

Вінницький національний технічний університет
Факультет комп'ютерних систем та автоматики
Кафедра Автоматизації та інтелектуальних інформаційних технологій

Пояснювальна записка
до магістерської кваліфікаційної роботи

на тему: «Розробка системи «Розумний будинок» на базі Raspberry Pi 3 із використанням модулів Arduino»

Виконав: студент 2 курсу, групи 1АКІТ-18м
спеціальності

151 – «Автоматизація та комп'ютерно-інтегровані технології»

(шифр і назва напрямку підготовки, спеціальності)

Орлюк Є. А.

(прізвище та ініціали)

Керівник к.т.н., доц. Богач І. В.

(прізвище та ініціали)

Рецензент .

(прізвище та ініціали)

Вінниця ВНТУ –2019 року

АНОТАЦІЯ

У даній магістерській роботі розроблено систему, що керує приватним будинком. Дана система має захищений вебсервер для керування системою через мережу Internet, вона є простою в установці та налаштуванні, модульна, інтелектуальна та із можливістю розширення в залежності від будинку. В її комплект входять: контролер який керує системою, набір датчиків (протікання, руху, освітлення, задимлення, температури та тиску та інші), WiFi-реле для уникнення підключення використовуючи кабелі, модуль GSM, RFiD модуль.

ANNOTATION

In this master's thesis the system that manages a private house is developed. This system has a secure web server for managing the system over the Internet, and it is easy to install and configure, modular, intelligent and expandable depending on the home. It includes: controller that controls the system, a set of sensors (leakage, movement, lighting, smoke, temperature and pressure, etc.), WiFi relay to avoid connection using cables, GSM module, RFiD module.

ЗМІСТ

ВСТУП.....	6
1 ДОСЛІДЖЕННЯ ОСНОВНИХ ПОНЯТЬ ПРЕДМЕТНОЇ ГАЛУЗІ ТА ОГЛЯД АНАЛОГІВ	9
1.1 Історія розвитку поняття «розумний будинок»	9
1.2 Основні функції розумного будинку	10
1.3 Огляд існуючих систем “Розумний будинок”	12
1.3.1 Система “Smart Bus”	12
1.3.2 Система розумного будинку “Кластер”	12
1.3.3 Система для управління будинком “CLAP”	14
2 ПРОЕКТУВАННЯ СИСТЕМИ «РОЗУМНИЙ БУДИНОК»	16
2.1 Характеристики комп’ютеру RaspberryPi 3 B+.....	16
2.2 Переваги платформи Raspberry Pi для розумового будинку	17
2.3 Компоненти розумного будинку	18
2.4 Система управління розумним будинком	20
3 РОЗРОБКА КОМПОНЕНТІВ СИСТЕМИ.....	24
3.1 Компонент електронного замка з використанням RFID модулю “RC522”	24
3.2 Підготовка WiFi реле до роботи в системі	26
3.3 Розробка інтерфейсу для адміністратора системи.....	29
3.4 Розробка модулю захисту системи.....	32
3.5 Розробка системи управління будинком	35
3.6 Сигналізація для будинку із СМС оповіщенням	38
4 ІНТЕГРАЦІЯ АПАРАТНОГО ТА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ТА ТЕСТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ СИСТЕМИ.....	41
4.1 Інтеграція апаратного та програмного забезпечення системи	41
4.1.1 Інтеграція модулів системи.....	41
4.1.2 Інтеграція програмного коду прошивки реле	43
4.2 Тестування програмного забезпечення системи.....	44

4.2.1 Тестування інтерфейсу для адміністратора системи	45
4.2.2 Тестування модулю захисту системи	48
4.2.3 Тестування модулю обробки запитів	51
5 ЕКОНОМІЧНИЙ РОЗДІЛ	53
5.1 Технологічний аудит розробленої системи «розумний будинок» на базі Raspberry Pi 3 із використанням модулів Arduino	53
5.2 Розрахунок витрат на виконання роботи.....	60
5.3 Розрахунок економічного ефекту від можливої комерціалізації даної розробки «розумний будинок»	64
ВИСНОВКИ.....	70
СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ	72
ДОДАТКИ.....	79
Додаток А (обов'язковий). Технічне завдання	80
Додаток Б (обов'язковий). Графічна частина	84
Додаток В (обов'язковий). Лістинг конфігурації вебсерверу	92
Додаток Г (обов'язковий). Лістинг вебінтерфейсу	93
Додаток І (обов'язковий). Лістинг модулю обробки запитів для перевірки реле	99

ВСТУП

Актуальність. На даний час все більш актуальним стає покращення якості житла та зменшення його вартості. Багато різних компаній розробляють різні розумні будинки для покращення та прискорення життя. Кожна з розробок має свою унікальність та використовує певні принципи автоматизації, але націлена та застосовується під конкретні потреби кінцевих споживачів.

В цілому автоматизація приватного будинку призводить до зменшення витрат часу господарем на обслуговування будинку та нагляд за ним, зменшення вартості комунальних послуг, підвищення зручності та надійності систем будинку. Показник захищеності будинку розраховується як імовірність виникнення кричної ситуації та ймовірність отримати збитки процесів експлуатації будинку [1-7].

Актуальною є розробка розумного будинку, який забезпечить надійність, прискорить виконання повсякденних функцій, зменшить витрати та на початку встановлення та підключення буде суттєво вигідною для більшості населення в порівнянні з існуючими аналогами.

Метою дослідження є спрощення процесів передачі та обробки даних системи «розумний» будинок, що дозволить суттєво зменшити складність та вартість системи та суттєво підвищить швидкість роботи системи.

В рамках розв'язання поставленої мети *необхідно вирішити наступні задачі:*

1. Проаналізувати сучасний стан проблеми «розумний» будинок і знайти максимально зручні інструменти для розв'язання поставленої мети.
2. Дослідити сучасні методи взаємодії програмно-апаратних комплексів, виявити їх переваги та недоліки та підібрати найбільш зручні комбінації для вирішення поставленої мети.
3. Спроекувати апаратне забезпечення для системи "розумний будинок".
4. Реалізувати програмний додаток з використанням досліджених технологій.

5. Провести інтеграцію апаратного та програмного забезпечення та виконати тестування системи.

Об'єктом дослідження є процеси управління та моніторингу житлових будинків.

Предметом дослідження є методи, засоби та інструменти автоматизації процесів управління та моніторингу житлових будинків.

Використані в роботі *методи дослідження*- методи схемотехнічного, комп'ютерного та програмного моделювання та порівняння і експеримент для визначення переваг та недоліків, тестування та для визначення ефективності роботи створеного комплексу[1,3,4,8].

Наукова новизна одержаних результатів. Вперше запропоновано метод захисту вебсерверу системи, який присвоює кожному пристрою користувача унікальний ідентифікатор, що на етапі формування враховує унікальний адрес мережевого адаптера пристрою (MAC адресу), а також логін та пароль користувача. Даний метод, на відміну від існуючих, не використовує сторонніх серверів автентифікації, що робить систему не залежною від інших. Цей метод дає змогу збільшити захищеність системи не зменшуючи її швидкодію. Це досягається за рахунок того, що унікальний ідентифікатор використовується при кожному запиті користувача до серверу і у випадку не співпадіння – алгоритм методу блокує доступ пристрою до вебсерверу, а у разі повторних запитів – сервер блокує IP адресу пристрою, що ускладнює спроби перенавантаження серверу однотипними запитами.

Практичне значення результатів роботи - запропоновано програмно-апаратний комплекс, що достатньо простий в порівнянні з аналогами, що суттєво збільшує кількість кінцевих користувачів, та створений таким чином, що може бути модифікований та удосконалений відповідно до інтеграції нових елементів будинку.

Джерелами дослідження слугували підручники про автоматизовані системи, електронні джерела, документація по мові програмування Python,

прикладі реалізованих підсистем під управлінням Raspberry Pi, підручники по захисту інформації в IoT.

Апробація. Робота та окремі її аспекти, одержані узагальнення та висновки були оприлюднені на XLVII науково-технічній конференції підрозділів Вінницького національного технічного університету [9,10] та на VIII Міжнародній науково-практичній інтернет-конференції «Сучасний рух науки» (жовтень 2019) [11].

Публікації результатів роботи. Основні ідеї роботи подано до публікації в два збірника науково-технічних конференцій [10, 11].

1 ДОСЛІДЖЕННЯ ОСНОВНИХ ПОНЯТЬ ПРЕДМЕТНОЇ ГАЛУЗІ ТА ОГЛЯД АНАЛОГІВ

1.1 Історія розвитку поняття «розумний будинок»

«Розумний будинок» - це будинок, який має високорозвинені автоматизовані системи для контролю і моніторингу будь-якої функції будинку - освітлення, контролю температури, мультимедіа, безпеки, роботи з вікнами і дверима, якості повітря або будь-яких інших завдань, пов'язаних з необхідністю або комфортом. Процеси автоматизації в такій системі здійснюються за допомогою використання обчислювальної техніки. У більшості випадків такі системи «розумного» будинку складаються із таких компонентів як основний вузол управління системою – центральний контролер, канали передачі сигналів, датчики, інтерфейс за допомогою якого здійснюється взаємодія користувача із системою, реле для управління зовнішнім навантаженням, сервоприводи для забезпечення керування рухомими приладами [1,2, 12].

Довгий час автоматизація процесів житлового будинку була темою наукової фантастики, доки в 1898 р. Не з'явилися перші ознаки інтелектуального управління, коли Нікола Тесла представив перший бездротовий пульт дистанційного керування, який посилав радіохвилі для управління мініатюрним катером. Але нічого насправді не було важливим до 20 століття, коли в 1901 році відбувся бум побутової техніки з пилососом з двигуном. У наступні 20 років відбулася революція в побутовій техніці з холодильниками, сушарками для одягу, пральними машинами, прасками і тостерами, але це можна було дозволити собі як розкіш через високу ціну[13].

Інженери експериментували із автоматизацією повсякденних процесів у житловому будинку в 1930-х роках, але справжнє відкриття зробив в 1966 році Джим Сазерленд розробивши першу систему домашньої автоматизації під назвою "Echo IV", яка могла контролювати температуру, керувати списком покупок або включати та вимикати прилади. Ще одна спроба виникла в 1969

році, коли з'явився кухонний комп'ютер Honeywell - це машина, яка могла створювати рецепти, але ніколи не мала комерційного успіху через величезну ціну. Коли в 1971 році з'явився мікропроцесор, ціна електроніки почала швидко падати, і технологія стала доступною для всіх[12].

Термін "Розумний дім" вперше був введений в дію Американською асоціацією будівельників у 1984 р. До кінця століття комп'ютери та робототехніка поєднувались у багатьох продуктах, намагаючись зробити їх більш розумними. Для опису цього поняття зазвичай використовувався новий термін - Domotics - але, незважаючи на зусилля, щоб зробити цю технологію доступною, вона все ще була дорогою і була доступною тільки заможним людям.

У 1998 році в Уотфорді була відкрита демонстрація під назвою "Integer Millennium House", де було представлено, як автоматизація будинку може бути інтегрована у будинку з опалювальними системами, автоматичним контролем ґрунту у саду, охоронними системами, освітленням та контролем за дверима. Поступово, коли технології стали доступнішими, ці технології почали інтегруватися в інші будинки. Зі зростанням популярності зростало більше інвестицій у те, щоб зробити їх дешевшими та ефективнішими і, врешті-решт, доступними для звичайних людей[14].

1.2 Основні функції розумного будинку

До основних функцій розумного будинку відносять: моніторинг, контроль, ефективність та інтелект. Розглянемо кожний з них більш детально.

Моніторинг - користувач повинен мати можливість відстежувати в режимі реального часу або за допомогою історії, з будь-якого віддаленого місця розташування, статус будинку. Розумний будинок повинен надавати оперативні дані і статистику про більшість інтегрованих систем, таких як споживання

енергії, споживання води, моніторинг температури і вологості, стан опалювальної системи, оповіщення про доступ до безпеки, сканування близькості і підрахунок присутності людей. Програмний інструмент, доступний з будь-якого пристрою, повинен обробляти ці дані і надавати цінну статистику[15].

Контроль – встановлюючи систему «розумний» будинок у свої житло, користувач повинен мати можливість повного або часткового контролю процесів у будинку, а також управляти приладами у ньому. Але, в той же час, система не повинна бути спроектована так, щоб керування будинком здійснювалось виключно за допомогою неї. Система має виконувати роль помічника у домі, а контроль повинен здійснюватися людиною. Це дозволить запобігти виникненню неконтрольованих ситуацій у випадок відмов чи збоїв системи[16].

Ефективність – «розумний» будинок повинен бути набагато ефективнішим ніж звичайний будинок. Перш за все, «розумний» будинок повинен забезпечувати низькі показники енергоспоживання. Отже, основне програмне забезпечення повинно мати алгоритм управління енергетичними процесами, щоб безперервно контролювати і аналізувати дані від внутрішніх і зовнішніх давачів і постійно надавати статистику користувачеві для того щоб продумувати як коригувати всі системи, що споживають електроенергію. Крім того, якщо у будинку є паралельна система відновлюваної енергії, така як сонячні панелі, внутрішня система будинку повинна управляти накопиченням енергії і розподілом її залежно від навантаження будинку і погодних умов.

Інтелект –дім повинен «знати» в будь-який час, скільки людей знаходиться всередині, а також скільки людей в кожній кімнаті. Це важливо, це може бути основним фактором в управлінні системами опалення, освітленням, а також в налаштуванні безпеки і сигналізації, і може бути легко досягнуто за допомогою давачів наближення і магнітних лічильників дверей. Завдяки цій перевазі він повинен мати можливість вимикати електроприлади, вимикати світло і економити опалювальну енергію в приміщеннях, коли в них немає

людей, а також знати, коли користувач покидає дім на тривалий час, аналізуючи відсутність руху у будинку, або за допомогою попередньо налаштованого планувальника[3; 16, 17].

1.3 Огляд існуючих систем “Розумний будинок”

1.3.1 Система “Smart Bus”

Дана система використовує роздільну логіку, у даній системі відсутній центральний процесор, якщо користувач має один керований та керуючий пристрій – він з легкістю може називати себе власником «розумного» будинку. Частіше всього, такою парою можуть являтися блок реле у поєднанні із DDP панеллю. В подальшому користувач може розширювати свою систему по мірі своїх потреб[8, 16].

Система “SmartBus” дає можливість керувати освітленням , медіа системами, захистом від надзвичайних ситуацій таких як протікання води та пожежа, керування системами мікроклімату таких як опалення та кондиціонування та іншими різними електроприладами.

Однією з переваг даної системи «розумний» будинок “SmartBus” є її доступність у базовій комплектації, система у такому варіанті комплектації може вважатись найдоступнішою із інших систем на українському ринку. Також до переваги даної системи можна віднести те, що немає необхідності у центральному процесорі, його покупці та установці, що робить систему децентралізованою[18].

Недоліком цієї системи є її ціна, яка становить 950 доларів за систему для однокімнатної квартири та 1650 доларів для двокімнатної квартири, що робить систему недоступною для середньостатистичної людини.

1.3.2 Система розумного будинку “Кластер”

Дана система являється модульною і призначена для виконання багатьох задач для управління процесами у будинку. Система може керувати освітленням, шторами на вікнах, приладами для очищення повітря та кондиціонування, управлінням підлогою з теплим контуром та опаленням у будинку. У даній системі існує можливість курування медіа системами, системою охорони та сигналізацією. Завдяки продуманому дизайну елементів управління системою, вона дає можливість інтегрувати її у будь який інтер'єр будь якого будинку, цим самим підкреслити його. Для зручності користувача, керування системою може здійснюватися за допомогою девайсів, під управлінням операційної системи IOS.

Система дає змогу керувати освітленням двоповерхового приватного будинку, різними типами кондиціонерів, теплою підлогою з електричним контуром, радіаторами опалення, кухонними системами витяжки, а також витяжками у санвузлі, шторами у будь-які кімнаті. Також у системі існує можливість керування усіма пристроями, що керуються за допомогою команд, що передаються інфрачервоним світлом[19].

Для зручного керування системою, для користувача можлива установка панелей з клавішами та дисплеєм для відображення інформації. Освітлення регулюється клавішами на панелі, а також за допомогою датчиків присутності і датчиків руху. Дані датчики вмикають світлові прилади, коли користувач входить у будь яку кімнату. Для більшої зручності у системі існує можливість керування за допомогою девайсів iPhone та iPad. Дана модульна система «розумного» будинку призначена для управління дев'ятнадцятьма вимикачами, дев'ятьма приладами регулювання яскравості освітлення, п'ятьма кондиціонерами, дев'ятьма радіаторами опалення, шторами на шести окремих вікнах.

Переваги: у системі присутня можливість використання вільних груп для керування системою безпеки, медіа системами, а також іншим електрообладнанням [20].

Недоліки: у базовій комплектації відсутній модуль сповіщення, але є можливість його додаткового встановлення.

Ціна такої системи становить приблизно 1 мільйон 400 тисяч гривень.

1.3.3 Система для управління будинком "CLAP"

CLAP - це, за словами розробників, "перша повноцінна система умного будинку, повністю створена в Україні". Як будь-яка інша система, CLAP представляє собою комплекс програмно-технічного забезпечення. Апаратна частина складається з координаційного хабу, панелі управління, об'єднаного з відеодомофоном, терморегуляторами для контролю опалювальних приладів і набором датчиків, які вимірюють температуру в квартирі, вологість, сигналізують про рух і аналізують інші параметри[21-22].

До досконалості системи розумного будинку CLAP виробник віднесуть простоту підключення, гнучкість настройки, функції охоронної та пожежної сигналізації, а також можливість економії на оплаті комунальних послуг. Тепер конкретніше розпишіть по кожному з пунктів.

До складу системи входять датчики для захисту від пожеж, диму, підриву і навіть від високої концентрації вуглекислого газу.

CLAP інтегрована з охоронними компаніями - партнерами «УКРБУД», і, у випадку спрацьовування сигналізації, CLAP передає сигнал не тільки власнику, але і на пульт охорони, який зможе оперативно відреагувати на виклик[21].

Система автоматично зчитує показники всіх облікових записів і фіксує їх, може автоматично платити за рахунками (потрібно лише прив'язати банківську карту), прогнозувати витрати на комунальні послуги та планувати роботу розумного будинку з урахуванням вартості енергоресурсів залежно від часу доби.

За словами розробників, за місяць економія в квартирі, при використанні системи CLAP, може скласти до 40% від загальної суми по комунальних платежах[22].

З решти можна відзначити можливість трансляції відео з домофону та відеоспостереження на смартфон, рівні доступу та функції виклику потрібного майстра (сантехнік, електрик тощо) прямо з програми. Остання функціональність першого часу буде доступна лише для мешканців будинків від «УКРБУД».

Вартість такої системи змінюється в залежності від набору елементів, однак вартість базового набору стартує від 500 тис.грн.

