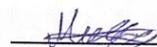


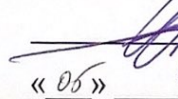
Комплексна бакалаврська кваліфікаційна робота на тему:

**«ІНФОРМАЦІЙНА СИСТЕМА «РОЗУМНИЙ БУДИНОК» НА БАЗІ ЛОКАЛЬНОЇ
КЛІЄНТ-СЕРВЕРНОЇ АРХІТЕКТУРИ. РОЗРОБЛЕННЯ ІНТЕРФЕЙСУ
КОРИСТУВАЧА»**

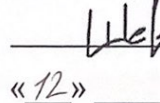
Виконав: студент 4 курсу, групи ІСТ-216
спеціальності 126 – «Інформаційні
системи та технології»

 Макар МАКСИМНЮК

Керівник: к.т.н., доц. каф. САІТ

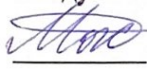
 Дмитро ПРОЦЕНКО
« 06 » 06 2025 р.

Рецензент: к.т.н., доц. каф. КН

 Олександр ШЕВЧУК
« 12 » 06 2025 р.

Допущено до захисту

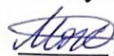
Завідувач кафедри САІТ

 д.т.н., проф. Віталій МОКІН
« 06 » 06 2025 р.

Вінницький національний технічний університет
Факультет інтелектуальних інформаційних технологій та автоматизації
Кафедра системного аналізу та інформаційних технологій
Рівень вищої освіти – перший (бакалаврський)
Галузь знань – 12 Інформаційні технології
Спеціальність 126 – «Інформаційні системи та технології»
Освітньо-професійна програма – Прикладні інформаційні технології

ЗАТВЕРДЖУЮ

Завідувач кафедри САІТ

 д.т.н., проф. Віталій МОКІН

“28” 03 2025 року

**ЗАВДАННЯ
НА КОМПЛЕКСНУ БАКАЛАВРСЬКУ КВАЛІФІКАЦІЙНУ РОБОТУ
СТУДЕНТУ**

Максимнюку Макару Віталійовичу

1. Тема роботи: «Інформаційна система «розумний будинок» на базі локальної клієнт-серверної архітектури. Розроблення інтерфейсу користувача»

керівник роботи: Проценко Д. П., к.т.н., доц. каф. САІТ

затверджені наказом ВНТУ від «20» 03 2025 року № 97

2. Термін подання студентом роботи 31.05.25 р.

3. Вихідні дані до роботи:

1) План будинку.

2) Можливість керування через графічний інтерфейс освітленням, вентиляцією та іншими електричними приладами будинку.

4. Зміст текстової частини:

1) Загальна характеристика об'єкту досліджень.

2) Вибір ІТ-інфраструктури для створення інтерфейсу користувача.

3) Налаштування інтерфейсу користувача та інтеграція автоматизації.

4) Розширення функціоналу інтерфейсу системи «розумний будинок».

5. Перелік ілюстративного матеріалу:

1) Критерії до вибору інструментів візуалізації;

2) Блок-схема алгоритму автоматизації роботи освітлення та вентиляції;

3) Конфігурування графічних елементів на плані;

4) Розробка графічного інтерфейсу користувача для окремих кімнат;

5) Апаратна реалізація інформаційної системи «розумний будинок» у вигляді макету;

6) Вигляд фрагменту головного екрану в мобільному додатку Home Assistant.

6. Консультанти розділів роботи:

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв

7. Дата видачі завдання 28.03.2021

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва та зміст етапу	Термін виконання		Примітка
		початок	закінчення	
1	Загальна характеристика об'єкту досліджень	01.04	15.04	Вик
2	Вибір ІТ-інфраструктури для створення інтерфейсу користувача	15.04	30.04	Вик
3	Налаштування інтерфейсу користувача та інтеграція автоматизації.	01.05	10.05	Вик
4	Розширення функціоналу інтерфейсу системи «розумний будинок»	10.05	20.05	Вик
5	Оформлення матеріалів до захисту БКР	20.05	30.05	Вик

Студент



Макар МАКСИМНЮК

Керівник роботи



Дмитро ПРОЦЕНКО

АНОТАЦІЯ

Бакалаврська кваліфікаційна робота складається з 71 сторінки формату А4, на яких є 29 рисунків, список використаних джерел містить 18 найменувань.

У роботі досліджено підходи до розробки інтерфейсів користувача для систем розумного будинку з урахуванням вимог зручності, інтуїтивності, адаптивності та ефективної взаємодії. Проаналізовано функціональні можливості середовища Home Assistant, його архітектуру та інструменти створення графічного інтерфейсу на базі Lovelace. Обґрунтовано вибір локальної клієнт-серверної моделі, сформульовано вимоги до структури й функціональності інтерфейсу, реалізовано інтерактивний план приміщень із картками керування пристроями та сценаріями. Проведено тестування створеного інтерфейсу за критеріями швидкодії, доступності на різних пристроях та загальної зручності користування. Отримані результати засвідчують ефективність запропонованого рішення для побутового застосування та подальшого розвитку інтелектуальних систем керування середовищем проживання.

Ключові слова: Home Assistant, «розумний будинок», інтерфейс, YAML, інформаційна система.

ABSTRACT

The bachelor's qualification work consists of 71 pages of A4 format, on which there are 29 figures, the list of used sources contains 18 names.

The work explores approaches to developing user interfaces for smart home systems, taking into account the requirements of convenience, intuitiveness, adaptability and effective interaction. The functional capabilities of the Home Assistant environment, its architecture and tools for creating a graphical interface based on Lovelace are analyzed. The choice of a local client-server model is justified, requirements for the structure and functionality of the interface are formulated, an interactive plan of the premises with device control cards and scenarios is implemented. The created interface is tested according to the criteria of speed, availability on different devices and general ease of use. The results obtained demonstrate the effectiveness of the proposed solution for domestic use and further development of intelligent living environment control systems.

Keywords: Home Assistant, "smart home", interface, YAML, information system.

ЗМІСТ

ВСТУП.....	3
1 ЗАГАЛЬНА ХАРАКТЕРИСТИКА ОБ’ЄКТУ ДОСЛІДЖЕНЬ	6
1.1 Завдання та вимоги до інформаційних систем типу «розумний будинок»	6
1.2 Аналіз засад побудови інтерфейсів користувача для інформаційних систем управління на базі локальних архітектур	11
1.3 Особливості побудови інтерфейсу користувача в системі Home Assistant. .	15
1.4 Висновки	19
2. ВИБІР ІТ-ІНФРАСТРУКТУРИ ДЛЯ СТВОРЕННЯ ІНТЕРФЕЙСУ КОРИСТУВАЧА	20
2.1 Вибір інструментів візуалізації в системі розумного будинку на базі сервера Home Assistant.....	20
2.2 Вибір апаратної частини інформаційної системи «розумний будинок» на базі Home Assistant.....	25
2.3 Висновки	27
3. НАЛАШТУВАННЯ ІНТЕРФЕЙСУ КОРИСТУВАЧА ТА ІНТЕГРАЦІЯ АВТОМАТИЗАЦІЇ.....	28
3.1 Налаштування інтерфейсу користувача.....	28
3.2 Створення сутностей розумного будинку з використанням додатку ESPHome	29
3.3 Створення автоматизації в Home Assistant.....	38
3.4 Висновки	42
4. РОЗШИРЕННЯ ФУНКЦІОНАЛУ ІНТЕРФЕЙСУ СИСТЕМИ «РОЗУМНИЙ БУДИНОК».....	44
4.1 Різновиди інформаційних панелей в Lovelace	44
4.2 Розробка графічного інтерфейсу користувача для окремих кімнат	47
4.3 Практична реалізація системи розумний будинок	51
4.4 Висновки	54
ВИСНОВКИ.....	56
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	58
Додаток А (обов’язковий) Технічне завдання.....	61
Додаток Б (обов’язковий) Протокол перевірки кваліфікаційної роботи	63
Додаток В (обов’язковий) Ілюстративна частина.....	64

ВСТУП

Актуальність теми. У сучасному світі концепція «розумного будинку» стрімко трансформується з футуристичної ідеї у повсякденну реальність, яка охоплює все більше житлових, офісних і комерційних приміщень. Впровадження інформаційних технологій в побут і створення взаємопов'язаних інтелектуальних пристроїв на базі Інтернету речей (IoT) сприяє підвищенню комфорту, безпеки, енергоефективності та якості життя.

Однією з ключових переваг локальних систем є високий рівень приватності. Оскільки всі дані обробляються і зберігаються безпосередньо на пристроях у межах локальної мережі, зникає потреба у передаванні чутливої інформації в хмару. Це суттєво знижує ризики витоку даних, зовнішнього стеження або несанкціонованого доступу, що особливо важливо при інтеграції камер відеоспостереження, розумних замків чи детекторів присутності. На відміну від хмарних сервісів, де конфіденційність часто залежить від політик сторонніх компаній, локальні рішення дають повний контроль користувачеві над усіма аспектами обробки даних.

Крім того, різноманіття протоколів зв'язку та низька сумісність окремих рішень ускладнює інтеграцію і потребує технічної обізнаності від користувачів. Іншою проблемою є швидке моральне старіння пристроїв, які можуть втратити підтримку або стати застарілими через кілька років після придбання.

Успішність впровадження будь-якої інформаційної системи напряду залежить не лише від її функціональності, а й від того, наскільки зручно та зрозуміло користувач може налаштовувати та керувати її поведінкою.

Сучасні тренди в розробці інтерфейсів для систем розумного будинку зосереджуються на простоті, доступності та адаптивності до різних платформ — смартфонів, планшетів, комп'ютерів, розумних годинників, а також голосових асистентів. Умовно інтерфейс виконує дві головні функції: відображення стану системи в реальному часі та надання зручних засобів керування її компонентами. Це охоплює як графічні панелі керування освітленням, температурою, безпекою,

так і механізми створення складних сценаріїв автоматизації без необхідності програмування.

Таким чином, інтерфейс у системах «розумного будинку» — це не лише технічний елемент, а ключовий фактор, який визначає успішність взаємодії користувача з усією системою. Його якість безпосередньо впливає на рівень комфорту, ефективність управління та загальну привабливість смарт-інфраструктури.

Мета і задачі дослідження. Метою даної роботи є підвищення ефективності функціонування інформаційної системи «розумний будинок» шляхом розроблення її на базі локальної клієнт-серверної архітектури та створення зручного й адаптивного інтерфейсу користувача, для забезпечення надійної, швидкої та безпечної взаємодії користувача з елементами автоматизованої системи управління освітленням, кліматом, безпекою та іншими компонентами побутової інфраструктури без необхідності підключення до зовнішніх хмарних сервісів. Реалізація такого підходу дозволяє підвищити автономність системи, зменшити залежність від зовнішніх факторів та забезпечити більший рівень конфіденційності й стабільності у процесі керування розумним середовищем.

Для досягнення поставленої мети необхідно розв'язати такі задачі:

- проаналізувати засади побудови інтерфейсів користувача для інформаційних систем управління на базі локальних архітектур;
- проаналізувати сучасні підходи до реалізації інтерфейсів у системах «розумний будинок» з урахуванням принципів зручності, інтуїтивності, доступності та адаптивності;
- визначити вимоги до функціональності, структури та дизайну інтерфейсу користувача для керування компонентами розумного будинку
- обґрунтувати вибір ІТ-інфраструктури, необхідної для реалізації локальної клієнт-серверної системи з інтерактивним графічним інтерфейсом.

- розробити інтерфейс користувача з урахуванням обраної архітектури, типових сценаріїв взаємодії та особливостей представлення даних у режимі реального часу.

- провести тестування розробленого інтерфейсу з оцінюванням його зручності, швидкодії, адаптивності до різних пристроїв та загальної ефективності взаємодії користувача з системою.

Об’єктом дослідження є інформаційні технології, що застосовуються для побудови систем «розумний будинок» на базі локальної клієнт-серверної архітектури, в контексті реалізації інтерактивного інтерфейсу користувача для ефективного керування елементами автоматизації побутового середовища.

Предметом дослідження є методи, засоби та принципи розробки інтерфейсу користувача для інформаційної системи «розумний будинок», реалізованої на основі локальної клієнт-серверної архітектури.

Публікації. За результатами даної роботи була зроблена доповідь на тему «Розроблення інтерфейсу користувача в інформаційній системі «розумний будинок» на базі Home Assistant» на Міжнародній науково-практичній інтернет-конференції Молодь в науці: дослідження, проблеми, перспективи (МН-2025) з публікацією тез [1].

Робота виконувалась на замовлення кафедри САІТ Вінницького національного технічного університету.

1 ЗАГАЛЬНА ХАРАКТЕРИСТИКА ОБ'ЄКТУ ДОСЛІДЖЕНЬ

1.1 Завдання та вимоги до інформаційних систем типу «розумний будинок»

Світовий ринок розумних будинків демонструє стійке зростання, що підтверджується актуальними статистичними даними. Так, у 2024 році його обсяг оцінюється в понад 127 мільярдів доларів США, а до 2030 року очікується зростання до понад 537 мільярдів доларів. Деякі аналітики навіть прогнозують досягнення позначки в понад 1,4 трильйона доларів до 2034 року. Середній щорічний темп зростання (CAGR) коливається в межах 20–27 відсотків, що свідчить про значну інвестиційну привабливість цього напрямку. Основними рушіями розвитку ринку виступають підвищений попит на енергоефективні рішення, зростання кількості пристроїв IoT, широке впровадження 5G-зв'язку, а також збільшення інтересу до автоматизованого управління освітленням, кліматом, системами безпеки та побутовими приладами.

Серед країн-лідерів у впровадженні технологій розумного будинку першою за масштабами виступають Сполучені Штати Америки, де вже у 2024 році нараховується понад 69 мільйонів розумних домогосподарств. Очікується, що до 2027 року їхня кількість сягне 93 мільйонів. Північна Америка загалом формує понад третину глобального ринку, тоді як Азійсько-Тихоокеанський регіон демонструє найвищу динаміку зростання. Особливо активними у цьому напрямі є Китай, Південна Корея та Японія, які інвестують значні ресурси у побудову розумної інфраструктури міст. У європейських країнах зростає інтерес до впровадження розумного освітлення, систем захисту та енергозберігаючих рішень, що також посилює вплив цього тренду на рівні домогосподарств.

Серед основних переваг, які забезпечують системи «розумного будинку», слід відзначити значне зниження витрат на енергоспоживання завдяки автоматичному керуванню опаленням, вентиляцією та освітленням. За даними досліджень, правильне впровадження таких систем дозволяє знизити

споживання енергії на 20–30 відсотків. Додатковим плюсом є підвищення рівня безпеки житла, адже сучасні смарт-системи можуть контролювати доступ, повідомляти про спроби вторгнення, виявляти дим, витоки газу або води та надсилати сповіщення на мобільні пристрої власника. Не менш важливим аспектом є підвищення комфорту: мешканець отримує можливість дистанційно керувати побутовими пристроями, формувати індивідуальні сценарії поведінки систем залежно від часу доби, присутності в приміщенні чи погодних умов.

Особливу роль у розвитку інтелектуальних систем управління відіграє інтеграція голосових помічників, таких як Alexa, Siri чи Google Assistant. Ці системи створюють своєрідний інтерфейс між користувачем і складною технічною архітектурою, забезпечуючи інтуїтивне керування через голосові команди або мобільні застосунки. Додатковий імпульс отримав ринок завдяки впровадженню універсального протоколу Matter, який стандартизує взаємодію між пристроями різних виробників і забезпечує високу сумісність, безпеку та масштабованість розумних рішень.

Важливим фактором, що сприяє зростанню популярності концепції розумного будинку, є зміна демографічних та соціальних пріоритетів. Міленіали, які є найактивнішою віковою категорією споживачів технологій, дедалі частіше надають перевагу житлу з вбудованими системами автоматизації. Згідно з опитуваннями, близько 47 відсотків людей цієї категорії вже мають щонайменше один пристрій розумного будинку, тоді як серед представників старшого покоління цей показник становить близько 24 відсотків.

Сучасні інформаційні системи типу «розумний будинок» є складними програмно-апаратними комплексами, що забезпечують автоматизоване керування інфраструктурою житлового простору (рис. 1.1). Вони поєднують різноманітні сенсорні пристрої, виконавчі механізми, контролери, локальні або хмарні сервери та інтерфейси взаємодії з користувачем. Незважаючи на велику кількість технічних реалізацій, усі такі системи мають спільні функціональні завдання та повинні відповідати певним вимогам щодо надійності, зручності використання, гнучкості конфігурації та безпеки [2].

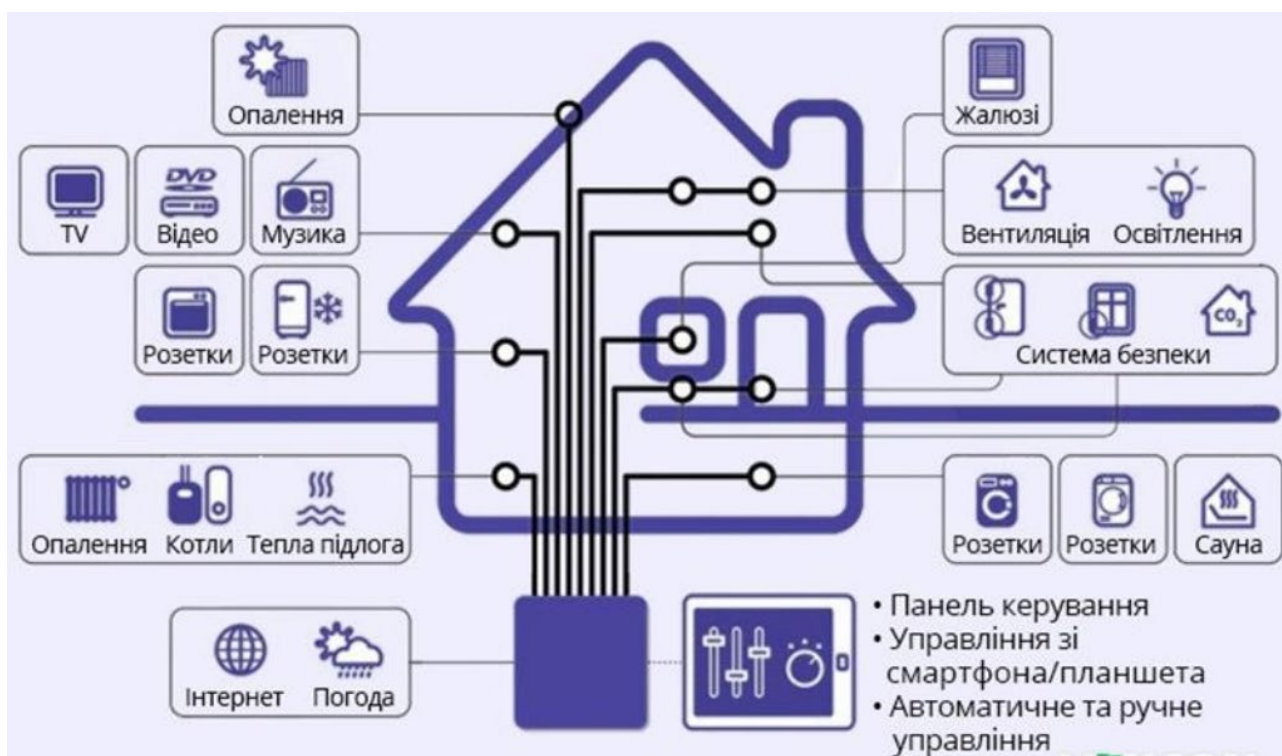


Рисунок 1.1 – Структурна схема інформаційної системи «розумний будинок»

Основне завдання системи «розумний будинок» — підвищити рівень комфорту, безпеки та енергоефективності житлового або комерційного простору. Це досягається за рахунок автоматизованого або напівавтоматичного керування освітленням, опаленням, вентиляцією, системами охорони, електроживленням та іншими підсистемами. Наприклад, система може самостійно регулювати температуру повітря відповідно до заданих користувачем параметрів або графіка, вмикати освітлення при виявленні руху в темний час доби, а також повідомляти про аварійні ситуації, такі як витік газу чи вторгнення [3-5].

