

Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя
(повне найменування вищого навчального закладу)

Комп'ютерно-інформаційних систем і програмної інженерії
(назва факультету)

Комп'ютерних систем та мереж
(повна назва кафедри)

ПОЯСНЮВАЛЬНА ЗАПИСКА

до дипломного проекту (роботи)

Магістр

(освітній (освітньо-кваліфікаційний) рівень)

на тему: **Методи та засоби побудови комп'ютерної системи
відображення показників мікроклімату та споживання ресурсів
«Розумного будинку»**

Виконав: студент (ка) 6 курсу, групи СІМ-61
напряму підготовки (спеціальності)

123 “Комп'ютерна інженерія”

(шифр і назва напряму підготовки, спеціальності)

Кунинець Д.В.

(підпис)

(прізвище та ініціали)

Керівник

Лецишин Ю.З.

(підпис)

(прізвище та ініціали)

Нормоконтроль

Луцик Н.С.

(підпис)

(прізвище та ініціали)

Рецензент

Петрик М.Р.

(підпис)

(прізвище та ініціали)

АНОТАЦІЯ

Методи та засоби побудови комп'ютерної системи відображення показників мікроклімату та споживання ресурсів «Розумного будинку» // Кваліфікаційна робота за освітнім рівнем «магістр» // Кунинець Діана Василівна // Тернопільський національний технічний університет імені Івана Пулюя, факультет комп'ютерно-інформаційних систем і програмної інженерії, група СІм-61 // Тернопіль, 2021 // с. –73, рис. –34, табл. -0, аркушів А1 –8, додат. –2, бібліогр. –21.

Ключові слова: мікроклімат, відображення показників, розумний будинок.

Кваліфікаційну роботу магістра присвячено розробленню методів та засобів побудови комп'ютерної системи відображення показників мікроклімату та споживання ресурсів «Розумного будинку».

Проведено огляд методів та засобів побудови комп'ютерної системи відображення показників мікроклімату розумного будинку.

Розроблено методи та засоби побудови комп'ютерної системи відображення показників мікроклімату розумного будинку. Запропоновано до існуючих методів контролю параметрів мікроклімату додати можливість прогнозування авто регресійними методами споживання енергоресурсів розумним будинком при різних налаштуваннях параметрів мікроклімату.

Наведено програмну реалізацію додатку, що реалізує запропоновані методи та засоби побудови комп'ютерної системи відображення показників мікроклімату розумного будинку.

Розглянуто вимоги з охорони праці, при роботі з терміналами розумного будинку. Також розглянуто питання державної системи моніторингу довкілля, як складова частина національної інформаційної інфраструктури, сумісної з аналогічними системами інших країн.

ANNOTATION

Methods and means of developing a computer system for "Smart Home" indicators of microclimate and resource consumption displaying // Master thesis // Kunynets Diana Vasylivna // Ternopil Ivan Pul'uj National Technical University, Faculty of Computer Information Systems and software engineering, group CIm – 61 // Ternopil, 2021// p. –73, fig. –34 , tab. –0 , Sheets A1 – 8, Add. – 2, Ref. – 21.

Key words: microclimate, display indicators, smart home.

The master's thesis is devoted to the development of methods and tools for building a computer system for displaying microclimate indicators and resource consumption of the "Smart Home".

A review of methods and means of building a computer system for displaying the microclimate of a smart home.

Methods and means of building a computer system for displaying the microclimate of a smart home have been developed. It is proposed to add to the existing methods of control of microclimate parameters the ability to predict auto regression methods of energy consumption by a smart home with different settings of microclimate parameters.

The software implementation of the application that implements the proposed methods and means of building a computer system for displaying the microclimate of a smart home is presented.

The requirements for labor protection when working with smart home terminals are considered. The issue of the state system of environmental monitoring as an integral part of the national information infrastructure compatible with similar systems of other countries is also considered.

ЗМІСТ

ВСТУП.....	7
РОЗДІЛ 1 ОГЛЯД ТА АНАЛІЗ ІСНУЮЧИХ СИСТЕМ МОНІТОРИНГУ ТА ВІДОБРАЖЕННЯ ДАНИХ РОЗУМНОГО БУДИНКУ	10
1.1 СИСТЕМА КЕРУВАННЯ RENAУ NEA SMART 2.0	10
1.2 ІНТЕЛЕКТУАЛЬНА СИСТЕМА МОНІТОРИНГУ STEAGLE	15
1.3 СИСТЕМА LOXONE	18
1.3 ДОМАШНЯ АВТОМАТИЗАЦІЯ НА ОСНОВІ ANDROID І ВЕБ ДОДАТКІВ ЗА ДОПОМОГОЮ RASPBERRY PI	21
1.4 ВИСНОВКИ ДО РОЗДІЛУ 1	26
РОЗДІЛ 2 ПОБУДОВА МЕТОДІВ ТА ЗАСОБІВ ВІДОБРАЖЕННЯ ПОКАЗНИКІВ МІКРОКЛІМАТУ РОЗУМНОГО БУДИНКУ	28
2.1 МЕТОДИ КОНТРОЛЮ МІКРОКЛІМАТУ РОЗУМНОГО БУДИНКУ	28
2.1.1 Методи контролю температури	29
2.1.2 Методи контролю вентиляції	32
2.1.3 Методи контролю вологості.....	32
2.1.4 Методи контролю кондиціонування	33
2.2 РОЗРАХУНКИ ДЛЯ ПЕРЕДБАЧЕННЯ ТЕПЛООБМІНУ РОЗУМНОГО БУДИНКУ	33
2.3 ПРОГНОЗУВАННЯ СПОЖИВАННЯ РОЗУМНОГО БУДИНКУ ПРИ НАЛАШТУВАННІ МІКРОКЛІМАТУ	38
2.3.1 Основні описові статистики для часових рядків	41
2.4 МОДЕЛЮВАННЯ ЕНЕРГОСПОЖИВАННЯ	43
2.4.1 Тренд енергоспоживання розумного будинку	44
2.4.2 Сезонні коливання.....	45
2.4.3 Аномальні події енергоспоживання розумного будинку.....	46
2.5 ВИСНОВКИ ДО РОЗДІЛУ 2	47
РОЗДІЛ 3 ПРОГРАМНА РЕАЛІЗАЦІЯ СИСТЕМИ МОНІТОРИНГУ МІКРОКЛІМАТУ РОЗУМНОГО БУДИНКУ	48

3.1 МЕТОДИ ТА ЗАСОБИ РЕАЛІЗАЦІЇ ЗАСТОСУНКУ	48
3.1.1 .NET core	48
3.1.2. C#	49
3.2.3 UDP Protocol	50
3.2 ОПИС РОЗРОБЛЕНОГО ЗАСТОСУНКУ	51
3.3 ОПИС ІНТЕРФЕЙСУ	56
3.4 ВИСНОВКИ ДО РОЗДІЛУ 3	62
РОЗДІЛ 4 ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ.....	63
4.1 ОХОРОНА ПРАЦІ	63
4.2 ДЕРЖАВНА СИСТЕМА МОНІТОРИНГУ ДОВКІЛЛЯ, ЯК СКЛАДОВА ЧАСТИНА НАЦІОНАЛЬНОЇ ІНФОРМАЦІЙНОЇ ІНФРАСТРУКТУРИ, СУМІСНОЇ З АНАЛОГІЧНИМИ СИСТЕМАМИ ІНШИХ КРАЇН.....	66
4.3 ВИСНОВКИ ДО РОЗДІЛУ 4	69
ВИСНОВКИ.....	70
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	71
ДОДАТОК А ОПУБЛІКОВАНІ ТЕЗИ КОНФЕРЕНЦІЙ ЗА НАПРЯМОМ ДИПЛОМНОЇ РОБОТИ МАГІСТРА	73
ДОДАТОК Б ЛІСТИНГ КОДУ	81

ВСТУП

Актуальність теми роботи. З розвитком техніки в житті людей все більше використовується пристроїв автоматизації на рівні побуту. Ці пристрої автоматизації в повсякденному житті набирають форми «розумного будинку». «Розумні будинки» виконують багато функцій від охорони з сигналізацією та системою розпізнавання обличчя та голосу [1], до автоматизованих вікон та дверей, автоматизованого керування мікроклімату за допомогою датчиків температури, вологості, освітленості та ін. [2]. Всі ці функції потребують налаштування, моніторингу, керування та обліку ресурсів розумного будинку з відображенням у зручній формі для кінцевого користувача.

Ергономіка користувацького інтерфейсу розумного будинку повинна враховувати поточний стан будинку, програмування режимів роботи, налаштування обладнання і статистики витрат ресурсів «розумного будинку». Основні параметри затрат енергоресурсів та водопостачання, параметри мікроклімату такі як, рівень освітленості, вологість повітря, температура приміщення мають бути поділені між вкладками користувацького інтерфейсу.

Застосування статистичних методів обчислення енергоресурсів, має виконуватись back end частиною за стосунку і мають відповідати сучасним математичним моделям процесів газу, електро та водоспоживання і давати прогноз їх витрати при різних параметрах мікроклімату в будинку.

Загалом спрогнозувати сценарії споживання «розумного будинку» надзвичайно складно. Однак є необхідність у створенні застосунку для керування і прогнозування роботи «розумного будинку» з відповідним відображенням і налаштуванням його роботи.

Мета і задачі дослідження. Метою дослідження є розроблення методів та засобів побудови комп'ютерної системи відображення показників мікроклімату та споживання ресурсів «Розумного будинку».

Для досягнення поставленої мети необхідно розв'язати такі задачі:

— аналіз відомих методів та засобів побудови комп'ютерної системи відображення показників мікроклімату та споживання ресурсів «Розумного будинку»;

— розроблення і обґрунтування методів та засобів побудови комп'ютерної системи відображення показників мікроклімату та споживання ресурсів «Розумного будинку»;

— побудова та налаштування програмної частини комп'ютерної системи відображення показників мікроклімату та споживання ресурсів «Розумного будинку»;

— розгортання та тестування програмної частини комп'ютерної системи відображення показників мікроклімату та споживання ресурсів «Розумного будинку».

Об'єкт дослідження — процес керування та обліку показників мікроклімату та споживання ресурсів «Розумного будинку».

Предмет дослідження — методи та засоби побудови комп'ютерної системи відображення показників мікроклімату та споживання ресурсів «Розумного будинку».

Методи дослідження базуються на положеннях:

— теорії інформації для розроблення методів та засобів побудови комп'ютерної системи відображення показників мікроклімату та споживання ресурсів «Розумного будинку».

Наукова новизна одержаних результатів.

1. Удосконалено методи відображення показників мікроклімату та споживання ресурсів «Розумного будинку», шляхом моделювання особливостей споживання та статистичних методів його оцінки, що уможливило створення додатку для керування режимами роботи розумного будинку.

2. Набула подальшого розвитку авто регресійна модель споживання енергоресурсів для задачі прогнозування споживання розумного будинку, що

уможливило створення додатку для керування режимами роботи розумного будинку.

Практичне значення одержаних результатів полягає в наступному: отримані результати уможливили побудову комп'ютерної системи відображення показників мікроклімату та споживання ресурсів «Розумного будинку» у вигляді програмного додатку.

РОЗДІЛ 1

ОГЛЯД ТА АНАЛІЗ ІСНУЮЧИХ СИСТЕМ МОНІТОРИНГУ ТА ВІДОБРАЖЕННЯ ДАНИХ РОЗУМНОГО БУДИНКУ

Сучасні системи розумного будинку в більшості випадків керуються через мобільні додатки[1]. Це має свої переваги та недоліки. Смартфони стали невід’ємною частиною життя людей, відповідно з їх допомогою легко та просто керувати системою «Розумний будинок». Проте бувають випадки, коли смартфон виходить з ладу, або заряд батареї закінчився. В такому випадку може допомогти веб застосунок.

1.1 Система керування REHAU NEA SMART 2.0

REHAU NEA SMART 2.0 – це сучасна система для моніторингу та керування опаленням та охолодженням[2]. У ній є великий спектр функцій, наприклад, можливість налаштування для декількох систем та підсистем регулювання температури (рис.1.1).

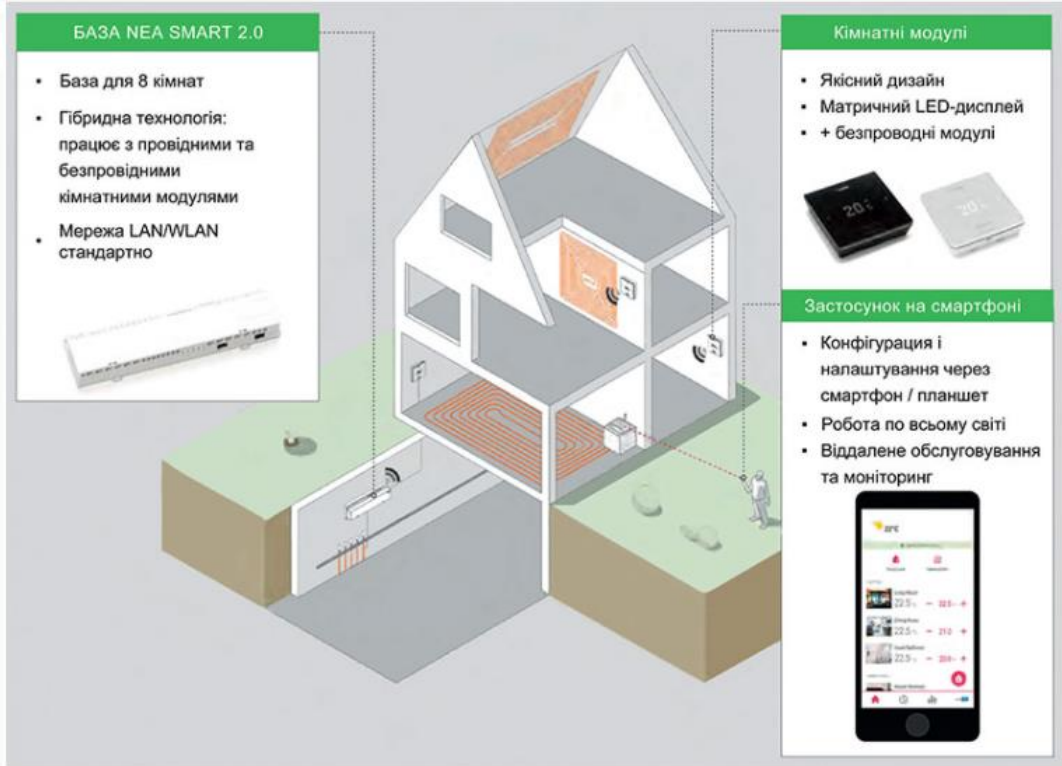


Рис.1.1. Загальна схема системи REHAU NEA SMART 2.0

На рисунку 1.2 зображено структурну схему системи.

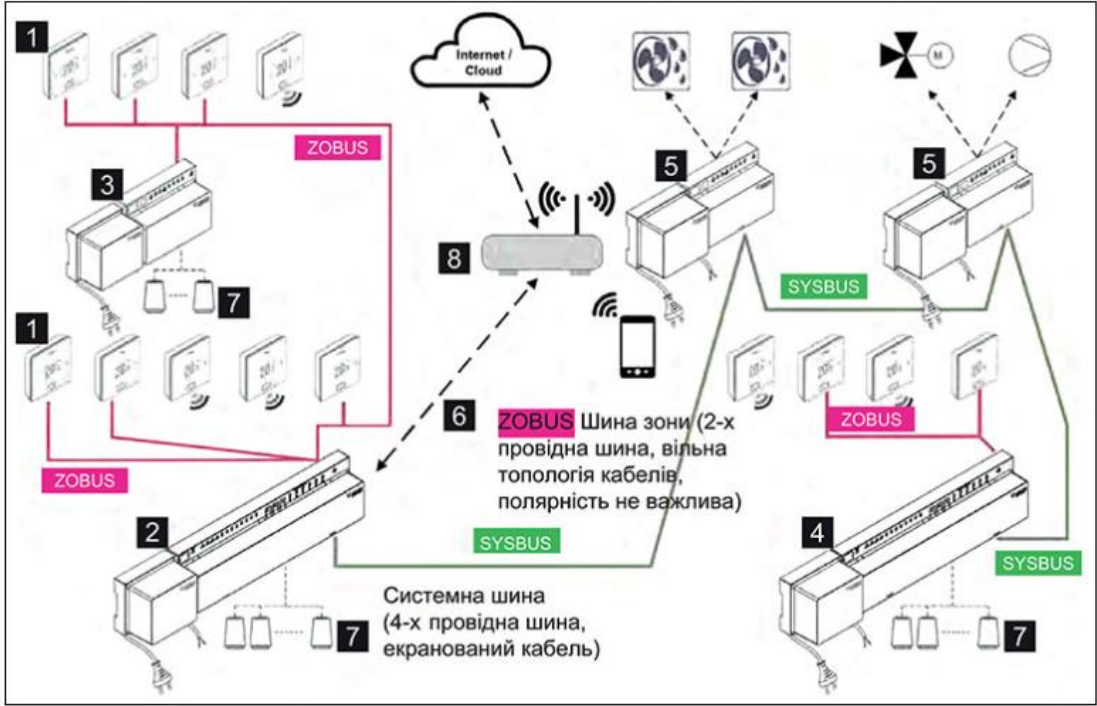


Рис.1.2. Структурна схема системи NEA SMART 2.0

Загальна структура системи NEA SMART 2.0 (див.рис.1.2) включає в себе:

- кімнатний блок;
- центральний пристрій керування з трансформатором;
- модуль розширення на більш ніж 4 кімнати;
- центральний пристрій керування з трансформатором;
- універсальний модуль розширення для мереж змішаного типу;
- інтерфейс LAN- / WLAN;
- привід керування клапанами на гребінці;
- роутер.

Керування системою можна здійснити за допомогою ПК, смартфона чи планшета, незважаючи на місцезнаходження, використовуючи мережу LAN/WLAN. Також система моніторингу має підключення до хмари, завдяки цьому користувач має можливість оптимізації, аналізу і обслуговування віддалено.

Важливими особливостями системи є:

- лаконічний дизайн кімнатних блоків, які мають можливість керування вручну;
- автоматичний контроль роботи;
- керування через веб-браузер чи застосунок;
- стандартне виконання з інтерфейсом LAN / WLAN;
- високий рівень комфорту та ефективності;
- можливість інтегрування із системами «розумний дім»;
- придатний для нових об'єктів та при модернізації старих систем.

Додатковими функціями системи є охолодження кімнат, автоматичне або ручне перемикання між станами, просушування кімнат, регулювання температури.

Керування системою може здійснюватись через:

- кімнатні блоки;
- браузер;

- хмарний сервіс (див. рис.1.2).

Ці методи керування можна використовувати самостійно, або ж поєднувати між собою. Використання лише одного засобу керування приводить до недоступності повного набору функцій. При керуванні системою лише через кімнатні блоки, користувач може змінювати налаштування та переключати режими роботи. Використовуючи інтегровані веб-сторінки, користувачу відкривається доступ до зміни параметрів системи, встановлення таймінгів та кімнатних налаштувань. Застосунок дає доступ до повного набору операцій в системі (відображення статистики, керування, налаштування).

Також кожен з методів керування має свої особливості. Наприклад, використання застосунку вимагає смартфона та доступу до мережі Інтернет. Керування з допомогою веб-сторінки можна здійснювати лише з одного пристрою, що не можна назвати зручним. Втім є можливість одночасно використовувати два або три засоби керування.

У застосунку є можливість персоналізації кімнат використовуючи ім'я, зображення чи іконку. В окремих випадках можна об'єднувати кімнати в зони, полегшуючи пошук потрібного приміщення.

Застосунок дозволяє обрати для обраної кімнати (кімнат) власний режим роботи з декількох запропонованих, що зображені на рисунку 1.3.

Доступні режими роботи:

- нормальний;
- знижений;
- режим «за часом»: користувач може визначати часові проміжки роботи системи у певному режимі, а система автоматично перемикатиме їх;
- режим очікування: всі температурні налаштування вимкнені, в приміщенні буде підтримуватися температура не нижче $+5^{\circ}\text{C}$, щоб запобігти замерзанню;
- автоматичний: встановлюється налаштування температури та часу, враховуючи місцезнаходження користувача (якщо користувач покидає межі дому чи наближається, система регулює опалення);

- відпустка: користувач налаштовує певну температуру за його відсутності;
- режим «вечірка»: користувач обирає значення температури в кімнаті на час не довше 4-х годин, система автоматично перемикає зі зниженого режиму на нормальний на обраний час і автоматично повертається у попередній режим, коли минає заданий час для вечірки.



Рис.1.3. Вибір режиму роботи для приміщення через застосунок системи NEA SMART 2.0

Ще однією функцією, якою володіє система NEA SMART 2.0, є Geofencing. Це функція «розумної» геолокації, яка відслідковує розташування пульта керування, тобто смартфона, щоб вимірювати відстань користувача від дому. Якщо користувач знаходиться поза межами дому на великій відстані, то система переводить опалення у знижений режим для економії витрат на опалення. Якщо ж користувач починає приближатися до будинку, система включає опалення. Відстань, за якої система вимикає та вмикає опалення налаштовується користувачем.

Geofencing корисна функція, оскільки зазвичай смартфон завжди поруч з власником і це дає можливість відслідковувати місцезнаходження користувача. Проте ця функція доступна тільки для кімнат, у яких налаштований режим «за

часом». Також потрібно врахувати те, що якщо активувати Geofencing на планшеті та залишити його вдома, то режим опалення не зміниться, оскільки місцезнаходження користувача визначатиметься вдома.

