjectExplorer.BuildStep.Enabled">true</value>
          <value type="QString"
key="ProjectExplorer.ProjectConfiguration.Id">Qt4ProjectManager.MakeStep</value>
        </valuemap>
        <value type="int" key="ProjectExplorer.BuildStepList.StepsCount">2</value>
        <value type="QString"
key="ProjectExplorer.ProjectConfiguration.DefaultDisplayName">C6opka</value>
        <value type="QString" key="ProjectExplorer.ProjectConfiguration.DisplayName">C6opka</value>
        <value type="QString"
key="ProjectExplorer.ProjectConfiguration.Id">ProjectExplorer.BuildSteps.Build</value>
      </valuemap>
      <valuemap type="QVariantMap" key="ProjectExplorer.BuildConfiguration.BuildStepList.1">
        <valuemap type="QVariantMap" key="ProjectExplorer.BuildStepList.Step.0">
          <value type="bool" key="ProjectExplorer.BuildStep.Enabled">true</value>
          <value type="QString"
key="ProjectExplorer.ProjectConfiguration.Id">Qt4ProjectManager.MakeStep</value>

```

```

    <value type="QString" key="Qt4ProjectManager.MakeStep.MakeArguments">clean</value>
  </valuemap>
  <value type="int" key="ProjectExplorer.BuildStepList.StepsCount">1</value>
  <value type="QString"
key="ProjectExplorer.ProjectConfiguration.DefaultDisplayName">Очистка</value>
    <value type="QString" key="ProjectExplorer.ProjectConfiguration.DisplayName">Очистка</value>
    <value type="QString"
key="ProjectExplorer.ProjectConfiguration.Id">ProjectExplorer.BuildSteps.Clean</value>
  </valuemap>
  <value type="int" key="ProjectExplorer.BuildConfiguration.BuildStepListCount">2</value>
  <value type="bool" key="ProjectExplorer.BuildConfiguration.ClearSystemEnvironment">>false</value>
  <valuelist type="QVariantList" key="ProjectExplorer.BuildConfiguration.CustomParsers"/>
  <value type="bool" key="ProjectExplorer.BuildConfiguration.ParseStandardOutput">>false</value>
  <valuelist type="QVariantList" key="ProjectExplorer.BuildConfiguration.UserEnvironmentChanges"/>
  <value type="QString" key="ProjectExplorer.ProjectConfiguration.DisplayName">Отладка</value>
  <value type="QString"
key="ProjectExplorer.ProjectConfiguration.Id">Qt4ProjectManager.Qt4BuildConfiguration</value>
    <value type="int" key="Qt4ProjectManager.Qt4BuildConfiguration.BuildConfiguration">2</value>
  </valuemap>
  <valuemap type="QVariantMap" key="ProjectExplorer.Target.BuildConfiguration.1">
    <value type="QString" key="ProjectExplorer.BuildConfiguration.BuildDirectory">/home/pouser/build-
rostik_qt_admin-qt5-Release</value>
    <value type="QString"
key="ProjectExplorer.BuildConfiguration.BuildDirectory.shadowDir">/home/pouser/build-rostik_qt_admin-
qt5-Release</value>
    <valuemap type="QVariantMap" key="ProjectExplorer.BuildConfiguration.BuildStepList.0">
      <valuemap type="QVariantMap" key="ProjectExplorer.BuildStepList.Step.0">
        <value type="bool" key="ProjectExplorer.BuildStep.Enabled">>true</value>
        <value type="QString"
key="ProjectExplorer.ProjectConfiguration.Id">QtProjectManager.QMakeBuildStep</value>
          <value type="bool" key="QtProjectManager.QMakeBuildStep.QMakeForced">>false</value>
          <valuelist type="QVariantList" key="QtProjectManager.QMakeBuildStep.SelectedAbis"/>
        </valuemap>
        <valuemap type="QVariantMap" key="ProjectExplorer.BuildStepList.Step.1">
          <value type="bool" key="ProjectExplorer.BuildStep.Enabled">>true</value>
          <value type="QString"
key="ProjectExplorer.ProjectConfiguration.Id">Qt4ProjectManager.MakeStep</value>
            </valuemap>
            <value type="int" key="ProjectExplorer.BuildStepList.StepsCount">2</value>
            <value type="QString"
key="ProjectExplorer.ProjectConfiguration.DefaultDisplayName">Сборка</value>
              <value type="QString" key="ProjectExplorer.ProjectConfiguration.DisplayName">Сборка</value>
              <value type="QString"
key="ProjectExplorer.ProjectConfiguration.Id">ProjectExplorer.BuildSteps.Build</value>
                </valuemap>
                <valuemap type="QVariantMap" key="ProjectExplorer.BuildConfiguration.BuildStepList.1">
                  <valuemap type="QVariantMap" key="ProjectExplorer.BuildStepList.Step.0">
                    <value type="bool" key="ProjectExplorer.BuildStep.Enabled">>true</value>
                    <value type="QString"
key="ProjectExplorer.ProjectConfiguration.Id">Qt4ProjectManager.MakeStep</value>
                      <value type="QString" key="Qt4ProjectManager.MakeStep.MakeArguments">clean</value>
                    </valuemap>
                    <value type="int" key="ProjectExplorer.BuildStepList.StepsCount">1</value>
                    <value type="QString"
key="ProjectExplorer.ProjectConfiguration.DefaultDisplayName">Очистка</value>
                      <value type="QString" key="ProjectExplorer.ProjectConfiguration.DisplayName">Очистка</value>
                      <value type="QString"
key="ProjectExplorer.ProjectConfiguration.Id">ProjectExplorer.BuildSteps.Clean</value>
                        </valuemap>
                        <value type="int" key="ProjectExplorer.BuildConfiguration.BuildStepListCount">2</value>

```

```

<value type="bool" key="ProjectExplorer.BuildConfiguration.ClearSystemEnvironment">false</value>
<valuelist type="QVariantList" key="ProjectExplorer.BuildConfiguration.CustomParsers"/>
<value type="bool" key="ProjectExplorer.BuildConfiguration.ParseStandardOutput">false</value>
<valuelist type="QVariantList" key="ProjectExplorer.BuildConfiguration.UserEnvironmentChanges"/>
<value type="QString" key="ProjectExplorer.ProjectConfiguration.DisplayName">Выпуск</value>
<value type="QString"
key="ProjectExplorer.ProjectConfiguration.Id">Qt4ProjectManager.Qt4BuildConfiguration</value>
  <value type="int" key="Qt4ProjectManager.Qt4BuildConfiguration.BuildConfiguration">0</value>
  <value type="int" key="QtQuickCompiler">0</value>
</valuemap>
<valuemap type="QVariantMap" key="ProjectExplorer.Target.BuildConfiguration.2">
  <value type="int" key="EnableQmlDebugging">0</value>
  <value type="QString" key="ProjectExplorer.BuildConfiguration.BuildDirectory">/home/pouser/build-
rostik_qt_admin-qt5-Profile</value>
  <value type="QString"
key="ProjectExplorer.BuildConfiguration.BuildDirectory.shadowDir">/home/pouser/build-rostik_qt_admin-
qt5-Profile</value>
    <valuemap type="QVariantMap" key="ProjectExplorer.BuildConfiguration.BuildStepList.0">
      <valuemap type="QVariantMap" key="ProjectExplorer.BuildStepList.Step.0">
        <value type="bool" key="ProjectExplorer.BuildStep.Enabled">true</value>
        <value type="QString"
key="ProjectExplorer.ProjectConfiguration.Id">QtProjectManager.QMakeBuildStep</value>
          <value type="bool" key="QtProjectManager.QMakeBuildStep.QMakeForced">false</value>
          <valuelist type="QVariantList" key="QtProjectManager.QMakeBuildStep.SelectedAbis"/>
          </valuemap>
          <valuemap type="QVariantMap" key="ProjectExplorer.BuildStepList.Step.1">
            <value type="bool" key="ProjectExplorer.BuildStep.Enabled">true</value>
            <value type="QString"
key="ProjectExplorer.ProjectConfiguration.Id">Qt4ProjectManager.MakeStep</value>
              </valuemap>
              <value type="int" key="ProjectExplorer.BuildStepList.StepsCount">2</value>
              <value type="QString"
key="ProjectExplorer.ProjectConfiguration.DefaultDisplayName">Сборка</value>
                <value type="QString" key="ProjectExplorer.ProjectConfiguration.DisplayName">Сборка</value>
                <value type="QString"
key="ProjectExplorer.ProjectConfiguration.Id">ProjectExplorer.BuildSteps.Build</value>
                  </valuemap>
                  <valuemap type="QVariantMap" key="ProjectExplorer.BuildConfiguration.BuildStepList.1">
                    <valuemap type="QVariantMap" key="ProjectExplorer.BuildStepList.Step.0">
                      <value type="bool" key="ProjectExplorer.BuildStep.Enabled">true</value>
                      <value type="QString"
key="ProjectExplorer.ProjectConfiguration.Id">Qt4ProjectManager.MakeStep</value>
                        <value type="QString" key="Qt4ProjectManager.MakeStep.MakeArguments">clean</value>
                        </valuemap>
                        <value type="int" key="ProjectExplorer.BuildStepList.StepsCount">1</value>
                        <value type="QString"
key="ProjectExplorer.ProjectConfiguration.DefaultDisplayName">Очистка</value>
                          <value type="QString" key="ProjectExplorer.ProjectConfiguration.DisplayName">Очистка</value>
                          <value type="QString"
key="ProjectExplorer.ProjectConfiguration.Id">ProjectExplorer.BuildSteps.Clean</value>
                            </valuemap>
                            <value type="int" key="ProjectExplorer.BuildConfiguration.BuildStepListCount">2</value>
                            <value type="bool" key="ProjectExplorer.BuildConfiguration.ClearSystemEnvironment">false</value>
                            <valuelist type="QVariantList" key="ProjectExplorer.BuildConfiguration.CustomParsers"/>
                            <value type="bool" key="ProjectExplorer.BuildConfiguration.ParseStandardOutput">false</value>
                            <valuelist type="QVariantList" key="ProjectExplorer.BuildConfiguration.UserEnvironmentChanges"/>
                            <value type="QString"
key="ProjectExplorer.ProjectConfiguration.DisplayName">Профилирование</value>
                              <value type="QString"
key="ProjectExplorer.ProjectConfiguration.Id">Qt4ProjectManager.Qt4BuildConfiguration</value>

```

```

    <value type="int" key="Qt4ProjectManager.Qt4BuildConfiguration.BuildConfiguration">0</value>
    <value type="int" key="QtQuickCompiler">0</value>
    <value type="int" key="SeparateDebugInfo">0</value>
  </valuemap>
  <value type="int" key="ProjectExplorer.Target.BuildConfigurationCount">3</value>
  <valuemap type="QVariantMap" key="ProjectExplorer.Target.DeployConfiguration.0">
    <valuemap type="QVariantMap" key="ProjectExplorer.BuildConfiguration.BuildStepList.0">
      <value type="int" key="ProjectExplorer.BuildStepList.StepsCount">0</value>
      <value type="QString"
key="ProjectExplorer.ProjectConfiguration.DefaultDisplayName">Развёртывание</value>
      <value type="QString"
key="ProjectExplorer.ProjectConfiguration.DisplayName">Развёртывание</value>
      <value type="QString"
key="ProjectExplorer.ProjectConfiguration.Id">ProjectExplorer.BuildSteps.Deploy</value>
    </valuemap>
    <value type="int" key="ProjectExplorer.BuildConfiguration.BuildStepListCount">1</value>
    <valuemap type="QVariantMap" key="ProjectExplorer.DeployConfiguration.CustomData"/>
    <value type="bool" key="ProjectExplorer.DeployConfiguration.CustomDataEnabled">>false</value>
    <value type="QString"
key="ProjectExplorer.ProjectConfiguration.Id">ProjectExplorer.DefaultDeployConfiguration</value>
  </valuemap>
  <value type="int" key="ProjectExplorer.Target.DeployConfigurationCount">1</value>
  <valuemap type="QVariantMap" key="ProjectExplorer.Target.RunConfiguration.0">
    <value type="bool" key="Analyzer.Perf.Settings.UseGlobalSettings">>true</value>
    <value type="bool" key="Analyzer.QmlProfiler.Settings.UseGlobalSettings">>true</value>
    <value type="bool" key="Analyzer.Valgrind.Settings.UseGlobalSettings">>true</value>
    <valuelist type="QVariantList" key="CustomOutputParsers"/>
    <value type="int" key="PE.EnvironmentAspect.Base">2</value>
    <valuelist type="QVariantList" key="PE.EnvironmentAspect.Changes"/>
    <value type="QString"
key="ProjectExplorer.ProjectConfiguration.Id">Qt4ProjectManager.Qt4RunConfiguration:/home/pouser/rost
ik_qt_admin/rostik_qt_admin.pro</value>
    <value type="QString"
key="ProjectExplorer.RunConfiguration.BuildKey">/home/pouser/rostik_qt_admin/rostik_qt_admin.pro</va
lue>
    <value type="bool" key="RunConfiguration.UseCppDebugger">>false</value>
    <value type="bool" key="RunConfiguration.UseCppDebuggerAuto">>true</value>
    <value type="bool" key="RunConfiguration.UseLibrarySearchPath">>true</value>
    <value type="bool" key="RunConfiguration.UseQmlDebugger">>false</value>
    <value type="bool" key="RunConfiguration.UseQmlDebuggerAuto">>true</value>
    <value type="QString" key="RunConfiguration.WorkingDirectory.default">/home/pouser/build -
rostik_qt_admin-qt5-Debug</value>
  </valuemap>
  <value type="int" key="ProjectExplorer.Target.RunConfigurationCount">1</value>
</valuemap>
</data>
<data>
  <variable>ProjectExplorer.Project.TargetCount</variable>
  <value type="int">1</value>
</data>
<data>
  <variable>ProjectExplorer.Project.Updater.FileVersion</variable>
  <value type="int">22</value>
</data>
<data>
  <variable>Version</variable>
  <value type="int">22</value>
</data>
</qtcreeator>

```

serverwindow.cpp

```
#include "serverwindow.h"
#include "ui_serverwindow.h"
#include "mainwindow.h"
#include "main.h"

bool double_click = false;
QTextBrowser* logs;
int devices_count = 0;
QLabel* server_text;

ServerWindow::ServerWindow(QWidget *parent) :
    QMainWindow(parent),
    ui(new Ui::ServerWindow)
{
    ui->setupUi(this);
    connect(ui->adminPanelButton, &QPushButton::clicked, this,
        &ServerWindow::on_adminPanelButton_clicked);
    logs = ui->textLogs;
    qDebug() << ui->label->fontInfo().family();
    qDebug() << ui->label->fontInfo().pixelSize();
    server_text = ui->label;
    server_text->setText("Сервер работает на порту " + QString::number(*serverPort) +
        "<br>Подключенных устройств: " + QString::number(devices_count));
}

void ServerWindow::on_adminPanelButton_clicked(){
    if(double_click){
        double_click = false;
    }
    else{
        double_click = true;
        MainWindow * w = new MainWindow(this);
        w->show();
    }
}

void ServerWindow::updateDevicesCount(bool is_adding){
    if(is_adding) devices_count++;
    else {
        logs->append("Один из клиентов отключился");
        devices_count--;
    }
    server_text->setText("Сервер работает на порту " + QString::number(*serverPort) +
        "<br>Подключенных устройств: " + QString::number(devices_count));
}

ServerWindow::~ServerWindow()
{
    delete ui;
}
```

serverwindow.h

```
#ifndef SERVERWINDOW_H
#define SERVERWINDOW_H

#include <QMainWindow>
#include <QTextBrowser>
```

```

namespace Ui {
class ServerWindow;
}
extern QTextBrowser* logs;
class ServerWindow : public QMainWindow
{
    Q_OBJECT

public:
    explicit ServerWindow(QWidget *parent = nullptr);
    ~ServerWindow();
    static void updateDevicesCount(bool is_adding);

private slots:
    void on_adminPanelButton_clicked();
private:
    Ui::ServerWindow *ui;

};

```

```

#endif // SERVERWINDOW_H

```

```

serverwindow.ui

```

```

<?xml version="1.0" encoding="UTF-8"?>
<ui version="4.0">
<class>ServerWindow</class>
<widget class="QMainWindow" name="ServerWindow">
<property name="geometry">
<rect>
<x>0</x>
<y>0</y>
<width>800</width>
<height>600</height>
</rect>
</property>
<property name="windowTitle">
<string>ServerWindow</string>
</property>
<widget class="QWidget" name="centralwidget">
<layout class="QGridLayout" name="gridLayout">
<item row="0" column="0">
<widget class="QLabel" name="label">
<property name="font">
<font>
<family>Roboto</family>
<pointsize>12</pointsize>
</font>
</property>
<property name="text">
<string>&lt;html&gt;&lt;head&gt;&lt;body&gt;&lt;p&gt;Сервер работает на порту
1234&lt;/p&gt;&lt;p&gt;Подключенных устройств: 0&lt;/p&gt;&lt;/body&gt;&lt;/html&gt;</string>
</property>
<property name="alignment">
<set>Qt::AlignLeading|Qt::AlignLeft|Qt::AlignTop</set>
</property>
</widget>
</item>

```

```

<item row="0" column="1">
  <widget class="QTextBrowser" name="textLogs"/>
</item>
<item row="1" column="0" colspan="2">
  <widget class="QPushButton" name="adminPanelButton">
    <property name="font">
      <font>
        <family>Roboto</family>
        <pointsize>12</pointsize>
        <weight>50</weight>
        <bold>>false</bold>
        <kerning>>true</kerning>
      </font>
    </property>
    <property name="text">
      <string>Админ-панель</string>
    </property>
  </widget>
</item>
</layout>
</widget>
<widget class="QMenuBar" name="menubar">
  <property name="geometry">
    <rect>
      <x>0</x>
      <y>0</y>
      <width>800</width>
      <height>29</height>
    </rect>
  </property>
</widget>
<widget class="QStatusBar" name="statusbar"/>
</widget>
<resources/>
<connections/>
</ui>

```

style.qss

```

* {
  font-family: "Roboto", Arial, sans-serif;
  font-size: 14px;
  color: #333;
}

MainWindow {
  background-color: #f5f5f5;
}

QPushButton {
  background-color: #007BFF;
  color: white;
  border: none;
  font-size: 14px;
  padding: 10px 20px;
  border-radius: 5px;
}

QPushButton:hover {

```



```

    background-color: #0056b3;
}

QPushButton:pressed{
    background-color: #004080;
}

QLineEdit {
    border: 1px solid #ccc;
    border-radius: 5px;
    padding: 5px;
    background-color: #fff;
}

QLineEdit:focus {
    border-color: #007BFF;
}

QLabel {
    font-size: 14px;
    color: #555;
}

QTableView {
    border: 1px solid #ccc;
    gridline-color: #e0e0e0;
    selection-background-color: #007BFF;
    selection-color: white;
}

QCheckBox {
    spacing: 5px;
}

QCheckBox::indicator {
    width: 18px;
    height: 18px;
}

QTextBrowser{
    border: 1px solid #ccc;
    border-radius: 5px;
    padding: 5px;
    background-color: #fff;
}

```

tcpserver.cpp

```

#include "tcpserver.h"
#include <QDebug>
#include <serverwindow.h>

TcpServer::TcpServer(QObject *parent) : QTcpServer(parent)
{
    connect(this, &QTcpServer::newConnection, this, &TcpServer::onNewConnection);
}

void TcpServer::onNewConnection(){
    QTcpSocket *clientSocket = nextPendingConnection();

```

```

clients.append(clientSocket);
connect(clientSocket, &QTcpSocket::readyRead, this, &TcpServer::onReadyRead);
connect(clientSocket, &QTcpSocket::disconnected, this, &TcpServer::onDisconnect);
qDebug() << "Клиент присоединился: " << clientSocket->peerAddress().toString();
logs->append("Клиент присоединился: " + clientSocket->peerAddress().toString());
ServerWindow::updateDevicesCount(true);
}

void TcpServer::sendToClient(QTcpSocket* pSocket, QByteArray& data, QString type = "data", int code =
0){
    QJsonObject jsonResponse;
    jsonResponse["type"] = type;
    jsonResponse["code"] = code;
    jsonResponse["data"] = QString(data);
    jsonResponse["size"] = data.size() / 1024.0;
    QByteArray jsonData = QJsonDocument(jsonResponse).toJson();
    QByteArray checksum = QCryptographicHash::hash(data, QCryptographicHash::Sha256).toHex();
    jsonResponse["checksum"] = QString(checksum);
    pSocket->write(QJsonDocument(jsonResponse).toJson());
    if (!pSocket->waitForBytesWritten(5000)){
        qDebug() << pSocket->errorString();
        logs->append("Ошибка записи данных в сокет: " + pSocket->errorString());
    };
}

void TcpServer::onDisconnect(){
    ServerWindow::updateDevicesCount(false);
}

QByteArray TcpServer::modelToJson(QSqlTableModel* model){
    QJsonArray jsonArray;
    for (int row = 0; row < model->rowCount(); ++row){
        QJsonObject jsonObject;
        for (int column = 0; column < model->columnCount(); ++column){
            jsonObject[model->headerData(column, Qt::Horizontal).toString()] =
                model->data(model->index(row, column)).toString();
        }
        jsonArray.append(jsonObject);
    }
    QJsonDocument jsonDoc(jsonArray);
    QByteArray arr = jsonDoc.toJson();
    qDebug() << arr;
    return arr;
}

QByteArray TcpServer::queryToJson(QSqlQueryModel* model){
    QJsonArray jsonArray;
    for (int row = 0; row < model->rowCount(); ++row){
        QJsonObject jsonObject;
        for (int column = 0; column < model->columnCount(); ++column){
            jsonObject[model->headerData(column, Qt::Horizontal).toString()] =
                model->data(model->index(row, column)).toString();
        }
        jsonArray.append(jsonObject);
    }
    QJsonDocument jsonDoc(jsonArray);
    qDebug() << "Json готов к отправке";
    QByteArray arr = jsonDoc.toJson();
    qDebug() << arr;
    return arr;
}

```

```

void TcpServer::onReadyRead(){
    QTcpSocket* clientSocket = (QTcpSocket*)sender();
    QByteArray data = clientSocket->readAll();
    QString str = data;
    if (str == "get_devices"){

        qDebug() << "Запрос: получить устройства";
        logs->append("Запрос: получить устройства");

        QSqlTableModel* model = new QSqlTableModel;
        model->setTable("devices");
        model->select();
        QByteArray jsonData = modelToJson(model);

        qDebug() << "Json готов к отправке";
        logs->append("Сформированный Json готов к отправке");

        sendToClient(clientSocket, jsonData);
    } else if (str == "get_danger_devices"){

        qDebug() << "Запрос: получить устройства под угрозой";
        logs->append("Запрос: получить устройства под угрозой");

        QSqlTableModel* model = new QSqlTableModel;
        model->setTable("devices");
        model->setFilter("voltage > voltage_max or voltage < voltage_min or amperage > amperage_max or
temperature > temperature_max");
        model->select();
        QByteArray jsonData = modelToJson(model);

        qDebug() << "Json готов к отправке";
        logs->append("Сформированный Json готов к отправке");

        sendToClient(clientSocket, jsonData);
    } else if (str == "get_devices_history"){

        qDebug() << "Запрос: получить историю аварийных случаев";
        logs->append("Запрос: получить историю аварийных случаев");

        QSqlQueryModel* qmodel = new QSqlQueryModel;
        QSqlQuery query;
        query.prepare("select * from device_emergency_history JOIN devices on devices.id =
device_emergency_history.device_id");
        if (!query.exec()){
            qDebug() << "Ошибка выполнения запроса к БД!";
            logs->append("Ошибка выполнения запроса к БД!");
            qmodel->deleteLater();
            return;
        }
        qDebug() << query.result();
        qmodel->setQuery(query);
        qDebug() << qmodel;
        QByteArray jsonData = queryToJson(qmodel);
        qDebug() << jsonData;

        qDebug() << "Json готов к отправке";
        logs->append("Сформированный Json готов к отправке");

        sendToClient(clientSocket, jsonData);
    }
}

```

```

} else if(str.startsWith("log_")){

    qDebug() << "Запрос: авторизация";
    logs->append("Запрос: авторизация");

    QList<QString> arr = str.split('_');
    QString login = arr.at(1);
    QString password = arr.at(2);
    QSqlQuery query;
    query.prepare("SELECT * FROM users WHERE username = '" + login + "' AND password = '" +
password + "'");
    if (!query.exec()){
        qDebug() << "Ошибка" << query.lastError().text() + " " + query.executedQuery();
        logs->append("Ошибка" + query.lastError().text() + " " + query.executedQuery());
        return;
    }
    else {
        if (query.size() > 0){
            query.next();
            int a = query.value(4).toInt();
            if (a == 2){
                QByteArray jsonData = "true";

                qDebug() << "Json готов к отправке";
                logs->append("Сформированный Json готов к отправке");

                sendToClient(clientSocket, jsonData);
            }
            else {
                QByteArray jsonData = "Вы вошли в систему, однако функционал вашей роли реализован в
другой программе. Это клиентская версия.";

                qDebug() << "Json готов к отправке";
                logs->append("Сформированный Json готов к отправке");

                sendToClient(clientSocket, jsonData, "error", 1);
            }
        }
        else {
            QByteArray jsonData = "false";

            qDebug() << "Json готов к отправке";
            logs->append("Сформированный Json готов к отправке");

            sendToClient(clientSocket, jsonData, "data", 0);
        }
    }
} else if(str.startsWith("filter_devices")){
    qDebug() << "Запрос: получить устройства по фильтру";
    logs->append("Запрос: получить устройства по фильтру");
    QList<QString> arr = str.split('_');
    QString minVoltage, maxVoltage, minAmperage, maxAmperage, minTemperature, maxTemperature;
    int count = 0;
    for (int i = 0; i < arr.size(); i++){
        if (isNumber(arr.at(i))){
            switch(count){
                case 0:
                    minVoltage = arr.at(i);
                    break;
                case 1:

```

```

        maxVoltage = arr.at(i);
        break;
    case 2:
        minAmperage = arr.at(i);
        break;
    case 3:
        maxAmperage = arr.at(i);
        break;
    case 4:
        minTemperature = arr.at(i);
        break;
    case 5:
        maxTemperature = arr.at(i);
        break;
    }
    count++;
}
}
QString filter = "voltage >= " + minVoltage + " and voltage <= " + maxVoltage +
    " and amperage >= " + minAmperage + " and amperage <= " + maxAmperage +
    " and temperature >= " + minTemperature + " and temperature <= " + maxTemperature;
QSqlTableModel* model = new QSqlTableModel;
model->setTable("devices");
model->setFilter(filter);
model->select();
QByteArray jsonData = modelToJson(model);

qDebug() << "Json готов к отправке";
logs->append("Сформированный Json готов к отправке");

sendToClient(clientSocket, jsonData);
} else {
    QByteArray jsonData = "Введен неверный запрос.";
    qDebug() << "Неверный запрос, отправлен ответ";
    logs->append("Получен неверный запрос");
    sendToClient(clientSocket, jsonData, "error", 404);
}
qDebug() << "Полученные данные: " << str;
}

bool TcpServer::isNumber(const QString& str){
    bool ok;
    str.toInt(&ok);
    return ok;
}

void TcpServer::onError(QAbstractSocket::SocketError err){
    qDebug() << "Ошибка: " << err;
    logs->append("Ошибка: " + QString(err));
}

```

tcpserver.h

```

#ifndef TCPSERVER_H
#define TCPSERVER_H

#include <QTcpServer>
#include <QTcpSocket>

```

```

#include <QSqlTableModel>
#include <QJsonArray>
#include <QJsonObject>
#include <QJsonDocument>
#include <QTimer>
#include <QDataStream>
#include <QIODevice>
#include <QSqlQuery>
#include <QtSql>
#include <QCryptographicHash>

class TcpServer : public QTcpServer
{
public:
    TcpServer(QObject *parent = nullptr);
private slots:
    void onNewConnection();
    void onError(QAbstractSocket::SocketError);
    void onReadyRead();
    void get_devices(QTcpSocket*);
    QByteArray modelToJson(QSqlTableModel* model);
    void sendToClient(QTcpSocket* pSocket, QByteArray& data, QString type, int code);
    QByteArray queryToJson(QSqlQueryModel* model);
    void onDisconnect();
    bool isNumber(const QString& str);
private:
    QList<QTcpSocket*> clients;
};

#endif // TCPSERVER_H

```

userwindow.cpp

```

#include "userwindow.h"
#include "ui_userwindow.h"
#include "historywindow.h"
bool double_click = false;
UserWindow::UserWindow(QWidget *parent) :
    QMainWindow(parent),
    ui(new Ui::UserWIndow)
{
    ui->setupUi(this);
    connect(ui->historyUserButton, &QPushButton::clicked, this,
        &UserWindow::on_historyUserButton_clicked);
    QSqlTableModel * model = new QSqlTableModel(this);
    model->setTable("devices");
    model->select();
    model->setEditStrategy(QSqlTableModel::OnManualSubmit);
    ui->devicesUserTable->setModel(model);
    ui->devicesUserTable->setEditTriggers(QAbstractItemView::NoEditTriggers);
}
void UserWindow::on_historyUserButton_clicked(){
    if(double_click){
        double_click = false;
    }
    else{
        double_click = true;
        HistoryWindow * w = new HistoryWindow(this);
        w->show();
    }
}

```

```

    }
}
UserWindow::~UserWindow()
{
    delete ui;
}

```

userwindow.h

```

#ifndef USERWINDOW_H
#define USERWINDOW_H

#include <QMainWindow>
#include <QSqlQueryModel>
#include <QSqlQuery>
#include <QSqlTableModel>
#include <QMessageBox>
#include <QDebug>
namespace Ui {
class UserWIndow;
}

class UserWindow : public QMainWindow
{
    Q_OBJECT

public:
    explicit UserWindow(QWidget *parent = nullptr);
    ~UserWindow();
private slots:
    void on_historyUserButton_clicked();
private:
    Ui::UserWIndow *ui;
};

#endif // USERWINDOW_H

```

userwindow.ui

```

<?xml version="1.0" encoding="UTF-8"?>
<ui version="4.0">
<class>UserWIndow</class>
<widget class="QMainWindow" name="UserWIndow">
<property name="geometry">
<rect>
<x>0</x>
<y>0</y>
<width>800</width>
<height>454</height>
</rect>
</property>
<property name="windowTitle">
<string>UserWindow</string>
</property>
<widget class="QWidget" name="centralwidget">
<layout class="QVBoxLayout" name="verticalLayout">
<item>
<widget class="QLabel" name="label">

```

