MOBILE AND WEB APPLICATION ENABLED IOT ECOSYSTEM FOR SMART LIVING

BY: HABAMUNGU TAKIZALA Victoire

ROLL NUMBER: 202150358

Research project submitted in partial fulfilment of the requirement for the award of Advanced Diploma in Electronics and Telecommunication Technology

Supervised by: Eng. Jean de Dieu HAGENIMANA

September 2024

DECLARATION-A

This research project is entirely unique with no submissions made to universities or other educational institutions for credit toward a degree or other academic honours. No portion of this study may be duplicated or reproduced without permission from ULK POLYTECHNIC INSTITUTE or the author.

Student name: HABAMUNGU TAKIZALA Victoire

Signature: Date:

DECLARATION-B

I confirm that the work reported in this research project was carried out by the candidate under my supervision and it has been submitted with my approval as the UPI supervisor.

Supervisor name:

Signature: Date:

DEDICATION

I would like to dedicate this work to my parents, my older siblings, my little sister and my friends, they have provided me with invaluable financial and moral support but also presence throughout this career path.

ACKNOWLEDGEMENT

I want to express my heartfelt gratitude first and foremost to God, for his protection all along this journey and to several individuals who have been instrumental in completing this research project. Firstly, I am immensely thankful to my parents, TAKIZALA Gabriel, and Willermine MWANGAZA for their invaluable advice and financial support throughout this journey, and secondly to all my siblings and family members for their moral support and encouragement.

I sincerely appreciate my supervisor Mr Jean de Dieu for his guidance and expertise and the university for providing the necessary resources and environment for this study. Special thanks go to the HOD KARIKURUBU Emmanuel and the dedicated lab staff, MUSHIMYIMANA Jean Damascene and HAGENIMANA Jean de Dieu, for their technical assistance and support.

ABSTRACT

As the world has adopted the use of digital, smart living solutions are becoming increasingly essential in improving convenience, efficiency, and sustainability in modern urban environments. This research project, titled **"Mobile and Web Application Enabled IoT Ecosystem for Smart Living"**, aims to address key challenges in the current IoT usage, such as non-interoperability, limited remote access, security concerns, and lack of energy monitoring. The study proposes the development of a mobile and web-based IoT ecosystem powered by an ESP32 microcontroller, integrating a Progressive Web Application (PWA), an SMS gateway, and energy monitoring systems. The system will focus on three core areas: remote IoT control of electrical devices, real-time energy consumption monitoring, and seamless communication through an SMS gateway, allowing users to manage their smart homes efficiently.

By the use of Python, JavaScript, and C++, the platform provides users with a flexible, secure, and personalized solution to control their IoT devices across varying locations. Additionally, the study has investigated existing limitations in current IoT systems by offering a more user-centric alternative that enhances both functionality and security.

These findings contribute to the advancement of smart living technologies, promoting energy savings, enhanced security, and improved connectivity while mitigating the limitations found in conventional IoT ecosystems.

Keywords: IoT System, Smart Living, ESP32.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ACRONYMS AND ABBREVIATIONS

API: Application Programming Interface

CLI: Command Line Interface

CSS: Cascading Style Sheets

ESP32: Espressif Systems 32-bit Microcontroller

FTP: File Transfer Protocol

GPIO: General Purpose Input/Output

GUI: Graphical User Interface

HTML: Hypertext Markup Language

HTTP: Hypertext Transfer Protocol

HTTPS: Hypertext Transfer Protocol Secure

IDE: Integrated Development Environment

IoT: Internet of Things

JS: JavaScript

JSON: JavaScript Object Notation

LDR: Light Dependent Resistor

PCB: Printed Circuit Board

REST: Representational State Transfer

Rwf: Rwandan Franc

SQL: Structured Query Language

SSID: Service Set Identifier

SSL: Secure Sockets Layer

TAM: Technological Adoption Model

TLS: Transport Layer Security

UI: User Interface

URL: Uniform Resource Locator

UX: User Experience

VSC: Visual Studio Code

Wi-Fi: Wireless Fidelity

HOD: Head of Department

CHAPTER 1: GENERAL INTRODUCTION

1.0 Introduction

Smart living solutions depend on the balance between physical devices and digital platforms as technology advances. In complex urban environments, smart tech saves time and improve convenience and enjoyment. However, non-operability among different technologies and platforms, particularly with the Internet of Things (IoT), remains a major challenge.

The project "Mobile and Web Application Enabled IoT Ecosystem for Smart Living: A Case Study in Bridging Internet and Mobile Communication by an SMS Gateway" introduces a complete solution to this problem. Both a web application and Progressive Web Application (PWA) utilizing Python, JavaScript, and C++ will be created, running on an ESP32 as the main hardware. The system will focus on three core areas: remote IoT control of electrical devices, real-time energy consumption monitoring, and seamless communication through an SMS gateway, allowing users to manage their smart homes efficiently. This project is all about making things better for consumers by connecting mobile and internet stuff to make smart living even smarter. We're also going to look at how current IoT systems have their limits and show how this super cool new tech can deal with those issues. In this chapter, we'll lay out our research goals and approach, with a focus on creating and rolling out this awesome IoT ecosystem.

1.1 Background of the study

Smart homes are getting popular, and that means we need more third-party services to handle all the IoT devices, thanks to new automation rules. These services are convenient, but they don't always give us the customization and control we want. Most centralized platforms that manage IoT systems set limits on what we can do, which makes it hard to add new devices or create custom functions (Aloi, et al., 2016). Also, relying on third-party platforms for control can open us up to security risks (Kumar & Sharu, 2020). According to research by Kumar and Sharu in 2020, there are serious security and privacy issues in IoT environments, especially when they rely on different cloud computing services (Kumar & Sharu, 2020). Securing IoT devices can be tricky because they have limited resources and vary so much. This is why it is important to move towards less centralized ways of managing IoT, so we can have more control without sacrificing security. The challenges of integrating different systems and the security issues mentioned in these studies suggest that future smart living solutions need to focus on easy-to-use designs and stronger security for IoT technologies.

1.2 Statement of the Problem

Today's IoT systems for managing and communicating with electrical devices have their limitations. With the IoT revolution, we have seen advancements like Wi-Fi and Bluetooth for online device control, but these solutions are not always robust enough for anytime/anywhere command capability. The connections often have limited range and reliability, which is not ideal for controlling devices remotely, especially when the user is far from home or in an area with poor cellular coverage. Additionally, IoT systems lack personalization, often trapping users in third-party silos that dictate how devices can be managed and connected. This not only limits functionality but also raises privacy and security problems. Another issue is the lack of energy monitoring in current IoT solutions. Most offline system options do not provide detailed energy consumption estimates for individual electrical components, making it challenging for users to optimize their energy usage and costs. Lastly, the digital divide poses a communication challenge between internet-connected users and those without smartphone or internet access, especially when trying to connect across different technology platforms or communicate without digital surveillance, such as in international scenarios where one party is limited to internet communication while the other desires an alternative connection.

1.3 Research objectives

1.3.1 General Objective

The general objective of this research is to design a mobile and web application-enabled IoT ecosystem for smart living and bridging internet and mobile communication by an SMS gateway

1.3.2 Specific Objectives

After looking at some research and checking out what is currently out there for IoT systems, here is what we're aiming to do in this study:

1. To Create and set up an IOT system to control electrical appliances remotely.

2. To integrate a flexible and personalized platform for all users.

3. To add an advanced energy monitoring system to save energy.

4. To create an SMS gateway system for notifications and alerts.

1.4 Research questions

To guide this research and ensure that it meets its objectives, the following research questions have been formulated:

1. How to create and set up an IoT system to control electrical appliances remotely?
2. How to integrate a flexible and personalized platform for all users?
3. How to add an advanced energy monitoring system to save energy?
4. How to create an SMS gateway system for alerts and notifications?

1.5 Scope

The scope of this study is limited to developing and testing a smart living ecosystem that connects mobile devices, web applications, and IoT devices through a unique platform.

The research will focus on developing software with a web application and Progressive Web App (PWA) for better mobile accessibility. The system will include a frontend created using HTML, CSS, and JavaScript with AJAX for asynchronous communication. A Python backend with Django Rest Framework (DRF) will provide high-level server-side logic. The hardware component will be an ESP32 microcontroller programmed in C++, enabling remote communication and control of electrical devices. The project aims to create a customizable platform and implement an advanced energy monitoring system. Additionally, the project will explore telecommunication systems by developing an SMS gateway to investigate the interaction between mobile communication and internet-connected devices.

1.6 Significance of study

This study addresses significant challenges in smart living and communication. It aims to improve the control of electrical devices through a remote IoT command system, overcoming the limitations of traditional Wi-Fi and Bluetooth technologies. By developing a customizable platform, the research addresses the need for flexible device management, reducing reliance on third-party services.

Additionally, it explores telecommunication systems to bridge communication gaps, particularly when reaching someone outside the country who is not using the Internet. This is crucial for maintaining contact in areas with inconsistent internet access, as seen in neighboring regions. The integration of an SMS gateway will enhance communication reliability, offering practical solutions to real-world connectivity issues.

1.7 Organization of the Study

This research is organized into the following chapters:

Chapter 1: General Introduction

This chapter provides an overview of the study, including the background, problem statement, research objectives, significance, and scope of the study.

Chapter 2: Literature Review

This chapter examines existing literature related to IoT ecosystems, smart living technologies, energy monitoring, and telecommunication systems. It also identifies gaps in the current research that this study aims to address.

Chapter 3: Research Methodology

This chapter outlines the research design, methods, and tools used to develop and implement the IoT ecosystem. It also details the data collection and analysis processes employed in the study.

Chapter 4: System Design, Analysis, and Implementation

This chapter presents the design and technical analysis of the IoT ecosystem, including the integration of the ESP32 microcontroller, software development, and the implementation of the SMS gateway. It also discusses the testing and evaluation of the system.

Chapter 5: Conclusion and Recommendations

This chapter summarizes the findings of the study, draws conclusions, and provides recommendations for future research and practical applications of the IoT ecosystem developed in this study.

CHAPTER 2: LITERATURE

2.0 Introduction

This chapter will explore existing research and expert opinions on IoT ecosystems, smart living technologies, energy monitoring, and telecommunication systems. The goal is to understand the current state of these technologies, their applications, and the challenges they pose. We aim to identify gaps in the development of a customizable IoT ecosystem for promoting smart living through remote commands and communication. Understanding the evolution of IoT ecosystems and smart living technologies requires delving into current research. By identifying what is functioning effectively and what is not, we can uncover new opportunities for innovation. This review forms the base of work for developing an IoT ecosystem to develop smart living conditions.

2.1 Concepts, Opinions, and Ideas from Experts

The leading type of smart environment around the globe and, primarily, for managing energy consumption or communication systems is in IoT ecosystems. Some trends and challenges are becoming apparent as this research explores these areas.

