BSE25-28

Smart Cloud-Based Soil Advisor

Software Design Document

Name (s):

NAME	REGISTRATION NUMBER
PRAJAPATI HELI ANILKUMAR	21/X/20194/PS
MUGOYA MOSES	21/U/10447/PS
MPAGI NELLY	21/U/10649/PS
MUHANGI ELIODA	22/U/6342

Lab Section:

Workstation:

Date: 05/02/2025

Contents

1 INTRODUCTION 5
1.1 Purpose 5
1.2 Scope 5
1.2.1 Goal 5
1.2.2 Objectives 5
1.2.3 Benefits 6
1.3 Document Overview 6
1.4 Reference Materials 6
1.5 Definitions and Acronyms 7
2 SYSTEM OVERVIEW 8
3 SYSTEM ARCHITECTURE 9
3.1 Architectural Design 9
3.2 Decomposition description 11
3.2.1. Sensor Layer 11
3.2.2. Communication Layer 11
3.2.3. Processing Layer 12
3.2.4. Application Layer 12
3.3 Design Rationale 14
4 DATA DESIGN 16
4.1 Data Description 16
4.1.1. User management module 16
4.1.2. Farm Management Module 16
4.1.3. Device Management Module 16
4.1.4. Sensor Data Module 17
4.1.5. Crop and Recommendation Module 17
4.2 Data Dictionary 17
5 COMPONENT DESIGN 22
5.1 Pseudocode for Advisor 22
5.2 Pseudocode for Recommendations 24
5.3 Pseudocode for the Notify function 24

6 HUMAN INTERFACE DESIGN 26
6.1 Overview of User Interface 26
6.1.1. User Registration Interface 26
6.1.2. Farm Registration Interface 26
6.1.3. Device Registration Interface 26
6.1.4. Dashboard Interface 27
6.1.5. Farm Dashboard interface 27
6.1.6. AI Advisor Agent Interface 27
6.2 Screen Images 27
6.2.1. Account Registration Form 27
6.2.2. Login Form 28
6.2.3. Device Registration Screen Form 29
6.2.4. Dashboard Screen 30
6.2.5. Farm Dashboard Screen 30
6.2.6. Device Management Screen 31
6.2.7. Farm Dashboard with Advisory Screen 32
6.2.8. Soil NPK Sensor 32
6.2.9. Soil Temperature Sensor 33
6.2.10. Soil Moisture Sensor 33
6.2.11. LCD Screen 34
6.2.12. ESP32 Microcontroller 34
6.3 Screen Objects and Actions 34
7 REQUIREMENTS MATRIX 38
8 APPENDICES 39

Table of Figures

Figure 3.1: Architectural Design of the Smart cloud-based soil advisor system 9
Figure 3.2: Embedded System Architecture for Smart Cloud-Based Soil Advisor...................... 10
Figure 3.3: Structural Decomposition Diagram for Smart cloud-based soil advisor system........ 13
Figure 4.1: Entity relationship Diagram. 17
Figure 6.1: Account Registration Form 28
Figure 6.2: Login Form 29
Figure 6.3: Device registration screen form 29
Figure 6.4: Dashboard screen 30
Figure 6.5: Farm dashboard screen 31
Figure 6.6: Device Management Screen 32
Figure 6.7: Farm dashboard with Advisory screen 32
Figure 6.8: Soil NPK sensor for capturing (Nitrogen, phosphorous, and potassium) 33
Figure 6.9: Soil Temperature sensor 33
Figure 6.10: Soil moisture sensor 33
Figure 6.11: LCD screen 34
Figure 6.12: ESP32 micro-controller 34

Index of Tables

Table 1: Definitions and Acronyms 7
Table 2: User Information table 18
Table 3: Projects Table (Farms) 18
Table 4: Device Table (Registered sensors) 18
Table 5: Farm_Crop Table 19
Table 6: Sensor_Data Table 19
Table 7: Crops_table. 20
Table 8: Recommendation_Table 20
Table 9: Functions and Function Parameters 21
Table 10: Screen Objects and Actions 34
Table 11: Requirements Matrix Table 38

1 INTRODUCTION

1.1 Purpose

This Software Design Document specifies the architecture and system design of the Smart cloud-based soil advisor. It provides the software development team with a comprehensive understanding of the system and its implementation. The intended audience includes software developers, software architects, software testers, project managers, and agricultural experts.

1.2 Scope

The Smart cloud-based soil advisor is a web-based and IoT-integrated platform designed to collect, analyse, and provide recommendations on soil nutrient levels. The system captures data from soil sensors (such as moisture, temperature, and NPK sensors), processes the data in the cloud, and provides advisory insights on suitable crops for planting.

1.2.1 Goal

To develop an intelligent, real-time system that enables farmers and agricultural organizations to monitor soil health, optimize crop selection, and improve farm productivity based on data-driven insights.

1.2.2 Objectives

1. To design a system that integrates IoT sensors for real-time soil data collection.
2. To develop a cloud-based platform that processes, analyses soil data and deploys the interface.
3. To provide actionable insights and recommendations to farmers based on soil nutrient levels.
4. To enable users to create multiple farm projects and register soil-monitoring devices.

1.2.3 Benefits

1. **Enhanced Productivity**: Farmers can make informed decisions on crop selection and soil management.
2. **Real-Time Monitoring:** The system shall make continuous tracking of soil health parameters improves farm efficiency.
3. **Sustainable Agriculture:** The system shall enable optimal use of fertilizers and resources and minimize environmental impact.
4. **Data-Driven Insights**: The system offers graphical reports and trend analysis to aid planning.

1.3 Document Overview

This design document consists of the following chapters: -

1. **Introduction**: Describes the purpose, system scope, and document overview.
2. **System Overview**: Provides a high-level description of the system's functionality.
3. **System Architecture**: Details the architectural design, decomposition description, and design rationale.
4. **Data Design** – Covers the data description and data dictionary.
5. **Component Design**: Provides an overview of the system's core components.
6. **Human Interface Design**: Includes user interface, screen images, and screen objects together with their actions.
7. **Requirements Matrix**: provides a cross-reference that traces components and data structures to the system requirements.
8. **Appendices**: Additional references and supporting materials.

1.4 Reference Materials

[1] J. F. Mortensen, "Declining Soil Fertility in Sub-Saharan Africa: A Review," Agriculture, Ecosystems & Environment, vol. 174, pp. 75–86, 2013.doi: 10.1016/j.agee.2013.05.014

[2] L. Zingore, "Addressing Soil Fertility Constraints in Africa: The Role of Integrated Soil Fertility Management," Nutrient Cycling in Agroecosystems, vol. 88, no. 1, pp. 1–15, 2010.doi:10.1007/s10705- 010-9363-1

[3] H. K. Adhikari and S. Chakraborty, “Soil Quality Monitoring: A Sensor-Based Approach,” Sensors, vol. 21, no. 7, pp. 234–245, 2021.doi:10.3390/s21072345

[4] P. A. Sánchez, “Soil Fertility and Hunger in Africa,” *Science*, vol. 295, no. 5562, pp. 2019–2020, 2002.doi:10.1126/science.1065256

1.5 Definitions and Acronyms

Table 1: Definitions and Acronyms

ACRONYMS AND ABBREVIATIONS	MEANING
CBSA	Smart cloud-based soil advisor
IoT	Internet of Things
API	Application Programming Interface
NPK	Nitrogen, Phosphorus, Potassium (key soil nutrients)

2 SYSTEM OVERVIEW

Soil health is a critical factor in agricultural productivity [1], yet many farmers lack real-time data to make informed decisions about nutrient management. The Smart cloud-based soil advisor aims to address this challenge by providing a web-based and IoT-enabled platform for real-time soil monitoring, data analysis, and advisory services [2], [3].

