Palestine Technical University - Kadoorie
Faculty of Engineering and Technology
Department of Computer Systems Engineering

SmartHome – An IoT-based Home Automation System

By:

Abdalrahman Mohammad
Baraa Ghanim

Supervisor:

Dr. Osama Hamed

Graduation Project submitted in Partial Fulfillment of the Requirement for the Bachelor's Degree in Computer Systems Engineering.

Tulkarm, Palestine
2023-2024

إهداء

لغزة الصامدة الأبية، شرف الأمة وشعلة التحرير.
لكل بندقية حرة أبت الصمت وسط ضجيج الحرب.
لكل حر ثار دفاعاً عن فلسطين الحبيبة.
للضفة التي تقاوم رغم الصعاب.
للمدن للمخيمات للقرى التي ثارت نصرةً لغزة.
للشهداء للأسرى للجرحى لشعب فلسطين الحر.
لكل شريف وصلته صورة الحدث فلم يتخاذل وعمل بقدر استطاعته.
لليمن السعيد، للبنان الشقيق.
لفلسطين، للأقصى وطوفانه.
للمقاومة الباسلة في غزة عزنا، شرفنا وأملنا.
للسابع من أكتوبر.
لغزة...

ABSTRACT

The rapid evolution of Internet of Things (IoT) technologies has given rise to the concept of home automation, opening avenues for the creation of intelligent and efficient living spaces.

This thesis embarks on an exploration into the development and implementation of a SmartHome system, utilizing IoT principles. In a market saturated with standalone smart devices, our project takes a distinctive approach.

Rather than necessitating the complete replacement of existing devices, our goal is to enhance these devices with small, cost-effective components.

This project allows users to retain their current appliances, enabling remote control, monitoring, and seamless communication between devices. Users gain the freedom to personalize their SmartHome without the constraints of a single brand or specific conditions.

Beyond the convenience and interconnected nature of our approach, a key advantage lies in its affordability. By transforming conventional devices into smart ones, our system offers a cost-effective alternative to the more expensive smart devices common in the market.
Unlike many existing smart devices that face difficulties in seamless communication with each other, our system ensures a smooth and integrated experience, with effective communication and coordination among various devices within the SmartHome environment.

This thesis explores the details of updating existing devices, ensuring they not only adjust to users' preferences but also contribute to making smart home technology accessible to everyone.

TABLE OF CONTENT

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

In this chapter, we will discuss the outlines of our project.

1.1 Overview

For centuries, humans have sought comfort in their surroundings, and as technology has progressed, the Internet of Things (IoT) has emerged as a powerful tool to enhance this comfort.

In our project, we present an automated home system designed to Enable users with exceptional control and convenience. Through the integration of IoT technologies, individuals gain the freedom to regulate various aspects of their living space directly from their mobile devices or other compatible devices.

Our system enables users to effortlessly control, monitor, and automate devices according to their preferences, providing a customized and responsive living environment.
Moreover, the project prioritizes home safety by implementing features to prevent accidents and reduce potential dangers.
In essence, our automated home system not only reflects the evolution of technology but also addresses the fundamental human desire for a secure and comfortable living space.

1.2 Problem statement

In the era of technological advancements, the emergence of the Internet of Things (IoT) has become a prevailing trend, promising enhanced comfort for users by utilizing technological improvements.

While the market already offers smart devices showcasing IoT capabilities, significant issues persist. Current solutions often present as standalone devices, each requiring its dedicated application for control.
Interconnectivity, a core principle of IoT, remains a challenge, with many devices limited to a single brand, preventing the concept of a fully integrated smart home.
Moreover, the high costs associated with these standalone devices force users into complete replacements, changing their existing devices.

1.3 Proposed solution

Recognizing the previosu challenges, our project addresses these limitations head-on. Instead of advocating for the replacement of current devices, we propose a transformative approach.
Our system enables users to retain their existing devices and seamlessly incorporate IoT functionalities through the addition of some electronic components.
This not only ensures a comprehensive interconnection between all household devices but also provides users with wide customization options, free from any constraints, by using a web site represents the users home, the user can control and monitor all the devices in all rooms with an organized design and options to control the communication between different devices.
In addition, our solution stands out as a cost-effective alternative, offering the benefits of smart home technology without the need for a replacement change of the existing device.

1.4 Objectives

Our project aims to make a small wooden model, with a web site to control the elements in this model.

- The website will visually represent all rooms within the home, allowing independent presentation and control of each room's components.
- The website will facilitate the control of multiple microcontrollers within the same home, ensuring a centralized platform for managing diverse home devices.
- Each device in the model will be equipped with a dual control mechanism, allowing users to control them manually through the physical model and remotely through the website.
- The system will feature a timer mechanism accessible through the website, enabling users to set schedules for device operations based on their preferences.
- The website should provide the user with some statistics about home devices.
- The system will incorporate a reliable safety mechanism equipped with alarms and user notifications to ensure safety and security.
- Ensuring seamless interaction, the system will support simultaneous connections between devices and the website, allowing real-time monitoring and control.
- To enhance security, the system will implement mechanisms to restrict unauthorized access to the microcontroller interface and ensure that not every microcontroller can access the central database.
- The website will include features to restrict user accessibility, providing control measures to limit access to specific home devices based on user permissions.

1.5 Scope of Work

The project serves two categories:

- Admin - which represents the home owner.
- User - which represents the home resisdents.

CHAPTER 2

SYSTEM REQUIREMENTS

In this chapter, we will list all functional and non-functional requirements in the system

System requirements encompasses all the activities the new system must perform or support and the constraints that the new system must meet, they are separated into two categories:

2.1 Functional Requirements

They are the activities that the system must perform:

1. Create new account

- Only admin has the ability to create accounts and they can create accounts.
- They have the ability to create admin privilege accounts or user accounts.
- They can customize user access abilities based on their preferences.

2. Edit users information

- Admin can alter every user info except their name, since its a primary key.
- They can alter user access ability to give or restrict access to some other home features or rooms.

3. Access restrictions

- Admin have access to every home feature.
- Users get their access ability from the Admin.

4. Real time temperature and humidity monitory

- Every user in the house can see the temperature and humidity at real time through his web account using gauges.
- The temperature and humidity are also shown on the main LCD that is attached to the microcontroller in the front aspect of the home model.
- The temperature and humidity are calculated by the dht11 sensor attached to the microcontroller.

5. Temperature and humidity statistics

- Admin can see statistics about temperature and humidity on any selected day.
- both temperature and humidity will be shown using a chart to show the average temperature and humidity for all day hours.

6. Power consumption statistics

- Admin can see the On time period of each device in home on any selected day.
- The On time periods will be shown for all devices using chart.

7. On/Off records

- Admin can see the On/Off records for any selected device by clicking on its part on the power consumption chart.
- the records table shows the exact time the device was toggled and the one who toggled it.

8. Control home LCD

- Admin can choose what to show on the home LCD screen.
- They can choose between showing time and date or showing temperature and Humidity.

9. Control home's door

- Admin has a responsive icon that shows the state of the door, and controls it.

10. Control all devices at once

- Admin has icon of bedtime which when clicked turns off all the devices of the home.

11. Ligh Control

- Two of the three rooms on the house contains light bulbs, which can be controlled by the user inside this room's page.
- The button that controls the light shows its state at real time, and changes if any other user changed the state of the led.

12. RGB control

- Two of the three rooms on the house contains RGB lights, which can be controlled by the user inside this room's page.
- The button that controls the RGB shows its state at real time, and changes if any other user changed the state of the led.
- User can control the color of the RGB light using color input.
- User can control the intensity of the color he chose.
- User can return to the previous color the led was showing before it changed.

13. Fan control

- One room of the home contains a fan which is a DC motor.
- The button that controls the fan shows its state at real time, and changes if any other user changed the state of the fan.
- User can set the speed of the fan as they want.

14. TV control

- One room of the home contains a TV remote which is actually an IR transmitter connected with the microcontroller which sends IR signals based on the button pressed on the web page.
- The interface of the remote is very similar to actual TV remote control.
- The remote contains buttons from 0 to 9, On/Off button, volume control buttons, channel control buttons and undo button which undo the previous pressed button whether it is volume of channel or even a number.

15. Smart socket

- Admin has access to the smart socket page.
- Smart socket is a mobile socket that can be moved any where in house or even out side the house and can be controlled the web page.
- It uses another small esp01 microcontroller attached with a relay and 5V power supply. the small size of esp01 gave the mobility feature.
- It only needs WIFI connection then you can control it using the web page.
- The button that controls the smart socket shows its state at real time, and changes if any other user changed the state of smart socket.
- There is another small push button attached with the smart socket module that also toggle the socket manually.
- If someone pressed the manual pushbutton the state of the web socket button would change.

16. Timing ability

- All home devices have a time input, that the user enters the amount of time that is desired to toggle the device.
- A count down timer will appear to show how long to toggle the device.
- Also it shows what the next state of the device will be after the amount of time elapses.

17. safety mechanisms

- In case a fire happens inside the home, the MQ2 sensor that attached to the microcontroller sends a signal to it which in turn turns on a buzzer inside the house, shows a fire state on the home's LCD, shows an alert on the admin page and sends and urgent Email to all admins.
- The fire state won't stop until admin goes the to pop up alert on his page and confirms that it is ok now.

18. Connection checker

- In case some microcontroller loses connection, all the web pages attached with it would show a sticky alert that the microcontroller lost connection, with identifying the id of this microcontroller.

19. LED indication

- The home model contains a small green LED that turns on when the microcontroller is connected with WIFI, and turns off when the connection is lost.

20. Central LCD

- The home model contains a LCD screen that shows Time or date depending on admin's preference.
- It shows the important safety alerts.

21. Ordinary switches

- All home LEDs and RGBs have physical ordinary switches to control them beside the website.
- When a switch toggles the effect goes simultaneously to the website to reflect the new state of the device.

2.2 Non-Functional Requirements

These are system characteristics other than the activities the system must perform or support.

1. Responsive user interfaces

- All the web pages are responsive and suitable for almost all devices, which gives users the freedom to control the home from any where using any device.

2. User friendsly interfaces

- This is accomplished in our website by using recognizable real-life symbols and icons.
- As in door controller button, lights switch buttons, TV remote control, temperature statistics, power usage.
- The usage of gauges and charts makes it easy to users to comprehend the data quickly.

3. Reliability

- The usage of website and the ordinary manual usage for the most of home devices, gives the whole system the reliability and it can works under unexpected circumstances as losing WIFI connection through home.

4. Security

We assured to apply security with both sides:

- **Website Security**
  - It is accomplished using the login mechanism.
  - All passwords are hashed before being stored in the data base.
- **Microcontroller security**
  - Each microcontroller has and id and password that it uses to send and get data from system.
  - All microcontrollers passwords are hashed in the data base.

5. Recovery

- In case the electricity blinks or the microcontroller loses power for some reason, when it comes back again everything will return to its last state.

5. Affordability

- Our system uses much cheaper components than any other smart device in the stores.
- Our system enables users to enhance their existing devices without the need for purchasing new ones.

6. Customized Interconnectivity

- Our system ensures seamless interconnection among all home components and can be altered to match exactly what a user needs.

7. portability

- By designing the system in a way that it can use multiple microcontrollers, The user can move some components to anywhere even outside the home as in the smart socket.

CHAPTER 3

HARDWARE COMPONENTS

In this chapter, we will list all hardware components used in this project

Note: Some components used in this project are employed just for modeling purposes, and they may not meet the standards of high quality or reliability required for real-world home applications, but they convey the same picture and can be used exactly the same way.

1. Microcontrollers

Two types of microcontrollers are used in this project:

- **ESP32-wroom-32**

The ESP32, a versatile and powerful microcontroller, has emerged as a pivotal component in modern embedded systems. Developed by Espressif Systems, the ESP32 offers a wealth of features, including dual-core processing, built-in Wi-Fi and Bluetooth capabilities, and a variety of GPIO pins for interfacing with external devices. Its low power consumption and cost-effectiveness make it an ideal choice for a wide range of applications, from Internet of Things (IoT) projects to home automation systems. The ESP32's open-source nature and strong community support contribute to its popularity, promoting innovation and ease of integration in diverse electronic projects.

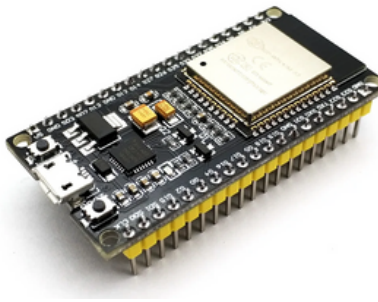

Figure 3.1: ESP32

This microcontroller is used in our project as the controller of everything except the smart socket.

- **ESP01**

The ESP-01, a compact and cost-effective module developed by Espressif Systems, stands as a great choice for IoT and embedded systems. Despite its small size, the ESP-01 delivers robust Wi-Fi capabilities, making it well-suited for applications demanding wireless connectivity. Though it features a single GPIO pin, its simplicity can be an advantage for specific projects where space and resources are constrained.

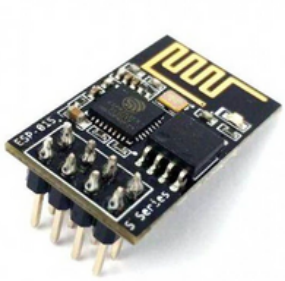

Figure 3.2: ESP01

This microcontroller serves as the brain of our smart socket in the project. We opted for it due to its compact size, enhancing the device's portability.

2. ESP01 relay board

This compact board features a socket for the ESP-01 and is equipped with a relay module. Its primary function is to cut or connect the live phase of the smart socket, effectively controlling its power state—enabling users to turn it on or off as needed.

Figure 3.3: ESP01 board

3. Relay module

A relay module is an essential electronic component that acts as an electrically operated switch. It consists of a coil and one or more switches, allowing it to control high-power devices using a low-power signal.

In our project, the relay module plays a crucial role for the AC lights. It enables the interruption or establishment of the live phase, effectively managing the power flow to turn the light on or off.

The compact and versatile nature of relay modules makes them invaluable in automation and control applications, providing a reliable means to control electrical devices remotely or in response to specific conditions.

Figure 3.4: Relay module

4. Socket & Plug

Ordinary socket and plug.

They're used to make the model of the smart socket.

Figure 3.5: Plug

Figure 3.6: Socket

5. Switches

Ordinary switches to turn on/off the lights and RGBs.

Figure 3.7: Switch

6. 5V power supplies

A 5V power supply is a fundamental component in electronic systems, delivering a stable and regulated voltage to support the operation of various devices. Its role is crucial in maintaining consistent and reliable power for microcontrollers, sensors, relays, RGBs and other components.
in our project they're used to operate both ESP32 and ESP01 microcontrollers, and for all other components relays, motor, stepper motor, lcd, RGBs, sensors.

Figure 3.8: Power supply

7. Bulbs

Ordinary AC bulbs, to show how we can control AC devices using our system.

Figure 3.9: Bulb

8. RGB strips

An RGB strip, equipped with red, green, and blue LEDs, serves as a vibrant and customizable lighting. This flexible strip allows dynamic color changes, providing a visually pleasing and customizable atmosphere.

Figure 3.10: RGB

9. DC motor

The DC motor, a staple in electromechanical systems, converts electrical energy into mechanical motion. Its versatility enables precise control of movement, making it an essential component in various applications, from robotics to industrial machinery.
in our project it is employed to simulate a fan and how we can control it through our smart home system.

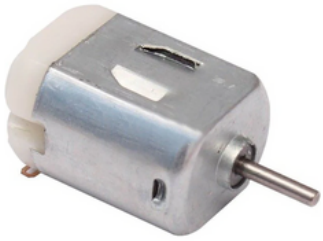

Figure 3.11: DC motor

10. Stepper motor

A stepper motor is a specialized electromechanical device known for its ability to convert digital signals into precise and incremental rotational motion. It operates in distinct steps, offering accurate control and positioning. Commonly used in various applications, such as robotics and automation, the stepper motor provides reliable and controlled movement in response to digital inputs.
in our project it is used to represent the door shutter, but in real world it is replaced by the actual gate motor.

Figure 3.12: Stepper motor

11. Lcd 1602

The LCD 1602 is a compact and versatile alphanumeric display module with 16 columns and 2 rows. Widely used in electronics projects, it features a backlight for clear visibility in different lighting conditions. This simple yet effective module serves as a user-friendly interface, displaying essential information in a variety of applications.
in our project it is used to show the temperature and humidity, date and time or any safety alerts inside the home model.

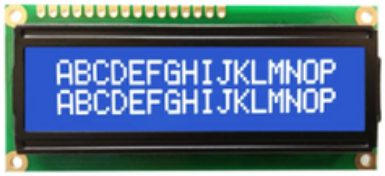

Figure 3.13: LCD

12. Active buzzer

The active buzzer is a small, straightforward audio device used in electronic circuits. It produces a continuous sound when voltage is applied, making it easy to use for audible alerts or signals in various applications.
in our project it is used as an alert to a safety incident.

