

ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ
ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
КАФЕДРА КОМП'ЮТЕРНИХ НАУК

КВАЛІФІКАЦІЙНА РОБОТА

на тему: «Розробка засобів інтелектуального управління та контролю системи «розумній дім»»

на здобуття освітнього ступеня бакалавра
зі спеціальності 122 Комп'ютерні науки

(код, найменування спеціальності)

освітньо-професійної програми Комп'ютерні науки
(назва)

*Кваліфікаційна робота містить результати власних досліджень.
Використання ідей, результатів і текстів інших авторів мають посилання
на відповідне джерело*

(підпис)

Ельбезрі Антон
(Ім'я, ПРИЗВИЩЕ здобувача)

Виконав:
здобувач вищої освіти
група КНД-41

Ельбезрі Антон

Керівник:
І науковий ступінь,
вчене звання

Іщеряков С.М

Рецензент:
науковий ступінь,
вчене звання

(Ім'я, ПРИЗВИЩЕ)

**ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ**
Навчально-науковий інститут інформаційних технологій

Кафедра Комп'ютерних наук

Ступінь вищої освіти Бакалавр

Спеціальність Комп'ютерні науки

Освітньо-професійна програма Комп'ютерні науки

ЗАТВЕРДЖУЮ

Завідувач кафедри Комп'ютерних наук

_____ Віктор ВИШНІВСЬКИЙ
«_____» _____ 2024 р.

**ЗАВДАННЯ
НА БАКАЛАВРСЬКУ РОБОТУ СТУДЕНТУ**

Ельбезрі Антон Назійович

(прізвище, ім'я, по батькові)

1. Тема роботи: «Розробка засобів інтелектуального управління та контролю системи «розумній дім»» Керівник роботи професор кафедри, д. т. н., професор Іщераков С.М.

(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом вищого навчального закладу від «27» лютого 2024 року №36. 2. Строк подання студентом роботи «31» травня 2024 року

3. Вхідні дані до роботи: Науково-технічна література з питань, що пов'язані з розробкою систем розумного дому.

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

4.1 Аналіз предметної області;

4.2 Розгляд алгоритмів пошуку найкоротшого шляху;

4.3 Вибір технології розробки та інструментальних

4.4 засобів;

5. Перелік графічного матеріалу
- 5.1 Блок - схеми
- 5.2 Результат роботи створеної програми
6. Дата видачі завдання 23.02.2024

КАЛЕНДАРНИЙ ПЛАН

№ з / п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Аналіз поставленої задачі	23.02-01.03.24	
2	Підбір науково-технічної літератури	02.03-15.03.24	
3	Аналіз алгоритмів комп'ютерного зору	16.03-31.03.24	
4	Проектування програмного застосунку	01.04-07.04.24	
5	Розробка програмного застосунку	08.04-30.04.24	
6	Написання вступу, висновків, реферату	01.05-05.05.24	
7	Написання основних розділів пояснювальної записки	06.05-20.05.24	
8	Розробка демонстраційних матеріалів	21.05-25.05.24	

Студент _____ Ельбезрі А. Н.
(підпис) (прізвище та ініціали)

Керівник роботи _____ Іщераков С.М.
(підпис) (прізвище та ініціали)

РЕФЕРАТ

Текстова частина бакалаврської роботи: 56с., 1 табл., 20 рис., 11 джерел.

Об'єкт дослідження – система “розумний дім”, її компоненти та механізми управління.

Предмет дослідження – процеси та методи, які використовуються для створення інтелектуальної системи управління “розумний дім”.

Мета роботи – створити ефективну та надійну систему, яка б забезпечувала автоматизацію домашніх процесів.

Методи дослідження – аналіз існуючих рішень та систем розумного дому для визначення найкращих практик та поточних тенденцій..

Сфера застосування – різні аспекти повсякденного життя та бізнесу..

Ключові слова: РОЗУМНИЙ ДІМ.

ЗМІСТ

ВСТУП	8
1 ОПИС ПРЕДМЕТНОЇ ОБЛАСТІ ЗАВДАННЯ	9
1.1 Поняття “Розумний дім”	10
1.2 Постановка завдання	12
1.3 Основні вимоги до програмного продукту	13
1.4 Висновок до розділу	14
2 ОГЛЯД ТА ПОРІВНЯЛЬНА ХАРАКТЕРИСТИКА ПОДІБНИХ ПРОГРАМ	15
2.1 Ajax Systems	16
2.2 Ecoisme	18
2.3 Honeywell	21
2.4 Висновок до розділу	23
3. ПРОЕКТУВАННЯ СИСТЕМИ	24
3.1 Планування розробки системи	24
3.1 Опис функціональності	26
3.2 Інтеграція з іншими пристроями	27
3.2 Вибір технологій для розробки	28
3.3 Архітектура системи	34
3.3.1 Модульна архітектура	37
3.4 Тестування	38
3.5 Висновок до розділу	42
ВИСНОВКИ	45
ПЕРЕЛІК ПОСИЛАНЬ	46
Додаток А	47
Додаток Б.....	80

ВСТУП

В сучасному світі, завдяки стрімкому розвитку технологій, концепція "розумного дому" стає все більш актуальною та важливою. Розумні системи управління та контролю, що вбудовані у житлові приміщення, не лише забезпечують комфорт та безпеку мешканців, але й відкривають нові можливості для енергоефективного використання ресурсів та оптимізації побутових процесів.

Розробка засобів інтелектуального управління та контролю системи "розумний дім" є відповіддю на зростаючі потреби сучасного суспільства в зручному та безпечному житті. Ці системи використовують передові технології в області інформаційних та комунікаційних технологій, датчиків, штучного інтелекту та Інтернету речей для автоматизації різних аспектів побутового життя. Завдяки системам "розумний дім", мешканці можуть контролювати освітлення, температуру, безпеку, аудіо- та відеосистеми в будинку за допомогою смартфонів або голосових помічників.

Крім того, такі системи можуть аналізувати та оптимізувати споживання електроенергії, води та інших ресурсів, що дозволяє ефективно використовувати їх та зменшувати витрати. Однак, реалізація інтелектуального управління та контролю в "розумному домі" потребує вдосконалення технологій, розробки нових алгоритмів та програмного забезпечення. Також важливо забезпечити сумісність та безпеку взаємодії різних систем та пристроїв.

Мета даної роботи полягає у розробці та впровадженні ефективних та надійних засобів управління та контролю системи "розумний дім".

Для досягнення цієї мети будуть вивчені існуючі технології та алгоритми, розроблені та впроваджені нові методи управління та контролю, а також проведені експерименти та тестування для підтвердження ефективності та надійності розроблених засобів.

Актуальність: з огляду на зростаючу важливість енергозбереження, "розумний дім" набуває особливої значущості, оскільки він дозволяє оптимізувати споживання ресурсів. Це не тільки сприяє зниженню витрат для кінцевого

споживача, але й зменшує екологічний вплив, що є критично важливим у контексті глобальних кліматичних змін.

Об'єкт дослідження: система “розумний дім”, її компоненти та механізми управління.

Предмет дослідження: процеси та методи, які використовуються для створення інтелектуальної системи управління “розумний дім”.

Мета роботи: створити ефективну та надійну систему, яка б забезпечувала автоматизацію домашніх процесів, підвищення комфорту проживання, енергоефективності та безпеки житлових приміщень

Для досягнення мети роботи необхідно вирішити такі завдання:

- 1) Вивчити сучасні операційні системи для розумних будинків.
- 2) Дослідити різні види пристроїв, що входять до системи розумного дому.
- 3) Перевірити працездатність та коректність роботи.
- 4) Розглянути способи підключення і взаємодії між пристроями.

1 ОПИС ПРЕДМЕТНОЇ ОБЛАСТІ ЗАВДАННЯ

1.1 Поняття “Розумний дім”

Розумний дім — це сучасна технологічна система, що працює на основі штучного інтелекту (ШІ), призначена для автоматизації та полегшення повсякденних завдань без втручання людини. В основі цієї системи лежить операційна система multi-room, яка об'єднує всі електроприлади у домашню мережу з доступом до Інтернету, дозволяючи керувати ними за допомогою пульта, комп'ютера або смартфона.

Система аналізує параметри в приміщенні і дозволяє налаштовувати роботу приладів під індивідуальні потреби користувача за допомогою кількох кнопок. У більшості сучасних розумних домів контролер взаємодіє з іншими пристроями через радіосигнал. Завдяки датчикам система може працювати автономно, без участі людини. Наприклад, якщо датчики виявляють зміну температури або вологості, система автоматично підлаштовує ці параметри до оптимальних значень.

Основні функції розумного дому:

Розумне освітлення:

- Світло вмикається та вимикається автоматично при появі або виході людини з кімнати.
- Можна налаштувати освітлення для різних сценаріїв, таких як вечірка або романтична вечір.

Мікроклімат:

Система автоматично регулює температуру та вологість повітря в кожній кімнаті.

- Вентиляційна система забезпечує свіже та чисте повітря.
- Комфортне пересування:

- Система визначає траєкторію пересування членів родини і налаштовує параметри (температура, музика, освітлення) під їхні уподобання.

Енергозбереження:

- Температура в будинку регулюється залежно від зовнішніх умов.
- Батареї можуть автоматично вмикатися ввечері перед сном та вимикатися вранці.

Додаткові можливості:

- Сповіщення та оповіщення:
- Прилади можуть повідомляти про необхідність обслуговування або техогляду.
- Сигналізація може одночасно оповістити службу безпеки та власника у разі появи незваних гостей.

Індивідуальні налаштування:

- Система розпізнає, хто з членів родини пересувається по будинку, і налаштовує параметри відповідно до їхніх уподобань.
- Розумний замок використовує Bluetooth для фіксації входів і виходів, а також надання доступу іншим користувачам через згенеровані ключі.

Автоматизація комерційних процесів:

- Облік продукції через зчитування штрих-кодів або RFID-тегів.
- Автоматичне створення списків покупок.

1.2 Постановка завдання

Розумний будинок є системою побутової техніки та інших пристроїв, що виконують операції і допомагають в повсякденному житті на основі автоматизованого управління. У сучасному світі, домашня автоматизація є гнучким набором систем. Людина самостійно вибирає функціонал, виходячи зі своїх побажань. Так, кожен власник подібного житла, може вибрати пристрої для установки і налаштовувати, відштовхуючись від персональних потреб.

В рамках виконання роботи необхідно розробити засоби інтелектуального управління та контролю системи розумного дому. Особливу увагу необхідно приділити доступності та зрозумілості керування програмним продуктом, щоб користувач не мав проблем під час використання програми. Також звернути увагу на дизайн застосунку, він має бути приємним, лаконічним та інтуїтивно зрозумілим для користувача.

Для розробки використовувати технологію Android Studio, створення дизайну в Figma, та мови програмування Java та Kotlin. Можливості користувача – віддалене керування, автоматизація рутинних процесів, моніторинг енергоспоживання, контролювання безпеки дому, голосове керування.

1.3 Основні вимоги до програмного продукту

Програма, з точки зору функціональності, має бути інтуїтивно зрозумілим і зручним у використанні, надавати можливість користувачеві повноцінно використовувати програму.

Функціональні вимоги — це вимоги до програмного забезпечення, які описують внутрішню роботу системи, її поведінку: обчислення даних, маніпулювання даними, обробка даних та інші специфічні функції, які має виконувати система.

Функціональні вимоги:

- Управління пристроями: Можливість дистанційного управління всіма підключеними пристроями (освітленням, опаленням, системами безпеки тощо).

Автоматизація сценаріїв (наприклад, включення світла за певних умов, регулювання температури).

- Моніторинг та звітність: Реальний час моніторинг стану пристроїв.

Зберігання та аналіз даних про споживання енергії, температуру, вологість тощо.

- Сповіщення та алерти: Сповіщення про незвичайні ситуації (наприклад, витік води, відкриття дверей). Налаштування різних типів повідомлень (SMS, push-сповіщення, електронна пошта).

- Інтеграція з іншими системами: Підтримка інтеграції з голосовими асистентами (наприклад, Google Assistant, Amazon Alexa). Сумісність з різними стандартами пристроїв IoT (Zigbee, Z-Wave, Wi-Fi).

Нефункціональні вимоги — це вимоги до програмного забезпечення, які задають критерії для оцінки якості його роботи. На відміну від функціональних вимог, які визначають що система повинна робити, нефункціональні вимоги визначають якою система повинна бути.

Нефункціональні вимоги:

- Продуктивність: Мінімальні затримки в управлінні пристроями. Висока швидкість обробки та передачі даних.

- Надійність: Стійкість до збоїв та можливість відновлення після збоїв. Безперебійна робота системи.

- Масштабованість: Можливість додавання нових пристроїв без значного впливу на продуктивність системи.

- Легкість у використанні: Інтуїтивно зрозумілий інтерфейс користувача. Легкість встановлення та налаштування системи.

Для роботи додатку потрібне з'єднання з мережею Інтернет.

Технічні вимоги:

- Платформа та архітектура: Підтримка різних операційних систем (iOS, Android, Windows). Використання гнучкої архітектури (наприклад, мікросервіси) для полегшення оновлення та підтримки.
- Інтерфейси та протоколи: Відкриті API для інтеграції з іншими системами та додатками. Підтримка стандартних протоколів зв'язку (HTTP, MQTT).

1.4 Висновок до розділу

В даному розділі було проаналізовано предметну область завдання: мобільні додатки, їх призначення. Розглянуті принципи розробки та проектування мобільних додатків

2 ОГЛЯД ТА ПОРІВНЯЛЬНА ХАРАКТЕРИСТИКА ПОДІБНИХ ПРОГРАМ

2.1 Ajax Systems

Ajax Systems - це українська компанія, що спеціалізується на розробці та виробництві систем безпеки для розумного дому та бізнесу. Заснована в 2011 році, компанія швидко стала лідером на ринку бездротових систем безпеки, завоювавши довіру користувачів у більш ніж 100 країнах світу

Основні характеристики Ajax Systems:

- Бездротові технології: Всі пристрої Ajax працюють на бездротових технологіях, що дозволяє швидко та легко встановлювати системи без потреби проводів.
- Сучасний дизайн: Продукція Ajax відрізняється стильним та сучасним дизайном, що дозволяє легко інтегрувати її у будь-який інтер'єр.
- Надійність і безпека: Системи Ajax відомі своєю високою надійністю та безпекою. Вони використовують передові технології шифрування та мають вбудовані заходи захисту від вторгнень.
- Мобільний додаток: Ajax має зручний мобільний додаток для керування системою з будь-якого місця за допомогою смартфона.
- Широкий вибір пристроїв: Компанія пропонує різноманітні пристрої для різних потреб, включаючи датчики руху, дверей/вікон, вогню, протікання води, камери відеоспостереження та багато іншого.

