Smart home

Project Team

Tran Ly Dang Khoa	16685
Nguyen Thai Minh	15643
Tran Dang Tu	16664
Truong Nguyen Thien An	16772
Vo Khanh Minh	16035
Nguyen Duy Phu Quang	15890

Session 2019-2023

Supervised by

Dr. Le Trong Nhan

Department of Computer Science

Vietnamese German University
Binh Duong, Vietnam

Dec, 2022

Student's Declaration

We declare that this project titled "*Smart home*", submitted as requirement for the award of degree of Bachelors in Computer Science, does not contain any material previously submitted for a degree in any university; and that to the best of our knowledge, it does not contain any materials previously published or written by another person except where due reference is made in the text.

We understand that the management of Department of Computer Science, National University of Computer and Emerging Sciences, has a zero tolerance policy towards plagiarism. Therefore, We, as authors of the above-mentioned thesis, solemnly declare that no portion of our thesis has been plagiarized and any material used in the thesis from other sources is properly referenced.

We further understand that if we are found guilty of any form of plagiarism in the thesis work even after graduation, the University reserves the right to revoke our BS degree.

Tran Ly Dang Khoa	Signature: ________________
Nguyen Thai Minh	Signature: ________________
Tran Dang Tu	Signature: ________________
Truong Nguyen Thien An	Signature: ________________
Vo Khanh Minh	Signature: ________________
Nguyen Duy Phu Quang	Signature: ________________

____

Verified by Plagiarism Cell Officer
Dated:

Certificate of Approval

The Department of Computer Science, National University of Computer and Emerging Sciences, accepts this thesis titled *Smart home*, submitted by Tran Ly Dang Khoa (16685), Nguyen Thai Minh (15643), Tran Dang Tu (16664), Truong Nguyen Thien An (16772), Vo Khanh Minh (16035), and Nguyen Duy Phu Quang (15890), in its current form, and it is satisfying the dissertation requirements for the award of Bachelors Degree in Computer Science.

Supervisor

Dr. Le Trong Nhan   Signature: ________________

____

Coordinator of FYP

FYP Coordinator
Vietnamese German University, Binh Duong

____

Dr. Head of Computer Science Department

HoD of Department of Computer Science
National University of Computer and Emerging Sciences

Acknowledgements

Without the support of our supervisor, this report and research could not be completed. We would like to thank Dr. Nhan for his consistent assistance and knowledgeable advice. As our supervisor, he provided us with useful information on the topic. His enthusiasm, knowledge, and exacting attention to detail have been an inspiration and kept our work on track from our first encounter with AIot by Python code until the final of the demo and this report.

Tran Ly Dang Khoa

Nguyen Thai Minh

Tran Dang Tu

Truong Nguyen Thien An

Vo Khanh Minh

Nguyen Duy Phu Quang

Abstract

AIoT (Artificial Intelligence of Things) is a relatively new term that has recently become a hot topic that combines two of the hottest acronyms, AI (Artificial Intelligence) and IoT (Internet of Things). IoT consists of interconnected things with built-in sensors and has the potential to generate and collect a vast amount of data. IoT is an emerging technology that is making our world smarter. The idea of a connected world cannot be imagined without IoT [3]. An IoT-based Smart Home is one such example. In the IoT-enabled Smart Home environment various things such as lighting, home appliances, pumps, security cameras, etc. All are connected to the Internet and allowing users to monitor and control things regardless of time and location. Individual IoT systems can be integrated into a large-scale system for various modern applications. With that comes a lot of collected or real-time data, an intelligent and efficient data processing is essential to make effective use of the information generated from these data. The data can be analyzed and utilized with AI for problem-solving or decision-making. Without AI, IoT would have limited value. AI can multiply the value of IoT; conversely, IoT can promote the learning and intelligence of AI. However, there are many challenges while deploying AIoT in practice. For instance, machine learning is one of the key technologies to be utilized in AIoT systems. Besides, there are many other issues such as complexity, efficiency, scalability, accuracy, and robustness related to the increasingly modern AIoT systems and applications. In this project, we implement a simple Smarthome system with a mobile app, Adafruit, and Python.

Keywords: Python, AIot, Machine Learning, Smart home.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

A smart home also referred to as a connected home or eHome is an environment for living that has highly advanced automatic systems. A smart home appears "intelligent" because its daily activities are monitored by a computer. A smart home consists of many technologies via home networking for improving the quality of living. A smart home is a place that has highly advanced automatic systems for controlling and monitoring lighting and temperature, home appliances, multi-media equipment, security systems, and many other functions[1]. IoT plays an important role in building a smart home. Through IoT, almost every object of our daily life in a home can be connected to the Internet. IoT allows monitoring and controlling all of these connected objects regardless of time and location. In this smart home Python project, our goal is to automate certain tasks in our home using internet-connected devices. We will be using a variety of smart home devices, including smart pump, security cameras, and lighting control systems, all of which can be controlled and monitored using Python, a mobile app, and an Adafruit dashboard. In this report, we will present the results of our smart home Python project, which aimed to automate certain tasks in our home using internet-connected devices. To evaluate the success of our project, we will consider a number of criteria, including:

1. Completeness: Did our project achieve all of the goals we set out to accomplish?

2. Accuracy: Did our project accurately control and monitor the smart home devices as intended?

3. User-friendliness: Was our project easy for users to understand and use?

By considering these criteria, we will be able to assess the overall effectiveness and practicality of our smart home Python project. Overall, our aim is to provide a thorough and unbiased evaluation of the project's performance, highlighting both its strengths and areas for improvement

1.1 Concept of Smart Home

In a paper on smart home technology, the concept of a smart home would generally refer to a residence that uses internet-connected devices to enable the remote monitoring and control of appliances and systems, such as lighting, heating, security, and entertainment. These devices are often referred to as "smart home devices" or "home automation devices." Smart home technology has the potential to significantly improve the quality of life for homeowners, as it allows for greater convenience, comfort, and energy efficiency. For example, a smart security camera can alert homeowners to any unusual activity in their homes. However, the adoption of smart home technology has not been without its challenges. One concern is the potential for data privacy breaches, as smart home devices often collect and transmit large amounts of personal information. Additionally, the integration of different smart home devices from different manufacturers can be difficult, as there is often a lack of standardization in terms of protocols and interfaces.

