Software Design Description

<Smart Farming System using IOT>

Version <1.0>

Prepared by

F4NT4STIC
SECJ4423 – Real-time Software Engineering
Malaysia Japan International Institute of Technology
Universiti Teknologi Malaysia
(*Lecturer: Nor Azizah Binti Saadon*)

Revision Page

a. Overview

The current version of the document describes the introduction and specific requirements of the system. The scope, definition, reference, purpose, acronyms, abbreviation, and overview of the system are included in the introduction. External interface requirements, system features (including use case diagram, sequence diagram, and use case specification), performance requirements, design constraints, software system attributes, and other system requirements are included in specific requirements.

b. Target Audience

Stakeholders and system analysts.

c. Project Team Members

1. Ruhul Quddus Tamim (Project Leader)
2. Md Yusuf Bin Forkan
3. Shafi Ahmed
4. Syafiq Ibnu Ramadhan

d. Version Control History

Version	Primary Author(s)	Description of Version	Date Completed
1.0	Ruhul Quddus Tamim MD Yusuf Bin Forkan Shafi Ahmed Syafiq Ibnu Ramadhan	Combined SRS, SDD, STD as SD	15/05/2022
1.1			

Note:

This document is adapted from ***IEEE SA - 830-1998 -*** *IEEE Recommended Practice for Software Requirements Specifications* and ***IEEE SA - 1016-1998 -*** *IEEE Recommended Practice for Software Design Descriptions*.

Table of Contents

1 Introduction 1
1.1 Purpose 1
1.2 Scope 1
1.3 Definitions, Acronyms and Abbreviations 1
1.4 References 2
1.5 Overview 2
1.6 Real-time profile and modeling tools 3
1.7 Innovation solution for IOT and Data analytics 4
2 System Software Process Model 5
2.1 Introduction 5
3 System Software Requirements 6
3.1 Functional 6
3.2 Non-Functional 6
3.2.1 Performance 7
3.2.2 Security 7
3.3 Constraints 7
3.4 Interface 7
4 System Software Architectural Design 7
4.1 System Architecture 7
4.2 Software Architecture – Component Diagram 8
4.3 Class Diagram 9
5 Detailed Description of Components 10
5.1 Complete Package Diagram 11
5.2 Detailed Descriptions 12
5.2.1 Module Manage User 12
5.2.1.1 P001: Package <Manage User> 12

5.2.1.2 Sequence Diagrams 12

5.2.1.3 Sequence Diagrams 13

5.2.1.4 Sequence Diagrams 14

5.2.1.5 Sequence Diagrams 15

6 Data Design 16

6.1 Data Description - ERD 16

6.2 Data Dictionary 16

7 User Interface Design 19

7.1 Overview of User Interface 19

7.2 User Interface 19

8 Requirements Matrix 23

1 Introduction

1.1 Purpose

Agriculture plays a vital role in human life because it is the primary source of food and development of the country's economic factor. Due to the increment in population and decrement in rainfall amount, there is a fundamental scarcity of food – Hence, the importance of precision agriculture [1]. Precision agriculture can be defined as the skill and ability to use modern technology to improve and increase crop production and decrease human effort. WSN is one of the methods used to monitor and control different parameters in the agriculture field. WSN is low-cost, low power consumption. It comprises groups of sensor nodes that monitor several conditions, such as temperature, humidity, moisture, etc. WSN in agriculture assists in collecting data, monitoring environmental parameters, suitable water and fertilizer given to plants for good crop production, and helping farmers in real-time data gathering.

This SD describes the "Smart Farming System," which is to make the system's data collection using Raspberry Pi 3. SD is the description, specification, development, and deployment of a new software application. They may involve the internal production of specialized applications and database construction.

1.2 Scope

The proposed system has been implemented for real-time monitoring of agricultural fields depending on WSN, which contains several sensor nodes, base station nodes, and various sensors. Each sensor node is equipped with light, temperature, humidity, and soil moisture sensors that collect data at certain intervals and then send these data to be stored in a database at the base station using Wi-Fi technology and MQTT protocol. The data is then displayed on a web page as a user interfaces for real-time monitoring. The suggested system has two portions: hardware side and software side

1.3 Definitions, Acronyms and Abbreviations

Acronyms / Abbreviations	Definitions
SD	System Development
WSN	Wireless Sensor Network
MQTT	Message Queuing Telemetry Transport

1.4 References

1. Farzad Kiani, Amir Seyyedabbasi, "Wireless Sensor Network and Internet of Things in Precision Agriculture," (IJACSA) International Journal of Advanced Computer Science and Applications, Vol. 9, No. 6, 2018.
2. Aarti Rao Jaladi, Karishma Khithani, Pankaja Pawar, Kiran Malvi, Gauri Sahoo," Environmental Monitoring Using Wireless Sensor Networks(WSN) based on IoT," International Research Journal of Engineering and Technology (IRJET) Volume: 04 Issue: 01 | Jan -2017

3. N. Sakthipriya "An Effective Method for Crop Monitoring Using Wireless Sensor Network," Middle East Journal of Scientific, 2014

4. D. Mahir and O. Semih, "A wireless application of drip irrigation automation supported by soil moisture sensors," Gazi University, Scientific Research and Essays Vol. 6(7), ISSN 1992-2248 Academic Journals, pp. 1573-1582, 4 April 2011.

5. J.S. Awati and V.S. Patil, "Automatic Irrigation Control by using wireless sensor networks," Journal of Exclusive Management Science Vol 1 Issue 6 – ISSN 2277 – 5684, June 2014.

6. S. Zain-Aldeen, S. Ramzy and H. Basil, "Wirelessly Controlled Irrigation System," Dept. of Electrical Engineering University of Basrah, Iraq J. Electrical and Electronic Engineering, 2014.

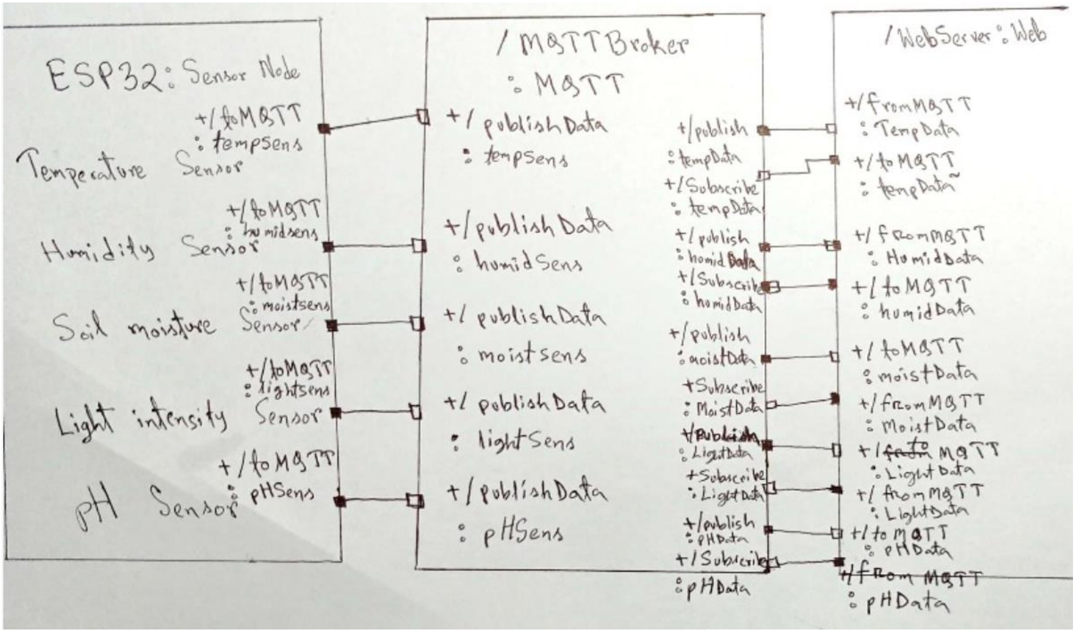
1.5 Overview

This section presents an overview of some works related to this work: [4] suggested a wireless sensor network system to control irrigation and monitor a soil parameter in real-time—this project comprises three components: a base station, a sensor unit, and a valve unit. The base station was a control master device used to receive and analyze sensor data to control the irrigation process. The sensor unit was used to gather data from the soil, and the valve unit was employed to provide crops water. This system did not consider the time for irrigation and crop water needs. [5] used 8051 micro-controller, number of sensors to control farm door, switch on, switch off lighting, and fire detection. This system did not take into account the farm irrigation process. [6] wireless sensor networks to design a smart irrigation system; it contained Arduino Uno, ATmega328 Micro-controller, soil moisture, temperature and humidity sensors, and Xbee technology. [7] proposed a drip irrigation system that used raspberry pi as a base station unit, Arduino ATmega328 as a sensor unit, numbers of sensors, and ZigBee technology. The sensor unit collected the data and then sent it to the base station wirelessly. In [8], the proposed system is developed to monitor soil's humidity, temperature, and moisture. They used ZigBee technology, solar panel, and mobile application to interface with the user. Efficiency, low cost, and low power consumption of the WSN are considered in the project and dealt with.

