

Veronica Enterprise Oil Palm Plantation Worker Management System

Eshra Novera Sandrah¹, Nurezayana Zainal^{1*}

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

*Corresponding Author: nurezayana@uthm.edu.my
DOI: <https://doi.org/10.30880/aitcs.2026.07.01.041>

Article Info

Received: 15 June 2025
Accepted: 21 June 2026
Available online: 30 June 2026

Keywords

Worker management, Oil palm plantation, QR code technology, web-based application, prototyping model.

Abstract

The Veronica Enterprise Oil Palm Plantation Worker Management System is a web-based application developed to improve the efficiency of managing plantation workforce operations. This project aims to resolve existing challenges caused by manual methods, such as delays in task assignment, inaccurate attendance records, and lack of real-time monitoring. The objectives of this project are to design the system using an object-oriented approach, to develop a comprehensive web-based platform for managing worker information and daily activities, and to test the system's functionality and usability through alpha and beta testing phases. The system will use a web-based approach, integrating Quick Response (QR) code technology for attendance tracking, task scheduling modules, and real-time activity monitoring, combined with report generation and staff management features. The prototyping methodology will be applied throughout the development process, enabling continuous refinement based on user feedback. Through this system, Veronica Enterprise is expected to improve task organization, reduce manual workload, and gain better insights for decision-making based on accurate and centralized data.

1. Introduction

Farm operations management plays a vital role in ensuring smooth workflows within agricultural enterprises. It involves managing day-to-day tasks such as worker scheduling, task assignments, attendance tracking, and monitoring of productivity. Effective handling of these responsibilities contributes to better resource use, reduced waste, and improved overall efficiency. In oil palm plantations, where multiple workers and activities must be coordinated daily, streamlined operations are essential for maintaining high productivity and profitability [1].

At Veronica Enterprise, an oil palm plantation managed by Mrs. Veronica in Tenom, these farm operations are currently performed manually. Workers carry out tasks such as fertilizing, harvesting, cleaning palm fronds, and transporting harvested fruits to the factory. Attendance is tracked using a logbook, and task completion is verified manually. This paper-based system lacks centralized data storage and is highly dependent on human input, making it prone to inefficiencies and errors.

Several key issues have been identified with the existing system. Firstly, managing several plantation sections without a real-time monitoring tool consumes a great deal of time and effort. Secondly, manual attendance records often contain inaccuracies, impacting payroll calculations and administrative planning.

This is an open access article under the CC BY-NC-SA 4.0 license.



Thirdly, task assignments are not efficiently managed, with no digital tool to inform workers of their daily duties which often lead to miscommunication, task delays, and lower productivity.

To address these challenges, this project proposes the development of a web-based Oil Palm Plantation Worker Management System that digitizes and centralizes daily operations. The system will feature modules such as login and signup, profile management, task scheduling, Quick Response (QR) code-based attendance tracking, real-time notifications, report generation, and staff records management. The objectives of this project are threefold: first, to design the system using an object-oriented approach; second, to develop a comprehensive web-based solution to manage worker-related tasks and activities; and third, to test the system's usability and functionality through alpha and beta testing phases. By achieving these goals, the system is expected to streamline plantation management, reduce manual errors, and improve overall operational visibility and decision-making [1].

2. Related Work

2.1 Overview of Oil Palm Plantation

Oil palm plantations play a significant role in global agriculture, particularly in countries like Indonesia and Malaysia, which are the largest producers of palm oil globally [2]. These plantations are essential not only for the production of palm oil but also for supporting local economies through employment and trade. In Malaysia, where Veronica Enterprise operates, oil palm cultivation is a key component of the agricultural sector, driving both the domestic economy and export trade.

Veronica Enterprise, as an oil palm plantation, operates in a highly competitive and environmentally sensitive industry. The company is tasked with managing large-scale plantation operations that include scheduling tasks for workers, tracking attendance, and monitoring the health of the crops. This requires a highly efficient management system to streamline operations, improve productivity, and ensure compliance with sustainability standards. Palm oil is crucial for Veronica Enterprise's operations, and like many other companies in the industry, it faces challenges related to productivity, resource management, and maintaining sustainability [3].

In addition to the economic benefits, the rapid expansion of oil palm plantations has raised concerns regarding environmental impact, including deforestation and loss of biodiversity [4]. These issues are particularly relevant to Veronica Enterprise, as it seeks to balance economic growth with environmentally responsible practices. Veronica Enterprise is committed to improving operational efficiency while adhering to sustainable practices in line with industry standards, such as those set by the Roundtable on Sustainable Palm Oil [5].

For Veronica Enterprise, a tailored plantation management system is crucial to effectively address operational challenges and implement sustainable practices. The Veronica Enterprise Oil Palm Plantation Worker Management System will play an important role in improving task scheduling, real-time monitoring, and attendance tracking, while also facilitating better resource allocation and environmental monitoring.

2.2 Web-Based Application

A web-based application is software that runs on web browsers and is accessed through the internet. Unlike traditional desktop applications, web-based applications do not require installation, as they operate entirely online. They are built using web technologies such as HTML, CSS, JavaScript, and backend frameworks like Node.js, Python Flask, or Java Spring Boot. The web-based application for Veronica Enterprise serves as the platform for managing plantation operations. Users, including the admin and workers, can log into the system from any device with internet access. This ensures flexibility and accessibility, enabling real-time updates and collaboration among users. Key functionalities include task scheduling, attendance tracking, financial management, and report generation. Web-based applications offer several advantages, such as easy updates, cross-platform compatibility, and centralized data storage. For this project, the web-based nature of the system ensures scalability and ease of maintenance, allowing Veronica Enterprise to manage multiple plantations seamlessly. Furthermore, its user-friendly interface enhances usability for all stakeholders involved, ensuring the system meets operational needs efficiently [6].

