APPLIED INFORMATION TECHNOLOGY AND COMPUTER SCIENCE
e-ISSN: 2773-5141

AITCS

Vol. 6 No. 2 (2025) 170-189
https://publisher.uthm.edu.my/periodicals/index.php/aitcs

Smart Durian Farming: IoT Based Tree and Fruit Fall Monitoring System

Muhamad Haiqal Mahathir[1], Rosmamalmi Mat Nawi[1*]

[1] *Fakulti Sains Komputer dan Teknologi Maklumat, Universiti Tun Hussein Onn Malaysia, Parit Raja, Batu Pahat, 86400, MALAYSIA*

*Corresponding Author: rosmamalmi@uthm.edu.my
DOI: https://doi.org/10.30880/aitcs.2025.06.02.010

Article Info

Received: 13 June 2025
Accepted: 3 November 2025
Available online: 30 November 2025

Keywords

Web-Based System, IoT, Smart Farming, Machine Learning, Durian Cultivation

Abstract

The Smart Durian Farming project addresses key challenges in traditional durian cultivation by developing a complete Internet of Things (IoT)-based system enhanced with computer vision for intelligent orchard management. Manual monitoring is often labour-intensive, error-prone, and inefficient, resulting in delayed fruit collection, exposure to animal threats, and environmental losses. To solve this, the final version of the system integrates ESP32-CAM, tilt sensors, and vibration sensors to detect durian falls in real time. Images are processed using YOLOv5 with a custom-trained model via Roboflow, while the backend leverages Laravel, Firebase Realtime Database, and MySQL to log events, store data, and support seamless API communication. A responsive web-based dashboard enables farmers, admins, and managers to receive real-time alerts, view classified images, and manage harvest and inventory operations efficiently. Key outcomes of the completed system include accurate durian fall detection, minimized manual work, improved harvest tracking, and faster response to orchard events. The project ultimately enhances productivity and supports sustainable farming practices by reducing waste, optimizing labor, and laying the groundwork for future expansion into automated irrigation and predictive analytics.

1. Introduction

The Smart Durian Farming Project (IoT-D) addresses critical challenges in modern durian cultivation, where manual monitoring of large orchards proves inefficient and prone to oversight. As the scale of durian farms increases, these labor-intensive methods become impractical, leading to delayed fruit collection, reduced quality, and significant yield losses due to animal interference and environmental factors. This project leverages Internet of Things (IoT) technology and advanced computer vision to revolutionize orchard management [1]. The system employs tilt sensors, cameras, and a web-based application to provide real-time monitoring of fruit falls, and potential animal threats. Designed to optimize orchard management, the IoT-D system delivers accurate notifications, reduces false positives, and supports data-driven decisions for improved yields and reduced waste [2]. It also offers scalability for future integration with smart farming solutions like automated irrigation and climate control. The project involves developing key modules, including real-time fruit fall detection, object recognition, and a web-based interface for notifications and reporting. A developed prototype will be tested on a small-scale orchard. This approach contributes to efficient monitoring, improved harvesting, enhanced profitability, and flexible solution adaptable to various farm sizes. This project aims to transform durian farming and contribute to smarter, sustainable agriculture.

2. Literature Review

This section shows the literature review for this project. Section 2.1 covers the fundamental concept of IoT. Section 2.2 introduces the real time fruit fall detection, and section 2.3 describes the image classification. In section 2.4, four types of IoT based monitoring systems are compared with the proposed system. The comparison highlights the architecture, strength, and limitations of the existing system, to make a comparison between the existing system and the developed system in this project.

2.1 Fundamental concept of IoT

The Internet of Things (IoT) is a system that connects electronic devices, everyday objects, and even body elements using sensors, software, and other technologies to collect and transfer data over the internet through an automated process [3]. IoT is crucial in environmental monitoring, requiring the ability to perceive natural events autonomously [4]. IoT architecture consists of four layers which are the physical layer, which includes tools and gathering sensors information, the network layer, responsible for data packet routing between networks, the processing layer, which ensures reliable data flow and error management, and the application layer, which provides user interfaces to manage connected devices and perform various functions. Examples of IoT applications include smart home apps, industrial automation, and healthcare monitoring systems.

2.2 Real Time Fruit Fall Detection

The research investigates the detection of real-time fruit falls utilizing advanced agricultural technology. The study combines two detection methods: vision-based systems powered by deep learning algorithms such as YOLO or LedNet, and physical detection using tilt sensors [5]. The high-resolution cameras continuously monitor orchards, processing real-time imagery to identify fruits even in challenging conditions. Tilt sensors placed on protective nets detect impact events when fruits fall, complementing the visual system. The combined use of these data streams improves detection accuracy by cross-validating events visually and physically, effectively filtering out false positives. This technology enables immediate notifications for fruit collection, reducing losses from spoilage or animal interference. The innovative approach demonstrates how combining multiple sensing technologies can create more reliable and efficient agricultural monitoring systems, improving crop management and increasing harvest yield.

2.3 Image Classification with YOLOv5

Image classification involves identifying and categorizing the subject of an image into predefined groups, such as animals, fruits, or environmental elements [6]. This fundamental computer vision task supports various applications, including object detection, segmentation, and video analysis. YOLOv5, a widely used deep learning-based object detection model, enhances image classification by efficiently identifying and localizing objects within an image. Unlike traditional methods that rely on manual feature extraction or computationally expensive models, YOLOv5 utilizes convolutional neural networks (CNNs) to perform classification and localization in a single step. Its fast inference speed and lightweight architecture make it ideal for real-time applications. YOLOv5 is particularly effective for tasks such as monitoring fallen durians, detecting animal intrusions, and analyzing environmental changes in resource-constrained environments. With its support for various hardware platforms and robust tools for model training, YOLOv5 has become a versatile choice for developing machine learning solutions tailored to specific field applications.

2.4 Comparison with Existing System

Hairudin states that an IoT-based Durian Tree Monitoring System designed to improve the efficiency of monitoring and irrigation, enhancing the overall health and yield of durian crops [8]. Their work use Raspberry Pi 4 and sensors for temperature, humidity, and soil moisture. The system automates irrigation based on soil moisture readings while data is managed via ThingSpeak Cloud, allowing farmers to remotely monitor trees through graphical data representations. This system improves irrigation decision-making, reduces water wastage and labor costs, and supports sustainability with solar energy, making it suitable for remote agricultural settings. It provides accurate information on durian tree health, aiding agricultural decision-making.

The article presents the development of a Durian Tree Watering System [9] that leverages IoT technology and solar energy to enhance irrigation efficiency. It integrates environmental sensors, a Raspberry Pi for data processing, and a high-pressure water pump, allowing for automated watering based on real-time soil moisture data. Users can monitor and control the system remotely via the Blynk mobile app, promoting effective management of irrigation. The system's reliance on solar power contributes to sustainability and cost reduction, although challenges such as data accuracy, power supply maintenance, and component integration are acknowledged. Overall, the project aims to improve crop health and reduce water waste in durian plantations, supporting sustainable agricultural practices.

The article presents an IoT-based durian fall detector [10] designed to improve the efficiency of durian farmers by utilizing a tilt sensor to detect when durians fall from trees. The system architecture includes a perception layer with the tilt sensor, a network layer using a WeMos D1R2 microcontroller for Wi-Fi connectivity, a middleware layer for cloud data processing, and an application layer that notifies users via the Blynk mobile app. Through testing with various durians, the device achieved an accuracy rate of 86%, enabling farmers to harvest durians at their optimal ripeness and enhancing the quality of fruit available in the market.

REDtone smart durian farming solution [11] has been developed to enhance the quality of Malaysian durians, especially the Musang King variety. The system includes precision farming tools that deliver real-time data on soil moisture, water levels, weather forecasts, pest presence, and soil quality. It also features remote monitoring with CCTV and end-to-end tracking from farm to distribution. Weather prediction, water, and soil monitoring sensors create an optimal growth environment by monitoring and controlling microclimates, improving water use, nutrient management, and soil composition. The system is equipped with wireless 3G/4G connectivity and satellite options, allowing for 24/7 CCTV monitoring via mobile devices. This aids in security, tracking intrusions, and monitoring daily operations, particularly in remote farm areas. Ultrasonic sensors emit sound waves that deter animals like monkeys and squirrels, protecting crops. IoT tracking systems monitor the entire supply chain, ensuring product authenticity, origin, and quality. They also offer real-time data for logistics, helping manage inventory and ensuring traceability and compliance with quality standards.