The CBSA system integrates IoT sensors to collect key soil parameters such as moisture, temperature, and nutrient levels (NPK). This data is transmitted to a cloud-based database, where it is processed and analysed. The system then provides farmers and agricultural organizations with graphical insights, trend analysis, and recommendations on crop selection and soil treatment based on historical and real-time data.

To address this challenge, we plan to develop a web-based application that will collect soil data from various sensors, analyse it to assess soil nutrient levels, provide graphical insights on soil health trends, and offer recommendations on suitable crops and fertilizer application [4]. The Smart cloud-based soil advisor) will primarily support farmers and agricultural organizations, especially in rural areas where access to advanced soil testing equipment and expert guidance is limited.

Key Features of Smart Cloud-Based Soil advisor

1. **Real-Time Soil Monitoring** – Captures and processes soil moisture, temperature, and nutrient levels from IoT sensors.
2. **Data Visualization and Reports** – Displays soil data trends, analysis, and graphical reports to aid decision-making.
3. **Crop Suitability Advisory** – Recommends suitable crops based on soil conditions and nutrient composition.
4. **Device Integration** – Supports multiple IoT sensor devices for automated data collection.
5. **Automated Alerts & Notifications** – Notifies users about critical soil conditions, such as low moisture or nutrient deficiency.
6. **Cloud-Based Data Processing** – Ensures scalable and secure storage of soil health records with predictive analytics.

3 SYSTEM ARCHITECTURE

3.1 Architectural Design

Major components of the system

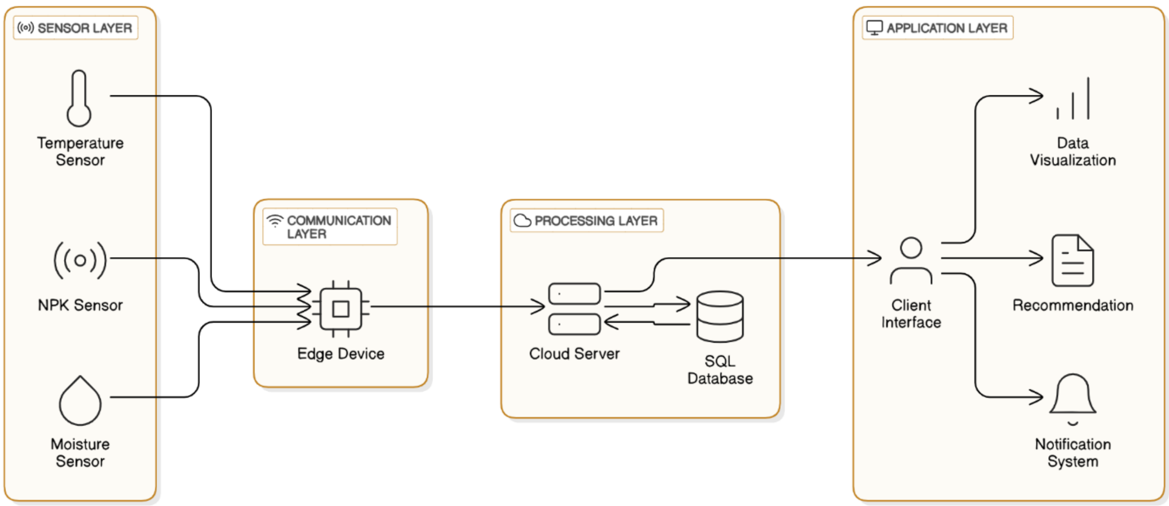

Figure 3.1: Architectural Design of the Smart cloud-based soil advisor system

Figure 3.1 contains the architecture specifically designed to emphasise real-time data collection and analysis, providing farmers with actionable insights that can improve crop yields and promote sustainable farming practices.

The system collects vital soil data using sensors that measure nutrient levels (NPK), moisture, and temperature. These parameters are key for managing fertilization, irrigation, and optimal planting schedules.

Data from the sensors is wirelessly transmitted to the cloud via an edge device (ESP32), ensuring reliable connectivity even in remote locations by buffering data when connectivity is intermittent.

In the cloud, the data undergoes processing, where machine learning is applied for error correction to ensure the accuracy of the sensor readings. The corrected data is stored in an SQL database for further analysis.

The user interface provides farmers with crop recommendations, notifications about critical soil conditions, and visual data insights through charts and graphs. It is designed to be user-friendly, supports multiple languages, and works well in areas with limited internet access.

The embedded system architecture for the Smart Cloud-Based Soil Advisor

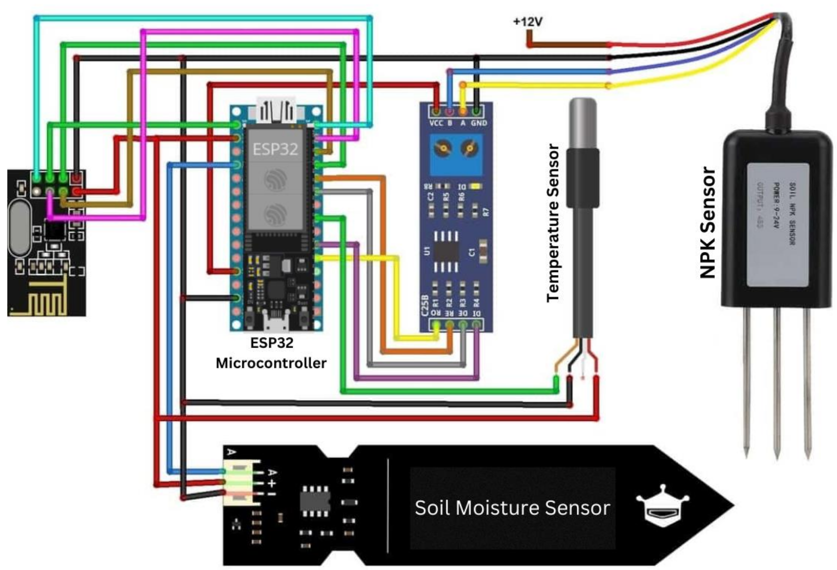

Figure 3.2: Embedded System Architecture for Smart Cloud-Based Soil Advisor.

The above figure 3.2 illustrates the embedded system architecture for the Smart Cloud-Based Soil Advisor which is designed to monitor and analyze critical soil parameters using a network of interconnected sensors and communication modules. The figure illustrates the key components of the system:

ESP32 Microcontroller: The central processing unit of the system, responsible for receiving data from the various sensors and transmitting it to the cloud server for further analysis and storage.

Soil Moisture Sensor: This sensor detects and measures the water content in the soil, providing vital information to assess soil moisture levels and prevent over-irrigation or drought conditions.

Temperature Sensor: Monitors the temperature of the soil, ensuring optimal environmental conditions for crop growth and detecting any abnormalities in the soil temperature that could affect plant health.

Soil NPK Sensor: Measures the concentration of essential nutrients in the soil, specifically Nitrogen (N), Phosphorus (P), and Potassium (K), which are critical for plant nutrition and growth.

3.2 Decomposition description

The Smart cloud-based soil advisor for smallholder farmers in Uganda is structured into multiple layers, each performing distinct yet interconnected functions to ensure the efficient collection, transmission, processing, and visualization of soil data. The decomposition of the system into layers allows for modularity and ease of maintenance. The following is a detailed description of the major components and features within each layer.

3.2.1. Sensor Layer

The Sensor Layer forms the foundation of the system by collecting essential soil data, which is critical for decision-making in agriculture. This layer consists of various sensors that measure key soil parameters:

- **Temperature Sensor Feature:** This feature continuously monitors the soil temperature, an important factor for crop health and growth. Soil temperature influences nutrient uptake and plant metabolism, and by tracking this data, farmers can adjust planting times and other activities.
- **Moisture Sensor Feature:** This sensor tracks the soil's moisture content, ensuring that the water needs of crops are met. It helps in determining the optimal irrigation schedules, preventing under- or over-watering.
- **Nutrient Sensor (NPK) Feature:** This sensor measures the levels of Nitrogen (N), Phosphorus (P), and Potassium (K) in the soil. These nutrients are essential for plant growth, and the data provided by this sensor helps farmers apply the right fertilizers in the correct quantities, avoiding wastage and improving crop yield.