Chapter 2

Review of Literature

In this section, we introduce papers that we applied to our project, as well as some key features that provide us with materials to use. Firstly, the Internet of things (IoT) for building a smart home system, 2017 by Timothy Malche and Priti Maheshwary. A smart home system consists of applications built on top of IoT infrastructure. The smart home applications can have the following main functions, Alert The smart home system can sense its environment and send alerts to the user on a registered device or account. The alert consists of information related to environmental data. This information may include the level of different gases in the environment, temperature, humidity, light intensity, etc. Users can control things from the same place or from a remote location. This function even allows users to automate activities such as automatically switching on/off the air-conditioner when the room temperature is high/low. Home intelligence creates an integrated environment in the smart home in which the AI mechanism can identify and suitably react according to changing conditions and events. Some scenarios for illustration are automatically preparing coffee as soon as the user arrives, sending an alert to the user whenever suspected activity is detected at the door or inside the home, automatically ordering stuff whenever there is a shortage in the refrigerator, and sending notification to the electrician/plumber whenever maintenance is needed, etc. Alerts may be sent to the user on regular basis at a predefined time. This paper gave us an idea of how we can construct the IoT smart home structure for our project, further all the functions we got are based on the papers.

Secondly is a book from Dr. Le Trong Nhan, Build your own IoT gateway with Python. With the specific guide of how to use Adafruit, Google teachable machine,... The specific steps help us a lot in our project. In the tutorial, the opposite direction of communication with the data sent from the IoT Gateway to the Data Feed will be presented. For example, we can monitor the temperature of the incubator every 30 seconds. This data will be transformed into a graph on the Dashboard. Readers may think that a smart system can turn itself off when the desired temperature is reached. However, with real systems, the need for manual control is always necessary for possible exceptions. Also, how Microbit gateway associates with us, the serial of Gateway connection also fully supports many programming languages, from drag and drop to Python. The main purposes of this chapter are as follows: • Programming to receive data on Microbit circuit • Install library for Serial communication • Implement the program to send commands from Gateway for Microbit for the convenience of readers; small Python programs are introduced in each section. All the materials above helped us a lot in our project, improving the quality of the project and giving us the guidance we needed.

Chapter 3

Project Vision

3.1 Problem Statement

In our life nowadays, everything is becoming more and more convenient, for example, Tesla can help to drive you home with their autopilot, or Siri of Apple and Google Assistant of Google are being integrated into your smartphones to help you with minor works such as setting the alarm, finding music, article, etc or even tell jokes. However, there is still no big upgrade for our homes. Some companies have tried that and then fail, and some still find their way to help customers' homes become smart homes. The Internet of Things (IoT) is happening now. By implementing IoT, we can build a smart home system. A smart home is an application that is a combination of technology and services that specialize in the home environment with specific functions aimed at improving the efficiency, comfort, and security of the occupants. Smart homes filled with connected products are loaded with possibilities to make our lives easier, more convenient, and more comfortable. IoT is one of the most significant global economic drivers for the smart home market growth. At home, IoT-based equipment provides energy-saving features as well as communication through high-speed technology such as wifi. With the exponential growth of AI (machine learning, artificial intelligence), AI drives the IoT market toward a higher trajectory. But smart home still remains unfamiliar to many Vietnamese, so our project will initialize a small smart home project to get people to understand how a smart home project can be useful as well as applying IoT, thus AI.

3.2 Business Opportunity

With the evolution of IoT in our daily life nowadays, we believe that our project can have some business opportunities when it can become a real-life solution for our life, helping it become easier and more convenient. With IoT use, smart home has become an important issue. Many solutions were developed to make home automation more convenient and accessible 24/7, which can be applied in Vietnam. Applying intelligence in things connected to the Internet plays an important role in business model engineering. IoT makes a significant contribution to improving our product in daily life, thus, home security. This creates value for the customer and captures it in revenue. Through the IoT, we can apply smart homes to increase convenience and security in our homes.

3.3 Objectives

Our project's objective is to describe the integration of three loosely coupled components: smart home, IoT, and cloud computing. We propose a centralized real-time event processing application to orchestrate and timely manage the vast data flow in an efficient and balanced way, utilizing the strengths of each component's strengths. We describe the advantages and benefits of each standalone component and its possible complements, which may be achieved by integrating it with the other components providing new benefits raised from the whole compound system. Since these components are still in the development stage, the integration among them may change and provide a robust paradigm that generates a new generation of infrastructure and applications. As we follow up on the progress of each component and its corresponding impact on the integrated compound, we will constantly consider additional components to be added, resulting in new service models and applications.

3.4 Project Scope

This project only consists of easy-to-use features, which include some simple AI consist and real-time data updates. We gonna propose easy-to-approach AI and IoT-based projects for smart homes to introduce everyone to the application of IoT in the smart home. we explain the integration of classic smart home, IoT, and cloud computing. Starting by analyzing the basics of smart home, IoT, cloud computing, and event processing systems. We discuss their complementarity and synergy, detailing what is currently driving their integration. We also discuss what is already available in terms of platforms and projects implementing the smart home, cloud computing, and IoT paradigm

3.5 Constraints

AI in IoT raises the alarm about security and personal data. In an IoT context, it is important to underline that the violation of data ethics not only includes broken expectations, hacking, and breaches on the data level but also hacking on the device level that results in potential device takeover. In defining data ethics, we apply the definition by problems related to data (including generation, recording, curation, processing, dissemination, sharing, and use), algorithms (including artificial intelligence, artificial agents, machine learning, and robots), and corresponding practices (including responsible innovation, programming, hacking and professional codes), to formulate and support morally good solutions (e.g The balance between individual rights and the standard/public good is a well-known dilemma that IoT needs to and continuously has addressed. The critical question for managers as well as social and policymakers is: How do we make sure that we can protect user privacy and, at the same time, use data and infrastructure to optimize knowledge about users for product and service innovation and serve the common good at the same time? To address this question, managers using IoT/data for DBMS have to consider and provide answers to at least five associated aspects as identified in the research on information ethics: governance, transparency, accountability, privacy, and trust. Sometimes IoT users' lack of control and mistrust in DBMS is the case in which users share data with a company, and the company agrees to use it for other purposes than those the user would typically expect. Data sharing for ‘unknown’ purposes damage the trust in the DBMS and the company that owns them.

Chapter 4

Software Requirements Specifications

This chapter will have the functional and nonfunctional requirements of the project.

4.1 List of Features

1. Turn on the light and pump through Smartphone
2. Monitor the temperature and humidity of the house
3. Check the weather outside
4. Timer to turn the pump on/off
5. Open/Close door using face recognition
6. Camera base monitoring

4.2 Functional Requirements

Controlling and monitoring smart devices: The app should allow the user to control and monitor their smart devices, such as turning lights on and off and checking the status of locks.
Creating and managing automated tasks: The app should allow the user to create and manage automated tasks, such as setting the pump to turn on at a certain time.
Receiving notifications: The app should be able to send notifications to the user, such as

alerts when a door is unlocked.
Providing a user-friendly interface: The app should have a user-friendly interface that is easy to navigate and understand.
Ensuring data security: The app should ensure the security of the user's data, including the data transmitted between the app and the smart devices.