Застосунок системи керування опаленням може підтримувати декількох користувачів. Для цього першому користувачу надається роль менеджера, який може реєструвати інших користувачів через застосунок. Для цього потрібно перейти в розділ керування обліковим записом, після в меню керування користувачами.

1.2 Інтелектуальна система моніторингу Steagle

Характеристики системи[3]:

- до 60 бездротових пристроїв (датчиків або брелків охорони);
- дистанційна робота за допомогою Інтернету за допомогою особистого кабінету Steagle і мобільного застосунка (зміна режимів роботи охоронної системи, зберігання і перегляд історії, зміна пароля системи);
- наявність зовнішнього живлення / живлення PoE;
- відправлення SMS-повідомлень на необмежену кількість номерів; можливість активації / деактивації датчиків (вимкнення / ввімкнення оповіщення);
- налаштування шаблонів повідомлень SMS за типами датчиків;
- роздільні паролі для налаштування системи, зміни режимів, доступу в особистий кабінет;
- можливість зміни режимів за допомогою особистого кабінету, клавіатури або брелків;
- радіус роботи датчиків в панельних залізобетонних будинках - 30-50 метрів, на відкритому просторі – 200-500 метрів.

Контрольний пристрій - центральна панель Steagle ESH-1 оснащена роз'ємом RJ-45 для підключення до мережі інтернет і до блоку живлення PoE

(рис.1.4). Живлення здійснюється за стандартом IEEE 802.3af з напругою 48 В, що дозволяє розмістити центральну панель від інжектора на відстані до 100 м. Прийом сигналу від датчиків, сирен, брелоків і інших додаткових пристроїв відбувається по радіоканалу на частоті 433,92 МГц. Радіосигнал надійно захищений кодом KEELOQ - динамічне кодування контрольних передач, яке використовує високозахищений алгоритм кодування KEELOQ® з ключем шифрування і захищеною від читання програмованою комбінацією коду.



Рис.1.4. Вигляд центральної панелі керування системою

Можна багато говорити про переваги даної системи, але хотілося б звернути увагу на ключовому гідність системи – web-інтерфейсі у мобільному застосунку. Ці інструменти дозволяють керувати системою дистанційно з будь-якої точки світу за допомогою інтернету. Також за допомогою web інтерфейсу є можливість поставити на охорону і зняти з охорони, змінити умови оповіщення, редагувати пристрої і датчики (рис.1.5), фіксувати події в журналі, а також формувати всі можливі запити. І багато іншого.

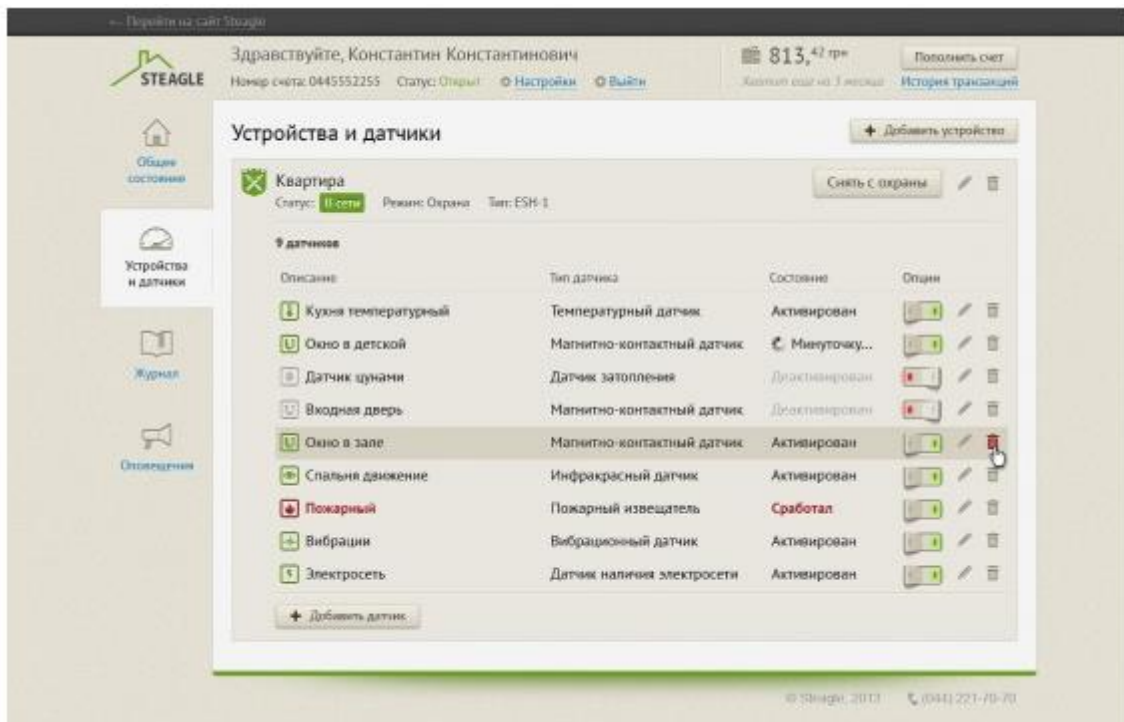


Рис.1.5. Вікно веб-застосунку

Всі події, які відбуваються або відбувалися в системі, реєструються на серверах і зберігаються в режимі real-time протягом півроку (рис.1.6). Далі події потрапляють в архів, доступ до них можливий за запитом. Дані зберігаються в зашифрованому вигляді. Завдяки розширеному функціоналу журналу особистого кабінету можна побудувати будь-який запит в базу даних подій.

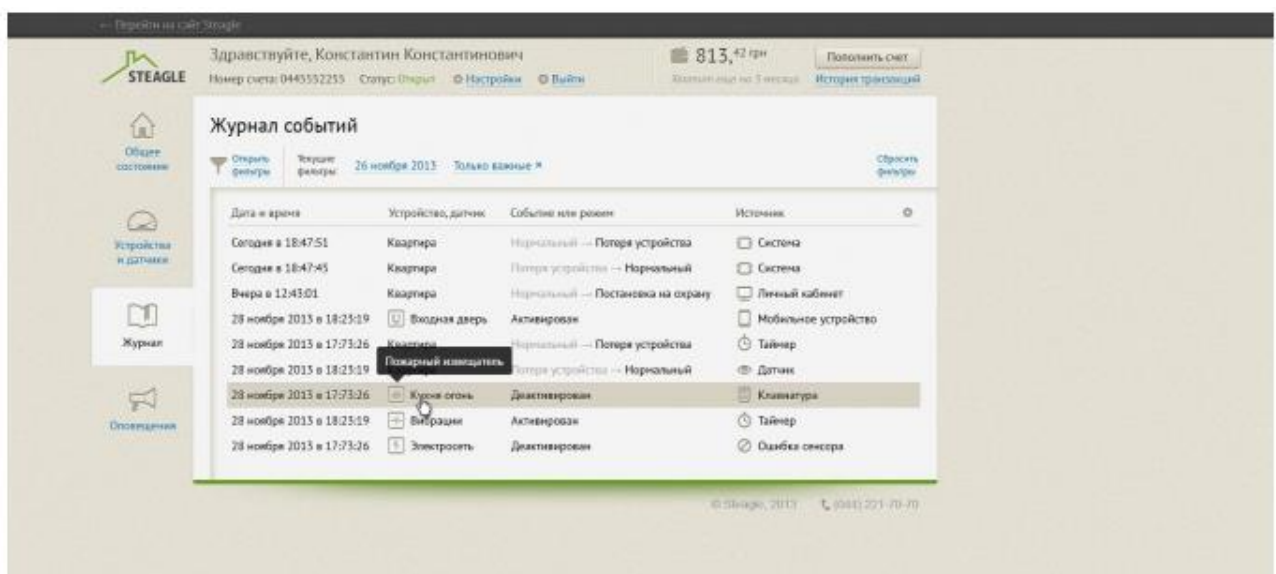


Рис.1.6. Вікно журналу подій

До системи підключені наступні шлюзи оповіщення:

- sms інформування;
- голосові повідомлення;
- e-mail інформування;
- виїзд оперативної групи.

Є можливість гнучко налаштувати всі умови оповіщення в особистому кабінеті в будь-який час (рис.1.7).

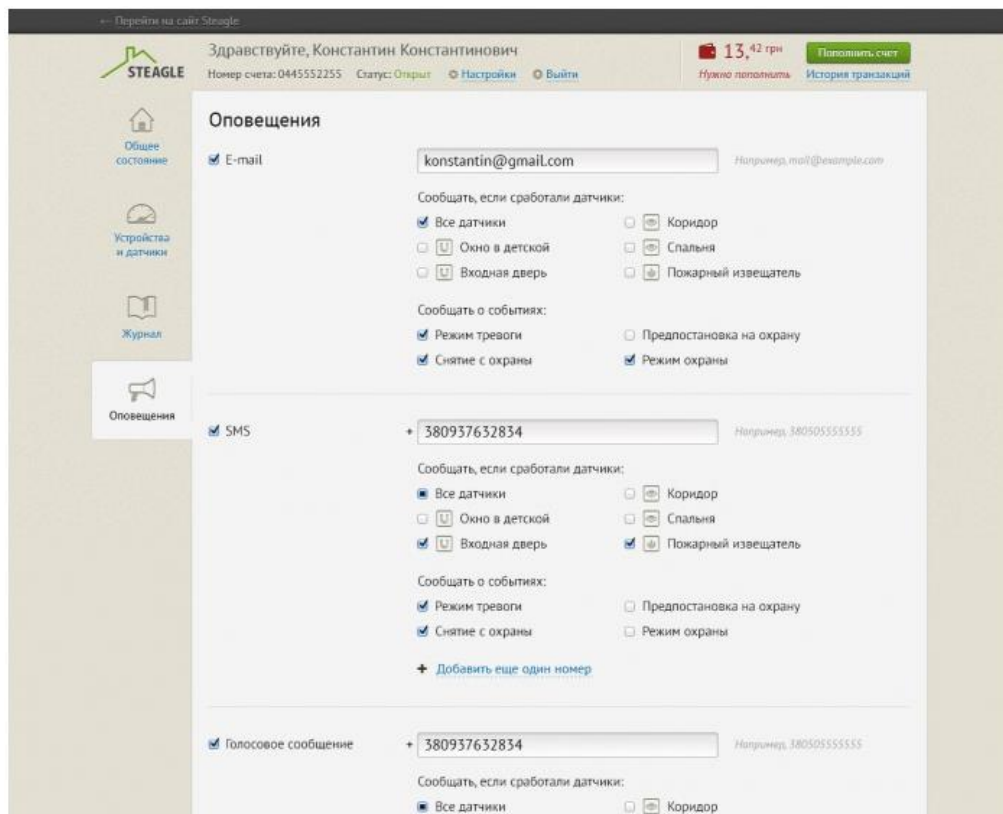


Рис.1.7 Вкладка оповіщення

1.3 Система LOXONE

Австрійська компанія пропонує інтелектуальну автоматизацію та інтуїтивно зрозумілий контроль освітлення, багатокімнатне аудіо, опалення, безпеку, затінення, управління енергією та багато іншого[4].

Web-інтерфейс створюється автоматично, лише потрібно вказати URL адресу Miniserver'a. Якщо смартфон недоступний і немає можливості

скористатися додатком Loxone App, то виручить будь-який комп'ютер з будь-яким браузером.

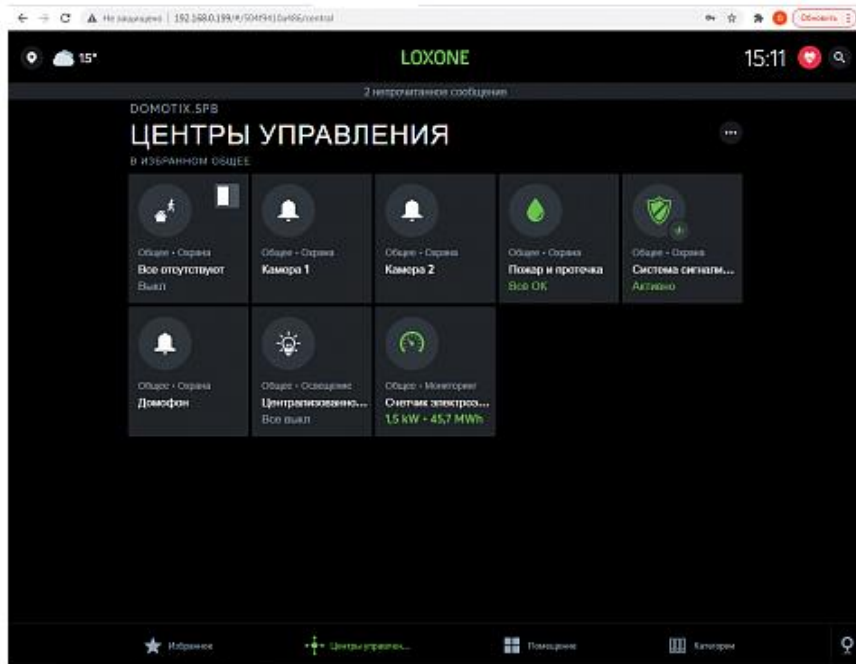


Рис.1.8. Вкладка центру управління системою

З допомогою застосунку можна відслідковувати поточний стан освітлення, клімату та інших показників у окремих кімнатах (рис.1.9).

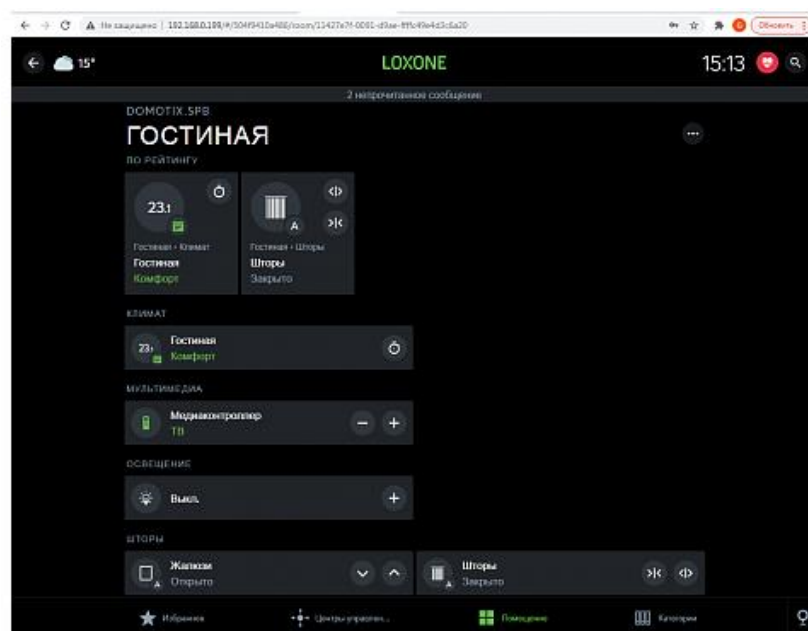


Рис.1.9. Сторінка поточного стану системи у вітальні

Також є можливість відслідковувати дані за категоріями та персоналізувати їх, тобто додавати певні категорії у «вибране» для швидкого доступу.

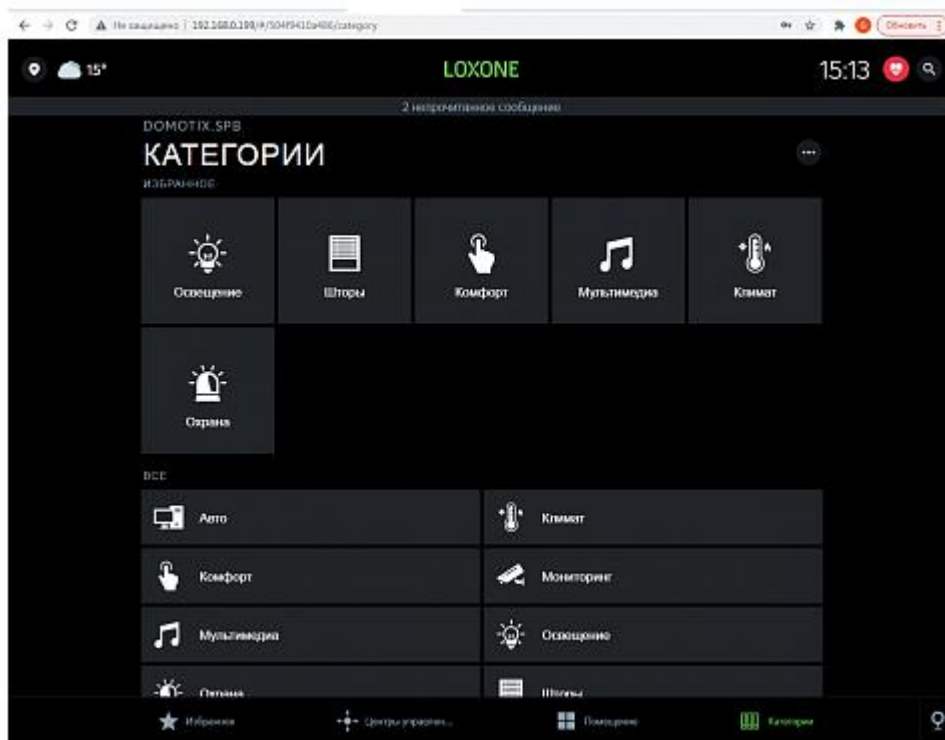


Рис.1.10. Сторінка відображення категорій управління

Для користування потрібно:

- будь-який інтернет-браузер з html5 або вище;
- генерується автоматично;
- розумний будинок доступний у будь-який момент.

Miniserver оснащений web-сервером, який автоматично генерує web-інтерфейс.

Потрібно просто ввести адресу контролера Miniserver в адресний рядок браузера (Microsoft Internet Explorer, Mozilla Firefox, Google Chrome) або браузера смартфона. Не потрібно ніякого додаткового обладнання - все зробить Miniserver, немає необхідності купувати якісь ліцензії, все абсолютно безкоштовно. Даний спосіб використовується в основному при пуск-налагодженні системи.

1.3 Домашня автоматизація на основі Android і веб-додатків за допомогою Raspberry Pi

Це гнучка, недорога система дистанційного керування побутовою технікою, розроблена доцентами університету Аттики. Управління здійснюється або з Android або веб-додатку. Обидва додатки спілкуються з загальним сервером, який надає дані стану в режимі реального часу, як для клієнтів браузера, так і для Android. Система заснована на Raspberry Pi вбудованій системі.

Метою роботи була розробка легко налаштованого і інтуїтивно зрозумілого інтерфейсу, але потужного і надійного, щоб забезпечити своєму користувачеві повний моніторинг і контроль можливостей над своїми домашніми і домашніми пристроями[5]. Це досягається або з Android або веб-додатку. Завдання полягало в тому, щоб знайти спосіб, в який два додатки будуть спілкуватися з загальним сервером, який буде надавати дані про стан в режимі реального часу не тільки в додатку Android, але і в веб-додатку, а тим часом мати можливість контролювати підключені пристрої. Крім того, потрібно було зберегти веб-технології HTML, PHP і JavaScript без необхідності створення Java Applet.

Ця проблема вирішилась, використовуючи протокол зв'язку WebSockets, який призначений для реалізації в веб-браузерах і веб-серверах, але може використовуватися будь-яким клієнтом/сервером. Але реальна важливість це те, що WebSockets забезпечив спосіб побудови масштабованих, в режимі реального часу веб-додатків. Так, за допомогою стандарту «Java Web Sockets JSR-356» реалізували сервер WebSocket, який здатний одночасно обслуговувати як в режимі реального часу Web додаток, так і Android, контролюючи підключені пристрої за допомогою бібліотеки PI4J. Відповідний сервер, який працює на додатку, є Glassfish.

Завдання полягало в тому, щоб знайти спосіб, в який два додатки будуть спілкуватися з загальним сервером, який буде надавати дані про стан в режимі

реального часу не тільки в додатку Android, але і в веб-додатку. Крім того, ми повинні були зберегти веб-технології HTML, PHP і JavaScript без необхідності створення Java Applet.

Проблема виникає з того, що у є 2 додатки, засновані на абсолютно різних технологіях. Одна технологія повинна бути реалізована в PHP-HTML5, а інша з Android-Java. Зв'язок кожної окремої програми з основним сервером здійснюється без особливих проблем:

- веб-частина буде просто мати деякі PHP-скрипти, які будуть називатися скриптами Python, які в свою чергу будуть контролювати стан портів GPIO. Веб-додаток отримує дані статусу з запитом Long Poll Ajax, але не в режимі реального часу;

- частина Android буде контактувати з сервером Java TCP/IP (те ж саме, що інформує його про дані) і таким чином контролює GPIO. Java TCP/IP сервер буде інформувати додаток Android для будь-якої зміни стану в режимі реального часу.

Є деякі проблеми, які виникають:

- потрібно окремо вирішувати кожну комунікацію;
- потрібно запитати сервер з запитами Long Poll Ajax, споживаючи непотрібні дані;
- не досягається 100% оновлення даних в реальному часі. Якщо, наприклад, запит Ajax має 5 секунд застосування даних, то дані оновлюються кожні 5 секунд, а не коли оновлення дійсно можливе;
- запуск конфлікту: що відбувається, коли додаток Android доручив відкрити пристрій, і веб-додаток ще не бачив зміни статусу;
- можливе перевантаження Apache, і, отже, Raspberry (через обмеження ємності), Ajax запитів.

