

SMARTPLUG HOME  
AUTOMATION SYSTEM (SHAS) USING  
INTERNET OF THINGS (IoT)

WAN MUHAMMAD ADAM BIN AZMI  
CA20021

BACHELOR OF COMPUTER SCIENCE  
(COMPUTER SYSTEM AND  
NETWORKING)

UNIVERSITI MALAYSIA PAHANG

## UNIVERSITI MALAYSIA PAHANG

### DECLARATION OF THESIS AND COPYRIGHT

Author's Full Name : WAN MUHAMMAD ADAM BIN AZMI

Date of Birth

Title : SMARTPLUG HOME AUTOMATION SYSTEM (SHAS)  
USING INTERNET OF THINGS (IoT)

Academic Session : 2022/2023

I declare that this thesis is classified as:

- CONFIDENTIAL (Contains confidential information under the Official Secret Act 1997)\*
- RESTRICTED (Contains restricted information as specified by the organization where research was done)\*
- OPEN ACCESS I agree that my thesis to be published as online open access (Full Text)

I acknowledge that Universiti Malaysia Pahang reserves the following rights:

1. The Thesis is the Property of Universiti Malaysia Pahang
2. The Library of Universiti Malaysia Pahang has the right to make copies of the thesis for the purpose of research only.
3. The Library has the right to make copies of the thesis for academic exchange.

Certified by:



(Student's Signature)

Wan Muhammad Adam Bin Azmi

28/7/2023

(Supervisor's Signature)

Ts. Dr. Nor Syahidatul Nadiyah Binti Ismail

Name of Supervisor

Date: 28/7/2023

NOTE : \* If the thesis is CONFIDENTIAL or RESTRICTED, please attach a thesis declaration letter.

## THESIS DECLARATION LETTER (OPTIONAL)

Librarian,  
*Perpustakaan Universiti Malaysia Pahang,*  
Universiti Malaysia Pahang,  
Lebuhraya Tun Razak,  
26300, Gambang, Kuantan.

Dear Sir,

### CLASSIFICATION OF THESIS AS RESTRICTED

Please be informed that the following thesis is classified as RESTRICTED for a period of three (3) years from the date of this letter. The reasons for this classification are as listed below.

Author's Name  
Thesis Title

Reasons	(i)
	(ii)
	(iii)

Thank you.

Yours faithfully,

---

(Supervisor's Signature)

Date:

Stamp:

Note: This letter should be written by the supervisor, addressed to the Librarian, *Perpustakaan Universiti Malaysia Pahang* with its copy attached to the thesis.



## SUPERVISOR'S DECLARATION

I/We\* hereby declare that I/We\* have checked this thesis/project\* and in my/our\* opinion, this thesis/project\* is adequate in terms of scope and quality for the award of the degree of \*Doctor of Philosophy/ Master of Engineering/ Master of Science in .....

---

(Supervisor's Signature)

Full Name : TS DR NOR SYAHIDATUL NADIAH BINTI ISMAIL

Position : SENIOR LECTURE

Date :

---

(Co-supervisor's Signature)

Full Name :

Position :

Date :



## STUDENT'S DECLARATION

I hereby declare that the work in this thesis is based on my original work except for quotations and citations which have been duly acknowledged. I also declare that it has not been previously or concurrently submitted for any other degree at Universiti Malaysia Pahang or any other institutions.

A handwritten signature in black ink, appearing to read "Wan Muhammad Adam Bin Azmi", is written above a horizontal line.

(Student's Signature)

Full Name : WAN MUHAMMAD ADAM BIN AZMI

ID Number : CA20021

Date : 28/7/2023

SMARTPLUG HOME  
AUTOMATION SYSTEM (SHAS) USING  
INTERNET OF THINGS (IoT)

WAN MUHAMMAD ADAM BIN AZMI

Thesis submitted in fulfillment of the requirements  
for the award of the degree of  
Bachelor of Computer Science (Computer System and Networking) with honor

Faculty of Computing  
UNIVERSITI MALAYSIA PAHANG

JANUARY 2023

## **ACKNOWLEDGEMENTS**

First, I would like to express my special gratitude to my supervisor, Dr Nor Syahidatul Nadiah Binti Ismail for her guidance and support for me to complete my project. The supervision and suggestion guide me to the correct direction on idea generation. It is also enabled me to expand my knowledge and experiences on doing my final year project.

Next, I would like to thanks my parent and friends who help me to generate idea on the project. In addition, my friends also contribute in knowledge sharing which help me a lot in completing the proposal.

Finally, I would like to show my gratitude to the project coordinator Dr Danakorn for coordinate my project by giving guidance and encouragement for me to complete the proposal. Moreover, the guidance in proposal writing enable me to understand the correct format in writing the proposal.

## ABSTRAK

Dalam era teknologi yang berkembang pesat selari dengan arus permodenan, penggunaan “Internet of Things” (IoT) menjadi salah satu aspek yang kritikal dalam memastikan urusan seharian menjadi lebih mudah. Oleh itu, pendekatan harus di ambil dalam menerapkan penggunaan IoT dalam urusan seharian. “Smartplug Home Automation System” merupakan salah satu pendekatan dalam mengaplikasikan automasi rumah dengan membenarkan pengguna mengawal peralatan elektronik rumah hanya dengan menggunakan peranti pintar. Teknologi terkini terutamanya Internet of Things (IoT) boleh diterapkan sebagai salah satu pendekatan dalam memastikan pengguna turut boleh menggunakan teknologi canggih untuk memudahkan urusan seharian. Projek yang dicadangkan ini menggunakan IoT untuk mengawal perkakasan elektronik yang disambung kepada Smartplug Home Automation dalam masa nyata dan boleh diletakkan di dalam rumah atau mana-mana lokasi yang mempunyai punca kuasa yang boleh disambung. Dalam projek ini, sistem ini terdiri daripada 3 komponen utama iaitu, Arduino ESP8266, 2 Channel AC Relay dan Extension Plug. 2 Channel AC Relay akan mengawal arus bervoltan tinggi yang disambung terus daripada Extension Plug. Data dihantar melalui modul Wi-Fi yang dipasang dengan papan Arduino ESP8266. Data dihantar ke pangkalan data untuk memproses status perkakasan elektronik yang disambung dan membenarkan pengguna mengawal suis hanya dengan menggunakan telefon pintar. Matlamat sistem yang dicadangkan ini adalah untuk memudahkan pengguna mengawal perkakasan elektronik di dalam rumah tanpa perlu menutup suis secara fizikal. Metodologi Model Iteratif dan Incremental digunakan sebagai panduan untuk melengkapkan sistem yang dicadangkan daripada fasa perancangan awal hingga fasa penggunaan sistem. Pada akhirnya, objektif sistem yang dicadangkan dipenuhi dan kefungsan sistem seharusnya dapat mengawal perkakasan elektronik dengan menggunakan telefon pintar secara masa nyata dan boleh di akses dari mana-mana lokasi yang mempunyai jaringan internet.

## **ABSTRACT**

In an era of technology that is growing rapidly in line with the current of modernization, the use of the "Internet of Things" (IoT) has become one of the critical aspects in ensuring that daily affairs become easier. Therefore, an approach should be taken in applying the use of IoT in daily affairs. Smartplug Home Automation System is one approach in applying home automation by allowing users to control home electronic equipment just by using a smart device. The latest technology, especially the Internet of Things (IoT), can be applied as one of the approaches in ensuring that users can also use advanced technology to facilitate daily affairs. This proposed project uses IoT to control electronic hardware connected to Smartplug Home Automation in real time and can be placed in the house or any location that has a pluggable power source. In this project, this system consists of 3 main components namely, Arduino ESP8266, 2 Channel AC Relay and Extension Plug. 2 Channel AC Relay will control the high voltage current connected directly from the Extension Plug. Data is sent via a Wi-Fi module installed with an Arduino ESP8266 board. The data is sent to a database to process the status of the connected electronic hardware and allow the user to control the switch simply by using a smartphone. The goal of this proposed system is to make it easier for users to control electronic equipment in the home without having to physically close the switch. Iterative and Incremental Model methodology is used as a guide to complete the proposed system from the initial planning phase to the system deployment phase. In the end, the objectives of the proposed system are fulfilled and the functionality of the system should be able to control electronic hardware using a smart phone in real time and can be accessed from any location that has an internet network.

## TABLE OF CONTENT

<b>DECLARATION</b>	
<b>TITLE PAGE</b>	
<b>ACKNOWLEDGEMENTS</b>	<b>ii</b>
<b>ABSTRAK</b>	<b>iii</b>
<b>ABSTRACT</b>	<b>iv</b>
<b>TABLE OF CONTENT</b>	<b>v</b>
<b>LIST OF TABLES</b>	<b>x</b>
<b>LIST OF FIGURES</b>	<b>xi</b>
<b>LIST OF SYMBOLS</b>	<b>xiii</b>
<b>LIST OF ABBREVIATIONS</b>	<b>xiv</b>
<b>CHAPTER 1 INTRODUCTION</b>	<b>1</b>
1.1 Introduction	1
1.2 Problem Statement	2
1.3 Objective	3
1.4 Scopes	3
1.5 Thesis Organization	3
<b>CHAPTER 2 LITERATURE REVIEW</b>	<b>ERROR! BOOKMARK NOT DEFINED.</b>
2.1 Introduction	5
2.2 Existing System	5
2.2.1 Smartplug With Wireless Wi-Fi Technology (System 1)	5
2.2.2 Smartplug With Wireless ZigBee Technology (System 2)	10

2.2.3	Smartplug With Wireless Bluetooth Technology (System 3)	15
2.3	Analysis of comparison on existing system	18
2.4	Relevance of Comparison with project title	20
2.5	Summary	21
<b>CHAPTER 3 METHODOLOGY</b>		<b>22</b>
3.1	Introduction	22
3.2	Project Methodology	22
3.2.1	Traditional Method (Waterfall Model)	22
3.2.2	Prototype Model	23
3.2.3	Joint Application Development (JAD) Model	23
3.2.4	Rapid Application Development Model	23
3.2.5	Model for Proposed Project	24
3.2.6	Requirement Planning Phase	24
3.2.6.1	User Requirement	25
3.2.7	Design Phase	27
3.2.8	Construction Phase	28
3.2.9	Cutover Phase	28
3.3	Project Requirement	29
3.3.1	Functional requirements	29
3.3.2	Non-functional Requirements	29
3.3.3	Constraints	30
3.3.4	Limitation	30
3.4	Proposed Design	31
3.4.1	Architecture Design	31

3.4.2	General Flowchart	32
3.4.3	Hardware Flowchart	35
3.4.4	Login Flowchart	36
3.4.4.1	Subprocess Flowchart	37
3.4.5	Context Diagram	38
3.4.6	Use Case Diagram	39
3.4.7	Activity Diagram	40
3.4.8	Storyboard	42
3.4.8.1	Login Interface	43
3.4.8.2	Set Up Device Interface	44
3.4.8.3	Home Interface	45
3.5	Data Design	46
3.5.1	ERD	46
3.5.2	Database Dictionary	47
3.6	Hardware and Software	48
3.6.1	Hardware Support	48
3.6.1.1	ESP8266 Microcontroller	48
3.6.1.2	2 Channel AC Relay	49
3.6.1.3	Base Board for NodeMCU v3	50
3.6.1.4	DC Male Power Jack Connector To DG126 Converter	51
3.6.1.5	Extension Plug Socket	52
3.6.2	Software Support	55
3.6.2.1	Adruino IDE	55
3.6.2.2	Blynk Application	56

3.7	Interface Design	57
3.8	Proof of Initial Concept	59
	3.8.1 Blynk Configuration	59
	3.8.2 Blynk App Interface	60
	3.8.3 Hardware Setup	63
3.9	Testing/Validation Plan	66
	3.9.1 Blynk Application Testing Plan	66
	3.9.2 Sensors, Hardware, and Database Connection Testing Plan	67
	3.9.3 Scheduler Automation Testing Plan	68
3.10	Potential Use of Proposed Solution	69
3.11	Propose Interface Design (Using Blynk Application)	70
<b>CHAPTER 4 RESULTS AND DISCUSSION</b>		<b>71</b>
4.1	Introduction	71
	4.1.1 Development Environment	71
4.2	Mobile Application System	72
	4.2.1 Login Interface	72
	4.2.2 Forgot Page Interface	74
	4.2.3 Home Interface	75
4.3	Hardware Implementation	77
4.4	Prototype of Smart Blind Stick with Emergency Button	79
4.5	Blynk Application	80
	4.5.1 Blynk Connection	81
4.6	Testing and Discussion	81

4.6.1	Alert on SmartStick Application	82
4.6.2	Result on Ultrasonic Sensor Detection	83
4.6.3	Result on Water Sensor Detection	85
4.6.4	Result on Timer Setting Function	85
4.7	Conclusion	85
<b>CHAPTER 5 CONCLUSION</b>		<b>87</b>
5.1	Introduction	87
5.1.1	Hardware Constraints	87
5.1.2	Time Constraints	87
5.1.3	Cost Constraints	88
5.1.4	Technical Constraints	88
5.1.5	Future Works	88
<b>REFERENCES</b>		<b>90</b>
<b>APPENDIX A</b>		<b>92</b>
<b>APPENDIX B</b>		<b>93</b>

## LIST OF TABLES

Table 1	System one features	10
Table 2	System two features	14
Table 3	System three features	13
Table 4	Compare the existing system	18
Table 5	Login database dictionary	47
Table 6	Data database dictionary	47
Table 7	Hardware support	53
Table 8	Blynk application testing plan	66
Table 9	Sensors, hardware and database connection testing plan	67
Table 10	Scheduler automation testing plan	68

## LIST OF FIGURES

Figure 1.1	Flow for PSM 1	4
Figure 1.2	Flow for PSM 2	4
Figure 2.1	Implementation of hardware	7
Figure 2.2	Slave circuit hardware	8
Figure 2.3	Power consumption information	9
Figure 2.4	Connection of the smart plug between the appliances and HEMS controller	11
Figure 2.5	Simulation design of the proposed power node	12
Figure 2.6	Smartplug hardware	13
Figure 2.7	Graphical user interface of the proposed HEMS	14
Figure 2.8	Smartplug design	15
Figure 2.9	Android Application For Smartplug	16
Figure 2.10	App Modules	17
Figure 3.1	The Rapid Application Development Phase	24
Figure 3.2	Result of google form – Question 1	26
Figure 3.3	Result of google form – Question 2	26
Figure 3.4	Result of google form – Question 3	27
Figure 3.5	Result of google form – Question 4	27
Figure 3.6	Architecture Design of Propose System	32
Figure 3.7	General Flowchart	34
Figure 3.8	Hardware Flowchart	35
Figure 3.9	Login Flowchart	36
Figure 3.10	Scheduler Automation Flowchart	37
Figure 3.11	Context Diagram	38
Figure 3.12	Use Case Diagram	39
Figure 3.13	Hardware Activity Diagram	40
Figure 3.14	System Activity Diagram	41
Figure 3.15	Storyboard	42
Figure 3.16	Login Page	43
Figure 3.17	Set Up Device Interface	44
Figure 3.18	Home Interface	45
Figure 3.19	ERD Diagram	46
Figure 3.20	ESP8266 Microcontroller	48

Figure 3.21	2 Channel AC Relay	49
Figure 3.22	Base Board for NodeMCU v3	50
Figure 3.23	DC Male Power Jack Connector To DG126 Converter	51
Figure 3.24	Extension Plug Socket	52
Figure 3.25	Adruino Logo	55
Figure 3.26	Blynk Logo	56
Figure 3.27	Login Page Button	57
Figure 3.28	Home Page Button	58
Figure 3.29	Blynk Desktop Configuration	59
Figure 3.30	Process that user need to follow in order to use the Blynk App.	62
Figure 3.31	Hardware setup for Smartplug Home Automation System	63
Figure 3.32	Circuit diagram and label for ESP8266 Microcontroller	64
Figure 3.33	Electrical Appliances than can be used on the Smartplug Home Automation	65
Figure 3.34	Gantt Chart	70

## LIST OF SYMBOLS

% Percentage

## LIST OF ABBREVIATIONS

AC	Alternating Current
Amp	Ampere
App	Application
DC	Direct Current
Fig	Figure
IOT	Internet of things
SDLC	Software Development Life Cycle
SP	Smartplug
V	Voltage
Wi-Fi	Wireless Fidelity – High Fidelity

# CHAPTER 1

## INTRODUCTION

### 1.1 Introduction

Home Automation System refers to the integration of various electronic devices in a home to be controlled and monitored remotely. This can include lighting, heating and cooling systems, appliances, and security systems. There are two main types of home automation systems: Wired and Wireless. Wired systems use physical wiring to connect the devices, while wireless systems use Wi-Fi or other wireless communication protocols to connect and control the devices [1].

Examples of home automation systems include smart thermostats, smart lighting systems, and smart security systems. These systems allow for remote control and monitoring of the devices, as well as automation of certain tasks and routines [2].

The implementation of a Smartplug Home Automation System brings several advantages. First, it allows for easy control and monitoring of various electrical devices through a single mobile app. This can improve energy efficiency by reducing the waste of electricity from forgotten appliances. Second, the Smartplug can convert any non-smart electrical appliance into a smart IoT device, increasing the versatility of the Home Automation System. Third, the form of an extension plug allows for portability and use in any location with a power source. Fourth, it is simple to use, requiring only a connection to WiFi. Overall, the Smartplug Home Automation System improves the convenience and efficiency of controlling and monitoring home appliances and devices [3].

## 1.2 Problem Statement

Many people often forget to close the electrical appliances switch at home. It becomes a burden for someone who is away from home to return home to turn off the plug. It is also wasting time and energy for someone to get back home to do so. If the person does not do so, this will result in excessive waste of energy.

The lack of an effective and convenient solution for remote monitoring and control of home smoke detection systems remains a critical problem. Conventional smoke detectors can only alert occupants in the event of a fire, but do not provide a means to remotely monitor and control the situation. This problem is further compounded by the limitations of current Home Automation Systems, which have restrictions in terms of the devices that can be connected and controlled, as well as their portability. To address these challenges, there is a pressing need for a Smartplug Home Automation System that integrates smoke detection capabilities and offers a more versatile and convenient solution for remote monitoring and control of home smoke detection systems.

Consider a scenario in which a fire has started in your kitchen due to a faulty appliance. With a Smartplug Home Automation System that integrates a smoke detection sensor, the system will immediately detect the smoke and activate its safety features. For example, the system could automatically cut power to the faulty appliance to prevent the fire from spreading, as well as sound an alarm to alert any occupants in the home. Furthermore, you could receive an alert on your smartphone that provides real-time information about the situation and allows you to take action from a remote location. For example, you could remotely turn off the main power to your home to further prevent the spread of the fire, or contact emergency services for assistance. This scenario highlights the benefits of integrating a smoke detection sensor into a Smartplug Home Automation System, as it provides a more comprehensive solution for ensuring the safety and security of your home in the event of a fire.

Other than that, sometimes user will tend to forgot to turn off electrical appliances at home when they go out. In this case by using Smartplug Home Automation System, user will not have to worry because they can check the status and get

notification of the Smartplug in the mobile app. The status of the Smartplug will be shown in the mobile app as on or off and user can turn on or off the Smartplug by simple tap on the button in the mobile app. This will ease the user to control the electrical appliances at home and also will prevent excessive waste of energy if the plug being left on at that ime.

### **1.3 Objective**

1. To study exisiting Smartplug Home Automation System
2. To develop an Internet of Things (IoT) Smartplug Home Automation System
3. To evaluate the functionality of Smartplug Home Automation System by using Blynk Application.

### **1.4 Scopes**

The scopes of this project :

- **User Scope**

1. House owner who want to automate electrical appliances at home.
2. Business owners that require plug to be control wirelessly

- **System Scope**

1. Control the smartplug wirelessly using Blynk app.
2. Automate smartplug on and off time.

### **1.5 Thesis Organization**

This thesis is divided into five chapters. PSM 1 is covered in chapters one through three, while PSM 2 is covered in chapters four and five. The first chapter discusses the introduction, problem statements, project objectives, scope, and methodology. While Chapter 2 contains a review of the literature, Chapter 3 explains the methodology used in this project. The discussion and outcomes of the projects will then be discussed in Chapter 4. Finally, the conclusion and project outcome will be summarised in Chapter 5.

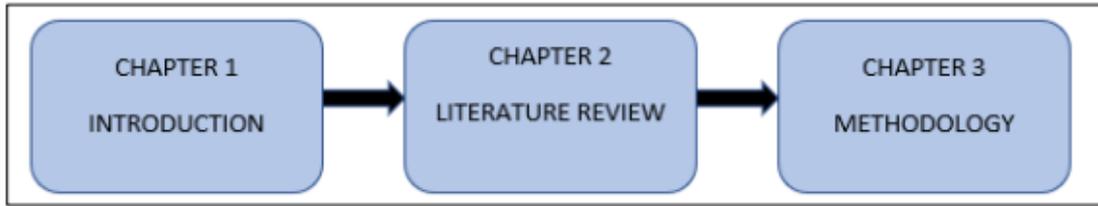


Figure 1.1 Flow for PSM 1

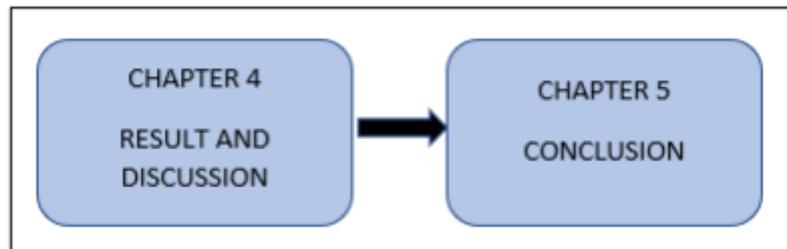


Figure 1.2 Flow for PSM 2

## **CHAPTER 2**

### **LITERATURE REVIEW**

#### **2.1 Introduction**

Nowadays, it is crucial to monitor and manage linked loads in homes, offices, hotels, hospitals, etc. This data presents a major possibility for the development of Internet of Things (IoT) systems, smart homes, smart buildings, and smart cities. It also allows for the regulation of loads. To increase a facility's energy efficiency, cloud computing makes real-time data accessible and enables real-time decision-making. Electrical parameters are typically measured across the entire installation in homes or other structures. Smart plugs enable individualised measurements from each connected device to the installation. This opens up a wide range of opportunities for modifying individual devices to optimise the installation's use of electricity. For instance, these devices can be connected at times when the cost of electricity is lower thanks to information from individual devices. The ability to exert control through the use of Smartplug, which enable manual or automatic control by the user, makes all of this possible.

#### **2.2 Existing System**

##### **2.2.1 Smartplug With Wireless Wi-Fi Technology (System 1)**

The wireless technology most commonly used in the implementation of SPs (Smartplugs) is Wi-Fi technology. This is because most homes have a Wi-Fi network to access the Internet. On the other hand, microcontrollers on the market with Wi-Fi

capabilities are numerous and inexpensive, such as the ESP8266 and ESP32, widely used in a large number of devices.

Ghazal (Ghazal, Akmal, Iyanna, & Ghoudi, 2016) developed a system where SPs function as slave units and the master unit coordinates the system in the house where the SPs are installed [4]. The system communicates with the master unit using the Zigbee protocol. In turn, the master unit uses the Wi-Fi connection of the home so that the user can perform the control and information functions. The devices work with voltages from 5 to 220 V AC can be measured with a maximum current of 5 A limited by a fuse. Current measurements operate with a 20 A ACS712 sensor. The SP was built based on the ATmega328 microcontroller. SPs that use directly implemented Wi-Fi technology can also be found in the state-of-the-art review. In this group is Thongkhao et al. (ThongKao, 2016). They developed an SM with Wi-Fi communication to remotely turn on and off electrical devices and monitor electrical energy consumption. Data is also sent to a web application so that it is accessible to the user on computers or mobile devices. A STPM01 device measures the electrical variables.

The smart plug has the same dimensions as a typical plug and has been optimized to fit inside it. To evaluate the accuracy of the system, an energy analyzer was used to compare readings for current, voltage, reactive power, apparent power, and power factor to those reported by the slave units. The system was tested with several loads, including a 12.5 W eco lamp, a 22 W fan, and a 100 W light lamp. The maximum error recorded in these tests was 5%. An 8 A solid state relay is used in the slave plug with a 6.3 A fuse, and it can measure a maximum load of  $240 \times 63 = 1512$  W before burning the fuse. For higher ratings, slave plugs can be designed by changing the choice of electronics. The cost of making the master and slave unit is 180 dirhams and 400 dirhams for the master

unit, respectively. These costs are based on prototype costs and can be significantly reduced during production.

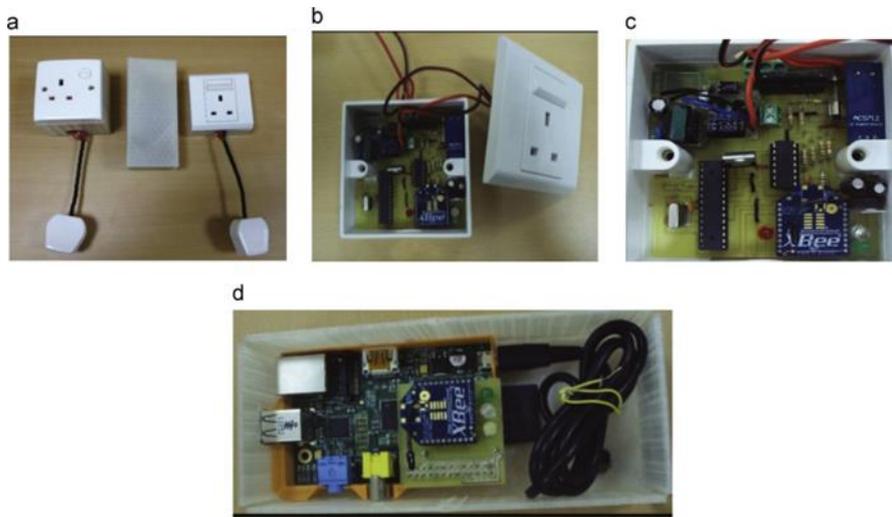


Figure 2.1 : Hardware of implemented smart plug. (a) Finished smart plug. (b) Insides of slave unit. (c) Close-up of slave unit. (d) Insides of master unit.

