

**«Санкт-Петербургский государственный электротехнический университет  
«ЛЭТИ» им. В.И.Ульянова (Ленина)»  
(СПбГЭТУ «ЛЭТИ»)**

<b>Направление</b>	09.03.04 - Программная инженерия
<b>Профиль</b>	Разработка программно-информационных систем
<b>Факультет</b>	КТИ
<b>Кафедра</b>	МО ЭВМ

*К защите допустить*

Зав. кафедрой

А.А. Лисс

**ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА  
БАКАЛАВРА**

**Тема: РАЗРАБОТКА ПРОГРАММНОГО КОМПЛЕКСА ДЛЯ  
МОНИТОРИНГА ЧАСТОТНЫХ ПРЕОБРАЗОВАТЕЛЕЙ**

Студент		<hr/>	Н.А. Кузнецов
		<i>подпись</i>	
Руководитель	К.Т.Н., доцент	<hr/>	Н.В. Мальцева
	<i>(Уч. степень, уч. звание)</i>	<i>подпись</i>	
Консультанты	специалист	<hr/>	М.Н. Нуйкина
	<i>(Уч. степень, уч. звание)</i>	<i>подпись</i>	
		<hr/>	А.А. Трусов
		<i>подпись</i>	
	К.Т.Н., доцент	<hr/>	М.М. Заславский
	<i>(Уч. степень, уч. звание)</i>	<i>подпись</i>	

Санкт-Петербург

2025

## ЗАДАНИЕ НА ВЫПУСКНУЮ КВАЛИФИКАЦИОННУЮ РАБОТУ

Утверждаю

Зав. кафедрой МО ЭВМ

\_\_\_\_\_ А.А. Лисс

«\_\_\_» \_\_\_\_\_ 20\_\_ г.

Студент        Кузнецов Н.А.

Группа    1303

Тема работы: Разработка программного комплекса для мониторинга частотных преобразователей

Место выполнения ВКР: СПбГЭТУ «ЛЭТИ»

Исходные данные (технические требования):

Разработать систему мониторинга частотных преобразователей

Содержание ВКР:

Введение, Обзор предметной области, Формулировка требований к решению и постановка задачи, Описание решения, Характеристика решения, Безопасность жизнедеятельности, Заключение

Перечень отчетных материалов: пояснительная записка, иллюстративный материал

Дополнительные разделы: безопасность жизнедеятельности

Дата выдачи задания

«\_\_\_» \_\_\_\_\_ 20\_\_ г.

Дата представления ВКР к защите

«\_\_\_» \_\_\_\_\_ 20\_\_ г.

Студент

\_\_\_\_\_

Н.А. Кузнецов

Руководитель к.т.н., доцент

(Уч. степень, уч. звание)

\_\_\_\_\_

Н.В. Мальцева

Консультант специалист

(Уч. степень, уч. звание)

\_\_\_\_\_

М.Н. Нуйкина

# КАЛЕНДАРНЫЙ ПЛАН ВЫПОЛНЕНИЯ ВЫПУСКНОЙ КВАЛИФИКАЦИОННОЙ РАБОТЫ

Утверждаю  
Зав. кафедрой МО ЭВМ  
\_\_\_\_\_ А.А. Лисс  
« \_\_\_\_ » \_\_\_\_\_ 20 \_\_\_\_ г.

Студент Кузнецов Н.А. Группа 1303  
Тема работы: Разработка программного комплекса для мониторинга  
частотных преобразователей

№ п/п	Наименование работ	Срок выполнения
1	Обзор литературы по теме работы	24.03 – 31.03
2	Обзор предметной области	01.04 – 06.04
3	Формулировка требований к решению	07.04 – 13.04
4	Описание метода решения	14.04 – 20.04
5	Характеристики решения	21.04 – 27.04
6	Безопасность жизнедеятельности	28.04 – 29.04
7	Приложение А. Направления развития	30.04
8	Оформление пояснительной записки	01.05 – 06.05
9	Оформление иллюстративного материала	07.05 – 08.05
21	Предзащита	20.05

Студент \_\_\_\_\_ Н.А. Кузнецов

Руководитель к.т.н., доцент \_\_\_\_\_ Н.В. Мальцева  
(Уч. степень, уч. звание)

Консультант специалист \_\_\_\_\_ М.Н. Нуйкина  
(Уч. степень, уч. звание)

## РЕФЕРАТ

Пояснительная записка 58 стр., 18 рис., 4 табл., 19 ист.

**МОНИТОРИНГ, ЧАСТОТНЫЙ ПРЕОБРАЗОВАТЕЛЬ, ВРЕМЕННОЙ РЯД.**

**Объектом исследования** является программное обеспечение систем мониторинга частотных преобразователей.

**Предметом исследования** являются архитектурные и технологические решения для сбора, обработки и визуализации данных частотных преобразователей.

**Цель работы:** Разработка программного комплекса для мониторинга частотных преобразователей.

В работе представлен анализ и разработка программного комплекса для мониторинга частотных преобразователей. Проведено сравнение существующих систем мониторинга по ключевым критериям: частота опроса данных, максимальное количество поддерживаемых устройств, количество отслеживаемых параметров и кроссплатформенность. Разработано решение на базе микросервисной архитектуры, включающее кроссплатформенное клиентское приложение, высокопроизводительные сервисы обработки временных рядов, а также архитектура баз данных для хранения временных рядов и метаданных. Предложенная система обеспечивает возможность высокочастотного опроса оборудования, расширенный набор отслеживаемых параметров и возможность визуализации. Комплекс поддерживает как облачное, так и локальное развертывание с учетом требований информационной безопасности предприятий.

## **ABSTRACT**

The paper presents the analysis and development of a software package for monitoring frequency converters. A comparison of existing monitoring systems was carried out according to key criteria: the frequency of data polling, the maximum number of supported devices, the number of monitored parameters, and cross-platform compatibility. A microservice architecture-based solution has been developed, including a cross-platform client application, high-performance time series processing services, as well as a database architecture for storing time series and metadata. The proposed system provides the possibility of high-frequency polling of equipment, an expanded set of monitored parameters and the possibility of visualization. The complex supports both cloud and on-premises deployment, taking into account the requirements of enterprise information security.

## СОДЕРЖАНИЕ

ВВЕДЕНИЕ .....	10
1 Обзор предметной области .....	12
1.1 Принцип отбора аналогов .....	12
1.2 Аналоги .....	12
1.2.1 ABB Ability™ Condition Monitoring for Drives .....	12
1.2.2 Siemens MindSphere Drive Analytics .....	12
1.2.3 Schneider Electric EcoStruxure Machine Advisor .....	13
1.2.4 Emerson Plantweb Optics .....	13
1.2.5 Rockwell Automation FactoryTalk® .....	14
1.3 Критерии сравнения аналогов .....	14
1.3.1 Частота опроса данных .....	14
1.3.2 Максимальное количество устройств .....	14
1.3.3 Количество параметров мониторинга .....	15
1.3.4 Кроссплатформенность .....	15
1.4 Таблица сравнения аналогов .....	15
1.5 Выводы по итогам сравнения .....	16
2 Выбор метода решения .....	17
2.1 Требования .....	17
2.2 Выбор технологий .....	17
2.2.1 Dart и Flutter .....	17
2.2.2 Go .....	17
2.2.3 Python с Flask .....	18
2.2.4 MongoDB .....	18
2.2.5 InfluxDB .....	18
3 Описание метода решения .....	19
3.1 Связь с аппаратным комплексом .....	19
3.2 Описание основных компонент системы и их взаимодействия .....	19
3.2.1 Базы данных .....	19

3.2.2 Сервисы обработки временных рядов .....	19
3.2.3 Сервис отчетов .....	20
3.2.4 Сервис уведомлений .....	20
3.2.5 Бэкенд .....	20
3.2.6 Клиентская часть .....	21
3.3 Клиентское приложение .....	22
3.3.1 Архитектура клиентского приложения .....	22
3.3 Части приложения .....	25
3.3.3 Главная страница .....	26
3.3.4 Страница графиков .....	27
3.3.5 Страница событий .....	29
3.3.6 Страница статистики .....	30
3.3.7 Страница аналитики .....	31
3.3.8 Страница настроек .....	32
3.3.9 Локальная база данных Flutter Hive .....	33
3.4 Серверное приложение по обработке клиентских запросов .....	34
3.4.1 Модули серверного приложения .....	34
3.4.3 Особенности реализации .....	35
3.4.4 Взаимодействие с клиентской частью .....	35
3.5 Сервис уведомлений .....	36
3.5.1 Архитектура сервиса уведомлений .....	36
3.5.2 Функциональные возможности .....	37
3.5.3 Механизм работы .....	38
3.6 Сервис отчетов .....	38
3.6.1 Архитектура сервиса отчетов .....	38
3.6.2 Функциональные возможности .....	40
3.6.4 Взаимодействие с другими компонентами системы .....	40
3.7 Сервис обработки низкочастотных данных .....	41
3.7.1 Архитектура сервиса .....	41
3.7.2 Принцип работы сервиса .....	42

3.8 Сервис обработки высокочастотных данных .....	43
3.9 Панель администратора .....	43
3.10 Развертывание системы .....	45
4 ХАРАКТЕРИСТИКИ РЕШЕНИЯ .....	48
5 Безопасность жизнедеятельности .....	50
5.1 Основные положения .....	50
5.2 Анализ факторов воздействия .....	50
5.3 Снижение воздействия вредных факторов .....	53
5.3.1 Требования к освещению рабочего места .....	53
5.3.2 Защита от электромагнитных излучений .....	53
5.3.3 Обеспечение комфортного микроклимата .....	54
5.3.4 Снижение шумового воздействия .....	54
5.3.5 Снижение психофизиологических нагрузок .....	54
5.4 Выводы .....	55
ЗАКЛЮЧЕНИЕ .....	56
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ .....	57
ПРИЛОЖЕНИЕ А. НАПРАВЛЕНИЕ РАЗВИТИЯ .....	59
ПРИЛОЖЕНИЕ Б. КОНФЕРЕНЦИЯ “НАУКА НАСТОЯЩЕГО И БУДУЩЕГО” .....	60



## **ОПРЕДЕЛЕНИЯ, ОБОЗНАЧЕНИЯ И СОКРАЩЕНИЯ**

Частотный преобразователь - электронное устройство, предназначенное для управления частотой и напряжением электропитания асинхронных или синхронных электродвигателей. Используется для регулирования скорости вращения двигателя и оптимизации энергопотребления.

Мониторинг промышленного оборудования - система или процесс непрерывного наблюдения, анализа и регистрации параметров работы промышленного оборудования с целью повышения надежности, выявления неисправностей, оптимизации производительности и предотвращения аварийных ситуаций.

JWT-token (JSON Web Token) стандарт передачи информации между сторонами в формате JSON, закодированной и подписанной для обеспечения безопасности. Используется для аутентификации и авторизации пользователей.

CPU (Central Processing Unit) - центральный процессор — основной вычислительный элемент компьютера, выполняющий арифметические, логические и управляющие операции, необходимые для выполнения программ.

RAM (Random Access Memory) - оперативная память компьютера, в которой временно хранятся данные и команды, необходимые процессору для выполнения задач. Обеспечивает высокую скорость доступа к информации.

Кроссплатформенность - свойство программного обеспечения или оборудования, позволяющее его использование на различных операционных системах или аппаратных платформах без необходимости значительных изменений.

Фреймворк - набор готовых инструментов, библиотек и архитектурных решений, предоставляющий основу для разработки программного обеспечения.

## ВВЕДЕНИЕ

В современном промышленном производстве частотные преобразователи играют ключевую роль в управлении частотными преобразователями. Разработка специализированного программного обеспечения [1] для визуализации и анализа данных является актуальной задачей, поскольку существующие решения имеют ограничения при адаптации под конкретные производственные условия. Например, недостаточная частота опроса данных для анализа формы сигнала или отсутствие возможности добавления специфических параметров мониторинга. Кроме того, зарубежные решения могут быть недоступны, а требования информационной безопасности предприятий часто не позволяют использовать облачные сервисы, требуя локального развертывания системы. Создание комплексной системы мониторинга направлено на оптимизацию технического обслуживания за счет раннего обнаружения отклонений в работе оборудования, предотвращения внеплановых простоев и более точного планирования профилактических работ на основе реальной нагрузки устройств. Это позволяет существенно снизить эксплуатационные расходы и продлить срок службы оборудования.

Объектом исследования является программное обеспечение систем мониторинга частотных преобразователей. Предметом исследования является архитектурные и технологические решения для сбора, обработки и визуализации данных частотных преобразователей.

Целью является разработка программного комплекса для мониторинга частотных преобразователей. Для достижения цели необходимо решить следующие задачи:

- Провести обзор существующих систем мониторинга частотных преобразователей.
- Определить критерии сравнения систем мониторинга частотных преобразователей и произвести сравнение.

- Разработать кроссплатформенное клиентское приложение для визуализации данных и управления системой.
- Разработать серверную часть приложения для обработки временных рядов и клиентских запросов.
- Разработать систему контейнеризации и развертывания программного комплекса.

Для определения существующих подходов решения, их преимуществ и недостатков, проведем обзор предметной области.

## **1 Обзор предметной области**

### **1.1 Принцип отбора аналогов**

Поиск аналогов производился по научным базам данных IEEE Xplore с использованием поисковых запросов: "industrial equipment monitoring system", "frequency converter monitoring", "industrial IoT monitoring platform". Дополнительно использовались коммерческие решения, найденные через Google Scholar по запросу "industrial monitoring solution". Отбирались системы, имеющие функциональность мониторинга промышленного оборудования с акцентом на преобразователи частоты.

