

ДЕРЖАВНИЙ УНІВЕРСИТЕТ ТЕЛЕКОМУНІКАЦІЙ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
Кафедра інженерії програмного забезпечення автоматизованих систем

Пояснювальна записка

до магістерської роботи
на ступінь вищої освіти магістр

на тему: **«ДОСЛІДЖЕННЯ ТА ПРОЕКТУВАННЯ ІНФОРМАЦІЙНОЇ
СИСТЕМИ ОБРОБКИ ТА ВІЗУАЛІЗАЦІЇ ДАНИХ НА ОСНОВІ ІОТ-
ПЛАТФОРМИ THINGSBOARD»**

Виконав: студент 6 курсу, групи ІСДМ-61
спеціальності 126 Інформаційні системи та технології
освітня програма «Інформаційні системи та технології»
(шифр і назва спеціальності)

_____ Джерелейко А.О. _____
(прізвище та ініціали)

Керівник _____ Тушич А.М. _____
(прізвище та ініціали)

Рецензент _____
(прізвище та ініціали)

Нормоконтроль _____ Чорна В.М. _____
(прізвище та ініціали)

НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

Кафедра Інженерії програмного забезпечення автоматизованих систем

Ступінь вищої освіти - «Магістр»

Спеціальність підготовки 126 Інформаційні системи та технології

Освітня програма «Інформаційні системи та технології»

ЗАТВЕРДЖУЮ

Завідувач кафедри ІСТ

К.П.Сторчак

“ _____ ” _____ 2021 року

З А В Д А Н Н Я НА МАГІСТЕРСЬКУ РОБОТУ СТУДЕНТУ

Джерелейко Артем Олександрович

(прізвище, ім'я, по батькові)

1. Тема роботи: «Дослідження та проектування інформаційної системи обробки та візуалізації даних на основі IoT-платформи Thingsboard»

Керівник роботи: Тушич Аліна Миколаївна, доктор філософії

(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом вищого навчального закладу від _____ року № _____

2. Строк подання студентом роботи _____

3. Вхідні дані до роботи :

1. Науково-технічна література

2. Дослідження обробки даних

3. Дослідження візуалізації даних

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити):

1. Аналіз та дослідження обробки та візуалізації даних

2. Проектування інформаційної системи на основі IoT-платформи Thingsboard

5. Перелік графічного матеріалу

1. Титульний слайд

2. Мета і завдання

3. Візуалізація даних

4. ThingsBoard

5. Архітектура інформаційної системи

6. Розробка інформаційної системи

7. Унікальність інформаційної системи

8. Висновки

6. Дата видачі завдання _____

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів магістерської роботи	Строк виконання етапів роботи	Примітка
1	Підбір науково-технічної літератури		
2	Вивчення матеріалів для подальшої взаємодії з ними		
3	Дослідження методів обробки та візуалізації даних		
4	Аналіз IoT-платформи Thingsboard		
5	Визначення технічного завдання		
6	Проектування інформаційної системи		
7	Вступ, висновки, реферат		
8	Розробка демонстраційних матеріалів		
9	Попередній захист роботи		

Студент _____ Джерелейко А.О.
(підпис) (прізвище та ініціали)

Керівник роботи _____ Тушич А.М.
(підпис) (прізвище та ініціали)

РЕФЕРАТ

Текстова частина магістерської роботи 70 с., 57 рис., 20 джерел.

Об'єкт дослідження – процес створення інформаційної системи обробки та візуалізації даних.

Предмет дослідження – обробка та візуалізація інформації.

Мета роботи – дослідити методи обробки та візуалізації інформації та проектування інформаційної системи на досліджених результатах.

Методи дослідження – методи теорії інформації, методи наукового моделювання, методи дослідження інформаційних систем.

Проведено дослідження методів обробки та візуалізації даних, визначено їх цілі, типи та задачі які вони вирішують.

Досліджено та проведено огляд частини вбудованого функціоналу IoT-платформи Thingsboard.

На основі результатів виконаних досліджень було розроблено інформаційну систему обробки та візуалізації інформації, яка надає можливість відображати опрацьовані дані та працювати з ними.

ІОТ, ІНТЕРНЕТ РЕЧЕЙ, ІОТ – ПЛАТФОРМА, ІНФОРМАЦІЙНА СИСТЕМА, ВІЗУАЛІЗАЦІЯ, ОБРОБКА ДАНИХ.

ЗМІСТ

ВСТУП.....	10
1 ОГЛЯД ПРЕДМЕТНОЇ ОБЛАСТІ.....	11
1.1 Візуалізація даних	11
1.1.1 Використання візуалізації даних	13
1.1.2 Табличне представлення даних	22
1.1.3 Використання графіків	25
1.1.4 Оформлення графіків.....	25
1.1.5 Вимоги до систем візуалізації даних у IoT	26
1.2 Обробка даних	28
1.2.1 Цілі, завдання та види обробки інформації.....	30
1.2.2 Властивості інформації	31
1.2.3 Методи обробки даних	33
1.2.4 Технічні засоби обробки інформації.....	36
1.2.5 Засоби прийому та передачі інформації.	37
2 ОПИС ТЕХНОЛОГІЙ ТА ЗАСОБІВ	39
2.1 IoT платформа	39
2.1.1 Технічні характеристики платформи	40
2.1.2 IoT платформа для Інтернету Речей.....	41
2.1.3 Архітектура Інтернет речей	42
2.1.4 Технічні характеристики	44
2.1.5 Практичні приклади застосування	45
2.2 Дослідження IoT платформи ThingsBoard.....	45

2.2.1 Архітектура платформи.....	46
2.2.2 Підключення пристроїв до платформи.....	49
3 РОЗРОБКА ІНФОРМАЦІЙНОЇ СИСТЕМИ.....	51
3.1 Розробка віджетів для інформаційної системи	51
3.3 Розробка схеми інформаційної системи	58
3.2 Проектування ланцюгів правил.....	66
3.4 Опис роботи інформаційної системи	72
ВИСНОВОКИ	77
ПЕРЕЛІК ПОСИЛАНЬ.....	79
ДЕМОНСТРАЦІЙНІ МАТЕРІАЛИ	81

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

API	Прикладний програмний інтерфейс
REST	Representational State Transfer
HTTP	HyperText Transfer Protocol
IoT	Internet Of Things
EOM	Електронна обчислювальна машина
JSON	JavaScript Object Notation
HTML	HyperText Markup Language
CSS	Cascading Style Sheets
JS	Java Script
Thingsboard CE	Thingsboard Customer Edition

ВСТУП

Сучасність потребує високих швидкостей обробки інформації, зручних форм її зберігання, відображення та передачі. Необхідно також мати динамічні способи звернення до інформації, способи пошуку даних можливість задавати часові інтервали, можливість реалізовувати складну математичну та логічну обробку даних.

Будь-яка діяльність людини є процес збору та обробки інформації, прийняття на її основі рішень та їх виконання. З появою сучасних засобів обчислювальної техніки інформація стала виступати як один з найважливіших ресурсів науково-технічного прогресу. Дані - це знання, виражені в сигналах та повідомленнях. Кожну людину у світі оточує море інформації різних видів.

У цій роботі визначаються вимоги, технологічні та програмні засоби для побудови інформаційної системи обробки та візуалізації даних, досліджуються вже існуючі методи обробки та візуалізації даних та методи їх покращення.

Ціль даної роботи провести аналіз і розробку інформаційної системи обробки та візуалізації даних, яку можна буде використовувати в повсякденних цілях або та підприємстві.

Задачі поставлені в роботі – дослідження теоретичного матеріалу, який дозволить побудувати інформаційну систему.

Кінцевою метою є представлення інформаційної системи яка опрацьовує та візуалізовує дані, для комфортної та правильної роботи з ними операторів або звичайних користувачів.

1 ОГЛЯД ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Візуалізація даних

За словами медиків, що на інструкції з застосування до ліків містить лише текст, а людина отримує інформацію з неї лише 70% інформації. Якщо ж додати до інструкції картинки, то людина вже розуміє 95% випадків.

Візуалізація даних – це представлення масивів інформації в комфортній для сприймання людині формі. Існують такі типи візуалізації:

- Просте візуальне подання кількісної інформації за допомогою схематичних форм. До цих груп можна віднести всім знайомі кругові та лінійні діаграми, гістограми та спектрограми, таблиці та різні точкові графіки.;
- Дані які візуалізуються можуть бути приведені до форми, що покращує сприйняття та аналіз такої інформації. Приклад: карта, тимчасова лінія, полярний графік, графік з паралельними осями та діаграма Ейлера;
- Концептуальна візуалізація допомагає розробляти складні ідеї, концепції за допомогою діаграм Ганта, концептуальних карт, графів із мінімальним шляхом та інших видів діаграм;
- Стратегічна візуалізація приводить до візуальної форми різні дані про особливості роботи організацій. Це всілякі діаграми продуктивності, життєвого циклу та графіки структур підприємств;
- Структурну інформацію досить просто графічно організувати за допомогою дерев, пірамід та карт даних допоможе метафорична візуалізація, прикладом якої карти метрополітенів;
- Комбінована візуалізація об'єднує складні графіки в одну логічну схему.

Вже доведено, що візуальна інформація краще сприймається людиною і дозволяє ефективно та швидко донести користувачу головну мету інформації, яку

вона в собі несе. Сприйняття візуальної інформації фізіологічно є основною для людини. Дослідження, що підтверджують переваги візуалізації інформації що:

- У середовищі, де ми живемо близько 90% інформації про навколишній світ людина отримує за допомогою зору;
- В межах 70% рецепторів які відповідають за сенсорне знаходяться в очах;
- Половина нейронів головного мозку відповідають за обробку візуальної інформації;
- При роботі з візуальними даними на 19% менше використовується когнітивна функція головного мозку, яка відповідає за аналіз та обробку інформації;
- Продуктивність людини, яка працює з візуальною інформацією на 17% вища;
- На 4,5% краще згадуються елементи візуальної інформації;
- Приблизно у 60000 разів сприймання візуальної інформація швидше порівняно з текстовою;
- 10% інформації людина може запам'ятати з почутого, 20% – із текстової інформації що прочитала, та 80% – із побаченого графічного матеріалу;
- Людина краще виконує інструкцію, якщо вона описана разом з ілюстраціями на 323%;

Очевидно, що люди схильні краще обробляти візуальну інформацію. Крім швидкості обробки нашим мозком, візуалізація даних має ще кілька переваг:

- Акцентування уваги. За допомогою графіків можна легко звернути увагу користувача показники виділені червоним кольором (рис. 1.4);

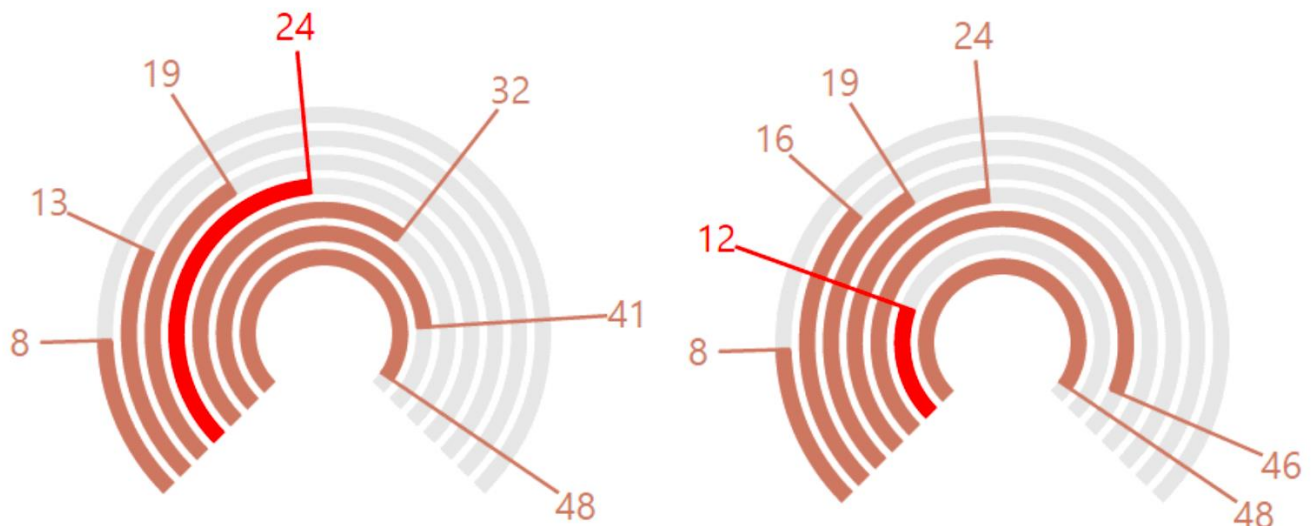


Рисунок 1.4 – Круговий графік

- Проаналізувати великий набір даних зі складною структурою;
- Утримування уваги людини та зменшення інформаційного навантаження;
- Ясність даних та однозначність, що виводяться;
- Виділення відносин та взаємозв'язків, які містяться в інформації;
- Естетика та привабливість даних;

Едвард Тафті, автор книг по візуалізаціях даних, описує це як інструмент для зображення даних; підштовхує глядача задуматися про суть, не методології подання; не спотворити те, що мають показати дані. "The Visual Display of Quantitative Information" Edward Tufte.

1.1.1 Використання візуалізації даних

Успіх візуалізації безпосередньо залежить від правильності її застосування, саме від вибору типу графіка, його вірного використання та оформлення.

Графіки дозволяють показати ідею, яку несуть в собі дані, точно та найповніше, саме тому дуже важливо вибрати відповідний тип діаграми, та не

помилитися з цим вибором. На рисунку 1.5 представлено алгоритм за яким можна легко здійснити вибір:



Рисунок 1.5 – Алгоритм вибору відповідного графіка

Ціль візуалізації інформації – реалізувати основну ідею інформації, тобто, для чого потрібно показувати відібрані дані, передати всю суть та якого ефекту слід добитися - показання розподілу даних, виявлення відносин в інформації, порівняння або композиції даних.

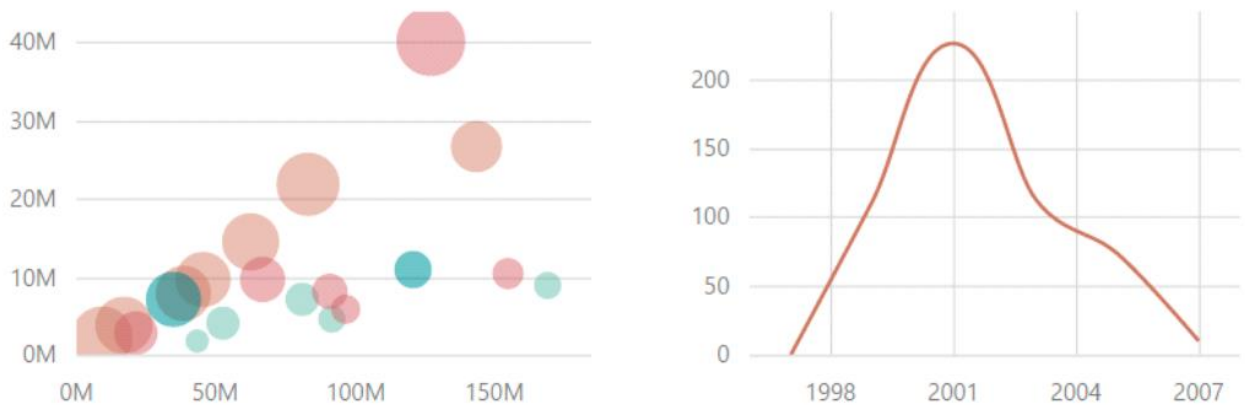


Рисунок 1.6 – Графіки з цілями показу відносин даних та розподілу даних

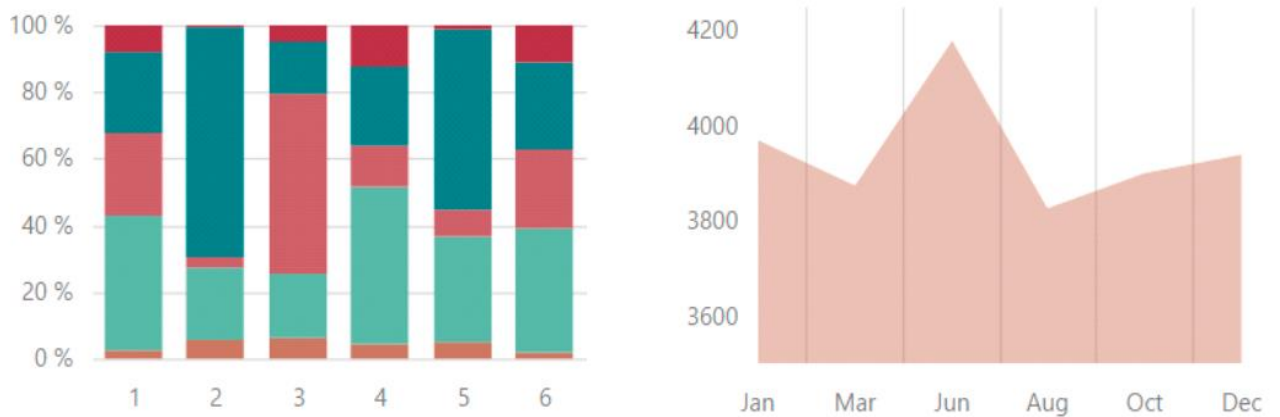


Рисунок 1.7 – Графіки композиції та порівняння даних

Відносини в даних – це зв'язок між ними, або краще сказати як вони залежать один від одного. Наявність чи відсутність залежностей між змінними виявляється за допомогою відносин. Якщо головна ідея інформації включає фрази "відноситься до", "знижується/підвищується", то слід прагнути відображати саме відносини в даних.

Розподіл даних – те, як дані розташовуються щодо чогось, скільки об'єктів відносяться до певної послідовності числових значень. Головна ідея при цьому міститиме фрази "концентрація", "в діапазоні від/до", "частотність", "розподіл".

Композиція даних – з'єднання даних з цілю аналізу загальної картини в цілому, порівняння компонентів, що становлять відсоток від цілого. Ключові фрази для композиції даних є "відсоток від цілого", "склала %", "частка".

Порівняння даних – об'єднання декількох груп даних для порівняльної характеристики деяких показників, виявлення того, як об'єкти можуть співвідноситися один з одним. Порівняння компонентів, які можуть змінюються з часом. Ключові фрази при порівнянні описуються словами - "підвищується/знижується", "більше/менше ніж", "рівно", "змінюється".

Після того як було визначено мету візуалізації потрібно визначити тип даних. Дані можуть за своєю структурою або типом бути дуже різними, але в найпростішому випадку виділяються часові дані та безперервні числові, логічні дані, дискретні дані, географічні.

Безперервні числові дані містять в собі інформацію залежності однієї числової величини від іншої, як приклад графік функцій $y=2x$.

Безперервні часові дані це дані про події, що відбувалися на якомусь проміжку часу, як приклад може слугувати графік температур, який вимірюється кожен день та зберігається.

Дискретні дані містять в собі залежності величин які відносяться до певних категорій, наприклад порівняльний графік графік продажу товарів у різних магазинах.

Географічні дані можуть містити в собі різну інформацію, пов'язану з географічними координатами або іншими даними які відносяться до розташування, геологією та іншими географічними показниками, наприклад - це географічна карта якою більшість з нас користується кожен день.

Логічні дані відображають логічне співвідношення компонентів щодо один одного, як приклад, генеалогічне дерево сім'ї.

Залежно від вхідних даних та мети можна вибрати найбільш підходящий графік. Правильним вибором буде вибрати за принципом чим простіше тим краще, а не експериментувати з різними графіками задля різноманітності. Тільки

В випадках коли використовуємо специфічні дані слід використовувати специфічні типи діаграм.

В більшості випадків добре підійдуть найпоширеніші графіки:

- Лінійні – Line (рис. 1.8);
- З областями – Area (рис. 1.9);
- Колонки та гістограми – Bar (рис. 1.10);
- Кругова діаграма – Pie, Doughnut (рис. 1.11);
- Полярний графік – Radar (рис. 1.12);
- Точковий графік – Scatter, Bubble (рис. 1.13);
- Карти – Map (рис. 1.14);
- Деревя – Tree, Mental map, Tree map;
- Часові діаграми – Time line, Gantt, Waterfall (рис. 1.15).

