VIETNAM NATIONAL UNIVERSITY - HO CHI MINH CITY
HO CHI MINH CITY UNIVERSITY OF TECHNOLOGY
FACULTY OF COMPUTER SCIENCE AND ENGINEERING

Internet of Things - CO3037

Smart Scheduling Irrigation System

Instructors/Lecturers: Dr. Le Trong Nhan

Authors: Bui Trung Hai - 2052082
La Vi Luong - 2052590
Tran Do Dang Khoa - 2053142

HO CHI MINH CITY, 4/2024

Contents

1 Introduction

1.1 Outline

Agriculture has always been a crucial industry in our society. From ancient times, when its products were manually collected and gathered, we found out that these processes were repetitive. Since then, the demand for an alternative way of making these works automatically done started to rise. People invented ways to simplify those processes, such as building waterways from streams or rivers to reduce the water carrying distance from the water source to the crops, or inventing agrimotor for faster and more efficient product gathering.

With the help of AI and modern technology, nowadays, our inventions are not only aiming to simplify those jobs, but rather making them become autonomical. In this project, we want to actualize a watering system for crops that automatically produces a fertilizer mixer and provides water and fertilizer to the plants. This document outlines the design for an Internet of Things (IoT) based irrigation system. This system utilizes Modbus 485 for communication and aims to automate the irrigation process while providing remote monitoring capabilities.

1.2 System structure

The system is provided with 8 actuators (or so called "relays") which includes:

- 3 for the fertilizer mixer (the IDs are 1, 2, 3 respectively)
- 3 for the area selector (the IDs are 4, 5, 6 respectively)
- 1 for the pump in process (the ID is 7)
- 1 for the pump out process (the ID is 8)

At the end, we will also have applications for monitoring and controlling the system from afar.

2 Requirements

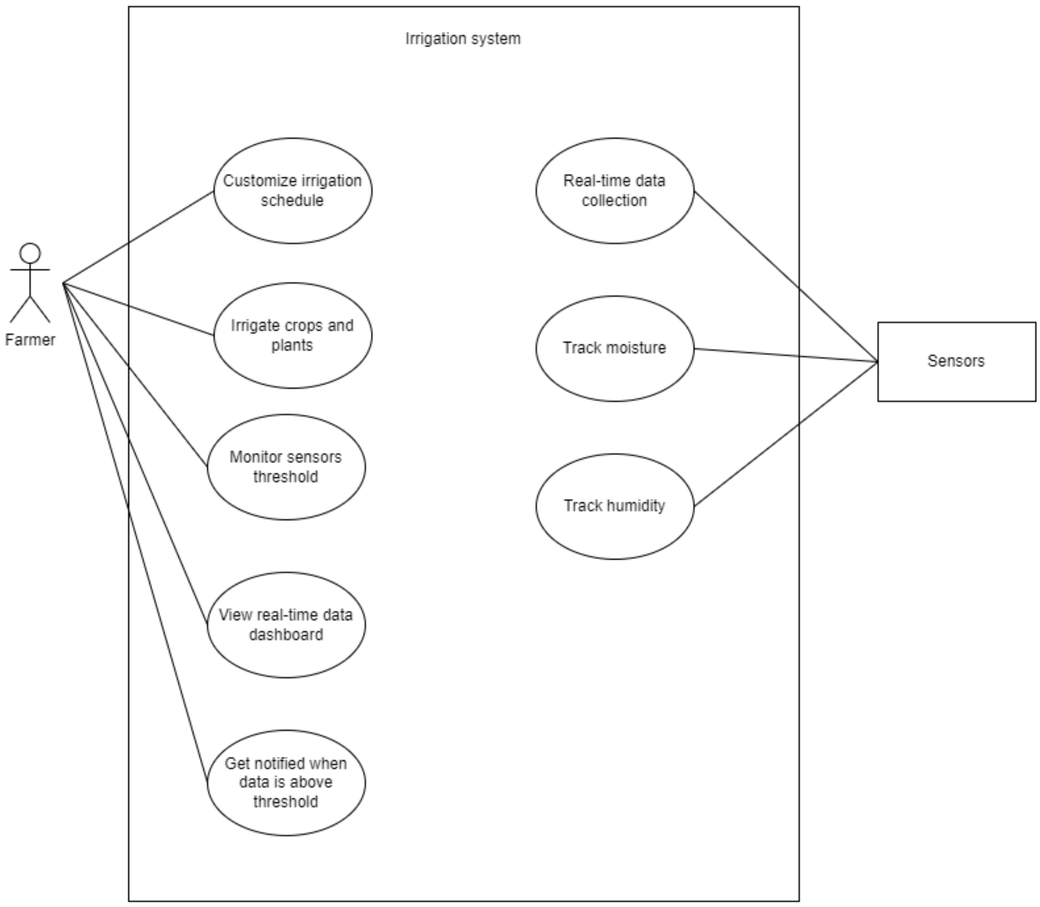
2.1 Functional requirements:

- The system shall allow the user to schedule the irrigation process
- The system shall notify the user if the fertilizer dispensing or pump operation times out
- The system shall provide notifications to a smartphone device

2.2 Non-functional requirements:

- The system shall be user-friendly and provide clear user interface elements for monitoring and control
- The system shall be reliable and operate consistently within defined parameters

3 Use cases



Irrigation system use-case diagram

3.1 Use case 1: Customize irrigation schedule

Actor	User
Description	Users can schedule the time for the irrigation process
Trigger	Users click on ”Schedule” button
Preconditions	User is on the application
Postconditions	User can set a specific time for the irrigation process to take place
Normal flow	1. User selects the ”Schedule” option from the application menu. 2. The application displays a scheduling interface. 3. User specifies the desired irrigation schedule. 4. User confirms the schedule. 5. The application saves the schedule and displays a confirmation message.
Alternative flows	
Exceptions	• User selects ”Cancel” at step 3 to return to the main menu without saving the schedule.

Table 1: Customize Irrigation Schedule

3.2 Use case 2: Irrigate crops and plants

Actor	User, Irrigation System
Description	System executes irrigation based on schedule or manual initiation, activates relays, monitors (simulated) flow, and logs irrigation event details.
Trigger	Irrigation schedule execution or user initiates irrigation manually
Preconditions	User has created an irrigation schedule (optional) or manually initiated irrigation
Postconditions	Irrigation cycle is complete
Normal flow	1. The irrigation system executes the irrigation schedule (if defined) or user initiates irrigation manually. 2. The system activates the fertilizer mixer relays based on the schedule (if applicable). • System monitors simulated flow sensor data • Upon reaching the target volume (or timeout), the system deactivates the fertilizer mixer relays. 3. The system activates the pump-in relay.
Alternative flows	• If flow sensor data is unavailable, the system sends a notification to the user's smartphone. • If the fertilizer dispensing or pump operation times out, the system notifies the user.
Exceptions	• System malfunction: The system logs the error and attempts to recover or notifies the user for intervention.

Table 2: Irrigate Crops and Plants

3.3 Use case 3: Monitor thresholds

Actor	User
Description	User monitors simulated timeouts used to represent thresholds
Trigger	User selects "Monitor" option from the application menu
Preconditions	User is logged in to the irrigation system application
Postconditions	User is informed about the irrigation system's status and simulated timeouts.
Normal flow	1. User selects the "Monitor" option from the application menu. 2. The application displays the current status of the irrigation system, including: * Upcoming scheduled irrigation (if applicable) * Completion status of the last irrigation cycle * Timeout information (for demonstration purposes) * Fertilizer dispensing timeout * Pump operation timeout
Alternative flows	• User selects specific details (e.g., fertilizer tank levels) for further information (applicable if additional sensors are implemented).
Exceptions	None
Notes	Simulated timeouts are used in this demonstration due to the lack of real sensors.