2.1.1 IoT for Energy Management

Energy Management Systems (EMS) leverage IoT technology to minimize energy consumption and actively engage users in managing their energy use. These systems offer real-time data on energy usage, down to the level of individual appliances, providing users with a clear understanding of where energy is consumed. This level of detail enables users to make more informed decisions, such as identifying energy-hungry devices and adjusting their usage patterns accordingly. By offering insights into energy trends, EMS empowers users to take direct action to reduce waste and optimize efficiency(Batista et al., 2017).

Home Energy Management Systems (HEMS) are a subset of EMS specifically designed for residential use. They track, monitor, and analyze energy consumption in real time, allowing homeowners to see exactly how much energy is being used at any moment. HEMS often include features such as alerts, automation, and recommendations that encourage sustainable practices. For example, the system can automatically turn off appliances when not in use or suggest more energy-efficient settings for devices. This helps reduce overall energy costs while promoting environmentally friendly behavior, supporting the goal of long-term sustainability (Cano-Ortega & Sánchez-Sutil, 2022).

2.1.2 Smart Living Technologies

Smart homes and cities are becoming increasingly intelligent through the use of IoT technology, which aims to improve quality of life by automating everyday tasks and managing resources more efficiently. These systems can control various aspects of a home or city, such as lighting, heating, traffic management, and energy consumption, making life more convenient and sustainable. However, despite these advancements, IoT systems still face significant challenges, particularly when it comes to data privacy, security vulnerabilities, and the ability to adapt to different environments and needs.

Fortunately, advancements in cloud computing and data analytics are helping to address these challenges. These technologies provide the infrastructure needed to manage and process vast amounts of data collected by IoT devices, enabling more secure and scalable systems. This progress allows us to build flexible IoT solutions that can be tailored to different applications, such as real-time flood monitoring in smart cities. By integrating cloud and data analytics, IoT systems can become more versatile, better equipped to handle a wide range of tasks, and more reliable in protecting sensitive information (Mori, Kundaliya, Naik, & Shah, 2022).

2.1.3 Custom IoT Systems

Custom IoT systems are super flexible and customizable, which makes them perfect for specific uses like agriculture. They can bring together different sensors, microcontrollers, and communication protocols to give personalized functions that you can't get with off-the-shelf solutions. Take the Smart IoT Planting System, for example – it is a great way custom IoT frameworks can meet the special needs of farming. Unlike standard IoT solutions, these systems can be adjusted to meet specific user requirements and integrate different components according to the application's needs. (Naghib, Navimipour, Hosseinzadeh, & Sharifi, 2022).

2.1.4 Design of IoT Systems Using Web Frameworks Such as Python Django

The use of Python Django for IoT systems has gained traction due to its robust capabilities in handling complex data and perfectly integrating components. Recent studies have shown the effectiveness of combining Python Django with Ajax for real-time data communication and control in IoT systems. (Sebastian, 2024).

Bluetooth is a wireless technology standard for transmitting data between fixed and mobile devices over short distances and for establishing personal area networks. Zigbee wireless technology was created specifically for low-cost sensors and control devices, and it is extensively utilized in a variety of application (Afonso et al., 2023; Batista et al., 2017)

In a study by Yixin Sun and Hoekyung Jung, the optimization of organizational operations using IoT was examined, with a focus on the role of Python Django in facilitating effective communication between devices and web servers. Their research included detailed architectural figures, demonstrating how Django can be integrated with other technologies like MQTT for efficient data handling and device management (Alfonso, Garcés, Castro, & Cabot, 2021).

2.1.5 Research gaps

The use of IoT systems has advanced, but some areas need more attention and my project will have objectives of filling these gaps:

1. IoT systems more customizable for different needs

IoT systems need to be more customizable to meet the specific needs of different users and environments. Right now, many IoT solutions are designed with a "one-size-fits-all" approach, which may not work well in every situation. By making these systems more flexible, they can be adapted to various use cases, whether it's for a home, a business, or an industrial setting. Customizability will allow users to fine-tune IoT devices according to their unique requirements, improving efficiency and usability (Cano-Ortega & Sánchez-Sutil, 2022).

2. Better security and privacy for IoT in smart homes and cities

Security and privacy are critical issues for IoT devices, especially in smart homes and cities where sensitive data is constantly being collected. Weak security can expose users to risks such as data theft, hacking, and even physical harm. Improving security protocols is necessary to ensure that personal information stays protected and IoT systems remain safe from cyber-attacks (Mori, Kundaliya, Naik, & Shah, 2022).

3. New technologies can improve IoT systems

There is still a lot of untapped potential when it comes to exploring how new technologies can enhance IoT systems. Technologies like artificial intelligence (AI), machine learning, and blockchain could greatly improve the way IoT devices function, making them smarter and more efficient. Research is needed to see how these innovations can be applied to real-world IoT systems (Naghib, Navimipour, Hosseinzadeh, & Sharifi, 2022).

4. Expanding energy management research beyond just homes

Research on energy management using IoT devices has so far been mainly focused on homes, but there is a need to expand this research to other areas like businesses and industries. IoT systems can play a huge role in optimizing energy use on a larger scale, helping companies save money and reduce their environmental impact (Cano-Ortega & Sánchez-Sutil, 2022).

2.2 Theoretical perspectives

This study's theoretical framework focuses on how IoT, Telecommunication, and Energy management can be combined to create new opportunities for smart living environments. The IoT theoretical perspectives for this study are divided into three main areas within the IoT Ecosystems: IoT for Remote Command and Control, Energy Monitoring Management, and Telecommunication Systems in an Internet of Things Ecosystem. Each of these areas is guided by general theories and principles.

2.2.1 Technology Acceptance Model

According to the Technology Acceptance Model (TAM), two things matter for people to accept new technology: how useful they think it is and how easy it is to use. **Usefulness** means that people believe controlling devices with IoT will help them manage things better and save time (Alqahtani, Nourah, Sanaa Sharaf, & Rashid Mehmood, 2022). IoT systems can improve smart spaces by collecting data, sharing information, and letting users easily interact with devices.

The second factor, **ease of use**, is about making sure people find the system simple to use. For remote control to work well, the system needs to be safe with secure passwords, encryption, and ways to detect threats. However, these safety features should not make the system hard to use. By creating easy-to-understand interfaces that work on different devices, we can make sure more people will use and enjoy IoT systems (Alqahtani, Nourah, Sanaa Sharaf, & Rashid Mehmood, 2022).

2.2.2 Innovation Diffusion Theory

Diffusion Theory (IDT) offers some insightful ideas about why people embrace new technologies. Essentially, whether a new technology catches on often boils down to a few key aspects (Rogers, 2003).

People tend to adopt new technologies when they see a clear advantage over what they're already using. If the new gadget or application makes life noticeably easier or saves time, it's more likely to gain traction. (Rogers, 2003).

Then there’s compatibility, how well the new technology fits with users’ existing habits and systems. If the new tool meshes well with what people are already doing and integrates smoothly with their current setup, they’re more likely to adopt it. So, a new app that works seamlessly with popular social media platforms or existing devices will generally find a warmer reception (Rogers, 2003).

Another factor is complexity. If a technology is too complicated or requires a steep learning curve, people might shy away from it. Simplicity and ease of use are huge selling points. The easier something is to understand and operate, the more likely people will use it. Think of how quickly user-friendly apps become popular compared to those that are confusing or hard to navigate (Rogers, 2003).

Trialability also plays a role. When people can try out a new technology on a small scale before committing fully, they’re more inclined to give it a chance. This might mean offering free trials or demos so users can test the waters without making a big investment (Rogers, 2003).

Lastly, there’s observability is how visible the benefits of the new tech are to others. If the positive effects are clear and evident, people are more likely to jump on board. For instance, seeing friends or colleagues successfully using a new tech and enjoying its benefits can make others want to try it too (Rogers, 2003)

2.3 Related Studies

2.3.1 IoT-Based Smart Home Systems

In their study, Alqahtani et al. (2022) explored the implementation of IoT technologies in smart home environments, focusing on energy efficiency and user control. They demonstrated how IoT devices could optimize energy usage and enhance user convenience through real-time data collection and remote management (Alqahtani, Nourah, Sanaa Sharaf, & Rashid Mehmood, 2022).

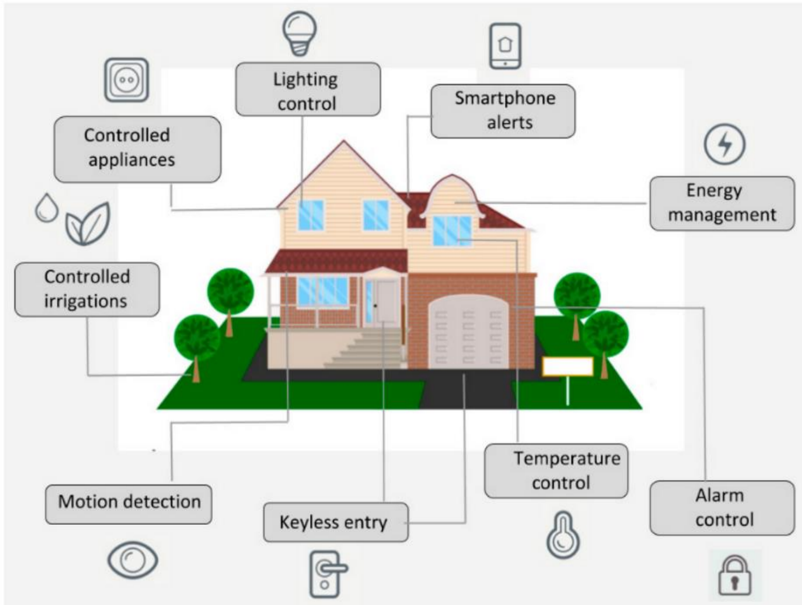

Figure 1 IoT-Based Smart Home Systems (Alqahtani, Nourah, Sanaa Sharaf, & Rashid Mehmood, 2022)

1. **Energy Monitoring Solutions:** Wang, Zhenki, and Fagui (2019) reviewed various energy monitoring systems, highlighting the importance of integrating IoT with energy management for improved efficiency. Their research outlined the benefits of real-time monitoring and data analytics in reducing energy consumption and costs (Wang, Zhenxi, & Fagui, 2019)

