*Изм.*

*Лист*

*№ докум.*

*Подп.*

*Дата*

*Лист*

*3*

КП.09.02.07.192.10.ПЗ

*Разраб.*

*Пров.*

*Рамазанова Ю.А.*

ншина

*Разработка модуля информационной системы конфигурирования объектов телеметрии*

*Лит.*

*Листов*

*60*

*ГАПОУ «Альметьевский политехнический техникум»*

**Содержание**

[Введение 4](#_Toc131791867)

[1 Проектирование информационной системы 6](#_Toc131791868)

[1.1 Анализ предметной области и построение бизнес-процессов 6](#_Toc131791869)

[1.2 Проектирование диаграммы прецедентов 22](#_Toc131791870)

[1.3 Проектирование диаграммы классов 23](#_Toc131791871)

[1.4 Проектирование схемы данных и словаря данных 24](#_Toc131791872)

[1.5 Проектирований интерфейса информационной системы 29](#_Toc131791873)

[1.6 Разработка алгоритма информационной системы 31](#_Toc131791874)

[1.7 Выбор программных и технических средств для разработки информационной системы 32](#_Toc131791875)

[2 Разработка информационной системы 34](#_Toc131791876)

[2.1 Программирование на стороне клиента 34](#_Toc131791877)

[2.2 Программирование на стороне сервера 42](#_Toc131791878)

[2.3 Разработка подсистемы безопасности 45](#_Toc131791879)

[3 Тестирование информационной системы 47](#_Toc131791880)

[4 Оценивание информационной системы для выявления возможности ее модернизации 50](#_Toc131791881)

[Заключение 53](#_Toc131791882)

[Список используемых источников 54](#_Toc131791883)

[Приложение А – Техническое задание на разработку информационной системы 55](#_Toc131791884)

[Приложение Б – Инструкция пользователя и программиста 58](#_Toc131791885)

[Приложение В – Электронный формат пояснительной записки и информационной системы 60](#_Toc131791886)

*Камалов А.Р.*

ншина

*Камалов А.Р.*

ншина

# Введение

Темой курсового проекта является «Разработка модуля информационной системы конфигурирования объектов телеметрии»

Курсовой проект является составляющей частью ПМ 05 «Проектирование и разработка информационных систем».

Цель курсового проекта – разработать веб-приложение по управлению телеметрическими системами.

Для достижения поставленной цели необходимо реализовать следующие задачи:

* сбор исходных данных об объектах телеметрии для разработки проектной документации на информационную систему;
* разработка проектной документации на разработку информационной системы в соответствии с требованиями заказчика;
* разработка подсистемы безопасности информационной системы в соответствии с техническим заданием;
* разработка модулей информационной системы в соответствии с техническим заданием;
* тестирование информационной системы на этапе опытной эксплуатации с фиксацией выявленных ошибок кодирования в разрабатываемых модулях информационной системы;
* разработка технической документации на эксплуатацию информационной системы;
* оценка информационной системы для выявления возможности ее модернизации.

Информационные системы по контролю данных телеметрических систем стали необходимостью на предприятиях для обеспечения получения, преобразования, передачу по каналу связи, приём, обработку и регистрацию измерительной информации и информации о различных событиях с целью контроля на расстоянии различных объектов и процессов. Диспетчеру предприятия предоставляется возможность просматривать необходимую информацию конфигурации объекта, событий и значений-показателей, архива значений за определенный период, графиков, построенных на базе выбранных свойств на заданном промежутке, настройки схемы подключений и событий объектов, ввод конфигурации свойств.

Веб-приложение разрабатывается для интеграции его как веб-модуль в программно-технический комплекс «КРОН-ТМ», внедряемой компанией «ООО ЦИТ «Бисофт». Данный программно-технический комплекс выполняет ряд функций, без которых сегодня невозможен процесс добычи и подготовки нефти:

* сбор данных с объектов
* обработка и систематизация накопленной информации, её анализ и отображение в наглядном виде
* ведение истории хода технологического процесса и событий системы
* формирование и выдача аварийных сигналов
* диагностика оборудования, состояния связи с объектами.
* передача управляющих воздействий оператора на технологический процесс
* контроль корректности получаемой информации.

# 1 Проектирование информационной системы

# 1.1 Анализ предметной области и построение бизнес-процессов

В функции приложения «КРОН-ТМ» входит отображение информации с OPC-сервера:

* накопленных значений с БЭСКЖ
* мгновенных дебитов скважин
* давления на устьях и т. д.

Получаемые значения записываются в базу данных TM\_OPC СУБД MicrosoftSQLServer с последующей интеграцией с корпоративной информационной системой «КРОН».

Дополнительные возможности приложения "КРОН-ТМ":

* предоставляется справочная информация из КИС «КРОН» по режиму скважин и параметрам добываемой жидкости
* производится расчёт количества добытой жидкости за предыдущие сутки и текущее отклонение от режима скважины.

Функциональность предусматривает регистрацию фактов потери и восстановления связи с объектом, наличия нулевых замеров, а также квитирования данных событий диспетчером, для оценки оперативности реагирования персонала.

Функционал программы предоставляет доступ к архиву замеров за выбранный интервал времени, для анализа динамики изменений значений.

Телеметрия— область науки и техники, занимающаяся вопросами разработки и эксплуатации телеметрических систем — комплекса автоматизированных средств, обеспечивающих получение, преобразование, передачу по каналу связи, приём, обработку и регистрацию измерительной информации и информации о различных событиях с целью контроля на расстоянии различных объектов и процессов.

Сущность телеметрии заключается в преобразовании измеряемой величины (или величин) в информационный сигнал, пригодный для передачи по каналу связи; передачи и приёма информации; декодировании, преобразовании и регистрации на приёмной стороне телеметрической информации с объекта.

Это необходимая дисциплина в сфере добывающих компаний, стремящихся создать передовые методы отслеживания состояния технологического устройства, режима работы, конфигурации свойств, событий определенного объекта.

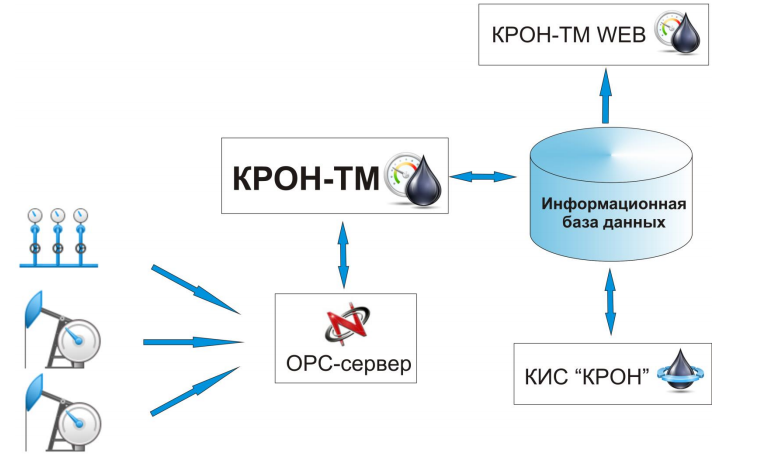


Рисунок 1.1 – Структурная схема

В качестве основных элементов системы телемеханики выступают:

1. OPC-сервер – производит опрос полевых контроллеров, установленных в шкафах КИП на объектах. Опрос производится по протоколу ModbusTCP посредством беспроводных средств коммуникации.

2. Клиентское приложение «КРОН-ТМ» - осуществляет опрос тегов OPCсервера и передачу информации в базу данных TM\_OPC СУБД MSServer, а также предоставляет пользователю обработанную и структурированную информацию из базы данных.

3. База данных – место хранения архива замеров по скважинам, а также информации по всем замеряемым объектам. В базе данных производится обработка информации и подготовка для отображения в интерфейсах «КРОНТМ» и «КРОН-ТМWEB». Кроме этого в базу данных регулярно подкачивается информация, касающаяся режима по скважинам.

4. «КРОН-ТМWEB» – является упрощённой версией клиента «КРОН-ТМ», доступной через web-браузер с любого компьютера в локальной сети или сети Интернет. Функционал данной системы предусматривает отображение информации формы мониторинга без возможности изменения настроек, конфигурирования объектов и квитирования событий.

5. КИС «КРОН» - информационная система предприятия, интегрированная с базой данных КРОН-ТМ. Замеры из базы данных системы телемеханики регулярно, с заданной периодичностью, подкачиваются в систему "КРОН".

Также КИС «КРОН» является источником информации по режимам скважин.

В фонд телеметрических систем входят разные объекты:

* Блок-гребенка (БГ)
* Групповая замерная установка (ГЗУ)
* Мультифазный насос (МФН)
* Вакуумная установка (УВС)
* Скважины
* Длинноходовая глубиннонасосная установка (ДГНУ)

БГ предназначен для распределения потоков подаваемой блочными кустовыми насосными станциями пластовой или чистой воды под высоким давлением в нагнетательные скважины с целью поддержания пластового давления.

Для БГ обычно настраиваются следующие свойства:

* Количество жидкости по фланцам
* Давление линейное по БГ
* Состояние связи

ГЗУ предназначена для обеспечения контроля за количеством жидкости, добываемой из нефтяных скважин, и попутного газа с выдачей результата в блок управления.

Для ГЗУ обычно настраиваются следующие свойства:

* Количество жидкости по фланцам
* Текущий фланец (переключатель скважин многоходный (ПСМ))
* Состояние связи

МФН диспергирует и прокачивает газожидкостную смесь, стабилизируя работу погружного электроцентробежного насоса и обеспечивая возможность добычи нефти в скважинах с содержанием свободного газа до 75 % на входе насоса.

Для МФН обычно настраиваются следующие типы свойств:

* Давление линейное входное и выходное
* Состояние насоса
* Авария преобразователя частот (ПЧ)
* Готовность к работе
* Выходная частота ПЧ
* Сила тока
* Работа от сети
* Работа от ПЧ
* Состояние связи

УВС – это система на базе насосов, предназначенная для создания и поддержания уровня технического вакуума посредством откачки воздушной или газовой смеси. Подобные установки работают как в стационарных, так и в мобильных производственных системах.

* Для УВС обычно настраиваются следующие типы свойств:
* Положение клапана
* Состояние компрессора
* Давление линейное (выходной/входное/на линии)
* Состояние УВС
* Режим управления УВС
* Состояние связи

Скважина — горная выработка круглого сечения, пробуренная с поверхности земли или с подземной выработки без доступа человека к забою под любым углом к горизонту, диаметр которой намного меньше её глубины. Бурение скважин проводят с помощью специального бурового оборудования.

* Для Скважины обычно настраиваются следующие типы свойств:
* Давление линейное (выходной/входное/на линии)
* Количество жидкости
* Состояние связи

ДГНУ предназначена для подъема нефти с забоя скважин.

Для ДГНУ настраиваются следующие типы свойств:

* Положение (координата) плунжера
* Давление линейное на выкиде
* Мощность двигателя ДГНУ
* Количество циклов спуска-подъема
* Режим работы ДГНУ
* Состояние ДГНУ, защиты, связи
* Скорость плунжера
* Момент ДГНУ
* Наработка ДГНУ
* Высота
* Замедление
* Пауза
* Ускорение

Далее представлены требования, выдвигаемые к разрабатываемому веб-модулю.

Требования к серверной части: Back-end

1. Реализовать авторизацию
2. Реализовать доступ по ролям (администратор - доступна настройка объектов, пользователь - доступен только просмотр)
3. CRUD по объектам
4. CRUD по типу объектов
5. CRUD по свойствам у объектов
6. CRUD по событиям на объектах
7. CRUD по архивам данных у объектов
8. Получение архива по всем свойствам с временным промежутком
9. Сделать для архива фильтрацию по времени
10. Хранение данных в БД

Требования к клиентской части:

Верхняя шапка для всех страниц кроме страницы авторизации:

1. Мониторинг объектов
2. Все объекты
3. События
4. Архив значений
5. Тренды

Также в шапке указать Логин пользователя и кнопку выхода.