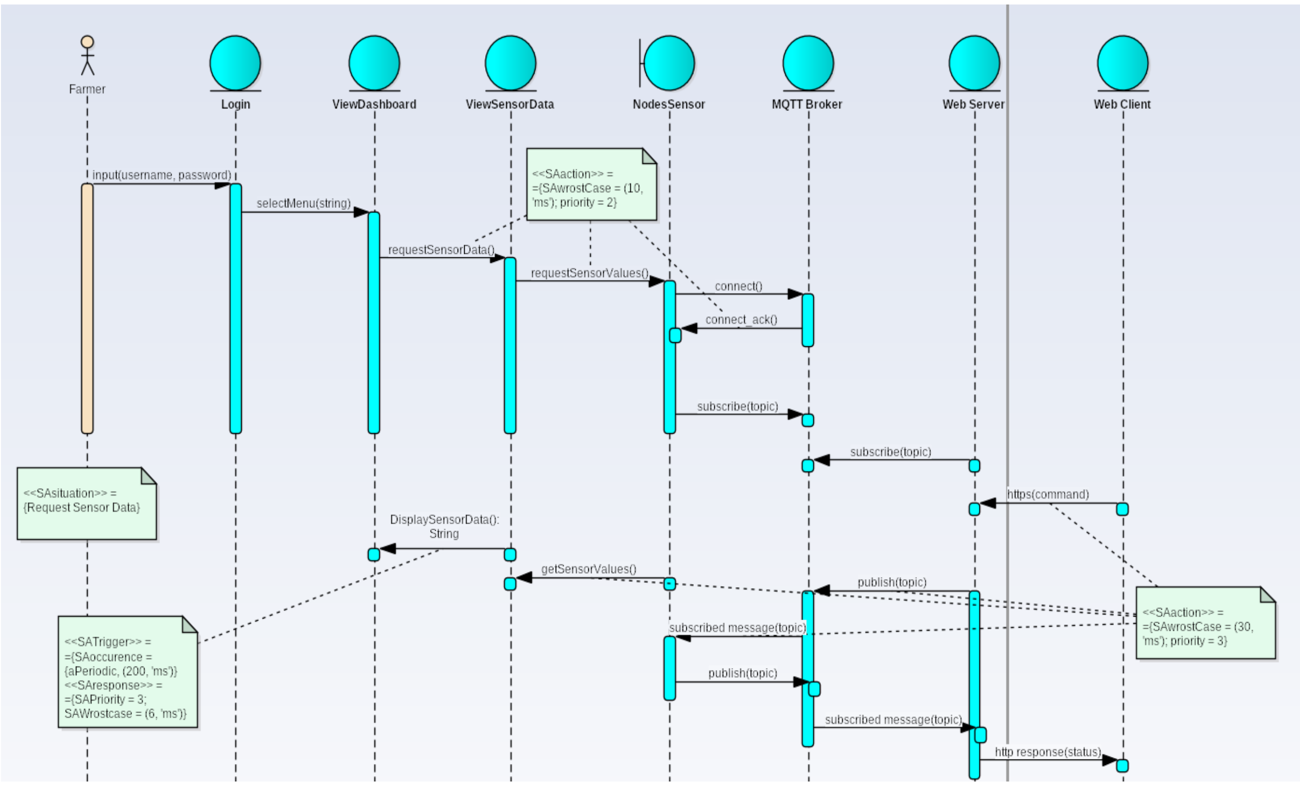
1.6 Real-time profile and modeling tools

Real-Time Modelling Profile and tool: Rational Rose Real-Time (RoseRT) for UML-RT

ii) Structure Diagram of MQTT Broker capsule



Timing Sequence Diagram for Request Sensor Data use cases scenario in UML-RT



iii) **Strength:**

- Robust design pattern centered on the concept of active objects. Active objects have an encapsulation shell that encapsulates their data, their run-to-completion behavior, and most importantly, their thread of execution, thus preventing thread interaction problems.

Weakness:

- Does not support the activity diagram.
- It is not open source.

1.7 Innovation solution for IoT and Data analytics

IoT networks can transfer data without human-to-human or human-to-computer interaction. As a result, our world is more connected than ever before. The IoT has infiltrated homes all around the country in the form of **smart thermostats, smart speakers**, and more.

Internet of Things (IoT) analytics enables organizations to leverage the massive amounts of data generated by IoT devices using analytics stacks. IoT analytics is often considered a subset of big data and involves combining heterogeneous streams and transforming them into consistent and accurate insights.

It has already made a tremendous impact in many application domains such as smart cities, sustainable living, manufacturing, and healthcare. IoT analytics analyzes **data from disparate data sources, including sensors, actuators, and other objects connected to the internet**.

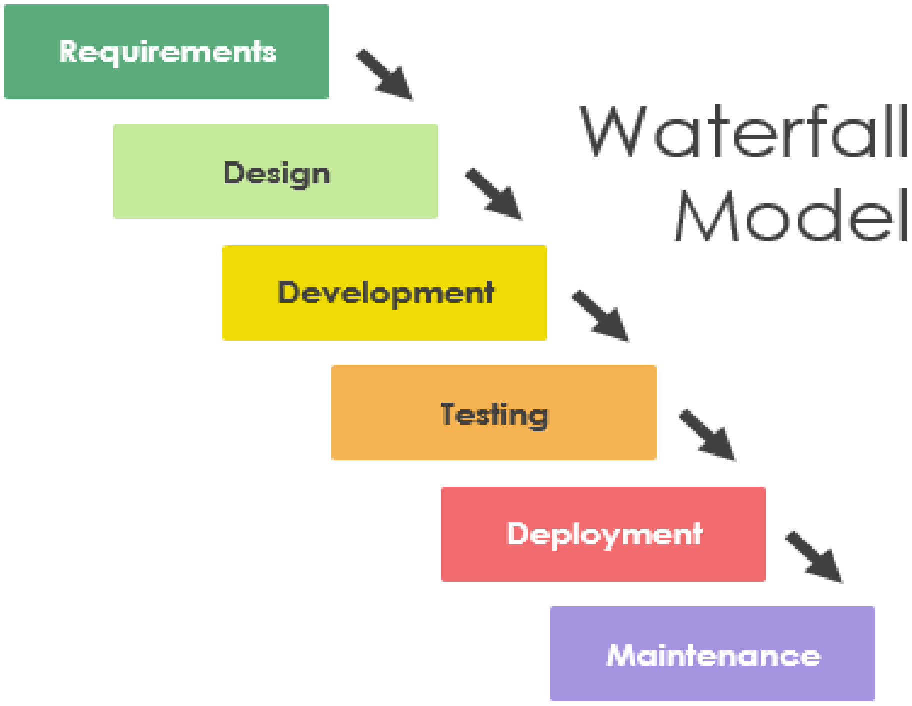
Efficient processor designs combined with AI-enhanced intelligence, end-to-end security, development platforms and tools for multiple workloads, devices, and clouds, and massive scalability are essential elements of an IoT solution.

2 System Software Process Model

2.1 Introduction

Software processes, methodologies, and frameworks range from specific prescriptive steps that an organization can use directly in day-to-day work to flexible frameworks that an organization uses to generate a custom set of measures tailored to the needs of a specific project or group.

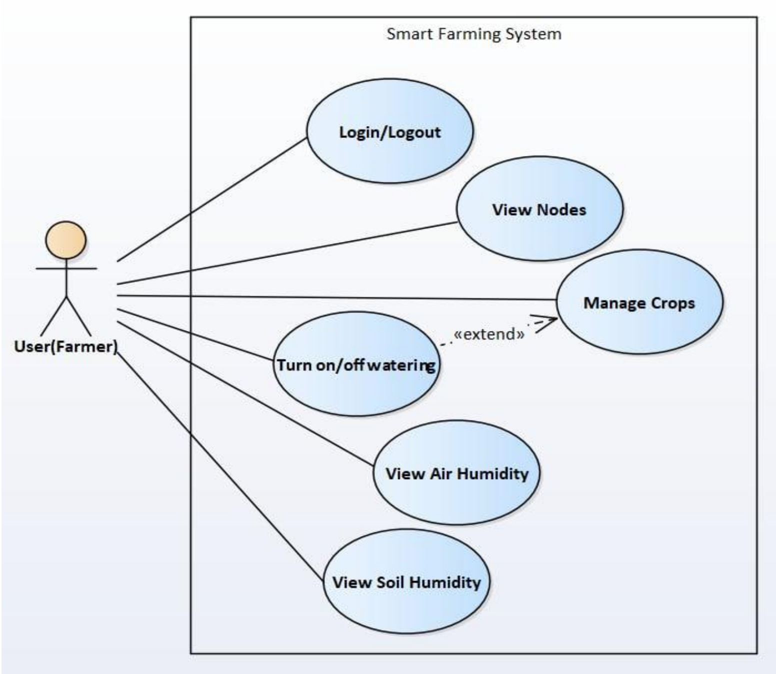
For the project, the Waterfall SDLC model is used.