У розв'язанні проблеми допоміг WebSockets. WebSockets - це протокол, який забезпечує повний дуплекс зв'язок через одне TCP-з'єднання. Це відносно нова технологія, яка стала запропонованою специфікацією IETF (Internet

Engineering Task Force) у грудні 2011 року. Технічно, WebSockets дозволяє встановити між клієнтом і сервером довгоочікуване єдине з'єднання TCP, усуваючи необхідність опитування сервера і дозволяючи відправляти повідомлення назад і вперед, зберігаючи з'єднання відкритим. Але реальне значення полягає в тому, що WebSockets надає спосіб створення масштабованих веб-додатків в реальному часі. За допомогою стандарту Java Web Sockets JSR-356 нам вдалося реалізувати сервер WebSocket, який здатний обслуговувати одночасно Web і додаток Android, без необхідності наявності двох різних серверів або методів управління GPIO. Спілкування дуже просте і дотримується правил WebSockets, що диктують використання текстових повідомлень. Текстові повідомлення в нашому випадку знаходяться у формі JSON. Після того, як нам вдалося створити центральну точку зв'язку між додатками/пристроями, ми повинні були дати можливість серверу керувати підключеними пристроями. Це було досягнуто за допомогою PJ4J бібліотеки.

Архітектура запропонованої системи проілюстрована на рисунку 1.11. Ця система складається, по суті, з трьох основних компонентів: Web-Frontend, Android-додаток і апаратний компонент, який розміщує основний сервер. Основний сервер пов'язує окремі додатки (Web-Frontend і Android) з Raspberry Pi і перевіряє порти GPIO, надаючи в режимі реального часу дані про операційний стан двох додатків.

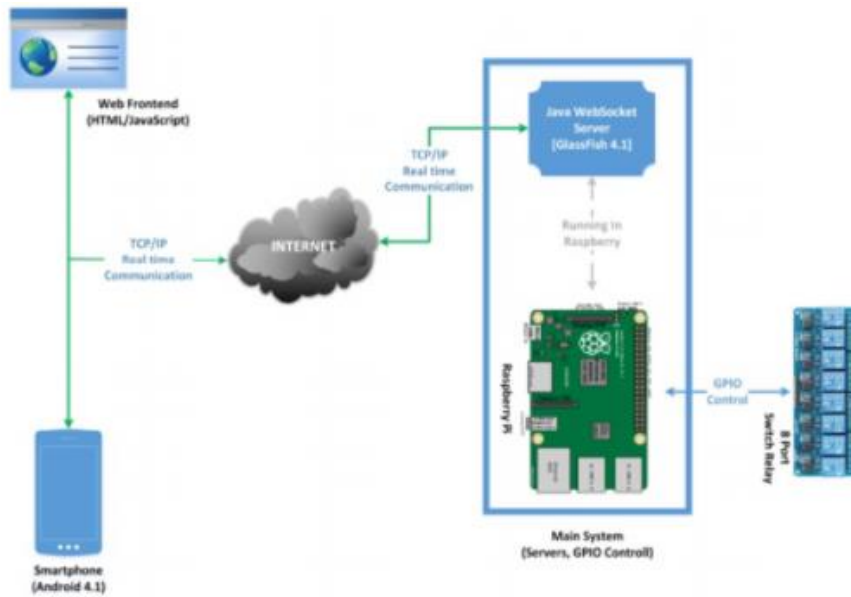


Рис.1.11. Схема архітектури системи домашньої автоматизації

Програмне забезпечення, яке було розроблене/використане, стосується додатків Web і Android, а також основного сервера. Як операційну систему було вибрано Raspbian, який базується на операційній системі Debian Wheezy Linux і був оптимізований для використання з Raspberry Pi.

Для розробки Web-додатку, що стосується частини клієнта, були використані технології HTML5 та JavaScript. На стороні сервера є сторінки PHP, які зчитують статичні дані з файлів JSON. Дані, які інформують про порти GPIO, які використовуються або пристрої, підключені до кожної кімнати. Для реалізації веб-додатку використано CS6 Adobe Dreamweaver.

Веб-додаток дозволяє керувати віддаленими пристроями через веб-сайт. Цей веб-сайт має 3 основні інтерфейси:

- Login Form (рис.1.12). Інтерфейс входу - простий скрипт php, який в свою чергу підключений до бази даних SQLITE. Після успішного входу користувача, піднімається екран вибору кімнати (рис.1.13);



Рис.1.12. Сторінка входу в систему



Рис.1.13. Сторінка вибору кімнати

– екран для вибору кімнати (див. рис.1.13). На цьому екрані користувач вибирає цікаву кімнату. Кімнати зберігаються у файлі JSON. Файл JSON містить назву кімнати, її опис та тип (тобто, якщо це вітальня, спальня тощо). Для кожної кімнати є об'єкт JSON;

– вид на кімнату і відповідні підключені пристрої (рис.1.14). На цьому екрані відображається назва обраного приміщення, його опис і всі підключені пристрої, які належать цій кімнаті, згруповані на дошці. Для кожного пристрою на платі є репрезентативна іконка (за типом пристрою), її назва, тип, опис, і, нарешті, кнопка стану, яка вказує стан пристрою (відкрити або закрити) і дві операційні кнопки (ON, OFF). Вся ця інформація для кожного пристрою зберігається у файлах JSON. Спочатку всі кнопки стану вимкнені і забарвлені в сірій колір. Це вказує на те, що статус кожного пристрою ще не отримано від сервера. Як тільки сервер отримає дані про стан, індикаторні вогні змінять колір і будуть активовані відповідні кнопки стану. Якщо, наприклад, відкривається

пристрій, то відповідна кнопка стану стане зеленою, а її операційна кнопка OFF буде активована.



Рис.1.14. Сторінка вигляду кімнати та підключених пристроїв

У цій роботі запропоновано і впроваджено дешеву систему, яка забезпечить інтелектуальний контроль будинку за автентифікацією користувача. Це було досягнуто за допомогою використання недорогих пристроїв і розробки двох зручних інтерфейсів для Web і Android додатків. Сервер WebSocket, що працює на картці Raspberry Pi, здатний служити одночасно в режимі реального часу як веб-додаток, так і Android, контролюючи підключені пристрої.

1.4 Висновки до розділу 1

Під час пошуку існуючих систем керування та моніторингу розумних будинків було виявлено, що більша частина цих систем для керування та відображення використовує мобільні застосунки. Невелика частина систем використовує програмне забезпечення для персонального комп'ютера. Щодо веб-застосунків, то їх кількість значно менша за мобільні додатки.

В результаті дослідження існуючих веб-застосунків для моніторингу даних розумного будинку було виявлено, що деякі з них мають незручний

користувацький інтерфейс, відображено не усі можливі та необхідні параметри. Дизайн таких застосунків не лаконічний, важкий для сприйняття користувачем. Це є вагомим обґрунтуванням розробки якісного, зручного та багатофункціонального веб-застосунку для моніторингу даних мікроклімату розумного будинку.

Ще одним недоліком існуючих рішень є те, що застосунки не виконують обчислень спожитих ресурсів. У користувацькому інтерфейсі немає можливості переглянути середнє значення спожитої електроенергії, газу чи води. Ця функція є дуже корисною, оскільки в будь-який момент часу користувач може перевірити значення витрачених ресурсів.

РОЗДІЛ 2

ПОБУДОВА МЕТОДІВ ТА ЗАСОБІВ ВІДОБРАЖЕННЯ ПОКАЗНИКІВ МІКРОКЛІМАТУ РОЗУМНОГО БУДИНКУ

2.1 Методи контролю мікроклімату розумного будинку

Для підтримки заданого рівня мікроклімату і якості повітря в приміщенні, необхідно розробити регуляторну систему з економічним споживанням і потужністю для зазначених умов експлуатації. Системи опалення, вентиляції, та кондиціонування складаються з ряду підсистем, кожна з яких можуть мати нелінійні характеристики, які варіюються з час[6]. Більш того, параметри динамічного стану зазвичай змінюються в залежності від погоди і порушень. Використання звичайної схеми управління для такої технології і системи управління з великою кількістю регульованих параметрів виявилися неефективними.

Для управління системою опалення, вентиляції, та кондиціонування на об'єкті може використовуватися класична система регулювання, хоча система може впливати при зміні параметрів об'єкта. Впровадження додаткових елементів управління має певні недоліки, такі як незручні налаштування параметрів, перешкоди та незначні відхилення від заданих значень.

Враховуючи як свої переваги, так і недоліки, нечіткий контроль на основі є хоча б адекватним альтернативним рішенням[7]. Однією з головних переваг таких систем є легка міграція без зміни структури контуру управління. У порівнянні з традиційними методами автоматичного управління, застосування нечітких систем дозволяють швидко аналізувати дані і отримувати результати з високим рівнем точності.

Особливістю методів вирішення задач нечіткої логіки є наявність набору правил, що складаються з набору умов і висновків. Нечіткі методи керування зараз є одним з найбільш важливих кроків в розвитку розумних технологій для створення високоорганізованої системи управління. Спеціальна система

контролю якості, зокрема, має низьку чутливість до змін параметрів управління об'єктом. Синтез систем управління з нечіткою логікою в додатку сучасної апаратної та програмної підтримки простіше, ніж синтез традиційних систем управління. Однак цього зазвичай недостатньо для контролю температурного режиму повітря. Система управління складається з кулера, обігрівача, зволожувача повітря, регульованого вентилятора, що забезпечує подачу повітря, нечіткого контролера, датчиків зовнішньої температури, відкритого датчика вологості, датчиків температури всередині, датчиків вологості в приміщенні, зони управління та системи каналів повітря.

2.1.1 Методи контролю температури

Регулювання температури повітря в системі «розумний будинок» здійснюється за допомогою спеціальних датчиків, розміщених всередині і зовні. Вони вимірюють температуру, і якщо вона не відповідає зазначені критерії, відправляють сигнал на основну панель управління[8]. Далі застосовується ряд заходів для досягнення оптимальної продуктивності, включаючи автоматичне регулювання температури, яка підтримує задану температуру приміщення. Крім того, доступний діапазон температур від нуля до 125°C.

Завдання полягає в тому, щоб визначити відповідну температуру рівня управління регулятором цифрового аналогового перетворювача, вхідні змінні яких такі: e , яка є різницею між бажаною і фактичною температурою за Цельсієм, і Δe , перша похідна зміни температури під час обчислювання циклу, у °C/хв.

$$\Delta e = T_3(t) - T_n(t), \quad (2.1)$$

де $T_3(t)$ і $T_n(t)$ є заданими і поточними температурами у °C/хв. Природно, чим більше різниця температур в конкретний момент, тим більше необхідне збільшення охолодження або швидкість нагріву.

Швидкість зміни температури вказується як

$$\Delta e = \frac{e(t_1) - e(t_2)}{t_1 - t_2}. \quad (2.2)$$

Коли температура кімнати наближається до заданої точки, швидкість зміни температури знизиться, наприклад, під час етапу охолодження кондиціонера. Визначаємо лінгвістичні змінні для нечітких e і Δe встановлює за допомогою відповідної ідентифікації функції приналежності $\mu(e)$ і $\mu(\Delta e)$. Ми будемо дві функції приналежності.

В першій аргументом є різниця температур (e) (рис.2.1 (а)), і в другій, аргументом є швидкість зміни температури (рис.2.1 (б)). Перша функція має температурний діапазон від -6 до 8 °С, і друга функція має температурний діапазон -6 до 8 °С /хв. Для $\mu(e)$ і $\mu(\Delta e)$ (рис.2.1) ідентифікатори визначаються як: відхилення сильно позитивне (PB), відхилення помірно позитивне (PM), відхилення трохи позитивне (PS), відхилення дорівнює нулю (Z), відхилення трохи негативне (NS), відхилення помірно негативне (NM), і відхилення сильно негативне (NB). Результат від спільного ефекту двох членських функцій значення вихідного параметра визначається через відповідну програму, вбудовану в логічний пристрій. За допомогою функції приналежності (див.рис.2.1 (а)), бажаний режим роботи систем опалення та охолодження дано.

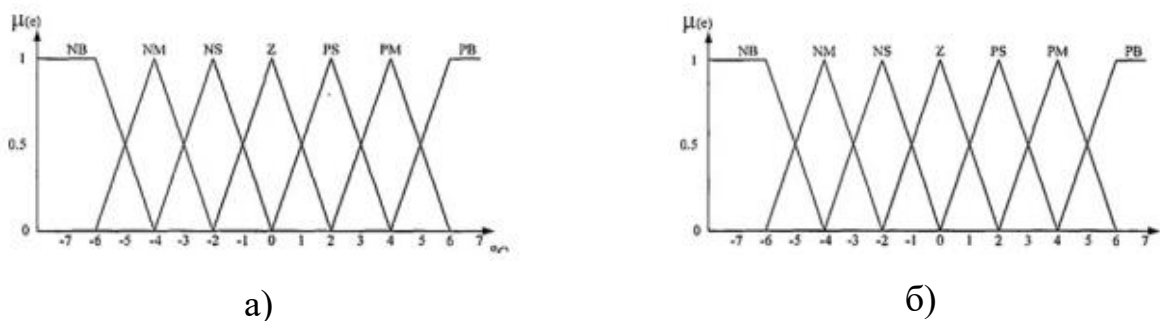


Рис.2.1 – Лінгвістичні функції приналежності для вхідних параметрів

Нечіткими змінними є ідентифікатори: «сильне охолодження» (C3), «помірне охолодження» (C2), «невелике охолодження» (C1), «незмінне» (NO), «нагрівання 1» (H1) і «нагрівання 2» (H2), і розраховуються за допомогою

частоти обертання вентилятора на основі заданого основного правила (див.рис.2.1(b)). Нечіткі змінні, що відповідають ідентифікаторам частоти обертання вентилятора «висока» (швидко), «нормальна» (помірна), «низький» (низький) і «нульовий» (Z). Функція виводу аксесуарів (рис.2.2) показує обробку правила, і підсумовує сигнали відповіді для генерації виводу команди.

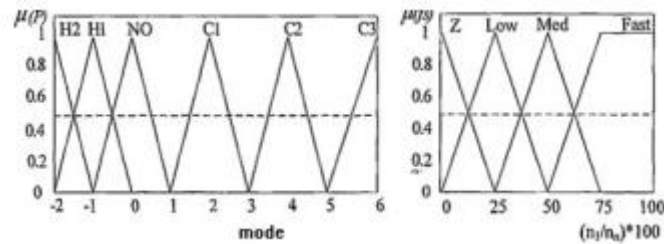


Рис.2.2 – Лінгвістичні функції приналежності для вихідних параметрів

Вибрані функції на цьому рисунку забезпечують вихід, що складається з двох рівнів нагріву (H1 і H2), трьох охолодження рівні (C1, C2, C3), і нормальний рівень (NO), і можна уявити кілька різних рівнів додаткового опалення або охолодження, із значенням нагріву більше H1 або H2; в крім того, значення C3 більше, ніж у C2 і C1[9].

При підключенні до вихідної приналежної функції та відповідної функції нечіткого контролю, отримано чіткий сигнал до контрольної дії[21]. У цьому регістрі, керуючий сигнал визначається виходячи з рівня нагрівання або охолодження з даних $[-2, -1, 0, 1, 2, 3\dots]$. Зв'язки між входом і виходом перераховані в таблиці нечітких правил (Таблиця 1). Кожному запису відповідає нечітке правило. Наприклад, якщо поточна внутрішня температура є 30°C , а задана температура 24°C , що означає $e = -6^\circ\text{C}$ (NB), і якщо відхилення сильно позитивне, то приміщення охолоджується до рівня C3, швидкість обертання вентилятора «дуже висока» (швидко), і швидкість нагріву буде дорівнювати нулю. Логічне позначення є наступним: якщо $e = \text{NB}$, використовується C3 і швидкість обертання вентилятора є швидкою. Досягається зв'язок між вхідними та вихідними даними через виразне лінгвістичне перетворення вхідних даних.

Вхідні функції членства можна знайти за допомогою імплікації та підсумовування за допомогою бази правил та дефузифікації лінгвістичного виходу в числових значеннях (ступінь нагріву або охолодження). Від'ємні числа відносяться до ступеня нагріву, а позитивні значення відносяться до ступеня охолодження. Вихідне значення контролера є дискретним числом $[-2, 1, 0, 1, 2, 3\dots]$.

2.1.2 Методи контролю вентиляції

Свіже повітря - одне з головних переваг будинку. Однак, надія на те, що свіже повітря буде кімнаті виключно через вентиляційні отвори, велика помилка. Особливо це стосується приміщення зі складною геометрією і функціональністю. Головна особливість інтелектуальної системи управління вентиляцією є можливість аналізу необхідності вентиляції приміщення.

«Інтелектуальна» вентиляція здатна не тільки приносити свіже повітря в будинок, але також гарантує, що повітря чисте. Така система контролюється на основі розкладу, наприклад, вентиляція включається автоматично в певний час, наприклад, за годину до сну, або детектори диму спрацьовують у випадку високих концентрацій вуглекислого газу [10]. За оптимальними параметрами повітря система буде працювати в економічному режимі. Інтелектуальна вентиляція, яка гарантує, що вдома є постійний приплив свіжого повітря, видаляє забруднене повітря, не пропускає вуличний шум і створює здоровий клімат, все без участі користувача або контролю.

2.1.3 Методи контролю вологості

Оптимальною вологістю для нормального життя людини вважається 40%-60%. Відхилення в будь-якому напрямку можуть привести до негативних наслідки. Зокрема, надмірно сухе повітря зменшує стійкість організму до вірусних захворювань, і занадто велика вологість в приміщенні сприяє росту цвілі і борошнистої роси, а також викликає різноманітність алергічних реакцій.

Досягти оптимального рівня вологості не так просто. А нормальний кондиціонер не здатний впоратися з такими завданнями. На допомогу в цій ситуації приходять спеціальні зволожувачі і осушувачі, які можуть бути успішно використані в системі «розумний будинок». Спеціальні датчики контролю рівня вологості посилають сигнали на блок управління, який автоматично налаштовує продуктивність шляхом осушення або зволоження повітря.

2.1.4 Методи контролю кондиціонування

Кондиціонери використовуються в багатьох квартирах і будинках. Сучасні моделі характеризуються високою функціональністю і різноманітність конструкцій. Однак таке обладнання не завжди використовується ефективно [11]. Для вирішення цієї проблеми може бути використана система «розумний будинок», яка встановить роботу обладнання, а саме, автоматичне ввімкнення або вимкнення всіх кондиціонерів, встановлених в різних приміщеннях, або кожного блоку окремо, регулювання температура, інтенсивності і напрямку потоку повітря.

Багатоконтурна система керування параметрами мікроклімату потребує задання оптимальних його параметрів, які впливають на контури управління температурою, вологістю, вентиляцією та кондиціонування. Ці параметри мають бути враховані при створенні користувацького інтерфейсу з можливістю створення прогнозу при кожному варіанті налаштувань і донесенню його до користувача у зручній формі.

2.2 Розрахунки для передбачення теплообміну розумного будинку

Для кожної зони крите середовище характеризується високим рівнем зайнятості і рухом людей, які впливають на проектування параметрів мікроклімату. Особливе значення має регулювання рівня вологості для внутрішнього середовища.

Виходячи з наведеного вище, і з урахуванням мети окремих приміщень, призначених для різних видів діяльності, один може визначити мінімальні конструктивні параметри об'єму витрат для окремих приміщень. Розрахунки повинні брати до уваги кількість повітряної водяної пари, що утворюється при потовиділенні, а також кількість людей. Розрахунок об'ємного потоку виключно на основі необхідних даних повітряного обміну (зазвичай з 60 до 120 м² / год на людину) недостатньо, тому що потрібна корекція для пост-крапельного і теплового попиту.

Після визначення загального обсягу генерації води в приміщенні (Q_{mv}), об'ємна витрата повітря, необхідна для видалення вологи з повітря, визначається виходячи з різниці між абсолютною вологістю і внутрішньою подачею повітря, і розраховується наступним чином:

$$Q_{ma} = \frac{A_{mi}}{x_a} - x_m, \quad (2.3)$$

$$V - a = \frac{A_{ma}}{p_a}. \quad (2.4)$$

Кількість повітря, необхідного для нейтралізації фізичного тепла навантаження (Q_s) визначається виходячи з різниці температур між внутрішньою частиною кімнати і свіжим повітрям зовні, і обчислюється наступним чином:

$$V_a = \frac{Q_s(\text{термозавантаження})}{0.34 * \Delta t}. \quad (2.5)$$

Слід зазначити, що під час фізичних навантажень людини організм споживає з повітря відчутну кількість кисню. Таким чином, чим інтенсивніше діяльність, призначена для приміщення, тим більш важливим є забезпечення необхідного повітрообміну, незалежно від того, скільки насправді тепла і вологості приміщення задовольнить нормативні вимоги або розраховані дані. Щоб забезпечити необхідний комфорт, кімнати в експлуатації повинні

безперервно постачати постійним напливом зовнішнього повітря. На основі порівняння внутрішніх і зовнішніх факторів, мікроконтролер обчислює цільову температуру в будівлі кожну хвилину. У цьому розрахунку використовуються не тільки отримані дані від датчиків на конкретний момент, але так само і фоновий стан мікроклімату будівлі. Особливості цього алгоритму наступні:

- можливість підтримувати бажаний клімат в приміщенні;
- скоординоване управління всім клімат-контролем системи, що дозволяє досягнути оптимальне співвідношення економії теплових ресурсів та якості кліматичного контролю;
- моніторинг фактичного нагріву та вентиляції, а також перерозподіл тепла, коли схеми працюють неправильно.

