



Software Engineering Department

Braude College

Capstone Project Phase A

An Internet of Things Sensor's System – Early Detection System for Avocado Tree Diseases

By:

Bahaldeen Swied

Rabea Lahham

Advisor:

Dr. Naomi Unkelos Shpigel



Project Code:

25-1-D-13

Link to GitHub:

<https://github.com/oRABiiA/AgriTech>

Table of Contents

Abstract	4
1. Introduction	4
2. Literature Review	5
2.1.Wireless Sensor Network (WSN) in Agriculture	5
2.2. IOT in Agriculture	5
2.3 Precision Agriculture Using IoT	6
2.3.1 Sensors	6
2.3.1.1 Air Humidity Sensor	6
2.3.1.2 Soil Moisture Sensor	7
2.3.1.3 Cameras for Early Disease Detection	7
2.4. Image Analysis in Agriculture	7
2.5 Avocado Diseases	8
2.5.1 Root rot	8
2.5.2 laurel wilt	8
3. Research	9
3.1 Scheduled Meetings	9
3.1.1 Meetings Summary	9
3.1.1.1 Meeting Number 1 - 20.11.2024	9
3.1.1.2 Meeting Number 2 - 27.11.2024	9
3.1.1.3 Meeting Number 3 - 18.12.2024	9
3.1.1.4 Meeting Number 4 - 8.1.2025	9
4. Engineering Process	10
4.1 Development Process	10
4.2 User-Centered Design Approach	10
4.3 Workflow	10
4.4 Solution	11
4.4.1 Algorithm	11
4.5 Technologies Review	15
4.5.1 Hardware Side	15
4.5.1.1 Technical Details	15
4.5.1.2 Communication Protocols	16
4.5.2 Software Side	16
4.5.2.1 HTTP Connection Protocol	16
4.5.2.2 Client-Side Technologies	16
4.5.2.1.1 React	16
4.5.2.1.2 JavaScript	17
4.5.2.1.3 Tailwind CSS	17
4.5.2.1.4 Chart.js	17
4.5.2.3 Server-Side Technologies	17
4.5.2.2.1 Python	17
4.5.2.2.2 Firebase	18

4.5.2.2.3 Node.js	18
4.6 Architecture Diagram	18
4.6.1 Software Architecture	18
4.6.2 Hardware Architecture	19
5. Work Artifacts	19
5.1 Requirements	19
5.1.1 Functional Requirements	19
5.1.2 Non-Functional Requirements	20
5.2 Use Case Diagram	22
5.3 Activity Diagrams	23
5.3.1 Login	23
5.3.2 View Data	24
5.3.3 Update Data	24
5.3.4 Generate Report	25
5.3.5 Update Calendar	25
5.3.6 Check Alerts	26
5.4 Website Screens	27
6. Expected Achievements	28
6.1 Challenges	29
6.2 Success Criteria	29
6.3 Evaluation	30
7. Testing Process	30
7.1 Testing Scope	30
7.2 Objectives	30
7.3 Testing Approach	31
7.4 Test Cases	31
8. Appendices	35
8.1 Model Placement	35
8.1.1 Soil and Air Humidity Sensors	35
8.1.2 M5Stick C Plus 2 Controller	35
8.1.3 Camera	36
References	37

Abstract

In recent years, the Internet of Things (IoT) has emerged as a transformative force in modern agriculture, revolutionizing data collection, analysis, and utilization across orchards and research environments. This project addresses a specific challenge avocado growers face: monitoring various environmental and plant health parameters to detect diseases early. The main obstacle lies in lacking a comprehensive system to effectively gather, analyze, and transmit live data from diverse orchard sensors.

To address this challenge, we will develop an IoT-based sensor system aimed at detecting diseases in avocado trees. This solution establishes a cohesive platform for collecting, monitoring, and analyzing data, integrating a network of sensors that measure temperature, humidity, soil moisture, and leaf conditions within a cloud-connected environment. By centralizing and visualizing real-time sensor data, the system allows growers and researchers to monitor changes, swiftly recognize disease indicators, and obtain more profound insights into plant health and orchard conditions.

A key contribution of this project lies in its capacity for remote communication and data-driven decision-making, serving as a proof of concept for an integrated IoT framework in precision agriculture. Collaboration between local farmers and agricultural research institutions underpins this initiative, showcasing the potential for cooperative efforts in enhancing sustainable farming practices. Throughout the development process, we embraced Agile methodologies, ensuring flexibility and responsiveness to evolving requirements.

As one of the first targeted IoT applications for avocado disease detection within our institution, this project paves the way for further advancements in smart farming. Our work demonstrates the potential of inter-institutional partnerships in fostering innovation, setting a foundation for future developments in IoT-enabled agriculture, and reinforcing the critical role of data-driven strategies in modern crop management.

1. Introduction

Avocados have surged in global popularity due to their distinctive flavor and nutritional benefits, driving substantial growth in production across various regions. According to recent data [13], Mexico leads the world in avocado output with an impressive **2.53 million** tons, followed by Colombia and Peru. Notably, Israel also ranks among the top ten producers, contributing approximately **189.67 thousand** tons annually [13]. This significant share underscores the fruit's strategic importance in Israeli agriculture, where high-quality yields and efficient farming practices are paramount. However, as global demand for avocados intensifies, so do the challenges associated with diseases that can drastically impact crop health and productivity.

Avocado trees are highly susceptible to a variety of diseases, such as Armillaria Root Rot, Verticillium Wilt, and Phytophthora Canker, which can lead to significant yield losses and reduced fruit quality [14]. These diseases often exhibit overlapping symptoms or remain undetected until the tree's health deteriorates severely, complicating treatment efforts. Additionally, other factors such as nutritional deficiencies (e.g., boron and zinc), pest damage (mites, thrips, and the Melolonthidae complex), and abiotic stresses (sunburn,

hailstone damage, and overripe fruits) further contribute to losses during preharvest and harvest stages. also, there are some issues in packinghouses, These challenges collectively lead to an average economic loss of \$80.29 USD per ton of produced fruit, with rejection rates of 5.78% on farms and 5.68% in packinghouses [1].

Traditional detection and management methods, which rely heavily on manual inspection, are time-consuming, error-prone, and often inadequate for early-stage identification of these issues.

To address these challenges, This project presents an IoT-based sensor system integrated with machine learning and image processing to enhance early disease detection in avocado farming. By collecting real-time environmental data and analyzing images of affected plant parts, the system provides accurate and timely insights. Machine learning models classify diseases based on patterns in the data, improving precision and reliability. This innovative approach promotes sustainable farming, reduces chemical use, increases productivity, and strengthens Israel's position in the global avocado market, showcasing the transformative potential of precision agriculture.

2. Literature Review

2.1. Wireless Sensor Network (WSN) in Agriculture

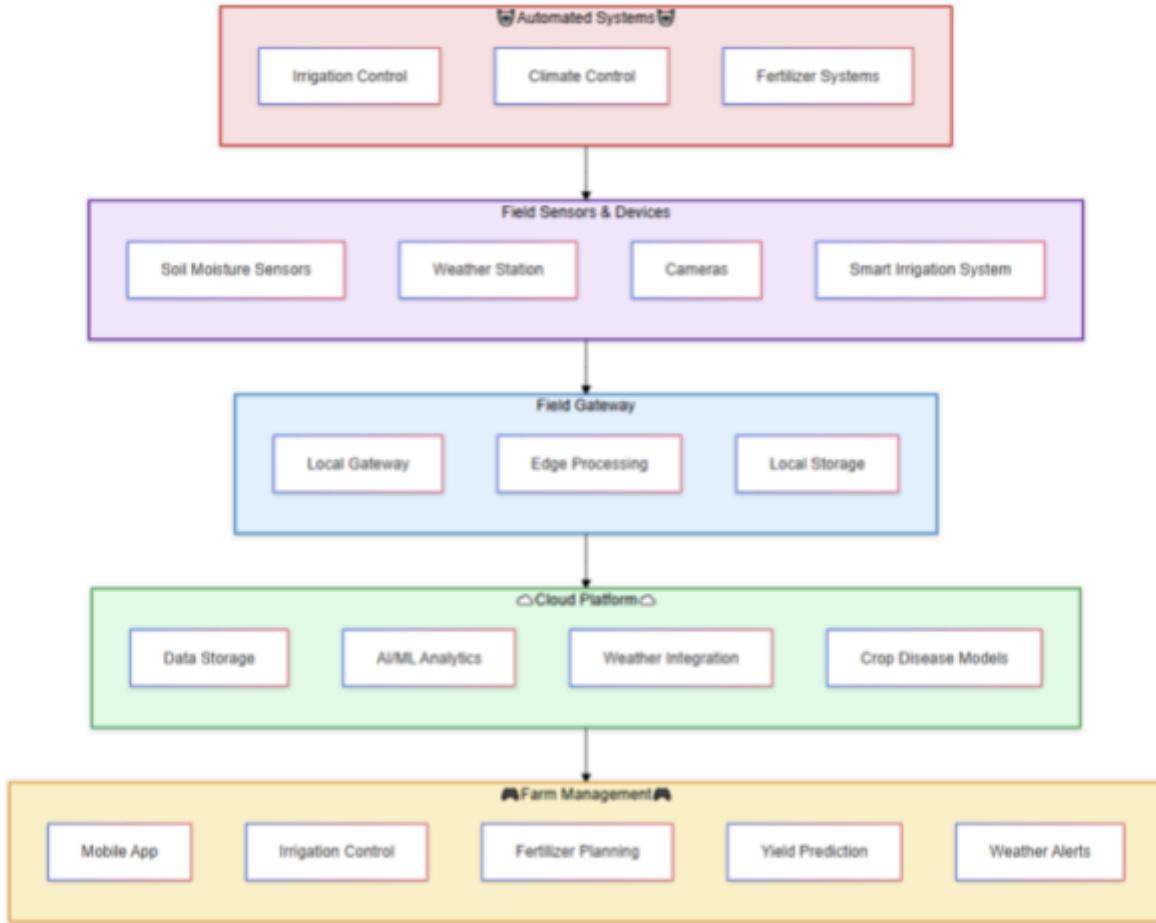
The integration of advanced technologies into agriculture has been a focus recently, aiming to enhance efficiency, sustainability, and productivity.

WSNs have emerged as a cornerstone for precision agriculture due to their adaptability, low cost, and energy efficiency. These networks facilitate real-time monitoring of environmental parameters like soil moisture, temperature, humidity, and pH, which are critical for optimizing agricultural practices. Studies have demonstrated the effectiveness of WSNs in monitoring agricultural fields through nodes powered by solar energy, which ensures uninterrupted communication and data transmission. For instance, the use of solar-powered routing protocols has proven to be energy-efficient and reliable, particularly in environments with limited infrastructure. By integrating advanced communication technologies, energy-efficient systems, and automation, smart agriculture can address pressing challenges such as resource scarcity and environmental sustainability [2].

