Kingdom of Saudi Arabia
Ministry of Higher Education
Imam Abdulrahman Bin Faisal University
College of Computer Sciences & Information Technology

My Smart Safe Home

Design and Implementation of an Automated IoT-Based Smart Home System

A project submitted
in partial fulfillment of the requirements for the degree of
Bachelor of Science in Computer Science

by
Hesham Abed Almusllam (2150006030)
Khalid Alshahrani (2170000575)
Ahmad Saad Alshobaiki(2180001707)
Abdulaziz Ahmed Almuqrin (2170006529)
Saif Abdullah Alshiban (2180002659)

Supervised by
Dr. Farmanullah Jan

May 2022

DECLARATION

We hereby declare that this project report entitled "**My Smart Safe Home**" is based on our original work except for citations and quotations, which have been duly acknowledged. We also declare that it has not been previously or concurrently submitted for any other degree or award at any university or any other institutions. This project work is submitted in the partial fulfillment of the requirements for the degree of Bachelor of Science in Computer Science at Computer Science Department, College of Computer Sciences and Information Technology, Imam Abdulrahman Bin Faisal University.

Project Team:

Group Number: 5		
Team Member's Name	Signature	Date
Hesham Abed Almusllam	Hesham	December 9, 2021
Khalid Alshahrani	Khalid	December 9, 2021
Ahmad Saad Alshobaiki	Ahmad	December 9, 2021
Abdulaziz Ahmed Almuqrin	Abdulaziz	December 9, 2021
Saif Abdullah Alshiban	Saif	December 9, 2021

Project Supervisor:

Name	Signature	Date
Dr. Farmanullah Jan	farmanullah Jan	May 16, 2022

ACKNOWLEDGEMENT

First, we would like to express our sincerest gratitude to Allah for giving us the knowledge and strength to push through every struggle, and for blessing us with all the people that makes life more meaningful.

We also would like to thank our instructor **Dr. Farmanullah Jan** for supervising us and for being constantly patient and considerate of everyone. We would also like to thank **Dr. Dhiaa Musleh and Dr. Khansa Abdullah Al Mofeez** for going out of his way to extend a helping hand by giving us the necessary information to ensure that we are fully ready to undertake the project.

Last but not the least, nothing but gratitude to each and every member of the team for their cooperation, and for being reliable by having and covering for each other throughout the whole semester.

ABSTRACT

The Internet of things (IoT) refers to the interconnection of the electronic devices beyond the computers and smartphones that exchange data with other devices over the internet or other communication systems. The IoT bridges the gap between the living world and the machine world. With the intent of making the living experience more convenient and secure, this paper talks about the implementation of an IoT concept, i.e., a smart home. A smart home is an IoT application that integrates the use of embedded microcontrollers, sensors, and actuators to automate the appliances used in everyday life and thus provide a better and more comfortable living experience for everyone.

In addition to motivation, goals, and the scope of the study, this paper has all the necessary software engineering documents needed to fully understand the management plan, system requirements, and software design to implement the proposed system.

Table of Contents

LIST OF TABLES

LIST OF FIGURES

Chapter 1: Introduction

1. Introduction

With the advent of the communications technology, the automation systems have gotten a lot of attention from the relevant community around the world. A notable application is the so-called Internet of Things. As mentioned earlier, the Internet of things (i.e., IoT) refers to the interconnection of the electronic devices beyond the computers and smartphones that exchange data with other devices over the internet or other communication systems. Current example of such systems are the smart homes, smart cities, smart security systems, smart cars, smart factories, autonomous agriculture, and among others.

In this project, we will design and implement a smart home (SH). It is an IoT application that allows users to remotely manage and monitor household equipment. An SH is a home with an automated system that includes sensors, actuators, and controllers to improve comfort, automation, safety, and security for occupants. Smart gadgets, such as smartphones, smart televisions (TVs), smart washing machines, smart refrigerators, and smart sensors, have become ubiquitous in people's everyday lives in the modern era. These intelligent gadgets can communicate and interact with one another to create a smart environment effectively.

To manage communication between the smart devices within SHs, an automated system should be built. Many automation systems have been created, and some are currently being sold as the commercial goods as well; of which, some of these items are used to manage household appliances remotely or locally.

2. Problem Statement

The cost of the electronic control has quickly reduced since the introduction of the microcontrollers (also called dedicated microprocessors) at the turn of the century, and home automation has developed. Home automation systems are not commonly used, and they are still seen to be at the realm of enthusiasts or the wealthy.

Various automation technologies are utilized to equip household appliances with the needed systems for simple monitoring and control, such as remote control for TVs, fans, air conditioners, and music players. With recent increase in the usage of the electricity in homes and the quick progress of information technology, a new age of home appliances' control has begun, utilizing the mobile devices with short-range communication interfaces such as the well-known buddy (i.e., WIFI). All these systems and technologies are beneficial for controlling home appliances from within the house, but they do not allow people to monitor and manage their houses from outside. Although such systems always allow contact with inhibitors to promote ease, comfort, safety, and energy efficiency, their usefulness and communication range are limited. Most of these systems don't make use of the IoT's strong characteristics, which seeks to connect billions of smart devices (e.g., phones, computers, sensors, and actuators) to the internet. Our goal is to build a smart home prototype utilizing the capabilities of commonly available IoT devices.

3. Motivation

The reason behind choosing this topic is to make our homes more functional, secure, and energy efficient. Connecting our home appliances to each other and to our smartphones allows us to control, monitor, and interact with them. Creating our own smart home ecosystem can enhance our lives drastically.

From monitoring the power usage per device, turning off the lights remotely, to the possibility of integrating artificial intelligence (AI) that could manage the home ecosystem, the idea of SH is a huge improvement that would change people's living experiences by more.

4. Justification

The reasons why this project should be as proposed and not in another way around or even choosing another topic entirely is as follows:

- *The project is within the scope of the academic domain,*
- *The topic is shaping up to be a large field of research and development and will be very important in the near future, and*
- *By working on this project, we will further sharpen our IoT relevant skills necessary for the future work and probable research.*

5. Aims and Objectives

For this project, we are going to create a smart home using IoT devices and the corresponding software platform. The project will be composed of the design and implementation of the smart home integrated with a microcontroller serving as the main controller of all other devices available at home, such as the sensors and actuators. To undertake the task, we aim:

- *To build a smart home prototype,*
- *Integrate the smart home prototype with IoT devices,*
- *Use a server to host smart home data,*
- *Create a user-friendly mobile interface, and*
- *Write software code.*

➢ **Summary of Project Deliverables**

- **Project management-related deliverables:**
  - Software project management plan (SPMP).
  - Software requirements specification (SRS).
  - Software design specification (SDS).
  - Final report.
- **Product-related deliverables:**
  - A prototype for a smart house design.
  - Microcontrollers, sensors, and actuators.
  - A server to host smart home data.
  - User-friendly mobile interface.
  - Software code.

5.1 Scope / Limitation of the Study

The scope of this project is to build and show a Smart home prototype embedded with IoT devices and not to build industry-grade devices. Limitations include time and budget constraint.

5.2 Significance of The Study

The topic is shaping up to be a large field of research and development and will be important in the future. Additionally, the country aims to make use of IoT and Smart Home technologies in their Vision 2030 program.

6. Project Organization

This report is organized into multiple sections:

Chapter 1: Introduction - This section supplies an overview of the purpose, scope, goals and other related details to the project. Its aim is to supply context to what.

Chapter 2: Background and Review of Literature – This section will have write ups of literatures related to the study used as references to further understand the topic.

Chapter 3: Methodology / Requirement Engineering – This section will have specifications on the method employed to work on the project and the requirements needed.

Chapter 4: Design – This section will have specifications related to the design of the project

Chapter 5: Conclusion – Final closing remarks.

Chapter 2: Background and Review of Literature

1. Introduction

This section offers contemporary literatures related to the study used as references to further understand the topic if necessary. All references are enlisted at the end of this document.

2. Literature Review

A low-cost, low-power SH implementation using IoT was proposed in [4]. The presented SH uses embedded systems, 3G technologies, and ZIGBEE technologies as a solution to the problems faced by the current SH systems such as discrete functions, poor portability, and weak updating capability. They used the Samsung S3C2440A microcontroller as a gateway device because of its low-power consumption and high-performance capabilities. The Samsung S3C2440A also minimizes the overall system's cost by eliminating the need to configure additional modules as it provides built-in interfaces of common system components.

The ZIGBEE communication protocols is then used to create their own local home network in the mesh-network topology, as opposed to the usual star-network topology. The MAXSTEAM XBEE-PRO(S2) is integrated to the devices to create a wireless network of devices because it is a low-cost and low-power device. The microprocessor is then connected to a remote management platform using a 3G interface module, TDM330. The interface was chosen after considering portability, security, and ease of use. Once connected, data can then be remotely processed and transmitted by the remote management platform only after a verified user logs in. Overall, the SH design was excellent as it took into consideration the cost and power consumption of the whole system while using innovative techniques such as integrating ZIGBEE network protocols to enable wireless communications between the embedded devices, as opposed to more traditional techniques.

In[6], a low cost WIFI based automated SH was proposed. The study used an Arduino Mega Microcontroller along with a WIFI module chip ESP8266 to facilitate wireless communications on the Arduino Mega, to build the home automation system. Additionally, several sensors were utilized, such as a temperature sensor, a humidity sensor, a motion detector sensor, and more. These sensors are connected to the Arduino Mega Microcontroller. The sensors' role is to gather data for the Microcontroller to read as input or act upon as output and perform an action to the respective connected appliance. More specifically, the Microcontroller can receive and transmit serial data, trigger an interrupt on a low value, provide 8- bit PWM output and more.

In addition, they also used a relay board to serve as an actuator between the Arduino Mega and the connected appliances in response to an input signal. On the software part, they used a Wi-Fi based android application, Virtuino, for remote control of the system as it has a user-friendly interface and works efficiently with Arduino Mega. The Arduino Mega is programmed to interact with the Virtuino application.

After the implementation, the results of the proposed low-cost Smart Home were then verified and validated. They were successfully able to control and monitor the SH using Virtuino. The electrical lights can be switched on and off using the mobile app. Also, the temperature and humidity sensors were able to capture precise values in the SH and display it in the mobile app. With these parameters, it made it possible to further automate the SH system such as switching a fan or an AC on or off depending on certain conditions. Furthermore, the motion sensor was able to detect any movement in an area and activate a buzzer. This can also be then automated further to create a more functional security system.

Ultimately, the proposed SH system was implemented successfully. However, there are a few limitations to note. The microcontroller that was utilized in this study does not have a built-in Wi-Fi interface, therefore, to communicate with the Arduino Mega, devices must be connected locally. Unless of course as mentioned in the paper, they used an extra WIFI module chip (ESP8266) to facilitate wireless communications with the Arduino Mega. Using an additional chip resulted to the system costing more and consuming more power, when compared to using a microcontroller with a built-in WIFI interface. Another limitation is that while the implementation is successful, the system is not fully automated. There is no action to switch on and off appliances based on the captured data from the temperature and humidity sensors. There is also no action to open or close doors (as a basic security measure), which can be done in accordance with the motion sensor.

The lack of IoT technology usage, unfriendly UI's, limited wireless transmission range, and high costs are the main motivation of paper [8]. They presented an economical IoT-based home automation system with a friendly user interface. They utilized NodeMCU microcontroller as the gateway to connect different sensors and transmit data to Adafruit IO, a cloud server. The data can then be accessed using If This Then That (IFTTT) remotely. The appliances are then controlled using a set of relay modules. The devices are all placed deliberately in a well-planned home prototype. The system is designed with cost, energy-consumption, convenience, safety, and security in mind.

The idea was implemented following a step-by-step conceptual framework. They modelled the system by considering the layout and structure of the SH prototype and analyzing the cost of selected components. The components used were: NodeMCU ESP8266 microcontroller board, 12V DC Power Source, 8-channel 5V DC relay module, HCR04 ultrasonic module, PIR motion sensor, DHT11 temperature and humidity sensor, MQ2 gas sensor, LED bulbs, Mini fan, 1v5V DC motor, RFID, and 12 V DC brushless submersible water pump.

The following software components are also used: NX10, Adafruit.IO, Arduino IDE, MQTT protocol server, and IFTTT. The SH was then designed using the NX10 Software and built with plywood. The microcontroller, sensors, and actuators used were installed in predefined places to sense the considered stimuli and act on it accordingly. The wiring of lights and measuring devices were arrayed on the ceiling and sides of the SH prototype. The NodeMCU microcontroller has a 32-bit TensilicaXtensa LX106 core clocked at MHZ. It has a built-in Wi-Fi interface and can run self-contained applications, hence why it was chosen. Several actuators were used to interact with the devices physically.

On the software side, Adafruit.IO is then used as the IoT server for the system to monitor and control appliances through the Internet with the MQTT protocol. MQTT is a TCP-based publish messaging protocol designed for lightweight machine-to-machine communications. Lastly, Arduino IDE is then used to write software code. After the implementation, the project is then enhanced and optimized whenever an error was found to further improve the performance of the system. Finally, the proposed idea was implemented fittingly. The concepts discussed and the proposed solutions were all tackled and solved adequately. A very clear, concise, and informative paper.

In [1] A home automation system method was used with IoT, which entails remotely supervising and managing home appliances IoT. They use electronic devices to control basic home appliances and make them function just by interacting with the user interface in their devices. This sort of house is often referred to as a smart house. They created a system as well as an html website that allows us to control the entire communication system. The language used was C, and the username and password were supplied so that when the communication board was started, it immediately connected to the device’s hotspot.

The approach employed was:

- *They utilized an Arduino development board, with an in-built Wi-Fi and is used to control the communication board,*
- *When they plug in, the board initializes all the components over there, and then it receives a signal to check if the given device's hotspot is available or not; if it can't find the device, it restarts the initialization process, but if it does, it connects to the local WIFI, and.*
- *Once the board has been connected, the order to read data from the server is issued.*

The designed scheme:

- **Optocoupler and Triac Circuit**: The microcontroller is connected to the circuit through the driver. The infrared ray receives 0 or 1 input from the microcontroller (IR). When a ray strikes a photodiode, it becomes active and forward biased, causing current to flow across the circuit to the terminals T1 and T2. The photodiode is engaged, and the gate terminal triggers when the input is set to 1, and T1 and T2 are connected at the same time, resulting in a closed circuit. When the input is set to zero, the photodiode operates in reverse bias, resulting in an open circuit.
- **Methodology**: The microcontroller and all connected devices as well as a UI from which we can control the whole communication board.

The communication board starts and initializes all the components over there as soon as they plug in, and then it receives feedback to check if the given device's hotspot is available or not; if it can't find the device, it restarts the components, but if it does, it connects to the local Wi-Fi. After, the order to read data from the server is issued. Then it decides if the preset value is 0 or 1. If the value is 0, it returns to "reading data from the webpage," and if the value is 1, users may manage their home appliances through their device.

- **Experiment Specifics**: They planned a Bluetooth-based system to control household equipment. However, if they developed a project that relied on Bluetooth, the product would be basic and restricted in scope for future growth. With the aid of Internet of Things ideas, they utilized Arduino as a development board. As the realm of automation and IoT expands, so does the breadth of research and development. The use of Arduino and the Internet of Things (IoT) would enable them to conduct extensive and efficient research for their idea.

The Arduino Wi-Fi module is a simple but efficient development board that is simple to install with only a basic understanding of embedded C. They have chosen Arduino as their microcontroller platform. After that, they learned how to utilize embedded C by browsing the internet. The Arduino IDE is an open development platform for exchanging microcontroller-related knowledge and information, as well as coding. They also used YouTube to get a clear concept of how to code in embedded C. With the resources and expertise listed above, they set up the system to operate the appliances in the house. The only thing left was to produce the required result.

In [7] the system architecture is about SWO-based Analytic Platform. SWOAP is a SWO-based analytic platform. SWOAP is a management platform made up of a resource core layer and a knowledge-building framework layer. Through the management platform, the resource core layer collects smart appliance and legacy appliance information. Each resource is generated using the 1st layer, and connections between them are established. This is how we plan to use various data. The knowledge-framework Layer analyzes home data based on the resource notion.

- **Resource Core Layer:** This layer is responsible for managing data utilized across the platform. The data is divided into domains and abstracted as a resource. Each domain has its own data structure in the resource. One of their platform's advantages is the ability to create resource-relationship information. Internal analysis creates implicit knowledge in the form of relational information.
- **Knowledge Framework Layer**: Model-driven analysis is carried out using this paradigm. They suggested a PMML-based model description schema (Predictive Modelling Markup Language). PMML is the industry standard for statistical and data mining models. It's simple to create a model on one system with one application and deploy it on another system with another application using PMML.

- **SWOA Protocol**: Developed a technique for aggregating device information in a series. SWOAP sends a request to the administration platform for all the home site ids. Then, to obtain device information for each house, SWOAP sends all the device ids to the management platform. Finally, SWOAP uses the device id obtained in the previous step to request device information. In this study, the APIs were created using a standard specification based on JSON. Real-time event reception in a web context is one of the most difficult challenges. Polling API, web socket, WS-event, and other methods are utilized, but they are unreliable.

- **Testbed and test scenario:** They installed smart metering devices in ten houses in order to collect genuine data. The smart metering gadget is then connected with a rice pot. By metering agent, the smart metering gadget was renamed SWO, and they regularly broadcast their basic description and power usage statistics. Based on the home address, the platform has at least 10 device domain resources and 10 physical space domain resources, and the power use data has become numerous content resources. Furthermore, the platform creates relation objects between them. For getting aggregated resource data, sensory data for resources, and so on, they provide web service APIs.

In reference [2] Microgrids or nano grids, intelligent loads in smart homes (IoT enabled loads), electric vehicles, and distributed energy storage systems are all being fueled by IoT devices and technology (for example, IoT sensors). Customer roles are changing as a result of this growth. To suit their demands, customers can now generate electricity locally from renewable sources and engage in power exchanges with the electric power network. Customers may employ Internet of Things sensors to offer relevant power generation and consumption data to intelligent loads, allowing them to use power more effectively, decrease power waste, and regulate expenses. Distributed energy storage solutions beat centralized energy storage systems in terms of flexibility, control, scalability, and dependability.

To account for any changes in power supply from renewable sources, distributed energy storage devices, such as batteries and electric vehicles, are essential. If production falls short of demand, rechargeable batteries can help make up the difference. Excess energy from renewable sources may also be stored in rechargeable batteries or sent into the electric power grid. Reduces the danger of harmful environmental exposures including carbon monoxide and smoke, as well as making home life more easy and pleasant.

Security Considerations in [5] With the proliferation of smartphones, a plethora of fixed appliances have been established to offer a variety of functionalities. Smart homes, intelligent cities, smart health care, and smart car services have received a lot of attention in recent years. As a result, numerous sensors, small, rooted devices, and home machines have been analyzed and utilized constantly by several companies, universities, and research institutions. They were then increasingly sensible in order to give logical services to users. Nevertheless, In the IoT (Internet of Things) environment, an increasing phenomenon of user privacy data leakage and security vulnerability should not be overlooked. If the service infrastructure is designed without taking into account predictable security flaws, user privacy data leakage, social infrastructure paralysis, and economic losses, as well as a risk to human life as a severe case, can occur as a result of malicious outsider attacks.

To determine the service organization that supports security elements, the proper security features required for each part of the servicing infrastructure must be defined. Furthermore, several services necessitate a distinct defense issue that is tailored to individual qualities. As an example, data (e.g., user's private information) security is essential for intelligent traffic service and intelligent medical department, for the smart city and automated farm services, the authentication scheme is more important. As a result, we must carefully examine the security requirements and necessity of a specific IoT service. the security requirements for the home automation service, which the smart home service, which has emerged developed as a hot item among major IT firms, and particularly explains a security element for each section of the smart home-based system in the IoT ecosystem.

For the IoT energy demand in [12], Energy disaggregation of demand is an important step toward the management of energy as it enables feedback to the user regarding energy usage. It allows the user to receive recommendations, based on this, steps could be taken to reduce consumption. The energy disaggregation model is currently being developed by training a machine learning model. Energy disaggregation is like a state estimation problem in that we deduce the power states of the electrical system's appliances from the output, which is the aggregate power. It is critical to have an accurate model of a system to estimate its states. The use of a sequential energy demand model of appliances as the basis for the energy disaggregation model is a significant contribution of this work.

Internet of Things (IoT) is a new technology that is transforming our world. Without the Internet of Things, the concept of a linked world would be impossible to comprehend. The FLIP (Frugal Labs IoT Platform) is used to create IoT-enabled Smart Homes. The features of the Smart Home and their applications are discussed, as well as the FLIP architecture to implement the Smart Home. This paper's proposal is utilized for observing and regulating the ecosystem of the Home Automation. Urban areas in India will soon be transformed into smart cities as a result of the digital India initiative. An environment where communication and services are heavily reliant on the Internet is a smart city. A smart home system is part of a smart city's infrastructure. A smart home is built on top of IoT infrastructure[9].

2.1. Smart Home Functionalities

1) *Notifications*: Can sense its surroundings and send alerts to the user as needed. The alert is connected to environmental data. This data might include the concentrations of various gases in the atmosphere, as well as temperature, humidity, and light intensity.

2) *Monitoring*: The most essential function. With the aid of numerous sensors and video feeds, a smart house can monitor its surroundings. Monitoring is a vital role since it keeps track of every activity in a smart home, which is the fundamental necessity upon which any subsequent action or choice may be made.

3) *Remote control*: - the SH should allow user to control different activities. Such as switching lights, air-conditioner, and appliances on/off, lock/unlock doors, open/close windows, etc.

4) *Intelligent system*: - Intelligence is the most significant function of smart home. Integrating AI to the smart home does not only give brains to the smart home but is also very important for a security point of view.

2.2. Smart Home Applications

While the scope of a smart home's applications is unlimited, this paper shows a few of them, which are detailed below.

1) *Efficient Lighting***:** Used to save energy by adjusting lighting to different circumstances, therefore minimizing needless energy use.

2) *Smart Appliances***:** Used to acquire appliance status information and to control appliances remotely. It is also used to schedule chores for a certain time and to integrate appliances in real time.

3) *Security***:** Is used to notify users when a security breach has occurred. The user can also receive a full report from the intrusion detection application. The main objective of this software is to monitor suspicious behavior in a smart home, warn the user, and take the appropriate security measures.

4) *Smoke/Gas Detection***:** This application senses the smart home environment for good health, secure Optical detection, ionization, and air sampling techniques are all implemented in this application.

2.3. Flip Architecture

It is an open source IoT platform aimed at anybody interested in learning and working with IoT to turn their concept into a "Proof of Concept" created by Frugal Labs in Bangalore, India. FLIP is a whole IoT platform. App, SDK, device gateway, bridge and storage are the four levels of the FLIP architecture.

The cloud layer comprises of the database and the broker. The broker establishes connections with all the devices and stores the data that is received from the devices in a database. For backend processing, the cloud layer includes three major structures: a Mosquito MQTT protocol broker, a Mongo DB database, and Node.js.

App & SDK tier is the top level. The software, which is made up of a web app and a dashboard, is used to visualize data using widgets and graphs. The rule engine of the SDK is written in Python. The Python SDK may be used in two ways. The first is to define logic for your device, i.e.

2.3. The Suggested System

The FLIP architecture was used in the proposed system in this study. There are four primary application modules in the proposed system. Smart lights and appliances, intrusion detection, and smoke/gas detection displays are all examples of smart technology. Temperature, humidity, light intensity, and motion detection sensing are all considered in the basic device setup for smart home lighting management.

FLIP gadget is integrated to sensors, lighting, air conditioner, camera, windows and door system, in the planned smart home system. The gateway connects the flip gadget to the Internet. The proposed smart home network's gateway plays an essential function since it adds an additional security layer to the smart home network, making it more secure. The proposed smart home system can monitor the environment from a remote location, produce alerts and notifications if certain terms are fulfilled, adjust room lighting and temperature automatically.

The Internet of Things is a network of computer systems and physical things with which humans interact. Things can be actual, virtual, stable, or movable, but they will always communicate with one other. This is known as a form of communication between objects. The internet of everything is a reality that allows objects to setup themselves without the need for human involvement. IoT may be used in a variety of settings, including industries, labs, and homes. With changes in socioeconomic structure, family structure, information technology development, and human security of the home environment, comfort, higher efficiency requirements, and growing demand for home automation, home automation is becoming more popular[3].

All Sensory and Relay Hardware, as well as User Interface and Processing Hardware, make up the Smart Homes System. All the devices will be connected to a smart central controller, which will be connected to a switch module for each connected device, allowing us to access each device individually. This device will also be linked to the router for internet access, allowing us to interact with the user if permission is ever necessary. Kang Bing describes the IoT in Smart Homes System as a layered design[4].The layers are as follows:

1) **Sensing Layer**: This is the layer that collects data from the environment as well as household appliances and gadgets. The data is subsequently sent to the middle layer,

2) **Network Layer**: This layer sends the data it receives to the top layer (Application layer) for sorting and processing, and

3) **Application Layer**: This layer does the actual analysis, with the results being used by triggers to initiate a specific action.

Energy Efficiency Currently in Use The systems function on the basic principle of conserving energy by considering two essential parameters: the presence of people in the house and the temperature. External (weather) Conditions in the Area all gadgets that give security features to our home are included in the security system. This method protects the safety of everyone who lives in the house. Cameras, infrared sensors, magnetic door sensors, smoke sensors, gas sensors, and automatic doors and windows are among the security systems utilized.

All the above-mentioned automated doors and windows are passive devices, meaning they do not communicate information but react to information provided by the controller (such as opening a specific window or door).

Other devices, on the other hand, are all sensors that detect a certain thing/object and after cause suitable modifications in the respective devices' interfaces. Interface While security and energy economy are vital, a nice user interface is also essential. Because the user interacts with the entire house, we must ensure that he has access to everything he wants at any time.

The user interface, which is one of the most essential parts and the only portion visible to the user, conducts this. Because the user interacts with the system through the interface, it is a key part of the system. It informs the user and is also used to adjust the settings of a certain gadget, as well as helping us in our daily activities a person's way of life daily.

Smart home automation concept is an enhancement to normal life experience, in this paper[13] its was discussing the implementation of a smart home with a alarm system for security, the alarm would go off by actuators that are connected to the micro controller , these actuators are infrared sensors , door magnetics and emergency button and many more, these inputs would alarm the system which will analyze the warning then would make an appropriate solution for it like turning off the electricity or telling the user about a gas leak.