Figure 3.14: Buzzer

13. LED

A small green LED is used in our model as an indicator to WIFI connection, in a way that it turns on when the connection is stable, and turn off when the microcontroller loses connection.

Figure 3.15: Led

14. IR reciever

The IR (Infrared) receiver is a compact device that detects and interprets infrared signals, commonly used in remote-controlled electronic devices. It translates incoming infrared signals into electrical signals, enabling devices to respond to user inputs wirelessly.
In our project we used it to extract the IR codes of the TV remote we have so that we can upload them to the microcontroller to transmit them later.

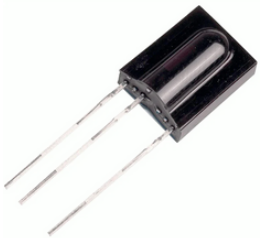

Figure 3.16: IR reciever

15. IR transmitter

The IR (Infrared) transmitter is a compact device that emits infrared signals. It is commonly used in remote control systems, sending coded signals to IR receivers in electronic devices. The IR transmitter facilitates wireless communication and is employed in various applications, providing a means to remotely control or communicate with compatible devices.
in our project we used it to transmit the previous stored IR signals to control the TV as its original remote does but using web interface.

Figure 3.17: IR transmitter

16. DHT11 sensor

The DHT11 sensor is a small and affordable device that measures temperature and humidity. With a straightforward digital output, it's commonly used to provide real-time environmental data in electronic projects.
in our project it is used to read the temperature and humidity, and show them on the LCD screen and sending them to the data base.

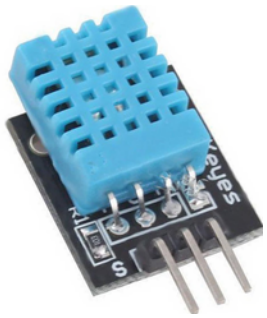

Figure 3.18: DHT 11 sensor

17. MQ2 sensor

The MQ2 sensor is a compact gas sensor widely used for detecting various gases in the environment. Its simple interface and analog output make it suitable for electronic projects where gas detection is essential. The MQ2 sensor is particularly known for its ability to sense gases such as methane, propane, carbon monoxide, and smoke, making it a versatile choice for applications related to air quality monitoring and safety.
in our project it is used for fire detection.

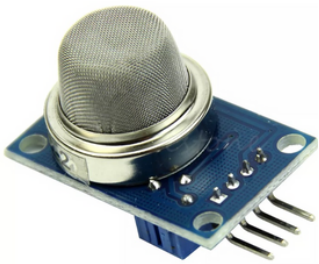

Figure 3.19: MQ2 sensor

CHAPTER 4

METHODOLOGY AND TECHNOLOGIES

This chapter discusses the Methodology, techniques, and technologies we used to build this project.

4.1 Overview of the used Technologies

the system we made is divided into two main components: microcontroller technologies and web technologies.

- **Microcontroller Technologies**

  Integrated Development Environment

  - Visual Studio Code

  IDE extenions

  - PlatformIO

  Programming language

  - c++
  - c

  Libraries

  - WiFi.h
  - DHT.h
  - Stepper.h
  - LiquidCrystal.h
  - HTTPClient.h
  - Arduino_JSON.h
  - IRremote.h
  - LiteLED.h

- **Web Technologies**

  Integrated Development Environment

  - Visual Studio Code

  Front-End Technologies

  - HTML
  - CSS
  - JavaScript
  - jQuery
  - Bootstrap

Front-End tools and libraries

- sweetalert2
- justgage
- chart.js

Back-End Technologies

- php

Database

- MySQL database

Back-End tools and libraries

- PHPMailer

4.2 Integrated Development Environment

4.2.1 Visual Studio Code

Visual Studio Code (VS Code) is a free, lightweight source-code editor developed by Microsoft. Known for its versatility, it is widely used for programming in various languages and platforms, including web development and microcontrollers.

The editor boasts a clean interface, efficient code editing features, and support for version control. With a robust extension system, developers can customize it to their needs, making VS Code a popular choice across different domains and skill levels. Its cross-platform compatibility further enhances its appeal, supporting Windows, macOS, and Linux.

Figure 4.1: VS code logo

4.3 IDE extenions

4.3.1 PlatformIO

PlatformIO (PIO) is an open-source ecosystem that streamlines Internet of Things (IoT) and embedded systems development. As an extension for popular IDEs like Visual Studio Code and Atom, PlatformIO simplifies coding, building, and uploading firmware for a range of microcontroller platforms. Widely adopted for its versatility, it serves developers working on diverse projects, from web development to IoT applications, providing a unified platform across various microcontrollers.[4.1][4.2]

Figure 4.2: PlatformIO logo

4.4 Microcontroller programming languages

4.4.1 C / C++

C/C++ serves as the primary programming language for Arduino microcontrollers, providing a robust foundation for embedded systems development. Programmers use C/C++ to write code that dictates the behavior of Arduino boards, from reading sensor inputs to controlling outputs.
The Arduino IDE compiles the code into binary files, which are then uploaded to the microcontroller, defining its functionality. This streamlined process, coupled with the user-friendly nature of Arduino's simplified C++ variant, enables developers to create diverse projects, ranging from basic electronics to advanced IoT applications.

Figure 4.3: C / C++ logo

4.5 Microcontroller used libraries

4.5.1 Wifi.h

An Arduino library, used to connect the microcontrollers to a WIFI network.

4.5.2 DHT.h

An Arduino library, used to convert dht11 sensor readings into readable form.

4.5.3 Stepper.h

An Arduino library, used to control the movement of the stepper motor.

4.5.4 LiquidCrystal.h

An Arduino library, used to control the LCD screen.

4.5.5 HTTPClient.h

An Arduino library, used to send http requests to some page.

4.5.6 Arduino_JSON.h

An Arduino library, used to convert Strings into JSON format and vice versa.

4.5.7 IRremote.h

An Arduino library, used to control IR transmitters and senders.

4.5.8 LiteLED.h

An Arduino library, used to control used to control RGBs.

4.6 Front-End Technologies

4.6.1 HTML, CSS, Javascript

HTML, CSS, and JavaScript are foundational languages in web development.
HTML (HyperText Markup Language) serves as the structural backbone of a webpage, defining elements such as headings and paragraphs.
CSS (Cascading Style Sheets) complements HTML by styling and formatting its elements, enhancing the visual presentation of the page.
JavaScript, a dynamic scripting language, adds interactivity and functionality to webpages. It enables developers to create dynamic content, handle user input, and interact with the browser, contributing to a seamless and engaging user experience.
Together, these three languages play essential roles in shaping the design and functionality of any website.

Figure 4.5: HTML, CSS, JS logo

4.6.2 Bootstrap

Bootstrap is a widely-used front-end framework known for simplifying web development. Utilized to create responsive pages, it provides pre-designed components, streamlining the process of building visually consistent and efficient web applications.

Figure 4.6: Bootstrap logo

4.6.3 JQuary

jQuery, a lightweight JavaScript library, enhances web development by simplifying tasks like document traversal and event handling.
Widely used for creating interactive web pages, jQuery's concise syntax and robust functionality make it a preferred choice for developers seeking efficient and streamlined web development.

Figure 4.7: jQuery logo

4.7 Front-End tools and libraries

4.7.1 Sweetalert2

SweetAlert2 is a JavaScript library for stylish and responsive pop-up alerts, providing a simple API for customization. Widely used for its ease and feature-rich design, it enhances the presentation of alerts in web applications.[4.3]
we used it to show alerts when a fire happens, and for pop-up windows.

Figure 4.8: Sweetalert logo

4.7.2 justgage

JustGage is a JavaScript library for creating customizable and dynamic gauges and meters. With a simple API, it's easy to integrate into web applications, making it a popular choice for visualizing data effectively, particularly in dashboard and monitoring contexts.[4.4]
we used it to show temperature and humidity in gauge forms.

Figure 4.9: Justgage logo

4.7.3 chart.js

Chart.js is a user-friendly JavaScript library for creating dynamic and visually attractive charts on web pages. With a simple API, it supports various chart types, making it a popular choice for developers to integrate responsive and engaging data visualizations seamlessly into their applications. [4.5]
we used it to show power usage and temperature records.

Figure 4.10: Chart.js logo

4.8 Back-End Technologies

4.8.1 php

PHP is a server-side scripting language crucial for web development. Its open-source nature allows seamless integration with HTML to create dynamic web pages. Known for versatility and community support, PHP enables developers to handle server-side scripts, manage databases, and generate dynamic content, making it a foundational choice for diverse web applications.

Figure 4.11: PHP logo

4.9 Database

4.9.1 MySQL

MySQL is a prominent open-source relational database management system vital for web development. It efficiently organizes and retrieves data, providing reliability, speed, and compatibility with multiple programming languages. Widely used for managing databases in dynamic websites and web applications, MySQL is a fundamental element in the web development landscape.

Figure 4.12: MySQL logo

4.10 Back-End tools and libraries

4.10.1 PHPMailer

PHPMailer is a popular open-source library that simplifies email integration in PHP. It streamlines the creation and sending of emails, supporting multiple protocols like SMTP. With its secure and user-friendly design, PHPMailer is a preferred choice for developers looking to seamlessly incorporate email functionality into their PHP-based projects. [4.6]

we used it to send alert email to Admins when a fire happens.

Figure 4.13: PHPMailer logo

4.11 Methodology

Our system is structured into three essential components: microcontrollers, web system, and a database.
These components collaboratively communicate to deliver the desired functionalities.

- **The microcontrollers**
  they handle the physical interaction with devices, by sending and receiving data. two types of microcontrollers:
  - **ESP01**: used only in smart socket.
  - **ESP32**: used for all other functionalities.

- **The web system**

  it serves as the user interface, allowing seamless control and monitoring from various devices.

- **The database**

  The database functions as both a repository and an intermediary stage between the microcontroller and the web system[4.8]. It stores and retrieves information, ensuring synchronization across the entire system.

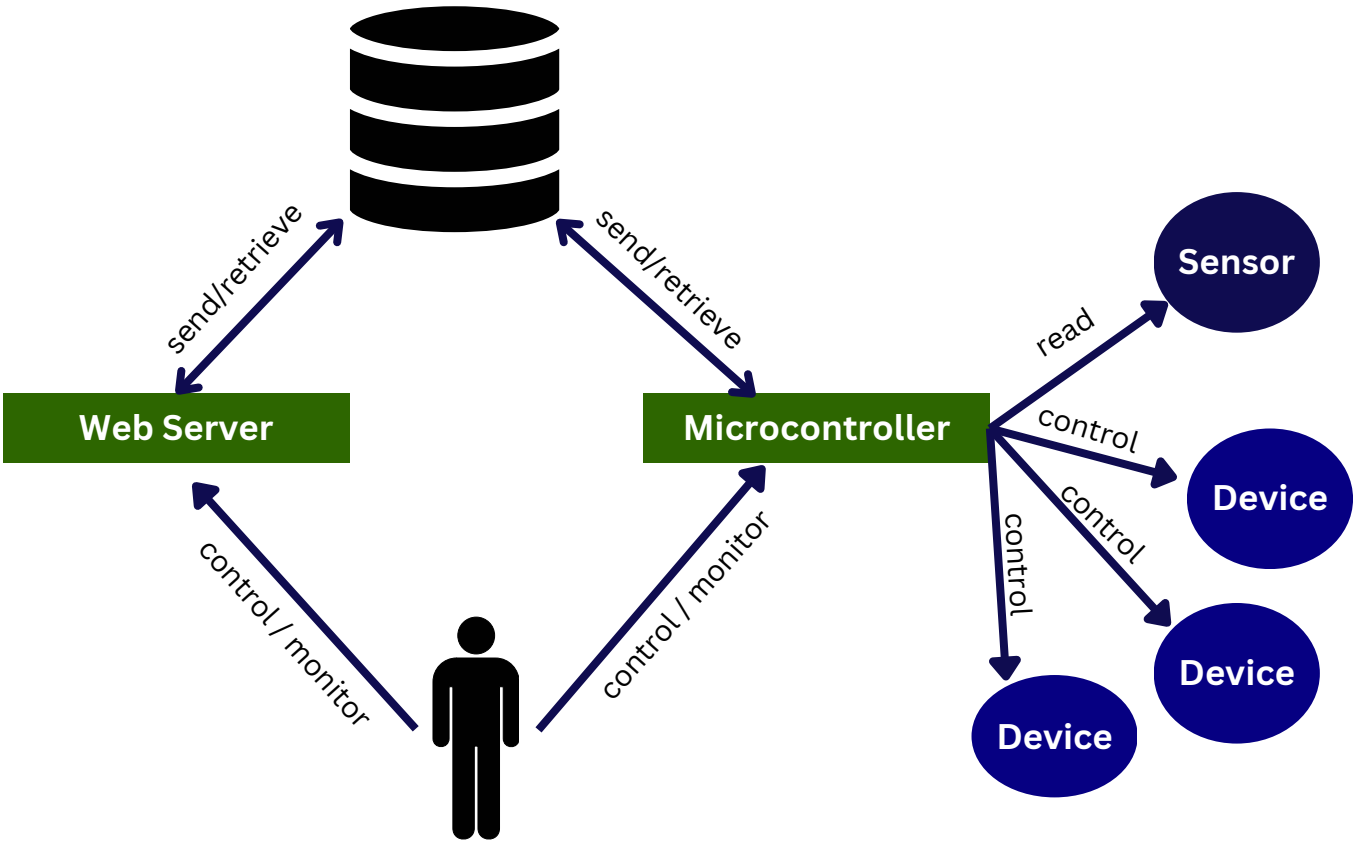

Figure 4.14: A diagram describes how system components works

4.11.1 How microcontrollers control devices

- **Manually**

  1. The user can interact with the switches connected to the microcontroller, and so it can accomplish the functionality attached to that switch.
  2. When a change occurs on the microcontroller, it promptly sends an HTTP request to the web server, triggering the update of the relevant table associated with these data.

- **Using website**

  1. The user interacts with the website, initiating the transmission of this interaction to the web server.
  2. Subsequently, the web server updates the relevant data in the corresponding table within the database.
  3. Simultaneously, the microcontroller sends HTTP requests every 500 milliseconds to retrieve data from the database. Upon detecting a change, it takes control of the device associated with that specific alteration.

4.11.2 How microcontrollers show readings

- **To the model**

  1. if some sensor detects something new, the microcontroller sends this data to the other related components. for example:
     a. dht11 sensor sends temperature and humidity to LCD screen.
     b. MQ2 sensor sends alarm to the buzzer and LCD screen when a fire happens.

- **To website**

  1. if some sensor detects something new, the microcontroller sends an HTTP request to the web server, triggering the update of the relevant table associated with these data.
  2. Simultaneously, the website utilizes JavaScript to dynamically update related data on the webpages. It reads the data from the database every 500 milliseconds to reflect any new information.

4.11.3 How microcontroller's code is written

We opted to implement the Command Pattern for the primary devices in the home, situated in most rooms. This decision facilitates ease of management and enables the execution of microcommands to meet users' specific needs (as we used it in turn off all devices as an example). The Command Pattern was chosen as it offers a structured approach to command execution, enhancing the flexibility and customization of device control within the smart home system.[4.7]

In our implementation ROOM works as the Invoker, and we have RGB, led, Fan and TV as receivers.