Один із ключових принципів побудови таких систем полягає у забезпеченні інтелектуальної адаптації до поведінки користувача. Сучасні реалізації «розумного будинку» здатні аналізувати шаблони дій мешканців, формувати сценарії керування, які відповідають їхнім звичкам, та динамічно підлаштовуватись під змінні умови (погода, рівень освітленості, час доби, присутність у приміщенні тощо). Таким чином, одне з важливих завдань полягає

у впровадженні логіки, що забезпечує передбачувану, але не нав'язливу поведінку системи, яка не вимагає постійного втручання з боку людини.

Другим принциповим завданням є інтеграція гетерогенних пристроїв та протоколів зв'язку. У типовому «розумному будинку» використовуються пристрої з різними стандартами комунікації: Wi-Fi, Zigbee, Z-Wave, Bluetooth, Ethernet, 433 МГц, інфрачервоний зв'язок, а також локальні дротові інтерфейси. Одне із завдань системи полягає у забезпеченні сумісності між цими пристроями, що досягається через використання універсальних шлюзів, брокерів повідомлень (наприклад, MQTT), або програмних інтеграторів, таких як Home Assistant чи openHAB. У такому випадку виникає необхідність організації єдиного центру керування, що акумулює всі дані та надає інтерфейс доступу до них у зручній формі [6-8].

Важливим завданням є забезпечення високої надійності та відмовостійкості системи, особливо в контексті критичних підсистем, таких як охоронні чи пожежні. В таких випадках системи мають працювати без збоїв у реальному часі, з можливістю автономної роботи при втраті мережевого з'єднання або електропостачання. Це накладає вимоги на апаратну частину (блоки резервного живлення, дублювання ключових вузлів) та на програмну архітектуру (локальна обробка подій, відсутність залежності від хмари, ведення журналів подій).

Не менш важливим є завдання розробки ефективного та зручного інтерфейсу користувача, який дозволяє контролювати та конфігурувати систему без спеціальних знань. Інтерфейс має бути адаптивним, інтуїтивно зрозумілим, доступним з різних пристроїв — смартфонів, планшетів, ПК, а також за допомогою голосових помічників. При цьому важливо дотримуватись принципів інклюзивного дизайну, зокрема підтримки для людей похилого віку або осіб з порушеннями зору. Саме інтерфейс виступає ключовим елементом, через який користувач сприймає всю систему, тому його якість безпосередньо впливає на рівень задоволеності та прийняття технології.

Крім функціональних завдань, до систем типу «розумний будинок» пред'являються вимоги загального характеру, які можна умовно поділити на технічні, експлуатаційні та ергономічні.

До технічних вимог належать стабільність функціонування, можливість розширення системи (масштабованість), енергоефективність, сумісність з великою кількістю пристроїв, а також підтримка відкритих стандартів та протоколів. В умовах динамічного розвитку ринку ці вимоги набувають особливого значення: користувач повинен мати можливість додавати нові пристрої без необхідності повністю змінювати архітектуру.

Експлуатаційні вимоги охоплюють легкість встановлення, простоту технічного обслуговування, оновлення програмного забезпечення та мінімальну потребу у втручанні під час роботи. У разі розгортання системи в побутових умовах, вона має бути зрозумілою та доступною навіть для користувача без технічної освіти.

Ергономічні вимоги зосереджені на зручності взаємодії, швидкості доступу до функцій, зрозумілості повідомлень та адаптації інтерфейсу до різних типів користувачів. Особливо це стосується візуального представлення даних (графіки, індикатори стану, вікна повідомлень) та логіки навігації між розділами.

Окремо варто виділити вимоги до безпеки, які мають подвійний аспект: фізичний (захист житла) і інформаційний (захист даних). Перший аспект охоплює взаємодію з системами сигналізації, камерами, датчиками руху чи вікон, а другий — шифрування даних, автентифікацію користувачів, обмеження доступу, ведення журналів дій. У разі локального розгортання важливо передбачити ізоляцію системи від зовнішньої мережі або забезпечити безпечне VPN-з'єднання, якщо передбачено зовнішній доступ.

Крім вищезазначених, сучасні тенденції формують додаткові вимоги до гнучкості конфігурації та автоматизації. Користувач повинен мати змогу самостійно налаштовувати сценарії поведінки системи без програмування: наприклад, включення освітлення при заході сонця, зменшення температури вночі, вимкнення всіх пристроїв при виході з дому. Це можливо завдяки

інтерфейсам візуального програмування, таким як YAML-конфігурації в Home Assistant або блокові редактори на зразок Node-RED [9].

Підсумовуючи, можна зазначити, що системи «розумний будинок» покликані не лише автоматизувати побутові процеси, а й створити максимально комфортне, безпечне та гнучке середовище для користувача. Для цього необхідно враховувати не лише технічні параметри, а й зосереджуватись на питаннях інтеграції, інтерфейсу, захисту даних і персоналізації. Збалансоване поєднання функціональних завдань і вимог до реалізації формує основу ефективної та сучасної інформаційної системи типу «розумний будинок», здатної адаптуватися до різноманітних сценаріїв використання.

1.2 Аналіз засад побудови інтерфейсів користувача для інформаційних систем управління на базі локальних архітектур

Інтерфейс користувача є ключовим елементом будь-якої інформаційної системи управління, адже саме через нього забезпечується комунікація між людиною та технічними засобами, що виконують певні функції автоматизації, моніторингу чи керування. У випадку систем управління на базі локальних архітектур, до яких належать і розумні будинки, роль інтерфейсу не зводиться лише до графічного оформлення: він має бути інструментом, який дозволяє ефективно та надійно взаємодіяти з усією функціональною логікою системи в умовах автономної роботи, без постійної залежності від зовнішніх серверів чи хмарних платформ.

Побудова інтерфейсів для локальних інформаційних систем управління має низку особливостей, які відрізняють їх від традиційних веб-інтерфейсів або мобільних застосунків із хмарною підтримкою. Передусім йдеться про вимоги до швидкодії, безпеки, стабільності доступу до локального серверного обладнання та універсальності використання на різних пристроях у межах локальної мережі. Також значною мірою на архітектуру інтерфейсу впливають особливості середовища, в якому він застосовується, зокрема фізична структура

простору, кількість і типи контрольованих пристроїв, роль користувачів (наприклад, адміністратор, гість, обмежений користувач) та сценарії їхньої взаємодії із системою.

У загальному випадку інформаційна система управління з локальною архітектурою передбачає наявність серверного програмного забезпечення, яке розгорнуто на пристрої в межах локальної мережі (наприклад, Raspberry Pi, домашній сервер, NAS тощо), а також клієнтської частини, доступної через браузер, мобільний застосунок або спеціальний GUI-модуль (рис. 1.2).

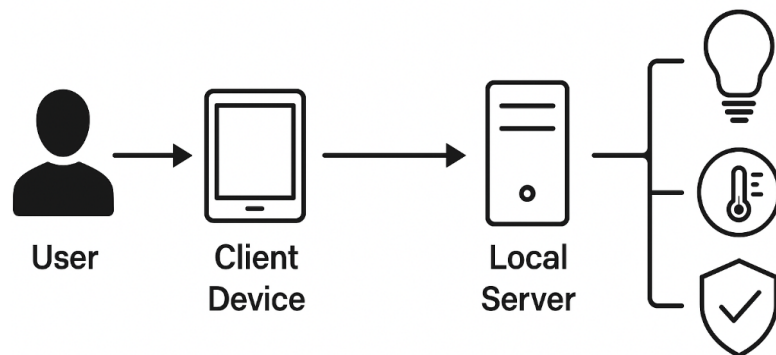


Рисунок 1.2 - Загальна схема взаємодії користувача з локальною системою керування через інтерфейс

Класичним прикладом є платформи на зразок Home Assistant, Domoticz або openHAB, що дозволяють будувати інтерфейси, орієнтовані на локальний контроль пристроїв з високим ступенем конфігурованості.

Однією з фундаментальних засад побудови таких інтерфейсів є локальність обробки та збереження даних. На відміну від хмарних рішень, де користувач взаємодіє через API із зовнішніми сервісами, у локальній архітектурі усі дані зберігаються безпосередньо на локальному сервері, і весь інтерфейс формує запити до нього напряму. Це накладає обмеження на швидкість передачі даних, а також вимагає врахування рівня навантаження на серверну частину, що безпосередньо впливає на продуктивність інтерфейсу. Типову структуру клієнт-серверної взаємодії в системі Home Assistant наведено на рисунку 1.3.

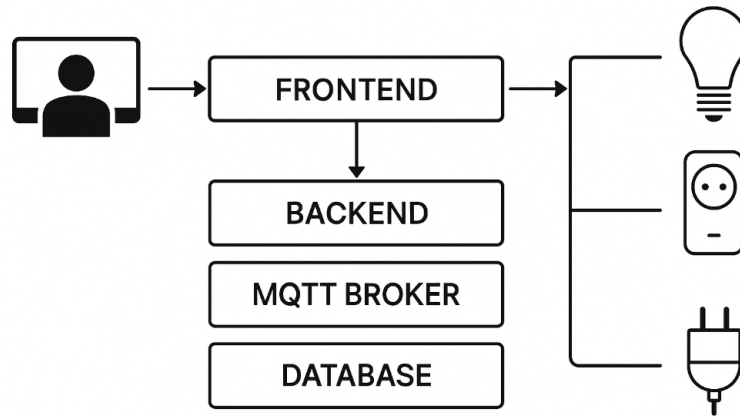


Рисунок 1.3 - Структура клієнт-серверної взаємодії в системі Home Assistant

Ще одним важливим принципом побудови таких інтерфейсів є інтуїтивність та зручність використання, що забезпечує доступність системи для широкого кола користувачів без необхідності технічної підготовки. У розрізі систем «розумного будинку» інтерфейс має не лише відображати стан пристроїв (увімкнено/вимкнено, температура, вологість, рівень освітлення тощо), але й дозволяти формувати сценарії взаємодії, налаштовувати тригери, встановлювати графіки чи логіку поведінки на основі умов (рис. 1.4).

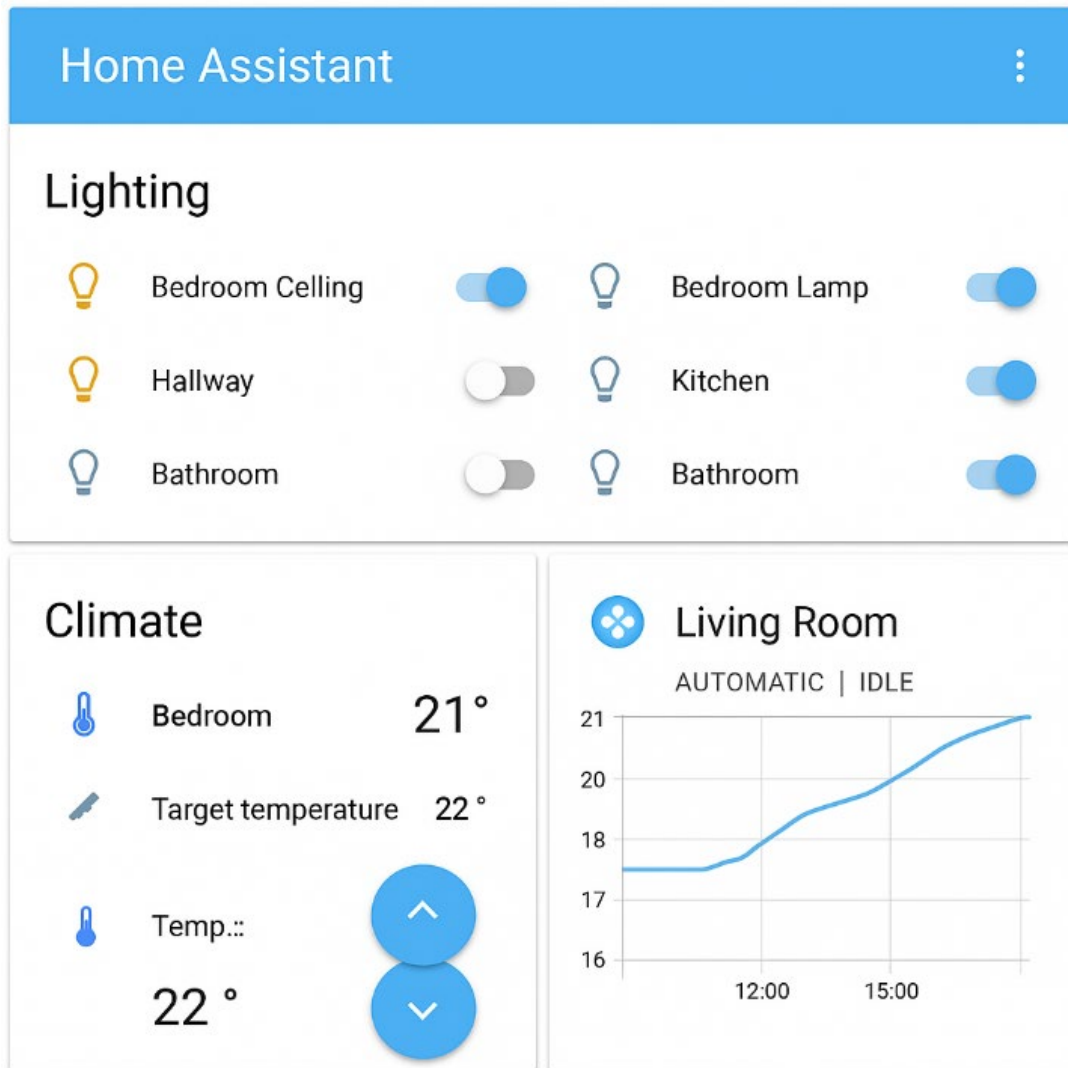


Рисунок 1.4 - Приклад панелі керування освітленням та кліматом у Home Assistant

Окрему увагу слід приділити адаптивності інтерфейсу, яка означає коректне відображення й зручність керування незалежно від розмірів екрана чи пристрою, з якого здійснюється доступ. У контексті локальних архітектур це передбачає підтримку браузерного відображення на ПК, планшетах, смартфонах, а також, за потреби, інтеграцію з інтерфейсами через голосових помічників або настінні панелі керування.

З технічного погляду, для локальних інформаційних систем інтерфейс має бути тісно інтегрованим із серверною частиною через внутрішні API. У системах на зразок Home Assistant цей зв'язок реалізується за допомогою WebSocket або

REST-запитів, які дозволяють оновлювати дані в реальному часі без повного перезавантаження сторінки.

Не менш важливою є можливість персоналізації інтерфейсу, що забезпечує адаптацію до конкретного користувача або контексту використання. Це може включати створення власних панелей для різних приміщень, налаштування рівнів доступу, розмежування адміністративних і користувацьких прав, а також можливість вибору теми оформлення, мови інтерфейсу, режиму доступу (гостьовий, повний тощо).

Щодо безпеки, локальні інтерфейси мають перевагу — всі дані зберігаються в межах локальної мережі. Проте це вимагає реалізації автентифікації, захисту внутрішнього доступу до системи та шифрування трафіку (наприклад, через HTTPS або VPN).

Ще один аспект, що є надзвичайно важливим — реакція в реальному часі. Інтерфейс має не лише відображати дані, а й оперативно реагувати на зміну стану. Наприклад, при фіксації руху — оновлюється індикатор присутності або активується сценарій освітлення.

Підсумовуючи, побудова інтерфейсів користувача для інформаційних систем управління на базі локальних архітектур передбачає застосування ряду важливих принципів: простота, ефективність, адаптивність, безпека та гнучкість. Інтерфейс у таких системах не є другорядною частиною — він фактично визначає користувацький досвід, а отже, і рівень ефективності всієї автоматизованої системи.

1.3 Особливості побудови інтерфейсу користувача в системі Home Assistant.

У процесі розроблення інтерфейсів для систем типу «розумний будинок» важливе значення має врахування як технічних можливостей обраної архітектури, так і психофізіологічних особливостей користувачів. Існує ряд наукових досліджень, що висвітлюють вплив дизайну інтерфейсу на зручність,

швидкість реагування, інклюзивність і загальний користувацький досвід. Особливої уваги заслуговують ті проекти, в яких інтерфейс адаптовано під різні вікові категорії, сценарії керування та формати взаємодії (мобільний застосунок, веб-інтерфейс, голосове управління тощо).

У дослідженні [2] представлено оцінку трьох типів інтерфейсів за рівнем інтелекту (низький, середній і високий) у системах розумного будинку. Дослідники дійшли висновку, що користувачі похилого віку краще реагують на простіші інтерфейси, у той час як молодь демонструє кращі результати при роботі з більш складною автоматизованою логікою. Це свідчить про необхідність адаптації інтерфейсу до когнітивних можливостей користувача залежно від віку.

Інше дослідження [3], проведене групою авторів на платформі *Frontiers in Psychology*, стосувалося впливу дизайну слайдерів (розмір кнопок, колір, орієнтація) на зручність управління. Виявлено, що горизонтальні слайдери з яскравим маркером та кнопками від 5 мм є найбільш зручними для людей усіх вікових груп, особливо в умовах швидкої взаємодії на сенсорному екрані.

Досвід розроблення інтерфейсів також досліджувався в контексті використання різних пристроїв для керування: персональних комп'ютерів, медіа-терміналів та смартфонів. Показано, що ПК-інтерфейс краще підходить для планування складних сценаріїв, у той час як мобільний застосунок більш придатний для оперативного керування. Згідно з результатами, мобільні інтерфейси частіше використовувались у повсякденному житті.

