Smart Farm

Graduation Project

by

1- Raheek Ayman Mohamed Helal

2- Sama Eslam Mohamed Helal

3- Mohamed Ahmed Awad El-saeed

4- Mohamed Gamal Abbas Ali

5- Heba Mahmoud Fathy Allam

A project submitted in partial fulfilment of the requirements for the degree of Bachelor of Science Information Technology

Supervised by

Prof. Dr. Magdy Zakaria

Academic Year 2024/2025

Abstract

This project develops a smart farming system by integrating embedded systems with a unified web interface to optimize agricultural operations. The embedded systems provide real-time data from the farm environment, while the web interface allows users to monitor, analyze, and control farming activities. Key functionalities include automated irrigation, climate control, and plant health monitoring. The system features a user-friendly dashboard and advanced data processing for decision support.

Keywords: Smart farming, embedded systems, web interface, real-time monitoring, automated irrigation, climate control, plant health monitoring, data visualization, decision support, agricultural technology.

Acknowledgement

The content of this single page is left to the discretion of the student. It is suggested however that the page makes reference to guidance received by the student from his or her supervisor and project committee members. Reference should also be made to any financial assistance received to carry out the project. Any extraordinary assistance received by the student for example in word processing, data collection, data analysis, and so on, should be properly acknowledged. Example acknowledgement can be found in books, reports and also papers. The acknowledgements should not exceed 250 words.

Table of Content

List of Figures

List of Tables

Chapter 1: Introduction

1.1 Introduction

The agricultural industry is undergoing a transformation with the integration of technology to address challenges such as **resource management**, **crop health**, and **climate unpredictability**. Traditional farming methods often rely heavily on manual labor and intuition, leading to inefficiencies and missed opportunities for optimization. To overcome these challenges, **smart farming systems** have emerged as a modern solution that leverages embedded systems and web-based technologies.

This project introduces a **Smart Farm system** designed to improve agricultural productivity and sustainability. By combining embedded systems with a comprehensive web interface, the system enables **real-time monitoring**, **automated control**, and **data-driven decision-making**. The Smart Farm system ensures optimal resource utilization through functionalities like automated irrigation, climate control, and plant health analysis. Furthermore, the intuitive web interface empowers users to access **real-time data**, analyze trends, and **manage farm operations effectively**, fostering innovation in agricultural practices.

1.2 Problem Statement

Traditional farming methods often result in inefficiencies that hinder productivity and sustainability. These methods, which rely heavily on manual processes for critical tasks like irrigation, temperature control, and plant health monitoring, are not equipped to address the complexities of modern agriculture. Such limitations lead to several significant challenges:

- **Wasted Water:** Inefficient irrigation systems frequently result in overwatering, depleting valuable water resources.
- **Limited Monitoring:** Farmers lack real-time access to vital information about crop and environmental conditions, delaying necessary interventions.

- **Unhealthy Crops:** The inability to detect plant diseases or nutrient deficiencies early can compromise crop health and reduce yields.
- **High Labor Costs:** Dependence on manual labor for routine farming tasks drives up operational costs, making traditional practices increasingly unsustainable.

The **Smart Farm project** addresses these challenges by introducing a smart farming system that automates key agricultural tasks and provides a user-friendly interface for real-time monitoring and management. By leveraging advanced technology, the project aims to optimize resource utilization, enhance crop health, and reduce labor dependency, thereby transforming traditional farming into a more efficient and sustainable practice.

1.3 Project Objectives

The **Smart Farm project** aims to revolutionize traditional farming practices by integrating advanced technology to address inefficiencies and enhance productivity. The key objectives of the project are:

1. **Automate Farm Operations:**
    - Develop systems to autonomously manage irrigation, temperature, and lighting by analyzing real-time data collected from sensors, ensuring optimal farm conditions with minimal manual intervention.
2. **Comprehensive Crop Monitoring:**
    - Build an integrated system to continuously monitor critical environmental parameters, such as soil moisture, temperature, humidity, and plant health, providing farmers with actionable insights.
3. **Real-Time Data Visualization:**
    - Design a user-friendly web interface that offers live updates, detailed metrics, and historical trends, empowering farmers to remotely oversee and manage their farm operations.
4. **Optimize Resource Efficiency:**

- Implement technology to reduce water and energy waste, enhancing resource utilization and promoting sustainable farming practices while improving crop yields.

5. **Proactive Alert System:**
    - Develop a robust notification mechanism that promptly informs farmers of critical issues, such as low soil moisture, equipment malfunctions, or abnormal environmental conditions, enabling rapid response and mitigation.
6. **Scalable and Flexible Architecture:**
    - Engineer the system with scalability in mind, allowing it to accommodate larger farms, additional sensors, or new functionalities as agricultural needs evolve.

By achieving these objectives, the Smart Farm project aims to contribute to sustainable agriculture and empower farmers with the tools needed to maximize crop yield and resource efficiency.

1.4 Project Scope

The **Smart Farm project** aims to transform traditional farming practices by developing an advanced, technology-driven solution. The project encompasses the design, implementation, and testing of an integrated system that leverages embedded devices and a web-based interface to optimize agricultural operations. The scope of the project includes:

1. **Resource Optimization:**
    - Implement automated irrigation and climate control systems to minimize water and energy waste.
2. **Enhanced Crop Management:**
    - Provide tools to maintain optimal growing conditions and ensure early detection of plant health issues, resulting in improved crop yields.
3. **User-Friendly Interface:**
    - Develop a web-based dashboard that offers farmers an intuitive way to monitor their farms, control operations, and access real-time data remotely.

4. **Data-Driven Agriculture:**
   - Enable informed decision-making through advanced analytics and live monitoring of critical environmental parameters.
5. **Scalable and Flexible Design:**
   - Build a system architecture that supports the integration of additional sensors, features, or expansions to accommodate growing farm needs.
6. **Sustainability and Cost Efficiency:**
   - Focus on reducing environmental impact and farming costs by automating routine tasks and optimizing resource utilization.

1.5. Project Methodology

1. Research and Requirements Gathering

- Conduct a comprehensive review of existing smart farming technologies to identify best practices and innovative features.
- Collaborate with stakeholders, including farmers and agricultural experts, to define functional and non-functional requirements.
- Evaluate potential hardware and software solutions for system components such as sensors, actuators, and the web interface.

2. System Design

- Develop a detailed architecture for the embedded systems and web interface, specifying data flow, communication protocols (e.g., MQTT, HTTP), and system interactions.
- Create mockups for the web dashboard to visualize real-time data, alerts, and control functionalities.
- Design scalable solutions to accommodate future expansions and additional features.

3. Hardware and Software Development

- Implement the embedded system using sensors (soil moisture, temperature, humidity, light) and actuators for irrigation, climate control, and lighting systems.
- Build the web interface using modern frameworks to ensure a responsive and intuitive user experience.
- Integrate backend services for real-time data processing, analytics, and notifications.

4. Integration and Testing

- Test individual system components (sensors, actuators, dashboard) to ensure reliability and accuracy.
- Conduct end-to-end testing of the integrated system in a simulated farm environment to verify communication and functionality.
- Identify and resolve any bugs or performance bottlenecks.

5. Deployment and Validation

- Deploy the system in a real-world farm setting for pilot testing.
- Collect feedback from users to refine system features and usability.
- Monitor system performance under varying conditions to ensure scalability and robustness.

1.6. Project Timeline

The project timeline visually represents the key phases and milestones of the **Smart Farm project**. It provides an overview of the tasks and their corresponding durations, from initial research and requirements gathering to the final deployment and validation stages. Each phase is outlined in a linear sequence, ensuring a clear progression of work. The timeline includes:

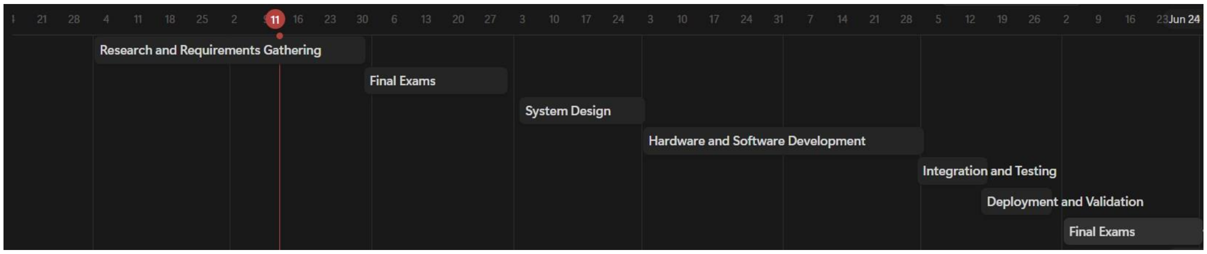


Figure 1.1 – Project Timeline

1.7. Project Organization

Chapter 1 introduces the project objectives, the motivation of the project, the approach used in this project, the contribution of this project, the scope of the work, and project layout.

Chapter 2 provides the reader with an overview of the literature review.

Chapter 2: Literature Review

2.1 Introduction

The **Related Work** section explores previous research, technologies, and systems that are relevant to the **Smart Farm project**. This section highlights existing approaches to smart farming, including automated systems for irrigation, climate control, and crop monitoring. By reviewing the current state of the field, we identify both the advancements and limitations in smart farming technologies. This enables the Smart Farm project to build on the knowledge gained from previous efforts while addressing existing gaps and challenges in the industry.