2 ПРОЕКТУВАННЯ СИСТЕМИ «РОЗУМНИЙ БУДИНОК»

2.1 Характеристики одноплатного комп'ютеру RaspberryPi 3 B+

Raspberry Pi 3 Model B + - одноплатний комп'ютер від Raspberry Pi Foundation, який працює на базі оновленого 4х-ядерного 64-бітного SoC Broadcom BCM2837B0 і збільшеною тактовою частотою 1.4GHz. Бездротовий модуль Wi-Fi підтримує дводіапазонний стандарт IEEE 802.11ac, Bluetooth - 4.2 BLE. Комп'ютер володіє Gigabit Ethernet, що працює через USB 2.0 шину, що забезпечить швидкість передачі даних до 300Mbps. Обсяг оперативної пам'яті - 1GB, формфактор і розташування всіх портів, що забезпечує сумісність більшості існуючих аксесуарів[23].

Raspberry Pi 3 Model B + як і її попередники відрізняється високим рівнем надійності, простотою настройки, величезним співтовариством і високою якістю виконання.

Опишемо деякі особливості Raspberry Pi 3 B + у таблиці 2.1.

Таблиця 2.1 – Основні технічні характеристики RaspberryPi 3 B+

Компонент	Опис
SoC	Broadcom BCM2837B0 чотирьохядерний A53 (ARMv8) 64-розрядний з тактовою частотою 1,4 ГГц
GPU	Broadcom Videocore-IV
RAM	1 ГБ LPDDR2 SDRAM
Network	Gigabit Ethernet (через USB-канал), 2,4 ГГц та 5 ГГц 802.11b / g / n / ac Wi-Fi
Bluetooth	Bluetooth 4.2, Bluetooth Low Energy
Storage	MicroSD
GPIO	40-pin GPIO header, populated

Нова система Raspberry Pi 3B + BCM2837 на мікросхемі (SoC) має вдосконалене компонування та систему відведення тепла, що допомогло підвищити її продуктивність від 1,2 ГГц до 1,4 ГГц. Новий контролер USB Ethernet пропонує гігабітний зв'язок при теоретичній максимальній пропускній здатності 300 Мбіт / с, завдяки використанню одного каналу USB [24].

2.2 Переваги платформи Raspberry Pi для розумового будинку

Raspberry Pi – платформа для розробки із серії Pi. Її можна розглядати як одноплатний комп'ютер, який працює на операційних системах типу Linux. Плата має не лише багато функцій, вона також має надзвичайну швидкість обробки даних, що робить її придатною для вимог сучасного програмного забезпечення. Платформа RaspberryPi спеціально розроблена для любителів та інженерів, які цікавляться системами Linux та IoT (InternetofThings) [25].

Переваги Raspberry Pi 3 B+:

- а) підтримка Linux операційних систем;
- б) ціна; комп'ютер має низьку вартість в порівнянні із іншими;
- в) продуктивність;
- г) наявність великої кількості програмного забезпечення для моделювання проекту;
- д) готовність до використання;
- е) велика кількість готових рішень з використанням даної платформи, що також вказує на якість та ефективність її використання.

Вебсервер – система, яка встановлюється на ЕОМ, призначена для забезпечення функціонування систем типу «клієнт-сервер». Вебсервер приймає запит (request) від клієнта, обробляє його і повертає відповідь (response) клієнту, який, у більшості випадків представлений у вигляді HTML сторінки [25].

2.3 Компоненти розумного будинку

Система «розумний» будинок включає в себе наступні вузли автоматизації [19, 24]:

Розглянемо основні елементи кухні.

Світло. Реалізовано два режими включення світла – ручний – вмикається звичайним вимикачем та автоматичний – вмикається по давачу руху, давач відслідковує наявність людини в приміщенні та через 3 хвилини після покидання людиною приміщення, світло вимикається.

Витяжка. Вмикається вручну, або автоматично по інформації із давача диму, давач фіксує концентрацію диму в повітрі та посилає сигнал на сервер, який вмикає реле [26].

Жалюзі. Реалізовано два режими – ручний, у якому управління жалюзі здійснюється за допомогою сервопривода, який контролюється потенціометром, при зміні положення ручки потенціометра положення жалюзі змінюється та автоматичний, у якому регуляція жалюзі здійснюється автоматично за показниками давача освітленості.

Моніторинг температури та вологості у приміщенні. Показники зчитуються із давача температури та вологості, після цього інформація виводиться у вебінтерфейс.

Моніторинг диму, пожежі, протікання води. Давач диму та давач вологості фіксують та передають дані на сервер, сервер обробляє дані та вмикає звукове оповіщення та відсилає господареві СМС-сповіщення.

Опишемо елементи санвузла.

Світло. Реалізовано два режими включення світла – ручний – вмикається звичайним вимикачем та автоматичний – вмикається по давачу руху, давач відслідковує наявність людини в приміщенні та через одну хвилину після покидання людиною приміщення, світло вимикається.

Моніторинг температури та вологості у приміщенні. Показники зчитуються із датчиків, після цього інформація виводиться у вебінтерфейс[6].

Моніторинг протікання води. Датчик вологості фіксує та передають дані на сервер, сервер обробляє дані та вмикає звукове оповіщення та відсилає господареві СМС-сповіщення.

Автоматичне включення витяжки. Датчик руху фіксує рух та вмикає витяжку.

Розглянемо основні елементи вітальні.

Світло. Реалізовано два режими включення світла – ручний – вмикається звичайним вимикачем, автоматичний – вмикається по датчику руху, датчик відслідковує наявність людини в приміщенні та через одну хвилину після покидання людиною приміщення, світло вимикається та нічний режим – вмикаються дві малопотужні лампи.

Жалюзі. Реалізовано два режими – ручний, у якому управління жалюзі здійснюється за допомогою сервопривода, який контролюється потенціометром, при зміні положення ручки потенціометра положення жалюзі змінюється та автоматичний, у якому регуляція жалюзі здійснюється автоматично за показниками датчика освітленості.

Моніторинг температури та вологості у приміщенні. Показники зчитуються із датчиків, після цього інформація виводиться у вебінтерфейс.

Розглянемо основні спальні 1.

Світло. Реалізовано два режими включення світла – ручний – вмикається звичайним вимикачем, автоматичний – вмикається по датчику руху, датчик відслідковує наявність людини в приміщенні та через одну хвилину після покидання людиною приміщення, світло вимикається та нічний режим – вмикаються дві малопотужні лампи.

Жалюзі. Реалізовано два режими – ручний, у якому управління жалюзі здійснюється за допомогою сервопривода, який контролюється потенціометром, при зміні положення ручки потенціометра положення жалюзі

змінюється та автоматичний, у якому регуляція жалюзі здійснюється автоматично за показниками датчика освітленості.

Моніторинг температури та вологості у приміщенні. Показники зчитуються із датчиків, після цього інформація виводиться у вебінтерфейс.

Розглянемо основні елементи спальні 2.

Світло. Реалізовано два режими включення світла – ручний – вмикається звичайним вимикачем, автоматичний – вмикається по датчику руху, датчик відслідковує наявність людини в приміщенні та через одну хвилину після покидання людиною приміщення, світло вмикається та нічний режим – вмикаються дві малопотужні лампи.

Жалюзі. Реалізовано два режими – ручний, у якому управління жалюзі здійснюється за допомогою сервопривода, який контролюється потенціометром, при зміні положення ручки потенціометра положення жалюзі змінюється та автоматичний, у якому регуляція жалюзі здійснюється автоматично за показниками датчика освітленості.

Моніторинг температури та вологості у приміщенні. Показники зчитуються із датчиків, після цього інформація виводиться у вебінтерфейс.

Вхідні ворота. Реалізовано відкривання воріт по RFID модулю. Відкривання воріт здійснюється за допомогою піднесення до визначеного місця заздалегідь внесеної в пам'ять RFID-мітки [15, 17].

2.4 Система управління розумним будинком

На рисунку 2.1 наведено запропоновану спрощену структуру системи. Інформація поступає на датчики (B_1 - B_n), далі по каналах зв'язку поступає на головний контролер системи (CPU). Керування системою здійснюється користувачем через вебінтерфейс (U_1/U_2) шляхом використання вебсерверу, яким керує одноплатний комп'ютер Raspberry Pi 3 B+, що дає можливість здійснювати управління системою через локальну мережу. Для

цього потрібно підключити комп'ютер до домашнього маршрутизатору кабелем, або за допомогою WiFi, після чого ввести адресу серверу у локальній мережі, після чого користувачеві буде доступний весь інтерфейс для управління системою.

Дана система «розумний» будинок дає змогу інтелектуального управління, а також централізованого контролю процесами та приладами у приватних будинках [27].

Використання подібної системи у своєму будинку, користувачеві відкриваються можливості:

- у своєму середовищі проживання встановлювати індивідуальні параметри навколишнього середовища у будинку такі як освітлення, температуру повітря, гучність медіа систем;
- здійснювати управління необхідною системою (освітлення, клімат, тощо);
- мати доступ до інформації про стан систем життєзабезпечення свого будинку в реальному часі, при цьому знаходячись безпосередньо у будинку, або за його межами [28].

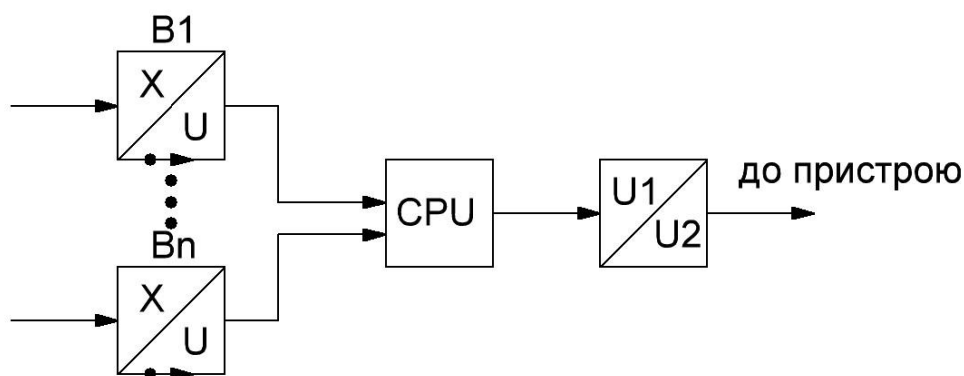


Рисунок 2.1 – Спрощена структурна схема системи

Основні складові системи управління можна розділити на наступні:

- а) центральний процесор управління / одноплатний комп'ютер;
- б) давачі (температури, освітленості, задимленості, руху та ін.);

- в) керуючі пристрої (диммери, реле, ПЧ-емітери та ін.);
- г) інтерфейси управління (кнопкові вимикачі, web інтерфейс);
- д) власна мережа управління, що об'єднує вищевказані елементи;
- е) керовані пристрої (світильники, витяжки, серво та ін.);
- є) допоміжні мережі (Ethernet, GSM мережа).

У даній системі основною задачею центрального процесору є управління усіма підключеними до нього компонентами, серед них реле та датчики, які можуть бути підключені як кабелями так і бездротовою технологією WiFi. Також використання одноплатного комп'ютеру RaspberryPi 3 дає можливість делегувати на нього велику кількість задач, таких як забезпечення функціонування вебінтерфейсу користувача, використовуючи його обчислювальні можливості, розробити програмне забезпечення для обробки запитів користувача та модулю захисту системи[3].

Алгоритм взаємодії датчиків, реле із центральним процесором можна описати наступним чином:

- по власній мережі управління інформація від датчиків або інтерфейсів надходить до центрального процесора управління;
- підготовлене програмне забезпечення центрального процесора обробляє дану інформацію та генерує команди для реле, що в подальшому керує навантаженням.

Способи генерації команд, а також форма і склад відображуваної інформації про стан систем закладається на етапі розробки програмного забезпечення з урахуванням вимог проекту [29].

Отже, «розумний» будинок складається з таких частин:

- пристрої – всі користувацькі електронні побудові речі, управління якими необхідно автоматизувати;
- датчики – пристрої для збору інформації із навколишнього середовища у будинку;

- мікроконтролери – апаратні системи, що об'єднують датчики в групи, розрізняють також центральний процесор управління – мікроконтролер, що посилає від сервера інформацію в кінцеві вузли.

- канали передачі даних – логічні та фізичні канали, по яким передаються дані з урахуванням потреб (безпека, швидкість тощо);

- обладнання користувача – пристрої, за допомогою яких користувач через сервер керує системою розумного дому [27, 30].

3 РОЗРОБКА КОМПОНЕНТІВ СИСТЕМИ

3.1 Компонент електронного замка з використанням RFID модулю “RC522”

Компонент системи дозволяє керувати входними воротами віддалено та за допомогою карток RFID [31].

Модулі, що входять до складу, схема їх підключення зображена на рисунку 3.1:

- а) контролер ESP8266;
- б) RFID зчитувач “RC522”;
- в) OLED дисплей 128X64 IEC SSD1306;
- г) твердотільне реле;
- д) резистор 220 Ом;
- е) RGB світлодіод;
- ж) карти з RFID.

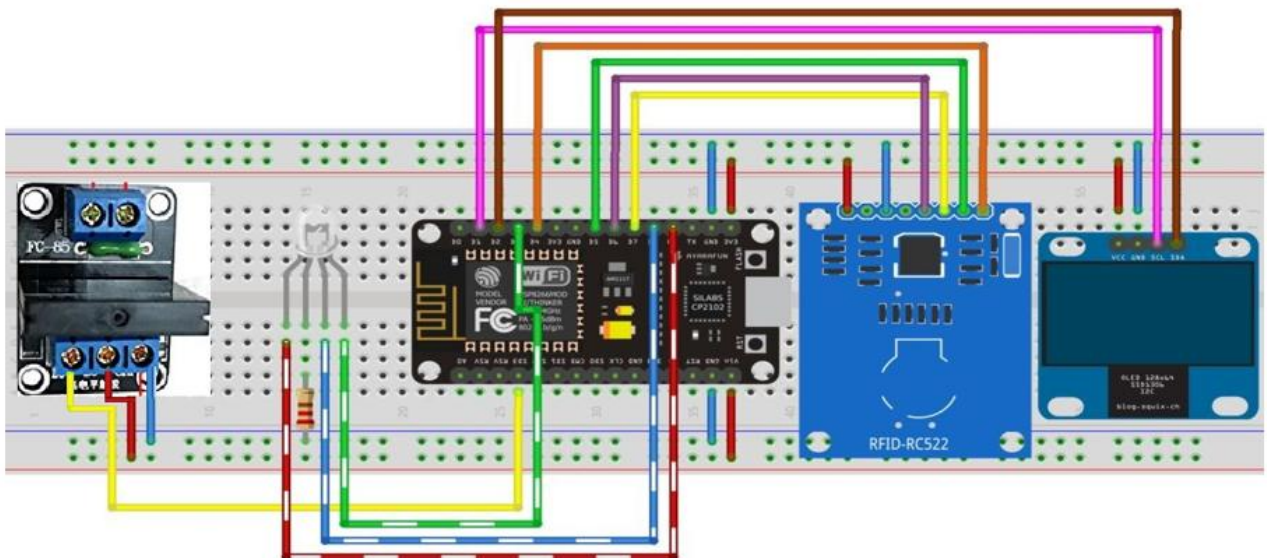


Рисунок 3.1 - Схема підключення

RFID зчитувач “RC522” та дисплей встановлюються зовні біля воріт. За замовчуванням у режимі «заблоковано», реле знаходиться у вимкненому стані, на його контакти не подається сигнал від контролера, це стан «Зачинено». Як тільки користувач підносить карту до зчитувача, система зчитує код карти та звіряє із базою кодів у пам’яті, якщо код карти знаходиться у базі – контролер подає сигнал на реле, яке подає напругу на контакти блокуючого механізму, що відкривається. Це стан - «Відчинено» і ворота відчиняються разом із цим на дисплей виводиться повідомлення “Open”, показано на рисунку 3.2 (для прототипу було використано LCD дисплей) [32].



Рисунок 3.2 - Відкриття за допомогою карти, яка знаходиться в базі

Якщо код карти не знаходиться у базі, контролер не подає сигнал на серво-привод, а тільки виводить на дисплей повідомлення “Access denied!!!” і ворота залишаються закритими – показано на рисунку 3.3.

RGB світлодіод встановлений для індикації поточного стану реле. Для коректної роботи компоненту, його потрібно приєднати до автономного

живлення усієї системи[30]. При наявності додаткового автономного живлення навіть при відключенні централізованого енергопостачання, користувач зможе контролювати вхідні ворота.

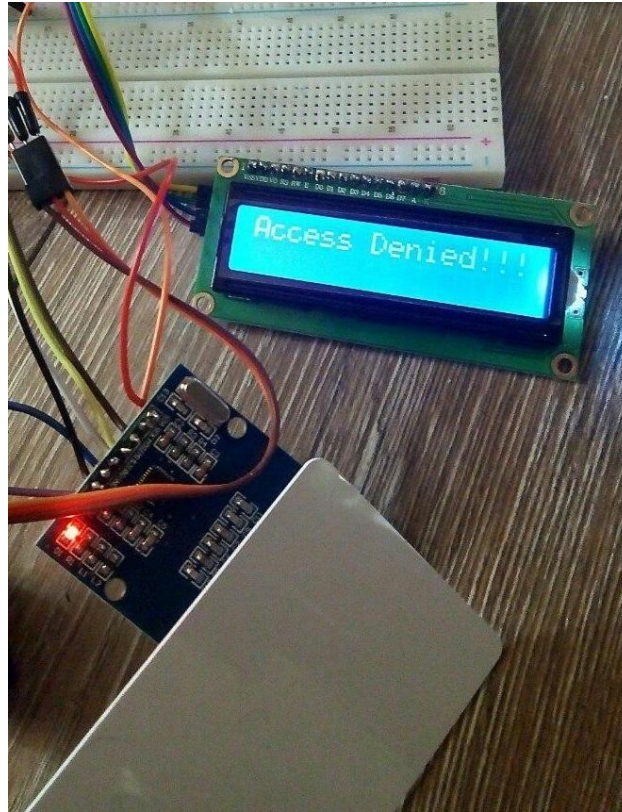


Рисунок 3.3 - Карта, що не знаходиться в базі – доступ заборонено

3.2 Підготовка WiFi реле до роботи в системі

У якості реле для системи, було обрано WiFi реле Sonoff. Нижче наведено його характеристики, а також переваги.

Характеристики реле Sonoff Basic:

- а) діапазон напруги: 90-250В змінного струму (50 / 60Гц);
- б) максимальний ток: 10А;
- в) максимальна потужність: 2200 Вт;
- г) розміри: 88 * 38 * 23мм (Д * Ш * В);
- д) власність: 5% -95%;

е) частота: 2,4 ГГц;

є) робоча температура: 0 ° C-40 ° C (32 ° F-104 ° F).

Переваги реле Sonoff Basic

а) підтримка Wi-Fi-мереж;

б) підтримка відстеження статусу підключених пристроїв, які були встановлені в eWeLink в реальному часі;

в) підтримка віддаленого включення або виключення для підключених пристроїв;

г) підтримка до восьми тимчасових завдань, призначених для кожного додаткового пристрою;

д) можливість з'єднання декількох реле у одну систему.

