МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ В. Н. КАРАЗІНА
НАУКОВО-НАВЧАЛЬНИЙ ІНСТИТУТ
КОМП'ЮТЕРНОЇ ФІЗИКИ ТА ЕНЕРГЕТИКИ

Кафедра інформаційних технологій в фізико-енергетичних системах

ПОЯСНЮВАЛЬНА ЗАПИСКА

до дипломної роботи
за освітньо-кваліфікаційним рівнем бакалавр

«Пристрій збору інформації з використанням технології «Інтернету речей»»

Виконав: студент 4 курсу, групи НФ-43
Половинко Д.О.
спеціальність:
105 Прикладна фізика та наноматеріали
освітня програма: Прикладна фізика енергетичних систем
Керівник: ст. викладач Кофман О.Й.

Рецензент:

Харків – 2020

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Харківський національний університет імені В.Н.Каразіна

Навчально-науковий інститут комп'ютерної фізики та енергетики
Кафедра інформаційних технологій у фізико-енергетичних системах
Рівень вищої освіти (освітньо-кваліфікаційний рівень) бакалавр
Спеціальність 105 Прикладна фізика та наноматеріали

ЗАТВЕРДЖУЮ

Завідувач кафедри

_____ Руслан СУХОВ
підпис
"____" _____2020 року

З А В Д А Н Н Я

НА ДИПЛОМНУ РОБОТУ

Половинко Данило Олександрович

1.Тема роботи «Пристрій збору інформації з використанням технології «Інтернету речей»» керівник роботи: ст. викладач Кофман Олег Йосипович, затверджені наказом по університету від 5 грудня 2020 року № 0210–05/2249.

2. Строк подання студентом роботи 06.06.2020

3. Перелік питань, які потрібно розробити:

1) Розглянути принципи архітектури «Інтернету речей»;
2) Розробити структурну схему прилад для збору інформації;
3) Розробити програмне забезпечення для прилад для збору даних о параметрах навколишнього середовища;
4) Провести тестування пристрою у комплексі з приладом для вимірювання відносної вологості газів.

4. План роботи

№ з/п	Назви етапів роботи
1	Вибір теми.
2	Затвердження теми на кафедрі
3	Робота над літературою та іншими джерелами
4	Написання вступу до роботи та першого розділу
5	Написання другого розділу
6	Написання третього розділу
7	Завершення написання дипломної роботи (висновки та література)
8	Подача науковому керівнику
9	Відгук, рецензія, підготовка до захисту
10	Попередній захист
11	Захист

5.Дата видачі завдання

Студент _____ Данило ПОЛОВИНКО
підпис

Керівник роботи _____ Олег КОФМАН
підпис

РЕФЕРАТ

Данило Половинко «Прилад збору інформації з використанням технології «Інтернету речей»»

Кваліфікаційна робота бакалавра з прикладної фізики. – Х.: Харківський національний університет імені В. Н. Каразіна, 2020.

Справжня випускна кваліфікаційна робота містить результати порівняльного аналізу існуючих систем дистанційного контролю пристроїв Інтернету речей та розробку системи контролю стану речовини в комплексі використовуючи Інтернету речей.

У роботі буде розглянуті питання актуальності вимірювання відносної вологості газів і який функціонал може бути використаний разом з даною технологією.

КЛЮЧОВІ СЛОВА: ІНТЕРНЕТ РЕЧЕЙ IOT, ВЕБ-РОЗРОБКА, ARDUINO, RASPBERRY PI, ВИМІРЮВАННЯ ТЕМПЕРАТУРИ ТА ВОЛОГОСТІ.

РЕФЕРАТ

Данил Половинко «Устройство сбора информации с использованием технологии «Интернет вещей»»

Квалификационная работа бакалавра по прикладной физике. – Х.: Харьковский национальный университет имени В.Н.Каразина, 2020.

Настоящая выпускная квалификационная работа содержит результаты сравнительного анализа существующих систем дистанционного контроля устройств Интернета вещей и разработку системы контроля состояния вещества в комплексе используя Интернета вещей.

В работе будет рассмотрены вопросы актуальности измерения относительной влажности газов и функционал который может быть использован вместе с данной технологией.

КЛЮЧЕВЫЕ СЛОВА: ИНТЕРНЕТ ВЕЩЕЙ IOT, ВЕБ-РАЗРАБОТКА, ARDUINO, RASPBERRY PI, ИЗМЕРЕНИЕ ТЕМПЕРАТУРЫ И ВЛАЖНОСТИ.

ABSTRACT

Danil Polovynko «Information gathering device using the Internet of Things technology»

Qualification work of the bachelor of applied physics. – Kharkiv. – V.N. Karazin Kharkiv national university, 2020.

This final qualification work contains the results of a comparative analysis of existing systems for remote control of Internet of Things devices and the development of a system for monitoring the state of matter in the complex using the Internet of Things.

The paper will consider the relevance of measuring the relative humidity of gases and what functionality can be used alongside this technology.

KEYWORDS: INTERNET OF THINGS IOT, WEB DEVELOPMENT, ARDUINO, RASPBERRY PI, TEMPERATURE AND HUMIDITY MEASUREMENT.

ЗМІСТ

ВСТУП 8
1. ПОСТАНОВКА ЗАВДАННЯ ТА АНАЛІЗ ЛІТЕРАТУРНИХ ДЖЕРЕЛ 10
2 ТЕОРИТИЧНА ЧАСТИНА 12
2.1 Огляд існуючих платформ інтернету речей 12
2.1.1 Платформа дистанційного контролю пристроїв IoT ThingSpeak 13
2.1.2 Система дистанційного контролю пристроїв IoT Xively 15
2.2 Порівняльний аналіз систем геолокації 18
2.3 Розробка вимог до реалізації системи моніторингу та управління 21
2.4 Модуль обробки даних та його оцінка 23
3. ПРАКТИЧНА ЧАСТИНА 26
3.1 Розробка архітектури системи 26
3.1.1 Розробка дизайну і інтерфейсу 31
3.2 Вибір інструментів реалізації 32
3.3 Реалізація системи геолокації 34
3.4 Реалізація системи контролю параметрів 35
3.5 Тестування системи 37
ВИСНОВОК 39
ПЕРЕЛІК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ 41

ВСТУП

Актуальність теми. В даний час великого поширення набули пристрої Інтернету речей - пристрої мають доступ до мережі Інтернет, здатні взаємодіяти один з одним або з зовнішнім середовищем. Згідно з прогнозами компанії IDC, світовий парк встановлених кінцевих точок Інтернету речей (Internet of Things, IoT), який в кінці 2016 році становив 14,9 млрд, в 2025 р перевищить 82 млрд. При таких темпах Інтернет речей незабаром може стати настільки ж незамінним, як і сама мережа Інтернет.[1]

Об’єкт дослідження – популярні системи дистанційного контролю пристроїв Інтернету речей.

Предмет дослідження – механізми розгорнення функціональності пристроїв Інтернету речей.

Методи дослідження.

В процесі даної роботи будуть досліджені існуючі системи дистанційного контролю пристроїв Інтернету речей, будуть виявлені їх переваги та недоліки, будуть вироблені пропозиції (вимоги) щодо підвищення безпеки пристроїв інтернету речей, на підставі яких буде розроблена власна система збору та контролю пристроїв у фізичних системах IoT (Internet of Things - Інтернет речей).

Повсюдно розгорнуті мережі датчиків відкривають новий світ корисних додатків - від інтелектуальних систем освітлення, опалення та кондиціонування, автономних вуличних ліхтарів, здатних спілкуватися один з другом, регулювати свою яскравість у міру необхідності і проводити моніторинг транспортних потоків, до вбудованих в одяг мікросенсорів, що стежать за пульсом і іншими важливими параметрами життєдіяльності людського організму.

Разом з тим всі ці нововведення створюють цілий ряд проблем з інформаційної

безпекою. З'являється можливість попадання мережі датчиків в руки

зловмисника. Таке особливо ймовірно, якщо мережа безпосередньо управляє фізичними пристроями за допомогою комунікацій "машина-машина" (machine-to-machine, M2M).

Практичне значення отриманих результатів: За результатами дослідження будуть розроблені вимоги до реалізації системи контролю стану і геолокації пристроїв Інтернету речей

Структура і обсяг роботи. Бакалаврська робота складається зі вступу, трьох розділів, висновків, списку використаних джерел. Загальний обсяг роботи становить 35 сторінок, із них основного тексту 29 сторінок, 10 рисунків за текстом, список використаних джерел із 8 найменувань на 1 сторінці.

1. ПОСТАНОВКА ЗАВДАННЯ ТА АНАЛІЗ ЛІТЕРАТУРНИХ ДЖЕРЕЛ

Головною метою роботи є підвищення безпеки пристроїв Інтернету Речей. Для її досягнення буде проведено дослідження існуючих технологій та сервісів інтернету речей (далі - IoT - Internet of Things), результати якого, будуть покладені в основу розробки власної системи для роботи з пристроями IoT, що забезпечує їх інформативність за допомогою контролю стосовно відносної вологості повітря та аналізу даних, що передаються.

