Palestine Polytechnic University

College of Information Technology and Computer Engineering

Department of Computer System Engineering

"Smart Home Lighting System"

Team members:

Asma Jaafreh

Hiba Almtour

Supervisor:

Dr. Radwan Tahboub

Palestine

May 31, 2022

بسم الله الرحمن الرحيم

﴿يَرْفَعِ اللَّهُ الَّذِينَ آمَنُوا مِنْكُمْ وَالَّذِينَ أُوتُوا الْعِلْمَ دَرَجَاتٍ ۚ وَاللَّهُ بِمَا تَعْمَلُونَ خَبِيرٌ﴾

سورة المجادلة ... الآية (١١)

الإهداء

إلى سقف غرفتي أولاً الذي تحمل جنون أفكاري وخيباتي وتقلبات مزاجي، إلى الحاضرين بروحهم غائبين بجسدهم، إلى أمي الحبيبة التي سهرت معي ليالي الجد والعمل، إلى جامعتي وأساتذتي، إلى أولئك الذين ننتظر خروجهم من خلف قضبان الظلم والاحتلال، إلى الوطن الأغلى والأحب لقلبي وروحي فلسطين، ولكل من أهداني يومًا بريق أمل لأكمل واستمر... أهدي أول ثمرات نجاحي، وادعو الله بأن تكون أولها وليس آخرها وأن يعمر قلبي بحب السعي للعلم والعمل.

Acknowledgment

A part of our effort and the successes of any project depends largely on the encouragements of others.

We would like to start with a special gratitude to our university for the help and support provided in its facilities for the past five years, and for our computer engineering college for the large effort had been spent during our study to reinforce us with all knowledge needed.

And no thanks could be enough for our supervisor Dr. Radwan Tahboub, for his support, effort and guidelines he gave us during the course, special thanks to Eng. Wael Takroury for his support and help.

Abstract

A smart city is a future trend solution that uses various electronic Internet of Things (IoT) sensors to collect data. One of the steps to reach the smart city is the smart lighting system, which uses IoT-enabled sensors, bulbs, or adapters to allow users to manage their home's lighting with their smartphone or smart home management platform.

This project presents a design and implements a smart lighting with IoT technology and standards to control the homes lighting intensity remotely by controlling some resources which affect to light intensity such as (curtain motor, etc.), connect the controller with smartphone via Ionic standard application in IOS and Android, also we used API to link between the databases on the server and the mobile application . This controller can deal with multiple sensors and take one or more actions.

The system will be built and verified to meet the requirements of this project, ensuring that it performs as it should and meets the needs of the project.

CONTENT

List of Tables:

List of Figure:

Chapter 1

1 Introduction

1.1 Overview

A smart city is a future trend solution that uses various electronic Internet of Things (IoT) sensors to collect data. One of the steps to reach the smart city is the smart lighting system, which uses IoT-enabled sensors, bulbs, or adapters to allow users to manage their home's lighting with their smartphone or smart home management platform. Smart lighting solutions can be built in a couple of ways. Smart bulbs are WiFi-enabled bulbs that can be controlled individually by a smart assistant or mobile app. Most have the ability to dimmess the light. Light switches operate as an adapter for regular light bulbs and can control groups of lights.[1]

We also aim in our project to develop an intelligent system that controls and regulates the intensity of ambient light so that it is adjusted to be suitable for eye comfort.

1.2 Motivation

Where you work, learn, or relax, the place you're in makes a difference. Whether enhancing wellbeing, boosting productivity or just making you feel good.

In this project, we aim to achieve comfort for homeowners and provide more comfort for their eyes from annoying lighting that destroys eye health. So, they can control the lighting of their homes, any part of it in a smart way without causing any effort. This is done using a smart application with automatic and manual controls that enables them to increase or decrease the intensity of the lighting and open or close part of the curtains.

1.3 Importance

One of the most benefits of smart lighting system is providing peace of mind to homeowners, allowing them to monitor and control their homes' lighting remotely. For example, users can program their lights to be on and curtains closed, upon their arrival at night so they never come back to a dark home again. This improves efficiency Instead of leaving the lights on all day.

There are many other benefits of this system such as:

1- **Smart lighting for a good night's sleep.**

   Smart lighting system can automatically turn the light off throughout your bedroom after prayer of eshaa (Our smart system is flexible as it allows the user to enter prayer schedules or set other appointments as desired). This will allow your body to produce melatonin (a hormone that helps to regulate your sleep cycle which the body starts producing it when the light outside dims). Naturally, increasing your chances of having a good night's rest.

2- **Smart lighting for waking up.**

   Smart lighting system can automatically open part of the curtains in your bedroom at sunrise (When the user determines that the height of the window is for example 50% when it is 6:00am). This will allow your body to produce cortisol (a hormone to help regulate your sleep cycle, as it is responsible for waking you up and keeping you alert throughout the day).

3- **Smart lighting and eye comfort.**

   There are many different studies that show the effect of the intensity of inappropriate lighting on the comfort and health of the eyes. From these studies, the researcher Khairat ElBaradei, who prepared a research on the subject, showed that a research was conducted on mice, and one of the results was to reach the danger of the white “neon” bulbs on the eye and the abilities of to see.

   Doctors also warn against the effect of strong lighting, as well as weak lighting, which causes eye fatigue. If the lighting is uncomfortable, some signs of stress will appear, such as burning, redness, itching and headache.[2]

   Based on what was mentioned and many other studies reviewed, we found the importance of our project and the strong need to provide appropriate lighting intensity to achieve a certain level of eye comfort.

   Smart lighting system can automatically providing the appropriate light intensity by dimming of the lighting throughout your house.

1.4 Objectives

The main objective of this project is to maintain eye comfort by providing the appropriate light intensity by controlling curtains opening/closing and the lights intensity in room increasing/decreasing (automatically or manually). Our smart system is flexible because it is based on human preferences as each user has the ability to set their own schedule using the smart mobile application for example: “Alex likes the lights to be high between 10-12 AM but when it is 5 PM, she likes low light ".

1.5 Short Description of the System

The system will use the Internet of Things (IoT) Technologies to develop a smart system that controls and regulates the intensity of ambient lighting so that it is adjusted according to several factors, including eye comfort. This system mainly depends on sensing the intensity of ambient light caused by different light sources while a person is working, such as lighting through windows, television, mobile phones while working on it, as well as home lighting (room). Some of these sources can be controlled and some cannot be controlled completely or directly. For example, natural lighting is difficult to control directly, but it is possible to close or open part of the curtains for example, to reduce or increase the intensity of natural lighting assuming that we are during the day. While during the night, the electric lighting is usually adjusted by changing the intensity of lighting in the room increasing or decreasing.

Figure 1.1 shows that our smart system consists of three blocks: hardware component (sensors, ect..), server and mobile application. The system will use sensors to carry out the previous operations, and these sensors will be connected with the microcontroller, where the data will be downloaded to it and sent via the Internet to a mobile application so that the user can see all the data.

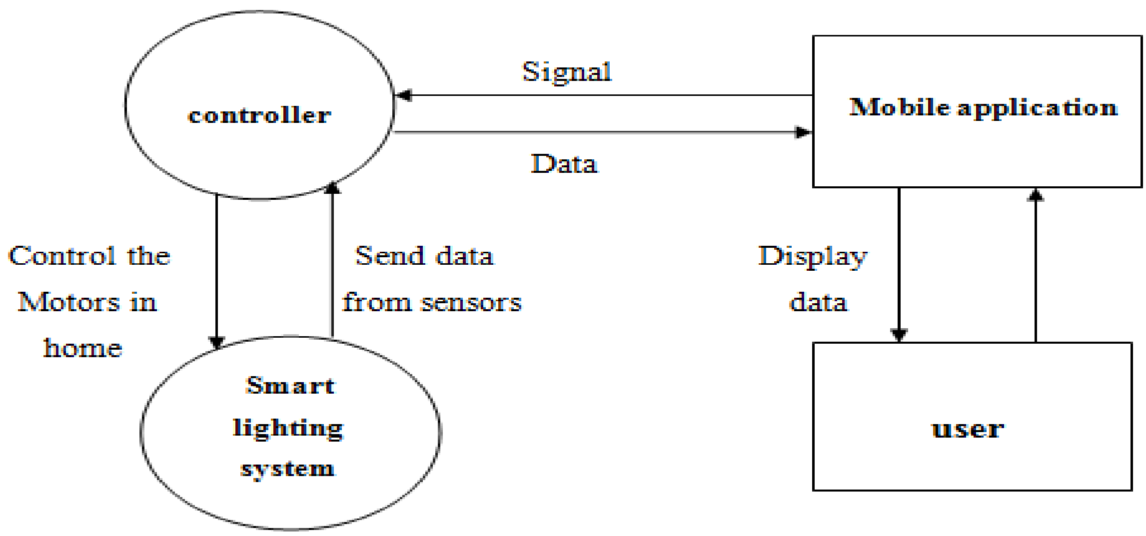


Figure 1.1: Context Diagram of Smart Lighting System.

1.6 Problem Statement

1.6.1 Problem Analysis

Improper light affects eye comfort, as it leads to dry, irritated eyes and blurred vision. Blue light may also damage the retina. This is called phototoxicity. Research shows that exposure to blue light may lead to the release of toxic molecules into the photoreceptor cells. This causes damage that may lead to age-related macular degeneration, or AMD. [3]

1.6.2 Requirements

The system should have the following requirements:

Functional Requirement:

Control the intensity of the lights in the room, by controlling the following items:

1) Control curtains open/close.
2) Control lights in room.
3) Control other resources (like TV light recourse.), for future work.

A mobile application should meet the following requirements:

- Control buttons for opening and closing curtains.
- Control buttons for increasing and decreasing the intensity of lights.
- switch buttons for Auto light
- switch buttons for Auto window height

- Control buttons for adding a new schedule whether for light or for curtain.
- Select box for choosing light intensity or curtain height.
- Control buttons for save.
- Control buttons for setting.

We need a Database in our project to require data processing, analytics, and storage.

Non-functional Requirement:

- Availability: The system can be accessed from any device, anywhere and anytime using the smart mobile application as long as there is internet.
- Scalability: The system is scalable as it can be applied to multiple homes or an entire city by increasing the number of users and hardware components.
- Simplicity: The system is simple for anyone to use, whether they are expert in technology or not.
- Ease of access: the access of the system will be easy from mobile application which is easy to use.
- Security: we use SHA 256 for hashing the password which provide security to the system.

1.6.3 Expected Results

At the end of the project, it is expected to achieve all its objectives and requirements , also design and implement smart lighting system based on Arduino, using mobile application as interfaces to provide the easiest environment for the users allowing them to monitor and control their homes lighting system remotely.

The smart lighting system is user friendly, improve efficiency and Providing eye comfort, which reduces the possibility of exposure to some diseases such as (irritated eyes).

1.7 Overview of the Rest of Report Sections

Report consists of six chapters; the following is a brief description of the topics that are covered in each chapter:

Chapter 1: Introduction

This chapter presents general idea about the project , motivation and importance, objectives, short description of the system, and Problem Statement (Problem analysis, list of requirements, expected results).

Chapter 2: Theoretical Background

This chapter talks in more details about the basic component used in the project, theoretical background, literature overview, and design constraints.

Chapter 3: Design

This chapter details the design concepts and shows the general block diagram of the system, schematic diagrams, and explains how system works.

Chapter 4: Software and Hardware Implementation (Second Semester).
This chapter describes the implementations of the software and the hardware components that are used in this project, such as the circuit connection, microcontroller, the IDEs used to build the project codes, and the user interface.

Chapter 5: Validation and Testing (Second Semester).
In this chapter, we will describes and discusses the steps, results of the implementation, and testing of the hardware and software parts.

Chapter 6: Conclusion (Second Semester).
In this chapter gives information about conclusion and provides for more feature that can be done in the future.

Chapter 2

2 Background

2.1 Overview

This chapter describes briefly the theoretical background of the project, literature overview, short description of the hardware and software parts that are used in the system, and design constraints.

2.2 Theoretical Background

This section will provide some information about some used technologies in the project.

IoT and Cloud Computing Overview

The internet of Things (IoT) is influencing our lifestyle from the way we react to the way we behave. From smart lighting that you can control with your smart phone to smart Cars providing the shortest route or your Smart watch which is tracking your daily activities. IoT is a giant network with connected devices. These devices gather and share data about how they are used and the environment in which they are operated. It’s all done using sensors, sensors are embedded in every physical device. It can be your mobile phone, electrical appliances, Pecos barcode sensors and almost everything that you come across in day-to-day life. These sensors continuously emit data about the working state of the devices. IoT provides a common platform for all these devices to dump their data, and a common language for all the devices to communicate with each other. Data is emitted from various sensors and send to IoT platform security, IoT platform integrates the collected data from various sources further analytics is performed on the data and valuable information is extracted as per requirement. Finally the result is shared with other devices for better user experience automation and improving efficiencies.

Cloud computing: is a model for on-demand access to a shared pool of configurable resources (e.g., computers, networks, servers, storage, applications, services, software) that can be provisioned as Infrastructure as a Service (IaaS) or Software as a Service (SaaS). “One of the most important outcomes of the IoT is an enormous amount of data generated from devices connected to the Internet. Many IoT applications require massive data storage, huge processing speed to enable real time decision making, and high-speed broadband networks to stream data, audio, or video. Cloud computing provides an ideal back-end solution for handling huge data streams and processing them for the unprecedented number of IoT devices and humans in real time”.[4]

2.3 Literature Overview

First Previous Graduation Project:

“IoT Smart Home Controllers**”**

This project is one of the applications on the home automation system.

Authors:

- Sojood Othman Sweiti.
- Samah Yacoub Fannoun.

