

КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ  
БУДІВНИЦТВА І АРХІТЕКТУРИ

Автоматизації і інформаційних технологій

(факультет)

Інформаційних технологій

(кафедра)

ПОЯСНЮВАЛЬНА ЗАПИСКА  
ДО АТЕСТАЦІЙНОЇ ВИПУСКНОЇ РОБОТИ  
НА ЗДОБУТТЯ ОСВІТНЬОГО РІВНЯ «БАКАЛАВР»

на тему: «Розробка інформаційного забезпечення системи та баз даних  
"розумного" дому»

Дем'янов Нікіта Вікторович

(прізвище, ім'я та по батькові студента повністю)

Київ 2023 р.

**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ  
БУДІВНИЦТВА І АРХІТЕКТУРИ**

**Автоматизації і інформаційних технологій**

(факультет)

**Інформаційних технологій**

(кафедра)

**ЗАТВЕРДЖУЮ**

Завідувач кафедри ІТ

д.т.н., професор Цюцюра С.В.

„\_\_\_” \_\_\_\_\_ 2023 року

**ПОЯСНЮВАЛЬНА ЗАПИСКА**

**ДО АТЕСТАЦІЙНОЇ ВИПУСКНОЇ РОБОТИ**

**НА ЗДОБУТТЯ ОСВІТНЬОГО РІВНЯ «БАКАЛАВР»**

на тему: «Розробка інформаційного забезпечення системи та баз даних  
"розумного" дому»

Виконав: студент 4-го курсу, групи КН-42с

Спеціальності: 122 «Комп'ютерні науки»

Спеціалізація: «Інформаційні управляючі  
системи і технології»

(шифр і назва напрямку підготовки, спеціальності)

Дем'янов Н.В.

(прізвище та ініціали)

Керівник к.т.н., доц. Горда О.В.

(прізвище та ініціали)

Рецензент к.т.н., доц. Єрукаєв А.В.

(прізвище та ініціали)

Київ 2023 р

**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ**

**БУДІВНИЦТВА І АРХІТЕКТУРИ**

Факультет: автоматизації і інформаційних технологій

Кафедра: інформаційних технологій

Освітній рівень: «бакалавр» за ОП

Спеціальність: 122 «Комп'ютерні науки»

Спеціалізація: Інформаційні управляючі системи і технології.

**ЗАТВЕРДЖУЮ**

Завідувач кафедри ІТ  
д.т.н., професор Цюцюра С.В.

„\_\_\_” \_\_\_\_\_ 2023 року

**ЗАВДАННЯ**

**ДО ВИКОНАННЯ АТЕСТАЦІЙНОЇ ВИПУСКНОЇ РОБОТИ  
НА ЗДОБУТТЯ ОСВІТНЬОГО РІВНЯ «БАКАЛАВР»**

Дем'янов Нікіта Вікторович

Тема роботи: Розробка інформаційного забезпечення системи та баз даних "розумного" дому

затверджена наказом ректора КНУБА № 1811/2 від «17» листопада 2022 р.

2. Керівник роботи: к.т.н, доцент Горда Олена Володимирівна, кафедри інформаційних технологій проектування і прикладної математики

3. Строк подання студентом роботи до захисту: \_\_\_\_\_.

4. Зміст пояснювальної записки за розділами:

Р.1. Аналіз та дослідження проблеми

Р.2. Проектування інформаційного забезпечення

Р.3. Практична реалізація

Р.4. Бізнес план

5. Інформаційні слайди:

С.1. Розробка ІЗ та баз даних «розумного» будинку \_\_\_\_\_.

- C.2. Обґрунтування актуальності даної теми .
- C.3. Об'єкт, предмет та мета дослідження .
- C.4. Дерево основних цілей .
- C.5. Рішення та вибір системи .
- C.6. Дерево функцій .
- C.7. Загальні можливості структури інтерфейсу .
- C.8. Розширена концептуальна схема .
- C.9. Фізична модель БД .
- C.10. Файлова структура об'єкту .
- C.11. Схема зв'язків БД .
- C.12. Приклад роботи БД .
- C.13. Схема виробничих потоків .
- C.14. Висновок .

## 6. Календарний план виконання атестаційної випускної роботи

Види робіт та їх зміст	Дата виконання
Р. 1. Аналіз та дослідження проблеми	15.04.2023-23.04.2023
Р. 2. Проектування інформаційного забезпечення	24.04.2023-03.05.2023
Р. 3. Практична реалізація	05.05.2023-15.05.2023
Р. 4. Бізнес план	08.04.2023-14.04.2023
Остаточне оформлення роботи	23.05.2023-24.05.2023
Направлення роботи на рецензування	24.05.2023-28.05.2023
Попередній захист роботи на кафедрі	13.06.2023-14.06.2023

## 7. Консультанти розділів атестаційної випускної роботи

Розділ	Прізвище, ініціали та посада консультанта, представника комісії	дата	підпис
Ергономіка інформаційних технологій	д.т.н. проф. Терентьев О.О.		
Прийом програмного продукту	к.т.н. доц., Срукаєв А.В.		

8. Дата видачі завдання: \_\_\_\_\_ р.

Керівник

Бакалавр

Горда О.В.

(підпис)

(прізвище та ініціали)

Дем'янов Н.В.

(підпис)

(прізвище та ініціали)

## АНОТАЦІЯ

**Дем'янов Н.В. Розробка інформаційного забезпечення системи та баз даних "розумного" дому.**

Дипломна бакалаврська робота за спеціальністю – «Комп'ютерні науки» – Київський національний університет будівництва та архітектури, Київ, 2023 рік.

Бакалаврську роботу присвячено дослідженню основних тенденцій, принципів та реалізацій систем «розумних» будинків, їх інформаційного забезпечення та баз даних в цілому.

У роботі досліджено основні принципи та технології розробки інформаційного забезпечення для "розумного" будинку, розроблено базу даних для зберігання та обробки інформації, що отримується від різних датчиків.

*Ключовими словами є: "інформаційне забезпечення", "розумний будинок", "база даних", "датчики", "функціональні можливості".*

## ABSTRACT

**Demianov N.V. Development of information support system and databases for a "smart" home.**

Bachelor's thesis in Computer Science – Kyiv National University of Construction and Architecture, Kyiv, 2023.

The bachelor's thesis is devoted to the research of the main trends, principles, and implementations of "smart" home systems, their information support, and databases in general.

*Key words: "information support", "smart home", "database", "sensors", "functional capabilities".*

## **ЗМІСТ**

1 АНАЛІЗ ТА ДОСЛІДЖЕННЯ ПРОБЛЕМ.....	13
1.1 Загальні та теоретичні відомості, концепції та проблематика «розумного» дому .....	13
1.2 Постановка та аналіз проблеми.....	15
1.3 Дерево основних цілей .....	17
1.4 Аналіз існуючих рішень.....	19
1.4.1 Рішення розумних будинків. ....	19
1.4.2 Рішення баз даних.....	23
1.5 Постановка задачі. ....	27
1.6 Проблеми взаємозв'язку баз даних з реалізаціями систем розумного будинку.....	28
2. Проектування інформаційного забезпечення.....	31
2.1. Ключові питання постановки завдання проектування інформаційного забезпечення.....	31
2.2. Дерево функцій .....	32
2.3. Вибір технології зв'язку .....	34
2.4. Вибір технології забезпечення безпеки.....	36
2.5. Вибір інтерфейсу користувача .....	37
2.5.1. Вимоги до інтерфейсу .....	37
2.5.2. Загальні можливості структури інтерфейсу.....	38
2.5.3. Вибір типу інтерфейсу користувача .....	41
2.6. Проектування моделі бази даних .....	43
2.6.1. Створення концептуальної моделі хабу «розумного» будинку	
44	
2.6.2. Створення логічної моделі хабу «розумного» будинку .....	46
2.6.3. Створення фізичної моделі хабу «розумного» будинку.....	48

2.7. Загальний вигляд сутностей бази даних.....	49
3. ПРАКТИЧНА РЕАЛІЗАЦІЯ.....	53
3.1. Вибір програмного інструментарію.....	53
3.1.1. Бази даних.....	53
3.1.2. Мова програмування.....	54
3.1.3. Програмне середовище.....	57
3.2. Архітектура розроблюваного об'єкта.....	59
3.3. Файлова структура об'єкту.....	60
3.4. Розробка бази даних хабу та підключення до неї.....	62
3.5. Методи роботи з базою даних хабу.....	65
3.6. Створення та програмування інформаційного забезпечення датчиків «розумного» дому.....	71
3.6.1. Датчик відкриття та закриття дверей, вікон, шафи, тощо.....	71
3.6.2. Датчик розумного термостату.....	72
3.7. Програмування інформаційного забезпечення та баз даних інших пристроїв та датчиків розумного будинку.....	75
4. БІЗНЕС ПЛАН.....	77
4.1. Впровадження продукту.....	77
4.1.1. Визначні риси по відношенню до продукту конкурентів.....	78
4.1.2. Необхідність продукту споживачу.....	79
4.1.3. Очікувані основні фінансові результати.....	80
4.2. Проектований продукт, або вид послуг.....	82
4.2.1. Потреби продукту по відношенню до проєкту.....	82
4.2.2. Особливості і відмінні риси розроблюваного проєкту, від інших реалізацій.....	83
4.2.3. Патенти, авторські свідоцтва та наочне зображення розроблюваного проєкту.....	84

4.2.4. Попередня оцінка реалізації ціни реалізації продукту .....	84
4.2.5. Очікувана величина прибутку .....	85
4.2.6. Характеристика якісних показників продукту проекту.....	85
4.2.7. Організація сервісу продукту проекту .....	85
4.3. Конкуренція .....	85
4.3.1. Найбільші виробники аналогічного продукту.....	85
4.3.2. Фінансово–аналітичний аналіз роботи компаній .....	86
4.3.3. Сутність предмету конкуренту.....	87
4.3.4. Конкурентний рівень цін .....	88
4.4. План виробництва .....	88
4.4.1. Місце виробництва товарів на діючому підприємстві. ....	89
4.4.2. Необхідність виробничих потужностей .....	89
4.4.3. Умови закупки сировини. Репутація та досвід постачальника. 90	
4.4.4. Виробнича кооперація.....	91
4.4.5. Лімітація об'єктів підприємства .....	91
4.4.6. Проблема устаткування.....	92
4.4.7. Схема виробничих потоків. ....	93
4.4.8. Оцінка можливих втрат.....	95
4.5. Організаційний план.....	96
4.5.1. Кваліфікація фахівців .....	96
4.5.2. Умови прийняття на роботу.....	98
4.5.3. Установи для пошуку робітників.....	98
4.5.4. Біографічні дані найнятих робітників .....	98
4.5.5. Організаційна структура підприємства .....	99
4.6. Стратегія фінансування.....	100

4.6.1. Необхідна кількість засобів для реалізації проєкту .....	100
4.6.2. Джерела фінансових ресурсів.....	100
ВИСНОВОК .....	102
СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ .....	105

## ВСТУП

**Актуальність дослідження.** Тема "Розробка інформаційного забезпечення системи та баз даних "розумного" дому" (РД) є дуже актуальною в сучасному світі, де технології стають все більш інтегрованими в повсякденне життя людей. Розумний дім – це концепція, що полягає у використанні різних технологій для автоматизації та контролю різних функцій в будинку, таких як освітлення, опалення, вентиляція, безпека, розваги та багато інших.

Розробка даного інформаційного поля може вирішити багато проблем, пов'язаних з управлінням та контролем систем в будинку. Наприклад, така система може забезпечити автоматичний контроль різних параметрів у місці проживання, таких як температура, вологість, освітлення та інші, і забезпечити ефективне використання енергії.

Крім того, інформаційне забезпечення системи та баз даних можуть забезпечити безпеку будинку, наприклад, забезпечити відеоспостереження та систему виявлення вторгнення, що знижує ризики злому та крадіжки в будинку. Таке впровадження також може забезпечити зручний та простий спосіб контролювання систем в будинку через мобільний додаток або інший пристрій.

Отже, розробка інформаційного забезпечення системи та баз даних для "розумного" дому має великий потенціал для покращення якості життя людей та забезпечення ефективного використання енергії. Така тема дипломного проєкту є дуже актуальною та має потенціал для розвитку в майбутньому.

**Мета дослідження.** В основі лежить розробка програмного забезпечення для керування "розумним" будинком, яке забезпечує автоматизацію процесів управління побутовими пристроями, системами будинку та успішній імплементації розробленої системи та її впровадженні у реальне життя.

**Об'єкт дослідження.** Система розумний будинок.

**Предмет дослідження.** Програмне забезпечення та бази даних, які забезпечують автоматизацію процесів управління різноманітними системами IoT.

**Методика та завдання дослідження.** Методика дослідження дипломного проєкту буде базуватись на наступних кроках:

1. Аналіз літературних джерел та наукових публікацій, що стосуються розробки систем "розумного" будинку та вивчення існуючих технологій та рішень.
2. Визначення вимог до функціональності досліджених системи керування та розробка плану реалізації цих вимог.
3. Розробка архітектури програмного забезпечення та баз даних системи IoT з урахуванням встановлених вимог.
4. Реалізація програмного забезпечення та баз даних системи керування "розумним" будинком та їх тестування.
5. Впровадження розробленої системи в реальний житловий простір та вивчення її функціональності в реальних умовах.
6. Зібрання та аналіз результатів використання системи керування "розумними" пристроями та визначення можливих напрямків її подальшого вдосконалення.

Методика забезпечує ефективну розробку та реалізацію інформаційної системи керування, можливість вивчення її функціональності в реальних умовах з метою подальшого вдосконалення.

**Практична значимість.** Результати отримані під час дослідження можуть бути використані як в учбовому так і в реальному просторі для проєктування певних публічних місць та приватних житлових будинків.

**Результати дослідження.** Розробка та реалізація інформаційного забезпечення системи "розумних" речей, що дозволяє автоматизувати роботу різних пристроїв та систем у будинку, забезпечувати комфорт та безпеку користувача, а також дистанційно керувати будинком за

допомогою Інтернету. Результатом є також розробка оптимальної структури баз даних, що забезпечують ефективну обробку, зберігання та взаємодію різних даних в системі.

### **Короткий зміст розділів:**

- У першому розділі частково описана загальна та теоретична інформація щодо відомостей, концепцій та проблематики систем IoT. Описує постановку та аналіз проблеми, дерево можливих цілей, та знайомить з ринком існуючих рішень, як баз даних так і систем розумних речей, в цілому. Постановляє задачу та відповідає на питання проблем взаємозв'язку баз даних з реалізаціями «розумних» систем.
- Другий розділ надає інформацію щодо проектування інформаційного забезпечення, а саме: розбирає ключові питання постановки завдання, описує дерево можливих функцій, надає теоретичні дані щодо вибору технології зв'язку, безпеки, інтерфейсу. Надає матеріал про створення концептуальної, фізичної і логічної форми представлення бази даних, та бд в цілому.
- Інформація описана у третьому розділі спрямована на практичну реалізацію завдання. Надає теоретичні відомості щодо вибору бази даних, мови програмування та середовища програмування, що в свою чергу створює базу програмного інструментарію, для подальшої практичної реалізації бази даних, таких сенсорів як: хаб, датчик закриття/відкриття, розумний термостат, тощо.
- Четвертий розділ описує роботу мовою бізнес плану, створює резюме продукту, описує проєктований продукт та вид послуги, оцінює ринок збуту, конкуренцію, стратегію маркетингу, план виробництва та організаційний план, сутність юридичного плану, оцінку ризиків і страхування, фінансовий план та стратегію фінансування.

# 1 АНАЛІЗ ТА ДОСЛІДЖЕННЯ ПРОБЛЕМ

## 1.1 Загальні та теоретичні відомості, концепції та проблематика «розумного» дому

Зв'язок інформаційних технологій зі змінами в суспільстві та побуті людей необхідно розглядати у контексті тенденцій, що склалися в останні роки. Однією з найбільш важливих є розвиток технології "розумного будинку". Ця технологія дозволяє людям автоматизувати та контролювати побутові процеси за допомогою комп'ютерів та інтернету речей.

Розумний будинок – це інноваційна технологія, яка надає можливість автоматизувати та контролювати різноманітні пристрої та системи в домі. Такий будинок є не тільки більш зручним для проживання, але й більш енергоефективним та безпечним. У зв'язку зі зростаючим попитом на цю технологію, дослідження у цій області стали особливо актуальними.[1]

Однак, використання розумного будинку також викликає певні проблеми та виклики. Однією найбільш суттєвою з них є збільшення кількості залежності людей від технологій та втрата можливості самостійного контролю побутових процесів, хоча для кожної людини, це суб'єктивна річ.

Тому, метою даної дипломної роботи є детальний аналіз інноваційної технології розумного будинку та її інформаційного забезпечення, вивчення основних принципів її функціонування та застосування в різних сферах життя людини. Також робиться спроба оцінити позитивні та негативні аспекти впровадження розумного будинку та запропонувати шляхи вдосконалення даної технології.

Для досягнення поставленої мети, в роботі є намір розглянути основні аспекти розумного будинку, зокрема, архітектурні особливості, системи управління, типи та реалізації баз даних, комунікації та безпеки. Також проаналізовані приклади впровадження розумного будинку в різних країнах світу, а також наукові статті та дослідження в цій галузі.

Для забезпечення ефективної роботи розумного будинку необхідно використовувати сучасні технології, такі як штучний інтелект, машинне навчання та інтернет речей. Тому у дипломній роботі були застосовані саме ці інформаційні ресурси та детальний аналіз технологій, що лягають в основу розумного будинку, та їхніх можливостей.

У контексті впровадження розумного будинку в життя людей, розглянуто соціально–економічні та екологічні аспекти. Дослідження ролі розумного будинку в зниженні споживання енергії та захисту довкілля є важливою складовою даної теми.

На основі існуючого на сьогоднішній день досвіду були запропоновані рекомендації для вдосконалення системи розумного будинку та покращення якості життя людей. Існуючі рекомендації можуть бути корисними для розробників технологій "розумного будинку", інженерів, дослідників та всіх, хто цікавиться розвитком інформаційних технологій та їхнього впливу на суспільство.

У цілому, дана дипломна робота є важливим кроком у розумінні та дослідженні технології "розумного будинку", її інформаційних можливостей та ризиків. Результати можуть бути корисними для всіх зацікавлених сторін, які працюють над вдосконаленням та розвитком цієї галузі. Крім того, дана робота може бути корисна для споживачів, які планують встановити розумний будинок у своєму житті, дозволяючи їм краще розуміти можливості та переваги цієї технології.

Однак, варто зазначити, що технологія "розумного будинку" ще не є повністю зрілою та має свої виклики та обмеження. Серед них можна відзначити високу вартість встановлення та підтримки систем, несумісність між різними системами, можливість хакерських атак та порушення приватності.[2]

У дипломній роботі проаналізовані ці виклики та обмеження, а також, теоретично, запропоновано можливі шляхи їхнього подолання.

Висновки роботи є важливим етапом у розвитку технології "розумного будинку" та її впровадженні у побутове використання. Вони

допоможуть краще зрозуміти потенціал цієї технології та сприятимуть її подальшому вдосконаленню.

Загальна мета дипломної роботи – дослідити технологію "розумного будинку", а саме її інформаційного забезпечення, проаналізувати її особливості, можливості та виклики, а також запропонувати рекомендації для вдосконалення та покращення якості життя людей. Результати дослідження можуть бути корисними для всіх зацікавлених сторін та внести свій внесок у розвиток цієї важливої технології.

## **1.2 Постановка та аналіз проблеми.**

Однією з головних проблем при впровадженні, або навіть теоретичному описі є проблема безпеки. Оскільки "розумний" будинок зв'язаний з Інтернетом та мережею зв'язку, його системи можуть стати об'єктом кібератак та викрадення даних. Зловмисник може мати доступ до фізичних пристроїв, що керують побутовою технікою, зможе змінювати їх налаштування та виконувати інші дії, що можуть призвести до пошкодження майна та навіть загрози життю мешканців. Питання приватності, також знаходиться в основних позиціях, оскільки багато модулів системи smart house здатні збирати та передавати дані, які стосуються приватних даних мешканців, недостатній захист яких, може призвести до витоку особистої інформації та порушення конфіденційності. Тому, як розробникам необхідно включати у свої системи механізми захисту та шифрування даних, так і нам треба розбиратись в питанні захисту, хоча б на мінімальному рівні, щоб забезпечити безпеку особистих даних.[7]

Іншою проблемою є вартість встановлення та обслуговування. Необхідно встановлювати спеціальні пристрої та програмне забезпечення, а також проводити їх регулярне оновлення. Це може призвести до високих витрат на встановлення та підтримку системи, що зробить її недоступною для більшості споживачів.

Також, виникає проблема відсутності єдиного стандарту для "розумного" будинку. Різні виробники використовують різні протоколи з чим може виникати проблема несумісності між різними компонентами системи, що призводить до складнощів у їх інтеграції та обміну даними. Це може призвести до того, що деякі компоненти системи не будуть працювати належним чином або не будуть взаємодіяти між собою, що погіршить функціональність системи в цілому.

Реалізація системи «розумного» будинку потребує встановлення великої кількості пристроїв, яка може погіршити рівень екологічної загрози. З однієї сторони це веде до збільшення споживаної електроенергії, що веде до значних витрат, а з іншої – замовленням простору джерелом високої частоти, таким як електромагнітне поле.

Нарешті, існує проблема недостатньої надійності "розумних" будинків. В разі збою в системі, її мешканці можуть бути позбавлені контролю над побутовою технікою, або навіть стати свідками аварійних ситуацій. Для запобігання таким ситуаціям, необхідно розробляти системи, які працюють належним чином навіть у разі виникнення непередбачуваних ситуацій.

З іншого боку, "розумний" будинок може стати джерелом нових можливостей для управління енергозбереженням та зменшенням викидів CO<sub>2</sub>. Наприклад, система управління освітленням та опаленням може забезпечувати оптимальний рівень комфорту для мешканців та зменшувати витрати на електроенергію. Крім того, "розумний" будинок може інтегруватися зі смарт-мережами та розумними міськими інфраструктурами, що дозволить забезпечити ефективне управління ресурсами та зменшення негативного впливу на довкілля.