Используя описанную методологию поиска, были отобраны пять наиболее релевантных систем мониторинга. Рассмотрим последовательно каждую из них.

### **1.2 Аналоги**

#### **1.2.1 ABB Ability™ Condition Monitoring for Drives**

Данная система [2] представляет собой специализированное решение для мониторинга частотных преобразователей производства ABB. Она работает по трехуровневой архитектуре: датчики подключаются к шлюзу ABB Ability Edge, который передает данные в облачную платформу.

Основные возможности включают мониторинг более 25 параметров в реальном времени. Среди них - электрические характеристики (напряжение, ток, мощность), показатели качества электроэнергии, включая коэффициент нелинейных искажений до 50-й гармоники, а также механические параметры (вибрация, температура подшипников). Частота опроса составляет от 10 до 100 Гц для основных параметров, а вибродиагностика выполняется с частотой 1 кГц через внешние датчики.

Система оснащена продвинутыми аналитическими функциями: прогнозирование остаточного ресурса изоляции, ранняя диагностика межвитковых замыканий, анализ дисбаланса ротора. Однако решение работает только с оборудованием ABB.

#### **1.2.2 Siemens MindSphere Drive Analytics**

Эта платформа [3] предлагает гибридную архитектуру, сочетающую edge-вычисления и облачную аналитику. Она поддерживает современные протоколы промышленного интернета вещей, включая OPC UA, MQTT и REST API, что обеспечивает интеграцию с существующей инфраструктурой предприятия.

Система собирает более 50 параметров с частотных преобразователей SINAMICS. Она позволяет регистрировать мгновенные значения с периодом 20 мс, выполнять осциллографическую запись процессов с разрешением 1 мс, анализировать профили скорости и момента. Особенностью является глубокая диагностика, включающая спектральный анализ тока, контроль параметров силовых транзисторов IGBT и создание цифровых двойников.

Аналитический блок поддерживает разработку предиктивных моделей на Python и R, а также обеспечивает интеграцию с PLM-системой Teamcenter. Однако система требует квалифицированного персонала для настройки и имеет задержку облачной аналитики около 200 мс.

### **1.2.3 Schneider Electric EcoStruxure Machine Advisor**

Это решение [4] предназначено для мониторинга приводов Altivar и использует архитектуру с локальными серверами и облачной платформой. Система поддерживает промышленные протоколы Modbus TCP и Ethernet/IP.

Функциональные возможности включают базовый мониторинг 12 стандартных параметров с частотой опроса от 50 до 200 мс. Система позволяет записывать аварийные осциллограммы и хранить исторические тренды за 365 дней. Аналитический блок работает на основе правил стандарта IEC 61131-3, что упрощает адаптацию для специалистов по автоматизации.

Визуализация данных осуществляется через Aveva PI System, а открытый API позволяет интегрировать решение с системами CMMS. Главными ограничениями являются отсутствие встроенного анализа гармоник и относительно низкая максимальная частота опроса.

### **1.2.4 Emerson Plantweb Optics**

Данная система [5] использует беспроводные датчики (WirelessHART, Zigbee) для косвенного мониторинга состояния оборудования. В состав системы входят виброакселерометры с частотой опроса 1 кГц, беспроводные термопары и токовые клещи с частотой 10 Гц.

Основные преимущества включают быстрый монтаж (до 1 дня), автономную работу датчиков до 5 лет и эффективное обнаружение различных неисправностей. Однако система не предоставляет прямого доступа к параметрам частотных преобразователей и имеет задержку данных до 1 секунды.

### **1.2.5 Rockwell Automation FactoryTalk®**

Это решение [6] глубоко интегрировано с платформой ControlLogix PLC и средой разработки Studio 5000. Оно обеспечивает мониторинг 8-12 основных параметров с частотой опроса от 100 до 500 мс.

Система включает встроенные инструменты визуализации и конфигурируется через Add-On Instructions. Аналитические возможности содержат пакет Energy Monitoring, алгоритмы расчета экономии энергии и интеграцию с ЕАМ-системами. Главным недостатком является жесткая привязка к экосистеме Rockwell.

## **1.3 Критерии сравнения аналогов**

Для сравнения аналогов были отобраны важные критерии систем мониторинга частотных преобразователей.

### **1.3.1 Частота опроса данных**

Максимальная частота, с которой система может опрашивать устройства (в Гц). Чем больше частота (меньше период), тем больше точек данных может анализировать система. Критерий важен для обеспечения точного мониторинга быстроизменяющихся параметров частотных преобразователей и своевременного обнаружения отклонений, например, форма синусоиды напряжения, для определения качества питания.

### **1.3.2 Максимальное количество устройств**

Количество устройств, которые система может одновременно обслуживать в рамках одного кластера с сохранением времени на ответ не более 1с. для большинства сценариев использования. Самый времязатратный сценарий использования это построение графиков, для которого сервер должен сделать запрос в influx. Для стандартного запроса на получение  $10.000 \cdot (\text{количество оборудования})$  точек время ответа сервера не должно составлять более 1с.

### 1.3.3 Количество параметров мониторинга

Число основных параметров, которые система отслеживает для каждого частотного преобразователя. Такие как: входное/выходное напряжение, ток, частоту, температуру и параметры состояния.

### 1.3.4 Кроссплатформенность

Доступность системы на различных платформах и операционных системах, что обеспечивает возможность использования на различных участках и помещениях предприятия.

## 1.4 Таблица сравнения аналогов

У рассмотренных аналогов выделены параметры по выделенным критериям, результаты представлены в таблице 1.

Таблица 1 – Сравнение аналогов

Система	Параметры сравнения			
	Частота опроса (Гц)	Макс. кол-во устройств	Кол-во параметров	Поддерживаемые платформы
ABB Ability	50	150	20	Web, Windows, Android, iOS
Siemens MindSphere	50	500	50	Web, Windows, Android, iOS
EcoStruxure	20	200	12-16	Web, Windows, Android, iOS
Plantweb Optics	10	300	10-14	Windows, web
Rockwell FactoryTalk	10	100	8-12	Windows, web

### **1.5 Выводы по итогам сравнения**

По результатам сравнения можно сделать вывод, что Siemens MindSphere лидирует по техническим характеристикам: максимальная частота опроса (50 Гц), максимальная масштабируемость (до 500 устройств), наибольшее количество отслеживаемых параметров (50) и широкая поддержка различных платформ. Все рассмотренные системы являются кроссплатформенными, что является необходимым требованием для современных промышленных систем мониторинга, обеспечивая доступ к данным с различных устройств на производстве. Все решения способны обслуживать десятки устройств одновременно, что позволяет применять их на крупных промышленных предприятиях.

Однако для каждого конкретного случая мониторинга требуются свои дополнительные специфические параметры, которые не отслеживаются в готовых решениях. Например, для диагностики качества выходного напряжения частотного преобразователя необходим анализ гармонических составляющих напряжения, который предусмотрен не во всех рассмотренных системах. Кроме того, для точного анализа формы сигнала переменного тока частотой 50 Гц необходима период опроса менее 1 мс (1000 Гц).



## **2 Выбор метода решения**

### **2.1 Требования**

На основе проведенного анализа существующих систем определены ключевые требования к разрабатываемому решению:

- Обеспечение поддержки периода опроса 300 кГц для точного анализа формы сигнала переменного тока частотой 50 Гц.
- Расширяемый набор отслеживаемых параметров, минимальные: сила тока, напряжение и мощность по трем фазам на входе и на выходе частотного преобразователя, межфазные углы, пик-фактор.
- Возможность анализа гармонических составляющих напряжения для диагностики качества выходного сигнала.
- Масштабируемость до 100 устройств с сохранением высокой производительности, для мониторинга больших производств.
- Развертывание как в облаке, так и локально в соответствии с требованиями информационной безопасности.

### **2.2 Выбор технологий**

Для реализации этих требований предлагается разработать распределенную систему на базе микросервисной архитектуры. Сначала определим архитектуру основных компонентов (рис. 1) и используемые технологии:

#### **2.2.1 Dart и Flutter**

Для визуализации измерений Dart с фреймворком Flutter [7], [8] предоставляет высокопроизводительные виджеты для построения графиков в реальном времени. Его архитектура позволяет эффективно отображать большие наборы данных, что критично для анализа формы сигнала переменного тока. Кроссплатформенность Flutter обеспечит доступ к системе мониторинга как с мобильных устройств, так и через веб-интерфейс, что важно для оперативного контроля параметров оборудования.

#### **2.2.2 Go**

Язык Go [9] подходит для обработки высокочастотных измерений благодаря эффективной работе с памятью и поддержке параллельных вычислений. Его производительность позволит обрабатывать данные в реальном времени от множества устройств одновременно, выполняя анализ гармонических составляющих напряжения без потери производительности. Go также хорошо подходит для развертывания как в облаке, так и в локальной сети благодаря простой компиляции в независимый исполняемый файл.

### **2.2.3 Python с Flask**

Python с фреймворком Flask [10], [11] обеспечит REST API для конфигурации системы и доступа к данным. Flask легко масштабируется и может быть развернут как в облаке, так и локально, поддерживая различные сценарии развертывания согласно требованиям безопасности.

### **2.2.4 MongoDB**

MongoDB [12] обеспечит гибкое хранение конфигураций для расширяемого набора отслеживаемых параметров. Её горизонтальная масштабируемость позволит обслуживать растущее количество устройств, а поддержка шардинга поможет распределить нагрузку при масштабировании системы. MongoDB может быть развернута как в облаке, так и локально, обеспечивая необходимый уровень безопасности.

### **2.2.5 InfluxDB**

InfluxDB [13] специализированная база данных для хранения временных рядов. Она обеспечит эффективное хранение и быстрый доступ к данным о токе, напряжении и мощности по фазам. Встроенные функции агрегации и анализа временных рядов упростят анализ. InfluxDB хорошо масштабируется для работы с множеством устройств и поддерживает как облачное, так и локальное развертывание.

### **3 Описание метода решения**

#### **3.1 Связь с аппаратным комплексом**

Для понимания как данные приходят в систему и какой формат они имеют нужно рассмотреть связь с аппаратным комплексом, вместе с которым описываемая система формирует готовый продукт.

Датчики на частотных преобразователях передают «сырые» данные на контроллер, расположенный на предприятии, на контроллере по полученным данным происходит расчет необходимых метрик. Контроллер опрашивает все подключенные устройства по протоколу Modbus и публикует полученные данные в локальный MQTT брокер. Далее данные по протоколу http отправляются на сервер.

Данные от контроллера имеют два типа. Низкочастотные данные (с частотой 5-10 Гц) напрямую пишутся в Influx, например, средние показатели силы тока, напряжения, мощности, межфазового угла, пик-фактор и другие. Высокочастотные данные (с частотой 300 кГц) контроллер периодически и по запросу пользователя отправляет в виде сжатого csv-файла на сервер.

#### **3.2 Описание основных компонент системы и их взаимодействия**

Для удобства создание, поддержки и развития системы поделена на сервисы которые связаны REST API. Далее рассмотрим основные сервисы и их описание.

##### **3.2.1 Базы данных**

Система использует две нереляционные базы данных. InfluxDB хранит показания по параметрам оборудования. MongoDB хранит информацию о конфигурации оборудования, превышения и предупреждения, аномалии, данные об организациях и пользователях, журналы событий.

##### **3.2.2 Сервисы обработки временных рядов**

Данные оборудования обрабатываются двумя сервисами. Первый сервис просматривает ряды из Influx и определяет превышения нижних и верхних пороговых значений, формируя записи о событиях и сохраняя их в

Mongo, так же отправляет данные в сервис уведомлений. Второй сервис на основе математических моделей, диагностических правил и нейросети для выявления аномалий по высокочастотным данным определяет шаблонные случаи неисправности компонентов и также формирует записи в Mongo и отправляет данные в сервис уведомлений.

### **3.2.3 Сервис отчетов**

Сервис отвечает за формирование отчетов в форматах pdf и excel, которые далее будут периодически рассылаться пользователям системы или по запросу в клиентском приложении. Так же сервис подготавливает данные для ежедневной, еженедельной и другой статистики оборудования для быстрого формирования отчетов, например расчета квартилей параметров по дням.

### **3.2.4 Сервис уведомлений**

Принимает запросы от сервисов обработки временных рядов и формирует уведомления на основе шаблонов. Обеспечивает информирование пользователей о подписке, изменениях настроек подписки и отписке. Формирует и рассылает уведомления о превышениях пороговых значений и выявленных аномалиях в работе оборудования. Также отправляет пользователям уведомления с периодическими отчетами о состоянии и функционировании оборудования согласно настроенному расписанию.

### **3.2.5 Бэкенд**

Серверная часть системы обрабатывает клиентские запросы и обеспечивает взаимодействие с базами данных. К основным функциям бэкенда относятся:

- Аутентификация и авторизация пользователей с разграничением прав доступа
- Агрегация данных из InfluxDB с различными временными интервалами для оптимизации отображения
- Предоставление клиентской части списка уведомлений об аномалиях и превышениях пороговых значений

- Взаимодействие с сервисом отчетов для формирования и экспорта данных по запросу пользователя
- Управление пользовательскими настройками, комментариями и метаданными
- Кэширование часто запрашиваемых данных для оптимизации производительности и снижения нагрузки на базы данных

### **3.2.6 Клиентская часть**

Клиентская часть системы реализована на фреймворке Flutter, что обеспечивает кроссплатформенность приложения и единую кодовую базу для веб-интерфейса, мобильных и десктопных приложений. Интерфейс пользователя предоставляет следующие возможности:

- Визуализация данных с оборудования в виде графиков, диаграмм и таблиц
- Настройка параметров мониторинга и пороговых значений для различных типов оборудования
- Просмотр истории уведомлений о превышениях и аномалиях
- Работа с отчетами: просмотр, экспорт, настройка параметров формирования
- Управление пользовательскими предпочтениями и настройками интерфейса

Для наглядности взаимодействия сервисов рассмотрим схему представленную на рис. 1.