На рисунках 2.1 - 2.3 буде зображено інтерфейс користувача Ajax System

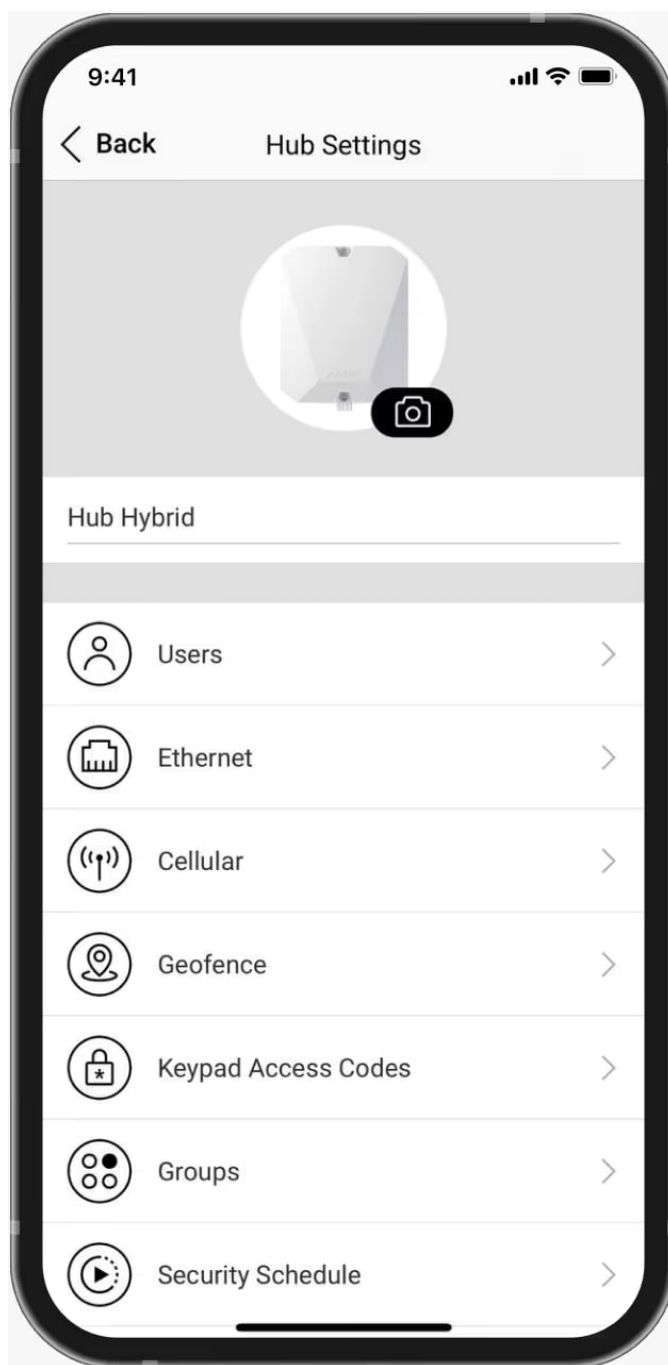


Рисунок 2.1 - налаштування

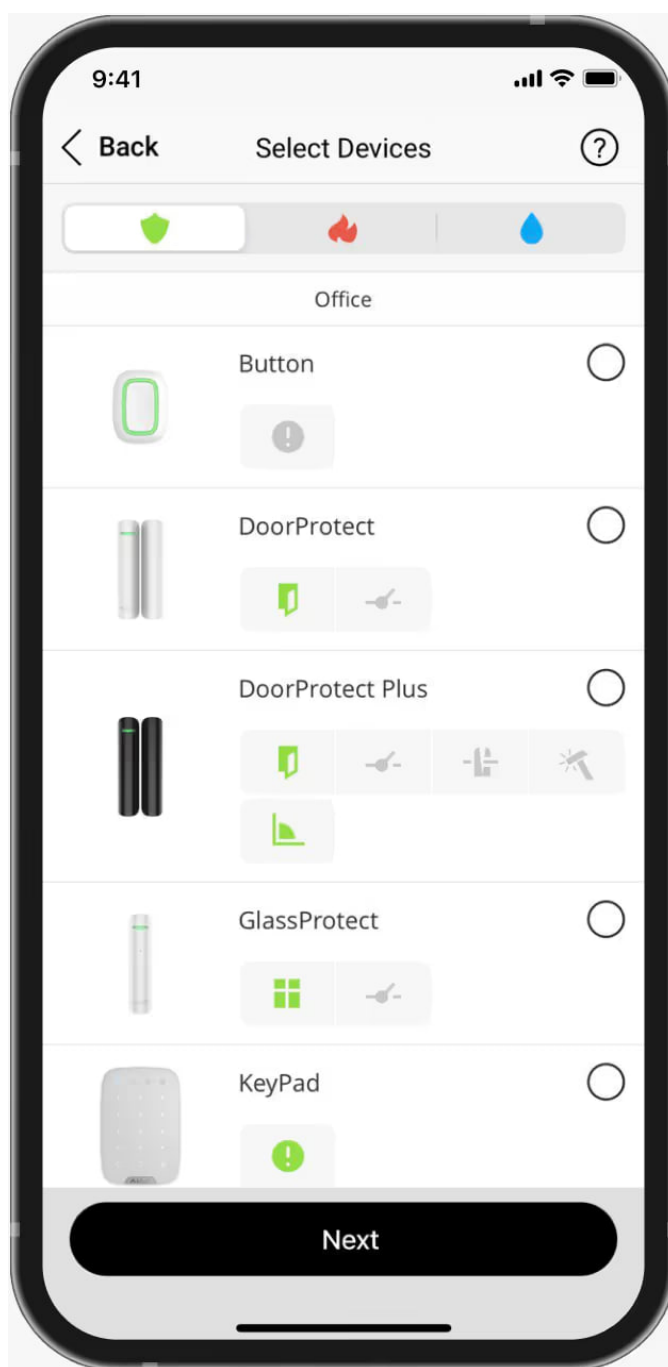


Рисунок 2.2 - вибір пристрою користувачем

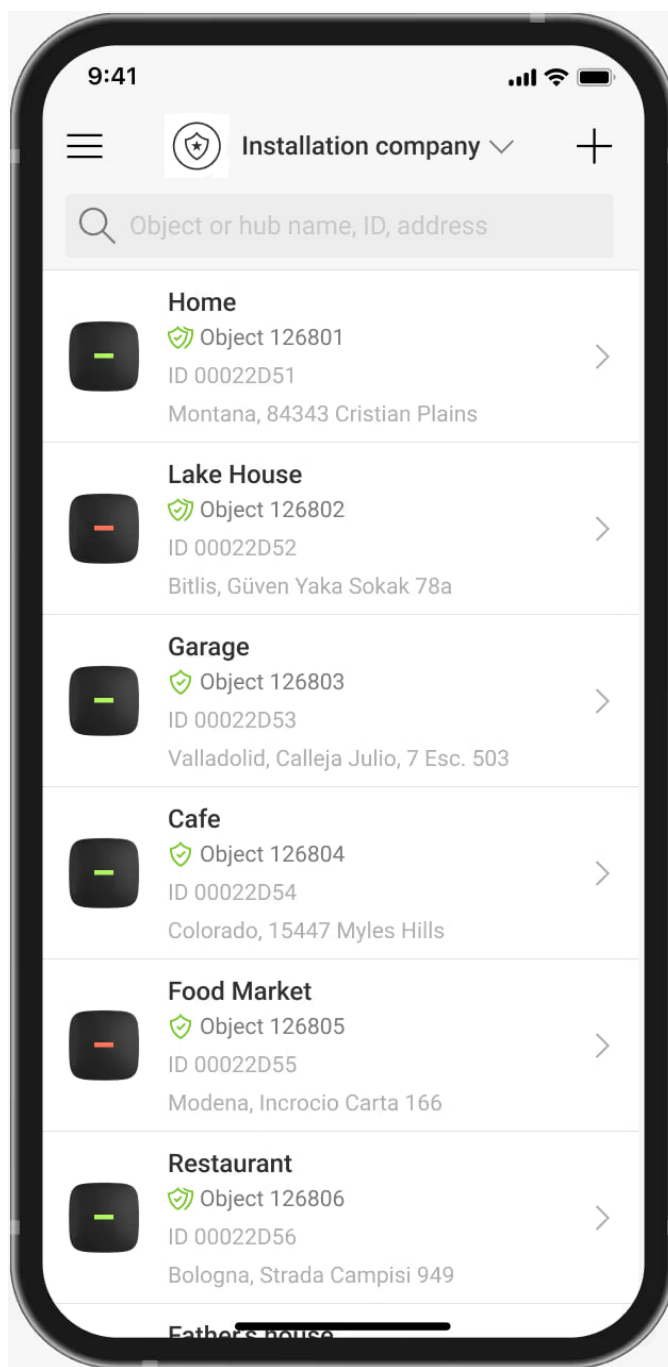


Рисунок 2.3 - список всіх об'єктів користувача

2.2 Ecoisme

Ecoisme - це компанія, що спеціалізується на розробці інноваційних рішень для енергоефективності та моніторингу енергоспоживання в домогосподарствах. Основною метою Ecoisme є надання користувачам інструментів для ефективного управління електроенергією та зменшення витрат на комунальні послуги..

Основні характеристики Ecoisme:

- Бездротові технології: Всі пристрої працюють на бездротових технологіях, що дозволяє легко встановлювати системи без потреби проводів.
- Дизайн: Продукція відрізняється стильним та гарним дизайном, що дозволяє легко інтегрувати її у будь-який інтер'єр.
- Надійність і безпека: Системи Ecoisme відомі своєю високою надійністю та безпекою. Вони використовують передові технології шифрування та мають вбудовані заходи захисту від вторгнень.
- Мобільний додаток: зручний мобільний додаток для керування системою з будь-якого місця за допомогою смартфона.
- Широкий вибір пристроїв: Компанія пропонує різноманітні пристрої для різних потреб, включаючи датчики руху, дверей/вікон, вогню, протікання води, камери відеоспостереження та багато іншого.

На рисунках 2.4 - 2.6 буде зображено інтерфейс користувача Ecoisme

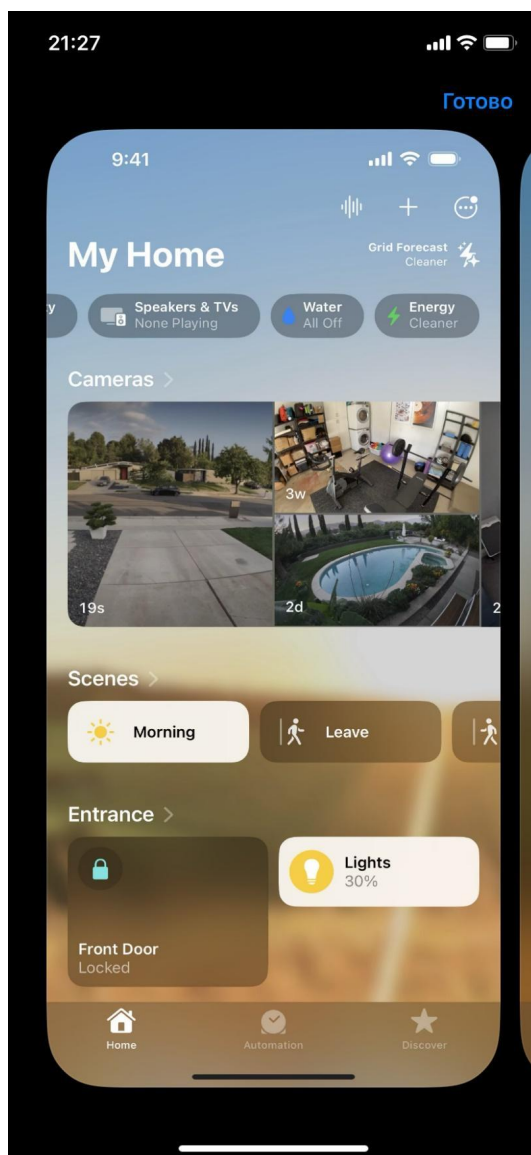


Рисунок 2.4 - сторінка об'єкту

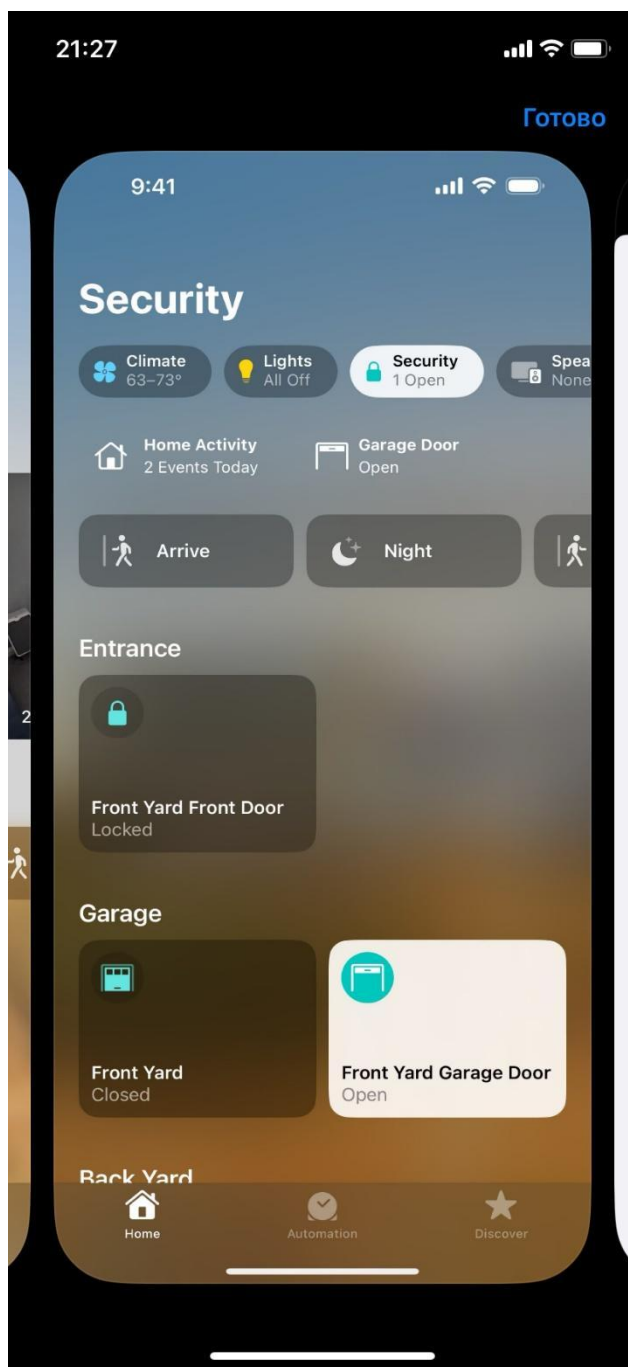


Рисунок 2.5 - сторінка безпеки

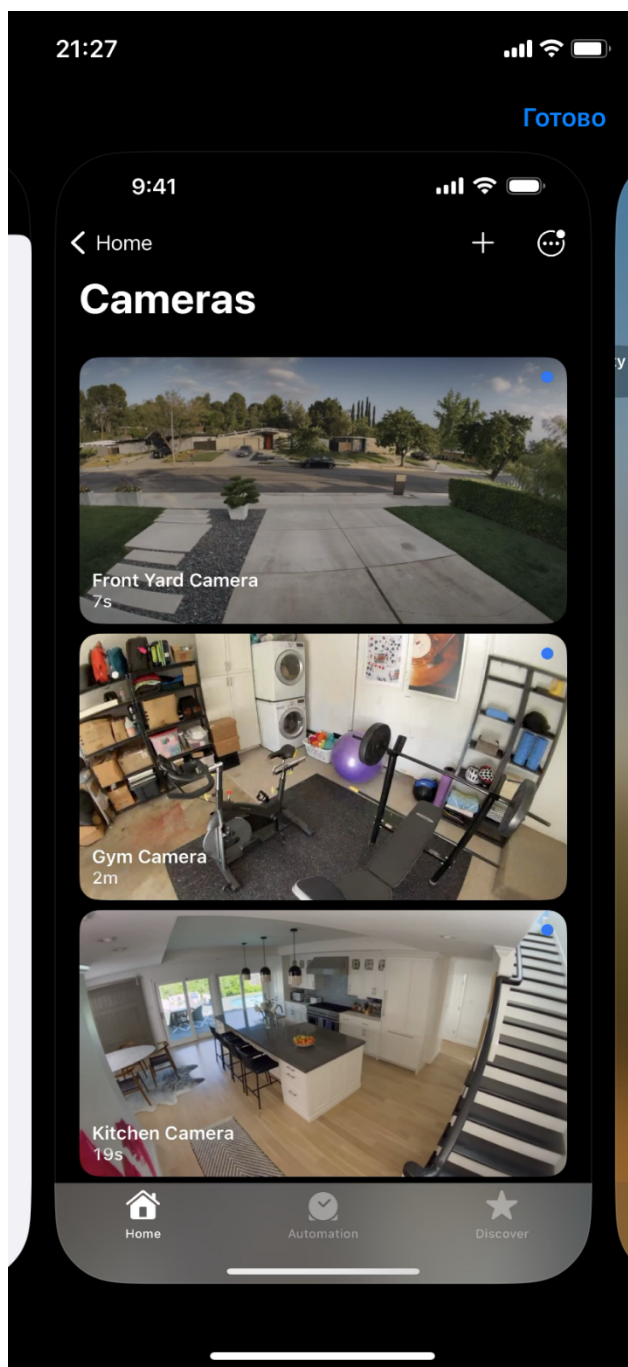


Рисунок 2.6 - Список активных камер видеонаблюдения

2.3 Honeywell

Honeywell - це міжнародна корпорація, яка спеціалізується на розробці та виробництві різноманітних технологій, включаючи системи автоматизації промисловості, авіаційні системи, а також рішення для дому та будівель. Компанія була заснована у 1906 році та швидко стала однією з провідних у світі у своїх галузях.

В контексті розумного дому Honeywell пропонує різні продукти та рішення для забезпечення безпеки та комфорту у домашніх умовах:

- Розумні термостати: Honeywell виробляє розумні термостати, які дозволяють користувачам керувати температурою в своєму домі за допомогою мобільного додатку. Ці термостати можуть автоматично регулювати температуру відповідно до графіку користувача та зовнішніх умов, що дозволяє зменшити споживання енергії та забезпечити комфорт.
- Системи безпеки для дому: Honeywell пропонує рішення для моніторингу безпеки в домі, включаючи системи відеоспостереження, датчики руху та диму, системи сигналізації та контролю доступу. Ці системи дозволяють користувачам залишатися під захистом та отримувати сповіщення про будьякі підозрілі події.
- Автоматизація освітлення: Honeywell також пропонує рішення для автоматизації освітлення в домі. Вони можуть включати у себе розумні вимикачі, датчики руху та програмовані сценарії освітлення, що дозволяють ефективно використовувати енергію та створювати бажану атмосферу.

На рисунках 2.7 буде зображено інтерфейс користувача Honeywell

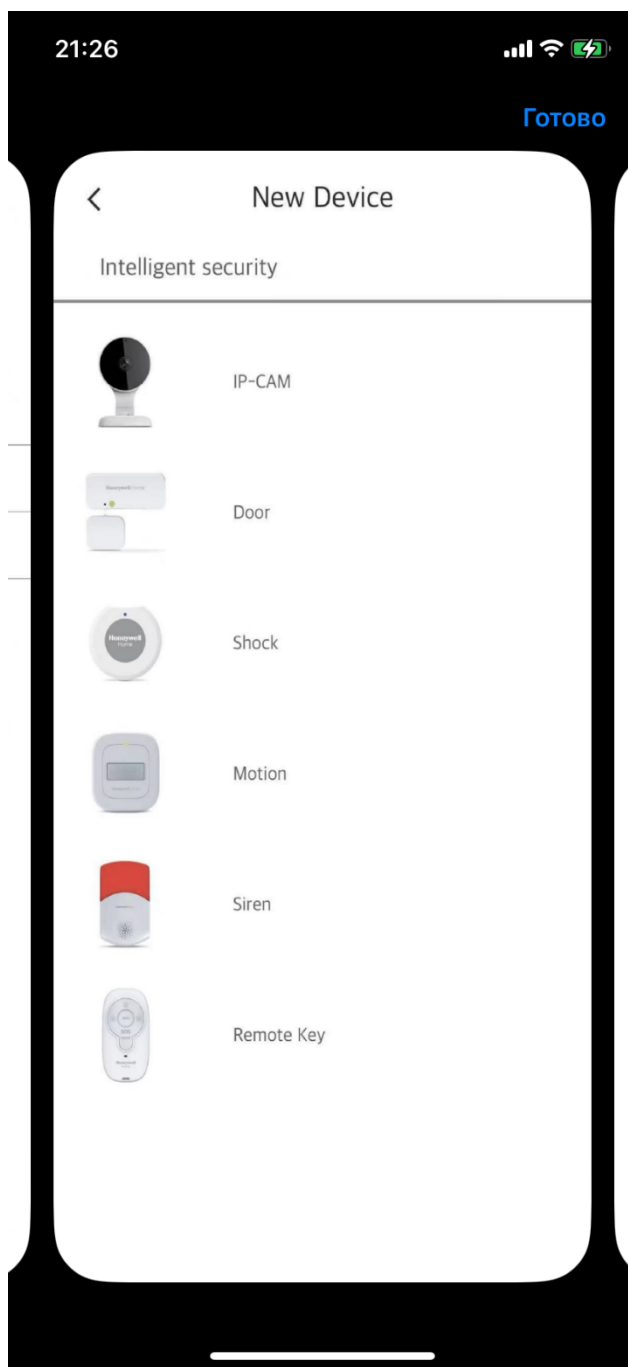


Рисунок 2.7 - меню додавання нового пристрою

2.4 Висновок до розділу

В результаті детального огляду та порівняльного аналізу різних програмних продуктів в даному сегменті ринку можна зробити кілька ключових висновків. Перш за все, спостерігається загальна тенденція до поступового розвитку функціональності, спрямованої на полегшення користувацького досвіду та підвищення продуктивності.

3 ПРОЕКТУВАННЯ СИСТЕМИ

3.1 Планування розробки системи

Планування — це процес визначення та організації всіх необхідних кроків, ресурсів і завдань для створення, впровадження та підтримки програмного продукту або системи. Цей процес включає визначення цілей, строків, бюджетів і ризиків, а також координацію роботи команди розробників, дизайнерів, тестувальників та інших учасників проекту.

Мета планування — забезпечити ефективну та своєчасну розробку системи, яка відповідає вимогам користувачів і замовників.

Основні етапи планування розробки системи:

Визначення мети та обсягу проекту:

- Мета проекту: Що саме повинна виконувати система.
- Обсяг проекту: Які функції та завдання повинні бути включені в систему.

Збір і аналіз вимог:

- Функціональні вимоги: Які конкретні функції повинна виконувати система.
- Нефункціональні вимоги: Вимоги до продуктивності, безпеки, масштабованості тощо.
- Вимоги користувачів: Які потреби та очікування мають користувачі.

Створення технічного завдання:

- Документ, що містить детальні вимоги до системи, який слугує основою для подальших етапів розробки.

Проектування системи:

- Архітектура системи: Визначення структури системи, взаємодії між компонентами.