Sonoff - це бездротовий комутатор WiFi, який може підключатися до широкого спектру побутової техніки. Sonoff передає дані на хмарну платформу через домашній WiFi-роутер, який дозволяє користувачам дистанційно керувати підключеними пристроями через мобільний додаток під назвою eWeLink [33].

Sonoff також можна використовувати з прошивкою з відкритим кодом під назвою ESP Easy. ESP Easy дозволяє користувачам повністю налаштувати Sonoff (IP-адреса, DNS, шлюзи, датчики, правила тощо) за допомогою веб-інтерфейсу. Sonoff з ESP Easy стає дуже популярним пристроєм для домашньої автоматизації самостійно.

Для даної системи – реле Sonoff буде використовуватись із прошивкою із відкритим кодом, так як в такому режимі – реле не буде відправляти дані на хмарні середовища, а буде мати можливість працювати у локальній мережі.

Для підготовки реле потрібно завантажити файл прошивки із відкритого ресурсу, змінити декілька стрічок коду в прошивці, таких як модель реле, SSID та пароль WiFi мережі, в які буде працювати реле. Прошивка реле буде здійснюватись за допомогою UART контролера [34].

Реле Sonoff має спеціальні виводи для прошивки за допомогою UART контролеру, які показано на рисунку 3.4.

Спосіб підключення реле до UART контролеру наведено у таблиці 3.1.

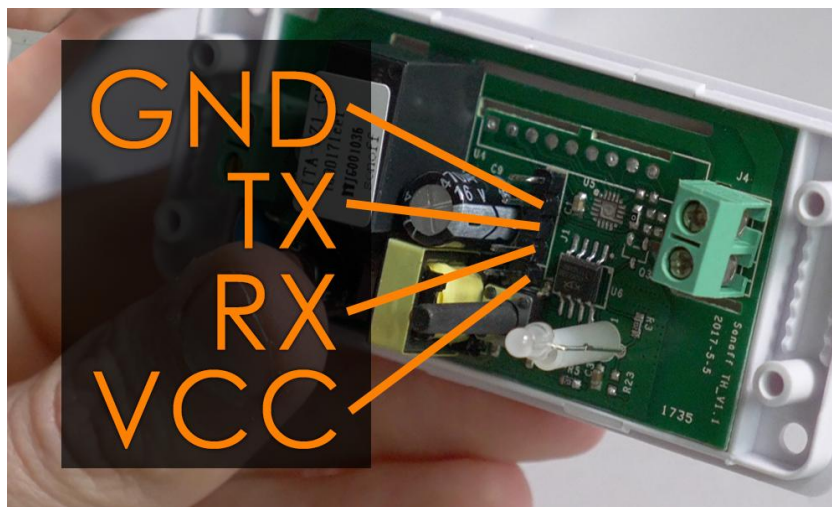


Рисунок 3.4 – Виводи реле для прошивки

Таблиця 3.1 – Спосіб підключення реле до UART контролеру для прошивки

UART	Sonoff
Vcc	Vcc
GNG	GND
RX	TX
TX	RX

Для прошивки потрібно використати середовище розробки Arduino IDE. Після завантаження файлів прошивки, потрібно завантажити їх у Arduino IDE, підключити реле через UART контролер до комп'ютера, в IDE обрати COM порт, до якого підключено реле і натиснути кнопку прошивки. Після цього реле буде готове до використання у даній системі. Перевірити роботу реле можна шляхом введення IP адреси реле у браузері, як результат буде відображено сторінку управління реле, приклад сторінки наведено на рисунку 3.5.

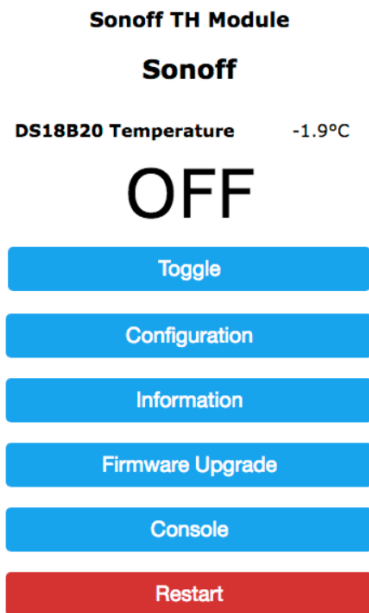


Рисунок 3.5 – Сторінка управління реле

3.3 Розробка інтерфейсу для адміністратора системи

Для попереднього налаштування системи, додавання нових реле в систему, реєстрації нових користувачів в системі, потрібно розробити інтерфейс адміністратора системи, до якого доступ буде мати конкретна людина – адміністратор будинку.

Для розробки інтерфейсу адміністратора було обрано фреймворк мови програмування Python – Flask, SCC фреймворк – Bootstrap, для зберігання даних біло використано базу даних MySQL [1].

Реєстрація адміністрація у системі відбувається після першої ініціалізації системи, при відкриванні основної сторінки системи у браузері.

Після авторизації до інтерфейсу, сторінка авторизації показана на рисунку 3.6, адміністратор може додавати нові реле до системи, реєструвати нових, або блокувати існуючих користувачів.

Додавання нового користувача відбувається після преходу на сторінку «Add new user», де у відповідні поля потрібно ввести необхідні параметри, такі як ID користувача, ім'я, MAC адреса користувацького девайса, IP адреса в

локальній мережі, приклад сторінки додавання нового користувача, показано на рисунку 3.7.

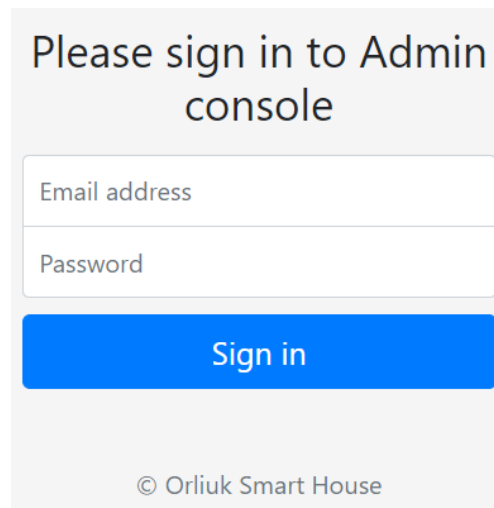


Рисунок 3.6 – Сторінка авторизації до інтерфейсу адміністратора

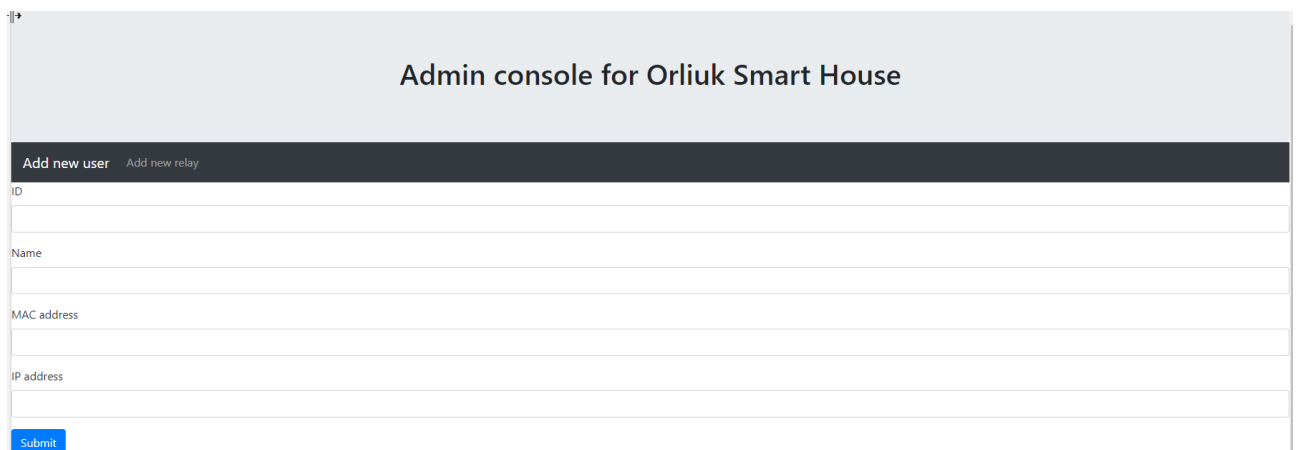


Рисунок 3.7 – Сторінка додавання нового користувача у систему

Розглянемо основні поля форми.

ІД користувача – обов’язкове поле, встановлюється БД автоматично.

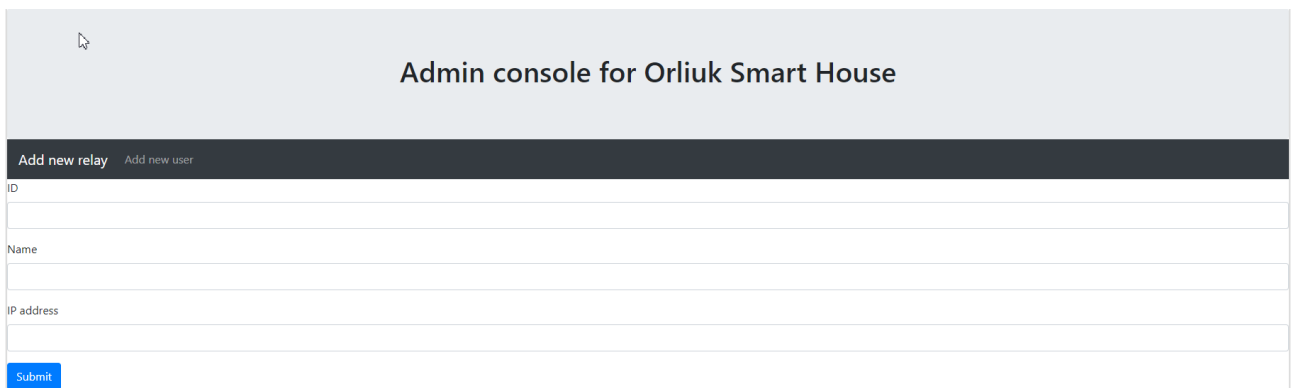
Ім’я – обов’язковий параметр.

Логін – обов’язковий параметр, повинен містити в собі тільки букви латинського алфавіту, цифри, а також символ нижнього підкреслювання і складатись не більше ніж з 10 символів.

Пароль – обов’язковий параметр, пароль користувачеві встановлює адміністратор системи. Пароль зберігається у базі зашифрований алгоритмом “md5”[4].

MAC адреса користувацького девайсу – обов’язковий параметр, цей параметр є унікальним для кожного мережевого адаптеру всіх девайсів. Дізнатись MAC адресу можна різними способами, в залежності від операційної системи девайсу.

Додавання нових реле відбувається на сторінці «Add new relay», де у відповідні поля потрібно ввести необхідні параметри, такі як ID реле та його IP адреса, приклад сторінки додавання нового реле показано на рисунку 3.8.



The screenshot shows a web interface titled "Admin console for Orliuk Smart House". Below the title bar, there are two navigation links: "Add new relay" and "Add new user". The main content area contains a form with three input fields: "ID", "Name", and "IP address". At the bottom left of the form is a blue "Submit" button.

Рисунок 3.8 – Приклад сторінки додавання нового реле до системи

ID реле – обов’язковий параметр, повинен містити в собі тільки букви латинського алфавіту, цифри, а також символ нижнього підкреслювання і складатись не більше ніж з 10 символів. По даному параметру налаштовано прив’язку конкретного реле у вебінтерфейсі користувача та на стороні модулю обробки запитів.

Ім’я – обов’язковий параметр.

IP адреса – обов’язковий параметр, IP адресу реле можна дізнатись одразу після налаштування реле для роботи у системі на сторінці управління реле.

Дані додаються у MySQL базу даних, що має назву “shouse” у відповідні таблиці.

Таблиця "users" зберігає в собі інформацію про користувачів у системі, приклад даних представлено на рисунку 3.9.

```
mysql> select * from users;
+-----+-----+-----+-----+-----+
| id | name | login      | password                                     | mac_id |
+-----+-----+-----+-----+-----+
| 1 | Batya | first_user | 1b3231655cebb7a1f783eddf27d254ca | ff:ff:ff:ff:ff:ff |
+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)
```

Рисунок 3.9 – Приклад даних у таблиці "users"

Таблиця "admins" зберігає інформацію про адміністраторів системи у даній таблиці, при додаванні запису – поле "password" додається зашифрованим алгоритмом "md5", приклад даних представлено на рисунку 3.10.

```
mysql> use shouse; select * from admins;
Database changed
+-----+-----+
| login      | password                                     |
+-----+-----+
| SuperAdmin | 2034f6e32958647fdff75d265b455ebf |
+-----+-----+
1 row in set (0.00 sec)
```

Рисунок 3.10 – Приклад даних у таблиці "admins"

Таблиця "relays" зберігає інформацію про реле, що додані у систему, приклад даних представлено на рисунку 3.11.

3.4 Розробка модулю захисту системи

Для того щоб зробити основний сервер системи захищеним від небажаних запитів, а також заборонити можливість незареєстрованим користувачам керувати системою, було розроблено метод захисту вебсерверу

системи, який присвоює кожному пристрою користувача унікальний ідентифікатор, що на етапі формування враховує унікальний адрес мережевого адаптера пристрою (MAC адресу), а також логін та пароль користувача[35].

```
mysql> use shouse; select * from relays;
Database changed
+-----+-----+-----+
| id | name  | ip_id |
+-----+-----+-----+
| 1  | REL_1 | 192.168.0.234 |
| 4  | REL_2 | 192.168.0.12  |
| 6  | REL_4 | 192.168.0.14  |
+-----+-----+-----+
3 rows in set (0.00 sec)
```

Рисунок 3.11 – Приклад даних у таблиці “relays”

Даний метод, на відміну від існуючих, не використовує сторонніх серверів автентифікації, що робить систему не залежною від інших. Цей метод дає змогу збільшити захищеність системи не зменшуючи її швидкодію.

Це досягається за рахунок того, що унікальний ідентифікатор використовується при кожному запиті користувача до серверу і у випадку не співпадіння – алгоритм методу блокує доступ пристрою до вебсерверу, а у разі повторних запитів – сервер блокує IP адресу пристрою, що ускладнює спроби перенавантаження серверу однотипними запитами [36].

Для написання модулю було обрано мову програмування Python. Модуль працює на окремому TCP порту - 7557, що показано на рисунку 3.12. Програму було інтегровано в операційну систему, як systemd сервіс.

```
tcp 0 0 0.0.0.0:5775 0.0.0.0:* LISTEN 24333/python
```

Рисунок 3.12 – Порт модулю безпеки

Принцип роботи модулю безпеки. При реєстрації нового користувача, для нього створюється ключ, що складається із його логіна, паролю, та MAC адреси

його девайсу. Це значення записується у окрему базу даних, що має назву “index_keys”, приклад даних показано на рисунку 3.13.

```
mysql> select * from index_keys;
+-----+-----+-----+-----+-----+-----+
| id | val_ids |
+-----+-----+-----+-----+
| 1 | knwxy_zxjwg8786baahjggcf6kcd8jiik7ci7a9hfkk:kk:kk:kk:kk:kk |
+-----+-----+-----+-----+
1 row in set (0.00 sec)
```

Рисунок 3.13 – Приклад даних у таблиці “index_keys”

При кожному натисканню користувача на будь-яку піктограму у вебінтерфейсі, до HTTPзапиту додаються такі його параметри, як логін, пароль (в md5форматі) та MACадреса, приклад запиту показано на рисунку 3.14.

Рисунок 3.14 – Приклад HTTPGETзапиту від користувача

Опис параметрів запиту описано в таблиці 3.2.

Таблиця 3.2 – Опис параметрів клієнтського запиту.

Параметр	Опис
my-house.my	FQDN серверу системи
/control	Розділ керування системою
rel_id	IDреле, стан якого потрібно змінити
cmd	Команда, що віддається на оброку реле
luid	Логін користувача
puid	Пароль, у md5 форматі
huid	MAC адреса девайсу

Модуль приймає HTTP GET запити від вебінтерфейсу, що надходять на його порт, після цього зашифровує параметри, приклад функції для шифрування наведено на рисунку 3.15, що надійшли у запиті, отримане значення порівнює із значенням у базі даних. У випадку, якщо зашифроване значення збігається із значенням у базі – запит передається основному модулю обробки запитів на порт 55577. Якщо ж значення не збігаються – модуль ігнорує даний запит, а користувач не отримує ніякої відповіді від серверу. Дані про некоректні запити записуються у журнал програми на сервері [37].

```
def caesar_code(text, shift):  
    shift_text = ''  
  
    for c in text:  
        if c not in alphabet_list:  
            shift_text += c  
            continue  
  
        i = (alphabet_list.index(c) + shift) % len(alphabet_list)  
        shift_text += alphabet_list[i]  
  
    return shift_text
```

Рисунок 3.15 – Лістинг функції для шифрування

3.5 Розробка системи управління будинком

Розглянемо перелік основних вузлів системи управління та опис принципу роботи.

Для забезпечення контролю «Розумним» будинком, потрібно розробити наступні вузли системи управління:

- а) вебінтерфейс (frontend);
- б) вебсервер (server);
- в) основний модуль обробки запитів (backend)[27].

Вебінтерфейс являє собою вебсторінку, яка створюється за допомогою мови розмітки тексту HTML. Для забезпечення зручного вигляду, використовується каскадні таблиці стилів CSS. Для реалізації даного інтерфейсу було обрано HTML 5 та CSS 3, так як дані версії найкраще підтримуються усіма браузерами.

Користувач безпосередньо взаємодіє із вебінтерфейсом, приклад якого показано на рисунку 3.12, для керування системою. Кожна піктограма відповідає за конкретне WiFi реле, що підключене до системи. Користувач натискаючи на піктограму, активує це реле, що керує певним вузлом у будинку [38].