The system will have RS-485 and GSM technologies to monitor alarms remotely, the RS-485 will act as a medium or as a relay which will be the center of the system structure, the hardware design will use UART interface along with LAN interface , A/D converter and IO interfaces, the operating system used in this is RTX which will be installed on LPC2378, as stated in this paper [13]the bus will be connected to ethernet module ,which will act as medium between the bus and the remote for data exchange and sending commands, which will lead to a successful communication.

Home automation can use many communications to connect home appliances to smartphones such as Wi-Fi, NFC and Bluetooth, in this article [10]a Bluetooth based home automation was implemented, the reason behind choosing Bluetooth in this system is because it has low cost and easy installation process, it was also proven that Bluetooth is faster than GSM and Wi-Fi since Bluetooth has a maximum range of 100 meters, thus communications is faster in short ranges.

Arduino uno is the microcontroller chosen for this system, it was connected by some actuators which are an ultrasonic sensor (HC-SR04), soil moisture sensor, a Bluetooth module (HC-66) and home appliances, the HC-SR04 will be used to measure water level in the water tank to check if its full or empty, the soil sensor will be used to check the water amount in the soil to manage a garden.

In the article [10] the communication of Arduino along with the other devices is divided into two functionalities which are: control and monitor ,these commands are accessible in the software interface provided by the Bluetooth module, choosing monitor will allow the user to check on the installed sensor such as water level or soil moisture , while choosing control will show accessible home appliances such as air conditioner , Television and light system.

Both articles discussed in this literature review[10][13]talked about enhancing the smart home experience by implementing home automation system that can connect the house , office or garden to Wi-Fi , NFC or Bluetooth , in [13]the usage of the bus connected the microcontroller to the rest of the home appliances and actuators which created a smart ecosystem the communicate through GSM and WIFI.

LPC2378ARM was used as a microcontroller, since it contains a strong and suitable hardware interface, then RTX was installed on the microcontroller to calculate data coming from the appliances.

In [10], the system was implemented through an Arduino uno, which was connected through a Bluetooth interface that handles communication between devices, the connection is done by transmitting ASCII data from the phone, through the module and then to the board or vice versa, the ASCII data represent a specific function in the system which is shown in the table below.

Table 1 ASCII Operation from: Smart home automation. Smart home automation system using Bluetooth technology

ASCII Data	Operation
0	Turn ON light 1
1	Turn OFF light 1
2	Turn ON light 2
3	Turn OFF light 2
4	Turn ON FAN
5	Turn OFF FAN
6	Turn ON air Conditioner
7	Turn OFF air Conditioner
8	Turn ON television
9	Turn OFF television
S	Check status of all appliances & sensors
W	Check status of water level indicator
M	Check status of soil moisture sensor

This literature review is going to discuss an implementation of an Arduino based home automation system[11].The system will have two main components, the hardware which consists of the Arduino and the sub modules that completes the system in term of functionality and the android application, which is the software that will act is the main interface for controlling and monitoring the hardware system and its modules. Communication will be done via Bluetooth, since the system will be in a home or an office environment, it will be mostly short ranged, so Bluetooth can be utilized in these kinds of environments, because Bluetooth connection is effortlessly instantaneous and reliable.

The microcontroller used in this implementation, is the Arduino Uno, this microcontroller supports connection to Bluetooth via special module (HC-05), the Arduino is integrated with ATMEGA 328 processor for high performance and low power consumption. The Arduino will be connected to two LDRS (Light Dependent Resistor), and between them there will be a 10K and a 20K ohm resistors, LDRS will act as a sensor inside the house to detect how many people are in a particular room, the Bluetooth module (HC-05) will act as interface for communication, the module has six pins that will connect to the Arduino:

1. *VCC pin will connect to 5V pin in the board,*
2. *GND will connect to the ground, and*
3. *TXD and RXD pins will connect to RX and TX pins in the board respectfully.*

The STATE pin used as feedback to check if Bluetooth is working properly, and the KEY pin is used to set data mode to low, a temperature sensor (LM35) will be connected to the Arduino to monitor temperature in the home, it has three pins:

1. *Vs pin will be connected to the Arduino pin (5V).*
2. *Vout pin will connect to Arduino pin (Analog).*
3. *GND will connect to Ground of the board.*

Software programming of the board was done in Arduino IDE, via C / C++ languages through Arduino IDE integrated development environment, the code contains two parts which are setup and loop which will:

- **Setup**: this function will be called once to declare pins to variables.
- **Loop**: this function will be called repeatedly until power shut off the board, this function will have all code that needs to be repeated such as switch on/off the temperature sensor.

The last stage of development is testing, the purpose of this crucial step is to find errors, faults and trying to correct them, such as checking if there's an established connection via the Bluetooth module via STATE pin led, AN ohmmeter was used to check for proper resistance in the circuit and to ensure secure connection for a circuit without shorting, this includes checks for voltage and wattage so the power to fry the modules connected to the board. Last tests include checking for connection between the android app and the board, TYPE commands are used to send data to the board, the board would reply with task executed or invalid command based on the command.

3. Comparative analysis

Literature Review	*Indoor control*	*Outdoor control*	*Security*	*Safety*	*Monitoring*	*Energy management*	*Wireless interface*	*Controller*	*Real implementation*	*Smart phone*	*Web based*
Jabbar, W. A., Alsibai, M. H., Amran, N. S., & Mahayadin [6]	✓				✓		Wi-Fi	Arduino mega	✓	✓	✓
Muhammad Asadullah [10]	✓				✓		Bluetooth	Arduino Uno	✓	✓	
Bing, K., Fu, L., Zhuo, Y., &Yanlei [4]	✓		✓	✓	✓		MAXSTREAM XBEE-PRO(S2), Ethernet, 3G	SAMSUNG S3C2440A			✓
Zhangjinglu, Chenlili.[13]	✓	✓	✓		✓		Wi Fi	philips LPC2378ARM	✓	✓	
P. U. Okorie, A. A.[11]	✓		✓		✓	✓	Bluetooth	Arduino uno	✓	✓	
Bedi , G., Venayagamoorthy , G. K., Singh , R., Brooks , R. R., & Wang. [2]	✓		✓			✓	WiFi/Bluetooth				✓
Han, J.-H., Jeon, Y., & Kim, J.[5]	✓		✓	✓	✓		Wi Fi/ Ethernet	PC sever			✓

Raiker, G. A., B., S. R., Loganathan, U., Agrawal, S., Thakur, A. S., K., A., . . . Thomson, M. [12]	✓		✓		✓	✓	Wi Fi/ Ethernet	PC sever	✓	✓	
Agarwal, K. a. [1]	✓	✓	✓	✓	✓	✓	Bluetooth and Wi-Fi	Arduino BT board	✓	✓	✓
MSSH	✓	✓	✓	✓	✓		Wi-Fi	Arduino	✓	✓	

Chapter 3: Software Project Management Plan

Revision History

Name	Date	Reason For Changes	Version
All members	Sep 30, 2021	Prepared initial version	0.1
All members	Oct. 30, 2021	Continued writing the SPMP and worked on unfinished task	0.2
All members	Dec. 9, 2021	Complete review - Final version	1.0

1. Project Overview

This section of the provides an overview of the purpose, scope and objectives of the project for which the Plan has been written, the project assumptions and constraints, a list of project deliverables, a summary of the project schedule and budget.

1.1 Purpose, Scope, and Objectives

- The purpose of this project to design and implement an automated Smart Home using the Internet of Things technology. We will meet our project criteria by achieving the planned outcome and result of the project, which is a fully functional prototype of a smart home.
- We will implement this project with safety and convenience under consideration. However, we will not consider artificial intelligence, as it could get in the way of our main idea which is a fully functional smart home based on hardware and software.
- The objectives of our projects are as listed:
    1. *Design and build a fully functional prototype of a smart home.*
    2. *Implement low energy and resources saving techniques.*
    3. *Integrate security functions and measures.*
    4. *Connect hardware and software via cloud computing.*
- This project is related to Internet of Things and Smart Automation Systems. A field of interest that is currently a growing market and has many domain applications. A variety of research papers have also been published regarding many different applications of the field.
- The Smart Home project can be integrated with other Smart Automation Systems, such as smart appliances that was developed independently.
- We will provide the Software Requirements Specification (SRS) for a more official statement of project requirements.

1.2 Assumptions, Constraints and Risks

- This project will be implemented under the assumption that the smart house will contain a variety of features such as a smart lighting system, smoke and fire detectors, and some doors and window detectors for security purposes, the home will also employ some form of energy-saving technology.
- The imposed constraints are as listed:

    1. *schedule,*
    2. *budget, and*
    3. *resources,*

- The possible risks for this project are as listed:

    1. *Device hijacking,*
    2. *Data siphoning,*
    3. *Denial of service attacks,*
    4. *Data breaches,*
    5. *Device theft,*
    6. *Man-in-the-Middle or Device "spoofing", and*
    7. *Power shortage*.

1.3 Project Deliverables

- The deliverables of this projects are as listed:

Project management-related deliverables:

1. Software project management plan (SPMP).
2. Software requirements specification (SRS).
3. Software design specifications (SDS).
4. Final report.

Project-related deliverables:

1. A prototype for Smart House Design.
2. Microcontrollers, sensors, and actuators.
3. A server to host smart home data.
4. User-friendly mobile interface.
5. Software code.

- Each requirement will be submitted according to the specified dates in the syllabus.
- Software requirements will be submitted both online and physically to our supervisors.
- Hardware requirements will be submitted physically.
- The delivery media will be through email.

1.4 Schedule and Budget Summary

- The schedule for the project documentation starts from 29th August until the end of the 1st semester, 5th December. The final report will be delivered by 9-12-2021. No budget will be provided.
- The work schedule will be done in 2 parts. The completion of documents and all required paperwork will be done during the 1st semester of the academic year. The implementation of the project will be done during the 2nd semester of the academic year.

1.5 Evolution of the Plan

The structure of this Project Plan follows the recommendations of IEEE Std 10581998.

- Any unscheduled updates to this plan must be declared to all the team members.
- The SPMP for this project will be under control and monitored, so that any changes made in the plan will be updated and available to the team members. There are weekly meetings with team members and the supervisors to review the project management process and discuss the updates on the project. In case of changes, the current and future documents will be updated simultaneously and will be available to the team and supervisors.

1.6 References

- IEEE Standard for Software Project Management Plans
- IEEE Recommended Practice for Software Requirements Specifications
- IEEE Standard for Information Technology—Systems Design— Software Design Description

1.7 Definitions and Acronyms

Table 2 Definitions and Acronyms

Acronyms	Definitions
IoT	Internet of things - describes a network of everyday things or appliances that communicate with each other.
SH	Smart Home - An application of IoT where everyday home appliances is connected to a network to improve a person living experience
MC	Microcontroller – a device capable of receiving, processing, and transmitting data, and is often embedded into other devices.
Waterfall model	The waterfall model is a sequential software development process in which progress is regarded as flowing smoothly downhill through phases such as conception, initiation, analysis, design, construction, testing, and so on.
Arduino	It's a free and open-source electronics platform with simple hardware and software. Arduino boards can detect inputs such as light on a sensor, a finger on a button, or a Twitter post and convert them to outputs such as turning on an LED, triggering a motor, or publishing anything online.

1.8 Document Structure

The structure of this document follows the IEEE Standard for Software Project Management Plans

Section 1 - The first section is an overview of the project. This part covers the project's goal, scope, and objectives, as well as its assumptions, constraints, risks, deliverables, schedule, budget, and plan evolution. It includes a list of all other sources of information and documents cited in the plan, as well as a glossary of abbreviations and acronyms used in the text.

Section 2 - Project Management. The exterior interface, the project's internal organizational structure, and each member's roles and responsibilities are all defined in this section.

Section 3 - Managerial Process Plans. This section details the project's project management procedures. It also specifies the plans for project start-up, risk management, project work plans, tracking plans, and project close-out.

Section 4 - Technical Process Plans. This section defines the technological solutions for developing the various work products in terms of process models, implementation methodologies, tools, and techniques. It also shows strategies for setting up and maintaining the project infrastructure, as well as product acceptance.

Section 5 - Supporting Process Plans. A documentation table is included in this part to ensure that everything is in order. If the ongoing process is a success

Section 6 - Additional Plans. This section outlines the additional plans required to meet contractual obligations, conditions and product specifications

2. Project Organization

2.1 External Interfaces

- **The parent organization**:

  The parent organization is the university itself (IAU), specifically the College of Computer Science and Information Technology department. The Graduation Unit serves as the upper management and we will be working under our supervisor, Dr Farmanullah Zaman Mohammad Jan. The supervisor will be overseeing the project documentation and implementation process.

- **The customer**:

  The College of Computer Science and Information systems will also act as the client, the final project will be delivered to the college at the end.

  - *There will be no external or subcontracted organization interfaces in this project.*
  - *The organizational chart is as shown below:*

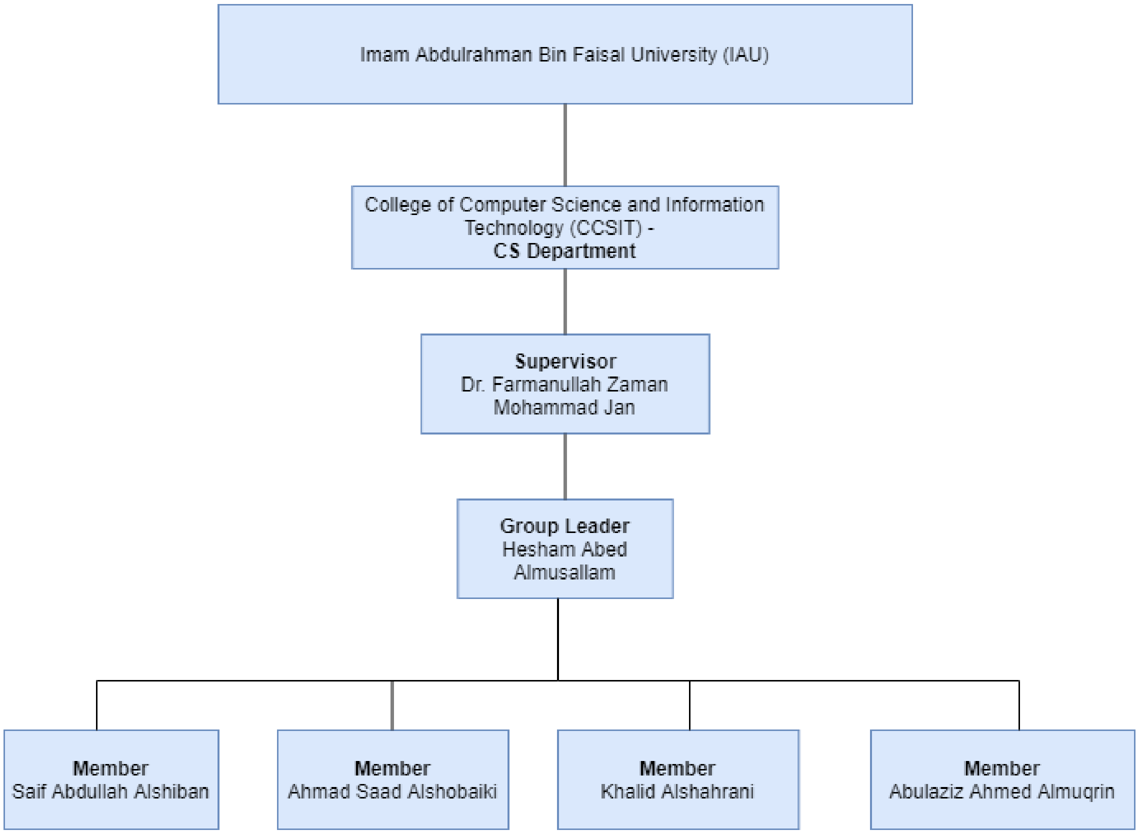


Figure 1 Organization Structure

2.2 Internal Structure

The internal structure consists of five students undertaking the Project Proposal & Project Implementation course. Four team members and one leader who will manage the tasks of the members. Everyone works under the Supervisor, who oversees everyone.

The internal structure is as shown below:

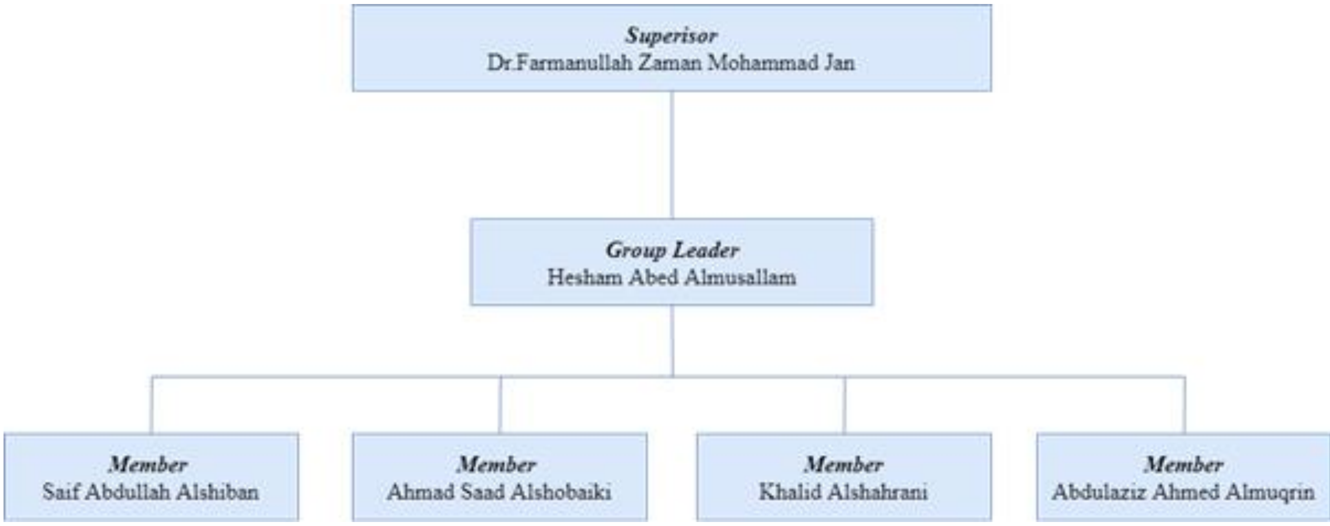


Figure 2 Internal Structure

2.3 Roles and Responsibilities

The roles and responsibilities are as listed below:

Table 3 Roles and Responsibilities

#	Name	Roles and Responsibilities
1	Dr. Farmanullah Jan	**Supervisor –** Oversees the whole project and guides the whole group towards achieving the project goals.
2	Hesham Abed Almusallam	**Team Leader** – Responsible for managing and coordinating all the teams' activities Directs the team to the goal. Divides and assigns tasks among the team. Ensures that tasks are completed. **Team Member** – Works on all the tasks related to the project
3	Khalid Alshahrani	**Team Member** – Works on all the tasks related to the project. May also be assigned special tasks according to the leader and project requirements **Developer –** Build and develop software code. Also works on hardware. **Tester –** Verifies and validates the project
4	Ahmad Saad Alshobaiki	**Team Member** – Works on all the tasks related to the project. May also be assigned special tasks according to the leader and project requirements **Developer –** Build and develop software code. Also works on hardware. **Tester –** Verifies and validates the project

5	Abdulaziz Ahmed Almuqrin	**Team Member** – Works on all the tasks related to the project. May also be assigned special tasks according to the leader and project requirements **Developer –** Build and develop software code. Also works on hardware. **Tester –** Verifies and validates the project
6	Saif Abdullah Alshiban	**Team Member** – Works on all the tasks related to the project. May also be assigned special tasks according to the leader and project requirements **Developer –** Build and develop software code. Also works on hardware. **Tester –** Verifies and validates the project

3. Managerial Process Plans

This section of the Project Management Plan specifies the project management processes for the project. This section defines the plans for project Startup, risk management, project work, project tracking and project closeout.

3.1 Startup Plan

3.1.1. Estimates

The estimates for this project can be listed as follows:

1. **Hardware cost** – On average, each microprocessors cost SR 100 and each sensors cost SR 10. This excludes the price of the smart home prototype.
2. **Documents** – For all documents to be submitted in this project, each document may charge in the range of 100SR for printing and 25SR for binding.
3. **Schedule** – The project will be split into two phases. Phase 1 for documentation of the project and will last from 29-August until 5-December 2021. Phase 2 for implementation of the project and will be done during the 2nd semester. Total time of the project is anticipated to be 24 weeks.
4. **Resources** – Software resources include – Arduino IDE, software code, and IoT mobile application. Hardware resources include – electronic devices such as microprocessors and sensors, plywood for constructing the SH prototype, and a workstation. A detailed breakdown will be provided in section 3.2.4 Budget Allocation.

3.1.2. Staffing

The team consists of five members to complete the documentation for the project of this scale by the agreed upon deadline. All five members are in their final year in Imam Abdulrahman Bin Faisal University. All members have experience and skills in database development, designing software program in C for the mobile application and smart home model. They all have the necessary research skills, designing electronic circuit, writing, decision making, teamwork, and communication skills.

3.1.3 Project Staff Training

Table 4 Project Staff Training

Training Topic	Skills to be obtained from the training	Method of training	Trainees
Arduino Programming Skills	How to program Arduino and hardware components to work together	Online tutorials and lessons about both topics. Instructions from Supervisor	All team members
Electronic Circuit Designing.	Design electronic circuit. Obtain knowledge about electronic circuit components. How to bind and connect electronic components together		

3.2 Work Plan

3.2.1 Work Breakdown Structure

This section specifies the various work activities to be performed in the project and depicts the relationships among these work activities.

Table 5 Work Breakdown Structure

Phase 1 – Documentation	Phase 2 - Implementation
Problem Statement, Motivation, and Literature Review **Estimated Duration:** 2 weeks **Deliverables:** Report #1	**Project Implementation** – Start implementing the proposed project based on previous documentation **Deliverables:** Prototype of the IoT-based Smart Home **Pre-requisite:** Complete Phase 1 documents
Software Project Management Plan (SPMP) - define and document management processes for the project.	**Project Testing** – Verify and Validate the project based on proposed criterions. **Deliverables:** STP Document **Pre-requisite:** SRS & SDS

Estimated Duration/Submission: 2 weeks / 24-10-2021 **Deliverables:** Report #2 **-** SPMP document	
Software Requirements Specification (SRS) - Identify requirements needed to implement the IoT-based Smart Home **Estimated Duration:** 2 weeks **Deliverables:** Report #3 - SRS document	**Project Acceptance –** Evaluate whether the implemented project is accepted based on a defined standard. **Deliverables:** Final Working prototype of the IoT-based Smart Home
Software Design Specification (SDS) - Specify the Design of the IoT-based Smart Home **Estimated Duration:** 2 weeks **Deliverables:** Report #4 **-** SDS document	
Final Report – Compilation of all reports **Estimated Duration/Submission:** 2 weeks/ 9-12-2021 **Deliverables:** Report #5 **-** Final Report	

3.2.2 Schedule Allocation

The work mentioned above presented in a timeline to contribute to the time management process.

Table below shows these activities in a sequence of two semesters. The work plan from week 18 still has not started as the tasks assigned are to be done during the 2nd semester of the academic year 2021-2022.

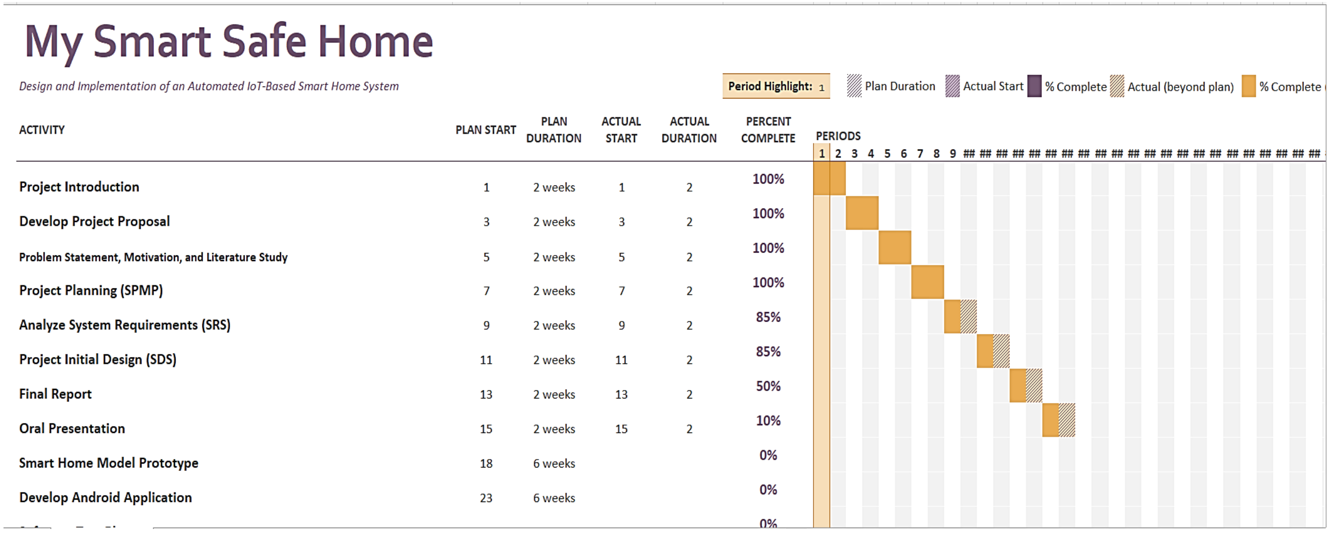


Figure 3 MSSH Project Timeline

3.2.3 Resource Allocation

Resources include both human and non-human resources. Table below lists some of the necessary resources.

Table 6 Resource Allocation

	Computing Resource	**Software Tools**	**Special Testing and Simulation Facilities**	**Administrative Support**	**No. of Personnel and Skill Level**
Problem Statement	Personal Computer Mobile Phone	Microsoft Office	-	Supervisor Feedback and Evaluator Feedback	All team members
SPMP					
SRS					
SDS					
Final Report					

Project Implementation		Arduino IDE, IoT Mobile App, Cloud Server, Database Server	Construction appliances		
Project Testing			-		
Project Acceptance					

3.2.4 Budget Allocation

A breakdown of the necessary resource budgets for all the work activities.