- Дизайн бази даних: Структура і моделі даних.
- Дизайн інтерфейсу користувача: Макети та прототипи інтерфейсів.

Розробка плану проекту:

- Розклад завдань: Декомпозиція проекту на окремі завдання та визначення строків їх виконання.
- Визначення критичних точок: Основні віхи та контрольні точки проекту.
- Розподіл завдань: Призначення відповідальних за виконання конкретних завдань.

Тестування та забезпечення якості:

- План тестування: Визначення підходів і методів тестування системи.
- Контроль якості: Перевірка відповідності системи вимогам та стандартам.

Впровадження та підтримка:

- План впровадження: Заходи для розгортання системи у користувачів.
- Підтримка користувачів: Стратегія підтримки та обслуговування системи після впровадження.

Планування розробки системи є критично важливим етапом, який дозволяє:

- Забезпечити зрозумілу та структуровану реалізацію проекту.
- Зменшити ризики та непередбачені витрати.
- Забезпечити відповідність системи вимогам користувачів і замовників.
- Оптимізувати використання ресурсів та зусиль команди.

- Підвищити ймовірність своєчасного завершення проекту.

Без ретельного планування розробка системи може зіткнутися з численними проблемами, затримки, недосягнення очікувань користувачів та низьку якість кінцевого продукту. Тому детальне планування є основою успішної реалізації будь-якого проекту розробки системи.

3.1 Опис функціональності

Система інтелектуального управління “розумний дім” має на меті забезпечити автоматизацію та оптимізацію різних процесів у домогосподарстві.

Ось детальний опис основних функцій:

- Керування освітленням: Система дозволяє автоматично регулювати освітлення в залежності від часу доби, присутності людей у приміщенні, а також створювати різні сценарії освітлення для різних ситуацій.
- Кліматичні системи: Автоматизація опалення, вентиляції, кондиціонування та водопостачання для підтримки оптимального мікроклімату в будинку. Безпека: Включає в себе відеоспостереження, пожежну та охоронну сигналізацію, що забезпечує захист від несанкціонованого доступу та інших загроз.
- Керування побутовими приладами: Інтеграція побутових приладів з системою для їх дистанційного керування та автоматизації повсякденних завдань. Догляд за двором: Автоматизація поливу саду, керування ролетами чи шторами, відкриття воріт тощо.
- Енергозбереження: Система “розумний дім” спрямована на зниження витрат енергії через ефективне управління ресурсами та обладнанням.

- Контроль стану будівлі: Моніторинг стану будівлі, обладнання та комунікацій, що дозволяє своєчасно виявляти несправності та запобігати аварійним ситуаціям.
- Інтерфейси керування: Можливість керування системою через різні інтерфейси, такі як мобільні додатки, голосові команди, центральний контролер у вигляді планшета чи комп'ютера.
- Інтеграція з інтернетом речей: З'єднання з різними пристроями та сервісами через інтернет для створення єдиної інтегрованої системи.

Ці функції в сукупності створюють зручну, безпечну та ефективну житлову середу, що адаптується до потреб та звичок мешканців. “Розумний дім” не лише полегшує повсякденне життя, але й сприяє економії ресурсів та підвищенню безпеки.

3.2 Інтеграція з іншими пристроями

Інтеграція пристроїв та сервісів — це процес об'єднання різних пристроїв, таких як освітлення, опалення, кондиціювання повітря, систем безпеки, а також сервісів, наприклад, голосового помічника або мобільного додатка, для створення автоматизованої та зручної системи керування вашим житлом.

Нижче будуть наведені основні складові інтеграції:

- Пристрої: Це електронні пристрої, які можуть бути підключені до мережі Інтернет та керуються за допомогою мобільного додатка, веб-інтерфейсу або голосових команд. Це можуть бути "розумні" лампочки, термостати, датчики руху, камери відеоспостереження, домофони тощо.
- Протоколи зв'язку: Для того щоб пристрої могли спілкуватися між собою та з центральним контролером, вони часто використовують різні протоколи зв'язку, такі як Wi-Fi, Bluetooth, Zigbee, Z-Wave тощо. Важливо, щоб

пристрої використовували сумісні протоколи для ефективної роботи всієї системи.

- Центральний контролер: Це центральний пристрій або програмне забезпечення, яке керує всією системою "Розумного дому". Він може бути представлений в різних формах, від фізичних пристроїв до хмарних платформ або навіть домашніх серверів. Його основна функція - це обробка команд від користувача та керування підключеними пристроями.
- Мобільні додатки: Багато систем "Розумний дім" постачаються з мобільними додатками, які дозволяють користувачам керувати своїм домом з будь-якого місця, де є доступ до Інтернету. Ці додатки дозволяють вмикати/вимикати пристрої, регулювати температуру, переглядати відео з камер спостереження та отримувати сповіщення про події в домі.
- Інтеграція з голосовими асистентами: Деякі системи "Розумного дому" можуть бути інтегровані з голосовими асистентами, такими як Amazon Alexa, Google Assistant або Apple Siri. Це дозволяє користувачам керувати своїм домом за допомогою голосових команд.
- Сценарії та автоматизація: Одним з ключових переваг "Розумного дому" є можливість налаштування різних сценаріїв та автоматизованих дій. Наприклад, ви можете налаштувати сценарій "Вечірній відпочинок", який автоматично вимкне світло в усіх кімнатах, знизить температуру та увімкне спокійну музику за вашим запитом.

3.2 Вибір технологій для розробки

Розробка системи "розумний дім" включає в себе вибір різноманітних технологій для забезпечення інтелектуального управління та контролю. Ось деякі засоби та технології, які були використані при розробці програмного продукту:

Для розробки інтерфейсу мобільного застосунку був використаний програмний засіб Figma.

Figma — це популярний інструмент для проєктування інтерфейсів, який дозволяє розробникам і дизайнерам створювати, співпрацювати та тестувати проєкти в реальному часі. Він відомий своїми потужними функціями для командної роботи і можливістю працювати безпосередньо в браузері, що робить його особливо зручним для віддаленої співпраці.

Ось деякі особливості Figma:

- Доступність: Figma працює в браузері, що означає, що користувачі можуть отримати доступ до своїх проєктів з будь-якого комп'ютера без необхідності встановлювати програмне забезпечення.
- Кросплатформеність: Підтримує роботу на Windows, macOS, і Linux.
- Мультикористувацька підтримка: Декілька користувачів можуть одночасно працювати над одним проєктом, що полегшує командну роботу і комунікацію.
- Коментарі: Користувачі можуть залишати коментарі прямо на макетах, що спрощує обговорення і внесення змін.
- Інструменти для дизайну: Включає в себе все необхідне для створення макетів, від базових фігур до складних компонентів і варіантів.
- Компоненти: Дозволяють створювати повторювані елементи, які можна легко змінювати і оновлювати в масштабах всього проєкту.
- Прототипування: Дозволяє створювати інтерактивні прототипи для тестування користувацького досвіду безпосередньо в Figma.

- Інтеграції: Підтримує інтеграції з іншими інструментами, такими як Slack, Zeplin, і JIRA.
- Плагіни: Велика бібліотека плагінів, що дозволяє розширити функціональність Figma і автоматизувати рутинні задачі.
- Версійність: Користувачі можуть переглядати історію змін і повертатися до попередніх версій проєктів.
- Автоматичне збереження: Всі зміни автоматично зберігаються в реальному часі.
- Компоненти: Дозволяють створювати універсальні елементи, які можна використовувати в різних частинах проєкту.
- Варіанти: Дозволяють створювати декілька варіантів компонентів для різних станів (наприклад, активний/неактивний стан кнопки).

Переваги Figma

- Швидка Колаборація: Завдяки можливості співпраці в реальному часі, команди можуть ефективніше працювати разом.
- Зручний Інтерфейс: Інтуїтивно зрозумілий інтерфейс робить Figma доступною навіть для новачків.
- Гнучкість: Підходить для різних етапів розробки — від початкового дизайну до створення інтерактивних прототипів.
- Спільне Використання: Легко ділитися проєктами з клієнтами та командою без необхідності завантаження файлів.

Недоліки Figma

- Залежність від Інтернету: Оскільки Figma працює в браузері, стабільний доступ до Інтернету є обов'язковим.
- Продуктивність: На дуже великих або складних проєктах іноді можуть виникати проблеми з продуктивністю.

На рисунках 3.1 - 3.4 буде зображено інтерфейс Figma.