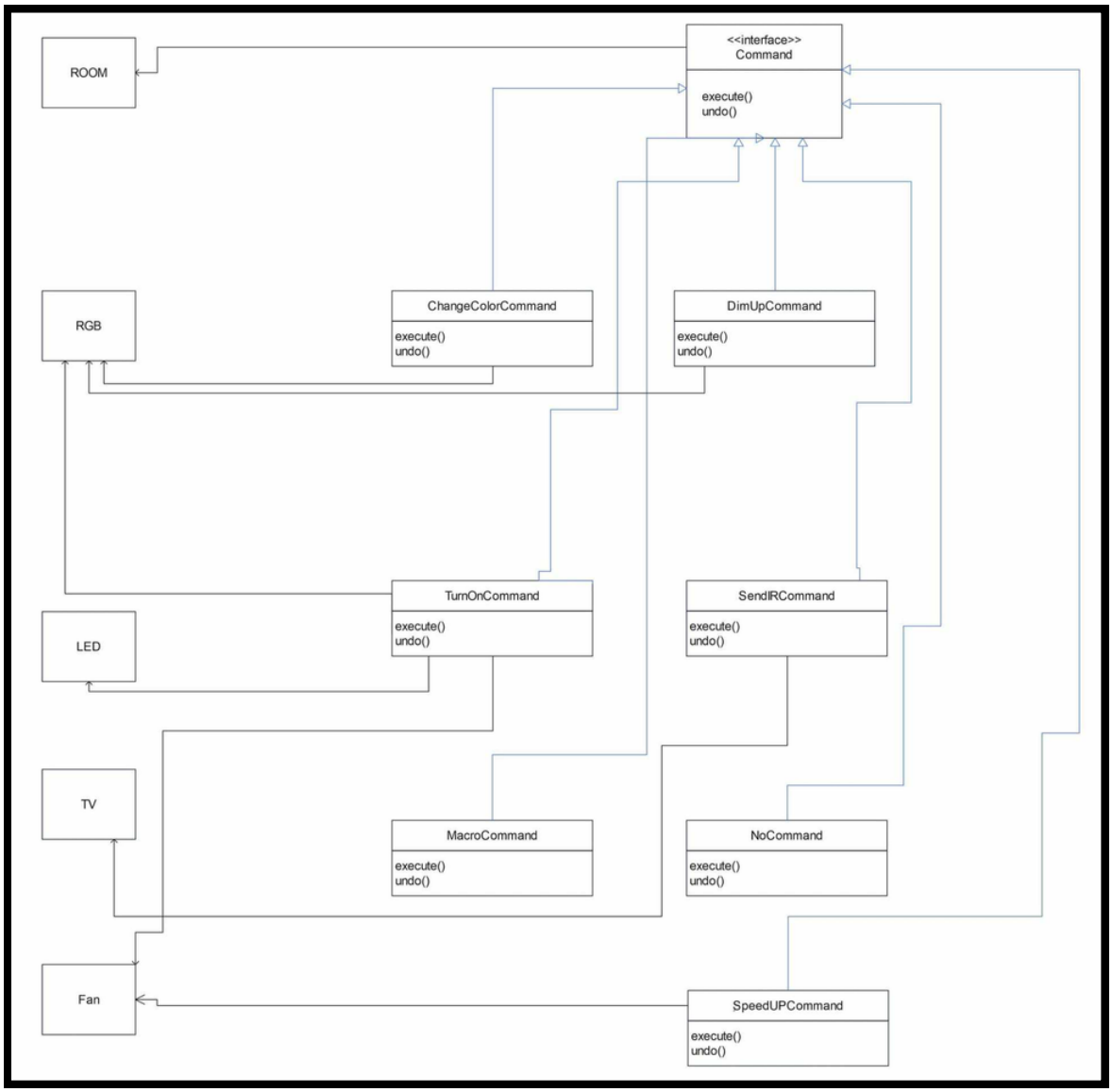

Figure 4.15: The command pattern

If some room doesn't contain all of these devices, this is not a problem since in the constructor we assign all devices to No(device) class which does nothing.

We designed the No(device) classes as singleton classes to ensure uniformity across all rooms. By implementing them as singletons, we guarantee that regardless of the number of rooms, they all reference and utilize the same object, promoting consistency and efficient resource utilization.

After the user makes an instance of the class ROOM he can set whatever devices they want.

```
ROOM()
{
    NoCommand *noCmd = &NoCommand::getInstance();
    ledoncommand = noCmd;
    changecolorcommand = noCmd;
    dimupcommand = noCmd;
    presstvbuttoncommand = noCmd;
    speedupcommand = noCmd;
    led = &NoLed::getInstance();
    rgb = &NoRgb::getInstance();
    tv = &NoTV::getInstance();
    fan = &NoFan::getInstance();
    ledbuttonclicked = false;
    rgbbuttonclicked = false;
    ledbuttonclickedbytimer = false;
    rgbbuttonclickedbytimer = false;
    fantoggledbytimer = false;
    startException = true;
}
```

Figure 4.16: A picture shows the constructure of ROOM class

CHAPTER 5

SOFTWARE ANALYSIS & DESIGN

This chapter shows the UML diagrams for project

5.1 System's Architecture

This section will present a high-level overview of the anticipated system architectures.

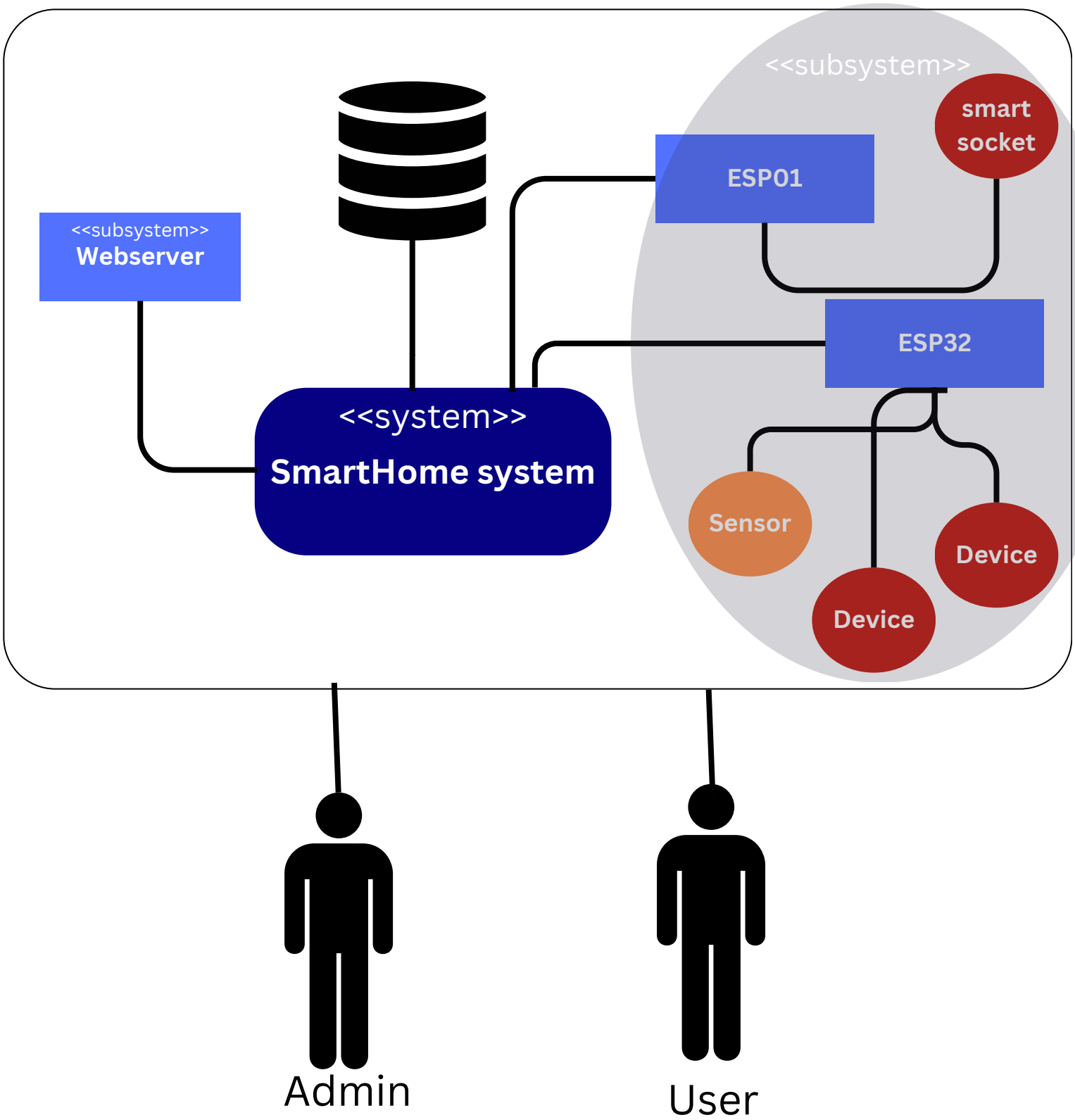

Figure 5.1: Overview of the system's architecture

5.2 Use Case Diagrams

As mentioned earlier, our system caters to two distinct categories: Admin and User. Accordingly, we will create separate use case diagrams for each category.

note: user access levels are determined by the choices made by the admin, and the ultimate access permissions granted will be shown in the diagrams.

5.2.1 use case diagram

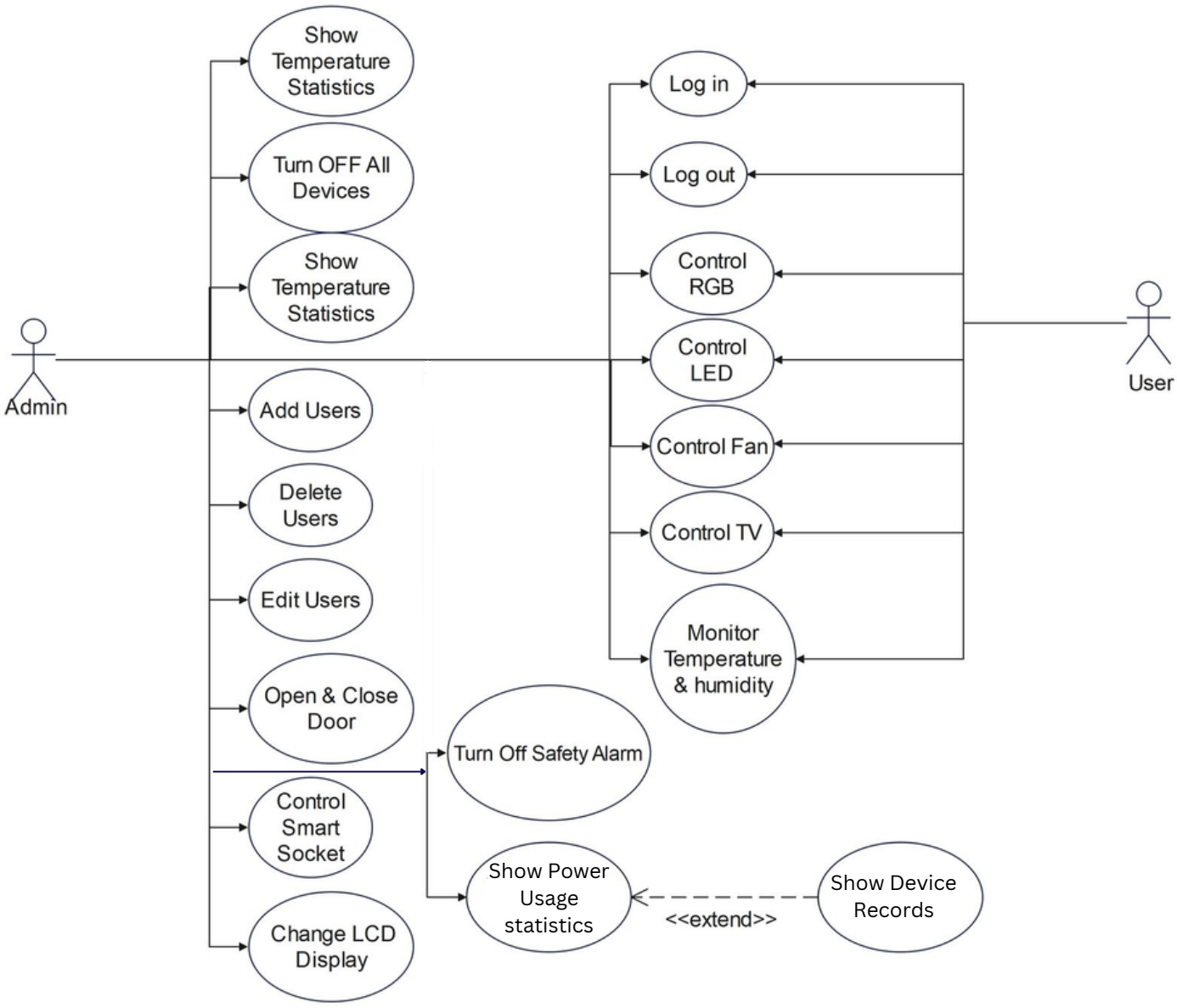

Figure 5.3: Use Case diagram

5.3 Use Case Documentation

5.3.1 Use Case Documentation for opening room page on the website

Use case name:	enter room's page	
Scenario:	some user enter a room page	
Triggering event	some user goes to a url of some room page	
Brief description:	if an Admin or normal user tries to go to room's page or smart socket page	
Actors	Admin and normal user	
Related use cases	login use case	
Stockholders	Admin	
Preconditions:	the user/admin has logged in to his account	
Postcondition	the user/admin will be directed to the room's page if he was authorized. else will be kicked out to login page with error.	
Flow of activates	User	Webserver
	1. Access to smart home site. 2. goes to some page's url. 3. user enters/ leaves room's page	1.1 show smart home page 2.1 if there is no session go to login page. 2.2 validate user's role. 2.3 if user's role is admin or contains the number of this room display the page, else kick them out

Figure 5.4: Use Documentation for entering room's page

5.3.2 Use Case Documentation for microcontroller send/read from database

Use case name:	microcontroller read/send data	
Scenario:	some microcontroller sends/ want to retriever some data	
Triggering event	a microcontroller sends http request to send/get data	
Brief description:	one of the microcontrollers want to read data from the data base, or it wants to update some data there	
Actors	ESP32 and ESP01	
Related use cases	–	
Stockholders	Admin	
Preconditions:	The Webserver must be operating	
Postcondition	the request of that microcontroler will be accpeted and it will retreive that data it wants, or updata the desired table. or the request gets refuesed.	
Flow of activates	microcontroller	Webserver
	1. Initiates an HTTP request with specified parameters. 2. adds authentication id and password to the HTTP parameter. 3. send the request to the server 4. receive a signal about the request state, and get data if it was a retrieving request.	3.1 receive the https request and validates the id and password to check if it is authorized. 3.2 if so access the data base and apply the request(send data or get data) 3.3 retrieve signal that operation is done/refused. and retrieve data if there is so.