Отже, варто зазначити, що розробка інформаційного забезпечення системи та баз даних "розумного" дому є складним та багатограним завданням, яке вимагає врахування різних аспектів та проблем. Проте, правильний аналіз та розв'язання цих проблем дозволить створити

комфортний, економічний та безпечний "розумний" будинок, який відповідатиме сучасним вимогам та потребам мешканців.

### 1.3 Дерево основних цілей

Метою досліджень в роботі є розробка інформаційного забезпечення системи та баз даних "розумного" дому.

Дерево цілей опису даної мети наведено на рисунку 1.1.

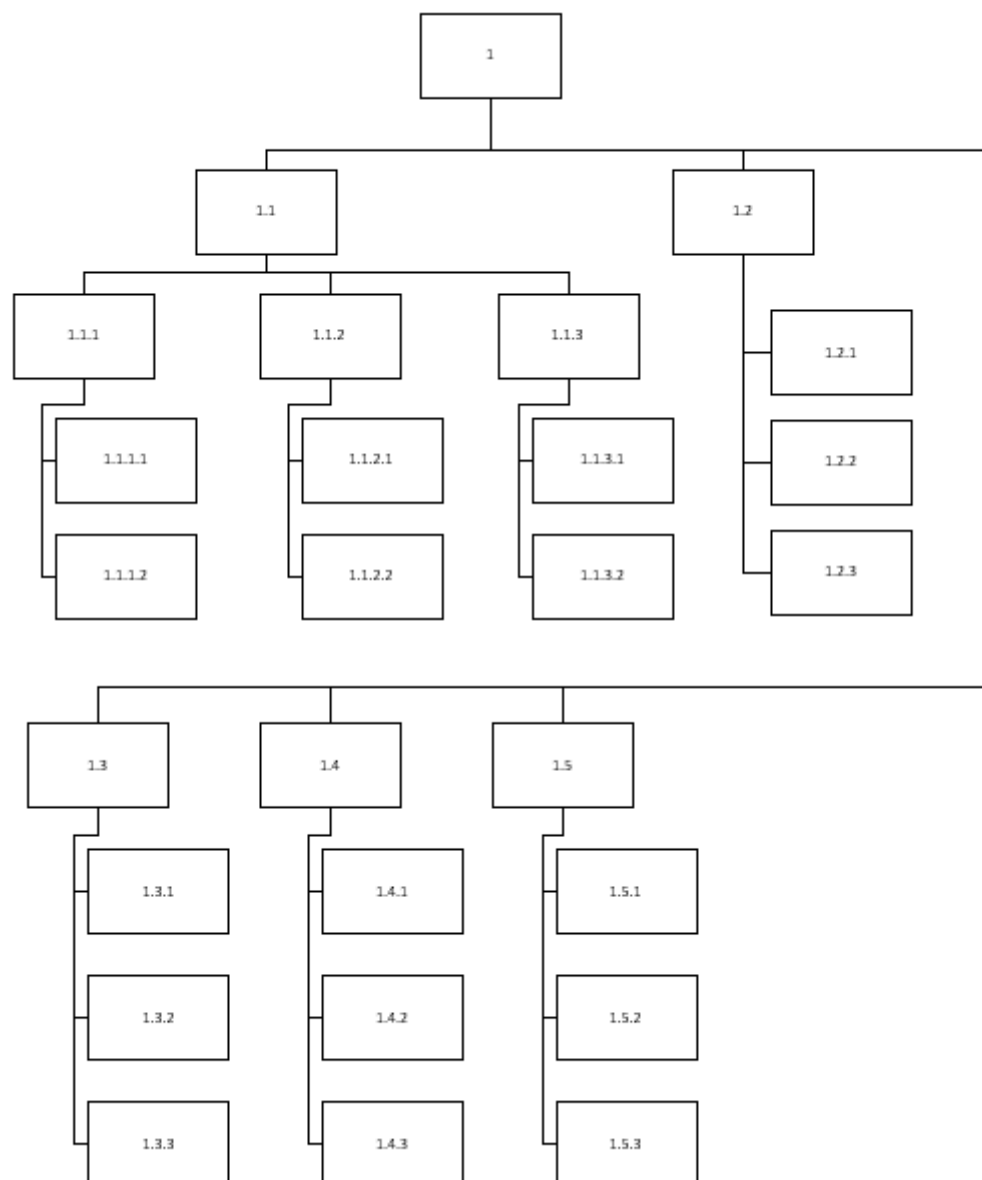


Рисунок 1.1 – Дерево цілей досліджень роботи.

1.1. Розробка інформаційного забезпечення системи та баз даних для "розумного" дому.

1.1.1. Розробка програмного забезпечення для контролю та управління електричними приладами та пристроями "розумного" дому.

1.1.1.1. Розробка модулів для контролю за електропотужністю та енергоспоживанням.

1.1.1.2. Розробка інтерфейсу для управління електричними пристроями та приладами.

1.1.2. Розробка програмного забезпечення для контролю та управління системами безпеки "розумного" дому.

1.1.2.1. Розробка модулів для виявлення порушень безпеки та відповідного реагування на них.

1.1.2.2. Розробка інтерфейсу для управління системами безпеки.

1.1.3. Розробка програмного забезпечення для контролю та управління системами комфорту "розумного" дому.

1.1.3.1. Розробка модулів для контролю за температурою, освітленням, вентиляцією та іншими параметрами комфорту.

1.1.3.2. Розробка інтерфейсу для управління системами комфорту.

1.2. Розробка мобільного додатку для керування "розумним" будинком.

1.2.1. Розробка інтерфейсу мобільного додатку.

1.2.2. Розробка функціоналу мобільного додатку для керування "розумним" будинком.

1.2.3. Тестування мобільного додатку та виявлення та виправлення помилок.

1.3. Забезпечення безпеки та захисту даних "розумного" дому.

1.3.1. Розробка системи автентифікації та авторизації користувачів.

1.3.2. Захист даних користувачів та інформації про систему "розумного" дому.

1.3.3. Виявлення та виправлення можливих вразливостей системи безпеки та захисту даних.

1.4. Підвищення ефективності та енергоефективності "розумного" дому.

1.4.1. Розробка алгоритмів оптимального керування системами "розумного" дому для забезпечення енергоефективності.

1.4.2. Використання датчиків та інших пристроїв для моніторингу та аналізу енергоспоживання в "розумному" домі.

1.4.3. Визначення та підвищення ефективності роботи систем "розумного" дому, зокрема, шляхом оптимізації ресурсів.

1.5. Підвищення зручності та доступності "розумного" дому для користувачів з різними потребами.

1.5.1. Розробка інтерфейсу та функціоналу "розумного" дому для людей з обмеженими можливостями.

1.5.2. Врахування потреб користувачів різних вікових та соціальних груп у процесі розробки та вдосконалення "розумного" дому.

1.5.3. Надання можливості користувачам з віддаленого доступу до систем "розумного" дому за допомогою інтернету та мобільних додатків.

## **1.4 Аналіз існуючих рішень.**

### ***1.4.1 Рішення розумних будинків.***

На ринку існує велика кількість різних рішень розумних будинків, які надають користувачам можливість контролювати різні системи, такі як освітлення, опалення, вентиляцію та інші. У даному розділі розглянуті деякі з найбільш поширених та ефективних рішень, що вже існують на ринку, включаючи їх переваги та недоліки. Ви дізнаєтесь, які можливості надають різні системи розумного будинку та як вони можуть допомогти зробити життя більш комфортним та безпечним.[8]

1. Amazon Alexa: ця платформа є однією з найпопулярніших на ринку інформаційного забезпечення для розумних будинків. Вона забезпечує інтеграцію з різноманітними пристроями розумного будинку, такими як

замки, освітлення, термостати та інші, і дозволяє керувати ними за допомогою голосових команд.

2. Google Home: ця платформа є конкурентом Amazon Alexa і пропонує аналогічний набір функцій та інтеграцію з різними пристроями розумного будинку. Вона також забезпечує керування за допомогою голосових команд.

3. Apple HomeKit: ця платформа спеціалізується на інтеграції з продуктами Apple, такими як iPhone, iPad та Apple Watch. Вона забезпечує можливість керувати пристроями розумного будинку за допомогою голосових команд, а також з допомогою мобільних пристроїв.

4. Samsung SmartThings: ця платформа розроблена компанією Samsung і забезпечує інтеграцію з різноманітними пристроями розумного будинку, включаючи замки, освітлення, термостати та інші. Вона дозволяє керувати пристроями за допомогою мобільних пристроїв.

5. Hubitat Elevation: ця платформа пропонує інтеграцію з різноманітними пристроями розумного будинку, включаючи замки, освітлення, термостати та інші. Вона забезпечує можливість керування пристроями розумного будинку за допомогою мобільних пристроїв та веб-інтерфейсу.

6. OpenHAB: ця платформа є відкритою і забезпечує інтеграцію з різноманітними пристроями розумного будинку, включаючи замки, освітлення, термостати та інші. Вона дозволяє налаштувати складні сценарії автоматизації та керувати пристроями за допомогою мобільних пристроїв та веб-інтерфейсу.

7. Home Assistant: ця платформа також є відкритою та забезпечує інтеграцію з різноманітними пристроями розумного будинку, включаючи замки, освітлення, термостати та інші. Вона дозволяє налаштувати складні сценарії автоматизації та керувати пристроями за допомогою мобільних пристроїв та веб-інтерфейсу.

8. Xiaomi Mi Smart Home Hub: це пристрій, який дозволяє керувати різноманітними пристроями розумного будинку від Xiaomi та інших

виробників. Він може підключатися до інтернету через Wi-Fi та дозволяє керувати пристроями за допомогою мобільного додатку Mi Home. Хаб підтримує інтеграцію з різноманітними пристроями, такими як розумні лампи, замки, камери, сенсори та інші. Інформація з порівнянням наведена у таблиці 1.1

Таблиця 1.1 – опис рішень систем розумних пристроїв.

Система	Виробник	Наявні широкі можливості інтеграції?	Підтримувані протоколи	Підтримувані голосові помічники	Наявний стандартний набір можливостей?
Alexa	Amazon	+	Zigbee, Z-Wave, Wi-Fi, Bluetooth	Amazon Alexa, Google Assistant, Apple Siri	+
Home	Google	+	Zigbee, Z-Wave, Wi-Fi, Bluetooth	Amazon Alexa, Google Assistant, Apple Siri	+
HomeKit	Apple	+	Zigbee, Z-Wave, Wi-Fi, Bluetooth	Apple Siri	+
SmartThings	Samsung	+	Zigbee, Z-Wave, Wi-Fi, Bluetooth	Amazon Alexa, Google Assistant	+
Elevation	Hubitat	+	Zigbee, Z-Wave, Wi-Fi, Bluetooth	Amazon Alexa, Google Assistant	+

Продовження таблиці 1.1 – опис рішень систем розумних пристроїв.

Система	Виробник	Наявні широкі можливості інтеграції?	Підтримувані протоколи	Підтримувані голосові помічники	Наявний стандартний набір можливостей?
OpenHAB	OpenHAB Foundation	+	Zigbee, Z-Wave, Wi-Fi, Bluetooth	Amazon Alexa, Google Assistant	+
Home Assistant	Home Assistant Community	+	Zigbee, Z-Wave, Wi-Fi, Bluetooth	Amazon Alexa, Google Assistant	+
Mi Smart	Xiaomi	+	Zigbee, Z-Wave, Wi-Fi, Bluetooth	Amazon Alexa, Google Assistant	+

З порівняння систем розумного будинку, які представлені в таблиці, можна зробити декілька висновків. По-перше, всі системи мають загальні можливості, такі як контроль освітлення, звукові команди та мобільний додаток для управління, який знаходиться в стандартному наборі можливостей. По-друге, різні системи мають різний рівень інтеграції з іншими системами та пристроями, що може бути важливим фактором при виборі. По-третє, деякі системи, такі як Apple HomeKit та Samsung SmartThings, працюють зі спеціальним апаратним забезпеченням, тоді як інші можуть бути підключені до пристроїв через різні протоколи бездротового зв'язку. І нарешті, вартість систем розумного будинку може варіюватися від невеликої вартості до значної, залежно від можливостей та виробника. В кінцевому рахунку, вибір системи розумного будинку

залежить від потреб користувачів та їх бюджету, і перед покупкою варто ознайомитись з характеристиками кожної системи для забезпечення оптимального вибору

### ***1.4.2 Рішення баз даних.***

Бази даних є невід'ємною частиною багатьох програмних продуктів та інформаційних систем. Зараз на ринку існує велика кількість різних рішень баз даних, які надають користувачам можливість зберігати, організувати та оптимізувати свої дані. У даному розділі розглянуті деякі з найбільш поширених та ефективних рішень баз даних, що вже існують на ринку, включаючи їх переваги та недоліки. Ви дізнаєтесь, які можливості надають різні системи баз даних та як вони можуть допомогти зробити роботу з даними більш ефективною та зручною.

1. Oracle Database: Є однією з найбільш потужних та розширюваних систем управління базами даних. Вона забезпечує підтримку для широкого спектру додатків, включаючи додатки з бізнес-аналітики та високошвидкісні транзакційні системи. Oracle Database підтримує масштабування та високу доступність, а також має широкий набір інструментів для розробки та управління базами даних.

2. Microsoft SQL Server: Є популярною реляційною системою управління базами даних, яка надає підтримку для різних типів даних та має різноманітні інструменти для аналізу та візуалізації даних. SQL Server підтримує велику кількість високошвидкісних транзакцій та масштабування, а також забезпечує високу доступність.

3. MySQL: Є однією з найпопулярніших відкритих реляційних систем управління базами даних. Вона підтримує широкий спектр додатків, включаючи веб-додатки, електронну комерцію та високошвидкісні транзакційні системи. MySQL підтримує масштабування та високу

доступність, а також має широкий набір інструментів для розробки та управління базами даних.

4. MongoDB: Є документальною системою управління базами даних, яка дозволяє зберігати та опрацьовувати нереляційні дані. MongoDB підтримує масштабування та високу доступність, а також має вбудований механізм шарування, що дозволяє розміщувати дані на різних серверах.

5. PostgreSQL: Є відкритою реляційною системою управління базами даних, яка надає підтримку для багатьох типів даних та має розширюваний архітектурний підхід. PostgreSQL підтримує масштабування та високу доступність, а також надає велику кількість інструментів для розробки та управління базами даних.

6. SQLite: Є легкового виконання реляційної системи управління базами даних, яка дозволяє зберігати дані на клієнтському комп'ютері. SQLite підтримує велику кількість типів даних та має малу вимогу до ресурсів системи, що робить його ідеальним рішенням для додатків з обмеженими ресурсами.

7. Firebase: Firebase є обlačною системою управління базами даних, розробленою компанією Google, яка спеціалізується на розробці мобільних та веб-додатків. Firebase підтримує різні типи даних та надає можливості реального часу для збереження та синхронізації даних на різних пристроях. Firebase також має багато інших функцій, таких як аутентифікація користувачів, зберігання файлів та аналітика, що робить його дуже популярним серед розробників додатків.

8. InfluxDB: InfluxDB є нереляційною системою управління базами даних, яка спеціалізується на зберіганні та аналізі даних датчиків та IoT-пристроїв. InfluxDB має вбудовані механізми масштабування та високої доступності, а також має багато інших функцій, таких як запити за часовими мітками, статистичні аналізи та візуалізація даних. InfluxDB є дуже гнучкою та може бути використана для різних типів додатків, що використовують дані датчиків.

Таблиця 1.2 – опис рішень баз даних для написання під системи IoT.

Назва бази даних	Тип бази даних	Мова запитів	Підтримувані операційні системи	Особливості
MongoDB	NoSQL	MongoDB	Windows, Linux, macOS	Гнучкість у роботі зі структурованими та неструктурованими даними, горизонтальне масштабування
MySQL	Relational	SQL	Windows, Linux, macOS	Відкритий код, підтримка транзакцій, гарна підтримка індексів
PostgreSQL	Relational	SQL	Windows, Linux, macOS	Відкритий код, підтримка транзакцій, можливість використовувати різні типи даних, розширені можливості з безпеки
InfluxDB	Time Series	InfluxQL	Windows, Linux, macOS	Оптимізована для роботи з часовими рядами, гарна підтримка агрегаційних функцій
Cassandra	NoSQL	CQL	Windows, Linux, macOS	Гнучкість у роботі зі структурованими та неструктурованими даними, можливість масштабування на кілька вузлів, відмінна мінливість, підтримка транзакцій на рівні запису
Firebase	Real-time	Firebase API	Windows, Linux, macOS	Інтегрований механізм синхронізації, можливість прямого зв'язку з мобільними додатками

Продовження таблиці 1.2 – опис рішень баз даних для написання під системи IoT.

Назва бази даних	Тип бази даних	Мова запитів	Підтримувані операційні системи	Особливості
Oracle Database	Relational	SQL	Windows, Linux, macOS	Надійна платформа, підтримка великих обсягів даних, можливість використовувати більшість функцій SQL, відмінна масштабованість
SQLite	Relational	SQL	Windows, Linux, macOS	Невеликий розмір, легкість використання та налаштування, відсутність необхідності в окремому сервері для роботи з базою даних, підтримка транзакцій, швидкий доступ до даних
Microsoft SQL Server	Relational	T-SQL	Windows, Linux, macOS	Гарна підтримка індексів та транзакцій, можливість використовувати більшість функцій SQL, відмінна масштабованість

За результатами порівняльної характеристики баз даних, що використовуються в розробці розумних будинків, сформувався висновок про те, що кожна база даних має свої переваги та недоліки і обирається в залежності від конкретних потреб проекту. Наприклад, якщо потрібна висока продуктивність та підтримка великої кількості користувачів, то більш доцільно використовувати PostgreSQL або MongoDB. У свою чергу, якщо важливо максимально спростити розробку та налаштування бази даних, то можна обрати SQLite. Крім того, вибір бази даних залежить від характеру даних, що зберігаються, та вимог до безпеки та зберігання історії змін. Отже, при розробці розумного будинку важливо враховувати

всі ці фактори і обирати базу даних, яка краще підходить для конкретного проекту.

### 1.5 Постановка задачі.

Метою дипломного проекту є створення інформаційної системи та бази даних для "розумного" дому, яка дозволить автоматизувати роботу різних пристроїв та забезпечить комфортний та безпечний режим проживання у будинку. Реалізація проекту передбачає розробку бази даних, інтерфейсу для користувача, алгоритмів автоматизації роботи пристроїв, системи безпеки та контролю доступу, а також проведення тестування та відладки системи. Очікуваними результатами є створення робочої інформаційної системи, зручного та зрозумілого для користувача інтерфейсу, а також функціональності для автоматизації роботи пристроїв у будинку та забезпечення безпеки проживання[9].

Основні завдання дипломного проекту:

1. Розробка бази даних для зберігання інформації про пристрої, які підключені до "розумної" системи будинку.
2. Розробка інтерфейсу для управління "розумним" домом. Інтерфейс має бути зрозумілим та зручним для користувача.
3. Розробка алгоритмів для автоматизації роботи різних пристроїв у будинку, наприклад, системи опалення, кондиціонування, освітлення, охоронної системи та інших.
4. Розробка системи безпеки та контролю доступу.
5. Тестування та відлагодження системи на реальних пристроях та в реальних умовах використання.

Очікуваними результатами дипломного проекту є:

1. Розроблена база даних та інформаційна система "розумного" дому.
2. Розроблений зручний та зрозумілий інтерфейс для користувача.
3. Розроблені та впроваджені алгоритми для автоматизації роботи пристроїв у будинку.

4. Розроблена система безпеки та контролю доступу.
5. Проведене тестування та відлагодження системи.

## **1.6 Проблеми взаємозв'язку баз даних з реалізаціями систем розумного будинку**

Системи розумного будинку можуть включати в себе різні пристрої, такі як датчики руху, камери відеоспостереження, термостати, освітлювальні системи та інші пристрої, які збирають та обробляють великий обсяг даних. Ці дані можуть зберігатись в базах даних, які можуть взаємодіяти з іншими системами розумного будинку. Проте, можуть виникати проблеми з взаємозв'язком баз даних з реалізаціями систем розумного будинку. Наприклад, можуть виникати проблеми зі сумісністю пристроїв та протоколів, які використовуються для передачі даних між пристроями та базами даних. Також можуть виникати проблеми з недостатньою захистом даних, що може призвести до порушення приватності користувачів та можливого зловживання даними. Іншою проблемою може бути складність інтеграції баз даних з іншими системами розумного будинку, особливо якщо ці системи розроблені різними виробниками та використовують різні протоколи та стандарти. Щоб уникнути цих проблем, необхідно ретельно планувати та досліджувати взаємозв'язок між базами даних та системами розумного будинку перед їх впровадженням. Також необхідно враховувати вимоги щодо захисту даних та використання стандартів та протоколів, які забезпечують сумісність та інтеграцію з іншими системами розумного будинку.

З впровадженням систем розумного будинку, може з'явитися необхідність зберігати та обробляти велику кількість різноманітних даних, які можуть включати в себе інформацію про стан пристроїв, датчиків, відеоспостереження, температуру, вологість та інші параметри. Ці дані збираються та передаються до центральної системи розумного будинку для обробки та аналізу. Для забезпечення роботи таких систем, необхідні бази

даних, які можуть зберігати, оновлювати та зчитувати дані з різних джерел. Проте, при взаємозв'язку баз даних з системами розумного будинку, можуть виникати різні проблеми. Одна з таких проблем – це проблеми зі сумісністю між різними протоколами та стандартами, які використовуються для передачі даних між пристроями та базами даних. Наприклад, якщо одна система використовує стандарт ZigBee, а інша – Wi-Fi, то можуть виникати складнощі з передачею даних між ними. Також можуть виникати проблеми зі стандартами забезпечення безпеки та приватності, що може призвести до можливого зловживання даними. Іншою проблемою може бути складність інтеграції баз даних з іншими системами розумного будинку, особливо якщо ці системи розроблені різними виробниками та використовують різні протоколи та стандарти. Це може призвести до складнощів зі зберіганням та передачею даних між різними системами, що може вплинути на роботу всієї системи розумного будинку. Для забезпечення ефективної та безпечної роботи систем розумного будинку, необхідно ретельно планувати та досліджувати потенційні проблеми, які можуть виникнути при взаємозв'язку баз даних з системами розумного будинку. Іншою проблемою може бути обмежена масштабованість баз даних, особливо якщо система розумного будинку постійно збільшується. Чим більше даних, тим більше ресурсів необхідно для зберігання та обробки цих даних. Якщо база даних не може масштабуватися, то це може призвести до проблем з продуктивністю та надійністю всієї системи. Якщо дані зберігаються в різних форматах або з різними структурами, то це може затруднити їх обробку та аналіз, що може призвести до зниження ефективності та точності роботи системи розумного будинку. Крім того, при роботі з базами даних можуть виникати проблеми з безпекою та конфіденційністю даних. Недостатня захищеність баз даних може призвести до викрадення та витоку даних, що може мати серйозні наслідки для приватності користувачів та безпеки їхніх будинків. Отже, для успішної реалізації систем розумного будинку, необхідно забезпечити ефективний взаємозв'язок між базами даних та іншими

системами розумного будинку, забезпечити стандартизацію даних та їх форматів, розробити ефективні засоби захисту та безпеки даних та забезпечити масштабованість баз даних для забезпечення продуктивності та надійності системи розумного будинку.