The hardware design of the slave circuit is shown in Figure 2.2. Calculating power consumption begins with reading current a voltage measurements. The voltage level is first brought down from 220 V AC to 5 V AC using the signal conditioning circuit in (A4–B5) to make the maximum use of the analog-to-digital converter resolution. The high and low voltage parts of the circuit are connected through buffers (B6) for protection. A 5 A fuse is also added for protection. The relay in F3 is controlled by the microcontroller in B11–C11 and is responsible for connecting or disconnected the supply. We use the ACS712 20 A current sensor (E3) and connect its output to another buffer and signal conditioning circuit before feeding the output to the Atmega328 microcontroller. The XBee module in B8 is connected serially to the microcontroller and is used to communicate with it. The master circuit is an embedded Linux board connected

with the microcontroller network through a XBee module and to the home area network through a Wifi module.

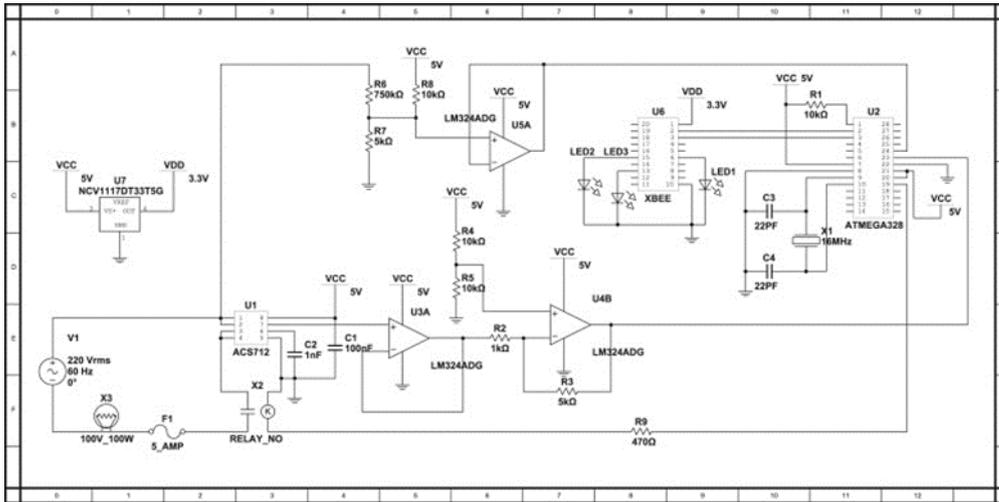


Figure 2.2 : Slave circuit hardware.

The mobile application (left-side of Figure. 2.3) exposes the power consumption information and provides the user with the ability to control all smart plugs. It starts by asking the user to design a digital model of their home by dragging and dropping different rooms as in Fig. 2.3(a). The purpose of this feature is for the user to easily localize the smart plugs. Next, the user places markers where the smart plugs were installed in the home as in Fig. 2.3(b). Clicking on an active marker brings the screen in Fig. 2.3(c) which shows a meter indicating the current level of power consumption and how much money and power has this smart plug consumed this month. Power consumption over time is also displayed to allow the user to identify patterns of usage and alter behavior accordingly as seen in Fig. 2.3(d). Real-time and detailed consumption information from participating households becomes available to distribution companies as shown in the

right side of Fig. 2.3. This information can then be data mined for patterns of consumptions or used as basis for awareness campaigns.

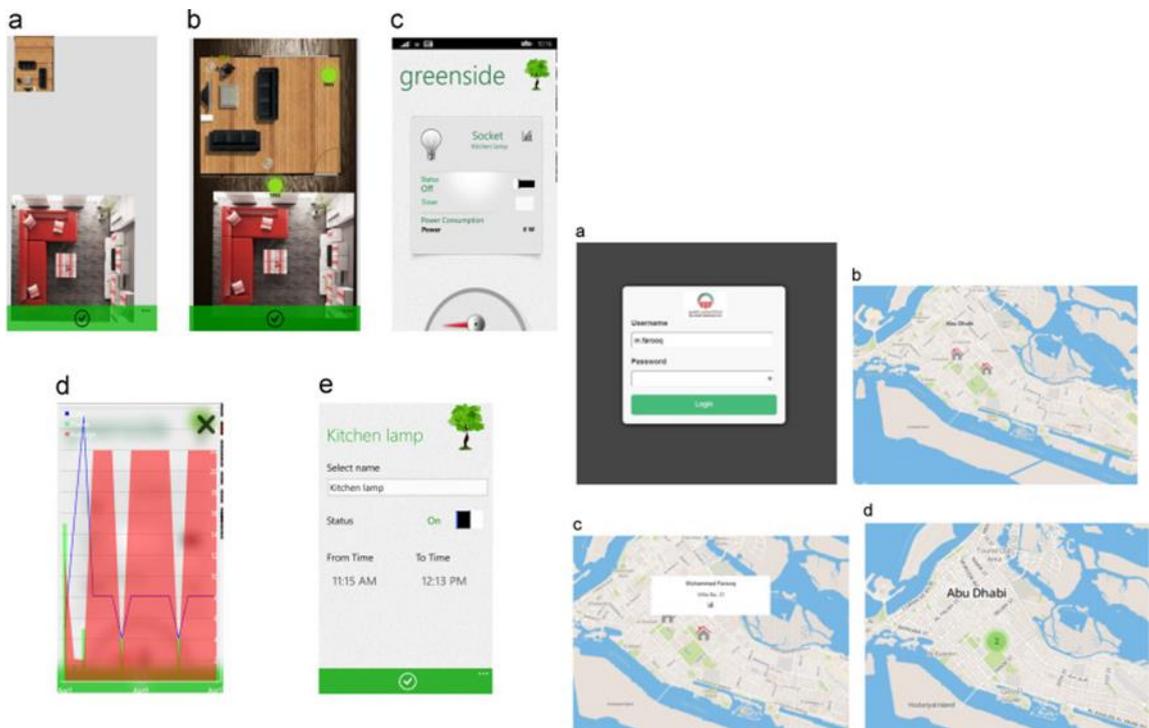


Figure 2.3 : Mobile application providing power consumption information and control over smart plugs. Left: (a) User designs a digital model of home. (b) User marks locations of smart plug installations. (c) Power consumption and control are available to user. (d) Power consumption data over time is available to user. (e) User can program certain

smartplug for automated conservation. Right: (a) Login view. (b) Map view. (c) Marker clicked. (d) Marker clustering.

Table 1 : System one features

<b>Advantages</b>	<b>Disadvantages</b>
Can be controlled through Wi-Fi	Only 1 socket can be controlled at a time
Can measure power usage	No physical on or off button
Automated conservation mode	
Statistic of power consumption	
Easy install device design	

### 2.2.2 Smartplug With Wireless ZigBee Technology (System 2)

This ZigBee technology is also widely used in the development of general IoT devices, or more specifically in the electrical sector, such as SMs or SPs. In the literature review, different devices have been identified with the implementation of this technology.

Maytham (Maytham, et al., 2015) built a prototype SP to monitor the consumption of electrical devices in homes and control of the electrical energy consumed [5]. The communication implemented in the SP relies on the Zigbee protocol to send the voltage and current measurements, in addition to managing the on/off relay to control the on/off switching of the electrical device to which the SP is connected. Elma (Elma & Selamogullari, 2015) proposed an SP with Zigbee communication; an algorithm that controls the power using electronic switches does the power control.

Figure 2.4 shows the smart plug connected between the electrical appliances and the HEMS controller. The smart plug implementation steps begin with Zigbee nodes

configuration to initialize the system parameters and to indicate the maximum and the minimum values of the measured signal.

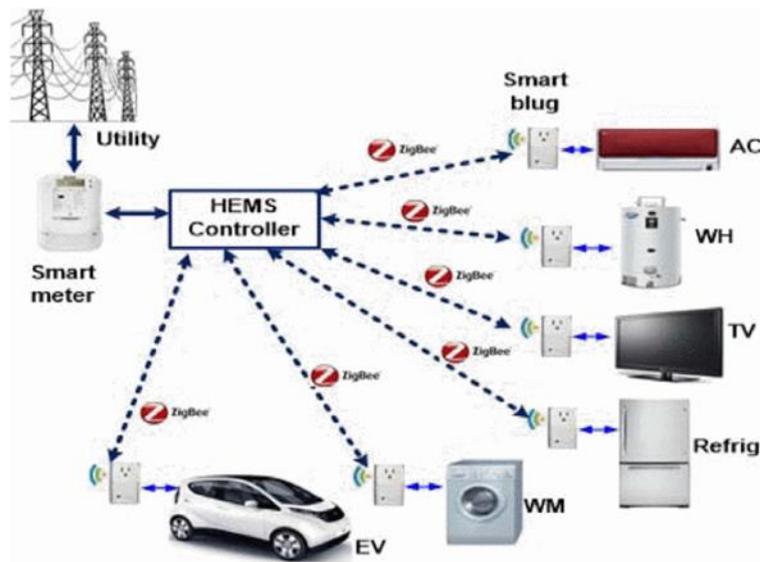


Figure 2.4 : Connection of the smart plug between the appliances and HEMS controller

The proposed power node shown in Figure 2.5 represents a wireless communication transceiver node which is located at each home appliance. This transceiver node has the essential elements used for measuring the energy parameters as well as receiving control signals required for managing energy saving. The first input channel or the current channel, is connected to (AD0) or pin 20 of the Zigbee Pro. Current is sensed when the household current supply passes through the current sensor. A voltage signal proportional with that current is produced and delivered to the conditioning circuit.

An op-amp circuit is utilized to convert the AC resultant current signal into average DC form suitable for Zigbee input to process and to send the current value by wireless means.

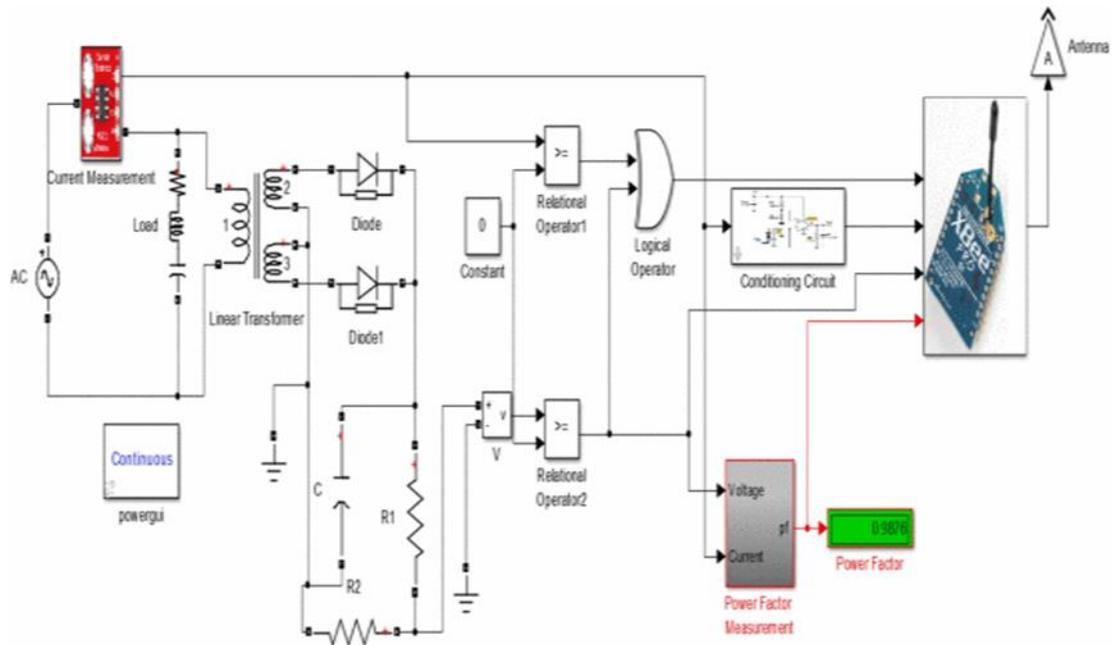


Figure 2.5 : Simulation design of the proposed power node

The performance of the prototype card is tested in terms of data transmission rate and accuracy of the received data. Analytical estimation of energy saving is also made. The current sensor type ACS712  $\pm 5$  A is used and its output is connected to the Zigbee port. Figure 2.6 shows a prototype of the developed smartplug. The circuit board sizes are designed suitable to be placed inside the plug. To test the accuracy of the received

data, an energy analyzer is used to determine readings for current and voltage. The received signals are compared with that measured and displayed on the oscilloscope.



Figure 2.6 : Smart plug hardware

The HEMS includes GUI and associated software to assist customers in monitoring the total and individual power consumption, monitor the on/off status of home appliances, the room temperature, and water temperature for WH as shown in Figure 2.7. As depicted in the figure, the dashboard can display the power consumption in (Watt) for each load, the room and water heater temperature in °C, the total power consumption and

also the total energy in (kWh). Furthermore, it is possible to turn on/off each home appliance manually from a push button switch display in the developed dashboard.

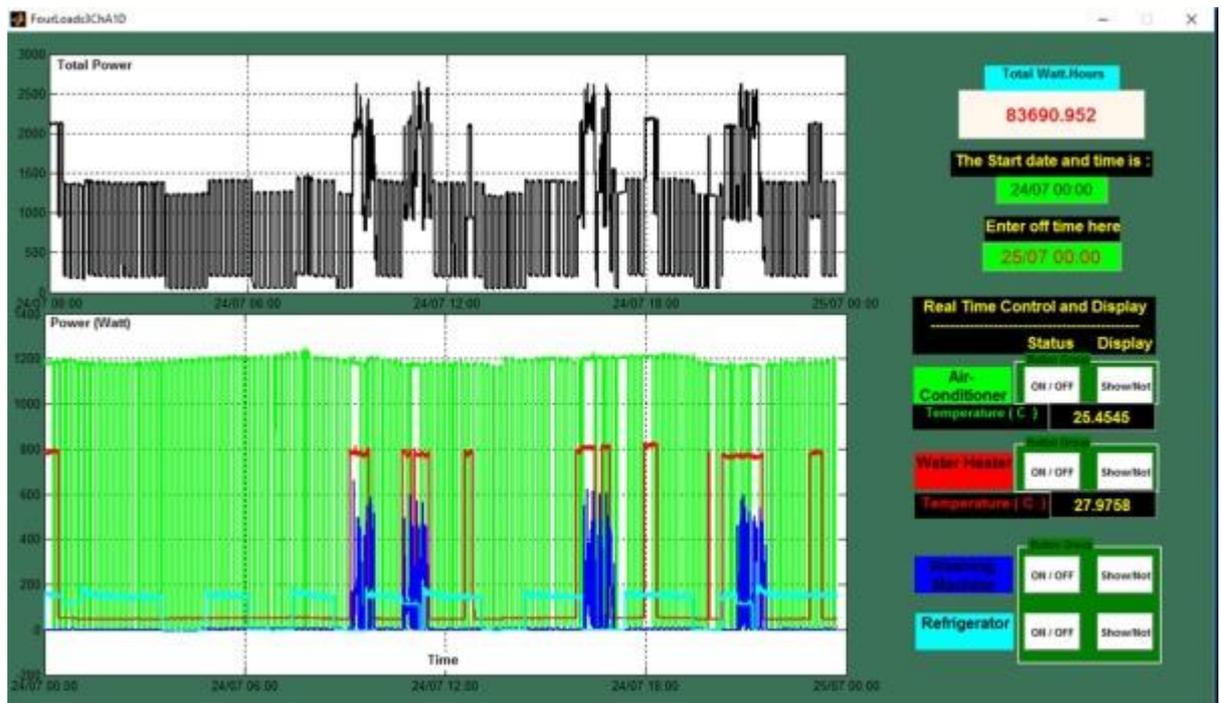


Figure 2.7 : Graphical user interface of the proposed HEMS

Table 2 : System two features

Advantages	Disadvantages
Can monitor power consumption	HEMS coordinator need a PC to control, does not have mobile apps
Control on or off plug by using ZigBee network linked wwith HEMS controller	Limited to only 1 socket per device
ZigBee RF Module can pickup test signal and transform it into data that can be monitored	Need to have HEMS controller in order to use the system
Equipped with wireless sensor that can measure accurate power consumption signals	

### 2.2.3 Smartplug With Wireless Bluetooth Technology (System 3)

In the literature, fewer devices with Bluetooth wireless technology are found. Nonetheless, as Horvat et al. (Horvat, Lukac, Pavlovic, & Starcevic, 2015) implemented an SP to monitor and control electricity consumption in households [6]. The communication is developed with a Bluetooth protocol. The Smartplug is controlled via a smart phone with an app designed for the Android operating system.

Figure 2.8 shows a prototype of the developed smartplug. Smartplug is a simple in design, very compact, which is important because it will occupy no more than one physical socket on a power strip. Also, outlet contains a button which provides alternative way of basic functionalities. Along with the button, there is RGB indication, informing about the current load of the connected appliance and signaling the working status or progress during the firmware upgrade. RGB indication can be switched on or off. Colors change from yellow, followed by green and blue, ending with red as signal of maximum load. Basic functionalities consist of power parameters known as active and reactive power, current and voltage levels and power factor. There is possibility to limit the power consumption, so when power consumption reaches given value, device will be turned off. Device can change its state from on to off and vice versa in two ways, instant control or setting the countdown timer.



Figure 2.8 : Smartplug design

Application provides all the features that Smart Plug offers, in a user-friendly way. Main challenge for application is to design intuitive UI that is uniform across wide

range of screen sizes and resolutions. Application is divided in three screens. First screen presents a list of accessible plugs, second screen is dedicated to plug control and third screen provides advanced configuration options for device.

Based on Figure 2.9, Android 4.3 introduces a built-in platform support for Bluetooth Low Energy, and provides API that apps can use to discover, connect, and read/write characteristics [2]. This could be achieved in earlier versions via third party libraries. Since 50.1 % of all Android devices are running on 4.3 and newer platform versions [3], it is not hard to justify selection of the platform.



Figure 2.9 : Android Application For Smartplug

Figure 2.10 shows how the application is divided into modules. First one represents graphical user interface (GUI) and second module is background functionality. GUI contains three screens. Second screen is the most important, because it represents the device, and provides user intuitive way to control and monitor Smart Plug.

Application functionality is implemented as a background service and is completely decoupled from graphical user interface.

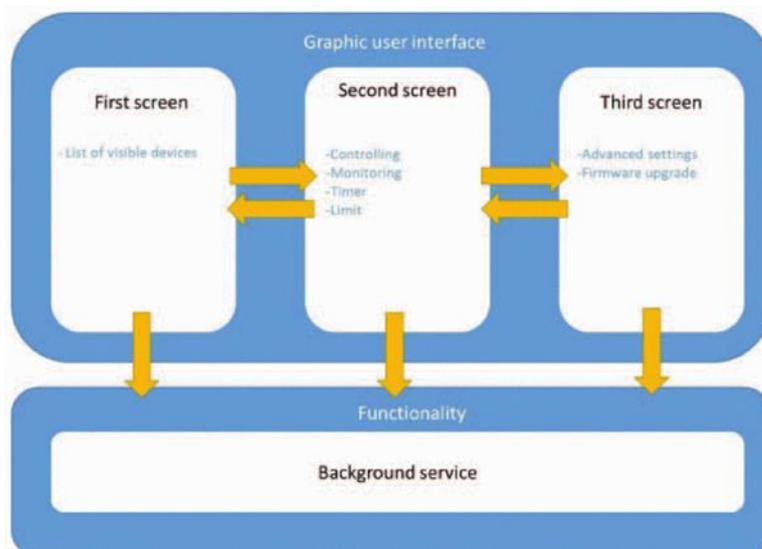


Figure 2.10 : App Modules

Table 3 : System three features

Advantages	Disadvantages
Only requires bluetooth connection no Wi-Fi network needed	Limited connection within bluetooth range
RGB indication light for notification when changed state	Limited to only 1 socket can be controlled
Countdown timer	No physical on or off button
User friendly interface mobile apps for controlling plug	

### 2.3 Analysis of comparison on existing system

Following an evaluation of each of the selected existing systems. The following is a summary of the comparison of three existing systems chosen for the literature study and the proposed system.

Table 4 : Compare the existing systems

Specifications	System 1	System 2	System 3	System 4
Hardware Used	<ul style="list-style-type: none"> <li>- ATmega328 microcontroller</li> <li>- ACS712 sensor</li> <li>- XBee module</li> <li>- AC Relay</li> </ul>	<ul style="list-style-type: none"> <li>- HEMS controller</li> <li>- Wireless communication transceiver node</li> <li>- AC Relay</li> <li>- ZigBee RF Module</li> <li>- XBee module</li> <li>- AC Relay</li> </ul>	<ul style="list-style-type: none"> <li>- Bluetooth module</li> <li>- BLE devices</li> <li>- Aduino ESP32</li> <li>- AC Relay</li> </ul>	<ul style="list-style-type: none"> <li>- Arduino ESP8266 V3</li> <li>- NodeMCU WiFi Controller Board</li> <li>- Base Board for NodeMCU V3</li> <li>- DC Male Power Jack Connector To DG126</li> <li>- 2 Channel AC Relay</li> </ul>
Software	<ul style="list-style-type: none"> <li>- Android Studio</li> <li>- C++</li> </ul>	<ul style="list-style-type: none"> <li>- C++</li> <li>- HEMS Controller Web App</li> </ul>	<ul style="list-style-type: none"> <li>- Android Studio</li> <li>- C++</li> <li>- Aduino IDE</li> </ul>	<ul style="list-style-type: none"> <li>- Aduino IDE</li> <li>- C++</li> </ul>
Mobile Application	Standalone Android app	No	Standalone Android app	Blynk App supports both Android

				and IOS users
Database	Web-based	Web-Based	Bluetooth API	Blynk API
Wi-Fi Communication	Yes	Yes	No	Yes
Cloud-Based	Yes	Yes	No	Yes
Number Of Plug Socket Available	1	1	1	2
Additional Features	<ul style="list-style-type: none"> <li>- Measure power usage</li> <li>- Generate statistic of power consumption</li> </ul>	<ul style="list-style-type: none"> <li>- Monitor power consumption</li> <li>- Accurate sensor to measure power consumption</li> </ul>	<ul style="list-style-type: none"> <li>- Only requires bluetooth connection no Wi-Fi network needed</li> <li>- Countdown timer</li> <li>- Have physical button to on or off Smartplug</li> </ul>	<ul style="list-style-type: none"> <li>- Have physical button to on or off Smartplug</li> <li>- Have more than one socket to control</li> <li>- Can be access anywhere with internet connection</li> </ul>

To summarize, the hardware which was commonly used in three of the existing smartplug system was an AC Relay. the AC Relay module will be used in the proposed system to control high voltage and high current load from the main power source. It also will convert AC power to DC power in order to be able to use any electrical appliances on the smartplug.

WiFi Module is present on all existing smartplug system except for “Smartplug With Wireless Bluetooth Technology (System 3)”.System 3 uses bluetooth module in order to communicate between the smartplug hardware and the user controllable interface. Why bluetooth is not implemented to the proposed system is because bluetooth have range

limitations that prohibit the user to control the smartplug from a far location. Wifi module will be the most suitable because the system will be able to be use even it is from a far location with a network connection. This option will be more suitable for the proposed system in order to fully utilized the capability of the smartplug.

All existing systems were designed to control electrical appliances that are connected to the smartplug by using user controllable user interface. Based on the existing system, “Smartplug With Wireless Wi-Fi Technology (System 1)” used Standalone Mobile App. “Smartplug With Wireless ZigBee Technology (System 2)” used Web App. “Smartplug With Wireless Bluetooth Technology (System 3)” used Standalone Mobile App. Different approach is taken for the proposed system in order to make sure the system is be able to use on cross platform smartphone operating system. In the proposed system, Blynk App is chosen because it is easy to install on Android and IOS respectively. This approach use the same concept from the System 1 and System 3 which is by using Mobile Apps but better support which also supports IOS users. Web based approach from System 2 is not chosen because not suitable for ease of use for the end user.

#### **2.4 Relevance of Comparison with project title**

As to compare the three-existing system with the new proposed system, the AC Relay module will be used in the proposed system to control high voltage and high current load from the main power source. It also will convert AC power to DC power in order to be able to use any electrical appliances on the smartplug. Other than that that ESP8266 board module is used as wifi module that give the microcontroller access to the WiFi network. Moreover the ESP8266 board is is planned to be used for the data exchange over the WiFi with the database to connect with user smart applications that were proposed in this new system. Next, the use of Blynk App is utilized in the development of the mobile application .

Other than that, the Blynk App is proposed to be used as a mobile app software that allow user to monitor the status of the socket on the smartplug and and control the on off button through the app. The Blynk App is integrated with the Blynk API and the smartplug hardware in order to able to utilize the system. The Arduino IDE software will

be a proper software to program the ESP8266 board as referring to all the existing systems. The proposed system will be developed on the Internet of Things (IoT) concept, making it a cloud-based system that is entirely reliant on the internet for communication networks in real time.

