СХІДНОУКРАЇНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ВОЛОДИМИРА ДАЛЯ

Навчально-науковий інститут (факультет) інформаційних технологій та електроніки

Кафедра інформаційних технологій та програмування

ПОЯСНЮВАЛЬНА ЗАПИСКА

до кваліфікаційної випускної роботи

освітній ступінь бакалавр
спеціальність 121 „Інженерія програмного забезпечення”
(шифр і назва спеціальності)

спеціалізація „Інженерія програмного забезпечення”
(назва спеціалізації)

на тему „Мобільний додаток до системи розумний будинок”

Виконав: студент групи ІПЗ-20д _____ Копатько Д. О.
(підпис) (ініціали і прізвище)

Керівник _____ Лифар В. О.
(підпис) (ініціали і прізвище)

Завідувач кафедри _____ Захожай О. І.
(підпис) (ініціали і прізвище)

Рецензент Ратов Д. В.

Київ – 2024

СХІДНОУКРАЇНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ВОЛОДИМИРА ДАЛЯ

Навчально-науковий інститут (факультет) інформаційних технологій та електроніки
Кафедра інформаційних технологій та програмування
Освітній ступінь бакалавр
спеціальність 121 „Інженерія програмного забезпечення”
(шифр і назва спеціальності)
спеціалізація „Інженерія програмного забезпечення”
(назва спеціалізації)

ЗАТВЕРДЖУЮ
Завідувач кафедри

_____ Захожай О.І.
“___” _____ 2024 року

З А В Д А Н Н Я
НА КВАЛІФІКАЦІЙНУ ВИПУСКНУ РОБОТУ СТУДЕНТУ

Копатько Даніїл Олексійович
(прізвище, ім’я, по батькові)

1. Тема роботи: Реалізація мобільного додатку до системи “розумний будинок”

керівник роботи Доц., д.т.н. Лифар В.О.
(прізвище, ім’я, по батькові, науковий ступінь, вчене звання)

затверджений наказом університету від “_06_”_травня_2024 року №171/15.15-С

2. Строк подання студентом роботи 08.06.2024р.
3. Вихідні дані до роботи: Об’єктом даної роботи є процес

Розробка мобільного додатку до системи “розумний будинок”

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити):

поширення мобільних операційних систем у світі, важливість для дому мати
мобільний додаток, порівняння мов програмування для розробки додатку. Висновки

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслеників)

6. Консультанти розділів роботи

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

7. Дата видачі завдання 30.03.2024р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів виконання кваліфікаційної випускної роботи	Строк виконання етапів	Примітка
1	Одержання завдання на виконання роботи	30.03.24	виконано
2	Укладання і погодження з керівником плану і етапів виконання роботи	06.04.24	виконано
3	Узагальнення даних літературних джерел, укладання розділу «Аналіз предметної галузі»	13.04.24	виконано
4	Аналіз шляхів виконання завдання. Вибір і погодження з керівником оптимального шляху	20.04.24	виконано
5	Укладання та тестування програмного продукту	27.04.24	виконано
6	Укладання, оформлення та погодження пояснювальної записки з керівником	04.05.24	виконано
7	Здача готової пояснювальної записки на	08.06.24	виконано

	кафедру		
8	Укладання доповіді і презентації	10.06.24	виконано

Студент _____ Копатько Д. О.
підпис (ініціали і прізвище)

Керівник роботи _____ Лифар В.О.
підпис (ініціали і прізвище)

ЛИСТ ПОГОДЖЕННЯ І ОЦІНЮВАННЯ

дипломної роботи студента гр. ІПЗ-20д Копатько Д. О.

Науковий керівник:

Професор, д.т.н. _____ Лифар В. О.

Оцінка наукового керівника: _____

Рецензент

ПІБ, місто роботи, посада

Оцінка рецензента: _____

Кінцева оцінка за результатами захисту:

Голова ЕК
Професор кафедри ІТП
д.т.н. _____ Меняйленко О.С.

РЕФЕРАТ

У сучасному світі технології стали невід'ємною частиною нашого повсякденного життя. Однією з таких технологій є концепція "Розумного будинку" (Smart Home), яка забезпечує автоматизацію та управління різноманітними системами і пристроями в будинку. Даний реферат присвячений розробці мобільного додатку до системи "Розумний будинок", що дозволяє інтегрувати та контролювати різні компоненти розумного будинку з єдиного інтерфейсу.

Основною метою даної роботи є розробка мобільного додатку для системи "Розумний будинок", який забезпечить зручний та інтуїтивно зрозумілий інтерфейс для користувачів, дозволяючи їм ефективно керувати умовами у своєму домі.

До системи були інтегровані різні типи сенсорів та пристроїв, такі як температурні датчики, датчики вологості, диму та газу, розумні лампи, термостати, розетки, дверні замки та камери спостереження. Це дозволило забезпечити комплексне управління умовами у будинку, зокрема регулювання температури та вологості, освітлення, безпеки та енергоефективності.

У результаті виконаної роботи було створено мобільний додаток для системи "Розумний будинок", який забезпечує зручний та ефективний інтерфейс для управління умовами у будинку.

Зміст

ВСТУП[1]

"Розумний будинок" - це сучасна технологічна концепція, яка перетворює наше уявлення про житло, інтегруючи інноваційні рішення для забезпечення комфорту, безпеки та ефективності. Система розумного будинку використовує мережу взаємопов'язаних пристроїв, датчиків і автоматизованих систем, які дозволяють користувачам дистанційно керувати та моніторити різні аспекти їхнього будинку через мобільні додатки або голосові команди. Від освітлення і опалення до систем безпеки і управління побутовими приладами - все це можна контролювати за допомогою єдиного інтерфейсу.

Розумні будинки сприяють оптимізації енергоспоживання, підвищують рівень безпеки та забезпечують персоналізований комфорт для кожного члена сім'ї. Завдяки впровадженню технологій інтернету речей (IoT) та штучного інтелекту (AI), розумний будинок може адаптуватися до потреб і звичок мешканців, автоматично налаштовуючи роботу пристроїв відповідно до заданих сценаріїв.

Розробка і впровадження системи розумного будинку відкриває нові горизонти в управлінні житлом, роблячи його більш зручним, безпечним та екологічно відповідальним. Це не лише покращує якість життя, але й допомагає економити час і ресурси, пропонуючи ефективні рішення для повсякденних задач.

1

РОЗДІЛ 1. АНАЛІТИЧНИЙ ОГЛЯД

Концепція "Розумний будинок" стає все більш популярною в сучасному світі, де технології інтегруються у всі аспекти нашого життя. Розумний будинок представляє собою інтегровану систему, що об'єднує різноманітні пристрої та технології для створення комфортного, безпечного і енергоефективного житлового середовища. Основною метою цієї системи є надання користувачам можливості керувати та контролювати свої будинки за допомогою мобільних додатків або голосових команд, роблячи їхнє життя зручнішим і більш ефективним.

Основними компонентами системи розумного будинку є сенсори і датчики, пристрої управління, центральний хаб і мобільні додатки. Сенсори і датчики вимірюють різні параметри середовища, такі як температура, вологість, освітленість, рух, дим та газ. Ці дані використовуються для автоматичного регулювання роботи пристроїв, таких як освітлення, опалення, вентиляція і кондиціонування повітря. Пристрої управління включають розумні лампи, термостати, розетки, дверні замки, камери спостереження і побутову техніку, які можуть бути керовані дистанційно через мобільні додатки або голосові команди. Центральний хаб координує роботу всіх підключених пристроїв і сенсорів, забезпечуючи їх взаємодію і обмін даними. Мобільні додатки надають користувачам інтерфейс для керування пристроями, отримання сповіщень і налаштування сценаріїв автоматизації. Функціональність системи розумного будинку включає автоматизацію, моніторинг і контроль, безпеку та енергоефективність. Автоматизація дозволяє налаштовувати автоматичні сценарії дій, такі як ввімкнення світла при вході в кімнату або регулювання температури залежно від часу доби. Моніторинг і контроль забезпечують можливість відстежувати стан будинку в режимі реального часу і керувати пристроями з будь-якої точки світу. Системи безпеки включають відеоспостереження, сигналізацію, датчики диму та витоку газу, що забезпечують додатковий рівень захисту для мешканців і їхнього майна. Енергоефективність досягається за рахунок оптимізації використання енергії, автоматичного вимкнення непотрібних пристроїв і регулювання освітлення та опалення.

Серед основних переваг системи розумного будинку можна виділити зручність і комфорт, підвищену безпеку, економію ресурсів та

персоналізацію. Автоматизація повсякденних задач робить життя більш зручним і приємним, дозволяючи користувачам зосередитися на більш важливих справах. Системи безпеки та моніторингу забезпечують додатковий рівень захисту для мешканців і їхнього майна, зменшуючи ризик інцидентів і збитків. Енергоефективні рішення допомагають знизити витрати на електроенергію та інші ресурси, що сприяє збереженню довкілля. Можливість налаштування системи під індивідуальні потреби і вподобання користувача дозволяє створити максимально комфортні умови для кожного члена родини.

Незважаючи на численні переваги, система розумного будинку стикається з низкою викликів. Висока вартість початкових інвестицій в обладнання і установку може бути бар'єром для багатьох споживачів. Складність інтеграції різних пристроїв і забезпечення їх сумісності може вимагати значних технічних знань і навичок. Питання безпеки і конфіденційності також є важливими, оскільки зберігання і передача даних потребують надійного захисту від несанкціонованого доступу і кібератак. Надмірна залежність від технологій може бути проблематичною у разі технічних збоїв або відсутності інтернет-зв'язку.

У підсумку, система "Розумний будинок" представляє собою перспективне рішення для сучасного житла, пропонуючи широкий спектр можливостей для підвищення комфорту, безпеки та ефективності. Незважаючи на наявні виклики, розвиток технологій і зниження вартості обладнання сприяють все більшому поширенню цих систем. У майбутньому очікується, що розумні будинки стануть невід'ємною частиною повсякденного життя, пропонуючи ще більше функцій і можливостей для покращення якості життя.

1. **1. Мови програмування**

Для створення мобільних додатків "Розумний будинок" використовують кілька мов програмування та фреймворків.

Ось основні з них:

Для Android:

1. **1. 1. Java**

- Традиційна мова для розробки Android-додатків.

  Java широко використовується для розробки різних типів програмного забезпечення, включаючи корпоративні додатки, мобільні додатки (особливо для Android), веб-додатки та серверні рішення.

  Переваги:

  - **Портативність**: Завдяки JVM (Java Virtual Machine) код, написаний на Java, може виконуватися на будь-якій платформі, яка підтримує JVM.
  - **Велика стандартна бібліотека**: Набір бібліотек Java включає багатий набір готових функцій для вирішення багатьох задач.
  - **Масштабованість**: Добре підходить для великих проектів та корпоративних додатків.
  - **Висока продуктивність**: Завдяки сучасним методам оптимізації та JIT-компіляції.
  - **Безпека**: Механізми безпеки, вбудовані в JVM, роблять Java безпечною для мережевих додатків.

  Відмінності:

  - **Синтаксис**: Строгий синтаксис з багатьма правилами та типізацією.

- **Швидкість виконання**: Може бути повільнішою у порівнянні з компільованими мовами, такими як C++.

1. 1. 2. Kotlin

- Сучасна мова програмування, офіційно підтримувана Google і активно використовується для розробки Android-додатків. Kotlin стала офіційно підтримуваною мовою для розробки Android-додатків, що значно підвищило її популярність серед мобільних розробників. Вона також використовується для серверних та веб-додатків, а завдяки своїй гнучкості і сучасним можливостям, Kotlin отримує все більше визнання в спільноті розробників.

Переваги:

- **Простий синтаксис**: Легше для вивчення та використання, ніж Java, завдяки більш лаконічному коду.
- **Повна сумісність з Java**: Можна використовувати разом з існуючим Java-кодом.
- **Безпечність типів**: Зменшує ризик виникнення NullPointerException завдяки вбудованій підтримці nullable і non-nullable типів.
- **Функціональні можливості**: Підтримує функціональне програмування, що дозволяє писати більш гнучкий та зрозумілий код.

Відмінності:

- **Новизна**: Менше ресурсів та спільнот у порівнянні з Java.
- **Підтримка**: Вимагає Java для виконання, так як компілюється в байт-код JVM.

Для iOS:

1. 1. 3. Swift

- Основна мова для розробки додатків під iOS. Розроблена Apple і широко використовується для створення сучасних та продуктивних додатків.
  Swift широко використовується для розробки мобільних додатків для платформ Apple, таких як iOS, macOS, watchOS та tvOS.