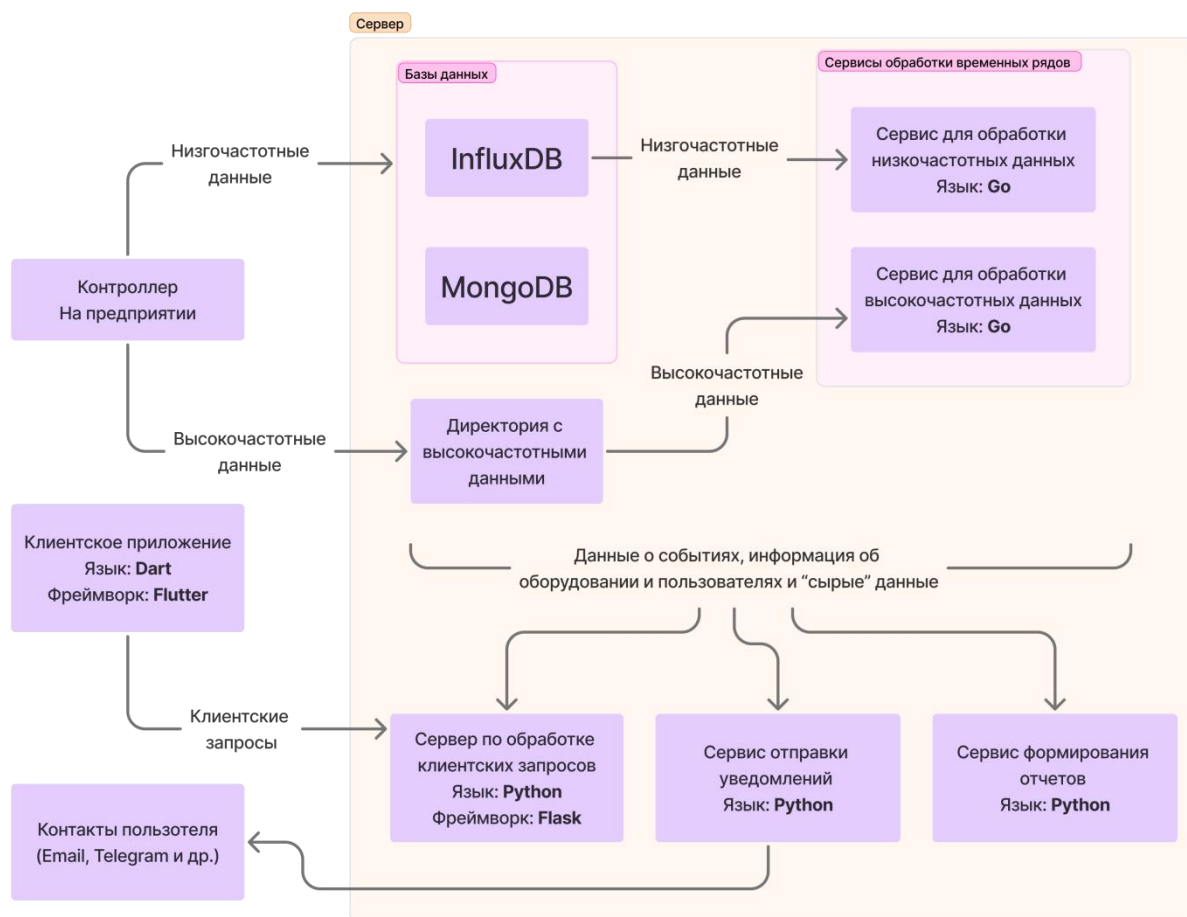


Рисунок 1 – Схема взаимодействия компонентов

### 3.3 Клиентское приложение

#### 3.3.1 Архитектура клиентского приложения

Для создания программного обеспечения была выбрана «Чистая архитектура», схематически представленная на рис. 2. Данный подход представляет собой многоуровневую структуру, разделяющую приложение на определенные слои: уровень данных (Data layer), уровень домена (Domain layer) и уровень представления (Presentation layer).

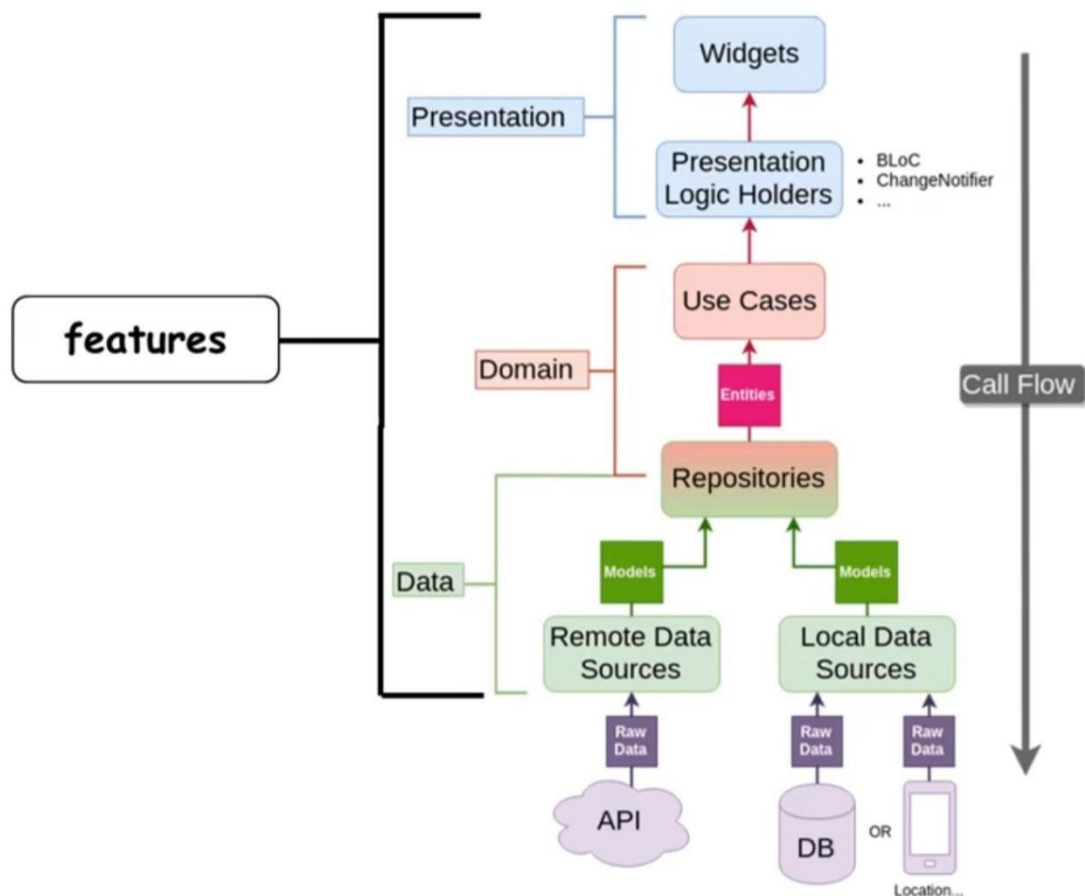


Рисунок 2 – «Чистая архитектура»

Преимущества выбранной архитектуры включают изоляцию слоев, независимость от фреймворков, высокую тестируемость, масштабируемость и адаптивность к изменениям. Например, изменение формата передачи данных с JSON на XML затронет только уровень данных, не требуя модификации бизнес-логики или интерфейса.

Уровень представления взаимодействует с пользователем и отвечает за отображение данных. Здесь находятся UI компоненты, виджеты и BLoC, который играет ключевую роль как посредник между UI и бизнес-логикой. BLoC принимает события (events) от пользовательского интерфейса, обрабатывает их и изменяет состояние приложения отправляя состояния (state), которое затем отображается в UI. Взаимодействие BLoC с UI изображено на рис. 3.

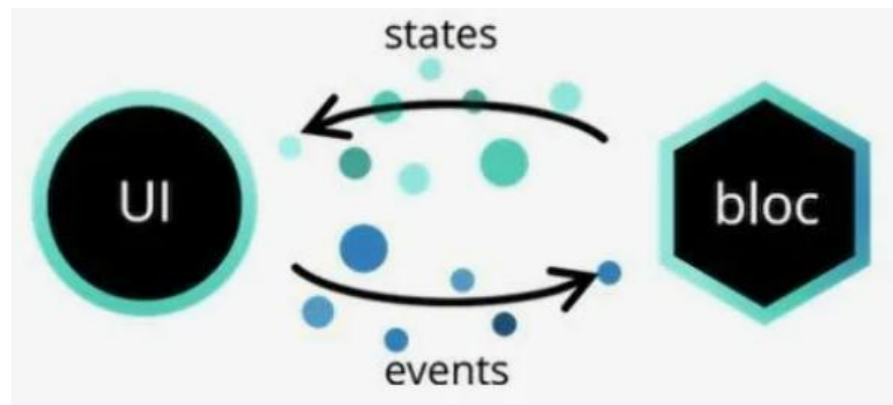


Рисунок 3 – BLoC

Уровень домена содержит ключевую бизнес-логику и правила. Он включает Use Cases (конкретные бизнес-операции), Entities (основные бизнес-объекты) и Repository Interfaces (абстрактные определения методов для доступа к данным). Использование абстрактных интерфейсов репозитория поддерживает принцип инверсии зависимостей, делая домен независимым от конкретных реализаций источников данных.

Уровень данных отвечает за получение и хранение данных из различных источников. Компоненты этого уровня включают Repositories (реализация интерфейсов из домена), Data Sources (компоненты для взаимодействия с удаленными API и локальным хранилищем), Models (объекты для работы с данными), API и DB. На этом уровне все вызовы Data Sources в репозиториях обертываются в конструкции try-catch для обработки потенциальных ошибок. Для удобной передачи результатов используется тип Either.

Поток вызовов (Call Flow) организован так, чтобы минимизировать зависимости между слоями. Пользователь взаимодействует с UI, что генерирует события для BLoC. BLoC обрабатывает события, вызывая соответствующие Use Cases, которые выполняют бизнес-логику через интерфейсы репозитория. Репозитории обращаются к Data Sources, которые взаимодействуют с API или локальным хранилищем. Результаты проходят обратно по цепочке до BLoC, который обновляет состояние, а UI реагирует на изменения.



Благодаря VLoC система обладает высокой реактивностью — интерфейс автоматически обновляется при изменении состояния. Ошибки корректно передаются до уровня представления в понятной пользователю форме.

Выбранная архитектура обеспечивает высокую степень модульности, тестируемости и гибкости приложения. Четкое разделение ответственности позволяет разрабатывать каждый компонент независимо, создавая масштабируемый код, способный адаптироваться к изменяющимся требованиям без существенной переработки кода.

### **3.3 Части приложения**

Для понятного взаимодействия пользователя с приложением реализовано разделение функционала по тематическим страницам. Каждая страница представляет собой блок возможностей, что позволяет пользователю находить нужную информацию и выполнять необходимые действия.

Главная страница служит информационной панелью, предоставляющей комплексный обзор текущего состояния всего контролируемого оборудования. Здесь пользователь может не только отслеживать ключевые показатели в режиме реального времени, но и настраивать пороговые значения параметров, определяющих границы штатной работы оборудования.

На странице предупреждений система собирает и отображает все выявленные события, требующие внимания. Каждое событие можно детально проанализировать с помощью графического представления всех контролируемых параметров, что позволяет установить причинно-следственные связи и принять обоснованные решения по устранению проблем.

Страница графиков является инструментом визуализации данных, обеспечивающим гибкий анализ как высокочастотных, так и низкочастотных показателей за любой выбранный период времени. Функционал включает различные способы агрегации данных, что позволяет исследовать

информацию на разных уровнях детализации и выявлять скрытые закономерности.

Страница аналитики предоставляет углубленный инструментарий для покомпонентного мониторинга состояния частотного преобразователя. Здесь доступен как оперативный контроль текущих событий, так и анализ долгосрочных трендов, что позволяет прогнозировать возможные проблемы и планировать профилактические мероприятия.

Страница статистики обеспечивает пользователя исчерпывающей информацией о режимах эксплуатации оборудования и накопленных статистических данных. Это позволяет оценивать эффективность использования оборудования и принимать решения по оптимизации производственных процессов.

Страница настроек предназначена для персонализации системы под индивидуальные потребности пользователя. Здесь можно управлять личными данными, настраивать параметры уведомлений, определять предпочтительные форматы отображения информации и задавать другие пользовательские настройки, обеспечивающие комфортную работу с системой.

### **3.3.3 Главная страница**

Главная страница приложения представляет собой панель мониторинга (см. рис. 4) , где отображается все подключенное оборудование в формате информационных карточек. Каждая карточка содержит основные параметры конкретной единицы оборудования: текущий статус работы (работает, включено, выключено и не работает), графики выбранных параметров, обновляющиеся в реальном времени.