## **2.5 Summary**

In summary, the Smartplug Home Automation System is most suitable for use at any houses that need to integrate their electrical appliances at home with the smartplug. In addition, the smartplug is able to control electrical appliances connected to the smartplug wirelessly even when user is not at home and also is able to monitor the status of the plug and control by using Blynk App in smartphone. Lastly, the Smartplug Home Automation System is also embedded with IoT technology integrated with the Blynk API to allow user to set timer in the smartplug in order to prevent power wastage from unused electrical appliances at home. The user is only required to log into Blynk App to be able to use the Smartplug Home Automation System.

## **CHAPTER 3**

### **METHODOLOGY**

#### **3.1 Introduction**

This chapter will go over the project methodology, which will describe the project development workflow. The system development methodology is a framework that guides the developer through the process of structuring, planning, and developing the system. Traditional Method (Waterfall Model), Prototype Model, Joint Application Development (JAD) Model, and Rapid Application Development (RAD) Model are among the methodologies available for system implementation.

#### **3.2 Project Methodology**

The methodology is important in system development because it provides a clear framework for the developer to develop a system, which reduces the number of defects and delays during system deployment. A well-chosen methodology allows the developer to deliver a stable system, make better time and budget estimates, and provide a clear understanding of the system to the customer, developers, and end users. Traditional Method (Waterfall Model), Prototype Model, JAD Model, and RAD Model are examples of system development methodologies. The methodologies include both pros and cons.

##### **3.2.1 Traditional Method (Waterfall Model)**

This is one of the most well-known systems development life cycle models. It is characterised by distinct objectives for each phase of system development, with each phase requiring completion before proceeding to the next phase; the developer is required to follow the specified sequence; and returning to previous phases is prohibited [7]. The advantages of this model are that it is simple and easy to comprehend and that it ensures

the system is delivered on time because each phase has a deadline that the developer must meet in order to deploy the system on time [8]. However, this model is unsuitable for changes that are irreversible; consequently, it will be very difficult to make any system modifications during the testing phase.

### **3.2.2 Prototype Model**

This system development model employs the technique of prototyping, which can aid in the comprehension of requirements. The prototype is created, and prototype testing is conducted [9]. The testing phase of the prototype allows both the client to gain a better understanding of the system and the developer to gather the system's requirements. This model has the potential to reduce system failure because it provides the developer with clear system requirements [10]. However, it may be more expensive to build the prototype, and if the client is overly involved, there is a high risk of deviating from the requirements and making numerous modifications to the prototype.

### **3.2.3 Joint Application Development (JAD) Model**

It is a method for accelerating the development of a system by bringing together all customers and designers or developers to precisely clarify the business requirements necessary for developing the system [11]. It improves the efficiency of system deployment time and ensures a better understanding of the customer in order to meet their satisfactions, as JAD provides a forum for the customer and developer to identify the necessary requirements [12]. Due to the difficulty of gathering all stakeholders at once, this model may waste time and resources when planning and scheduling stakeholders.

### **3.2.4 Rapid Application Development Model**

RAD is an agile methodology that permits reverting to a prior phase. In contrast to the Traditional Method, RAD emphasises rapid prototyping and encourages customer or user feedback during the testing phase [13]. After the requirements collection phase concludes, system development and testing iterations are permitted. There are a number of disadvantages to using the Rapid Application Development (RAD) model, including its reliance on the team to identify the system's requirements, its suitability for modularized systems, which necessitated the developer's possession of advanced modelling skills, and its expense due to the high cost of automated code generation.

However, RAD model characteristics have surpassed the limitations of the Traditional Method (Waterfall Model).

### 3.2.5 Model for Proposed Project

The RAD model is the most suitable methodology for this proposed project, as determined by the evaluation. As this model is applicable to projects with a short duration and ensures prompt delivery. In addition, it is simpler to incorporate any changes because iterations are permitted between the deployment and testing phases. The RAD model includes four phases, including the Requirements Planning Phase, the Design Phase, the Construction Phase, and the Cutover Phase. The phases sequence for the RAD model is shown below. Figure 3.1 below shows the Rapid Application Development Phase (RAD) lifecycle.

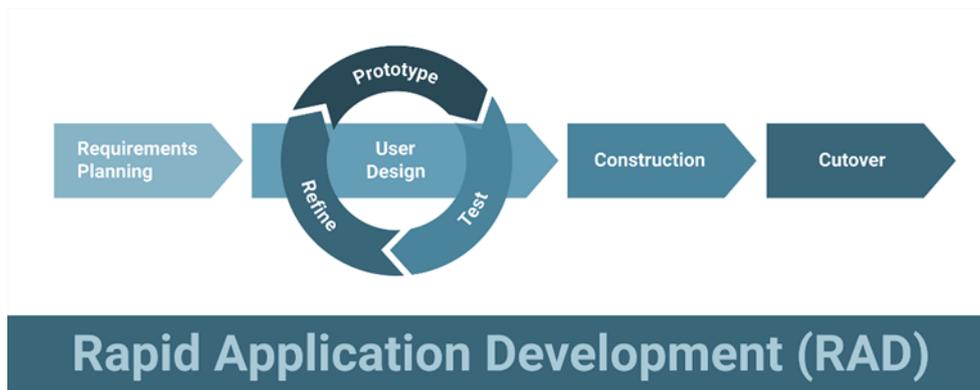


Figure 3.1 The Rapid Application Development Phase

### 3.2.6 Requirement Planning Phase

In requirement planning phase, the activities such as identify the objectives of the system and collect the requirements needed based on the objectives will be carried out. There are 3 objectives are identified for this propose system. It is the initial stage before the development phase and the requirements collection ensure better understanding of the system.

The propose project is to develop a smartplug home automation system using Arduino. The function of the smartplug is to automate and control electrical appliances via network in real time by using Blynk app. The target users will be user that away from home that

need to monitor electrical appliances at home, as they can use this smartplug home automation system to monitor the status and control the plug in real time.

In this phase, understanding the requirements of the users is needed for example identify the type of electrical appliances that can be monitored such as status of light at outside house, security cameras, and the fans connected to the smartplug. Moreover, the threshold for each socket in the smartplug also needed to study to ensure the smartplug to be reliable to be left at home as the user leaves the house.

### **3.2.6.1 User Requirement**

Google form survey is conducted for requirement planning that is open to community, UMP staff and students to collect the limitations of the previous system, seek for improvements, and acquire new requirements. It takes 25 respondents from Universiti Malaysia Pahang and community to collect reliable and credible data on the necessity of smartplug system usage.

A sets of questions will be ask to gather the requirements as follows :

- i. Question 1 - Review on existing traditional extension plug (Satisfied/Unsatisfied)
- ii. Question 2 - Do you interested on smartplug to replace existing traditional extension plug that be able to control through your phone?
- iii. Question 3 - How much socket do you need on extension plug or smartplug?
- iv. Question 4 - Do you prefer Android App or IOS app to use the smartplug system?

Result obtained from the survey above will be presented in visual form which shown in figure below. Figure 3.2 illustrates that 64% respondeants are unsatisfied on using existing traditional extension plug, and 16% are statisfied with current traditional extension plug, and 20% are not sure. According to the findings in Figure 3.3, 68% of respondents are interested in smartplug system to replace exisiting traditional extension plug. Moreover, in Figure 3.4, 84% of respondents need more than 1 socket for the extension plug. Other than that, in Figure 3.5, 80% respondents prefer to use both Android

and IOS app to use the smartplug system. Therefore, the proposed system must be implemented in order to make a difference to ease daily life of the user.

Review on existing traditional extension plug  
25 responses

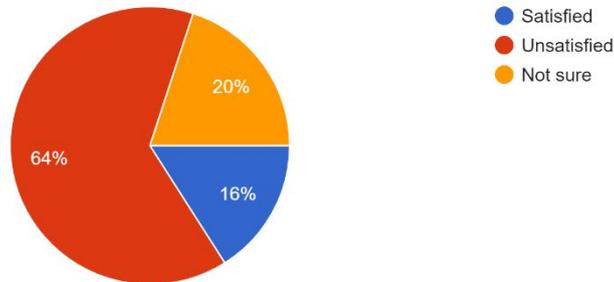


Figure 3.2 Result of google form – Question 1

Do you interested on smartplug to replace existing traditional extension plug that be able to control through your phone?  
25 responses

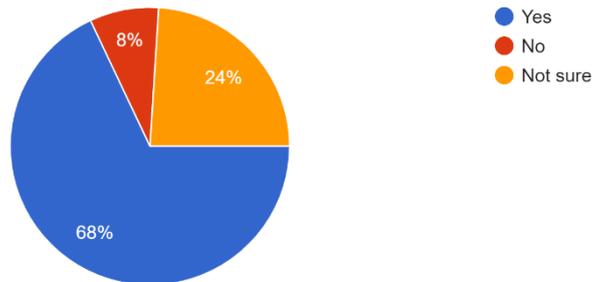


Figure 3.3 Result of google form – Question 2

How much socket do you need on extension plug or smartplug?  
25 responses

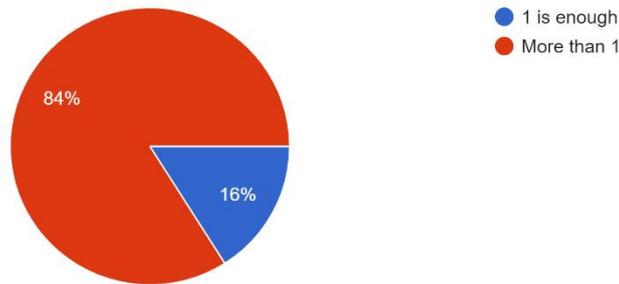


Figure 3.4 Result of google form – Question 3

Do you prefer Android App or IOS app to use the smartplug system?  
25 responses

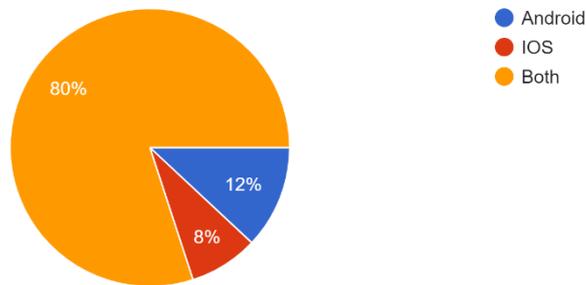


Figure 3.5 Result of google form – Question 4

### 3.2.7 Design Phase

After gathering the system's requirements, the design phase will commence, during which system models will be created. Such tasks as system design and interface design will be executed. In system design, techniques such as flowchart, use case diagram, and context diagram are used to illustrate how the system interacts with the end user. The flowchart used to illustrate the process from reading water quality data to making a decision by analysing the threshold level to alert sending and data storage. In the use case diagram, depict the user's interaction with the system. The users consist of the officer and the administrator. Users, also known as actors, can perform activities such as viewing reports,

managing databases, assigning tasks, etc. The context diagram depicts the data flow from the Arduino microcontroller to the system, as well as the data flow from the system to the user or vice versa. Next, a draught of the interface design will be created, using primarily storyboards to illustrate the navigation in the water monitoring system's interface. In this phase, the appropriate construction strategy will also be chosen.

In addition to conducting a literature review, other existing related systems will be analysed, which will aid in the design phase of the proposed system and in the generation of ideas. It is also one way to enhance the system's capabilities. The current system comprises an Internet of Things-integrated smartplug home automation system, an Internet of Things based on real-time electrical appliance monitoring system, and a low-cost, real-time, autonomous smartplug timer and notification system. Comparing the existing system to the proposed system for system improvement is also performed.

### **3.2.8 Construction Phase**

System development will be the primary focus of the construction phase. The proposed system will include stages such as development and testing. Implementing the proposed system requires the writing of code for the smartplug home automation system. Arduino IDE will be used to write the microcontroller's code, whereas the Blynk app will be utilised for interface design and system deployment. In the Arduino IDE, C++ is one of the programming languages used. In this phase, any system modifications or enhancements will also be implemented. System development, coding, and system testing are among the tasks that will be performed

### **3.2.9 Cutover Phase**

This is the deployment phase, during which the system's development is complete and it is prepared for launch. The smartplug home automation system will be subjected to testing to ensure its performance and functionality. The purpose is to ensure that the system adheres to the objectives and scope of the smartplug home automation system. In this phase, a sample of electrical appliances with different types of functionality will be used to test the system and determine whether it meets the requirements and objectives of this system. In the phase designed to reduce the error rate of the proposed system, tasks

such as minor debugging and code correction will be performed. After testing is complete, the system is ready for launch; however, in this project, testing will be conducted on the system prototype.

### **3.3 Project Requirement**

The project requirements consist of few key elements such as functional and nonfunctional requirements, limitations and the potential constraints will be encountered in the development of this project.

#### **3.3.1 Functional Requirement**

- i. The system shall integrate with Blynk API to store the smartplug status and control the smartplug through Blynk App user interface.
- ii. The system shall turn off socket of the smartplug when user tap the off button.
- iii. The system shall turn on socket of the smartplug when user tap the on button.
- iv. The system shall allow user to connect to the WiFi to establish network connection for the smartplug.
- v. The system shall allow user to set timer for the smartplug to on or off at specific time.
- vi. The system shall allow user to control the socket of the smartplug individually.
- vii. The system shall allow user to turn off the smartplug socket physically without using the app.

#### **3.3.2 Non-Functional Requirement**

- i. The system shall designed with user friendly and simple user interface layout.
- ii. The system shall designed with clear button to on or off the smartplug socket.

- iii. The system shall allow user to view the status of socket of the smartplug whether it is on or off.
- iv. The system shall support more than one electrical appliances at a time.
- v. The system shall be able to accommodate new functions without major update on the smartplug
- vi. The system shall allow more than one user to use the smartplug at a time
- vii. The system shall support wireless connection to be able to connect to the WiFi.

### **3.3.3 Constraints**

- i. This Smartplug Home Automation System is designed to control, monitor the status, set timer for the electrical appliances connected to the smartplug. This system does not monitor the current or power usage of the connected electrical appliances.
- ii. Users are only able to use the Blynk App to control the smartplug and need to use specific Blynk account in order to use the smartplug system.
- iii. All users are unable to create their own Blynk account because the Blynk API need to be integrated and programmed to only one specific account. This is because the system is only designed for prototype uses and not ready for commercial use.
- iv. This system require internet connections at both ends which is the smartplug need to be connected to WiFi and the user need to have internet connection to use the Blynk App in order to fully utilize the function of the smartplug.

### **3.3.4 Limitations**

- i. The Smartplug Home Automation System need to properly setup and handled by installer to be able to use the system. It is because the system is only designed for prototype uses and not ready for commercial use.

- ii. Only supports 2 electrical appliances at a time connected to the socket of the smartplug. More plug can be added but need to use multiple socket plug extension to be used but it will still be controlled by each of the socket which is all device at the extension at a time.
- iii. More plugs can be added but need to redesigned the hardware structure of the smartplug. For example, if need to upgrade to 3 socket or 4, the AC Relay need to be replaced to support 3 channel or 4 channel relay of the new added socket.

### **3.4 Proposed Design**

#### **3.4.1 Architecture Design**

In this proposed system, the microcontroller used is ESP8266. Since the system is a smartplug home automation system, it consists of 2 Channel AC Relay and a DC Male Power Jack Connector To DG126 Converter. The relay and power jack connector for andruino board are all directly connected to the ESP8266 microcontroller. The ESP8266 has a build in Wi-Fi module which is together with the ESP8266 NodeMCU board. The Wi-Fi module enables the data to be sent to the database. The database will send data or instruction to the Blynk App to the user be able to retrieve the data response. The user will be able to turn on or off the intended socket from the smartplug. The user also can send actions to the ESP32 through manual physical button on the smartplug. A detailed design with components related are shown in Figure 3.6 below.

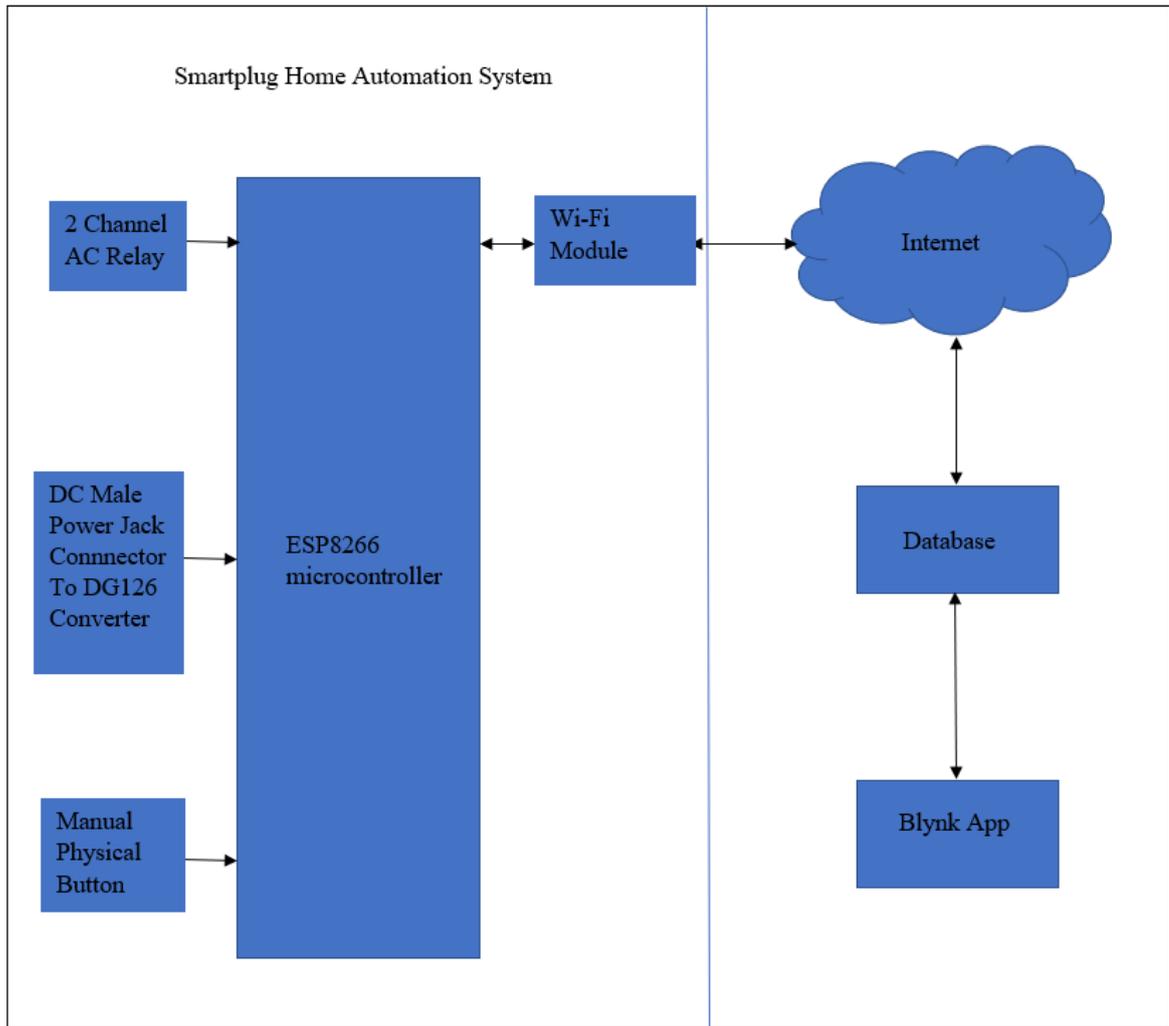


Figure 3.6 : Architecture Design of Propose System

### 3.4.2 General Flowchart

The proposed system is Smartplug Home Automation System, which will have the capability to turn on off the socket of the smartplug wirelessly from any location, and monitor the status of connected electrical appliances connected to the smartplug. As the first step, the system has to be powered on to establish the connection between the microcontroller and the components. As a second step, the relay will be activated where it will allow current to pass through the smartplug to the socket. Since, all of the components in this system generate analogue signals, the analogue modules that connect the sensors to the ESP8266 microcontroller are used to convert the analogue signals to readable digital signal.

As the part of the Smartplug Home Automation System, the ESP8266 microcontroller will also receive the signal from the physical button and the signal will be analysed by the microcontroller whether the switch of the socket in on or off status signal is received by the ESP8266 and the receive signal will process based on the programming language code added to the microcontroller whether to cut or pass through the current flow. In case the switch is on, it will pass through the current flow and the ESP8266 will show the status on in the Blynk App.

As the part of the automation of the smartplug, the ESP8266 microcontroller will receive the processed signal and data is transmitted and store in the Blynk API database through using the Wi-Fi module. User will be able to automate the smartplug by set a scheduler to be able to turn and off the socket of the smartplug automatically at the desired time. Figure 3.7 depicts the overall flow chart of the system, beginning with the hardware implementation and progressing to the software implementation.

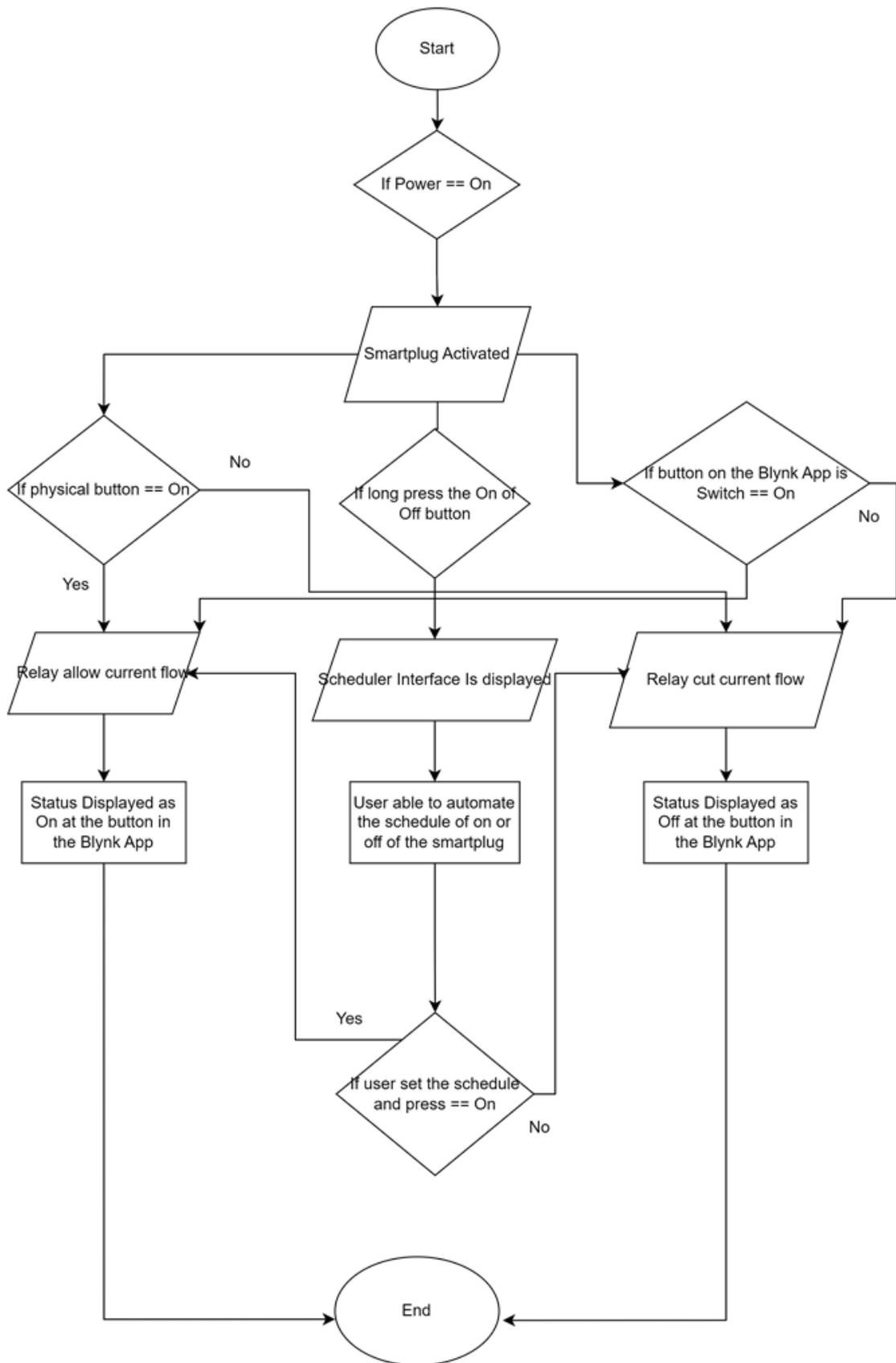


Figure 3.7 : General Flowchart

### 3.4.3 Hardware Flowchart

The hardware used in this system includes an ESP8266, AC Relay and an 3 pin plug socket with switch. The hardware is needed to regulate power from the power source or power supply, in which the AC Relay will pass through current flow, cut the current flow from reaching the electrical appliances connected to the socket. Any interaction on the physical switch or virtual switch in the blynk app will send the signal to ESP8266 for data processing, resulting in automatic current flow or cut on the AC Relay. The status of the plug will be displayed on the virtual switch on the Blynk App. To allow data transmission ESP8266 must be turned on, and the power source or power supply must be available, as the ESP8266 require power to function. The flowchart for the hardware is shown in Figure 3.8 below.