Планована до розробки система повинна бути конкурентоспроможна щодо існуючих комерційних систем контролю пристроїв IoT. Більш того, вона повинна мати певні переваги в питанні забезпечення безпеки пристроїв, оскільки головною метою роботи є саме підвищення безпеки. Виходячи з цього, перш за все, необхідно досліджувати існуючі системи дистанційного контролю пристроїв IoT, проаналізувати реалізовані ними механізми забезпечення безпеки пристроїв, виявити недоліки і сформулювати власні пропозиції щодо можливості підвищення безпеки, які в подальшому будуть реалізовані в ході розробки власної системи.

Таким чином розглянувши пристрій мереж інтернету речей, можна точно сказати, що в плані апаратної частини немає нічого загадкового і складного. Зробити простеньку IoT-мережу може будь-хто, здатний купити досить дешеві на сьогоднішній день компоненти і написати код з пари рядків. Однак для того, щоб розробити і причинити в життя серйозні проекти як, наприклад, реалізацію концепції розумного будинку або навіть розумного міста, потрібно докласти величезну кількість зусиль. Адже для того, щоб всі ці пристрої працювали між собою потрібна платформа, здатна контролювати всі протікають процеси.

Так само не варто забувати, що в хмарах інтернету речей можуть використовуватися і інші технології, які допомагають розкрити його потенціал в більшій мірі. Такими можуть виступати і BigData, і BlockChain, і

нейромережі з машинним навчанням. Але ж кожна з останніх перерахованих технологій являє собою окрему велику область комп'ютерних наук.

Отримана в результаті роботи система, дозволить підвищити безпеку пристроїв IoT за допомогою реалізації технології геолокації, а також контролю параметрів пристроїв. Розроблена система повинна бути здатна конкурувати з популярними комерційними системами дистанційного контролю пристроїв IoT.

2 ТЕОРИТИЧНА ЧАСТИНА

2.1 Огляд існуючих платформ інтернету речей

Планована платформа повинна бути здатна конкурувати з популярними платформами, тому в якості об'єктів дослідження необхідно вибрати популярні або ж активно розвивинені платформи дистанційного контролю пристроїв інтернету речей. У підсумку, для розгляду були обрані системи входять в статті про найбільш популярних платформах для інтернету речей: Top 49 Tools For The Internet of Things, Internet of Things Platforms. [2] [3]

- *ThingSpeak* - хмарний сервіс для роботи з пристроями інтернету речей, володіє відкритим інтерфейсом, дозволяє візуалізувати дані. [4]
- *Xively* - хмарний сервіс для роботи з пристроями інтернету речей. Також дозволяє візуалізувати дані, дозволяє відстежувати місце розташування пристроїв, володіє відкритим інтерфейсом для сторонніх користувачів. [5]

При дослідженні будемо розглядати наведені системи з боку функціональних можливостей, що надаються системою. подібна система повинна володіти:

- Швидким та легким інтерфейсом
- Засобами для взаємодії з пристроями - для отримання даних від пристроїв, що використовуються в подальшому для обробки.
- Можливостями для обробки одержуваних даних – механізми обробки, аналізу та візуалізації даних.
- Засобами для забезпечення безпеки пристроїв.

Забезпечення безпеки пристроїв, так чи інакше, зводиться до реалізації взаємодії пристроїв з серверами системи. Пристрої з деякою періодичністю відправляють інформацію зібрану з пристрію на сервер, які сервер аналізує і повертає людині отримані дані в чіткому та зрозумілому вигляді.

Розділимо засоби реалізації інформаційності пристроїв на два типу:

- Здатність обробляти параметри, що передаються пристроями.
- Здатність обробляти геолокацію, передану пристроями.

Основні функції котрі реалізуються у подібних технологіях:

- *Збір* - механізм, що реалізує отримання даних від пристроїв, зберігаючи у базі
- *Аналіз* – аналізуюча вхідні дані функція, котра є сигналом для забезпечння безпеки, або утримання під контролем якогось об’єкту. Наприклад : перегрів електронного щитку, на що пристрій відреагує та повідомить свого користувача о небезпеці.
- *Візуалізація* – функція котра полегшує користування технологіями інтернету речей, та може бути здатна для користування людиною навіть далекою від сфери інформаційних технологій.
- *Геолокація* – фунцкція контролю переміщення пристрію або роспознавання, де саме зараз є ваш пристрій. Дуже зручно, коли пристроїв декілька відстежувати їх на мапі.
- *Агрегація* призначених для користувача пристроїв / винесення їх на глобальну карту - підвищення зручності спостереження користувачем за всіма своїми пристроями. Елемент користувацького фукнціонала, що впливає на зручність контролю за безпекою пристроїв.

Будемо порівнювати системи за наявністю в них і методам реалізації даних функціональних можливостей.

2.1.1 Платформа дистанційного контролю пристроїв IoT ThingSpeak

ThingSpeak - хмарний сервіс для роботи з пристроями інтернету речей. Розташовується за адресою: https://thingspeak.com/.

Заявлений функціонал сервісу:

- *Збір даних*. Можливість з’єднання пристрою та сервісу. Сервер зберігає усі дані у БД.
- *Аналіз і візуалізація даних*. Отримана інформація відібражається у додатку. Людина може різноманітними засобами впливати або просто відстежувати інформацію навколо пристрію.
- *Спонука до дії.* Людина може виставляти певні рамки спрацьовування та виконнання певних дій у пристрої після повернених з нього даних.

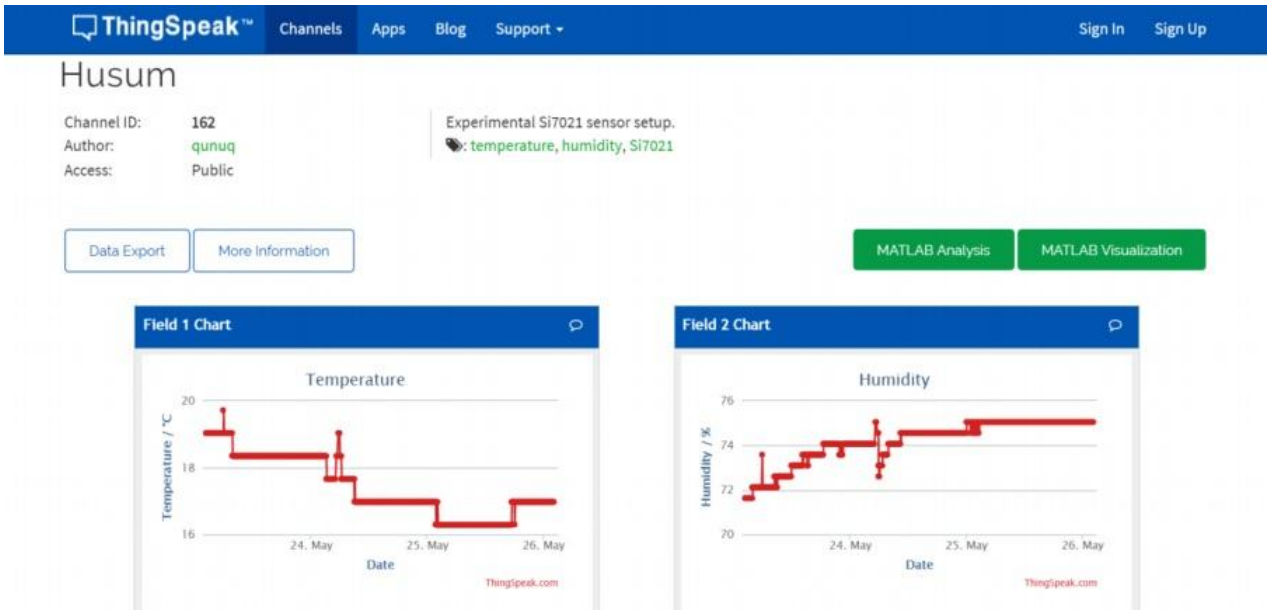


Рисунок 1.Інтерфейс сервісу Thing speak.

На рисунку 1 є приклад інтерфейсу даної платформи. наведено загальну інформація про пристрій: id пристрої, власник, тип доступу, а також дані, отримані від пристрою, візуалізовані за допомогою графіків.

ThingSpeak сумісний з пристроями на платформах Arduino і Raspberry Pi.

Проведемо оцінку системи за наведеними в попередньому розділі критеріям.

Дані будуть оцінюватися від “1” до “5”.

