IE2090 - Professional Engineering Practice and Industrial Management

2^{nd} Year, 2^{nd} Semester

Final Report

Smart Plant Caring System

Submitted to

Sri Lanka Institute of Information Technology

In partial fulfillment of the requirements for the

Bachelor of Science Special Honors Degree in Information Technology

18/06/2024

Declaration

We declare that the this project report or part of it was not a copy of a document done by any organization, university any other institute or a previous student project group at SLIIT and was not copied from the Internet or other sources.

Project Details

Project Title	Smart Plant Caring System
Project ID	PEP_19

Group Members

Name
Lakshan T.R
Rangana W.P.M
Devinda M.C.G
Rajapaksha K.V

Abstract

In the face of urban space constraints, our project introduces a smart plant caring system designed for the modern urban dweller. This innovative system seamlessly integrates into the limited kitchen spaces, providing a convenient and efficient way to cultivate fresh garnish herbs. With its automated care features, users can enjoy the benefits of gardening without the extensive time commitment typically required. Preliminary feedback from users has highlighted the system's ease of use and the freshness of the produce as significant advantages. Overall, our smart plant caring system offers a sustainable solution to urban gardening challenges, promoting healthier lifestyles through access to fresh ingredients.

Acknowledgement

The completion of this project could not have been accomplished without the support and guidance of many individuals. The lecturer in charge of the module Narmada Gamage Madam and other assistant lecturers for their kind guidance and encouragement on the project. The videos uploaded to the courseweb enhanced our knowledge as to how the research should be carried out and how the final submission should be done. A special thank you is also directed to the lecturers who always made time to respond to our inquiries through email.

Furthermore, we express our gratitude to our colleagues who supported us in giving us their suggestions and recommendations to improve the report. The assignment would not have been completed without the effort and cooperation received from all the group members. We also want to express our gratitude to our parents for their love, support, and inspiration, which have given us the willpower to finish this project

Table of Contents

Declaration ii
Abstract iii
Acknowledgement iv
Table of Contents v
List of Figures vi
List of Tables vi
List of Acronyms and Abbreviations vi
1. Introduction 1
1.1 Problem Statement 1
1.2 Product Scope 1
1.3 Project Report Structure 2
2. Methodology 3
2.1 Requirements and Analysis 3
2.2 Design 8
2.3 Implementation 12
2.4 Testing 17
3. Conclusion 20
4. References 22

List of Figures

Figure 2-1 Software Interface 5
Figure 2-2 Design Diagram 8
Figure 2-3 High Level Architecture Diagram 9
Figure 2-4 Use Case Diagram 9
Figure 2-5 Activity Diagram for Watering System 10
Figure 2-6 Activity Diagram for Soil Moisture check 10
Figure 2-7 Activity Diagram for Monitoring Application 11
Figure 2-8 Activity Diagram for Light Sysytem 11
Figure 2-9 Activity Diagram for Display System 11
Figure 2-10 Soil moisture Senssing 12
Figure 2-11 Pump Control 12
Figure 2-12 compare with the threshold value 12
Figure 2-13 Light Timer implementation 13
Figure 2-14 Display Sensor Reading 13
Figure 2-15 Display details in LCD Display 13
Figure 2-16 Mobile App Interface 01 14
Figure 2-17 Mobile App Interface 02 15
Figure 2-18 Desktop view 15
Figure 2-19 Physical Structure 16

List of Tables

Table 1 Testing Table 17

List of Acronyms and Abbreviations

LED - Light Emitting Diode

IoT – Internet Of Things

LCD – Liquid Crystal Display

IDE – Integrated Development Environment

1. Introduction

1.1 Problem Statement

Urban living often restricts individuals from engaging in traditional gardening due to limited space and busy lifestyles. This situation deprives people of the mental relaxation that comes with nurturing plants and the practical benefits of having fresh garnished herbs readily available. To address these issues, we introduce the Plant Caring System project. This system aims to facilitate urban customers in effortlessly cultivating garnish herbs in their kitchens, thereby providing both mental relaxation and access to fresh plant ingredients

1.2 Product Scope

The Plant Caring System is designed with the objective of enabling urban residents to cultivate herbs in limited spaces, such as their kitchens, without requiring significant time and effort. The system automates essential plant care tasks, ensuring optimal growth conditions for the herbs. The project is scheduled for completion within a three-month timeframe and has an estimated budget of Rs 12000.00. By enhancing operational efficiency, promoting environmental responsibility, and improving customer satisfaction through automated plant care solutions, the system offers numerous benefits.

1.3 Project Report Structure

This report is structured into five chapters, each providing comprehensive details about the Plant Caring System project. We have tried to present the information in an understandable manner for the reader.

- **Chapter 1: Introduction** - This chapter provides a basic introduction to the project, including the problem statement and product scope.
- **Chapter 2: Methodology** - This chapter is divided into four subsections: Requirements and Analysis, Design, Implementation, and Testing, describing the project development process in detail.
- **Chapter 3: Project Evaluation -** This chapter evaluates the project under three sub-headings: Assessment of Project Results, Lessons Learned, and Future Work, providing insights into the project's outcomes and future direction.
- **Chapter 4: Conclusion** - This chapter summarizes the key points of the project.
- **Chapter 5: References** - This chapter lists all the references used in the preparation of the report.

The report is organized chronologically to ensure that readers can follow the development and evaluation process of the Plant Caring System project easily.