2.3 Quick Response (QR) Code Technology

Quick Response (QR) code technology is a widely adopted solution for quick and accurate data collection, particularly in attendance tracking systems. Each user is assigned a unique Quick Response (QR) code that can be scanned to log specific events, such as clocking in and out of work. Common features of this technology include real-time scanning, automated timestamps, and the ability to generate detailed attendance reports. These features ensure that data is captured accurately and efficiently, minimizing human error and enhancing accountability [7]; [8]. In this project, Quick Response (QR) code technology will be used in the attendance module to streamline the process of recording worker attendance. Workers will scan their unique Quick Response (QR) codes upon starting and ending their tasks, which will automatically log their attendance in real time. This approach eliminates the need for manual logbooks, reducing inaccuracies and making it easier to manage attendance records. The data collected through Quick Response (QR) scanning will also integrate with the system's reporting module, enabling admin to review attendance trends and worker punctuality effortlessly. The integration of Quick Response (QR) code technology not only improves the efficiency of attendance tracking but also enhances transparency and accountability among workers.

2.4 Comparison with the Existing System

To understand where the Veronica Enterprise Oil Palm Plantation Worker Management System stands, three well-known farm-management tools were examined: FarmERP, Farmbrite, and Conservis. Each offers a solid foundation for general agriculture, yet none is tailored to the unique day-to-day realities of an oil-palm plantation.

FarmERP is an enterprise-grade platform built for large, diversified farms. It combines task scheduling, workforce planning, inventory, and finance into a single dashboard. While its breadth is impressive—covering everything from crop planning to supply-chain tracking—the system can feel heavy for smaller teams because many modules are designed for multi-crop operations and detailed enterprise resource-planning (ERP) workflows.

Farmbrite focuses on ease of use for small to mid-size farms. Its strength lies in simple task boards, expense logging, and basic livestock or crop records. Farmers value its clean interface and straightforward reports. However, Farmbrite does not support Quick Response (QR)-based attendance or real-time activity feeds, making it less effective for plantations where dozens of field workers clock in and out at multiple sites each day.

Conservis is a cloud platform aimed at row-crop producers that need real-time field insights and financial visibility. It excels at live data capture from machinery and integrates smoothly with accounting packages. Still, its core modules revolve around field-level inputs (seed, chemical, yield) rather than worker oversight, so attendance and staff-centric metrics remain manual add-ons. Table 1 shows these systems comparison.

Table 1 *System's Comparison*

Features/Modules	FarmERP	Farmbrite	Conservis	Veronica Enterprise Oil Palm Management System
Login/SignUp Module	Yes	Yes	Yes	Yes
Profile Module	Yes	Yes	Yes	Yes
Task Scheduling Module	Yes	Yes	Yes	Yes
Notification Module	Yes	Yes	Yes	Yes
Attendance Module (via QR Code)	No	No	No	Yes
Activity Monitoring Module	Yes	Yes	Yes	Yes
Report Generation Module	Yes	Yes	Yes	Yes
Staff Management Module	Yes	Yes	Yes	Yes

All three benchmark systems cover the basics—logins, profiles, task boards, notifications, and high-level activity tracking. Where Veronica Enterprise's solution sets itself apart is the built-in Quick Response (QR) code

attendance module and live activity monitoring tuned for plantation workflows. By trimming the complex, multi-crop features of FarmERP, retaining Farmbrite's user-friendly layout, and borrowing Conservis's real-time mindset, the proposed system delivers exactly what an oil-palm estate needs: fast QR check-ins, instant task updates, and concise reports without the clutter of features a plantation manager would never use.

3. Methodology

The Prototyping Model is a software development methodology that focuses on creating a working model or prototype of the system early in the development process. This approach is especially suitable for projects where user requirements might change or evolve during the development process. By building prototypes, stakeholders can see how the system will function and provide feedback for refinements before the final version is developed [9]. The Prototyping Model was chosen for this project because it ensures the system aligns closely with the needs Mrs Veronica, and her workers. The model enables iterative improvements, reducing the risk of delivering a product that does not meet user expectations.

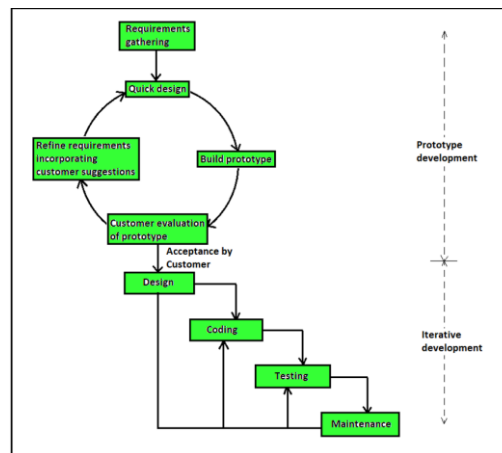


Fig. 1 *Prototype Model Process*

Figure 1 illustrates the phases of the Prototyping Model, a widely-used iterative software development approach. The process begins with the Requirement Analysis Phase, where user needs and system functionalities are identified. Based on these requirements, the initial Prototype Development Phase creates a basic model of the system for users to review and provide feedback. This is followed by the Prototype Refinement Phase, where changes are made based on user input, iterating until the prototype aligns with user expectations. Once finalized, the system undergoes the Testing and Evaluation Phase, ensuring that it meets all functional and non-functional requirements before deployment. Table 2 shows the software development activities in developing the Veronica Enterprise Oil Palm Plantation Worker Management System.