При взаимодействии пользователя с карточкой оборудования через наведение курсора появляются дополнительные элементы управления:

- Настройки отображаемых параметров
- Доступ к детальной информации об оборудовании
- Возможность редактирования конфигурации и пороговых значений

- Отображение непросмотренных событий для данного оборудования

Система навигации позволяет осуществлять быстрый переход между функциональными разделами непосредственно из карточки оборудования. При выборе определенного раздела (аналитика, графики, статистика) осуществляется автоматическая фильтрация данных по выбранному оборудованию, что сокращает время доступа к необходимой информации.

В верхней части страницы размещена сводная информация о количестве функционирующего оборудования и ключевых параметрах системы. Графики в карточках оборудования отображают динамику изменения параметров, достаточным для оценки текущего состояния оборудования.

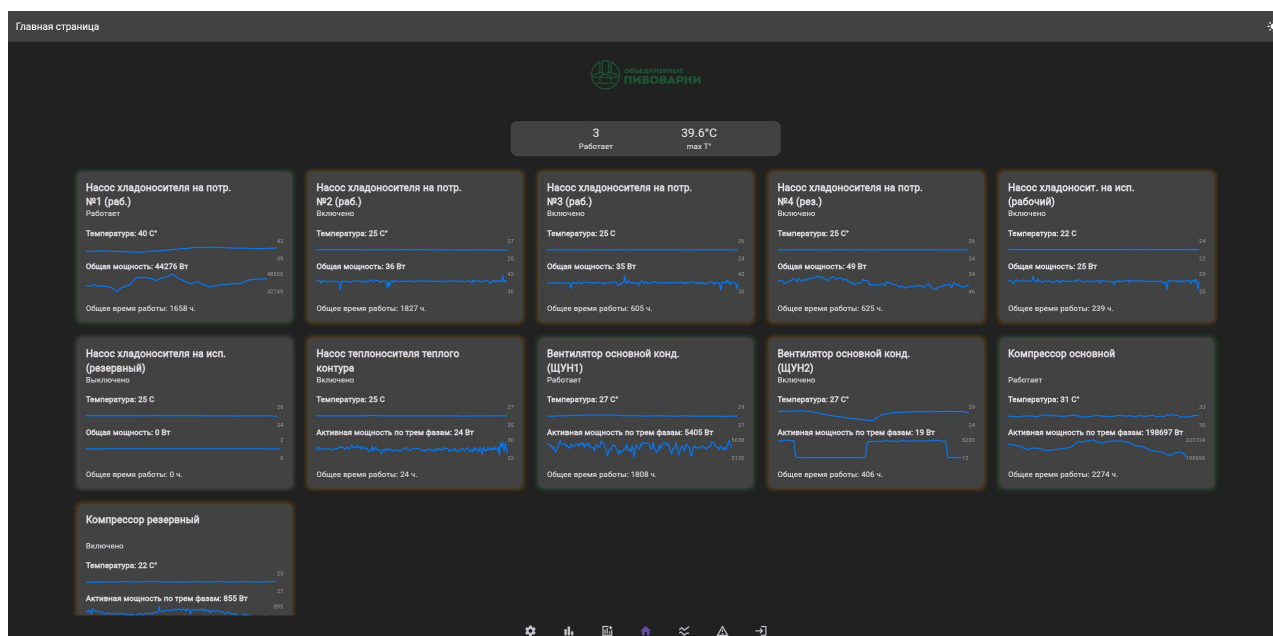


Рисунок 4 – Главная страница

### 3.3.4 Страница графиков

Просмотр высокочастотных и низкочастотных данных возможен при переключении режима отображения. В низкочастотном режиме (см. рис. 5) можно выбрать одно или нескольких оборудования, нужные параметры, функцию, интервал и период агрегации. Так же есть возможность просматривать графики в реальном времени, при котором графики обновляются с периодом равным интервалу агрегации.

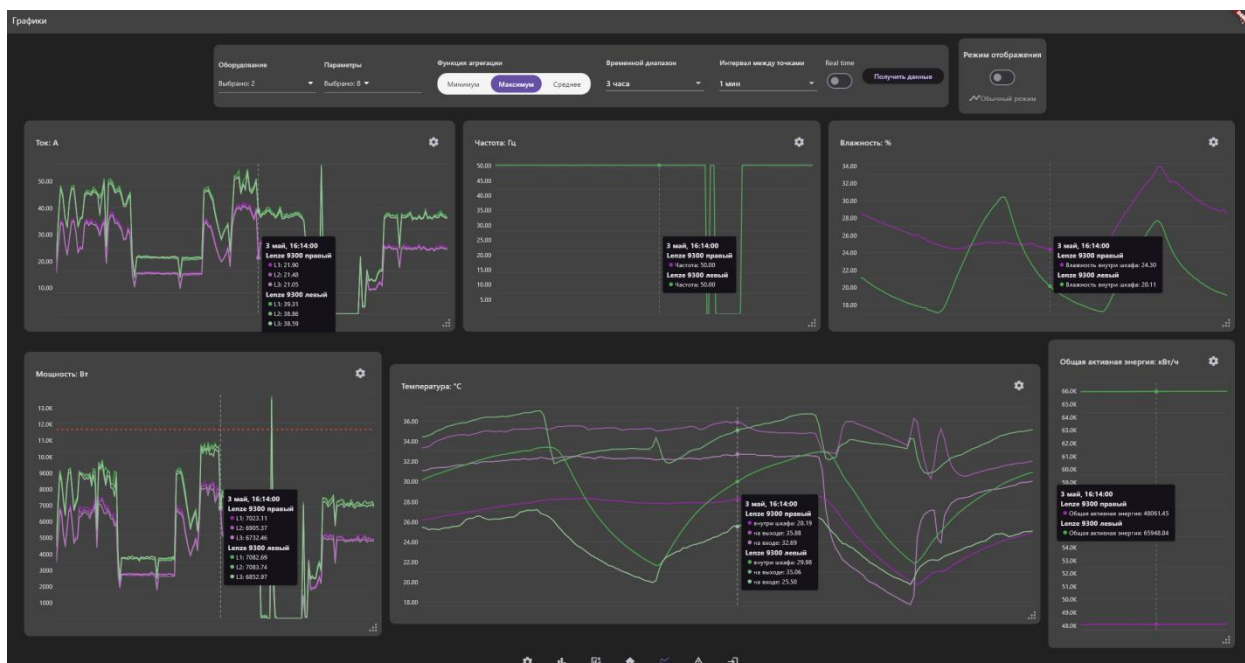


Рисунок 5 – Низкочастотный режим страницы графиков

В высокочастотном режиме (см. рис. 6) можно просмотреть высокочастотные «сэмплы» данных, так же к данным применяются быстрые преобразования Фурье, результаты которых отображается на графике АЧХ ниже.

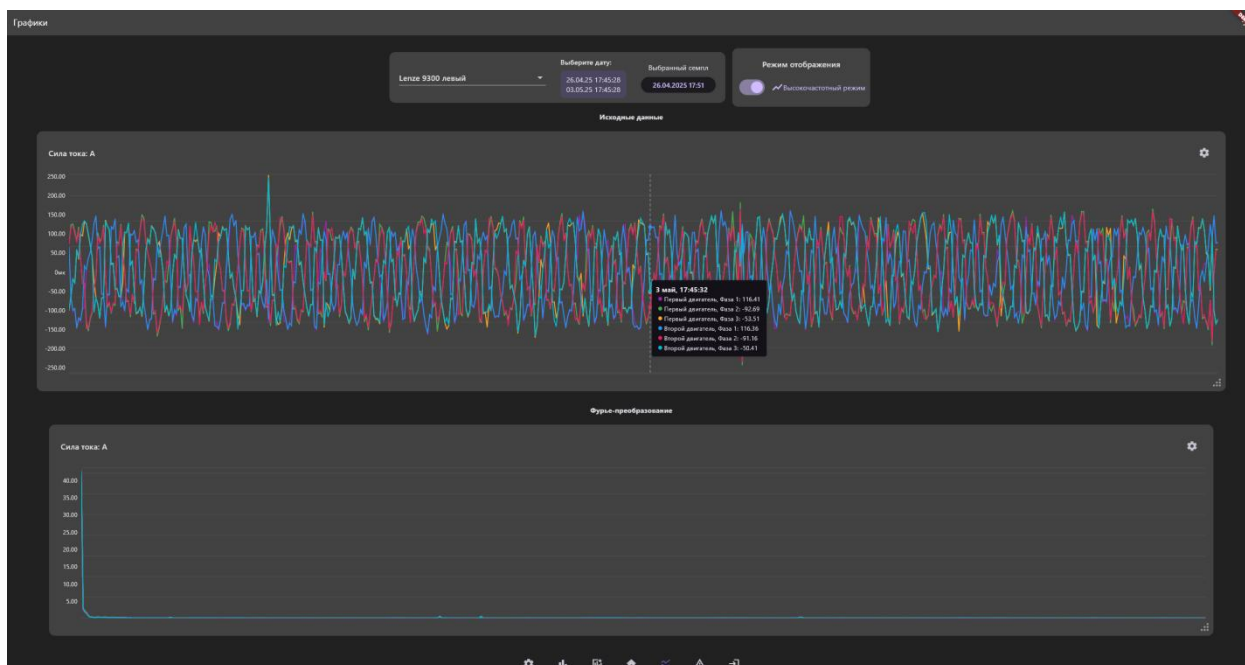


Рисунок 6 – Высокочастотный режим страницы графиков

Система визуализации данных включает ряд функциональных особенностей:

- Адаптивное изменение размеров карточек графиков путем движения мышью за угол в выделенной области
- Возможность масштабирования выделенного участка графика с возвратом к исходному масштабу по двойному нажатию (аналогично InfluxDb)
- Синхронизированная подсказка, отображающая значения на всех графиках в соответствующей временной точке
- Оптимизация отображения больших массивов данных через механизм децимации, при котором количество отображаемых точек соответствует количеству доступных пикселей по горизонтали. При этом алгоритм отбора точек сохраняет критические значения, выбирая точки с максимальным отклонением от соседних значений, что обеспечивает корректное отображение экстремумов и аномалий в данных

### **3.3.5 Страница событий**

На странице предупреждений (см. рис. 7) отображается список событий или превышений нижних или верхних пороговых значений низкочастотных данных. События группируются по времени и оборудованию, для удобства просмотра. На каждой карточке отображается детально параметры события, кнопка, чтобы отметить его как прочитанное и кнопка чтобы скопировать ссылку на событие. По нажатию на события открывается интерфейс, описанный ранее для страницы графиков. Графики запрашиваются в нужном диапазоне для выбранных параметров. Событие выделяется на всех графиках. После выбор превышения доступно добавления комментария к нему.

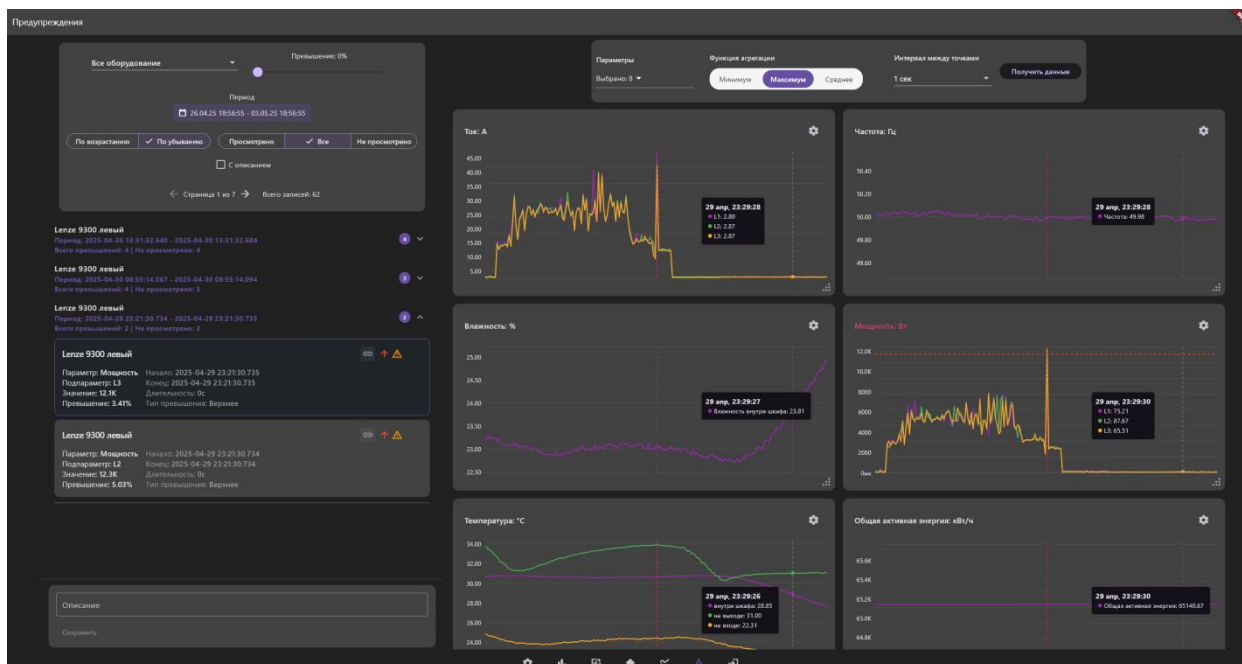


Рисунок 7 – Страница событий

### 3.3.6 Страница статистики

На странице статистике (см. рис. 8) подводятся итоги работы оборудования за выбранный период по каждому дню. Статистика представлена в виде календаря для удобства просмотра.

