SUP'COM
Higher School of Communication of Tunis

Scope statement: Hydroponic Tomato Farm Monitoring and Management System - Smart Hydro

Authors:

-Ben Frej Angela -Kaouech Mohamed

Faculty Adviser:

- MOHAMED BECHA Kaaniche

Contents

1 Concept

1.1 Project context

This project focuses on developing a hydroponic farming management system tailored for tomato cultivation, incorporating IoT, Machine Learning, and Location-Based Services (LBS). The system will leverage IoT sensors to monitor plant health and environmental conditions in real-time, while a machine learning model will analyze captured images to detect plant diseases. These components will be integrated through a Jakarta EE middelware, hosted on a WildFly server, with MQTT facilitating efficient communication between IoT sensors and the back-end infrastructure. A progressive web application will provide a user-friendly interface for farmers, allowing them to view sensor data, receive disease alerts, and access actionable insights on mobile or desktop devices.

1.2 Problem statement

Hydroponic farming presents unique challenges in environmental control and plant health monitoring. Traditional manual inspection for diseases is time-consuming, and delayed intervention can lead to significant crop losses. This project addresses these issues by combining IoT data with machine learning for early disease detection and automated system adjustments.

1.3 Objectives

- Integrate IoT sensors to continuously monitor environmental conditions in a hydroponic tomato farm.
- Develop a computer vision model to detect common tomato plant diseases from captured images.
- Implement a Progressive Web Application (PWA) for real-time data visualization and remote management.
- Establish a reliable middleware using Jakarta EE and WildFly to handle IoT data, ML predictions, and user interactions.

2 Client

The primary clients are agricultural technology companies, hydroponic farm operators, and agronomists focused on optimizing indoor crop production and health monitoring through innovative technologies.

3 Functional need

3.1 Sensors and IoT network

The IoT network will utilize a variety of sensors to continuously monitor the health and growth conditions of tomato plants. The main sensors include:

- Temperature Sensor: Measures the ambient temperature around the plants and inside the pot, helping to ensure optimal growing conditions. Alerts will be generated if the temperature goes outside of the ideal range for tomato growth.
- Humidity Sensor: Tracks the humidity level in the environment, which is crucial for plant health. High or low humidity levels can stress the plants and make them more susceptible to diseases.
- Conductivity sensor: Measures the electrical conductivity of a solution, which correlates to the concentration of ions, typically in water. This sensor is often used in water quality monitoring .
- pH Sensor: Maintains the pH of the hydroponic solution, which affects nutrient uptake and overall plant health. Regular monitoring allows for quick adjustments to keep the pH within the optimal range.

A display will show the value of each sensor. The sensors must send the data to the board, which stores them in the Cloud. An alert system is needed to notify the user of the issue (elevated temperature ...) for a variable duration, and can be disabled by the user.

3.2 PWA application

This application will enable the user to:

- View the results (table for nutriments, humidity, and temperature evolution) with an option to choose the period of visualization.
- Predict their health status (whether they are healthy or there is a disease).
- Utilize Location-Based Services (LBS) to pinpoint the nearest user's location to the plant that has a disease

4 Machine learning integration

In the Hydroponic Farming Management System, the YOLOv10 (You Only Look Once, version 10) model is used as the core image processing architecture for detecting disease symptoms in tomato plants. YOLO models are known for their speed and accuracy, making them highly suitable for real-time detection tasks in dynamic environments like hydroponic farms.

It divides images into a grid and simultaneously predicts bounding boxes and class probabilities for each cell, allowing for fast, single-pass detection. YOLOv10 incorporates an

improved backbone network and an enhanced Feature Pyramid Network (FPN), which enhances its ability to detect objects of varying sizes. Its anchor-free mechanism and optimized loss function further increase efficiency, making it ideal for applications requiring rapid detection on edge devices.