Table 7 Budget Allocation

Tool or Equipment	Cost	Quantity
NodeMCU ESP8266	50	1
Jumper wires	92	4
IC buffers	34	6
Transistors	20	6
Lamps with holder	105	10
Bread board	115	1
Crocodile clips cables	20	3
Mq2 gas sensor	80	2

IR Flame Sensor Detector	50	2
Micro servo motor SG90	259	9
Power cable	17	1
Micro bit kit	350	1
Total:	SR 1,192	

3.3 Project Tracking Plan

3.3.1. Requirements Management

- The SPMP for this project will be under control and monitored, so that any changes made in the plan will be updated and available to the team members.
- Any requirement changes made to this project will be updated and declared to everyone involved.
- There are weekly meetings with team members and supervisors to review the project management process and discuss the updates on the project.
- In case of changes, the current and future documents will be updated simultaneously and will be available to the team and supervisors.

3.3.2. Schedule Control

- The progress made in the project will be in accordance with the declared dates in the syllabus unless deadlines change
- We will follow the Gantt chart to keep track of important deadlines as specified above. Changes to the timeline may be made depending on the situation.
- Each member is responsible for ensuring that the progress of the project is following the plan and all tasks are completed on time.
- A meeting is conducted weekly to discuss work progress, and questions and answers, which ensures the scheduled plan is followed.

- The timeline is tentative, and changes can be made. If so, it will be updated regularly. Therefore, the weekly team meetings among the projects team members and the supervisors to discuss the project's progress are important to manage the changes according to the schedule.

3.3.3 Quality Control

- The quality of the project will be measured against how closely we meet our project objectives mentioned in section 1.1.
- Software testing phase will verify and validate whether the project is up to quality and meets the project objectives.
- The supervisor and all team members ensure that the system is of high quality, and several tests will be conducted to evaluate the product. These tests are carried out during meetings, and they are carried out numerous times. It runs with a different test case each time to ensure that there are no mistakes in the system and that all functionalities work as expected.

3.3.4 Reporting

- Weekly group meetings with the supervisor will be held to report the project progress.
- Physical meetings, official email, and voice calls will be the main communication tools for the project.
- Meetings will be held weekly, and we will discuss project progress, requirement changes (if any), tasks done, and feedbacks.

3.3.5 Project Metrics

- The project metrics that we initialize will be used to track the project progress to keep up with our expectations.
  - Collection and updating the metrics will be done every week and analyzing the results will be done against the metric we defined.

- Some metrics to consider in this project are:

Table 8 Project Metrics

Metric	**Question**	**Description**
Time	What is the status of the project in relation to the original schedule?	In sections 3.2.1 and 3.2.2, the progress is compared to the schedule.
Value	What percentage of the objectives were met?	The project's progress will be monitored using the project's requirement specification (SRS).
Resource	How much resource did the project consume?	The progress is compared to the schedule in sections 3.2.2.
Quality	Has the quality issue been resolved in order to complete the project successfully?	After testing the system, there may be some flaws, thus any problems should be resolved before the project's final submission date.
Scope	Was there a change in scope during the project? How many times has this been done?	Depending on how the requirements evolve, the scope of the project may shift from its original scope.

3.4 Risk Management Plan

- The leader will Create a plan B if the initial plan didn’t go as planned; the procedures of planning will be affected by the factors which led to the disturbance of the first plan.
- We will keep track of risks by setting a criterion for every aspect of the project, every function should work as planned, if that’s not the case then a new risk is discovered.

- Documentation of the risk will be done when a new risk is detected, the report will include description of the risk and procedures to eliminate the risk.
- The team members and the leader will be responsible for risk detection and mitigation.
- The leader will communicate with the members and the parent organization (the supervisor) if there's any issues between these entities, then it will be solved.

Table 9 Risk Management Plan

Risk	Definition	Category	Cause	Response	Probability	Impact
Hardware defects	Hardware cables and modules could stop working as intended	Hardware errors	Hardware is broken or cheap	Buy official hardware brand and check for defects	High	High
Communication loss	The Arduino could lose connection to the cloud or vice versa	Hardware and software errors	Weak and slow internet	Check for WIFI connection before starting	Medium	High
Timing	Keeping up with the tasks before deadlines	Team management error	Slow task completion	Finishing tasks on time	Low	Medium
Loss of power	The Arduino could lose power or voltage	Hardware errors	Dead battery or power shortage	Charge the battery or keep the plug on the socket	Medium	High
Budget shortage	Budget of the project may not be enough	Team management errors	Expensive hardware kit or low budget	Save money and buy affordable kits	Highest	Medium

Software bugs and errors	Software errors and bugs	Software error	Lack of debugging and error handling	Use error handling and debugging	High	Medium
Electrical hazard	Dangers while handling the hardware	Hardware errors	Poor Electricity handling or circuit shortage.	Use reliable wires and do not use conductive elements	Low	High
Team members absence	Team members could have different schedules	Team management errors	Personnel not aligning their schedules together	Make a common timetable and meetings	High	Medium
Loss of information	Unsaved Data maybe lost	Team management and software errors	Data failed to be saved or hardware malfunction	Make a backup file for data and retrieve it	Medium	Medium

3.5 Project Closeout Plan

- A final report will be written detailing the and all previous documents will be compiled together.
- After project closeout, all required deliverables will have been submitted to the appropriate unit.
- A backup copy of all the documents will also be stored by each of the group members for archiving.

4. Technical Process Plans

4.1 Process Model

- Since the waterfall model approach divides the project into sequential steps, this project will be implemented using the waterfall model.
- The process will progress to the next stage after each activity is performed.
- The milestones are organized by the college of computer science and information technology's senior project plan (CCSIT).

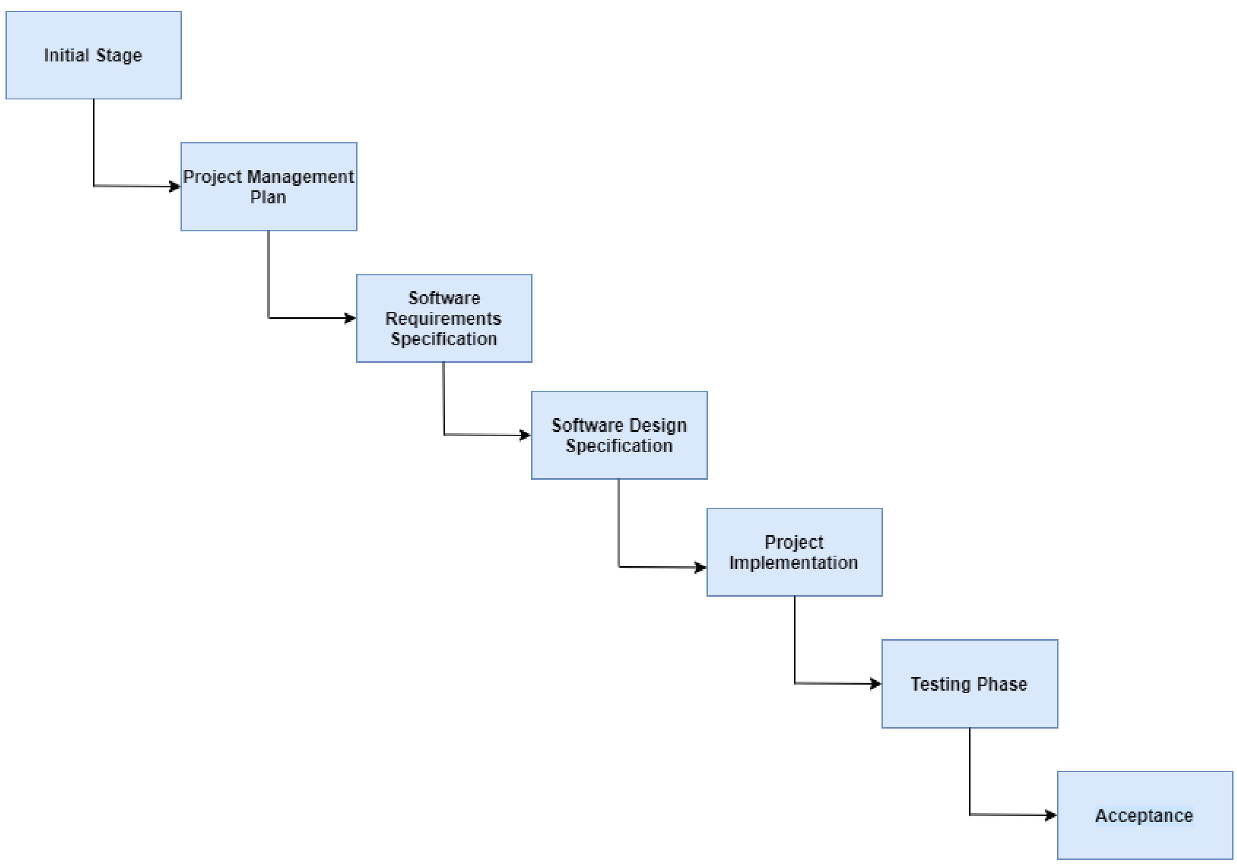


Figure 4 Waterfall Model

4.2 Methods, Tools, and Techniques

- We will follow a sequential approach in the implementation of this project. Completing each phase before proceeding to the next one will ensure that a high-quality project.
- Microsoft Office tools will be used for documentation.
- Arduino Programming environment will be used to write software code in C.
- A cloud server and a database server will be used to store IoT data.
- An IoT mobile app will be utilized to provide a user interface to interact with IoT devices
- An SH prototype will be modelled and constructed using cardboard.

4.3 Infrastructure

The needed hardware, operating system, software, and networks must be provided for the installation and maintenance of the development environment.

- Hardware: Each team member must have a computer and a mobile device.
- Software: Blynk, Arduino IDE and Microsoft Office are all necessary.
- Operating system: Any operating system will function on PCs; however Windows 10 is recommended.
- Any operating system for mobile devices.
- Network: a wireless network is a must (Wi-Fi).

4.4 Product Acceptance

The product will be measured against the project metrics defined in 3.3.6 Project Metrics to identify whether the product is acceptable.

5. Supporting Process Plans

5.1 Documentation

The following papers will be prepared to document the process:

Table 10 Documentation Plan

Document Name	Document Description	Document Template	Prepared By	Reviewed By
Software Project Management Plan (SPMP)	This document will contain information about how the project will be managed	IEEE Std 1058 – 1998	All Team Members	Dr. Farmanullah Jam Dr. Dhiaa Abdulrab Ali Musleh
System Requirements Specification (SRS)	to describe what the software will do and contains the expected requirements	IEEE Std 830 – 1998		
System Design Specification (SDS)	a detailed plan of the development process. It contains an overview, context and goals, proposed solution, timeline, and other information needed for project development.	IEEE Std 1016 – 1998		
Final Report	a final version of the compilation of the previous documents altogether.	Official College Final Report Template + Compilation of previous reports		

Software Test Plan (STP)	a written report of the test plan related to the system	IEEE Std 829 – 1998		

6. Additional Plans

All personnel involved must wear proper safety equipment and follow standard lab safety rules especially when working with electronics.

Chapter 4: System Requirement Specification

Revision History

Name	Date	Reason For Changes	Version
All members	Sep 30, 2013	Prepared initial version	0.1
All members	Oct. 30, 2013	Updated section 3	0.2
…			
All members	Nov. 14, 2013	Complete review - Final version	1.0

1.Introduction

This section summarizes the topics covered in this Smart Automation System Software Requirements Specifications (SRS) paper, these topics include the following: The SRS's goal, intended users, range, scope, list of abbreviations and acronyms, sources used in this document, and the report's abstract are all discussed.

1.1. Purpose

The goal of this paper is to provide a complete and exhaustive overview of themobileapplicat ion and smart home model specifications (My Safe Smart Home).

This document will detail all the application's interfaces, as well as system restrictions, performance requirements, and functional and non-functional requirements for the mobile app and smart home model. In addition, the SRS is aimed towards the following audiences: Project development team: This document will be used as a reference by the development team to aid them in the process of developing and implementing the software. Project supervisor: the project supervisor will analyze the paper and make ideas to help improve the project.

1.2. Scope

The end product that will be provided at the end will have both software and hardware elements, we will develop and engineer the automated home titled MSSH, this project will provide services to owners of the smart home to control, monitor and interact with their home remotely.

MSSH will provide many services to homeowners that which will improve the quality of living in the future, this system will be created to offer benefits. to provide the customer control over opening and shutting the door, turning on and off the lights, security sensors and alarms and other activities that demands effort on behalf of the homeowners. The final output will be a miniature working model of a smart house made with Arduino and other technologies that assist automate the process of regulating household appliances. These features will be a useful tool for providing owners with feedback in the event of an unsettling circumstance. A connection between the Arduino present in the smart house model and the MSSH application is expected to operate the system.

1.3 Definitions, Acronyms, and Abbreviations

Table 1 and 2 shows the definitions and acronyms used in this document, respectively.

Table 11-Definitions

Term	Definition
Software Requirement Specification (SRS)	Software Requirement Specification is a description of a software system to be developed, laying out functional and non-functional requirements.
Software Project Management Plans (SPMP)	Software Project Management Plan is a detailed plan that states the standards, procedures and project management tools that must be adopted by the project team. It includes several plans, such as plans regarding the project's cost, schedule, and resources.
Arduino	It's an open-source development boards used for production of hardware smart devices that can be controlled, monitored and interacted with by the end user.
Android studio	Official IDE for developing Android applications, it has a fast and flexible emulator
Activity Diagram	A diagram that shows step-by-step roadmap of the activity inside the application.
SQLite	relational database management system (RDBMS)

Table 12-Acronyms

Acronyms	Definition
MSSH	My Safe Smart Home
SRS	Software Requirement Specification
SPMP	Software Project Management Plans
App	Mobile Application
WIFI	Wireless Fidelity

1.4. Overview

The goal of the project is to create a model for a smart house that allows users to control their home electronics using their smartphones. Some objectives must be specified in order to meet the project's objective, and after they are met, the project will be finished. Using a Mobile application, the system should be able to command the home. The following is how this documentation is arranged to explain all the project's requirements:

1. Introduction: the subjects included in this Software Requirements Specifications (SRS) paper are introduced in this section. It includes an explanation of the SRS's goal, intended users, scope, definitions, acronyms, and references, as well as a summary of the whole document.

2. Overall Description: This section represents the MSSH System's product perspective and functionality. It also goes into great depth on the targeted users' qualities.

3. Specific Requirements: This section shows how the Smart Home Automation System interacts with people, infrastructure, and application.

4. **Performance Requirements:** This section explains how to specify properties like protection, confidentiality, ease of maintenance, and security.

5. **Design Constraints:** This section presented the architectural limitations of the software, such as physical limitations and scripting languages.

6. **Software System Attributes:** This section explains how to define attributes like confidentiality, anonymity, ease of maintenance, and health.

2. Overall description

This section gives a general overview of MSSH (My Safe Smart Home). It is broken down into three portions. The system's perspective is explained in the first subsection, the system's functioning and characteristics are explained in the second and third subsections, respectfully

2.1 Product perspective

MSSH (my safe smart home) is a self-contained, it consists of a mobile application that is connected to a smart home model via Wi-Fi the mobile application will be developed using Blynk and the smart home model will be constructed using Arduino other electronic components, the system uses Cloud of Things to connect to the mobile application database, which is utilized to store the information of the homeowners. The system allows homeowners to quickly control household appliances and hardware from a distance it offers smart home control features such as opening and closing doors, turning lights on and off, add its detecting smoke and fire and sounding alarm in the event of an emergency.

2.1.1. System interfaces

The blynk mobile application and the smart home model are the two primary components of the system interfaces. Both pieces will be connected in order to carry out the user's orders. Within the range of the WIFI connection, the system allows the user to control the smart home remotely.

2.1.2. User interfaces

The system's primary users are homeowners, and the system provides them with an easy-to-read and use interface. To access the system, each family member needs to have a username and password. The user will be prompted to enter their login and password on the first screen. To link the smart home model with the mobile app, the user must first connect through WIFI from his or her mobile phone. After that, the user may go to the appliance page, where they can see all the working appliances in their home. In the settings page, the user can update his or her information. Users can read about MSSH on the about MSSH manual if they have any questions regarding the application or how it is used.

2.1.3. Hardware interfaces

The hardware of the entire system is represented by the smart home model. The appliances in the home model are controlled by Arduino to function properly, the system will require the following components: Arduino boards, WIFI modules, wires, LEDs, motors, breadboards, and sensors are just a few of the components you'll need. The Arduino IDE software should be installed on each member of the development team's PCs to program these hardware components.

2.1.4. Software interfaces

The Android mobile application represents the software of the system, by using blynk cloud mobile application and by utilizing blynk cloud services to create the mobile application interfaces and their elements, the mobile application reflects the system's software.

2.1.5. Communications interfaces

WIFI will be used to communicate between the mobile application and the home model. WIFI module in the Arduino board will be coded to link to the mobile application.

2.1.6. Memory constraints

To work successfully, the user of the application must have enough space on their mobile dev ice while downloading it.

2.1.7. Apportioning of requirements

This project will be carried out using a smart home automation paradigm, to turn a smart home automation system into a smart home, part of the systems equipment will need to be replaced with more efficient and compatible can technology that is compatible with` real gadgets into real house in the future.

In general, there are two types of home automation systems wired and wireless, white systems required an engineer to connect devices to the Arduino microcontroller while wireless systems do not, it necessitates the installation of electrical extensions during the initial stages of construction, the other option is wireless home automation system, which requires that the gadgets utilized support this capability and can be installed after the house has been built, when transforming a smart home model into a genuine smart home, some tiny equipment such as small a delight to a large LED lights must be replaced stronger and larger motors are being used to open and close the house doors, so in general real smart homes require larger appliances.

2.2. Product functions

The homeowner is the primary user of My Safe Smart Home. The system offers a wide range of options to the user. The following are the key application functions provided by the system:

Account:

- *To have his or her own personal account, the user must first register with the system.*
- *The user should be able to correctly login and logout.*
- *The user can make changes to some of his or her information.*
- *If the user forgets his or her password, an E-mail will be sent to the user's E-mail address with a random password.*
- *The user has the ability to modify his or her own password.*

Devices that are controlled:

- *The user has the ability to turn on and off the lights in the rooms.*
- *The hue of the light can be changed.*
- *If there is a gas or smoke leak, a fire or smoke alarm should be activated.*

- *The user has control of the house doors by opening and closing them, combining the opening closing of several appliances.*

2.3. User characteristics

Despite it is intended for a single user, the homeowner, following is the list of the general quantities of homeowners:

- **Educational Level:** The technique can be used by students at any level of study.
- **Experience:** The user does not need to have any prior experience.
- **Technical Expertise:** No technical expertise is required, all that is required as the user understands how to update the mobile phone application.

2.4. Constraints

The system's restrictions are as follows:

- *Safety precautions should be taken when assembling equipment.*
- *The user's device must run on the Android operating system.*
- *Arduino should be able to run software.*
- *The application can only be used inside the WIFI range.*
- *The users are identified by their username and password.*

2.5. Assumptions and dependencies

This section describes the assumptions and dependencies that affect the requirements contained in the SRS.

2.5.1. Assumptions

The following are the system's assumptions:

- *This technology is said to provide greater comfort and lessen movement effort.*
- *For the project's development, a budget will be established.*
- *Some of the resources are likely to be available locally.*
- *It is assumed that this system correctly responds to commands.*

2.5.2. Dependencies

The following are the system's dependencies:

- *The WIFI connection is required for smart home automation.*
- *It is contingent on the Arduino hardware being correctly connected to Android stu dio.*

3. Specific requirements

The external interface need, as well as all of the system's high - quality and innovative requirements, will be explained in this section. It offers a comprehensive overview of the system and all of its capabilities.

3.1External interface requirements

This section goes through the user interfaces, software, hardware, and networking technology in detail. In addition to describing in detail all of the system's transactions with the user.

3.1.1. User interfaces

The Android smartphone app will have a basic and user-friendly interface. All family members who live in the smart house can effortlessly manage their environment, all of the icons in the mobile app that indicate the equipment in the smart home are conscience and easy to understand for all people, The following are the user interfaces provided in the project's application:

3.1.1.1. Login Interface

When users launch the program, they should be able to view the login interface. The login interface includes:

- **Login: A** registered user should be able to access the program by entering a correct username and password.

Table 13-Login interface

Field Name	Data type	Size	Constraints	Description

email	Varchar	32	Primary key	Username is used to identify the user when attempting to log in
Password	Varchar	32	At least 8 characters, include numbers and special characters.	Password used to secure the user’s account.

When new users follow the button, they will be directed to the platform's registration form.

3.1.1.2. Register Interface

To create an account for each family member, house owners must submit their information by filling out a registration form. Each member of the household has their own membership and can access the system.

Table 14- Register interface

Field Name	Data type	Size	Constraints	Description
Username	Varchar	32	Primary key	Username is used to identify the user when attempting to log in
Password	Varchar	32	At least 8 characters, include numbers and special characters.	Password used to secure the user’s account.
E-mail	Varchar	32	Needs to be in proper email format	Used to generate a new password if the user forgets the old password.

3.1.1.3. Control panel Interface:

Home interface is the main interface in MSSH application.it has around7 buttons (lights, doors & windows), these buttons will switch on/off the intended device, such as turning lights and doors on or displaying their status, also each button is labeled as the name of the room its located in.

3.1.1.4. Monitoring panel interface:

This interface will show all the sensors status and present value, such as showing the room temperature and humidity, it can also display the fire and smoke detector value .

3.1.1.5. Change my password Interface:

This interface has 2 fields current password and new password, it will be used to change the users password.

Table 15-change my password interface

Field Name	Data type	Size	Constraints	Description
Current Password	Varchar	32	At least 8 characters, include numbers and special characters.	Password used to secure the user's account.
New Password	Varchar	32	Needs to be the same as password	Used to confirm the user's new password

3.1.1.6.bForgot my password interface:

This interface contains an e-mail field, this e-mail will be used to generate a new password for the user based on the email.

Table 16-forgot my password interface

Field Name	Data type	Size	Constraints	Description
E-mail	Varchar	32	Needs to be in a proper e-mail format	Used to restore the user's account.
New password	New Password	Varchar	32	Needs to be the same as password

3.1.1.7. My profile Interface:

Users should be able to see the profile icon right side corner of the screen where they may update their name., also the user can go to the change my password interface from this page, deleting the user account is also possible from this page.

Table 17-my profile Interface

Field Name	Data type	Size	Constraints	Description
Username	Varchar	32	Primary key	Username is used to identify the user when attempting to log in
E-mail	Varchar	32	Proper e-mail format	Used to verify the user

3.1.2. Hardware Interfaces

Smart dashboard will be used to control the smart home appliances all the components will be embedded into the smart home model, Arduino board has an ATmega processor and uses 5 voltage and has a maximum speed of 16 megahertz, it can handle many analog and digital interfaces, with a USB input, all basic components can be connected to our projects which are:

- **Temperature on humidity sensor:** used to sense temperature and humidity in a specific radius inside the house.
- **Gas, fire, and smoke detector:** these sensors will detect dangerous leaks and will show the value of the sensor in real time.
- **Connecting wires and breadboards**: this will be used to connect the devices to the Arduino board.
- **LED lights:** the house will be lit by led lights that will be connected to the board and can be controlled via the application.
- **Motors and servos:** will be activated to open and close doors and windows and view their status on the application.

3.1.3. Communication Interface

We'll establish a wireless connection bridge between the mobile application and the mobile device, allowing the Smart home model and the mobile application to communicate. Because mobile application doesn't have an inbuilt Wi-Fi module, the Wi-Fi module will be connected to the board. In slave mode, it enables wireless data transmission. The microcontroller has a big Wi-Fi range.

3.2. Functional requirements

The MSSH mobile application's functionalities are used to classify the system's functional requirements. This section delves into the specifics of how these features are linked to the smart home model.

3.2.1 Sign In:

Users must login with a valid account in order to utilize the system. The following sections help compensate the login interface:

3.2.1.1 Login

After registering, users must input a valid username and password to access the MSSH application. An error will occur if an invalid account is typed in the forms, and users will be required to enter a valid login and password. Users are then moved to the next interface, which is the home interface, after the account has been authenticated by confirming its authenticity in the database.

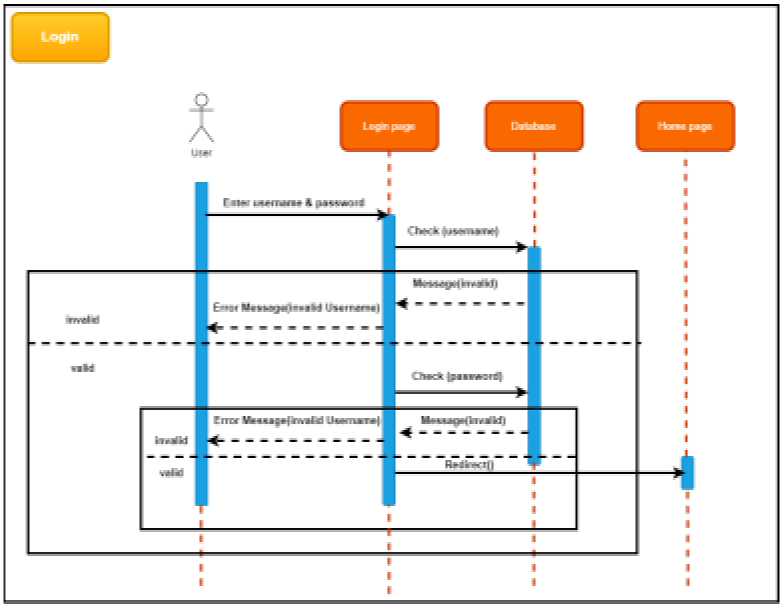


Figure 5-Login activity diagram

3.2.1.2 Forget Password

It is possible for users to request a new password if they forget their current one. Simply click the Forget Password link and input the E-mail address that was used to create the account. The MSSH application will provide the user a new random password through email. The diagram below describes the login and forget password activities.