Table 2 *Software development activities in developing Veronica Enterprise Oil Palm Plantation Worker Management System*

Phase	Tasks	Outputs
Planning	- Proposed the project - Determine the project schedule, activities and output	- Project proposal - Gantt chart
Requirement Analysis	- Gather requirements from stakeholders - Conduct feasibility study - Model the system's processes and components	- Requirement document - Feasibility study report - System models
Design and Prototyping	- Create system wireframes and prototypes using Figma. - Create system architecture and high-level design	- Wireframes and system architecture. - UML diagrams - Design specification

Table 2 (cont.)

Phase	Tasks	Outputs
Refinement	Collect feedback, make enhancements, and retest prototypes	Updated prototypes
Testing	- Plan and design test cases - Set up test environment and execute test cases - Track and resolve defects - Update documentation as needed	- Test reports - Defect report - Verified software - Updated documentation

4. Analysis and Design

4.1 Requirement Analysis

System requirements analysis is a vital step in the development of any software system. This phase focuses on understanding and documenting the needs and constraints of the proposed system to ensure that the final solution addresses the specific goals and challenges effectively.

Functional requirements define the specific operations, features, and interactions that the system must provide to achieve its goals and satisfy user needs. These requirements outline the system's expected behavior, detailing the tasks it should perform, the processes it should support, and the interactions it should facilitate. Table 3 shows the functional requirements for the proposed application.

Table 3 Functional Requirements

Module	Function
Login/Signup	Enables users to register and log in, with secure authentication for both admins and workers.
Profile	Allows users to view and update their profiles, including details like name, contact information, and address.
Task Scheduling	Allows admins to assign tasks such as fertilizing, harvesting, and cleaning, with real-time updates for workers to view their schedules.
Attendance	Facilitates clocking in and out using QR code scanning, ensuring accurate attendance records.
Activity Status	Tracks the real-time status of tasks and allows workers to update completion progress.
Notifications	Sends reminders to workers for upcoming tasks and updates the admin on task completion status.
Generate Report	Generates statistical reports on worker attendance and task progress for analysis and decision-making.
Staff Management	Allows admins to view worker details and remove workers from the system if necessary.

On the other hand, non-functional requirements describe the quality attributes and constraints of the system, ensuring its performance, reliability, and user experience. Unlike functional requirements, which focus on what the system does, non-functional requirements address how the system operates under various conditions. Table 4 shows the non-functional requirements for the proposed application.

Table 4 *Non-Functional Requirements*

Module	Description
Performance	The system must handle different levels of usage effectively. It should respond quickly to user actions, process tasks efficiently, and support more users or data as needed without slowing down.
Reliability	The system must work consistently and dependably. It should be available most of the time, handle unexpected errors gracefully, and recover quickly from failures to minimize disruptions.
Security	The system must ensure the protection of sensitive data by using secure user authentication, role-based access control, and data encryption. This will help prevent unauthorized access and data breaches.
Usability	The system should have an easy-to-use interface that works smoothly on various devices, including desktops, tablets, and smartphones, making it accessible for all users.

4.2 UML Diagram

Unified Modeling Language (UML) is a standardized modeling language widely used in software engineering to visually represent a system's structure and behavior. UML provides a set of graphical notations and conventions that facilitate the design, analysis, and communication of object-oriented software systems. By using UML diagrams, developers and stakeholders can achieve a common understanding of a system's architecture and functionality [10].

Figure 2 presents the Use Case Diagram for the Veronica Enterprise Oil Palm Plantation Worker Management System, which is designed to streamline operations and improve efficiency. The diagram identifies the key interactions between the two main actors, workers and administrators, and the system's functionalities. These functionalities include login and signup, managing tasks, attendance tracking, monitoring activities, and generating reports. The system ensures seamless communication and operational management for all stakeholders.

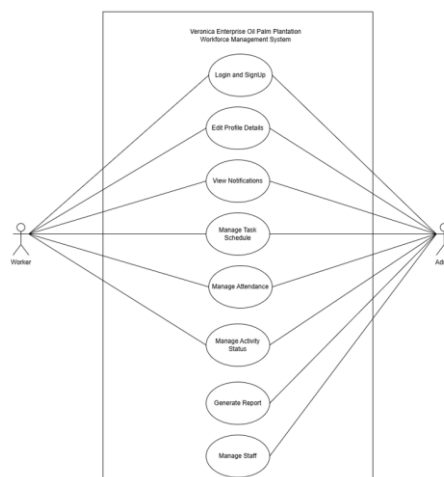


Fig. 2 *Use case Diagram*

5. Implementation and Testing

This chapter describes the system implementation process and the system testing for the Veronica Enterprise Oil Palm Plantation Management System.

5.1 System Implementation

System implementation covers all activities related to the development and coding of the system, including the creation of the source code, setup, and integration of features.

The Login and Signup interface is shown in Figure 4(a) and Figure 4(b). The Login and Signup Module of the Veronica Enterprise Oil Palm Plantation Management System allows users to access the system based on their roles, admin or worker. The login screen requires a username and password for authentication, ensuring secure access. The signup feature is intended for new users to register with relevant details, which are then stored in the database for future verification. These interfaces ensure that only authorized personnel can interact with the system.

