

МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра МО ЭВМ

КУРСОВАЯ РАБОТА (КУРСОВОЙ ПРОЕКТ)
по дисциплине «Введение в нереляционные базы данных»
Тема: Создание сервиса климат-контроля

Студенты гр. 0381

Преподаватель

Березовская В.В.

Ионина К.С.

Заславский М.М.

Санкт-Петербург

2023

ЗАДАНИЕ НА КУРСОВУЮ РАБОТУ (КУРСОВОЙ ПРОЕКТ)

Студенты Березовская В.В.

Ионина К.С.

Группа 0381

Тема работы (проекта): Создание сервиса климат-контроля

Исходные данные:

Создать сервис для сбора данных о климате в протяженном помещении, управлении климатической техникой, мониторинге ее состояния. Необходимый (но не достаточный функционал) - визуализация состояния помещений, страницы приборов, настройки стратегий работы приборов, уведомления о расходниках.

Содержание пояснительной записки:

Содержание пояснительной записки:

«Содержание»

«Введение»

«Сценарий использования»

«Модель данных»

«Разработанное приложение»

«Выводы»

«Приложение»

«Литература»

Предполагаемый объем пояснительной записки:

Не менее 29 страниц.

Дата выдачи задания: 01.09.2023

Дата сдачи реферата: 25.12.2023

Дата защиты реферата: 25.12.2023

Студентка	_____	Березовская В.В.
Студентка	_____	Ионина К.С.
Преподаватель	_____	Заславский М.М.

АННОТАЦИЯ

Курсовой проект представляет собой разработку сервиса, предназначенного для сбора информации о климатических условиях в протяженных помещениях, управления климатической техникой и непрерывного мониторинга ее состояния. Результат работы – обеспечение пользователя инструментами для контроля и управления климатом в помещениях. В проекте будет использоваться система управления базами данных InfluxDBX. Проект будет реализован с использованием набора синтетических данных. Найти исходный код и всю дополнительную информацию можно по ссылке: <https://github.com/moevm/nosql2h23-climat>.

SUMMARY

The course project is the development of a service designed to collect information about climate conditions in large premises, control climate control equipment and ensure the continuity of its condition. The result of the work is the provision of user tools for monitoring and managing the indoor climate. The project will use the InfluxDB database management system. The project will be implemented using a synthetic data set. You can find the source code and all additional information at the link: <https://github.com/moevm/nosql2h23-climat>.

СОДЕРЖАНИЕ

1.	Введение	6
1.1.	Актуальность решаемой проблемы	
1.2.	Постановка задачи	
1.3.	Качественные требования к решению	
2.	Сценарии использования	10
2.1.	Макет UI	
2.2.	Сценарии использования для конкретной задачи	
2.3.	Вывод	
3.	Модель данных	22
3.1.	Нереляционная модель данных	
3.2.	Реляционная модель данных	
3.3.	Сравнение моделей данных	
4.	Разработанное приложение	30
4.1.	Краткое описание	
4.2.	Снимки экрана приложения	
5.	Выводы	31
5.1.	Достигнутые результаты	
5.2.	Будущее развитие приложения	
6.	Приложения	31
7.	Литература	31

1. ВВЕДЕНИЕ

Цель работы: разработка и создание сервиса, предназначенного для сбора информации о климатических условиях в протяженных помещениях, управления климатической техникой и непрерывного мониторинга ее состояния.

Функционал разрабатываемого сервиса включает в себя визуализацию текущего состояния помещений с использованием графических интерфейсов, создание страницы с информацией о работе и состоянии климатических приборов, возможность настройки стратегий и режимов функционирования оборудования, а также систему уведомлений, предупреждающих о необходимости замены расходных материалов или об ошибках в работе.

1.1 Актуальность решаемой проблемы

Создание сервиса для сбора данных о климате в протяженных помещениях, управления климатической техникой и мониторинга ее состояния - актуальная задача в наше время из-за повышенного интереса к энергоэффективности, улучшению качества воздуха внутри помещений, а также потребности в удаленном мониторинге и управлении. Сервис может позволить оптимизировать использование ресурсов и обеспечить возможность оперативного реагирования на потребности климатических систем, что важно для повышения комфорта и безопасности в различных типах помещений - от офисов до жилых домов и общественных пространств.

1.2 Постановка задачи

Необходимо создать сервис для сбора данных о климате в протяженном помещении, управлении климатической техникой, мониторинге ее состояния. Необходимый (но не достаточный функционал) - визуализация состояния помещений, страницы приборов, настройки стратегий работы приборов, уведомления о расходниках. В таблице 1 представлены требования верхнего уровня к разрабатываемой системе.