2. Methodology

2.1 Requirements and Analysis

2.1.1 Functional requirements

- **User Interfaces**

By analyzing requirements able to identify mobile and web interfaces of the system and the display of the system as the user interfaces.

Using the mobile app, watering schedules can be set based on real-time soil moisture data. The app also manages grow lights for plants. You can also see the current temperature and humidity. Notification Ping when the pump and light system is running. The web interface shows important information such as the soil moisture levels (as a percentage), temperature and humidity. A physical show is also used to display all the necessary information about plants.

- **Hardware Interfaces**

1. **NodeMCU ESP8266**: This is the main controller of our system. This board is used for WiFi connectivity. It can be programmed using the Arduino IDE and can communicate with the Arduino board via serial communication.
2. **Soil Moisture Sensor**: This sensor measures the moisture level in the soil. It usually has a digital output (wet/dry) and an analogue output that gives a proportional response. It can be connected to an analogue input pin on the Arduino.
3. **DHT11 Temperature and Humidity Sensor**: This sensor can measure ambient temperature and humidity. It uses a single-wire interface to communicate with the Arduino.
4. **Water Pump**: This is controlled by the Arduino based on the soil moisture level. You can use a relay module to control the pump, as the Arduino cannot drive the pump directly due to power constraints.
5. **16x2 LCD Display**: This display will show the temperature, humidity, and soil moisture level. It communicates with the Arduino via either a parallel interface or an I2C interface, depending on the model.

The logical and physical characteristics of each interface are as follows:

• **Arduino - NodeMCU ESP8266**: They communicate via serial communication. Physically, the TX/RX pins of both boards are connected (TX to RX and RX to TX).

• **Arduino - Soil Moisture Sensor**: The sensor is connected to an analogue input pin on the Arduino. The sensor's output voltage changes proportionally with the moisture level, which the Arduino reads as an analogue value.

• **Arduino - DHT11 Sensor**: The sensor communicates with the Arduino via a single-wire interface. The sensor sends a data packet every 2 seconds, which the Arduino reads.

• **Arduino - Water Pump**: The pump is controlled by a relay module, which acts as a switch. The Arduino controls the state of the relay (ON/OFF) based on the soil moisture level.

• **Arduino - LCD Display**: The display communicates with the Arduino via a parallel or I2C interface. The Arduino sends characters to be displayed to the LCD, which then renders the characters.

- **Software Interfaces**

To build this Software Interface, we use:

• Arduino IDE -Version 2.3.2

• Arduino IoT

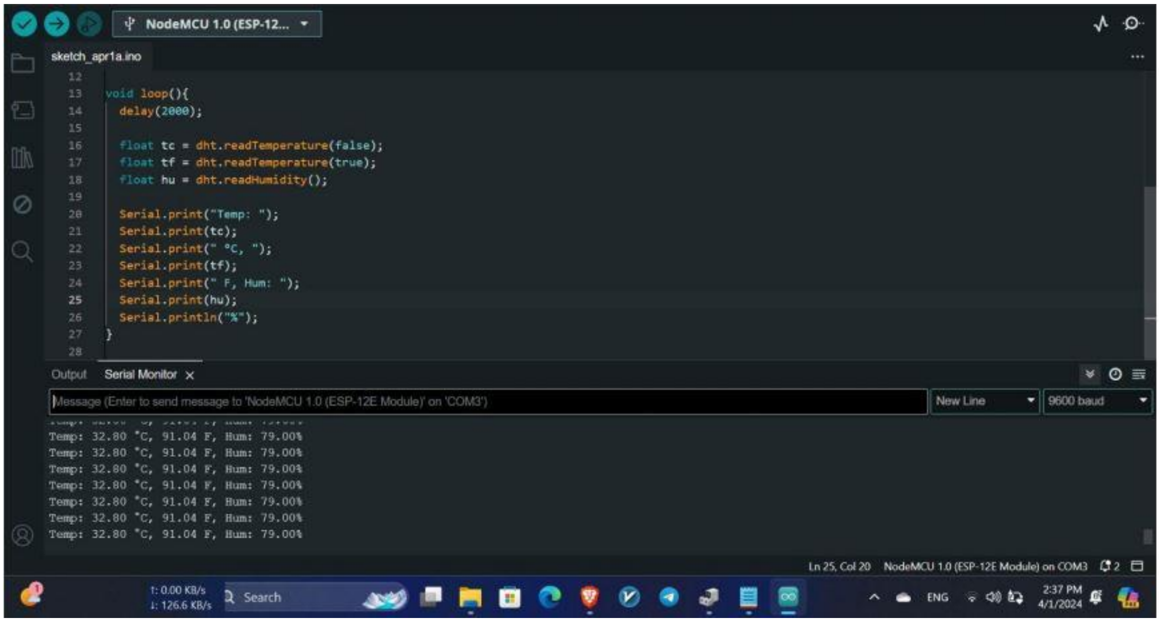

Figure 2-1 Software Interface

- **Communications Interfaces**

For The smart plant caring system project, various communication interfaces are essential to enable data exchange, remote monitoring, and control functionalities. These interfaces facilitate interactions between different system components, such as sensors, microcontrollers, user interfaces, and external networks.

• Wi-Fi (802.11

- Enables the system to connect to local wireless networks for internet access.

- Allows remote monitoring and control of the plant caring system via a mobile app or web interface.

• Cellular Communication

- 2G/3G/4G/LTE

- Provides connectivity to cellular networks for remote monitoring and control in areas with cellular coverage.