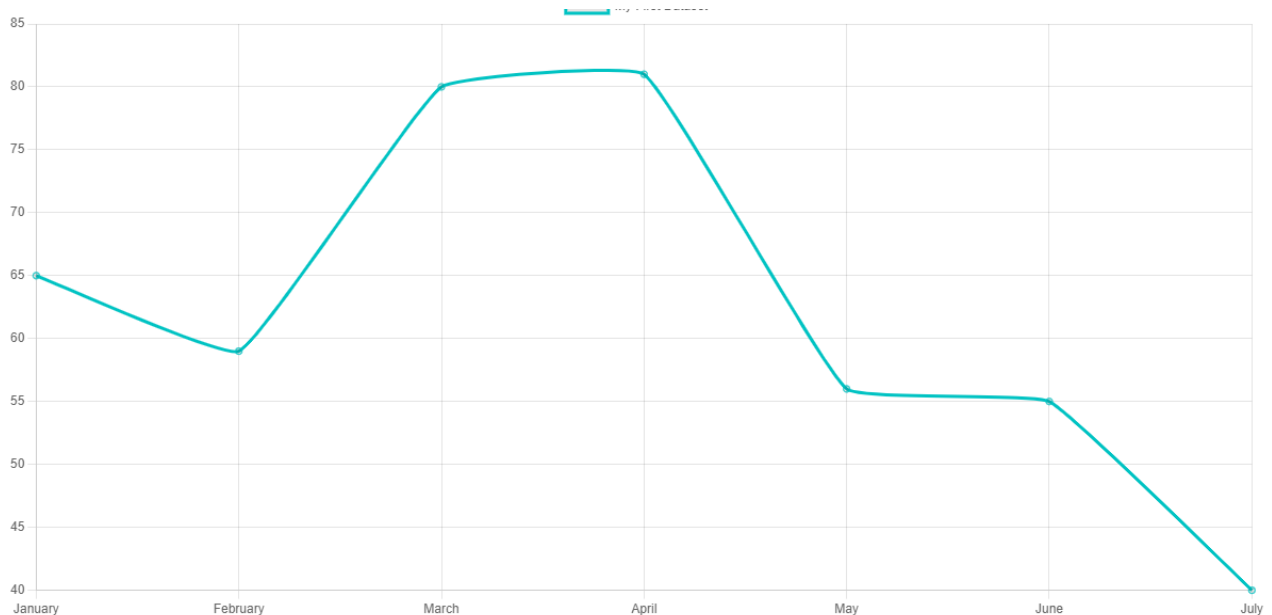


Рисунок 1.8 – Лінійний графік

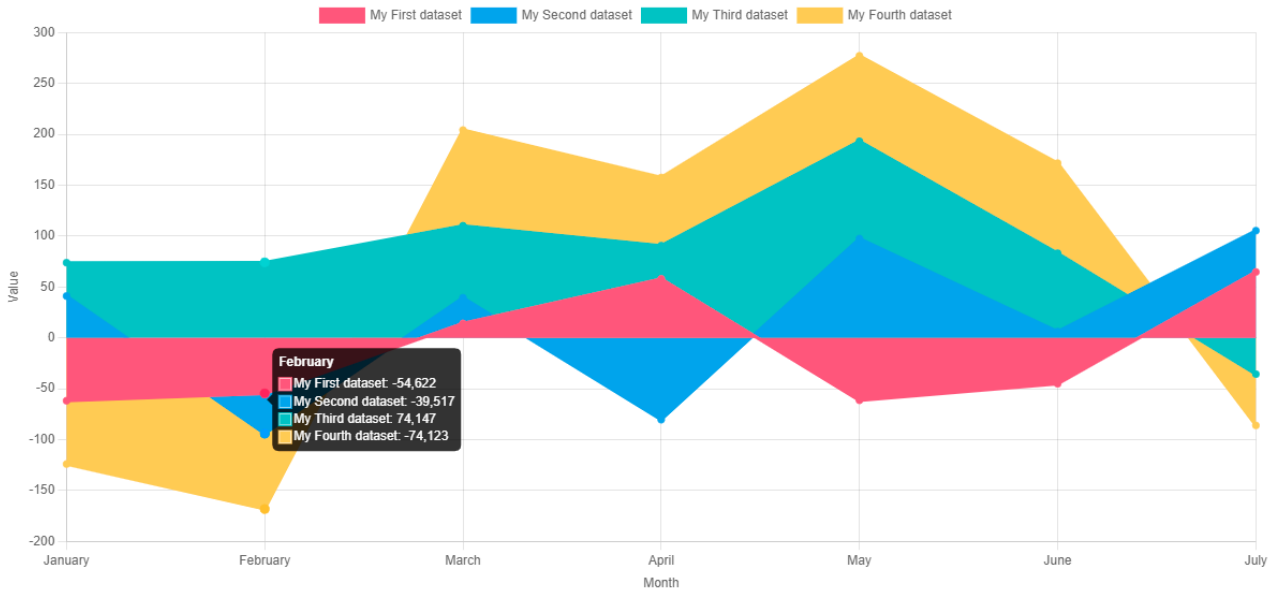


Рисунок 1.9 – Графік з областями

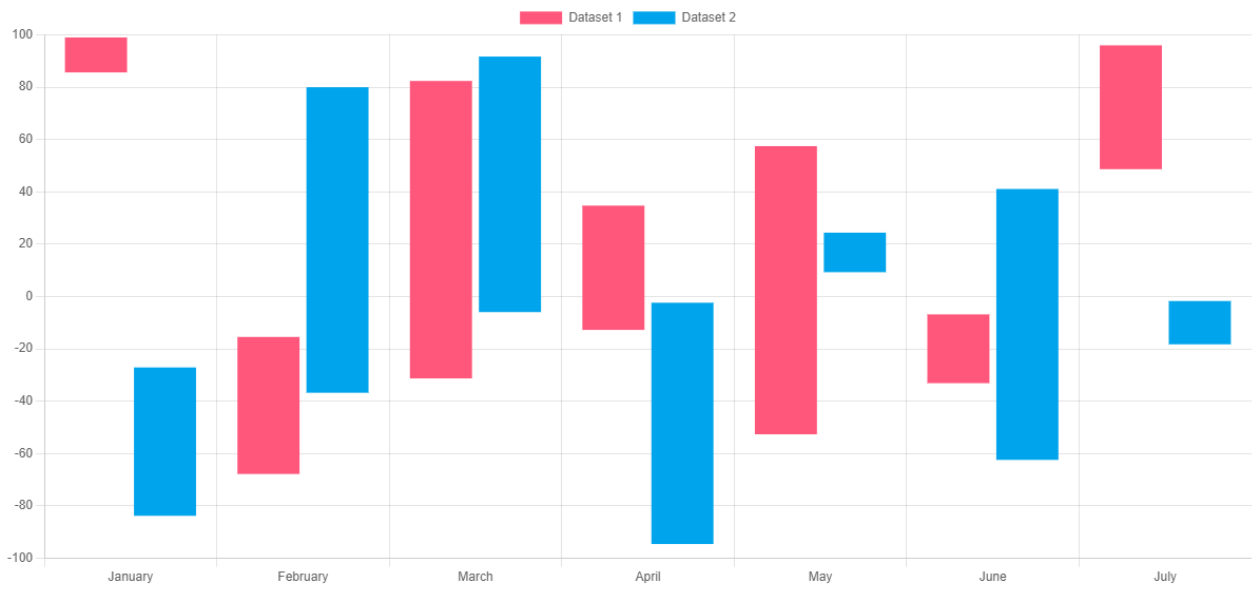


Рисунок 1.10 – Колонковий графік

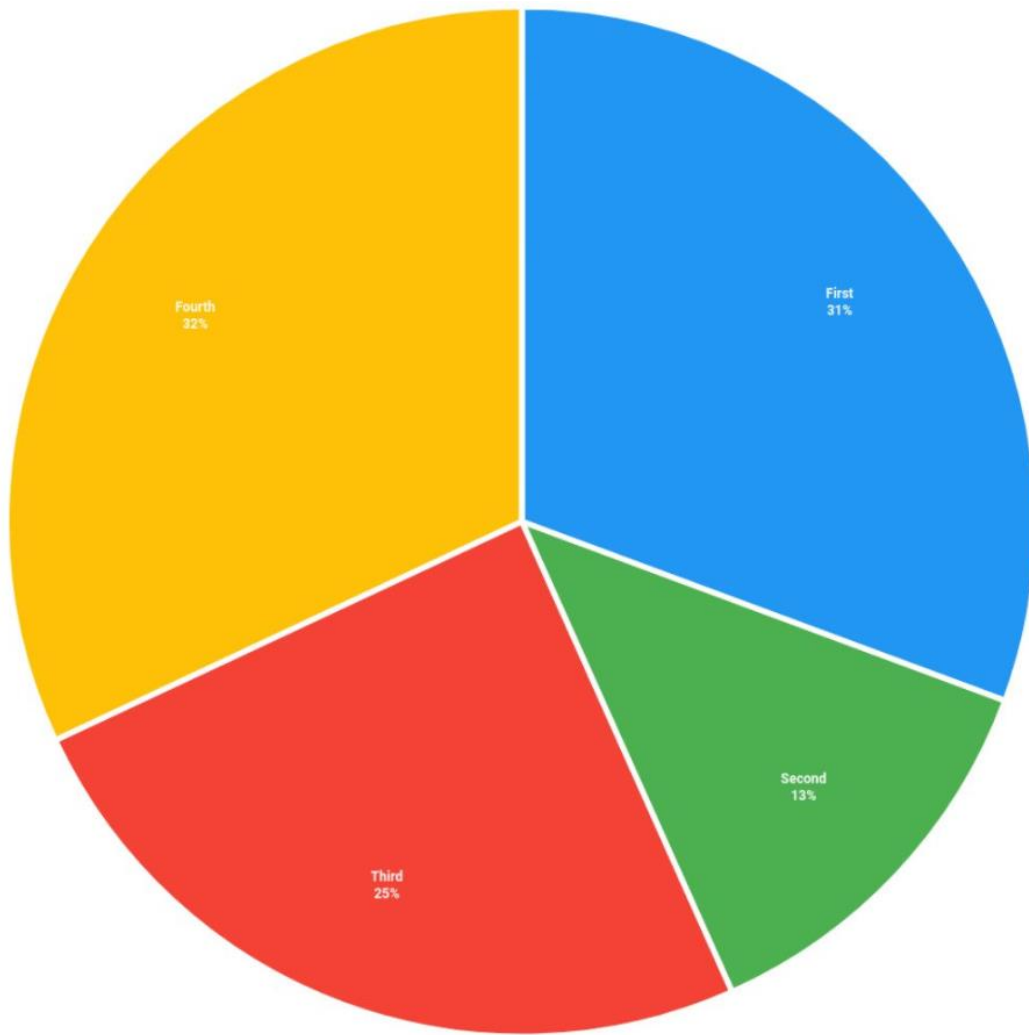


Рисунок 1.11 – Круговой графік

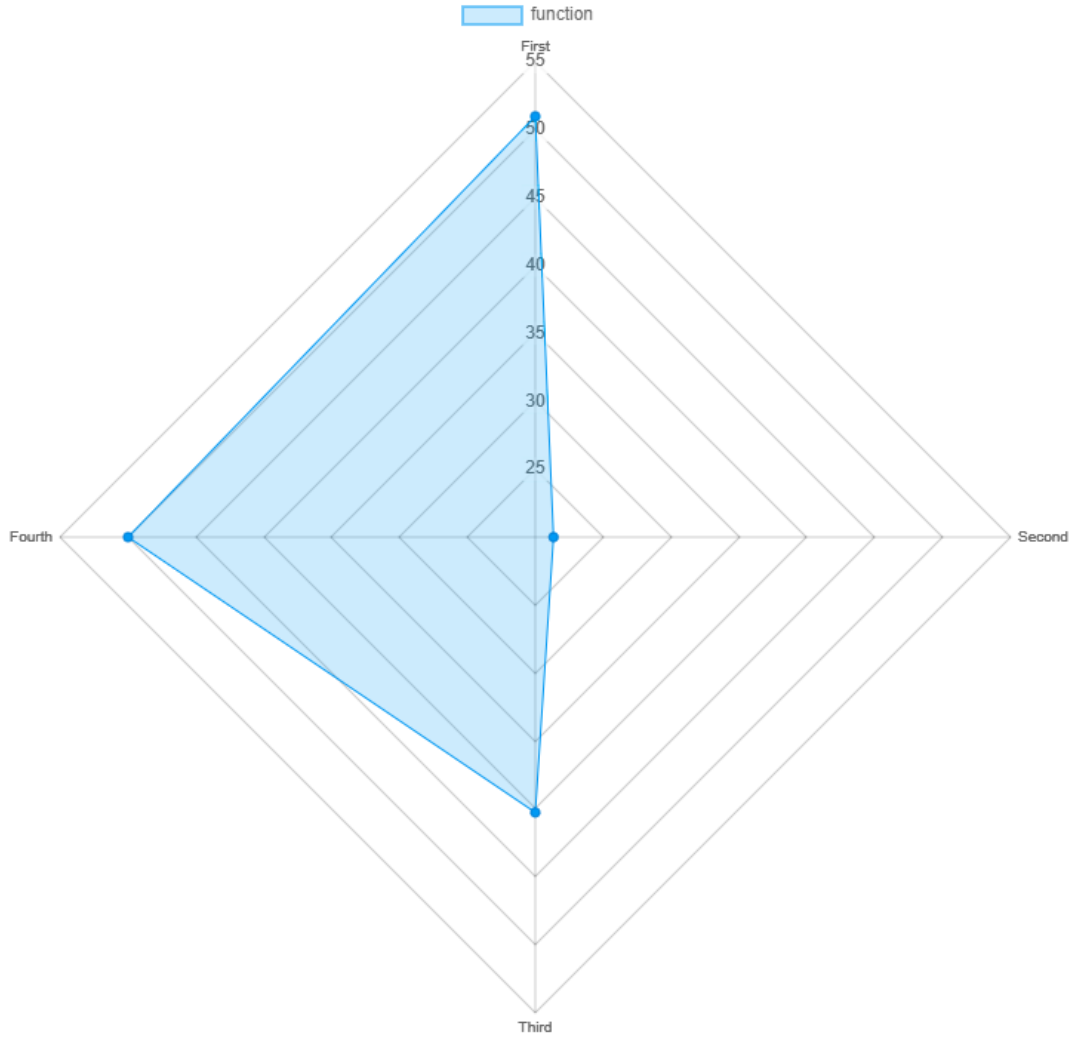


Рисунок 1.12 – Полярний графік

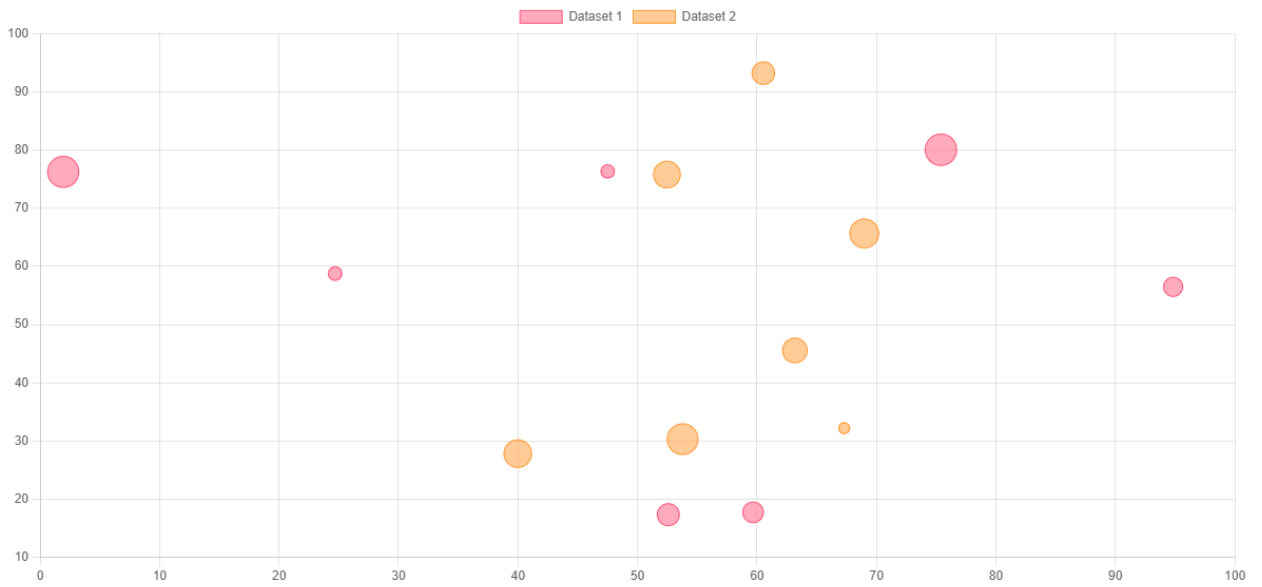


Рисунок 1.13 – Точковий графік

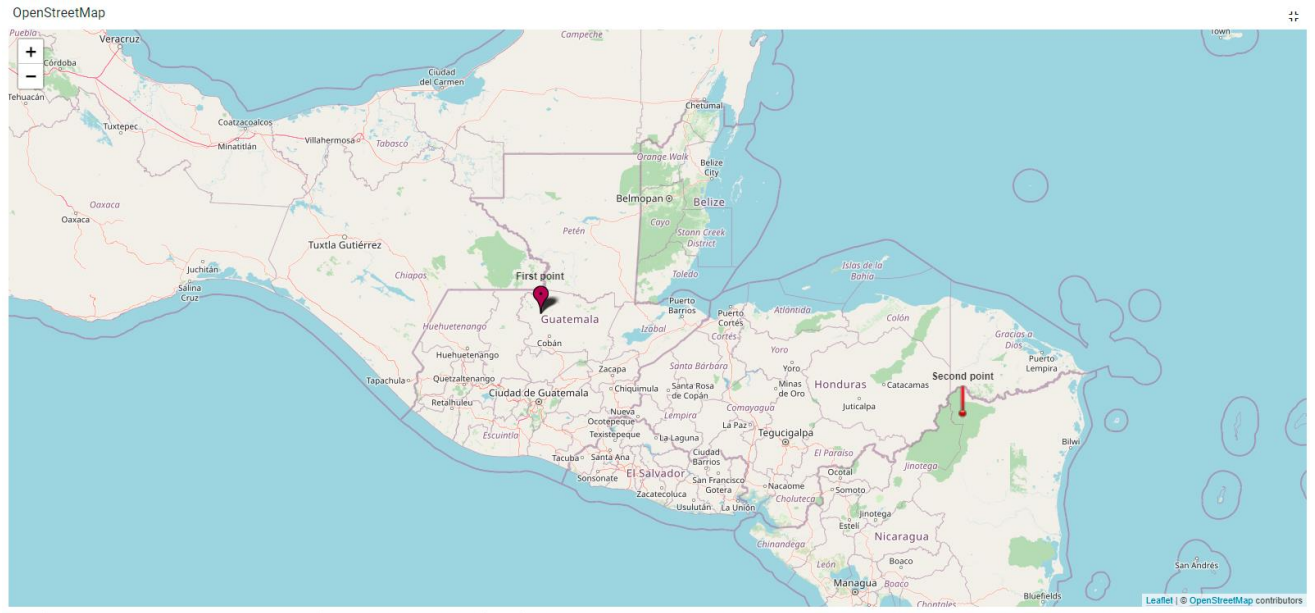


Рисунок 1.14 – Карта з маркерами

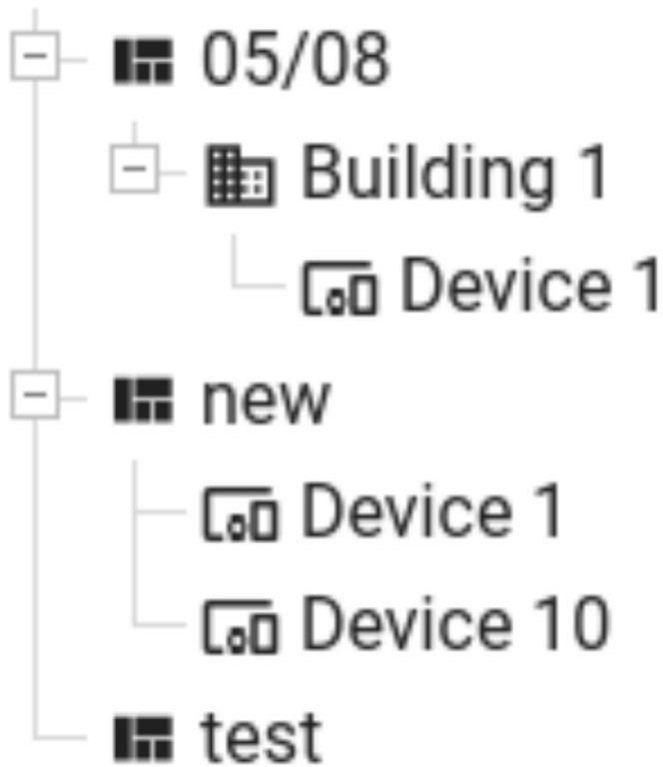


Рисунок 1.15 – Графік дерево



Рисунок 1.16 – Часовий графік

1.1.2 Табличне представлення даних

Графіки та таблиці в аналітичному звіті служать не лише ілюстрованими матеріалами, а містять у собі саму суть, ядро отриманої в ході дослідження інформації, оскільки це просто унеможлиблює висловлення у суто словесній формі висновків дослідження.

Усі угруповання значень різноманітних змінних, які передбачені програмою дослідження, являються основою статистичних таблиць, які узагальнюють оброблену соціологічну інформацію. Таблиця – це перелік інформації, зазвичай числових даних, приведених у певну систему та рознесених за стовпцями та графами – рядками (рис. 1.17). Відповідно кажучи, процедура побудови таблиць не являється якоюсь особливою математичною операцією. Це скоріше певне відображення рядів розподілів, одержаних у результаті розрахунків. Основна перевага цієї форми полягає в тому, що в ній ємно і коротко подаються пояснення

значень відповідних угруповань. Добре побудована таблиця дозволяє ясніше уявити, описати і пояснити сенс і сутність соціального явища, що досліджується.

Serial Number ↑	Latest reading	Status	Battery level
water meter A-1	0.00 m ³	Inactive	
water meter A-2	0.00 m ³	Inactive	
water meter B-1	0.00 m ³	Inactive	
water meter B-2	0.00 m ³	Inactive	

Рисунок 1.17 – Табличне представлення даних

Таблиці характеризується:

- назвою (якщо таблиць декілька, то додаються й номером);
- кількістю стовпців та назвами (заголовками стовпців);
- кількістю рядків та назвами (заголовками рядків);
- вмістом, що знаходяться на перетині стовпців та рядків.

Побудова таблиці відбувається за певними правилами. Будь-яка статистична таблиця описується за допомогою таких параметрів.

Заголовок – назва таблиці, яке розкриває всю структуру угруповання змінних які описуються, чи характером зв'язку-залежностей між змінними. Поруч у назві або у підзаголовку іноді вказуються загальні всім змінних одиниці виміру: відсотки, кількість відповідей, середній бал.

Підлягаюче – те, що підлягає опису, простіше кажучи вказівку змінної, що аналізується, і тих конкретних значень, які вона набуває або може набути.

Сказане – безпосередньо сам опис, тобто чисельні значення, рознесені за графами – клітинами таблиці або осередками.

Таблиці бувають прості: лінійні, групові та комбінаційні.

Прості представляють простий список окремих значень тієї чи іншої змінної з кількісною чи якісною характеристикою кожної з них окремо. Тому іноді їх називають також переліковими.

Групові таблиці зазвичай містять угруповання одиниць сукупності за однією ознакою, а комбінаційні – за двома і більше ознаками. Такі таблиці являють собою щось більше, ніж звичайний список даних. Це одночасно і спосіб, і в той же час результат систематизації даних. У комбінаційних таблицях, для того щоб уникнути зайвого нагромадження даних, які призводять до ускладнення їх сприйняття, іноді пропускають абсолютні величини (частоти) тих чи інших значень змінних, залишаючи лише пропорції чи відсотки.

Етапи приведення до табличного виду:

- 1) аналіз інформації та виділення об'єктів, про які йдеться;
- 2) виділення властивостей об'єктів та (або) відносин між ними;
- 3) визначення того, чи можна об'єкти об'єднати в деякі підмножини, і в залежності від цього визначення кількості рівнів та ступенів у заголовках;
- 4) визначення загальної кількості стовпців та порядку їх розташування;
- 5) визначення найменувань стовпців та типу даних, які там будуть розміщуватися;
- 6) вибір порядку розміщення рядків та визначення назви кожного рядка таблиці;
- 7) занесення до осередків таблиці реквізитів - даних (рядково або по стовпцях);

1.1.3 Використання графіків

Важливо не тільки правильно вибрати тип графіка, також не менш важливо правильно використовувати графіки:

- Навантажувати графік занадто великою кількістю інформації не потрібно. 4 – 5 це являється оптимальною кількістю різних типів категорій та даних, інакше слід розділити таку діаграму на декілька штук;
- Правильно підібрана шкала та масштаб для графіка. Для графіків з областями та гістограм слід починати шкалу значень з нуля. Постаратися не використовувати інвертовані шкали – це часто вводить користувача в оману щодо даних;
- При використанні кругових діаграм та графіків, де використовується відсоток від загальної частки, сума всіх значень має складати 100%, так щоб не залишалось вільного місця;
- Для того, щоб сприйняття інформації краще сприймалася, слід впорядкувати інформацію за значеннями, за алфавітом, або за логічним змістом.

1.1.4 Оформлення графіків

Ніщо так не тішить око, як графіки які оформлені привильно. Наявність графічного сміття найбільш псує враження від діаграм. Основні принципи оформлення:

- використання палітри схожих кольорів, обов'язково всі кольори повинні бути не яскраві, і слід їх обмежити до шести штук;
- всі допоміжні і другорядні лінії мають бути простими і такими, щоб не кидатися в очі;

- всюди, де можливо, використовувати лише горизонтальні написи;
- колір із прозорістю краще використовувати для графіків з областями;
- щоб не плутатися слід для кожної категорії на графіках використовувати різний колір.

1.1.5 Вимоги до систем візуалізації даних у IoT

Обробка великих обсягів даних включає етапи: збору та первинної обробки інформації, завантаження у сховище, аналіз даних та надання результатів у зручному для сприйняття вигляді. Вимога обробки у режимі реального часу робить завдання ще складнішим. Традиційні NoSQL бази даних обробляють запити надто повільно, тому доцільно мати ще й базу даних безпосередньо в оперативній пам'яті сервера (in-memory). Якщо даних потрібно багато – масштабоване розподілене (in-memory-data-grid) оперативної пам'яті кількох серверів кластера сховище.

При створенні інформаційної системи обов'язково повинні враховуватися особливості об'єкта. Аналітик повинен бути не тільки фахівцем в області Big Data, а й добре знати предметну область, а в ідеалі бути експертом, який детально знається на аналізованих і прогнозованих процесах, що допомагає зрозуміти - які дані потрібні, як проводити їх очищення, вибрати методи аналізу та інтерпретувати результати.

Щоб уникнути ситуацій які призводять до аварій та інших подій, потрібно ретельно підійти до питань візуалізації даних. Оператору потрібно мати на інформаційній панелі (дашборд) зручне представлення найважливіших показників та можливість своєчасного виявлення негативних тенденцій та проблем. У його розпорядженні мають бути засоби детального аналізу ситуації та оперативного формування докладних звітів, включаючи прогностичні системи, що допомагають прийняти оптимальне рішення.

Користувачеві потрібна можливість самостійного налаштування параметрів пошуку та передачі результатів, створення власної панелі спостереження (дашборд), вибудовування візуалізацій та панелей даних на перевагу необхідності виконання операцій. Адже цим все частіше займаються не програмісти, а спеціалісти у конкретній галузі, які проводять кінцеве настроювання рішення. Система повинна забезпечувати можливість пошуку даних в інтерактивному режимі з використанням фільтрів, швидко визначати аномалії та викиди у великих та швидкозмінних даних, давати різноманітні способи представлення статичних даних та тимчасових рядів.

Дуже важливою є швидкість обробки інформації, потокова візуалізація інформації. Чи немає чогось незвичайного в даних, що надходять? Який може статися інцидент і за який час? Система повинна сигналізувати про вихід параметрів за задані межі та про реакцію на це автоматизованих комплексів.

Дані датчиків є безперервними і мають різні вимоги щодо часу знімання інформації (мілісекунди, наносекунди). Слід враховувати, що сигнали з'являються нерегулярно, мати можливість аналізу у тимчасових вікнах появи сигналу, у різних зрізах та з можливістю їх повторного відтворення (цього дня або з архіву). Система має змінювати графічне уявлення у режимі реального часу, забезпечувати можливість деталізації без зупинки живого графічного представлення даних.

Потрібно мати можливість прогнозу за поточковими даними, використання передиктивних моделей, заснованих на шаблонах даних про поведінку в минулому, використовувати чужий накопичений досвід (прогностичне обслуговування, розумна логістика, виявлення клінічних патернів, тощо) Складні формати даних, що надходять з датчиків, вимагають можливості підтримки різних стандартів та інформації для подальшого перетворення та обробки. Крім цього потрібно обробляти дані з SQL та NoSQL баз даних. Причому система повинна забезпечувати до них єдиний інтерфейс, а виконання складноструктурованих запитів не повинно вимагати від користувача знання мови SQL.

