Automated Greenhouse Irrigation System for Quality Seeds

P. A. Isuru Udayanga

2021

Automated Greenhouse Irrigation System for Quality Seeds

A thesis submitted for the Degree of Master of Information Technology

P. A. Isuru Udayanga
University of Colombo School of Computing
2021

DECLARATION

The thesis is my original work and has not been submitted previously for a degree at this or any other university/institute.

To the best of my knowledge, it does not contain any material published or written by another person, except as acknowledged in the text.

Student Name: P. A. Isuru Udayanga
Registration Number: 2018MIT035
Index Number: 18550352

________________________ 04.09.2021
Signature Date

This is to certify that this thesis is based on the work of P. A. Isuru Udayanga under my supervision. The thesis has been prepared according to the format stipulated and is of an acceptable standard.

Certified by:

Supervisor Name: Ms. C.R Wijesinghe

________________________ ____29/11/2021___________
Signature Date

ACKNOWLEDGEMENTS

I'd like to express my gratitude to everyone who has helped with this project in any way. My sincere thanks to my supervisor, Ms. C.R Wijesinghe, Senior Lecturer at the University of Colombo, Sri Lanka, for her meticulous attention, kindness, and generosity cannot be adequately expressed in words. Her informative remarks and perceptive advice had a big impact on the current research. Since the beginning of the endeavor, she has supported me in every way possible.

In addition, I'd want to convey my heartfelt gratitude to the Hayleys Agriculture Staff's professional team for their assistance and direction. I'd like to thank Mr. Mario Dias, Hayleys Agriculture's Assistant Manager, in particular, for sharing his expertise, experience, and practices during the development of this System.

Thank you to all of the MSc in IT degree program classmates at the University of Colombo who provided helpful feedback to help enhance the project's outcomes. In addition, I'd want to express my gratitude to my office colleagues for their contributions to the system's development and, in particular, for the additional assistance they gave.

ABSTRACT

Watering is the most common activity and the most labor-intensive job in everyday farming or gardening activities. You want to be able to control the amount of water that reaches your plants no matter what the weather is like, whether it's too hot and dry or too overcast and wet. This Automated Greenhouse Irrigation System could be useful for watering plants when they are in need. The Arduino Mega microcontroller has been used to control the water pump and Read data from Soil Moisture and Humidity sensors which are placed in the greenhouse. It is programmed to detect humidity, temperature, and moisture to regulate the amount of water required by the plant, release water from the tank, and keep the moisture level constant. The system is set up to feed all sensor readings to the main ERP system for further analysis about the plants and their behavior for the environment change.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ACRONYMS

IDE	- Integrated Development Environment
ERP	- Enterprise resource planning
SAP	- System Applications and Products
UI	- User Interface
AVR	- Automatic Voltage Regulator
ISM	- Industrial, Scientific and Medical
LE	- Low Energy
BR	- Basic Rate
EDR	- Enhanced Data Rate
SQL	- Structured Query Language
DB	- Database
RAM	- Random Access Memory
API	- Application Programming Interface
FCL	- Framework Class Library
CLR	- Common Language Runtime
OOP	- Object-Oriented Programming
ABAP	- Advanced Business Application Programming
MVC	- Model View Controller
XML	- Extensible Markup Language
UML	- Unified Modeling Language

Chapter 1 Introduction

1.1.Project Overview

Farmers are now having severe problems irrigating their crops in the sector of agriculture. It's because the availability of power isn't properly approached. They need to pump water until the field is properly irrigated, even if it is available, and they must stop doing other things that are vital for them too, and use their energy and time. But there is a solution - “Automated Greenhouse Irrigation System for Quality Seeds” benefits farmers and others who irrigating their gardens as well.

The occurrence of the disease and the plant transpiration should be reduced to high relative humidity (above 80%-85%). Sufficiency of ventilation or subsequent heating and ventilation on plant surfaces and greenhouse structures can prevent condensation. During warmer months the usage of a cooling system increases the humidity of the greenhouse air. Humidity regulation inside the greenhouse may be problematic during seasons under warm and humid external circumstances. The evaporative cooling system is a greenhouse positioned on dry sites, as a considerable volume of water can be evaporated into the input air, which leads to significant decreases in the temperature.

This automated irrigation system senses the moisture content of the soil and automatically turns the pump when the power is on. An appropriate irrigation routine is of great importance because the primary reason for this is the absence of land-saved water owing to the lack of rain. We utilize this automatic watering system for this purpose, and in all climate settings, the system is highly useful.

Because relative humidity alone cannot tell us anything about the absolute water holding capacity of air, a separate measurement is occasionally employed to assess the soil's absolute moisture state. The vapor strength deficit is a measure of the difference between the humidity the air holds at a given time and the moisture quantity it can carry when the air is saturated.

Pressure deficit assessment can help us to see how easy it is for plants to sweat more key values that stimulate sweat, but that too high values produce sweating, prevent sweat and can begin to focus on leaf and greenhouse surfaces.

This project has three functional components. Moisture sensors, humidity sensors, and a motor/water pump are involved. The moisture sensor detects the level of moisture in the soil, whereas the humidity sensor detects the level of water in the air. Water is supplied to the plants via the water pump.

The Arduino Mega 2560 is used in this project to control the flow of water through the motor in the field. Connect the microcontroller to the motor driver and the driver to the water pump using the schematic. A 12-volt battery can power the motor, and current measurements show that the battery life is adequate. The Arduino IDE software is used to program the microcontroller. The moisture sensor detects moisture in the soil, while the humidity sensor detects moisture in the air and sends a signal to the microcontroller if watering is required. The water is supplied to the plants by the motor/water pump until the necessary moisture level is reached. Irrigation techniques have advanced significantly because of technological advancements. The use of technology in irrigation has proven to be quite beneficial, as it provides efficiency and accuracy.

1.2.Background of the Study

For over three decades, “**Hayleys Agriculture**” has led the agriculture industry in Sri Lanka, and the Hayleys Agro Quality Seeds firm has been the sole F1 hybrid flower seed production and growing company in Sri Lanka, with a monthly turnover of more than 30 million rupees.

Quality Seed Co. Ltd is situated in Oluganthota, Boralanda, Srilanka and they have more than 400 employees working there when the plant pollination happens because those plants grow inside greenhouses and there are no insects or bees to do the pollination, so they do it manually using a smooth paintbrush and for other periods they have like 200 employees working there to do the watering and check the condition of plants. There is a unique thing about this company, they use only rainwater for plants throughout the year.

This company do a different kind of productions such as Hybrid Flower seeds, Open Pollinated Flower Seeds, Foliage Splits, Plugs, Linners, Growing Flowers for the local market, Grow Young Plants, etc. They mainly grow Gerbera, Roses, Petunia, Chrysanthymum as Flowering Plants and Gerbera, Roses, Babies' Breath, Lizianthus as Cut Flowers.

Hayleys Agro Quality Seeds Co. Ltd have more than 70 Greenhouses to maintain and they do not use any system for watering. Manually they do the watering for the plants. There are two main aspects to consider when doing the manual watering, time of watering and amount of water needs for plants and there is no way that we can measure the soil moisture by looking at it. Sometimes they forget to put water plants and they even forget to record how much water they added to plants, for some holidays people are not there to put water to plants. There are some Standards for plants that require this much water and fertilizer to get a better harvest. Otherwise, they won't be able to export some products if they were not able to maintain those standards and they will not able to earn that what they have planned.

So, they have a requirement of automating this process with a strong solution at a low cost that will not cost more than hiring people to do the watering.

1.3.Motivation

The major goal of this project is to automate the plant irrigation procedure. The procedure could also help to improve present manual irrigation systems. By automating this manual approach, the labour cost, as well as the money spent on the existing procedure, can be reduced. This technique provides a highly effective method of autonomously irrigating plants with minimal human interaction.

- Automate irrigation System
- Reduce the cost of greenhouse management
- Produce quality plants
- Maintain International Standards
- Provide more accurate reports

1.4.Objectives

Standalone .net Application

- This is running in a separate Computer to do the configurations and Analysis the system

Arduino Hardware Module

- This is configured using Arduino Programming to Measure Soil Moisture, Humidity and Temperature. All the Sensors are connected to Arduino Circuit Board
- Arduino is a programmable hardware platform that is used to intelligently control circuits. The Arduino board is the heart of an integrated circuit chip that can be programmed in the C++ programming language.

Integrate with Hayleys ERP System (SAP S/4 Hana)

- This needs to be done from the ERP system and there is a separate interface to communicate with .Net Application directly to transfer data to SAP ERP system for Analysis

SAP Fiori Application (Supports IOS, Android, and Web)

- Which provides analytical data to Top Management

User Manual

- Which has all the functionalities and how to configure those functionalities.

1.5.Scope of the Study

To lead towards Greenhouse Watering automation, the scope is to provide below functionalities,

Measure Soil Moisture, Humidity Level, Temperature

- Using Arduino Mega to Measure those values through Capacitive Analog Soil Moisture Sensor and Temperature and Humidity Sensor (DHT11)

Automated Alert System

- When plants have an unusual water consumption (Less or High)
- When sensors are not working
- When water tank water level is low

Crop Management

- That helps to identify which plants need to be replaced
- Within how many days we can get the harvest from certain plants

Pest / Disease Management

- Inform or Remind Employees to add recommend chemicals or do certain procedures to eliminate or reduce different types of unwanted creatures such as cockroaches, ants, wasps, spiders, silverfish, etc.

Report generation- Daily/Monthly Water Consumption Report

- Which provides plant wise Reports that can use to analyze the water consumption

Analysis of the Plants life cycle using the above measurements and predict the best conditions for plants

Integrate with Hayleys PLC ERP System (SAP S/4 Hana)

- They can see all details which are capturing through the Watering System to generate new reports which need those data

Develop an Analytical, Mobile and Web Application using the Fiori Application which is provided by SAP (Supports IOS, Android, and Web)

- These apps can display very user-friendly outputs for the users that help to make decisions about the company and the performance of employees. Mostly Analytical Apps are used by Top Management.

1.6.Structure of the Dissertation

This report depicts the progress of the ongoing implementation-based project. The first chapter contains an introduction and summary of the project, emphasizing the problem area, why the motivation to perform this project, and what the project's objectives and scope are. The background and a comparison of similar systems based on this concept are covered in Chapter 2. Chapter 03 outlined the approach and technologies used, as well as a detailed discussion of the system's design and analysis, as well as implementation details. The methodologies employed in the implementation, as well as the outcomes gained with the suggested system, are evaluated in Chapter 4. Lastly, Chapter 05 analyzes the outcome and future solutions advancements.

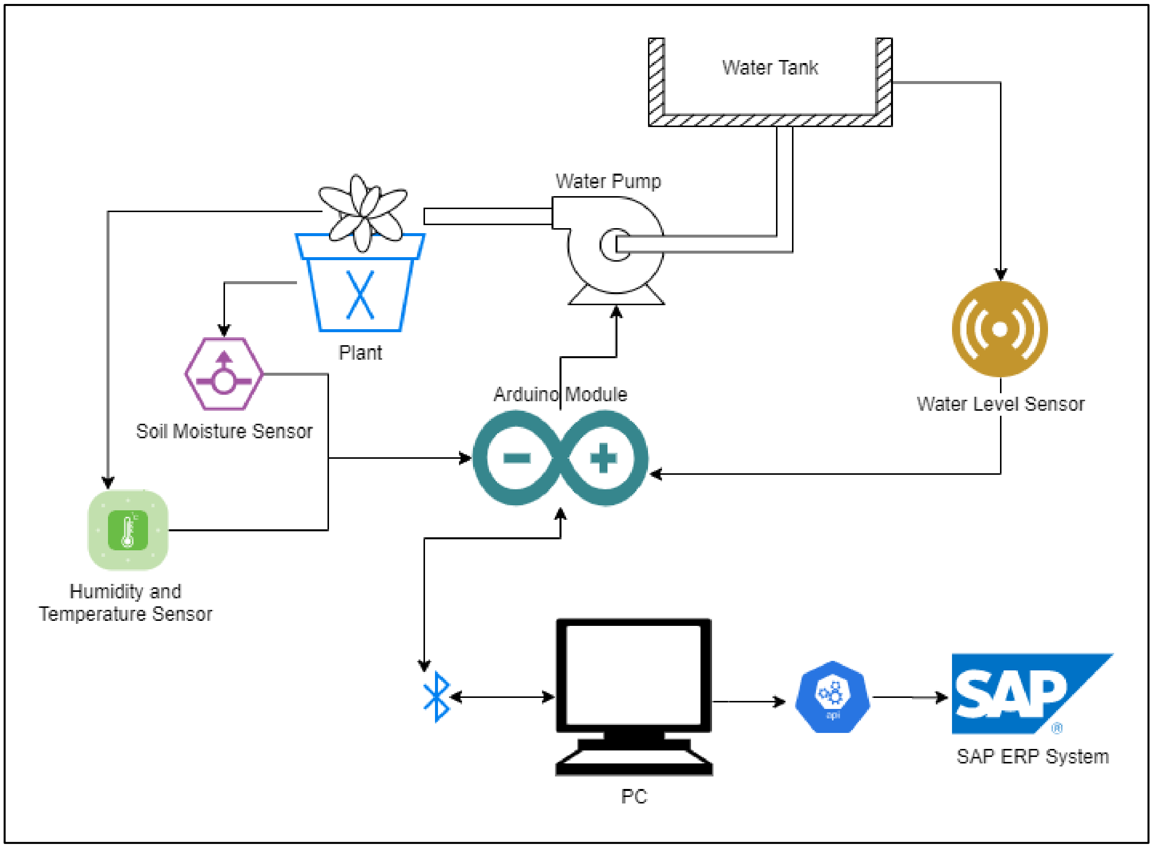


Figure 1.1 High-level System Design

Chapter 2 Background

2.1.Introduction

A summary of the business background and an overview of the current system and its disadvantages is provided in this section.

Hayleys Seeds and Flowers Division, a subsidiary Of Hayleys PLC, has more than three decades of expertise and is a prominent F1 hybrid flower seed manufacturer. Its clients are the world's leading seed producers, some even rank among the world's top five seed breeders. Their staff are very skilled and competent, with highly specialized services like hand pollination as hybrid F1 flower seed production. Over the years they have created allegiance to perfect dependability and confidence, which makes them the first opinion of the first breeders of seed

.

- Specialists in cost-competitive F1 Hybrid flower seed production
- Over three decades of Industry Experience
- Skilled and qualified Labor force
- Good processes and practices ensure high germination rates for our seeds
- A high degree of reliability & credibility

2.2.Requirement Analysis

2.2.1. Functional Requirements

This is the AS-IS system of Hayleys Quality Seeds and the Current system is a fully manual process.

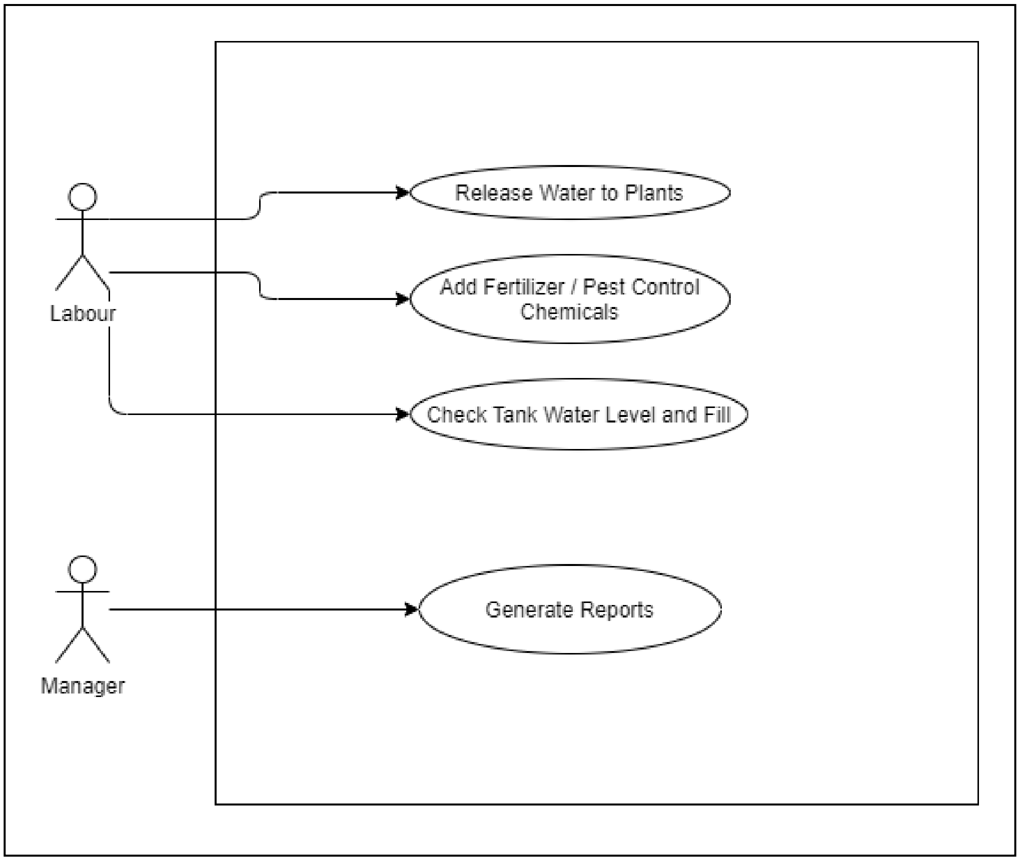

Labour
Manager
Release Water to Plants
Add Fertilizer / Pest Control Chemicals
Check Tank Water Level and Fill
Generate Reports

Figure 2.2.1 AS-IS System Use Case

All of the functions that are operating in the company is manual and they want to Automate all the operation and minimize the human interaction.

- **Release Water to Plants.**
  - Watering plants twice a day and there is no specific water amount for plants. Sometimes they forget to water plants. There is no specific procedure to do the watering and that will affect the final harvest or Product.

- **Add Fertilizer / Pest Control Chemicals**
  - Add fertilizer to plants without measuring and not on time. Pest controlling is also not effective because most of the time these chemicals will be applied to plants after the plants got infected by insects.

- **Check Tank Water Level and Fill**
  - Currently, they do this manually. The company has a massive pond that collects rainwater for plants. They will pump water from that pond to the tanks.

- **Generate Reports**
  - This will be done by the manager through an excel sheet. The manager has to collect data from labours daily and update his excel to generate a monthly report. There are no correct values in excel they used to put average numbers to the excel.

2.2.2. Non - Functional Requirements

2.2.2.1. Performance and scalability

This is all about no many delays during the system transactions and with more users how the system will behave. With No of Greenhouses how this system will perform and capture data from sensors and push to the ERP system.

2.2.2.2. Reliability

How the system will work without any failure on any condition. If something goes wrong system mean time between failures should be very low.

2.2.2.3. Availability

This describes how the system is available for the user at any time without any interruption and also access the system from anywhere.

2.2.2.4. Maintainability

This describes how much time takes to do system upgrades, Bug Fixes. If some component goes wrong the other components should be working properly and easy to get that component back online.

2.2.2.5. Usability

Should be easy to use by the Manager or System Planner. Always try to reduce the number of clicks. UI should be much easier to use for Higher Management.

2.3.Review of Similar Systems

AUTOMATED IRRIGATION SYSTEM BASED ON SOIL MOISTURE USING ARDUINO

This project uses an Arduino to automate farm irrigation and soil moisture control utilizing a soil moisture sensor and the L293D module. When the electricity is turned on, this automatic irrigation system detects the moisture content of the soil and switches the pump on. The proper use of irrigation systems is critical since the major cause is a lack of land-reserved water owing to a lack of rain, as well as spontaneous water use, which wastes enormous volumes of water. As a result, we employ this automatic plant watering and soil moisture monitoring system, which is quite useful in all climates. India is a predominantly agricultural country. Irrigation becomes difficult in dry locations or when rainfall is scarce. As a result, it must be mechanized for proper plant watering and managed remotely by a farmer. When the land becomes dry, the pump will begin to irrigate it. The goal of the implementation is to reduce water consumption, and automatic irrigation can be employed to save time and power. The goal of this project's implementation was to show how autonomous plant irrigation may be used to save water and time. [9].

AUTOMATIC PLANT WATERING SYSTEM

This project was undertaken because India is primarily an agricultural country, and the rate at which water resources are dwindling poses a serious concern, necessitating the development of a smart and efficient irrigation system. We implemented sensors in this project that detect humidity in the soil (agricultural field) and deliver water to the field that requires it. The project uses a PIC16F877A microcontroller to regulate the water supply and the irrigation field. Sensors are located in each field; however, they are not active until there is water on the field. When the field becomes dry, sensors detect the need for water and transmit a signal to the microcontroller. Until the sensors are disabled again, the microcontroller distributes water to that particular field that has a water requirement. If there are multiple signals for water requirements, the microcontroller will prioritize the first signal received and irrigate the fields accordingly [10].

AUTOMATED IRRIGATION PROJECT

In the absence of a human, an automatic watering system provides a solution for watering the lawn and garden automatically. Watering does not require individual presence because the system is automated and takes into account one or more of the different irrigation schedule factors available, such as soil moisture monitoring, evapotranspiration predictions, leaf water potential canopy temperature, and so on. This preserves the lawn's greenery while also providing the right amount of water for the garden. Irrigation systems based on soil moisture, timer-based irrigation systems, drip irrigation, sprinkle irrigation, and so on. are the various types of irrigation systems that can be automated. Moisture-based technology has been chosen for this stage of soil development. The system is regulated in this project by a soil moisture sensing sensor that regulates the flow of water. Soil moisture sensor, control circuit, gate valve, pumping unit, timer, power supply, and programming guidance are the key components required [11].

2.4.Related Technologies

Arduino – Which needs to communicate with all the electronic components in the system.

Bluetooth – Which needs to communicate with Arduino System and PC

SQL – Which needs to communicate with DB.

.Net C# - To develop the Standalone Application

C++ - To develop Arduino System

SAP ABAP – To Develop reports in the SAP System

UI5 – To Develop the UI5 Application in the SAP System

2.5.Related Design Strategies

The whole project will be developed under Agile methodology. It's a combination of iterative and incremental process models with a focus on process adaptability and customer satisfaction. Able to project tasks into small incremental builds and those builds are provided in iterations. The agile method is easy to deliver a product with customer satisfaction.

Following Principles will be followed during the development.

- Interaction with all the individuals in the business
- Always do the demo with actual working software
- Continuous customer interaction

Chapter 3 Design

3.1.Introduction

This chapter displays the design of the system presented and details the methods and techniques used for the development of the system. Few techniques and technologies are being employed to implement this system more attractively and efficiently

3.2.System Architecture

Since the solution is based on Standalone Application, Web-based UI5 Fiori Application and Arduino Hardware Module, identified minimum hardware and software requirements are listed as follows,

Standalone Application

- Intel Core I3 Processor
- 4GB Ram
- 256 GB HDD
- Network Access
- Windows 10
- Bluetooth Connectivity
- UPS

Fiori Application (SAP)

- Mobile Phone (Android 8+ or iPhone IOS 10+)
- 2GB Ram
- 32 GB Storage
- Touch Screen

Arduino Module

- Arduino Uno Board
- Bluetooth Module
- Humidity Sensor
- Soil Moisture Sensor
- Water Level Sensor
- Water Pump
- UPS

3.3.Application Development Architecture

The Hardware module will be developed using Arduino Mega Board which has the highest number of inputs. Arduino was born as an accessible tool for rapid prototyping at the Ivrea Interaction Design Institute for students without electronic and programming backgrounds. When the Arduino board arrived in the larger community, its offering began to be distinct from simple 8-bit boards, wearables, 3D printers, embedded environments to goods suitable for IoT applications. All Arduino boards are fully open-source, enabling users to construct them independently and finally to suit their specific demands. The program is open source, too, and is expanded by users worldwide [3].

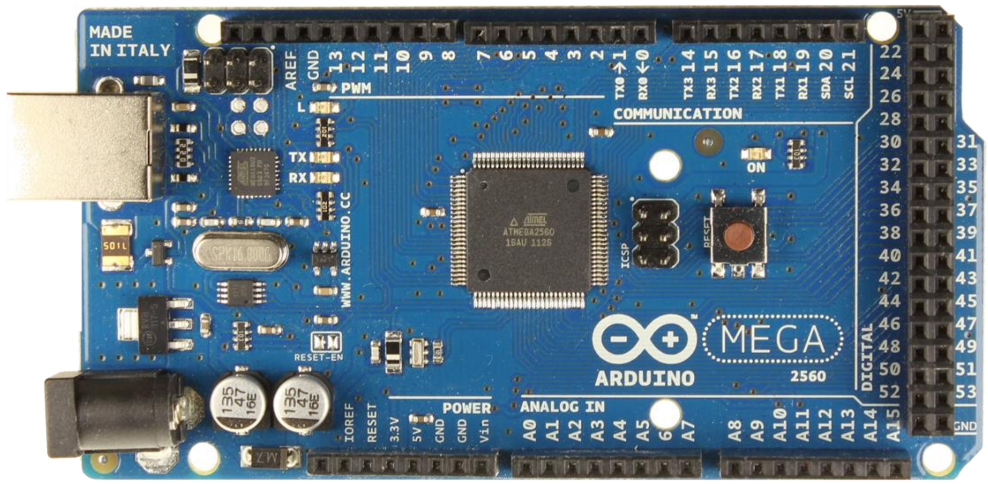


Figure 3.1 Arduino Mega Board

The Fiori Application will be developed using Model View Controller (MVC) architecture. Figure 3.2 shows the MVC architecture.