## **2. Проектування інформаційного забезпечення**

### **2.1. Ключові питання постановки завдання проектування інформаційного забезпечення**

Для написання інформаційного забезпечення розумного будинку необхідно розглянути декілька ключових питань, таких як:

1. Визначення дерева функцій та можливостей розумного будинку: це допоможе зрозуміти, які бази даних і яке програмне забезпечення необхідно розробити.

2. Вибір технології зв'язку: розумний будинок використовує різні технології зв'язку, такі як Wi-Fi, Bluetooth, ZigBee тощо. Необхідно визначитись з технологією, що буде використовуватись, і забезпечити підтримку цієї технології в базі даних та програмному забезпеченні.

3. Забезпечення безпеки: необхідно розглянути питання забезпечення безпеки даних та зв'язку в розумному будинку. Для цього можна використовувати різні технології, такі як шифрування, автентифікація користувача тощо.

4. Розробка інтерфейсу користувача: для забезпечення зручності користування розумним будинком, необхідно розробити інтерфейс користувача. Функціональність інтерфейсу користувача може бути різною, залежно від потреб користувачів, технічних можливостей та можливостей програмного забезпечення.

5. Проектування моделі бази даних (концептуальна, фізична та логічні представлення): для правильної реалізації інформаційного забезпечення та баз даних розумного дому, стоїть задача визначення правильної концептуальної, логічної та фізичної форми представлення бази даних, щоб забезпечити оптимальну продуктивність та безпеку даних. Завдання оперує такими даними як:

- Концептуальна форма представлення бази даних описує відносини між сутностями в системі та їх атрибутами. Це допомагає визначити,

які сутності повинні бути включені до бази даних та як вони пов'язані між собою. Концептуальна модель допомагає зрозуміти, які дані необхідні для системи, та допомагає уникнути зайвих даних.

- Логічна форма представлення бази даних описує відносини між таблицями та структуру даних, які використовуються для зберігання даних. Логічна модель дозволяє визначити, які таблиці повинні бути створені та які взаємозв'язки повинні бути встановлені між ними. Фізична форма представлення бази даних описує, як дані фактично зберігаються на диску або в інших засобах зберігання.
- Фізична модель визначає, які типи даних використовуються для зберігання даних, які індекси та ключі повинні бути встановлені та як дані можна розмістити на диску для оптимальної продуктивності.

6. Тестування та налагодження: після розробки програмного забезпечення та баз даних необхідно провести тестування та налагодження системи. Це допоможе виявити та виправити помилки та недоліки системи перед її використанням.

## **2.2. Дерево функцій**

Дерево функцій – це інструмент для опису ієрархічної структури функцій, які повинні бути виконані для досягнення певної мети. У контексті розробки інформаційного забезпечення та баз даних розумного дому, дерево функцій використане для опису функціональних блоків системи, які повинні бути реалізовані, щоб забезпечити користувачам зручне та ефективне управління різними пристроями та системами, що входять до складу розумного дому. Але в контексті науково дослідницької роботи дерево функцій слугує для відображення можливих функцій і не потребує виконання всіх аспектів в повній мірі.

Головна мета роботи представлена вершиною дерева функцій, яке включає гілки, що представляють різні функції або завдання. Ці завдання пов'язані з досягненням головної мети та інших підзавдань. Функціональна

декомпозиція системи базується на її функціональних частинах, кожен елемент яких розглядається окремо з урахуванням його ролі в системі.

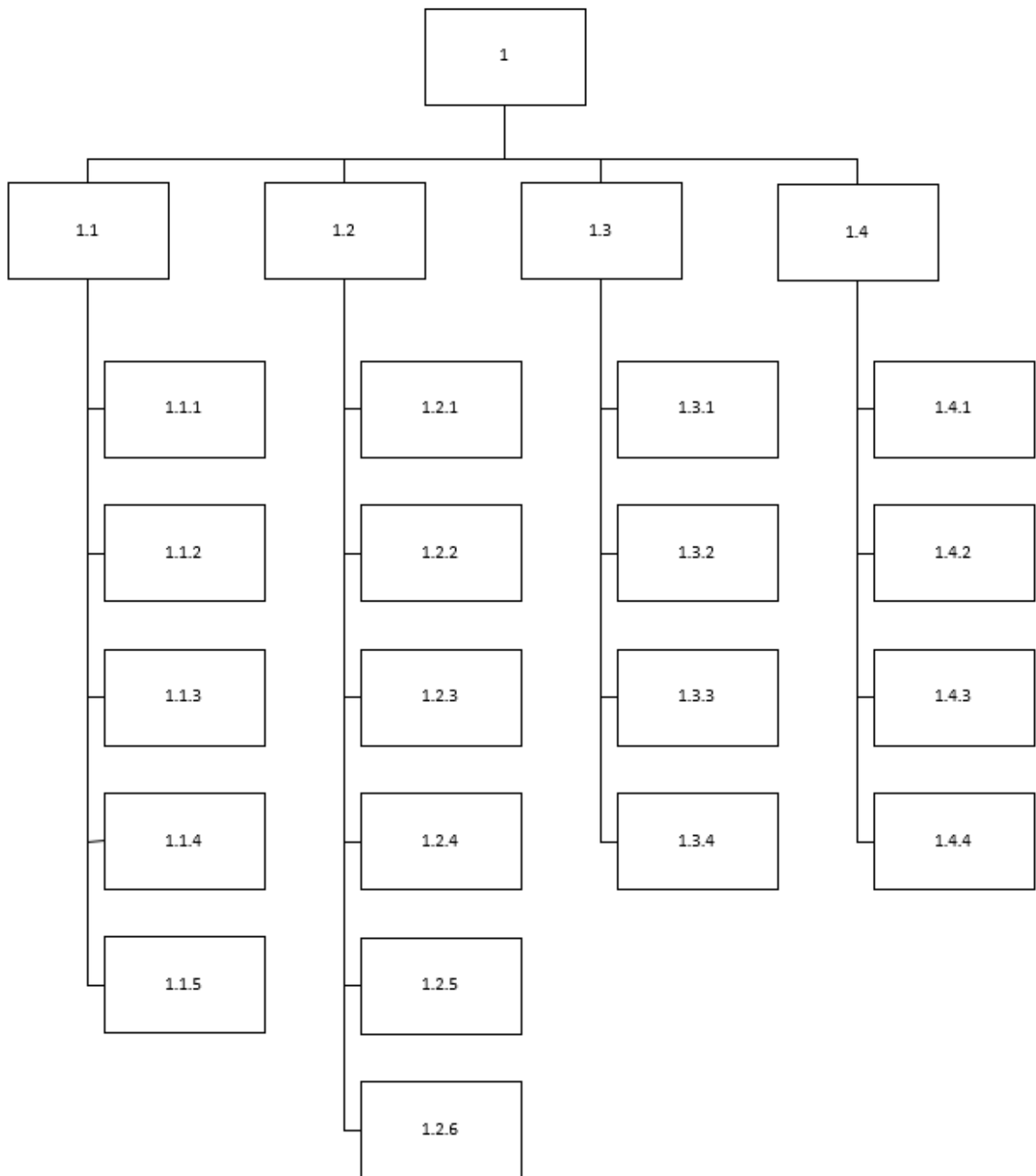


Рисунок 2.1 – Дерево функцій розроблюваного проєкту.

1 Створення інформаційної системи та баз даних "розумного" будинку

1.1 Функції збору інформації з сенсорів:

1.1.1 Зчитування даних з датчиків температури та вологості повітря

1.1.2 Зчитування даних з датчиків рівня освітленості

1.1.3 Зчитування даних з датчиків руху

- 1.1.4 Зчитування даних з датчиків відкриття та закриття дверей та вікон
  - 1.1.5 Зчитування даних з датчиків рівня газу
  - 1.2 Функції керування різними пристроями:
    - 1.2.1 Управління освітленням в кімнатах
    - 1.2.2 Управління опаленням та кондиціонером
    - 1.2.3 Управління вентиляцією
    - 1.2.4 Управління водопостачанням та водовідведенням
    - 1.2.5 Управління дверима та вікнами
    - 1.2.6 Управління системою безпеки (сигналізацією та відеоспостереженням)
  - 1.3 Функції моніторингу системи:
    - 1.3.1 Аналіз даних з датчиків температури та вологості повітря для підтримки оптимального рівня комфорту
    - 1.3.2 Моніторинг витрат електроенергії та води
    - 1.3.3 Моніторинг рівня безпеки (виявлення вторгнень та інших небезпек)
    - 1.3.4 Оповіщення власника про аварійні ситуації та проблеми з системою
  - 1.4 Функції збереження та аналізу даних:
    - 1.4.1 Зберігання даних з датчиків у базі даних
    - 1.4.2 Аналіз даних для забезпечення максимальної ефективності роботи системи
    - 1.4.3 Підготовка звітів та статистики щодо використання енергії та води
    - 1.4.4 Забезпечення можливості редагування та налаштування системи
- 2.3. Вибір технології зв'язку**

Працездатність системи "розумного" дому має необхідність вибору технології зв'язку, яка дозволить підключати до неї різні пристрої та сенсори, а також забезпечить безпеку даних, переданих між ними та сервером. Найбільш популярними технологіями зв'язку для розумного

дому є Wi-Fi, Bluetooth, ZigBee та Z-Wave. Розглянемо їх характеристики у порівнянні:

Таблиця 2.1 – Порівняльна характеристика технологій зв'язку

Технологія	Частота	Швидкість передачі даних	Дальність передачі даних	Безпека
Wi-Fi	2.4GHz/5GHz	до 300Mbps/600Mbps	до 100м	WPA2-PSK
Bluetooth	2.4GHz	до 24Mbps	до 100м	AES-128
ZigBee	2.4GHz	до 250kbps	до 75м	AES-128
Z-Wave	868.4MHz/908.4MHz	до 100kbps	до 100м	AES-128

Найбільш популярні технології – Wi-Fi та Bluetooth – мають досить високу швидкість передачі даних, але можуть бути не досить надійними та безпечними в деяких умовах. ZigBee та Z-Wave є більш стійкими до перешкод та мають надійну безпеку, проте швидкість передачі даних є меншою.

Кожна з технологій має свої переваги та недоліки. На основі порівняння наведеної вище таблицю та враховуючи специфіку даної роботи, є рекомендація використання ZigBee або Z-Wave технологій зв'язку. Ці технології є більш стійкими до перешкод та мають надійну безпеку, що є важливим для системи "розумного" дому, де забезпечення конфіденційності та цілісності даних є критично важливою. Також, ZigBee та Z-Wave мають достатньо швидкість передачі даних для потреб розумного дому, що дозволяє забезпечити зручність використання системи користувачем. Але для дослідницької роботи, обрано використовувати технології Wi-Fi та Bluetooth.

## 2.4. Вибір технології забезпечення безпеки

Для забезпечення безпеки інформації в системі "розумного" дому можна використовувати різні технології, такі як:

- SSL / TLS: Ці протоколи забезпечують безпеку передачі даних, шифруючи їх і запобігаючи доступу до них з боку зловмисників.
- VPN: Це безпечний тунель між двома пристроями, що забезпечує захист передачі даних з одного кінця до іншого.
- Firewalls: Це пристрої, які контролюють доступ до системи і забезпечують захист від несанкціонованого доступу.
- Антивірусні програми: Ці програми виявляють інфіковані файли і захищають систему від шкідливих програм.
- Ідентифікація та автентифікація: Ці технології забезпечують захист від несанкціонованого доступу.

Таблиця 2.2 – Порівняльна характеристика технологій безпеки

Технологія	Переваги	Недоліки
SSL / TLS	Ефективний захист від перехоплення даних	Потребує додаткових ресурсів для шифрування даних
VPN	Забезпечує безпечну передачу даних	Потребує додаткових налаштувань
Firewalls	Ефективний захист від несанкціонованого доступу	Може обмежувати легітимний доступ до системи
Антивірусні програми	Захист від шкідливих програм	Може сповільнювати роботу системи
Ідентифікація та автентифікація	Забезпечує захист від несанкціонованого доступу	Можливість зламування захисту

Найкращою технологією забезпечення безпеки буде використання SSL / TLS. Ця технологія є ефективною у захисті від перехоплення даних та не вимагає значних ресурсів для їх шифрування. Застосування інших технологій, таких як VPN, Firewalls, Антивірусні програми та ідентифікація та автентифікація може бути виправданим залежно від конкретних потреб проекту, проте в даному випадку SSL / TLS є оптимальним вибором з точки зору ефективності та витрат на реалізацію.

## **2.5. Вибір інтерфейсу користувача**

### ***2.5.1. Вимоги до інтерфейсу***

Для розробки інформаційного забезпечення та баз даних розумного дому необхідний інтерфейс, який би дозволяв легко та зручно взаємодіяти з даними та змінювати їх. Основні вимоги до інтерфейсу створення інформаційного забезпечення та баз даних мають бути такі:

1. Легкість використання: інтерфейс повинен бути простим та зрозумілим для будь-якого користувача без великого досвіду роботи з базами даних та програмним забезпеченням.
2. Функціональність: інтерфейс повинен забезпечувати можливість додавання, видалення та редагування даних у базах даних, а також взаємодію з іншими системами, такими як мобільні додатки та інтернет-сервіси.
3. Зручність відображення даних: інтерфейс повинен мати зручну та логічну структуру відображення даних, що дозволить користувачеві легко орієнтуватися та знайти потрібну інформацію.
4. Надійність та безпека: інтерфейс повинен мати захист від несанкціонованого доступу до даних та збереження їх в надійному місці.

5. Можливість масштабування: інтерфейс повинен дозволяти легко розширювати бази даних та додавати нові функціональні можливості без перезапуску системи.
6. Підтримка стандартів: інтерфейс повинен дотримуватися стандартів програмування та баз даних, що дозволить його легко інтегрувати з іншими системами та розробляти додаткові функції.

Загалом, інтерфейс створення інформаційного забезпечення та баз даних має бути зрозумілим, зручним та наділяти додаткові функції. Наприклад, важливо, щоб інтерфейс підтримував мову структурованого запиту до баз даних SQL (Structured Query Language), що є стандартом в області управління реляційними базами даних.

Окрім цього, важливо, щоб інтерфейс мав можливість відслідковувати та логувати дії користувачів. Це дозволить забезпечити контроль за даними та зберіганням історії змін, що може бути корисним у випадку виявлення помилок або потреби відновити попередні налаштування.

Інтерфейс повинен також дозволяти користувачам забезпечувати автоматичне оновлення та поновлення баз даних, що дозволяє уникнути можливих проблем зі сумісністю між версіями та забезпечити надійну та стабільну роботу системи.

Нарешті, важливо, щоб інтерфейс забезпечував можливість імпорту та експорту даних з інших джерел. Це дозволяє користувачам ефективно обмінюватись даними з іншими системами та розробляти різноманітні інтеграції.

### ***2.5.2. Загальні можливості структури інтерфейсу***

Для забезпечення зручного та інтуїтивно зрозумілого інтерфейсу розробки інформаційного забезпечення та баз даних розумного будинку, його структура може включати наступні компоненти:

1. Панель навігації: це важливий елемент інтерфейсу, що дозволяє користувачам легко переміщатись між різними розділами програми та виконувати необхідні дії. Панель навігації може містити посилання на такі розділи, як "Домашня сторінка", "Керування пристроями", "Управління правами користувачів", "Системні налаштування", тощо.
2. Розділ "Керування пристроями": цей розділ може містити інформацію про всі пристрої, що підключені до системи розумного будинку, та дозволяти користувачам керувати ними. Тут можуть бути доступні такі функції, як вмикання та вимикання пристроїв, налаштування їх параметрів, створення та редагування розкладів роботи пристроїв тощо.
3. Розділ "Управління правами користувачів": цей розділ може містити інформацію про всіх користувачів системи та їх рівень доступу до функцій та даних. Тут можуть бути доступні такі функції, як створення нових користувачів, надання та зміна рівнів доступу, налаштування паролів та інші аспекти управління користувачами.
4. Розділ "Системні налаштування": цей розділ може містити інформацію про налаштування системи та баз даних, а також дозволяти користувачам змінювати різні параметри. Тут можуть бути доступні такі функції, як налаштування зв'язку з Інтернетом, налаштування автоматичного оновлення, тощо.
5. Розділ "Домашня сторінка": цей розділ може містити основну інформацію про систему та найважливіші функції, які можуть бути виконані з використанням інтерфейсу. Наприклад, на цій сторінці можуть бути доступні швидкі посилання на найбільш використовувані функції, а також повідомлення про стан пристроїв або інформація про сповіщення та нагадування.
6. Розділ "Створення та налаштування баз даних": цей розділ може містити інструменти для створення та редагування баз даних, налаштування зв'язків між ними, додавання нових таблиць та полів, встановлення індексів, забезпечення безпеки та інші налаштування.

7. Розділ "Пошук та фільтрація даних": цей розділ може містити інструменти для пошуку та фільтрації даних, що містяться в базі даних. Користувачі можуть шукати та фільтрувати дані за певними параметрами, такими як дата, тип даних, пристрій, який збирав дані тощо.
8. Розділ "Візуалізація даних": цей розділ може містити інструменти для візуалізації даних, зібраних системою розумного будинку. Користувачі можуть створювати графіки, діаграми та інші візуальні елементи, щоб краще розуміти дані та виявляти тенденції.
9. Розділ "Налаштування сповіщень": цей розділ може містити інструменти для налаштування сповіщень, які надходять з пристроїв розумного будинку. Користувачі можуть встановлювати правила та налаштовувати рівні сповіщень для різних пристроїв, щоб бути в курсі стану системи.
10. Розділ "Аналітика та статистика": цей розділ може містити інструменти для аналізу та відображення статистики використання системи розумного будинку. Користувачі можуть переглядати статистику використання пристроїв, аналізувати тенденції та виявляти незвичайні показники.
11. Розділ "Журнал подій": цей розділ може містити інформацію про всі події, які відбуваються в системі розумного будинку, такі як зміни стану пристроїв, нагадування про розклад робіт, сповіщення про непередбачені ситуації тощо. Журнал подій може допомогти користувачам відслідковувати та аналізувати події в системі.
12. Розділ "Налаштування профілю користувача": цей розділ може містити інформацію про профіль користувача, таку як ім'я, контактну інформацію, налаштування безпеки та приватності. Користувачі можуть змінювати свої налаштування, щоб забезпечити безпеку та приватність своїх даних.
13. Розділ "Довідка та підтримка": цей розділ може містити інформацію про довідку та підтримку для користувачів. Користувачі можуть отримувати

допомогу та відповіді на свої запитання, переглядати часті запитання та відповіді, а також звертатися до служби підтримки.