Table 3: Use Case: Monitor Thresholds (Simulated Timeouts)

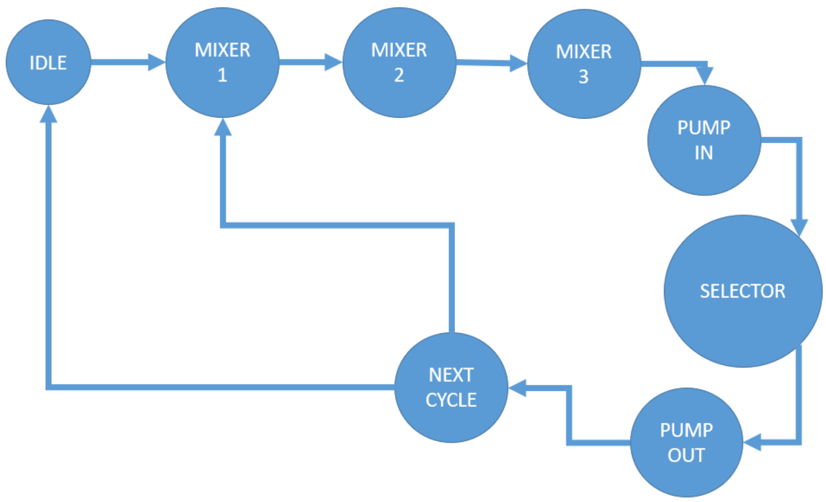
3.4 Use case 4: View real time data dashboard

Actor	User
Description	User views a real-time data visualization dashboard of the irrigation system.
Trigger	User selects "Dashboard" option from the application menu
Preconditions	User is logged in to the irrigation system application
Postconditions	User has a real-time overview of the irrigation system's operation
Normal flow	1. User selects the "Dashboard" option from the application menu 2. The application displays a real-time data visualization dashboard: • Scheduled irrigation details • Irrigation cycle status • Simulated timeout indicators
Alternative flows	

Table 4: View Real-time Data Dashboard

4 Design

4.1 Scheduler



Finite state machine for the irrigation system

The irrigation system employs 8 actuators, including 3 dedicated to the fertilizer mixer, 3 for selecting the irrigation area, 1 for the pump that brings water in (pump in), and 1 for the pump that removes water (pump out). The setup consists of two water pails and three fertilizer tanks serving as sources for the fertilizers

The operation begins with the fertilizer mixer drawing precisely 20ml from the first fertilizer tank, verified by a flow sensor to ensure accurate measurement. Once the flow sensor confirms the intake of 20ml, the mixing process proceeds to ensure the fertilizer is properly blended.

After the mixing is complete, the water pump in process is initiated to draw water from the water pails. If the system detects a lack of water through the sonar sensors, the pump in operation is halted after a predefined timeout to prevent damage or inefficiency.

The area selector actuators then determine the specific irrigation zones to receive the mixed fertilizer solution. The pump out actuator ensures the removal of any excess or waste water from the system. This setup ensures precise control and automation in the irrigation process, optimizing resource usage and maintaining system integrity through real-time monitoring and feedback.

4.2 Technical Stack

4.2.1 Frontend

Flutter

The mobile application for your IoT irrigation system will be built using Flutter, a popular framework for developing cross-platform apps. This choice offers several advantages for the project, including:

- Flutter is a cross-platform tool, allowing us to write the codebase once and deploy it on both Android and iOS devices, saving development time and resources.
- Flutter features hot reload, allowing you to see changes in the UI almost instantly after making code modifications, which significantly speeds up the development process, leading to fast development.
- A vast developer community supports Flutter, providing numerous resources, tutorials, and libraries to aid you in development.

In conclusion, Flutter provides a compelling set of advantages for building the mobile application component of your real-time tracking irrigation system. Its cross-platform capabilities, real-time rendering, UI flexibility, and strong community support make it an ideal choice for creating a user-friendly and efficient mobile dashboard for the project.

4.2.2 Backend

Python

The backend of your IoT irrigation system will be built using Python, a versatile and powerful programming language. This choice offers several advantages for the project, including:

- Python's ease of development and readability enable rapid development and easy maintenance, crucial for a project requiring quick iterations and clear, collaborative coding.
- A vast and active community supports Python, offering abundant resources, tutorials, and forums, ensuring efficient problem-solving and continuous learning opportunities.
- Python's cross-platform compatibility ensures seamless interfacing with diverse IoT devices, providing flexibility in an environment with heterogeneous devices.
- Python excels at integrating with other technologies and protocols commonly used in IoT, such as HTTP, MQTT, and CoAP, facilitating smooth communication between devices and the backend.
- Python's dynamic nature allows for rapid prototyping, enabling quick testing and iteration, which is vital in the fast-evolving IoT landscape.