Year: 2021

University: Palestine Polytechnic University.

Summary:

“*Smart home is a term that is commonly used to refer to homes where the appliances, light system, air conditioning, TVs etc. are capable of communicating with each other and can be controlled remotely according to a predefined schedule or via some kind of interface. This project presents a design and implements a smart controller compatible with IoT technology and standards to control home devices remotely such as (light system, water motor and curtain motor, etc.), connect the controller with smart phone via Ubidots standard application in IOS and Android compatible with IFTTT and Google Assistant. This controller can deal with multiple sensors and take one or more actions*.”[5]

The difference between our project and previous project :

- In terms of smart lighting, the previous project was only controlled turning on and off the smart lighting , but in our project we will specialize and expand in controlling the intensity of the smart home lighting to become suitable for eye comfort.
- In the previous project, the curtains were opened and closed by user action either using a mobile app (remotely) or manually, but in our project, the curtains were opened and closed automatically according to the sunrise and eshaa prayer (without human interaction) in order to provide suitable light intensity for eye comfort.

Second Previous Graduation Project:

"IoT Smart Home Lighting System using Arduino UNO and ESP8266 **"**

This project is one of the applications on the home automation system.

Authors:

- Kai Loung, Vincent Yong; Yen Leow, Chee; Jannatheran, Edwin Nair

Year: 2021

Summary:

"Today, a wide variety of IoT applications and services have appeared and one of them is the Smart Home. The objective of this article is to present basic electrical wiring using Arduino UNO board and ESP8266, writing in C programming Language in Arduino software Integrated Development Environment (IDE) to develop and build a prototype Smart Home Lighting System that displays the lighting of two fluorescent lamps when the switch button was turned on by the user from ESP8266 Web Server. The system installation has a real-time synchronization that is necessary and safe for immersive IoT applications."[6]

The difference between our project and previous project :

- The previous project was using ESP8266 microcontroller, but in our project we use ESP32 microcontroller.
- In our project, we control light intensity according to eye comfort by controlling several factors like opening/closing of curtains, but in the previous project, they don't deal with light intensity.

.**Third Previous Graduation Project:**

“A smart lighting system using wireless sensor actuator network”

This project is one of the applications on the home automation system.

Authors:

- Sanjib Kuamr Panda
- Arun Kumar

Year: 2017

University: National University of Singapore.

Summary:

“Indoor lighting accounts for 15-20% of the total energy consumption amongst electrical loads in residential buildings. This paper presents the design, implementation, and testing of a smart lighting system for better visual comfort, high reliability along with energy saving. Environmental conditions such as natural daylight, interior light intensity level, and occupancy state are gathered from distributed sensors. Based on sensors data, the microcontroller is programmed to control lighting intensity and to achieve better energy efficiency. Light-emitting diode (LED) dimmer circuit is designed to be controllable by the PWM signal of the microcontroller and is tested for different types of LED lamps. Dimmable LED lighting system incorporated with smart wireless sensor control is able to achieve a significant amount of energy saving.”[7]

The difference between our project and previous project :

- The previous project was controlling the dimming of the light load according to the dimming theory, but in our project we will specialize and expand the control of the intensity of the smart home lighting to become suitable for eye comfort by controlling several factors such as opening / closing the curtains.
- In our project, we provide the design and implementation of a smart lighting system in order to provide an appropriate light intensity for eye comfort but the previous project focused on energy saving. We also have a smart app that allows users to control their home lighting remotely, which was not available in the previous project.

2.4 Hardware Components

In this section we illustrate the hardware components that can be used to build the system:

2.4.1 Light Dependent Resistor Sensor (LDR)

LDR is a component that has a (variable) resistance that changes with the light intensity that falls upon it. This allows them to be used in light sensing circuits. It is used to measure the light intensity, then send it to the microcontroller to take the right decision for dimmers motor.[8]

LDR Specification :

Application	Light Dependent Resistors, LDRs are often used in circuits where it is necessary to detect the presence or the level of light
Max Power	200 W
Resistance	50-100K OHW
Diameter	3-20 Mm

Table 2.1: LDR Specification

2.4.2 Microcontroller:

ESP32

ESP32 is a series of low-cost power systems on a chip microcontroller. It is an advanced version of the ESP8266 series. And it was developed for the lack of security, which was in ESP8266.

This subsection illustrates the alternative for ESP32 and explains why we don’t use it.

- **Raspberry pi**

Raspberry Pi is a single-board computer. It is a credit-card-sized computer with low cost, which plugs into a computer monitor or TV, and to operate it, a user can use a standard keyboard and mouse to operate it. The single-board consists of a fully functional computer with its dedicated memory, processor and it runs an operating system.[9]

This table shows the main difference in specifications for ESP32 and Raspberry Pi .[9]

FEATURE	RASPBERRY PI	ESP 32
Core count	Dual-core	Single/dual-core
Clock frequency	48kHz / 133kHz	160kHz / 240kHz
Internal Flash Memory	2 MB	4 MB
GPIO(total)	26	34
WiFi	X	✓
Bluetooth	X	✓
Ethernet	X	✓
Price	$4	$4 – $12

Table 2.2: ESP 32 VS Raspberry Pi 3

- **ESP 8266**

All ESP8266 variants have an ESP8266EX core processor and a Tensilica L106 32- bit microcontroller unit. This is a low cost, high performance, low power consumption, easy to program, wireless SoC (System-On-Chip).[10]

This table shows the main difference in specifications for ESP8266 and ESP32.[10]

FEATURE	ESP8266	ESP32
MCU	Xtensa Single-core 32-bit L106	Xtensa Dual-Core 32-bit LX6 with 600 DMIPS
Hardware/Software PWM	None / 8 channels	None / 16 channels
Bluetooth	**X**	Bluetooth 4.2 and BLE
Typical Frequency	80 MHz	160 MHz
SRAM	**X**	✓
Flash	**X**	✓
GPIO	17	34
Price	(3 - 6)$	(6 - 12)$

Table 2.3: ESP8266 VS ESP32.

2.4.3 Motors: There are different options of the motors that can be used in our project:

- **First Option: DC Motor**

DC motors are electromagnetic devices that use the interaction of magnetic fields and conductors to convert electrical energy to mechanical energy for rotation. There are many types of DC motors out in the market. The brushed and brushless motors are the most common DC motors.[11]

- **Second Option: Stepper Motor**

Stepper motors are motors that move in slow, precise and discrete steps. Valued for their precise position control, they find a myriad of applications such as desktop printers, security cameras, and CNC milling machines. [11]

Table 2.4 summarizes the main features that may affect our choice. [11]

Characteristics	DC Motors	Stepper Motors
Control characteristics	Simple; no extras needed	Simple; microcontroller needed
Speed Range	Moderate (depends on type)	Low (200-2000 RPMs)
Reliability	Moderate	High
Efficiency	Average	Low
Torque/speed characteristics	High torque at low speeds	Maximum torque at low speeds
Cost	Low	Low

Table 2.4: DC Motors Vs Stepper Motors

Chosen Option:

We will choose DC Motors, because the speed range for DC motors is better than stepper motors.

2.4.4 Dimmers:

Dimmers are devices connected to a light fixture and used to increase or decrease voltage to a dimmable lighting fixture in order to adjust its intensity and lower the brightness of light. [12]

The three main types of dimmer switches:

- Resistive Dimmers
- Fluorescent Dimmers
- Inductive Dimmers

They are used to control the intensity of lights to provide light that is suitable to the eye comfort**.**

2.4.5 Mobile

To use the mobile application, to control the intensity of lights and curtains remotely.

2.4.6 Relay:

The relay module is an electrically operated switch that allows us to turn on or off a circuit using voltage and/or current much higher than a microcontroller could handle. There is no connection between the low voltage circuit operated by the microcontroller and the high power circuit. The relay protects each circuit from each other. [13]

2.4.7 PIR Motion Sensor

The PIR sensor itself has two slots in it, each slot is made of a special material that is sensitive to IR. When the sensor is idle, both slots detect the same amount of IR and the ambient amount radiated from the room or walls. When a warm body like a human or animal passes by, it first intercepts one half of the PIR sensor, which causes a positive differential change between the two halves. So, this sensor is very useful to detect the motion.

The PIR acts as a digital output so all you need to do is listen for the pin to flip high (detected) or low (not detected). Power the PIR with 5V and connect ground to ground. Then connect the output to a digital pin.[14]

The PIR motion sensor has three pins:

1. VCC: Power (5-12V)
2. GND: Ground
3. OUT: Digital output signal (0 or 1)

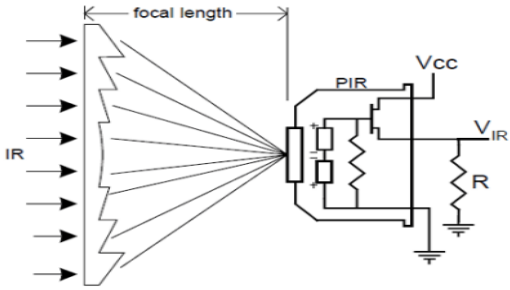


Figure 2.1: PIR Sensor Schematic[14]

PIR Sensor Specification :

Application	Motion-activated nightlight
	Alarm systems
	Robotics & Holiday animated props
Supply Voltage	3.3V DC to 5V DC
Detection range	6 meters
Size	Small
Frequency	2.4 GHz

Table 2.5: PIR Sensor Specification

2.4.8 Light Source (LED)

This is complementary to the LDR sensor.

2.4.9 Wires

Used to connect components.

2.5 System Scope

Smart home lighting systems will be the necessity of recent and coming generations. It is time saving, energy saving, and beneficial for all people. It will add comfort in the life of people which they deserve by providing more comfort for their eyes from annoying lighting that destroys eye health, which leads to enhancing wellbeing, boosting productivity or just making you feel good.

A smart light solution integrates four distinct components: sensors/devices, connectivity, data processing, and a user interface. Our smart home lighting system falls under the smart home system, which is one of the steps to reach the smart city.

2.6 Design Constraints

- All devices must be constantly connected to WiFi.
- The mobile application will not run without the Internet.
- The system should operate in an ideal environment because the reflective surfaces affect the intensity of the light which negatively affects our smart project.

Chapter 3

3 System Design

3.1 Overview

This chapter discusses the conceptual description of the system, it shows the system requirement analysis, a block diagram of the system, structural diagram, flow chart, detailed design, schematic diagrams.

3.2 Detailed System Description

As shown in Figure 3.2, the system will utilize Light Dependent Resistor sensor (LDR) to continuously monitor and measure the amount of light intensity and PIR Motion Sensor to detect the motion. Then sensed data will be sent to the microcontroller to analyze it, in order to know when the light intensity is nearly unsuitable for eye comfort and if there is any movement in the house. In order to send this data through internet to the cloud, this data will be stored in a cloud, analyzed and give some suggestions to the homeowner, They can take actions, update tasks, and schedule the system. The system always guarantees that the light intensity is changing increasing or decreasing according to several factors, including eye comfort. The PIR sensor will detect the motion by sending the data to the microcontroller to determine if there is any motion in the home, thus directing the motors to make the right decision.

Furthermore, our system is an event driven system. So depending on the collected and analyzed data ,the system may be programmed to trigger some events, based either on some predefined conditions or a feedback.

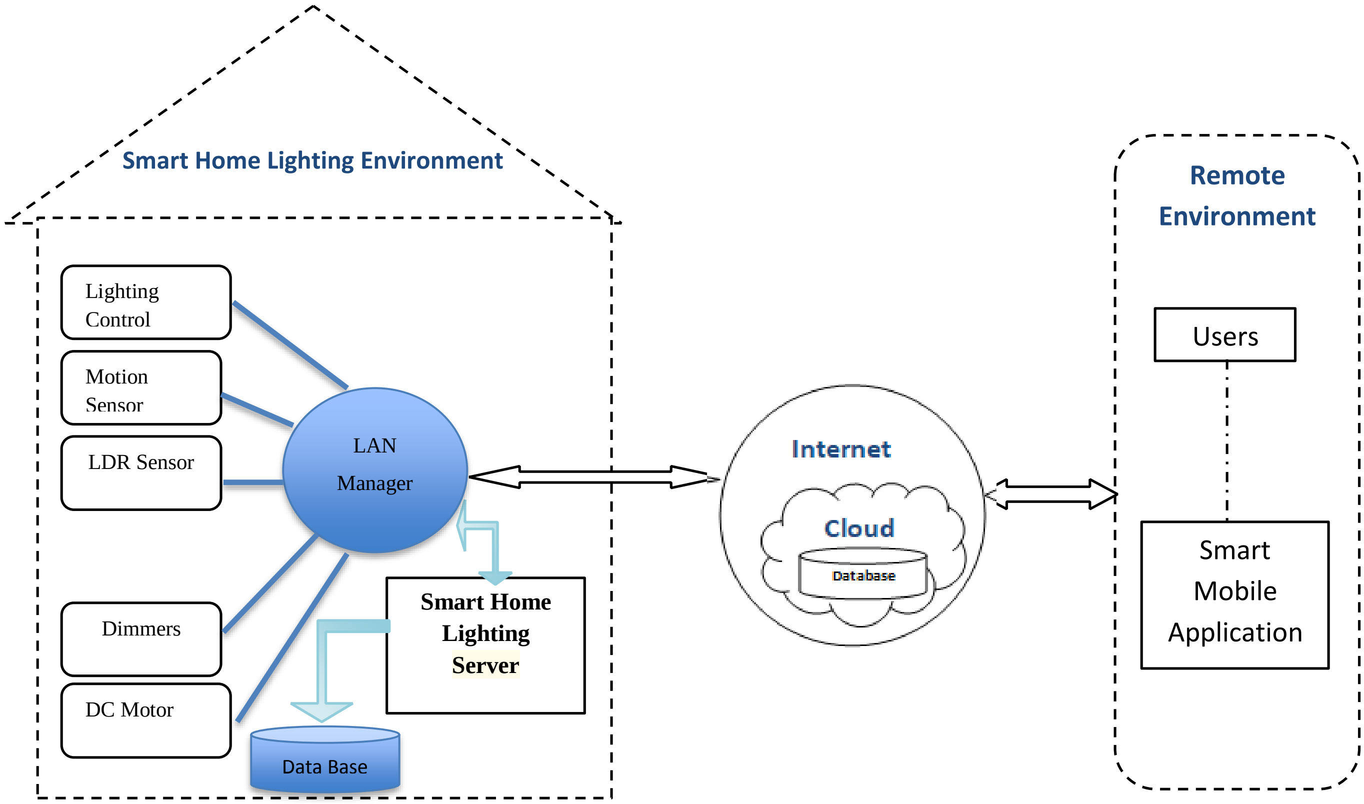


Figure 3.1: Graphical Representation of Smart Home Lighting System.