Програмний комплекс дозволяє архівувати та графічно відображати всі встановлені і виміряні параметри клімату в режимі реального часу, і розрахувати їх відповідно до заданого алгоритму контрольні дії. Щохвилини контролер передає інформацію про 28 виміряних і обчислених значеннях. Програма забезпечує аналіз систем управління в реальному часі та видачу текстового режиму і діагностичних сигналів тривоги. Кліматична зміна параметрів, стратегія управління, налаштування коригування алгоритму, коефіцієнти калібрування всіх датчиків вимірювання, і граничні значення виміряних вище величин, які є видані відповідними діагностичними та сигнальними повідомленнями, все подано в табличній формі. При передачі заданих значень і кліматичних параметрів, які впливають на управління контролером, введені дані автоматично контролюються в термінах прийнятності, і, таким чином, передача неправильних значень уникається.

Програмна система дозволяє встановлювати динаміку клімату протягом дня в табличній формі і графічно відображати відповідно до зазначених вимог.

Теплові процеси нагрітого простору - це ряд взаємопов'язаних субпроцесів теплообміну та теплопередачі між елементами системи, що включають в себе внутрішнє повітря, огорожу будівлі, внутрішній вміст, обігрівачі та систему вентиляції. Ці елементи взаємодіють між собою і навколишнім середовищем

через тепло і масову передачу. Крім того, тепло передається через конвекцію від радіаторів до внутрішнього повітряного простору. Тепло від внутрішньої до зовнішньої поверхні корпусу пропускається через теплообмін.

Крім теплових процесів, маса процеси передачі повинні розглядатися через необхідність вентиляція приміщення. Таким чином, нагріту будівлю можна відобразити на графіку $G = \{X, U\}$, де X - множина вершин, кожна з яких x_k ($k = 1, n$), що відповідає внутрішньому повітряному простору k -й будинок, і U - це безліч ребер, кожен з яких відповідає тепловому або матеріальному потоку між сусідніми приміщеннями. Також потрібно виділити вершину x_0 , яка представляє собою навколишнього повітря.

Кожна вершина графа x_k може мати термодинамічні параметри повітря, V_k і T_k , які є об'ємом і температурою, відповідно. Для U_{ij} кожне ребро відповідає вектору $(H_{ij}, c_{ij}, p_{ij}, F_{ij}, \lambda_{ij})$, де H_{ij} товщина, F_{ij} - поверхня, c_{ij} - це конкретне тепло, p_{ij} - це щільність, а λ_{ij} - огорожа теплопровідності між елементами конструкцій i та j . На додаток до цих системних параметрів визначення чисто теплові процеси, необхідно також ввести параметри відповідає за масову передачу, пов'язану з вентиляцією приміщень і можливим обміном повітряних мас між кімнатами. Цей параметр можна переглянути навскіс в симетрична матриця G_{ij} ($i, j = 0, n$) для ваги (або об'єму) повітряного потоку між i -м і j -м елементами.

Кожен край структурного графа u_{ij} символізує тепло потоку j з q_{ii} для елемента x_i до елемента x_j . У свою чергу, спека потік q_{ii} визначається законами теплопередачі відповідно з якими, в межах паркану між i -м і j -м номерами, утворюється нестійке температурне поле $T_{ii}(x, t)$, де $x \in [0, H_{ii}]$ - лінійна координата, перпендикулярна до поверхонь забору. Задане поле температури, описане диференціальним рівнянням теплопровідності є

$$\frac{\partial T_{ij}}{\partial t} = a_{ij} \frac{\partial^2 T_{ij}}{\partial x^2}, \quad (2.6)$$

де $a_{ij} = \frac{\lambda_{ij}}{c_{ij} p_{ij}}$ коефіцієнт теплопровідності. На межі огорожі, $x = 0$ і $x = H_{ij}$, температура задовольняє третій тип граничних умов, завдяки чому конвективний тепловий потік від повітря до огорожі є тепловий потік всередині корпусу, тобто,

$$a \left(T_i - T_{ij}(x, t) \right)_{x=0} = -\lambda_{ij} \left(\frac{\partial T_{ij}(x, t)}{\partial x} \right), \quad (2.7)$$

$$a \left(T_{ij}(x, t) \right)_{x=H_{ij}} = -\lambda_{ij} \left(\frac{\partial T_{ij}(x, t)}{\partial x} \right)_{x=H_{ij}}, \quad (2.8)$$

де a - конвективний коефіцієнт теплопередачі. У свою чергу температура T_i кожного структурного елемента може бути визначається на основі рівняння теплового балансу, яке має наступна форма:

$$c_v p_B V_i \frac{dT_i}{dt} = a \sum_{j=0}^n F_{ij} (T_{ij}(0, t) - T_i) + c_p \sum_{j=0}^n G_{ij} (T_j - T_i) + a F_{ai} (T_{ai} - T_i) + Q_i(t), \quad (2.9)$$

де c_v і c_p - специфічне ізохоричне і ізобаричне тепло ємності повітря, відповідно; p_B - щільність повітря; F_{ai} і T_{ai} - площа поверхні і температура внутрішнього акумулятора в i -му номері, відповідно; $Q_i(t)$ є тепловим потоком від тепло, що подається з обігрівачів.

Формули системи 2.6 і 2.9 повинні бути доповнені через систему диференціальних рівнянь, що описують теплові процеси у внутрішньому акумуляторі, який складається з різних виробничих обладнання, меблі та інші об'єкти, що заповнюють простір:

$$c_i \frac{dT_{ai}}{dt} = a F_{ai} (T_i - T_{ai}), \quad (2.10)$$

де c_i - ємність теплоакумуляторів.

Таким чином, математична модель теплових процесів будівля, що нагрівається, є системою $n(n - 1)/2$ диференціала рівняння в часткових похідних форми (2.6) з границею умови (2.7) і (2.8), і система $2n$ звичайного диференціала рівняння теплового балансу (2.9) і (2.10). Для інтеграції системи (2.6), (2.9) і (2.10) необхідно встановити початкові умови для температури T_i ($i = 1, n$), початковий розподіл температурних полів $T_{ij}(x, 0)$, і законів тепла $Q_i(t)$ та масове перенесення $G_{ij}(t)$. Результати такої інтеграції будуть функцією $T_i(t)$, тобто зміною температури повітря.

2.3 Прогнозування споживання розумного будинку при налаштуванні мікроклімату

Прогнозування споживання розумного будинку при налаштуванні мікроклімату є складною багато параметричною задачею яка базується на теорії стохастичних процесів. Складність полягає у тому що на часовий ряд енергоспоживання (газу, електроенергії) накладаються добові, тижневі місячні і сезонні коливання, які можуть мати стаціонарний або не стаціонарний характер див.рис.2.1.

Для зручності можна провести класифікацію випадкових процесів та відповідних їм часових рядів на детерміновані та випадкові процеси (часові ряди). Детермінованим називають процес, який приймає вказане значення з ймовірністю одиниця. Наприклад, його значення можуть точно визначатися будь-якою математичною функцією від моменту часу t , як у наступному прикладі: $x_t = A \cos(2\pi ft - \theta)$. Коли ж ми будемо говорити про випадковий процес і випадковий часовий ряд, то, як правило, будемо передбачати, що він істотно випадковий, тобто не є детермінованим[12].

Стохастичні процеси поділяються на стаціонарні та нестаціонарні. Стохастичний процес є стаціонарним, якщо він знаходиться у певному сенсі в статистичній рівновазі, тобто його властивості з ймовірнісної точки зору не залежать від часу. Процес нестаціонарний, якщо ці умови порушуються.

Як правило, термін «часовий ряд» і сам по собі передбачає, що цей ряд є одномірним (скалярним). Часто важливо розглянути спільну динаміку набору часових рядків $x_t = x_{1t}, \dots, x_{kt}$. Такий набір називають багатовимірним часовим рядом або векторним часовим рядом. Відповідно, говорять також про багатовимірні (векторні) випадкові процесах.

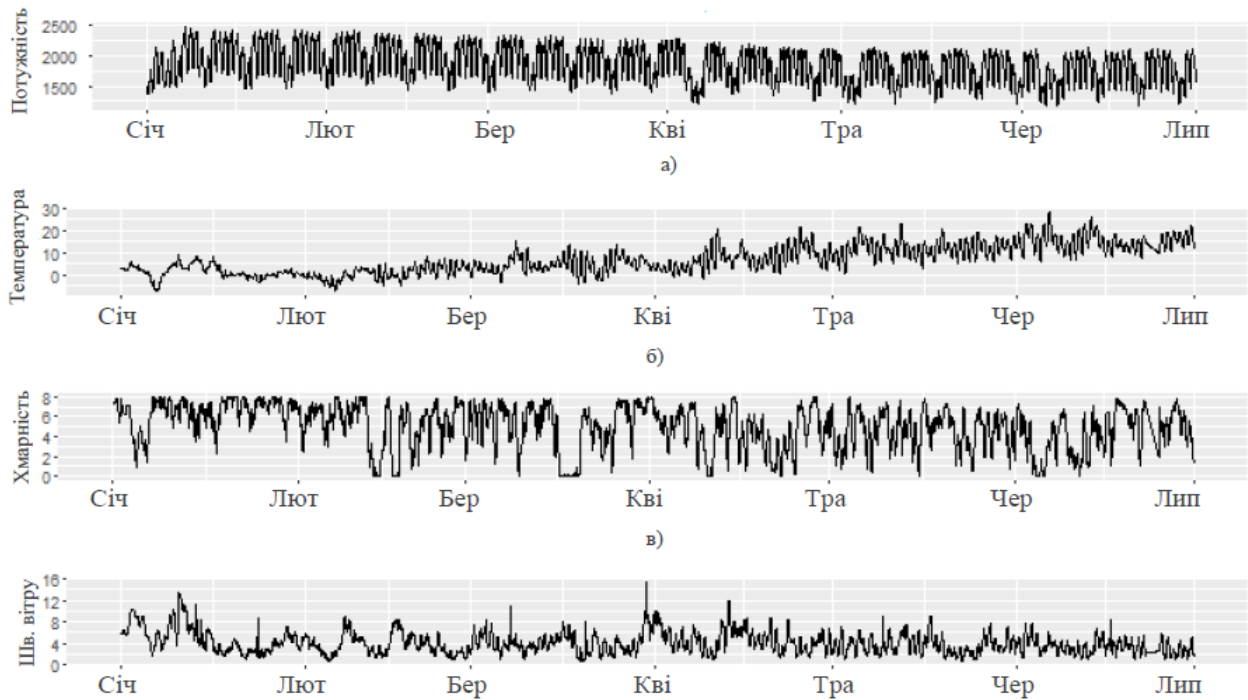


Рис.2.1. Приклад енергоспоживання розумного будинку

При аналізі споживання енергоресурсів:

- тенденція відповідає повільній зміні, що проходить в деякому певному напрямку, який зберігається протягом значного проміжку часу. Тенденцію називають також трендом або довготривалим рухом;
- циклічні коливання - це більш швидка, ніж тенденція, квазіперіодична динаміка, в якій є фаза зростання і фаза убутання. Найчастіше цикл пов'язаний з флуктуаціями економічної активності;
- сезонні коливання відповідають змінам, які відбуваються регулярно протягом року, тижня або доби. Вони пов'язані з сезонами і ритмами людської активності;

- календарні ефекти - це відхилення, пов'язані з певними передбачуваними календарними подіями, такими як святкові дні, кількість робочих днів за місяць, високосність року тощо;
- випадкові флуктуації - безладні рухи відносно великої частоти. Вони породжуються впливом різнорідних подій на досліджувану величину (несистематичний або випадковий ефект);
- викиди - це аномальні рухи часового ряду, пов'язані з нечастими подіями, які різко, але лише дуже короткочасно відхиляють ряд від загального закону, за яким він рухається;
- структурні зрушення - це аномальні рухи часового ряду, пов'язані з подіями, що не часто відбуваються, мають стрибкоподібний характер і змінюють тенденцію.

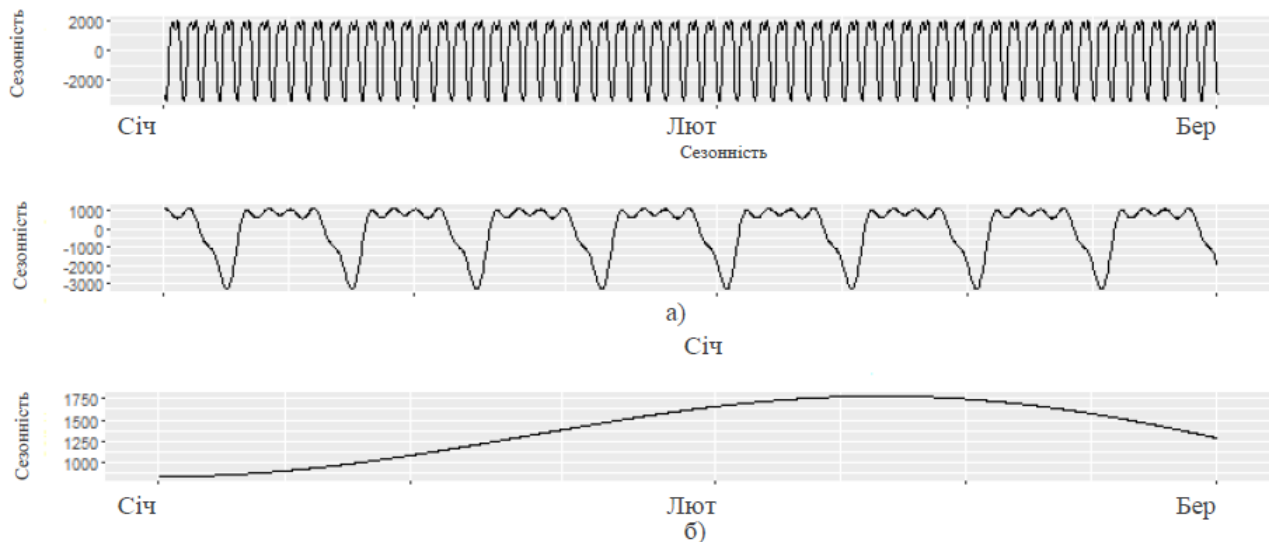


Рис.2.2. Сезонні складові в даних енергоспоживання, отримані на основі авторегресійної моделі (а – доба, б – тиждень, в- рік)

Ряди енергоспоживання можна вважати такими, що представляють ті або інші види таких рухів майже в чистому вигляді. Але велика частина їх має дуже складний вигляд. У них можуть проявлятися, наприклад, як загальна тенденція зростання, так і деякі сезонні зміни, на які можуть накладатися випадкові

флуктуації. Часто для аналізу рядків часу виявляється корисним ізольований розгляд окремих компонент.

2.3.1 Основні описові статистики для часових рядків

Припустимо, у нас є деякі дані (часовий ряд) $x = x_1, \dots, x_T$. Середнє і дисперсія часового ряду розраховуються за звичайними формулами:

$$\bar{x} = \sum_{t=1}^T x_t, \quad (2.11)$$

$$s^2 = \frac{1}{T} \sum_{t=1}^T (x_t - \bar{x})^2, \quad (2.12)$$

Вибіркова автоковаріація k -го порядку обчислюється як

$$c_k = \frac{1}{T} \sum_{t=1}^{T-k} (x_t - \bar{x})(x_{t+k} - \bar{x}), \quad (2.13)$$

Якщо часовий ряд слабо стаціонарний, то ці описові статистики є аналогом відповідних теоретичних величин і при деяких припущеннях будуть їх заможними оцінками.

В теорії часових рядків при розрахунку дисперсії та підковаріацій прийнято ділити на T . Замість цього при розрахунку дисперсії, наприклад, можна було б ділити на $T-1$, що дало б незміщену оцінку, а при розрахунку підступності k -го порядку – на $T-k$ за кількістю доданих. Виправданням даної формули може служити простота розрахунків і те, що в такому вигляді цей вираз гарантує позитивну невизначеність матриці вибірових автоковаріацій C_T :

$$C_T = \begin{pmatrix} c_0 & c_1 & \cdots & c_{T-1} \\ c_1 & c_0 & \cdots & c_{T-2} \\ \vdots & \vdots & \ddots & \vdots \\ c_{T-1} & c_{T-2} & \cdots & c_0 \end{pmatrix}. \quad (2.14)$$

Це відображає важливу властивість відповідної матриці Γ_T справжніх автоковаріацій. Позитивно невизначеною є будь-яка матриця B , яку можна уявити у вигляді $B = A'A$, де A - деяка матриця.

У нашому випадку $A = \frac{1}{\sqrt{T}}\hat{X}$, оскільки можна уявити матрицю C_T таким чином:

$$C_T = \frac{1}{T}\hat{X}'\hat{X}, \quad (2.15)$$

де \hat{X} – матриця, складена з центрованих значень ряду $\hat{x}_t = x_t - \bar{x}$:

$$\hat{X} = \begin{pmatrix} \hat{x}_1 & 0 & \cdots & 0 \\ \hat{x}_2 & \hat{x}_1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \hat{x}_T & \hat{x}_{T-1} & \cdots & \hat{x}_1 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \hat{x}_T \end{pmatrix}.$$

Статистичною оцінкою автокореляції k -го порядку є вибірковий коефіцієнт автокореляції:

$$r_k = \frac{c_k}{c_0}. \quad (2.17)$$

При аналізі зміни величин c_k і r_k , залежно від значення k , зазвичай користуються вибірковими автоковаріаційною та автокореляційною функціями[12], що визначаються як послідовності $\{c_k\}$ і $\{r_k\}$ відповідно.

Особливу роль в аналізі стаціонарних часових рядів відіграє вибіркова автокореляційна функція, яка використовується як інструмент для розпізнавання типу процесу. Оскільки автокореляційна функція симетрична відносно

нульового значення ($r_k = r_{-k}$), на графіку зображують тільки праву половину функції (для $k > 0$).

Вибіркова крос-коваріація двох часових рядків, x_t і y_t , розраховується за формулою

$$\frac{1}{T} \sum_{t=1}^{T-k} (x_{t+k} - \bar{x})(y_t - \bar{y}). \quad (2.18)$$

Вона характеризує взаємозв'язки двох рядів у часі, з різною величиною зсуву k . Слід пам'ятати, що на відміну від автоковаріації, кросковаріація не є симетричною по k , тому її слід розглядати і при додатних, і при негативних k .

Вибіркова крос-кореляція визначається як

$$\frac{\sum_{t=1}^{T-k} (x_{t+k} - \bar{x})(y_t - \bar{y})}{\sqrt{\sum_{t=1}^T (x_t - \bar{x})^2 \sum_{t=1}^T (y_t - \bar{y})^2}}. \quad (2.19)$$

2.4 Моделювання енергоспоживання

Порівняно простою моделлю часового ряду може служити модель виду:

$$x_t = \mu_t + \varepsilon_t, t = 1, 2, \dots, T. \quad (2.20)$$

Тут часовий ряд буде розглядатися як сума повністю детермінованої послідовності $\{\mu_t\}$, яку можна назвати систематичною складовою, і випадкової послідовності $\{\varepsilon_t\}$, що є білим шумом.

Якщо μ_t залежить від вектора невідомих параметрів θ :

$$\mu_t = \mu_t(\theta), \quad (2.21)$$

то це модель регресії, і параметри можна оцінити методом найменших квадратів (МНК). Детермінована компонента μ_t , як правило, сама моделюється як складається з декількох компонентів.

Цей прийом дозволяє залишатися в рамках лінійної регресії і значно спрощує моделювання.

2.4.1 Тренд енергоспоживання розумного будинку

Вивчаючи тренди, слід мати на увазі, що існує, взагалі кажучи, кілька їх різновидів.

Першим і самим очевидним типом тренда представляється тренд середнього, коли часовий ряд виглядає як коливання близько повільно зростаючої або спадної величини.

Другий тип трендів - це тренд дисперсії. У цьому випадку в часі змінюється амплітуда коливань змінної. Іншими словами, процес гетероскедастичний. Часто економічні процеси з зростаючим середнім мають і зростаючу дисперсію.

Третій і більш тонкий тип тренду, візуально не завжди спостерігається, - зміна значущості однієї з компонент часового ряду (наприклад, зменшення величини сезонних коливань), або, скажімо, зміна величини кореляції між поточним і попереднім значеннями ряду, тобто тренд автоковаріації та автокореляції.

Проводячи розкладання ряду на компоненти, ми, як правило, маємо на увазі під трендом зміна середнього рівня змінної.

Прийнято виділяти чотири основні способи апроксимації часових рядів і відповідно чотири види трендів.

Поліноміальний тренд:

$$\tau_t = a_0 + a_1 t + \dots + a_p t^p. \quad (2.25)$$

Експоненційний тренд:

$$\tau_t = e^{a_0 + a_1 t + \dots + a_p t^p}. \quad (2.26)$$

Гармонійний тренд:

$$\tau_t = A \cos(2\pi f t + \varphi). \quad (2.27)$$

Тренд, що виражається логістичною функцією:

$$\tau_t = \frac{k}{1 + b e^{-at}}. \quad (2.28)$$

Оцінювання параметрів поліноміального та експоненційного трендів не представляє складнощі. Після заміни змінних у першому випадку та логарифмування функції у другому випадку використовується звичайний метод найменших квадратів.