The waterfall model breaks project activities into linear sequential phases, where each step depends on the deliverable of the previous one and corresponds to a specialization of tasks. The approach is typical for some regions of engineering design.

3 System Software Requirements

3.1 Functional



- The system shall let the user login into the system and view the homepage.
- The system shall let the user view all the activated and deactivated sensors.
- The system shall let the user manage their crops by turning on/off the watering.
- The system shall let the user view all the Air and Soil humidity values according to the date of the sample.

3.2 Non-Functional

Usability
The GUI (Graphical User Interface) shall be easy to learn and use by users of any technical background. A built-in help feature shall be available on all pages to guide the users with the available functions on that page. Easy-to-understand documentation must be provided with the system. The system shall support several languages.

Reliability
The system shall never crash or hang other than as the result of an operating system error. The plan shall provide graceful degradation in the face of network delays and failures or when handling large graphs.

Security and Privacy
The system, at any time, shall be accessed only by the authenticated users. The system must end the session automatically when an open session is not used for a specific period.

Maintainability
The document shall be easy for the users who execute the system daily, the system admin who wishes to edit or develop further, and the personnel in charge of the maintenance.

3.2.1 Performance

1. The system response time must be 8.5 to less than 5 seconds.
2. The system should be capable of supporting at least 50000 users to use the system concurrently without any error.
3. The database should be able to store unlimited data collected by the sensors.
4. The system server shall be capable of supporting an arbitrary number of active users who are currently monitoring their plants.
5. The system shall not disclose the user information to any third party.

3.2.2 Security

3.3 Constraints

- For non-functional requirements, the system shall be able to authorize users via email and password.
- The user manual should be structured clearly and simple, for example. The manual should explain to the user what the software should do and how it functions. The manual is used as a reference so that the user can understand the system quickly.
- Users should be able to access the system from any device with an internet connection and internet browsing.
- The information of the users is only able to be accessed by the authority that the admin authorizes.
- The system shall support up to 50000 users simultaneously with loading time at most 8.5 to less than 5 seconds.

3.4 Interface

When a user uses Smart Monitor Farming for the first time, the user needs to log in using the email and password provided by the admin. After that, the user can log in to the system successfully. When the user logs in to the system, the system shall display the main page that provides services relevant to the user's identity.

The user can click on the Node page, Air humidity page, Manage Crops page, Graph page, and Soil Humidity page, where it will show the user all the values in real-time collected by the sensors, and the user can control watering the crops from the Manage Crops page.