- Suitable for outdoor deployments where Wi-Fi or other wireless options are limited.

• Cloud Connectivity

- HTTP/HTTPS

- Facilitates communication with cloud-based servers for data storage, analysis, and remote access.

- Enables seamless integration with cloud platforms (e.g., AWS, Google Cloud) for scalable data management.

• Microcontroller Communication Protocols

- I2C (Inter-Integrated Circuit)

- Common protocol for connecting sensors and peripherals to a microcontroller, allowing data exchange over short distances.

- Suitable for intra-board communication within the system.

• SPI (Serial Peripheral Interface)

- Enables high-speed communication between microcontrollers and peripherals (e.g., display modules, memory devices).

- Supports full-duplex data transfer for efficient data exchange.

2.1.2 Non – Functional requirements

- **Performance Requirements**

  • A proper power supply should be provided to the product.

  • Soil moisture should provide accurate readings for efficient watering.

  • The water pump should work when the soil moisture drops below a certain threshold. So the pump should work with a good response time.

  • The temperature and humidity sensors must provide real-time and accurate data.

  • The light timer should switch on and off at desired intervals with consistent timing.

- **Safety Requirements**

  • Electrical safety

    Avoid contact with electrically conductive liquids and materials, and electrical components to avoid short circuits.

  • Usage safety

After installation, the product must be handled with care as it contains water.

Water may leak in case of product failure

• Environment safety

Dispose of the electronic components responsibly.

- **Security Requirements**

• Data Privacy:

Users should be aware that their user data (e.g. soil moisture, temperature) travel over the Internet.

• Physical Security:

Place the product in a safe and stable place.

Protect electronic devices from environmental factors.

- **Software Quality Attributes**

• Reliability:

Our product aims to ensure reliable operation, with a target of at least 99% uptime during normal operation.

It should be able to recover from failures gracefully without causing data loss or system instability.

• Maintainability:

The system should be designed with modular components and clear documentation to facilitate ease of maintenance and future enhancements.

The codebase should adhere to coding standards and best practices to simplify troubleshooting and debugging.

• Usability:

The user interface should be intuitive and easy to navigate, allowing users to interact with the system without extensive training.

A usability testing process will be conducted to ensure that the product meets the needs and expectations of our customers.

2.2 Design

The system will get soil moisture, temperature, humidity and time as inputs. The current soil moisture value will calculated from the soil moisture sensor and according to that value system will decide to whether it is need to water the plants. If it is needed to water the plants system will send a signal to the water pump controlling the relay module to turn on the pump. The grow light will turn on and off in real-time. A turn-on signal will be generated and sent to the light-controlling relay module by the system when the light turning on time is detected. And it will turn off the light similar to the light turning on event.

The temperature and humidity captured by the temperature sensor and those values will be sent to the system. The display will show the current temperature, humidity and soil moisture levels. From the user application user can observe all the above information and user can get pump and light status remotely. From the application user can set a threshold value for the soil moisture and set the light turning on and turned on for duration.