Важливою вимогою до систем візуалізації є підтримка геопросторової аналітики в режимі реального часу: географічних карт, SVG (Scalable Vector Graphics – масштабована векторна графіка) файлів, а також можливість відтворення архівних записів.

Часто потрібні можливості експорту результату в різні формати (CVS, JPEG, PDF, фрагменти коду для встановлення безпосередньо на web-сторінки) і засоби об'єднання з іншими програмами за допомогою API. Не варто забувати про популярність мобільних додатків та можливість адаптації зображення до екрану портативного пристрою. І, звичайно, система повинна мати можливість обробки необхідного обсягу інформації, який може бути дуже великим.

1.2 Обробка даних

Обробка даних - вся сукупність операцій (збір, введення, запис, перетворення, зчитування, зберігання, знищення, реєстрація), що здійснюються за допомогою апаратних і програмних засобів, включаючи обмін каналами передачі даних.

Основними видами інформації за формою подання, способами кодування та зберігання, що є найважливішим для інформатики, є:

- графічний або наочний – перший тип, для якого застосовується метод зберігання інформації про навколишній світ у вигляді наскальних малюнків, а пізніше у вигляді картин, фотографій, схем, малюнків на папері, полотні, мармурі та ін. матеріали із зображенням картин реального світу;

- звук – світ навколо нас сповнений звуків і завдання їх збереження та відтворення було вирішено винаходом звукозаписувальних пристроїв у 1877 р. Його різновидом є музична інформація – для цього виду був винайдений спосіб кодування за допомогою спеціальних символів.

- текст - спосіб кодування людської мови спеціальними символами - літерами, причому різні народи мають різні мови і використовують різні набори букв для представлення мови; особливого значення цей метод набув після винаходу паперу та друку;

- числовий – кількісна міра предметів та їх властивостей у світі; особливого значення воно набуло з розвитком торгівлі, економіки та грошового обміну; подібно до текстової інформації, для відображення використовується метод кодування спеціальними символами - цифрами, а системи кодування (числа) можуть бути різними;

- відеоінформація – спосіб збереження «живих» картин навколишнього світу, які з'явилися з винаходом кіно.

Існують також види інформації, для яких досі не існує способів їх кодування та збереження - це тактильна інформація, що передається відчуттями, органолептична інформація, передана запахами та смаками, та ін.

Клода Шеннона вважають творцем загальної теорії інформації та основоположником цифрової комунікації. Світову популярність принесла йому фундаментальна праця 1948 року «Математична теорія комунікації», яка вперше обґрунтувала можливість використання двійкового коду для передачі інформації. Єдність законів обробки інформації в системах різної природи (фізичних, економічних, біологічних тощо) є фундаментальною основою теорії інформаційних процесів, що визначає її загальне значення та специфіку.

Об'єктом вивчення цієї теорії є інформація - поняття значною мірою абстрактне, яке існує «само по собі» поза зв'язком із конкретною галуззю знання, в якій воно використовується. Інформаційні ресурси в сучасному суспільстві грають не менше, а часто і більше, ніж матеріальні ресурси. Знання кому, коли і куди продати товар можна оцінити не менше, ніж сам товар. У зв'язку з цим велика роль буде відведена способам обробки інформації. З'являється все більше досконалих комп'ютерів, нових, зручних програм, сучасних методів зберігання, передачі та захисту інформації. З позицій ринку інформація вже давно є товаром і

ця обставина вимагає інтенсивного розвитку практики, промисловості та теорії комп'ютеризації суспільства.

Комп'ютер як інформаційне середовище не тільки дозволив зробити якісний стрибок в організації промисловості, науки та ринку, а й визначив нові цінні сфери виробництва: комп'ютери, телекомунікації, програмне забезпечення. Тенденції комп'ютеризації суспільства пов'язані з появою нових професій, пов'язаних з комп'ютерною технікою та різними категоріями користувачів комп'ютера. Якщо в 60-70-х роках у цій сфері домінували комп'ютерні спеціалісти (інженери-електроніки та програмісти), які створювали нові комп'ютерні засоби та нові пакети прикладних програм, то сьогодні розширюється категорія користувачів комп'ютерів – представників різних галузей знань, згідно з комп'ютерами в вузькому сенсі, але здатні використовувати їх для вирішення своїх конкретних проблем.

1.2.1 Цілі, завдання та види обробки інформації

Інформаційні процеси завжди займали особливе місце в науці, техніці та житті суспільства. Людство постійно еволюціонує та постійно приділяє особливу увагу автоматизації процесів, але слід зазначити, що внутрішній зміст залишається постійно незмінним.

Збір інформації – це діяльність суб'єкта, під час якої він постійно отримує інформацію про об'єкт, який його цікавить. Збір інформації здійснюється за допомогою людських ресурсів, або за допомогою технічних засобів та систем. Наприклад, людина може отримати інформацію про графік рух поїздів або автобусів самостійно, запам'ятавши їхні графіки руху, або від інших людей, або за допомогою якихось документи, що складені людиною, або за допомогою технічних засобів телефонного дзвінка в довідку або скориставшись мережею інтернет.

Обмін інформацією – це процес, у ході якого джерело та одержувач постійно обмінюються інформацією. Якщо в повідомленнях, які передаються, виявляються помилки, то відбувається повторний обмін цієї інформації. Під час обміну інформацією між джерелом інформації та одержувачем інформації створюється «інформаційний баланс», результат якого в ідеальному випадку такий, що всі джерело та одержувач в кінці сесії обміну інформацією володіють одними й тими же даними. Обмін інформації відбувається за допомогою сигналів або повідомлень. Джерелами інформації можуть виступати будь-які елементи реального світу, які володіють певними властивостями та здібностями.

Накопичення інформації – це формування вихідного, несистематизованого масиву інформації. Масив записаних даних може бути такий, що відображає цінну інформацію, що частіше використовується. Інша частина масиву даних особливої цінності може не представляти, але може знадобитися в подальшому.

Зберігання інформації – це процес який забезпечує видачу інформації на запити користувачів в терміни які були встановлені.

Обробка інформації – це процес перетворення даних відповідно до алгоритму. Після вирішення поставленого завдання які відносяться до обробки обробки інформації результат видається користувачам у вигляді який їм потрібен. Це реалізується після вирішення завдання по видачі інформації. Видача інформації, частіше, проводиться з допомогою пристроїв ЕОМ таких як текстів, таблиць, діаграм, графіків та інших.

1.2.2 Властивості інформації

Як і будь-який об'єкт, інформація має властивості. Характерною відмінною особливістю інформації з інших об'єктів природи та суспільства є дуалізм: на властивості інформації впливають як властивості вихідних даних, що становлять

її змістовну частину, так і властивості методів, що фіксують цю інформацію. З погляду інформатики найважливішими є такі загальні якісні властивості: об'єктивність, достовірність, повнота, точність, актуальність, корисність, цінність, своєчасність, зрозумілість, доступність, стислість та ін.

Об'єктивність інформації. Об'єктивний – існуючий поза та незалежно від людської свідомості. Інформація – це відображення зовнішнього об'єктивного світу. Інформація об'єктивна, якщо вона залежить від методів її фіксації, чийогось думки, судження. приклад. Повідомлення «На вулиці тепло» несе суб'єктивну інформацію, а повідомлення «На вулиці 22 °С» – об'єктивну, але з точністю, яка залежить від похибки засобу вимірювання. Об'єктивну інформацію можна отримати за допомогою справних датчиків вимірювальних приладів. Відбиваючись у свідомості людини, інформація може спотворюватися (переважно чи меншою мірою) залежно від думки, судження, досвіду, знань конкретного суб'єкта, і, в такий спосіб, перестати бути об'єктивною.

Достовірність інформації. Інформація є достовірною, якщо вона відображає справжній стан справ. Об'єктивна інформація завжди є достовірною, але достовірна інформація може бути як об'єктивною, так і суб'єктивною. Достовірна інформація допомагає нам прийняти правильне рішення.

Недостовірна інформація може бути з наступних причин:

- навмисне спотворення (дезінформація) або ненавмисне спотворення суб'єктивної властивості;
- спотворення внаслідок впливу перешкод («зіпсований телефон») та недостатньо точних засобів її фіксації.

Повнота інформації. Інформацію можна назвати повною, якщо її достатньо для розуміння та прийняття рішень. Неповна інформація може призвести до помилкового висновку або рішення. Точність інформації визначається ступенем її близькості до реального стану.

1.2.3 Методи обробки даних

Розрізняють такі методи обробки даних: централізована, децентралізована, розподілена та інтегрована.

Принцип централізованої обробки даних не відповідає високим вимогам до надійності процесу обробки, ускладнюється розвиток систем і не здатність забезпечити необхідні тимчасові параметри при діалоговій обробці даних у розрахованому на багато користувачів режимі. Короткочасний вихід з ладу центральної ЕОМ призводив до фатальних наслідків системи в цілому, оскільки доводилося дублювати функції центральної ЕОМ, значно збільшуючи витрати на створення та експлуатацію систем обробки даних (рис. 1.18).

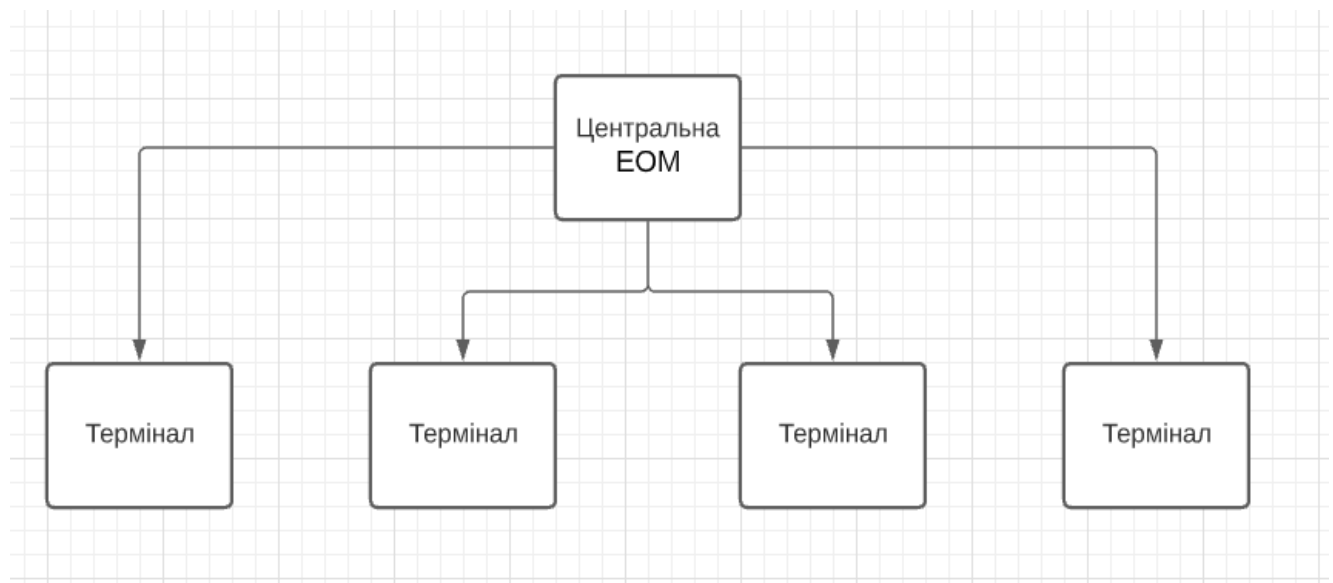


Рисунок 1.18 – Принцип централізованої обробки даних.

Централізована обробка даних передбачає наявність обчислювальних центрів. При цьому спосіб користувач доставляє на обчислювальний центр вихідну інформацію та отримує результати обробки у вигляді результатних документів. Особливістю такого способу обробки є складність і трудомісткість налагодження швидкого, безперебійного зв'язку, велика завантаженість обчислювального центру інформацією (оскільки великий обсяг), регламентацією

термінів виконання операцій, організація безпеки системи від можливого несанкціонованого доступу.

Децентралізована обробка даних, в даний час існує три види технологій децентралізованої обробки даних:

- ПК не об'єднані в локальну мережу (дані зберігаються в окремих файлах та на окремих дисках). Для отримання показників провадиться перезапис інформації на комп'ютер. Недоліки: відсутність взаємопов'язаності завдань, неможливість обробки великих обсягів інформації, ненадійний захист від несанкціонованого доступу;
- ПК об'єднані в локальну мережу, що призводить до створення єдиних файлів даних. Недоліки: ПК не розрахований на великі обсяги інформації;
- ПК об'єднані в локальну мережу, до якої включаються спеціальні сервери.

Розподілена обробка даних - обробка даних, що виконується на незалежних, але пов'язаних між собою комп'ютерах, що становлять розподілену систему.

Для реалізації розподіленої обробки даних було створено багатомашинні асоціації, структура яких розробляється по одному з наступних напрямків:

- багатомашинні обчислювальні комплекси;
- комп'ютерні (обчислювальні) мережі.

Багатомашинний обчислювальний комплекс – група встановлених поруч обчислювальних машин, об'єднаних за допомогою спеціальних засобів сполучення та виконують спільно єдиний інформаційно-обчислювальний процес.

Розподілений спосіб обробки даних заснований на розподілі функцій обробки між різними ЕОМ, включеними до мережі. Цей спосіб може бути реалізований двома шляхами: перший передбачає встановлення ЕОМ у кожному вузлі мережі (або на кожному рівні системи), при цьому обробка даних здійснюється однією або декількома ЕОМ в залежності від реальних можливостей системи та її потреб на поточний момент часу (рис. 1.19).

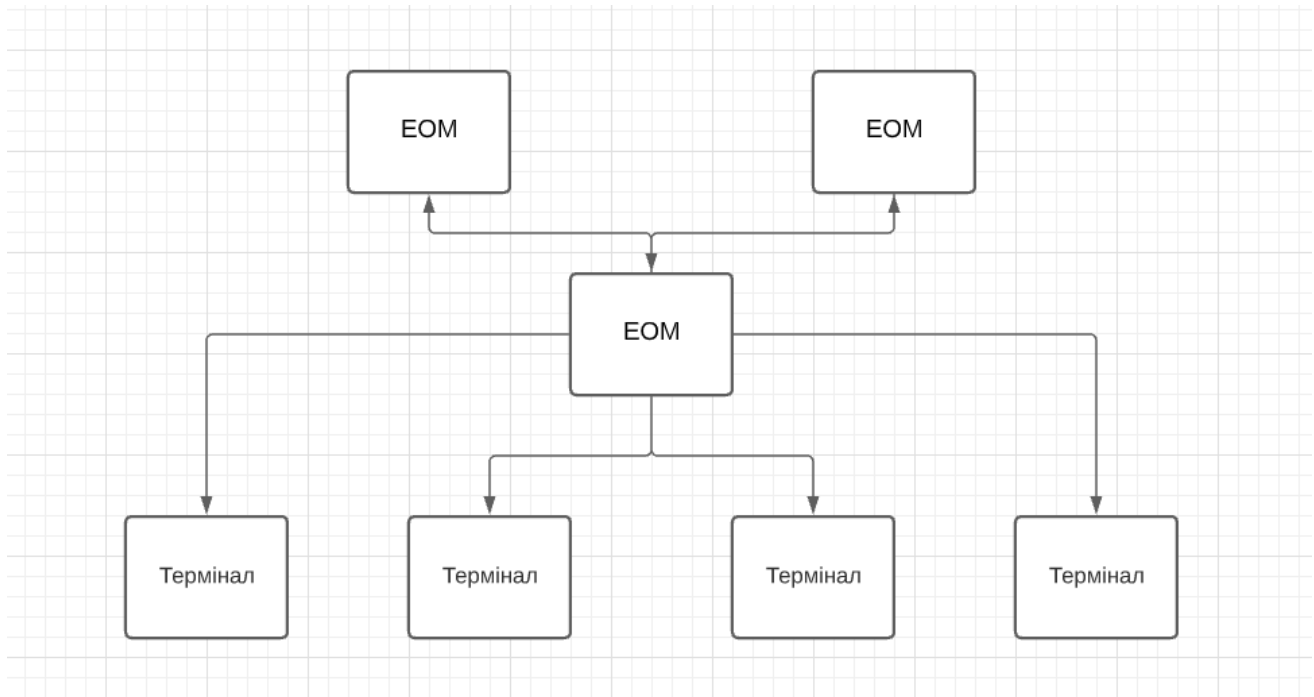


Рисунок 1.19 - Принцип розподіленої обробки даних.

Другий шлях – розміщення великої кількості різних процесорів усередині однієї системи. Такий шлях застосовується у системах обробки банківської та фінансової інформації, там, де необхідна мережа обробки даних (філії, відділення тощо).

Переваги розподіленого способу: можливість обробляти у визначені терміни будь-який обсяг даних; високий ступінь надійності, так як при відмові одного технічного засобу є можливість моментальної заміни на інший; скорочення часу та витрат на передачу даних; підвищення гнучкості систем, спрощення розробки та експлуатації програмного забезпечення тощо.

Розподілений спосіб ґрунтується на комплексі спеціалізованих процесорів, тобто кожна ЕОМ призначена для вирішення певних завдань або завдань свого рівня.

Інтегрований метод обробки інформації передбачає створення інформаційної моделі керованого об'єкта, т. е. створення розподіленої БД. Такий спосіб забезпечує максимальну зручність для користувача. З одного боку, бази даних передбачають колективне користування та централізоване управління. З іншого боку, обсяг інформації, різноманітність розв'язуваних завдань вимагають

розподілу БД. Технологія інтегрованої обробки інформації дозволяє покращити якість, достовірність та швидкість обробки, оскільки обробка проводиться на основі єдиного інформаційного масиву, одноразово введеного в ЕОМ. Особливістю цього способу є відділення технологічно та за часом процедури обробки від процедур збору, підготовки та введення даних.

1.2.4 Технічні засоби обробки інформації

Допоміжні засоби – це обладнання, що забезпечує працездатність основних засобів, а також обладнання, що полегшує та робить управлінську працю комфортнішою. До допоміжних засобів обробки інформації відносяться засоби оргтехніки та ремонтно-профілактичні засоби. Оргтехніка представлена дуже широкою номенклатурою коштів - від канцелярських товарів до засобів доставки, розмноження, зберігання, пошуку та знищення основних даних, засобів адміністративно-виробничого зв'язку і так далі, що робить роботу управлінця зручною та комфортною.

Основні засоби – це знаряддя праці автоматизованої обробки інформації. До них відносяться засоби: реєстрації та збору інформації; прийому та передачі даних; підготовки даних; введення; обробки інформації та відображення інформації.

Одержання первинної інформації та її реєстрація є одним із трудомістких процесів. Тому широко застосовуються пристрої для механізованого та автоматизованого виміру, збору та реєстрації даних. Номенклатура цих засобів дуже велика. До них відносять: електронні ваги, різноманітні лічильники, табло, витратоміри, касові апарати, машинки для рахунку банкнот, банкомати та багато іншого. Сюди ж відносять різні реєстратори виробництва, призначені для оформлення та фіксації відомостей про господарські операції на машинних носіях.

1.2.5 Засоби прийому та передачі інформації.

Під передачею інформації розуміється процес пересилання даних (повідомлень) від одного пристрою до іншого. Взаємодіюча сукупність об'єктів, пристроїв передачі та обробки даних називається мережею. Об'єднують пристрої, призначені для передачі та прийому інформації. Вони забезпечують обмін інформацією між місцем її виникнення та місцем її обробки. Структура засобів і методів передачі даних визначається розташуванням джерел інформації та засобів обробки даних, обсягами та часом на передачу даних, типами ліній зв'язку та іншими факторами. Засоби передачі даних є абонентськими пунктами (АП), апаратурою передачі, модемами, мультиплексорами.

Засоби підготовки даних представлені пристроями підготовки інформації на машинних носіях, пристроями передачі інформації з документів на носії, що включають пристрої ЕОМ. Ці пристрої можуть здійснювати сортування та коригування.

Кошти введення служать для сприйняття даних з машинних носіїв та введення інформації у комп'ютерні системи.

Кошти обробки інформації грають найважливішу роль комплексі технічних засобів обробки інформації. До засобів обробки відносять комп'ютери, які у свою чергу поділяють на чотири класи: мікро, малі (міні); великі та суперЕОМ.

Засоби відображення інформації використовують для виведення результатів обчислення, довідкових даних та програм на машинні носії, друк, екран тощо.

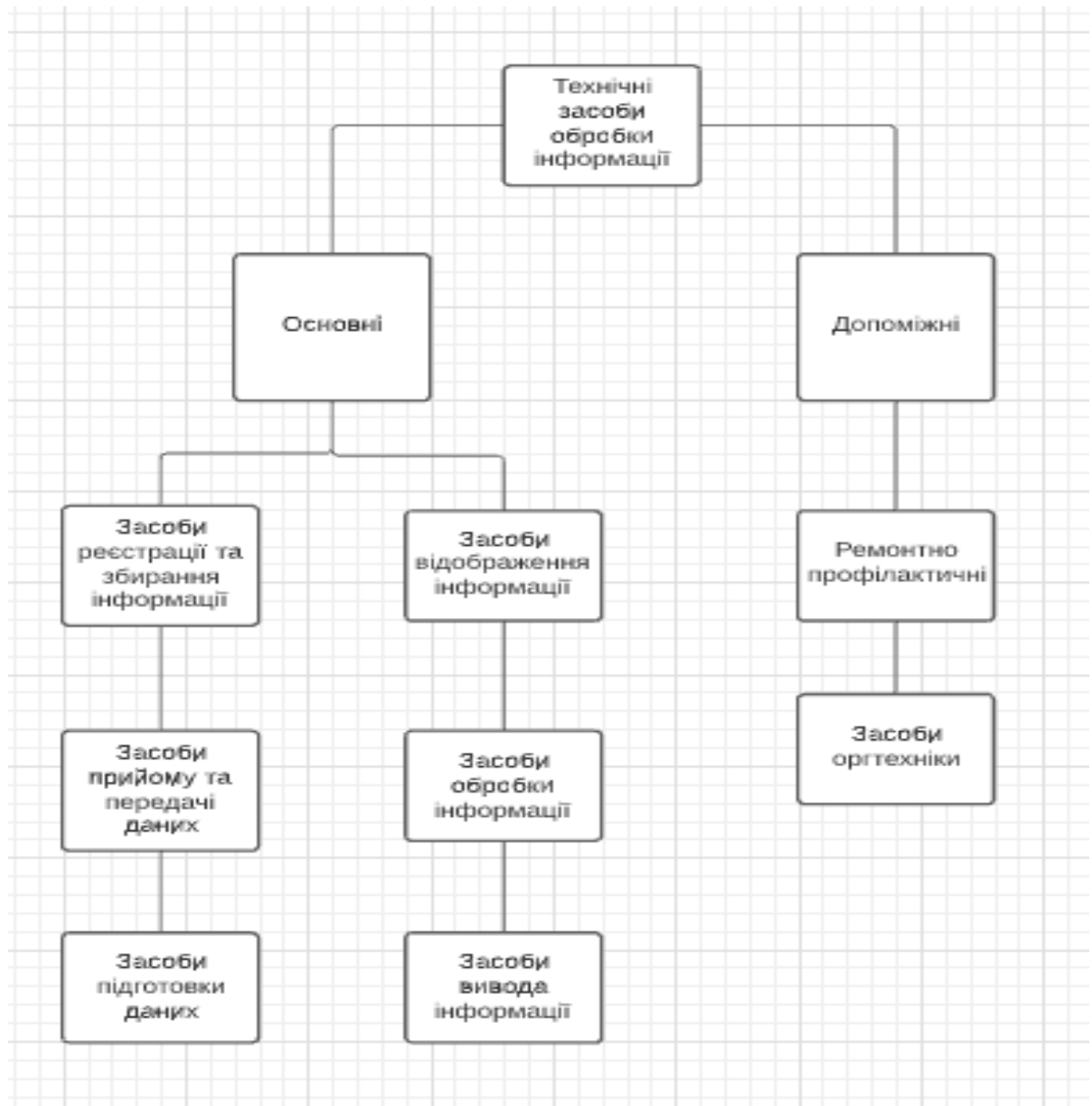


Рисунок 1.20 – Класифікація технічних засобів обробки інформації

2 ОПИС ТЕХНОЛОГІЙ ТА ЗАСОБІВ

2.1 IoT платформа

IoT платформа – це комплекс програм, які застосовуються для підключення різноманітних інтернет речей до хмарної інфраструктури зберігання інформації та надання віддаленого доступу до них.

IoT-платформа це місток, що поєднує такі поняття як «речі» та «інтернет». IoT-платформа є ключовим інструментом розробки IoT-додатків та сервісів, який поєднує фізичні об'єкти (речі) та мережу (інтернет).

З моменту появи терміна «Інтернет речей» мережі, що складаються з великої кількості пристроїв, що спілкуються між собою, швидко розвиваються. Внаслідок цього, IoT (Internet of Things) стає однією з основних технологій у суспільстві. З погляду технологічних та технічних аспектів розвитку IoT в даний час існує чіткий поділ між апаратними та програмними платформами для підключення пристроїв, причому більшість постачальників пропонують саме програмні IoT платформи.