Table 1 highlights the advantages of the proposed system over existing solutions. While many systems focus on ease of implementation with basic or moderate features, the proposed system offers a more advanced and comprehensive approach. It leverages real-time data collection through a web-based platform, providing better accessibility and usability compared to mobile or cloud-based alternatives. With high levels of automation, ubiquitous access, and seamless control, the proposed system ensures efficient operation, aligning with or exceeding the capabilities of more advanced systems. Furthermore, its user-friendly and customizable interface enhances the overall user experience, making it more suitable for practical applications. By adopting a web-based monitoring system, the proposed solution ensures scalability and efficiency, positioning itself as a robust and future-ready platform for effective management and operation

Table 1 *Comparison of Existing System*

Features/System	IoT Based Durian Monitoring System [8]	Durian Tree Watering System and Solar Energy [9]	IoT Based Durian Fall Detector with Tilt Sensor [10]	Redtone Smart Durian Farming [11]	The proposed system
Ease of implementation	Moderate	Easy	Easy	Complex	Complex
Real-time data collection	Mobile application	Mobile application	Cloud Server	Application	Web application
Ubiquitous access	Moderate	Basic	Basic	High	High
Automation and control	Moderate	Simple	Basic	High	High
User interface	Simple	Basic	Simple	Moderate	User-Friendly
Monitoring system	Mobile-based	Mobile-based	Cloud-based	Mobile-based	Web-based

3. Methodology

This section aims to justify the adopted methodology, emphasizing the selected software development model. The chosen model, the System prototype Model, is a detailed and structured method that adheres to Object-Oriented Programming (OOP) principles. This model supports rapid development and frequent evaluation, enabling faster identification of issues and improving system reliability over time [12].

3.1 System Prototype Model

The system prototype methodology was chosen for this project due to its suitability for the limited timeframe and user needs, which will be evaluated upon project completion. This approach enhances communication between developers and internal users by creating a preliminary version (prototype) and refining it through testing and feedback until a satisfactory product is achieved. It involves repetitive phases of analyzing, designing, and implementing to ensure the final system meets user requirements and can be deployed. The system prototype model also speeds up error detection and allows the development process to start before all system requirements are fully defined.

The project's implementation follows a structured methodology across several phases. During the planning phase, the system to be created is determined, and consultations with the supervisor finalize the title. Information on IoT development and durian management is gathered from the client. In the analysis phase, the necessity of hardware advancements in IoT and system monitoring is examined, recognizing devices for IoT creation and incorporating new functions and features based on feedback. The design phase involves creating schematics for IoT devices and sensors, establishing a rough draft of the system interface, modules, and database layout. A demonstration setup is successfully created, linking sensors and actuators to a microcontroller and cloud server, with the UI, API layer, and communication protocols arranged for efficient data management. The implementation phase sees the designed IoT and systems advanced until the final system is completed, resulting in a cohesive, interconnected system with the database. Finally, during the final system phase, the system is tested to verify the precision of data collected by IoT devices, identifying and addressing system errors to ensure data accuracy. This structured approach ensures the development of a reliable and efficient IoT-based system for deployment of the system to use in real scenarios.

4. System Analysis and Design

This section provides a detailed description of the analysis and structural design used in the creation of IoT-D system. This section discusses the general system architecture, unified modelling language which are use case diagram, sequence diagram, activity diagram, class diagram, data dictionary and lastly user interface design for Smar Durian Farming.

4.1 General System Architecture

The architectural diagram for the IoT- D system is shown in figure 1. Firstly, the architecture of the IoT-D system is divided into three layers which are IoT layer, backend logic layer and application layer. The IoT layer includes a tilt sensor for detecting falling durian fruit, confirmed by an AI camera. Data processed in this layer is sent to Firebase for cloud storage. The backend logic retrieves this data to generate reports on total durian falls and send notifications about animal threats and durian falls. The application layer allows farmers and managers to access a user interface, log in to the monitoring website, manage system information such as on durian, orchard, device, and view orchard monitoring data and fallen durian fruits. They can also obtain total durian production reports and extract files from the system.

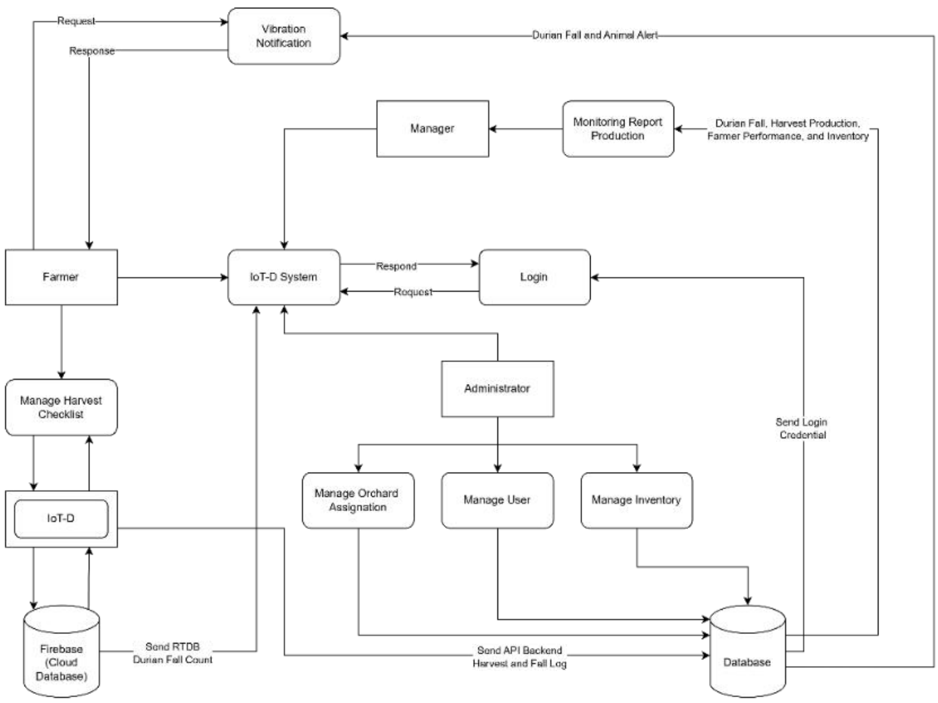


Fig. 1 *System architecture diagram of IoT Based Tree & Fruit Fall Monitoring System*

4.2 Requirement Analysis

The system requirement analysis is the process of constructing clear and autonomous system structure to explain the documentation of system and limitations, ensuring the user needs cooperation. The functionalities of the modules are explained in Table 2.

Table 2 *Functional Requirement of proposed system*

No	Module	Functionalities
1	Tilt sensor and AI Camera (IoT device)	Collect data using tilt sensors, tilt sensors, and AI cameras, where the AI cameras double-confirm durian falls detected by tilt sensors through object recognition
2	Monitoring website	An interface of module farmer and administrator access to the dashboard that provides durian orchard information, which includes the real time data durian fall count, environmental condition and device statues.
3	System Management	Allow users to manage both IoT devices and orchards efficiently such as CRUD function on managing details and operationality.
4	Profile Management	Allow the user involves farmers and administrator to edit their profile when their personal information changes
5	Notification	An interface alerts durian falls, animal threats, or adverse environmental conditions. These real-time notifications help optimize collection routes and ensure prompt durian collection, preserving their quality and reducing potential losses.
6	Reporting	Access both historical and real-time data related to durian farming activities, including the generation and display of detailed reports on orchard identification numbers, durian types, total harvested quantities, fallen count, validated fallen count, device, and dates.
7	User Management	Administrator responsible for managing the user role and assignation orchard to farmer.
7	Harvest Management	Logs harvested durians, generates automated harvest checklists, and validate fall data for accurate record-keeping
8	Inventory Management	Tracks the current stock of harvested durians, updates quantities in real-time, and supports efficient post-harvest handling.

The Non-functional requirements of the module are explained in Table 3.