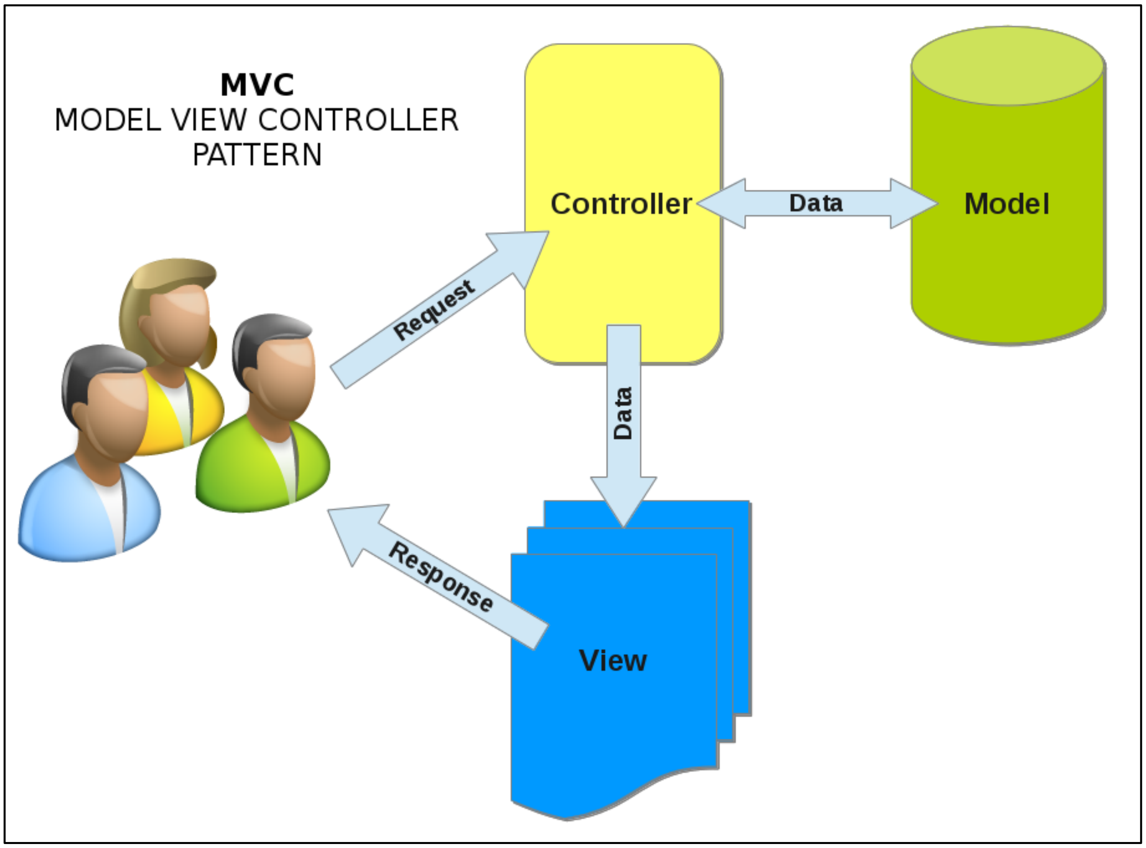


Figure 3.2 MVC Architecture

3.4.Proposed System Design

This solution is designed and integrated into the implementation in three consecutive steps.

Three phases are,

- Standalone Application Development
- Arduino Hardware Module Development
- UI5 Fiori Application Development

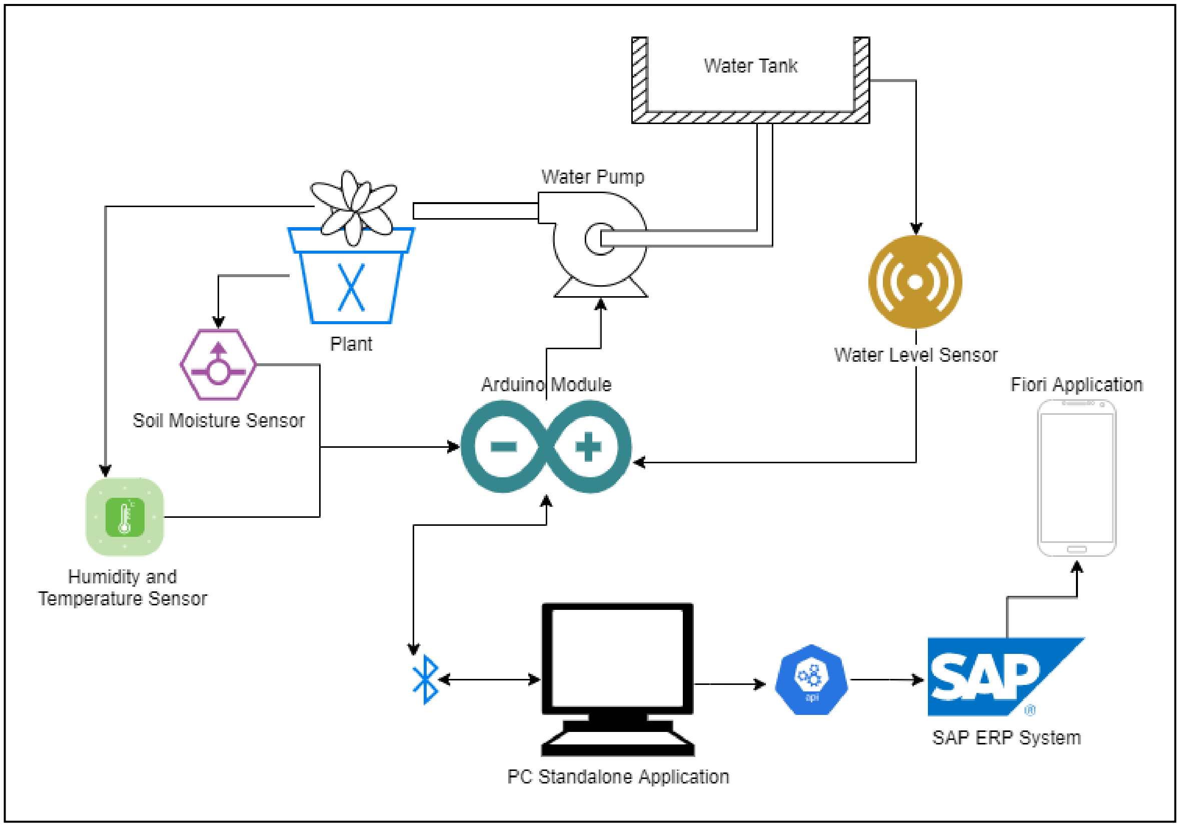


Figure 3.3 High-Level System Design

The Proposed system has two actors and one cooperating system. System Planner and Manager are the two actors and the system communicates with both the users.

Figure 3-4 below shows a graphical representation of the principal actors,

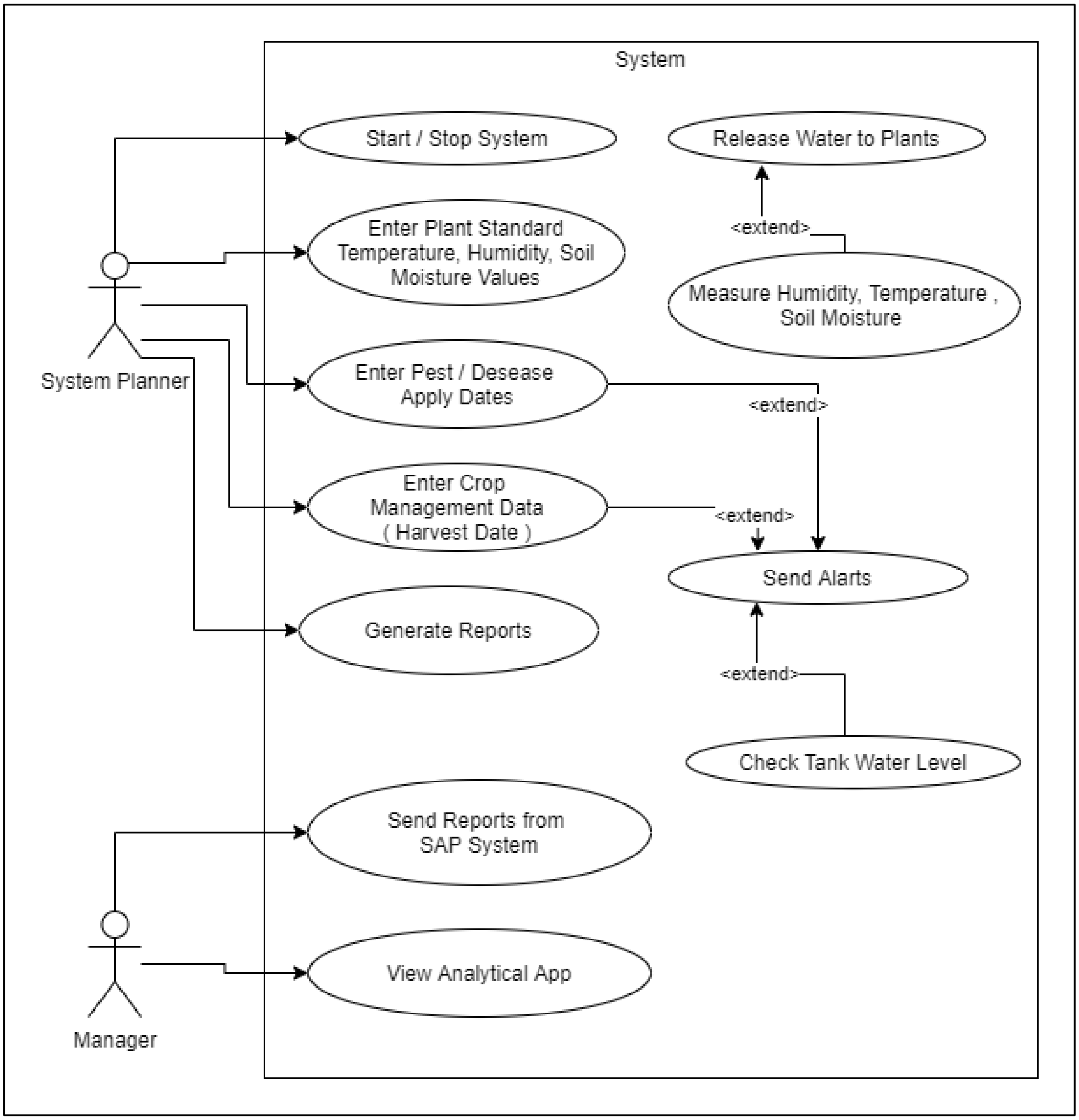

System
Start / Stop System
Release Water to Plants
<extend>
Enter Plant Standard Temperature, Humidity, Soil Moisture Values
Measure Humidity, Temperature , Soil Moisture
System Planner
Enter Pest / Desease Apply Dates
<extend>
Enter Crop Management Data (Harvest Date)
<extend>
Send Alarts
Generate Reports
<extend>
Check Tank Water Level
Send Reports from SAP System
View Analytical App
Manager

Figure 3.4 Use Case of Proposed System

In a cooperative system, the system works with two actors. The system contains the following users.

- Manager
- System Planner

The functions of the user roles can be classified as follows according to the system,

Manager

- Send Reports from SAP System
- View Analytical App

System Planner

- System Start / Stop
- Enter, Edit, Delete Plant Standard Temperature, Humidity and Soil Moisture levels to the system.
- Enter, Edit, Delete Pest / Disease chemical apply Dates
- Enter, Edit, Delete Crop Management Dates
- Generate Reports

System

- Automatically check Temperature, Humidity and Soil Moisture to release water to Plants
- Check water tank level and send alerts
- Send alerts when Harvest, Pest apply dates reached

The users will be shown their responsibility and tasks by the case of the proposed system which decreases the labour per man. This solution was broken into a few additional modules to facilitate development and maintenance.

- Plant Management
- Plant Line Management
- User Management
- Error Detection
- Pest and Crop Management
- Reports
- Alerts

The Manager role has the control of creating Users and allowing Privileges. Manager role provides below functions,

- Login

Use-Case Name	Login
Use-Case ID	01
Priority	High
Type	Enter to the System
Actors	Manager, Planner
Description	Login to System
A typical course of flow	The system displays the login screen The user should enter the username and password, If the username and password are correct, the System proceeds with the task
Alternative flow	If the username and password do not match, the user cannot log in to the system

Table 3.1 Login Use Case

- Logout

Use-Case Name	Logout
Use-Case ID	02
Priority	High
Type	Exit the System
Actors	Manager, Planner
Description	Logout from System
A typical course of flow	The user should click the logout button, then it will logout from the system

Table 3.2 Logout Use Case

- View All Records

Use-Case Name	View Records
Use-Case ID	03
Priority	High
Type	View Details
Actors	Manager
Description	Managers can view all the sensor input data from the system and also from the SAP system.
A typical course of flow	The manager can select the relevant task to view details

Table 3.3 View Records Use Case

- User Management

Use-Case Name	User Management
Use-Case ID	04
Priority	High
Type	View / Update / Delete
Actors	Manager
Description	Manager can add new users and change user details
A typical course of flow	The user should go to the user management screen. Then Manager role can Add, Change, Delete any user from the system

Table 3.4 User Management Use Case

- Generate Reports

Use-Case Name	Generate Reports
Use-Case ID	05
Priority	High
Type	View Reports
Actors	Manager, Planner
Description	Managers can generate reports according to different criteria
A typical course of flow	The user should go to reporting screen and select which report that user needs and filtration

Table 3.5 Generate Reports Use Case

System Planner role has the control of Arduino System and System Planner has the main functionalities of the system. System Planner role provides below functions,

- Login

Use-Case Name	Login
Use-Case ID	06
Priority	High
Type	Enter to the System
Actors	Manager, Planner
Description	Login to System
A typical course of flow	The system displays the login screen The user should enter the username and password, If the username and password are correct, the System proceeds with the task
Alternative flow	If the username and password do not match, the user cannot log in to the system

Table 3.6 Login Use Case

- Logout

Use-Case Name	Logout
Use-Case ID	07
Priority	High
Type	Exit the System
Actors	Manager, Planner
Description	Logout from System
A typical course of flow	The user should click the logout button, then it will logout from the system

Table 3.7 Logout Use Case

- System Start / Stop

Use-Case Name	System Start / Stop
Use-Case ID	08
Priority	High
Type	System Start / Stop
Actors	Planner
Description	System Start / Stop
A typical course of flow	The planner can start or stop the system at any time for any reason. If the planner stops the system, then it will stop all the sensor inputs and sends an alert to the Manager

Table 3.8 System Start/Stop Use Case

- Manage Plant Standard Temperature

Use-Case Name	Manage Plant Standard Temperature
Use-Case ID	09
Priority	High
Type	View / Update / Delete
Actors	Planner
Description	Manage Plant Standard Temperature
A typical course of flow	Planners can set or edit every plant line minimum and max temperature. IF the plant line temperature goes up or down, the system will automatically send an alert to the Manager or Planner

Table 3.9 Manage Plant Standard Temperature Use Case

- Manage Plant Humidity Level

Use-Case Name	Manage Plant Humidity Level
Use-Case ID	10
Priority	High
Type	View / Update / Delete
Actors	Planner
Description	Manage Plant Humidity Level

A typical course of flow	Planners can set or edit every plant line minimum and max humidity level. IF the plant line humidity goes up or down, the system will automatically send an alert to the Manager or Planner

Table 3.10 Manage Plant Humidity Level Use Case

- Manage Plant Soil Moisture Level

Use-Case Name	Manage Plant Soil Moisture Level
Use-Case ID	11
Priority	High
Type	View / Update / Delete
Actors	Planner
Description	Manage Plant Soil Moisture Level
A typical course of flow	Planners can set or edit every plant line soil moisture level depending on the plant. IF the plant line soil moisture goes down, the system will automatically be pumping water to the plant line. The planner can set the water pumping amount to the plant.

Table 3.11 Manage Plant Soil Moisture Level Use Case

- Manage Plant Line Pest chemical apply Dates

Use-Case Name	Manage Plant Line Pest chemical apply Dates
Use-Case ID	12
Priority	High
Type	View / Update / Delete
Actors	Planner
Description	Manage Plant Line Pest chemical apply Dates
A typical course of flow	Planners can set or edit every plant line chemical apply date. IF the plant line reaches the date, the system will automatically send an alert to the Manager or Planner

Table 3.12 Manage Plant Line Pest Chemical Apply Dates Use Case

- Manage Plant Harvest Dates

Use-Case Name	Manage Plant Line Pest apply Dates
Use-Case ID	13
Priority	High
Type	View / Update / Delete
Actors	Planner
Description	Manage Plant Line Pest apply Dates
A typical course of flow	Planners can set or edit every plant line harvest date. IF the plant line

	reaches the date, the system will automatically send an alert to the Manager or Planner

Table 3.13 Manage Plant Line Pest Apply Dates Use Case

- Generate Reports

Use-Case Name	Generate Reports
Use-Case ID	14
Priority	High
Type	View Reports
Actors	Manager, Planner
Description	Managers can generate reports according to different criteria
A typical course of flow	The user should go to reporting screen and select which report that user needs and filtration

Table 3.14 Generate Reports Use Case

3.5.UML Diagrams

3.5.1. Data Flow Diagram of the System

Data Flow of the proposed System

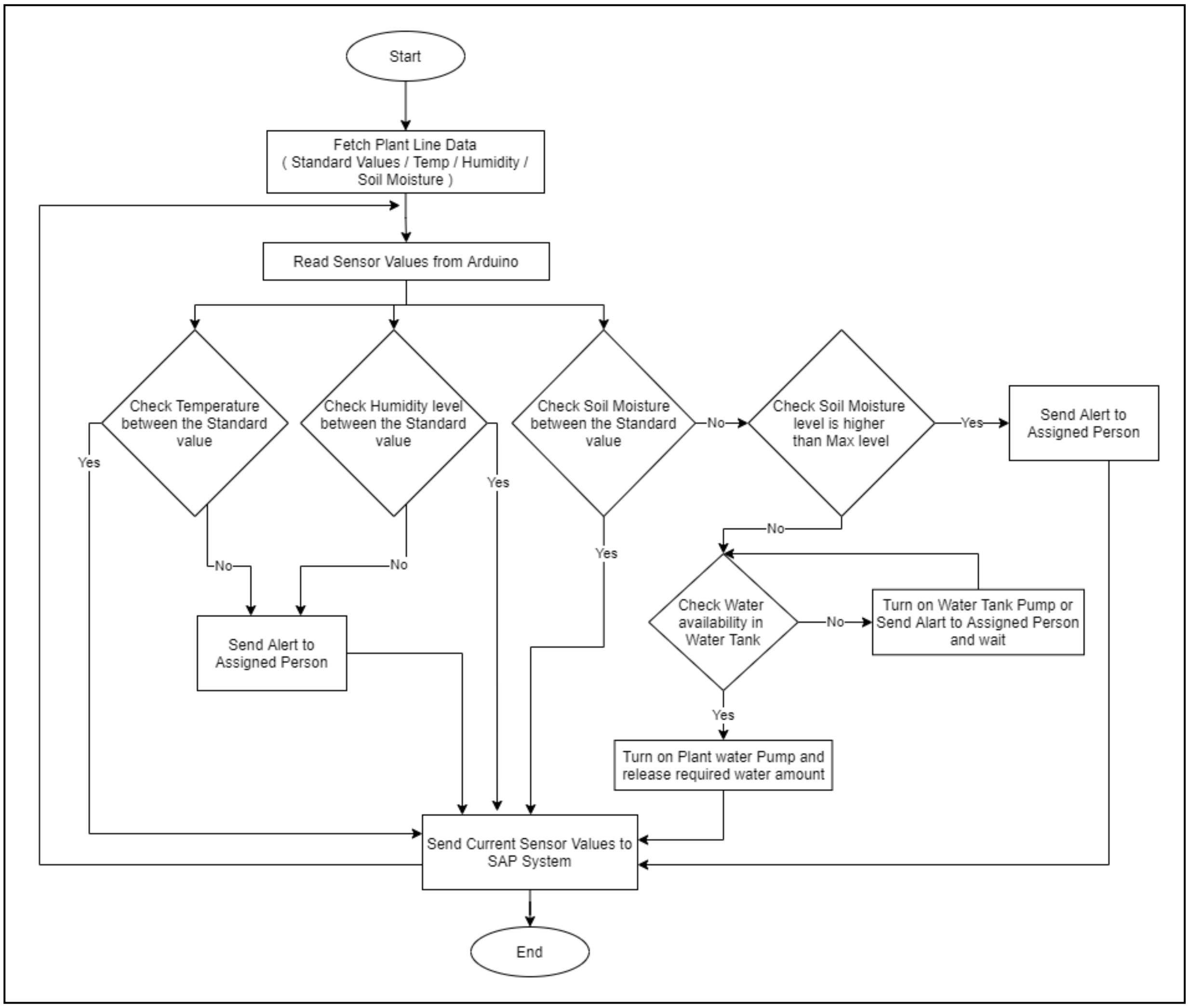

Start
Fetch Plant Line Data
(Standard Values / Temp / Humidity / Soil Moisture)
Read Sensor Values from Arduino
Check Temperature between the Standard value
Check Humidity level between the Standard value
Check Soil Moisture between the Standard value
No
Check Soil Moisture level is higher than Max level
Yes
Send Alert to Assigned Person
Yes
Yes
No
No
Yes
No
Check Water availability in Water Tank
No
Turn on Water Tank Pump or Send Alart to Assigned Person and wait
Send Alert to Assigned Person
Yes
Turn on Plant water Pump and release required water amount
Send Current Sensor Values to SAP System
End

Figure 3.5 Data Flow Diagram of the System

3.5.2. Database Design of the System

Database design is the database model structuring of data. Displaying which data must be saved and the interrelationship of data items. Database conception is used for specifying the structure and SAP system/database conception of business items that use the system. The following Entity Relationship Chart explains the structure of the Standalone application and SAP system relational database schema.

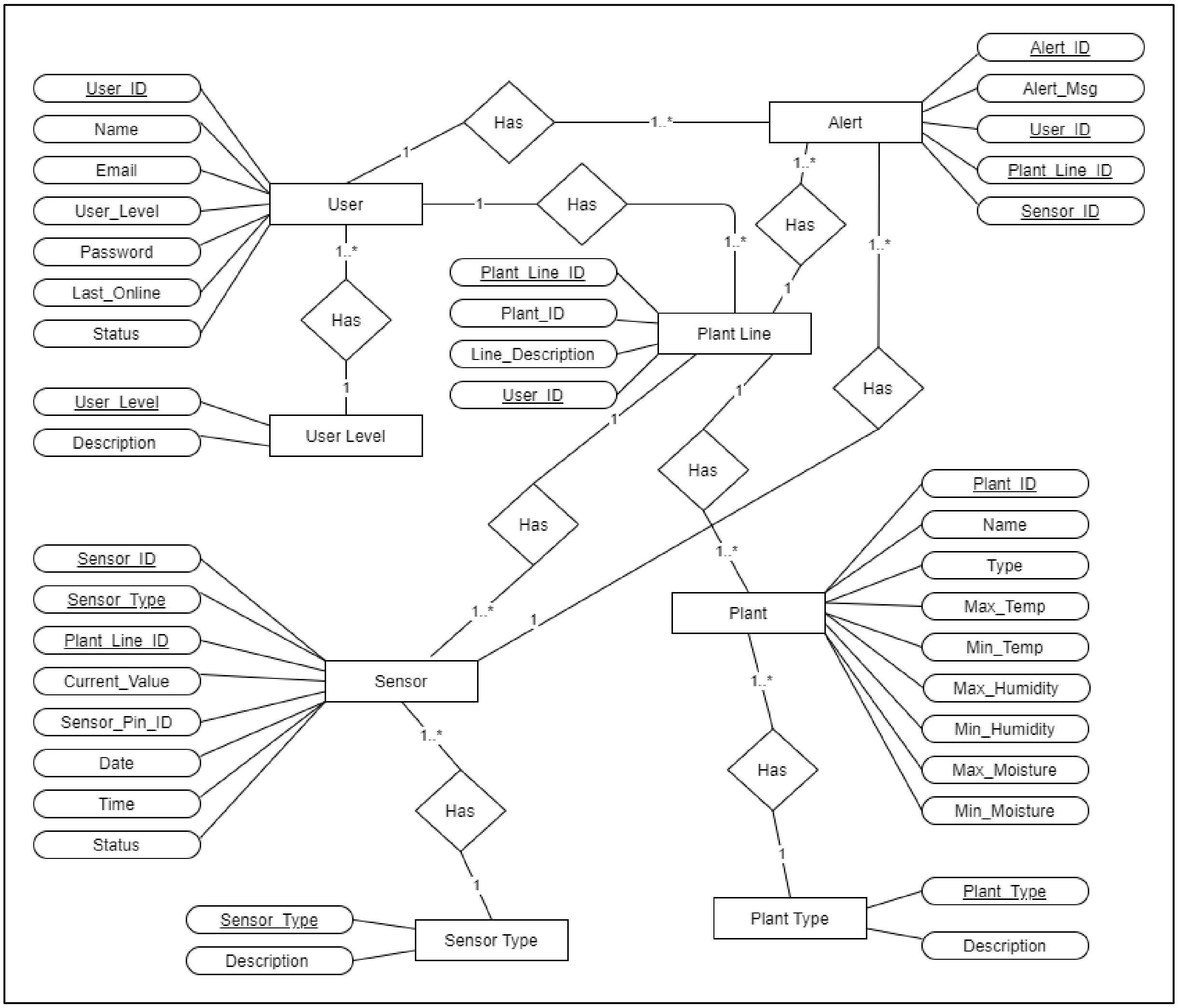


Figure 3.6 ER Diagram

3.5.3. Class Diagram of the System

A Unified Modeling Language (UML) class diagram is a form of a static structure diagram that defines the structure of a system by illustrating the classes of the system, its characteristics, its actions (methods) and the interaction between objects. The graph below shows the system's overall class system.