In conclusion, Python provides a compelling set of advantages for building the backend component of your real-time tracking irrigation system. Its ease of use, extensive libraries, strong community support, and cross-platform capabilities make it an ideal choice for creating a reliable and efficient backend for the project.

4.3 Workflow

4.3.1 System Initialization

When the user opens the Flutter application, the system initializes and establishes a connection to the backend server. During this initialization phase, the application loads the initial dashboard view. This dashboard serves as the central hub for monitoring and controlling the irrigation system.

4.3.2 Dashboard Overview

The dashboard displays real-time data from the irrigation system. This includes information such as water levels in the pails, the levels of fertilizer in the tanks, the status of the pumps, and readings from various sensors.

4.3.3 Fertilizer Mixing Process

When the user initiates the fertilizer mixing process, the system activates the actuators responsible for drawing fertilizer. Specifically, it draws 20ml from the first fertilizer tank. The flow sensor monitors the volume, ensuring that exactly 20ml is dispensed. Once this amount is reached, the system stops drawing fertilizer, and the dashboard updates to reflect the status of the mixing process.

4.3.4 Water Pump In Process

Following the completion of the fertilizer mixing, the system checks the water level in the pails using sonar sensors. If sufficient water is detected, the pump in process is activated, drawing water into the system.

4.3.5 Area Selection and Irrigation

The user can select the specific area to be irrigated through the application. The area selector actuators then position themselves to direct the mixed fertilizer solution to the chosen area.

4.3.6 Pump Out Process

Once the irrigation process is complete, the pump out process is activated to remove any excess or waste water from the system.

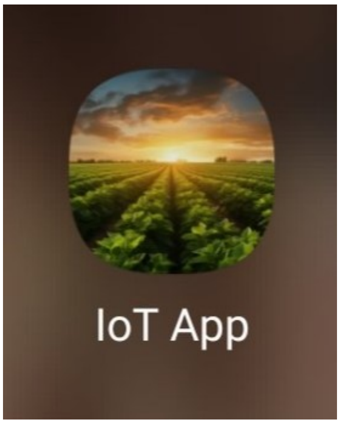
4.3.7 Real-time Monitoring

Throughout the operation, the application continuously monitors the status of the irrigation system.

5 Demonstration

5.1 General Design

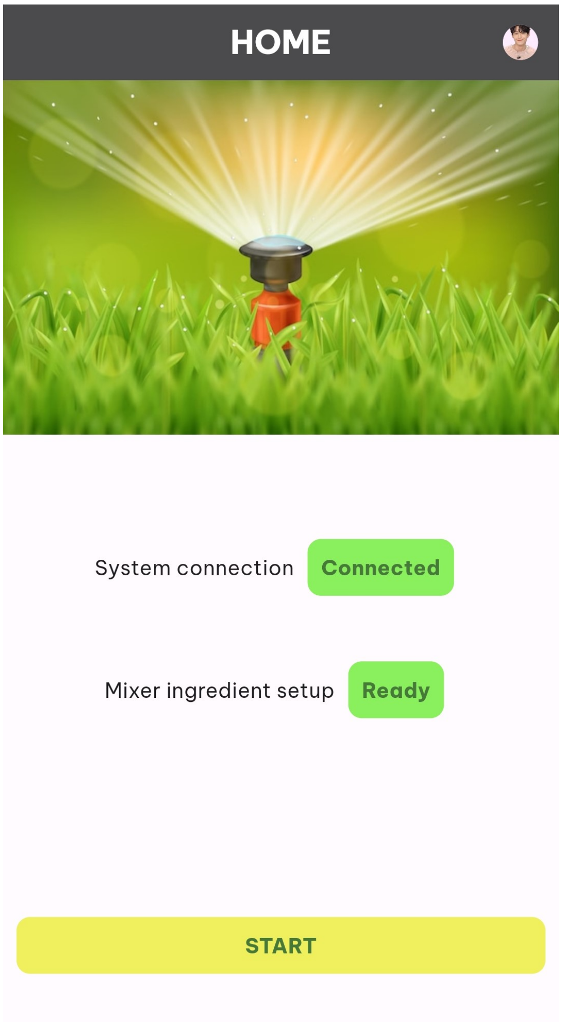
The idea behind the design of this application is to make the user interfaces as simple and understandable as possible. Therefore, the application uses a theme palette including the following colors: gray, green, olive green, red, orange, yellow, and blue.
Below is the application icon:



Application Icon

The process of watering crops are divided into sub-processes. Those can be divided into 3 groups: the mixer-related group, the area selection group, and the pump-related group. The pages design are built based on these so we will have 3 pages and 1 more for the home page.
The code can be access through Github: https://github.com/cosmicleon/IoT-Lab

5.2 Home Page



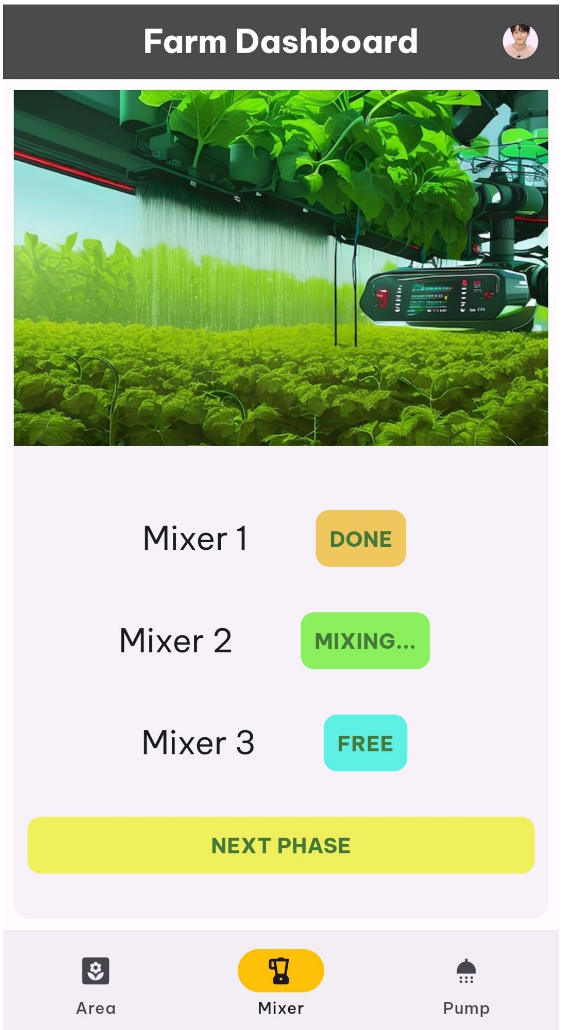
Home Page

This is the first page the user will interact once entering the app. The design of the home page is simple. The page featuring 3 functionalities:

- Checking the system connection: Checking the connection is important. There will be a status box that will display "Connected" in green if the application connection with the device is currently active, or "Disconnected" in red if the opposite occurs. The demo server used for this project is by default the main server. If by any means the user wants to change connection, they can press on the status box to navigate to a sub-section where they can alter the connection.

- Mixer ingredient setup: To ensure the fertilizer and water are ready for the go, there will be a box indicating the ingredient status before starting. The process can still start without the ingredient though.
- Start button: When the connection is stable enough to progress, pressing this button will navigate the user to the triple-feature page where they monitor and control the flow.