Table 3 *Non-functional Requirement of proposed system*

No	Requirement	Functionalities
1	Performance	The system should update the dashboard with real-time data within 2-5 seconds of an event being detected by IoT devices. Moreover, the system should handle up multiple IoT devices and scale to accommodate large orchard with minimal latency.
2	Reliability	IoT devices should retry data transmission in case of network failures, and the system should log failed attempts.
3	Usability	The web-based application must be accessible and fully functional on both desktop and mobile devices. Additionally, use a simple, intuitive UI for monitoring, reporting, and device control
4	Security	All data transmitted between Io devices, Firebase, and the Laravel application must be encrypted using TLS/SSL
5	Interoperability	Ensure the system supports REST APIs for integration with third-party tools of future enhancements.

4.3 System Analysis and Design

The visualization system architecture for the system is illustrated by Unified Modelling Language (UML). It offers standardized design notation for IT professionals to enhance the understanding of system flow [13]. There are four types included in UML which are use case diagram, sequence diagram, class diagram, and activity diagram.

4.4 Use Case Diagram

The use case diagram illustrates the comprehensive functionalities of the IoT-D System, emphasizing the interactions among three user roles Admin, Farmer, and Manager. The admin has the highest level of access, including the ability to manage users, devices, orchards, durians, inventory, and stock movements. The Farmer can log in, edit their profile, view dashboards, manage orchards and devices, receive real-time notifications for durian falls and environmental changes, and view harvest data. They can also handle harvest checklists, which are automatically generated by the system, and perform new data entries. Meanwhile, the Manager primarily focuses on monitoring and analysis tasks such as viewing dashboards, accessing production reports, analyzing fruit fall data, monitoring notifications, and reviewing inventory status. Appendix B illustrates the use case diagram of the proposed system.

4.5 Sequence Diagram

The sequence diagram for the durian fruit fall monitoring system illustrates the interaction between users and various system components. It highlights the role of IoT sensors that are attached to durian tree, which detect fruit falls and potential animal threats using AI-powered image processing and object recognition. Upon detecting an event, the sensors transmit real-time data to Firebase, capturing environmental changes and fall incidents. This information is immediately reflected on the monitoring website, ensuring that users have access to up-to-date insights. Simultaneously, the system processes and stores detailed event data, such as durian fall timestamps and object classifications, in a MySQL database for analysis, record-keeping, and report generation. The detailed sequence of these processes is illustrated in the sequence diagram included in the Appendix.

4.6 Activity Diagram

The activity diagram for the IoT-D System outlines the key operational workflow from user login to data handling. The process begins by verifying the user's if the ID is not detected, the operation is terminated. Once authenticated, the system actively monitors durian falls and potential animal threats. When a durian fall is detected, the event is recorded in Firebase, and an alert notification is sent to the user. In cases of animal interference, the system issues a separate alert to warn of potential danger. These real-time notifications keep users, especially farmers, informed of critical events in the orchard. Monitoring data is displayed on the website interface, where users have the option to handle harvest-related data. If chosen, the system retrieves relevant records from the database, generates a harvest checklist, and updates the inventory accordingly to ensure accurate stock tracking and orchard management. The activity diagram of the system is shown in figure 2.

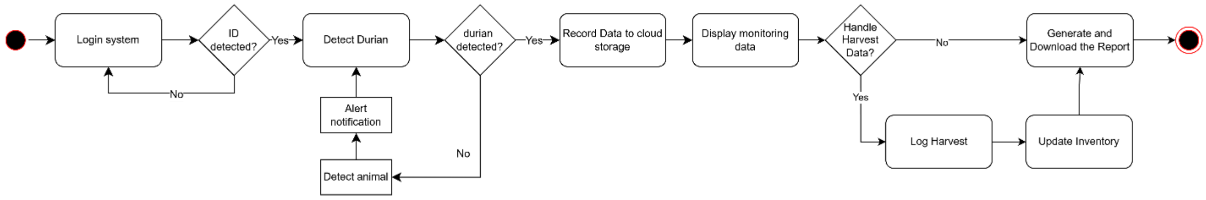


Fig. 2 Activity *Diagram of IoT Based Tree & Fruit Fall Monitoring System*

4.7 Database Design

A class diagram is a type of Unified Modelling Language (UML) that represents the static structure of a system by showing the classes, attributes, operation, method, and the relationship between the objects. There are two types of database design in this Smart Durian System which are logical design (Class diagram) and physical design (data dictionary) which can be found in Appendix C.

4.8 IoT-D Prototype and Algorithm

Figure 3 illustrates the main monitoring device setup for the IoT-D System. The design integrates multiple components, including a tilt sensor and an AI camera module, mounted on a flexible stand to optimize positioning. The wiring connects to a microcontroller, which processes sensor input and transmits data in real time. This setup enables accurate detection of durian falls and supports object recognition capabilities, making it a key element in the overall smart orchard monitoring solution.

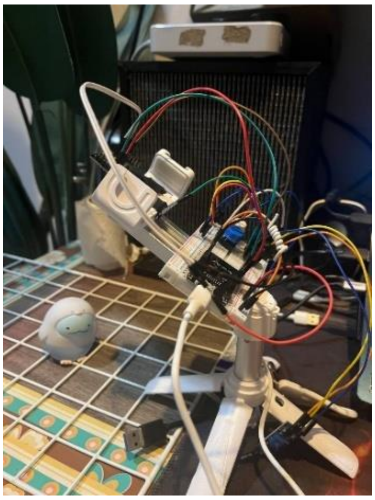


Fig. 3 *Prototype of IoT-D*

Figure 4 shows the integration of IoT-D system, YOLOv5 with ESP32-CAM devices for real-time image classification, detecting objects like durians or animals. When images are uploaded, YOLOv5 processes them to enable automated decision-making, such as logging vibration events and updating Firebase and MySQL databases. This edge-level machine learning deployment enhances agricultural monitoring by reducing manual labor and improving fruit fall detection accuracy. By leveraging AI-driven image classification, farmers can efficiently track harvesting conditions, monitor disturbances, and maintain comprehensive records. The system showcases how AI can optimize smart farming, enable precise, automated monitoring while minimize human intervention in agricultural environments.

Algorithm: ESP32-CAM Image Classification and Logging Service

```
Input:
    - Uploaded ESP32-CAM images (via HTTP POST)
    - YOLOv5 model weights and detection script
    - Firebase and MySQL API endpoints
Output:
    - Classified images
    - Vibration/harvest logs to Firebase and MySQL API
    - Web dashboard for image browsing
Steps:
1. Initialize Flask web server with endpoints for:
    - Image upload (/upload)
    - Image listing (/list-images/original, /list-
images/classified)
    - Image serving (/images/<type>/<filename>)
    - Web dashboard (/)
2. On image upload:
    a. Save received image to the original images directory
    b. Start asynchronous classification task for the image
3. Image classification (YOLOv5):
    a. Run YOLOv5 detection on the uploaded image
    b. Save the detected/classified image to the classified
directory
    c. Parse YOLOv5 label file for detections:
        - Count number of durians (class 0)
        - Detect presence of animal (class 1)
    d. If durian detected:
        - Log detection count to MySQL API (logType=1)
        - Update vibrationCount in Firebase for the sensor
    e. If animal detected:
        - Log to MySQL API (logType=2)
        - Do not update the Firebase
    f. Otherwise, do not log
```

Fig. 4 *Algorithm Integration IoT-D system, YOLOv5 with ESP32-CAM*

5. Result and Discussion

Based on figure 5 the IoT-D system effectively detects and classifies objects using YOLOv5, as demonstrated by result graph of precision and recall mAP. The training metrics show consistent improvement over epochs, indicating effective learning and accurate predictions. Precision and recall values quickly rise and stabilize above 0.9, reflecting reliable detection with minimal false positives and missed objects. The mAP metrics further validate the model's ability to correctly localize and classify durians, twigs, and animals. These results confirm that the system is well-trained for real-time fruit fall detection and classification tasks in smart durian farming

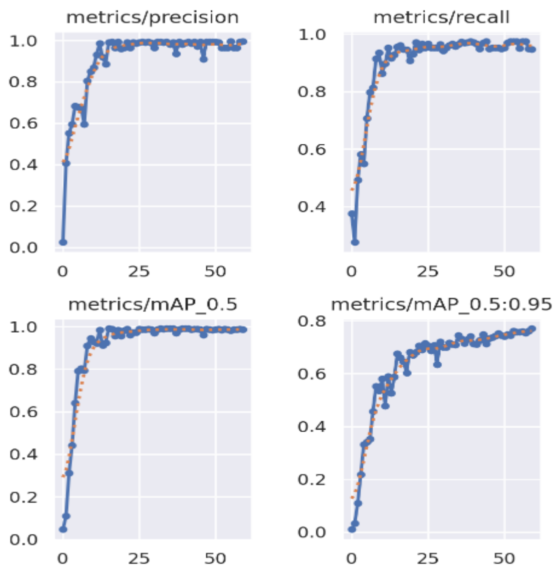


Fig. 5 Result Graph

5.1 System Implementation

Figure 6 illustrates the prototype of the IoT-D application system, integrated with a miniature orchard simulation. The setup features an ESP32-CAM microcontroller connected to a tilt vibration sensor and a physical push button. These components are mounted on a stable tripod stand, with the sensor strategically attached to a netting structure that mimics a durian-catching net. The sensor is designed to detect vibrations resulting from durians falling onto the net, enabling real-time detection and classification via the ESP32-CAM and Edge Impulse object detection model. This physical arrangement facilitates accurate data capture for testing and validation of the smart orchard monitoring system.