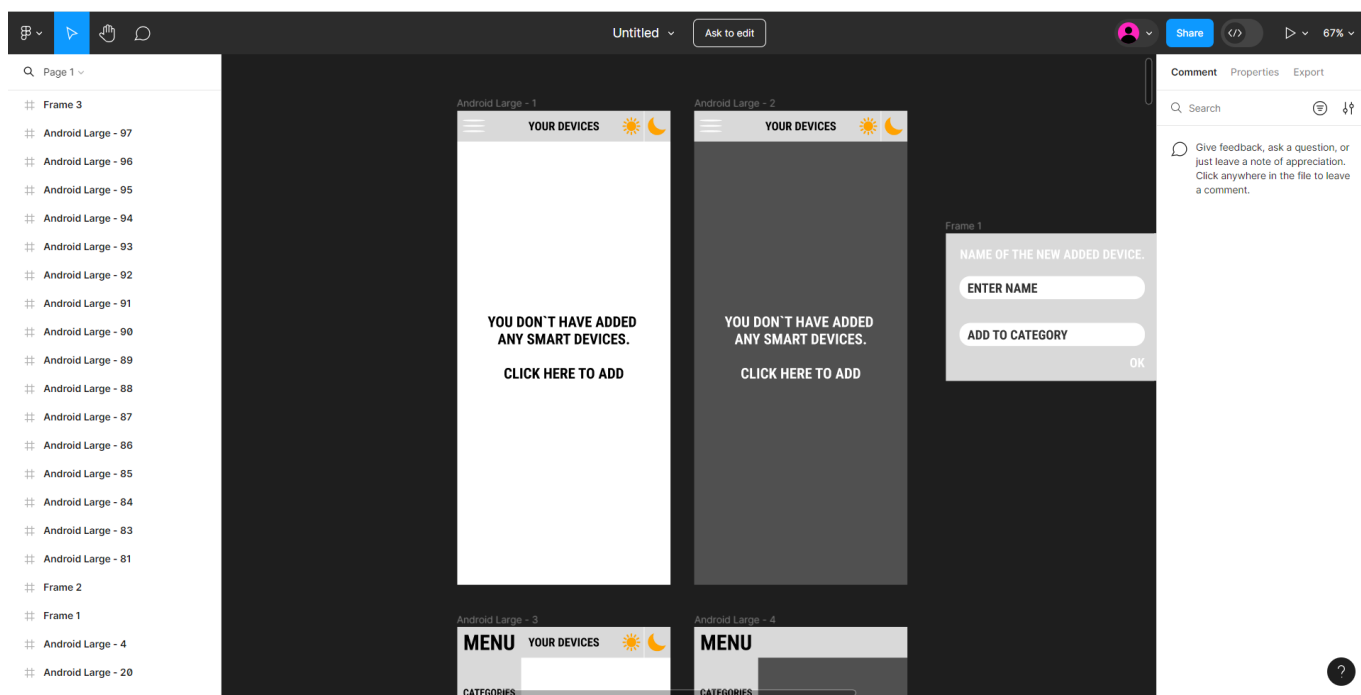


Рисунок 3.1 - основний інтерфейс програми

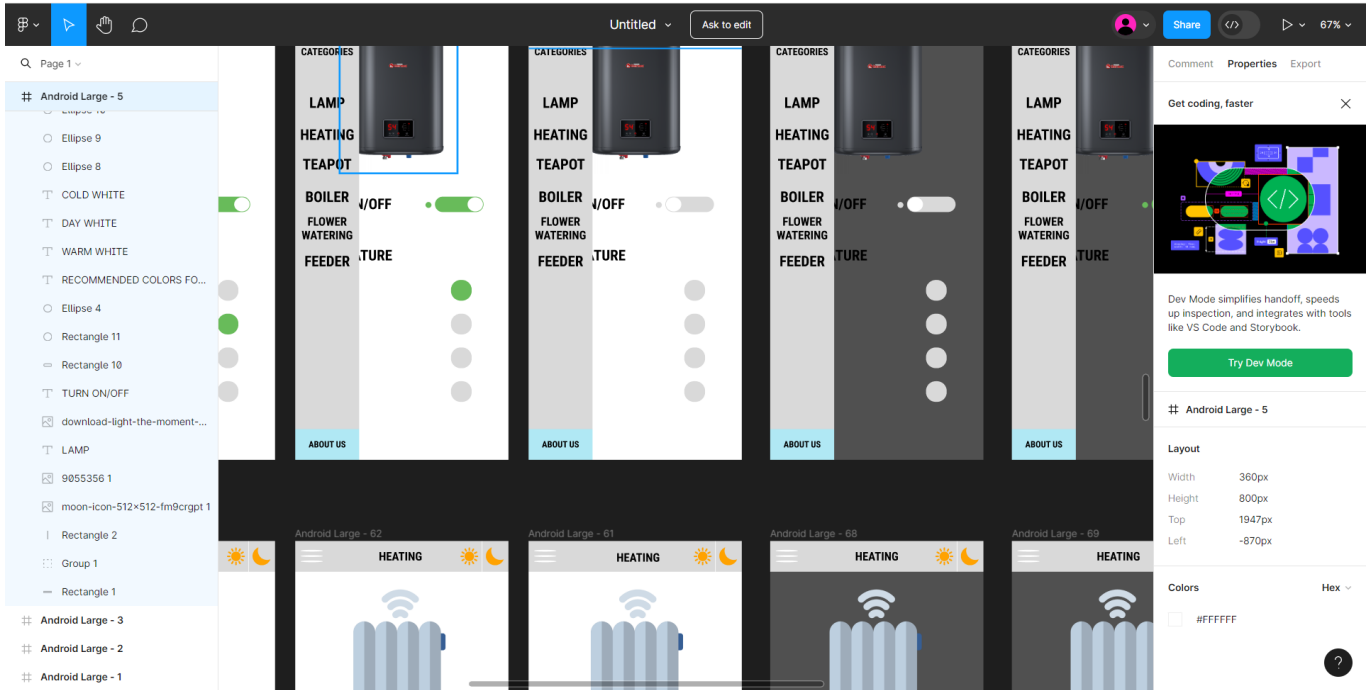


Рисунок 3.2 - структура фрейму

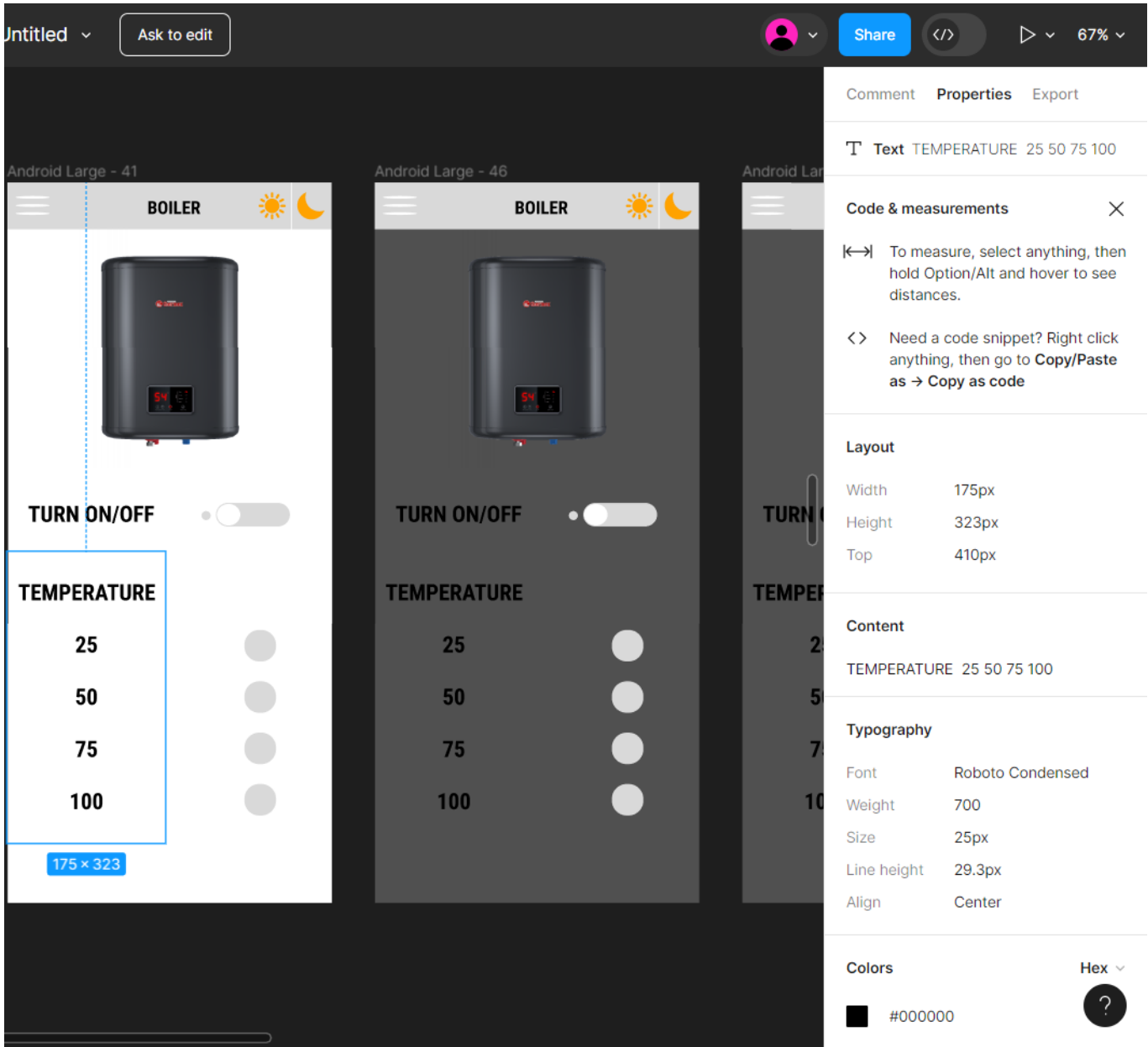


Рисунок 3.3 - параметры избраного элемента

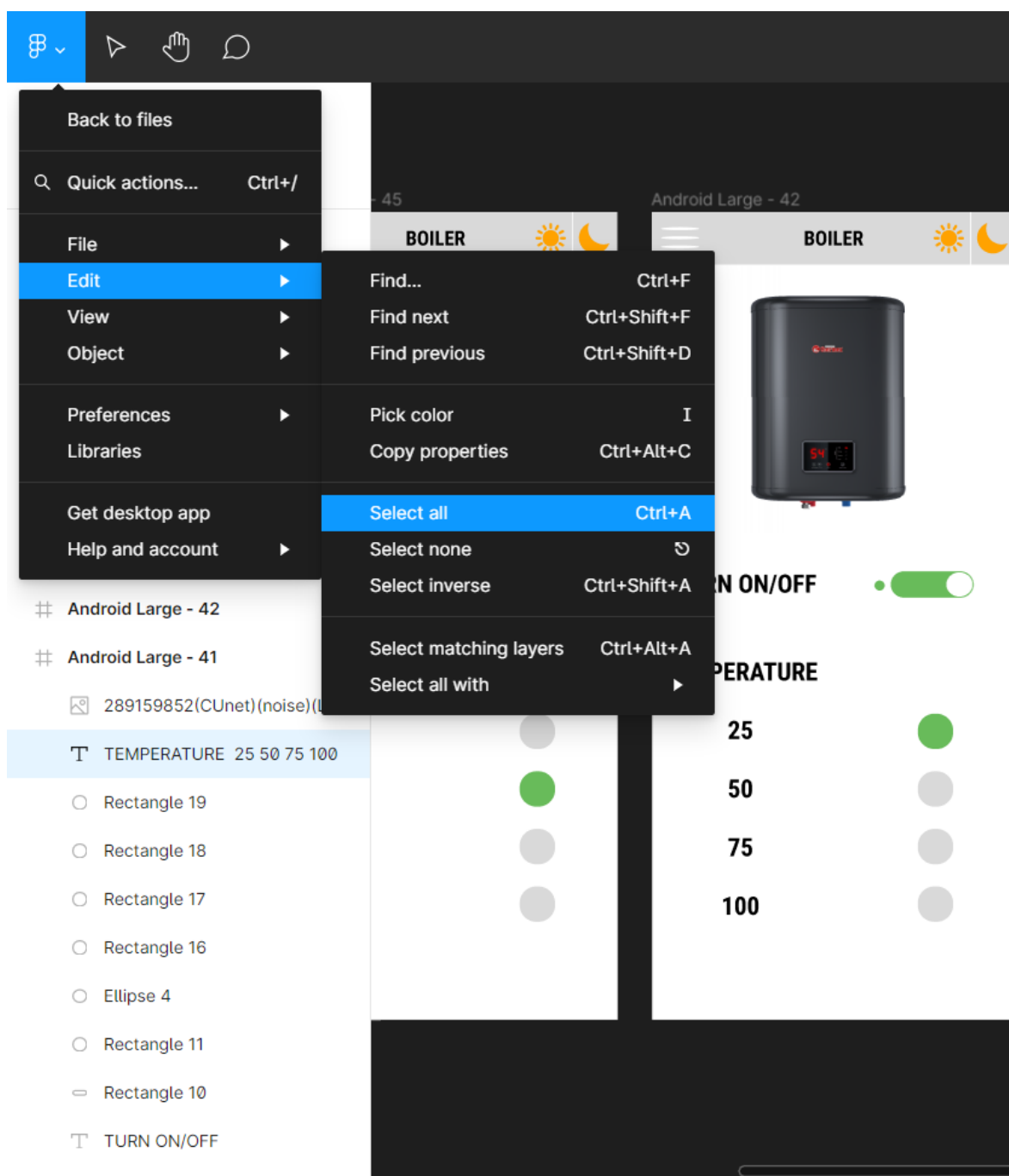


Рисунок 3.4 - налаштування проекту

Android Studio – офіційне інтегроване середовище розробки для платформи Android, яке прийшло на зміну плагіну ADT для платформи Eclipse. Середовище побудоване на базі вихідного коду продукту IntelliJ IDEA Community Edition, від компанії JetBrains, і розвивається в рамках відкритої моделі розробки та поширюється під ліцензією Apache 2.0.

Середовище є крос-платформним, ним можна користуватись на машинах під керуванням Linux (для тестування використовується Ubuntu),

macOS, Windows, Chrome OS. Android Studio надає засоби для розробки застосунків не тільки для смартфонів і планшетів, але й для носимих пристроїв на базі Wear OS, телевізорів з Android TV, окулярів віртуальної та доповненої реальності, і автомобільних інформаційно-розважальних систем з Android Auto. Для застосунків, які розроблялись з використанням Eclipse і ADT Plugin, передбачено інструмент для автоматичного імпорту існуючого проекту в Android Studio.

Android Studio адаптована для виконання типових завдань розробки застосунків для Android – до складу середовища включені засоби для спрощення тестування програм на сумісність з різними версіями платформи та інструменти для проектування застосунків, які працюють на пристроях з екранами різної роздільної здатності (планшети, смартфони, ноутбуки, годинники, окуляри тощо). Окрім можливостей, присутніх в IntelliJ IDEA, в Android Studio реалізовано кілька додаткових функцій – наприклад, нова уніфікована підсистема збірки, тестування і розгортання застосунків, заснована на інструментарії Gradle, та з підтримкою засобів безперервної інтеграції.

Для прискорення розробки IDE надає колекцію типових елементів інтерфейсу та візуальний редактор для їх компоновання (на жаль, неідеальний, як і в оригінальній IntelliJ IDEA, але його можливостей буде достатньо для більшості користувачів), який надає зручний попередній перегляд різних станів інтерфейсу (наприклад, можна подивитися як застосунок буде виглядати в різних версіях Android і на екранах різних розмірів). Для створення нестандартних інтерфейсів надається майстер створення власних інтерфейсних елементів, який підтримує використання шаблонів. Також до середовища були додані функції для завантаження типових прикладів коду з GitHub.

До складу Android Studio також включені модифіковані з урахуванням особливостей платформи Android розширені інструменти рефакторингу, перевірки сумісності з попередніми версіями, виявлення проблем з

продуктивністю, моніторингу споживання пам'яті та оцінки зручності використання. До редактору додано режим швидкого внесення правок. Система підсвічування синтаксису, статичного аналізу та виявлення помилок розширена підтримкою Android API. Інтегровано також підтримку оптимізатора коду ProGuard, вбудовані засоби генерації цифрових підписів та надано інтерфейс для керування локалізацією застосунків.

Разом з тим, системні вимоги середовища є досить скромними – на відміну від, наприклад, Microsoft Visual Studio, з Android Studio можна цілком комфортно працювати навіть на пересічних офісних ПК.

Для роботи над застосунком було використано портативну Android Studio версії 4.2.1 від PortApps з вбудованою OpenJDK 64 біт, що дозволило виконувати повний спектр завдань з розробки мобільних застосунків без необхідності повноцінного встановлення IDE. Загальний вигляд Android Studio, запущеної на нашій машині, показано на рисунку 3.5.

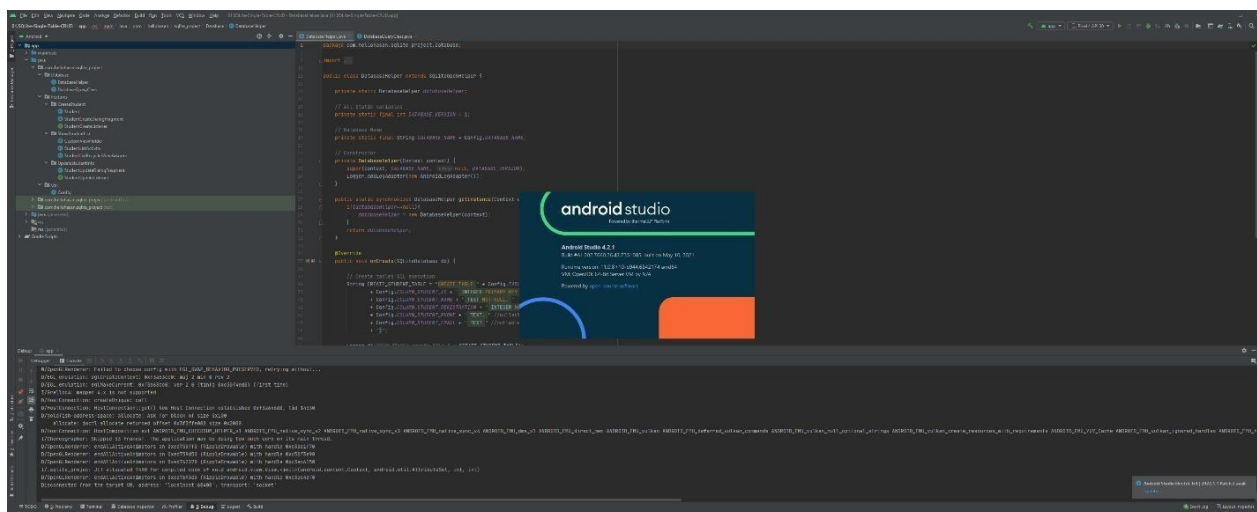


Рисунок 3.5 - Android Studio в режимі відлагодження застосунку

3.3 Архітектура системи

Типовий додаток для Android містить кілька компонентів, включаючи активності, фрагменти, сервіси, постачальники контенту та приймачі широкомовних повідомлень. Більшість із цих компонентів додатка ви декларуєте у своєму маніфесті додатка. Потім операційна система Android використовує цей

файл, щоб вирішити, як інтегрувати ваш додаток в загальний користувацький досвід на пристрої. Враховуючи, що типовий додаток для Android може містити кілька компонентів, а користувачі часто взаємодіють з кількома додатками протягом короткого часу, додатки повинні адаптуватися до різних видів робочих процесів і завдань, керованих користувачем.

Майте на увазі, що мобільні пристрої також обмежені в ресурсах, тому в будь-який момент операційна система може закрити деякі процеси додатків, щоб звільнити місце для нових.

Враховуючи умови цього середовища, можливо, що ваші компоненти додатка можуть запускатися окремо та не в певному порядку, а операційна система або користувач можуть у будь-який час знищити їх. Оскільки ці події не під вашим контролем, ви не повинні зберігати або утримувати в пам'яті будь-які дані або стан додатка в компонентах додатка, і ваші компоненти додатка не повинні залежати один від одного.

Коли у додатку визначається новий тип даних, необхідно призначити йому єдине джерело правди (Single Source of Truth, SSOT). SSOT є власником цих даних, і тільки SSOT може змінювати або модифікувати їх. Для досягнення цього SSOT надає дані з використанням незмінного типу, а для модифікації даних SSOT надає функції або приймає події, які можуть викликатися іншими типами.

Приклад коду з реалізацією SSOT:

Припустимо, у вас є тип даних User. Створіть репозиторій, який буде джерелом правди для цього типу даних.

```
data class User(val id: String, val name: String)
```

```
class UserRepository {
```

```
    private val userLiveData = MutableLiveData<User>()
```

```
    // Метод для отримання даних користувача
```

```
    fun getUser(): LiveData<User> = userLiveData
```

```

// Метод для оновлення даних користувача
fun updateUser(newUser: User) {
    userLiveData.value = newUser
}

class UserViewModel(private val userRepository: UserRepository) :
ViewModel() {
    val user: LiveData<User> = userRepository.getUser()

    // Метод для оновлення даних користувача через ViewModel
    fun updateUser(name: String) {
        val currentUser = user.value
        if (currentUser != null) {
            val updatedUser = currentUser.copy(name = name)
            userRepository.updateUser(updatedUser)
        }
    }
}

```

В наведеному вище прикладі коду, `UserRepository` є SSOT для типу даних `User`. Тільки репозиторій може змінювати дані користувача, і він надає ці дані як незмінний тип `LiveData`, який може спостерігатися у `ViewModel` та `View`.

З ростом розміру додатків для Android важливо визначити архітектуру, яка дозволяє додатку масштабуватися, підвищує його надійність і полегшує його тестування.

Структура розробки мобільного додатка на базі Android включає в себе 3 шари, які ідуть згори донизу шар користувацького інтерфейсу, доменів, та шар даних.

На рівні інтерфейсу користувача відображаються екран з даними програми, при зміні яких інтерфейс повинен оновлюватись, цей шар складається з двох

частин – елементів інтерфейсу та State Holder, функція якого збереження даних, обробка логіки та надання даних до інтерфейсу.

Доменний рівень включає в себе бізнес-логіку, яка повторно використовується кількома моделями перегляду, він є необов'язковим шаром, і частіше використовується лише в складних програмах. Він включає в себе класи які є варіантами використання, інтеракторами, які відповідають за певну функціональність.

Рівень даних містить сховища, що включають джерела даних, що диференціюються залежно від типу та виконують функції надання доступу даних до програми, вирішення конфліктів між джерелами, опис бізнес-логіки, відокремлення даних від решти програми. Джерела поділяються на внутрішні – локальна база даних, та зовнішні – передусім мережа Інтернет.

3.3.1 Модульна архітектура

Модульна архітектура є важливим аспектом проектування системи розумного дому. Вона визначає структуру системи як набір взаємозалежних модулів, які можуть бути розроблені, тестовані, впроваджені та підтримувані окремо.

Основними перевагами модульної архітектури є зменшення складності, підвищення швидкості розробки, полегшення впровадження нових функцій та забезпечення легкості та гнучкості утримання.

Нижче будуть наведені основні аспекти модульної архітектури:

- Модулі: Вся система розумного дому розбивається на окремі модулі, кожен з яких виконує певну функцію або набір функцій. Наприклад, можливі модулі: керування освітленням, керування температурою, безпека, інтеграція з голосовим асистентом тощо.

- Модульність: Кожен модуль повинен бути якомога більш незалежним, що дозволяє розробникам працювати над ними окремо без впливу на інші частини системи.
- Інтерфейси: Кожен модуль має чітко визначений інтерфейс, через який він взаємодіє з іншими модулями. Це дозволяє розділити відповідальності між модулями та забезпечити їх взаємодію без прямого залежності.
- Розширюваність: Модульна архітектура сприяє легкості в розширенні системи шляхом додавання нових модулів або модифікації існуючих, не впливаючи на інші частини системи.
- Тестування та утримання: Кожен модуль може бути протестований окремо, що спрощує виявлення та виправлення помилок. Крім того, модульна архітектура полегшує підтримку системи, оскільки проблеми можуть бути локалізовані до конкретного модуля.
- Патерни проектування: При проектуванні модульної архітектури можна використовувати патерни проектування, такі як Singleton, Factory, Observer тощо, для забезпечення ефективності та гнучкості системи.
- Розділення відповідальності: Кожен модуль в системі відповідає за виконання певної конкретної функції або набору функцій. Це дозволяє зосередитися на реалізації конкретного функціоналу без необхідності розуміння всієї системи.
- Залежності: Модулі можуть бути залежними один від одного, але ця залежність має бути мінімізована. Це досягається за допомогою визначення чітких інтерфейсів, через які модулі спілкуються.

- Перевикористання коду: Чітко визначені модулі дозволяють зручно перевикористовувати код. Якщо функціональність потрібна в іншій частині системи, розробники можуть просто використати відповідний модуль замість переписування коду з нуля.

3.4 Тестування

Тестування – це процес оцінки системи або її компонентів з метою виявлення, чи відповідають вони встановленим вимогам та очікуванням. Тестування забезпечує виявлення помилок, дефектів або вразливостей у програмному забезпеченні, що допомагає підвищити його якість, надійність та безпеку.

Мета тестування:

- Виявлення дефектів: Знайти помилки або дефекти до того, як програмне забезпечення буде використано кінцевими користувачами.
- Забезпечення якості: Підтвердити, що програмне забезпечення відповідає визначеним вимогам та стандартам.
- Підтвердження відповідності: Переконатися, що система працює згідно з очікуваннями замовника або специфікацією.
- Поліпшення продуктивності: Визначити вузькі місця та оптимізувати роботу системи.
- Зниження ризиків: Зменшити ймовірність виникнення проблем під час експлуатації системи.

Тестування програмного забезпечення складається з декількох етапів, кожен з яких має свої завдання та цілі. Нижче наведено докладний опис основних етапів тестування.

ПЛАНУВАННЯ ТЕСТУВАННЯ

На цьому етапі визначаються цілі, обсяг і підхід до тестування. Планування тестування є критично важливим, оскільки воно закладає основу для всього процесу тестування.

Ключові завдання

- Визначення цілей тестування: Чітке розуміння, що саме потрібно досягти за допомогою тестування (наприклад, виявлення дефектів, перевірка відповідності вимогам).
- Визначення обсягу тестування: Визначення меж тестування, які частини системи будуть тестуватися.
- Розробка тестової стратегії: Вибір підходу до тестування, методологій, видів тестування та інструментів.
- Оцінка ресурсів і термінів: Визначення необхідних ресурсів (людських, апаратних, програмних) та встановлення термінів виконання робіт.
- Створення тестового плану: Документування всіх вищезначених аспектів у вигляді тестового плану.

Результати

- Тестовий план, який включає стратегію тестування, обсяг робіт, ресурси, графік та критерії успішності тестування.

РОЗРОБКА ТЕСТОВИХ СЦЕНАРІЇВ

На цьому етапі створюються конкретні тестові сценарії (тест-кейси) на основі вимог до системи. Тестові сценарії визначають конкретні дії, вхідні дані та очікувані результати для кожного тесту.

Ключові завдання

- Аналіз вимог: Вивчення вимог до системи для визначення функцій, які підлягають тестуванню.
- Розробка тестових сценаріїв: Створення тест-кейсів, які охоплюють всі визначені функції та можливі сценарії використання системи.
- Підготовка тестових даних: Визначення та створення наборів даних, необхідних для виконання тестів.
- Автоматизація тестів (за потреби): Написання автоматизованих тестових сценаріїв для виконання повторюваних тестів.

Результати

- Набір тестових сценаріїв та тестових даних, готових до виконання.

ВИКОНАННЯ ТЕСТУВАННЯ

На цьому етапі проводиться фактичне виконання тестових сценаріїв, збір та документування результатів.

Ключові завдання

- Виконання тестів: Запуск ручних або автоматизованих тестів відповідно до плану.
- Збір результатів: Фіксація результатів кожного тесту, включаючи успішні та невдалі тести.
- Виявлення дефектів: Ідентифікація та документування виявлених дефектів для подальшого аналізу та виправлення.
- Регресійне тестування: Перевірка, що виправлені дефекти не вплинули на існуючий функціонал.

Результати

- Звіти про результати тестування, включаючи виявлені дефекти та їх класифікацію.

АНАЛІЗ РЕЗУЛЬТАТІВ ТЕСТУВАННЯ

Аналіз результатів тестування допомагає зрозуміти, наскільки успішно пройдені тести, які проблеми виявлені та як їх виправити.

Ключові завдання

- Аналіз успішності тестів: Перевірка, чи відповідають результати тестів очікуванням.
- Ідентифікація причин дефектів: Аналіз виявлених дефектів для визначення їх причин.
- Класифікація дефектів: Пріоритизація дефектів за ступенем їх впливу на систему (критичні, серйозні, незначні).
- Розробка рекомендацій: Визначення кроків для виправлення дефектів та покращення процесу розробки.

Результати

- Докладний звіт про аналіз результатів тестування та рекомендації щодо виправлення дефектів.

ВИПРАВЛЕННЯ ВИЯВЛЕНИХ ДЕФЕКТІВ

На цьому етапі команда розробників виправляє виявлені дефекти та впроваджує необхідні зміни у систему.

Ключові завдання

- Виправлення дефектів: Внесення змін у вихідний код для усунення виявлених помилок.

- Перевірка виправлень: Виконання повторних тестів для перевірки, що дефекти виправлені (регресійне тестування).
- Документування змін: Фіксація внесених змін для подальшого контролю версій та аналізу.