Figure 5.5: Use Documentation microcontroller interaction with the data base

5.4 Domain model class diagram

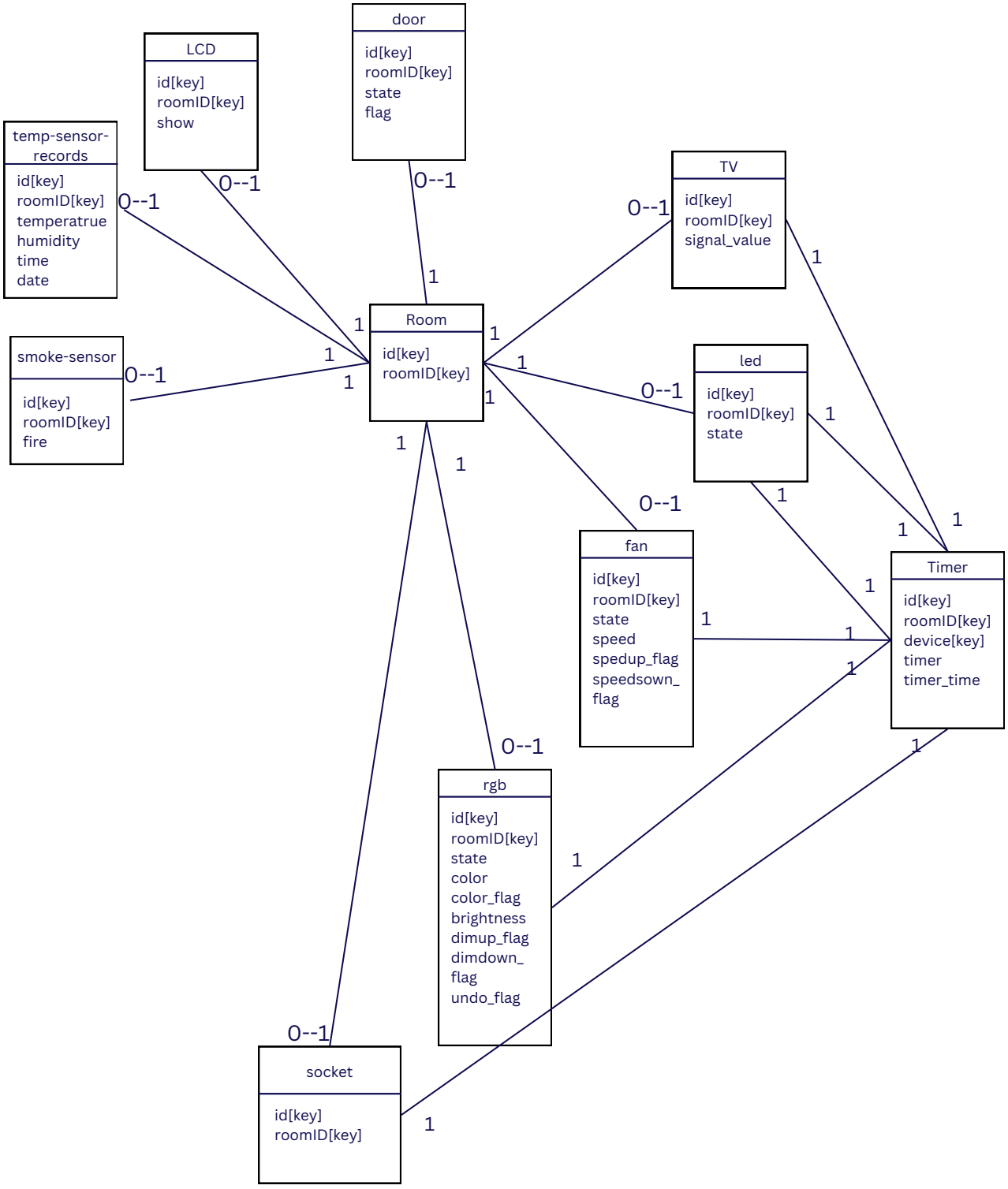

Figure 5.6: Domain model class diagram for the system

5.5 sequence diagram

5.5.1 sequence diagram for toggling a switch on the model

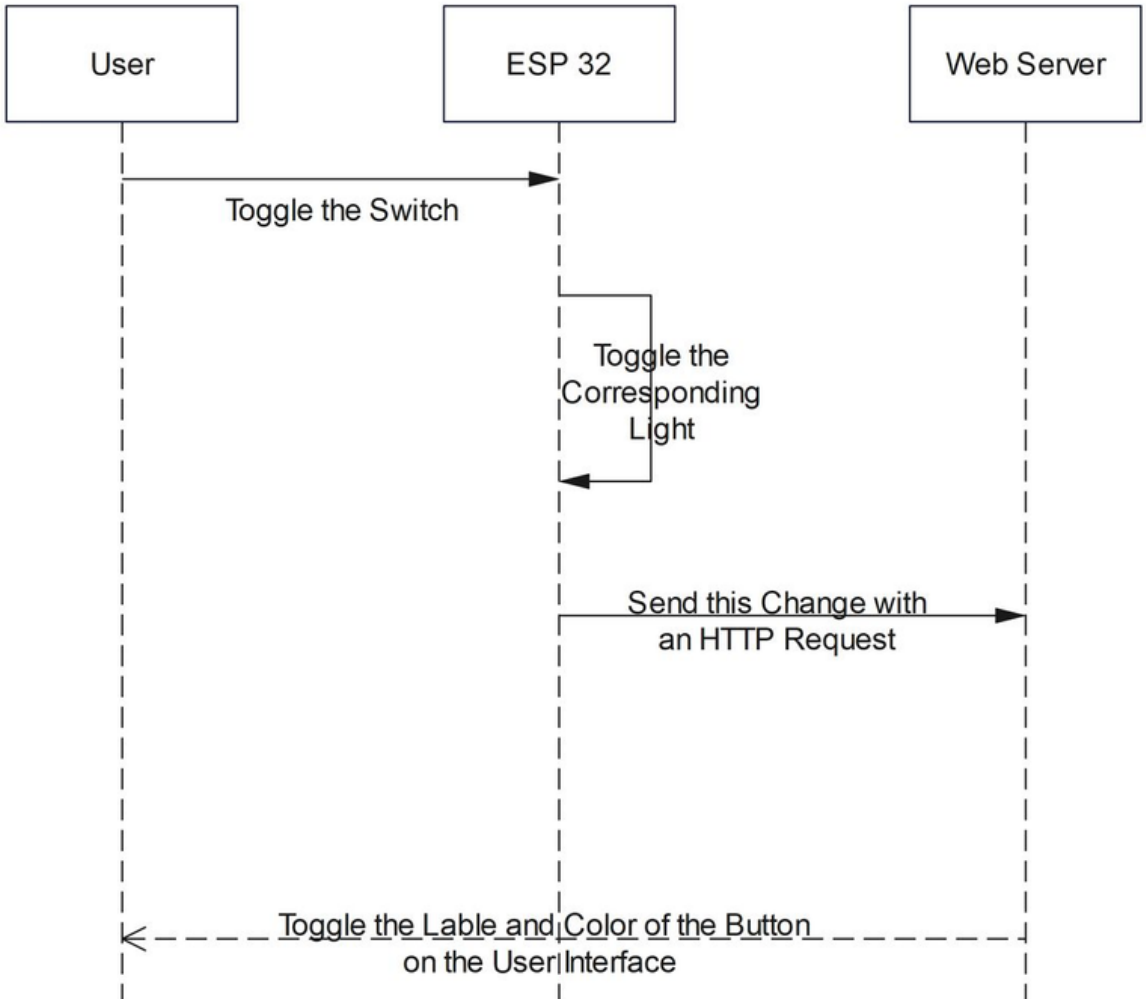

Figure 5.7: Sequence diagram for toggling a switch on the model

5.6 Activity diagram

5.6.1 Activity diagram for speeding up the fan

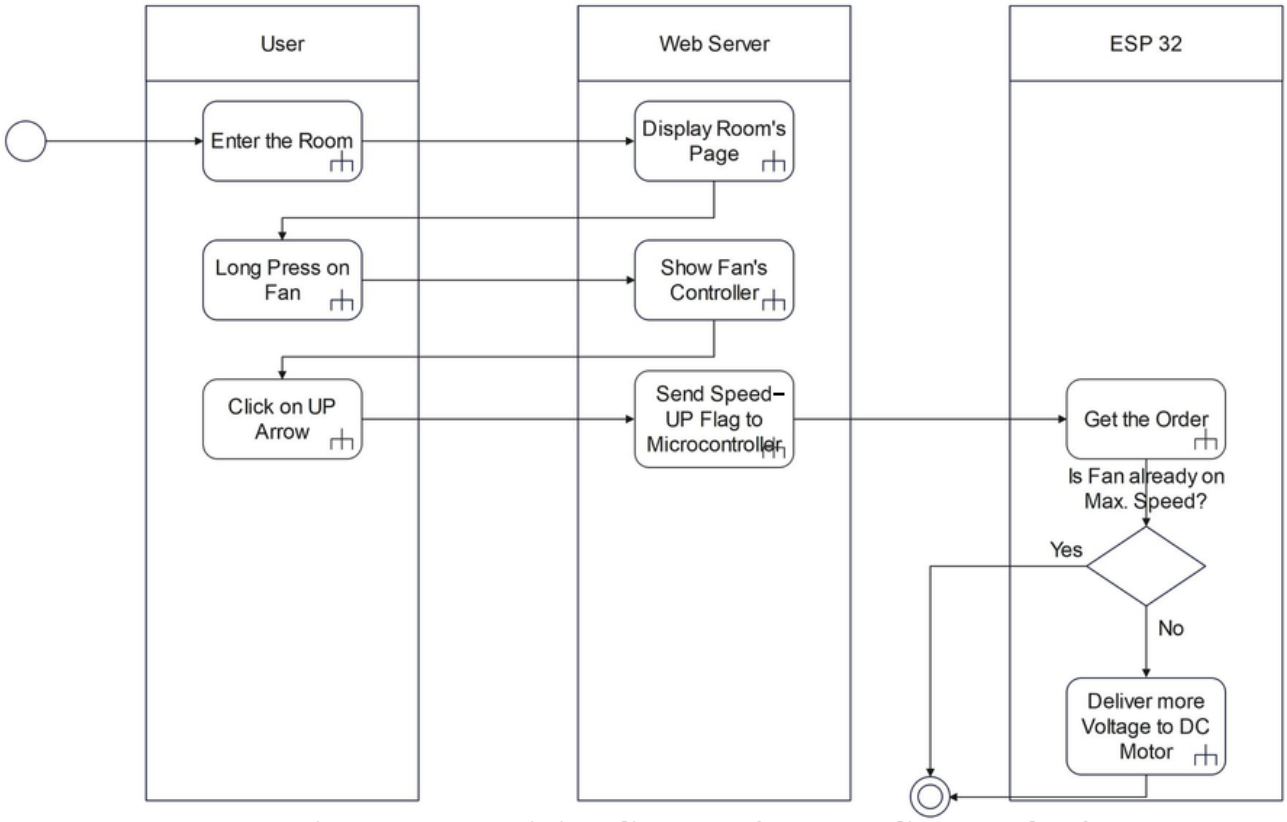

Figure 5.8: Activity diagram for speeding up the fan

5.6.2 Activity diagram for changing the color of an RGB light

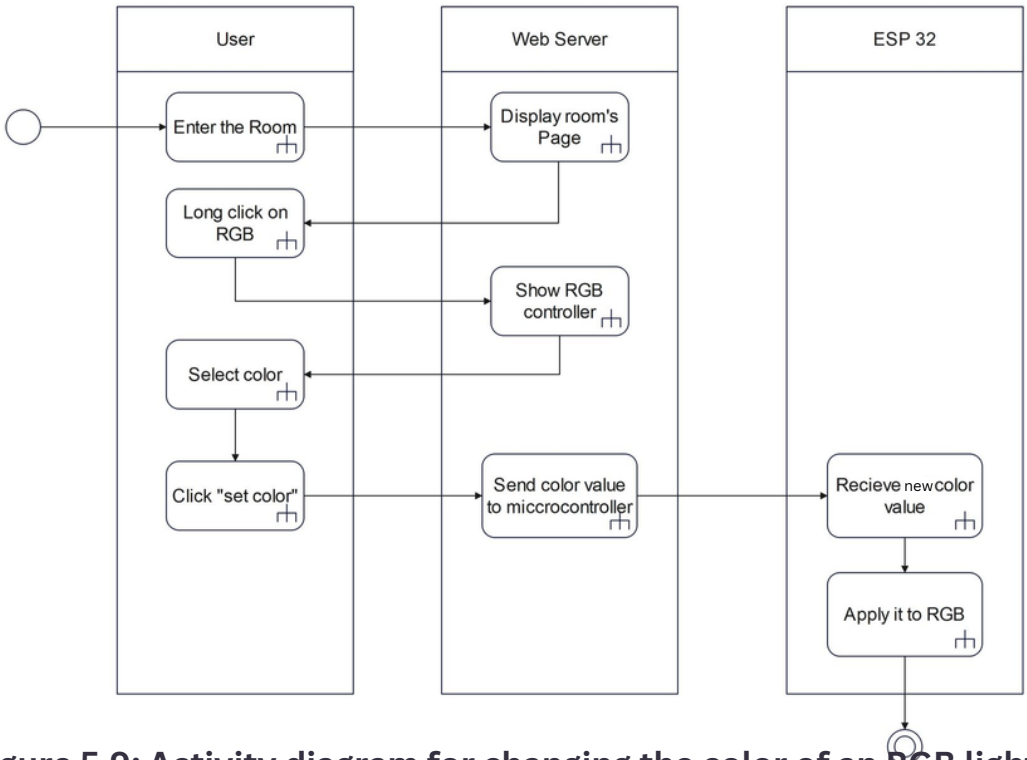

Figure 5.9: Activity diagram for changing the color of an RGB light

5.6.3 Activity diagram for setting a timer for smart socket

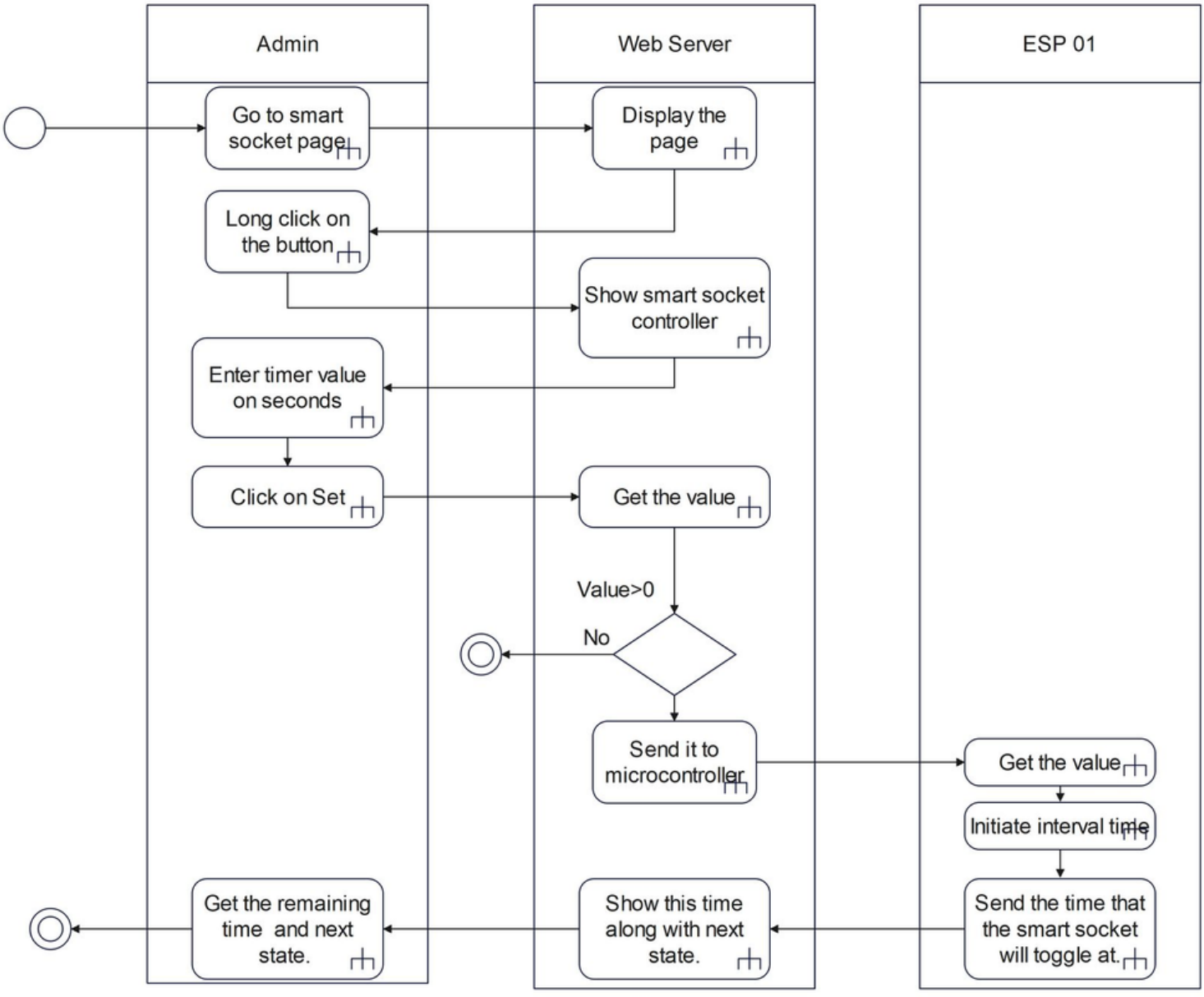

Figure 5.10: Activity diagram for setting a timer for smart socket

5.7 state machine diagram

5.7.1 state machine diagram for door entity

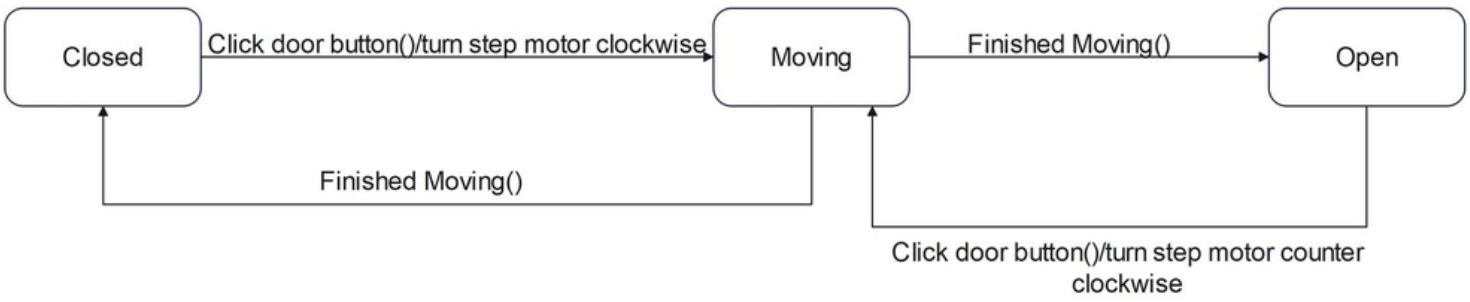

Figure 5.11: State machine diagram for door entity

5.7.2 state machine diagram for fire entity