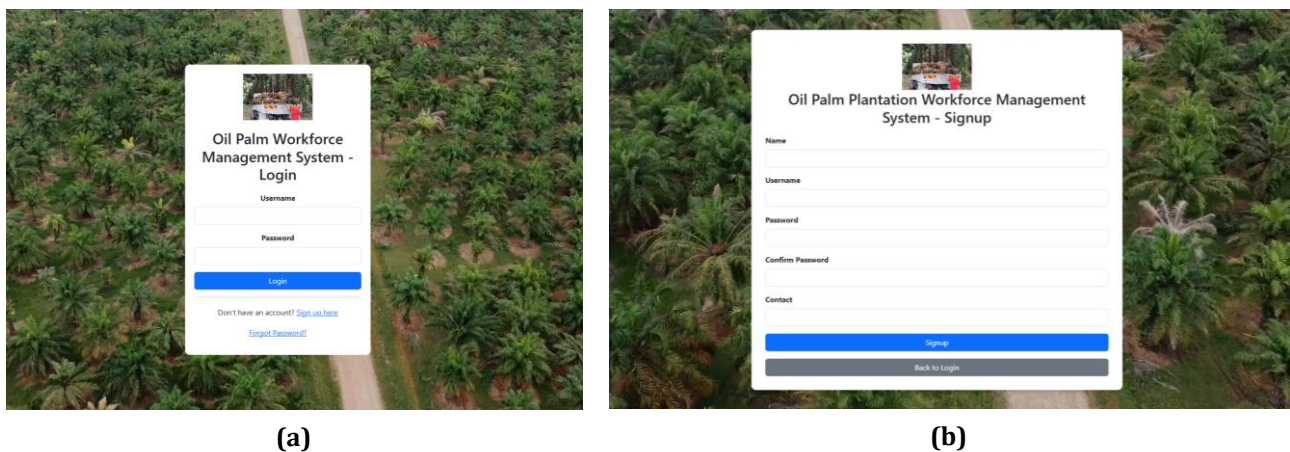


Fig. 4 Figure (a) Login Interface; (b) Signup Interface

The Profile interface is shown in Figure 5. The Profile Module allows both Admin and Workers to view and manage their personal information. It displays user details such as name, contact, and role. Users can update their profile information to keep their current data.

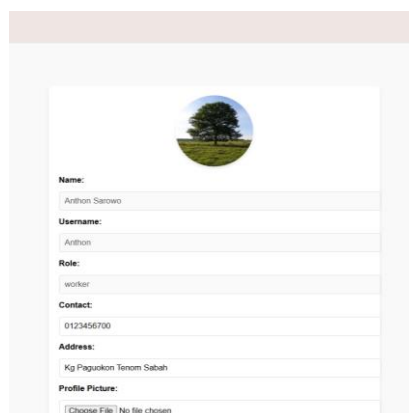
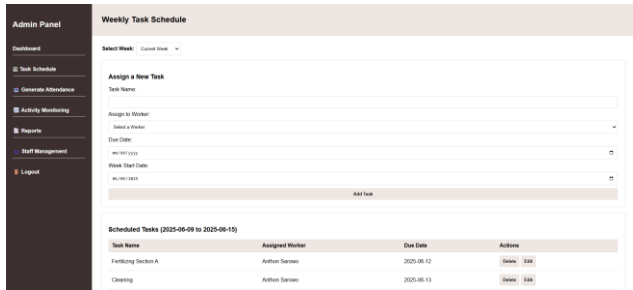


Fig. 5 Profile Interface

The Task Schedule interface is shown in Figure 6(c) and Figure 6(d). This module enables Admin to assign tasks to workers based on specific dates and sections of the plantation. It provides an organized interface for creating, viewing, and managing task assignments. Workers can also view their assigned tasks through their dashboard and task schedule page, which helps improve task clarity and workflow efficiency.



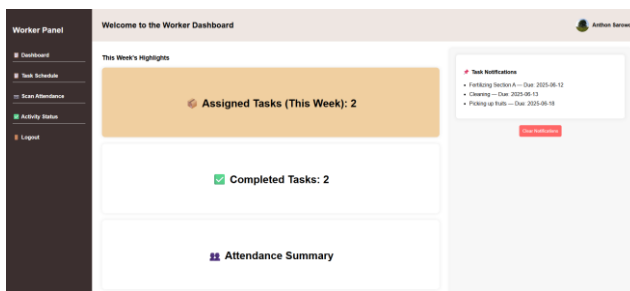
(c)



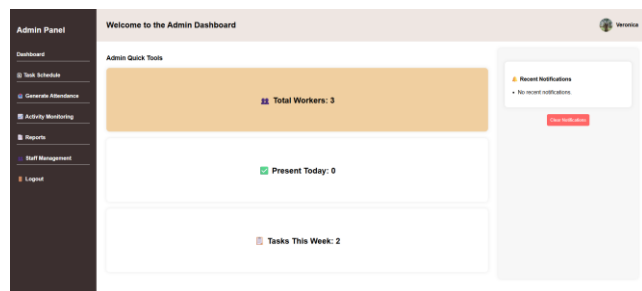
(d)

Fig. 6 Figure (c) Task Schedule Interface for Admin; (d) Task Schedule Interface Worker

The Notification interface is shown in Figure 7(e) and Figure 7(f). The Notification Module enhances communication between admin and workers. Admin can send real-time notifications regarding task assignments and updates. Workers receive these alerts through their dashboards. Both roles can view and clear notifications.



