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

AITCS

Vol. 6 No. 2 (2025) 868-887
https://publisher.uthm.edu.my/periodicals/index.php/aitcs

PalmCare: Smart Mobile Platform for Oil Palm Plantation Management

Veshal A/L Ramesh[1], Nur Liyana Sulaiman[1*]

[1] *Faculty of Computer Science and Information Technology, Universiti Tun Hussein Onn Malaysia, 86400 Parit Raja, Johor, MALAYSIA*

*Corresponding Author: nrliyana@uthm.edu.my
DOI: https://doi.org/10.30880/aitcs.2025.06.02.047

Article Info

Received: 15 June 2025
Accepted: 20 November 2025
Available online: 30 November 2025

Keywords

Oil Palm, Near Field Communication, Rule-Based System, Management Information System

Abstract

PalmCare is a mobile platform designed to address operational inefficiencies at Felda Mayam Plantation, a 10-acre oil palm plantation in Pahang owned by Mr. Arbaleesparan. The current manual processes for task management, worker performance monitoring, and harvest record-keeping rely on paper-based systems and spreadsheets, leading to delays and miscommunication. The objective of this project to design PalmCare using object-oriented approach, develop mobile application and test using User Acceptance Testing. By applying the Software Development Life Cycle Model, PalmCare built with Flutter and a PHP framework, Supabase for real-time database synchronization. Unified Modelling Language diagrams use for requirement modelling and the 3-tier Model-View-Controller architecture was applied. The overall average System Usability Scale mean score is 75.0 over 100, which indicates good usability where users generally find the system easy to use and efficient. The significance of this project to oil palm industry is improved productivity, cost savings, and sustainable plantation management.

1. Introduction

The oil palm tree, which can only be grown in tropical climates, yields premium oil that is mostly used for cooking in developing nations. Moreover, it finds application in food items, cleaning agents, cosmetics, and, to a lesser degree, biofuel [1]. However, oil palm growth in Malaysia began in 1870 mostly for aesthetic purposes. Malaysia's oil palm plantations have grown dramatically over time. 20 million metric tons (or 27%) of the 72 million metric tons of oil palm that was grown worldwide in 2018 came from this nation [2]. The oil palm plantation, Felda Mayam Plantation in Pahang, has been operational since 1972, covering an area of 10 acres. Owned by Mr. Arbaleesparan A/L Kalimuthu, the plantation's operations are currently managed manually, relying on paper-based systems and simple spreadsheets. Managers and supervisors assign tasks, monitor worker performance, and keep harvest records manually. Communication within the plantation mainly occurs through phone conversations and in-person meetings, causing delays in information flow and responses. Worker activities, task completion, and fruit ripeness are not tracked in real time or only tracked minimally, leading to inefficiencies in the plantation's operations. This manual and unintegrated approach results in inefficiencies and delayed decision-making.

The objectives of PalmCare are to design PalmCare: Smart Mobile Platform for Oil Palm Plantation Management using an object-oriented approach. To develop a PalmCare: Smart Mobile Platform for Oil Palm Plantation Management using Android technology and to test PalmCare: Smart Mobile Platform for Oil Palm Plantation Management by applying functionalities testing user acceptance testing.

PalmCare aimed at modernizing the management of oil palm plantations, addressing key challenges like manual data management, inefficient labor allocation, and delayed task reporting. The platform integrates Near Field Communication (NFC) technology, Artificial Intelligence (AI)-driven task management, and real-time tracking to optimize resource allocation, improve productivity, and reduce errors. It features modules for user management, plantation mapping, task assignment, real-time analytics, activity monitoring, and worker training. PalmCare also provides structured training for new workers, ensuring seamless integration into the workforce. By centralizing operations and automating routine tasks, PalmCare enhances decision-making, reduces operational costs, and boosts overall plantation efficiency, marking a significant step towards digital transformation in plantation management.

The remainder of this paper is organized as follows, Section 2 provides a literature review focus on technology used and comparison between existing systems and PalmCare. Section 3 discusses the methodology used for system development. Section 4 presents the system design and implementation. Section 5 presents Results and Discussions including Testing. Finally, Section 6 concludes the paper.

2. Literature review

This literature review delves into the technologies that underpin PalmCare: Smart Mobile Platform for Oil Palm Plantation Management. By examining existing systems, it provides a foundation for understanding the current state of technology ensuring the application is designed to effectively meet the needs of PalmCare and address common industry challenges.

2.1 Management Information System in Oil Palm Plantation

Management Information Systems (MIS) are becoming increasingly important in oil palm plantations for efficient data management and labor oversight. Traditional methods of task allocation, inventory management, and resource tracking lack integration, which hampers quick decision-making [3]. MIS centralizes data on a single platform, enabling managers to track workers, workers assign applications, and worker performance in real time [4]. This integration supports better resource use, reduces waste, and improves labor management by providing up-to-date information on employee availability and performance.

MIS helps with precise resource distribution, reducing costs and environmental impact, and plays a key role in harvest tracking and crop yield forecasting. This helps plantations prepare for market fluctuations and align financial planning with production. Additionally, MIS supports sustainability by tracking environmental factors and ensuring compliance with regulations such as the Roundtable on Sustainable Palm Oil (RSPO) [5].

However, challenges like high initial costs, the need for reliable network connectivity, and the importance of accurate data must be addressed. With proper training and maintenance, MIS remains a valuable tool for modernizing plantation management, improving productivity, optimizing resources, and achieving sustainable growth [6].

2.2 Technology used

The PalmCare: Smart Mobile Platform for Oil Palm Plantation Management leverages several key technologies to enhance its functionality. The mobile application framework, particularly Flutter, simplifies app development by providing pre-built components and enabling code reuse across platforms. Integrated Development Environments (IDEs) like Visual Studio and Android Studio combine essential development tools into one application, boosting developer productivity with features like auto-completion and error detection. Supabase provides the database backbone with its Realtime Database offering Realtime data synchronization, offline capabilities, and advanced querying, essential for managing the app's live updates and complex data needs [7].

Near Field Communication (NFC) technology can significantly enhance operations in oil palm plantations by enabling efficient attendance tracking and reporting. Durable NFC tags, resistant to harsh environmental conditions, can be placed at key locations or equipment within the plantation [8]. Workers can use smartphones to scan these tags to mark attendance, update task statuses, or report activities. This eliminates manual reporting, streamlines workforce management, and improves efficiency. NFC tags can also store critical data like task instructions, equipment details, or material placements, which workers can access instantly. Applications leveraging NFC technology, such as attendance and reporting tools, ensure accurate monitoring, improve productivity, and support better decision-making in plantation operations.

A Rule-Based System (RBS) is a decision-making framework that uses predefined "if-then" rules to make decisions or trigger actions based on specific inputs [9]. It consists of three components: a set of rules, working memory to store facts and state, and an inference engine to apply the rules [10]. RBSs are valued for their simplicity, transparency, and ease of maintenance, making them effective in automating repetitive tasks and reducing errors. A Rule-Based System (RBS) in an oil palm plantation management context can be specifically configured to address late attendance issues. The system activates predefined rules only when it detects a worker with late attendance. For instance, the RBS can automatically adjust task assignments by reassigning critical tasks

from late workers to available ones, ensuring minimal disruption to operations. Additionally, the system can trigger notifications to current workers where they get to know that they have been reassign the task for the day. This targeted use of RBS ensures timely decision-making and efficient management, focusing on maintaining productivity and accountability.

2.3 Study of existing related systems

This section discusses the introduction and background study of three existing systems related to oil palm plantation management: Sawit+ [11], IXCHEL Palm Oil Estate Management System (POEMS) [12], and SawitCare [13]. Sawit+ is a plantation management software designed to optimize operational performance and track activities in palm oil estates. It's applied in various oil palm plantations in Southeast Asia to monitor real-time data, track yields, and help in resource management. The functionalities and features of these three systems have been identified and analyzed. Additionally, a comparison between these existing systems and the proposed system has been conducted.