1. Страница авторизации
2. Страница мониторинга объектов (отобразить название объекта, показатели, последние событие, дату события). Реализовать фильтрацию по типам объекта.
3. Страница просмотра объектов - список всех объектов. Возможность выбрать объект и вывод информации по нему и его показатели.
4. На выбранном объекте реализовать кнопку “Настройка объекта”. При ее нажатии появление диалогового окна, где можно настроить свойства.
5. Страница просмотра всех событий (вывод информаций по событию, возможность указывать временной промежуток)
6. Страница архив значений

- шахматка жидкости - вывод по дням показателя количества жидкости

- двухчасовки - вывод по часам показателей количества жидкости

- отбраковка замеров - вывод информации по отбракованным показателям

1. Страница тренды. Отображение в виде графиков архива значений по свойствам. Возможность указать конкретные свойства и задать временной промежуток.

Подробное описание

Запуск системы

Для запуска приложения WEB "КРОН-ТМ2" необходимо наличие на необходимо установленного Web браузера и доступа в интернет.

После загрузки приложение WEB "КРОН-ТМ2" должен предлагать ввести данные учетной записи (Пользователь, пароль).

После входа в систему должна загрузиться главная форма приложения.

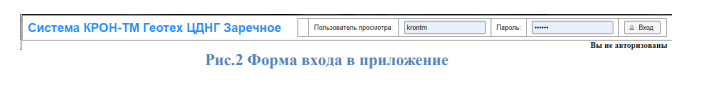


Рисунок 2 – Форма входа в приложение

Описание интерфейса

Главная форма – форма мониторинга свойств, должна открываться при запуске Web приложения и иметь вид, представленный на Рисунке 3



Рисунок 3 – Мониторинг свойств

Мониторинг свойств должен быть представлен в виде таблицы с объектами, измеренными значениями и событиями. В верхней части таблицы должно находиться меню, состоящее из следующих пунктов:

* Мониторинг свойств
* События
* Архив значений
* Шахматка жидкости
* 2-х часовки жидкости
* Все свойства
* Тренды

Если выбраны все объекты, то в последней кнопке меню будет отображаться надпись "Все объекты", если выбран конкретный объект, будет отображаться имя этого объекта. Выбор объекта можно будет осуществить в форме "Мониторинг свойств" щелчком мыши по названию объекта в таблице данных или в глобальном фильтре, находящийся в верхнем правом углу.

После выбора объекта должна загрузиться форма с информацией о свойствах, путях и текущих значениях объекта. В нижней области форм отображается архив событий за последние 6 часов, однако в фильтре можно выбрать любой период и нажать на кнопку “Обновить”.

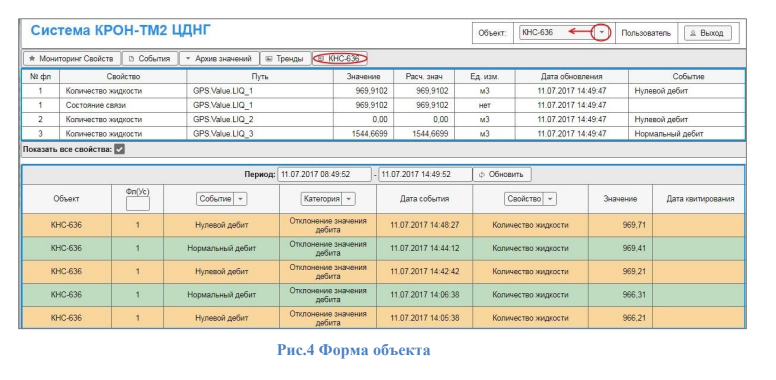


Рисунок 4 – Форма объекта

В правом нижнем углу должна находиться кнопка экспорта данных в файл формата Excel - кнопка с логотипом Excel.

Описание столбцов таблицы:

\*Объект - Наименование объекта.

\*Тип - тип НПО (Скважина, ГЗУ, БГ, ДГНУ и т. д.)

\*No фл- номер фланца.

\*Описание (Скважина) - описание свойства или скважины

\*Накоп.пред.сут – количество жидкости, добытое скважиной за прошедшие сутки (разница между накопленным значением БЭСКЖ на 00:00 часов прошедших суток и текущих).

\*Накоп.тек.сут - количество жидкости добытое с 00:00 часов текущих суток.

\*Qреж, м3/сут – режимный дебит скважины, соответствующий режиму ИС «КРОН».

\*Qсут, м3/сут – дебит скважины, рассчитанный за прошедшие сутки

\*Откл. от реж, м3 – отклонение фактического дебита скважины, рассчитанного по данным за прошедшие сутки от режимного дебита.

\*Pлин,атм– давление на устье скважины, в атмосферах, замеряемое датчиком давления «Метран». Если значение продублировано на все скважины объекта, то давление замеряется на общей трубе.

\*Дата обновления – дата и время последней обработки соответствующего тега

\*Событие – последнее событие по данному объекту нефтедобычи.

Регистрация событий

События представляют собой зарегистрированные факты изменения состояния объекта: наличия нулевых замеров, возобновления поступления положительных замеров, потери и восстановления связи с объектом.

В предпоследней колонке формы мониторинга должны отображаться последние события по каждому нефтепромысловому объекту.

Кроме этого, перейдя в форму "События" должен отобразиться список всех зарегистрированных событий, имеющий вид, представленный на Рисунке 5. По умолчанию должны отобразиться события за последние 6 часов, однако в фильтре можно выбрать любой период и нажать на кнопку “Обновить”.

Данная форма должна отображать полную историю событий всех объектов, если выбраны "ВСЕ ОБЪЕКТЫ" или полную историю событий выбранного объекта, если выбран один объект.

Для удобства поиска события должна быть возможность фильтрации нажатием на кнопку в названии столбца.

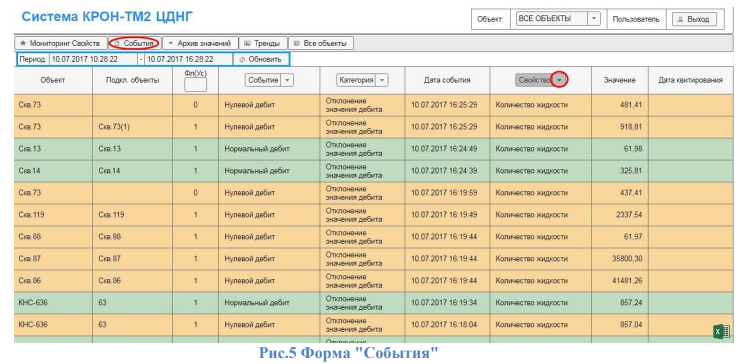


Рисунок 5 – Форма событие

В форме должен отображаться объект, на котором возникло событие, категория события, свойство и дата его регистрации. Сравнивая дату квитирования с датой возникновения события, можно производить оценку оперативности реагирования диспетчера на возникновение внештатных ситуаций.

Архив значений

Интерфейс Web приложения должен позволять просматривать архив по значениям свойств. Данная информация за выбранный интервал времени должна быть доступна в форме «Архив значений», запускаемая нажатием на кнопку "Архив значений" в меню.

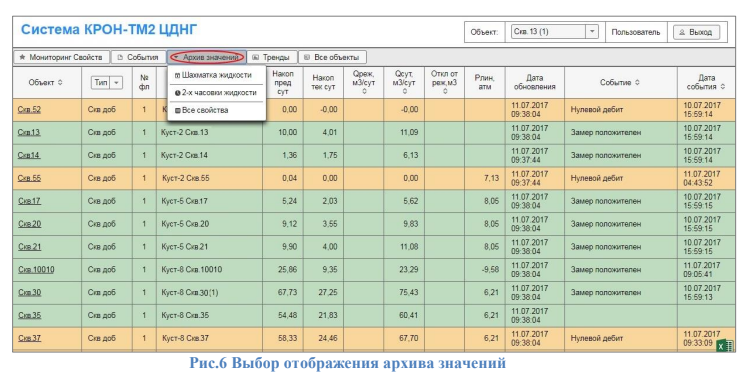


Рисунок 6 – Выбор отображения архива значений

В выпадающем списке необходимо выбрать форму отображения количества добытой жидкости в разрезе дня, "Шахматка жидкости" (Рисунок 7) и в разрезе двухчасовок (Рисунок 8). Форма "Все свойства" должна содержать значения всех свойств, описанных для выбранного объекта или всех объектов. Представление данных вкладки "Все свойства" должно состоять из объекта, свойств объекта, значения, даты и времени получения значения.

В формах должна быть реализована фильтрация для выбора периода и значения отбора, а для формы "Все свойства" должна быть фильтрация по Свойству.

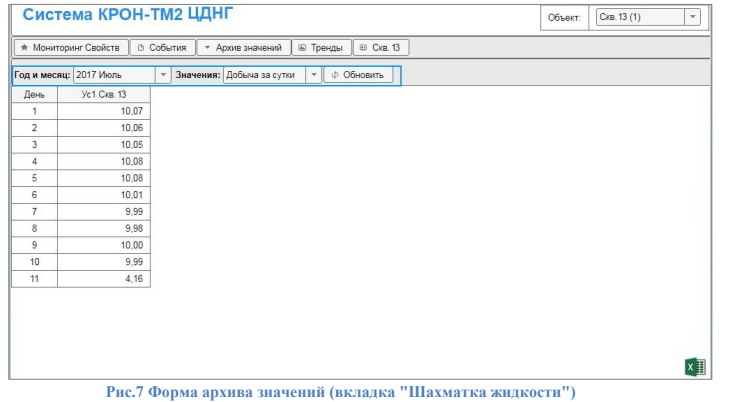


Рисунок 7 – Форма архива значений (вкладка «Шахматка жидкости»)

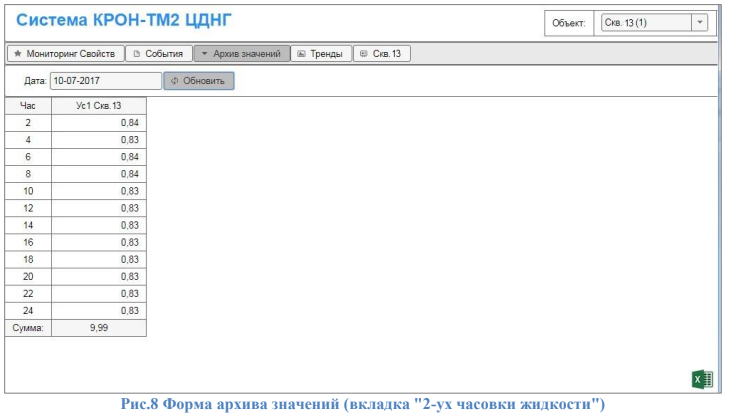


Рисунок 8 – Форма архива значений (вкладка «2-ух часовки жидкости»)

Тренд свойств

В приложении должен быть разработан инструмент для отображения исторического (архивного) тренда свойств объекта, т.е. отображать в виде графиков значения архивных данных. Для этого в форме необходимо выбрать:

* интересуемый временный промежуток
* интересуемые показатели, установив галочки

После всех настроек необходимо нажать на кнопку "Запрос истории".

По умолчанию период должен быть настроен на 6 часов. После выставления необходимого периода еще раз нажать на кнопку "Запрос истории".