2.2. IOT in Agriculture

In the field of agriculture, low-power wireless sensor networks (WSNs) and IoT platforms play a critical role in enabling real-time monitoring and management of environmental and crop-related factors. Programmable, open-source devices are often preferred for their flexibility in customizing sensor behavior, integrating new peripherals, and ensuring compatibility with the deployment environment. However, agricultural IoT deployments face unique challenges, such as interference caused by crop movement, environmental factors like humidity and temperature fluctuations, and power constraints in battery-operated nodes. Wireless communication protocols, such as Bluetooth, ZigBee, and LoRaWAN, are commonly used in agricultural IoT systems, each offering trade-offs in range, power consumption, and data transfer rates. Additionally, high humidity and extreme temperatures significantly impact communication reliability and sensor performance, highlighting the need

for durable, low-power hardware and efficient communication protocols to ensure long-term stability and accurate data collection in harsh agricultural environments. These insights guide the design of robust IoT systems tailored for precision agriculture, such as disease detection in avocado trees [3].



(Figure 1. IoT Architecture)

2.3 Precision Agriculture Using IoT

2.3.1 Sensors

2.3.1.1 Air Humidity Sensor

The integration of intelligent humidity sensors in Wireless Sensor Networks (WSNs) offers a transformative approach to precision agriculture by optimizing data handling and energy efficiency. Traditional humidity sensors generate excessive data, increasing power consumption and processing demands. The intelligent sensor model, as proposed, integrates a conventional sensor with an embedded processor to process data using advanced algorithms. This approach reduces data collection by up to 50% and minimizes post-processing latency, thereby enhancing power efficiency and overall system performance [4]. Beyond improving operational efficiency, these sensors play a crucial role

in disease detection by monitoring environmental conditions that influence pathogen growth. For example, they detect variations in soil and atmospheric moisture levels that favor fungal or bacterial diseases, enabling early warnings and localized control measures. By integrating sensor data with disease prediction models, farmers can proactively mitigate risks, optimize pesticide usage, and prevent outbreaks, making intelligent humidity sensors indispensable for sustainable farming and plant health management [4].

2.3.1.2 Soil Moisture Sensor

Soil moisture sensors play a vital role in modern agriculture by enabling efficient water management and supporting precision irrigation, ultimately conserving water and improving crop yields. These sensors provide real-time data about soil moisture levels, ensuring optimal irrigation scheduling and reducing overwatering, which can lead to plant diseases caused by excessive soil moisture. Various types of sensors are available, each with unique advantages. Besides improving irrigation practices, these sensors contribute to disease management by preventing conditions favorable for pathogen growth, such as excessive humidity in the root zone. Early detection of moisture imbalances helps in identifying potential disease outbreaks and maintaining soil health, demonstrating the sensors' dual role in enhancing productivity and mitigating risks [5].

2.3.1.3 Cameras for Early Disease Detection

There are several diseases the avocado plant can suffer from, such as Phytophthora root rot (Prr), and laurel wilt (Lw).

Utilizing cameras and classification methods, the system can do early-stage detection using a Machine Learning process.

Current detection methods, such as visual scouting, are labor-intensive, expensive, and prone to inaccuracies, especially during early, asymptomatic stages. Alternatives like microarray and DNA-based methods are accurate but costly and slow. Automated techniques using multispectral imaging and artificial intelligence offer a promising solution by leveraging spectral reflectance data to differentiate between healthy and stressed plants. These methods have shown success in other crops, prompting their application in this study to develop a low-cost, efficient detection system for avocado diseases and nutrient deficiencies [6].

2.4. Image Analysis in Agriculture

Real-time image analysis using lightweight convolutional neural networks (CNNs), such as ShuffleNet V2, has become a transformative tool in precision agriculture. This approach leverages computational efficiency to analyze high-resolution images of crops, enabling farmers to monitor spatial changes in plant health and growth without requiring extensive computational resources. By extracting meaningful features from images, ShuffleNet V2 facilitates applications such as detecting crop diseases, estimating yield potential, and monitoring overall plant health [7].

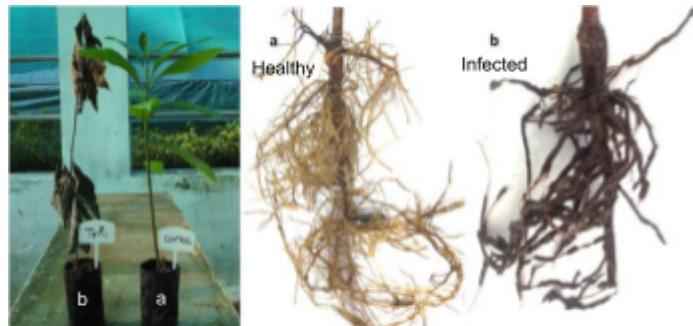
Integrating ShuffleNet V2 into agricultural image analysis has advanced tasks such as crop classification, weed detection, and disease identification. Its ability to process data in real-time, even on edge devices, makes it particularly suitable for field applications where computational resources are limited. Key advancements include using ShuffleNet V2 for feature extraction and classification, providing a balance between speed and accuracy [7]. Unlike traditional methods, which often require significant computational power, this model delivers reliable predictions while maintaining energy efficiency [7].

Despite its advantages, challenges remain in agricultural image analysis, including variations in environmental conditions, limited labeled datasets, and the need for robust models to handle diverse scenarios. Addressing these issues involves leveraging techniques like transfer learning, data augmentation, and hybrid models that combine ShuffleNet V2 with other frameworks to enhance robustness and accuracy. Continued innovation in this space positions lightweight CNNs like ShuffleNet V2 as essential tools for driving sustainability and productivity in modern agriculture [7].

2.5 Avocado Diseases

2.5.1 Root rot

Phytophthora root rot (PRR), is a critical issue in avocado cultivation, leading to root damage, reduced water and nutrient uptake, canopy decline, and tree mortality. The fungus thrives in poorly drained soils and has been reported globally, though its prevalence varies by region [8,9]. Traditional visual assessment of PRR severity is limited by assessor expertise and environmental factors, prompting the development of digital methods. Image analysis using RGB photography and satellite imagery has shown strong correlations between canopy porosity and disease severity, offering more accurate and scalable assessment tools [8].

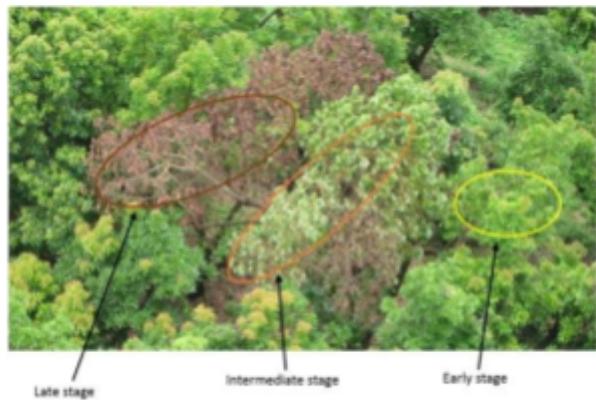


(Figure 2. healthy root vs infected root [12])

2.5.2 laurel wilt

Laurel Wilt (LW) is a lethal disease caused by the fungus *Raffaelea lauricola*, spread by the redbay ambrosia beetle. It disrupts tree vascular systems, causing rapid death. Since its U.S. discovery in 2002, LW has severely impacted avocado trees and other Lauraceae species, threatening Florida's \$35 million avocado industry and altering ecosystems [10,11].

Early detection through remote sensing techniques enables targeted control, while management includes tree removal,



(Figure 3. laurel wilt infected stages [24])

fungicides, and public awareness. However, challenges like beetle reproduction and the lack of biological controls persist. LW poses a global threat due to expanding beetle ranges and trade [10,11].

3. Research

3.1 Scheduled Meetings

Throughout the semester, we held several meetings with the students and Professor Uzzi Rozen from the Mechanical Engineering department. Regularly engaging with the mechanical engineering students was a key aspect of our work. During these meetings, we presented the progress made and the tasks completed within the given period. Additionally, these discussions helped clarify topics and address any uncertainties we had. After each meeting, we scheduled the next session and outlined the progress expected to be achieved by then.

3.1.1 Meetings Summary

3.1.1.1 Meeting Number 1 - 20.11.2024

In this meeting, we held a Zoom session with Keren, the manager of the agricultural farm. The discussion focused on highlighting the challenges faced in the agricultural field and exploring potential solutions that could benefit the industry. Keren also suggested several topics that we could take on as projects and work to develop further. We took the opportunity to ask her numerous agriculture-related questions and sought clarification on aspects that were unclear to us. This meeting served as a foundational step for our project, providing us with a starting point for our research and future development.

3.1.1.2 Meeting Number 2 - 27.11.2024

In this meeting, we conducted a Zoom session with the mechanical engineering students to review their proposed projects and evaluate their potential contributions. The students presented their group projects, explaining the objectives and expected outcomes of each. Our role was to assess whether their proposed projects aligned with our goals and requirements. Each of us engaged with different groups to gain a deeper understanding of their plans. Afterward, we reconvened and held a brainstorming session with our advisor to determine which group's project would be the most suitable for us to collaborate on.

3.1.1.3 Meeting Number 3 - 18.12.2024

In this meeting, we visited the college's FAB Lab, where the mechanical engineering students showcased their project presentations, outlining the deliverables for their final work. We reviewed all the presentations and identified several projects that piqued our interest. Following the meeting, we held a brainstorming session with our advisor and decided to collaborate with the group working on early disease detection for avocado trees. This group stood out as being more prepared and organized than the others, making them an ideal choice for efficient collaboration on the software aspect of the project.

3.1.1.4 Meeting Number 4 - 8.1.2025

In this meeting, we visited the college's FAB Lab and held a session with the mechanical engineering team and their professor, Uzzi Rozen. During the meeting,

we began with an introductory discussion where each member shared their progress on the project and the stage they had reached. We also raised several questions for which we sought clarification. The central topic of the meeting revolved around understanding the sensors that would be integrated into the prototype they were building. For example, we inquired about the type of camera that would be included. Additionally, we brought up several points regarding the sensors, highlighting potential bugs or issues that might arise in the future and affect our project. We summarized all the points discussed and concluded the meeting.

4. Engineering Process

4.1 Development Process

This section outlines the workflow and methodologies employed in our project. The primary objective is to develop a system that can effectively analyze data gathered from sensors and, most critically, determine whether an avocado tree is affected by a specific disease. Achieving this goal involves leveraging data processing techniques, backend frameworks, frontend technologies, and image processing. The approach emphasizes scalability, ensuring the system can handle large volumes of data, while also prioritizing flexibility and user-friendliness for an optimal user experience.

4.2 User-Centered Design Approach

Our project adopts a user-centered design (UCD) approach, prioritizing the needs and requirements of avocado growers and agricultural managers throughout the development process. This methodology ensures that the final system not only meets technical and functional specifications but also delivers practical, user-friendly solutions for detecting diseases in avocado trees, aligning closely with the real-world challenges faced by its users.