Figure 3.1 depicts the smart home lighting system main components and their inter-connectivity. On the left block, the smart home lighting environment, we can see the devices connected to a local area network [LAN]. This enables the communication among the devices and outside of it. Connected to the LAN is a server and its database. The server controls the devices, logs its activities, and executes the appropriate commands. The smart home server also transfers data to the cloud and remotely activate tasks in it using APIs, application programming interface processes. The connection to the internet allows the users to communicate with the s mart home lighting to get current information and remotely activate tasks.

3.3 Block Diagram

The functional block diagram of the proposed system is illustrated in Figure 3.2

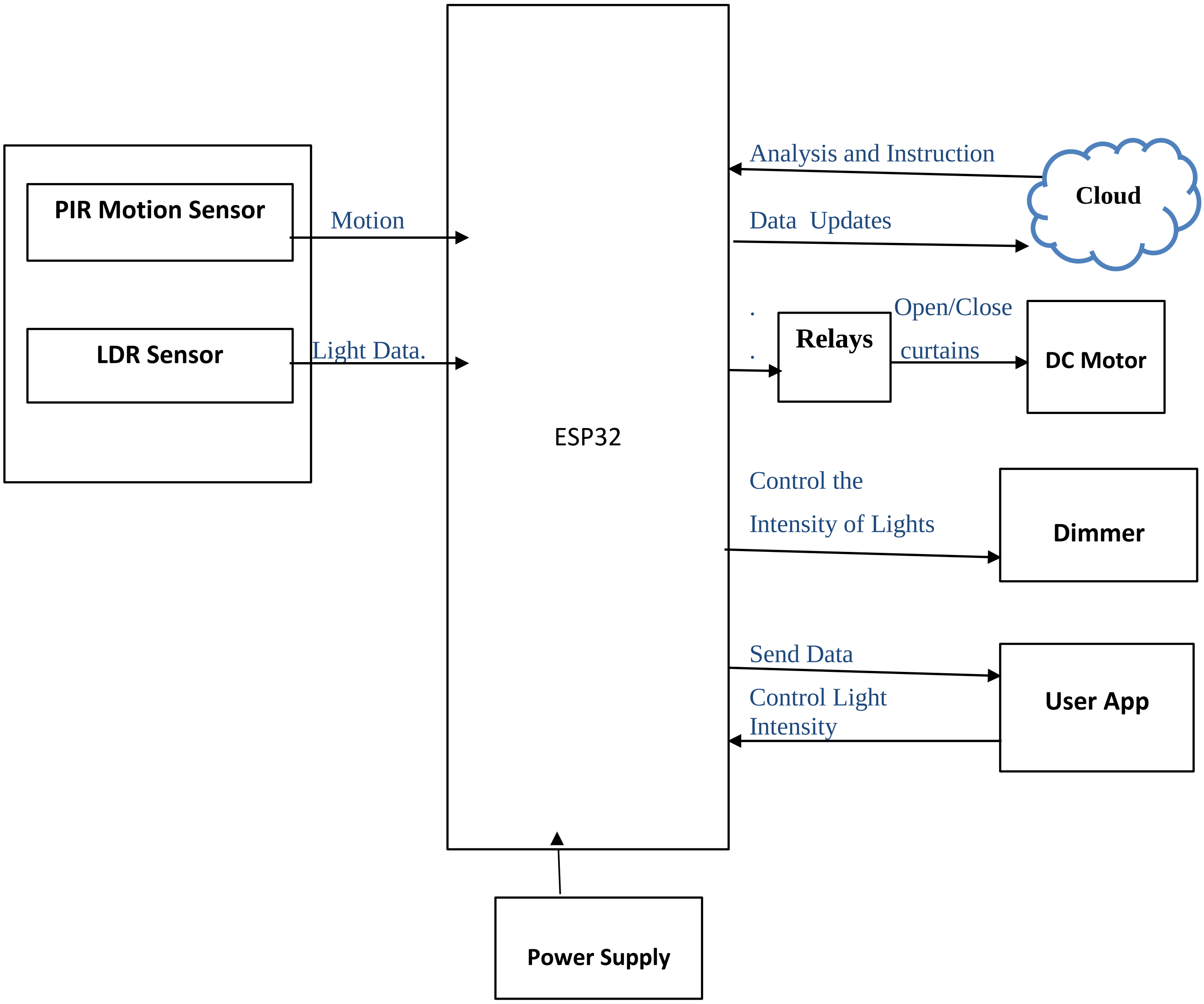


Figure 3.2: Functional Block Diagram

In order to achieve the described requirement in section 1.6, the following design will fulfill it.

3.4 Hardware System Design

3.4.1 Connect the PIR Motion to ESP32

PIR sensors allow the detection of motion based on the detection of infrared energy emitted by a moving body, meaning that we can use it to know when someone enters and leaves a certain room**.** For this project we need to wire a PIR motion sensor to the ESP32 board, The connection diagram between them is quite simple.[14]

Three pins need to be connected:

- PIR VCC to ESP32 dev board 5V
- PIR DATA to ESP32 GPIO through a 1K Resistor (D2 in this tutorial)
- PIR GND to ESP32 GDN

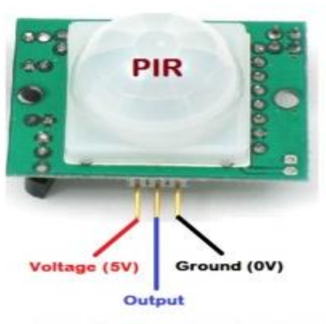


Figure 3.3: PIR Pins[14]

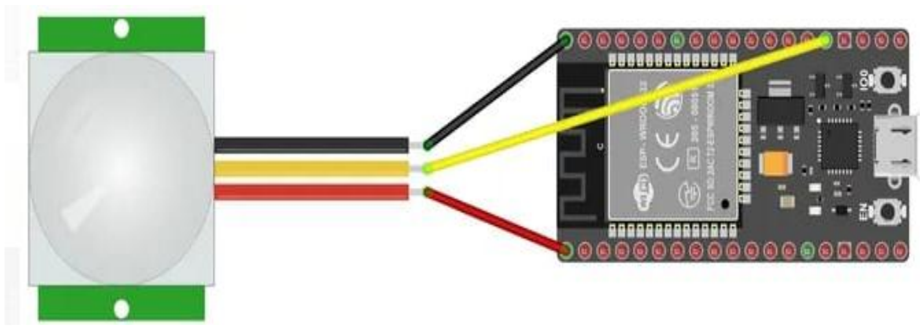


Figure 3.4: Connections Between ESP32 and PIR Motion Sensor

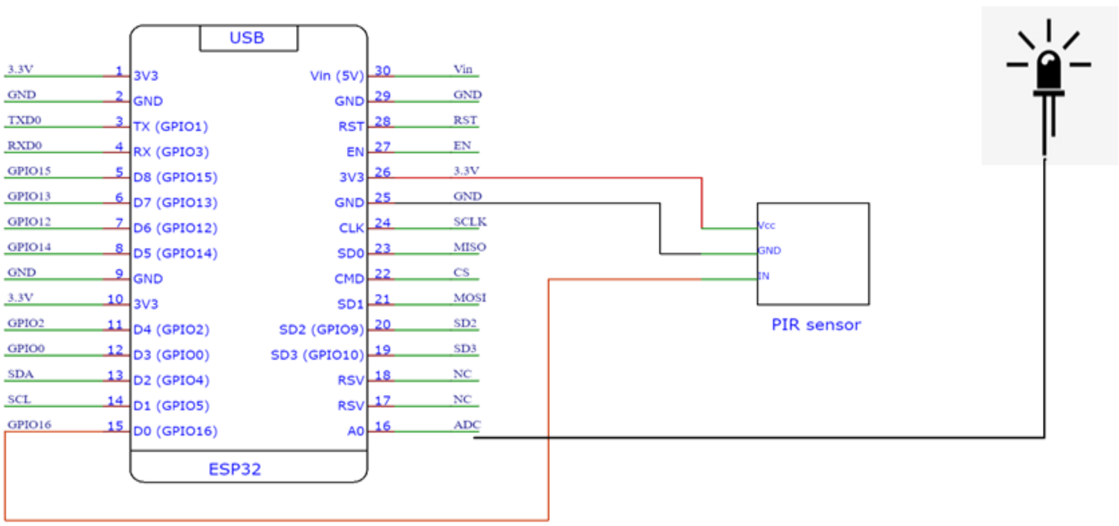


Figure 3.5: Schematic Diagram for ESP32 and PIR Motion Sensor

3.4.2 Connect the DC Motor to ESP32

The DC motor works with a high voltage that can burn ESP32 ⇒ We cannot connect the DC motor directly to ESP32 . We need a hardware device between the DC motor and ESP32 .

The device takes three responsibilities :

★ Protecting ESP32 from the high voltage

★ Receiving the signal from ESP32 to change the pole of the power supply to control the motor's direction .

★ Amplifying the PWM signal from ESP32 (current and voltage) to control the motor's speed.[15]

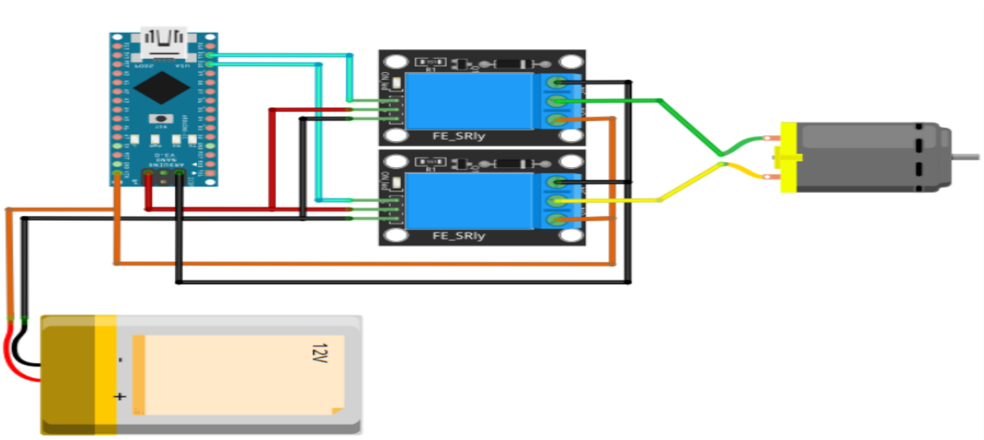


Figure 3.6: Connections Between ESP32 and DC Motor

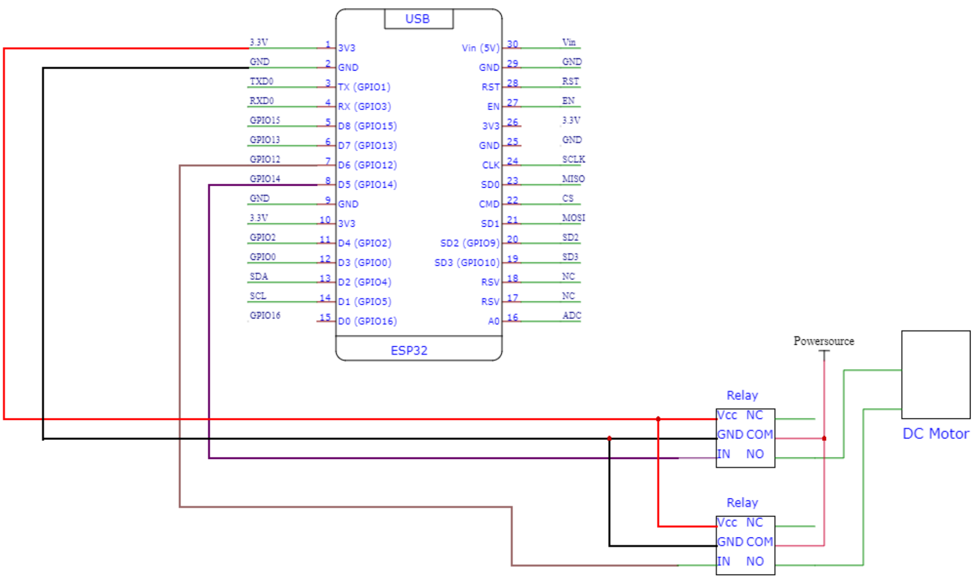


Figure 3.7: Schematic Diagram for ESP32 and DC Motor

3.4.3 Connect the Dimmer to ESP32

The module is connected to the mains via the AC-IN terminal block and the bulb is connected to the LOAD terminal block. On the electronic side, the pins are connected as follows:[16]