Table 1 *Comparison between the existing systems and PalmCare*

Features	Sawit+	IXCHEL Palm Oil Estate Management System (POEMS)	SawitCare	PalmCare: Smart Mobile Platform for Oil Palm Plantation Management
Sign Up and Login	✓	✓	✓	✓
Manage User	✓	✓	✓	✓
Manage Plantation Map	X	X	✓	✓
Manage Plantation Analytics	X	X	✓	✓
Generate plantation analytics report	X	✓	✓	✓
Track Workers Activity	X	X	X	✓
Update Working Status	X	X	✓	✓
New Worker Training Tutorial	X	X	X	✓
Worker Attendance Record using NFC	X	X	X	✓
Real-Time Weather and Price Update	X	X	X	✓

"✓ = Yes" and "X = No"

PalmCare stands out from other plantation management systems like Sawit+, IXCHEL POEMS, and SawitCare by offering unique and innovative modules. One key feature is its Worker Training Module, which provides step-by-step tutorials, instructional videos, and a ripeness prediction system to train workers in plantation operations, ensuring quality control and operational readiness. PalmCare also integrates Real-Time Fruit Price Updates from the Malaysian Palm Oil Board (MPOB) and combines this with delivery records to provide accurate Estimated Income Calculations for plantation managers. This feature offers a more detailed financial overview compared to other systems.

Additionally, PalmCare includes a Rule-Based System for decision-making, offering data-driven insights tailored to the specific needs of the plantation. These exclusive modules improve efficiency, reduce costs, and address gaps left by competing systems. PalmCare combines advanced capabilities with a user-friendly design at a cost-effective price, distinguishing it as an innovative solution that empowers both management and workers, making it a more comprehensive tool than other systems that focus on basic functionalities or research.

3. Methodology

The development of PalmCare, a mobile application designed for efficient oil palm plantation management, followed the structured phases of the Software Development Life Cycle (SDLC). In the requirement gathering phase, the system scope and functionalities were identified through stakeholder interviews with plantation managers and supervisors. A Gantt chart was also developed to outline the project timeline and guide key development activities.

The analysis and design phase involved conducting a feasibility study covering technical, operational, and economic aspects, followed by the creation of system architecture diagrams, User Interface (UI) /User Experience (UX) mockups, and Unified Modelling Language (UML) models including use case diagram, class diagram, sequence diagram and activity diagram. These artifacts provided a clear technical blueprint for development.

In the implementation phase, the system was built using Flutter for the mobile frontend, Hypertext Markup Language (HTML), Cascading Style Sheets (CSS), JavaScript and Hypertext Preprocessor (PHP) for website and Supabase for backend services, such as authentication and real-time database integration. TensorFlow models trained on image datasets were deployed via TensorFlow Lite to enable palm fruit ripeness detection using the device camera.

During the testing phase, both Functional Testing and System Usability Scale (SUS) based User Acceptance Testing (UAT) were conducted. Functional testing ensured each module met its intended requirements, while SUS UAT gathered usability feedback from real plantation users to improve the user experience.

The deployment phase involved packaging the application as an Android Package Kit (APK), hosting the website and distributing it to workers, supervisors, and managers. Initial user onboarding sessions were held to familiarize users with app functions.

Finally, in the maintenance phase, feedback loops were established to gather user input, bugs were resolved, Artificial Intelligence (AI) models were retrained with new data.

Table 3 *Software development activities and tasks*

Phases	Tasks	Outcome (Deliverable)
Requirement Gathering	• Conduct interviews and surveys with plantation stakeholders. • Observe current workflows and identify pain points. • Collect user needs and expectations. • Analyse existing manual systems.	• List of detailed user requirements. • Initial problem statement.
Analysis and Design	• Define system objectives and scope. • Conduct feasibility studies (technical, operational, economic). • Identify user roles and permissions. • Plan technology stack and tools. • Outline project timeline and milestones. • Design system architecture including frontend, backend, and AI components. • Develop UML diagrams (Use Case Diagram, Class Diagram, Sequence Diagram and Activity Diagram) with General System Architecture. • Create User Interface prototypes for mobile and website interfaces.	• Feasibility report. • System requirements specification. • Project plan with timeline. • System architecture diagrams. • UML diagrams. • User Interface mockups and wireframes. • Approved design and requirement documents.

Table 3 *(Cont)*

Phases	Tasks	Outcome (Deliverable)
Analysis and Design	• Review and validate system design with stakeholders	
Implementation	• Set up development environment. • Develop frontend with Flutter. • Develop backend using Supabase. • Integrate AI models (TensorFlow Lite). • Implement role-based access. • Perform unit and integration testing during development.	• Developed functional application modules. • Integrated AI features. • Source code in version control.
Testing	• Conduct functional testing for all modules. • Perform usability testing (User Acceptance Testing). • Validate AI model accuracy and performance. • Fix bugs and optimize system.	• Tested and validated application. • Test reports. • Issue log and resolutions. • Collect user feedback
Deployment	• Package and deploy APKs to target users. • Conduct user onboarding. • Ensure initial accessibility for plantation workers, supervisors, and managers.	• Deployed application.
Maintenance	• Monitor app performance and stability. • Collect and respond to user feedback. • Perform regular bug fixes, performance tuning, and feature updates. • Continuously improve AI models for fruit ripeness detection.	• Maintenance logs and reports

4. Analysis and Design

This section focuses on the analysis and design phase of PalmCare: Smart Mobile Platform for Oil Palm Plantation Management. Using Unified Modeling Language (UML) diagrams, it visualizes system interactions and functionalities. Additionally, it details the database design and interface design for user interactions, providing a foundation for the application's development and implementation.

Applied Information Technology and Computer Science Vol. 6 No. 2 (2025) p. 868-887

4.1 System Requirements Analysis

This section outlines the functional, non-functional, and user requirements for the proposed system. Functional requirements specify the tasks the system must perform for users, while non-functional requirements encompass the system's behavioral properties. The functional requirements, non-functional requirements and user requirement analysis are detailed in **Table A.1**, **Table A.2 and Table A.3** in Appendix A.

4.2 Unified Modelling Language (UML)

As a widely adopted tool, Unified Modelling Language (UML) enhances software design, development, and documentation efficiency through its standardized and versatile approach [14]. The detailed activity diagram and sequence diagram of Update Working Status module following by use case diagram, class diagram of PalmCare are shown below.

4.2.1 Use Case Diagram

The use case diagram for PalmCare: Smart Mobile Platform for Oil Palm Plantation Management provides a detailed representation of the interactions between the 3 actors (Manager, Supervisor and Current Worker) and the 9 use cases. This diagram helps to identify the different roles users play and the specific functionalities they interact with, ensuring all user requirements are clearly defined and addressed. **Fig. 1** shows the case diagram of PalmCare which are the interactions between the system's functions.