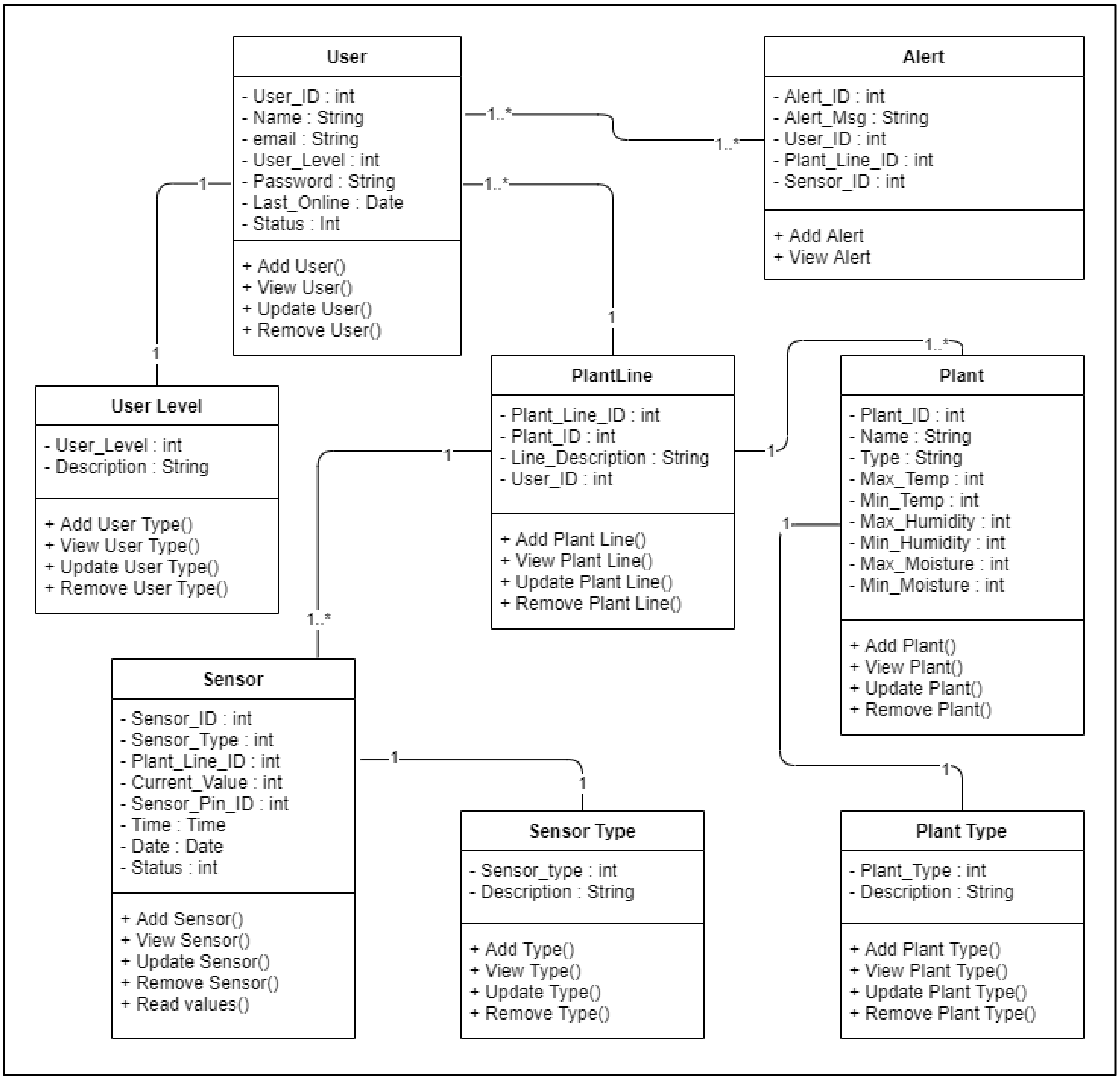


Figure 3.7 Class Diagram

3.5.4. Sequence Diagram of the System

Following tasks can be handled by a manager role:

- Send Reports from SAP System
- View Analytical App

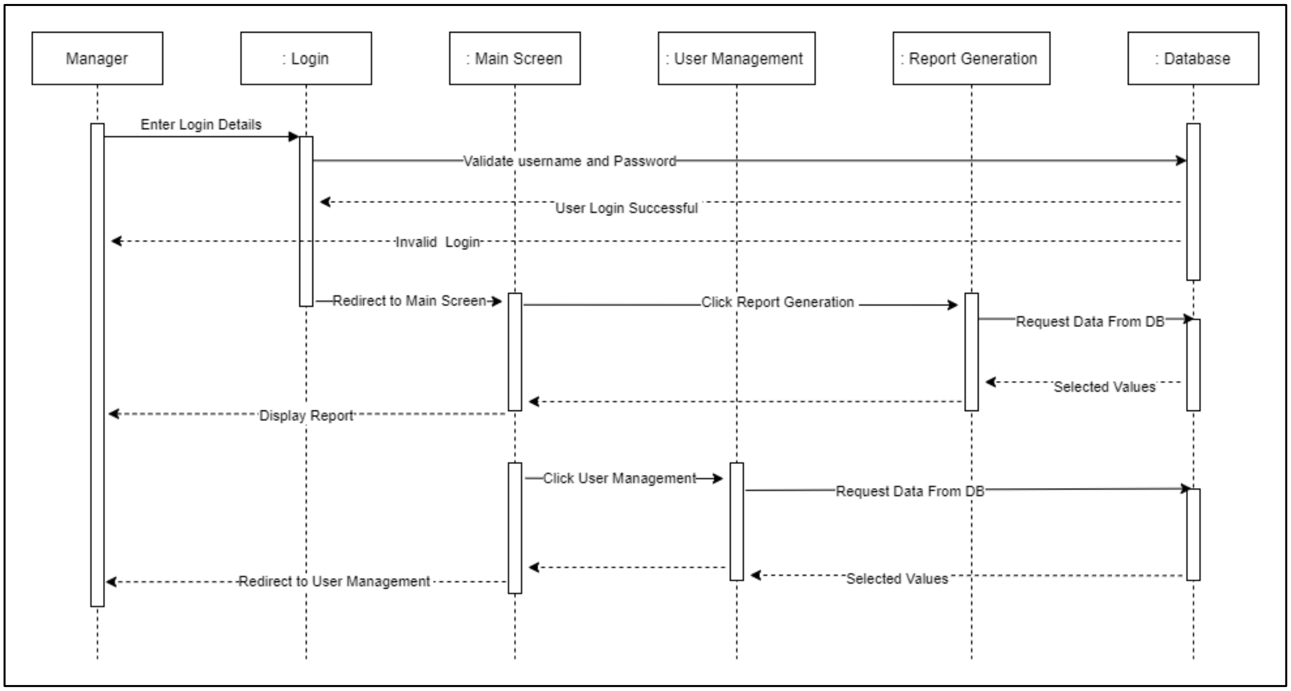


Figure 3.8 Sequence Diagram: Manager

Following tasks can be handled by a Planner role:

- System Start / Stop
- Enter, Edit, Delete Plant Standard Temperature, Humidity and Soil Moisture levels to the system.
- Enter, Edit, Delete Pest / Disease chemical apply Dates
- Enter, Edit, Delete Crop Management Dates
- Generate Reports

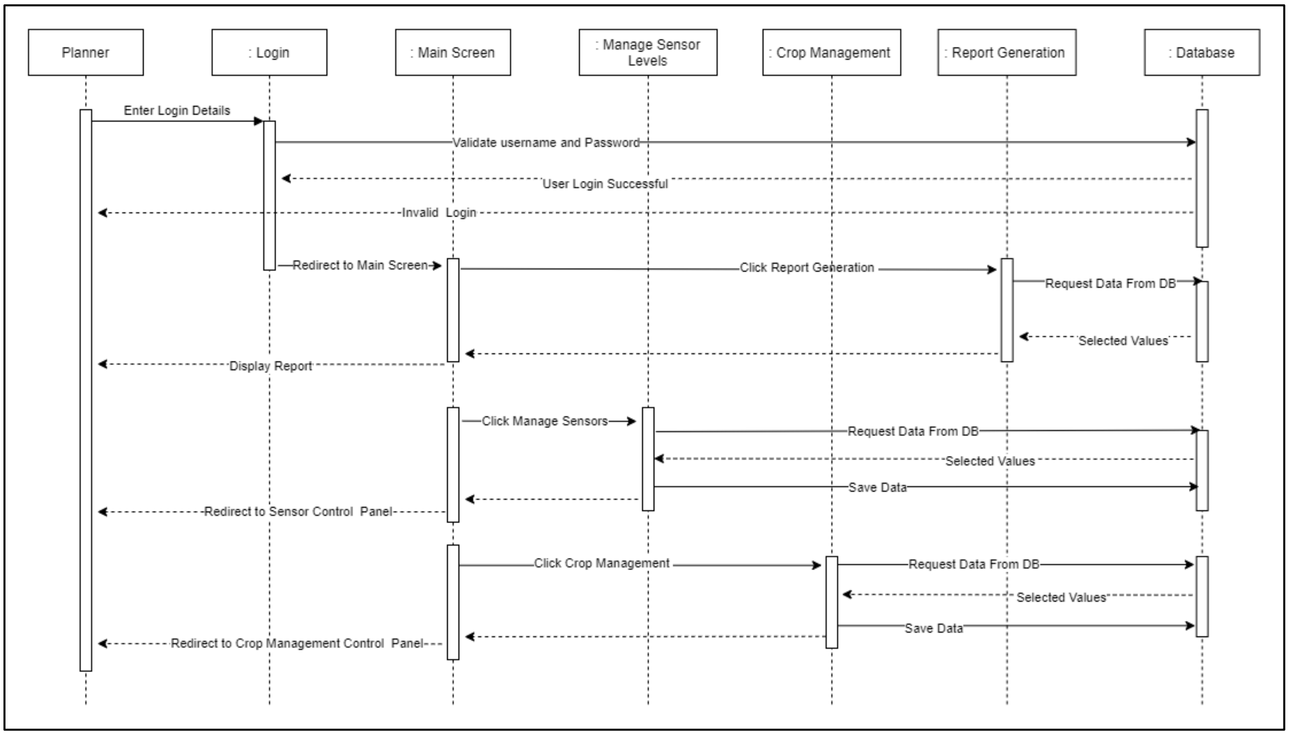


Figure 3.9 Sequence Diagram: Planner

3.6.Main User Interfaces

This section disputes the system interface structure with a few key user interfaces.

Standalone Application

- Login Screen

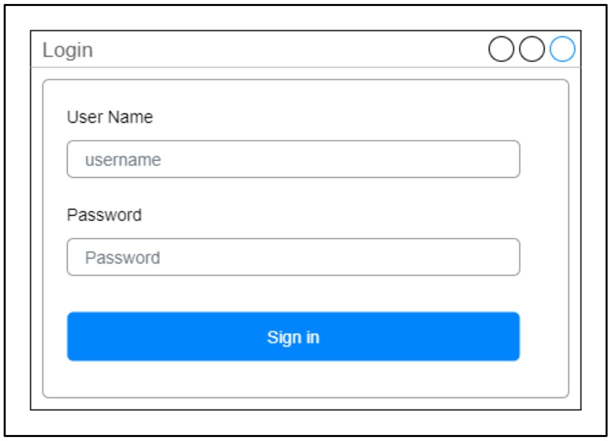


Figure 3.10 Standalone Application Login

- Main Screen – Manager

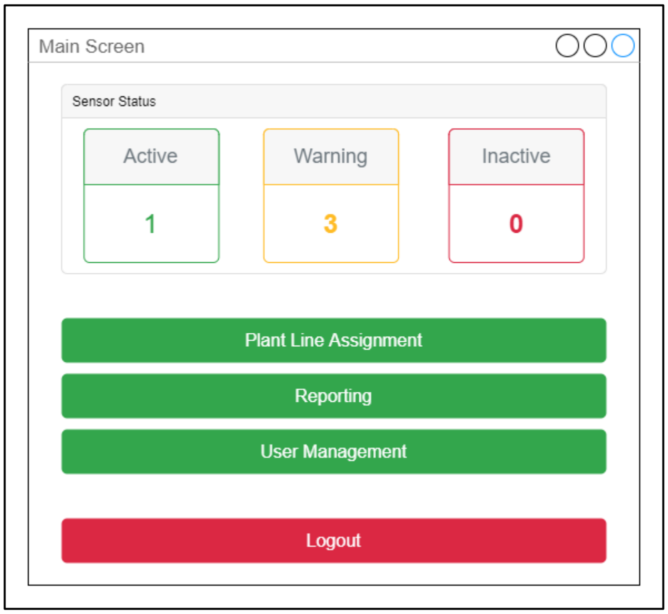
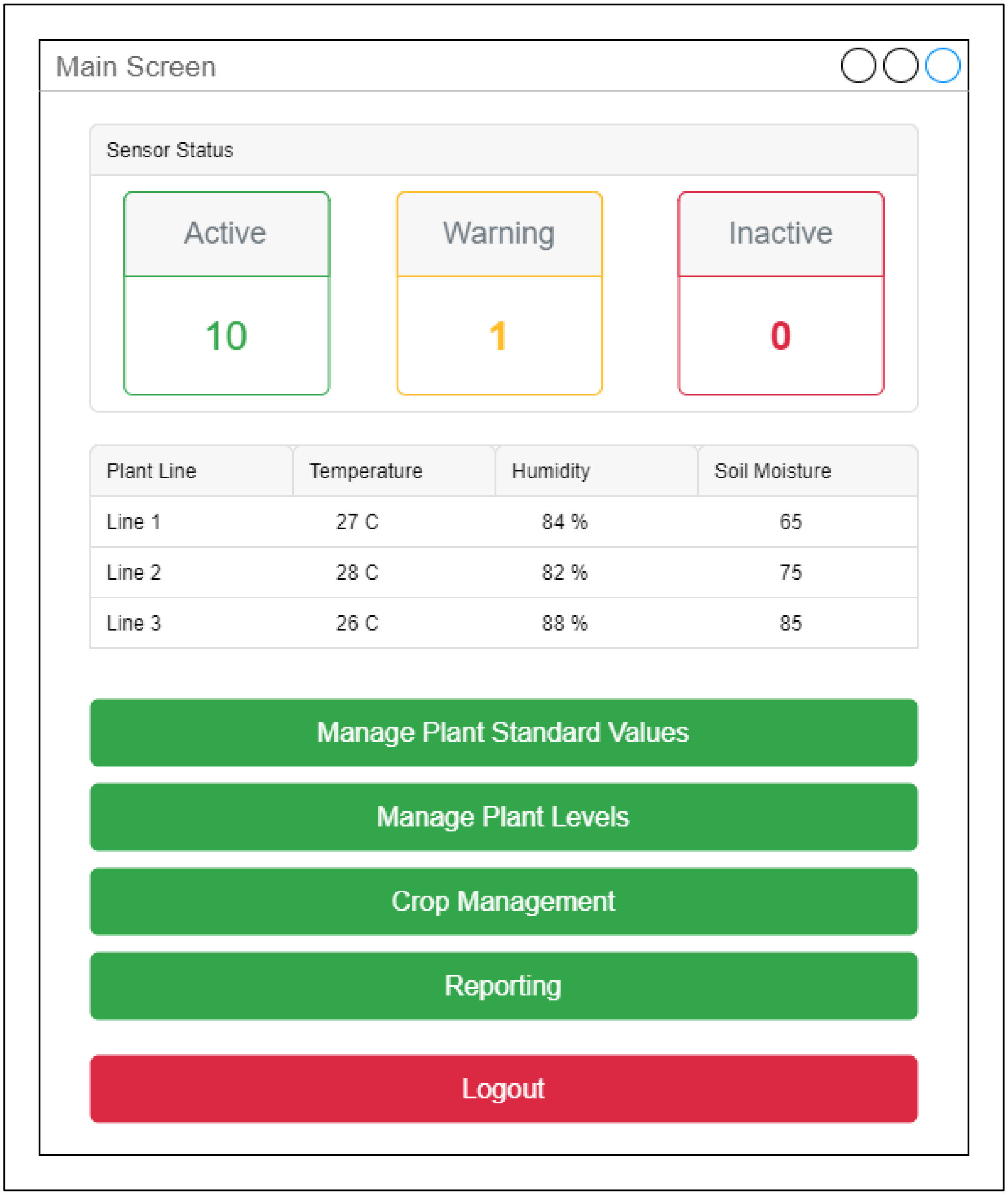


Figure 3.11 Main Screen: Manager

- Main Screen – Planner



Main Screen

Sensor Status

Active 10

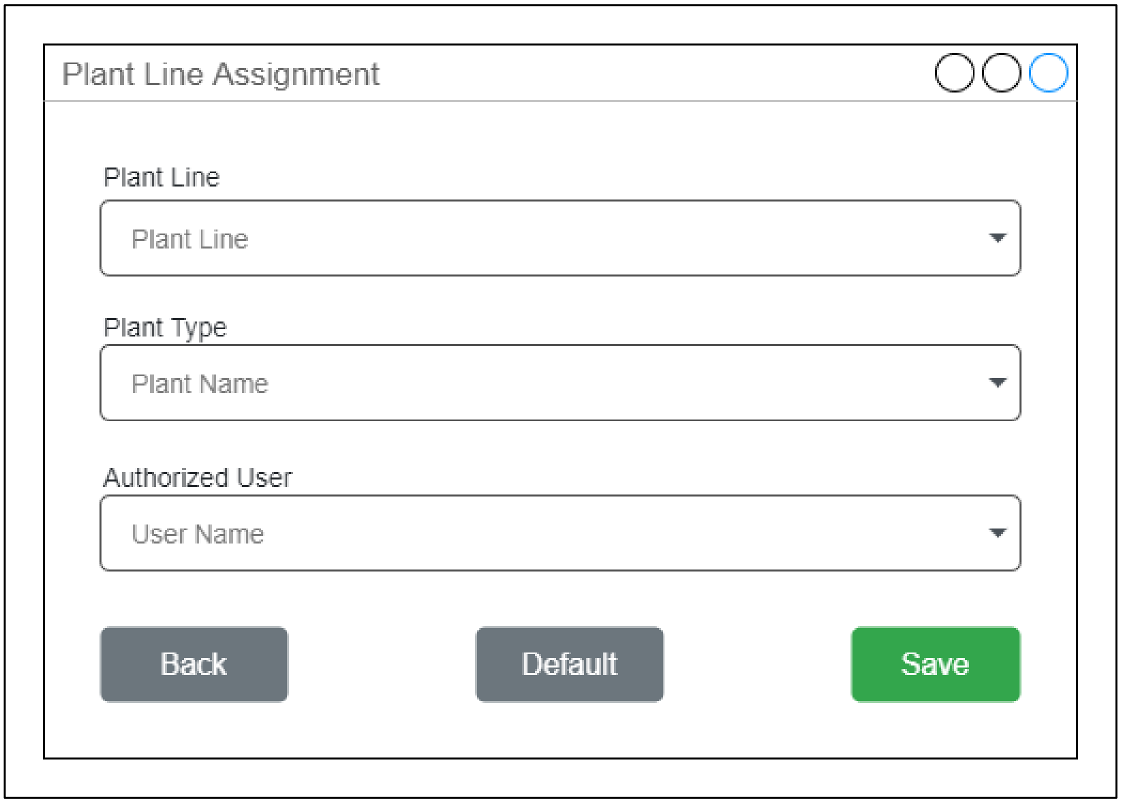
Warning 1

Inactive 0

Plant Line	Temperature	Humidity	Soil Moisture
Line 1	27 C	84 %	65
Line 2	28 C	82 %	75
Line 3	26 C	88 %	85

Figure 3.12 Main Screen Planner

- Plant Line Assignment Screen



Plant Line Assignment

Plant Line: Plant Line

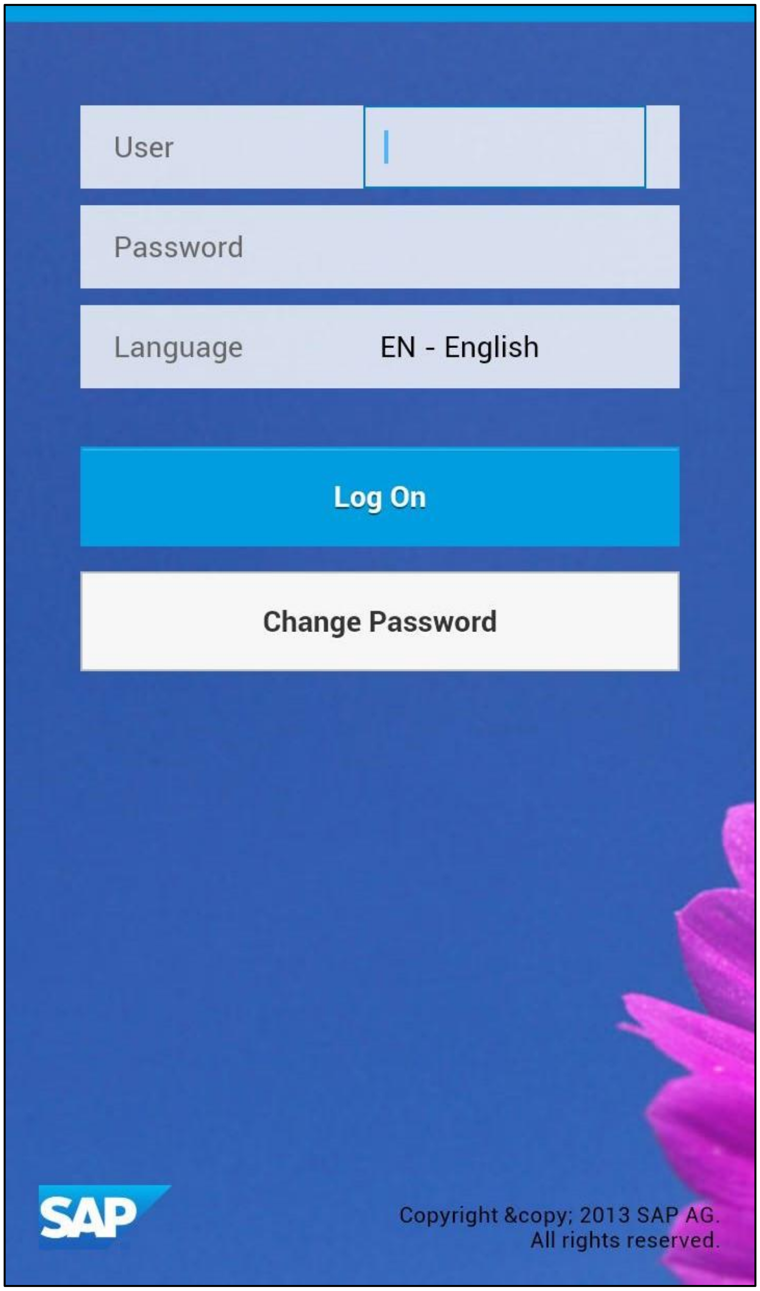
Plant Type: Plant Name

Authorized User: User Name

Figure 3.13 Plant Line Assignment Screen

Fiori UI5 Application

- Login Screen



User

Password

Language EN - English

Copyright &copy; 2013 SAP AG.
All rights reserved.