3.2.2. Communication Layer

The Communication Layer is responsible for transmitting the raw data collected by the sensor layer to the processing layer. It ensures efficient data transmission from the field to the cloud. This layer includes:

- **Edge Device/ESP32 Feature:** The ESP32 is an edge device that acts as an intermediary between the sensors and the cloud server. It collects sensor data, performs basic preprocessing (such as filtering or compression), and transmits the data to the cloud server via Wi-Fi or other communication protocols like LoRaWAN. The ESP32 is selected for its low power consumption, cost-effectiveness, and versatility in IoT applications.

3.2.3. Processing Layer

Raw sensor data is sent to the cloud server, where machine learning models are used for error correction. This ensures that faulty or inaccurate sensor readings are detected and corrected before analysis. The system cleans and normalizes the data, which is then stored in an SQL database for further processing and retrieval. This layer comprises the following components:

- **Cloud Server Feature:** The cloud server serves as the primary processing hub where data from the edge device is collected, stored, and analysed. The cloud server is responsible for running data processing algorithms and machine learning models to identify patterns in the soil data and provide recommendations based on the analysed information.
- **SQL Database Feature:** This database is used to store the large volumes of data generated by the sensors over time. The SQL database stores historical data on soil conditions, enabling time-series analysis, and provides a structured environment for efficient querying and data management. It ensures data integrity and supports scalable data storage, which is crucial for long-term monitoring and analytics.

3.2.4. Application Layer

The Application Layer is the user-facing layer of the system. It is responsible for interfacing with the end-users (farmers, agricultural advisors) and provides visualization and interaction with the processed data. The application layer is made up of the following features:

- **Client Interface Feature:** This feature is the primary interface through which users interact with the system. The interface is designed to be simple and intuitive, providing farmers with easy access to real-time and historical soil data. The client interface includes:

- **Data Visualization Feature:** This feature provides graphical representations of soil temperature, moisture levels, and nutrient concentrations (NPK). Farmers can view charts, graphs, and trends over time, which helps them make informed decisions regarding irrigation and fertilization.
- **Crop Recommendation:** Based on sensor data, the system suggests suitable crops for the current soil conditions.

- **Notifying System Feature:** This feature notifies users when critical soil data, such as low moisture levels or nutrient deficiencies, are detected. These notifications are delivered via SMS, email, or within the client interface, depending on user preferences.

The structural decomposition diagram below further decomposes the system to show the different system modules and components.

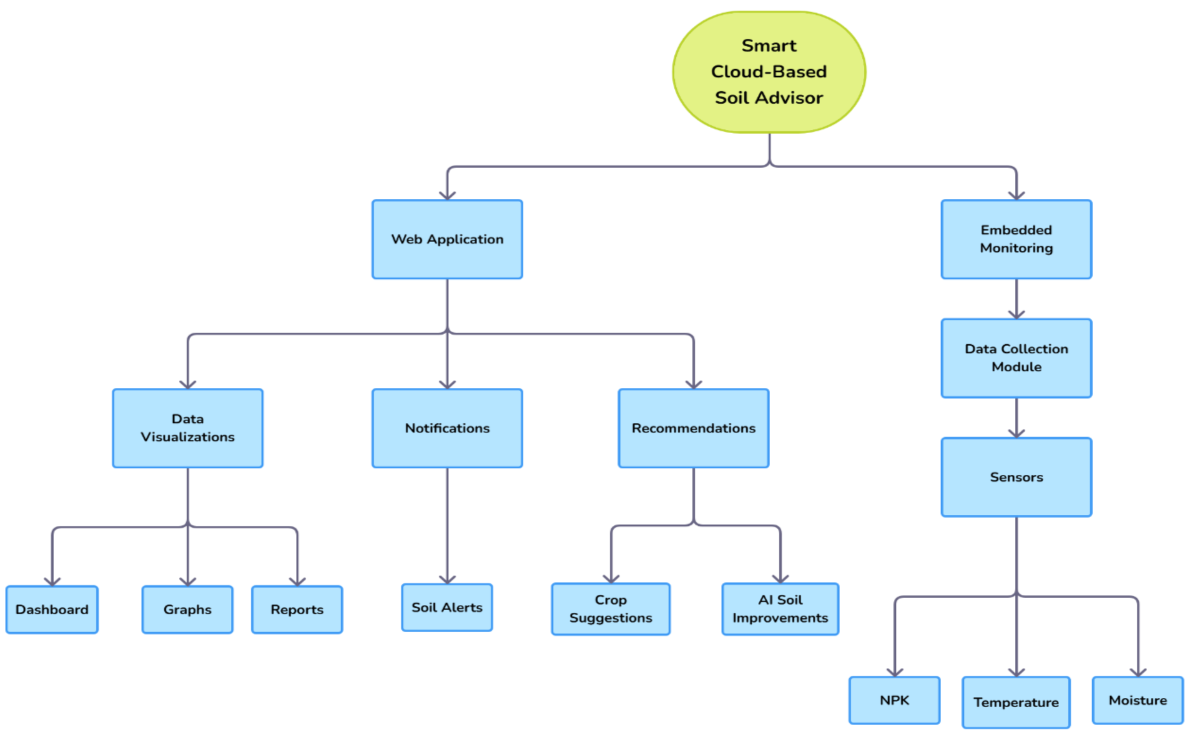

Figure 3.3: Structural Decomposition Diagram for Smart cloud-based soil advisor system

3.3 Design Rationale

The Smart cloud-based soil advisor for farmers in Uganda will adopt a layered architecture, which is a widely used architectural style in system design that organizes software components into layers. Each layer is responsible for handling specific concerns and interacts with the adjacent layers in a controlled and systematic way. The decision to implement a layered architecture for this system is driven by several key advantages, particularly in terms of reusability, modularity, scalability, maintainability, and parallel development. The architecture will be divided into three primary layers: User Interface Layer, Business Logic Layer, and Data Access Layer.

Layered Architecture Overview:

- **User Interface Layer:** The user interface layer represents the topmost layer of the system and is responsible for facilitating interaction between the users and the system. This layer provides a simple, intuitive interface for users to view the results and recommendations generated by the system.
- **Business Logic Layer:** The business logic layer forms the core of the system's functionality. It is responsible for implementing the algorithms, rules, and logic that govern the system's behaviour. In this case, the business logic includes processing sensor data, applying machine learning models, generating recommendations for irrigation, fertilization, and crop selection, and executing notifications based on predefined rules. The Business Logic Layer will handle all the computational tasks, including:
    - Data validation and analysis.
    - Running machine learning models for error reduction.
    - Generating recommendations based on soil conditions.
    - Determining when to trigger notifications for farmers.
- **Data Access Layer:** The data access layer is responsible for managing interactions with the underlying databases or data storage systems. In the context of this system, it will handle data retrieval, storage, and transmission between the sensors and the cloud platform.

Justification for Using Layered Architecture:

- **Component Reusability and Extensibility:** One of the key benefits of layered architecture is its reusability. The components in each layer can be reused across multiple applications or platforms. For example, if the system were to expand to include a desktop application or a mobile app for farmers, the User Interface Layer can be replaced or customized for the new platform, while the Business Logic Layer and Data Access Layer remain unchanged.
- **Parallel Development:** Teams can work simultaneously on different layers, such as user interface, business logic, and data storage, leading to faster development with fewer dependencies.
- **Independent Deployment, Maintenance, and Updates:** Layers can be updated separately without disrupting the entire system, allowing for efficient upgrades, independent maintenance, and reducing system downtime.
- **Testing and Debugging:** Each layer can be tested in isolation, making it easier to identify and resolve bugs, ensuring a more reliable and stable system.