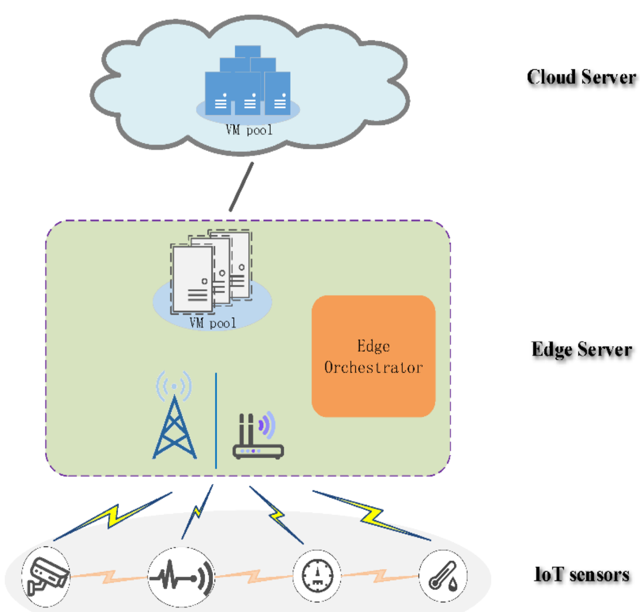

Figure 2 Connected devices to save energy (Wang, Zhenxi, & Fagui, 2019)

2. **SMS Gateways for Communication:** Al-Turjman and Mohammad (2019) investigated the role of SMS gateways in bridging mobile and internet communications. Their findings emphasized the effectiveness of SMS for reliable messaging and its integration with web applications to facilitate seamless communication between users and systems (Al-Turjman & Mohammad , 2019)

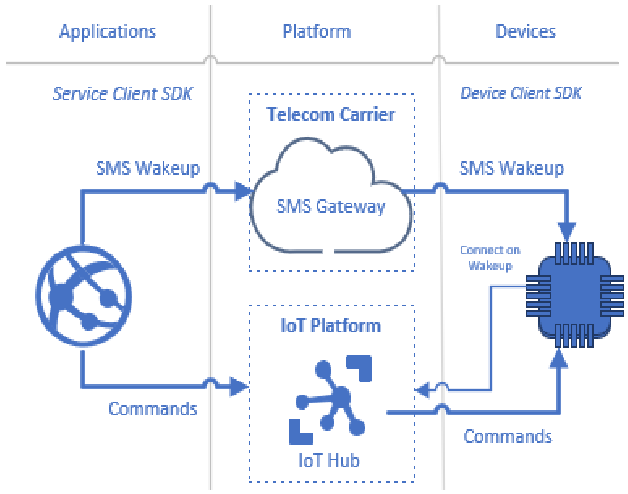

Figure 3 SMS Gateway in IoT System (Research, 2024)

3. **Telecommunication Systems in IoT:** Another study examined the integration of telecommunication systems within IoT frameworks, focusing on how these systems support device connectivity and data transmission. The study highlighted key challenges and solutions for ensuring efficient communication in IoT ecosystems (Zilenkov, Dinis, Ilya, & Yanina, 2018).

4. **Web and Mobile App Development for IoT:** Li, Franglin, Yuanchin, and Brad (2017) explored the development of web and mobile applications for IoT systems, detailing best practices for creating user-friendly interfaces and ensuring compatibility with various IoT devices. Their work provided valuable insights into integrating IoT functionalities with web and mobile platforms (li, Franglin, Yuanchin , & Brad, 2017).

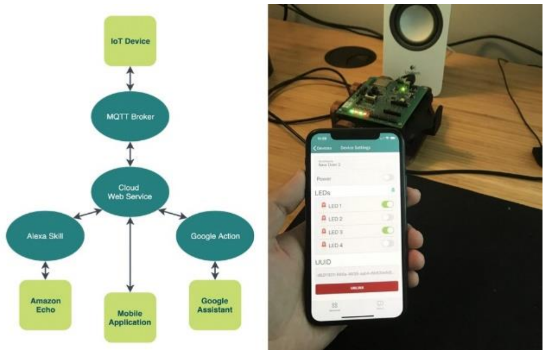

Figure 4 Integrating Mobile App to control IoT tools (li, Franglin, Yuanchin , & Brad, 2017)

CHAPTER 3: RESEARCH METHODOLOGY

3.0 Introduction

The research methodology clearly shows the systematic way it will be used to explore how well mobile and web app-connected IoT systems work in smart living. This part explains how data will be collected and analyzed to make sure the research is solid and transparent, and that the findings are credible. This research uses a mix of numbers and qualitative methods to assess how IoT systems, energy monitoring solutions, and SMS gateways are used in smart living environments.

3.1 Research Design

In this study, we will use a mix of action research and case study methods to look into how Mobile and web app-enabled IoT Ecosystems for Smart Living can make connected devices last longer and use energy better.

Action research is all about solving real-world problems by working together with people to solve specific issues with IoT systems and energy monitoring. It is about continuously testing and refining solutions based on feedback and observations and evaluating outcomes to make ongoing improvements (Somekh, 2006).

On the other hand, the case study research dives deep into specific examples, examining real-world instances of IoT and SMS gateway applications, understanding their operational context and performance, and identifying trends and best practices. By putting these two approaches together, We aim to get a comprehensive understanding of how IoT technologies impact smart living environments and how SMS gateways facilitate communication and enhance system functionality.

3.2 System Users

The users for this research will be simulated for real-life situations involving the use of IoT systems at home and work. The sample includes:

1. Main User: Will represent a regular user of the IoT system, using the technology both at home and at work. This user is the main focus in simulating interactions with IoT systems.

2. Other Users: will be people linked to the main user who also interacts with the IoT systems in different settings, such as family members and office staff.

3.4 Research Instrument

The main research tool for this study involves simulating how a typical user interacts with IoT systems at home and at work. This includes creating scenarios for user engagement in both settings, keeping detailed records of interactions, and noting down user feedback. This method provides a practical understanding of how IoT systems are used and their impact on daily life in personal and professional contexts.

3.5 Data Gathering Procedures

The data-gathering process for this study is designed to simulate real-life scenarios in which users interact with IoT systems in both home and work environments:

1. Simulation Setup: The setup includes smart devices and sensors connected to the IoT ecosystem, controlled by the ESP32 hardware. Specific scenarios are developed to simulate typical user activities, such as remotely controlling devices, monitoring energy usage, and utilizing the SMS gateway for communication.

2. Data Collection Instruments:

- **System Logs:** After implementation, Trials are made to capture all system activities, including user commands, sensor data, and system responses to simulate their feedback.
- **Energy Usage Reports:** Real-time reports are generated to monitor and analyze energy consumption settings during the study.

3.6 Data Analysis and Interpretation

3.6.1 Data Analysis

The analysis focuses on the data gathered from simulations, which is related to real-life interactions with the IoT system.

1. **Quantitative Analysis:**

- **Interaction:** The simulations track how often users interact with different parts of the IoT system, such as controlling devices at home, at work, and monitoring energy.
- **Energy Consumption:** The simulations provide data on energy usage, allowing us to compare how much energy is consumed before and after using the IoT system.

- **System Performance:** The simulations also measure how well the system performs, including how quickly it responds and how accurate it is.

2. **Qualitative Analysis:**

- **User Feedback:** Feedback from the simulations is analyzed to see how satisfied users are with the IoT system. Here, I look for common themes, like ease of use or any challenges users faced without such IoT systems.
- **Scenario Evaluation:** Each simulation scenario is evaluated to determine how well it represents real-life situations and how effectively the IoT system performs.

3. **Comparative Analysis:**

- **Before and after:** The simulations allow us to compare user experiences and system performance before and after implementing the IoT system to demonstrate improvements or benefits.

3.6.2 Expectations

After simulation of user experience, the interpretation of the data will show how the IoT system can impact user experiments, system effectiveness, and daily life.

1. **User Satisfaction:** The simulations will show that people can really like using the IoT system. It will be easy for them to control things like lights or appliances from far away, and they will appreciate how simple it is to keep track of their energy use. The SMS feature will be especially helpful because it makes communication smooth and easy.
2. **System is effective:** The IoT system will do a great job in the simulations. It will be quick to respond and to do what people want without any problems most of the time.
3. **Impact on Daily Life:** For day-to-day life, the use of the IoT system will make life easier for people. They can manage their smart devices at home or work without much effort, and they feel more in control of their energy usage, which can help them save money.
4. **For the future:** These results will show that connecting mobile phones with IoT systems will make life more convenient. In the future, it will be important to make these systems even more connected and to fix any little problems that come up, so living in a smart home can be even better.

3.8 Limitations of the Study

There were a few things that could limit the findings of this research.

1. **Simulation vs. Real Life:** The study relied on simulations, which are not exactly the same as real-life situations. While I tried to make the simulations as realistic as possible, they might not capture every detail of how the IoT system would work in real homes or workplaces.

2. During simulation, the research focused on a small number of users. Although this helped us understand their experiences, it might not fully represent how everyone would interact with the IoT system.

3. The IoT system was tested on specific devices and software. Different devices or systems might perform differently, so the results might not be the same for all technologies.

4. The ESP32 could only handle 5 devices controlled for its well functioning. To prevent heat and burning as it is connected to the internet and users may need more devices to be controlled remotely.

5. The study was conducted over a short period of time. We couldn't see how the IoT system would perform over months or years, which means I might have missed long-term issues or benefits.

CHAPTER 4: SYSTEM DESIGN, ANALYSIS, AND IMPLEMENTATION

4.0 Introduction

This chapter presents the comprehensive design, analysis, and implementation strategy for our project. The design process focuses on the IoT system architecture, including hardware, software, and communication protocols. This chapter will cover system drawings, technical calculations, hardware and software specifications, cost estimation, and the final implementation of the project. Each section will demonstrate the logic and processes behind the design process and provide a detailed analysis of the system components.

4.1 Drawings and Structure of the System

In this section, we present the system's structural and functional designs through diagrams and schematics. These illustrations are essential for understanding the overall architecture and communication flow within the IoT ecosystem. The drawings include both physical parts and logical flowcharts, showing how components such as the microcontroller and communication modules are interconnected. Also, they explain the interaction between the hardware devices and the web-based platform, as well as the data flow for remote control, energy monitoring, and SMS integration.

4.1.1 System Architecture Diagram:

The image shown in Figure 5 provides a high-level overview of the IoT ecosystem's architecture. It illustrates the interaction between the hardware components (such as the ESP32 and sensors), the backend Django web application, and the mobile/PWA interface. The architecture design highlights how data flows between devices and the central platform, enabling real-time control, energy monitoring, and SMS communication.