```

        <property name="text">
            <string>Добро пожаловать!</string>
        </property>
        <property name="alignment">
            <set>Qt::AlignCenter</set>
        </property>
    </widget>
</item>
<item>
    <widget class="QTableView" name="devicesUserTable"/>
</item>
<item>
    <widget class="QPushButton" name="historyUserButton">
        <property name="text">
            <string>История аварийных случаев</string>
        </property>
    </widget>
</item>
</layout>
</widget>
<widget class="QMenuBar" name="menubar">
    <property name="geometry">
        <rect>
            <x>0</x>
            <y>0</y>
            <width>800</width>
            <height>29</height>
        </rect>
    </property>
</widget>
<widget class="QStatusBar" name="statusbar"/>
</widget>
<resources/>
<connections/>
</ui>

```

enteripwindow.cpp

```

#include "enteripwindow.h"
#include "ui_enteripwindow.h"
#include "main.h"
#include "mainwindow.h"

bool double_click_ip = false;

EnterIpWindow::EnterIpWindow(QWidget *parent) :
    QMainWindow(parent),
    ui(new Ui::EnterIpWindow)
{
    ui->setupUi(this);
    connect(ui->connectPushButton, &QPushButton::clicked, this,
            &EnterIpWindow::on_connectPushButton_clicked);

    setWindowFlags(Qt::Dialog);
}

EnterIpWindow::~EnterIpWindow()
{
    delete ui;
}

```



```

bool EnterIpWindow::isNumber(const QString& str){
    bool ok;
    str.toInt(&ok);
    return ok;
}

void EnterIpWindow::on_connectPushButton_clicked()
{
    if(double_click_ip){
        double_click_ip = false;
    }
    else {
        double_click_ip = true;
        QString ip = ui->ipLineEdit->text();
        int port = 0;
        if(isNumber(ui->portLineEdit->text())){
            port = ui->portLineEdit->text().toInt();
        }
        else {
            port = 1234;
        }
        //client->error = false;
        try{
            client = new TcpClient(ip, port);
            client->checkConnection();
        }
        catch (...){
            client->error = true;
        }
        if (client->error){
            QMessageBox::critical(this, "Не удалось подключиться", "Введите корректный адрес и порт.");
            client->error = false;
        }
        else{
            MainWindow * w = new MainWindow();
            w->show();
            this->close();
        }
    }
}

```

enteripwindow.h

```

#ifndef ENTERIPWINDOW_H
#define ENTERIPWINDOW_H

#include <QMainWindow>
#include <QMessageBox>

namespace Ui {
class EnterIpWindow;
}

class EnterIpWindow : public QMainWindow
{
    Q_OBJECT

```

```
public:
    explicit EnterIpWindow(QWidget *parent = nullptr);
    ~EnterIpWindow();
```

```
private slots:
    void on_connectPushButton_clicked();
    bool isNumber(const QString& str);
```

```
private:
    Ui::EnterIpWindow *ui;
};
```

```
#endif // ENTERIPWINDOW_H
```

enteripwindow.ui

```
<?xml version="1.0" encoding="UTF-8"?>
<ui version="4.0">
    <class>EnterIpWindow</class>
    <widget class="QMainWindow" name="EnterIpWindow">
        <property name="geometry">
            <rect>
                <x>0</x>
                <y>0</y>
                <width>294</width>
                <height>133</height>
            </rect>
        </property>
        <property name="windowTitle">
            <string>Main Window</string>
        </property>
        <widget class="QWidget" name="centralwidget">
            <layout class="QGridLayout" name="gridLayout">
                <item row="0" column="0">
                    <widget class="QLineEdit" name="ipLineEdit">
                        <property name="placeholderText">
                            <string>IP-Адрес</string>
                        </property>
                    </widget>
                </item>
                <item row="0" column="2">
                    <widget class="QLineEdit" name="portLineEdit">
                        <property name="placeholderText">
                            <string>Порт</string>
                        </property>
                    </widget>
                </item>
                <item row="1" column="0" colspan="3">
                    <widget class="QPushButton" name="connectPushButton">
                        <property name="text">
                            <string>Подключиться</string>
                        </property>
                    </widget>
                </item>
            </layout>
        </widget>
        <widget class="QMenuBar" name="menubar">
            <property name="geometry">
                <rect>
```

```

        <x>0</x>
        <y>0</y>
        <width>294</width>
        <height>29</height>
    </rect>
</property>
</widget>
<widget class="QStatusBar" name="statusbar"/>
</widget>
<resources/>
<connections/>
</ui>

```

graphwindow.cpp

```

#include "graphwindow.h"
#include "ui_graphwindow.h"
#include "main.h"

```

```

GraphWindow::GraphWindow(QWidget *parent) :
    QMainWindow(parent),
    ui(new Ui::GraphWindow)
{
    ui->setupUi(this);
    customPlot = new QCustomPlot(this);
    setCentralWidget(customPlot);
    getDevices();
    QTimer *timer = new QTimer(this);
    QObject::connect(timer, &QTimer::timeout, this, &GraphWindow::getDevices);
    timer->start(5000);
}

```

```

void GraphWindow::getDevices() {
    QList<JsonObject> jsonArray = client->sendDataList("get_devices");
    int s = jsonArray.size();
    customPlot->clearGraphs();
    customPlot->clearItems();
    QVector<double> x(s),
        yVoltage(s),
        yAmperage(s),
        yTemperature(s);
    for (int i = 0; i < s; i++) {
        x[i] = i + 1;
        QString str = jsonArray[i]["voltage"].toString();
        yVoltage[i] = jsonArray[i]["voltage"].toString().toInt();
        yAmperage[i] = jsonArray[i]["amperage"].toString().toInt();
        yTemperature[i] = jsonArray[i]["temperature"].toString().toInt();
    }
    customPlot->addGraph();
    customPlot->graph(0)->setData(x, yVoltage);
    customPlot->graph(0)->setPen(QPen(Qt::red));
    customPlot->graph(0)->setName("Напряжение");

    customPlot->addGraph();
    customPlot->graph(1)->setData(x, yAmperage);
    customPlot->graph(1)->setPen(QPen(Qt::green));
    customPlot->graph(1)->setName("Ток");
}

```

```

customPlot->addGraph();
customPlot->graph(2)->setData(x, yTemperature);
customPlot->graph(2)->setPen(QPen(Qt::blue));
customPlot->graph(2)->setName("Температура");

customPlot->xAxis->setLabel("№ Устройства");
customPlot->yAxis->setLabel("Значения");
customPlot->xAxis->setRange(1, s);
customPlot->yAxis->setRange(0, 300);

customPlot->legend->setVisible(true);
customPlot->replot();
customPlot->show();
}
GraphWindow::~~GraphWindow()
{
    delete ui;
}

```

graphwindow.h

```

#ifndef GRAPHWINDOW_H
#define GRAPHWINDOW_H

#include <QMainWindow>
#include "/home/pouser/Desktops/Desktop1/qcustomplot/qcustomplot.h"
namespace Ui {
class GraphWindow;
}

class GraphWindow : public QMainWindow
{
    Q_OBJECT

public:
    explicit GraphWindow(QWidget *parent = nullptr);
    ~GraphWindow();

private slots:
    void getDevices();

private:
    Ui::GraphWindow *ui;
    QCustomPlot *customPlot;
};

#endif // GRAPHWINDOW_H

```

graphwindow.ui

```

<ui version="4.0">
<author/>
<comment/>
<exportmacro/>
<class>GraphWindow</class>
<widget name="GraphWindow" class="QMainWindow">
<property name="geometry">
<rect>

```

```

<x>0</x>
<y>0</y>
<width>800</width>
<height>600</height>
</rect>
</property>
<property name="windowTitle">
<string>Main Window</string>
</property>
<widget name="menubar" class="QMenuBar"/>
<widget name="centralwidget" class="QWidget"/>
<widget name="statusbar" class="QStatusBar"/>
</widget>
<pixmapfunction/>
<connections/>
</ui>

```

historywindow.cpp

```

#include "historywindow.h"
#include "ui_historywindow.h"
#include "main.h"

```

```

HistoryWindow::HistoryWindow(QWidget *parent) :
    QMainWindow(parent),
    ui(new Ui::HistoryWindow)
{
    ui->setupUi(this);

    model = new QStandardItemModel(this);
    ui->historyTable->setModel(model);
    ui->historyTable->setEditTriggers(QAbstractItemView::NoEditTriggers);
    getHistory();
    QTimer *timer = new QTimer(this);
    QObject::connect(timer, &QTimer::timeout, this, &HistoryWindow::getHistory);
    timer->start(5000);
}

void HistoryWindow::getHistory() {
    QList<QJsonObject> jsonArray = client->sendDataList("get_devices_history");
    model->clear();
    model->setColumnCount(13);
    model->setHorizontalHeaderLabels(QStringList() << "Дата аварии" << "№" << "Название" <<
"Включено"
                                << "Напряжение" << "Мин. Напряжение" << "Макс. Напряжение"
                                << "Ток" << "Макс. Ток" << "Температура" << "Макс. Температура"
                                << "Емкость" << "№ Расположения");
    for(QJsonObject jobj : jsonArray) {
        QList<QStandardItem*> rowItems;
        rowItems.append(new QStandardItem(jobj["creation_time"].toString()));
        rowItems.append(new QStandardItem(jobj["id"].toString()));
        rowItems.append(new QStandardItem(jobj["name"].toString()));
        rowItems.append(new QStandardItem(jobj["turned"].toString()));
        rowItems.append(new QStandardItem(jobj["voltage"].toString()));
        rowItems.append(new QStandardItem(jobj["voltage_min"].toString()));
        rowItems.append(new QStandardItem(jobj["voltage_max"].toString()));
        rowItems.append(new QStandardItem(jobj["amperage"].toString()));
        rowItems.append(new QStandardItem(jobj["amperage_max"].toString()));
        rowItems.append(new QStandardItem(jobj["temperature"].toString()));
    }
}

```

```

        rowItems.append(new QStandardItem(jobj["temperature_max"].toString()));
        rowItems.append(new QStandardItem(jobj["capacity"].toString()));
        rowItems.append(new QStandardItem(jobj["placement_id"].toString()));
        model->appendRow(rowItems);
    }
}

```

```

HistoryWindow::~HistoryWindow()
{
    delete ui;
}

```

historywindow.h

```

#ifndef HISTORYWINDOW_H
#define HISTORYWINDOW_H

#include <QMainWindow>
#include <QMessageBox>
#include <QTimer>
#include <QJsonObject>
#include <QList>
#include <QStandardItem>

namespace Ui {
class HistoryWindow;
}

class HistoryWindow : public QMainWindow
{
    Q_OBJECT

public:
    explicit HistoryWindow(QWidget *parent = nullptr);
    ~HistoryWindow();
private slots:
    void getHistory();
private:
    QStandardItemModel *model;
    Ui::HistoryWindow *ui;
};

#endif // HISTORYWINDOW_H

```

historywindow.ui

```

<?xml version="1.0" encoding="UTF-8"?>
<ui version="4.0">
<class>HistoryWindow</class>
<widget class="QMainWindow" name="HistoryWindow">
<property name="geometry">
<rect>
<x>0</x>
<y>0</y>
<width>800</width>
<height>300</height>
</rect>
</property>

```

```

<property name="windowTitle">
<string>HistoryWindow</string>
</property>
<widget class="QWidget" name="centralwidget">
<layout class="QGridLayout" name="gridLayout">
<item row="0" column="0">
<widget class="QTableView" name="historyTable"/>
</item>
</layout>
</widget>
<widget class="QMenuBar" name="menubar">
<property name="geometry">
<rect>
<x>0</x>
<y>0</y>
<width>800</width>
<height>29</height>
</rect>
</property>
</widget>
<widget class="QStatusBar" name="statusbar"/>
</widget>
<resources/>
<connections/>
</ui>

```

main.cpp

```

#include "main.h"
#include "mainwindow.h"
#include <QApplication>
#include <QTableView>
#include "tcpclient.h"
#include "enteripwindow.h"

```

```

TcpClient* client = new TcpClient("121.11.11.1", 1234);

```

```

//bool createConnection(){
//  QSqlDatabase db = QSqlDatabase::addDatabase("QPSQL");
//  db.setDatabaseName("microcontrollers_study");
//  db.setUserName("admin");
//  db.setHostName("localhost");
//  db.setPassword("admin");
//  if(!db.open()){
//    qDebug() << "Ошибка подключения к БД: " << db.lastError();
//    return false;
//  }
//  return true;
//}
int main(int argc, char *argv[])
{
  QApplication a(argc, argv);
  //createConnection();
  QDir d;
  QFile file(d.absolutePath() + "../rostik_qt/style.qss");
  if (file.open(QFile::ReadOnly | QFile::Text)){
    a.setStyleSheet(QLatin1String(file.readAll()));
    qDebug() << "Стили загружены.";
  }
}

```

```

else{
    qDebug() << file.errorString();
    qDebug() << file.fileName();
}
if (client->error){
    EnterIpWindow w;
    w.show();
    return a.exec();
}
MainWindow w;
w.show();
return a.exec();
}

```

main.h

```

#ifndef MAIN_H
#define MAIN_H
#include <QTcpSocket>
#include <tcpclient.h>
#include <QDir>
#include <QFile>
extern TcpClient* client;
#endif // MAIN_H

```

mainwindow.cpp

```

#include "mainwindow.h"
#include "userwindow.h"
#include "ui_mainwindow.h"
#include "main.h"

MainWindow::MainWindow(QWidget *parent)
: QMainWindow(parent)
, ui(new Ui::MainWindow)
{
    ui->setupUi(this);
    connect(ui->auth_button, &QPushButton::clicked, this,
        &MainWindow::on_authButton_clicked);
}

MainWindow::~MainWindow()
{
    delete ui;
}

void MainWindow::on_authButton_clicked() {
    QString login = ui->login_tb->text();
    QString password = QCryptographicHash::hash(ui->password_tb->text().toUtf8(),
        QCryptographicHash::Sha256).toHex();
    QString str = "log_" + login + "_" + password;
    QString str_res = client->sendDataString(str.toUtf8());
    QList<QString> arr = str_res.split('_');

    if (arr[0] == "DATA" && arr[1] == "true") {
        UserWindow * w = new UserWindow();
        w->show();
        this->close();
    }
}

```



```

    }
    else if (arr[0] == "DATA" && arr[1] == "false"){
        QMessageBox::information(this, "Не найдено", "Введен неверный логин или пароль.");
    }
    else {
        QMessageBox::information(this, "Ошибка", arr[1]);
    }
}

```

mainwindow.h

```

#ifndef MAINWINDOW_H
#define MAINWINDOW_H

#include <QMainWindow>
#include <QDebug>
#include <QMessageBox>
#include <QList>

QT_BEGIN_NAMESPACE
namespace Ui { class MainWindow; }
QT_END_NAMESPACE

class MainWindow : public QMainWindow
{
    Q_OBJECT

public:
    MainWindow(QWidget *parent = nullptr);
    ~MainWindow();
private slots:
    void on_authButton_clicked();

private:
    Ui::MainWindow *ui;
};
#endif // MAINWINDOW_H

```

mainwindow.ui

```

<?xml version="1.0" encoding="UTF-8"?>
<ui version="4.0">
<class>MainWindow</class>
<widget class="QMainWindow" name="MainWindow">
<property name="geometry">
<rect>
<x>0</x>
<y>0</y>
<width>657</width>
<height>432</height>
</rect>
</property>
<property name="windowTitle">
<string>Main Window</string>
</property>
<widget class="QWidget" name="centralwidget">
<layout class="QGridLayout" name="gridLayout">

```

```

<item row="0" column="2">
<spacer name="verticalSpacer_2">
<property name="orientation">
<enum>Qt::Vertical</enum>
</property>
<property name="sizeHint" stdset="0">
<size>
<width>20</width>
<height>42</height>
</size>
</property>
</spacer>
</item>
<item row="1" column="0">
<spacer name="horizontalSpacer_5">
<property name="orientation">
<enum>Qt::Horizontal</enum>
</property>
<property name="sizeHint" stdset="0">
<size>
<width>93</width>
<height>20</height>
</size>
</property>
</spacer>
</item>
<item row="1" column="1" colspan="3">
<widget class="QLabel" name="label">
<property name="font">
<font>
<pointsize>10</pointsize>
</font>
</property>
<property name="text">
<string>Добро пожаловать в программу учета электронных приборов!</string>
</property>
</widget>
</item>
<item row="1" column="4">
<spacer name="horizontalSpacer_6">
<property name="orientation">
<enum>Qt::Horizontal</enum>
</property>
<property name="sizeHint" stdset="0">
<size>
<width>93</width>
<height>20</height>
</size>
</property>
</spacer>
</item>
<item row="2" column="2">
<spacer name="verticalSpacer_3">
<property name="orientation">
<enum>Qt::Vertical</enum>
</property>
<property name="sizeHint" stdset="0">
<size>
<width>20</width>
<height>42</height>

```

```

</size>
</property>
</spacer>
</item>
<item row="3" column="1">
<spacer name="horizontalSpacer_7">
<property name="orientation">
<enum>Qt::Horizontal</enum>
</property>
<property name="sizeHint" stdset="0">
<size>
<width>93</width>
<height>20</height>
</size>
</property>
</spacer>
</item>
<item row="3" column="2">
<widget class="QLabel" name="label_2">
<property name="text">
<string>Введите данные для авторизации:</string>
</property>
</widget>
</item>
<item row="3" column="3">
<spacer name="horizontalSpacer_8">
<property name="orientation">
<enum>Qt::Horizontal</enum>
</property>
<property name="sizeHint" stdset="0">
<size>
<width>93</width>
<height>20</height>
</size>
</property>
</spacer>
</item>
<item row="4" column="1">
<spacer name="horizontalSpacer">
<property name="orientation">
<enum>Qt::Horizontal</enum>
</property>
<property name="sizeHint" stdset="0">
<size>
<width>93</width>
<height>20</height>
</size>
</property>
</spacer>
</item>
<item row="4" column="2">
<widget class="QLineEdit" name="login_tb">
<property name="placeholderText">
<string>Логин</string>
</property>
</widget>
</item>
<item row="4" column="3">
<spacer name="horizontalSpacer_3">
<property name="orientation">

```

```

    <enum>Qt::Horizontal</enum>
  </property>
  <property name="sizeHint" stdset="0">
    <size>
      <width>93</width>
      <height>20</height>
    </size>
  </property>
</spacer>
</item>
<item row="5" column="1">
  <spacer name="horizontalSpacer_2">
    <property name="orientation">
      <enum>Qt::Horizontal</enum>
    </property>
    <property name="sizeHint" stdset="0">
      <size>
        <width>93</width>
        <height>20</height>
      </size>
    </property>
  </spacer>
</item>
<item row="5" column="2">
  <widget class="QLineEdit" name="password_tb">
    <property name="placeholderText">
      <string>Пароль</string>
    </property>
  </widget>
</item>
<item row="5" column="3">
  <spacer name="horizontalSpacer_4">
    <property name="orientation">
      <enum>Qt::Horizontal</enum>
    </property>
    <property name="sizeHint" stdset="0">
      <size>
        <width>93</width>
        <height>20</height>
      </size>
    </property>
  </spacer>
</item>
<item row="6" column="2">
  <widget class="QPushButton" name="auth_button">
    <property name="text">
      <string>Войти</string>
    </property>
  </widget>
</item>
<item row="7" column="2">
  <spacer name="verticalSpacer_4">
    <property name="orientation">
      <enum>Qt::Vertical</enum>
    </property>
    <property name="sizeHint" stdset="0">
      <size>
        <width>20</width>
        <height>42</height>
      </size>
    </property>
  </spacer>
</item>

```

```

    </property>
  </spacer>
</item>
<item row="8" column="2">
  <spacer name="verticalSpacer">
    <property name="orientation">
      <enum>Qt::Vertical</enum>
    </property>
    <property name="sizeHint" stdset="0">
      <size>
        <width>20</width>
        <height>42</height>
      </size>
    </property>
  </spacer>
</item>
</layout>
</widget>
<widget class="QMenuBar" name="menubar">
  <property name="geometry">
    <rect>
      <x>0</x>
      <y>0</y>
      <width>657</width>
      <height>29</height>
    </rect>
  </property>
</widget>
<widget class="QStatusBar" name="statusbar"/>
</widget>
<resources/>
<connections/>
</ui>

```

rostik_qt.pro

```

QT += core gui network
greaterThan(QT_MAJOR_VERSION, 4): QT += widgets

```

```

CONFIG += c++11

```

```

# You can make your code fail to compile if it uses deprecated APIs.
# In order to do so, uncomment the following line.
#DEFINES += QT_DISABLE_DEPRECATED_BEFORE=0x060000 # disables all the APIs
deprecated before Qt 6.0.0

```

```

SOURCES += \
  ../Desktop/qcustomplot/examples/text-document-integration/qcpdocumentobject.cpp \
  ../Desktop/qcustomplot/qcustomplot.cpp \
  enteripwindow.cpp \
  graphwindow.cpp \
  historywindow.cpp \
  main.cpp \
  mainwindow.cpp \
  tcpclient.cpp \
  userwindow.cpp \
  workwindow.cpp

```

```

HEADERS += \

```

```

../Desktop/qcustomplot/examples/text-document-integration/qcpdocumentobject.h \
../Desktop/qcustomplot/qcustomplot.h \
enteripwindow.h \
graphwindow.h \
historywindow.h \
main.h \
mainwindow.h \
tcpclient.h \
userwindow.h \
workwindow.h

```

```

FORMS += \
    enteripwindow.ui \
    graphwindow.ui \
    historywindow.ui \
    mainwindow.ui \
    userwindow.ui \
    workwindow.ui

```

```

# Default rules for deployment.
qnx: target.path = /tmp/${TARGET}/bin
else: unix:!android: target.path = /opt/${TARGET}/bin
!isEmpty(target.path): INSTALLS += target

```

```

rostik_qt.pro.user

```

```

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE QtCreatorProject>
<!-- Written by QtCreator 6.0.2, 2024-09-16T14:44:59. -->
<qtcreator>
<data>
<variable>EnvironmentId</variable>
<value type="QByteArray">{eefd286a-512b-4646-a68b-0e52c907dc78}</value>
</data>
<data>
<variable>ProjectExplorer.Project.ActiveTarget</variable>
<value type="int">0</value>
</data>
<data>
<variable>ProjectExplorer.Project.EditorSettings</variable>
<valuemap type="QVariantMap">
<value type="bool" key="EditorConfiguration.AutoIndent">true</value>
<value type="bool" key="EditorConfiguration.AutoSpacesForTabs">false</value>
<value type="bool" key="EditorConfiguration.CamelCaseNavigation">true</value>
<valuemap type="QVariantMap" key="EditorConfiguration.CodeStyle.0">
<value type="QString" key="language">Cpp</value>
<valuemap type="QVariantMap" key="value">
<value type="QByteArray" key="CurrentPreferences">CppGlobal</value>
</valuemap>
</valuemap>
<valuemap type="QVariantMap" key="EditorConfiguration.CodeStyle.1">
<value type="QString" key="language">QmlJS</value>
<valuemap type="QVariantMap" key="value">
<value type="QByteArray" key="CurrentPreferences">QmlJSGlobal</value>
</valuemap>
</valuemap>
<value type="int" key="EditorConfiguration.CodeStyle.Count">2</value>
<value type="QByteArray" key="EditorConfiguration.Codec">UTF-8</value>
<value type="bool" key="EditorConfiguration.ConstrainTooltips">false</value>

```

```

<value type="int" key="EditorConfiguration.IndentSize">4</value>
<value type="bool" key="EditorConfiguration.KeyboardToolTips">false</value>
<value type="int" key="EditorConfiguration.MarginColumn">80</value>
<value type="bool" key="EditorConfiguration.MouseHiding">true</value>
<value type="bool" key="EditorConfiguration.MouseNavigation">true</value>
<value type="int" key="EditorConfiguration.PaddingMode">1</value>
<value type="bool" key="EditorConfiguration.PreferSingleLineComments">false</value>
<value type="bool" key="EditorConfiguration.ScrollWheelZooming">true</value>
<value type="bool" key="EditorConfiguration.ShowMargin">false</value>
<value type="int" key="EditorConfiguration.SmartBackspaceBehavior">0</value>
<value type="bool" key="EditorConfiguration.SmartSelectionChanging">true</value>
<value type="bool" key="EditorConfiguration.SpacesForTabs">true</value>
<value type="int" key="EditorConfiguration.TabKeyBehavior">0</value>
<value type="int" key="EditorConfiguration.TabSize">8</value>
<value type="bool" key="EditorConfiguration.UseGlobal">true</value>
<value type="bool" key="EditorConfiguration.UseIndenter">false</value>
<value type="int" key="EditorConfiguration.Utf8BomBehavior">1</value>
<value type="bool" key="EditorConfiguration.addFinalNewLine">true</value>
<value type="bool" key="EditorConfiguration.cleanIndentation">true</value>
<value type="bool" key="EditorConfiguration.cleanWhitespace">true</value>
<value type="QString" key="EditorConfiguration.ignoreFileTypes">*.md, *.MD, Makefile</value>
<value type="bool" key="EditorConfiguration.inEntireDocument">false</value>
<value type="bool" key="EditorConfiguration.skipTrailingWhitespace">true</value>
</valuemap>
</data>
<data>
<variable>ProjectExplorer.Project.PluginSettings</variable>
<valuemap type="QVariantMap">
<valuemap type="QVariantMap" key="AutoTest.ActiveFrameworks">
<value type="bool" key="AutoTest.Framework.Boost">true</value>
<value type="bool" key="AutoTest.Framework.CTest">false</value>
<value type="bool" key="AutoTest.Framework.Catch">true</value>
<value type="bool" key="AutoTest.Framework.GTest">true</value>
<value type="bool" key="AutoTest.Framework.QtQuickTest">true</value>
<value type="bool" key="AutoTest.Framework.QtTest">true</value>
</valuemap>
<valuemap type="QVariantMap" key="AutoTest.CheckStates"/>
<value type="int" key="AutoTest.RunAfterBuild">0</value>
<value type="bool" key="AutoTest.UseGlobal">true</value>
<valuelist type="QVariantList" key="ClangCodeModel.CustomCommandLineKey"/>
<value type="bool" key="ClangCodeModel.UseGlobalConfig">true</value>
<value type="QString" key="ClangCodeModel.WarningConfigId">Builtin.BuildSystem</value>
<valuemap type="QVariantMap" key="ClangTools">
<value type="bool" key="ClangTools.AnalyzeOpenFiles">true</value>
<value type="bool" key="ClangTools.BuildBeforeAnalysis">true</value>
<value type="QString" key="ClangTools.DiagnosticConfig">Builtin.DefaultTidyAndClazy</value>
<value type="int" key="ClangTools.ParallelJobs">1</value>
<valuelist type="QVariantList" key="ClangTools.SelectedDirs"/>
<valuelist type="QVariantList" key="ClangTools.SelectedFiles"/>
<valuelist type="QVariantList" key="ClangTools.SuppressedDiagnostics"/>
<value type="bool" key="ClangTools.UseGlobalSettings">true</value>
</valuemap>
</valuemap>
</data>
<data>
<variable>ProjectExplorer.Project.Target.0</variable>
<valuemap type="QVariantMap">
<value type="QString" key="DeviceType">Desktop</value>
<value type="QString" key="ProjectExplorer.ProjectConfiguration.DefaultDisplayName">qt5</value>
<value type="QString" key="ProjectExplorer.ProjectConfiguration.DisplayName">qt5</value>

```

```

<value type="QString" key="ProjectExplorer.ProjectConfiguration.Id">{d84de845-86d7-4330-b435-516f5fdd6885}</value>
<value type="int" key="ProjectExplorer.Target.ActiveBuildConfiguration">0</value>
<value type="int" key="ProjectExplorer.Target.ActiveDeployConfiguration">0</value>
<value type="int" key="ProjectExplorer.Target.ActiveRunConfiguration">0</value>
<valuemap type="QVariantMap" key="ProjectExplorer.Target.BuildConfiguration.0">
  <value type="int" key="EnableQmlDebugging">0</value>
  <value type="QString" key="ProjectExplorer.BuildConfiguration.BuildDirectory">/home/pouser/build-rostik_qt-qt5-Debug</value>
  <value type="QString"
key="ProjectExplorer.BuildConfiguration.BuildDirectory.shadowDir">/home/pouser/build-rostik_qt-qt5-Debug</value>
  <valuemap type="QVariantMap" key="ProjectExplorer.BuildConfiguration.BuildStepList.0">
    <valuemap type="QVariantMap" key="ProjectExplorer.BuildStepList.Step.0">
      <value type="bool" key="ProjectExplorer.BuildStep.Enabled">true</value>
      <value type="QString"
key="ProjectExplorer.ProjectConfiguration.Id">QtProjectManager.QMakeBuildStep</value>
      <value type="bool" key="QtProjectManager.QMakeBuildStep.QMakeForced">false</value>
      <valuelist type="QVariantList" key="QtProjectManager.QMakeBuildStep.SelectedAbis"/>
    </valuemap>
    <valuemap type="QVariantMap" key="ProjectExplorer.BuildStepList.Step.1">
      <value type="bool" key="ProjectExplorer.BuildStep.Enabled">true</value>
      <value type="QString"
key="ProjectExplorer.ProjectConfiguration.Id">Qt4ProjectManager.MakeStep</value>
    </valuemap>
    <value type="int" key="ProjectExplorer.BuildStepList.StepsCount">2</value>
    <value type="QString"
key="ProjectExplorer.ProjectConfiguration.DefaultDisplayName">Сборка</value>
    <value type="QString" key="ProjectExplorer.ProjectConfiguration.DisplayName">Сборка</value>
    <value type="QString"
key="ProjectExplorer.ProjectConfiguration.Id">ProjectExplorer.BuildSteps.Build</value>
  </valuemap>
  <valuemap type="QVariantMap" key="ProjectExplorer.BuildConfiguration.BuildStepList.1">
    <valuemap type="QVariantMap" key="ProjectExplorer.BuildStepList.Step.0">
      <value type="bool" key="ProjectExplorer.BuildStep.Enabled">true</value>
      <value type="QString"
key="ProjectExplorer.ProjectConfiguration.Id">Qt4ProjectManager.MakeStep</value>
      <value type="QString" key="Qt4ProjectManager.MakeStep.MakeArguments">clean</value>
    </valuemap>
    <value type="int" key="ProjectExplorer.BuildStepList.StepsCount">1</value>
    <value type="QString"
key="ProjectExplorer.ProjectConfiguration.DefaultDisplayName">Очистка</value>
    <value type="QString" key="ProjectExplorer.ProjectConfiguration.DisplayName">Очистка</value>
    <value type="QString"
key="ProjectExplorer.ProjectConfiguration.Id">ProjectExplorer.BuildSteps.Clean</value>
  </valuemap>
  <value type="int" key="ProjectExplorer.BuildConfiguration.BuildStepListCount">2</value>
  <value type="bool" key="ProjectExplorer.BuildConfiguration.ClearSystemEnvironment">false</value>
  <valuelist type="QVariantList" key="ProjectExplorer.BuildConfiguration.CustomParsers"/>
  <value type="bool" key="ProjectExplorer.BuildConfiguration.ParseStandardOutput">false</value>
  <valuelist type="QVariantList" key="ProjectExplorer.BuildConfiguration.UserEnvironmentChanges"/>
  <value type="QString" key="ProjectExplorer.ProjectConfiguration.DisplayName">Отладка</value>
  <value type="QString"
key="ProjectExplorer.ProjectConfiguration.Id">Qt4ProjectManager.Qt4BuildConfiguration</value>
  <value type="int" key="Qt4ProjectManager.Qt4BuildConfiguration.BuildConfiguration">2</value>
</valuemap>
<valuemap type="QVariantMap" key="ProjectExplorer.Target.BuildConfiguration.1">
  <value type="QString" key="ProjectExplorer.BuildConfiguration.BuildDirectory">/home/pouser/build-rostik_qt-qt5-Release</value>

```