Графічне представлення вимог зображено на рисунку 2.2

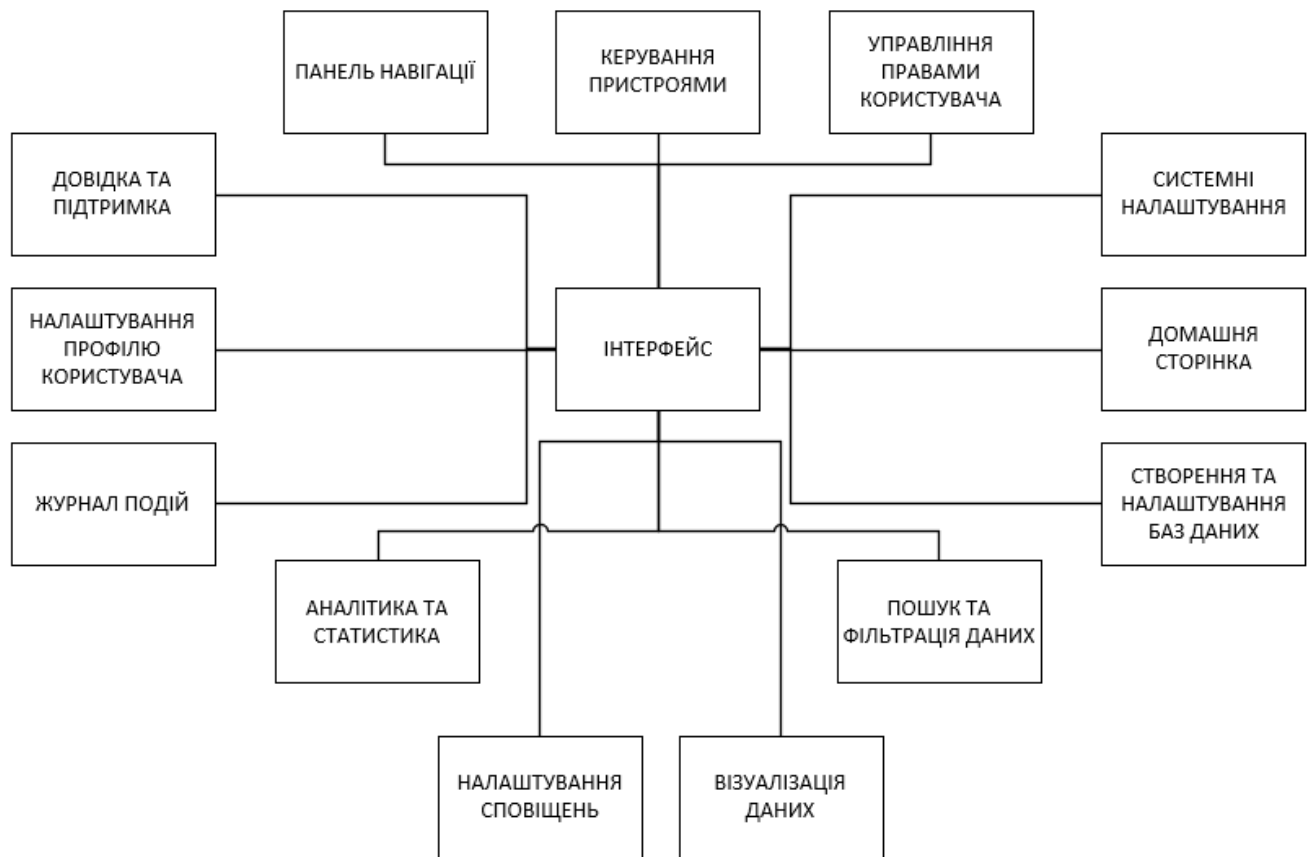


Рисунок 2.2 – графічне представлення загальних можливостей структури інтерфейсу

### ***2.5.3. Вибір типу інтерфейсу користувача***

Дана підтема досліджує різні типи інтерфейсу, такі як текстовий, графічний, віртуальна реальність та інші. Для кожного типу інтерфейсу розглядаються його переваги та недоліки, включаючи легкість використання, швидкість виконання завдань, підтримку користувачів з

обмеженими можливостями та ін. Тож опис «простих» типів описаний в таблиці нижче:

Таблиця 2.3 – Порівняльна характеристика технологій зв'язку

Тип інтерфейсу	Опис	Переваги	Недоліки
Графічний	Інтерфейс, з графічних елементів	Легкий у використанні,	Потребує певних знань, обмежує функціонал

Продовження таблиці 2.3 – Порівняльна характеристика технологій зв'язку

Тип інтерфейсу	Опис	Переваги	Недоліки
Текстовий	Інтерфейс, на використанні текстових команд	Дозволяє доступ до всіх функцій, легкий у використанні	Складний у вивченні, не так зручний у використанні
Голосовий	Інтерфейс, на розпізнаванні мови	Забезпечує зручність, легкий у використанні	Потребує використання спеціального обладнання
Сенсорний	Інтерфейс, на використанні сенсорних елементів	Забезпечує зручність, не вимагає додаткових знань	Дорогий, складний у використанні
VR інтерфейс	Інтерфейс, заснований на використанні VR	Забезпечує максимальну іммерсивність, зручність	Дорогий

Опис «складних» варіантів, виведених із таблиці вище:

1. Веб–інтерфейс: інтерфейс користувача, який доступний через веб–браузер. Це може бути простий веб–інтерфейс, що дає можливість користувачеві взаємодіяти з домашніми пристроями з будь–якого місця, де є доступ до Інтернету.
2. Мобільний додаток: додаток, який можна встановити на смартфон або планшет. Мобільний додаток може мати більш простий та зручний інтерфейс, спеціально адаптований для маленьких екранів. Крім того, мобільний додаток може використовувати додаткові можливості смартфона, такі як GPS або датчики руху.
3. Голосовий інтерфейс: інтерфейс, що дає можливість користувачеві керувати домашніми пристроями за допомогою голосових команд. Голосовий інтерфейс може бути зручним для людей, які мають обмежені можливості взаємодії з домашніми пристроями через інші інтерфейси.

Отже, з огляду на дослідницьку мету проекту, оптимальним вибором став веб–інтерфейс, але з урахуванням складності в деяких аспектах, робота продемонструє використання різних інтерфейсів. Вони забезпечать простоту використання та масштабованість, знизять вартість розробки та підтримки. Також, дані інтерфейси забезпечать широку доступність для користувачів, оскільки можуть бути реалізованими на будь–якому пристрої з доступом до Інтернету.

## **2.6. Проектування моделі бази даних**

База даних розумного будинку повинна містити три рівні представлення: концептуальний, фізичний та логічний[10,11].

- Концептуальне представлення бази даних – це високорівневе описання структури та залежностей між даними в предметній області розумного будинку. Наприклад, можна визначити таблиці, які містять інформацію про пристрої, розташовані в будинку, їхні властивості та параметри, інформацію про розклади їх роботи та інші

дані. Концептуальне представлення допомагає зрозуміти взаємозв'язки між різними об'єктами в системі та визначити ключові аспекти бази даних.

- Фізичне представлення бази даних – це опис технічних аспектів зберігання даних, таких як формати файлів, типи дискових пристроїв, що використовуються для зберігання інформації про пристрої розумного будинку. Фізичне представлення бази даних включає в себе детальні відомості про розташування та організацію даних на дисках, а також процеси забезпечення безпеки і зменшення ризиків втрати даних.
- Логічне представлення бази даних – це опис способів, якими дані можуть бути доступні та опрацьовуватися за допомогою запитів. Логічне представлення включає в себе опис схем бази даних, запитів, що дозволяють виконувати різні операції з даними, та інших елементів, що дозволяють взаємодіяти з інформацією про пристрої розумного будинку. Логічне представлення бази даних дозволяє здійснювати пошук, фільтр тощо.

Наступні розділи роботи наводять детальний опис кожного рівня представлення бази даних хабу розумного будинку.

### ***2.6.1. Створення концептуальної моделі хабу «розумного» будинку***

Концептуальне представлення бази даних розумного будинку почалось з аналізу предметної області. Основне питання розбору є визначення об'єктів та їх взаємозв'язків в системі розумного будинку. В рамках даної роботи, вибір впав на такі об'єкти, як пристрої освітлення, опалення, вентиляції, системи безпеки тощо, а також інформацію про присутність людей в будинку та їх поведінку. Що в ході роботи привело до створення таких сутностей як: ”користувачі”, ”пристрої”, ”пристрої користувачів”, ”сенсори”, ”дані від сенсорів”, ”дії користувачів” та ”налаштування пристроїв”.

Далі, створена модель, яка описує зв'язки між різними об'єктами в системі. Для цього використано діаграмами взаємозв'язків та діаграми класів. Дані діаграми показали зв'язки та відношення, такі як залежність сутності "користувачів" та "пристроїв" від сутності "пристроїв користувачів", залежності блоку "Дані від сенсорів" з сутністю "Сенсори", "Пристрої" та "Користувачі", та іншими, зображеними на рисунку нижче, тощо.

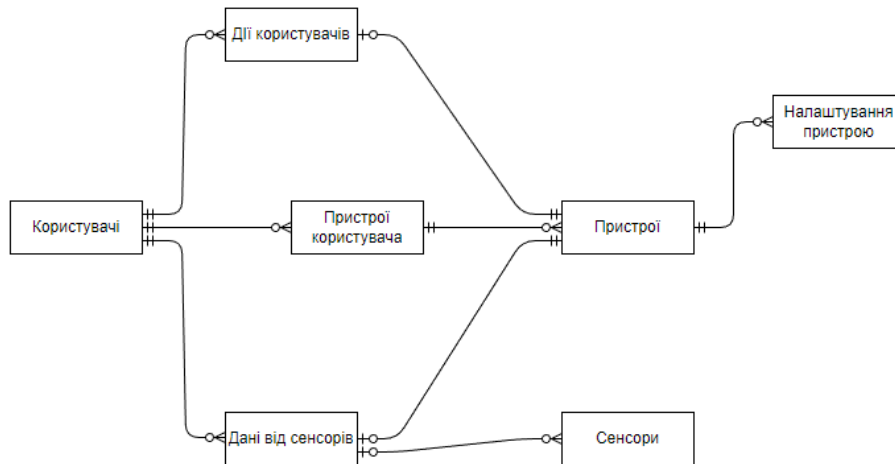


Рисунок 2.2 – концептуальна модель опису зв'язків розроблювальної бази даних хабу розумного будинку

Крім того, необхідно визначити ключові атрибути кожного об'єкту. Наприклад, для сутності користувача це стали такі атрибути, як унікальний ідентифікатор користувача, ім'я користувача, електронна пошта користувача, пароль користувача та рівень доступу користувача, тощо.

Детальний опис кожного атрибуту визначено та показано на рисунку 2.3 нижче.

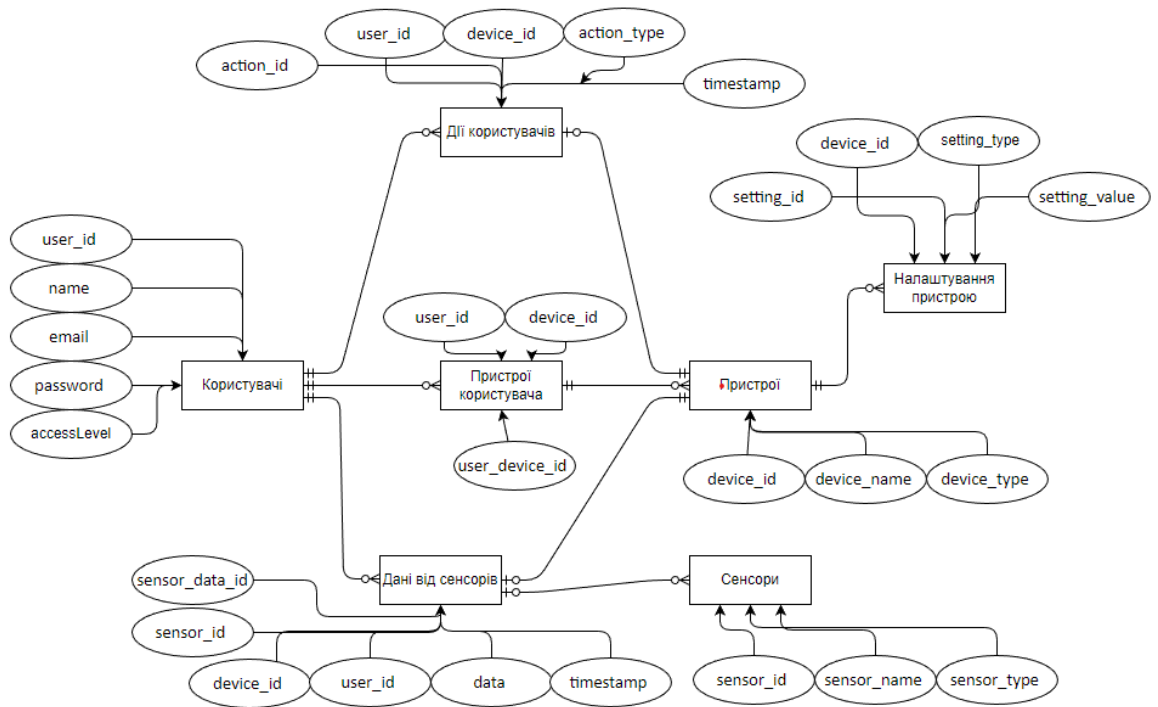


Рисунок 2.3 – концептуальна модель опису атрибутів розроблювальної бази даних хабу розумного будинку

Завершуючи створення концептуального представлення бази даних розумного будинку, створено та описано всі знайдені об'єкти та їх зв'язки між собою, а також визначені ключові атрибути. Це надало загальний огляд предметної області та визначило структуру бази даних для подальшої розробки фізичного та логічного представлень.

### 2.6.2. Створення логічної моделі хабу «розумного» будинку

Після створення концептуального представлення бази даних розумного будинку, наступним етапом стало створення логічного представлення. Логічне представлення полягає в перетворенні концептуальної моделі в модель, яка враховує особливості конкретної системи управління базою даних.

Детальний опис визначено та показано на рисунку 1.6.2.1 нижче.

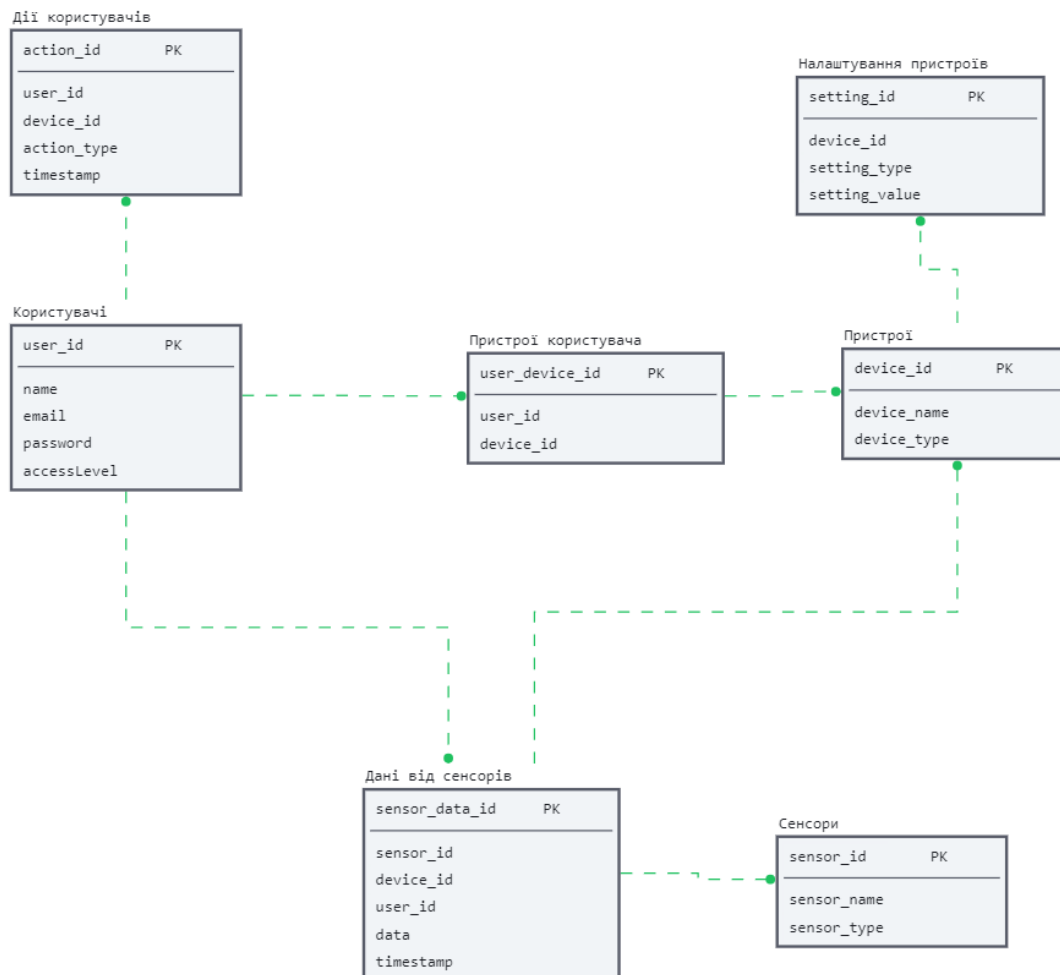


Рисунок 2.4 – описана логічна модель з урахуванням взаємозв’язків.

На етапі створення логічного представлення визначено сутності, атрибути та зв’язки між ними, відповідно до концептуальної моделі. При цьому, враховані особливості використовуваної системи управління базою даних. Враховано, що в базі даних розумного будинку використані різні типи датчиків, що вимагає створення окремих таблиць для зберігання їх даних.

Також на етапі створення логічного представлення були визначені обмеження цілісності даних, такі як обов’язковість заповнення певних атрибутів, обмеження на діапазон значень атрибутів та заборона на видалення записів, що мають зв’язки з іншими записами в базі даних.

Завершуючи створення логічного представлення бази даних розумного будинку, були описати всі таблиці та їх зв’язки, а також визначені обмеження цілісності даних застосовані в програмування самого проєкту. Це дозволило визначити структуру бази даних, яка відповідає

концептуальній моделі та потребам системи управління базою даних і перейти до створення фізичного представлення.

### 2.6.3. Створення фізичної моделі хабу «розумного» будинку

Після створення логічного представлення бази даних, наступним етапом стало створення фізичного представлення. Фізичне представлення полягає в перетворенні логічної моделі в модель, яка відображає реальні характеристики фізичної реалізації бази даних та їх типів даних.

На етапі створення фізичного представлення була визначена структура та параметри фізичної реалізації бази даних. Це включило в себе вибір типу сервера баз даних, налаштування параметрів бази даних та визначення способу зберігання даних.

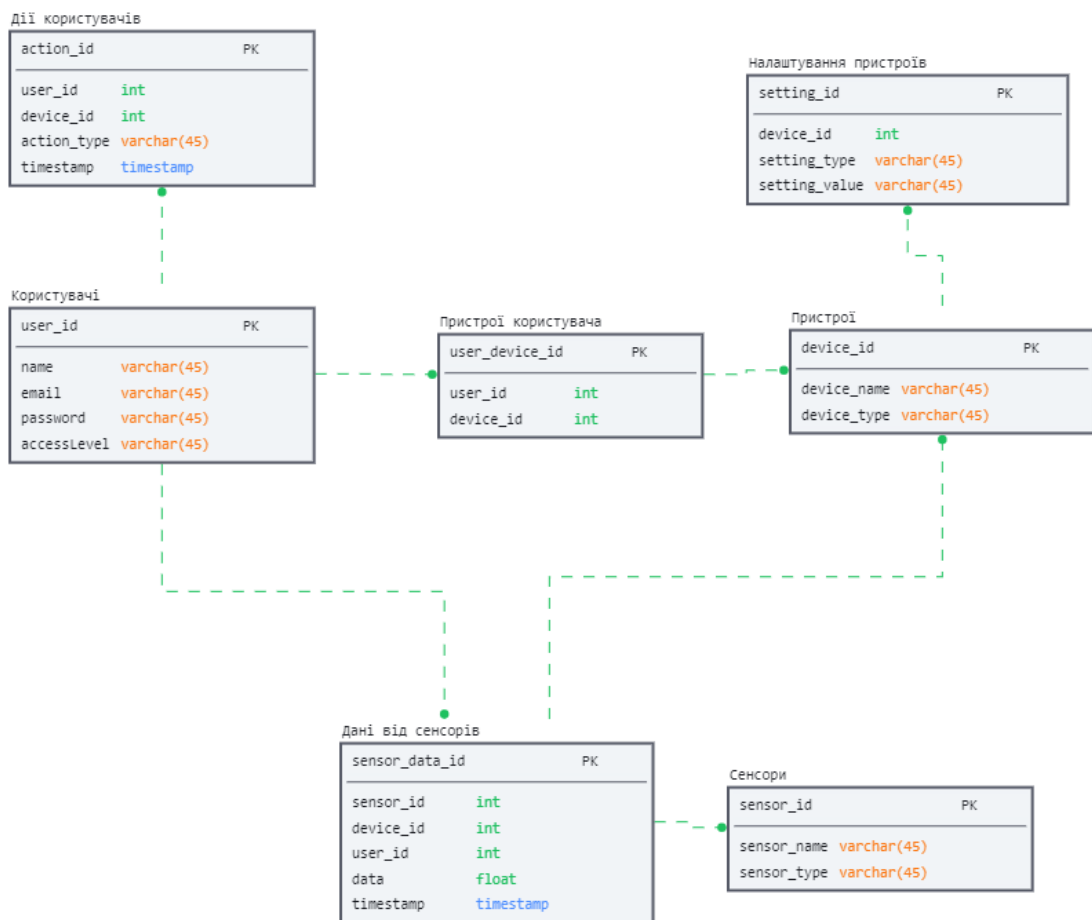


Рисунок 2.5 – фізична модель база даних

Також на етапі створення фізичного представлення були визначені параметри забезпечення безпеки даних, такі як:

- Аутентифікація та авторизація – процеси ідентифікації користувачів та визначення їхніх прав доступу до даних. Для цього доцільно використовувати різні механізми, такі як логіни та паролі, двофакторна аутентифікація, біометричні технології тощо.
- Шифрування даних – процес перетворення звичайного тексту в зашифрований, що забезпечує конфіденційність даних. Для цього доцільно використовувати різні алгоритми шифрування, такі як AES, RSA, Blowfish тощо.
- Захист від SQL-ін'єкцій – це захист від вразливості, при якій зловмисник може використати SQL-запит для витягування або зміни даних в базі даних. Для захисту від цього доцільно використовувати параметризовані запити, а також фільтрацію введених користувачем даних.
- Резервне копіювання – забезпечення можливості відновлення даних в разі їх втрати або пошкодження. Для цього доцільно використовувати різні методи резервного копіювання, такі як повний бекап, інкрементальне копіювання тощо.
- Моніторинг та аудит – процес відстеження доступу до даних та змін в базі даних з метою виявлення потенційних загроз безпеці. Для цього доцільно використовувати системи моніторингу та аудиту, такі як Microsoft SQL Server Audit, Oracle Audit Vault тощо.

Завершуючи створення фізичного представлення бази даних розумного будинку, забезпечено її реалізацію згідно з встановленими параметрами та забезпечено правильну інтеграцію з системою управління базою даних. Це дозволило забезпечити ефективне та безпечне функціонування бази даних та системи розумного будинку в цілому

## **2.7. Загальний вигляд сутностей бази даних**

Після опису створення концептуального, логічного, та фізичного представлення бази даних, сформовано загальний вид сутностей(таблиць)

для початку розробки, програмування та подальшої розробки елементарної системи «розумного» будинку.