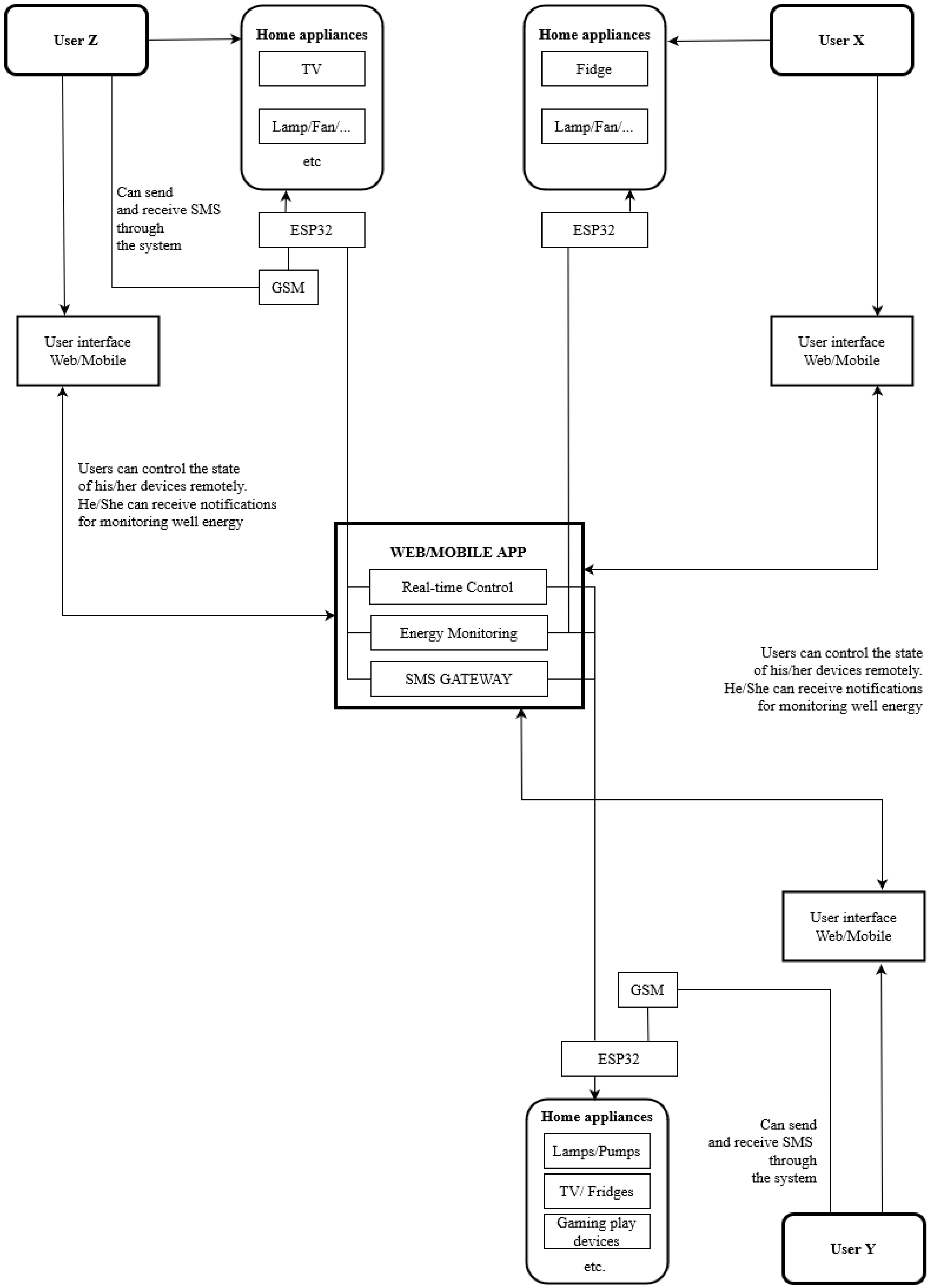

Figure 5 Global Architecture System, Researcher, 2024

4.1.2 Circuit design

The following circuit diagrams as shown in Figure 6 and Figure 7, show the detailed electrical connections. These diagrams are crucial for understanding how power is distributed and how components are wired to ensure seamless communication and operation.

- **Real-time control**

Figure 6 shows the connections for controlling devices and home appliances remotely. The pins 12, 33, 23, 27, and 34 represent the output pins to be connected to relays according to their respective devices (appliances).

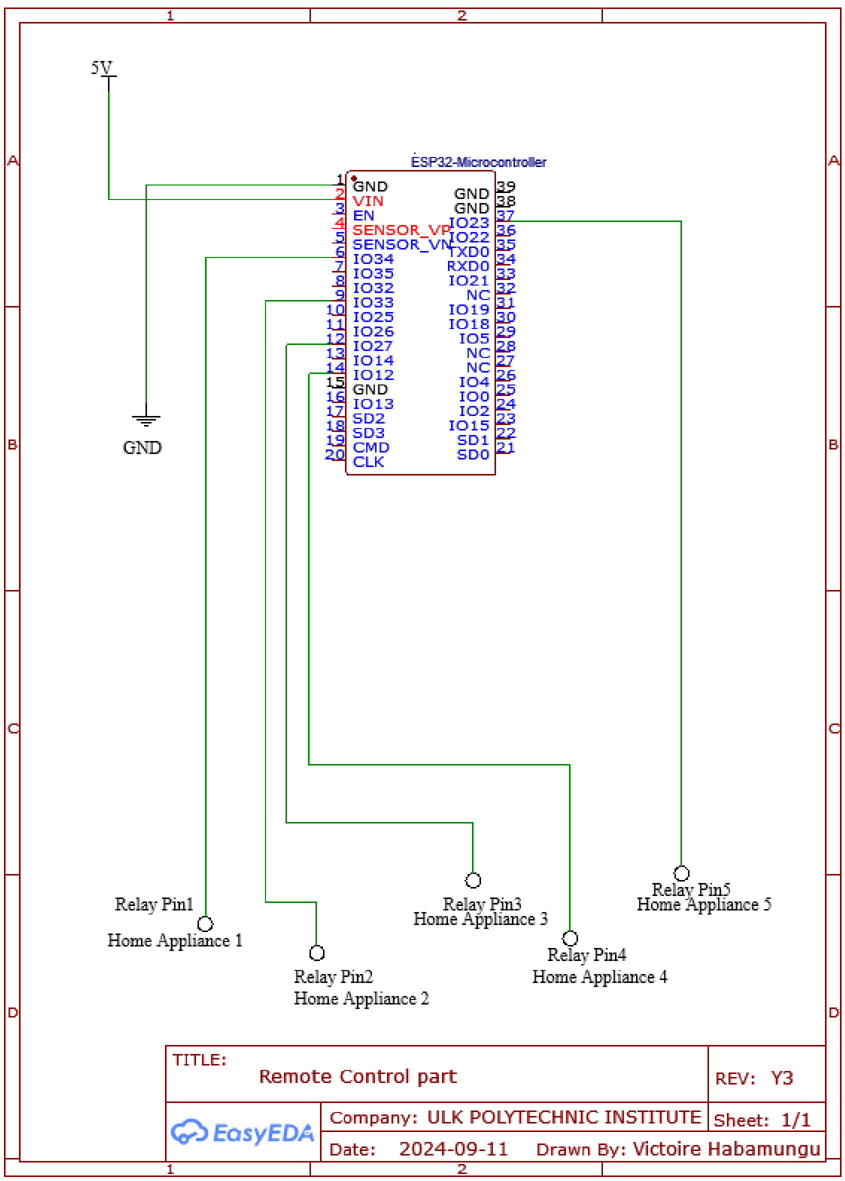

Figure 6 Real-time control, Researcher, 2024

- **Messaging**

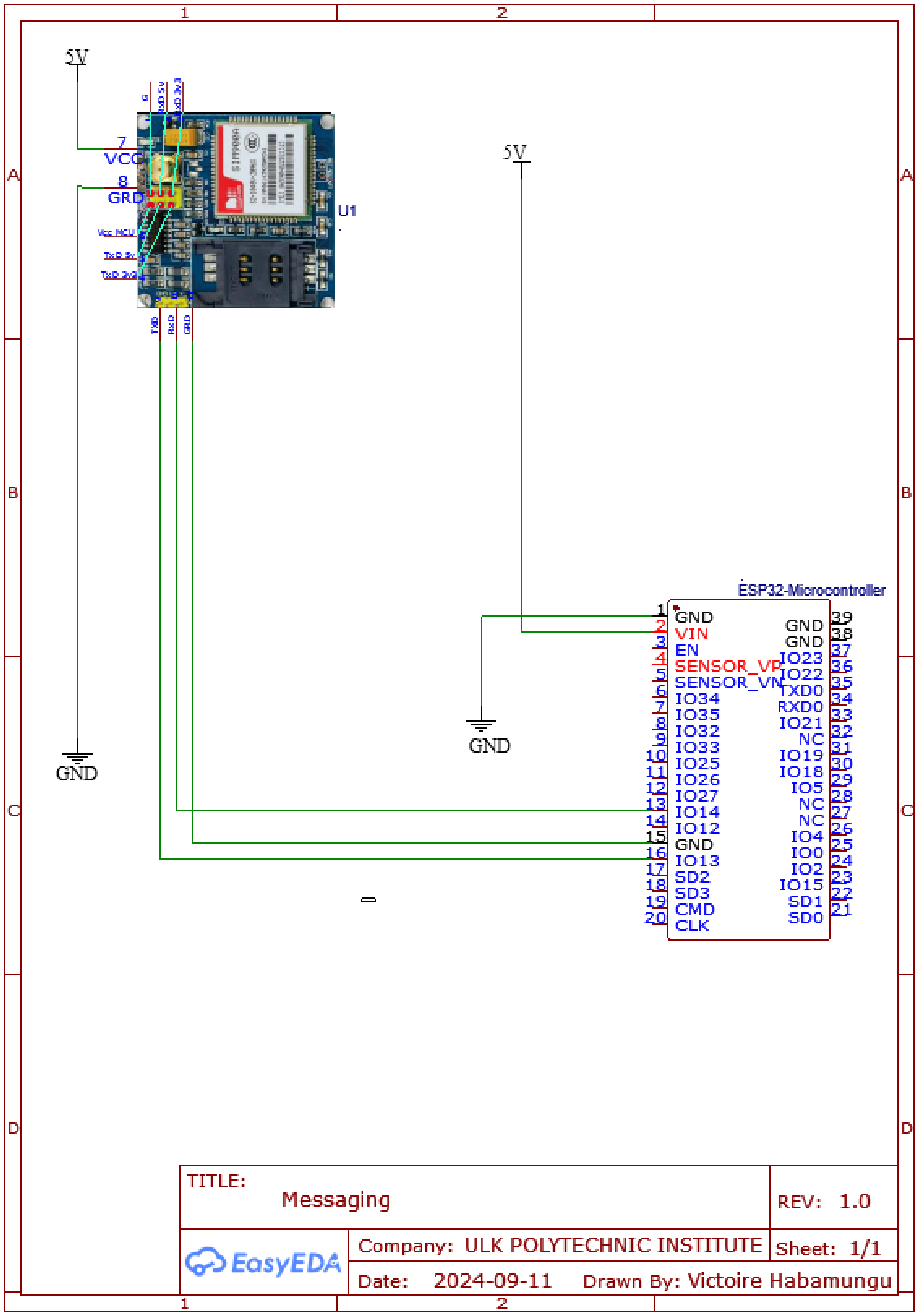

Figure 7 SMS Gateway-Hardware part, Researcher, 2024

4.1.3. Flowcharts

The flowcharts illustrate the logical flow of data and commands within the system. These diagrams outline the sequence of operations, including how commands are sent from the user interface to the IoT devices and how energy data is collected, processed, and reported. They also show the process for sending and receiving SMS messages through the gateway, ensuring smooth integration between mobile communication and internet services.