Таблиця 1

Оцінка системи ThingSpeak

Фунція	Методи	Оцінка
Збір даних	Пристрої підключаються безпосередньо за допомогою відкритого API ThingSpeak. Всі дані зберігаються в базі, є можливість експорту даних пристроїв в форматі json, в тому числі і сторонніми користувачами, якщо пристрій позначено як загальнодоступне. Існує відкрите API для зручного доступу до пристроїв з допомогою програмних засобів.	5
Аналіз даних	Все, отримані від пристроїв, дані аналізуються. Існує можливість отримання з масиву параметрів нових параметрів (обробка даних). Доступний аналіз за допомогою MATLAB. Існує можливість установки "реагування" на певні події.	5
Візуалізація даних	Дані візуалізуються за допомогою графіків, діаграм, "Цифрових вимірювальних приладів", MATLAB.	4
Геолокація пристроїв	Система зберігає дані про місцезнаходження пристроїв,проте не візуалізує їх за допомогою локації на картах.	3
Агрегація пристроїв	Повний список призначених для користувача пристроїв доступний тільки їх власнику. Відсутня локація пристроїв на карті.	2

2.1.2 Система дистанційного контролю пристроїв IoT Xively

Xively - ще один хмарний сервіс для роботи з пристроями інтернету речей. Розташовується за адресою: https://xively.com/.

За своїм функціоналом Xively схожий на Thingspeak, основним незручністю є відсутність відкритого API для отримання даних про пристрої стороннім користувачем і закрита реєстрація. Для того, щоб зареєструватися необхідно подати заявку.

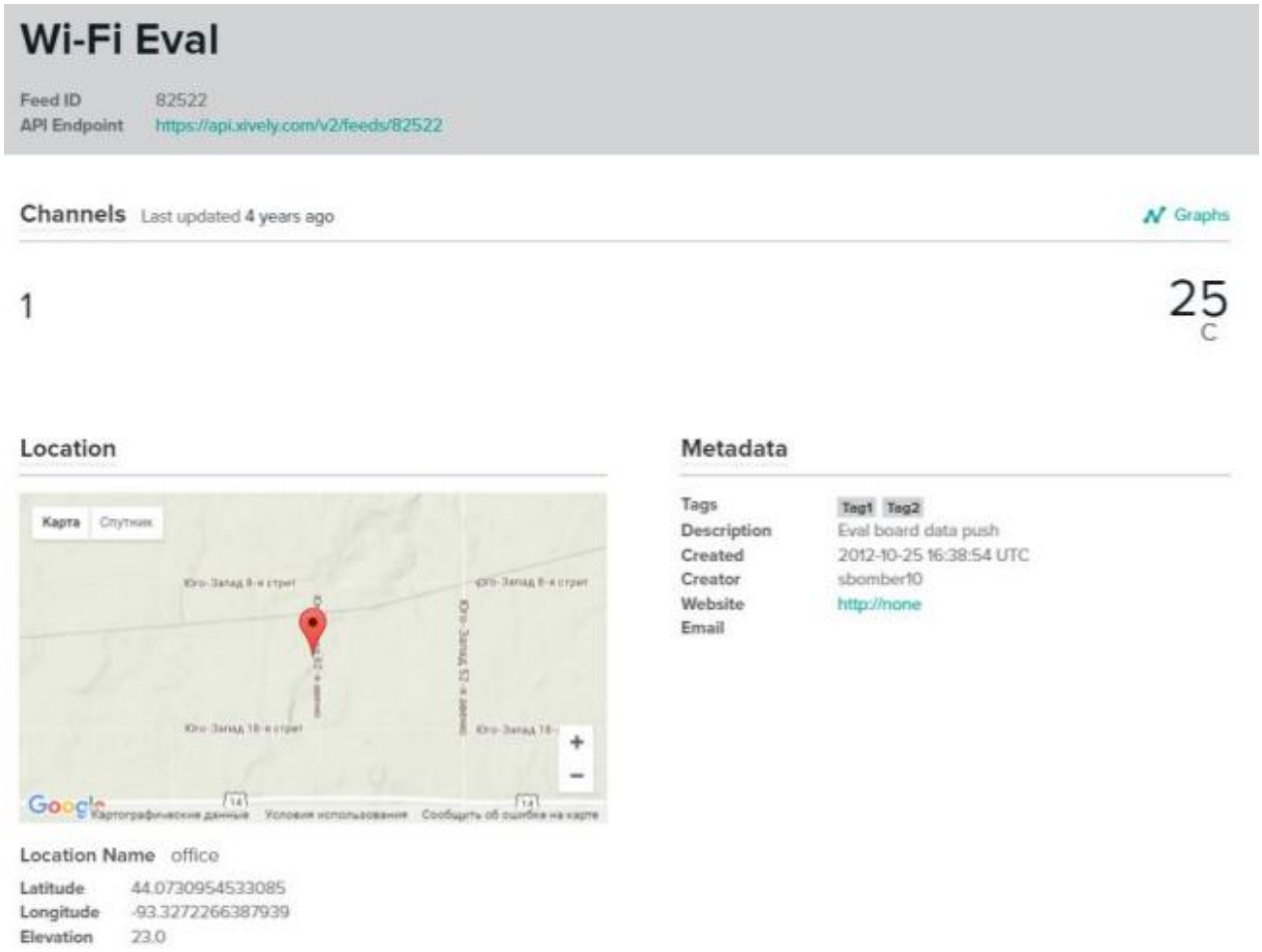


Рисунок 2. Інтерфейс системи Xively

На рисунку 2 зображений інтерфейс пристрою, який містить загальну інформацію про пристрій: теги, опис, дата додавання, власник, сайт і пошта для зв'язку. Відображаються поточні дані, є можливість їх візуалізації з допомогою графіків. Відображається поточне розташування пристрою.

Заявлений функціонал сервісу:

- Збір і зберігання даних. Кожен потік даних являє собою каналпевного типу. До кожного з пристроїв повинен бути прив'язаний хочаб один канал.
- Геолокація. Зберігання даних про географічне положення пристрою.

Висновок на екран поточного місцезнаходження пристрою за допомогою бібліотек javascript для роботи з картами.

- Тригери. Можливість установки певних умов, при активації яких відбуваються певні дії, наприклад повідомлення вбудь-якої з соціальних мереж (Twitter, Facebook). Використовуються кошти Zapier.
- API для розробників.
- Візуалізація даних за допомогою Javascript. Xively також працює з пристроями на базі Arduino, Raspberry Pi, Electric Imp.

Таблиця 2

Оцінка системі Xively

Функція	Методи	Оцінка
Збір даних	Дані про пристрої виходять з різних сервісів- таким чином пристрої з різних мереж агрегируются на одній глобальній карті світу. У базі зберігається тільки поточний стан пристрою.	5
Аналіз даних	Отримані системою дані не аналізуються.	1
Візуалізація даних	Дані не візуалізуються. поточний стан пристрою виводиться в текстовому форматі: "поле: значення".	1
Геолокація пристроїв	Кожен пристрій відзначається на глобальній карті в відповідність зі своїм поточним місцезнаходженням і Категорія пристроїв, до яких вона належить	4

	(Транспорт, будинок і т.д.).	
Агрегація пристроїв	Система агрегує пристрою не тільки одного конкретного користувача в одному місці, але і пристрої інших користувачів по всьому світу.	5

2.2 Порівняльний аналіз систем геолокації

Наведемо порівняння описаних систем по реалізованому функціоналу. за
повну реалізацію функції будемо ставити "+", за часткову - "+/-", в разі відсутності подібного функціоналу - "-":

Таблиця 3

Порівняльний аналіз систем

	Збір даних	Аналіз даних	Візуалізація даних	Геолокація пристроїв	Агрегація пристроїв
ThingSpeak	+	+	+	+/-	+/-
Xively	+	+	+	+	+/-
Запланована система	+	+	+	+	+

У підсумку, можна відзначити, що система Xively найбільш коректна до вимог, реалізуючи в тій чи іншій мірі кожне з них, за винятком відстеження переміщень.

Однак ThingSpeak надає більш серйозні можливості для аналізу даних, такі як MATLAB, більш широкий спектр можливостей для візуалізації даних, відкритий програмний API. Основним недоліком сервісу

ThingSpeak є відсутність обробки географічних даних переданих пристроями. Сервіс Thingful реалізує в собі ідею про агрегації різних пристроїв в одному місці, розміщенні їх на одній глобальній карті, що дозволяє користувачеві досить легко контролювати місце розташування своїх пристроїв і пристроїв інших користувачів, його зацікавили.

Реалізована в ході роботи дипломної роботи система буде містити в собі кожен з наведених вище компонентів. Основним її перевагою перед системами Xively і ThingSpeak буде більш досконала робота з географічними даними, а саме - геолокація пристроїв, відстеження їх переміщень, перевірка знаходження пристрою в допустимої області, локація всіх пристроїв користувача на одній карті для більш зручного контролю за ними - Xively і ThingSpeak лише зберігають поточне місце розташування пристроїв, але ніяк не обробляють передані географічні дані, не дозволяють їх візуалізувати і не дозволяють "зібрати" всі призначені для користувача пристрої на одній карті. перевагою реалізованої системи перед Thingful буде наявність аналізу одержуваних даних, можливості візуалізації даних, можливості відстеження переміщень пристроїв.