#### Таблиця "Користувачі" (Users)

- user\_id (INT, PRIMARY KEY) – унікальний ідентифікатор користувача
- name (VARCHAR) – ім'я користувача
- email (VARCHAR) – електронна пошта користувача
- password (VARCHAR) – пароль користувача
- accessLevel(VARCHAR) – рівень доступу користувача

#### Таблиця "Пристрої" (Devices)

- device\_id (INT, PRIMARY KEY) – унікальний ідентифікатор пристрою
- device\_name (VARCHAR) – назва пристрою
- device\_type (VARCHAR) – тип пристрою (наприклад, освітлення, термостат, замок на двері, тощо)

#### Таблиця "Пристрої користувача" (User\_Devices)

- user\_device\_id (INT, PRIMARY KEY) – унікальний ідентифікатор зв'язку між користувачем та пристроєм
- user\_id (INT, FOREIGN KEY) – ідентифікатор користувача, що володіє пристроєм
- device\_id (INT, FOREIGN KEY) – ідентифікатор пристрою

#### Таблиця "Сенсори" (Sensors)

- sensor\_id (INT, PRIMARY KEY) – унікальний ідентифікатор сенсору
- sensor\_name (VARCHAR) – назва сенсору
- sensor\_type (VARCHAR) – тип сенсору (наприклад, температура, вологість, рух, тощо)

#### Таблиця "Дані від сенсорів" (Sensor\_Data)

- sensor\_data\_id (INT, PRIMARY KEY) – унікальний ідентифікатор запису даних від сенсору

- sensor\_id (INT, FOREIGN KEY) – ідентифікатор сенсору, що зібрав дані
- device\_id (INT, FOREIGN KEY) – ідентифікатор пристрою, до якого підключений сенсор
- user\_id (INT, FOREIGN KEY) – ідентифікатор користувача, що має доступ до даних
- data (FLOAT) – числове значення, що вимірюється сенсором
- timestamp (TIMESTAMP) – час, коли був запис даних

#### Таблиця "Дії користувачів" (User\_Actions)

- action\_id (INT, PRIMARY KEY) – унікальний ідентифікатор дії користувача
- user\_id (INT, FOREIGN KEY) – ідентифікатор користувача, що виконав дію
- device\_id (INT, FOREIGN KEY) – ідентифікатор пристрою, на якому виконано дію
- action\_type (VARCHAR) – тип дії (наприклад, увімкнути, вимкнути, змінити налаштування, тощо)
- timestamp (TIMESTAMP) – час, коли була виконана дія

#### Таблиця "Налаштування пристроїв" (Device\_Settings)

- setting\_id (INT, PRIMARY KEY) – унікальний ідентифікатор налаштування
- device\_id (INT, FOREIGN KEY) – ідентифікатор пристрою, для якого налаштування
- setting\_type (VARCHAR) – тип налаштування (наприклад, температура, час роботи, тощо)
- setting\_value (VARCHAR) – значення налаштування

```
MariaDB [(none)]> show databases;
+-----+
| Database |
+-----+
| diplomainfdb |
| information_schema |
| mysql |
| performance_schema |
| phpmyadmin |
| test |
| trytocatmme |
| wpmy |
+-----+
8 rows in set (0.001 sec)
```

Рисунок 2.6 – вибір створеної бази даних

```
MariaDB [(none)]> use diplomainfdb;
Database changed
MariaDB [diplomainfdb]> show tables;
+-----+
| Tables_in_diplomainfdb |
+-----+
| device_settings |
| devices |
| sensor_data |
| sensors |
| user_action |
| user_devices |
| users |
+-----+
7 rows in set (0.000 sec)
```

Рисунок 2.7 – вибір показу таблиць створеної бази даних

### 3. ПРАКТИЧНА РЕАЛІЗАЦІЯ

#### 3.1. Вибір програмного інструментарію

##### 3.1.1. Бази даних

Дві популярні бази даних, які розглянуті для використання для розробки інформаційного забезпечення розумних будинків – MySQL та MongoDB.

**MySQL:** MySQL – це реляційна база даних, яка використовує SQL (Structured Query Language) для роботи з даними. Вона підтримує транзакції ACID (Atomicity, Consistency, Isolation, Durability) і забезпечує високу швидкість операцій з даними. MySQL підтримує багато типів даних, включаючи числа, текст, дати, час та бінарні дані.

**MongoDB:** MongoDB – це нереляційна база даних, яка використовує документи для зберігання даних. Вона підтримує широкий спектр операцій, включаючи збереження, оновлення, вилучення та пошук даних. MongoDB дозволяє зберігати дані у форматі JSON, що дозволяє більш легкий доступ до даних для розробників.

Таблиця 3.1 – порівняння баз даних для використання в практичній реалізації продукту

	MySQL	MongoDB
Тип бази даних	Реляційна	Нереляційна
Мова запитів	SQL	MongoDB Query Language
Схема даних	Статична	Динамічна
Підтримка транзакцій	Так	Ні
Підтримка реплікації	Так	Так
Масштабованість	Горизонтальна та вертикальна	Горизонтальна

Продовження таблиці 3.1 – порівняння баз даних для використання в практичній реалізації продукту

	MySQL	MongoDB
Доступність	Доступний на багатьох операційних системах	Доступний на багатьох операційних системах
Додаткові можливості	Широкі можливості для оптимізації запитів	Можливість зберігання складних даних

Для розробки інформаційного забезпечення та насамперед баз даних обрано для використання MySQL. Це пов'язано з тим, що MySQL є реляційною базою даних, яка підтримує транзакції ACID, що є важливим аспектом для систем з великим обсягом даних та високою надійністю. Також MySQL має багато інструментів для оптимізації запитів та широко використовується в індустрії, що дає можливість студенту отримати додатковий досвід та знання професії. Однак, якщо головною задачею є створити «складний» розумний будинок, в основі якого збереження складних даних, наприклад, документів, які містять вкладені об'єкти та масиви, MongoDB буде кращим вибором. Також MongoDB підтримує горизонтальне масштабування, що дає можливість легко розширювати базу даних за потребою. Загалом, вибір бази даних залежить від конкретних потреб проекту. Враховуючи особливості задачі дипломного проекту та розумних будинків, MySQL є кращим вибором з точки зору надійності та оптимізації запитів.

### ***3.1.2. Мова програмування***

Для проектування інформаційного забезпечення "розумного" дому необхідною задачею є вибір мови програмування, яка найкраще підходить для вирішення поставлених задач. Тож, вибір мови програмування залежить від різних факторів, таких як складність проекту, доступність ресурсів, вимоги до продуктивності, масштабованості та безпеки.

Даний розділ проектування інформаційного забезпечення для "розумного" дому описує ресурсну базу наступних завдань:

- Вибір мови програмування для розробки додатків та систем.
- Розробка структури бази даних та вибір системи управління базами даних (СУБД).
- Визначення алгоритмів та інтерфейсів для взаємодії між різними пристроями та системами. Вибір системи зберігання даних та забезпечення їх безпеки.

Мови програмування для проектування "розумного" дому:

Python – це інтерпретована мова програмування високого рівня, яка зазвичай використовується в розробці веб–додатків та наукових обчислень. Однак, Python також може бути використаний для створення програмного забезпечення для розумного будинку. Python має читабельний та простий синтаксис, що робить його добрим вибором для розробки прототипів та швидкого прототипування.

Java – це об'єктно–орієнтована мова програмування, що має широке застосування в розробці додатків. Java є платформонезалежною, тому програми, написані на Java, можуть працювати на будь–якій операційній системі. Java підтримує багато фреймворків, бібліотек та інших інструментів, що зроблюють його відмінним вибором для розробки розумного будинку.

JavaScript – це мова програмування, яка часто використовується в розробці веб–додатків, але може бути використана й для розробки додатків для розумного будинку. JavaScript дозволяє створювати динамічні та інтерактивні веб–сторінки та додатки. JavaScript має велику кількість бібліотек та фреймворків, що полегшують розробку програмного забезпечення.

C#/C++ – це компільовані мова програмування, які використовуються в розробці високопродуктивного програмного забезпечення. Однією з головних їх переваг є те, що вона дає розробникам більший контроль над пам'яттю та швидкістю виконання програми. Ці мови є дуже потужними,

але вимагає від розробника більш високого рівня знань та досвіду, ніж Python або Java. C++ та C# можуть бути використані для розробки додатків для розумного будинку, які потребують високої продуктивності та низької затримки. Це може включати додатки для керування освітленням та опаленням, моніторингу безпеки, автоматизації процесів в будинку та інше.

Таблиця 3.2 – порівняння мов програмування для використання в практичній реалізації продукту

Мова програмування	Переваги	Недоліки
Python	Простий синтаксис, велика кількість бібліотек, хороша продуктивність	Менш ефективний для високопродуктивних додатків
Java	Велика кількість фреймворків, хороша масштабованість, висока продуктивність	Вимагає великої кількості ресурсів для запуску, менш простий синтаксис
JavaScript	Легка взаємодія з компонентами веб-додатків, широкі можливості для розробки інтерфейсів	Менш ефективний для розробки масштабних додатків
C++/C#	Висока продуктивність, можливість взаємодії з апаратними пристроями, ефективні алгоритми	Складний синтаксис, вимагає більш великої кількості коду

Тож, вибір мови програмування для проектування інформаційного забезпечення "розумного" дому залежить від різних факторів. Для простих додатків можна використовувати мови з простим синтаксисом, такі як Python або JavaScript. Для більш складних додатків з високою продуктивністю та ефективними алгоритмами можна використовувати мови, такі як Java або C++/C#. Крім того, важливим фактором є наявність бібліотек та фреймворків для розробки додатків, які можуть значно полегшити процес розробки.

Під час вибору мови програмування для проектування "розумного" дому, враховано вимоги до продуктивності, функціональності та зручності розробки, щоб забезпечити ефективну та швидку розробку інформаційного забезпечення та баз даних.

Розділ проектування інформаційного забезпечення став важливим етапом у розробці "розумного" дому, оскільки він дозволив визначити необхідні компоненти системи, функціональності та взаємодії між ними. Вибір мови програмування для реалізації цих компонентів залежав від вимог до продуктивності та функціональності, а також від наявності необхідних бібліотек та фреймворків. Тож обрано мову програмування C#.

### ***3.1.3. Програмне середовище***

Для розробки програмного забезпечення для системи розумного дому на мові C# можна використовувати різні програмні середовища, такі як SharpDevelop, MonoDevelop та Microsoft Visual Studio. Однак, у цій роботі порівняємо SharpDevelop та Microsoft Visual Studio.

SharpDevelop – це безкоштовне та відкрите програмне середовище для розробки програм на мовах .NET, включаючи C#. Воно має вбудовану підтримку для системи керування версіями Git, а також може працювати з іншими системами керування версіями. Окрім того, SharpDevelop має можливість візуальної розробки інтерфейсу користувача за допомогою вбудованого дизайнера форм.

Microsoft Visual Studio – це платне програмне середовище для розробки програм на мовах .NET, включаючи C#. Воно має повний набір інструментів для розробки програмного забезпечення, включаючи вбудовану підтримку Git, систему керування проектами та велику кількість бібліотек та інструментів. Також Visual Studio має багато інтегрованих засобів для тестування програм та налагодження коду.

Таблиця 3.3 – порівняння програмних середовищ для використання в практичній реалізації продукту

	SharpDevelop	Microsoft Visual Studio
Вартість	Безкоштовне	Платне(є безкоштовні варіанти)
Підтримка Git	Є	Є
Дизайнер форм	Є	Є
Кількість бібліотек	Обмежені	Велика
Інструменти для тестування та налагодження	Обмежені	Багато інтегрованих засобів
Підтримка мови C#	Є	Є

За результатами порівняльної таблиці можна зробити висновок, що Microsoft Visual Studio є кращим вибором для розробки програм для системи розумного дому на мові C#. Вона має більш широкий функціонал, включаючи велику кількість бібліотек та інструментів, що дозволяє більш ефективно розробляти програмне забезпечення для системи розумного дому. Крім того, Visual Studio має багато інтегрованих засобів для тестування програм та налагодження коду, що робить процес розробки більш продуктивним та менш часовим. Також Visual Studio має вбудовану підтримку Git, що спрощує роботу з системою керування версіями.

Зважаючи на ці переваги, можна зробити висновок, що Microsoft Visual Studio є оптимальним вибором програмного середовища для

розробки інформаційного забезпечення та баз даних розумного дому на мові C#.

### **3.2. Архітектура розроблюваного об'єкта**

Архітектура інформаційного забезпечення розумного будинку включає набір компонентів та систем, які співпрацюють для забезпечення ефективного функціонування та управління будинком. Основними складовими архітектури є[3]:

**Сенсорна мережа:** Це набір різноманітних датчиків, розташованих по всьому будинку, які збирають різні типи даних, наприклад, температуру, освітлення, рух, вологість тощо. Ці дані передаються до центральної системи для подальшої обробки.

**Центральна система:** Це мозок розумного будинку, яка обробляє та аналізує дані, отримані від сенсорної мережі. Вона виконує різні функції, такі як збір даних, аналіз, прийняття рішень та керування різними системами будинку. Центральна система може бути здійснена як окремий комп'ютер або сервер.

**Мережа зв'язку:** Це комунікаційна інфраструктура, що забезпечує зв'язок між різними компонентами системи розумного будинку. Вона може включати проводову (наприклад, Ethernet) та бездротову (наприклад, Wi-Fi, Bluetooth, Zigbee) мережі для передачі даних та команд.

**Системи управління:** Ці системи забезпечують керування різними аспектами будинку, такими як освітлення, опалення, кондиціонування повітря, безпека, аудіо-відео системи тощо. Вони взаємодіють з центральною системою для виконання команд і забезпечення належної реакції на зміни умов.[5]

**База даних:** Це централізоване сховище для зберігання даних, що стосуються розумного будинку. Вона може включати дані про налаштування та стан різних систем, зберігати історію подій, виконаних команд та рішень, а також дозволяти здійснювати пошук та аналіз даних для подальшої оптимізації роботи систем.

Користувацький інтерфейс: Це інтерфейс, який дозволяє користувачам керувати та моніторити різні системи розумного будинку. Він може бути реалізований у вигляді мобільних додатків, веб-інтерфейсів, пультів дистанційного керування та інших пристроїв.

Ця приблизна до розроблюваної архітектура графічно представлена у вигляді блок-схеми на рисунку, де кожен компонент представлений окремим блоком зі зв'язками між ними, які вказують на напрямок потоку даних та команд. Для покращення забезпечення масштабованості та ефективності розумного будинку, також доцільно розширити використання різними протоколами та стандартами, такими як MQTT, REST API, Wi-Fi, Bluetooth LE та іншими, для забезпечення співпраці між різними компонентами системи.



Рисунок 3.1 – архітектура розроблюваного об'єкта

### 3.3. Файлова структура об'єкту

Файлова структура проєкту розробки інформаційного забезпечення та баз даних розумного будинку включає наступні компоненти:

1. Коренева папка проекту: Це основна папка проекту, яка містить всі інші підпапки та файли. Вона може мати назву, що відображає назву проекту або ідентифікатор.
2. Документація: Ця папка містить всю документацію, пов'язану з проектом. Вона може включати такі файли, як специфікації вимог, технічні описи, діаграми архітектури, звіти про тестування та будь-яку іншу документацію, необхідну для розуміння та розвитку проекту.
3. Джерела коду: Ця папка містить всі файли, пов'язані з розробкою програмного забезпечення для розумного будинку. Вона може бути поділена на декілька підпапок, залежно від організації коду. Наприклад, можна мати підпапки для серверної частини, клієнтського додатку, модулів датчиків тощо. У цих папках будуть розташовані файли з розширенням, пов'язані з мовою програмування, наприклад, .py (Python), .java (Java), .js (JavaScript) тощо.
4. Бази даних: Ця папка містить файли, пов'язані з базами даних, використовуваними у розумному будинку. Якщо використовується реляційна база даних, можуть бути файли з розширенням .sql, які містять сценарії створення таблиць та запитів. Якщо використовується нереляційна база даних, така як MongoDB, то можуть бути файли конфігурації та збереження даних.
5. Ресурси: Ця папка містить різноманітні ресурси, необхідні для проекту. До неї можуть входити файли зображень, логотипів, шрифтів, аудіо- та відеофайлів, CSS-файли для оформлення веб-інтерфейсу та інші ресурси, які використовуються в процесі розробки.
6. Тестові файли: Ця папка містить файли, пов'язані з тестуванням розробленого програмного забезпечення. Вона може містити скрипти або файли конфігурації для автоматизованих тестів, зразки вхідних та очікуваних вихідних даних, звіти про тестування та інші матеріали, які допомагають забезпечити якість програмного продукту.
7. Додаткові папки: Залежно від потреб проекту, можуть бути додаткові папки для специфічних компонентів або модулів. Наприклад, папка для

інтеграції з іншими системами, папка для документації API, папка для статичних сторінок веб-сайту тощо.

Ця файлова структура дозволяє організувати проєкт у логічні групи файлів та директорій, забезпечуючи зручність розробки, управління та збереження даних. Важливо пам'ятати, що сама структура може варіюватися залежно від конкретних вимог та потреб проєкту, і можна адаптувати її відповідно до власних вимог і стандартів розробки.

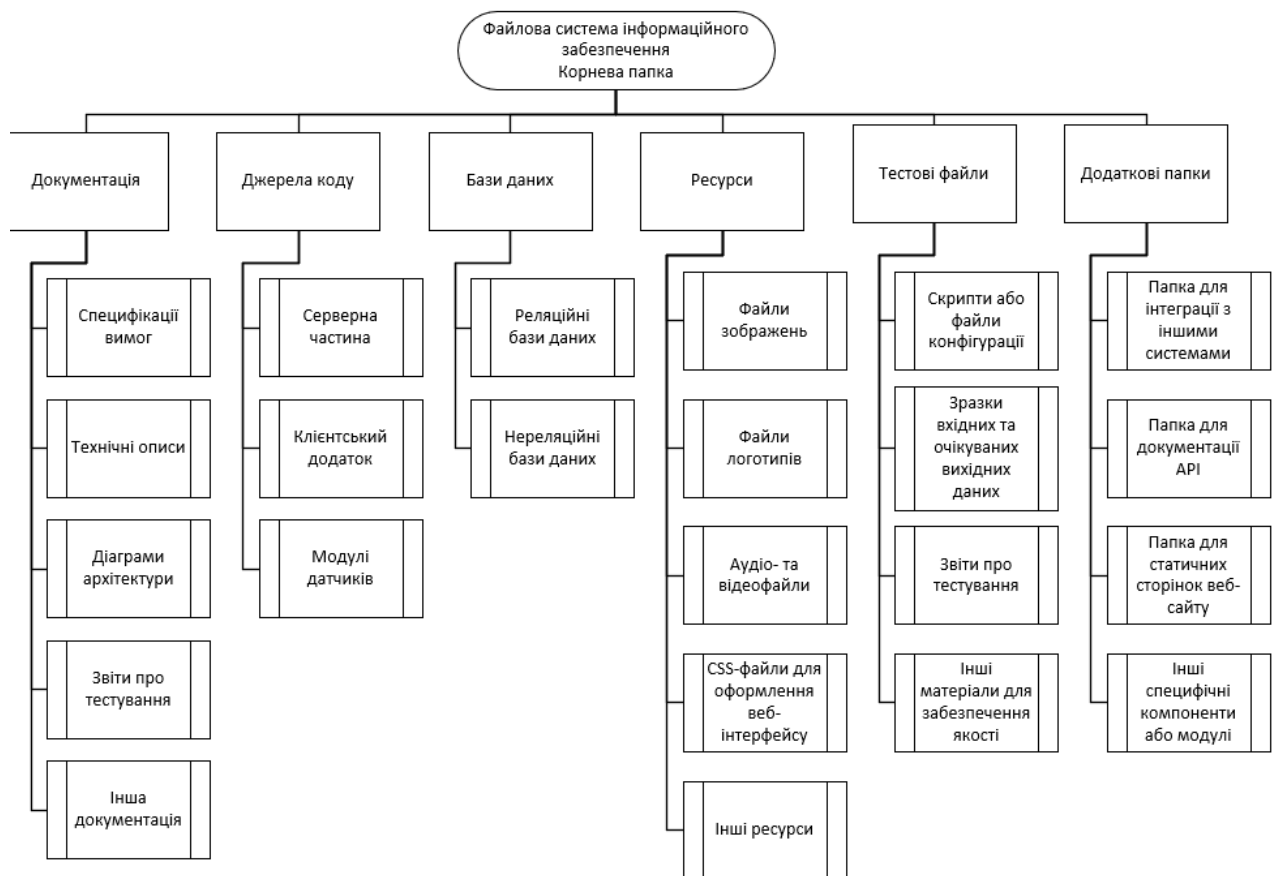


Рисунок 3.2 – Розширена файлова структура розроблюваного продукту

### 3.4. Розробка бази даних хабу та підключення до неї

Для створення бази даних розумного дому, який буде забезпечувати зберігання даних про користувачів, їх пристрої, дані від сенсорів, дії користувачів та налаштування пристроїв, потрібно спочатку створити базу даних та таблиці за допомогою мови SQL. Нижче наведено приклад створення таблиць за зазначеними у завданні вимогами:

```

CREATE TABLE Users (
    user_id INT PRIMARY KEY,
    name VARCHAR,
    email VARCHAR,
    password VARCHAR,
    accessLevel VARCHAR
);

CREATE TABLE Devices (
    device_id INT PRIMARY KEY,
    device_name VARCHAR,
    device_type VARCHAR
);

CREATE TABLE User_Devices (
    user_device_id INT PRIMARY KEY,
    user_id INT REFERENCES Users(user_id),
    device_id INT REFERENCES Devices(device_id)
);

CREATE TABLE Sensors (
    sensor_id INT PRIMARY KEY,
    sensor_name VARCHAR,
    sensor_type VARCHAR
);

CREATE TABLE Sensor_Data (
    sensor_data_id INT PRIMARY KEY,
    sensor_id INT REFERENCES Sensors(sensor_id),
    device_id INT REFERENCES Devices(device_id),
    user_id INT REFERENCES Users(user_id),
    data FLOAT,
    timestamp TIMESTAMP
);

CREATE TABLE User_Actions (
    action_id INT PRIMARY KEY,
    user_id INT REFERENCES Users(user_id),
    device_id INT REFERENCES Devices(device_id),
    action_type VARCHAR,
    timestamp TIMESTAMP
);

```

```

CREATE TABLE Device_Settings (
    setting_id INT PRIMARY KEY,
    device_id INT REFERENCES Devices(device_id),
    setting_type VARCHAR
);