4.3 Workflow

The workflow for this project began with an initial discussion with Dr. Noami, during which the potential focus areas were explored, ultimately leading to the decision to pursue a project in the field of smart agriculture. This was followed by a comprehensive literature review, where articles and research papers were analyzed to gain a deeper understanding of the technologies and challenges associated with smart agriculture. Several meetings were held with farm & mechanical engineering teams to refine the project direction. In the first meeting with Keren, challenges in agriculture were discussed, and various project ideas were explored. Subsequent meetings included reviewing project proposals from mechanical engineering students and evaluating their feasibility, as well as selecting the early disease detection system for avocado trees after presentations at the FAB Lab. Another meeting at the FAB Lab allowed for the clarification of sensor requirements and the addressing of technical questions collaboratively. These discussions and insights led to the refinement of the final project idea, defining the key technical and engineering requirements needed for implementation. Currently, the project is in the documentation stage, which includes writing the project book with sections on the project idea, literature review, and engineering process. Additionally, we are planning Phase B, focusing on estimating the timeline and steps for the actual project development. (see Figure 4).



(Figure 4. Project Workflow Diagram)

4.4 Solution

4.4.1 Algorithm

Data Collection

The system begins with data collection, utilizing sensors distributed across the field. These sensors continuously monitor various environmental and tree-specific parameters, such as soil hydration levels, temperature, humidity, and tree images. The data is stored in the cloud for further processing.

Image Analysis Using Built-In Camera ML Model

Images captured by the sensors are processed directly by a ShuffleNet V2 machine-learning model embedded in the camera. This model is pre-trained to detect tree health issues and determines whether a tree is infected or healthy. Specifically, it identifies key diseases like root rot and laurel wilt, which can significantly impact tree health. The camera saves the

detection result (infected or not) in the Mosquito cloud, eliminating the need for additional image processing in the system.

Data Integration and Analysis

Once the detection results are saved in the Mosquito cloud, the system retrieves them along with other environmental data from the field sensors. This includes soil hydration levels and atmospheric conditions. By integrating the detection results from the camera with environmental data, the system conducts a detailed analysis to provide a holistic understanding of the field's overall condition. This integrated approach ensures the detection of potential issues that may not be apparent from the images alone.

Generating Actionable Insights

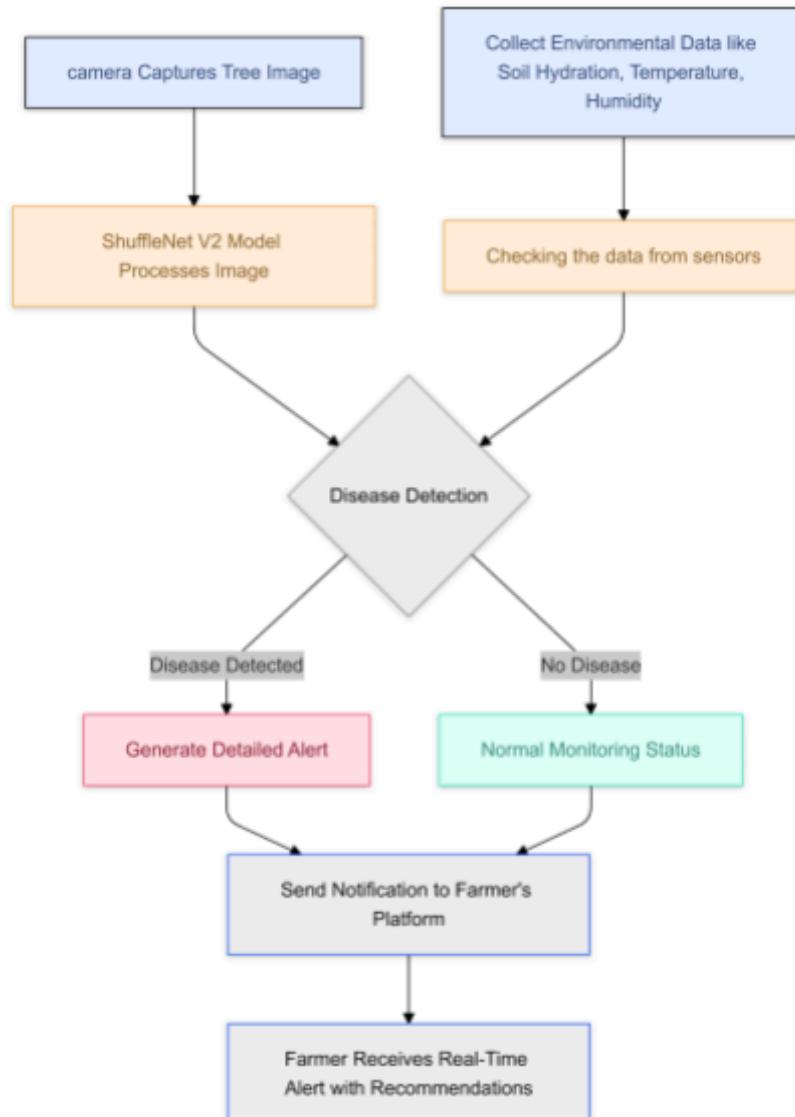
The results of the analysis are compiled into actionable insights. The system generates alerts and sends notifications directly to the farmer's profile through our platform. These alerts include a detailed report on the following:

- Whether trees require attention
- Potential issues identified
- Recommendations for addressing them

For example, if low soil hydration levels are detected, the system may recommend irrigation. If the detection results from the ShuffleNet V2 indicate disease presence, specific treatment actions are suggested.

Real-Time Updates

Our algorithm ensures timely updates by conducting multiple checks during the day and night. This schedule allows the system to provide farmers with up-to-date information, empowering them to make informed decisions to maintain the health and productivity of their fields. By leveraging the camera's built-in ML capabilities and integrating them with environmental data, the system provides a robust solution for proactive field management.



(Figure 5. Diseases Detection Algorithm)

4.4.2 Engineering Design

Our proposed solution is an advanced software system connected to a cloud server, designed to retrieve and process information collected by sensors strategically placed in the field near avocado trees. The model is divided into two main parts (See *Figure 6* and *Figure 7*). The first part, the bracket model, incorporates the M5StackStick C Plus2, the YL69 ground moisture unit, and the ENV IV sensor. This component is positioned directly beneath the tree to optimize data collection. The second part, the camera, is placed a few steps away from the tree to capture a wider view, ensuring a better angle for capturing the entire tree and monitoring its structure and surroundings. Both parts are connected via Wi-Fi, with the M5StackStick C Plus2 transmitting collected data in real-time to the Mosquitto cloud

using a JSON format and a communication protocol determined by mechanical engineering students, most likely MQTT.

The system decodes the sensor data and presents it to users in an intuitive and organized format, ensuring ease of access and interpretation. Additionally, the camera sensor itself performs disease detection directly using onboard processing capabilities. This eliminates the need for external deep-learning algorithms or further image processing on separate devices. The camera's integrated intelligence identifies diseases affecting individual trees, reducing the need for farmers to physically inspect their crops. By automating this critical aspect of plant care, the system significantly enhances efficiency and reduces response times to potential threats.

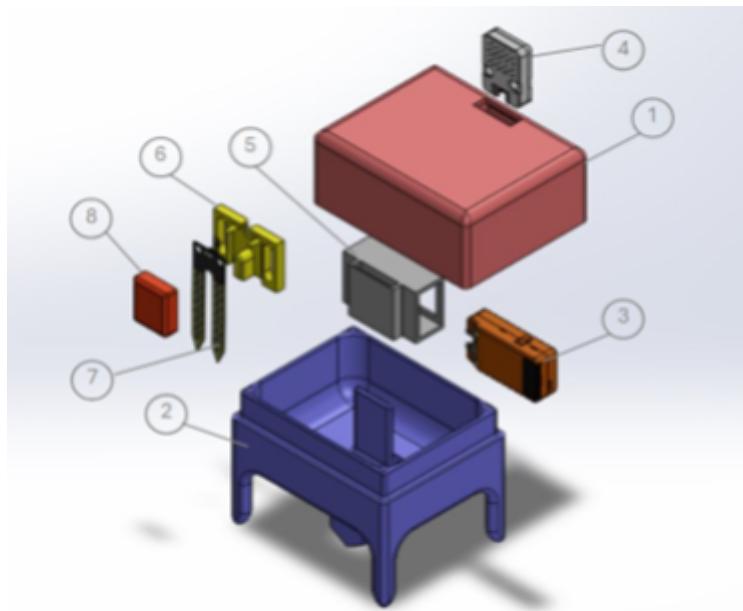
Beyond disease detection, the system functions as a comprehensive agricultural management tool. It includes features such as:

- **Reminders for Key Agricultural Tasks:** Schedule and track events like irrigation, fertilization, and pest control.
- **Harvest Planning:** Estimates optimal fruit-picking times based on visual and environmental data.

This integrated solution delivers a full suite of tools tailored to the farmer's needs, providing a seamless and efficient way to manage and optimize avocado cultivation. By minimizing manual intervention and offering actionable insights, the system empowers farmers to focus on growing healthy, productive crops with greater ease and confidence.



(Figure 6. Camera Sensor)



(Figure 7. Sensor's Bracket Model)

Part number	Description	Quantity
1	Front cover	1
2	Bottom cover	1
3	M5StackStick C Plus2	1
4	ENV IV – Sensor unit	1
5	Controller chassis	1
6	Back cover chassis for ground monitor unit	1
7	YL69 ground moisture unit	1
8	Front cover for ground monitor unit chassis	1

(Figure 8. Sensor's Table Description)

4.5 Technologies Review

4.5.1 Hardware Side

4.5.1.1 Technical Details

- **Custom code for M5Stack to manage sensors and actuators**
designed to integrate sensors & actuators into the M5Stack device, enabling real-time monitoring and control for avocado tree health.
- **MQTTX for testing MQTT communication**
MQTTX is a powerful cross-platform MQTT desktop client designed for testing and debugging MQTT communication. It is particularly useful for projects like ours, where IoT devices (e.g., M5Stack) transmit sensor data and receive control commands via MQTT.

- **Home Assistant for smart data display**

Home Assistant is an open-source platform designed to control and automate smart home devices. It is highly extensible and can also be used to display and manage data collected from IoT systems. By integrating with our IoT sensors, Home Assistant provides a centralized, user-friendly dashboard for real-time monitoring and control.

4.5.1.2 Communication Protocols

- **MQTT**

MQTT is a lightweight, publish/subscribe messaging protocol designed for low-bandwidth, high-latency, or unreliable networks. It is widely used in IoT applications for communication between devices, sensors, and servers due to its simplicity and efficiency.

- **HTTP**

HTTP is an application-layer protocol used for transferring data on the web. It is the foundation of data communication between clients and servers.