(e)



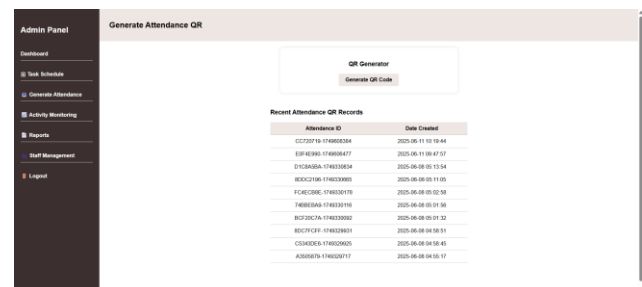
(f)

Fig. 7 Figure (e) Notification Interface Admin; (f) Notification Interface Worker

The Attendance interface is shown in Figure 8(g) and Figure 8(h). The Attendance Module uses QR code scanning to mark worker attendance. Admins generate a unique QR code, which workers scan using their dashboard interface. Each scan records the time and worker ID into the database, providing a real-time and efficient attendance tracking method.



(g)



(h)

Fig.8 Figure (g) Scan Attendance Interface; (h) Generate Attendance Interface

The Activity Status and Activity Monitoring interface is shown in Figure 9(i) and Figure 9(j). This module allows workers to update the status of their assigned tasks, such as marking them as "In Progress" or "Completed". This feature ensures transparency and allows Admin to monitor task progress remotely, contributing to better management of field activities.

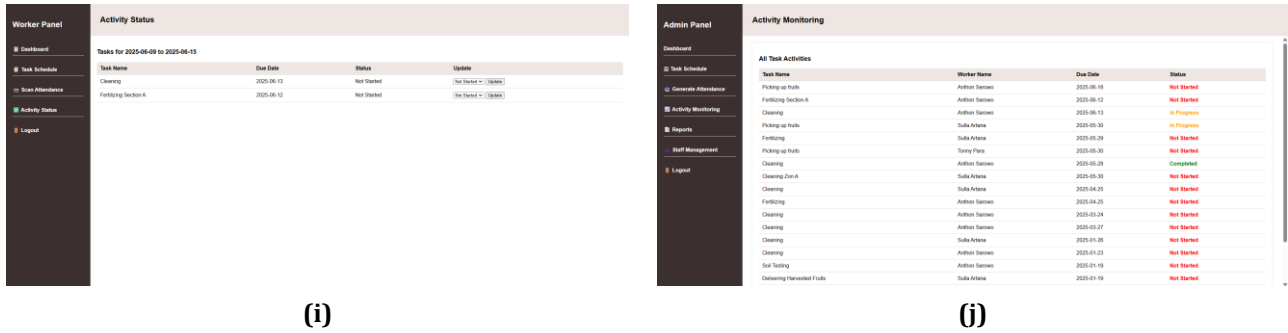


Fig. 9 Figure (i) Activity Status Interface; (j) Activity Monitoring Interface

The Generate Report interface is shown in Figure 10 (k). Admin can generate reports related to worker attendance and task completion using this module. These reports help in analyzing worker performance and managing plantation operations effectively. The system outputs data in a structured format for easy review. While, the Staff Management interface is shown in Figure 10 (l). It is exclusive to Admins and is used to manage worker data. Admins can view all registered staff, search for specific workers to see all their details, and remove users when needed. It ensures smooth user management and maintains up-to-date records.

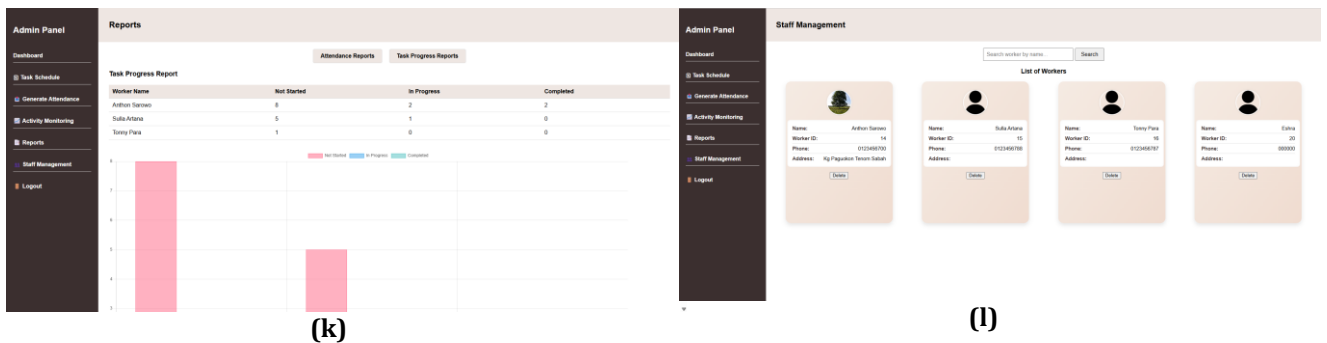


Fig. 10 Figure (k) Generate Report Interface; (l) Staff Management Interface

5.2 Testing

Before the system is released, it needs to be tested thoroughly to make sure it works correctly and meets all the required criteria. This testing helps ensure the system runs smoothly and performs as expected. The system is tested using a detailed test plan and User Acceptance Testing (UAT) to check its functionality and user satisfaction.

5.2.1 Functional Testing Plan

The functional testing plan explains how the system's features will be tested to make sure they work as they should. Table 5 to Table 12 shows the test plan for each module of the Veronica Enterprise Oil Palm Plantation Worker Management System, covering both the worker and admin interfaces.