2.2 Background

Smart Farming Technologies:

- Smart farming integrates IoT devices like sensors and automated systems to enhance farming efficiency. These technologies enable real-time data collection on environmental factors to optimize farming practices.

Embedded Systems in Agriculture:

- Embedded systems automate agricultural processes, reducing labor costs and improving crop yields. They include sensors and actuators for irrigation and climate control, with machine learning for disease detection.

Communication Protocols for Data Transfer:

- MQTT and HTTP protocols are commonly used to transmit data from farm sensors to cloud platforms or servers. MQTT is favored for its efficiency and low bandwidth usage in real-time applications.

Web-Based Farm Management Systems:

- Web-based platforms allow remote monitoring and control of farm operations through user-friendly dashboards. They provide real-time data visualization, enhancing decision-making and operational efficiency.

2.3 Review of Relevant Work

2.3.1 Relevant Projects (Systems)

2.3.1.1 FarmAssist:

FarmAssist is an open-source project focused on providing a smart farming solution using IoT devices. It integrates sensors for real-time monitoring and decision-making processes, offering features like automated irrigation and soil moisture tracking. The system uses NodeMCU ESP8266 and MQTT for data communication, enabling efficient farm management.

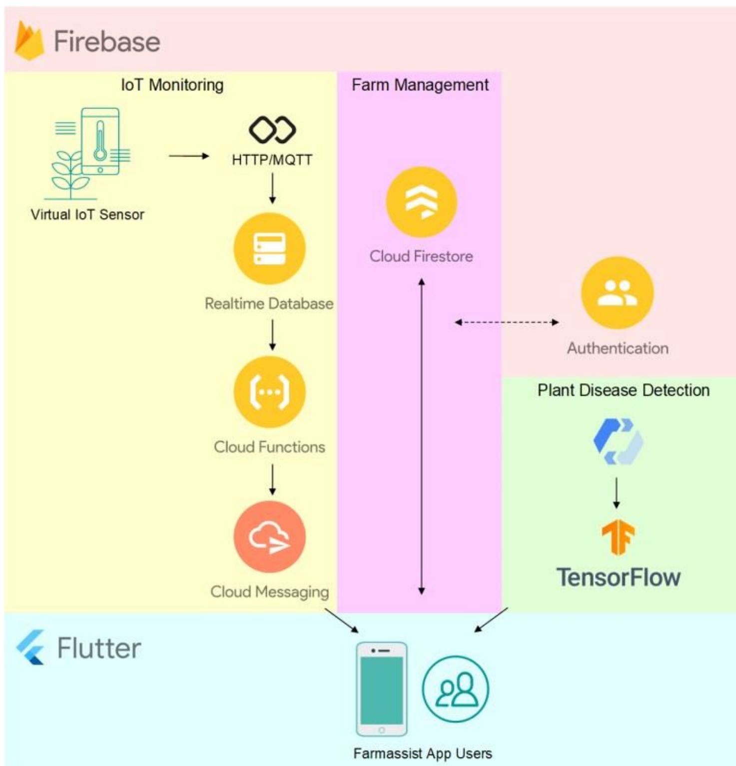


Figure 2.1 – Farmassist Architecture

2.3.1.2 IoT-Based Smart Agriculture Monitoring System:

The project utilizes NodeMCU ESP8266 and various sensors to monitor agricultural conditions such as soil moisture, temperature, and humidity. It automates irrigation based on sensor data, improving resource management. The system also integrates with cloud platforms for remote monitoring and control.

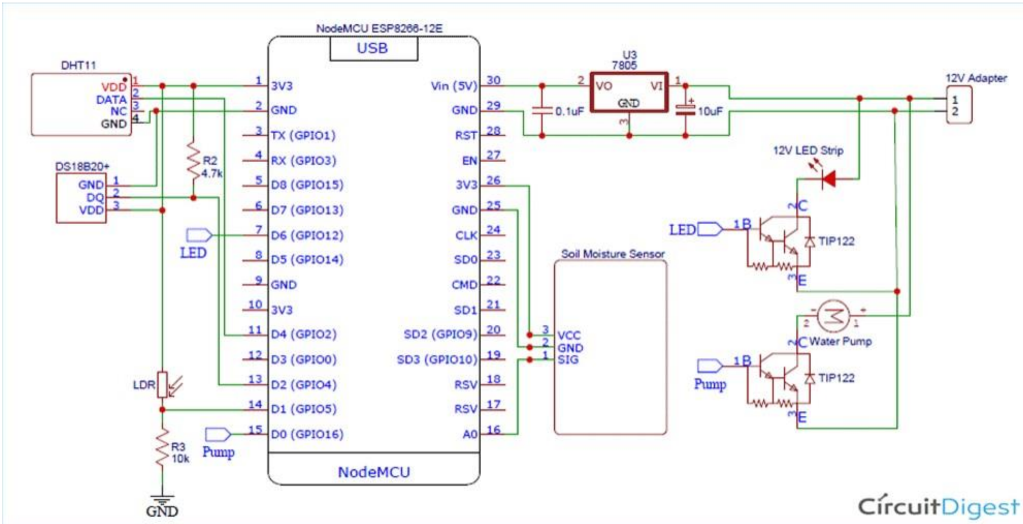


Figure 2.2 – Smart Agriculture System Circuit Diagram

2.3.2 Relevant Research

2.3.2.1 [Smart Agriculture: A Review (ResearchGate)]

This paper reviews various smart agriculture technologies, focusing on IoT, big data, and AI applications for optimizing agricultural practices. It discusses automated systems for irrigation, climate control, and crop monitoring.

Figure 2.3 – The Smart Farm Conceptual Framework

2.4 Relevant Work and Our Own Work

The related works highlight the role of IoT devices, automation, and cloud-based monitoring in improving agricultural practices. Similarly, our project focuses on automating farm operations using real-time data from sensors for irrigation, climate control, and plant health. While FarmAssist and the IoT-based monitoring systems automate irrigation and data collection, our system will additionally offer a comprehensive web interface, advanced analytics, and machine learning capabilities for disease detection. This integration of data processing and decision support differentiates our work from existing solutions.

Chapter 3: System Analysis

3.1 Introduction

Chapter 3 presents a detailed analysis of the Smart Farm system, focusing on its requirements, design, and the underlying technologies that support its functionality. This chapter outlines the core components of the system, including embedded systems, sensors, actuators, and the web interface. It also examines the various data processing techniques and the role of automation in optimizing farm management. The system's scalability, reliability, and security are also evaluated to ensure its effectiveness in real-world agricultural environments.

3.2 System Requirements

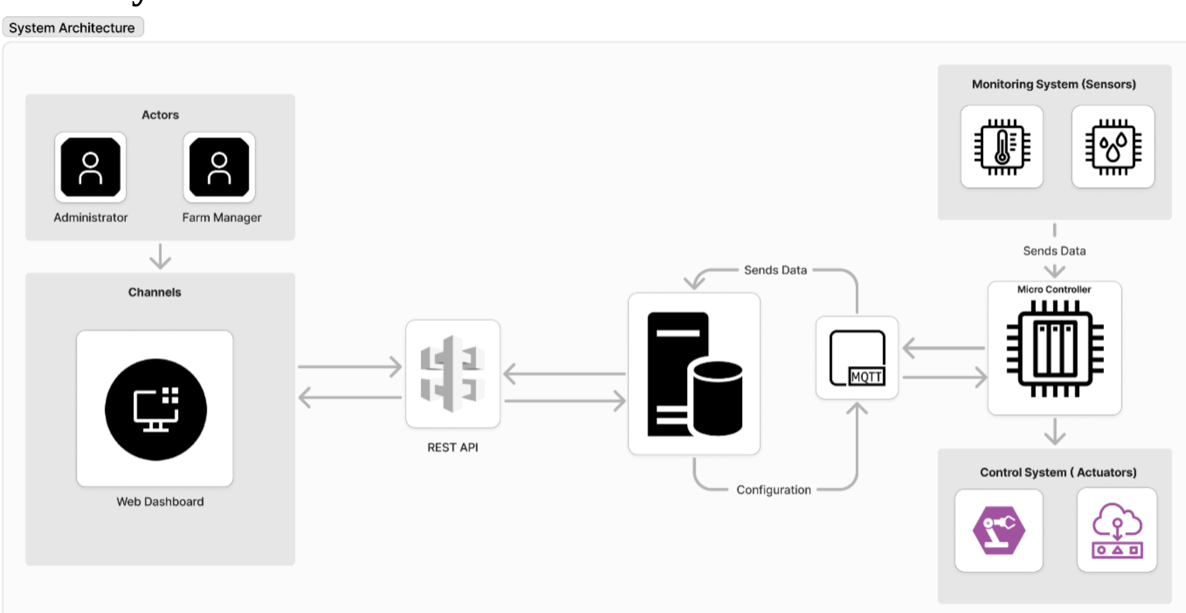
3.2.1 Functional Requirements

Name	Description	Priority
Environmental Monitoring	Collect data from sensors such as soil moisture, temperature, humidity, and light.	High
Irrigation Control	Automate irrigation based on sensor data and weather conditions, with manual override options.	High
Climate Control	Adjust greenhouse temperature, humidity, lighting, and control ventilation and heating systems.	High
Plant Health Monitoring	Monitor plant stress using light, temperature, and humidity sensors, and alert for any deviations.	Medium
Dashboard Display	Show real-time data like temperature, humidity, soil conditions, and system status.	High
System Control	Allow manual control of irrigation and climate settings, and scheduling of tasks.	High
Notification System	Send real-time alerts via email/SMS for critical conditions like low soil moisture or sensor failure.	High
Reporting	Generate performance reports including plant health trends, farm metrics, and environmental impact.	Medium
User Authentication	Secure login with role-based access (e.g., Admin, Farm Worker).	High
Weather Integration	Display weather forecasts for planning farm activities.	Medium