- **ZigBee/LoRaWAN**

Both Zigbee and LoRaWAN are wireless communication technologies designed for the Internet of Things. Both are used to connect devices, but they are optimized for different use cases based on their range, power consumption, and data transmission needs. Zigbee is a low-power, short-range wireless communication protocol designed for home automation, smart lighting, and similar IoT applications. It operates on the IEEE 802.15.4 standard. While LoRaWAN is a protocol built on LoRa (Long Range) modulation technology. It is designed for long-range, low-power IoT communication, often used in smart cities, agriculture, and industrial IoT.

Both options are available to us for communication, mechanical engineering students will decide which one to use based on the resources they have and depending on the environmental factors.

4.5.2 Software Side

4.5.2.1 HTTP Connection Protocol

HTTP is a unidirectional, stateless protocol that operates on top of reliable connection-oriented protocols like TCP or SCTP. TCP ensures data delivery through mechanisms like three-way handshakes and retransmission of lost packets. Each HTTP request opens a separate TCP connection between the client and server, which is terminated after the response is received [15].

4.5.2.2 Client-Side Technologies

4.5.2.1.1 React

React is a popular JavaScript library for building user interfaces [16], it allows the creation of reusable UI components that update efficiently based on changes in application data [17].

React makes it painless to create interactive UIs. Design simple views for each state in your application, and React will efficiently update and render just the right components when your data changes [18].

4.5.2.1.2 JavaScript

In our project, we will be using React with JavaScript. JavaScript is a versatile, object-oriented scripting language widely used for creating dynamic and interactive web content, enhancing user experience by enabling developers to modify webpage behavior and appearance based on user interactions. Running directly in the browser, it integrates seamlessly with HTML and CSS, allowing control over document structure, content, and styles. Its event-driven nature enables responsive actions to inputs like clicks and keystrokes, while support for asynchronous execution via callbacks and Promises facilitates efficient handling of tasks like data fetching [19].

4.5.2.1.3 Tailwind CSS

Tailwind CSS is a utility-first CSS framework that provides a collection of pre-defined classes to style elements directly in your HTML. Unlike traditional CSS frameworks, it focuses on building custom designs quickly by composing classes for spacing, typography, colors, and more, rather than relying on pre-designed components. Tailwind is highly customizable, enabling developers to extend or override its default styles using a configuration file [20].

4.5.2.1.4 Chart.js

We will introduce charts in our system to users, the charts will show a visual presentation of our data collected from the sensors, which clarifies many methodologies farmers and farm managers could benefit from. Chart.js is a free JavaScript library for making HTML-based charts. It is one of the simplest visualization libraries for JavaScript and comes with many chart types like bar, line, scatter, pie charts, and many more [21].

4.5.2.3 Server-Side Technologies

4.5.2.2.1 Python

Python is a versatile and powerful programming language known for its simplicity, readability, and vast ecosystem of libraries, making it an essential tool for modern development and data-driven applications. Python is strong in giving tools that can provide helpful implementation in machine learning and data manipulation. (e.g. Pandas, NumPy, Scikit-learn). These tools streamline the data preparation pipeline, allowing us to preprocess and explore data with ease.

We will be implementing the Convolutional Neural Networks (CNNs) model in our system to make image processing, Python is the language of choice due to frameworks such as TensorFlow, Keras, and PyTorch. These libraries provide pre-built functions and modules for designing, training, and deploying CNNs efficiently, enabling us to focus on innovation rather than low-level implementation.

4.5.2.2.2 Firebase

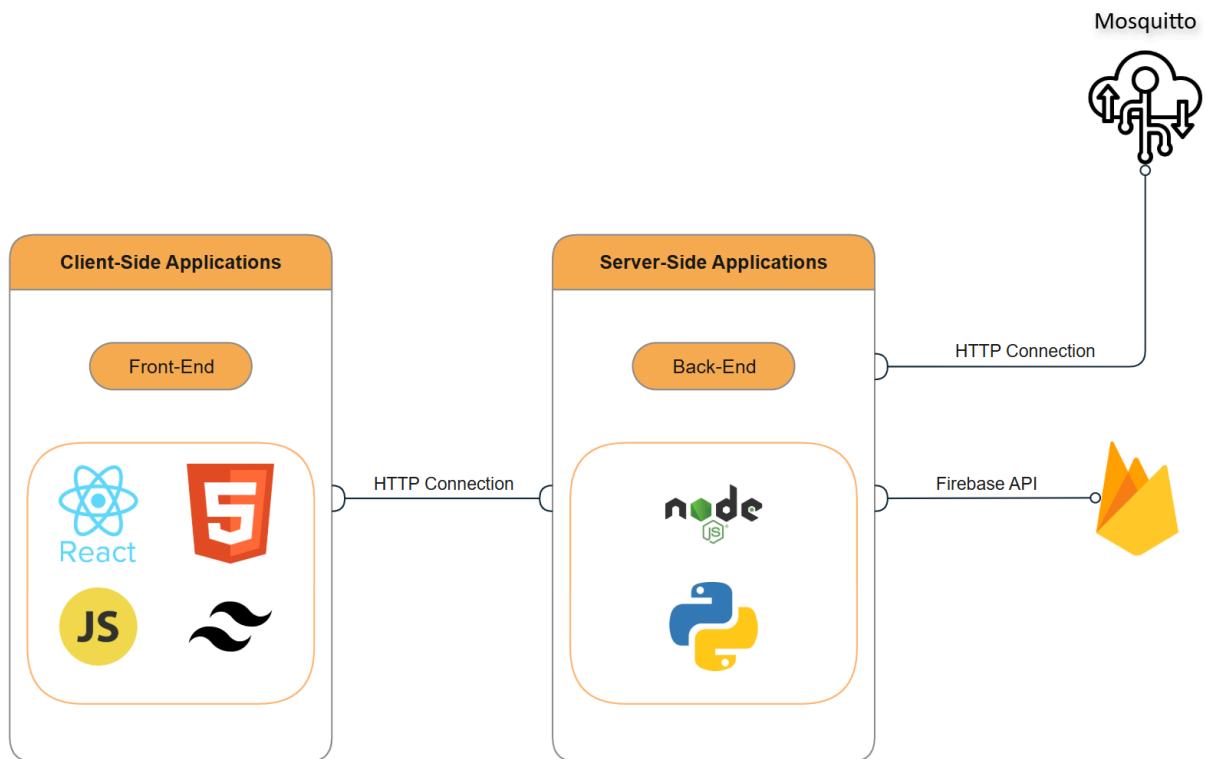
The Firebase Real-time Database allows you to build rich, collaborative applications by allowing secure access to the database. The Firebase Real-time Database is a NoSQL database from which we can store and sync the data between our users in real-time. Firebase Real-time Database is a solution that stores data in the cloud and offers an easy way to sync your data among various devices, and it is a cloud-hosted database. Data is stored as JSON and synchronized in real-time to every connected client [22].

4.5.2.2.3 Node.js

Node.js is a cross-platform, open-source JavaScript runtime environment that runs on the V8 engine, allowing JavaScript code to execute outside the browser on systems like Windows, macOS, Linux, and more. It enables developers to use JavaScript for server-side scripting and command-line tools, unifying web application development under a single language for both client and server. With its event-driven, asynchronous I/O architecture, Node.js is optimized for high throughput and scalability, making it ideal for real-time applications like browser games and communication programs, as well as handling numerous input/output operations efficiently [23].

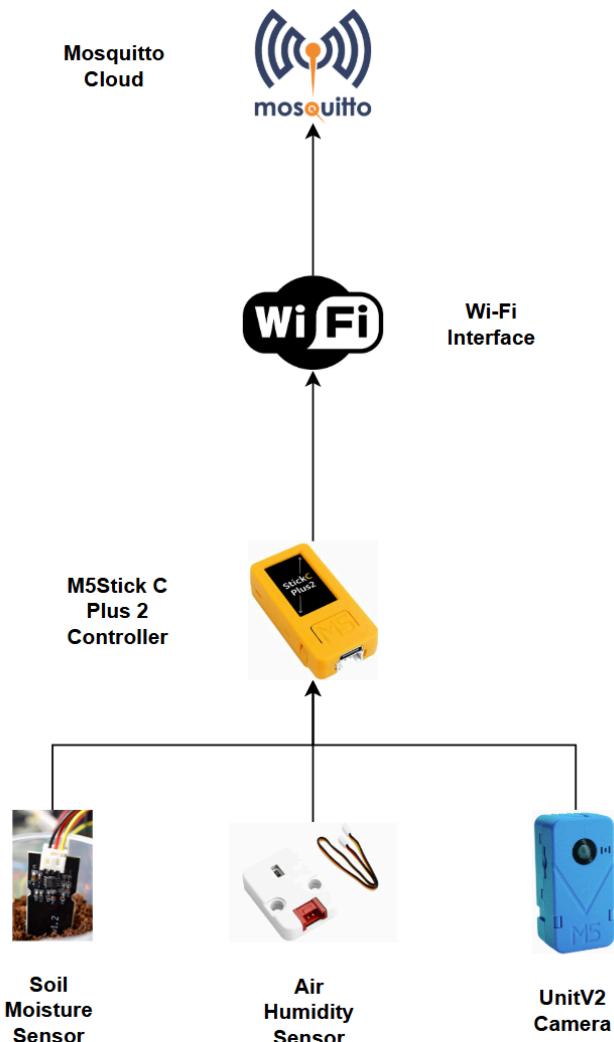
4.6 Architecture Diagram

4.6.1 Software Architecture



(Figure 9. Software Architecture)

4.6.2 Hardware Architecture



(Figure 10. Hardware Architecture)

5. Work Artifacts

5.1 Requirements

The system seeks to provide a scalable and efficient solution for farmers and agricultural managers to monitor and maintain avocado tree's health effectively. We will provide two tables of *Functional & Non-Functional* Requirements of our system.

5.1.1 Functional Requirements

No.	Requirement
1	The system should support user login & registration.
2	The system should support updating user settings.