Таблица 1 — Перечень требований назначения

Идентификатор	Наименование требования	Примечание
CL-1	Система должна обеспечить сбор данных о климате в протяженных помещениях	
CL-2	Система должна обеспечить управление климатической техникой	
CL-3	Система должна мониторить состояние климатической техники	
CL-4	Система должна иметь возможность визуализации состояний помещений	
CL-5	Система должна иметь страницы приборов	
CL-6	Система должна иметь стратегии настройки приборов	Стратегия – набор из нескольких заранее придуманных обобщенных сценариев («Если показание выше N, включите прибор Ж»)
CL-7	Система должна иметь возможность просмотра и поиска по истории изменений состояния приборов с привязкой по времени.	
CL-8	Система должна иметь функционал массового импорта-экспорта	возможность импорта экспорта всех данных из системы в машино-читаемом формате (json, xml, csv на ваш выбор) - пользовательские интерфейсы для импорта и экспорта

		Зачем это нужно: - потренироваться в создании простого модуля бакапа - посмотреть, какие возможности предоставляет ваша СУБД для работы с дампами БД
CL-9	Система должна иметь возможность отправлять уведомления пользователям	В случае изменения параметров климата за пределами заданных порогов

1.3 Качественные требования к решению

В таблице 2 представлены качественные требования к решению.

Таблица 2 — Качественные требования к решению

Идентификатор	Наименование требования	Примечание
CL-10	Система должна обеспечить интуитивно понятный интерфейс для различных категорий пользователей с минимальным порогом входа	Удобство использования
CL-11	Система должна обеспечить возможность расширения функциональности (внедрений новых устройств/помещений)	Масштабируемость
CL-12	Система должна обеспечивать бесперебойную работу без сбоев или	Надежность и стабильность

	задержек.	
CL-13	Система должна быть разработана на базе СУБД InfluxDB	
CL-14	Система должна работать с синтетическими данными	

2. СЦЕНАРИИ ИСПОЛЬЗОВАНИЯ

2.1 Макет UI

На рисунке 1 изображен макет UI (user interface) — пользовательский интерфейс/

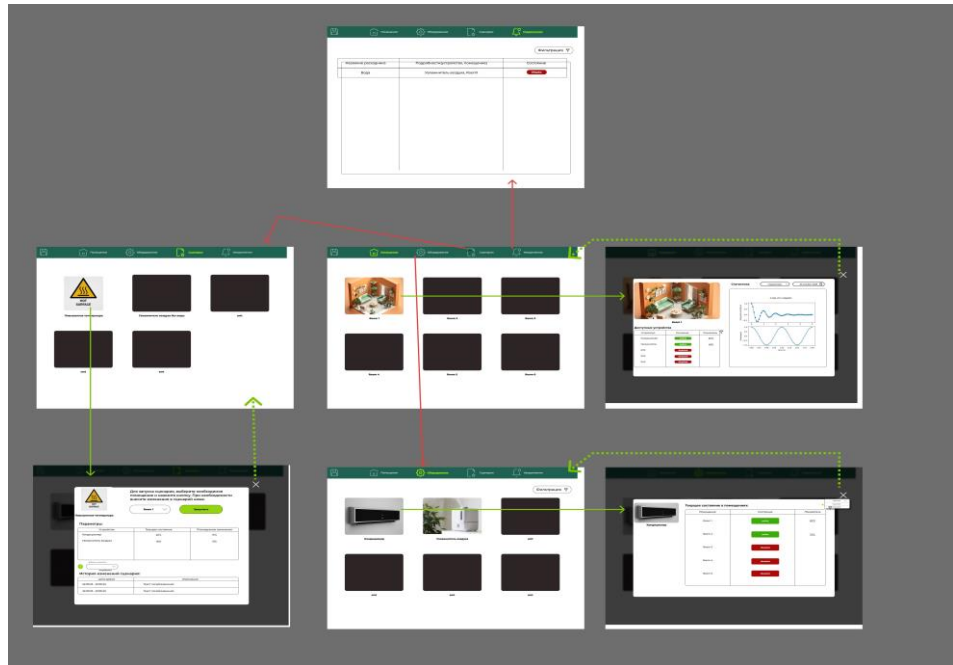


Рисунок 1 — Макет UI

2.2 Сценарии использования для конкретной задачи

На рисунке 1 представлены варианты использования при работе с программой.

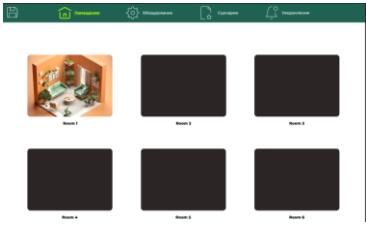



Рисунок 1 — Варианты использования при работе с программой

В таблица 3 — таблице 7 представлены сценарии использования разрабатываемого приложения (сервиса).