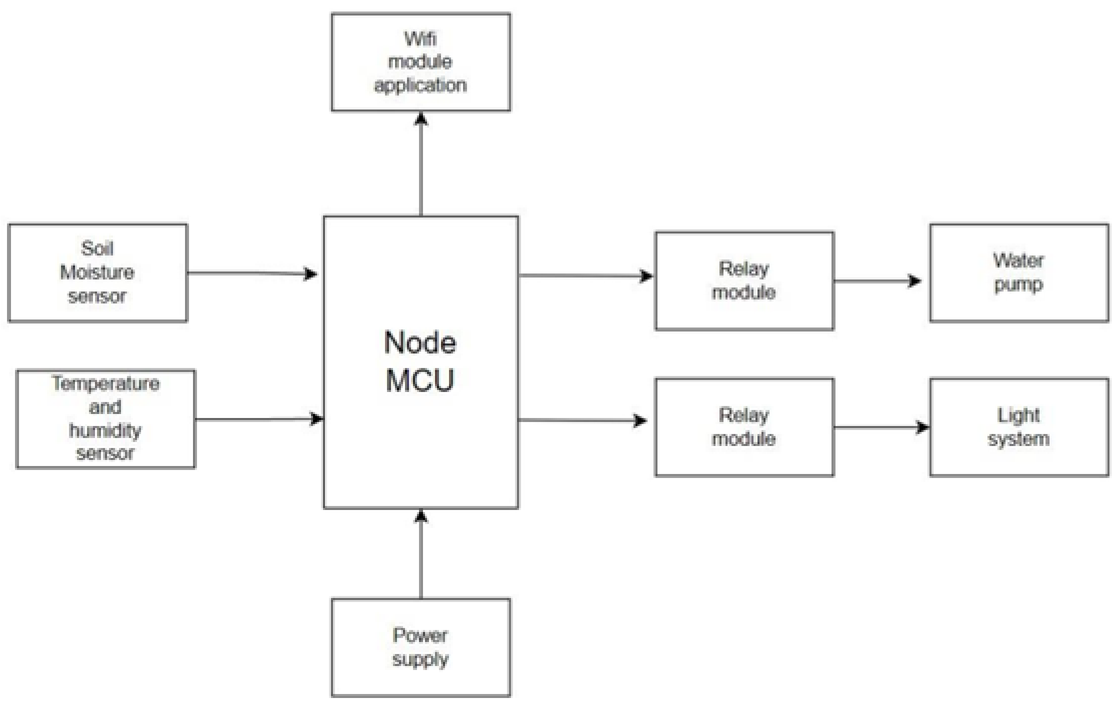

Figure 2-2 Design Diagram

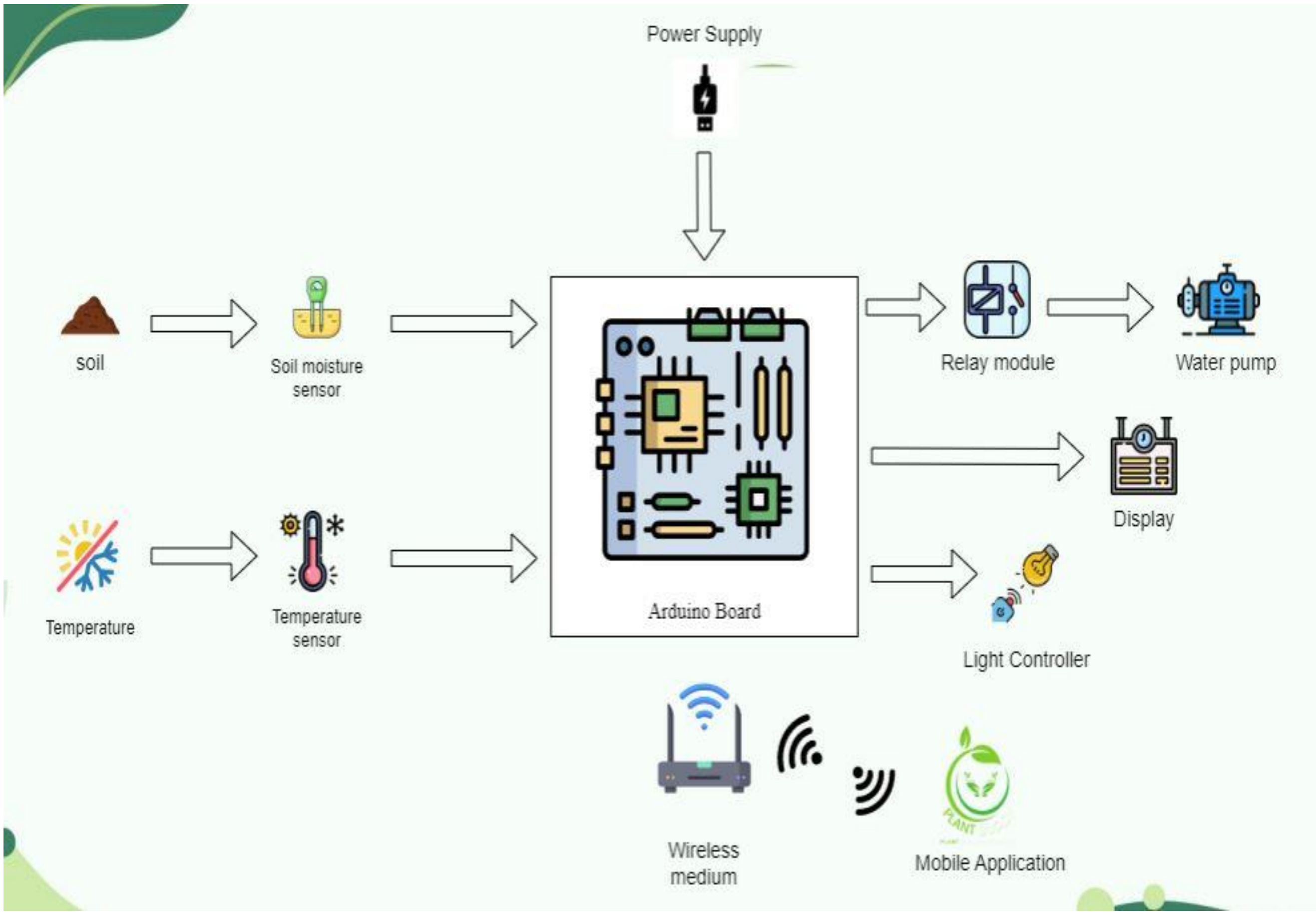

Figure 2-3 High Level Architecture Diagram

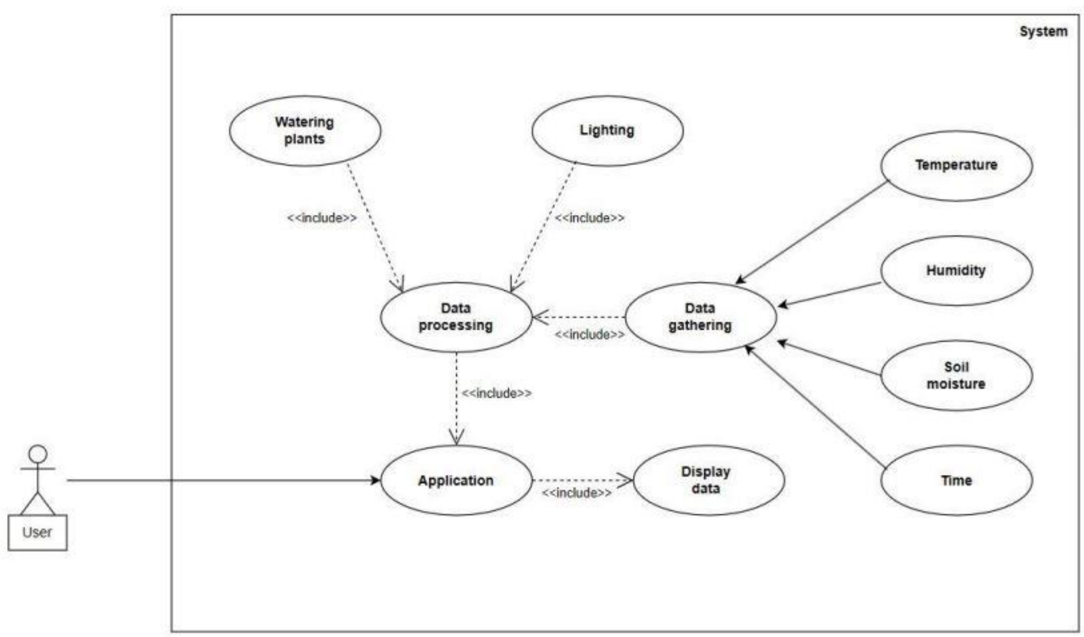

Figure 2-4 Use Case Diagram

Similar to a flowchart or data flow diagram, an activity diagram visually displays a series of actions or the flow of control in a system.