4 DATA DESIGN

4.1 Data Description

This section describes how the system information will be transformed into data structures, and how the data and system entries will be stored, processed, and organized during the system development. The information domain will include the various soil parameters that the Soil Advisor System measures, such as temperature, humidity, NPK levels, pH, and soil moisture. It will also include sensor data, user inputs, external data (e.g., weather), and advisory outputs based on the soil condition.

4.1.1. User management module

This module is responsible for managing user accounts, authentication, and authorization. Users will be able to register, log in, and manage their farms through the system. Each user can own multiple farms, and their credentials are securely encrypted and stored, ensuring access control. The system will track when users were added to help with system auditing and management.

4.1.2. Farm Management Module

The farm management module allows users to register and manage their farms within the system. Each farm is associated with a specific user and can be linked to various crops and IoT devices. The system records key details like the farm's name and location, enabling better monitoring and analysis. Since farms can grow multiple crops, a dedicated structure manages the relationship between farms and crops, ensuring flexibility for different farming practices.

4.1.3. Device Management Module

This module manages IoT devices that collect soil and environmental data from farms. Each device is uniquely identifiable and linked to a specific farm. The system tracks the operational status of each device, allowing users to monitor whether their sensors are active or inactive. This helps in ensuring real-time data collection and troubleshooting faulty devices efficiently.

4.1.4. Sensor Data Module

This module collects, stores, and processes data from IoT devices deployed on farms. Sensors measure key soil and environmental parameters such as moisture levels, temperature, and nutrient levels (nitrogen, phosphorus, potassium). The collected data is linked to the farm it was gathered from, allowing farmers to make data-driven decisions. This module forms the foundation for generating recommendations and insights.

4.1.5. Crop and Recommendation Module

This module provides decision support by suggesting suitable actions based on collected sensor data. It maintains a database of crops with their ideal soil conditions, including required nutrient levels and moisture content. By comparing real-time sensor data against these ideal conditions, the system generates recommendations for improving soil health and optimizing crop yield. These insights help farmers apply the right interventions, such as irrigation adjustments or fertilization, improving productivity and sustainability.

4.2 Data Dictionary

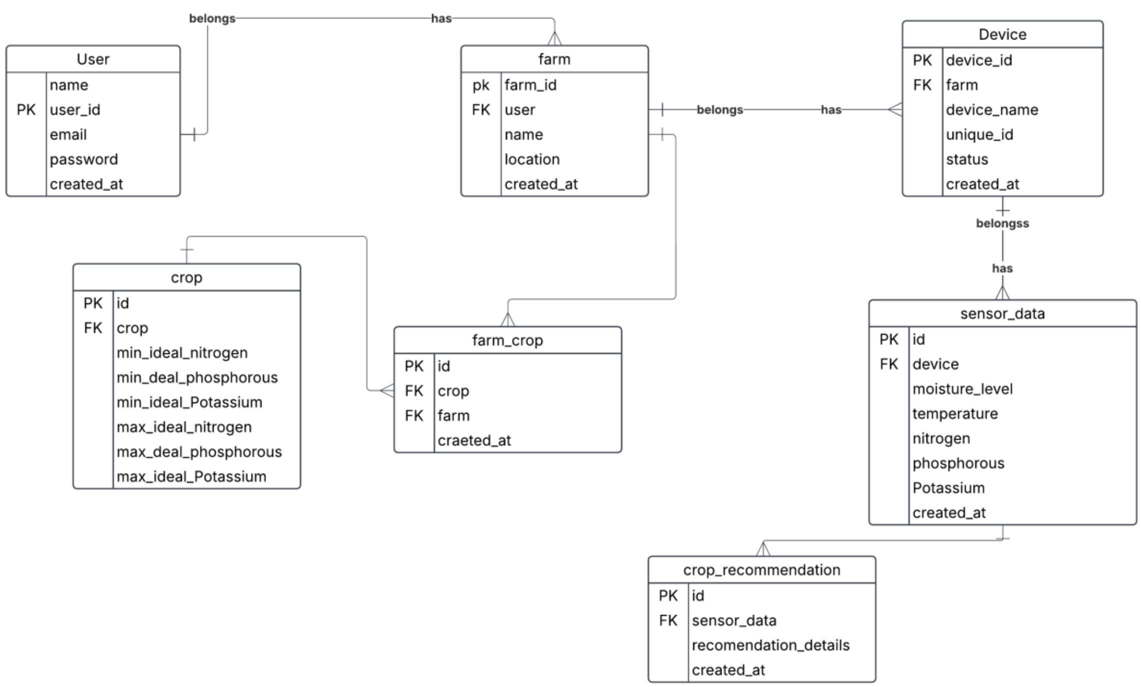

Figure 4.1: Entity relationship Diagram.

i. User Information table

Table 2: User Information table

Field	**Data Type**	**Field size**	**Data Description**
id	INTEGER	25	Primary key for user
Name	VARCHAR	255	Org’s or user’s name
Email	VARCHAR, unique	255	Email for Auth
Password	VARCHAR	255	User password(hashed)
created_at	TIMESTAMP	50	Account creation

ii. Farm Table

Table 3: Projects Table (Farms)

Field	**Data Type**	**Fieldsize**	**Description**
Id	INT	25	Unique primary project
user	INT	25	Reference to the owner
Crop	INT(FK)	25	Reference to crop
farm	VARCHAR	255	project(farm) name
farm_location	TEXT	25	Farm Location
Created_at	TIMESTAMP	20	Date the project was added

iii. Device Table (Registered sensors)

Table 4: Device Table (Registered sensors)

Field	**Data Type**	**Fieldsize**	**Description**
Id	INT(PK)		Primary key device

farm	INT (FK)	10	Reference to the project
device_name	VARCHAR	255	sensor name
Unique_id	VARCHAR	50	Device Id for Authentication and authorisation
status	ENUM(active, inactive)	10	Device_status whether its actively sending data or not
Created_at	TIMESTAMP	20	Date of registration

iv. Farm_Crop Table

Table 5: Farm_Crop Table

Field	**Data Type**	**Fieldsize**	**Description**
Id	INTEGER		Primary key device
crop	VARCHAR	100	Reference to the project
farm	INT(FK)	255	sensor name
Crop	INT(FK)	50	sensor_type

v. Sensor_Data Table

Table 6: Sensor_Data Table

Field	**Data Type**	**Fieldsize**	**Description**
Id	INT	25	Primary key for project
device	INT	20	Reference to the device
moisture	FLOAT	25	Moisture content (%)
Nitrogen	FLOAT	20	Nutrient composition

phosphorous	FLOAT	10	Phosphorus level
potassium	FLOAT	10	Potassium level
created_at	TIMESTAMP	20	Time stamp for the data

vi. Crops_table.

Table 7: Crops_table.

Field	**Data Type**	**Fieldsize**	**Description**
Id	INT	20	Primary key project identifier
crop_name	VARCHAR	50	Crop_name
moisture_requirement	FLOAT	50	Ideal moisture range
min_ideal_nitrogen	FLOAT	50	minimum N levels required
min_deal_phosphorous	FLOAT	10	Minimum P levels required
min_ideal_Potassium	FLOAT	10	minimum K levels required
max_ideal_nitrogen	FLOAT	10	Maximum N levels required
max_deal_phosphorous	FLOAT	10	Maximum P levels required
max_ideal_Potassium	FLOAT	10	Maximum K levels required

vii. Recommendation_Table

Table 8: Recommendation_Table

Field	**Data Type**	**Fieldsize**	**Description**
Id	INT	20	Primary for Recommendation

sensor_data	INT	10	Reference to the sensor data of farm
recommendation_details	TEXT	255	Recommendation details explanation
created_date	TIMESTAMP	25	Farm Location

Functions and Function Parameters:

Table 9: Functions and Function Parameters

Function Name	Description	Input Parameters
User Signup	Registers a new user/organization.	name, email, password
User Login	Authenticates a user into the system.	email, password
Create Farm	Adds a new farm under a user's farms list.	user_id, name, Crops,location
Assign Crop to Farm	Links a crop to a farm with custom conditions.	farm_id, crop_id, moisture_requirement, nutrient_requirement
Generate Crop Recommendation	Suggests crops based on sensor data.	sensor_data_id
User Logout	Logs a user out of the system.	auth_token

5 COMPONENT DESIGN

This section contains pseudocodes of the major components of the system.