```

    <value type="QString"
key="ProjectExplorer.BuildConfiguration.BuildDirectory.shadowDir">/home/pouser/build-rostik_qt-qt5-
Release</value>
    <valuemap type="QVariantMap" key="ProjectExplorer.BuildConfiguration.BuildStepList.0">
    <valuemap type="QVariantMap" key="ProjectExplorer.BuildStepList.Step.0">
    <value type="bool" key="ProjectExplorer.BuildStep.Enabled">true</value>
    <value type="QString"
key="ProjectExplorer.ProjectConfiguration.Id">QtProjectManager.QMakeBuildStep</value>
    <value type="bool" key="QtProjectManager.QMakeBuildStep.QMakeForced">>false</value>
    <valuelist type="QVariantList" key="QtProjectManager.QMakeBuildStep.SelectedAbis"/>
    </valuemap>
    <valuemap type="QVariantMap" key="ProjectExplorer.BuildStepList.Step.1">
    <value type="bool" key="ProjectExplorer.BuildStep.Enabled">true</value>
    <value type="QString"
key="ProjectExplorer.ProjectConfiguration.Id">Qt4ProjectManager.MakeStep</value>
    </valuemap>
    <value type="int" key="ProjectExplorer.BuildStepList.StepsCount">2</value>
    <value type="QString"
key="ProjectExplorer.ProjectConfiguration.DefaultDisplayName">Сборка</value>
    <value type="QString" key="ProjectExplorer.ProjectConfiguration.DisplayName">Сборка</value>
    <value type="QString"
key="ProjectExplorer.ProjectConfiguration.Id">ProjectExplorer.BuildSteps.Build</value>
    </valuemap>
    <valuemap type="QVariantMap" key="ProjectExplorer.BuildConfiguration.BuildStepList.1">
    <valuemap type="QVariantMap" key="ProjectExplorer.BuildStepList.Step.0">
    <value type="bool" key="ProjectExplorer.BuildStep.Enabled">true</value>
    <value type="QString"
key="ProjectExplorer.ProjectConfiguration.Id">Qt4ProjectManager.MakeStep</value>
    <value type="QString" key="Qt4ProjectManager.MakeStep.MakeArguments">clean</value>
    </valuemap>
    <value type="int" key="ProjectExplorer.BuildStepList.StepsCount">1</value>
    <value type="QString"
key="ProjectExplorer.ProjectConfiguration.DefaultDisplayName">Очистка</value>
    <value type="QString" key="ProjectExplorer.ProjectConfiguration.DisplayName">Очистка</value>
    <value type="QString"
key="ProjectExplorer.ProjectConfiguration.Id">ProjectExplorer.BuildSteps.Clean</value>
    </valuemap>
    <value type="int" key="ProjectExplorer.BuildConfiguration.BuildStepListCount">2</value>
    <value type="bool" key="ProjectExplorer.BuildConfiguration.ClearSystemEnvironment">>false</value>
    <valuelist type="QVariantList" key="ProjectExplorer.BuildConfiguration.CustomParsers"/>
    <value type="bool" key="ProjectExplorer.BuildConfiguration.ParseStandardOutput">>false</value>
    <valuelist type="QVariantList" key="ProjectExplorer.BuildConfiguration.UserEnvironmentChanges"/>
    <value type="QString" key="ProjectExplorer.ProjectConfiguration.DisplayName">Выпуск</value>
    <value type="QString"
key="ProjectExplorer.ProjectConfiguration.Id">Qt4ProjectManager.Qt4BuildConfiguration</value>
    <value type="int" key="Qt4ProjectManager.Qt4BuildConfiguration.BuildConfiguration">0</value>
    <value type="int" key="QtQuickCompiler">0</value>
    </valuemap>
    <valuemap type="QVariantMap" key="ProjectExplorer.Target.BuildConfiguration.2">
    <value type="int" key="EnableQmlDebugging">0</value>
    <value type="QString" key="ProjectExplorer.BuildConfiguration.BuildDirectory">/home/pouser/build-
rostik_qt-qt5-Profile</value>
    <value type="QString"
key="ProjectExplorer.BuildConfiguration.BuildDirectory.shadowDir">/home/pouser/build-rostik_qt-qt5-
Profile</value>
    <valuemap type="QVariantMap" key="ProjectExplorer.BuildConfiguration.BuildStepList.0">
    <valuemap type="QVariantMap" key="ProjectExplorer.BuildStepList.Step.0">
    <value type="bool" key="ProjectExplorer.BuildStep.Enabled">true</value>
    <value type="QString"
key="ProjectExplorer.ProjectConfiguration.Id">QtProjectManager.QMakeBuildStep</value>

```

```

    <value type="bool" key="QtProjectManager.QMakeBuildStep.QMakeForced">false</value>
    <valuelist type="QVariantList" key="QtProjectManager.QMakeBuildStep.SelectedAbis"/>
  </valuemap>
  <valuemap type="QVariantMap" key="ProjectExplorer.BuildStepList.Step.1">
    <value type="bool" key="ProjectExplorer.BuildStep.Enabled">true</value>
    <value type="QString"
key="ProjectExplorer.ProjectConfiguration.Id">Qt4ProjectManager.MakeStep</value>
  </valuemap>
    <value type="int" key="ProjectExplorer.BuildStepList.StepsCount">2</value>
    <value type="QString"
key="ProjectExplorer.ProjectConfiguration.DefaultDisplayName">Сборка</value>
    <value type="QString" key="ProjectExplorer.ProjectConfiguration.DisplayName">Сборка</value>
    <value type="QString"
key="ProjectExplorer.ProjectConfiguration.Id">ProjectExplorer.BuildSteps.Build</value>
  </valuemap>
    <valuemap type="QVariantMap" key="ProjectExplorer.BuildConfiguration.BuildStepList.1">
    <valuemap type="QVariantMap" key="ProjectExplorer.BuildStepList.Step.0">
      <value type="bool" key="ProjectExplorer.BuildStep.Enabled">true</value>
      <value type="QString"
key="ProjectExplorer.ProjectConfiguration.Id">Qt4ProjectManager.MakeStep</value>
      <value type="QString" key="Qt4ProjectManager.MakeStep.MakeArguments">clean</value>
    </valuemap>
      <value type="int" key="ProjectExplorer.BuildStepList.StepsCount">1</value>
      <value type="QString"
key="ProjectExplorer.ProjectConfiguration.DefaultDisplayName">Очистка</value>
      <value type="QString" key="ProjectExplorer.ProjectConfiguration.DisplayName">Очистка</value>
      <value type="QString"
key="ProjectExplorer.ProjectConfiguration.Id">ProjectExplorer.BuildSteps.Clean</value>
    </valuemap>
      <value type="int" key="ProjectExplorer.BuildConfiguration.BuildStepListCount">2</value>
      <value type="bool" key="ProjectExplorer.BuildConfiguration.ClearSystemEnvironment">false</value>
      <valuelist type="QVariantList" key="ProjectExplorer.BuildConfiguration.CustomParsers"/>
      <value type="bool" key="ProjectExplorer.BuildConfiguration.ParseStandardOutput">false</value>
      <valuelist type="QVariantList" key="ProjectExplorer.BuildConfiguration.UserEnvironmentChanges"/>
      <value type="QString"
key="ProjectExplorer.ProjectConfiguration.DisplayName">Профилирование</value>
      <value type="QString"
key="ProjectExplorer.ProjectConfiguration.Id">Qt4ProjectManager.Qt4BuildConfiguration</value>
      <value type="int" key="Qt4ProjectManager.Qt4BuildConfiguration.BuildConfiguration">0</value>
      <value type="int" key="QtQuickCompiler">0</value>
      <value type="int" key="SeparateDebugInfo">0</value>
    </valuemap>
      <value type="int" key="ProjectExplorer.Target.BuildConfigurationCount">3</value>
      <valuemap type="QVariantMap" key="ProjectExplorer.Target.DeployConfiguration.0">
      <valuemap type="QVariantMap" key="ProjectExplorer.BuildConfiguration.BuildStepList.0">
        <value type="int" key="ProjectExplorer.BuildStepList.StepsCount">0</value>
        <value type="QString"
key="ProjectExplorer.ProjectConfiguration.DefaultDisplayName">Развёртывание</value>
        <value type="QString"
key="ProjectExplorer.ProjectConfiguration.DisplayName">Развёртывание</value>
        <value type="QString"
key="ProjectExplorer.ProjectConfiguration.Id">ProjectExplorer.BuildSteps.Deploy</value>
      </valuemap>
        <value type="int" key="ProjectExplorer.BuildConfiguration.BuildStepListCount">1</value>
        <valuemap type="QVariantMap" key="ProjectExplorer.DeployConfiguration.CustomData"/>
        <value type="bool" key="ProjectExplorer.DeployConfiguration.CustomDataEnabled">false</value>
        <value type="QString"
key="ProjectExplorer.ProjectConfiguration.Id">ProjectExplorer.DefaultDeployConfiguration</value>
      </valuemap>
        <value type="int" key="ProjectExplorer.Target.DeployConfigurationCount">1</value>

```

```

<valuemap type="QVariantMap" key="ProjectExplorer.Target.RunConfiguration.0">
  <value type="bool" key="Analyzer.Perf.Settings.UseGlobalSettings">true</value>
  <value type="bool" key="Analyzer.QmlProfiler.Settings.UseGlobalSettings">true</value>
  <value type="bool" key="Analyzer.Valgrind.Settings.UseGlobalSettings">true</value>
  <valuelist type="QVariantList" key="CustomOutputParsers"/>
  <value type="int" key="PE.EnvironmentAspect.Base">2</value>
  <valuelist type="QVariantList" key="PE.EnvironmentAspect.Changes"/>
  <value type="QString"
key="ProjectExplorer.ProjectConfiguration.Id">Qt4ProjectManager.Qt4RunConfiguration:/home/pouser/rost
ik_qt/rostik_qt.pro</value>
  <value type="QString"
key="ProjectExplorer.RunConfiguration.BuildKey"/>/home/pouser/rostik_qt/rostik_qt.pro</value>
  <value type="bool" key="RunConfiguration.UseCppDebugger">>false</value>
  <value type="bool" key="RunConfiguration.UseCppDebuggerAuto">true</value>
  <value type="bool" key="RunConfiguration.UseLibrarySearchPath">true</value>
  <value type="bool" key="RunConfiguration.UseQmlDebugger">>false</value>
  <value type="bool" key="RunConfiguration.UseQmlDebuggerAuto">true</value>
  <value type="QString" key="RunConfiguration.WorkingDirectory.default">/home/pouser/build-rostik_qt-
qt5-Debug</value>
</valuemap>
  <value type="int" key="ProjectExplorer.Target.RunConfigurationCount">1</value>
</valuemap>
</data>
<data>
  <variable>ProjectExplorer.Project.TargetCount</variable>
  <value type="int">1</value>
</data>
<data>
  <variable>ProjectExplorer.Project.Updater.FileVersion</variable>
  <value type="int">22</value>
</data>
<data>
  <variable>Version</variable>
  <value type="int">22</value>
</data>
</qtcreator>

```

style.qss

```

* {
  font-family: "Roboto", Arial, sans-serif;
  font-size: 14px;
  color: #333;
}

QMainWindow {
  background-color: #f5f5f5;
}

QPushButton {
  background-color: #007BFF;
  color: white;
  border: none;
  font-size: 14px;
  padding: 10px 20px;
  border-radius: 5px;
}

QPushButton:hover{

```

```

    background-color: #0056b3;
}

QPushButton:pressed{
    background-color: #004080;
}

QLineEdit {
    border: 1px solid #ccc;
    border-radius: 5px;
    padding: 5px;
    background-color: #fff;
}

QLineEdit:focus {
    border-color: #007BFF;
}

QLabel {
    font-size: 14px;
    color: #555;
}

QTableView {
    border: 1px solid #ccc;
    gridline-color: #e0e0e0;
    selection-background-color: #007BFF;
    selection-color: white;
}

QCheckBox {
    spacing: 5px;
}

QCheckBox::indicator {
    width: 18px;
    height: 18px;
}

QTextBrowser{
    border: 1px solid #ccc;
    border-radius: 5px;
    padding: 5px;
    background-color: #fff;
}

```

tcpclient.cpp

```

#include "tcpclient.h"
#include <QDebug>

QTcpSocket socket;
bool error;
TcpClient::TcpClient(const QString &host, int port, QObject *parent) : QObject(parent)
{
    error = false;
    socket = new QTcpSocket(this);
    connect(socket, &QTcpSocket::connected, this, &TcpClient::onConnected);
    connect(socket, &QTcpSocket::readyRead, this, &TcpClient::onReadyRead);
}

```

```

        connect(socket, &QTcpSocket::errorOccurred, this, &TcpClient::onError);
        socket->connectToHost(host, port);
    }
    QList<QJsonObject> TcpClient::sendDataList(const QByteArray &msg){
        QList<QJsonObject> none;
        socket->waitForConnected(5000);
        if (socket->isOpen()){
            socket->write(msg);
            socket->waitForBytesWritten(5000);
        }
        else {
            qDebug() << "Не удалось подключиться";
            return none;
        }
        socket->waitForReadyRead(5000);
        QByteArray data = socket->readAll();
        QJsonDocument jsonDoc = QJsonDocument::fromJson(data);
        QJsonObject jsonResponse = jsonDoc.object();

        for (int i = 0; i < jsonResponse.keys().size(); i++){
            QString val = jsonResponse[jsonResponse.keys().at(i)].toString();
            if (val != ""){
                qDebug() << jsonResponse.keys().at(i) << " - " << val;
            }
            else {
                qDebug() << jsonResponse.keys().at(i) << " - " <<
                jsonResponse[jsonResponse.keys().at(i)].toDouble();
            }
        }
        QString recievedChecksum = jsonResponse["checksum"].toString();
        jsonResponse.remove("checksum");
        QByteArray jsonData = QJsonDocument(jsonResponse).toJson();
        QString str = jsonResponse["data"].toString();
        QByteArray jsonValue = str.toUtf8();
        QByteArray calculatedChecksum = QCryptographicHash::hash(jsonValue,
        QCryptographicHash::Sha256).toHex();
        QJsonDocument jsonDoc2 = QJsonDocument::fromJson(jsonValue);
        if (recievedChecksum != QString(calculatedChecksum)){
            qDebug() << "Несовпадение контрольных сумм.";
            return none;
        }
        else if (/*jsonResponse["data"].isArray()*/ true){
            QJsonArray jsonArray = /*jsonResponse["data"].toArray()*/jsonDoc2.array();
            QList<QJsonObject> arr;
            for (const QJsonValue &value : jsonArray){
                if (value.isObject()){
                    QJsonObject jsonObject = value.toObject();
                    qDebug() << "Объект: " << jsonObject;
                    arr.append(jsonObject);
                }
            }
            return arr;
        }
        else {
            qDebug() << "Получен не массив";
        }
        return none;
    }
}

bool TcpClient::checkConnection(){

```

```

    return socket->waitForConnected(5000);
}
QString TcpClient::sendDataString(const QByteArray &msg){
    QList<QJsonObject> none;
    socket->waitForConnected(5000);
    if (error){
        qDebug() << "Не удалось подключиться";
        return "ERROR_Не удалось подключиться к серверу.";
    }
    if (socket->isOpen()){
        socket->write(msg);
        socket->waitForBytesWritten(5000);
    }
    else{
        qDebug() << "Не удалось подключиться2";
        return "ERROR_Не удалось подключиться к серверу.";
    }
    socket->waitForReadyRead(5000);
    QByteArray data = socket->readAll();
    QJsonDocument jsonDoc = QJsonDocument::fromJson(data);
    QJsonObject jsonResponse = jsonDoc.object();

    for (int i = 0; i < jsonResponse.keys().size(); i++){
        QString val = jsonResponse[jsonResponse.keys().at(i)].toString();
        if (val != ""){
            qDebug() << jsonResponse.keys().at(i) << " - " << val;
        }
        else {
            qDebug() << jsonResponse.keys().at(i) << " - " <<
jsonResponse[jsonResponse.keys().at(i)].toDouble();
        }
    }
    QString recievedChecksum = jsonResponse["checksum"].toString();
    jsonResponse.remove("checksum");
    QByteArray jsonData = QJsonDocument(jsonResponse).toJson();
    QString str = jsonResponse["data"].toString();
    QByteArray jsonValue = str.toUtf8();
    QByteArray calculatedChecksum = QCryptographicHash::hash(jsonValue,
QCryptographicHash::Sha256).toHex();
    if (recievedChecksum != QString(calculatedChecksum)){
        qDebug() << "Несовпадение контрольных сумм.";
        return "ERROR_Несовпадение контрольных сумм.";
    }
    else if (jsonResponse["type"].toString() == "error"){
        return "ERROR_" + jsonResponse["data"].toString();
    }
    else {
        return "DATA_" + jsonResponse["data"].toString();
    }
}

void TcpClient::onConnected(){
    qDebug() << "Подключено к серверу";
}
void TcpClient::onError(QAbstractSocket::SocketError err){
    qDebug() << "Ошибка: " << err;
    error = true;
}
void TcpClient::onReadyRead(){
    // QByteArray data = socket->readAll();

```

```

// QJsonDocument jsonDoc = QJsonDocument::fromJson(data);
// if (jsonDoc.isArray()) {
//     QJsonArray jsonArray = jsonDoc.array();
//     for (const QJsonValue &value : jsonArray) {
//         if (value.isObject()) {
//             QJsonObject jsonObject = value.toObject();
//             qDebug() << "Устройство: " << jsonObject;
//         }
//     }
// }
// else {
//     qDebug() << "Получен не json-массив";
// }
}

```

tcpclient.h

```

#ifndef TCPCLIENT_H
#define TCPCLIENT_H

#include <QTcpSocket>
#include <QJsonArray>
#include <QJsonObject>
#include <QJsonDocument>
#include <QCryptographicHash>

class TcpClient : public QObject
{
    Q_OBJECT

public:
    TcpClient(const QString &host, int port, QObject *parent = nullptr);
    QList<QJsonObject> sendDataList(const QByteArray &data);
    QString sendDataString(const QByteArray &data);
    bool error;
    bool checkConnection();
private slots:
    void onConnected();
    void onError(QAbstractSocket::SocketError);
    void onReadyRead();

private:
    QTcpSocket *socket;
};

#endif // TCPCLIENT_H

```

userwindow.cpp

```

#include "userwindow.h"
#include "ui_userwindow.h"
#include "historywindow.h"
#include "tcpclient.h"
#include "main.h"
#include "workwindow.h"

bool double_click = false;
bool ignoreActive = true;

```

```

UserWindow::UserWindow(QWidget *parent) :
    QMainWindow(parent),
    ui(new Ui::UserWindow){
    ui->setupUi(this);
    connect(ui->historyUserButton, &QPushButton::clicked, this,
        &UserWindow::on_historyUserButton_clicked);
    connect(ui->workButton, &QPushButton::clicked, this,
        &UserWindow::on_workButton_clicked);
    setWindowFlags(Qt::Dialog);
    model = new QStandardItemModel(this);
    ui->devicesUserTable->setModel(model);
    ui->devicesUserTable->setEditTriggers(QAbstractItemView::NoEditTriggers);
    getDevices();
    QTimer *timer = new QTimer(this);
    QObject::connect(timer, &QTimer::timeout, this, &UserWindow::getDevices);
    timer->start(5000);
}

void UserWindow::getDevices(){
    if (isActiveWindow() || ignoreActive){
        ignoreActive = false;
        QList<QJsonObject> jsonArray = client->sendDataList("get_devices");
        model->clear();
        model->setColumnCount(12);
        model->setHorizontalHeaderLabels(QStringList() << "№" << "Название" << "Включен" <<
            "Напряжение" << "Мин. Напряжение" << "Макс. Напряжение"
                << "Ток" << "Макс. Ток" << "Температура" << "Макс. Температура"
                << "Емкость" << "№ Расположения");
        for(QJsonObject jobj : jsonArray){
            QList<QStandardItem*> rowItems;
            rowItems.append(new QStandardItem(jobj["id"].toString()));
            rowItems.append(new QStandardItem(jobj["name"].toString()));
            rowItems.append(new QStandardItem(jobj["turned"].toString()));
            rowItems.append(new QStandardItem(jobj["voltage"].toString()));
            rowItems.append(new QStandardItem(jobj["voltage_min"].toString()));
            rowItems.append(new QStandardItem(jobj["voltage_max"].toString()));
            rowItems.append(new QStandardItem(jobj["amperage"].toString()));
            rowItems.append(new QStandardItem(jobj["amperage_max"].toString()));
            rowItems.append(new QStandardItem(jobj["temperature"].toString()));
            rowItems.append(new QStandardItem(jobj["temperature_max"].toString()));
            rowItems.append(new QStandardItem(jobj["capacity"].toString()));
            rowItems.append(new QStandardItem(jobj["placement_id"].toString()));
            model->appendRow(rowItems);
        }
    }
}

void UserWindow::on_historyUserButton_clicked(){
    if (double_click){
        double_click = false;
    }
    else{
        double_click = true;
        HistoryWindow * w = new HistoryWindow(this);
        w->show();
    }
}

void UserWindow::on_workButton_clicked(){
    if (double_click){

```



```

        double_click = false;
    }
    else{
        double_click = true;
        WorkWindow * w = new WorkWindow(this);
        w->show();
    }
}

```

```

UserWindow::~UserWindow()
{
    delete ui;
}

```

userwindow.h

```

#ifndef USERWINDOW_H
#define USERWINDOW_H

#include <QMainWindow>
#include <QMessageBox>
#include <QDebug>
#include <QTimer>
#include <QJsonObject>
#include <QList>
#include <QStandardItem>

namespace Ui {
class UserWindow;
}

class UserWindow : public QMainWindow
{
    Q_OBJECT

public:
    explicit UserWindow(QWidget *parent = nullptr);
    ~UserWindow();
private slots:
    void on_historyUserButton_clicked();
    void getDevices();
    void on_workButton_clicked();
private:
    QStandardItemModel *model;
    Ui::UserWindow *ui;
};

#endif // USERWINDOW_H

```

userwindow.ui

```

<?xml version="1.0" encoding="UTF-8"?>
<ui version="4.0">
<class>UserWindow</class>
<widget class="QMainWindow" name="UserWindow">
<property name="geometry">
<rect>
<x>0</x>

```

```

        <y>0</y>
        <width>800</width>
        <height>454</height>
    </rect>
</property>
<property name="windowTitle">
    <string>UserWindow</string>
</property>
<widget class="QWidget" name="centralwidget">
    <layout class="QGridLayout" name="gridLayout">
        <item row="1" column="0" colspan="2">
            <widget class="QTableView" name="devicesUserTable"/>
        </item>
        <item row="2" column="0">
            <widget class="QPushButton" name="historyUserButton">
                <property name="text">
                    <string>История аварийных случаев</string>
                </property>
            </widget>
        </item>
        <item row="2" column="1">
            <widget class="QPushButton" name="workButton">
                <property name="text">
                    <string>Работа с данными</string>
                </property>
            </widget>
        </item>
        <item row="0" column="0" colspan="2">
            <widget class="QLabel" name="label">
                <property name="text">
                    <string>Добро пожаловать!</string>
                </property>
                <property name="alignment">
                    <set>Qt::AlignCenter</set>
                </property>
            </widget>
        </item>
    </layout>
</widget>
<widget class="QMenuBar" name="menubar">
    <property name="geometry">
        <rect>
            <x>0</x>
            <y>0</y>
            <width>800</width>
            <height>29</height>
        </rect>
    </property>
</widget>
<widget class="QStatusBar" name="statusbar"/>
</widget>
<resources/>
<connections/>
</ui>

```

workwindow.cpp

```

#include "workwindow.h"
#include "ui_workwindow.h"

```

```

#include "main.h"
#include "graphwindow.h"
#include "userwindow.h"

WorkWindow::WorkWindow(QWidget *parent) :
    QMainWindow(parent),
    ui(new Ui::WorkWindow)
{
    ui->setupUi(this);
    connect(ui->filterButton, &QPushButton::clicked, this,
        &WorkWindow::on_filterButton_clicked);
    connect(ui->graphButton, &QPushButton::clicked, this,
        &WorkWindow::on_graphButton_clicked);
    connect(ui->importCsvButton, &QPushButton::clicked, this,
        &WorkWindow::on_importCsvButton_clicked);
    connect(ui->importJsonButton, &QPushButton::clicked, this,
        &WorkWindow::on_importJsonButton_clicked);
    model = new QStandardItemModel(this);
    double_click = false;
    ui->filterTable->setModel(model);
    ui->filterTable->setEditTriggers(QAbstractItemView::NoEditTriggers);
    getDevices(client->sendDataList("get_devices"));
}

void WorkWindow::on_filterButton_clicked(){
    QString voltageMin = QString::number(ui->spinBoxVoltageMin->value());
    QString voltageMax = QString::number(ui->spinBoxVoltageMax->value());
    QString amperageMin = QString::number(ui->spinBoxAmperageMin->value());
    QString amperageMax = QString::number(ui->spinBoxAmperageMax->value());
    QString temperatureMin = QString::number(ui->spinBoxTemperatureMin->value());
    QString temperatureMax = QString::number(ui->spinBoxTemperatureMax->value());

    QString strFilter = "v_" + voltageMin + "_" + voltageMax +
        "_a_" + amperageMin + "_" + amperageMax +
        "_t_" + temperatureMin + "_" + temperatureMax;
    getDevices(client->sendDataList("filter_devices_" + strFilter.toUtf8()));
}

void WorkWindow::on_graphButton_clicked(){
    if(double_click){
        double_click = false;
    }
    else{
        double_click = true;
        GraphWindow * w = new GraphWindow(this);
        w->show();
    }
}

void WorkWindow::on_importCsvButton_clicked(){
    QList<QJsonObject> arr = client->sendDataList("get_devices");
    QJsonArray jsonArray;
    for (const QJsonObject& jsonObject : arr){
        jsonArray.append(jsonObject);
    }
    QString folderPath = QFileDialog::getExistingDirectory(nullptr, "Выберите папку для сохранения",
        QDir::homePath());
    if(!folderPath.isEmpty()){
        QFile file(folderPath + "/devices.csv");
        if(!file.open(QIODevice::WriteOnly)){

```

```

        QMessageBox::critical(this, "Не получилось открыть файл", file.errorString());
    }
    QTextStream out(&file);
    QJsonObject firstObject = jsonArray[0].toObject();
    QStringList headers = firstObject.keys();
    out << headers.join(",") << "\n";
    for (const QJsonValue &value : jsonArray){
        QJsonObject obj = value.toObject();
        QStringList row;

        for (const QString &header : headers){
            row << obj.value(header).toString();
        }
        out << row.join(",") << "\n";
    }
    file.write(out.readAll().toUtf8());
    file.close();
}

void WorkWindow::on_importJsonButton_clicked(){
    QList<QJsonObject> arr = client->sendDataList("get_devices");
    QJsonArray jsonArray;
    for (const QJsonObject& jsonObject : arr){
        jsonArray.append(jsonObject);
    }
    QJsonDocument jsonDoc(jsonArray);
    QString folderPath = QFileDialog::getExistingDirectory(nullptr, "Выберите папку для сохранения",
QDir::homePath());
    if(!folderPath.isEmpty()){
        QFile file(folderPath + "/devices.json");
        if(!file.open(QIODevice::WriteOnly)){
            QMessageBox::critical(this, "Не получилось открыть файл", file.errorString());
        }
        file.write(jsonDoc.toJson(QJsonDocument::Indented));
        file.close();
    }
}

void WorkWindow::getDevices(QList<QJsonObject> jsonArray){
    model->clear();
    model->setColumnCount(12);
    model->setHorizontalHeaderLabels(QStringList() << "№" << "Название" << "Включен" <<
"Напряжение" << "Мин. Напряжение" << "Макс. Напряжение"
        << "Ток" << "Макс. Ток" << "Температура" << "Макс. Температура"
        << "Емкость" << "№ Расположения");
    for(QJsonObject jobj : jsonArray){
        QList<QStandardItem*> rowItems;
        rowItems.append(new QStandardItem(jobj["id"].toString()));
        rowItems.append(new QStandardItem(jobj["name"].toString()));
        rowItems.append(new QStandardItem(jobj["turned"].toString()));
        rowItems.append(new QStandardItem(jobj["voltage"].toString()));
        rowItems.append(new QStandardItem(jobj["voltage_min"].toString()));
        rowItems.append(new QStandardItem(jobj["voltage_max"].toString()));
        rowItems.append(new QStandardItem(jobj["amperage"].toString()));
        rowItems.append(new QStandardItem(jobj["amperage_max"].toString()));
        rowItems.append(new QStandardItem(jobj["temperature"].toString()));
        rowItems.append(new QStandardItem(jobj["temperature_max"].toString()));
        rowItems.append(new QStandardItem(jobj["capacity"].toString()));
        rowItems.append(new QStandardItem(jobj["placement_id"].toString()));
    }
}

```