3.2.2 Non-Functional Requirements

Name	Description	Priority
Real-Time Data Updates	Ensure data transmission has less than 2 seconds latency to provide real-time updates.	High
System Uptime	Guarantee 99.9% uptime for the web interface and backend services.	High
Data Protection	Use HTTPS for secure communication and encrypt sensitive user data using industry-standard algorithms.	High
Access Control	Implement role-based access control (Admin, Worker) to restrict user access based on roles.	High
Feature Expansion	Design the system to allow easy integration of new features/modules as needed.	Medium
Ease of Use	Ensure that 90% of users can navigate the system without prior training.	High
Interface Responsiveness	Maintain system responsiveness across different devices and screen sizes.	High
Error Logging	Provide detailed logs for system issues, sensor data transmission, and user interactions.	High
Environmental Standards	Ensure system compliance with agricultural and environmental regulations.	Medium
Data Standards	Adhere to JSON or XML standards for data exchange between system components.	Medium

3.3 System Architecture



3.4 Development Methodology

3.4.1 Use Case Description

Use Case: Control and Monitor Farms to Reduce Waste and Increase Efficiency Using Smart Embedded Systems

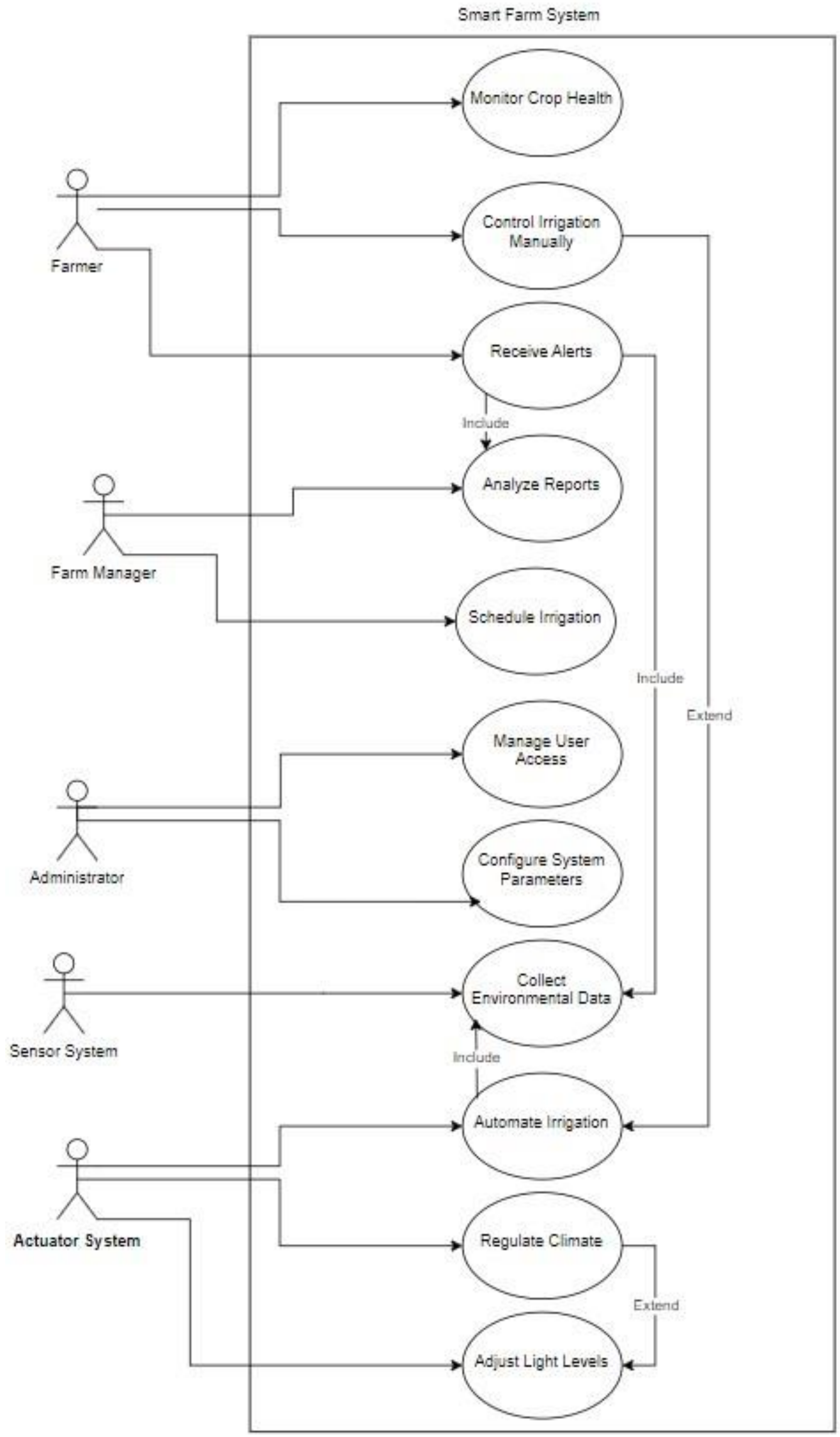
Actors:

- **Farm Manager**: Oversees operations, manages farm data, schedules automation, and generates reports.
- **Administrator**: Manages user access, roles, system configurations, and oversees high-level operations.
- **Monitoring System**: Embedded sensors that collect real-time data.
- **Control System**: Actuators and automation systems.

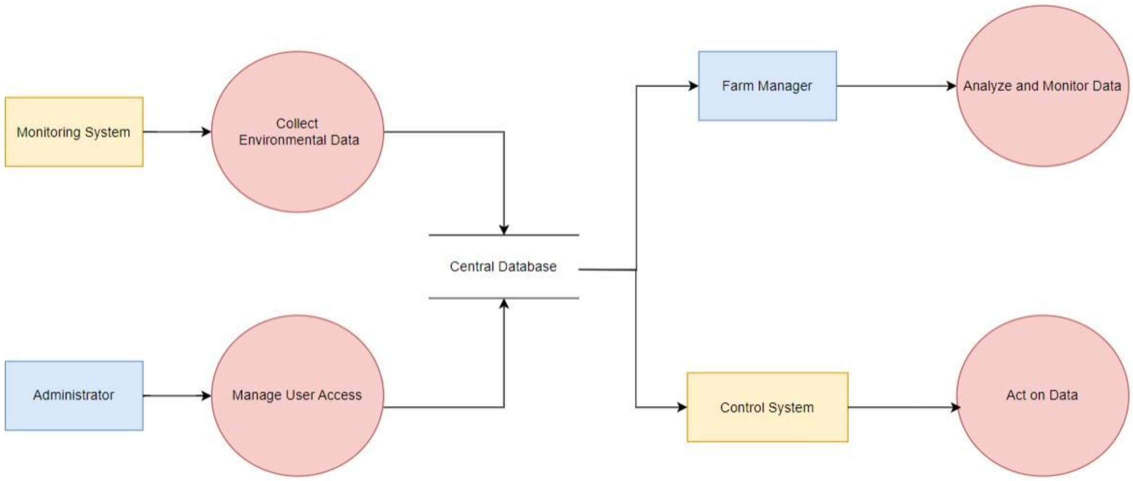
Description:

- The **Smart Farm System** is designed to automate and optimize farm operations using advanced embedded systems and a web-based interface.
- The **Monitoring System** continuously collects data from sensors, including soil moisture, temperature, humidity, and light. This data is processed and transmitted to the backend in real-time using protocols like MQTT/HTTP.
- The **Control System** automates critical tasks such as irrigation, climate control, and plant health monitoring.
- Farm managers interact with the system via a **web dashboard** that displays real-time data, provides historical trends, and allows manual overrides or scheduled automation.
- Administrators manage **user roles, system configurations**, and **data security** to ensure smooth operations. The system is designed to be **scalable**, supporting multiple sensors and allowing future feature expansions