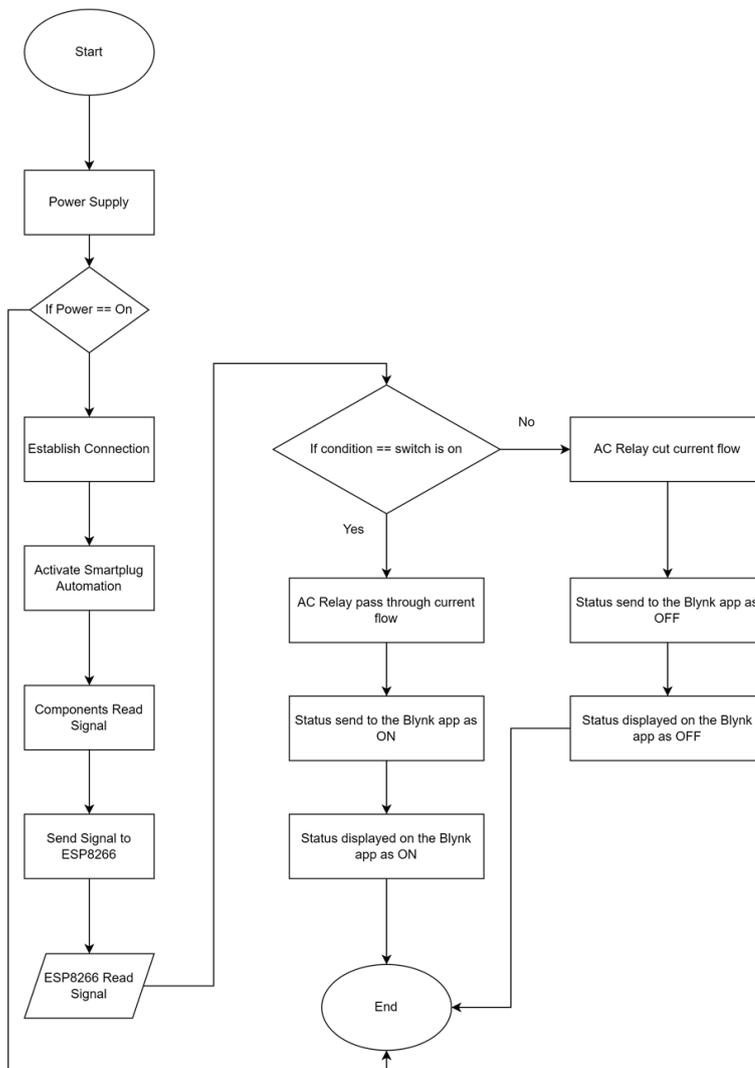


Figure 3.8 : Hardware Flowchart

### 3.4.4 Login Flowchart

The user need to login before access into the system. The valid email and password need to be entered. User can monitor the Smartplug Home Automation plug status. The email and password will be used for verification. If the email and password match with the database, then user will be authorized to access the system. Based on the Figure 3.9 below, it shows the flowchart for login interface.

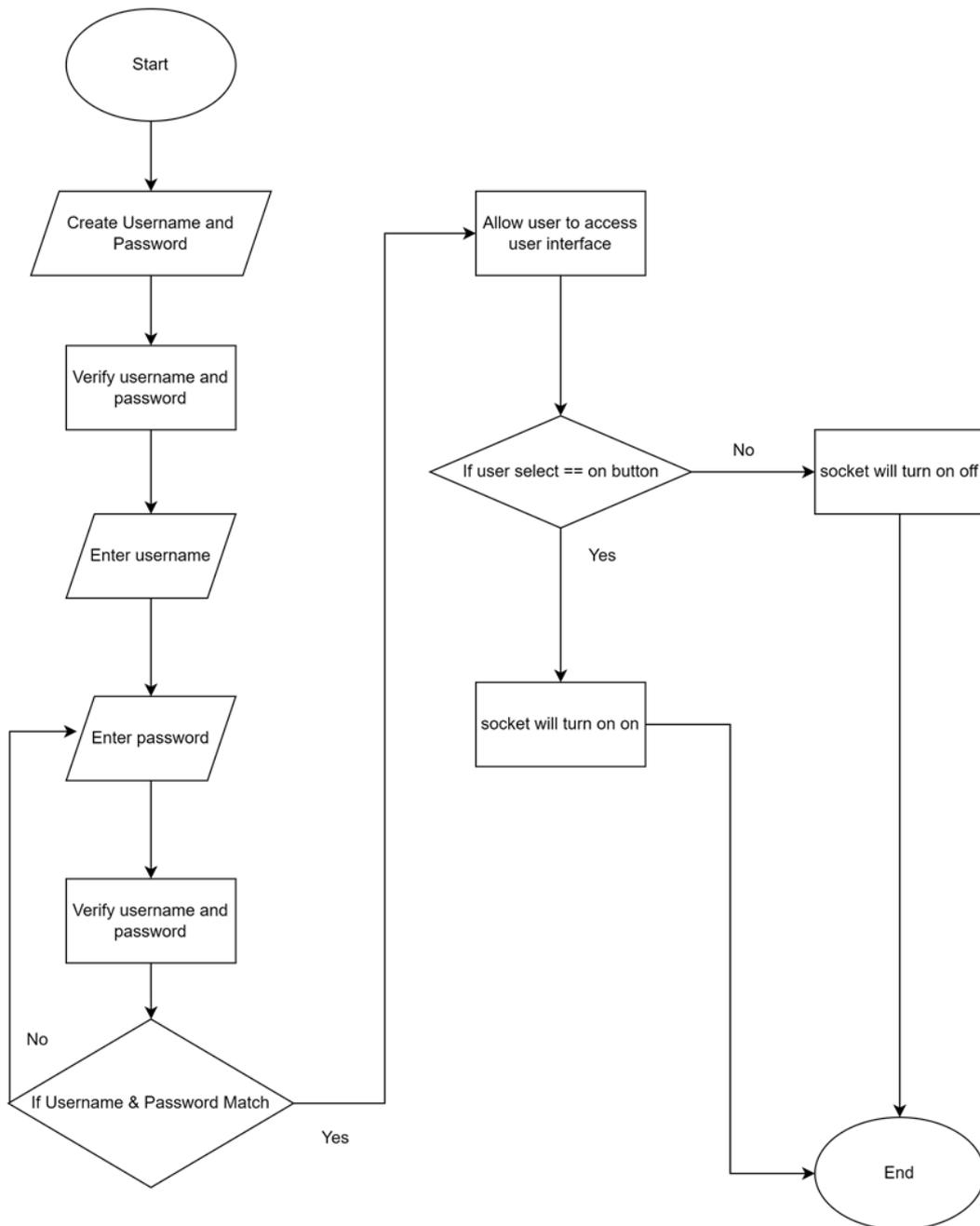


Figure 3.9 : Login Flowchart

### 3.4.4.1 Subprocess Flowchart

The Smartplug Home Automaton System will perform automatic on or off on the plug by setting up time scheduler on the home interface. The user will set the desired time for the plug to be on with the scheduler. If the scheduler is started, then the plug will be on. If the scheduler time is ended, the plug will be off. The scheduler also will allow user to set automatically on or off at different date or time. The ESP8266 module will send signal the to the AC Relay. The Figure 3.10 are the sub-process for the automation system.

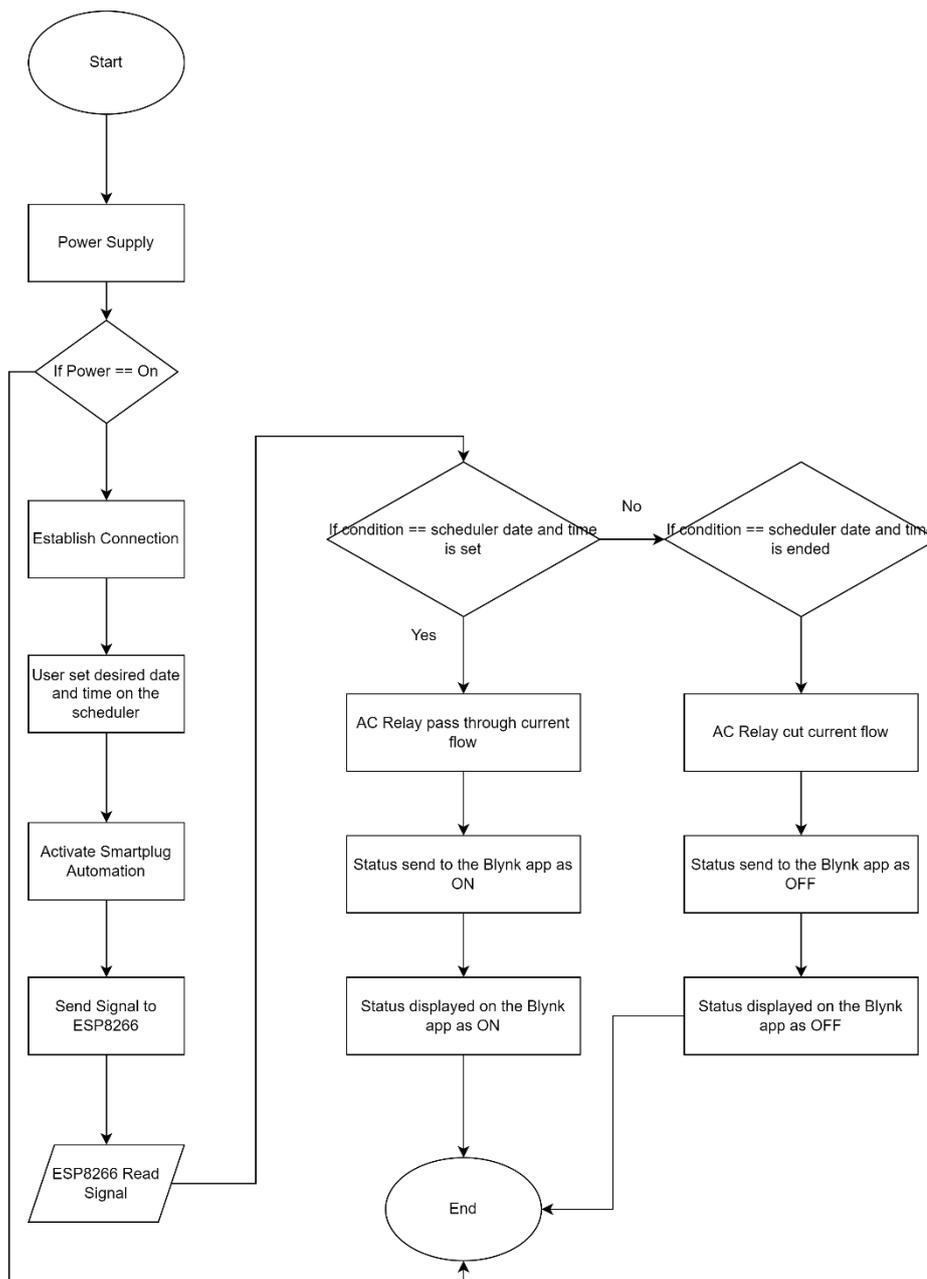


Figure 3.10 : Scheduler Automation Flowchart

### 3.4.5 Context Diagram

A context diagram is a high-level visual representation of a system that shows the relationships between that system and its environment. It typically includes the system itself, along with the external entities that interact with it, such as people, other systems, or physical devices. The diagram is often used to provide a broad overview of a system and its components, and can help to identify the inputs, outputs, and processes involved in the system's operation. Based on the Figure 3.11, the user and the Arduino ESP8266 Module are examples of external entities in this system. The user will request real time data status and the system will send the status of the Smartplug displayed on the homepage of the system. Arduino ESP8266 Module will send status, display status, control and automate the Smartplug.

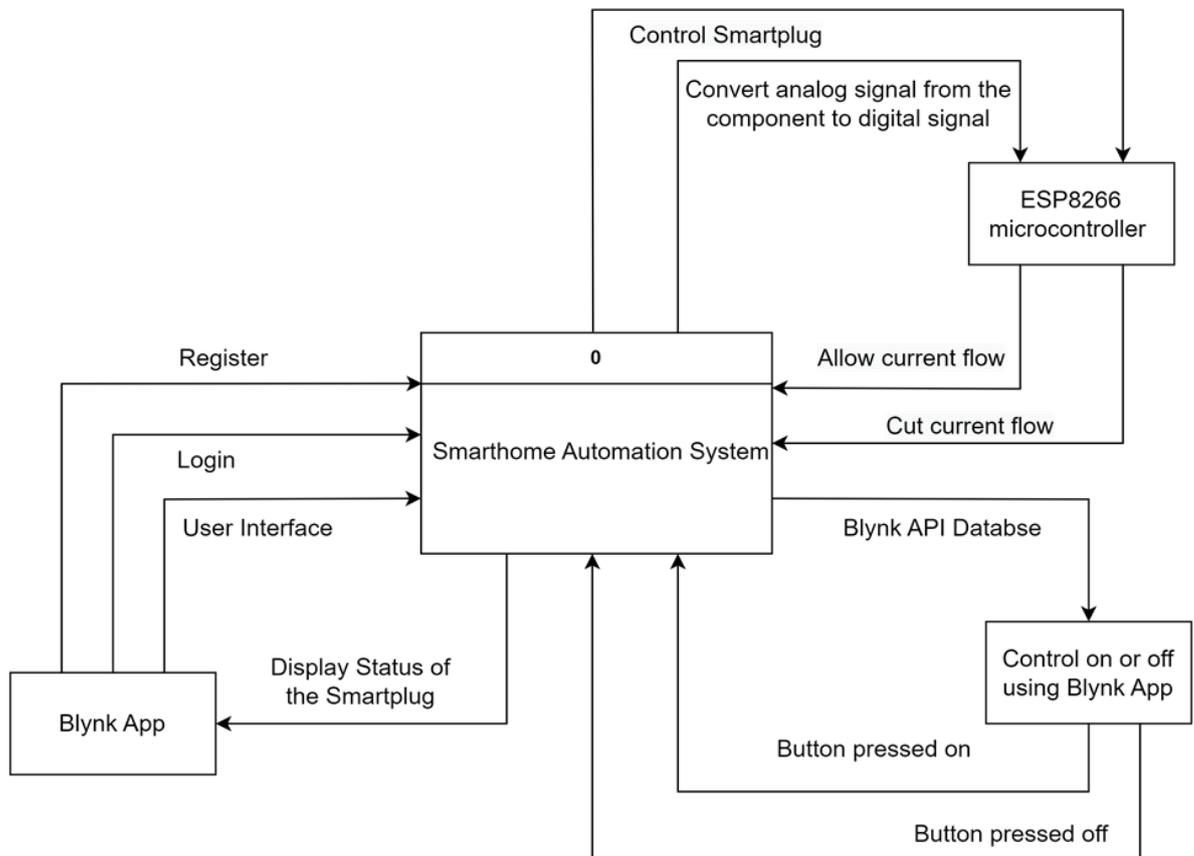


Figure 3.11 : Context Diagram

### 3.4.6 Use Case Diagram

The use case diagram is a visual illustration of the system's behavior where the relationships between the system, actors, and the use case are depicted in Figure 3.12. In use case describes the activities which the actors could operate through the system. In this system, the user will be able to control the Smartplug Home Automation System by using the Blynk App. The user also can view the status of the Smartplug. Moreover, Admin or Developer will be able to add new features and create the demo account for the user to use.

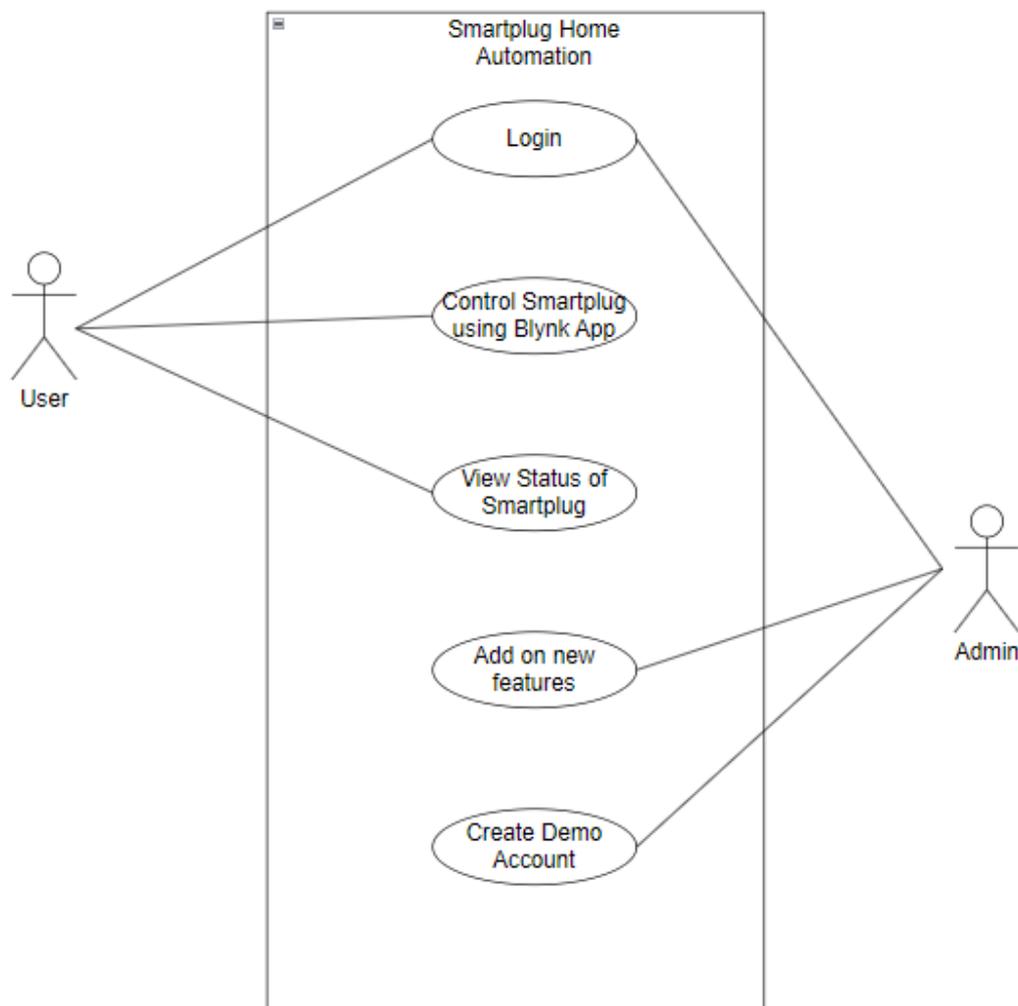


Figure 3.12 : Use Case Diagram

### 3.4.7 Activity Diagram

Figure 3.13 Shows hardware activity diagram as part of the Smartplug Home Automation System process, the ESP8266 microcontroller will read signal from the Blynk. If the ESP8266 read signal from the Blynk app it will then send the signal to the relay. The relay will allow the current to pass through or cut the current from reaching the plug. The data is then uploaded to the cloud which is the Blynk cloud. After that the status will be displayed on the button on the Blynk App for the user to monitor.

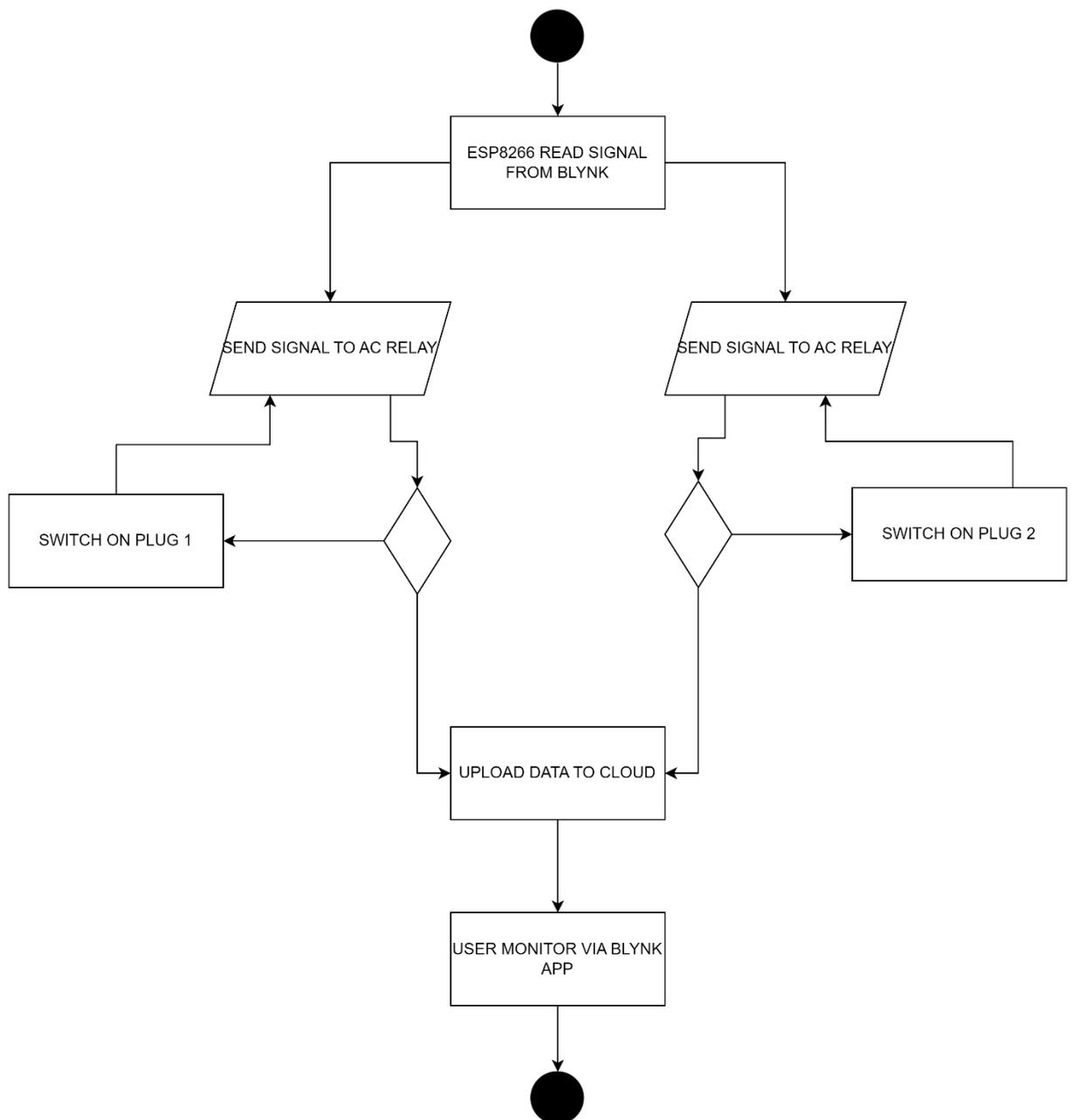


Figure 3.13 : Hardware Activity Diagram

The system's activity diagram is shown in Figure 3.14. User who want to use this isystem will receive a demo account from admin to be able to use the system. User will need to add new device and connect it to the user Wi-Fi in order to connect the system with Blynk Cloud. Admin have extra feautures which is to modified or add any function to the Blynk app. From the Blynk app home interface, user and also adin can control plug switch button and monitor plug status.

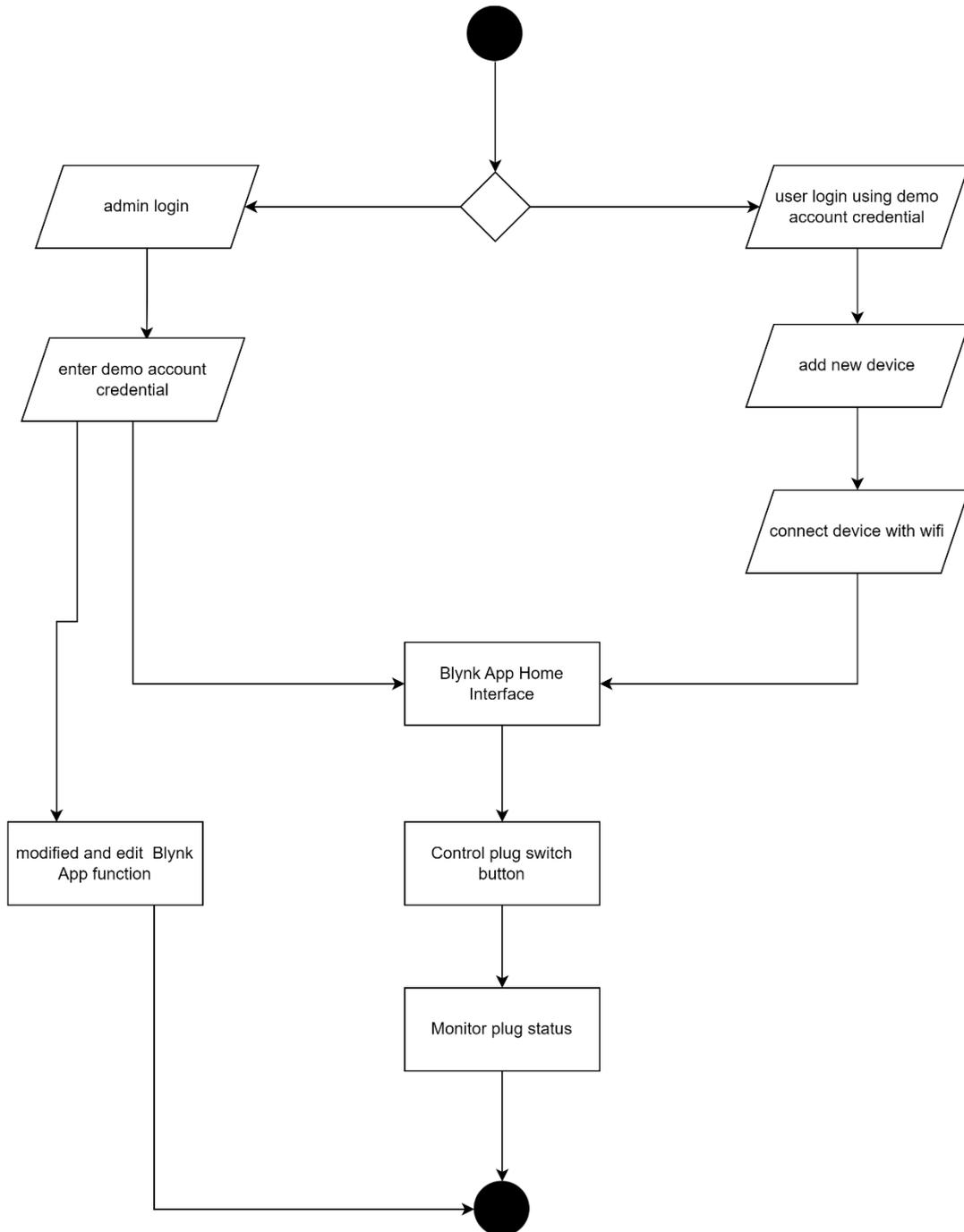


Figure 3.14 : System Activity Diagram

### 3.4.8 Storyboard

Based on the Figure 3.15 it shows the storyboard for Smartplug Home Automation System.