Table 5 Test plan for Login and Signup module

Function	Roles	Test Case	Expected Result	Actual Result
User Registration (Signup)	Worker	Register with valid information (name, username, password, confirm password, and contact number)	Account is created, and the user is redirected to the login page.	Pass

User Registration (Invalid)	Worker	Register with missing or blank information (e.g., leaving name or password blank)	Error message displayed, such as "Please fill out this field."	Pass
Login with Valid Credentials	Worker	Login with a valid username and password	User is logged in successfully and redirected to the dashboard.	Pass
	Admin	Login with pre-set credentials	Admin is logged in successfully and redirected to the admin dashboard.	Pass
Login with Invalid Credentials	Worker/Admin	Login with incorrect username or password	Error message displayed: "Incorrect or No user found!"	Pass
Forgot Password	Worker	Click on "Forgot Password", enter a registered contact if valid then allow user reset password	A password can be reset if contact match.	Pass
User Logout	Worker/Admin	Click "Logout" button	User is logged out and redirected to the login page	Pass

Table 6 Test plan for Profile module

Function	Roles	Test Case	Expected Result	Actual Result
View Profile	Worker	View personal profile details (name, contact, etc.)	Profile details are displayed correctly for the logged-in worker	Pass
Edit Profile	Worker	Update personal information (name, contact details)	The updated information is saved and displayed correctly	Pass
View Admin Profile	Admin	View admin profile details	Admin profile details are displayed correctly	Pass
Edit Admin Profile	Admin	Update admin information (e.g., contact details)	The admin's updated information is saved and displayed correctly	Pass
Profile Picture Update	Worker/Admin	Upload and update profile picture	The new profile picture is uploaded and displayed correctly	Pass
Password Change	Worker/Admin	Change password and confirm the changes	The password is updated successfully and the user can log in with the new password	Pass

Table 7 Test plan for Task Schedule module

Function	Roles	Test Case	Expected Result	Actual Result
Assign Task	Admin	Assign a task to a worker	The task is assigned to the worker and displayed in the worker's task list	Pass

Update Task Status	Worker	Update task status (e.g., in progress, completed)	The status of the task is updated successfully and reflected in the system	Pass
View Task Assignments	Admin	View task assignments for all workers	The admin sees the list of all tasks assigned with corresponding statuses	Pass
Update Task Details	Admin	Modify task details such as deadlines or worker assignment	The task details are updated successfully in the system	Pass
Delete Task	Admin	Delete a task	The selected task is removed from the task list and the database	Pass

Table 8 *Test plan for Notification module*

Function	Roles	Test Case	Expected Result	Actual Result
Send Task Assignment Notification	Admin	Assign a task to a worker and send a notification about the new task	The worker receives a notification about the new task assignment	Pass
Send Task Completion Notification	Worker	Mark a task as completed and send a notification to the Admin	The Admin receives a notification about the task completion	Pass
Receive Task Update Notification	Worker	Admin update the assigned task	The worker receives a notification about the updated task status	Pass
Notification View	Worker/Admin	View a list of received notifications	The list of notifications is displayed correctly, showing all received notifications	Pass
Notification Clear	Worker/Admin	Clear all notifications	All notifications are removed from the list after the "clear" action	Pass

Table 9 *Test plan for Attendance module*

Function	Roles	Test Case	Expected Result	Actual Result
Generate QR Code	Admin	Generate a QR code for worker attendance	A unique QR code is generated and displayed for the worker to scan	Pass
Scan QR Code	Worker	Scan the generated QR code to mark attendance	The worker's attendance is recorded successfully in the system	Pass
View Attendance History	Admin	View the list of attendance records for all workers	The admin can see all attendance logs, including worker ID, date, time, and status	Pass
View Worker Attendance	Admin	View attendance history for a specific worker	The admin can view detailed attendance records for a specific worker	Pass

Update Attendance Record	Admin	Update an attendance record (e.g., change the status to "Absent" or "Leave")	The attendance record is updated correctly in the system	Pass
Delete Attendance Record	Admin	Delete a specific attendance record	The selected attendance record is removed from the system	Pass
Scan Invalid QR Code	Worker	Scan an invalid or expired QR code	An error message is displayed: "Invalid QR Code. Please try again."	Pass

Table 10 *Test plan for Activity Status module*

Function	Roles	Test Case	Expected Result	Actual Result
View Current Activity Status	Worker/Admin	View the current status of an ongoing task (e.g., in progress, completed)	The current status of the task is displayed correctly (e.g., in progress, completed)	Pass
Update Task Status	Worker	Update the status of a task (e.g., mark as in progress, completed)	The task status is updated successfully and reflected in the system	Pass
View Completed Tasks	Admin	View completed tasks and their status	Completed tasks are listed with their completion status and details	Pass
Activity Status Notification	Admin	Receive a notification when a task status is updated (e.g., completed)	A notification is sent to the admin when the task status changes	Pass
Real-Time Task Progress Update	Worker	Real-time update of task status (e.g., marking a task as completed or in progress)	The task status is updated in real-time without delay	Pass
Update Incomplete Task Status	Worker	Update a task marked as "Not Started" to "in progress" or "completed"	The task is updated successfully and the status is reflected in the system	Pass
View Task Status History	Admin	View the history of a worker's task statuses (e.g., completed, in progress, etc.)	The system displays a complete history of task statuses for each worker	Pass

Table 11: *Test plan for Generate Report module***Attendance Report Generation**

Function	Roles	Test Case	Expected Result	Actual Result
----------	-------	-----------	-----------------	---------------