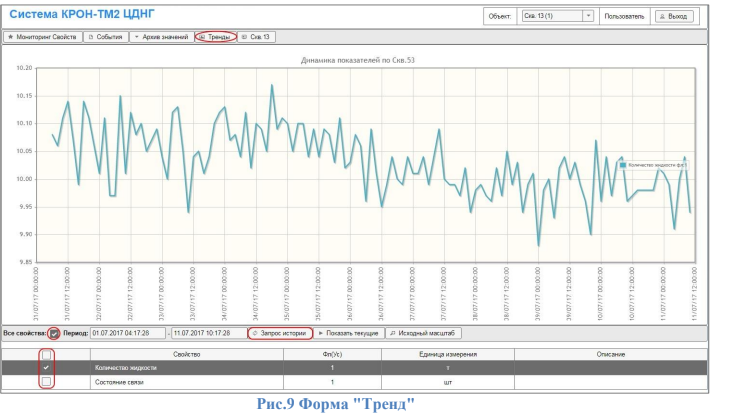


Рисунок 9 – Форма тренд

Режимы работы

В приложении должна быть реализована работа в трех режимах:

* Просмотр
* Управление
* Разработка

По умолчанию приложение должно быть настроено на режим просмотра. В данном режиме управление объектами будет не доступно. Управление объектом должно быть доступно в режиме управления и разработки, путем выбора пункта "Управление объектом" из раздела "Объекты" главного меню. Затем программа запрашивает пароль для доступа к управлению.

Режим управления должен действовать ровно столько, сколько прописано в форме "Настройки", пункт "Время действия управления". Редактирование формы объекта и добавление свойств объекта должно быть доступно в режиме разработки. Выход из программы в режиме просмотра так же должен сопровождаться подтверждением завершения работы

Интерфейс программы должен быть представлен несколькими формами, отображающими текущую и архивную информацию, а также формой настроек.

Во всех формах должно быть принято четыре вида индикации:

- объект и АСУ в норме

- АСУ авария (состояние связи, состояние двери, блокировка работы и т. п. не соответствуют заданным условиям или алгоритму работы объекта)

- Технологическое предупреждение (отклонение значения дебита, задержка действия объекта)

- Технологическая авария (не рабочее состояние (авария) объекта или агрегата, отсутствие питания, максимальные давление и т.д.)

Настройка объектов

В настройки объекта будет можно задать название, выбрать компанию, период обработки, состояние объекта и выбрать объект синхронизации с которого будут скопированы свойства.

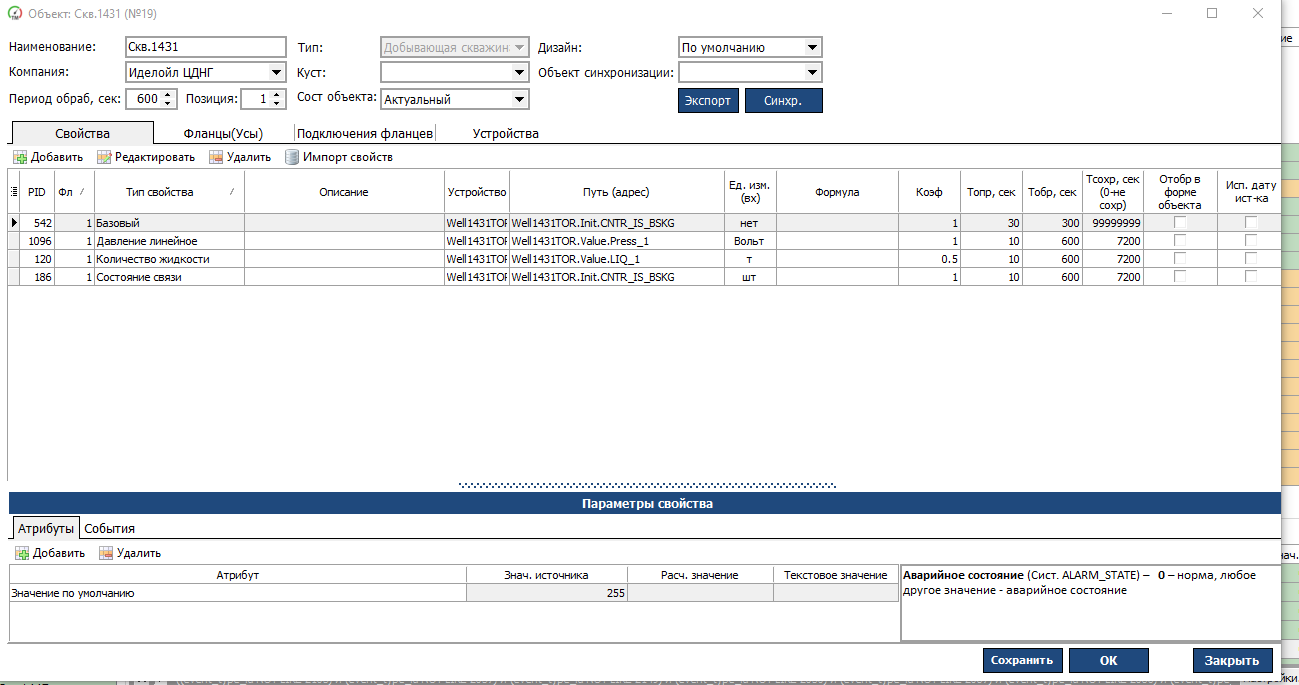


Рисунок 10 – Настройка объектов

Техническое задание на разработку информационной системы приведено в приложении А.

# 1.2 Проектирование диаграммы прецедентов

Диаграмма прецедентов или диаграмма вариантов использования — диаграмма, отражающая отношения между акторами и прецедентами и являющаяся составной частью модели прецедентов, позволяющей описать систему на концептуальном уровне.

Основное назначение диаграммы — описание функциональности и поведения, позволяющее заказчику, конечному пользователю и разработчику совместно обсуждать проектируемую или существующую систему.

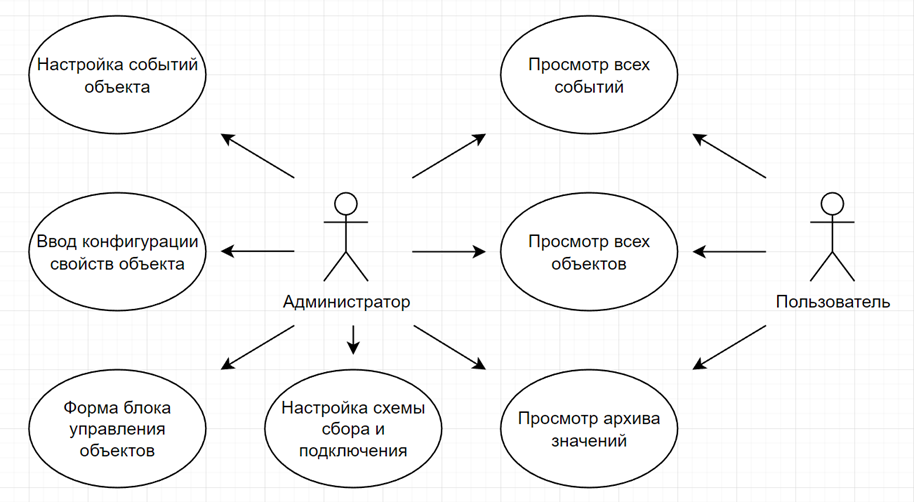


Рисунок 11 – Диаграмма прецедентов

# 1.3 Проектирование диаграммы классов

Исходя из анализа предметной области, построения бизнес-процессов и диаграммы прецедентов была построена диаграмма классов. Основными сущностями будут объект, свойства, события, логи свойств.

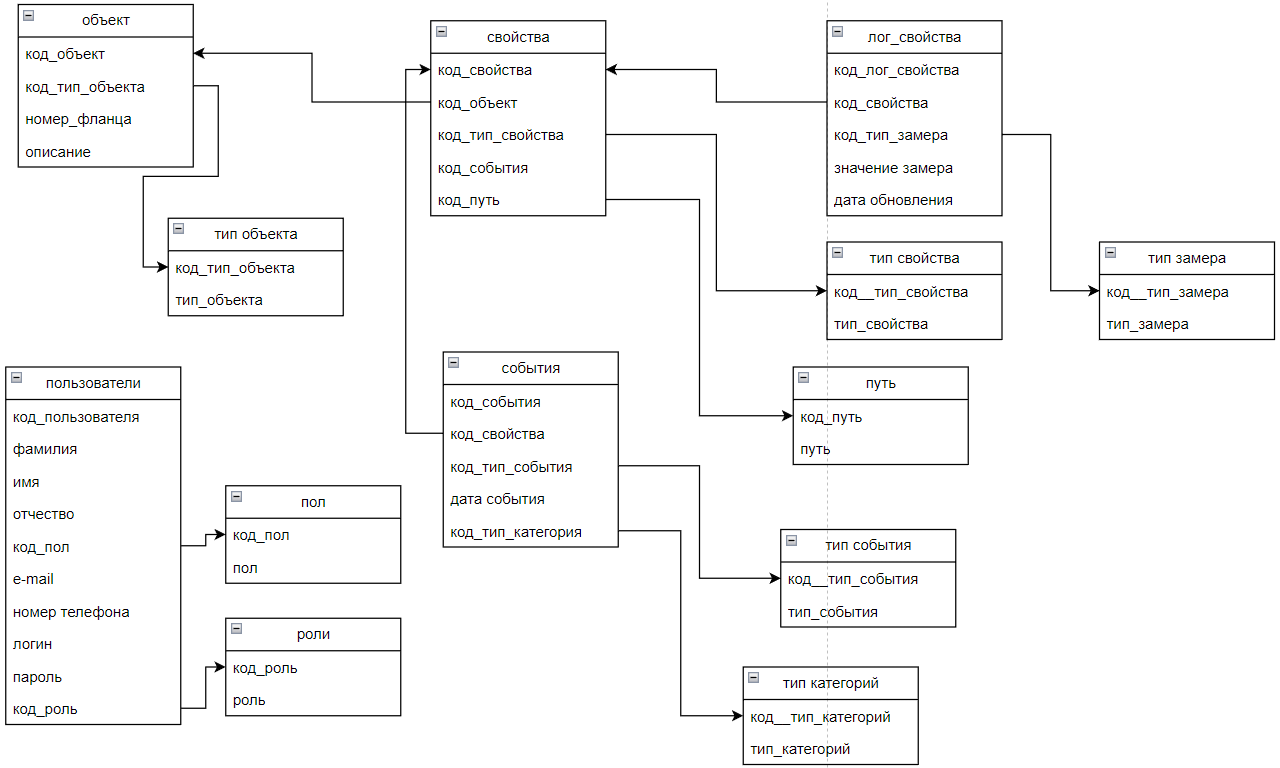


Рисунок 12 – Диаграмма классов

# 1.4 Проектирование схемы данных и словаря данных

При создании информационной системы необходимо учитывать, как интерфейс будет взаимодействовать с серверной частью. Однако более важным является построение и дизайн базы данных. Отношения между вашими формами данных приведут к построению схемы базы данных.

Схема базы данных включает в себя описание содержания, структуры и ограничений целостности, используемые для создания и поддержки базы данных, и может представлять собой логическую конфигурацию либо целой реляционной базы данных, либо ее части. Схема базы данных может существовать как в виде наглядного представления базы данных, так и в виде набора формул.