Окреме значення має врахування потреб користувачів похилого віку. У межах одного з емпіричних досліджень запропоновано рекомендації щодо вертикального розміщення елементів, використання шрифтів не менше 16 pt та забезпечення чіткого контрасту для елементів управління. Це дозволяє підвищити доступність системи та зменшити ризик помилкових дій.

Останнім часом зростає кількість публікацій, присвячених організації візуальних панелей в Home Assistant. Наприклад, використання карток, розділів і панелей із drag-and-drop підтримкою у Lovelace UI дозволяє користувачеві

самостійно створювати оптимальну конфігурацію інтерфейсу, що покращує сприйняття системи і спрощує процес управління [11].

Одним із ключових елементів взаємодії користувача з системою «розумний будинок» є інформаційна панель, яка виконує функцію центрального інтерфейсу управління. У системі Home Assistant, яка базується на локальній архітектурі, головним інструментом для створення таких панелей є Lovelace UI — динамічне, конфігуруване середовище, що дозволяє формувати кастомізовані панелі з використанням віджетів, карток, інтерактивних елементів та сценаріїв.

Основний принцип побудови панелей у Home Assistant полягає у максимальній візуальній доступності до основних керованих сутностей: перемикачів, сенсорів, контролерів клімату, безпеки, мультимедіа тощо. Кожна інформаційна панель складається з одного або кількох розділів (views), які можуть мати різне відображення залежно від типу пристрою: настільного ПК, планшета чи мобільного телефона [12-14].

Lovelace UI підтримує декларативний підхід до конфігурування, де кожна панель описується у вигляді YAML-файлу або керується через графічний редактор. Це дозволяє створювати адаптивні та динамічні представлення даних. Наприклад, користувач може створити окремі розділи для керування освітленням, безпекою, температурою, мультимедійними пристроями, а також відображення журналів подій або системного стану.

Ключовими елементами Lovelace-панелей є картки (cards), які виконують роль візуальних контейнерів. Серед найпопулярніших:

- entities: перелік сенсорів, перемикачів, індикаторів;
- history-graph: графіки зміни параметрів у часі;
- sensor: відображення конкретного значення з датчика;
- button: елементи ручного керування;
- picture-elements: візуалізація поверх плану будинку;
- alarm-panel: елементи безпеки;
- media-control: мультимедійне керування.

Окрему увагу приділяють адаптивності та мобільній оптимізації, зокрема використанню технологій CSS grid, media queries та підтримці тем оформлення. Система автоматично підлаштовується до розміру екрану, що дозволяє зберігати читабельність та функціональність навіть на пристроях з малим екраном [15-17].

Користувач може організувати візуальну ієрархію елементів відповідно до своїх пріоритетів — наприклад, винести найважливіші зони (вітальня, кухня, спальня) на головний екран, а допоміжні функції (наприклад, лог журналу чи автоматизації) перенести до вкладок. Додатково підтримується логіка умовного відображення, при якій елементи інтерфейсу можуть змінюватись залежно від стану пристрою чи сценарію (наприклад, відображення кнопки лише вночі).

На основі численних практичних реалізацій панелей у Home Assistant, можна сформулювати такі методичні підходи:

- розділяти панелі на логічні зони за типом керованих пристроїв або за просторовим принципом (приміщення);
- уникати надмірного перевантаження інформацією — не більше 6–8 елементів на один екран мобільного пристрою;
- забезпечити зворотний зв'язок для кожної дії (зміна кольору, анімація перемикача тощо);
- використовувати іконки, текстові підписи й індикатори одночасно для підвищення наочності.

Home Assistant також підтримує використання кастомних компонентів та карток (через HACS), що дозволяє розширити функціональність стандартних елементів і створювати інтерфейси з індивідуальними сценаріями, графіками, HTML-блоками, інтерактивними картами та звуковими повідомленнями.

Таким чином, створення інформаційних панелей у Home Assistant є гнучким процесом, що поєднує декларативну структуру конфігурації з великою кількістю візуальних та логічних елементів, що дозволяє реалізувати інтерфейс користувача будь-якого рівня складності — від базового контролю до повноцінної SCADA-моделі у межах локального розумного середовища.

1.4 Висновки

У цьому розділі було проаналізовано архітектурні, функціональні та інтерфейсні особливості побудови інформаційних систем типу «розумний будинок» з акцентом на реалізацію в умовах локальної клієнт-серверної взаємодії. Визначено, що використання локальної архітектури забезпечує вищу приватність, автономність та швидкість реагування системи в порівнянні з хмарними рішеннями, що особливо актуально для побутових середовищ із вимогами до надійності, безпеки та конфіденційності.

Особливу увагу приділено структурі інтерфейсу користувача, який є ключовою складовою ефективної взаємодії з системою. Проведено аналіз принципів побудови інтерфейсів у середовищі Home Assistant. Встановлено, що інтерфейс має відповідати критеріям простоти, адаптивності, доступності для користувачів різного віку, а також підтримувати персоналізацію та швидкий зворотний зв'язок.

У ході дослідження розглянуто типові задачі систем «розумний будинок», включаючи моніторинг освітлення, клімату, безпеки, енергоспоживання та автоматизацію сценаріїв. Наведено приклади структури інтерфейсів, принципи реалізації в реальному часі через WebSocket, індикацію подій, використання MQTT-брокера та бази даних. Особливу увагу приділено дослідженню вимог до дизайну елементів інтерфейсу, зокрема відповідно до сучасних наукових джерел, які підкреслюють важливість адаптації до вікових і когнітивних особливостей користувачів.

Проведене дослідження підтвердило доцільність використання локального середовища на базі Home Assistant як гнучкої платформи для реалізації інтерфейсу системи «розумний будинок». Розроблений підхід може бути масштабований і адаптований під індивідуальні потреби користувачів, забезпечуючи при цьому високий рівень автономності, безпеки та ефективності взаємодії.

2 ВИБІР ІТ-ІНФРАСТРУКТУРИ ДЛЯ СТВОРЕННЯ ІНТЕРФЕЙСУ КОРИСТУВАЧА

2.1 Вибір інструментів візуалізації в системі розумного будинку на базі сервера Home Assistant

У системах розумного будинку візуалізація даних відіграє критично важливу роль. Вона забезпечує користувача можливістю контролю, моніторингу і аналізу параметрів функціонування будинку в режимі реального часу [18]. На базі Home Assistant — популярного рішення з відкритим кодом для автоматизації побутових процесів — можна реалізувати широкий спектр візуальних інтерфейсів. Проте, ефективність таких інтерфейсів безпосередньо залежить від правильно обраного інструменту візуалізації.

У цьому розділі буде проведено аналіз основних інструментів візуалізації, які можна використовувати в екосистемі Home Assistant, а також обґрунтовано вибір оптимального варіанта з огляду на технічні та користувацькі аспекти.

У процесі побудови системи візуалізації для розумного будинку на базі Home Assistant особливу увагу слід приділяти вибору відповідних інструментів. Їх ефективність визначається рядом ключових вимог, які повинні забезпечувати як технічну сумісність, так і зручність для кінцевого користувача.

Насамперед, важливою є повна або часткова сумісність інструменту з Home Assistant. Це означає підтримку стандартних механізмів інтеграції, зокрема через API, MQTT або WebSocket-протоколи. Така сумісність забезпечує швидкий обмін даними між сервером і панеллю візуалізації, а також можливість інтерактивної роботи з пристроями [17-20].

Наступним критерієм є гнучкість у налаштуванні інтерфейсу. Інструмент повинен дозволяти вільне компонування елементів, зміну їх вигляду, стилізації, а також забезпечувати можливість персоналізації відповідно до індивідуальних

потреб користувача. Це особливо важливо у випадку використання панелей керування на планшетах або сенсорних екранах.

В умовах активного використання мобільних пристроїв необхідною є підтримка адаптивного дизайну або мобільних застосунків, що дозволяє керувати системою не лише з комп'ютера, а й зі смартфона чи планшета. Це значно підвищує зручність повсякденного користування системою.

Окрім цього, інструмент має підтримувати візуальні елементи, необхідні для комплексного моніторингу: графіки для аналізу історичних даних, карти для просторового відображення пристроїв, віджети для швидкого доступу до ключових параметрів, а також індикатори стану.

Ще одним критично важливим аспектом є розширюваність: інструмент повинен дозволяти підключення додаткових модулів, плагінів або скриптів для розширення функціоналу. Це забезпечує гнучкість у майбутньому без потреби зміни всієї системи.

Нарешті, не варто забувати про оптимізацію продуктивності. Обраний інструмент не повинен створювати значного навантаження на сервер або клієнтський пристрій. Легка та ефективна реалізація дозволяє забезпечити швидкодію, стабільність та надійність функціонування навіть на відносно слабкому обладнанні. На рисунку 2.1 зображено основні критерії до вибору інструментів візуалізації

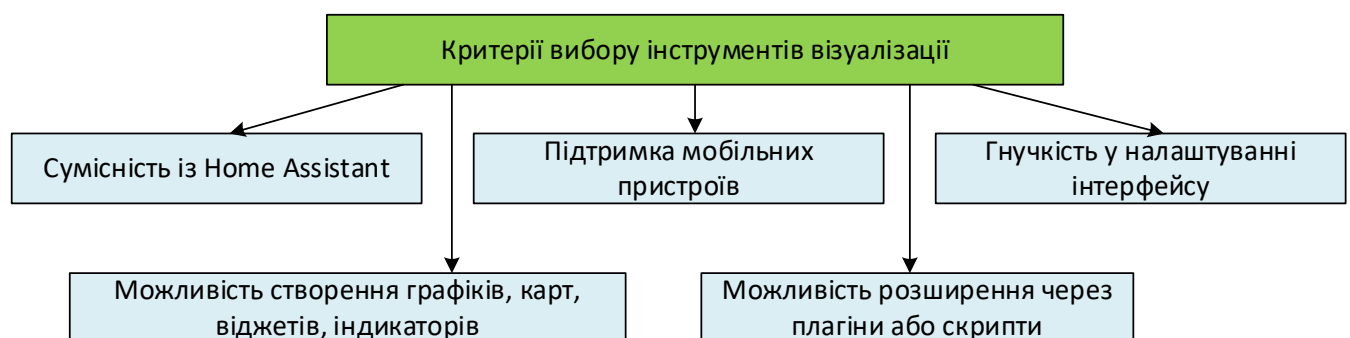


Рисунок 2.1 - Критерії до вибору інструментів візуалізації

Таким чином, лише поєднання вищезгаданих вимог дозволяє створити ефективну, надійну та зручну систему візуалізації для розумного будинку на базі Home Assistant.

У системі Home Assistant існує кілька популярних підходів до візуалізації даних, які відрізняються за складністю налаштування, функціональністю, зовнішнім виглядом і ступенем гнучкості. Серед найпоширеніших варто виділити Lovelace Dashboard, Grafana, Node-RED Dashboard, Floorplan та HADashboard (AppDaemon). Кожен із цих інструментів має власні переваги й обмеження в контексті сумісності з Home Assistant, можливості гнучкого налаштування інтерфейсу, підтримки мобільних пристроїв, побудови графіків та карт, візуального стилю, розширюваності та навантаження на систему.

Lovelace Dashboard є основним і вбудованим інтерфейсом Home Assistant. Він не потребує встановлення сторонніх рішень і пропонує високу сумісність, оскільки є нативною частиною системи. Користувач може як вручну редагувати YAML-конфігурації, так і користуватися інтерактивним редактором для налаштування зовнішнього вигляду. Панель адаптивна до мобільних пристроїв і підтримує створення динамічних карт, графіків, віджетів, сенсорів тощо. Lovelace підтримує темну і світлу тему, розширюється за допомогою кастомних карт і модулів. Завдяки тому, що інтерфейс оптимізований під Home Assistant, навантаження на систему є мінімальним.

Grafana, хоч і не є частиною Home Assistant, ідеально підходить для створення глибоких аналітичних панелей. Для її роботи зазвичай потрібне проміжне сховище даних, як-от InfluxDB або Prometheus, через яке здійснюється зв'язок із Home Assistant. Grafana не дає змоги керувати пристроями, однак забезпечує максимально деталізовані графіки, побудову історичних аналітичних звітів, агрегацій та обчислень. Інтерфейс Grafana гнучкий у налаштуванні, підтримує теми, плагіни та має адаптивну верстку для мобільних пристроїв. Оскільки Grafana працює як окрема служба, її використання вимагає додаткових ресурсів, хоча вона досить добре оптимізована.

Node-RED Dashboard дозволяє поєднати візуальне програмування логіки автоматизації з елементами візуалізації. Він підтримує сумісність з Home Assistant через WebSocket або MQTT, і дозволяє створювати прості графіки, кнопки, перемикачі та індикатори. Інтерфейс порівняно простий і менш гнучкий у плані дизайну, проте дозволяє створювати досить інформативні панелі. Node-RED добре працює на мобільних пристроях, має можливість розширення через додаткові модулі, але через універсальність потребує більше часу на налаштування.

Floorplan — це SVG-інструмент для створення інтерактивного плану приміщення. Він дозволяє візуально розмістити пристрої на графічній карті будинку, що значно полегшує просторове сприйняття системи. Візуальна частина дуже приваблива, однак потребує знання роботи з векторною графікою. З Floorplan можна керувати пристроями безпосередньо з карти, але аналітичні можливості цього рішення обмежені. Працює через Lovelace або окремі компоненти Home Assistant.

Нарешті, HADashboard — це інструмент, який працює разом з AppDaemon і дозволяє створювати висококастомізовані панелі керування, зокрема для настінних сенсорних екранів. Він має гнучкий, хоч і дещо застарілий, дизайн. Його використання потребує окремого налаштування та програмування, але дозволяє створити власну структуру інтерфейсу з урахуванням потреб користувача. HADashboard добре працює з мобільними пристроями, дозволяє використовувати теми, має доступ до всіх даних Home Assistant і може працювати автономно.

В таблиці 2.1 наведено порівняння засобів візуалізації в Home Assistant. Для поставленої задачі найкраще підходить Lovelace Dashboard.

Основні характеристики Lovelace Dashboard в Home Assistant:

- гнучкість  налаштування інтерфейсу: використання різних карток (Cards) для відображення пристроїв та даних та налаштування через YAML або інтерактивний інтерфейс;

- різноманітність карток: Entity Card, Glance Card, History Graph Card, Media Player Card, Map Card, Sensor Card, Picture Elements Card;
- групування пристроїв: організація пристроїв по вкладках (Tabs) та переглядах (Views);
- автоматизація та сценарії: можливість запуску автоматизацій та сценаріїв через інтерфейс;
- підтримка кастомних карток і компонентів: можливість інтеграції сторонніх плагінів для розширення функціональності;
- адаптивний дизайн: підтримка мобільних пристроїв з адаптивним інтерфейсом;
- інтеграція з іншими платформами: підтримка інтеграцій з Google Assistant, Amazon Alexa, Siri та мобільними додатками.

Ці характеристики дозволяють створити персоналізовану, зручну та функціональну панель управління для розумного будинку.

Таблиця 2.1 - Порівняльна таблиця засобів візуалізації

Критерій	Lovelace	Grafana	Node-RED Dashboard	Floorplan	HADashboard
Сумісність із Home Assistant	Висока	Через InfluxDB	Через API/MQTT	Висока	Висока
Гнучкість налаштування інтерфейсу	Висока	Висока	Середня	Висока	Висока
Підтримка мобільних пристроїв	Так	Так	Так	Обмежена	Так
Графіки / карти / віджети	Базові	Розширені	Базові	Карти	Віджети
Підтримка тем оформлення	Так	Так	Обмежена	Через CSS	Так
Розширюваність (плагіни/скрипти)	Висока	Висока	Висока	Обмежена	Середня
Навантаження на систему	Низьке	Середнє	Середнє	Низьке	Середнє
Можливість керування пристроями	Повна	Немає	Повна	Часткова	Повна
Рівень технічних знань для налаштування	Середній	Високий	Середній	Високий	Високий

2.2 Вибір апаратної частини інформаційної системи «розумний будинок» на базі Home Assistant

Для реалізації інформаційної системи «розумний будинок» було обрано архітектуру з локальним сервером на базі Home Assistant, контролером з Wi-Fi підключенням, а також сенсором кліматичного моніторингу. Така архітектура дозволяє забезпечити автономну роботу, гнучку інтеграцію з різними пристроями, мінімальне енергоспоживання та широкі можливості масштабування.

В якості центрального вузла системи обрано MLLSE mini PC m2 Air — мініатюрний комп'ютер, що поєднує низьке енергоспоживання та достатню обчислювальну потужність для хостингу серверної частини Home Assistant. Основні технічні характеристики пристрою:

- процесор: Intel Celeron J4125 (4 ядра, 2.0–2.7 GHz);
- оперативна пам'ять: 8 GB DDR4;
- накопичувач: 128 GB SSD (M.2 SATA);
- графіка: Intel UHD Graphics 600;
- порти введення/виведення: 2×USB 3.0, 2×USB 2.0, HDMI, LAN (1 Gbit/s), аудіо роз'єм, microSD;
- інтерфейси: Wi-Fi 802.11ac, Bluetooth 4.2;
- розміри: 12.8 × 12.8 × 3.8 см;
- живлення: 12V / 3A, енергоспоживання до 15 Вт.

Пристрій встановлюється у локальній мережі будинку, де виконує функцію основного сервера. На нього розгортається система Home Assistant OS (на основі Linux), яка забезпечує централізоване керування пристроями, візуалізацію даних та реалізацію сценаріїв автоматизації. Перевагою використання саме локального сервера є можливість функціонування без постійного підключення до Інтернету, що підвищує надійність і безпеку системи.

Для збору інформації з периферійних сенсорів та керування виконавчими пристроями обрано контролер ESP32, який працює під прошивкою ESPHome. Це мікроконтролер нового покоління, що підтримує бездротовий зв'язок, має потужну обчислювальну базу та широкі можливості для підключення зовнішніх пристроїв.

ESP32 взаємодіє з сервером Home Assistant по локальній Wi-Fi мережі, використовуючи інтеграцію ESPHome, яка дозволяє легко конфігурувати логіку роботи пристрою за допомогою YAML-файлів. Контролер може працювати як сенсорний вузол, як виконавчий пристрій (наприклад, для керування реле), або як комбінований вузол з двостороннім обміном інформацією.