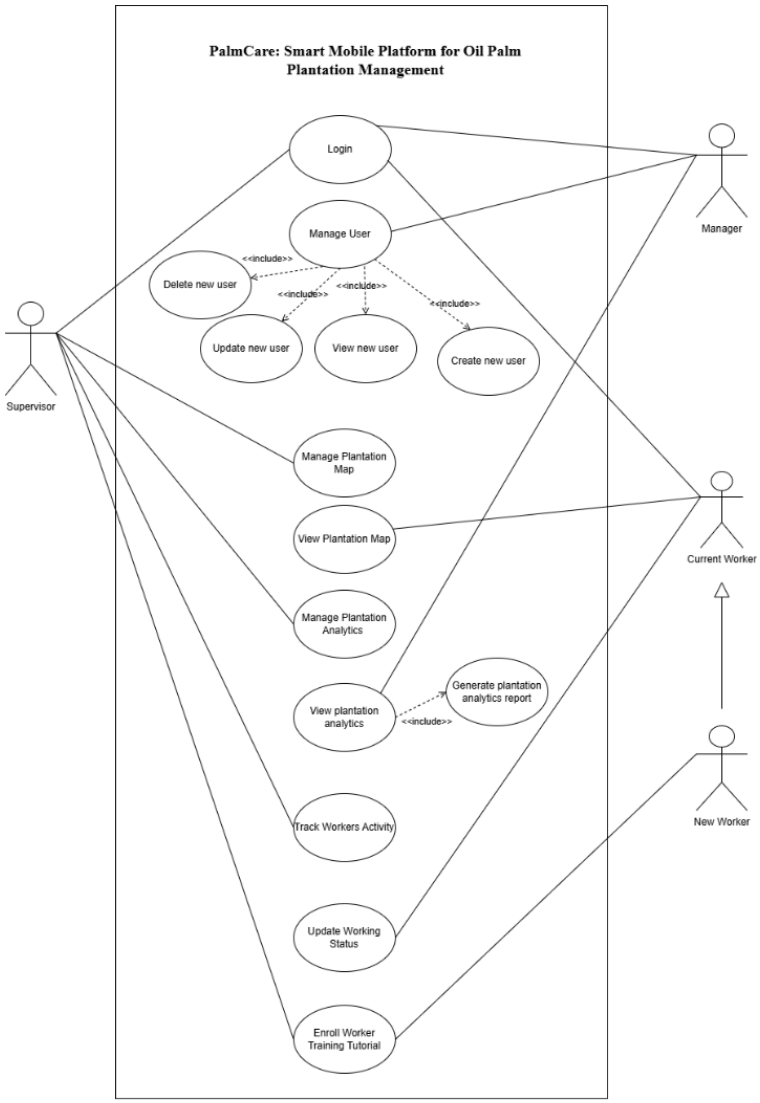

Fig. 1 *Use case diagram of PalmCare*

4.2.2 Behavior Model

The activity diagram details the process workers follow to update their working status, including attendance and task reporting using NFC. Workers start by accessing the attendance page, where they scan their NFC tag or opt for "Manual Attendance" if NFC issues arise. Late clock-ins (after 10 AM) are flagged as late entries, and tasks may be reassigned accordingly. After attendance, workers are redirected to the plantation map to view assigned blocks and tasks. In emergencies, workers notify their supervisor for appropriate action. Task reporting involves scanning the NFC tag or manually selecting the block, specifying completed work, and filling in required details before submitting updates. This system ensures accurate attendance tracking and task reporting while addressing late entries and emergencies efficiently. The activity diagram and sequence diagram of update working status module is detailed in **Fig. A.1** and **Fig. A.2** in Appendix A.

4.2.3 Class Diagram

The class diagram is a graphical representation of the database conceptual model. It highlights the 16 core entities and their relationship with the developed system. It involves mainly student and staff classes, which are inherited from the user class. The PalmCare class diagram is detailed in **Fig A.3** in Appendix A.

4.3 General System Architecture

The architecture depicted follows a three-tier model with distinct layers of Presentation Tier**,** Application Tier, and Data Tier. The three-tier model is to achieve the goals of building a modular, secure, and role-based plantation management system. The Presentation Tier includes user-facing interfaces like the App UI (for supervisors and workers) and Web UI (for managers). The Application Tier processes user inputs through Controllers and Models, linking the interface with backend logic, while the Data Tier utilizes a Supabase database to store and manage system data, accessed securely via APIs. This system architecture is designed for a plantation management system to streamline activities, resources, and data. It supports multiple user roles Supervisors, Workers, and Managers providing tailored interfaces and functionalities such as user management, plantation analytics, worker tracking, and training modules. Business logic and data storage are integrated through Supabase, ensuring secure operations and efficient data handling. This design promotes scalability, security, and intuitive user experience. **Fig A.4** shows the general system architecture of PalmCare in Appendix A.

4.4 Interface Design

The interface design for the PalmCare: Smart Mobile Platform for Oil Palm Plantation Management System consists of a mobile application and a web-based system tailored to different user roles. The web-based system developed using HTML, CSS, JavaScript, and PHP, is designed specifically for plantation managers. It provides access to key administrative features such as user account management, worker monitoring, delivery tracking, data analytics, and report generation. The web system emphasizes accessibility, data visibility, and efficiency for high-level decision-making. The mobile application was developed using the Flutter Framework, ensuring a responsive and user-friendly experience for both Android and iOS users. It allows supervisors and workers to manage tasks, perform NFC-based attendance and activity tracking, and conduct AI-powered fruit ripeness predictions.

4.4.1 Web Interface Design for Plantation Manager

Fig.2 login interface serves as the entry point for Manager to access the PalmCare system. It includes User Authentication where it allows Manager to log into the system securely using the username and password. The interface is clean and simple, asking the manager for basic credentials.

In **Fig.3**, the dashboard interface acts as the central control and monitoring hub for PalmCare, providing an overview of key performance metrics and system data. It tracks essential data such as the Worker Management section displays the total number of workers, helping manager to monitor labour resources. It distinguishes between active and inactive workers. Harvest Tracking section displays the harvest target and progress toward achieving it, allowing manager to track how much palm oil has been harvested and whether the target is being met. Weather Analytics section shows real-time weather conditions such as temperature, humidity, rain intensity helping manager adjust plantation operations based on environmental factors. Performance Metrics section tracks the number of tasks completed and the total deliveries made, providing insights into plantation activities and logistics. It provides a centralized overview of operations, making it easier to monitor all critical aspects of plantation management. Help manager make data-driven decisions. Allows real-time tracking of key performance metrics, ensuring timely responses to any operational challenges.

In **Fig.4**, The Worker Management Interface is designed to streamline workers data handling by allowing manager/administrators to add, view, edit and delete worker details, including their role and contact information. It also enables the updating of worker status where promoting current workers to supervisor and supports filtering workers based on specific attributes for easier management. It enhances workforce management by centralizing worker information, making it easier to access and update records. The ability to filter workers based on different attributes saves time and increases the efficiency of processes done.

Finally, on **Fig.5** the Delivery Details Interface manages and tracks the oil palm fruit delivered from the plantation. It provides delivery information such as receipt reference numbers, weight, and dates with the option to view receipts and export data in CSV. This interface aids in maintaining accurate delivery records, supporting smooth logistics, and ensuring timely shipments. Together, these interfaces support the efficient operation of PalmCare, allowing for seamless management of workers, deliveries, and plantation performance metrics.