Переваги:

- **Продуктивність**: Висока продуктивність завдяки компіляції в рідний код.
- **Безпека**: Сильна типізація та управління пам'яттю зменшують кількість помилок.
- **Сучасний синтаксис**: Зручний та інтуїтивно зрозумілий, підтримує функціональне та об'єктно-орієнтоване програмування.
- **Інтероперабельність**: Можна легко інтегрувати з Objective-C.

Відмінності:

- **Обмежена платформа**: Використовується переважно для розробки під iOS/macOS.
- **Ком’юніті**: Менше ресурсів та бібліотек у порівнянні з деякими іншими мовами.

1. 1. 4. JavaScript (React Native)

- Раніше основна мова для розробки iOS-додатків. Все ще використовується, але Swift поступово її витісняє.
  Java Script використовується для розробки різноманітних веб-додатків, включаючи веб-інтерфейси, інтерактивні веб-сторінки, ігри, мобільні додатки, а також для розробки серверних додатків за допомогою платформи Node.js. Вона є однією з основних технологій веб-розробки та має широкий спектр застосувань.

Переваги:

- **Крос-платформеність**: Можна писати один код для Android та iOS.

- **Велика спільнота**: Багато доступних бібліотек та інструментів.
- **Висока продуктивність**: Завдяки використанню нативних компонентів.
- **Гарний досвід розробки**: Завдяки гарячому перезавантаженню (hot reload).

Відмінності:

- **Менеджмент стану**: Потребує сторонніх бібліотек для управління станом (наприклад, Redux).
- **Продуктивність**: Може бути нижчою у порівнянні з нативними мовами для складних додатків.

Висновок:

Вибір мови програмування залежить від конкретних вимог вашого проекту. Java та Kotlin є чудовим вибором для Android-додатків, Swift - для iOS, а крос-платформені рішення, такі як React Native та Flutter, дозволяють створювати додатки для обох платформ одночасно. Кожна мова має свої переваги та недоліки, тому важливо враховувати їх при плануванні розробки.

РОЗДІЛ 2. МОДЕЛЬ ПРОЕКТУ ТА ПОСТАНОВКА ЗАДАЧІ

2. 1. Як працює автоматизація розумного будинку?

Домашня автоматизація - це система гаджетів, підключених до інтернету через бездротові протоколи зв'язку рис. 1.1.1.. Цими гаджетами керують такі контролери, як голос, асистент, різні додатки, електронні інтерфейси тощо. Кожен гаджет підключений до інтернету через WiFi/BLE або іншим способом, що дозволяє вам керувати речами у вашому домі, не перебуваючи там, або навіть контролювати все автоматично.

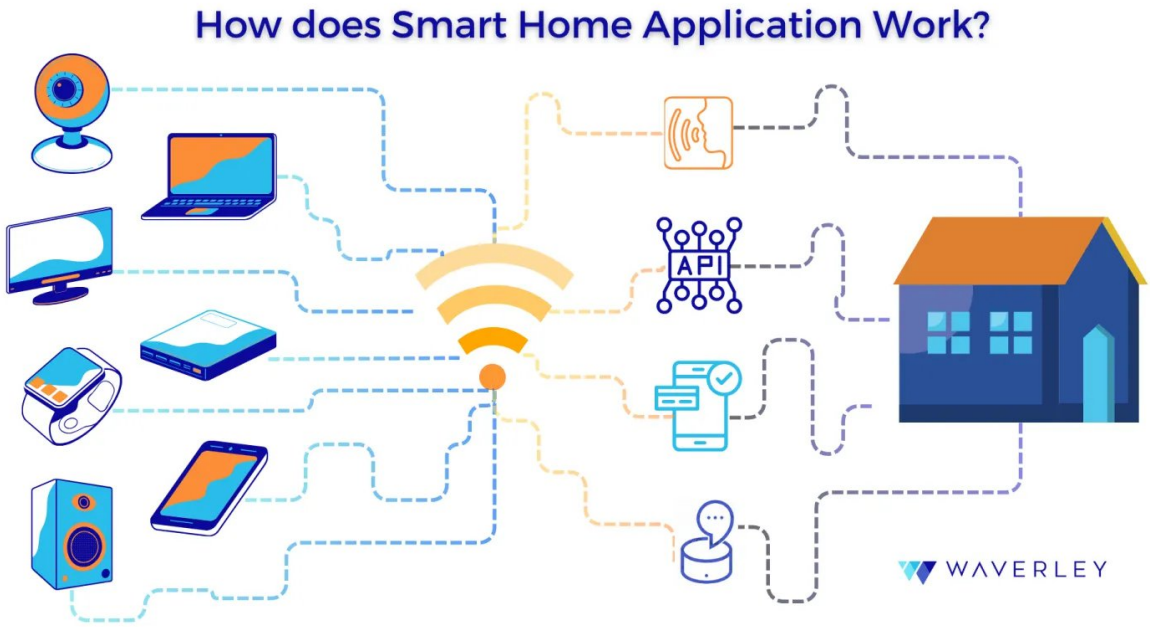


Рис. 1.1.1. - Як працює автоматизація розумного будинку?

2. 2. Компоненти системи "Розумний будинок"

Сучасна система "Розумний будинок" об'єднує інженерні рішення, впроваджені на етапі будівництва, з інформаційними засобами керування та контролю всередині будівлі. Вона відповідає за контроль і покращення житлового простору, підвищення його функціональності, комфорту та енергоефективності. Такі можливості забезпечуються створенням цілісної екосистеми з автоматизованих датчиків і пристроїв, які керуються за допомогою сучасних бездротових технологій. "Розумний будинок" є системою інтелектуальної автоматики для управління інженерними системами будівлі, здатною задовольнити всі потреби власника будинку:

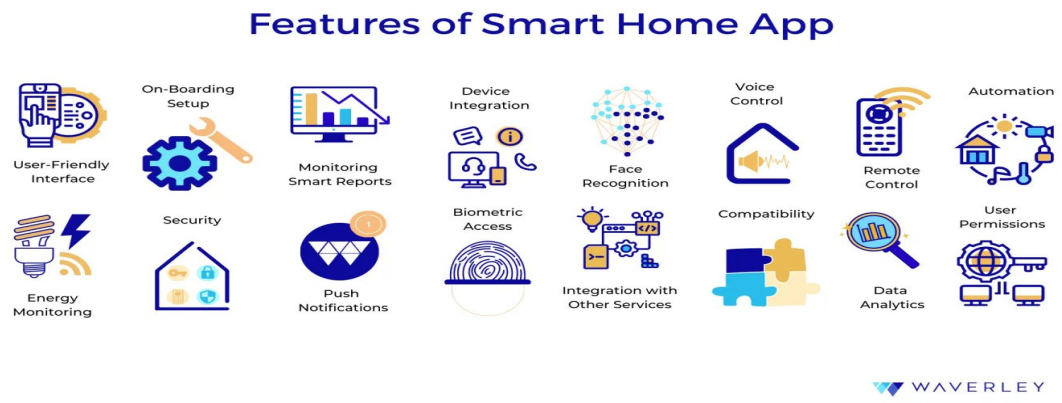
регулювання температури, вологості, освітлення, керування аудіосистемою, забезпечення безпеки та інше.

Основними компонентами системи розумного будинку є:

- **Сенсори і датчики**: Вони вимірюють різні параметри середовища, такі як температура, вологість, освітленість, рух, дим та газ. Ці дані використовуються для автоматичного регулювання роботи пристроїв.
- **Пристрої управління**: До них відносяться розумні лампи, термостати, розетки, дверні замки, камери спостереження і побутова техніка. Всі ці пристрої можуть бути керовані дистанційно.
- **Центральний хаб**: Це центральний контролер, який координує роботу всіх підключених пристроїв і сенсорів, забезпечуючи їх взаємодію і обмін даними.
- **Мобільні додатки**: Забезпечують інтерфейс для користувача, через який він може керувати пристроями, отримувати сповіщення і налаштовувати сценарії автоматизації.

Особливості програми Розумний будинок

Вважається, що програмне забезпечення для смартфона, яке керує приладами розумного будинку та автоматизує будинок, є додатком для розумного будинку. Рис. 1.1.2



1.1.2 - Особливості розумного будинку

2. 3. Функціональність

Система розумного будинку забезпечує наступну функціональність:

- **Автоматизація**: Налаштування автоматичних сценаріїв, таких як ввімкнення світла при вході в кімнату або регулювання температури залежно від часу доби.
- **Моніторинг і контроль**: Можливість відстежувати стан будинку в режимі реального часу і керувати пристроями з будь-якої точки світу.
- **Безпека**: Встановлення систем відеоспостереження, сигналізації, датчиків диму та витоку газу для забезпечення безпеки житла.
- **Енергоефективність**: Оптимізація використання енергії шляхом автоматичного вимкнення непотрібних пристроїв і регулювання освітлення та опалення.

2. 4. Переваги

Серед основних переваг системи розумного будинку можна виділити:

- **Зручність і комфорт**: Автоматизація повсякденних задач робить життя більш зручним і приємним.
- **Підвищена безпека**: Системи безпеки та моніторингу забезпечують додатковий рівень захисту для мешканців і їх майна.
- **Економія ресурсів**: Енергоефективні рішення допомагають знизити витрати на електроенергію та інші ресурси.
- **Персоналізація**: Можливість налаштування системи під індивідуальні потреби і вподобання користувача.

2. 5. Виклики

Незважаючи на численні переваги, система розумного будинку стикається з низкою викликів:

- **Висока вартість**: Початкова інвестиція в обладнання і установку може бути досить високою.
- **Складність інтеграції**: Підключення різних пристроїв і забезпечення їх сумісності може бути складним завданням.
- **Питання безпеки і конфіденційності**: Зберігання і передача даних потребують надійного захисту від несанкціонованого доступу і кібератак.
- **Залежність від технологій**: Надмірна автоматизація може призвести до залежності від технологій, що може бути проблематичним у разі технічних збоїв.

2. 6. Взаємозв'язки між компонентами

- **Сенсори та датчики -> Центральний хаб:** передають дані про стан середовища.
- **Центральний хаб -> Пристрої управління:** надсилає команди на основі отриманих даних і налаштованих сценаріїв.
- **Центральний хаб -> Мобільні додатки:** синхронізує дані та забезпечує користувачам можливість дистанційного управління.
- **Мобільні додатки -> Пристрої управління:** користувачі надсилають команди безпосередньо через додатки.
- **Голосові асистенти -> Центральний хаб:** отримують та передають голосові команди на хаб для виконання.

2. 7. Діаграма

Для побудови діаграми моделі предметної галузі можна використати UML діаграму класів або діаграму компонентів, яка відображає взаємодію між основними компонентами системи. Нижче наведено спрощену UML діаграму класів для системи "Розумний будинок":

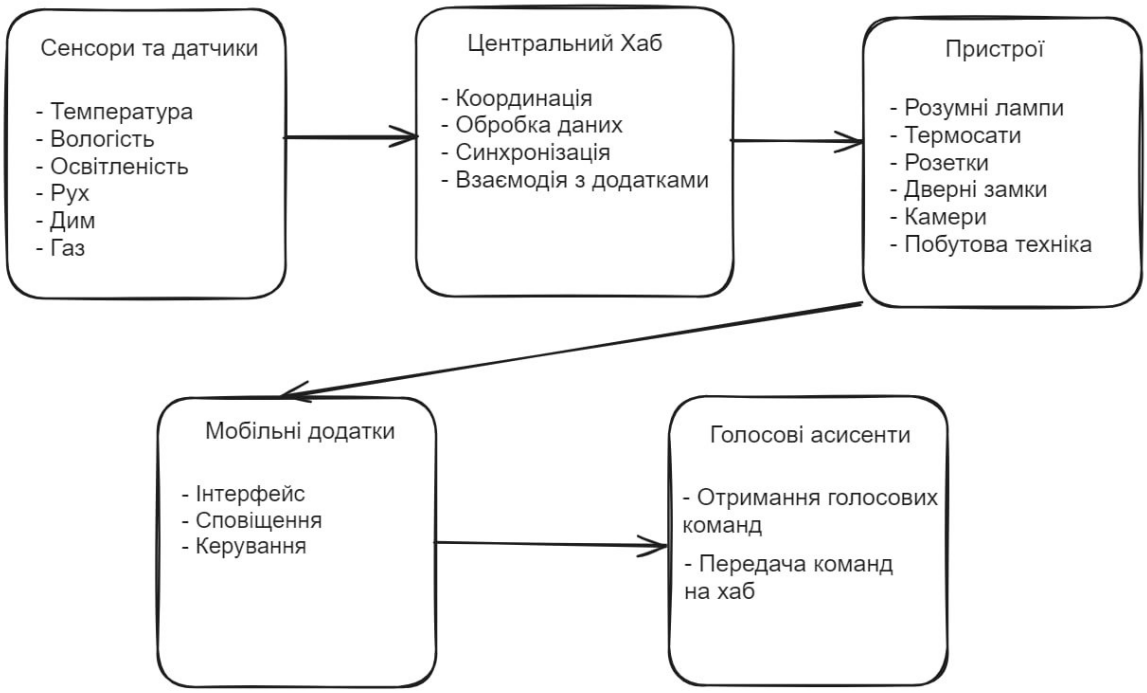


Рисунок 1.6.1 - Діаграма

Пояснення до діаграми

1) **Сенсори та датчики** передають дані про стан середовища до **Центрального хабу**.
2) **Центральний хаб** обробляє отримані дані та синхронізує їх з **Мобільними додатками**, а також надсилає команди до **Пристроїв управління**.
3) **Мобільні додатки** дозволяють користувачам керувати пристроями, налаштовувати сценарії автоматизації та отримувати сповіщення.
4) **Голосові асистенти** отримують голосові команди від користувачів і передають їх на **Центральний хаб** для виконання.