3.4.2 **Use Case Diagram**

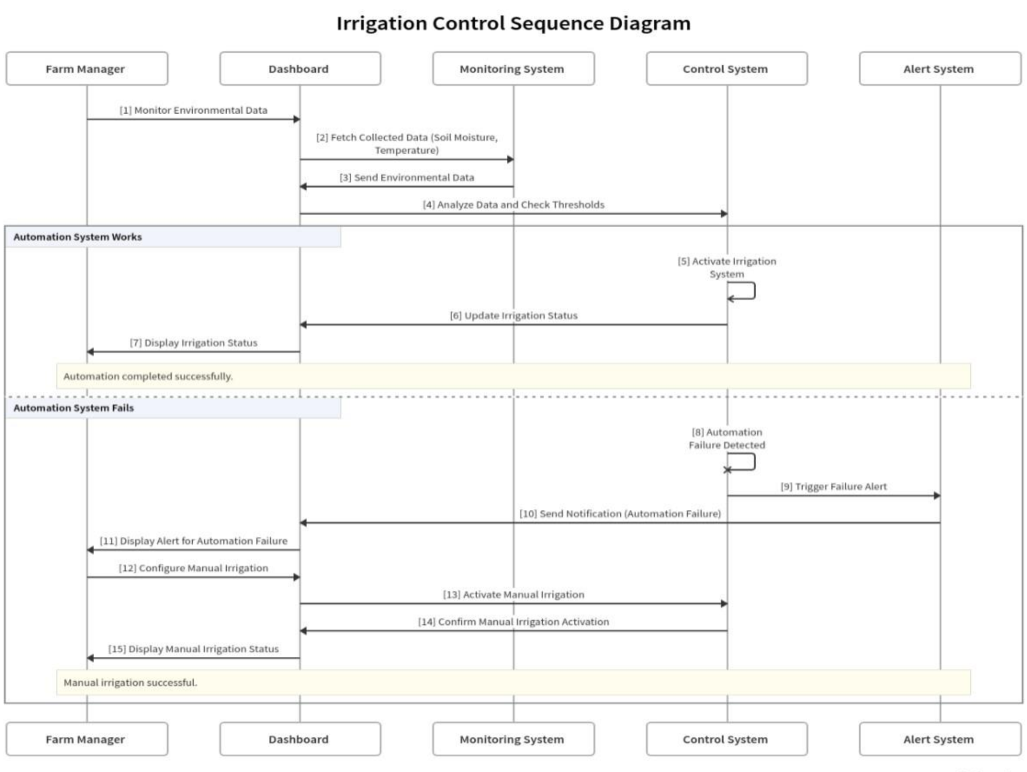


3.4.3 Data Flow Diagram

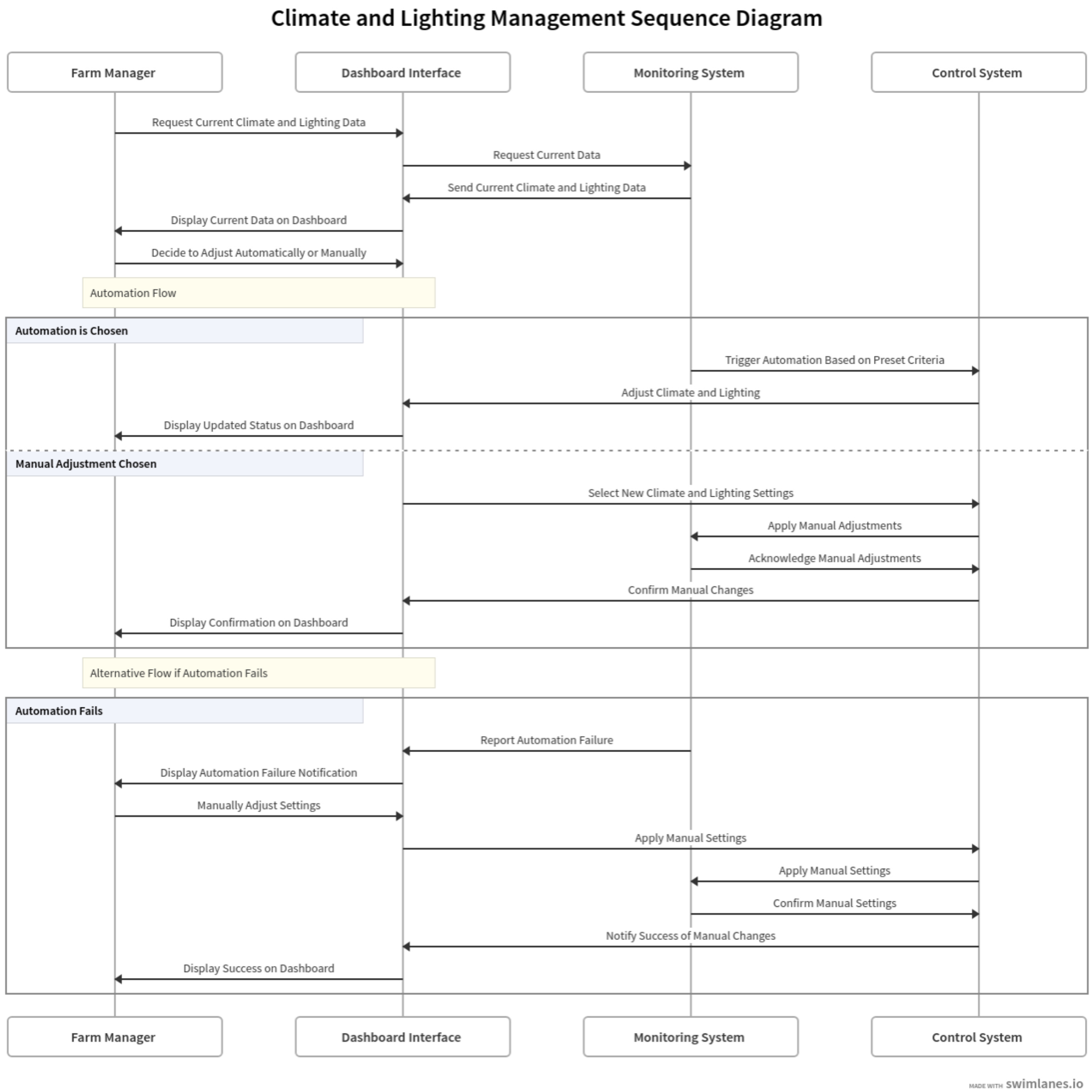


3.4.4 Sequence Diagram

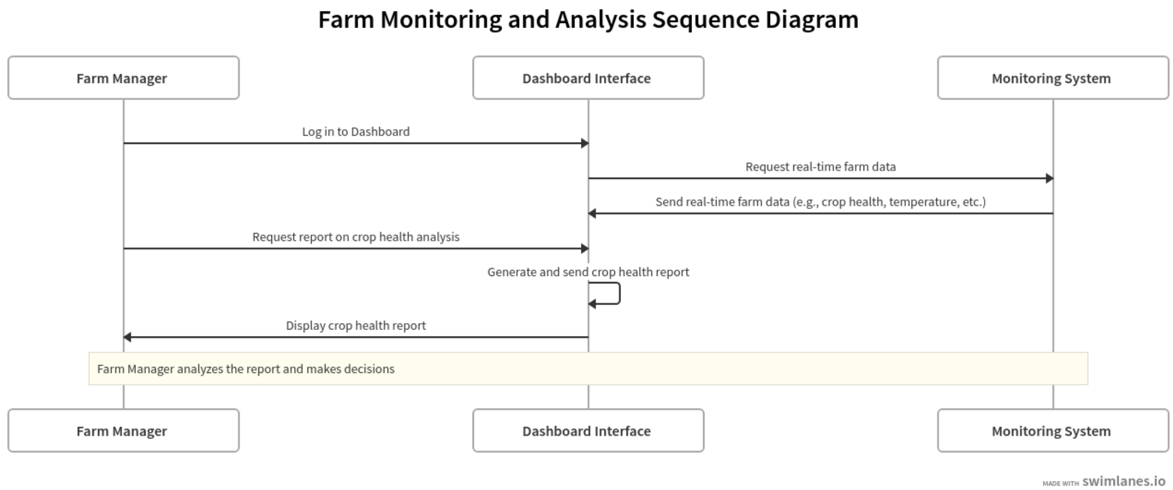
3.4.4.1 Control Irrigation



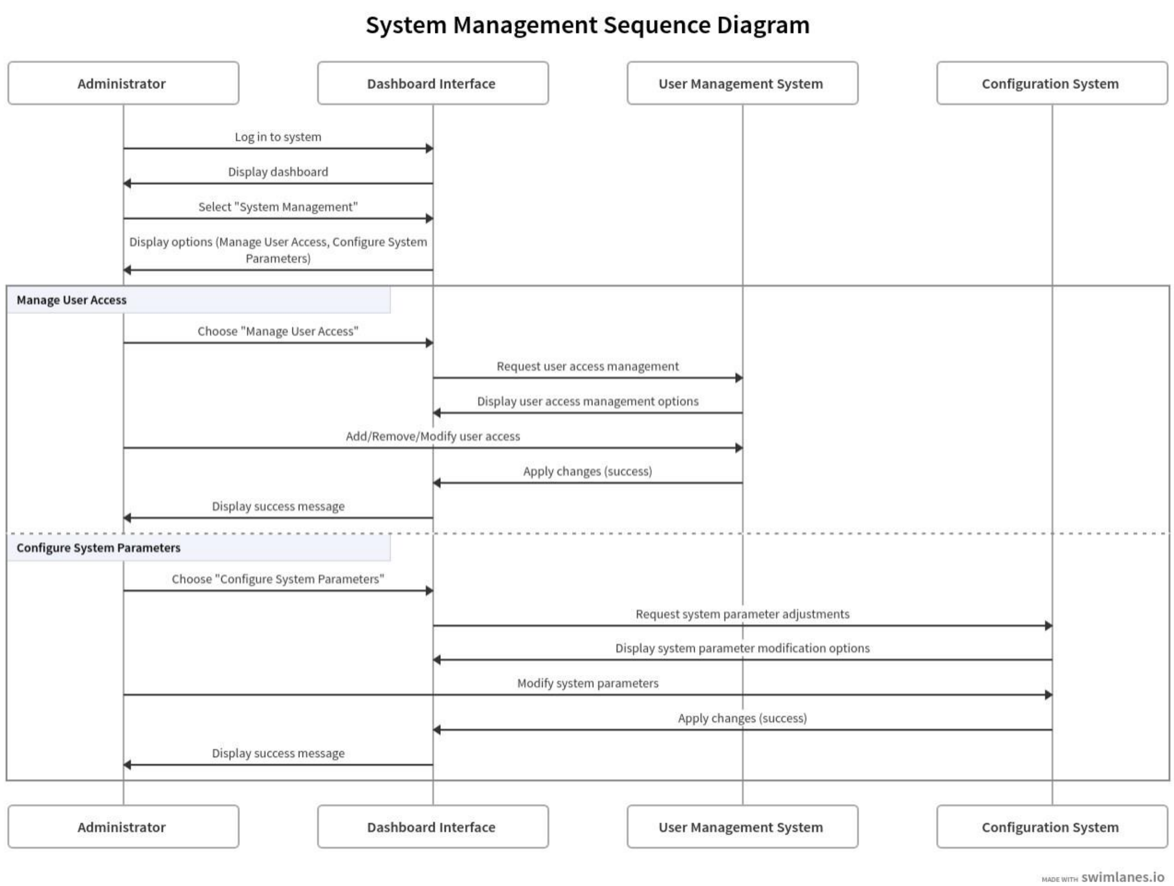
3.4.4.2 Climate and Lighting Management



3.4.4.3 Farm Monitoring and Analysis



3.4.4.4 System Management



3.5 Summary

This chapter provides a detailed analysis of the Farm Automation and Monitoring System, focusing on system functionalities, actor interactions, and process flows. It lays the groundwork for understanding the system's structure, behavior, and data flow.

Chapter 4

System Design

Chapter 4: System Design

In this chapter, we can design the system considering the determined requirements. In section 4.1, we provide the class diagram which includes the classes of the system, their attributes, operations, and relations. Then we provide the design of database tables in section 4.2. The design of the mobile application is included in section 4.3. And the schematic diagram in section 4.4.