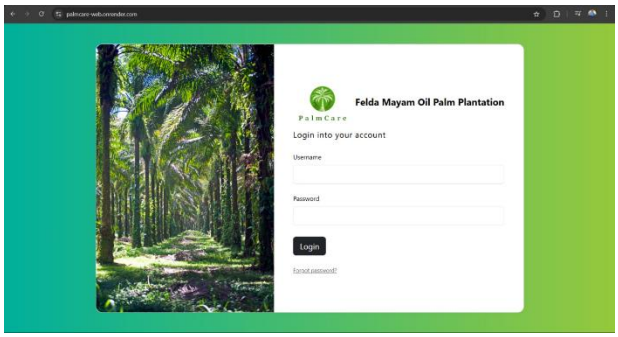

Fig.2 Login interface

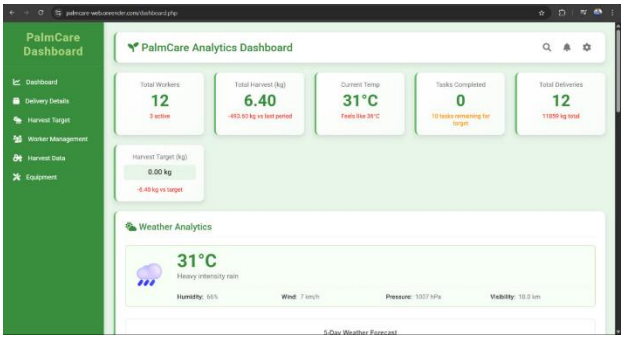

Fig.3 Dashboard interface

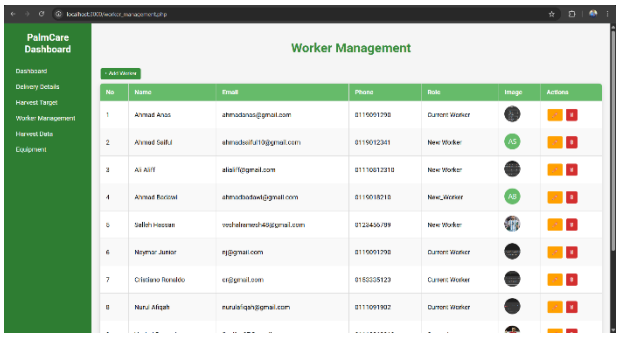

Fig.4 Worker Management interface

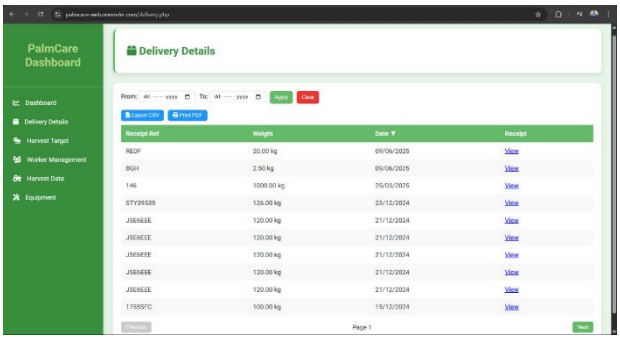

Fig.5 Delivery Details interface

4.4.2 Mobile Application Interface Design for Supervisor

The Supervisor Dashboard Interface provides a comprehensive overview of essential plantation data and offers additional functionality to support plantation management as shown in **Fig.6**. Harvest Summary section displays the total weight of oil palm fruits harvested in ton for the current period, helping the supervisor track production progress and determine whether the harvest goals are being met.

In **Fig.7**, the Plantation Map Interface offers a visual representation of the plantation, enabling the supervisor to manage and monitor plantation activities efficiently. It provides several key features such as Navigate Plantation Areas where the map displays the layout of the plantation, including different blocks and trees, allowing the supervisor to monitor specific areas and track the status of ongoing activities across the plantation.

The Activity Interface tracks all activities performed on the plantation, including tasks such as harvesting, fertilizing, and pruning as shown in **Fig.8**. It provides Activity Summary which is a quick view of how many trees have been harvested, fertilized, and pruned, allowing supervisors to track overall progress. Detailed Activity Log section lists specific activities by worker, type of task, tree, date and time which picture proof ensuring accountability and proper scheduling.

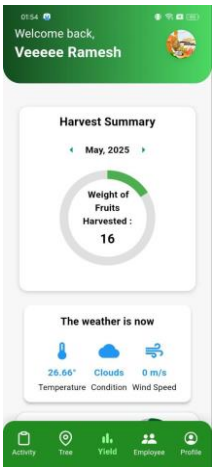

Fig.6 Dashboard interface

Fig.7 Plantation Map interface

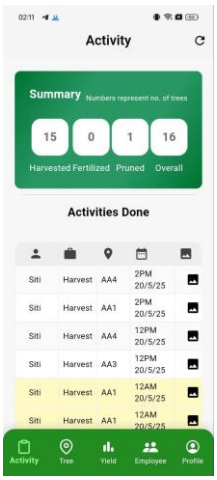

Fig.8 Activity interface

4.4.3 Mobile Application Interface Design for Current Worker

In **Fig.9**, The Current Worker Dashboard Interface is the main screen that workers interact with. It provides workers with several key options for managing their daily tasks. Workers can mark their attendance for the day using the NFC scanning functionality, ensuring that they are properly recorded as present on the plantation.

The Worker Attendance Using NFC Interface is used to register a worker's attendance by scanning an NFC tag as shown in **Fig.10**. The worker places their phone near an NFC tag to mark their attendance for clock in and clock out. This ensures that the system automatically records their presence at the plantation without requiring manual input. If the NFC scanning fails or is unavailable, workers can manually mark their attendance through the alternative option Manual Attendance.

In **Fig.11**, the Activity Interface allows workers to track their daily tasks and activities. Activity Summary which located at the top, the worker can see a summary of completed tasks, including the number of trees that have been harvested, fertilized, and pruned, with an overall total count. As for the Detailed Activity Log section, below the summary, the interface displays a log of activities performed which is worker, type of task, tree, date and time which picture proof. This helps ensure that workers stay on task and that supervisors can easily track completed activities.

In **Fig.12**, The Worker Report Using NFC Interface enables workers to scan NFC tags of blocks to log their work activities. The worker places their phone near the NFC tag on the block NFC reader they are working on to record the specific task. This ensures that the activity is logged accurately and tied to the correct location and task. If NFC scanning is not available or functional, workers can manually report their activities by selecting Manual Report option.

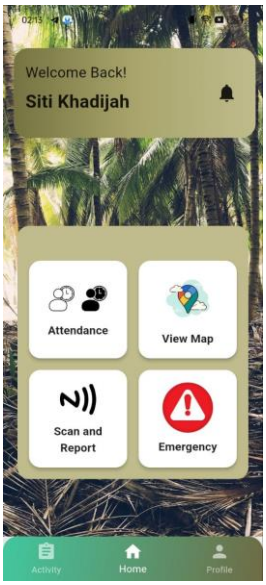

Fig.9 Current Worker dashboard interface

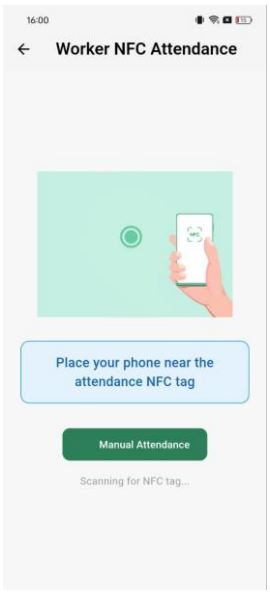

Fig.10 Worker Attendance using NFC interface

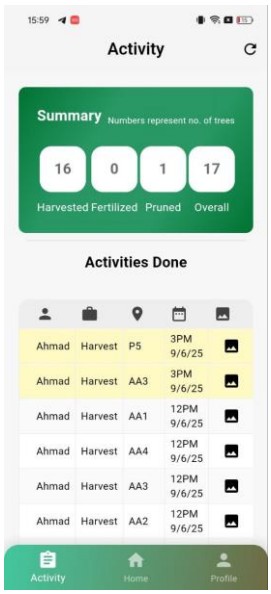

Fig.11 Activity interface

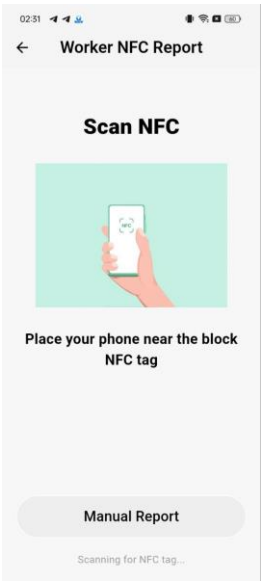

Fig.12 Worker Report using NFC interface

4.4.4 Mobile Application Interface Design for New Worker

In **Fig.13**, The New Worker Training Interface is part of the training module for workers. This screen is designed to guide new workers through a series of training steps. The first step of the training involves an introduction to oil palm cultivation basics.