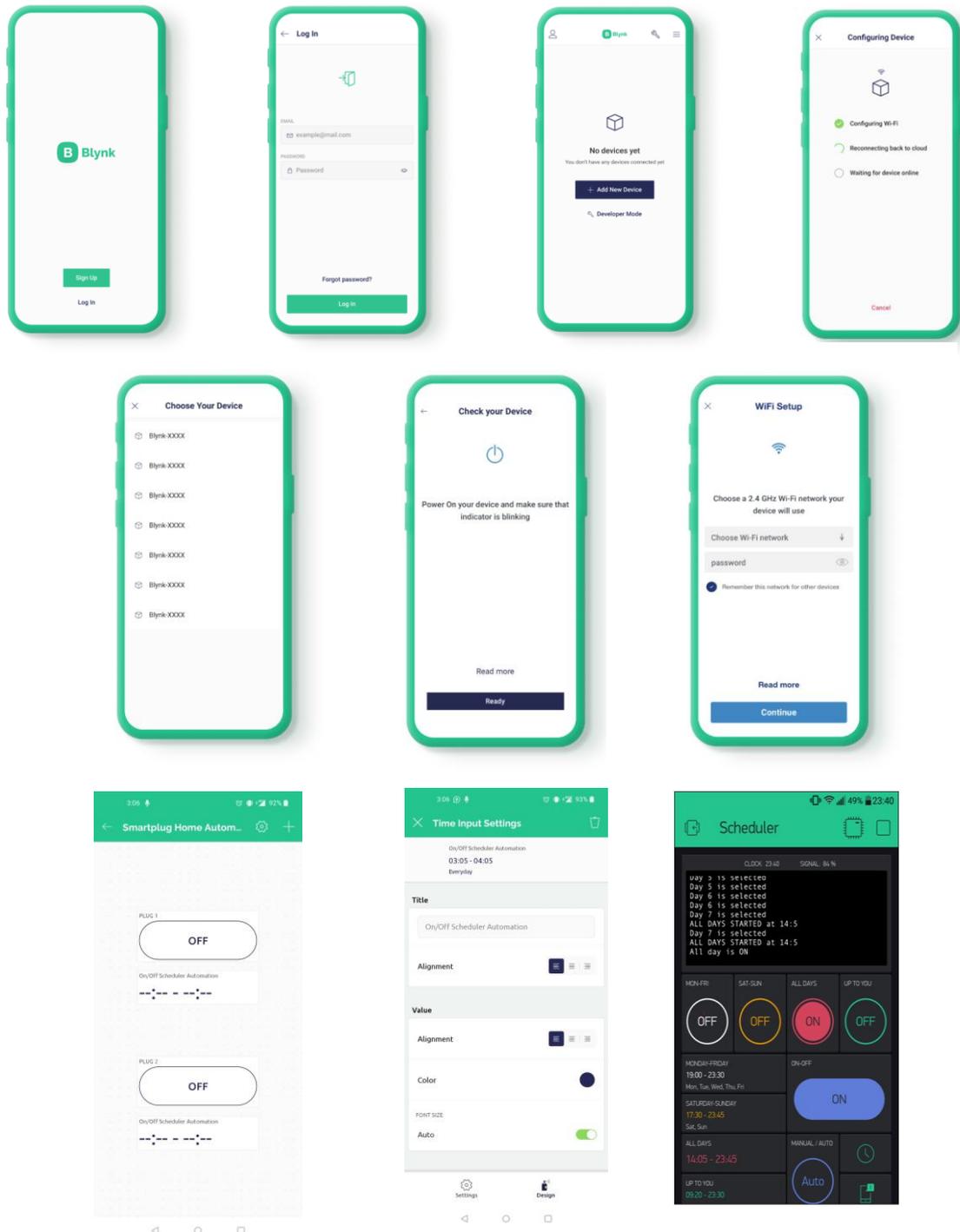


Figure 3.15 : Storyboard

### 3.4.8.1 Login Interface

A login interface is a screen or set of screens that allows a user to enter their login credentials, such as a username and password, to access a system or application. Login interfaces are commonly used for accessing websites, online services, and other applications that require authentication to use them. In some cases, a login interface may also include additional features, such as the ability to reset a forgotten password or create a new account. The user will be provided with Demo Account to login and be able to use the prototype of Smartplug Home Automation System. Based on Figure 3.16 it shows the login interface for the proposed system.

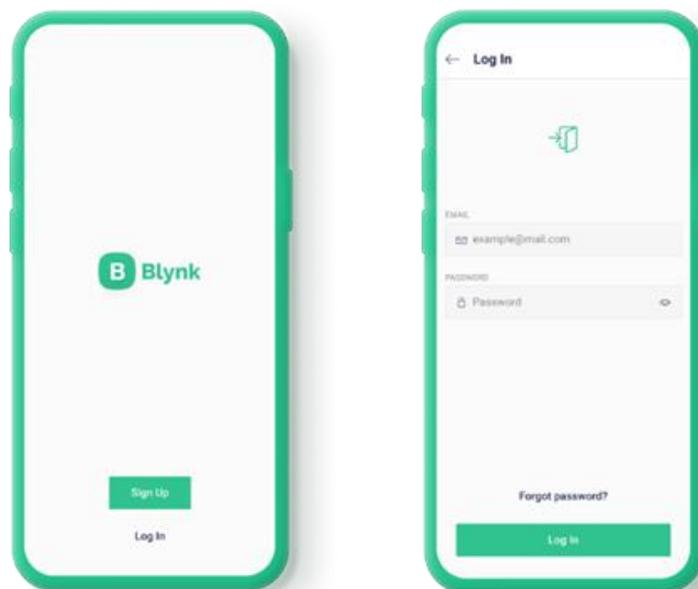


Figure 3.16 : Login Page

### 3.4.8.2 Set Up Device Interface

A set up device interface is a screen or set of screens that allows a user to connect the Smartplug Home Automation System with the Blynk App. The user need to add new device and then connect the Smartplug to the WiFi in order to use all the Smartplug Home Automation System functionality. The design is simple and very straight forward in order to ease user to navigate through the set up. Figure 3.17 shows the process of Set Up Device interface.

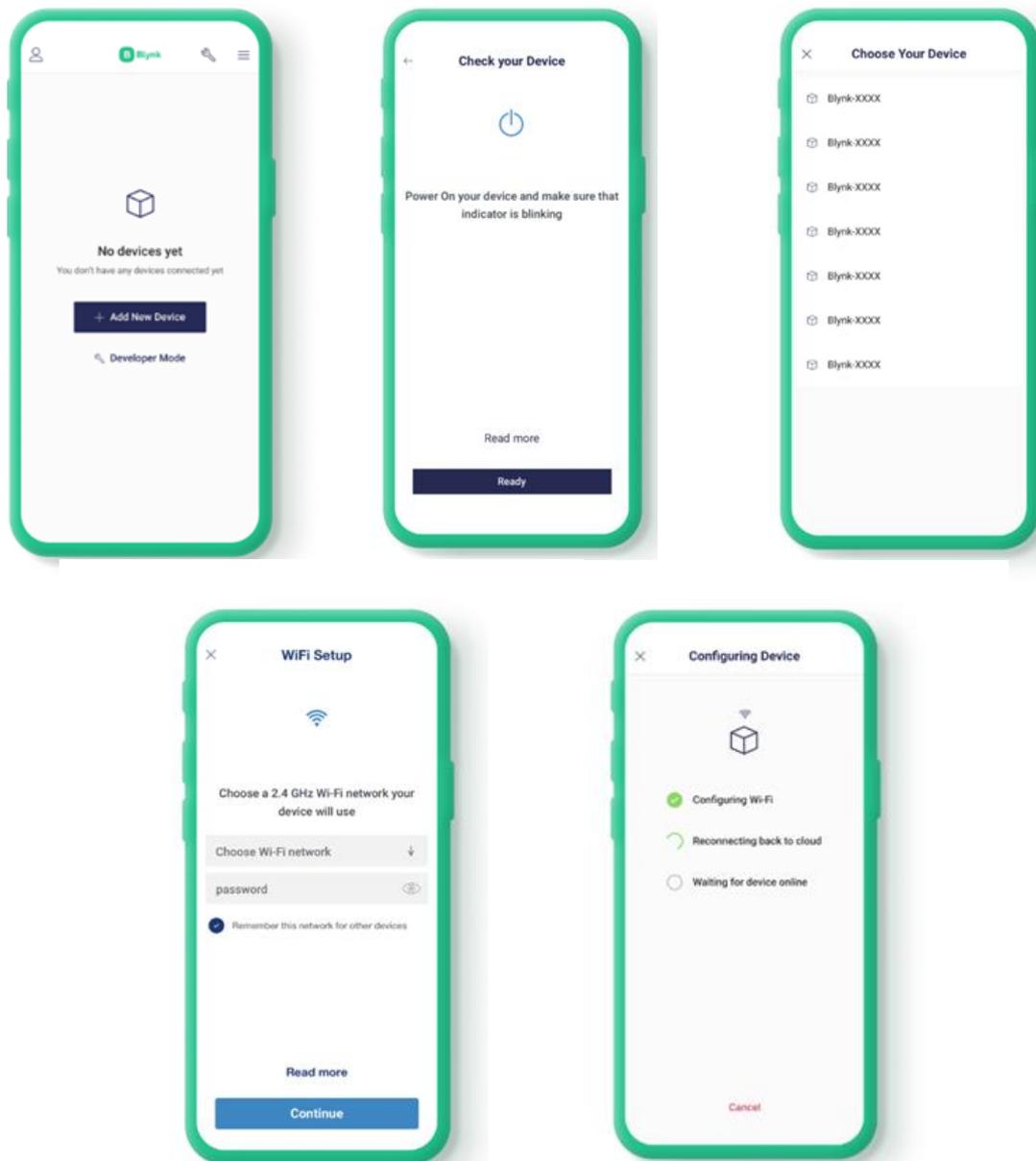


Figure 3.17 : Set Up Device Interface

### 3.4.8.3 Home Interface

A home interface is a screen or page within an application or system that serves as the starting point or central hub for accessing the various features and functions of the application. The home interface often provides an overview of the user's current activity or status within the application, and may include links, buttons, or other controls for accessing the various features and options available. The design and layout of a home interface can vary depending on the specific application or system, but they are commonly designed to be easy to navigate and use. Based on the Figure 3.18, it shows the Home Interface which consist of the on off plug functionality and also time scheduler for the Smartplug Home Automation System. When user tap on the on or off button on the interface, the plug of the Smartplug will also be turned on or off. User will also be able to automate the Smartplug by set a desired time on the scheduler to set the Smartplug on or off at the desired time. The Smartplug will automatically turned on and of based on the time settings in the scheduler.

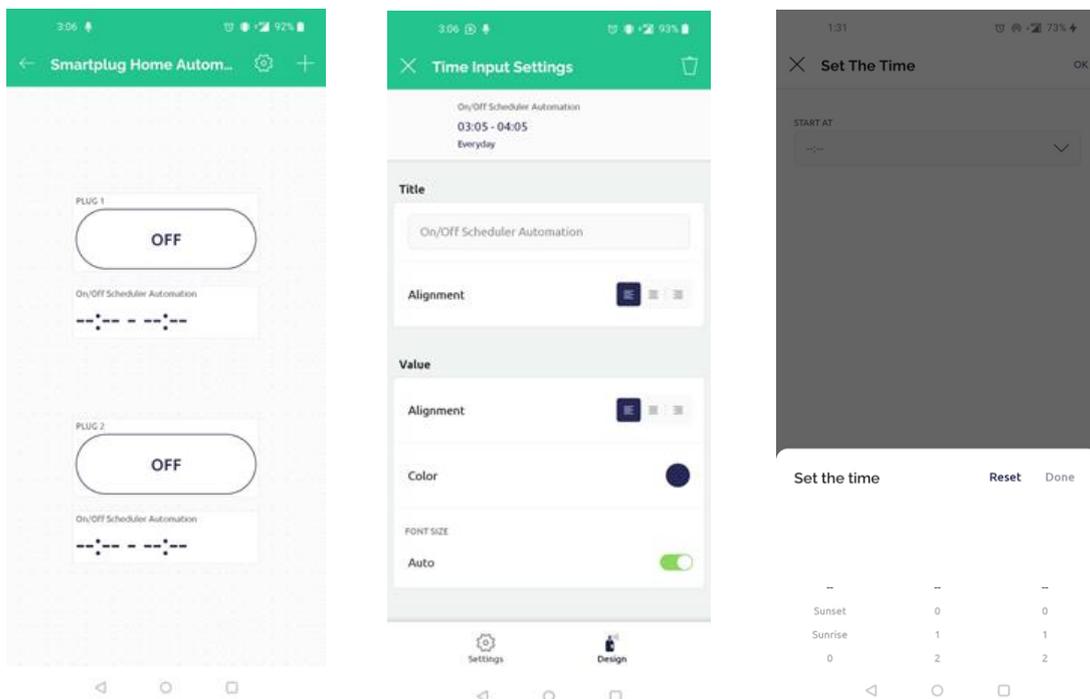


Figure 3.18 : Home Interface

### 3.5 Data Design

#### 3.5.1 ERD

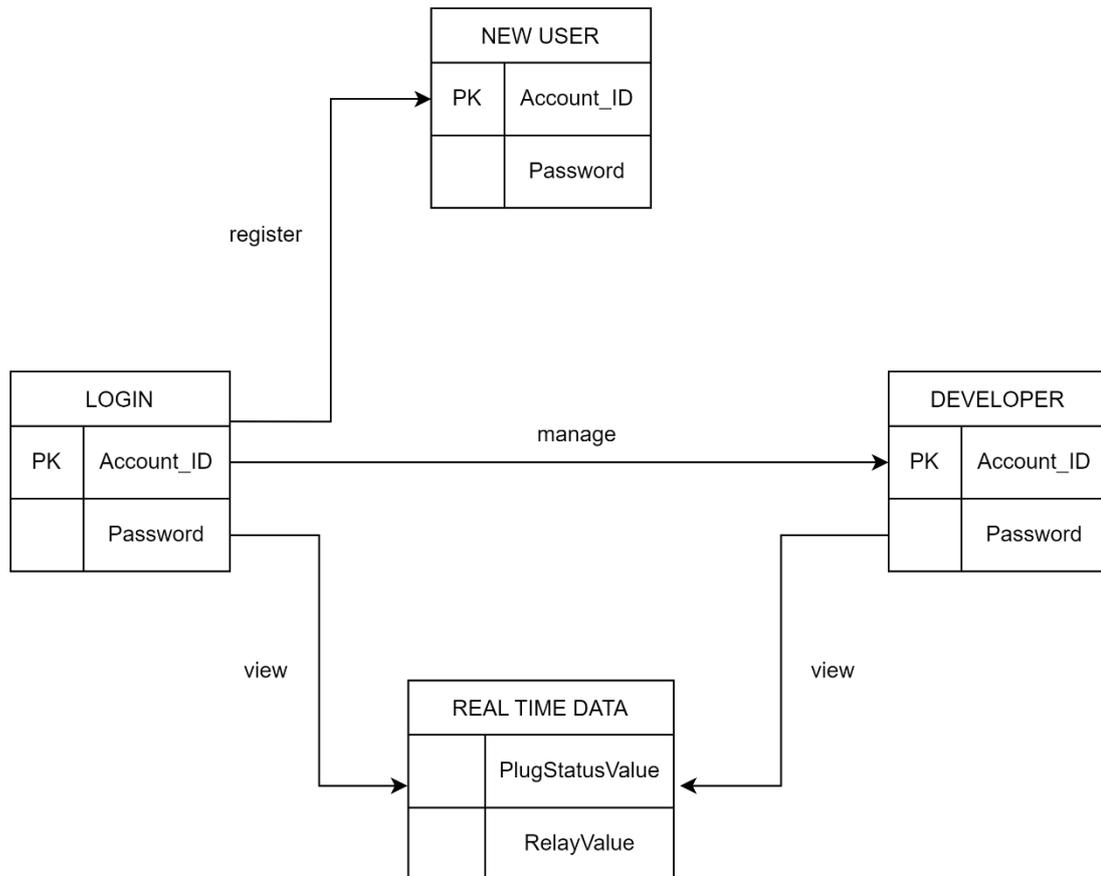


Figure 3.19 : ERD Diagram

### 3.5.2 Database Dictionary

#### 3.5.2.1 Login

**Table 5 : Login Database Dictionary**

Attribute	Data Type	Length	Description	Key
Account_ID	int	11	Account_ID	(PK)
Username	nvarchar	30	Username	
Password	nvarchar	-	Password	
FirstName	nvarchar	50	First_Name	
LastName	nvarchar	50	Last_Name	

#### 3.5.2.2 Data

**Table 6 : Data Database Dictionary**

Attribute	Data Type	Length	Description	Key
Data_ID	int	11	Data_ID	(PK)
plug_status	nvarchar	20	Plug_status	
plug_on	nvarchar	20	Plug_on	
plug_off	nvarchar	20	Plug_off	
dateTime	timestamp		Date_Time	

## 3.6 Hardware and Software

### 3.6.1 Hardware Support

#### 3.6.1.1 ESP8266 Microcontroller

The ESP8266 is a low-cost microcontroller with built-in WiFi capabilities. It is made by Espressif Systems, a Chinese company. It is an incredibly popular platform for IoT projects because it is cheap and easy to use, and it allows for connected devices to be easily controlled and monitored from a remote location. Moreover, ESP8266 is based on the Tensilica Xtensa LX106 processor and has 80 MHz of clock speed. It has 32 kilobytes of instruction RAM, 80 kilobytes of data RAM, and 4 megabytes of flash memory. The chip also includes a 10-bit ADC, which can be used to read analog sensors, and a variety of digital interfaces such as SPI, I2C, UART, and I2S. Other than that, the ESP8266 can be programmed using the Arduino IDE, which makes it easy to get started with the platform. There are also a variety of third-party libraries and frameworks available for the ESP8266 that make it easy to interface with different types of sensors and devices. It's main use case is being a microcontroller with built-in WiFi that could connect your DIY project to internet or to be used as standalone device with build-in software and logic. You can also use it as a WiFi module to connect other microcontroller to internet and make IoT projects. It has become a popular choice for IoT and home automation projects because of its low cost and ease of programming. Additionally, the ESP8266 is a very power-efficient device, making it well suited for battery-powered projects. Figure 3.20 below shows the components of ESP8266 Microcontroller.

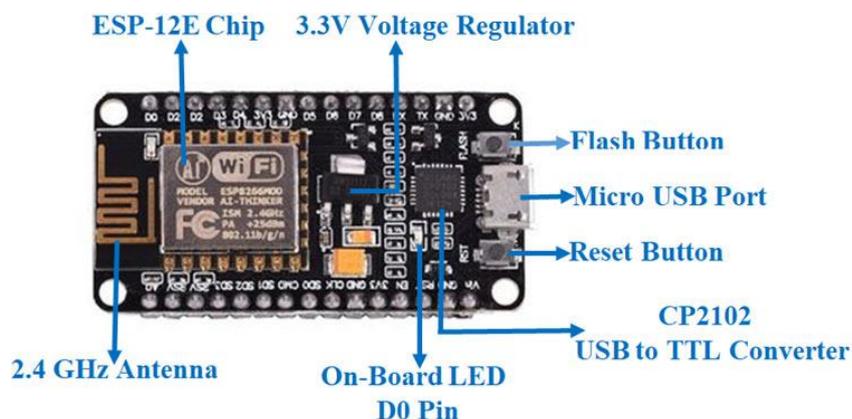


Figure 3.20 : ESP8266 Microcontroller

### 3.6.1.2 2 Channel AC Relay

A 2-channel AC relay is an electronic device that is used to control the flow of electricity in an AC circuit. It consists of two separate relay channels, each of which can be used to switch an AC load on or off. Moreover, each channel of the relay has a set of input terminals, which are used to control the state of the relay. The input terminals are typically connected to a control circuit, such as a microcontroller or a switch. When a voltage is applied to the input terminals, the relay switches on, closing the contacts of the relay and allowing electricity to flow through the load. When the voltage is removed from the input terminals, the relay switches off, opening the contacts and breaking the circuit. Other than that, the 2-channel AC relay typically uses electromagnet as the switching mechanism, which is designed to handle the AC voltage and current. It could be used in various application such as home automation, industrial process, and control, lighting and appliances control and etc. For safety purpose, it's important to ensure the relay rated voltage and current is higher than the load. It also important to use proper technique when handling it, such as proper wiring and grounding, because if it was used improperly it could cause damage to the equipment or even fire hazard. Figure 3.21 below shows the components of 2 Channel AC Relay.

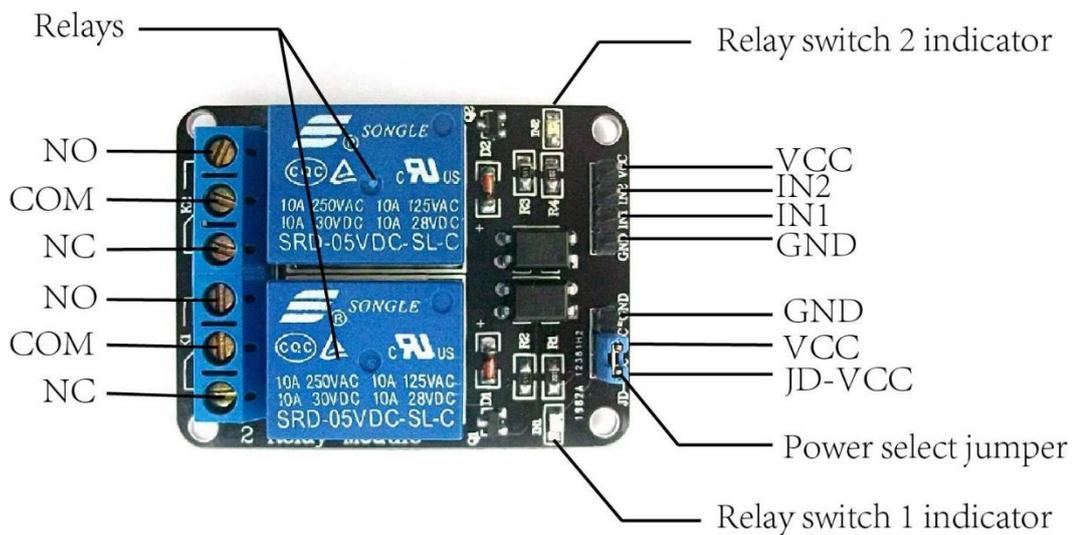


Figure 3.21 : 2 Channel AC Relay

### 3.6.1.3 Base Board for NodeMCU v3

The NodeMCU Base Board is an expansion board designed to work with the NodeMCU V3 ESP8266 ESP-12E. It is powered by a 6V to 24V DC input and extends the GPIO of the NodeMCU to header pins. It also defines standard interfaces to interact with sensors or other boards and features the Lua scripting language. The board features a power jack, a power switch, a reset button, a status LED, a power LED, a connector for a micro SD card, and a USB port. The power jack and switch allow the user to control the power supply to the NodeMCU. The voltage divider circuit ensures that the voltage level of the I/O pins is within the acceptable range. The reset button, status LED, and power LED allow for easy debugging. The micro SD card connector can be used to store files or as an additional memory. The USB port can be used to program the NodeMCU. All these features make the NodeMCU Base Board a great platform for prototyping and building Internet of Things applications. Figure 3.22 below shows the components of Base Board for NodeMCU v3.

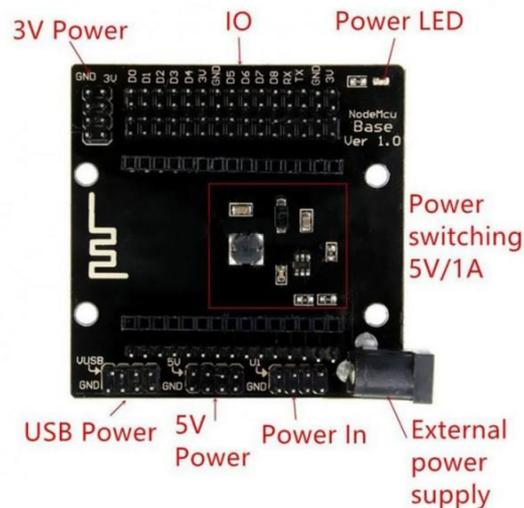


Figure 3.22 : Base Board for NodeMCU v3

### 3.6.1.4 DC Male Power Jack Connector To DG126 Converter

The DC Male Power Jack Connector To DG126 Converter is a great fit for Arduino power jack. It has a current rating of 1A, body orientation of straight, and voltage rating of 16V. It has a mounting type of cable mount and can be used as a voltage input for regulated external power supplies that do not use a barrel jack connector. It also works as a DC voltage input for the onboard power supply for Arduino boards. The adapter allows you to connect a barrel jack connector to bare wires with one end having screw terminals and the other having a 5.5 mm \* 2.1 mm, center-positive connector. The connector is also compatible with the female barrel power input found on many Arduino boards and has an outer diameter of metal plug of 0.216in (5.5mm) and an inner diameter of metal plug of 0.083in (2.1mm). This DC Male Power Jack Connector To DG126 Converter is an excellent fit for Arduino power jack and can be used to power your Arduino board. Figure 3.23 below shows DC Male Power Jack Connector To DG126 Converter.