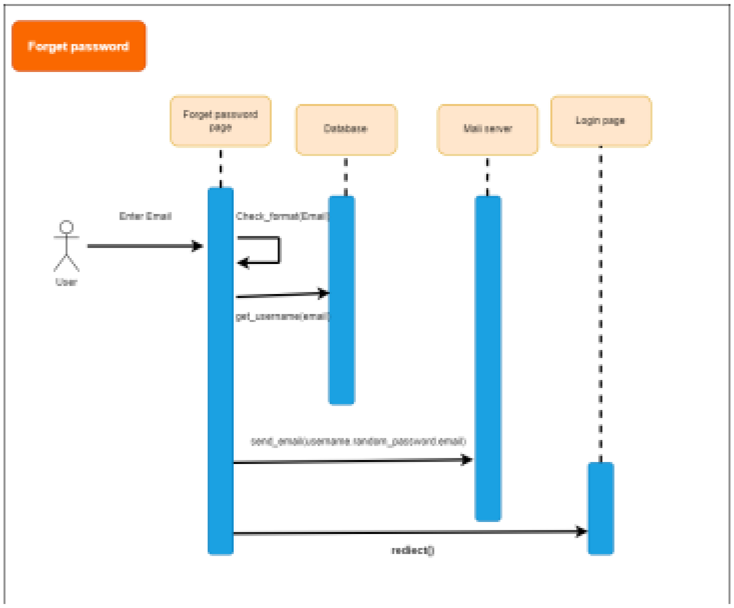


Figure 6-"Forget password" Activity Diagram

3.2.2. Register

Users who are new to the MSSH application should fill out the registration form to create an account. The username, password, confirm password and e-mail address. are all necessary information on the form. The program must check its database once the user has supplied a username to ensure that all usernames entered are unique. If the user inputs an existing username, a warning notice will show, requesting that the user update his or her username. The user must then submit a password for his or her account as well as an email address in a certain format. If the format is incorrect or the passwords do not match, a notification will show, otherwise the user will be successfully registered and will be able to access to the system. The interface activity diagram for the register is shown in the graphic below:

Figure 7-"Register" Activity Diagram

3.2.3 Control panel

This is the main interface of the blynk application, it will display the buttons , sliders , switches, and other controls necessary to operate the lights , doors and windows for each room in the smart house.

The user's main functionality is the ability to turn lights on and off. Its goal is to make lighting control more comfortable and simpler for system users. People may find it difficult to walk from one side of the room to the other due to the distribution of light switches throughout the space. They may, for example, have leg and foot injuries, be elderly, have female family members who are pregnant, or have physical limitations.

Users must select a specific light in the room they want to manage to turn on or off the lights. They can then turn it on or off by swiping the toggle button left or right. The action must be implemented to the smart home by the Arduino device as soon as the user decides on the action to be conducted. The blynk mobile application sends commands to the Arduino board over a wireless WIFI connection.

Operators should be able to open or close the door or window when they select this portion of the appliance interface. To open or close the doors or windows, the user simply swipes the toggle button left or right. The action of sliding the door or window open or closed must be accomplished as soon as the user hits the button.

3.2.4 Monitoring panel

This the observing interface that can be reached from the control panel, in this page the user can view the current estimates of the smart measuring devices such as temperature , humidity and gas.

Homeowners have the opportunity to check the temperature of their rooms at any time. The temperature is displayed by selecting the monitoring panel in the home interface.

3.2.5 Help

The MSSH/blynk help interface provides a quick summary of how to use the blynk mobile application and assists the user in understanding how it works. For example, if a user wishes to control appliances but forgets to connect his or her mobile device to the Arduino device by WIFI, the gadget will not work until the user properly links the device via Wi-Fi, this page can be visited from the main interface.

3.2.5 My profile

My profile interface allows users to make changes to their information stored in the database. When users access this screen, they have three options: update their username , change their account password, or delete the account, The options are outlined in the following sections.

3.2.5.1 Update username

In this part, users can update their username, the same restrictions apply for username uniqueness as in the registration form, as all data is stored in the same database.

3.2.5.2 Change password

If users want to update their password, they can do so by entering their current password first, then they can input their new desired password in the text area. The system displays a warning message if any invalid inputs are typed in any field.

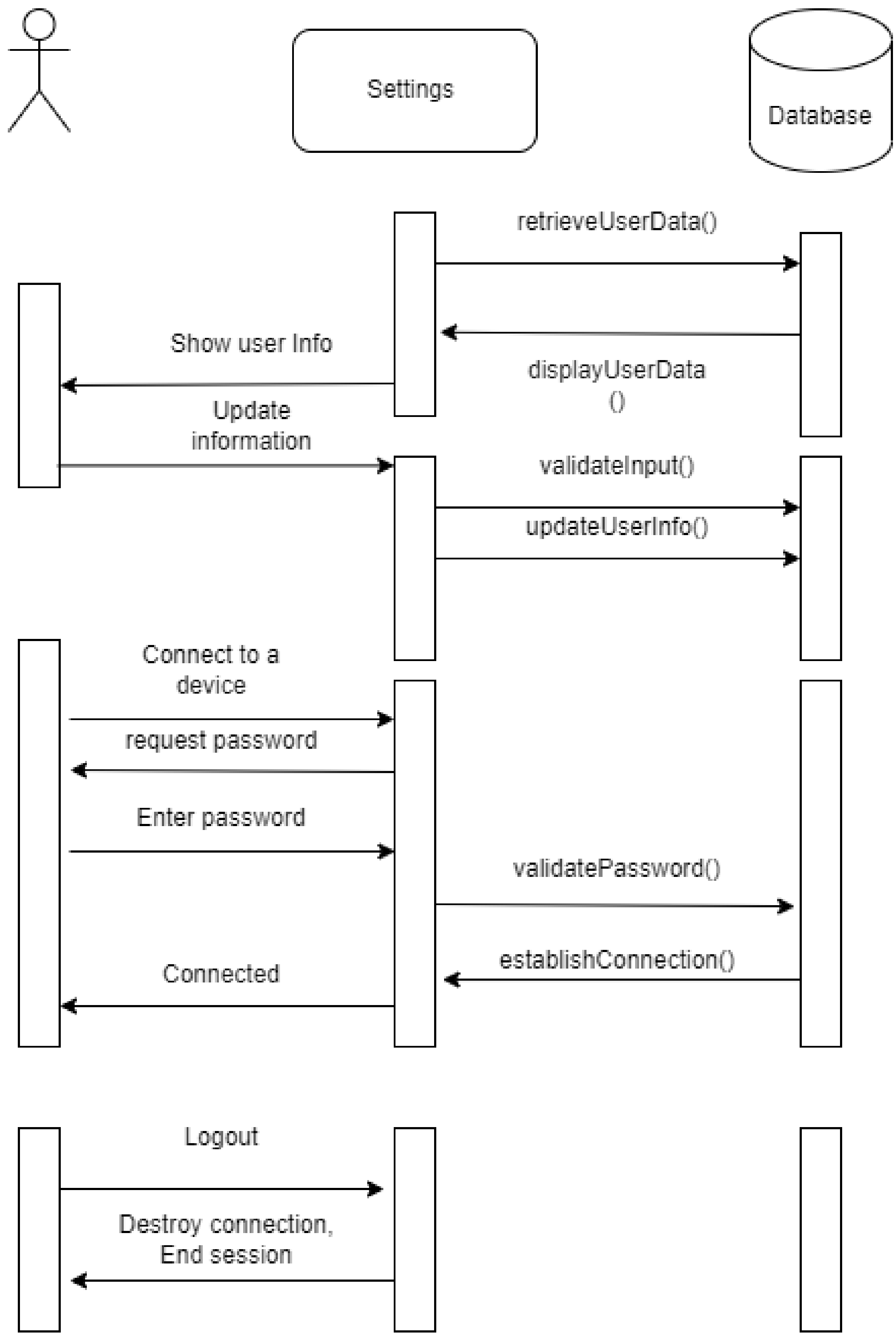


Figure 8-"Settings" Activity Diagram

3.3. Performance requirements

The system's response time, capacity, update confirmation, and the distance between the Android mobile application and the smart home device are among the estimated performance requirements listed below.

Response time:

When customers press a button on the mobile app, the command will be transferred through Wi-Fi from the app to the Arduino board in the smart home model within 10-60 seconds. Users will see the action take place as soon as the button is pressed, However the timings may change in future drafts.

Capacity:

The MSSH system gives its users a lot of remote-control options. Users must have enough memory space on their mobile devices to install the blynk program for these functionalities to work smoothly and accurately.

Information updates confirmation:

The blynk application stores all relevant information about the system's users in a database. All updates made by app users will be validated with a notification, and the update will take place in real time. If the state of a sensor changes, it will be updated in the database.

Distance between MSSH application and smart home model:

Both the mobile device and the smart home device must be within the range of the WIFI connection area for commands to execute.

3.4.Design constraints

This section outlines the design restrictions that must be met in order for the implementation to be successful

Mobile platform:

The main design constraint is that the MSSH application will be built for smartphones running the most recent operating system, which must enable WIFI services.

Development language:

The Smart Home Automation System will be created utilizing the Arduino IDE programming language C, which is compatible with the mobile platform. Blynk and Arduino IDE will be the platform that we will use in this project. The Arduino board is used in the smart house model, which is programmed in C.

Time constrains:

The project's duration is restricted. The scheduling specifics for this project are shown in the included Gantt chart in SPMP document.

3.5. Software system attributes

Some requirements can be regarded as system properties, and these requirements are detailed in the sub - sections below.

3.5.1 Reliability requirement

The MSSH system is a stable system that consistently performs high-level operations and activities. Users benefit from its features, particularly in the event of an emergency, when the fire detector detect high levels of smoke the user will be notified. For all valid and invalid conditions, the system should perform all functionalities correctly and without mistakes.

3.5.2 Availability requirement

The MSSH system should be available to execute user commands 24 hours a day, seven days a week. Both the MSSH application and the smart home model require a WIFI connection to the mobile device. Only inside the WIFI range can the system perform its functions. All user accounts' data is stored in the cloud database.

3.5.3 Security requirement

When connecting a WIFI device to the Arduino board in the home model, the system should request users for a password for security reasons. This is done to prevent non-homeowners from operating the smart home if the app is installed on their devices.

3.5.4 Maintainability requirement

MSSH system maintenance comprises both the MSSH mobile application and the smart home model. To ensure that the system's users are completely satisfied, project developers should spend as much time as possible repairing faults that may emerge in the system.

3.5.5 Portability requirement

Smartphones running the most recent operating system can use the MSSH system. Only within the range of the WIFI connection can commands from the mobile application be applied to the smart home model. If a user's WIFI connection is lost, the system should display a warning message and the commands should be rejected.

3.6 Other Requirements

This section will go over the system's additional requirements.

3.6.1 Usability Requirements

The system should be simple to use and efficient. Users can operate appliances due to the system's flexibility. Furthermore, the user's technical details should be hidden (i.e. the user should not be able to see the code or the database).

3.6.2 Safety Requirements

The MSSH mobile application should not interfere with any other apps on the device. Any data stored on the device, or any application functionality must not be risked by the application.

Chapter 5: Software Design Specifications

Revision History

Name	Date	Reason For Changes	Version
All members	Sep 30, 2013	Prepared initial version	0.1
All members	Oct. 30, 2013	Updated section 3	0.2
…			
All members	Nov. 14, 2013	Complete review - Final version	1.0

1. Introduction

System Design Specifications (SDS) is a document that describes in detail the data, components, and interfaces described in Software Requirement Specifications (SRS) for implementing a Smart home automation system (SHAS). It is concerned with developing a successful and proficient framework execution that meets the SHAS requirements. SDS will be carried out in two stages to provide all of the information required for project programming. These are the two phases.

- **The initial design phase:** It includes high-level designs for main user interfaces, system architectures, data design, and component design.
- **The detailed design phase:** It includes system architecture decomposition, detailed data descriptions, sub-interfaces, user interface error messages, and pseudocode for each application's function.

This section defines the purpose and scope of the SDS, as well as the definitions, acronyms, and abbreviations used in this document.

1.1 Purpose

The goal of this document is to define detailed design specifications for the proposed MSSH Android application in order to facilitate simple execution. It depicts the system architecture design and user interface design as they relate to the SRS. Also clarifies data and component design in order to provide a detailed description of this application's data. This project's target audiences are as follows:

- **Project supervisor:** The project supervisor will go over the document and make suggestions for improvements.
- **Application development team**: This document will be used as the primary reference and guide for the implementation of My Smart Safe Home by the team members.

1.2 Scope

This document covers all design aspects required to complete the MSSH system. The system will include unique features and services that will allow homeowners to control their smart homes. The following tools are used to model data, relationships between entities, data processes, system architecture, and logical system design:

- Entity-relationship Diagram (ERD)
- Logical Scheme (ERD Mapping)
- Context Data Flow Diagram
- Data Flow Diagrams (DFDs), and
- Sequence Diagrams

It also collects details about the system's database, which is used to store information, as well as pseudocode for the system's functionalities and the system's prototype (i.e. smart home model and simple illustration of systems interfaces).

1.3 Definitions, Acronyms, and Abbreviations

Table 1 and 2 show the acronyms and definitions used in this document, respectively.

Table 18List of Definitions Table

Term	Definition
Software Requirement Specification (SRS)	A Software Requirement Specification is a description of a software system that will be developed that includes both functional and non-functional requirements. [1]
Software Design Specification (SDS)	A Software Design Specification is a design document that describes all aspects of the software's data, architectural, interface, and component-level design. [1]
Arduino	It's an open-source electronics platform with simple hardware and software. Arduino boards can read inputs such as a light on a sensor, a finger on a button, or a Twitter message and convert them into outputs such as activating a motor or turning on an LED.
SQLite	SQLite is an open-source SQL embedded database, where its database implementations are embedded in Android studio. No connection establishment is need here such as JDBC or ODBC.
Wi-Fi	Wi-Fi is a wireless networking technology that connects computers (laptops and desktops), mobile devices (smart phones and wearables), and other devices (printers and video cameras) to the Internet. It enables these devices, as well as many others, to communicate with one another, resulting in the formation of a network.[3]

Table 19List of Acronyms

Acronyms	Definition
MSSH	My Smart Safe Home
CS	Computer Science
DBMS	Database Management System
RDBM	Relational Database Management System
IEEE	Institute of Electrical and Electronics Engineering
App	Mobile Application
SRS	Software Requirement Specification
SDS	Software Design Specification
ERD	Entity Relationship Diagram
DFD	Data Flow Diagram
IDE	Integrated Development Environment
MB	Megabyte
OS	Operating System
DB	Database
GUI	Graphical User Interface
ER	Entity Relationship

1.4 References

1. AlShallali, Bashair M., et al. Software Design Specification. 2016.
2. Mitchell, B., "What Is Bluetooth Wireless Networking? ", (2016, October 31). , Retrieved from: https://www.lifewire.com/definition-of-bluetooth-816260
3. Cisco "What Is Wi-Fi?" Retrieved from https://www.cisco.com/c/en/us/products/wireless/what-is-wifi.html

2. System overview

The proposed My Smart Safe Home is primarily aimed at homeowners who want to remotely control devices and appliances in their homes. All homeowner's need is the mobile application installed on their mobile device, as well as a house with appropriate appliances to work with the application. In this project, a smart home prototype will be built that includes all the components required to successfully execute the system's functionalities. The Android official IDE will be used to develop the mobile application.

2.1General Functionality:

Arduino boards attached to the appliances in the smart home model will perform all system functions. In the smart home model, commands are sent from the mobile application to the Arduino board via a Wi-Fi connection. These features only allow homeowners to remotely control their smart home. These features are as follows:

2.1.1 Account functionalities

Users of the MSSH mobile application can do the following in their personal account:

- *To have his or her own personal account, users must register in the system.*
- *User will be able to successfully login and logout.*
- *Users can make changes to some of their information.*
- *If a user forgets their password, an email should be sent to the user's email address, that would give the user the ability to change the password.*
- *Users have the ability to change their own passwords, at any given time.*
-

2.1.2 Appliances control functionalities

The primary functions of MSSH are to remotely control appliances in the smart home. This is accomplished by simply clicking on the functionality icon in the mobile app. The following are the control functions of the appliances:

- *Changing the status of lights in rooms by turning them on and off.*
- *Fire alarm, in case of emergencies.*
- *Managing the house's doors by opening and closing them.*

The Appliances control functionalities might increase or change in the next drafts.

3. Design Considerations

Many of the challenges that must be addressed or overcome before attempting to build a complete design solution are described in this section.

3.1 Assumptions and Dependencies

In this section, we will describe any assumptions or dependencies regarding the system and its use. These assumptions and dependencies are divided into subsections:

- *Related software or hardware.*
- *Operating systems.*
- *End-user characteristics.*
- *Possible and/or probable changes in functionality.*

3.1.1 Software or Hardware

The mobile application can be run on any Android device. In terms of programming, the Arduino board and other components will be programmed using the Arduino IDE and the C programming language. Cloud computing is used to embed the database within the mobile application. In the mobile application development studio, in order for the program to work, the phone must run on the smart phone operating system. The mobile device should be an smartphone device and must have a sufficient memory space to install the mobile application.

3.1.2 Operating System

All smartphone OS systems are compatible with the system (latest version). The source code for the system will be written in C and will run on any smartphone device.

3.1.3 End-User Characteristics

The primary end user of the project is the homeowner. In section 2.3 of the Software Requirements Specification (SRS), general characteristics of the specified users are listed.

3.1.4 Possible Changes in Functionality

The end of the design phase may differ from the beginning of the design phase. It's possible that old plans won't be realized, or that better designs with improved interfaces may be developed.

3.2 General Constraints

This section describes any global limits or limitations that have a substantial impact on the design of the system. Such constraints may be imposed by any of the following (the list is not exhaustive):

- *Hardware or software environment.*
- *End-user environment.*
- *Availability or volatility of resources.*
- *Standards compliance.*
- *Interoperability requirements.*
- *Interface/protocol requirements.*
- *Data repository and distribution requirements.*
- *Security requirements (or other such regulations).*
- *Memory and other capacity limitations.*
- *Performance requirements.*
- *Network communications.*
- *Verification and validation requirements (testing).*
- *Other means of addressing quality goals.*
- *Other requirements described in the requirements specification.*

3.2.1 Hardware or Software Environment

The project application should work and run on any cloud-based mobile device. To store and access the essential data, this application will be connected to a chosen backend.

3.2.2 Interface Requirements

The proposed application must provide all users with user-friendly interfaces that are free of ambiguity.

3.2.3 Data Repository and Distribution Requirements

Blynk Cloud will be used to store data depending on the ease of use or the efficiency of integrating the backend with our project.

3.2.4 Security Requirements

To utilize the application, the user must first create an account. Users will be able to access the program when their username and password have been verified and registered to authenticate their account, in order to ensure the application's security. A "forgot password" page is also available for users who have forgotten their password. This will allow the user to use their registered email to change their password.

3.2.5 Performance Requirements

All performance requirements specified in the SRS document, section 3, of the proposed system are required to be tested. Further enhancements may be required if any of the defined performance measures failed.

3.2.6 Verification and Validation Requirements (Testing)

At this stage, verification and validation are used to ensure that the application meets the requirements. It enables us to correct the detected fault as soon as possible.

4. User Interface Design

4.1 Overview of User Interface

The Login page allows the user to sign into the system. It is the major interface in the suggested mobile application. If the user has already registered with the system, he or she will be able to access his or her account. If the user forgets his or her password, there is a "lost password?" link. There is a new user registration link for first-time users, where they must complete the registration form while keeping in mind basic formatting requirements. Entering a unique username, password, and email address are examples of these fields. The user will be directed to the home page after clicking the login button. Each end-user has the same interface as the system's main user, but distinct privileges and authorizations. The user will see two main functionalities which are:

Connection: By entering a valid password, users should be able to connect to mobile application and establish a connection.

Control Appliances: Users can control appliances by selecting the ones they want to operate.

4.2 Screen Images

This section displays the initial design of the mobile application interfaces. Some Interfaces may be added, removed, or modified in later revisions.