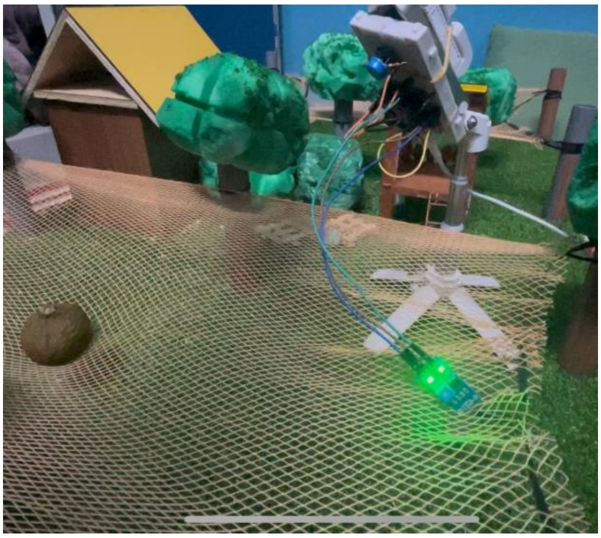


Fig. 6 *Prototype of IoT-D*

Figure 7 displays the login interface of the IoT-D web-based application system. This login page serves as a universal access point for all user roles, allowing users to authenticate using their registered email and password. The interface features a minimalist and clean design, promoting ease of use and accessibility. A "Remember me" option is provided for convenience, along with a link for password recovery, ensuring a user-friendly and secure login experience for system administrators, farmers, and other stakeholders.

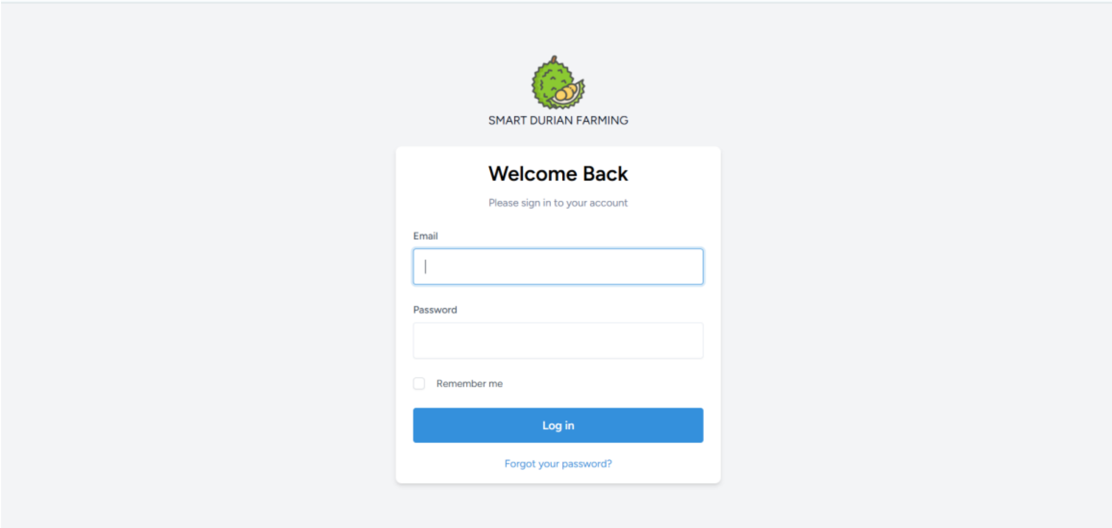


Fig. 7 *Login Page of IoT-D*

Figure 8 illustrates the administrator dashboard of the IoT-D web-based system. This centralized interface provides a comprehensive overview of critical information, including the total durian fall count, durian production by type, and a real-time notification panel that logs events such as durian falls and animal detections per device. Additionally, users can monitor orchard status, connected IoT devices, production records, and current weather conditions powered by the Open-Meteo API. This structured layout enhances operational awareness and decision-making.

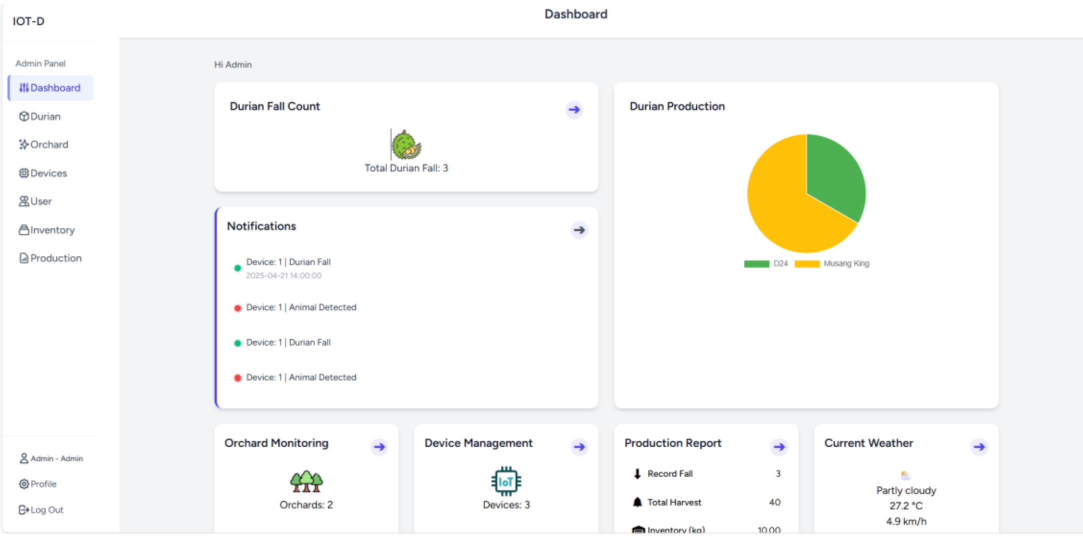


Fig. 8 *Dashboard of IoT-D system*

Figure 9 illustrate the administrative interface of the IoT-D application system. The admin panel provides comprehensive CRUD (Create, Read, Update, Delete) capabilities for managing Durians, Orchards, and IoT devices. The interface enables administrators to add new durian varieties, assign orchard locations with detailed parameters such as tree count, device type, and durian type, and monitor real-time data such as durian falls and device connectivity. Device management includes status toggling, detailed information access, and edit/delete functions, ensuring smooth hardware supervision. In contrast, the farmer interface is restricted to read-only access, allowing staff to view relevant operational data without the ability to make modifications. This design maintains centralized authority, minimizes errors, and supports streamlined decision-making. Overall, this structure ensures effective information flow and operational oversight across the system.