Наведемо більш детальну оцінку планованої системи за пред'явленими раніше критеріям:

Таблиця 4

Оцінка запланованої до реалізації системи

Критерій	Реалізація	Відзнака від конкуренту
Збір даних	Пристрої взаємодіють з системою безпосередньо, всі отримані дані зберігаються в базі і зберігаються	ThingSpeak пристрої також взаємодіють

	необмежений час. Існує можливість додавання додаткових модулів для обробки даних різних типів.	безпосередньо, проте немає відкритого API.
Аналіз даних	Всі отримані дані піддаються аналізу. Користувач має можливість задати граничні значення для кожного з параметрів, що передаються пристроєм, а також допустиму область місцезнаходження пристрою (за допомогою географічних координат центру допустимої зони і її радіусу)	ThingSpeak - аналіз даних в реалізованої системі менш функціональний, так, наприклад, відсутня можливість написання власних правил і роботи з MATLAB
Візуалізація даних	Всі числові дані візуалізуються з допомогою графіків.	Thingspeak - дані також візуалізуються з допомогою графіків.
Геолокація пристроїв	Система зберігає дані про поточний місцезнаходження пристроїв, візуалізує їх за допомогою API Google Maps.	Конкурентів немає

Агрегація пристроїв	Повний список призначених для користувача пристроїв доступний власнику у вигляді списку і у вигляді маркерів на географічній карті, відповідних поточного розташування кожного з пристроїв. Також існує глобальна карта на якій відображаються всі відкриті пристрої.	Thingful - система також дозволяє агрегувати призначені для користувача пристрою на глобальній карті, надаючи, однак, додатковий інтерфейс для відображення тільки своїх пристроїв на окремій карті.

2.3 Розробка вимог до реалізації системи моніторингу та управління

Розглянувши популярні дистанційного контролю пристроїв інтернету речей, такі як Xively та ThingSpeak, ми можемо пред'явити ряд вимог до розробляється в рамках бакалаврської роботи власній системі дистанційного контролю і геолокації пристроїв IoT. Функціональні вимоги, що пред'являються до системи:

- Здатність додавання користувачем власних пристроїв IoT.
- Наявність механізму взаємодії пристроїв з системою, тобто наявність можливості прямої передачі даних від пристрою із сервером, на якому розгорнута система (підключення пристроїв).
- Зберігання переданих даних в базі даних.

- Аналіз отриманих даних: на входження в допустимі межі, на знаходження в допустимої області, - генерація проблем. Можливість розширення списку реалізованих методів аналізу з боку сервера.
- Геолокація пристроїв, відстеження переміщення, висновок географічних даних на карту: для кожного пристрою на локальну карту виводиться вся історія його переміщень і допустима область знаходження, всі призначені для користувача пристрої виводяться на призначену для користувача локальну карту, всі відкриті пристрої користувачів виводяться на глобальну карту світу.
- Реалізація системи у вигляді веб-додавання, тобто доступ користувачів до системи здійснюється за допомогою веб-браузера.

Призначені для користувача вимоги, що пред'являються до системи:

- Оригінальність інтерфейсу
- Можливість зручного доступу до своїх пристроїв, виведення списку своїх пристроїв, відображення всіх призначених для користувача пристроїв на мапі.
- Візуалізація даних, отриманих від пристроїв.

Вимоги до безпеки системи:

- Система реєстрації і авторизації користувачів.
- Можливість задавати граничні значення для кожного з параметрів, переданих пристроєм, наприклад максимальна і мінімальна температура для термометра. Аналіз на входження значень в допустимі межі.
- Можливість завдання границь допустимої області місцезнаходження пристроїв. Перевірка на знаходження пристрою в допустимої області.

- Виведення списку виявлених проблем із зазначенням пристрою, дати і часу, опису проблеми.

2.4 Модуль обробки даних та його оцінка

Для реалізації технології я обрав датчик DHT11 з модулем ESP8266.

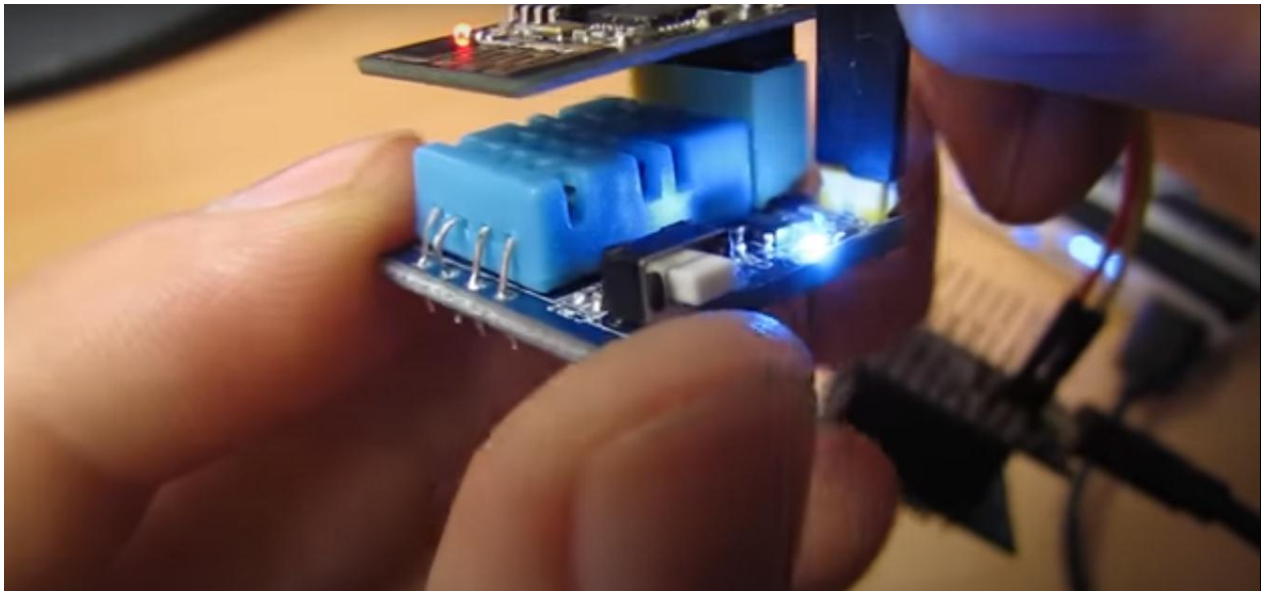


Рисунок 3. Датчик DHT11 з модулем ESP8266

Датчики DHT11 - дуже популярні в середовищі Ардуіно і часто використовуються в проектах метеостанцій і розумного будинку.

Датчик складається з двох частин - ємнісного датчика температури і гігрометра. Перший використовується для вимірювання температури, другий - для вологості повітря. Що знаходиться всередині чіп може виконувати аналого-цифрові перетворення і видавати цифровий сигнал, який зчитується за допомогою мікроконтролера.

Що ми маємо по датчику DHT11 на ділі:

- Струм - 2,5 мА (максимальне значення при перетворенні даних);
- Вимірює вологість в діапазоні від 20% до 80%. Похибка може складати до 5%;

- Застосовується при вимірюванні температури в інтервалі від 0 до 50 градусів (точність - 2%)
- Габаритні розміри: 15,5 мм довжина; 12 мм широта; 5,5 мм висота;
- Живлення - від 3 до 5 Вольт;
- Одне вимір в одиницю часу (секунду). Тобто, частота становить 1 Гц;
- 4 коннектора. Між сусідніми відстань в 0,1 ".

Що ж ми отримали, обравши саме такий датчик з модулем:

Таблиця 5

Оцінка датчику DHT11 з модулем ESP8266

Плюси	Мінуси
Низька ціна за одну одиницю, що надає можливость придбати декілька таких пристроів для розповсюдження.	Відсутність документації у посилці. Було дуже важко знайти схему знайти в інтернеті. Без неї не розібратися.
Дуже швидкий відклик та якщо є документація то розбереться навіть людина далека від сфери інформаційних технологій.	Є погрішність с температурою у 2-3 градуси по цельсію, але цю проблему можна вирішити.