4.2.1 Login

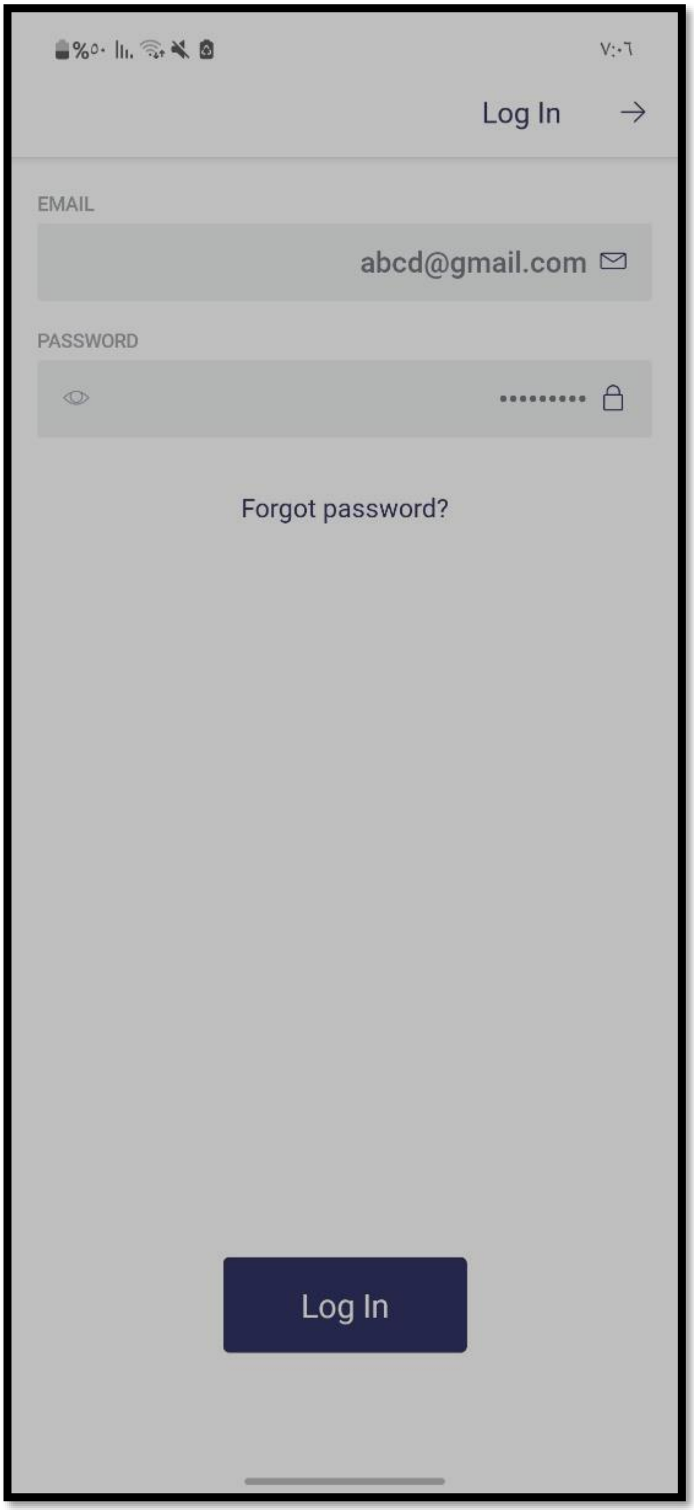


Figure 9 - login page

4.2.2 Registration page

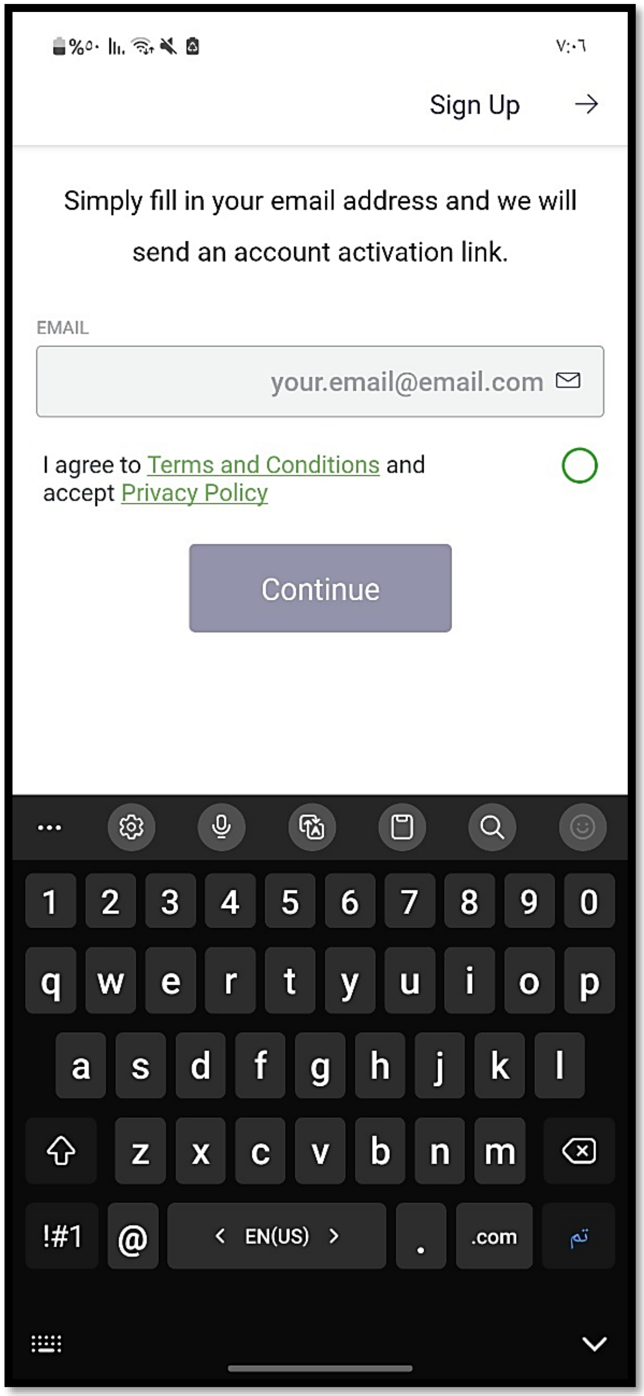


Figure 10 - registration page

4.2.3 Change Password interface

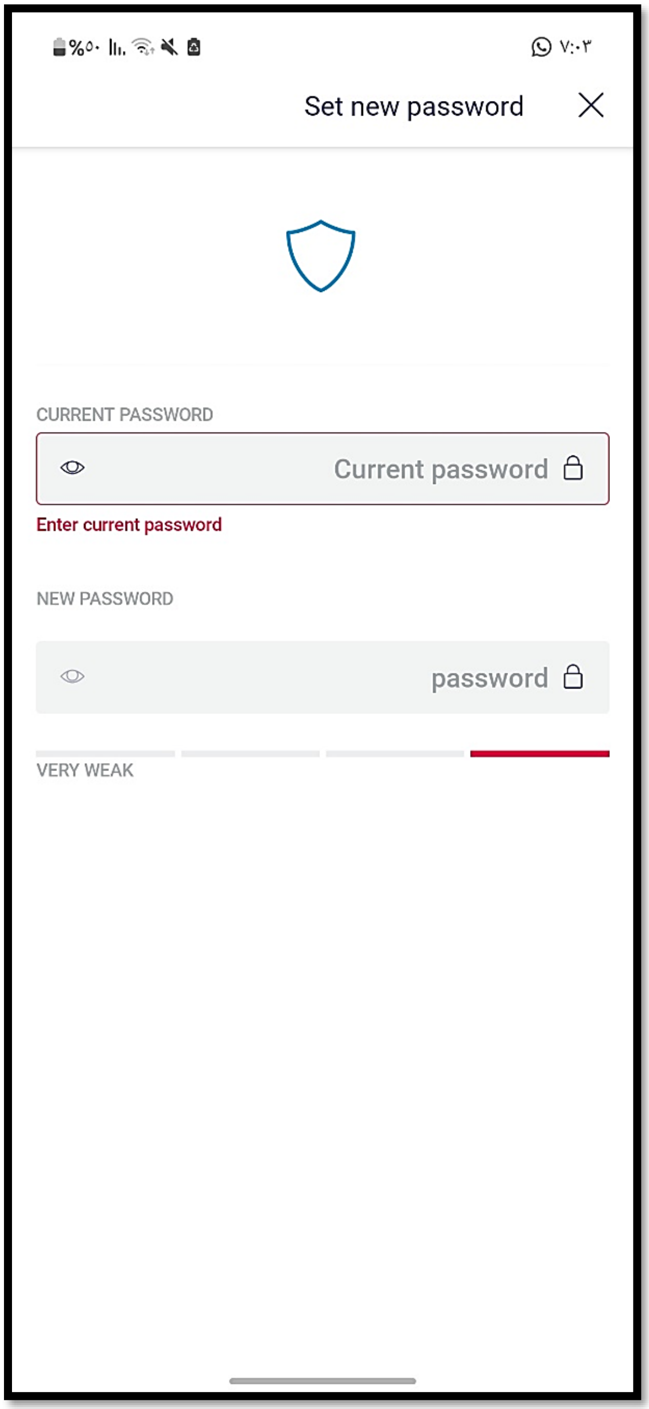


Figure 11- Change password page

4.2.4 Forgot Password UI

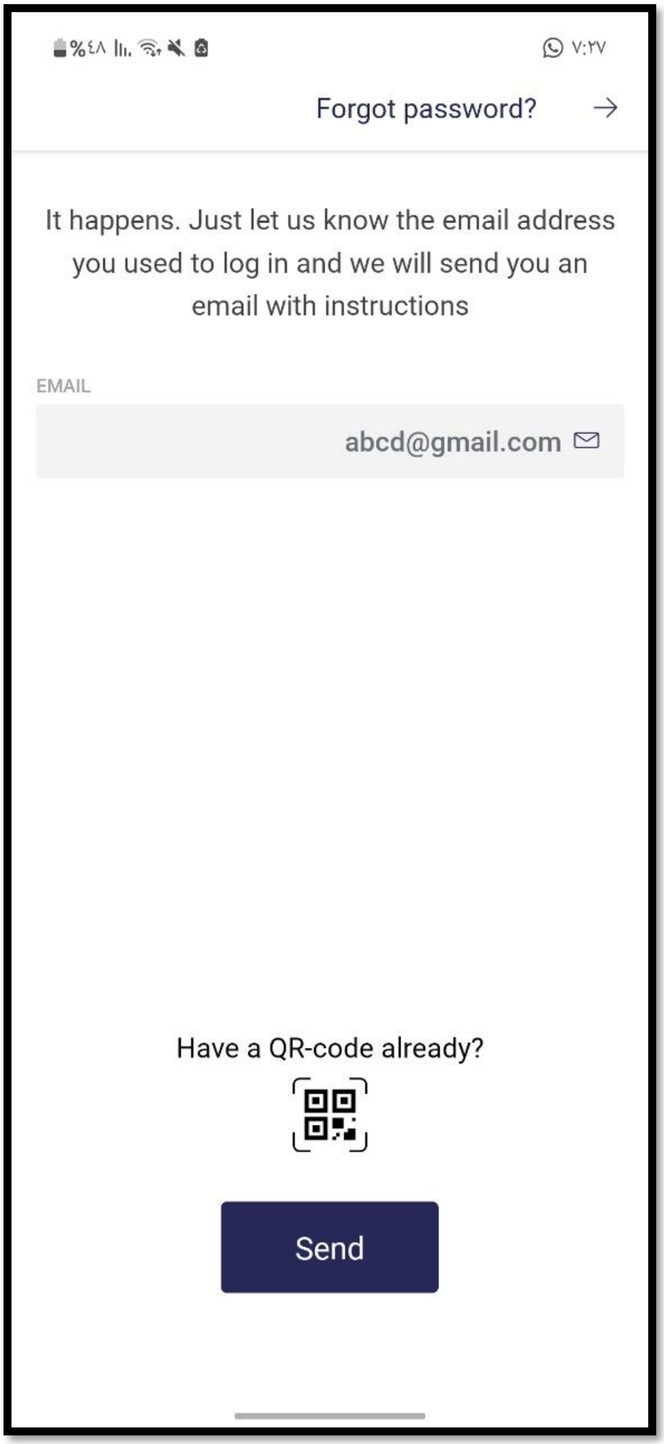


Figure 12- Forgot my password page

4.2.5 Control panel

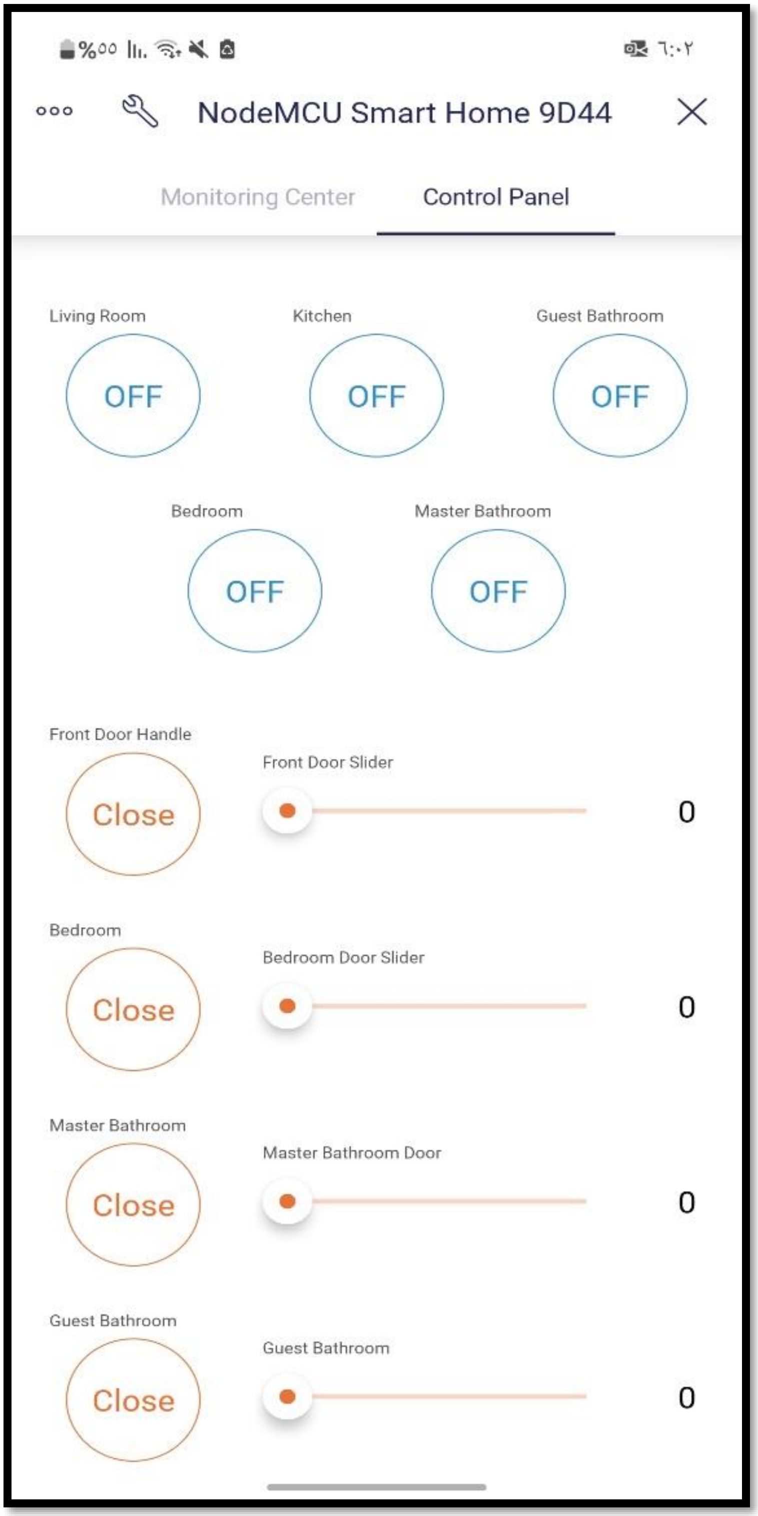


Figure 13- Control panel

4.2.6 Monitoring panel

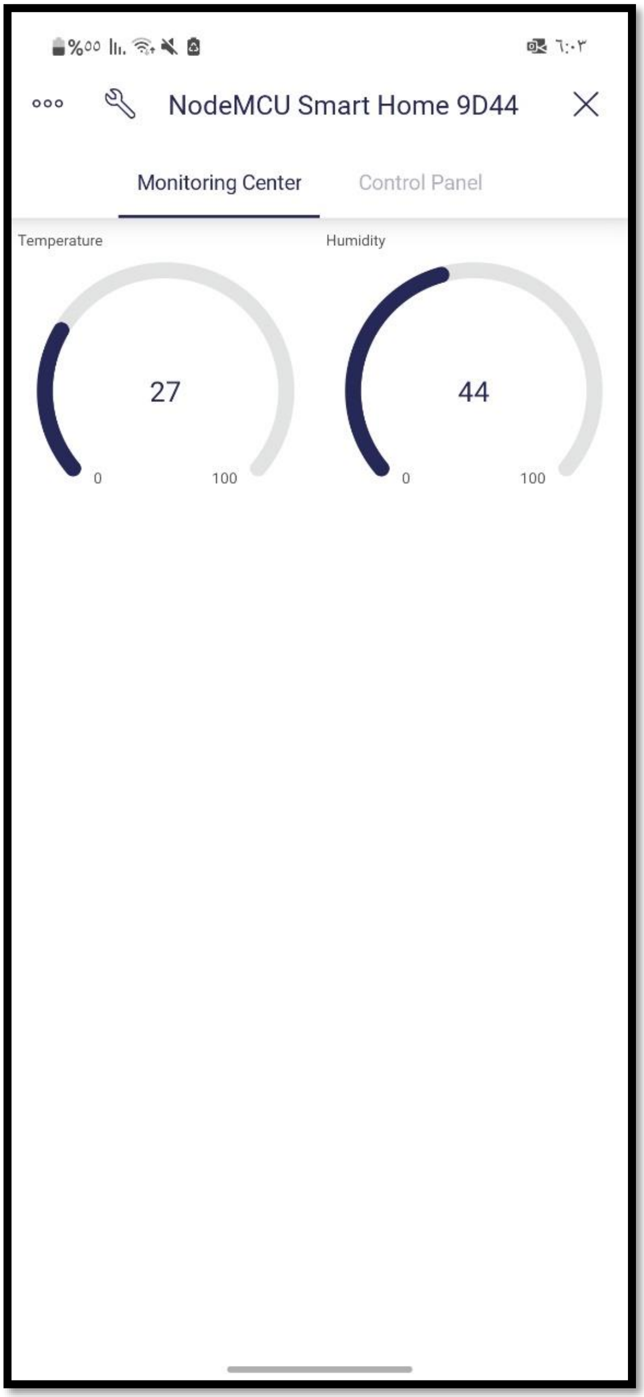


Figure 14 - Monitoring panel

4.2.7 My profile interface

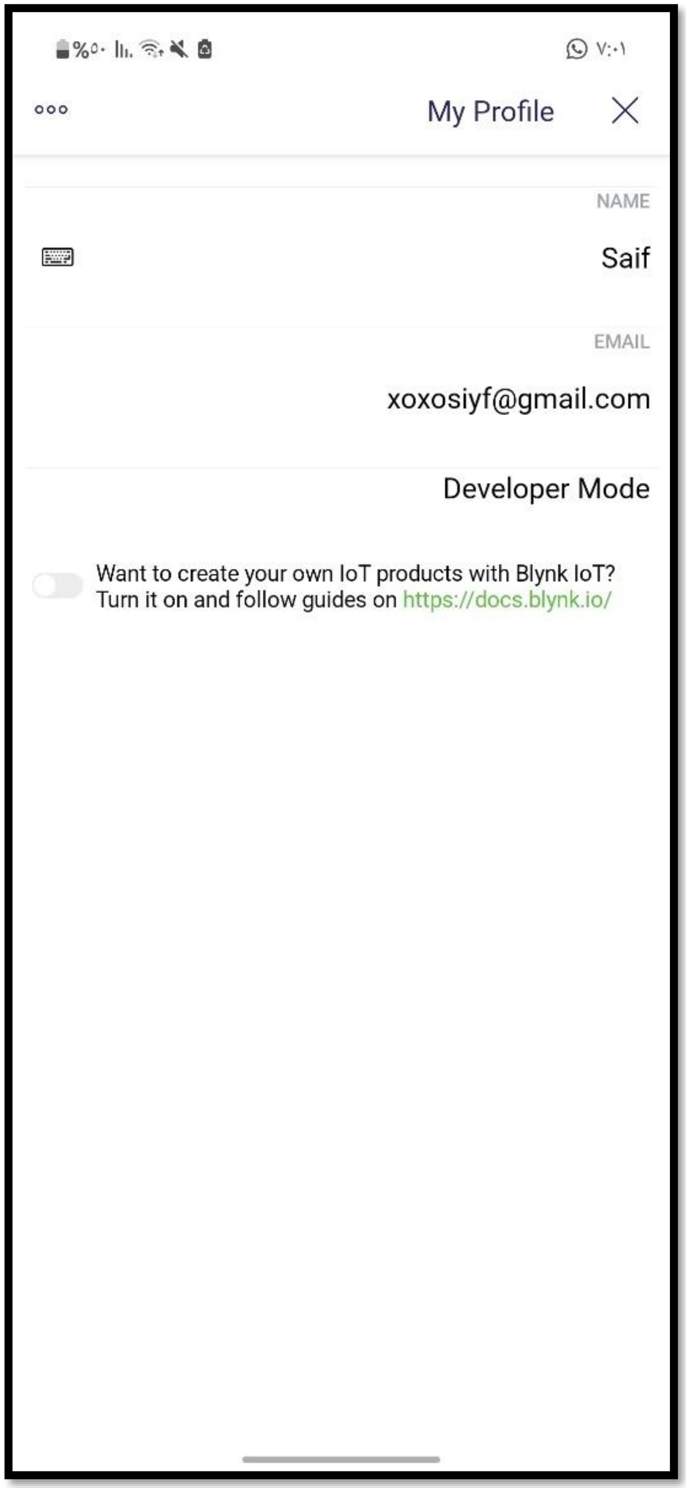


Figure 15- my profile interface

4.2.8 Change username page

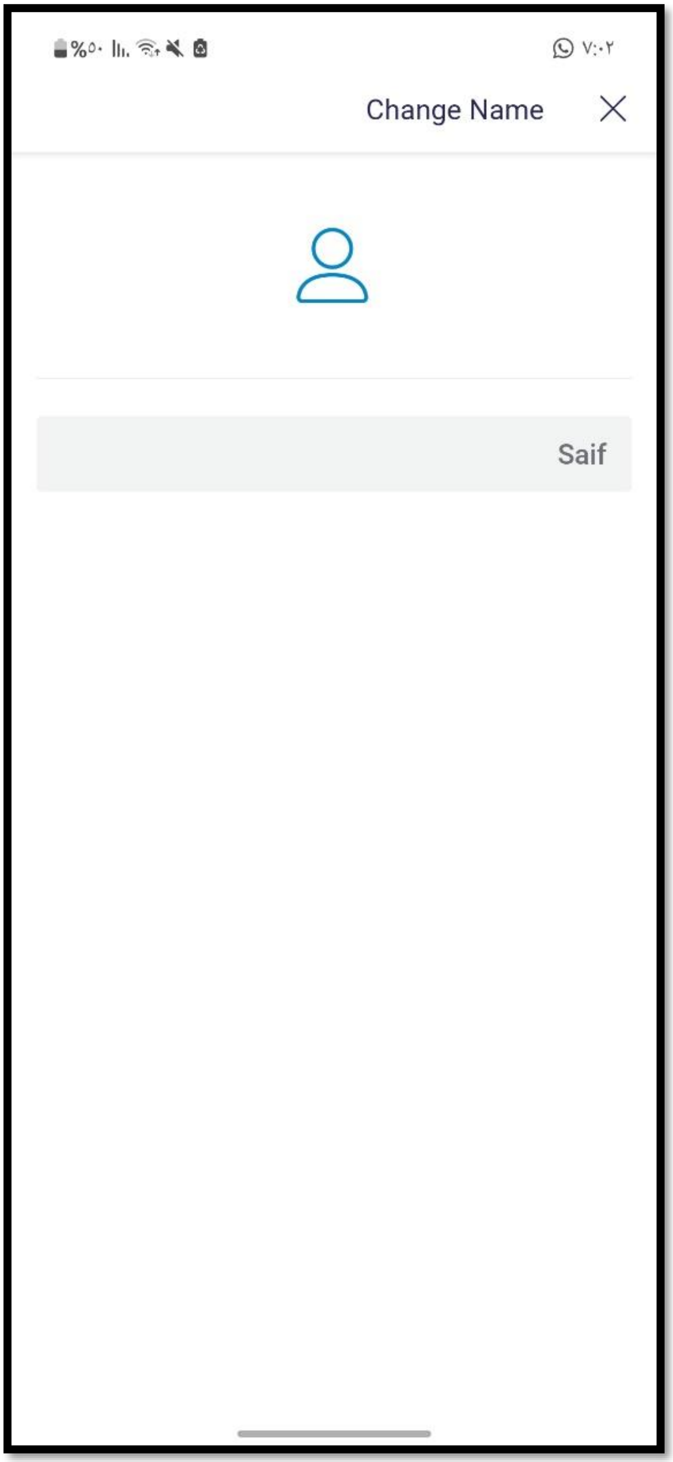


Figure 16 - change username page

4.2.9 Log out pane

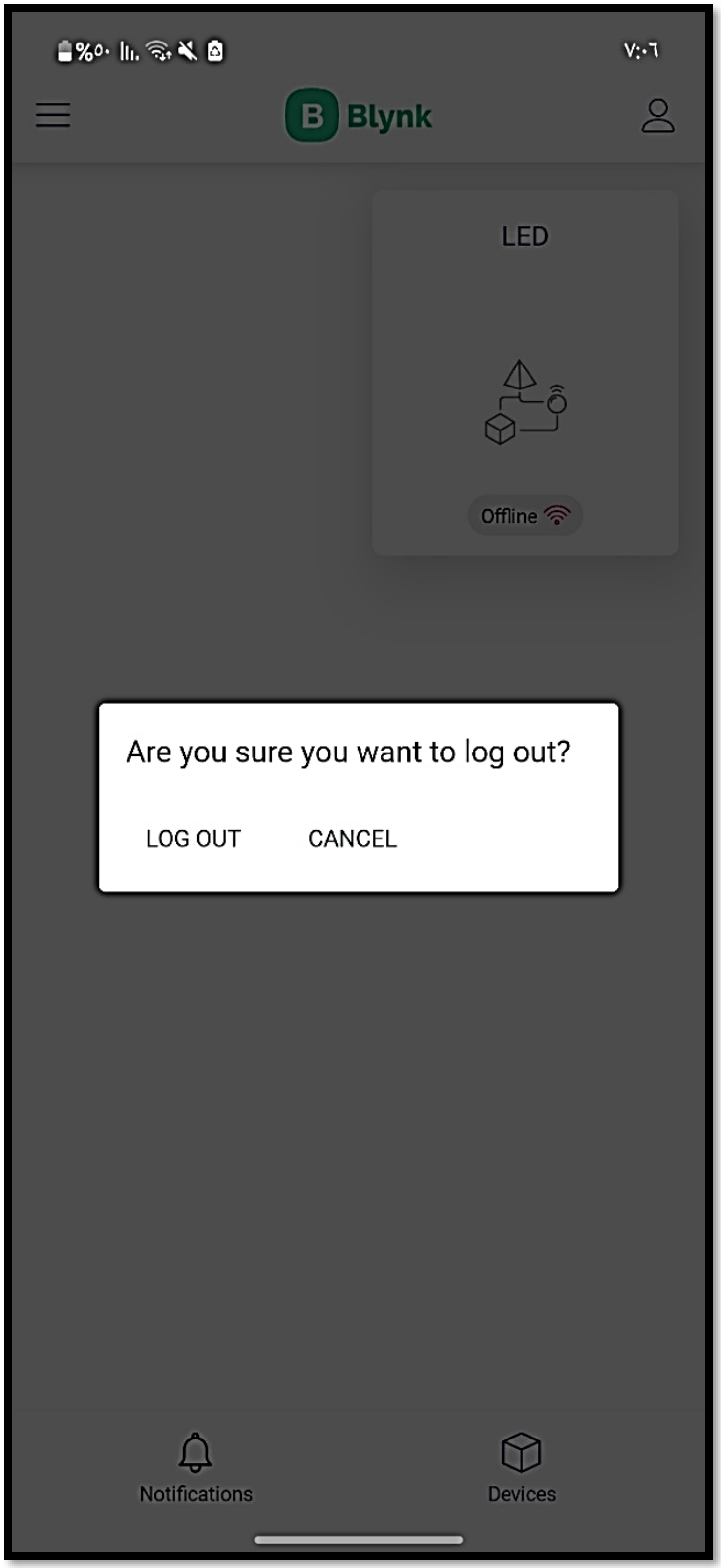


Figure 17- log out alert message

4.2.10 Profile options pane

Figure 18 profile options pane

4.2.11 Navigation side panel

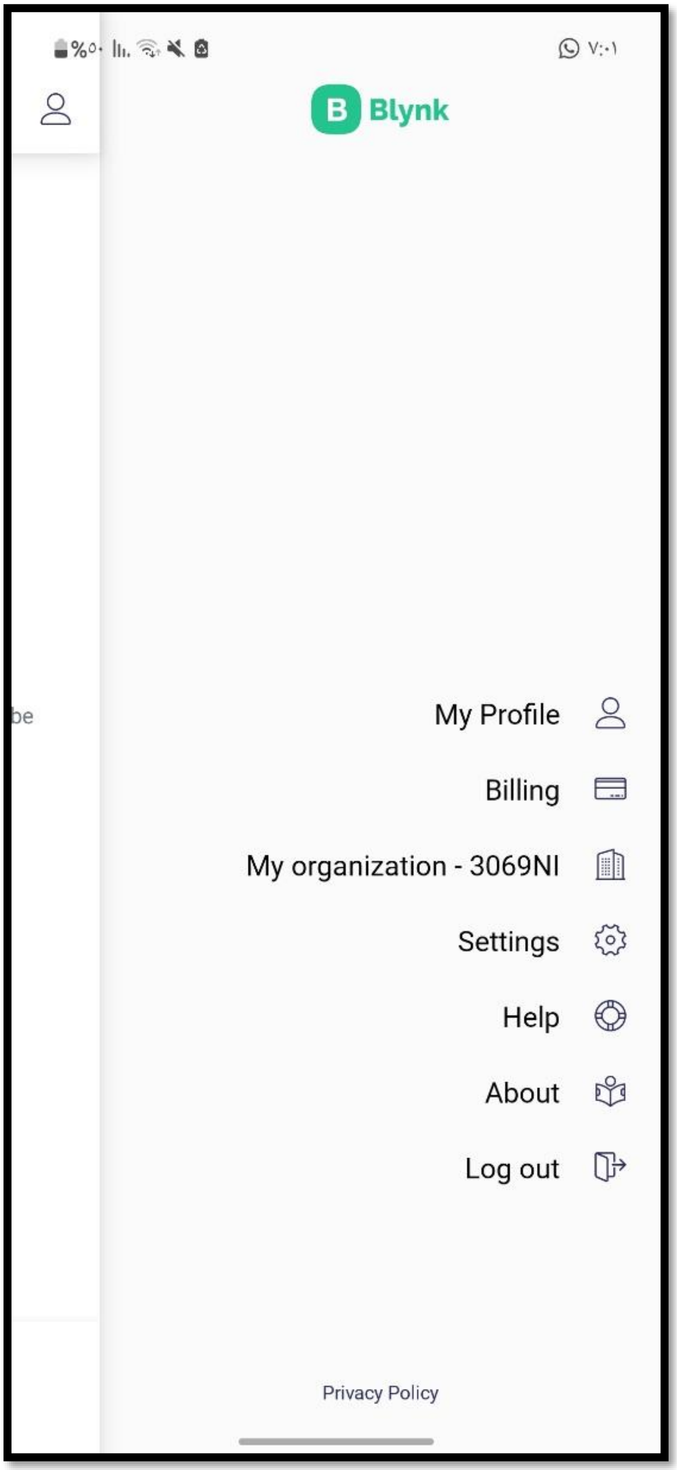


Figure 19 - navigation panel

4.3 Screen Objects and Actions

A discussion of screen objects and actions associated with those objects.

Table 20 Objects and Actions Table

Object	Type	Action
Login	Button	On click, check whether the account is valid then log in the user
Create New Account	Link	Redirects the user to the registration page on click
Forgot Password	Link	Redirects the user to the forgot password page on click
Register	Button	On click, the system checks some conditions on the user input and if it passes, creates a new account for the user, if it fails, show an error message
Back	Button	Redirects the user to the previous screen
Change Password	Button	On click, the system checks some conditions on the user input and if it passes, updates the password for the user, if it fails, show an error message
Send Email	Button	On click, the system checks some conditions on the user input and if it passes, sends a randomly generated password to the user email, if it fails, show an error message
Logout	Button	On click, disconnect the connection to the device and end the session
Home	Button	Redirects the user to the Home Page where the main buttons are displayed
help	Button	Redirects the user to the Tutorial page that describes how to operate the application
Settings	Button	Redirects the user to the settings page that contains user information, setting up connection, and logout button
Lights buttons	User Interface element	Contains buttons to control the lights in the SH remotely
Doors and Windows buttons	User Interface element	Contains buttons to control the doors and windows in the SH remotely
Sensors display	User Interface element	Displays information on the different sensors located in the SH. A sensor will blink if it senses danger based on some preset conditions

Temperature display	User Interface element	Displays information about the temperature in the different locations in the SH.
Setup Connection	Button	Redirects the user to the setup connection page where the user will be asked for a password to connect to the device.