Результати

- Оновлений код без виявлених дефектів, підтверджений повторним тестуванням.

ЗАВЕРШЕННЯ ТЕСТУВАННЯ

Завершення тестування включає підготовку остаточного звіту про тестування та підтвердження готовності системи до релізу.

Ключові завдання

- Підготовка звіту про тестування: Документування всіх результатів тестування, включаючи виявлені та виправлені дефекти.
- Оцінка якості продукту: Аналіз, чи відповідає система встановленим критеріям якості.
- Розгляд критеріїв завершення тестування: Перевірка, чи досягнуті всі цілі тестування та чи виконані всі тестові сценарії.
- Передача системи в експлуатацію: Підтвердження готовності системи до релізу та передача її у використання.

Результати

- Остаточний звіт про тестування.
- Підтвердження готовності системи до релізу.

Тестування системи "розумний дім" включає в себе ряд етапів, спрямованих на перевірку її функціональності, надійності і зручності використання. Ось деякі кроки, які були виконані під час тестування системи:

- Перевірка базової функціональності: проведена перевірка всіх основних функцій системи, таких як управління освітленням, температурою, безпекою та іншими аспектами. Переконалися, що кожна функція працює так, як очікувалося, і реагує на введення користувача.
- Тестування інтеграції: Оскільки система інтегрується з іншими пристроями та послугами, наприклад, з голосовими асистентами, перевірили, що ця інтеграція працює належним чином.
- Тестування зв'язку: перевірили якість зв'язку між різними пристроями системи, це важливо тому що система використовує бездротові технології, такі як Wi-Fi та Bluetooth. Перевірили, що пристрої можуть взаємодіяти між собою без перешкод.
- Тестування безпеки: Переконалися, що система має належні заходи безпеки для захисту від несанкціонованого доступу та втручання. Протестували датчики безпеки, такі як датчики руху або диму, вони надсилають сповіщення в разі виявлення небезпеки.
- Тестування автоматизації: Оскільки система пропонує функцію автоматизації, наприклад, запуск пральної машини або регулювання температури в певний час. Було перевірено, що ці сценарії працюють належним чином.
- Тестування енергоефективності: Було перевірено, наскільки ефективно система керує енергоспоживанням в будинку. Це може включати

вимірювання споживання електроенергії різними пристроями та оцінку ефективності програм енергозбереження.

- Тестування користувацького досвіду: Запросили користувачів, щоб вони випробували систему і надали зворотний зв'язок щодо її зручності використання, інтуїтивності і загального враження.
- Тестування в реальних умовах: Було частково проведено тестування системи в реальних умовах експлуатації, наприклад, у різних частинах будинку з різними умовами та режимами використання.

3.5 Висновок до розділу

Тестування є складним та багатоетапним процесом, який вимагає ретельного планування, виконання та аналізу. Кожен етап тестування відіграє важливу роль у забезпеченні якості, надійності та безпеки програмного забезпечення. Виконання всіх етапів у повному обсязі дозволяє виявити та виправити дефекти на ранніх стадіях, зменшити ризики та забезпечити успішний реліз продукту.

ВИСНОВКИ

1. В роботі проведено аналіз найпопулярніших систем розумного дому та визначено їх основні переваги та недоліки, які були ретельно проаналізовані для покращення планування програмного продукту.

2. Вивчено архітектурне рішення SSOT, яка передбачає існування єдиного джерела, де зберігаються та з якого отримуються всі дані організації або системи. Основна мета SSOT полягає в забезпеченні точності, узгодженості та актуальності даних, щоб уникнути дублювання та розбіжностей в різних джерелах даних.

3. Розроблений програмний продукт показав, що система "розумний дім" дуже актуальна, та її попит в майбутньому буде лише збільшуватися.

4. Були визначені потреби користувачів і технічних специфікацій, що дозволило створити програмний продукт, який розроблений на базі сучасних технологій, які регулярно підтримуються розробниками, що робить систему більш стабільною, також розроблений продукт максимально відповідає очікуванням ринку.

5. В цілому, робота показала перспективність, конкурентоспроможність та актуальність розробленої системи. Для впровадження та популяризації програмного продукту його необхідно оновлювати, розширювати функціональність програми та підтримувати зворотній зв'язок з користувачами.

ПЕРЕЛІК ПОСИЛАНЬ

1. Що таке розумний будинок. URL: <https://funduk.ua/uk/technoblog/layfkhaki/chto-takoe-umnyy-dom/> (дата звернення: 28.05.2024).
2. Функціональні вимоги. *Visure*. URL: <https://visuresolutions.com/uk/blog/functional-requirements/> (дата звернення: 28.05.2024).
3. Що таке нефункціональні вимоги. *Visure*. URL: <https://visuresolutions.com/uk/blog/non-functional-requirements/> (дата звернення: 28.05.2024).
4. Ajax. *Ajax System*. URL: <https://ajax.systems/ua/> (date of access: 28.05.2024).
5. Ecoisme. *Ecois.me*. URL: <https://ecois.me/> (date of access: 29.05.2024).
6. HoneywellHome Home. *Honeywell*. URL: <https://www.honeywellhome.com/us/en> (date of access: 29.05.2024).
7. Життєвий цикл розробки програмного забезпечення. *EPAM Campus*. URL: <https://training.epam.ua/ua/blog/581> (дата звернення: 29.05.2024).
8. Інтеграція SONOFF та Matter. *HomeSmart+*. URL: <https://homesmart.com.ua/yntehratsyia-sonoff-ta-matter/> (дата звернення: 28.05.2024).
9. ANDROID STUDIO. URL: <https://developer.android.com/studio> (date of access: 25.05.2024).
10. FIGMA. URL: <https://www.figma.com/> (date of access: 29.05.2024).
11. Архітектура програмного забезпечення. *Wezom*. URL: <https://wezom.com.ua/ua/blog/arhitektura-programmnogo-obespecheniya> (дата звернення: 29.05.2024).
12. Типи тестування. *EPAM Campus*. URL: <https://rdp.epam.com/ua/blog/469> (дата звернення: 29.05.2024).

Додаток А

Лістинг програми:

```
package com.anton s application.app.modules.android large six.ui
import android.activity.viewModels
import com.anton s application.app.R
import com.anton s application.app.app components.base.BaseActivity
import com.anton s application.app.data binding.Activity Android Large Six Binding
import com.anton s application.app.modules.android large six.
data".viewmodel.Android Large Six VM
import kotlin. String
import kotlin. Unit
class Android Large Six Activity :
Base Activity Activity Android Large Six
Binding>(R.layout.activity_android_large_six) {
private val viewModel: Android Large Six VM by viewModels<Android Large Six
VM>()
override fun uninitialized(): Unit
{
viewModel.nav Arguments = intent.extras?.getBundle("bundle")
binding.android Large Six VM = viewModel
override fun setUp Clicks(): Unit {
companion object {
const val TAG: String
- "ANDROID_LARGE_SIX_ACTIVITY"
package com.anton s application.app.modules.frame one.ui
import android.activity.viewModels
import com.anton s application.app.R
import com.anton s application.app.app components.base.BaseActivity
import com.anton s application.app.data binding.Activity Frame One Binding
import com.anton s application.app.modules.frame one."data".viewmodel.Frame One
VM
import kotlin.String
import kotlin. Unit
classFrame One Activity : BaseActivity<Activity Frame One
Binding>(R.layout.activity_frame_one)
private var viewModel: Frame One VM by viewModels<Frame One VM>()
{
```

```

override fun uninitialized(): Unit {
    viewModel.nav Arguments = intent.extras?.getBundle("bundle")
    binding.frameOneVM = viewModel
    override fun setUpClicks(): Unit {
        companion object {
            const val TAG: String = "FRAME_ONE_ACTIVITY"
            package com.anton.s.application.app.modules.frameone."data".viewmodel
            import android.os.Bundle
            import android.lifecycle.MutableLiveData
            import android.lifecycle.ViewModel
            import com.anton.s.application.app.modules.frameone."data".model.FrameOneModel
            import org.koin.core.KoinComponent
            class FrameOneVM : ViewModel(), KoinComponent {
                val frameOneModel: MutableLiveData<FrameOneModel> = MutableLiveData(FrameOneModel())
                var newArguments: Bundle? = null
            }
            package com.anton.s.application.app.modules.androidlarge2020.ui
            import android.activity.viewModels
            import com.anton.s.application.app.R
            import com.anton.s.application.app.components.base.BaseActivity
            import com.anton.s.application.app.databinding.ActivityAndroidLarge2020Binding
            import com.anton.s.application.app.modules.androidlarge2020."data".viewmodel.AndroidLarge2020VM
            import kotlin.String
            import kotlin.Unit
            class AndroidLarge2020Activity :
                BaseActivity<ActivityAndroidLarge2020Binding>(R.layout.activity_android_large_2020) {
                private val viewModel: AndroidLarge2020VM by viewModels<AndroidLarge2020VM>()
                override fun uninitialized(): Unit {
                    viewModel.nav Arguments = intent.extras?.getBundle("bundle")
                    binding.androidLarge2020VM = viewModel
                    override fun setUpClicks(): Unit {
                        companion object {
                            const val TAG: String = "ANDROID_LARGE_TWO_ACTIVITY"
                            package com.anton.s.application.app.modules.androidlarge2020.ui

```

```

import android.activity.viewModels
import com.anton s application.app.R
import com.anton s application.app.app components.base.BaseActivity
import com.anton s application.app.databinding.Activity Android
i.largeTwentytwoBinding
import com.anton s application.app.modules.android large twenty
two."data".viewModel. Android Large Twenty two VM
import kotlin. String
import kotlin.Unit
class Android Large Twenty two Activity :
Base Activity<Activity Android Large Twenty two
Binding>(R.layout.activity_android_large_twenty two)
private val viewModel: Android Large Twenty two VM by viewModels<Android
Large Twenty two VM>()
override fun uninitialized(): Unit {
viewModel.nav Arguments = intent.extras?.getBundle("bundle")
binding.android Large Twenty two VM - viewModel
}
override fun setUp Clicks(): Unit (
companion object
const val TAG: String - "ANDROID_LARGE_TWENTY TWO_ACTIVITY"
package com.anton s application.app.modules.android large twenty two."data".model
import com.anton s application.app.R
import com.anton s application.app.app components.di.MyApp
import kotlin. String
data class Android Large Twenty two Model(
•
TODO Replace with dynamic value
var txt Teapot: String? - MyApp.getInstance().resources.getString(R.string.lbl_teapot)
T000 Replace with dynamic value
var txt Turn on off: String? =
Myapp.getInstance().resources.getString(R.string.lbl_turn_on_off)
• TOD0 Replace with dynamic value
var txt Temperature tw en: String? -
MyApp.getInstance().resources.getstring(R.string.msg_temperature_25_50_75_100)
package com.anton s application.app.modules.android large twenty six."data".model
import com.anton s application.app.R
import com.anton s application.app.app components.di.MyApp

```

```

import kotlin.String
data class Android Large Twenty six Model(
    TODO Replace with dynamic value
    var txt Teapot: String? = MyApp.getInstance().resources.getString(R.string.lb1_teapot)
    /...
    TODO Replace with dynamic value
    var
    txt Turn on off: String? =
    MyApp.getInstance().resources.getString(R.string.lb1_turn_on_off)
    TODO Replace with dynamic value
    var txt Temperature tw en: String? =
    MyApp.getInstance().resources.getString(R.string.msg_temperature_25_5e_75_100)
package com.anton s application.app.modules.android large twelve.ui
import android.activity.viewModels
import com. anton s application.app.R
import com.anton s application.app.app components.base.BaseActivity
import com.anton s application.app.data binding.Activity Android Large Twelve
Binding
import com.anton s application.app.modules.android large twelve.
data".viewModel.Android Large Twelve VM
import kotlin.String
import kotlin.Unit
class Android Large Twelve Activity :
Base Activity<Activity Android Large Twelve
Binding>(R.layout.activity_android_large_twelve)
private val viewModel: Android Large Twelve VM by viewModels<Android Large
Twelve VM>()
{
    override fun uninitialized(): Unit {
        viewModel.nav Arguments = intent.extras?.getBundle("bundle")
        binding.android Large Twelve VM =
        viewModel
    }
    override fun setUp Clicks(): Unit {
        companion object {
            const val TAG: String = "ANDROID_LARGE_TWELVE_ACTIVITY"
        }
package com.anton s application.app.modules.androidlargetwenty. data".model
import com. antonsapplication.app.R
import com. antonsapplication.app.appcomponents.di.MyApp

```

```

import kotlin.String
data class AndroidLargeTwentyModel(
/**
TODO Replace with dynamic value
var txtMenu: String? - MyApp.getInstance().resources.getString(R.string.lbl_menu)
•
TODO Replace with dynamic value
var txtLamp: String? - MyApp.getInstance().resources.getString(R.string.lbl_lamp)
/*
*TODO Replace with dynamic value
var txtTurnonoff: String? =
MyApp.getInstance().resources.getString(R.string.lbl_turn_on_off)
•
TODO Replace with dynamic value
var txtRecommended: String? -
MyApp.getInstance().resources.getString(R.string.msg_recommended_colors)
•
TODO Replace with dynamic value
var txtWarmwhite: String? =
MyApp.getInstance().resources.getString(R.string.lbl_warm_white)
/**
*TODO Replace with dynamic value
var txtDaywhite: String? - MyApp.getInstance().resources.getString(R.string.lbl_da
_day_white)
TODO Replace with dynamic value
var txtColdwhite: String? =
MyApp.getInstance().resources.getString(R.string.lbl_cold_white)
•
TODO Replace with dynamic value
var txtCategories: String? -
MyApp.getInstance().resources.getString(R.string.lbl_categories)
•
TODO Replace with dynamic value
var txtLampOne: String? -
MyApp.getInstance().resources.getString(R.string.lbl_lamp)
/**
*TODO Replace with dynamic value

```

```

var txtHeating: String? -
MyApp.getInstance().resources.getString(R.string.lbl_heating)
TODO Replace with dynamic value
var txtTeapot: String? - MyApp.getInstance().resources.getString(R.string.lbl_teapot)
android:layout_width="@dimen/_30pxh"
android:scaleType="fitxy"
android:src="@drawable/img_moon_icon_512x512_fm9crgpt"
tools:ignore="ContentDescription"
tools:src="@drawable/img_moon_icon_512x512_fm9crgpt"/>
</LinearLayout>
</androidx.appcompat.widget.Toolbar>
<LinearLayout android:id="@+id/linearColumn2ehb@1tte"
android:layout_height="wrap_content"
android:layout_width="match_parent"
android:orientation="vertical"
android:paddingBottom="@dimen/_7pxv"
android:paddingEnd="@dimen/_21pxh"
android:paddingStart="@dimen/_21pxh" android:paddingTop="@dimen/_7pxv">
<ImageView android:id="@+id/image2ehbe1ttecfoo"
android:layout_gravity="center_horizontal"
android:layout_height="@dimen/_279pxh"
android:layout_width="@dimen/_267pxh"
android:scaleType="fitxy"
imageRes="@(@drawable/img_2ehb01_ttec_foo)"
tools:ignore="ContentDescription" tools:src="@drawable/img_2ehb01_ttec_foo"/>
<LinearLayout android:id="@+id/linearRowturnonoff"
android:layout_height="wrap_content"
android:layout_marginEnd="@dimen/_25pxh"
android:layout_marginTop="@dimen/_15pxv"
android:layout_width="match_parent" android:orientation="horizontal">
<TextView android:ellipsize="end"
android:id="@+id/txtTurnonoff"
android:layout_gravity="center_vertical"
android:layout_height="wrap_content"
android:layout_width="wrap_content"
android:singleLine="true"
android:text="@ (androidLarge105VM.androidLarge105Model.txtTurnonoff)"
style="@style/txtRobotoCondensedbold25" tools:text="@string/lbl_turn_on_off"/>

```

```

<View
    android:background="@drawable/rectangle_bg_blue_gray_100_01_radius_5"
    android:id="@+id/viewView"
    android:layout_gravity="bottom"
    android:layout_height="@dimen/_10pxh"
    android:layout_marginBottom="@dimen/_8pxv"
    android:layout_marginStart="@dimen/_51pxh"
    android:layout_marginTop="@dimen/_11pxv"
    android:layout_width="@dimen/_10pxh"/>
    <com.google.android.material.switchmaterial.SwitchMaterial
        android:id="@+id/switchMaterialSwitch"
        android:layout_gravity="center_vertical"
        android:layout_height="wrap_content"
        android:layout_marginStart="@dimen/_6pxh"
        android:layout_marginTop="@dimen/_2pxv" android:layout_width="wrap_content"/>
    </LinearLayout>
    <LinearLayout android:gravity="center_vertical"
        android:id="@+id/linearRowfeed2040eigh"
        android:layout_height="wrap_content"
        android:layout_marginBottom="@dimen/_5pxv"
        android:layout_marginEnd="@dimen/_40pxh"
        android:layout_marginStart="@dimen/_40pxh"
        android:layout_marginTop="@dimen/_56pxv"
        android:layout_width="match_parent" android:orientation="horizontal">
        <TextView android:gravity="center_vertical"
            android:id="@+id/txtFeed2040eighty"
            android:layout_height="wrap_content"
            android:layout_marginBottom="@dimen/_1pxv"
            android:layout_weight="1"
            android:layout_width="match_parent"
            android:text="@{(androidLarge105VM.androidLarge105Model.txtFeed2040eighty)}"
            android:id="@+id/viewView"
            android:
            layout_gravity="bottom"
            android:layout_height="@dimen/_10pxh"
            android:layout_marginBottom="@dimen/_8pxv"
            android:layout_marginStart="@dimen/_51pxh"

```

```

android:layout_marginTop="@dimen/_11pxv"
android:layout_width="@dimen/_10pxh"/>
<com.google.android.material.switchmaterial.SwitchMaterial
android:id="@+id/switchMaterialSwitch"
android:layout_gravity="center_vertical"
android:layout_height="wrap_content"
android:layout_marginStart="@dimen/_6pxh"
android:layout_marginTop="@dimen/_2pxv" android:layout_width="wrap_content"/>
</LinearLayout>
<LinearLayout android:gravity="center_vertical"
android:id="@+id/linearRowfeed2040eigh"
android:layout_height="wrap_content"
android:layout_marginBottom="@dimen/_5pxv"
android:layout_marginEnd="@dimen/_40pxh"
android:layout_marginStart="@dimen/_40pxh"
android:layout_marginTop="@dimen/_56pxv"
android:layout_width="match_parent" android:orientation="horizontal">
<TextView android:gravity="center_vertical"
android:id="@+id/txtFeed2040eighty"
android:layout_height="wrap_content"
android:layout_marginBottom="@dimen/_1pxv"
android:layout_weight="1"
android:layout_width="match_parent"
android:text="@{androidLarge105VM.androidLarge105Model.txtFeed2040eighty}"
style="@style/txtRobotoCondensedbold25"
tools:text="@string/msg_feed_20_40_80"/>
<LinearLayout android:id="@+id/linearColumnviewOne"
android:layout_height="wrap_content"
android:layout_marginStart="@dimen/_149pxh"
android:layout_marginTop="@dimen/_56pxv"
android:layout_width="wrap_content" android:orientation="vertical">
<View
android:background="@drawable/rectangle_bg_blue_gray_100_01_radius_16"
android:id="@+id/viewViewOne"
android:layout_gravity="center_horizontal"
android:layout_height="@dimen/_35pxh" android:layout_width="@dimen/_35pxh"/>
<View
android:background="@drawable/rectangle_bg_blue_gray_100_01_radius_16"