Исходя из диаграммы классов, можно составить схему данных, которая будет предоставлена ниже:

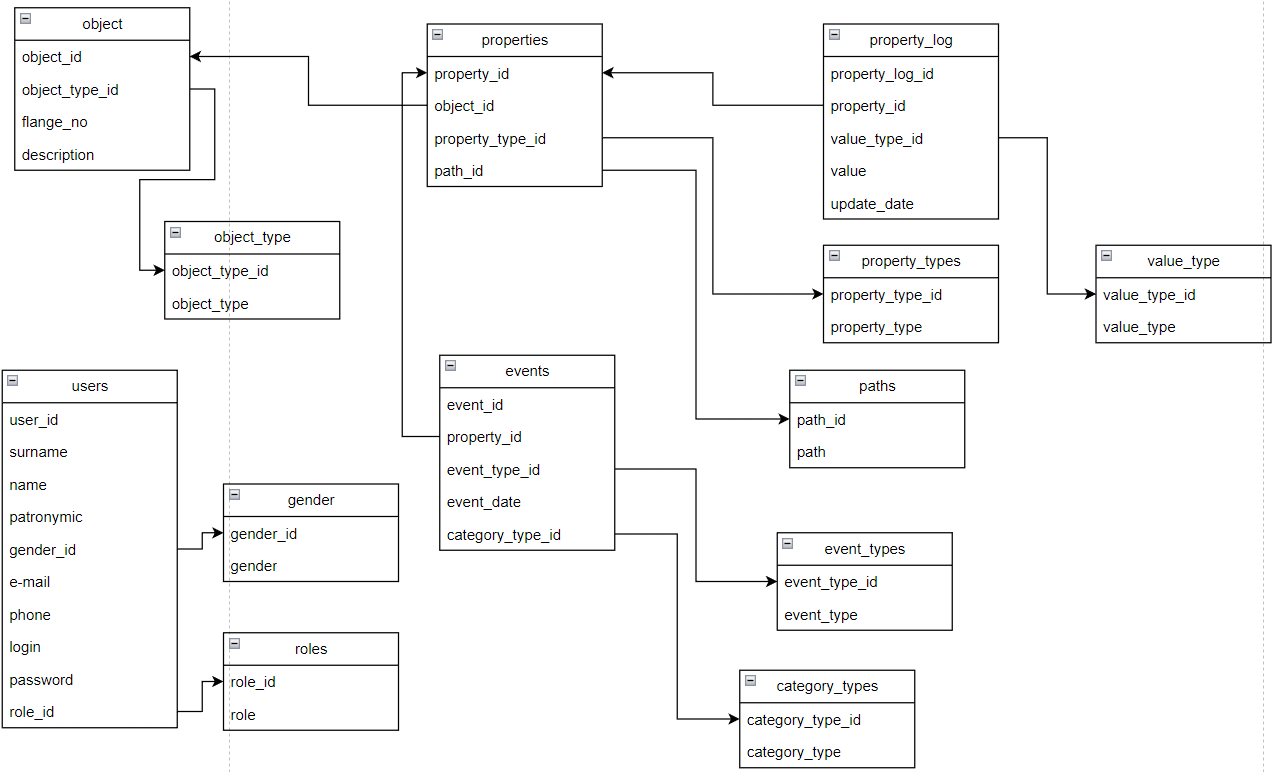


Рисунок 13 – Схема базы данных

Таблица «users» предназначена для хранения информации о пользователях.

Таблица 1 – Пользователи

|  |  |  |  |
| --- | --- | --- | --- |
| Наименование | Тип поля | Ключ | Описание |
| user\_id | Число (11) | Первичный ключ | Код пользователя |
| surname | Строка (30) | Нет | Фамилия |
| name | Строка (30) | Нет | Имя |
| patronymic | Строка (30) | Нет | Отчество |
| gender\_id | Число (11) | Внешний ключ | Пол |
| e-mail | Строка (30) | Нет | Почта |
| phone | Строка (30) | Нет | Номер телефона |
| login | Строка (30) | Нет | Логин |
| password | Строка (30) | Нет | Пароль |
| role\_id | Число (11) | Внешний ключ | Код роли |

Таблица «object» предназначена для хранения данных о технических установках.

Таблица 2 – Объект

|  |  |  |  |
| --- | --- | --- | --- |
| Наименование | Тип поля | Ключ | Описание |
| object\_id | Число (11) | Первичный ключ | Код объекта |
| object\_type\_id | Число (11) | Внешний ключ | Тип объекта |
| flange\_no | Число (11) | Нет | Номер фланца |
| description | Строка (255) | Нет | Описание |

Таблица «properties» предназначена для хранения данных о свойствах объекта.

Таблица 3 – Свойства

|  |  |  |  |
| --- | --- | --- | --- |
| Наименование | Тип поля | Ключ | Описание |
| property\_id | Число (11) | Первичный ключ | Код свойства |
| object\_id | Число (11) | Внешний ключ | Объект |
| property\_type\_id | Число (11) | Внешний ключ | Тип свойства |
| path\_id | Число (11) | Внешний ключ | Путь к свойству |

Таблица «property\_log» предназначена для хранения замеров с датчиков контроллера.

Таблица 4 – Логи свойств

|  |  |  |  |
| --- | --- | --- | --- |
| Наименование | Тип поля | Ключ | Описание |
| property\_log\_id | Число (11) | Первичный ключ | Код логов свойств |
| property\_id | Число (11) | Внешний ключ | Свойство |
| value\_type\_id | Число (11) | Внешний ключ | Тип замера |
| value | Строка (50) | Нет | Значение замера |
| update\_date | Дата | Нет | Дата обновления |

Таблица «events» предназначена для хранения событий объекта.

Таблица 5 – События

|  |  |  |  |
| --- | --- | --- | --- |
| Наименование | Тип поля | Ключ | Описание |
| event\_id | Число (11) | Первичный ключ | Код события |
| property\_id | Число (11) | Внешний ключ | Свойство |
| event\_type\_id | Число (11) | Внешний ключ | Тип события |
| event\_date | Дата | Нет | Дата события |
| category\_type\_id | Число (11) | Внешний ключ | Категория события |

Таблица «object\_type» предназначена для хранения типов (видов) объекта.

Таблица 6 – Типы объекта

|  |  |  |  |
| --- | --- | --- | --- |
| Наименование | Тип поля | Ключ | Описание |
| object\_type\_id | Число (11) | Первичный ключ | Код типа объекта |
| object\_type | Строка (50) | Нет | Тип объекта |

Таблица «property\_types» предназначена для хранения типов (видов) свойств объекта.

Таблица 7 – Типы свойства

|  |  |  |  |
| --- | --- | --- | --- |
| Наименование | Тип поля | Ключ | Описание |
| property\_type\_id | Число (11) | Первичный ключ | Код типа свойства |
| property\_type | Строка (50) | Нет | Тип свойства |

Таблица «paths» предназначена для хранения пути до свойства объекта.

Таблица 8 – Пути

|  |  |  |  |
| --- | --- | --- | --- |
| Наименование | Тип поля | Ключ | Описание |
| path \_id | Число (11) | Первичный ключ | Код пути |
| path | Строка (255) | Нет | Путь |

Таблица «value\_type» предназначена для хранения типов (видов) замеров с датчиков объекта.

Таблица 9 – Типы замеров

|  |  |  |  |
| --- | --- | --- | --- |
| Наименование | Тип поля | Ключ | Описание |
| value\_type\_id | Число (11) | Первичный ключ | Код типа замера |
| value\_type | Строка (50) | Нет | Тип замера |

Таблица «events\_types» предназначена для хранения типов (видов) событий.

Таблица 10 – Типы события

|  |  |  |  |
| --- | --- | --- | --- |
| Наименование | Тип поля | Ключ | Описание |
| event\_type\_id | Число (11) | Первичный ключ | Код типа события |
| event\_type | Строка (50) | Нет | Тип события |

Таблица «category\_types» предназначена для хранения типов (видов) категорий событий.

Таблица 11 – Типы категории

|  |  |  |  |
| --- | --- | --- | --- |
| Наименование | Тип поля | Ключ | Описание |
| category\_type\_id | Число (11) | Первичный ключ | Код типа категории |
| category\_type | Строка (50) | Нет | Тип категории |

Таблица «gender» предназначена для хранения пола пользователя.

Таблица 12 – Пол пользователя

|  |  |  |  |
| --- | --- | --- | --- |
| Наименование | Тип поля | Ключ | Описание |
| gender\_id | Число (11) | Первичный ключ | Код пола |
| gender | Строка (50) | Нет | Пол пользователя |

Таблица «roles» предназначена для хранения ролей пользователя.

Таблица 13 – Роли пользователя

|  |  |  |  |
| --- | --- | --- | --- |
| Наименование | Тип поля | Ключ | Описание |
| role\_id | Число (11) | Первичный ключ | Код роли |
| role | Строка (50) | Нет | Роль пользователя |

# 1.5 Проектирований интерфейса информационной системы

Проектирование простого и понятного интерфейса также важная часть разработки информационной системы.

Информационная система не имеет определенную возрастную группу людей, с ней будут работать, как и молодые, так и возрастные люди, поэтому очень важно, чтобы интерфейс был интуитивно понятен и не вызывал сложностей во время работы с ним.

Для обеспечения эргономичности информационной системы было выделено несколько качеств:

* Простая навигация, понятное меню
* Логичная и очевидная структура
* Логичная и приятная для восприятия цветовая схема
* Цвета шрифта и фона высококонтрастны по отношению друг к другу

Цветовая схема — это набор нескольких оттенков, которые будут использоваться на сайте. Она поможет создать более продуманный и целостный дизайн, а ещё упростит работу с информационной системой.

Цветовая палитра была разработана, опираясь на фирменный стиль компании – в бело-голубом цвете. Это хорошее и приятное сочетание цветов для восприятия глаза и не вызовет усталости при длительной работе с информационной системой, но также остается строгой в соответствии с тем, что информационная система – корпоративная.

Основная часть интерфейса была представлена в пункте 1.1, но в процессе разработки макет может меняться – конечный результат может не совпадать с предоставленным макетом.



Рисунок 14 – Шапка приложения

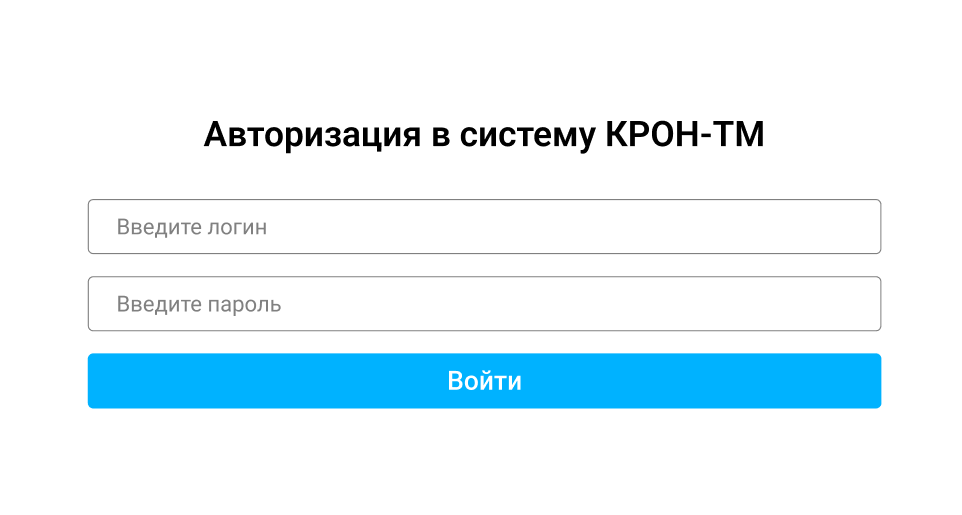


Рисунок 15 – Форма авторизации

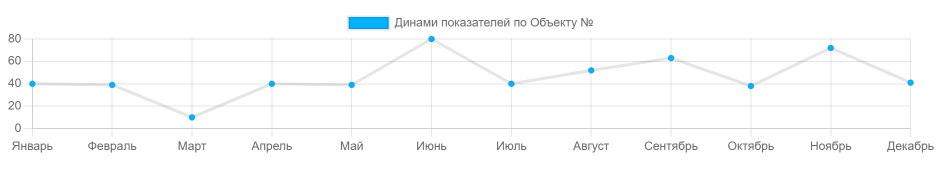


Рисунок 16 – Форма тренд

# 1.6 Разработка алгоритма информационной системы

Алгоритм – набор инструкций, описывающих порядок действий исполнителя для достижения некоторого результата.