```

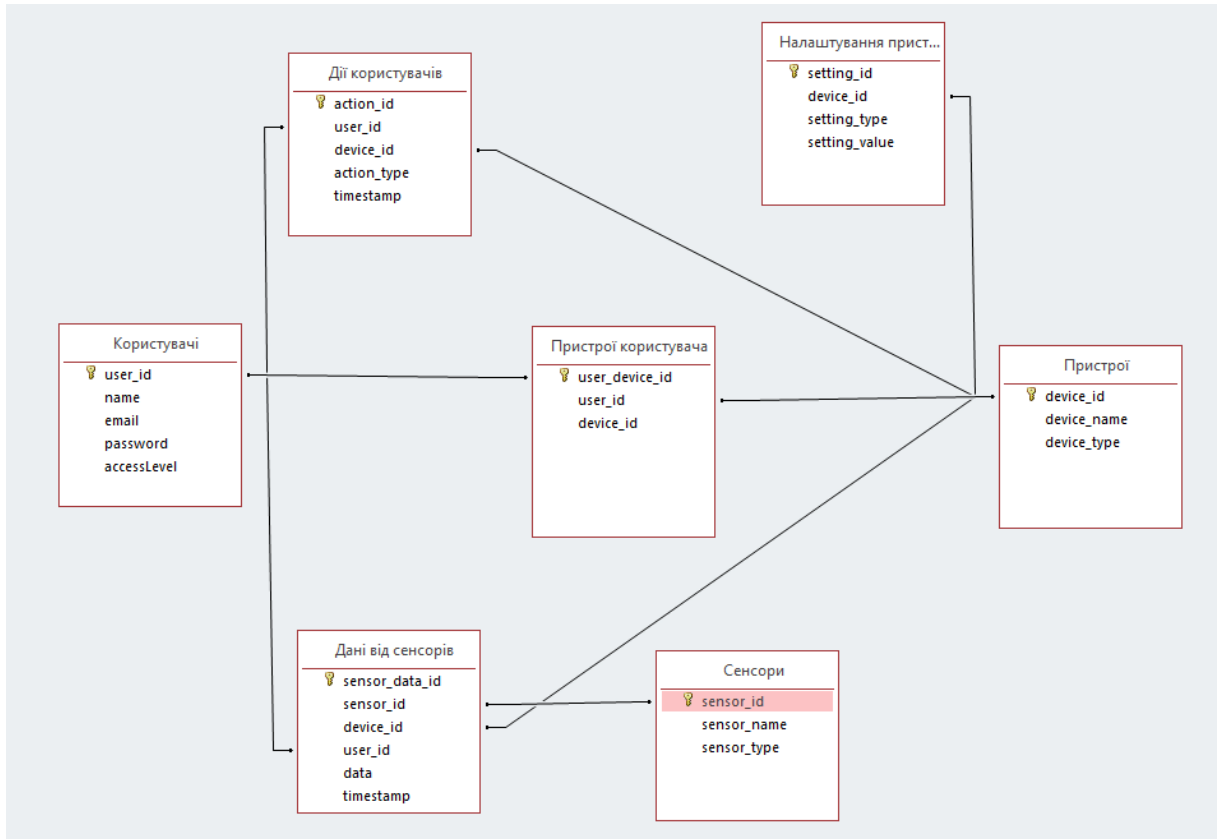


Рисунок 3.3 – Загальна схема роботи та зв'язків бд

Для підключення до бази даних через середовище Visual Studio та мову C# можна використати клас SqlConnection з простору імен System.Data.SqlClient. Нижче наведено приклад коду для підключення до бази даних :

```

using System.Data.SqlClient;
string connectionString = "Data
Source=<server_name>;Initial
Catalog=<database_name>;Integrated Security=True";
// або зазначте логін і пароль для підключення до бази
даних
// string connectionString = "Data
Source=<server_name>;Initial Catalog=<database_name>;User
ID=<user_id>;Password=<password>";

```

```

using (SqlConnection connection = new
SqlConnection(connectionString))
{
    try
    {
        connection.Open();
        Console.WriteLine("Connection Open!");
        // Ваш код для роботи з базою даних тут
    }
    catch (Exception ex)
    {
        Console.WriteLine("Can not open connection ! " +
ex.Message);
    }
}

```

У змінній `connectionString` необхідно замінити `<server_name>` на назву сервера, на якому розташована база даних, `<database_name>` – на назву бази даних. Якщо для підключення необхідний логін і пароль, то їх також необхідно зазначити в рядку підключення. Але у даній роботі використовується `localhost`, тож `127.0.0.1`. Зв'язок з базою даних відбувається через об'єкт класу `SqlConnection`. У прикладі коду є використання директиви `using` – вона гарантує, що ресурси підключення відключені після виконання коду в межах конструкції `using`. Всі запити до бази даних виконуються через цей об'єкт. У прикладі коду виконується спроба відкрити з'єднання з базою даних і, якщо вона успішно відкрилась, виконати деякі дії з базою даних. У випадку, якщо з'єднання не вдалося відкрити, буде виведено повідомлення про помилку на екран.

### 3.5. Методи роботи з базою даних хабу

Так, як основні методи роботи з майже любою базою даних це додавання, оновлення, пошук, видалення та отримання, то даний проєкт, також, має перелік цих функціональних об'єктів:

#### 1. Додавання запису в таблицю

```

        public void AddNewRecord(string name, int age, string
email)
        {
            using (SqlConnection connection = new
SqlConnection(connectionString))
            {
                string sql = "INSERT INTO Persons (Name, Age,
Email) VALUES (@Name, @Age, @Email)";
                using (SqlCommand command = new SqlCommand(sql,
connection))
                {
                    command.Parameters.AddWithValue("@Name",
name);

                    command.Parameters.AddWithValue("@Age", age);
                    command.Parameters.AddWithValue("@Email",
email);

                    connection.Open();
                    command.ExecuteNonQuery();
                }
            }
        }

```

## 2. Отримання всіх записів з таблиці

```

public List<Person> GetAllRecords()
{
    List<Person> persons = new List<Person>();
    using (SqlConnection connection = new
SqlConnection(connectionString))
    {
        string sql = "SELECT * FROM Persons";
        using (SqlCommand command = new SqlCommand(sql,
connection))
        {
            connection.Open();
            using (SqlDataReader reader =
command.ExecuteReader())
            {

```

```

        while (reader.Read())
        {
            persons.Add(new Person
            {
                Id = (int)reader["Id"],
                Name = reader["Name"].ToString(),
                Age = (int)reader["Age"],
                Email = reader["Email"].ToString()
            });
        }
    }
}
return persons;
}

```

### 3. Оновити запис в таблиці за ID

```

public void UpdateRecord(int id, string name, int age,
string email)
{
    using (SqlConnection connection = new
SqlConnection(connectionString))
    {
        string sql = "UPDATE Persons SET Name=@Name,
Age=@Age, Email=@Email WHERE Id=@Id";
        using (SqlCommand command = new SqlCommand(sql,
connection))
        {
            command.Parameters.AddWithValue("@Id", id);
            command.Parameters.AddWithValue("@Name",
name);
            command.Parameters.AddWithValue("@Age", age);
            command.Parameters.AddWithValue("@Email",
email);
            connection.Open();

```

```

        command.ExecuteNonQuery();
    }
}

```

#### 4. Видалити запис таблиці за ID

```

public void DeleteRecord(int id)
{
    using (SqlConnection connection = new
SqlConnection(connectionString))
    {
        string sql = "DELETE FROM Persons WHERE Id=@Id";
        using (SqlCommand command = new SqlCommand(sql,
connection))
        {
            command.Parameters.AddWithValue("@Id", id);
            connection.Open();
            command.ExecuteNonQuery();
        }
    }
}

```

#### 5. Знайти запис таблиці за критеріями(наприклад, всі особи старше 30 років, для встановлення спрощеного пакету оновлення)

```

public List<Person> FindRecordsByCriteria(int minAge)
{
    List<Person> persons = new List<Person>();
    using (SqlConnection connection = new
SqlConnection(connectionString))
    {
        string sql = "SELECT * FROM Persons WHERE
Age>=@MinAge";
        using (SqlCommand command = new SqlCommand(sql,
connection))
        {
            command.Parameters.AddWithValue("@MinAge",
minAge);
            connection.Open();

```

```

        using (SqlDataReader reader =
command.ExecuteReader())
    {
        while (reader.Read())
        {
            persons.Add(new Person
            {
                Id = (int)reader["Id"],
                Name = reader["Name"].ToString(),
                Age = (int)reader["Age"],
                Email = reader["Email"].ToString()
            });
        }
    }
}
return persons;

```

Тож, просто підсумовуючи, дані методи виконують наступні речі:

- Оновлення даних: створення методу, що дозволяє оновити існуючий запис у таблиці бази даних. Метод може отримувати в якості параметрів унікальний ідентифікатор запису та нові дані для оновлення. Після виконання метод повертає значення, що вказує на успішне або невдале оновлення даних.
- Видалення даних: створення методу, що дозволяє видалити існуючий запис з таблиці бази даних. Метод може отримувати в якості параметра унікальний ідентифікатор запису. Після виконання метод повертає значення, що вказує на успішне або невдале видалення даних.
- Пошук даних: створення методу, що дозволяє здійснити пошук записів у таблиці бази даних за заданими параметрами. Метод може отримувати в якості параметрів назву стовпців та значення, за якими потрібно здійснити пошук. Після виконання метод повертає список записів, що задовольняють критерії пошуку.

- Сортування даних: створення методу, що дозволяє сортувати записи у таблиці бази даних за певними критеріями. Метод може отримувати в якості параметрів назву стовпців та порядок сортування (зростання або спадання). Після виконання метод повертає список записів, відсортованих за вказаними критеріями.

- Групування даних: створення методу, що дозволяє групувати записи у таблиці бази даних за певними критеріями. Метод може отримувати в якості параметрів назву стовпців, за якими потрібно здійснювати групування та в подальшому відображення. Після виконання метод повертає групований список.

В подальшому дані методи виступають базисом для розвинутих методів, які вузько спеціалізуються завдяки підключенню предметної області. До прикладу:

- Метод для управління клімат–контролем: дозволяє встановлювати температуру та вологість повітря в будинку.

- Метод для управління системою безпеки: дозволяє включати/вимикати систему відеоспостереження, встановлювати паролі для доступу до різних зон будинку, включати/вимикати сигналізацію та інше.

- Метод для управління освітленням: дозволяє включати/вимикати світло в будинку, регулювати яскравість та кольорову температуру освітлення, налаштовувати різні сценарії освітлення та інше.

- Метод для управління побутовими приладами: дозволяє включати/вимикати побутові прилади (телевізор, пральну машину, холодильник та інше) з дистанції, налаштовувати різні режими роботи та інше.

- Метод для управління системою звуку: дозволяє включати/вимикати аудіосистему, регулювати гучність, налаштовувати різні звукові ефекти та інше.

Та інші методи напряму зв'язані з базою даних та пристроями розумного дому, тощо.

### **3.6. Створення та програмування інформаційного забезпечення датчиків «розумного» дому**

#### **3.6.1. Датчик відкриття та закриття дверей, вікон, шафи, тощо.**

SQL-запит для створення бази даних датчика відкриття/закриття:

```
CREATE DATABASE door_sensor;
USE door_sensor;

CREATE TABLE door_events (
    id INT NOT NULL AUTO_INCREMENT,
    event_type VARCHAR(10) NOT NULL,
    status VARCHAR(35) NOT NULL,
    timestamp      TIMESTAMP      NOT      NULL      DEFAULT
CURRENT_TIMESTAMP,
    PRIMARY KEY (id)
);
```

Ця база даних містить таблицю door\_events з наступними полями:

- id – унікальний ідентифікатор події
- event\_type – тип події (open або close)
- timestamp – час, коли сталася подія

Найпростіші методи роботи з базою даних та пристроєм:

```
// Метод додавання нового запису про відкриття/закриття дверей
public static void AddDoorStatus(bool isOpen)
{
    string status = isOpen ? "відкрито" : "закрито";
    string query = $"INSERT INTO DoorStatus (Status,
TimeStamp) VALUES ('{status}', '{DateTime.Now.ToString("yyyy-
MM-dd HH:mm:ss")}');";
```

```

        using (SqlConnection connection = new
SqlConnection(connectionString))
        using (SqlCommand command = new SqlCommand(query,
connection))
        {
            connection.Open();
            command.ExecuteNonQuery();
        }
    }
    // Метод для отримання статистики відкриття/закриття
дверей за останню годину
    public static Dictionary<string, int>
GetDoorStatsLastHour()
    {
        DateTime oneHourAgo = DateTime.Now.AddHours(-1);
        string query = $"SELECT Status, COUNT(*) FROM
DoorStatus WHERE TimeStamp >= '{oneHourAgo.ToString("yyyy-MM-
dd HH:mm:ss")}' GROUP BY Status;";
        Dictionary<string, int> stats = new Dictionary<string,
int>();
        using (SqlConnection connection = new
SqlConnection(connectionString))
        using (SqlCommand command = new SqlCommand(query,
connection))
        {
            connection.Open();
            SqlDataReader reader = command.ExecuteReader();
            while (reader.Read())
            {
                stats.Add(reader.GetString(0),
reader.GetInt32(1));
            }
        }
        return stats;
    }
}

```

### ***3.6.2. Датчик розумного термостату***

**SQL–запит для створення бази даних датчику розумного термостату:**

```
CREATE TABLE temperature_data (  
    id INT NOT NULL PRIMARY KEY IDENTITY(1,1),  
    temperature FLOAT NOT NULL,  
    timestamp DATETIME NOT NULL  
);
```

**Методів, які використовуються з базою даних розумного термостату в хабі розумного будинку:**

```
// Додати нове значення температури до бази даних  
public void AddTemperature(float temperature)  
{  
    string query = "INSERT INTO temperature_data  
(temperature, timestamp) VALUES (@temperature, @timestamp)";  
    using (SqlConnection connection = new  
SqlConnection(connectionString))  
        using (SqlCommand command = new SqlCommand(query,  
connection))  
        {  
command.Parameters.AddWithValue("@temperature", temperature);  
        command.Parameters.AddWithValue("@timestamp",  
DateTime.Now);  
        connection.Open();  
        command.ExecuteNonQuery();  
        }  
    }  
  
// Отримати середню температуру за останню годину  
public float GetAverageTemperatureLastHour()  
{  
    string query = "SELECT AVG(temperature) FROM  
temperature_data WHERE timestamp >= DATEADD(hour, -1,  
GETDATE())";  
    using (SqlConnection connection = new  
SqlConnection(connectionString))
```

```

        using (SqlCommand command = new SqlCommand(query,
connection))
    {
        connection.Open();
        object result = command.ExecuteScalar();
        if (result != null && result != DBNull.Value)
        {
            return (float)result;
        }
        else
        {
            return 0f;
        }
    }
}

```

// Отримати максимальну температуру за певний період часу

```

public float GetMaxTemperature(DateTime startDate,
DateTime endDate)
{
    string query = "SELECT MAX(temperature) FROM
temperature_data WHERE timestamp BETWEEN @startDate AND
@endDate";

    using (SqlConnection connection = new
SqlConnection(connectionString))
        using (SqlCommand command = new SqlCommand(query,
connection))
    {
        command.Parameters.AddWithValue("@startDate",
startDate);
        command.Parameters.AddWithValue("@endDate",
endDate);

        connection.Open();
        object result = command.ExecuteScalar();
        if (result != null && result != DBNull.Value)
        {

```

```
        return (float)result;
    }
    else
    {
        return 0f;
    }
}
```

Ці методи дозволяють додавати нові значення температури до бази даних, отримувати середню температуру за останню годину та максимальну температуру за певний період часу. Для виконання операцій з базою даних використовується клас `SqlConnection`, який встановлює з'єднання з базою даних, і клас `SqlCommand`, який виконує запити до бази даних. Для передачі параметрів в запити використовується метод `AddWithValue`.

### **3.7. Програмування інформаційного забезпечення та баз даних інших пристроїв та датчиків розумного будинку**

Програмний код з частиною опису перенесений та винесений в додатках. Наступні прилади описуються в них:

- «Розумні» системи освітлення,
- «Розумні» розетки, та контроль споживання електроенергії,
- «Розумні» аварійні системи.

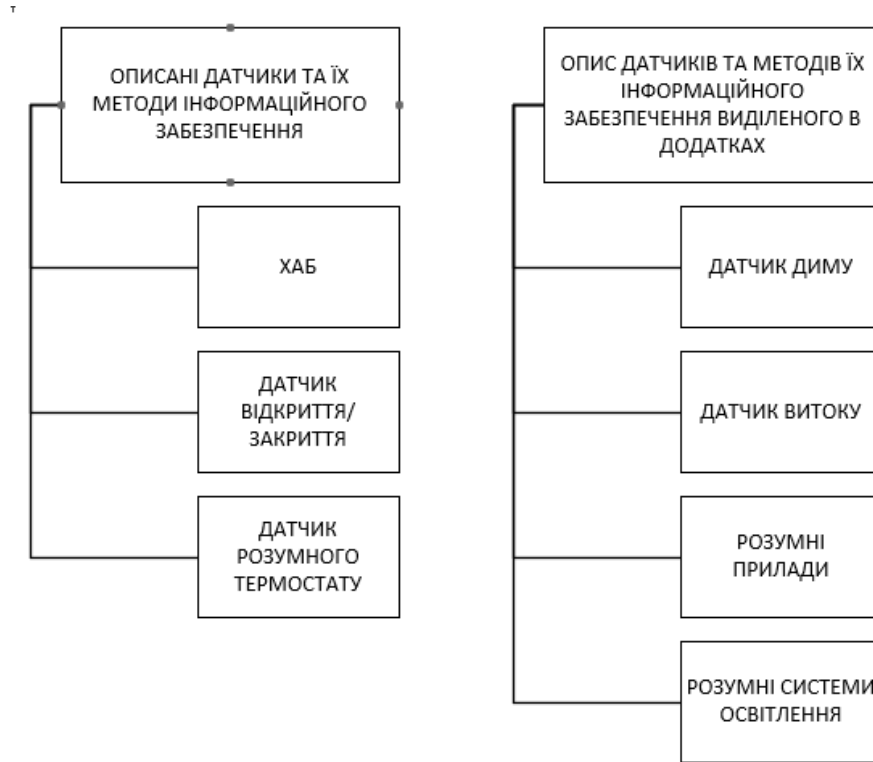


Рисунок 3.4 – висвітлені та виведені в опис та додатки модулі «розумного дому»

## 4. БІЗНЕС ПЛАН

### 4.1. Впровадження продукту

Впровадження продукту дипломної роботи "Розробка інформаційного забезпечення і баз даних розумного будинку" має наступний теоретичний план впровадження:

1. Аналіз потреб і можливостей. В роботі проведений необхідний аналіз ринку розумних будинків, визначені конкурентні переваги та недоліки інших продуктів, визначені потенційні клієнти і їх потреби, а також створені можливі відгуки про можливості власного продукту.

2. Розробка продукту. В процесі написання за мету взяте розроблення інформаційного забезпечення для розумного будинку, бази даних для зберігання інформації про стан сенсорів та його систем, інтерфейсу користувача та інших необхідних елементів.

3. Тестування. Після розробки продукту необхідним кроком стало проведення його тестування на різних етапах розробки, задля виявлення та виправлення можливих помилок та проблем.

4. Маркетингові дослідження. Необхідність вивчення потенційних клієнтів, їх потреб та вимог, проведення маркетингових досліджень, для визначення, які функції та можливості можуть бути додані до продукту, щоб він був привабливішим для потенційних клієнтів.

5. Випробування в реальних умовах. Після успішного тестування і маркетингових досліджень, теорія роботи продукту в реальних умовах та в різних будинках, була не фізично протестована, в плані співвідношення розробленого інформаційного середовища з існуючими, реально працюючими пристроями та реалізаціями, щоб переконатися в його працездатності та досконалості.

6. Впровадження. Після успішного тестування та випробування, в хід пішов розділ «бізнес плану» для опису питання «впровадження продукту на ринку». Для цього були використані різні теоретичні методи продажу,

такі як продажі через Інтернет і продажі через різні торгові мережі, а також укладання угод з партнерами та постачальниками для забезпечення надійного постачання та підтримки продукту.

7. Підтримка та поновлення. Впровадження продукту створило необхідність забезпечення оновлення та підтримки. Це включає в себе надання технічної підтримки користувачам, виправлення помилок, додавання нових функцій та можливостей, а також регулярне випускання оновлень програмного забезпечення.

8. Моніторинг та аналіз. При повній інтеграції продукту в життя, слід проводити постійний моніторинг та аналіз роботи продукту, щоб виявляти можливі проблеми та недоліки, а також вдосконалювати його для кращого задоволення потреб користувачів.

9. Розвиток продукту. Останнім етапом впровадження є сувора необхідність в постійній роботі над розвитком продукту, щоб він відповідав новим технологічним вимогам та потребам користувачів. Це включає в себе випуск нових версій програмного забезпечення, додавання нових можливостей, а також розширення лінійки продуктів для відповіді на різні потреби користувачів.

#### ***4.1.1. Визначні риси по відношенню до продукту конкурентів***

Залежно від конкретної ситуації розвитку, визначено, що можуть використовуватись різні стратегії для залучення фінансування і реалізації проекту. Наприклад, можна залучити інвестиції від бізнес-ангелів, венчурних фондів або корпорацій. Іншим варіантом може бути залучення кредиту від банку або користування грантами та програмами державної підтримки для стартапів.

Процес впровадження продукту містить в собі різні етапи, починаючи від розробки концепції та прототипування, і закінчуючи випуском та підтримкою продукту. Під час цього процесу важливо забезпечувати якість розробки, оцінювати ризики та проводити тестування

продукту, а також залучати фахівців з різних галузей для розробки різних компонентів продукту, таких як інтерфейс, база даних, логіка програмування тощо.

Після того, як продукт буде випущений на ринок, важливо буде забезпечити його підтримку та розвиток. Це може включати в себе надання технічної підтримки користувачам, проведення оновлень та виправлення помилок, а також розробку нових функцій та можливостей для забезпечення конкурентоспроможності продукту на ринку. Крім того, важливо мати ефективну стратегію монетизації продукту, яка дозволить забезпечити прибуток від його реалізації.

Можливо також видати продукт як безкоштовний, і підтримка його може здійснюватися небайдужою спільнотою, наприклад, відкритою спільнотою розробників або ентузіастів, які підтримують продукт та допомагають в розвитку і підтримці. Це може бути вигідно для проектів з відкритим кодом, де ком'юніті може приймати участь у розробці продукту, додаванні нових функцій та покращенні його функціональності. Така спільнота може також надавати технічну підтримку користувачам продукту, вирішувати проблеми, які з'являються, та вносити зміни, щоб забезпечити кращу функціональність та якість продукту.

Однак, важливо зазначити, що залежно від конкретної ситуації, видання продукту як безкоштовного може мати свої недоліки, зокрема, недостатність фінансування для розвитку та підтримки продукту, обмежена кількість функцій та можливостей, а також менша стабільність продукту. Тому, перед тим, як приймати рішення про такий варіант реалізації продукту, вирішено ретельно зважити всі переваги та недоліки цих підходу.

#### ***4.1.2. Необхідність продукту споживачу***

Залежно від того, який саме продукт конкурентів порівнюється з продуктом розроблюваної дипломної роботи, виділено різнобічні визначні риси.

Однією з проаналізованої визначної риси продукту дипломної роботи стала його інноваційність, яка полягає в тому, що він може включати у себе новітні технології, які ще не застосовуються конкурентами. Наприклад, це може бути використання штучного інтелекту, інтернету речей, блокчейн–технологій та інших.