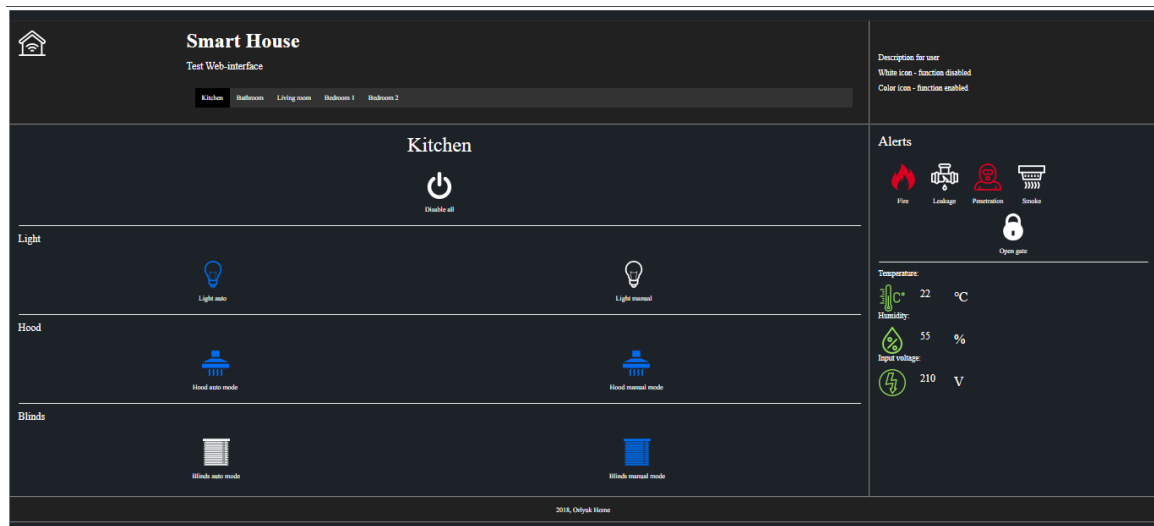


Рисунок 3.12 – Вебінтерфейс системи

Вебсервер являє собою програму для забезпечення роботи клієнт-серверних додатків, шляхом прийому, обробки та віддачі клієнтських запитів. У даній системі, вебсервер служить для віддачі вебінтерфейсу користувачеві, прийому користувацьких запитів, та прийому статусів реле. Для реалізації вебсерверу в даній системі, було обрано поширений вебсервер Apache2. Даний сервер було обрано тому що він має гарну сумісність із Raspbian, він є простим у конфігурації, також даний сервер ідеально обробляє статичний контент, а так як у даній системі буде тільки статичний контент – можна вважати даний вибір найбільш вдалим для потреб даної системи[39].

Основний модуль обробки запитів. Для обробки запитів користувача і на їх основі керувати реле, було розроблено окремий модуль, що являє собою програму, написаною на мові програмування Python 3. Модуль працює на окремому TCP порту - 55577, що показано на рисунку 3.13 і може приймати і опрацьовувати HTTP запити, які надходять на цей порт.

tcp	0	0 0.0.0.0:55577	0.0.0.0:*	LISTEN	23341/python
-----	---	-----------------	-----------	--------	--------------

Рисунок 3.13 – TCP порт, на якому працює backend

Програма була інтегрована у операційну систему Raspbian як systemd сервіс. На рисунку 3.14 наведено код перевірки реле.

Мова програмування Python є одною з найкращих для написання серверної частини програм для Unix систем.

```

url_switch_on = 'http://%s/control?cmd=event,T1' % relay_ip
url_switch_off = 'http://%s/control?cmd=event,T0' % relay_ip
sonoff_url = 'NOT_INIT'
loop_time = 5

def main():
    loop_counter = 0
    while (True):
        try:
            if loop_counter%2 == 0:
                sonoff_url = url_switch_on
            else:
                sonoff_url = url_switch_off

            r = requests.post(sonoff_url)

            if r.status_code == 200:
                print("Sonoff return code: SUCCESS\n")
            else:
                print("Sonoff return code: FAILED\n")

        except Exception:
            print("UNABLE TO SEND COMMAND TO SONOFF\n")

        loop_counter = loop_counter + 1
        time.sleep(loop_time)

if __name__ == "__main__":
    main()

```

Рисунок 3.14 – Код програми для перевірки реле

Керування системою відбувається наступним чином. Кожне реле має набір параметрів – це унікальне ім'я та його IP адреса в мережі. Користувач тисне на піктограму, якій відповідає певне реле, таким чином посиляється HTTP GET запит до основного модуля обробки запитів, де ідентифікується на

зміну стану якого саме реле надійшов запит, перевіряється поточний його статус, та якщо статуси не співпадають – модуль надсилає сигнал реле на зміну його стану. Також, користувачеві у вебінтерфейс відправляється результат виконання його запиту [40].

Для демонстрації послідовності запитів у системі та між її основними компонентами, було розроблено UMLдіаграму, яка зображена на рисунку 3.15.

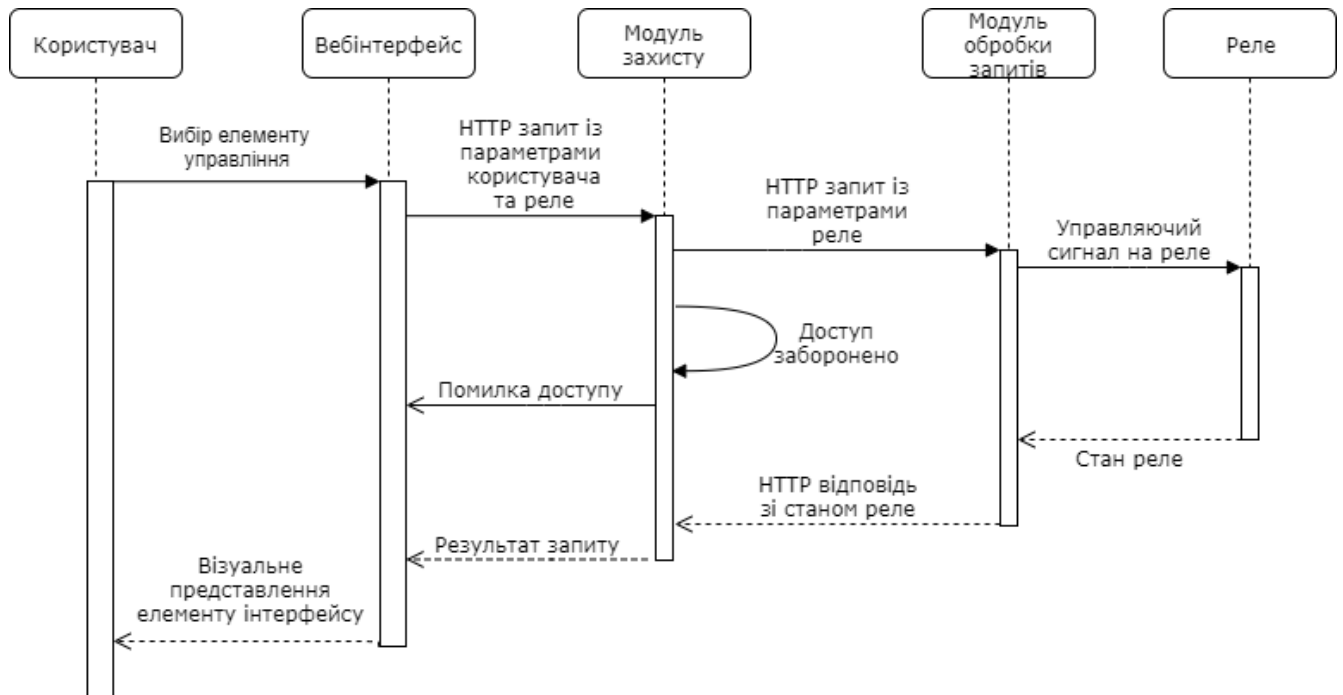


Рисунок 3.15 – UMLдіаграма послідовностей запитів у системі

3.6 Сигналізація для будинку із СМС оповіщенням

Сигналізація дозволяє контролювати критичні ситуації у будинку та оповіщати користувача про них шляхом відправки СМС. Модулі системи встановлюються у будинку в прихованому місці, давачі розводяться по будинку у відповідні місця [41].

Для можливості відправки СМС у GSM-модуль встановлюється SIMкарта із оплаченим тарифним планом, який включає в себе безлімітні

СМС на один номер. Також для коректної відправки СМС антена із GSM-модуля виводиться за межі будинку [42].

Модулі, що входять до складу:

- а) GSM-модуль “NEOWAY M590”;
- б) давач протікання води;
- в) давач руху;
- г) давач Холла;
- д) реле вентилятора;
- е) зумер;
- ж) давач диму;
- з) давач газу.

Схема підключення модулів зображена на рисунку 3.16.

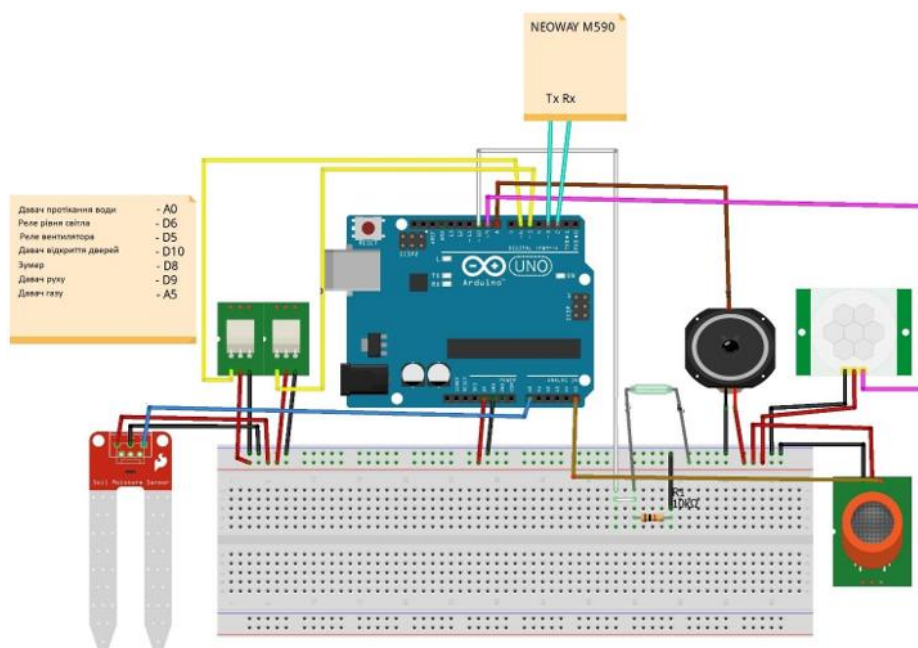


Рисунок 3.16 - Схема підключення модулів

Принцип роботи схеми підключення полягає в наступному. Система встановлюється у режим «Захищено» шляхом відправки СМС із номера користувача на номер системи із текстом «Enable» або натисненням кнопки.

Після цього встановлюється коротка затримка щоб дати можливість користувачеві покинути будинок [43].

На дверях встановлено давач Холла для того щоб контролювати положення дверей. Коли давач відхиляється, система відправляє користувачеві СМС із текстом “Unauthorized penetration!!!” та подає гучний сигнал на зумер. Встановлення системи в режим «Не захищено» встановлюється шляхом відправки СМС із номера користувача на номер системи із текстом «Disable» або натисненням кнопки. Після цього користувач отримує повідомлення із підтвердженням відключення.

Моніторинг та контроль критичних ситуацій відбувається за таким же алгоритмом, тільки замість давача Холла встановлено інші давачі такі як: давач руху, давач диму, давач газу, давач протікання.

4 ІНТЕГРАЦІЯ АПАРАТНОГО ТА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ТА ТЕСТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ СИСТЕМИ

4.1 Інтеграція апаратного та програмного забезпечення системи

Після розробки програмного забезпечення для усієї системи, його потрібно налаштувати для роботи на комп'ютері RaspberryPi 3. Обраний комп'ютер, який використовується у системі як головний сервер працює під управлінням операційної системи Raspbian що є модифікацією одного із найпоширеніших Linux дистрибутива Debian. Дана модифікація операційної системи найкраще оптимізована для роботи даного комп'ютера у якості машини для підтримки клієнт-серверних сервісів, що у випадку розроблюваної системи потрібно.

4.1.1 Інтеграція модулів системи

Програмне забезпечення системи, написане на мові програмування Python. Дана мова програмування використовується для розробки серверних додатків. Програмне забезпечення, що написане на мові Python добре підтримується обраною операційною системою [44].

Для коректної роботи програмного забезпечення системи із операційною системою комп'ютера, було обрано метод інтеграції коду додатку як окремого сервісу системи. Це досягається за допомогою вбудованого механізму багатьох Linux систем – “Systemd”. Кожен модуль системи має головний файл, що включає в себе сценарій запуску сервісу, місце розташування файлів конфігурації, спеціальні файли із кодом модулю, що відповідають за роботу сервісу у мережі. Запуск сервісу можливий одразу після завантаження операційної системи. Після запуску сервісу він має власні процеси, що запусненні у середовищі операційної системи, завдяки вбудованих команд

можливо контролювати статус даних процесів, що дає можливість відслідковувати роботу модулю. У кожному модулю присутня система логування. Кожна дія користувача, а також кожен запит між модулями записується у окремий файл журналу, приклади файлів журналів показано у таблиці 4.1. Система логування дозволяє бачити всі дії, що були виконані системою, це дозволяє у випадку збою системи швидко знайти корінь проблеми [45].

Таблиця 4.1 – Приклади файлів журналів модулів системи

Сервіс (модуль системи)	Розташування файлу журналу у файловій системі ОС	Приклад запису у файлі журналу
Вебсервер	/var/log/nginx/access.log	192.168.0.22 - - [06/Nov/2019:19:10:38 +0600] "POST 0.0.0.0:7557/relay02/switchHTTP/1.1" 200 177 "-" "Mozilla/5.0 (iPhone; CPU iPhone OS 6_0 like Mac OS X) AppleWebKit/536.26 (KHTML, like Gecko) Version/6.0 Mobile/10A5376e Safari/8536.25
Модуль захисту	/var/log/shouse/securitysrv.log	DEBUG [191106 19:10:39] Got request from user: 187VGZxDKSVsakkeaccde43dg rel02 act-switch HTTP/1.1 200 0.00225 auth-ok REDIRECTED dest-0.0.0.0:55577
Модуль обробки запитів	/var/log/shouse/reqhandlersrv.log	DEBUG [191106 19:10:41] POST X-Forwarded-For=192.168.0.22 /relay02/switch HTTP/1.1 200 0.00148cur_st:OFFsw_to:ON

Обраний метод інтеграції також дозволяє змінювати основні налаштування системи в одному файлі, що спрощує механізм конфігурації.

Після зміни будь якої конфігурації модулю необхідно провести перезавантаження сервісу за допомогою стандартних команд операційної системи, що наведені у таблиці 4.2.

Розробка програмного забезпечення системи проводилась із урахуванням особливостей системи Raspbian. Для реалізації методів було використано лише вбудовані бібліотеки мови програмування Python версії 3.5. Програмний код усіх модулів було оптимізовано враховуючи технічні характеристики одноплатного комп'ютеру RaspberryPi 3 B+ [46].

Таблиця 4.2 – Приклади команд для управління сервісами (модулями системи) за допомогою команд операційної системи

Команда	Дія
<code>service nginx stop/start/restart/reload/status</code>	Зупинити/запустити/перезапустити/оновити файли конфігурації/показати статус вебсерверу
<code>service securitysrvstop/start/status</code>	Зупинити/запустити/показати статус модулю захисту системи
<code>service reqhandlersrvstop/start/status</code>	Зупинити/запустити/показати статус модулю обробки запитів

4.1.2 Інтеграція програмного коду прошивки реле

Апаратне забезпечення реле, що використовуються у системі дозволяють використовувати стороннє програмне забезпечення. У даній системі у якості основного програмного забезпечення для WiFi реле SonoffBasic було обрано ПЗ “Tasmota”. Обране програмне дає можливість використовувати додаткову функціональність пристрою та за рахунок цього розширити можливості реле.

ПЗ SonoffTasmota можна завантажити на більшість пристроїв Sonoff. У даній системі використовуються реле марки Basic. Даний тип реле працює під управлінням мікроконтролера із вбудованим WiFi модулем ESP8266.

Це означає, що даний тип реле має великий потенціал у розробці даної системи [47].

До основних переваг програмного забезпечення Sonoff Tasmota належать:

- повна сумісність із локальною мережею;
- підтримка протоколу MQTT;
- підтримка HTTP-запитів;
- використання додаткових виводів GPIO;
- підтримка датчиків;
- оновлення методом OTA;
- емуляція Belkin WeMo та Phillips Hue.

Ці всі можливості включені у програмне забезпечення, що є безкоштовним і його можна завантажити із відкритих ресурсів.

Процес інтеграції нового програмного забезпечення у реле досить простий і займає не багато часу. Завантаження нового програмного забезпечення проводиться через вбудовані GPIO виводи, що розташовані під корпусом реле. Для завантаження програмного реле потрібне середовище розробки Arduino IDE та кабель UART. Увесь процес встановлення доступний на вебсайті проекту Tasmota, що дозволяє швидко і без помилок оновити ПЗ кожного реле [48].

Програмне забезпечення Sonoff Tasmota - це дуже швидкий і зручний спосіб управління пристроями Sonoff. Воно дає можливість використовувати протоколи MQTT та HTTP для управління реле, що робить його сумісним із більшістю операційних систем та вебсерверів. переходом до програмного забезпечення на основі Tasmota [49].

4.2 Тестування програмного забезпечення системи

Тестування є одним із найголовніших складових при розробці будь якого програмного чи апаратного забезпечення. Процес тестування являє собою

перевірку системи на коректність роботи із поставленими задачами, частіше за все, ці задачі описані у технічній документації до розробки, що називається «специфікація». У процесі тестування повинні бути знайдені невідповідності у поточній роботі системи і тій роботі, яка очікується від системи. Такі невідповідності називають «баги». Жодна система не може вважатися завершеною без попереднього тестування[50].

У процесі пошуку багів потрібно дотримуватись певних визначених принципів тестування, це дозволяє систематизувати процес тестування і за рахунок цього знайти максимальну кількість багів.

На даному етапі потрібно визначити, описати та виконати основний набір тестів для розробленої системи[51].

4.2.1 Тестування інтерфейсу для адміністратора системи

Для того щоб впевнитись що розроблений інтерфейс для адміністратора системи працює коректно, було розроблено наступний перелік тестів, що описано у таблиці 4.3[51].

Таблиця 4.3 – Основний перелік тестів для перевірки інтерфейсу для адміністратора системи.

№	Назва тесту	Кроки для відтворення	Результат, що очікується
1	2	3	4
1	Відкриття сторінки авторизації адміністратора	Запустити сервіси системи, у браузері перейти на головну сторінку	Поява сторінки авторизації у вікні браузера
2	Доступність введення даних у поля форми	Ввести електронну пошту та пароль адміністратора	Символи з'являються у полях по мірі введення

	авторизації		
--	-------------	--	--

Продовження таблиці 4.3

1	2	3	4
3	Перевірка відправки даних форми на сервер	Натиснути кнопку "Sign in"	У консолі для розробників браузера видно код відповіді серверу «HTTP 1.1/ 200 - Allowed», атака жоголовної вікно інтерфейсу
4	Перевірка роботи пункту головного меню «Addnewuser»	Натиснути на пункт меню «Addnewuser»	Відкрилась сторінка для додавання нового користувача у систему
5	Перевірка роботи пункту головного меню «Addnewrelay»	Натиснути на пункт меню «Addnewrelay»	Відкрилась сторінка для додавання нового реле у систему
6	Доступність введення даних у поля форми додавання	Ввести параметри реле	Символи з'являються у полях по мірі введення
7	Некоректні дані у полі «Логін»	Ввести символи кирилиці у поле «Логін»	Символи неможливо ввести
8	Коректні дані у полі «Логін»	Ввести символи латинського алфавіту, а також цифри.	Символи вводяться
9	Перевірка відправки даних форми на сервер	Натиснути кнопку "Submit"	У консолі для розробників браузера видно код відповіді серверу «HTTP 1.1/ 200 - Allowed».