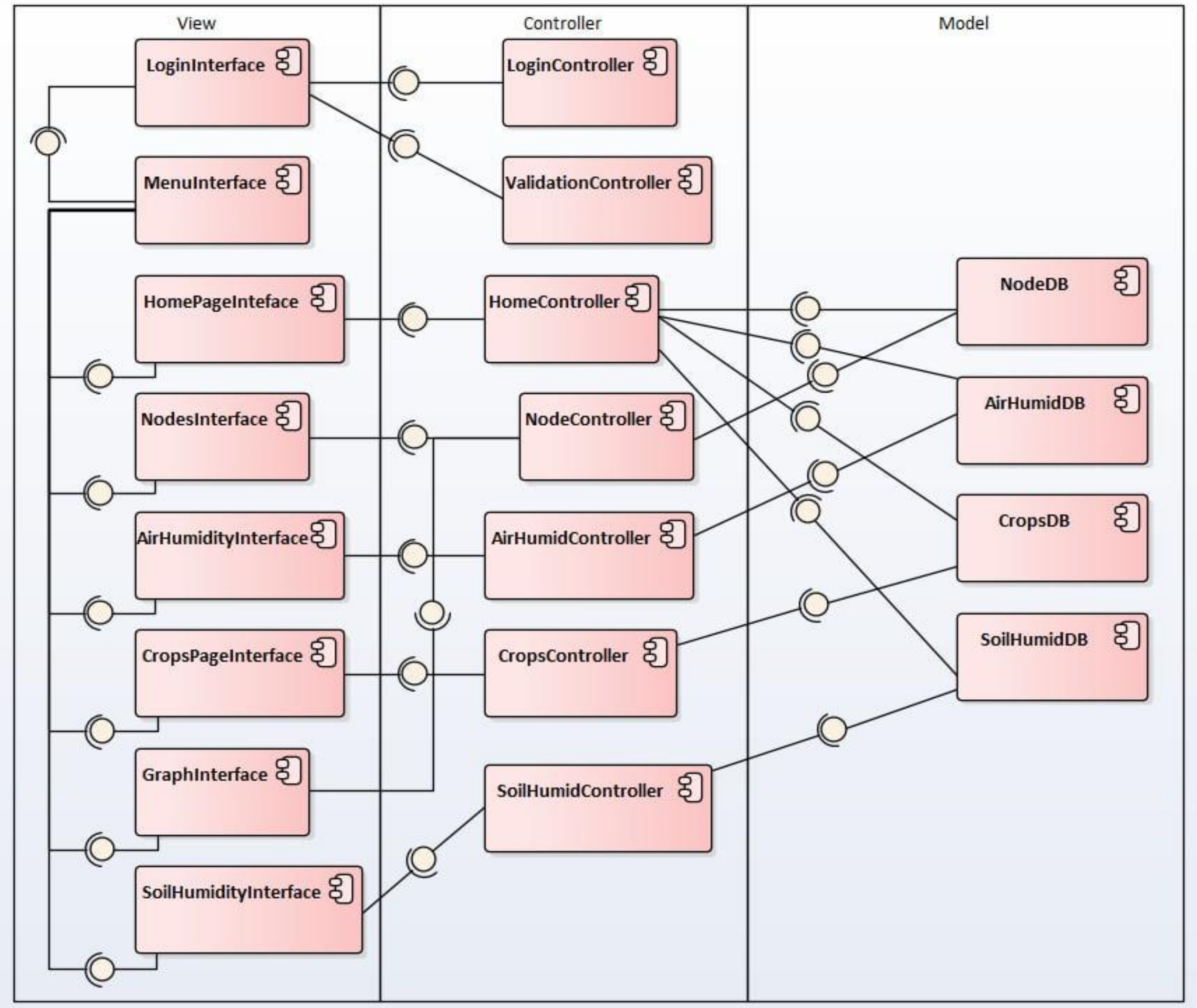
4 System Software Architectural Design

4.1 System Architecture

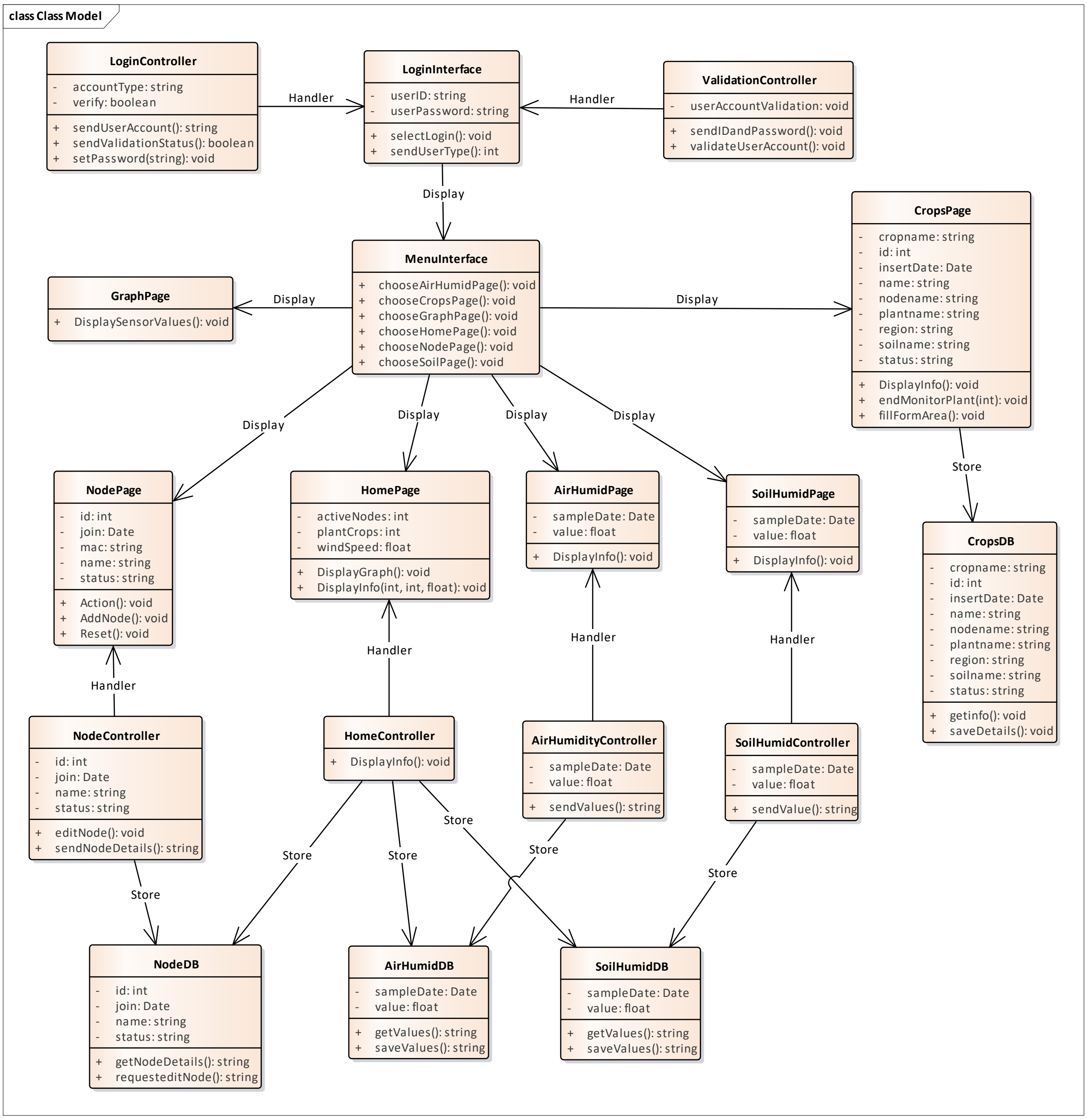
The chosen architectural design pattern is a model view controller (MVC). The main reason that causes us to select this model is that the system has multiple ways to view and interact with data and the future requirements for interaction and presentation of data are unknown.

MVC is an acronym for model, view, and controller, and it is a product development architecture. By using MVC, we could create applications that separate the application's different aspects. Since MVC allows the data to change independently of its representation and vice versa, it enables us to focus on one part of the implementation at a time. For example, we can focus on the controller (input logic) without depending on models and views (business logic and UI logic). Using MVC allows us to modify our system quickly, and multiple developers can work simultaneously on the model, controller, and views.

4.2 Software Architecture – Component Diagram



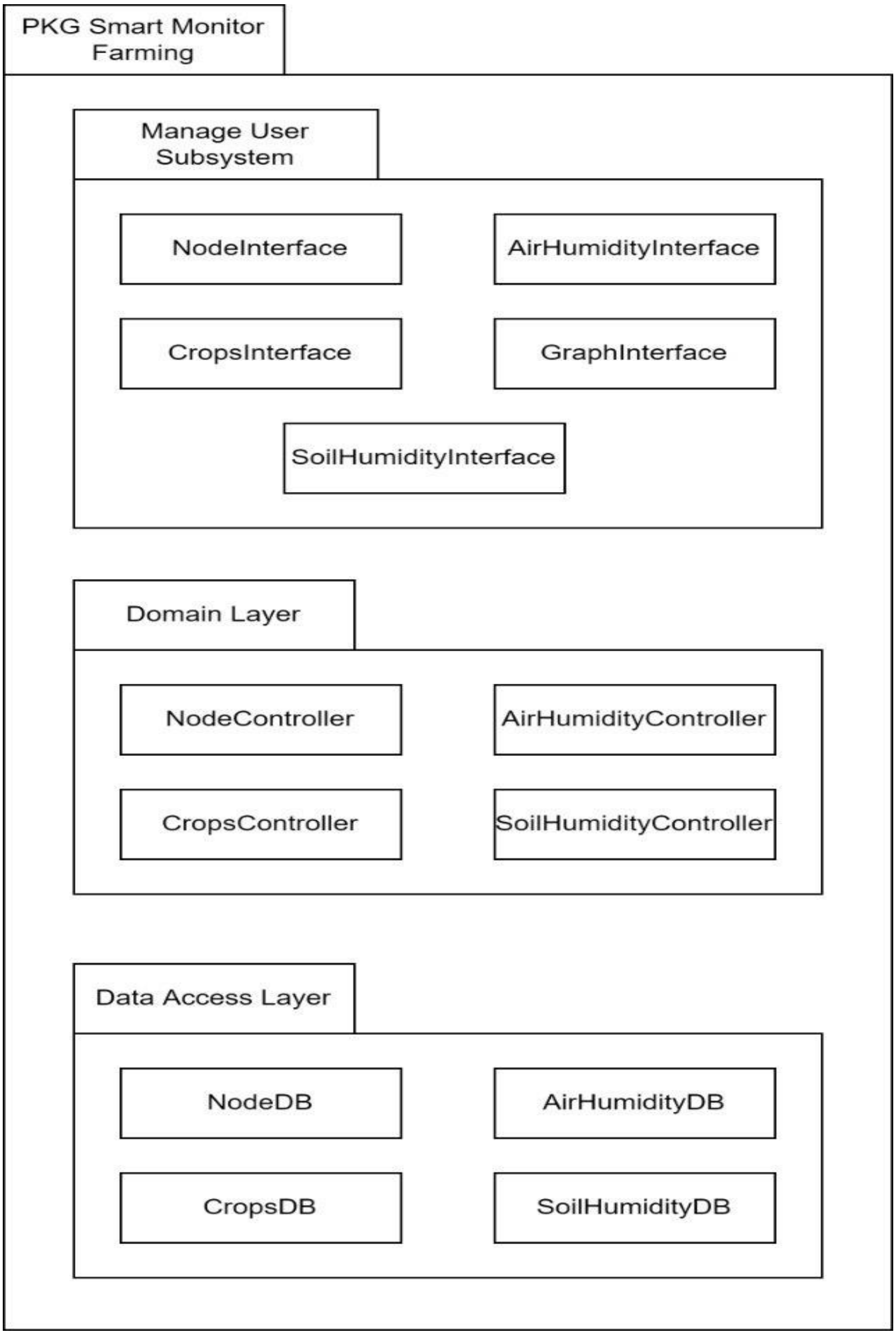
4.3 Class Diagram



5 Detailed Description of Components

The proposed Smart monitor Farming system consists of one subsystem: The manage User module. The subsystem consists of the view, domain, and data access layers. The view layer is a layer that allows the user to interact with the system. The domain layer includes controller classes related to the primary function and logic of the subsystem. The data access layer is a layer that accesses the data from the database.

5.1 Complete Package Diagram



5.2 Detailed Descriptions

5.2.1 Module Manage User

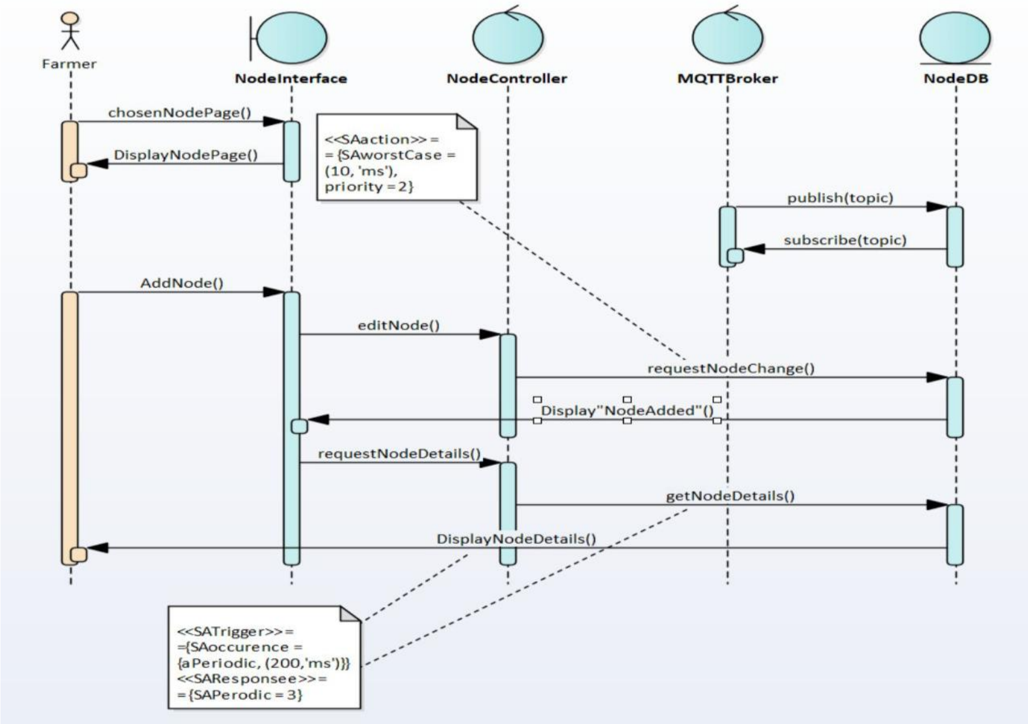
5.2.1.1 P001: Package <Manage User>

This module's diagram is the same as the package diagram. Hence, for this module diagram, we followed what we used for the package diagram