5.1 Pseudocode for Advisor

```
START Chatbot System
  FUNCTION receive_user_input()
      DISPLAY "Enter your question:"
      user_question ← GET user input
      RETURN user_question
  END FUNCTION

  FUNCTION is_domain_relevant(user_question)
      keywords ← ["soil", "moisture", "fertilizer", "crop", "nutrients", "temperature", "phosphorus",
"potassium", "irrigation"]
      FOR word IN keywords:
          IF word IN user_question THEN
              RETURN TRUE
      END FOR
      RETURN FALSE
  END FUNCTION

  FUNCTION fetch_latest_sensor_data()
      latest_data ← QUERY database FOR most recent soil readings
      RETURN latest_data
  END FUNCTION

  FUNCTION prepare_gpt_prompt(user_question, sensor_data)
      prompt ← "
          Soil Data:
          - Moisture: {sensor_data.moisture}%
```

```
        - Nitrogen: {sensor_data.nitrogen}
        - Temperature: {sensor_data.temperature}°C

        User Question: {user_question}
        Provide a detailed farming recommendation.
    "
    RETURN prompt
END FUNCTION

FUNCTION send_request_to_gpt(prompt)
    response ← CALL OpenAI API WITH prompt
    generated_advice ← PARSE response FROM GPT
    RETURN generated_advice
END FUNCTION

FUNCTION save_chat_history(user_question, generated_advice)
    STORE (user_question, generated_advice) IN database
END FUNCTION

FUNCTION chatbot_response()
    user_question ← receive_user_input()

    IF NOT is_domain_relevant(user_question) THEN
        RETURN "Sorry, I can only provide soil and crop-related advice."

    sensor_data ← fetch_latest_sensor_data()
    formatted_prompt ← prepare_gpt_prompt(user_question, sensor_data)
    advisory_response ← send_request_to_gpt(formatted_prompt)
    save_chat_history(user_question, advisory_response)
    RETURN advisory_response
END FUNCTION
```

```
  CALL chatbot_response()

END Chatbot System
```

5.2 Pseudocode for Recommendations

```
START Recommendation System
  FUNCTION fetch_data()
    RETURN QUERY database FOR soil data
  END FUNCTION

  FUNCTION generate_recommendations(data)
    recommendation ← ""
    IF data.nitrogen < threshold THEN recommendation ← "Apply nitrogen."
    IF data.moisture < threshold THEN recommendation ← recommendation + " Increase irrigation."
    IF data.temperature < range THEN recommendation ← recommendation + " Monitor temperature."
    RETURN recommendation
  END FUNCTION

  CALL generate_recommendations(fetch_data())
END Recommendation System
```

5.3 Pseudocode for the Notify function

```
START Notification System
  FUNCTION fetch_critical_alerts()
    alerts ← QUERY database FOR any critical soil condition alerts
    RETURN alerts
  END FUNCTION
```

```
  FUNCTION send_notification(alerts)
    IF alerts NOT EMPTY THEN
      notification_message ← "Urgent soil issue: " + alerts
      SEND notification_message TO user
    ELSE
      RETURN "No critical alerts."
  END FUNCTION

  FUNCTION notify_user()
    alerts ← fetch_critical_alerts()
    send_notification(alerts)
  END FUNCTION

  CALL notify_user()

END Notification System
```

6 HUMAN INTERFACE DESIGN

6.1 Overview of User Interface

The Smart Cloud-based Soil Advisor System provides an intuitive and user-friendly interface designed to assist users in managing farms, monitoring soil health, and receiving AI-powered recommendations. The system is structured into multiple interfaces, each serving a distinct function:

6.1.1. User Registration Interface

This interface allows new users to create accounts. It includes:

- A Signup form for entering user details (name, email, password).
- Account verification through email or phone OTP authentication.
- Login & Password recovery options for registered users.

6.1.2. Farm Registration Interface

This interface enables users to register and manage farms:

- Create a new farm profile by entering farm name, location, and size.
- View and manage registered farms under an organization or individual account.
- Assign team members for collaborative farm management.

6.1.3. Device Registration Interface

Users can register IoT devices (sensors) that collect soil data. Features include:

- Adding a new device by entering device ID and farm location.
- Configuring sensors to collect moisture, temperature, and nutrient data.
- Monitoring device status (active/inactive).

6.1.4. Dashboard Interface

- The dashboard provides an overview of farms owned by the farmer and also shows the available farms under management with their operation status of that particular farm and the crops being grown on the farm.

6.1.5. Farm Dashboard interface

- Each farm has its own dashboard for visualisation on the of its soil parameters, the moisture, temperature and soil nutrients (Potassium, Nitrogen, Nitrogen).
- This also contains the specific recommendations for that particular farm. The visualisation graphs and also of how the nutrients and the moisture levels degrade with time.

6.1.6. AI Advisor Agent Interface

The AI-powered recommendation system provides insights for improved farming:
This provides a user prompts that allows a farmer to ask questions on how they can improve their soil quality and fertiliser applications.

6.2 Screen Images

6.2.1. Account Registration Form

This screen allows new users to create an account by filling in the required registration details such as name, email, and password, as shown in Figure 6.1 below.