Продовження таблиці 4.3

10	Перевірка запису нових даних у базу даних	За допомогою інтерфейсу командного рядка увійти в базу системи та перевірити наявні дані	Нові дані записались у базу даних
11	Перевірка запису паролю в базу у зашифрованому вигляді	За допомогою інтерфейсу командного рядка увійти в базу даних системи та перевірити тип запису паролю	Пароль записано у зашифрованому вигляді

4.2.2 Тестування модулю захисту системи

Для того щоб впевнитись що розроблений модуль захисту системи працює згідно із поставленими задачами, підготовлено наступний список тестів, які представлені у таблиці 4.4.

Таблиця 4.4 – Список тестів для перевірки модулю захисту системи

№	Назва тесту	Кроки для відтворення	Результат, що очікується
1	Відправка запиту від зареєстрованого користувача	Увійти до системи із даними користувача, що є у базі даних та обрати піктограму реле у вебінтерфейсі	У журналах системи з'явиться запис із кодом відповіді «HTTP 1.1/ 200 - Allowed»

Продовження таблиці 4.4

№	Назва тесту	Кроки для відтворення	Результат, що очікується
2	Відправка запиту від не зареєстрованого користувача	Увійти до системи із даними користувача, що немає у базі даних та обрати піктограму реле у вебінтерфейсі	У журналах системи з'явиться запис із кодом відповіді «HTTP 1.1/ 403 - Forbidden»
4	Блокування повторного запиту від незареєстрованого користувача	Увійти до системи із даними користувача, що немає у базі даних та декілька разів обрати піктограму реле у вебінтерфейсі	У журналах системи з'явиться запис із кодом відповіді «HTTP 1.1/ 410 - Blocked»
5	Перевірка запуску сервісу	Запустити сервіс командою «service securitysrvstart»	На екрані з'явиться повідомлення «OK: Security service started. Port 5775 »
6	Перевірка статусу сервісу	Перевірити статус сервісу командою «service securitysrvstatus»	На екрані з'явиться повідомлення «OK: Security service already started. Port 5775 Uptime HH:MM:SS »
7	Перевірка зупинки сервісу	Зупинити роботу сервісу командою «service securitysrvstop»	На екрані з'явиться повідомлення «OK: Security service stopped»

Продовження таблиці 4.4

№	Назва тесту	Кроки для відтворення	Результат, що очікується
8	Перевірка роботи системи після зупинки сервісу	Зупинити роботу сервісу командою «servicesecuritysrvstop», відкрити вебінтерфейс системи	У вебінтерфейсі з'явиться повідомлення «Security service stopped, please start service for use system»
9	Перевірка генерації ключа для користувача	За допомогою інтерфейсу командного рядка увійти в базу системи та перевірити тип запису ключа	Ключ записаний у форматі «username&md5password&macaddress»
10	Відправка запиту із вебінтерфейсу із усіма параметрами	За допомогою програми «Postman» відправити запит до модулю захисту із обов'язковими параметрами	У вікні відповіді від серверу, з'явиться повідомлення із кодом «HTTP 1.1/ 200- Allowed»
12	Відправка запиту із вебінтерфейсу без одного із параметрами	За допомогою програми «Postman» відправити запит до модулю захисту без одного із обов'язкових параметрами	У вікні відповіді від серверу, з'явиться повідомлення із кодом «HTTP 1.1/ 400 – BadRequest»

4.2.3 Тестування модулю обробки запитів

Щоб перевірити коректність роботи модулю захисту обробки запитів, підготовлено наступний список тестів, які представлені у таблиці 4.5.

Таблиця 4.5 – Основний перелік тестів для перевірки модулю обробки запитів.

№	Назва тесту	Кроки для відтворення	Результат, що очікується
1	2	3	4
1	Перевірка запуску сервісу	Запустити сервіс командою «servicereqhandlersrvstart»	На екрані з'явиться повідомлення «OK: Request-handler servicestarted. Port 55577 »
2	Перевірка статусу сервісу	Перевірити статус сервісу командою «servicereqhandlersrvstatus»	На екрані з'явиться повідомлення «OK: Request-handler alreadystarted. Port 55577 Uptime HH:MM:SS »
3	Перевірка зупинки сервісу	Зупинити роботу сервісу командою «servicereqhandlersrvstop»	На екрані з'явиться повідомлення «OK: Request-handler service stopped»
4	Перевірка роботи системи після зупинки сервісу	Зупинити роботу сервісу командою «servicereqhandlersrvstop», відкрити вебінтерфейс системи	У вебінтерфейсі з'явиться повідомлення «Request-handler service stopped, please start service for use system»

Продовження таблиці 4.5

1	2	3	4
5	Обробка запитів із коректними параметрами	За допомогою програми «Postman»відправити запит на адрес 192.168.0.xxx:55577 із вказанням обов'язкових параметрів реле	У вікні відповіді від серверу, з'явиться повідомлення із кодом «HTTP 1.1/ 200 - Allowed », піктограма статусу реле зміниться у вебінтерфейсі
6	Обробка запитів із не коректними параметрами	За допомогою програми «Postman»відправити запит на адрес 192.168.0.xxx:55577 із некоректними реле	У вікні відповіді від серверу, з'явиться повідомлення із кодом «HTTP 1.1/ 400–BadRequest », піктограма статусу реле не зміниться у вебінтерфейсі
7	Обробка запитів до неіснуючого реле	За допомогою програми «Postman»відправити запит на адрес 192.168.0.xxx:55577 із вказанням IPадреси неіснуючого реле	У вікні відповіді від серверу, з'явиться повідомлення із кодом «HTTP 1.1/ 404–NotFound », у вебінтерфейсі не з'явиться ніяких змін
8	Стрес-тест модулю	Відіслати до 50ти запитів за секунду на одне реле	Модуль припине свою роботу по причині перевантаження

5 ЕКОНОМІЧНИЙ РОЗДІЛ

5.1 Технологічний аудит розробленої системи «розумний будинок» на базі Raspberry Pi 3 із використанням модулів Arduino

Як було зазначено у попередніх розділах роботи, сьогодні багато різних компаній розробляють так звані «розумні будинки» для покращення життя людини. Кожна із розробок має свою унікальність, використовує певні принципи автоматизації та застосовується під конкретні потреби кінцевих споживачів. Автоматизація приватного будинку приводить до зменшення витрат часу на обслуговування будинку, зменшення вартості комунальних послуг, підвищення зручності та надійності систем функціонування будинку[2, 52].

У загальному вигляді концепція «розумний будинок» являє собою спеціальну систему, вбудовану в житлове приміщення (квартиру або будинок) з метою забезпечення всім мешканцям цього будинку безпеки, комфорту та раціонального використання ресурсів.

Сьогодні існує велика кількість комплексних систем «розумний будинок», більшість з яких є надзвичайно складними в установці, монтажі, підключенні, але основним недоліком таких систем є велика ціна, що автоматично робить їх недоступними для пересічного громадянина.

Тому у виконаній нами роботі була поставлена задача спростити процеси передачі та обробки даних системи «розумний будинок», суттєво зменшити складність і вартість системи та суттєво підвищити швидкодію її роботи, забезпечити надійність виконання повсякденних функцій, зменшити витрати на встановлення та підключення системи «розумний будинок».

Під час проведення досліджень нами було спроектовано апаратне забезпечення для системи «розумний будинок»; реалізовано програмний додаток з використанням досліджених технологій; проведено інтеграцію апаратного та програмного забезпечення і виконано тестування системи.

В результаті було розроблено програмно-апаратний комплекс, який є достатньо простим порівняно з аналогами, суттєво збільшує кількість кінце-вих користувачів і створений таким чином, що може бути модифікований та удосконалений відповідно до інтеграції нових елементів будинку[53].

Для оцінювання технічного рівня та комерційного потенціалу нашої розробки проведемо її технологічний аудит. Для проведення технологічного аудиту було залучено трьох експертів, які мають необхідні знання і досвід роботи в досліджуваній області. Експертами були запрошені: к.т.н., доцент Кривогубченко С.Г., к.т.н. Кулик Я.А. та к.т.н., доцент Папінов В.М.

Оцінювання технічного рівня та комерційного потенціалу розробки здійснювалося за критеріями, які наведені у таблиці 5.1.

Таблиця 5.1 – Критерії оцінювання технічного рівня та комерційного потенціалу розробки і їх можлива бальна оцінка

Бали (за 5-ти бальною шкалою)					
Критерій	0	1	2	3	4
1	2	3	4	5	6
Технічна здійсненність концепції:					
1	Достовірність концепції не підтверджена	Концепція підтверджена експертними висновками	Концепція підтверджена розрахунками	Концепція перевірена на практиці	Перевірено роботоздатність продукту в реальних умовах
Ринкові переваги (недоліки):					
2	Багато аналогів на малому ринку	Мало аналогів на малому ринку	Кілька аналогів на великому ринку	Один аналог на великому ринку	Продукт не має аналогів на великому ринку

Продовження таблиці 5.1

1	2	3	4	5	6
3	Ціна продукту значно вища за ціни аналогів	Ціна продукту дещо вища за ціни аналогів	Ціна продукту приблизно дорівнює цінам аналогів	Ціна продукту дещо нижче за ціни аналогів	Ціна продукту значно нижче за ціни аналогів
4	Технічні та споживчі властивості продукту значно гірші, ніж в аналогів	Технічні та споживчі властивості продукту трохи гірші, ніж в аналогів	Технічні та споживчі властивості продукту на рівні аналогів	Технічні та споживчі властивості продукту трохи кращі, ніж в аналогів	Технічні та споживчі властивості продукту значно кращі, ніж в аналогів
5	Експлуатаційні витрати значно вищі, ніж в аналогів	Експлуатаційні витрати дещо вищі, ніж в аналогів	Експлуатаційні витрати на рівні експлуатаційних витрат аналогів	Експлуатаційні витрати трохи нижчі, ніж в аналогів	Експлуатаційні витрати значно нижчі, ніж в аналогів
Ринкові перспективи					
6	Ринок малий і не має позитивної динаміки	Ринок малий, але має позитивну динаміку	Середній ринок з позитивною динамікою	Великий стабільний ринок	Великий ринок з позитивною динамікою

Продовження таблиці 5.1

1	2	3	4	5	6
7	Активна конкуренція великих компаній на ринку	Активна конкуренція	Помірна конкуренція	Незначна конкуренція	Конкурентів немає
Практична здійсненність					
8	Відсутні фахівці як з технічної, так і з комерційної реалізації ідеї	Необхідно наймати фахівців або витратити значні кошти та час на навчання наявних фахівців	Необхідне незначне навчання фахівців та звільнення їх штату	Необхідне незначне навчання фахівців	Є фахівці з питань як з технічної, так і з комерційної реалізації ідеї
9	Потрібні значні фінансові ресурси, які відсутні. Джерела фінансування ідеї відсутні	Потрібні незначні фінансові ресурси. Джерела фінансування відсутні.	Потрібні значні фінансові ресурси. Джерела фінансування є	Потрібні незначні фінансові ресурси. Джерела фінансування є	Не потребує додаткового фінансування.

Продовження таблиці 5.1

1	2	3	4	5	6
10	Необхідна розробка нових матеріалів	Потрібні матеріали, що використовуються у військово-промисловому комплексі	Потрібні дорогі матеріали	Потрібні досяжні та дешеві матеріали	Всі матеріали для реалізації ідеї відомі та давно використовуються у виробництві
11	Термін реалізації ідеї більший за 10 років	Термін реалізації ідеї більший за 5 років. Термін окупності інвестицій більше 10-ти років	Термін реалізації ідеї від 3-х до 5-ти років. Термін окупності інвестицій більше 5-ти років	Термін реалізації ідеї менше 3-х років. Термін окупності інвестицій від 3-х до 5-ти років	Термін реалізації ідеї менше 3-х років. Термін окупності інвестицій менше 3-х років
12	Необхідна розробка регламентних документів та тримання великої кількості дозвільних документів на виробництво та реалізацію продукту	Необхідно отримання великої кількості дозвільних документів на виробництво та реалізацію продукту, що вимагає значних коштів та часу	Процедура отримання дозвільних документів для виробництва та реалізації продукту вимагає незначних коштів та часу	Необхідно тільки повідомлення відповідним органам про виробництво та реалізацію продукту	Відсутні будь-які регламентні обмеження на виробництво та реалізацію продукту

Результати оцінювання технічного рівня та комерційного потенціалу розробки зведено в таблицю 5.2.

Таблиця 5.2 – Результати оцінювання технічного рівня та комерційного потенціалу розробки

Критерії	Прізвище, ініціали експерта		
	Кривогубченко С.Г.	Кулик Я.А.	Папінов В.М.
	Бали, виставлені експертами:		
1	3	3	2
2	2	2	2
3	2	2	2
4	3	3	2
5	2	2	2
6	2	3	3
7	2	2	2
8	2	2	2
9	3	3	3
10	2	2	2
11	2	2	3
12	3	3	2
Сума балів	СБ ₁ = 28	СБ ₂ = 29	СБ ₃ = 27

Для визначення технічного рівня та комерційного потенціалу розробки скористаємося таблицею 5.3, в якій представлено узагальнені результати можливого експертного оцінювання.

Таблиця 5.3 – Технічні рівні та комерційний потенціал розробки

Середньоарифметична сума балів $\overline{СБ}$, розрахована на основі висновків експертів	Технічні рівні та комерційний потенціал розробки
0 – 10	Низький
11 – 20	Нижче середнього
21 – 30	Середній
31 – 40	Вище середнього
41 – 48	Високий

Далі розрахуємо середньоарифметичне значення експертних оцінок. Середньоарифметичне значення $\overline{СБ}$:

$$СБ_{\text{сер}} = \frac{СБ_1 + СБ_2 + СБ_3}{3} = \frac{28 + 29 + 27}{3} = 28.$$

Порівнявши результати експертного оцінювання технічного рівня та комерційного потенціалу нашої розробки з даними таблиці 4.3, можна зробити висновок, що технічний рівень та комерційний потенціал нашої розробки знаходиться у межах «середній – вище середнього».

Це пояснюється тим, що нами запропоновано такий метод захисту веб-серверу системи, який присвоює кожному пристрою користувача унікальний ідентифікатор, що на етапі формування враховує унікальний адрес мережевого адаптера пристрою, а також логін та пароль користувача, що робить систему незалежною і дає змогу підвищити захищеність системи, не зменшуючи її швидкодію за рахунок того, що унікальний ідентифікатор використовується при кожному запиті користувача до серверу і у випадку не співпадіння – алгоритм методу блокує доступ пристрою до веб-серверу, а у разі повторних запитів – сервер блокує IP адресу пристрою, що ускладнює спроби перенавантаження серверу однотипними запитами.

5.2 Розрахунок витрат на виконання роботи

Витрати на розробку системи «розумний будинок» складаються із таких статей:

1). Основна заробітна плата Z_o виконавців, яка розраховується за формулою (5.1):

$$Z_o = \frac{M}{T_p} \cdot t \text{ [грн]}, \quad (5.1)$$

де M – місячний посадовий оклад конкретного виконавця, грн. В 2019 році величини окладів знаходиться в межах (4173...15550) грн/місяць;

T_p – число робочих днів в місяці; прийmemo $T_p = 21$ день;

t – число робочих днів роботи виконавців.

Зроблені розрахунки зведемо до таблиці 5.4:

Таблиця 5.4 – Основна заробітна плата виконавців

Найменування посади виконавця	Місячний посадовий оклад, грн	Оплата за робочий день, грн	Число днів роботи	Витрати на оплату праці, грн
1. Науковий керівник роботи	12000	571,43	25 годин	2380,95
2. Студент-магістрант	1800	85,71	49	4199,79
3. Консультант з економічної частини	11400	542,86	2,5 годин	226,19
4. Інші консультанти	8000	380,95	1 день: 6 годин	380,95
Загалом				$Z_o = 7187,88 \approx 7200$ грн

2). Додаткова заробітна плата Z_d виконавців розраховується як (10...12)% від величини їх основної заробітної плати, тобто (5.2):

$$Z_d = (0,1...0,12) \cdot Z_o. \quad (5.2)$$

Для нашого випадку:

$$Z_d = 0,12 \times 7200 = 864 \text{ [грн]}.$$

3). Нарахування на заробітну плату $H_{зп}$ виконавців розраховуються за формулою (5.3):

$$H_{зп} = (Z_o + Z_d) \cdot \frac{\beta}{100}, \quad (5.3)$$

де β – ставка обов'язкового єдиного внеску на державне соціальне страхування, %. $\beta = 22\%$. Тоді:

$$H_{зп} = (7200 + 864) \times 0,22 = 1774,08 \approx 1775 \text{ [грн]}.$$

4). Амортизація основних засобів A , які використовувались під час виконання магістерської роботи, може бути розрахована за формулою (5.4):

$$A = \frac{Ц \cdot H_a}{100} \cdot \frac{T}{12} \text{ [грн]}, \quad (5.4)$$

де $Ц$ – загальна балансова вартість основних засобів, грн;

H_a – річна норма амортизаційних відрахувань. Для нашого випадку можна прийняти, що $H_a = (5...25)\%$;

T – термін використання основних засобів, місяці.

Зроблені розрахунки зведено в таблицю 5.5.

Таблиця 5.5 – Розрахунок амортизаційних відрахувань

Найменування обладнання, приміщень тощо	Балансова вартість, грн.	Норма амортизації, %	Термін використання, міс.	Величина амортизаційних відрахувань, грн
1. Комп'ютерна техніка, принтери тощо	30000	20	2,2	1100
2. Приміщення університету та кафедри	10000	5	2,2	91,67
Всього				1191,67 \approx 1192

5). Витрати на матеріали M розраховуються за формулою (5.5):

$$M = \sum_1^n H_i \cdot \Pi_i \cdot K_i - \sum_1^n V_i \cdot \Pi_v \text{ [грн]}, \quad (5.5)$$

де H_i – витрати матеріалу i -го найменування, кг; Π_i – вартість матеріалу i -го найменування; K_i – коефіцієнт транспортних витрат, $K_i = (1,1 \dots 1,15)$; V_i – маса відходів матеріалу i -го найменування; Π_v – ціна відходів матеріалу i -го найменування; n – кількість видів матеріалів.

6). Витрати на комплектуючі K розраховуються за формулою (5.6):

$$K = \sum_1^n H_i \cdot \Pi_i \cdot K_i \text{ [грн]}, \quad (5.6)$$

де H_i – кількість комплектуючих i -го виду, шт.; Π_i – ціна комплектуючих i -го виду; K_i – коефіцієнт транспортних витрат, $K_i = (1,1 \dots 1,15)$; n – кількість видів комплектуючих.