Entity Name	Manage Node
Method Name	selectNodeDetails(), editNodeDetails()
Input	Click on the node interface page
Output	Display Successful message that a new node is added as well as display all the information of the nodes
Algorithm	1. Choose to create a new node. 2. Fill up the form to create a new node and submit, which calls the method editNodeDetails(). 3. After creating a new node, it will display that a node has been added, and all the node details will be available.

5.2.1.2 Sequence Diagrams

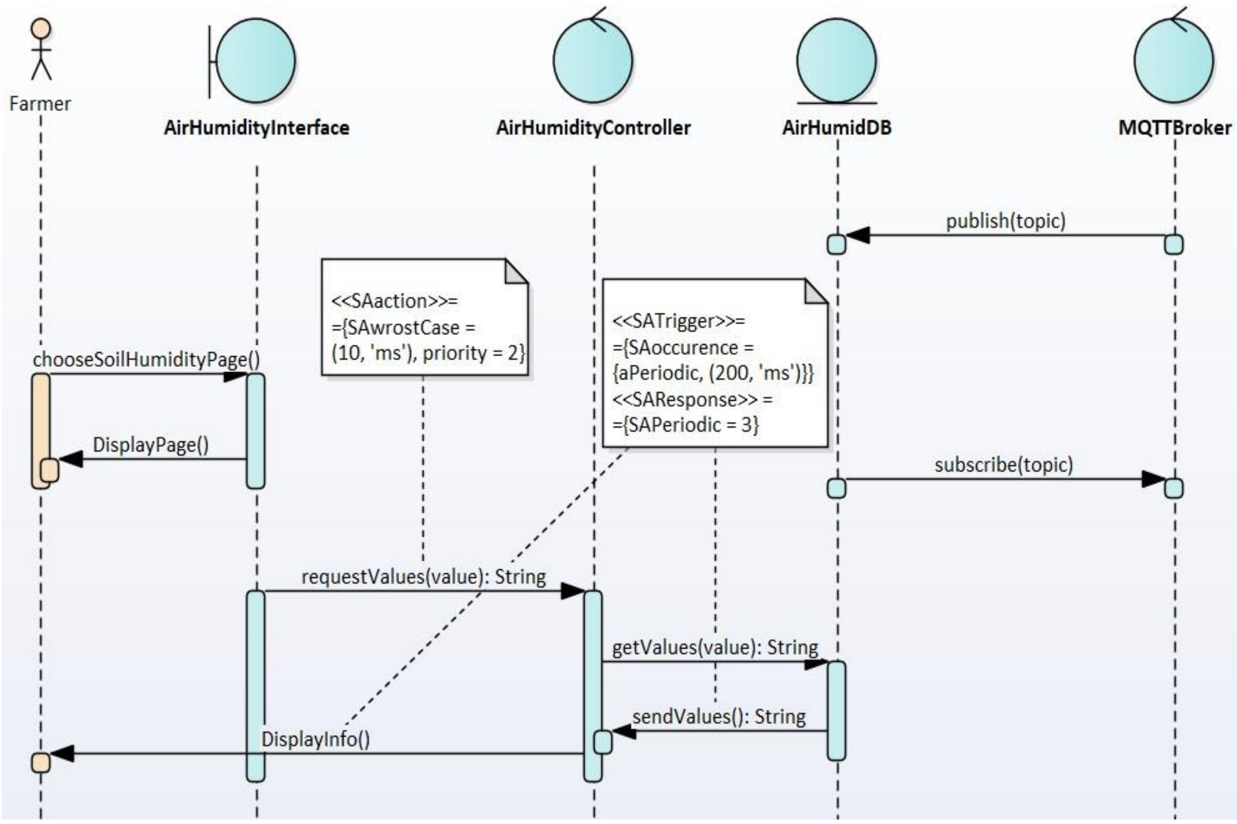
a)



SD001: Sequence diagram for Manage Node

Entity Name	View Air Humidity
Method Name	SendValues(), getValues()
Input	Click on the Air humidity interface page
Output	Display all the values of air humidity in real-time
Algorithm	1. Click on the button "Air Humidity." 2. The page will display all the values of air-humidity

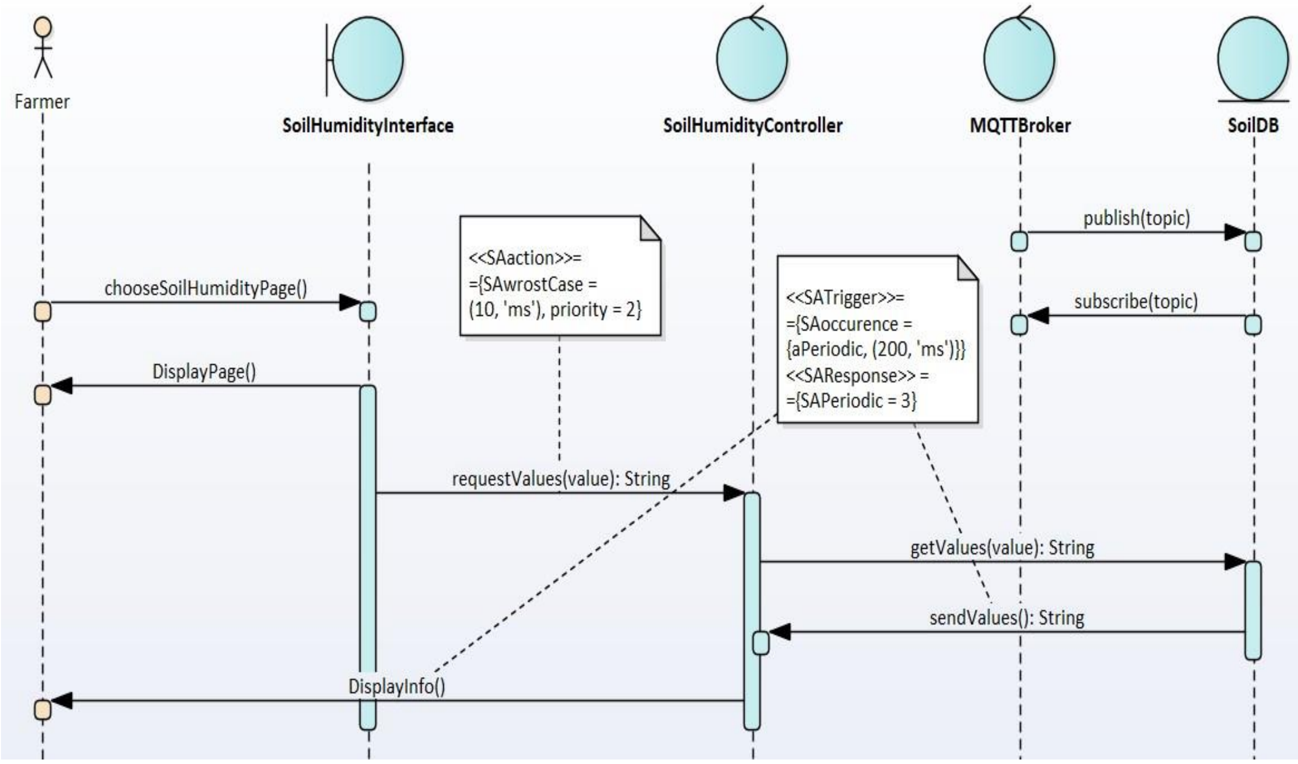
5.2.1.3 Sequence Diagrams



SD002: Sequence diagram for View Air Humidity

Entity Name	View Soil Humidity
Method Name	SendValues(), getValues()
Input	Click on the Soil humidity interface page
Output	Display all the values of air humidity in real-time
Algorithm	3. Click on the button "Soil Humidity." 4. The page will display all the values of air-humidity