Figure 3.23 : DC Male Power Jack Connector To DG126 Converter

### 3.6.1.5 Extension Plug Socket

Extension plug socket is an electrical device used to connect an electrical appliance to an outlet. It works by connecting the appliance's power cord to the plug and then connecting the plug to the outlet. Extension plug sockets are usually made of plastic or metal and have a live part, a neutral part, and an earth part. The live part typically has two pins and is used to connect the hot wire from the appliance to the live wire from the outlet. The neutral part also has two pins and is used to connect the neutral wire from the appliance to the neutral wire from the outlet. The earth part consists of a single pin, which is used to connect the earth wire from the appliance to the earth wire from the outlet. The Extension Plug Socket is controlled by AC Relay that is connected to the ESP8266. Figure 3.24 below shows Extension Plug Socket before been modified to integrate with Smartplug Home Automation System.



Figure 3.24 : Extension Plug Socket

Table 7 : Hardware Support

Hardware	Specification	Purpose
Laptop	Brand: Lenovo  Processor: Intel Core i7  Windows 10 OS  RAM: 32GB  System Type: 64-bit OS	A device that needed to develop the system, it is the medium for code writing, software installation and system designing.
ESP8266 Microcontroller	Processor: L106 32-bit RISC microprocessor core running at 80 or 160 MHz On board 3.3V voltage regulator  IEEE 802.11 b/g/n Wi-Fi  Support Wi-Fi Direct (P2P), soft-AP	It acts as a microcontroller between AC Relay and Extension Plug Socket
2 Channel AC Relay	Input Rating. Supply voltage (DC- & DC+): 5V. Input switch voltage: 4 to 5V. Input release voltage: 0 to 1V.  Output Rating. Rated load voltage: 100V to 240V AC. Load current: 0.10 to 2A. Surge current: 30 A (60 Hz, 1 cycle)	Control AC Load on or off
Base Board for NodeMCU v3	Input Voltage: 6V – 24V DC.  PCB Thickness: 1.6mm.  NodeMCU Pin Distance between sides: ~28.0mm.  Pin Pitch: 2.54mm (0.1in)	Expansion board designed to work with the NodeMCU V3 ESP8266 ESP-12E. The voltage divider circuit ensures that the voltage level of the I/O pins is within the acceptable range

<p>DC Male Power Jack Connector To DG126 Converter</p>	<p>DC Jack(Male) to DG126 Converter.</p> <p>Fit for Arduino Power Jack.</p> <p>Current rating: 1A.</p> <p>Body Orientation: Straight.</p> <p>Voltage rating: 16V.</p>	<p>Works as a DC voltage input for the onboard power supply for Arduino boards</p>
<p>Extension Plug Socket</p>	<p>Rated Current : 13A / AC 250v - 50Hz.</p> <p>Rated Voltage: Support up to 1800w Total MAX.</p> <p>13amp Plug Top with Sirim approved</p> <p>Cable size : 40/016 x3c Wire</p>	<p>Connecting the appliance's power cord to the plug and then connecting the plug to the outlet. Also connected to ESP8266 Microcontroller in order to automate the plug on and off function.</p>

## 3.6.2 Software Support

### 3.6.2.1 Arduino IDE

The Arduino Integrated Development Environment (IDE) is a software application used for writing and uploading code to Arduino boards. It is an open-source application developed by Arduino.cc and supports the C and C++ programming languages. The Arduino IDE is designed for different operating systems, such as Windows, Mac OS X, and Linux. It contains a text editor for writing code, a message area, and a console area. The IDE also includes a toolbar with buttons for compiling and uploading code to the board, a serial monitor to view the board's output, and a menu bar with options such as File, Edit, Sketch, and Tools. The Arduino IDE also includes an Examples section that includes sample code to help users understand the basics of programming an Arduino board. The Arduino IDE is an essential tool for programming Arduino boards, and it is very easy to use. Figure 3.25 shows the Arduino Logo.



Figure 3.25 : Arduino Logo

### 3.6.2.2 Blynk Application

Blynk is a mobile application development platform designed to enable users to quickly and easily create mobile apps that can interact with hardware devices such as Arduino, Raspberry Pi, and other microcontrollers. The Blynk app provides a drag-and-drop interface for users to create their own user interfaces, or “Blynks”, with a variety of widgets and visual elements. Users can then connect the mobile app to their Blynk account, which allows them to control their hardware with the mobile app. Blynk also provides users with the ability to store their data in the cloud, as well as access to an analytics dashboard for monitoring their hardware. Blynk is an easy to use and powerful platform for creating mobile applications that interact with hardware devices. Figure 3.26 shows the Blynk logo



Figure 3.26 : Blynk Logo

### 3.7 Interface Design

These are the function on the Blynk Application

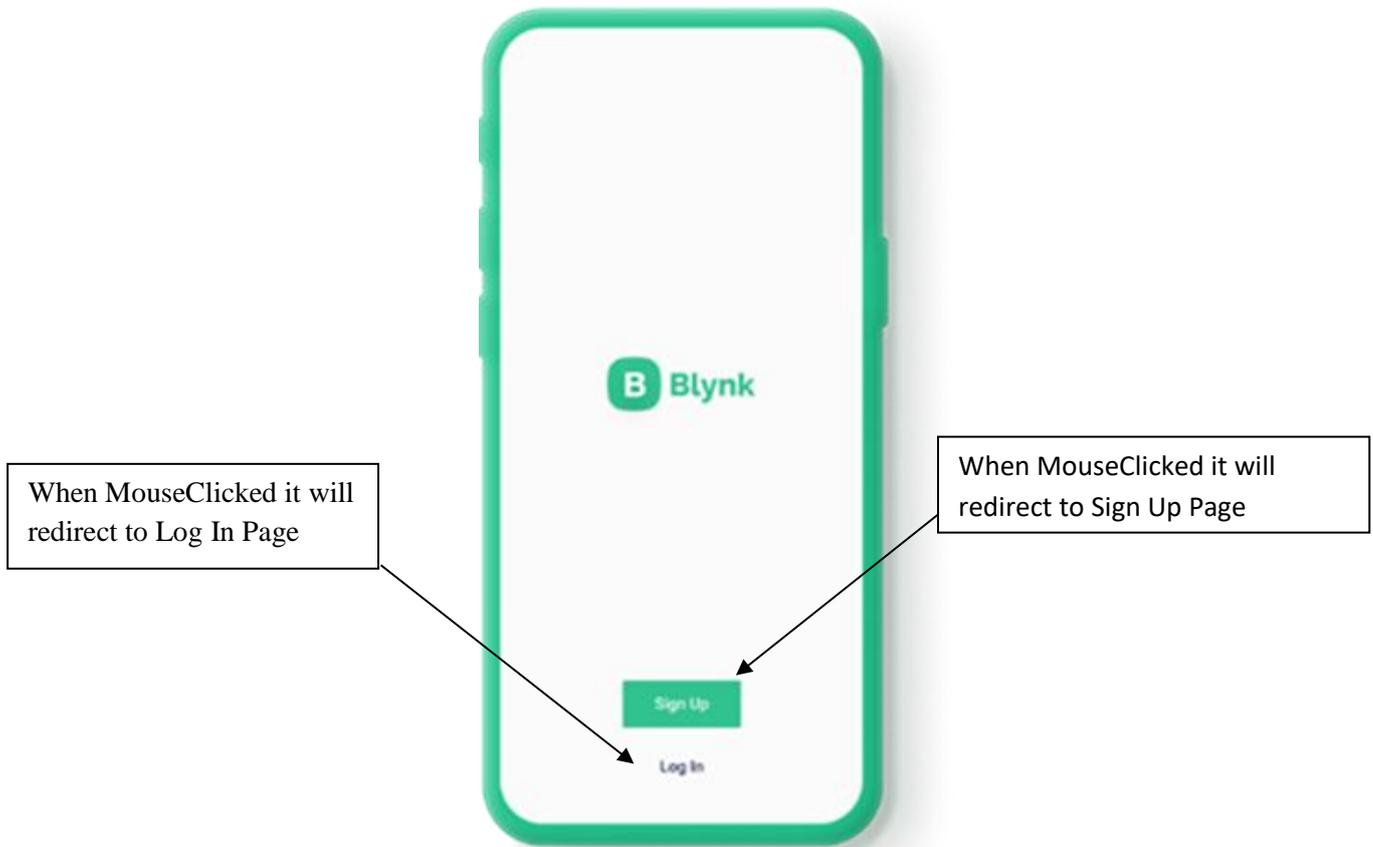
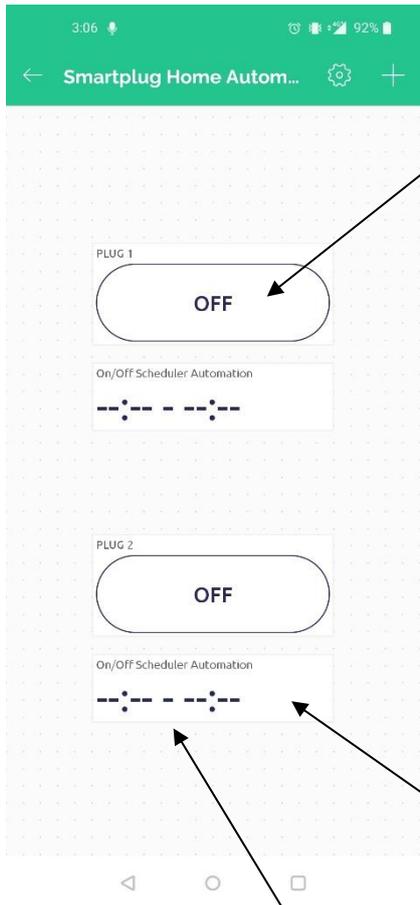


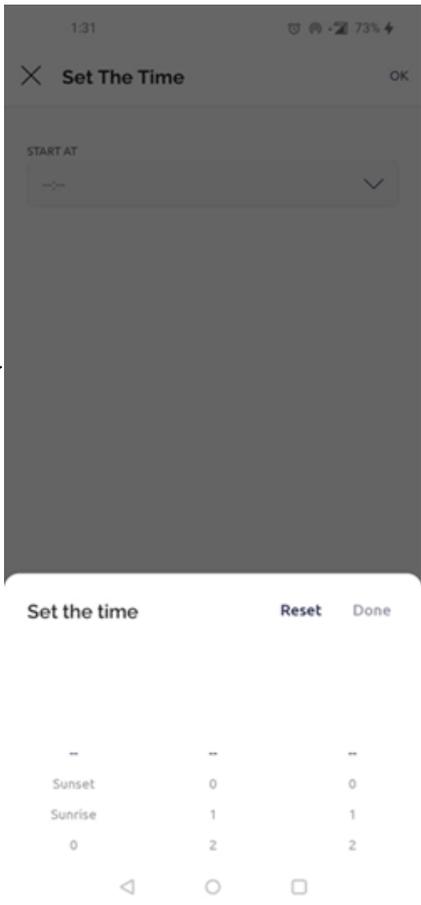
Figure 3.27 : Login Page Button



When MouseClicked it will turn on or off socket on the plug, the text will also will displayed on or off depends on the signal received

At the scheduler page, user will be able to set the time and day of the plug to be automated

When MouseClicked it will redirect to scheduler page to set up scheduled timer for the plug to automatically turned on and off



On/Off Scheduler Automation  
**03:05 - 04:05**  
 Everyday

After the scheduler is set, it will display the time and day on the button

Figure 3.28 : Home Page Button

## 3.8 Proof of Initial Concept

### 3.8.1 Blynk Configuration

Blynk desktop configuration is used to set up the Blynk App to integrate with the ESP8266 Microcontroller. The assigned template ID is used in the code in order to connect the hardware with the Blynk API. Figure 3.29 below shows the configuration setting in the Blynk desktop.

The screenshot displays the Blynk Desktop configuration interface for a template named "SMARTPLUG". The interface is divided into several sections:

- Info Tab:** Contains fields for Template Name (SMARTPLUG), Hardware (ESP8266), Connection Type (WIFI), Description (This is my template), Template ID (TMPLQPR5FAI2), Manufacturer (My organization 5361DU), Offline Ignore Period (00 hrs 00 mins 00 secs), and Hotspot Prefix (Hotspot Prefix).
- Template Image (Optional):** A dashed box containing an "Add image" button and instructions: "Upload from computer or drag-n-drop .png or .jpg, minimum width 500px".
- Firmware Configuration:** A code block showing the following configuration:

```
#define BLYNK_TEMPLATE_ID "TMPLQPR5FAI2"
#define BLYNK_DEVICE_NAME "SMARTPLUG"
```

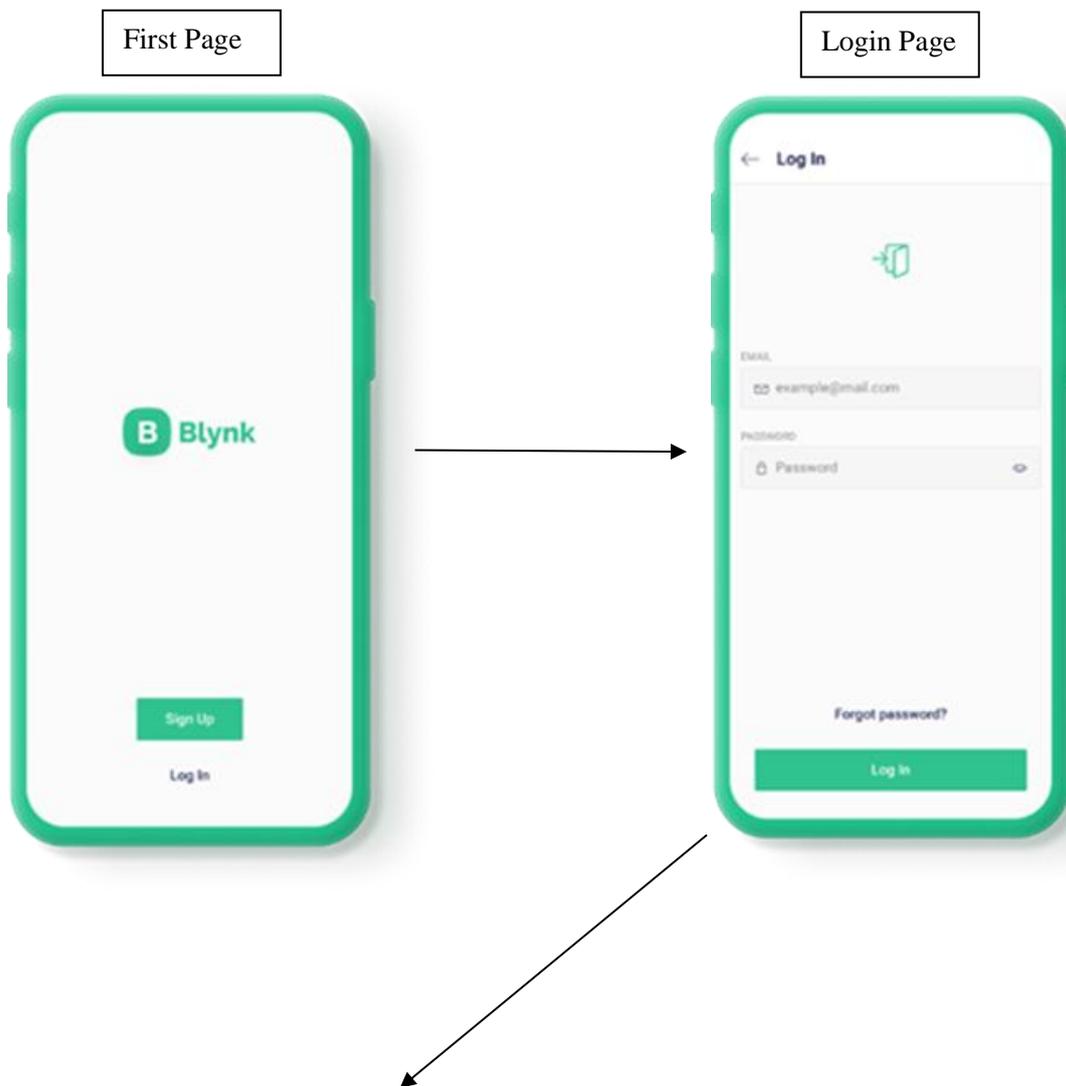
Below the code, a note states: "Template ID and Device Name should be included at the top of your main firmware".
- Buttons:** "Cancel" and "Save And Apply" buttons are visible at the top right.
- Datstreams Tab:** Shows a search bar and a table of 2 datstreams.

Id	Name	Alias	Color	Pin	Data Type	Units	Is Raw	Min	Max	Decimals	Default Value	Actions
1	RELAY_1	RELAY 1	Blue	V0	Integer		false	0	1	-	0	
2	RELAY_2	RELAY 2	Green	V1	Integer		false	0	1	-	0	

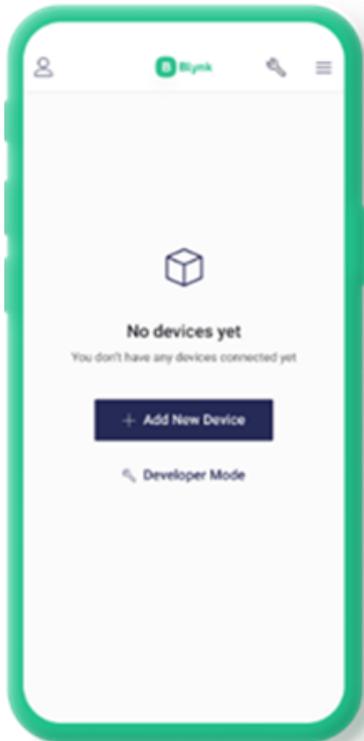
Figure 3.29 : Blynk Desktop Configuration

### 3.8.2 Blynk App Interface

In order for the user to use the Smartplug Home Automation System, the user need to install Blynk App from the Play Store or Appstore. After that, user need to login with the given demo account by the developer. Moreover, user will need to connect to the Wi-Fi in order to connect the hardware to the Blynk Server. Figure 3.30 below shows the process that user need to follow in order to use the Blynk App.



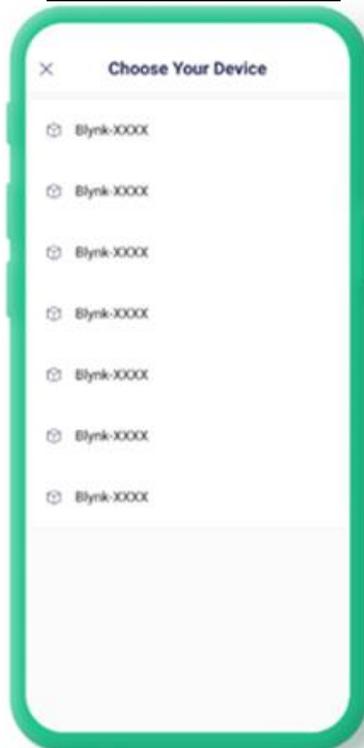
Add New Device Page



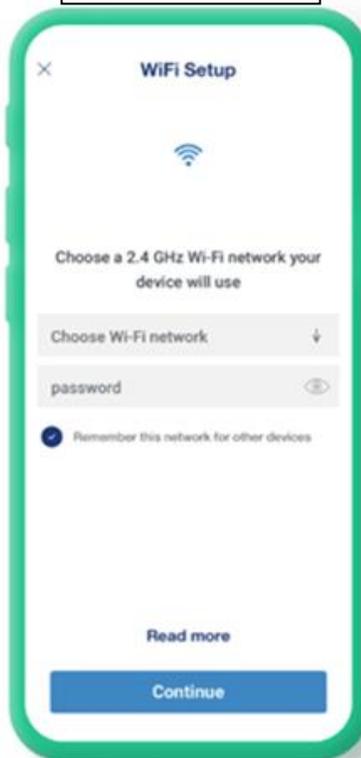
Device Ready Page



Select Device Page



Wifi Setup Page



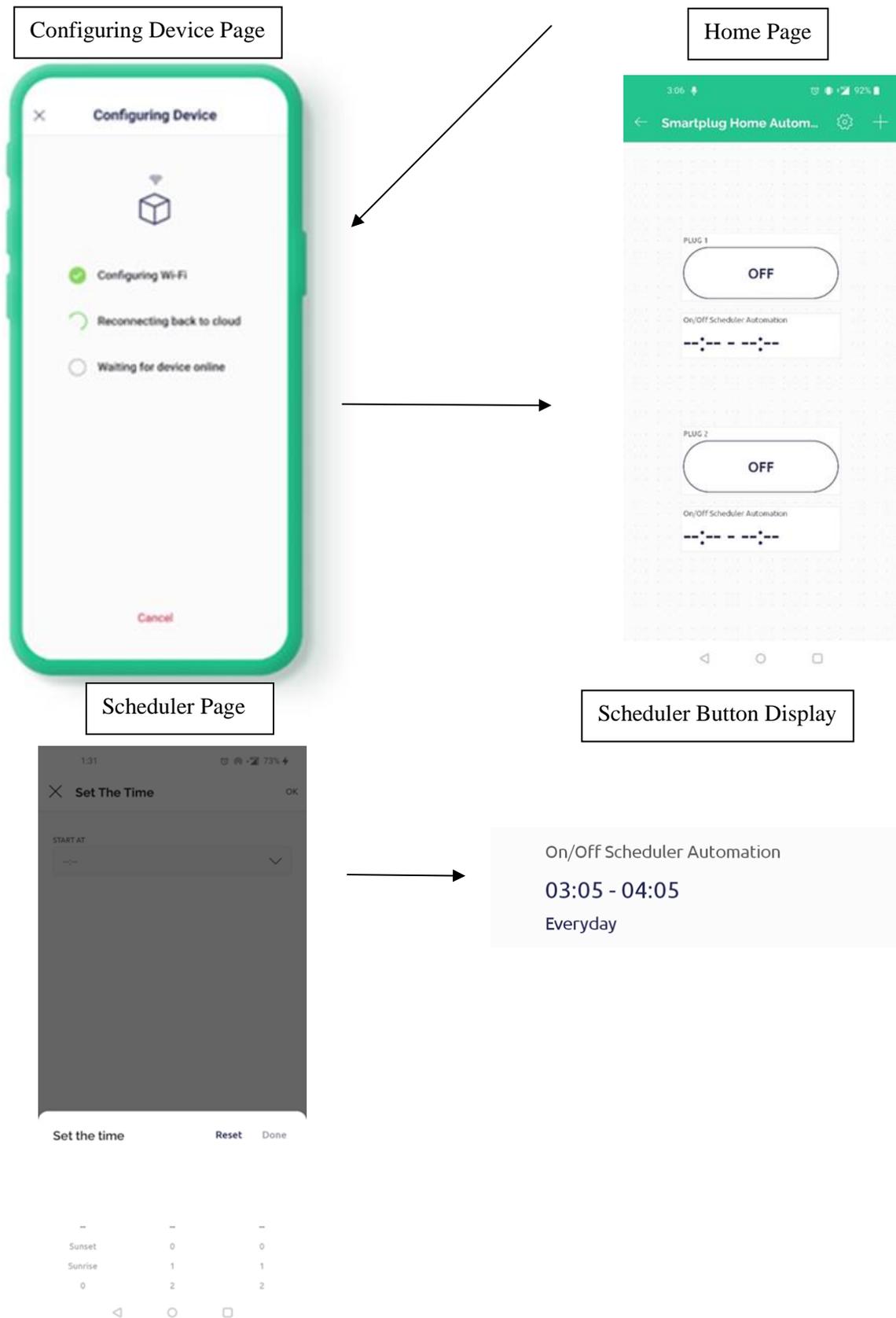


Figure 3.30 : Process that user need to follow in order to use the Blynk App.