Ця модель охоплює всі основні аспекти функціонування системи "Розумний будинок", включаючи компоненти, їх взаємозв’язки, а також основні функції та процеси, що забезпечують ефективне управління житловим простором.

2. 8. Інструментарій для розробки Android

Розробка сучасних Android-додатків для автоматизації розумного будинку вимагає знайомства з різними інструментами та фреймворками. Ось деякі популярні з них, на які варто звернути увагу:

- **Android Studio:** Офіційне інтегроване середовище розробки (IDE) для розробки додатків для Android, яке надає набір потужних функцій та інструментів.
- **Сервіси Google Play:** Важлива бібліотека, яка пропонує API для взаємодії з сервісами Google, включаючи push-сповіщення, автентифікацію та служби визначення місцезнаходження.
- **Firebase:** Потужна внутрішня платформа, яка надає функціональність бази даних у реальному часі, хмарний обмін повідомленнями та автентифікацію користувачів, серед інших функцій.
- **Платформи Інтернету речей:** Платформи Інтернету речей, такі як Samsung SmartThings, Amazon Alexa та Google Assistant, які пропонують API для інтеграції з широким спектром розумних пристроїв.

2. 9. Кращі практики розробки додатків для розумного будинку

Щоб забезпечити успішну розробку додатків для Android в контексті автоматизації розумного будинку, зверніть увагу на наступні найкращі практики:

- **Дизайн, орієнтований на користувача:** Надаємо перевагу зручному інтерфейсу, який пропонує інтуїтивно зрозумілу навігацію та безперешкодний користувацький досвід для контролю та управління пристроями розумного будинку.
- **Належне тестування та налагодження:** Ретельно протестуємо додаток на різних пристроях і забезпечте сумісність з різними версіями Android, щоб забезпечити надійну роботу без помилок.

- **Оптимізована продуктивність:** Оптимізуємо продуктивність програми, щоб забезпечити плавну та швидку взаємодію з розумними пристроями, навіть при інтенсивному використанні або низькій якості мережі.
- **Постійні оновлення та підтримка:** Треба бути в курсі останніх версій Android, патчів безпеки та виправлень помилок, щоб забезпечити постійне покращення роботи з додатком для ваших користувачів.
- **Конфіденційність і безпека даних:** Впроваджуємо надійні заходи безпеки, включаючи шифрування та безпечну автентифікацію, щоб захистити дані користувачів і забезпечити безпечну взаємодію з підключеними пристроями.

Оскільки попит на розумні будинки продовжує зростати, розробка додатків для Android для автоматизації розумного будинку відкриває перед розробниками величезні можливості. Враховуючи основні функції, використовуючи правильні інструменти та дотримуючись найкращих практик, ви можете створювати найсучасніші додатки для Android, які забезпечують безперебійний контроль та автоматизацію розумних пристроїв.

Ключові міркування щодо розробки додатків Android для автоматизації розумного будинку

Однак є кілька ключових міркувань, які розробники повинні мати на увазі, щоб забезпечити успіх своїх додатків для Android у сфері автоматизації розумного будинку.

2. 9. 1. Безшовна інтеграція з пристроями розумного будинку

Одним з найважливіших аспектів розробки додатків для Android для автоматизації розумного будинку є забезпечення безперешкодної інтеграції з широким спектром розумних пристроїв. Від систем освітлення і термостатів до камер спостереження і дверних замків - додаток повинен мати можливість ефективно взаємодіяти з різними пристроями, використовуючи такі протоколи, як Wi-Fi, Bluetooth або Zigbee. Це надасть користувачам уніфіковану платформу управління, що покращить загальний користувацький досвід.

Основні висновки:

- Забезпечення безперешкодної інтеграції з різними пристроями розумного будинку
- Підтримка декількох протоколів зв'язку, таких як Wi-Fi, Bluetooth і Zigbee
- Створення уніфікованої платформи управління для покращення користувацького досвіду

2. 9. 2. Зручний інтерфейс та можливості персоналізації

Створення зручного інтерфейсу є запорукою успіху будь-якого додатку для Android, і автоматизація розумного будинку не є винятком. Додаток повинен пропонувати інтуїтивно зрозумілу навігацію, візуально привабливий дизайн і прості для розуміння елементи керування. Крім того, надання опцій персоналізації, які дозволяють користувачам налаштовувати додаток відповідно до своїх уподобань, додає значного рівня зручності та підвищує залученість користувачів.

Основні висновки:

- Створіть інтуїтивно зрозумілий та візуально привабливий користувацький інтерфейс
- Використовуйте зрозумілі елементи керування для легкої навігації
- Запропонуйте варіанти персоналізації, щоб підвищити залученість користувачів

2. 9. 3. Розширені заходи безпеки

З поширенням "розумних" домашніх пристроїв забезпечення безпеки даних користувачів та їхньої приватності набуває першочергового значення. Розробники додатків для Android повинні впроваджувати надійні заходи безпеки для захисту від потенційних вразливостей і несанкціонованого доступу. Шифрування, автентифікація та управління дозволами - це найважливіші аспекти, на яких розробники повинні

зосередитися, щоб створити безпечні програми для автоматизації розумного будинку.

Основні висновки:

- Впроваджуйте надійні заходи безпеки для захисту даних користувачів і конфіденційності
- Зосередьтеся на шифруванні, автентифікації та управлінні дозволами
- Будьте в курсі найкращих практик безпеки та усувайте потенційні вразливості.

2. 9. 4. Інтеграція з голосовими помічниками

Голосові помічники, такі як Amazon Alexa і Google Assistant, набули величезної популярності в екосистемі "розумного будинку". Інтеграція додатків для Android з голосовими помічниками дозволяє користувачам керувати своїми розумними пристроями за допомогою команд природною мовою. Інтегруючи технологію розпізнавання голосу в додаток, розробники можуть забезпечити користувачам зручну роботу без допомоги рук.

Основні висновки:

- Інтеграція додатків для Android з популярними голосовими помічниками
- Керуйте розумними пристроями за допомогою команд природною мовою
- Впроваджуйте технологію розпізнавання голосу для роботи у режимі "вільні руки

2. 9. 5. Енергоефективність та оптимізація

Розробники повинні надавати пріоритет енергоефективності при створенні додатків для Android для автоматизації розумного будинку, щоб

забезпечити оптимальну роботу акумулятора розумних пристроїв. Впровадження енергозберігаючих технологій, таких як push-сповіщення замість безперервного опитування, може значно збільшити час автономної роботи програми. Крім того, оптимізація коду та продуктивності додатку може сприяти безперебійній роботі користувачів.

Основні висновки:

- Зосередьтеся на енергоефективності, щоб оптимізувати роботу акумулятора
- Впроваджуйте енергозберігаючі технології, такі як push-сповіщення
- Оптимізуйте код і продуктивність програми для безперебійної роботи користувачів

РОЗДІЛ 3. РЕАЛІЗАЦІЯ ДОДАТКУ ДО СИСТЕМИ

“РОЗУМНИЙ БУДИНОК”

3. 1. Датчики

3. 1. 1. Датчики вологості в системі "Розумний будинок"

Датчики вологості є важливим компонентом системи "Розумний будинок". Вони відіграють ключову роль у забезпеченні комфорту, здоров’я та енергоефективності житлового приміщення.

Цей код, написаний на мові програмування Kotlin.

import android.os.Bundle

import android.widget.Toast

import androidx.appcompat.app.AppCompatActivity

У цих рядках імпортуються необхідні класи для розробки Android-додатка. `Bundle` використовується для передачі даних між компонентами Android, `Toast` використовується для відображення повідомлень на екрані, а `AppCompatActivity` - це базовий клас для всіх активностей Android, який дозволяє використовувати сучасні можливості бібліотеки підтримки для створення додатків, які сумісні з різними версіями Android.

```
class MainActivity : AppCompatActivity() {
```

Цей рядок оголошує клас `MainActivity`, який є головною активністю (екраном) додатка. Він успадковує функціональність від `AppCompatActivity`.

```
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)

        // Симулюємо отримання даних від датчика вологості
        val currentHumidity = 70 // Значення датчика вологості

        if (currentHumidity > 60) {
            // Включення вентиляції через центральний хаб
            turnOnVentilationThroughHub()
        } else {
            // Вимкнення вентиляції через центральний хаб
            turnOffVentilationThroughHub()
        }
    }
```

Метод `onCreate` викликається при створенні активності. У цьому методі ви визначаєте початковий стан активності, встановлюєте її макет (layout), ініціалізуєте змінні та інші дії, необхідні для початку роботи активності. У даному випадку, симулюється отримання даних від датчика вологості. Якщо значення вологості (`currentHumidity`) більше 60, викликається метод `turnOnVentilationThroughHub()`, в іншому випадку - `turnOffVentilationThroughHub()`.

```
    private fun turnOnVentilationThroughHub() {
        // Код для включення вентиляції через центральний хаб
        Toast.makeText(this, "Вентиляція включена через центральний хаб", Toast.LENGTH_SHORT).show()
    }

    private fun turnOffVentilationThroughHub() {
        // Код для вимкнення вентиляції через центральний хаб
        Toast.makeText(this, "Вентиляція вимкнена через центральний хаб", Toast.LENGTH_SHORT).show()
    }
```

Ці два методи `turnOnVentilationThroughHub()` та `turnOffVentilationThroughHub()` відповідають за включення та вимкнення вентиляції через центральний хаб відповідно. Вони використовують `Toast` для відображення повідомлення про те, що вентиляція була увімкнена або вимкнена через центральний хаб.

Що таке датчики вологості?

Датчики вологості (гігрометри) вимірюють рівень вологості повітря в приміщенні. Вологість визначається як відсоток водяної пари, що міститься в повітрі. Ці датчики можуть бути автономними або інтегрованими в інші розумні пристрої.

Як працюють датчики вологості?

Більшість сучасних датчиків вологості використовують електронні компоненти, такі як ємнісні або резистивні елементи, які змінюють свої електричні властивості залежно від вологості повітря. Сигнали з цих елементів перетворюються в цифрові дані, які передаються до центрального хаба або інших пристроїв для подальшої обробки та аналізу.

Навіщо потрібні датчики вологості?

1. **Комфорт**
    - Оптимальний рівень вологості в приміщенні сприяє створенню комфортних умов для проживання. Занадто висока або низька вологість може викликати дискомфорт.
2. **Здоров'я**
    - Підтримка належного рівня вологості допомагає запобігти розвитку цвілі, грибків та інших шкідливих мікроорганізмів, які можуть викликати алергії та інші захворювання.
3. **Захист будівлі і меблів**
    - Занадто висока вологість може призвести до пошкодження стін, меблів і інших матеріалів, сприяючи їх руйнуванню.

4. **Енергоефективність**
    - Регулювання вологості повітря допомагає оптимізувати роботу систем опалення та кондиціонування, що знижує споживання енергії і витрати на її оплату. Наприклад, вологе повітря здається теплішим, ніж сухе при тій самій температурі, тому контроль вологості може допомогти зменшити витрати на опалення.