Представление данных (включает Импорт данных/ Экспорт данных)

Таблица 3 — UC-01.Работа со стартовым окном программы

UC-01.Работа со стартовым окном программы	
Первичное действующее лицо	Пользователь, имеющий доступ к Сервису климат-контроля (программе), далее по тексту Пользователь.
Предварительные условия	Пользователь запустил сервис
Описание	Пользователь просматривает содержимое стартового окна программы.
Выходные условия	Пользователь просмотрел содержимое стартового окна программы.
Нормальный поток развития 1 – Работа с начальным окном программы	<p>1. Система отображает стартовое окно программы</p>  <p>: ,</p> <p>содержащее следующие элементы:</p> <ul style="list-style-type: none"> • Панель проекта; • Коллекция контролируемых помещений. <p>Панель проекта содержит следующие элементы:</p> <p>Кнопка «Массовый импорт-экспорт»</p>  <p>— предназначена для возможности осуществления импорта/экспорта всех данных из системы в машино-читаемом формате. (Импорт данных/ Экспорт данных)</p>





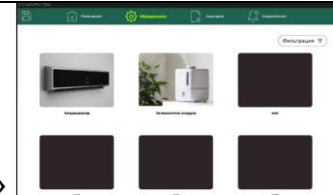
	<ul style="list-style-type: none"> • Кнопка «Помещения»  Помещения — при нажатии происходит переход в стартовое окно программы; • Кнопка «Оборудование»  Оборудование — при нажатии происходит переход к окну «Устройства» (см. УС-03); • Кнопка «Сценарии»  Сценарии — при нажатии происходит переход к окну «Сценарии» (см. УС-04); • Кнопка «Уведомления»  Уведомления — при нажатии происходит переход к окну «Уведомления о расходниках» (см. УС-05). <p>Коллекция контролируемых помещений содержит помещения, состояние которых возможно регулировать (при нажатии на них).</p>
--	--

Таблица 4 — УС-03.Контроль за состоянием оборудования

УС-03.Контроль за состоянием оборудования	
Первичное действующее лицо	Пользователь, имеющий доступ к Сервису климат-контроля (программе), далее по тексту Пользователь.
Предварительные условия	Пользователь на панели проекта нажал кнопку «Оборудование».
Выходные условия	Пользователь настроил параметры оборудования.
Нормальный поток развития 1 – Работа с состоянием помещени	1. Система отображает окно



«Устройства» ,
содержащие все установленные в
помещениях устройства.

2. Пользователь нажимает на область отображения устройств
3. Система отображает диалоговое окно, содержащее таблицу, столбцами которой являются:
 - «Помещение» – в данном столбце отображаются все помещения, в которых установлено устройство;
 - «Состояние» – индикатор, отображающий состояние устройства в данный момент («active» - устройство работает, «disabled»- устройство отключено)
 - «Показатель» – поле для отображения показателей устройств в помещении.
4. Пользователь нажимает на индикатор состояния устройства **disabled**
5. Система изменяет индикатор **active** на противоположный. (active -> disabled; disabled -> active).
6. Пользователь изменяет показатель устройства в конкретной комнате с помощью нажатия на редактируемое поле «Показатель»

Текущее состояние в помещениях:

Помещение	Состояние	Показатель
Room 1	active	22°C
Room 2	active	13°C

Пользователь с помощью ручного ввода с




	<p>клавиатуры своего устройства изменит температуру помещения в комнате: room2.</p> <p>Примечание. При изменении показателя устройства (температуры кондиционера), показатель в помещении (температура в помещении) изменится не сразу, а в течении 30 минут.</p> <p>7. Пользователь нажимает на  для закрытия окна.</p>
--	---

Таблица 5 — UC-04.Работа со стратегиями настройки приборов

UC-04.Работа со стратегиями настройки приборов	
Первичное действующее лицо	Пользователь, имеющий доступ к Сервису климат-контроля (программе), далее по тексту Пользователь.
Предварительные условия	Пользователь на панели проекта нажал кнопку «Сценарии».
Выходные условия	Пользователь активировал стратегию настройки прибора.
Нормальный поток развития 1 – Работа со стратегиями настройки приборов.	<p>1. Система отображает окно «Сценарии», содержащее стратегии настройки приборов.</p> <p>2. Пользователь нажимает на сценарий</p>  <p>Повышенная температура</p> <p>3. Система отображает диалоговое окно</p>  <p>содержащее:</p>