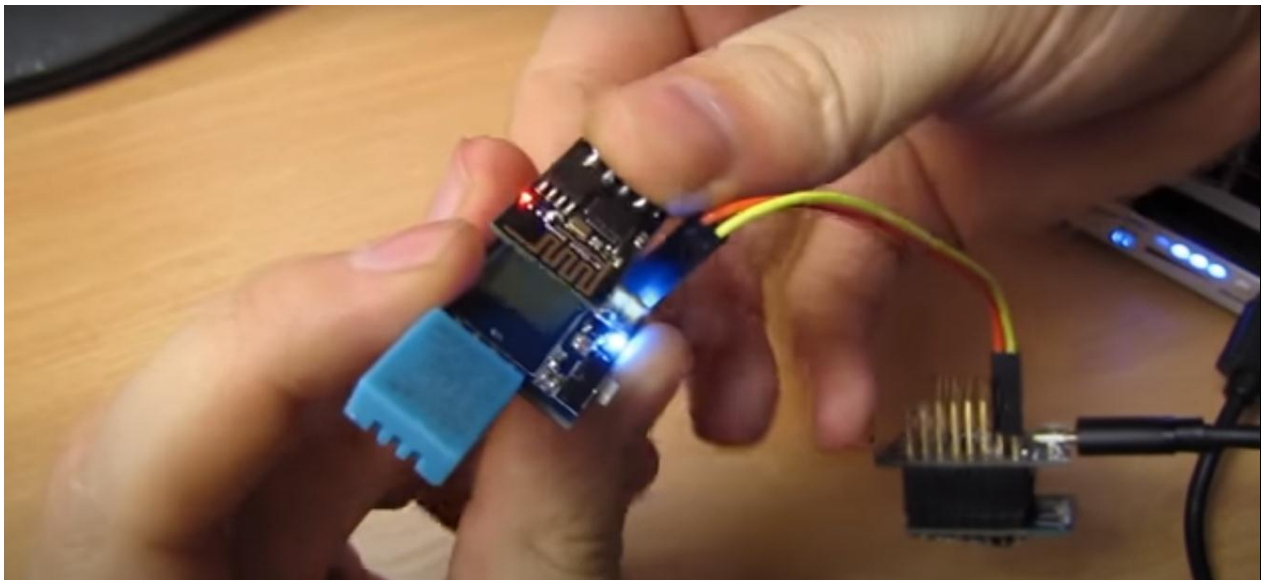


Рисунок 4. Вирішення проблеми з погрішністю температури.

Вирішення полягає у тому, щоб відігнути сам модуль від плати. Тоді модуль припинює нагрівати її і вона починає працювати коректно.

Висновки по обраному датчику: даний датчик є непоганим бюджетним варіантом для контролю вологості та температури повітря та працює коректно як в домі так й на вулиці. Переважно те, що він потребує дуже мало енергії, але в наборі не має корпусу, що робить його до таких речей як дощ або бруд.

3. ПРАКТИЧНА ЧАСТИНА

3.1 Розробка архітектури системи

Реалізований додаток, згідно заявленим вимогам, є веб-додатком. Відзначимо переваги даного підходу:

- Зручність роботи користувача - досить просто відкрити сторінку в своєму браузері.
- Низькі вимоги до обчислювальних можливостей клієнта, оскільки велика частина логіки додатка сконцентрована саме на сервері.
- Немає необхідності завантажувати клієнт системи, займаючи тим самим пам'ять пристрою користувача.

Як основний недолік можна виділити необхідність наявності доступу в Інтернет, інакше - відсутність автономності системи. Тобто додаток повинен мати клієнт-серверну архітектуру, в якої сервером виступає веб-сервер, клієнтом виступає веб-браузер користувача. Також сервер повинен обробляти інформацію, що отримується від пристроїв IoT.

Наведемо список учасників взаємодії:

- Пристрої IoT - передають дані на сервер.
- Сервер - виконує дві ролі: приймає і обробляє дані, передаються пристроями IoT, а також віддає оброблені дані в вигляді веб-сторінки за запитом клієнта.
- Клієнт - відправляє запити на отримання інформації до веб-сервера, може представлятися у вигляді браузера користувача.

Наведемо спрощену схему взаємодії всіх учасників на рисунку 4.

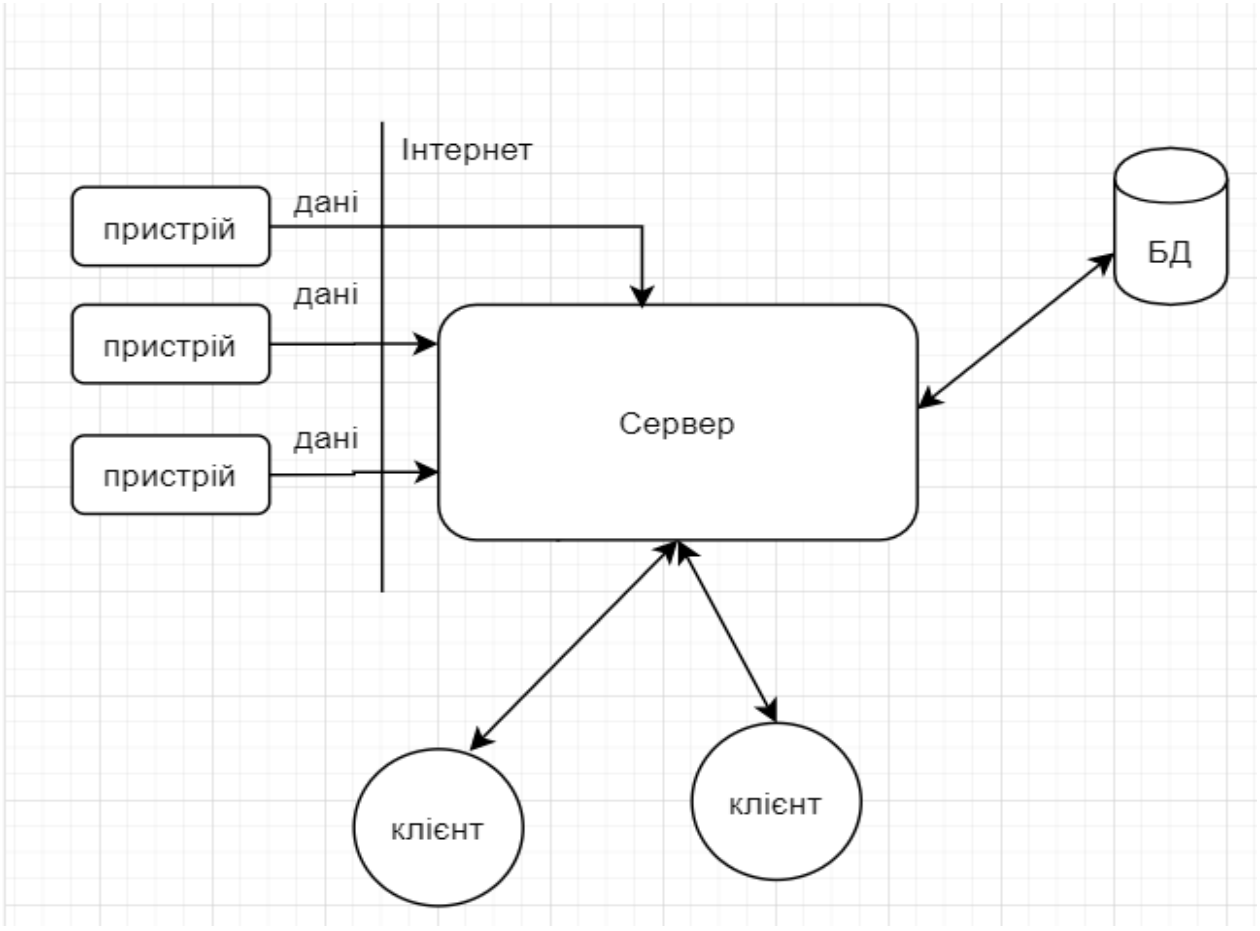


Рисунок 5. Схема взаємодії учасників системи.

Таким чином, для реалізації системи необхідна наявність нижченаведених
модулів:

- Модуль прийому даних на сервері - дозволяє приймати дані від пристроїв, підтримувати передачу даних в різних форматах від різних типів пристроїв.
- Модуль обробки отриманих від пристроїв даних - включає в себе: обробку географічних даних, обробку параметрів пристроїв, аналіз на входження в допустимі межі і генерація проблеми, про якої оповіщається кінцевий користувач.
- Модуль взаємодії з базою даних - необхідний для запису і вилучення необхідних даних.

- Модуль перетворення даних в зручний для користувача вигляд - включає в себе механізми візуалізації даних за допомогою графіків, а також роботу з картами.
- Модуль реєстрації і авторизації користувачів - крім авторизації і реєстрації користувачів включає в себе базовий інтерфейс – 27 можливість зміни персональних даних - таких як пароль і поштову адреса, можливість додавання нових пристроїв в свій список і т.д.
- Модуль безпеки системи - для захисту системи від різних атак (SQL-injection, XSS)

Наведемо більш детальну схему роботи реалізованої системи. На малюнку 5 відображена взаємодія пристроїв з сервером, на малюнку 3 – взаємодія клієнта і сервера.