In **Fig.14**, the AI Ripeness Prediction Interface uses artificial intelligence (AI) to assist workers in identifying the ripeness of palm fruits, which is critical for the harvesting process. Workers are taught to assess whether the palm fruits are ripe or not. This interface provides a practical training session where workers can test their ability to assess ripeness by either using the camera or selecting an image from the gallery. The "Practice Ripeness Assessment" button likely activates the feature where the worker practices identifying ripe fruits. This section leverages AI to predict whether a palm fruit is ripe based on the image uploaded or taken with the camera.

In **Fig.15**, the Ripeness Prediction Feedback Interface shows the results of the AI's analysis on the ripeness of a palm fruit. The interface shows a picture of a palm fruit with the ripeness prediction result, including the percentage confidence, 60.1% confidence that the fruit is ripe. This helps workers verify their assessment and learn how to make more accurate predictions over time. Workers can either take a new picture of a palm fruit using the Camera or use an existing image from the Gallery to check the ripeness of the fruit.

Fig.13 New Worker Training interface

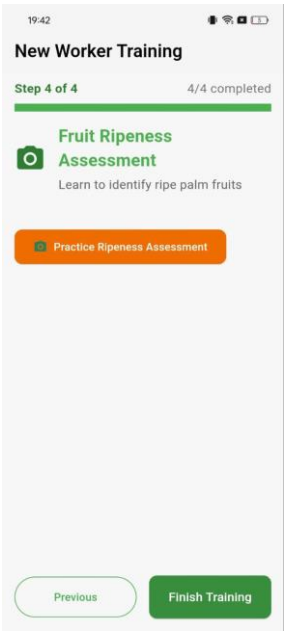

Fig.14 AI Ripeness Prediction interface

Fig.15 Ripeness Prediction feedback interface

5. Results and Discussion

The PalmCare system was developed utilizing a variety of modern technologies and platforms selected for their usability, performance, and cost-effectiveness. The mobile app was built with the Flutter Framework using Dart for cross-platform compatibility. The backend infrastructure was built on Supabase, a PostgreSQL-based database with real-time database management, authentication, and secure storage services and Firebase for real-time notifications. The main IDE for development and testing was Visual Studio Code, while Google Colab was used to build and test AI models for predicting fruit freshness. These models were converted to TensorFlow Lite (TFLite) format and implemented into the app. The combination of these technologies ensures a scalable and efficient application tailored to address the needs of small-scale oil palm plantation like PalmCare. The system includes essential modules such as Worker Report and Attendance via NFC, Task Assignment, Plantation Mapping, Activity Tracking, Fruit Ripeness Detection, and Training Modules. Git and GitHub were used to maintain version control, which allowed for collaborative development. These modules were designed to streamline operations, improve management efficiency, and support effective worker management.

5.1 Testing

For the PalmCare: Smart Mobile Platform for Oil Palm Plantation Management, testing was carried out through Functional Testing as shown in Table 2 and System Usability Scale (SUS) User Acceptance Testing.

Table 2 *Test Case Result of Manage Plantation Map Module*

Test Case ID	Description	Expected Result	Actual Result	Pass/Fail
TC003-01	The system shall allow Supervisors to create a new plantation block by drawing boundaries on the map.	New block is created and saved with visible boundaries on the map.	New block is created and saved with visible boundaries on the map.	Pass
TC003-02	The system shall validate boundary input (e.g., no overlapping with existing blocks).	System prevents saving and shows error if boundaries overlap or are invalid.	System prevents saving and shows error if boundaries overlap or are invalid.	Pass
TC003-03	The system shall allow Supervisors to edit/update existing plantation block details.	Changes are saved and reflected immediately on the map.	Changes are saved and reflected immediately on the map.	Pass
TC003-04	The system shall allow Supervisors to add new trees to a selected block on the map.	New tree markers appear on the map within the selected block.	New tree markers appear on the map within the selected block.	Pass
TC003-05	The system shall enable task assignment based on specific blocks to selected workers.	Task is saved and worker receives the task tied to the correct plantation block.	Task is saved and worker receives the task tied to the correct plantation block.	Pass
TC003-06	The system shall allow NFC tags to be written for plantation blocks at NFC stations.	NFC tag is successfully written with block ID and can be scanned for reporting.	NFC tag is successfully written with block ID and can be scanned for reporting.	Pass
TC003-07	The system shall send real-time notifications to workers when tasks are assigned or updated.	Assigned workers receive notifications immediately on their mobile devices.	Assigned workers receive notifications immediately on their mobile devices.	Pass
TC003-08	The system shall allow zooming and rotating the map for better usability.	User can smoothly interact with the map view.	User can smoothly interact with the map view.	Pass

5.1.1 User Acceptance Testing (UAT)

User Acceptance Testing (UAT) for the PalmCare: Smart Mobile Platform for Oil Palm Plantation Management was conducted to validate its functionality, usability, and relevance in a real-world plantation environment. The testing involved 14 participants, including a Plantation Manager, Supervisor, Current Workers, and New Workers. The UAT was structured using a Google Form consisting of four main sections: Demographic Information, Functional Testing, UI/UX Evaluation, and the System Usability Scale (SUS).

The outcome of the functional feedback testing for the Plantation Manager. The results reflect the completion status of various key functionalities based on system testing. Based on the results, 81.8% of functionalities are fully functional and meet the required standard, which reflects a high level of satisfaction with usability and functionality. Meanwhile, 18.2% of functionalities showing they need further improvement. These features are essential for monitoring productivity and making informed decisions, and their current limitations could hinder effective management. The test also revealed that while the core system is stable and operational, enhancements in analytics and reporting are necessary to optimize the user experience for the Plantation Manager. Overall, the feedback indicates a strong foundation with targeted areas for improvement to fully support managerial decision-making processes. All criteria have passed the test indicating that the system meets the required standard as shown in **Fig.16**.

Fig.16 Functional Feedback Plantation Manager

The outcome of the functional feedback testing for the Supervisor role in the plantation management system. The chart evaluates the completion status of multiple functionalities, indicating whether each is fully implemented or still requires improvement. Based on the results, 81.8% of functionalities are fully functional and meet the required standard, which reflects a high level of satisfaction with usability and functionality. Meanwhile, 18.2% of functionalities showing they need further improvement. These areas are important for effective worker supervision and daily task management, and their current limitations highlight the need for further development to enhance system performance. Overall, all criteria have passed the test. It indicates that the system meets the required standard. The results indicate that while most core functionalities are in place, improvements are still necessary to ensure more seamless and efficient supervisory experience as shown in **Fig.17** below.

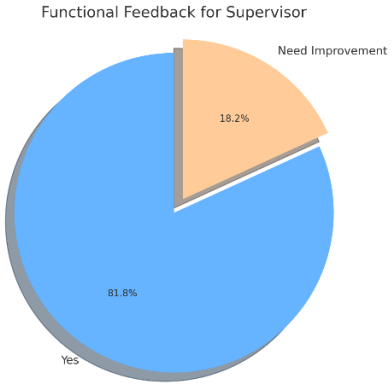

Fig.17 Functional Feedback Supervisor

The outcome of the functional feedback testing for the Current Worker role in the plantation management system. Based on the results, 97.2% of functionalities are fully functional and meet the required standard, which reflects a high level of satisfaction with usability and functionality. Meanwhile, 2.8% of functionalities showing they need further improvement. This includes features related to task reporting, NFC usage, manual reporting,

viewing assigned work, and notification handling. The consistent full scores reflect a high level of satisfaction with the system's usability and reliability from the worker's perspective. Overall, the results suggest that the system successfully supports the day-to-day operational needs of workers, providing a smooth and effective user experience as shown in **Fig.18** below.