4.1 Class Diagram

A class diagram describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among objects. Figure 5.1 shows the class diagram that includes classes representing all system components.

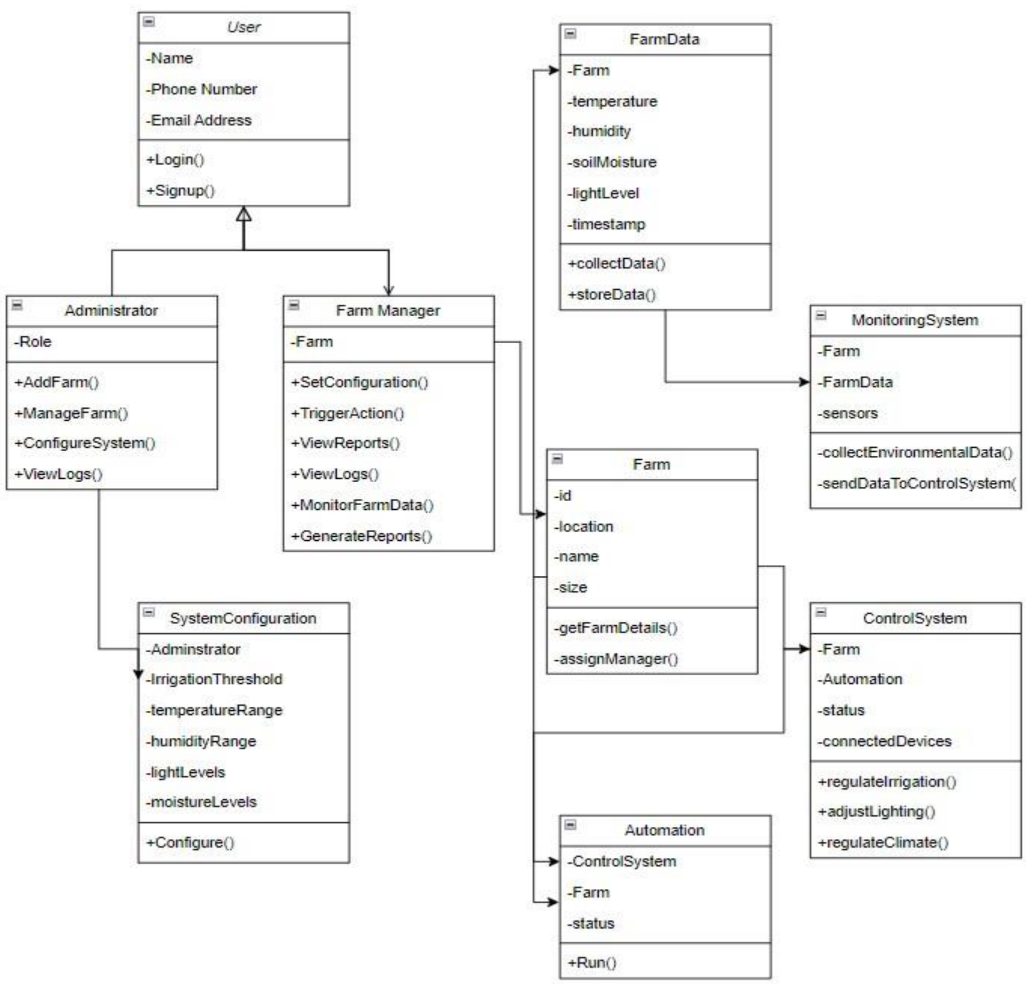


Figure 4.1 - Class diagram

4.2 Data base Design or Algorithms

ERD (Entity Relationship Diagram) is a data modeling technique has graphically illustrated an information system's entities and the relationship between those entities. This ERD show 6 entities: (Users, schedules, inputs, actions, settings, and results) and the relationship between them. Each entity contains attributes with identifying the foreign key and the primary key.

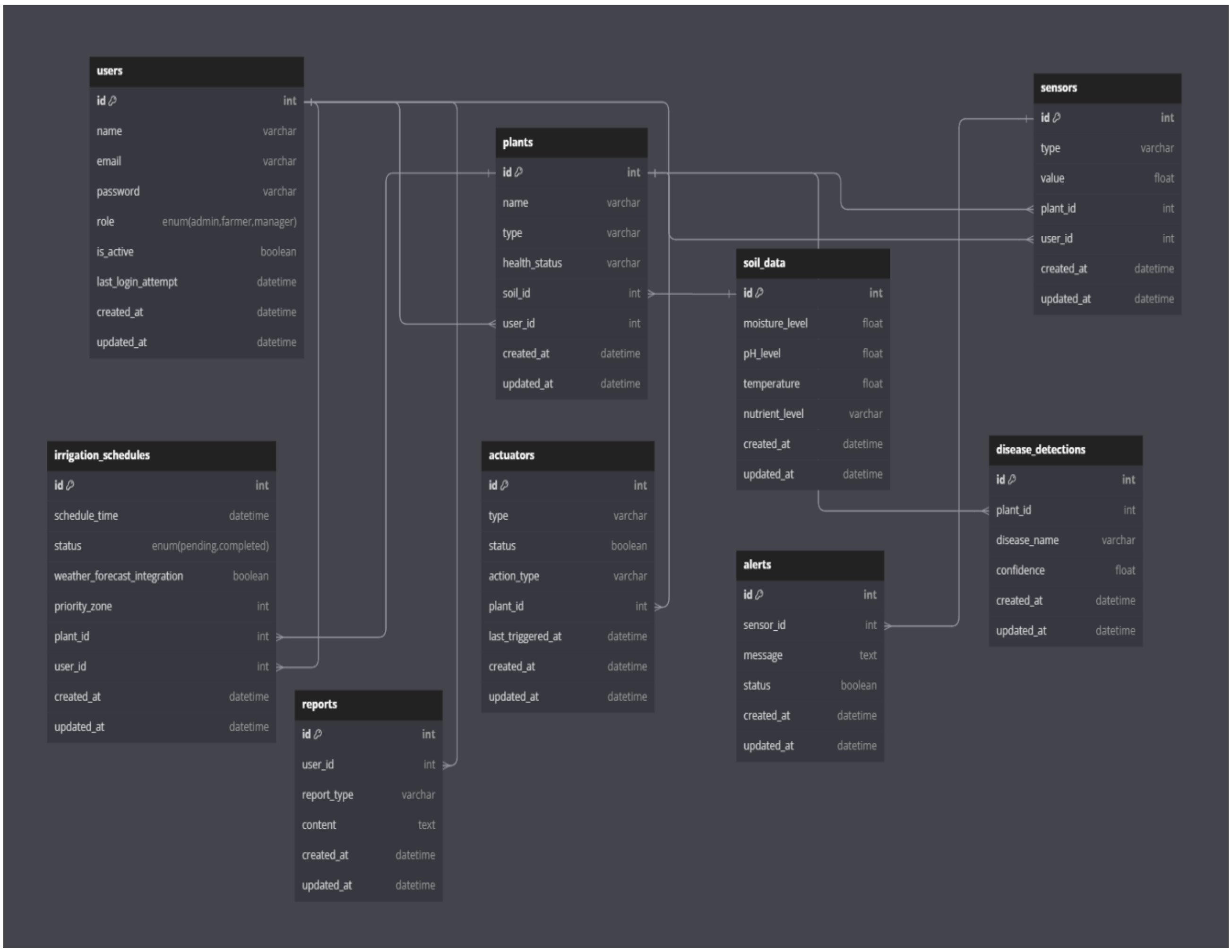


Figure 4.2 - Database design

4.3 Interface Design

Sign-Up Screen

Description:

This screen allows new users to sign up for the Smart Farm system. It includes fields for entering name, email, password, and confirming the password.