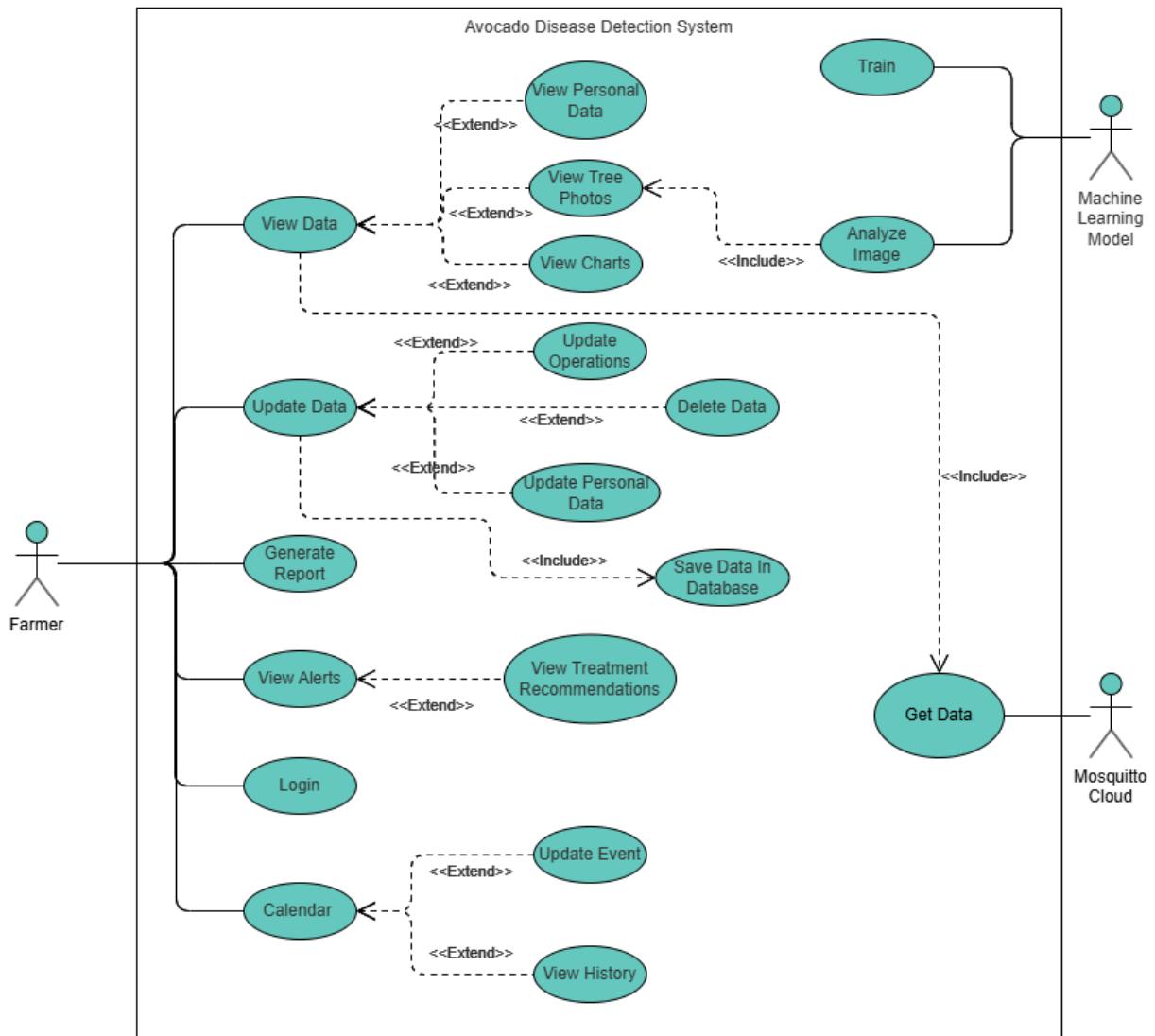
3	The system should allow users to access it via the WEB application.
4	The system shall allow multiple users to access data for the same farm.
5	The system shall provide role-based access control.
6	The system should show notifications for users when new data has been fetched.
7	The system should notify users when a disease is detected, specifying the type of disease and severity.
8	The system should provide recommendations for intervention, such as treatment options or preventive measures.
9	The system should identify key visual features, such as spots, discoloration, or patterns indicative of disease.
10	The system should classify images into predefined disease categories or "healthy" using machine learning models.
11	The system should identify unknown patterns that could indicate emerging diseases.
12	The system should store historical data for trend analysis and predictive modeling.
13	The system should provide a user-friendly dashboard to visualize environmental data, images, and analysis results.
14	The system should generate reports on the health of avocado trees and detected diseases.
15	The system should allow chart generation to visualize the collected data.
16	The system should show different types of charts on the collected data.

5.1.2 Non-Functional Requirements

No.	Requirement	Type
1	The image analysis for disease detection shall be completed within 30 seconds per image under normal conditions.	Performance
2	The system shall be able to handle data from more than 1 sensor simultaneously without performance degradation.	
3	The cloud infrastructure shall scale dynamically to handle increased workloads during peak data transmission periods.	Scalability
4	The system shall maintain an uptime of 99%, ensuring minimal downtime for data access and processing.	Availability

5	The system shall provide real-time data even in locations with limited network connectivity	
6	The system shall provide an intuitive and user-friendly interface, accessible via web browsers and mobile devices.	Usability
7	Users shall require no more than 20 minutes of training to navigate and operate the system efficiently.	
8	The system shall support multiple languages to cater to farmers in different regions.	
9	The system shall ensure accurate data transmission with no error rate	Reliability
10	The system shall automatically retry data transmission in case of communication failures of less than 20 sec.	
11	The system architecture shall be modular, enabling developers to update or replace individual components without affecting the entire system.	Maintainability
12	The system shall support custom configurations for sensor thresholds, task reminders, and notification preferences based on user requirements.	Flexibility
13	The system shall accommodate various types of sensors and communication protocols without major modifications.	
14	The interface shall be optimized for use on devices with varying screen sizes, including smartphones and tablets.	Accessibility

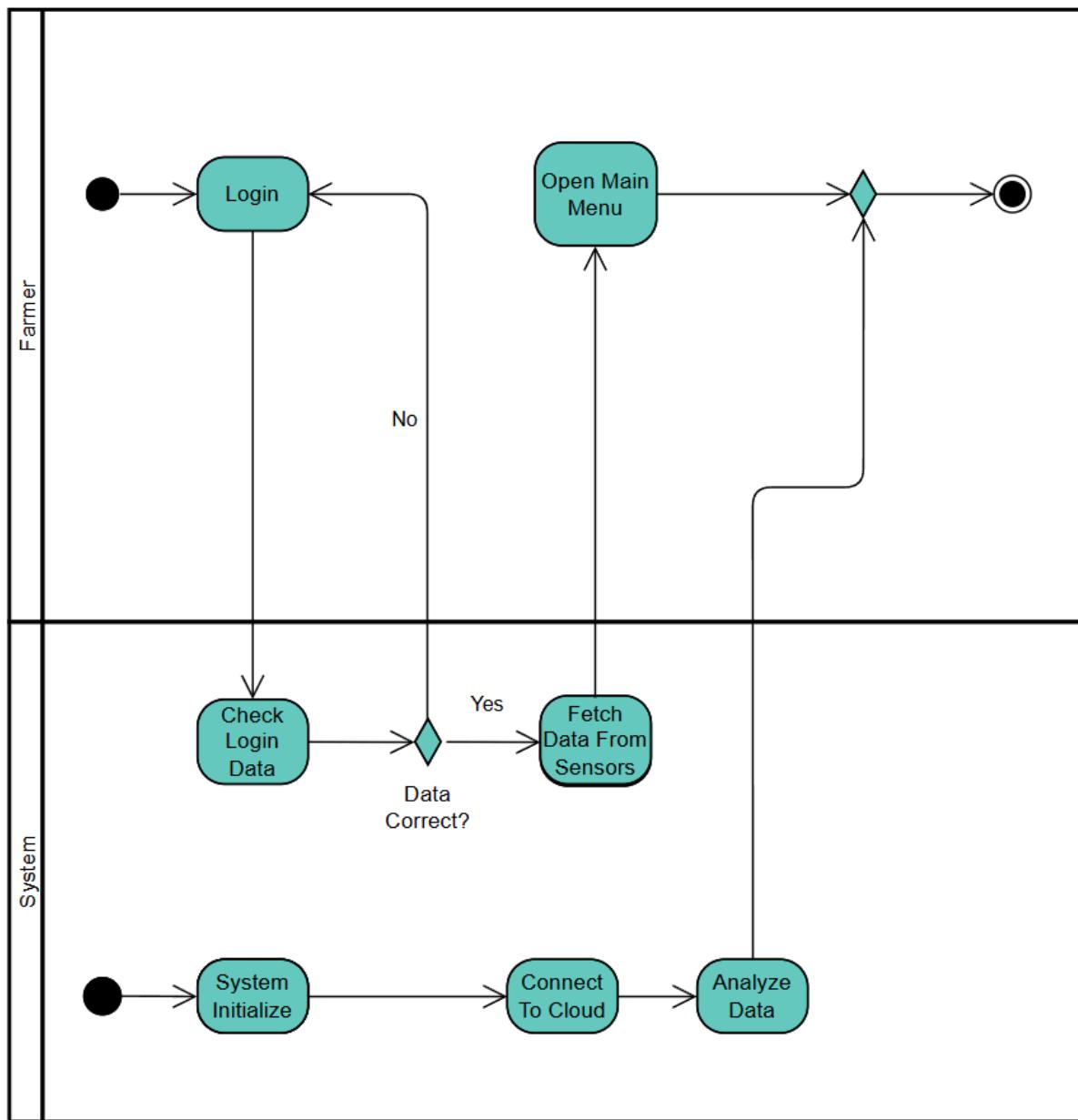
5.2 Use Case Diagram



(Figure 11. Use Case Diagram)