This graph(figure 1) compares the AP (Average Precision) against latency for various YOLO versions and other object detection models. YOLOv10 (shown in red) consistently achieves the highest AP at each latency point, indicating superior accuracy and speed. This makes it ideal for real-time applications, as it offers a better balance of precision and efficiency compared to previous YOLO models and competing architectures.

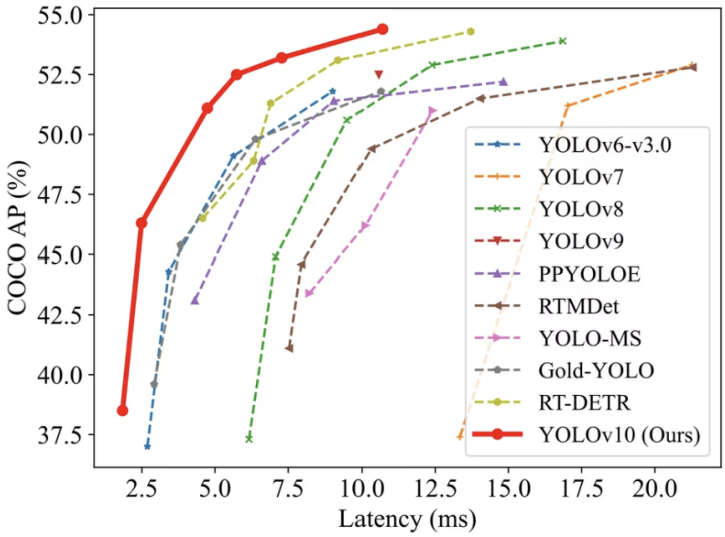

Figure 1: Architecture

5 Equipment

After conducting research, we have identified the following elements as necessary:

- The BME280: Humidity and temperature Sensor
- Ph sensor
- Raspberry Pi 4
- Conductivity sensor
- A camera with a minimum resolution of 160x160 for a distance of 20 cm.
- Connecting wires.

6 Technologies choice

6.1 Back-end

- MongoDB: A NoSQL document-oriented database used for storing user data. It is practical and easy to use in conjunction with Jakarta EE.
- MQTT: A lightweight publish-subscribe network protocol used to communicate sensor data to an MQTT broker in the cloud.

6.2 Middleware

- JAX-RS: Java API for RESTful Web Services is a Java programming interface for creating web services with a REST architecture.
- WildFly: WildFly, formerly known as JBoss Application Server or JBoss, is a free and open-source Java EE application server written in Java and released under the GNU LGPL license. It can be used on any operating system that provides a Java virtual machine.
- Mosquitto: Mosquitto is a widely used MQTT broker, serving as an intermediary for efficient and reliable messaging between IoT devices and the back-end system.

6.3 Front-End

- PWA (Progressive Web App): A PWA is a cross-platform web application that provides a native-like experience to users. It allows you to develop applications for multiple platforms using web technologies like HTML, CSS, and JavaScript. PWAs are easy and quick to code and do not require prior web development knowledge.

6.4 Iot integration

- Flogo: Flogo is an open-source project designed for creating lightweight, event-driven microservices and workflows, making it suitable for orchestrating data flows, data pre-processing, and automation in IoT scenarios.

7 Architecture

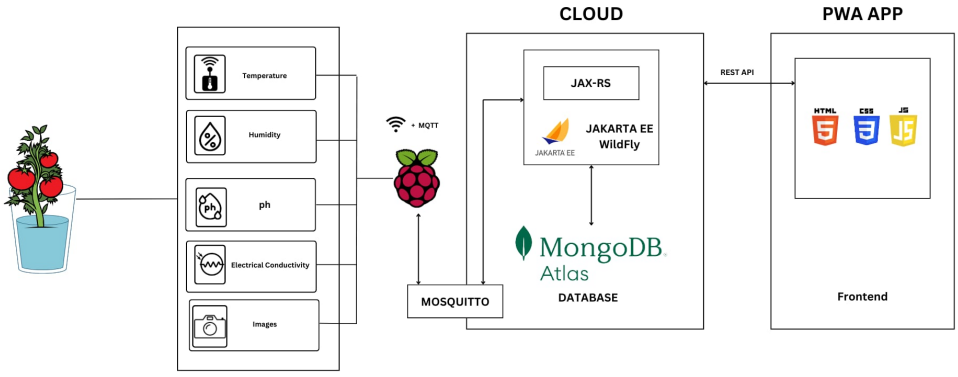

Figure 2: Architecture

The diagram above describes the main architecture of the plant monitoring system which is mainly composed by Backend, Middleware and Frontend.