Figure 3.14: Fiori Application Login

- Main Screen – Mobile Phone

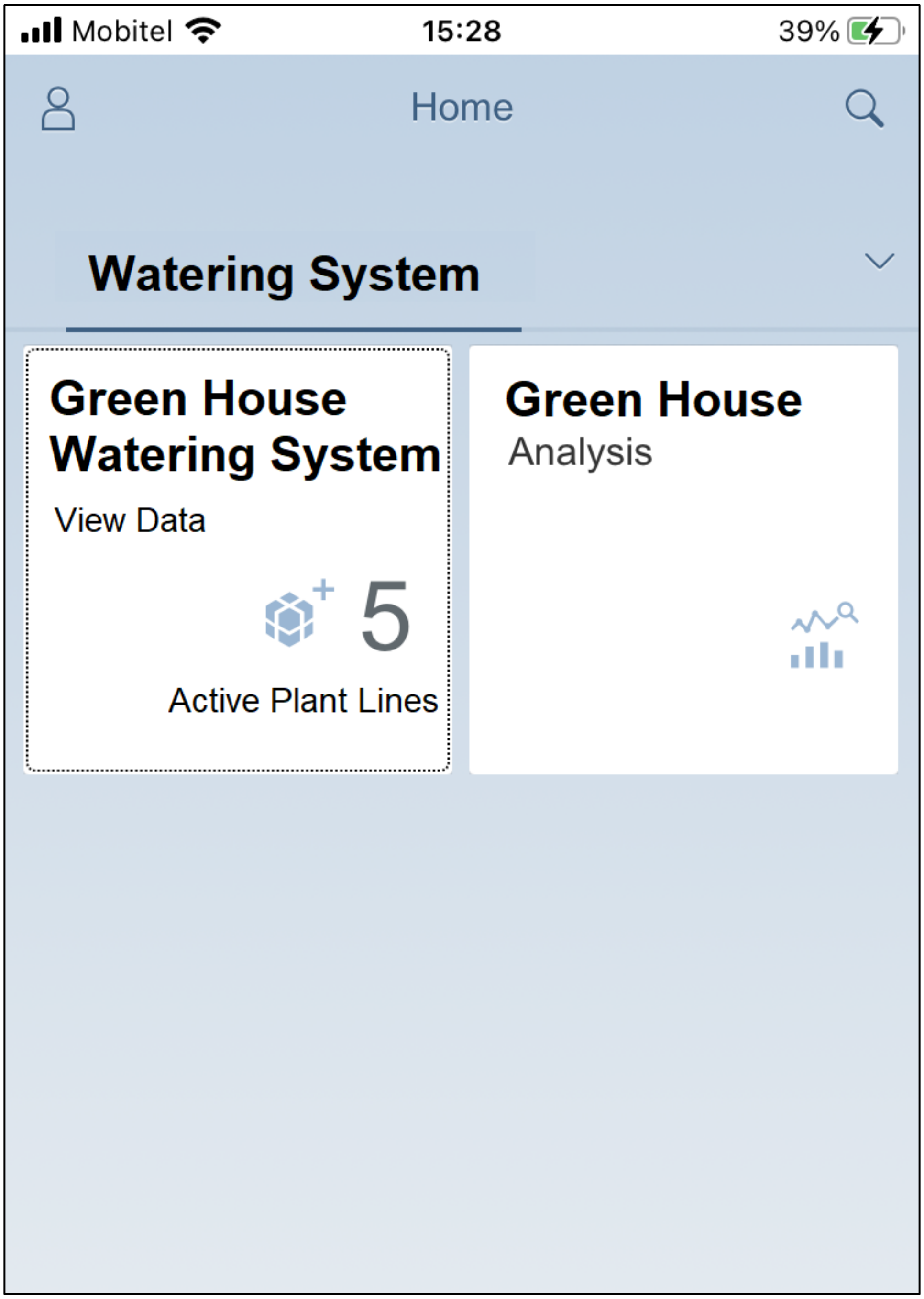


Figure 3.15 Fiori Application Home Screen

- Application Home Screen – Mobile Phone

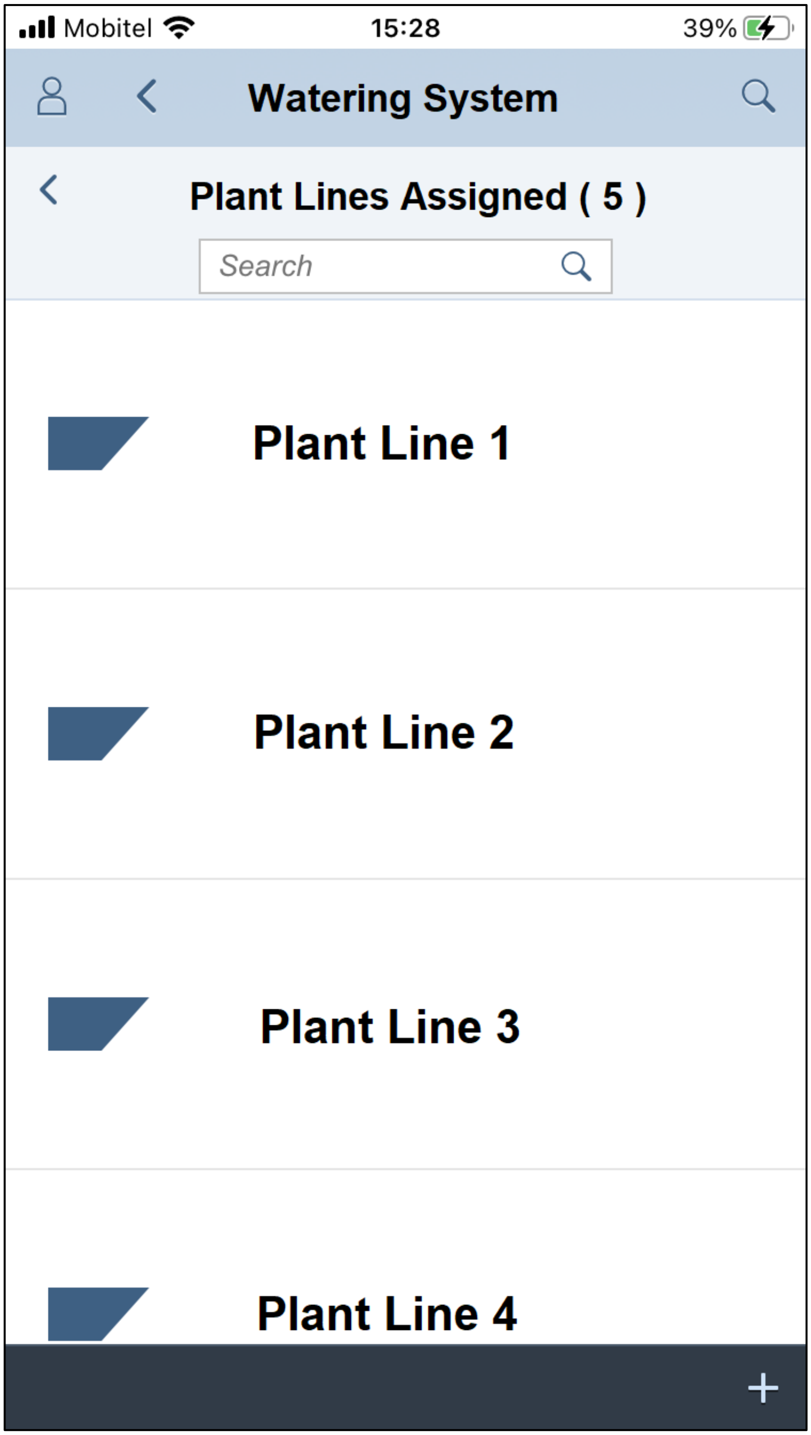
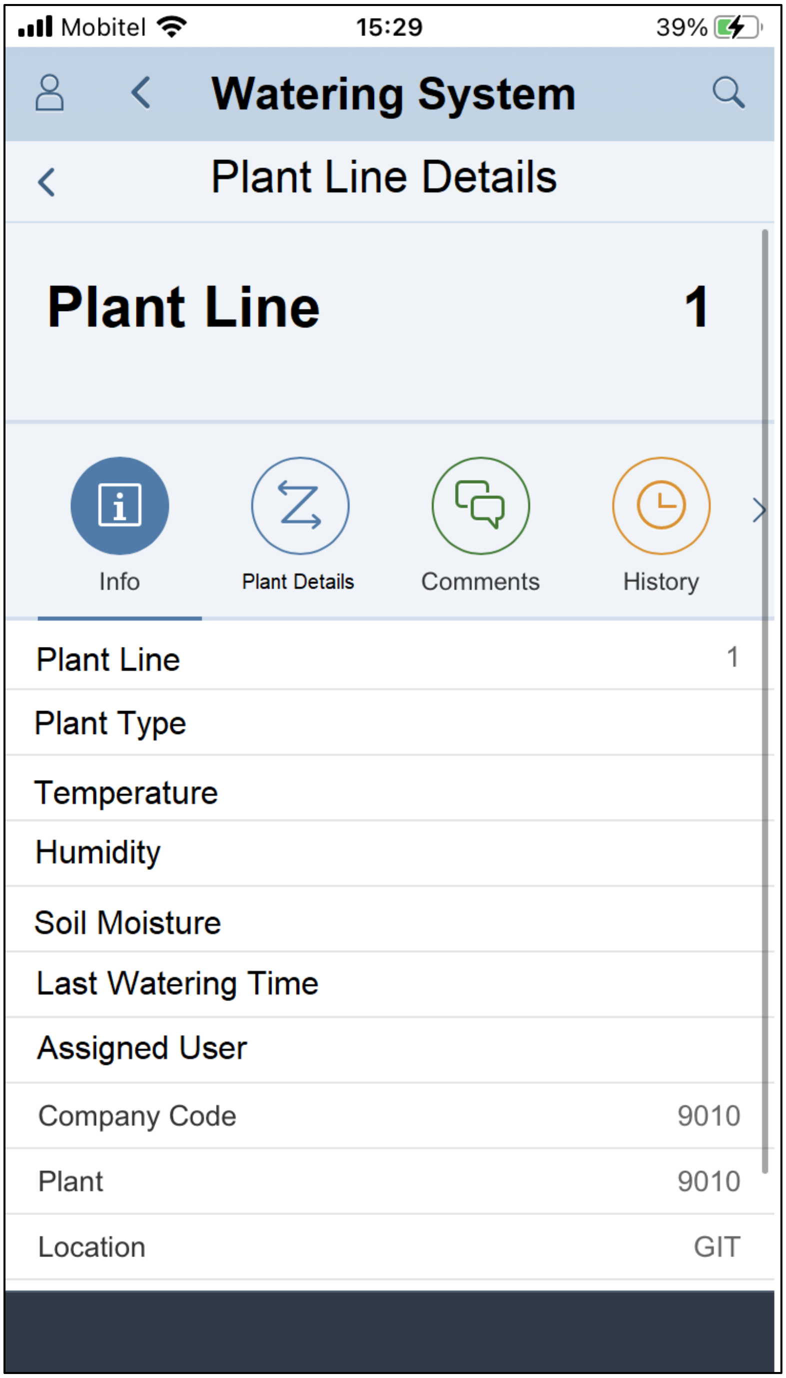


Figure 3.16 Fiori Application Main Screen

- Application Detail Screen – Mobile Phone



Watering System

Plant Line Details

Plant Line **1**

Plant Line	1
Plant Type	
Temperature	
Humidity	
Soil Moisture	
Last Watering Time	
Assigned User	
Company Code	9010
Plant	9010
Location	GIT

Figure 3.17 Fiori Application Detail View

Chapter 4 Implementation

4.1.Implementation Details

This section describes the system's implementation. It includes details about the various steps involved in the development of the system and the advantages of the selection of the environment. The software development process section describes which methodology has been used to develop this system. The code explanation section explains how the implement the code in various environments. It also shows how to implement the modules in various frameworks, hardware infrastructure, development methodologies, tools, database and application development architecture.

4.2.Software Development Process

Agile software engineering is the method to develop this software which is focused on continuous improvement. The development guidelines are designed to create a simple and effective software development process to motivate and be committed to delivering quality software. To accelerate the adaptation of an Agile Method, it often takes customer feedback. An effective way of getting feedback is through an operational prototype.

4.3.Implementation Environment

Hardware information

System Manufacturer: Apple Inc.

System Model: Mac Book Pro

Processor: Intel® Core™ i5-7360U CPU @ 2.30GHz (4 CPUs)

Memory: 8192MB RAM

SSD: 256GB

Display Devices: Intel® Iris® Plus Graphics 640

Sound Devices: Cirrus Logic CS8409

Sound Capture Devices: Cirrus Logic CS8409

Video Capture Devices: Facetime HD Camara

Software information

Operating System: Windows 10 Pro 64-bit (Build 19042)

DirectX Version: DirectX 12

Visual Studio: Microsoft Visual Studio Community 2019 (16.9.4)

SAP S/4 Hana: Fiori 2 (IU5)

Arduino: Arduino IDE (1.8.49)

4.4.Code Explanations

The system is currently under development in the windows environment. Using Visual Studio Community version for developing the standalone application with integrating to Hayleys PLC main ERP SAP S/4 Hana. The system is developed using the WPF application with the MahaApps Plugin which allows creating more user attractive interfaces. Using Arduino IDE for doing read the sensor values and control Water pumps. Arduino uses C language to write the code and from the serial monitor, we can read the return values that need to pass to the standalone system.

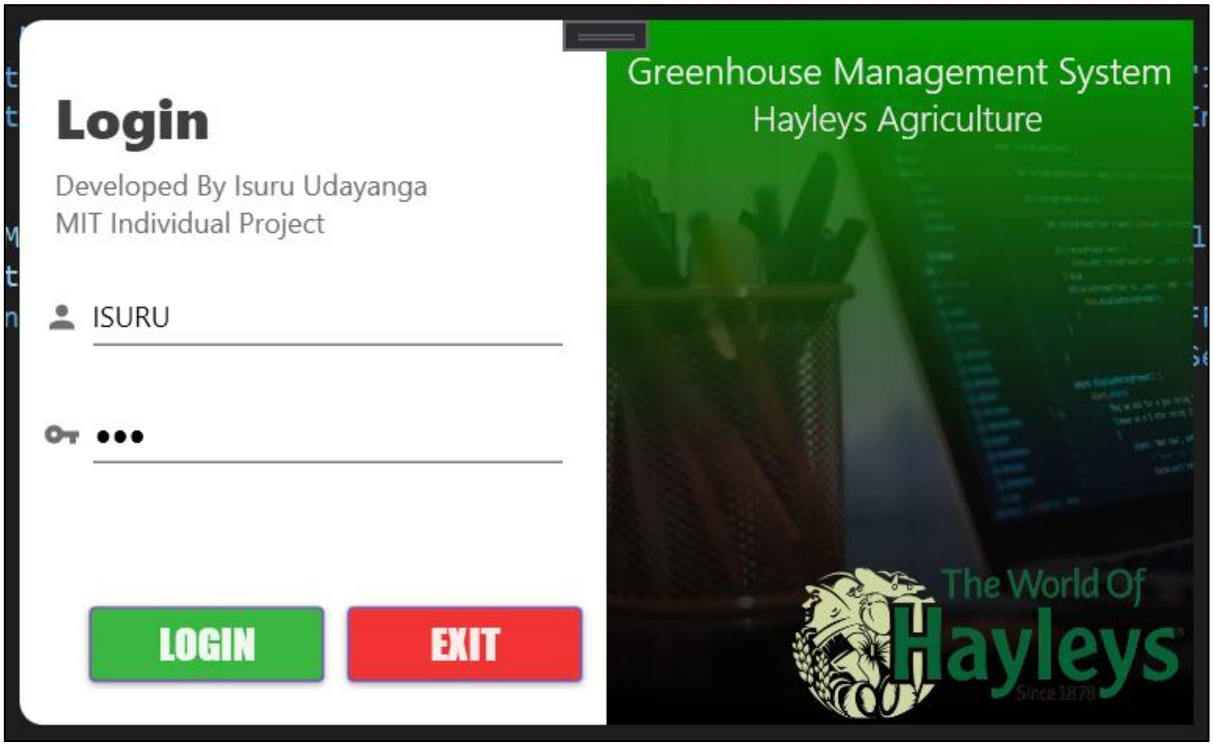
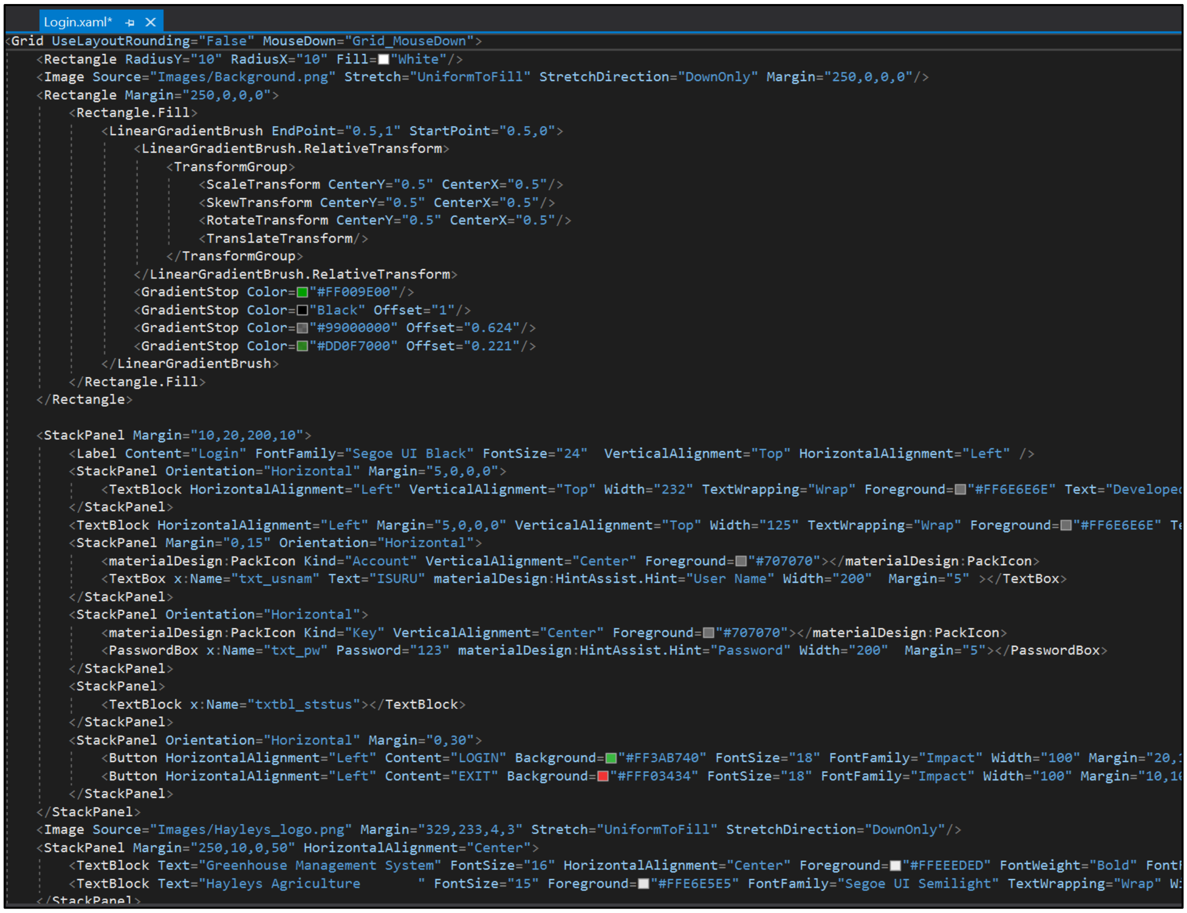


Figure 4.1 Login Interface

Log in Interface is the first interface which shows in figure 5-1 of this application and it is used to login to the system using user ID and Password. All the user names and passwords are stored in the SAP system to make this application more secure and real-time access to all data from anywhere. Login interface creates a connection between this application and the SAP system through an SAP Connector. In this interface system will read data from the SAP system regarding the user data which has user priority and it will pop up the relevant dashboard for the user. There are two user Priorities in this system, User and Manager.

Figure 5-2 shows the XML code for Login Screen.

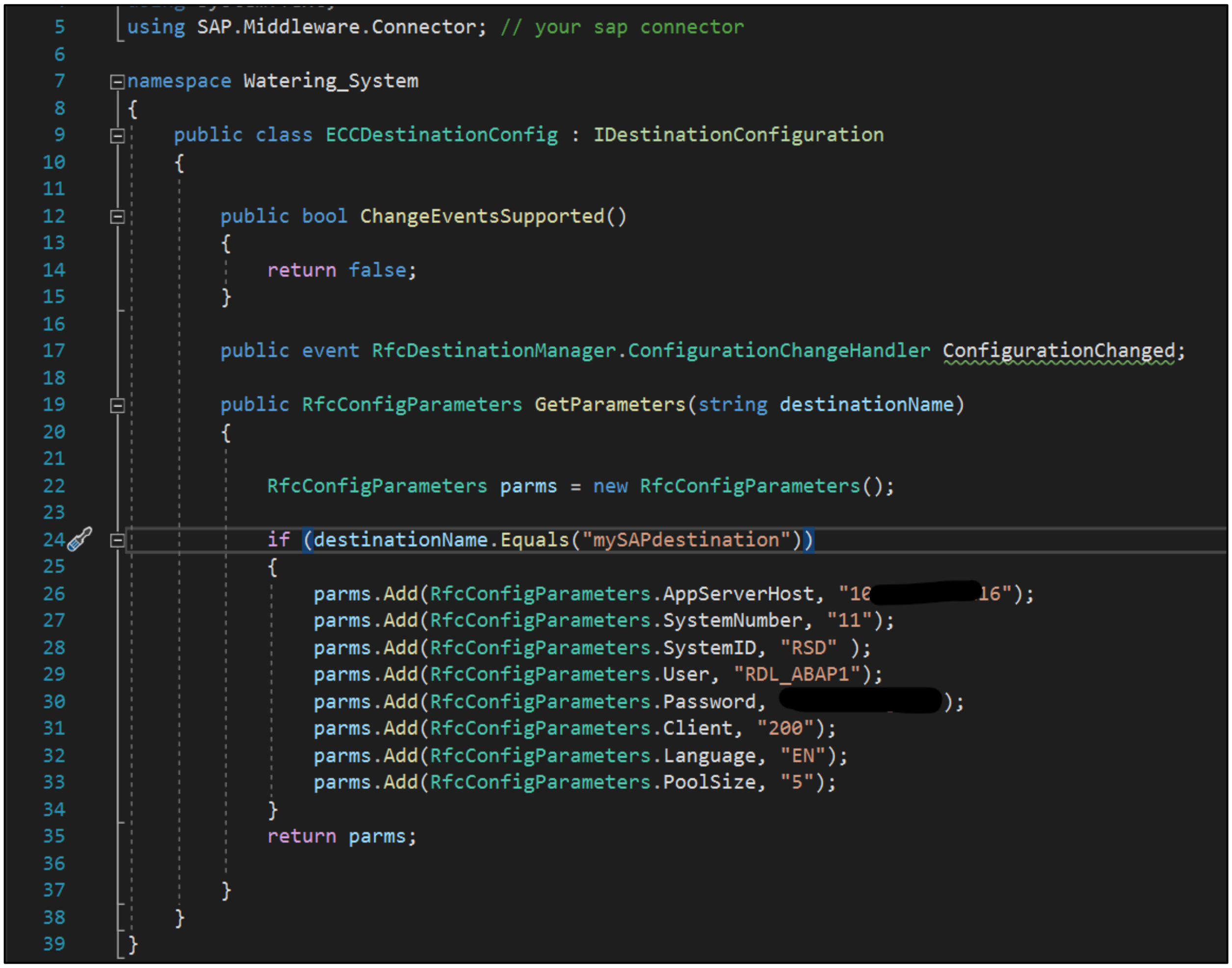


Login.xaml*

```
<Grid UseLayoutRounding="False" MouseDown="Grid_MouseDown">
    <Rectangle RadiusY="10" RadiusX="10" Fill="White"/>
    <Image Source="Images/Background.png" Stretch="UniformToFill" StretchDirection="DownOnly" Margin="250,0,0,0"/>
    <Rectangle Margin="250,0,0,0">
        <Rectangle.Fill>
            <LinearGradientBrush EndPoint="0.5,1" StartPoint="0.5,0">
                <LinearGradientBrush.RelativeTransform>
                    <TransformGroup>
                        <ScaleTransform CenterY="0.5" CenterX="0.5"/>
                        <SkewTransform CenterY="0.5" CenterX="0.5"/>
                        <RotateTransform CenterY="0.5" CenterX="0.5"/>
                        <TranslateTransform/>
                    </TransformGroup>
                </LinearGradientBrush.RelativeTransform>
                <GradientStop Color="#FF009E00"/>
                <GradientStop Color="Black" Offset="1"/>
                <GradientStop Color="#99000000" Offset="0.624"/>
                <GradientStop Color="#DD0F7000" Offset="0.221"/>
            </LinearGradientBrush>
        </Rectangle.Fill>
    </Rectangle>

    <StackPanel Margin="10,20,200,10">
        <Label Content="Login" FontFamily="Segoe UI Black" FontSize="24"  VerticalAlignment="Top" HorizontalAlignment="Left" />
        <StackPanel Orientation="Horizontal" Margin="5,0,0,0">
            <TextBlock HorizontalAlignment="Left" VerticalAlignment="Top" Width="232" TextWrapping="Wrap" Foreground="#FF6E6E6E" Text="Develope
        </StackPanel>
        <TextBlock HorizontalAlignment="Left" Margin="5,0,0,0" VerticalAlignment="Top" Width="125" TextWrapping="Wrap" Foreground="#FF6E6E6E" Te
        <StackPanel Margin="0,15" Orientation="Horizontal">
            <materialDesign:PackIcon Kind="Account" VerticalAlignment="Center" Foreground="#707070"></materialDesign:PackIcon>
            <TextBox x:Name="txt_usnam" Text="ISURU" materialDesign:HintAssist.Hint="User Name" Width="200"  Margin="5" ></TextBox>
        </StackPanel>
        <StackPanel Orientation="Horizontal">
            <materialDesign:PackIcon Kind="Key" VerticalAlignment="Center" Foreground="#707070"></materialDesign:PackIcon>
            <PasswordBox x:Name="txt_pw" Password="123" materialDesign:HintAssist.Hint="Password" Width="200"  Margin="5"></PasswordBox>
        </StackPanel>
        <StackPanel>
            <TextBlock x:Name="txtbl_ststus"></TextBlock>
        </StackPanel>
        <StackPanel Orientation="Horizontal" Margin="0,30">
            <Button HorizontalAlignment="Left" Content="LOGIN" Background="#FF3AB740" FontSize="18" FontFamily="Impact" Width="100" Margin="20,
            <Button HorizontalAlignment="Left" Content="EXIT" Background="#FFF03434" FontSize="18" FontFamily="Impact" Width="100" Margin="10,1
        </StackPanel>
    </StackPanel>
    <Image Source="Images/Hayleys_logo.png" Margin="329,233,4,3" Stretch="UniformToFill" StretchDirection="DownOnly"/>
    <StackPanel Margin="250,10,0,50" HorizontalAlignment="Center">
        <TextBlock Text="Greenhouse Management System" FontSize="16" HorizontalAlignment="Center" Foreground="#FFEEEDED" FontWeight="Bold" Font
        <TextBlock Text="Hayleys Agriculture       " FontSize="15" Foreground="#FFE6E5E5" FontFamily="Segoe UI Semilight" TextWrapping="Wrap" W
    </StackPanel>
```