При розробці системи «розумний будинок» витрати на матеріали та комплектуючі склали приблизно 1000 грн.

7). Витрати на силову електроенергію V_e розраховуються за формулою (5.7):

$$V_e = \frac{V \cdot \Pi \cdot \Phi \cdot K_{\Pi}}{K_d}, \quad (5.7)$$

де V – вартість 1 кВт-год. електроенергії, в 2019 р. $V \approx 2,2$ грн/кВт;

Π – установлена потужність обладнання, кВт; $\Pi = 0,95$ кВт;

Φ – фактична кількість годин роботи обладнання, годин.

Прийmemo, що $\Phi = 135$ годин;

K_{Π} – коефіцієнт використання потужності; $K_{\Pi} < 1 = 0,9$.

K_d – коефіцієнт корисної дії, $K_d = 0,75$.

Тоді витрати на електроенергію будуть дорівнювати:

$$V_e = \frac{V \cdot \Pi \cdot \Phi \cdot K_{\Pi}}{K_d} = \frac{2,2 \cdot 0,95 \cdot 135 \cdot 0,8}{0,75} = 300,96 \approx 301 \text{ [грн]}.$$

8). Інші витрати $V_{\text{інш}}$ (опалення, освітлення, прибирання тощо) можна прийняти як (100...300)% від суми основної заробітної плати виконавців (5.8):

$$V_{\text{інш}} = (0,1 \dots 3) \times 3_0. \quad (5.8)$$

Для нашого випадку отримаємо:

$$V_{\text{інш}} = 1,5 \times 7200 = 10800 \text{ [грн]}.$$

9). Сума всіх попередніх статей витрат складає витрати на виконання цієї роботи безпосередньо магістрантом – V .

$$B = 7200 + 864 + 1775 + 1192 + 1000 + 301 + 10800 = 23132 \text{ [грн]}.$$

10). Загальні витрати Z_{AG} на завершення виконаної нами роботи з розробки системи «розумний будинок» розраховуються за формулою (5.9):

$$Z_{AG} = \frac{B}{\beta}, \quad (5.9)$$

де β – коефіцієнт, який характеризує етап (стадію) виконання нашої роботи.

Прийmemo, що $\beta \approx 0,6$ [23], оскільки робота потребує свого продовження та завершення.

$$\text{Тоді: } Z_{AG} = \frac{23132}{0,6} = 38553,33 \text{ [грн]} \text{ або приблизно } 39 \text{ тисяч гривень.}$$

Тобто прогнозовані витрати на розробку системи «розумний будинок» становлять приблизно 39 тисяч гривень.

5.3 Розрахунок економічного ефекту від можливої комерціалізації даної розробки «розумний будинок»

Проведені дослідження ринку «розумних будинків» в Україні показали, що попит на розробки, подібні до нашої, становить від 20-40 штук на рік в ціновому діапазоні від 30 тисяч грн до 100 тисяч грн. За середні значення візьmemo 30 штук за ціною 50 тисяч грн.

Оскільки розроблена нами система «розумний будинок» набагато дешевша за аналоги (зі збереженням багатьох можливостей цих аналогів), то її можна буде реалізовувати на ринку дещо дешевше (наприклад, за 45 тисяч грн, тобто на 5 тисяч грн дешевше), що спричинить зростання попиту на нашу розробку [54].

Не викликає сумнівів, що наша система «розумний будинок» буде користуватися підвищеним попитом на ринку принаймні протягом 3-х років після її впровадження, тобто, якщо наша розробка буде впроваджена з 1 січня 2021 року (оскільки ця робота потребує певного завершення), то її результати будуть виявлятися протягом 2021-го, 2022-го та 2023-го років [55].

Прогноз зростання попиту на нашу розробку складає по роках:

1-й рік після впровадження (2021 р.) – приблизно +20 шт.;

2-й рік після впровадження (2022 р.) – приблизно +40 шт.;

3-й рік після впровадження (2023 р.) – приблизно +60 шт.

Можливе збільшення чистого прибутку $\Delta\Pi_i$, що його може отримати потенційний інвестор від фінансування та впровадження розробленої нами системи «розумний будинок» становитиме (5.10) [2]:

$$\sum_1^n (\Delta C_0 \cdot N + C_0 \cdot \Delta N)_i \cdot \lambda \cdot \rho \cdot \left(1 - \frac{\nu}{100}\right), \quad (5.10)$$

де ΔC_0 – покращення основного якісного показника від впровадження результатів розробки у цьому році. Таким показником є зміна ціни нової розробки відносно аналогів, грн.

Для нашого випадку це буде $\Delta C_0 = (45-50) = -5$ тис. грн, тобто наша розробка буде продаватися дешевше, ніж аналоги;

N – основний кількісний показник, який визначає обсяг діяльності у цьому році до впровадження результатів розробки; $N = 30$ шт.;

ΔN – покращення основного кількісного показника від впровадження результатів розробки. Таке покращення по роках становитиме, відповідно: +20 (2021 р.) шт.; +40 (2022 р.) шт. та +60 (2023 р.) шт.;

C_0 – основний якісний показник, який визначає обсяг діяльності (тобто ціну) у році після впровадження результатів розробки, грн; $C_0 = 45$ тис. грн;

n – кількість років, протягом яких очікується отримання позитивних результатів від впровадження розробки;

λ – коефіцієнт, який враховує сплату податку на додану вартість;
 $\lambda = 0,8333$;

ρ – коефіцієнт, який враховує рентабельність продукту. Рекомендується приймати $\rho = (0,2 \dots 0,5)$; візьмемо $\rho = 0,45$;

ν – ставка податку на прибуток. У 2019 році $\nu = 18\%$.

Тоді збільшення чистого прибутку для потенційного інвестора $\Delta\Pi_1$ протягом першого року від реалізації нашої розробки (2021 р.) складатиме:

$$\Delta\Pi_1 = [-5 \cdot 30 + 45 \cdot 20] \cdot 0,8333 \cdot 0,45 \cdot \left(1 - \frac{18}{100}\right) \approx 231 \text{ тис. [грн]}.$$

Величина чистого прибутку $\Delta\Pi_2$ для потенційного інвестора від можливого впровадження нашої розробки протягом другого (2022 р.) року складе:

$$\Delta\Pi_2 = [-5 \cdot 30 + 45 \cdot 40] \cdot 0,8333 \cdot 0,45 \cdot \left(1 - \frac{18}{100}\right) \approx 507 \text{ тис. [грн]}.$$

Величина чистого прибутку $\Delta\Pi_3$ для потенційного інвестора від можливого впровадження нашої розробки протягом третього (2023 р.) року складе:

$$\Delta\Pi_3 = [-5 \cdot 30 + 45 \cdot 60] \cdot 0,8333 \cdot 0,45 \cdot \left(1 - \frac{18}{100}\right) = 784 \text{ тис. [грн]}.$$

Далі розрахуємо теперішню вартість інвестицій PV, що повинні бути вкладені в нашу розробку: $PV = (2 \dots 5) \times 39$.

Для нашого випадку $PV = (2 \dots 5) \times 39 = 5 \times 39 = 195$ тис. грн.

Розраховуємо абсолютний ефект вкладених інвестицій $E_{\text{абс}}(5.11)$:

$$E_{abc} = \text{ПП} - PV, \quad (5.11)$$

де ПП – приведена вартість всіх можливих чистих прибутків від можливого впровадження нашої розробки, грн;

PV – теперішня вартість інвестицій $PV = 195$ тис. грн.

Приведена вартість всіх чистих прибутків ПП розраховується наступним чином (5.12):

$$\text{ПП} = \sum_1^t \frac{\Delta\Pi_i}{(1 + \tau)^t}, \quad (5.12)$$

де $\Delta\Pi_i$ – збільшення чистого прибутку у кожному із років, протягом яких виявляються результати виконаної та впровадженої роботи, грн;

t – період часу, протягом якого виявляються результати впровадженої роботи, роки. Для нашого випадку $t = 3$ роки;

τ – ставка дисконтування. Керуючись даними статистики за 2019 рік та прогнозами експертів, приймемо ставку дисконтування $\tau = 0,11$ (11%);

t – період часу від моменту отримання прибутків до початку впровадження розробки.

Тоді приведена вартість всіх чистих прибутків ПП, що їх може отримати потенційний інвестор від можливого впровадження нашої розробки, складе:

$$\text{ПП} = \frac{231}{(1+0,11)^2} + \frac{507}{(1+0,11)^3} + \frac{784}{(1+0,11)^4} \approx 187 + 371 + 516 = 1074 \text{ тис. [грн]}.$$

Абсолютний ефект від можливого впровадження нашої розробки (при прогнозованому ринку збуту) за три роки (2021-2023 рр.) складе:

$$E_{abc} = 1074 - 195 = 879 \text{ тис. грн. або } 293 \text{ тис. грн щорічно.}$$

Оскільки $E_{abc} > 0$, то вкладання коштів в нашу розробку може бути доцільним.

Далі розрахуємо внутрішню норму дохідності E_v вкладених коштів:

$$E_B = \sqrt[T_{\text{ж}}]{1 + \frac{E_{\text{абс}}}{PV}} - 1, \quad (5.13)$$

де $E_{\text{абс}}$ – абсолютний ефект вкладених коштів; $E_{\text{абс}} = 879$ тис. грн;
 PV – теперішня вартість початкових інвестицій $PV = 195$ тис. грн;
 $T_{\text{ж}}$ – життєвий цикл розробки, роки. $T_{\text{ж}} = 4$ роки (з 2020 р. по 2023 р.).

Для нашого випадку:

$$E_B = \sqrt[4]{1 + \frac{879}{195}} - 1 = \sqrt[4]{1 + 4,507} - 1 = \sqrt[4]{5,507} - 1 = 1,5318 - 1 = 0,5318 = 53,18\%.$$

Далі визначимо ту мінімальну дохідність, нижче за яку інвестор не буде вкладати кошти в нашу розробку. Мінімальна дохідність або мінімальна (бар'єрна) ставка дисконтування $\tau_{\text{мін}}$ визначається за формулою (5.14):

$$\tau = d + f, \quad (5.14)$$

де d – середньозважена ставка за депозитними операціями в комерційних банках; в 2019 році в Україні $d = (0,15...0,50)$;

f – показник, що характеризує ризикованість вкладень; зазвичай, $f = (0,05...0,5)$, але може бути і значно більше. Прийmemo, що $f = 35\%$.

Для нашого випадку отримаємо:

$$\tau_{\text{мін}} = 0,15 + 0,35 = 0,50 \text{ або } \tau_{\text{мін}} = 50\%.$$

Оскільки величина $E_B = 53,18\% > \tau_{\text{мін}} = 50\%$, то потенційний інвестор може бути зацікавлений у фінансуванні нашої розробки.

Далі розраховуємо термін окупності коштів, вкладених у нашу розробку. Термін окупності $T_{\text{ок}}$ можна розрахувати за формулою:

$$T_{\text{ок}} = \frac{1}{E_{\text{в}}}. \quad (5.15)$$

Для нашого випадку термін окупності $T_{\text{ок}}$ коштів складе:

$$T_{\text{ок}} = \frac{1}{0,5318} \approx 1,88 \text{ років,}$$

що свідчить про потенційну доцільність комерціалізації нашої розробки.

Результати виконаної економічної частини магістерської кваліфікаційної роботи зведено у таблицю 5.6:

Таблиця 5.6 – Економічна частина роботи. Зведена таблиця

Показники	Задані у ТЗ	Досягнуті у магістерській кваліфікаційній роботі	Висновок
1. Витрати на розробку системи «розумний будинок»	Небільше 40 тис. грн	39 тис. грн.	Досягнуто
2. Абсолютний ефект від впровадження системи «розумний будинок», тис. грн	не менше 250 тис. грн за рік	293 тис. грн щорічно	Виконано
3. Внутрішня норма дохідності інвестицій, %	Не менше 50%	53,18%	Досягнуто
4. Термін окупності вкладених коштів, роки	до 3-х років	1,88 років	Виконано

Таким чином, основні техніко-економічні показники розробленої нами системи «розумний будинок», визначені у технічному завданні, виконані.

ВИСНОВКИ

В данній магістерській роботі розроблено систему «Розумний будинок» на базі Raspberry Pi 3 із використанням модулів Arduino. В роботі було досліджено основні поняття предметної галузі, розглянуто аналоги, спроектовано систему «Розумний будинок», спроектовано систему в цілому, розроблено компоненти системи та проведено інтеграцію апаратного та програмного забезпечення та виконання тестування системи. В економічній частині роботи проведено технологічний аудит, виконано розрахунок витрат та економічного ефекту від комерційного впровадження.

Було проведено розрахунки розробленої системи з економічної точки зору. Опишемо найголовніші економічні показники системи:

- прогнозовані витрати на розробку – 39 тис. грн;
- абсолютна ефективність розробки – 293 тис. грн щорічно;
- внутрішня норма дохідності інвестицій – 53,18%.
- термін окупності витрат для виробника – 1,88 років.

У ході виконання магістерської кваліфікаційної роботи було проведено огляд існуючих систем «Розумний будинок», наведено їх переваги та недоліки та виділено основні переваги система, що розроблялась.

До переваг розробленої системи можна віднести такі:

- низька собівартість системи, за рахунок низької вартості компонентів;
- простота у монтажі та налаштуванні;
- доступність більшості жителів країни;
- придатність до модернізації – дає можливість удосконалювати систему в подальшому;
- простота в управлінні та контролі за рахунок web-інтерфейсу

-автономне живлення системи, що дає можливість навіть при короткочасній відсутності напруги в мережі підтримувати працездатність системи;

До недоліків системи можна віднести наступні:

- низька швидкодія системи – після вибору користувачем певної дії у інтерфейсі, проходить близько 1.5 секунди, доки система виконає задану дію;

-реле, що використовуються у системі мають звук спрацювання;

- необхідність перезавантажувати систему приблизно один раз на тиждень.

Дані недоліки системи при необхідності можуть бути усунені за рахунок виконання системи як гнучкої розробки, що може дороблятися і постійно вдосконалюватися.

З врахуванням переваг системи, аналізу та економічних розрахунків дана система цілком реально зможе зайняти своє місце на ринку собі подібних, витіснити ряд конкурентів та користуватися попитом.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Розумний будинок: [Електронний ресурс]: //– Режим доступу до ресурсу: http://www.elektrik-v-dom.com.ua/ukr/umniy_dom.html(дата звернення 09.12.2019 р.).– Назва з екрану.
2. Розумний дім – технологія економії: [Електронний ресурс]: // rightchoice – Режим доступу до ресурсу: <http://right-choice.com.ua/rozumniy-dimtehologiya-ekonomiyi-zruchnosti-i-komfortu-visokogo-rivnya>(дата звернення 09.12.2019 р.).– Назва з екрану.
3. Афонин В.С.Организация информационного взаимодействия элементов системы "Умный дом"/ В.С.Афонин, А.Г.Зрюмова, А.А.Кузнецов, Р.А.Забеляев, Р.А.Дьякин // Ползуновский альманах: журнал/ Алтайский государственный технический университет им. И. И. Ползунова (Барнаул). - Барнаул : Изд-во Алт. гос. техн. ун-та им. И. И. Ползунова, 2017. - № 4/3 - С. 170-172.
4. Кирколуп Е.Р. Примеры реализации подсистем "Умного дома" на плате Arduino / Кирколуп Е.Р., Кудрявцева А.А. // Ползуновский альманах. : журнал/ Алтайский государственный технический университет им. И. И. Ползунова (Барнаул). - Барнаул : Изд-во Алт. гос. техн. ун-та им. И. И. Ползунова, 2019.- № 2. - С. 132-135.
5. Ramlee R. A. Smarthomesystemusingandroidapplication, InformationandCommunicationTechnology (ICoICT) /, M.A.Othman, M.H.Leong, M.M.IsmailandS.S.Ranjit, - InternationalConferenceof, Bandung, 2013. - P. 277-280.
6. Системи Розумного будинку. [Електронний ресурс]. - Режим доступу: <https://www.smarthouse.ua/ua/>(дата звернення 09.12.2019 р.). - Назва з екрану.
7. Керуючі пристрої «Розумного» будинку: [Електронний ресурс]: // sutem – Режим доступу до ресурсу: <http://sutem.com.ua/7121smartbus.php>(дата звернення 09.12.2019 р.). – Назва з екрану.
8. Матеріали XLVII науково-технічної конференції підрозділів Вінницького національного технічного університету (НТКП ВНТУ-2018)

[електронне мережне наукове видання]: збірник доповідей. – Вінниця, ВНТУ, 2018. – ISBN 978-966-641-737-7.

9. Орлюк Є. А. Розробка системи «Розумний будинок» на базі "Arduino" [Електронний ресурс] / Є.А. Орлюк. // Науково-технічна конференція факультету комп'ютерних систем і автоматики: електронне видання. – Електронні дані. – [Вінниця: ВНТУ, 23 березня, 2018.] – Режим доступу: <https://conferences.vntu.edu.ua/index.php/all-fksa/all-fksa-2018/paper/view/4541/4612>(дата звернення 09.12.2019 р.). - Назва з екрану.

10. Орлюк Є.А. Розробка системи «Розумний будинок» на базі RaspberryPi з використанням модулів Arduino. / Богач І.В., Орлюк Є.А., Коломієць О.В. / Сучасний рух науки: тези доп. VIII міжнародної науково-практичної інтернет-конференції, 3-4 жовтня 2019 р. – Дніпро, 2019. – Т.2. – 703 с. – С.646-649.

11. Історія розумний дім: [Електронний ресурс]: // wiki – Режим доступу до ресурсу: http://wiki.tntu.edu.ua/%D0%A0%D0%BE%D0%B7%D1%83%D0%BC%D0%BD%D0%B8%D0%B9_%D0%B4%D1%96%D0%BC(дата звернення 09.12.2019 р.). – Назва з екрану.

12. Jepson L. Arduino Cookbook / L. Jepson, B. Jepson. – Sebastopol: O'REILLY, 2011. – 658 с.

13. Гололобов В.Н. «Умный дом» своими руками. / В.Н. Гололобов – М.: ИТ Пресс, 2007. – 416 с.

14. Sainath T.N. Acoustic landmark detection and segmentation using the McAulay-Quatier sinusoidal model. / T.N.Sainath. - Master's thesis, MIT, 2015. – PP 10-17.

15. Система розумний дім: [Електронний ресурс]. - Режим доступу: <https://ecotown.com.ua/news/Systema-Rozumnyy-dimzmenshuye-vytraty-na-komposluhu-do-30/>(дата звернення 09.12.2019 р.).– Назва з екрану.