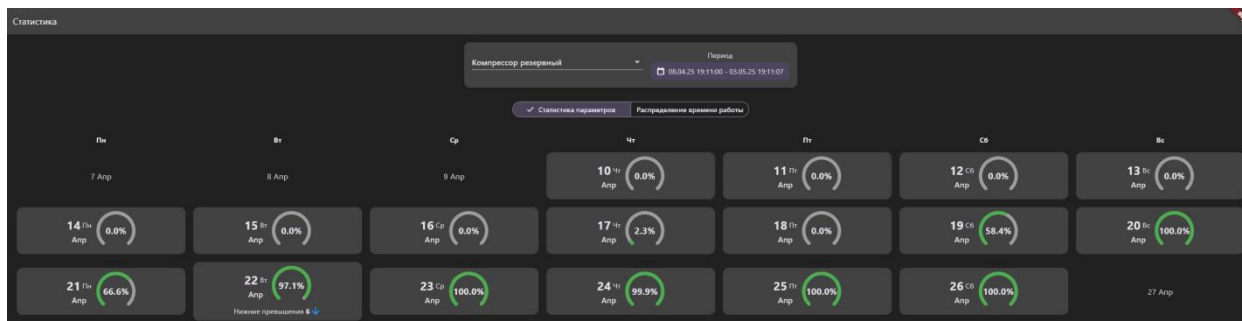


Рисунок 8 – Календарь статистики

Каждый день выделен в отдельный виджет, на котором отображается дата, процент работы за данный день и количество верхних и нижних превышений порогов. По нажатию на день открывается детальная статистика по каждому параметру и превышениям (см. рис. 9). Здесь представлены максимальные, минимальные, средние, медианные, значения 25 и 75 перцентилей в виде таблицы и в виде boxPlot так же на них для удобства

представлены пороговые значения. По событиям подводится следующая статистика: максимальный, минимальный и средний процент превышений и максимальная, минимальная и средняя длительности превышения.



Рисунок 9 – Статистика по параметрам

При переключении в раздел «Распределение процента работы» (см. рис. 10) можно построить тепловую карту работы оборудования за период, на ней отражена зависимость часа суток от дня недели, каждая ячейка градиентом отображает режимы работы, при наведении показываются точные значения.

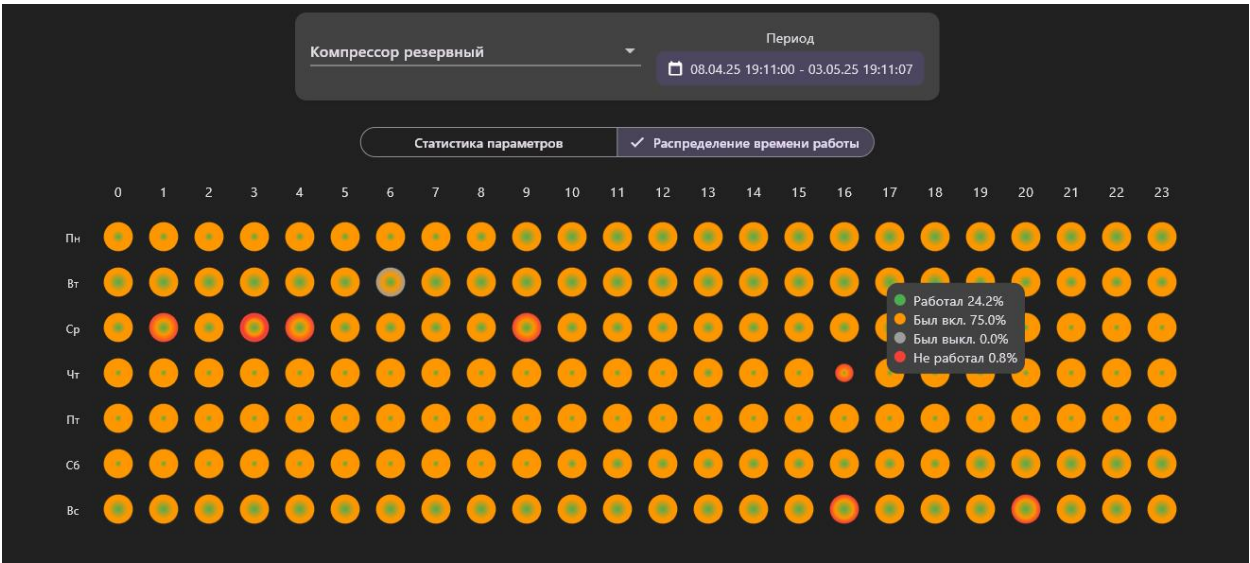


Рисунок 10 – Распределение процента работы

3.3.7 Страница аналитики

На странице аналитики (см. рис. 11) можно определить текущее состояние оборудования, его компонентов и тренд изменения.

Под сводной панелью расположены раскрывающиеся карточки для каждого компонента: на карточке отображаются процент состояния, число зафиксированных аномалий, отмеченных замечаний и рекомендаций, а при разворачивании — подробный список замечаний с пометками «auto» или

«human», перечень рекомендаций с возможностью добавить новую и текстовое уведомление о тенденции изменения состояния (например, «Состояние не критично снижается 1 неделю»).

Ниже карточек находится раздел с графиками, отображающими тренд изменения состояния оборудования за выбранный период.

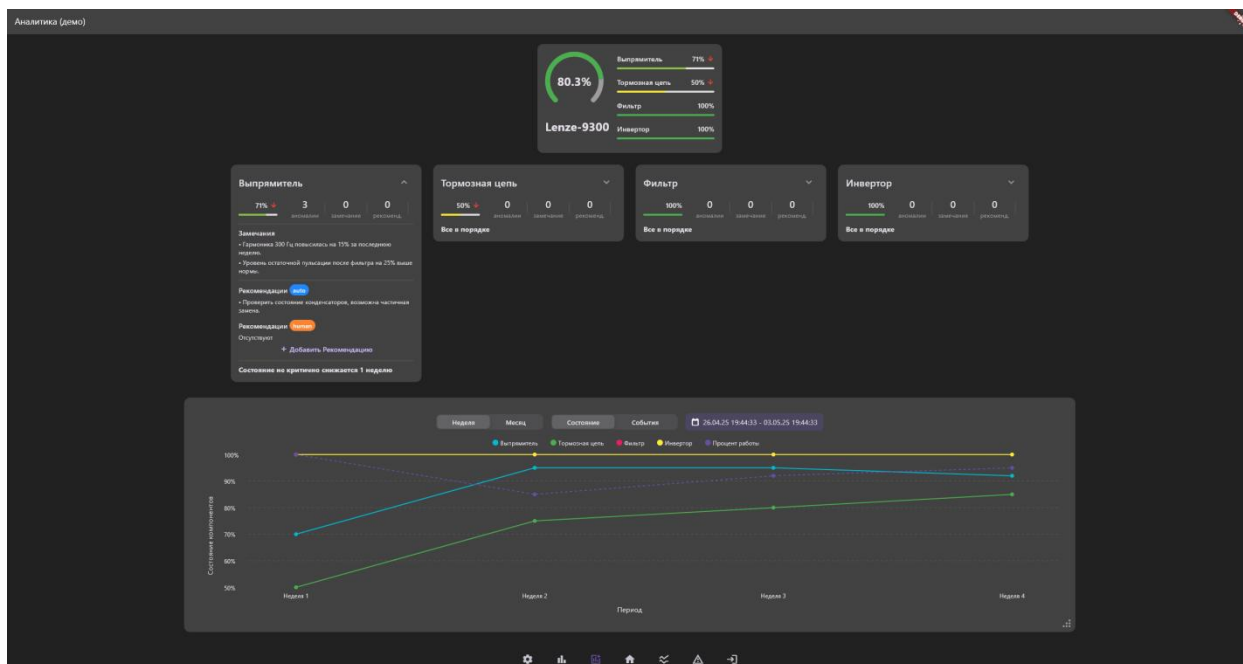


Рисунок 11 – Страница аналитики

### 3.3.8 Страница настроек

Страница настроек (см рис. 12) предоставляет пользователю возможность для персонализации работы с системой и управления данными. В части настройки уведомлений пользователь может указать предпочтительные контакты для получения оповещений, включая электронную почту и телеграмм аккаунт. Также доступна настройка типов уведомлений – пользователь выбирает, какие категории сообщений хочет получать: предупреждения или периодические отчеты. Важной функцией страницы настроек является возможность редактирования персональных данных.

Для удобства работы с интерфейсом система поддерживает импорт и экспорт визуальных настроек. Пользователь может сохранять свои предпочтения по оформлению – цветовые схемы, расположение элементов



интерфейса – на сервер, а затем загружать их на других устройствах или после переустановки системы. Это полезно для быстрого восстановления привычного рабочего пространства. Управление кешированными данными позволяет оптимизировать производительность системы. Пользователь может вручную очистить временные файлы, чтобы освободить место на устройстве.

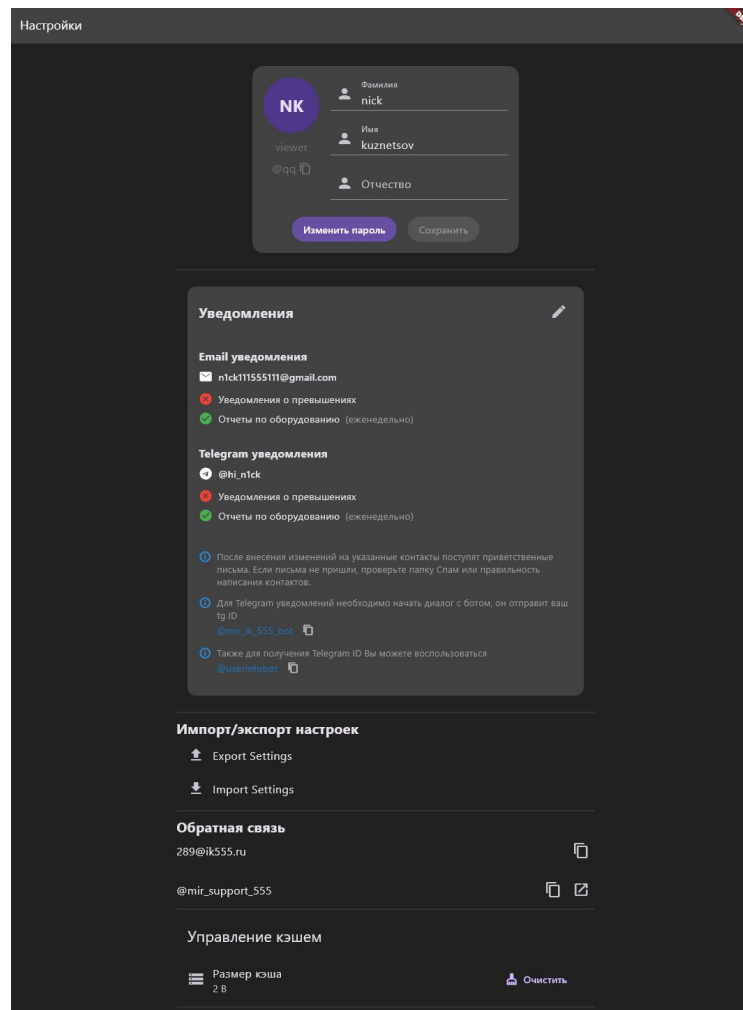


Рисунок 12 – Страница настроек

### 3.3.9 Локальная база данных Flutter Hive

Для хранения настроек системы использована Hive – легковесная NoSQL-база данных, оптимизированная для Flutter. Она обеспечивает высокую производительность и простоту использования.

Преимущества Hive:

- Быстрый доступ к данным (ключ-значение)

- Отсутствие зависимостей от нативных библиотек
- Поддержка сложных типов данных через адаптеры

Настройки пользователя (тема, параметры уведомлений, сохраненные фильтры) сохраняются локально и загружаются при запуске приложения. Nive обеспечивает стабильную работу даже на слабых устройствах.

### **3.4 Серверное приложение по обработке клиентских запросов**

Серверная часть отвечает на клиентские запросы приложения и осуществляет связь с базами данных и разделение запросов разных организаций. Для быстрого развертывания сервера был выбран фреймворк Flask за простоту и большого количества функционала «из-коробки»

#### **3.4.1 Модули серверного приложения**

Основной модуль `flask_server` приложения, который отвечает за настройку и запуск сервера. Он регистрирует все blueprints для разных функциональных частей приложения и настраивает middleware, такие как CORS и JWT. В зависимости от режима работы (DEBUG или PRODUCTION), сервер запускается либо с HTTP, либо с HTTPS.

`Warnings_flask` Отвечает за работу с предупреждениями о превышении параметров оборудования. Предоставляет эндпоинты для получения, фильтрации и обновления предупреждений. Каждое предупреждение содержит информацию о времени, оборудовании и проценте превышения. Реализованы функции сериализации для корректного преобразования данных MongoDB в JSON.

Модуль `statistics` предоставляет API для получения статистических данных о работе оборудования. Включает эндпоинты для получения box-plot графиков, статистики по предупреждениям и процентного соотношения работы оборудования. Модуль взаимодействует с сервисом InfluxDB через отдельный API для получения и обработки временных рядов данных.

Модуль `settings` управляет настройками пользователей и их аутентификацией. Позволяет экспортировать и импортировать пользовательские настройки, устанавливать параметры уведомлений, а также

управлять пользователями (добавление, удаление, изменение информации). Содержит функционал проверки административных прав пользователей.

Модуль `influx_parser_flask` обеспечивает взаимодействие с сервисом InfluxDB для получения и анализа временных рядов данных. Предоставляет эндпоинты для получения данных в реальном времени, процента работы оборудования и выявления аномальных значений.