Застосування датчиків вологості в системі "Розумний будинок"

1. **Автоматичне регулювання клімату**
    - Датчики вологості можуть інтегруватися з системами опалення, вентиляції та кондиціонування (HVAC), щоб автоматично регулювати рівень вологості, підтримуючи комфортні умови.
2. **Моніторинг та сповіщення**
    - Користувачі можуть отримувати сповіщення про зміну рівня вологості через мобільні додатки. Це дозволяє вчасно вжити заходів, якщо рівень вологості виходить за межі оптимальних значень.
3. **Інтеграція з іншими розумними пристроями**
    - Датчики вологості можуть бути частиною комплексних систем управління кліматом разом з датчиками температури, руху та

освітлення. Наприклад, у разі підвищення вологості, система може автоматично вмикати осушувач повітря або вентиляцію.

4. **Захист від конденсації**
    - У приміщеннях з підвищеним ризиком утворення конденсату (наприклад, ванні кімнати або кухні), датчики вологості можуть контролювати роботу витяжних вентиляторів, запобігаючи утворенню конденсату і цвілі.

3. 1. 2. Температурні датчики в системі "Розумний будинок"

Що таке температурні датчики?

Температурні датчики – це пристрої, які вимірюють температуру навколишнього середовища. Вони можуть бути використані для контролю температури всередині приміщень, на відкритому повітрі або навіть всередині певних пристроїв. Температурні датчики можуть бути автономними або інтегрованими в інші системи для забезпечення комплексного управління кліматом.

import android.os.Bundle

import android.widget.Toast

import androidx.appcompat.app.AppCompatActivity

У цих рядках імпортуються необхідні класи для розробки Android-додатка. `Bundle` використовується для передачі даних між компонентами Android, `Toast` використовується для відображення повідомлень на екрані, а `AppCompatActivity` - це базовий клас для всіх активностей Android, який дозволяє використовувати сучасні можливості бібліотеки підтримки для створення додатків, які сумісні з різними версіями Android.

class MainActivity : AppCompatActivity() {

Цей рядок оголошує клас `MainActivity`, який є головною активністю (екраном) додатка. Він успадковує функціональність від `AppCompatActivity`.

```
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)

        // Симулюємо отримання даних від температурного датчика
        val currentTemperature = 25 // Значення температурного датчика

        if (currentTemperature > 22) {
            // Включення опалення через центральний хаб
            turnOnHeatingThroughHub()
        } else {
            // Вимкнення опалення через центральний хаб
            turnOffHeatingThroughHub()
        }
    }
```

Метод `onCreate` викликається при створенні активності. У цьому методі ви визначаєте початковий стан активності, встановлюєте її макет (layout), ініціалізуєте змінні та інші дії, необхідні для початку роботи активності. У даному випадку, симулюється отримання даних від температурного датчика. Якщо значення температури (`currentTemperature`) більше 22 градусів Цельсія, викликається метод `turnOnHeatingThroughHub()`, в іншому випадку - `turnOffHeatingThroughHub()`.

```
    private fun turnOnHeatingThroughHub() {
```

```
        // Код для включення опалення через центральний хаб

        Toast.makeText(this, "Опалення включено через центральний хаб", Toast.LENGTH_SHORT).show()

    }


    private fun turnOffHeatingThroughHub() {

        // Код для вимкнення опалення через центральний хаб

        Toast.makeText(this, "Опалення вимкнено через центральний хаб", Toast.LENGTH_SHORT).show()

    }
```

Ці два методи `turnOnHeatingThroughHub()` та `turnOffHeatingThroughHub()` відповідають за включення та вимкнення опалення через центральний хаб відповідно. Вони використовують `Toast` для відображення повідомлення про те, що опалення було увімкнене або вимкнене через центральний хаб.

Як працюють температурні датчики?

Більшість сучасних температурних датчиків використовують один з кількох принципів вимірювання температури:

- **Терморезистори (NTC/PTC):** Змінюють свій опір залежно від температури.
- **Термоелементи:** Створюють електричну напругу, пропорційну температурі.
- **Інфрачервоні датчики:** Вимірюють температуру за допомогою інфрачервоного випромінювання.

Ці пристрої перетворюють фізичні зміни в електричні сигнали, які можуть бути оброблені і передані для подальшого використання в системі розумного будинку.

Навіщо потрібні температурні датчики в сучасному житті?

1. **Підтримка комфортного клімату**
    - Температурні датчики дозволяють автоматично підтримувати оптимальну температуру в приміщенні, що забезпечує комфорт для мешканців. Вони можуть керувати системами опалення, вентиляції та кондиціонування повітря (HVAC), забезпечуючи стабільну і комфортну температуру в будинку.

2. **Енергоефективність**
    - Завдяки точному контролю температури, датчики допомагають знизити споживання енергії. Наприклад, система може автоматично знижувати температуру в будинку, коли нікого немає вдома, або регулювати обігрів у залежності від часу доби.
3. **Безпека**
    - Датчики температури можуть виявляти небезпечні температурні умови, такі як перегрівання електроприладів або систем, і автоматично вимикати їх для запобігання пожежам або іншим аварійним ситуаціям.
4. **Захист обладнання і матеріалів**
    - У деяких випадках підтримка певної температури є критично важливою для збереження стану обладнання або матеріалів. Наприклад, серверні кімнати або винні погреби потребують точного контролю температури для правильного функціонування і зберігання.
5. **Здоров'я**
    - Правильна температура в приміщенні важлива для здоров'я мешканців. Занадто висока або низька температура може негативно впливати на самопочуття, викликати захворювання або загострення хронічних захворюваннь.
6. **Автоматизація домашніх процесів**
    - Температурні датчики можуть бути інтегровані з іншими розумними пристроями, такими як штори, вентилятори, обігрівачі та кондиціонери, для створення складних сценаріїв автоматизації. Наприклад, система може автоматично

закривати штори і вмикати кондиціонер, коли температура піднімається вище заданого рівня.

Ціль температурних датчиків

Основна ціль температурних датчиків – забезпечити точний і надійний контроль температури в різних умовах, підвищуючи комфорт, безпеку та енергоефективність.

Вони слугують ключовим компонентом у системах управління кліматом, забезпечуючи автоматичне регулювання температури і запобігаючи виникненню небезпечних ситуацій.

Наскільки сильно потрібні температурні датчики?

Температурні датчики є вкрай важливими в сучасному житті. Вони забезпечують численні переваги:

- **Комфорт:** Підтримка комфортної температури без необхідності ручного регулювання.
- **Економія енергії:** Зменшення витрат на опалення і кондиціонування завдяки автоматичному регулюванню.
- **Безпека:** Запобігання аварійним ситуаціям, пов'язаним з перегрівом або замерзанням.
- **Збереження здоров'я:** Підтримка здорового мікроклімату в приміщенні.
- **Автоматизація:** Інтеграція з іншими розумними пристроями для створення комплексних сценаріїв управління будинком.

3. 1. 3. Освітлювальні датчики

Освітлювальні датчики є важливим компонентом сучасних систем автоматизації будинку, що забезпечують комфорт, енергоефективність та безпеку. Їх використання у системах "Розумний будинок" дозволяє

автоматично керувати освітленням, регулювати його яскравість і включення/вимикання залежно від зовнішніх умов та присутності людей в приміщенні. У даній роботі розглянуто принципи роботи, функції, переваги та перспективи використання освітлювальних датчиків у системах "Розумний будинок".

Принципи роботи освітлювальних датчиків

Освітлювальні датчики (або датчики освітлення) фіксують рівень природного та штучного світла в приміщенні чи на відкритому просторі. Вони використовують фотодіоди або фототранзистори для перетворення світлової енергії в електричні сигнали. Залежно від типу датчика, він може мати аналоговий або цифровий вихід, що передає інформацію про рівень освітленості на центральний хаб системи "Розумний будинок" або безпосередньо на освітлювальні прилади.

```
import android.os.Bundle

import android.widget.Toast

import androidx.appcompat.app.AppCompatActivity


class MainActivity : AppCompatActivity() {


    override fun onCreate(savedInstanceState: Bundle?) {

        super.onCreate(savedInstanceState)

        setContentView(R.layout.activity_main)


        // Симулюємо отримання даних від датчика світла

        val currentLux = 60 // Значення датчика світла


        if (currentLux < 50) {
```

```
            // Включення світла через центральний хаб
            turnOnLightThroughHub()
        } else {
            // Вимкнення світла через центральний хаб
            turnOffLightThroughHub()
        }
    }

    private fun turnOnLightThroughHub() {
        // Код для включення світла через центральний хаб
        Toast.makeText(this, "Освітлення включено через центральний хаб", Toast.LENGTH_SHORT).show()
    }

    private fun turnOffLightThroughHub() {
        // Код для вимкнення світла через центральний хаб
        Toast.makeText(this, "Освітлення вимкнено через центральний хаб", Toast.LENGTH_SHORT).show()
    }
}
```

1) **`import android.os.Bundle`:** Цей рядок імпортує клас `Bundle` з пакету `android.os`. Клас `Bundle` використовується для передачі даних між компонентами Android, такими як активності та фрагменти. У методі `onCreate` об'єкт типу `Bundle` передається, щоб зберегти стан активності при зміні конфігурації, наприклад, при повороті екрану.

2) **`import android.widget.Toast`:** Цей рядок імпортує клас `Toast` з пакету `android.widget`. `Toast` - це коротке повідомлення, яке відображається короткий час на екрані, зазвичай, для показу користувачеві деякої інформації або підтвердження певної дії.
3) **`import androidx.appcompat.app.AppCompatActivity`:** Цей рядок імпортує клас `AppCompatActivity` з пакету `androidx.appcompat.app`. `AppCompatActivity` є базовим класом для всіх активностей у додатку, який дозволяє використовувати сучасні можливості бібліотеки підтримки для створення додатків, які сумісні з різними версіями Android.
4) **`class MainActivity : AppCompatActivity() { ... }`:** Цей рядок визначає клас `MainActivity`, який є основною активністю додатка. Він успадковує функціональність від `AppCompatActivity`.
5) **`override fun onCreate(savedInstanceState: Bundle?) { ... }`:** Цей метод `onCreate` викликається системою при створенні активності. В ньому ви визначаєте початковий стан активності, встановлюєте її макет (заданий у `setContentView`), ініціалізуєте деякі змінні, які використовуються в активності, тощо.
6) **`val currentLux = 60`:** Цей рядок створює змінну `currentLux` і присвоює їй значення 60, що симулює значення датчика світла.
7) **`if (currentLux < 50) { ... } else { ... }`:** Цей рядок використовує умовний оператор `if-else`, щоб перевірити, чи значення датчика світла менше 50. Якщо це так, викликається метод `turnOnLightThroughHub()`, інакше викликається метод `turnOffLightThroughHub()`.
8) **`private fun turnOnLightThroughHub() { ... }`:** Цей метод визначає, як включати світло через центральний хаб. У цьому прикладі використовується об'єкт `Toast` для відображення повідомлення "Освітлення включено через центральний хаб".
9) **`private fun turnOffLightThroughHub() { ... }`:** Цей метод визначає, як вимикати світло через центральний хаб. У цьому прикладі також використовується об'єкт `Toast` для відображення повідомлення "Освітлення вимкнено через центральний хаб".

Функції освітлювальних датчиків

1. **Автоматичне регулювання освітлення**

Однією з основних функцій освітлювальних датчиків є автоматичне регулювання освітлення в приміщеннях. Датчики визначають рівень природного світла і, залежно від встановлених налаштувань, регулюють яскравість штучного освітлення. Це дозволяє забезпечити оптимальні умови для роботи або відпочинку, не потребуючи ручного втручання.

2. **Виявлення присутності**

Освітлювальні датчики можуть бути інтегровані з датчиками руху для виявлення присутності людей у приміщенні. Це дозволяє автоматично вмикати світло при вході людини в кімнату і вимикати його при виході, що сприяє економії енергії і підвищенню зручності користувачів.

3. **Енергоефективність**

Завдяки використанню освітлювальних датчиків, система "Розумний будинок" може значно знизити споживання електроенергії. Автоматичне вимкнення світла в порожніх приміщеннях та регулювання яскравості відповідно до рівня природного освітлення дозволяють зменшити енергетичні витрати і підвищити енергоефективність будинку.

4. **Підвищення безпеки**

Освітлювальні датчики можуть використовуватися для підвищення безпеки як всередині будинку, так і на прилеглій території. Вони можуть автоматично вмикати освітлення в разі виявлення руху біля будинку, що відлякує потенційних зловмисників і забезпечує кращу видимість у темний час доби.