Для реалізації базових задач збору даних було обрано цифровий сенсор DHT11, який забезпечує вимірювання температури повітря та відносної вологості. Сенсор є недорогим, простим у використанні та достатньо точним для побутових умов. Технічні характеристики DHT11:

- діапазон температури: $0^{\circ}\text{C} \dots +50^{\circ}\text{C}$;
- точність температури: $\pm 2^{\circ}\text{C}$;
- діапазон вологості: $20\% \dots 90\% \text{ RH}$;
- точність вологості: $\pm 5\% \text{ RH}$;
- частота оновлення: до 1 Гц;
- інтерфейс: однолінійний цифровий (власний протокол);
- живлення: 3.3 – 5.5 В;
- споживання: до 2.5 мА під час вимірювання.

Підключення сенсора до ESP32 здійснюється через один цифровий пін, а конфігурування зчитування даних реалізується в YAML-файлі ESPHome. У системі Home Assistant значення з DHT11 відображаються у вигляді сенсорів, доступних для візуалізації, збереження історії та використання в автоматичних сценаріях, наприклад, активації вентиляції або кондиціонування при перевищенні температурного порогу.

2.3 Висновки

Здійснено порівняльний аналіз інструментів для створення інтерфейсу користувача (Lovelace Dashboard, Grafana, Node-RED Dashboard, Floorplan та HADashboard (AppDaemon)) Вибрано оптимальне рішення для створення інтерфейсу користувача (Lovelace Dashboard)

Обраний набір апаратних засобів дозволяє реалізувати повноцінну систему розумного будинку з локальним зберіганням даних, автономною роботою без зовнішніх сервісів, високою надійністю та розширюваністю. MLLSE mini PC m2 Air забезпечує достатню обчислювальну потужність для централізованого керування, ESP32 з ESPHome - гнучкість інтеграції з датчиками та виконавчими пристроями, а сенсор DHT11 - базовий рівень моніторингу середовища. Така конфігурація є оптимальною з точки зору вартості, енергоспоживання та простоти налаштування.

3 НАЛАШТУВАННЯ ІНТЕРФЕЙСУ КОРИСТУВАЧА ТА ІНТЕГРАЦІЯ АВТОМАТИЗАЦІЇ

3.1 Налаштування інтерфейсу користувача

Створення візуалізації в Lovelace в Home Assistant дозволяє налаштувати інтерфейс користувача, що надає чітке та зручне відображення даних та станів пристроїв у вашому розумному домі. Алгоритм створення візуалізації в Lovelace виглядає так:

Перш за все, потрібно увійти в інтерфейс Home Assistant. Далі, в налаштуваннях потрібно перейти до розділу Lovelace або Dashboard. У цьому розділі ви зможете створювати та налаштовувати свої панелі, що дозволяють візуалізувати дані з різних пристроїв, сенсорів і інших компонентів системи.

Почати створення візуалізації можна за допомогою базових елементів, таких як карти. Карти дозволяють відображати різноманітну інформацію: від простих кнопок для керування пристроями до складних графіків та списків. Щоб додати карту, потрібно натискати "Редагувати" в верхньому правому куті інтерфейсу та вибирати "Додати картку". Home Assistant пропонує кілька типів карт: Glance, Entities, History Graph, Thermostat та інші. Кожна карта має свої налаштування для відображення конкретних даних.

Для кожної карти потрібно визначити, що саме буде виводитися. Наприклад, можна вибрати картку Glance, щоб швидко відображати стан кількох пристроїв, таких як температура, вологість або рівень освітленості. Для картки History Graph потрібно вибрати сенсор, дані якого ви хочете візуалізувати у вигляді графіку. Для картки Entities ви можете додавати кілька елементів, наприклад, світла, температури, датчики руху.

Наступним етапом є налаштування розташування карт. Для цього потрібно вибрати бажану сітку (наприклад, 1x1 або 2x2) та організувати карти відповідно до того, як ви хочете їх бачити на екрані. Lovelace дозволяє змінювати порядок

карт, зберігати панелі в окремих групах і навіть створювати спеціальні теми для різних сценаріїв використання (наприклад, денний та нічний режими).

Після цього можна налаштувати інтерфейс додатково, додаючи кнопки для керування пристроями, сповіщення, графіки енергоспоживання та інші інтерактивні елементи. Кнопки можуть бути налаштовані так, щоб виконувати дії, наприклад, увімкнення чи вимкнення світла або запуск автоматизацій. Важливо також додати сповіщення, які будуть інформувати користувача про зміни в станах пристроїв або важливі події, наприклад, коли температура в кімнаті перевищує певну межу.

Після завершення налаштування карток і елементів інтерфейсу ви можете зберегти зміни та перевірити, як виглядатиме ваша візуалізація на екрані. Lovelace також дозволяє створювати декілька панелей з різними елементами для окремих кімнат або груп пристроїв, що робить інтерфейс ще зручнішим для користувача.

Якщо потрібно більш детально налаштувати вигляд карт, можна редагувати їх вручну через YAML, що дозволяє мати більше гнучкості при налаштуванні стилів і параметрів кожної картки. YAML дозволяє детальніше налаштувати відображення інформації, кольори, шрифти та інші візуальні елементи.

В результаті, ви отримаєте повноцінну візуалізацію, яка не тільки надає зручний доступ до інформації, але й дозволяє оперативно керувати пристроями у вашому розумному будинку.

3.2 Створення сутностей розумного будинку з використанням додатку ESPHome

ESPHome — це потужна платформа для створення розумних пристроїв на базі ESP8266 та ESP32. Вона дозволяє створювати кастомні рішення для домашньої автоматизації без потреби у складних програмуваннях, просто шляхом налаштування YAML файлів. ESPHome підтримує широкий спектр

пристроїв, таких як сенсори, перемикачі, регулятори, кнопки, датчики руху та багато іншого, що дозволяє зібрати індивідуальну систему для будь-яких потреб.

Процес створення елементів у ESPHome зазвичай включає визначення типу пристрою, вибір піну для підключення та налаштування параметрів, таких як інтервали оновлення даних або дії, що виконуються при зміні стану. Наприклад, для сенсорів можна вказати період оновлення значень, для перемикачів — пін, до якого підключено реле або світло. ESPHome також підтримує інтерфейси для інтеграції з Home Assistant через API або MQTT, що дозволяє користувачам інтегрувати свої пристрої у широку екосистему розумного будинку.

Налаштування пристроїв через YAML конфігурацію надає гнучкість у виборі функціоналу, а зручний веб-інтерфейс дозволяє без труднощів завантажити нові налаштування на пристрій та здійснювати моніторинг його роботи. Всі ці можливості дають користувачам повний контроль над своїми розумними пристроями, дозволяючи створювати індивідуальні рішення для автоматизації. На рисунку 3.1 зображено вигляд пристрою на ESP 32 з конфігуванням сутностей.

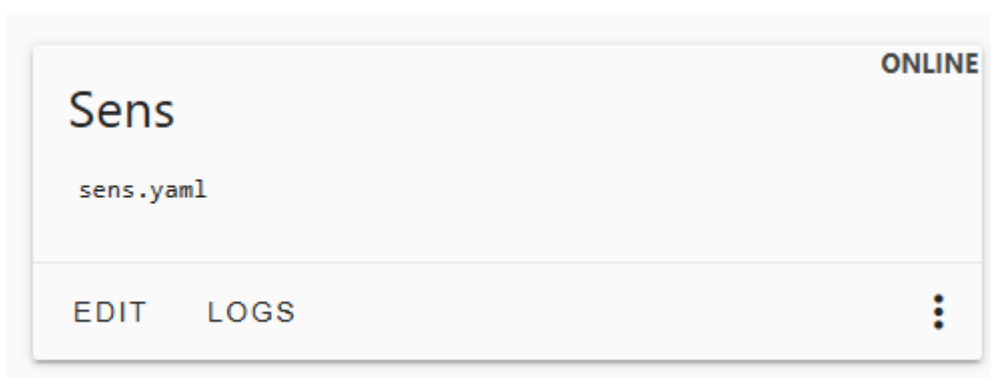


Рисунок 3.1 – Пристрій в ESPHome

На рисунку 3.2 наведено конфігурацію YAML для налаштування сенсора температури та вологості DHT 11, який підключається до вивода піна 4. Температура та вологість записується у відповідні змінні та оновлюються кожні 20 с.

```

captive_portal:
sensor:
  - platform: dht
    pin: 4
    temperature:
      name: "Living Room Temperature"
    humidity:
      name: "Living Room Humidity"
    update_interval: 20s

```

Рисунок 3.2 – Конфігурація сенсора температури та вологості DHT 11

Також потрібно налаштувати цифрові виводи для керування освітленням та вентиляцією відповідна конфігурація приведена на рисунку 3.3. Слід відмітити що конфігурація для перемикачів освітлення аналогічна тому на рисунку наведено тільки конфігурація для вмикання вентиляції

```

switch:
  - platform: gpio
    pin: GPIO 5
    name: "Вентиляція"

```

Рисунок 3.3 – Конфігурація перемикача вентиляції

Після прошивки контролера отримуємо лог (рис. 3.4) з значеннями сенсорів в терміналі, отже конфігурація працює і можна продовжувати створювати візуальні об'єкти на їх основі

```

[01:37:54][D][dht:049]: Got Temperature=23.7°C Humidity=53.0%
[01:37:54][D][sensor:094]: 'Living Room Temperature': Sending state 23.70000 °C with 1 decimals of accuracy
[01:37:54][D][sensor:094]: 'Living Room Humidity': Sending state 53.00000 % with 0 decimals of accuracy
[01:38:14][D][dht:049]: Got Temperature=23.7°C Humidity=53.0%
[01:38:14][D][sensor:094]: 'Living Room Temperature': Sending state 23.70000 °C with 1 decimals of accuracy
[01:38:14][D][sensor:094]: 'Living Room Humidity': Sending state 53.00000 % with 0 decimals of accuracy

```

Рисунок 3.4 – Лог в терміналі конфігуратора

На головному екрані одержуємо створені сутності для керування елементами розумного будинку, які прив'язані до пристрою (рис. 3.5)

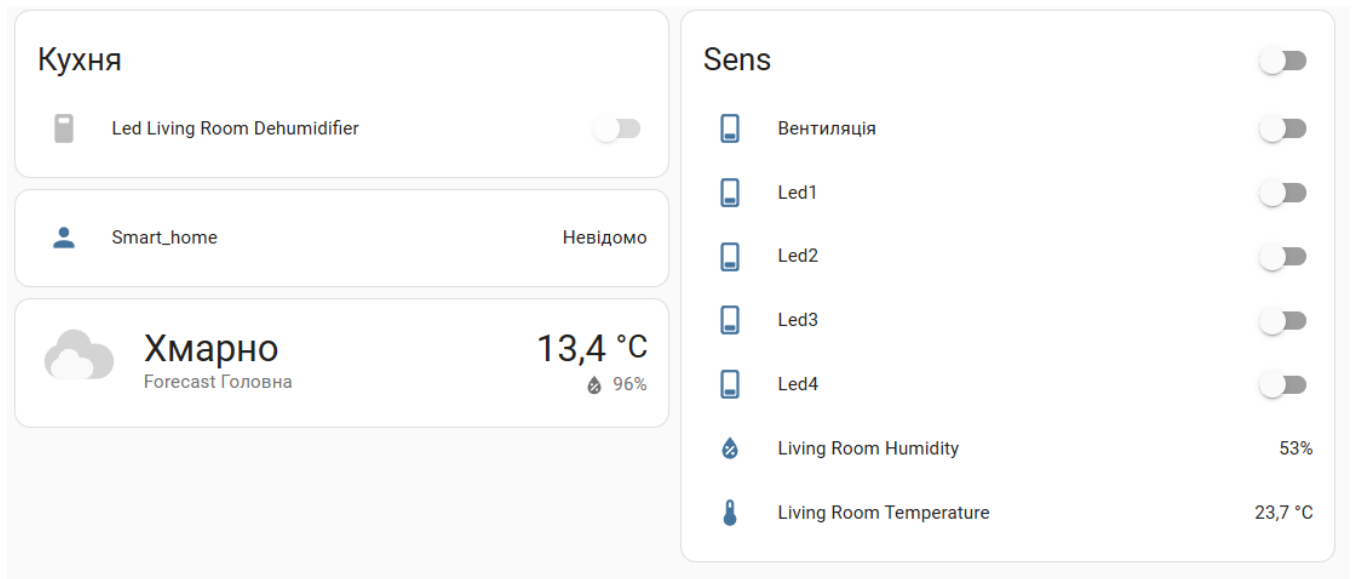


Рисунок 3.5 – Графічні елементи керування та моніторингу на головному вікні

Використовуючи інструменти налаштування інформаційних панелей створена панель, яка зображена на рисунку 3.6

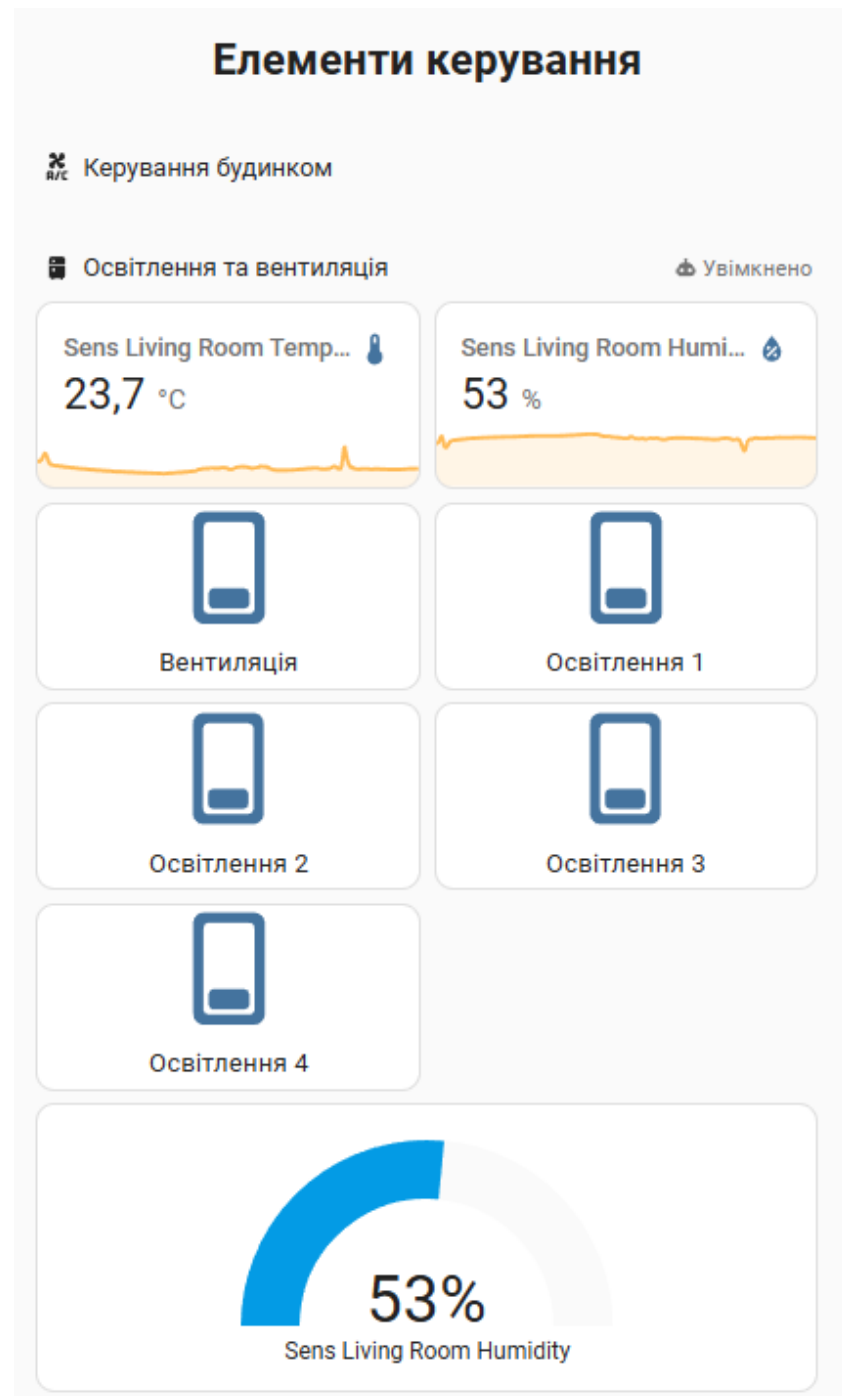


Рисунок 3.6 - Інформаційна панель керування пристроями розумного будинку

Використовуючи план будинку рисунку 3.7 здійснимо конфігурування розташування та характеристик елементів розумного будинку за допомогою відповідної YAML конфігурації (рис. 3.8)



Рисунок 3.7 – Розташування елементів розумного будинку на плані

✕ Конфігурація картки Елементи зображення



```

1  type: picture-elements
2  elements:
3  - type: state-icon
4    style:
5      left: 35%
6      top: 30%
7    entity: switch.sens
8    icon: mdi:fan
9    title: Вентиляція
10   state_color: true
11 - type: state-icon
12   style:
13     left: 20%
14     top: 25%
15   entity: switch.sens_led1
16   icon: mdi:chandelier
17 - type: state-icon
18   style:
19     left: 50%
20     top: 30%
21   entity: switch.sens_led2
22   icon: mdi:ceiling-light-outline
23 - type: state-icon
24   style:
25     left: 50%
26     top: 80%
27   entity: switch.sens_led3
28   icon: mdi:candelabra-fire
29 - type: state-icon
30   style:
31     left: 15%
32     top: 80%
```

ВІЗУАЛЬНИЙ РЕДАКТОР

СКАСУВАТИ ЗБЕРЕГТИ

Рисунок 3.8 – Конфігурування графічних елементів на плані

Home Assistant дозволяє збирати дані з різних сенсорів, таких як датчики температури, вологості, руху, вуглекислого газу, рівня освітленості, споживання енергії і багато інших. Ці дані можуть зберігатися протягом тривалого періоду, що дозволяє аналізувати зміни показників у часі. Це надзвичайно корисно для оцінки ефективності енергоспоживання, виявлення аномалій у роботі систем або для моніторингу змін в умовах середовища.