```

        model->appendRow(rowItems);
    }
}

```

```

WorkWindow::~WorkWindow()
{
    delete ui;
}

```

workwindow.h

```

#ifndef WORKWINDOW_H
#define WORKWINDOW_H

```

```

#include <QMainWindow>
#include "/home/pouser/Desktop/qcustomplot/qcustomplot.h"
#include "/home/pouser/Desktop/qcustomplot/examples/text-document-integration/qcpdocumentobject.h"
#include <QFileDialog>
#include <QFile>
#include <QMessageBox>

```

```

namespace Ui {
class WorkWindow;
}

```

```

class WorkWindow : public QMainWindow
{
    Q_OBJECT

```

```

public:
    explicit WorkWindow(QWidget *parent = nullptr);
    ~WorkWindow();

```

```

private slots:
    void on_filterButton_clicked();
    void on_importJsonButton_clicked();
    void on_importCsvButton_clicked();
    void on_graphButton_clicked();
    void getDevices(QList<QJsonObject> jsonArray);

```

```

private:
    QStandardItemModel *model;
    bool double_click;
    Ui::WorkWindow *ui;
};

```

```

#endif // WORKWINDOW_H

```

workwindow.ui

```

<?xml version="1.0" encoding="UTF-8"?>
<ui version="4.0">
<class>WorkWindow</class>
<widget class="QMainWindow" name="WorkWindow">
<property name="geometry">
<rect>
<x>0</x>
<y>0</y>

```

```

<width>800</width>
<height>600</height>
</rect>
</property>
<property name="windowTitle">
<string>Main Window</string>
</property>
<widget class="QWidget" name="centralwidget">
<layout class="QGridLayout" name="gridLayout">
<item row="3" column="2">
<widget class="QSpinBox" name="spinBoxTemperatureMax">
<property name="maximum">
<number>1000</number>
</property>
<property name="value">
<number>300</number>
</property>
</widget>
</item>
<item row="1" column="1">
<widget class="QLabel" name="label_2">
<property name="text">
<string>От:</string>
</property>
</widget>
</item>
<item row="3" column="3">
<widget class="QPushButton" name="importCsvButton">
<property name="text">
<string>Импорт csv</string>
</property>
</widget>
</item>
<item row="5" column="1" colspan="2">
<widget class="QPushButton" name="filterButton">
<property name="text">
<string>Отфильтровать данные</string>
</property>
</widget>
</item>
<item row="1" column="2">
<widget class="QLabel" name="label_3">
<property name="text">
<string>До:</string>
</property>
</widget>
</item>
<item row="2" column="3">
<widget class="QPushButton" name="graphButton">
<property name="text">
<string>График</string>
</property>
</widget>
</item>
<item row="0" column="0" colspan="4">
<widget class="QTableView" name="filterTable"/>
</item>
<item row="4" column="0">
<widget class="QLabel" name="label_6">
<property name="layoutDirection">

```

```

    <enum>Qt::LeftToRight</enum>
  </property>
  <property name="text">
    <string>Tok</string>
  </property>
  <property name="alignment">
    <set>Qt::AlignRight|Qt::AlignTrailing|Qt::AlignVCenter</set>
  </property>
</widget>
</item>
<item row="2" column="1">
  <widget class="QSpinBox" name="spinBoxVoltageMin">
    <property name="maximum">
      <number>999</number>
    </property>
  </widget>
</item>
<item row="3" column="1">
  <widget class="QSpinBox" name="spinBoxTemperatureMin">
    <property name="maximum">
      <number>999</number>
    </property>
  </widget>
</item>
<item row="4" column="1">
  <widget class="QSpinBox" name="spinBoxAmperageMin">
    <property name="maximum">
      <number>999</number>
    </property>
  </widget>
</item>
<item row="2" column="2">
  <widget class="QSpinBox" name="spinBoxVoltageMax">
    <property name="maximum">
      <number>1000</number>
    </property>
    <property name="value">
      <number>300</number>
    </property>
  </widget>
</item>
<item row="4" column="3">
  <widget class="QPushButton" name="importJsonButton">
    <property name="text">
      <string>Импорт json</string>
    </property>
  </widget>
</item>
<item row="4" column="2">
  <widget class="QSpinBox" name="spinBoxAmperageMax">
    <property name="maximum">
      <number>1000</number>
    </property>
    <property name="value">
      <number>300</number>
    </property>
  </widget>
</item>
<item row="2" column="0">
  <widget class="QLabel" name="label_4">

```

```

    <property name="text">
    <string>Напряжение</string>
    </property>
    <property name="alignment">
    <set>Qt::AlignRight|Qt::AlignTrailing|Qt::AlignVCenter</set>
    </property>
  </widget>
</item>
<item row="3" column="0">
  <widget class="QLabel" name="label_5">
    <property name="text">
    <string>Температура</string>
    </property>
    <property name="alignment">
    <set>Qt::AlignRight|Qt::AlignTrailing|Qt::AlignVCenter</set>
    </property>
  </widget>
</item>
</layout>
</widget>
<widget class="QMenuBar" name="menubar">
  <property name="geometry">
    <rect>
      <x>0</x>
      <y>0</y>
      <width>800</width>
      <height>29</height>
    </rect>
  </property>
</widget>
<widget class="QStatusBar" name="statusbar"/>
</widget>
<resources/>
<connections/>
</ui>

```

————— (Device.java) —————

```
package com.example.diplom_main;
```

```

public class Device {
    private String id, name, turned, voltage, voltageMin, voltageMax,
        amperage, amperageMax, temperature, temperatureMax,
        capacity, placementId;

    public Device(String id, String name, String turned, String voltage, String voltageMin, String voltageMax,
        String amperage, String amperageMax, String temperature, String temperatureMax,
        String capacity, String placementId) {
        this.id = id;
        this.name = name;
        this.turned = turned;
        this.voltage = voltage;
        this.voltageMin = voltageMin;
        this.voltageMax = voltageMax;
        this.amperage = amperage;
        this.amperageMax = amperageMax;
        this.temperature = temperature;
        this.temperatureMax = temperatureMax;
        this.capacity = capacity;
        this.placementId = placementId;
    }
}

```



```
double amperage = Double.parseDouble(device.getAmperage());
double amperageMax = Double.parseDouble(device.getAmperageMax());

double temperature = Double.parseDouble(device.getTemperature());
double temperatureMax = Double.parseDouble(device.getTemperatureMax());

if (voltage < voltageMin || voltage > voltageMax ||
    amperage > amperageMax ||
    temperature > temperatureMax) {
    holder.itemView.setBackgroundColor(Color.RED);
}
else {
    holder.itemView.setBackgroundColor(Color.WHITE);
}
}

@Override
public int getItemCount() {
    return deviceList.size();
}

static class DeviceViewHolder extends RecyclerView.ViewHolder {
    TextView idTextView, nameTextView, turnedTextView, voltageTextView,
    voltageMinTextView, voltageMaxTextView, amperageTextView,
    amperageMaxTextView, temperatureTextView, temperatureMaxTextView,
    capacityTextView, placementIdTextView;

    DeviceViewHolder(View itemView) {
        super(itemView);
        idTextView = itemView.findViewById(R.id.idTextView);
        nameTextView = itemView.findViewById(R.id.nameTextView);
        turnedTextView = itemView.findViewById(R.id.turnedTextView);
        voltageTextView = itemView.findViewById(R.id.voltageTextView);
        voltageMinTextView = itemView.findViewById(R.id.voltageMinTextView);
        voltageMaxTextView = itemView.findViewById(R.id.voltageMaxTextView);
        amperageTextView = itemView.findViewById(R.id.amperageTextView);
        amperageMaxTextView = itemView.findViewById(R.id.amperageMaxTextView);
        temperatureTextView = itemView.findViewById(R.id.temperatureTextView);
        temperatureMaxTextView = itemView.findViewById(R.id.temperatureMaxTextView);
        capacityTextView = itemView.findViewById(R.id.capacityTextView);
        placementIdTextView = itemView.findViewById(R.id.placementIdTextView);
    }
}

package com.example.diplom_main;

import static com.example.diplom_main.StaticData.tcpClient;

import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;
import androidx.appcompat.app.AppCompatActivity;

public class EnterIpActivity extends AppCompatActivity {

    private EditText ipEditText, portEditText;
```

```

private Button connectButton;
private EnterIpActivity thisActivity = this;

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_enter_ip);

    ipEditText = findViewById(R.id.ipEditText);
    portEditText = findViewById(R.id.portEditText);
    connectButton = findViewById(R.id.connectButton);

    // Обработчик кнопки
    connectButton.setOnClickListener(v -> onConnectButtonClicked());
}

private void onConnectButtonClicked() {
    Toast.makeText(this, "Подключение...", Toast.LENGTH_SHORT).show();
    String ip = ipEditText.getText().toString();
    String portStr = portEditText.getText().toString();
    int port = portStr.isEmpty() || !isNumber(portStr) ? 1234 : Integer.parseInt(portStr);
    new Thread() -> {
        try {
            tcpClient = new TcpClient(ip, port);
            boolean response = tcpClient.checkConnection(); // Предполагаемая команда проверки
            runOnUiThread() -> {
                if (response) {
                    Intent intent = new Intent(EnterIpActivity.this, MainActivity.class);
                    Toast.makeText(thisActivity, "Подключение успешно", Toast.LENGTH_LONG).show();
                    thisActivity.startActivity(intent);
                    thisActivity.finish();
                } else {
                    Toast.makeText(this, "Не удалось подключиться. Введите корректный адрес и порт.",
Toast.LENGTH_SHORT).show();
                }
            });
        } catch (Exception e) {
            runOnUiThread() -> Toast.makeText(this, "Не удалось подключиться: " + e.getMessage(),
Toast.LENGTH_SHORT).show();
        }
    }.start();
}

private boolean isNumber(String str) {
    try {
        Integer.parseInt(str);
        return true;
    } catch (NumberFormatException e) {
        return false;
    }
}

@Override
protected void onDestroy() {
    super.onDestroy();
}
}

_____ (GraphActivity.java) _____
package com.example.diplom_main;

```

```

import android.os.Bundle;
import android.os.Handler;
import android.os.Looper;
import android.view.View;
import android.widget.ProgressBar;

import androidx.appcompat.app.AppCompatActivity;
import com.github.mikephil.charting.charts.LineChart;
import com.github.mikephil.charting.components.XAxis;
import com.github.mikephil.charting.components.YAxis;
import com.github.mikephil.charting.data.Entry;
import com.github.mikephil.charting.data.LineData;
import com.github.mikephil.charting.data.LineDataSet;
import org.json.JSONArray;
import org.json.JSONObject;
import java.util.ArrayList;
import java.util.List;
import java.util.concurrent.ExecutorService;
import java.util.concurrent.Executors;
import static com.example.diplom_main.StaticData.tcpClient;

public class GraphActivity extends AppCompatActivity {

    private LineChart lineChart;
    private Handler handler;
    private ExecutorService executorService;
    private ProgressBar progressBar;
    private boolean isFirstLoad = true;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_graph);

        // Инициализация графика
        lineChart = findViewById(R.id.lineChart);
        progressBar = findViewById(R.id.progressBar);

        // Настройка графика
        lineChart.getDescription().setEnabled(false);
        lineChart.getLegend().setEnabled(true);
        XAxis xAxis = lineChart.getXAxis();
        xAxis.setPosition(XAxis.XAxisPosition.BOTTOM);
        xAxis.setLabelCount(10, true);
        xAxis.setAxisMinimum(1f);
        YAxis yAxis = lineChart.getYAxis();
        yAxis.setAxisMinimum(0f);
        yAxis.setAxisMaximum(300f);
        lineChart.getAxisRight().setEnabled(false);

        // Инициализация ExecutorService и Handler
        executorService = Executors.newSingleThreadExecutor();
        handler = new Handler(Looper.getMainLooper());

        // Загрузка данных
        progressBar.setVisibility(View.VISIBLE);
        getDevices();

        // Периодическое обновление каждые 15 секунд
        handler.postDelayed(new Runnable() {

```

```

@Override
public void run() {
    getDevices();
    handler.postDelayed(this, 15000);
}
}, 15000);
}

private void getDevices() {
    executorService.execute(() -> {
        try {
            String response = tcpClient.sendDataString("get_devices");
            if (response.startsWith("DATA_")) {
                String jsonData = response.substring(5);
                JSONArray jsonArray = new JSONArray(jsonData);

                List<Entry> voltageEntries = new ArrayList<>();
                List<Entry> amperageEntries = new ArrayList<>();
                List<Entry> temperatureEntries = new ArrayList<>();

                for (int i = 0; i < jsonArray.length(); i++) {
                    JSONObject obj = jsonArray.getJSONObject(i);
                    float x = i + 1;
                    voltageEntries.add(new Entry(x, obj.getInt("voltage")));
                    amperageEntries.add(new Entry(x, obj.getInt("amperage")));
                    temperatureEntries.add(new Entry(x, obj.getInt("temperature")));
                }

                LineDataSet voltageDataSet = new LineDataSet(voltageEntries, "Напряжение");
                voltageDataSet.setColor(android.graphics.Color.RED);
                voltageDataSet.setLineWidth(2f);
                voltageDataSet.setDrawCircles(false);

                LineDataSet amperageDataSet = new LineDataSet(amperageEntries, "Ток");
                amperageDataSet.setColor(android.graphics.Color.GREEN);
                amperageDataSet.setLineWidth(2f);
                amperageDataSet.setDrawCircles(false);

                LineDataSet temperatureDataSet = new LineDataSet(temperatureEntries, "Температура");
                temperatureDataSet.setColor(android.graphics.Color.BLUE);
                temperatureDataSet.setLineWidth(2f);
                temperatureDataSet.setDrawCircles(false);

                LineData lineData = new LineData(voltageDataSet, amperageDataSet, temperatureDataSet);

                handler.post(() -> {
                    lineChart.setData(lineData);
                    lineChart.getXAxis().setAxisMaximum(jsonArray.length());
                    lineChart.invalidate();

                    if (isFirstLoad) {
                        progressBar.setVisibility(View.GONE);
                        lineChart.setVisibility(View.VISIBLE);
                        isFirstLoad = false;
                    }
                });
            }
        } catch (Exception e) {
            e.printStackTrace();
        }
    });
}

```

```

    });
}

@Override
protected void onDestroy() {
    super.onDestroy();
    handler.removeCallbacksAndMessages(null);
    executorService.shutdown();
}
}

}

————— (HistoryActivity.java) —————
package com.example.diplom_main;

import android.os.Bundle;
import android.os.Handler;
import android.os.Looper;
import android.view.View;
import android.widget.ProgressBar;

import androidx.appcompat.app.AppCompatActivity;
import androidx.recyclerview.widget.LinearLayoutManager;
import androidx.recyclerview.widget.RecyclerView;
import org.json.JSONArray;
import org.json.JSONObject;
import java.util.ArrayList;
import java.util.List;
import java.util.concurrent.ExecutorService;
import java.util.concurrent.Executors;
import static com.example.diplom_main.StaticData.tcpClient;

public class HistoryActivity extends AppCompatActivity {

    private RecyclerView historyRecyclerView;
    private HistoryAdapter historyAdapter;
    private List<HistoryItem> historyList;
    private Handler handler;
    private ExecutorService executorService;
    private ProgressBar progressBar;
    private boolean isFirstLoad = true;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_history);

        // Инициализация UI
        historyRecyclerView = findViewById(R.id.historyRecyclerView);
        progressBar = findViewById(R.id.progressBar);

        // Инициализация RecyclerView
        historyList = new ArrayList<>();
        historyAdapter = new HistoryAdapter(historyList);
        historyRecyclerView.setLayoutManager(new LinearLayoutManager(this));
        historyRecyclerView.setAdapter(historyAdapter);

        // Инициализация ExecutorService и Handler
        executorService = Executors.newSingleThreadExecutor();
        handler = new Handler(Looper.getMainLooper());
    }
}

```

```

// Начальная загрузка данных
progressBar.setVisibility(View.VISIBLE);
getHistory();

// Периодическое обновление каждые 15 секунд
handler.postDelayed(new Runnable() {
    @Override
    public void run() {
        getHistory();
        handler.postDelayed(this, 15000);
    }
}, 15000);
}

private void getHistory() {
    executorService.execute(() -> {
        try {
            String response = tcpClient.sendDataString("get_devices_history");
            if (response.startsWith("DATA_")) {
                String jsonData = response.substring(5);
                JSONArray jsonArray = new JSONArray(jsonData);
                List<HistoryItem> newHistory = new ArrayList<>();

                for (int i = 0; i < jsonArray.length(); i++) {
                    JSONObject obj = jsonArray.getJSONObject(i);
                    String id = obj.has("id") ? obj.getString("id") : "" + (i+1);
                    HistoryItem item = new HistoryItem(
                        obj.getString("creation_time"),
                        id,
                        obj.getString("name"),
                        obj.getString("turned"),
                        obj.getString("voltage"),
                        obj.getString("voltage_min"),
                        obj.getString("voltage_max"),
                        obj.getString("amperage"),
                        obj.getString("amperage_max"),
                        obj.getString("temperature"),
                        obj.getString("temperature_max"),
                        obj.getString("capacity"),
                        obj.getString("placement_id")
                    );
                    newHistory.add(item);
                }

                handler.post(() -> {
                    historyList.clear();
                    historyList.addAll(newHistory);
                    historyAdapter.notifyDataSetChanged();

                    if (isFirstLoad) {
                        progressBar.setVisibility(View.GONE);
                        historyRecyclerView.setVisibility(View.VISIBLE);
                        isFirstLoad = false;
                    }
                });
            }
        } catch (Exception e) {
            e.printStackTrace();
        }
    });
}

```

```

    }

    @Override
    protected void onDestroy() {
        super.onDestroy();
        handler.removeCallbacksAndMessages(null);
        executorService.shutdown();
    }
}

————— (HistoryAdapter.java) —————
package com.example.diplom_main;

import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.TextView;
import androidx.recyclerview.widget.RecyclerView;
import java.util.List;

public class HistoryAdapter extends RecyclerView.Adapter<HistoryAdapter.HistoryViewHolder> {

    private List<HistoryItem> historyList;

    public HistoryAdapter(List<HistoryItem> historyList) {
        this.historyList = historyList;
    }

    @Override
    public HistoryViewHolder onCreateViewHolder(ViewGroup parent, int viewType) {
        View view = LayoutInflater.from(parent.getContext())
            .inflate(R.layout.item_history, parent, false);
        return new HistoryViewHolder(view);
    }

    @Override
    public void onBindViewHolder(HistoryViewHolder holder, int position) {
        HistoryItem item = historyList.get(position);
        holder.creationTimeTextView.setText(item.getCreationTime());
        holder.idTextView.setText(item.getId());
        holder.nameTextView.setText(item.getName());
        holder.turnedList.setText(item.getTurned());
        holder.voltageTextView.setText(item.getVoltage());
        holder.voltageMinTextView.setText(item.getVoltageMin());
        holder.voltageMaxTextView.setText(item.getVoltageMax());
        holder.amperageTextView.setText(item.getAmperage());
        holder.amperageMaxTextView.setText(item.getAmperageMax());
        holder.temperatureTextView.setText(item.getTemperature());
        holder.temperatureMaxTextView.setText(item.getTemperatureMax());
        holder.capacityTextView.setText(item.getCapacity());
        holder.placementIdTextView.setText(item.getPlacementId());
    }

    @Override
    public int getItemCount() {
        return historyList.size();
    }

    static class HistoryViewHolder extends RecyclerView.ViewHolder {
        TextView creationTimeTextView, idTextView, nameTextView, turnedTextView, voltageTextView,

```



```
voltageMinTextView, voltageMaxTextView, amperageTextView, amperageMaxTextView,
temperatureTextView, temperatureMaxTextView, capacityTextView, placementIdTextView;
```

```
HistoryViewHolder(View itemView) {
    super(itemView);
    creationTimeTextView = itemView.findViewById(R.id.creationTimeTextView);
    idTextView = itemView.findViewById(R.id.idTextView);
    nameTextView = itemView.findViewById(R.id.nameTextView);
    turnedTextView = itemView.findViewById(R.id.turnedTextView);
    voltageTextView = itemView.findViewById(R.id.voltageTextView);
    voltageMinTextView = itemView.findViewById(R.id.voltageMinTextView);
    voltageMaxTextView = itemView.findViewById(R.id.voltageMaxTextView);
    amperageTextView = itemView.findViewById(R.id.amperageTextView);
    amperageMaxTextView = itemView.findViewById(R.id.amperageMaxTextView);
    temperatureTextView = itemView.findViewById(R.id.temperatureTextView);
    temperatureMaxTextView = itemView.findViewById(R.id.temperatureMaxTextView);
    capacityTextView = itemView.findViewById(R.id.capacityTextView);
    placementIdTextView = itemView.findViewById(R.id.placementIdTextView);
}
}
```

————— (HistoryItem.java) —————

```
package com.example.diplom_main;
```

```
public class HistoryItem {
    private String creationTime, id, name, turned, voltage, voltageMin, voltageMax,
        amperage, amperageMax, temperature, temperatureMax, capacity, placementId;

    public HistoryItem(String creationTime, String id, String name, String turned, String voltage,
        String voltageMin, String voltageMax, String amperage, String amperageMax,
        String temperature, String temperatureMax, String capacity, String placementId) {
        this.creationTime = creationTime;
        this.id = id;
        this.name = name;
        this.turned = turned;
        this.voltage = voltage;
        this.voltageMin = voltageMin;
        this.voltageMax = voltageMax;
        this.amperage = amperage;
        this.amperageMax = amperageMax;
        this.temperature = temperature;
        this.temperatureMax = temperatureMax;
        this.capacity = capacity;
        this.placementId = placementId;
    }
}
```

// Геттеры

```
public String getCreationTime() { return creationTime; }
public String getId() { return id; }
public String getName() { return name; }
public String getTurned() { return turned; }
public String getVoltage() { return voltage; }
public String getVoltageMin() { return voltageMin; }
public String getVoltageMax() { return voltageMax; }
public String getAmperage() { return amperage; }
public String getAmperageMax() { return amperageMax; }
public String getTemperature() { return temperature; }
public String getTemperatureMax() { return temperatureMax; }
public String getCapacity() { return capacity; }
public String getPlacementId() { return placementId; }
```



```

@Override
public void onClick(View v) {
    String login = loginEditText.getText().toString();
    String password = hashPassword(passwordEditText.getText().toString());
    String request = "log_" + login + "_" + password;

    AtomicReference<String[]> arrRef = new AtomicReference<>(new String[] {});

    // Показать ProgressBar
    progressBar.setVisibility(View.VISIBLE);

    executorService.execute(() -> {
        String response;
        try {
            response = tcpClient.sendDataString(request);
            arrRef.set(response.split("_"));
        } catch (Exception e) {
            mainHandler.post(() -> {
                Toast.makeText(MainActivity.this, e.getMessage(), Toast.LENGTH_LONG).show();
                // Скрыть ProgressBar в случае ошибки
                progressBar.setVisibility(View.GONE);
            });
            return;
        }

        mainHandler.post(() -> {
            String[] arr = arrRef.get();
            Log.d("MainActivity", arr[0] + " " + arr[1]);
            if (arr[0].equals("DATA") && arr[1].equals("true")) {
                // Успешная авторизация, переход на UserActivity
                Intent intent = new Intent(MainActivity.this, UserActivity.class);
                startActivity(intent);
                finish();
            } else if (arr[0].equals("DATA") && arr[1].equals("false")) {
                Toast.makeText(MainActivity.this, "Неверный логин или пароль",
                    Toast.LENGTH_SHORT).show();
            } else {
                Toast.makeText(MainActivity.this, arr[1], Toast.LENGTH_SHORT).show();
            }
            // Скрыть ProgressBar после завершения загрузки
            progressBar.setVisibility(View.GONE);
        });
    });
}

// Хеширование пароля с использованием SHA-256
private String hashPassword(String password) {
    try {
        MessageDigest digest = MessageDigest.getInstance("SHA-256");
        byte[] hash = digest.digest(password.getBytes());
        StringBuilder hexString = new StringBuilder();
        for (byte b : hash) {
            String hex = Integer.toHexString(0xff & b);
            if (hex.length() == 1) hexString.append('0');
            hexString.append(hex);
        }
        return hexString.toString();
    } catch (NoSuchAlgorithmException e) {

```

```

        e.printStackTrace();
        return "";
    }
}
}

```

(StaticData.java)

```

package com.example.diplom_main;

```

```

public class StaticData {
    public static TcpClient tcpClient;
    public static boolean firstLaunch = true;
}

```

(TcpClient.java)

```

package com.example.diplom_main;

```

```

import android.util.Log;
import org.json.JSONArray;
import org.json.JSONException;
import org.json.JSONObject;
import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;
import java.io.PrintWriter;
import java.net.Socket;
import java.security.MessageDigest;
import java.security.NoSuchAlgorithmException;
import java.util.ArrayList;
import java.util.List;

```

```

public class TcpClient {
    private static final String TAG = "TcpClient";
    private String host;
    private int port;
    private Socket socket;
    private PrintWriter out;
    private BufferedReader in;
    private boolean error = false;

    public TcpClient(String host, int port) {
        this.host = host;
        this.port = port;
        connect();
    }

    private void connect() {
        try {
            Log.d(TAG, "Подключение...");
            socket = new Socket(host, port);
            out = new PrintWriter(socket.getOutputStream(), true);
            in = new BufferedReader(new InputStreamReader(socket.getInputStream()));
            Log.d(TAG, "Подключено к серверу");
            error = false;
        } catch (Exception e) {
            Log.e(TAG, "Ошибка подключения: " + e.getMessage());
            error = true;
        }
    }

    public boolean checkConnection() {

```

```

        boolean a = socket != null && socket.isConnected() && !socket.isClosed();
        return a;
    }

    public String sendDataString(String msg) {
        if (error) {
            return "ERROR_ Не удалось подключиться к серверу.";
        }

        try {
            // Отправка данных
            out.println(msg);
            out.flush();
            Log.d(TAG, "Данные отправлены: " + msg);

            // Ожидание 3 секунды перед чтением ответа
            Thread.sleep(3000);
            Log.d(TAG, "Ожидание завершено.");

            // Чтение ответа
            StringBuilder jsonBuilder = new StringBuilder();
            String line;

            try {
                while ((line = in.readLine()) != null && !line.isEmpty()) {
                    jsonBuilder.append(line);
                    Log.d(TAG, "Полученные данные: " + line);
                    if (line.equals("{}")) break;
                }
            } catch (IOException e) {
                Log.e(TAG, "Ошибка чтения данных: " + e.getMessage());
                return "ERROR_ Ошибка чтения данных.";
            }
            Log.d(TAG, "Приступил к JSON");
            String jsonData = jsonBuilder.toString();
            Log.d(TAG, "Полученный JSON: " + jsonData);

            if (jsonData.isEmpty()) {
                Log.e(TAG, "Пустой ответ от сервера.");
                return "ERROR_ Пустой ответ от сервера.";
            }

            JSONObject jsonResponse = new JSONObject(jsonData);

            // Проверка контрольной суммы
            String receivedChecksum = jsonResponse.getString("checksum");
            jsonResponse.remove("checksum");
            String dataStr = jsonResponse.getString("data");
            String calculatedChecksum = calculateChecksum(dataStr);

            if (!receivedChecksum.equals(calculatedChecksum)) {
                Log.e(TAG, "Несовпадение контрольных сумм.");
                return "ERROR_ Несовпадение контрольных сумм.";
            } else if (jsonResponse.getString("type").equals("error")) {
                return "ERROR_" + jsonResponse.getString("data");
            } else {
                return "DATA_" + jsonResponse.getString("data");
            }
        } catch (JSONException e) {
            Log.e(TAG, "Ошибка JSON: " + e.getMessage());
        }
    }

```

```

        return "ERROR_Ошибка JSON.";
    } catch (InterruptedException e) {
        Log.e(TAG, "Ошибка прерывания потока: " + e.getMessage());
        Thread.currentThread().interrupt(); // Восстанавливаем статус прерывания потока
        return "ERROR_Прервано ожидание.";
    } catch (Exception e) {
        Log.e(TAG, "Неожиданная ошибка: " + e.getMessage());
        return "ERROR_Неожиданная ошибка.";
    }
}

private String calculateChecksum(String data) {
    try {
        MessageDigest digest = MessageDigest.getInstance("SHA-256");
        byte[] hash = digest.digest(data.getBytes());
        StringBuilder hexString = new StringBuilder();
        for (byte b : hash) {
            String hex = Integer.toHexString(0xff & b);
            if (hex.length() == 1) hexString.append('0');
            hexString.append(hex);
        }
        return hexString.toString();
    } catch (NoSuchAlgorithmException e) {
        e.printStackTrace();
        return "";
    }
}

public void close() {
    try {
        if (socket != null) socket.close();
        if (out != null) out.close();
        if (in != null) in.close();
    } catch (IOException e) {
        Log.e(TAG, "Ошибка при закрытии соединения: " + e.getMessage());
    }
}
}

}

----- (UserActivity.java) -----
package com.example.diplom_main;

import android.content.Intent;
import android.os.Bundle;
import android.os.Handler;
import android.os.Looper;
import android.view.View;
import android.widget.Button;
import android.widget.ProgressBar;
import android.widget.TextView;
import androidx.appcompat.app.AppCompatActivity;
import androidx.recyclerview.widget.LinearLayoutManager;
import androidx.recyclerview.widget.RecyclerView;
import org.json.JSONArray;
import org.json.JSONObject;
import java.util.ArrayList;
import java.util.List;
import java.util.concurrent.ExecutorService;
import java.util.concurrent.Executors;
import static com.example.diplom_main.StaticData.tcpClient;

```

```

public class UserActivity extends AppCompatActivity {

    private RecyclerView devicesRecyclerView;
    private Button historyButton, workButton;
    private DeviceAdapter deviceAdapter;
    private List<Device> deviceList;
    private Handler handler;
    private ExecutorService executorService;
    private ProgressBar progressBar;
    private boolean isActivityVisible = true;
    private Runnable dataFetchRunnable;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_user);

        // Инициализация UI
        devicesRecyclerView = findViewById(R.id.devicesRecyclerView);
        historyButton = findViewById(R.id.historyButton);
        workButton = findViewById(R.id.workButton);
        progressBar = findViewById(R.id.progressBar);

        // Инициализация RecyclerView
        deviceList = new ArrayList<>();
        deviceAdapter = new DeviceAdapter(deviceList);
        devicesRecyclerView.setLayoutManager(new LinearLayoutManager(this));
        devicesRecyclerView.setAdapter(deviceAdapter);