➢ **Flowchart of Real-time Control**

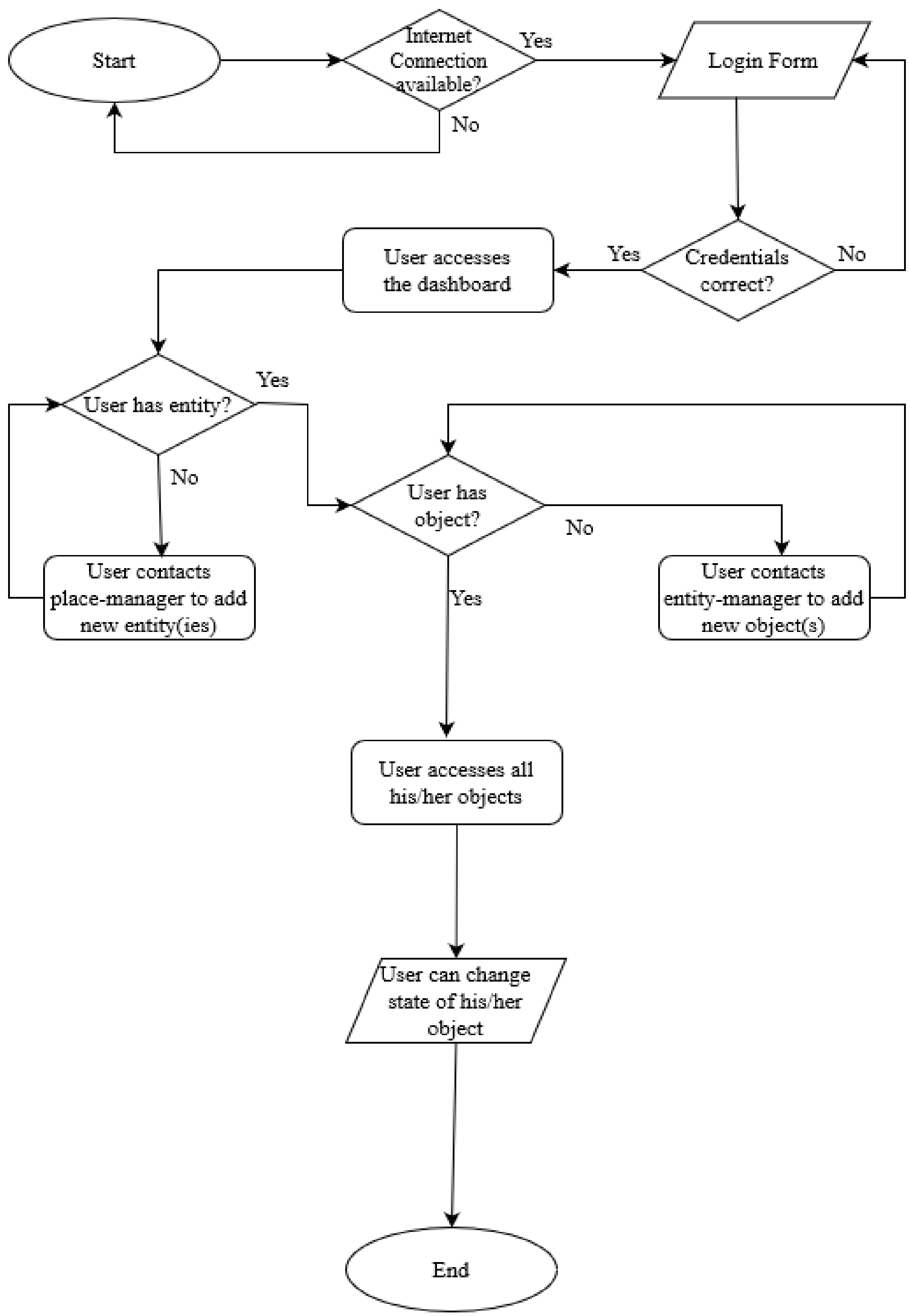

Figure 8 Real-Time Control, Researcher, 2024

- **Flowchart of Energy Monitoring**

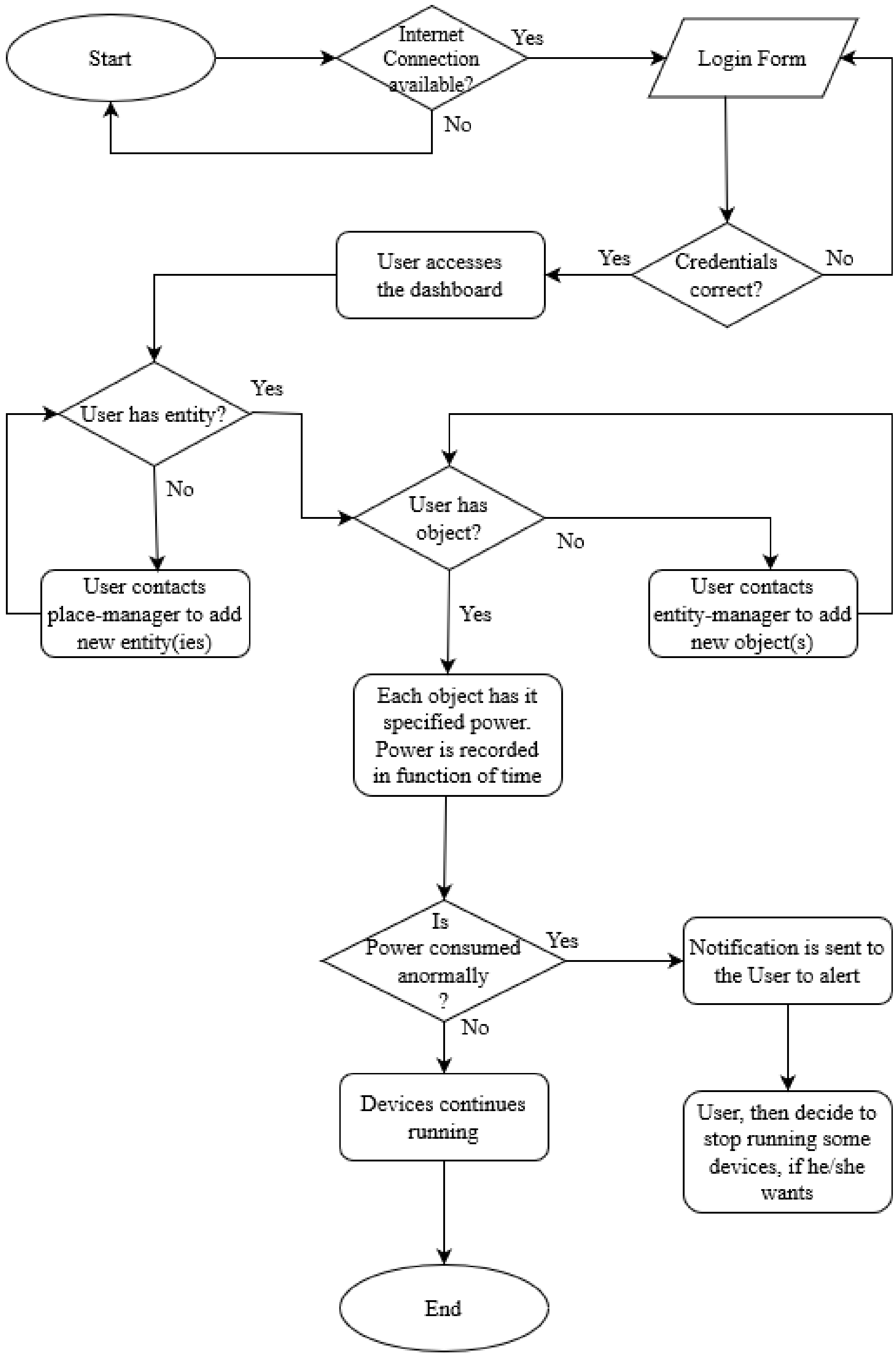

Figure 9 Energy Monitoring, Researcher, 2024

- **Flowchart of the SMS Gateway**

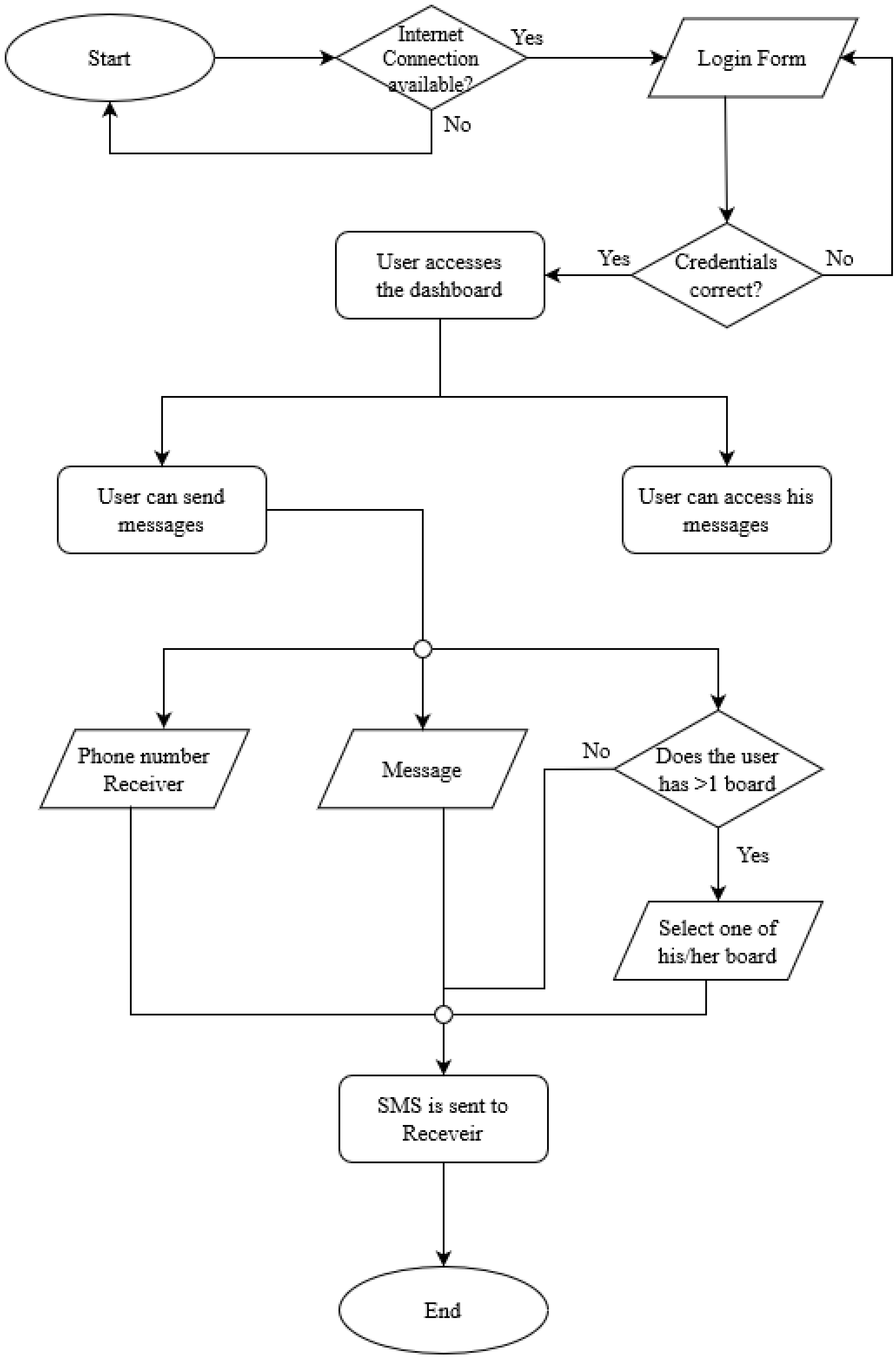

Figure 10 SMS Gateway, Researcher, 2024

4.2 Calculations

In this section, we provide some calculations based on which our system is designed.