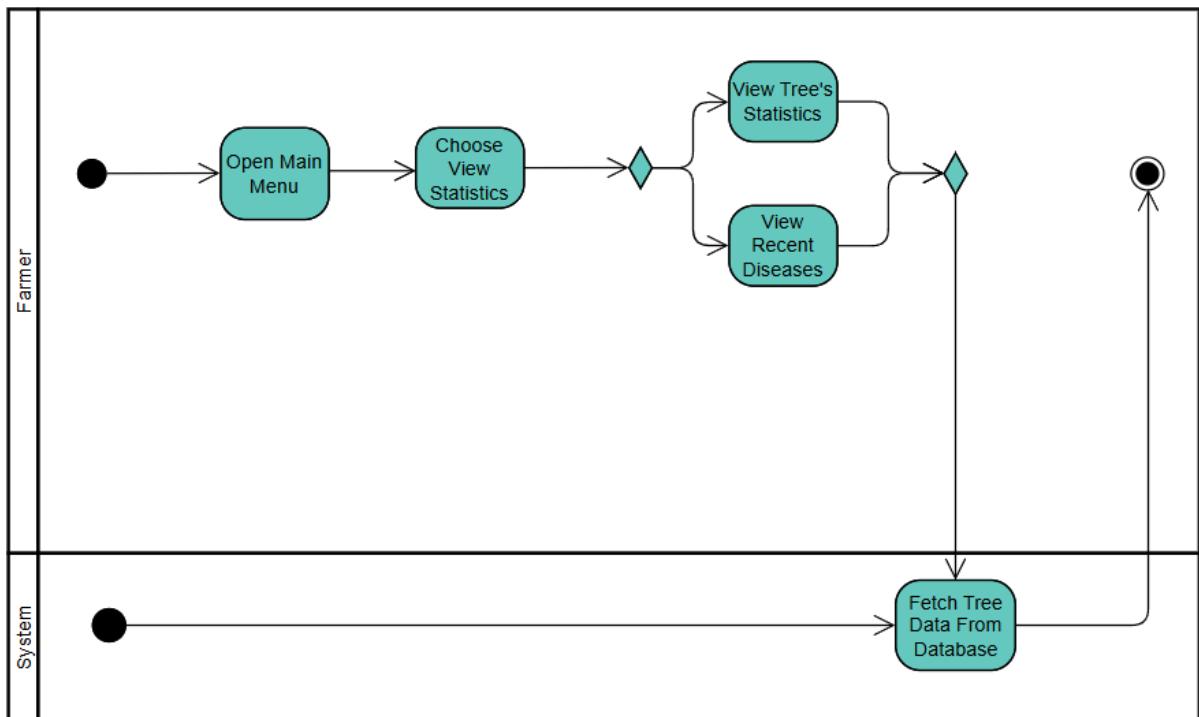
5.3 Activity Diagrams

5.3.1 Login



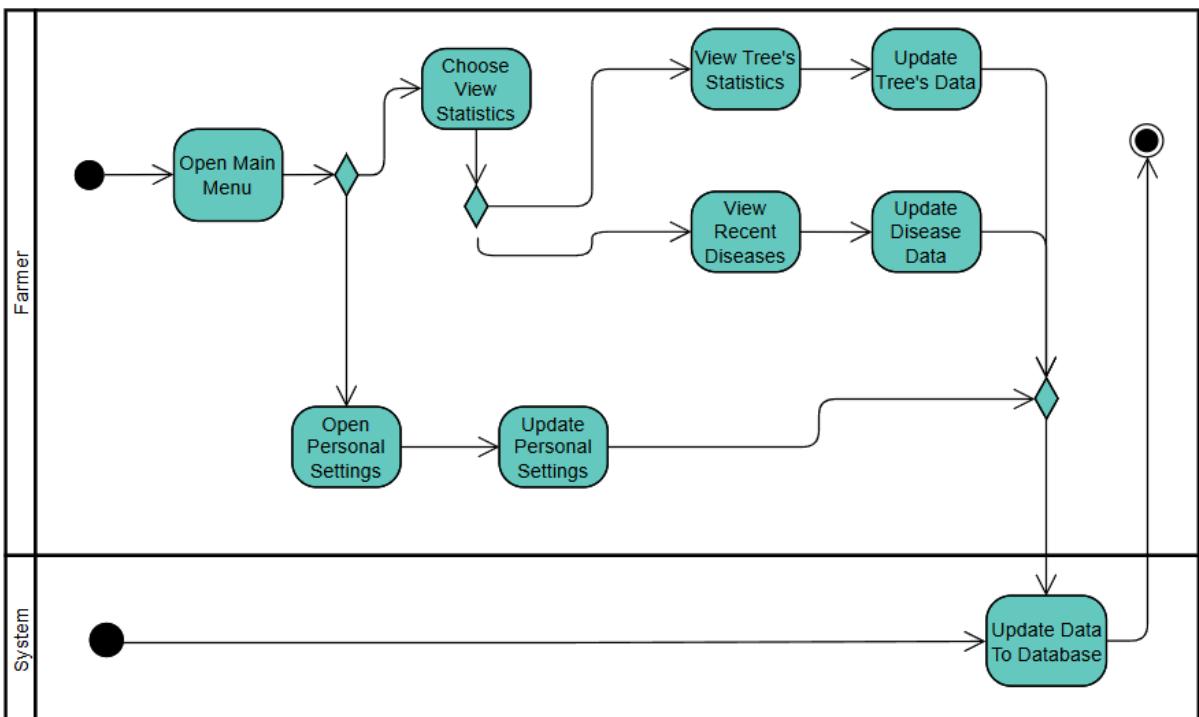
(Figure 12. Login Activity Diagram)

5.3.2 View Data



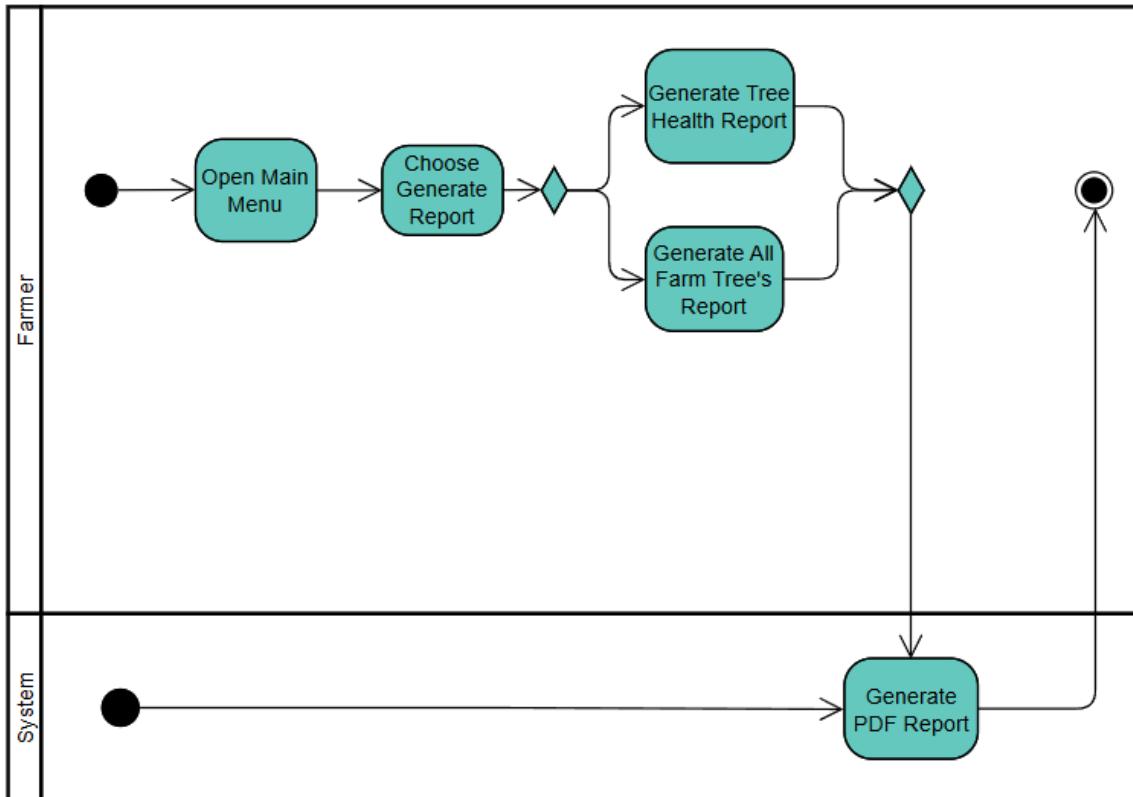
(Figure 13. View Data Activity Diagram)

5.3.3 Update Data



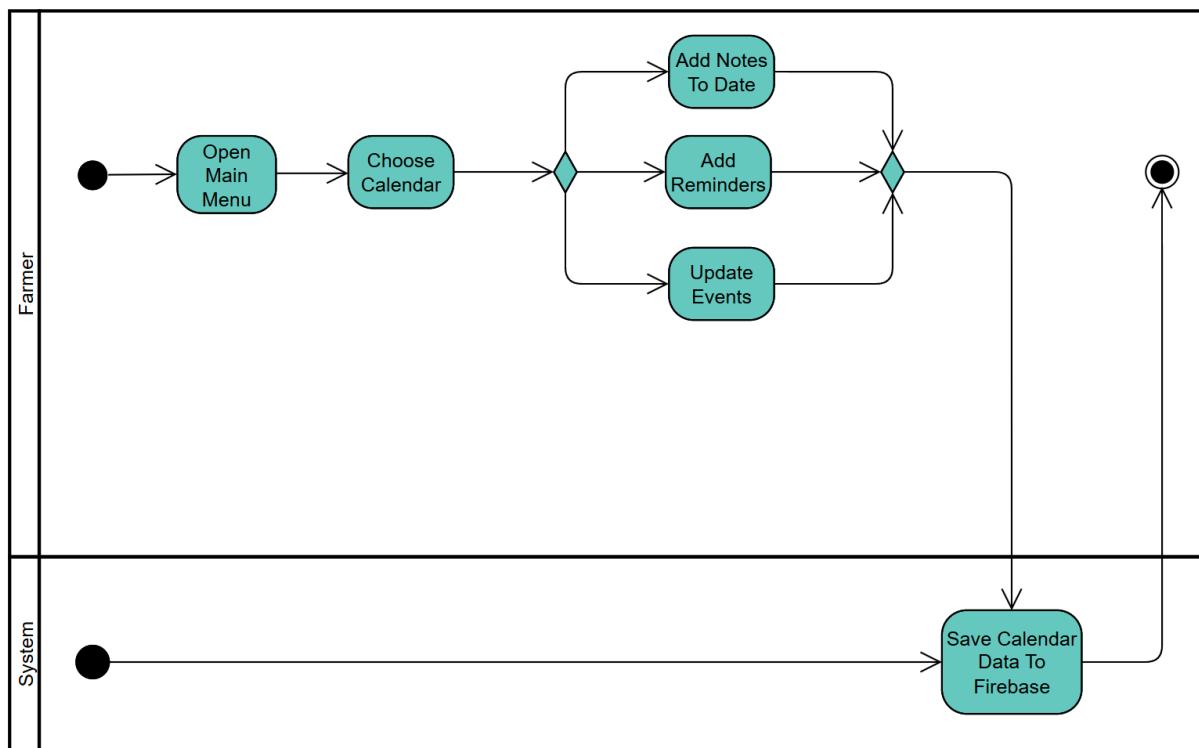
(Figure 14. Update Data Activity Diagram)