	<ul style="list-style-type: none"> • Выпадающий список Room1 - предназначен для выбора комнаты, в которой необходимо запустить стратегию настройки; • Кнопку «Запустить» - предназначена для активации стратегии в выбранном помещении; • Область отображения справочной информации для выбранной стратегии. • Таблицу «Параметры», столбцами которой являются: «Устройство» - наименование устройства, «Текущее состояние» - текущее состояние в помещении, «Планируемое изменение» - редактируемое поле для ввода желаемого показателя Текущего состояния. • Область для добавления нового устройства в таблицу «Параметры». Область содержит кнопку «Добавить» и выпадающий список «Добавить устройство». • Кнопку «Сортировка» с контекстным меню для выбора параметра сортировки. • Таблицу «История изменений сценария», столбцами которой являются: дата и время изменения сценария и введенное пользователем изменение. <p>4. Пользователь с помощью выпадающего списка выбирает необходимое помещение</p> <p>5. Пользователь в таблице «Параметры» вводит изменяет «Планируемое значение». (*по умолчанию</p>
--	---

	<p>планируемое значение равно текущему состоянию).</p> <p>6. Пользователь с помощью выпадающего «Добавить устройство» выбирает необходимое устройство и нажимает кнопку «Добавить».</p> <p>7. Система добавляет выбранное пользователем устройство в таблицу «Параметры».</p> <p>8. Пользователь нажимает кнопку «Сортировка», выбирает необходимый пункт в контекстном меню кнопки</p> <p>9. Система сортирует значения в таблице в соответствии с параметром, выбранным пользователем.</p> <p>10. Пользователь нажимает кнопку «Запустить».</p> <p>11. Система запускает настроенную стратегию, обновляет таблицу «История изменений сценария» в соответствии с измененными параметрами.</p> <p>12. UC-04 завершен.</p>
--	---

Таблица 6 — UC-05.Контроль за состоянием расходных материалов

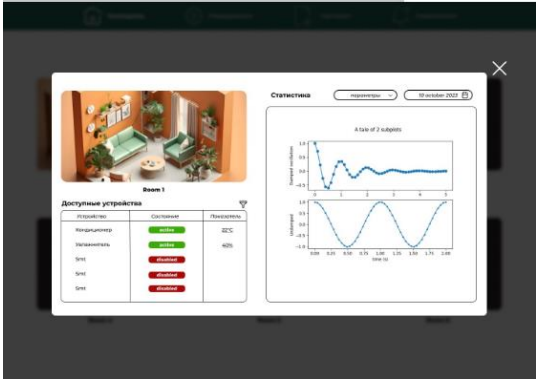
UC-05.Контроль за состоянием расходных материалов	
Первичное действующее лицо	Пользователь, имеющий доступ к Сервису климат-контроля (программе), далее по тексту Пользователь.
Предварительные условия	Пользователь на панели проекта нажал кнопку «Уведомления».
Выходные условия	Пользователь просмотрел уведомления.
Нормальный поток развития 1 – Работа с начальным окном программы	1. Система отображает окно «Уведомления о расходниках»

	<div data-bbox="890 150 1417 533"> </div> <p>содержащее таблицу, столбцами которой являются:</p> <ul style="list-style-type: none"> • «Название расходника» — поле для отображения названия расходника; • «Подробности» — поле для отображения устройства с помещением, для которого пришло уведомление. • «Состояние» — индикатор уведомления. <p>2. Пользователь просматривает уведомления Системы.</p> <p>3. UC-05 завершен.</p>
--	--

Анализ данных

Таблица 7 — UC-02.Работа с состоянием помещения

UC-02.Работа с состоянием помещения	
Первичное действующее лицо	Пользователь, имеющий доступ к Сервису климат-контроля (программе), далее по тексту Пользователь.
Предварительные условия	Пользователь на стартовом окне программы нажал на необходимое помещение 
Описание	Пользователь просматривает содержимое стартового окна программы.

Выходные условия	Пользователь просмотрел визуализацию состояния помещения, настроил состояние климатической техники в помещении (анализ данных)
Нормальный поток развития 1 – Работа с состоянием помещения	<p>1. Система отображает диалоговое окно «Состояние помещения»</p>  <p>содержащее следующие элементы:</p> <ul style="list-style-type: none"> • область «Доступные устройства» – см. Нормальный поток развития 2; • область «Статистика» – см. Нормальный поток развития 3. <p>2. Пользователь нажимает на  для закрытия диалогового окна.</p>
Нормальный поток развития 2 – Работа с доступными устройствами	<p>1. Система отображает таблицу доступных устройств, столбцами которой являются:</p> <ul style="list-style-type: none"> • Устройство – наименование устройства, установленного в помещении. • Состояние – индикатор, отображающий состояние устройства в данный момент («active» - устройство работает, «disabled»- устройство отключено.) • Показатель – поле для отображения показателей устройств в помещении. <p>2. Пользователь нажимает на индикатор состояния устройства</p>