- ✓ Vcc at pin 5 or 3.3V of the microcontroller
- ✓ GND to ground GND of the microcontroller
- ✓ Z-C to pin 1 (GPIO possible 0,1,2,4,5,7,12,13,14,15,16,17,18,19,21,22,23,25,26,27,32, 33,34,35,36,39)
- ✓ PWM to pin2 (GPIO possible 0,1,2,3,4,5,12,13,14,15,16,17,18,19,21,22,23,25,26,27,32,33)

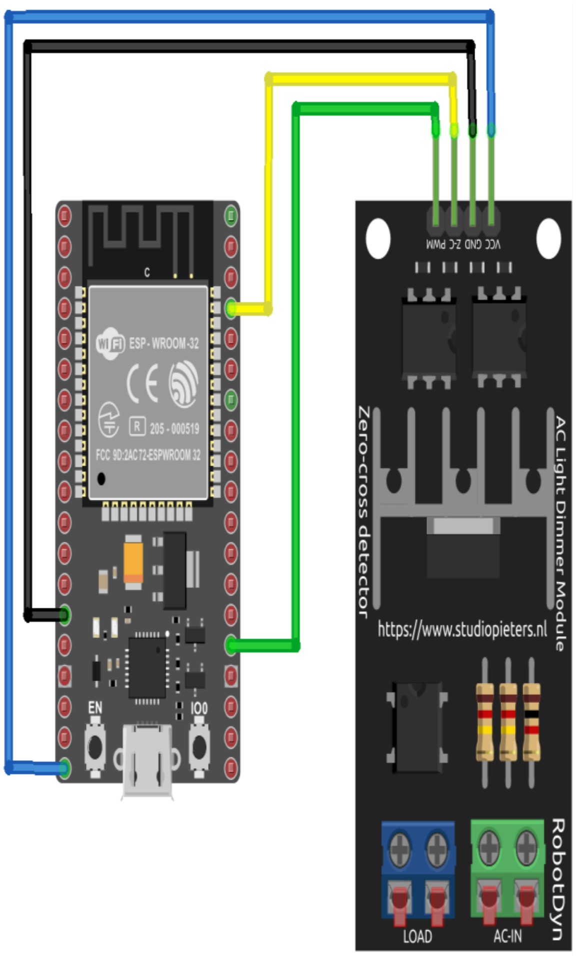


Figure 3.8: Connections Between ESP32 and Dimmer

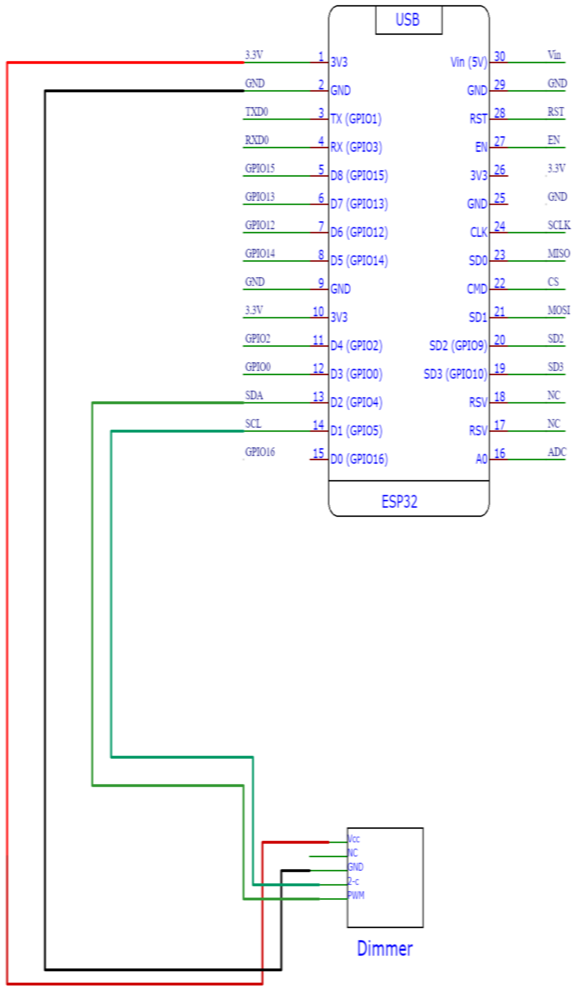


Figure 3.9: Schematic Diagram for ESP32 and Dimmer

3.4.4 Connect the LDR Sensor to ESP32

LDR stand for Light Dependent Resistor it means LDR is a register which resistance depend on falling light intensity on LDR. When light intensity increases then its resistance will decrease and vice versa.

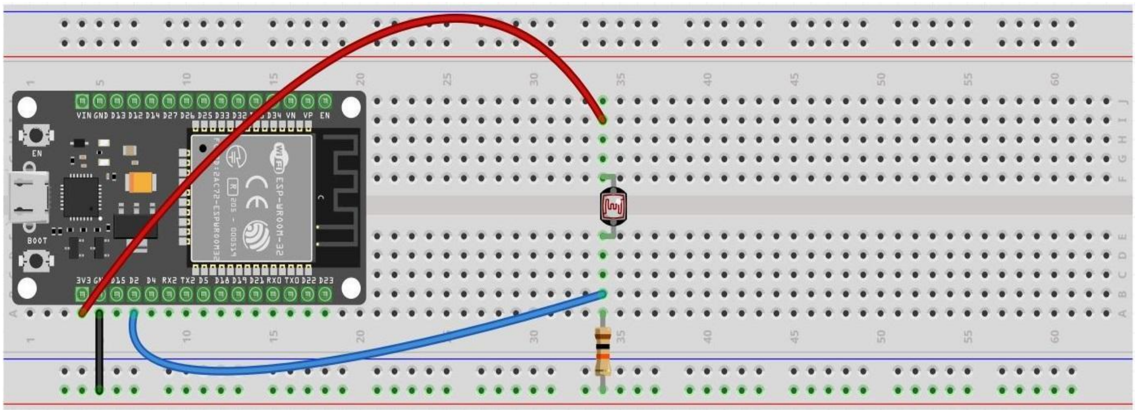


Figure 3.10: Connections Between ESP32 and LDR Sensor

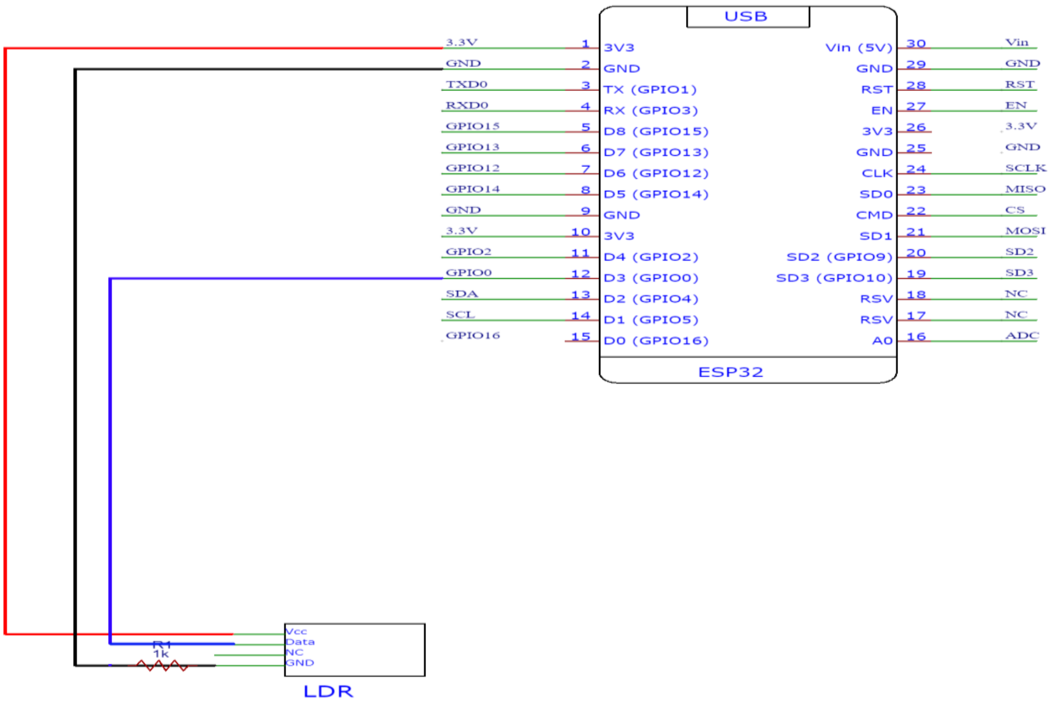


Figure 3.11: Schematic Diagram for ESP32 and LDR Sensor

We will using D2 pin as Analog Input and LDR connected with 10k resistor, which connected to ground and LDR connected to 3.3 volt , Joint point of Resistor and LDR is connected to D2 pin of ESP32 board.

The schematic diagram in figure 3.12 shows the connection of the whole system, such that it shows the connection between the ESP32 and the sensors, and also shows the connection between the devices and the ESP32.

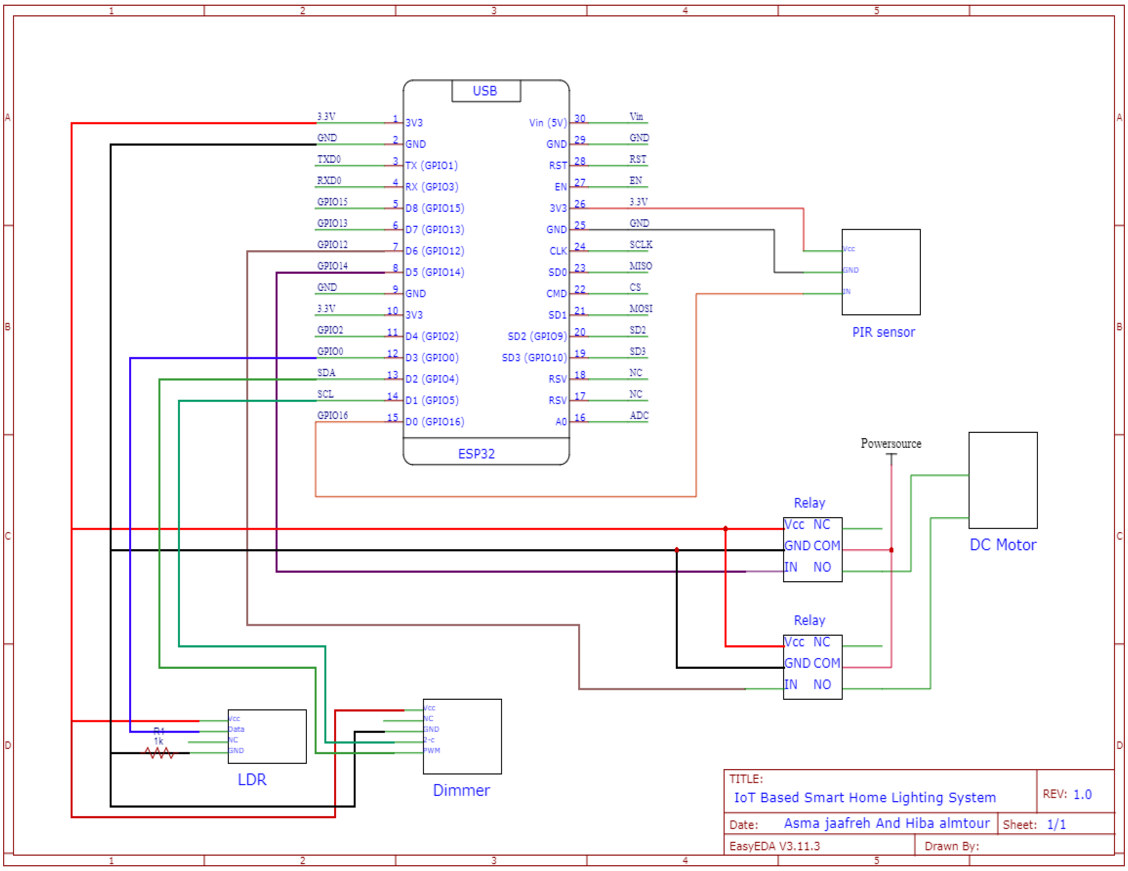


Figure 3.12: Schematic Diagram

3.5 Software System Design

The smart lighting System will be controlled by a mobile application. The sensors will send the readings to the server and store them on the database to be accessible by the mobile application, then display the readings on the mobile application to allow the users to show them as shown in figure 3.13. The mobile application also will control the curtains and the light intensity by storing the status of these devices on the database through the server then the microcontroller will get the status and apply it on the devices as shown in figure 3.14.