Since our system consumes power, here are some indications that helped us when implementing:

Formula: Power(W)=Current(A)*Voltage(V)

For One ESP, we have: P= 0.15A* 5V=0.75W

The consumption of power for the devices controllable remotely depends on their specifications. They will be included in the energy monitoring.

Formula: Energy (Wh)= Power(W)*Time(h)

Security of Microcontroller:

To protect the microcontroller from heating and burning, delayed response, and damaging issues, we need to consider the fact that it is connected to the internet.

As a solution, We adopted to use 5 pins over the total of 30 or 36 pins to prevent all those issues. To identify the time response, we use this method:

Formula: Data Rate (bps)=Message size (bits)×Messages per second.

Considering this formula, if each energy update is 1 Kilobytes and sent every 10 seconds, the required data rate would be approximately:

1000×8÷10=800bps. This explains how the response will be delayed if many devices are connected to one microcontroller.

4.3 Specifications

4.3.1 Hardware specifications

Components	Specifications
ESP32	➢ Microprocessor: Tensilica Xtensa LX6 ➢ Voltage:3.3V ➢ Analog input pins:12-bit, 18 Channel ➢ DAC Pins:8-bit, 2 Channel ➢ Digital I/O Pins:39 (of which 34 is a normal GPIO pin) ➢ DC Current on I/O Pins:40mA ➢ DC Current on 3.3V Pin:50mA ➢ SRAM:520KB ➢ Communication: SPI(4), I2C(2), I2S(2), CAN, UART(3) ➢ Wi-Fi:11 b/g/n ➢ Bluetooth:2 – Supports BLE and Classic Bluetooth
GSM Module	➢ GSM/GPRS SIM900 Module ➢ Quad-Band 850 / 900/ 1800 / 1900 MHz ➢ Low power consumption: 1.5mA (sleep mode), 5V, 12V
Peripherals	Specifications depend on devices to be controlled

Table 1 Hardware Specifications

4.3.2 Software specifications

Type	Specifications

Programming languages	Backend	Python- Django Framework
	Frontend	JavaScript
	Arduino/ESP32	C++
Communication Protocols	Wi-Fi	ESP-Server
	HTTP/HTTPS	for the Django API communication.
	SMS Protocol	for sending/receiving messages.
Database	PostgreSQL	Saving data

Table 2 Software Specifications

4.4 Cost Estimation

The estimated total cost (APPENDIX B) for this project to be implemented is 252000FRWF.

4.5 Implementation

4.5.1 Hardware Setup

The hardware assembly was done following a structured wiring diagram as shown in Figures 6 and 7, ensuring correct connections between the ESP32 and its peripherals. Power management was handled by providing a stable power supply through USB and external voltage for the GSM Module, depending on the energy requirements of the components. Before integrating all parts, each component and actuator was individually tested to ensure functionality. The connections are also the same as shown in those figures. The Pins to be connected to the controllable devices are also defined and chosen in function of recommendations made after doing calculations in point 4.2 of this chapter. The Arduino code (APPENDIX A) was uploaded to the microcontroller via Arduino IDE.

4.5.2 Software Setup

The software component of the system consists of the ESP32 firmware, the backend (Django), and the frontend (PWA).

- **ESP32** : The ESP32 was programmed using the Arduino IDE, with C++ as the primary programming language. The firmware manages communication between the sensors, actuators, and the Django backend. Libraries such as WiFi.h were used to connect the ESP32 to the network, and HTTPClient. enabled communication with the backend for sending SMS and receiving control commands.

- **Backend (Django)**: A Django project was created to handle the web application and API interactions. The backend manages user authentication, device control, energy data logging, and SMS message processing. The Django REST Framework was used to build the APIs that allow the ESP32 to send and retrieve data. A MySQL database stores all the user information, device states, and energy reports. Security was enforced using Django's authentication features to ensure that only authorized users could control the devices.

- **Progressive Web App (PWA)**: The frontend was developed using JavaScript and a PWA framework like React, allowing users to interact with the IoT system from any device. The PWA provides real-time control of smart devices and displays energy usage data. Offline functionality was enabled through service workers, ensuring that users can still interact with their devices even when not connected to the internet.

4.5.3 System Integration and Deployment

The integration phase involved connecting the ESP32 to the Django backend and ensuring smooth communication between all components.

- **ESP32 and Backend Communication**: The ESP32 connected to Wi-Fi and communicated with the Django server via HTTP, receiving commands to control devices.

- **Energy Monitoring**: Sensors sent real-time energy data to the backend, where it was stored in a database. Users could access detailed reports on energy usage through the PWA to help manage consumption.

- **SMS Gateway**: An SMS gateway allows users to send and receive messages through the system, linked to the Django backend for smooth mobile-internet communication.

- **Testing and Debugging**: Each part of the system was tested separately before full integration. Unit tests ensured the backend worked correctly, and functional tests confirmed that users could control devices, monitor energy, and send SMS messages without issues.

After integration and testing, the system was deployed.

- **Backend Deployment**: The Django app was hosted on a cloud platform Heroku. SSL certificates ensured secure communication, and a PostgreSQL database stored data in the cloud.
- **PWA Deployment**: The Progressive Web App was deployed on Vercel, giving users access from both mobile and desktop devices. Offline capabilities were enabled through service workers.

4.6 Results

From simulations that we made and tests that we did, these are the results we obtained from them:

4.6.1 Simulations

The simulations demonstrated the way users interact with the web and mobile interface of the system. We saw how well commands were done and executed for the parts of registration, login, addition of entities, of objects, and how they were displayed. Regardless of the time, the results remained the same for the same simulations. Here are some pictures for energy monitoring and for real time control:

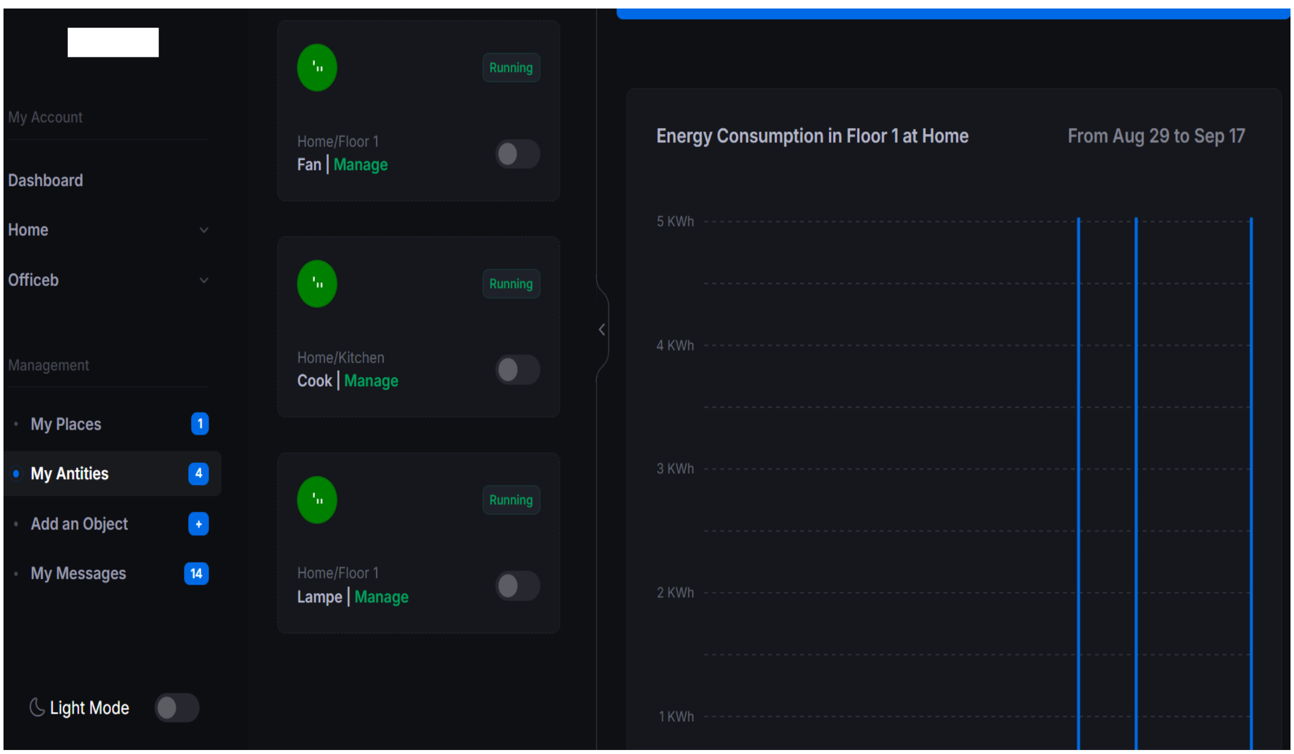

Figure 11 Energy Monitoring (Web view), Researcher, 2024

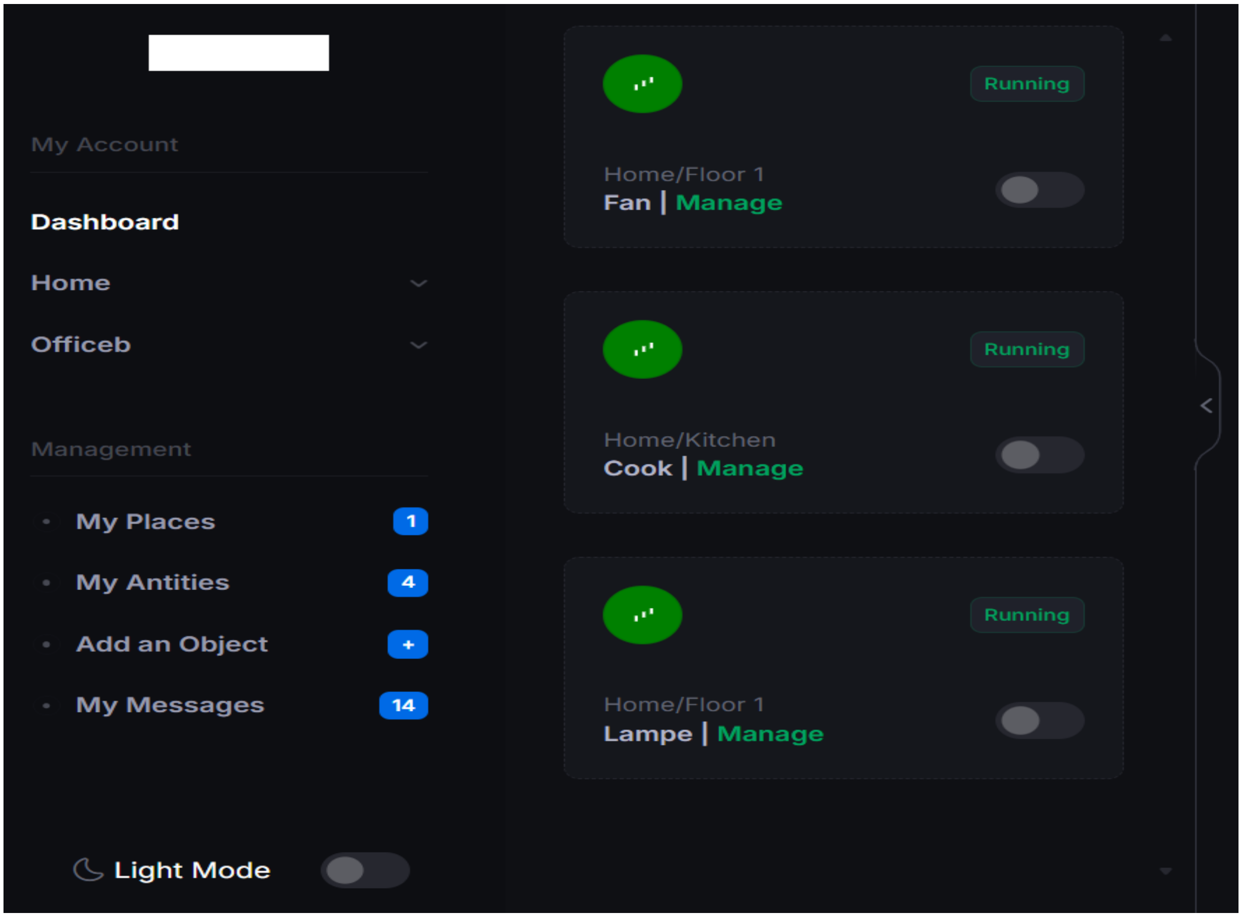

Figure 12 Real time Control (Web view), Researcher, 2024

We also experienced, by simulating a user, the interaction of hardware and software. First on sending messages from the interfaces then receiving them on the receiver's mobile phone as SMS. The same commands were reproduced multiple times to make sure the responses were coherent over time.

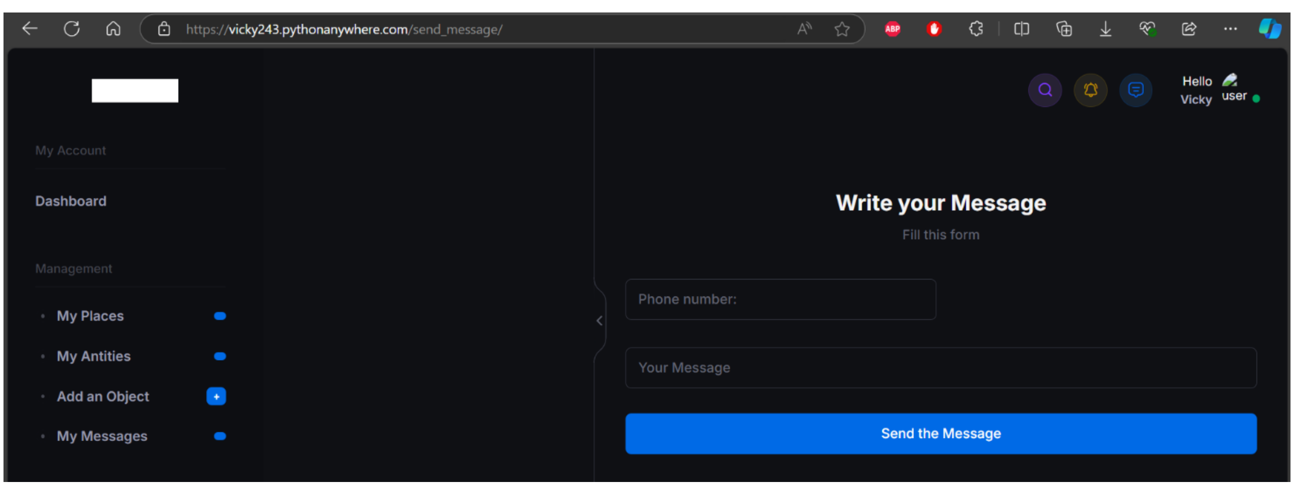

Figure 13 Form for writing a message from WebView, Researcher, 2024

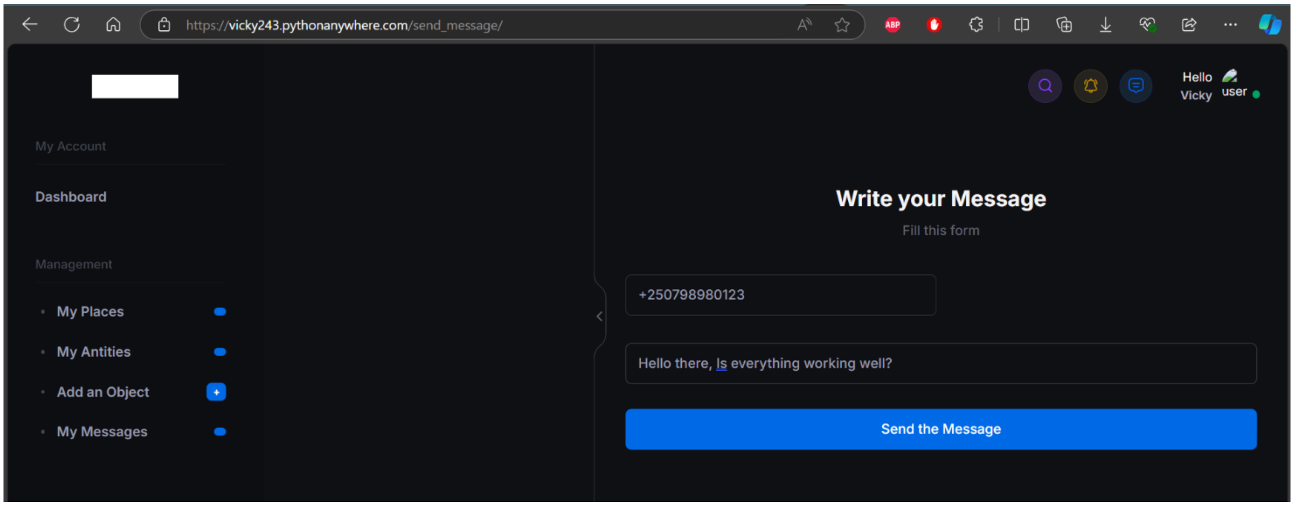

Figure 14 The phone number of receiver and content of the message are filled, Researcher, 2024

Then the message is received in form of SMS on receiver side after some seconds of delay:

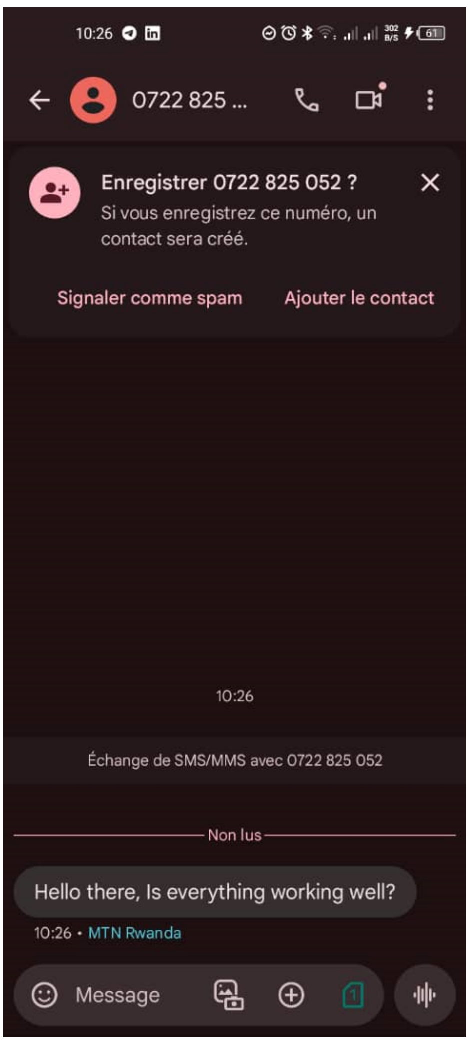

Figure 15 SMS received on Recipient side, Research, 2024

4.6.2 Hardware testing

We tested the hardware part of the system by uploading the code to the ESP32 board. The code contained Wi-Fi parameters to allow the board to be connected to the internet. Tests were made to ensure that the system responded to commands received from the server (for remote control) and received SMS on mobile phones as we connected a GSM module to another ESP32(for SMS Gateway). Messages were well received and commands were well executed. The same commands were reproduced multiple times to make sure the responses are coherent over time.