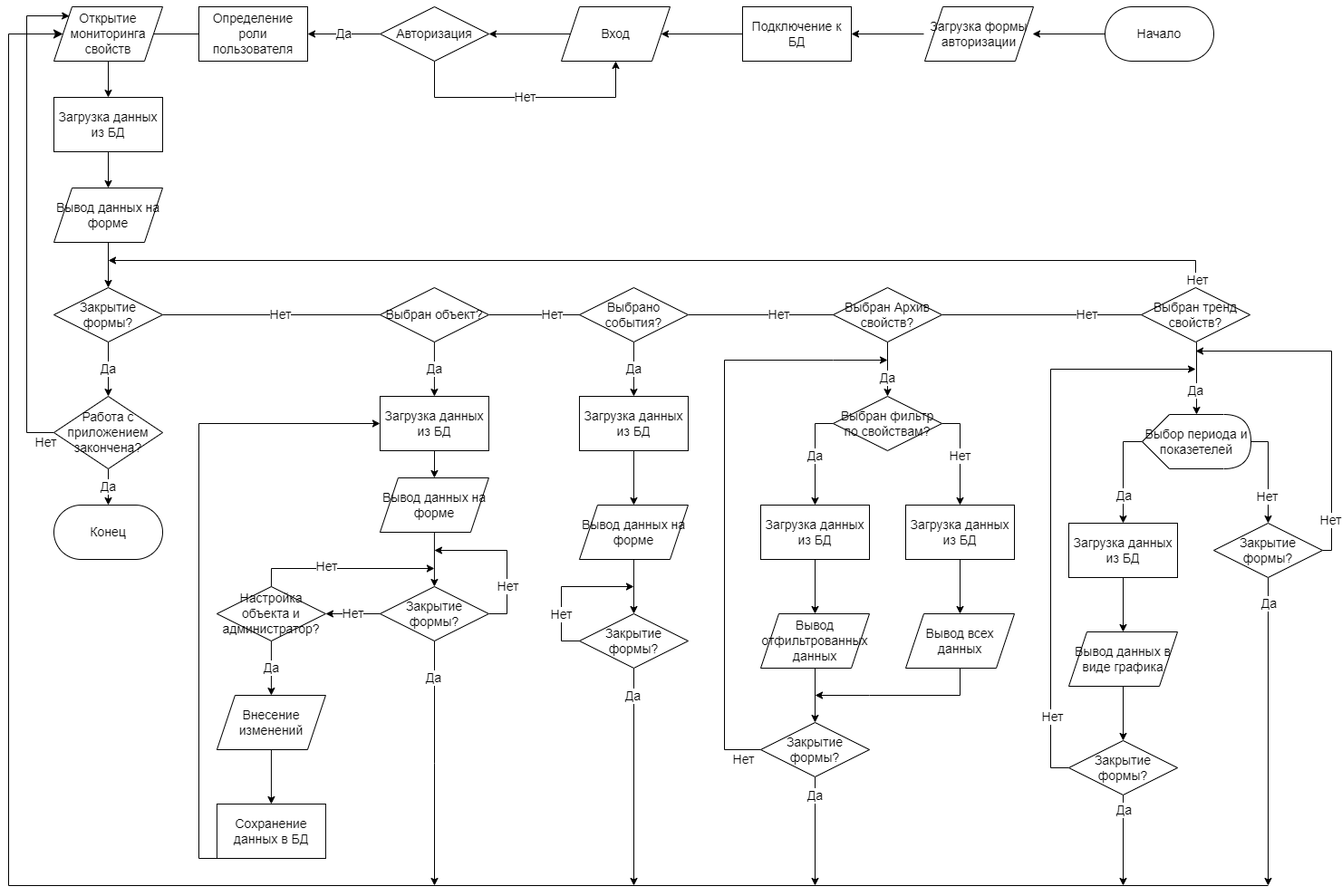


Рисунок 17 – Алгоритм работы ИС

Инструкция пользователя и администратора приведена в приложении Б.

# 1.7 Выбор программных и технических средств для разработки информационной системы

Информационная система конфигурирования объектов телеметрии необходимая система для сферы добычи и переработки, для ее разработки требуется не только высокое знание сферы и разработки приложений, но и использования современных технологий разработки.

Для разработки информационной системы были выбраны актуальные языки и фреймворки в результате этого получился следующий стек разработки:

Frontend разработка будет вестись на фреймворке Vue.js (JavaScript).

Vue.js – это прогрессивный фреймворк для разработки пользовательских интерфейсов и одностраничных веб-приложений на языке JavaScript.

Vue.js создан пригодным для постепенного внедрения. Его ядро в первую очередь решает задачи уровня представления, что упрощает интеграцию с другими библиотеками и существующими проектами. С другой стороны, Vue.js полностью подходит и для создания сложных одностраничных приложений (SPA, Single-Page Applications), если использовать его совместно с современными инструментами и дополнительными библиотеками.

Vue.js облегчит не только разработку пользовательской части, но и облегчит дальнейшее масштабирование системы.

Backend разработка будет вестись на фреймворке Spring Framework (Java).

Именно Java нередко используется для backend разработки в качестве основного языка программирования, и слывет в кругу разработчиков, как «вечнозеленый». Spring Framework по праву держит пальму первенства среди наиболее востребованных фреймворков, применяемых для разработки автономных и производственных веб-приложений.

Базой сервиса приложения станет Spring boot.

Все программные платформы на базе Spring boot предлагают массу полезных опций, среди которых: кэширование, управляемость настройками и транзакциями, наблюдение и обеспечение безопасности. А в программной платформе Spring boot, к тому же, отсутствуют большие траты на конфигурацию.

СУБД MySQL — свободная реляционная система управления базами данных. Разработку и поддержку MySQL осуществляет корпорация Oracle.

Технические средства для пользователя:

Процессор: Intel Core i3-2xxx/amd fx-xx/ryzen 3 или более современная версия. Оперативная память: не менее 2гб свободной памяти после запуска ОС и прочих сторонних приложений. Встроенная или десктопная видеокарта 2 поколения NVIDIA/2 поколения Radeon или более современная версия. Разрешение экрана: не менее 1600x900

Поддерживаемые настольные операционные системы:

* windows 7 64 бит и более поздние
* Apple OS X 10.7 и более поздние
* Ubuntu 16.04 и более поздние

Поддерживаемые браузеры для операционных систем ПК:

* Google Chrome 28.0 и более поздние
* Mozilla Firefox 47.0 и более поздние
* Apple Safari 8.0 и более поздние (для OS X)
* Microsoft Edge 13.0 и более поздние
* Opera 36.0 и более поздние

Скорость соединения не менее 1 мбит/c

Технические средства для сервера: процессор Intel Core i7-7800 или AMD Ryzen 7 1700 или более современная версия; оперативная память 4 Гбайт и выше (рекомендуется 8-16 Гбайт); жесткий диск или твердотельный накопитель 160Гб и более; USB-порт не менее 4 штук. Видеокарта 10 поколения NVIDIA/5 поколения Radeon или более современная версия. Операционная система: windows server 2012 или более поздняя. Скорость соединения не менее 100 мбит/c.

# 2 Разработка информационной системы

# 2.1 Программирование на стороне клиента

Информационная системы создавалась на основе многостраничной информационной системы.

Многостраничная информационная система является одним из наиболее распространенных типов информационных систем в интернете. Она состоит из нескольких страниц, которые могут быть связаны между собой различными способами, например, через навигационное меню или гиперссылки. Каждая страница может содержать уникальный контент, такой как текст, изображения, видео или аудио, и предназначена для определенной цели, например, предоставления информации о продукте или услуге, обучения пользователей или обмена информацией.

Преимущества многостраничной информационной системы заключаются в том, что она позволяет более полно и подробно представить информацию о компании, продукте или услуге, а также улучшить пользовательский опыт за счет удобной навигации и разделения контента на отдельные страницы. Кроме того, это может быть лучшим выбором для SEO-оптимизации, так как каждая страница может быть оптимизирована для определенных ключевых слов и запросов.

Клиентская часть разработана на фреймоворке Vue.js и библиотеке семейства PrimeFaces: PrimeVue, PrimeFlex, PrimeIcons.

Vue.js — JavaScript-фреймворк с открытым исходным кодом для создания пользовательских интерфейсов в реактивном стиле.

PrimeFaces — это библиотека компонентов пользовательского интерфейса с открытым исходным кодом для приложений.

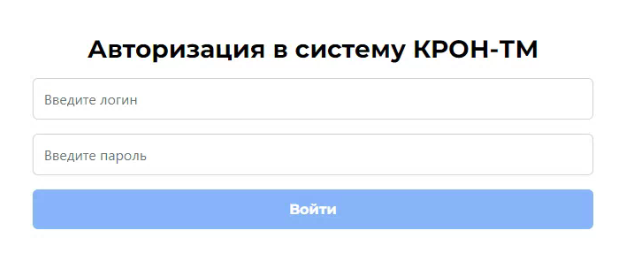


Рисунок 18 – Форма авторизации

Ниже представлен код формы авторизации:

<template>

<form class="w-full xl:w-5 p-3" @submit.prevent="auth">

<p class="text-center text-xl xl:text-3xl font-bold mb-3">Авторизация в систему КРОН-ТМ</p>

<InputText placeholder="Введите логин" type="text" class="w-full mb-3" v-model="username" required/>

<InputText placeholder="Введите пароль" type="password" class="w-full mb-3" v-model="password" required/>

<Button type="submit" label="Войти" class="w-full mb-3" :disabled="password.length < 6"/>

</form>

</template>

<script>

export default {

data() {

return {

username: '',

password: '',

}

},

methods: {

auth() {

this.$store.dispatch('onLogin', {

username: this.username,

password: this.password

}).catch((e) => {

this.$toast.add({

severity: 'error',

detail: e.response.data === 'Bad credentials' ? 'Неверный логин или пароль' : e.response.data,

life: 3000

});

});

}

}

}

</script>

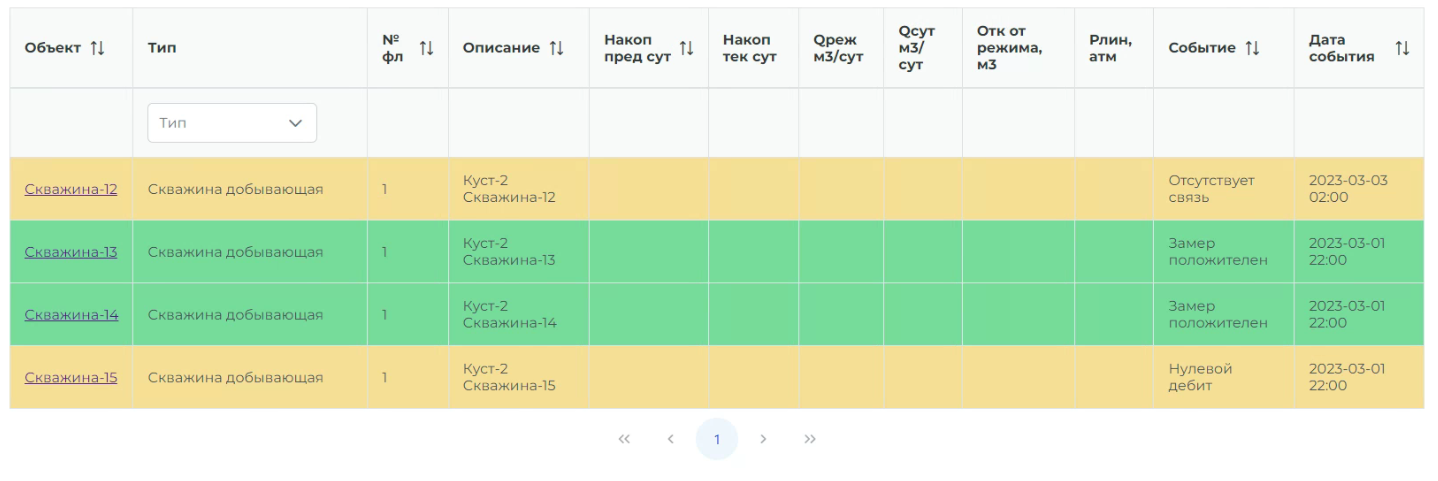


Рисунок 19 – Мониторинг свойств

Ниже представлен код мониторинга свойств:

<template>

<MyDataTable

:array="array"

:value="objects"

:loading="loading"

:filters="filters"

:pagination="true"

filterDisplay="row"

:options="{types: types}"

/>

</template>

<script>

import objectapi from "../service/object";

import {FilterMatchMode} from "primevue/api";

import MyDataTable from "../components/datatables/MyDataTable.vue";

export default {

components: {MyDataTable},

data() {

return {

objects: this.$store.state.object.objects,

types: null,

loading: true,

filters: {

object\_type: {value: null, matchMode: FilterMatchMode.IN}

},

array: [

{header: 'Объект', field: 'object\_name', sortable: true, link: {name: "object", param: 'object\_id'}},

{header: 'Тип', field: 'object\_type', data: 'object\_type', option: 'types'},

{header: '№ фл', field: 'flange\_no', sortable: true},

{header: 'Описание', field: 'description', sortable: true},

{header: 'Накоп пред сут', field: '', sortable: true},

{header: 'Накоп тек сут', field: ''},

{header: 'Qреж м3/сут', field: ''},

{header: 'Qсут м3/сут', field: ''},

{header: 'Отк от режима, м3', field: ''},

{header: 'Рлин, атм', field: ''},

{header: 'Событие', field: 'event\_type', sortable: true},

{header: 'Дата события', field: 'event\_date', sortable: true},

]

}

},

methods: {

async getAllTypes() {

try {

let types = await objectapi.getAllObjectTypes();

this.types = types.data.map(a => a.object\_type);

this.loading = false;

} catch (e) {

this.$toast.add({severity: 'error', detail: 'Произошла ошибка', life: 3000});

}

},

},

mounted() {

this.getAllTypes();

}

}

</script>

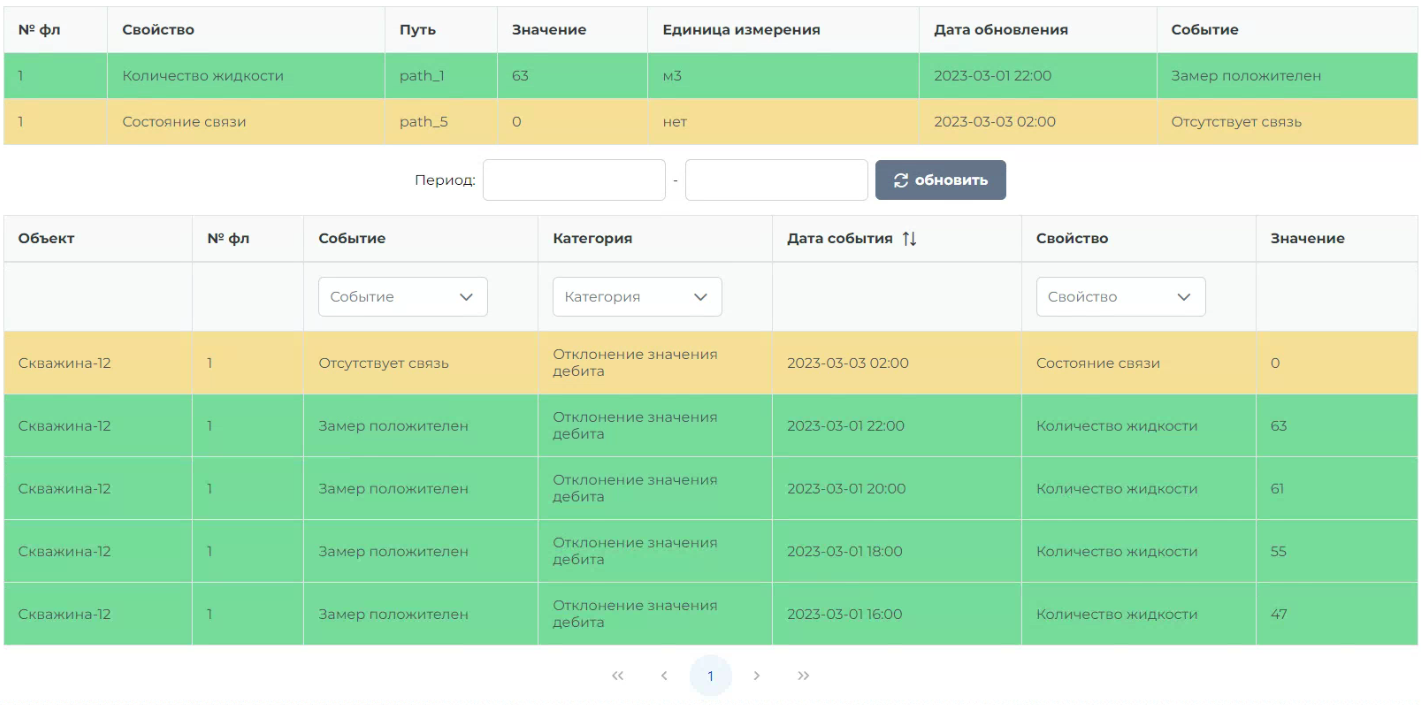


Рисунок 20 – Форма объекта

Ниже представлен код формы объекта:

<template>

<MyDataTable

:array="propertyColumns"

:value="objectProperties"

:loading="loading"

/>

<form class="flex w-12 justify-content-center align-items-center p-3 gap-2" @submit.prevent="getObjectEvents">

<p>Период:</p>

<Calendar v-model="date\_start" showTime hourFormat="24"/>

-

<Calendar v-model="date\_end" showTime hourFormat="24"/>

<Button type="submit" label="обновить" severity="secondary" icon="pi pi-sync"/>

</form>

<MyDataTable

:array="eventColumns"

:value="objectEvents"

:loading="loading"

:filters="filters"

:pagination="true"

filterDisplay="row"

:options="{types: types, categories: categories, properties: properties}"

/>

</template>

<script>

import eventapi from "../service/event";

import categoryapi from "../service/category";

import propertyapi from "../service/property";

import {FilterMatchMode} from "primevue/api";

import MyDataTable from "../components/datatables/MyDataTable.vue";

export default {

components: {MyDataTable},

data() {

return {

objectEvents: null,

objectProperties: null,

types: null,

categories: null,

properties: null,

loading: true,

date\_start: null,

date\_end: null,

filters: {

event\_type: {value: null, matchMode: FilterMatchMode.IN},

category: {value: null, matchMode: FilterMatchMode.IN},

property\_type: {value: null, matchMode: FilterMatchMode.IN}

},

eventColumns: [

{header: 'Объект', field: 'object\_name'},

{header: '№ фл', field: 'flange\_no'},

{header: 'Событие', field: 'event\_type', data: 'event\_type', option: 'types'},

{header: 'Категория', field: 'category', data: 'category', option: 'categories'},

{header: 'Дата события', field: 'event\_date', sortable: true},

{header: 'Свойство', field: 'property\_type', data: 'property\_type', option: 'properties'},

{header: 'Значение', field: 'value'}

],

propertyColumns: [

{header: '№ фл', field: 'flange\_no'},

{header: 'Свойство', field: 'property\_type'},

{header: 'Путь', field: 'path'},

{header: 'Значение', field: 'value'},

{header: 'Единица измерения', field: 'value\_type'},

{header: 'Дата обновления', field: 'event\_date'},

{header: 'Событие', field: 'event\_type'},

]

}

},

methods: {

async getObjectEvents() {

try {

this.loading = true;

let [objectEvents, objectProperties, types, categories, properties] = await Promise.all([

eventapi.getOneObjectEvents(

this.$route.params.id,

this.date\_start != null ? this.date\_start.toLocaleDateString('sv-SE') + ' ' + this.date\_start.toLocaleTimeString() : null,

this.date\_end != null ? this.date\_end.toLocaleDateString('sv-SE') + ' ' + this.date\_end.toLocaleTimeString() : null

),

propertyapi.getObjectProperties(this.$route.params.id),

eventapi.getAllEventTypes(),

categoryapi.getAllCategoryTypes(),

propertyapi.getAllPropertyTypes(),

]);

this.objectEvents = objectEvents.data;

this.objectProperties = objectProperties.data;

this.types = types.data.map(a => a.event\_type);

this.categories = categories.data.map(a => a.category);

this.properties = properties.data.map(a => a.property\_type);

this.loading = false;

} catch (e) {

this.$toast.add({severity: 'error', detail: 'Произошла ошибка', life: 3000});

}

},

},

mounted() {

this.getObjectEvents();

},

created() {

this.$watch(

() => this.$route.params,

() => this.getObjectEvents()

);

},

}

</script>

Остальные блоки в информационной системе выполнены аналогичным образом и приведены в приложении В.

# 2.2 Программирование на стороне сервера

Программирование на стороне сервера использовано для обработки запросов от клиента, доступа к базам данных, генерации динамических веб-страниц и других функций. Оно также может быть использовано для создания API (Application Programming Interface), которые позволяют различным приложениям взаимодействовать друг с другом.

Серверная часть информационной системы реализована в виде REST-API сервиса на фреймворке языка Java – Spring boot.

За хранение данных отвечает СУБД MySQL.

Рассмотрим пример функционала формирования списка объектов для мониторинга свойств.

Создание MonitoringResponse:

@Data

@Builder

@AllArgsConstructor

@NoArgsConstructor

public class MonitoringResponse {

private Integer object\_id;

private String object\_name;

private String object\_type;

private String flange\_no;

private String description;

private String event\_type;

@JsonFormat(pattern="yyyy-MM-dd HH:mm")

private LocalDateTime event\_date;

}

Формирование списка объектов для мониторинга свойств:

public List<MonitoringResponse> getAll() throws Exception {

var objects = objectRepository.findAll();

if (objects.isEmpty()) {

throw new Exception("Объекты не найдены");

}

List<MonitoringResponse> objectsDto = new ArrayList<>();

for (Object obj : objects) {

var event = eventRepository.getLastEventsFromObject(obj.getObject\_id());

MonitoringResponse objectDto = MonitoringResponse.builder()

.object\_id(obj.getObject\_id())

.object\_name(obj.getObject\_name())

.object\_type(obj.getObjectType().getObject\_type())

.flange\_no(obj.getFlange\_no())

.description(obj.getDescription())

.event\_type(event.getEventType().getEvent\_type())

.event\_date(event.getEvent\_date())

.build();

objectsDto.add(objectDto);

}

return objectsDto;

}

Обработка запроса от клиента:

@RestController

@RequestMapping("/api/v1/object")

@RequiredArgsConstructor

public class ObjectController {

private final ObjectService objectService;

@GetMapping

public ResponseEntity getAllObjects() {

try {

return ResponseEntity.ok(objectService.getAll());

} catch (Exception e) {

return ResponseEntity.badRequest().body(e.getMessage());

}

}

@GetMapping("/types")

public ResponseEntity getAllObjectTypes() {

try {

return ResponseEntity.ok(objectService.getAllTypes());

} catch (Exception e) {

return ResponseEntity.badRequest().body(e.getMessage());

}

}

}

Полный листинг программы приведен в приложении В.

# 2.3 Разработка подсистемы безопасности

Разработка подсистемы безопасности информационной системы - это процесс создания механизмов и процедур, которые обеспечивают защиту информации, хранимой и обрабатываемой в информационной системе, от возможных угроз и уязвимостей. Это включает в себя анализ угроз и уязвимостей, разработку политики безопасности, механизмов защиты, процедур обеспечения безопасности, обучение пользователей, тестирование и аудит безопасности, а также постоянное совершенствование подсистемы безопасности. Целью разработки подсистемы безопасности является обеспечение конфиденциальности, целостности и доступности информации в информационной системе.

В информационной системе разработана безопасная авторизация по JWT-токенам (JSON Web Token).

JSON Web Token — это открытый стандарт для создания токенов доступа, основанный на формате JSON. Как правило, используется для передачи данных для аутентификации в клиент-серверных приложениях.