        // Обработчики кнопок
        historyButton.setOnClickListener(v -> onHistoryButtonClicked());
        workButton.setOnClickListener(v -> onWorkButtonClicked());

        // Инициализация ExecutorService и Handler
        executorService = Executors.newSingleThreadExecutor();
        handler = new Handler(Looper.getMainLooper());

        // Показать ProgressBar и получить данные
        progressBar.setVisibility(View.VISIBLE);
        getDevices();

        // Создание Runnable для периодического обновления данных
        dataFetchRunnable = new Runnable() {
            @Override
            public void run() {
                if (isActivityVisible) {
                    getDevices();
                    handler.postDelayed(this, 5000); // Каждые 5 секунд
                }
            }
        };

        // Запуск периодического обновления данных
        handler.postDelayed(dataFetchRunnable, 5000);
    }

    private void getDevices() {
        executorService.execute(() -> {
            try {

```

```

String response = tcpClient.sendDataString("get_devices");
if (response.startsWith("DATA_")) {
    String jsonData = response.substring(5);
    JSONArray jsonArray = new JSONArray(jsonData);
    List<Device> newDevices = new ArrayList<>();

    for (int i = 0; i < jsonArray.length(); i++) {
        JSONObject obj = jsonArray.getJSONObject(i);
        String id = obj.has("id") ? obj.getString("id") : "" + (i+1);
        Device device = new Device(
            id,
            obj.getString("name"),
            obj.getString("turned"),
            obj.getString("voltage"),
            obj.getString("voltage_min"),
            obj.getString("voltage_max"),
            obj.getString("amperage"),
            obj.getString("amperage_max"),
            obj.getString("temperature"),
            obj.getString("temperature_max"),
            obj.getString("capacity"),
            obj.getString("placement_id")
        );
        newDevices.add(device);
    }

    handler.post(() -> {
        deviceList.clear();
        deviceList.addAll(newDevices);
        deviceAdapter.notifyDataSetChanged();
        progressBar.setVisibility(View.GONE); // Скрыть ProgressBar после получения данных
    });
}
} catch (Exception e) {
    e.printStackTrace();
}
});
}

private void onHistoryButtonClicked() {
    Intent intent = new Intent(this, HistoryActivity.class);
    startActivity(intent);
}

private void onWorkButtonClicked() {
    Intent intent = new Intent(this, WorkActivity.class);
    startActivity(intent);
}

@Override
protected void onPause() {
    super.onPause();
    isActivityVisible = false;
    handler.removeCallbacks(dataFetchRunnable);
}

@Override
protected void onResume() {
    super.onResume();
    isActivityVisible = true;
}

```



```

        handler.postDelayed(dataFetchRunnable, 5000);
    }

    @Override
    protected void onDestroy() {
        super.onDestroy();
        handler.removeCallbacksAndMessages(null);
        executorService.shutdown();
    }
}

```

————— (WorkActivity.java) —————
package com.example.diplom_main;

```

import android.content.Intent;
import android.os.Bundle;
import android.os.Environment;
import android.os.Handler;
import android.os.Looper;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.ProgressBar;
import android.widget.Toast;
import androidx.appcompat.app.AppCompatActivity;
import androidx.recyclerview.widget.LinearLayoutManager;
import androidx.recyclerview.widget.RecyclerView;
import org.json.JSONArray;
import org.json.JSONObject;
import java.io.File;
import java.io.FileWriter;
import java.util.ArrayList;
import java.util.List;
import java.util.concurrent.ExecutorService;
import java.util.concurrent.Executors;
import static com.example.diplom_main.StaticData.tcpClient;

public class WorkActivity extends AppCompatActivity {

    private RecyclerView filterRecyclerView;
    private EditText voltageMinEditText, voltageMaxEditText, temperatureMinEditText,
        temperatureMaxEditText, amperageMinEditText, amperageMaxEditText;
    private Button filterButton, graphButton, importCsvButton, importJsonButton;
    private DeviceAdapter deviceAdapter;
    private List<Device> deviceList;
    private ExecutorService executorService;
    private Handler handler;
    private ProgressBar progressBar;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_work);

        // Инициализация UI
        filterRecyclerView = findViewById(R.id.filterRecyclerView);
        voltageMinEditText = findViewById(R.id.voltageMinEditText);
        voltageMaxEditText = findViewById(R.id.voltageMaxEditText);
        temperatureMinEditText = findViewById(R.id.temperatureMinEditText);
        temperatureMaxEditText = findViewById(R.id.temperatureMaxEditText);
    }
}

```

```

    amperageMinEditText = findViewById(R.id.amperageMinEditText);
    amperageMaxEditText = findViewById(R.id.amperageMaxEditText);
    filterButton = findViewById(R.id.filterButton);
    graphButton = findViewById(R.id.graphButton);
    importCsvButton = findViewById(R.id.importCsvButton);
    importJsonButton = findViewById(R.id.importJsonButton);
    progressBar = findViewById(R.id.progressBar);

    // Инициализация RecyclerView
    deviceList = new ArrayList<>();
    deviceAdapter = new DeviceAdapter(deviceList);
    filterRecyclerView.setLayoutManager(new LinearLayoutManager(this));
    filterRecyclerView.setAdapter(deviceAdapter);

    // Инициализация ExecutorService и Handler
    executorService = Executors.newSingleThreadExecutor();
    handler = new Handler(Looper.getMainLooper());

    // Начальная загрузка данных
    getDevices("get_devices");

    // Обработчики кнопок
    filterButton.setOnClickListener(v -> onFilterButtonClicked());
    graphButton.setOnClickListener(v -> onGraphButtonClicked());
    importCsvButton.setOnClickListener(v -> onImportCsvButtonClicked());
    importJsonButton.setOnClickListener(v -> onImportJsonButtonClicked());
}

private void showProgressBar() {
    progressBar.setVisibility(View.VISIBLE);
    filterRecyclerView.setVisibility(View.GONE);
}

private void hideProgressBar() {
    progressBar.setVisibility(View.GONE);
    filterRecyclerView.setVisibility(View.VISIBLE);
}

private void onFilterButtonClicked() {
    String voltageMin = voltageMinEditText.getText().toString().isEmpty() ? "0" :
voltageMinEditText.getText().toString();
    String voltageMax = voltageMaxEditText.getText().toString().isEmpty() ? "300" :
voltageMaxEditText.getText().toString();
    String amperageMin = amperageMinEditText.getText().toString().isEmpty() ? "0" :
amperageMinEditText.getText().toString();
    String amperageMax = amperageMaxEditText.getText().toString().isEmpty() ? "300" :
amperageMaxEditText.getText().toString();
    String temperatureMin = temperatureMinEditText.getText().toString().isEmpty() ? "0" :
temperatureMinEditText.getText().toString();
    String temperatureMax = temperatureMaxEditText.getText().toString().isEmpty() ? "300" :
temperatureMaxEditText.getText().toString();

    String filterStr = "filter_devices_v_" + voltageMin + "_" + voltageMax +
        "_a_" + amperageMin + "_" + amperageMax +
        "_t_" + temperatureMin + "_" + temperatureMax;
    getDevices(filterStr);
}

private void onGraphButtonClicked() {
    Intent intent = new Intent(this, GraphActivity.class);

```

```

        startActivity(intent);
    }

    private void onImportCsvButtonClicked() {
        showProgressBar();
        executorService.execute(() -> {
            try {
                String response = tcpClient.sendDataString("get_devices");
                if (response.startsWith("DATA_")) {
                    String jsonData = response.substring(5);
                    JSONArray jsonArray = new JSONArray(jsonData);
                    StringBuilder csvContent = new StringBuilder();
                    JSONObject firstObject = jsonArray.getJSONObject(0);
                    String[] headers = {"id", "name", "turned", "voltage", "voltage_min", "voltage_max",
                        "amperage", "amperage_max", "temperature", "temperature_max", "capacity",
                        "placement_id"};
                    csvContent.append(String.join(",", headers)).append("\n");

                    for (int i = 0; i < jsonArray.length(); i++) {
                        JSONObject obj = jsonArray.getJSONObject(i);
                        List<String> row = new ArrayList<>();
                        for (String header : headers) {
                            if (header == "id") {
                                String id = obj.has(header) ? obj.getString(header) : ""+(i+1);
                                row.add(id);
                            }
                            else row.add(obj.getString(header));
                        }
                        csvContent.append(String.join(",", row)).append("\n");
                    }

                    File file = new
File(Environment.getExternalStoragePublicDirectory(Environment.DIRECTORY_DOWNLOADS),
"devices.csv");
                    try (FileWriter writer = new FileWriter(file)) {
                        writer.write(csvContent.toString());
                        handler.post(() -> {
                            Toast.makeText(WorkActivity.this, "CSV сохранен в Downloads",
Toast.LENGTH_SHORT).show();
                            hideProgressBar();
                        });
                    }
                }
            } catch (Exception e) {
                handler.post(() -> {
                    Toast.makeText(WorkActivity.this, "Ошибка: " + e.getMessage(),
Toast.LENGTH_SHORT).show();
                    hideProgressBar();
                });
            }
        });
    }

    private void onImportJsonButtonClicked() {
        showProgressBar();
        executorService.execute(() -> {
            try {
                String response = tcpClient.sendDataString("get_devices");
                if (response.startsWith("DATA_")) {
                    String jsonData = response.substring(5);

```

```

        File file = new
File(Environment.getExternalStoragePublicDirectory(Environment.DIRECTORY_DOWNLOADS),
"devices.json");
        try (FileWriter writer = new FileWriter(file)) {
            writer.write(jsonData);
            handler.post() -> {
                Toast.makeText(WorkActivity.this, "JSON сохранен в Downloads",
Toast.LENGTH_SHORT).show();
                hideProgressBar();
            });
        }
    } catch (Exception e) {
        handler.post() -> {
            Toast.makeText(WorkActivity.this, "Ошибка: " + e.getMessage(),
Toast.LENGTH_SHORT).show();
            hideProgressBar();
        });
    }
}

private void getDevices(String msg) {
    showProgressBar();
    executorService.execute() -> {
        try {
            String response = tcpClient.sendDataString(msg);
            if (response.startsWith("DATA_")) {
                String jsonData = response.substring(5);
                JSONArray jsonArray = new JSONArray(jsonData);
                List<Device> newDevices = new ArrayList<>();

                for (int i = 0; i < jsonArray.length(); i++) {
                    JSONObject obj = jsonArray.getJSONObject(i);
                    String id = obj.has("id") ? obj.getString("id") : "" + (i+1);
                    Device device = new Device(
                        id,
                        obj.getString("name"),
                        obj.getString("turned"),
                        obj.getString("voltage"),
                        obj.getString("voltage_min"),
                        obj.getString("voltage_max"),
                        obj.getString("amperage"),
                        obj.getString("amperage_max"),
                        obj.getString("temperature"),
                        obj.getString("temperature_max"),
                        obj.getString("capacity"),
                        obj.getString("placement_id")
                    );
                    newDevices.add(device);
                }

                handler.post() -> {
                    deviceList.clear();
                    deviceList.addAll(newDevices);
                    deviceAdapter.notifyDataSetChanged();
                    hideProgressBar();
                });
            }
        } catch (Exception e) {

```

```

        handler.post(() -> {
            Toast.makeText(WorkActivity.this, "Ошибка: " + e.getMessage(),
Toast.LENGTH_SHORT).show();
            hideProgressBar();
        });
    }