Платформи IoT забезпечують безшовну інтеграцію різних апаратних засобів, використовуючи протоколи зв'язку, застосовуючи різні типи топології (пряме підключення або шлюз) та використовуючи SDK у разі потреби тощо.

Використовуючи інтерфейси інтеграції з північним кордоном, що надаються платформою, ви також можете передавати зібрані дані IoT в певні системи аналізу та зберігання даних, а також передавати дані на підключені пристрої (конфігурація, повідомлення) або між ними (елементи управління, події), використовуючи різні види користувацьких додатків.

Найпопулярнішими програмними IoT платформами є: Microsoft Azure IoT, Amazon Web Services (AWS) IoT, Google Cloud, ThingWorx IoT, IBM Watson, Artik

від Samsung Electronics, Cisco IoT Cloud Connect, Salesforce IoT Cloud та багато інших.

2.1.1 Технічні характеристики платформи

Критеріями відмінності програмних IoT платформ одна від одної є:

- масштабованість – кількість кінцевих пристроїв, які можуть підключатися до платформи, включаючи ефективне балансування навантаження серверів;
- простота використання – гнучкість API інтеграції та простота управління вихідним кодом;
- варіанти розгортання – публічна або приватна хмара;
- безпека – захист даних шляхом шифрування, контролю доступу користувачів тощо.
- база даних – варіант зберігання даних, одержуваних із пристроїв, наявність гібридних хмарних баз даних тощо.

Серед протоколів, що використовуються платформами IoT, найпопулярнішими є MQTT, CoAP, HTTP/HTTPS, AMQP, XMPP, DDS.

Сучасні IoT-платформи виконують такі завдання:

1. Забезпечують інтеграцію різних апаратних засобів за допомогою спеціальних інтерфейсів, протоколів зв'язку, а також засобів зберігання, обробки та інтелектуального аналізу даних.
2. Зводять різні протоколи та формати даних у єдиних інтерфейс. Таким чином гарантують точну передачу інформації та взаємодію з усіма кінцевими пристроями.
3. Забезпечують безперебійну роботу кінцевих пристроїв (датчиків, сенсорів, контролерів тощо), оновлення на них.
4. Відповідають за надійне зберігання інформації у базах даних

5. Здійснюють моніторинг поточного та прогнозування майбутнього стану технологічного обладнання.
6. Інтегрують та класифікують дані з використанням методів машинного навчання (аналітика) у реальному часі.
7. Візуалізують зібрану інформацію як таблиць, графіків, діаграм.

2.1.2 IoT платформа для Інтернету Речей

Кількість пристроїв, які вважаються інтернет-речами, постійно зростає. Саме тому незабаром IoT отримає статус однієї з найважливіших сучасних технологій. Якщо розглядати розвиток інтернет речей з позиції технологій, то сьогодні спостерігається чіткий поділ між ПЗ та апаратними засобами, що використовуються для підключення даних пристроїв. Більшість постачальників надають програмні IoT платформи.

IoT платформи призначені для інтеграції апаратного забезпечення шляхом застосування мережевих протоколів, різних топологій мережі та набору засобів розробки Software Development Kit у разі виникнення такої необхідності.

Застосування інтерфейсів інтеграції також дозволяє передавати інформацію, отриману від інтернет речей, у різні системи аналізу та зберігання інформації. У користувача також є можливість передавати на підключені пристрої або обмінюватися інформацією між ними за допомогою різних видів додатків.

На даний момент найбільш затребуваними програмними платформами для інтернету речей вважаються:

- хмарна платформа Azure від компанії Microsoft;
- комерційну публічну хмару Web Services від компанії Amazon;
- Cloud Platform від компанії Google;
- платформа для інновацій промислових ThingWorx;
- суперкомп'ютер Watson від компанії IBM та ін.

2.1.3 Архітектура Інтернет речей

Класична архітектура інтернету речей включає:

- IoT-пристрої. Вони збирають показання з датчиків та виконують фізичні дії. Можуть бути персональними, носимими та вбудованими.
- Шлюзи, які отримують інформацію від пристроїв та передають їм команди виконання дій. Як правило, представлені апаратним маршрутизатором чи програмним забезпеченням; використовують різні протоколи.
- Сервер, де зберігаються, обробляються та аналізуються показання датчиків. Може бути реалізований з урахуванням віртуального сервера, реальної машини чи через хмару.
- Клієнтська частина реалізується через мобільний або веб-додаток. Забезпечує доступ до даних пристроїв та наочне представлення результатів аналізу.

У загальному вигляді схема такого сервісу представлена на рисунку 2.1 нижче:

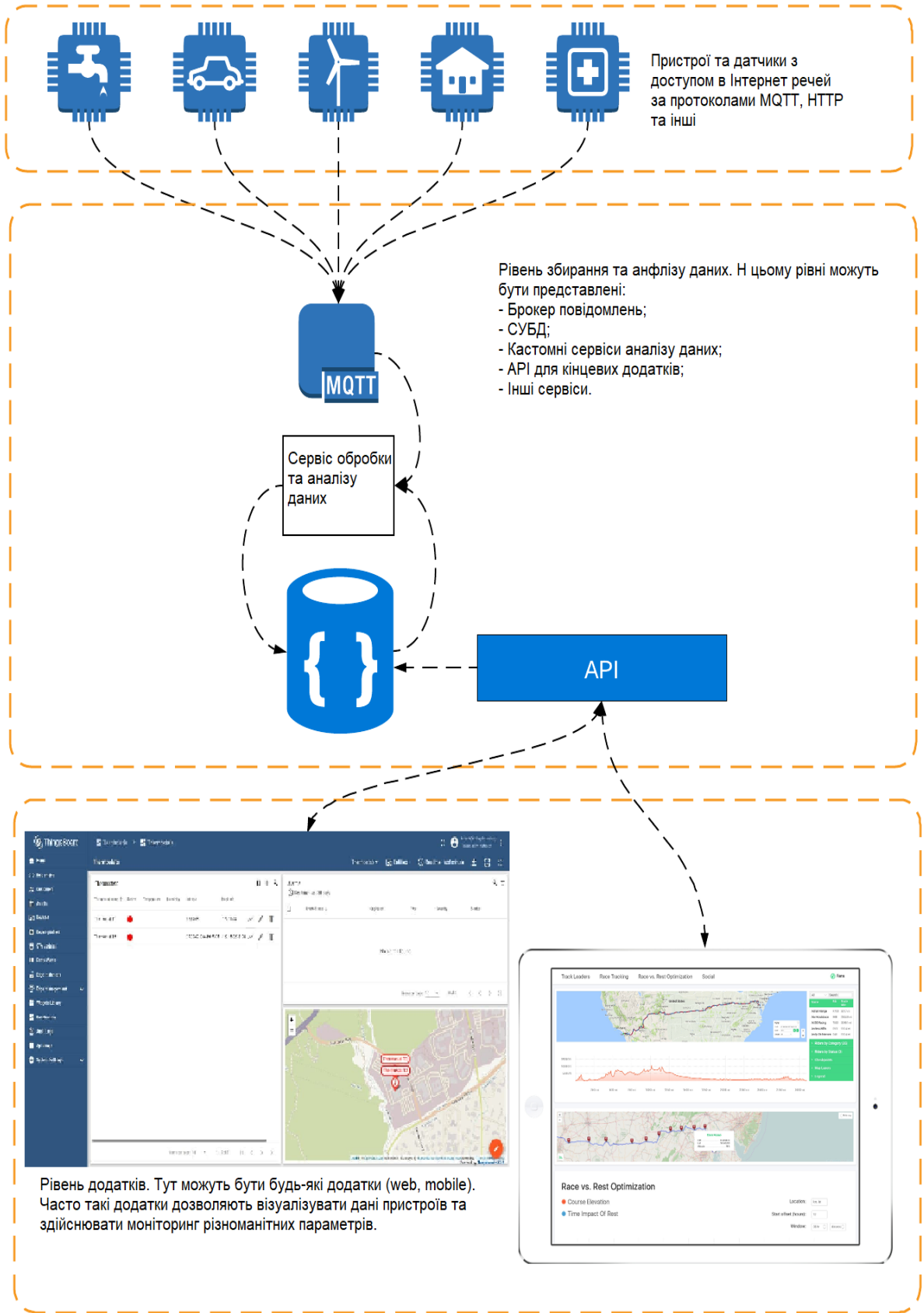


Рисунок 2.1 – Схема класичної архітектури Інтернету речей

2.1.4 Технічні характеристики

Програмні IoT платформи відрізняються за такими технічними характеристиками:

- Масштабованість – максимально можлива кількість інтернет речей, підключених до платформи.
- Простота застосування – гнучкість програмного інтерфейсу програми та простота роботи з вихідним кодом.
- Способи розгортання – приватне чи публічне хмарне сховище.
- Безпека – забезпечення захисту за допомогою шифрування, моніторингу доступу тощо.
- База даних – організаційна структура зберігання інформації, що передається з пристроїв.

У процесі використання IoT платформ застосовується безліч протоколів. Найбільш затребуваними варіантами вважаються Constrained Application Protocol (CoAP), Message Queue Telemetry Transport (MQTT), DirectDraw Surface (DDS), eXtensible Messaging and Presence Protocol (XMPP) та HyperText Transfer Protocol Secure (HTTP/HTTPS).

Більшість програмних платформ IoT може підтримувати аналітики в режимі онлайн. IoT плати підтримують як пакетну (Spark, Samza), і інтерактивну (Spark MLLIB) аналітику. До того ж програмні IoT платформи підтримують прогностичну аналітику, яка ґрунтується на різних методах машинного та статистичного навчання.

2.1.5 Практичні приклади застосування

Платформи IoT активно застосовуються виробниками та постачальниками смарт-пристроїв. Вони оснащують свою продукцію функціями онлайн-моніторингу, дистанційного керування, налаштування повідомлень, інтеграції з мобільними пристроями тощо.

IoT платформи активно використовуються і у промисловому секторі. Компанії, що працюють у цьому секторі, оптимізують свою роботу шляхом інтелектуального обслуговування обладнання, аналізу зібраних даних у режимі он-лайн тощо. IoT платформи також застосовуються в процесі створення системи смарт-міста, призначеної для застосування державними та приватними корпораціями, а також кінцевими користувачами.

Варто додати, що система «смарт-місто» використовується для забезпечення безпеки на міських вулицях та у спорудах різного призначення. Ця система також необхідна для інтелектуального моніторингу мереж та екологічної ситуації.

2.2 Дослідження IoT платформи ThingsBoard

ThingsBoard – це платформа IoT з відкритим кодом, яка забезпечує швидкий розвиток, управління та масштабування проєктів IoT. Головна мета платформи надати готове хмарне або локальне рішення IoT, яке забезпечить серверну інфраструктуру для додатків IoT.

За допомогою IoT платформи можна:

- Надавати пристрої, активи та клієнтів, а також визначення відносин між ними;
- Збирати та візуалізувати дані з пристроїв;

- Аналізувати вхідну телеметрів та запускати тривоги за допомогою складної обробки подій;
- Керувати своїми пристроями за допомогою віддалених викликів процедур (RPC);
- Створювати робочі процеси на основі подій життєвого циклу пристрою, подій REST API, запиту RPC тощо;
- Створювати динамічні та адаптивні інформаційні панелі та презентувати своїм клієнтам телеметрію пристроїв або активів і статистику;
- Вмикати функції, специфічні для випадку використання, за допомогою настанованих ланцюжків правил;
- Передача даних пристроїв в інші системи;

2.2.1 Архітектура платформи

Архітектура платформи включає:

- Масштабованість: горизонтально масштабована платформа, яка створена з використанням передових технологій з відкритим кодом;
- Відмовостійкість: кожен вузол в кластері є ідентичним;
- Надійність та ефективність: один серверний вузол може обробляти десятки або сотні тисяч пристроїв залежно від варіанту використання. Кластер може обробляти мільйони пристроїв.
- Довговічність: платформа підтримує різні реалізації черги для забезпечення надзвичайно високої довговічності повідомлень;
- Налаштування: додавати новий функціонал легко за допомогою налаштувань віджетів і вузлів механізму правил.

На схемі нижче показано ключові компоненти системи та інтерфейси, які надаються (рис 2.2).

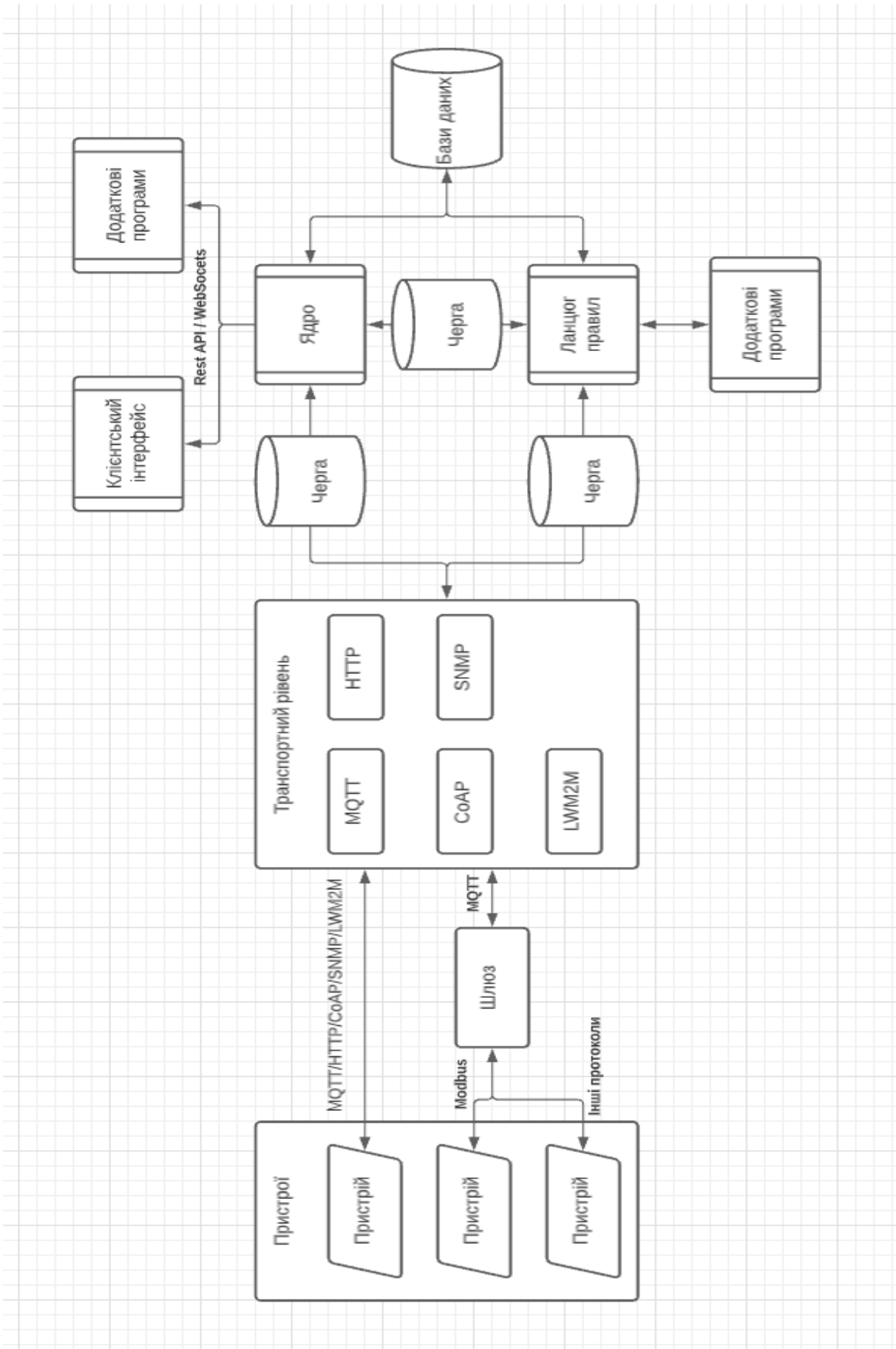


Рисунок 2.2 Архітектурна схема платформи

Платформу можна розділити на 4 функціональні частини:

1. Транспортна частина.

Платформа надає API на основі MQTT, HTTP, CoAP та LwM2M, які доступні для програм вашого пристрою. Кожен з API протоколу надається окремим компонентом сервера. MQTT Transport також надає API шлюзу для використання шлюзами, які представляють декілька підключених пристроїв та/або датчиків.

Після того, як транспортний засіб отримує повідомлення від пристрою, воно аналізується і переміщується до довготривалої черги повідомлень. Доставка повідомлень підтверджується на пристрій лише після того, як відповідне повідомлення підтверджено чергою повідомлень.

2. Частина ядра.

Ядро відповідає за обробку викликів REST API і підписок на WebSocket. Він також відповідає за збереження актуальної інформації про активні сеанси пристрою та моніторинг стану підключення пристрою. Ядро використовує Actor System для реалізації акторів для основних сутностей: орендарів і пристроїв. Вузли платформи можуть приєднатися до кластера, де кожен вузол відповідає за певні розділи вхідних повідомлень.

3. Частина конфігурації правил.

Конфігурація за допомогою правил є серцем системи і відповідає за обробку вхідних повідомлень. Також цей двигун використовує Actor System під капотом для реалізації акторів для основних сутностей: ланцюжків правил і вузлів правил. Вузли можуть приєднуватися до кластера, де кожен вузол відповідає за певні розділи вхідних повідомлень.

Підписка відбувається на вхідний канал даних із черги (черг) і підтверджує повідомлення лише після його обробки. Існує кілька доступних стратегій, які контролюють порядок або обробку повідомлень і критерії підтвердження повідомлення.

Правила можуть працювати в двох режимах: спільному та ізольованому. У спільному режимі система правил обробляє повідомлення, які належать кільком клієнтам. В ізольованому режимі правила можна налаштувати на обробку повідомлень лише для певного клієнта.

4. Частина користувацького інтерфейсу.

Платформа надає легкий компонент, написаний за допомогою фреймворку Express.js для розміщення статичного вмісту веб-інтерфейсу. Ці компоненти повністю не мають стану і не мають великої кількості конфігурацій. Статичний веб-інтерфейс містить пакет програм. Після завантаження програма починає використовувати REST API і WebSockets API, надані ядром.

2.2.2 Підключення пристроїв до платформи

Платформа надає велику кількість різних методів підключення пристроїв до платформи, алгоритм вибору правильного варіанту підключення пристроїв наведено нижче (рис. 2.3).

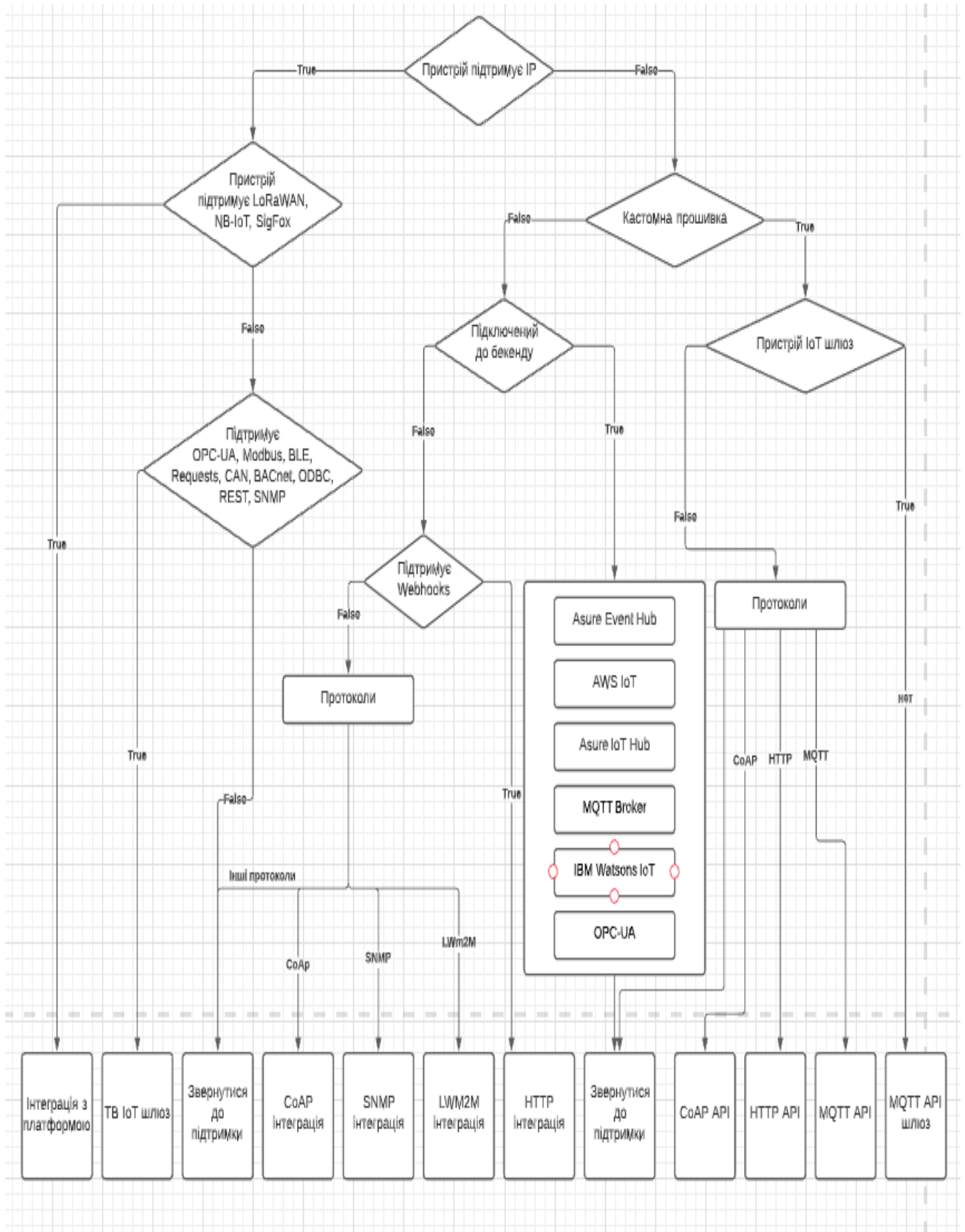


Рисунок 2.3 – Блок схема алгоритму підключення пристроїв

3 РОЗРОБКА ІНФОРМАЦІЙНОЇ СИСТЕМИ

3.1 Розробка віджетів для інформаційної системи

Віджети ThingsBoard – це додаткові модулі інтерфейсу користувача, які легко інтегруються в будь-яку інформаційну панель IoT. Вони забезпечують функції кінцевого користувача, такі як візуалізація даних, дистанційне керування пристроєм, керування сигналізацією та відображення статичного користувацького вмісту HTML. Відповідно до наданих функцій, кожне визначення віджета представляє певний тип віджета.

Платформа вже має певні набори віджетів для використання користувачами, деякі з них ми будемо використовувати для побудови власної інформаційної системи. Ще частина віджетів була написана та додана до системи для реалізації наших потреб в інформаційній системі.

Створення нових віджетів.

Щоб створити нове визначення віджета, перейдіть до «Бібліотеки віджетів» і відкрийте наявний «Пакет віджетів» або створіть новий (Рис. 3.1).

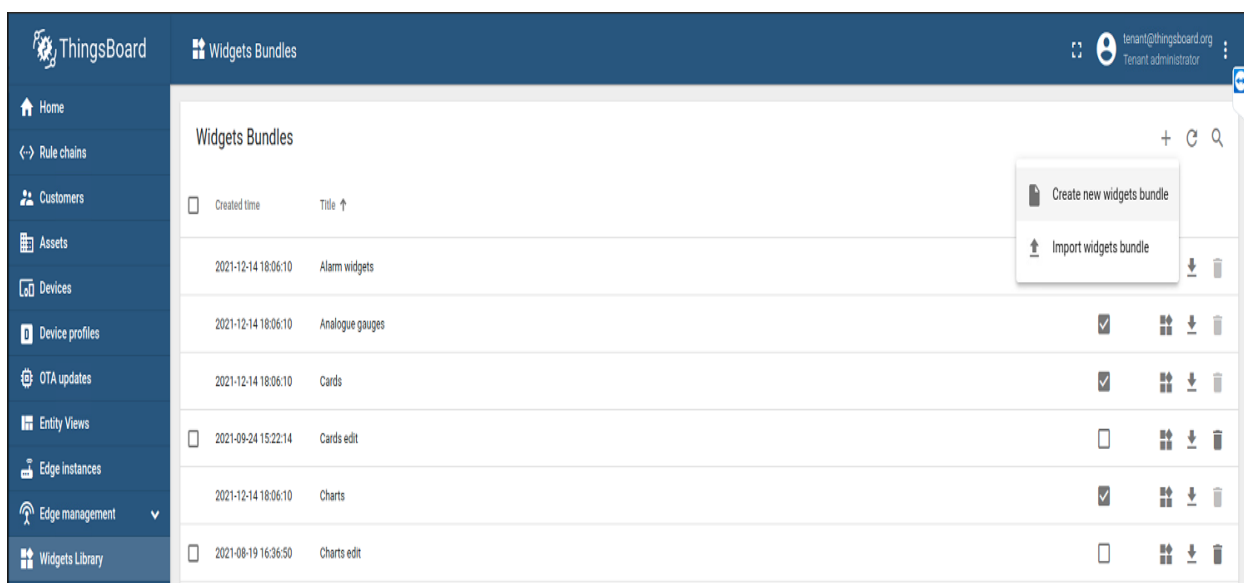


Рисунок 3.1 – Створення нового Пакету віджетів

У перегляді «Пакет віджетів» натисніть велику кнопку «+» у нижній правій частині екрана, а потім натисніть кнопку «Створити новий віджет» (Рис. 3.2).