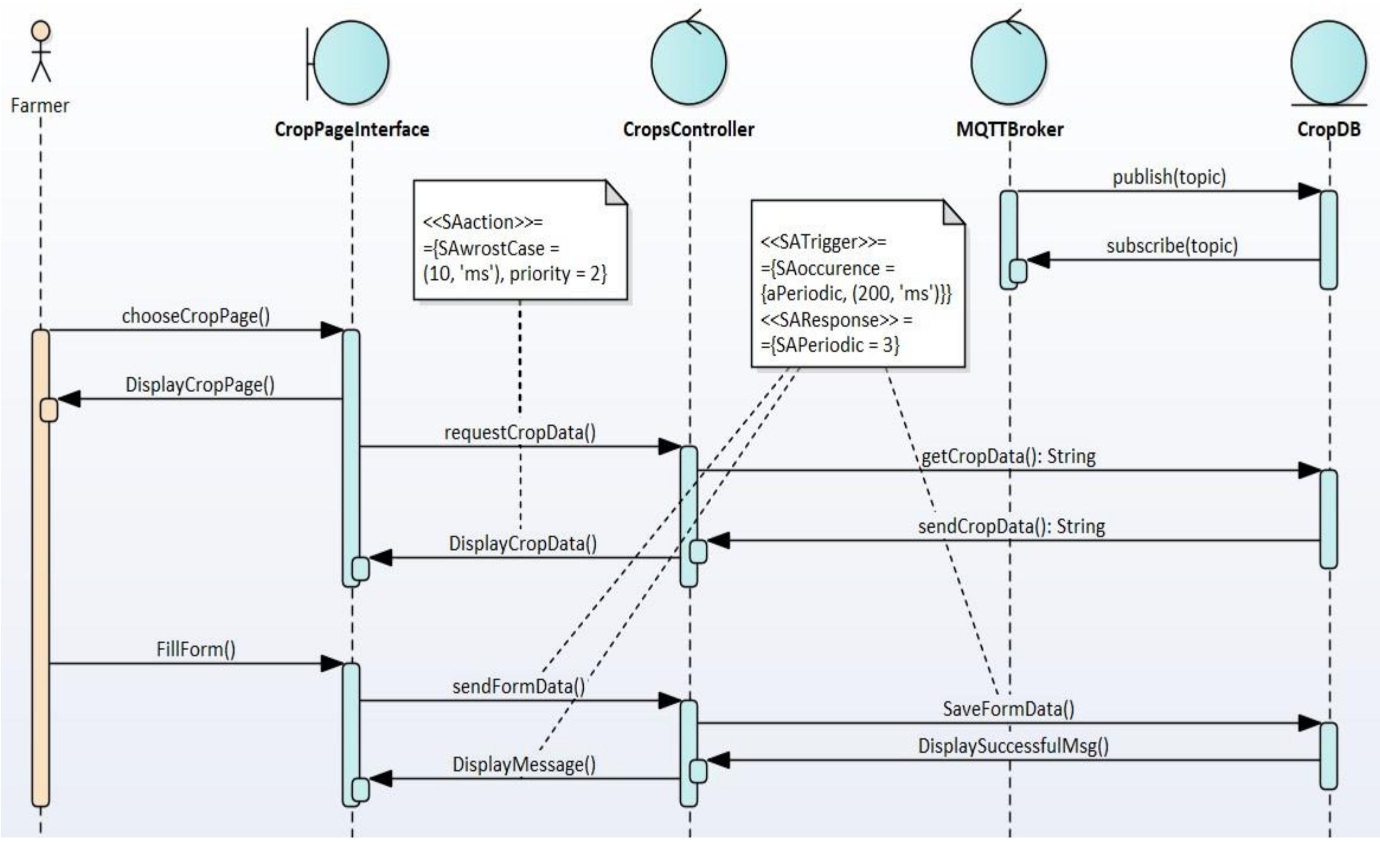
5.2.1.4 Sequence Diagrams



SD003: Sequence diagram for View Soil Humidity

Entity Name	Manage Crops
Method Name	FillForm(), saveDetails(), getinfo()
Input	Click on the Manage Crops interface page
Output	Display all the information on the particular crops added by the farmer
Algorithm	1. If the user wants to add more crops, the user will input all the details of a crop in the form and click on the submission, and it will call the FillForm() method, which will pass the values to the controller saveDetails(), and it will be saved into the database 2. The page will display all the information on active crops of a farmer by calling the method get info() which is before and after creating the form.

5.2.1.5 Sequence Diagrams



SD003: Sequence diagram for Manage Crops

6 Data Design

6.1 Data Description - ERD

The proposed Smart Monitor Farming system consists of one subsystem: The manage User subsystem. The subsystem consists of the view, domain, and data access layers. The view layer is a layer that allows the user to interact with the system. The domain layer includes controller classes related to the primary function and logic of the subsystem. The data access layer is a layer that accesses the data from the database.

6.2 Data Dictionary

Entity: NodeDB

Attribute Name	Constraints
id **int private**	default:
name **string private**	default:
status **string private**	default:
join: **Date**	default:

Operations

Methods	Notes	Parameters
getNodeDetails() string public	SD	**int** [in] id
requestEditNode() void public	SD	

Entity: AirHumidDB

Attribute Name	Constraints
value **int private**	default:
sampleDate Date **private**	default:

Operations

Methods	Notes	Parameters
getValues() string public	SD	**float** [in] value **sampleDate** [in] Date
saveValues() void public	SD	**float** [in] value sampleDate [in] Date

Entity: SoilDB

Attribute Name	Constraints
value **int private**	default:
sampleDate Date **private**	default:

Operations

Methods	Notes	Parameters
getValues() string public	SD	**float** [in] value **sampleDate** [in] Date
saveValues() void public	SD	**float** [in] value sampleDate [in] Date

Entity: CropsDB

Attribute Name	Constraints
id string **private**	default:
name **string private**	default:
crop **string private**	default:
region **string private**	default:
soilname string **private**	default:
nodename **string private**	default:
plantname **string private**	default:
status **string private**	default:
insertDate **Date private**	default:

Operations

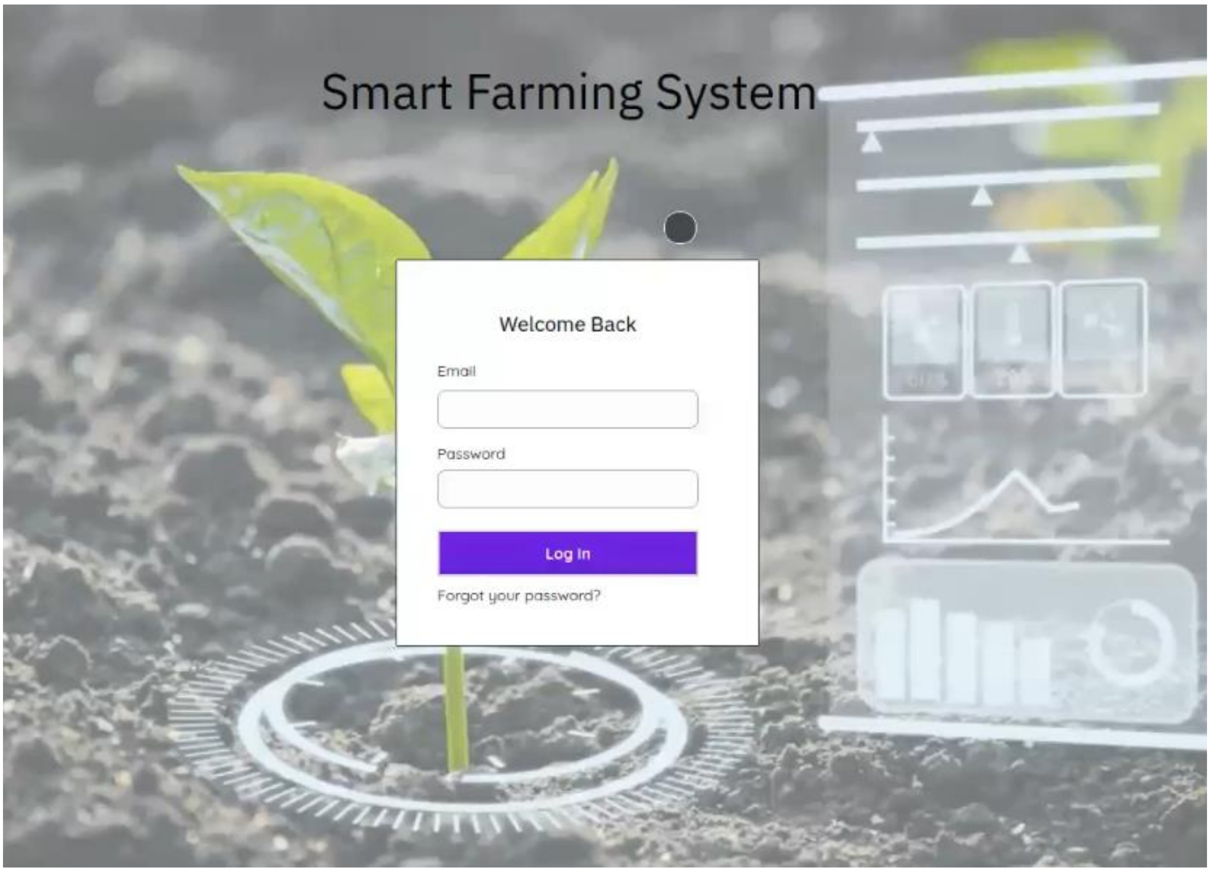
Methods	Notes	Parameters
saveDetails() string public	SD	
getInfo() string public	SD	