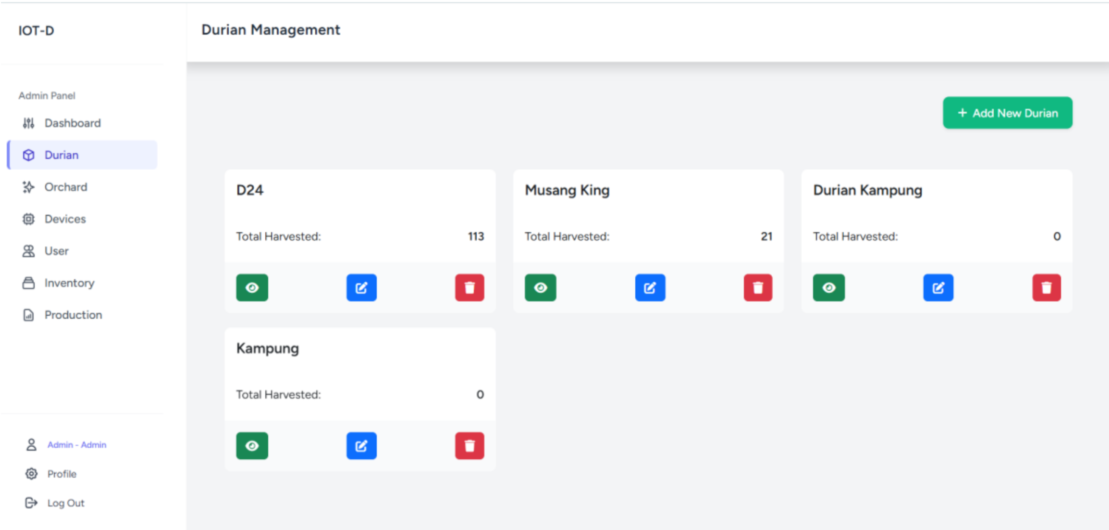


Fig. 9 *System management (durian) Page of IoT-D*

Figure 10 below illustrates the functionalities available on the admin and staff sides for orchard and operations. The admin has the capability to assign orchards to farmers, facilitating the management of orchard assignments. The user management interface allows the admin to view all users, including their roles and assigned orchards, ensuring efficient oversight of user permissions and responsibilities. The farmer can view the orchard assigned to them by the admin. This interface includes a notification log that keeps farmers informed the fallen durians, ensuring that they are alert and response.

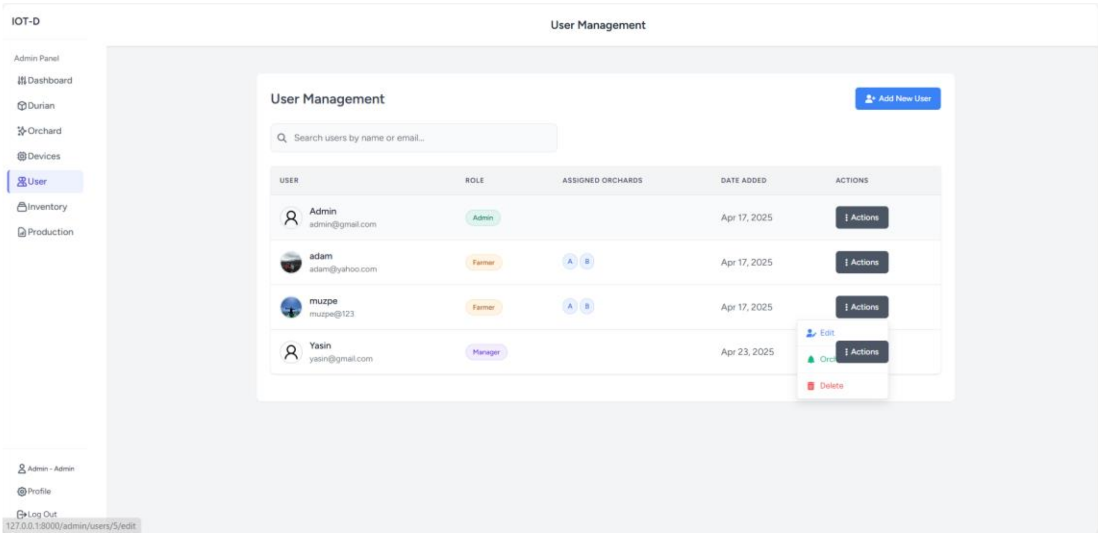


Fig. 10 *User management (orchard assignation farmer) of IoT-D system*

Figure 11, displayed on the manager interface for reports and analytics, provide a comprehensive overview of all operations, including the harvest report, fall monitoring report, inventory report, and a summary of recent activities. The manager can simply click on the desired report, which will then be presented in both graphical and tabular formats. Additionally, the manager can filter the data to easily locate specific information and export reports in various document formats, such as PDF and Excel. This interface ensures that the manager has full visibility into operations performed by both farmers and admins, allowing for smooth and efficient workflow smoothly.

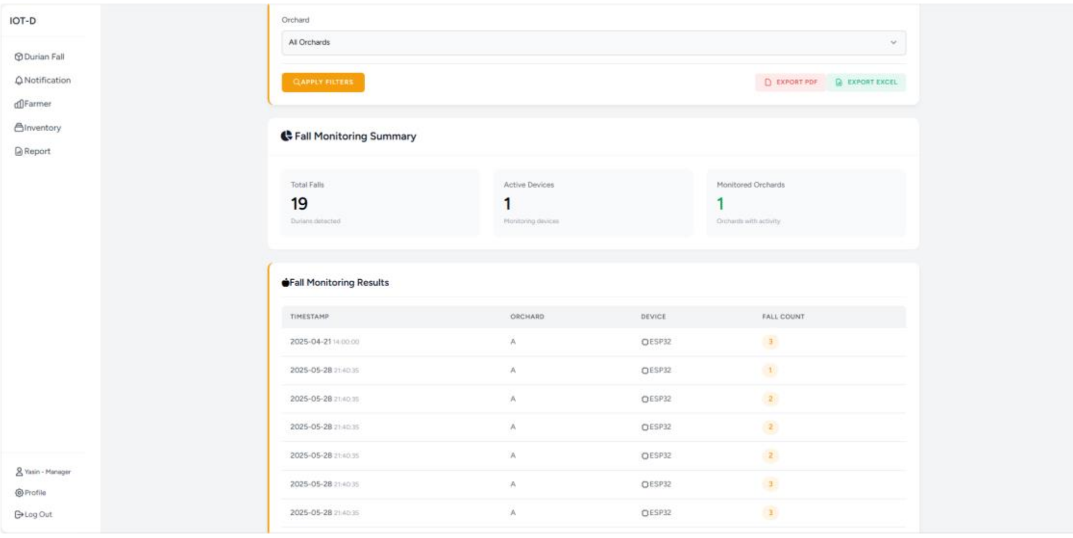


Fig. 11 *Reporting operation Manager of IoT-D system*

5.2 System Testing

Before going live, the IoT-D system must undergo thorough testing to ensure it meets all specified requirements and is free from critical bugs. This testing phase is essential to guarantee the system's reliability, functionality, and overall performance. A comprehensive test plan was implemented for each module, covering various scenarios to validate expected outcomes. Notably, the system was deployed and tested on a live server environment, rather than a local server, indicating that the system has already been taken online for real-world validation.

5.2.1 Test Plan

A test plan outlines the system testing strategies, goals, timeline, deliverables, estimates, and resources. It provides guidance for the testing activities and describes the overall testing process. Tables 4 to 13 present the detailed test plans according to each system module, covering functionalities such as IoT device detection, dashboard interaction, notification handling, user and profile management, and harvest reporting

Table 4 *IoT-D Devices Module*

Test	Expected Result	Actual Result
Detect durian fall via tilt sensor	Tilt sensor detects vibration or tilt; triggers durian falls count increment	PASS
AI camera classifies object locally (Edge Impulse)	AI camera identifies objects (e.g., durian) using Edge Impulse, logs result locally example durian" or animal.	PASS
Data is successfully sent to the database.	Real Time Databases and Databases receive the total durian fall count and log for classification of durian or animal.	PASS
Multi-sensor input sync (Tilt + AI camera)	Events detected by both sensors should be synchronized and logged as one event	PASS
Handle invalid or null sensor data	Device handles invalid or null values without crashing	PASS

Table 5 *Monitoring Webpage (Dashboard) Module*

Test	Expected Result	Actual Result
User (Admin, Farmer, Manager) login with user email and password	Successful login and access to the monitoring website	PASS
Display real-time durian fall data	Dashboard shows fall count increment shortly after device triggers event	PASS
Show correct orchard and device status	Orchard and device cards correctly update status	PASS
Notifications Display	All triggered events are shown in the Notifications section	PASS
Weather Widget	Weather data loads accurately with temperature and conditions	PASS

Table 6 *Profile Management Module*

Test	Expected Result	Actual Result
View Profile	User profile info (name, email, role) is displayed correctly	PASS
Role Restriction	Staff cannot edit their role or others' roles	PASS
Notification Display	All triggered events are shown in the Notifications section	PASS
Unauthorized Access	Redirected to login page	PASS