	<div data-bbox="890 152 1225 215" data-label="Text"> <p>disabled</p> </div> <div data-bbox="839 226 1522 443" data-label="Text"> <p>3. Система изменяет индикатор на противоположный. (active -> disabled; disabled -> active).</p> </div> <div data-bbox="890 271 1254 336" data-label="Text"> <p>active</p> </div> <div data-bbox="839 454 1522 577" data-label="Text"> <p>4. Пользователь нажимает на кнопку «Сортировка»</p> </div> <div data-bbox="1129 506 1193 573" data-label="Image"> </div> <div data-bbox="839 584 1522 674" data-label="Text"> <p>5. Система отображает контекстное меню кнопки «Сортировка»</p> </div> <div data-bbox="887 680 1203 815" data-label="Image"> </div> <div data-bbox="839 824 1522 1014" data-label="Text"> <p>6. Пользователь выбирает необходимый параметр, Система сортирует значения таблицы. (по убыванию выбранного параметра, если значение числовое)</p> </div> <div data-bbox="839 1021 1158 1061" data-label="Text"> <p>7. UC-02 завершен.</p> </div>
<p>Нормальный поток развития 3 – Работа со статистикой помещения</p>	<div data-bbox="839 1077 1522 1160" data-label="Text"> <p>1. Система отображает панель с двумя графиками:</p> </div> <div data-bbox="839 1211 1522 1413" data-label="List-Group"> <ul style="list-style-type: none"> • график зависимости температуры воздуха от времени за текущие сутки; • график зависимости влажности воздуха от времени за текущие сутки; </div> <div data-bbox="887 1413 1321 1787" data-label="Figure"> </div> <div data-bbox="839 1832 1522 1933" data-label="Text"> <p>2. Пользователь нажимает кнопку «Параметры»</p> </div> <div data-bbox="1114 1877 1420 1926" data-label="Image"> </div> <div data-bbox="839 1944 1522 1991" data-label="Text"> <p>3. Система отображает контекстное</p> </div>

	<div data-bbox="1109 152 1385 342"> <div>параметры ▾</div> <div>температура ✓</div> <div>влажность воздуха ✓</div> <div>свет</div> <div>что-то ещё</div> </div> <p>меню кнопки</p> <ol style="list-style-type: none"> Пользователь выбирает два необходимых параметра. По выбранным Пользователем параметрам, Система отображает графики зависимости параметра от времени за текущие сутки. Оператор нажимает на область отображения текущей даты <div data-bbox="896 779 1228 828"> <div>10 october 2023</div> <div>📅</div> </div> и с помощью выпадающего списка (календарь) изменяет дату. *Даты, информация по статистике которых не известна, Система дизейблит. Система обновляет графики в соответствии с выбранной пользователем датой. UC-02 завершен.
--	--

2.3 Вывод

Исходя из вышеописанных сценариев для пользователя будет преобладать операция чтения (просмотр контента).

3. МОДЕЛЬ ДАННЫХ

3.1 Нереляционная модель данных

На рисунке 2 представлено графическое представление нереляционной модели.

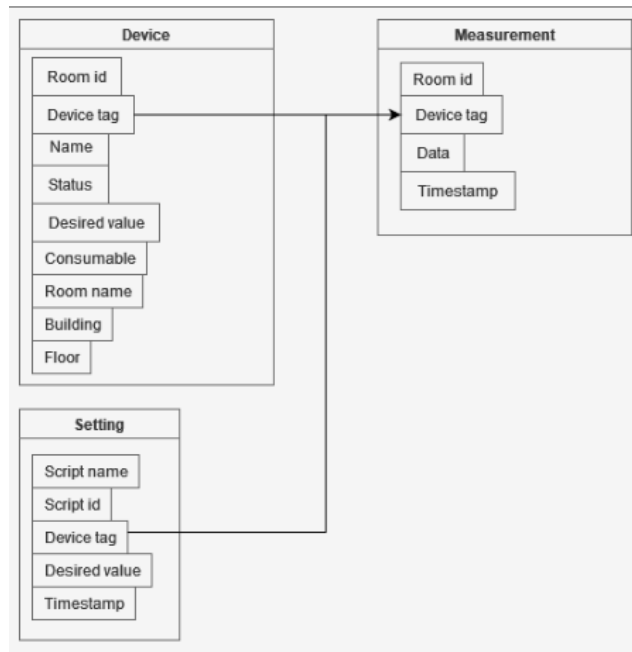


Рисунок 2 — Графическое представление модели

Описание назначений коллекций, типов данных и сущностей:

Коллекции:

1. Device — хранит информацию об устройствах

- Room id, int, 4B
- Room name, varchar(255), 255B
- Device tag, varchar(4), 4B
- Name, varchar(255), 255B
- Status, int, 4B
- Desired value, int, 4B
- Consumable, int, 4B
- Building, int, 4B
- Floor, int, 4B

2. Measurement — хранит измерения датчиков

- Device tag, varchar(4), 4B
- Room id, int, 4B
- Data, int, 4B
- Timestamp, int, 4B

3. Setting — хранит настройки для одного устройства

- Script name, varchar(255), 255B
- Script id, int, 4B
- Device tag, varchar(4), 4B
- Desired value, int, 4B
- Timestamp, int, 4B

Оценка удельного объема информации, хранимой в модели

Пусть имеется F устройств и частота записи показателей датчиков - раз в минуту.

- «Чистый» объём:
- «Device» — 538B
- «Measurement» — 16B
- «Setting» — 271B

Тогда общий объём БД при хранении данных за год: $(538+8409600+271)*F$ байт

Итог: $F*8\text{MB}$ данных.

Избыточность модели

На рисунке 3 изображены данные об избыточности модели.

$$\begin{aligned} \text{Чистый объем} &: 6 \times 267 + 10 \times 259 + (275 + 12 + 8 + 8935200) \times F \\ \text{Избыточность} &= \frac{4126 + F \times 8935495}{6 \times 267 + 10 \times 259 + (275 + 12 + 8 + 8935200) \times F} \end{aligned}$$

Рисунок 3 — Избыточность модели

Направление роста модели при увеличении количества объектов каждой сущности

При добавлении объекта в коллекцию Device за год работы может появиться до 525600 объектов в коллекции Measurement.

При добавлении объектов в коллекцию Setting никаких дополнительных объектов добавлено не будет.

Запросы к модели, с помощью которых реализуются сценарии использования

На рисунке 4 отображены запросы к модели, с помощью которых реализуются сценарии использования

UC-01.Работа со стартовым окном программы.

```
from(bucket: "test")
|> range(start: 0, stop: now())
|> filter(fn: (r) => r["_measurement"] == "Device")
|> group(columns: ["Room_name"])
|> distinct(column: "Room_name")
```

UC-02.Работа с состоянием помещения.

```
from(bucket: "test")
|> range(start: v.timeRangeStart, stop: v.timeRangeStop)
|> filter(fn: (r) => r["_measurement"] == "Measurement")
|> filter(fn: (r) => r["Room_id"] == "%selected_room%")
|> group(columns: ["Device_tag"])
```

UC-03.Контроль за состоянием оборудования.

```
from(bucket: "test")
|> range(start: v.timeRangeStart, stop: v.timeRangeStop)
|> filter(fn: (r) => r["_measurement"] == "Device")
|> filter(fn: (r) => r["Device_tag"] == "%selected_device_tag%")
```

UC-04.Работа со стратегиями настройки приборов.

```
from(bucket: "test")
|> range(start: v.timeRangeStart, stop: v.timeRangeStop)
|> filter(fn: (r) => r["_measurement"] == "Setting")
|> filter(fn: (r) => r["Device_tag"] == "%selected_device_tag%")
```

UC-05.Контроль за состоянием расходных материалов.

```
from(bucket: "test")
|> range(start: v.timeRangeStart, stop: v.timeRangeStop)
|> filter(fn: (r) => r["_measurement"] == "Device")
|> project(columns: ["Name", "Consumable"])
```

Рисунок 4 — Запросы к модели

3.2 Реляционная модель данных

На рисунке 5 представлено графическое представление нереляционной модели.