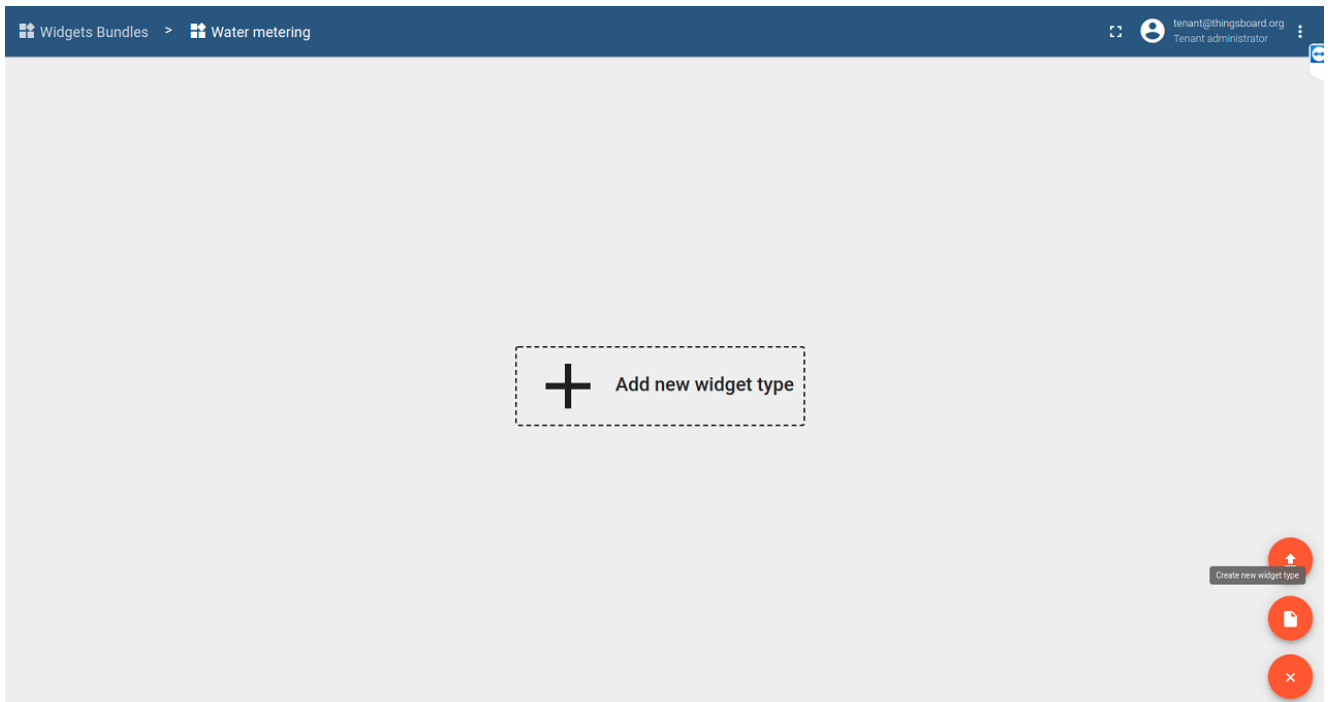


Рисунок 3.2 – Створення нового віджета

Повинно з'явитися вікно «Виберіть тип віджета» (Рис. 3.3) з вибором параметрів, що відповідають типу віджета, який ви плануєте розробити.

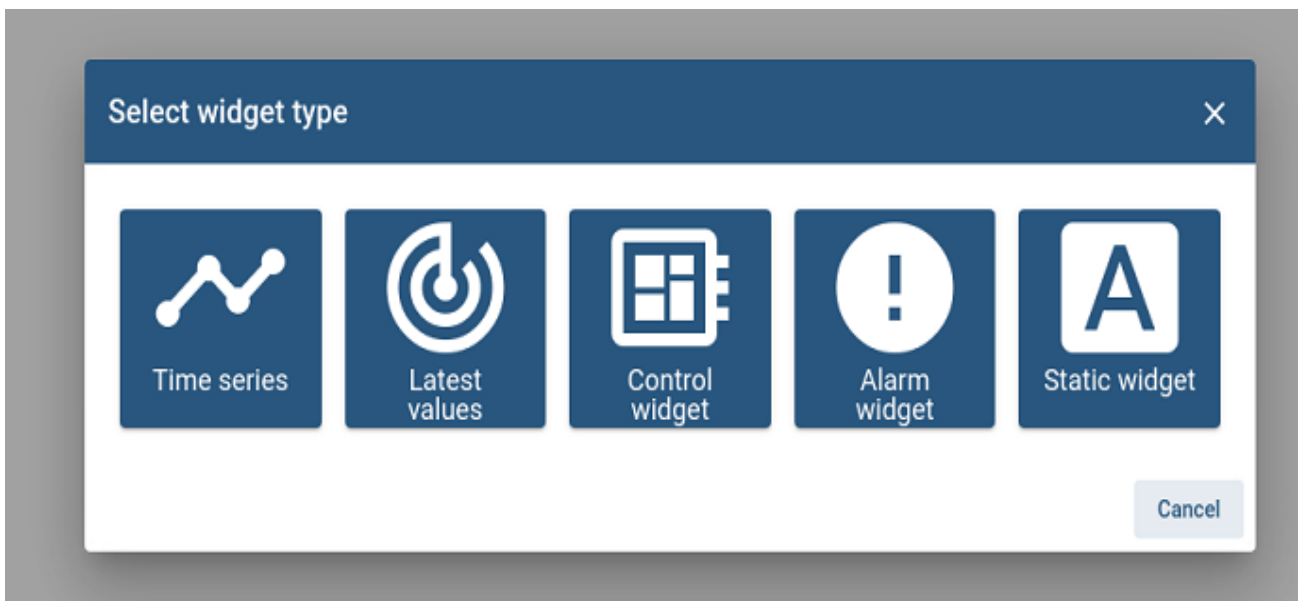


Рисунок 3.3 – Вибір типу віджета який створюємо

Відповідно до наданих функцій, кожне визначення віджета представляє певний тип віджета. Існує п'ять типів віджетів:

1. Latest values;

Відображає останні значення атрибута певного об'єкта або точки даних часового ряду (наприклад, будь-який віджет віджета Gauge або Entities Table). Цей тип віджетів використовує значення атрибутів сутності або часових рядів як джерело даних. Цифровий датчик у прикладі відображає поточне значення температури (Рис. 3.4).

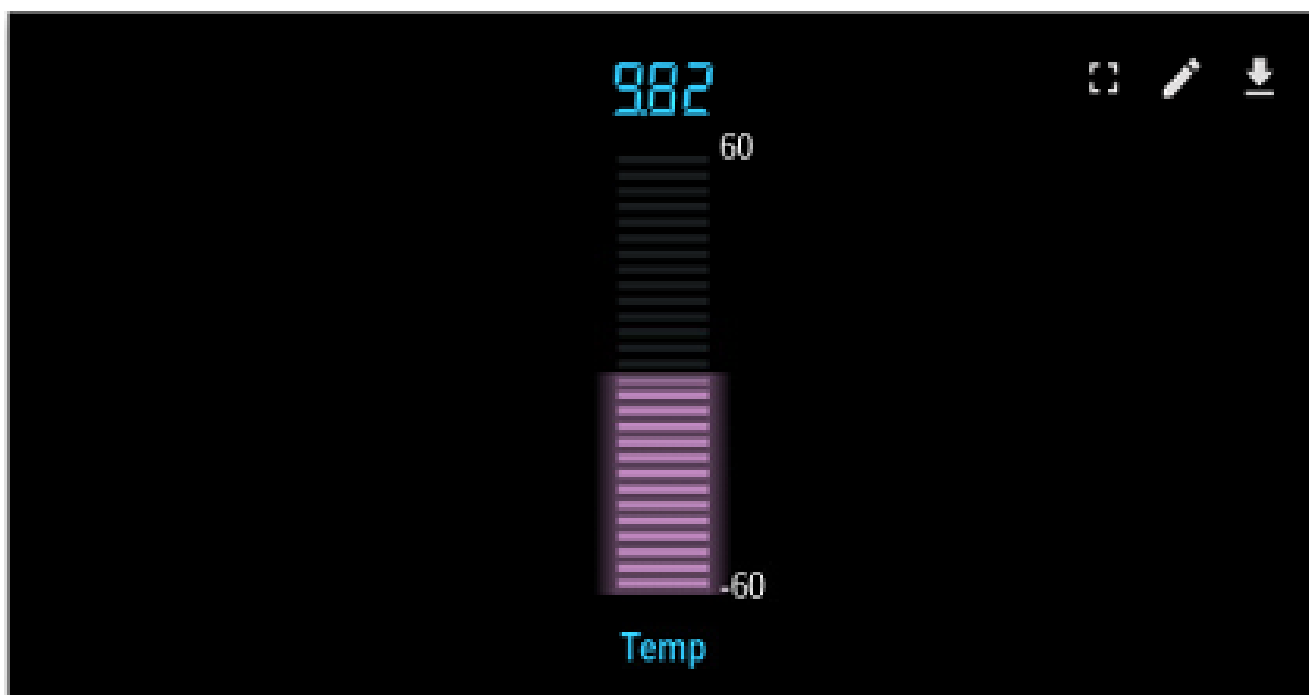


Рисунок 3.4 – Цифрова діаграма для відображення показників

2. Timeseries;

Відображає історичні значення за вибраний період часу або останні значення у певному часовому вікні (наприклад, «Лінійна діаграма» або «Гістограма»). Цей тип віджетів використовує лише значення часових рядів сутності як джерело даних. Щоб задати часові рамки відображених значень, використовуються налаштування Timewindow. Часове вікно можна вказати на сторінці інформаційної панелі або в деталях віджета. Це може бути або реальний час - динамічно змінений часовий проміжок для певного останнього інтервалу, або історія - фіксований історичний часовий проміжок. Усі ці параметри є

частиною конфігурації віджетів часового ряду. У прикладі «Лінійна діаграма часових рядів» відображає значення показників пристрою в режимі реального часу (Рис. 3.5).

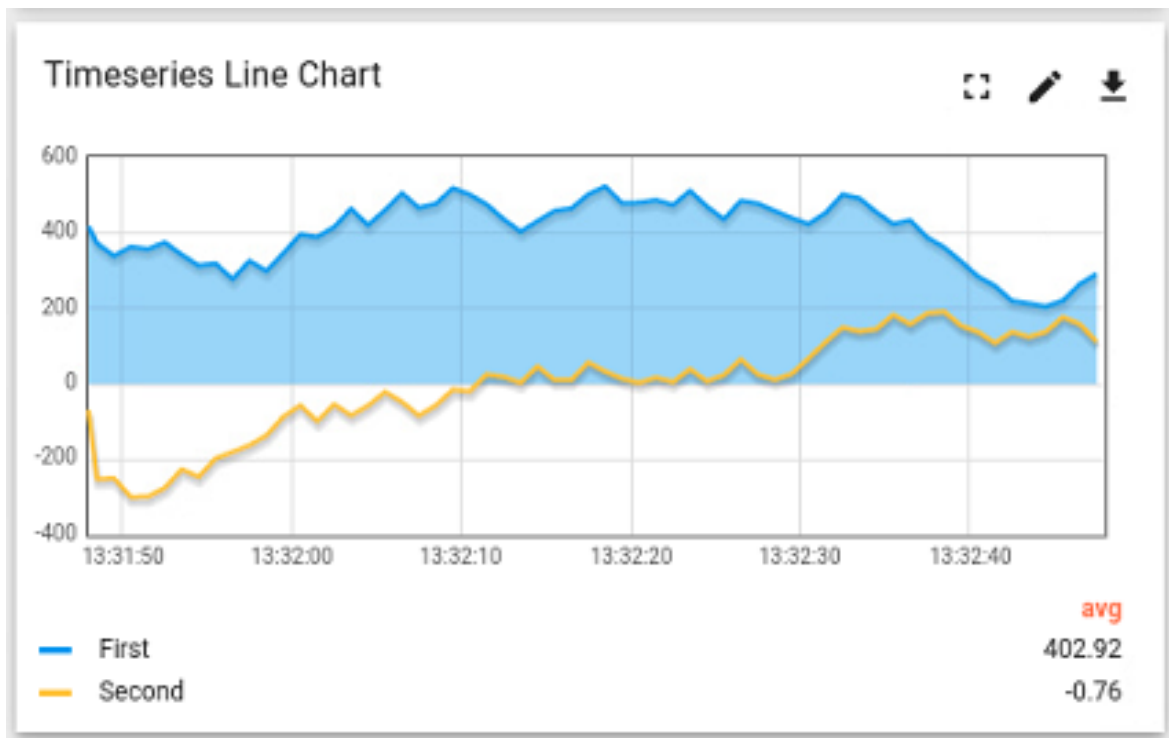


Рисунок 3.5 – Лінійна діаграма відображення в реальному часі

3. RPC (Control widget);

Віджет керування дозволяє надсилати RPC-команди на пристрої, обробляє та візуалізує відповідь з пристрою (наприклад, «Raspberry Pi GPIO Control»).

Віджети RPC налаштовуються, вказуючи цільовий пристрій як цільову кінцеву точку для команд RPC. У прикладі віджет «Основне керування GPIO» надсилає команди перемикачання GPIO та виявляє поточний статус перемикача GPIO (Рис. 3.6).

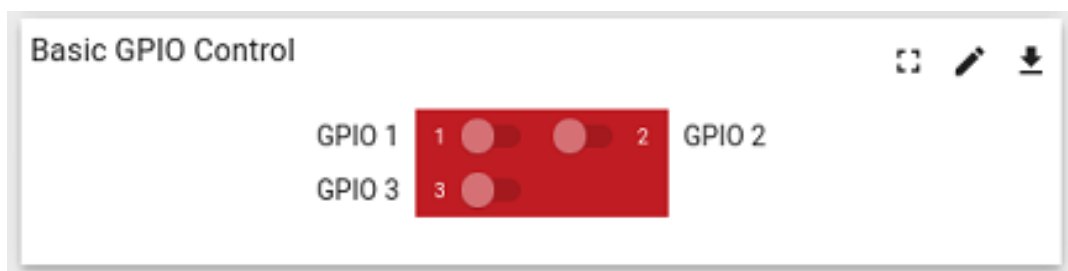


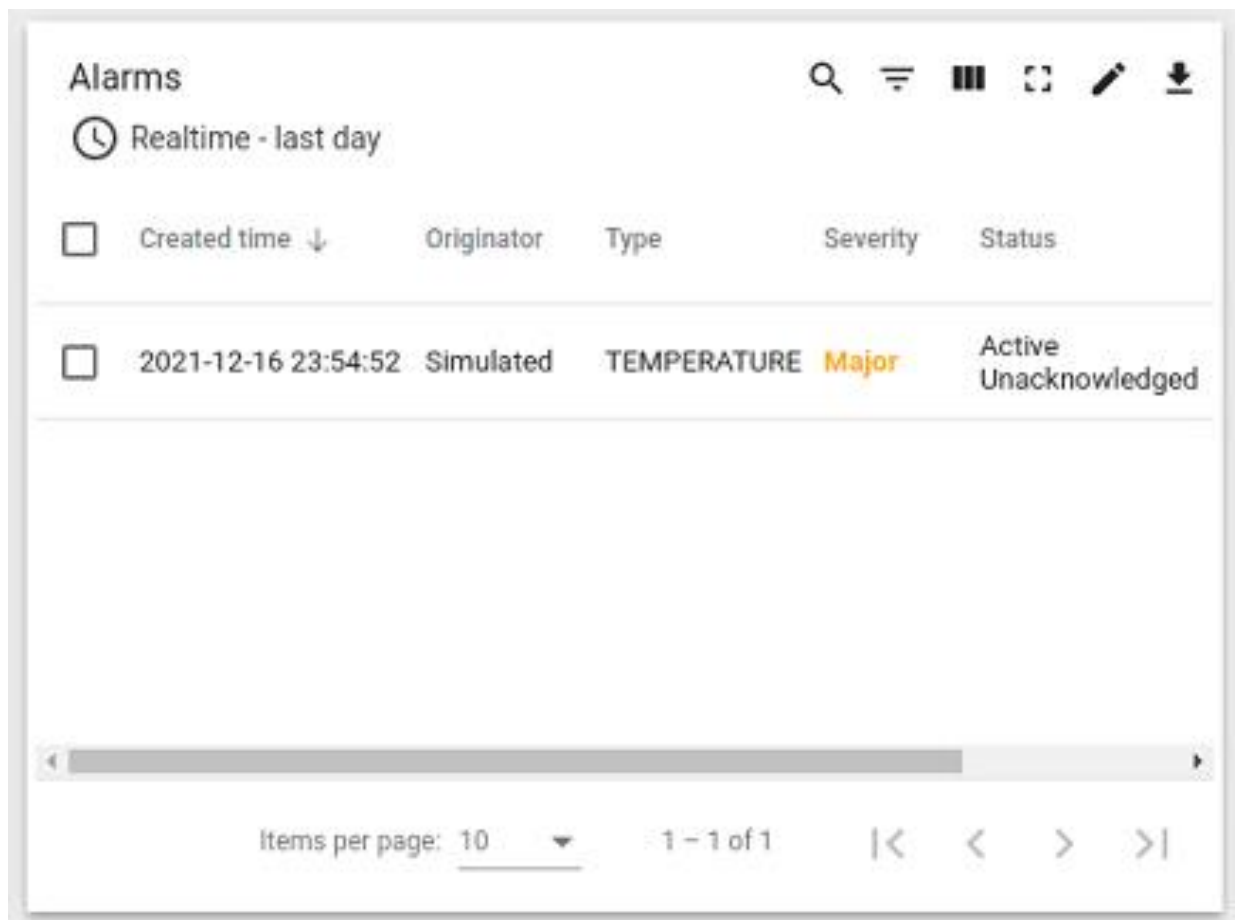
Рисунок 3.6 - Raspberry Pi GPIO Control

4. Alarm widget;

Відображає тривоги, пов'язані з вказаним об'єктом у певному часовому вікні (наприклад, «Таблиця тривог»). Віджет нагадування налаштовується шляхом визначення об'єкта як джерела тривог та відповідних полів тривоги. Як віджети часового ряду, віджети тривоги мають конфігурацію часового вікна, щоб задати часовий проміжок відображених тривог.

Крім того, конфігурація містить параметри «Статус тривоги», «Ступінь тривоги» та «Тип сигналу». Параметр «Статус тривоги» визначає статус тривоги, що отримується. «Ступінь тривоги» контролює частоту отримання сигналів у секундах. «Тип сигналізації» допомагає визначити основну причину тривоги. Наприклад, «Висока температура» та «Низька вологість» – це два різні сигнали.

У прикладі віджет «Таблиця тривог» відображає останній нагадування для пристрою в режимі реального часу (Рис. 3.7).



The screenshot shows a web interface for 'Alarms'. At the top, there is a title 'Alarms' and a toolbar with icons for search, filter, list, refresh, edit, and download. Below the title, there is a clock icon and the text 'Realtime - last day'. A table with one row of data is displayed. The table has columns for 'Created time', 'Originator', 'Type', 'Severity', and 'Status'. The data row shows a timestamp '2021-12-16 23:54:52', 'Simulated' as the originator, 'TEMPERATURE' as the type, 'Major' as the severity, and 'Active Unacknowledged' as the status. At the bottom of the interface, there is a pagination control showing 'Items per page: 10' and '1 - 1 of 1'.

Created time ↓	Originator	Type	Severity	Status
2021-12-16 23:54:52	Simulated	TEMPERATURE	Major	Active Unacknowledged

Рисунок 3.7 – Таблиця тривог

Основні поняття сигналізації нижче:

- **Originator** – це об’єкт, який викликає тривогу. Наприклад, пристрій А є ініціатором тривоги, якщо ThingsBoard отримує від нього показники температури і піднімає сигнал «Висока температура», оскільки показник перевищує поріг.
- **Type** – тип сигналізації допомагає визначити основну причину тривоги. Наприклад, «Висока температура» та «Низька вологість» — це два різні сигнали.
- **Severity** – кожна тривога має серйозність, яка є критичною, серйозною, другорядною, попереджувальною або невизначеною (відсортовано за пріоритетом у порядку спадання).
- **Lifecycle** – сигнал може бути активним або знятим. Коли платформа створює тривожний запис, він зберігає час початку та закінчення тривожного запису. За замовчуванням час початку та час закінчення однакові. Якщо стан тригера повторюється, платформа оновлює час завершення. Платформа може автоматично стирати сигнал тривоги, коли відбувається подія, яка відповідає умові очищення нагадування. Умови очищення сигналізації не є обов’язковими. Користувач може очистити сигнал тривоги вручну.

Крім активного та очищеного стану тривоги, Платформа також відстежує, чи хтось підтвердив тривогу. Підтвердження нагадування можливе за допомогою віджета інформаційної панелі або вкладки інформації про об’єкт.

Існує 4 можливі значення поля «status»:

- **Активний непідтверджений** – тривожний сигнал не очищений і ще не підтверджений;
- **Active acknowledged** – тривога не очищена, але вже підтверджена;
- **Cleared unacknowledged** – сигнал уже знято, але ще не підтверджено;
- **Cleared acknowledged** – тривога вже була очищена та підтверджена.

5. Static widget.

Відображає статичний вміст HTML (наприклад, «HTML-картка»). Статичні віджети не використовують жодних джерел даних і зазвичай налаштовуються шляхом вказівки статичного вмісту HTML і, за бажанням, стилів CSS. Прикладом статичного віджета є «HTML-картка», яка відображає вказаний вміст HTML (Рис. 3.8).

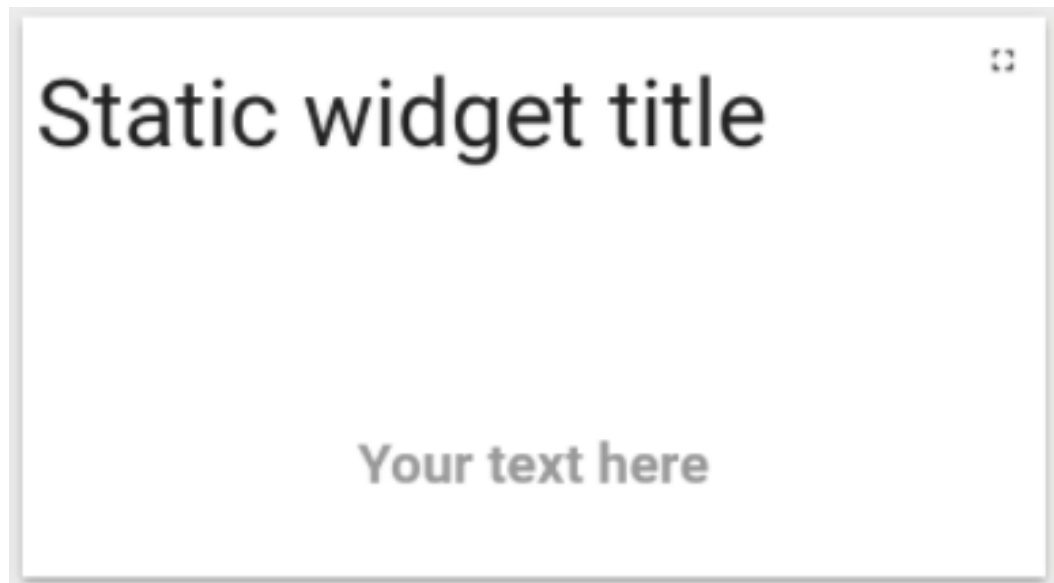


Рисунок 3.8 - HTML-картка

Для побудови інформаційної системи було використано такі віджети:

1. HTML картки;
2. Карти;
3. Таблиці;
4. Лінійну та стовпчикову діаграму.

3.3 Розробка схеми інформаційної системи

Автоматизована система «Обліку води» дозволить відстежувати кількість витраченої води в реальному часі або переглянути графіки витрачення води за конкретний період часу.

До функцій системи включений автоматичний підрахунок кількості кубометрів води по всій агрегації або по конкретному користувачу, що допоможе більш розумно використовувати водяні запаси земні.