Разработка JwtFilter:

public class JwtFilter extends OncePerRequestFilter {

private final JwtService jwtService;

private final UserDetailsService userDetailsService;

@Override

protected void doFilterInternal(

HttpServletRequest request,

HttpServletResponse response,

FilterChain filterChain

) throws ServletException, IOException {

final String header = request.getHeader("Authorization");

if (header == null || !header.startsWith("Bearer ")) {

filterChain.doFilter(request, response);

return;

}

final String token = header.substring(7);

if (jwtService.validateToken(token)) {

UserDetails userDetails = userDetailsService.loadUserByUsername(jwtService.extractToken(token));

UsernamePasswordAuthenticationToken authToken = new UsernamePasswordAuthenticationToken(

userDetails,

null,

userDetails.getAuthorities()

);

SecurityContextHolder.getContext().setAuthentication(authToken);} else {

response.setStatus(HttpServletResponse.SC\_FORBIDDEN);

response.getWriter().write(jwtService.extractToken(token));

return;}

filterChain.doFilter(request, response);}}

Авторизация по JWT-токенам позволяет идентифицировать пользователя и ограничивать его время авторизации в системе. Тем самым защищая информационную систему от несанкционированному доступу к API сервису.

Также все пароли в базе данных шифруются по bcrypt технологии.

bcrypt — адаптивная криптографическая хеш-функция формирования ключа, используемая для защищенного хранения паролей.

Ниже предоставлен код шифратора:

public PasswordEncoder passwordEncoder(){return new BCryptPasswordEncoder();

# 3 Тестирование информационной системы

Тестирование информационной системы - это процесс проверки работоспособности и соответствия заданным требованиям программного обеспечения. Оно включает в себя несколько этапов, таких как функциональное тестирование, тестирование производительности, тестирование безопасности и тестирование совместимости. Цель тестирования - обеспечить стабильную работу системы, убедиться в ее правильном функционировании и выявить возможные ошибки и недочеты для их устранения и улучшения информационной системы. Важно проводить тестирование не только перед запуском системы, но и в процессе ее использования, чтобы обеспечить ее надежность и эффективность.

В ходе разработки информационной системы были проведены:

1. Ручное тестирование - тестирование, проводимое вручную, без использования автоматизированных средств.

2. Автоматизированное тестирование - тестирование, проводимое с помощью специальных программных средств, которые позволяют автоматизировать процесс тестирования.

3. Функциональное тестирование - проверка соответствия системы требованиям и способности выполнять заданные функции.

4. Нефункциональное тестирование - проверка не функциональных аспектов системы, таких как производительность, безопасность, совместимость и т.д.

5. Тестирование на разных уровнях - тестирование на разных уровнях абстракции системы, например, модульное тестирование, интеграционное тестирование, системное тестирование и т.д.

6. Тестирование на разных стадиях жизненного цикла - тестирование на разных стадиях жизненного цикла разработки системы, например, тестирование на этапе анализа требований, проектирования, разработки и т.д.

7. Тестирование на разных платформах - тестирование на разных операционных системах, браузерах, устройствах и т.д.

В ходе разработки информационной системы, были проведены тестирования методом черного ящика, описанные в таблицах 14 и 15.

Таблица 14 – Тест-кейс №1

|  |  |
| --- | --- |
| Идентификатор | Login\_1.1 |
| Приоритет | Высокий |
| Модуль | Страница авторизации |
| Подмодуль | Вход |
| Шаги | 1. В поле логин ввести: admin  2. В поле пароль ввести: qweasd  3. Нажать на кнопку войти |
| Ожидаемый результат | 3. Успешный вход в систему |

Таблица 15 – Тест-кейс №2

|  |  |
| --- | --- |
| Идентификатор | Filter\_1.1 |
| Приоритет | Высокий |
| Модуль | Страница мониторинга свойств |
| Подмодуль | Фильтр по типу объекта |
| Шаги | 1. Перейти на страницу мониторинга свойств  2. В фильтре над типами объекта выбрать объект Скважина добывающая |
| Ожидаемый результат | 2. Список объектов отфлитровался по выбранному типу |

В ходе тестирования информационной системы были выявлены ошибки, которые описаны в таблице 16:

Таблица 16 – Тест-кейс №3

|  |  |
| --- | --- |
| Идентификатор | Filter\_1.1 |
| Приоритет | Высокий |
| Модуль | Страница регистрации |
| Подмодуль | Уникальность логина пользователя |
| Шаги | 1. Перейти на страницу регистрации  2. В поле логин ввести: admin  3. Заполнить остальные поля произвольными полями  4. Нажать кнопку регистрация |
| Ожидаемый результат | 4. Вывод ошибки «Данный пользователь уже зарегистрирован» |
| Фактический результат | Ошибка не вышла, пользователь зарегистрирован |

Для исправления ошибки было добавлено поле уникальности в сущности базе данных:



# 4 Оценивание информационной системы для выявления возможности ее модернизации

Оценка информационной системы необходима для выявления ее текущего состояния и определения того, насколько она соответствует потребностям бизнеса. Она позволяет выявить слабые места и проблемы в работе системы, а также потенциал для улучшения ее функциональности и производительности.

Выявление возможностей модернизации информационной системы позволяет бизнесу сохранять конкурентное преимущество и обеспечивать эффективность своей работы. Модернизация может включать в себя улучшение интерфейса, оптимизацию процессов, обновление программного обеспечения и аппаратного обеспечения, а также внедрение новых технологий.

Оценка информационной системы и выявление возможностей ее модернизации помогает бизнесу:

− Снизить затраты на обслуживание системы

− Увеличить производительность и эффективность работы

− Улучшить качество обслуживания клиентов

− Обеспечить конкурентоспособность на рынке.

В процессе оценки эффективности текущей информационной системы были выявлены ее слабые стороны, которые могут быть устранены путем модернизации.

Путем анализа бизнес-процессов, которые автоматизированы в текущей информационной системе, технологических решений, безопасности информационной системы, качества данных, включая проверку точности, полноты и актуальности, а также производительности и удобства работы с информационной системы, было принято решение о необходимости модернизации системы.

Улучшение авторизации пользователя добавлением refresh токенов.

Когда время жизни access token подходит к концу (приложение может само проверять время жизни, или дождаться пока во время очередного использования сервер ответит, что токен истек), приложение использует refresh token, чтобы обновить оба токена и продолжить использовать новый access token.

Это позволит исключить постоянную повторную авторизацию пользователя в системе.

Построение маршрута до места объекта. В информационной системе используется api Яндекс карт, для отображения местоположения объекта.

API позволяет отмечать на карте местоположение пользователя. Это может быть полезно, например, когда нужно построить маршрут от текущего местоположения пользователя или показать ему объекты, расположенные поблизости (геотаргетинг).

Для удобства нахождения маршрута до объекта, было принято решение, что требуется внедрить в информационную систему построение маршрута от текущего местоположения до технологического объекта.

Вывод данных в виде графиков для двухчасовок и шахмотки жидкости. Это позволит увидеть динамику данных, собранных с объектов.

Вывод данных в виде графиков имеет ряд преимуществ для информационной системы объектов телеметрии. В частности, графики позволяют наглядно отображать изменения параметров объектов во времени, что упрощает анализ данных и выявление тенденций, а также может повысить эффективность принятия решений, так как позволяет быстро определить, какие параметры требуют внимания и корректировки.

Графики могут помочь в выявлении аномалий и нештатных ситуаций, так как изменения параметров будут более заметны на графиках, чем в таблицах данных и могут быть полезны для визуализации результатов анализа данных и их представления заказчикам или пользователям информационной системы объектов телеметрии.

Таким образом, вывод данных в виде графиков является эффективным инструментом для анализа и визуализации данных в информационной системе объектов телеметрии.

Импорт данных в excel документ. Импорт данных мониторинга свойств, двухчасовок, шахмотки жидкости, тренда свойств позволит передавать данные или архивировать их не только внутри информационной системы, но и пользователям не связанных с ней.

Также данная модернизация имеет ряд преимуществ:

− Удобство работы с данными. Excel предоставляет широкие возможности для работы с данными, такие как фильтрация, сортировка, поиск, расчеты и др. Это позволяет быстро и удобно анализировать, и обрабатывать большие объемы информации.

− Возможность создания отчетов и дашбордов. Excel позволяет создавать различные отчеты и дашборды на основе импортированных данных. Это позволяет быстро и наглядно отображать изменения параметров объектов во времени и выявлять тенденции.

− Универсальность формата. Excel является универсальным форматом для обмена данными между различными приложениями и системами. Это позволяет легко обмениваться данными с другими пользователями и системами.

# Заключение

Создание информационной системы конфигурирования объектов телеметрии является сложным и многоэтапным процессом, который требует использования современных технологий и методов разработки программного обеспечения. Основная цель этого модуля – обеспечить максимальную точность и надежность сбора и анализа данных, а также упростить процесс управления системой.

В процессе разработки модуля необходимо провести анализ требований и определить функциональные возможности системы. Также необходимо определить архитектуру модуля, выбрать подходящие технологии и инструменты для его разработки, а также провести тестирование и сопровождение.

Важным этапом разработки является создание графического интерфейса пользователя, который должен быть интуитивно понятным и удобным в использовании. Также необходимо обеспечить поддержку различных протоколов связи и форматов данных, чтобы модуль мог работать с различными типами объектов телеметрии.

Одной из ключевых задач модуля является настройка параметров сбора данных, которая может быть выполнена как вручную, так и автоматически на основе анализа полученных данных.

Разработка модуля информационной системы конфигурирования объектов телеметрии является важным шагом в создании современной и эффективной системы мониторинга и управления объектами телеметрии. Его успешная реализация позволит значительно повысить точность и надежность сбора и анализа данных, а также упростить процесс настройки и управления системой.

Успешная разработка модуля информационной системы конфигурирования объектов телеметрии может значительно повысить эффективность управления объектами телеметрии.

# Список используемых источников

1. ГОСТ 19.201-78 Единая система программной документации. Техническое задание. Требования к содержанию и оформлению. ГОСТ 34.602-89 Информационная технология. Комплекс стандартов на автоматизированные системы.

2. Гриффитс Дэвид. Vue.js: Up and Running: Building Accessible and Performant Web Apps / David Griffiths. - Sebastopol, CA: O'Reilly Media, 2018. - 174 с.

3. Шоу Томас. Vue.js in Action / Thomas Shaw. - Shelter Island, NY: Manning Publications, 2018. - 304 с.

4. Лью Эван. Vue.js: Understanding its Tools and Ecosystem / Evan Leu. - Sebastopol, CA: O'Reilly Media, 2018. - 180 с.

5. Wall, C. Spring Boot в действии / Крейг Уоллс. - М.: ДМК Пресс, 2017. - 496 с.

6. Felippe, W. Spring Boot: A Practical Guide for Beginners to Learn Quickly / Wilson Felippe. - Independently published, 2019. - 144 с.

7. Канер К., Фолк Дж., Нгуен Х. Тестирование программного обеспечения. - СПб.: Питер, 2010. - 672 с.

8. Криспин Л., Грегори Дж. Agile Testing: практическое руководство для тестировщиков и команд Agile. - М.: ООО "Издательство "Эком", 2011. - 576 с.

9. Копленд Л. Проектирование тестов программного обеспечения: Практическое руководство. - М.: ДМК Пресс, 2008. - 400 с.

10. Брош Э., Дуглас К. Проектирование объектно-ориентированных приложений с использованием UML. - СПб.: Питер, 2018. - 480 с.

11. Шнейдер Г. Проектирование баз данных: практикум на SQL Server. - М.: ООО "Издательство "Эком", 2012. - 304 с.

12. Розенберг Дж., Стивенсон М. Проектирование объектно-ориентированных систем: методы и паттерны. - СПб.: Питер, 2019. - 544 с.