5.3.4 Generate Report



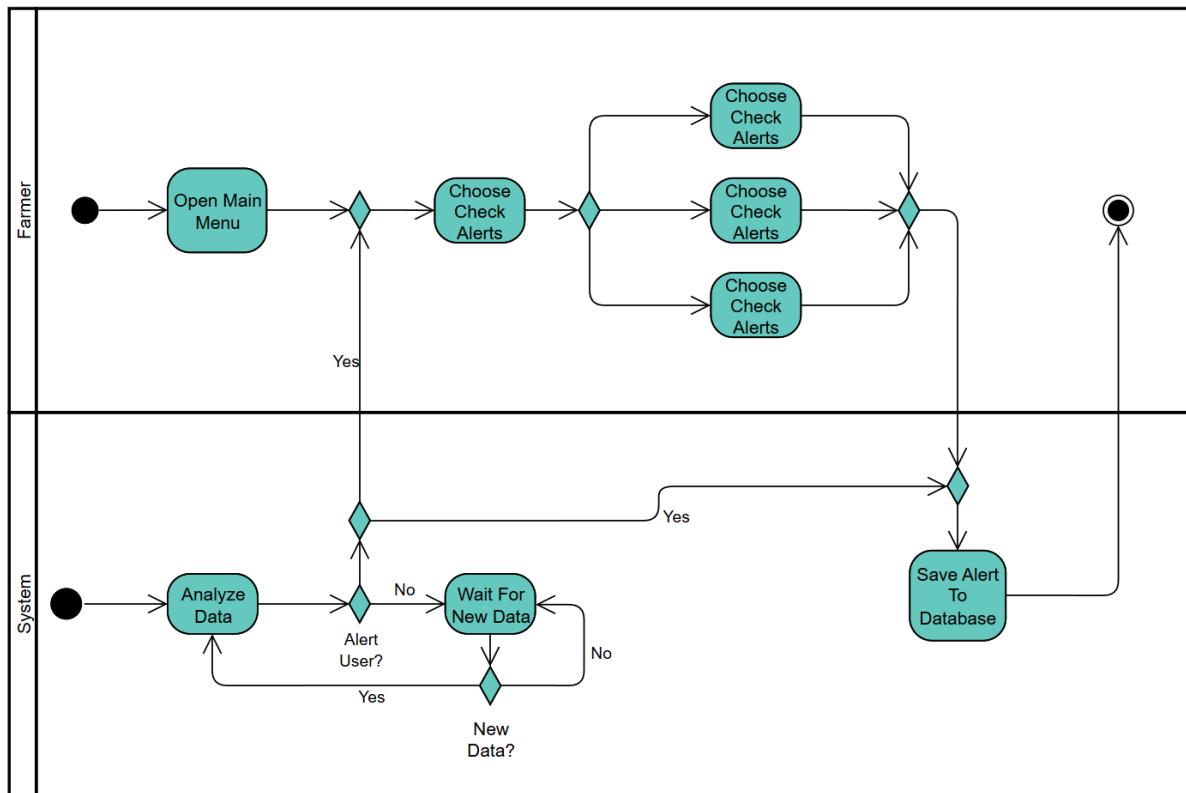
(Figure 15. Generate Report Activity Diagram)

5.3.5 Update Calendar



(Figure 16. Update Calendar Activity Diagram)

5.3.6 Check Alerts

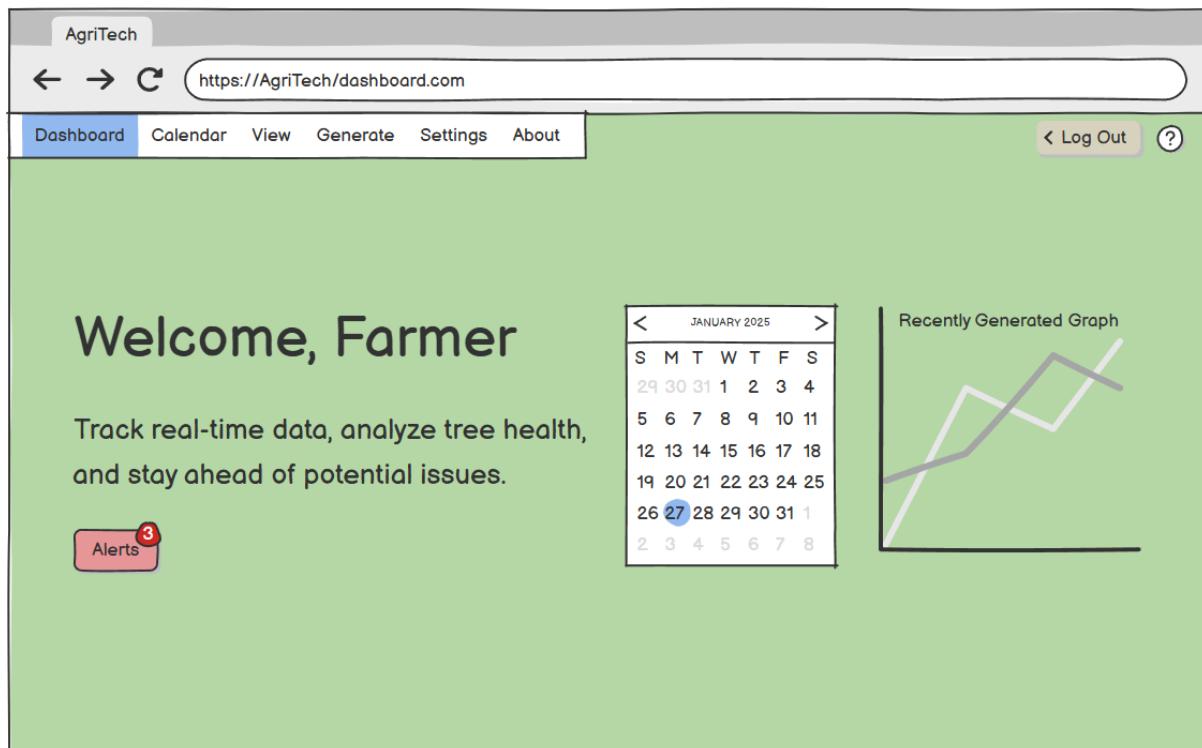


(Figure 17. Check Alerts Activity Diagram)

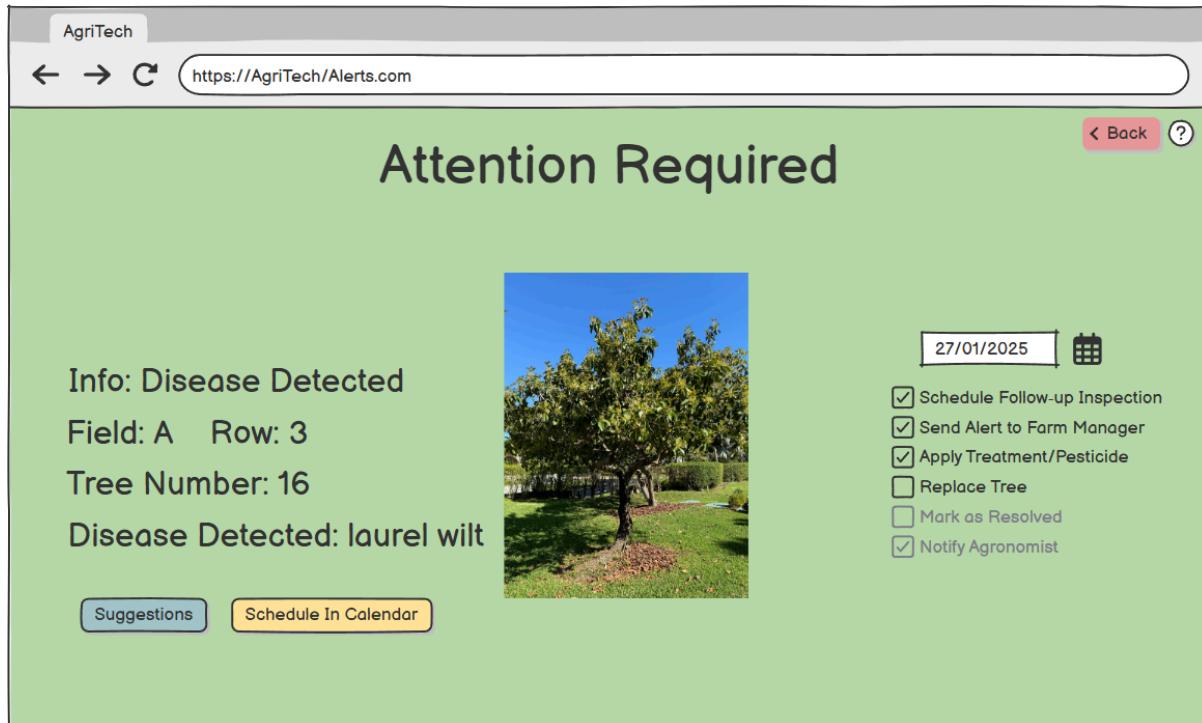
5.4 Website Screens



(Figure 18. Landing Screen Page)



(Figure 19. Dashboard Screen Page)



(Figure 20. Alert Screen Page)

6. Expected Achievements

1. Seamless API Integration:

Ensuring smooth communication between IoT devices, sensors, and the central system for real-time data collection and analysis.

2. Effective Use of Agile Methodology:

Facilitating iterative development, continuous feedback, and collaboration to meet project milestones efficiently.

3. Demonstration of Robust IoT and Data Monitoring Capabilities:

Showcasing a reliable system that consistently collects, processes, and visualizes real-time environmental and disease-related data from IoT sensors, ensuring accurate and uninterrupted monitoring.

4. User-Friendly Interface:

Developing an intuitive and visually appealing dashboard to present real-time environmental and disease insights.

5. Scalable Architecture:

Designing the system to handle increasing data from additional sensors or regions without performance degradation.

6. High Data Accuracy:

Achieving precise disease detection and environmental data readings through advanced machine learning and image processing algorithms.

6.1 Challenges

1. Environmental Interference:

The accuracy of sensor readings can be affected by external environmental conditions such as extreme weather, interference from nearby electronic devices, and hardware limitations. For example, heavy rain can cause soil moisture sensors to provide excessive readings, making it seem like the soil is more hydrated than it actually is.

2. Model Training and Accuracy:

The reliability of the disease detection model depends on the quality and quantity of labeled training data. Limited datasets and subtle symptom variations can lead to misclassification of diseases.

Examples:

- **Limited Data:** If only a few labeled images of diseased avocado trees are available, the model may fail to generalize well to new cases.
- **Similar Symptoms:** Diseases like **root rot** and **laurel wilt** may present similar early-stage symptoms (e.g., wilting leaves), making classification challenging.

3. Weak Latency Connection:

The system relies on real-time data transmission from IoT sensors and cameras to detect avocado tree diseases early. However, a weak internet connection or high network latency can significantly slow down data transfer, leading to delayed decision-making and potential disease spread before intervention. e.g. Many avocado orchards are located far from urban centers, where internet infrastructure is weak or unavailable.

4. User Training and Adoption:

Farmers and agricultural technicians may have varying levels of technical expertise, making it crucial to design an intuitive interface and provide adequate training.

A lack of **hands-on training sessions** before deployment may lead to confusion about system operation.

6.2 Success Criteria

1. High Disease Detection Accuracy:

Achieving a specified accuracy rate (e.g., $\geq 90\%$) in identifying avocado tree diseases using image processing and machine learning models.

2. Data Processing:

Ensuring the system can process and analyze environmental and image data with minimal latency (e.g., ≤ 45 seconds).

3. Seamless IoT Integration:

Reliable and consistent data collection from IoT sensors, with uptime $\geq 99\%$.

4. Actionable Insights Delivery:

Generating accurate and actionable recommendations for disease management and prevention.

5. Farmer Adoption and Usability:

Positive feedback from users (e.g., farmers, agricultural technicians) about the system's ease of use and effectiveness, with a target of achieving no less than 85% satisfaction rate in usability.

6. Data-Driven Decision Making:

Empowering stakeholders to make informed decisions based on system insights, demonstrated by real-world case studies or testimonials.