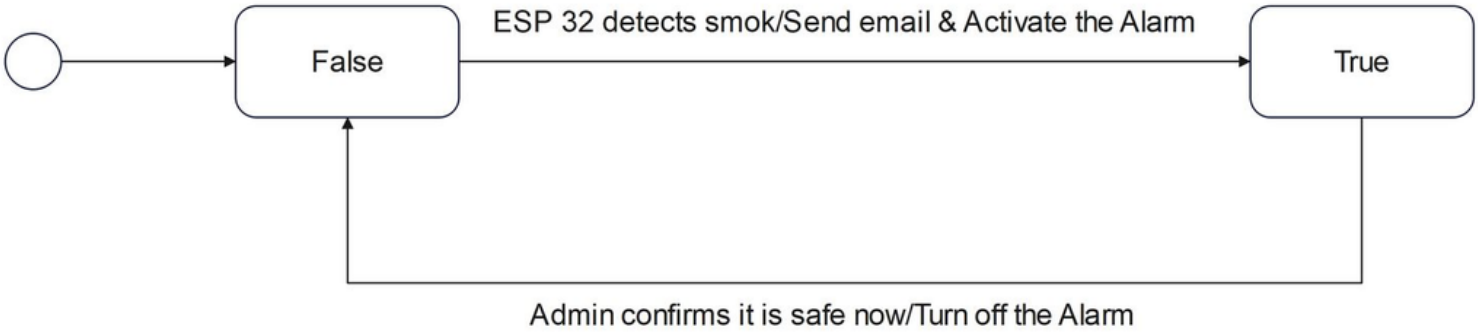

Figure 5.12: State machine diagram for fire entity

5.8 Network diagram

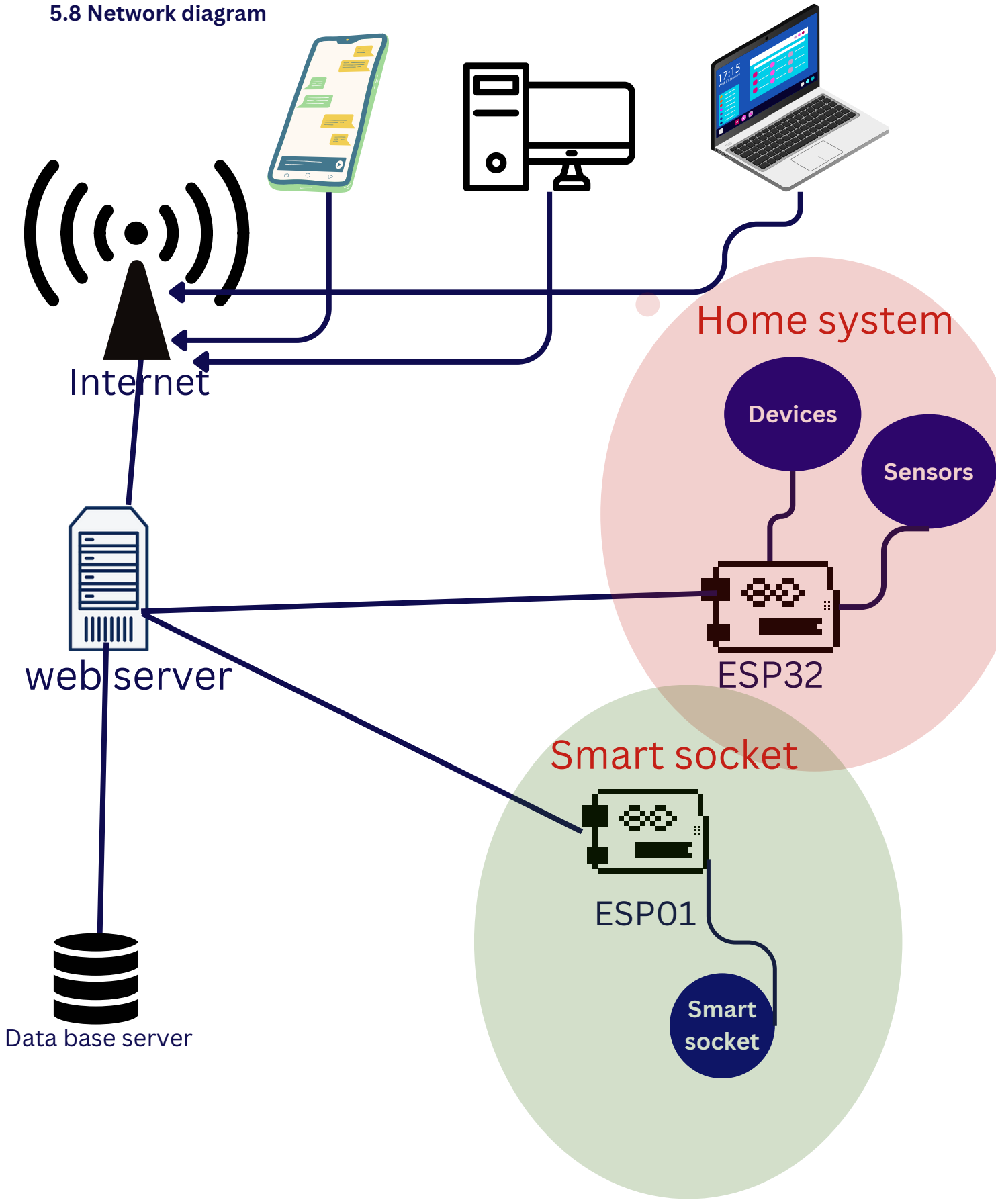

Figure 5.13: Network diagram for the Smart home system

CHAPTER 6

USER INTERFACES

This chapter shows the system interfaces for all users

6.1 Login page

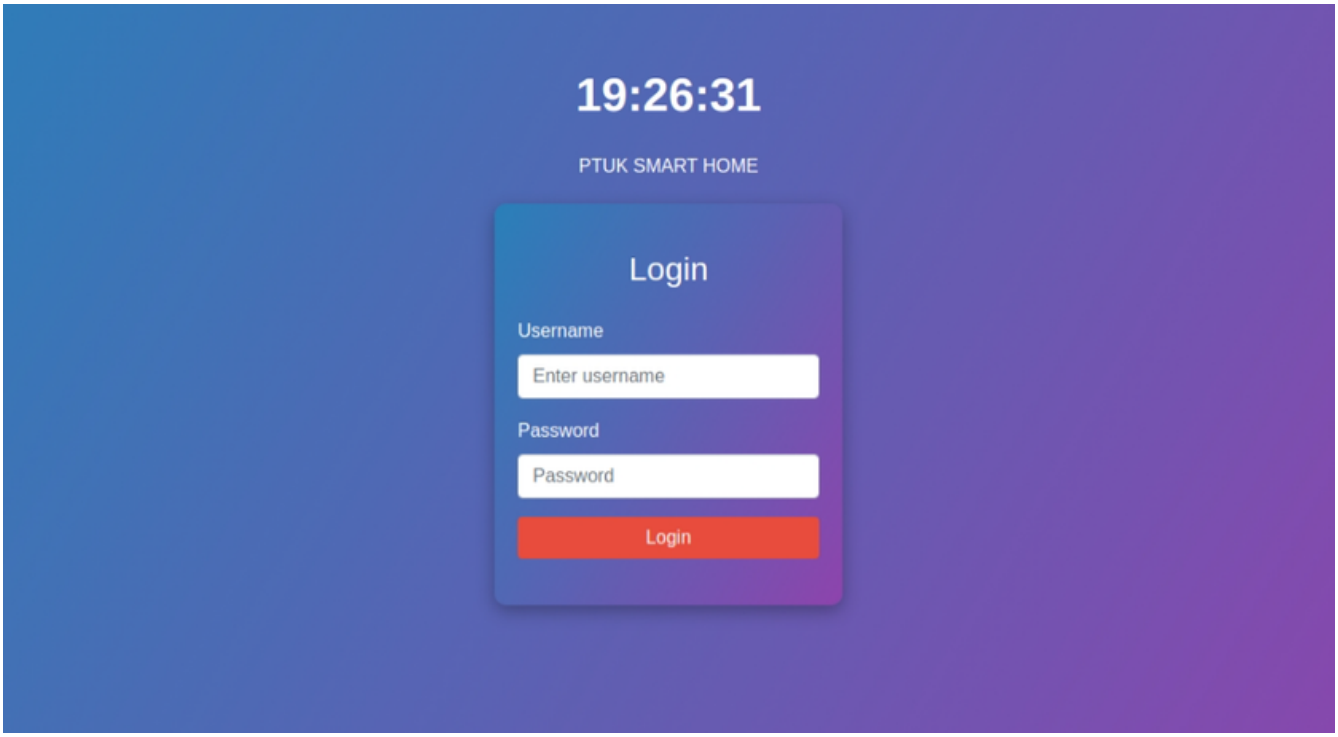

Figure 6.1: Login page

6.2 User pages

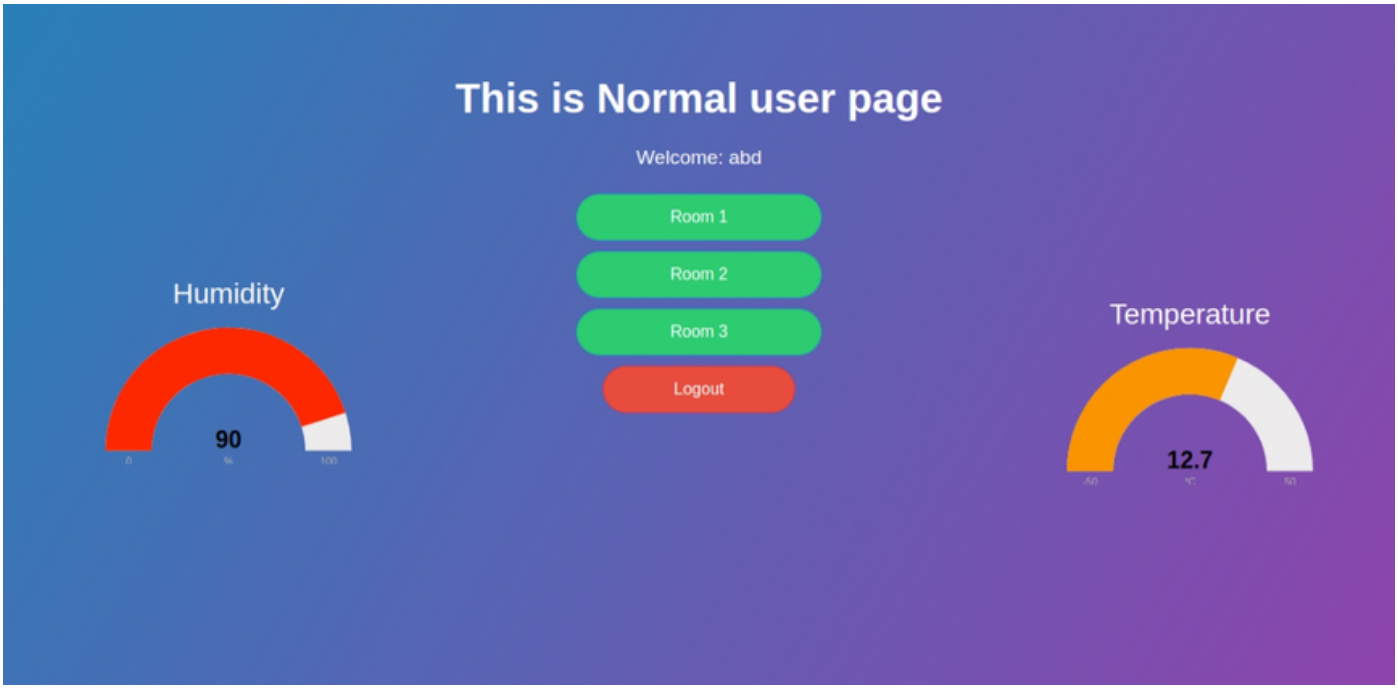

Figure 6.2: User's main page

6.3 Room's pages

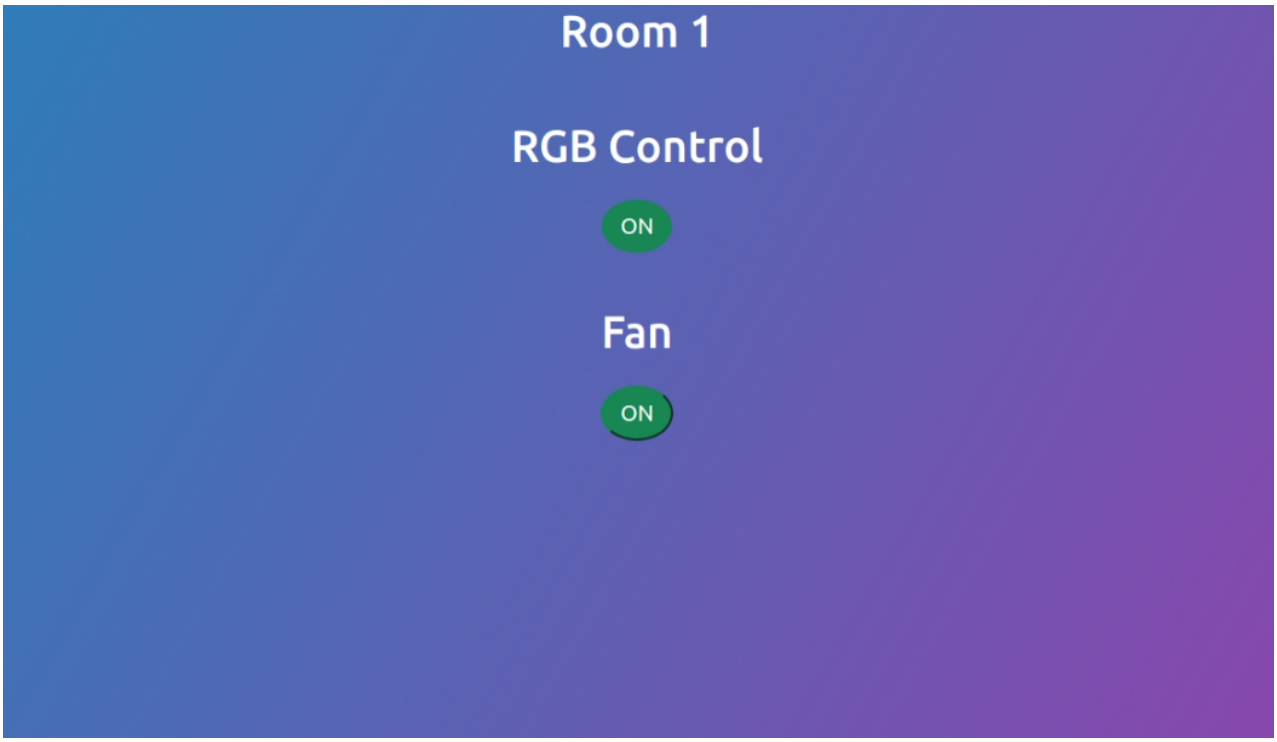

Figure 6.3: Room1 page

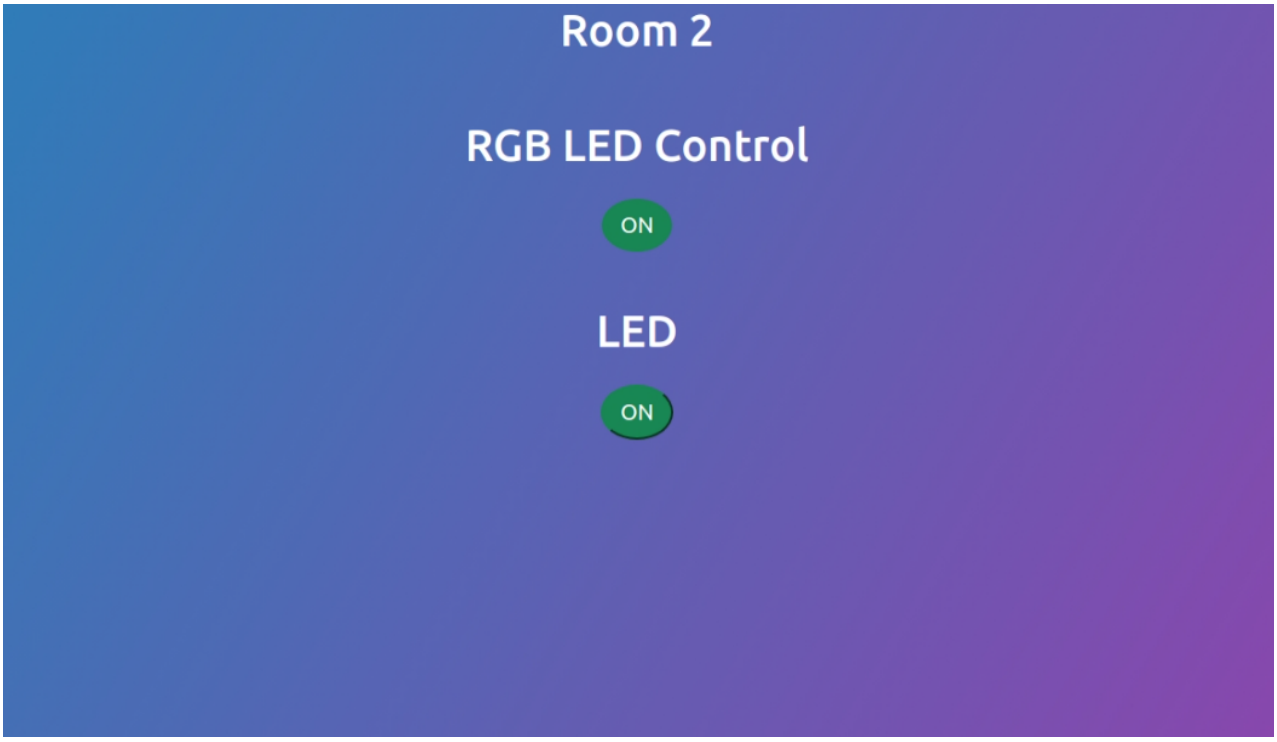

Figure 6.4: Room2 page

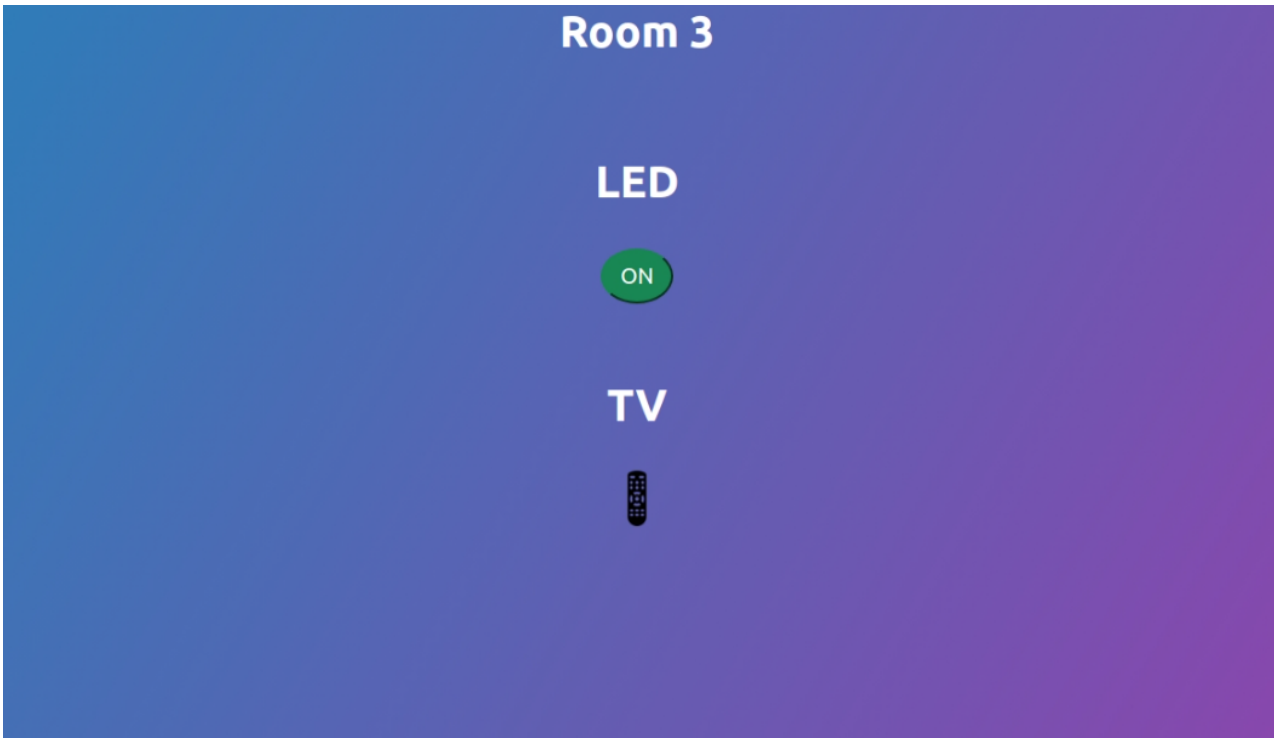

Figure 6.5: Room3 page

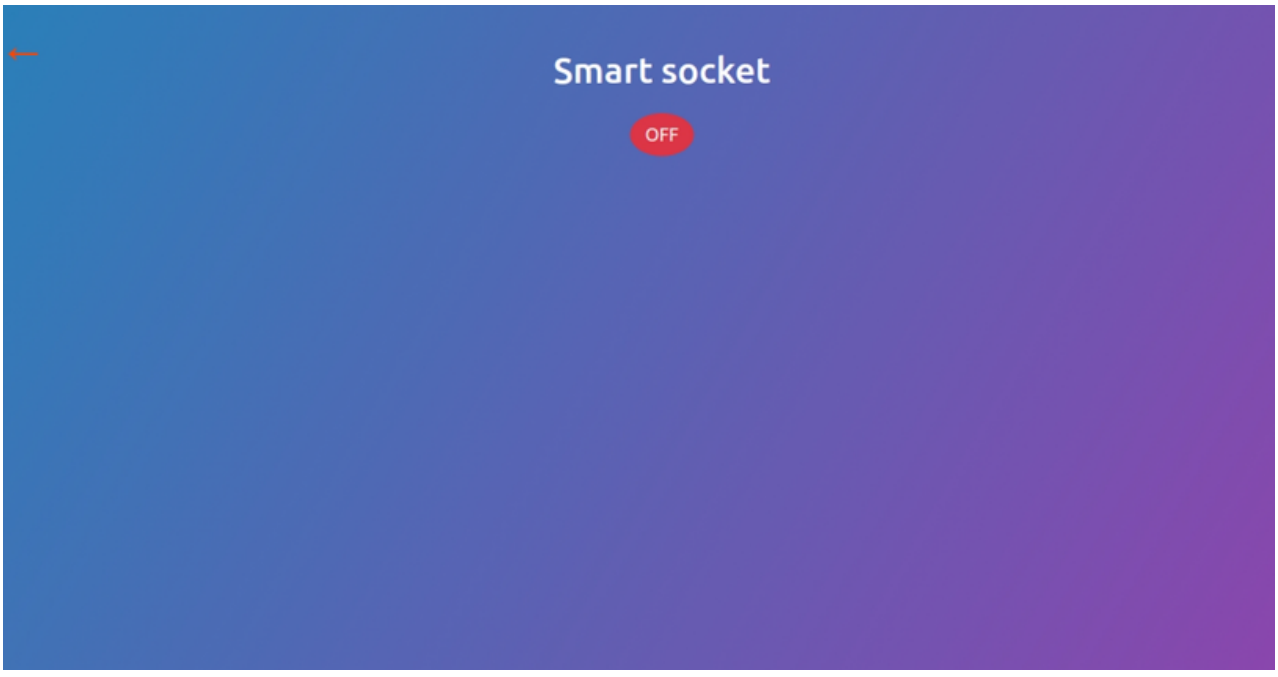

Figure 6.6: Smart socket page

6.4 Devices' controllers

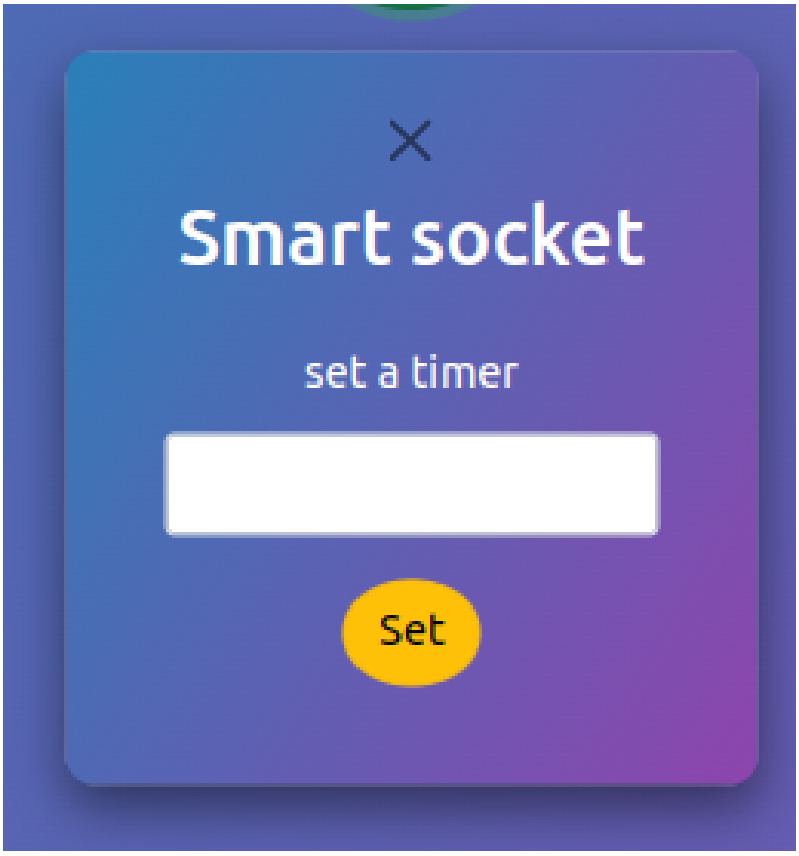

Figure 6.7: Smart socket controller

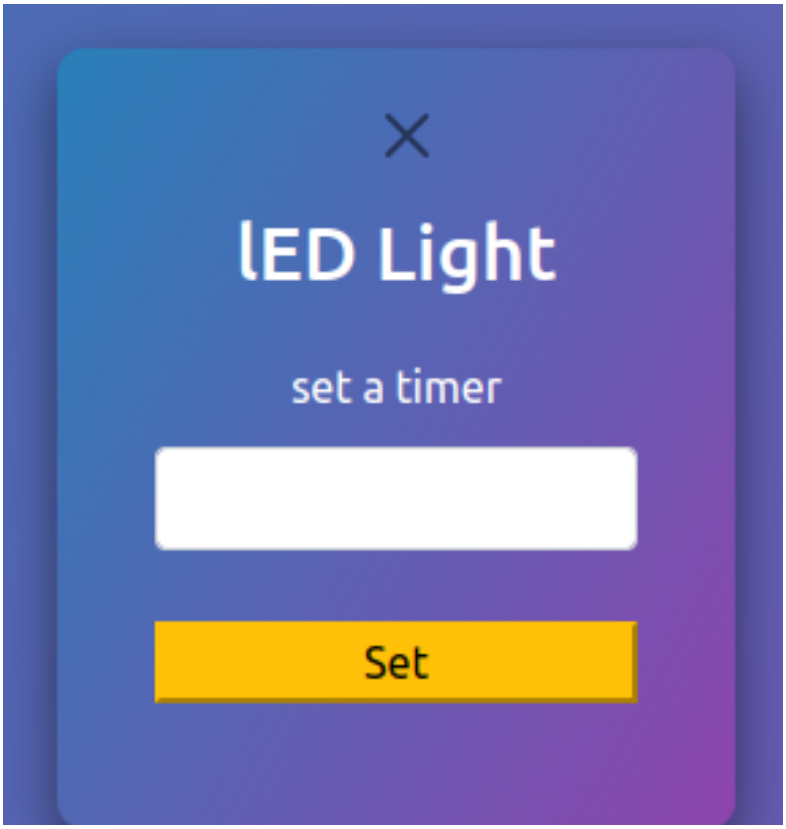

Figure 6.8: LED controller

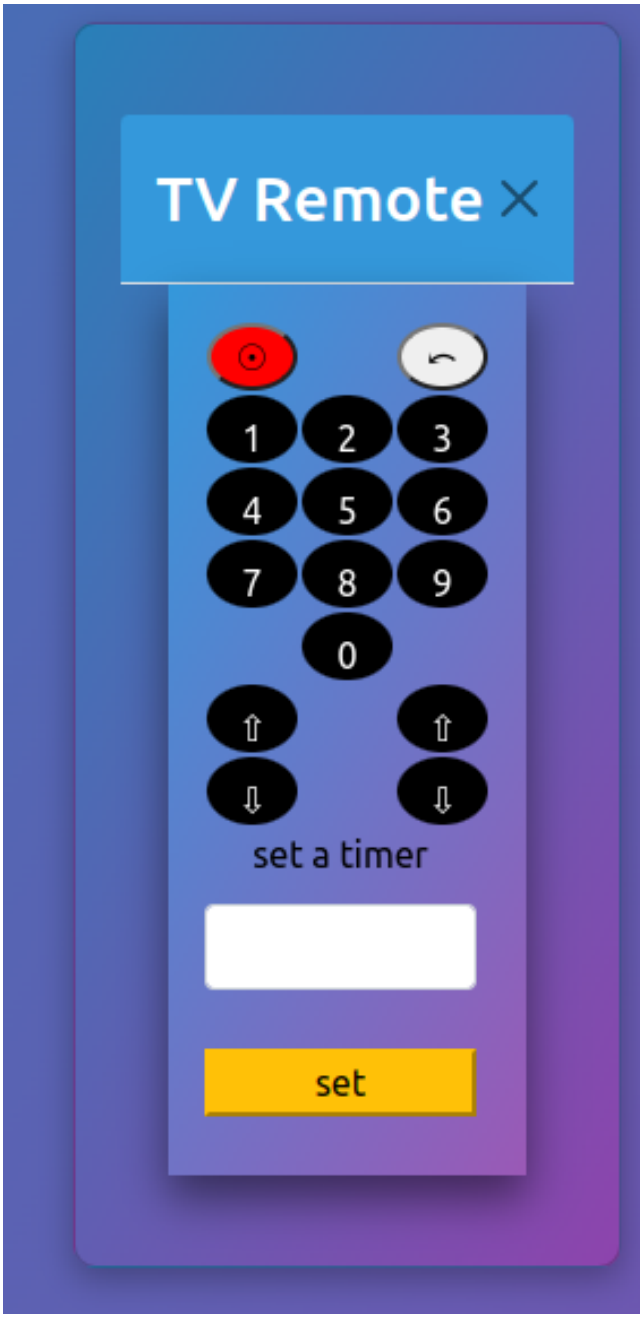

Figure 6.9: Remote controller

Figure 6.10: RGB controller

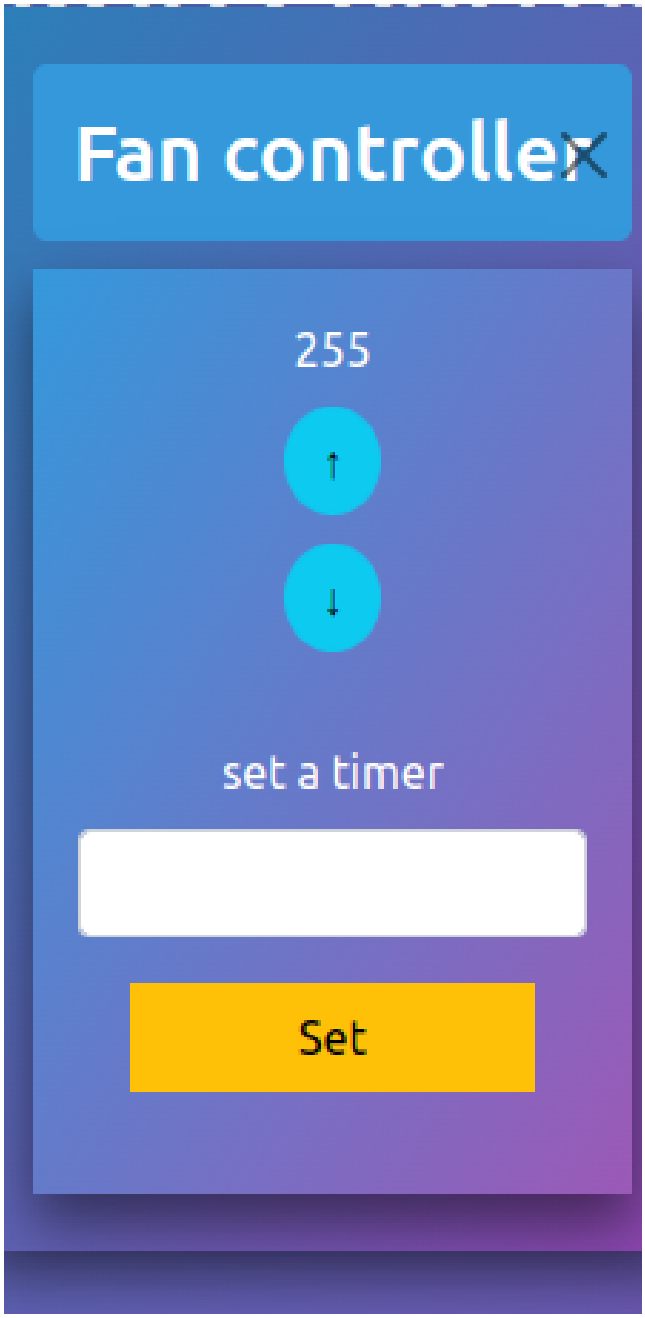

Figure 6.11: Fan controller

6.5 Admin interfaces