    @Override
    protected void onDestroy() {
        super.onDestroy();
        executorService.shutdown();
    }
}

```

ПРИЛОЖЕНИЕ Б. СЦЕНАРИЙ И РЕЗУЛЬТАТЫ ТЕСТОВЫХ ИСПЫТАНИЙ

АННОТАЦИЯ

В данном программном документе представлены сценарий и результаты тестовых испытаний по разработке программы «DeviceMonitor». Документ содержит основные разделы, включая: схему тестирования и тестовые примеры, и результаты.

СОДЕРЖАНИЕ

1. Схема тестирования	3
2. Тестовые примеры и результаты	4
3. Схема Тестирования	15

1. СХЕМА ТЕСТИРОВАНИЯ

Общие предпосылки для проведения тестирования:

1. Все таблицы базы данных должны содержать по 1-3 записи;
2. База данных должна быть активной.

На рисунке 25 изображена схема тестирования программы.

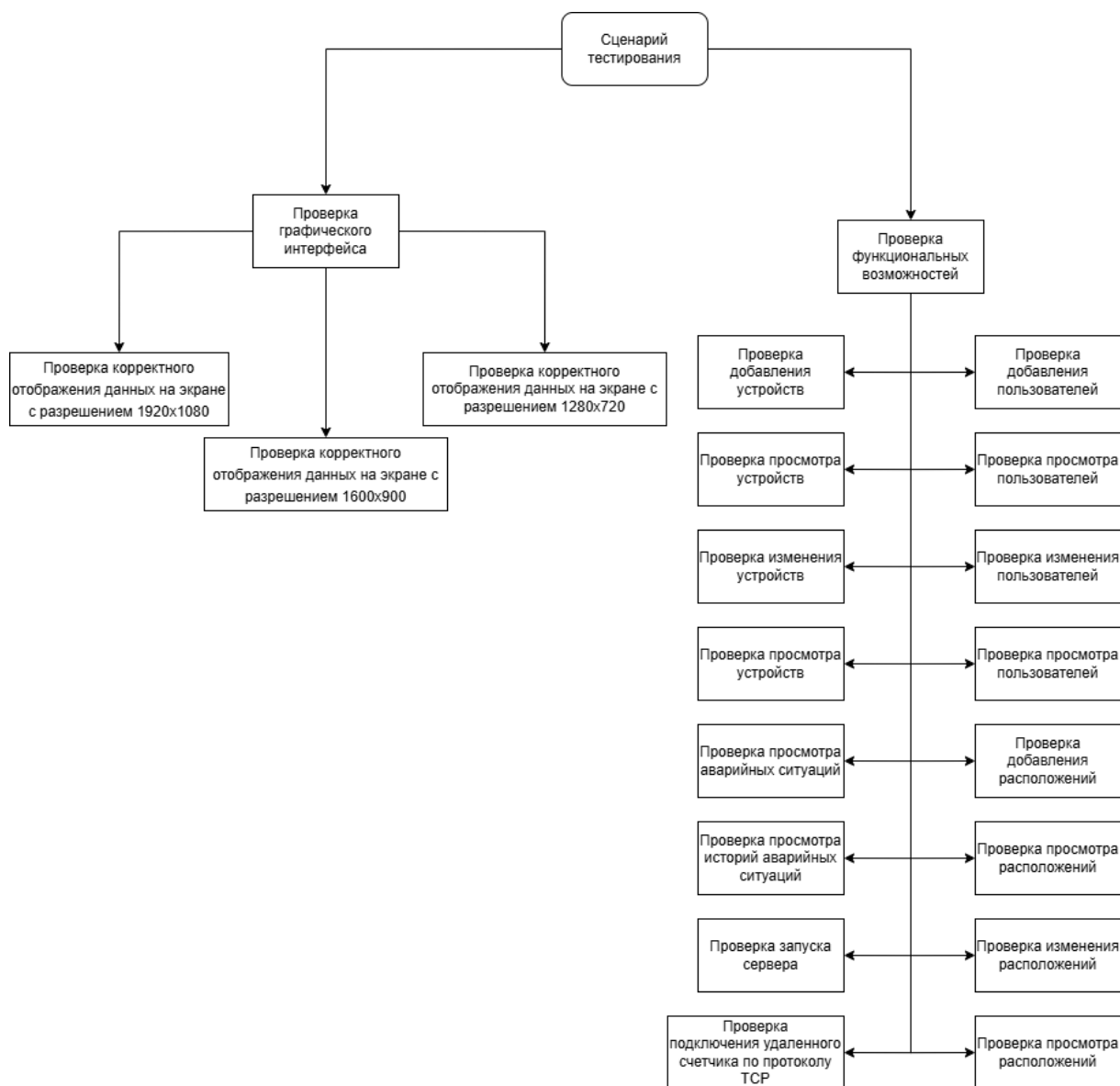


Рисунок 25 - Схема тестирования

2. ТЕСТОВЫЕ ПРИМЕРЫ И РЕЗУЛЬТАТЫ

Тестирования интерфейса производится по схеме на таблице 7.

Таблица 7 – Схема тестирования интерфейса

№ п/п	Предпосылки	Шаги тестирования	Данные тестирования	Ожидаемый результат	Фактический результат	Приоритет
1	2	3	4	5	6	7
Тестирование серверной части						
Проверка корректного отображения интерфейса на устройстве с разрешением 1920x1080						
1		Открыть страницу сервера	Не предусмотрено	Страница и данные на ней верно отображаются и не вылезают за рамки элементов	Страница и данные на ней верно отображаются и не вылезают за рамки элементов.	Высокий
2		Нажать на кнопку «Админ-панель»	admin, admin1	Страница и данные на ней верно отображаются и не вылезают за рамки элементов	Страница и данные на ней верно отображаются и не вылезают за рамки элементов.	Высокий
3		Нажать на кнопку «Войти»	5000, «ШИБП-1-1», «Альфа», 220, 209, 231, 50, 70, 37, 80	Страница и данные на ней верно отображаются и не вылезают за рамки элементов	Страница и данные на ней верно отображаются и не вылезают за рамки элементов	Высокий
4		Нажать на кнопку «Пользователи»	Vasek123, Pom123_xs	Страница и данные на ней верно отображаются и не вылезают за рамки элементов	Страница и данные на ней верно отображаются и не вылезают за рамки элементов	Средний
5		Нажать на кнопку «История аварийных случаев»	Не предусмотрено	Страница и данные на ней верно отображаются и не вылезают за	Страница и данные на ней верно отображаются и не вылезают за	Низкий

№ п/п	Предпосылки	Шаги тестирования	Данные тестирования	Ожидаемый результат	Фактический результат	Приоритет
				рамки элементов	рамки элементов	
6		Нажать на кнопку «Расположения»	Альфа	Страница и данные на ней верно отображаются и не вылезают за рамки элементов	Страница и данные на ней верно отображаются и не вылезают за рамки элементов	Низкий
Проверка корректного отображения интерфейса на устройстве с разрешением 1600x900						
1		Открыть страницу сервера	Не предусмотрено	Страница и данные на ней верно отображаются и не вылезают за рамки элементов	Страница и данные на ней верно отображаются и не вылезают за рамки элементов.	Высокий
2		Нажать на кнопку «Админ-панель»	admin, admin1	Страница и данные на ней верно отображаются и не вылезают за рамки элементов	Страница и данные на ней верно отображаются и не вылезают за рамки элементов.	Высокий
3		Нажать на кнопку «Войти»	5000, «ШИБП-1-1», «Альфа», 220, 209, 231, 50, 70, 37, 80	Страница и данные на ней верно отображаются и не вылезают за рамки элементов	Страница и данные на ней верно отображаются и не вылезают за рамки элементов	Высокий
4		Нажать на кнопку «Пользователи»	Vasek123, Pom123_xs	Страница и данные на ней верно отображаются и не вылезают за рамки элементов	Страница и данные на ней верно отображаются и не вылезают за рамки элементов	Средний

№ п/п	Предпосылки	Шаги тестирования	Данные тестирования	Ожидаемый результат	Фактический результат	Приоритет
5		Нажать на кнопку «История аварийных случаев»	Не предусмотрено	Страница и данные на ней верно отображаются и не вылезают за рамки элементов	Страница и данные на ней верно отображаются и не вылезают за рамки элементов	Низкий
6		Нажать на кнопку «Расположения»	Альфа	Страница и данные на ней верно отображаются и не вылезают за рамки элементов	Страница и данные на ней верно отображаются и не вылезают за рамки элементов	Низкий
Проверка корректного отображения интерфейса на устройстве с разрешением 1280x720						
1		Открыть страницу сервера	Не предусмотрено	Страница и данные на ней верно отображаются и не вылезают за рамки элементов	Страница и данные на ней верно отображаются и не вылезают за рамки элементов.	Высокий
2		Нажать на кнопку «Админ-панель»	admin, admin1	Страница и данные на ней верно отображаются и не вылезают за рамки элементов	Страница и данные на ней верно отображаются и не вылезают за рамки элементов.	Высокий
3		Нажать на кнопку «Войти»	5000, «ШИБП-1-1», «Альфа», 220, 209, 231, 50, 70, 37, 80	Страница и данные на ней верно отображаются и не вылезают за рамки элементов	Страница и данные на ней верно отображаются и не вылезают за рамки элементов	Высокий
4		Нажать на кнопку «Пользователи»	Vasek123, Pom123_xs	Страница и данные на ней верно отображаются	Страница и данные на ней верно отображаются	Средний

№ п/п	Предпосылки	Шаги тестирования	Данные тестирования	Ожидаемый результат	Фактический результат	Приоритет
				ся и не вылезают за рамки элементов	ся и не вылезают за рамки элементов	
5		Нажать на кнопку «История аварийных случаев»	Не предусмотрено	Страница и данные на ней верно отображаются и не вылезают за рамки элементов	Страница и данные на ней верно отображаются и не вылезают за рамки элементов	Низкий
6		Нажать на кнопку «Расположения»	Альфа	Страница и данные на ней верно отображаются и не вылезают за рамки элементов	Страница и данные на ней верно отображаются и не вылезают за рамки элементов	Низкий
Тестирование клиентской Desktop части						
Проверка корректного отображения интерфейса на устройстве с разрешением 1920x1080						
1	На введенном адресе и порте запущен сервер	Открыть окно ввода IP	168.192.0.104, 1234	Страница и данные на ней верно отображаются и не вылезают за рамки элементов	Страница и данные на ней верно отображаются и не вылезают за рамки элементов	Высокий
2	Открыта окно авторизации	Ввести логин и пароль в поля	Vasya123, xscxz2231	Страница и данные на ней верно отображаются и не вылезают за рамки элементов	Страница и данные на ней верно отображаются и не вылезают за рамки элементов	Высокий
3	Пользователь авторизован, открыто главное окно		Не предусмотрено	Страница и данные на ней верно отображаются и не вылезают за рамки элементов	Страница и данные на ней верно отображаются и не вылезают за рамки элементов	Высокий

№ п/п	Предпосылки	Шаги тестирования	Данные тестирования	Ожидаемый результат	Фактический результат	Приоритет
4	Администратор авторизован, открыто главное окно	Нажать «Работа с данными», ввести фильтр, нажать «Отфильтровать», нажать «Сохранить в виде json», «Сохранить в виде csv»	200, 230, 10, 40, 20, 50 (слева направо)	Страница и данные на ней верно отображаются и не вылезают за рамки элементов	Страница и данные на ней верно отображаются и не вылезают за рамки элементов	Средний
5	Администратор авторизован, открыто окно работы с данными	Нажать «График»	Не предусмотрено	Страница и данные на ней верно отображаются и не вылезают за рамки элементов	Страница и данные на ней верно отображаются и не вылезают за рамки элементов	Низкий
6	Пользователь авторизован, открыто главное окно	Нажать «История аварийных устройств»	Не предусмотрено	Страница и данные на ней верно отображаются и не вылезают за рамки элементов	Страница и данные на ней верно отображаются и не вылезают за рамки элементов	Низкий
Проверка корректного отображения интерфейса на устройстве с разрешением 1600x900						
1	На введенном адресе и порте запущен сервер	Открыть окно ввода IP	168.192.0.104, 1234	Страница и данные на ней верно отображаются и не вылезают за рамки элементов	Страница и данные на ней верно отображаются и не вылезают за рамки элементов	Высокий
2	Открыто окно авторизации	Ввести логин и пароль в поля	Vasya123, xscxz2231	Страница и данные на ней верно отображаются и не вылезают за рамки элементов	Страница и данные на ней верно отображаются и не вылезают за рамки элементов	Высокий

№ п/п	Предпосылки	Шаги тестирования	Данные тестирования	Ожидаемый результат	Фактический результат	Приоритет
3	Пользователь авторизован, открыто главное окно		Не предусмотрено	Страница и данные на ней верно отображаются и не вылезают за рамки элементов	Страница и данные на ней верно отображаются и не вылезают за рамки элементов	Высокий
4	Администратор авторизован, открыто главное окно	Нажать «Работа с данными», ввести фильтр, нажать «Отфильтровать», нажать «Сохранить в виде json», «Сохранить в виде csv»	200, 230, 10, 40, 20, 50 (слева направо)	Страница и данные на ней верно отображаются и не вылезают за рамки элементов	Страница и данные на ней верно отображаются и не вылезают за рамки элементов	Средний
5	Администратор авторизован, открыто окно работы с данными	Нажать «График»	Не предусмотрено	Страница и данные на ней верно отображаются и не вылезают за рамки элементов	Страница и данные на ней верно отображаются и не вылезают за рамки элементов	Низкий
6	Пользователь авторизован, открыто главное окно	Нажать «История аварийных устройств»	Не предусмотрены	Страница и данные на ней верно отображаются и не вылезают за рамки элементов	Страница и данные на ней верно отображаются и не вылезают за рамки элементов	Низкий
Проверка корректного отображения интерфейса на устройстве с разрешением 1280x720						
1	На введенном адресе и порте запущен сервер	Открыть окно ввода IP	168.192.0.104, 1234	Страница и данные на ней верно отображаются и не вылезают за рамки элементов	Страница и данные на ней верно отображаются и не вылезают за рамки элементов	Высокий

№ п/п	Предпосылки	Шаги тестирования	Данные тестирования	Ожидаемый результат	Фактический результат	Приоритет
2	Открыта окно авторизации	Ввести логин и пароль в поля	Vasya123, xscxz2231	Страница и данные на ней верно отображаются и не вылезают за рамки элементов	Страница и данные на ней верно отображаются и не вылезают за рамки элементов	Высокий
3	Пользователь авторизован, открыто главное окно		Не предусмотрено	Страница и данные на ней верно отображаются и не вылезают за рамки элементов	Страница и данные на ней верно отображаются и не вылезают за рамки элементов	Высокий
4	Администратор авторизован, открыто главное окно	Нажать «Работа с данными», ввести фильтр, нажать «Отфильтровать», нажать «Сохранить в виде json», «Сохранить в виде csv»	200, 230, 10, 40, 20, 50 (слева направо)	Страница и данные на ней верно отображаются и не вылезают за рамки элементов	Страница и данные на ней верно отображаются и не вылезают за рамки элементов	Средний
5	Администратор авторизован, открыто окно работы с данными	Нажать «График»	Не предусмотрено	Страница и данные на ней верно отображаются и не вылезают за рамки элементов	Страница и данные на ней верно отображаются и не вылезают за рамки элементов	Низкий
6	Пользователь авторизован, открыто главное окно	Нажать «История аварийных устройств»	Не предусмотрены	Страница и данные на ней верно отображаются и не вылезают за рамки элементов	Страница и данные на ней верно отображаются и не вылезают за рамки элементов	Низкий
Тестирование клиентской Android части						

№ п/п	Предпосылки	Шаги тестирования	Данные тестирования	Ожидаемый результат	Фактический результат	Приоритет
Проверка корректного отображения интерфейса на устройстве с разрешением 2400x1080						
1	На введенном адресе и порте запущен сервер	Открыть окно ввода IP	168.192.0.104, 1234	Страница и данные на ней верно отображаются и не вылезают за рамки элементов	Страница и данные на ней верно отображаются и не вылезают за рамки элементов	Высокий
2	Открыта окно авторизации	Ввести логин и пароль в поля	Vasya123, xscxz2231	Страница и данные на ней верно отображаются и не вылезают за рамки элементов	Страница и данные на ней верно отображаются и не вылезают за рамки элементов	Высокий
3	Пользователь авторизован, открыто главное окно		Не предусмотрено	Страница и данные на ней верно отображаются и не вылезают за рамки элементов	Страница и данные на ней верно отображаются и не вылезают за рамки элементов	Высокий
4	Администратор авторизован, открыто главное окно	Нажать «Работа с данными», ввести фильтр, нажать «Отфильтровать», нажать «Сохранить в виде json», «Сохранить в виде csv»	200, 230, 10, 40, 20, 50 (слева направо)	Страница и данные на ней верно отображаются и не вылезают за рамки элементов	Страница и данные на ней верно отображаются и не вылезают за рамки элементов	Средний
5	Администратор авторизован, открыто окно работы с данными	Нажать «График»	Не предусмотрено	Страница и данные на ней верно отображаются и не вылезают за рамки элементов	Страница и данные на ней верно отображаются и не вылезают за рамки элементов	Низкий

№ п/п	Предпосылки	Шаги тестирования	Данные тестирования	Ожидаемый результат	Фактический результат	Приоритет
6	Пользователь авторизован, открыто главное окно	Нажать «История аварийных устройств»	Не предусмотрены	Страница и данные на ней верно отображаются и не вылезают за рамки элементов	Страница и данные на ней верно отображаются и не вылезают за рамки элементов	Низкий
Проверка корректного отображения интерфейса на устройстве с разрешением 1080x1920						
1	На введенном адресе и порте запущен сервер	Открыть окно ввода IP	168.192.0.104, 1234	Страница и данные на ней верно отображаются и не вылезают за рамки элементов	Страница и данные на ней верно отображаются и не вылезают за рамки элементов	Высокий
2	Открыта окно авторизации	Ввести логин и пароль в поля	Vasya123, xscxz2231	Страница и данные на ней верно отображаются и не вылезают за рамки элементов	Страница и данные на ней верно отображаются и не вылезают за рамки элементов	Высокий
3	Пользователь авторизован, открыто главное окно		Не предусмотрено	Страница и данные на ней верно отображаются и не вылезают за рамки элементов	Страница и данные на ней верно отображаются и не вылезают за рамки элементов	Высокий
4	Администратор авторизован, открыто главное окно	Нажать «Работа с данными», ввести фильтр, нажать «Отфильтровать», нажать «Сохранить в виде json», «Сохранить в виде csv»	200, 230, 10, 40, 20, 50 (слева направо)	Страница и данные на ней верно отображаются и не вылезают за рамки элементов	Страница и данные на ней верно отображаются и не вылезают за рамки элементов	Средний

№ п/п	Предпосылки	Шаги тестирования	Данные тестирования	Ожидаемый результат	Фактический результат	Приоритет
5	Администратор авторизован, открыто окно работы с данными	Нажать «График»	Не предусмотрено	Страница и данные на ней верно отображаются и не вылезают за рамки элементов	Страница и данные на ней верно отображаются и не вылезают за рамки элементов	Низкий
6	Пользователь авторизован, открыто главное окно	Нажать «История аварийных устройств»	Не предусмотрено	Страница и данные на ней верно отображаются и не вылезают за рамки элементов	Страница и данные на ней верно отображаются и не вылезают за рамки элементов	Низкий
Проверка корректного отображения интерфейса на устройстве с разрешением 2560x1440						
1	На введенном адресе и порте запущен сервер	Открыть окно ввода IP	168.192.0.104, 1234	Страница и данные на ней верно отображаются и не вылезают за рамки элементов	Страница и данные на ней верно отображаются и не вылезают за рамки элементов	Высокий
2	Открыта окно авторизации	Ввести логин и пароль в поля	Vasya123, xscxz2231	Страница и данные на ней верно отображаются и не вылезают за рамки элементов	Страница и данные на ней верно отображаются и не вылезают за рамки элементов	Высокий
3	Пользователь авторизован, открыто главное окно		Не предусмотрено	Страница и данные на ней верно отображаются и не вылезают за рамки элементов	Страница и данные на ней верно отображаются и не вылезают за рамки элементов	Высокий
4	Администратор авторизован, открыто главное окно	Нажать «Работа с данными», ввести фильтр, нажать	200, 230, 10, 40, 20, 50 (слева направо)	Страница и данные на ней верно отображаются и не вылезают за	Страница и данные на ней верно отображаются и не вылезают за	Средний

№ п/п	Предпосылки	Шаги тестирования	Данные тестирования	Ожидаемый результат	Фактический результат	Приоритет
		«Отфильтровать», нажать «Сохранить в виде json», «Сохранить в виде csv»		рамки элементов	рамки элементов	
5	Администратор авторизован , открыто окно работы с данными	Нажать «График»	Не предусмотрено	Страница и данные на ней верно отображаются и не вылезают за рамки элементов	Страница и данные на ней верно отображаются и не вылезают за рамки элементов	Низкий
6	Пользователь авторизован , открыто главное окно	Нажать «История аварийных устройств»	Не предусмотрены	Страница и данные на ней верно отображаются и не вылезают за рамки элементов	Страница и данные на ней верно отображаются и не вылезают за рамки элементов	Низкий

3. СХЕМА ТЕСТИРОВАНИЯ

На рисунке 26 изображена схема тестирования серверной части программного комплекса.

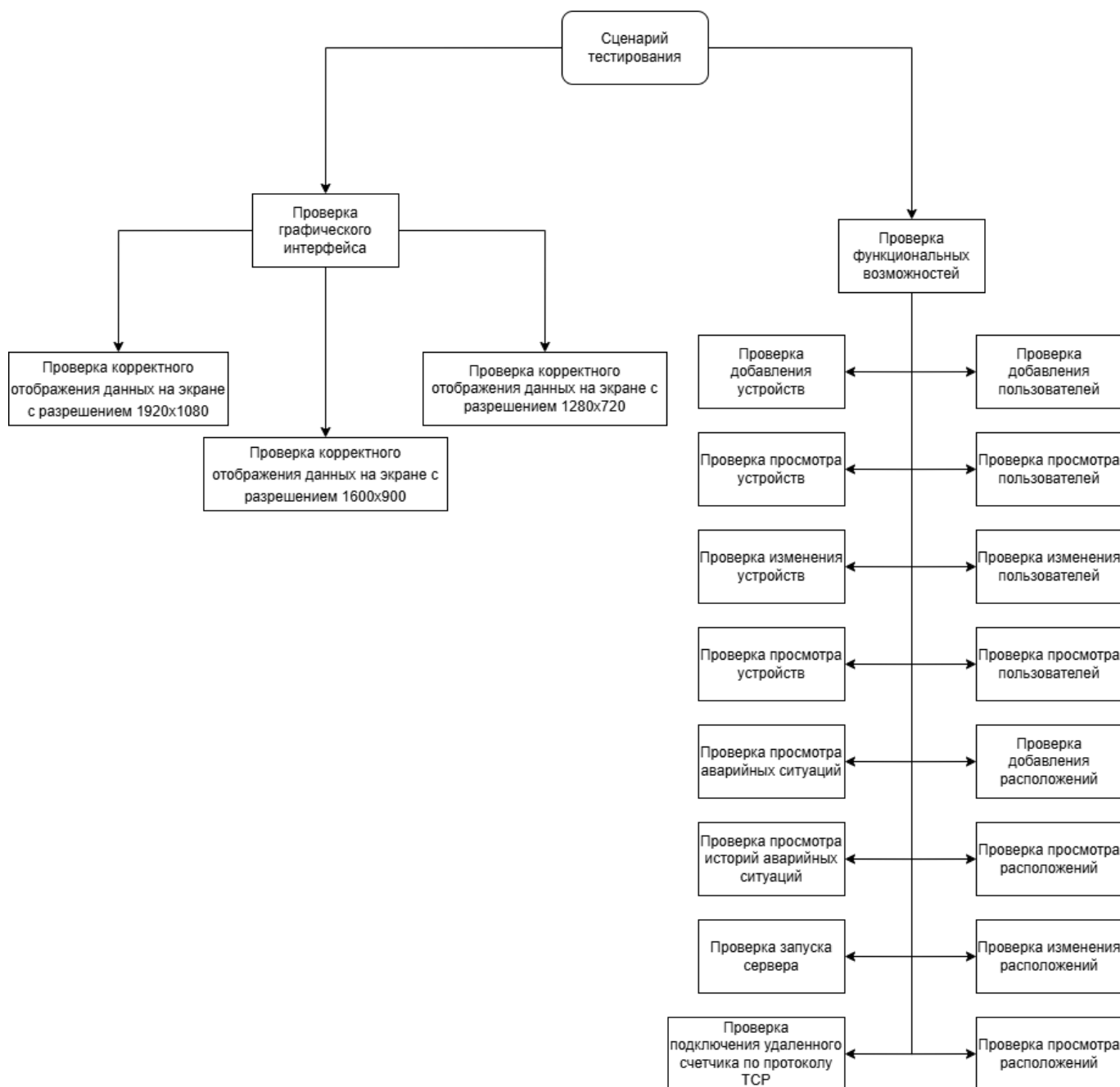


Рисунок 26 - Схема тестирования серверной части

На рисунке 27 изображена схема тестирования клиентской части программного комплекса.

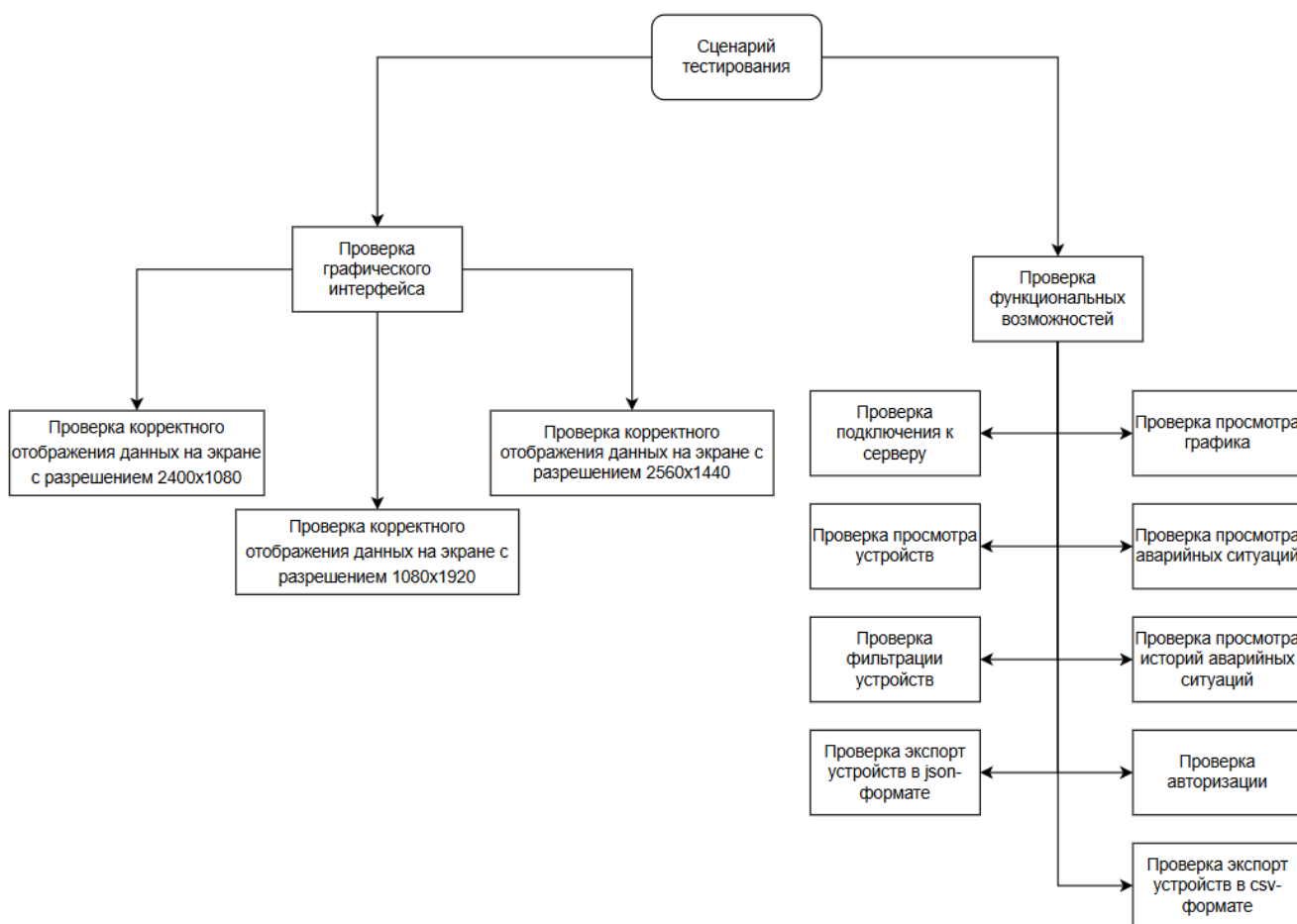


Рисунок 27 – Схема тестирования клиентской части

В таблице 8 описаны проводимые тесты программного комплекса.

Таблица 8 – Тестирование

№ п/п	Предпосылки	Шаги тестирования	Данные тестирования	Ожидаемый результат	Фактический результат	Приоритет
1	2	3	4	5	6	7
Тестирование серверной части						
1		Открыть страницу сервера	Не предусмотрено	Приложение не выдает ошибку, выводится порт запущенного сервера	Приложение не выдает ошибку, выводится порт запущенного сервера	Высокий
2	На введенном адресе и порте запущен сервер	Открыть страницу сервера, ввести корректный адрес и	127.0.0.1, 1235	Вывелось сообщение об успешном подключении и	Вывелось сообщение об успешном подключении и	Высокий

№ п/п	Предпосылки	Шаги тестирования	Данные тестирования	Ожидаемый результат	Фактический результат	Приоритет
		порт, нажать «Подключить счетчик»		дальнейшей асинхронной передачи данных	дальнейшей асинхронной передачи данных	
3		Открыть страницу сервера, нажать «Админ-панель», ввести данные, нажать «Войти»	admin, admin1	Открылось окно сервера, открылась форма авторизации, данные ввелись успешно, открылась админ-панель	Открылось окно сервера, открылась форма авторизации, данные ввелись успешно, открылась админ-панель	Высокий
4	Администратор авторизован, открыта админ-панель	Ввести данные, нажать добавить устройство, изменить данные в таблице и нажать «Сохранить изменения», удалить устройство из таблицы и нажать «Сохранить изменения»	Добавление : 5000, «ШИБП-1-1», «Альфа», 220, 209, 231, 50, 70, 37, 80, Изменение: Название – «ШИБП-1-2», Удаление: первая колонка	Таблица устройств успешно отобразилась, таблица аварийных ситуаций успешно отобразилась с корректным и данными. Данные успешно ввелись, устройство появилось в таблице, данные в таблице изменились, устройство исчезло из таблицы	Таблица устройств успешно отобразилась, таблица аварийных ситуаций успешно отобразилась с корректным и данными. Данные успешно ввелись, устройство появилось в таблице, данные в таблице изменились, устройство исчезло из таблицы	Высокий
5	Администратор авторизован, открыта админ-панель	Нажать «Пользователи», ввести данные, нажать	Добавление : vasek123, Po123_zx Изменение: Логин – «vasek1234	Открылась страница пользователей, таблица пользователей успешно	Открылась страница пользователей, таблица пользователей успешно	Средний

№ п/п	Предпосылки	Шаги тестирования	Данные тестирования	Ожидаемый результат	Фактический результат	Приоритет
		добавить пользователя, изменить данные в таблице и нажать «Сохранить изменения» , удалить пользователя из таблицы и нажать «Сохранить изменения»	5», Удаление: первая колонка	отобразилась, Данные успешно ввелись, пользователь появился в таблице, данные в таблице изменились , пользователь исчез из таблицы	отобразилась, Данные успешно ввелись, пользователь появился в таблице, данные в таблице изменились , пользователь исчез из таблицы	
6	Администратор авторизован , открыта админ-панель	Нажать «Расположения», ввести данные, нажать добавить расположение, изменить данные в таблице и нажать «Сохранить изменения» , удалить расположение из таблицы и нажать «Сохранить изменения»	Добавление : «Зета» Изменение: «Зета-2» Удаление: первая колонка	Открылась страница расположений, таблица расположений успешно отобразилась, Данные успешно ввелись, расположение появилось в таблице, данные в таблице изменились , расположение исчез из таблицы	Открылась страница расположений, таблица расположений успешно отобразилась, Данные успешно ввелись, расположение появилось в таблице, данные в таблице изменились , расположение исчез из таблицы	Средний
7	Администратор авторизован , открыта админ-панель	Нажать «История аварийных устройств»	Не предусмотрено	Открылась страница пользователей, таблица пользователей успешно отобразилась, Данные успешно ввелись, пользователь	Открылась страница пользователей, таблица пользователей успешно отобразилась, Данные успешно ввелись, пользователь	Низкий

№ п/п	Предпосылки	Шаги тестирования	Данные тестирования	Ожидаемый результат	Фактический результат	Приоритет
				ь появился в таблице, данные в таблице изменились , пользователь исчез из таблицы	ь появился в таблице, данные в таблице изменились , пользователь исчез из таблицы	
Тестирование клиентской части						
1	На введенном адресе и порте запущен сервер	Открыть окно ввода IP, ввести данные, нажать «Подключиться»	168.192.0.104, 1234	Приложение не выдает ошибку, открывается страница об авторизации	Приложение не выдает ошибку, открывается страница об авторизации	Высокий
2	Открыта окно авторизации	Ввести логин и пароль в поля, нажать «Войти»	Vasya123, xscxz2231	Появилась круговая загрузка, после завершения которой открылось основное окно отслеживания устройств	Появилась круговая загрузка, после завершения которой открылось основное окно отслеживания устройств	Высокий
3	Пользователь авторизован, открыто главное окно		Не предусмотрено	Открыто окно со списком устройств, который появляется после завершения загрузки, в списке отображаются физические показатели, аварийные устройства выделяются	Открыто окно со списком устройств, который появляется после завершения загрузки, в списке отображаются физические показатели, аварийные устройства выделяются	Высокий

№ п/п	Предпосылки	Шаги тестирования	Данные тестирования	Ожидаемый результат	Фактический результат	Приоритет
				красным фоном	красным фоном	
4	Администратор авторизован, открыто главное окно	Нажать «Работа с данными», ввести фильтр, нажать «Отфильтровать», нажать «Сохранить в виде json», «Сохранить в виде csv»	200, 230, 10, 40, 20, 50 (слева направо)	Открылось окно работы с данными, данные ввелись, после загрузки появились отфильтрованные данные, вывелось сообщение об успешном сохранении, вывелось сообщение об успешном сохранении	Открылось окно работы с данными, данные ввелись, после загрузки появились отфильтрованные данные, вывелось сообщение об успешном сохранении, вывелось сообщение об успешном сохранении	Средний
5	Администратор авторизован, открыто окно работы с данными	Нажать «График»	Не предусмотрено	Открылось окно, на котором, после загрузки, появился график с физическим и показателями устройств	Открылось окно, на котором, после загрузки, появился график с физическим и показателями устройств	Низкий
6	Пользователь авторизован, открыто главное окно	Нажать «История аварийных устройств»	Не предусмотрены	Открылась страница истории аварийных устройств, где, после загрузки, отображаются записи обо всех аварийных ситуациях с	Открылась страница истории аварийных устройств, где, после загрузки, отображаются записи обо всех аварийных ситуациях с	Низкий

№ п/п	Предпосыл ки	Шаги тестирован ия	Данные тестирован ия	Ожидаемый результат	Фактически й результат	Приоритет
				датой и фиксацией физических показателей	датой и фиксацией физических показателей	

ПРИЛОЖЕНИЕ В. РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ

АННОТАЦИЯ

В данном программном документе приведено руководство пользователя на разработку программы «DeviceMonitor».

В данном программном документе, в разделе «Назначение программы» указано функциональное и эксплуатационное назначение программы. Где и кем она будет использоваться.

В разделе «Условия выполнения программы» указаны технические средства и программные средства для использования программы.

В разделе «Выполнение программы» указаны действия для загрузки и эксплуатации программного продукта.

СОДЕРЖАНИЕ

<u>1. Назначение программы</u>	3
<u>2. Условия выполнения программы</u>	4
<u>3. Выполнение программы</u>	5
<u>3.1. Действия для загрузки программы</u>	5
<u>3.2. Выполнение программы с описанием функций</u>	5

1. НАЗНАЧЕНИЕ ПРОГРАММЫ

Функциональное назначение программы "DeviceMonitor" заключается в автоматизации процессов мониторинга и управления электронными устройствами, используемыми в промышленных и производственных процессах. Программа обеспечивает сбор, анализ и визуализацию данных с устройств, предоставляя удобный интерфейс для отслеживания их состояния. Основные функции включают отображение устройств при выходе показателей за допустимые пределы, что позволяет пользователям своевременно принимать меры для оптимальной работы оборудования.

Программа предназначена для повышения удобства и оперативности взаимодействия пользователей с системой, улучшая общий опыт использования и повышая уровень удовлетворённости за счёт централизованного управления и анализа данных. Это особенно важно в условиях развития технологий и увеличения числа электронных устройств, где надежная работа систем критически важна для бесперебойного функционирования предприятий.

2. УСЛОВИЯ ВЫПОЛНЕНИЯ ПРОГРАММЫ

В Таблице 9 представлены максимальные (или рекомендуемые) и минимальные технические средства для использования программы.

Таблица 9- Технические средства

№	Тип оборудования	Наименование оборудования
1	2	3
Максимальные или рекомендуемые		
1	Память на диске:	200 МБ+
2	Процессор:	2 GHz четырёхъядерный или эквивалент
3	Оперативная память	3 GB
Минимальные требования		
1	Память на диске:	100 МБ+
2	Процессор:	1 GHz двухъядерный или эквивалент
3	Оперативная память	512 МБ

В Таблице 10 представлены программные средства для использования программы.

Таблица 10- Программные средства

№	Тип средства	Название средства	Назначение
Программные средства Desktop частей			
1	Операционная система	Astra Linux	Организация взаимодействия программ и пользователя
2	Среда разработки	Qt Creator 4	Среда запуска
3	Сервер базы данных	PostgreSQL	Хранение данных.
4	Графическая оболочка	pgAdmin 3	Разработка базы данных и функционала работы с ней.
Программные средства Android части			
1	Операционная система	Android 9.0+	Организация взаимодействия программ и пользователя
2	Среда разработки	Android Studio	Среда запуска

3. ВЫПОЛНЕНИЕ ПРОГРАММЫ

3.1. Действия для загрузки программы

Для того, чтобы запустить Desktop серверной части программного комплекса «DeviceMonitor» необходимо открыть исполняемый файл программы, предварительно установив драйвер Qt для PostgreSQL под названием libqt5sql5-psql, а также развернуть базу данных из файла. Для запуска Desktop клиентской части необходимо открыть исполняемый файл программы. Для запуска Android-части необходимо запустить установочный файл (.apk), после чего открыть установленную программу.

3.2. Выполнение программы с описанием функций

Начинается серверная часть программного комплекса с запуска сервера и открытия одноименного окна. Здесь можно увидеть информацию о порте сервера, количество подключенных устройств, отображение сетевых логов, поля для ввода адреса и порта удаленного счетчика для дальнейшего подключения и кнопка открытия админ-панели.

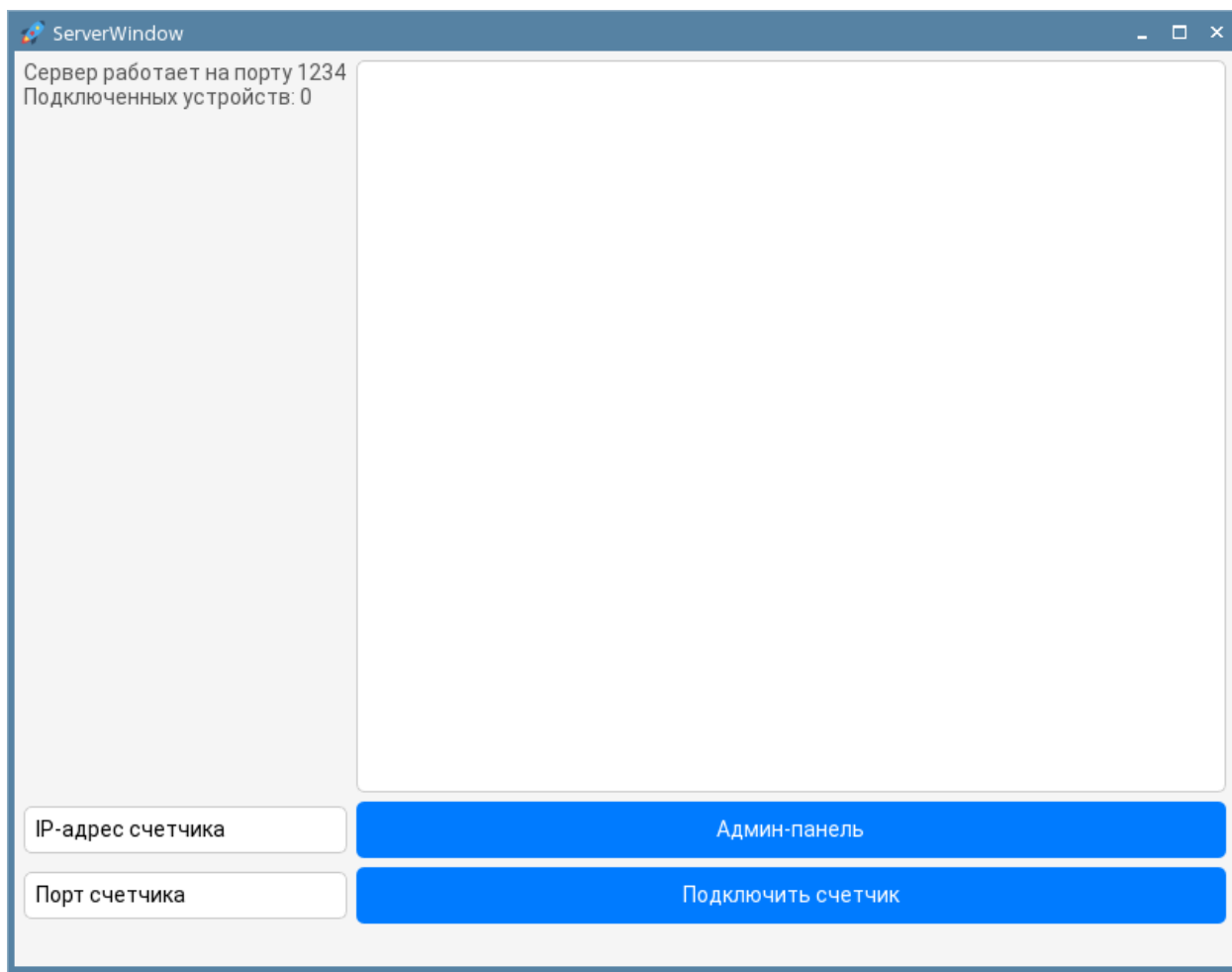


Рисунок 28 – Окно сервера

При открытии админ-панели откроется окно авторизации, где можно ввести данные пользователя и авторизоваться.

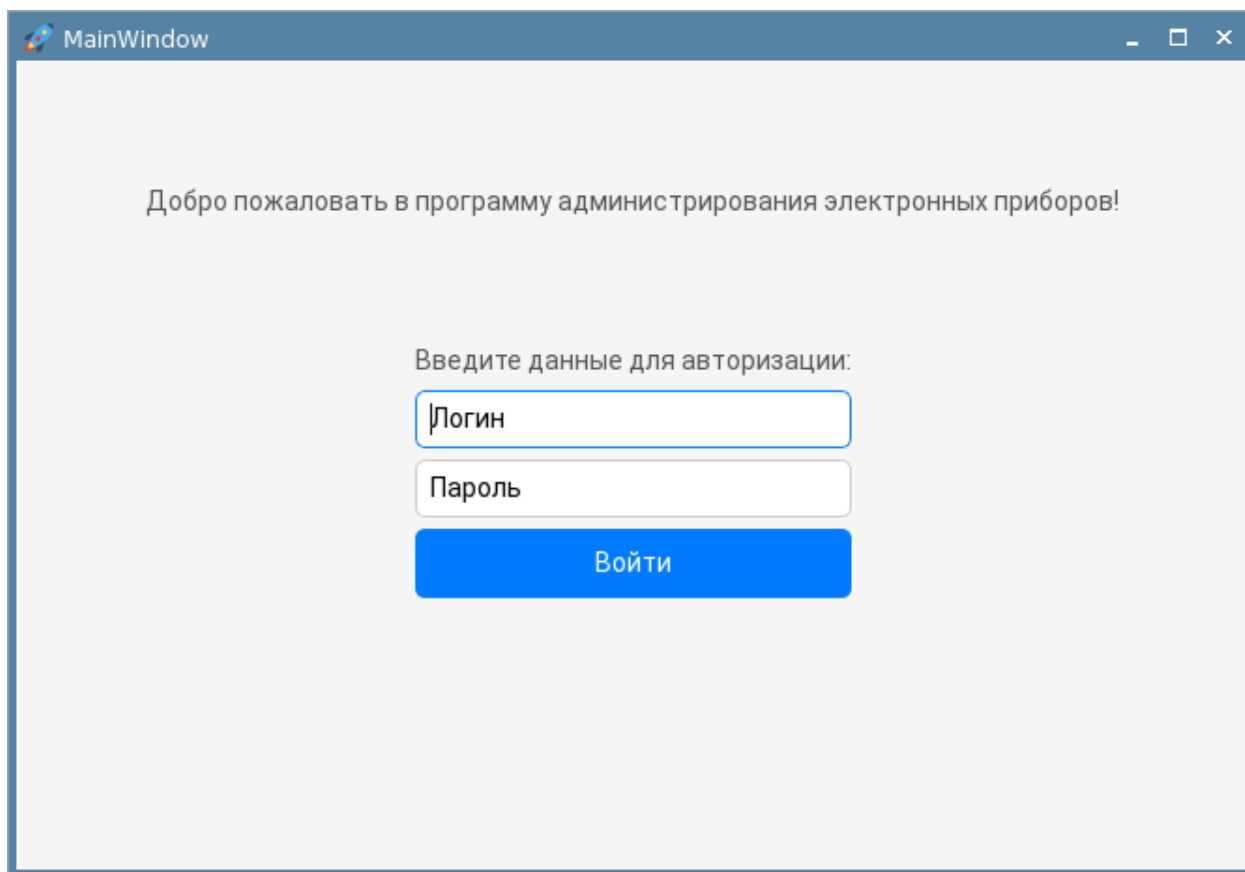


Рисунок 29 – Страница регистрации

Авторизовавшись, в случае наличия роли «Администратор» от роли пользователь попадает на основную страницу админ-панели, где отображаются устройства, аварийные устройства, можно перейти на страницу пользователей, страницу расположений, страницу историй аварийных случаев, а также можно добавить устройство в БД или изменить его.

AdminWindow

Панель администратора

Все устройства

Аварийные устройства

	id	name	turned	voltage	voltage_max	voltage_min	amperage	impera
1	399	ШП-Р	false	260	231	209	25	40
2	402	ШП	false	229	231	209	42	40
3	404	ЩМ-120	false	223	231	209	20	40
4	405	ШИБП 1-2	false	244	231	209	44	40

	id	name	turned	voltage	voltage_max	voltage_min	amperage	impera
1	397	ШИБП-Р	true	211	231	209	40	40
2	398	ЩМ-120	false	224	231	209	3	40
3	399	ШП-Р	false	260	231	209	25	40
4	400	ШИБП 1-1	false	213	231	209	34	40

Расположения

Пользователи

История аварийных случаев

Сохранить изменения

Емкость

Напряжение

Температура

Макс. Температура

Название

Мин. Напряжение

Ток

☐ Включено ли устройство

▼

Макс. Напряжен...

Макс. Ток

Добавить устройство

Рисунок 30 – Главная страница администратора

При нажатии на кнопку «Расположения» открывается одноименное окно, которое содержит в себе список расположений устройств в БД и дает возможность его изменять, пополнять или удалять расположения из БД.

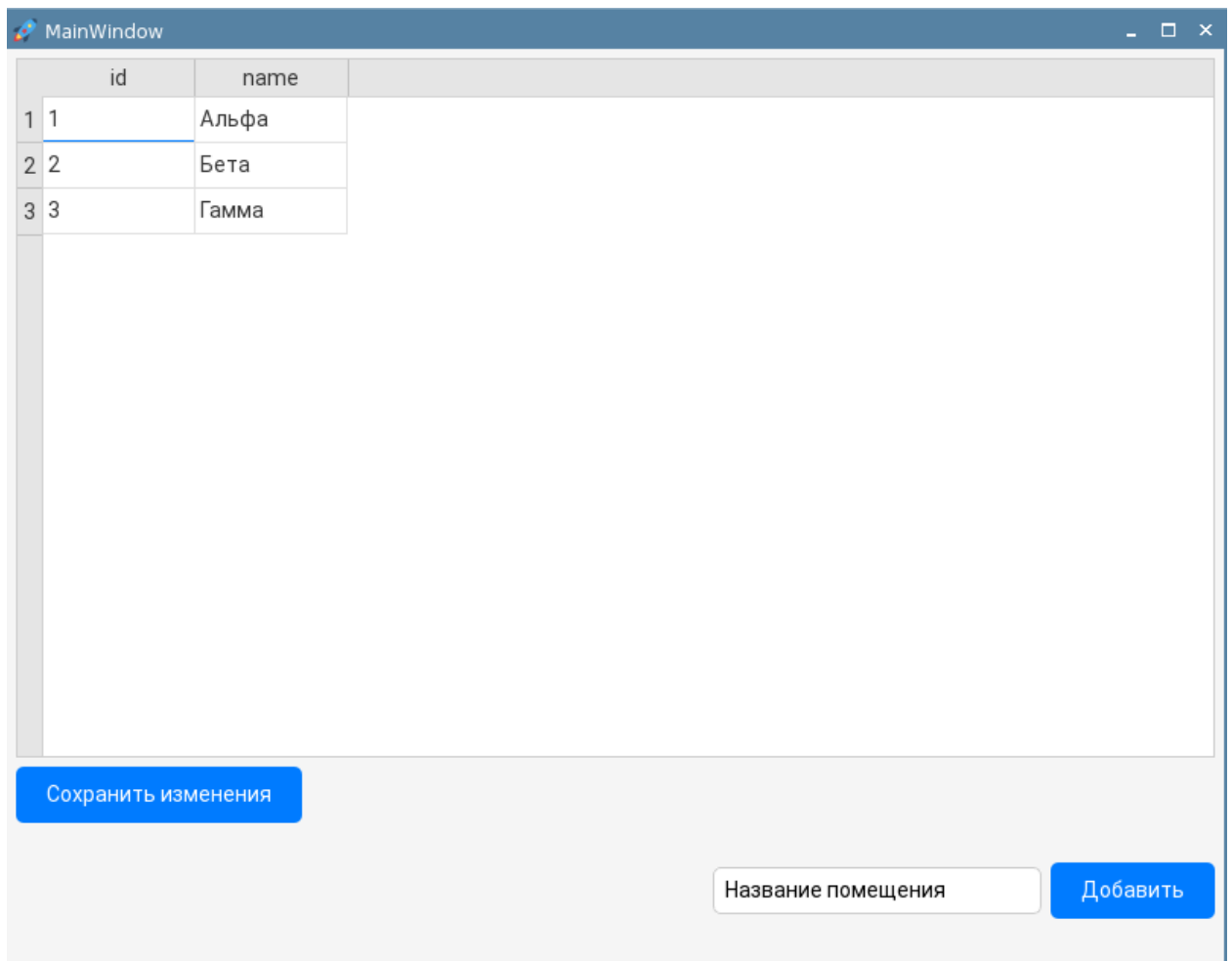


Рисунок 31 – Страница расположений

При нажатии на кнопку «Пользователи» открывается одноименное окно, которое содержит в себе список пользователей в БД и дает возможность его изменять (за исключением пароля), пополнять или удалять пользователей из БД.

	id	username	creation_time	role_id
1	1	admin	07.11.2024 ...	1
2	2	vasya123	07.11.2024 ...	2
3	3	asdf	11.11.2024 ...	2

Сохранить изменения

Логин ☐ Администратор

Пароль

Рисунок 32 – Страница пользователей

Нажав на админ-панели кнопку «История аварийных случаев» откроется одноименное окно, содержащее в себе все записи об аварийных ситуациях устройств с дублем их тогдашних показателей.

	id	name	turned	voltage	voltage_max	voltage_min	amperage	imp
1	1	ШИБП 1-1	true	225	231	209	36	40
2	2	ШП	false	226	231	209	36	40
3	3	ШИБП-Р	true	231	231	209	22	40
4	4	ШИБП 1-3	false	122	231	209	28	40
5	5	ШИБП 1-1	false	213	231	209	43	40
6	6	ШИБП 1-3	false	217	231	209	53	40
7	7	ШИБП-Р	false	175	231	209	47	40

Рисунок 33 – Страница историй аварийных случаев

Клиентская часть по функционалу идентична между Android и Desktop версиями, потому что будет описана вместе. Начинается программа с ввода IP адреса и порта подключения к серверу на рисунках 34-35.

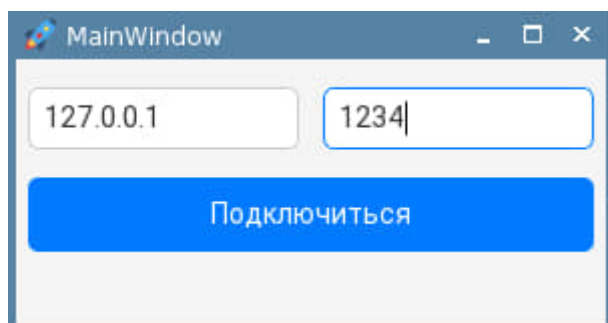


Рисунок 34 – Страница подключения по IP Desktop версии



Рисунок 35 - Страница подключения по IP Android версии

После подключения к серверу открывается страница авторизации, где пользователь вводит свой логин и пароль для входа в программу на рисунках 36-37.

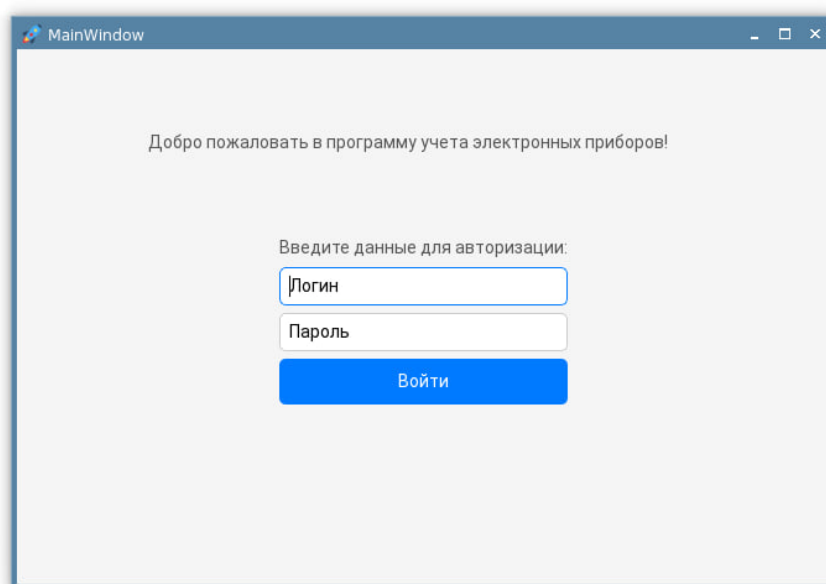


Рисунок 36 - Страница авторизации Desktop версии

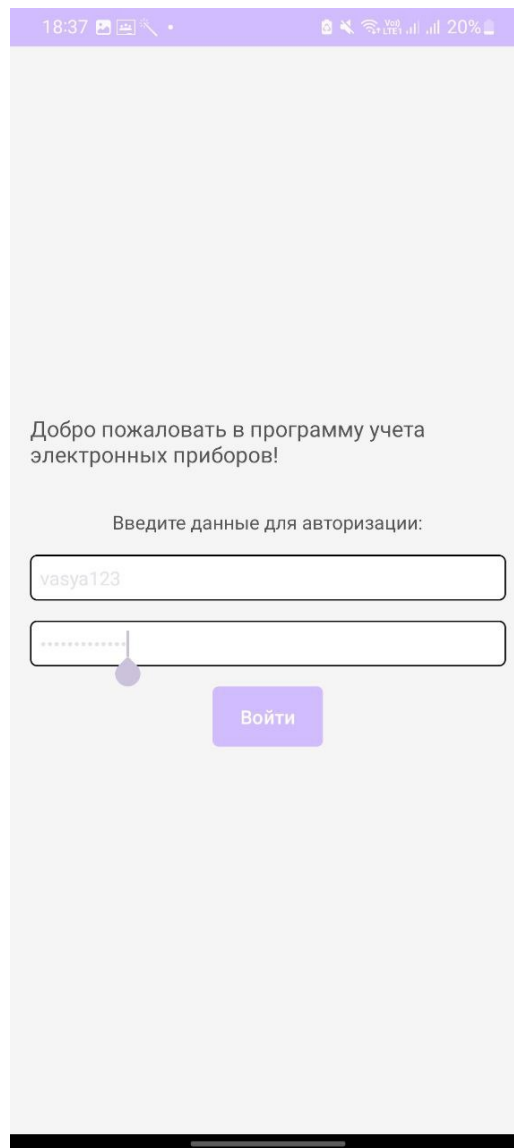


Рисунок 37 – Страница авторизации Android версии

После успешной авторизации, открывается страница отображения устройств на рисунках 38-39, откуда можно перейти на страницу истории аварийных случаев и на страницу работы с данными.

UserWindow

Добро пожаловать!

	№	Название	Включен	Напряжение	ин. Напряжение	исс. Напряжен	Ток	Макс.
1	408	ШИБП 1-3	false	209	209	231	13	40
2	409	ШИБП 1-3	false	230	209	231	10	40
3	410	ШИБП 1-3	false	216	209	231	60	40
4	411	ШИБП-Р	true	230	209	231	23	40
5	412	ШИБП 1-1	true	212	209	231	59	40
6	413	ЦМ-120	false	253	209	231	8	40
7	414	ШИБП 1-2	false	212	209	231	20	40
8	415	ШИБП 1-3	true	211	209	231	27	40
9	416	ШИБП 1-3	true	212	209	231	20	40
10	417	ШП-Р	false	213	209	231	27	40

История аварийных случаев

Работа с данными

Рисунок 38 – Страница устройств Desktop версии

18:37

Главная страница

ШИБП 1-3		
ID	Включено	Напряжение
1	false	231
Мин. напряжение	Макс. напряжение	Сила тока
209	231	30
Макс. сила тока	Температура	Макс. температура
40	67	60
Емкость	ID размещения	
36	1	

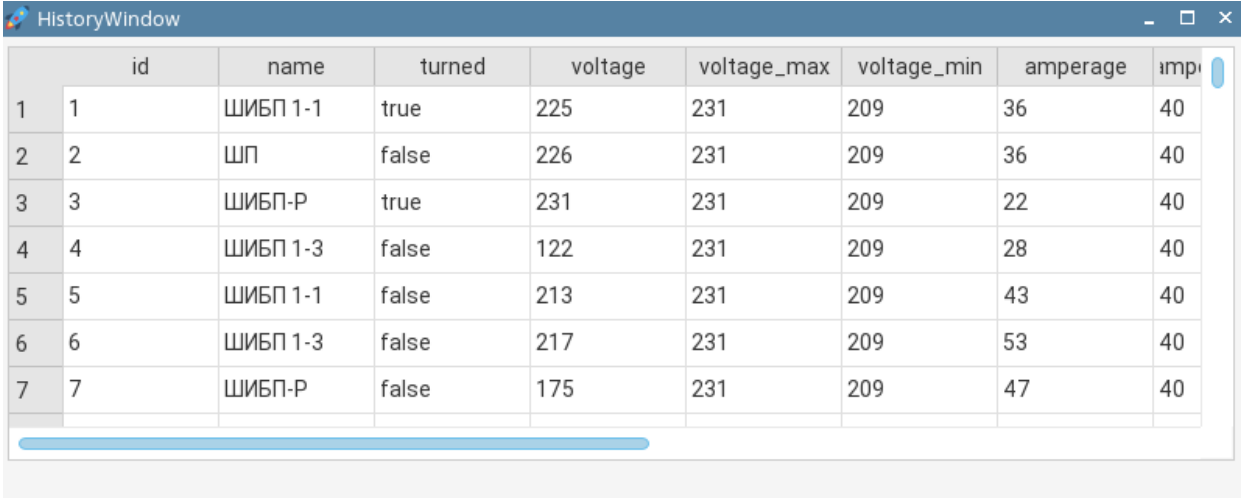
ШИБП-Р		
ID	Включено	Напряжение
2	false	250
Мин. напряжение	Макс. напряжение	Сила тока
209	231	23
Макс. сила тока	Температура	Макс. температура

История аварийных случаев

Работа с данными

Рисунок 39 – Страница устройств Android версии

При нажатии на кнопку «История аварийных случаев» откроется одноименная страница на рисунках 40-41, на которой отображаются себе все записи об аварийных ситуациях устройств с дублем их тогдашних показателей.



The screenshot shows a window titled 'HistoryWindow' containing a table with 9 columns: an index column, 'id', 'name', 'turned', 'voltage', 'voltage_max', 'voltage_min', 'amperage', and 'imp'. The table contains 7 rows of data. Below the table is a horizontal scrollbar.

	id	name	turned	voltage	voltage_max	voltage_min	amperage	imp
1	1	ШИБП 1-1	true	225	231	209	36	40
2	2	ШП	false	226	231	209	36	40
3	3	ШИБП-Р	true	231	231	209	22	40
4	4	ШИБП 1-3	false	122	231	209	28	40
5	5	ШИБП 1-1	false	213	231	209	43	40
6	6	ШИБП 1-3	false	217	231	209	53	40
7	7	ШИБП-Р	false	175	231	209	47	40

Рисунок 40 – Страница истории аварийных случаев Desktop версии

ШИБП 1-1		
Время создания	ID	Включено
Вт мая 13 18:37:52 2025	1	false
Напряжение	Мин. напряжение	Макс. напряжение
25	209	231
Сила тока	Макс. сила тока	Температура
60	40	21
Макс. температура	Емкость	ID размещения
60	78	2
ЩМ-120		
Время создания	ID	Включено
Вт мая 13 18:37:52 2025	2	false
Напряжение	Мин. напряжение	Макс. напряжение
234	209	231
Сила тока	Макс. сила тока	Температура
32	40	80

Рисунок 41 – Страница истории аварийных случаев Android версии

Если на странице вывода устройств нажать на кнопку «Работа с данными», то откроется соответствующая страница на рисунках 42-43, на которой можно фильтровать данные, импортировать данные в формате csv и json по нажатию соответствующих файлов, а также открыть страницу графика.

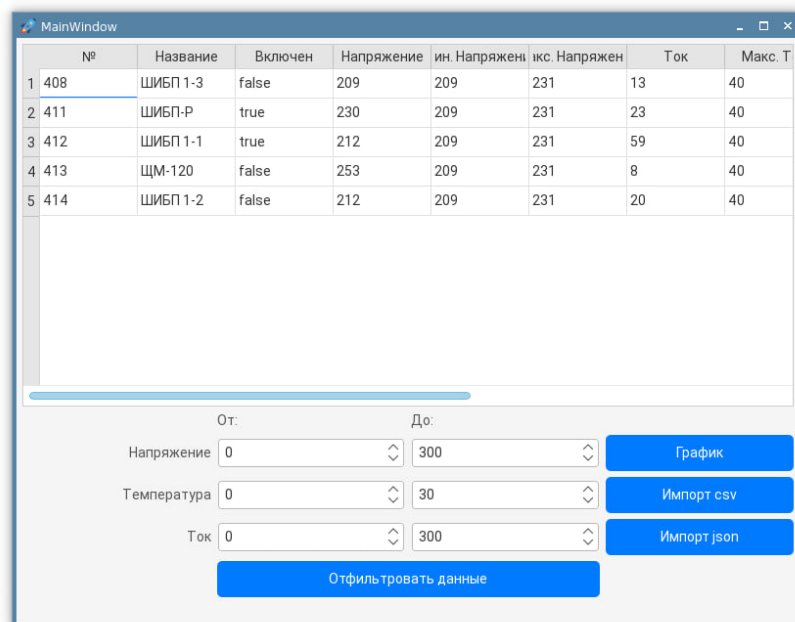


Рисунок 42 – Страница работы с данными Desktop версии



Рисунок 43 – Страница работы с данными Android версии

При нажатии кнопки «График» откроется страница графика, где динамически отображаются физические показатели устройств на рисунках 44-45.

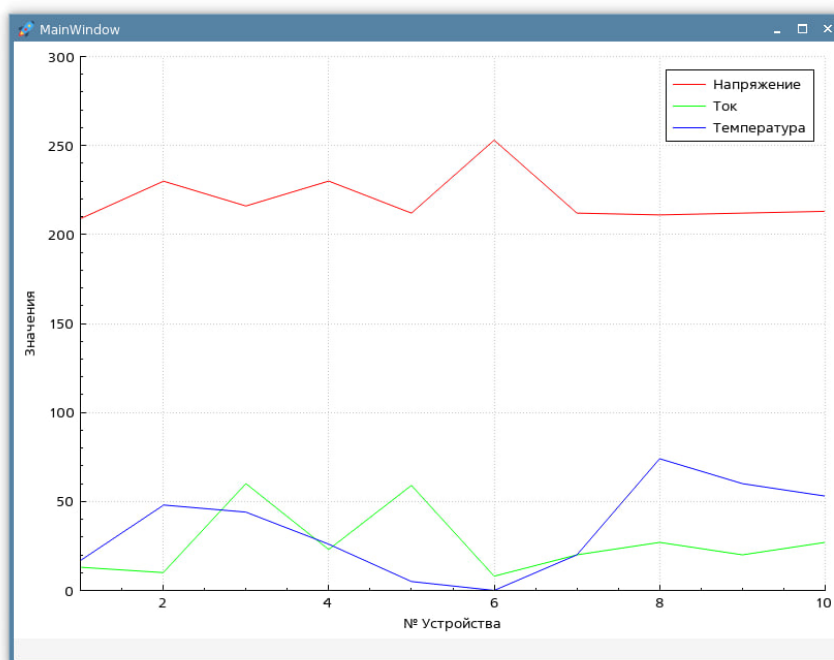


Рисунок 44 – Страница графика Desktop версии

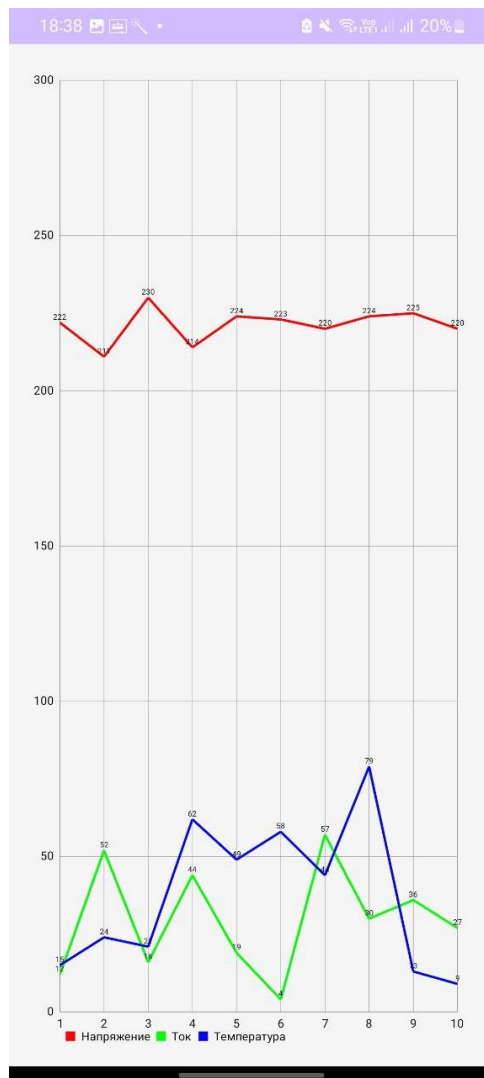


Рисунок 45 – Страница графика Android версии

ПРИЛОЖЕНИЕ Г. СКРИПТ БАЗЫ ДАННЫХ

АННОТАЦИЯ

В данном программном документе представлен скрипт базы данных, структурированный следующим образом: наименование программы, область применения, скрипт базы данных.

1.1. Наименование программы: Документ содержит информацию о наименовании программы, что является ключевым идентификатором проекта.

1.2. Область применения программы: Описана область, в которой предполагается использование программы. Это важно для понимания контекста её применения.

1.3. Скрипт базы данных: Представлен скрипт базы данных, написанный на языке PostgreSQL для хранения данных приложения.

Данный документ служит ориентиром для разработчиков, предоставляя необходимую информацию о программе, её структуре и функциональности.

СОДЕРЖАНИЕ

<u>1. Скрипт базы данных</u>	3
<u>1.1. Наименование программы</u>	3
<u>1.2. Область применения программы</u>	3
<u>1.3. Скрипт базы данных</u>	3
<u>1.4. Словарь данных</u>	5

1. СКРИПТ БАЗЫ ДАННЫХ

1.1. Наименование программы

Наименование программы «DeviceMonitor».

1.2. Область применения программы

Приложение «DeviceMonitor» предоставляет функционал для контроля аварийных ситуаций устройств. Приложение включает в себя удобный интерфейс для добавления и отслеживания устройств, возможность подключать счетчик по сетевому протоколу, а также запускать сервер и подключаться к нему.

1.3. Скрипт базы данных

--Create-блок

--Таблица ролей пользователей