Table 7 *System Management (devices, durian, and orchard) Module*

Test	Expected Result	Actual Result
Creation of (durian, devices, orchard)	Administrator can create new data successfully	PASS
Edit exists of (durian, devices, orchard)	Administrator can edit data as needed	PASS
Delete of (durian, devices, orchard)	Administrators can delete data when they are no longer needed	PASS
Accessibility of data (durian, devices, orchard)	All triggered events are shown in the Notifications section	PASS
Accessibility all of data	All data are easily accessible to all user	PASS

Table 8 *Notification module*

Test	Expected Result	Actual Result
Durian fall notification	Real-time notification is triggered and displayed (e.g., "Durian Fall Detected - Device X")	PASS
Animal threat detected	Notification is shown (e.g., "Animal Threat Detected - Device Y")	PASS
Notifications visibility for farmer (view-only)	Notifications are viewable but cannot be deleted or modified by farmer	PASS
Admin clears notification	Notification is removed from the panel and not visible in farmer view	PASS
Manager visibility	All data notification visible as summary on manager	PASS

Table 9 *User Management Module*

Test	Expected Result	Actual Result
Verify user addition functionality	User should be added successfully and appear in the user list with correct details.	PASS
Verify user deletion functionality	User should be removed from the user list, and a confirmation message should appear	PASS
Verify assigning an orchard to a user	Orchard should be assigned to the user, and the assignment should be visible under "Current Assignments".	PASS
Verify removing an orchard from a user	Orchard should be removed from the user's list of assignments.	PASS
Verify search functionality for users	Matching users should be displayed based on the search criteria.	PASS

Table 10 *Harvest management Module*

Test	Expected Result	Actual Result
Verify the addition of a new harvest entry	The new harvest entry should be added successfully and appear in the "Harvest Records" list with a "Pending" status.	PASS
Verify editing an existing harvest entry	The harvest entry should be updated successfully, and changes should reflect in the "Harvest Records" list.	PASS
Validate checklist generation for a harvest entry	The checklist fields (Grade, Condition, Storage Location, etc.) should populate accurately based on the entry's data.	PASS
Harvest Distribution	The graph should accurately reflect the data from the "Harvest Records" (e.g., total harvest by type or date).	PASS
Record Validation	The system should display an appropriate error message, and the entry should not be saved.	PASS

Table 12 *Farmer performance Module*

Test	Expected Result	Actual Result
Verify the "Assigned Orchards" section	The assigned orchards should be displayed accurately with all details (name, location, etc.).	PASS
Verify the "Harvest Performance" graph	The graph should correctly represent the harvest data with accurate labels, values, and dates.	PASS
Verify the "Recent Harvests" table	The table should list recent harvests with accurate details example date, location, weight.	PASS
Validate sorting in "Recent Harvests" table	The table should be sorted correctly based on the selected column.	PASS
Verify data consistency across modules	The data displayed in Farmer Performance should match the corresponding data in other relevant modules.	PASS

Table 13 *Reporting Manager Module*

Test	Expected Result	Actual Result
Generation of Harvest Report in PDF	The Harvest Report should generate successfully in PDF format with all the correct data based on the filters applied.	PASS
Display of real-time data in Fall Monitoring Report	All real-time data (Total Falls, Active Devices, etc.) should be displayed accurately based on the selected filters.	PASS
Export of Inventory Report in Excel	The Inventory Report should be downloaded successfully in Excel format with all the correct data based on the filters applied.	PASS
Verify data consistency across reports	Data should be consistent across all reports for the same filters.	PASS
Verify error handling for invalid filters	The system should display an appropriate error message and prevent report generation for invalid filters.	PASS

User Acceptance is an important phase in the development of the lifecycle of IoT-D Monitoring System, ensuring the system meets the real-world and expectations of its intended users which are administrator, farmer, and manager. The purpose of UAT is to validate that the system functions correctly from the user's perspective in terms of user interface and system functionality, also to identify any gaps between system functionality and user expectations before deployment. User satisfaction will be rated on a scale from 1 (not satisfied) to 5 (very satisfied) once data from these individuals has been gathered and analyzed.

Table 14 *Result of user acceptance testing for admin user*

No	Acceptance Requirement	Actual Result				
		1	2	3	4	5
User Interface						
I.	Dashboard displays real-time durian fall count, notifications, weather				/	
II.	Admin menu displays all sections: Dashboard, Durian, Orchard, Devices, Users					/
III.	Notification panel is visible and clearly highlights alerts					/
IV.	Profile pages are accessible with edit options					/
System Function						
V.	Can create, edit, and delete Durian entries				/	
VI.	Can create, edit, and delete Orchard entries					/
VII.	Can register, assign, and remove IoT devices				/	
VIII.	Can create and manage user accounts with specific roles					/
IX.	Receives real-time alerts for falls, animals, and weather conditions					/
X.	Manage inventory and storage					/
XI.	Can assign orchards to users				/	

Table 15 *Result of user acceptance testing for farmer user*

No	Acceptance Requirement	Actual Result				
		1	2	3	4	5
User Interface						
I.	Farmer dashboard displays own orchard data and current alerts				/	
II.	Navigation is limited to personal orchards, devices, and monitoring views					/
III.	Notifications for fall and threats are prominently displayed				/	
System Function						
V.	Farmer can monitor durian falls and threats in real-time				/	
VI.	Farmer can view but not edit orchard/device configurations					/
VII.	Farmer receives alerts based on device activity from their assigned orchards					/
VIII.	Farmer can perform harvest checklists by log harvest.				/	
IX.	Farmer can remove stock (stock-out) when durians are sold or transported					/

Table 16 *Result of user acceptance testing for manager user*

No	Acceptance Requirement	Actual Result				
		1	2	3	4	5
User Interface						
I.	The Manager dashboard shows overview of all orchard performances				/	
II.	Notifications and production summaries are visible				/	
System Function						
V.	Manager can view all orchard and durian data but cannot modify them				/	
VI.	Manager can monitor sensor alerts across multiple orchards					/
VII.	Manager can generate and export reports (fallen durian, harvest, inventory) in PDF and excel				/	

6. Conclusion

In conclusion, the IoT-D System represents an innovative solution tailored to address the challenges of durian orchard management by seamlessly integrating IoT sensors, AI-based image processing, and cloud technologies. The system architecture features IoT sensors, including tilt sensors and AI cameras, supported by a middleware layer that utilizes Firebase for efficient data storage. The application layer provides an intuitive user interface, enabling farmers and managers to log in, monitor real-time data, and generate comprehensive reports on durian production.

The methodology employed, specifically the system prototype development approach, thoroughly details the requirements analysis, ensuring the robustness and effectiveness of the proposed system. The anticipated outcomes of this project include enhanced operational efficiency in durian farming, facilitated by real-time monitoring and data-driven decision-making, ultimately contributing to the sustainability and profitability of the durian farming industry.

Future work should prioritize enhancing system robustness by developing redundant communication pathways and integrating more resilient hardware components. Pilot projects and user feedback will be essential for iterative improvements, ensuring the system adapts to the dynamic needs of real-world orchard environments. Further studies and comparisons with existing systems, alongside a detailed analysis of related work, will yield valuable insights into the strengths and limitations of the proposed system. As the prototype development progresses towards becoming a smarter solution in durian agriculture, the Smart Durian Farming: IoT-Based Tree and Fruit Fall Monitoring System stands as a significant contribution to fostering a more connected, intelligent, and productive approach to durian yield production

Acknowledgement

The authors would like to thank the Faculty of Computer Science and Information Technology, Universiti Tun Hussein Onn Malaysia for its support.

Conflict of Interest

Authors declare that there is no conflict of interests regarding the publication of the paper.

Author Contribution

This journal requires that all authors take public responsibility for the content of the work submitted for review. The contributions of all authors must be described in the following manner:

The authors confirm contribution to the paper as follows: **study conception and design**: Muhamad Haiqal Mahathir, Rosmamalmi Mat Nawi; **data collection**: Muhamad Haiqal Mahathir; **analysis and interpretation of results**: Muhamad Haiqal Mahathir, Rosmamalmi Mat Nawi; **draft manuscript preparation**: Muhamad Haiqal Mahathir, Rosmamalmi Mat Nawi. All authors reviewed the results and approved the final version of the manuscript.