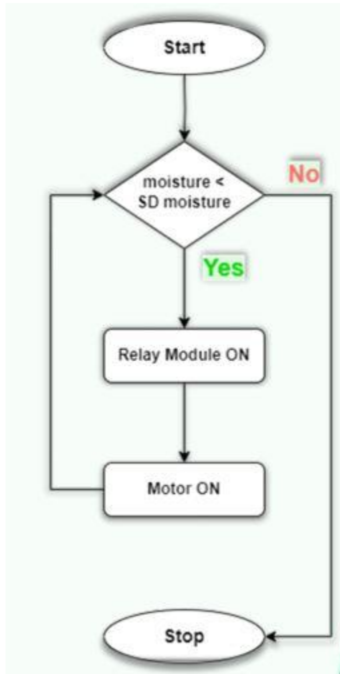

Figure 2-5 Activity Diagram for Watering System

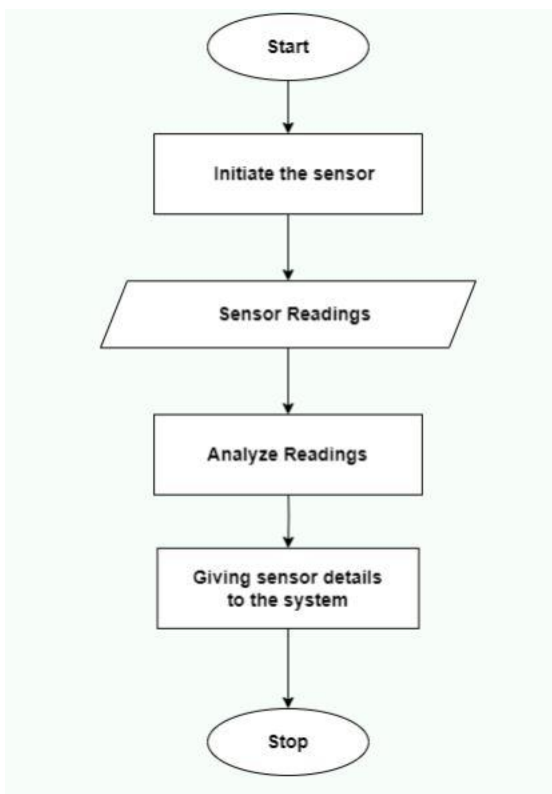

Figure 2-6 Activity Diagram for Soil Moisture check

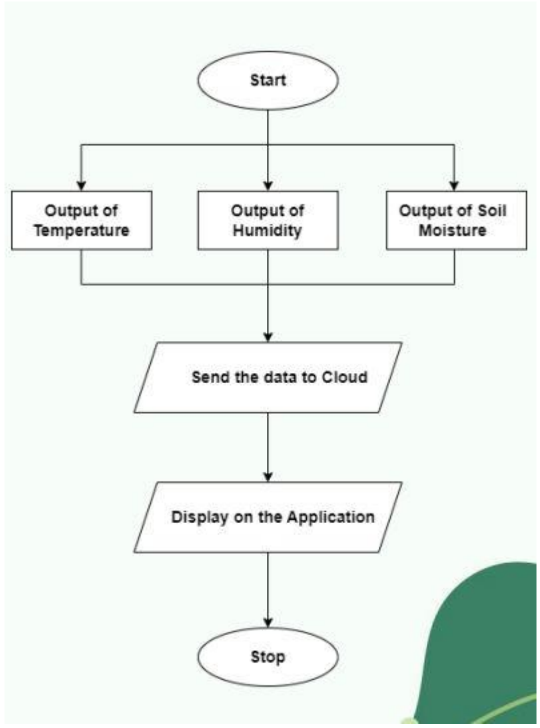

Figure 2-7 Activity Diagram for Monitoring Application

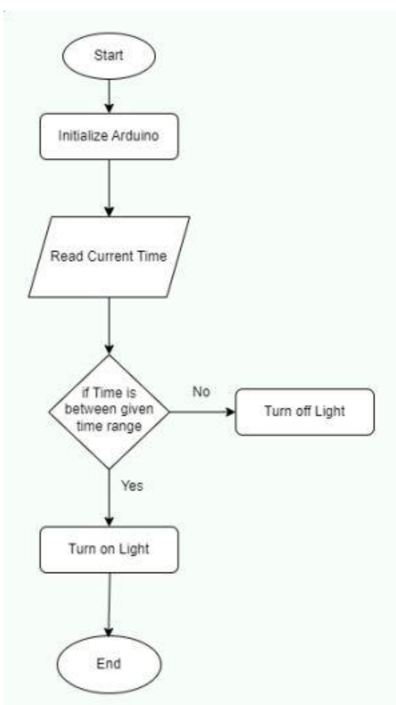

Figure 2-8 Activity Diagram for Light Sysytem

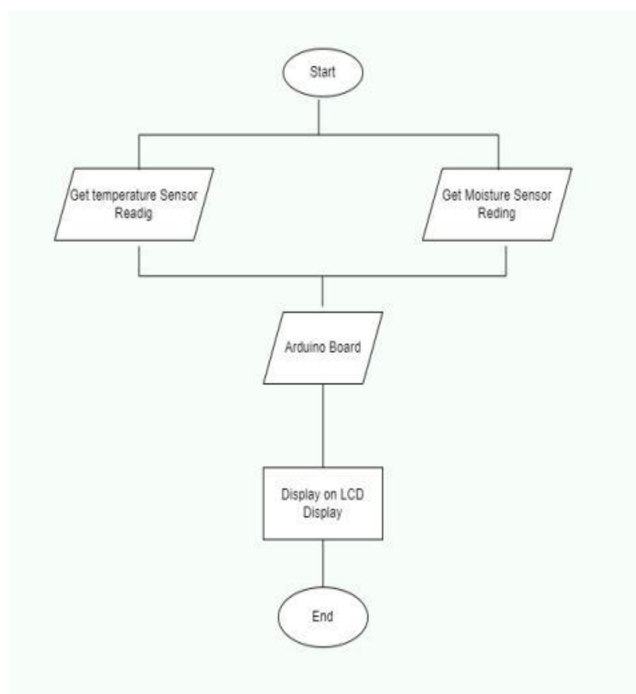

Figure 2-9 Activity Diagram for Display System

2.3 Implementation

1. Soil moisture sensing and pump control

Soil moisture value is by soil moisture sensor. A default threshold value is set in the system and user can set their own value. The soil moisture sensor will generate values according to the soil status. This value cannot be used directly for the pump activation. So the soil moisture reading needs to map to a percentage value. This can be done by defining lowest reading value and the highest reading value. Then we can map soil moisture value between the defined values and get a percentage value

```
//constant for dry sensor
const int DryValue = 1024;

//constand for wet value
const int WetValue = 600;

//variables for soilMoisture
int soilMoistureValue;
int soilMoisturePercent;
```

Figure 2-10 Soil moisture Senssing

```
//Get Soil Moisture Value
soilMoistureValue = analogRead(SENSOR_IN1);

//Determine soil Moisture percentage value
soilMoisturePercent = map(soilMoistureValue, DryValue, WetValue, 0, 100);

//Keep Values Between 0 and 100
soilMoisturePercent = constrain(soilMoisturePercent, 0, 100);
```

Figure 2-11 Pump Control

After getting the current soil moisture percentage system will compare it with the threshold value. If the current soil moisture percentage is less than the defined soil moisture percentage system will turn on the pump and water the plants until the moisture value get into the proper state.

```
//see if pump should be triggerd
if (soilMoisturePercent <= pump_trigger){
  //Turn Pump On
  pumpOn();

}else {
  //Turn Pump Off
  pumpOff();
}
```