Переваги використання освітлювальних датчиків

1. **Комфорт**

Автоматичне керування освітленням забезпечує комфортні умови для проживання та роботи, дозволяючи користувачам не турбуватися про ввімкнення або вимкнення світла. Датчики регулюють освітлення так, щоб забезпечити оптимальний рівень світла в будь-який час доби.

2. **Економія ресурсів**

Завдяки ефективному використанню енергії, освітлювальні датчики допомагають знизити витрати на електроенергію. Це особливо важливо в умовах зростання цін на енергоносії та необхідності збереження природних ресурсів.

3. **Зручність**

Інтеграція освітлювальних датчиків у систему "Розумний будинок" дозволяє створювати складні сценарії автоматизації, що підвищує зручність користування.

3. 1. 4. Датчики руху

Що таке датчики руху?

Датчики руху – це пристрої, які виявляють фізичний рух в певній зоні. Вони можуть використовувати різні технології для виявлення руху, включаючи інфрачервоне (PIR), ультразвукове, мікрохвильове і лазерне випромінювання. Датчики руху можуть бути інтегровані в системи освітлення, безпеки, опалення та інші компоненти розумного будинку для забезпечення автоматизації і підвищення ефективності їх роботи.

Виклики та перспективи розвитку

Незважаючи на численні переваги, використання датчиків руху має і деякі виклики. Це, зокрема, можливість хибних спрацьовувань, складність встановлення і налаштування, а також висока вартість деяких рішень. Проте, з розвитком технологій ці проблеми поступово вирішуються, що робить датчики руху все більш доступними і надійними.

Навіщо потрібні датчики руху?

1. **Автоматизація освітлення** Датчики руху широко використовуються для автоматичного вмикання і вимикання освітлення в приміщеннях. Це дозволяє забезпечити комфорт для мешканців, оскільки світло вмикається лише тоді, коли в приміщенні є люди, і вимикається, коли приміщення порожнє. Це

також сприяє економії енергії, зменшуючи витрати на електроенергію.

2. **Системи безпеки** Датчики руху є ключовим компонентом систем безпеки, виявляючи рух в зоні контролю. Вони можуть активувати сигналізацію, надсилати сповіщення користувачам або службі безпеки у разі виявлення несанкціонованого доступу до будинку. Це допомагає запобігти крадіжкам і підвищити загальний рівень безпеки.
3. **Автоматизація клімат-контролю** Датчики руху можуть використовуватися для регулювання роботи систем опалення, вентиляції та кондиціонування повітря (HVAC).
4. **Підвищення комфорту** Інтеграція датчиків руху з іншими розумними пристроями дозволяє створювати комплексні сценарії автоматизації, які підвищують комфорт мешканців. Наприклад, датчик може включати музику або відкривати штори при вході людини в кімнату.
5. **Енергозбереження** Використання датчиків руху дозволяє значно знизити споживання енергії, оскільки електроприлади та освітлення працюють тільки тоді, коли це необхідно. Це не тільки зменшує витрати на електроенергію, але й сприяє збереженню природних ресурсів.

Функціональні можливості датчиків руху

1. **Регулювання чутливості** Датчики руху можуть мати можливість налаштування чутливості, що дозволяє регулювати їх роботу залежно від конкретних потреб користувачів. Це дозволяє уникати хибних спрацьовувань і забезпечує більш точну роботу пристроїв.
2. **Інтеграція з іншими системами** Сучасні датчики руху можуть бути інтегровані з іншими компонентами системи "Розумний будинок", такими як освітлення, системи безпеки, клімат-контроль та інші. Це дозволяє створювати комплексні сценарії автоматизації для підвищення ефективності і комфорту.
3. **Дистанційне керування і моніторинг** Завдяки інтеграції з мобільними додатками, користувачі можуть дистанційно контролювати роботу датчиків руху, отримувати сповіщення про виявлений рух і змінювати налаштування датчиків в режимі реального часу.

4. **Запам'ятовування шаблонів руху** Деякі датчики руху мають можливість запам'ятовування шаблонів руху, що дозволяє їм адаптуватися до звичайних дій мешканців і знижувати кількість хибних спрацьовувань.

Переваги використання датчиків руху

1. **Підвищення безпеки** Автоматичне виявлення руху і активація сигналізації допомагає запобігти крадіжкам і підвищити загальний рівень безпеки будинку.
2. **Економія енергії** Використання датчиків руху для автоматизації освітлення і клімат-контролю значно знижує споживання електроенергії, що сприяє економії коштів і збереженню природних ресурсів.
3. **Підвищення комфорту і зручності** Автоматизація побутових процесів за допомогою датчиків руху робить життя більш комфортним і зручним, оскільки зменшує потребу в ручному керуванні освітленням і іншими системами.
4. **Інтеграція з іншими розумними пристроями** Датчики руху можуть бути інтегровані з іншими компонентами системи "Розумний будинок", створюючи комплексні рішення для автоматизації і підвищення ефективності житлового простору.

3. 1. 5. Датчики газу та диму

Системи "Розумний будинок" надають можливість максимально забезпечити комфорт, безпеку та енергоефективність у приватних житлових приміщеннях. Одним із ключових елементів безпеки таких систем є датчики диму та газу. Вони виконують важливу функцію, надаючи можливість вчасно реагувати на небезпечні ситуації, пов'язані з витоком газів або появою диму.

Навіщо потрібні датчики диму та газу?

1. **Безпека мешканців**
   Датчики диму та газу є необхідними для забезпечення безпеки мешканців будинку. Вони виявляють небезпечні речовини у повітрі, такі як кількісні та димові гази, попереджаючи про можливе виникнення пожежі або інших небезпечних ситуацій.
2. **Попередження про пожежу**
   Датчики диму здатні виявляти навіть слабкий дим, що може свідчити про початок пожежі. Вчасне сповіщення мешканців про пожежу дозволяє запобігти поширенню вогню та максимально швидко реагувати.
3. **Виявлення витоку газів**
   Датчики газу, такі як метан або пропан, допомагають вчасно виявляти їх витік у повітря. Це дозволяє запобігти отруєнням та вибухам, що можуть виникнути внаслідок неправильного використання газового обладнання.
4. **Інтеграція з системою "Розумний будинок"**

   Датчики диму та газу можуть бути інтегровані в систему "Розумний будинок", що надає можливість автоматично вмикати сигналізацію або надсилати сповіщення на мобільні пристрої мешканців у разі виявлення небезпеки.

5. **Енергоефективність та економія**

   Вчасне виявлення витоку газів чи початку пожежі дозволяє ефективно втручатися та уникнути великих збитків, а також забезпечує економію на енергоресурсах та ремонтних роботах.

Застосування датчиків диму та газу в сучасному житті

Датчики диму та газу є невід'ємною складовою систем "Розумний будинок". Вони дозволяють максимально забезпечити безпеку та комфорт у побуті, реагуючи на небезпечні ситуації вчасно та ефективно.

```
import android.os.Bundle
```

```
import android.widget.Toast

import androidx.appcompat.app.AppCompatActivity
```

У цих рядках імпортуються необхідні класи для розробки Android-додатка. Bundle використовується для передачі даних між компонентами Android, Toast використовується для відображення повідомлень на екрані, а AppCompatActivity - це базовий клас для всіх активностей Android, який дозволяє використовувати сучасні можливості бібліотеки підтримки для створення додатків, які сумісні з різними версіями Android.

```
class MainActivity : AppCompatActivity() {
```

Цей рядок оголошує клас `MainActivity`, який є головною активністю (екраном) додатка. Він успадковує функціональність від `AppCompatActivity`.

```
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)

        // Симулюємо отримання даних від датчиків газу та диму
        val isGasDetected = false // Дані від датчика газу
        val isSmokeDetected = true // Дані від датчика диму

        if (isGasDetected || isSmokeDetected) {
            // Відсилання сигналу про небезпеку через центральний хаб
            sendAlertThroughHub()
        } else {
            // Сигнал про небезпеку не потрібний
```

```
            Toast.makeText(this, "Небезпеки не виявлено",
Toast.LENGTH_SHORT).show()

        }

    }
```

Метод `onCreate` викликається при створенні активності. У цьому методі ви визначаєте початковий стан активності, встановлюєте її макет (layout), ініціалізуєте змінні та інші дії, необхідні для початку роботи активності. У даному випадку, симулюється отримання даних від датчиків газу та диму. Якщо хоча б один з цих датчиків виявляє небезпеку (змінні `isGasDetected` або `isSmokeDetected`), викликається метод `sendAlertThroughHub()`, в іншому випадку відображається повідомлення, що небезпека не виявлена.

```
    private fun sendAlertThroughHub() {

        // Код для відсилання сигналу про небезпеку через центральний хаб

        Toast.makeText(this, "Небезпека виявлена! Сигнал надіслано через
центральний хаб", Toast.LENGTH_SHORT).show()

    }
```

Цей метод `sendAlertThroughHub()` відповідає за відсилання сигналу про небезпеку через центральний хаб. Використовується `Toast` для відображення повідомлення про те, що небезпека була виявлена та сигнал був надісланий через центральний хаб.

3. 2. Пристрої управління

3. 2. 1. Розумні лампи

У сучасному світі технології надають нам набагато більше можливостей у керуванні різними аспектами нашого життя. Однією з інноваційних розробок у цьому контексті є розумні лампи. Ці технологічно продумані пристрої вписуються в системи "Розумний будинок", роблячи життя мешканців більш комфортним, ефективним та енергоефективним.

Навіщо потрібні розумні лампи?

1. **Ефективне керування освітленням**

   Розумні лампи надають користувачам можливість керувати освітленням за допомогою мобільного додатку чи голосових команд. Це дозволяє регулювати яскравість, колір та режими освітлення в будь-який час, навіть якщо ви не знаходитесь вдома.

2. **Створення атмосферних ефектів**

   За допомогою розумних ламп можна створювати різноманітні атмосферні ефекти, використовуючи різні кольори та режими освітлення. Це особливо корисно для створення романтичного настрою, робочого середовища або для релаксації під час перегляду фільмів.

3. **Енергоефективність**

   Розумні лампи зазвичай мають функцію автоматичного вимкнення, яка дозволяє ефективно використовувати електроенергію. Вони можуть вимикатися автоматично, коли ніхто не перебуває в приміщенні або встановлено певний графік вимикання/увімкнення.

4. **Інтеграція з системами "Розумний будинок"**

   Розумні лампи легко інтегруються з іншими пристроями системи "Розумний будинок", такими як датчики руху або системи безпеки.

Це дозволяє створювати різноманітні автоматизовані сценарії освітлення, забезпечуючи більшу зручність та безпеку.

5. **Віддалене керування**

   Завдяки підключенню до мережі Wi-Fi, розумні лампи можна керувати віддалено з будь-якої точки світу за допомогою смартфона чи іншого пристрою з доступом до Інтернету. Це надає більшу гнучкість та зручність у керуванні освітленням.

Застосування розумних ламп в сучасному житті

Розумні лампи стають все більш популярними в сучасному житті та широко використовуються в приватних будинках, офісах та громадських приміщеннях. Вони надають користувачам більшу гнучкість, ефективність та зручність у керуванні освітленням.