References

Journal

[1] Wang, Z., Wang, Y., Lan, P., & Sun, F. (2019). The Application of the Internet of Things Technology in Apple Production. *2019 IEEE 11th International Conference on Advanced Infocomm Technology, ICAIT 2019*, 109–112. https://doi.org/10.1109/ICAIT.2019.8935895

[2] Rajan, P. (2024). Integrating IoT Analytics into Marketing Decision Making: A Smart Data-Driven Approach. *International Journal of Data Informatics and Intelligent Computing*, *3*(1), 12–22. https://doi.org/10.59461/IJDIIC.V3I1.92

[3] Miorandi, D., Sicari, S., de Pellegrini, F., & Chlamtac, I. (2012). Internet of things: Vision, applications and research challenges. *Ad Hoc Networks*, *10*(7), 1497–1516. https://doi.org/10.1016/J.ADHOC.2012.02.016

[4] Firouzi, F., Farahani, B., Weinberger, M., DePace, G., & Aliee, F. S. (2020). IoT Fundamentals: Definitions, Architectures, Challenges, and Promises. *Intelligent Internet of Things: From Device to Fog and Cloud*, 3–50. https://doi.org/10.1007/978-3-030-30367-9_1

[5] Kang, H., & Chen, C. (2020). Fast implementation of real-time fruit detection in apple orchards using deep learning. *Computers and Electronics in Agriculture*, *168*, 105108. https://doi.org/10.1016/J.COMPAG.2019.105108

[6] Sumit Singh. (2022, October 19). *What is image classification AI model & how to build it?* https://www.labellerr.com/blog/how-to-build-an-image-classification-ai-model/

[7] Culjak, I., Abram, D., Pribanic, T., Dzapo, H., & Cifrek, M. (2012). *A brief introduction to OpenCV*.

[8] Hairudin, M. I., Mustafa, M. Z., & Ahad, R. (2023). Durian Tree Watering System Uses Iot And Solar Energy. Research and Innovation in Technical and Vocational Education and Training, 3(1), Article 1. https://doi.org/10.30880/ritvet.2023.03.01.002

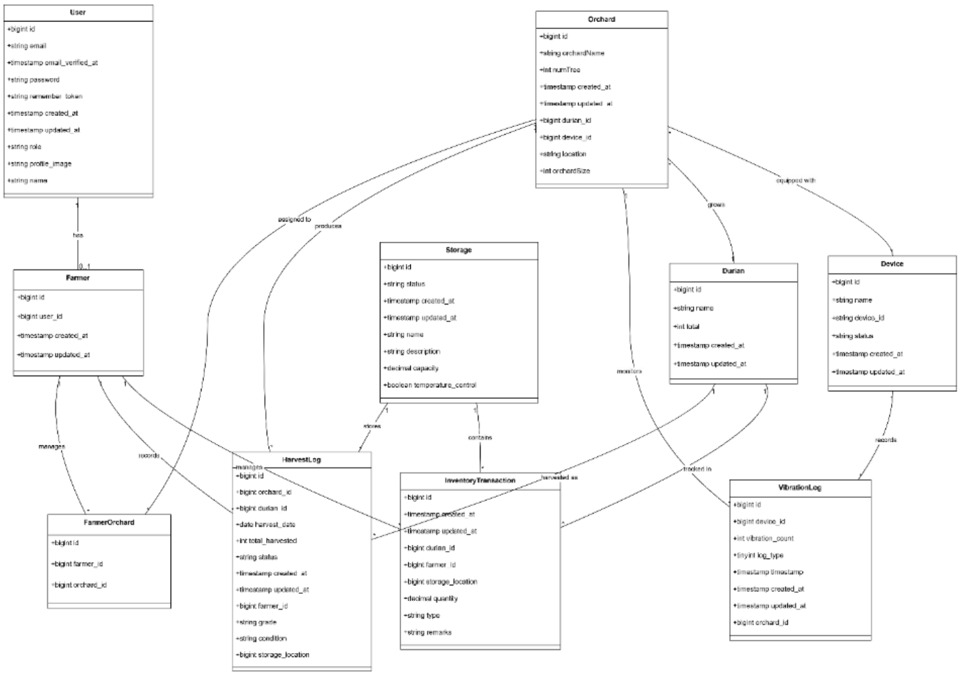
[9] Purwanto, E. S., Calvinsyah, G. C., & Mulia, W. A. (2024). IoT-Based Durian Fall Detector with Tilt Sensor: Enhancing Durian Farmer Harvest Efficiency (No. 03). ICIC International 学会. https://doi.org/10.24507/icicel.18.03.285

[10] REDtone to develop IoT solutions for durian farming with Bangi Farm Resort. (2018, August 6). Digital News Asia. https://www.digitalnewsasia.com/business/redtone-develop-iot-solutions-durian-farming-bangi-farm-resort

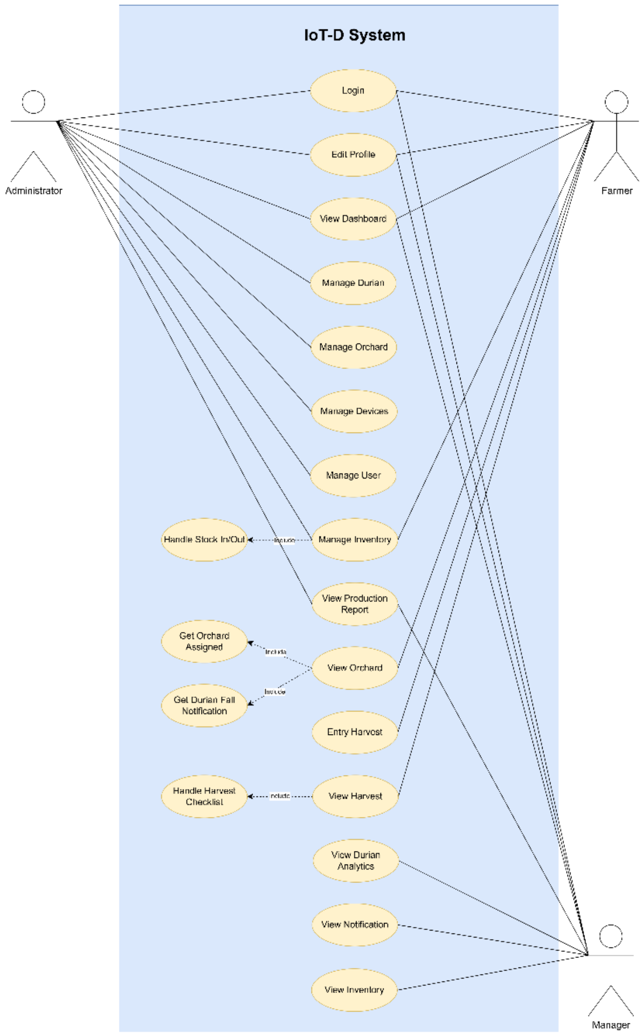
[11] Oktaviana, S., Febriani, D., Yoshana, I., & Payanta, L. (2020). FoodX, a System to Reduce Food Waste (p. 365). https://doi.org/10.1109/IC2IE50715.2020.9274576

[12] Pressman, R. and Maxim, B. (2014) Software Engineering: A Practitioner's Approach, 8th Edition. McGraw-Hill. Boston.