Однією з основних можливостей платформи є створення графіків та історичних даних. Наприклад, користувач може створювати графіки, що відображають зміни температури, вологості чи споживання енергії протягом дня, тижня або місяця. Ці графіки дозволяють не лише оцінити ефективність роботи різних систем, але й виявити тренди та аномалії, що можуть свідчити про потребу

в коригуванні налаштувань чи заміні обладнання. На рисунках 3.9 та 3.10 зображено графіки вологості та температури відповідно

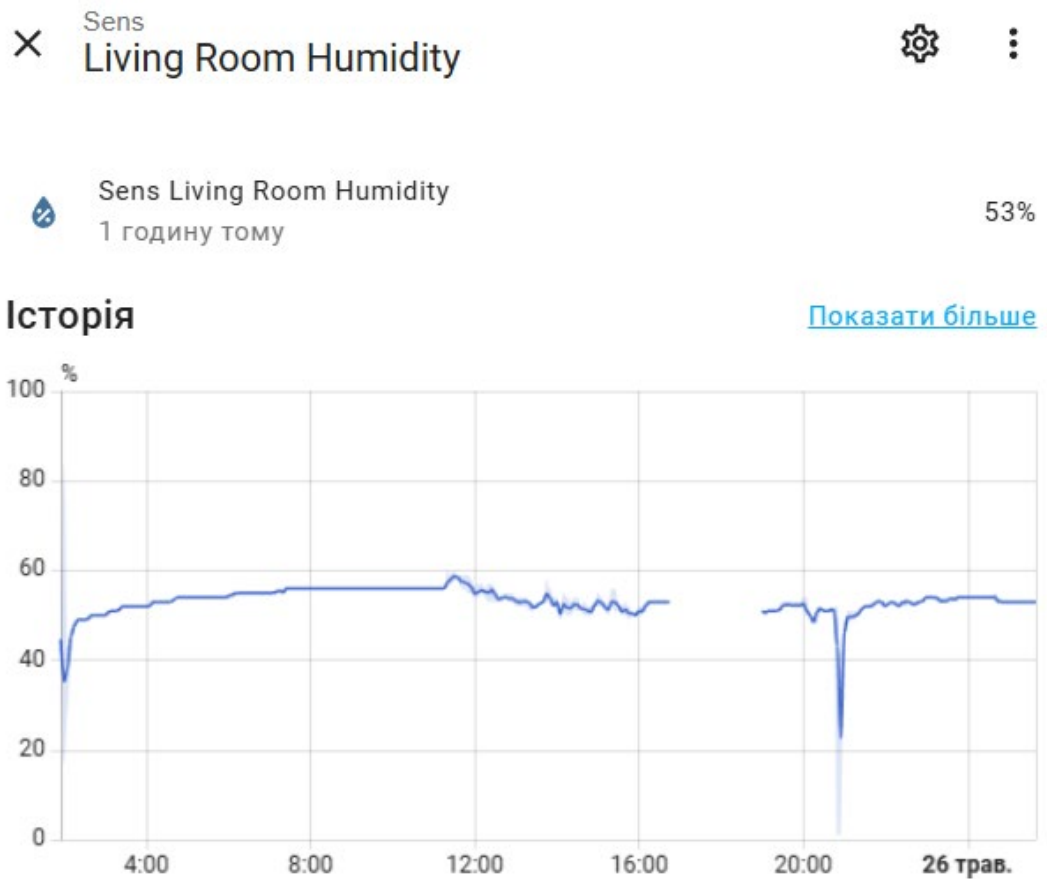


Рисунок 3.9 - Графік зміни вологості

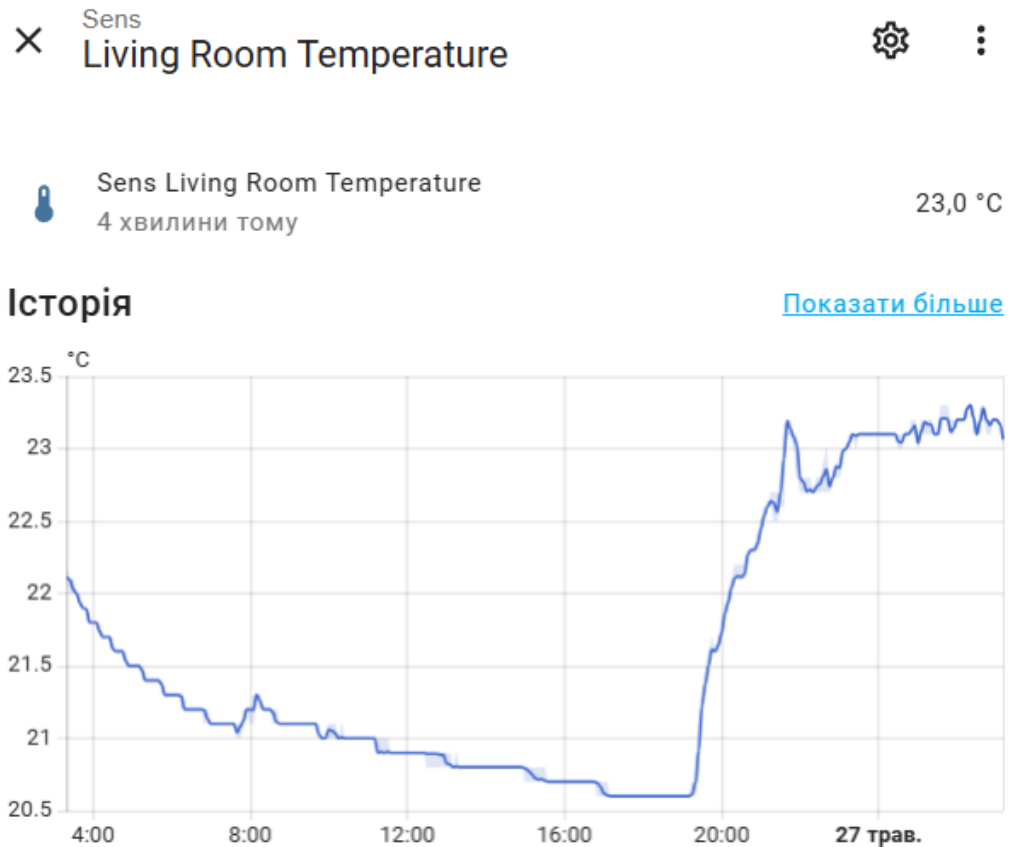


Рисунок 3.10 - Графік зміни температури

Ще однією потужною функцією є можливість використання додаткових інтеграцій для більш глибокого аналізу. Home Assistant має можливість інтегрувати сторонні сервіси та платформи для більш складного аналізу даних. Це може включати інтеграцію з платформами для аналізу даних, такими як Grafana або InfluxDB, що дозволяють створювати більш детальні візуалізації, зберігати дані в більш зручному форматі для аналізу, а також застосовувати розширені алгоритми для прогнозування чи виявлення аномалій.

Окрім того, Home Assistant підтримує використання алгоритмів машинного навчання для автоматизації та покращення роботи систем. Наприклад, за допомогою інтеграцій з платформами, що підтримують машинне навчання, можна створювати моделі для прогнозування температури, оптимізації використання енергії чи виявлення аномальних змін в параметрах. Це дозволяє системі адаптуватися до змін і самостійно приймати рішення щодо налаштувань або реакції на зміни.

Однією з цікавіших можливостей є також використання даних для створення звітів та отримання сповіщень. Користувач може налаштувати отримання сповіщень про зміни в показниках (наприклад, надмірне підвищення температури або зниження рівня вологості), що дозволяє оперативно реагувати на проблеми. Дані також можуть бути використані для генерування щотижневих або щомісячних звітів, що дозволяють оцінити ефективність роботи системи та виявити можливі зони для покращення.

3.3 Створення автоматизації в Home Assistant

Створення автоматизації в Home Assistant дозволяє налаштувати дії, які виконуються автоматично на основі заданих умов. Спершу потрібно визначити, що саме ви хочете автоматизувати. Це може бути включення світла, коли датчик руху фіксує рух, або вимкнення кондиціонера, коли температура в кімнаті досягає певного рівня.

Щоб створити автоматизацію, потрібно зайти в налаштування Home Assistant і вибрати розділ автоматизацій. Потім натискаєте "Додати автоматизацію". Home Assistant надає два варіанти: використовувати вбудований майстер для створення автоматизації або вручну написати YAML код для детальнішого налаштування. Зазвичай починають із порожньої автоматизації, щоб налаштувати її з нуля. На рисунку 3.11 зображено ресурси пристрою в інтеграції ESPHome. Виділено елементи керування та сенсори.

Sens

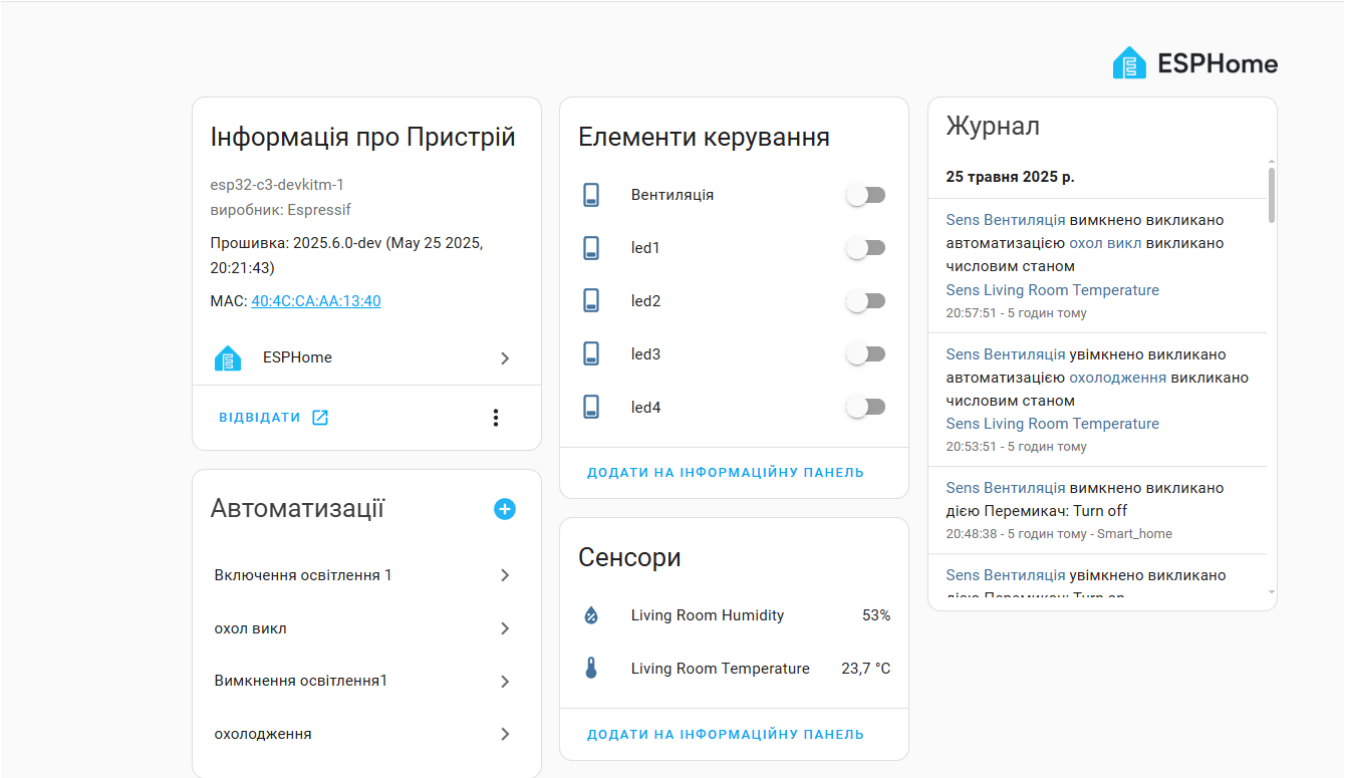


Рисунок 3.11 – Ресурси пристрою в інтеграції ESPHome

Першим кроком є налаштування тригера приклад якого наведено для системи вентиляції (рис. 3.12).

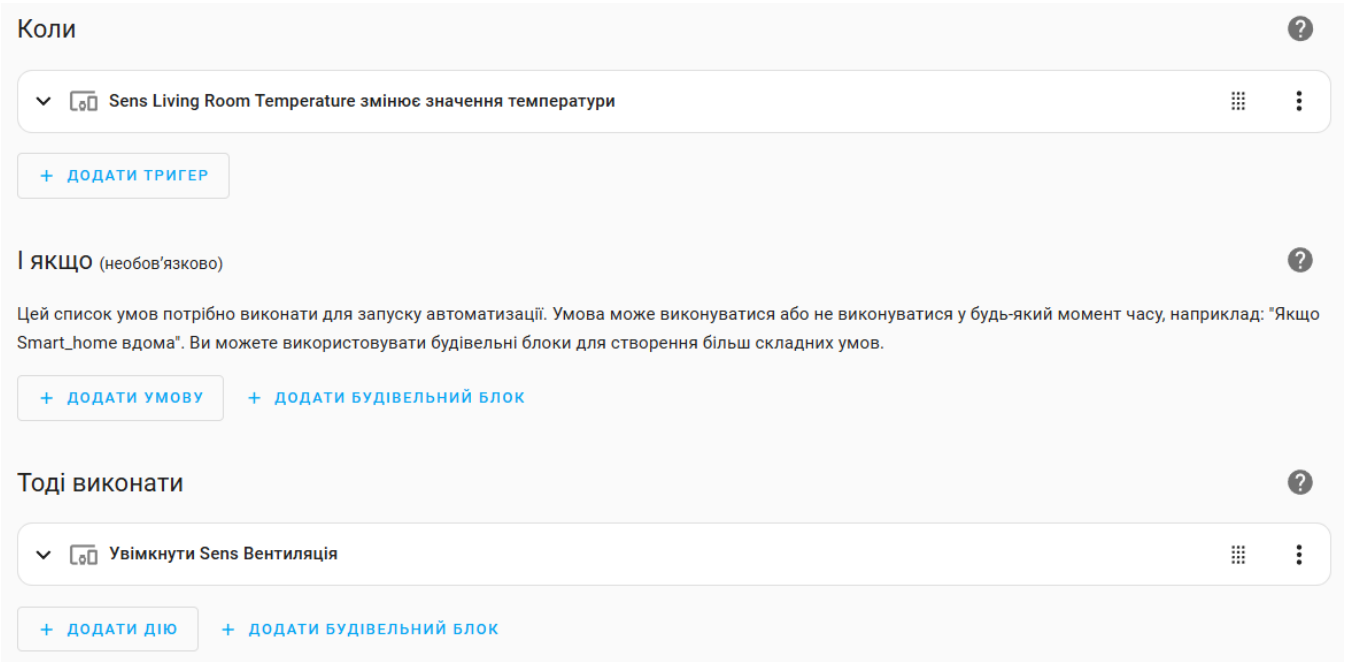


Рисунок 3.12 – Налаштування тригера

Тригер — це умова, при виконанні якої автоматизація буде спрацьовувати. Це може бути зміна стану пристрою (наприклад, якщо датчик руху фіксує рух або якщо температура досягає певного значення), часова умова (наприклад, автоматизація спрацьовує кожного дня о певній годині) або подія, яка має статися.

Наприклад, можна налаштувати тригер, щоб він спрацьовував, коли температура виміряна датчиком перевищує 30 градусів.

Наступним кроком є налаштування умов. Умова — це додатковий критерій, який визначає, чи буде автоматизація виконана, якщо тригер спрацює. Наприклад, можна встановити умову, що автоматизація спрацює тільки тоді, коли певне пристрій, наприклад, світло, вимкнене. Умова може бути за часом або на основі стану пристрою. Це дозволяє більш точно контролювати, коли автоматизація повинна бути активована.

Після цього настає черга дії, яка визначає, що саме відбудеться після того, як тригер спрацює, і умови виконанні. Дія може бути різною: включення або вимкнення пристроїв, відправка сповіщень на телефон, запуск сценаріїв або інших автоматизацій. Наприклад, можна налаштувати дію, щоб включити світло в коридорі, коли датчик руху фіксує рух. Дії можуть бути дуже різноманітні, в залежності від потреб.

Після налаштування тригера, умови і дії, зберігаємо автоматизацію і тестуємо її. Можна вручну запустити автоматизацію або спровокувати тригер, щоб переконатися, що автоматизація працює правильно. Якщо все налаштовано правильно, автоматизація буде виконуватися автоматично за умовами, які були зконфігуровано. Якщо потрібно, можна повернутися до налаштувань і змінити автоматизацію, якщо вона не працює так, очікувалосьь.

Таким чином, процес створення автоматизації включає визначення тригерів, умов та дій, які разом дозволяють налаштувати потрібний сценарій в системі Home Assistant. Розроблено алгоритм автоматизації освітлення та вентиляції (рис. 3.13)

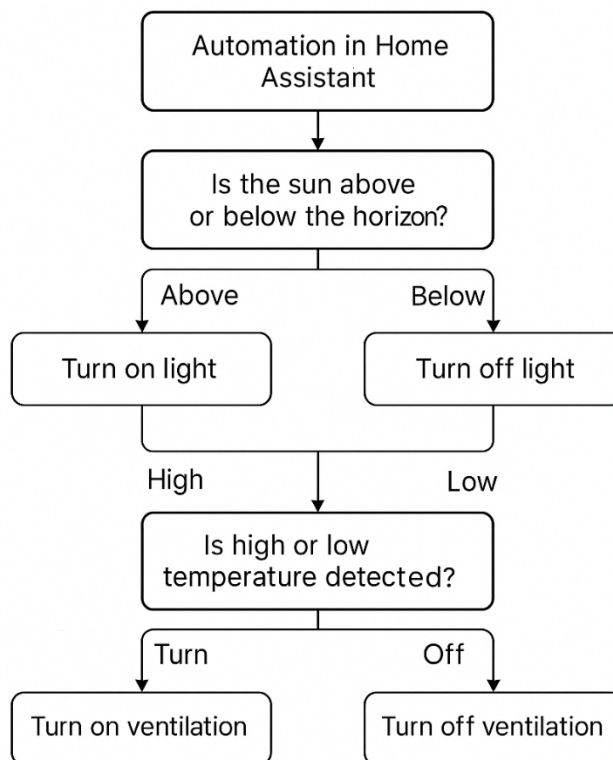


Рисунок 3.13 – Блок-схема алгоритму автоматизації роботи освітлення та вентиляції

Алгоритм автоматизації в Home Assistant складається з двох основних умов: одна керує освітленням, а інша — вентиляцією в залежності від температури. Спочатку перевіряється, чи зайшло сонце — якщо так, увімкнеться світло. Якщо сонце ще не зайшло, то світло вимикається. Окрім того, вентиляція активується при високій температурі та вимикається при низькій температурі, щоб підтримувати комфортний мікроклімат у приміщенні. Ця автоматизація допомагає зменшити енергоспоживання та підтримувати оптимальні умови для життя.

На рисунку 3.14 зображено список автоматизацій проєкту, передбачено автоматичне керування температурою бойлера, керування зовнішнім освітленням, а також вентиляцією на кухні.

	Ввімкнення бойлера
	Вимкнення бойлера
	Вимкнення освітлення1
	Включення освітлення 1
	охол викл
	охолодження

Рисунок 3.14 - Список автоматизацій «розумного будинку»

3.4 Висновки

В результаті виконання налаштувань інтерфейсу користувача в системі Home Assistant із використанням візуального редактора Lovelace було досягнуто гнучкого та зручного відображення елементів керування пристроями розумного будинку. Реалізовано конфігурацію карток різного типу, що дозволило представити інформацію про стан сенсорів, освітлення, кліматичні параметри та інші пристрої у зрозумілому для користувача вигляді.