2.4.2 Сезонні коливання

Для моделювання сезонної складової s_t можна використовувати формулу:

$$\vartheta_t = \lambda_1 \delta_{1t} + \dots + \lambda_h \delta_{ht}, \quad (2.30)$$

де δ_{jt} - сезонні фіктивні змінні, що відповідають s сезонам. Ці змінні будуються таким чином: $\delta_{jt} = 1$, коли спостереження відноситься до сезону j , і $\delta_{jt} = 0$ в іншому випадку. Використання в лінійній регресії повного набору таких змінних пов'язано з однією особливістю. У сумі вони дають одиницю:

$$\delta_{1t} + \dots + \delta_{ht} = 1. \quad (2.31)$$

Тому, коли скоро в регресії є константа, то буде мати місце лінійна залежність, і $\lambda_1, \dots, \lambda_h$ не можна буде оцінити однозначно. Таким чином, потрібно

накласти на коефіцієнти $\lambda_1, \dots, \lambda_h$ яке-небудь нормуюче обмеження. Зокрема, можна покласти один з коефіцієнтів рівним нулю, що еквівалентно невикористанню відповідною змінною при побудові регресії. Однак більш вдале нормування складається у тому, щоб покласти

$$\lambda_1 + \dots + \lambda_h = 0. \quad (2.32)$$

При цьому сезонна компонента як би центрується, тобто в середньому вплив ефекту сезонності на рівень ряду виявляється рівним нулю. Підставимо це обмеження в сезонну компоненту, виключивши коефіцієнт λ_1 :

$$-(\lambda_2 + \dots + \lambda_h)\delta_{1t} + \lambda_2\delta_{2t} + \dots + \lambda_h\delta_{ht} = \lambda_2(\delta_{2t} - \delta_{1t}) + \dots + \lambda_h(\delta_{ht} - \delta_{1t}). \quad (2.33)$$

Нові змінні $\delta_{2t} - \delta_{1t}, \dots, \delta_{ht} - \delta_{1t}$ будуть вже лінійно незалежними і їх можна використовувати в лінійній регресії в якості факторів. При потребі, оцінку λ_1 можна отримати з оцінок інших коефіцієнтів за формулою $\lambda_1 = -(\lambda_2 + \dots + \lambda_h)$. Це дасть оцінку структури сезонності $\lambda_1, \dots, \lambda_h$. Трактувати її слід так: в j -му сезоні сезонність призводить до відхилення від основної динаміки ряду на величину λ_j .

2.4.3 Аномальні події енергоспоживання розумного будинку

При моделюванні часового ряду часто відкидаються аномальні спостереження, що різко відхиляються від напрямку еволюції ряду. Такого роду викиди, замість винятку, можна моделювати за допомогою фіктивних змінних, що відповідають фіксованим моментам часу. Припустимо, що в момент t^* в економіці відбулася якась важлива подія (наприклад, відбулася відставка уряду). Тоді можна побудувати фіктивну змінну $\delta_t^{t^*}$, яка дорівнює нулю завжди, крім моменту $t = t^*$, коли вона дорівнює одиниці: $\delta_t^{t^*} = (0, \dots, 0, 1, 0, \dots, 0)$.

Описана фіктивна змінна придатна тільки для моделювання короткочасного відхилення часового ряду. Якщо ж в економіці стався структурне зрушення, що викликало стрибок в динаміці ряду, то слід використовувати фіктивну змінну іншого виду: $(0, \dots, 0, 1, \dots, 0)$. Ця змінна дорівнює нулю до деякого фіксованого моменту t^* , а після цього моменту стає рівною одиниці.

Зауважимо, що останні два види змінних не можна використовувати для прогнозування, оскільки вони відносяться до одиничних непрогнозованих подій.

Передбачення енергоспоживання розумного будинку в залежності від заданих параметрів мікроклімату в будинку є складною задачею, оскільки, вона має стохастичний і коливний характер і при різній температурі, вологості освітленості і вентиляції будуть накладатись свої параметри авто регресії на параметри добових, тижневих та інших коливань енергоспоживання розумного будинку.

2.5 Висновки до розділу 2

Багатоконтурна система керування параметрами мікроклімату потребує задання оптимальних його параметрів, які впливають на контури управління температурою, вологістю, вентиляцією та кондиціонування. Ці параметри мають бути враховані при створенні користувацького інтерфейсу з можливістю створення прогнозу при кожному варіанті налаштувань і донесенню його до користувача у зручній формі.

Передбачення енергоспоживання розумного будинку в залежності від заданих параметрів мікроклімату в будинку є складною задачею, оскільки, вона має стохастичний і коливний характер і при різній температурі, вологості освітленості і вентиляції будуть накладатись свої параметри авто регресії на параметри добових, тижневих та інших коливань енергоспоживання розумного будинку.

РОЗДІЛ 3

ПРОГРАМНА РЕАЛІЗАЦІЯ СИСТЕМИ МОНІТОРИНГУ МІКРОКЛІМАТУ РОЗУМНОГО БУДИНКУ

3.1 Методи та засоби реалізації застосунку

Програмний комплекс побудований за принципами сервіс орієнтованої архітектури побудови програм. Кожен сервіс цієї архітектури реалізовано з використанням різних технологій.

Для клієнтського рівня було використано такий набір технологій: мова програмування JavaScript, компонента Razor Pages.

Для серверного рівня було використано такі технології: мова програмування C# з використанням фреймворка Asp .Net core MVC, веб-сервер IIS.

Для рівня бази даних було обрано MSSQL.

3.1.1 .NET core

.NET Core - це нова версія .NET Framework, яка є безкоштовною платформою розробки з відкритим вихідним кодом[13], підтримуваною Microsoft. Це крос-платформний фреймворк, який працює на операційних системах Windows, macOS і Linux.

.NET Core може бути використаний для створення різних типів додатків, таких як мобільні, веб, хмарні, інтернету речей, машинне навчання, мікросервіси, ігри тощо.

.NET Core написаний з нуля, щоб зробити його модульним, легким, швидким і крос-платформним фреймворком. Він включає в себе основні функції, які необхідні для запуску базового додатка .NET Core. Інші функції надаються як пакети NuGet, які можна додати у додаток в міру необхідності. Таким чином, додаток .NET Core прискорює продуктивність, зменшує слід пам'яті і стає простим в обслуговуванні.

Основні характеристики[13]:

- фреймворк з відкритим вихідним кодом, що підтримується Microsoft і доступний на GitHub під ліцензіями MIT і Apache 2;
- кросплатформність – працює на операційних системах Windows, macOS і Linux. Є різні терміни виконання для кожної операційної системи, яка виконує код і генерує один і той же вихід;
- широкий спектр застосунків – можливість розробляти та запускати різні типи додатків (мобільні, веб, хмарні, машинне навчання, ігри тощо);
- підтримка декількох мов – можливість використовувати мови програмування C#, F# та Visual Basic;
- модульна архітектура – .NET Core підтримує підхід модульної архітектури з використанням пакетів NuGet. Існують різні пакети NuGet для різних функцій, які можуть бути додані до проекту .NET Core при необхідності;
- CLI інструменти – включає інструменти CLI (інтерфейс командного рядка) для розробки та безперервної інтеграції;
- гнучке розгортання – застосунок може бути розгорнутий в широкій або загальній системі або за допомогою Docker Containers;
- сумісність з .NET Framework і Mono API за допомогою специфікації .NET Standard.

3.1.2. C#

C # – це сильно типізована об'єктно-орієнтована мова програмування з відкритим вихідним кодом, простим, сучасним, гнучким та універсальним[14]. Вона забезпечує сучасним розробникам гнучкість та можливості для створення програмного забезпечення.

Переваги C#[15]:

- не дозволяє перетворення типів, що може призвести до втрати даних або інших проблем. C # дозволяє розробникам писати безпечний та ефективний код;

- не потрібно приділяти багато уваги таким проблемам, як втрата пам'яті;
- концепція збірки добре вирішує питання контролю версій;
- простота в розробці (багата бібліотека класів полегшує реалізацію багатьох функцій);
- кросплатформність (створення додатків .NET, які можна розгорнути на платформах Windows, Linux та Mac. Додатки C # також можна розгорнути у хмарі);
- підтримка розподіленої системи.

3.2.3 UDP Protocol

User Datagram Protocol, або UDP, - це протокол зв'язку[18], який використовується в Інтернеті для особливо чутливих до часу передач, таких як відтворення відео або пошук DNS. Це прискорює зв'язок, формально не встановлюючи з'єднання до передачі даних. Це дозволяє передавати дані дуже швидко, але це також може призвести до втрати пакетів в транзиті - і створити можливість для експлуатації у вигляді DDoS-атак.

Як і всі мережеві протоколи, UDP є стандартизованим методом передачі даних між двома комп'ютерами в мережі. У порівнянні з іншими протоколами, UDP здійснює цей процес простим способом: він відправляє пакети (одиниці передачі даних) безпосередньо на цільовий комп'ютер, не встановлюючи спочатку з'єднання, вказуючи порядок зазначених пакетів, або перевіряючи, чи прибули вони за призначенням. (Пакети UDP називаються «датаграмами».)

UDP швидший, але менш надійний, ніж TCP, інший загальний транспортний протокол. У зв'язку з TCP два комп'ютери починають з'єднання за допомогою автоматизованого процесу, який називається «рукоштовання».

Комунікації UDP не проходять через цей процес. Замість цього один комп'ютер може просто почати надсилати дані іншому.

Крім того, у повідомленнях TCP вказується порядок отримання пакетів даних та підтвердження надходження пакетів за призначенням. Якщо пакет не

надходить - наприклад, через перевантаження в мережах посередників - TCP вимагає повторної відправки. Комунікація UDP не включає жодного з цих функцій.

Ці відмінності створюють деякі переваги. Оскільки UDP не вимагає «рукостискання» або перевірки належного надходження даних, він здатний передавати дані набагато швидше, ніж TCP.

Однак така швидкість створює компроміси. Якщо datagram UDP втрачено в транзиті, він не буде повторно відправлений. В результаті, програми, які використовують UDP, повинні мати можливість переносити помилки, втрати та дублювання.

Технічно, такі втрати пакетів є менш недоліком UDP, ніж наслідком того, як побудований Інтернет. Більшість мережевих маршрутизаторів не виконують замовлення пакетів і підтвердження прибуття за дизайном[19], тому що це вимагає нездійсненної кількості додаткової пам'яті. TCP - це спосіб заповнити цю прогалину, коли цього вимагає програма.

3.2 Опис розробленого застосунку

Для зберігання даних користувачів було створено базу даних за допомогою MSSQL.

Оскільки реальних показників датчиків температури, вологості, та освітленості немає, то для функціонування додатку всі дані було згенеровано (рис.3.1).

```

0 references
static void Main(string[] args)
{
    var client = new UdpClientManager(Hostname, Port);
    Task.Factory.StartNew(async () => {
        while (true)
        {
            try
            {
                var measurement = new Measurement
                {
                    MeasurementTypeId = 1,
                    RoomId = 1,
                    Timestamp = DateTimeOffset.Now,
                    UnitOfMeasurementId = 1,
                    UserId = UserID,
                    Value = random.NextDouble(),
                };
                var json = JsonConvert.SerializeObject(measurement);
                client.Send(json);
            }
            catch { }
        }
    });
}

```

Рис.3.1. Генерування даних бази даних

На рис.3.2 показано схему зв'язків між таблицями в базі даних.

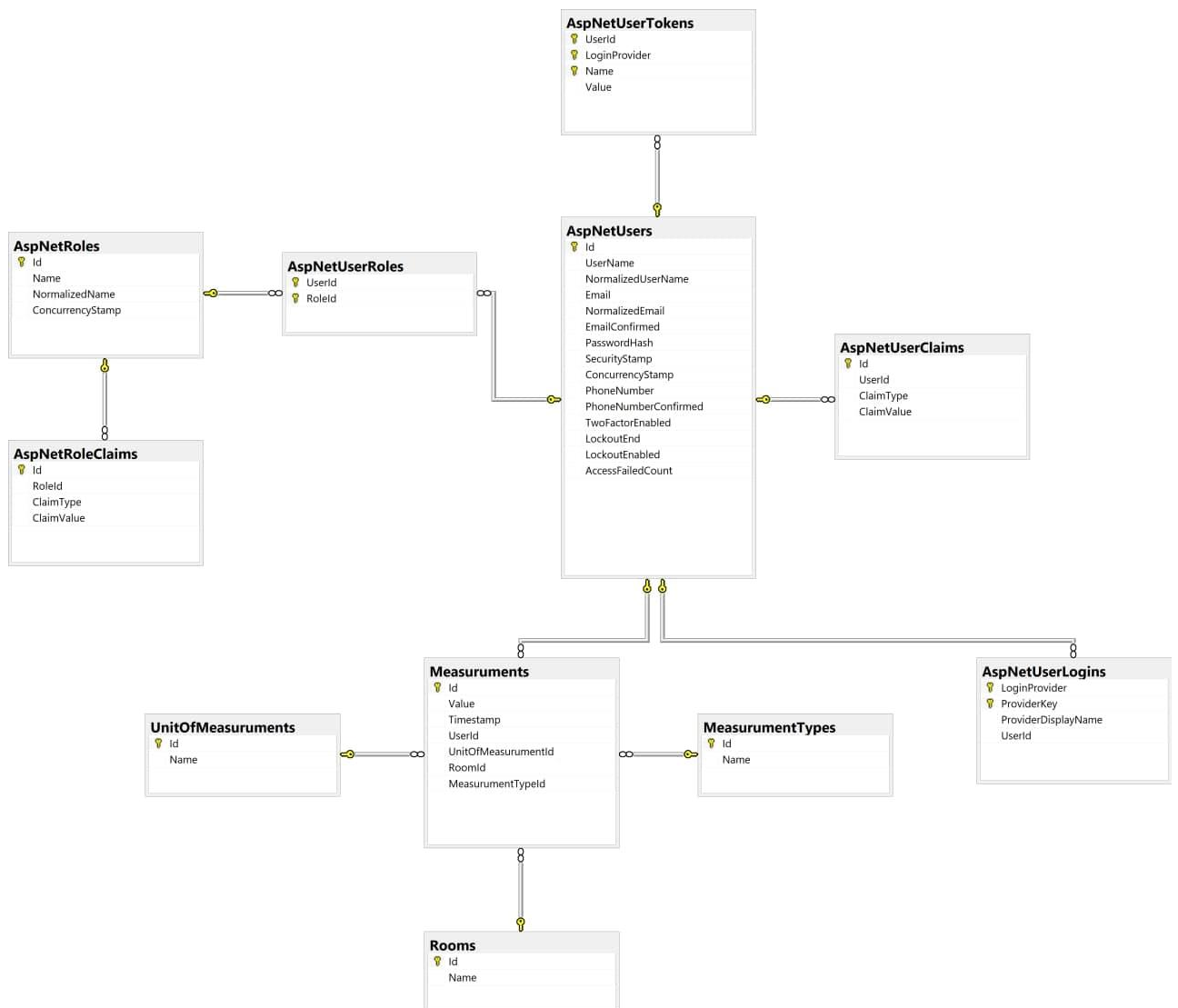


Рис.3.2. Схема зв'язків між таблицями в базі даних

Для оперування базою даних використовується інтерфейс `ApplicationDbContext` [16], який синхронізується з `MSSQL Server` за допомогою технології `Entity Framework` [17]. Лістинг коду для оперування базою даних зображений на рис.3.3.

```

4  using Microsoft.Extensions.Configuration;
5  using System;
6  using System.Collections.Generic;
7  using System.Text;
8
9  namespace CreativeHouse.Data
10 {
11     16 references
12     public class ApplicationDbContext : IdentityDbContext
13     {
14         4 references
15         public DbSet<UnitOfMeasurement> UnitOfMeasurements { get; set; }
16         9 references
17         public DbSet<Measurement> Measurements { get; set; }
18         11 references
19         public DbSet<MeasurementType> MeasurementTypes { get; set; }
20         11 references
21         public DbSet<Room> Rooms { get; set; }
22
23         0 references
24         public ApplicationDbContext(DbContextOptions<ApplicationDbContext> options)
25             : base(options)
26         {
27         }
28
29         0 references
30         protected override void OnConfiguring(DbContextOptionsBuilder optionsBuilder)
31         {
32             base.OnConfiguring(optionsBuilder);
33             optionsBuilder.UseSqlServer("Server=(localdb)\\MSSQLLocalDB;Database=aspnet-CreativeHouse;Trusted_Connection=True;MultipleActiveRes

```

Рис.3.3. Лістинг коду програми для оперування базою даних

Структурна схема підключення декількох користувачів до сервера зображена на рис.3.4.

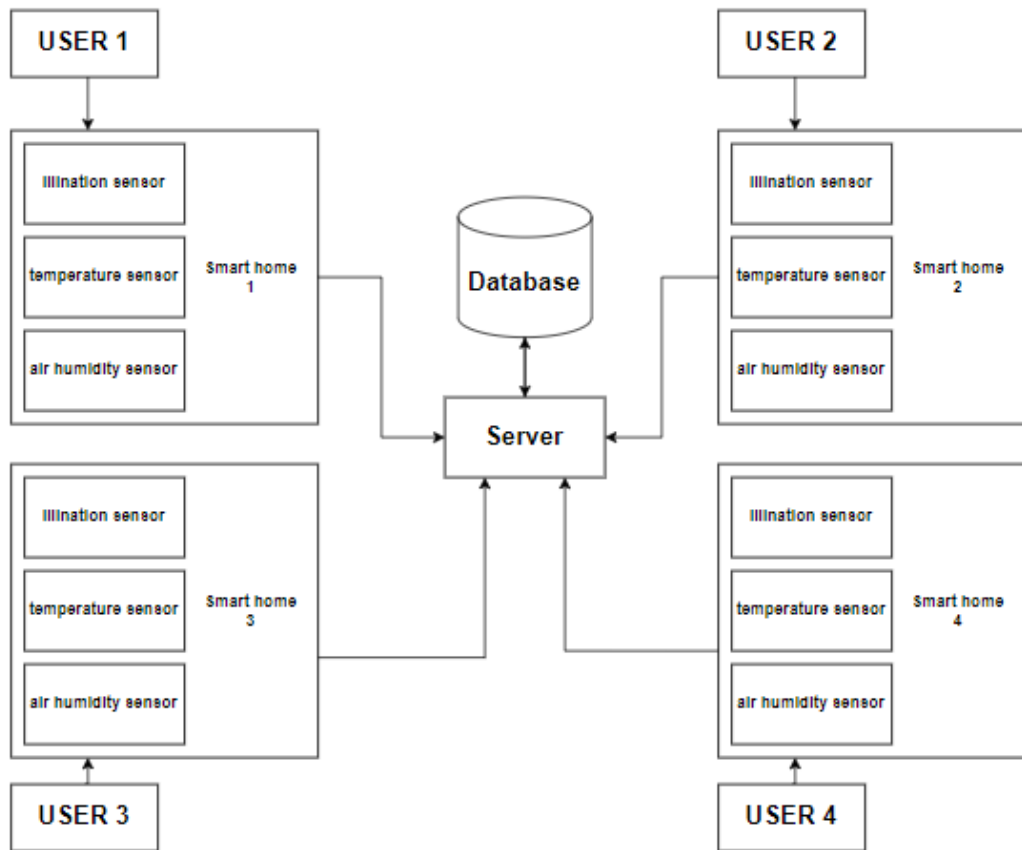


Рис.3.4. Схема підключення декількох користувачів

Для передачі даних на сервер використовується UDP Server. Оскільки це найпростіший протокол без встановлення з'єднання, він є ефективним, оскільки потрібна передача невеликої кількості інформації для багатьох користувачів. Модуль для підключення до UDP Server показано на рис.3.5.

```

namespace UDPClient
{
    2 references
    public class UdpClientManager
    {
        private readonly UdpClient _client;

        1 reference
        public UdpClientManager(string hostname, int port)
        {
            _client = new UdpClient();
            _client.Connect(hostname, port);
        }

        3 references
        public void Send(string message)
        {
            var datagram = Encoding.ASCII.GetBytes(message);
            _client.Send(datagram, datagram.Length);
        }
    }
}

```

Рис.3.5. Лістинг модуля для підключення до UDP Server

Для передачі результату на користувацький інтерфейс використовується метод, зображений на рис.3.6.

```
// GET: Measurements
3 references
public async Task<IActionResult> Index()
{
    var applicationDbContext = _context.Measurements
        .Include(m => m.MeasurementType)
        .Include(m => m.Room)
        .Include(m => m.UnitOfMeasurement)
        .Include(m => m.User);
    return View(await applicationDbContext.ToListAsync());
}
```

Рис.3.6. Лістинг коду передачі результатів на користувацький інтерфейс

Обчислення середніх значень вимірюваних параметрів по днях відображено на рис.3.7.

```
// GET: Measurements
0 references
public async Task<IActionResult> CalculateAverageMesurumentsPerDay(int typeId)
{
    var applicationDbContext = _context.Measurements.Where(x => x.MeasurementTypeId.Equals(typeId))
        .Include(m => m.MeasurementType)
        .Include(m => m.Room)
        .Include(m => m.UnitOfMeasurement)
        .Include(m => m.User)
        .GroupBy(x => x.Timestamp.Date).Select(x => new
    {
        Timestamp = x.Key,
        Average = x.Average(s => s.Value)
    });
    return View(await applicationDbContext.ToListAsync());
}
```

Рис.3.7. Лістинг коду обчислення середніх значень вимірювань по днях

Створення діаграми показників вологості повітря зображується на рис.3.8.


```

function showChart() {
  //myAmounts = myInvoices.AmountList;
  //myCategories = myInvoices.CategoryList;
  console.log(myAmounts);
  console.log(myCategories);
  let popCanvasName = document.getElementById("invChart");
  let barChartName = new Chart(popCanvasName, {
    type: 'bar',
    data: {
      labels: myCategories,
      datasets: [{
        label: 'Air Humidity',
        data: myAmounts,
        backgroundColor: [
          'rgba(255, 99, 132, 0.6)',
          'rgba(54, 162, 235, 0.6)',
          'rgba(255, 206, 86, 0.6)',
          'rgba(75, 192, 192, 0.6)',
          'rgba(153, 102, 255, 0.6)',
        ]
      }]
    },
    options: {
      responsive: false,
      scales: {
        yAxes: [{
          ticks: {
            beginAtZero: true
          }
        }]
      }
    }
  });
}

```

Рис.3.8. Лістинг коду створення діаграми показників вологості повітря

3.3 Опис інтерфейсу

В результаті виконаної роботи створено веб застосунок для моніторингу даних мікроклімату розумного будинку, який має такі функції:

- реєстрація/авторизація;
- керування кімнатами, що підключені до системи;
- керування параметрами відображення показників мікроклімату;
- відображення статистичних даних показників мікроклімату.