7 User Interface Design

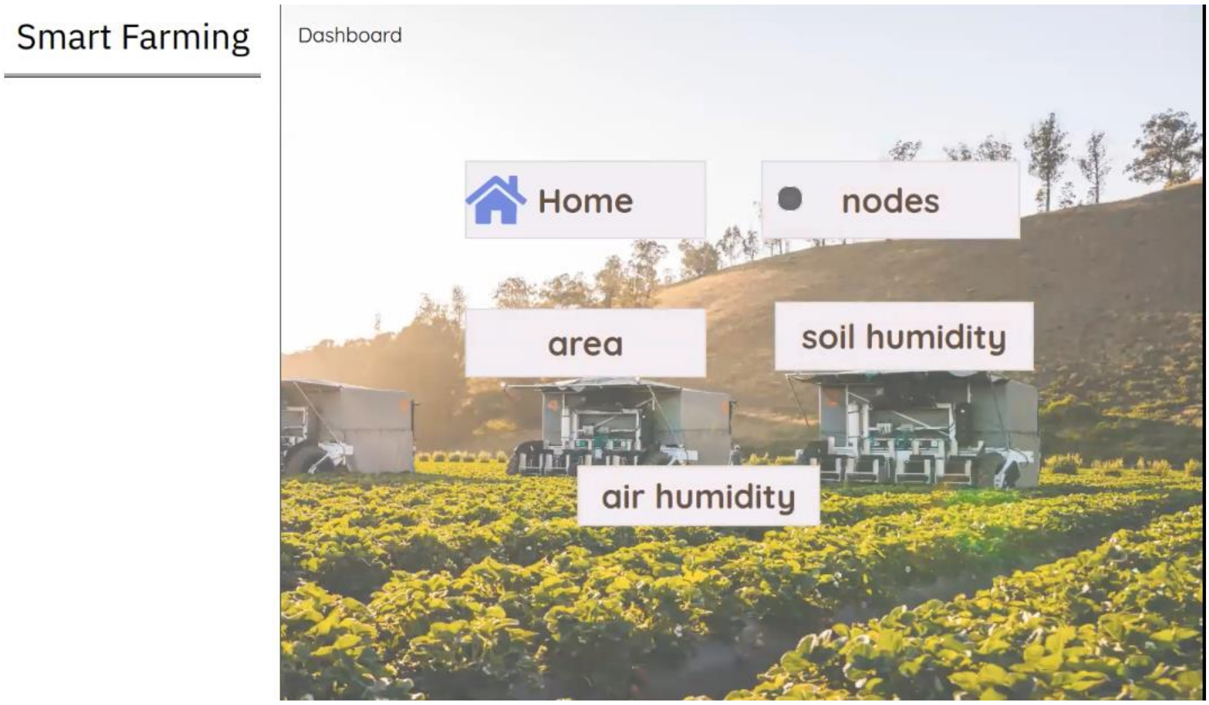
7.1 Overview of User Interface

For the Manage User subsystem, after the user logs into their account, the system will display a menu that only shows functions that are relevant to the user. The menu will show five functions: Node interface, Air Humidity interface, Manage Crops interface, Graph interface, and Soil humidity interface.