Інтерфейс отримав структуровану побудову — з можливістю виведення на окремі панелі даних по кімнатах, групах пристроїв та сценаріях автоматизації. Створення карток типу Entities, Glance, History Graph забезпечило можливість як перегляду даних у реальному часі, так і історичних змін параметрів.

Використання YAML-конфігурацій розширило функціональність візуалізації, дозволивши точніше налаштувати стилі, кольори, шрифти, логіку дій при натисканні та інтерактивні сценарії. Реалізація кнопок для управління

пристроями, додавання сповіщень, енергографіків та можливість інтерактивного плану приміщень із клікабельними зонами зробили інтерфейс не лише функціональним, а й інтуїтивно зрозумілим.

4 РОЗШИРЕННЯ ФУНКЦІОНАЛУ ІНТЕРФЕЙСУ СИСТЕМИ «РОЗУМНИЙ БУДИНОК»

4.1 Різновиди інформаційних панелей в Lovelace

Крім базового функціоналу системи створимо додаткові вкладки на бічній панелі з використанням інструменту «інформаційні панелі» для швидкого переходу на інтерфейс окремих кімнат (рис. 4.1).

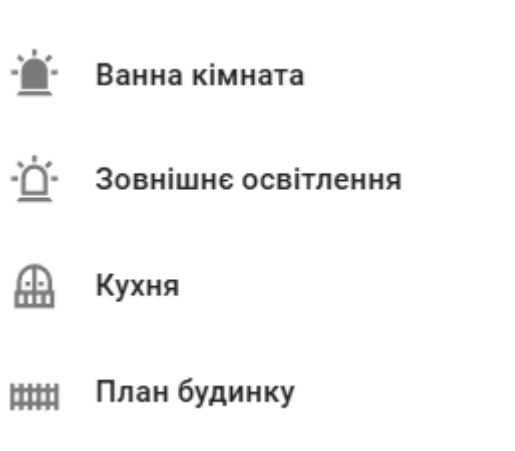


Рисунок 4.1 – Бічна панель для переходу до інтерфейсу окремих кімнат

В системі Home Assistant кожна карта (card) в інтерфейсі Lovelace представляє собою автономний програмно-конфігурований компонент, який поєднує в собі візуалізацію та взаємодію із сутностями домашньої автоматизації. Щоб описати багатоманіття можливостей, картки можна категоризувати за функціональним призначенням. Навіть сам лише перелік безлічі типів карток є індикатором того, наскільки гнучким і модульним є інтерфейс Lovelace. На рисунку 4.2 наведено інтерфейс налаштування інформаційних карток які додаються на панель керування.

× Додати на інформаційну панель

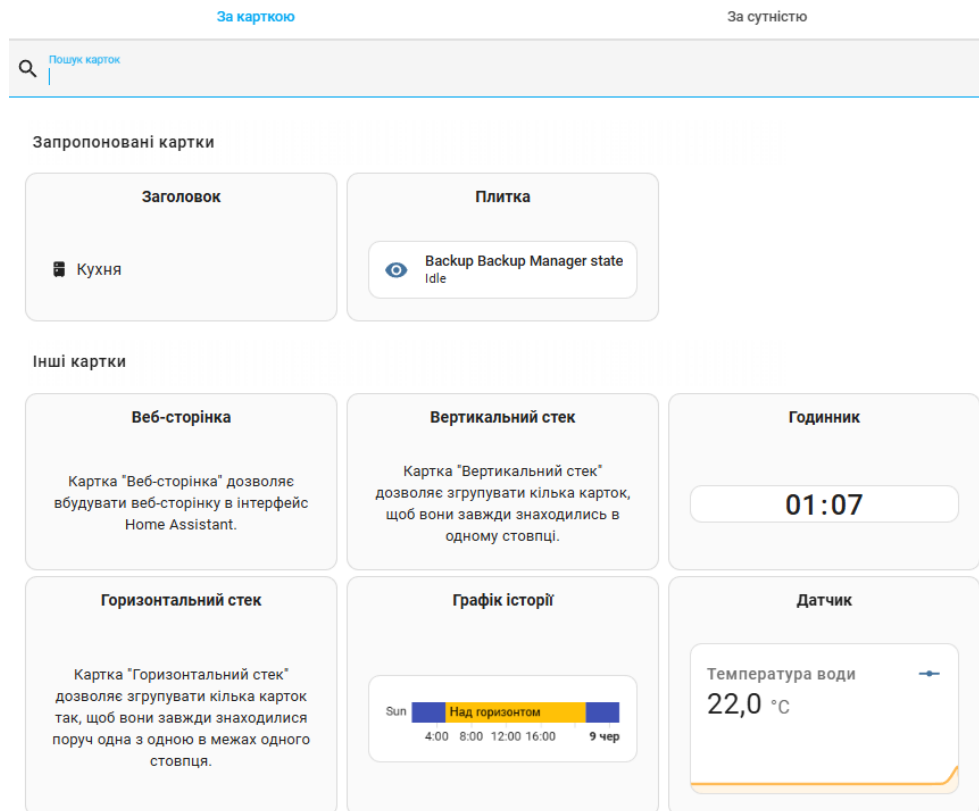


Рисунок 4.2 - Вибір інформаційної панелі

Картки специфічних пристроїв та сервісів охоплюють картки, орієнтовані під певні види інтегрованих пристроїв чи служб. Наприклад, карту Alarm panel використовують для керування сигнальною системою та переходу між режимами охорони (arm_home, arm_away, arm_night), а її конфігурація включає можливість ввести PIN-код та змінювати режими home-assistant.io. Для керування світлом доступна картка типу light, яка виводить тумблери та яскравість або колір; humidifier — для керування зволожувачами; thermostat — для налаштування температурного режиму через колесо або слайдер; media control — для відтворення музики чи звуку на медіапристроях; plant status — для моніторингу рівня вологості рослин; weather forecast — для показу погоди; calendar — для перегляду подій; map — для візуалізації геозон і трекерів. Для кожного з них передбачена оптимізована панель з елементами керування, налаштованими на певну сутність та контекст.

Карти для відображення даних у вигляді графіки чи чисел, наприклад Sensor (або entity) налаштована на вивід одного показника — наприклад, поточної температури або вологості. Набір таких карток формує «інформаційні блоки» на дашборді. Для перегляду трендів використовують History graph, що виводить дані сенсора за певний проміжок часу (до восьми сутностей одночасно). Якщо ж потрібно показати статистичні показники — середнє, мінімум, максимум — обирають Statistics graph. Найкоротший шлях до розуміння порогових станів — Gauge, який відображає значення у вигляді кругової чи смугової шкали з кольоровими зонами в залежності від заданих меж.

Найбільш гнучкі варіанти комбінованих карток для роботи із зображеннями - Picture entity, Picture glance, Picture Elements, а також Area. У Picture entity картка містить зображення сутності, як-от камери, яке може змінюватися залежно від стану (живий стрім або статичне зображення). Picture glance дозволяє накласти невеличкі іконки на зображення, кожна з яких асоціюється із сутністю та може виконувати toggle або відкривати інформаційний діалог home-assistant.io. Найпотужніша серед них — Picture Elements: вона дозволяє позиціонувати по координатах state badges, іконки, текстові лейбли, графічні кнопки (action buttons), інші зображення, умови показу та кастомні елементи home-assistant.io. Це ідеальна база для створення інтерактивного плану дому, де на основі SVG-файла можна клікнути на ліжко або люстру й отримати відповідний інтерфейс.

Організаційні картки призначені для структуризації дашбордів важливими є стекові картки: Vertical Stack, Horizontal Stack, Grid. Вони дозволяють згрупувати кілька карток у одному просторі — розташувати поруч або підряд. Крім того, для оформлення вкладки використовуються групування, де можна формувати блоки з заголовками, умовами видимості та адаптивним розміщенням. Самі розділи формують структуру як модульні блоки: наприклад, розділ «Клімат» може містити картку thermostat, для поточної температури та history graph для тренду.

4.2 Розробка графічного інтерфейсу користувача для окремих кімнат

Для створення інтуїтивно зрозумілого інтерфейсу необхідно крім елементів керування на плані будинку мати окремі інтерфейси для кімнат, які об'єднують пристрої та сутності, що територіально розміщені в певному приміщенні. На рисунку 4.3 наведено інтерфейс керування для ванної кімнати

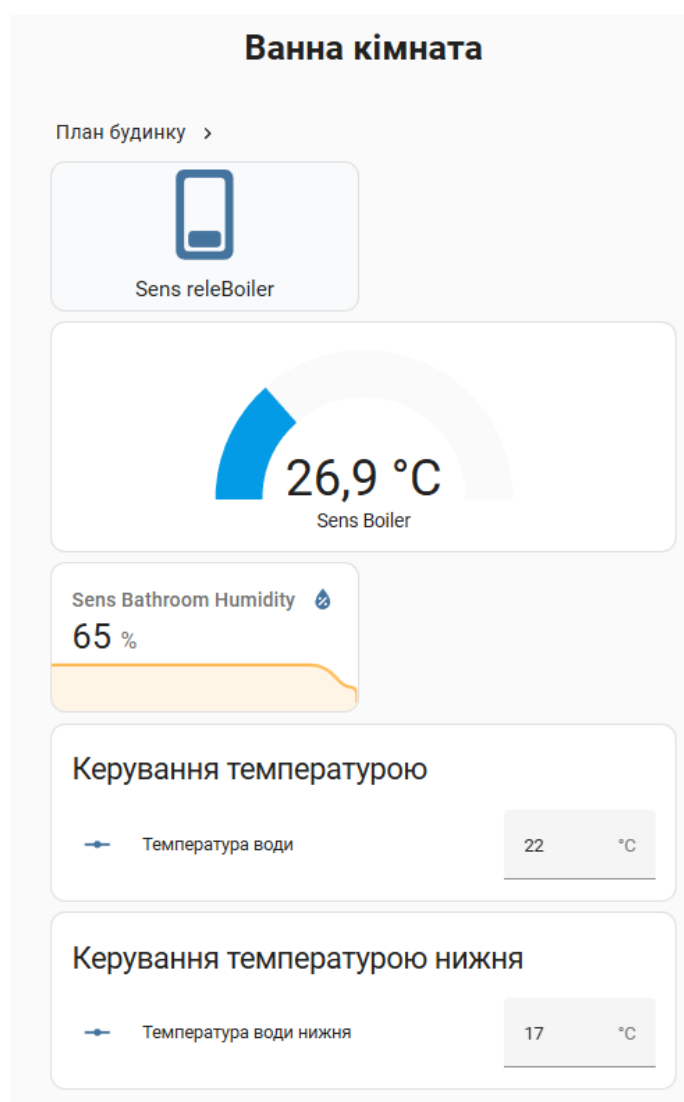


Рисунок 4.3 – Інтерфейс керування елементами «розумного будинку» у ванній кімнаті

Для побудови були використані картки кнопок, виводу цифрового значення для температури води бойлера та вологості в кімнаті, також

зконфігуровано елементи задання порогів температур для керування роботою бойлера. Лістинг конфігураційного файлу з прописаними елементами введення температур наведено на рисунку 4.4.

```
# Loads default set of integrations. Do not remove.
default_config:
-
# Load frontend themes from the themes folder
frontend:
- themes: !include_dir_merge_named themes
-
automation: !include automations.yaml
script: !include scripts.yaml
scene: !include scenes.yaml
-
input_number:
- target_temperature_living_room:
... name: Температура води
... min: 10
... max: 60
... step: 0.5
... unit_of_measurement: "°C"
... mode: box # або "slider"
- target_temperature_living_below:
... name: Температура води нижня
... min: 10
... max: 60
... step: 0.5
... unit_of_measurement: "°C"
... mode: box # або "slider"
```

Рисунок 4.4 - Лістинг конфігураційного файлу

На рисунку 4.5 наведено інтерфейс керування для системи зовнішнього освітлення, яке вмикається після заходу сонця.

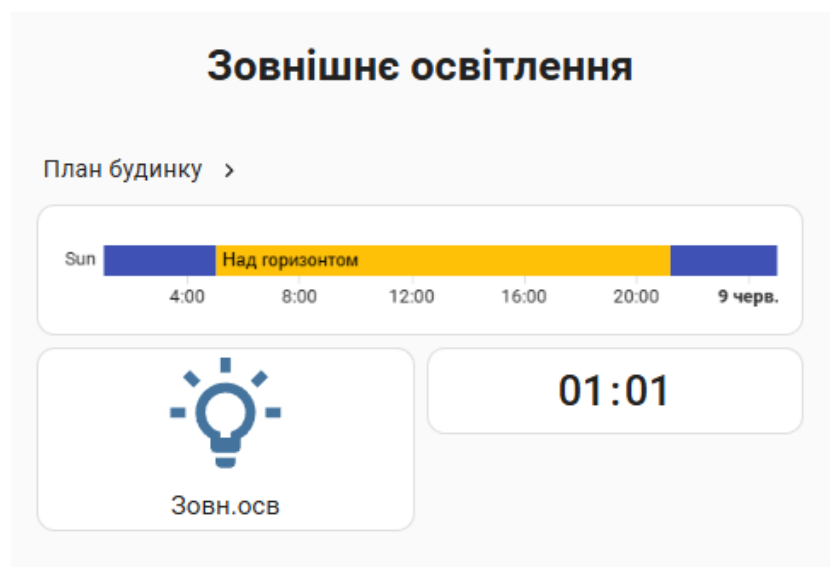


Рисунок 4.5 – Інтерфейс керування зовнішнім освітленням

Для побудови інтерфейсу використані картки стану сутності, положення сонця та годинника.

На рисунку 4.6 наведено інтерфейс керування елементами «розумного будинку» на кухні. Для побудови були використані картки кнопок, виводу цифрового значення для температури та вологості на кухні, також зконфігуровано елементи автоматизації вентиляції.

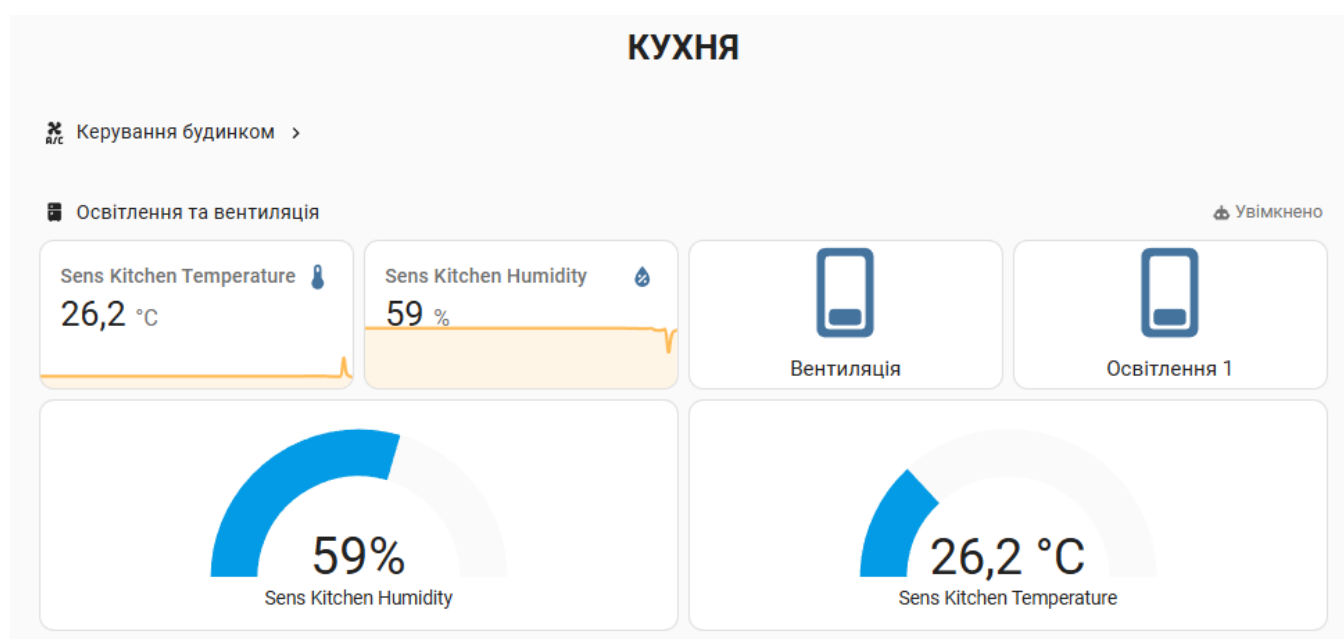


Рисунок 4.6 – Інтерфейс керування елементами «розумного будинку» на кухні

Для збільшеної кількості виконавчих пристроїв та сенсорів розширимо конфігурацію ESPHome (рис. 4.7-4.8).

```

sensor:
- platform: dht
  pin: 4
  temperature:
    name: "Kitchen Temperature"
  humidity:
    name: "Kitchen Humidity"
  update_interval: 20s
- platform: dht
  pin: 3
  temperature:
    name: "Boiler"
  humidity:
    name: "Bathroom Humidity"
  update_interval: 20s

```

Рисунок 4.7 – Конфігурація сенсорів

```

# Example configuration entry
switch:
- platform: gpio
  pin: GPIO 5
  name: "Вентиляція"
- platform: gpio
  pin: GPIO 6
  name: "led1"
- platform: gpio
  pin: GPIO 7
  name: "led2"
- platform: gpio
  pin: GPIO 8
  name: "led3"
- platform: gpio
  pin: GPIO 9
  name: "led4"
- platform: gpio
  pin: GPIO 10
  name: "led5"
- platform: gpio
  pin: GPIO 18
  name: "releBoiler"

```

Рисунок 4.8 – Конфігурація перемикачів

В результаті одержуємо вигляд інтерфейсу керування будинку з додатковими компонентами (рис. 4.9)

× Конфігурація картки Елементи зображення



Конфігурація

Видимість

Макет

Параметри картки

Елементи

Значок стану

Вентиляція

×

Значок стану

switch.sens_led1

×

Значок стану

switch.sens_led2

×

Значок стану

switch.sens_led3

×

Значок стану

switch.sens_led4

×

Бейдж стану

вологість

×

Бейдж стану

sensor.sens_kitchen_temperature

×

Бейдж стану

sensor.sens_boiler

×

Значок стану

switch.sens_releboiler

×

Значок стану

switch.sens_led5

×

Додати новий елемент

Рисунок 4.9 – Конфігурування графічних елементів на плані

4.3 Практична реалізація системи розумний будинок

На рисунку 4.10 наведено апаратну реалізацію системи «розумний будинок». Сервер розгорнуто на міні-комп'ютері MLLSE mini PC m2 air. Периферійні пристрої системи зібрано на макетній платі. До складу системи входять Wi-fi контролер ESP, сенсори температури та вологості DHT11 та набір світлодіодів, які підключено до відповідних виводів контролера, які відповідають конфігурації цифрових перемикачів. Такий підхід дозволяє швидко налаштовувати логіку та функціонал системи.