Першим етапом роботи з застосунком є реєстрація/авторизація[20]. На рис.3.9 зображено вигляд сторінки реєстрації.

CreativeHouse Home Register Login

Register

Create a new account.

Email

Password

Confirm password

Use another service to register.

There are no external authentication services configured. See [this article](#) for details on setting up this ASP.NET application to support logging in via external services.

© 2021 - CreativeHouse - [Privacy](#)

Рис.3.9. Сторінка реєстрації нового користувача

Для реєстрації нового користувача необхідно заповнити всі поля реєстрації, а саме електронну пошту, пароль та поле підтвердження пароля. Після чого необхідно підтвердити реєстрацію через лист, який буде надіслано на електронну пошту нового користувача.

Якщо користувач вже зареєстрований, то на сторінці входу (рис.3.10) потрібно заповнити поле електронної пошти та паролю і здійснити вхід.

CreativeHouse Home Register Login

Log in

Use a local account to log in. Use another service to log in.

Email

Password

Remember me?

[Log in](#)

[Forgot your password?](#)

[Register as a new user](#)

There are no external authentication services configured. See [this article](#) for details on setting up this ASP.NET application to support logging in via external services.

© 2021 - CreativeHouse - [Privacy](#)

Рис.3.10. Сторінка входу користувача

У нового користувача є можливість зберегти свої дані для входу, поставивши галочку у відповідному полі. Це пришвидшує та полегшує вхід в застосунок.

На сторінці користувача (рис.3.11) відображаються всі кімнати, підключені до системи. Користувач може додавати кімнати та видаляти, змінювати назву та переглядати дані мікроклімату.

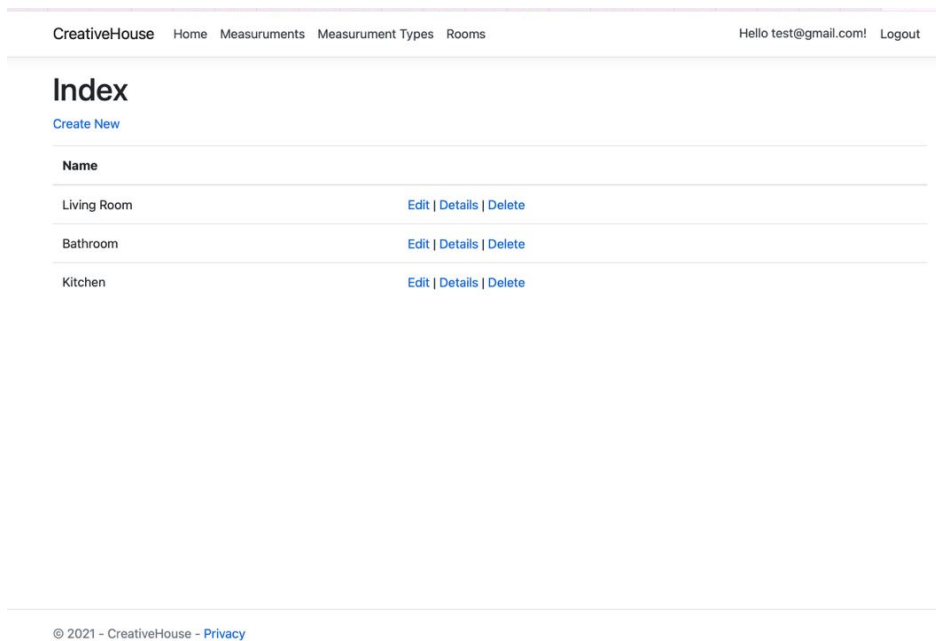


Рис.3.11. Сторінка користувача з відображенням підключених до системи кімнат

На сторінці типів вимірювання (рис.3.12) користувач має можливість змінити або видалити показники мікроклімату. Це дає змогу модифікувати застосунок та підлаштувати його до будь-якої системи моніторингу мікроклімату, а користувач в свою чергу має змогу видаляти непотрібні для себе показники.

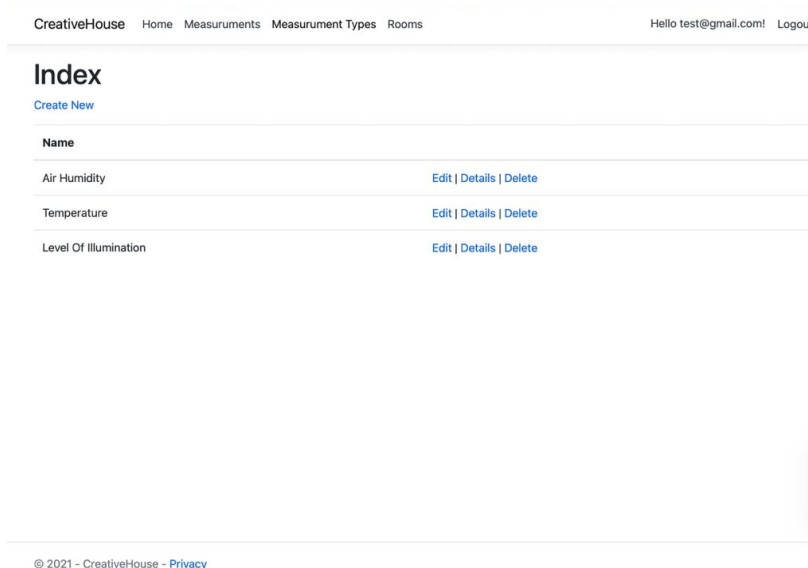


Рис.3.12. Сторінка типів вимірювань

Відображення даних мікроклімату представлено у вигляді графіків та діаграм. Для їх відслідковування користувачу потрібно вибрати необхідну кімнату з переліку. Після цієї дії відкриється сторінка з даними показників рівня освітленості, температури та вологості повітря.

Рівень вологості повітря відображено у вигляді діаграми(3.13). Кожна точка показує час та дату, коли показник було зафіксовано, та рівень вологості повітря у той момент часу.

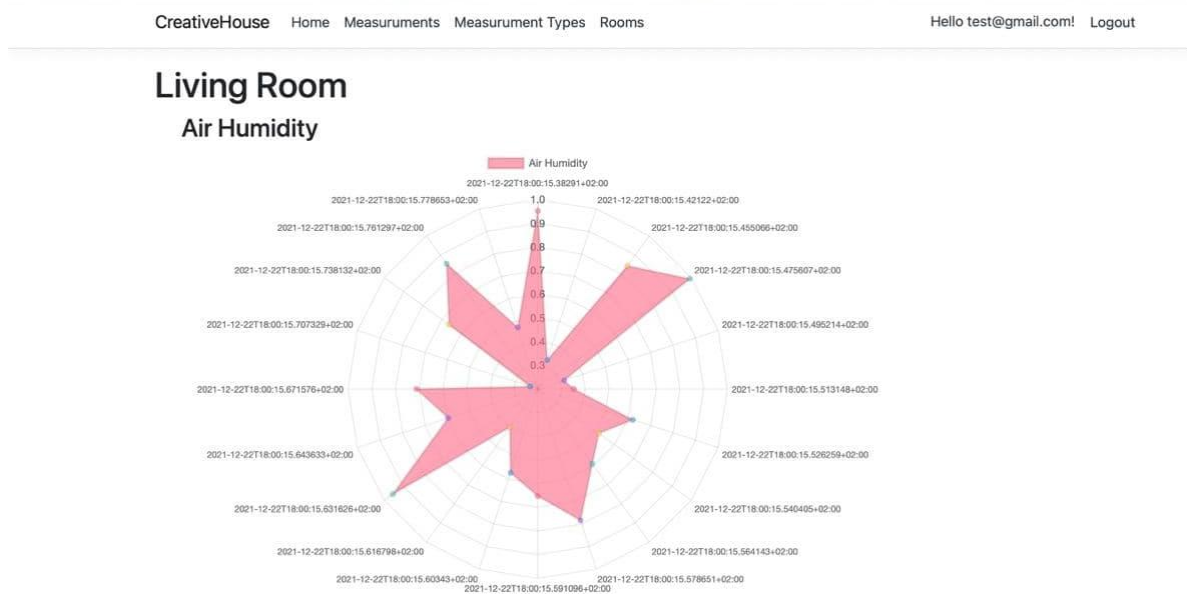


Рис.3.13. Діаграма рівня вологості повітря в кімнаті

Рис.3.14 відображає діаграму показників температури повітря в кімнаті. Вертикальна вісь показує температуру в градусах за Цельсієм. Горизонтальна вісь відображає час та дату фіксування температури.

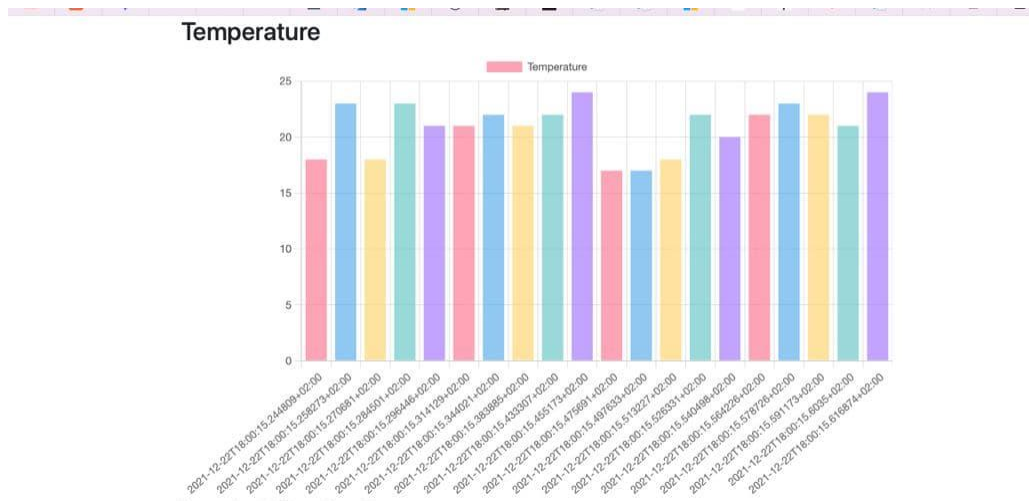


Рис.3.14. Діаграма рівня температури в кімнаті

Наступний графік (рис.3.15) дозволяє відслідковувати зміну рівня освітленості в кімнаті протягом певного часу. Це дозволить користувачеві визначити найоптимальніший час для навчання, роботи чи відпочинку, враховуючи кількість природнього освітлення.

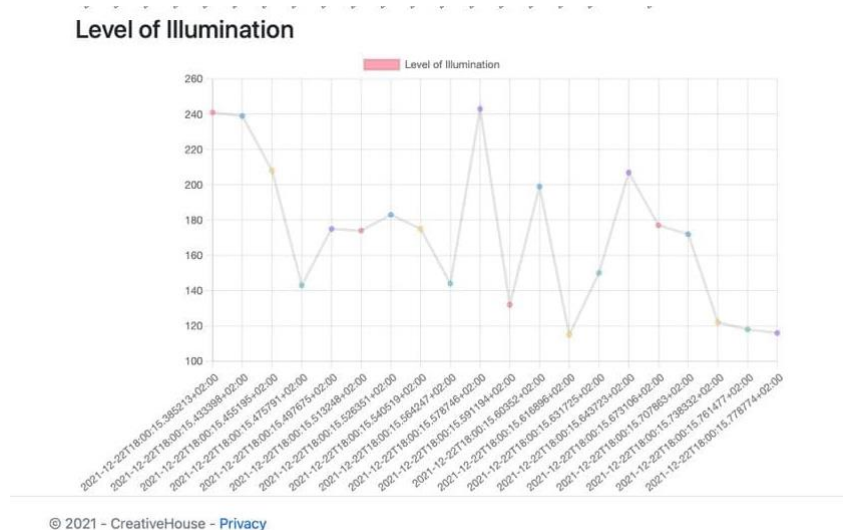


Рис.3.15. Графік рівня освітленості кімнати

Користувачеві також доступна функція видалення кімнат зі списку (рис.3.16). Завдяки цьому при переміщенні датчиків мікроклімату з однієї кімнати в іншу, користувач може створити нову кімнату в застосунку з новою назвою.

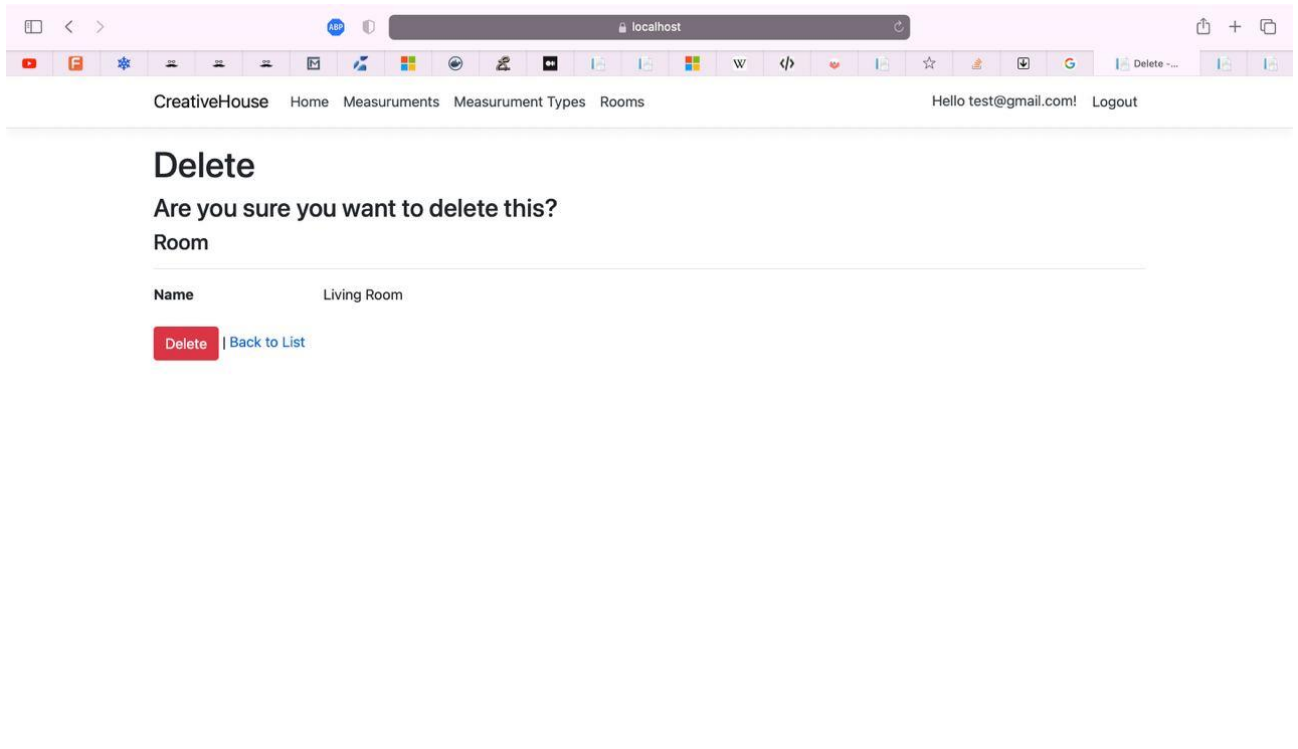


Рис.3.16. Сторінка видалення кімнати зі списку користувача

3.4 Висновки до розділу 3

Програмний комплекс відображення показників мікроклімату та споживання ресурсів «Розумного будинку» побудовано за принципами сервіс орієнтованої архітектури побудови програм. Кожен сервіс для оптимальної роботи цієї архітектури реалізовано з використанням різних адаптованих до цих задач технологій.

Клієнтський рівень було реалізовано за допомогою мови програмування JavaScript, та її компоненти Razor Pages.

Серверний рівень було реалізовано за допомогою таких технологій: мова програмування C# з використанням фреймворка Asp .Net core MVC, веб-сервер IIS.

Для ефективної роботи бази даних було обрано середовище MSSQL.

РОЗДІЛ 4

ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ

При дослідженні методів та засобів побудови комп'ютерної системи відображення показників мікроклімату та споживання ресурсів розумного будинку було дотримано усіх норм та правил з охорони праці та техніки безпеки. Враховуючи той факт, що виконання досліджень вимагає використання засобів обчислювальної техніки, то необхідним є створенням безпечних умов праці при використанні персональної електронно-обчислювальної машини (ПЕОМ). Також потрібно приділити особливу увагу питанням електро- та пожежної безпеки.

4.1 Охорона праці

З урахуванням вимог ДСанПіН 3.3.2.007-98[32] та НПАОП 0.00-7.15-18[33] необхідно визначити небезпечні і шкідливі фактори, які впливають на користувачів ВДТ (візуально дисплейний термінал) ПЕОМ при експлуатації, дослідити ці фактори, 90 розглянути їхній вплив, принципи їх нормування і способи запобігання їхнього шкідливого впливу на людину.

Основними шкідливими та небезпечними виробничими факторами, які пов'язані з використанням ПЕОМ, є такі:

- електромагнітне випромінювання радіочастотного діапазону;
- наявність іонізуючого рентгенівське випромінювання;
- випромінювання оптичного діапазону (ультрафіолетове, інфрачервоне і випромінювання видимого діапазону);
- електростатичне поле;
- недостатня освітленість робочої зони;
- підвищений рівень шуму;
- значна напруга зорових органів і пов'язане з цим перевтомлення користувача ПЕОМ;

- значне навантаження на пальці і кисті рук, що при відсутності профілактики і медичного контролю, може викликати професійні захворювання;
- тривале перебування в одному й тому ж самому положенні сидячи, що викликає застійні явища в організмі людини;
- відблиски на екрані монітора;
- можливість ураження електричним струмом;
- можливість виникнення пожежі.

Робоче приміщення з ПЕОМ по рівню безпеки ураження людей електричним струмом згідно ПУЕ можна віднести, до приміщень без підвищеної небезпеки, тому що:

- відносна вологість повітря не перевищує 75%;
- матеріал підлоги (паркет) є діелектриком;
- температура повітря не досягає значень, більших +30 °С;
- відсутня можливість одночасного дотику людини до з'єднаних з землею металоконструкцій будівлі, технологічних апаратів, механізмів і т.п., з одного боку, і до металевих корпусів електроустаткування - з іншого боку;
- відсутні хімічно агресивні середовища.

Для того, щоб не допустити ураження працівника електричним струмом при виникненні аварійних ситуацій, необхідно заземлити все обладнання, що працює від мережі 220 В, 50 Гц. Опір нульового дроту повинен бути таким, щоб при замиканні на корпус або нульовий дріт; виникав струм короткого замикання, сила якого повинна перевищувати в 1,4 рази номінальний струм спрацювання автомата струмового захисту (при струмі короткого замикання менше 100 А).

Тобто струм короткого замикання при виникненні аварійної ситуації в рази перевищує номінальний струм спрацювання автомата, що задовольняє встановленим нормам $K > 1,4$.

Облаштування робочих місць, обладнаних відеотерміналами, повинно забезпечувати:

- належні умови освітлення приміщення і робочого місця, відсутність відблисків;

- оптимальні параметри мікроклімату (температура, відносна вологість, швидкість руху та рівень іонізації повітря);
- належні ергономічні характеристики основних елементів робочого місця.

А також необхідно вжити заходи проти таких небезпечних і шкідливих чинників, як наявність шуму та вібрації, електромагнітного, ультрафіолетового і інфрачервоного випромінювання, електростатичного поля між екраном монітору і оператором, наявність пилу, озону, оксидів азоту й аероіонізації.

Будівлі і ті їх частини, в яких розташовуються ЕОМ, повинні мати не нижче II ступеня вогнестійкості.

Службові приміщення, в яких розташовані ПЕОМ, не повинні межувати з приміщеннями, де рівні шуму та вібрації перевищують норму (механічні цехи, майстерні тощо).

Відповідно до ДСанПіН 3.3.2.007-98 «Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин» площа приміщення, у яких розташовують відеотермінали, визначається, виходячи з розрахунку на одне робоче місце - не менше 6,0 кв. м, об'єм - не менше 20,0 куб. м, з урахуванням максимальної кількості осіб, які одночасно працюють у зміні.

Заземлені конструкції, що знаходяться в приміщеннях (батареї опалення, водопровідні труби, кабелі із заземленим відкритим екраном тощо), мають бути надійно захищені діелектричними щитками або сітками від випадкового дотику. У приміщеннях з ПЕОМ слід щоденно проводити вологе прибирання, повинні бути медичні аптечки першої допомоги, а також система автоматичної пожежної сигналізації.