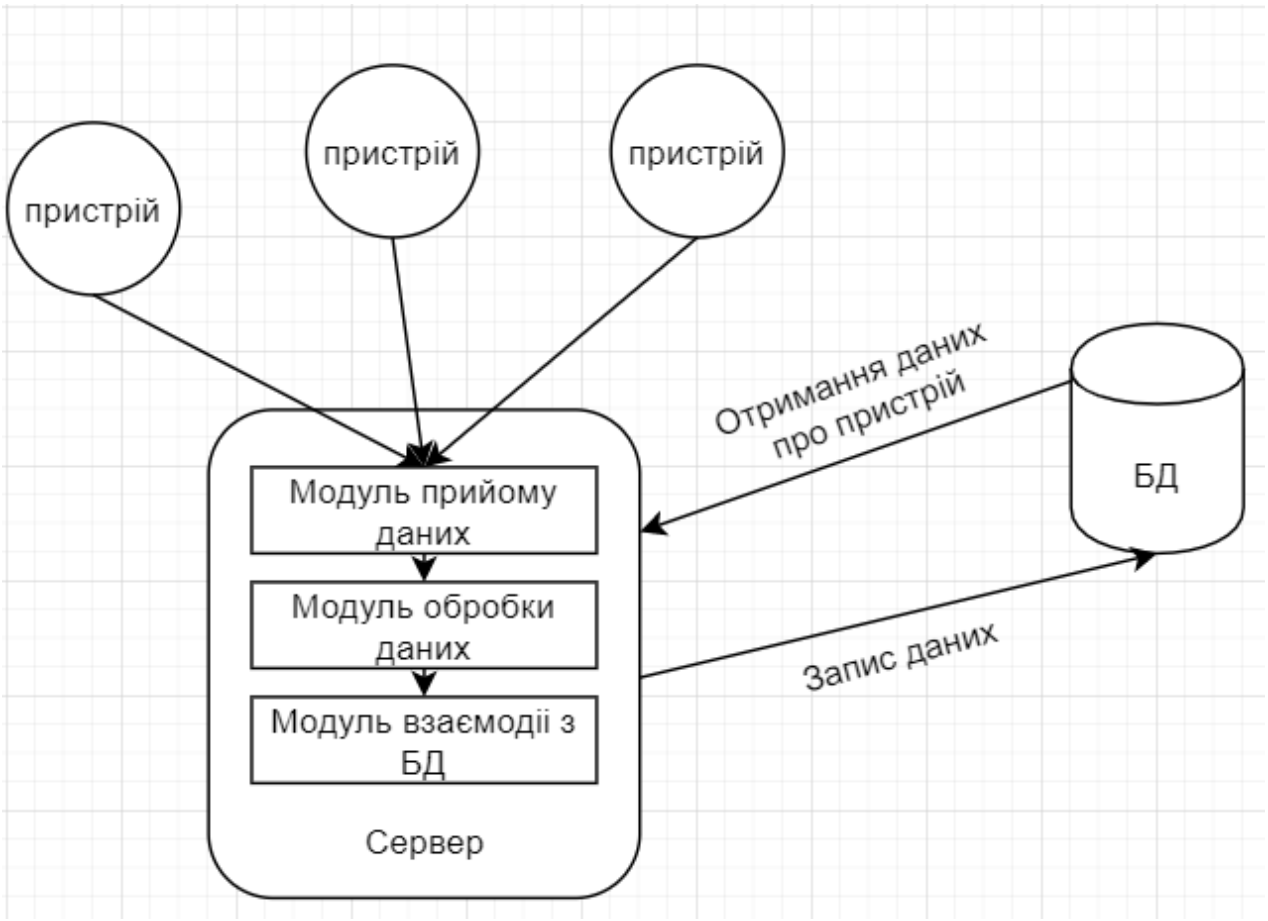


Рисунок 6. Взаємодія пристріїв з сервером

Дані від пристроїв надходять на модуль прийому даних, де дані різних типів приводяться до єдиного вигляду, тобто нормалізуються. Нормалізовані дані надходять в модуль обробки даних, де відбувається їх аналіз та перевірка на коректність. Потім всі дані і результати обробки записуються в базу даних. При надходженні на сервер запиту користувача, користувач аутентифікується, запит перевіряється на коректність, необхідні дані витягуються з БД і повертаються користувачу в зручному для сприйняття вигляді.