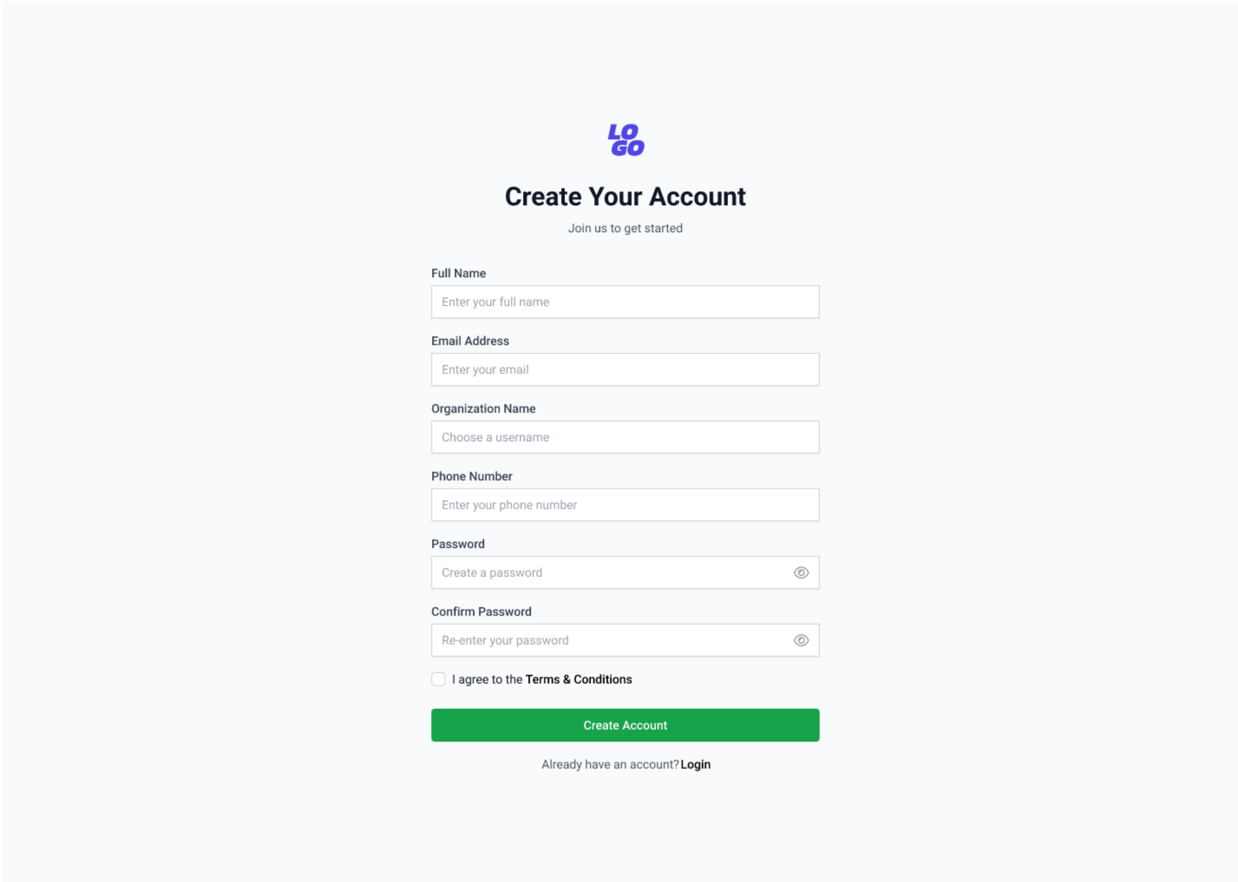

Figure 6.1: Account Registration Form

6.2.2. Login Form

This screen enables registered users to log into the system by entering their credentials, including username/email and password, as shown in Figure 6.2.

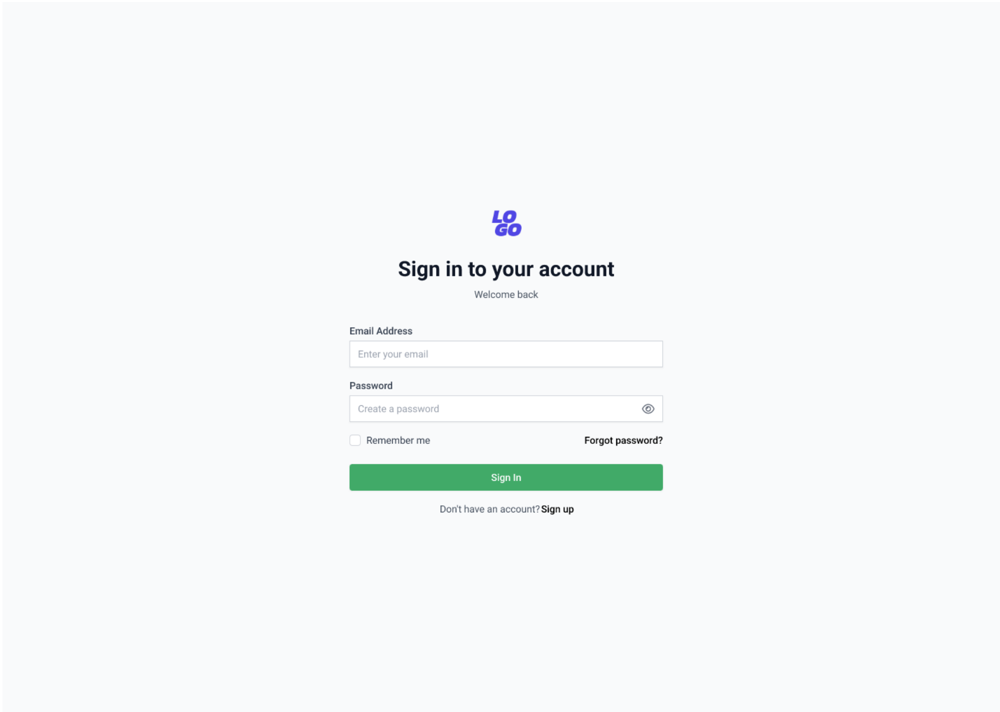

Figure 6.2: Login Form

6.2.3. Device Registration Screen Form

This screen facilitates the addition of new IoT devices to the system by collecting device-specific information such as device ID, type, and farm location, as seen in Figure 6.3.

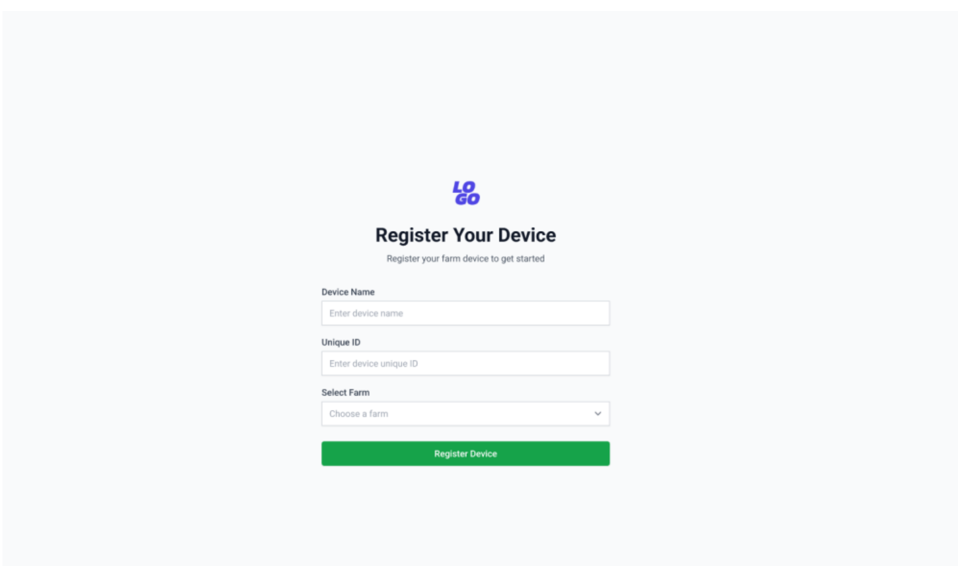

Figure 6.3: Device registration screen form

6.2.4. Dashboard Screen

Once logged in, users are taken to the dashboard screen which gives an overview of all registered farms and devices. The screen contains options to add new farms and view the details of each farm, as shown in Figure 6.4.