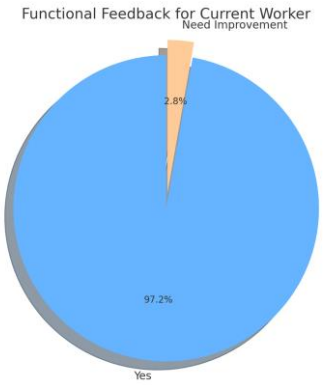

Fig.18 Average Functional Feedback Current Worker

The outcome of the functional feedback testing for the New Worker role in the plantation management system. Based on the results, 95.4% of functionalities are fully functional and meet the required standard, which reflects a high level of satisfaction with usability and functionality. Meanwhile, 4.6% of functionalities showing they need further improvement. The high average scores reflect positive user experience, with minor adjustments needed for a few functionalities. Overall, the results indicate that the system effectively supports New Workers, meeting most operational needs while offering opportunities for further refinement as shown in **Fig. 19** below.

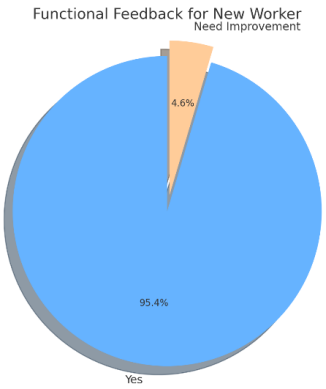

Fig.19 Average Functional Feedback New Worker

The overall average System Usability Scale (SUS) mean score is 75.0 over 100, which indicates good usability, as shown in **Fig. 20** below. This mean score suggests that the system is generally easy to use and efficient, though there may still be areas for improvement to enhance the user experience. This score is calculated using SUS feedback form receive from every 14 users that are working at oil palm plantation. This suggests that users generally find the system easy to use and efficient. However, there are likely areas where improvements can be made to further enhance the user experience and increase overall satisfaction. While the system is functional and meets the basic expectations of its users, optimizing certain features or addressing specific pain points could help elevate the usability to a higher level, making the system even more intuitive and user-friendly.

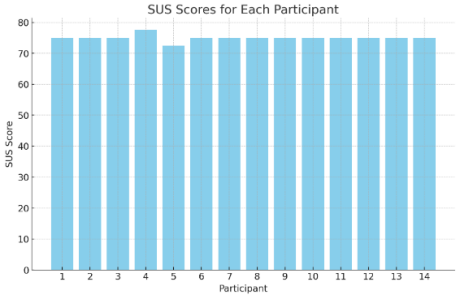

Fig.20 System Usability Scale (SUS) Scores for 14 participants

6. Conclusion

In summary, PalmCare addresses key issues in plantation management, including manual data handling, inefficient labor allocation, delayed and inaccurate task reporting, and the absence of real-time tracking systems. These challenges contribute to poor decision-making and reduced productivity. The objective of this project is to design PalmCare using an object-oriented approach, develop it as a mobile application, and evaluate it through User Acceptance Testing. Key features include real-time data collection, task reporting via Near Field Communication (NFC) technology, rule-based labor allocation, and structured training modules for new workers.

For future work, drone integration for aerial monitoring is an important enhancement for PalmCare. By incorporating drones, the system can conduct aerial mapping, assess tree health, and identify underperforming areas within the plantation. This allows for a high-level overview of the plantation's performance, providing valuable insights into the overall health of the crop and enabling more efficient resource allocation. Besides, the in-app chat module for PalmCare is a design to enhance communication and coordination between plantation workers, supervisors, and managers. By enabling real-time messaging, the module ensures that updates, instructions, and issues can be addressed instantly, even in remote areas where traditional communication methods may not be as effective. This feature supports both one-on-one conversations and group chats, allowing teams or departments to stay aligned and collaborate efficiently.

PalmCare plays a transformative role in modernizing the oil palm industry by providing a comprehensive digital management system tailored to the unique challenges faced by plantation owners, supervisors, and workers. Furthermore, PalmCare bridges the digital divide in rural plantations by offering a user-friendly, mobile-first platform, promoting the adoption of smart agriculture practices.

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 interest regarding the publication of the paper.

Author Contribution

The authors confirm contribution to the paper as follows: ***study conception and design:*** *Veshal A/L Ramesh, Nur Liyana Binti Sulaiman;* ***data collection:*** *Veshal A/L Ramesh, Nur Liyana Binti Sulaiman;* ***analysis and interpretation of results:*** *Veshal A/L Ramesh, Nur Liyana Binti Sulaiman;* ***draft manuscript preparation:*** *Veshal A/L Ramesh, Nur Liyana Binti Sulaiman. All authors reviewed the results and approved the final version of the manuscript.*

References

[1] Sergieieva, K. (2024, June 28). *Oil Palm Plantation: Cultivation & management tips for Growers*. EOS Data Analytics. https://eos.com/blog/oil-palm-plantation/

[2] Ritchie, H., & Roser, M. (2024, January 15). *Palm Oil*. Our World in Data. https://ourworldindata.org/palm-oil

[3] Sulaiman, N., & Wahab, M. H. A. (2018). The role of ICT in the Malaysian palm oil industry: Enhancing plantation productivity. *Journal of Information Systems and Technology Management*, *15*(1), 1–15. https://doi.org/10.4301/S1807-1775201815001

[4] AyaGrow. (2022). *Smart plantation management: Optimizing oil palm yield through digital platforms*. https://www.ayagrow.com

[5] Roundtable on Sustainable Palm Oil (RSPO). *PRISMA: PalmTrace RSPO Information System Management Application*. https://rspo.org

[6] Abdullah, R., Ismail, R., & Ahmad, N. (2021). *Management Information Systems in agriculture: Enhancing efficiency and sustainability*. Journal of Agricultural Informatics, 12(2), 45–58. https://doi.org/10.1234/jai.v12i2.5678

[7] Zhao, C., & Anisha Kothapa, A. (2024, August 16). *Report: Supabase Business Breakdown & Founding Story*. Contrary Research. https://research.contrary.com/company/supabase

[8] Lienau, J. (2022, April 29). *What is NFC? explaining near-field communication technology*. What Is NFC? Explaining Near-Field Communication Technology. https://onekeyresources.milwaukeetool.com/en/what-is-nfc

[9] Russel, S., & Norvig, P. (2016). Artificial intelligence: A modern approach, global edition. *Harlow: Pearson*.

[10] Wang, Y., Sun, Z., & Chen, Z. (2019). Development of energy management system based on a rule-based power distribution strategy for hybrid power sources. *Energy*, *175*, 1055–1066. https://doi.org/10.1016/J.ENERGY.2019.03.155

[11] Sawit+ (Version 6.2.2) [Mobile app]. Excelfixer Sdn Bhd. https://apps.apple.com/my/app/sawit/id1574634785

[12] IXCHEL POEMS (Version 3.1) [Website]. IXCHEL Technologies. https://www.ixchelent.com/web/palm.html

[13] SawitCare (Version 1.2) [Mobile app]. SawitCare Solutions.

[14] Osis, J., & Donins, U. (2017). A Standard for Designing a Software. *TopUML Modeling*, 3–51. https://doi.org/10.1016/B978-0-12-805476-5.00001-0

Appendix A:

Table A.1 *Functional Requirement*

Modules	Functionalities
Login	1. The system shall allow the user to login with a correct email and password. 2. The system shall redirect the user to the specific page based on their roles. 3. The system shall show an error message if a text field is left blank or if there is an invalid login attempt. 4. The system shall allow the user to change their password to the forget password page. 5. The system shall update the database with the new password if the user forgets their password.
Manage User	1. The system must allow the Manager to add new users by entering plantation, first name, last name, email, phone number, and role. 2. The system must validate all fields, ensuring no fields are empty and the email format is correct before saving. 3. The system must display an error message if validation fails and allow the Manager to re-enter the correct details. 4. The system must allow the Manager to edit existing user details, including plantation, name, email, phone number, and role. 5. The system must enable the Manager to delete users and display a confirmation message once completed.
Manage Plantation Map	1. The system must enable Supervisors to create new plantation blocks by defining boundaries on a map. 2. The system must allow Supervisors to edit or update the details and boundaries of existing plantation blocks. 3. The system must enable Supervisors to add new tree at specific block. 4. The system must allow Supervisors to edit or update the details of existing plantation trees. 5. The system must allow Supervisors to assign unique identifiers (IDs) to each plantation block and tree for easy reference. 6. The system must integrate task assignment functionality, enabling Supervisors to assign tasks to employees based on specific blocks.