### 3.8.3 Hardware Setup

All hardware was assembled accordingly as described in Chapter 3.6.3 and all the hardware decisions were used to complete the design. The AC Relay is located inside the Smartplug socket together with AC to DC converter. ESP8266 Microcontroller will send signal to the AC Relay to control the current flow as follows the user interaction in the Blynk App. Figure 3.31 below shows the hardware setup dor the Smartplug Home Automation System

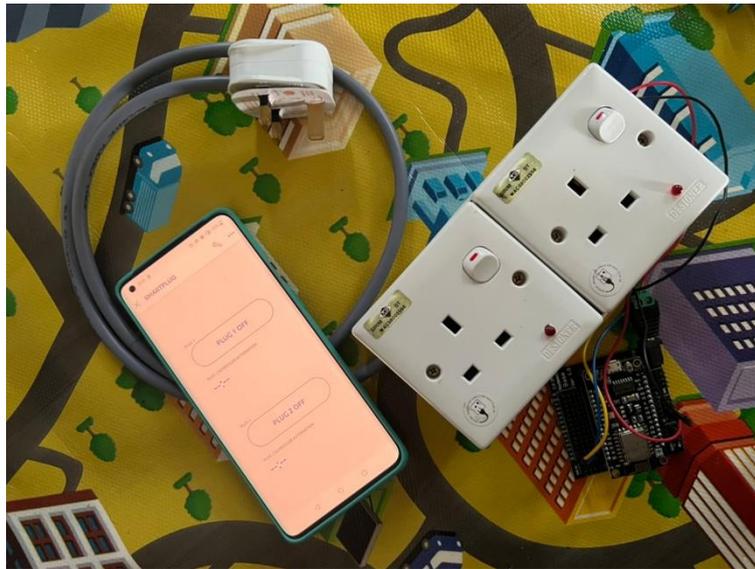


Figure 3.31 : Hardware setup for Smartplug Home Automation System

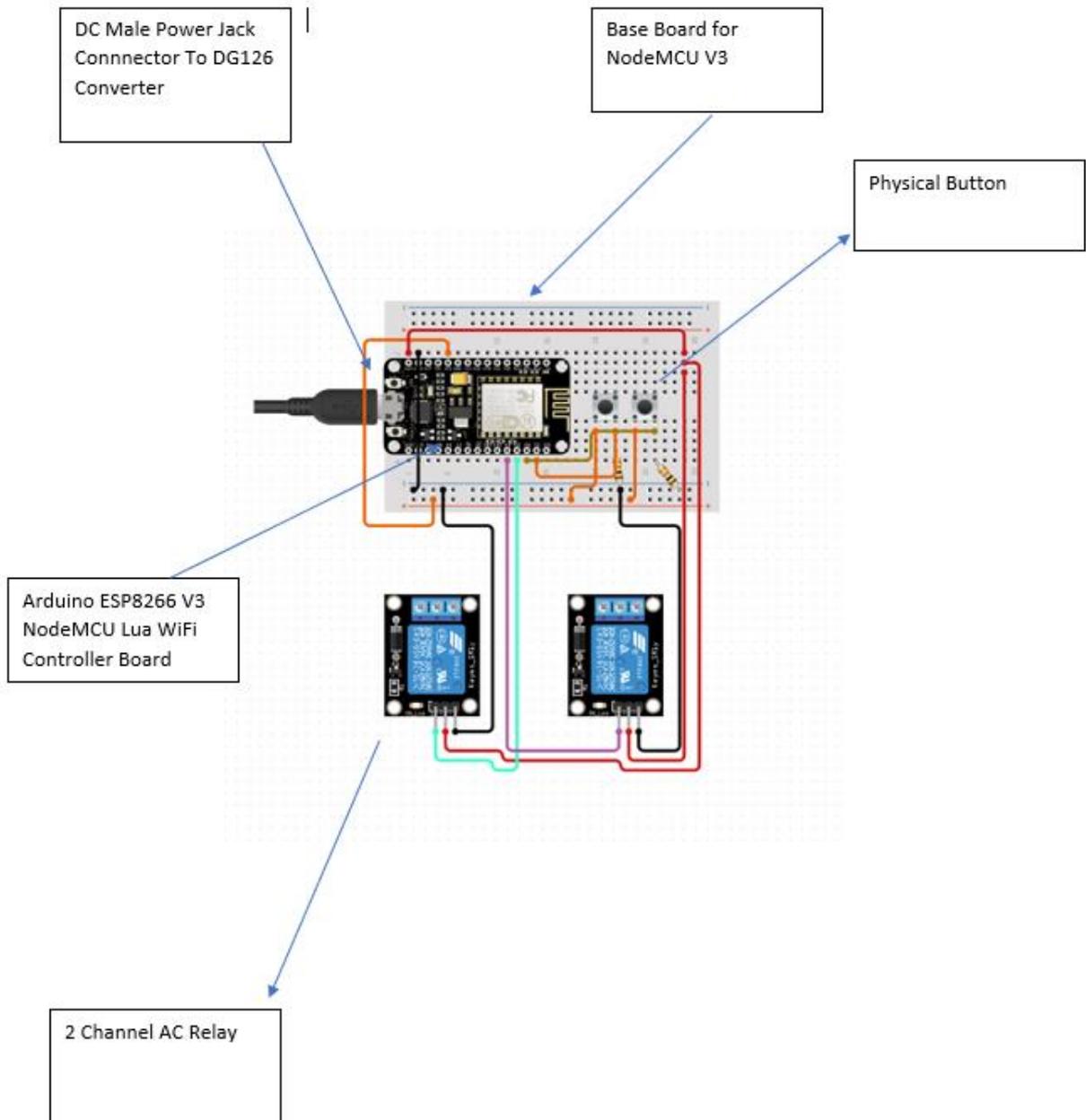


Figure 3.32 : Circuit diagram and label for ESP8266 Microcontroller



Figure 3.33 : Electrical Appliances than can be used on the Smartplug Home Automation

### 3.9 Testing/Validation Plan

#### 3.9.1 Blynk Application Testing Plan

Table 8 : Blynk Application Testing Plan

Test Case	Test Data	Expected Result	Actual Result	Pass / Fail	Comment
Login with correct email and password	Email: <a href="mailto:adamazmi.work@gmail.com">adamazmi.work@gmail.com</a> Password: 654321	Login successful and redirect to homepage			
Login with incorrect username and password	Email: <a href="mailto:aimansafwan98@gmail.com">aimansafwan98@gmail.com</a> Password: 123456	Login unsuccessful and error message pops-up and returns login page			
When button on or off for Plug 1 & 2 tab is clicked	ON/OFF	The plug should pass through or cut current flow from the power source to the electrical appliances connected			
Set the automation with the Scheduler	Time & Day	The plug should turn on and off automatically based on the schedule set			

### 3.9.2 Sensors, Hardware, and Database Connection Testing Plan

Table 9 : Sensors, Hardware, and Database Connection Testing Plan

Test Case	Test Data	Expected Result	Actual Result	Pass / Fail	Comment
When power On	Null	Proposed system is active and online on the Blynk App			
When switch of the Smartplug is turned off	Null	AC Relay cut current from flow			
When switch of the Smartplug is turned on	Null	AC Relay allow current flow			
ESP8266 Microcontroller when power is On	Null	LED light is on			
Database when power On	Null	Connection is successful with ESP8266 Microcontroller			

### 3.9.3 Scheduler Automation Testing Plan

Table 10 : Scheduler Automation Testing Plan

Test Case	Test Data	Expected Result	Actual Result	Pass / Fail	Comment
When the scheduler button is clicked	Null	Redirect to the scheduler page			
When set time button is clicked	Null	List of time option will be available to be choose			
When set day is clicked	Null	List of day option will be available to be choose			
When set button is clicked	Null	The time and day set will be displayed on the scheduler button			

### **3.10 Potential Use of Proposed Solution**

Smartplug Home Automation System is a type of automated home automation system that uses Internet of Things (IoT) technology to control and monitor various type of electrical appliances that can be connected to the Smartplug. Some smartplug may also have features such as control on or off by using mobile application. In this case, Smartplug Home Automation has an advantage in term of automation which is the scheduler function that allow user to schedule time for the smartplug to be on or off at any desired time automatically. Smartplug Home Automation System can be used in a variety of settings, including residential houses, workplace, cafeteria and even lab controlled environment testing. This Automation System can be useful especially for people who are leaving their house for vacation and set lamp automation for example to prevent thieves to enter or rob the house because if the lamp is on and off at scheduled time it will pretend like there is someone still in the house.

Some potential uses for an IoT Smartplug Home Automation System include :

- Residential houses : A Smartplug Home Automation System can help owner to control electrical appliances at home. People tend to forget to turn off electrical appliances when they go out from house, by using the Smartplug, user will able to monitor the status and control the Smartplug from anywhere with internet connection.
- Office workplace : Smartplug Home Automation System may help in saving electric current by set a scheduler timer for the office aircond to be automatically on and off only at office hour.
- Lab controlled environment testing : Research can be easier by using Smartplug Home Automation System for example to stress test hardware at certain amount of time and get accurate results because the test is start and stop accurately by using the scheduler function.
- Cafeteria : Traditional lamps and fans can be convert into smart lamps and fans and can be controlled remotely by only using mobile application. Hastle free for the cafeteria owner that will not require to physically on or off all the lamps and fans in the cafeteria.

### 3.11 Gantt Chart

The Gantt Chart shows the progress of the project with the timescale. The project will be following the proposed methodology. The Figure 3.34 shows about the Gantt Chart of the Smartplug Home Automation System.

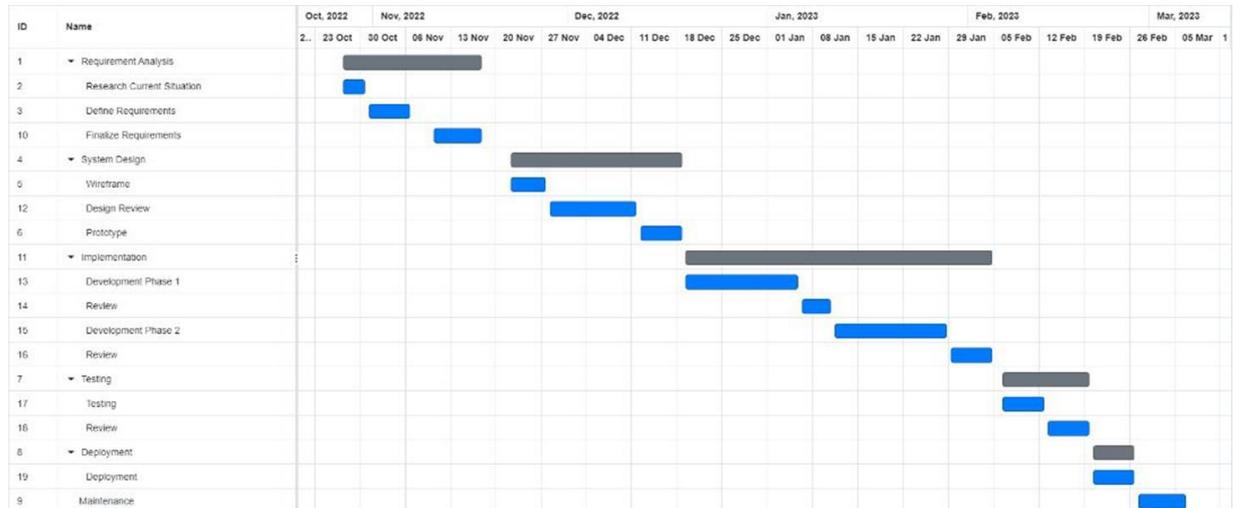


Figure 3.34 : Gantt Chart

## CHAPTER 4

### RESULT AND DISCUSSION

#### 4.1 Introduction

This chapter focuses on the overall process of system implementation, testing, and the results obtained during system implementation for a project. RAD methodology was applied to the system implementation and testing procedure sequence covered in the preceding chapter. This chapter will provide a comprehensive explanation of the system implementation method, the obtained results, and the testing technique used to ensure the project's objectives were met. During the implementation of the project system Arduino programming languages were used to configure hardware and software. This chapter will also include a description of the project's limitations. This project's results and discussion contribute to the planning and direction of future systems.

##### 4.1.1 Development Environment

Arduino IDE and Blynk are the programmes used to implement this system. The Arduino IDE is a programme of software that enables the writing of code, uploading of code, and monitoring of data from GPS sensor, humidity sensor and also gas detection sensor. The Arduino programming language was used to sketch the build code to communicate with the NodeMCU, from GPS sensor, humidity sensor and also gas detection sensor. While, the Blynk software was used in the creation of the mobile application to monitor and control the Smarthome Automation System.

The Smarthome Automation System utilized NodeMCU, where the Arduino IDE was used for writing build code. Arduino is a free and open-source electronics platform with simple hardware and software where it includes a text editor, and output area, a toolbar, and menus for uploading and compiling. Its standard programming language is C++ which uses GNU C++ compiler. The build code composed using the Arduino IDE interacts with the ESP8266 Node MCU and also ESP32. It allows to upload the

composed code into the Node MCU, which defines the hardware functions. The Figure 4.1 shows the Arduino IDE interface.

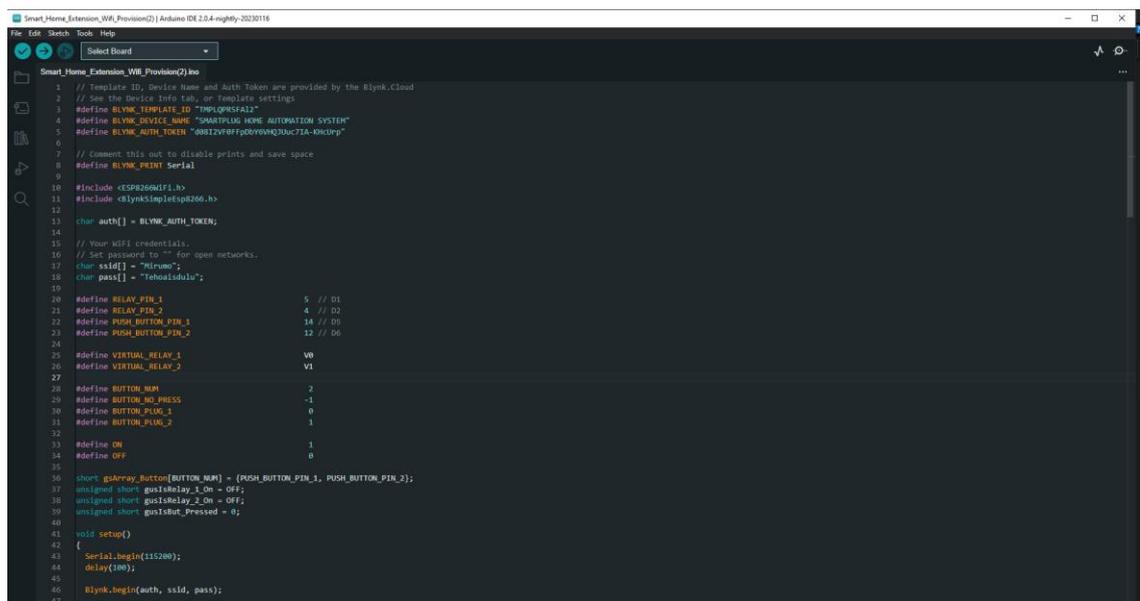


Figure 4.1 Android IDE Interface

## 4.2 Smartplug Home Automation System

The user of this system are for residential houses, office workplace, lab controlled environment and also cafeteria. The purpose of this sytem is to monitor and control electrical appliances and also as a fire alarm emergency detection. Moreover, this system also provided with humidity monitoring to monitor humidity in the place and automate the Air humifier connected to the Smartplug.

### 4.2.1 Login Interface

Because of the limitations of using Blynk software as prototype application for development. User will be provided with pre-registered demo account to login and be able to use the system. This interface is used to verify user identification and authorization. After the user enters their username and password, the system checks the data in the database to verify the information. If the validation is successful, the user will log in to use the system with ease. The administrator will determine the username and password. Therefore, there is no need to register. In Figure 4.2, the login homepage is displayed. In the Figure 4.3, login interface is displayed. In the figure 4.4 Server IP address and port number is displayed.

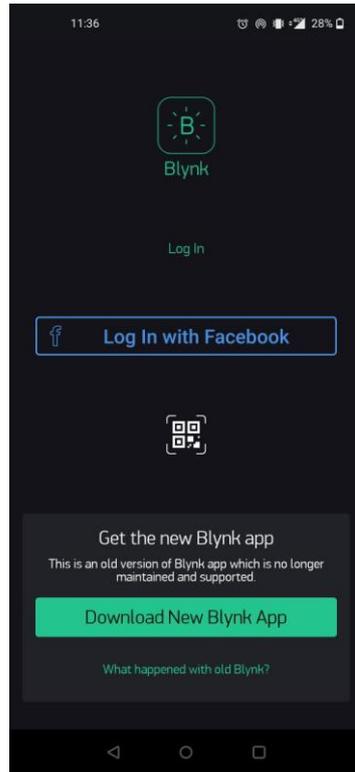


Figure 4.2 Login Homepage

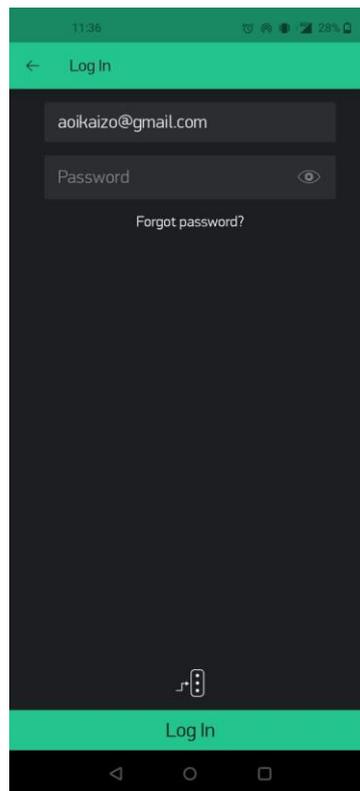


Figure 4.3 Login Interface

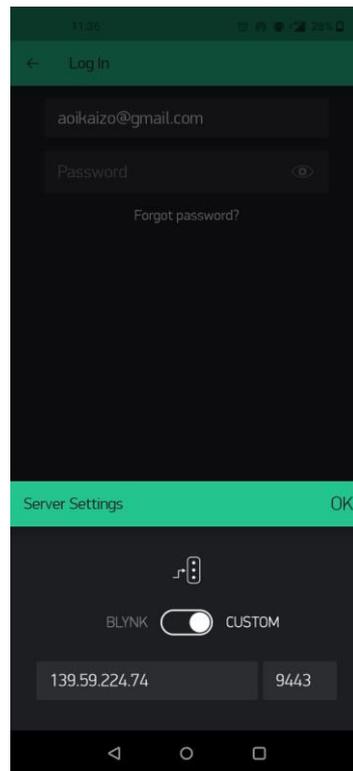


Figure 4.4 Server IP Address and port number

#### 4.2.2 Forgot Page Interface

The user will be able to reset the password of the account in the forgot password interface based on the validation of the email address used when purchasing this Smartplug Home Automation System. The developer will then email the user instructions on how to reset the password for the account with the username that shows in Figure 4.5.

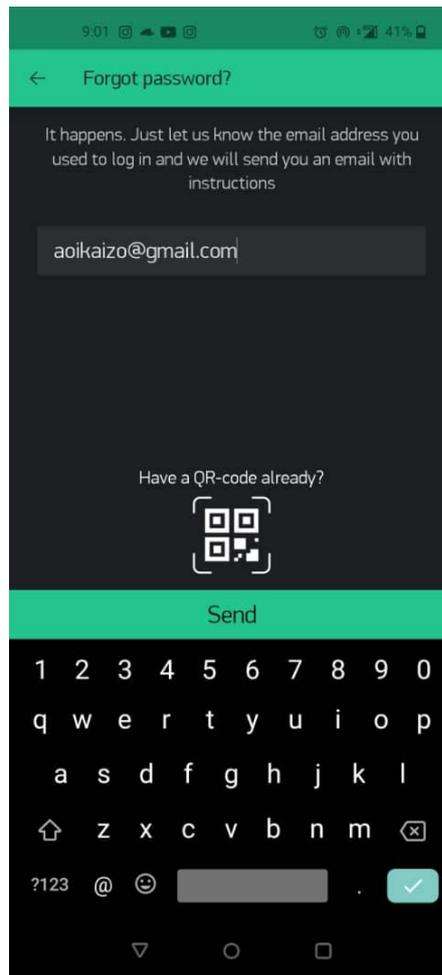


Figure 4.5 : Forgot password interface

### 4.2.3 Homescreen Interface

This is the interface where user can control and monitor the Smartplug Home Automation System. On and off button is used to turn of and on the electrical appliances. Gas dection gauge and alarm can also be seen and will light up and give notification to user if there is any gas or fire detected. This feature is also integrate for future upgrade that can allow alarm and alert for Fire and Rescure Department and provide accurate GPS coordinate to locating the house that using Smarthome Automation System. In the meantime email function is added to allow Fire and Rescue Department to get exact GPS location of the owner that will be hard coded into the system. Figure 4.6 shows the Homescreen Interface and Figure 4.7 shows the timer settings for uer to schedule the timer on and off.

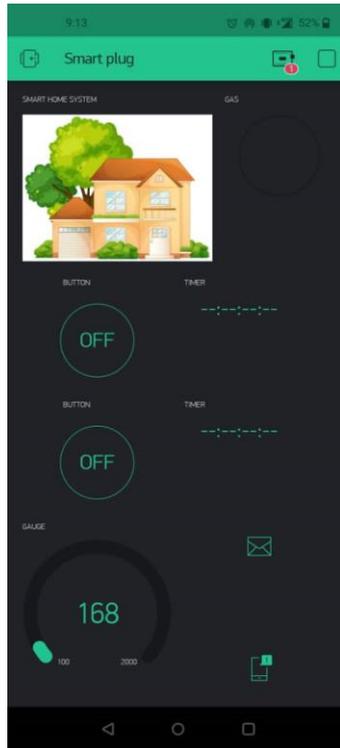


Figure 4.6 : Homescreen Interface

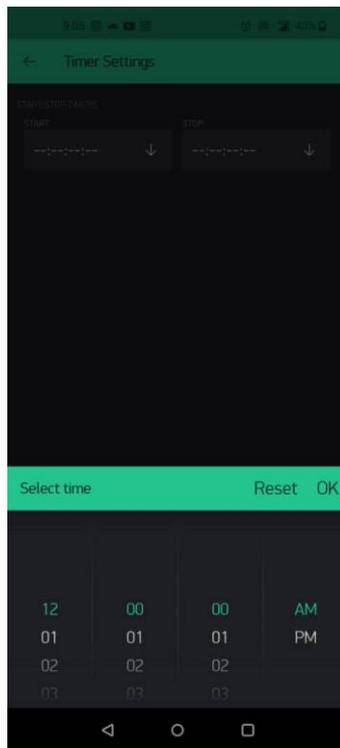


Figure 4.7 : Timer settings

### 4.3 Hardware Implementation

The major component of hardware in this system is the microcontroller. It performs the role of a controller for an integrated circuit that is intended to carry out a task and also act as a WiFi Module for internet connectivity. This system uses the ESP8266 NodeMCU microcontroller board since it has 34 digital input/output pins and incorporates an ESP32 board. The NodeMCU connects with a gas detection sensor. The gas sensor will detect if there is any gas leakage or smoke detected and will notify the user with a notification and for future planning will also alert the Fire and Rescue Department. Using a USB A connection to the computer, the code was uploaded to the NodeMCU, and the serial monitor was used for debugging. The hardware, including ESP8266 NodeMCU, Baseboard NodeMCU, AC Relay, AC to DC converter, gas detection sensor and other parts are shown in Figure 4.6 and 4.7.

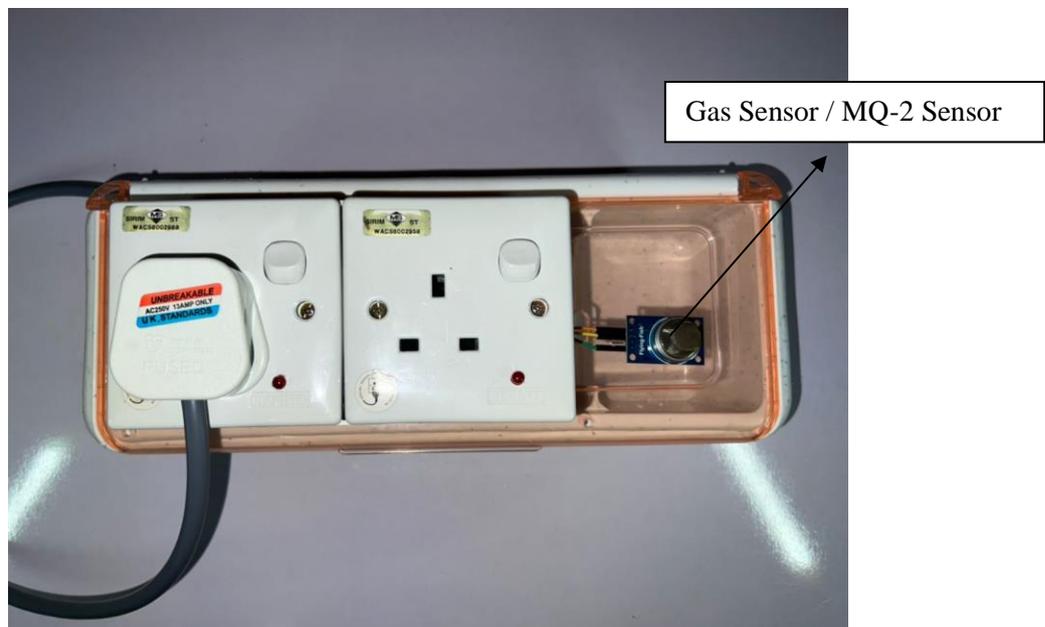


Figure 4.6 Smartplug Home Automation System Hardware

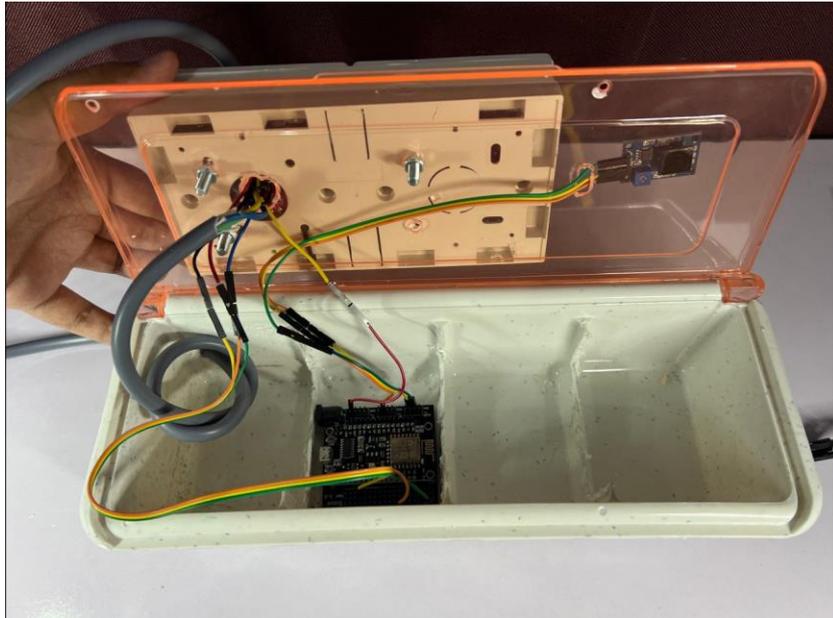


Figure 4.7 View inside of the prototype

The gas sensor or also known as MQ-2 sensor was used to detect gas leakage. It is particularly sensitive to flammable gases such as LPG, propane, methane, and natural gas. The sensor operates on the principle of a heated metal oxide semiconductor that changes its electrical resistance when it comes in contact with the target gas. The MQ-2 sensor consists of a heating element that heats up the sensing element, a metal oxide semiconductor, to a specific temperature. When the target gas molecules interact with the heated semiconductor, the sensor's resistance changes, which is then measured and converted into a detectable signal. This signal is typically processed by a microcontroller or a compatible device to provide gas concentration information.