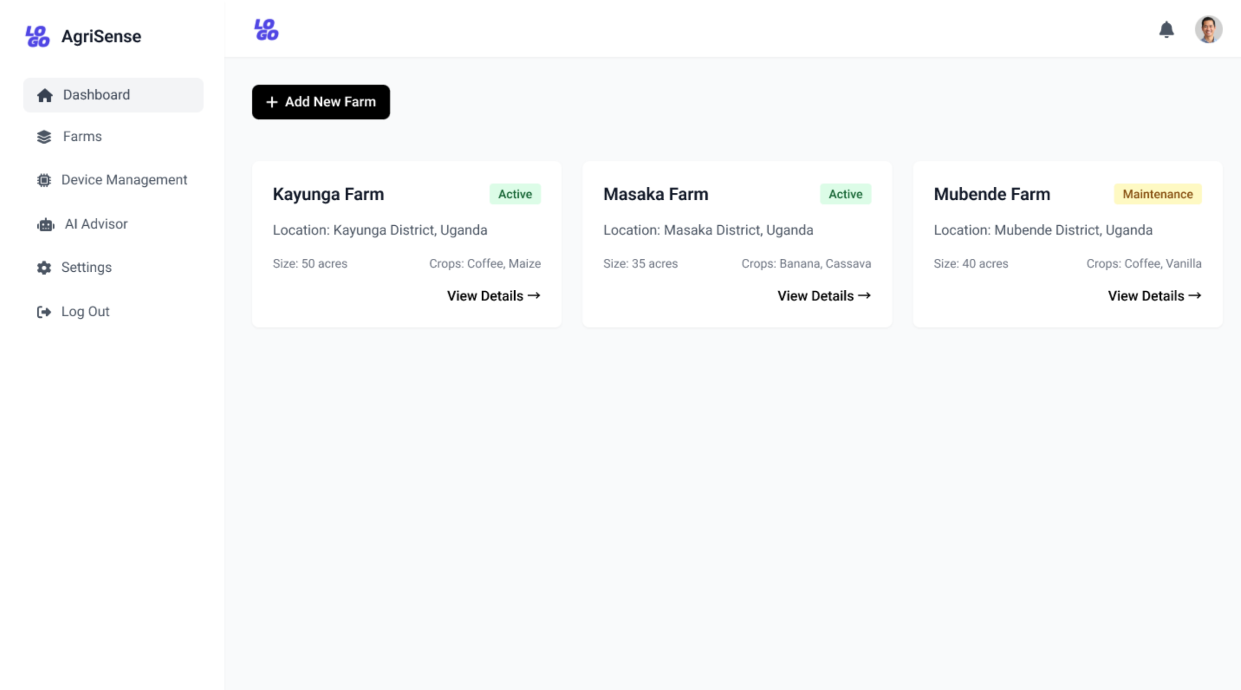

Figure 6.4: Dashboard screen

6.2.5. Farm Dashboard Screen

This screen provides detailed information about a selected farm, including device metrics and historical data on the farm's performance, as shown in **Figure 6.5**.

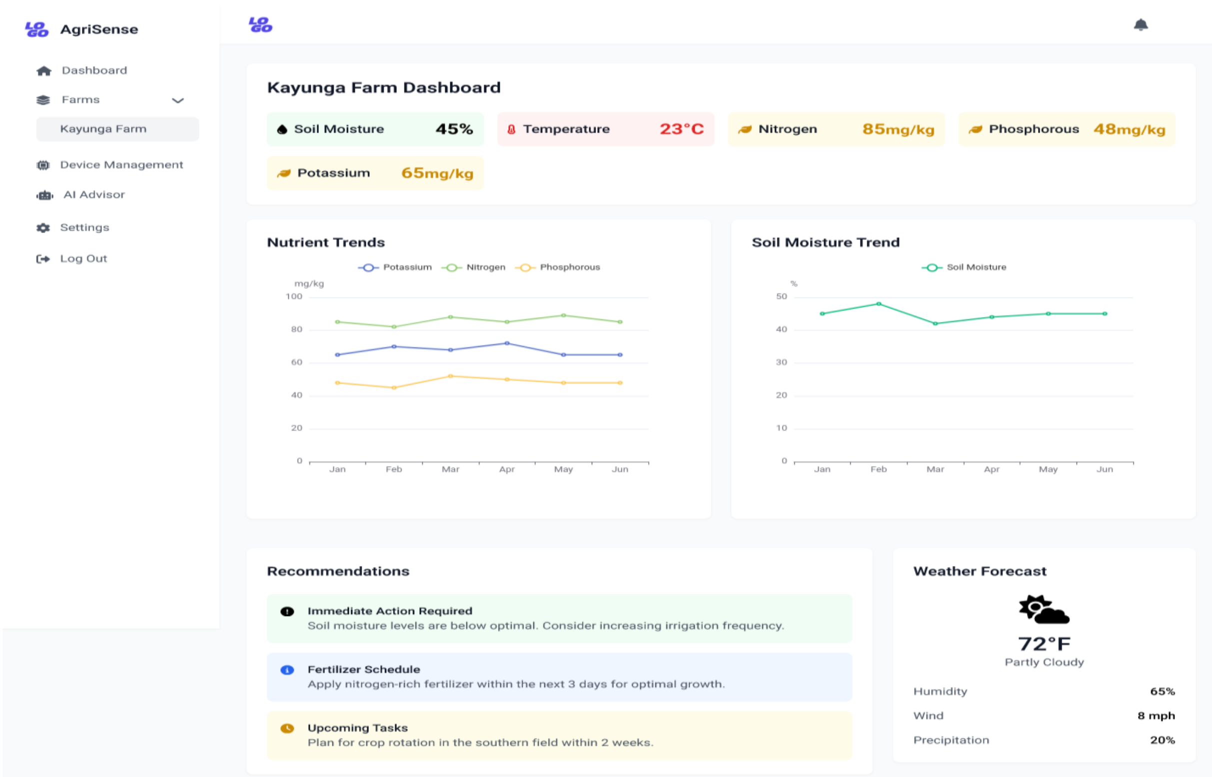

Figure 6.5: Farm dashboard screen

6.2.6. Device Management Screen

The device management screen displays a list of all registered devices. Users can add new devices, update existing ones, or view detailed information about each device, as shown in **Figure 6.6**.

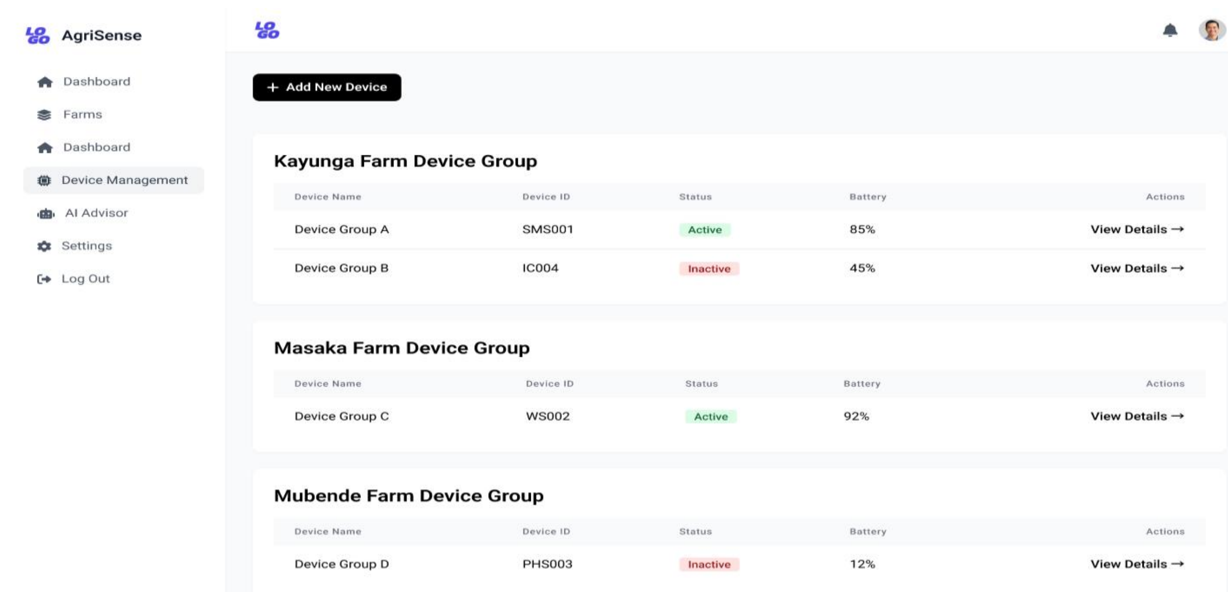

Figure 6.6: Device Management Screen

6.2.7. Farm Dashboard with Advisory Screen

This screen displays farm data alongside an AI-generated advisory system. A chatbot pops up, allowing users to type questions and receive advice. It also shows the farm's performance metrics, as seen in Figure 6.7.