4.3 Quality Attributes

Usability: The app should be easy to use and understand, with a clear and intuitive user interface. This might involve testing the app with users to identify and address any usability issues.

Reliability: The app should be reliable and work consistently, even in the event of network or device failures. This might involve implementing fault tolerance and error handling mechanisms to ensure that the app continues to function as expected.

Maintainability: The app should be easy to maintain and update over time. This might involve writing clear and well-documented code, as well as implementing

4.4 Non-Functional Requirements

Performance: The app should be responsive and perform well, even when there are many devices connected or when the network connection is slow. This might involve optimizing the app's design and code for efficiency, as well as implementing caching or other performance enhancements.

Scalability: The app should be able to handle an increasing number of devices and users without degrading performance. This might involve designing the app to be modular and able to scale horizontally, as well as implementing load balancing and other techniques to distribute workloads evenly.

4.5 Use Cases/ Use Case Diagram

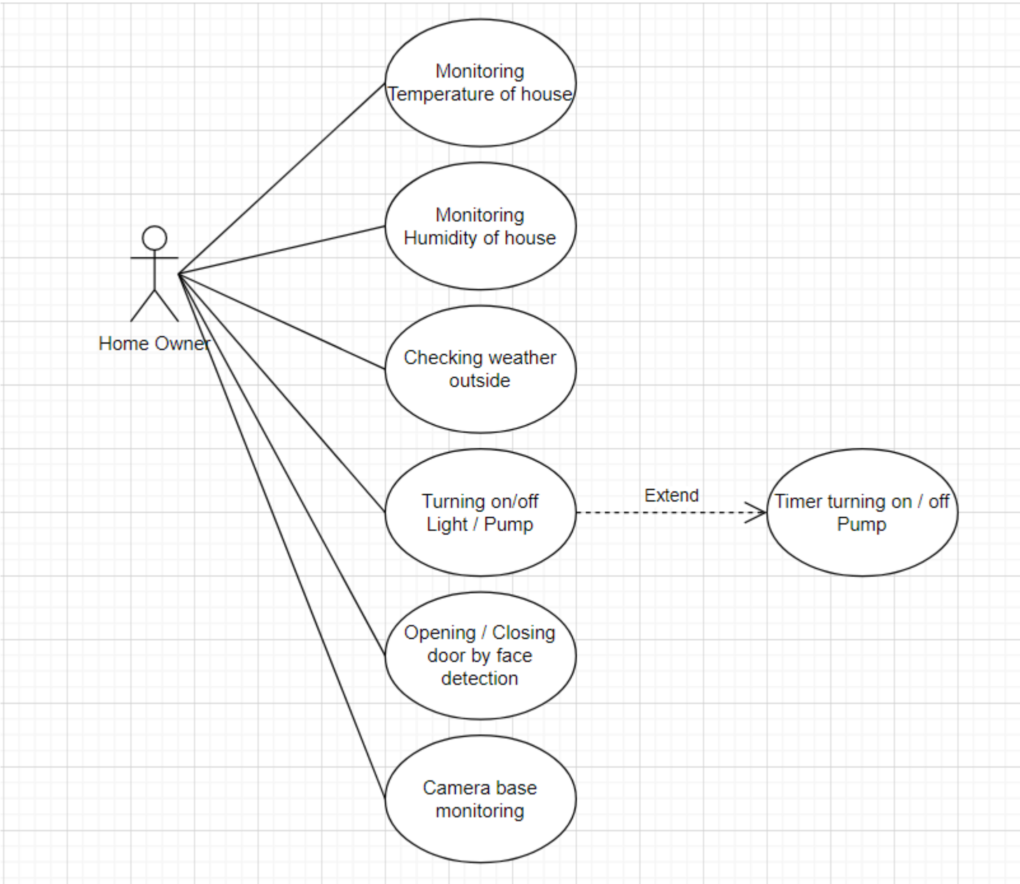

Figure 4.1: Use Case Diagram

4.6 Sequence Diagrams/System Sequence Diagram

The following are the sequence diagrams of buttons and functions in our project. The pump button is working similarly to the light buttons, while each of the other functions are working in a different way.