Унікальність інформаційної системи:

1. Дана система являється масштабованою, що дозволить легко впровадити новий функціонал до системи не змінюючи архітектури. В основі системи лежить сервіс-орієнтована архітектура.
2. Планується що система буде працювати на імпульсних лічильниках. Тобто самі лічильники змінювати не доведеться, так як вони найпопулярніші, і встановлені майже в кожній оселі. Якщо вже встановлено цифровий лічильник, то слід перевірити чи сумісний він з платформою.
3. В систему досить легко впровадити підрахунок вартості за кожен місяць.
4. Системою можна буде керувати з усіх пристроїв які підключаються до мережі інтернет та мають веб-браузер.

На рисунку 3.9 представлена схема інформаційної системи «Обліку води».

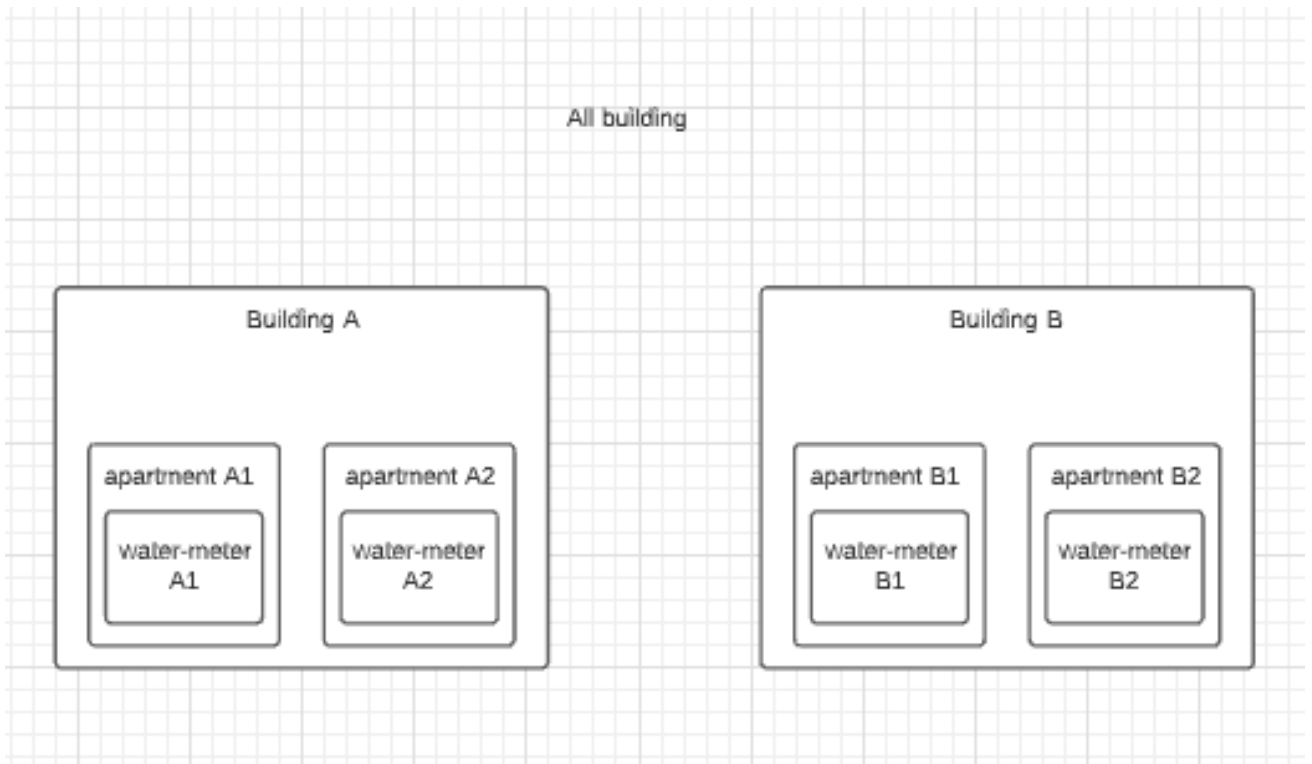


Рисунок 3.9 – Схема Інформаційної системи

Створення та конфігурація профілю пристрою описана на рисунках 3.10 – 3.11. До профілю пристрою вказується його назва та ланцюг правил який буде використовувати цей профіль.

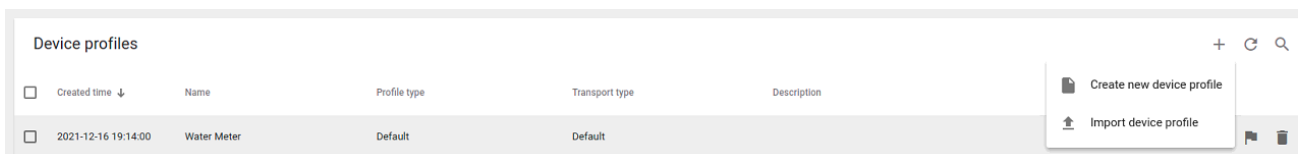


Рисунок 3.10 – Створення профілю пристрою

Add device profile ? ×

1 **Device profile details** — 2 **Transport configuration** Optional — 3 **Alarm rules (0)** Optional — 4 **Device provisioning** Optional

Name *
Water

Rule chain
Root Rule Chain ×

Mobile dashboard
Used by mobile application as a device details dashboard

Queue Name
Select from a drop-down list.

Device profile image
No image selected
Drop an image or click to select a file to upload. ×

Maximum upload file size: 512.0 KB

Description

Next: Transport configuration

Cancel Add

Рисунок 3.11 – Конфігурація налаштувань профілю пристрою

Також на профіль пристрою додаються правила тривог за параметрами. Процес створення правила показано на рисунку 3.12 – 3.13.

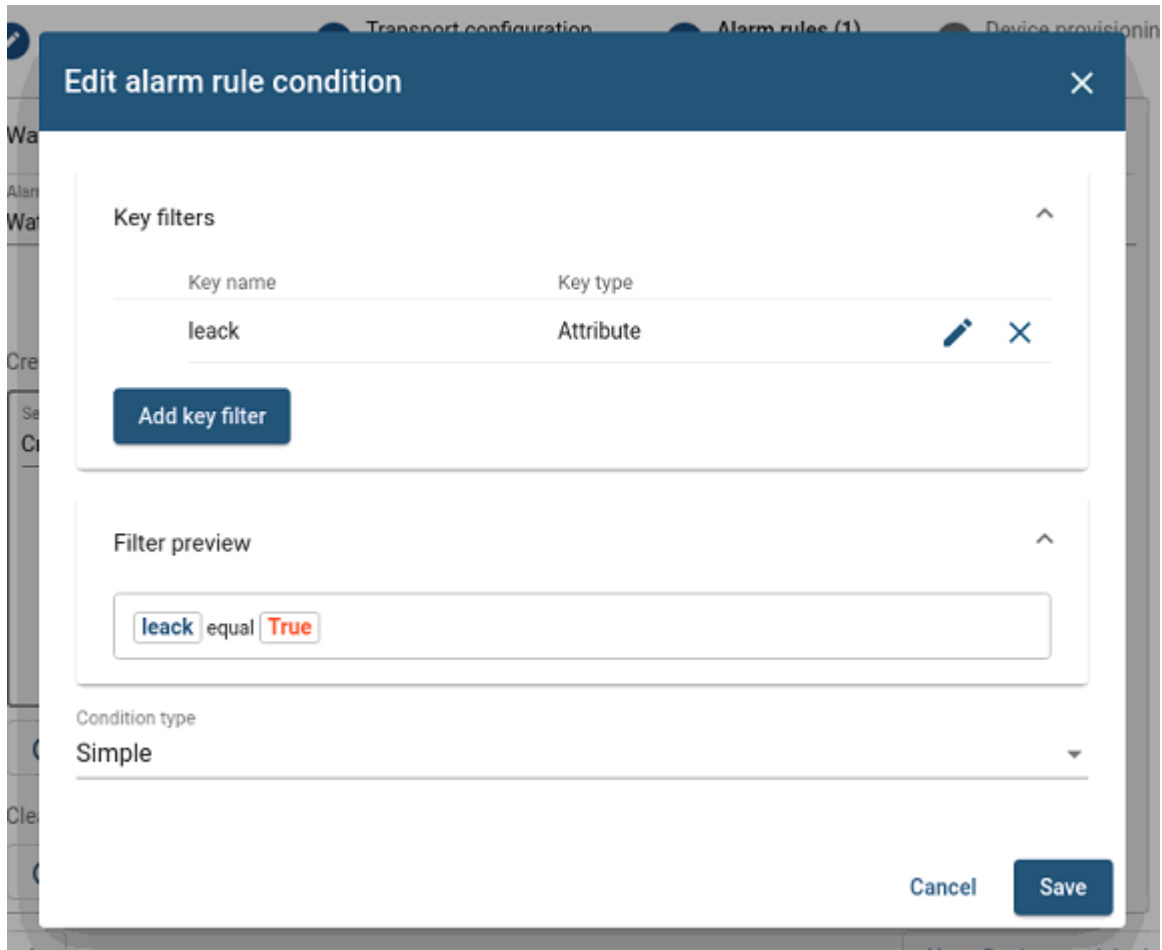


Рисунок 3.12 – Конфігурація правил тривоги

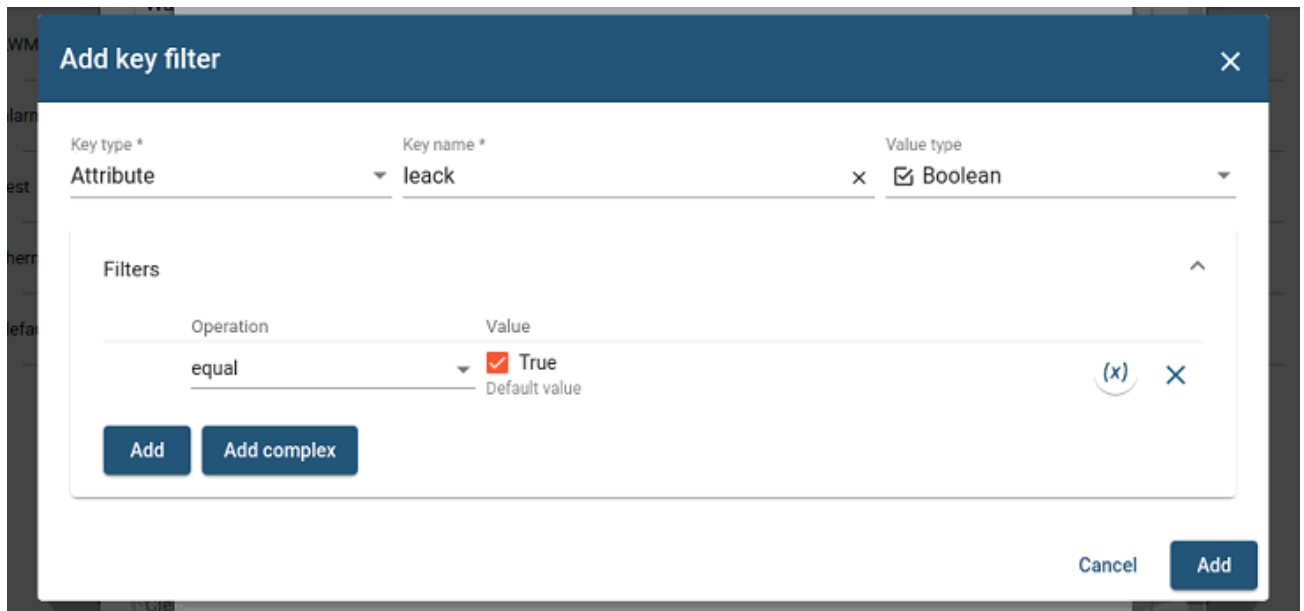


Рисунок 3.13 – Додавання правил за ключами

На рисунку 3.14 – 3.22 представлено всі правила які використовуються в інформаційній системі.

Daily Consumption Threshold Exceeded ^

Alarm type
Daily Consumption Threshold Exceeded

Advanced settings v

Create alarm rules

Severity
Critical v

`dailyConsumption` greater than `10000000` and `dailyConsumptionAI...` equal `Current tenant` `dailyConsumptionAI...`

Schedule: Active all the time

Details: `$(dailyConsumption)`

Рисунок 3.14 – Правило перевищення щоденного використання води

Weekly Consumption Threshold Exceeded ^

Alarm type
Weekly Consumption Threshold Exceeded

Advanced settings v

Create alarm rules

Severity
Critical v

`weeklyConsumption` greater or equal `1000000000` and `alarmEnabled` equal `Current tenant` `weeklyConsumption...`

Schedule: Active all the time

Details: `$(weeklyConsumption)`

Рисунок 3.15 – Правило перевищення недільного використання води

Low Battery ^

Alarm type
Low Battery

Advanced settings v

Create alarm rules

Severity
Warning v

`battery` less than `0` and `alarmEnabled` equal `Current tenant` `lowBattAlarmEnabled`

Schedule: Active all the time

Details: `$(battery)`

Рисунок 3.16 – Правило низького заряду батареї на пристроях

Low Temperature ^

Alarm type
Low Temperature

Advanced settings ▾

Create alarm rules

Severity
Warning ▾

`temperature` less or equal `1000` and `alarmEnabled` equal `Current tenant` `lowTempAlarmEnab...`

Schedule: Active all the time

Details: `$(temperature)`

Рисунок 3.17 – Правило низького температури

Device Inactive ^

Alarm type
Device Inactive

Advanced settings ▾

Create alarm rules

Severity
Major ▾

`active` equal `False` and `alarmEnabled` equal `Current tenant` `inactivityAlarmEnab...`

Schedule: Active all the time

Рисунок 3.18 – Правило активності пристрою

Leakage Detected ^

Alarm type
Leakage Detected

Advanced settings ▾

Create alarm rules

Severity
Critical ▾

`leakage` equal `True` and `alarmEnabled` equal `Current tenant` `leakageAlarmEnabled`

Schedule: Active all the time

Рисунок 3.19 – Правило виявлення витікання

Low Temperature Customer Alarm ^

Alarm type
Low Temperature Customer Alarm

Advanced settings v

Create alarm rules

Severity
Warning v

temperature less or equal 1000 and temperature equal Current customer cLowTempAlarmEn... o

Schedule: Active all the time o

Details: \${temperature} o

Рисунок 3.20 – Правило низької температури для клієнтів

Daily Consumption Threshold Exceeded Customer Alarm ^

Alarm type
Daily Consumption Threshold Exceeded Customer Alarm

Advanced settings v

Create alarm rules

Severity
Critical v

dailyConsumption greater than 10000000 and dailyConsumptionAI... equal Current customer cDailyConsumption... o

Schedule: Active all the time o

Details: \${dailyConsumption} o

Рисунок 3.21 – Правило перевищення щоденного використання води для клієнтів

Weekly Consumption Threshold Exceeded Customer Alarm ^

Alarm type
Weekly Consumption Threshold Exceeded Customer Alarm

Advanced settings v

Create alarm rules

Severity
Critical v

weeklyConsumption greater or equal 1000000000 and alarmEnabled equal Current customer cWeeklyConsumptio... o

Schedule: Active all the time o

Рисунок 3.22 – Правило перевищення недільного використання води для клієнтів

Майже за такою же схемою створюються пристрої (рис. 3.23 – 3.24) та активи (Рис. 3.25). Пристрої в свою чергу зв'язуються зв'язками з профілем (рис. 3.26) пристрою а активи зв'язуються з активами (рис 3.27).

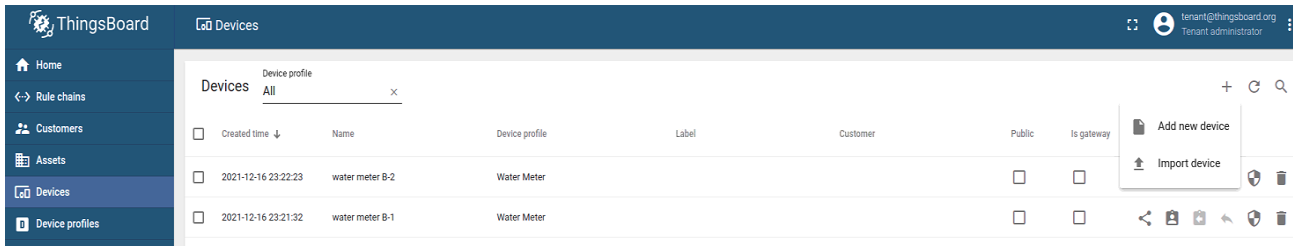


Рисунок 3.23 – Створення пристроїв

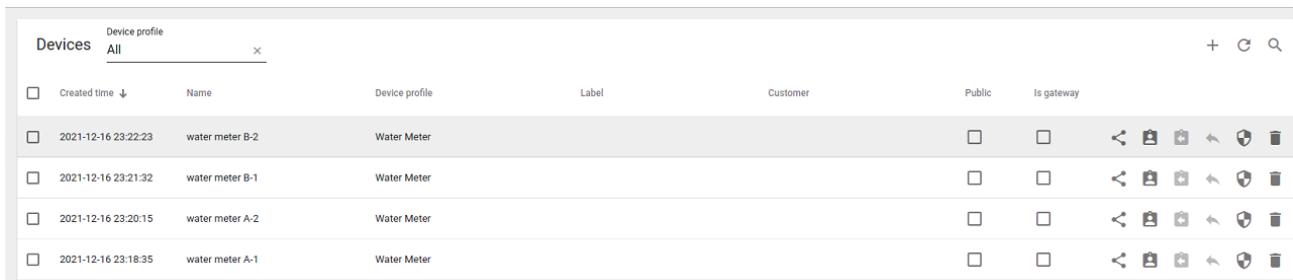


Рисунок 3.24 – Список створених пристроїв

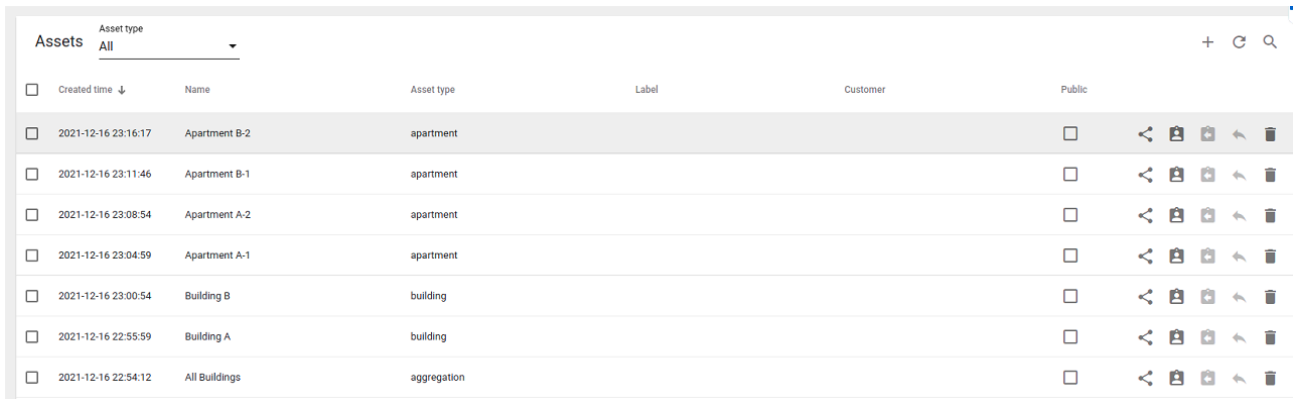


Рисунок 3.24 – Список створених активів

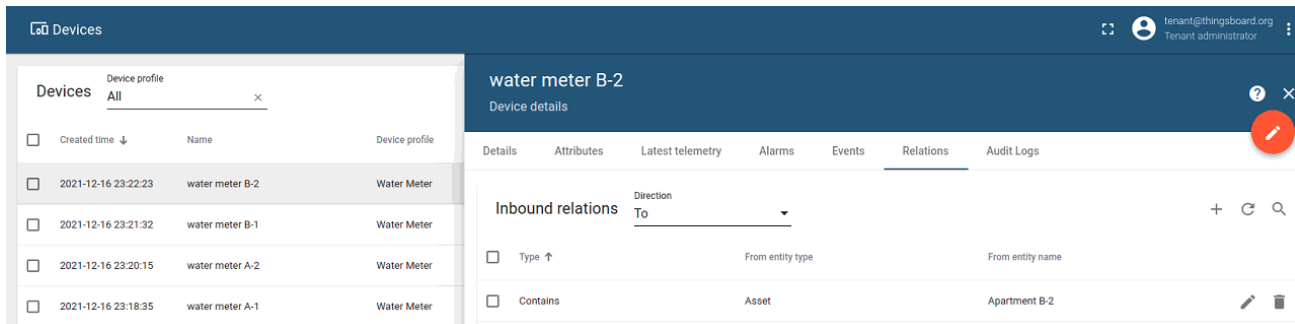


Рисунок 3.25 – Зв'язок пристрій - актив

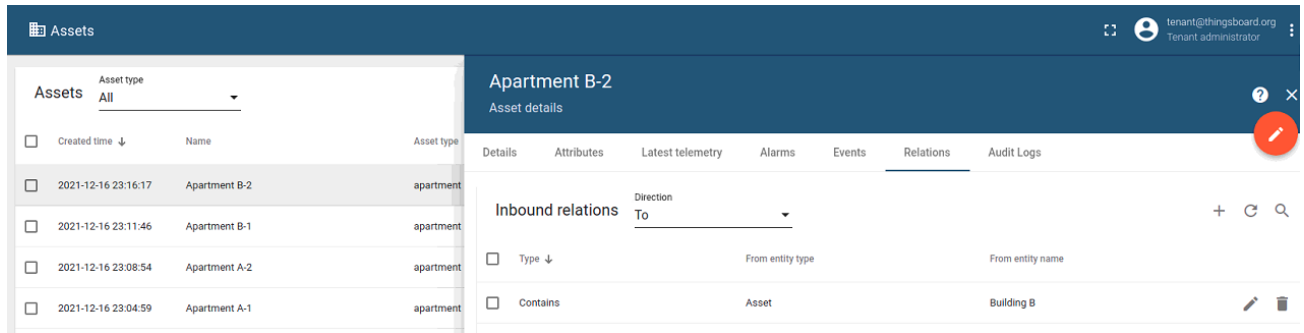


Рисунок 3.27 – Зв'язок актив - актив

3.2 Проектування ланцюгів правил

Ядро правил – це проста у використанні платформа для побудови робочих процесів на основі подій. Існує 3 основних компоненти:

Повідомлення – будь-яка вхідна подія. Це можуть бути вхідні дані з пристроїв, подія життєвого циклу пристрою, подія REST API, запит RPC тощо.

Вузол правила - функція, яка виконується для вхідного повідомлення. Існує багато різних типів вузлів, які можуть фільтрувати, трансформувати або виконувати певні дії щодо вхідного повідомлення.

Ланцюжок правил – вузли пов'язані один з одним відносинами, тому вихідне повідомлення від вузла правила надсилається наступним підключеним вузлом правила.

Ядро правил – це добре структура яка налаштовується для складної обробки подій. Ось кілька випадків, які можна налаштувати за допомогою ланцюжків правил:

- Перевірка та зміна даних для вхідної телеметрії або атрибутів перед збереженням у базі даних.
- Скопіюйте телеметрію або атрибути з пристроїв у пов'язані активи, щоб ви могли об'єднати телеметрію. Наприклад, дані з кількох пристроїв можна об'єднати у пов'язаний актив.

- Створити/оновити/очистити нагадування на основі визначених умов.
- Запускати дії на основі подій життєвого циклу пристрою. Наприклад, створюйте сповіщення, якщо пристрій працює в режимі онлайн/офлайн.
- Завантажити додаткові дані, необхідні для обробки. Наприклад, порогове значення температури навантаження для пристрою, яке визначено в атрибуті Клієнт або Орендар пристрою.
- Ініціювати виклики REST API до зовнішніх систем.
- Надсилати електронні листи, коли відбувається складна подія, і використовуйте атрибути інших об'єктів у шаблоні електронної пошти.
- Враховувати налаштування користувача під час обробки події.
- Здійснити виклики RPC на основі визначених умов.