Figure 4.2 Login Screen IU Design Code

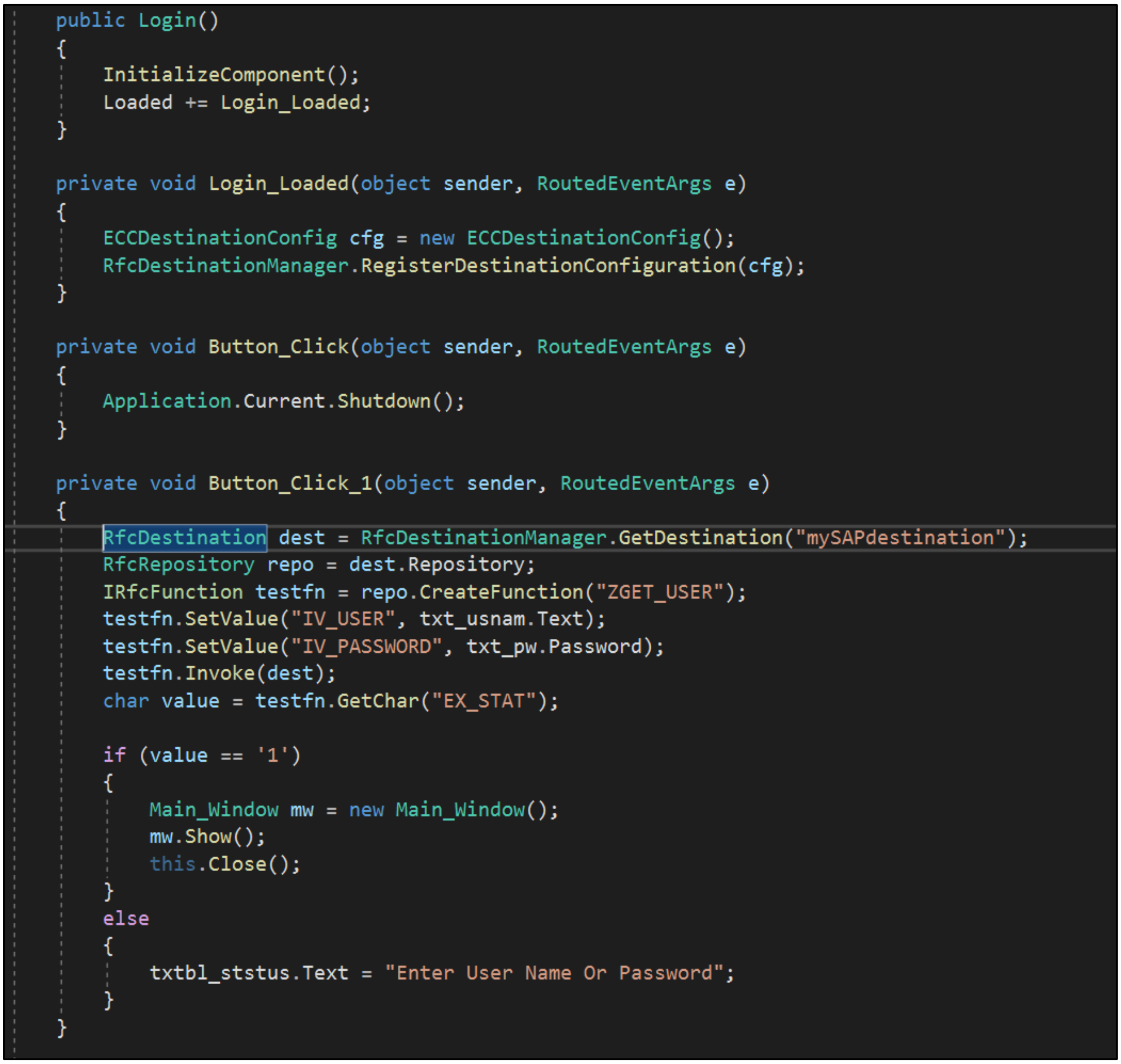
Figure 5-3 shows the code for the SAP connector which needs to connect with the SAP System.



```
using SAP.Middleware.Connector; // your sap connector

namespace Watering_System
{
    public class ECCDestinationConfig : IDestinationConfiguration
    {

        public bool ChangeEventsSupported()
        {
            return false;
        }

        public event RfcDestinationManager.ConfigurationChangeHandler ConfigurationChanged;

        public RfcConfigParameters GetParameters(string destinationName)
        {

            RfcConfigParameters parms = new RfcConfigParameters();

            if (destinationName.Equals("mySAPdestination"))
            {
                parms.Add(RfcConfigParameters.AppServerHost, "1[redacted]16");
                parms.Add(RfcConfigParameters.SystemNumber, "11");
                parms.Add(RfcConfigParameters.SystemID, "RSD" );
                parms.Add(RfcConfigParameters.User, "RDL_ABAP1");
                parms.Add(RfcConfigParameters.Password, [redacted]);
                parms.Add(RfcConfigParameters.Client, "200");
                parms.Add(RfcConfigParameters.Language, "EN");
                parms.Add(RfcConfigParameters.PoolSize, "5");
            }
            return parms;

        }
    }
}
```

Figure 4.3 SAP Connector Class

Figure 5-4 shows the code of establishing the connection between SAP and .Net application.



```
public Login()
{
    InitializeComponent();
    Loaded += Login_Loaded;
}

private void Login_Loaded(object sender, RoutedEventArgs e)
{
    ECCDestinationConfig cfg = new ECCDestinationConfig();
    RfcDestinationManager.RegisterDestinationConfiguration(cfg);
}

private void Button_Click(object sender, RoutedEventArgs e)
{
    Application.Current.Shutdown();
}

private void Button_Click_1(object sender, RoutedEventArgs e)
{
    RfcDestination dest = RfcDestinationManager.GetDestination("mySAPdestination");
    RfcRepository repo = dest.Repository;
    IRfcFunction testfn = repo.CreateFunction("ZGET_USER");
    testfn.SetValue("IV_USER", txt_usnam.Text);
    testfn.SetValue("IV_PASSWORD", txt_pw.Password);
    testfn.Invoke(dest);
    char value = testfn.GetChar("EX_STAT");

    if (value == '1')
    {
        Main_Window mw = new Main_Window();
        mw.Show();
        this.Close();
    }
    else
    {
        txtbl_ststus.Text = "Enter User Name Or Password";
    }
}
```

Figure 4.4 Establishing the SAP Connection

Admin Dashboard

The dashboard is the main screen of this application which has all the functionalities and controlling the system. Figure 5-5 interface shows users an overall idea about the system.

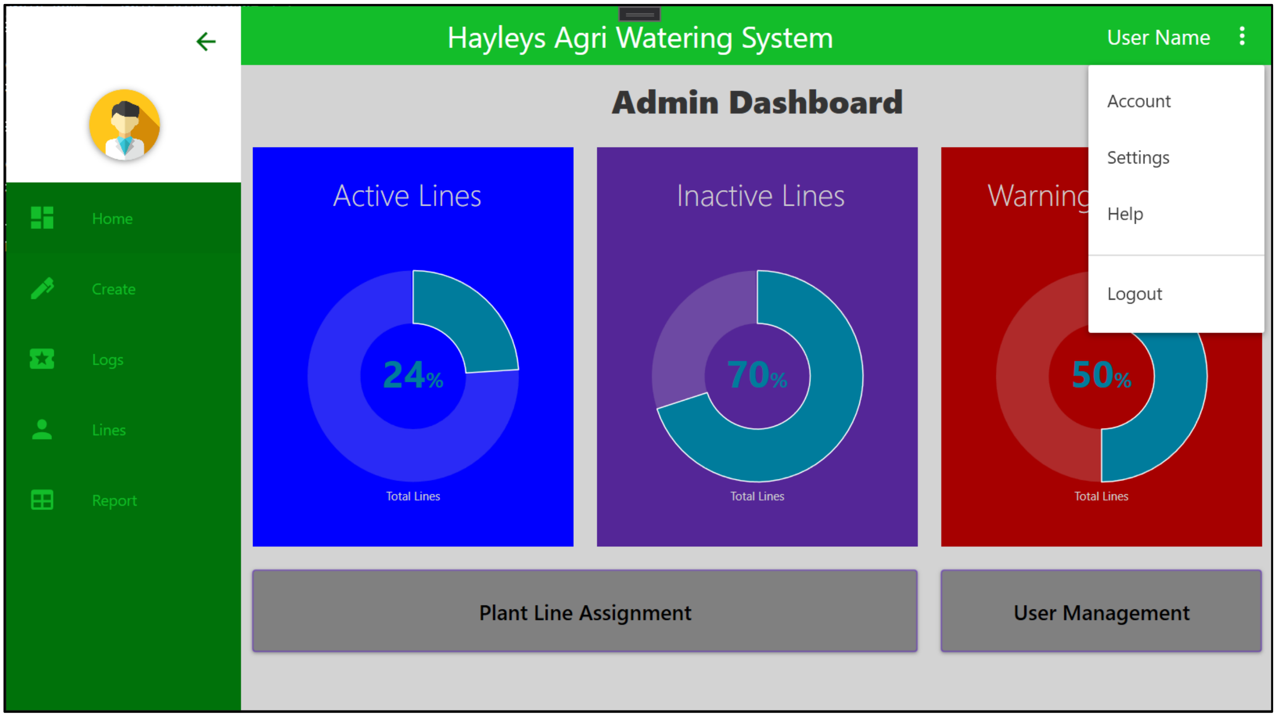
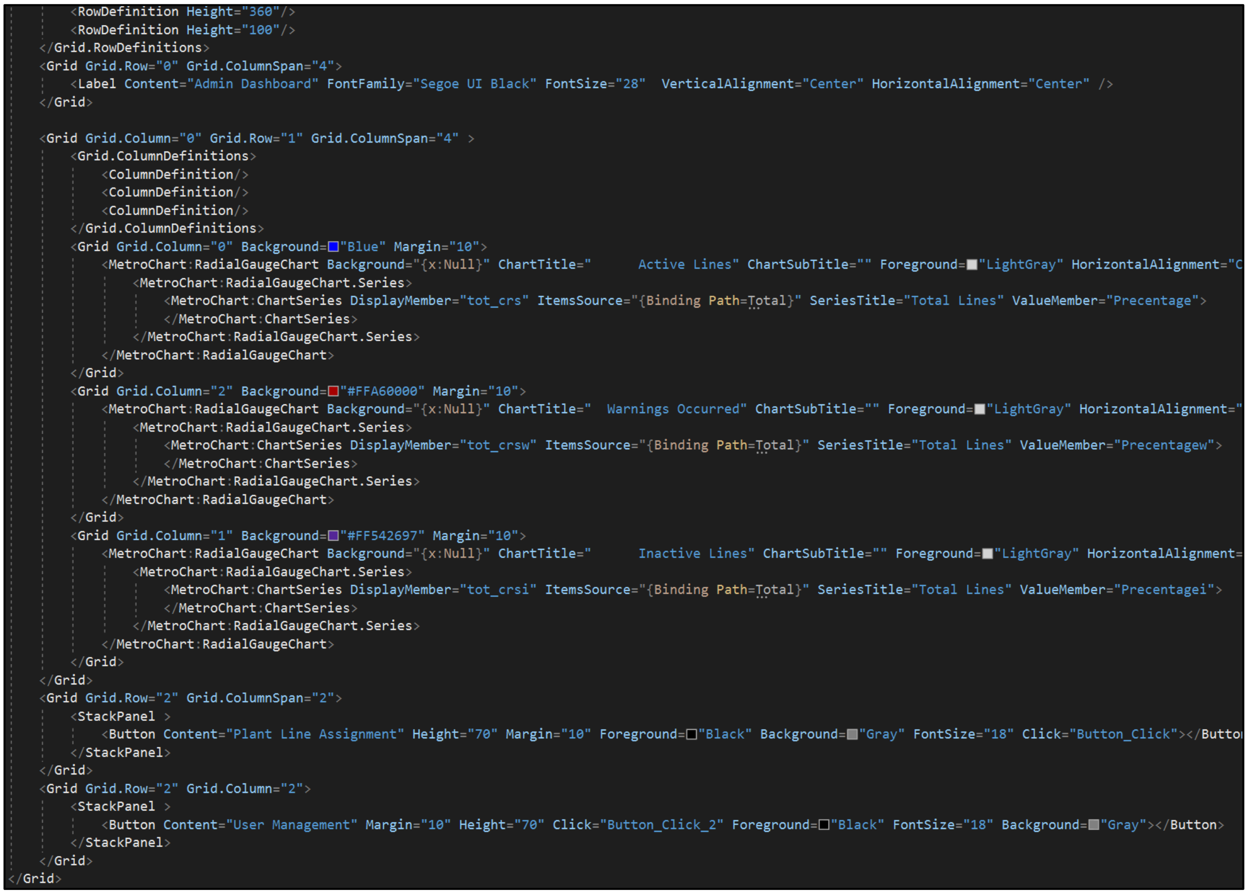


Figure 4.5 Manager Dashboard

Figure 5-6 shows the XML code for the dashboard.

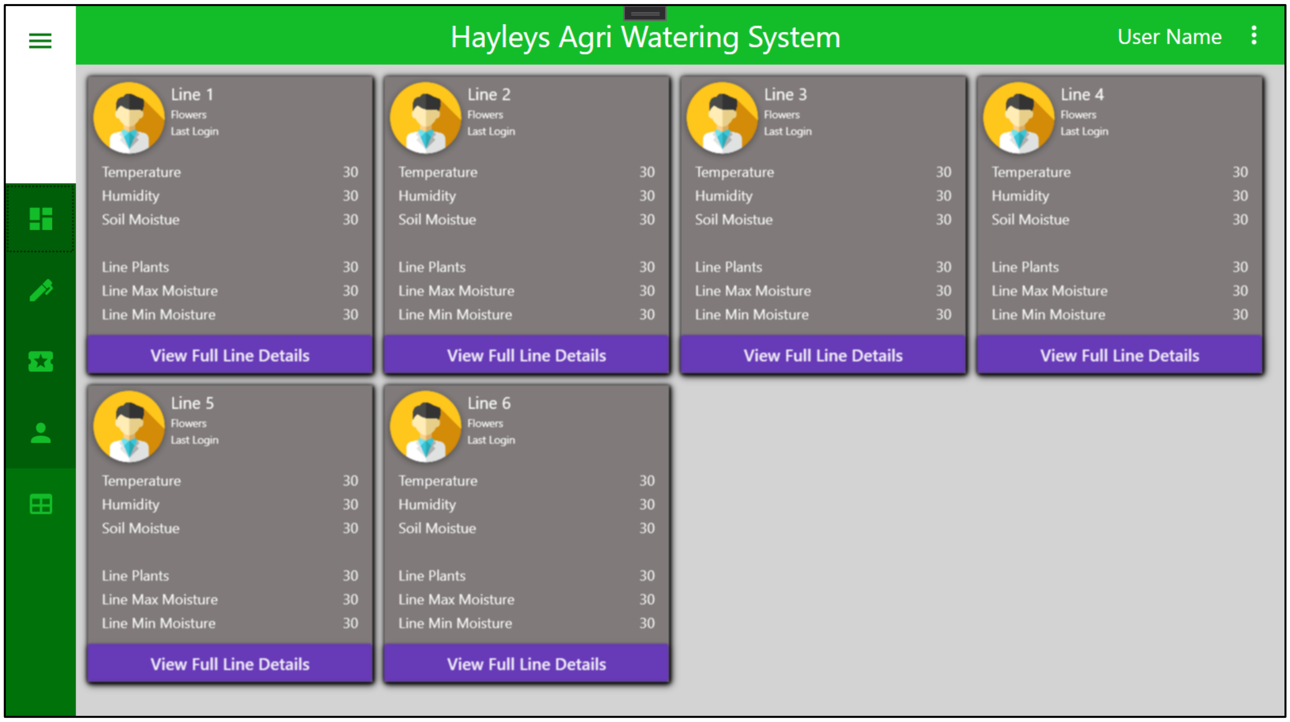


```
        <RowDefinition Height="360"/>
        <RowDefinition Height="100"/>
    </Grid.RowDefinitions>
    <Grid Grid.Row="0" Grid.ColumnSpan="4">
        <Label Content="Admin Dashboard" FontFamily="Segoe UI Black" FontSize="28"  VerticalAlignment="Center" HorizontalAlignment="Center" />
    </Grid>

    <Grid Grid.Column="0" Grid.Row="1" Grid.ColumnSpan="4" >
        <Grid.ColumnDefinitions>
            <ColumnDefinition/>
            <ColumnDefinition/>
            <ColumnDefinition/>
        </Grid.ColumnDefinitions>
        <Grid Grid.Column="0" Background=🟦"Blue" Margin="10">
            <MetroChart:RadialGaugeChart Background="{x:Null}" ChartTitle="      Active Lines" ChartSubTitle="" Foreground=⬜"LightGray" HorizontalAlignment="C
                <MetroChart:RadialGaugeChart.Series>
                    <MetroChart:ChartSeries DisplayMember="tot_crs" ItemsSource="{Binding Path=Total}" SeriesTitle="Total Lines" ValueMember="Precentage">
                    </MetroChart:ChartSeries>
                </MetroChart:RadialGaugeChart.Series>
            </MetroChart:RadialGaugeChart>
        </Grid>
        <Grid Grid.Column="2" Background=🟥"#FFA60000" Margin="10">
            <MetroChart:RadialGaugeChart Background="{x:Null}" ChartTitle="  Warnings Occurred" ChartSubTitle="" Foreground=⬜"LightGray" HorizontalAlignment="
                <MetroChart:RadialGaugeChart.Series>
                    <MetroChart:ChartSeries DisplayMember="tot_crsw" ItemsSource="{Binding Path=Total}" SeriesTitle="Total Lines" ValueMember="Precentagew">
                    </MetroChart:ChartSeries>
                </MetroChart:RadialGaugeChart.Series>
            </MetroChart:RadialGaugeChart>
        </Grid>
        <Grid Grid.Column="1" Background=🟪"#FF542697" Margin="10">
            <MetroChart:RadialGaugeChart Background="{x:Null}" ChartTitle="      Inactive Lines" ChartSubTitle="" Foreground=⬜"LightGray" HorizontalAlignment=
                <MetroChart:RadialGaugeChart.Series>
                    <MetroChart:ChartSeries DisplayMember="tot_crsi" ItemsSource="{Binding Path=Total}" SeriesTitle="Total Lines" ValueMember="Precentagei">
                    </MetroChart:ChartSeries>
                </MetroChart:RadialGaugeChart.Series>
            </MetroChart:RadialGaugeChart>
        </Grid>
    </Grid>
    <Grid Grid.Row="2" Grid.ColumnSpan="2">
        <StackPanel >
            <Button Content="Plant Line Assignment" Height="70" Margin="10" Foreground=⬛"Black" Background=◼"Gray" FontSize="18" Click="Button_Click"></Butto
        </StackPanel>
    </Grid>
    <Grid Grid.Row="2" Grid.Column="2">
        <StackPanel >
            <Button Content="User Management" Margin="10" Height="70" Click="Button_Click_2" Foreground=⬛"Black" FontSize="18" Background=◼"Gray"></Button>
        </StackPanel>
    </Grid>
</Grid>
```

Figure 4.6 Dashboard UI Code

Plant Line Detail View

Figure 5-7 interface shows users summary details about the Plant Lines.



Line 1
Flowers
Last Login

Temperature	30
Humidity	30
Soil Moistue	30
Line Plants	30
Line Max Moisture	30
Line Min Moisture	30

Line 2
Flowers
Last Login

Temperature	30
Humidity	30
Soil Moistue	30
Line Plants	30
Line Max Moisture	30
Line Min Moisture	30

Line 3
Flowers
Last Login

Temperature	30
Humidity	30
Soil Moistue	30
Line Plants	30
Line Max Moisture	30
Line Min Moisture	30

Line 4
Flowers
Last Login

Temperature	30
Humidity	30
Soil Moistue	30
Line Plants	30
Line Max Moisture	30
Line Min Moisture	30

Line 5
Flowers
Last Login

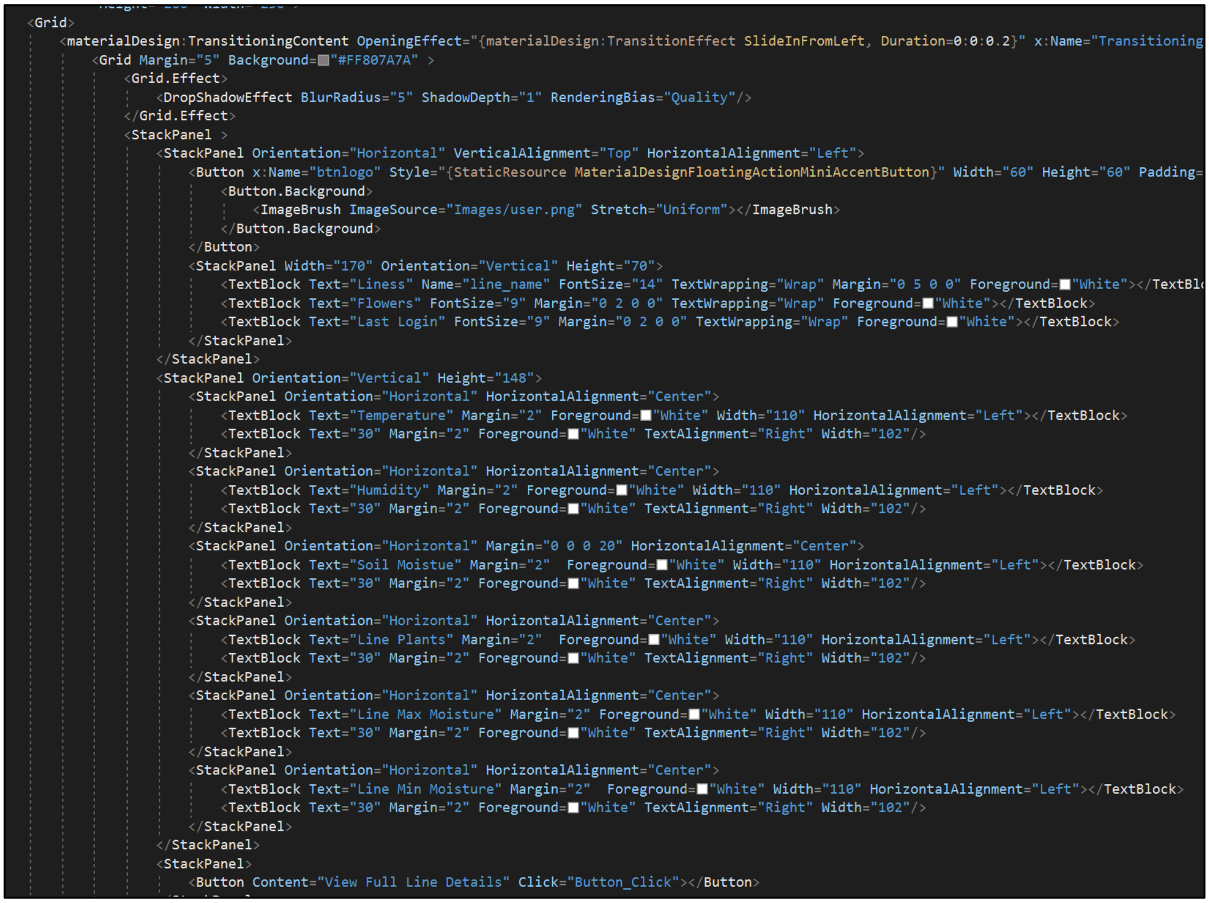
Temperature	30
Humidity	30
Soil Moistue	30
Line Plants	30
Line Max Moisture	30
Line Min Moisture	30

Line 6
Flowers
Last Login

Temperature	30
Humidity	30
Soil Moistue	30
Line Plants	30
Line Max Moisture	30
Line Min Moisture	30

Figure 4.7 Line Summary Details

Figure 5-8 shows the XML code for Line summery detail



```
<Grid>
    <materialDesign:TransitioningContent OpeningEffect="{materialDesign:TransitionEffect SlideInFromLeft, Duration=0:0:0.2}" x:Name="Transitioning
        <Grid Margin="5" Background=■"#FF807A7A" >
            <Grid.Effect>
                <DropShadowEffect BlurRadius="5" ShadowDepth="1" RenderingBias="Quality"/>
            </Grid.Effect>
            <StackPanel >
                <StackPanel Orientation="Horizontal" VerticalAlignment="Top" HorizontalAlignment="Left">
                    <Button x:Name="btnlogo" Style="{StaticResource MaterialDesignFloatingActionMiniAccentButton}" Width="60" Height="60" Padding=
                        <Button.Background>
                            <ImageBrush ImageSource="Images/user.png" Stretch="Uniform"></ImageBrush>
                        </Button.Background>
                    </Button>
                    <StackPanel Width="170" Orientation="Vertical" Height="70">
                        <TextBlock Text="Liness" Name="line_name" FontSize="14" TextWrapping="Wrap" Margin="0 5 0 0" Foreground=□"White"></TextBl
                        <TextBlock Text="Flowers" FontSize="9" Margin="0 2 0 0" TextWrapping="Wrap" Foreground=□"White"></TextBlock>
                        <TextBlock Text="Last Login" FontSize="9" Margin="0 2 0 0" TextWrapping="Wrap" Foreground=□"White"></TextBlock>
                    </StackPanel>
                </StackPanel>
                <StackPanel Orientation="Vertical" Height="148">
                    <StackPanel Orientation="Horizontal" HorizontalAlignment="Center">
                        <TextBlock Text="Temperature" Margin="2" Foreground=□"White" Width="110" HorizontalAlignment="Left"></TextBlock>
                        <TextBlock Text="30" Margin="2" Foreground=□"White" TextAlignment="Right" Width="102"/>
                    </StackPanel>
                    <StackPanel Orientation="Horizontal" HorizontalAlignment="Center">
                        <TextBlock Text="Humidity" Margin="2" Foreground=□"White" Width="110" HorizontalAlignment="Left"></TextBlock>
                        <TextBlock Text="30" Margin="2" Foreground=□"White" TextAlignment="Right" Width="102"/>
                    </StackPanel>
                    <StackPanel Orientation="Horizontal" Margin="0 0 0 20" HorizontalAlignment="Center">
                        <TextBlock Text="Soil Moistue" Margin="2"  Foreground=□"White" Width="110" HorizontalAlignment="Left"></TextBlock>
                        <TextBlock Text="30" Margin="2" Foreground=□"White" TextAlignment="Right" Width="102"/>
                    </StackPanel>
                    <StackPanel Orientation="Horizontal" HorizontalAlignment="Center">
                        <TextBlock Text="Line Plants" Margin="2"  Foreground=□"White" Width="110" HorizontalAlignment="Left"></TextBlock>
                        <TextBlock Text="30" Margin="2" Foreground=□"White" TextAlignment="Right" Width="102"/>
                    </StackPanel>
                    <StackPanel Orientation="Horizontal" HorizontalAlignment="Center">
                        <TextBlock Text="Line Max Moisture" Margin="2" Foreground=□"White" Width="110" HorizontalAlignment="Left"></TextBlock>
                        <TextBlock Text="30" Margin="2" Foreground=□"White" TextAlignment="Right" Width="102"/>
                    </StackPanel>
                    <StackPanel Orientation="Horizontal" HorizontalAlignment="Center">
                        <TextBlock Text="Line Min Moisture" Margin="2"  Foreground=□"White" Width="110" HorizontalAlignment="Left"></TextBlock>
                        <TextBlock Text="30" Margin="2" Foreground=□"White" TextAlignment="Right" Width="102"/>
                    </StackPanel>
                </StackPanel>
                <StackPanel>
                    <Button Content="View Full Line Details" Click="Button_Click"></Button>
```

Figure 4.8 Line Summary UI Code

Chapter 5 Testing and Evaluation

5.1.Introduction

Software testing takes a major role in Software quality assurance that provides how the accuracy of the data provides from the software and the analysis of the proposed system development. Software testing is based on specification, design and code evaluation of the software product to deliver a bug-free solution to the customer. Most of the testing and the different software components are done by using two basic methods validations and verifications which needs to capture accurate data to the system and provide accurate data from the system. The basic milestone of the system testing and evaluation is to decide whether the system functions and the user's requirements are met by the developed system.