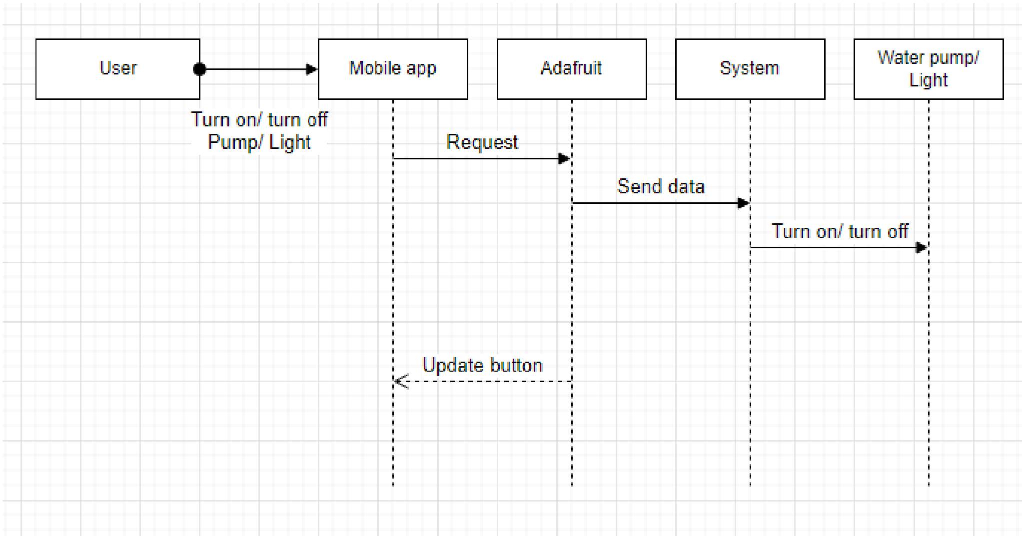

Figure 4.2: Sequence Diagram of Light/Pump button

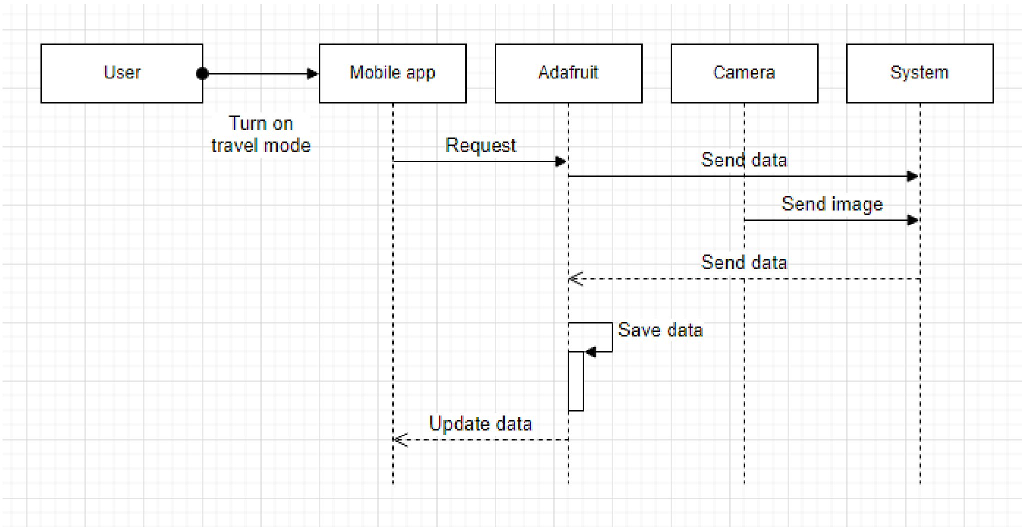

Figure 4.3: Sequence Diagram of Traveling button

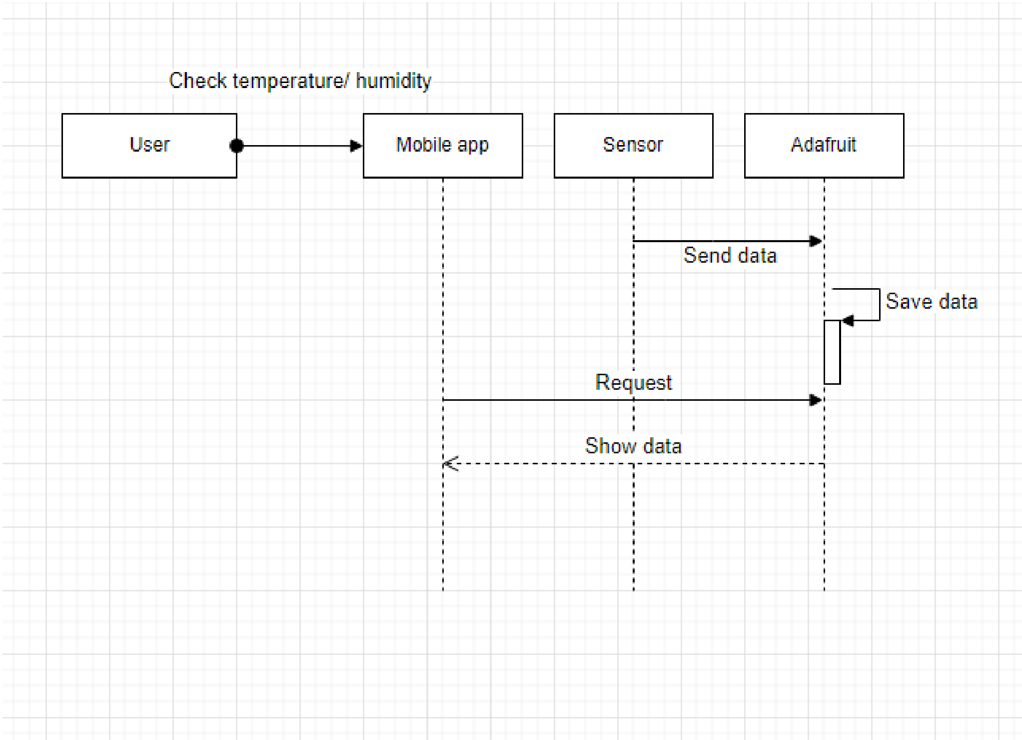

Figure 4.4: Sequence Diagram of Monitoring

4.7 Test Plan (Test Level, Testing Techniques)

The purpose of this test plan is to document the testing strategy and approach for the Smart Home project. The project involves building a software system to control and manage a variety of smart home devices, including lights, pump, security systems, and appliances.

The main objectives of the testing process are to:

- Verify that the system functions correctly and meets all specified requirements.
- Identify any performance issues or bottlenecks
- Ensure that the system is stable and reliable
- Identify and fix any defects or bugs

Test strategy: The testing process will follow a combination of manual and automated testing approaches. Specifically, we will use the following tools and techniques:

- Unit tests: We will use the Python unittest library to create a suite of unit tests that cover all of the core functionality of the system. These tests will be run automatically as part of the build process, and will ensure that all code is thoroughly tested and working as expected.

- Integration tests: We will use the Python pytest library to create a suite of integration tests that verify the integration between different components of the system. These tests will be run manually, and will ensure that all components are working together correctly.

- Acceptance tests: We will use the Python behave library to create a suite of acceptance tests that verify the overall functionality and usability of the system. These tests will be run manually, and will ensure that the system meets all specified requirements and provides a good user experience.

Test environment: The testing will be conducted on a development server running Ubuntu 20.04, with Python 3.8 and all necessary dependencies installed. The smart home devices will be simulated using a variety of mock objects and virtual devices.

Test cases: The test cases are organized by functional area and are numbered for easy reference.

- Device control:
  - Test case 1: Verify that the system can turn on and off a simulated light and pump.
  - Test case 2: Verify that the system can read the temperature and humidity of a sensor and temperature outside.
  - Test case 3: Verify that the system can recognize face and unlock the door.
- Scheduling:
  - Test case 4: Verify that the system can start and stop at certain time.
  - Test case 5: Verify that the system can recognize car when button is turned on.

- User interface:
  - Test case 6: Verify that the system provides a user-friendly interface for controlling devices.
  - Test case 7: Verify that the system provides a user-friendly interface for scheduling tasks.

Test data: The test cases will use a variety of test data, including different input values and expected output values. The test data will be stored in a separate file and will be used to drive the test cases.

Test execution: The test cases will be executed in the following order:

1. Run the unit tests
2. Run the integration tests
3. Run the acceptance tests

Each test case will be executed independently

4.8 Software Development Plan

The purpose of this Software Development Plan is to outline the steps and tasks involved in developing a software system for controlling and managing a variety of smart home devices. The project aims to provide a user-friendly interface for controlling devices, scheduling tasks, and accessing real-time data from the devices.