5. System Architecture

This section should provide a high-level overview of how the functionality and responsibilities of the system were partitioned and then assigned to subsystems or components.The main purpose here is to gain a general understanding of how and why the system was decomposed, and how the individual parts work together to provide the desired functionality.

5.1 Architectural Design Approach

Describe the architectural design approach. We took the 5-layer architecture design approach. The system is divided into 5 layers as follows:

Business layer: Organizes the entire IoT system, including apps, business and profit models, and user privacy.

Application layer: oversees providing the user with application-specific services. It outlines several applications for the system.

Processing layer: The middleware layer is another name for it. Data from the transport layer are stored, analyzed, and processed by it. It has the ability to manage and provide a variety of services to the lower layers. Cloud-based processing modules are among the technologies used.

Transport layer: Wireless networks are used to send sensor data from the perception layer to the processing layer and vice versa.

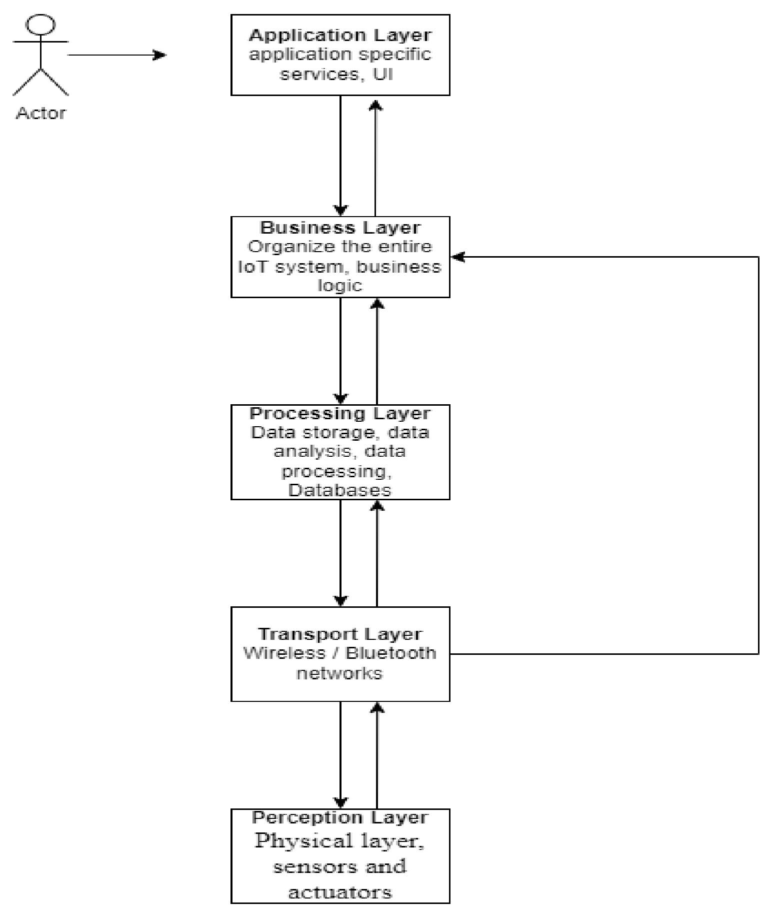
Perception layer: is the physical layer, which contains sensors for sensing and gathering environmental data. It detects certain physical factors or recognizes other smart items in the vicinity.

5.2 Architectural Design

We will provide and describe a figure that depicts the overall system architecture. This is a high-level overview of how responsibilities of the system were partitioned and then assigned to subsystems. The main purpose is to gain a general understanding of how and why the system was decomposed, and how the individual parts work together.

Our Architectural design is shown in the figure below.

Table 21Architectural design Table



Application layer: The user interacts with the system presented through a graphical user interface to present and view data. It also includes tools for users to enter data into the system.

Business layer: Organizes the entire IoT system, including apps, business and profit models, and user privacy. Deals with the business logic associated with the system.

Processing layer: Data from the transport layer are stored, analyzed, and processed by it. Backend processing modules inCloud-based. Cloud-based is among the technologies used.

Transport Layer: Wireless network are used to send sensor data from the perception layer to the processing layer and vice versa. Data may also be sent directly to the business layer for processing and display to the user interface in the application layer.

Perception Layer: the physical layer, which contains sensors for sensing and gathering environmental data.

5.3 Subsystem Architecture

This section provides a decomposition of the subsystems in architectural design.

5.3.1 General View of the System

The figure below illustrates how the user interacts and exchanges information with the system.

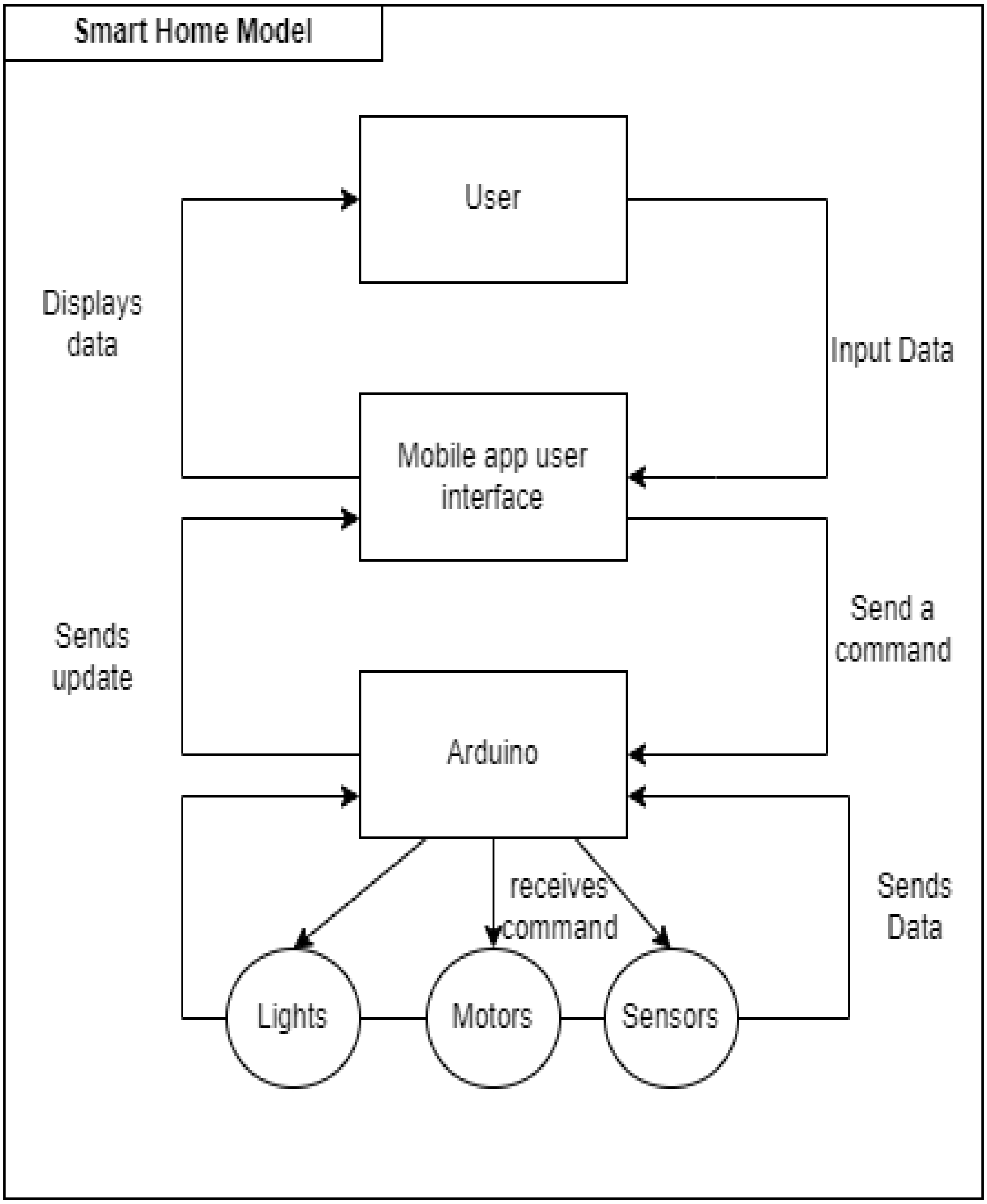


Figure 20General View of the System

5.3.2 User Subsystem

The figure below illustrates the common cases of user interaction with the mobile application to exchange information with the system.

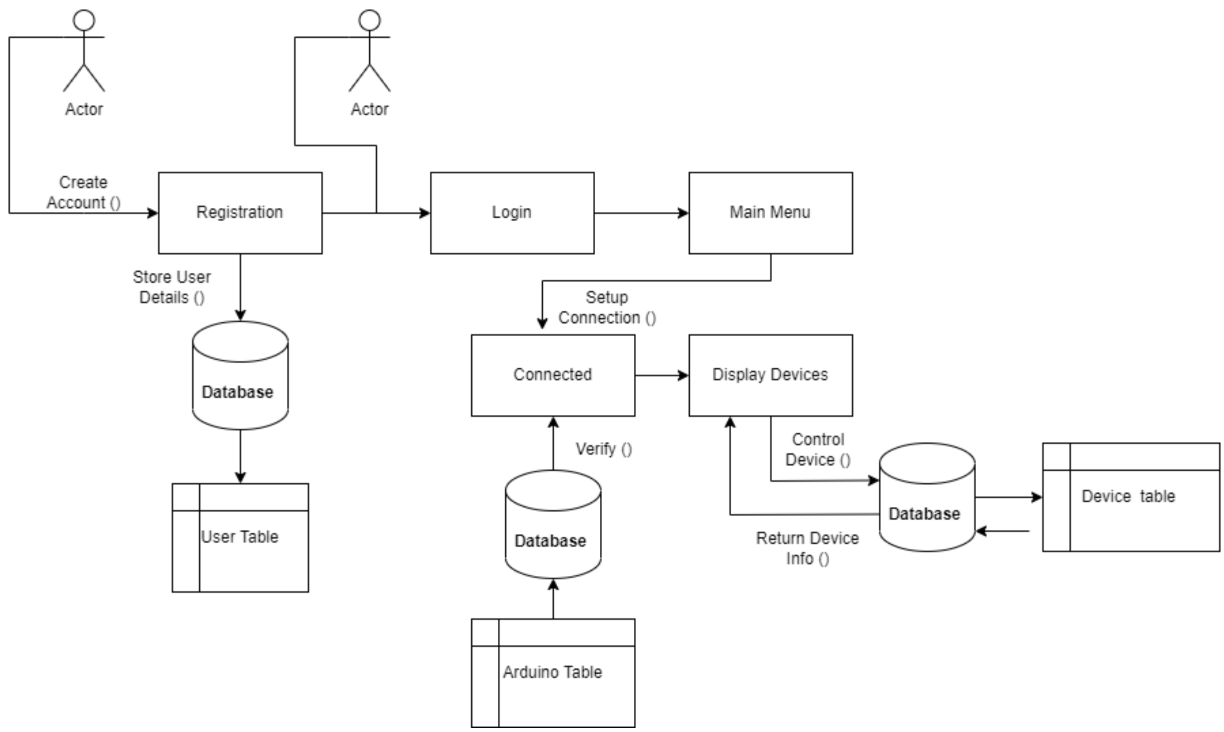


Figure 21User Subsystem

6. Component Design

In this section, we take a closer look at what each component does in a more systematic way. This section will provide a summary of the algorithm for each function listed previously in procedural description language (PDL) or pseudocode. Consider that this is the initial design and changes may be made.

The pseudocode for the common tasks is illustrated as follows:

6.1 Login

```
void Login(String username, String password){

    // If username exists in database
        // If password entered match password stored in database
            // login successfully
        // else show error dialog -> incorrect password
    // else shoe error dialog -> incorrect username
}
```

Figure 22 Login pseudocode

6.2 Registration

```
void Register(String username, String password, String email, String dob){

    // If no fields are empty
        // if the username is unique
            // if password is valid
                // if email is valid
                    // register the user
                // else show error message -> invalid email
            // else show error message -> invalid password
        // else show error message -> username not unique
    // else show error message -> one or more empty fields
}
```

Figure 23 Registration pseudocode

6.3 Forgot Password

```
void forgotPassword(String email){

    // If email format is valid
        // If username exists in the database with the associated email
            // send a random password to the associated email account
        // else show errpr -> user does not exist
    // else show error -> invalid email

}
```

Figure 24 Forgot Password pseudocode

6.4 Change Password

```
void changePassword(String oldPassword, String newPassword, String confirmPassword){
    // if no empty fields
        // if newPassword is valid
            // if newPassword and confirmPassword match
                // update user password
            // else show error -> newPassword and confirmPassword does not mathc
        // else show error -> newPassword is invalid
    // else show error -> there are empty fields
}
```

Figure 25 Change Password pseudocode

6.5 Setup Connection

```
void setupConnection(String arduinoName, String arduinoPassword){

    // If arduinoName exists in the database
        // if the password is correct
            // setup connection function
        // else show error message -> incorrect password
    // else show error message -> arduino does not exist

}
```

Figure 26 Setup connection pseudocode

6.6 Control Device

```
void controlDevice(Input userButtonClick){

    switch(userButtonClick){

        case "Lights": // call function lights()
        break;
        case "Door": // call function door()
        break;
        case "Window": // call function window()
        break;
        case "Smoke sensor": // call function sensors()
        break;
        case "Fire sensor": // call function sensors()
        break;
        case "Gas sensor": // call function sensors()
        break;
    }
}
```

Figure 27 Control Device pseudocode

6.7 Lights

```
void lights(){

    // if selected
        // switch on
    // else switch off
}
```

Figure 28 Lights pseudocode

6.8 Doors

```
void doors(){

    // if selected
        // open doors
    // else close doors
}
```

Figure 29 Doors pseudocode

6.9 Windows

```
void windows(){

    // if selected
        // open windows
    // else close windows
}
```

Figure 30 Windows pseudocode

6.10 Sensors

```
Int Sensors(){

    // if selected
        // return sensor readings

}
```

Figure 31 Sensors pseudocode

7. Detailed System Design

Each subsection of this section will refer to or contain a detailed description of a system software component.

7.1 Classification, Definition, and Responsibilities

This section describes the different components of the system and their responsibilities.

Table 22Component Table

Component	Classification	Definition and Responsibilities
Login	Function	This function allows users to log in to the system by entering their usernames and passwords.
Registration	Function	This function allows the application's first-time user to register by filling out a registration form.
Forget Password	Function	If the user forgets his or her password, this function allows them to receive a new one via email.
Setup Connection	Function	This feature enables the user to connect via Wi-Fi.
Control Devices	Function	The user can use this function to control the devices.
Logout	Function	Stops connection and ends user session.
Change Password	Function	The user can change his or her password using this function.
Edit settings	Function	The user can update his setting on this page.

7.2 Constraints

The following table illustrates the constraints and limitations of our system:

Table 23Constraints Table

Component	Limitations	Pre-condition	Post-condition
Register	No empty fields, username must be unique, password must be valid.	Fill out the form with all of the required information. Use the correct format for the E-mail and password fields.	Account will be registered, they can now login to the system.

Login	Users must have an account registered, no empty fields.	Enter a valid username and password.	Entered data will be checked against the database, users can then access the system.
Forget Password	Must provide a valid email. Users must have an account.	The e-mail address entered must be in a valid format. To receive a new password, the e-mail address entered must be associated with an existing account. Press the "Send" button.	A random password will be generated and sent to the user's email.
Edit settings	No empty fields, Valid input formats, Unique username.	All format requirements are met.	The information in the user's account will be updated.
Setup Connection	A user must have an account and a session exists	Choose a device to connect to, enter the password correctly	Connection established
Change Password	Must provide a valid password. Users must have an account.	The new password must match with the confirmation password	Password updated
Logout	The user should have an account to login and logout.	The user should be logged in.	User log out of the account.
Control devices	A user must have an account.	A user must be logged in to the system with an existing account.	Devices controlled from the mobile application.