Модуль `high_rate` отвечает за работу с высокочастотными данными. Предоставляет эндпоинты для получения конфигураций оборудования, выборки данных по временной метке и списка доступных выборок в указанном временном интервале. Поддерживает передачу бинарных файлов с сервера на клиент.

Модуль `equipment_flask` управляет информацией об оборудовании. Предоставляет эндпоинты для получения списка оборудования и обновления детальной информации о конкретном оборудовании. Поддерживает фильтрацию и сериализацию данных для передачи клиенту.

### **3.4.3 Особенности реализации**

Аутентификация и авторизация. Для обеспечения безопасности используется JWT-токены (JSON Web Tokens). Токены доступа выдаются на 30 минут, а токены обновления - на 30 дней. Большинство запросов защищены декораторами `token_required` или `jwt_required`. Эти декораторы написаны для удобного извлечения токенов из запросов, их проверки, а так же удобного извлечения данных из него. Такой подход помогает разделять запросы организаций.

Все модули включают обработку исключений для предотвращения сбоев сервера. В случае ошибок клиенту возвращается соответствующий код HTTP и сообщение об ошибке в формате JSON.

### **3.4.4 Взаимодействие с клиентской частью**

Серверная часть предоставляет REST API, который используется клиентской частью для получения и обновления данных. Все ответы

возвращаются в формате JSON. Для обеспечения безопасности все чувствительные операции требуют действительный JWT-токен.

### **3.5 Сервис уведомлений**

Сервис уведомлений представляет собой отдельное микросервисное приложение, обеспечивающее отправку различных типов уведомлений пользователям через электронную почту и мессенджер Telegram. Сервис также генерирует и рассылает периодические отчеты по расписанию.

#### **3.5.1 Архитектура сервиса уведомлений**

Сервис построен на базе Flask и включает несколько ключевых компонентов:

Основной модуль `main` содержит Flask-приложение с API-эндпоинтами для управления настройками уведомлений, обработки предупреждений и инициирования рассылки отчетов. Запускает и настраивает планировщик задач для регулярной рассылки отчетов.

Сервис уведомлений `notification_service` центральный компонент, который координирует работу различных обработчиков уведомлений (`email`, `Telegram`) и взаимодействует с базой данных для получения настроек пользователей. Поддерживает кеш настроек пользователей для оптимизации производительности.

Планировщик задач `scheduler_service` реализует механизм периодического выполнения задач на основе библиотеки `APScheduler`.

Обработчики уведомлений включают базовый класс `handler` и специализированные обработчики:

- `email_handler`: отправка уведомлений по электронной почте (см. рис. 13) с использованием SMTP, в виде форматированных HTML-писем с возможностью вложений (PDF-отчеты)
- `tg_handler`: отправка уведомлений через Telegram Bot API (см. рис. 14) с использованием библиотеки `Telethon`, в виде текстовые сообщения и документы (PDF-отчеты)

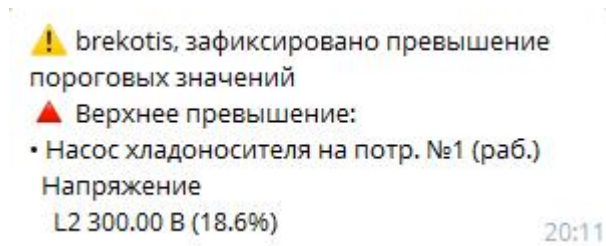


Рисунок 13 – Пример Telegram уведомлений

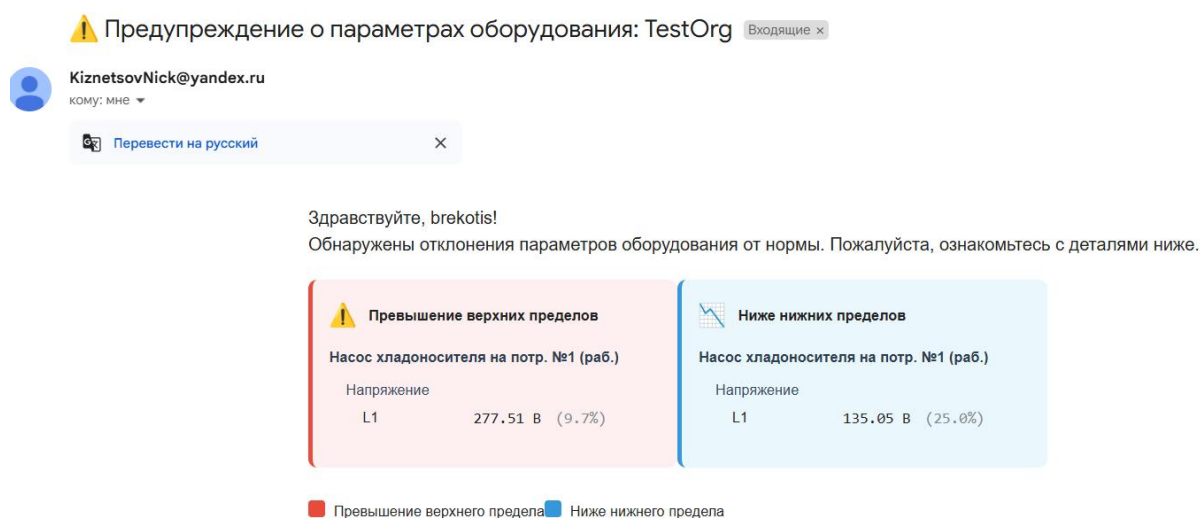


Рисунок 14 – Пример Email уведомлений

### 3.5.2 Функциональные возможности

Сервис поддерживает несколько типов уведомлений:

- Предупреждения о превышении параметров оборудования (верхние и нижние пороги)
- Предупреждения о событиях.
- Еженедельные отчеты о работе оборудования
- Ежемесячные отчеты о работе оборудования

Сервис поддерживает кеш настроек в памяти для ускорения обработки уведомлений. При изменении настроек кеш автоматически обновляется.

Для работы с Telegram API используется асинхронный подход с библиотекой Telethon, что позволяет эффективно обрабатывать большое количество уведомлений.

Для шаблонизации сообщений используется Jinja2 для создания HTML-шаблонов электронных писем и шаблонов текстовых сообщений для Telegram.

APScheduler настраивает и запускает еженедельную и ежемесячную рассылку отчетов по расписанию в соответствии с часовыми поясами организаций.

Сервис поддерживает работу с несколькими организациями, для каждой из которых ведется отдельный набор настроек пользователей.

### **3.5.3 Механизм работы**

Обработка предупреждений.

Сервисы обработки данных оборудования отправляют информацию о превышении параметров на эндпоинт ``/warning``. Сервис уведомлений определяет организацию и получает список пользователей из кеша. Для каждого пользователя проверяются настройки уведомлений. В зависимости от настроек, отправляются уведомления через соответствующие каналы.

Периодические отчеты. Планировщик запускает задачу отправки отчетов по расписанию. Сервис запрашивает данные для отчетов от сервиса отчетов. Для каждой организации и пользователя проверяются настройки уведомлений. Отчеты отправляются пользователям согласно их настройкам через выбранные каналы.

Управление настройками.

Клиентское приложение отправляет новые настройки на эндпоинт ``/set_up_settings``. Сервис сравнивает новые настройки с существующими. При добавлении нового канала связи пользователю отправляется приветственное сообщение. При изменении настроек пользователю отправляется уведомление об обновлении.

## **3.6 Сервис отчетов**

Сервис отчетов отвечает за агрегацию данных, аналитические вычисления и формирование PDF-отчетов о работе оборудования.

### **3.6.1 Архитектура сервиса отчетов**

Сервис реализован на языке Python с использованием Flask в качестве веб-фреймворка. В основе сервиса лежит обработчик запросов, который принимает POST-запросы от сервиса уведомлений и основного сервера,

запускает процессы генерации отчетов и возвращает сформированные PDF-файлы или json ответы.

Модуль содержит основной Flask-сервер с настроенными эндпоинтами. Он отвечает за обработку входящих запросов и координацию работы других модулей. Основным эндпоинтом является `"/reports"`, который принимает параметры для генерации отчета, включая идентификатор организации, временной диапазон и, при необходимости, список оборудования.

Модуль `warning_statistics` реализует функционал для агрегации и анализа данных о превышениях параметров оборудования. Он использует MongoDB для получения информации о зарегистрированных предупреждениях, а затем выполняет статистические расчеты, такие как нахождение максимального, минимального и среднего процента превышения, а также вычисление длительности отклонений от нормы.

Модуль `daily_service` отвечает за ежедневную обработку данных о работе оборудования. Реализует планировщик заданий на основе APScheduler, который запускает процессы сбора и обработки данных в определенное время (обычно в полночь) для каждой обслуживаемой организации с учетом их часовых поясов.

Модуль `metrics` предоставляет функционал для извлечения и агрегации метрик из InfluxDB. Он формирует и выполняет сложные запросы к временной базе данных для получения статистических показателей за указанный период, таких как минимальные, максимальные, средние значения, а также медианы и процентиля. Полученные результаты структурируются и сохраняются в MongoDB для дальнейшего использования при генерации отчетов.

Модуль `table` компонент для визуализации данных и формирования PDF-отчетов. Он использует библиотеку ReportLab для создания структурированных таблиц с данными об оборудовании, превышениях параметров и времени работы. Модуль включает функции для форматирования данных, создания стилей таблиц и визуального выделения

превышений. Результатом работы модуля является готовый PDF-документ с аналитическими данными.

### **3.6.2 Функциональные возможности**

Сервис отчетов создан для анализа данных о работе оборудования и формирования информативных отчетов. Основной задачей сервиса является агрегация данных из разных источников и представление их в удобном для восприятия формате. Сервис может формировать как периодические (ежедневные, еженедельные, ежемесячные) отчеты, так и отчеты по запросу за произвольный период времени.

В рамках ежедневной обработки данных сервис собирает информацию о работе оборудования, включая время работы и простоя, максимальные, минимальные и средние значения параметров, а также статистику по превышениям пороговых значений. Эти данные сохраняются в MongoDB для дальнейшего использования при формировании итоговых отчетов и запросах пользователей.

При формировании PDF-отчетов сервис создает наглядные таблицы с данными о работе оборудования за указанный период. В отчетах выделяются дни с превышениями пороговых значений параметров и дни с простоями оборудования. Для каждого типа оборудования формируется отдельная таблица, а в конце отчета добавляется сводная информация по всем превышениям.

### **3.6.4 Взаимодействие с другими компонентами системы**

Сервис отчетов интегрируется с другими компонентами системы мониторинга через API и общие базы данных. Основным потребителем услуг сервиса является сервис уведомлений, который запрашивает генерацию отчетов по расписанию или по специальным событиям. Сервис отчетов получает параметры запроса, формирует отчет и возвращает его в виде бинарных данных PDF-документа.

Для получения исходных данных сервис взаимодействует с MongoDB, где хранится информация об оборудовании, зарегистрированных



превышениях и результатах ежедневной обработки данных. Для работы с временными рядами данных сервис обращается к InfluxDB, используя токены доступа и параметры подключения, хранящиеся в конфигурации для каждой организации.

Взаимодействие с планировщиком задач APScheduler обеспечивает автоматическое выполнение регулярных процедур обработки данных. Планировщик запускает задачи ежедневной обработки данных в соответствии с расписанием, учитывая часовые пояса обслуживаемых организаций.

### **3.7 Сервис обработки низкочастотных данных**

Сервис обработки низкочастотных данных выполняет критически важную роль в системе мониторинга, отвечая за непрерывное наблюдение за параметрами оборудования и выявление отклонений от установленных пороговых значений. Данный компонент написан на языке Go, что обеспечивает высокую производительность и эффективное использование системных ресурсов при обработке потоков временных рядов.

#### **3.7.1 Архитектура сервиса**

Главным модулем является main, который координирует работу всей системы, инициализирует конфигурацию и запускает мониторинг для каждой организации в отдельной горутине. Это позволяет обслуживать несколько организаций параллельно, используя преимущества многопоточности языка Go.

Модуль models содержит определения всех структур данных, используемых в системе, включая модели оборудования, параметров, превышений и конфигурации.

В основе сервиса лежит компонент ThresholdChecker, который отвечает за периодический опрос источников данных и проверку значений параметров на соответствие пороговым значениям. Каждое оборудование группируется по периоду опроса, что позволяет оптимизировать нагрузку на базы данных и сеть, объединяя запросы к одинаковым временным интервалам.

Модуль `detail_check` реализует детальную проверку превышений после обнаружения экстремальных значений в указанном временном интервале. Он анализирует временные ряды точек для определения точного момента начала и окончания превышения, а также находит максимальное отклонение от нормы.

Компонент `notification_manager` отвечает за формирование и отправку уведомлений о зафиксированных превышениях. Он реализует механизм буферизации и пакетной отправки уведомлений, что позволяет снизить нагрузку на систему при множественных одновременных превышениях.

Для обработки исторических данных, например, после запуска сервиса или сбоя, используется модуль `historical`, который заполняет пробелы в данных путем обработки временных интервалов с момента последней проверки до текущего времени.

### **3.7.2 Принцип работы сервиса**

Работа сервиса начинается с инициализации и подключения к центральной базе данных MongoDB для получения списка обслуживаемых организаций. Для каждой организации создается отдельная конфигурация с параметрами подключения к соответствующим базам данных и запускается отдельный экземпляр мониторинга.