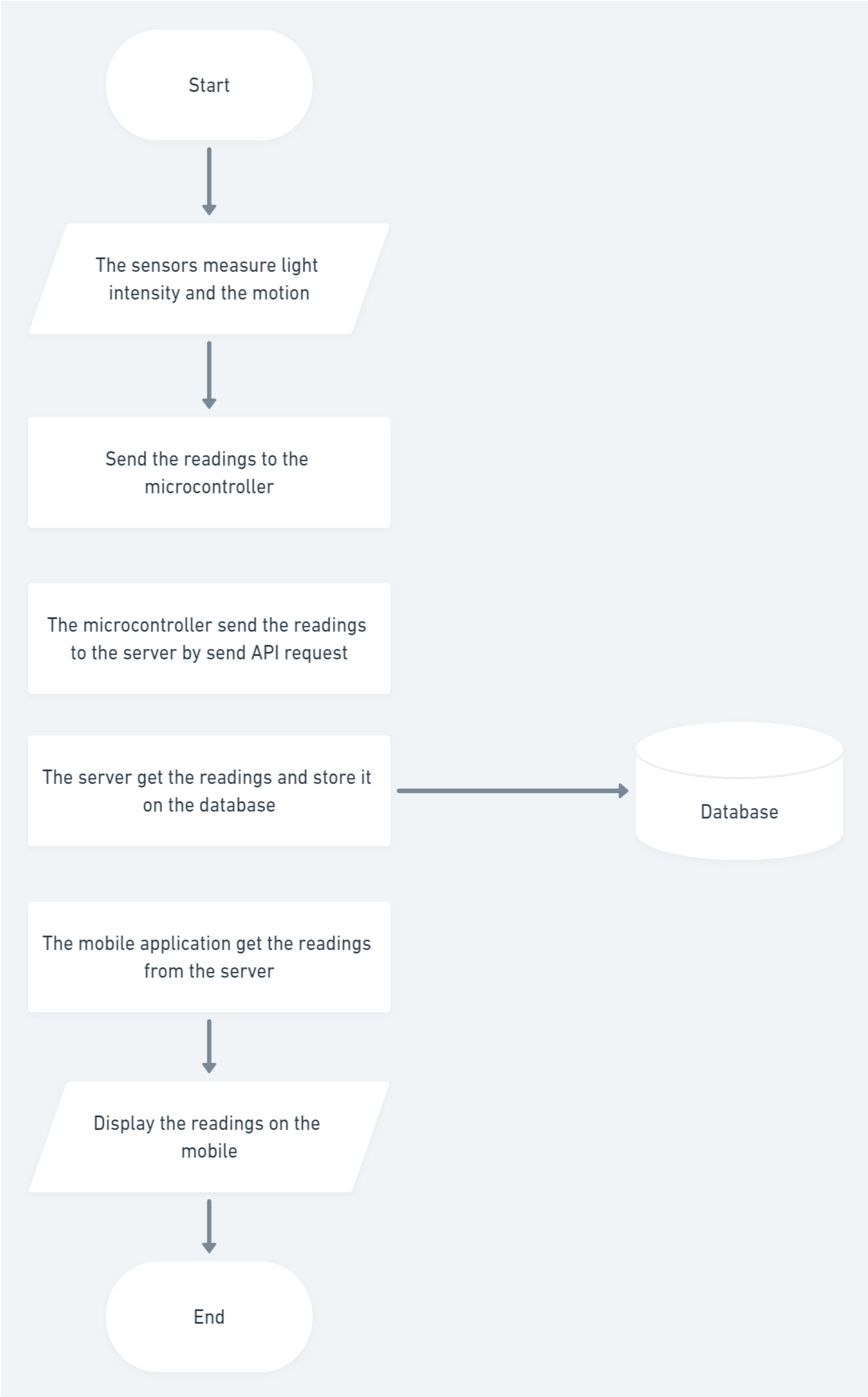


Figure 3.13: Flowchart for Send Data to the Mobile Application

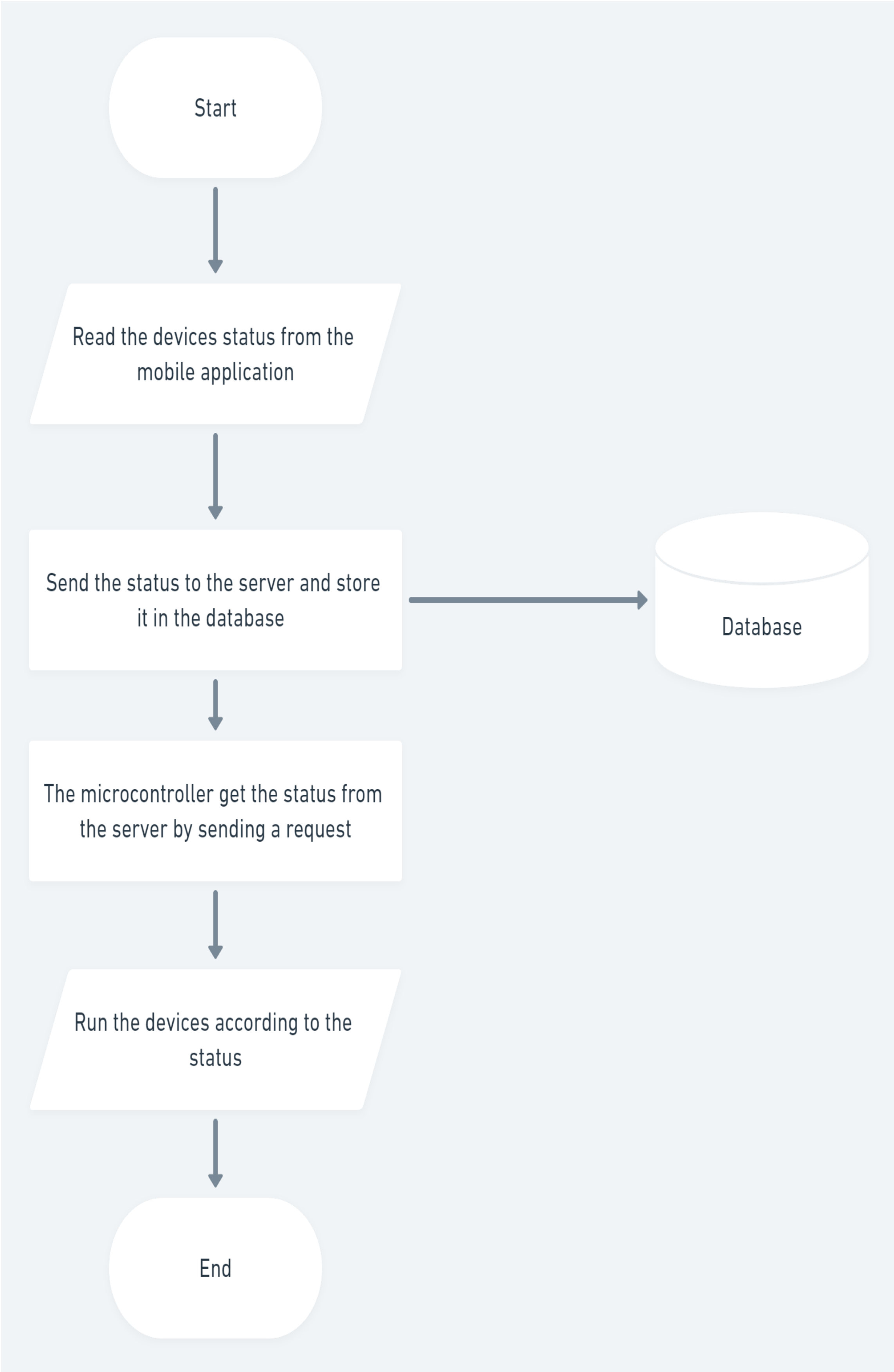


Figure 3.14: Flowchart for Send Data to Microcontroller

3.6 Sequence Diagram

A sequence diagram is a type of interaction diagram because it describes how and in what order a group of objects works together. This diagram is used by software developers and business professionals to understand the requirements for a new system or to document an existing process. Sequence diagrams are sometimes known as event diagrams or event scenarios. The sequence diagram of the proposed system shown in figure 3.15

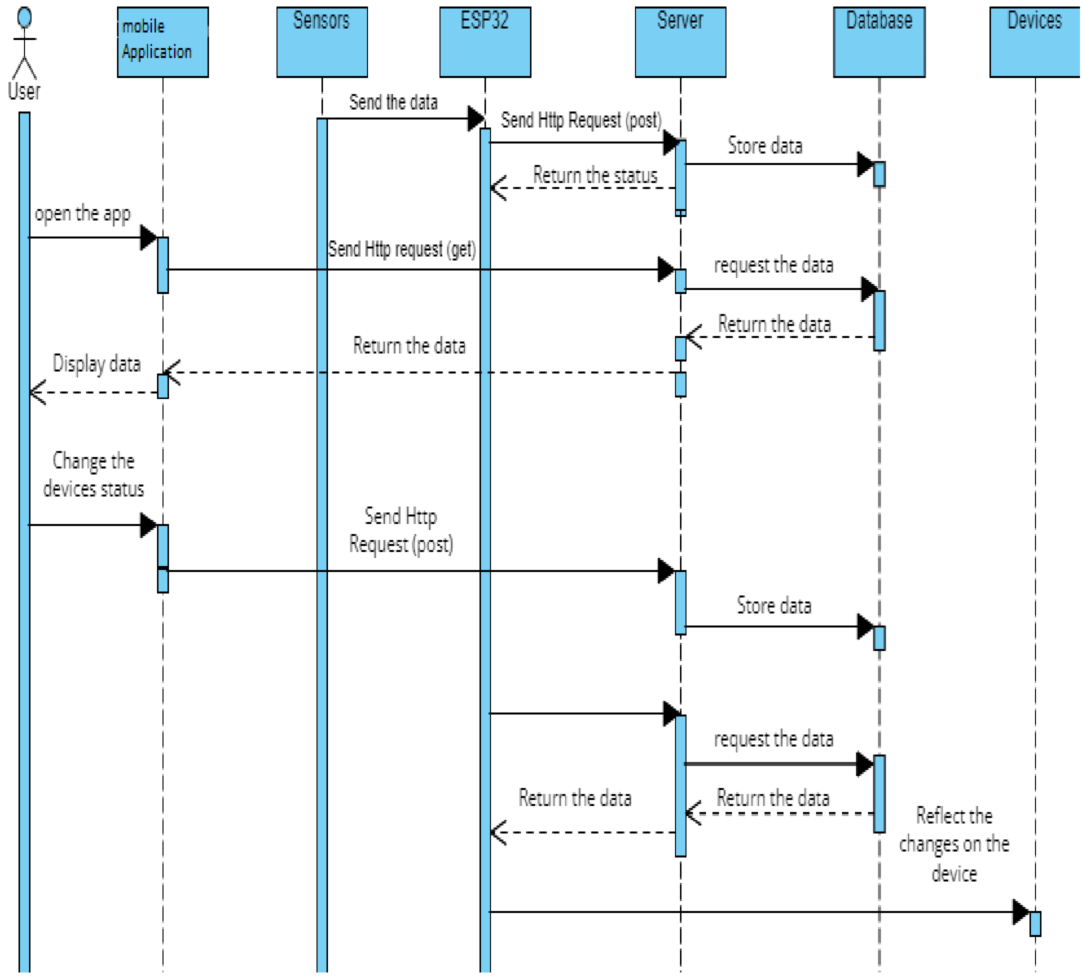


Figure 3.15: Sequence Diagram

3.7 Design Alternative

There are a lot of other strategies by which you can achieve the requirements described in section 1.6, such as :

- ✓ Smart Home controllers: You can control the intensity of lighting in your home depending on other smart devices you use in it, such as a smart door, using a Smartphone, Computer or Google assistant.
- ✓ Timer switches: They turn the lights on at a specific time and off at another time, and are also used as a safety element, as they may give a feeling that there is someone in the house while there is no one.

3.8 Conclusions and Summary

At the end of the project, we provide a suitable light intensity for eye comfort by achieving the requirements described in section 1.6 (we achieve the first and the second items, but the third item is for future work). Also we design and implement smart lighting systems using mobile applications as interfaces to provide the easiest environment for the users allowing them to monitor and control their homes lighting system remotely.

The smart lighting system is user friendly, improves efficiency and provides eye comfort, which reduces the possibility of exposure to some diseases such as (irritated eyes).

Chapter 4

4 Software

4.1 Overview

This chapter describes the implementations of the software that we used in this project, such as the IDEs used to build the project codes, and the user interface.

4.2 Software Implementation tools

This section will provide some information about the main programs and software technologies used in our project:

4.2.1 Arduino IDE: The open-source Arduino Software (IDE) makes it easy to write code and upload it to the board. This software can be used with any Arduino board and ESP microcontrollers [17]. In our project, we used Arduino IDE to writing code (each component owns part from code to get data and send to server) then make sure there are no errors to upload the program into ESP32. It's flexible and easy to use.

4.2.2 EasyEDA (PCB designer): Is an easier and powerful online PCB design tool that allows electronics engineers, educators, students, makers, and enthusiasts to design and share their projects [18]. This is a design tool integrated with the LCSC components catalog and PCB service that helps users to save time to make their ideas into real products. In our project, we used EasyEDA to design circuits then use this design to print it into PCB.

4.2.3 Ionic: is one of the most popular open source frameworks for developing hybrid mobile and tablet applications built by angular. We can build Android, iOS, Blackberry and other applications from various platforms. It was created by Drifty in 2013. Ionic offers many tools for developing hybrid applications for smartphones by using web technologies such as HTML5, CSS, SASS.[19]

4.2.4 NoSQL database is an easy-to-use database that supports many features such as Online Connection by uploading the database on a server that can be accessed via the Internet.

4.2.5 MongoDB is a source-available cross-platform document-oriented database program. Classified as a NoSQL database program, MongoDB uses JSON-like documents with optional schemas. We used it for build non-relational database (NoSQL), because it's better for big data.[20]

4.2.5 Visual Studio Code Is a lightweight but powerful source code editor which runs on computers and is available for Windows, macOS and Linux. It comes with built-in support for JavaScript, TypeScript and Node.js. We used VSC for build front-end and back-end web page that will be converted to phone application (APK extension).[21]

API stands for Application Programming Interface. We used it to link between the databases on the server and the mobile application, and it was programmed using the php language.

4.2.6 Heroku Is a container-based cloud Platform as a Service (PaaS). Developers use Heroku to deploy, manage, and scale modern apps. This platform is elegant, flexible, and easy to use, offering developers the simplest path to getting their apps to market . We used it for receiving a request form phone application and response action as we need.[22]

Heroku use MVC design pattern and separates presentation and interaction from the system data. The system is structured into three logical components that interact with each other. The Model component manages the system data and associated operations on that data. The View component defines and manages how the data is presented to the user. The Controller component manages user interaction (e.g., enter presses keypad, touch, etc.) as shown in the following figure.