Загальне освітлення має бути виконане у вигляді суцільних або переривчастих ліній світильників, що розміщуються збоку від робочих місць (переважно зліва) паралельно лінії зору працівників. Як джерело світла при штучному освітленні повинні застосовуватися, як правило, люмінесцентні лампи

типу ЛБ. Допускається у світильниках місцевого освітлення застосовувати лампи розжарювання.

Рівень освітленості на робочому столі в зоні розташування документів має бути в межах 300-500 лк. У разі неможливості забезпечити даний рівень освітленості системою загального освітлення допускається застосування світильників місцевого освітлення, але при цьому не повинно бути відблисків на поверхні екрана та збільшення освітленості екрана більше ніж до 300 лк.

При дотриманні цих вимог, виконання всіх видів робіт на ПЕОМ при дослідженні методів та засобів адаптивної селекції каналів зв'язку для розумного будинку, є безпечним з точки зору охорони праці, техніки безпеки та протипожежної безпеки.

4.2 Державна система моніторингу довкілля, як складова частина національної інформаційної інфраструктури, сумісної з аналогічними системами інших країн

Державна система моніторингу довкілля - це система спостережень, збирання, оброблення, передавання, збереження та аналізу інформації про стан довкілля, прогнозування його змін і розроблення науково-обґрунтованих рекомендацій для прийняття рішень про запобігання негативним змінам стану довкілля та дотримання вимог екологічної безпеки.

Система моніторингу є складовою частиною національної інформаційної інфраструктури, сумісної з аналогічними системами інших країн.

Система моніторингу - це відкрита інформаційна система, пріоритетами функціонування якої є захист життєво важливих екологічних інтересів людини і суспільства; збереження природних екосистем; відвернення кризових змін екологічного стану довкілля і запобігання надзвичайним екологічним ситуаціям.

Створення і функціонування системи моніторингу з метою інтеграції екологічних інформаційних систем, що охоплюють певні території, ґрунтується на принципах:

- узгодженості нормативно-правового та організаційно-методичного забезпечення, сумісності технічного, інформаційного і програмного забезпечення її складових частин;
- систематичності спостережень за станом довкілля та техногенними об'єктами, що впливають на нього;
- своєчасності отримання, комплексності оброблення та використання екологічної інформації, що надходить і зберігається в системі моніторингу;
- об'єктивності первинної, аналітичної і прогнозної екологічної інформації та оперативності її доведення до органів державної влади, органів місцевого самоврядування, громадських організацій, засобів масової інформації, населення України, заінтересованих міжнародних установ та світового співтовариства.

Відповідно до Положення про державну систему моніторингу довкілля (затверджене постановою Кабінету Міністрів України від 30.03.1998 №391) спостереження за станом об'єктів навколишнього природного середовища та рівнем його забруднення здійснюють 8 суб'єктів моніторингу довкілля - центральні органи виконавчої влади (Мінприроди, МНС, МОЗ, Мінагрополітики, Держжитлокомунгоспом, Держводгоспом, Держкомземом, Держкомлісгоспом) та їх органи на місцях, а також підприємства, установи та організації, що належать до сфери їх управління, які є суб'єктами системи моніторингу за загальнодержавною і регіональними (місцевими) програмами реалізації відповідних природоохоронних заходів.

Координацію діяльності суб'єктів системи моніторингу, розгляду поточних питань, пов'язаних з проведенням моніторингу довкілля, здійснює міжвідомча комісія з питань моніторингу довкілля із секціями за відповідними напрямками, склад та положення про яку затверджуються Кабінетом Міністрів України.

Система моніторингу спрямована на:

- підвищення рівня вивчення і знань про екологічний стан довкілля;

- підвищення оперативності та якості інформаційного обслуговування користувачів на всіх рівнях;
- підвищення якості обґрунтування природоохоронних заходів та ефективності їх здійснення;
- сприяння розвитку міжнародного співробітництва у галузі охорони довкілля, раціонального використання природних ресурсів та екологічної безпеки.

Основними завданнями суб'єктів системи моніторингу є:

- довгострокові систематичні спостереження за станом довкілля;
- аналіз екологічного стану довкілля та прогнозування його змін;
- інформаційно-аналітична підтримка прийняття рішень у галузі охорони довкілля, раціонального використання природних ресурсів та екологічної безпеки;
- інформаційне обслуговування органів державної влади, органів місцевого самоврядування, а також забезпечення екологічною інформацією населення країни і міжнародних організацій.

Система моніторингу ґрунтується на використанні існуючих організаційних структур суб'єктів моніторингу і функціонує на основі єдиного нормативного, організаційного, методологічного і метрологічного забезпечення, об'єднання складових частин та уніфікованих компонентів цієї системи.

Організаційна інтеграція суб'єктів системи моніторингу на всіх рівнях здійснюється органами Мінприроди на основі:

- загальнодержавної і регіональних (місцевих) програм моніторингу довкілля, що складаються з програм відповідних рівнів, поданих суб'єктами системи моніторингу;
- укладених між усіма суб'єктами системи моніторингу угод про спільну діяльність під час здійснення моніторингу довкілля на відповідному рівні.

4.3 Висновки до розділу 4

Виконання Україною міжнародних екологічних зобов'язань вимагає від національної системи моніторингу довкілля забезпечення достовірності й порівнюваності вимірювань в оцінках показників екологічної ситуації не лише по окремих регіонах України, але й на міжнародному рівні.

Тобто виникає необхідність переорієнтації всієї системи на новий комплекс первинних даних і на міжнародні екологічні стандарти. Державна система екологічного моніторингу прагне досягти таких основних цілей, що забезпечують вищу якість екологічного управління:

- підвищення рівня адекватності дійсному екологічному стану довкілля його інформаційної моделі, яка формується на основі даних систематичних спостережень, здійснюваних спеціальними службами міністерств і відомств, підприємствами, організаціями та установами в порядку виробничо-інформаційної діяльності, дослідних робіт і наукових досліджень;

- підвищення оперативності одержання та достовірності первинних даних за рахунок використання досконалих методик, сучасних контрольних-вимірювальних приладів і засобів комп'ютеризації процесів збирання, накопичення та оброблення екоінформації на всіх рівнях державного управління і місцевого самоврядування;

- підвищення рівня та якості інформаційного обслуговування споживачів екоінформації на всіх рівнях функціонування системи на основі мережного доступу до розподілених відомчих та інтегрованих банків даних, комплексного оброблення і використання інтегрованих банків даних, комплексного оброблення і використання інформації для прийняття відповідних рішень;

- підвищення якості обґрунтування природоохоронних заходів та ефективності їх здійснення; сприяння розвитку міжнародного співробітництва у галузі охорони довкілля, раціонального використання природних ресурсів та екологічної безпеки.

ВИСНОВКИ

В кваліфікаційній роботі магістра розроблено методи та засоби побудови комп'ютерної системи відображення показників мікроклімату та споживання ресурсів «Розумного будинку».

1. З аналізу існуючих систем керування та моніторингу розумних будинків було виявлено, що більша частина цих систем для керування та відображення використовує мобільні застосунки і лише невелика частина використовує веб-застосунки. Причому всі проаналізовані застосунки не виконують обчислень спожитих ресурсів електроенергії, газу чи води. Ця функція є дуже корисною, оскільки в будь-який момент часу користувач може перевірити значення витрачених ресурсів.

2. Багатоконтурна система керування параметрами мікроклімату потребує задання оптимальних його параметрів, які впливають на контури управління температурою, вологістю, вентиляцією та кондиціонування. Ці параметри мають бути враховані при створенні користувацького інтерфейсу з можливістю створення прогнозу при кожному варіанті налаштувань і донесенню його до користувача у зручній формі.

3. Передбачення енергоспоживання розумного будинку в залежності від заданих параметрів мікроклімату в будинку є складною задачею, оскільки, вона має стохастичний і коливний характер і при різній температурі, вологості освітленості і вентиляції будуть накладатись свої параметри авто регресії на параметри добових, тижневих та інших коливань енергоспоживання розумного будинку.

4. Програмний комплекс відображення показників мікроклімату та споживання ресурсів «Розумного будинку» побудовано за принципами сервіс орієнтованої архітектури побудови програм. Кожен сервіс, для оптимальної роботи цієї архітектури, було реалізовано з використанням різних адаптованих до цих задач технологій (Razor Pages, Asp .Net core MVC, веб-сервер IIS, MSSQL).

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Кращі системи "Розумний будинок" по виробниках 2021 року. ТОП 5 надійних та якісних систем "Розумний будинок" рейтингу. URL: <https://vencon.ua/ua/articles/rejting-sistem-umnyy-dom-po-proizvoditelyam> (дата звернення 16.12.2021).
2. REHAU NEA SMART 2.0 – управление микроклиматом через смартфон. URL: <https://aw-therm.com.ua/rehau-nea-smart-2-0-upravlenie-mikroklimate-cherез-smartfon/> (дата звернення 16.12.2021).
3. Обзор интеллектуальной системы мониторинга Steagle. URL: <https://www.bezpeka-shop.com/blog/obzor/obzor-intellektualnoy-sistemy-monitoringa-steagle/> (дата звернення 16.12.2021).
4. The versatile smart home solution. URL: <https://www.loxone.com/enen/smart-home/> (дата звернення 16.12.2021).
5. D. Kehagias, D. Nini. Home Automation Based on an Android and a Web Application Using Raspberry Pi. University of West Attica. 2015. С. 174-181.
6. A. Altayeva, B. Omarov, Y. Cho. Intelligent. Microclimate Control System Based on IoT. Gachon University. 2016. С.254-261.
7. J. Jaykumar, A. Blessy. Secure smart environment using IOT based on RFID. BNM Institute of Technology. 2014. С.2493-2496.
8. Клімат-контроль в розумному будинку. URL: <https://www.smarthouse.ua/ua/klimat-kontrol-v-umnom-dome.html> (дата звернення 16.12.2021).
9. D.M.G. Preethichandra. Design of a smart indoor air quality monitoring wireless sensor network for assisted living. Central Queensland University. 2013. С.1306-1309.
10. T. Yu and C. Lin, An intelligent wireless sensing and control system to improve indoor air quality: monitoring, prediction, and preaction. International Journal of Distributed Sensor Networks. 2015. С.144.

11. S. Abraham and X. Li. A cost-effective wireless sensor network system for indoor air quality monitoring applications. *Procedia Computer Science*. 2014.С.165-171.
12. В. И. Суслов, Н. М. Ибрагимов, Л. П. Талышева, А. А. Цыплаков. *Эконометрия*. Новосибирский государственный университет. 2005. С.315-329.
13. ASP.NET Core Tutorials. URL: <https://www.tutorialsteacher.com/core> (дата звернення 16.12.2021).
14. What is C#. URL:<https://www.c-sharpcorner.com/article/what-is-c-sharp/> (дата звернення 16.12.2021).
15. C Sharp – Features, Advantages and Disadvantages. URL: <https://urbannaturale.com/c-sharp-features-advantages-and-disadvantages/> (дата звернення 16.12.2021).
16. Основные классы в ASP.NET Core Identity. URL: <https://metanit.com/sharp/aspnet5/16.11.php> (дата звернення 16.12.2021).
17. Начало работы с EF Core. URL: <https://docs.microsoft.com/ru-ru/ef/core/get-started/overview/first-app?tabs=netcore-cli> (дата звернення 16.12.2021).
18. What is UDP? URL:<https://www.cloudflare.com/learning/ddos/glossary/user-datagram-protocol-udp/> (дата звернення 16.12.2021).
19. Лещин Ю. З., Романишин Н.Р., Наконечний В. В., Паламарчук А.О. Розробка системи зв'язку як інтегрованого елементу роботизованих систем // Зб. тез доповідей XXI Всеукр. наук.-пр. конф. Житомир, 2016. С. 102.
20. Марків В.А., Осухівська Г.М., Лещин Ю.З., Луцків А.М. Комп'ютерна система аутентифікації осіб // Матеріали XX наукової конференції ТНТУ ім. І. Пулюя. 2017. С. 90–91.
21. Leschyshyn Y., Scherbak L., Nazarevych O., Gotovych V., Tymkiv P., Shymchuk G. Multicomponent Model of the Heart Rate Variability Change-point // IEEE XVth International Conference on the Perspective Technologies and Methods in MEMS Design (MEMSTECH). 2019. P. 110–113.

Додаток А

Опубліковані тези конференцій за напрямом дипломної роботи магістра

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Тернопільський національний технічний університет імені Івана Пулюя (Україна)
Університет імені Г'єра і Марії Кюрі (Франція)
Маріборський університет (Словенія)
Технічний університет у Кошице (Словаччина)
Вільнюський технічний університет ім. Гедимінаса (Литва)
Білоруський національний технічний університет (Республіка Білорусь)
Міжнародний університет цивільної авіації (Марокко)
Наукове товариство ім. Т.Шевченка

АКТУАЛЬНІ ЗАДАЧІ СУЧАСНИХ ТЕХНОЛОГІЙ

Збірник
тез доповідей
Том I

**X Міжнародної науково-практичної
конференції молодих учених та студентів**
24-25 листопада 2021 року



УКРАЇНА
ТЕРНОПІЛЬ – 2021

УДК 001
А 43

Актуальні задачі сучасних технологій : зб. тез доповідей X міжнар. наук.-практ. конф. Молодих учених та студентів, (Тернопіль, 24–25 листоп. 2021.) / М-во освіти і науки України, Терн. націон. техн. ун-т ім. І. Пулюя [та ін.]. – Тернопіль: ФОП Паляниця В. А., 2021. Т. 1. – 155.

ISBN 978-617-7875-25-2

ПРОГРАМНИЙ КОМІТЕТ

Голова: Ясній Петро Володимирович – д.т.н., проф., ректор ТНТУ ім. І. Пулюя (Україна).

Заступник голови: Марущак Павло Орестович – д.т.н., проф. ТНТУ ім. І. Пулюя (Україна)

Вчений секретар: Маркович Ірина Богданівна – к.е.н., доц. ТНТУ ім. І. Пулюя (Україна)

Члени: Вухерер Т. – професор факультету інженерної механіки Маріборського університету (Словенія); Вінаш Я. – професор кафедри технології металів Технічного університету у Кошице (Словаччина); Прентковскіс О. – декан факультету Вільнюського технічного університету ім. Гедимінаса (Литва); Стахович Ф. – завідувач кафедри обробки матеріалів тиском Жешувського політехнічного університету ім. Лукасевича (Польща); Богданович А. – професор кафедри механіки Білоруського національного технічного університету (Республіка Білорусь); Менюу А. – д.т.н., професор Міжнародного університету цивільної авіації (Марокко); Андрейків О. – д.т.н., професор кафедри механіки Львівського національного університету ім. І. Франка, член-корр. НАН України.

Адреса оргкомітету:

ТНТУ ім. І. Пулюя, м. Тернопіль, вул. Руська, 56, 46001,

тел. **0506689327**, факс (0352) 255798

E-mail: konfmol@gmail.com

Редагування, оформлення, верстка: Мартиняк І.О.

СЕКЦІЇ КОНФЕРЕНЦІЇ, ЯКІ ПРЕДСТВЛЕНІ В ЗБІРНИКУ

- фізико-технічні основи розвитку нових технологій;
- нові матеріали, міцність і довговічність елементів конструкцій;
- сучасні технології в будівництві, машино- та приладобудуванні;
- сучасні технології на транспорті.
- компютерно-інформаційні технології та системи зв'язку

УДК 004.031.6: 004.056

Д.В. Кунинець, к.т.н.

Ю.З. Лещишин

Тернопільський національний технічний університет імені Івана Пулюя, Україна

ЗАСТОСУНОК ДЛЯ МОНІТОРИНГУ ДАНИХ РОЗУМНОГО БУДИНКУ

D. Kunynets, Ph.D. Yu. Z. Leshchyshyn

SMART HOME DATA MONITORING APPLICATION

З кожним роком в житті людей все більше з'являється автоматизації на рівні побуту. Інтернет речей вливається в повсякденне життя і разом з тим все більшу популярність набирають «розумні будинки». «Розумні будинки» оснащують сигналізацією, системами розпізнавання обличчя та голосу [1], автоматизованими вікнами та дверима, датчиками температури, вологості, освітленості та багатьма іншими [2]. В зв'язку з цим з'являється необхідність можливості керування та моніторингу даних цих підсистем максимально зручно для кінцевого користувача.

Для вирішення такої задачі необхідно проаналізувати існуючі засоби моніторингу даних розумного будинку, зокрема їх користувацький інтерфейс (front end) на рахунок ергономіки та зручності користування. А також статистичні методи обчислення спожитих ресурсів і т.п., тобто back end. Це надасть інформацію про функціональні можливості та зручність користування існуючих додатків та задасть напрям удосконалення або розробки нового застосунку.

Стосовно ергономіки користувацького інтерфейсу необхідно увагу приділити найбільш використовуваним вкладкам для відображення, якими є: поточний стан будинку, статистики витрат та налаштування і програмування режимів роботи обладнання «розумного будинку». Основні параметри затрат електроенергії, водопостачання, рівень освітленості, вологість повітря, температура приміщення необхідно відповідним чином поділити між цими вкладками. Втім актуальною залишається задача коректного та зручного відображення цих параметрів для користувача. Для її вирішення потрібно продумати структуру та дизайн застосунку. Він має бути зручний та зрозумілий для користувача, візуально лаконічний та з можливістю персоніфікації.

Стосовно статистичних методів обчислення спожитих ресурсів, тобто back end то для побудови застосунку необхідно використати такі статистичні методи, які відповідають сучасним математичним моделям процесів електро, газо та водоспоживання і враховують їх періодичність споживання та генерації (наприклад електроенергія від сонячних панелей). А також ці методи повинні давати прогноз витрати ресурсів для задач планування та оптимального використання (прання вдень чи вночі якщо є сонячні панелі і багато тарифний лічильник) або резервування (заряджання електричних акумуляторів, або нагрів теплового акумулятора теплої підлоги).

Загалом передбачити всі сценарії роботи «розумного будинку» дуже складно. Однак необхідність створення застосунку для керування «розумним будинком» потребує детального обґрунтування та вибору існуючих методів та засобів його реалізації. Що уможливить створення зручного у використанні та перспективного застосунку для моніторингу даних «розумного будинку».

Література:

1. В.А. Марків, Г.М. Осухівська, Ю.З. Лещишин, А.М. Луцків. Комп'ютерна система аутентифікації осіб // Матеріали XX наукової конференції ТНТУ ім. І. Пулюя. 2017. – С. 90–91.
2. Лещишин Ю.З., М.В. Павлюк. Проектування системи контролю та управління температурними режимами «розумного будинку» // Збірник тез доповідей ІХ Міжнародної науково-технічної конференції молодих учених та студентів „Актуальні задачі сучасних технологій“. ТНТУ 2020. С. 60–61.