Generate Attendance Report	Admin	Generate an attendance report for all workers based on a specific date or range	The report is generated and shows the attendance status for each worker on the selected date range	Pass
Filter Attendance by Worker	Admin	Generate a report for a specific worker's attendance	The report shows the attendance history for the selected worker only	Pass
Filter Attendance by Date	Admin	Generate a report for attendance by date range (e.g., weekly, monthly)	The report shows attendance for the specified date range	Pass
Download Attendance Report	Admin	Download the generated attendance report as PDF	The report is downloaded successfully in pdf format	Pass
Attendance Report Error Handling	Admin	Attempt to generate an attendance report with incorrect date range or worker ID	Error message is displayed: "Invalid date range or worker ID"	Pass
View Generated Attendance Report	Admin	View previously generated attendance reports	The list of previous reports is displayed with view options	Pass
Task Progress Report Generation				
Generate Task Progress Report	Admin	Generate a task progress report for all workers	The report is generated, showing each worker's assigned tasks, progress, and completion status	Pass
Filter Task Progress by Worker	Admin	Generate a task progress report for a specific worker	The report shows task progress for the selected worker only	Pass
Filter Task Progress by Date	Admin	Generate a report for task progress by date range (e.g., weekly, monthly)	The report shows task progress for the selected date range	Pass
Download Task Progress Report	Admin	Download the generated task progress report as PDF or CSV	The report is downloaded successfully in the selected format	Pass
Task Progress Report Error Handling	Admin	Attempt to generate a task progress report with incorrect date range or worker ID	Error message is displayed: "Invalid date range or worker ID"	Pass
View Generated Task Progress Report	Admin	View previously generated task progress reports	The list of previous reports is displayed with download/view options	Pass

Table 12 Test plan for Staff Management module

Function	Roles	Test Case	Expected Result	Actual Result
----------	-------	-----------	-----------------	---------------

View Staff List	Admin	View the list of all staff members	The list of all staff members is displayed with their details	Pass
Delete Staff	Admin	Delete a staff member from the system	The staff member is removed from the system, and the list is updated accordingly	Pass
Search Staff	Admin	Search for a specific staff member by name or role	The correct staff member is displayed based on the search criteria	Pass

5.2.2 User Acceptance Testing (UAT)

User Acceptance Testing (UAT) is an important step in the software development process. It ensures that the system works as expected and is ready to be used in real world situations. This phase checks the system's functionality, reliability, and how easy it is for users to interact with. The UAT was conducted using Google Forms and involved 30 randomly selected respondents. The results of UAT are shown in Table 13 and Table 14. Users rate their satisfaction on a scale from 1 (Very Dissatisfied) to 5 (Very Satisfied).

Table 13 Result of User Acceptance Testing

No.	Acceptance Requirement	Ranking				
		1	2	3	4	5
User Interface & User Experience						
I.	Clear, user-friendly navigation				√	
II.	Visually appealing design			√		
III.	Quick loading times				√	
IV.	Immediate feedback for actions				√	
V.	An enjoyable and satisfying experience				√	
System Functionality						
VI.	Attendance module functionality (QR code generation and scanning)					√
VII.	Task management functionality				√	
VIII.	Worker profile management functionality				√	
IX.	Notifications and reminders				√	
X.	Worker progress tracking and reporting					√

Table 14 Summary of User Acceptance Testing

No.	Acceptance Requirement	Ranking					Total
		1	2	3	4	5	
User Interface & User Experience							
I.	Clear, user-friendly navigation	-	-	9	11	10	30
II.	Visually appealing design	-	-	13	11	6	30
III.	Quick loading times	-	-	8	12	10	30
IV.	Immediate feedback for actions	-	-	9	13	8	30
V.	An enjoyable and satisfying experience	-	-	10	14	6	30
System Functionality							
VI.	Attendance module functionality (QR code generation and scanning)	-	-	7	11	12	30
VII.	Task management functionality	-	-	5	14	11	30
VIII.	Worker profile management functionality	-	-	6	14	10	30
IX.	Notifications and reminders	-	-	6	15	9	30
X.	Worker progress tracking	-	-	3	13	14	30

6. Conclusion

In conclusion, the Veronica Enterprise Oil Palm Plantation Worker Management System offers a modern and practical solution for managing plantation operations more efficiently. By integrating features such as Quick Response (QR) code-based attendance tracking, task scheduling, real-time activity monitoring, and automated report generation, the system addresses many of the issues found in traditional manual processes. It reduces human error, improves task coordination, and enables better decision-making through centralized and up-to-date information. The development of this system demonstrates how digital tools can improve routine operations, allowing plantation managers to manage their workforce more effectively.

One of the key advantages of the system is its ability to automate attendance through QR scanning, significantly reducing manual record-keeping. Administrators can manage and monitor tasks from a centralized dashboard, while workers receive real-time updates and can view their schedules through an intuitive interface. Reports for tasks and attendance are generated automatically, supporting documentation and planning. Additionally, the system's simple and user-friendly design ensures that both administrators and workers can adopt it quickly. However, the system also has some limitations. It depends on a stable internet connection, which may not always be available in remote plantation areas. Currently, it is designed for single-site operations and does not yet support multiple plantations. The absence of a mobile app also limits accessibility for field workers, and the reporting module could be improved with more detailed filtering options.