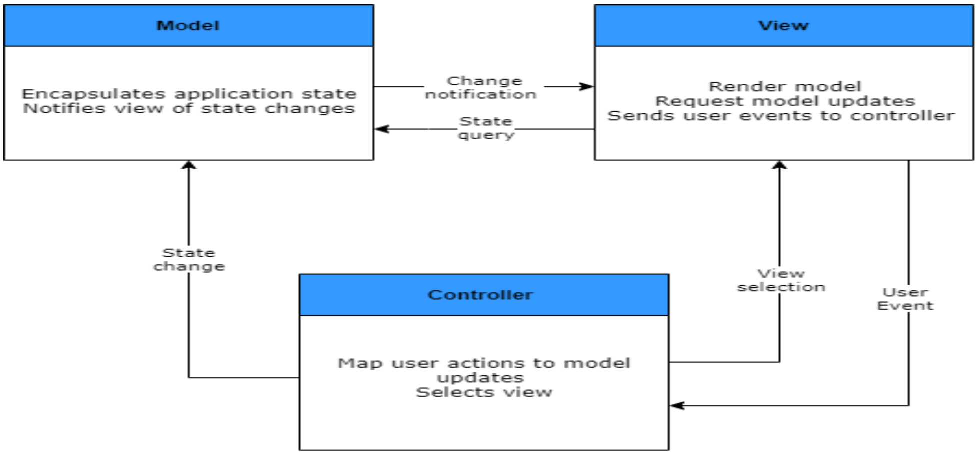


Figure 4.1: Model-View-Controller (MVC) Pattern

Advantages of MVC pattern:

- Allows the data to change independently of its representation and vice versa.
- Supports presentation of the same data in different ways with changes made in one representation shown in all of them.

4.2.7 Postman is a collaboration platform for API development. Postman's features simplify each step of building an API and streamline collaboration so you can create better APIs—faster. We used it to control the server that we built.

In general, this API is giving you the ability to send requests in Postman to connect to APIs you are working with. Your requests can retrieve, add, delete, and update data. Whether you are building or testing your own API, or integrating with a third-party API, you can try out your requests in Postman. Your requests can send parameters, authorization details, and anybody data you require. [23]

In the main we have 4 folders:

1. **User:**

   username: { type: String, unique: true, required: true, lowercase: true, trim: true },

   password: { type: String, required: true },

2. **Sensors**

   userId: { type : Schema.Types.ObjectId, ref: 'users',required : true },

   motion: {type : Boolean, required : true },

   LightIntensity: { type : Number, required : true },

3. **Setting**

   user: { type: Schema.Types.ObjectId, ref: 'users', required: true, index: true },

   autoLight: { type: Boolean, required: true, index: true }, autoWindow: { type: Boolean, required: true },

times: { type: { start: { type: Date, required: true },

end: { type: Date, required: true }, type: { type: Number, required: true },

value: { type: Number, required: true } }[], required: true }

4. Devices

userId: { type : Schema.Types.ObjectId, ref: 'users',required : true },

window: { type : Number, required : true },

light: { type : Number, required : true }, }

4.1.1 Used Programming Languages

- Arduino C/C++.

Figure 4.2: Logo Screen of Smart Light Application.

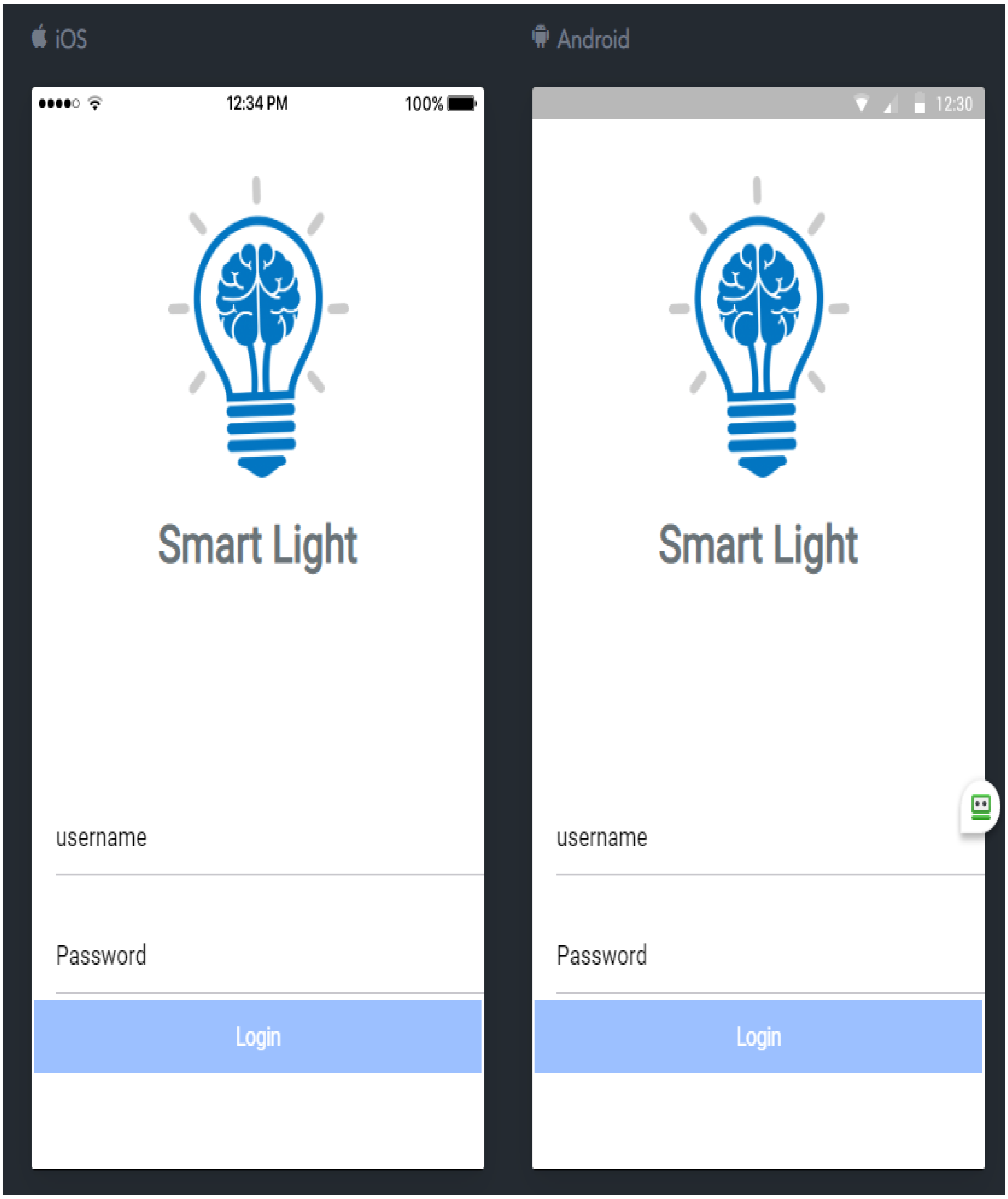


Figure 4.3: Main Screen of Smart Light Application

In general, this is the main screen. After the system detects that the username and password are correct, the system will allow the user to enter the Smart light application.

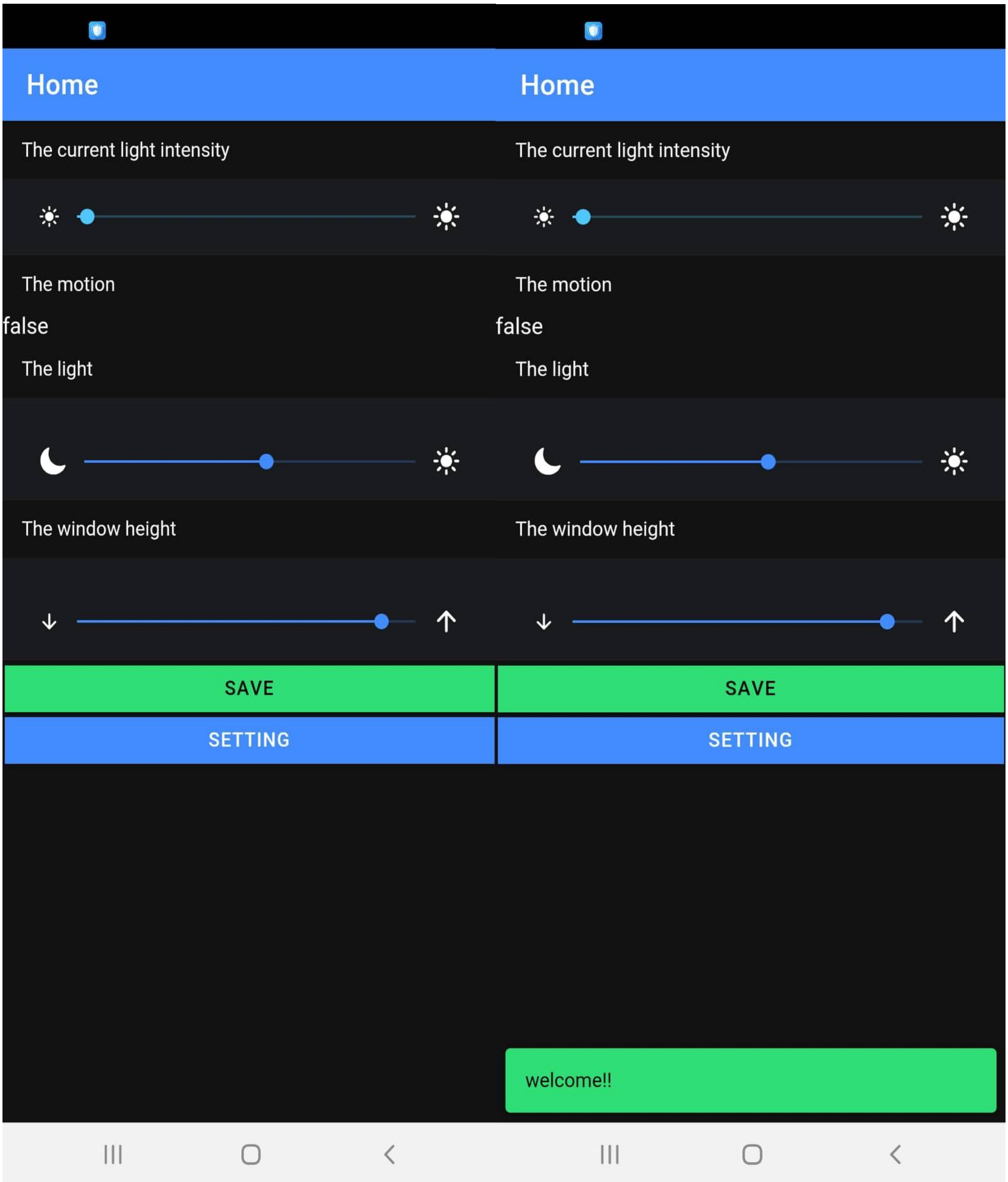


Figure 4.4: Home Screen of Smart Light Application(a)

After login, the Home interface will appear in which the two sensors (LDR sensor and PIR motion sensor) show their status. Also there is the light intensity and window height where the user can control them manually. We have two control buttons [Save and Setting], when the user clicks on Save, the data of the user will be saved and when he clicks on Setting, the Setting interface will appear.

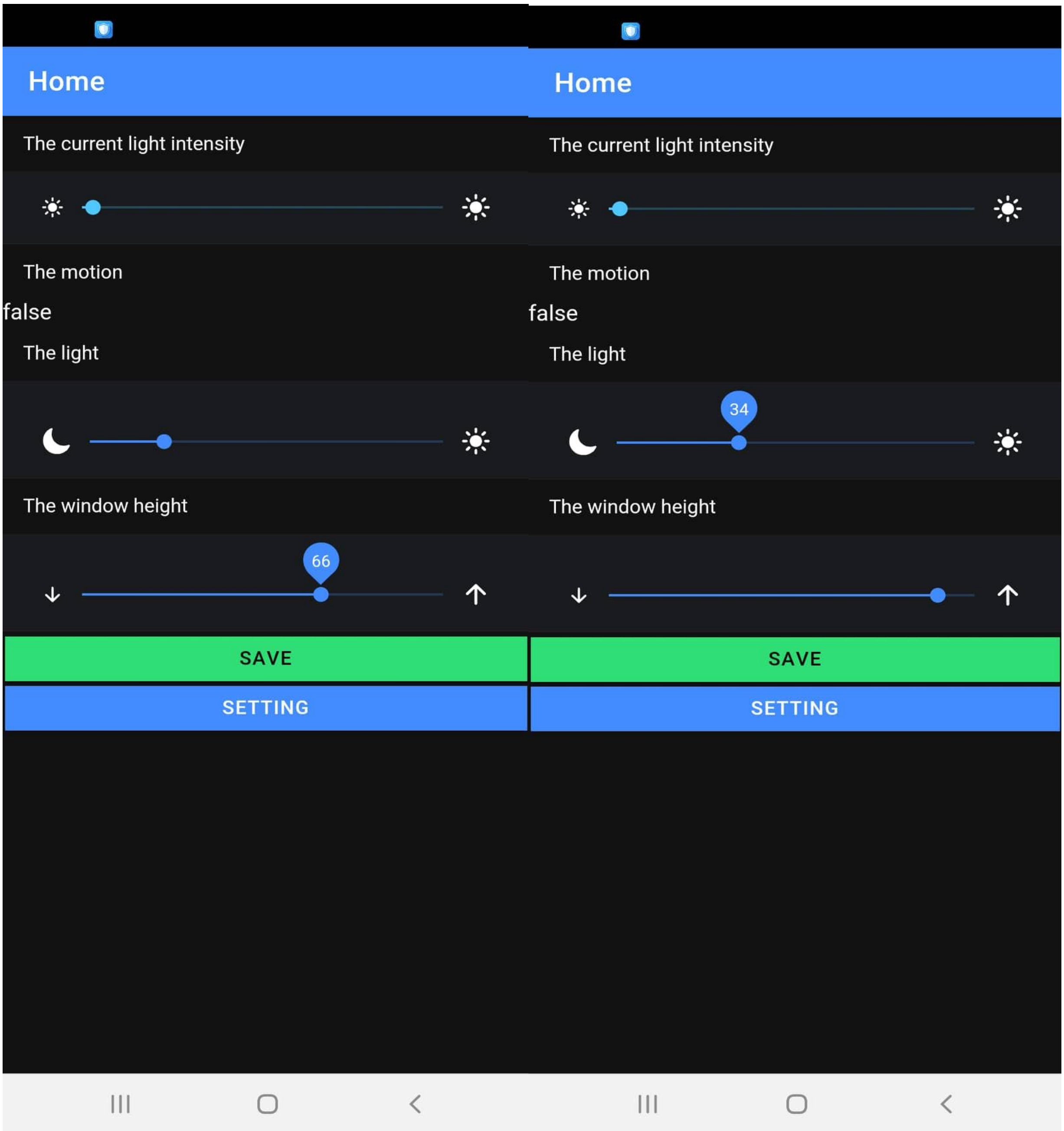


Figure 4.5: Home Screen of Smart Light Application (b)