Для реалізації інформаційної системи було розроблено наступні ланцюги правил:

- Головний ланцюг зображено на рисунку 3.28;

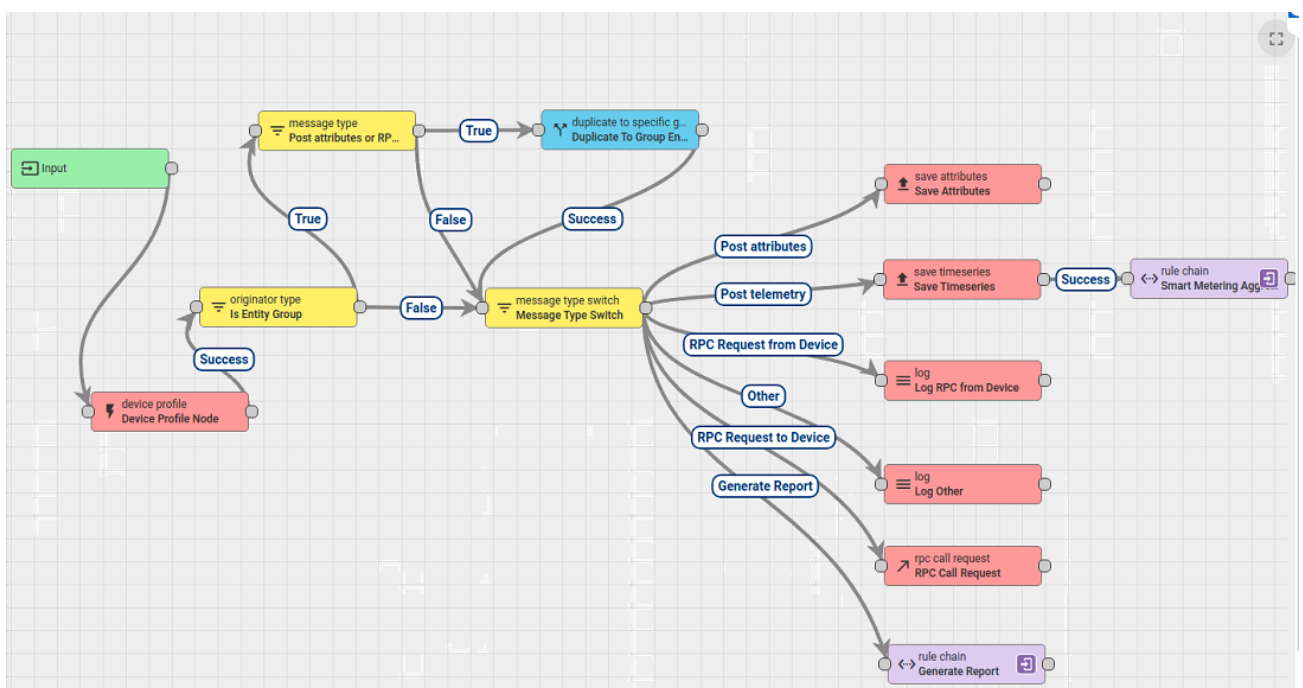


Рисунок 3.28 – Головний ланцюг правил

- Ланцюг для реалізації системи обліку води зображено на рисунку 3.29;

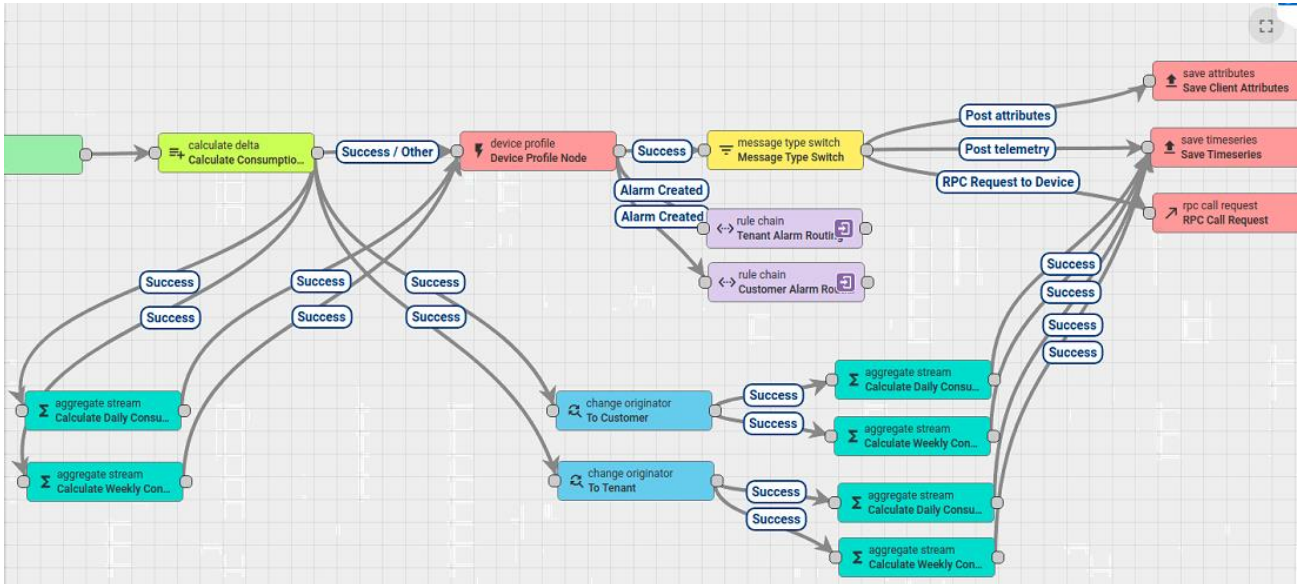


Рисунок 3.29 – Ланцюг реалізації системи обліку води

- Ланцюг для клієнтських тривог зображено на рисунку 3.30 – 3.31;

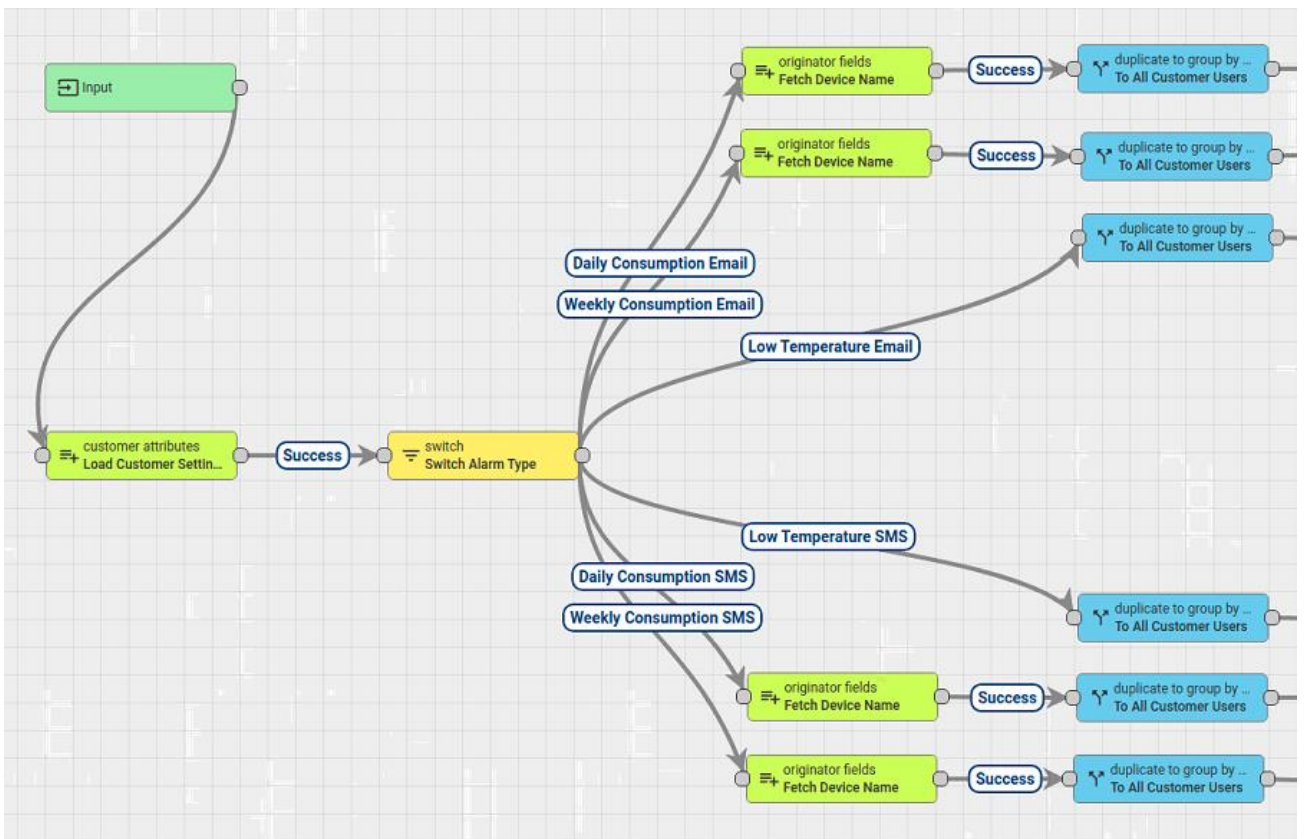


Рисунок 3.30 – Ланцюг тривог для клієнтів перша частина

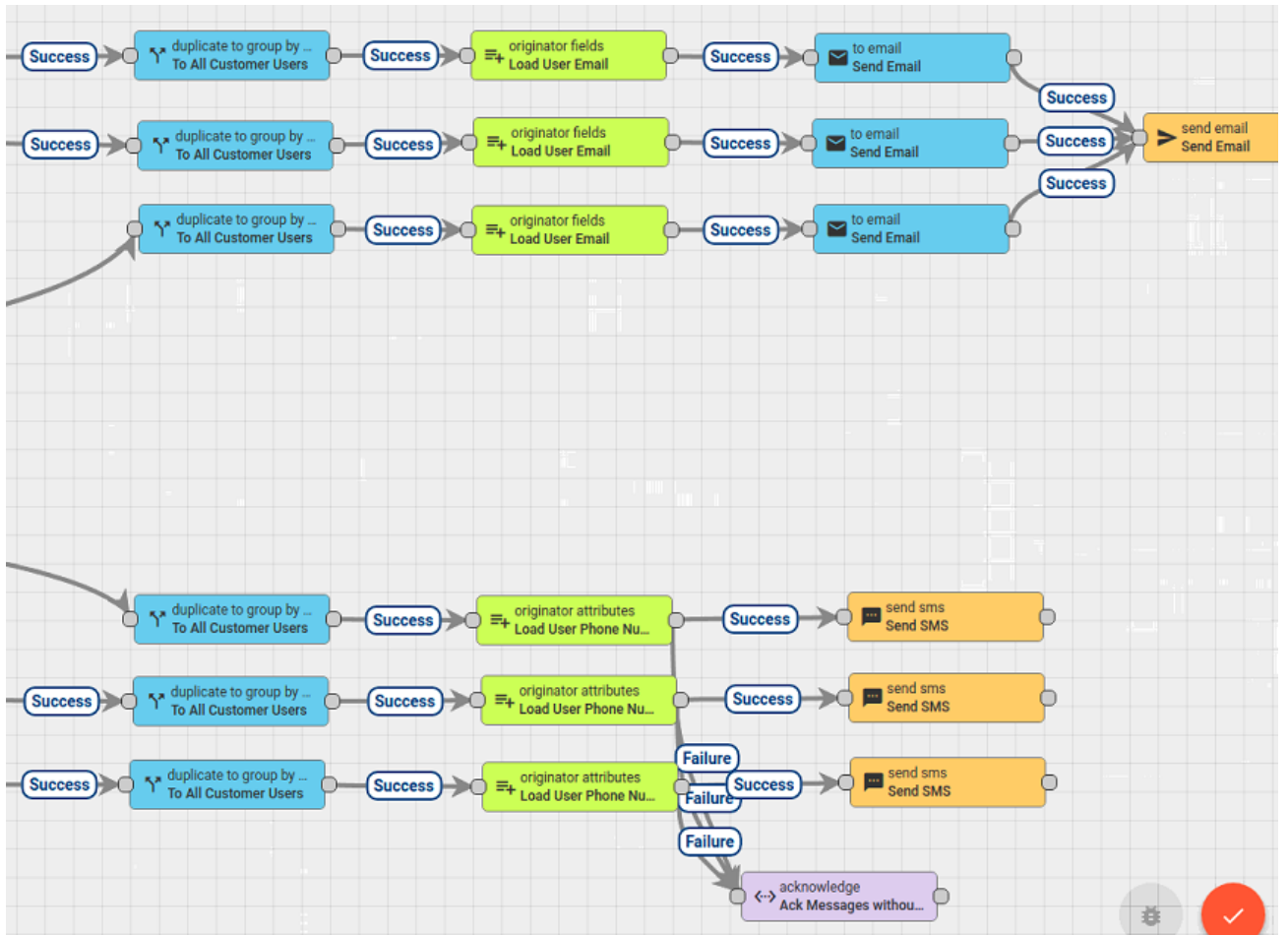


Рисунок 3.30 – Ланцюг тривог для клієнтів друга частина

- Ланцюг тривог для адміністраторів зображено на рисунку 3.32 – 3.33;

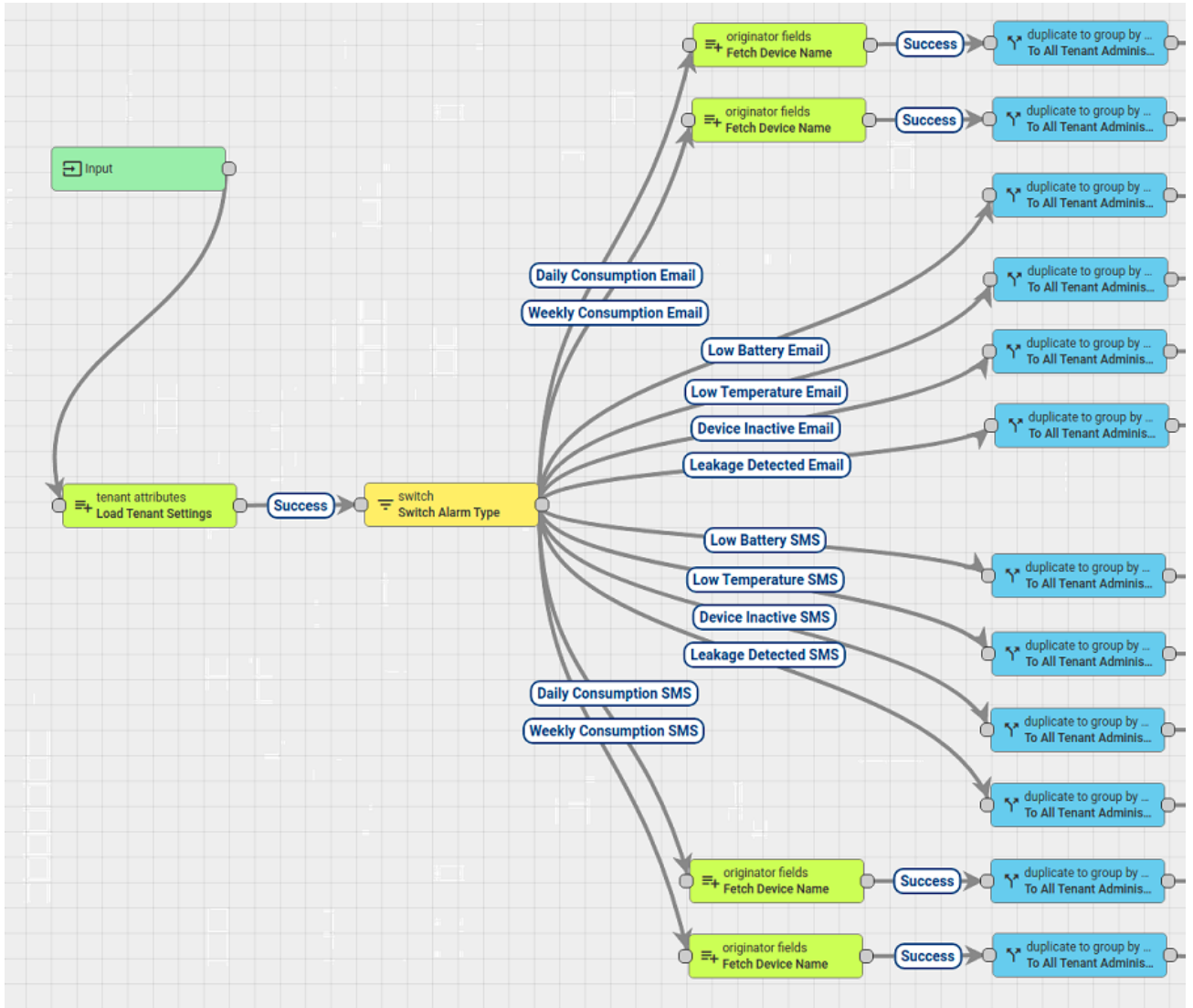


Рисунок 3.32 – Ланцюг тривоги для адміністраторів перша частина

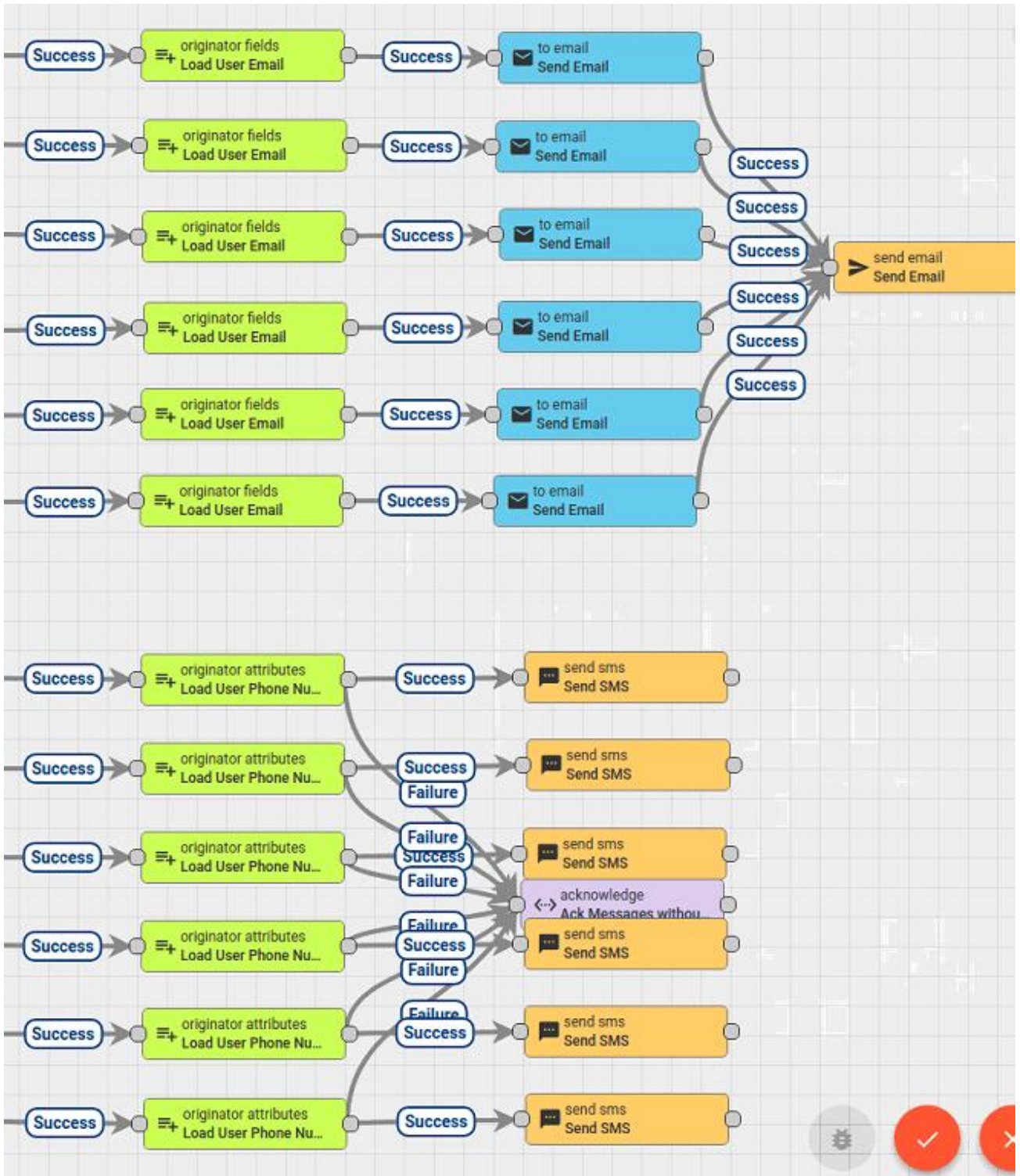


Рисунок 3.33 – Ланцюг тривиг для адміністраторів друга частина

3.4 Опис роботи інформаційної системи

Всі результати які віддають пристрої виводяться на одна панель відображення на яку ми можемо відстежувати всю інформацію яка візуалізується в правильних форматах.

Показники по витраті води по кожному пристрою сумуються та виводяться на HTML картці, також виводяться показники по всім активностям пристроїв, та їх заряд батареї. Пристрої по кожному з будинку також несуть в собі інформацію про своє розташування, яке в свою чергу приводиться до правильного формату даних та виводиться маркерами на карті.

Всю інформацію досить зручно та інформативно переглядати за допомогою таблиць, також всі тривоги повертаються в вигляді табличного запису.

Також на панель відображення інтегровано колонкову діаграму, яка виводить інформацію за останні сім днів.

Загальний вигляд панелі відображення представлено на рисунку 3.34.

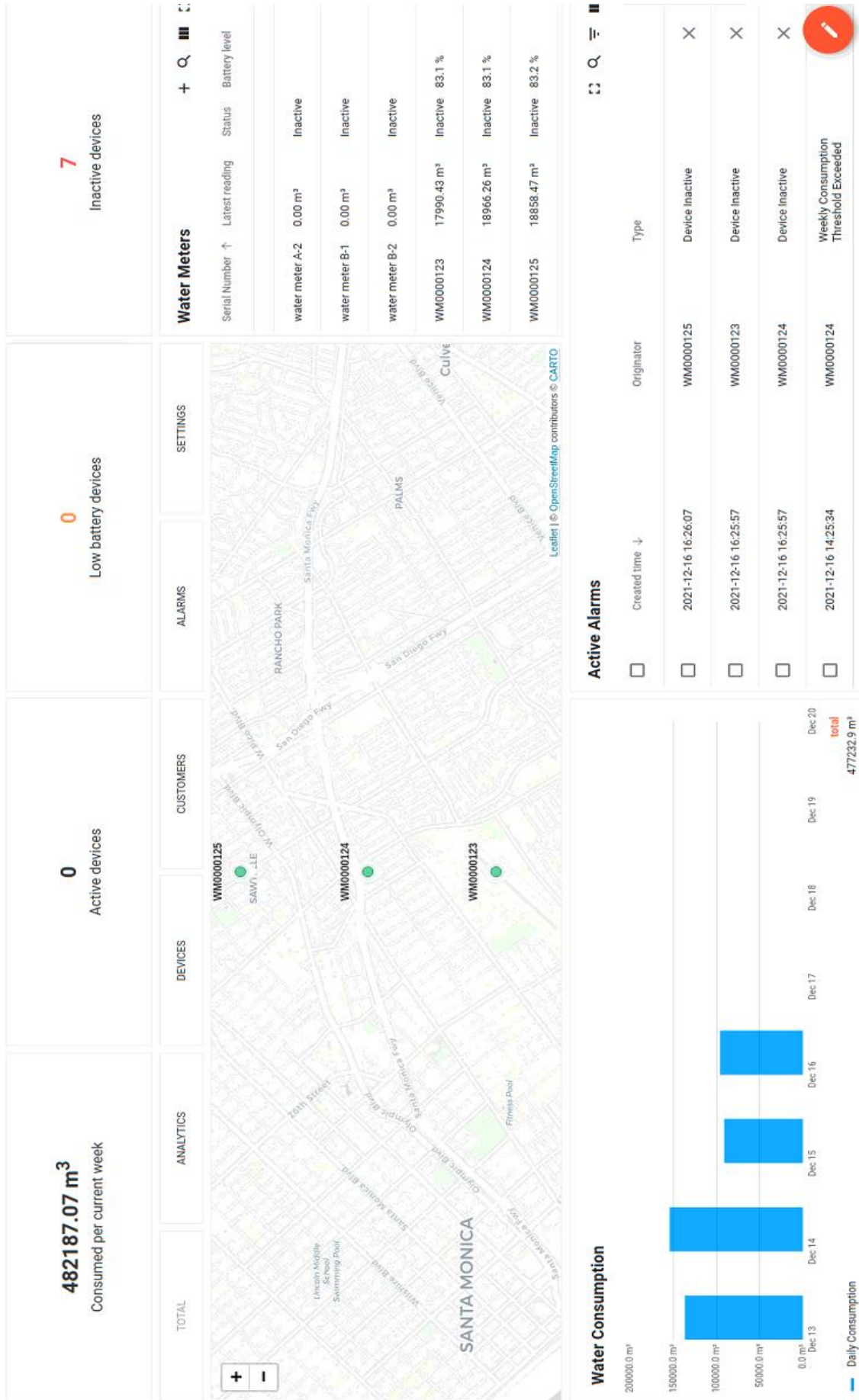


Рисунок 3.34 – Загальний вигляд панелі відображення

На панель відображення виводиться аналітика за витрачену кількість води, виводиться ця інформація на лінійну діаграму, що зображена на рисунку 3.35.