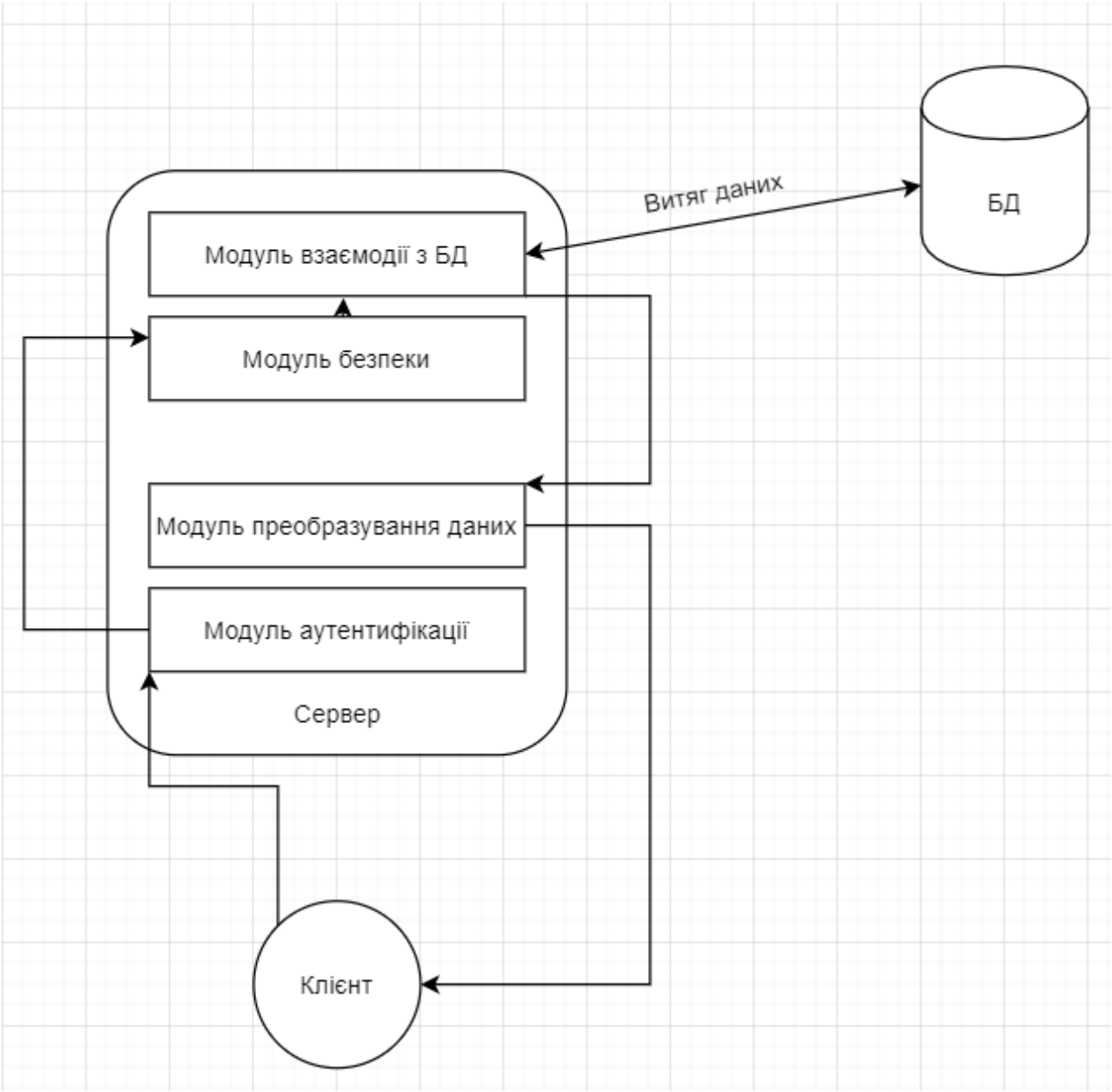


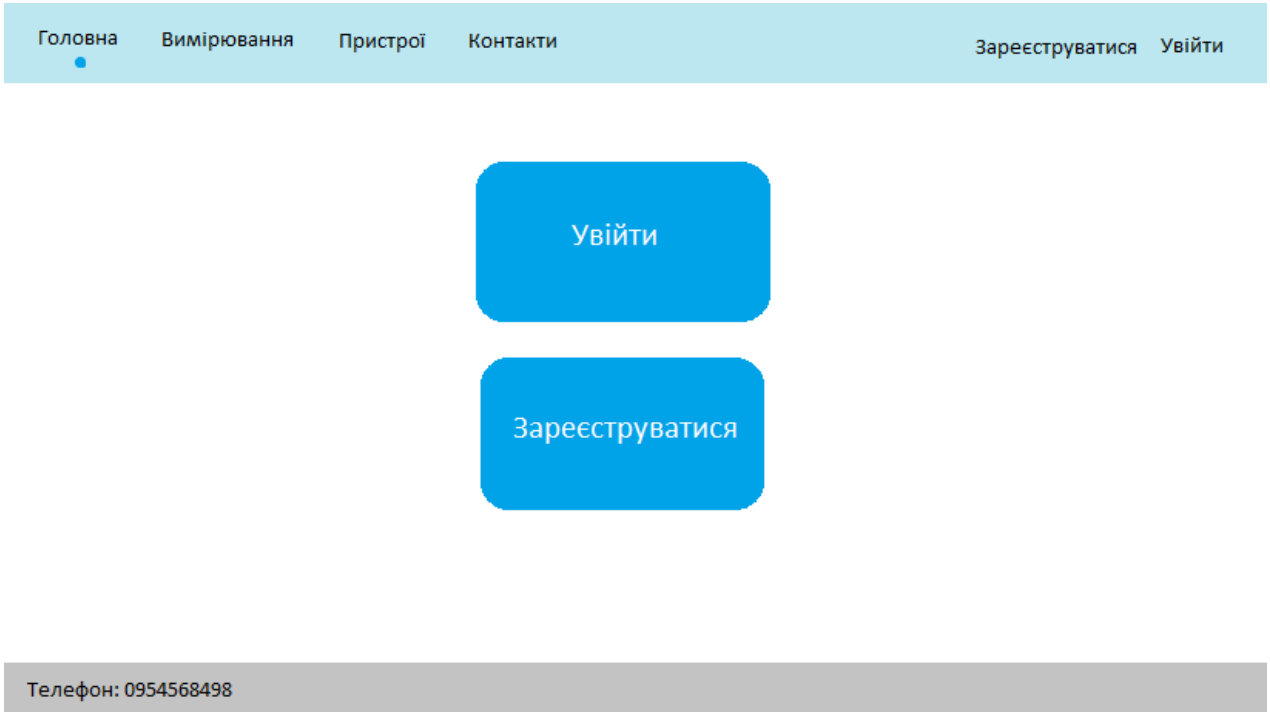
Рисунок 7. Взаємодія клієнту з сервером

Запропоновані в складі архітектури модулі дозволять в результаті:

- Забезпечити безпеку пристроїв IoT завдяки контролю за їх географічним положенням і коректністю даних, що передаються з допомогою модуля обробки даних.
- Забезпечити можливість взаємодії з різними типами пристроїв з різними форматами даних, що передаються за допомогою модуля прийому даних.

- Забезпечити безпеку користувача за допомогою модулів аутентифікації і безпеки.
- Забезпечити подання даних користувачеві в зручному вигляді з допомогою модуля перетворення даних

3.1.1 Розробка дизайну і інтерфейсу



Риснок 8. Головна сторінка сайту (макет)

На малюнку 7 зображена головна сторінка сайту, в тому випадку, якщо користувач не авторизований. У верхній частині розташовується фіксоване меню - при прокручуванні сторінки воно залишатиметься вгорі. З меню користувачеві доступні наступні сторінки:

- Головна - стартова сторінка, що надає лише додатковий інтерфейс.
- Глобальна карта - виведення всіх відкритих пристроїв на глобальну карту
- Вимірювання - виведення списку можливих вимірювань
- Пристрої - виведення всіх пристроїв користувача.

Також, в залежності від того, авторизований користувач чи ні – такі сторінки:

- Зареєструватися - сторінка з формою для реєстрації (НЕ авторизований)
- Увійти - сторінка з формою для логіна (не авторизований)

3.2 Вибір інструментів реалізації

Для виконання поставленого завдання нам буде необхідно реалізувати:

- Серверну частину, яка містить основну частину логіки програми.
- Клієнтську частину, що дозволяє користувачеві в зручному вигляді переглядати інформацію

Для реалізації серверної частини необхідно вибрати серверний мова програмування і використовувану базу даних.

Можливі варіанти вибору серверного мови програмування: PHP, ASP.NET, Java, Python (Django). В ході навчання були отримані навички створення веб-додатків за допомогою мови PHP, який на даний момент за статистикою є найбільш популярним засобом веб-розробки. [7]

PoweredBy	Все	.ru	.su	.рф	.by	.ua	другое
PHP/5.2.17	381527	315307	5596	32377	9451	13097	5699
PHP/5.3.28	264135	224961	3924	23420	5660	3556	2614
PHP/5.3.3-7+squeeze17	169218	164226	1030	1770	231	1147	814
PHP/5.3.3-7+squeeze19	165025	157370	1220	2774	424	2302	935

Рисунок 9. Найширші мови веб-програмування

Основними перевагами мови PHP є:

- Можливість вільного використання.
- Широкий функціонал, пристосований під вимоги веброзробки.
- Підтримка безлічі баз даних.
- Зручність у використанні.

Як недоліки можна виділити відсутність многопоточности, а отже, досить низьку продуктивність.

У підсумку, в силу зручності використання і досвіду роботи в якості серверного

мови програмування був обраний саме PHP.

Одними з найбільш популярних вільно розповсюджуваних СУБД на даний момент є MySQL - і PostgreSQL. [9]

Наведемо переваги і недоліки кожної з них. [10] Переваги MySQL:

- Простота в роботі. Зокрема, простота в установці і наявність графічного інтерфейсу, що забезпечує зручність роботи.
- Підтримується практично весь функціонал SQL.
- Безпека - підтримка функцій, що забезпечують безпеку.
- Масштабованість - легкість роботи з великими обсягами даних і простота масштабованості
- Швидкість - спрощення деяких стандартів SQL дозволяє значно підвищити продуктивність.

Основні недоліки MySQL:

- Обмеження в функціональності
- Можливі проблеми з надійністю

Основні переваги PostgreSQL:

- Дуже потужна функціональність
- Можливість розширення функціоналу за допомогою різних доповнень
- PostgreSQL є також об'єктно-орієнтованої СУБД

Основні недоліки PostgreSQL:

- Досить низька продуктивність.
- Популярність нижче, ніж у MySQL, а отже складніше знайти хостинг, що підтримує PostgreSQL.

З огляду на більшої швидкості роботи і відсутності необхідності зберігання складних структур даних в якості використовуваної СУБД була обрана MySQL.

Як клієнтських мов будуть використовуватися стандартні мови: HTML - в якості мови розмітки, CSS - як інструмент для стилізації додатки, JavaScript - для анімації і обробки даних на стороні клієнта. Також була необхідність

вибору бібліотек JavaScript для роботи з графіками і для роботи з картами. Для роботи з графіками була обрана бібліотека Chart.js (), що реалізує весь необхідний функціонал – можливість побудови графіків, діаграм, можливість їх стилізації, підключення анімацій. Для роботи з картами можна було використовувати відкритий API Google Maps і відкритий API Яндекс Карт. Якщо цільова система не має прив'язки до конкретної території (країні, континенту), то доцільно використовувати Google Maps, тому що надається ними глобальна карта світу набагато більш докладна.

3.3 Реалізація системи геолокації

При реалізації системи геолокації було необхідно забезпечити можливість відстеження поточного місцезнаходження, відстеження траєкторії його премещенія в часі, а також можливості аналізу його географічного положення на предмет знаходження в допустимої області.

Надсилаючи звіт про проблеми від пристрою в базу даних разом з параметрами, переданими пристроєм записуються також його поточні координати. Запис координат дозволить надалі відстежити всі переміщення пристрою. Під час обробки переданих пристроєм даних сервер, зокрема, перевіряє передані географічні координати на знаходження в допустимої області, яку вручну при необхідності задає власник пристрою (задаючи центр допустимої області та її радіус).

У підсумку реалізована система геолокації містить наступні можливості:

- Відображення всіх відкритих (загальнодоступних) пристроїв на глобальній карті світу.
- Відображення всіх пользовательзовательскіх пристроїв на одній карті (Агрегація їх в одному місці).
- Відображення поточного місцезнаходження пристрою.
- Відображення історії переміщення пристрою.
- Відображення допустимої області знаходження пристрою, можливість задати область вручну.

- Аналіз географічних координат, що передаються пристроєм, на знаходження в допустимої області. У разі виходу за межі області - генерація проблеми і оповіщення про неї власника пристрою.

Візуалізація географічних даних проводиться за допомогою API Google Maps для JavaScript. Наведемо в загальному вигляді алгоритм роботи з географічними
даними, переданими від пристроїв:

1. Передача даних від пристрою на сервер, нормалізація даних.
2. Отримання з бази даних допустимої області знаходження пристрою.
3. Визначення входження переданих географічних даних в допустимий радіус. Якщо пристрій "вийшло" за допустимі межі - генерація проблема, про яку оповіщається користувач.
4. При запиті користувачем інформації про пристрій, що включає географічні дані - дані запитуються у БД, потім наводяться до формату JSON, необхідному API Google Maps, який передається сервером в Javascript і обробляється вже в браузері клієнта.