## 4.4. Aduino Code

Figure 4.8 and 4.9 below shows the code uploaded in the Aduino IDE to properly integrate the sensor with the Smartplug Home Automation System.

```
1
2
3 #define BLYNK_PRINT Serial
4 #include <ESP8266WiFi.h>
5 #include <BlynkSimpleEsp8266.h>
6
7
8 char auth[] = "0kDMiqSZP8uM1V3D8D9Jlg-hd-2jbVS_";
9 char ssid[] = "Perintis robo";
10 char pass[] = "eheks123";
11 // char server[] = "139.59.224.74"; // IP for your Local Server
12 // int port = 8080;
13 BlynkTimer timer;
14
15 int ledPin1 = 5; // GPIO pin for LED 1
16 int ledPin2 = 4; // GPIO pin for LED 2
17 int mq2Pin = A0; // Analog input pin for MQ2 gas sensor
18
19 int gasThreshold = 400; // Set the gas threshold as per your requirements
20
21 WidgetLED notiled(V3);
22
23
24
25 void setup() {
26   Serial.begin(9600);
27   pinMode(ledPin1, OUTPUT);
28   pinMode(ledPin2, OUTPUT);
29
30   timer.setInterval(1000L, checkGas); // Run checkGas function every 1 second
31
32   Blynk.begin(auth, ssid, pass, IPAddress(139, 59, 224, 74), 8080);
33 }
34
35
36
37 BLYNK_WRITE(V0) // Virtual pin to control LED 1
38 {
39   int ledValue1 = param.asInt();
40   digitalWrite(ledPin1, ledValue1);
41 }
42
43 BLYNK_WRITE(V1) // Virtual pin to control LED 2
44 {
45   int ledValue2 = param.asInt();
46   digitalWrite(ledPin2, ledValue2);
47 }
```

Figure 4.8 : Shows the code for control Smartplug

```

47 }
48
49 void checkGas() {
50   int gasValue = analogRead(mq2Pin);
51   Blynk.virtualWrite(V2, gasValue); // Send gas sensor value to virtual LED widget on V2
52   Serial.print("MQ-2 Sensor Value: ");
53   Serial.println(gasValue);
54
55   if (gasValue > gasThreshold) {
56     digitalWrite(ledPin1, LOW); // Turn off LED 1
57     digitalWrite(ledPin2, LOW); // Turn off LED 2
58     Blynk.virtualWrite(V0, LOW);
59     Blynk.virtualWrite(V1, LOW);
60     notified.on();
61     Blynk.notify("Gas Detected!"); // Send notification in Blynk app
62     Blynk.email("aokalzo@gmail.com", "RUMAH ABU TERBAKAR", "SILA BAGI PERGI KE LATITUDE DAN LONGITUDE INI 3.54820164538800, 103.43408199547798"); // Send email notification
63   } else {
64     notified.off();
65   }
66 }
67
68 void turnOnLed() {
69   digitalWrite(ledPin1, HIGH); // Turn on LED 1
70   digitalWrite(ledPin2, HIGH); // Turn on LED 2
71   Blynk.virtualWrite(V0, HIGH);
72   Blynk.virtualWrite(V1, HIGH);
73
74 }
75
76 BLYNK_WRITE(V4) // Virtual pin to control the timer
77 {
78   int timerValue = param.asInt();
79
80   if (timerValue == 1) {
81     timer.setTimeout(5000L, turnOnLed); // Turn on the LED after 5 seconds (adjust the delay as needed)
82   }
83 }
84
85 void loop() {
86   Blynk.run();
87   timer.run();
88 }

```

Figure 4.9 : Shows the code for gas sensor

## 4.5 Blynk Application

Using Blynk, users can set up alerts to receive notifications on their smartphones when the gas sensor detects a high gas concentration. These alerts can include visual indicators, sounds, or even push notifications. This immediate notification enables users to quickly respond to potential gas leaks or fire hazards. Additionally, Blynk offers control capabilities for the smart plug. If the gas sensor detects a high gas concentration, the Blynk application is programmed to automatically cut off power to connected appliances or devices as a safety measure. This helps prevent any potential ignition sources and minimizes the risk of fire or explosion. By integrating Blynk with a smart plug equipped with a gas sensor, users gain the ability to remotely monitor and control their home's gas safety. It provides peace of mind, convenience, and an additional layer of protection against potential gas-related incidents.

### 4.5.1 Blynk Connection

Data retrieval from the Blynk server was required for the interface's registration function and data display. Therefore, the first action prior to data retrieval is to connect to the Blynk server. The connection to Blynk is written in C. Figure 4.10 depicts the code for the Blynk connect function. The Blynk function requires the address of the authentication token, the ssid, and the password. Validate the database connection using the Blynk.begin() function.

```
#define BLYNK_PRINT Serial
#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>

char auth[] = "0kDMiqSZP8uMlV3D8D9Jlg-hD-2jbv5_";
char ssid[] = "Perintis robo";
char pass[] = "eheks123";
// char server[] = "139.59.224.74"; // IP for your Local Server
// int port = 8080;
```

Figure 4.10 : Blynk connec function code

## 4.6 Testing and Discussion

This section will go over all the project data in terms of testing and discussion, it consists of hardware and software testing to ensure that the sensors, and mobile applications are on track to achieve the system's goal. The type of gas detected by the gas sensor will influence the result on the Smartplug. The Smartplug safety function such as auto cut off power and give user alert and notification should be tested to work properly integrated with the gas sensor. The hardware prototype, mobile application, and internet connection will be used for testing. The system was subjected to the User Acceptance Test for overall testing.

#### 4.6.1 Alert notification on Smartplug Blynk Application

When gas is detected on the Smartplug, the alert notification will show in the Blynk app and also in the notification panel of the user phone. Figure 4.11 shows the gas detected notification on the Blynk app. As a result, the system will automatically send email of the GPS coordinate that is already hardcoded in the system to the Fire and Rescue Team Department email address. This will instantly notify them the exact location of the user house and send rescue immediately. Figure 4.12 shows the email example sent to the Fire and Rescue Team Department email.

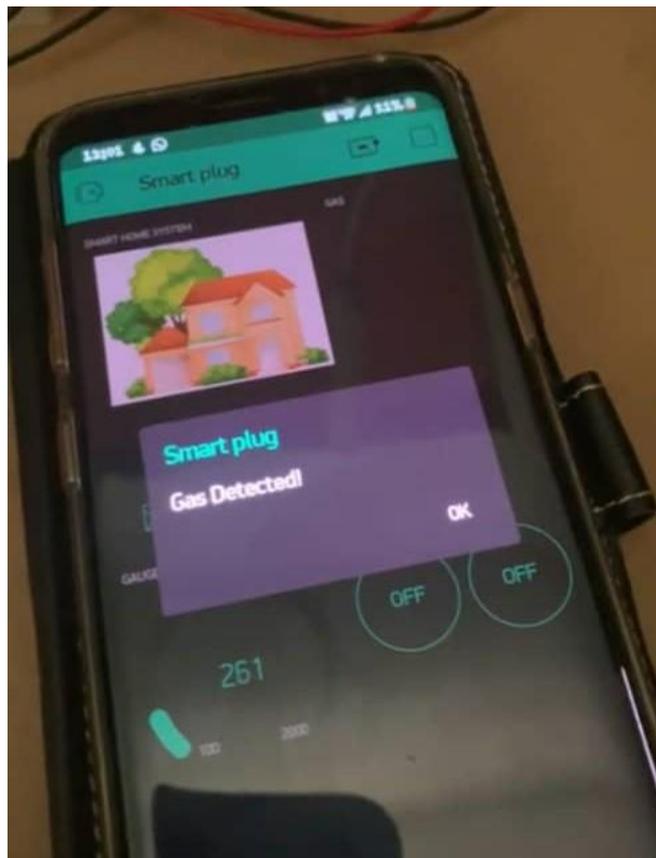


Figure 4.11 : Gas detected notification in the Blynk App

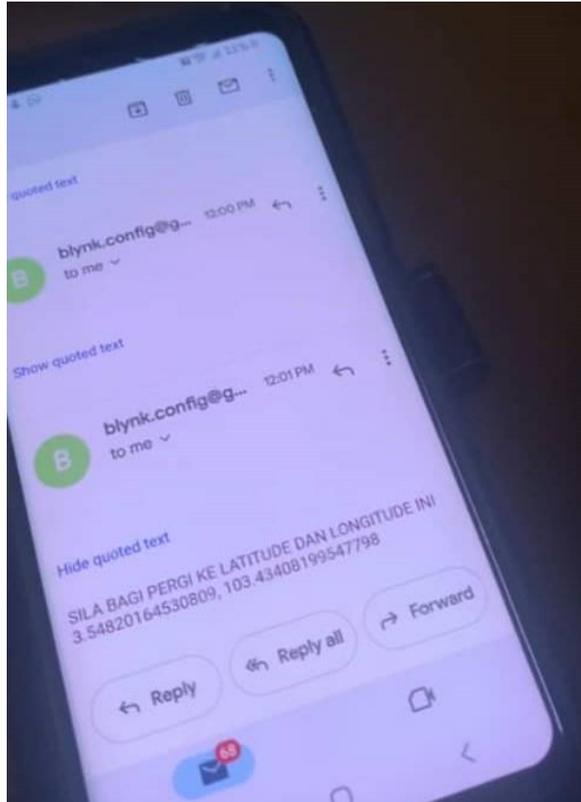


Figure 4.12 : Email example sent to the Fire and Rescue Team Department email.

#### 4.6.2 Result on Gas Sensor Detection

Table 4.1 depicts the condition of the alert notification and led light when the gas sensor begins to detect gas at a specific distance. To begin, as shown in the table, the gas sensor begins to provide output to the alert notification and led light in the app at a distance 50 cm and below. Second, the gas sensor will no longer detect gas at 50 cm and above

<b>Distance</b>	<b>&lt; 20 cm</b>	<b>30 cm</b>	<b>&lt; 50 cm</b>	<b>60 cm</b>	<b>75 cm</b>
<b>Output</b>					
<b>Detect Gas / Not Detect Gas</b>	Detect	Detect	Detect	Not Detect	Not Detect

Figure 4.13 shows the testing of the gas sensor. The system was tested with a lighter to detect gas in example of a gas leakage. The Smartplug will automatically cut off power to connected appliances or devices as a safety measure. It will also alerting the user by notification and directly email the coordinate of the house location to the Fire and Rescue Team Department email. This test was tested in minamal range to ensure the gas sensor works properly. The sensitivity of the gas sensor can be adjusted to increase the range of the detection. In the other hand, if the gas sensor is adjusted to more sensitive, it can lead to inaccurate detection. Thus we set it to the minimum level to increase the accuracy of the detection.



Figure 4.13 : Testing of the gas sensor

### 4.6.3 Result on Auto cut off power when detect gas leakage

Table 4.2 shows the state of Smartplug when gas sensor detects any gas leakage. To begin, as shown in the table, the Smartplug begins to turn off all plug when it detects any gas leakage using the gas sensor.

Condition \ Output	With Gas Detected	No Gas Detected
Plug Turn off / Stays on	Turn off	Stays on

Table 4.2 : Result on Auto cut off power when detect gas leakage

### 4.6.4 Result on timer settings function

Table 4.3 shows the state of Smartplug when a specific time is set on the timer. To begin, as shown in the table, ther Smartplug will automatically to turn off all plug when it reach the time set in the timer settings. Moreover, the Smartplug will keep the plug turned on if the timer does not reach the specific time yet.

Condition \ Output	Reached timer set time limit	Not reach timer set time limit
Plug Turn off / Stays on	Turn off	Stays on

## 4.7 Conclusion

The implementation of this system began with hardware, with all the required components placed together in the Nodemcu ESP8266 with the Nodemcu Base Board, followed by the development of the mobile application. After configuring the hardware and the mobile application with basic compose code to run both, the connection to the Blynk server was established. The sensors and connection to the Nodemcu were connected using jumper cables. The combined sensor code was uploaded into the Nodemcu using the Arduino Ide, and the mobile application was configured and installed. The Blynk server was simple to set up to connect the Nodemcu to the mobile application.

The testing was done to ensure that the gas sensor works properly when it detects any gas leakage. The testing was done to ensure that the Nodemcu could transmit the output via gas sensor detected a distance of 50 cm or less, and that the user will get alert and notification about the gas leakage and also notify Fire and Rescure Department Team through email that contains coordinate of the specific location. Finally, the Smartplug Home Automation System includes a safety feature measures that allows the smartplug to automatically cut off power to connected appliances or devices as a safety measure. This helps prevent any potential ignition sources and minimizes the risk of fire or explosion and potentially saving human lives.

## **CHAPTER 5**

### **CONCLUSION**

#### **5.1 Constraints**

These constraints, including those related to hardware, resources, time, and technology, were encountered during the deployment of the system.

##### **5.1.1 Hardware Constraints**

The implementation of the gas sensor was an extremely difficult process due to the sensitivity of the sensor. The sensor has limitations on detecting gas leakage further than 50cm. Thus making the sensor main objective prone to a failure from detecting gas leakage in the room. To counter the problem, a better sensor that is more expensive is needed to ensure the detecting being more accurate to achieve the main objective which is to detect any gas leakage in the room. Because of the budget limitation, a better sensor could not be replaced thus, the current gas sensor is modified to adjust the sensitivity as accurate as possible. Even though all hardware constraints were encountered, all hardware was made sure to be functioning before the proposal submission deadline.

##### **5.1.2 Time Constraints**

The system was entirely implemented before the deadline, but it could have benefited from several enhancements. Because the mobile application utilised Blynk Application with basic required functions, additional functions could be implemented, as will be discussed in the following section regarding future work. Due to the time constraint pertinent to the enhancement system's implementation, the improvement was added to the discussion of the system's future work.

### **5.1.3 Cost Constraints**

Students were unable to afford the necessary hardware for this system, and the need to repeatedly purchase microcontrollers and cables in order to maintain functionality increased the overall cost of completing the system.

### **5.1.4 Technical Constraints**

In order to implement this system, the Internet of Things and mobile applications were new concepts to master. Due to a dearth of understanding, the burden of inquiry and learning the programming language for Arduino, Blynk Application, and the setup of the ESP8266 NodeMCU with the sensors had increased. There were numerous failures from the beginning of code implementation until the system was completed, but the Smart Smartplug Home Automation System was completed after exhaustive research and learning about the concept and implementation process.

### **5.1.5 Future Work**

The implementation of future work was crucial for enhancing the system's reliability and functionality. This system lacks a few functions that could be introduced in the future to make it more efficient. The enhancement of the system will be the focus of these upcoming efforts. Even though the smartplug can detect gas leakage, there are few improvements that could be implemented in the system's future work.

The system lacks of more accurate gas sensor. A more accurate gas sensor is needed to make sure the gas leakage can be detected even from a far distance. The current gas sensor have limitations on detecting gas leakage far from the sensor. A better and more accurate gas sensor could results the Smartplug Home Automation System to work flawlessly to cover bigger area.

Aside from that, the current system is using hard coded GPS coordinates to locate the location of the Smartplug. By using a proper GPS sensor and system ntegration. The Smartplug Home Automation System will provide specific data even when it is changing it location.

As the mobile application was built with Blynk, it can be improved by using standalone mobile application that allows the developer to have full control on their mobile application function. It also can allow user to install the application easily without having the hassle to properly set up the Blynk App. In conclusion, Blynk application still provides sufficient amount of function that can be explored and integrated with the current prototype and successfully integrate the sensor with the Smartplug. As a result, the Blynk App provides user with proper control of the Smartplug Home Automation System.

## REFERENCES

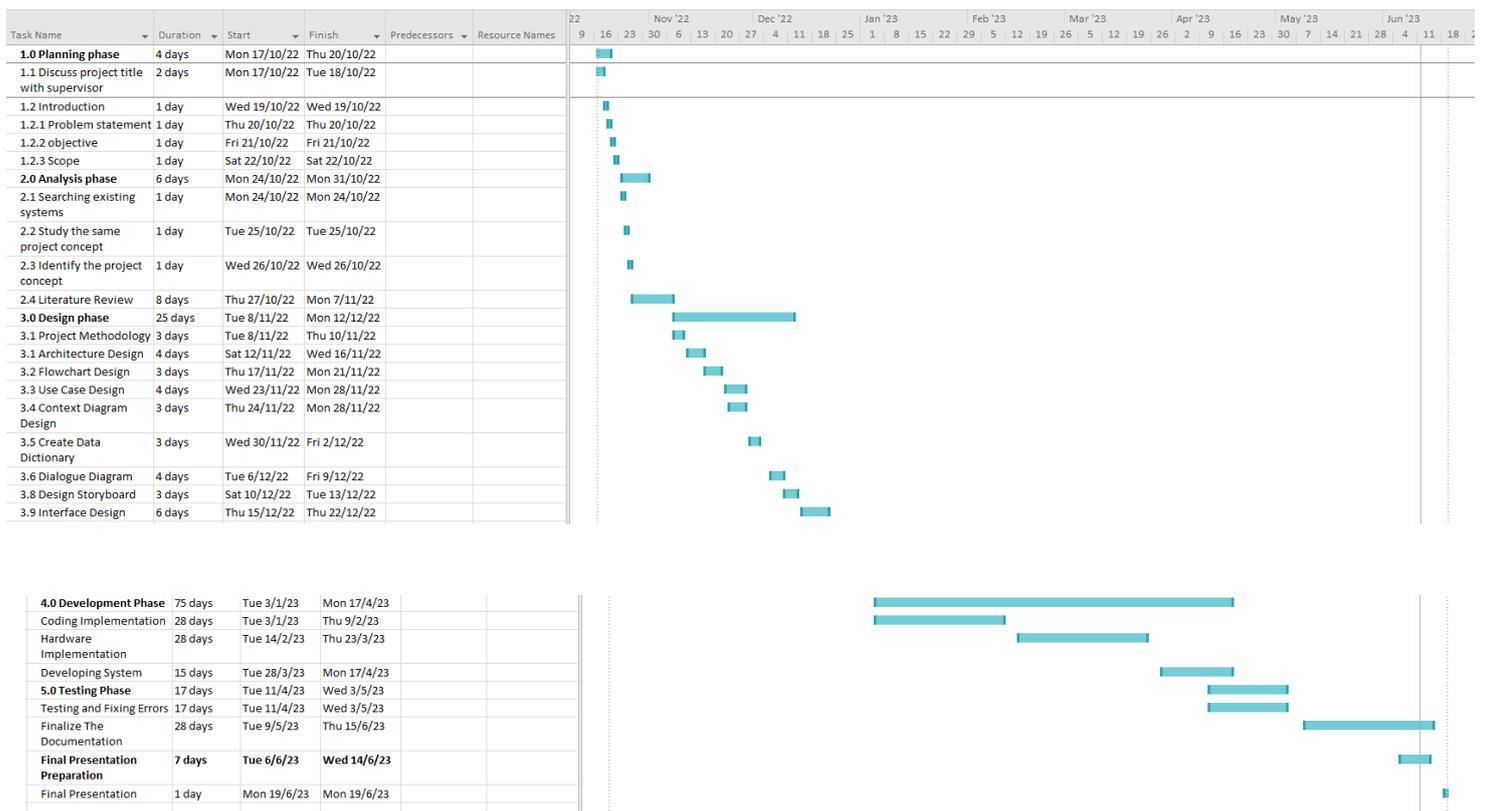
- [1] Jeong, D. J., Ewha Womans University, Park, J. H., Ryoo, H. Y., Ewha Womans University, & Ewha Womans University. (2017). Users' perception on the functions of smart plug. *International Journal of Human and Smart Device Interaction*, 4(1), 7–12. <https://doi.org/10.21742/ijhsdi.2017.4.1.02>
- [2] Kodali, R. K., Jain, V., Bose, S., & Boppana, L. (2016). IoT based smart security and home automation system. 2016 International Conference on Computing, Communication and Automation (ICCCA), 1286–1289.
- [3] Jordan. (2021, September 8). Electrical energy waste and misuse at home. Westline Electrical Services. <https://www.electricianinperth.com.au/blog/how-electrical-energy-loss-and-misuse-is-wasted-at-home/>
- [4] Ghazal, M., Akmal, M., Iyanna, S., & Ghoudi, K. (2016). Smart plugs: Perceived usefulness and satisfaction: Evidence from United Arab Emirates. *Renewable and Sustainable Energy Reviews*, 55, 1248–1259. <https://doi.org/10.1016/j.rser.2015.07.096>
- [5] Ahmed, M. S., Mohamed, A., Homod, R. Z., Shareef, H., Sabry, A. H., & Bin Khalid, K. (2015). Smart plug prototype for monitoring electrical appliances in Home Energy Management System. 2015 IEEE Student Conference on Research and Development (SCORED), 32–36.
- [6] Horvat, I., Lukac, N., Pavlovic, R., & Starcevic, D. (2015). Smart plug solution based on bluetooth low energy. 2015 IEEE 5th International Conference on Consumer Electronics - Berlin (ICCE-Berlin), 435–437.
- [7] “What is waterfall model? - Definition from WhatIs.com,” TechTarget, 2019. [Online]. Available: <https://searchsoftwarequality.techtarget.com/definition/waterfall-model>.

- [8] JAIGANESH PERIYASAMY, “What is Waterfall model and list its advantages, disadvantages,” 2013. [Online]. Available: <http://jobsandnewstoday.blogspot.com/2013/04/what-is-waterfall-model.html>.
- [9] M. Rouse, “What is the Prototyping Model?,” TechTarget, 2019. [Online]. Available: <https://searchcio.techtarget.com/definition/Prototyping-Model>.
- [10] “Top 12 Software Development Methodologies & its Advantages / Disadvantages | TatvaSoft,” TatvaSoft, 2015. [Online]. Available: <https://www.tatvasoft.com/blog/top-12software-development-methodologies-and-its-advantages-disadvantages/>.
- [11] D. Rottmann, “Joint Application Development (JAD).” [Online]. Available: [https://www.umsl.edu/~sauterv/analysis/488\\_f01\\_papers/rottman.htm](https://www.umsl.edu/~sauterv/analysis/488_f01_papers/rottman.htm).
- [12] Olalekan Stephen, “Joint Application Development (JAD): Advantages & Disadvantages - Video & Lesson Transcript | Study.com,” Study.com, 2019. [Online]. Available: <https://study.com/academy/lesson/joint-application-development-jad-advantagesdisadvantages.html>
- [13] Kissflow, “Rapid Application Development: Definition, Steps, Advantages and Case Study,” Kissflow, 2018. [Online]. Available: <https://kissflow.com/rad/rapid-applicationdevelopment/>.

# APPENDIX A

## GANTT CHART

The Gantt Chart displays the project's status together with its timeline. The project will adhere to the suggested methodology, which entails the initial planning phase for requirements, planning phase for requirements, requirements phase, analysis and design phase, implementation phase, testing phase, evaluation phase, and deployment phase.



## APPENDIX B

### USER ACCEPTANCE TEST (UAT)

#### 1.0 Testing Report

This section documents the user acceptability test for the Smartplug Home Automation System. Each module is tested to ensure that the system is adequate and conforms to the specifications. Acceptance of this testing indicates that the system has been exhaustively tested and is prepared for implementation. The identified vulnerabilities have been fixed.

#### 1.1 Mobile Application Testing Plan

Table 1.0: Mobile Application Testing

Test Case	Test Data	Expected Result	Actual Result	Pass/Fail	Comment
Login with correct username and password	Username: aoikaizo@gmail.com Password: maker	Login successful and redirect to the Blynk Apps	Login successfully and redirect to homepage	Pass	-
Login with incorrect username and password	Username: aoikaizo Password: 123456	Login unsuccessful and error message pops up and returns login page	Login unsuccessfully and error message popup	Pass	-
Alert Light	Blynk App	Light up when detect gas leakage	Light up when detect gas leakage	Pass	-

When view alert notification	Blynk app	Display the message of notification	Display the alert notification	Pass	-
------------------------------	-----------	-------------------------------------	--------------------------------	------	---

## 1.2 Sensors, Hardware, and Database Connecting Testing Plan

Table 1.2: Sensor, Hardware, and Database Connection Testing

Test Case	Test Data	Expected Result	Actual Result	Pass/Fail	Comment
When gas sensor detect gas leakage	Gas sensor	Automatically cut of power on all plug	Automatically cut of power on all plug	Pass	-
When plug is turn off in the Blynk app	Plug Socket	Plug is turned off	Plug is turned off	Pass	-
When plug is turn on in the Blynk app	Plug Socket	Plug is turned on	Plug is turned on	Pass	-
When timer setting is set at specific time and ends at the specific time	Plug Socket	Plug off	Plug will be off when reachd the specific time	Pass	-
When timer settings time is on going	Plug Socket	Plug on	Plug will stay on when the timer is ongoing	Pass	-