Appendix A: Database Design

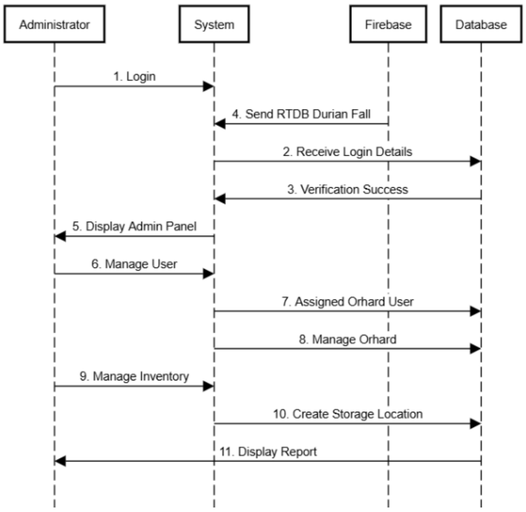


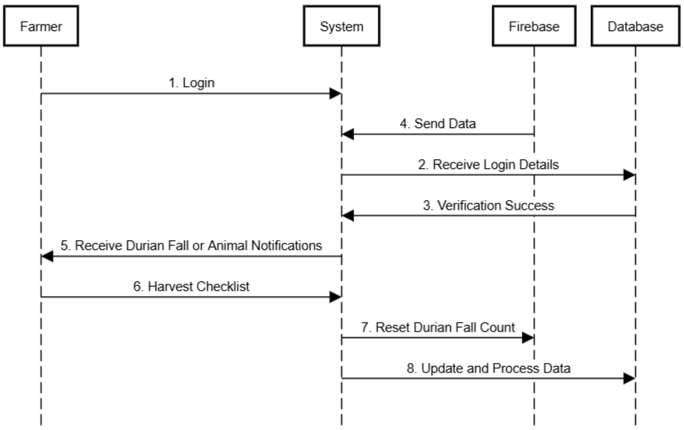
Appendix B: Use Case Diagram

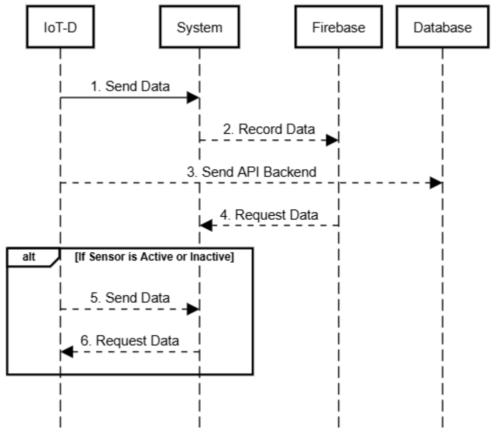


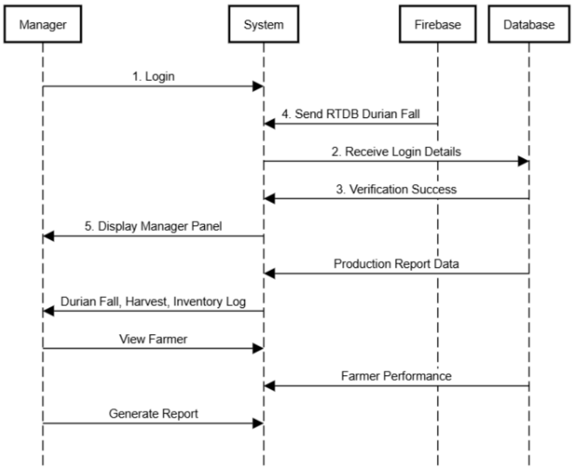
Applied Information Technology and Computer Science Vol. 6 No. 2 (2025) p. 170-189

Appendix C: Sequence Diagram

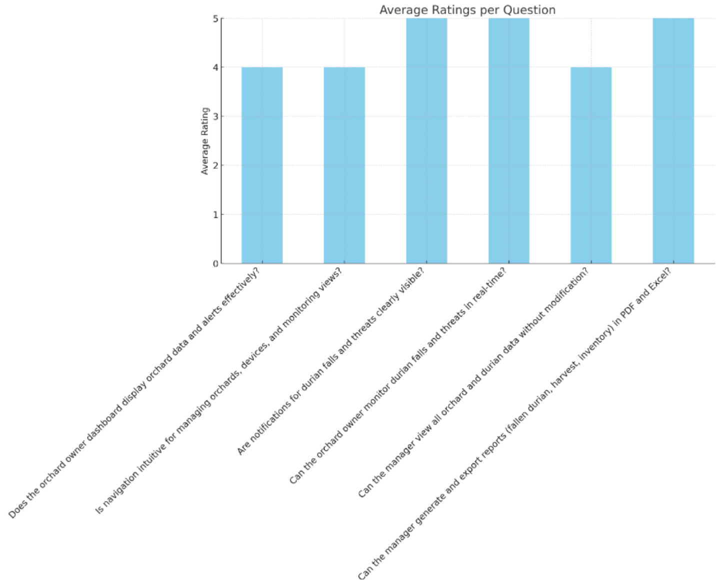


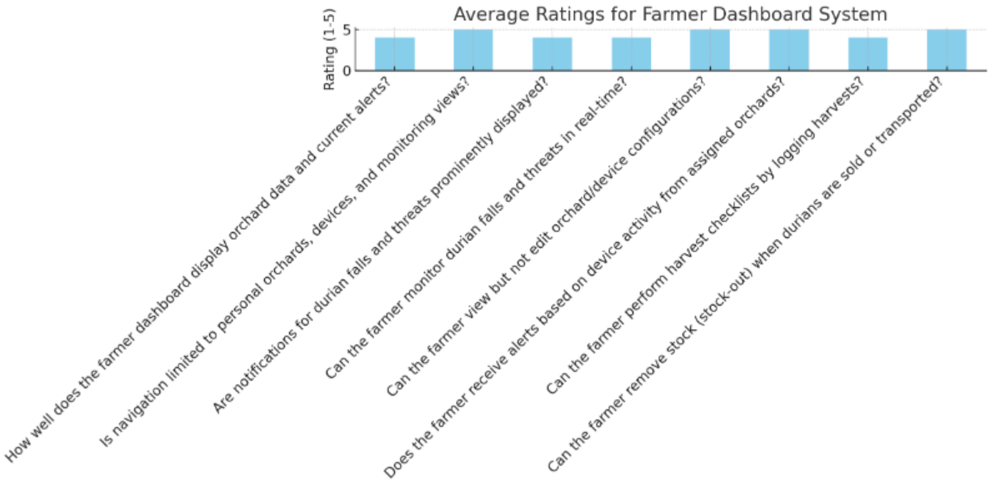


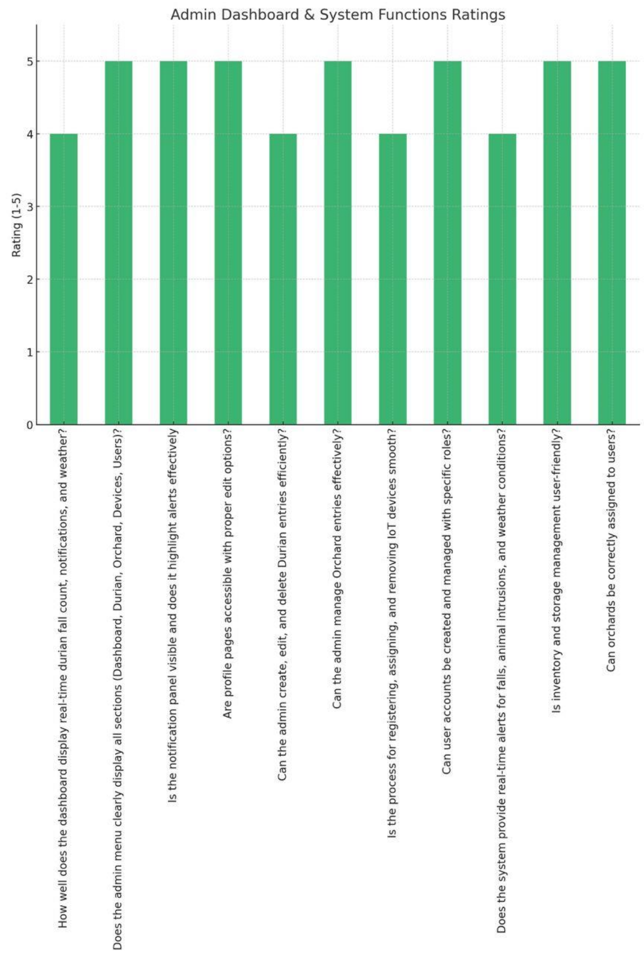




Appendix D: User Acceptance






Admin Dashboard & System Functions Ratings
Rating (1-5)
0
1
2
3
4
5
How well does the dashboard display real-time durian fall count, notifications, and weather?
Does the admin menu clearly display all sections (Dashboard, Durian, Orchard, Devices, Users)?
Is the notification panel visible and does it highlight alerts effectively
Are profile pages accessible with proper edit options?
Can the admin create, edit, and delete Durian entries efficiently?
Can the admin manage Orchard entries effectively?
Is the process for registering, assigning, and removing IoT devices smooth?
Can user accounts be created and managed with specific roles?
Does the system provide real-time alerts for falls, animal intrusions, and weather conditions?
Is inventory and storage management user-friendly?
Can orchards be correctly assigned to users?