5.2.Testing

Testing starts with implementation and code testing begins as unit testing to facilitate testing for complex scenarios. The capabilities, validations and verifications needed were designed before system implementation. A test plan is a detailed document that sets out the test strategy, test targets, software testing resources, test estimates and test results.

Test cases were designed according to the designed test plan at the implementation stage. Divide main test modules into small parts to make testing easy and reduce the complexity of testing for the user which is highly recommended for testing and provides better testing accuracy of the system. Testing small parts of the main modules helps the development team to do the relevant changes to the software. The following 4-1 to 4-6 tables specify some test cases.

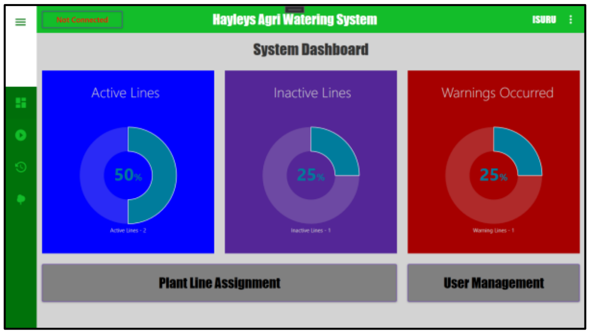
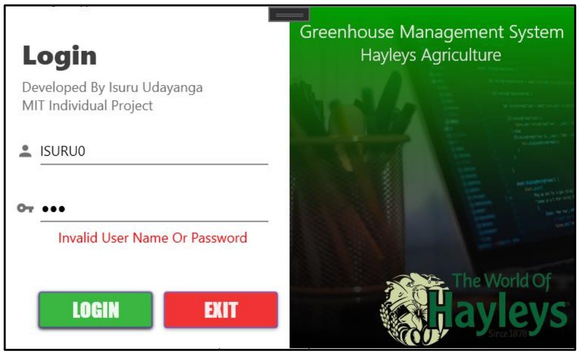
Test Case ID: 1

Test Case Title: Login to Dashboard

Module Name: Login

Test Case Priority: High

Expected Result: Login to the application from the login window

Step ID	Test Step	Test Data	Actual Result	Status
1	Open the application Enter valid login details and click Login	ISURU 123	Direct to Dashboard  Not Connected Hayleys Agri Watering System ISURU System Dashboard Active Lines 50% Active Lines - 2 Inactive Lines 25% Inactive Lines - 1 Warnings Occurred 25% Warning Lines - 1 Plant Line Assignment User Management	Pass
2	Open the application Enter invalid login details and click the login button	ISURU0 123	Notify Invalid User Name or Password  Login Developed By Isuru Udayanga MIT Individual Project ISURU0 ••• Invalid User Name Or Password LOGIN EXIT Greenhouse Management System Hayleys Agriculture The World Of Hayleys Since 1878	Pass

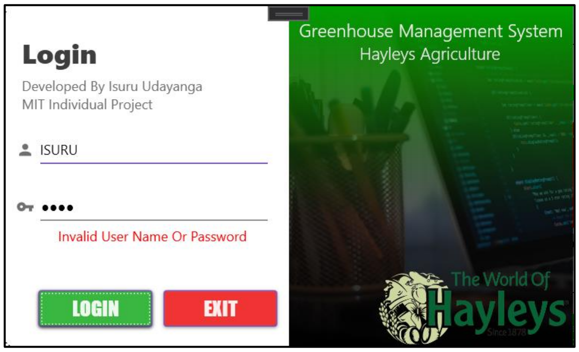
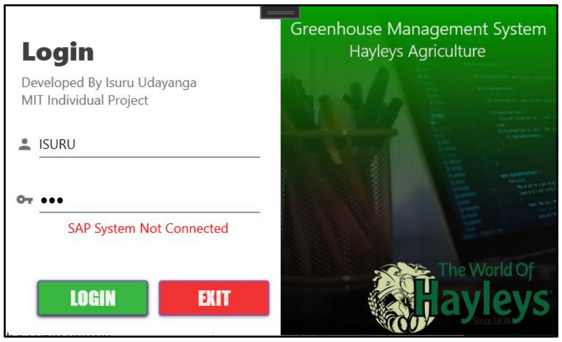
3	Open the application Enter valid username Enter invalid password Click Login button	ISURU 123@	Notify Invalid User Name or Password  Login Developed By Isuru Udayanga MIT Individual Project ISURU •••• Invalid User Name Or Password LOGIN EXIT Greenhouse Management System Hayleys Agriculture The World Of Hayleys Since 1878	Pass
4	Open the application Provide valid username Provide invalid password Disconnect SAP system from PC Click Login button	ISURU 123	Notify SAP System Not Connected  Login Developed By Isuru Udayanga MIT Individual Project ISURU ••• SAP System Not Connected LOGIN EXIT Greenhouse Management System Hayleys Agriculture The World Of Hayleys Since 1878	Pass

Table 5.1 Login to Dashboard

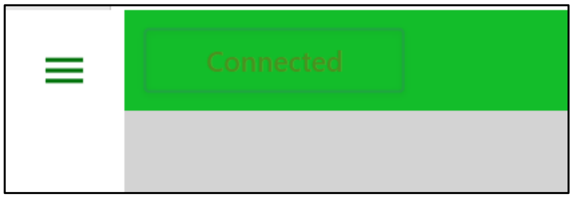
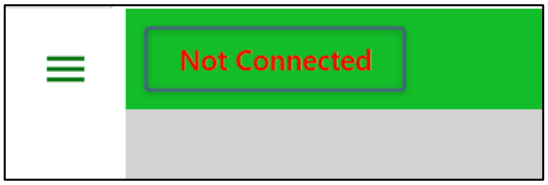
Test Case ID: 2

Test Case Title: Bluetooth Connectivity to Arduino Module

Module Name: Bluetooth

Test Case Priority: High

Expected Result: Automatically Arduino Bluetooth module should connect to Application

Step ID	**Test Step**	**Test Data**	**Actual Result**	**Status**
1	Login to Application Turn on the PC’s Bluetooth Turn on Arduino Module		Application Indicates it is connected to Bluetooth 	Pass
2	Login to Application Turn on the PC’s Bluetooth Turn off Arduino Module		Application Indicates it is not connected to Bluetooth 	Pass
3	Login to Application Turn off PC’s Bluetooth		Application Indicates it is not connected to Bluetooth	Pass

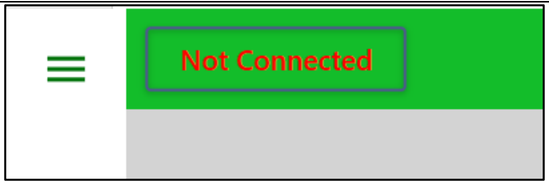
	Turn on Arduino Module			

Table 5.2 Bluetooth Connectivity to Arduino Module

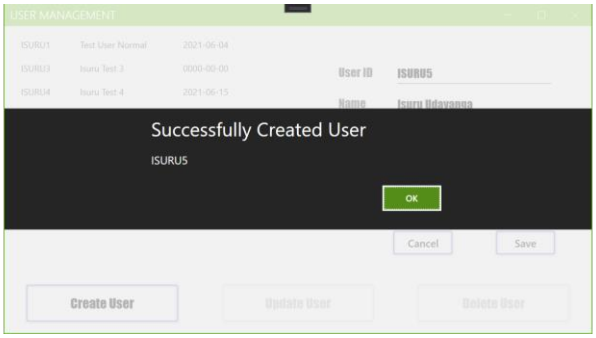
Test Case ID: 3

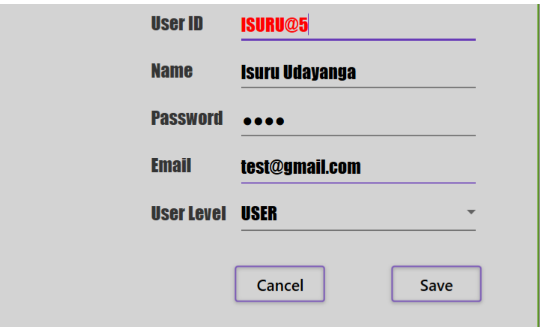
Test Case Title: Create User

Module Name: User Management

Test Case Priority: High

Expected Result: Successfully create a user and highlight the invalid fields when wrong data is entered.

Step ID	Test Step	Test Data	Actual Result	Status
1	Go to User Management Window Click Create User Button Enter a valid User ID Enter Name Enter valid Password Enter valid Email	ISURU5 Isuru Udayanga 1234 test@gmail.com User	Display successful alert message  ISURU1 Test User Normal 2021-06-04 ISURU3 Isuru Test 3 0000-00-00 ISURU4 Isuru Test 4 2021-06-15 User ID ISURU5 Name Isuru Udayanga Successfully Created User ISURU5 USER MANAGEMENT OK Cancel Save Create User Update User Delete User	Pass

	Select User Level Click Save Button			
2	Go to User Management Window Click Create User Enter an invalid User ID Enter Name Enter valid Password Enter valid Email Select User Level Click Save Button	ISURU@5 Isuru Udayanga 1234 test@gmail.com User	Highlight invalid User ID field  User ID ISURU@5 Name Isuru Udayanga Password •••• Email test@gmail.com User Level USER 	Pass
3	Go to User Management Window Click Create User	ISURU5 Isuru Udayanga 1234	Highlight invalid email field	Pass

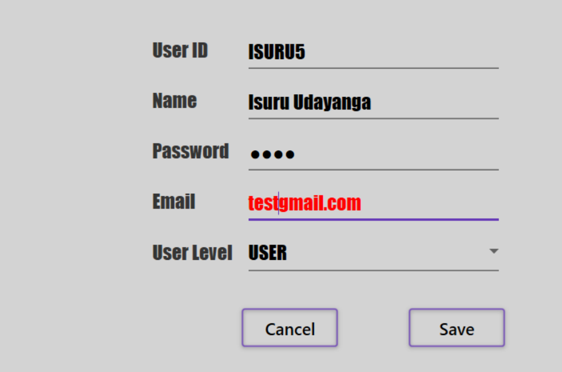
	Enter a valid User ID Enter Name Enter valid Password Enter invalid Email Select User Level Click Save Button	testgmail.com User	 User ID ISURU5 Name Isuru Udayanga Password •••• Email testgmail.com User Level USER 	

Table 5.3 Create Use

Test Case ID: 4

Test Case Title: Update Sensor Values to SAP System

Module Name: Arduino Module Integration

Test Case Priority: High

Pre-conditions: Arduino system and SAP system should be connected to Application

Expected Result: Successfully capture sensor data from Arduino and transfer it to Application via Bluetooth and save values to SAP System. After saving to the DB, the Application log tab should be able to view recorded values.

Step ID	Test Step	Test Data	Actual Result	Status
1	The application		Test Message to Display Values	Pass

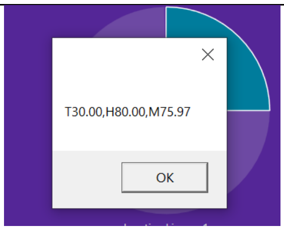
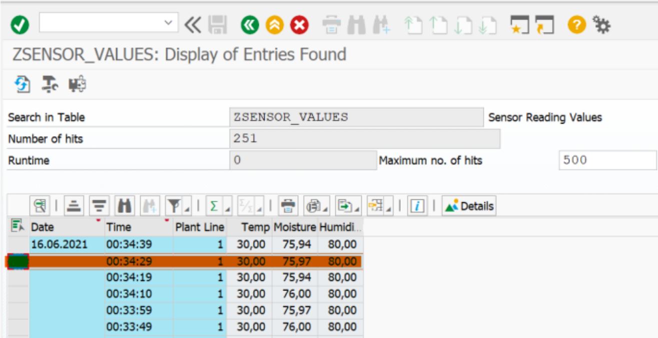
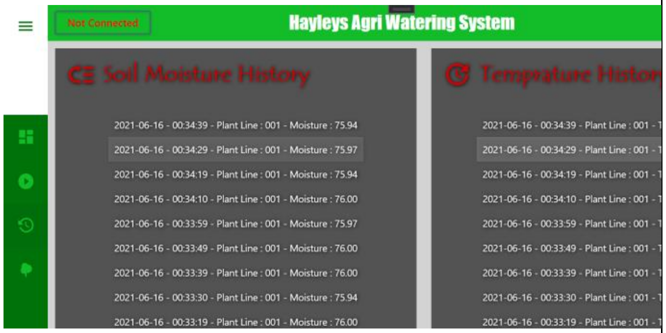
	will trigger a timer every 10 seconds to request Sensor data from Arduino.		 T30.00,H80.00,M75.97 	
2	Send the data to the SAP system which is received from Arduino	T30.0 H80.0 M75.79	Updated the SAP System Table  ZSENSOR_VALUES: Display of Entries Found Search in Table: ZSENSOR_VALUES Sensor Reading Values Number of hits: 251 Runtime: 0 Maximum no. of hits: 500 Date | Time | Plant Line | Temp | Moisture | Humidi... 16.06.2021 | 00:34:39 | 1 | 30,00 | 75,94 | 80,00 | 00:34:29 | 1 | 30,00 | 75,97 | 80,00 | 00:34:19 | 1 | 30,00 | 75,94 | 80,00 | 00:34:10 | 1 | 30,00 | 76,00 | 80,00 | 00:33:59 | 1 | 30,00 | 75,97 | 80,00 | 00:33:49 | 1 | 30,00 | 76,00 | 80,00	Pass
3	View captured data from Application	T30.0 H80.0 M75.79	Updated Application Log tab  Hayleys Agri Watering System Soil Moisture History 2021-06-16 - 00:34:39 - Plant Line : 001 - Moisture : 75.94 2021-06-16 - 00:34:29 - Plant Line : 001 - Moisture : 75.97 2021-06-16 - 00:34:19 - Plant Line : 001 - Moisture : 75.94 2021-06-16 - 00:34:10 - Plant Line : 001 - Moisture : 76.00 2021-06-16 - 00:33:59 - Plant Line : 001 - Moisture : 75.97 2021-06-16 - 00:33:49 - Plant Line : 001 - Moisture : 76.00 2021-06-16 - 00:33:39 - Plant Line : 001 - Moisture : 76.00 2021-06-16 - 00:33:30 - Plant Line : 001 - Moisture : 75.94 2021-06-16 - 00:33:19 - Plant Line : 001 - Moisture : 76.00 Temprature Histor 2021-06-16 - 00:34:39 - Plant Line : 001 - T 2021-06-16 - 00:34:29 - Plant Line : 001 - T 2021-06-16 - 00:34:19 - Plant Line : 001 - T 2021-06-16 - 00:34:10 - Plant Line : 001 - T 2021-06-16 - 00:33:59 - Plant Line : 001 - T 2021-06-16 - 00:33:49 - Plant Line : 001 - T 2021-06-16 - 00:33:39 - Plant Line : 001 - T 2021-06-16 - 00:33:30 - Plant Line : 001 - T 2021-06-16 - 00:33:19 - Plant Line : 001 - T	Pass

Table 5.4 Update Sensor Values to SAP System

Please refer to Appendix – B for rest of the test cases

5.3.Evaluation of the System

Evaluation is done by comparing to the current manual system and this system gives more information about the plants to analyze how the plants growth changes against the environmental condition with accurate data from the system. Compared to manual watering, this system can maintain the soil moisture of Green House Plants lines to the same level and change a certain amount when required. However, the results of the system are more accurate and they have never had this much detailed report in the past. All the feedbacks give a good impression on the system.

The overall feedback from Google Form of the proposed solution given by each user was collected and the following graphical charts from figure 4-8 to 4-17 shows the summary of feedback.

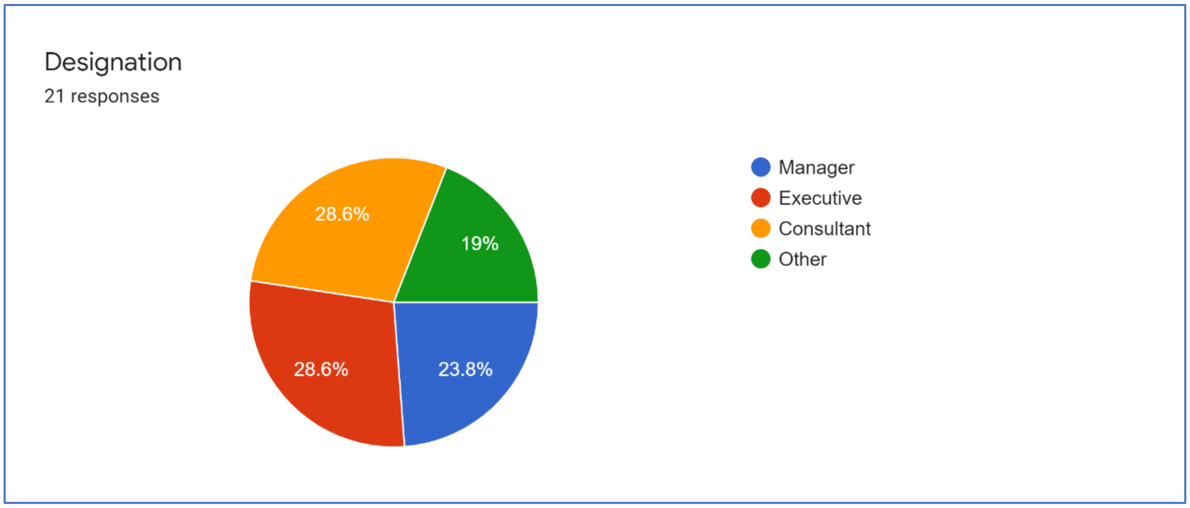


Figure 5.1 Designation Feedback Chart

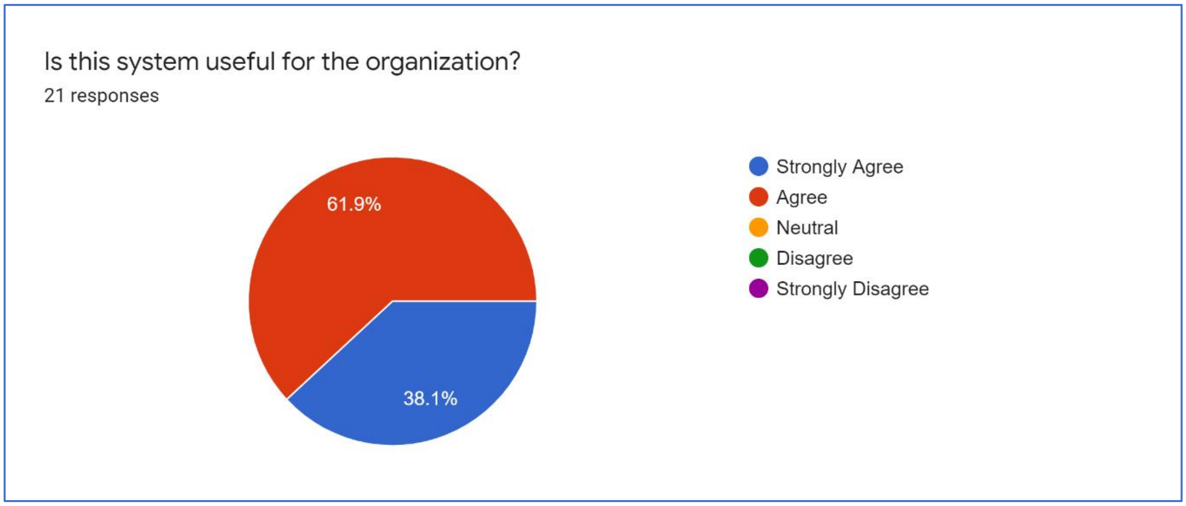


Figure 5.2 Question 1 Feedback Chart

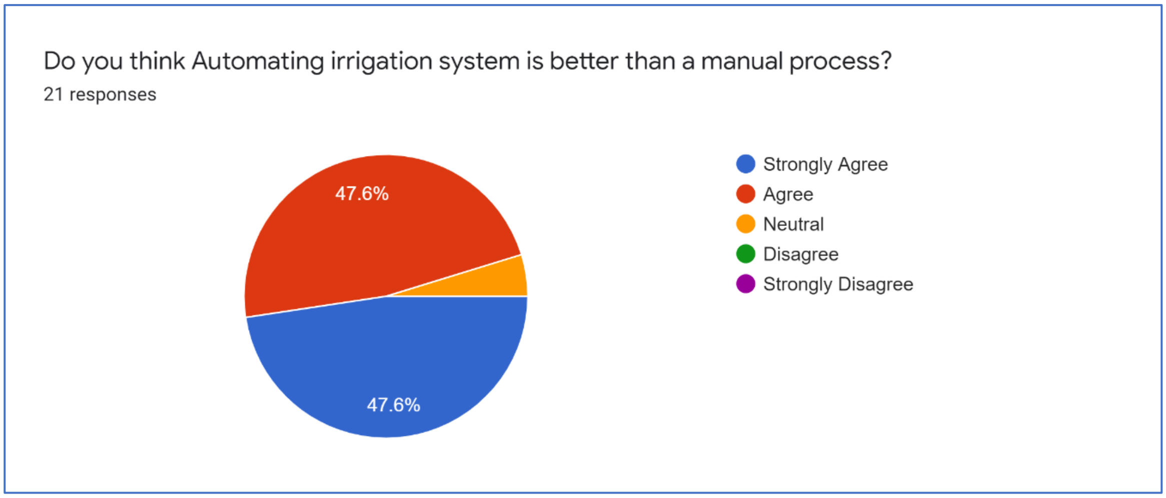
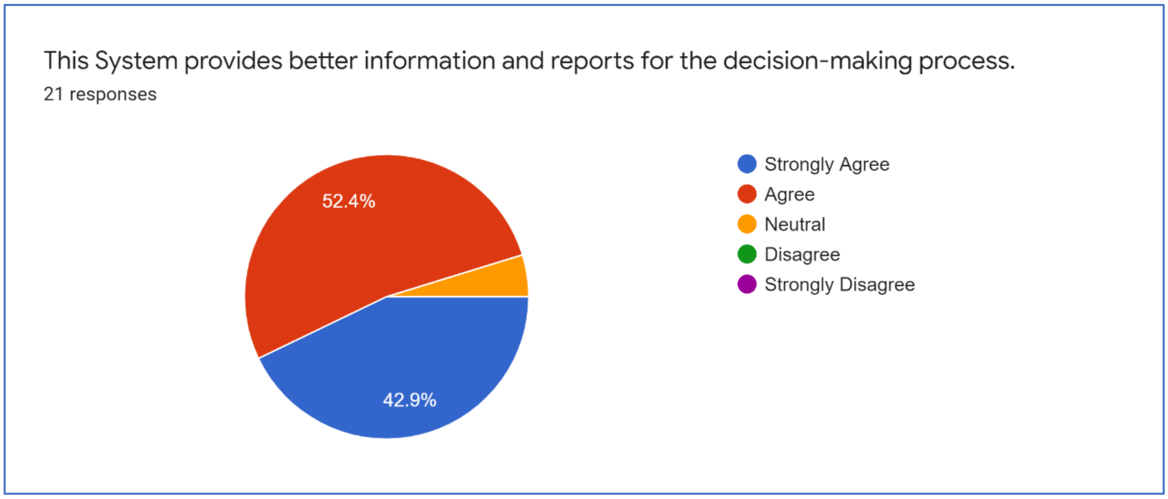


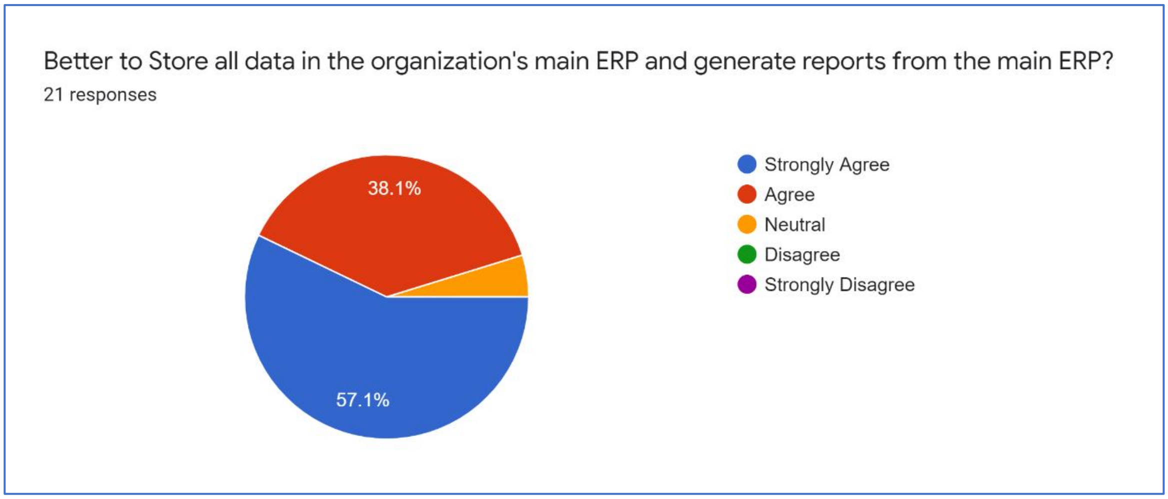
Figure 5.3 Question 2 Feedback Chart



This System provides better information and reports for the decision-making process.