Іншою визначною рисою стала більш висока продуктивність та швидкодія продукту порівняно з конкурентами, що забезпечує більш високу ефективність та зручність його використання.

Також важливим став підхід до дизайну та інтерфейсу користувача. Якщо продукт дипломної роботи має більш зручний та простий інтерфейс порівняно з конкурентами, то це стає важливим кроком до привернення більшої кількості користувачів та забезпечення більшого успіху.

Визначними рисами продукту також стали його можливості та особливості функціональності. Якщо продукт дипломної роботи має більше можливостей та функцій, ніж конкуренти, то він буде більш привабливим для користувачів.

Загалом, дані риси продукту дипломної роботи можуть включати у себе як технічні показники, так і підхід до дизайну та інтерфейсу користувача, а також функціональність та можливості продукту.

#### ***4.1.3. Очікувані основні фінансові результати***

При розробці інформаційного забезпечення даної теми, вирішено врахувати такі фактори, як витрати на розробку програмного забезпечення, інженерні рішення, рекламні витрати та інші витрати на підтримку бізнесу.

а) Прогнозні об'єми продажу на найближчі роки залежать від багатьох факторів, таких як ринкова конкуренція, зміна технологічних стандартів, та рекламні витрати. При оцінці можливих об'ємів продажів

були врахувані ці фактори та встановлена приблизна кількість продукту, що може бути продана на ринку. Наприклад, протягом першого року може бути продано 1000 пристроїв, протягом другого року – 3000 пристроїв, а протягом третього року – 5000 пристроїв.

б) Прибуток від продажів можна визначити шляхом множення прогнозованої кількості продукту на його ціну. Якщо, наприклад, ціна продажу одного пристрою складає 1000 доларів, то прибуток від продажу протягом першого року становитиме 1 мільйон доларів, протягом другого року – 3 мільйони доларів, а протягом третього року – 5 мільйонів доларів.

в) Витрати на виробництво включають витрати на матеріали, заробітну плату працівників, оренду приміщення та інші витрати, пов'язані з виробництвом. Вартість виробництва одного пристрою складе, приблизно, 500 доларів. Тоді загальні витрати на виробництво протягом першого року становитимуть 500 тисяч доларів, протягом другого року – 1,5 мільйона доларів, а протягом третього року – 2,5 мільйона доларів, як описують статистичні форуми та інформаційні збірки.

г) Валовий прибуток можна визначити шляхом віднімання витрат на виробництво від прибутку від продажу. Наприклад, якщо вартість виробництва одного пристрою складає 500 доларів, а ціна продажу становить 1000 доларів, то валовий прибуток від продажу одного пристрою становитиме 500 доларів. Тоді загальний валовий прибуток протягом першого року становитиме 500 тисяч доларів, протягом другого року – 1,5 мільйона доларів, а протягом третього року – 2,5 мільйона доларів. Рівень прибутковості вкладень в майбутню справу можна визначити, розділивши валовий прибуток на витрати на розробку та виробництво. Наприклад, якщо загальні витрати на розробку та виробництво становлять 2 мільйона доларів, а загальний валовий прибуток за три роки – 9 мільйонів доларів, то рівень прибутковості становитиме 4,5.

д) Термін повернення банківського кредиту може бути розрахований шляхом визначення чистого прибутку, який отримає підприємство протягом кожного року, та порівняння його з розміром банківського

кредиту та процентною ставкою. Наприклад, якщо розмір кредиту складає 1 мільйон доларів на три роки з процентною ставкою 5%, то загальна сума повернення кредиту складатиме 1,15 мільйона доларів. Якщо чистий прибуток за перший рік складе 500 тисяч доларів, то підприємство зможе повернути кредит впродовж двох років. Однак, якщо чистий прибуток за перший рік складе менше ніж 115 тисяч доларів, то підприємство не зможе повернути кредит за три роки.

Отже, для успішної реалізації проекту з розробки інформаційного забезпечення та баз даних розумного будинку потрібно враховувати очікувані об'єми продажу на найближчі три роки, прибуток від продажу, витрати на виробництво, валовий прибуток, рівень прибутковості та термін повернення банківського кредиту. Такий аналіз допоможе оцінити фінансові ризики та прийняти обґрунтовані рішення щодо реалізації проекту.

## **4.2. Проектований продукт, або вид послуг**

### ***4.2.1. Потреби продукту по відношенню до проекту***

Продукт проекту розробки інформаційного забезпечення та баз даних розумного будинку повинен задовольнити наступні речі:

1. Керування різними системами розумного будинку: освітленням, опаленням, вентиляцією, кондиціонуванням повітря, безпекою, медіа–системами та іншими.

2. Оптимізація витрат енергії та забезпечення збереження ресурсів, зокрема за рахунок автоматичного регулювання температури, освітлення та інших систем.

3. Моніторинг різних параметрів, таких як температура, вологість, рівень шуму та інших факторів, що дозволяє вчасно реагувати на небезпеку та запобігати нещасним випадкам.

4. Забезпечення віддаленого керування та моніторингу системами розумного будинку з будь-якої точки світу та зручного доступу до баз даних.

5. Забезпечення безпеки будинку та його мешканців шляхом виявлення та попередження потенційних загроз, таких як пожежі, протікання води, вторгнення в будинок та інші.

6. Забезпечення зручного та простого використання систем розумного будинку для мешканців будинку, в тому числі за допомогою мобільного додатку чи голосових команд.

7. Забезпечення максимальної стабільності та безперебійної роботи інформаційного забезпечення систем.

8. Забезпечення можливості розширення та модернізації IoT з метою вдосконалення їх функціональності.

#### ***4.2.2. Особливості і відмінні риси розроблюваного проєкту, від інших реалізацій.***

Особливості та відмінні риси, які дозволять віддати перевагу даному продукту проєкту над продуктами конкурентів, можуть бути наступні:

1. Інноваційність: продукт проєкту може бути більш інноваційним, ніж продукти конкурентів. Наприклад, він може містити нові технології, які не маються в інших продуктах, або мати унікальний дизайн.

2. Широкий функціонал: продукт проєкту може мати більший функціонал, ніж продукти конкурентів, що дозволить задовольнити більше потреб користувачів та надати їм більше можливостей.

3. Відкритість та гнучкість: продукт проєкту може бути більш відкритим та гнучким, ніж продукти конкурентів, що дозволить користувачам встановлювати його на будь-які пристрої та модифікувати його під свої потреби.

4. Висока якість та надійність: продукт проекту може мати вищу якість та надійність, ніж продукти конкурентів, що дозволить користувачам бути впевненими в його роботі та зменшить ймовірність поломок.
5. Зручність в користуванні: продукт проекту може бути більш зручним в користуванні, ніж продукти конкурентів, що зменшить час та зусилля, необхідні для налагодження та використання системи розумного будинку.
6. Підтримка та обслуговування: продукт проекту може мати кращу підтримку та обслуговування, ніж продукти конкурентів, що забезпечить користувачам швидке та якісне вирішення проблем.
7. Ціна: продукт проекту може мати більш конкурентоспроможну ціну, ніж продукти конкурентів

#### ***4.2.3. Патенти, авторські свідоцтва та наочне зображення розроблюваного проекту***

До моменту розробки продукту проекту не було поданих патентів або авторських свідоцтв, проте можливе подання заявки на їх отримання.

Існують наочні зображення продукту проекту, такі як схеми підключення та роботи, діаграми та інші візуалізації.

#### ***4.2.4. Попередня оцінка реалізації ціни реалізації продукту***

Попередня оцінка залежить від багатьох факторів, таких як складність проекту, трудомісткість його розробки та реалізації, кількість спеціалістів, які займатимуться розробкою та інші. Вартість витрат на виробництво продукту також залежить від багатьох чинників, таких як вартість компонентів, витрат на розробку та виробництво прототипу, витрат на тестування тощо. Так як проєкт виконано, суто в наукових та теоретичних рамках, точної ціни реалізації не буде наведено. Але по ходу просування розділу є певні приклади.

#### ***4.2.5. Очікувана величина прибутку***

Очікувана величина прибутку від продажу продукту проекту буде становити від 10% до 30% від загального обсягу продажів. Інформація отримана за допомогою інтернет статистичних сервісів, зокрема Statista.

#### ***4.2.6. Характеристика якісних показників продукту проекту***

1. Забезпечення максимальної ефективності та комфорту життя в розумному будинку;
2. Можливість моніторингу та управління всіма інформаційними системами будинку через один інтерфейс;
3. Висока надійність та безпека функціонування;
4. Можливість налаштування систем будинку з урахуванням особистих потреб користувача;
5. Додаткові функції, які полегшують життя, такі як підтримка здорового способу життя

#### ***4.2.7. Організація сервісу продукту проекту***

Відповідно до умов проекту, вирішено внести необхідність передбачення організації технічної підтримки та обслуговування продукту після його запуску на ринок. Для цього можуть бути створені спеціалізовані сервісні центри або забезпечена дистанційна підтримка.

### **4.3. Конкуренція**

#### ***4.3.1. Найбільші виробники аналогічного продукту.***

За даними дослідницької компанії Statista, найбільшими виробниками розумних систем для будинку є компанії:

- Samsung SmartThings
- Google Nest
- Amazon Alexa
- Apple HomeKit
- Philips Hue

Ці компанії є лідерами в своїй галузі та мають великий вплив на ринок розумних систем для будинку. Однак, це не означає, що вони єдині на ринку, адже існує багато інших компаній, які також виробляють розумні системи для будинку.

#### ***4.3.2. Фінансово–аналітичний аналіз роботи компаній***

За даними Statista, світовий ринок розумних систем для будинку продовжує зростати. У 2021 році він досяг значення понад 95 мільярдів доларів США, а до 2025 року очікується зростання до 174 мільярдів доларів США.

Що стосується окремих компаній, то багато залежить від їхньої стратегії та ринкової ситуації. Наприклад, компанії Samsung SmartThings, Google Nest, Amazon Alexa, Apple HomeKit та Philips Hue є провідними гравцями на ринку, їх продукти мають велику популярність і підтримку від споживачів. Вони також мають великі ресурси для проведення рекламних кампаній та впровадження нових моделей.

Інші компанії можуть мати меншу популярність та менші фінансові ресурси, що може впливати на їхні доходи та об'єм продажів. Проте, зростання ринку в цілому може допомогти їм збільшити свою частку на ньому.

Технічний сервіс також грає важливу роль у розумних системах для будинку, оскільки користувачі можуть звертатися до компаній з

проблемами та запитами на налагодження роботи системи. У цьому сенсі компанії, які мають відповідні ресурси для надання якісного сервісу, можуть здобути перевагу на ринку.

#### **4.3.3. Сутність предмету конкуренту**

1. Наявність датчиків, які забезпечують збір даних про різні параметри в будинку (наприклад, температуру, вологість повітря, освітленість тощо).
2. Можливість керування різними пристроями та системами в будинку, включаючи системи опалення, кондиціонування, безпеки, освітлення, базами даних, тощо.
3. Можливість керування різними пристроями та системами з допомогою мобільних додатків або голосових помічників, таких як Siri або Google Assistant.
4. Наявність алгоритмів, які дозволяють забезпечити автоматичне керування різними системами в будинку на основі зібраних даних (наприклад, автоматичне включення опалення або кондиціонування, коли температура опускається нижче або піднімається від заданого рівня).
5. Інтеграція з іншими "розумними" пристроями, такими як "розумні" телевізори, пристрої для стрімінгу музики або "розумні" дверні замки, тощо.

Рівень якості продукту конкурентів може значно відрізнятись в залежності від виробника та його цінової політики. Дизайн продукту також може бути різним в залежності від виробника та його цільової аудиторії.

Думка зацікавлених сторін щодо продукту конкурентів може бути різною. Наприклад, клієнти можуть бути задоволені функціональністю та якістю продукту, тоді як експерти можуть мати свої відгуки.

Крім того, важливим фактором успіху конкурентів є їх репутація на ринку та наявність задоволених клієнтів. Це може мати вплив на рівень лояльності споживачів і може бути перевагою в здобутті нових клієнтів. Також важливим фактором є наявність продуктових ліній та додаткових

послуг, які можуть бути пропоновані разом з продуктом, наприклад, технічна підтримка, гарантійний термін, сервісні послуги тощо.

Загалом, конкуренти є серйозним викликом для підприємства, що розробляє продукт проекту, тому важливо враховувати їх характеристики та дії при формуванні стратегії розвитку і маркетингу продукту.

#### ***4.3.4. Конкурентний рівень цін***

За даними досліджень роботи, ціни на продукти конкурентів значно відрізняються в залежності від характеристик, функцій та рівня якості. Загалом, більш високоякісні та функціональні продукти мають вищі ціни, а менш просунуті та менш функціональні продукти можуть мати більш доступні ціни.

Проте, конкуренція на ринку розумних будинків є доволі високою, тому деякі компанії можуть застосовувати різні стратегії ціноутворення, такі як знижки на певний період часу або пільгові умови при купівлі декількох продуктів. Також важливо зазначити, що ціни на продукти конкурентів можуть змінюватися з часом через різні фактори, такі як зміни витрат на виробництво, зміна попиту на ринку та інші чинники.

У будь-якому випадку, для того, щоб визначити оптимальну ціну на продукт проекту, був проведений детальний аналіз ринку, конкурентів та попиту споживачів, який є у відкритому доступі.

Нижче наведено, приклад, загальних середніх цін на деякі пристрої розумного дому у 2021 році:

- Розумні датчики руху: від \$20 до \$50
- Розумні термостати: від \$100 до \$250
- Розумні замки для дверей: від \$150 до \$300
- Розумні дверні дзвінки: від \$100 до \$250
- Розумні розетки та вимикачі: від \$20 до \$60
- Розумні камери спостереження: від \$50 до \$200
- Розумні димові та вуглекислотні детектори: від \$50 до \$150

#### **4.4. План виробництва**

#### ***4.4.1. Місце виробництва товарів на діючому підприємстві.***

Описуючи розробку інформаційного забезпечення розумного дому, місце виробництва товарів може бути важливим питанням. Оскільки інформаційне забезпечення розумного дому включає в себе різні пристрої, датчики та інші компоненти, які потрібно виробляти та збирати в одне ціле.

При розробці інформаційного забезпечення розумного дому, підприємству доцільно створити власну лінію виробництва, де вони будуть виготовляти всі компоненти та інформаційні оболочки, такі як бази даних, необхідні для створення розумного дому. Це може бути вигідно, оскільки це дозволить підприємству контролювати якість продукту та знизити витрати на логістику.

Однак, якщо підприємство не має достатньої кількості ресурсів та експертизи для виробництва всіх необхідних компонентів, то вони можуть вирішити купувати готові компоненти від інших виробників. Це може забезпечити швидке введення продукту на ринок та знизити витрати на виробництво.

Отже, вирішення питання про місце виробництва товарів на діючому або на знову створеному підприємстві залежить від конкретних умов та можливостей підприємства, яке займається розробкою інформаційного забезпечення розумного дому.

#### ***4.4.2. Необхідність виробничих потужностей***

Якщо підприємство, що займається розробкою інформаційного забезпечення розумного дому, вирішило створити власну лінію виробництва, то необхідність у виробничих потужностях буде залежати від обсягу виробництва та кількості компонентів, які планується виробляти.

Залежно від обсягу виробництва, може бути необхідно нарощувати виробничі потужності з року в рік. Це може означати будівництво нових

приміщень, розширення існуючих ліній виробництва або придбання нового обладнання.

Нарощування виробничих потужностей потребує значних інвестицій, тому необхідно зробити ретельний аналіз ринку та попиту на продукцію, перш ніж приймати рішення про розширення виробництва.

Якщо підприємство вирішило купувати готові компоненти від інших виробників, то необхідність у виробничих потужностях може бути меншою. Однак, підприємство все одно може потребувати складських приміщень для зберігання та перевезення компонентів.

Отже, необхідність у виробничих потужностях та їх нарощування залежатиме від конкретної стратегії виробництва, яку вибере підприємство.

#### ***4.4.3. Умови закупки сировини. Репутація та досвід постачальника.***

Для розробки інформаційного забезпечення розумного дому, підприємство може закупляти сировину та комплектуючі у різних постачальників на різних умовах.

Для вибору постачальників, підприємство може проводити дослідження ринку, знаходити відгуки клієнтів та аналізувати репутацію потенційних постачальників. Також можна враховувати рекомендації від інших підприємств, які вже мають досвід співпраці з цими постачальниками.

Для підприємства важливо враховувати такі параметри, як ціна, якість та доставку сировини та комплектуючих. Також не менш важливою є репутація постачальника та його досвід роботи з іншими підприємствами. Ці параметри можуть впливати на якість продукції, її вартість та терміни виконання замовлень.

Підприємство може вести документацію щодо співпраці з постачальниками, таку як контракти, угоди, звіти тощо. Це дозволить вести облік витрат та контролювати якість продукції.

Отже, вибір постачальників сировини та комплектуючих є важливим етапом розробки продукту, тому необхідно провести ретельний аналіз та дослідження ринку, щоб знайти надійних постачальників з хорошою репутацією та досвідом роботи з іншими підприємствами.

#### ***4.4.4. Виробнича кооперація.***

Так, дослідження описує виробничу кооперацію з різними виробниками компонентів та систем для розумного будинку, такими як виробники систем автоматизації освітлення, систем опалення, систем безпеки, систем контролю енергоспоживання та іншими. Також можлива співпраця з інформаційними технологічними компаніями для розробки інформаційного забезпечення для керування системою розумного будинку.

Отже, розробнику та стартаперу необхідно формулювати трактування питання іншим чином, саме: "Які компанії можуть бути потенційними партнерами для виробничої кооперації з метою розробки інформаційного забезпечення систем розумного дому?"

#### ***4.4.5. Лімітація об'єктів підприємства***

Так, лімітація об'ємів виробництва або поставок ресурсів можлива залежно від різних факторів, таких як:

1. Обмеження ресурсів: Якщо певний ресурс (наприклад, деталі, електронні компоненти тощо) стає обмеженим або складнодоступним, це може призвести до лімітації об'ємів виробництва. В такому випадку виробник може обмежити обсяги виробництва або поставок на певний період часу.
2. Фінансові обмеження: Недостатність фінансування може обмежити обсяги виробництва або поставок ресурсів. Якщо виробник не може знайти досить фінансування для розширення виробництва або

закупівлі необхідних ресурсів, то це може призвести до лімітації обсягів виробництва.

3. Ринкові умови: Зміна попиту на продукцію або послуги може призвести до лімітації об'ємів виробництва або поставок ресурсів. Якщо попит на певний продукт зменшується, то виробник може обмежити виробництво цього продукту або поставки його компонентів.

Тому можливість лімітації об'ємів виробництва або поставок ресурсів залежить від конкретної ситуації та обставин.

#### ***4.4.6. Проблема устаткування.***

Устаткування, необхідне для розробки інформаційного забезпечення систем розумного дому, включає наступні компоненти:

- Комп'ютери та периферійні пристрої для розробки програмного забезпечення.
- Мережеве обладнання для забезпечення зв'язку між різними компонентами системи розумного будинку.
- Сенсори та пристрої збору даних для моніторингу температури, вологості, освітлення, руху та інших параметрів в різних зонах будинку.
- Програмне забезпечення для аналізу даних та керування системою розумного будинку та баз даних.

Для придбання устаткування можна звернутися до різних постачальників техніки та електроніки. Наприклад, до магазинів комп'ютерної техніки, до спеціалізованих постачальників систем автоматизації освітлення та опалення, а також до постачальників мережевого обладнання. Також можна звернутися до виробників сенсорів та пристроїв збору даних та програмного забезпечення для систем розумного будинку. Для забезпечення якості та надійності системи

розумного будинку, слід віддавати перевагу високоякісному устаткуванню від довірених постачальників.

#### ***4.4.7. Схема виробничих потоків.***

На схемі виробничих потоків підприємства можна виділити наступні етапи:

1. Доставка сировини та комплектуючих виробів зі складів постачальників.
2. Приймання та огляд сировини та комплектуючих виробів на приймальному складі.
3. Підготовка сировини та комплектуючих виробів до переробки відповідно до технологічних вимог.
4. Переробка сировини та комплектуючих виробів на відповідних цехах згідно з технологічним процесом.
5. Контроль якості продукції на кожному етапі виробничого процесу.
6. Налаштування інформаційного забезпечення приладу.
7. Доставка готової продукції на склад готової продукції та звідти на склади клієнтів або безпосередньо до клієнтів.

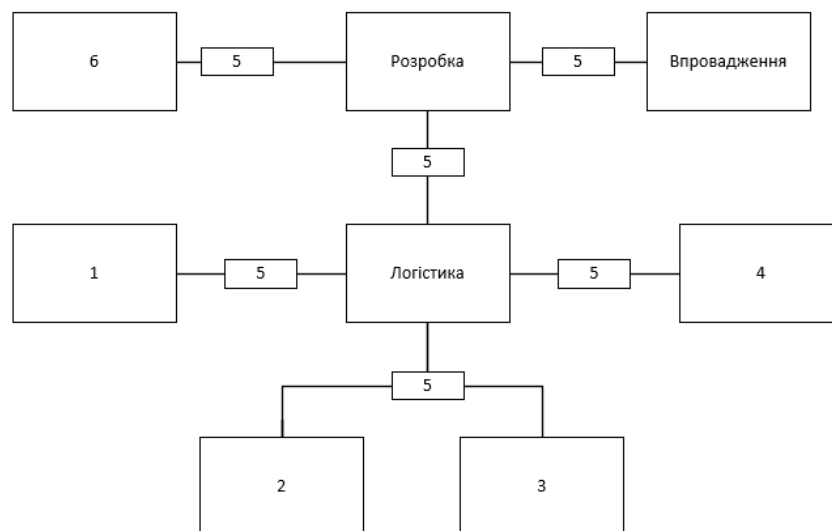


Рисунок 4.1 – Схема виробничих процесів

На схемі можуть бути відображені також етапи зберігання та транспортування сировини та готової продукції.

Контроль якості проведений на кожному етапі виробничого процесу. Наприклад, на етапі приймання сировини та комплектуючих виробів проведений візуальний контроль та вимірювання параметрів, а також перевірка документації. На етапі розробки проведений контроль відповідності технологічним вимогам та вимірювання параметрів якості. На етапі впровадження та маркування проведений контроль відповідності вимогам до упаковки та маркування.