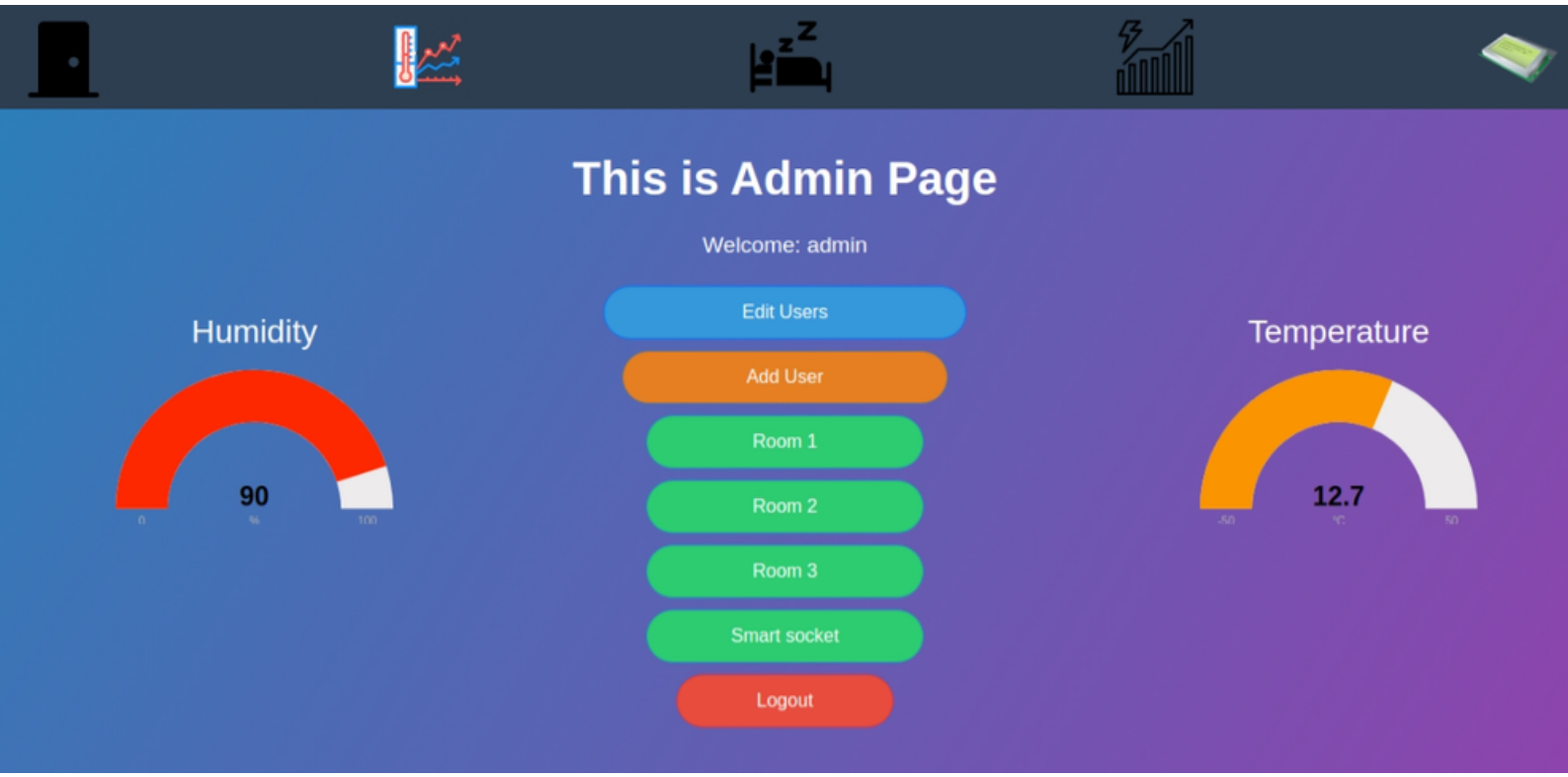

Figure 6.12: Admin's main page

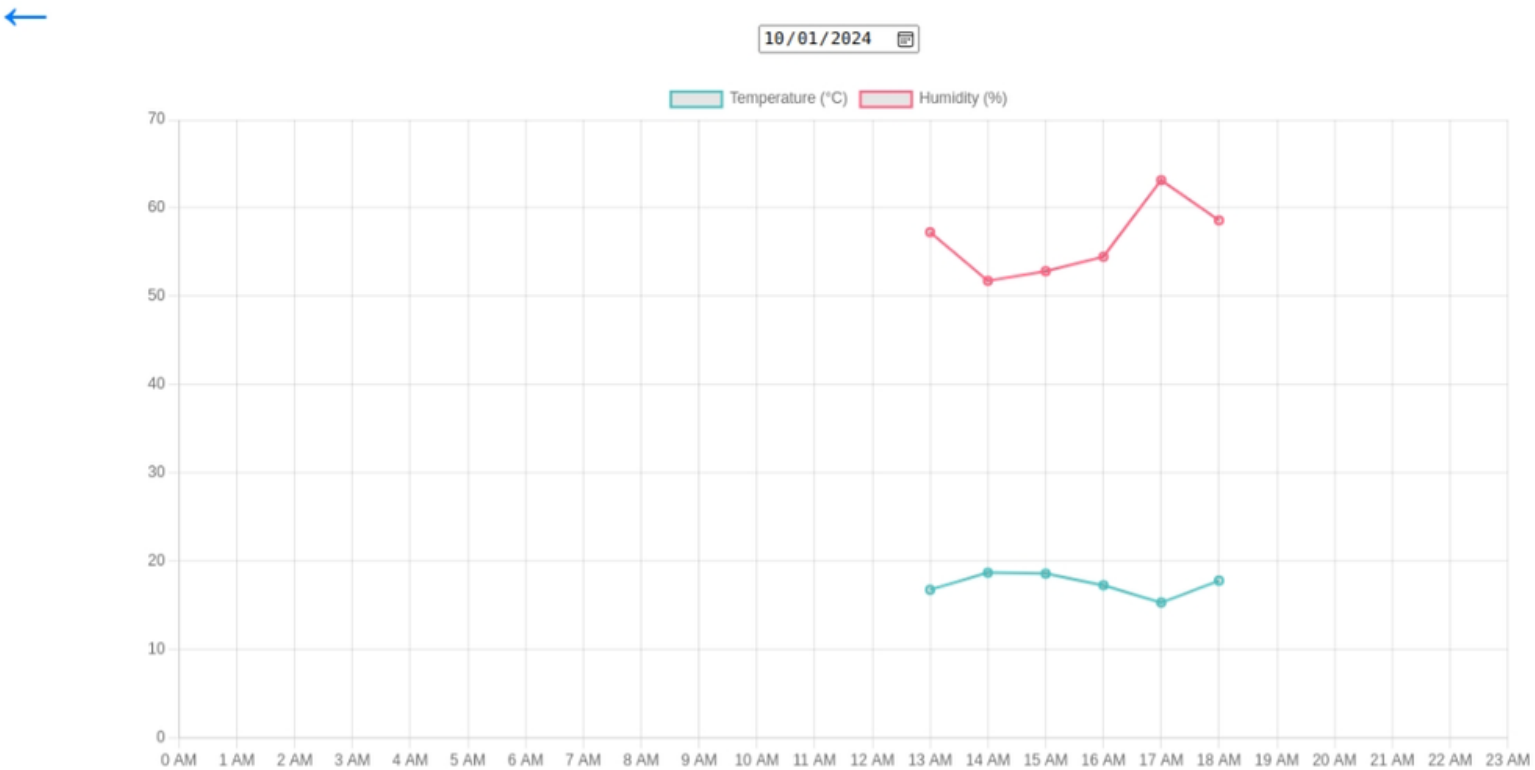

Figure 6.13: Temperature statistics page

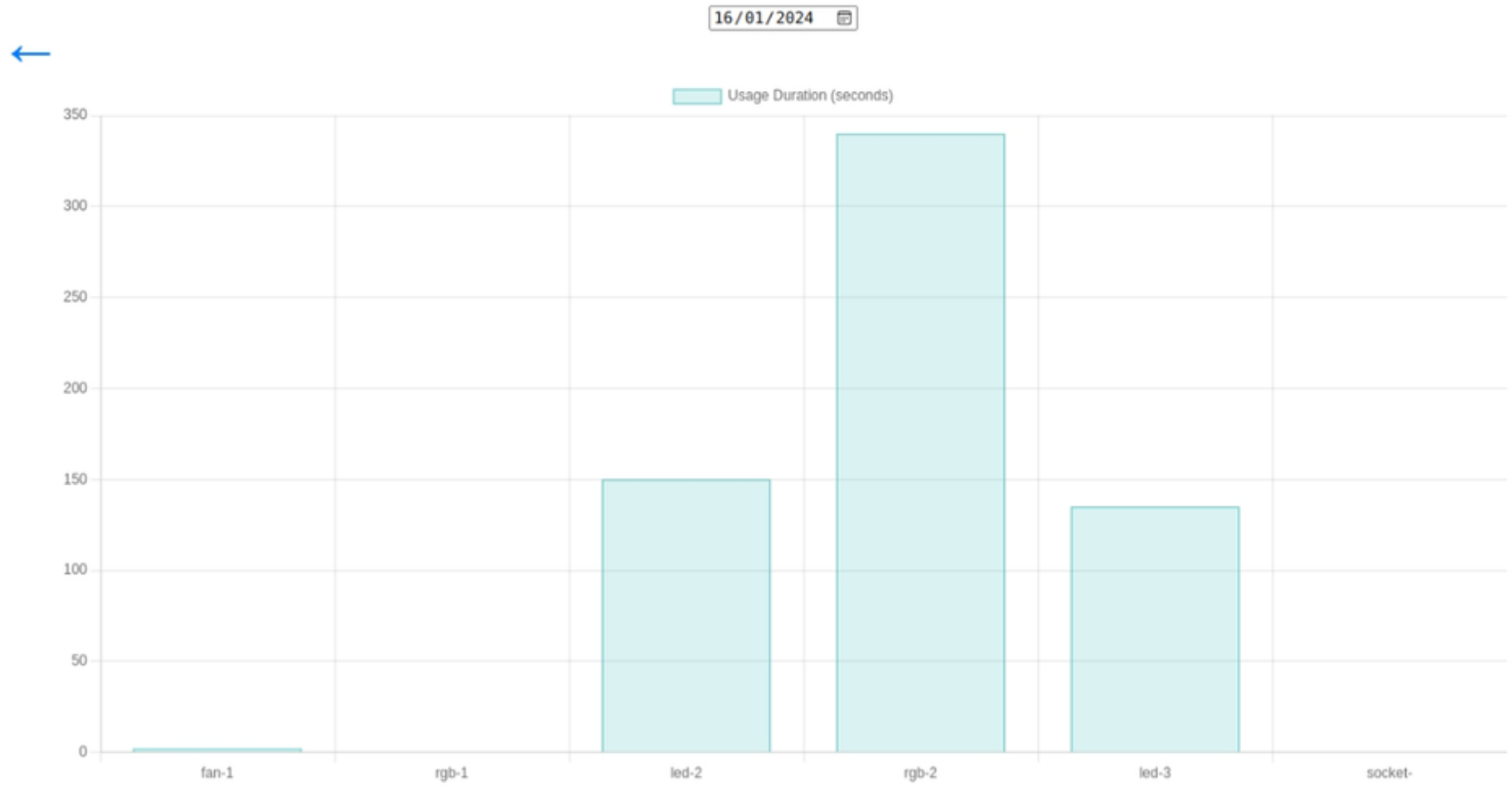

Figure 6.14: Power usage statistics page

Figure 6.15: Devices records page

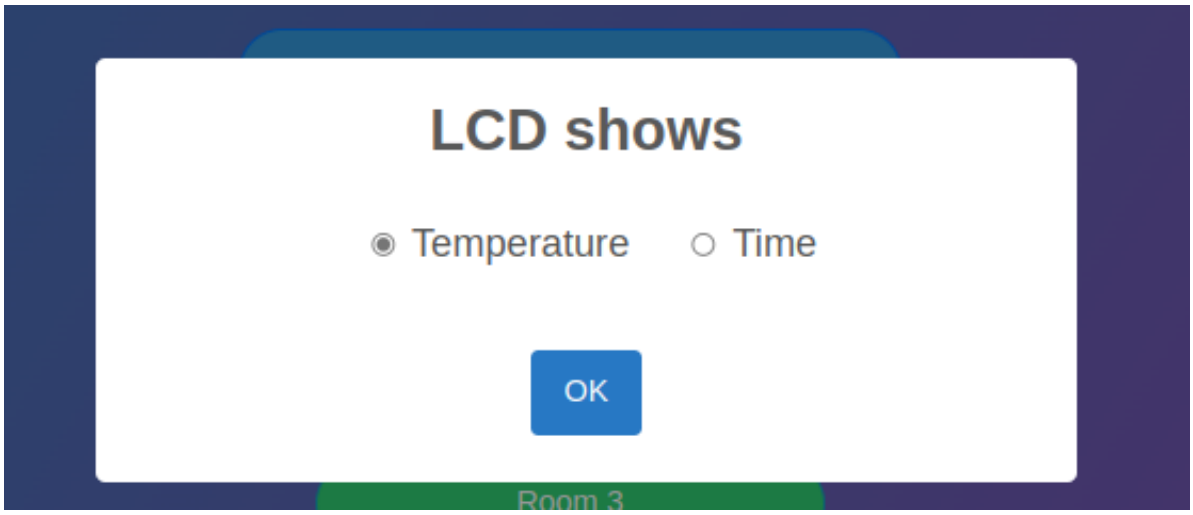

Figure 6.16: Changing LCD display

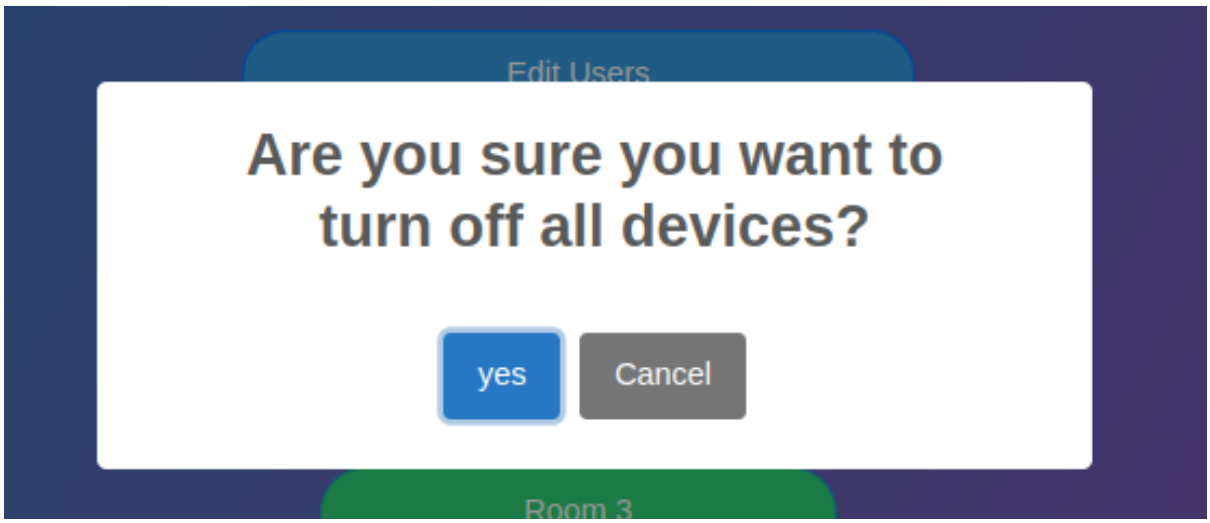

Figure 6.17: Turn off all devices

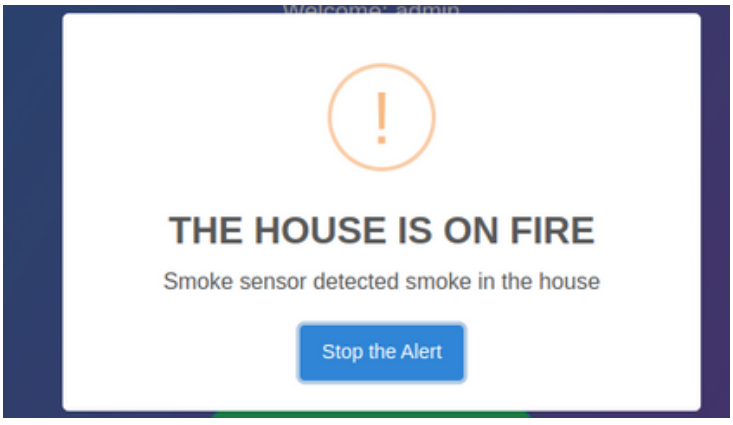

Figure 6.18: Fire alert

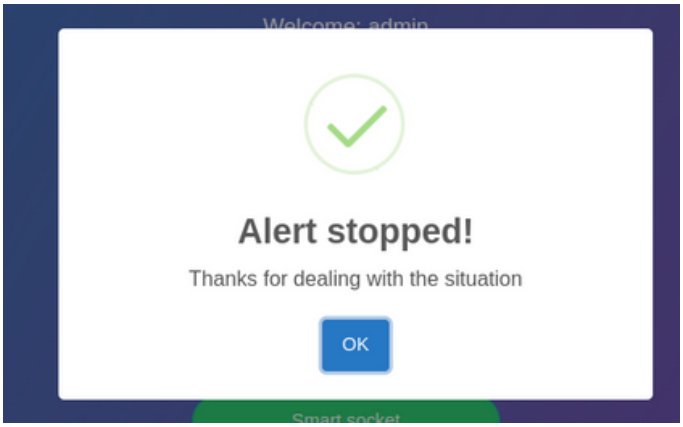

Figure 6.19: Stropping fire alert

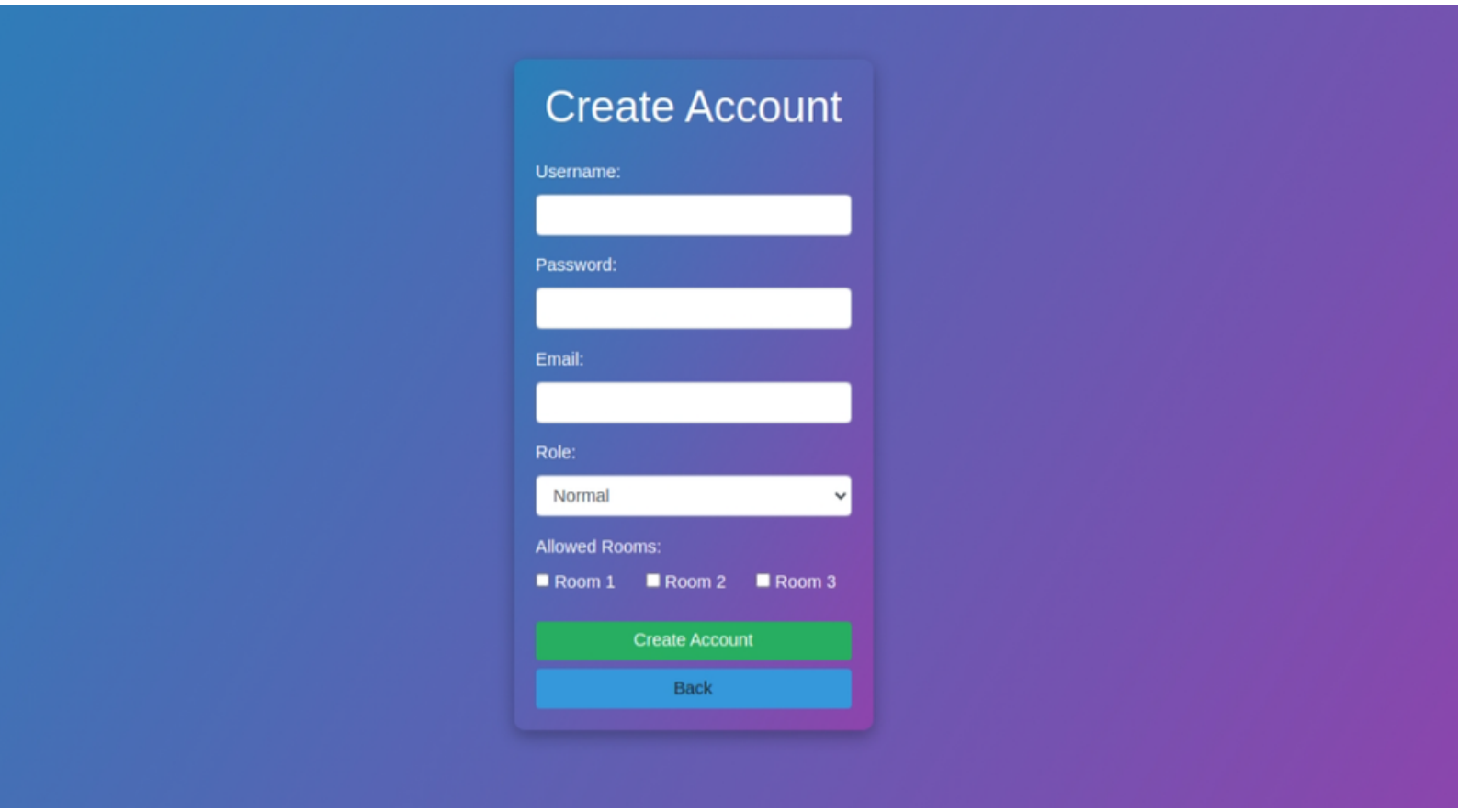

Figure 6.20: Add user page

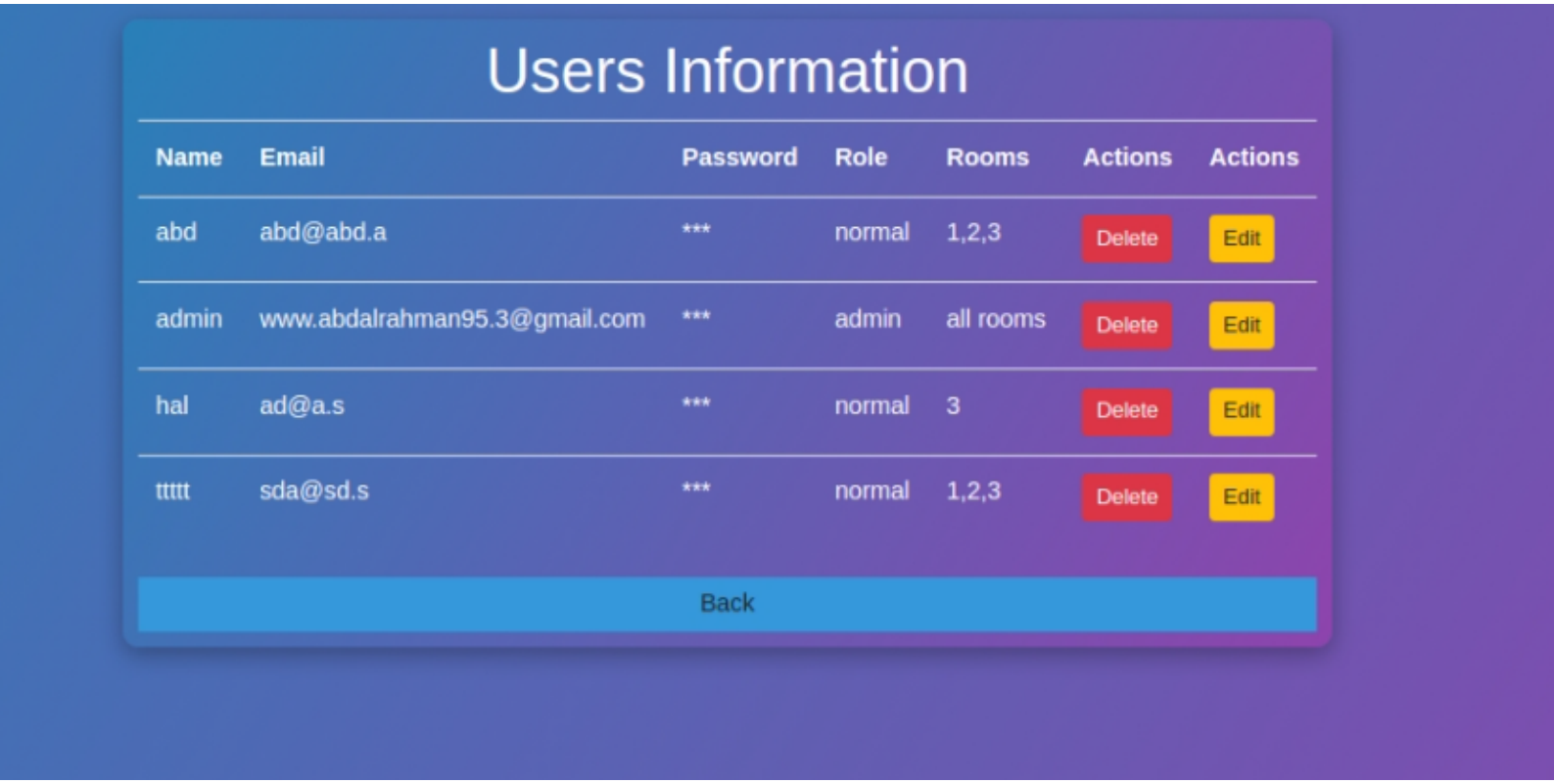

Figure 6.21: Edit user page

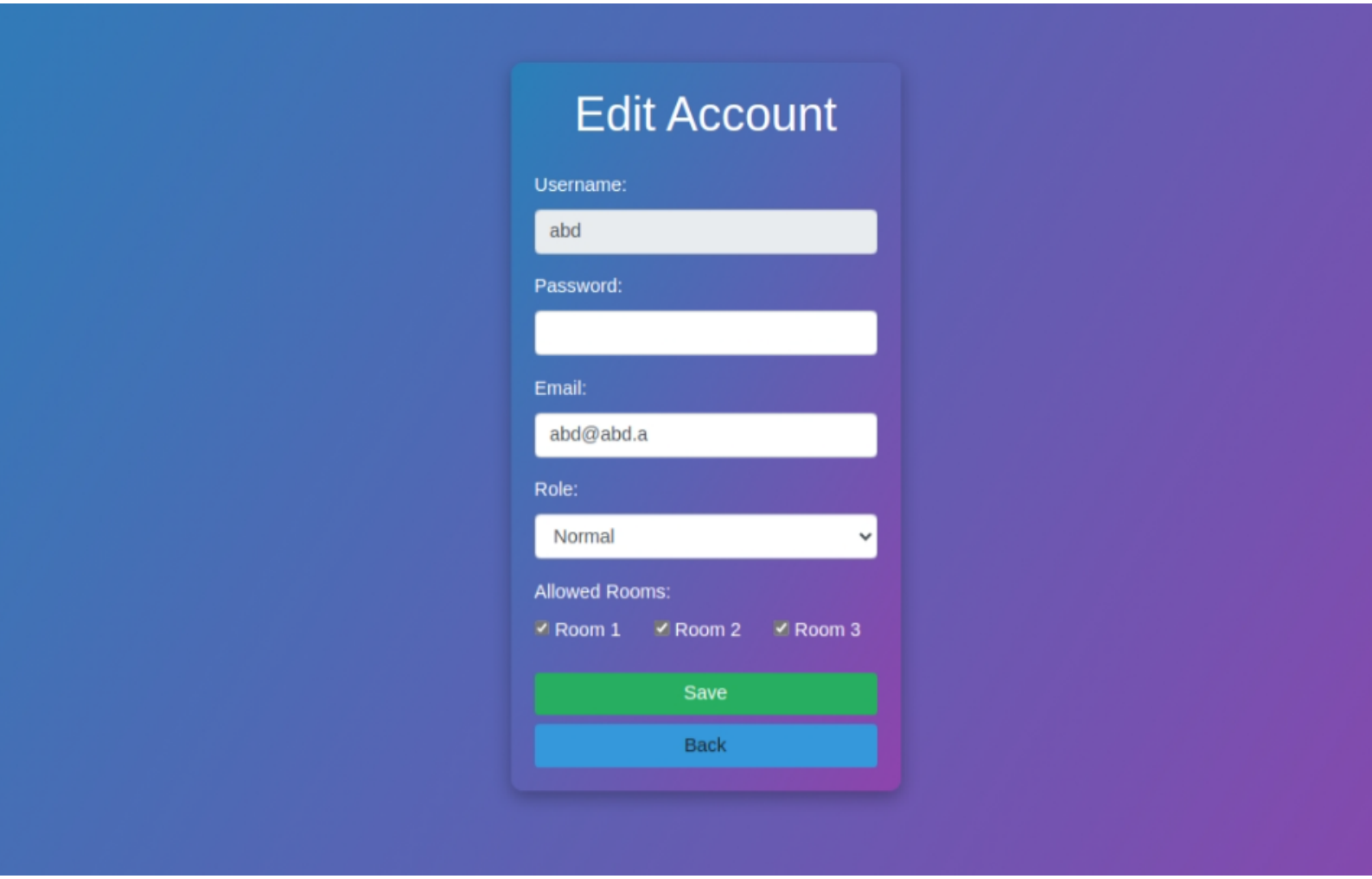

Figure 6.22: Edit a specified user account

6.6 Microcontroller loses connection interface

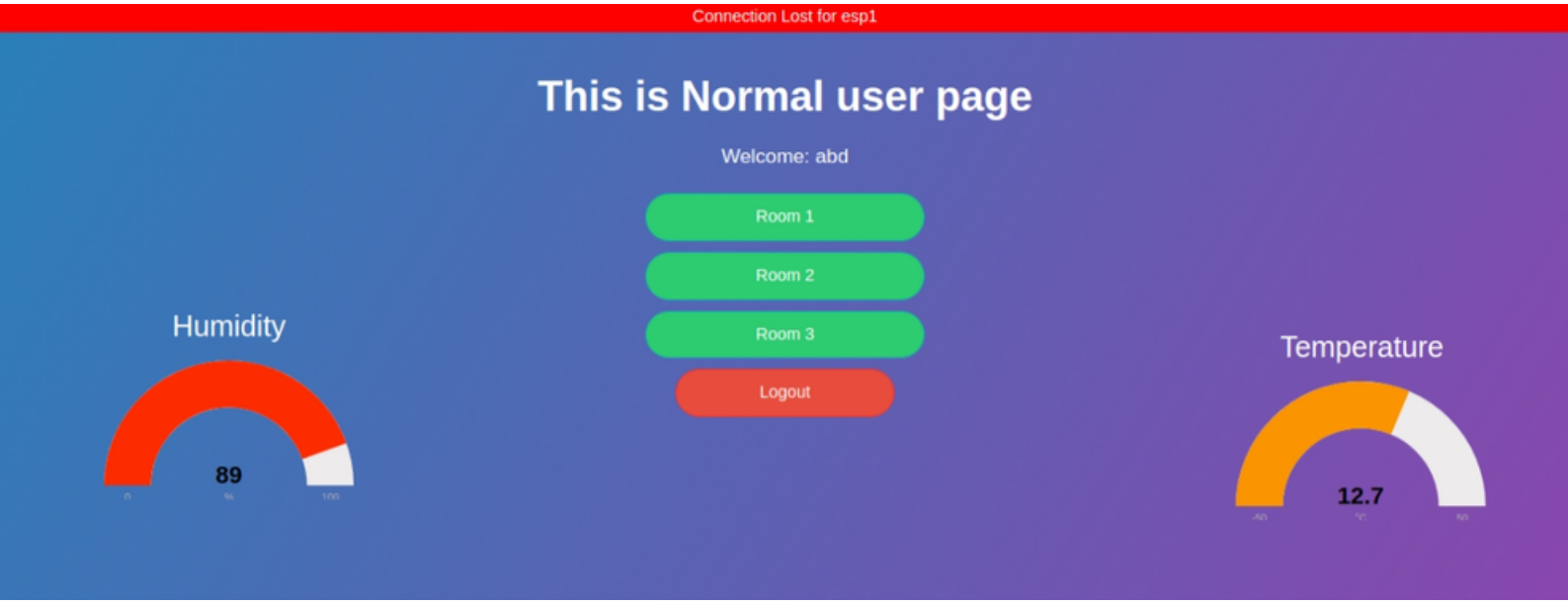

Figure 6.23: The interface of some page when microcontroller loses connection

CHAPTER 7

CONCLUSION & FUTURE VISION

This chapter describes recommendation of project, conclusions, and next steps

7.1 Conclusion