16. Функції та комфорт розумного дому: [Електронний ресурс]: // melask – Режим доступу: <http://melask.com.ua/rozumniy-dim/komfort.html>(дата звернення 09.12.2019 р.).- Назва з екрану.
17. T'oth L. OnNaiveBayesinspeechrecognition. InternationalJournalofAppliedMathematicsandComputerScience / L.T'oth, A.Kocsor, andJ.Csirik, 2005. - 15(2). – P.287–294.
18. Dickey N., HomeautomationusingCloudNetworkandmobiledevices, Southeastcon / N.Dickey, D.Banks, S.Sukittanon. - Proceedingsof IEEE, Orlando, FL, 2012. - P. 1-4.
19. Ransing R. Smarthomeforelderlycare, basedonWirelessSensorNetwork/ R.RansingandM.Rajput. - Nascent Technologies intheEngineeringField (ICNTE), InternationalConferenceon, NaviMumbai, 2015, P. 1-5.
20. Система CLAP. [Електронний ресурс]. - Режим доступу: <https://ub.com.ua/clap>.(дата звернення 09.12.2019 р.).- Назва з екрану.
21. Система CLAP. [Електронний ресурс]. - Режим доступу: <https://itc.ua/articles/perviy-vzglyad-na-sistemuumnogo-doma-clap/>(дата звернення 09.12.2019 р.).- Назва з екрану.
22. Полоцкий Р.Е.Что такое "Умный дом"? / Полоцкий Р.Е. //Алгоритм безопасности: журнал. - М. : Анна Поликанова, 2017. - № 4. - С. 4-7.
23. Jamil M.A. A pilotstudy: DevelopmentofhomeautomationsystemviaraspberryPi/ M.A.JamilandM.S.Ahmad. - BiomedicalEngineering (ICoBE), 2nd InternationalConferenceon, Penang, 2015. - P. 1-4.
24. GassonMark. D3.2: A studyon PKI andbiometrics / MarkGasson, MartinMeints, KevinWarwick // FIDIS deliverable, 2005. - № (3) 2, July.
25. Переваги «Розумного» будинку: [Електронний ресурс]: // klyuch – Режим доступу до ресурсу: <http://klyuch.com.ua/m/articles/economy/yakiperevagy-rozumnogo-domu-/>(дата звернення 09.12.2019 р.).– Назва з екрану.
26. Бармен С. Разработка правил информационной безопасности. / С.Бармен - М.: Вильямс, 2012. — 208 с. - ISBN 5-8459-0323-8.

27. Серикова М.В. К вопросу модульного построения систем обеспечения "Умного дома" / М.В.Серикова, В.А.Атрощенко, Н.Д.Чигликова // *Фундаментальные исследования.: научный журнал/ "Академия естествознания"*, издательский дом. – М.: Изд. дом Акад. естествознания, 2017. - № 10-1. - С. 45-50.

28. Сопер М. Э. Практические советы и решения по созданию « Умного дома » / М. Э. Сопер. – М.: НТ Пресс, 2007. – 432 с.

29. Тесля Е.А. «Умный дом» своими руками. Строим интеллектуальную цифровую систему в своей квартире / Е.А. Тесля – Санкт Петербург, 2008. – 224 с.

30. Хацко Д.И. Принципы и технологии построения "Умного дома" / Хацко Д.И., Зрюмова А.Г., Афонин В.С. // *Ползуновский альманах: журнал/ Алтайский государственный технический университет им. И. И. Ползунова (Барнаул)*. - Барнаул : Изд-во Алт. гос. техн. ун-та им. И. И. Ползунова, 2017. - № 4/3 - С. 202-205.

31. Харке В.Н. Умный дом. Объединение в сеть бытовой техники и систем коммуникаций в жилищном строительстве / В.Н. Харке– М.: Техносфера, 2006. – 292 с.

32. Ullah M. A. An Effective Approach to Build Smart Building Based on Internet of Things (IoT) / M.A.Ullah, A.R.Celik. - *Journal of Basic and Applied Scientific Research*, issues 6. - 2016. – P. 56-62.

33. Элсенпитер Т.Р., Велт Дж. Умный Дом строим сами / Т. Р. Элсенпитер, Дж Велт/ КУДИЦ-ОБРАЗ. 2015. – 384с.

34. Табунщик Г. В. Проектування та моделювання програмного забезпечення сучасних інформаційних систем / Г. В. Табунщик, Т.І. Каплієнко, О.А. Петрова – Запоріжжя : Дике Поле, 2016. – 250 с. ISBN 978-966-2752-07-0.

35. Ярочкин В.И. Информационная безопасность. / В.И. Ярочкин - М.: Изд-во "Академический проект", 2004. - 640 с.

36. Лапони́на О.Р. Основы сетевой безопасности: криптографические алгоритмы и протоколы взаимодействия. / О.Р. Лапони́на - М.: Изд-во "Интернет-университет информационных технологий - ИНТУИТ.ру", 2005. – 608 с.: ил.

37. Dombi J. Towards a general class of operators for fuzzy systems. / J. Dombi, IEEE Transaction on Fuzzy Systems, 2008. - 16(2):477–484.

38. Monk S. Programming Arduino Getting Started with Sketches / Simon Monk. – Chicago: McGraw-Hill Education, 2012. – 176 с.

39. Ковалев Р.Е. Управление системой "Умный дом" через мобильное устройство с учетом геоположения пользователя для управления отоплением в помещении / Ковалев Р.Е., Зрюмова А.Г., Зрюмов П.А. // Ползуновский альманах.: журнал/ Алтайский государственный технический университет им. И. И. Ползунова (Барнаул). - Барнаул : Изд-во Алт. гос. техн. ун-та им. И. И. Ползунова., 2017. - № 4/3 - С. 228-230.

40. Folea S. Smart home automation system using Wi-Fi low power devices, Automation Quality and Testing Robotics (AQTR) / S. Folea, D. Bordenca, C. Hotea and H. Valean. - IEEE International Conference on, Cluj-Napoca, 2012. - P. 569-574.

41. Шлюз в'язку «Розумного» будинку: [Електронний ресурс]: // sutem – Режим доступу: <http://sutem.com.ua/7141smartbus.php> (дата звернення 09.12.2019 р.). – Назва з екрану.

42. Felix C.. "Home automation using GSM," Signal Processing, Communication, Computing and Networking Technologies (ICSCCN) / C. Felix, I. Raglend. - International Conference on, Thuckafay, 2011 - P. 15-19.

43. Зовнішній датчик руху «Розумного» будинку: [Електронний ресурс]: // sutem – Режим доступу до ресурсу: <http://sutem.com.ua/7135smartbus.php> (дата звернення 09.12.2019 р.). – Назва з екрану.

44. Методичні рекомендації з комерціалізації розробок, створених в результаті науково-технічної діяльності – К.: Наказ Державного комітету

України з питань науки, інновацій та інформатики (Лист № 1/06-4-97 від 13.09.2010 р.).

45. Дашук Н.О. Система управления освещением в "Умном доме" / Н.О. Дашук // Молодой ученый: международный научный журнал/ "Издательство "Молодой ученый", о-во с ограниченной ответственностью. - Чита : Молодой ученый, 2017. - № 36 (170). - С. 18-19.

46. Пилон Дэн, Расс Майлз. Управление разработкой ПО. / Д. Пилон, М. Расс – Издательство «Питер», 2018. – 464 с. – ISBN 978-5-459-00522-6.

47. Программируемый выключатель света: [Электронный ресурс]: // Geektimes – Режим доступа до ресурсу: <https://geektimes.ru> (дата звернення 09.12.2019 р.). – Назва з екрану.

48. Gosztolya G. Using one-class classification techniques in the antiphoneme problem. / G. Gosztolya, A. B'anhalmi and L. T'oth. - In Proceedings of the Iberian Conference on Pattern Recognition and Image Analysis (IbPRIA), Porto, Portugal, 2009. - Volume LNCS 5524. – P. 433– 440.

49. Christophe L. Reducing Computational and Memory Cost for Cellular Phone Embedded Speech Recognition System [J]. Proceedings of the IEEE. \ L. Christophe, L. Georges, P. Nocera – 2012. - 85(9): 112-115.

50. Макконнелл С. Совершенный код. Практическое руководство по разработке программного обеспечения. / С. Макконнелл – Питер, Русская редакция, 2014. – 895 с. - ISBN: 978-5-7502-0064-1.

51. Канер Сэм и др. Тестирование программного обеспечения. Фундаментальные концепции менеджмента бизнес-приложений: Пер. с англ./ Сэм Канер, Джек Фолк, ЕнгКекНгуен. – К.: Издательство «ДиаСофт», 2001. – 544 с. – ISBN 966-7393-87-9.

52. Кабанова А.Б. Исследование интернета вещей и его применение в создании "Умного дома" / Кабанова А.Б., Бодрова А.А., Логвин В.И. // Символ

науки: международный научный журнал/ "Омега Сайнс", о-во с ограниченной ответственностью. - Уфа : [б. и.], 2016. - № 11-3 (23). - С. 73-75.

53. Резник С. ScrumTeamFoundation Server 2010. Профессиональный подход. /Резник С., Бьерк А., де ла Маза М.; пер. с англ. – М.: ЭКОМ Паблишерз, 2012. – 416 с.: ил. – ISBN: 978-5-9790-0153-1; ISBN: 978-0-470-94333-5.

54. Куперштейн В. «Microsoft Project 2013 в управлении проектами». - БХВ-Петербург, 2013. / В.Куперштейн – 432 с. - ISBN: 978-5-9775-0941-1.

55. Козловський В. О. Методичні вказівки до виконання студентами-магістрантами економічної частини магістерських кваліфікаційних робіт./ В.О.Козловський – Вінниця: ВНТУ, 2012.

ДОДАТКИ

Додаток А (обов'язковий).

Технічне завдання

ЗАТВЕРДЖЕНО

Зав. кафедри АІТ

_____ Квстний Р.Н.

" ____ " _____ 2019 р.

ТЕХНІЧНЕ ЗАВДАННЯ

на магістерську кваліфікаційну роботу

«Розробка системи «Розумний будинок» на базі Raspberry Pi 3 із використанням
модулів Arduino»

08-02.МКР.000.00.010ТЗ

Керівник роботи:

_____ к.т.н., доц. кафедри АІТ Богач І. В.

" ____ " _____ 2019 р.

Виконав:

_____ студент гр.1АКІТ-18м Орлюк Є. А.

" ____ " _____ 2019 р.

Вінниця 2019

1. Назва та галузь застосування.

Система для автоматизації деяких процесів у житловому приватному будинку. Розроблена система дозволяє керувати електроприладами у домі за допомогою смартфона.

2. Підстава для розробки.

Підставою для розробки магістерської кваліфікаційної роботи є наказ ВНТУ №254 від 02.10.2019

3. Мета та призначення розробки.

Мета виконання магістерської кваліфікаційної роботи – є спрощення процесів передачі та обробки даних системи «розумний» будинок, що дозволить суттєво зменшити складність та вартість системи та суттєво підвищить швидкодію роботи системи.

Призначення роботи – розробка системи «Розумний будинок».

4. Джерела розробки:

1. Орлюк Є. А. Розробка системи «Розумний будинок» на бази "Arduino" [Електронний ресурс] / Євгеній Анатолійович Орлюк Вінниця: ВНТУ, 23 березня, 2018. – Режим доступу до ресурсу: <https://conferences.vntu.edu.ua/index.php/all-fksa/all-fksa2018/paper/view/4541/4612> - Назва з екрану.
2. Dickey N., Banks D. and Sukittanon S. Home automation using Cloud Network and mobile devices, Southeastcon, 2012 Proceedings of IEEE, Orlando, FL, 2012, pp. 1-4.
3. Элсенпитер Т.Р., Велт Дж. Умный Дом строим сами / Т. Р. Элсенпитер, Дж Велт/ КУДИЦ-ОБРАЗ. 2015. – 384с.

4. Тесля Е.А. «Умный дом» своими руками. Строим интеллектуальную цифровую систему в своей квартире / Е.А. Тесля – Санкт Петербург, 2008. – 224 с.
5. Гололобов В.Н. «Умный дом» своими руками. / В.Н. Гололобов – М.: ИТ Пресс, 2007. – 416 с.
6. Felix C. and Raglend I. Jacob, "Home automation using GSM," Signal Processing, Communication, Computing and Networking Technologies (ICSCCN), 2011 International Conference on, Thuckafay, 2011, pp. 15-19.

5. Технічні вимоги:

- кількість користувачів, що можуть одночасно керувати системою – до 10;
- частота запитів, що можуть бути коректно оброблені системою – 1 запит за секунду;
- час обробки одного запиту – від 0.8 секунди до 1.2 секунди;
- мова програмування – Python;
- бібліотеки – JSON, Requests;

6. Економічні показники:

- прогнозовані витрати на розробку – 40 тис. грн;
- абсолютна ефективність розробки – не менше 250 тис. грн за рік;
- внутрішня норма дохідності інвестицій – не менше 50%;
- термін окупності витрат для виробника – до трьох років;

7. Стадії та етапи розробки:

Назва етапів дипломного проекту (роботи)	Строк виконання етапів проекту (роботи)
Аналіз стану проблеми і постановка задач	
Аналіз процесу розробки систем «Розумний будинок»	
Розробка структури системи	
Розробка алгоритму захисту системи	
Розробка підсистеми управління	
Обґрунтування вибору інструменту для розробки системи	
Програмна реалізація модулів системи	
Тестування роботи мобільного-додатку	
Оформлення матеріалів до захисту	

8. Порядок контролю та прийняття.

1. Рубіжний контроль. Провести до 10.11.2019.
2. Попередній захист магістерської кваліфікаційної роботи. Провести до 17.12.2019.
3. Захист магістерської кваліфікаційної роботи. Провести в період з 18.12.2019 до 19.12.2019.

Додаток Б (обов'язковий).

Графічна частина

Завідувачкаф. АПТ

д.т.н., професор

(підпис)

Кветний Р.Н

Науковий керівник

к.т.н., доцент

(підпис)

Богач І. В.

Тех. Контроль

к.т.н., доцент

(підпис)

Богач І. В.

Нормконтроль

к.т.н., доцент

(підпис)

Богач І. В.

Рецензент

(підпис)

(науковий ступінь, вчене звання, ініціали та прізвище)

Студент гр. 1АКІТ-18м

Орлюк Є.А.

(підпис)

(ініціали та прізвище)