Key Features:

- Secure sign-up process with email and password.
- Option to log in for existing users.

Design Considerations:

- Simple and secure sign-up process.
- Clear call-to-action buttons for sign-up and login.
- Minimalistic design to focus on functionality.

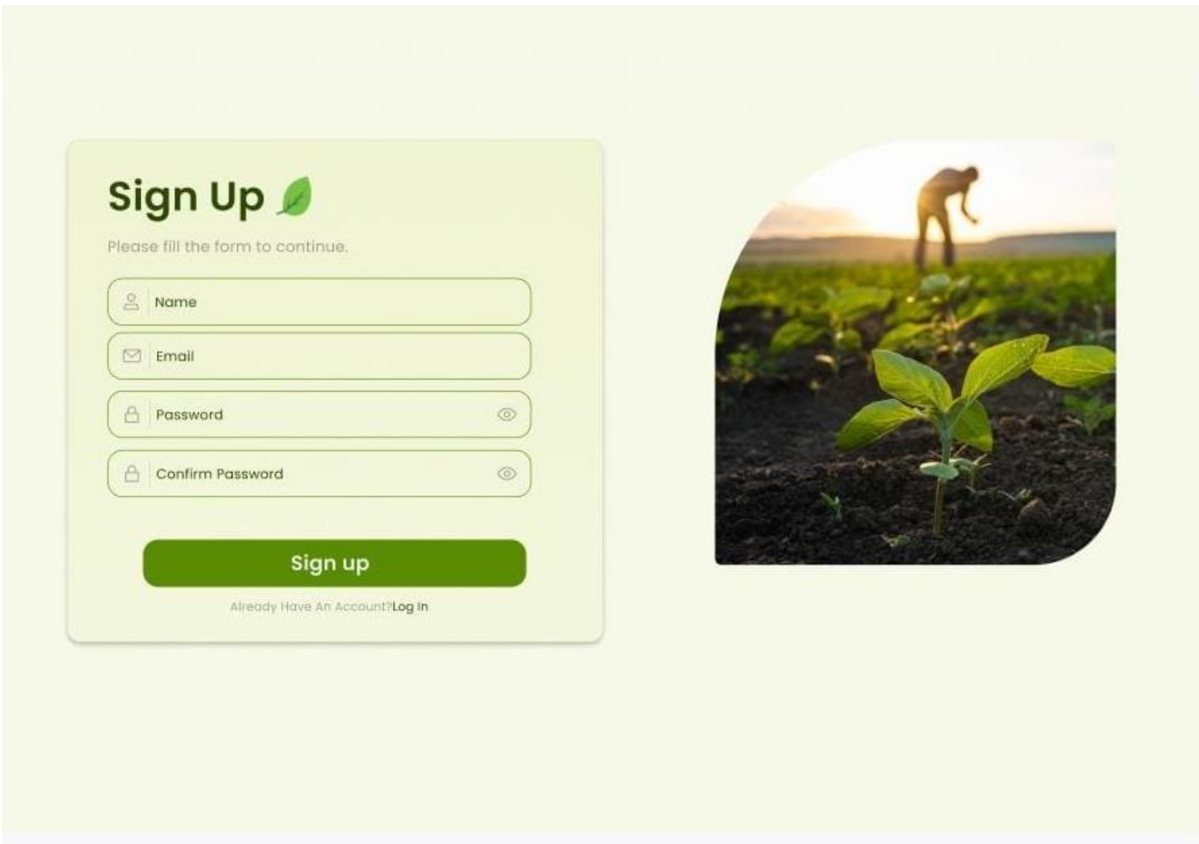


Figure 4.3 - Sign-up screen

Login Screen

Description:

This screen allows users to log in to the Smart Farm system. It includes fields for entering email and password, along with options for signing up or recovering a password.

Key Features:

- Secure login with email and password.
- Option to sign up for new users.
- Forgot password functionality.

Design Considerations:

- Simple and secure login process.
- Clear call-to-action buttons for login and sign-up.
- Minimalistic design to focus on functionality.

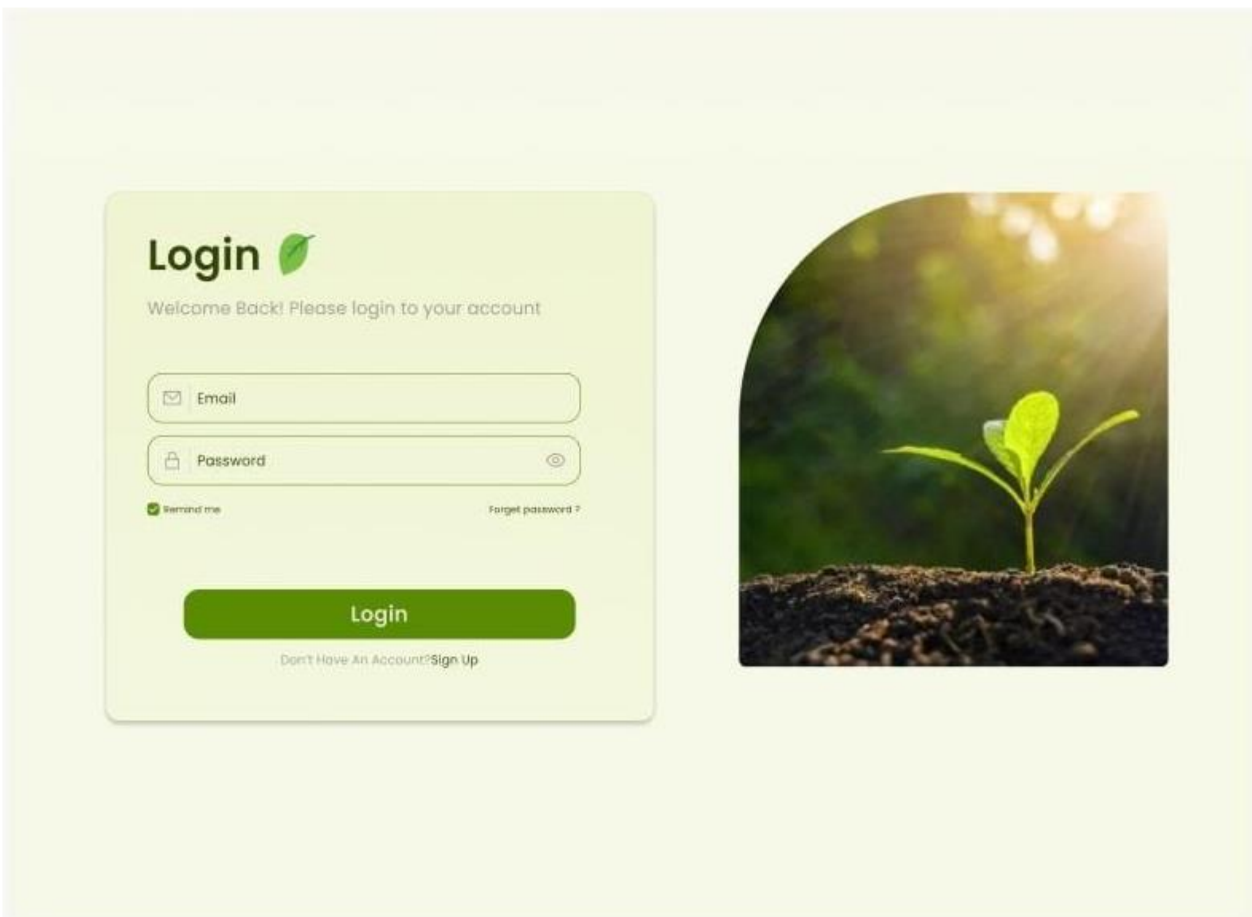


Figure 4.4 - Log-in screen

Home Page

Description:

This screen serves as the home page for the Smart Farm system. It provides an overview of the system's mission, features, and benefits. The interface includes navigation options for accessing different sections of the system.

Key Features:

- Overview of the system's mission and features.
- Navigation menu for accessing different sections.
- Call-to-action button to get started.

Design Considerations:

- Engaging and informative layout to attract users.
- Clear navigation options for easy access to features.
- Use of visuals and text to communicate the system's benefits.