Figure 2-12 compare with the threshold value

2. Light timer implementation

The grow light will turn on and off to real-time by the system to provide necessary light to the plants. This is achieved by the scheduler in the Arduino. When the defined time is reached system will trigger the relay module of the light and turn on it. After the defined duration is reached system will turn off the light.

```
//Light System

if (scheduler.isActive()){
  digitalWrite(light, LOW);
  light_Status =true;
}else {
  digitalWrite(light, HIGH);
  light_Status = false;
}
```

Figure 2-13 Light Timer implementation

3. Temperature, humidity and soil moisture display

First using the defined functions of the DHT sensor library we can get current temperature value and humidity value.

```
// Get Temperature and Humidity
temp = dht.readTemperature();
hum = dht.readHumidity();
```

Figure 2-14 Display Sensor Reading

Then we can display the details in the LED display using the functions which are defined by the display library.

```
//DISPLAY

// Set the cursor to the first column and first row
 lcd.setCursor(0, 0);
 lcd.print(" *-PLANT BUDDY-*");   // Print some text
 lcd.setCursor(0,1);   // first column second row
 lcd.print("S:");
 lcd.print(soilMoisturePercent); // soil moisture
 lcd.print(" T:");
 lcd.print(temp);    // current temperature
 lcd.print(" H:");
 lcd.print(hum);    // current humidity
```

Figure 2-15 Display details in LCD Display

4. Mobile Application Integration

The details of the system will send to the cloud by the internet and user can view information about plant status. To implement this function first we declare variables for each sensor reading values. These variable should implement correctly otherwise the system will not work properly. These variables will automatically updated by Arduino IOT and frequently read and write on these variables according to the sensor readings. The Arduino board will connect to the internet and update those variables to view and update the system.