8 Timeline and tasks

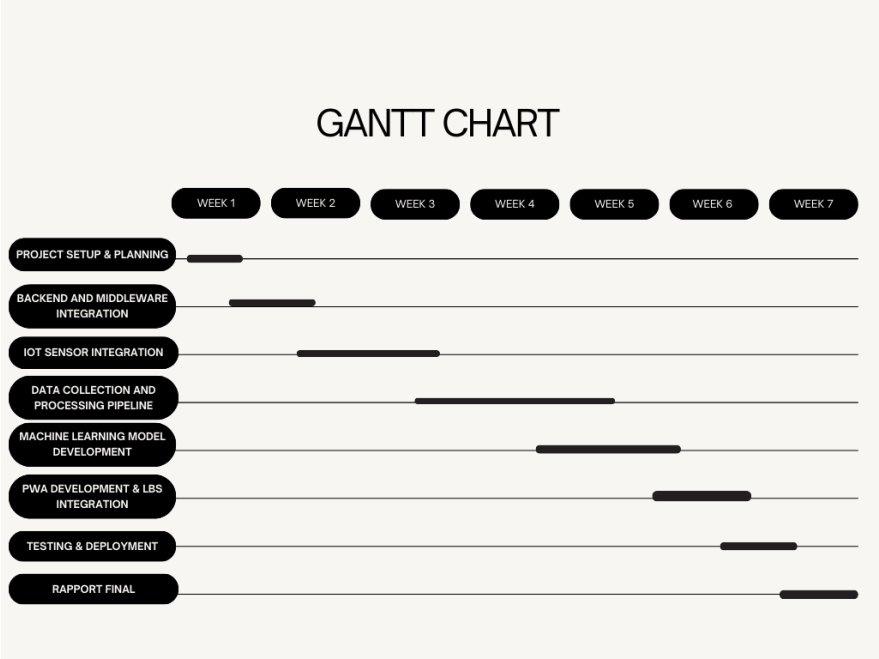

Figure 3: Gantt Diagram.

The Gantt diagram, which enables the project's progress to be graphically represented, is shown in the explanation above. It makes it possible to visually illustrate the project's progress and the length of each process.

9 Methodology

Throughout the project, we will be utilizing Extreme Programming (XP), an Agile software development framework renowned for its commitment to delivering superior software quality and enhancing the personal satisfaction of the development team. This methodology is exceptionally well-suited to our project due to its adaptability, specifically tailored for short-duration projects where last-minute requirement changes are common.

9.1 Principles

XP's principles align with those of agile methodologies but are distinguished by their extreme emphasis. XP is founded on:

- High responsiveness to changing customer needs.
- Teamwork, in our context represented by Pair Programming.
- Delivering high-quality work.
- Early, high-quality testing.

9.2 Process

The XP framework normally involves 5 phases or stages of the development process that iterate continuously:

1. Planning: In the initial stage, we create our own user stories and define the desired outcomes. Requirements are provided, and we estimate the stories, creating a release plan that breaks the project into iterations to cover the required functionality incrementally. If certain stories cannot be accurately estimated, we introduce "spikes," indicating that further research is necessary.
2. Design: While designing is an integral part of the planning process, it is essential enough to be highlighted separately. It is closely tied to one of the core XP values we will discuss later – simplicity. A well-thought-out design brings logical structure to the system, helping us avoid unnecessary complexities and redundancies.
3. Coding: This phase involves the actual implementation of code, and it follows specific XP practices, including adherence to coding standards, pair programming, continuous integration, and collective code ownership.