Table A.1 *(Cont)*

Modules	Functionalities
Manage Plantation Map	7. The system must enable Supervisors to write NFC tags for each plantation block at designated NFC stations. 8. The system must provide real-time notifications to workers regarding their assigned tasks and any schedule changes.
View Plantation Map	1. The system shall allow employees to view the plantation map. 2. The system must retrieve and display interactive plantation maps with task-specific overlays. 3. The system must allow workers to view task details linked to specific blocks. 4. The system shall display detailed information of selected blocks and trees.
Manage Plantation Analytics	1. The system shall allow the Supervisor to view harvest summaries with detailed breakdowns. 2. The system shall fetch and display real-time weather, humidity and temperature data. 3. The system shall enable the Supervisor to set, track, and overwrite harvest targets. 4. The system shall allow Supervisor to add delivery records, including proofs. 5. The system shall calculate estimated income based on delivery records. 6. The system shall fetch and show daily fruit price updates from Malaysian Palm Oil Board (MPOB).
View Plantation Analytics Report	1. The system shall display a real-time plantation analytics dashboard with metrics: Harvest Summary, Harvest Target, Weather Data, Delivery Records, Fruit Price Updates, and Estimated Income. 2. The system shall fetch real-time weather and fruit price updates from external sources such as Weather and Malaysian Palm Oil Board (MPOB). 3. The system shall provide the "Generate Report" feature for the Manager to compile analytics data into a formatted PDF document. 4. The system shall allow generating customized plantation reports. 5. The system shall allow the Manager to download or preview the generated report.
Track Workers Activity	1. The system must allow Supervisors to view a summary of work types performed by workers. 2. The system must display detailed activity reports, including worker names, task locations, and photo proof. 3. The system must highlight tasks reported manually in red. 4. The system must provide real-time updates on task completion reported through NFC and manual entries. 5. The system must support filtering and sorting activity data by various criteria.
Update Working Status	1. The system must enable workers to take attendance through NFC at the beginning of the day. 2. The system shall support manual attendance entry when NFC got problem. 3. The system must allow workers to view assigned blocks, trees, and tasks via the plantation map. 4. The system shall support NFC-based task reporting and upload picture proof for verification. 5. The system shall allow manual report submission when NFC got problem. 6. The Rule-Based System shall reassign tasks to available workers if a worker with late status is unable to perform assigned tasks.

Table A.1 *(Cont)*

Modules	Functionalities
Enroll Worker Training Module	1. The system must provide a structured learning path for new workers. 2. The system shall save the worker's progress automatically if they exit the training module mid-session. 3. The system must lock subsequent tutorials until the current one is marked as 'Done.' 4. The system must enable ripeness prediction training using image uploads. 5. The system shall prompt the worker to retry image uploads in case of failure during the ripeness prediction task. 6. The system must provide immediate feedback on ripeness levels (underripe, ripe, or overripe) during training. 7. The system must notify supervisors upon training completion.

Table A.2 *Non-Functional Requirement*

Modules	Functionalities
Performance	1. The system should operate effectively under anticipated load conditions, ensuring quick response times for both user interactions and data processing. 2. The NFC scanning feature must respond within 1 second of interaction.
Security	1. Robust security measures must be in place, including data encryption, secure login processes, and protection against security threats. With Supabase, the system should utilize the encryption for data in transit, secure user authentication mechanisms, and regular updates to safeguard against vulnerabilities. 2. Role-based access control must ensure that workers, supervisors, and managers can only access authorized data and functions.
Reliability	1. The system must ensure high availability, guaranteeing that services are consistently accessible to users without interruptions.
Usability	1. The system must provide a user-friendly interface, guaranteeing straightforward navigation and interaction for all types of users. 2. The plantation map should support zoom, pan, and detailed annotations for better user interaction.
Integration	1. The system must integrate seamlessly with the Malaysian Palm Oil Board (MPOB) to fetch daily fruit prices. 2. The plantation map module must use Google Maps APIs for accurate geospatial representation. 3. Data from NFC reporting stations must synchronize in real-time with the centralized database.
Compatibility	1. The application must run on Android devices with OS version 8.0 and above. 2. The web-based manager interface must support the latest versions of Chrome, Edge, and Firefox browsers.

Table A.3 *User Requirement Analysis*

No	Description
1	Manager, Supervisor, Current Worker and New Worker must be able to log in securely with role-specific access. All the users should also be able to reset passwords if needed.
2	Manager must be able to add, edit, delete, and view user profiles. Manager must be able to set role assignments for different user types.
3	Manager should be able to monitors plantation performance and generates reports.

Applied Information Technology and Computer Science Vol. 6 No. 2 (2025) p. 868-887

Table A.3 *(Cont)*

No	Description
4	Manager and Supervisor should be able to view daily fruit price updates from Malaysian Palm Oil Board (MPOB).
5	Supervisor should be able to add, update and delete blocks, trees at plantation map.
6	Supervisor able to assign unique identifiers (IDs) and label it to each plantation block and tree.
7	Supervisors should be able to write NFC tags for each plantation block at designated NFC stations.
8	The supervisor should be able to assign tasks to workers and monitor their completion status via activity logs and NFC-based reporting. Responds to emergency notifications raised by workers.
9	Supervisor able to tracks worker attendance using NFC scans or manual submissions when NFC is unavailable.
10	Current Worker able to mark attendance daily using NFC tags or submits photo proof if NFC systems are unavailable.
11	Current Worker should be able to access assigned tasks through the plantation map and updates work status after task completion via NFC-based reporting.
12	Current Worker able to report emergencies through the system to alert supervisors immediately.
13	Current Worker should be able to receives notifications for task updates from system.
14	New Worker should be able to complete structured training that includes video tutorials, written instructions on tasks like harvesting.
15	New Worker able to practice ripeness prediction by uploading fruit images and refining their understanding through system feedback.
16	Manager and Supervisor able to view harvest summaries with detailed breakdowns.
17	Manager and Supervisor able to display real-time weather, humidity and temperature data.
18	Supervisor able to set, track, and overwrite harvest targets.
19	Supervisor should be able to add delivery records, including proofs.
20	Manager and Supervisor able to view estimated income based on delivery records.