3.4 Реалізація системи контролю параметрів

При реалізації системи контролю параметрів було необхідно реалізувати такі можливості: можливість завдання користувачем максимальних і мінімальних меж для кожного з параметрів, що передаються пристроєм, можливість перевірки даних, що передаються на входження в допустимі межі, можливість візуалізації чисельних значень.

При передачі даних від пристрою на сервер кожен з параметрів піддається контролю, відповідно до заданих користувачем межами, якщо значення перевищує максимально допустимий, або, навпаки, менше мінімально можливого - система генерує проблему, про яку інформується користувач.

Система обробки отриманих даних допускає можливість масштабування,

тобто можуть бути додані нові правила для контролю переданих параметрів.

Наведемо схему роботи з переданими пристроями параметрами:

1. Передача даних від пристрою на сервер, нормалізація даних.
2. Отримання з бази даних допустимих меж для кожного з переданих параметрів.
3. Перевірка знаходження отриманих даних в допустимих межах. Якщо отриманий параметр завищений, або, навпаки, занижений – генерується проблема, про яку оповіщається власник пристрою.
4. При запиті користувачем інформації про пристрій, що включає географічні дані - дані запитуються у БД, потім наводяться до формату JSON, необхідному API Chart.js, який передається сервером в Javascript і обробляється вже в браузері клієнта. У браузері клієнта дані візуалізуються за допомогою графіків.

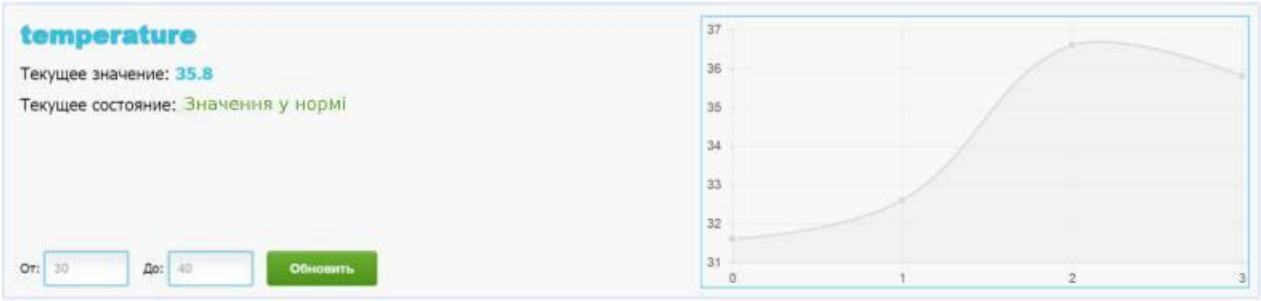


Рисунок 10. Відображення параметрів, що передаються пристроями.

На рисунку 10 зображено призначений для користувача інтерфейс, що відображає, дані, що передаються пристроєм. Кожен подібний блок даних містить:

- Назва параметра (в даному випадку - "temperature")
- Поточне значення параметра (в даному випадку - "35.8") - засноване наостанніх даних, переданих пристроєм.

- Оцінка поточного значення параметра на підставі заданих меж.Візуалізовані за допомогою графіка дані, максимальне кількість точок на графіку - 30.
- В лівому нижньому кутку - форма для поновлення допустимих меж значення параметрів. Повідомлення про порушення кордонів передається на сервер з допомогою технології AJAX.

У підсумку, реалізована система дозволяє користувачеві здійснювати контроль за переданими пристроями значеннями, як за допомогою автоматизованого аналізу на основі заданих правил, так і за допомогою візуальної оцінки графіків і діаграм.

3.5 Тестування системи

По закінченню розробки, було проведено тестування системи на коректність роботи. Для тестування було вибрано пристрій "a", для якого було запропоновано такі граничні умови:

Таблиця 6

Вихідні параметри для тестування системи

Параметр	Мінімальне значення	Максимальне значення
temperature	25	43
humidity	30.4	32.6
geolocation	Центр: 49.5210 п.ш, 36.1619 с.д.	Радіус: 700м

На вхід скрипту, йде обробка сигналів від пристрою, передавалися випадковим чином згенеровані дані. Дані, що виходять за допустимі границі, повинні були викликати помилку.

Наведемо кілька прикладів вхідних і вихідних параметрів. Наведемо: вхідні значення, вихідні значення, необхідне значення на виході, відповідність вимогам (знак "+", якщо система відпрацювала коректно, знак "-", якщо некоректно).

На вхід подаються географічні дані про пристрої, а також значення для кожного з параметрів (в даному випадку - temperature і humidity).

Таблиця 7

Кінцеве тестування приладу

Дані на вході	Повернуті дані	В чому проблема	Коректність
Довжина: 49.9808 Широта: 36.2527 temperatture: 25.3 humidity: 17.3	Error in temperature! Value is too low: [25.3] < [30], Error in humidity! Value is too low: [17.3] < [30.4]	Температура нижче заданної параметрично. Вологість нижча ніж задана.	+
Довжина: 49.9908 Широта: 36.2727 temperatture: 30.0 humidity: 30.4	No exeptions	Даний приклад не має виходу за рамки входних параметрів, тому система не буду видавати помилку.	+
Довжина: 49.8808 Широта: 36.3027 temperatture: 37.2 humidity: 48.2	Error in humidity! Value is too high: [17.3] < [30.4]	У даному випадку вологість вища ніж задана у параметрах.	+

В результаті, все вхідні параметри обробляються коректно.

ВИСНОВОК

В ході проведеної роботи були досліджені існуючи розвинені платформи інтернету речей для дистанційного контролю, а сама Xively та Thingspeak, був проведен аналіз та за допомогою порівняльних таблиць та оцінки по загольнодоступним веб-ресурсам та обран пристрій який буде сканувати загальну среду на обрані параметри та передавати їх на сервер з базою даних.
Робилася перевага в сторону тих платформ, які мають в наявності усі методи до реалізації мети завдання, а саме:

- Збирання даних
- Анализування даних
- Візуалізація даних
- Геолокація, та глобальна візуалізація даних про пристрій на карті
- Агрегація пристроїв

По власним да інсайдним оцінкам платформ вдалося зробити висновок, що система Xively у більшому сенсі раелізує усі перечислені критерії, але метод з передаванням географічних даних в неї реалізований не самим кращим способом.

А ось платформа Thingful надає нам переважливо увесь перетин заявлених за планом методів для роботи з обраним пристроєм та розкриває увесь спектр його можливостей.

За пристрій збору даних було обрано датчик DHT11 з модулем ESP8266. Дана кострукція по заявленим критеріям мала обробляти температуру да вологість повітря з чим успішно й впоралася.
На основі проаналізованих платформ була розроблена власна архітектура для дистанційного контролю пристрою Інтернету Речей.
Були реалізовані вспоможні методи такі саме як:

- Геолокація пристрою
- Зберігання у БД

Дана система спроможна конкурувати з різноманітними платформами у цій сфері за допомогою обширності та багатонаправленого напрямку використовування.

У момент проведення тестування власної платформи усі дані відібразилися коректно судячи за порівнянням вже існуючими платформами.

ПЕРЕЛІК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Лоуренс Круз, Інтернет речей та інформаційна безпека. 2013р. [Електронний ресурс]. (Дата звернення: 15.04.2020) URL:http://www.cisco.com/web/RU/news/releases/txt/2013/03/032813c.html Top 49 Tools For The Internet of Things. [Електронний ресурс]. (Дата звернення:16.04.2020) https://blog.profitbricks.com/top-49-tools-internet-of-things/
2. Internet of Things Platforms. Postscapes. [Електронний ресурс]. (Дата звернення:08.05.2020) http://postscapes.com/internet-of-things-platforms?order=rhits
3. Хмарний сервер ThingSpeak. [Електронний ресурс]. (Дата звернення: 09.05.2020) https://thingspeak.com/
4. Хмарний сервер Xively. [Електронний ресурс]. (Дата звернення: 09.05.2020) / http://xively.com/
5. Порівняльна табиця функціоналу сервісів IoT. [Електронний ресурс]. (Дата звернення 10.05.2020) http://www.dut.edu.ua/uploads/p_421_77609272.pdf
6. Датчик DHT11 и его совместимость с Arduino. [Елктрониний ресурс] (Дата звернення 11.05.2020) https://arduinomaster.ru/datchiki-arduino/datchiki-temperatury-i-vlazhnosti-dht11-dht22/
7. Андрій Травін, Поширеність серверів і серверних мов програмування в Рунеті. 2014р. [Електронний ресурс]. (Дата звернення: 12.05.2020) https://habrahabr.ru/company/openstat/blog/212211/