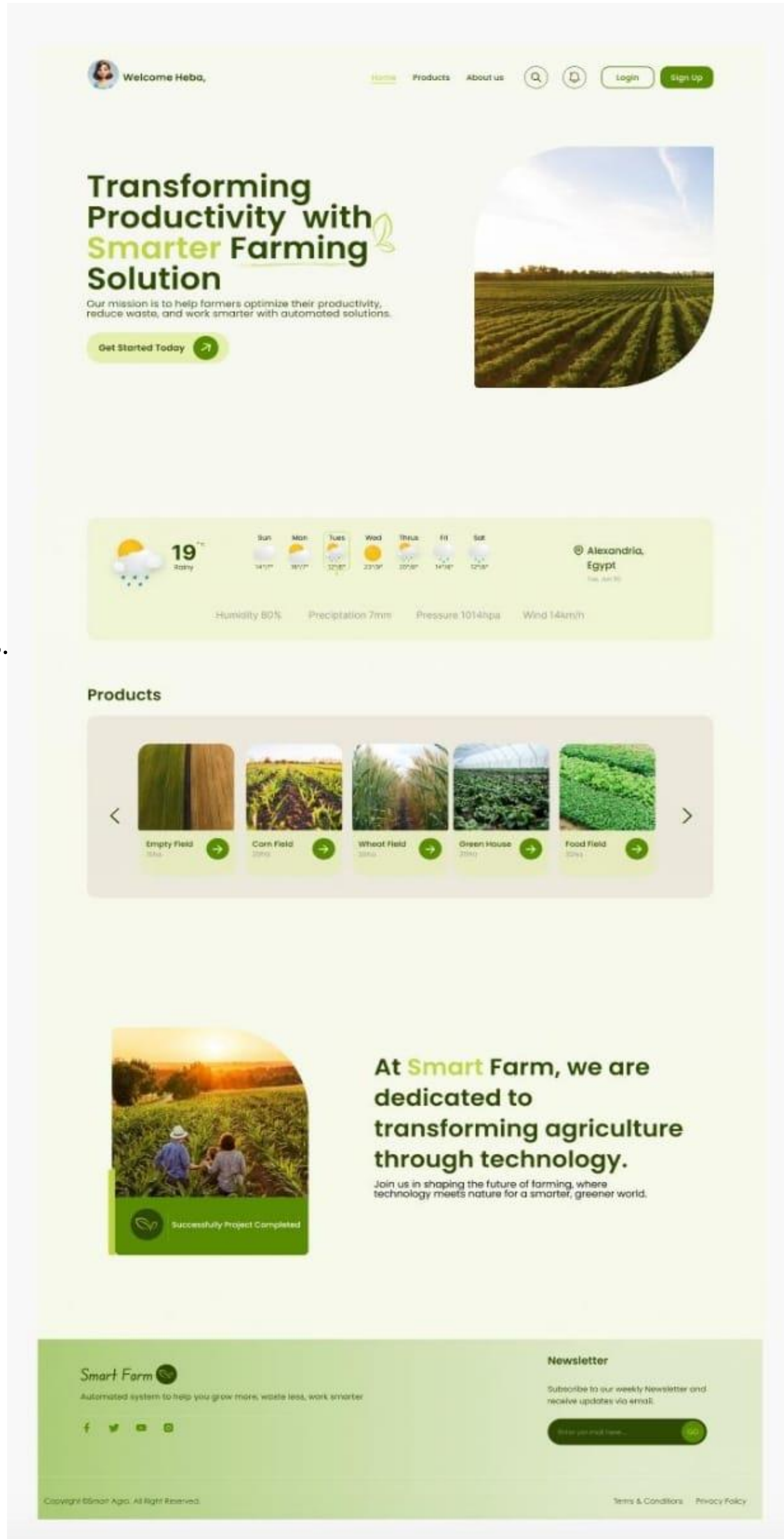


Figure 4.5 - Homepage screen

Smart Farm Dashboard

Description:

This screen serves as the main dashboard for the Smart Farm system. It provides an overview of all crops, including their health, growth rate, and harvest dates. The interface also includes navigation options for crop management, soil and water monitoring, weather, task management, and analytics.

Key Features:

- Overview of all crops and their status.
- Navigation menu for accessing different sections of the system.
- Visual representation of crop health and growth rates.

Design Considerations:

- Centralized dashboard for easy access to all features.
- Use of charts and graphs for data visualization.
- Intuitive navigation menu for seamless user experience.

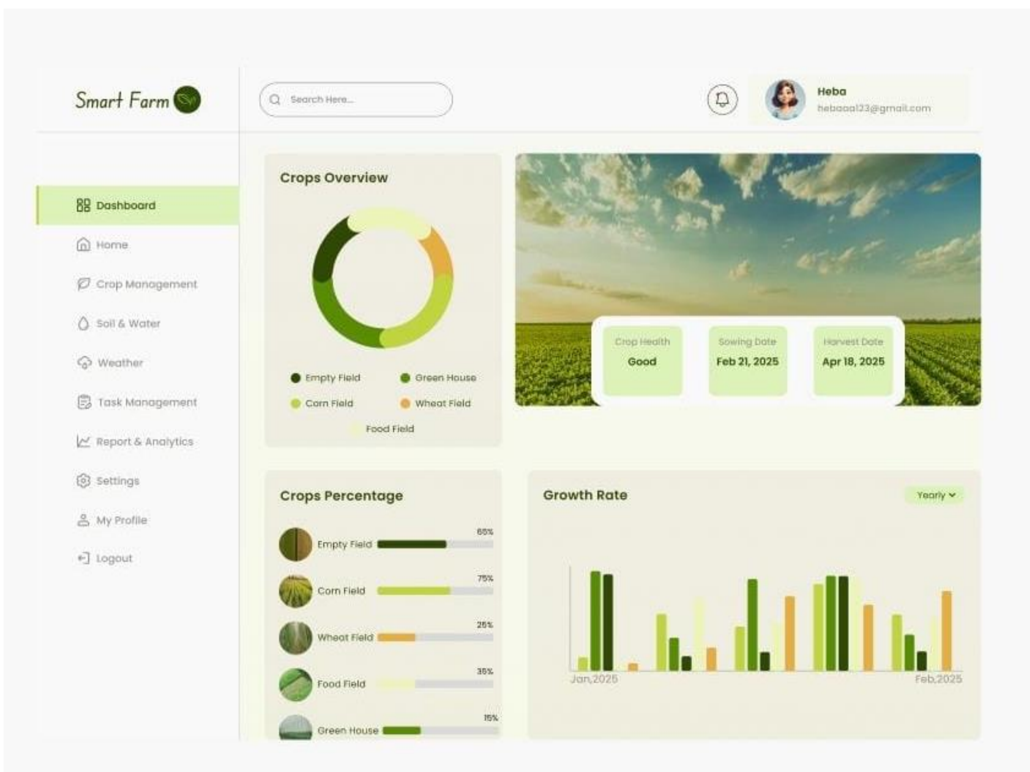


Figure 4.6 - Dashboard screen

Weather Overview Screen

Description:

This screen provides a comprehensive overview of the current weather conditions in Alexandria, Egypt. It displays key metrics such as temperature, air quality, wind speed, humidity, and precipitation. The interface also includes a weekly weather forecast to help users plan agricultural activities.

Key Features:

- Current weather conditions (temperature, air quality, wind speed, humidity, precipitation).
- Weekly weather forecast.
- Highlights section with detailed weather metrics.

Design Considerations:

- Clean and intuitive layout for easy readability.
- Use of icons and visual elements to represent weather conditions.
- Color-coded indicators for air quality and other metrics.

Figure 4.7 Weather Overview screen

Greenhouse Monitoring Screen

Description:

This screen monitors the environmental conditions inside a greenhouse. It displays key metrics such as plant health, water depth, soil moisture, humidity, and temperature. The interface also includes a task management section to track progress on agricultural tasks.

Key Features:

- Real-time monitoring of greenhouse conditions.
- Task management with progress tracking.
- Growth rate and historical data visualization.

Design Considerations:

- Clear separation of metrics and tasks for easy navigation.
- Progress bars and visual indicators for task completion.
- Consistent color scheme to represent different metrics.