7. Reduction in Disease Spread:

Measurable reduction in the spread of avocado tree diseases in test deployments (e.g., a decrease in infection rates by $\geq 50\%$).

6.3 Evaluation

Our evaluation will focus on assessing the system's performance, usability, and impact through a combination of technical benchmarks and real-world testing. Key metrics will include disease detection accuracy, data processing latency, and system uptime, measured against predefined thresholds. Usability will be evaluated through stakeholder feedback, and testing the user interface for accessibility and intuitiveness. Field tests will validate the system's ability to operate reliably in diverse environmental conditions, while scalability and energy efficiency will be examined under increasing loads. Lastly, the system's effectiveness in reducing disease spread and improving avocado yield will be measured to demonstrate its practical impact, ensuring alignment with project goals and stakeholder expectations.

7. Testing Process

7.1 Testing Scope

The testing scope encompasses validating the system's IoT integration, machine learning accuracy, data processing, user interface usability, scalability, and effectiveness in detecting avocado tree diseases under various environmental conditions.

7.2 Objectives

- Ensure that the system is able to accurately detect diseases for avocado trees.
- Functional & Non-functional requirements are satisfied.
- Stability of the system over large amounts of data and processing.
- Identify errors and unexpected behaviors in the system.

7.3 Testing Approach

- **Unit Testing:**

Each individual component, including IoT sensors, data processing algorithms, and machine learning models, will be tested in isolation to ensure correctness.

- **Integration Testing:**

The interaction between IoT devices, data collection modules, and the central system will be tested to ensure seamless data flow and communication.

- **System Testing:**

End-to-end testing will be conducted to validate the system's overall functionality, including disease detection accuracy and real-time data visualization on the dashboard.

- **Performance Testing:**

The system will be tested under various environmental conditions and loads to assess latency, scalability, and energy efficiency.

- **Usability Testing:**

Farmers and agricultural managers/technicians will test the user interface to ensure accessibility and ease of use.

- **Field Testing:**

Real-world testing will be conducted in agricultural settings to evaluate the system's robustness and effectiveness in detecting and mitigating avocado tree diseases.

7.4 Test Cases

No.	TestID	Precondition	Expected Result	Description
1	SensorConfigured	IoT sensors are properly connected and configured	Sensor data is accurately collected and transmitted	Test the functionality of IoT sensors to ensure they capture and send environmental data correctly

2	TrainingModel	The machine learning model is trained and deployed	The model correctly identifies disease with accuracy	Validate the accuracy of the disease detection model using a labeled dataset
3	SystemNetworkConnectivity	The system has access to stable network connectivity	Data from sensors updates on the dashboard	Test data flow from IoT sensors to the dashboard.
4	UserLogin	Test user has access to the system	User interface displays data in a clear, user-friendly way	Verify the usability and accessibility of the dashboard for end-users
5	EnviromentalFactors	Sensors are placed in varying environmental conditions	System collects and processes data accurately in all conditions	Evaluate the robustness of sensors under diverse environmental factors
6	InconvenienceData	Data irregularity is introduced in the sensor readings	System detects and flags the irregularity	Test the system's ability to flag irregularities
7	SystemComponents	All system components are operational	System performs end-to-end disease detection and reporting	Conduct full system testing from data collection to actionable insights

8	HistoricalData	Historical data is stored in the system	System generates trends and predictions based on data	Test the analytics and predictive capabilities of the system using historical data
9	UserInputError	User provides incorrect or incomplete input on the dashboard/Login	System provides appropriate error messages or guidance	Validate error handling and feedback mechanisms in the user interface
10	PlatformCompatibility	Dashboard accessed on multiple devices or browsers	Dashboard functions consistently across platforms	Test the cross-platform compatibility of the user interface
11	DataLossTransmission	Data packet loss during transmission	System retransmits or handles missing data gracefully	Evaluate the system's ability to handle data loss during transmission
12	IncreaseUserAccessResponse	Increase in the number of users accessing the dashboard	System remains responsive and operational	Test the system's ability to handle high concurrent user traffic
13	Diseaseldentification	Real-world disease symptoms on avocado trees	System correctly detects and classifies diseases	Conduct field testing to verify disease detection in real-world scenarios

14	ChartGeneration	Generating a Chart based on the provided data from sensors	System correctly generates and showcases the chart to the user on the dashboard	Evaluate the system's ability to generate the correct chart with readable resolution
15	ReportGeneration	Generating a report on the specific avocado tree condition	system correctly generates a PDF file to the user based on his inputs	Validate if the PDF file has been generated correctly and with good alignment for charts and text

8. Appendices

8.1 Model Placement

8.1.1 Soil and Air Humidity Sensors

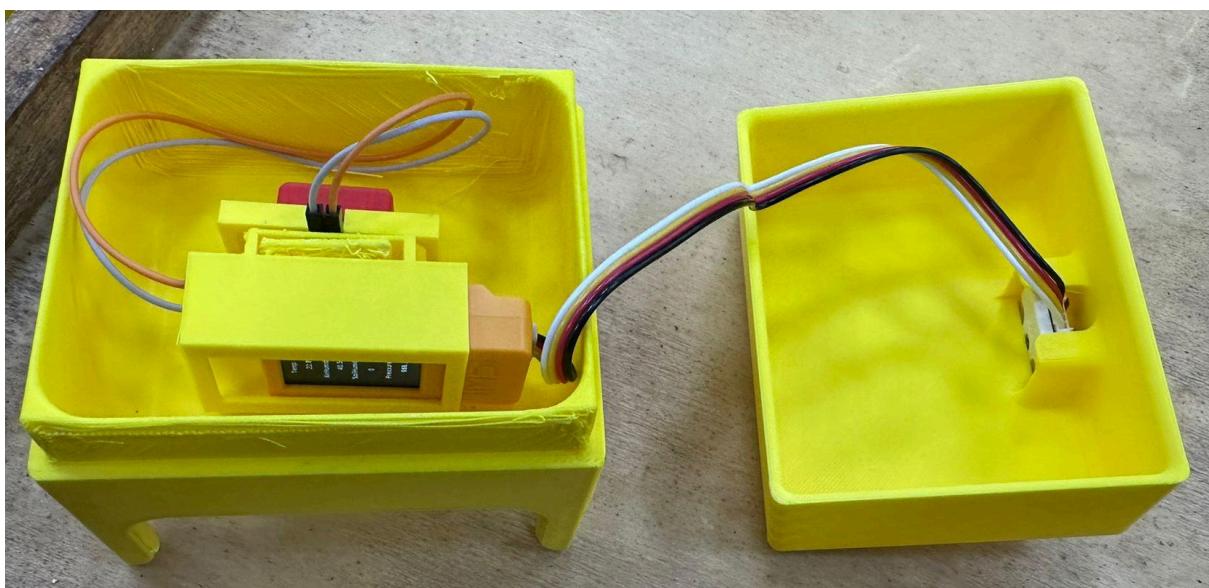
Enable continuous monitoring of critical environmental conditions.



(Figure 21. Sensor Model Bracket)

8.1.2 M5Stick C Plus 2 Controller

The M5Stick C Plus 2 controller acts as the central hub of the system. It integrates data from the sensors and the camera, processes it in real time, and facilitates visualization and communication. Its compact design and built-in Wi-Fi capabilities make it ideal for IoT applications in agriculture. see Figure 22 and Figure 23.



(Figure 22. M5Stack Controller)



(Figure 23. M5Stack Controller)

8.1.3 Camera

Captures leaf images and analyzes them to detect disease symptoms.



(Figure 24. Camera Placement)

References

- [1] Ramírez-Gil, J. G., López, J. H., & Henao-Rojas, J. C. (2019). Causes of Hass avocado fruit rejection in preharvest, harvest, and packinghouse: economic losses and associated variables. *Agronomy*, 10(1), 8.
- [2] Lakshmi, V. M., & Prabhu, T. (2024). In-depth Analysis and Investigation of Internet of Things and WSN for Precision Agriculture. *COMPUTER*, 24(11).
- [3] Tzounis, A., Katsoulas, N., Bartzanas, T., & Kittas, C. (2017). Internet of Things in agriculture, recent advances and future challenges. *Biosystems engineering*, 164, 31-48.
- [4] Shinghal, D. K., Noor, A., Srivastava, N., & Singh, R. (2011). Intelligent humidity sensor for-wireless sensor network agricultural application. *International Journal of Wireless & Mobile Networks (IJWMN)*, 3(1), 118-128.
- [5] Garg, A., Munoth, P., & Goyal, R. (2016, December). Application of soil moisture sensor in agriculture. In Proceedings of Internation Conference on Hydraulic (pp. 8-10).
- [6] Abdulridha, J., Ehsani, R., Abd-Elrahman, A., & Ampatzidis, Y. (2019). A remote sensing technique for detecting laurel wilt disease in avocado in presence of other biotic and abiotic stresses. *Computers and electronics in agriculture*, 156, 549-557.
- [7] Wang, C., Liu, B., Liu, L., Zhu, Y., Hou, J., Liu, P., & Li, X. (2021). A review of deep learning used in the hyperspectral image analysis for agriculture. *Artificial Intelligence Review*, 54(7), 5205-5253.
- [8] Salgadoe, A. S. A., Robson, A. J., Lamb, D. W., Dann, E. K., & Searle, C. (2018). Quantifying the severity of phytophthora root rot disease in avocado trees using image analysis. *Remote Sensing*, 10(2), 226.
- [9] Zentmyer, G. A. (1984). Avocado diseases. *International Journal of Pest Management*, 30(4), 388-400.
- [10] Olatinwo, R. O., Fraedrich, S. W., & Mayfield III, A. E. (2021). Laurel wilt: current and potential impacts and possibilities for prevention and management. *Forests*, 12(2), 181.
- [11] De Castro, A. I., Ehsani, R., Ploetz, R., Crane, J. H., & Abdulridha, J. (2015). Optimum spectral and geometric parameters for early detection of laurel wilt disease in avocado. *Remote Sensing of Environment*, 171, 33-44.

[12] Ayyandurai, M., Theradimani, M., Harish, S., Manonmani, K., Madhu, G. S., Raja, I. Y., ... & Pushpam, A. K. (2024). Bioprospecting of microbial agents and their metabolites as potential inhibitors of Phytophthora cinnamomi, the causal agent of avocado root rot. *Physiological and Molecular Plant Pathology*, 133, 102362.

[13]<https://www.statista.com/statistics/593211/global-avocado-production-by-country/>

[14] <https://greenarborists.com/five-common-diseases-of-avocado-trees/>

[15] [HTTP Connection Protocol](#)

[16] [React](#)

[17] [Why Use React?](#)

[18] [More React](#)

[19] [React & JavaScript](#)

[20] [Tailwind CSS](#)

[21] [ChartJS](#)

[22] [What is Firebase](#)

[23] [About NodeJS](#)

[24] <https://ucanr.edu/blogs/blogcore/postdetail.cfm?postnum=24850>