```
create table roles(  
    id serial primary key,  
    name varchar(50) not null --Название роли  
);
```

--Таблица аутентификационных данных пользователей

```
create table users(  
    id serial primary key,  
    username varchar(50) not null,  
    password varchar(50) not null,  
    role_id int references roles(id) on delete cascade --Роль пользователя (Администратор, Пользователь)  
);
```

--Таблица расположений устройств

```
create table placements(  
    id serial primary key,  
    name varchar(50) not null --Название расположения  
);
```

--Таблица устройств с физическими значениями

```
create table devices(  
    id serial primary key,  
    name varchar(70) not null,  
    turned boolean default true, --Включен ли прибор  
    voltage int not null, --Напряжение (с минимальным и максимальным значением)  
    voltage_max int not null,  
    voltage_min int not null,  
    amperage int not null, --Ток (с максимальным значением)  
    amperage_max int not null,  
    temperature int not null, --Температура (с максимальным значением)  
    temperature_max int not null,  
    capacity int not null, --Емкость  
    placement_id int references placements(id)  
);
```


--Таблица истории аварийных случаев (Если в БД появляется устройство с хотя бы одним значением, ниже минимального или выше максимального, то в таблицу добавляется запись со ссылкой на устройство и датой аварии)

```
create table device_emergency_history(  
    id serial primary key,  
    name varchar(70) not null,  
    turned boolean default true, --Включен ли прибор  
    voltage int not null, --Напряжение (с минимальным и максимальным значением)  
    voltage_max int not null,  
    voltage_min int not null,  
    amperage int not null, --Ток (с максимальным значением)  
    amperage_max int not null,  
    temperature int not null, --Температура (с максимальным значением)  
    temperature_max int not null,  
    capacity int not null, --Емкость  
    placement_id int references placements(id),  
    creation_time timestamp default CURRENT_TIMESTAMP --Автоматическая запись текущей даты и  
времени  
);  
--Function-блок
```

--Функция проверки значений устройства перед добавлением или изменением, и добавление в таблицу истории аварий записи при неудовлетворении условий минимальных или максимальных значений

```
create or replace function check_value_trigger()  
returns trigger as $$  
begin  
    if NEW.voltage > NEW.voltage_max or NEW.voltage < NEW.voltage_min or NEW.amperage >  
NEW.amperage_max or NEW.temperature > NEW.temperature_max then  
        insert into device_emergency_history (device_id) values (NEW.id);  
    end if;  
    return new;  
end;  
$$ LANGUAGE plpgsql;
```

--Trigger-блок

```
create trigger check_value_trigger after insert or update on devices for each row execute function  
check_value_trigger();
```

```
drop trigger check_value_trigger on devices
```

--Insert-блок

```
insert into roles (name) values ('Администратор');  
insert into roles (name) values ('Пользователь');
```

```
insert into users (username, password, role_id) values ('admin', 'admin1', 1);  
insert into users (username, password, role_id) values ('vasya123', '123', 2);
```

```
insert into placements(name) values ('Альфа');  
insert into placements(name) values ('Бета');
```

```
insert into devices(name, turned, voltage, voltage_max, voltage_min, amperage, amperage_max, temperature,  
temperature_max, capacity, placement_id) values ('ШИБП 1-1', true, 253, 231, 209, 36, 40, 52, 60, 150, 1);  
insert into devices(name, turned, voltage, voltage_max, voltage_min, amperage, amperage_max, temperature,  
temperature_max, capacity, placement_id) values ('ШИБП 1-2', true, 220, 231, 209, 32, 40, 44, 60, 150, 2);  
insert into devices(name, turned, voltage, voltage_max, voltage_min, amperage, amperage_max, temperature,  
temperature_max, capacity, placement_id) values ('ЦМ-120', true, 193, 231, 209, 44, 40, 51, 60, 210, 1);
```

--Select-блок

```
select * from roles;  
select * from users;  
select * from devices;  
select * from placements;
```

```
select * from device_emergency_history;
```

```
--Вывод всех устройств, находящиеся в текущий момент в аварийной ситуации
select 'Аварийная ситуация' as Статус, * from devices d join placements p on d.placement_id = p.id where
d.voltage > d.voltage_max or d.voltage < d.voltage_min or d.amperage > d.amperage_max or d.temperature >
d.temperature_max;
```

```
--Сору-блок
```

```
--Сохранение данных всех таблиц в отдельный файл формата csv
copy roles to '/tmp/roles.csv' with (format csv, header);
copy users to '/tmp/users.csv' with (format csv, header);
copy devices to '/tmp/devices.csv' with (format csv, header);
copy placements to '/tmp/placements.csv' with (format csv, header);
copy device_emergency_history to '/tmp/device_emergency_history.csv' with (format csv, header);
```

```
--Сохранение данных о всех аварийных устройствах в отдельный файл формата csv
copy (select 'Аварийная ситуация' as Статус, * from devices d join placements p on d.placement_id = p.id where
d.voltage > d.voltage_max or d.voltage < d.voltage_min or d.amperage > d.amperage_max or d.temperature >
d.temperature_max) to '/tmp/danger_devices.csv' with (format csv, header);
```