7.3 Uses/Interactions

7.3.1 Login

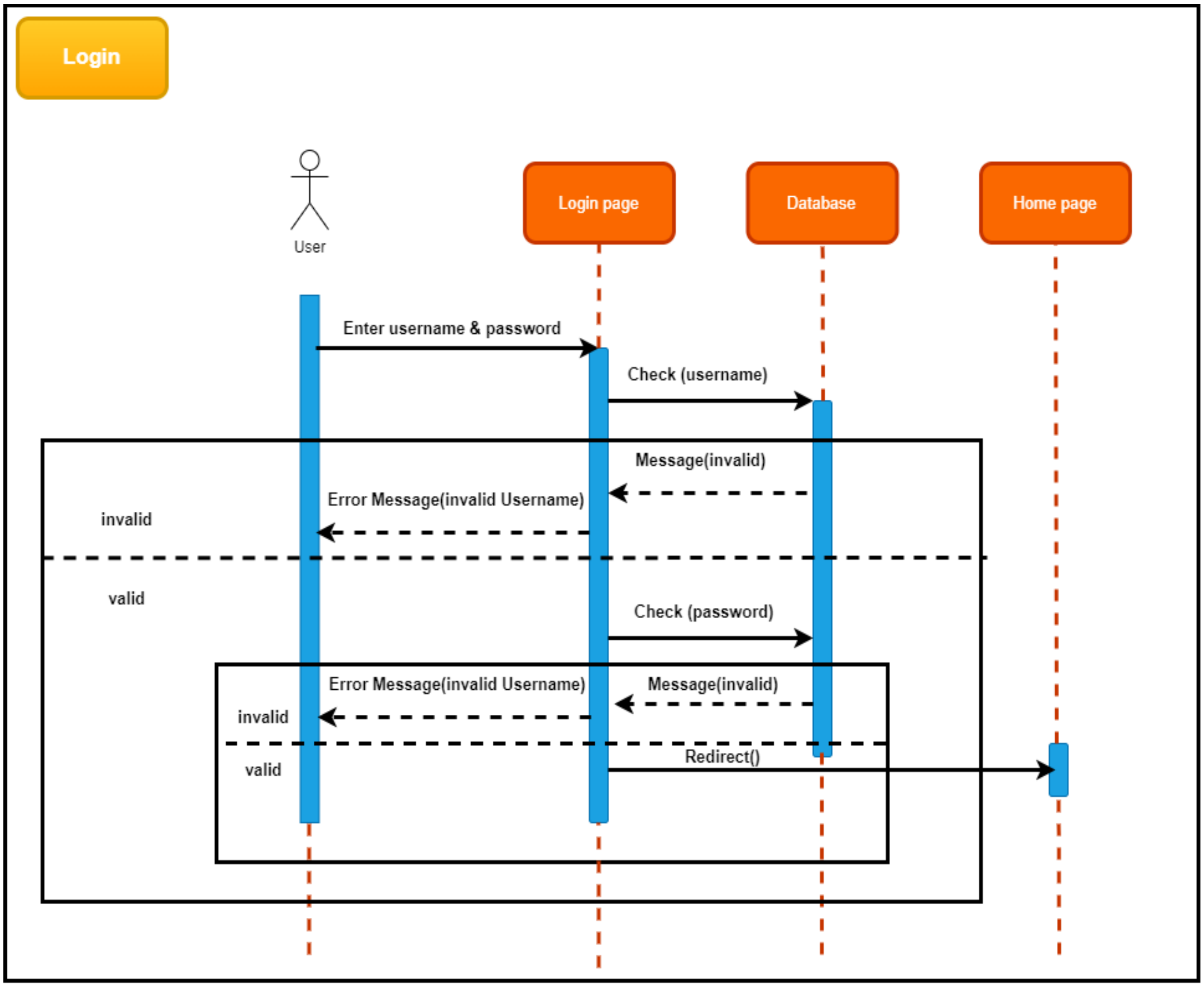


Figure 32 Login Uses/Interactions

7.3.2 Forget Password

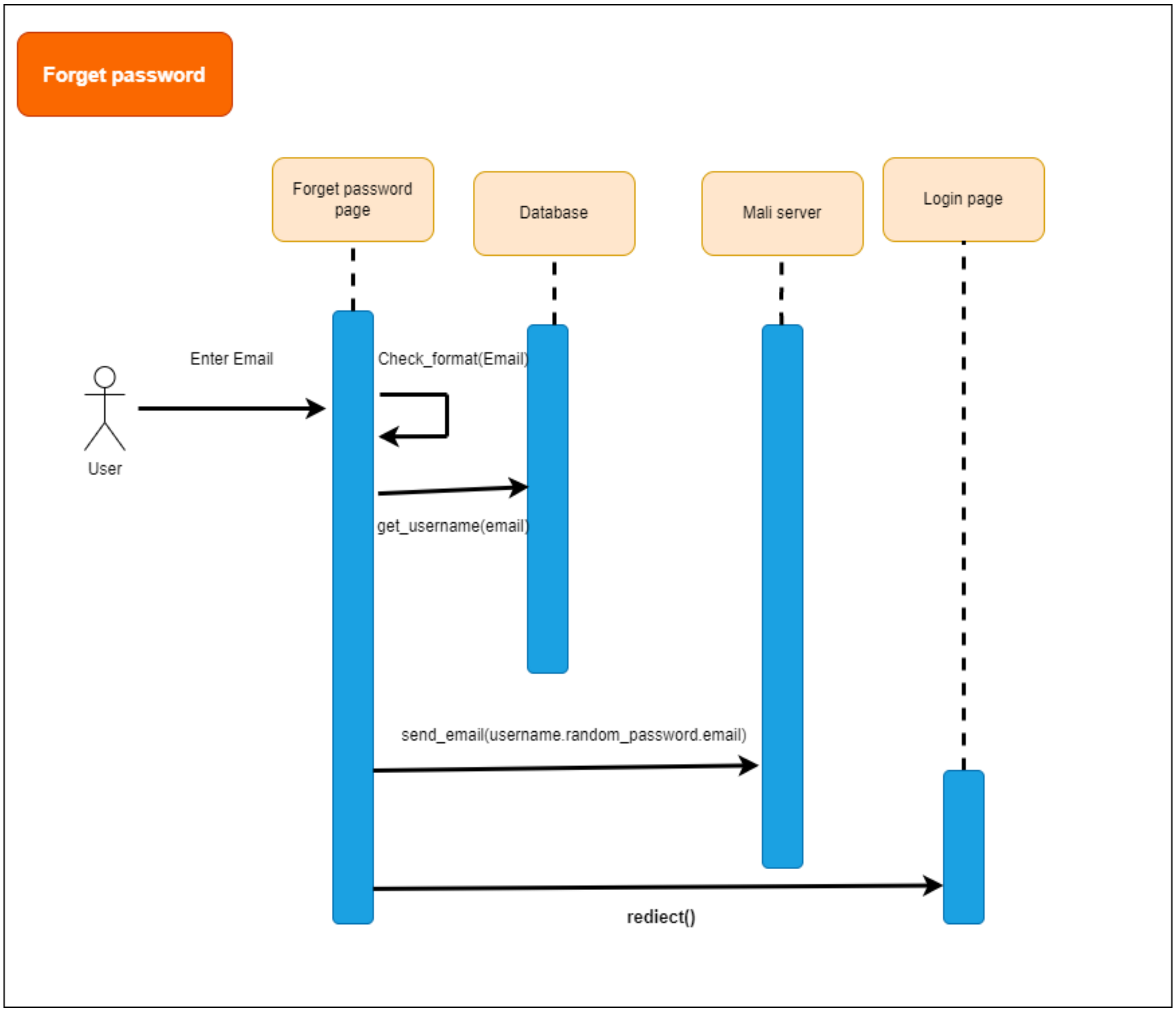


Figure 33 Forget password Uses/Interactions

7.3.3 Registration

Figure 34 Registration Uses/Interactions

7.3.4 Settings

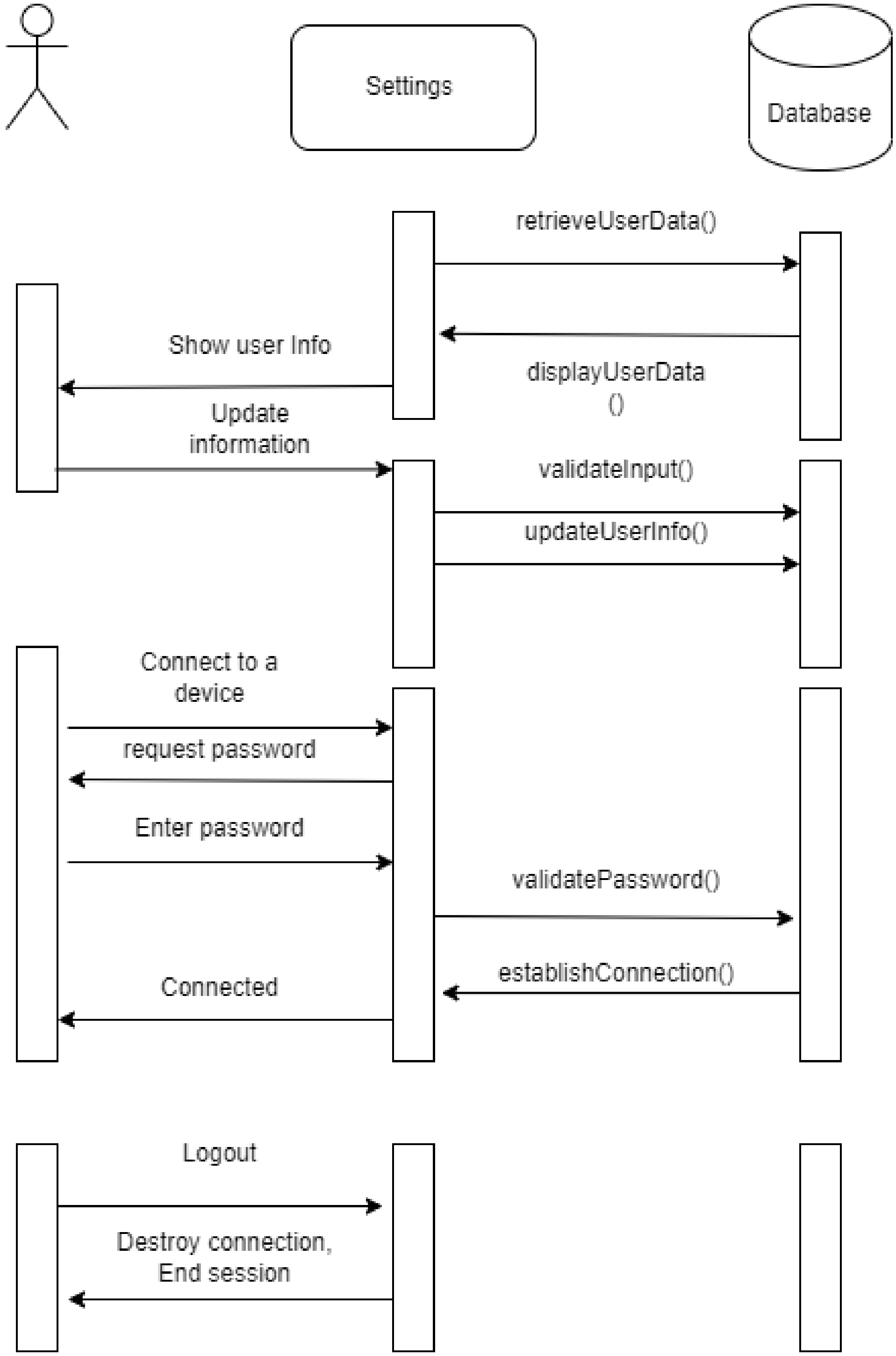


Figure 35 Settings Uses/Interactions

7.3.5 Lights Page

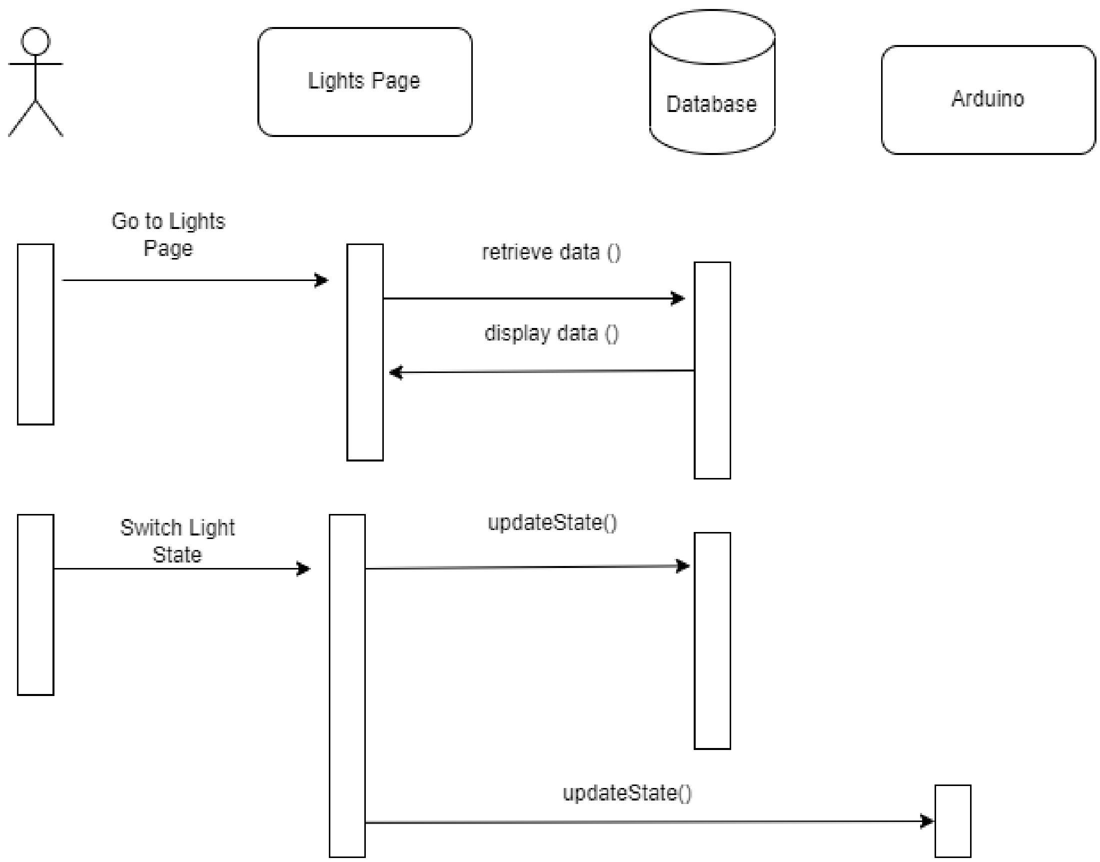


Figure 36 Light page Uses/Interactions

7.3.6 Temperature

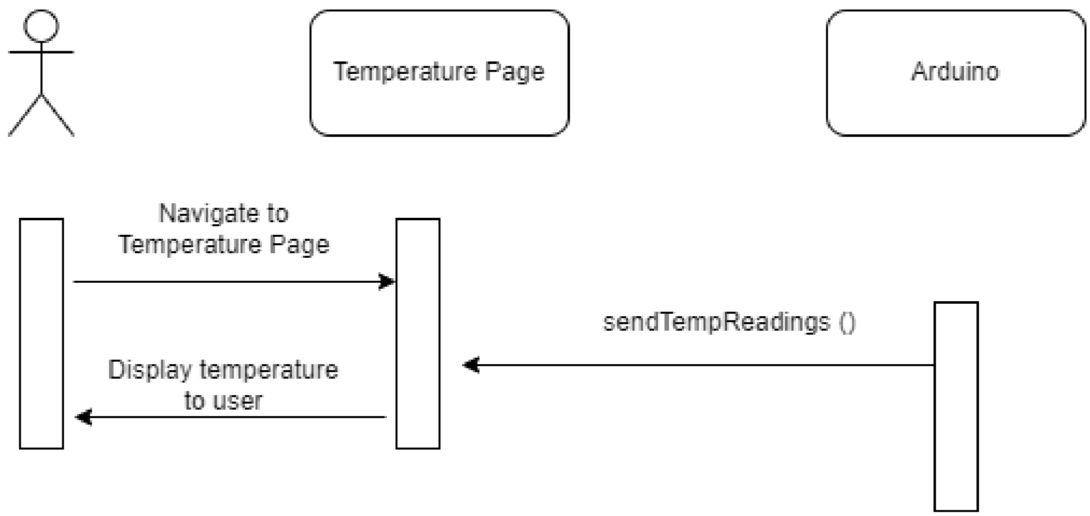


Figure 37 Temperature Uses/Interactions

7.3.7 Window / Doors

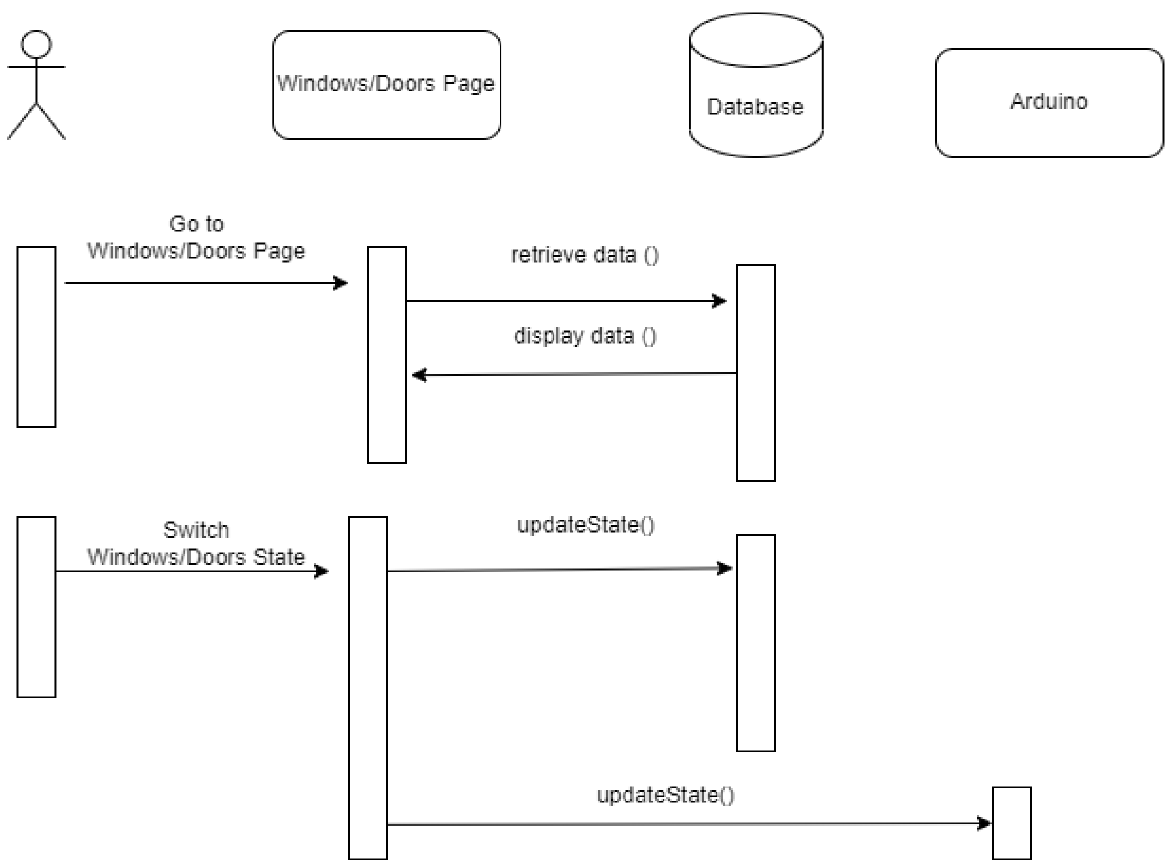


Figure 38 Windows & Doors page Uses/Interactions

7.3.8 Sensors

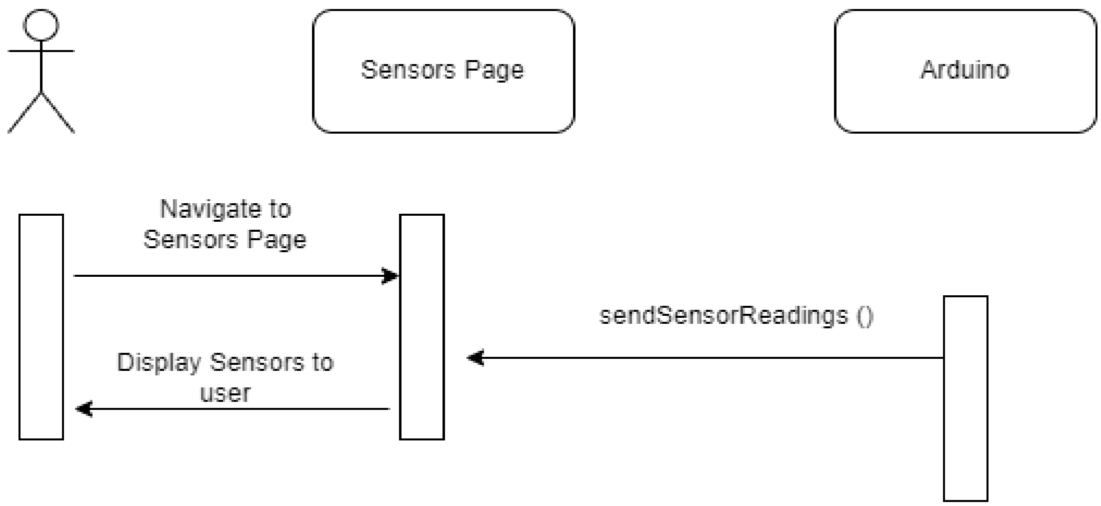


Figure 39 Sensors Uses/Interactions

7.4 Resources

Table 24 Resources

Type	Resources
Database	SQLite database or Fire base or SQL server.
OS	Mobile phone with Operating System
Wi-Fi Connection	Wi-Fi enabled PC, Phone, and mobile application modules
Hardware Resources	Arduino Kits, Android mobile phone, PC with Windows OS, Microbit kit.

7.5 Processing

A description of precisely how the components go about performing the duties necessary to fulfil their responsibilities.

7.5.1 Login

Table 25 Login Processing Table

Description	This function allows users to log in to the system by entering their usernames and passwords.

Input	Username and password
Output	Gain access to the mobile application
Constraints	Users must have an account registered, no empty fields.

7.5.2 Registration

Table 26 Registration Processing Table

Description	This function allows the application's first-time user to register by filling out a registration form.
Input	Username, password, email and confirm password.
Output	Accounts will be registered, they can now login to the system.
Constraints	No empty fields, username must be unique, password must be valid.

7.5.3 Forget Password

Table 27 Forget Password Processing Table

Description	If the user forgets his or her password, this function allows them to receive a new one via email.
Input	Email
Output	A randomly generated password will be sent to the user's email
Constraints	If the user forgets his or her password, this function allows them to receive a new one via email.

7.5.4 Edit Settings

Table 28 Edit Settings Processing Table

Description	The user can update his setting on this page.
Input	Username, Email, Password.
Output	The information in the user's account will be updated.
Constraints	No empty fields, Valid input formats, Unique username.

7.5.5 Setup Connection

Table 29 Setup Connection Processing Table

Description	This feature enables the user to connect via Wi-Fi.
Input	Choose a device to connect to, enter the password correctly.
Output	If the password was entered correctly, the user will see a message that says "The device has been successfully connected."
Constraints	The password entered must match the one supplied by the development team, or the connection will fail.

7.5.6 Change Password

Table 30 Change Password Processing table

Description	The user can change his or her password using this function.
Input	Old password, new password, retype new password
Output	Password will be updated
Constraints	Must provide a valid password. Users must have an account. The new password must match with the confirmation password

7.5.7 Logout

Table 31 Logout Processing table

Description	This function stops the connection and ends the user session
Input	Logout button clicked
Output	Disconnects from the device and ends the session.
Constraints	The user must have an account and have an existing session open

7.5.8 Control Devices

Table 32 Control Devices Processing table

Description	Users should be able to view a list of appliances and select which ones they want to control.
Input	Choose a device to control.

Output	Display the page of the device.
Constraints	To be controlled, each device must be connected to the Arduino and the user most be logged in.

7.6 Interface/Exports

To guarantee that the blynk operates as intended, developers should be familiar with the features and requirements of each subsystem. Similarly, the application's inputs, outputs, pre-conditions, post-conditions, needed processing, and constraints must all be fully understood. To guarantee that the system operates as intended, developers should be familiar with the features and requirements of each subsystem.

7.7 Detailed Subsystem Design

Refer to the SRS for section 3.2 Functional Requirements.

8. Requirements Traceability Matrix

Provide reference to the location of the Requirements Traceability Matrix that indicates traceability from the system requirements documented in the System Requirements Specification to the design elements documented in the System Design Description. Use a tabular format to show which system components satisfy each of the functional requirements from the SRS. Refer to the functional requirements by the numbers/codes that you gave them in the SRS.

Table 33Matrix Table

SRS Section	Technical Assumptions	Functional Requirement	SDS Section	System Component
3.2.1	The user should be able to access the system by entering a username and password.	Login	7.1	Login
3.2.1.2	A user will receive a randomly generated password by e-mail.	Forget Password	7.3	Forget Password
3.2.2	A first-time user should complete the registration form before using the application.	Registration	7.2	Registration
3.2.3	The main page that the user will be redirected to after logging in. It also has the main functionalities of the Smart Home	Home	4.3.5	Home Page
3.2.3.3.2	Change the users Password	Change Password	7.4	Change Password
3.2.3.3.3	Establish a connection with the Microcontroller device by providing a valid password	Setup Connection	7.5	Setup Connection
3.2.3.2	A UI that displays all possible devices to control	Control Device	7.6	Control Appliance
3.2.3.2.1	Switch lights on/off by pushing a button	Lights	7.7	Lights

3.2.3.2.2	Lock or unlock doors/Windows by pushing a button	Doors/Windows	7.8	Doors
3.2.3.3.4	Display sensors readings in the GUI	Sensors	7.10	Sensors

Chapter 6: Testing and implementation

1.Application Overview

MSSH system is a Smart home application can control basics faculties. Light can turn on and off, also real time opening doors and close when is needed. There are several modules like smoke deduction and gas leak.

2.Testing Scope

A functionality verification that needs connectivity to a third-party application cannot be tested, but the connectivity could be established due to some technical techniques. This section should be clearly documented, else it will be assumed that Testing covered all areas of the application.

In-Scope: Functional Testing for the following modules are in Scope of Testing

-Turning light on/off.

-Opening/Closing doors and windows.

Out of Scope: Performance Testing was not done for this application.

Items not tested: Verification of connectivity with the third-party system ‘Central repository system’ was not tested, as the connectivity could not be established due to some technical limitations. This can be verified during UAT (User Acceptance Testing) where the connectivity is available or can be established.

Functional Testing is included in the scope of testing for the following modules.

-Turning on/off the light

-Closing and opening doors and windows.

This application was not subjected to performance testing. Not tested items include: The connectivity with the third-party system 'Central repository system' was not verified since it could not be created owing to technical constraints. This may be confirmed during UAT (User Acceptance Testing) if connectivity is present or can be created.

3. Metrics for testing

- No. of test cases planned vs executed

Thing been Tested	Test Cases planned	Test Cases Excuted	TCs Pass	TCs Failed
Doors and Windows	36	30	16	12
Lights	15	15	12	12
Flame Dictator	2	1	1	1
Gas Dictator	2	1	0	0
Temperature	2	1	1	1

4. Testing Methodologies

4.1 Smoke Testing: This testing was done anytime a Build was received (deployed into a Test environment) for Testing to ensure that the primary functionality was operating properly before accepting the Build and starting testing.

4.2 System Integration Testing: This is testing done on the program under test to ensure that it functions properly. - Critical Business scenarios were evaluated to ensure that the application's most significant features functioned correctly and without issues.

4.3 Regression Testing:

-Regression testing was done every time a new build was delivered for testing, which included defect corrections and any new features.

-Regression testing is performed on the entire program, not just new features and bug patches.

-This testing guarantees that existing functionality continues to function properly when defects are fixed, and new features are introduced to the program.

-New feature test cases are added to existing test cases and performed.

5. Lessons Learned

S. No	Issues faced	Solutions
1	Smoke testing test cases required to be executed manually each time.	Smoke test cases were automated and the scripts were run, which ran fast and saved time.
2	Initially, Few testers were not having rights to change defect status in HP QC/ALM. Test lead need to perform this task.	Rights were obtained from Client, by explaining the difficulty.

Figure 40 - lessons learned

5.1. Recommendations

When the work related to hardware first you need to study about circuits. also, to decide which mobile application to work on its better to take a full view on that app. you need to do a lot of experiment on the equipment's to gain more knowledge.

5.2. Exit Criteria

All test cases should be run -True.

All problems of varying severity should be verified and closed - True.

Any open faults of minor severity - Prepare an action plan with projected closure dates.

Chapter 7: Progress report

1. Introduction

With the advent of the communications technology, automation systems have received great attention from the relevant communities around the globe. A notable application is the so-called Internet of Things (IoT). IoT refers to the interconnection of the electronic devices beyond computers and smartphones that exchange data with other devices over the internet or other communication systems. A current example of such systems is smart homes, smart cities, smart security systems, smart cars, smart factories, autonomous agriculture, and many other systems that rely on the electronic devices to communicate with each other without the human intervention.

In this project, we will design and implement a Smart Home (SH). It is an IoT application that allows users to remotely manage and monitor household equipment. SH is a home with an automated system that includes sensors, actuators, and controllers to improve comfort, automation, safety, and security for occupants. Smart gadgets, such as smartphones, smart televisions (TVs), smart washing machines, smart refrigerators, and smart sensors, have become ubiquitous in people's everyday lives in the modern world. These intelligent gadgets can communicate and interact with one another to create a smart environment.

To manage communication between smart devices within SHs, an automated system should be built. In this regard, many automation systems have been created, some of which are currently being sold as commercial goods as well. Some of such items are used to manage the household appliances remotely or locally.

The cost of electronic control has quickly reduced since the introduction of microcontrollers at the turn of the century, and home automation has developed. Home automation systems are not commonly used, and they are still seen as the realm of enthusiasts or the wealthy folks. Various automation technologies have been utilized to equip household appliances with the needed systems for simple monitoring and control, such as remote control for TVs, electric fans, air conditioners, and music players. With an increase in the use of electricity in homes and quick progress of information technology, a new age of home appliance control has begun utilizing mobile devices with short-range communication interfaces such as the, Wi-Fi, and among others.

All these systems and technologies are beneficial for controlling home appliances from within the house, but they do not allow people to monitor and manage their houses from outside. Although such systems always allow contact with inhibitors to promote ease, comfort, safety, and energy efficiency, their usefulness and communication range are limited. Most of these systems don't make the use of IoT's strong characteristics, which seeks to connect billions of smart devices (phones, computers, sensors, and actuators) to Internet.

Objectives:

- *Propose a low-power, low-cost home automation system using IoT,*
- *Design and implement the proposed IoT system, and*
- *Document the different project development phases (SPMP, SRS, SDS, and final report).*

2. Summary of the Proposal

The purpose of this project is to design and implement an automated Smart Home using IoT technology. To meet the planned outcome and result of the project, which is a fully functional prototype of a smart home. We will implement this project with safety and convenience, including the following:

1. *Design and build a fully functional prototype of a smart home,*
2. *Implement low energy and resources saving techniques,*
3. *Integrate security functions and measures,*
4. *Build and integrate the circuitry of related devices to the smart home, and*
5. *Connect hardware and software via cloud computing (Blynk platform).*

This project is related to IoT and Smart Automation Systems. A field of interest that is currently a growing market and has many relevant applications. Many research papers have also been published regarding many different applications of IoT. The Smart Home project can be integrated with other Smart Automation Systems, such as smart appliances that were developed independently.

3. Changes from the Proposal Phase and Justifications

3.1 Platform:

Our application is going to be implemented through the Blynk IoT platform.

Blynk is an IoT platform for businesses and developers with a fully integrated suite of IoT software development tools. Being one of the biggest IoT community, Blynk provides mobile and web app services, simple APIs, a variety of supported hardware devices, connection to the cloud, updates through network, and flexible connection of devices to a network using their Wi-Fi provisioning tools. Blynk is the perfect platform for the implementation of our project.

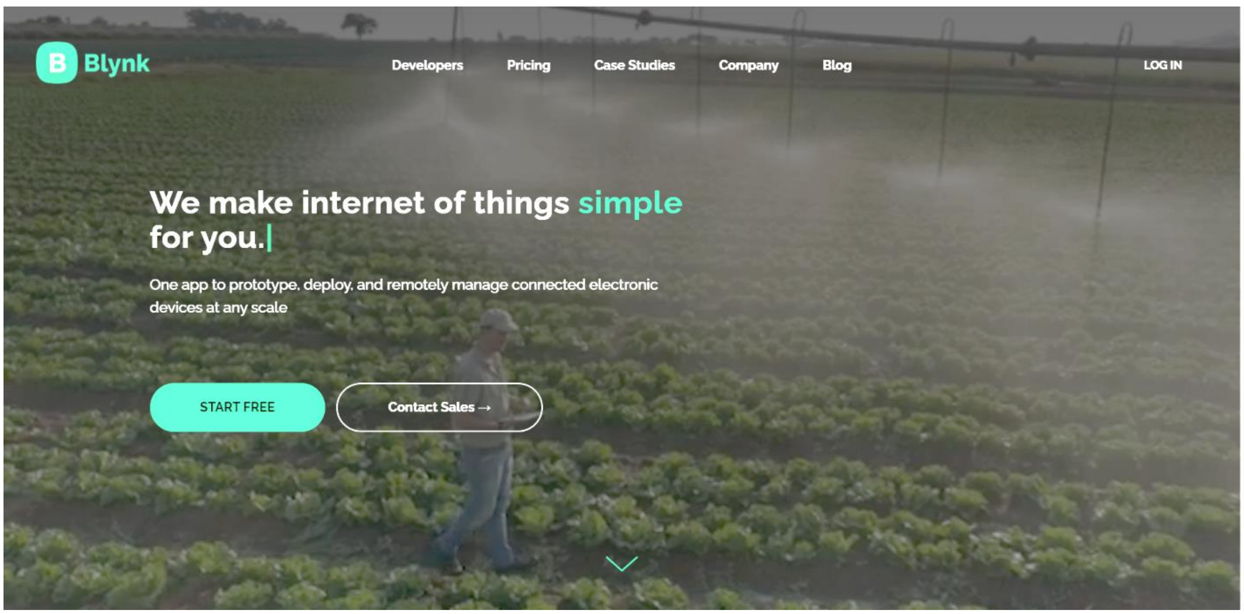


Figure 41 Blynk Homepage

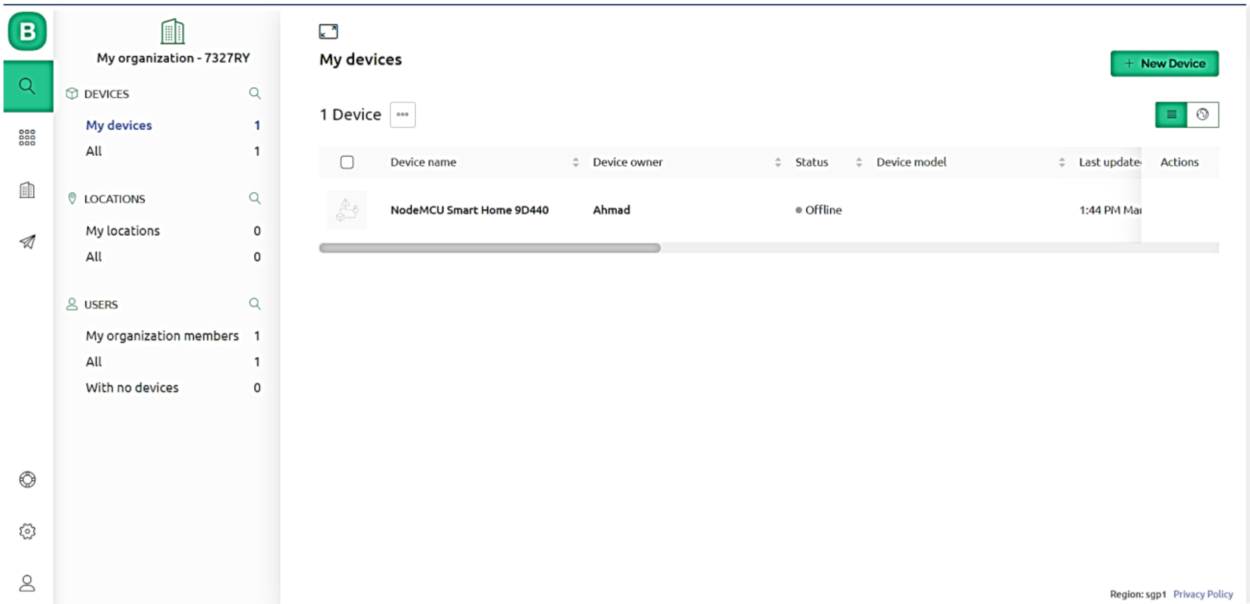


Figure 42 Blynk Dashboard

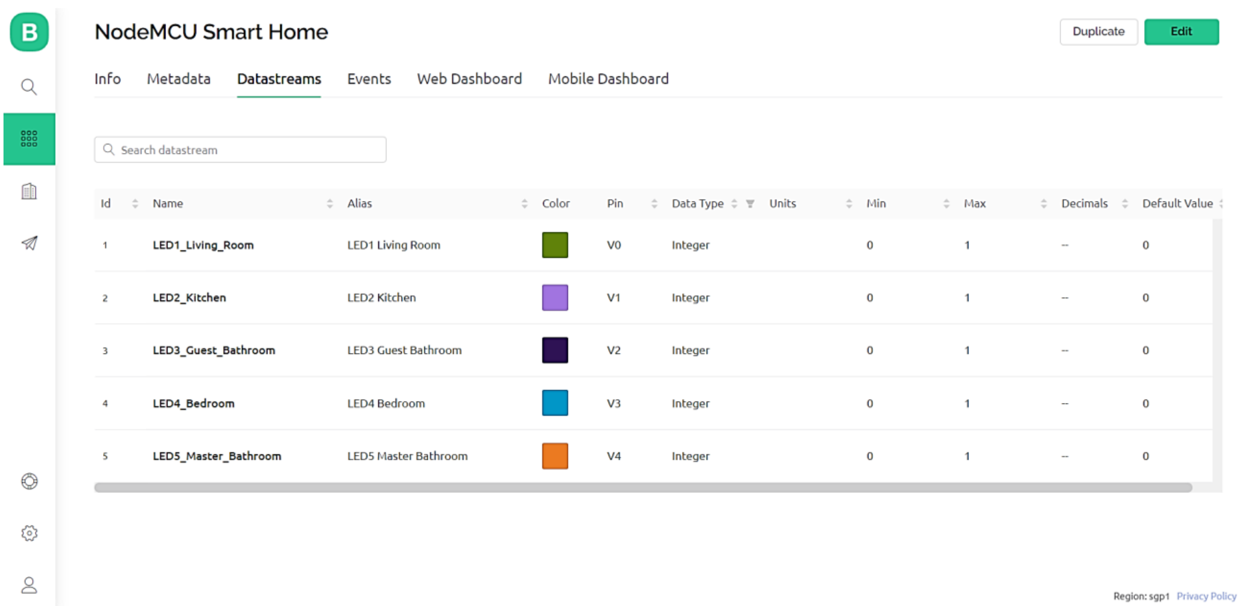


Figure 43 Configuring Data streams in our NodeMCU template

Above are some of the screenshots of the Blynk website and template. The website homepage redirects a user to their dashboard once logged in. In the dashboard, a user can create or modify their own templates in which they can configure their devices and create data streams to structure data.