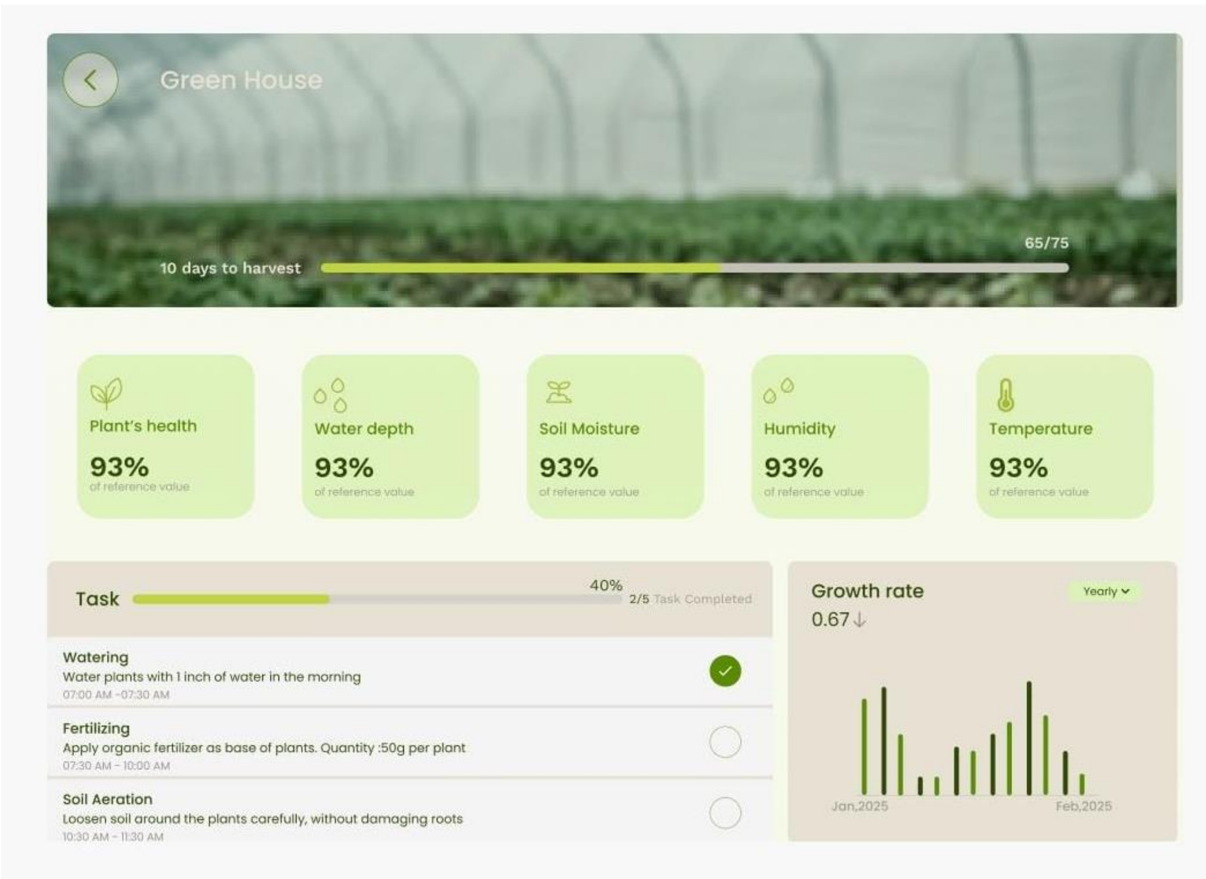


Figure 4.8 Greenhouse Monitoring screen

Field Monitoring Screens (Wheat, Corn, and Empty Fields)

Description:

These screens monitor the conditions of different fields (wheat, corn, and empty fields). They display metrics such as plant health, water depth, soil moisture, humidity, and temperature. The interfaces also include task management sections to track progress on agricultural activities.

Key Features:

- Real-time monitoring of field conditions.
- Task management with progress tracking.
- Growth rate and historical data visualization**.**

Design Considerations:

- Clear separation of metrics and tasks for easy navigation.
- Progress bars and visual indicators for task completion.
- Consistent color scheme to represent different metrics across all field types**.**

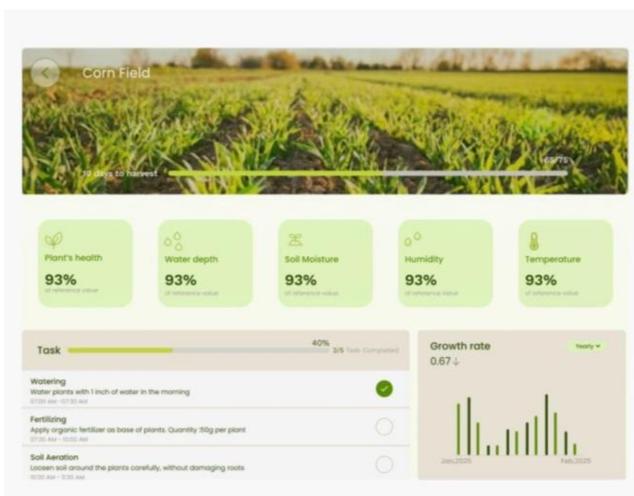


Figure 4.9.1 Corn screen

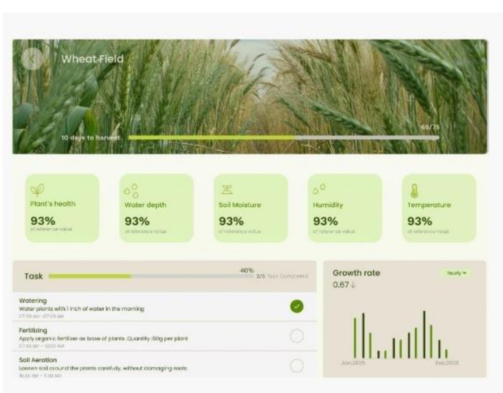


Figure 4.9.2 wheat screen

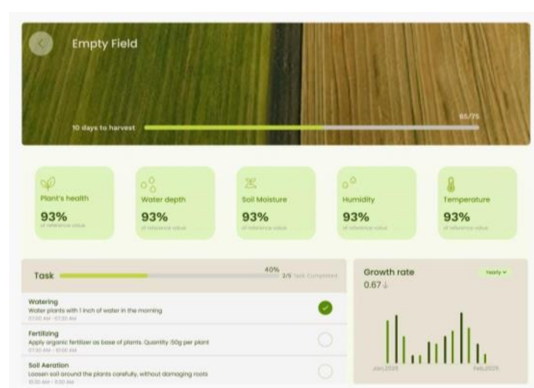


Figure 4.9.3 wheat screen

4.4 Schematics for Embedded System Design

The schematic diagram represents the hardware design of the embedded system used in the Smart Farm project. It includes various components such as sensors, actuators, microcontrollers, and communication modules that work together to monitor and control agricultural operations. The schematic provides a visual representation of how these components are interconnected to form a functional system.

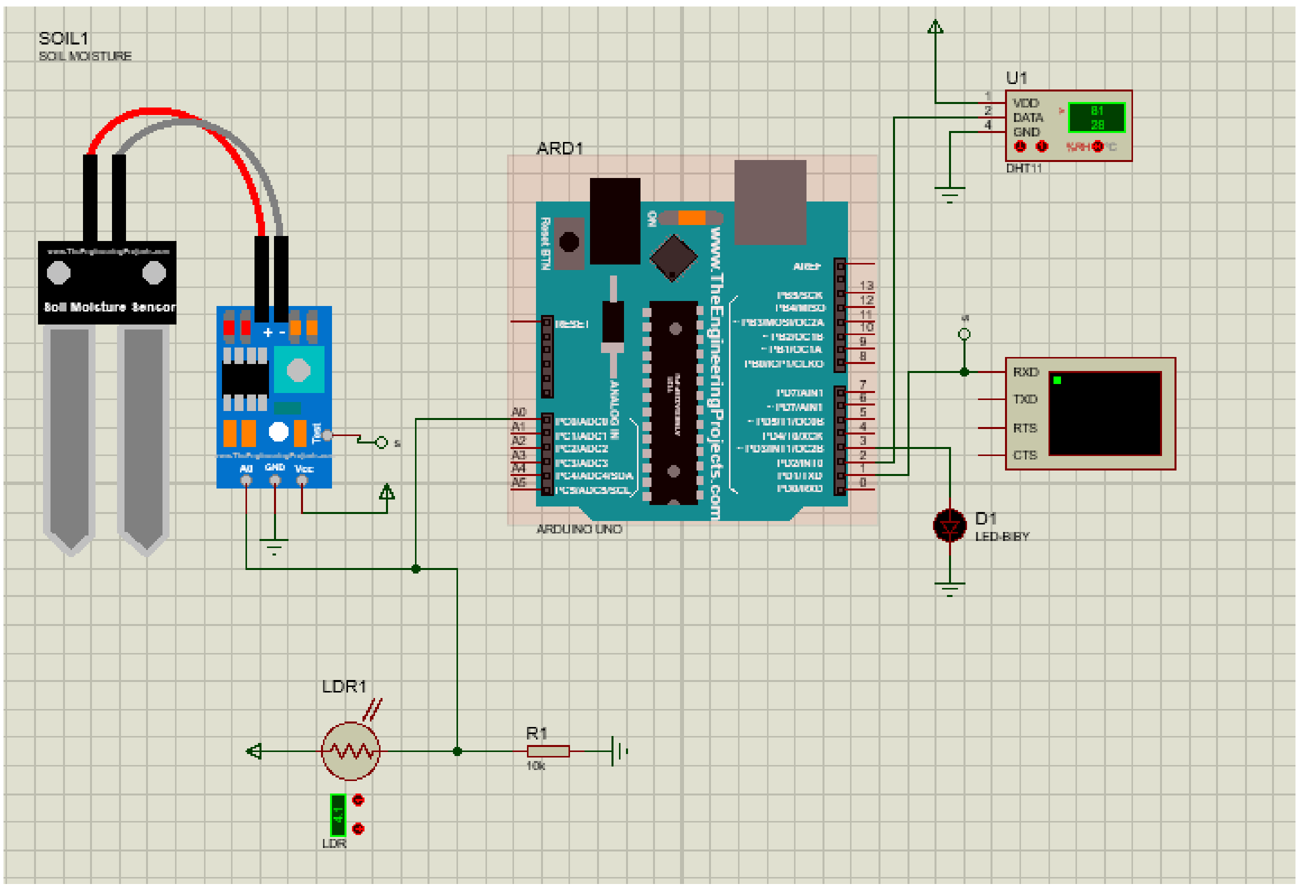


Figure 4.10 Schematics for Embedded System Design

4.5 Summary

In this chapter we make the reader take a look at our system design. Section 5.1 provides a class diagram. We also designed in Section 5.2 the tables of the database. And we showed the reader the interface design of the application with a description of its usage in Section 5.3. And Section 5.4 focuses on the hardware design of the embedded system, which forms the backbone of the Smart Farm project. The schematic diagram illustrates the connections between various components. We can now start the implementation process. In the next chapter, we will map our system design to implementation.