Даний графік аналітики виводить дані за сім днів вибраного відрізка часу, та додатково налаштовано вивід другого графіку (графік сірого кольору) в цей же період часу рік тому. Тому ми можемо легко проводити аналіз використання ресурсів за різними часовими відрізками.



Рисунок 3.35 – Аналітика по витраченому об'єму води

На рисунку 3.36 зображено таблицю тривоги на яку виводяться всі записи по тривогам.

Alarms						🔍	☰	☰
<input type="checkbox"/>	Created time ↓	Originator	Type	Severity	Status			
<input type="checkbox"/>	2021-12-16 16:26:07	WM0000125	Device Inactive	Major	Active Unacknowledged	...	✓	✕
<input type="checkbox"/>	2021-12-16 16:25:57	WM0000123	Device Inactive	Major	Active Unacknowledged	...	✓	✕
<input type="checkbox"/>	2021-12-16 16:25:57	WM0000124	Device Inactive	Major	Active Unacknowledged	...	✓	✕
<input type="checkbox"/>	2021-12-16 14:25:34	WM0000124	Weekly Consumption Threshold Exceeded	Critical	Active Unacknowledged	...	✓	✕
<input type="checkbox"/>	2021-12-16 14:25:32	WM0000124	Daily Consumption Threshold Exceeded	Critical	Active Unacknowledged	...	✓	✕
<input type="checkbox"/>	2021-12-16 14:25:30	WM0000123	Weekly Consumption Threshold Exceeded	Critical	Active Unacknowledged	...	✓	✕
<input type="checkbox"/>	2021-12-16 14:25:29	WM0000123	Daily Consumption Threshold Exceeded	Critical	Active Unacknowledged	...	✓	✕
<input type="checkbox"/>	2021-12-16 14:25:28	WM0000125	Daily Consumption Threshold Exceeded	Critical	Active Unacknowledged	...	✓	✕
<input type="checkbox"/>	2021-12-16 14:25:27	WM0000125	Weekly Consumption Threshold Exceeded	Critical	Active Unacknowledged	...	✓	✕

Рисунок 3.36 – Таблиця тривоги для інформаційної системи

Додатково деякі налаштування можна проводити за допомогою віджетів, приклад таких налаштувань зображено на рисунку 3.37.

System Alarms	
<input checked="" type="checkbox"/>	LOW BATTERY ALARM
LOW BATTERY THRESHOLD	
20	
<input checked="" type="checkbox"/>	LOW TEMPERATURE ALARM
LOW TEMPERATURE THRESHOLD	
19	
<input checked="" type="checkbox"/>	EXCEEDING DAILY CONSUMPTION ALARM
DAILY CONSUMPTION THRESHOLD	
100	
<input checked="" type="checkbox"/>	EXCEEDING WEEKLY CONSUMPTION ALARM
WEEKLY CONSUMPTION THRESHOLD	
500	
<input checked="" type="checkbox"/>	INACTIVITY ALARM

Рисунок 3.36 – Таблица тривог для інформаційної системи

ВИСНОВКИ

У ході виконання дипломного проекту було досліджено методи обробки та візуалізації інформації та побудовано на базі IoT-платформи ThingsBoard Customer Edition інформаційну систему обліку води в приватних будинках поділених на апартаменти (дуплекси). Таким чином, в ході роботи над дипломним проектом були дослідженні існуючі технології IoT (Інтернет речей).

Перш за все, було досліджено методи візуалізації інформації та методи її обробки, за результатами досліджень найкращий спосіб подання інформації залишається таблиці, тобто подання інформації в табличному вигляді.

Також в ході роботи було досліджено в цілому що таке IoT платформа та визначено їх характеристики, представлено та описано загальну архітектуру та визначено мету платформи. Серед великої кількості платформ було обрано Thingsboard CE. Це передумовлено на сам перед тим, що платформа являється Open-source проектом з відкритим вихідним кодом. Також слід зазначити, що платформа є досить гнучкою та підходить майже під всі застосування.

Створеною інформаційною системою можна керувати з усіх пристроїв в яких встановлено браузер та є доступ до мережі інтернет з усіх куточків світу, так як платформа реалізована на клієнт серверній архітектурі, та являється кроссбраузерною та мультиплатформенною системою.

Роботу над інформаційною системою планується продовжити. В поточній версії система повністю працездатна та готова до масштабування. В наступних версіях планується створити відмет який буде відображати вартість використаних ресурсів, для цього потрібно знайти API яке б передавало вартість по кожному регіону для автоматичної обробки або написати власне API. Також в наступних версіях планується реалізація опали по вирахуваній ціні, для цього потрібно дослідити API різних банків та вибрати більш підходящу реалізацію. Але така реалізація буде заточена під якусь конкретну країну.

На відміну від існуючих рішень реалізації інформаційної системи, розроблене рішення з використанням IoT платформи є добре розширюваним та масштабованим або навпаки таку реалізацію легко спростити до мінімального функціоналу. Технічні та нетехнічні рішення які використовувалися під час побудови системи є досить універсальними, та можуть в подальшому використовуватися в побудові інших інформаційних систем.

Платформа підтримує підключення десятків тисяч пристроїв до своєї бази в одно потоковому режимі, що дозволяє реалізувати досить велику інформаційну систему на своїй базі. В свою чергу запуск платформи в кластерному режимі роботи дозволяє запускати на своїй базі мільйони пристроїв, що дозволяє реалізовувати такі проекти як Smart City або інші в яких необхідно використання великої кількості пристроїв.

ПЕРЕЛІК ПОСИЛАНЬ

1. Ala Al-Fuqaha, Mohsen Guizani, Mehdi Mohammadi, Mohammed Aledhari, and Moussa Ayyash. Internet of things: A survey on enabling technologies, protocols, and applications. *IEEE Communications Surveys & Tutorials*, 17 (4): 2347-2376, 2015.
2. Pallavi Sethi and Smruti R. Sarangi. Internet of things: Architectures, protocols, and applications. 2017: 1-25, 012017.
3. ThingsBoard Documentation [Електронний ресурс] – Режим доступу до ресурсу: <https://thingsboard.io/docs/>.
4. ThingsBoard CE Open-source IoT Platform [Електронний ресурс] – Режим доступу до ресурсу: <https://github.com/thingsboard/thingsboard>.
5. Могильный С. Микрокомп'ютер Raspberry Pi - інструмент дослідника / С. Могильный. // Талком. – 2016. – 340с.
6. Arduino [Електронний ресурс] – Режим доступу до ресурсу: <https://www.arduino.cc/>.
7. Raspberry Pi [Електронний ресурс] – Режим доступу до ресурсу: <https://www.raspberrypi.org/>.
8. Росляков А. В. Интернет вещей: Учебное пособие / А. В. Росляков, С. В. Ваняшин, А. Ю. Гребешков. — Самара: ПГУТИ, 2015 — 136 с.
9. D. Evans, “The Internet of things: How the next evolution of the Internet is changing everything”, CISCO, San Jose, CA, USA, White Paper, 2011.
10. R. Khan, S. U. Khan, R. Zaheer, and S. Khan, “Future Internet: The Internet of Things architecture, possible applications and key challenges,” in Proc. 10th Int. Conf. FIT, 2012, pp. 257—260.
11. Васильков, А. Микрокомпьютеры для интернета вещей: от умного дома к поумневшему окружению [текст] / А. Васильков // Компьютерра, 14 июня 2013г. 109

12. Восков, Л.С. Web вещей — новый этап развития интернета вещей [текст] / Л.С. Восков , Н.А. Пилипенко // Качество. Инновации. Образование. — 2013. — № 2. — С. 44—49.
13. Гиббс, М. Интернет вещей – не только для «умных» [текст] / М. Гиббс // Сети/network world. — 2013. — №3.
14. Архитектура безопасности "Интернета вещей" [Электронный ресурс]. – 2018. – Режим доступа до ресурсу: <https://docs.microsoft.com/ru-ru/azure/iot-suite/iotsecurity-architecture>. .
15. AWS IoT Button [Электронный ресурс] – Режим доступа до ресурсу: <https://aws.amazon.com/ru/iotbutton/>.
16. V. A. F. Almeida, D. A. Menasc.e, R. Riedi, F. P. Ribeiro, R. Fonseca, and W. Meira Jr., "Analyzing Web Robots and their Impact on Caching," Proc. Sixth Workshop on WebCaching and Content Distribution, Boston, Massachusetts, June 20-22, 2001.
17. M. Arlitt and C. Williamson, "Web Server Workload Characterization: the Search for Invariants," Proc. 1996 ACM SIGMETRICS Conf. Measurement Comput. Syst., Philadelphia, Pennsylvania, May 1996, pp. 126-137.
18. T. Berners-Lee, R. Cailliau, H. Nielsen, and A. Pecret, "The World Wide Web," Comm. ACM, vol. 37, no. 8, pp. 76-82, Aug. 1994.
19. Дунаев, В. Сценарии для Web-сайта. PHP и JavaScript / В. Дунаев. - М.: БХВ-Петербург, 2017. - 576 с.
20. 10 Best IoT Platfroms for 2021[Электронный ресурс] – Режим доступа до ресурсу: <https://www.cybervisiontech.com/blog/10-best-iot-platforms-2021>.

ДЕМОНСТРАЦІЙНІ МАТЕРІАЛИ



ДЕРЖАВНИЙ УНІВЕРСИТЕТ ТЕЛЕКОМУНІКАЦІЙ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ
ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ



Кафедра Інформаційних систем та технологій

Дослідження та проектування інформаційної системи
обробки та візуалізації даних на основі IoT-платформи
ThingsBoard

Виконав: студентка групи ІСДМ-61
Джерелейко А.О.

2021

Мета роботи: дослідити методи обробки та візуалізації інформації та проектування інформаційної системи на досліджених результатах.

Об'єкт дослідження: процес створення інформаційної системи обробки та візуалізації даних.

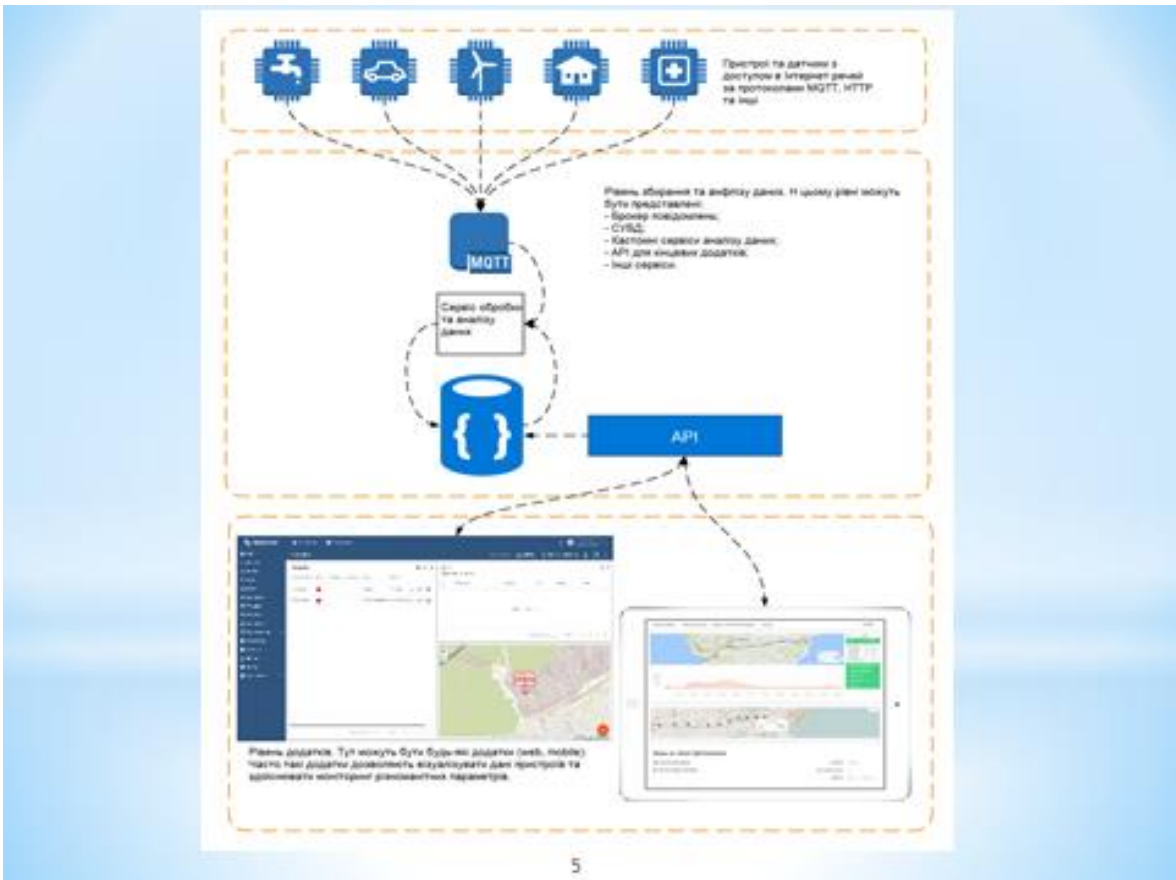
Предмет дослідження: обробка та візуалізація інформації.

Наукова новизна магістерської роботи: полягає у можливості застосування отриманих результатів для реалізації інформаційних систем з правильною обробкою та візуалізацією даних.

2



3



* ThingsBoard

ThingsBoard – це платформа IoT з відкритим кодом, яка забезпечує швидкий розвиток, управління та масштабування проектів IoT. Головна мета платформи надати готове хмарне або локальне рішення IoT, яке забезпечить серверну інфраструктуру для додатків IoT.

Для побудови інформаційної системи використовується остання версія платформи: 3.3.2.

Реалізація платформи:

- Back-end – Java;
- Front-end – Angular 11, JavaScripts, HTML, CSS;
- База даних – PostgreSQL;
- Бібліотеки - React JSON Form, Chart.js, Leaflet та багато інших.

6

The screenshot displays the ThingsBoard interface. At the top, a dashboard titled "All building" is shown on a grid background. It contains two main panels: "Building A" and "Building B". "Building A" contains two sub-panels for "apartment A1" and "apartment A2", each with a "water-meter" widget. "Building B" contains two sub-panels for "apartment B1" and "apartment B2", each with a "water-meter" widget.

Below the dashboard is the "Assets" table, which lists the following data:

Created time	Name	Asset type	Label	Custom	Public
2021-12-16 22:16:17	Apartment B2	apartment			<input type="checkbox"/>
2021-12-16 22:17:46	Apartment B1	apartment			<input type="checkbox"/>
2021-12-16 22:08:54	Apartment A2	apartment			<input type="checkbox"/>
2021-12-16 22:04:59	Apartment A1	apartment			<input type="checkbox"/>
2021-12-16 22:00:54	Building B	building			<input type="checkbox"/>
2021-12-16 22:00:59	Building A	building			<input type="checkbox"/>
2021-12-16 22:04:12	All Buildings	aggregation			<input type="checkbox"/>

7

Device profiles

Created time	Name	Profile type	Transport type	Description
2021-12-16 19:14:00	Water Meter	Default	Default	

Leakage Detected

Alarm type: Leakage Detected

Create alarm rules

Severity: Critical

leakage | equal | True | and | alarmEnabled | equal | Current tenant | leakageAlarmEnabled

Schedule: Active all the time

Devices

Created time	Name	Device profile	Label	Customer	Public	In priority
2021-12-16 23:22:23	water meter B-2	Water Meter			<input type="checkbox"/>	<input type="checkbox"/>
2021-12-16 23:21:30	water meter B-1	Water Meter			<input type="checkbox"/>	<input type="checkbox"/>
2021-12-16 23:20:15	water meter A-2	Water Meter			<input type="checkbox"/>	<input type="checkbox"/>
2021-12-16 23:18:35	water meter A-1	Water Meter			<input type="checkbox"/>	<input type="checkbox"/>

water meter B-2

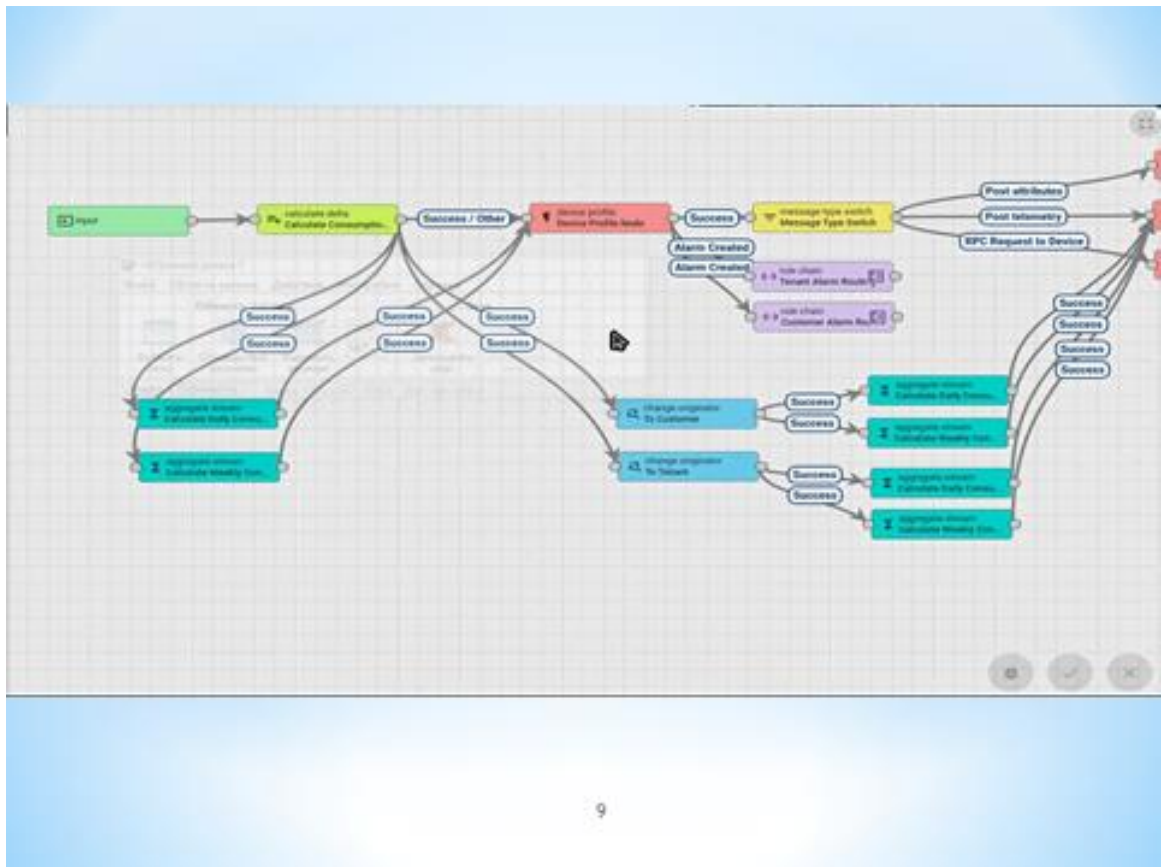
Device details

Details | Attributes | Latest telemetry | Alarms | Events | Relations | Audit Logs

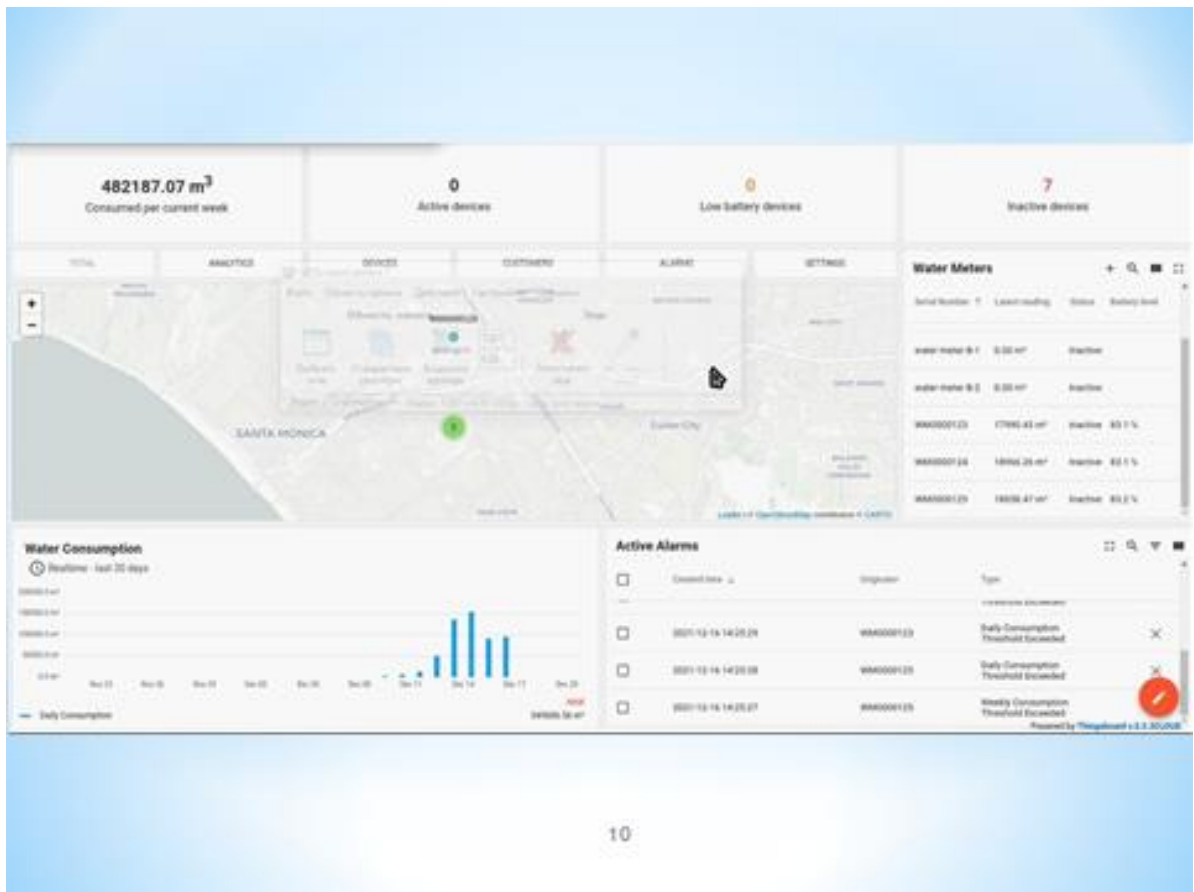
Inbound relations

Type	From entity type	From entity name
Contains	Asset	Apartment B-2

8



9



10

Унікальність інформаційної системи:

- Дана система являється масштабованою, що дозволить легко впровадити новий функціонал до системи не змінюючи архітектури. В основі системи лежить сервіс-орієнтована серверна архітектура.
- Планується що система буде працювати на імпульсних лічильниках. Тобто самі лічильники змінювати не доведеться, так як вони найпопулярніші, і встановлені майже в кожній оселі. Якщо вже встановлено цифровий лічильник, то слід перевірити чи сумісний він з платформою.
- В систему досить легко впровадити підрахунок вартості за кожен місяць.
- Системою можна буде керувати з усіх пристроїв які підключаються до мережі інтернет та мають веб-браузер.

11

* ВИСНОВКИ

У результаті виконання даної магістерської роботи було отримано наступні результати:

- досліджено методи візуалізації інформації та методи її обробки, за результатами досліджень найкращий спосіб подання інформації залишається таблиці, тобто подання інформації в табличному вигляді. Таким чином, в ході роботи над дипломним проектом були дослідженні існуючі технології IoT (Інтернет речей);
- досліджено IoT платформу ThingsBoard та на результатах визначено характеристики, представлено та описано загальну архітектуру та визначено мету платформи;
- реалізовано інформаційну систему обліку води в приватних будинках поділених на дві частини на базі Open-source IoT платформи Thingsboard CE.

На відміну від існуючих рішень реалізації інформаційної системи, розроблене рішення з використанням IoT платформи є добре розширюваним та масштабованим або навпаки таку реалізацію легко спростити до мінімального функціоналу. Технічні та нетехнічні рішення які використовувалися під час побудови системи є досить універсальними, та можуть в подальшому використовуватися в побудові інших інформаційних систем.

12

Апробація результатів магістерської роботи:

1. Джерелейко А. О. (Dzhereleyko A. O.), Яковенко Н. Д. (Yakovenko N. D.), Марченко Г. В. (Marchenko H. V.), Аташкаде Р. В. (Atashkade R. V.) «IoT у сучасному веб-розробленні» у фаховому виданні МОН «ЗВ'ЯЗОК» № 2 (2021)

2. Джерелейко А. О. (Dzhereleyko A. O.) «Спутникові технології в індустрії ІНТЕРНЕТ РЕЧЕЙ(IoT)» ВСЕУКРАЇНСЬКА НАУКОВО-ТЕХНІЧНА КОНФЕРЕНЦІЯ «СУЧАСНИЙ СТАН ТА ПЕРСПЕКТИВИ РОЗВИТКУ ІОТ» 9 квітня 2021 року

13

Дякую за увагу!