The scope of this project includes the following features and functionality:

Control of smart home devices such as lights, pumps, security systems, and appliances Scheduling of tasks to control the devices at specific times Real-time monitoring of device data such as temperature, humidity, and power usage User-friendly interface for controlling devices and accessing data The project will not include the development of any custom hardware or integration with any third-party systems.

Requirements The following functional and non-functional requirements have been identified for the project:

- The system must be able to control all of the specified smart home devices.
- The system must be able to schedule tasks to control the devices at specific times.
- The system must be able to retrieve real-time data from the devices.
- The system must provide a user-friendly interface for controlling devices and accessing data.
- The system must be scalable and able to support the addition of new devices in the future.
- The system must be secure and protect the privacy of user data.
- The system must have a response time of less than 1 second for all requests.

Design The overall architecture of the system will consist of a central server that communicates with the smart home devices and a mobile app that provides the user interface. The server will use the Python Adafruit-io library to communicate with the devices using the virtual COM serial communication protocol. The mobile app will use the REST API and MQTT client to communicate with the server and control the devices.

Implementation The following tasks will be completed as part of the implementation phase:

- Set up the development environment and install all necessary dependencies.
- Implement the server-side code for controlling the devices and providing the REST API.
- Implement the mobile app code for communicating with the server and controlling the devices.
- Test the system and fix any defects or issues that are discovered.

Testing: The testing process will consist of the following steps:

- Run a suite of unit tests to verify the core functionality of the system.
- Run a suite of integration tests to verify the integration between the server and the mobile app.
- Run a suite of acceptance tests to verify the overall functionality and usability of the system.
- The tests will be automated using the Python unittest, pytest, and behave libraries, respectively.

Deployment: The system will be deployed to a cloud-based server running Ubuntu 20.04, with Python 3.8 and all necessary dependencies installed. The server will be configured to automatically run the server code, and the mobile app will be available for download from the App Store and Google Play.

Maintenance: The system will be maintained through regular updates and patches to fix any defects or issues that are discovered. The development team will also be available to provide support and assistance to users as needed.

Risk management: The following risks have been identified for this project:

- Dependence on third-party libraries

4.9 Wire-frames

The following figure is showing our mobile app wireframe. This is where the function are going to be place and display on the mobile application.

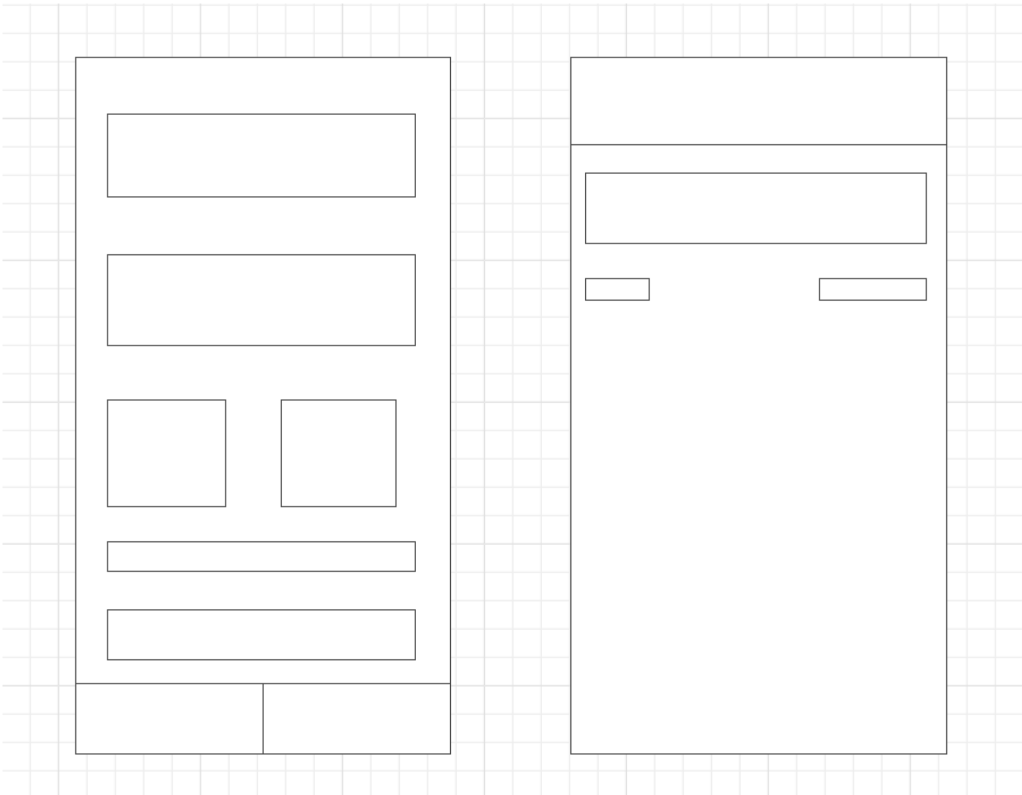

Figure 4.5: Wire Frame

4.10 UI Screens

With the sketching of the wireframe, we have create and coloring the UI for the mobile application. The following figure are our completed result after the working of designing team.

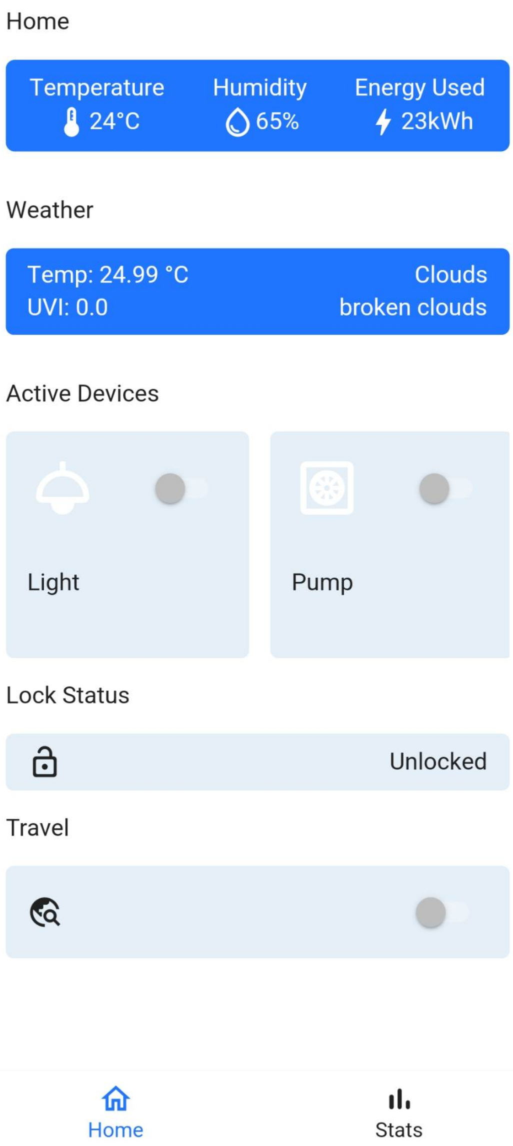

Figure 4.6: UI Screens

Chapter 5

Implementation Details

5.1 Architecture

The 5-layer model of IoT architecture is a reference model that divides the components and functions of an IoT system into five logical layers:

1. Device layer: This layer represents the physical devices and sensors that collect data from the physical world. It includes the hardware, firmware, and software components that are used to acquire, process, and transmit data from the sensors.