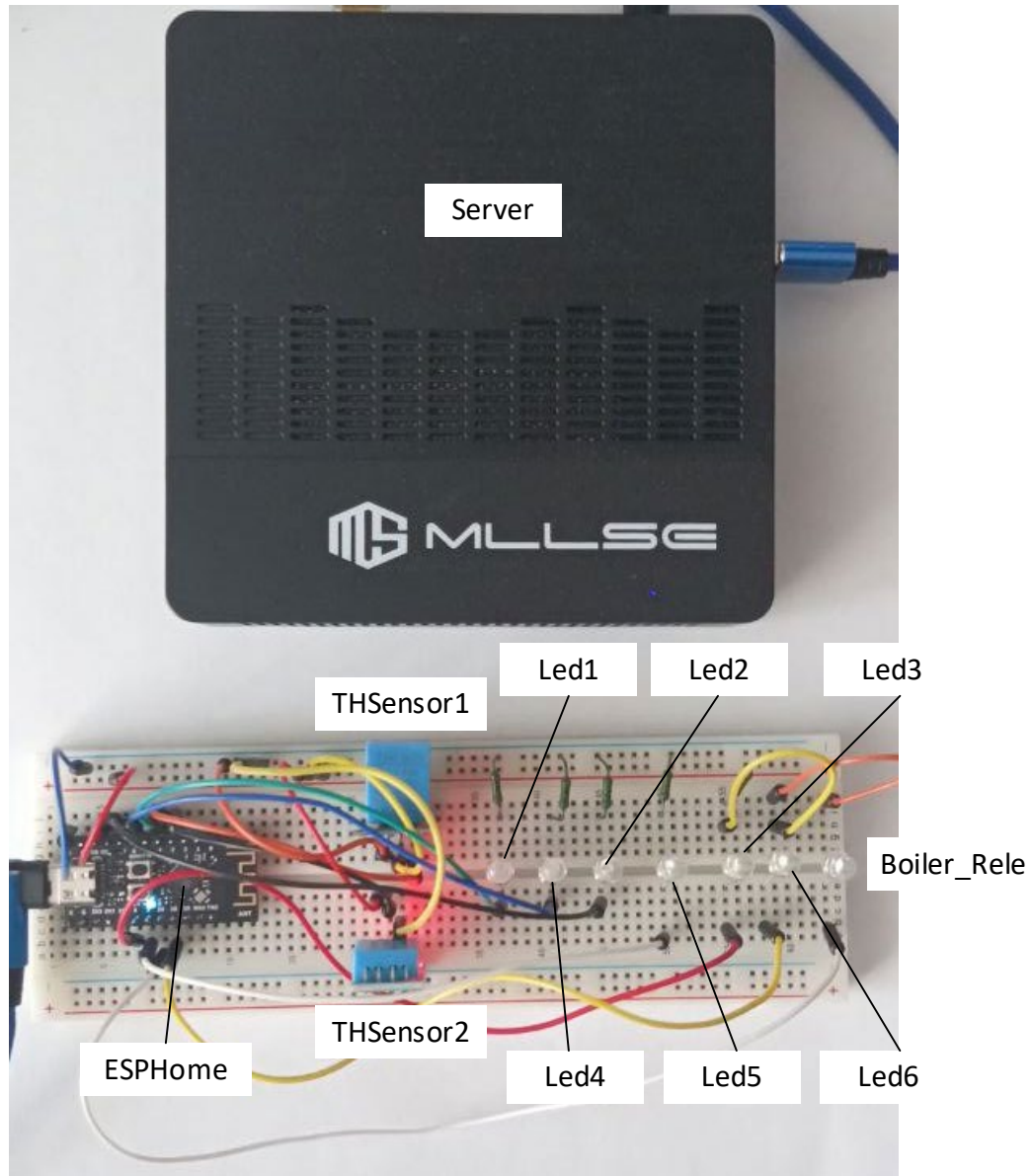


Рисунок 4.10 – Апаратна реалізація інформаційної системи «розумний будинок» у вигляді макету

Мобільний додаток Home Assistant є невід’ємною частиною екосистеми розумного будинку, забезпечуючи користувачам повний контроль над пристроями та автоматизаціями без необхідності доступу до веб-інтерфейсу через браузер. Цей застосунок доступний для Android та iOS і має низку функцій, оптимізованих саме для мобільних пристроїв, що робить його зручним, швидким та інформативним інструментом щоденного використання.

Однією з ключових переваг мобільного додатку є адаптивний інтерфейс, який автоматично підлаштовується під розмір екрану смартфона або планшета. Головна панель відображає Lovelace-інтерфейс (інформаційні картки, кімнати, іконки пристроїв) так само, як і веб-версія, проте з урахуванням особливостей сенсорного управління — наприклад, свайпи для перемикання між вкладками, зручні великі кнопки для активації сценаріїв або перемикачів, а також підтримка темної та світлої тем.

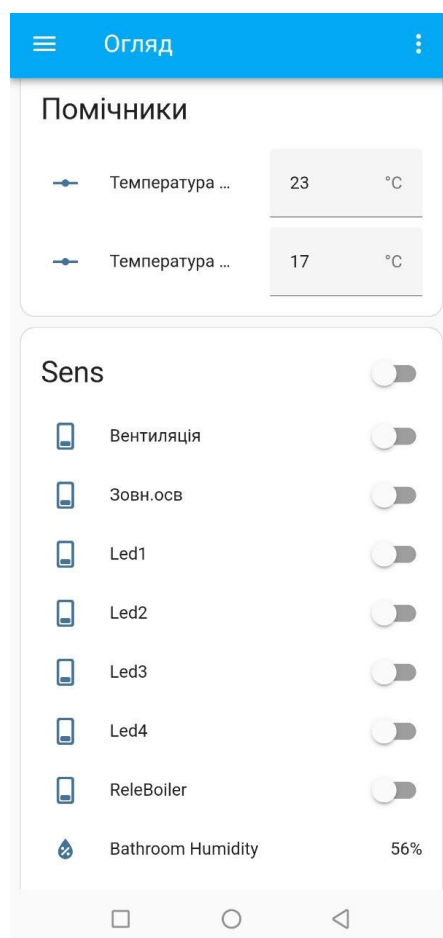


Рисунок 4.11 – Вигляд фрагменту головного екрану в мобільному додатку
Home Assistant

Особливу увагу у мобільному застосунку приділено системі сповіщень. Користувач може отримувати push-сповіщення про події в реальному часі, включаючи: виявлення руху, відкриття дверей, перевищення температури або спрацювання сигналізації. На відміну від десктопної версії, мобільний додаток дозволяє не тільки отримувати текстові повідомлення, але й реагувати на них

безпосередньо з екрана блокування, використовуючи інтерактивні кнопки. Наприклад, користувач може відразу з сповіщення увімкнути світло або відкрити двері.

Крім того, додаток дозволяє використовувати датчики самого телефону як інтегровані компоненти Home Assistant. Це означає, що геолокація користувача, рівень заряду акумулятора, стан з'єднання Wi-Fi, швидкість руху та інші сенсорні дані можуть бути використані в автоматизаціях. Наприклад, система може автоматично вимикати освітлення чи знижувати температуру, коли власник виходить з дому — завдяки інтеграції з GPS.

Серед додаткових функцій мобільного застосунку варто виділити підтримку віджетів (на Android), які дозволяють керувати обраними пристроями прямо з головного екрана смартфона, та інтеграцію з системними командами Siri або Google Assistant для голосового управління.

Загалом, мобільний додаток Home Assistant не є просто доповненням до основної системи — це повноцінний інструмент керування, який має низку унікальних можливостей, відсутніх у десктопній версії, і дозволяє реалізовувати концепцію «розумного будинку в кишені», забезпечуючи користувачеві постійний, мобільний та інтуїтивно зрозумілий доступ до всіх функцій автоматизованої системи.

4.4 Висновки

Здійснено практичне розширення функціональних можливостей інтерфейсу користувача системи «розумний будинок», реалізованої на базі Home Assistant. Запропоновано та впроваджено низку структурних і візуальних удосконалень, які покращують взаємодію з системою, підвищують зручність доступу до пристроїв та забезпечують більш гнучке управління побутовими процесами.

Було проаналізовано можливості системи Lovelace щодо налаштування інформаційних панелей та розглянуто класифікацію карток за функціональним

призначенням. Наведено приклади використання спеціалізованих карток для керування вентиляцією, освітленням, температурними режимами, моніторингом мікроклімату, а також карток для графічного аналізу та візуалізації даних у режимі реального часу. Особливу увагу приділено гнучким графічним карткам типу Picture Elements, які дозволяють будувати інтерактивні плани приміщень із розміщенням елементів керування.

У межах розробки інтерфейсів для окремих кімнат було створено індивідуальні вкладки, що відповідають ванній кімнаті, кухні, зовнішньому освітленню тощо. Кожна вкладка містить оптимізований набір керуючих елементів, сенсорних віджетів і налаштувань автоматизації, що значно спрощує навігацію та підвищує зрозумілість у повсякденному використанні.

Практична реалізація конфігурації сенсорів, перемикачів та сценаріїв керування здійснювалася з використанням платформи ESPHome та контролера ESP32. Це забезпечило гнучке підключення виконавчих пристроїв, зокрема сенсорів DHT11 та світлодіодів, до серверної частини системи, розгорнутої на компактному енергоефективному комп'ютері MLLSE mini PC m2 air. Такий підхід довів свою ефективність в умовах макетної реалізації, дозволяючи швидко адаптувати конфігурацію до зміни потреб.

Окремо було досліджено функціонал мобільного додатку Home Assistant, який продемонстрував високу ефективність в управлінні пристроями завдяки адаптивному інтерфейсу, інтеграції з сенсорами смартфона, push-сповіщенням у реальному часі та підтримці голосового управління. Його використання дозволяє повністю реалізувати концепцію мобільного доступу до системи, що є критично важливим у контексті сучасних тенденцій автоматизації.

ВИСНОВКИ

В ході виконання бакалаврської кваліфікаційної роботи було здійснено комплексне дослідження теоретичних і практичних аспектів побудови інтерфейсу користувача для системи «розумний будинок» на базі Home Assistant з використанням Lovelace UI. У результаті реалізовано функціональний, адаптивний та зручний у користуванні інтерфейс, який забезпечує ефективну взаємодію користувача з інтегрованими пристроями та сервісами домашньої автоматизації.

На основі аналізу сучасних архітектур та принципів організації інтерфейсів було встановлено, що Lovelace надає широкі можливості для кастомізації, інтеграції сенсорних, мультимедійних, охоронних та кліматичних систем. Визначено оптимальну структуру інтерфейсу, яка охоплює: головну панель з інтерактивною картою приміщень, окремі вкладки для кімнат і пристроїв, а також елементи керування сценаріями, сповіщеннями та безпекою. Особливу увагу приділено адаптації під різні типи пристроїв (настільні ПК, планшети, смартфони), забезпеченню інтуїтивної навігації, використанню інформативних карток та візуалізації реального часу.

Результатом проєкту є локальна клієнт-серверна система з можливістю безперервного моніторингу та керування компонентами розумного будинку без залежності від хмарних сервісів. Розроблений інтерфейс пройшов тестування за ключовими критеріями: зручність користування, швидкодія, стабільність роботи, адаптивність. Проведена оцінка показала високий рівень задоволеності користувачів і довела ефективність обраного підходу.

Таким чином, у роботі реалізовано усі поставлені задачі: від аналізу теоретичних засад та вибору архітектури - до безпосередньої реалізації, тестування та оцінки створеного інтерфейсу. Отримані результати можуть бути основою для подальшого розвитку системи - додавання нових функцій, інтеграції зі сторонніми сервісами, розширення сценаріїв автоматизації та впровадження голосового керування.

Таким чином, реалізоване розширення функціоналу інтерфейсу системи «розумний будинок» дозволяє підвищити її зручність, адаптивність та інтерактивність. Отримані результати можуть бути використані як основа для подальшого масштабування системи, включення нових типів пристроїв, а також впровадження складніших сценаріїв автоматизації в реальних умовах експлуатації.

Результати роботи передано для впровадження в Науково-дослідну лабораторію автоматизованих систем управління в енергетиці та транспорті (НДЛ АСУЕТ) кафедри САІТ ВНТУ.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Максимнюк М. В., Проценко Д. П., Розроблення інтерфейсу користувача в інформаційній системі «розумний будинок» на базі Home Assistant, матеріали Міжнародної науково-практичної інтернет-конференції Молодь в науці: дослідження, проблеми, перспективи (МН-2025). URL: <https://conferences.vntu.edu.ua/index.php/mn/mn2025/paper/viewFile/25294/21160>.
2. B. Zhang, P. L. P. Rau, and G. Salvendy, Design and evaluation of smart home user interface: Effects of age, tasks and intelligence level, *International Journal of Human–Computer Interaction*, vol. 25, no. 5, pp. 415–428, 2009. URL: <https://doi.org/10.1080/10447310902894132> (дата звернення: 28.05.2025).
3. F. Panico, V. De Maio, M. D'Ascoli, and R. Tedesco, Study on Smart Home Interface Design Characteristics Considering Age Differences, *Frontiers in Psychology*, vol. 13, Article 825238, 2022. URL: <https://doi.org/10.3389/fpsyg.2022.825238> (дата звернення: 28.05.2025).
4. S. Davidoff, M. K. Lee, C. Yiu, J. Zimmerman, and A. K. Dey, Principles of Smart Home Control, *Proceedings of UbiComp 2006: Ubiquitous Computing*, Berlin: Springer, pp. 19–34, 2006. URL: https://doi.org/10.1007/11853565_2 (дата звернення: 28.05.2025).
5. X. Bai, X. Wang, Y. Wang, and L. Xu, A Study on Smart Home Interface Design in Age-friendly Scenarios, *BMC Geriatrics*, vol. 22, no. 1, pp. 1–11, 2022. URL: <https://doi.org/10.1186/s12877-022-02843-3> (дата звернення: 28.05.2025).
6. Home Assistant, "Dashboard Chapter 1: Introducing Dashboards, 2024. URL: <https://www.home-assistant.io/lovelace/dashboards/> (дата звернення: 28.05.2025).
7. T. Cerquitelli, D. Giordano, and M. Viviani, Dashboarding IoT data for smart home management: A case study, *2021 IEEE Int. Conf. on Smart Computing (SMARTCOMP)*, pp. 243–248, 2021. URL: <https://doi.org/10.1109/SMARTCOMP52413.2021.00047> (дата звернення: 28.05.2025).

8. A. Moro, A. Valenzano, and S. Orlando, Home automation interfaces: A comparative evaluation of dashboard designs, *J. of Ambient Intelligence and Humanized Computing*, vol. 11, no. 10, pp. 4067–4081, 2020. URL: <https://doi.org/10.1007/s12652-019-01541-6> (дата звернення: 28.05.2025).
9. C. Buratti, C. Contoli, and R. Verdone, Design and evaluation of smart home dashboards based on user feedback, *Sensors*, vol. 22, no. 9, Article 3451, 2022. URL: <https://doi.org/10.3390/s22093451> (дата звернення: 28.05.2025).
10. I. Koren and M. Gvozdjak, Responsive UI design principles for smart environments: Implementation in Home Assistant, *ACM Trans. on Internet Technology*, vol. 21, no. 3, pp. 1–25, 2021. URL: <https://doi.org/10.1145/3451892> (дата звернення: 28.05.2025).
11. Home Assistant, Lovelace Dashboards. URL: <https://www.home-assistant.io/lovelace/> (дата звернення: 28.05.2025).
12. ESPHome, Smart Home Automation for ESP8266/ESP32. URL: <https://esphome.io/> (дата звернення: 28.05.2025).
13. Home Assistant Documentation, "Automation and Scripts." URL: <https://www.home-assistant.io/docs/automation/> (дата звернення: 28.05.2025).
14. N. Larson, Smart Home Automation with ESPHome and Home Assistant, *DigiKey Maker.io*, 2021. URL: <https://www.digikey.com/en/maker/projects/smart-home-automation/> (дата звернення: 28.05.2025).
15. S. M. H. Anik, X. Gao, H. Zhong, X. Wang, and N. Meng, Automation Configuration in Smart Home Systems: Challenges and Opportunities, *arXiv*, 2024. URL: <https://arxiv.org/abs/2408.00000> (дата звернення: 28.05.2025).
16. S. Marksteiner, V. J. Expósito Jiménez, H. Vallant, and H. Zeiner, An Overview of Wireless IoT Protocol Security in the Smart Home Domain, *arXiv*, 2018. URL: <https://arxiv.org/abs/1801.06500> (дата звернення: 28.05.2025).
17. D. Bouchabou, S. M. Nguyen, C. Lohr, B. Leduc, and I. Kanellos, A Survey of Human Activity Recognition in Smart Homes Based on IoT Sensors Algorithms, *arXiv*, 2021. URL: <https://arxiv.org/abs/2110.09311> (дата звернення: 28.05.2025).

18. F. Hall, L. Maglaras, T. Aivaliotis, L. Xagoraris, and I. Kantzavelou, Smart Homes: Security Challenges and Privacy Concerns, *arXiv*, 2020. URL: <https://arxiv.org/abs/2010.12321> (дата звернення: 28.05.2025).
19. M. Alhaisoni, F. Alhaidari, K. Alsubhi, and F. Alshehri, Designing Smart Home Dashboards for Enhanced User Experience: A Usability Study, *J. of Intelligent & Fuzzy Systems*, vol. 45, no. 2, pp. 1941–1951, 2023. URL: <https://doi.org/10.3233/JIFS-223220> (дата звернення: 28.05.2025).
20. S. Ghosh, S. Jain, and S. Kapoor, Evaluating User-Centric Design in Home Automation Interfaces: A Cognitive Load Perspective, *Human-Centric Computing and Information Sciences*, vol. 12, no. 1, Article 15, 2022. URL: <https://doi.org/10.1186/s13673-022-00277-3> (дата звернення: 28.05.2025).

Додаток А
(обов'язковий)

Міністерство освіти і науки України
Вінницький національний технічний університет
Факультет інтелектуальних інформаційних технологій та автоматизації

ЗАТВЕРДЖУЮ

Завідувач кафедри САІТ

_____ д.т.н., проф. Віталій МОКІН

« ____ » _____ 2025 р.

ТЕХНІЧНЕ ЗАВДАННЯ

на бакалаврську  кваліфікаційну роботу

ІНФОРМАЦІЙНА СИСТЕМА «РОЗУМНИЙ БУДИНОК» НА БАЗІ
ЛОКАЛЬНОЇ КЛІЄНТ-СЕРВЕРНОЇ АРХІТЕКТУРИ. РОЗРОБЛЕННЯ
ІНТЕРФЕЙСУ КОРИСТУВАЧА

08-34.БКР.018.00.000 ТЗ

Керівник: к.т.н., доц.