```

```

android:id="@+id/viewViewTwo"
android:layout_gravity="center_horizontal"
android:layout_height="@dimen/_35pxh"
android:layout_marginTop="@dimen/_22pxv"
android:layout_width="@dimen/_35pxh"/>
<View
android:background="@drawable/rectangle_bg_blue_gray_100_01_radius_16"
android:id="@+id/viewViewThree"
android:layout_gravity="center_horizontal"
android:layout_height="@dimen/_35pxh"
android:layout_marginTop="@dimen/_22pxv"
android:layout_width="@dimen/_35pxh"/>
</LinearLayout>
</LinearLayout>
</LinearLayout>
</LinearLayout>
</layout>
<?xml version="1.0" encoding="UTF-8"?>
<layout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto" xmlns:tools="
http://schemas.android.com/tools"
    >
    <data>
    <variable name="androidLarge106VM"
type="com.antonsapplication.app.modules.androidlarge106.data.viewmodel.AndroidL
arge106VM"/>
    </data>
    <LinearLayout android:id="@+id/linearAndroidLarge"
android:layout_height="match_parent"
android:layout_width="match_parent"
android:orientation="vertical" style="@style/groupStylegray_800">
    <androidx.appcompat.widget.Toolbar
android:gravity="center_horizontal"
android:id="@+id/toolbarToolbar"
android:layout_height="wrap_content"
android:layout_width="match_parent"
app:contentInsetStart="@dp" style="@style/groupStyleblue_gray_100_01">

```

```

<LinearLayout android:gravity="center_vertical"
android:id="@rid/linearRowsettingsOne"
android:layout_height="wrap_content"
android:layout_marginEnd="@dimen/_8pxh"
android:
layout_marginStart="@dimen/_8pxh"
android:layout_width="match_parent" android:orientation="horizontal">
<ImageView android:id="@+id/imageSettingsOne"
android:layout_gravity="center_vertical"
android:layout_height="@dimen/_21pxh"
android:layout_marginBottom="@dimen/_16pxv"
android:layout_marginTop="@dimen/_15pxv"
android:layout_width="@dimen/_38pxh"
android:scaleType="fitxy"
android:src="@drawable/img_settings"
tools:ignore="ContentDescription" tools:src="@drawable/img_settings"/>
<TextView android:ellipsize="end"
android:id="@+id/txtFeeding"
android:layout_gravity="center_vertical"
android:
layout_height="wrap_content"
android:layout_marginBottom="@dimen/_14pxv"
android:layout_marginStart="@dimen/_97pxh"
android:layout_marginTop="@dimen/_13pxv"
android:layout_width="wrap_content"
android:singleLine="true"
android:text="@{androidLarge106VM.androidLarge106Model.txtFeeding)"
style="@style/txtRobotoCondensedbold20" tools:text="@string/lb1_feeding"/>
<ImageView android:id="@+id/imageImage"
android:layout_gravity="center_vertical"
android:layout_height="@dimen/_30pxh"
android:layout_marginBottom="@dimen/_11pxv"
android:layout_marginStart="@dimen/_63pxh"
android:layout_marginTop="@dimen/_11pxv"
android:layout_width="@dimen/_30pxh"
android:scaleType="fitxy"
android:src="@drawable/img_9055356_1"
tools:ignore="ContentDescription" tools:src="@drawable/img_9055356_1"/>

```

```

<View android:background="@color/white_A700"
android:id="@rid/lineLine"
android:layout_gravity="center_vertical"
android:layout_height="@dimen/_52pxv"
android:layout_marginStart="@dimen/_6pxh" android:
layout_width="@dimen/_1pxh"/>
<ImageView android:id="@+id/imageMooniconOne"
android:layout_gravity="center_vertical"
android:layout_height="@dimen/_30pxh"
android:layout_marginBottom="@dimen/_11pxv"
android:layout_marginStart="@dimen/_6pxh"
android:layout_marginTop="@dimen/_11pxv"
android:layout_width="@dimen/_30pxh"
android:layout_width="@dimen/_30pxh"
android:scaleType="fitXY"
android:src="@drawable/img_moon_icon_512x512_fm9crgpt"
tools:ignore="ContentDescription"
tools:src="@drawable/img_moon_icon_512x512_fm9crgpt"/>
</LinearLayout>
</androidx.appcompat.widget.Toolbar>
<LinearLayout android:gravity="start"
android:id="@+id/linearColumnfeed2040e"
android:layout_height="wrap_content"
android:layout_width="match_parent"
android:orientation="vertical"
android:paddingBottom="@dimen/_7pxv"
android:paddingEnd="@dimen/_21pxh"
android:paddingStart="@dimen/_21pxh" android:paddingTop="@dimen/_7pxv">
<LinearLayout android:gravity="end"
android:id="@+id/linearColumn2ehb0itte"
android:layout_height="wrap_content"
android:layout_marginEnd="@dimen/_20pxh"
android:layout_width="wrap_content" android:orientation="vertical">
<ImageView android:id="@+id/image2ehb01ttecfoo"
android:layout_height="@dimen/_279pxh"
android:layout_marginEnd="@dimen/_4pxh"
android:layout_width="@dimen/_267pxh"
android:scaleType="fitXY"

```

```

imageRes-"@(@drawable/ing_2ehb01_ttec_foo)"
tools:ignore="ContentDescription"
tools:src="@drawable/img_2ehb01_ttec_foo"/>
<LinearLayout android: gravity="end"
android:id="@+id/linearRowturnonoff"
android:layout_height="wrap_content"
android:layout_marginTop="@dimen/_15pxv"
android:layout_width="wrap_content" android:orientation="horizontal">
<TextView android:ellipsize="end"
android:id="@+id/txtTurnonoff"
android:layout_gravity="center_vertical"
android:layout_height="wrap_content"
android:layout_width="wrap_content"
android:singleLine="true"
android:text="@ (androidLarge106VM.androidLarge106Model.txtTurnonoff)"
style="@style/txtRobotoCondensedbold25" tools:text="@string/lbl_turn_on_off"/>
<View
android:background="@drawable/rectangle_bg_green_400_01_radius_5"
android:id="@+id/viewView"
android:layout_gravity="bottom"
android:layout_height="@dimen/_10pxh"
android:layout_marginBottom="@dimen/_8pxv"
android:layout_marginStart="@dimen/_56pxh"
android:layout_marginTop="@dimen/_11pxv"
android:layout_width="@dimen/_10pxh"/>
<com.google.android.material.switchmaterial.SwitchMaterial
android:id="@+id/switchMaterialSwitch"
android:layout_gravity="center_vertical"
android:layout_height="wrap_content"
android:layout_marginStart="@dimen/_6pxh"
android:layout_marginTop="@dimen/_2pxv"
android:
layout_width="wrap_content"/>
</LinearLayout>
</LinearLayout>
<LinearLayout android:gravity="center_vertical"
android:id="@+id/linearRowfeed2048eigh"
android:layout_height="wrap_content"

```

```

android:layout_marginBottom="@dimen/_5pxv"
android:layout_marginEnd="@dimen/_40pxh"
android:layout_marginStart="@dimen/_40pxh"
android:layout_marginTop="@dimen/_56pxv"
android:layout_width="match_parent" android:orientation="horizontal">
android:layout_gravity="center_vertical"
android:layout_height="wrap_content"
android:layout_marginStart="@dimen/_6pxh"
android:layout_marginTop="@dimen/_2pxv" android:layout_width="wrap_content"/>
</LinearLayout>
</LinearLayout>
<LinearLayout android:gravity="center_vertical"
android:id="@+id/linearRowfeed2040eigh"
android:layout_height="wrap_content"
android:layout_marginBottom="@dimen/_5pxv"
android:layout_marginEnd="@dimen/_40pxh"
android:layout_marginStart="@dimen/_48pxh"
android:layout_marginTop="@dimen/_56pxv"
android:layout_width="match_parent" android:orientation="horizontal">
<TextView android:gravity="center_vertical"
android:id="@+id/txtFeed2040eighty"
android:layout_height="wrap_content"
android:layout_marginBottom="@dimen/_1pxv"
android:layout_weight="1"
android:layout_width="match_parent"
android:text="@{androidLarge106VM.androidLarge106Model.txtFeed2040eighty}"
style="@style/txtRobotoCondensedbold25"
tools:text="@string/msg_feed_20_40_80"/>
<LinearLayout android:id="@+id/linearColumnviewOne"
android:
layout_height="wrap_content"
android:layout_marginStart="@dimen/_149pxh"
android:layout_marginTop="@dimen/_56pxv"
android:layout_width="wrap_content" android:orientation="vertical">
<View
android:background="@drawable/rectangle_bg_blue_gray_100_01_radius_16"
android:id="@+id/viewViewOne"
android:layout_gravity="center_horizontal"

```

```

        android:layout_height="@dimen/_35pxh" android:
        layout_width="@dimen/_35pxh"/>
    <View
        android:background="@drawable/rectangle_bg_green_400_01_radius_16"
        android:id="@+id/viewViewTwo"
        android:layout_gravity="center_horizontal"
        android:layout_height="@dimen/_35pxh"
        android:layout_marginTop="@dimen/_22pxv"
        android:layout_width="@dimen/_35pxh"/>
    <View
        android:background="@drawable/rectangle_bg_blue_gray_100_01_radius_16"
        android:id="@+id/viewViewThree"
        android:layout_gravity="center_horizontal"
        android:layout_height="@dimen/_35pxh"
        android:layout_marginTop="@dimen/_22pxv" android:
        layout_width="@dimen/_35pxh"/>
</LinearLayout>
</LinearLayout>
</LinearLayout>
</LinearLayout>
</layout>
<?xml version="1.0" encoding="UTF-8"?>
<layout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto" xmlns:tools="
    http://schemas.android.com/tools">
    <data>
        <variable name="
        android:targetSdkVersion"
        type="com.antonsapplication.app.modules.androidlarge107.data.viewmodel.AndroidId"
        </data>
        <FrameLayout android:id="@+id/frameAndroidLarge"
            android:layout_height="match_parent"
            android:layout_width="match_parent" style="@style/groupStylewhite_4700">
            <LinearLayout android:id="@+id/linearColumnTurnOff"
                android:layout_gravity="bottom|start"
                android:layout_height="wrap_content"
                android:layout_marginBottom="@dimen/_155pxv"
                android:layout_marginEnd="@dimen/_196pxh"

```

```

android:layout_marginStart="@dimen/_21pxh-
android:layout_width="match_parent" android:orientation="vertical">
<TextView android:ellipsize="end"
android:id="@id/txtTurnonoff"
android:layout_height="wrap_content"
android:layout_width="match_parent"
android:singleLine="true"
android:text="@{(androidtarget107VM, androidtarget107Model.txtTurnonoff)"
style="style/txtRobotoCondensedbold25" tools:text="@string/lbl_turn_on_off"/>
<TextView android:id="@+id/txtFeed2040eighty"
android:layout_height="wrap_content"
android:layout_marginStart="@dimen/_40pxh"
android:layout_marginTop="@dimen/_50pxv"
android:layout_width="wrap_content"
android:text="@{fandroidLarge107vM, androidLarge107model.txtfeed2040eighty)"
style="@style/txtRobotoCondensedbold25" tools:text="@string/esg_feed_20.40_80"/>
</LinearLayout>
<LinearLayout android:id="@+id/linearColumnview"
android:layout_gravity="bottom"
android:layout_height="wrap_content"
android:layout_marginBottom="@dimen/_154pxv"
android:layout_marginEnd="@dimen/_47pxh"
android:layout_marginStart="@dimen/_215pxh"
android:layout_width="match_parent" android:orientation="vertical">
<LinearLayout android:gravity="end"
android:id="@+id/linearRowview"
android:layout_height="wrap_content"
android:layout_width="match_parent" android:orientation="horizontal">
roid:background="@drawable/rectangle_bg_blue_gray_100_01_radius_s
android:id="@+id/viewlew"
android:layout_gravity="bottom"
android:layout_height="@dimen/_10pxh"
android:layout_marginBottom="@dimen/_7pxv"
android:layout_marginTop="@dimen/_9pxv" android:
layout_width="@dimen/_10pxh"/>
<com.google.android.material.switchmaterial.SwitchMaterial
android:id="@+id/switchMaterialSwitch"
android:layout_gravity="center_vertical"

```

```
android:layout_height="wrap_content"
android:layout_marginStart="@dimen/_spxh" android:
layout_width="wrap_content"/>
</LinearLayout>
```

View

```
android:background="@drawable/rectangle_big_blue_gray_100_01_radlus_16"
android:id="#rid/viewWiewOne"
android:layout_height="@disen/_35pxh-
android:layout_marginEnd="@dimen/_15pkh-
android:layout_marginTop="@dimen/_115pxv"
android:layout_width="@dimen/_35pxh"/>
```

<View

```
android:background="@drawable/rectangle_bg_blue_gray_100_01_radius_16"
android:layout_gravity=start
android:layout_height="@dimen/_800pxv"
android:layout_width="@dimen/_108pxh">
<LinearLayout android:id="@+id/linearColumncategorie"
android:layout_gravity="bottom| center"
android:layout_height="wrap_content"
android:layout_marginBottom="@dimen/_17pxv"
android:layout_marginEnd="@dimen/_7pxh"
android:layout_marginStart="@dimen/_7pxh"
android:layout_marginTop="@dimen/_100pxv"
android:layout_width="match_parent" android:orientation="vertical">
<TextView android:ellipsize="end"
android:gravity="center_horizontal"
android:id="@+id/txtCategories"
android:layout_height="wrap_content"
android:layout_width="match_parent"
android:singleLine="true"
android:text="@{androidLarge107VM.androidLarge107Model.txtCategories}"
style="@style/txtRobotoCondensedbold17" tools:text="@string/lb1_categories"/>
<TextView android:ellipsize="end"
android:gravity="center_horizontal"
android:id="@+id/txtLamp"
android:layout_height="wrap_content"
android:
layout_marginTop="@dimen/_61pxv"
```

```

android:layout_width="match_parent"
android:singleLine="true"
android:text="@{androidLarge107VM.androidLarge107Model.txtLamp}"
style="@style/txtRobotoCondensedbold25" tools:text="@string/lbl_lamp"/>
<TextView android:ellipsize="end"
android:gravity="center_horizontal"
android:id="@+id/txtHeating"
android:layout_height="wrap_content"
android:layout_marginTop="@dimen/_23pxv"
android:layout_width="match_parent"
android:singleLine="true"
android:text="@{androidLarge107VM.androidLarge107Model.txtHeating}"
style="@style/txtRobotoCondensedbold25" tools:text="@string/lbl_heating"/>
<TextView android:ellipsize="end"
android:gravity="center_horizontal"
android:id="@+id/txtTeapot"
android:layout_height="wrap_content"
android:
layout_marginTop="@dimen/_20pxv"
android:layout_width="match_parent"
android:singleLine="true"
android:text="@{androidLarge107VM.androidLarge107Model.txtTeapot}"
style="@style/txtRobotoCondensedbold25" tools:text="@string/lbl_teapot"/>
<TextView android:ellipsize="end"
android:gravity="center_horizontal"
android:id="@+id/txtBoiler"
android:layout_height="wrap_content"
android:layout_marginTop="@dimen/_25pxv"
android:layout_width="match_parent"
android:singleLine="true"
android:text="@{androidLarge107VM.androidLarge107Model.txtBoiler}"
style="@style/txtRobotoCondensedbold25" tools:text="@string/lbl_boiler"/>
<TextView android:gravity="center_horizontal"
android:id="@+id/txtFlowerwatering"
android:layout_height="wrap_content"
android:layout_marginTop="@dimen/_14pxv"
android:layout_width="match_parent"
android:text="@{androidLarge107VM.androidLarge107Model.txtFlowerwatering}"

```

```

style="@style/txtRobotoCondensedbold20"
tools:text="@string/lbl_flower_watering"/>
<TextView android:ellipsize="end"
android:gravity="center_horizontal"
android:id="@+id/txtFeeder"
<TextView android:ellipsize="end"
android:gravity="center_horizontal"
android:id="@+id/txtFeeder"
android:layout_height="wrap_content"
android:layout_marginTop="@dimen/_17pxv"
android:layout_width="match_parent"
android:singleLine="true"
android:text="@{(androidLarge107VM.androidLarge107Model.txtFeeder}"
style="@style/txtRobotoCondensedbold25" tools:text="@string/lbl_feeder"/>
<TextView android:ellipsize="end"
android:gravity="center_horizontal"
android:id="@+id/txtSecurity"
android:layout_height="wrap_content"
android:layout_marginTop="@dimen/_25pxv"
android:layout_width="match_parent".
android:singleLine="true"
android:
text="@{(androidLarge107VM.androidLarge107Model.txtSecurity}"
style="@style/txtRobotoCondensedbold20" tools:text="@string/lbl_security"/>
<View android:id="@+id/viewViewFour"
android:layout_height="@dimen/_234pxh"
android:layout_width="@dimen/_64pxh"/>
<TextView android:ellipsize="end"
android:gravity="center_horizontal"
android:id="@+id/txtAboutus"
android:layout_height="wrap_content"
android:layout_width="match_parent"
android:singleLine="true"
android:text="@{androidLarge107VM.androidLarge107Model.txtAboutus}"
style="@style/txtRobotoCondensedbold15"
tools:text="@string/lbl_about_us"/>
</LinearLayout>
<LinearLayout android:id="@+id/linearColumnmenuOne"

```

```

android:layout_gravity="center"
android:layout_height="wrap_content"
android:layout_width="match_parent"
android:orientation="vertical" style="@style/groupStyleblue_gray_100_01">
<TextView android:ellipsize="end"
android:gravity="center_horizontal"
android:id="@+id/txtMenuOne"
android:layout_height="wrap_content"
android:layout_marginTop="@dimen/_4pxv"
android:layout_width="match_parent"
android:singleLine="true"
android:text="@{androidLarge107VM.androidLarge107Model.txtMenuOne}"
style="@style/txtRobotoCondensedbold36" tools:text="@string/lbl_menu"/>
<TextView android:ellipsize="end"
android:gravity="center_horizontal"
android:id="@+id/txtCategoriesOne"
android:layout_height="wrap_content"
android:layout_marginTop="@dimen/_53pxv"
android:layout_width="match_parent"
android:singleLine="true"
android:text="@{androidLarge107VM.androidLarge107Model.txtCategoriesOne}"
style="@style/txtRobotoCondensedbold17"
tools:text="@string/lbl_categories"/>
<TextView android:ellipsize="end"
android:gravity="center_horizontal"
android:id="@+id/txtLampOne"
android:layout_height="wrap_content"
android:
layout_marginTop="@dimen/_61pxv"
android:layout_width="match_parent"
android:singleLine="true"
android:text="@{androidLarge107VM.androidLarge107Model.txtLampOne}"
style="@style/txtRobotoCondensedbold25" tools:text="@string/lbl_lamp"/>
<TextView android:ellipsize="end"
android:gravity="center_horizontal"
android:id="@+id/txtHeatingOne"
<TextView android:ellipsize="end"
android:gravity="center_horizontal"