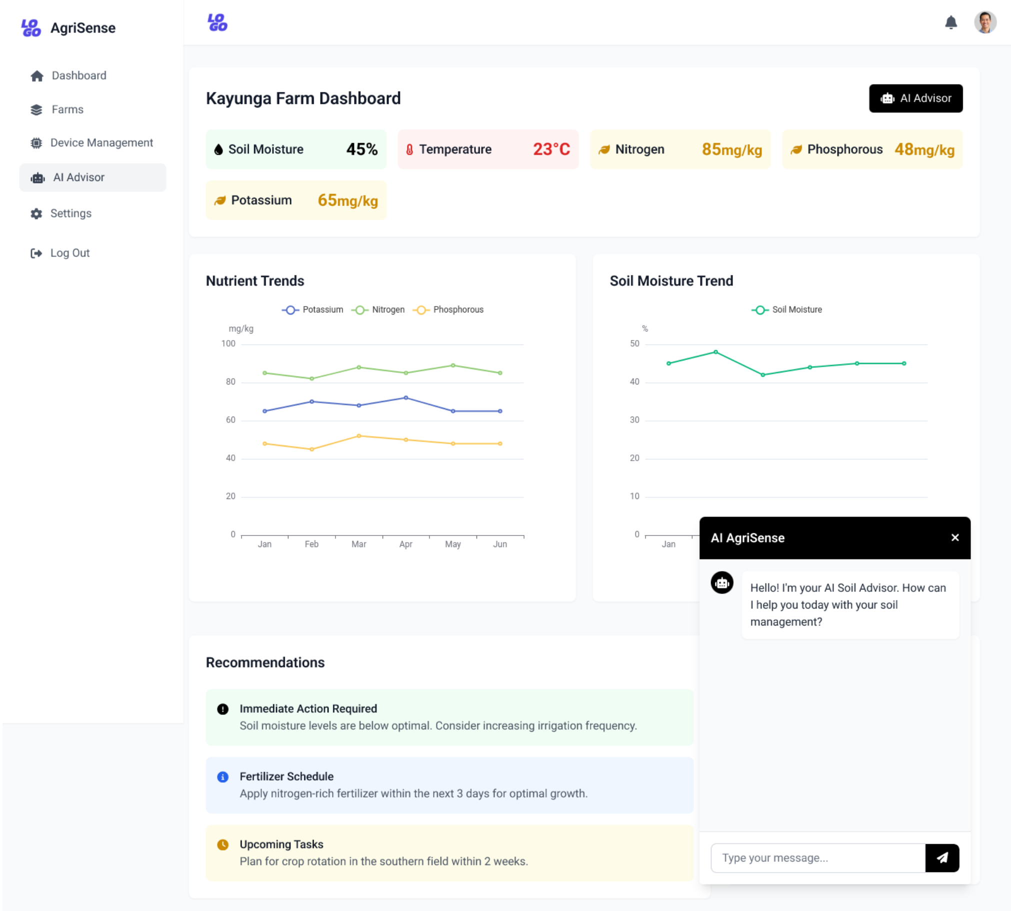

Figure 6.7: Farm dashboard with Advisory screen

6.2.8. Soil NPK Sensor

The Soil NPK sensor captures essential nutrient levels such as nitrogen, phosphorus, and potassium, and transmits this data to the farm's dashboard for real-time monitoring, as shown in **Figure 6.8**.

Figure 6.8: Soil NPK sensor for capturing (Nitrogen, phosphorous, and potassium)

6.2.9. Soil Temperature Sensor

This sensor captures and records the temperature of the soil, feeding this information back to the system for display on the farm dashboard, as seen in Figure 6.9.

Figure 6.9: Soil Temperature sensor

6.2.10. Soil Moisture Sensor

The Soil Moisture sensor monitors the moisture content in the soil and transmits this data to the system for visualization on the farm dashboard, as shown in Figure 6.10.

Figure 6.10: Soil moisture sensor

6.2.11. LCD Screen

The LCD screen displays real-time data from the sensors and IoT devices, including soil conditions and farm statistics, allowing users to keep track of their farm's performance, as shown in **Figure 6.11**.

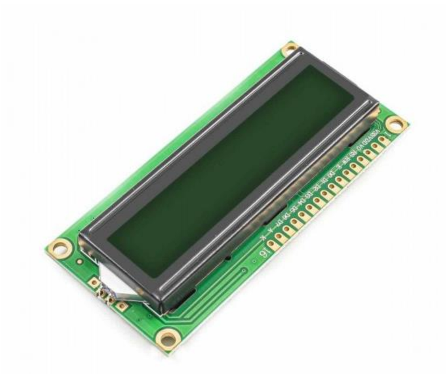

Figure 6.11: LCD screen

6.2.12. ESP32 Microcontroller

The ESP32 microcontroller is the central component that manages communication between the IoT sensors and the cloud-based system. It processes data from the devices and sends it to the dashboard for analysis, as shown in **Figure 6.12**.

Figure 6.12: ESP32 micro-controller

6.3 Screen Objects and Actions

Table 10: Screen Objects and Actions

Identifier	Object Name	Action
Figure 6.1: Registration Form	Account Registration Form	Displays the form where users can input details to register.

	Form fields	Allow the user to enter required registration details like name, email, and password.
	Create Account button	On clicking, the system registers the user and redirects to the login form.
Figure 6.2	Login Form	Displays the form for user authentication.
	Form fields	Allow the user to input their login credentials.
	Sign in button	Authenticates the user and grants access to the dashboard.
Figure 6.3	Device Registration Form	Displays the form for registering a new IoT device.
	Form fields	Allow the user to input device details such as device ID and type.
	Register Device button	Adds the new device and updates the dashboard with the device information.
Figure 6.4	Dashboard Screen	Displays an overview of all registered farms.
	Sidebar menu	Contains options like Dashboard, Farms, Device Management, AI Advisor, Settings, Logout.
	Add New Farm button	Opens a form to register a new farm.

	Notification bell icon	Pops up a notification window with recent alerts and messages.
	Profile pic icon	Opens a menu with profile options such as account settings and logout.
	View Farm Details button	Takes the user to the detailed view of the selected farm.
Figure 6.5	Farm Dashboard Screen	Displays detailed data about the selected farm, including device metrics and historical trends.
	Sidebar menu	Allows navigation to Dashboard, Farms, Device Management, AI Advisor, Settings, Logout.
Figure 6.6	Device Management Screen	Displays a list of registered devices with options to manage each one.
	Add New Device button	Opens a form to register a new device and add it to the farm.
	View Device Details button	Shows detailed information about a specific device, including status and recent readings.
Figure 6.7	AI Advisor Chatbot	A pop-up chatbot that allows users to type a message and receive farm optimization advice.

	Send icon button	Sends the typed message to the chatbot for AI-generated recommendations.
	Cross icon button	Closes the chatbot window.
Figure 6.8	Soil NPK Sensor	Captures soil nitrogen, phosphorus, and potassium levels, and transmits the data for display on the dashboard.
Figure 6.9	Soil Temperature Sensor	Captures soil temperature data and sends it for processing and visualization on the farm dashboard.
Figure 6.0	Soil Moisture Sensor	Monitors soil moisture levels and updates the system with real-time data.
Figure 6.11	LCD Screen	Displays real-time sensor data.
Figure 6.12	ESP32 Microcontroller	Manages data collection and communication between IoT sensors and the cloud-based system.

7 REQUIREMENTS MATRIX

Table 11: Requirements Matrix Table

Requirement	Components and data structures
R001	i. Soil Data Collection Component ii. Sensor Data Acquisition Module
R002	i. Machine Learning Model for Soil Health ii. Data Cleaning and Preprocessing Module
R003	i. Data Analysis Component ii. Data Cleaning and Preprocessing Component iii. Machine Learning Models Component
R004	i. Notification Component ii. Dashboard Component iii. Recommendations Component iv. Historical Data and Reports Component

8 APPENDICES

Link to the Blog: https://fyp-blog.onrender.com/

Link to the Project repository: https://github.com/HarrisonMoses/BSE25-28-CBSA.git

Link to the Software Requirement Document:

https://docs.google.com/document/d/1iB9znkpC0N7H4dISkEq429n_TfY9K4Ra/edit