In summary, our system is working great! It smoothly handles all the things it's supposed to do, meeting both the functional and non-functional requirements we mentioned at the beginning. Everything in our home setup works together seamlessly, making it user-friendly and effective. Our project successfully brought together all the necessary features, creating a smart home system that's easy to use and works really well.

7.2 Recommendations

some of the recommendations and improvements that can be done on the project:

- Some modifications on the system especially on the data base to make it easier for the system to serve different homes.
- Some modifications to the system especially the microcontroller code and the data base to make the system able to serve many devices of the same type in the same room (many LEDs for example).
- Consider using MQTT protocol which is widely recognized for its effectiveness in IoT environments.
- Consider the integration of a toolkit like Flutter to extend the system's accessibility by providing both a dedicated application and a user-friendly website interface.

7.3 Next step

- Parts of this system will be implemented in actual homes to assess its performance in real-world scenarios.

References

[4.1] PlatformIO vs. ArduinoIDE
https://stackshare.io/stackups/arduino-vs-platformio#:~:text=Debugging%20Support%3A%20Arduino%20IDE%20has,code%20during%20the%20development%20process.

[4.2] PlatformIO features
https://stackshare.io/stackups/arduino-vs-platformio#:~:text=Debugging%20Support%3A%20Arduino%20IDE%20has,code%20during%20the%20development%20process.

[4.3] what is sweetalert?
https://www.javatpoint.com/sweetalert#:~:text=SweetAlert%20is%20a%20method%20to,depend%20on%20the%20user%27s%20click.

[4.4] justgage library
https://github.com/toorshia/justgage

[4.5] chart.js step by step guide.
https://www.chartjs.org/docs/latest/getting-started/usage.html

[4.6] PHPMailer: Examples, Debugging, SMTP Settings
https://mailtrap.io/blog/phpmailer/

[4.7] 10 Trending Design Patterns in IoT Solutions and Architectures
https://www.linkedin.com/pulse/10-trending-design-patterns-iot-solutions-vishal-bhardwaj

[4.7] ESP32/ESP8266 Insert Data into MySQL Database using PHP and Arduino IDE
https://randomnerdtutorials.com/esp32-esp8266-mysql-database-php/