```
--Drop-блок
drop table devices_placements;
drop table devices;
drop table device_emergency_history;
drop table placements;
drop table roles;
```

1.4. Словарь данных

В Таблице 11 представлен словарь данных для разработанной базы данных, с которой взаимодействует программный комплекс «DeviceMonitor».

Таблица 11 – Словарь данных

№	Кл юч	Наименование	Тип данных	Обязательность заполнения	Назначение
1	2	3	4	5	6
Таблица - roles					
1	PK	id	serial	Not null	Уникальный идентификатор
2		name	varchar(70)	Not null	Наименование
Таблица - users					
3	PK	id	serial	Not null	Уникальный идентификатор
4		username	varchar(50)	Not null	Логин
5		password	varchar(50)	Not null	Пароль
6	FK	role_id	int	Not null	Ссылка на роль
Таблица - placements					
7	PK	id	serial	Not null	Уникальный идентификатор
8		name	varchar(50)	Not null	Наименование
Таблица - devices					

№	Кл юч	Наименование	Тип данных	Обязательность заполнения	Назначение
1	2	3	4	5	6
9	PK	id	serial	Not null	Уникальный идентификатор
1 0		name	varchar(70)	Not null	Наименование
1 1		voltage	int	Not null	Напряжение
1 2		voltage_min	int	Not null	Минимальное напряжение
1 3		voltage_max	int	Not null	Максимальное напряжение
1 4		amperage	int	Not null	Ток
1 5		amperage_max	int	Not null	Максимальный ток
1 6		temperature	int	Not null	Температура
1 7		temperature_m ax	int	Not null	Максимальная температура
1 8		capacity	int	Not null	Емкость
1 9	FK	placement_id	int	Not null	Ссылка на расположение
2 0		turned	boolean	Not null	Включено ли устройство
Таблица - device_emergency_history					
2 1	PK	id	serial	Not null	Уникальный идентификатор
2 2		name	varchar(70)	Not null	Наименование
2 3		voltage	int	Not null	Напряжение
2 4		voltage_min	int	Not null	Минимальное напряжение
2 5		voltage_max	int	Not null	Максимальное напряжение
2 6		amperage	int	Not null	Ток
2 7		amperage_max	int	Not null	Максимальный ток
2 8		temperature	int	Not null	Температура
2 9		temperature_m ax	int	Not null	Максимальная температура
3 0		capacity	int	Not null	Емкость
3 1		placement_id	int	Not null	Ссылка на расположение

№	Кл юч	Наименование	Тип данных	Обязательность заполнения	Назначение
1	2	3	4	5	6
3 2		turned	boolean	Not null	Включено ли устройство
3 3		creation_time	timestamp	default - CURRENT_TIMEST AMP	Время аварии

ПРИЛОЖЕНИЕ Д. БИЗНЕС ПРОЦЕССЫ АННОТАЦИЯ

В данном программном документе представлены бизнес-процессы, структурированный следующим образом: модель AS-IS, модель TO-BE.

Модель AS-IS: содержит модель IDEF0 AS-IS, до введения программного комплекса, бизнес-процесса и его описание, а также диаграмму прецедентов до внедрения программного комплекса;

Модель TO-BE: содержит модель IDEF0 TO-BE, после введения программного комплекса, бизнес-процесса и его описание, а также диаграмму прецедентов после внедрения программного комплекса;

СОДЕРЖАНИЕ

<u>1. БИЗНЕС ПРОЦЕССЫ</u>	3
<u>1.1. Модель AS-IS</u>	3
<u>1.2. Диаграмма прецедентов после внедрения ИС и модель TO-BE</u>	7

1. БИЗНЕС ПРОЦЕССЫ

1.1. Модель AS-IS

На рисунке 46 указан бизнес-процесс «Отслеживания устройств» модели А-0.

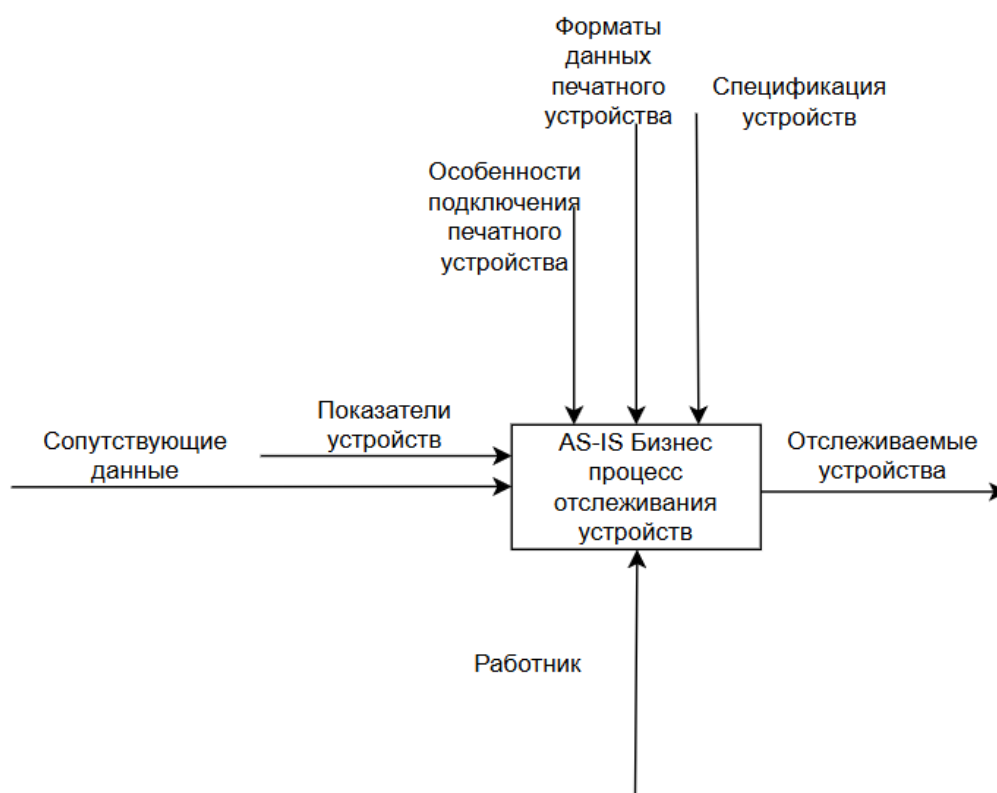


Рисунок 46 – Отслеживание устройств

У представленной модели А0 содержатся этапы от добавления устройства до подключения счетчика.

В модель передаются данные:

- Показатели устройств: физические показатели устройств, считываемые каким-либо образом, для дальнейшего внесения в БД.
- Сопутствующие данные: расположение устройства и наличие аварийного состояния у него.чѐ

Данные на выход модели:

- Отслеживаемые устройства

Механизмы модели: работник (заполняет записи).

На рисунке 47 продемонстрирована модель A1 этапа добавления устройства.

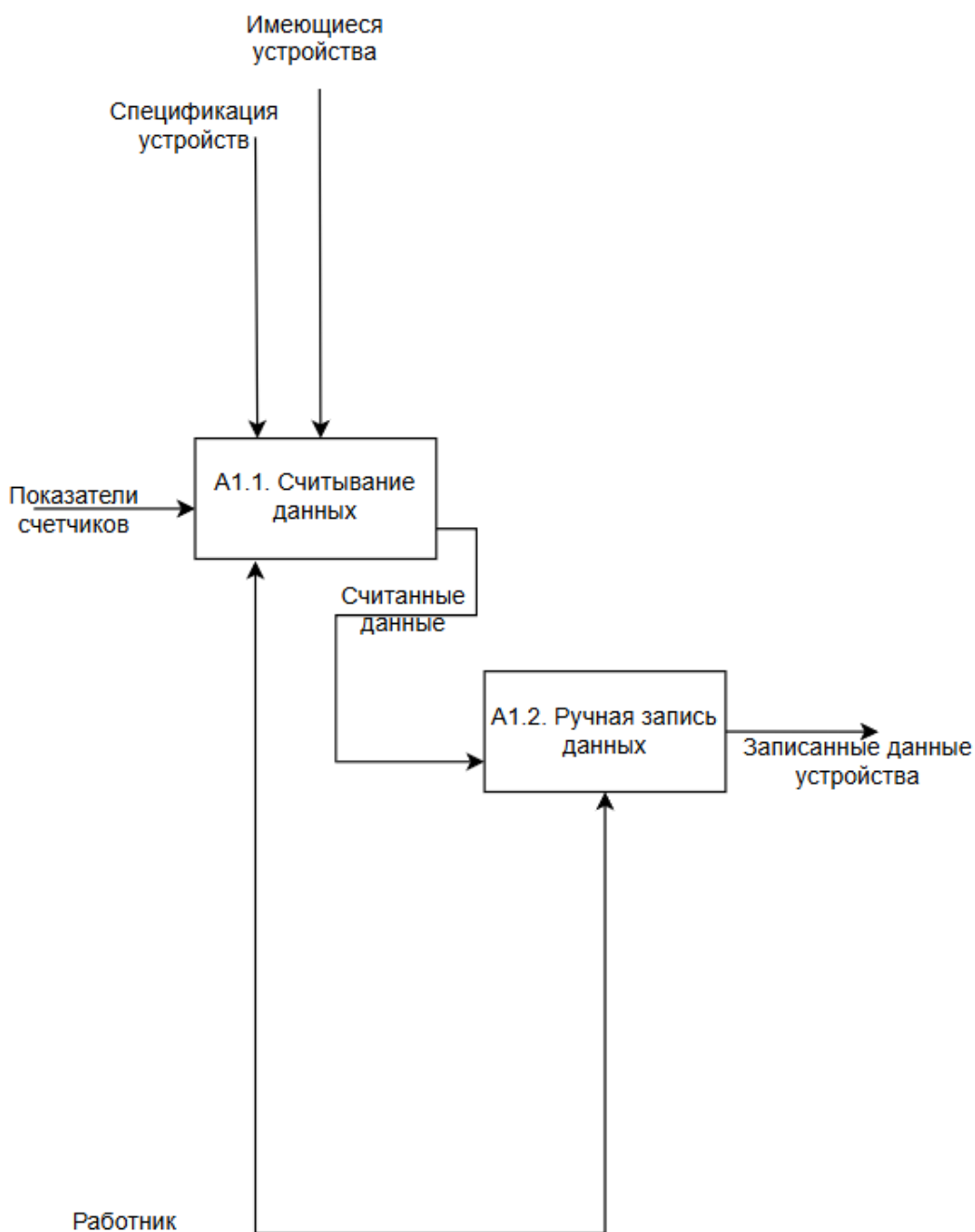


Рисунок 47 – Этап A1 Добавление устройства

На рисунке 48 продемонстрирована модель A2 этапа изменения устройства.

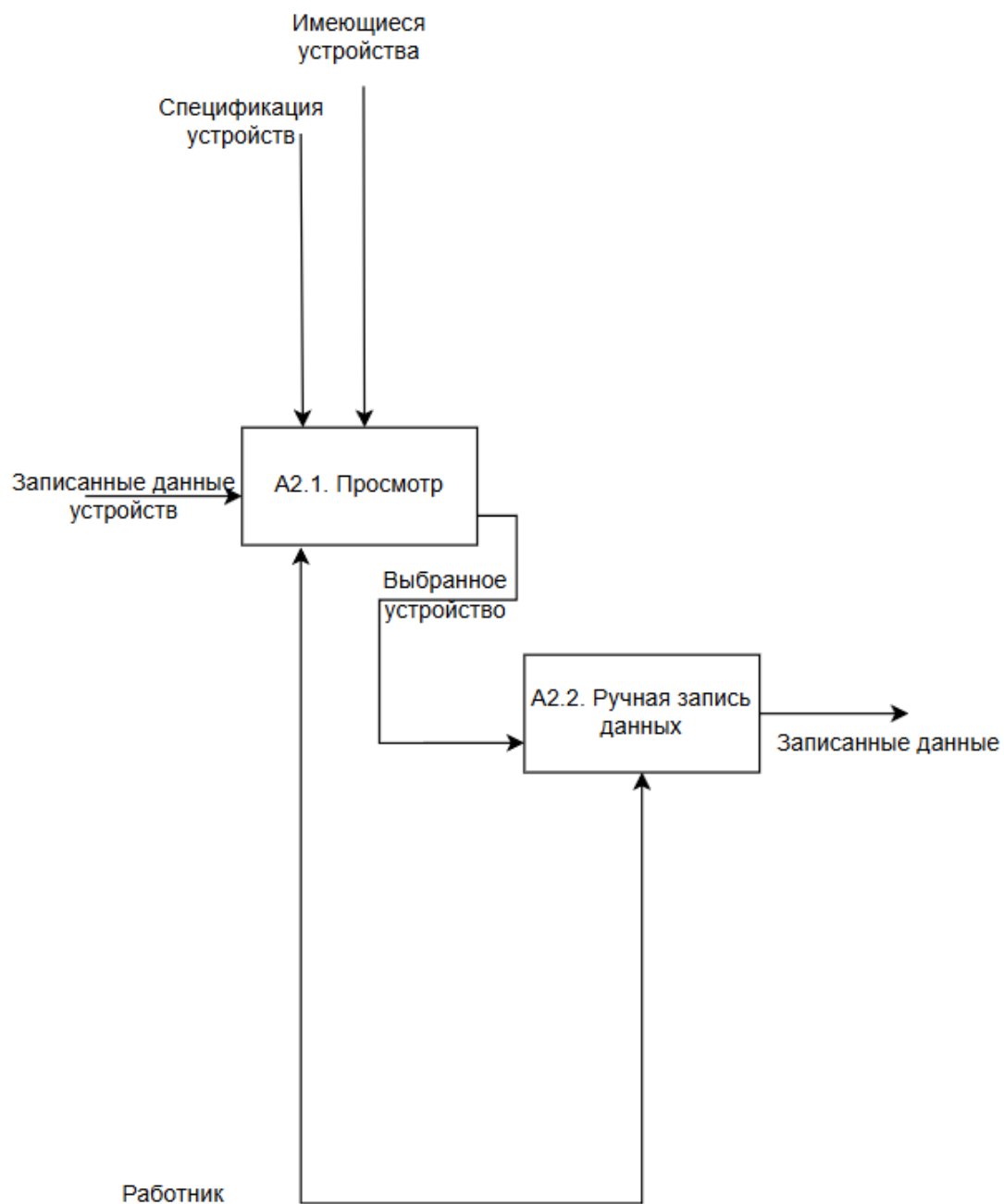


Рисунок 48 – Этап А2 Изменение устройства

На рисунке 49 продемонстрирована модель А3 этапа подключения счетчика.

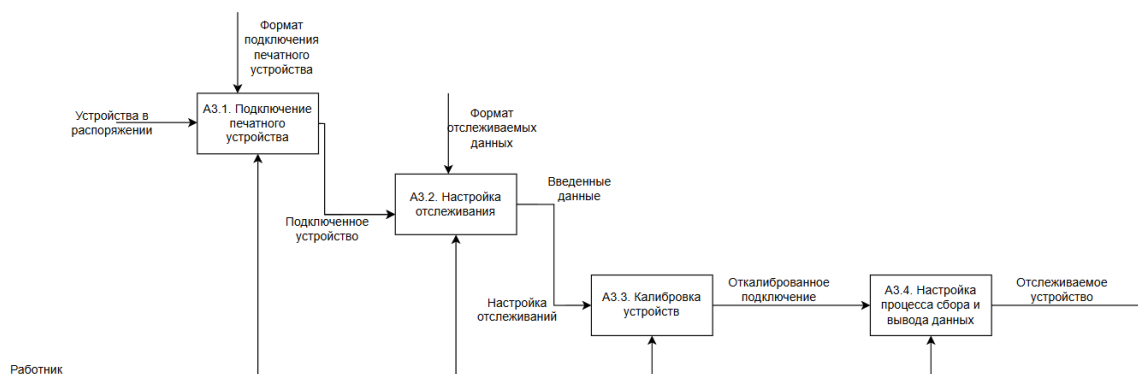


Рисунок 49 – Этап А3 Подключение счетчика

Проблемы бизнес-процесса до внедрения

1) Ручное ведение записей:

- Использование бумажных носителей для записи данных устарело и неэффективно. Это может привести к ошибкам, потере данных и задержкам в обработке информации.

2) Отсутствие автоматизации:

- Процесс не автоматизирован, что увеличивает время на выполнение задач и снижает общую производительность. Современные системы автоматизации могли бы значительно ускорить процесс и уменьшить количество ошибок.

3) Недостаток интеграции:

- Отсутствие интеграции между различными этапами процесса может привести к дублированию данных и необходимости многократного ввода одной и той же информации, что увеличивает риск ошибок.

Критические ошибки алгоритма:

1) Отсутствие централизованного хранения данных:

- Данные хранятся на бумажных носителях, что затрудняет их поиск и использование. Централизованная база данных могла бы обеспечить быстрый доступ к информации и её защиту.

2) Высокий риск потери данных:

- Бумажные записи могут быть легко утеряны или повреждены, что приведет к потере важной информации. Электронное хранение данных с резервным копированием могло бы минимизировать этот риск.

3) Невозможность быстрого анализа данных:

- Вручную записанные данные сложно и долго анализировать. Современные программные решения позволяют быстро обрабатывать и анализировать данные, что способствует принятию обоснованных решений.

Решение проблем данного бизнес-процесса:

1) Автоматизация сбора данных:

- Внедрение электронных форм для сбора данных вместо бумажных носителей. Это позволит автоматически собирать и обрабатывать данные, уменьшая вероятность ошибок и ускоряя процесс.

2) Централизованное хранение данных:

- Использование базы данных для хранения всех собранных данных. Это обеспечит быстрый доступ к информации, её защиту и возможность анализа.

3) Интеграция процессов:

- Интеграция различных этапов процесса через единую информационную систему. Это позволит избежать дублирования данных и улучшить взаимодействие между различными участниками процесса.

1.2. Диаграмма прецедентов после внедрения ИС и модель TO-BE

Модель AS-IS строится на основе записей показателей на бумажный носитель, а модель TO-BE на основе добавление устройств в приложение.

Эти изменения приведут к значительному сокращению времени на поиск и отсеивание на профессиональных фотографов, а также отсеивание по навыкам и месту работы.

Проект представляет из себя программный комплекс, состоящий из двух Desktop-приложений на фреймворке QT C++ и БД PostgreSQL и одного мобильного приложения на языке Java для ОС Android.

В приложении должно быть две основные роли: пользователь и администратор.

У пользователя и администратора две разных части приложения, серверная и клиентская. Серверная часть это Desktop-приложение, запускающее сервер и предоставляющая возможность пользоваться админ-панелью администратору после авторизации с соответствующей ролью. Клиентская часть – это два приложения с одним и тем же функционалом, одно из которых Desktop, другое – для Android.

У админа есть отдельная админ-панель, с необходимыми функциями, такими как CRUD-операции расположений, устройств и пользователей.

На рисунке 50 приведена контекстная диаграмма для модели бизнес-процессов деятельности приложения по размещению и отслеживанию. Контекстная диаграмма представляет собой самое общее описание системы и ее взаимодействие внутри себя.

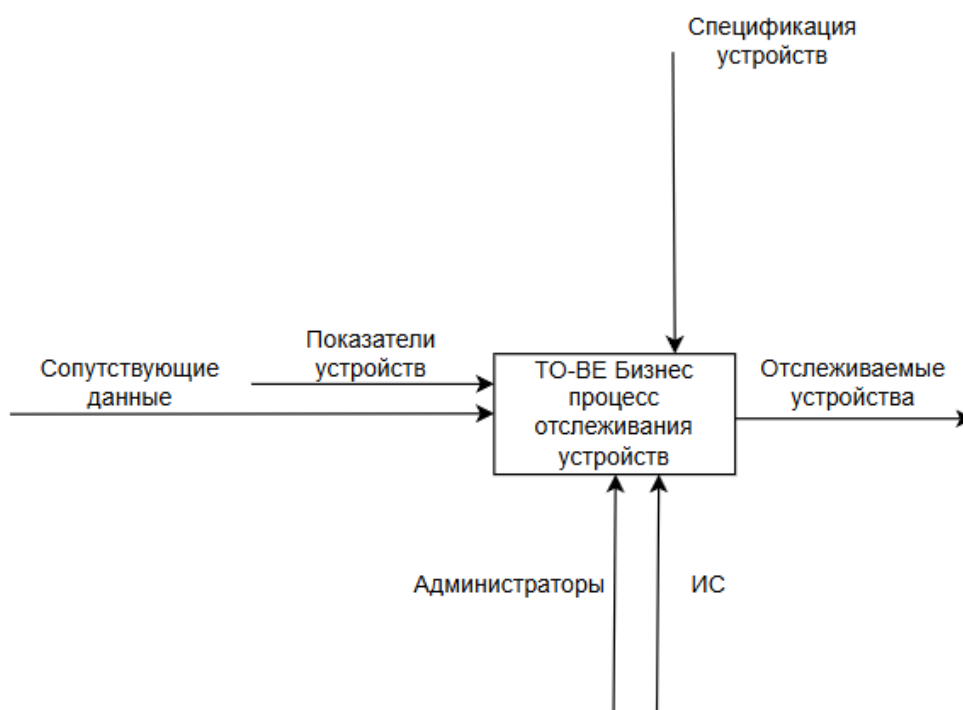


Рисунок 50 – Контекстная диаграмма

Для подробного рассмотрения работы приложения необходимо провести процесс декомпозиции контекстной диаграммы на рисунке 51.

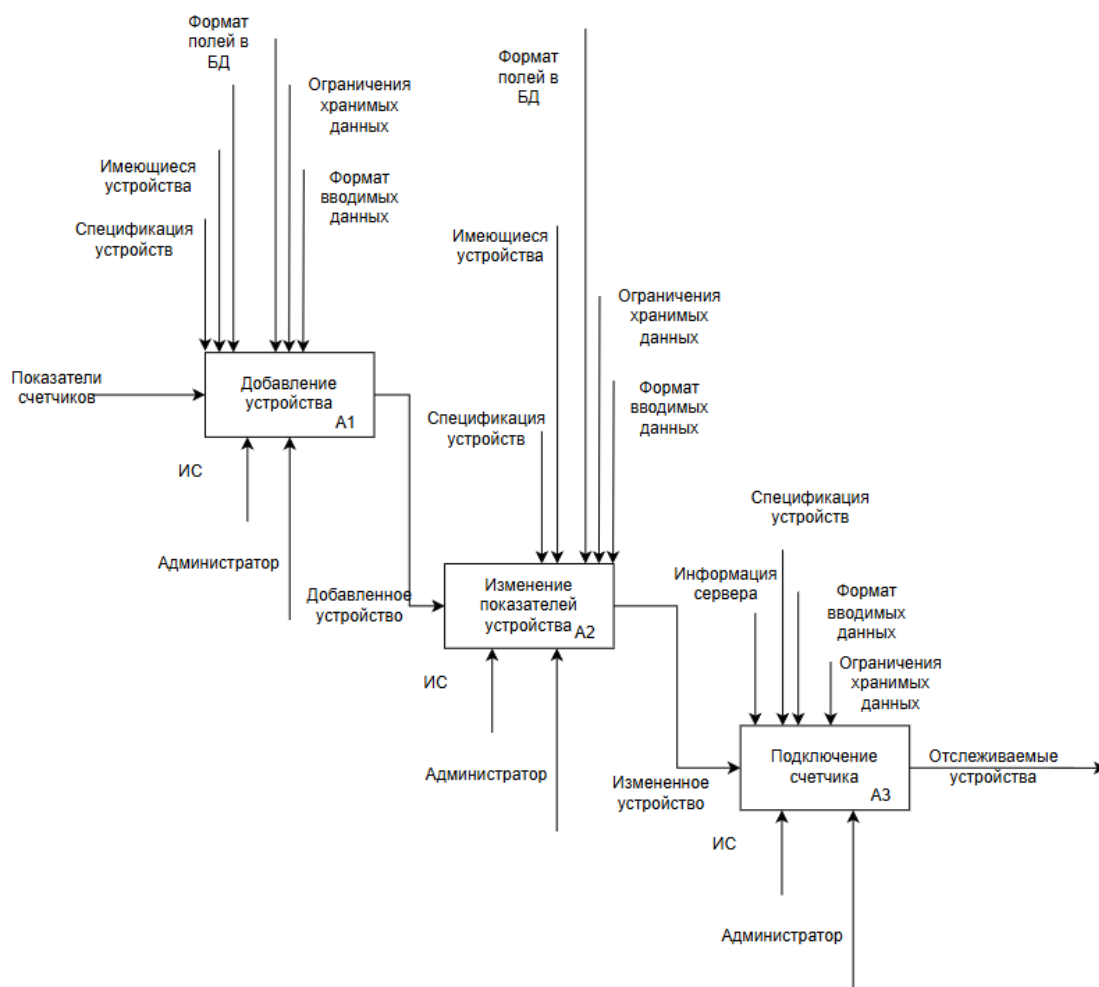


Рисунок 51 – Диаграмма декомпозиции

Ниже на рисунках 52-53 представлены модели IDEF0 для приложения контроля аварийных ситуаций устройств.

Ниже указан бизнес-процесс «Добавление устройства» модели А-1. Выходом блока «Добавление устройства» является добавленное устройство в БД в процессе выполнения данного бизнес-процесса.

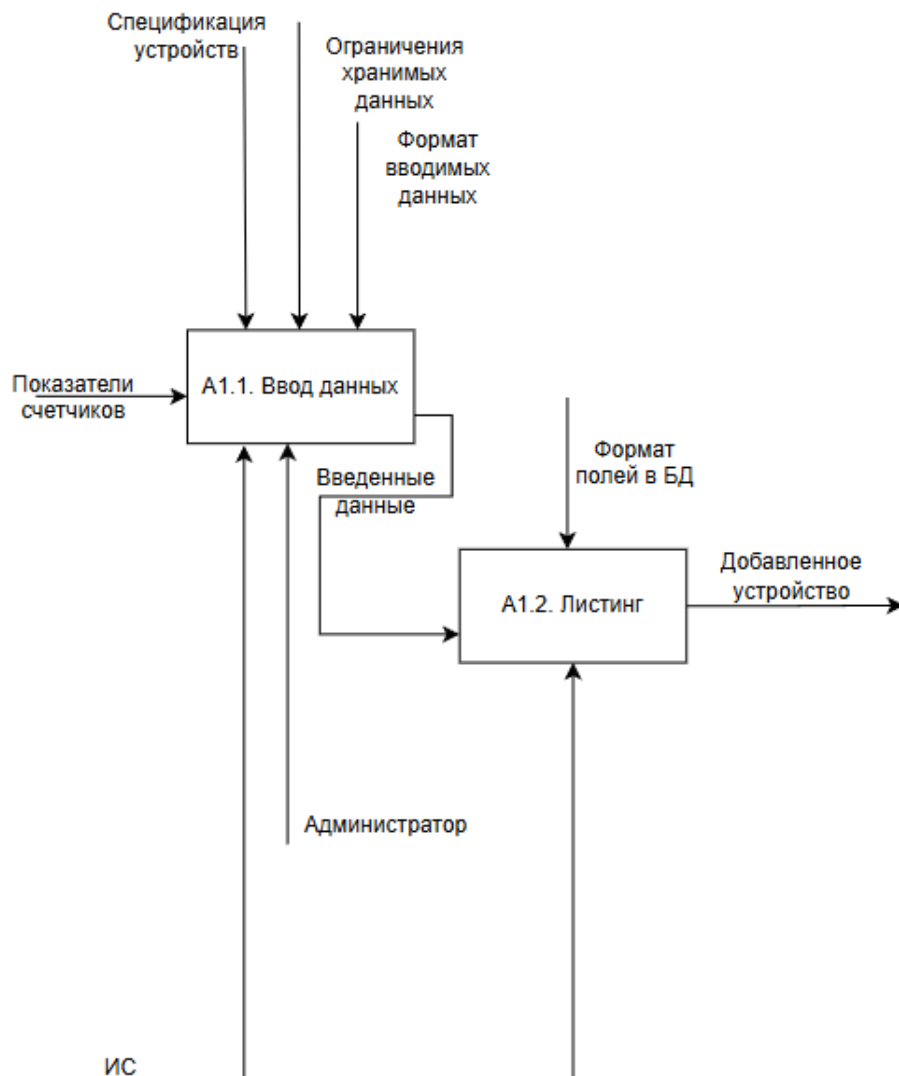


Рисунок 52 – Модель A1, добавление устройства

На рисунке 53 представлена модель A2, которая отвечает за изменение показателей устройства.

На рисунке 55 представлена диаграмма прецедентов для приложения контроля аварийных ситуаций устройств.

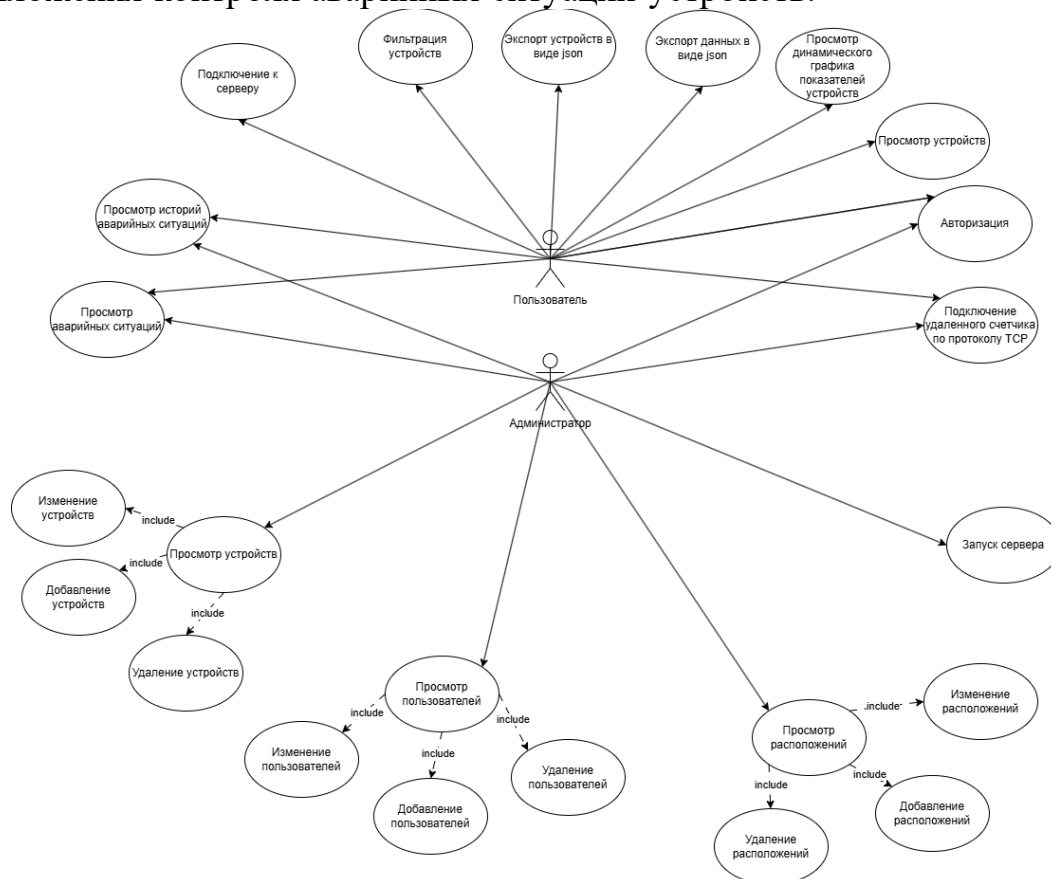


Рисунок 55 – Диаграмма прецедентов

На рисунке 56 представлен алгоритм авторизации внутри программы.

В алгоритме «Авторизация» пользователь открывает окно авторизации, вводит данные для авторизации, далее данные передаются в логику приложения, где она сравнивается с имеющимися данными из БД, и возвращает результат: найден ли пользователь с таким логином и паролем или же нет. Если результат не удовлетворителен, то пользователя переносит обратно на этап ввода данных для авторизации. Если же пользователь найден, то для начала проверяется наличие у него подходящей роли. Если роль не подходит, то пользователя возвращает на этап ввода данных для авторизации, если подходит, то пользователь авторизуется и переходит в основную часть приложения.

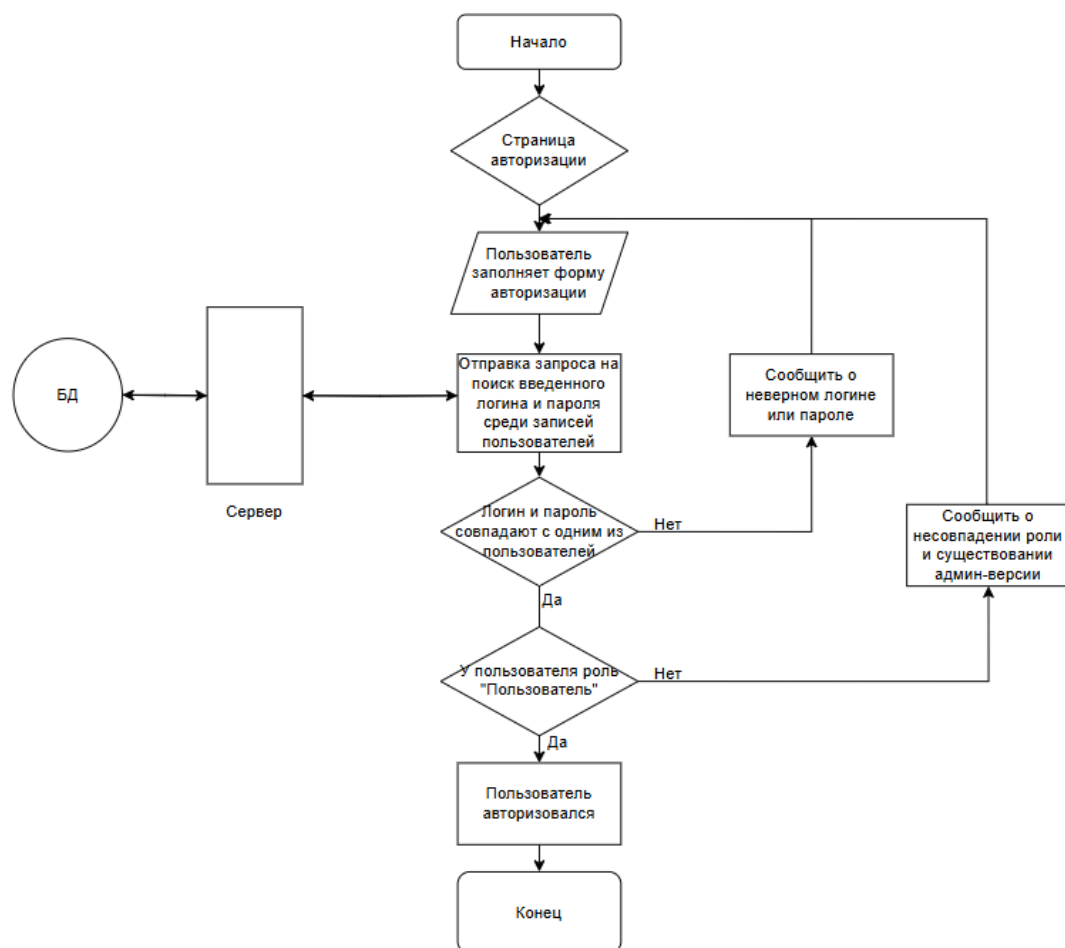


Рисунок 56 – Алгоритм авторизации