21 responses

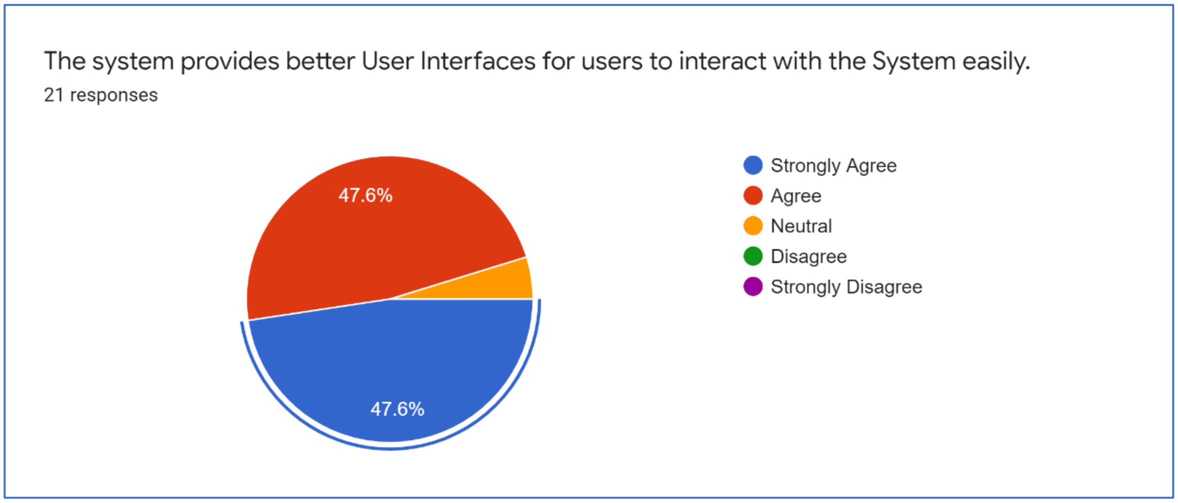
Figure 5.4 Question 3 Feedback Chart



Better to Store all data in the organization's main ERP and generate reports from the main ERP?

21 responses

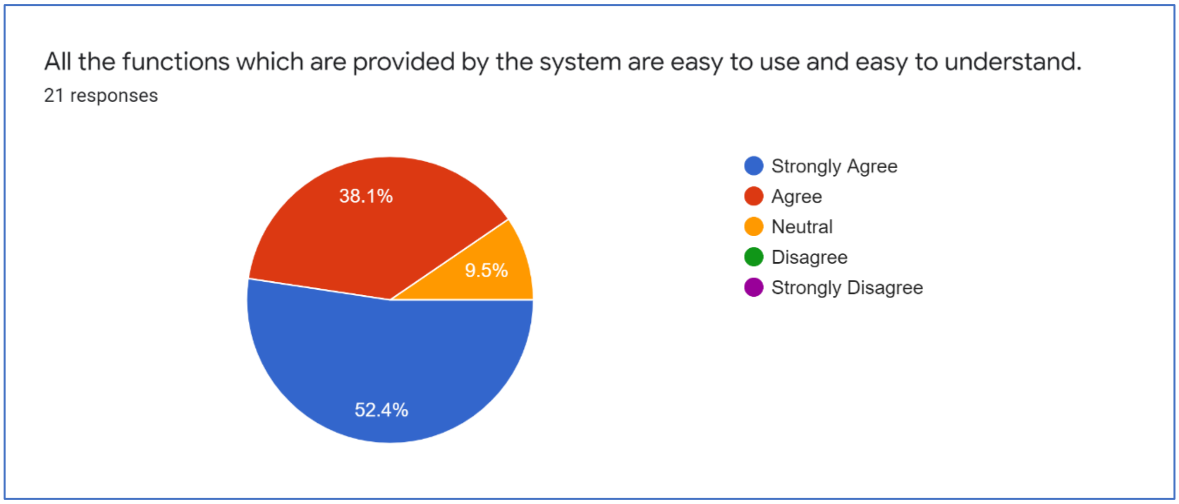
Figure 5.5 Question 4 Feedback Chart



The system provides better User Interfaces for users to interact with the System easily.

21 responses

Figure 5.6 Question 5 Feedback Chart



All the functions which are provided by the system are easy to use and easy to understand.

21 responses

Figure 5.7 Question 6 Feedback Chart

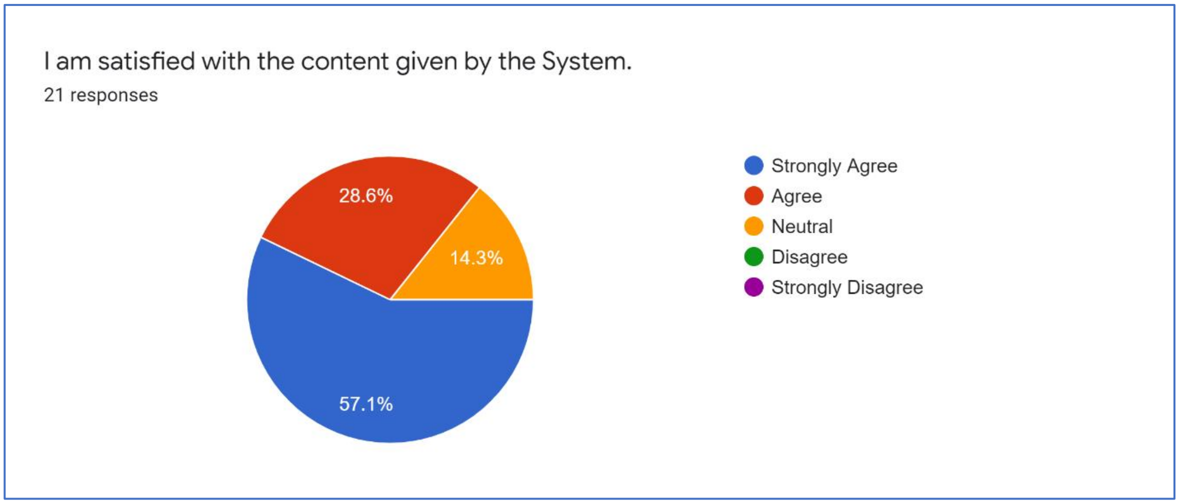


Figure 5.8 Question 7 Feedback Chart

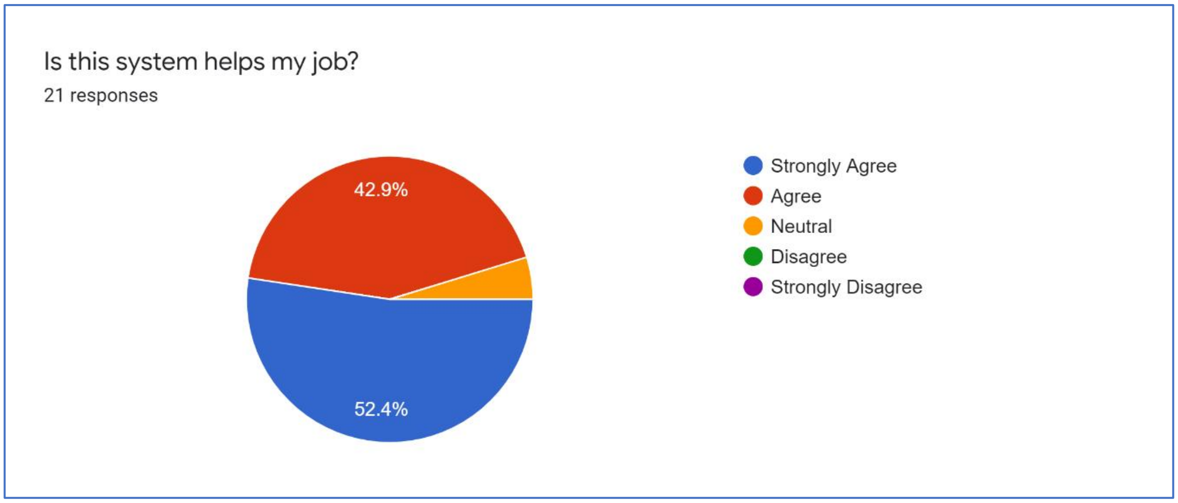


Figure 5.9 Question 8 Feedback Chart

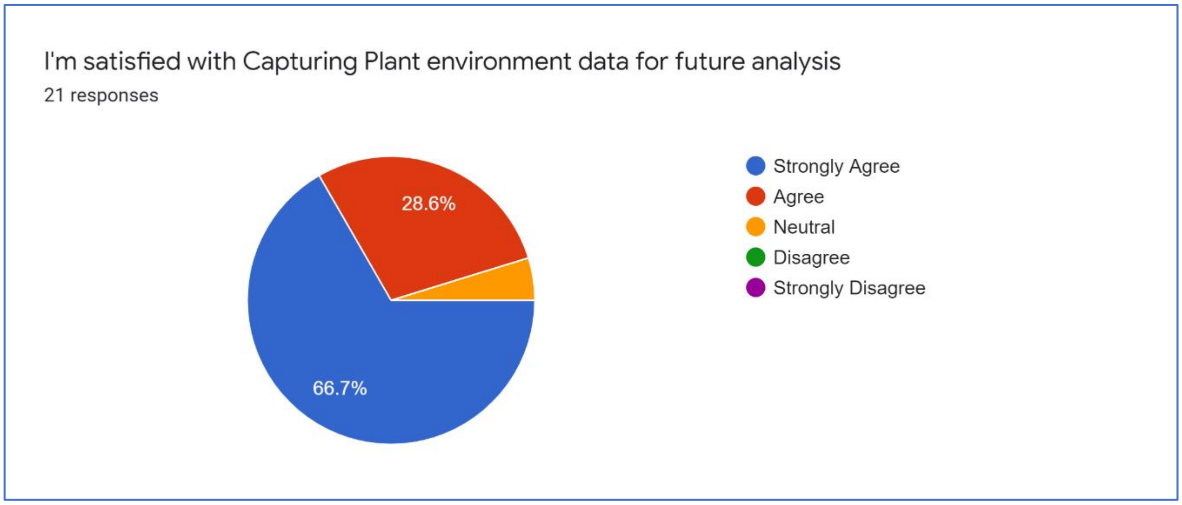


Figure 5.10 Question 9 Feedback Chart

Chapter 6 Conclusion

6.1.Introduction

The objectives and future improvements of the established system are discussed in this chapter. Further changes to improve the functionality of the system

6.2.Conclusion

"Automated Greenhouse Irrigation System" has been developed and successfully tested. The combined functionality of all the hardware components employed has been designed. The system was automatically tested to work. The humidity sensors monitor the humidity of the plants. The moisture sensor transmits the signal to the Arduino module, which triggers the water pump to switch it on and supply the water to the respective plant if the water level is lower than the intended and limit level. The system stops on its own and the water pump is turned OFF when the appropriate moisture level is achieved. The sensor values and water quantities are saved in the main ERP system for analytical and reporting purposes.

6.3.Further Work

The development of the Fiori App has to be delayed due to various limitations. The main functionalities of the system have been developed in the first place with a completed testing cycle. Fiori App development and enhancing report generation has to be developed. Currently, this system is only sensitive to Soil Moisture but needs to develop to control greenhouse Humidity levels from the sensors according to the required levels.

List of References

[1] Visual-paradigm.com. *What is Unified Modeling Language (UML)?*. [online] Available at: https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-uml/ [Accessed 23 February 2021].

[2] En.wikipedia.org. 2021. .NET Framework. [online] Available <https://en.wikipedia.org/wiki/.NET_Framework> [Accessed 2 February 2021].

[3] Arduino.cc. 2021. Arduino - Introduction. [online] Available at: <https://www.arduino.cc/en/guide/introduction> [Accessed 2 February 2021].

[4] Bluetooth® Technology Website. 2021. Bluetooth Radio Versions | Bluetooth® Technology Website. [online] Available at: <https://www.bluetooth.com/learn-about-bluetooth/radio-versions/> [Accessed 2 February 2021].

[5] SearchSQLServer. 2021. What is SQL (Structured Query Language)? - Definition from WhatIs.com. [online] Available at: <https://searchsqlserver.techtarget.com/definition/SQL> [Accessed 2 February 2021].

[6] PAT RESEARCH: B2B Reviews, Buying Guides & Best Practices. 2021. What is SAP HANA ? in 2020 - Reviews, Features, Pricing, Comparison - PAT RESEARCH: B2B Reviews, Buying Guides & Best Practices. [online] Available at: <https://www.predictiveanalyticstoday.com/what-is-sap-hana/> [Accessed 3 February 2021].

[7] www.javatpoint.com. 2021. SAP ABAP Tutorial - javatpoint. [online] Available at: <https://www.javatpoint.com/sap-abap> [Accessed 3 February 2021].

[8] Blogs.sap.com. 2021. SAPUI5: What is it and how does it work? | SAP Blogs. [online] Available at: <https://blogs.sap.com/2015/09/29/sapui5-for-dummies-what-is-it-and-how-does-it-work/> [Accessed 3 February 2021].

[9] Hamoodi, S., N.Hamoodi, A. and M.Haydar, G., 2020. *Automated Irrigation System Based On Soil Moisture Using Arduino Board*. [online] Available at: <http://beei.org/index.php/EEI/article/view/1736/1464> [Accessed 24 August 2020].

[10] Divani, D., Patil, P. and Punjabi, S., 2020. Automated Plant Watering System - IEEE Conference Publication. [online] Ieeexplore.ieee.org. Available at: <https://ieeexplore.ieee.org/document/7557245> [Accessed 24 August 2020].

[11] Gutierrez, J., Villa-Medina, J., Nieto-Garibay, A. and Porta-Gandara, M., 2014. Automated Irrigation System Using a Wireless Sensor Network and GPRS Module. IEEE Transactions on Instrumentation and Measurement, 63(1), pp.166-176 [Accessed 24 August 2020].

[12] Educative: Interactive Courses for Software Developers. 2021. What is C++?. [online] Available at: <https://www.educative.io/edpresso/what-is-cpp> [Accessed 3 February 2021].

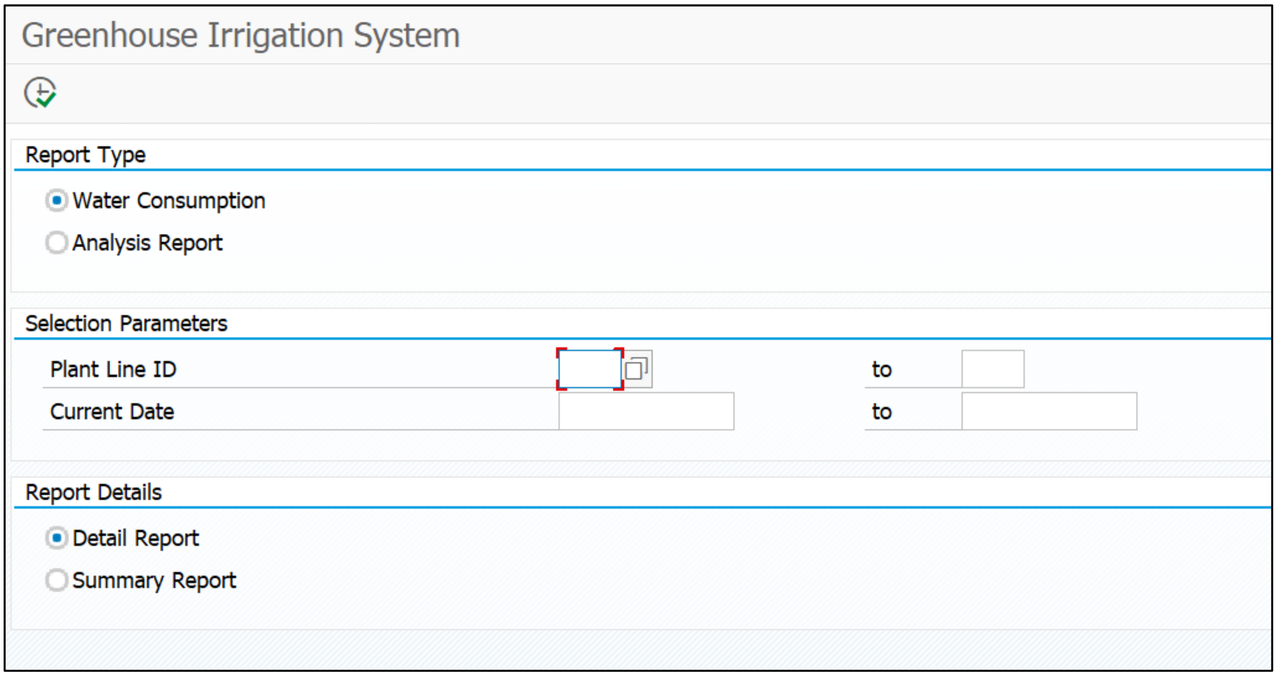
[13] Publication, I., 2020. *AUTOMATIC PLANT WATERING SYSTEM USING ARDUINO*. [online] Academia.edu. Available at: <https://www.academia.edu/40192969/AUTOMATIC_PLANT_WATERING_SYSTEM_USING_ARDUINO> [Accessed 24 August 2020].

[14] Patents.google.com. 2020. *US5097626A - Automatic Self-Watering System For Plants Growing In A Container - Google Patents*. [online] Available at: <https://patents.google.com/patent/US5097626A/en> [Accessed 24 August 2020].

A. Appendices– MIS Reports

Introduction

In the SAP System, there are many ways to take Report outputs and Adobe Forms is used to generate the below reports. Figure A-1 shows the Selection Screen for the reports which navigates to different report types with different selection parameters to make it easy to view required information.



Greenhouse Irrigation System

Report Type

● Water Consumption
○ Analysis Report

Selection Parameters

Plant Line ID to
Current Date to

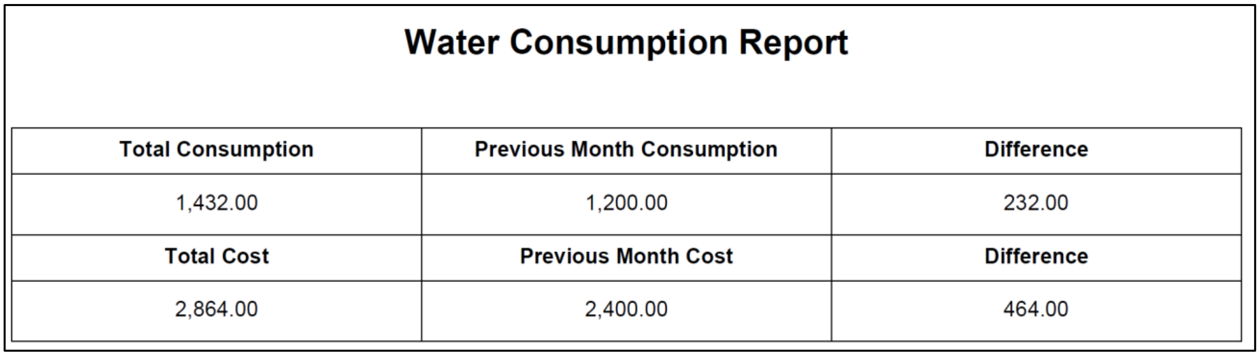
Report Details

● Detail Report
○ Summary Report

Figure A.1 Report Selection Screen

Water Consumption Report

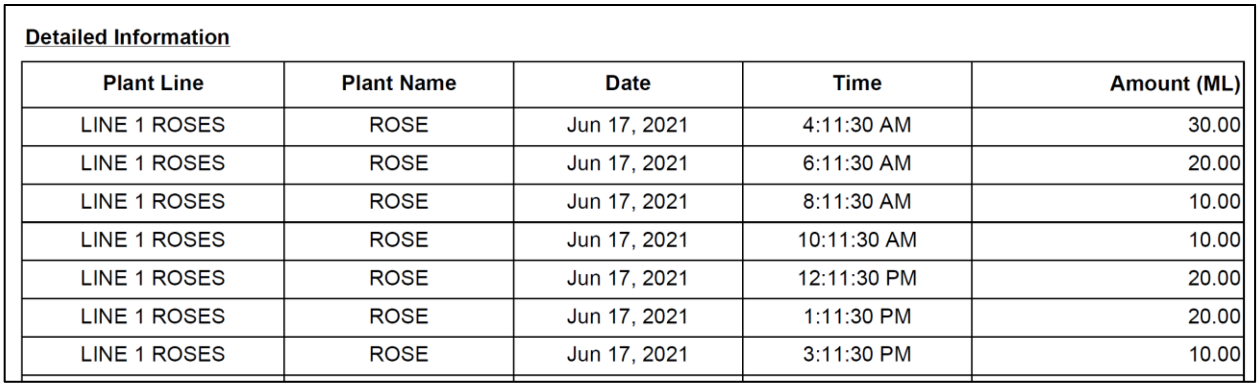
This report contains the total amount of water consumption for the specified period with the cost and total consumption for the previous period with cost to have a clear idea about the cost difference between the current period and the previous period. Figure A-2 shows the middle part of the report which contains the management information for decision making and a summary of the month for water consumption. Figure A-3 shows the detailed version of management information which contains all the recorded data with some calculations to convert captured sensor data to make valid information. The detailed Information part is separated into two parts which are Summary and Detailed. The detailed version contains all the recorded data with data and time and the summary version contains date-wise information to make readability easy for the user.