При контролі якості слід використовувати відповідні стандарти, наприклад, національні та міжнародні стандарти якості продукції.

Для контролю якості на підприємстві використовуються наступні методи:

1. Візуальний огляд виробу: для перевірки зовнішнього вигляду та виявлення можливих дефектів.
2. Електронний контроль: для перевірки правильності розташування елементів на платі та їх правильного з'єднання між собою.
3. Функціональний тест: для перевірки правильності роботи програмного забезпечення та виявлення можливих помилок та несправностей.

Для контролю якості використовуються наступні стандарти:

1. Стандарти якості програмного забезпечення ISO 9001 та ISO 12207.
2. Стандарти якості апаратного забезпечення ISO 9001 та ISO 14001.

Таким чином, схема виробничих потоків включає процеси забезпечення сировиною та комплектуючими виробами, переробки на цехах, а також процеси доставки готової продукції до клієнтів. Важливим елементом на схемі є процеси контролю якості, які допоможуть забезпечити високу якість готової продукції та задоволення потреб клієнтів.

#### ***4.4.8. Оцінка можливих витрат.***

Оцінка можливих витрат виробництва є важливим етапом розробки бізнес-плану і допомагає зрозуміти, скільки коштуватиме запуск та ефективна робота підприємства.

На першому етапі, для запуску підприємства, необхідно придбати устаткування для виробництва та програмного забезпечення для розробки систем розумного дому. Витрати на устаткування можуть складати близько 200 000 доларів США, а витрати на програмне забезпечення – ще близько 50 000 доларів США.

Далі необхідно буде витратити кошти на зарплату працівників, оплату комунальних послуг та інші витрати, пов'язані з підтримкою виробничого процесу. Орієнтовні витрати на ці витрати складатимуть близько 30 000 доларів США на місяць.

Динаміка витрат на виробництво може змінюватися в залежності від рівня попиту на продукцію та ефективності виробництва. Якщо попит на продукцію зростатиме, то витрати на виробництво також збільшуватимуться. Проте, зростання попиту може призвести до збільшення обсягів виробництва, що в свою чергу дозволить знизити витрати на одиницю продукції.

Таким чином, витрати на виробництво можуть змінюватися у залежності від внутрішніх та зовнішніх факторів, що впливають на підприємство. Необхідно постійно контролювати витрати та здійснювати регулювання витрат на підприємстві з метою забезпечення його стабільної та прибуткової роботи.

## **4.5. Організаційний план**

### ***4.5.1. Кваліфікація фахівців***

Для успішної розробки інформаційного забезпечення та баз даних необхідно мати команду фахівців з наступними кваліфікаційними вимогами:

1. Аналітик даних – має мати освіту у галузі аналізу даних або інформаційних технологій. Кваліфікований аналітик повинен бути здатним розуміти бізнес–потреби клієнта та розробляти оптимальні рішення для забезпечення заданих потреб. Заробітна плата для такого спеціаліста зазвичай становить від \$3000 на місяць.
2. Розробник програмного забезпечення – має мати освіту в галузі комп'ютерних наук або інформаційних технологій. Кваліфікований

розробник повинен бути здатним розробляти та підтримувати високоякісне програмне забезпечення для забезпечення потреб бізнесу. Заробітна плата для такого спеціаліста зазвичай становить від \$4000 на місяць.

3. Системний архітектор – має мати освіту в галузі інформаційних технологій та багаторічний досвід у проектуванні складних систем. Кваліфікований архітектор повинен бути здатним проектувати та підтримувати системи, які забезпечують потреби бізнесу та відповідають вимогам безпеки даних. Заробітна плата для такого спеціаліста зазвичай становить від \$5000 на місяць.

Організацію нового проекту можна здійснити шляхом проведення відбору кандидатів з наступними кроками:

1. Пошук та відбір кандидатів з необхідною кваліфікацією.
2. Проведення співбесід з кандидатами для оцінки
  - 2.1. Перевірка технічних знань та навичок кандидатів у відповідності з вимогами проекту.
  - 2.2. Оцінка досвіду кандидатів у сфері розробки інформаційного забезпечення та баз даних.
  - 2.3. Оцінка комунікативних навичок та здатності до співпраці з іншими фахівцями у команді.
3. Підготовка договору проекту та укладання угод з фахівцями.
4. Організація взаємодії між фахівцями та визначення процесів розробки.
5. Розподіл завдань між фахівцями та встановлення строків виконання.
6. Відстеження процесу розробки та вирішення проблем, що виникають на шляху виконання проекту.
7. Оцінка якості розробки та підготовка проекту до запуску в експлуатацію.

Загальна заробітна плата для команди фахівців залежить від рівня складності проекту та термінів його виконання. Для розробки інформаційного забезпечення та баз даних на ринку праці можна очікувати заробітну плату від \$10 000 на місяць та вище.

#### ***4.5.2. Умови прийняття на роботу.***

Залежно від специфіки проекту, можуть бути прийняті як фахівці на постійну роботу, так і сумісники (зовнішні експерти).

Для постійних фахівців передбачені повний робочий день, відпустки, оплата за хворобу та інші соціальні гарантії, які надаються компанією згідно з трудовим законодавством.

Сумісники, як правило, працюють на проекті за окремими угодами на певний термін та отримують гонорар за свою роботу. Умови співпраці з зовнішніми експертами можуть бути визначені в окремому договорі, який містить умови оплати та відповідальності сторін.

#### ***4.5.3. Установи для пошуку робітників***

Використовуються послуги будь-якої організації, яка займається наймом професіоналів з розробки інформаційного забезпечення та баз даних. Найчастіше це є аутсорсингові компанії, які мають широкий діапазон послуг, пов'язаних з розробкою програмного забезпечення та IT-рішень. Такі компанії можуть запропонувати клієнту кваліфікованих фахівців з потрібним профілем та досвідом, які зможуть працювати над проектом як на постійній основі, так і як зовнішні експерти на певний термін.

#### ***4.5.4. Біографічні дані найнятих робітників***

Якщо частина персоналу вже найнята і працює на підприємстві, необхідно надати коротку біографічну інформацію про цих співробітників, щоб забезпечити їх інтеграцію в команду та налагодження ефективної роботи.

У біографічних даних про співробітників повинна бути вказана їх кваліфікація та досвід роботи, щоб виявити їхні здібності та корисність для підприємства. Наприклад, вказати спеціалізацію, рівень кваліфікації, додаткові знання та навички, досвід роботи відповідно до обраної спеціалізації та інші відомості, які стосуються професійної діяльності співробітника. Така інформація допоможе керівництву визначити потреби в додаткових фахівцях, встановити міжособистісні зв'язки та забезпечити ефективну комунікацію в колективі.

#### ***4.5.5. Організаційна структура підприємства***

Організаційна структура підприємства повинна відображати структуру керування, що забезпечує оптимальне функціонування та розвиток підприємства. Основними елементами організаційної структури є підрозділи та посадові особи, які відповідають за виконання певних функцій та завдань на підприємстві.

##### **1. Виробничий відділ:**

- 1.1. Директор з виробництва – відповідальний за загальний контроль за виробничим процесом, організацію та планування виробництва;
- 1.2. Технічний директор – займається розробкою і впровадженням нових технологій, контролює якість продукції;
- 1.3. Інженери – відповідають за виробничий процес, проводять технічні розрахунки та контролюють їх виконання;
- 1.4. Робітники виробництва – виконують безпосередні роботи на виробництві.

##### **2. Відділ збуту:**

- 2.1. Директор зі збуту – займається реалізацією продукції, контролює стан ринку та конкуренцію;
- 2.2. Менеджер з продажів – займається пошуком нових клієнтів та підтримкою відносин з наявними;

2.3. Менеджер з логістики – планує та організовує доставку продукції до клієнтів.

3. Відділ фінансів:

3.1. Головний бухгалтер – відповідає за бухгалтерський облік, звітність та оподаткування;

3.2. Фінансовий директор – займається фінансовим плануванням, розробкою бюджету та контролює витрати;

3.3. Економіст – займається аналізом економічної діяльності

## **4.6. Стратегія фінансування**

### ***4.6.1. Необхідна кількість засобів для реалізації проєкту***

Для відповіді на це питання ретельно проаналізовано всі аспекти проєкту, визначивши всі необхідні ресурси та їх вартість. Це включає витрати на купівлю обладнання, матеріалів, найм працівників, оренду приміщення та інші витрати, пов'язані з реалізацією проєкту. Потім потрібно скласти детальний бюджет, визначити суму необхідних коштів та джерела їхнього отримання, наприклад, інвестиційні кошти, банківські кредити, власні кошти підприємства та інше. У цьому процесі можуть бути корисні консультації з фінансовими експертами або фахівцями з бізнес-планування.

### ***4.6.2. Джерела фінансових ресурсів.***

Джерелами фінансових ресурсів для створення або розширення підприємства є:

- Власні засоби, які можуть бути залучені через власників підприємства, які вкладають свої гроші у розвиток бізнесу.
- Кредити банків, які надають позички підприємствам на різні умови та під різні види забезпечення.

- Залучення засобів партнерів, наприклад, за допомогою укладення спільних угод та договорів про співпрацю з іншими підприємствами або фізичними особами.
- Залучення засобів акціонерів, які вкладають свої гроші у створення підприємства або розширення його діяльності шляхом купівлі акцій.

Форми отримання фінансових ресурсів можуть бути різними, наприклад:

- Позички підприємству в банку або іншому фінансовому установі.
- Емісія акцій для залучення коштів від інвесторів.
- Укладення договорів про співпрацю з партнерами та використання їхніх ресурсів.
- Використання власних коштів підприємства або власників.

## ВИСНОВОК

З метою забезпечення комфортної та безпечної обстановки в будинках, на сьогоднішній день з'явилася потреба в системах "розумних" будинків. Ці системи дають змогу збільшити ефективність використання ресурсів, зменшити споживання енергії та забезпечити високий рівень комфорту та безпеки для мешканців.

У цьому проекті розроблено інформаційне забезпечення для системи розумного будинку, що включає в себе базу даних та програмне забезпечення для її використання. Для цього були використані сучасні технології програмування, які дозволяють ефективно працювати з базою даних та забезпечувати її надійність та безпеку.

Результатом даної роботи є створення функціональних частин системи розумного будинку, які може бути використані для базового створення реалізації ідеї комфортної та в подальшого розширення ефективної обстановки в будинках. Це може бути корисним не лише для приватних осіб, але і для бізнес-організацій, які бажають забезпечити високий рівень комфорту та безпеки для своїх клієнтів та працівників.

Важливість створення інформаційного забезпечення для систем розумного будинку полягає у тому, що воно є необхідною складовою для ефективної та надійної роботи системи. Без відповідної бази даних та програмного забезпечення, система розумного будинку не може функціонувати належним чином, і не може забезпечити потрібний рівень комфорту та безпеки для мешканців.

Відповідно, створення інформаційного забезпечення для систем розумного дому є ключовим елементом у забезпеченні ефективної та безпечної роботи таких систем. Застосування відповідних технологій та підходів у процесі проектування та розробки дозволяє досягти максимальної продуктивності та забезпечити користувачів необхідними функціями та можливостями.

Крім того, системи розумного дому стають все більш популярними та доступними для звичайних користувачів, оскільки вони дозволяють автоматизувати багато рутинних процесів та зменшити споживання енергії, що сприяє економії коштів на оплату комунальних послуг. Більш того, вони забезпечують безпеку та комфорт проживання в приватному будинку, а також можуть допомогти у вирішенні багатьох інших побутових проблем. Тож завдяки відомостям даної дипломної роботи, людина яка мінімально розбирається у IoT, може створити елементарну систему, розібравшись в програмному забезпеченні приладу та підключенні/створенні своєї бази даних.

Таким чином, створення інформаційного забезпечення для систем розумного дому є необхідним елементом у забезпеченні їх продуктивності, безпеки та комфорту користувачів. Розробка таких систем вимагає знань та досвіду у галузі комп'ютерних наук та програмування, а також уважної аналізування потреб та вимог користувачів. Але попри це дана ніша робить значний внесок у розвиток сфери IT в цілому.

Тож у першому розділі було описано аналіз та дослідження проблеми розробки інформаційного забезпечення та баз даних розумного дому, в основі чого досліджені питання: аналіз існуючих рішень розумних будинків та баз даних, постановка задачі, проблематика взаємозв'язку баз даних з реалізаціями наявних систем.

Другий розділ описує процес проектування інформаційного забезпечення, а саме: розглянуті ключові питання постановки завдання проектування ІЗ, дерева функцій, вибору технології зв'язку, вибору технології забезпечення безпеки, вибору інтерфейсу користувача( вимог до інтерфейсу, загальних можливостей структури інтерфейсу, вибору типу інтерфейсу користувача) та в подальшому проектування моделі бази даних.

Третій розділ виконує задачу вибору програмного інструментарію для подальшого створення та розробки бази даних та інформаційного забезпечення різних датчиків.

Четвертий розділ забезпечує інформацією щодо впровадження продукту на ринок. Описує задачу проектування продукту, конкуренції, плану виробництва, організаційний план та стратегії фінансування.

## СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Бекман П. Навчання розумного дому: практичні поради для використання технологій інтернету речей у побуті. Київ: Видавництво "Університетська книга", 2019.
2. Буряк, В. М. (2017). Розумний будинок: інформаційні технології та системи управління. Київ: Видавництво НТУ "ХПІ".
3. Горобець, О. М., Горобець, О. А., & Смільник, О. В. (2015). Розумний будинок: огляд технологій та архітектури системи. Електроніка та зв'язок, 20(3), 76-83.
4. Козловський, В. В., & Христіна, М. О. (2018). Використання методів машинного навчання у розумному будинку. Проблеми телекомунікацій, 2(18), 68-74.
5. Колесник, С. М., & Кузьменко, О. В. (2019). Аналіз архітектури системи розумного будинку. Вісник НТУ "ХПІ". Серія: Нові рішення в сучасних технологіях, 2(32), 76-82.
6. Кравченко, В. О., Коберник, Н. М., & Парфенюк, О. В. (2020). Розумний будинок на основі інтернету речей. Збірник наукових праць Уманського державного педагогічного університету імені Павла Тичини, 2(159), 153-160.
7. Мельник, Ю. М., & Серета, І. В. (2016). Методи і технології управління розумним будинком. Науковий вісник Міжнародного гуманітарного університету, 20(3), 40-45.
8. Нестеренко, І. В., & Касьян, Є. С. (2017). Особливості створення та реалізації системи розумного будинку. Електроніка та зв'язок, 22(1), 94-100.
9. Петров, Ю. В., & Рудницький, Є. М. (2018). Інтернет речей та розумний будинок. Матеріали Всеукраїнської науково-практичної конференції "Інформаційні технології та комп'ютерна інженерія", 63-65.
10. Столярчук, О. А., & Боднар, І. М. (2019). Впровадження системи розумного будинку на основі мікроконтролерів. Збірник наукових праць

Кіровоградського національного технічного університету, 2(53), 112-117.

11. Трубачов, В. М., & Маслов, С. В. (2017). Моделювання та аналіз архітектури системи розумного будинку. Вісник Харківського національного університету Повітряних Сил імені Івана Кожедуба. Серія: "Радіоелектроніка та інформатика", 2(50), 42-46.
12. Філатов, Д. В., & Константинова, В. Є. (2019). Розумний будинок на базі системи "Розумне село". Вісник НТУ "ХПІ". Серія: Нові рішення в сучасних технологіях, 1(31), 68-74.
13. Al-Rousan, M., Al-Ali, A., & Khalifeh, H. (2018). A smart home energy management system using internet of things and big data analytics approach. *IEEE Access*, 6, 64242-64254.
14. Atzori, L., Iera, A., & Morabito, G. (2010). The Internet of Things: A survey. *Computer Networks*, 54(15), 2787-2805.
15. Bandyopadhyay, D., & Sen, J. (2011). Internet of things: Applications and challenges in technology and standardization. *Wireless Personal Communications*, 58(1), 49-69.
16. Chaouchi, H., & Lefevre, L. (2013). *The internet of things: Connecting objects to the web*. John Wiley & Sons.
17. Christin, D., Reinhardt, A., & Mogre, P. S. (2016). Integrating building automation systems with the internet of things using the MQTT protocol. In *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct* (pp. 1111-1116).
18. Gubbi, J., Buyya, R., Marusic, S., & Palaniswami, M. (2013). Internet of Things (IoT): A vision, architectural elements, and future directions. *Future Generation Computer Systems*, 29(7), 1645-1660.
19. Khalifa, A., & Gharib, T. F. (2019). Internet of Things (IoT) Based Smart Home System. *International Journal of Computer Science and Network Security (IJCSNS)*, 19(1), 209-216.
20. Liu, D., Keally, M., & Bo, W. (2017). Research on the Development of Smart Home System Based on IoT Technology. In *Proceedings of the 2017*

International Conference on Education, Management, Computer and Society (pp. 331-335).

21. Yu, X., Tian, S., & Zhang, X. (2017). Design and Implementation of Intelligent Home Control System Based on the Internet of Things. In Proceedings of the 2017 4th International Conference on Information Science and Control Engineering (pp. 1645-1649).

## Підключення датчика диму до системи

### 1. Створення таблиці у базі даних:

```
CREATE TABLE SmokeSensorData
(
    Id INT PRIMARY KEY IDENTITY,
    DateAndTime DATETIME,
    SmokeLevel INT
)
```

### 2. Зчитування даних з датчика диму:

```
// Припустимо, що ви маєте об'єкт датчика диму, який надає дані про рівень диму
```

```
SmokeSensor smokeSensor = new SmokeSensor();
```

```
// Отримання рівня диму з датчика
```

```
int smokeLevel = smokeSensor.GetSmokeLevel();
```

### 3. Запис даних в базу даних:

```
// Встановлення з'єднання з базою даних
```

```
SqlConnection connection = new
SqlConnection("ВАША_РЯДОК_ПІДКЛЮЧЕННЯ");
```

```
// Вставка даних в таблицю SmokeSensorData
```

```
string query = "INSERT INTO SmokeSensorData (DateAndTime,
SmokeLevel) VALUES (@DateAndTime, @SmokeLevel)";
```

```
SqlCommand command = new SqlCommand(query, connection);
```

```
command.Parameters.AddWithValue("@DateAndTime", DateTime.Now);
```

```
command.Parameters.AddWithValue("@SmokeLevel", smokeLevel);
```

```
// Виконання запиту
```

```
connection.Open();
```

```
command.ExecuteNonQuery();
```

```
connection.Close();
```

### 4. Отримання даних з бази даних:

```
// Встановлення з'єднання з базою даних
```

```
SqlConnection connection = new
SqlConnection("ВАША_РЯДОК_ПІДКЛЮЧЕННЯ");

// Вибірка даних з таблиці SmokeSensorData
string query = "SELECT DateAndTime, SmokeLevel FROM
SmokeSensorData";
SqlCommand command = new SqlCommand(query, connection);

// Виконання запиту та отримання результатів
connection.Open();
SqlDataReader reader = command.ExecuteReader();
while (reader.Read())
{
    DateTime dateTime = (DateTime)reader["DateAndTime"];
    int smokeLevel = (int)reader["SmokeLevel"];
    // Обробка отриманих даних
    // ...
}
connection.Close();
```

## Підключення датчика витоку води

### 1. Створення таблиці у базі даних:

```
CREATE TABLE WaterLeakSensorData
(
    Id INT PRIMARY KEY IDENTITY,
    DateAndTime DATETIME,
    LeakStatus BIT
)
```

### 2. Зчитування даних з датчика витоку води:

```
// Припустимо, що ви маєте об'єкт датчика витоку води, який надає
дані про стан витоку

WaterLeakSensor waterLeakSensor = new WaterLeakSensor();

// Отримання стану витоку води з датчика
bool leakStatus = waterLeakSensor.GetLeakStatus();
```

### 3. Запис даних в базу даних:

```
// Встановлення з'єднання з базою даних
SqlConnection connection = new SqlConnection("РЯДОК_ПІДКЛЮЧЕННЯ");

// Вставка даних в таблицю WaterLeakSensorData
string query = "INSERT INTO WaterLeakSensorData (DateAndTime,
LeakStatus) VALUES (@DateAndTime, @LeakStatus)";
SqlCommand command = new SqlCommand(query, connection);
command.Parameters.AddWithValue("@DateAndTime", DateTime.Now);
command.Parameters.AddWithValue("@LeakStatus", leakStatus);

// Виконання запиту
connection.Open();
command.ExecuteNonQuery();
connection.Close();
```

### 4. Отримання даних з бази даних:

```
// Встановлення з'єднання з базою даних
```

```
SqlConnection connection = new
SqlConnection("ВАША_РЯДОК_ПІДКЛЮЧЕННЯ");

// Вибірка даних з таблиці WaterLeakSensorData
string query = "SELECT DateAndTime, LeakStatus FROM
WaterLeakSensorData";
SqlCommand command = new SqlCommand(query, connection);

// Виконання запиту та отримання результатів
connection.Open();
SqlDataReader reader = command.ExecuteReader();
while (reader.Read())
{
    DateTime dateTime = (DateTime)reader["DateAndTime"];
    bool leakStatus = (bool)reader["LeakStatus"];
    // Обробка отриманих даних
    // ...
}
connection.Close();
```

## Підключення сенсорів розумного освітлення

## 1. Датчик руху:

```
// Припустимо, що ви маєте об'єкт датчика руху, який надає дані про рух
MotionSensor motionSensor = new MotionSensor();

// Перевірка наявності руху
bool motionDetected = motionSensor.IsMotionDetected();

// Керування освітленням на основі датчика руху
if (motionDetected)
{
    // Вмикаємо освітлення
    LightController.TurnOn();
}
else
{
    // Вимикаємо освітлення
    LightController.TurnOff();
}
```

## 2. Датчик освітленості:

```
// Припустимо, що ви маєте об'єкт датчика освітленості, який надає дані про рівень освітленості
LightSensor lightSensor = new LightSensor();

// Отримання рівня освітленості
double lightLevel = lightSensor.GetLightLevel();

// Керування освітленням на основі датчика освітленості
if (lightLevel < 50) // Припустимо, що поріговий рівень освітленості - 50
{
    // Вмикаємо освітлення
    LightController.TurnOn();
}
else
{
    // Вимикаємо освітлення
    LightController.TurnOff();
}
```