Fig. A.1 *Activity Diagram of Update Working Status*

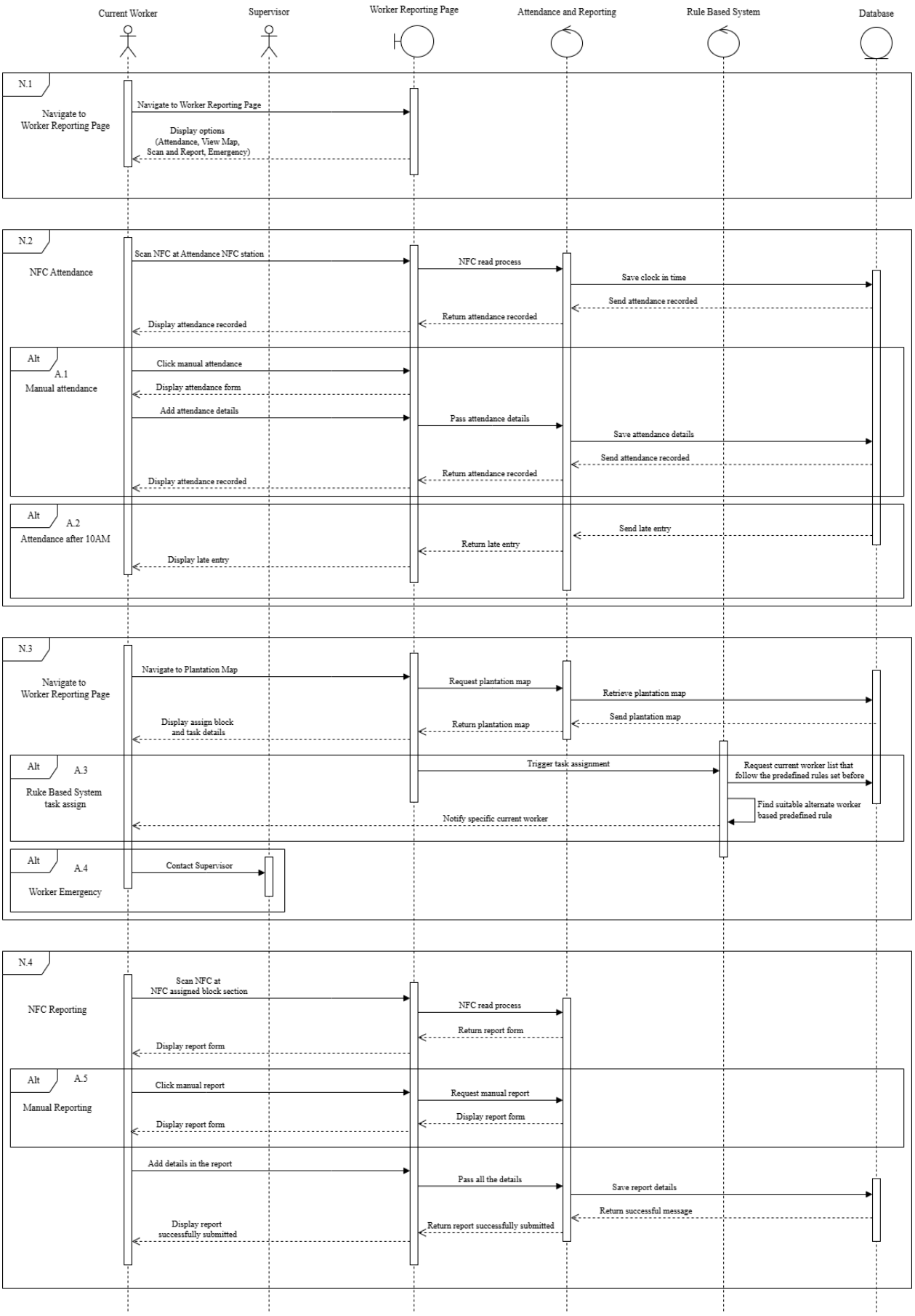

Fig. A.2 *Sequence Diagram of Update Working Status*

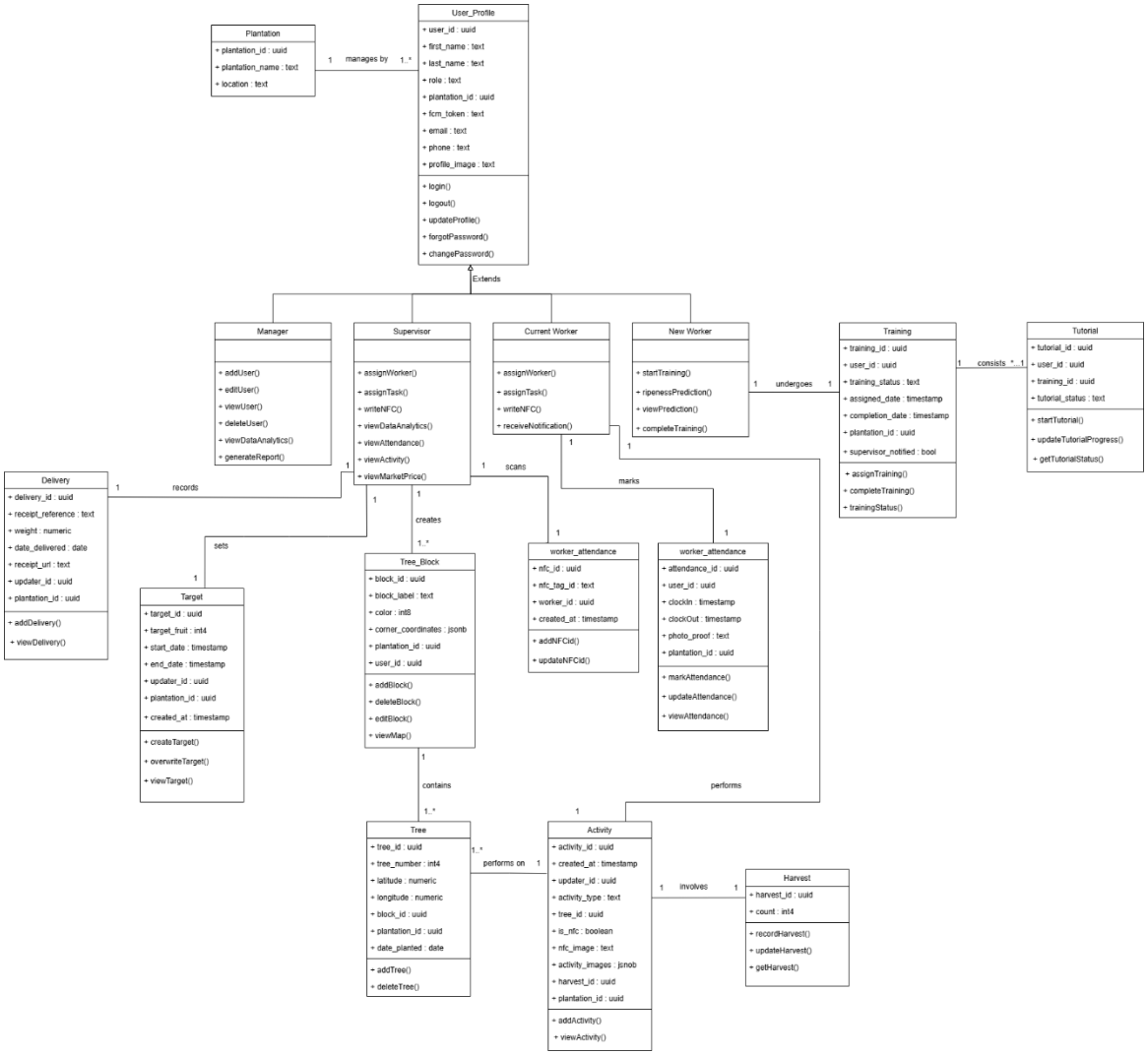

Fig. A.3 *Class Diagram of PalmCare*

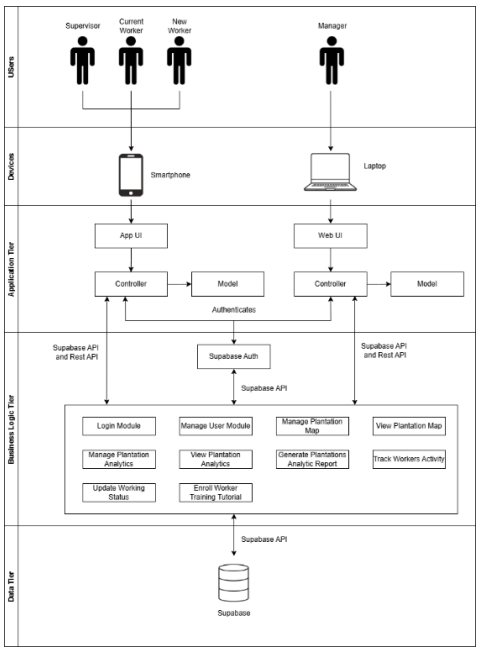

Fig. A.4 *General System Architecture of PalmCare*