Розумні лампи стають все більш популярними в сучасному житті та широко використовуються в приватних будинках, офісах та громадських приміщеннях. Вони надають користувачам більшу гнучкість, ефективність та зручність у керуванні освітленням.

import android.os.Bundle

import android.widget.Toast

import androidx.appcompat.app.AppCompatActivity

У цих рядках виконується імпорт необхідних класів для розробки Android-додатка. `Bundle` використовується для передачі даних між компонентами Android, `Toast` використовується для відображення коротких повідомлень на екрані, а `AppCompatActivity` - це базовий клас для всіх активностей Android, який дозволяє використовувати сучасні можливості бібліотеки підтримки для створення додатків, сумісних з різними версіями Android.

class MainActivity : AppCompatActivity() {

У цьому рядку визначається клас `MainActivity`, який є головною активністю (екраном) додатка. Він успадковує функціональність від `AppCompatActivity`.

```
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)

        // Симулюємо отримання стану розумних ламп
        val isLightOn = false // Стан лампи (увімкнено/вимкнено)

        if (isLightOn) {
            // Вимкнення розумної лампи через центральний хаб
            turnOffSmartLightThroughHub()
        } else {
            // Увімкнення розумної лампи через центральний хаб
            turnOnSmartLightThroughHub()
        }
    }
```

Метод `onCreate` викликається при створенні активності. У цьому методі визначається початковий стан активності, встановлюється її макет (layout), ініціалізуються змінні та інші дії, необхідні для початку роботи активності. У даному випадку, симулюється отримання стану розумних ламп (змінна `isLightOn`). Якщо лампа увімкнена, викликається метод `turnOffSmartLightThroughHub()` для вимкнення лампи через центральний хаб. Якщо лампа вимкнена, викликається метод `turnOnSmartLightThroughHub()` для увімкнення лампи через центральний хаб.

```
    private fun turnOnSmartLightThroughHub() {
        // Код для увімкнення розумної лампи через центральний хаб
```

```
        Toast.makeText(this, "Розумну лампу увімкнено через центральний хаб", Toast.LENGTH_SHORT).show()
    }

    private fun turnOffSmartLightThroughHub() {
        // Код для вимкнення розумної лампи через центральний хаб
        Toast.makeText(this, "Розумну лампу вимкнено через центральний хаб", Toast.LENGTH_SHORT).show()
    }
```

Ці методи `turnOnSmartLightThroughHub()` та `turnOffSmartLightThroughHub()` відповідають за ввімкнення та вимкнення розумних ламп через центральний хаб відповідно. Вони використовують `Toast` для відображення повідомлення на екрані про те, що лампу було увімкнено або вимкнено через центральний хаб.

3. 2. 2. Розумні термостати

Одним із ключових компонентів цієї системи є розумні термостати. Ці пристрої дозволяють автоматично контролювати клімат у приміщеннях, забезпечуючи оптимальні умови для мешканців та знижуючи витрати на енергію.

Навіщо потрібні розумні термостати?

1. **Енергоефективність**

   Розумні термостати дозволяють значно знизити витрати на енергію, автоматично регулюючи температуру в приміщеннях на основі присутності людей, часу доби та інших параметрів. Це допомагає оптимізувати використання енергії, зменшуючи витрати на опалення та кондиціонування.

2. **Комфорт**

Завдяки розумним термостатам користувачі можуть створювати комфортні умови в будинку, налаштовуючи температуру відповідно до своїх потреб. Пристрої можуть автоматично змінювати налаштування температури в залежності від часу доби або присутності людей у приміщенні.

3. **Дистанційне керування**

   Розумні термостати дозволяють користувачам керувати температурою в будинку за допомогою мобільних додатків або голосових команд. Це забезпечує зручність та гнучкість у керуванні кліматом, навіть коли користувачі знаходяться поза межами будинку.

4. **Інтеграція з іншими системами**

   Розумні термостати можуть інтегруватися з іншими пристроями системи "Розумний будинок", такими як датчики руху, розумні лампи та системи безпеки. Це дозволяє створювати складні сценарії автоматизації, які підвищують ефективність та комфорт у будинку.

5. **Аналітика та звітування**

   Розумні термостати надають детальну інформацію про споживання енергії та ефективність систем опалення та кондиціонування. Це дозволяє користувачам аналізувати та оптимізувати свої витрати, знаходячи найбільш ефективні способи використання енергії.

Застосування розумних термостатів у сучасному житті

Розумні термостати стають все більш популярними серед користувачів завдяки своїм численним перевагам. Вони широко використовуються як у приватних будинках, так і в комерційних приміщеннях, забезпечуючи ефективне керування кліматом, економію енергії та підвищення комфорту.

Розумні термостати є важливим елементом системи "Розумний будинок", що дозволяє ефективно керувати кліматом у приміщеннях, забезпечуючи комфорт та енергоефективність. Вони надають користувачам можливість зручно та гнучко контролювати температуру, інтегруючись з іншими розумними пристроями для створення комплексних систем автоматизації.

Використання розумних термостатів сприяє оптимізації споживання енергії та підвищенню якості життя мешканців.

```
import android.os.Bundle

import android.widget.Toast

import androidx.appcompat.app.AppCompatActivity


class MainActivity : AppCompatActivity() {


    override fun onCreate(savedInstanceState: Bundle?) {

        super.onCreate(savedInstanceState)

        setContentView(R.layout.activity_main)


        // Симулюємо отримання даних від розумного термостата

        val currentTemperature = 22 // Поточна температура


        if (currentTemperature < 20) {

            // Включення опалення через центральний хаб

            turnOnHeatingThroughHub()

        } else if (currentTemperature > 25) {

            // Включення кондиціонування через центральний хаб

            turnOnCoolingThroughHub()

        } else {

            // Вимкнення опалення та кондиціонування через центральний хаб
```

```
            turnOffHVACThroughHub()
        }
    }

    private fun turnOnHeatingThroughHub() {
        // Код для включення опалення через центральний хаб
        Toast.makeText(this, "Опалення включено через центральний хаб", Toast.LENGTH_SHORT).show()
    }

    private fun turnOnCoolingThroughHub() {
        // Код для включення кондиціонування через центральний хаб
        Toast.makeText(this, "Кондиціонування включено через центральний хаб", Toast.LENGTH_SHORT).show()
    }

    private fun turnOffHVACThroughHub() {
        // Код для вимкнення опалення та кондиціонування через центральний хаб
        Toast.makeText(this, "Опалення та кондиціонування вимкнено через центральний хаб", Toast.LENGTH_SHORT).show()
    }
}
```

У цьому коді ми маємо основну активність `MainActivity`, яка успадковує `AppCompatActivity`. Вона встановлює макет `activity_main` у методі `onCreate`.

В розділі `onCreate` ми симулюємо отримання даних від розумного термостата. Температура зберігається у змінній `currentTemperature`. Залежно від цієї температури, ми вирішуємо, чи потрібно включити опалення, кондиціонування або вимкнути систему опалення та кондиціонування через центральний хаб.

Методи `turnOnHeatingThroughHub()`, `turnOnCoolingThroughHub()` і `turnOffHVACThroughHub()` відповідають за відповідні дії з включенням опалення, кондиціонування або їх вимкненням через центральний хаб. Кожен з цих методів викликає `Toast.makeText`, щоб відобразити коротке повідомлення на екрані.

3. 2. 3. Розумні розетки

Розумні розетки є важливим компонентом системи "Розумний будинок", що дозволяє ефективно керувати електроприладами, забезпечуючи комфорт, безпеку та енергоефективність. Вони надають користувачам можливість дистанційного керування, автоматизації побутових процесів та моніторингу споживання енергії, що сприяє оптимізації витрат на електроенергію та підвищенню рівня комфорту в житлових приміщеннях. Використання розумних розеток допомагає створити більш сучасне, безпечне та ефективне середовище для життя та роботи.

Навіщо потрібні розумні розетки?

1. **Дистанційне керування**

Розумні розетки дозволяють користувачам керувати електроприладами за допомогою мобільних додатків або голосових команд. Це забезпечує зручність у використанні та можливість контролювати прилади навіть на відстані, що особливо корисно у випадку необхідності вимкнення забутого ввімкненого пристрою.

2. **Енергоефективність**

Завдяки розумним розеткам можна знизити витрати на електроенергію, автоматично вимикаючи електроприлади, які не використовуються. Вони можуть бути запрограмовані на вимкнення у визначений час або при відсутності руху в приміщенні, що допомагає зменшити енергоспоживання та знизити рахунки за електроенергію.

3. **Підвищення безпеки**

Розумні розетки допомагають підвищити безпеку в будинку, запобігаючи перегріванню або коротким замиканням електроприладів. Вони можуть автоматично вимикати прилади у випадку виявлення несправності або перевантаження, що знижує ризик пожеж та інших небезпечних ситуацій.

4. **Автоматизація побутових процесів**

Розумні розетки можуть бути інтегровані з іншими пристроями системи "Розумний будинок", такими як датчики руху, розумні лампи та термостати. Це дозволяє створювати складні сценарії автоматизації, наприклад, автоматичне ввімкнення кавоварки вранці або вимкнення всіх електроприладів при виході з будинку.

5. **Моніторинг споживання енергії**

Розумні розетки надають можливість моніторингу споживання електроенергії кожним приладом. Це дозволяє користувачам аналізувати витрати та знаходити способи оптимізації енергоспоживання, роблячи використання електроенергії більш раціональним та економним.

Застосування розумних розеток у сучасному житті

Розумні розетки стали невід'ємною частиною сучасного життя завдяки своїм численним перевагам. Вони широко використовуються у приватних будинках, офісах та комерційних приміщеннях, забезпечуючи зручність, безпеку та ефективність у керуванні електроприладами.

```
import android.os.Bundle

import android.widget.Toast
```

```
import androidx.appcompat.app.AppCompatActivity

class MainActivity : AppCompatActivity() {

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)

        // Симулюємо отримання даних від розумної розетки
        val isSocketOn = false // Стан розумної розетки (увімкнено/вимкнено)

        if (isSocketOn) {
            // Вимкнення розумної розетки через центральний хаб
            turnOffSocketThroughHub()
        } else {
            // Увімкнення розумної розетки через центральний хаб
            turnOnSocketThroughHub()
        }
    }

    private fun turnOnSocketThroughHub() {
        // Код для увімкнення розумної розетки через центральний хаб
        Toast.makeText(this, "Розумну розетку увімкнено через центральний хаб", Toast.LENGTH_SHORT).show()
```

```
    }

    private fun turnOffSocketThroughHub() {
        // Код для вимкнення розумної розетки через центральний хаб
        Toast.makeText(this, "Розумну розетку вимкнено через центральний хаб", Toast.LENGTH_SHORT).show()
    }
}
```

цьому коді ми маємо основну активність `MainActivity`, яка успадковує `AppCompatActivity`. Вона встановлює макет `activity_main` у методі `onCreate`.

В розділі `onCreate` ми симулюємо отримання даних від розумної розетки щодо її поточного стану. Стан розетки (увімкнено/вимкнено) зберігається у змінній `isSocketOn`. Залежно від цього стану, ми вирішуємо, чи потрібно вмикати або вимикати розумну розетку через центральний хаб.

Методи `turnOnSocketThroughHub()` і `turnOffSocketThroughHub()` відповідають за відповідні дії з увімкнення або вимикання розумної розетки через центральний хаб. Кожен з цих методів також викликає `Toast.makeText`, щоб відобразити коротке повідомлення на екрані.

3. 2. 4. Розумні дверні замки

Розумні дверні замки є важливим компонентом системи "Розумний будинок", що дозволяє ефективно керувати доступом до приміщень, забезпечуючи безпеку та комфорт. Вони надають користувачам можливість дистанційного керування, автоматизації процесів доступу та моніторингу історії відвідувань, що сприяє підвищенню рівня безпеки та зручності у житлових та комерційних приміщеннях. Використання розумних дверних замків допомагає створити більш сучасне, безпечне та комфортне середовище для життя та роботи.

Навіщо потрібні розумні дверні замки?

1. **Підвищення безпеки**

Розумні дверні замки забезпечують високий рівень безпеки, використовуючи сучасні технології шифрування та аутентифікації. Вони можуть бути відкриті за допомогою мобільного додатку, біометричних даних (відбитків пальців), кодів доступу або карток. Це дозволяє знизити ризик несанкціонованого доступу та підвищити безпеку приміщень.

2. **Дистанційне керування**

Однією з основних переваг розумних дверних замків є можливість дистанційного керування. Користувачі можуть відкривати та закривати двері за допомогою мобільного додатку, перебуваючи у будь-якій точці світу. Це особливо корисно у випадку необхідності надати доступ до приміщення гостям або сервісним працівникам, коли власники будинку відсутні.

3. **Автоматизація доступу**

Розумні дверні замки дозволяють автоматизувати процеси доступу до приміщень. Вони можуть бути інтегровані з іншими пристроями системи "Розумний будинок", такими як камери спостереження та системи сигналізації. Це дозволяє створювати складні сценарії автоматизації, наприклад, автоматичне відмикання дверей при наближенні власника або замикання дверей при виході з будинку.

4. **Історія доступу**

Розумні дверні замки надають можливість відстежувати історію доступу до приміщення. Користувачі можуть отримувати повідомлення про відкриття та закриття дверей, а також переглядати записи про всі спроби доступу. Це забезпечує додатковий рівень контролю та безпеки, дозволяючи оперативно реагувати на підозрілі дії.

5. **Зручність у використанні**

Використання розумних дверних замків забезпечує зручність та простоту у повсякденному житті. Відсутність необхідності носити ключі,

можливість швидкого та безпечного доступу до приміщення, а також інтеграція з іншими розумними пристроями роблять їх незамінними у сучасному будинку.