При запуске мониторинга для организации система сначала загружает информацию о всем оборудовании из MongoDB, группируя его по периодам опроса. Затем производится обработка исторических данных для заполнения возможных пробелов с момента последней проверки. После этого запускается цикл непрерывного мониторинга, где для каждой группы оборудования с одинаковым периодом опроса выполняется периодическая проверка параметров.

Процесс проверки параметров состоит из двух этапов. На первом этапе выполняется быстрая проверка экстремумов для определения потенциальных превышений. Если обнаружены экстремальные значения, превышающие пороговые, на втором этапе выполняется детальный анализ временного ряда

для точного определения начала, окончания и максимального значения превышения.

При обнаружении превышения создается запись в коллекции MongoDB и оно добавляется в список активных превышений. Система продолжает мониторинг этого параметра и обновляет запись, если превышение продолжается или значение отклонения увеличивается. Когда значение параметра возвращается в норму, превышение закрывается с указанием времени окончания.

Параллельно с мониторингом работает менеджер уведомлений, который собирает информацию о новых превышениях, группирует их и периодически отправляет пакеты уведомлений на сервис уведомлений для дальнейшей обработки и доставки пользователям. Уведомления формируются в структурированном формате JSON с разделением на верхние и нижние превышения, включая информацию о названии оборудования, параметрах, значениях и процентах отклонения.

### **3.8 Сервис обработки высокочастотных данных**

Сервис на основе математических моделей, диагностических правил и нейросети для выявления аномалий по высокочастотным данным определяет шаблонные случаи неисправности компонентов и также формирует записи в Mongo и отправляет данные в сервис уведомлений.

### **3.9 Панель администратора**

На странице администратора, можно посмотреть состояние контроллеров всех организаций и работу всех сервисов (см. рис. 15)

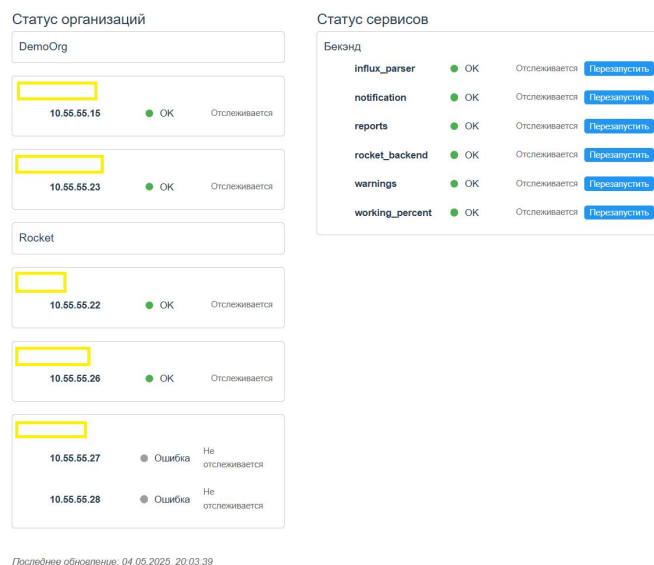


Рисунок 15 – Статусы организаций и сервисов

Так же можно просмотреть подробную информацию о пользователе: устройства, с которых пользователь заходил, история авторизаций, а так же узнать подробную статистику по количеству и времени, которые пользователь провел на той или иной странице (см. рис. 16 и рис. 17).

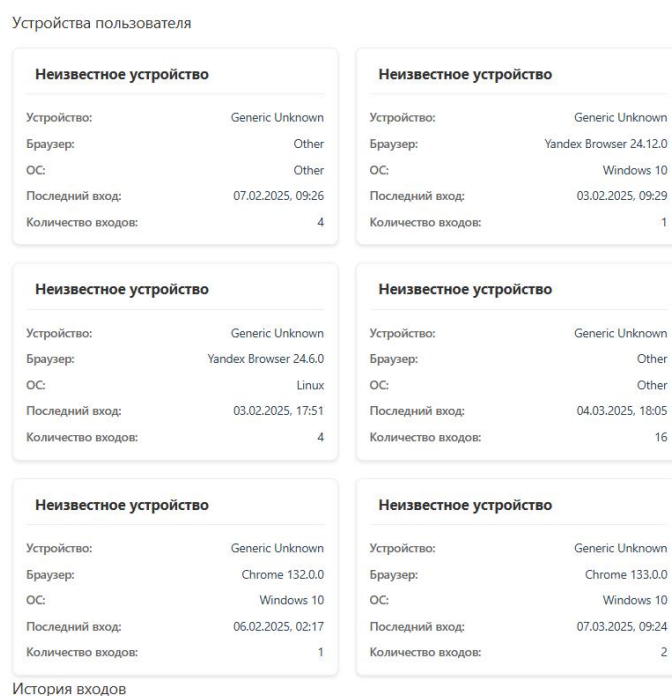


Рисунок 16 – Устройства пользователя



Рисунок 17 – Посещения пользователя

### 3.10 Развертывание системы

Система поддерживает развертывание как локальное, так и облачное. Архитектура развертывания обеспечивает простую установку и настройку системы в различных условиях эксплуатации (см. рис. 18).

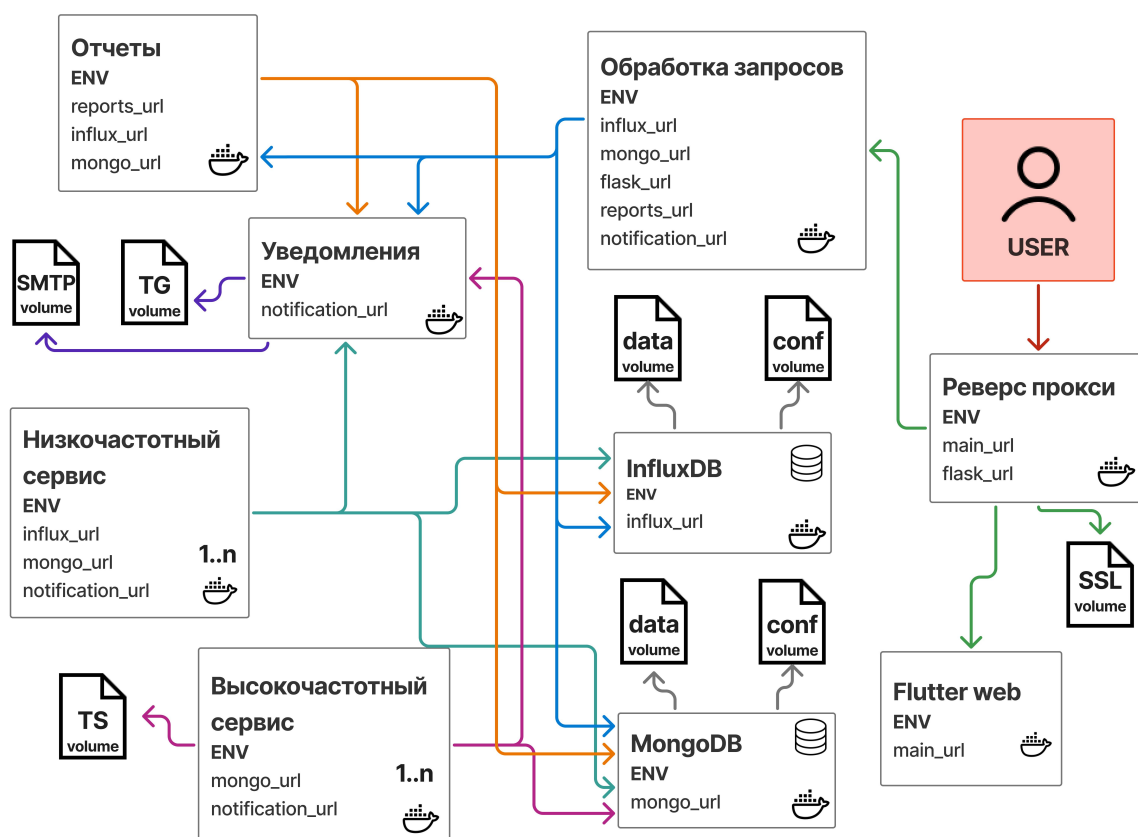


Рисунок 18 – Развертывание

Все компоненты системы собираются и запускается в своем изолированном контейнере Docker. Такой подход гарантирует стабильность работы каждого сервиса, изоляцию зависимостей и упрощает процесс обновления отдельных компонентов без влияния на работу остальной системы.

Управление контейнерами происходит через Docker Compose, что обеспечивает декларативное описание всей инфраструктуры системы в едином файле конфигурации. Файл `docker-compose.yml` содержит описание всех сервисов, их взаимосвязей, настроек сети и параметров запуска. Это позволяет воспроизводить идентичную среду выполнения на разных серверах и избегать проблем, связанных с различиями в конфигурации.

Все параметры запуска системы, включая порты сервисов, пути к volume-ам для хранения данных, расположение SSL-сертификатов и другие конфигурационные параметры, вынесены в ENV-переменные. Такой подход

позволяет адаптировать систему к требованиям конкретного окружения без изменения исходного кода или конфигурационных файлов. Для удобства настройки все переменные окружения могут быть определены в едином файле `.env`, который загружается Docker Compose при запуске системы.

Сетевая архитектура системы реализована с учетом требований безопасности. Все контейнеры подключены к изолированной виртуальной сети Docker (`app-network`), что обеспечивает защищенное взаимодействие между компонентами. Внешний доступ к сервисам системы предоставляется через обратный прокси-сервер NGINX, который обрабатывает входящие пользовательские запросы, обеспечивает SSL-терминацию и маршрутизацию запросов к соответствующим внутренним сервисам.

Для обеспечения сохранности данных при перезапуске или обновлении системы используются Docker volumes, которые обеспечивают хранение информации вне контейнеров. В системе настроены отдельные volumes для баз данных MongoDB и InfluxDB, конфигурационных файлов, SSL-сертификатов и других важных данных.

Процесс развертывания системы максимально упрощен и требует минимального вмешательства администратора. После настройки переменных окружения и подготовки необходимых сертификатов достаточно выполнить команду `docker-compose up -d`, и система будет автоматически собрана и запущена. Это позволяет разворачивать систему на любом сервере, имеющей установленный Docker, независимо от её операционной системы или аппаратной конфигурации. Такой подход к развертыванию обеспечивает высокую портативность системы, возможность быстрого масштабирования и простоту администрирования, что делает систему мониторинга оборудования подходящей как для небольших локальных инсталляций, так и для крупных промышленных развертываний в облачной инфраструктуре.

#### 4 ХАРАКТЕРИСТИКИ РЕШЕНИЯ

Для оценки результатов работы было выполнено тестирование скорости загрузки страниц с получением данных, возможность работы с необходимой частой данных и доступность на основных платформах web, windows, linux, android и ios.

Характеристики сервера:

- RAM: 10gb DDR4 2400 МГц
- CPU Intel(R) Xeon(R) Gold 2 ядра
- OS Debian GNU/Linux 11 (bullseye)

Характеристики устройств:

Desktop (Windows 11 / Ubuntu 20.04):

- CPU: AMD Ryzen 5 5600X (6 ядер/12 потоков, 3.7-4.6 ГГц)
- RAM: 16GB DDR4 3200 МГц
- GPU: NVIDIA RTX 3060 12GB
- Браузер: Google Chrome 119

iPhone 14:

- CPU: Apple A15 Bionic (6 ядер)
- RAM: 6GB LPDDR5
- GPU: 5-core Apple GPU
- OS: iOS 17

Google Pixel 9:

- CPU: Google Tensor G4
- RAM: 12GB LPDDR5X
- GPU: ARM Mali-G715
- OS: Android 15

В ходе тестирования были выполнены следующие сценарии использования (время указано в мс) (см. таблица 2). В измерение входит время от нажатию на кнопку перехода на страницу, до полной загрузки страницы (включены все вспомогательные запросы).

1. Главная страница



2. Низкочастотные графики по всем параметрам и оборудованию: 10 единиц оборудование, 20 параметров, 1Гц, последние 24 часа - суммарно 17 миллионов точек.
3. Высоко частотные графики по всем параметрам и оборудованию: 10 единиц оборудование, 9 параметров, 300кГц, 1 минута данных - суммарно 27 миллионов точек.
4. Получение одной страницы событий: 100 событий.
5. Получение статистики за месяц: одна единица оборудования.

Таблица 2 – Тестирование

Система	Тесты (время в мс.)				
	1	2	3	4	5
Windows	601	2695	1784	424	408
Linux	580	2250	1690	405	390
Web	690	3104	1911	473	440
android	710	3200	1950	485	455
ios	695	3153	1926	481	450

В данный момент система обслуживает 7 организаций и суммарно 100 единиц оборудование.

По результатам тестирования программное обеспечение успешно выполнило все основные сценарии использования и требования.

## **5 Безопасность жизнедеятельности**

### **5.1 Основные положения**

ВКР посвящена разработке программного комплекса для мониторинга частотных преобразователей и предполагает его использование в области промышленного производства. Основным пользователем разработанного комплекса будет инженер в офисе на производстве. Программный комплекс предполагается применять на персональных компьютерах, ноутбуках и мобильных устройствах (смартфонах, планшетах), с возможностью подключения к интернету.

### **5.2 Анализ факторов воздействия**

При создании программного комплекса для мониторинга частотных преобразователей необходимо учитывать различные опасные и вредные производственные факторы (ОиВПФ), определенные ГОСТ 12.0.003-2015 [14].