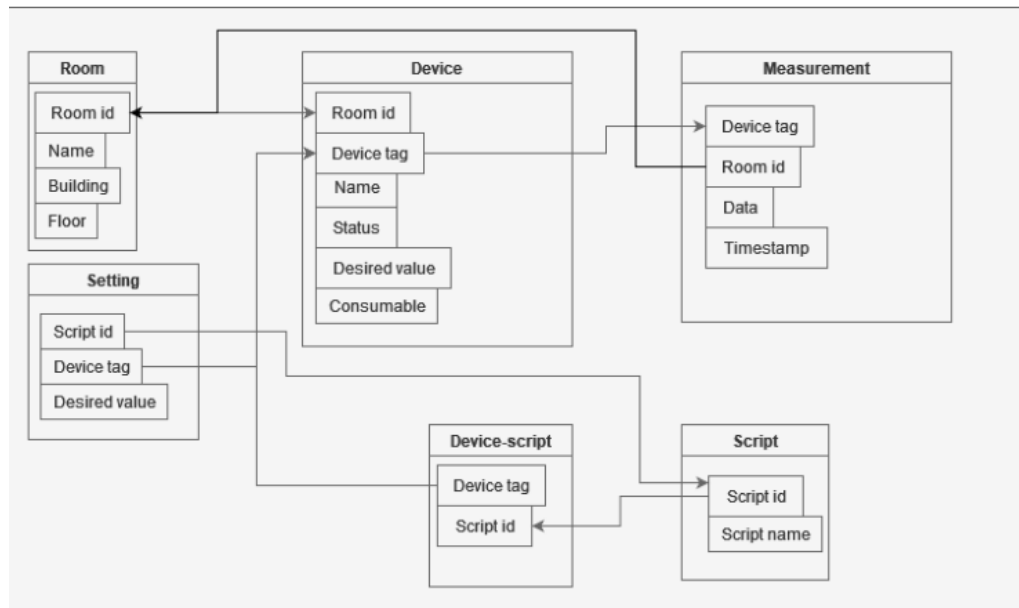


Рисунок 5 — Графическое представление модели

Описание назначений коллекций, типов данных и сущностей:

Таблицы:

1. Room — хранит информацию о помещениях

- Room id, int, 4B
- Name, varchar(255), 255B
- Floor, int, 4B
- Building, int, 4B

2. Device — хранит информацию об устройствах

- Room id, int, 4B
- Device tag, varchar(4), 4B
- Name, varchar(255), 255B
- Status, int, 4B
- Desired value, int, 4B
- Consumable, int, 4B

3. Measurement — хранит измерения датчиков

- Device tag, varchar(5), 5B
- Room id, int, 4B
- Data, int, 4B
- Timestamp, int, 4B

4. Setting — хранит настройки для одного устройства

- Script id, int, 4B
- Device tag, varchar(4), 4B
- Desired value, int, 4B

5. Script — хранит пресеты настроек

- Script id, int, 4B
- Script name, varchar(255), 255B

6. Device-script — хранит связи пресетов с устройствами

- Script id, int, 4B
- Device tag, varchar(4), 4B

Оценка удельного объема информации, хранимой в модели

На рисунке 6 изображен расчет оценки удельного объема информации, хранимой в модели

Пусть имеется F устройств и частота записи показателей датчиков - раз в минуту. Будем считать, что число помещений равно 6. Также будем считать, что число сценариев (Script) будет равно 10.

"Чистый" объем:

- "Room" -

$$(4 + 4 + 4 + 255) = 267B$$

259

- "Device" -

$$(4 + 4 + 255 + 4 + 4 + 4) = 275B$$

- "Measurement" - $5 + 4 + 4 + 4 + 4 = 21B$
- "Setting" -

$$(4 + 4 + 4) = 12B$$

- "Script" -

$$(4 + 255) = 259B$$

- "Device-script" - $4 + 4 = 8B$

$$[6 \times 267 + 10 \times 259 + (275 + 12 + 8 + 8935200) \times F] \text{ байт}$$

Итого:

$$[4192 + F \times 8935495] \text{ байт данных}$$

байт данных.

Рисунок 6 — Оценка удельного объема информации, хранимой в модели

Избыточность модели

На рисунке 7 изображены данные об избыточности модели.

$$\left[\frac{4192 + F \times 8935495}{6 \times 267 + 10 \times 259 + (275 + 12 + 8 + 8935200) \times F} \right]$$

Рисунок 7 — Избыточность модели

Направление роста модели при увеличении количества объектов каждой сущности

При добавлении объекта в коллекцию Device за год работы может появиться до 525600 объектов в коллекции Measurement.

При добавлении объекта в коллекцию Script будет создан объект в коллекции Device-script.

При добавлении объектов в коллекции Room, Measurement, Setting дополнительных объектов создаваться не будет.

Запросы к модели, с помощью которых реализуются сценарии использования

На рисунке 8 отображены запросы к модели, с помощью которых реализуются сценарии использования

UC-01.Работа со стартовым окном программы.

```
SELECT * FROM Room
```

UC-02.Работа с состоянием помещения.

```
SELECT * FROM Device
WHERE Room_id=%selected_room_id%

SELECT * from Measurement
WHERE Room_id=%selected_room_id%
GROUP BY Device_tag
```

UC-03.Контроль за состоянием оборудования.

```
SELECT * FROM Device
WHERE Device_tag=%selected_device_tag%
```

UC-04.Работа со стратегиями настройки приборов.

```
SELECT * FROM Script
WHERE Script_name=%selected_script_name%
```

```
SELECT * FROM Setting
WHERE script_id=%selected_script_id%
```

UC-05.Контроль за состоянием расходных материалов.

```
SELECT Name, Consumable FROM Device
```

Рисунок 8 — Запросы к модели

3.3 Сравнение моделей данных

Реляционная модель хранит информацию о помещениях в отдельной таблице, что уменьшает дублирование данных. Например в случае, когда есть 5 помещений и по 100 устройств в каждом из них, в реляционной модели названия комнат будут храниться всего 5 раз в таблице Room. В нереляционной модели названия комнат дублировались бы по 100 раз в каждом устройстве для каждой из 5 комнат.