```

```
android:id="@+id/txtFeeder"
android:layout_height="wrap_content"
android:layout_marginTop="@dimen/_17pxv"
android:layout_width="match_parent"
android:singleLine="true"
android:text="@{androidLarge107VM.androidLarge107Model.txtFeeder}"
style="@style/txtRobotoCondensedbold25" tools:text="@string/lbl_feeder"/>
<TextView android:ellipsize="end"
android:gravity="center_horizontal"
android:id="@+id/txtSecurity"
android:layout_height="wrap_content"
android:layout_marginTop="@dimen/_25pxv"
android:layout_width="match_parent"
android:singleLine="true"
android:text="@{androidLarge107VM.androidLarge107Model.txtSecurity}"
style="@style/txtRobotoCondensedbold20" tools:text="@string/lbl_security"/>
<View android:id="@+id/viewViewFour"
android:layout_height="@dimen/_234pxh"
android:layout_width="@dimen/_64pxh"/>
<TextView android:ellipsize="end"
android:gravity="center_horizontal"
android:id="@+id/txtAboutus"
android:layout_height="wrap_content"
android:layout_width="match_parent"
android:singleLine="true"
android:text="@{androidLarge107VM.androidLarge107Model.txtAboutus}"
style="@style/txtRobotoCondensedbold15" tools:text="@string/lbl_about_us"/>
</LinearLayout>
<LinearLayout android:id="@+id/linearColumnmenuOne"
android:layout_gravity="center"
android:layout_height="wrap_content"
android:layout_width="match_parent"
android:orientation="vertical" style="@style/groupStyleblue_gray_100_01">
<TextView android:ellipsize="end"
android:gravity="center_horizontal"
android:id="@+id/txtMenuOne"
android:layout_height="wrap_content"
android:layout_marginTop="@dimen/_4pxv"
```

```

android:layout_width="match_parent"
android:singleLine="true"
android:text="@{androidLarge107VM.androidLarge107Model.txtMenuOne}"
style="@style/txtRobotoCondensedbold36" tools:text="@string/lbl_menu"/>
<TextView android:ellipsize="end"
android:gravity="center_horizontal"
android:id="@+id/txtCategoriesOne"
android:layout_height="wrap_content"
android:layout_marginTop="@dimen/_53pxv"
android:layout_width="match_parent"
android:singleLine="true"
android:text="@{androidLarge107VM.androidLarge107Model.txtCategoriesOne}"
style="@style/txtRobotoCondensedbold17"
tools:text="@string/lb1_categories"/>
<TextView android:ellipsize="end"
android:gravity="center_horizontal"
android:id="@+id/txtLampOne"
android:layout_height="wrap_content"
android:layout_marginTop="@dimen/_61pxv"
android:layout_width="match_parent"
android:singleLine="true"
android:text="@{androidLarge107VM.androidLarge107Model.txtLampOne}"
style="@style/txtRobotoCondensedbold25" tools:text="@string/lbl_lamp"/>
<TextView android:ellipsize="end"
android:gravity="center_horizontal"
android:id="@+id/txtHeatingOne"
android:id="@+id/txtHeatingOne"
android:layout_height="wrap_content"
android:layout_marginTop="@dimen/_23pxv"
android:layout_width="match_parent"
android:singleLine="true"
android:text="@{androidLarge107VM.androidLarge107Model.txtHeatingOne}"
style="@style/txtRobotoCondensedbold25" tools:text="@string/lb1_heating"/>
<TextView android:ellipsize="end"
android:gravity="center_horizontal"
android:id="@+id/txtTeapotOne"
android:layout_height="wrap_content"
android:layout_marginTop="@dimen/_20pxv"

```

```

android:layout_width="match_parent"
android:singleLine="true"
android:text="@{androidLarge107VM.androidLarge107Model.txtTeapotOne)"
style="@style/txtRobotoCondensedbold25"
tools:text="@string/lbl_teapot"/>
<TextView android:ellipsize="end"
android: gravity="center_horizontal"
android:id="@-id/txtBoilerüne"
android: layout_height="wrap_content"
android:layout_marginTop="@dimen/_25pxv"
android: layout_width="match_parent"
android:singleLine="true"
android: text="@{androidLarge107VM.androidLarge107Model.txtBoilerOne)"
style="@style/txtRobotoCondensedbold25" tools:text="@string/lbl_boiler"/>
<TextView android:gravity="center_horizontal"
android:id="@+id/txtFlowerwatering1"
android: layout_height="wrap_content"
android:layout_marginTop="@dimen/_14pxv"
android: layout_width="match_parent"
android:text="@{androidLarge107VM.androidLarge107Model.txtFlowerwatering1)"
style="@style/txtRobotoCondensedbold20"
tools:text="@string/lbl_flower_watering"/>
<TextView
android:ellipsize="end"
android: gravity="center_horizontal"
android:id="@+id/txtFeederOne"
android: layout_height="wrap_content"
android:layout_marginTop="@dimen/_17pxv"
android:
layout_width="match_parent"
android:singleLine="true"
android: text="@{androidLarge107VM.androidLarge107Model.txtFeederOne)"
style="@style/txtRobotoCondensedbold25" tools:text="@string/lbl_feeder"/>
<View android:id="@+id/viewViewFive"
android:layout_height="@dimen/_267pxh"
android:layout_width="@dimen/_108pxh"/>
<LinearLayout android:id="@+id/linearColumnaboutus"
android:layout_height="wrap_content"

```

```

android:layout_width="match_parent"
android:orientation="vertical"
android:paddingBottom="@dimen/_16pxv"
android:paddingEnd="@dimen/_21pxh"
android:paddingStart="@dimen/_21pxh"
android:paddingTop="@dimen/_16pxv" style="@style/groupStylecyan_100">
<TextView android:ellipsize="end"
android:gravity="center_horizontal"
android:id="@+id/txtAboutusOne"
android:layout_height="match_parent"
android:layout_width="match_parent"
android:singleLine="true"
android:text="@{androidLarge107VM.androidLarge107Model.txtAboutusOne}"
style="@style/txtRobotoCondensedbold15" tools:text="@string/lbl_about_us"/>
</LinearLayout>
</LinearLayout>
</FrameLayout>
</FrameLayout>
</layout>
<?xml version="1.0" encoding="UTF-8"?>
<layout xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:app="http://schemas.android.com/apk/res-auto" xmlns:tools="
http://schemas.android.com/tools">
<data>
<variable name="androidLarge108VM"
type="com.antonsapplication.app.modules.androidlarge108.data.viewmodel.AndroidL
arge108VM"/>
</data>
<FrameLayout
android:id="@+id/frameAndroidLarge"
android:layout_height="match_parent"
android:layout_width="match_parent" style="@style/groupStylewhite_A700">
<LinearLayout android:id="@+id/linearColumnview"
android:layout_gravity="bottom|end"
android:layout_height="wrap_content"
android:layout_marginBottom="@dimen/_154pxv"
android:layout_marginEnd="@dimen/_42pxh"
android:layout_marginStart="@dimen/_220pxh"

```

```

android:layout_width="match_parent" android:orientation="vertical">
<LinearLayout android: gravity="end"
android:id="@rid/linearRowview"
android: layout_height="wrap_content"
android:layout_width="match_parent" android:orientation="horizontal">
<View
android:background="@drawable/rectangle_bg_green_400_01_radius_5"
android:id="@+id/viewview"
android:layout_gravity="bottom"
android: layout_height="@dimen/_10pxh"
android: layout_marginBottom="@dimen/_7pxv"
android:layout_marginTop="@dimen/_9pxv"
android:layout_width="@dimen/_10pxh"/>
<com.google.android.material.switchmaterial.SwitchMaterial
android:id="@+id/switchMaterialSwitch"
android: layout_gravity="center_vertical"
android:layout_height="wrap_content"
android: layout_marginStart="@dimen/_6pxh" android:
layout_width="wrap_content"/>
</LinearLayout>
<View
android:background="@drawable/rectangle_bg_blue_gray_100_01_radius_16"
android:id="@+id/viewViewOne"
android:layout_height="@dimen/_35pxh"
android:
layout_marginEnd="@dimen/_20pxh"
android:layout_marginTop="@dimen/_115pxv"
android:layout_width="@dimen/_35pxh"/>
<View
android:background="@drawable/rectangle_bg_green_400_01_radius_16"
android:id="@+id/viewViewTwo"
android:layout_height="@dimen/_35pxh"
android:layout_marginEnd="@dimen/_20pxh"
android: layout_marginTop="@dimen/_22pxv" android:
layout_width="@dimen/_35pxh"/>
<View
android:background="@drawable/rectangle_bg_blue_gray_100_01_radius_16"
android:id="@+id/viewViewThree"

```

```

android:layout_height="@dimen/_35pxh"
android:layout_marginEnd="@dimen/_20pxh"
android:layout_marginTop="@dimen/_22pxv"
android:layout_width="@dimen/_35pxh"/>
</LinearLayout>
<LinearLayout android:id="@+id/linearColumnturnonoff"
android:layout_gravity="bottom|start"
android:layout_height="wrap_content"
android:layout_marginBottom="@dimen/_155pxv"
android:layout_marginEnd="@dimen/_196pxh"
android:layout_marginStart="@dimen/_21pxh"
android:layout_width="match_parent" android:orientation="vertical">
<TextView android:ellipsize="end"
android:id="@+id/txtTurnonoff"
android:layout_height="wrap_content"
android:layout_width="match_parent"
style=#style/txtkodotolondenseddolaso tools:text=estring/10i_menu />
<ImageView android:id="S+id/imageSettingsFive"
android:layout_gravity="top|start"
android:layout_height="@dimen/_21pxh"
android:layout_marginTop="@dimen/_10pxv"
android:layout_width="@dimen/_38pxh"
android:scaleType="fitXY"
android:src="@drawable/img_settings_gray_800"
tools:ignore="ContentDescription" tools:src="@drawable/ing_settings_gray_800"/>
</FrameLayout>
<TextView android:ellipsize="end"
android:id="@+id/txtFeeding"
android:layout_gravity="center_vertical"
android:layout_height="wrap_content"
android:layout_marginBottom="@dimen/_14pxv"
android:layout_marginStart="@dimen/_46pxh"
android:layout_marginTop="@dimen/_13pxv"
android:layout_width="wrap_content"
android:singleLine="true"
android:text="@{androidLarge108VM.androidLarge108Model.txtFeeding}"
style="@style/txtRobotoCondensedbold20" tools:text="@string/lbl_feeding"/>
<ImageView android:id="@+id/imageImage"

```

```

android:layout_gravity="center_vertical"
android:layout_height="@dimen/_3epxh"
android:layout_marginBottom="@dimen/_11pxv"
android:layout_marginStart="@dimen/_63pxh"
android:layout_marginTop="@dimen/_11pxv"
android:layout_width="@dimen/_30pxh"
android:scaleType="fitxy"
android:src="@drawable/ing_9055356_1"
tools:ignore="ContentDescription" tools:src="@drawable/img_9055356_1"/>
<View android:background="@color/white_A700"
android:id="8+id/lineLine"
android:layout_gravity="center_vertical"
android:layout_height="@dimen/_52pxv"
android:layout_marginStart="@dimen/_6pxh"
android:layout_width="@dimen/_1pxh"/>
<ImageView android:id="@+id/imageMooniconOne"
android:layout_gravity="center_vertical"
android:layout_height="@dimen/_30pxh"
android:layout_marginBottom="@dimen/_11pxv"
android:layout_marginStart="@dimen/_6pxh"
android:layout_marginTop="@dimen/_11pxv"
android:layout_width="@dimen/_30pxh"
android:scaleType="fitxy"
android:src="@drawable/img_moon_icon_512x512_fm9crgpt"
tools:ignore="ContentDescription"
tools:src="@drawable/img_moon_icon_512x512_fm9crgpt"/>
</LinearLayout>
</androidx.appcompat.widget.Toolbar>
<ImageView
android:id="@+id/image2ehb01ttecfoo"
android:layout_gravity="center_horizontal"
android:layout_height="@dimen/_279pxh"
android:layout_marginTop="@dimen/_7pxv"
android:layout_width="@dimen/_267pxh"
android:scaleType="fitXY"
imageRes="@drawable/img_2ehb01_ttec_foo"
tools:ignore="ContentDescription" tools:src="@drawable/img_2ehb01_ttec_foo"/>
</LinearLayout>

```

```

<FrameLayout android:id="@+id/frameStackcategories"
    android:layout_gravity="start"
    android:layout_height="@dimen/_800pxv"
    android:layout_width="@dimen/_108pxh">
    <LinearLayout
        android:id="@+id/linearColumncategorie"
        android:layout_gravity="bottom|center"
        android:layout_height="wrap_content"
        android:layout_marginBottom="@dimen/_17pxv"
        android:layout_marginEnd="@dimen/_7pxh"
        android:layout_marginStart="@dimen/_7pxh"
        android:layout_marginTop="@dimen/_100pxv"
        android:layout_width="match_parent" android:orientation="vertical">
        <TextView android:ellipsize="end"
            android:gravity="center_horizontal"
            android:id="@+id/txtCategories"
            android:layout_height="wrap_content"
            android:layout_width="match_parent"
            android:singleLine="true"
            android:text="@{androidLarge108VM.androidLarge108Model.txtCategories}"
            style="@style/txtRobotoCondensedbold17" tools:text="@string/lbl_categories"/>
        <TextView android:ellipsize="end"
            android:gravity="center_horizontal"
            android:id="@+id/txtLamp"
            android:layout_height="wrap_content"
            android:layout_marginTop="@dimen/_61pxv"
            android:layout_width="match_parent"
            android:singleLine="true"
            android:text="@{androidLarge108VM.androidLarge108Model.txtLamp}"
            style="@style/txtRobotoCondensedbold25" tools:text="@string/lbl_lamp"/>
        <TextView android:ellipsize="end"
            android:gravity="center_horizontal"
            android:id="@+id/txtHeating"
            android:layout_height="wrap_content"
            android:layout_marginTop="@dimen/_23pxv"
            android:layout_width="match_parent"
            android:singleLine="true"
            android:text="@{androidLarge108VM.androidLarge108Model.txtHeating}"

```

```

style="@style/txtRobotoCondensedbold25"
tools:text="@string/lbl_heating"/>
<TextView android:ellipsize="end"
android:gravity="center_horizontal"
android:id="@+id/txtTeapot"
android:layout_height="wrap_content"
android:layout_marginTop="@dimen/_20pxv"
android:layout_width="match_parent"
android:singleLine="true"
android:text="@{androidLarge108VM.androidLarge108Model.txtTeapot}"
style="@style/txtRobotoCondensedbold25" tools:text="@string/lbl_teapot"/>
<TextView android:ellipsize="end"
android:gravity="center_horizontal"
android:id="@+id/txtBoiler"
android:layout_height="wrap_content"
android:layout_marginTop="@dimen/_25pxv"
android:layout_width="match_parent"
android:singleLine="true"
android:text="@{androidLarge108VM.androidLarge108Model.txtBoiler}"
style="@style/txtRobotoCondensedbold25"
tools:text="@string/lbl_boiler"/>
<TextView android:gravity="center_horizontal"
android:id="@+id/txtFlowerwatering"
android:layout_height="wrap_content"
android:layout_marginTop="@dimen/_14pxv"
android:layout_width="match_parent"
android:text="@{androidLarge108VM.androidLarge108Model.txtFlowerwatering}"
style="@style/txtRobotoCondensedbold20" tools:text="@string/lbl_flower_watering"/>
<TextView android:ellipsize="end"
android:gravity="center_horizontal"
android:id="@+id/txtFeeder"
android:layout_height="wrap_content"
android:layout_marginTop="@dimen/_17pxv"
android:layout_width="match_parent"
android:singleLine="true"
android:text="@{androidLarge108VM.androidLarge108Model.txtFeeder}"
<?xml version="1.0"
encoding="utf-8"

```

```
standalone="no"?>
<resources>
<color name="purple_200">#FFBB86FC</color>
<color name="purple_500">#FF6200EE</color>
<color name="purple_700">#FF3700B3</color>
<color name="teal_200">#FF03DAC5</color>
<color name="teal_700">#FF018786</color>
<color name="black">#FF000000</color>
<color name="white">#FFFFFFF</color>
<color name="blue_gray_100">#c8e2eb</color>
<color name="blue_gray_100_01">#d9d9d9</color>
<color name="gray_800">#505050</color>
<color name="green_400">#67bb5a</color>
<color name="yellow_400">#f8eb4e</color>
<color name="indigo_300">#6493c1</color>
<color name="cyan_100">#b0e8f4</color>
<color name="yellow_500">#f7e941</color>
<color name="black_900_cc">#cc000000</color>
<color name="yellow_800">#d79b2f</color>
<color name="black 900">#000000</color>
<color name="blue 100">#c3e2ec</color>
<color name="green_400_01">#68bb5a</color>
<color name="white_A700">#ffffff</color>
</resources>
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<resources>
<string name="app_name">anton_s_application1</string>
<string name="loading">Loading</string>
<string name="no_internet_connection">No internet connection!</string>
<string name="lbl_day_white">DAY WHITE</string>
<string name="lbl_enter_name">ENTER NAME</string>
<string name="lbl_20">20%</string>
<string name="lbl_e">0%</string>
<string name="lbl_your_devices">YOUR DEVICES</string>
<string name="lbl_menu">MENU</string>
<string name="lbl_cold_white">COLD WHITE</string>
<string name="msg_temperature_25_50_75_100">TEMPERATURE
\u202825
```

50

75

100</string>

<string name="lbl_categories">CATEGORIES</string>

<string name="lbl_humidity">HUMIDITY</string>

<string name="lbl_security">SECURITY</string>

<string

name="lbl_warm_white">WARM WHITE</string>

<string name="lbl_lamp">LAMP</string>

<string name="lbl_heating">HEATING</string>

<string name="lbl_add_to_category">ADD TO CATEGORY</string>

<string name="lbl_about_us">ABOUT Us</string>

<string name="msg_feed_20_40_80">FEED

\u202820

80</string>

<string name="lbl_feeder">FEEDER</string>

<string name="lbl_flower_watering">FLOWER WATERING</string>

<string name="lbl_heating2">HEATING </string>

<string name="lbl_feeding">FEEDING</string>

<string name="lbl_teapot">TEAPOT</string>

<string name="msg_we_recommend_to">WE RECOMMEND TO SUPPORT HUMIDITY ON 30% AND NOT ADD MORE. YOUR FLOWER CAN DIE.</string>