Water Consumption Report

Total Consumption	Previous Month Consumption	Difference
1,432.00	1,200.00	232.00
Total Cost	**Previous Month Cost**	**Difference**
2,864.00	2,400.00	464.00

Figure A.2 Management Information



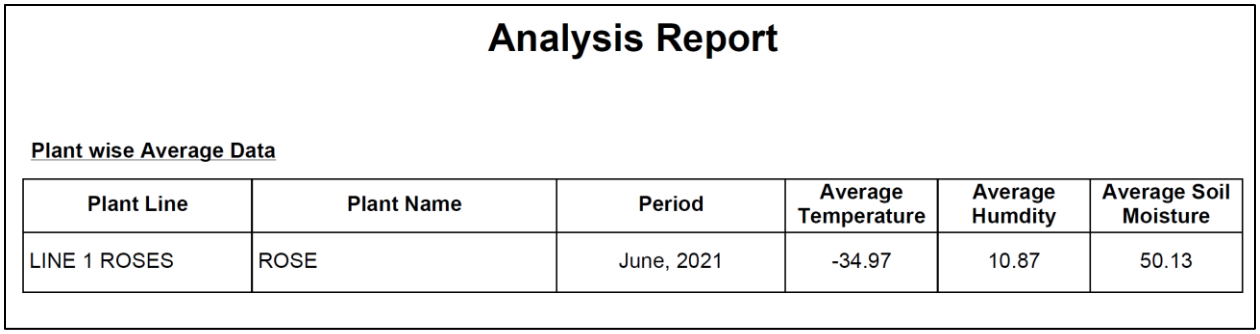
Detailed Information

Plant Line	Plant Name	Date	Time	Amount (ML)
LINE 1 ROSES	ROSE	Jun 17, 2021	4:11:30 AM	30.00
LINE 1 ROSES	ROSE	Jun 17, 2021	6:11:30 AM	20.00
LINE 1 ROSES	ROSE	Jun 17, 2021	8:11:30 AM	10.00
LINE 1 ROSES	ROSE	Jun 17, 2021	10:11:30 AM	10.00
LINE 1 ROSES	ROSE	Jun 17, 2021	12:11:30 PM	20.00
LINE 1 ROSES	ROSE	Jun 17, 2021	1:11:30 PM	20.00
LINE 1 ROSES	ROSE	Jun 17, 2021	3:11:30 PM	10.00

Figure A.3 Detailed Information

Analysis Report

The analysis report is the main report of this solution. All the captured data of the plants have been used to generate this report with selection parameters to segregate information as per the user requirement. This report contains three main sections to display data to users for making decisions easily and accurately by comparing old data which was captured. Figure A-4 shows the Plant line and Period wise data analysis with captured data to analyse plant behaviour with the sensor values. It will help users to understand how much water needs to plant with other conditions and make a decision about water moisture. Figure A-5 show the Harvest Details which is most important to the Management team in the company. It shows the selected period harvest and the previous period harvest to make decisions about continuing the Moisture parameter same or should be changed. Figure A-6 shows the detailed sensor data day-wise to give a better analysis of the plants and plant lines behaviour with temperature, humidity and soil moisture. Figures A-7 and A-8 show the full reports.

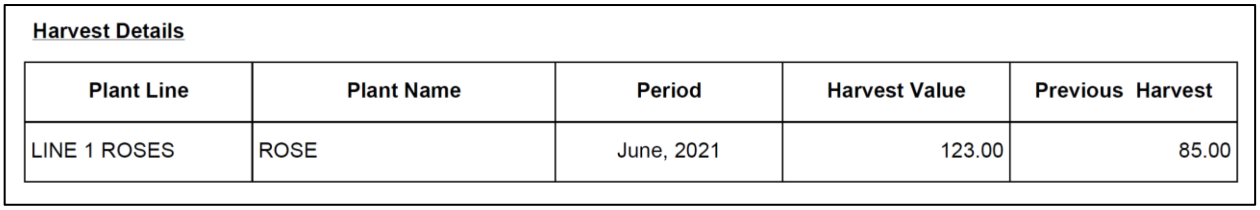


Analysis Report

Plant wise Average Data

Plant Line	Plant Name	Period	Average Temperature	Average Humdity	Average Soil Moisture
LINE 1 ROSES	ROSE	June, 2021	-34.97	10.87	50.13

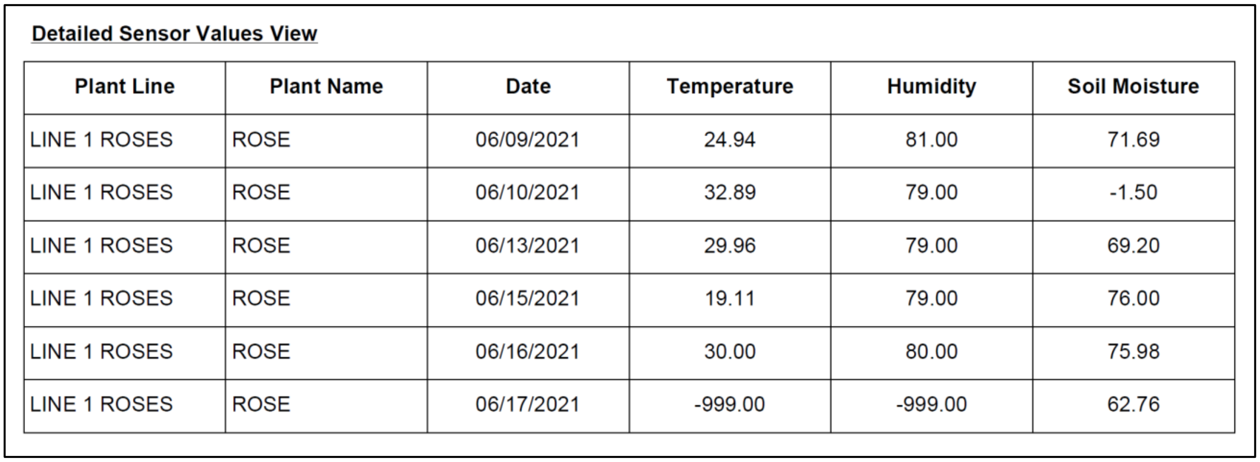
Figure A.4 Plant and Period Wise Data



Harvest Details

Plant Line	Plant Name	Period	Harvest Value	Previous Harvest
LINE 1 ROSES	ROSE	June, 2021	123.00	85.00

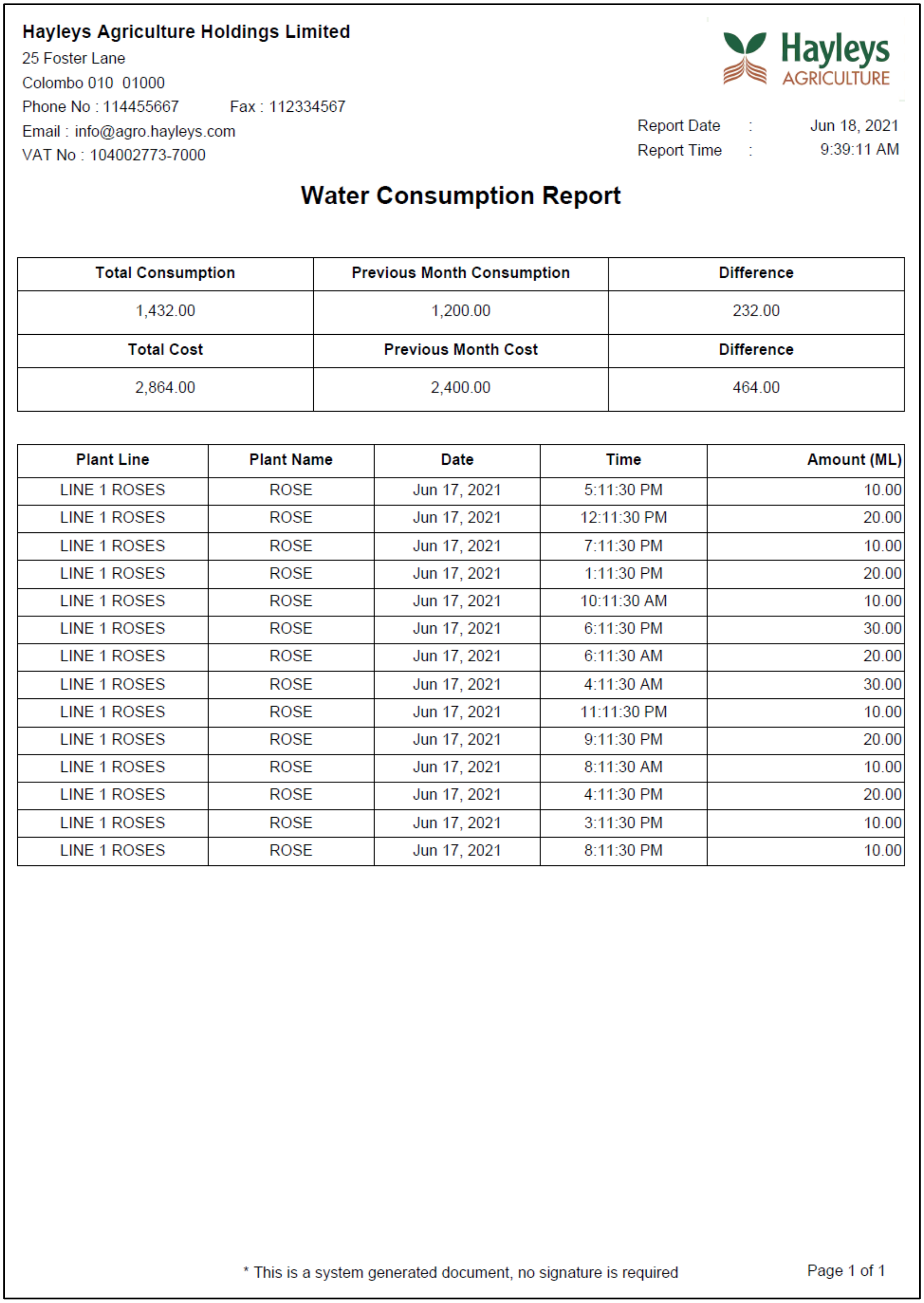
Figure A.5 Harvest Details



Detailed Sensor Values View

Plant Line	Plant Name	Date	Temperature	Humidity	Soil Moisture
LINE 1 ROSES	ROSE	06/09/2021	24.94	81.00	71.69
LINE 1 ROSES	ROSE	06/10/2021	32.89	79.00	-1.50
LINE 1 ROSES	ROSE	06/13/2021	29.96	79.00	69.20
LINE 1 ROSES	ROSE	06/15/2021	19.11	79.00	76.00
LINE 1 ROSES	ROSE	06/16/2021	30.00	80.00	75.98
LINE 1 ROSES	ROSE	06/17/2021	-999.00	-999.00	62.76

Figure A.6 Detailed Sensor Values View



Hayleys Agriculture Holdings Limited
25 Foster Lane
Colombo 010 01000
Phone No : 114455667   Fax : 112334567
Email : info@agro.hayleys.com
VAT No : 104002773-7000

Hayleys
AGRICULTURE

Report Date : Jun 18, 2021
Report Time : 9:39:11 AM

Water Consumption Report

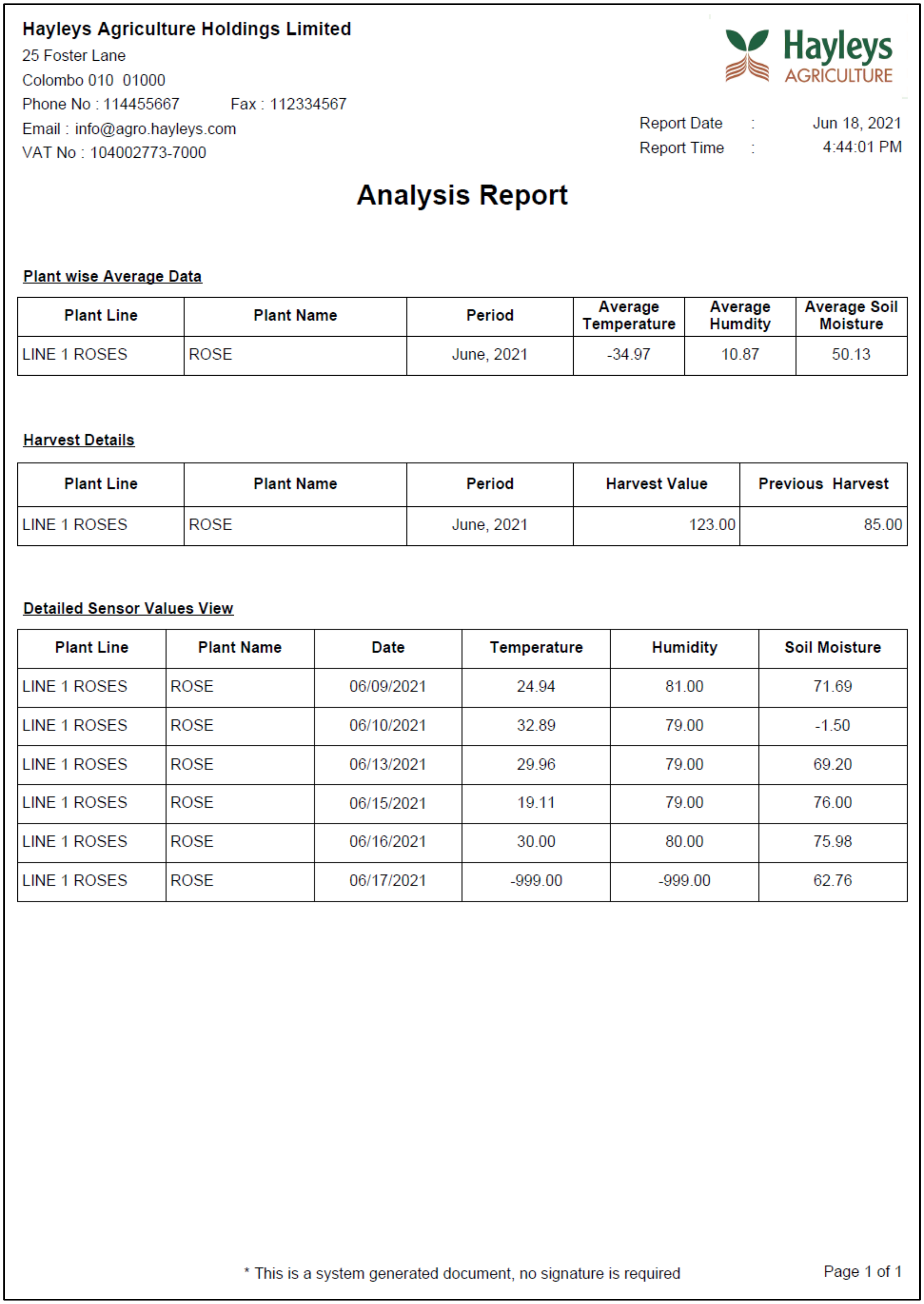
Total Consumption	Previous Month Consumption	Difference
1,432.00	1,200.00	232.00
Total Cost	**Previous Month Cost**	**Difference**
2,864.00	2,400.00	464.00

Plant Line	Plant Name	Date	Time	Amount (ML)
LINE 1 ROSES	ROSE	Jun 17, 2021	5:11:30 PM	10.00
LINE 1 ROSES	ROSE	Jun 17, 2021	12:11:30 PM	20.00
LINE 1 ROSES	ROSE	Jun 17, 2021	7:11:30 PM	10.00
LINE 1 ROSES	ROSE	Jun 17, 2021	1:11:30 PM	20.00
LINE 1 ROSES	ROSE	Jun 17, 2021	10:11:30 AM	10.00
LINE 1 ROSES	ROSE	Jun 17, 2021	6:11:30 PM	30.00
LINE 1 ROSES	ROSE	Jun 17, 2021	6:11:30 AM	20.00
LINE 1 ROSES	ROSE	Jun 17, 2021	4:11:30 AM	30.00
LINE 1 ROSES	ROSE	Jun 17, 2021	11:11:30 PM	10.00
LINE 1 ROSES	ROSE	Jun 17, 2021	9:11:30 PM	20.00
LINE 1 ROSES	ROSE	Jun 17, 2021	8:11:30 AM	10.00
LINE 1 ROSES	ROSE	Jun 17, 2021	4:11:30 PM	20.00
LINE 1 ROSES	ROSE	Jun 17, 2021	3:11:30 PM	10.00
LINE 1 ROSES	ROSE	Jun 17, 2021	8:11:30 PM	10.00

* This is a system generated document, no signature is required

Page 1 of 1

Figure A.7 Water Consumption Report



Hayleys Agriculture Holdings Limited
25 Foster Lane
Colombo 010 01000
Phone No : 114455667   Fax : 112334567
Email : info@agro.hayleys.com
VAT No : 104002773-7000

Report Date : Jun 18, 2021
Report Time : 4:44:01 PM

Analysis Report

Plant wise Average Data

Plant Line	Plant Name	Period	Average Temperature	Average Humdity	Average Soil Moisture
LINE 1 ROSES	ROSE	June, 2021	-34.97	10.87	50.13

Harvest Details

Plant Line	Plant Name	Period	Harvest Value	Previous Harvest
LINE 1 ROSES	ROSE	June, 2021	123.00	85.00

Detailed Sensor Values View

Plant Line	Plant Name	Date	Temperature	Humidity	Soil Moisture
LINE 1 ROSES	ROSE	06/09/2021	24.94	81.00	71.69
LINE 1 ROSES	ROSE	06/10/2021	32.89	79.00	-1.50
LINE 1 ROSES	ROSE	06/13/2021	29.96	79.00	69.20
LINE 1 ROSES	ROSE	06/15/2021	19.11	79.00	76.00
LINE 1 ROSES	ROSE	06/16/2021	30.00	80.00	75.98
LINE 1 ROSES	ROSE	06/17/2021	-999.00	-999.00	62.76

* This is a system generated document, no signature is required

Page 1 of 1

Figure A.8 Analysis Report

B. Appendices – Test Cases

Test Case ID: 5

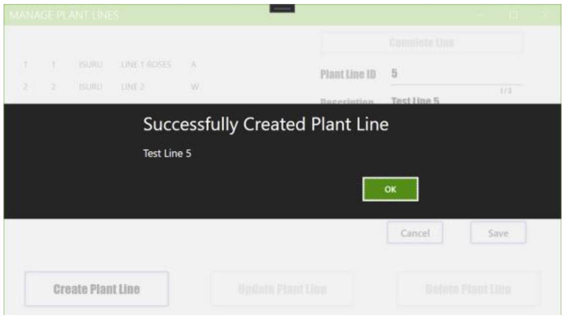
Test Case Title: Plant Line Create, Update, Delete

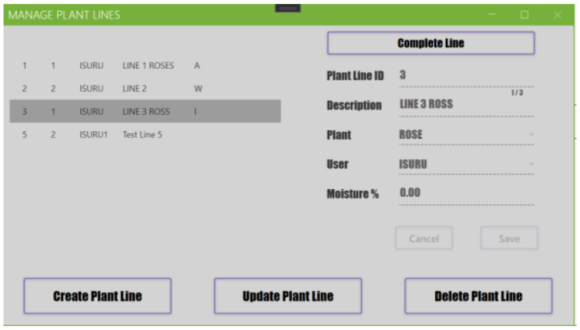
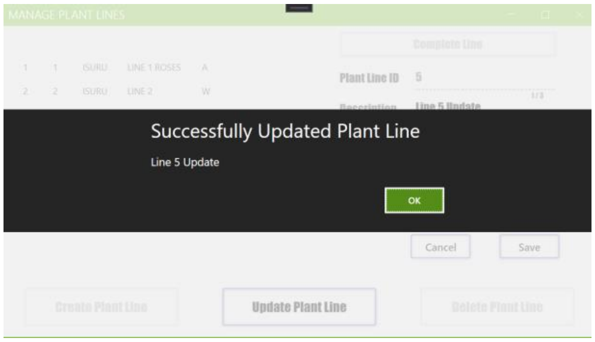
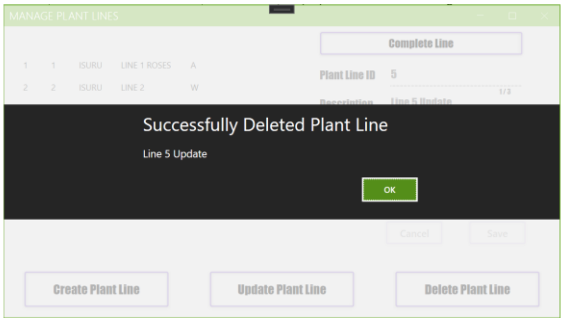
Module Name: Plant Line Management

Test Case Priority: High

Pre-conditions: SAP system should be connected to Application

Expected Result: Successfully Create, Update and Delete records and highlight invalid data fields

Step ID	Test Step	Test Data	Actual Result	Status
1	Go to Plant Line Management Window Click Create Plant Line Enter Plant Line ID Enter Description Select Plant Select User Enter Moisture Level Click Save Button	5 Test Line 5 Gerbera ISURU1 45.53	Display successful alert message  Successfully Created Plant Line Test Line 5 MANAGE PLANT LINES 1 1 ISURU LINE 1 ROSES A 2 2 ISURU LINE 2 W Complete Line Plant Line ID 5 Description Test Line 5 OK Cancel Save Create Plant Line Update Plant Line Delete Plant Line	Pass

2	Go to Plant Line Management Window Click Create Plant Line Select Plant Line from List		Successfully fill the relevant fields with data  MANAGE PLANT LINES 1 1 ISURU LINE 1 ROSES A 2 2 ISURU LINE 2 W 3 1 ISURU LINE 3 ROSS I 5 2 ISURU1 Test Line 5 Plant Line ID 3 Description LINE 3 ROSS 1/3 Plant ROSE User ISURU Moisture % 0.00 Complete Line Cancel Save Create Plant Line Update Plant Line Delete Plant Line	Pass
3	Go to Plant Line Management Window Click Create Plant Line Select Plant Line from List Click Update Change Values Click Save Button		Display successful alert message  MANAGE PLANT LINES 1 1 ISURU LINE 1 ROSES A 2 2 ISURU LINE 2 W Plant Line ID 5 Description Line 5 Update 1/3 Successfully Updated Plant Line Line 5 Update Complete Line OK Cancel Save Create Plant Line Update Plant Line Delete Plant Line	Pass
4	Go to Plant Line Management Window Click Create Plant Line Select Plant Line from List Click Delete Button		Display successful alert message  MANAGE PLANT LINES 1 1 ISURU LINE 1 ROSES A 2 2 ISURU LINE 2 W Plant Line ID 5 Description Line 5 Update 1/3 Successfully Deleted Plant Line Line 5 Update Complete Line OK Cancel Save Create Plant Line Update Plant Line Delete Plant Line	Pass

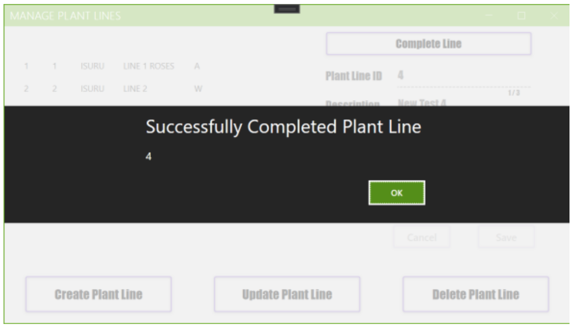
5	Go to Plant Line Management Window Click Create Plant Line Select Plant Line from List Click Delete Button		Display successful alert message  MANAGE PLANT LINES 1 1 ISURU LINE 1 ROSES A 2 2 ISURU LINE 2 W Plant Line ID 4 Description New Test 4 Successfully Completed Plant Line 4 Complete Line 1/3 OK Cancel Save Create Plant Line Update Plant Line Delete Plant Line	Pass

Table B.1 Plant Line Create, Update, Delete

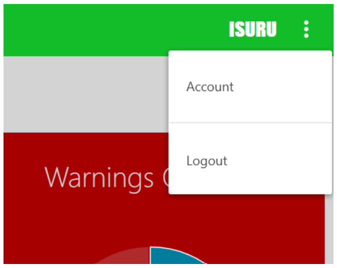
Test Case ID: 6

Test Case Title: Common Functions

Module Name: Common Functions

Test Case Priority: Medium

Expected Result: Satisfy the common system functionalities to run the system smoothly

Step ID	Test Step	Test Data	Actual Result	Status
1	Click Logout Button		Successfully Logout from the Application  	Pass

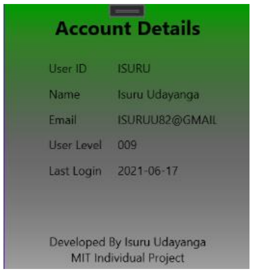
2	Click Account Button		Successfully display the Account Details  Account Details User ID ISURU Name Isuru Udayanga Email ISURUU82@GMAIL User Level 009 Last Login 2021-06-17 Developed By Isuru Udayanga MIT Individual Project	Pass

Table B.2 Common Functions

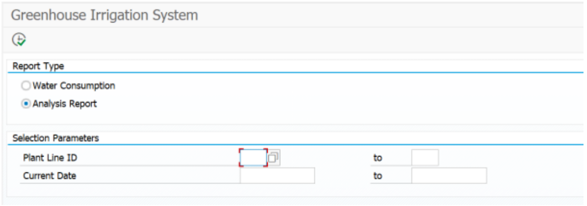
Test Case ID: 7

Test Case Title: Report Generation

Module Name: Reporting

Test Case Priority: High

Expected Result: Generate the report according to the variables in the selection screen

Step ID	Test Step	Test Data	Actual Result	Status
1	Login to SAP System Goto Reporting Tool		Successfully Logout from the Application  Greenhouse Irrigation System Report Type ○ Water Consumption ● Analysis Report Selection Parameters Plant Line ID to Current Date to	Pass

2	Select Analysis Report and execute the program with Plant Line ID and Date		Successfully display the Analysis Report according to the selected parameters.	Pass
3	Select Water Consumption and execute the program with Plant Line ID and Date Select Report Type for summary or detailed		Successfully display the water consumption Report according to the selected parameters.	Pass

Table B.3 Report Generation