В сценариях использования для нереляционной модели каждый сценарий требует выполнения одного запроса к одной таблице. В реляционной модели работа с настройками приборов требует двух запросов к двум таблицам.

По объему хранимых данных отличия моделей незначительны.

Для решаемой задачи приоритетнее выбрать NoSQL СУБД, так как позволяет более гибко хранить большие объёмы, а так же в отличие от реляционной модели чаще всего достаточно одного запроса для выполнения сценария. Данный фактор важнее, чем небольшой проигрыш в среднем объеме данных занимаемых данными в модели в сравнении с SQL СУБД.

4. РАЗРАБОТАННОЕ ПРИЛОЖЕНИЕ

4.1 Краткое описание

Back-end представляет собой программное обеспечение на основе JavaScript, осуществляющее обращения к базе данных с использованием скриптов. Front-end в свою очередь взаимодействует с API back-end приложения, реализованным с применением React.

4.2 Снимки экрана приложения

Снимки экрана приложения и переходы между ними изображены на рисунке 9.

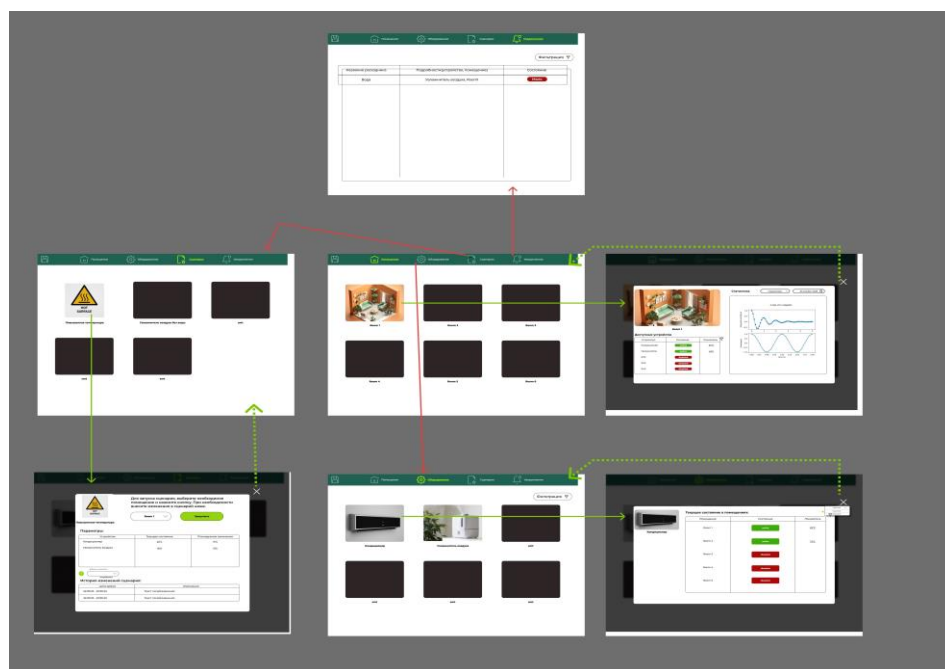


Рисунок 9 — Снимки экранов приложения

5. ВЫВОДЫ

5.1 Достигнутые результаты

В ходе работы был разработан сервис климат контроля, позволяющий просматривать содержимое БД с помощью таблиц и предоставляющий интерфейсы для поиска (фильтрации) данных.

5.2. Будущее развитие решения

Планируется разработка сценария авторизации пользователя, возможность добавления новых элементов данных в БД, а также разработка функционала импорта и экспорта всех данных приложения в машиночитаемом формате.

6. ПРИЛОЖЕНИЯ

Документация по сборке и разворачиванию сервиса

1. Скачать проект из репозитория (указан в п.7 «Литература»)
2. Запустить с помощью следующих команд:

```
npm run build && serve -s build
```

```
node server.js
```

Запустить базу данных с influxd.exe

7. ЛИТЕРАТУРА

1. Документация к InfluxDB <https://docs.influxdata.com/>
2. Ссылка на репозиторий проекта <https://github.com/moevm/nosql2h23-climat>