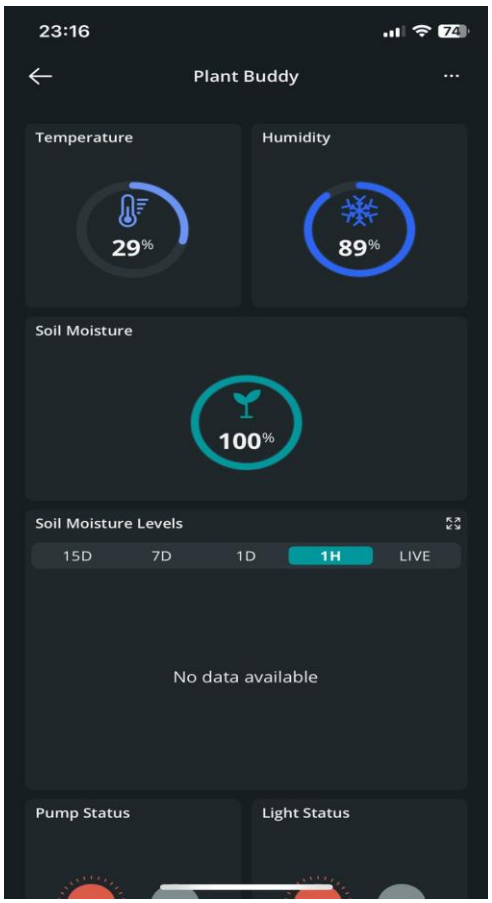

Figure 2-16 Mobile App Interface 01

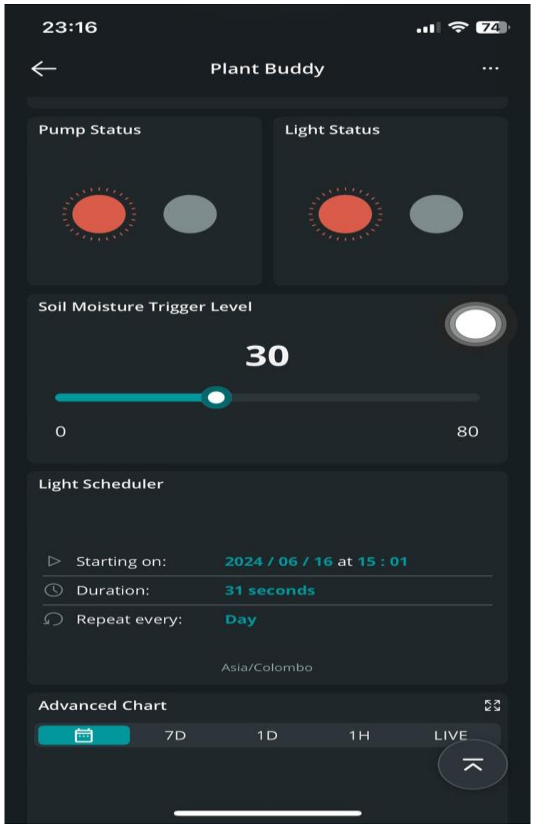

Figure 2-17 Mobile App Interface 02

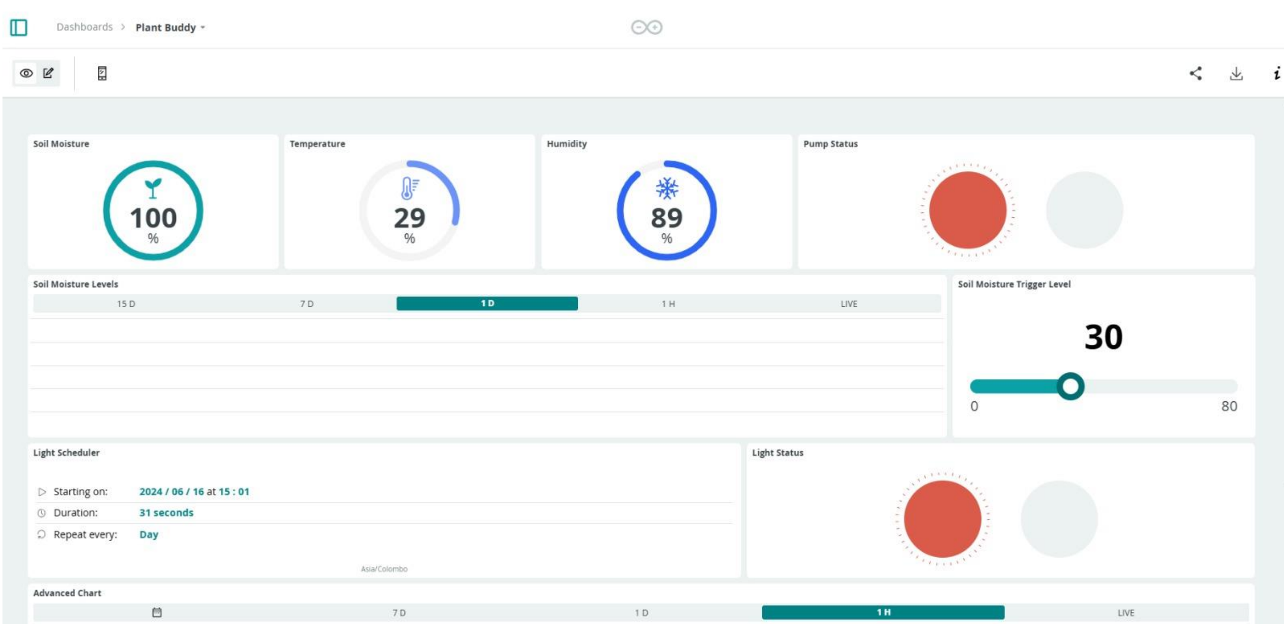

Figure 2-18 Desktop view

5. Physical Structure

Figure 2-19 Physical Structure

2.4 Testing

Table 1 Testing Table

S. No	Action	Inputs	Expected Output	Actual Output	Test Result
1	Increase the soil moisture	Soil moisture sensor values	Display the increased soil moisture value.	Display the increased soil moisture value.	PASS
2	Decreased the soil moisture	Soil moisture sensor values	Display the decreased soil moisture value.	Display the decreased soil moisture value	PASS
3	Reach the defined turn on trigger time	Current time	Light turned on	Light turned on	PASS
4	Reach the defined turn off trigger time	Current time	Light turned off	Light turned off	PASS
5	Increase the temperature around the system	Temperature sensor value	Display the increased temperature value in the display	Display the increased temperature value in the display	PASS
6	Decrease the temperature around the system	Temperature sensor value	Display the decreased temperature value	Display the decreased temperature value	PASS
7	Change the	Soil moisture	Pump turn on	Pump turn on	PASS

	soil moisture trigger level value	sensor values	when soil moisture less than the trigger value	when soil moisture less than the trigger value	
8	Increase the humidity around the system	Humidity sensor value	Display the Increased humidity value	Display the Increased humidity value	PASS
9	Decrease the humidity around the system	Humidity sensor value	Display the decreased humidity value	Display the decreased humidity value	PASS
10	Pump turned on	Current pump status	Display the pump turn on status in the application. Send notification to the use via the application	Display the pump turn on status in the application. Send notification to the use via the application	PASS
11	Pump turned off	Current pump status	Display the pump turn off status in the application. Send notification to the use via the application	Display the pump turn off status in the application. Send notification to the use via the application	PASS
12	Light turned	Current light	Display the	Display the	PASS

	on	status	light turn on status in the application	light turn on status in the application	
13	Light turned off	Current light status	Display the light turn off status in the application	Display the light turn off status in the application	PASS

3. Conclusion

Assessment of the Project Results

The Smart Plant Caring System successfully achieved its primary objective of monitoring and maintaining optimal conditions for plant growth. The integration of soil moisture sensors and DHT11 sensors(check temperature & humidity) enabled precise measurement of soil moisture levels, temperature, and humidity. The system's ability to display real-time data on an LCD screen provided immediate feedback, enhancing user interaction and monitoring capabilities.

The automatic Pumping system, controlled by predefined moisture trigger levels, demonstrated efficiency in maintaining soil moisture. The motor activation and deactivation based on soil moisture readings ensured that plants received adequate water without manual intervention. The incorporation of the Arduino Cloud for controlling the motor, light scheduling, and data analysis further enhanced the system's functionality, allowing for remote monitoring and adjustments. the project successfully combined hardware and software components to create a robust and automated plant care system.