Матеріали X Міжнародної науково-практичної конференції молодих учених та студентів
«АКТУАЛЬНІ ЗАДАЧІ СУЧАСНИХ ТЕХНОЛОГІЙ» – Тернопіль 24-25 листопада 2021 року

- | | | |
|-----|--|----|
| 9. | Ю.І. Пиндус, В.П.Калушка, Р.Р. Заверуха, О.Ю. Пиндус, Ю.І. Пинко | 77 |
| | ДОСЛІДЖЕННЯ ТЕПЛООВОГО БАЛАНСУ РОБОТИ ДВИГУНА НА
ДИЗЕЛЬНОМУ ПАЛИВІ ТА БІОПАЛИВІ | |
| 10. | Р.М. Рогатинський, В.Л. Дмитроца, М.В. Грубенюк, Р.П. Цапик | 79 |
| | ТРАНСПОРТУВАННЯ НАСИПНОГО ПАЛИВА ГВИНТОВИМИ
КОНВЕЄРАМИ | |
| 11. | Р.М. Рогатинський, Р. В. Хорошун, А.Д. Бобков, Р.Б. Шимків | 81 |
| | МОДЕЛЮВАННЯ РУХУ АВТОМОБІЛЯ ПО КРИВОЛІНІЙНІЙ ТРАСІ | |
| 12. | В.В. Ткачук, Ю.С. Шуберт | 83 |
| | ІМІТАЦІЙНЕ МОДЕЛЮВАННЯ РОБОТИ ЛОГІСТИЧНОГО СКЛАДУ | |

СЕКЦІЯ: КОМП'ЮТЕРНО-ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ ТА СИСТЕМИ ЗВ'ЯЗКУ

- | | | |
|-----|--|-----|
| 1. | О.В. Балакунець, Є.В. Тиш | 84 |
| | МЕТОДИ ТА ПРОГРАМНО-АПАРATНЕ ЗАБЕЗПЕЧЕННЯ СИСТЕМИ
РЕЗЕРВНОГО ЖИВЛЕННЯ В КОМП'ЮТЕРНИХ СИСТЕМАХ | |
| 2. | О.М. Барановський, А.В. Жилін, Г.С. Голич | 85 |
| | ЗАСТОСУВАННЯ МЕТОДУ МАШИННОГО НАВЧАННЯ ДЛЯ
ВИРІШЕННЯ ЗАДАЧІ ДЕТЕКТУВАННЯ ТОЧКОВИХ АНОМАЛІЙ У
МЕРЕЖЕВОМУ ТРАФІКУ ЗАСОБАМИ SIEM SPLUNK | |
| 3. | В.П. Волоський, Ю.З. Лещишин, Н.Р. Романишин | 87 |
| | КОМП'ЮТЕРНА СИСТЕМА КОНТРОЛЮ ТА БАЛАНСУВАННЯ ЛІТІЙ-
ІОННИХ АКУМУЛЯТОРНИХ БАТАРЕЙ | |
| 4. | А. В. Гайдар, В. А. Готович | 89 |
| | ЗАСТОСУВАННЯ КОМП'ЮТЕРНО-ІНФОРМАЦІЙНИХ ЗАСОБІВ В
ПРОЦЕСІ НАВЧАННЯ | |
| 5. | О.Р. Гончаренко, Є.В. Тиш | 90 |
| | СИСТЕМИ КЕРУВАННЯ СОНЯЧНИХ ТРЕКЕРІВ | |
| 6. | Р.О.Жаровський, Д.В.Дармопук | 91 |
| | ХАРАКТЕРИСТИКИ СИСТЕМ ЕЛЕКТРОННОГО НАВЧАННЯ | |
| 7. | С.А.Криськова | 92 |
| | ХАРАКТЕРИСТИКА ГЕОІНФОРМАЦІЙНИХ СИСТЕМ. ПОБУДОВА
ГЕОІНФОРМАЦІЙНОЇ КАРТИ «ВИДАТНІ УКРАЇНСЬКІ ВЧЕНІ» | |
| 8. | Д.В. Кунинець, Ю.З. Лещишин | 94 |
| | ЗАСТОСУНОК ДЛЯ МОНИТОРИНГУ ДАНИХ РОЗУМНОГО БУДИНКУ | |
| 9. | Р. М. Кучерешко | 95 |
| | СИСТЕМА ПІДТРИМКИ ПРИЙНЯТТЯ РІШЕНЬ НА ОСНОВІ НЕЧІТКОЇ
ЛОГІКИ ДЛЯ ВИЯВЛЕННЯ ШКІДЛИВИХ ПРОГРАМ | |
| 10. | А.Д. Лавренів, І.В. Бойко | 97 |
| | РОЗРОБКА МЕТОДІВ ДОСЛІДЖЕННЯ НЕЙРОННИХ МЕРЕЖ З
ВИКОРИСТАННЯМ СЕРЕДОВИЩА WOLFRAM MATHEMATICA ТА
МОВИ ПРОГРАМУВАННЯ C++ | |
| 11. | Я.Р. Лапшин | 99 |
| | АНАЛІЗ ЗАГРОЗ КІБЕРБЕЗПЕКИ ТА ДОСЛІДЖЕННЯ ЙОГО ВПЛИВУ | |
| 12. | Р.В. Ларіоник, Н.С. Луцик | 100 |
| | КОМП'ЮТЕРНА СИСТЕМА ДЛЯ ДИСТАНЦІЙНОГО КОНТРОЛЮ
ЯКОСТІ АТМОСФЕРНОГО ПОВІТРЯ | |
| 13. | Ю.З. Лещишин, З.В. Кузик | 101 |
| | МЕТОДИ ТА ЗАСОБИ АВТОМАТИЗОВАНОЇ РОЗРОБКИ ТЕХНІЧНОЇ
ДОКУМЕНТАЦІЇ МЕРЕЖЕВИХ КАБЕЛЬНИХ СИСТЕМ | |

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ТЕРНОПІЛЬСЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ
ІМЕНІ ІВАНА ПУЛЮЯ**

МАТЕРІАЛИ

ІХ НАУКОВО-ТЕХНІЧНОЇ КОНФЕРЕНЦІЇ

**«ІНФОРМАЦІЙНІ МОДЕЛІ,
СИСТЕМИ ТА ТЕХНОЛОГІЇ»**



8–9 грудня 2021 року

**ТЕРНОПІЛЬ
2021**

УДК 001
МЗ4

ПРОГРАМНИЙ КОМІТЕТ

Голова: Лупенко Сергій Анатолійович – докт. техн. наук, професор.

Співголови: Марущак Павло Орестовиц – проректор з наукової роботи, докт. техн. наук, професор.

Баран Ігор Олегович – канд. техн. наук, доцент, декан факультету ФІС.

Науковий секретар: Семенишин Галина Мирославівна – старший викладач.

Члени: докт. фіз.-мат. наук, професор В. Кривень; професор кафедри комп'ютерних наук, докт. техн. наук Я. Литвиненко; канд. техн. наук, доцент, Г. Осухівська; докт. техн. наук, професор М. Карпінський; канд. пед. наук, доцент Ж. Баб'як; докт. фіз.-мат. наук, професор М. Петрик; канд. техн. наук, доцент Н. Загородна.

ОРГАНІЗАЦІЙНИЙ КОМІТЕТ

Голова: Скоренький Юрій Любомирович – канд. техн. наук, доцент.

Члени: доцент кафедри комп'ютерних наук, к.т.н. В. Никитюк; доцент кафедри програмної інженерії, к.т.н. Д. Михалик; асистент М. Стадник; асистент Н. Шаблій; ст. викладач Л. Джиджора.

Матеріали ІХ науково-технічної конфіції «Інформаційні моделі, системи та технології» Тернопільського національного технічного університету імені Івана Пулюя, (Тернопіль, 8 – 9 грудня 2021 р.). – Тернопіль: Тернопільський національний технічний університет імені Івана Пулюя, 2021. – 210 с.

Адреса оргкомітету: ТНТУ ім. І. Пулюя, м. Тернопіль, вул. Руська, 56, 46001, тел. (0352) 52-41-33, факс (0352) 254983.

E-mail: conffis2021@gmail.com

Редагування, оформлення, верстка: Семенишин Г.М.

СЕКЦІЇ КОНФЕРЕНЦІЇ, ЯКІ ПРЕДСТВЛЕНІ В ЗБІРНИКУ

- Математичне моделювання;
- Інформаційні системи та технології;
- Комп'ютерні системи та мережі;
- Програмна інженерія та моделювання складних розподілених систем;
- Новітні фізико-технічні та освітні технології.

В збірнику надруковано тези доповідей ІХ науково-технічної конференції «Інформаційні моделі, системи та технології» (Тернопіль, 8–9 грудня 2021 р.) за такими науковими напрямками: математичне моделювання; інформаційні системи та технології; комп'ютерні системи та мережі; програмна інженерія та моделювання складних розподілених систем; новітні фізико-технічні та освітні технології.

Розрахований на науковців, викладачів та студентів вузів.

За зміст тез та дотримання норм академічної доброчесності відповідальність несе автор.

УДК 004.031.6: 004.056

Д.В. Кунинець, к.т.н., Ю.З. Лещин

(Тернопільський національний технічний університет імені Івана Пулюя, Україна)

ЗАСТОСУНОК ДЛЯ МОНІТОРИНГУ ДАНИХ РОЗУМНОГО БУДИНКУ

UDC 004.031.6: 004.056

D. Kunynets, Ph.D. Yu. Z. Leshchynshyn

SMART HOME DATA MONITORING APPLICATION

В житті людей з кожним днем з'являється все більше автоматизації, в тому числі у побуті. Інтернет речей користується попитом і разом з тим все більшу популярність набирають «розумні будинки». «Розумні будинки» оснащують різними підсистемами, такими як: сигналізація, система розпізнавання обличчя та голосу [1], автоматизовані вікна та двері, система регулювання температури, вологості, освітленості та багатьма іншими [2]. В зв'язку з цим з'являється необхідність можливості керування та моніторингу даних цих підсистем максимально зручно для кінцевого користувача.

Для вирішення такої задачі потрібно провести аналіз існуючих засобів моніторингу даних розумного будинку, зокрема їх користувацький інтерфейс (front end) на рахунок ергономіки та зручності користування. А також статистичні методи обчислення спожитих ресурсів і т.п., тобто back end. Завдяки цьому буде отримана інформація про функціональність та зручність користування існуючих додатків, що допоможе зі завданням удосконалення існуючих рішень або розробки нового застосунку.

Що стосується ергономіки користувацького інтерфейсу, то потрібно звернути увагу на найбільш використовувані закладки для відображення, наприклад: поточний стан будинку, статистики витрат, налаштування і програмування режимів роботи обладнання «розумного будинку». Основні параметри затрат електроенергії, водопостачання, рівень освітленості, вологість повітря, температура приміщення необхідно відповідним чином поділити між цими вкладками. Проте задача коректного та зручного відображення цих параметрів для користувача залишається актуальною. Для вирішення цієї задачі потрібно продумати структуру та дизайн застосунку. Він має бути зручний та зрозумілий для користувача, візуально лаконічний та з можливістю персоналізації.

Стосовно статистичних методів обчислення спожитих ресурсів, тобто back end то для побудови застосунку необхідно використати такі статистичні методи, які відповідають сучасним математичним моделям процесів електро, газо та водоспоживання і враховують їх періодичність споживання та генерації (наприклад електроенергія від сонячних панелей). А також ці методи повинні давати прогноз витрати ресурсів для задач планування та оптимального використання (прання вдень чи вночі якщо є сонячні панелі і багато тарифний лічильник) або резервування (заряджання електричних акумуляторів, або нагрів теплового акумулятора теплої підлоги).

Передбачити всі сценарії роботи «розумного будинку» практично неможливо. Необхідність створення застосунку для керування «розумним будинком» потребує детального обґрунтування та вибору існуючих методів та засобів його реалізації. Це дасть змогу створити зручний у використанні та перспективний застосунок для моніторингу даних «розумного будинку».

Література.

1. Марків В.А. Комп'ютерна система аутентифікації осіб / В.А. Марків, Г.М. Осухівська, Ю.З. Лещин, А.М. Луцків. // Матеріали ХХ наукової конференції ТНТУ ім. І. Пулюя. – 2017. – С. 90–91.
3. Лещин Ю.З. Проектування системи контролю та управління температурними режимами «розумного будинку» / Ю.З. Лещин, М.В. Павлюк. // Збірник тез доповідей ІХ Міжнародної науково-технічної конференції молодих учених та студентів „Актуальні задачі сучасних технологій“. ТНТУ – 2020. – С. 60–61.

А.М. Луцків, Г.А. Абоах, Р.К. Рувімбо, В.М. Соболю РОЗВ'ЯЗАННЯ ЗАДАЧ МАШИННОГО НАВЧАННЯ У СЕРЕДОВИЩАХ ІЗ РОЗПОДІЛЕНОЮ ПАМ'ЯТТЮ A.M. Lutskev, H.A. Aboah, R.K. Ruwimbo, V.M. Sobol RESOLVING MACHINE LEARNING TASKS IN DISTRIBUTED MEMORY ENVIRONMENT	130
С.В. Соленко, Р.О. Жаровський канд. техн. наук МОЖЛИВОСТІ СЕРЕДОВИЩА PLUTUS PLAYGROUND ДЛЯ НАПИСАННЯ ТА ТЕСТУВАННЯ СМАРТ-КОНТРАКТІВ S. Solenko, R. Zharovskiy Ph.D. PLUTUS PLAYGROUND ENVIRONMENT CAPABILITIES FOR WRITING AND TESTING SMART CONTRACTS	131
Д.В. Кунинець, Ю.З. Лещинин ЗАСТОСУНОК ДЛЯ МОНІТОРИНГУ ДАНИХ РОЗУМНОГО БУДИНКУ D. Kunynets, Ph.D. Yu. Z. Leshchyshyn SMART HOME DATA MONITORING APPLICATION	133
В. Хвостівський, Г. Осухівська, Л. Хвостівська ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ СИСТЕМИ ОПРАЦЮВАННЯ МЕРЕЖЕВОГО ТРАФІКУ V. Khvostivskyy, H. Osukhivska, L. Khvostivska NETWORK TRAFFIC PROCESSING SYSTEM SOFTWARE	134
Н. Шаблій, А. Шаблій АНАЛІЗ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ МЕТОДІВ БІОМЕТРИЧНОЇ АУТЕНТИФІКАЦІЇ ЗА КЛАВІАТУРНИМ ПОЧЕРКОМ N. Shabliy, A. Shabliy ANALYSIS OF INFORMATION TECHNOLOGIES OF BIOMETRIC AUTHENTICATION METHODS KEYSTROKE DYNAMICS	135
Н. Шаблій, А. Шаблій АРХІТЕКТУРА ІНФОРМАЦІЙНОЇ СИСТЕМИ БІОМЕТРИЧНОЇ АУТЕНТИФІКАЦІЇ ЗА КЛАВІАТУРНИМ ПОЧЕРКОМ N. Shabliy, A. Shabliy ARCHITECTURE OF BIOMETRIC AUTHENTICATION INFORMATION SYSTEM KEYSTROKE DYNAMICS	136
В.В. Яцишин, Ю.Р. Мельник, А.В. Волощук КОМПОНЕНТИ КОМП'ЮТЕРНОЇ СИСТЕМИ ПРОГНОЗУВАННЯ ПОВЕДІНКИ РУХУ АВТОМОБІЛЯ V.V. Yatsyshyn, Yu.R. Melnyk, A.V. Voloshchuk COMPONENTS OF COMPUTER SYSTEM FOR CAR MOVEMENT BEHAVIOR PREDICTION	137
І.М. Митчик ДО ПРОБЛЕМИ ТРАСУВАННЯ ПРОВОДОВОГО МОНТАЖУ I.M. Mytchuk O A PROBLEM ON TRACING OF WIRE ASSEMBLING	138

Додаток Б

Лістинг коду

Лістинг коду створення діаграм показників

```
function showChart1() {
    timestamps1 = this.measurements1.map(function (v) {
        return v.timestamp;
    });
    values1 = this.measurements1.map(function (v) {
        return v.value;
    });

    console.log(timestamps1);
    console.log(values1);
    let popCanvasName = document.getElementById("airChart");
    let barChartName = new Chart(popCanvasName, {
        type: 'radar',
        data: {
            labels: timestamps1,
            datasets: [{
                label: 'Air Humidity',
                data: values1,
                backgroundColor: [
                    'rgba(255, 99, 132, 0.6)',
                    'rgba(54, 162, 235, 0.6)',
                    'rgba(255, 206, 86, 0.6)',
                    'rgba(75, 192, 192, 0.6)',
                    'rgba(153, 102, 255, 0.6)',
                ]
            }]
        },
        options: {
            responsive: false
        }
    });
}

function getChartData1() {
    return fetch('https://localhost:5001/api/aggregator/1',
        {
            method: 'get',
            headers: {
                'Content-Type': 'application/json;charset=UTF-
8'
            }
        })
    .then(function (response) {
        if (response.ok) {
            return response.text();
        }
    });
}
```

```

        } else {
            throw Error('Response Not OK');
        }
    })
    .then(function (text) {
        try {
            return JSON.parse(text);
        } catch (err) {
            throw Error('Method Not Found');
        }
    })
    .then(function (responseJSON) {
        measurments1 = responseJSON;
        console.log(responseJSON);
        showChart1();
    })
}
getChartData1();
'rgba(153, 102, 255, 0.6)',
    ]
    }],
    },
    options: {
        responsive: false
    }
});
}
function getChartData2() {
    return fetch('https://localhost:5001/api/aggregator/2',
        {
            method: 'get',
            headers: {
                'Content-Type': 'application/json;charset=UTF-
8'
            }
        }
    )
    .then(function (response) {
        if (response.ok) {
            return response.text();
        } else {
            throw Error('Response Not OK');
        }
    })
    .then(function (text) {
        try {
            return JSON.parse(text);
        } catch (err) {
            throw Error('Method Not Found');
        }
    })
    .then(function (responseJSON) {
        measurments2 = responseJSON;
        console.log(responseJSON);
    })
}

```

```

        showChart2 ();
    })
}
getChartData2 ();

var timestamps3 = [];
var values3 = [];
var measurements3 = [];

function showChart3() {
    timestamps3 = this.measurements3.map(function (v) {
        return v.timestamp;
    });
    values3 = this.measurements3.map(function (v) {
        return v.value;
    });

    console.log(timestamps3);
    console.log(values3);
    let popCanvasName = document.getElementById("illumChart");
    let barChartName = new Chart(popCanvasName, {
        type: 'line',
        data: {
            labels: timestamps3,
            datasets: [{
                label: 'Level of Illumination',
                data: values3,
                backgroundColor: [
                    'rgba(255, 99, 132, 0.6)',
                    'rgba(54, 162, 235, 0.6)',
                    'rgba(255, 206, 86, 0.6)',
                    'rgba(75, 192, 192, 0.6)',
                    'rgba(153, 102, 255, 0.6)',
                ]
            }]
        },
        options: {
            responsive: false
        }
    });
}

function getChartData3() {
    return fetch('https://localhost:5001/api/aggregator/3',
        {
            method: 'get',
            headers: {
                'Content-Type': 'application/json;charset=UTF-
8'
            }
        }
    )
    .then(function (response) {
        if (response.ok) {

```

```

        return response.text();
    } else {
        throw Error('Response Not OK');
    }
})
.then(function (text) {
    try {
        return JSON.parse(text);
    } catch (err) {
        throw Error('Method Not Found');
    }
})
.then(function (responseJSON) {
    measurements3 = responseJSON;
    console.log(responseJSON);
    showChart3();
})
}
getChartData3();
//showChart();
</script>

```

ЛІСТИНГ КОДУ ВИМІРЮВАННЯ ПОКАЗНИКІВ ВОЛОГОСТІ

```

function showChart1() {
    timestamps1 = this.measurements1.map(function (v) {
        return v.timestamp;
    });
    values1 = this.measurements1.map(function (v) {
        return v.value;
    });

    console.log(timestamps1);
    console.log(values1);
    let popCanvasName = document.getElementById("airChart");
    let barChartName = new Chart(popCanvasName, {
        type: 'radar',
        data: {
            labels: timestamps1,
            datasets: [{
                label: 'Air Humidity',
                data: values1,
                backgroundColor: [
                    'rgba(255, 99, 132, 0.6)',
                    'rgba(54, 162, 235, 0.6)',
                    'rgba(255, 206, 86, 0.6)',
                    'rgba(75, 192, 192, 0.6)',
                    'rgba(153, 102, 255, 0.6)',
                ]
            }]
        }
    });
}

```

```

        ]]
    },
    options: {
        responsive: false
    }
});
}
function getChartData1() {
    return fetch('https://localhost:5001/api/aggregator/1',
        {
            method: 'get',
            headers: {
                'Content-Type': 'application/json;charset=UTF-
8'
            }
        })
    .then(function (response) {
        if (response.ok) {
            return response.text();
        } else {
            throw Error('Response Not OK');
        }
    })
    .then(function (text) {
        try {
            return JSON.parse(text);
        } catch (err) {
            throw Error('Method Not Found');
        }
    })
    .then(function (responseJSON) {
        measurements1 = responseJSON;
        console.log(responseJSON);
        showChart1();
    })
}
getChartData1();

```

Лістинг коду вимірювання показників температури

```

var timestamps2 = [];
var values2 = [];
var measurements2 = [];

function showChart2() {
    timestamps2 = this.measurements2.map(function (v) {
        return v.timestamp;
    });
    values2 = this.measurements2.map(function (v) {

```

```

        return v.value;
    });

    console.log(timestamps2);
    console.log(values2);
    let popCanvasName = document.getElementById("tempChart");
    let barChartName = new Chart(popCanvasName, {
        type: 'bar',
        data: {
            labels: timestamps2,
            datasets: [{
                label: 'Temperature',
                data: values2,
                backgroundColor: [
                    'rgba(255, 99, 132, 0.6)',
                    'rgba(54, 162, 235, 0.6)',
                    'rgba(255, 206, 86, 0.6)',
                    'rgba(75, 192, 192, 0.6)',
                    'rgba(153, 102, 255, 0.6)',
                ]
            }]
        },
        options: {
            responsive: false
        }
    });
}
function getChartData2() {
    return fetch('https://localhost:5001/api/aggregator/2',
        {
            method: 'get',
            headers: {
                'Content-Type': 'application/json;charset=UTF-
8'
            }
        })
    .then(function (response) {
        if (response.ok) {
            return response.text();
        } else {
            throw Error('Response Not OK');
        }
    })
    .then(function (text) {
        try {
            return JSON.parse(text);
        } catch (err) {
            throw Error('Method Not Found');
        }
    })
    .then(function (responseJSON) {
        measurments2 = responseJSON;
        console.log(responseJSON);
    });
}

```

```

        showChart2 ();
    })
}
getChartData2 ();

```

Лістинг коду вимірювання показників рівня освітленості

```

var timestamps3 = [];
var values3 = [];
var measurements3 = [];

function showChart3() {
    timestamps3 = this.measurements3.map(function (v) {
        return v.timestamp;
    });
    values3 = this.measurements3.map(function (v) {
        return v.value;
    });

    console.log(timestamps3);
    console.log(values3);
    let popCanvasName = document.getElementById("illumChart");
    let barChartName = new Chart(popCanvasName, {
        type: 'line',
        data: {
            labels: timestamps3,
            datasets: [{
                label: 'Level of Illumination',
                data: values3,
                backgroundColor: [
                    'rgba(255, 99, 132, 0.6)',
                    'rgba(54, 162, 235, 0.6)',
                    'rgba(255, 206, 86, 0.6)',
                    'rgba(75, 192, 192, 0.6)',
                    'rgba(153, 102, 255, 0.6)'
                ]
            }]
        },
        options: {
            responsive: false
        }
    });
}

function getChartData3() {
    return fetch('https://localhost:5001/api/aggregator/3',
        {
            method: 'get',
            headers: {
                'Content-Type': 'application/json;charset=UTF-

```



```
    }  
  })  
  .then(function (response) {  
    if (response.ok) {  
      return response.text();  
    } else {  
      throw Error('Response Not OK');  
    }  
  })  
  .then(function (text) {  
    try {  
      return JSON.parse(text);  
    } catch (err) {  
      throw Error('Method Not Found');  
    }  
  })  
  .then(function (responseJSON) {  
    measurements3 = responseJSON;  
    console.log(responseJSON);  
    showChart3();  
  })  
}  
getChartData3();
```