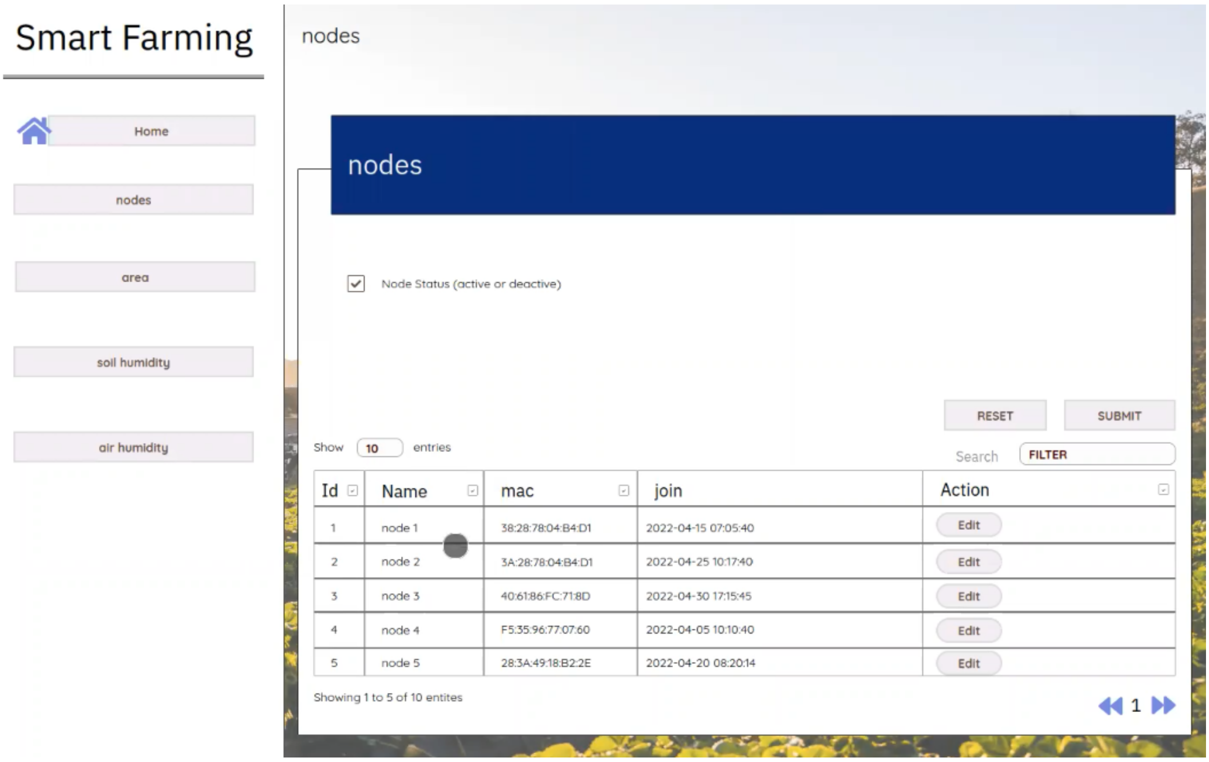
7.2 User Interface



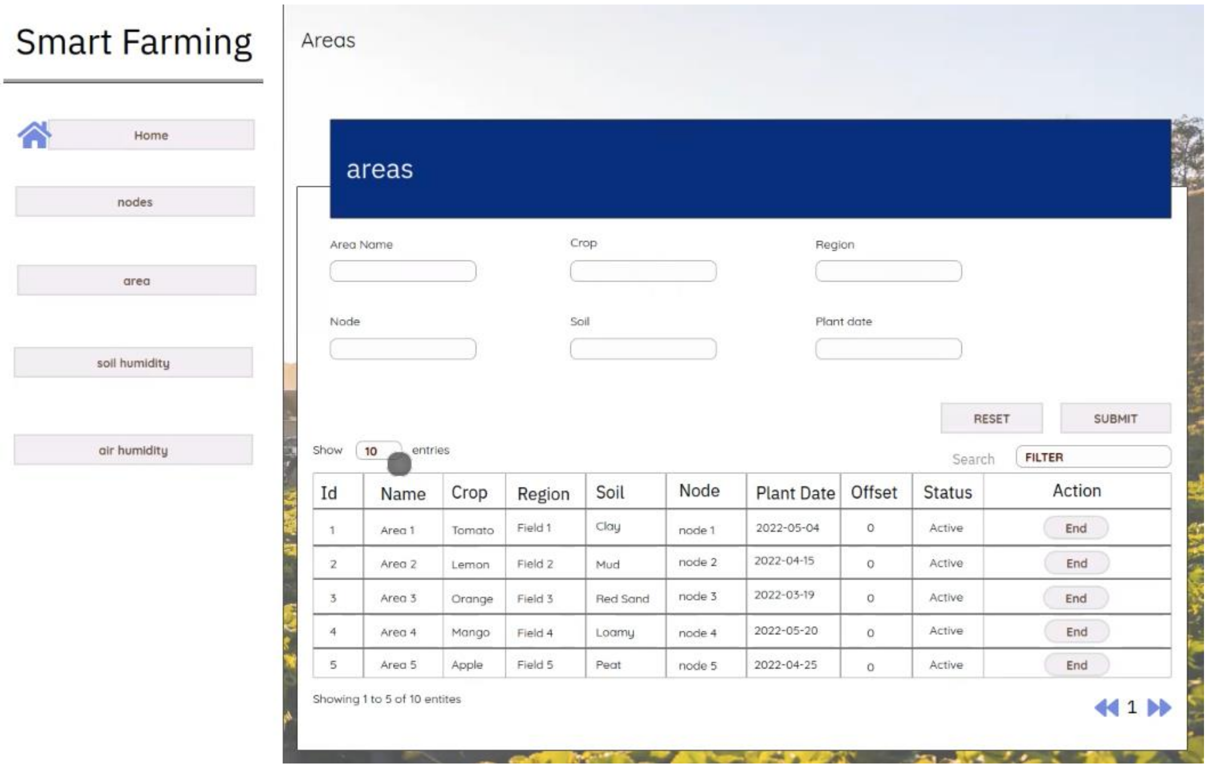
Login Interface



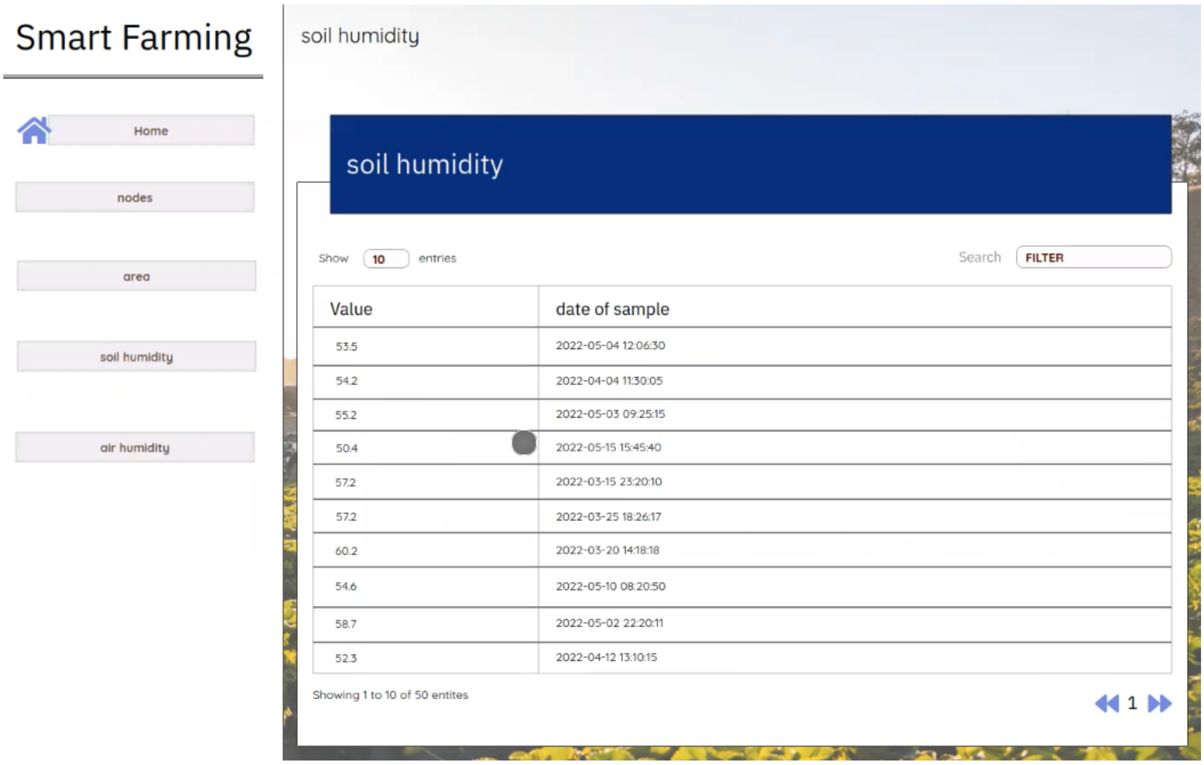
Home page interface



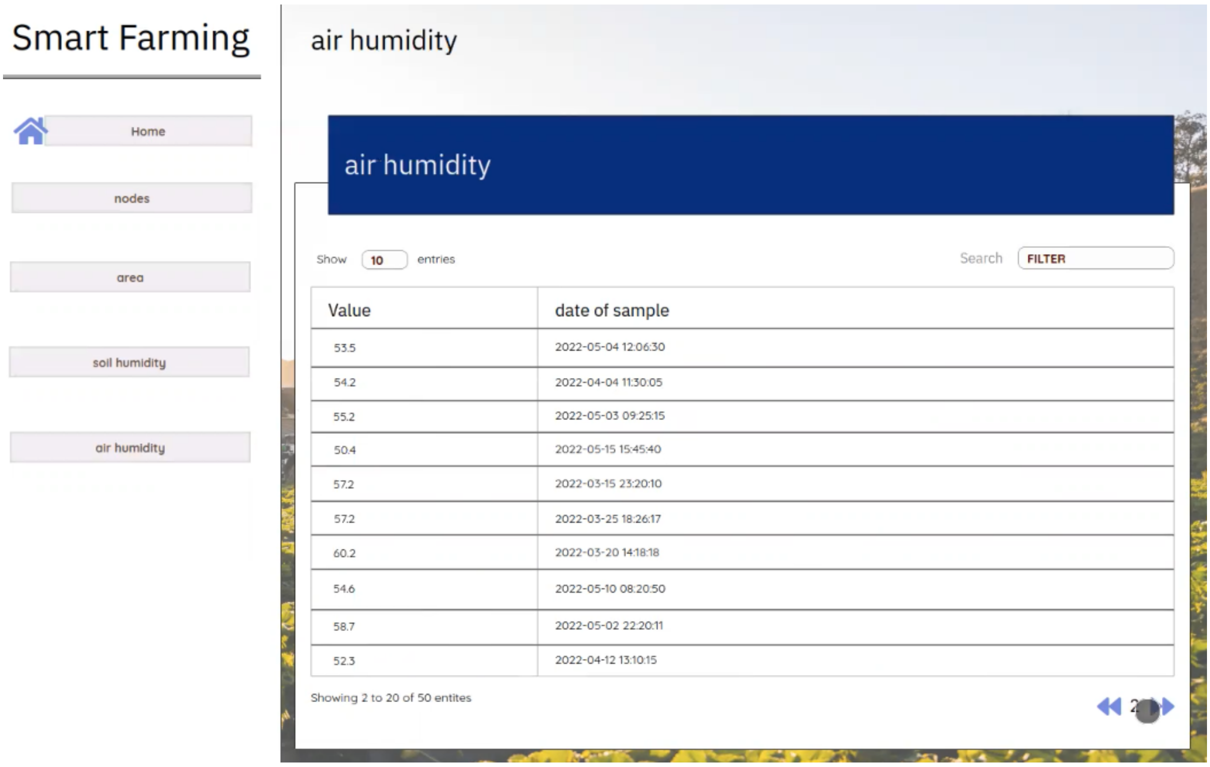
Node page interface



Areas interface(Manage Crops)



Soil Humidity interface



Air Humidity interface

8 Requirements Matrix

The use cases and respective packages are as below.

	P001			⋮
Module 1, UC001	X			
Module 1, UC002	X			
Module 1, UC003	X			
Module 1, UC004	X			
Module 1, UC005				

The packages and respective sequence diagrams for the scenarios are as below.

	SD001	SD002	SD003	SD004
P001	X	X	X	X

Appendices (if any)