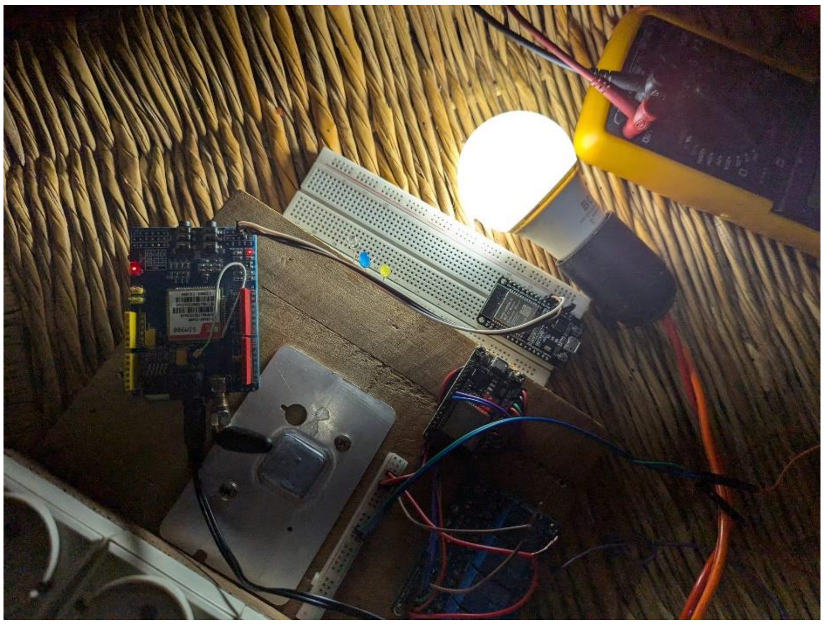

Figure 16 Testing hardware (Sept 14th), Researcher, 2024

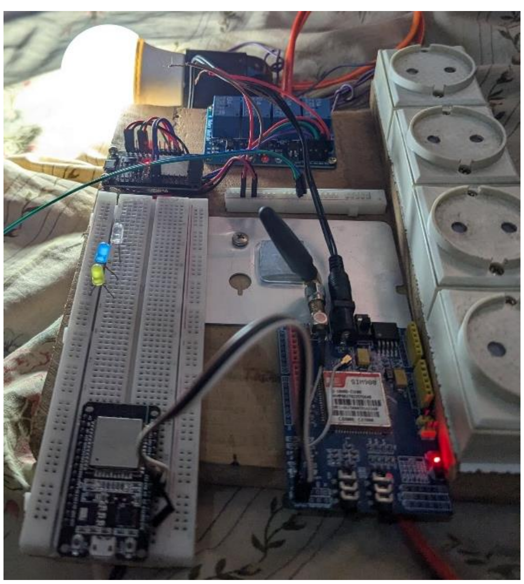

Figure 17 Test hardware (Sept, 5th), Researcher, 2024

4.7 Conclusion

In this chapter, we explained the design, analysis, and implementation of the IoT ecosystem for smart living. We detailed the system architecture, circuit designs, and flowcharts, followed by critical calculations and specifications that shaped the project. In the integration phase, We successfully connected hardware, software, and communication protocols, ensuring smooth interaction between the ESP32, Django backend, and the Progressive Web App. Finally, the deployment of both hardware and software components provided a fully functional solution for remote device control, energy monitoring, and SMS-based communication. This integrated system highlights the potential of IoT in enhancing smart living environments.

CHAPTER 5: CONCLUSIONS AND RECOMMENDATIONS

5.0 Introduction

In this final chapter, we summarize the outcomes of the project, and we propose recommendations based on the work accomplished. Additionally, this chapter offers suggestions for further study and improvements to develop the system's effectiveness and expand its operation. The project aimed to design and implement a comprehensive IoT ecosystem for smart living, integrating energy monitoring, remote control, and SMS communication. The conclusions given here reflect the achievements and challenges faced, as well as possible future directions(Afonso et al., 2023).

5.1 Conclusions

The "Mobile and Web Application Enabled IoT Ecosystem for Smart Living" successfully reached the primary objectives of improving the interaction between mobile devices, web applications, and IoT systems. The integration of the ESP32 with the Django backend and a Progressive Web App (PWA) provided a structured platform for users to monitor energy consumption in real time, control smart devices remotely, and send or receive SMS. This project also reached its specific objectives.

Key conclusions include:

- The system successfully provided a smart remote control with the interaction of Django and PWA with ESP connected to the internet.
- The system offers solid energy monitoring capabilities, that give users the possibilities to manage their smart devices from anywhere and personalized.
- The SMS gateway integration improves communication by linking mobile phone networks with internet services through the use of GSM, ESP, Django backend, and PWA.

5.2 Recommendations

Based on the project’s outcomes, these are the recommendations we made to improve the system and its deployment:

1. **Optimize Energy Usage**: Improve the real-time reporting system to handle big volumes of data efficiently and add predictive analytics to help users **anticipate** energy consumption.

2. **Expand Device Compatibility**: Integrating support for a big range of smart devices and protocols to ensure better compatibility across various IoT systems, improving the overall usability of the platform.

3. **Add new features**: integrate a system of reading values of sensors and automatically or manually, modify the state of associated devices i.e. devices that depend on sensor values depending on the wishes of the owner.

To our school and the Polytechnic institute especially, we recommend:

1. **To make available essential equipment** and tools for students to gain practical knowledge as we are technicians.
2. **To adapt the learning methods** and contents to the actual needs of society with the mean they have.
3. **To encourage innovation** and help students to develop their creative skills.

5.3 Suggestions for Further Study

The project achieved its goals, but these are areas where further research and development could improve more the system's functionality and applicability:

1. **Machine Learning Integration**: Look for the potential of integrating machine learning algorithms to provide user preferences based on devices he/she uses frequently.

2. **Ecology and Energy Savings**: Explore ways to incorporate renewable energy sources and smarter energy-saving features within the IoT ecosystem, contributing to a more eco-friendly and sustainable smart living environment.

References

Alfonso, I., Garcés, K., Castro, H., & Cabot, J. (2021). Self-adaptive architectures in IoT systems: a systematic literature review. *Journal of Internet Services and Applications* .

Aloi, G., G., C., G. , F., R., G., P., P., W. , R., & C., S. (2016). A Mobile Multi-Technology Gateway to Enable IoT Interoperability. *IEEE First International Conference on Internet-of-Things Design and Implementation (IoTDI)*.

Alqahtani, E., Nourah, J., Sanaa Sharaf, & Rashid Mehmood. (2022). Smart Homes and Families to Enable Sustainable Societies: A Data-Driven Approach for Multi-Perspective Parameter Discovery Using BERT Modelling.

Al-Turjman, F., & Mohammad , A. (2019). *IoT-enabled smart grid via SM: An overview.*

Cano-Ortega, A., & Sánchez-Sutil, F. (2022). IoT for Energy Management. *Energies, Sensors, Electronics, Smart Cities and IoT*.

Kumar, D., & Sharu, B. (2020). IoT Ecosystem: A Survey on Devices, Gateways, Operating Systems, Middleware and Communication. *International Journal of Wireless Information Networks*.

li, J.-J., Franglin, C., Yuanchin , L., & Brad, M. (2017). Programming IoT Devices by Demonstration Using Mobile Apps.

Mori, H., Kundaliya, J., Naik, K., & Shah, M. (2022). IoT technologies in smart environment: security issues and future enhancements. *Springer*.

Naghib, A., Navimipour, N. J., Hosseinzadeh, M., & Sharifi, A. (2022). A comprehensive and systematic literature review on the big data management techniques in the internet of things. *Springer link*.

Noura, M., Mohammed, A., & Martin, G. (2019). Interoperability in Internet of Things: Taxonomies and Open Challenges.

Rogers, E. M. (2003). *Diffusion of innovations.* Free Press.

Sebastian. (2024). Why Django is the Perfect Framework for Your IoT Projects. *clouddevs*.

Somekh, B. (2006). *Action Research, a methodology for change and development.*

Wang, L., Zhenxi, H., & Fagui, L. (2019). Energy-Efficient Collaborative Task Computation Offloading in Cloud-Assisted Edge Computing for IoT Sensors. *Recent Advances in Fog/Edge Computing in Internet of Thing*.

Zilenkov, A., Dinis, D., Ilya, M., & Yanina, K. (2018). *Power line communication in IoT-systems.*

APPENDICES

APPENDIX A: ARDUINO CODE (PART OF SMS GATEWAY)

```
#define SIM900 Serial2

#include <WiFi.h>

#include <HTTPClient.h>

#include <WiFiClientSecure.h>

#include <Arduino_JSON.h>



const char* ssid = "Home_Sweet_Home";

const char* password = "!Rehema!";

const char* serverName = "https://vicky243.pythonanywhere.com/gestion/messages/";

const long interval = 5000;

unsigned long previousMillis = 0;



String outputsState;

void setup() {

  Serial.begin(115200);

  SIM900.begin(19200, SERIAL_8N1, 16, 17);

  WiFi.begin(ssid, password);
```

```
  Serial.println("Connecting");
  while(WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
  Serial.println("");
  Serial.print("Connected to WiFi network with IP Address: ");
  Serial.println(WiFi.localIP());
}
void loop() {
  unsigned long currentMillis = millis();

  if(currentMillis - previousMillis >= interval) {

    if(WiFi.status()== WL_CONNECTED ){
      outputsState = httpGETRequest(serverName);
      Serial.println(outputsState);
      JSONVar myObject = JSON.parse(outputsState);

      if (JSON.typeof(myObject) == "undefined") {
        Serial.println("Parsing input failed!");
        return;
      }
```

```
    Serial.print("JSON object = ");
    Serial.println(myObject);


    JSONVar keys = myObject.keys();


    for (int i = 0; i < keys.length(); i++) {
      JSONVar value = myObject[keys[i]];
      Serial.print("GPIO: ");
      Serial.print(keys[i]);
      Serial.print(" - SET to: ");
      Serial.println(value);
      sendSMS(keys[i], value);
    }
    previousMillis = currentMillis;
  }
  else {
    Serial.println("WiFi Disconnected");
    digitalWrite(1, HIGH); // PIN 1 FOR AUTOMATION MODE
  }
 }
}
```

```
String httpGETRequest(const char* serverName) {
  WiFiClientSecure *client = new WiFiClientSecure;
  client->setInsecure();
  HTTPClient https;



  https.begin(*client, serverName);
  int httpResponseCode = https.GET();

  String payload = "{}";

  if (httpResponseCode>0) {
    Serial.print("HTTP Response code: ");
    Serial.println(httpResponseCode);
    payload = https.getString();
  }
  else {
    Serial.print("Error code: ");
    Serial.println(httpResponseCode);
  }
  https.end();
```

```
  return payload;
}
void sendSMS(String phoneNumber, String message) {
  SIM900.print("AT+CMGF=1\r");
  delay(100);







  // Use international format code for mobile numbers
  SIM900.print("AT+CMGS=\"");
  SIM900.print(phoneNumber);
  SIM900.println("\"");
  delay(100);

  SIM900.println(message);
  delay(100);
  SIM900.write(26);
  delay(100);
  delay(5000);
}
```

APPENDIX B: COST ESTIMATION

1. Hardware Cost

Components	**Units**	**Price/unit (RFW)**	**Total (RWF)**
ESP32	2	15500	31000
GSM Module	1	22000	22000
Relay	5	3000	15000
Connectors & wires	Function of meters	-	~8000
		TOTAL	**76000**

2. Software Cost

Types	**Elements**	**Prices (RWF)**
Hosting	Web Hosting	13800 per Month
	Database &Media Storage	7000 per month
Design & Firmware tools	Circuit design (EasyEDA)	Free
	Drawings (draw.io)	Free
	Arduino IDE& VSCode	Free
	TOTAL	**20800 per month**

3. Other utilities

Type	**Cost (RWF)**
Enclosure	5000
Soldering wires & accessories	10000
Transport	40000
Workforce	100000
Total: 155000	