These screens show that if the user wants to know the correct value of light intensity or window height, he just has to walk on the same reading line and the value will appear as shown in the figure 4.5

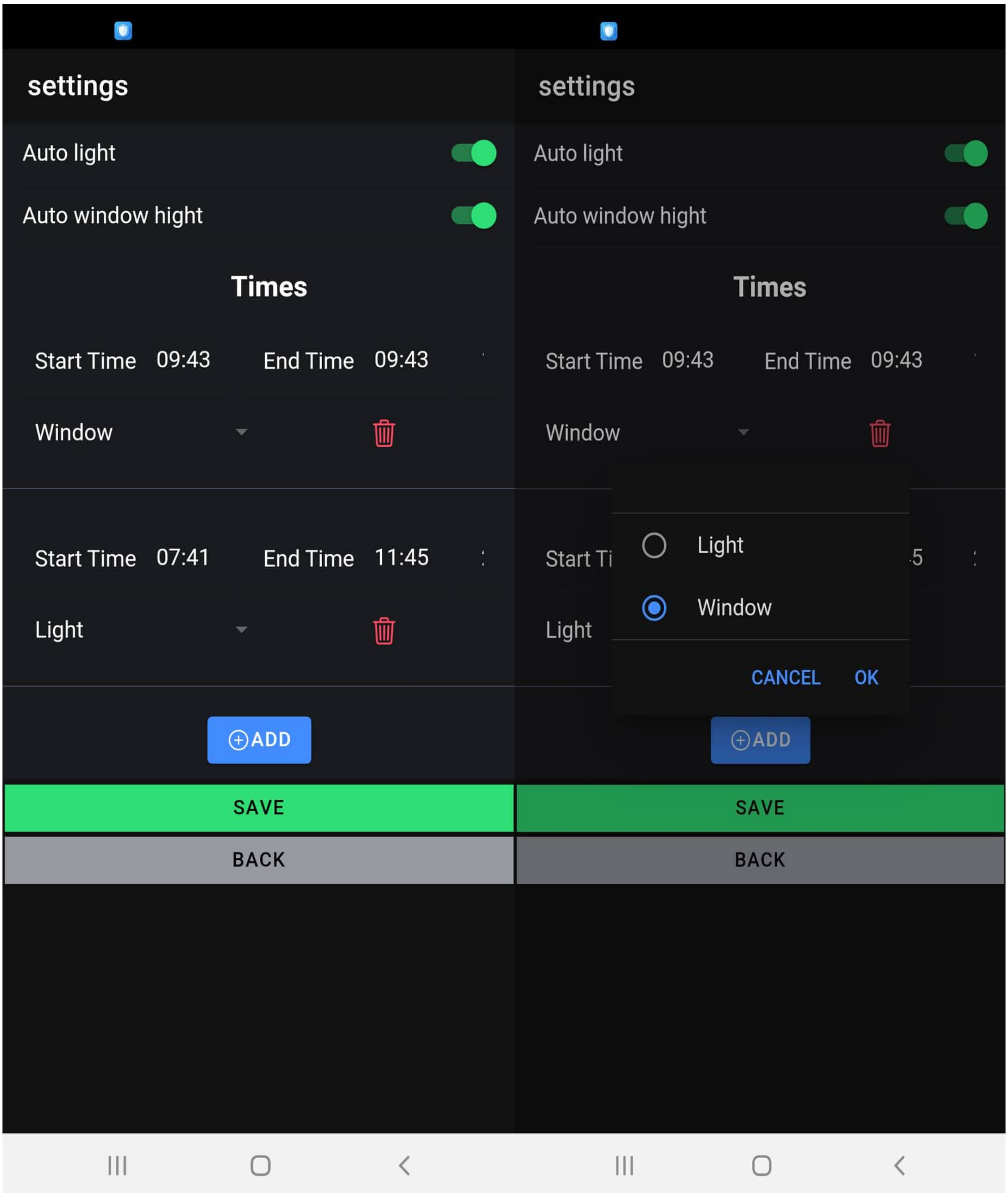


Figure 4.6: Settings Screen of Smart Light Application (a)

After clicking on the Setting button this interface will appear, where there are two switch buttons (Auto light and Auto window height) to take the reading automatically. Also we have an ADD button when the user clicks on it, he will be able to set a schedule. The Select box help the user to choose where he want the schedule for light or window and set the times (start and end time) in addition of the value to the light intensity or the height of the window. The BACK button When the user clicks on it, it will return to the previous interface.

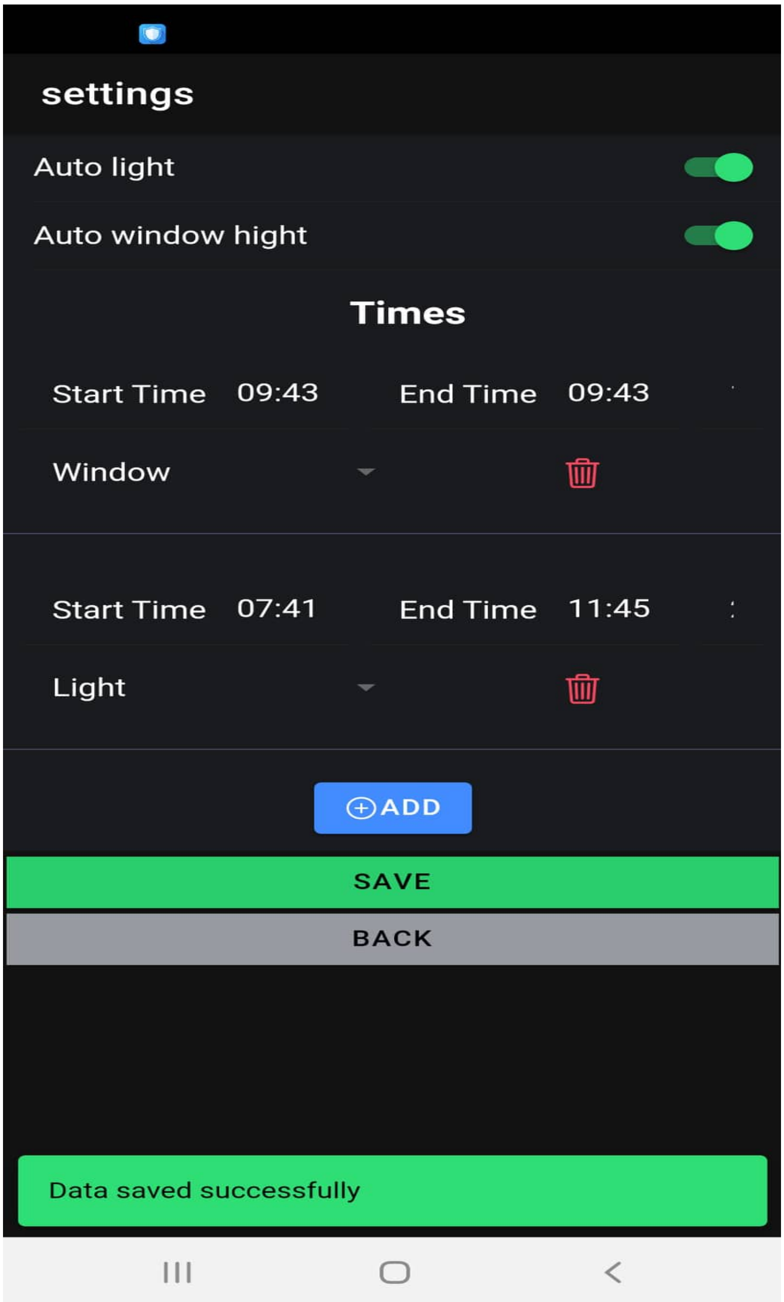


Figure 4.7: Settings Screen of Smart Light Application (b)

This screen shows that the application appears to the user that the data saved successfully when he clicks on the SAVE button.

Chapter 5

5 Validation and Testing

5.1 Overview

In this chapter, we will describe and discuss the steps, results of the implementation, and testing of the hardware and software parts.

5.2 Units testing

We start by testing all the parts to ensure that all of the functions work perfectly and without errors.

5.2.1 Hardware Testing

After completing the phase of connect of ESP32 pins with PIR motion sensor and LDR sensor, and make sure that microcontroller is working after we test that it can connect to Wi-Fi network, also be sure that all the components connected correctly with ESP32 pins and after we upload the arduino code to Esp32, the values (Light intensity ,the motion and Window height) should be correct as follows:

- **PIR motion sensor**: This sensor is working correctly and gives 2 reading for motion, as test cases below:

Condition	**Status**
There is motion	True
There isn't any motion	False

Table 5.1: Test Case for PIR Motion Sensor

- **LDR sensor**: This sensor requires some specific conditions to work correctly, so the environment in which the sensor is located must be ideal. Also the height of curtains affects this sensor reading.
  The following screens shows the testing of this sensor:

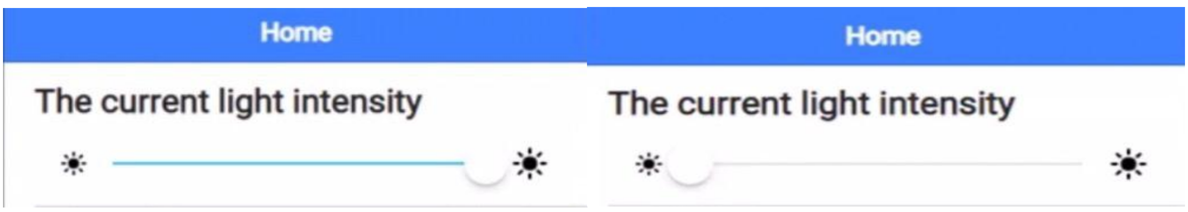


Figure 5.1: Screen of Test Case for LDR Sensor

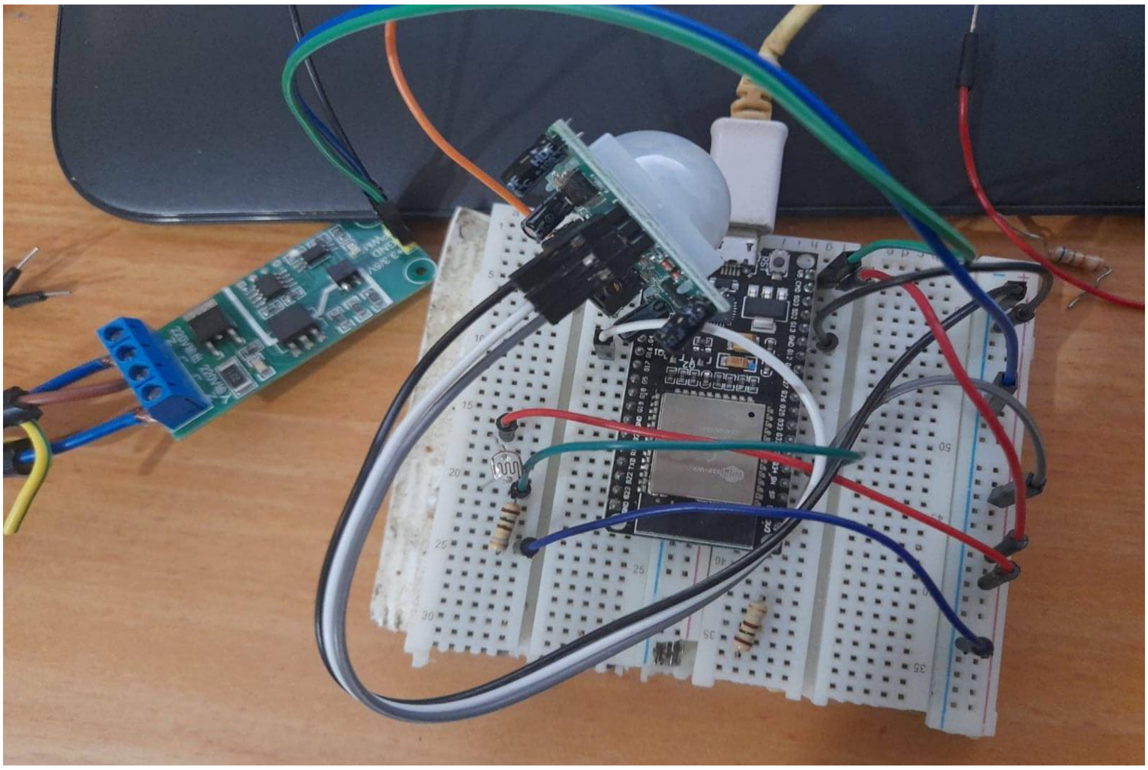


Figure 5.2: Connections Between ESP32 and LDR & PIR sensors

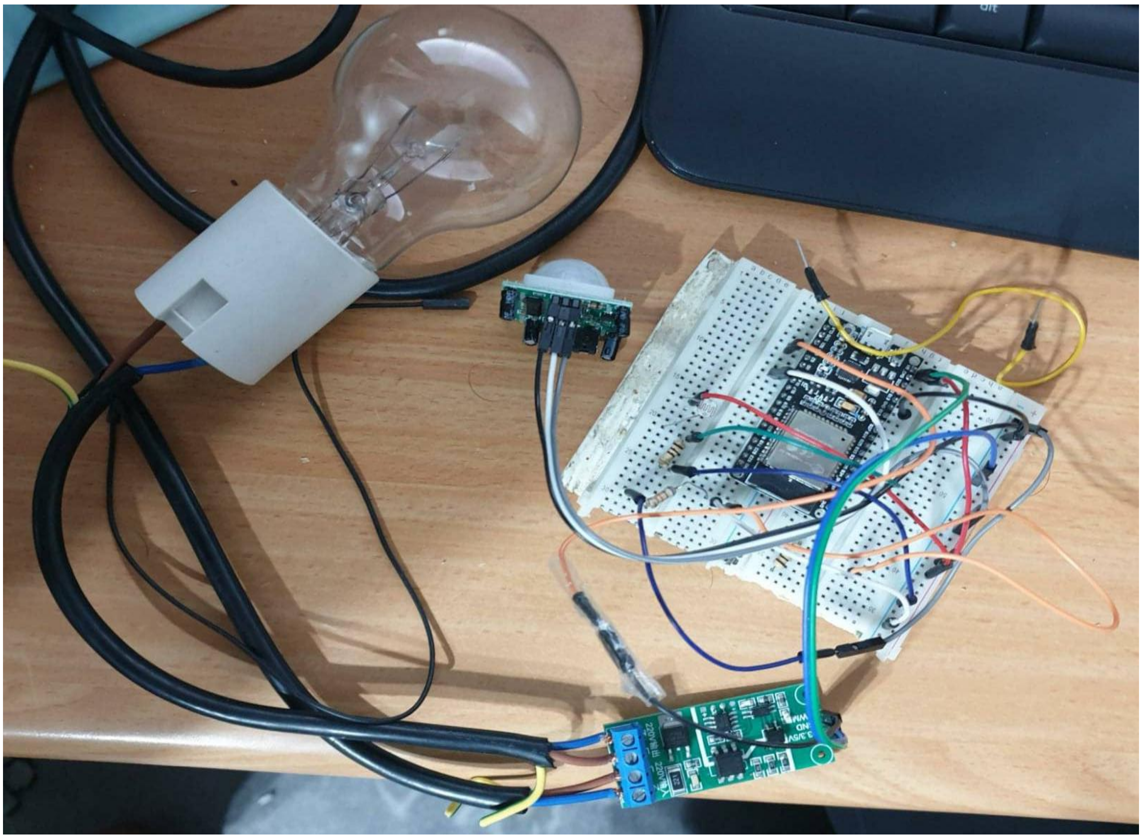


Figure 5.3: Connections Between ESP32 and Dimmer

Figure 5.4: Connections Between ESP32 and DC motor

5.2.2 Software Testing

In this section, we will describe the ways to test the software that we used to implement our system. The software includes the following:

5.2.2.1 Arduino Code Testing

We write code for each component to ensure that it works correctly (if the component is functional the result of the test is good, but if it is not functional the result of the test is defective), then all the component codes are collected in one file, each of them in one function.

5.2.2.2 Smart Application Testing (User interface)

The project needs from the user to login with his username and password, then if username or password error,the application shows him message as shown in Figure 5.2

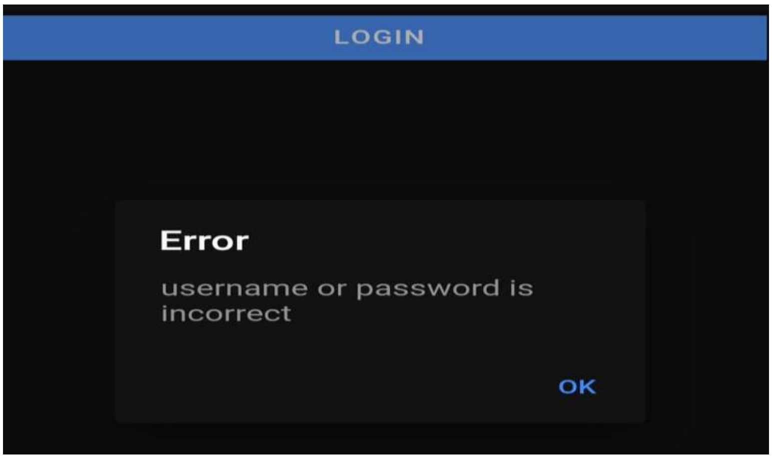


Figure 5.5: Screen of Error Login Case

5.3 Integration Testing

After ensuring that all the parts are working well, we started assembling and integrating the parts with each other to make the system ready to work.

5.4 Implementation issues and challenges

We faced several problems since the beginning of the project. One of these is the difficulty of connecting the hardware and software together and combining them in one system, and the electronic component is defective during hardware testing, like what happened with us while we were testing Dimmer. There are other problems related to poor internet, when the WiFi is weak signal, the connections between microcontroller and server are not responsive or not real time response. Also The system should operate in an ideal environment because the reflective surfaces affect the intensity of the light which negatively affects our smart project, and this environment is hard to obtain.

On the other hand, some of the electronic parts (hardware) were not available like (Dimmer) and needed time to be treated and delivered.

Programming is another challenge, such as learning programming languages and application environments.

Chapter 6

6 conclusion

6.1 Overview

In this chapter, we will conclude the challenges, final result and future work of our project.

6.2 Summary

At the end of this project, we are very glad to say it achieved all its objectives. We designed and built a smart light application which aims to provide suitable light intensity for eye comfort, the smart app provides an easy environment for users to make multiple decisions. Our smart app is flexible since it gives the user the choice whether he prefers set schedules or controls the light manually using the app. The smart system provides a suitable light intensity for eye comfort by achieving the requirements described in section 1.6.2 (For functional requirements, we achieve the first and the second items, but the third item is for future work). Also we design and implement smart lighting system using mobile applications as interfaces to provide the easiest environment for the users allowing them to monitor and control their homes lighting system remotely.

6.3 Future work

We suggest to expand our project, for examples:

- Expand it to be a product for sale in the electronic markets.
- Support more versions of components and new components.
- Control other resources (like TV light recourse.)

References

[1] Title: "Smart Lighting", last access to web page at Oct 2021, Available at: "*https://www.iotforall.com/use-case/smart-lighting*".

[2] Title: "right lighting conditions for the eye.", last access to web page at Oct 2021, Available at: "أهمية الإضاءة لراحة العيون!! | البوابة *(albawaba.com)*".

[3] Title: "Blue light", last access to web page at Oct 2021, Available at: "*https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6288536/*".

[4] Title: "IoT and Cloud computing overview", last access to web page at Oct 2021, Available at: " *https://www.intechopen.com/chapters/65877* ".

[5] Title: "IoT Smart Home Controllers", last access to web page at November 2021, Available at: "*http://scholar.ppu.edu/bitstream/handle/123456789/2423/graduation%20project%20document.pdf?sequence=1&isAllowed=y* ".

[6] Title: "IoT Smart Home Lighting System using Arduino UNO and ESP8266", last access to web page at December 2021, Available at: "*https://www.techrxiv.org/articles/preprint/IoT_Smart_Home_Lighting_System_using_Arduino_UNO_and_ESP8266/17161949/1/files/31734632.pdf* ".

[7] Title: "Smart Lighting System using Wireless Sensor Actuator Network", last access to web page at December 2021, Available at: "*https://www.researchgate.net/publication/324021855_A_smart_lighting_system_using_wireless_sensor_actuator_network*".

[8] Title: "Light Dependent Resistor sensor(LDR)", last access to web page at November 2021, Available at: " *https://www.indiamart.com/proddetail/ldr-light-dependent-resistor-sensor-ldr-18812839691.html* ".

[9] Title: "Esp32 vs raspberry pi", last access to web page at November 2021, Available At: " *Esp32 vs raspberry pi | All About Circuits*".

[10] Title: "ESP32 vs ESP8266", last access to web page at November 2021, Available At: " *https://www.electronicshub.org/esp32-vs-esp8266/* ".

[11] Title: "Stepper Motors vs. DC Motors", last access to web page at May 2022, Available At: "*bThe brushless DC motor only has one moving component – the rotor57c8a65bc348b68.png (549×482) (makeblock.com)*".

[12]Title: "Dimmer" , last access to web page at November 2021, Available at: "*https://upgradedhome.com/what-are-the-different-types-of-dimmer-switches/*".

[13]Title: "Relay" , last access to web page at November 2021, Available at: "*Relay - Wikipedia*".

[14]Title: "PIR Motion" , last access to web page at November 2021, Available at:"https://www.sunrom.com/get/966300 ".

[15]Title: "DC with ESP32" , last access to web page at May 2022, Available at: "*ESP32 - DC Motor | ESP32 Tutorial (esp32io.com)*"

[16]Title: "Dimmer with ESP32" , last access to web page at December 2021, Available at: "*https://www.aranacorp.com/en/using-an-ac-dimmer-with-esp32/*"

[17]Title: "Arduino IDE" , last access to web page at May 2022, Available at: "*https://www.arduino.cc/en/software*" .

[18]Title: "EasyEDA" , last access to web page at May 2022, Available at: "*EasyEDA - Wikipedia*"

[19]Title: "Ionic" , last access to web page at November 2021, Available at: "*https://exportdeveloper.com/tutorial/detail/58/Ionic-dependencies*" .

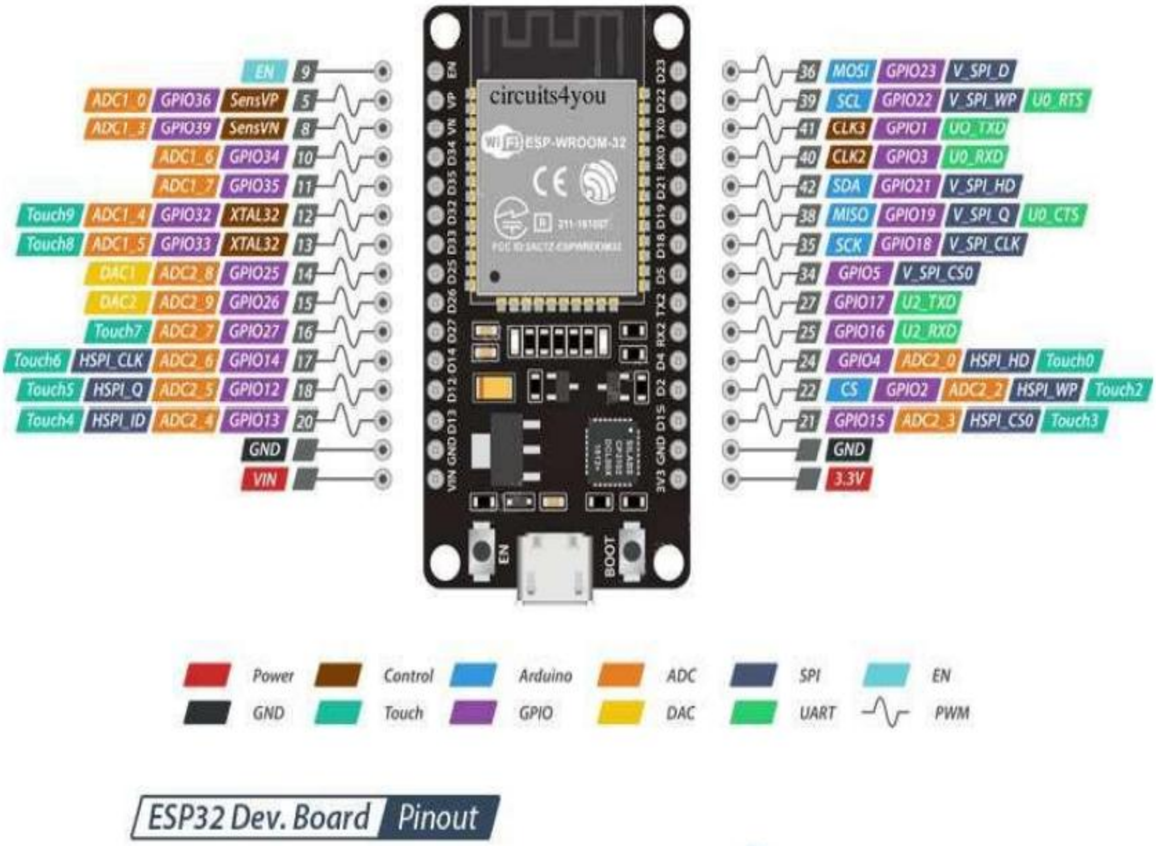
[20]Title: "mongodb" , last access to web page at May 2022, Available at: "*https://www.mongodb.com/what-is-mongodb*"

[21]Title: "visualstudio" , last access to web page at May 2022, Available at: "*https://code.visualstudio.com/docs*"

[22]Title: “heroku” , last access to web page at May 2022, Available at*:*
“https://www.heroku.com/about#:~:text=Heroku%20is%20a%20container%2Dbased,getting%20their%20apps%20to%20market ”

[23]Title: “ Postman ” , last access to web page at May 2022, Available at*:*
“ https://www.postman.com ”

Appendices



ESP32 Pinout