2. Network layer: This layer represents the communication infrastructure that is used to transmit data between the devices and the other layers of the system. It includes the communication protocols, networking hardware, and connectivity technologies that are used to transmit data over the network.

3. Edge computing layer: This layer represents the computing resources that are located near the devices and are used to process and analyze data in real-time. It includes edge servers, gateways, and other computing resources that are used to perform data processing, analytics, and decision-making tasks.

4. Platform layer: This layer represents the software and services that are used to manage and orchestrate the devices, edge computing resources, and applications in the system. It includes cloud services, software development kits (SDKs), and

other tools that are used to manage and control the devices and edge computing resources.

5. Application layer: This layer represents the applications and services that are built on top of the platform layer and that provide value to the end users. It includes the user interfaces, business logic, and other components that are used to build applications and services that are tailored to the specific needs of the users.

The 5-layer model of IoT architecture is intended to provide a logical framework for understanding the various components and functions of an IoT system and to help designers and developers understand the dependencies and relationships between the different layers of the system. It is not a prescriptive model, and the specific components and functions that are included in each layer will vary depending on the specific needs and requirements of the system.

Based on the 5-layer model of IoT architecture, we will divide the application into four basic components, including sensor node, central Gateway, server, and devices for data monitoring and remote control[2].

- Adafruit IO is a cloud-based platform provided by Adafruit Industries that enables users to collect, store, and process data from IoT devices and other sources. Adafruit IO provides a range of tools and services for building and deploying IoT applications, including APIs, libraries, and dashboards for collecting, visualizing, and analyzing data.

- The Python gateway would be a device that is located near the devices and sensors in the system, and that is used to perform data processing and analytics tasks on the data collected from the sensors. The Python gateway could use a variety of techniques and algorithms to process and analyze the data, such as machine learning, data filtering, data aggregation, and data visualization.

- The mobile application would be a software program that runs on a mobile device, such as a smartphone or tablet, and that provides a user interface for accessing and interacting with the IoT devices and data. The mobile application could use various techniques and technologies to communicate with the devices and data, such as APIs, libraries, or protocols, and could provide a range of features and functions for controlling the devices, scheduling tasks, and accessing data from the devices.

5.2 Implementation

5.2.1 Adafruit

Firstly, we create an Adafruit account and log in to it. Then, we created many feeds, each representing a physical component. There are two sensor feeds, five actuator feeds, and three feeds to save time and AI results. After that, we constructed an Adafruit Dashboard that has many blocks, and each block is connected to an existing feed. Our dashboard includes the following:

- 2 gauges for displaying the current temperature and humidity.
- 3 toggle buttons for showing the current state of the water pump, light, and travel mode.
- 3 text blocks for showing door locked status, car available status, and time for the auto pump.

5.2.2 Python Gateway

Python is responsible for several tasks:

1. Receiving information from temperature and humidity sensors and publishing them to the Adafruit server.
2. Using a camera to detect faces and unlock the door, and send status to Adafruit
3. Using a camera to detect a car that is available in the garage or not and send the status to Adafruit

4. Subscribing to the Adafruit server and waiting for commands to activate the pump, travel mode, auto pump mode and set time for auto pumping. The Python communicates with the Adafruit server using MQTT and Adafruit's Python library

5. Communicating with sensors and actuators using Modbus RTU over RS485 signals. This involves sending 8-byte commands, such as [1, 3, 0, 6, 0, 1, 100, 11], to retrieve data from sensors. To facilitate this communication, we have implemented a sendData class for sending and receiving information via a serial connection

6. The antiThief using a combination of the OpenCV and Adafruit IO libraries to perform object detection on a video stream from a webcam and to send and receive data from Adafruit IO. The video stream is captured using the VideoStream class from the imutils library and the object detection is performed using the cv2.dnn module from OpenCV. The object detection model being used is YOLOv3, and the model's weights and configuration file are read using cv2.dnn.readNet. The code then enters a loop where it reads frames from the video stream and performs object detection on each frame. The detected objects are displayed on the frame with bounding boxes and labels, and the resulting frames are displayed using the pyglet library. The code also makes use of the "AdafruitIO" library to send and receive data from Adafruit IO. The Client class is used to create a connection to the Adafruit IO service, and the MQTTClient class is used to subscribe to feeds and receive data. The code defines several functions to handle events such as successful connection and subscription, disconnection, and the receipt of a message. Finally, the code enters a loop where it continuously receives data from the actuator4 feed on Adafruit IO. If the received data is 1, the object detection loop described above is entered. If the received data is 0, the object detection loop is not entered, and the code waits for the next iteration of the loop. The code also receives data from the actuator5 feed and sends data to the actuator5 feed throughout the loop.

7. YOLO is a Convolutional Neural Network (CNN) for performing object detection in real time. CNNs are classifier-based systems that can process input images as structured arrays of data and recognize patterns between them (view image below). YOLO has the advantage of being much faster than other networks and still

maintaining accuracy.[**?**]OLO is able to make more precise predictions than other algorithms because it takes the entire image into consideration when making its predictions. It scores regions based on their similarities to pre-defined classes, giving it the ability to identify objects in any image, regardless of where it is located. This is due to the model being able to look at the whole image during testing, so its predictions are informed by the global context of the image.