Продовження додатка Б

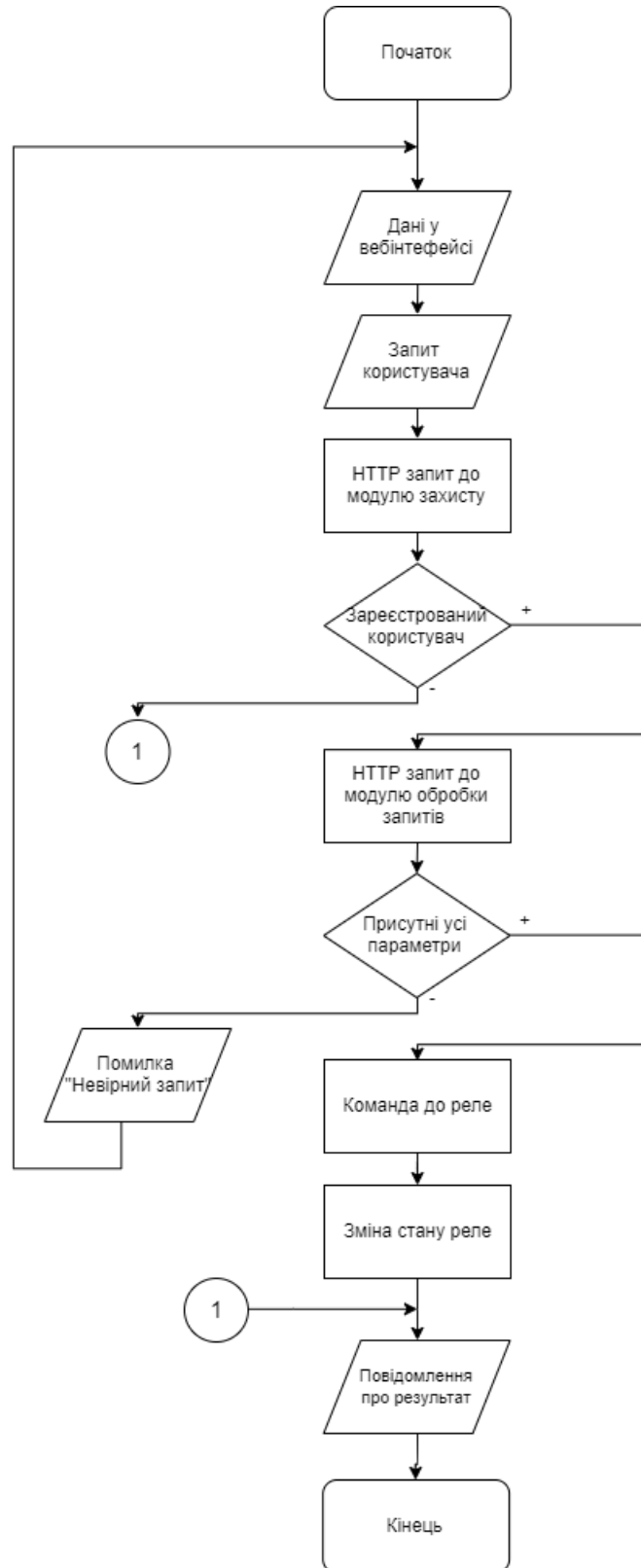


Рисунок Б.1 – Схема роботи системи

Продовження додатка Б

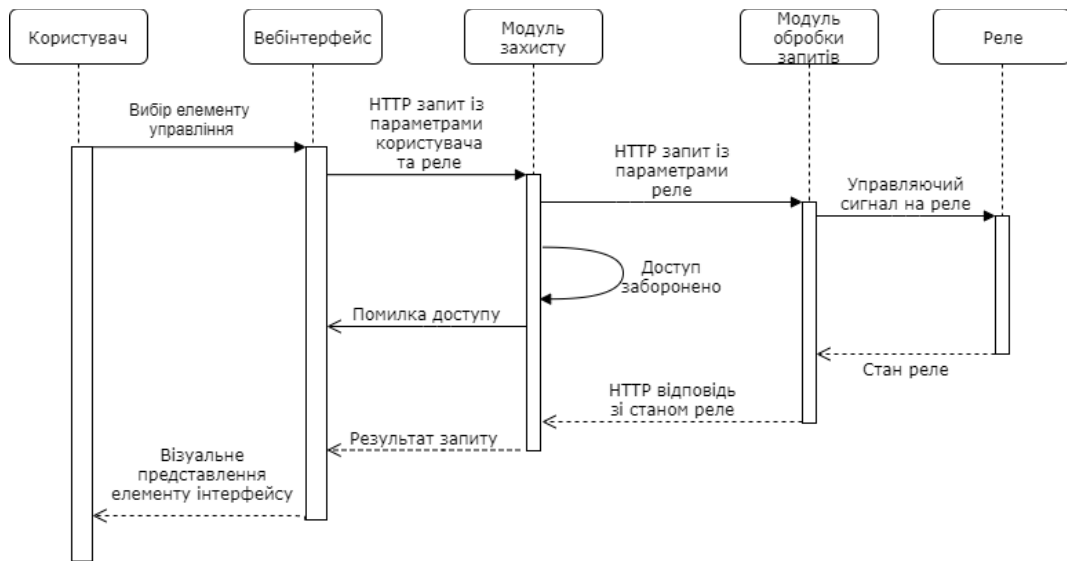


Рисунок Б.2 – UMLдіаграма послідовності

Продовження додатка Б

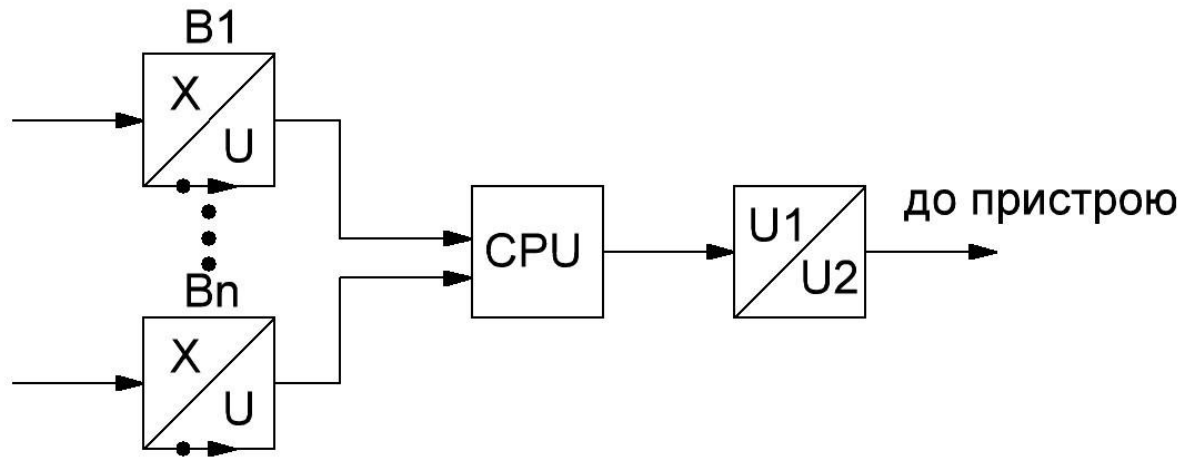


Рисунок Б.3 – Структурна схема системи

Продовження додатка Б

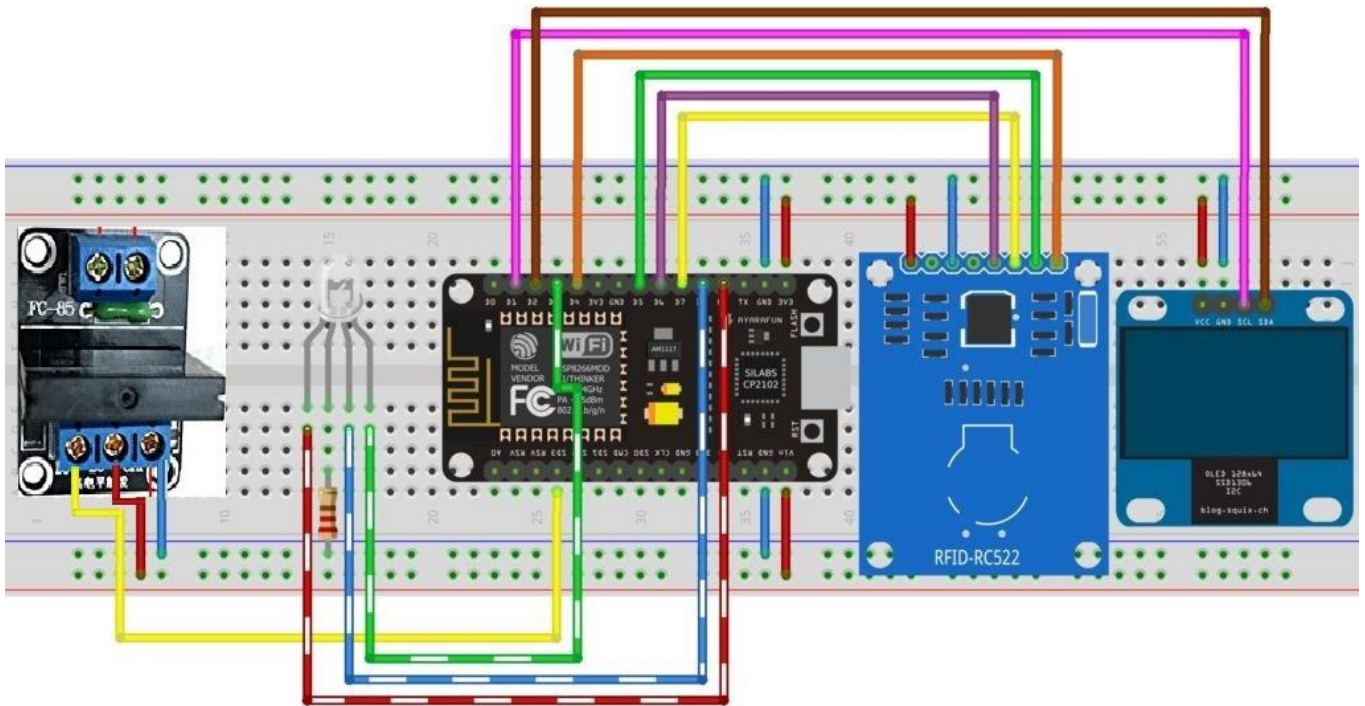


Рисунок Б.4 – Схема підключення елементів електронного замка

Продовження додатка Б

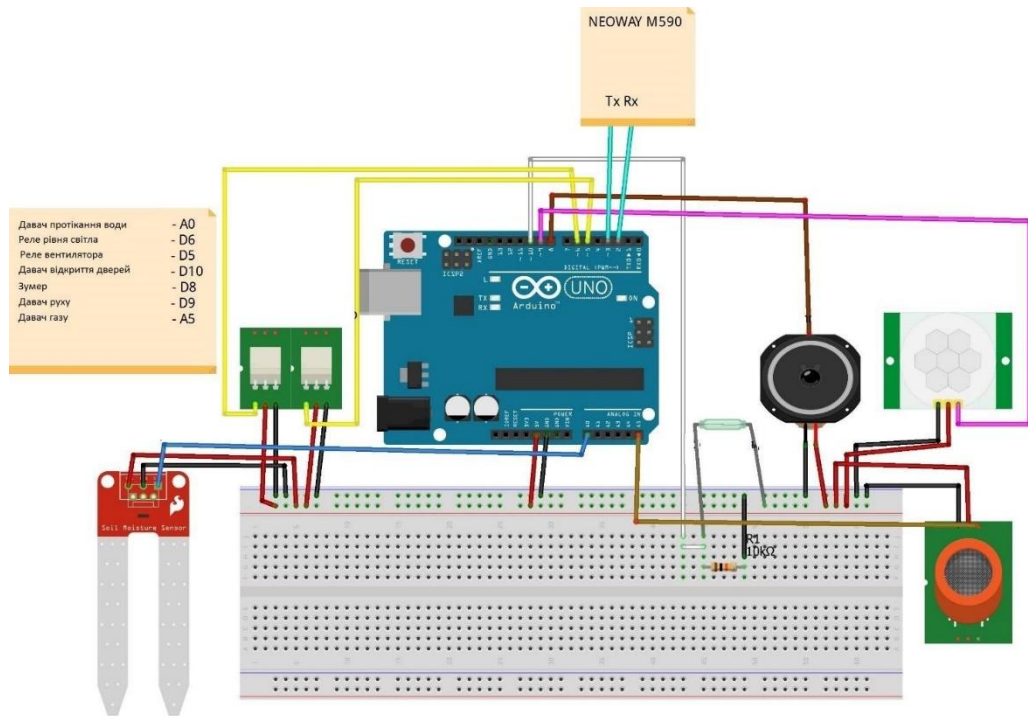


Рисунок Б.5 –Схема підключення елементів сигналізації

Продовження додатка Б

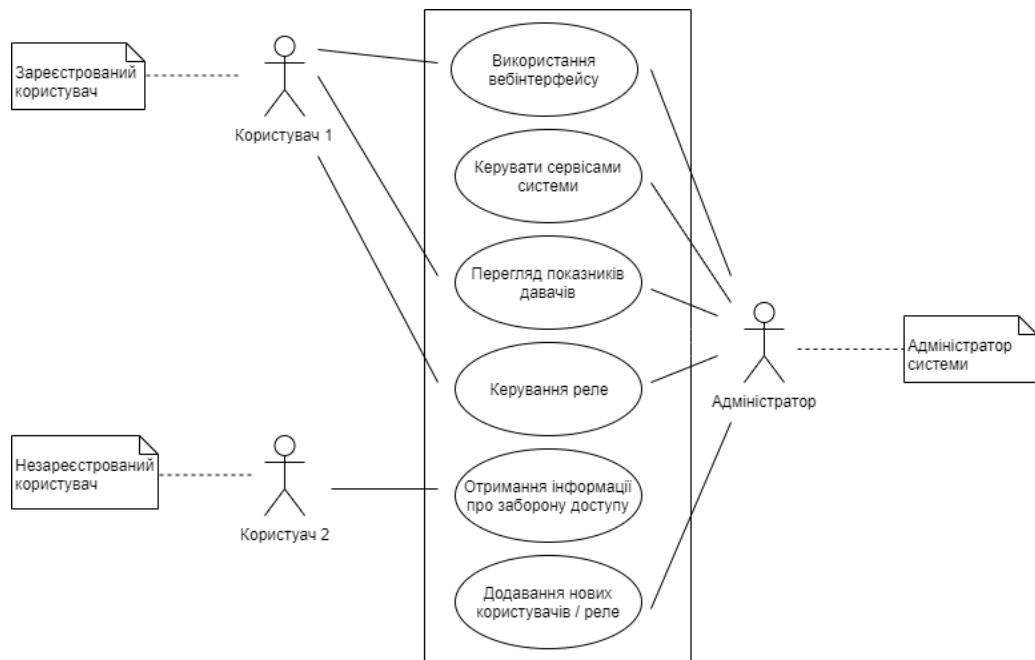


Рисунок Б.6 –UML діаграма прецедентів

Продовження додатка Б

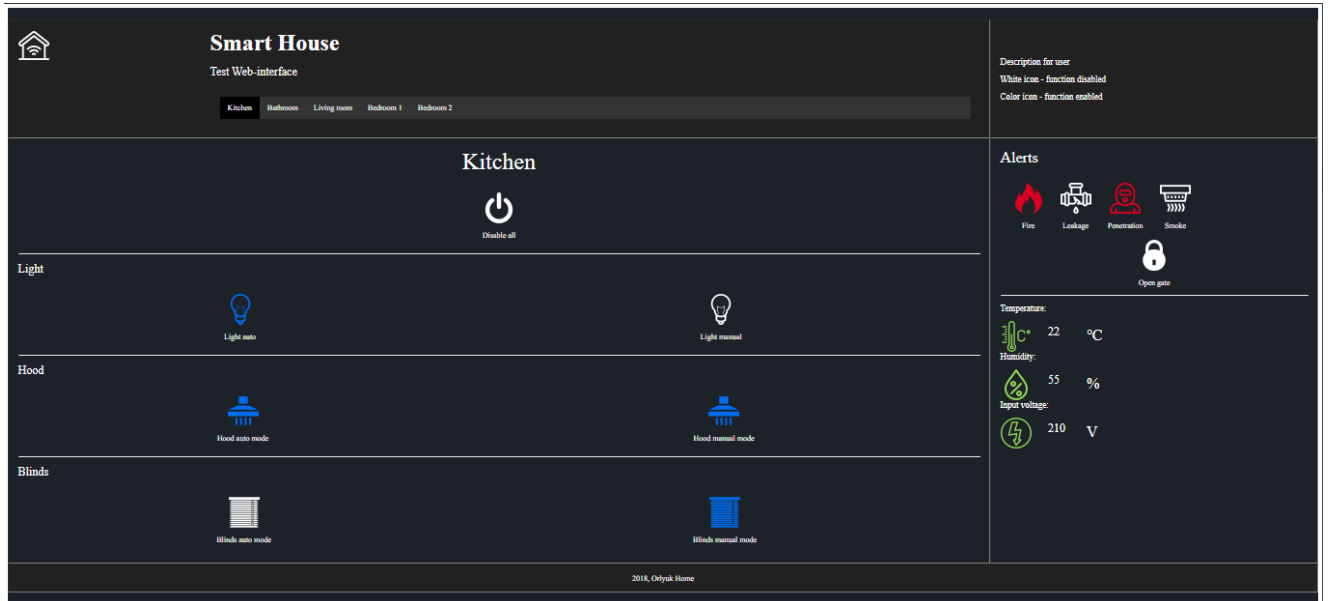


Рисунок Б.7 –Скріншот вигляду вебінтерфейсу

Додаток В (обов'язковий).
Лістинг конфігурації вебсерверу

```
server{
listen80;

server_nameshouse.local;

rewrite^https://shouse.local$request_uri? permanent;
}

server{
listen443ssl;

ssl_certificate/opt/shouse/system/ssl/shouse_local/ssl-cert.pem;
ssl_certificate_key/opt/shouse/system/ssl/shouse_local/ssl-key.pem;
    ssl_stapling on;

server_nameshouse.local;

root/var/www/shouse.local;

location/{
try_files$uri/index.html?$args;
}

location~ \.php$ {
try_files$uri=404;
fastcgi_split_path_info^(.+\.php)(/.+)$;
fastcgi_pass unix:/run/php/php7.0-fpm.sock;
fastcgi_indexindex.php;
include fastcgi_params;
}
}
```

Додаток Г (обов'язковий).

Лістинг вебінтерфейсу

```
<html><head>
<META content=text/html; charset=utf-8 http-equiv=Content-Type>
<style>
    ul {
        list-style-type: none;
        margin: 0;
        padding: 0;
        overflow: hidden;
        background-color: #333;
    }
    li {
        float: left;
    }
    li a {
        display: inline-block;
        color: white;
        text-align: center;
        padding: 14px 16px;
        text-decoration: none;
    }
    li a:hover {
        background-color: #111;
    }
    .active {
        background-color: rgb(0, 0, 0);
    }
</style>
</head><body>
```

```

<table width="100%" height="100%" cellspacing="0" cellpadding="0"
border="0" bgcolor="1d2228">
<tr>
<td>
<table width="100%" align="center" border="1" cellspacing="0"
cellpadding="20">
<tr>
<td bgcolor="212121" width="100%" valign="top" colspan="2">
<table width="100%" bgcolor="212121" cellspacing="0" cellpadding="0">
<tr>
<td bgcolor="212121" width="12%" valign="top">

</td>
<td bgcolor="212121" width="48%" valign="top">
<h3 align="left"><font color="FFFFFF" size="20" face="Open Sans">Smart
House on Arduino</font></h3>
<p><font color="FFFFFF" size="5" face="Open Sans">Test Web-
interface</font></p>
<table width="100%" cellspacing="0" cellpadding="20">
<tr>
<td bgcolor="212121" width="24%" valign="top">
<ul>
<li><a href="#home" class="active">Kitchen</a></li>
<li><a href="#news">Bathroom</a></li>
<li><a href="#contact">Living room</a></li>
<li><a href="#about">Bedroom 1</a></li>
<li><a href="#about">Bedroom 2</a></li>
</ul>
</td>
</tr>
</tr>
</table>
</td>

```

```

</td>
</table>
<td bgcolor="212121" width="25%" valign="middle">
<p><font color="FFFFFF" size="4" face="Open Sans">Description for user
</font></p>
<p><font color="FFFFFF" size="4" face="Open Sans">White icon - function
disabled</font></p>
<p><font color="FFFFFF" size="4" face="Open Sans">Color icon - function
enabled</font></p>
</td>
</td>
</tr>
<tr>
<td width="75%" valign="top" colspan="2">
<p align="center"><font color="FFFFFF" size="15" face="Open
Sans">Kitchen</font></p>
<table width="100%" cellspacing="0" cellpadding="20">
<tr>
<td align="center">
<p><a href=""></a></p>
<p><font color="FFFFFF" size="3" face="Open Sans">Disable all</font></p>
</td>
</tr>
</tr>
</table>
<hr>
<p><font color="FFFFFF" size="5" face="Open Sans">Light</font></p>
<table width="100%" cellspacing="0" cellpadding="20">
<tr>
<td align="center">
<p><a href=""></a></p>
<p><font color="FFFFFF" size="3" face="Open Sans">Light auto</font></p>
</td>

```

```

<td align="center">
<p><a href=""></a></p>
<p><font color="FFFFFF" size="3" face="Open Sans">Light manual</font></p>
</td>
</tr>
</table>
<hr>
<p><font color="FFFFFF" size="5" face="Open Sans">Hood</font></p>
<table width="100%" cellpadding="20">
<tr>
<td align="center">
<p><a href=""></a></p>
<p><font color="FFFFFF" size="3" face="Open Sans">Hood auto
mode</font></p>
</td>
<td align="center">
<p><a href=""></a></p>
<p><font color="FFFFFF" size="3" face="Open Sans">Hood manual
mode</font></p>
</td>
</tr>
</table>
<hr>
<p><font color="FFFFFF" size="5" face="Open Sans">Blinds</font></p>
<table width="100%" cellpadding="20">
<tr>
<td align="center">
<p><a href=""></a></p>
<p><font color="FFFFFF" size="3" face="Open Sans">Blinds auto
mode</font></p>
</td>
<td align="center">

```



```
<p align="center"><font color="FFFFFF" face="Open Sans">2018,  
</font><font color="FFFFFF" face="Open Sans">Orlyuk Home</font></p>  
</td>  
</tr>  
</table>  
</td>  
</tr>  
</table>  
</body>  
</html>
```

Додаток Г (обов'язковий).

Лістинг модулю обробки запитів для перевірки реле

```
#!/usr/bin/env python
"""request handler service"""
import json
import time
import requests
import logging

logging.basicConfig(filename="/var/log/shouse/reqhandlersrv.log",
                    level=logging.DEBUG)

url_switch_on = 'https://%s/control?cmd=event,T1' % relay_ip
url_switch_off = 'https://%s/control?cmd=event,T1' % relay_ip
sonoff_url = 'NOT_INIT'
loop_time = 5

def main():
    loop_counter = 0
    while (True):
        try:
            if loop_counter%2 == 0:
                sonoff_url = url_switch_on
            logging.debug(debug_message)
        else:
            sonoff_url = url_switch_off
            logging.debug(debug_message)

        r = requests.post(sonoff_url)
```

```
        if r.status_code == 200:
logging.debug("Sonoff return code: SUCCESS\n")
        else:
logging.debug("Sonoff return code: FAILED\n")

        except Exception:
logging.debug("UNABLE TO SEND COMMAND TO SONOFF\n")

        loop_counter =loop_counter + 1
        time.sleep(loop_time)

if __name__ == "__main__":
    main()
```