<string name="lbl_boiler">BOILER</string>

<string name="lbl_turn_on_off">TURN

ON/OFF</string>

<string

name="lbl_30">30%</string>

<string name="lbl_ok">OK</string>

<string name="lbl_add_humidity">ADD

HUMIDITY</string>

<string name="msg_you_don_t_have_added">YOU DON'T HAVE ADDED ANY SMART DEVICES.\u2028\u2028CLICK HERE TO ADD</string>

<string

name="lbl_10">10%</string>

<string name="msg_name_of_the_new">NAME OF THE NEW ADDED DEVICE.</string>

<string name="msg_recommended_colors">RECOMMENDED COLORS FOR YOUR LAMP</string>

```

</resources>
<?xml version="1.0" encoding="UTF-8"?>
<resources xmlns:tools=" http://schemas.android.com/tools ">
<style name="etSolidRounded">
<item name="android:textSize">@dimen/_22pxh</item>
<item name="android:textColor">@color/black_900_cc</item>
<item name="android:fontFamily">@font/roboto_condensed</item>
<item name="android:textFontWeight" tools:targetApi="p">700</item>
<item
name="android:background">@drawable/rectangle_bg_white_a700_radius_18</item>
</style>
<style name="groupStylegray_800">
<item name="android:background">@drawable/rectangle_bg_gray_800</item>
</style>
<style name="groupStyleblue_gray_100_01">
<item
name="android:background">@drawable/rectangle_bg_blue_gray_100_01</item>
</style>
<style name="groupStylewhite_A700cornerRadius">
<item
name="android:background">@drawable/rectangle_bg_white_a700_radius_18</item>
</style>
<style name="txtRobotoCondensedbold22_1">
<item name="android:textSize">@dimen/_22pxh</item>
<item name="android:textColor">@color/black_900_cc</item>
<item name="android:FontFamily">@font/roboto_condensed</item>
<item name="android:textFontWeight" tools:targetApi="p">700</item>
</style>
<style name="txtRobotoCondensedbold25_1">
<item name="android:textSize">@dimen/_25pxh</item>
<item name="android:textColor">@color/white_A700</item>
<item name="android:fontFamily">@font/roboto_condensed</item>
<item name="android:textFontWeight" tools:targetApi="p">700</item>
</style>
<style name="txtRobotoCondensedbold15">
<item name="android:textSize">@dimen/_15pxh</item>
<item name="android:textColor">@color/black_900</item>
<item name="android:fontFamily">@font/roboto_condensed</item>

```

```

<item name="android:textFontWeight" tools:targetApi="p">700</item>
</style>
<style name="txtRobotoCondensedbold25">
<item name="android:textSize">@dimen/_25pxh</item>
<item name="android:textColor">@color/black_900</item>
<item name="android:fontFamily">@font/roboto_condensed</item>
<item name="android:textFontWeight" tools:targetApi="p">700</item>
</style>
<style name="txtRobotoCondensedbold36">
<item name="android:textSize">@dimen/_36pxh</item>
<item name="android:textColor">@color/black_900</item>
<item name="android:fontFamily">@font/roboto_condensed</item>
<item name="android:textFontWeight" tools:targetApi="p">700</item>
</style>
<style name="txtRobotoCondensedbold17">
<item name="android:textSize">@dimen/_17pxh</item>
<item name="android:textColor">@color/black_900</item>
<item name="android:fontFamily">@font/roboto_condensed</item>
<item name="android:textFontWeight" tools:targetApi="p">700</item>
</style>
<style name="txtRobotoCondensedbold20">
<item name="android:textSize">@dimen/_20pxh</item>
<item name="android:textColor">@color/black_900</item>
<item name="android:fontFamily">@font/roboto_condensed</item>
<item name="android:textFontWeight" tools:targetApi="p">700</item>
</style>
<style name="groupStylecyan_100">
<item name="android:background">@drawable/rectangle_bg_cyan_100</item>
</style>
<style name="txtRobotoCondensedbold22">
<?xml
version="1.0" encoding="UTF-8">>
<manifest
package="com.antonsapplication.app"
http://schemas.android.com/apk/res/android
">
<uses-permission android:name="android.permission.INTERNET"/>

```

xmlns:android="

```

<uses-permission
android:name="android.permission.ACCESS_NETWORK_STATE"/>
<application android:allowBackup="false"
android:icon="@mipmap/ic_launcher"
android:label="@string/app_name" android:largeHeap="true"
android:name="com.antonsapplication.app.appcomponents.di.MyApp"
android:roundIcon="@mipmap/ic_launcher_round"
android:supportRtl=
android:theme="@style/Theme.MyApp" android:usesCleartextTraffic
<activity
android:name="com.antonsapplication.app.modules.androidlargeeight.ui.AndroidLarge
EightActivity"
<activity
weight.ui.AndroidLargeSixtyeightActivity" android:screenOrientation="port
<activity
android:name="com.antonsappli
roidlargeFiftyfiveActivity" android:screenOrientation="port
<activity android:exported="
android:name="com. antonsapplicat
roidlary
android:screenOrientation="portrait">
<intent-filter>
<action android:name
<category android:nam
</intent-filter>
</activity>
<activity
android:name="com.antonsapplication.app.modules.androidlargethirteen.ui
AndroidL.argeThirteenActivity"
android:screenOrientation="portrait"/1
package com.anton s application.app
import org.junit.Test
import org. junit.Assert.*
*
Example local unit test, which will execute on the development machine (host).
*
/

```

See [testing documentation](<http://d.android.com/tools/testing>).

```

class Example Unit Test {
@Test
fun addition_incorrect() {
assertEquals(4, 2 + 2)
package com.anton.s.application.app
import com.anton.s.application.app.components.base.BaseActivity
import com.anton.s.application.app.data.binding.Layout ProgressDialog Binding
class MainActivity: BaseActivity<Layout ProgressDialog Binding>(R.layout.layout_progress_dialog) {
override fun uninitialized() {
override fun setUp Clicks() {
<?xml version="1.0" encoding="UTF-8"?>
<shape android:shape="rectangle" xmlns:android="
http://schemas.android.com/apk/res/android ">
<solid android:color="@color/blue_gray_100_01"/>
<corners android:radius="@dimen/_5pxh"/>
</shape>
<?xml version="1.0" encoding="UTF-8"?>
<shape android:shape="rectangle" xmlns:android="
http://schemas.android.com/apk/res/android ">
<solid android:color="@color/cyan_100"/>
</shape>
<?xml version="1.0" encoding="UTF-8"?>
<shape android: shape="rectangle" xmlns: android="
http://schemas.android.com/apk/res/android ">
<solid android:color="@color/gray_800"/>
</shape>
<?xml version="1.0" encoding="UTF-8"?>
<shape android:shape="rectangle" xmlns:android="
http://schemas.android.com/apk/res/android ">
<solid android:color="@color/green_400_01"/>
<corners android:radius="@dimen/_16pxh"/>
</shape>
<?xml version="1.0" encoding="UTF-8"?>
<shape android:shape="rectangle" xmlns:android="
http://schemas.android.com/apk/res/android ">
<solid android:color="@color/green_400_01"/>
<corners android:radius="@dimen/_5pxh"/>

```

</shape>

ДОДАТОК Б ДЕМОНСТРАЦІЙНІ МАТЕРІАЛИ

ДЕРЖАВНИЙ УНІВЕРСИТЕТ ТЕЛЕКОМУНІКАЦІЙ
НАВНАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
КАФЕДРА КОМП'ЮТЕРНІ НАУКИ

Виконав: студент 4 курсу, групи КНДІ
Ельбезрі А.Н.
Керівник: к.т.н., доцент каф.
Іщераков С.М.

ВСТУП

- В сучасному світі, завдяки стрімкому розвитку технологій, концепція "розумного дому" стає все більш актуальною та важливою. Розумні системи управління та контролю, що вбудовані у житлові приміщення, не лише забезпечують комфорт та безпеку мешканців, але й відкривають нові можливості для енергоефективного використання ресурсів та оптимізації побутових процесів.
- Розробка засобів інтелектуального управління та контролю системи "розумний дім" є відповіддю на зростаючі потреби сучасного суспільства в зручному та безпечному житті. Ці системи використовують передові технології в області інформаційних та комунікаційних технологій, датчиків, штучного інтелекту та Інтернету речей для автоматизації різних аспектів побутового життя. Завдяки системам "розумний дім", мешканці можуть контролювати освітлення, температуру, безпеку, аудіо- та відеосистеми в будинку за допомогою смартфонів або голосових помічників.

АКТУАЛЬНІСТЬ РОБОТИ

- **Актуальність** з огляду на зростаючу важливість енергозбереження, “розумний дім” набуває особливої значущості, оскільки він дозволяє оптимізувати споживання ресурсів. Це не тільки сприяє зниженню витрат для кінцевого споживача, але й зменшує екологічний вплив, що є критично важливим у контексті глобальних кліматичних змін.
- **Об'єкт дослідження** система “розумний дім”, її компоненти та механізми управління.
- **Предмет дослідження** процеси та методи, які використовуються для створення інтелектуальної системи управління “розумний дім”.
- **Мета роботи:** створити ефективну та надійну систему, яка б забезпечувала автоматизацію домашніх процесів, підвищення комфорту проживання, енергоефективності та безпеки житлових приміщень.

ОПИС ПРЕДМЕТНОЇ ОБЛАСТІ ЗАВДАННЯ

- Розумний дім — це сучасна технологічна система, що працює на основі штучного інтелекту (ШІ), призначена для автоматизації та полегшення повсякденних завдань без втручання людини. В основі цієї системи лежить операційна система multi-room, яка об'єднує всі електроприлади у домашню мережу з доступом до Інтернету, дозволяючи керувати ними за допомогою пульта, комп'ютера або смартфона.
- Основні функції розумного дому:
- Розумне освітлення:
- Мікроклімат
- Система автоматично регулює температуру та вологість повітря в кожній кімнаті.
- Енергозбереження
- Температури будинку регулюється залежно від зовнішніх умов.

ОГЛЯД ТА ПОРІВНЯЛЬНА ХАРАКТЕРИСТИКА ПОДІБНИХ ПРОГРАМ

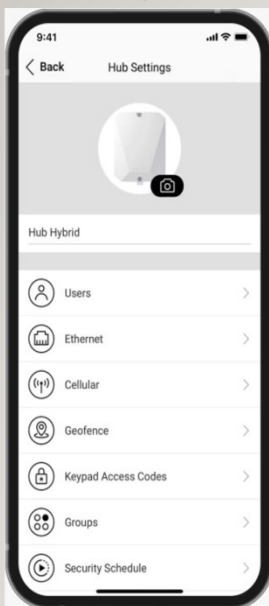
Ajax Systems

Ajax Systems - це українська компанія, що спеціалізується на розробці та виробництві систем безпеки для розумного дому та бізнесу. Заснована в 2011 році, компанія швидко стала лідером на ринку бездротових систем безпеки, завоювавши довіру користувачів у більш ніж 100 країнах світу.

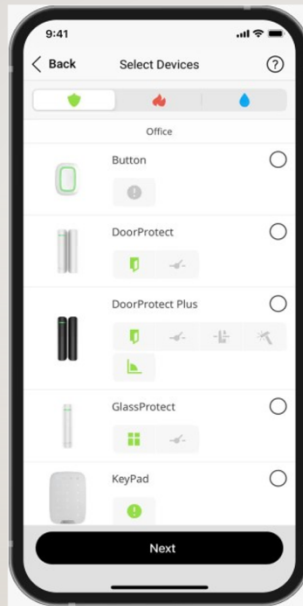
Основні характеристики Ajax Systems

- Бездротові технології: Всі пристрої Ajax працюють на бездротових технологіях, що дозволяє швидко та легко встановлювати системи без потреби проводів.
- Сучасний дизайн: Продукція Ajax відрізняється стильним та сучасним дизайном, що дозволяє легко інтегрувати її у будь-який інтер'єр.
- Мобільний додаток: Ajax має зручний мобільний додаток для керування системою з будь-якого місця за допомогою смартфона.
- Широкий вибір пристроїв: Компанія пропонує різноманітні пристрої для різних потреб, включаючи датчики руху, дверей/вікон, вогню, протікання води, камери відеоспостереження та багато іншого.

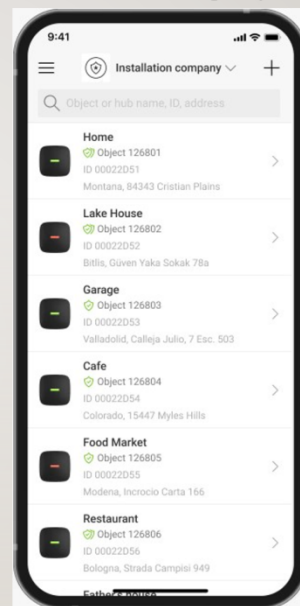
Налаштування



вибір пристрою користувачем



список всіх об'єктів користувача



ECOISME

- Ecoisme - це компанія, що спеціалізується на розробці інноваційних рішень для енергоефективності та моніторингу енергоспоживання в домогосподарствах. Основною метою Ecoisme є надання користувачам інструментів для ефективного управління електроенергією та зменшення витрат на комунальні послуги.

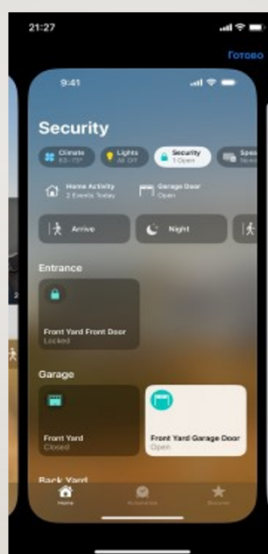
Основні характеристики Ecoisme:

- Бездротові технології: Всі пристрої працюють на бездротових технологіях, що дозволяє легко встановлювати системи без потреби проводів.
- Дизайн: Продукція відрізняється стильним та гарним дизайном, що дозволяє легко інтегрувати її у будь-який інтер'єр.
- Надійність і безпека: Системи Ecoisme відомі своєю високою надійністю та безпекою. Вони використовують передові технології шифрування та мають вбудовані заходи захисту від вторгнень.
- Мобільний додаток: зручний мобільний додаток для керування системою з будь-якого місця за допомогою смартфона.
- Широкий вибір пристроїв: Компанія пропонує різноманітні пристрої для різних потреб, включаючи датчики руху, дверей/вікон, вогню, протікання води, камери відеоспостереження та багато іншого.

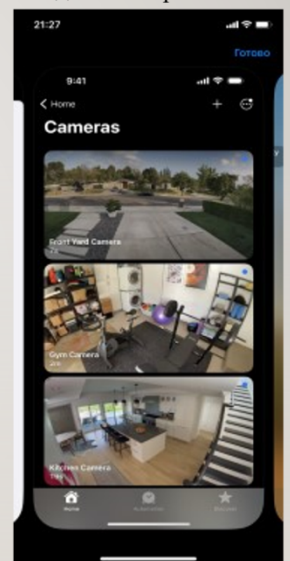
Сторінка об'єкту



Сторінка безпеки



Список активних камер відеоспостереження



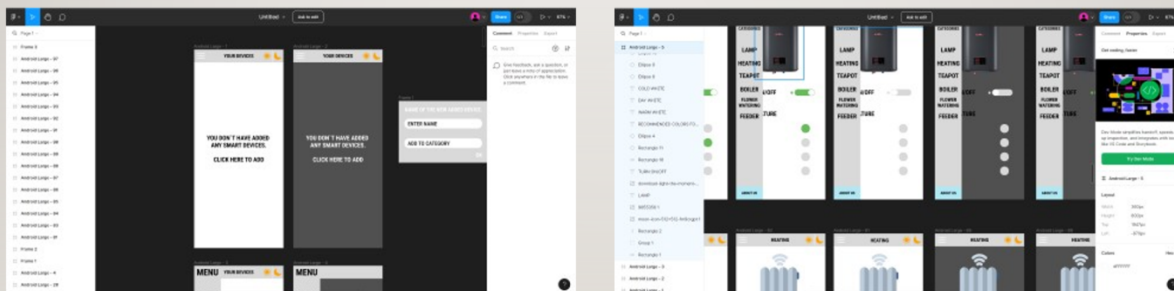
ПРОЕКТУВАННЯ СИСТЕМИ

- Планування — це процес визначення та організації всіх необхідних кроків, ресурсів і завдань для створення, впровадження та підтримки програмного продукту або системи. Цей процес включає визначення цілей, строків, бюджетів і ризиків, а також координацію роботи команди розробників, дизайнерів, тестувальників та інших учасників проекту.
- Мета планування — забезпечити ефективну та своєчасну розробку системи, яка відповідає вимогам користувачів і замовників.
- Без ретельного планування розробка системи може зіткнутися з численними проблемами, затримки, недосягнення очікувань користувачів та низьку якість кінцевого продукту. Тому детальне планування є основою успішної реалізації будь-якого проекту розробки системи.

ОСНОВНІ ЕТАПИ ПЛАНУВАННЯ РОЗРОБКИ СИСТЕМИ:

- | | |
|------------------------------------|---------------------------------|
| •Визначення мети та обсягу проекту | •Архітектура системи: |
| •Мета проекту | •Дизайн бази даних: |
| •Обсяг проекту | •Дизайн інтерфейсу користувача: |
| •Збір і аналіз вимог | •Розробка плану проекту: |
| •Функціональні вимоги | •Розклад завдань: |
| •Нефункціональні вимоги | •Визначення критичних точок: |
| •Вимоги користувачів | •Розподіл завдань: |
| •Створення технічного завдання | •Впровадження: |
| •Проектування системи | •План впровадження: |

ДИЗАЙН НАШОГО ЗАСТОСУНКУ, ЯКИЙ БУВ РОЗРОБЛЕНИЙ ЗА ДОПОМОГОЮ ТЕХНОЛОГІЙ FIGMA, ТА РЕАЛІЗОВАНИЙ У ANDROID STUDIO



ВИСНОВКИ

- В роботі проведено аналіз найпопулярніших систем розумного дому та визначено їх основні переваги та недоліки, які були ретельно проаналізовані для покращення планування програмного продукту.
- Вивчено архітектурне рішення SSOT, яка передбачає існування єдиного джерела, де зберігаються та з якого отримуються всі дані організації або системи. Основна мета SSOT полягає в забезпеченні точності, узгодженості та актуальності даних, щоб уникнути дублювання та ризиків в різних джерелах даних.
- Розроблений програмний продукт показав, що система "розумний дім" дуже актуальна, та її попит в майбутньому буде лише збільшуватися.
- Були визначені потреби користувачів і технічних специфікацій, що дозволило створити програмний продукт, який розроблений на базі сучасних технологій, які регулярно підтримуються розробниками, що робить систему більш стабільною, також розроблений продукт максимально відповідає очікуванням ринку.
- В цілому, робота показала перспективність, конкурентоспроможність та актуальність розробленої системи. Для впровадження та популяризації програмного продукту його необхідно оновлювати, розширювати функціональність програми та підтримувати зворотній зв'язок з користувачами.