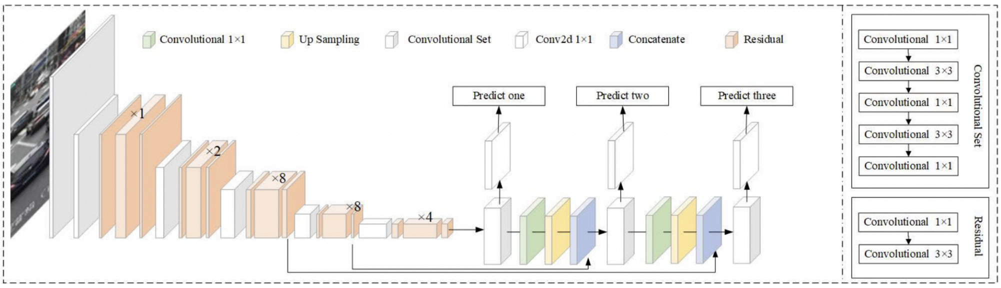

Figure 5.1: How YOLO v3 works

8. The faceDetect captures video from a webcam saves a frame as an image file and then uses the image classification model to predict the class label of the objects in the image. The imagedetector function loads the image file, resizes it to a fixed size (224x224 pixels), and normalizes the pixel values. It then runs the image through the image classification model, which returns a list of class probabilities. The function finds the class with the highest probability and prints it to the console.

5.2.3 Mobile Application

In this project, we developed a Flutter app for a smart home system. Therefore, this application can only run on Android or IOS phones/tablets. The app is designed to display the status of sensors, as well as allow users to toggle the light and water pump. Additionally, we included a feature that allows users to monitor the status of the home door and a car when the homeowner travels. Moreover, the user can set the time for the water pump to turn on automatically.

Upon launching the app, data is retrieved from the Adafruit IO server using a REST API. To maintain real-time data, the app uses the Stream function from the Flutter framework to send a GET request to the server every second.

In addition to data from the Adafruit IO server, the app also retrieves weather data for the location outside the home from openweathermap. This data is only obtained upon initial app launch or when the home tab is reopened.

To set up the daily pump routine, the user has the ability to select a specific time for the routine to occur. When the user saves their selection, the app creates a POST request and sends a string in the format "hh:mm" to the Adafruit IO server, with "hh" representing the hour and "mm" representing the minute. For example, the string "17:49" would correspond to a time of 5:49 PM.

Chapter 6

User Manual

Upon opening the app, user will see a screen showing some introduction to the app. User presses "Get Started" to go to the main screen.

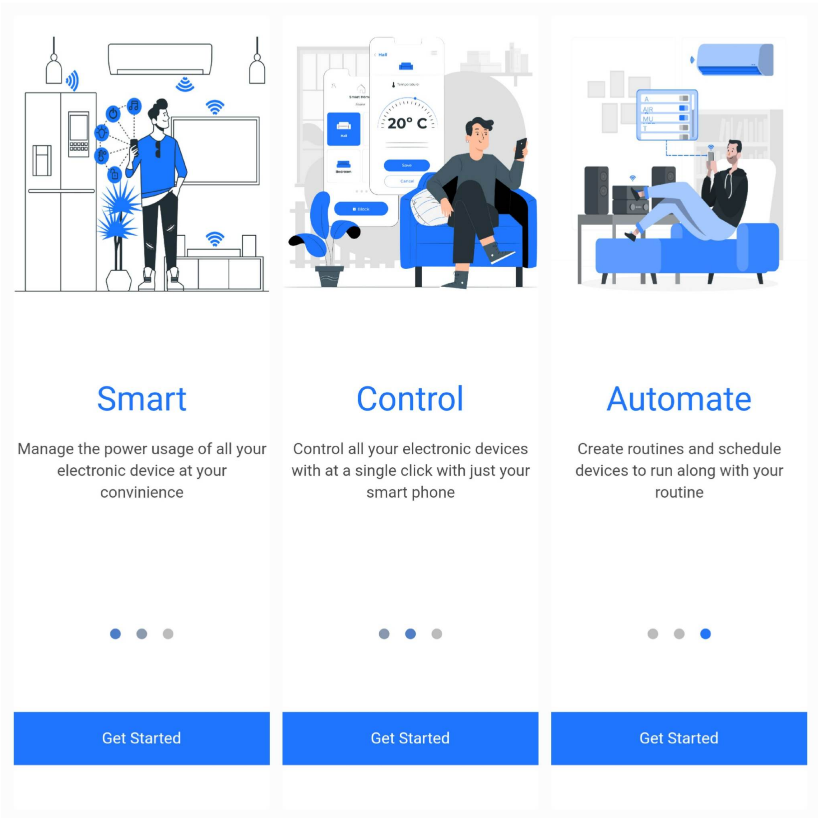

Figure 6.1: Introduction screen

6.1 Main Screen

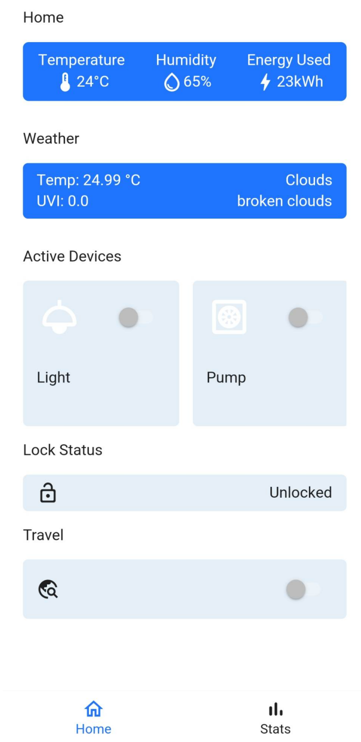

Figure 6.2: Mainscreen

The main screen displays 5 items of the application:

1. Home : display temperature, humidity, and energy usage for the day of the user's home.

2. Weather : display the temperatura, Ultra Violet Radiation Index (UVI) and a description of the weather outside the user's home.

3. Active Devices : Includes adjustable on/off status of 2 devices - light and pump.

4. Lock Status : display the lock status of the home's main door.

5. Travel : a toggle switch of the user's property protection mode.

6.2 Device Controller

Both devices, lights and pumps, have their own on/off switch for user convenience. Particularly for the pump, there is also a timer screen. (Figure 6.3)

The pump screen described allows the user to configure a daily routine for a pump. By