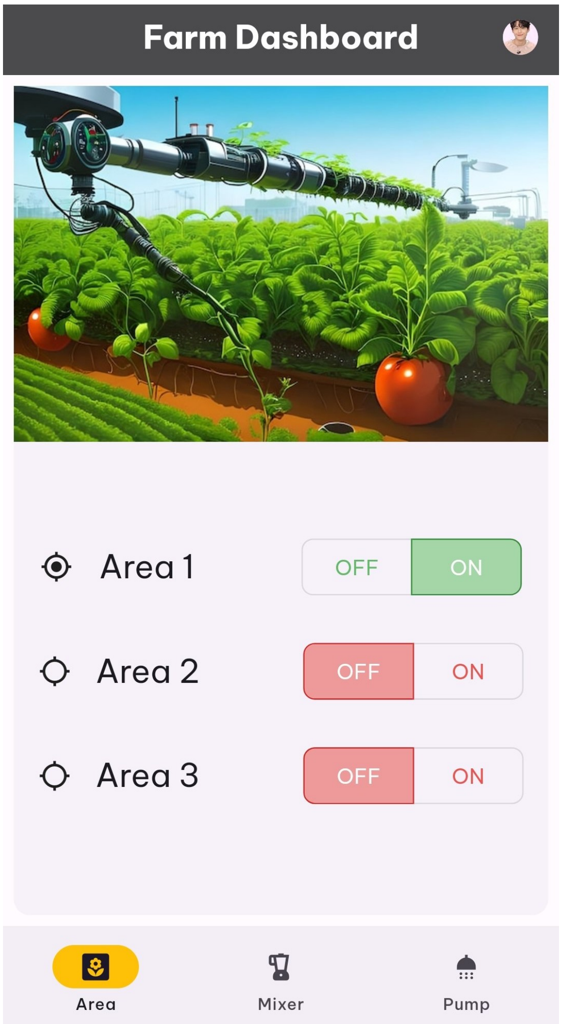
5.3 Mixer Activation Page



Mixer activation page

After pressing "Start" from the home page, the application will navigate to this page. There are three buttons next to three mixers' lables. By default the status of those 3 are "Free". If you press on any of them, the according mixer will be activate for this process. The user waits until the status changed from "Mixing..." to "Done" to proceed to the next phase.

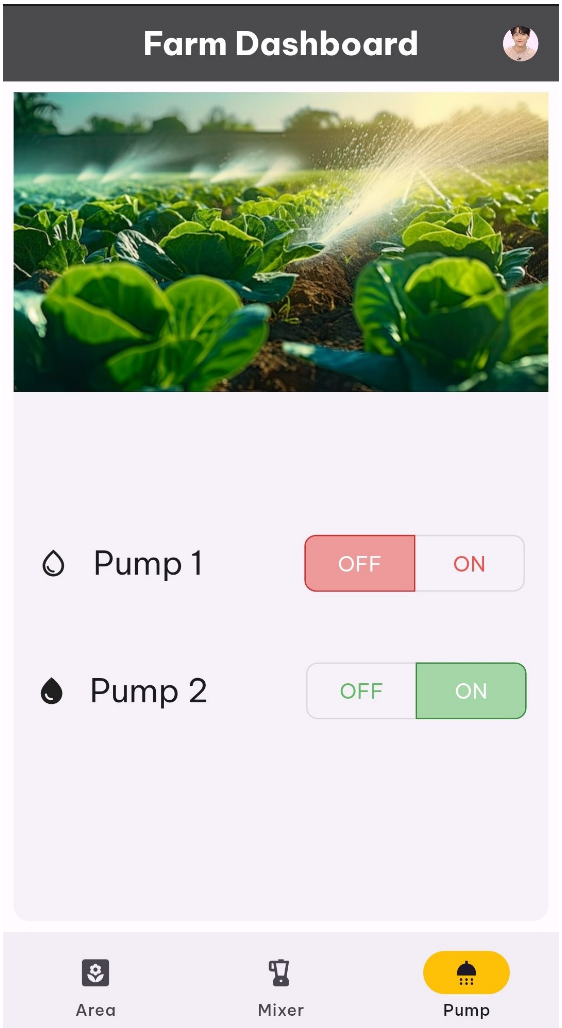
5.4 Area Selection Page



Area selection page

This is the area selection page. This page will display 3 toggle buttons for the user to choose which area to be watered. To ensure the watering process works correctly, the user should not change it in mid-process.

5.5 Pump Control Page



Pump control page

Finally is the pump controlling page. By choosing to turn the pumps on or off, the user can control the water flow to the mixer and the mixed ingredients to the crops.

6 Conclusion

In conclusion, we have successfully implemented the user interface for the mobile application of our IoT irrigation system using Flutter. This allows users to interact with a sleek, responsive, and user-friendly dashboard on both Android and iOS devices. The cross-platform capabilities of Flutter, combined with its hot reload feature, have significantly accelerated our development process, resulting in a robust and visually appealing UI.