В процессе разработки комплекса разработчики, включая автора ВКР, и во время эксплуатации пользователи подвергаются следующим воздействиям:

Физические факторы:

- Недостаточная освещенность рабочей зоны при длительной работе за монитором;
- Повышенный уровень электромагнитных излучений от компьютерного оборудования;
- Отклонения показателей микроклимата (температура, влажность, скорость движения воздуха) в рабочем помещении;
- Повышенный уровень шума от системных блоков, кондиционеров и другого оборудования.

Психофизиологические факторы:

- Статические физические нагрузки, связанные с длительным пребыванием в сидячем положении;
- Перенапряжение зрительных анализаторов при работе с большими объемами данных и программным кодом;

Произведена оценка разрабатываемого ПО (см. таблица 3) и оценка ПО, которое использовалось при разработке (см. таблица 4), в соответствии с ГОСТРИСО/МЭК ТО 12182-2002 [15].

Таблица 3 – Оценка разрабатываемого ПО

Программное обеспечение	Программный комплекс для мониторинга частотных преобразователей
Вид	Класс
Функция ПС	Мониторинг данных частотных преобразователей
Режим эксплуатации	Автоматическая обработка данных в режиме реального времени
Прикладная область информационной системы	Оборудование и аппаратура управления процессом (предназначена для промышленного применения в производственных системах)
Масштаб ПС	Средний
Представление данных	Реляционный (для хранения исторических данных), объектный (для обработки данных в реальном времени) и форматированный файл (для экспорта отчетов)
Критичность ПС	Организационная безопасность и частная собственность (некорректная работа может привести к простоям оборудования и экономическим потерям)
Класс пользователя	Специалист (инженер производства должен иметь опыт работы с промышленным оборудованием и понимание принципов работы частотных преобразователей)
Требуемые рабочие характеристики	Емкость — высокая (система должна хранить большие объемы исторических данных); время отклика — быстрое (оперативное отображение изменений параметров и аварийных ситуаций); производительность — средняя (одновременный мониторинг нескольких десятков устройств)

Стабильность ПС	Дискретное внесение изменений (обновления при появлении новых типов частотных преобразователей и добавлении новых функций)
Требование защиты	Защита от несанкционированного доступа: сильная; Контрольный след: сильная; Защита программ и данных: сильная
Требование надежности	Завершенность: высокая; отказоустойчивость: высокая; восстанавливаемость: высокая
Вычислительная система и среда	Серверное приложение с клиентской частью для персональных компьютеров и мобильных устройств (операционные системы Windows, Linux, Android, iOS)
Требования к вычислительным ресурсам	Оперативная память — не менее 4 ГБ для серверной части, 2 ГБ для клиентской. Внешняя память — не менее 100 ГБ для серверной базы данных, 500 МБ для клиентского приложения. Требования к локальной вычислительной сети — стабильное соединение с пропускной способностью не менее 100 Мбит/с
Готовность программного продукта	На стадии разработки, запатентованное
Использование программных данных	Для множества пользователей с конкурентным взаимным исключением (несколько инженеров могут одновременно мониторить состояние оборудования)
Исходный язык	Dart, Python, Go

Таблица 4 – Оценка ПО, использованного при разработке

Программное обеспечение	Android studio, Visual Studio Code, PyCharm
Вид	Класс

Функция ПС	Интегрированная среда разработки
Режим эксплуатации	Интерактивная обработка
Прикладная область информационной системы	Прикладная область информационной системы

### **5.3 Снижение воздействия вредных факторов**

Для обеспечения безопасных условий труда при разработке программного комплекса для мониторинга частотных преобразователей необходимо реализовать комплекс мероприятий, учитывающий нормативные требования и направленный на минимизацию воздействия выявленных опасных и вредных факторов.

#### **5.3.1 Требования к освещению рабочего места**

Требования к освещению рабочего места Согласно СанПиН 1.2.3685-21 [16] рабочее место должно иметь при искусственном освещении освещенность рабочей поверхности 500 лк всего из них 300 лк от общего освещения коэффициент пульсации освещенности не более 5% при естественном освещении коэффициент естественной освещенности КЕО при верхнем или комбинированном освещении не менее 3% КЕО при боковом освещении не менее 1% рабочее место должно быть оборудовано дополнительными источниками света для локального освещения документов с возможностью регулировки яркости и направления света для предотвращения бликов на экране монитора.

#### **5.3.2 Защита от электромагнитных излучений**

Защита от электромагнитных излучений В соответствии с СанПиН 1.2.3685-21 [16] уровень электромагнитного поля на рабочем месте не должен превышать по электрической составляющей в диапазоне частот 5 Гц - 2 кГц 25 В/м в диапазоне частот 2 кГц - 400 кГц 2,5 В/м по магнитной составляющей в диапазоне частот 5 Гц - 2 кГц 250 нТл в диапазоне частот 2 кГц - 400 кГц 25 нТл.

### **5.3.3 Обеспечение комфортного микроклимата**

Обеспечение комфортного микроклимата Согласно ГОСТ 30494-2011 [17] в помещении где проводится разработка программного обеспечения должны поддерживаться следующие параметры микроклимата температура воздуха в холодный период года допустимая 19-23°C в теплый период года допустимая 20-28°C относительная влажность в холодное время года допустимая не более 60% в теплое время года допустимая не более 65% скорость движения воздуха в холодный период года не более 0,1 м/с в теплый период года не более 0,15 м/с для поддержания требуемых параметров помещение должно быть оборудовано системами отопления кондиционирования и вентиляции.

### **5.3.4 Снижение шумового воздействия**

Снижение шумового воздействия Согласно СанПиН 1.2.3685-21 [16] предельно допустимый уровень шума на рабочем месте при выполнении высококонцентрированных работ не должен превышать 50 дБА.

### **5.3.5 Снижение психофизиологических нагрузок**

Снижение психофизиологических нагрузок Для минимизации негативного воздействия статических нагрузок при длительной работе за компьютером применяются требования установленные в ГОСТ Р ИСО 9241-5-2009 [18] стандарт регламентирует организацию рабочего пространства с учетом биомеханики человеческого тела рекомендуется обеспечивать правильную позу работающего которая характеризуется следующими параметрами бедра должны располагаться параллельно поверхности пола спина плечевой пояс и ноги следует поддерживать в вертикальном положении запястья должны находиться в нейтральном положении без чрезмерных изгибов направление взгляда необходимо поддерживать в диапазоне между горизонтальной линией и углом 60° к ней эргономическая рабочая мебель должна соответствовать нескольким ключевым критериям адаптивность к различным пользователям функциональное соответствие

выполняемым задачам конструктивная возможность для периодического изменения рабочей позы и доступность технического обслуживания дополнительной защитой от негативного влияния статических нагрузок служит соблюдение режима труда и отдыха согласно положениям статей 108 и 109 Трудового кодекса РФ [19] предусматривающих обязательные перерывы в работе рекомендуется также проведение комплекса производственной гимнастики для снятия мышечного напряжения через каждые 45-60 минут непрерывной работы.

#### **5.4 Выводы**

Анализ условий труда при разработке программного модуля для корпоративной базы данных выявил наличие потенциально вредных производственных факторов физической и психофизиологической природы по каждому фактору определены нормативные показатели и разработаны меры по минимизации их воздействия основные параметры рабочей среды включают уровень освещенности составляющий 480 лк при рекомендуемых 500 лк с преобладанием комбинированного освещения температурный режим поддерживается на уровне 22°C что соответствует нормативному диапазону 20-28°C для теплого периода года показатели шумового фона не превышают 42 дБА что создает комфортные условия для концентрации внимания особого внимания требует организация рабочего места так как используемое кресло не имеет достаточной поясничной поддержки и ограниченных возможностей регулировки что может привести к дискомфорту при длительной работе.

## **ЗАКЛЮЧЕНИЕ**

В рамках данной работы было успешно разработано решение для мониторинга работы частотных преобразователей. Разработанное решение включает кроссплатформенное клиентское приложения, сервисы обработки временных рядов, сервер обработки клиентских запросов, что позволяет решать задачи мониторинга частотных преобразователей в промышленных условиях. Система обладает следующими преимуществами по сравнению с рассмотренными аналогами. Высокая частота опроса данных: Разработанное решение обеспечивает частоту опроса 300 кГц, что позволяет проводить точный анализ формы сигнала переменного тока частотой 50 Гц. Это существенно превосходит возможности большинства рассмотренных аналогов. Расширяемый набор параметров мониторинга: разработанная система позволяет добавлять специфические параметры, необходимые для конкретного производства. Гибкость развертывания: Разработанное решение поддерживает как облачное, так и локальное развертывание, что позволяет учитывать требования информационной безопасности предприятий. Интеграция с машинным обучением: Система включает модели машинного обучения для обнаружения аномалий. Это расширяет функциональность по сравнению с традиционными системами мониторинга.



## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Pouliezios A. D., Stavrakakis G. S. Real time fault monitoring of industrial processes. – Springer Science & Business Media, 2013. – Т. 12
2. ABB. "ABB Ability™ Digital Powertrain Technical Guide." ABB Group, 2023.
3. Siemens. "MindSphere: The Industrial IoT as a Service Solution." Siemens Digital Industries Software, 2024
4. Schneider Electric. "EcoStruxure Architecture and Platform Strategy." Schneider Electric, 2023
5. Emerson. "Plantweb Optics Asset Performance Platform." Emerson Electric Co., 2024
6. Rockwell Automation. "FactoryTalk® Industrial Automation Software Suite." Rockwell Automation, Inc., 2024.
7. Dart. "Dart Programming Language Documentation." Dart.dev, 2024.
8. Flutter. "Flutter Framework Documentation." Flutter.dev, 2024.
9. Go. "The Go Programming Language Documentation." Golang.org, 2024.
10. Python Software Foundation. "Python Language Reference." Python.org, 2024.
11. Pallets Projects. "Flask Web Framework Documentation." Flask.palletsprojects.com, 2024.
12. MongoDB, Inc. "MongoDB Documentation." MongoDB.com, 2024.
13. InfluxData, Inc. "InfluxDB Time Series Database Documentation." InfluxData.com, 2024.
14. ГОСТ 12.0.003-2015 «Система стандартов безопасности труда. Опасные и вредные производственные факторы. Классификация». [Электронный ресурс]. URL: <https://docs.cntd.ru/document/1200136071> (дата обращения: 23.04.2025).
15. ГОСТ Р ИСО/МЭК ТО 12182-2002. Менеджмент риска. Информационная технология. Классификация программных средств. – Введ. 2003-07-01. – М.: ГОССТАНДАРТ РОССИИ, 2002. – 16 с.

16. СанПиН 1.2.3685–21. Гигиенические нормативы и требования к обеспечению безопасности и (или) безвредности для человека факторов среды обитания. – М.: ИИЦ Минздрава России, 2021. – 975 с.
17. ГОСТ 30494-2011. Здания жилые и общественные. Параметры микроклимата в помещениях. – Введ. 2013-01-01. – М.: Стандартинформ РОССИИ, 2019. – 18 с
18. ГОСТ Р ИСО 9241-5-2009. Эргономические требования к проведению офисных работ с использованием видеодисплейных терминалов (VDT). Часть 5. Требования к расположению рабочей станции и осанке оператора. – Введ. 2010-12-01. – М.: Стандартинформ РОССИИ, 2019. – 28 с.
19. Трудовой кодекс Российской Федерации. Официальный текст. – М.: Омега-Л, 2022. – 285 с

## **ПРИЛОЖЕНИЕ А. НАПРАВЛЕНИЕ РАЗВИТИЯ**

Использование методов машинного обучения для анализа исторических данных о работе оборудования позволит прогнозировать потенциальные неисправности. Это потребует разработки специализированных моделей, учитывающих специфику промышленного оборудования, а также создания механизмов автоматической валидации и переобучения моделей при появлении новых паттернов поведения оборудования.

### **Расширение поддержки типов промышленного оборудования**

1. Добавление поддержки разных типов промышленной электроники с учетом их специфики.

2. Масштабирование системы - увеличение количества поддерживаемого оборудования потребует оптимизации алгоритмов обработки и хранения данных для обеспечения производительности при росте объема информации. Следует исследовать возможности применения распределенных вычислений и новых методов компрессии временных рядов.

### **Автоматизация технического обслуживания**

1. Разработка системы рекомендаций - создание алгоритмов, способных на основе накопленных данных формировать конкретные рекомендации по обслуживанию оборудования. Система должна учитывать как текущее состояние оборудования, так и историю его эксплуатации и обслуживания.

2. Оптимизация планирования - внедрение механизмов автоматического планирования технического обслуживания с учетом прогнозируемого состояния оборудования, его загрузки и доступности обслуживающего персонала. Это позволит минимизировать простои и оптимизировать расходы на обслуживание.

## ПРИЛОЖЕНИЕ Б. КОНФЕРЕНЦИЯ «НАУКА НАСТОЯЩЕГО И БУДУЩЕГО»

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ  
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
«САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ ЭЛЕКТРОТЕХНИЧЕСКИЙ  
УНИВЕРСИТЕТ «ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)»  
(СПбГЭТУ «ЛЭТИ»)

Научно-практическая конференция с международным участием  
«НАУКА НАСТОЯЩЕГО И БУДУЩЕГО»  
для студентов, аспирантов и молодых ученых

### СЕРТИФИКАТ УЧАСТНИКА

по направлению конференции

Программная инженерия и автономные интеллектуальные системы

УЧАСТНИК

**Кузнецов Николай Александрович**

доклад

Разработка программного комплекса для мониторинга частотных  
преобразователей

Ответственный секретарь  
конференции



И.В. Матвеева

2025



НАУКА НАСТОЯЩЕГО  
И БУДУЩЕГО  
Научно-практическая конференция