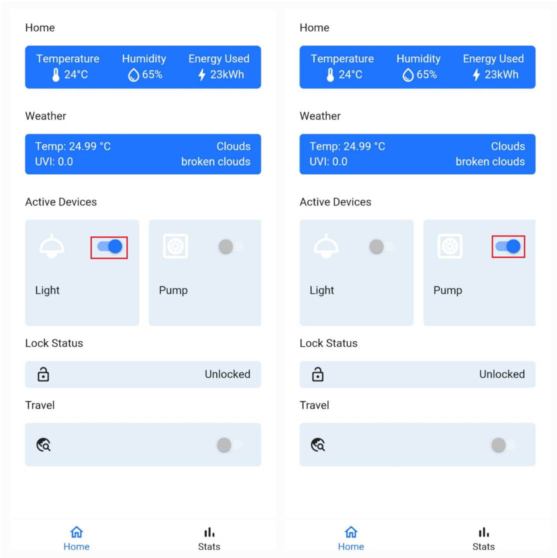

Figure 6.3: Mainscreen - device toggle switch

tapping on the designated time picker field, a dialog appears allowing the user to select the desired time. The user has the option to enable or disable the pump's automatic mode by using the provided toggle switch. Once the desired settings have been chosen, the user can save their changes by pressing the save button. (Figure 6.4)

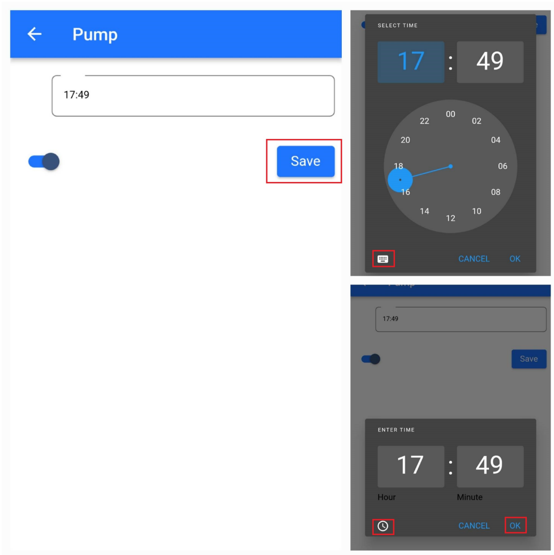

Figure 6.4: Pump timer screen

6.3 Safe Status

User can use the lock status item to check whether they have locked the main door or not (Figure 6.5). Besides, the travel mode helps users to protect their car in the garage. When the user leaves the house without taking the car, they turns on travel mode, the application performs tracking and will send an alarm when the car is moved out of the garage. (Figure 6.6)

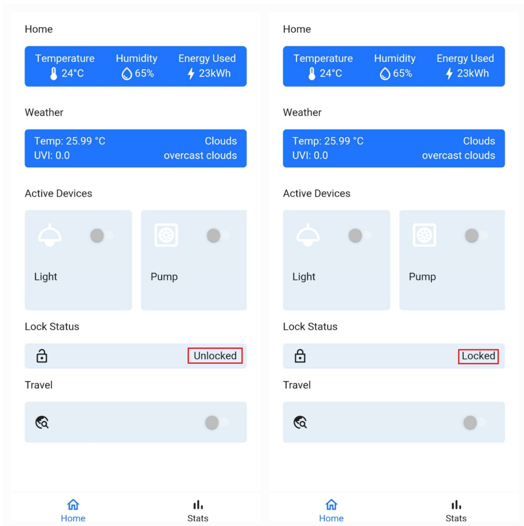

Figure 6.5: Mainscreen - lock status

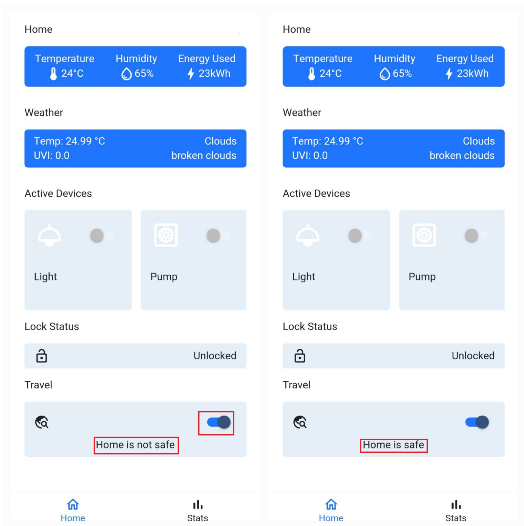

Figure 6.6: Mainscreen - travel mode

6.4 Power Useage

From the main screen, users can click "Stats" on the navigation bar to switch to the energy usage tracking page. Here, the screen displays the historical statistics of the home's electricity use along with the monetary value. Attached is an area chart for users to monitor.

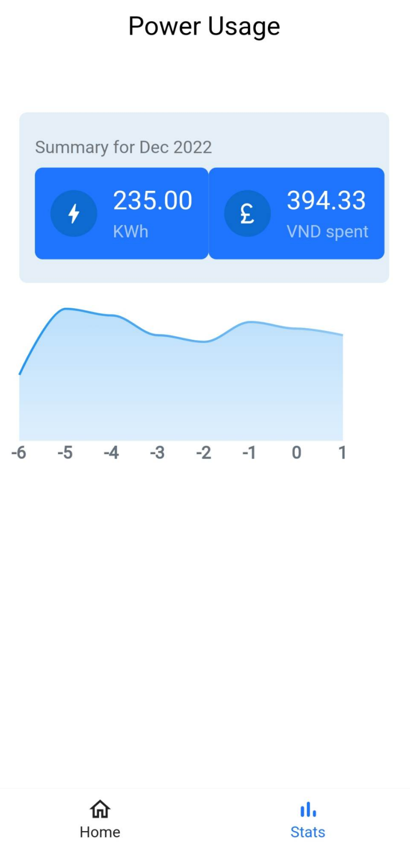

Figure 6.7: Statistics screen

Chapter 7

Conclusions and Future Work

The field of Smart home applications on smartphones appears to have a bright future. The feedback received about the field was positive, even though the usability of the prototype that was used for evaluation was considered to be low. The fact that the field was well received by all the users shows that even users with low technical interest might be interested in adopting the technology. This points to that Smart homes has the possibility to be a familiar integration to our future homes. The usability of the prototype created in this project affected the evaluation of the field of Smart home smartphone applications since the evaluation was conducted with this prototype. Since the usability of the prototype was considerably low, new user tests and field evaluations should have been performed if there had been more time available in this project.

About this project, first, some functions have been removed during work, they will be discussed again and completed soon. Second, there are still some functions that have not been completed as expected, so they will be revised and upgraded in the near future.

Bibliography

[1] Timothy Malche and Priti Maheshwary. Internet of things (iot) for building smart home system, 2017.

[2] Le Trong Nhan. *Build Your Own IoT Gateway with Python*. HCMUT, Vietnam, 2021.

[3] Tien-Wen Sung, Pei-Wei Tsai, Tarek Gaber, and Chao-Yang Lee. Artificial intelligence of things (aiot) technologies and applications. *Wireless Communications and Mobile Computing*, 2021:1–2, 08 2021.