Застосування розумних дверних замків у сучасному житті

Розумні дверні замки стають все більш популярними серед користувачів завдяки своїм численним перевагам. Вони широко використовуються у приватних будинках, офісах та комерційних приміщеннях, забезпечуючи високий рівень безпеки та зручності.

`import` інструкції використовуються для імпортування необхідних класів.

```
import android.os.Bundle

import android.widget.Toast
```

`MainActivity` - це основний клас додатку, який успадковує клас `AppCompatActivity`, щоб мати можливість використовувати функціонал для створення додатків на платформі Android.

```
class MainActivity : AppCompatActivity() {
```

У методі `onCreate` встановлюється макет `activity_main`.

```
override fun onCreate(savedInstanceState: Bundle?) {

    super.onCreate(savedInstanceState)

    setContentView(R.layout.activity_main)
```

Змінна `isDoorLocked` симулює отримання стану розумного дверного замка про його поточний стан (заблоковано чи розблоковано).

```
// Симулюємо отримання стану розумного дверного замка

val isDoorLocked = false // Стан дверного замка
(заблоковано/розблоковано)
```

Умовна конструкція `if-else` перевіряє стан дверного замка. Якщо замок заблокований (`isDoorLocked == true`), викликається метод

`unlockDoorThroughHub()`, що розблоковує дверний замок через центральний хаб. Інакше викликається метод `lockDoorThroughHub()`, що блокує дверний замок через центральний хаб.

```
if (isDoorLocked) {

    // Розблокування дверного замка через центральний хаб

    unlockDoorThroughHub()

} else {

    // Блокування дверного замка через центральний хаб

    lockDoorThroughHub()

}
```

Метод `lockDoorThroughHub()` відповідає за блокування розумного дверного замка. Він викликає `Toast.makeText`, щоб відобразити повідомлення про блокування замка через центральний хаб.

```
private fun lockDoorThroughHub() {

    // Код для блокування дверного замка через центральний хаб

    Toast.makeText(this, "Дверний замок заблоковано через центральний хаб", Toast.LENGTH_SHORT).show()

}
```

Метод `unlockDoorThroughHub()` відповідає за розблокування розумного дверного замка. Він також викликає `Toast.makeText`, щоб відобразити повідомлення про розблокування замка через центральний хаб.

```
private fun unlockDoorThroughHub() {

    // Код для розблокування дверного замка через центральний хаб

    Toast.makeText(this, "Дверний замок розблоковано через центральний хаб", Toast.LENGTH_SHORT).show()
```

}

Отже, код управляє станом розумного дверного замка, в залежності від його поточного стану відповідно розблоковує або блокує дверний замок через центральний хаб.

3. 2. 5. Камери спостереження

Камери спостереження є незамінним елементом системи "Розумний будинок", що забезпечує високий рівень безпеки та комфорту для мешканців. Вони надають можливість цілодобового моніторингу, дистанційного керування та інтеграції з іншими системами автоматизації. Використання камер спостереження сприяє запобіганню злочинам, підвищенню рівня безпеки та забезпеченню захисту майна та мешканців. У сучасному світі, де питання безпеки є надзвичайно важливим, камери спостереження стають необхідним атрибутом кожного будинку та комерційного приміщення.

Навіщо потрібні камери спостереження?

1. **Підвищення безпеки**

Камери спостереження забезпечують цілодобовий моніторинг території та приміщень будинку, дозволяючи виявляти та запобігати потенційним загрозам. Вони допомагають виявити підозрілих осіб, зафіксувати несанкціоновані проникнення та інші підозрілі дії, що сприяє швидкій реакції на можливі інциденти.

2. **Запобігання злочинам**

Наявність камер спостереження часто діє як стримуючий фактор для злочинців. Вони знають, що їхні дії можуть бути зафіксовані та використані як докази, що значно знижує ймовірність вчинення злочинів. Таким чином, камери спостереження сприяють зниженню рівня злочинності та підвищенню загальної безпеки.

3. **Дистанційний моніторинг**

Сучасні камери спостереження надають можливість дистанційного моніторингу у реальному часі за допомогою мобільних додатків або

комп'ютера. Це дозволяє власникам будинку контролювати ситуацію навіть на відстані, забезпечуючи спокій та впевненість у безпеці свого житла.

4. **Автоматизація процесів безпеки**

Камери спостереження можуть бути інтегровані з іншими системами "Розумного будинку", такими як датчики руху, розумні дверні замки та системи сигналізації. Це дозволяє створювати комплексні сценарії автоматизації, наприклад, автоматичне ввімкнення камер при виявленні руху або надсилання сповіщень у разі виявлення підозрілих дій.

5. **Збереження відеозаписів**

Камери спостереження дозволяють зберігати відеозаписи для подальшого аналізу. Це може бути корисно у випадку розслідування інцидентів, надання доказів правоохоронним органам або аналізу поведінки для підвищення рівня безпеки.

6. **Контроль доступу та моніторинг активності**

Камери спостереження допомагають контролювати доступ до будинку, відстежувати пересування людей та транспортних засобів на території. Це дозволяє своєчасно виявляти та реагувати на підозрілу активність, забезпечуючи захист мешканців та майна.

Застосування камер спостереження у сучасному житті

Камери спостереження широко використовуються не лише у приватних будинках, але й у комерційних та громадських приміщеннях.

Вони забезпечують ефективний моніторинг та безпеку, що є критично важливим у сучасному світі. Їх застосування включає охорону житлових приміщень, офісів, магазинів, громадських закладів та інших об'єктів.

1. Підключення необхідних бібліотек і класів:

```
import android.os.Bundle

import android.widget.Toast

import androidx.appcompat.app.AppCompatActivity
```

2. Створення активності MainActivity, яка розширює клас AppCompatActivity:

```
class MainActivity : AppCompatActivity() {
```

3. Перевизначення методу `onCreate`, який викликається при створенні активності:

```
override fun onCreate(savedInstanceState: Bundle?) {

    super.onCreate(savedInstanceState)

    setContentView(R.layout.activity_main)
```

У цьому методі викликається метод `setContentView`, який встановлює вміст для активності з розмітки `activity_main`.

4. Симуляція отримання стану камери спостереження:

```
    // Симулюємо отримання стану камери спостереження

    val isCameraActive = true // Стан камери (увімкнено/вимкнено)
```

5. Умовна конструкція `if-else`, яка перевіряє стан камери і викликає відповідний метод в залежності від стану:

```
    if (isCameraActive) {

        // Вимкнення камери спостереження через центральний хаб

        turnOffCameraThroughHub()

    } else {

        // Увімкнення камери спостереження через центральний хаб
```

```
        turnOnCameraThroughHub()
    }
```

6. Метод `turnOnCameraThroughHub()`, який відповідає за увімкнення камери:

```
private fun turnOnCameraThroughHub() {
    // Код для увімкнення камери спостереження через центральний хаб
    Toast.makeText(this, "Камеру спостереження увімкнено через центральний хаб", Toast.LENGTH_SHORT).show()
    // Додатковий код для управління камерою та отримання відеопотоку
}
```

7. Метод `turnOffCameraThroughHub()`, який відповідає за вимкнення камери:

```
private fun turnOffCameraThroughHub() {
    // Код для вимкнення камери спостереження через центральний хаб
    Toast.makeText(this, "Камеру спостереження вимкнено через центральний хаб", Toast.LENGTH_SHORT).show()
    // Додатковий код для управління камерою та зупинки відеопотоку
}
```

Цей код демонструє простий спосіб симуляції управління камерою спостереження через центральний хаб на платформі Android. При реальній реалізації вам потрібно буде використовувати спеціалізовані бібліотеки та сервіси для роботи з камерою та центральним хабом.

3. 2. 6. Побутові прилади

Розумні побутові прилади є важливим елементом системи "Розумний будинок", що дозволяє забезпечити високий рівень комфорту, ефективності та безпеки у житлових приміщеннях. Вони надають можливість автоматизації побутових процесів, економії енергії та ресурсів, а також підвищення рівня безпеки. Використання розумних побутових приладів допомагає створити сучасне, комфортне та безпечне середовище для життя, що відповідає потребам та вимогам сучасного суспільства.

Навіщо потрібні розумні побутові прилади?

1. **Комфорт та зручність**

Розумні побутові прилади забезпечують високий рівень комфорту завдяки можливості дистанційного керування та автоматизації процесів. Власники можуть керувати приладами за допомогою мобільних додатків або голосових команд, що дозволяє значно спростити повсякденні завдання. Наприклад, розумні холодильники можуть повідомляти про необхідність поповнення запасів продуктів, а розумні пральні машини – автоматично обирати оптимальний режим прання.

2. **Енергоефективність**

Однією з головних переваг розумних побутових приладів є їх висока енергоефективність. Вони можуть автоматично вимикатися після завершення роботи або переходити в режим енергозбереження, що допомагає знизити витрати на електроенергію. Деякі прилади також можуть аналізувати споживання енергії та пропонувати способи його оптимізації.

3. **Безпека**

Розумні побутові прилади підвищують рівень безпеки у будинку. Вони можуть бути оснащені датчиками для виявлення несправностей, таких як витоки води або газу, перегрів чи коротке замикання. Це дозволяє

своєчасно виявити та усунути потенційні загрози, запобігаючи аварійним ситуаціям.

4. **Автоматизація побутових процесів**

Завдяки інтеграції з іншими пристроями системи "Розумний будинок", розумні побутові прилади дозволяють автоматизувати побутові процеси. Наприклад, розумні кавоварки можуть автоматично готувати каву вранці, а розумні пилососи – виконувати прибирання у визначений час. Це забезпечує значну економію часу та підвищує ефективність домашніх справ.

5. **Моніторинг та діагностика**

Розумні побутові прилади надають можливість моніторингу та діагностики свого стану. Користувачі можуть отримувати повідомлення про необхідність обслуговування або заміни компонентів, що дозволяє підтримувати прилади у належному стані та продовжувати їхній строк служби.

Застосування розумних побутових приладів у сучасному житті

Розумні побутові прилади стають все більш популярними серед споживачів завдяки своїм численним перевагам. Вони широко використовуються у приватних будинках та квартирах, забезпечуючи комфорт, ефективність та безпеку у повсякденному житті. Їх застосування включає розумні холодильники, пральні машини, духові шафи, посудомийні машини, пилососи, кавоварки та інші пристрої, які значно полегшують виконання побутових завдань.

3. 3. Центральний хаб

Центральний хаб є ключовим компонентом системи "Розумний будинок", який виконує роль головного координаційного центру для всіх підключених пристроїв. Він забезпечує зв’язок між різними компонентами системи, такими як розумні дверні замки, датчики руху, камери спостереження, розумні термостати, лампи, розетки та побутові прилади. Центральний хаб дозволяє їм взаємодіяти один з одним та з користувачем, створюючи єдину інтегровану екосистему.

Основна функція центрального хабу полягає у забезпеченні безперебійної роботи всіх підключених пристроїв, координації їх дій та управлінні ними. Він забезпечує обробку даних, що надходять від різних датчиків та пристроїв, та приймає рішення щодо виконання певних дій. Наприклад, хаб може автоматично ввімкнути освітлення при виявленні руху, або регулювати температуру в приміщенні залежно від наявності людей та зовнішніх умов.

Центральний хаб також дозволяє користувачам керувати всіма пристроями з єдиного інтерфейсу, зазвичай через мобільний додаток або веб-портал. Це значно спрощує управління системою "Розумний будинок", оскільки всі функції зібрані в одному місці. Користувачі можуть налаштовувати сценарії автоматизації, отримувати сповіщення про події, контролювати енергоспоживання та багато іншого.

Однією з важливих функцій центрального хабу є забезпечення безпеки. Він відповідає за шифрування даних, захист від несанкціонованого доступу та забезпечення надійної роботи системи. У разі виявлення загроз або несправностей, хаб може автоматично повідомити користувача та вжити необхідних заходів для забезпечення безпеки будинку.

Центральний хаб також дозволяє інтегрувати різні протоколи зв’язку та стандарти, такі як Wi-Fi, Zigbee, Z-Wave, Bluetooth та інші. Це дозволяє підключати до системи різноманітні пристрої від різних виробників, забезпечуючи їхню сумісність та взаємодію.