Lessons Learned

Throughout the development of the Smart Plant Caring System, several valuable lessons were learned:

1. Sensor Calibration and Accuracy: Ensuring the accurate calibration of sensors is crucial. Initial discrepancies in sensor readings highlighted the importance of thorough testing and calibration to achieve reliable data.
2. System Integration: Integrating multiple components, such as sensors, LCD displays, motors, and cloud services, required careful planning and troubleshooting. This experience underscored the significance of system compatibility and the need for meticulous wiring and coding.

3. Power Management: Managing power consumption for the entire system, especially when integrating motors and LCD displays, was essential to maintain stable operation. This involved selecting appropriate power sources and optimizing code for energy efficiency.
4. Cloud Connectivity: Utilizing the Arduino Cloud introduced challenges related to connectivity and data synchronization. Learning to navigate these challenges and ensuring reliable communication between the cloud and hardware was a key learning point.

Future Work

The Smart Plant Caring System has laid a solid foundation for further enhancements and expansions. Future work could focus on the following areas:

1. Enhanced Sensor Suite: Incorporating additional sensors, such as pH sensors, light intensity sensors, and nutrient sensors, would provide a more comprehensive understanding of the plant's environment, enabling even more precise care.
2. Advanced Data Analytics: Leveraging machine learning algorithms to analyze historical data and predict optimal care schedules could further automate and optimize plant care. This could include predicting watering needs based on weather forecasts and plant growth patterns.
3. Mobile Application: Developing a mobile application for remote monitoring and control would increase user convenience and accessibility. This app could provide real-time alerts and allow users to adjust settings on the go.
4. Scalability and Customization: Expanding the system to cater to different types of plants with specific care requirements would enhance its versatility. Customizable care routines and modular design would allow users to tailor the system to their unique needs.
5. Energy Efficiency: Exploring alternative power sources, such as solar panels, and optimizing power consumption could make the system more sustainable and suitable for outdoor applications.

In conclusion, the Smart Plant Caring System project has successfully demonstrated the potential of combining sensor technology, automation, and cloud connectivity to improve plant care. The lessons learned during this project will guide future enhancements, aiming to create an even more efficient, user-friendly, and versatile plant care solution.

4. References

• [1]"Arduino - Home", Arduino. cc, 2022. Available: https://www.arduino.cc/

• [2]"Arduino Uno Rev3", Arduino Online Shop, 2022. Available:
https://store usa.arduino.cc/products/nodemcu-esp8266

• [3]“Water Your Garden with IoT - Soil Moisture Sensors,” “Water Your Garden with IoT - Soil Moisture Sensors,” www.youtube.com. https://www.youtube.com/watch?v=pgGpuws7f9o (accessed Apr. 26, 2022)

• [4]“DroneBot Workshop,” DroneBot Workshop, 2016. https://dronebotworkshop.com/