4. Testing: Testing is at the heart of Extreme Programming. It is an ongoing activity encompassing both unit tests, which are automated tests that verify the proper functioning of individual features, and acceptance tests, where customers assess whether the system aligns with the initial requirements.

5. Listening: Listening emphasizes continuous communication and feedback. Coaches and project managers play a pivotal role in conveying the business logic and expected value to the team, ensuring a shared understanding of the project's goals and requirements.

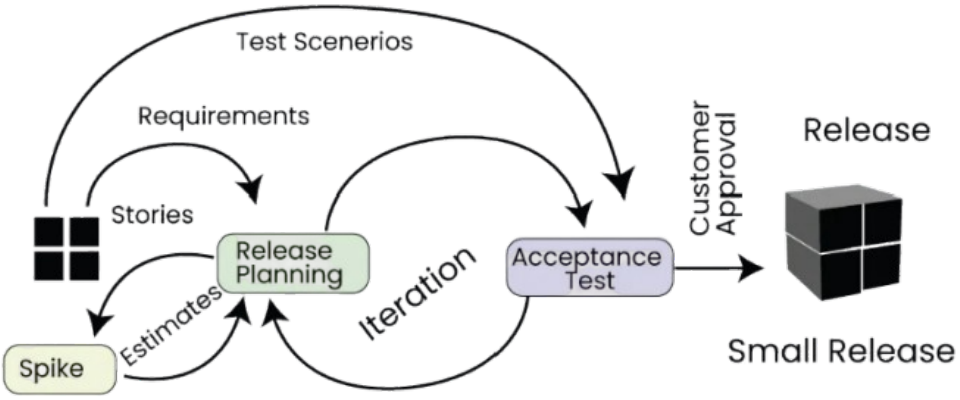

Figure 4: Extreme programming life-cycle

10 Limitations

As we embark on the development of our healthcare monitoring system, it's important to anticipate and address various challenges that may arise during its implementation.

- Internet Connectivity Interruptions: One limitation lies in the reliance on internet connectivity. In cases of internet outages, real-time monitoring and user alerts for critical health changes may not be achievable.

- Sensor Data Accuracy: There's the possibility of false alarms due to inaccuracies in sensor data or anomalies detected by the algorithm, potentially causing unnecessary concerns for users.

- Proximity of IoT Devices: The correct identification of health data may be affected by the proximity of IoT devices to each other. If IoT devices are placed very closely, they might detect changes in neighboring devices, impacting the accuracy of the data.

- Sensor Placement and Thresholds: Placing the IoT sensors correctly is critical. Inaccurate placement or poorly defined thresholds for measured metrics may lead to misinterpretations and unnecessary alerts.

- Data Volume and Processing: Handling a large volume of health data generated by multiple users and devices can be challenging. Processing and analyzing this data efficiently is essential for providing real-time insights.
- Image quality : A key limitation of the system involves image capture quality, as varying luminosity and lighting conditions can significantly affect the accuracy of disease detection. Low light or overexposure may obscure symptoms or create shadows, leading to potential misclassifications by the machine learning model

11 Business study

The Business Study section provides an overall view of our project's business model and marketing policy.

11.1 Business Model Canvas (BMC)

The Business Model Canvas serves as a visual representation of our project's main aspects, such as value proposition, customer segmentation, channels, cost structure, revenue stream, and more. The BMC is depicted in the figure below:

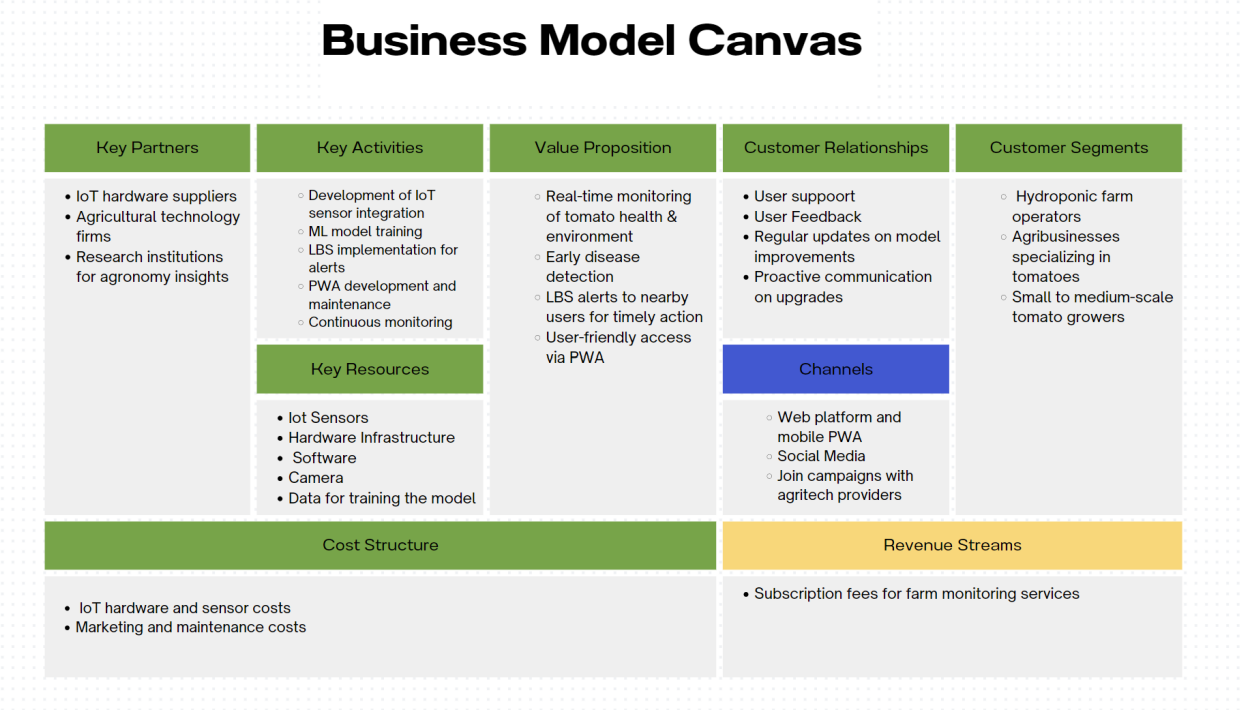

Figure 5: Business Model Canvas

11.2 Marketing policy

Our marketing policy focuses on delivering value to customers and establishing a robust presence in the agriculture technology market. The policy includes:

11.2.1 Product

- A sophisticated IoT architecture designed for seamless health and environmental monitoring in hydroponic tomato farms.
- User-friendly management of plant health monitoring accessible through a Progressive Web Application (PWA)
- Integration of Location-Based Services (LBS) to provide timely health alerts to the nearest users.
- Continuous improvement of disease detection accuracy through MLOps, ensuring the model adapts to new conditions.

11.2.2 Price

- Competitive pricing with customizable rates to suit individual users and healthcare institutions.
- Warranty and reliable after-sales support to maintain customer satisfaction and system reliability.

11.2.3 Promotion

- A multi-channel approach for promotion, including digital marketing, partnerships with agricultural co-ops, and a referral program.
- Joint promotional campaigns with agricultural technology providers to expand reach and customer base.

11.2.4 Place

- Our services are accessible through a web platform and mobile-friendly PWA, providing convenient, flexible access to health monitoring and insights for farmers and agribusinesses alike.

12 Deliverables

- Conceptual Document: This document presents in a detailed and structured manner the specifications and the services to be provided.
- Source code of the various project components on GitHub
- Prototype/simulation of the intelligent healthcare monitoring system.
- Demonstration Video: An mp4 format video that contains a demonstration of the proposed solution.