_____ Дмитро ПРОЦЕНКО

« __ » _____ 2025 р.

Розробив студент гр. 2ІСТ-216

_____ Макар МАКСИМНЮК

« __ » _____ 2025 р.

1. Підстава для проведення робіт.

Підставою для виконання роботи є наказ №__ по ВНТУ від «__» _____2025р., та індивідуальне завдання на БКР, затверджене протоколом №__ засідання кафедри САІТ від «__» _____ 2025р.

2. Джерела розробки:

1) Home Assistant Dashboards, інформаційний ресурс. URL: <https://www.home-assistant.io/dashboards/>;

2) Сервіс Home Assistant, інформаційний ресурс URL: <https://www.home-assistant.io/>.

3. Мета і призначення роботи.

Метою дослідження є підвищення ефективності функціонування інформаційної системи «розумний будинок» шляхом розроблення її на базі локальної клієнт-серверної архітектури та створення зручного й адаптивного інтерфейсу користувача.

4. Вихідні дані для проведення робіт: сенсори для вимірювання параметрів середовища, мікропроцесорна плата ESP32, система Home Assistant.

5. Методи дослідження:

Використання конфігуратора YAML, здійснення аналізу науково-технічної літератури, порівняльного аналізу інтерфейсних рішень, моделювання, проектування користувацького інтерфейсу, а також експериментального тестування системи в умовах реального використання.

Етапи роботи і терміни їх виконання:

а) Загальна характеристика об'єкту досліджень _____ – _____

б) Вибір ІТ-інфраструктури _____ – _____

в) Налаштування інтерфейсу користувача _____ – _____

г) Розширення функціоналу інтерфейсу системи «розумний будинок» _____ – _____

д) Практична реалізація ІоТ-системи _____ – _____

е) Оформлення матеріалів до захисту БКР _____ – _____

7. Очікувані результати та порядок реалізації

Інтерфейс інформаційної системи «розумний будинок».

8. Вимоги до розробленої документації

Текстова та ілюстративна частини роботи оформлені у відповідності до вимог «Методичних вказівок до виконання бакалаврських кваліфікаційних робіт для студентів спеціальностей: 124 «Системний аналіз», 126 «Інформаційні системи та технології» (освітня програма «Прикладні інформаційні технології»).

9. Порядок приймання роботи

Публічний захист «__» _____ 2025 р.

Початок розробки «__» _____ 2025 р.

Граничні терміни виконання БКР «__» _____ 2025 р.

Розробив студент групи 2ІСТ-216 _____ Макар МАКСИМНЮК

Додаток Б
(обов'язковий)

ПРОТОКОЛ ПЕРЕВІРКИ КВАЛІФІКАЦІЙНОЇ РОБОТИ

Назва роботи: «Інформаційна система «розумний будинок» на базі локальної клієнт-серверної архітектури. Розроблення інтерфейсу користувача»

Тип роботи: бакалаврська кваліфікаційна робота

Підрозділ: кафедра САІТ, ФІТА, гр. 2ІСТ-216

Коефіцієнт подібності текстових запозичень, виявлених у роботі
системою StrikePlagiarism 0,76 %

Висновок щодо перевірки кваліфікаційної роботи (відмітити потрібне):

- ☒ Запозичення, виявлені у роботі, є законними і не містять ознак плагіату, фабрикації, фальсифікації. Роботу прийняти до захисту
- ☐ У роботі не виявлено ознак плагіату, фабрикації, фальсифікації, але надмірна кількість текстових запозичень та/або наявність типових розрахунків не дозволяють прийняти рішення про оригінальність та самостійність її виконання. Роботу направити на доопрацювання.
- ☐ У роботі виявлено ознаки плагіату та/або текстових маніпуляцій як спроб укриття плагіату, фабрикації, фальсифікації, що суперечить вимогам законодавства та нормам академічної доброчесності. Робота до захисту не приймається.

Експертна комісія:

Віталій МОКІН, зав. каф. САІТ

_____ (підпис)

Євгеній КРИЖАНОВСЬКИЙ, доц. каф. САІТ

_____ (підпис)

Особа, відповідальна за перевірку _____ (підпис)

Сергій ЖУКОВ

З висновком експертної комісії ознайомлений(-на)

Керівник _____ (підпис)

Дмитро ПРОЦЕНКО, к.т.н., доц. каф. САІТ

Здобувач _____ (підпис)

Макар МАКСИМНЮК

Додаток В
(обов'язковий)

ІЛЮСТРАТИВНА ЧАСТИНА

ІНФОРМАЦІЙНА СИСТЕМА «РОЗУМНИЙ БУДИНОК» НА БАЗІ
ЛОКАЛЬНОЇ КЛІЄНТ-СЕРВЕРНОЇ АРХІТЕКТУРИ. РОЗРОБЛЕННЯ
ІНТЕРФЕЙСУ КОРИСТУВАЧА

Нормоконтроль: к.т.н., доцент

_____ Сергій ЖУКОВ

«__» _____ 2025 р.



Рисунок В.1 – Критерії до вибору інструментів візуалізації

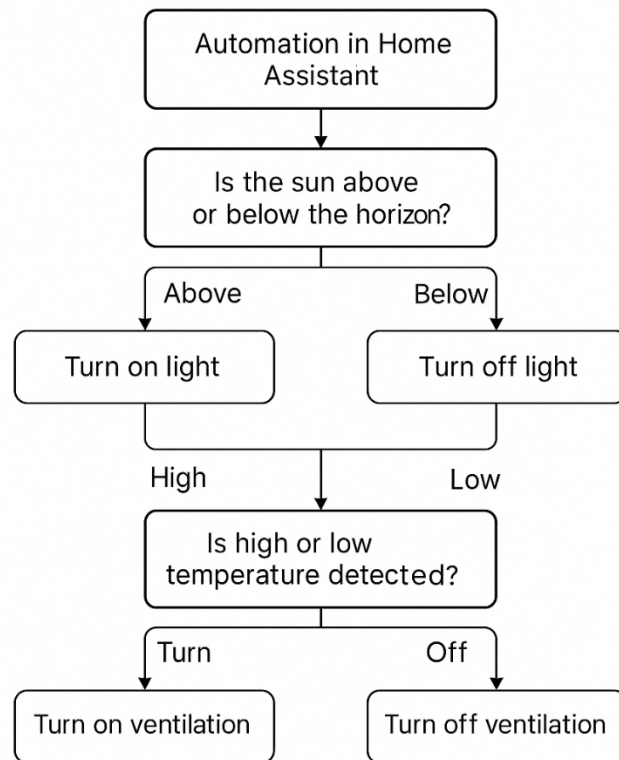


Рисунок В.2 - Блок-схема алгоритму автоматизації роботи освітлення та вентиляції

× Конфігурація картки Елементи зображення



Конфігурація Видимість Макет

Параметри картки ▾

Елементи

Значок стану
Вентиляція

×

Значок стану
switch.sens_led1

×

Значок стану
switch.sens_led2

×

Значок стану
switch.sens_led3

×

Значок стану
switch.sens_led4

×

Бейдж стану
вологість

×

Бейдж стану
sensor.sens_kitchen_temperature

×

Бейдж стану
sensor.sens_boiler

×

Значок стану
switch.sens_releboiler

×

Значок стану
switch.sens_led5

×

Додати новий елемент ▾

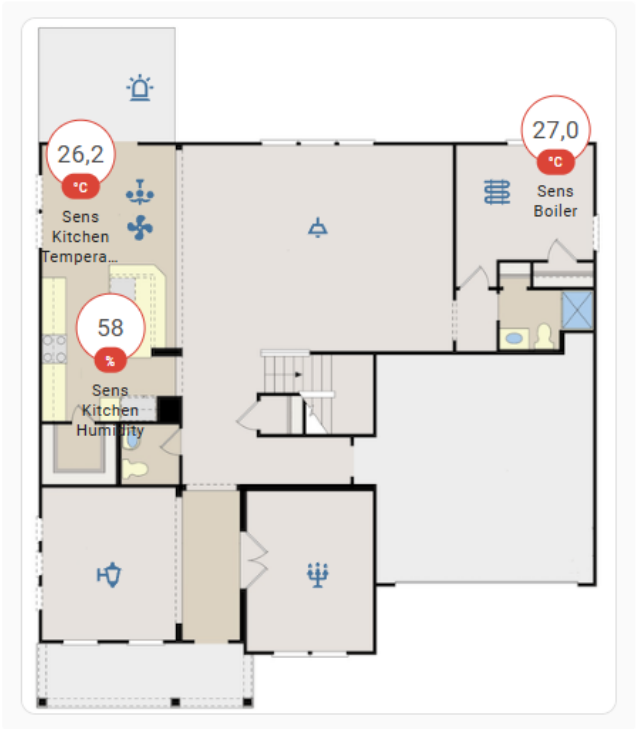


Рисунок В.3 - Конфігурування графічних елементів на плані

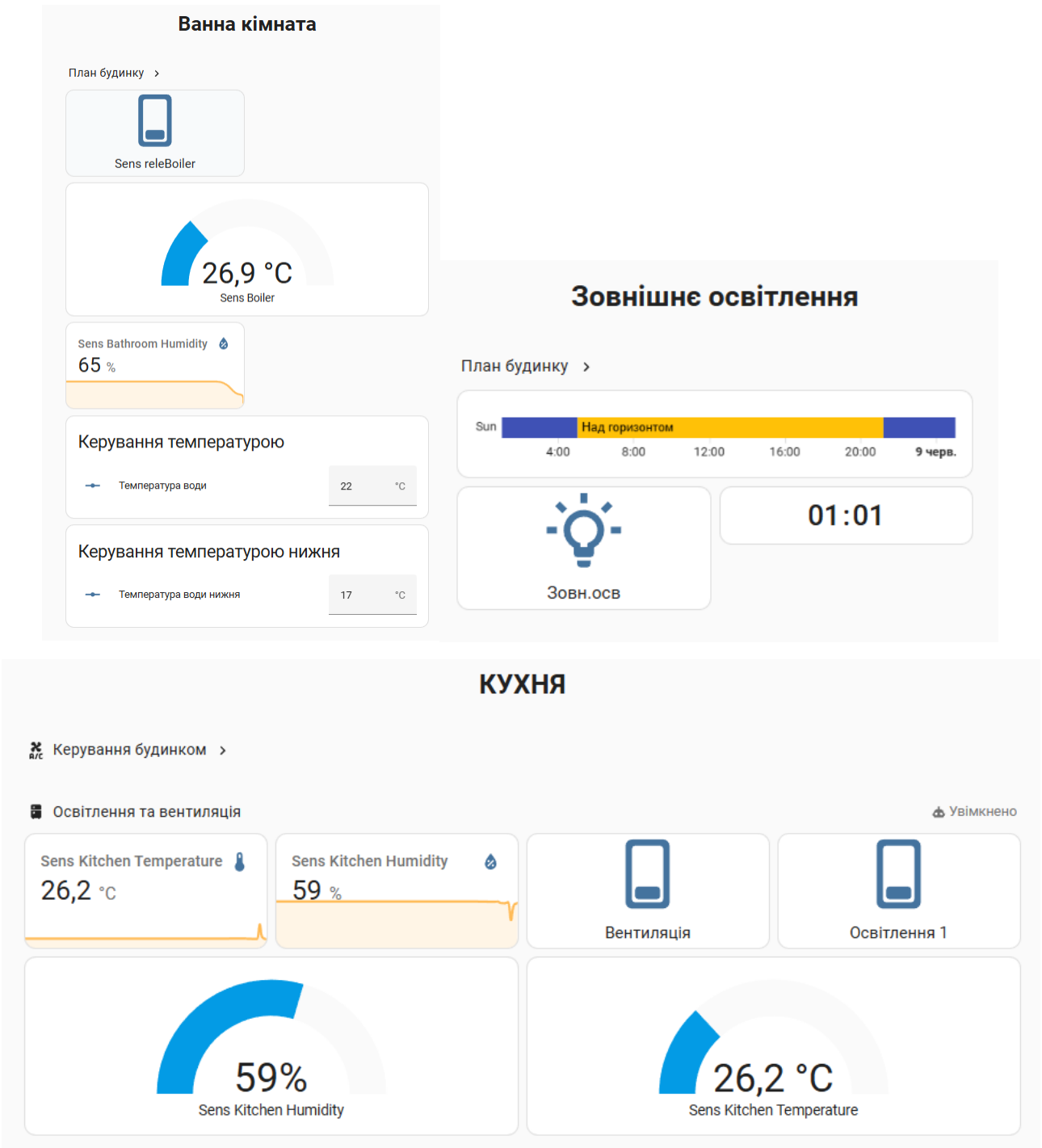


Рисунок В.4 – Розробка графічного інтерфейсу користувача для окремих кімнат

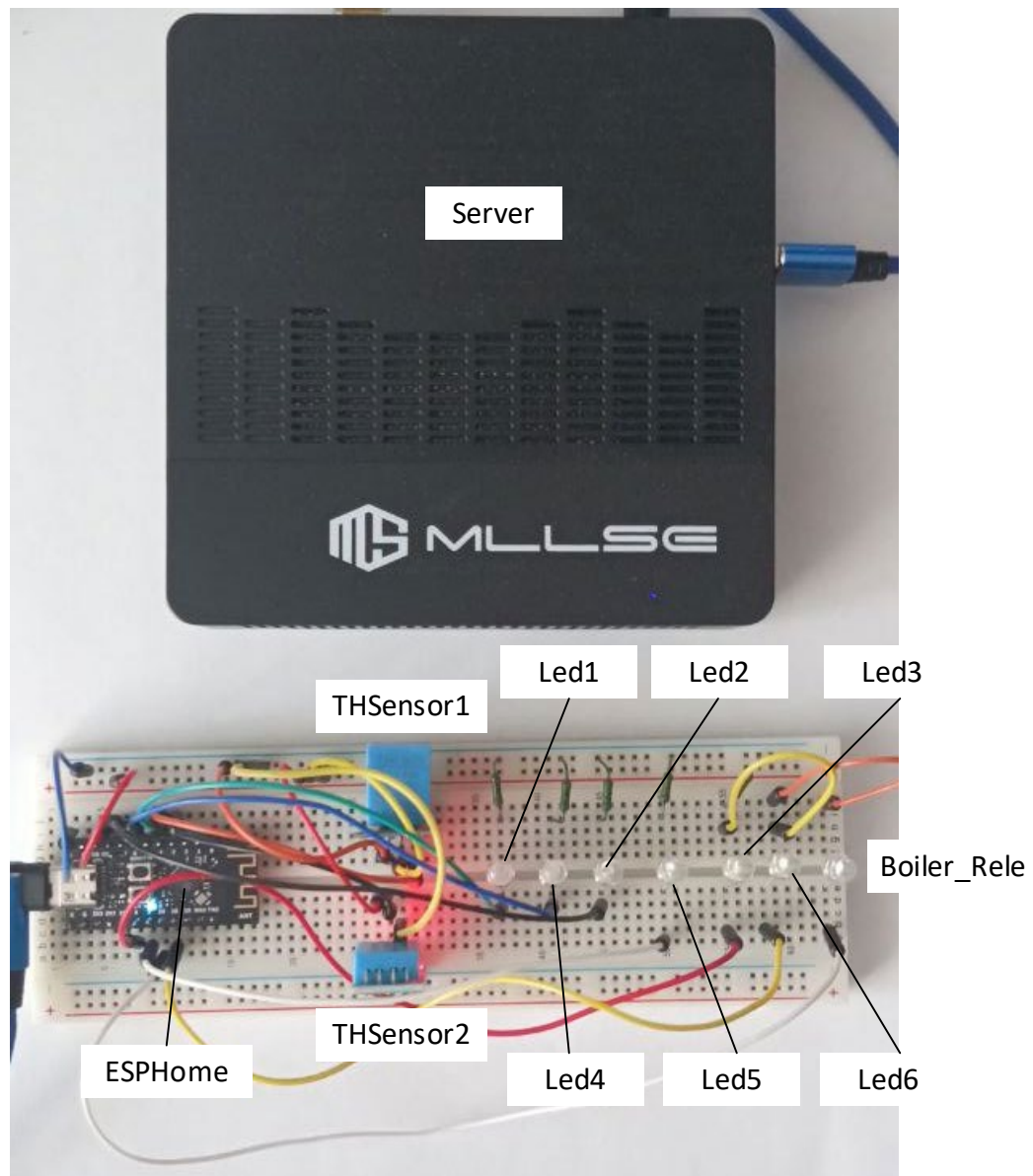


Рисунок В.5 - Апаратна реалізація інформаційної системи «розумний будинок»
у вигляді макету

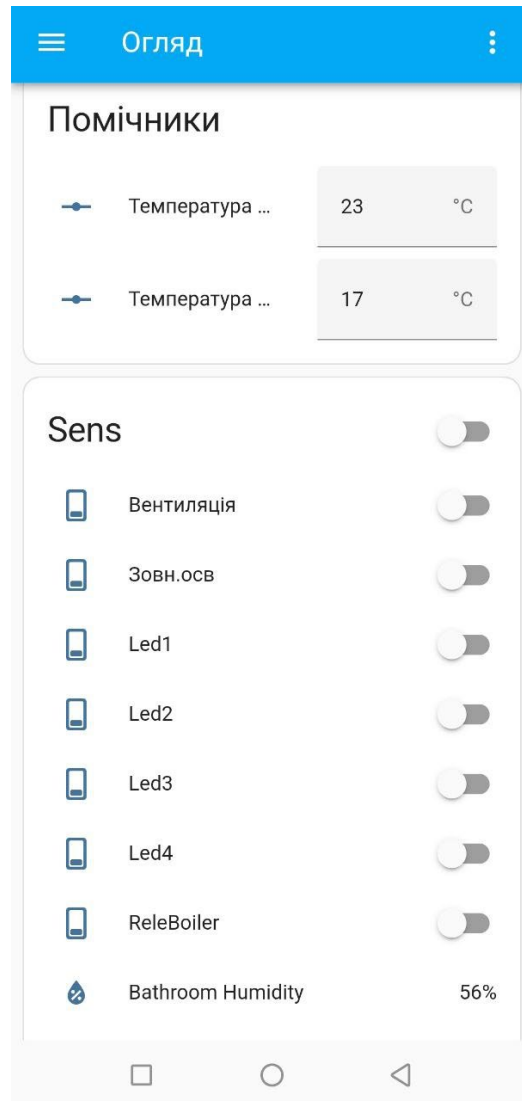


Рисунок В.6 – Вигляд фрагменту головного екрану в мобільному додатку
Home Assistant