Загалом, центральний хаб є невід'ємною частиною системи "Розумний будинок", що забезпечує її функціональність, інтеграцію та зручність у використанні. Він дозволяє створювати єдину мережу підключених пристроїв, яка забезпечує комфорт, безпеку та ефективність управління житлом.

```
import com.google.gson.Gson

import com.google.gson.JsonObject

import spark.Spark.*

import java.util.*
```

```
data class SensorData(
    val id: String,
    val type: String,
    val value: Double,
    val timestamp: Long
)

fun main() {
    // Перелік датчиків
    val sensors = mutableListOf<SensorData>()

    // Початкові дані для прикладу
    sensors.add(SensorData("1", "temperature", 25.5, Date().time))
    sensors.add(SensorData("2", "humidity", 60.0, Date().time))
    sensors.add(SensorData("3", "light", 800.0, Date().time))

    // Конфігурація Spark
    port(8080)
    ipAddress("localhost")

    // Маршрутизація запитів
```

```
// Отримати список всіх датчиків
get("/sensors") { _, _ ->
    Gson().toJson(sensors)
}

// Отримати дані конкретного датчика за його ідентифікатором
get("/sensor/:id") { req, _ ->
    val id = req.params(":id")
    val sensor = sensors.find { it.id == id }
    if (sensor != null) {
        Gson().toJson(sensor)
    } else {
        "{\"error\": \"Sensor not found\"}"
    }
}

// Додати новий датчик
post("/sensor") { req, _ ->
    val requestBody = Gson().fromJson(req.body(), JsonObject::class.java)
    val id = UUID.randomUUID().toString()
    val type = requestBody["type"].asString
    val value = requestBody["value"].asDouble
    val timestamp = Date().time
```

```
        val newSensor = SensorData(id, type, value, timestamp)
        sensors.add(newSensor)
        "{\"id\": \"$id\", \"type\": \"$type\", \"value\": $value, \"timestamp\": $timestamp}"
    }

    // Оновити дані існуючого датчика за його ідентифікатором
    put("/sensor/:id") { req, _ ->
        val id = req.params(":id")
        val sensor = sensors.find { it.id == id }
        if (sensor != null) {
            val requestBody = Gson().fromJson(req.body(), JsonObject::class.java)
            val newValue = requestBody["value"].asDouble
            val newTimestamp = Date().time
            sensor.value = newValue
            sensor.timestamp = newTimestamp
            "{\"id\": \"$id\", \"type\": \"${sensor.type}\", \"value\": $newValue, \"timestamp\": $newTimestamp}"
        } else {
            "{\"error\": \"Sensor not found\"}"
        }
    }

    // Видалити датчик за його ідентифікатором
```

```
    delete("/sensor/:id") { req, _ ->
        val id = req.params(":id")
        val sensor = sensors.find { it.id == id }
        if (sensor != null) {
            sensors.remove(sensor)
            "{\"message\": \"Sensor $id deleted successfully\"}"
        } else {
            "{\"error\": \"Sensor not found\"}"
        }
    }
}
```

Цей код представляє собою простий REST API сервер на Kotlin, який можна використовувати для управління даними датчиків у розумному будинку. Ось пояснення кожної частини коду:

1. **Імпорт бібліотек:**
   ```
   import com.google.gson.Gson
   import com.google.gson.JsonObject
   import spark.Spark.*
   import java.util.*
   ```
2. **Створення моделі SensorData:**
   ```
   data class SensorData(
       val id: String,
       val type: String,
       var value: Double,
       var timestamp: Long
   )
   ```
3. **Головна функція main():**
   ```
   fun main() {
       // Перелік датчиків
   ```

```
    val sensors = mutableListOf<SensorData>()

    // Початкові дані для прикладу
    sensors.add(SensorData("1", "temperature", 25.5, Date().time))
    sensors.add(SensorData("2", "humidity", 60.0, Date().time))
    sensors.add(SensorData("3", "light", 800.0, Date().time))

    // Конфігурація Spark
    port(8080)
    ipAddress("localhost")

    // Маршрутизація запитів
    // ...
}
```

В цій функції створюється перелік датчиків sensors з деякими початковими даними. Також налаштовується Spark, вказуючи порт і IP-адресу.

4. **Маршрутизація запитів:**

```
// Отримати список всіх датчиків
get("/sensors") { _, _ ->
    Gson().toJson(sensors)
}

// Отримати дані конкретного датчика за його ідентифікатором
get("/sensor/:id") { req, _ ->
    val id = req.params(":id")
    val sensor = sensors.find { it.id == id }
    if (sensor != null) {
        Gson().toJson(sensor)
    } else {
        "{\"error\": \"Sensor not found\"}"
    }
}

// Додати новий датчик
post("/sensor") { req, _ ->
    val requestBody = Gson().fromJson(req.body(), JsonObject::class.java)
```

```
    val id = UUID.randomUUID().toString()
    val type = requestBody["type"].asString
    val value = requestBody["value"].asDouble
    val timestamp = Date().time
    val newSensor = SensorData(id, type, value, timestamp)
    sensors.add(newSensor)
    "{\"id\": \"$id\", \"type\": \"$type\", \"value\": $value, \"timestamp\":
$timestamp}"
}

// Оновити дані існуючого датчика за його ідентифікатором
put("/sensor/:id") { req, _ ->
    val id = req.params(":id")
    val sensor = sensors.find { it.id == id }
    if (sensor != null) {
        val requestBody = Gson().fromJson(req.body(),
JsonObject::class.java)
        val newValue = requestBody["value"].asDouble
        val newTimestamp = Date().time
        sensor.value = newValue
        sensor.timestamp = newTimestamp
        "{\"id\": \"$id\", \"type\": \"${sensor.type}\", \"value\": $newValue,
\"timestamp\": $newTimestamp}"
    } else {
        "{\"error\": \"Sensor not found\"}"
    }
}

// Видалити датчик за його ідентифікатором
delete("/sensor/:id") { req, _ ->
    val id = req.params(":id")
    val sensor = sensors.find { it.id == id }
    if (sensor != null) {
        sensors.remove(sensor)
        "{\"message\": \"Sensor $id deleted successfully\"}"
    } else {
        "{\"error\": \"Sensor not found\"}"
```

```
        }
    }
```

3. 4. Мобільний додаток

Мобільний додаток є ключовим елементом системи "Розумний будинок", що надає користувачам зручний інтерфейс для управління всіма підключеними пристроями. Завдяки додатку, користувачі отримують можливість дистанційного контролю та керування різними аспектами свого будинку, що значно підвищує рівень комфорту, безпеки та ефективності.

Мобільний додаток дозволяє користувачам здійснювати моніторинг в режимі реального часу. Це означає, що вони можуть перевіряти стан своїх пристроїв, отримувати сповіщення про події та зміни, а також швидко реагувати на виникаючі ситуації, незалежно від їхнього місцезнаходження. Наприклад, при виявленні руху в будинку під час відсутності власника, додаток може надіслати сповіщення, що дозволить вчасно вжити заходів для запобігання небажаних наслідків.

Однією з важливих функцій мобільного додатку є можливість налаштування сценаріїв автоматизації. Користувачі можуть створювати комплексні алгоритми дій для своїх пристроїв, наприклад, автоматичне ввімкнення освітлення при заході сонця або регулювання температури в залежності від часу доби. Це дозволяє значно підвищити зручність користування системою "Розумний будинок" та оптимізувати побутові процеси.

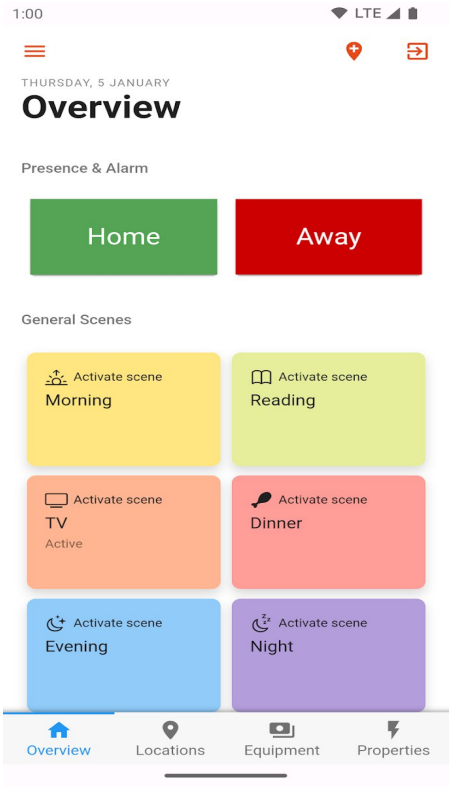
Мобільний додаток також сприяє підвищенню енергоефективності. Користувачі можуть контролювати споживання електроенергії своїми пристроями, вмикати або вимикати їх дистанційно, а також отримувати рекомендації щодо оптимізації використання енергоресурсів. Це не лише

допомагає знизити витрати на комунальні послуги, але й сприяє збереженню природних ресурсів.

Безпека є ще одним важливим аспектом, який забезпечується за допомогою мобільного додатку. Він надає можливість отримувати миттєві сповіщення про загрози, такі як витоки газу, диму або води, а також несанкціонований доступ до приміщень. Це дозволяє вчасно реагувати на потенційні небезпеки та забезпечувати захист мешканців і майна.

Мобільний додаток також підтримує інтеграцію з іншими популярними сервісами та платформами, що дозволяє розширити можливості системи "Розумний будинок". Наприклад, додаток може бути інтегрований з голосовими асистентами, такими як Amazon Alexa або Google Assistant, що дозволяє керувати пристроями за допомогою голосових команд.

Загалом, мобільний додаток є невід'ємною частиною сучасної системи "Розумний будинок", що забезпечує зручність, ефективність та безпеку управління житлом. Він надає користувачам можливість дистанційного контролю, автоматизації процесів та моніторингу стану пристроїв, що сприяє підвищенню якості життя та оптимізації побутових завдань. Завдяки своєму інтуїтивному інтерфейсу та широкому спектру функцій, мобільний додаток стає основним інструментом для взаємодії з системою "Розумний будинок" та реалізації всіх її переваг.



Система "Розумний будинок" являє собою перспективне рішення для сучасного житла, пропонуючи широкий спектр можливостей для підвищення комфорту, безпеки та ефективності. Незважаючи на наявні виклики, розвиток технологій і зниження вартості обладнання сприяють все більшому поширенню цих систем. У майбутньому очікується, що розумні будинки стануть невід'ємною частиною повсякденного життя, пропонуючи ще більше функцій і можливостей для покращення якості життя.

ВИСНОВОК

У дипломній роботі було розглянуто та розроблено додаток до системи "Розумний будинок", який забезпечує інтеграцію та управління різноманітними пристроями і сенсорами через центральний хаб. Основною метою даного проекту було створення зручного та ефективного інтерфейсу, який дозволяє користувачам контролювати умови в їхньому домі з використанням сучасних технологій.

Під час виконання роботи було досягнуто наступних результатів:

Аналіз існуючих рішень: Було проаналізовано сучасні рішення в області розумних будинків, що дало можливість визначити основні функціональні вимоги до додатку та виділити найважливіші аспекти для користувачів.

Розробка архітектури додатку: Було розроблено архітектуру, яка забезпечує гнучкість та масштабованість системи, що дозволяє додавати нові пристрої та сенсори без значних змін у базовій структурі додатку.

Інтеграція сенсорів і пристроїв: Здійснено інтеграцію з різними типами сенсорів (температурні, вологості, диму та газу) та пристроями (розумні лампи, термостати, розетки, дверні замки, камери спостереження). Це дозволяє користувачам в реальному часі отримувати інформацію та керувати своїм домом через мобільний додаток.

Реалізація центрального хабу: Було створено центральний хаб, який забезпечує координацію між різними компонентами системи, включаючи обробку даних від сенсорів та управління пристроями.

Таким чином, створений додаток для системи "Розумний будинок" відповідає сучасним вимогам до подібних рішень, забезпечуючи користувачам високий рівень комфорту та безпеки.

СПИСОК ВИКОРИСТАННОЇ ЛІТЕРАТУРИ

1. Kotlin в действии Книга, Дмитрий Жемеров и Светлана Исакова
2. Head First. Программирование для AndroidКнига, Дон Гриффитс и Дэвид Гриффит
3. Создание умного дома на базе Arduino Книга, Виктор Петин
4. «Kotlin в действии», 2017 год Авторы: Дмитрий Жемеров, Светлана Исакова.
5. RESTful Web APIs: Services for a Changing World. Книга, Леонард Ричардсон, Майкл Амундсен, и Сэм Руби
6. Изучаем Arduino: инструменты и методы технического волшебства. Книга, Джереми Блум.