3.2 Cost:

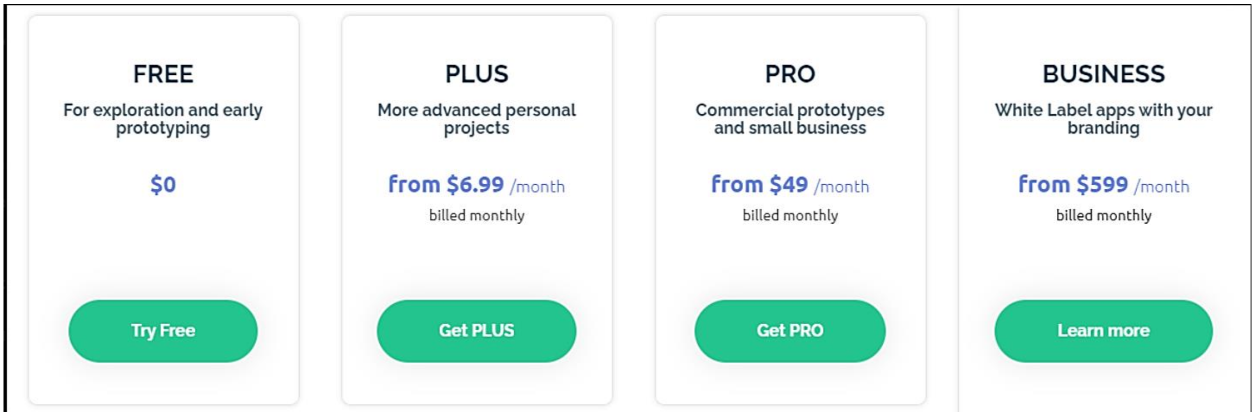


Figure 44 Subscription Costs when Billed Monthly

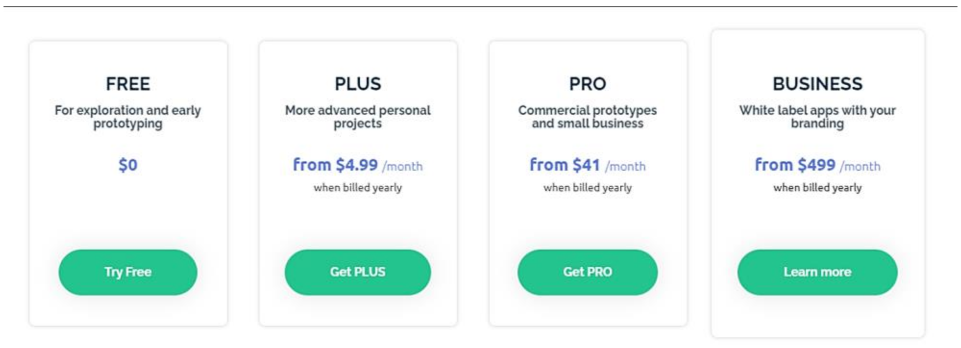


Figure 45Subscription Costs when Billed Yearly

As displayed in the figures above, Blynk offers free services mainly for exploration and prototyping. However, for more advanced personal projects, Blynk will require some form of subscription. The different subscription options contain different services as listed in their website. The following figure shows the differences between each package. However, the tools we need for our projects far is provided for free.

	Free	Plus	Pro	Business
Apps				
Blynk.App for iOS and Android Build UI, connect devices, and remotely monitor them. No-code.	✔	✔	✔	Standalone Mobile App Your own, branded IoT app ready to be published to App Store and Google Play
Adding multiple pages to a device dashboard	-	1 page or 1 device info tab	20 pages, 7 device Info tabs, 1 welcome page	20 pages, 7 device Info tabs, 1 welcome page
Mobile App Widgets	Basic mobile widgets i	PRO mobile widgets i	PRO mobile widgets i	PRO mobile widgets i
Widget limit per template	30	80	255	Unlimited
QR / barcode scanner	-	-	✔	✔
Widget customization	-	-	-	Contact us
Blynk.Console Web portal to configure and manage devices, users, data and organizations.	✔	✔	✔	Branded Web IoT portal with your company branding and URL
Web Console Widgets	Basic web widgets i	PRO web widgets i	PRO web widgets i	PRO web widgets i
Dashboard with multiple tabs	-	1 tab	3 tabs	3 tabs
Localization	-	-	-	Add-on required
Cloud				
Blynk.Cloud Secure cloud server ready to connect your devices.	✔	✔	✔	✔
Historical device data storage	1 week	3 months	6 months	12 months
Data export to CSV	-	✔	✔	✔
Private Business Server Your own private business server set up in your country / region	-	-	-	✔
Custom server configurations	-	-	-	✔

Figure 46 Subscription packages

Connectivity management				
Blynk.Edgent Open-source hardware libraries to connect any device to Blynk.Cloud using WiFi, Ethernet, or cellular (2G, 3G, LTE, etc) connection.	✓	✓	✓	✓
Blynk.Inject Connect devices to local WiFi network directly from Blynk app.	✓	✓	✓	✓
Device management				
Devices included in the plan Device is microcontroller with a unique Auth Token(for example single ESP32)	2	10	40	10,000
Add more devices	-	Upgrade to higher plan: Up to 20: $12.99 /mo	Upgrade to higher plan: Up to 100: $99 /mo Up to 500: $149 /mo Up to 1,000: $229 /mo Up to 5,000: $349 /mo	Upgrade to higher plan: Contact Sales
Static Tokens management	-	-	✓	✓
Alexa and Google Home support	-	Add-on coming soon	Add-on coming soon	✓
Fleet Management	-	-	Add-on coming soon	Add-on coming soon
Blynk.Air Update supported devices over-the-air	✓	✓	✓	✓
Local app to device over-the-air firmware updates	-	-	-	✓
User/Organization manager				
Users included in plan	5	10	40, 100 or 500	10,000
Roles and permissions management	Locked set of permissions for each role	Locked set of permissions for each role	Editable permissions for each role	Editable permissions for each role
Create and manage sub-organizations	-	-	✓	✓
Partner Flow	-	-	-	Add-on required

Figure 47 Subscription packages II

Support and Services				
Technical support	Forum only	Forum only	Ticketing system with 1-3 business days reply	Dedicated launch engineer
Uptime SLA	-	-	-	✓
Security				
Encrypted communication between Cloud, apps and devices (for supported hardware)	✓	✓	✓	✓
Built-in user verification over email	✓	✓	✓	✓
Device access controls	-	-	✓	✓
24/7 incident monitoring	-	-	-	✓
Full database encryption	-	-	-	Add-on required

Figure 48Subscription packages III

3.3 Security:

Blynk allows data storage in the cloud. And as such they implement several security measures. Some of these measures are as listed:

- *Every message sent through and from the cloud is encrypted and secure.*
- *User permissions are in place to manage who can see a user's devices and data*
- *Every user has a valid and verified email address*
- *Each device has their own unique OAuth token and Product ID*
- *The cloud is monitored constantly to allow quick reactions for potential incidents*.

Blynk servers also relies on the industry standard Transport Layer Security (TLS) protocol as well as for Blynk.Console and mobile apps

The Blynk.Cloud data transfers are encrypted, and database is fully isolated within the private network

3.4 License Issues

Blynk products and services may not be sold for commercial purposes without a paid licensing subscription plan acquired. The full Blynk ToS may be found in their website (https://blynk.io/tos)

3.5 Database

Blynk cloud-of-things platform provides many services for data analytics and as such there will be no use for a local and independent database

3.6 Short of actions

- Lack of practical experience proved to be a huge obstacle in building our circuitry
- A lot of delay in acquiring the required materials and devices such as the power supply, light builds, and servo motors.

4.Implementation Progress

4.1 Devices Testing

Some prominent devices used in this system include the following:

- *TowerPro SG 90 Micro Servo Motor x10,*
- *Relay Board 16 Channel 5V Module x1,*
- *LED lights x8,*
- *NodeMCU ESP8266 x1,*
- *Flame Detection Sensor x2,*
- *MQ-2 Gas Sensor x2, and*
- *Big bread board x1.*

4.1.1 TowerPro SG 90 Micro Servo Motor testing:

Table 34- servo testing

Servo Motor Number	Working (Y/N)
Servo Motor 01	Y
Servo Motor 02	Y
Servo Motor 03	Y
Servo Motor 04	Y
Servo Motor 05	Y
Servo Motor 06	Y
Servo Motor 07	Y
Servo Motor 08	Y
Servo Motor 09	Y
Servo Motor 10	Y

4.1.2 Relay Board 16 Channel 5V Module testing:

The relay board works but it has an issue connecting with the NodeMCU ESP8266. We need to make the signal level compatible. The ESP28266 operates @3.6 VDC and hence it cannot drive the target Relay module having 5VDC interface at its input. Interfacing a simple buffer IC such as 74HCT245 in between these two gadgets would surely solve this issue. Otherwise, we will also try the idea of Pull-Up resistors to pull up the high level (3.6 VDC) of ESP8266 to 5VDC. Another choice is to drive Relay module using Transistors as drivers.

4.1.3 LEDs:

All the LEDs works properly, also with the Blynk Application being installed on the Smart Phone.

4.1.4 Sensors:

All sensors work properly, also with the Blynk Application being installed on the Smart Phone.

4.2 Blynk configurations

We decided to use blynk cloud IoT platform to control and monitor the house via Wi-Fi. This platform allows us to connect our microcontroller to the internet, by pairing each pin on the NodeMCU kit to a virtual pin in the cloud, we can have steady Data Streams that would transfer data between the user and the house through a user interface that is designed and integrated in Blynk platform.

4.3 Software Code

By using the Arduino IDE, we were able to write the necessary codes for each light, servo motor, and any other device being connected in the circuit; and the libraries needed for our codes to run were imported In this platform, the code is written in the C language.

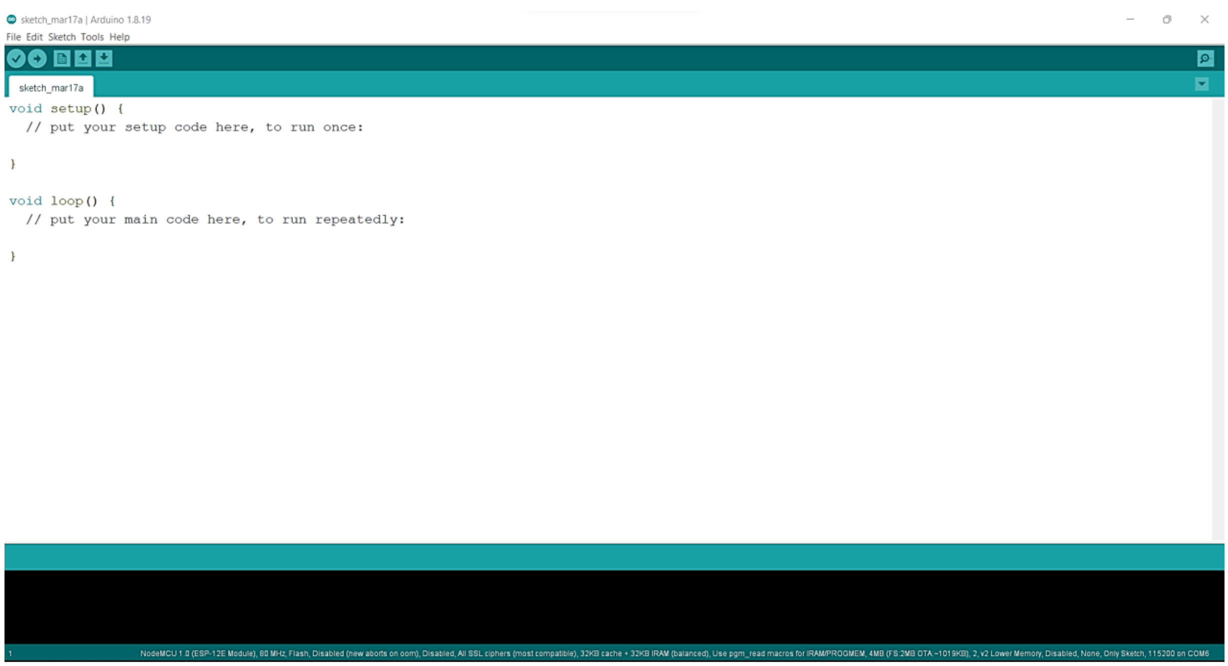


Figure 49 Arduino IDE

4.3.1 Arduino IDE:

The open-source Arduino Software (IDE) simplifies the process of writing code and uploading it to the board. This software is compatible with any Arduino board. The Arduino Integrated Development Environment (IDE), also known as the Arduino Software, includes a text editor for writing code, a message area, a text console, a toolbar with buttons for common functions, and a series of menus. It communicates with and uploads programs to the Arduino hardware.

4.3.2 Writing Sketches:

Sketches are programs created with the Arduino Software (IDE). These sketches are created in a text editor and saved with the. ino. file extension. The editor includes functions for cutting/pasting and searching/replacing text. The message area displays errors and provides feedback while saving and exporting. The console displays text output from the Arduino Software (IDE), including error messages and other information. The configured board and serial port are shown in the bottom right corner of the window. You can use the toolbar buttons to verify and upload programs, create, open, and save sketches, and launch the serial monitor. Additional commands can be found in the following five menus: File, Edit, Sketch, Tools, and Help. The menus are context sensitive, which means that only items relevant to the current task are available. Some details for some of the menus are as listed below:

4.3.3 File

- New Creates a new instance of the editor with the bare bones of a sketch already in place.
- Allows you to load a sketch file by browsing through your computer's drives and folders.
- Open Recent displays a list of the most recent sketches that are ready to be opened.
- Sketchbook Displays the current sketches in the sketchbook folder structure; clicking on any name opens the corresponding sketch in a new editor instance.
- Examples This menu item displays any examples provided by the Arduino Software (IDE) or library. All the examples are organized in a tree, making it easy to find them by topic or library.

4.3.4 Sketch

- Verify/Compile checks your sketch for errors while compiling it and displays memory usage for code and variables in the console area.
- Upload compiles the binary file and loads it onto the configured board via the configured Port.
- Using a Programmer to Upload This will overwrite the bootloader on the board; you will need to restore it using Tools > Burn Bootloader to be able to upload to the USB serial port

again. It does, however, allow you to use the entire capacity of the Flash memory for your sketch. Please keep in mind that this command will NOT cause the fuses to burn. To do so, use the Tools -> Burn Bootloader command.

- Export Compiled Binary Produces a.hex file that can be archived or sent to the board with other tools.
- Display the Sketch Folder This command opens the current sketch folder.
- Include Library Inserts #include statements at the beginning of your code to add a library to your sketch. See the libraries listed below for more information. You can also access the Library Manager from this menu item and import new libraries from.zip files.
- Add File, adds a new file to the drawing (it will be copied from its current location). The file is saved to the sketch's data subdirectory, which is designed for assets such as documentation. Because the contents of the data folder are not assembled, they are not included in the sketch software.

4.3.5 Tools

- Formatting Automatically This beautifully styles your code by indenting it such that the opening and closing curly braces line up and the statements within the curly braces are indented further.
- Archive Sketch Saves a.zip file containing a copy of the current sketch. The archive is saved in the same folder as the sketch.
- Reload and fix the encoding Corrects any mismatches between the editor's char map encoding and the char maps of other operating systems.
- Monitor for Serial Numbers The serial monitor window is opened, and data is exchanged with any connected board on the presently specified Port. If the board allows Reset via serial port opening, this generally resets the board.
- Board Choose the board you're working with. The descriptions of the various boards may be found below.
- Port This menu offers a list of all serial devices (both real and virtual) on your machine. When you access the top-level tools menu, it should automatically refresh.

- Programmer For utilizing a hardware programmer instead of the onboard USB-serial connection for programming a board or chip. Normally, you won't need this, but if you're flashing a bootloader onto a new microcontroller, you will.
- Bootloader should be burned. The choices in this menu allow you to burn a bootloader into an Arduino board's microcontroller. This is not necessary for typical Arduino board operation; however, it is handy if you acquire a new ATmega microcontroller (which normally come without a bootloader). Before burning the bootloader on the target board, make sure you've picked the proper board from the Boards option. This instruction also activated the appropriate fuses.

4.4 Smart Home Prototype

Figure 50 Latest Smart Home Progress

A simple smart home prototype is currently being built in this project. The smart home follows a simple, modern, western style house that consists of 4 rooms namely a living room with integrated kitchen, a guest bathroom, a single master bedroom, and a master bathroom. The blueprint of the smart home was first designed, then built with cut-outs of cardboard boxes. We also added another layer of walls to the house to let the wirings pass freely through the house while concealing them from plain view to preserve a pleasing aesthetic.

5. Issues Faced:

- *Lack of practical experience and advanced knowledge in digital and analog electronics, electronic circuit designing, and troubleshooting since it is not within our domain.*
- *Less coordination among team members.*
- *Having no access to the electronics laboratory.*
- *Provision of setting space at the college.*
- *Lack of sufficient tools needed to conduct electronic experiments such as the multimeter and the Digital-Analog Training System*

6. Testing Plan

The original test plan was to test the mobile application against standard use cases but since we moved to the Blynk IoT platform to build our application, our test plan is no longer necessary.

Currently, we do not have a predefined test plan except to check whether the devices are working properly or not. After the full implementation of the project, we will come up with use cases and test the product against all potential use cases we find.

7. What Have We Learned

While researching for the project, we have accumulated the knowledge necessary to design the layout of the smart house and gained the necessary information in writing the codes for the smart things by using the Blynk API By reading through the API documentation we obtained our fair share of experience from a cloud of things solution that enables us to present an interactive User Interface (UI) layout This whole experience in implementing the project would ultimately prepare us for more IoT projects and implementations.

On the other hand, one of the main challenges we are facing right now is constructing the circuit we need to be able to fully implement the project, such as dealing with the devices and their current and voltage requirements and connecting them all together along with the NodeMCU microcontroller to build an efficient and effective circuit. Finally, we are still in the learning process in the domain of electronics and digital hardware.

8. Conclusion

Overall, the project implementation can be decomposed into three parts. Building the smart home prototype, building the circuitry, and creating a software application. The smart home prototype is almost complete structure-wise except for the interior and exterior design of the house. On the other hand, the current main challenge we are facing right now is building the circuit and dealing with currents and voltages. Once we overcome this challenge, we will proceed to build the software application, which is anticipated to be less challenging as we have more expertise in this field rather than electronics and we have all the resources we require to accomplish it. Finally, once we complete all three parts, we will integrate them all together as one unit and proceed to design the interior and exterior of the house to build a fully functional and presentable smart home.

In conclusion, this paper talks about the summary of the proposal previously submitted and enumerates the changes we made from our proposal to the current working project, we showed the current implementation progress of the project, and the issues we faced during this period as well as the testing plan and lessons learned.

Chapter 8: Conclusion

1 Introduction

All things considered, MSSH strives to be a fully functional implementation of the concept of Internet of things. It strives to enhance the quality of living in housing and develop an idea of how houses in the future should look like. This project gives a glimpse of how technology can play an important part in augmenting housing and modernizing people’s experiences. From controlling doors and windows to sensing fire and gas leaks remotely, the model we based Internet of things on will be our future.

2 Lessons Learned

General Lessons Learned:

- *Manage your time effectively to avoid cramming.*
- *It is important to look for like-minded individuals to make the process less stressful.*
- *Create a fun and efficient working environment.*
- *Teamwork is important.*
- *Take the previous courses seriously to improve your technical skills and to be a reliable teammate to work with on this project at a high standard.*

Technical Lessons Learned

- *Gained more experience working with hardware.*
- *We gained a solid foundation with the concepts of Software Engineering.*
- *Improved programming skills.*

3.Recommendations for future works

Our future recommendations for readers who are interested in this field are to come up with different implementations of Smart Home concepts and innovate from existing technologies. It is also very important to find like-minded people to work through the project with, because throughout the whole process, you will face numerous problems, and the only way to overcome these challenges is to aid, encourage, and support one another.

References

[1] Agarwal, K. a. (2019). Review and Performance Analysis on Wireless Smart Home and Home Automation using IoT. *Third International conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC)*, 629-633.

[2] Bedi , G., Venayagamoorthy , G. K., Singh , R., Brooks , R. R., & Wang , K.-C. (April 2018). Review of Internet of Things (IoT) in Electric Power and Energy Systems. *IEEE Internet of Things Journal*, vol. 5, no. 2, pp. 847-870.

[3] Bhat, O., Bhat, S., & Gokhale, P. (2017). Implementation of IoT in Smart Homes. *International Journal of Advanced Research in Computer and Communication Engineering*, 149-154.

[4] Bing, K., Fu, L., Zhuo, Y., & Yanlei, L. (2011). Design of an Internet of Things-based Smart Home System. *The 2nd International Conference on Intelligent Control and Information Processing*, 921-924.

[5] Han, J.-H., Jeon, Y., & Kim, J. (2015). Security considerations for secure and trustworthy smart home system in the IoT environment. *Cyber Security System Research*, 1116-1118.

[6] Jabbar, W. A., Alsibai, M. H., Amran, N. S., & Mahayadin, S. K. (2018, June). Design and Implementation of IoT-Based. *2018 International Symposium on Networks, Computers and Communications (ISNCC)*, 1-6.

[7] Jeu Young Kim, H.-J. L.-Y.-H. (2015). Smart Home Web of Objects-based IoT Management Model and Methods for Home data mining. *Asia-Pacific Network Operations and Management Symposium (APNOMS)*, 327-331.

[8] Kian, T. K., Jabbar, W. A., Ramli, R. M., Zubir, S. N., Zamrizaman, N. S., Balfaqih, M., . . . Soltan, A. (2019). Design and Fabrication of Smart Home with Internet of Things Enabled Automation System. *IEEE Access*, 144059 - 144074.

[9] Malche, T., & Maheshwary, P. (2017). Internet of Things (IoT) for building Smart Home System. *International conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud)*, 65-70.

[10] Muhammad Asadullah, K. U. (5-7 april 2017). Smart home automation. *Smart home automation system using Bluetooth technology*, 6.

[11] P. U. Okorie, A. A. (26-28 June 2020). Design and Implementation of an Arduino Based Smart Home. *Arduino Based* , 6.

[12] Raiker, G. A., B., S. R., Loganathan, U., Agrawal, S., Thakur, A. S., K., A., . . . Thomson, M. (MARCH/APRIL 2021). Energy Disaggregation Using Energy Demand Model and IoT-Based Control. *IEEE TRANSACTIONS ON INDUSTRY APPLICATIONS*, VOL. 57, NO. 2, 1746-1754.

[13] Zhangjinglu, C. (19-20 Dec. 2015). Internet of Things. *The Internet of Things the Design and Implementation of Smart Home Control System*, 4.

[14] Smart home web of objects-based IoT management model and methods for home data mining https://ieeexplore.ieee.org/document/7275448

[15] Review and Performance Analysis on Wireless Smart Home and Home Automation using IoT https://ieeexplore.ieee.org/document/9032629

[16] Design and Fabrication of Smart Home with Internet of Things Enabled Automation System https://ieeexplore.ieee.org/document/8846205

< ----- EnD ---→