# Приложение А – Техническое задание на разработку информационной системы

**1. Общие сведения**

**1.1 Наименование информационной системы**

Информационная система конфигурирования объектов телеметрии: ИС «КРОН-ТМ»

**1.2 Перечень документов, на основании которых создается ИС**

ГОСТ 34.602-89 Техническое задание на создание автоматизированной системы.

**1.3 Цели и назначение создания автоматизированной системы**

Информационные системы по контролю данных телеметрических систем стали необходимостью на предприятиях для обеспечения получения, преобразования, передачу по каналу связи, приём, обработку и регистрацию измерительной информации и информации о различных событиях с целью контроля на расстоянии различных объектов и процессов.

**1.4 Наименование организация – Заказчика и Разработчика**

**1.4.1 Заказчик**

Начальник отдела производственного программного обеспечения ООО «СиДиАй» Маланьин М.В.

**1.4.2 Разработчик**

Студент группы ИС-192 Камалов А.Р.

**1.5 Плановые сроки начала и окончания работы**

Разработка ИС «КРОН-ТМ»: 01.12.2022 – 01.04.2023

Тестирование ИС «КРОН-ТМ»: 01.04.2023 – 01.05.2023

Ввод в эксплуатацию ИС «КРОН-ТМ»: 01.05.2023

**1.6 Общие сведения об источниках и порядке финансирования работ**

Финансирование разработки технического задания, рабочего проекта, программирование, наладку и поставку оборудования производится Разработчиком.

**2 Назначение и цели использования информационной системы**

**2.1 Назначение информационной системы**

ИС «КРОН-ТМ» предназначена для контроля данных телеметрических систем.

**2.2 Цели создания информационной системы**

Для оперативного контроля объектов телеметрии на расстоянии необходимо разработать информационную систему, которая позволит:

* сбор данных с объектов
* обработка и систематизация накопленной информации, её анализ и отображение в наглядном виде
* ведение истории хода технологического процесса и событий системы
* формирование и выдача аварийных сигналов
* диагностика оборудования, состояния связи с объектами.
* передача управляющих воздействий оператора на технологический процесс
* контроль корректности получаемой информации.

**3 Требования к системе**

Backend

* Реализовать авторизацию
* Реализовать доступ по ролям (администратор - доступна настройка объектов, пользователь - доступен только просмотр)
* CRUD по объектам
* CRUD по типу объектов
* CRUD по свойствам объектов
* CRUD по типу событий объектов
* Получение архива по всем свойствам с временным промежутком
* Фильтрация по времени в архиве
* Хранение данных в БД

Frontend

Верхняя шапка для всех страниц кроме страницы авторизации:

* Мониторинг объектов
* Все объекты
* События
* Архив значений
* Тренды

Указать в шапке Логин пользователя и кнопку выхода.

* Страницы информационной системы:
* Страница авторизации
* Страница мониторинга объектов (отобразить название объекта, показатели, последние событие, дату события). Реализовать фильтрацию по типам объектов
* Страница просмотра объектов - список всех объектов. Возможность выбрать объект и вывод информации по нему и его показатели
* На выбранном объекте реализовать кнопку “Настройка объекта”. При ее нажатии - появление диалогового окна, где можно настроить свойства.
* Страница просмотра всех событий (вывод информаций по событию, возможность указывать временной промежуток)
* Страница архив значений

Шахматка жидкости - вывод по дням показателя количества жидкости.

Двухчасовки - вывод по часам показателей количества жидкости.

Отбраковка замеров - вывод информации по отбракованным показателям.

* Страница тренды. Отображение в виде графиков архива значений по свойствам. Возможность указать конкретные свойства и задать временной промежуток.

# Приложение Б – Инструкция пользователя и программиста

**Инструкция для программиста**

Для запуска информационной системы необходимо предварительно установить на персональный компьютер базу данных MySql. После установки базы данных воспользоваться бэкапом базы данных kron-tm-api.sql. Рекомендуется разворачивать проект через IDE Intellij IDEA, он определит недостающие зависимости и установит их. В случае использования другого редактора кода необходимо предварительно установить Java версии 19 и jdk версии 19. Открыть папку server в IDE. Для запуска сервера необходимо нажать кнопку Run.



Рисунок 21 – Кнопка Run в Intellij IDEA

Для развертывания фронтенд части приложение необходимо открыть в новом окне папку client и в командой строке ввести команду npm run dev.

Данные для входа в учетную запись с правами администратора: логин: admin, пароль: qweasd.

**Инструкция для пользователя**

Открывается форма авторизации в информационную систему

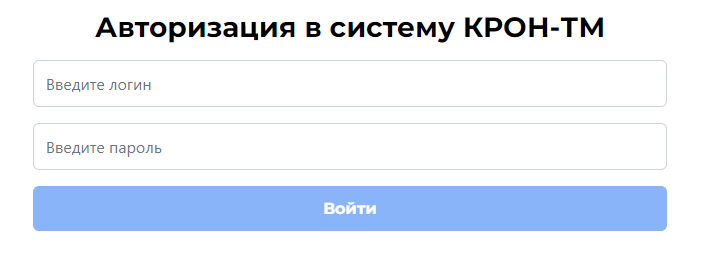


Рисунок 22 – Форма авторизации

Данные для входа в учетную запись с правами администратора: логин: user, пароль: qweasd.

После авторизации открывается главная форма Мониторинг свойств

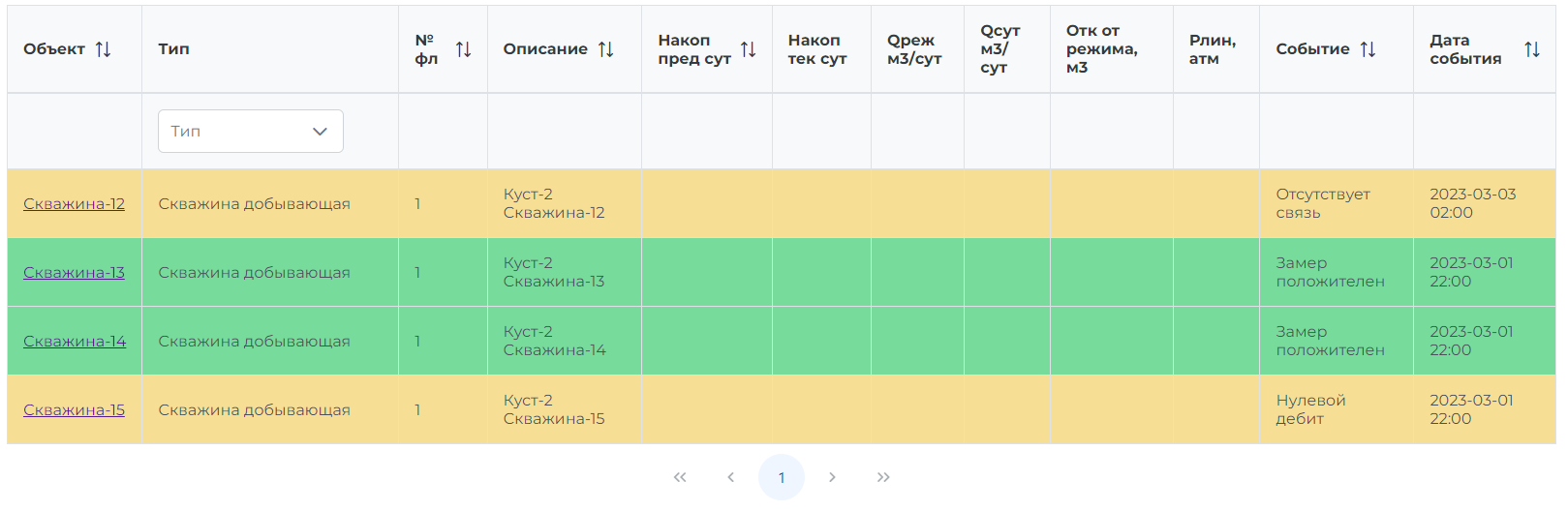


Рисунок 23 – Мониторинг свойств

Сверху формы расположена панель для навигации между формами.



Рисунок 24 – Панель навигации

При нажатии на объект открывается форма объекта

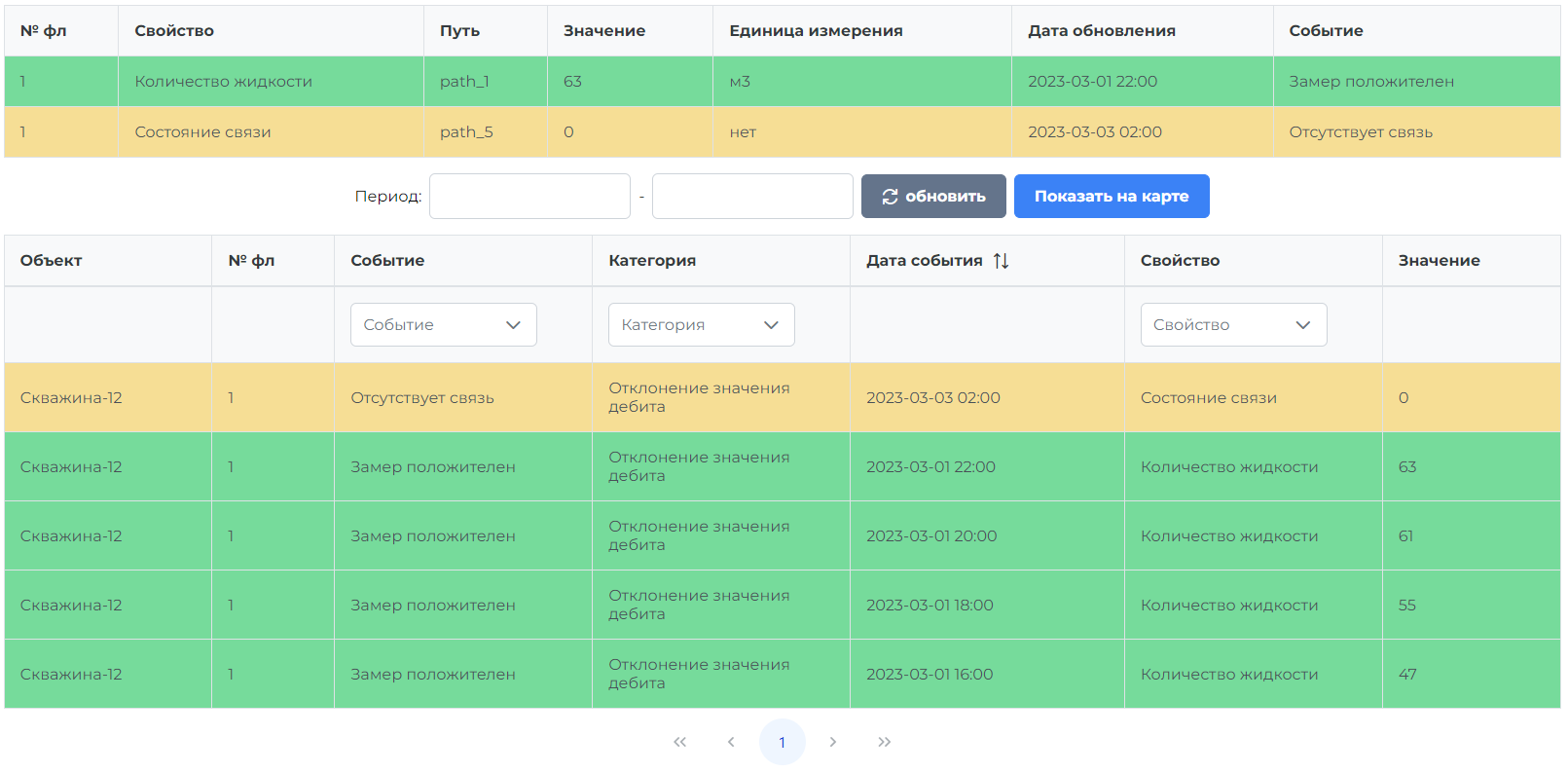


Рисунок 25 – Форма объекта

# Приложение В – Электронный формат пояснительной записки и информационной системы