To enhance the system and ensure it meets future operational needs, several improvements are recommended. First, developing a mobile app version would make the system more accessible for workers in the field. Adding more advanced filters for reports—such as by date, task type, or plantation section—would allow administrators to analyze data more effectively. Expanding the system to support multiple plantation sites within a single admin panel would greatly improve scalability. Additionally, improving the interface to better adapt to different screen sizes and devices would enhance the user experience. Features like export-to-PDF or Excel for attendance and task reports, as well as the addition of a performance tracking dashboard, would provide deeper insights and support long-term workforce management. Hence, these improvements would help the system grow in value and usability over time, ensuring it continues to support Veronica Enterprise's evolving needs.

Acknowledgement

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

Conflict of Interest

The 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:** Eshra Novera Binti Sandrah; **data collection:** Eshra Novera Binti Sandrah; **analysis and interpretation of results:** Eshra Novera Binti Sandrah, Nurezayana Binti Zainal; **draft manuscript preparation:** Eshra Novera Binti Sandrah, Nurezayana Binti Zainal. All authors reviewed the results and approved the final version of the manuscript.

References

- [1] Food and Agriculture Organization, "Oil palm: A key sector for sustainable development," FAO, 2021. [Online]. Available: <https://www.fao.org/palm-oil>
- [2] L. P. Koh and D. S. Wilcove, "Is oil palm development really destroying tropical biodiversity?" *Conservation Letters*, vol. 1, no. 2, pp. 60–64, 2008.
- [3] D. S. Wilcove and L. P. Koh, "Addressing the threats to biodiversity from oil-palm agriculture," *Biological Conservation*, vol. 143, no. 6, pp. 2273–2281, 2010.

- [4] Roundtable on Sustainable Palm Oil, "RSPO principles and criteria for sustainable palm oil production," RSPO, 2020. [Online]. Available: <https://rspo.org>
- [5] Techopedia, "Web-based application." [Online]. Available: <https://www.techopedia.com/definition/26002/web-based-application>
- [6] M. Ahmed, S. Khan, and R. Saeed, "The impact of QR code technology on real-time data collection," *International Journal of Computer Applications*, vol. 182, no. 1, pp. 25–30, 2020.
- [7] X. Zhou, Y. Wang, and H. Li, "QR code applications in workplace management," *Journal of Digital Innovation*, vol. 45, no. 3, pp. 215–230, 2018.
- [8] R. S. Pressman, *Software Engineering: A Practitioner's Approach*, 8th ed. McGraw-Hill, 2014.
- [9] J. Rumbaugh, I. Jacobson, and G. Booch, *The Unified Modeling Language Reference Manual*, 2nd ed. Addison-Wesley, 2010.
- [10] A. Dennis, B. H. Wixom, and R. M. Roth, *System Analysis and Design: An Object-Oriented Approach with UML*, 4th ed. Wiley, 2008.
- [11] M. Rana, "The prototyping model in software development," *International Journal of Software Engineering*, vol. 22, no. 3, pp. 140-150, 2021

Appendix A: Gantt Chart

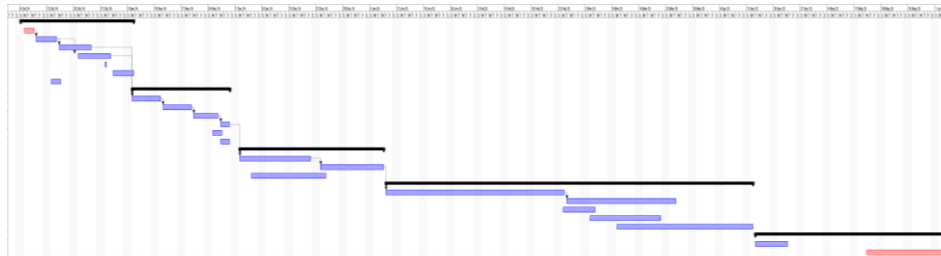


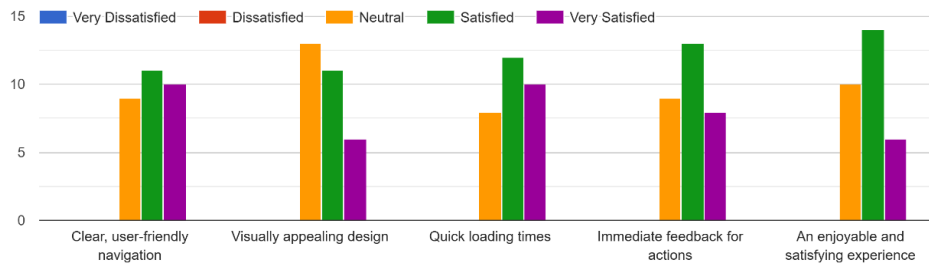
Fig. A.1 Project Gantt Chart

Name	Duration	Start	Finish
Planning Phase	21 days	10/6/24, 8:00AM	11/4/24, 5:00PM
Define project scope and objectives	3 days	10/6/24, 8:00AM	10/9/24, 5:00PM
Conduct stakeholder meetings	4 days	10/10/24, 8:00AM	10/15/24, 5:00PM
Conduct studies	7 days	10/16/24, 8:00AM	10/24/24, 5:00PM
Develop project proposal	7 days	10/20/24, 8:00AM	10/29/24, 5:00PM
Title defense	1 day	10/27/24, 8:00AM	10/28/24, 5:00PM
Evaluate feasibility	4 days	10/30/24, 8:00AM	11/4/24, 5:00PM
Create Gantt chart and project timeline	3 days	10/13/24, 8:00AM	10/16/24, 5:00PM
Analysis Phase	20 days	11/4/24, 8:00AM	11/29/24, 5:00PM
Requirements validation	6 days	11/4/24, 8:00AM	11/11/24, 5:00PM
Use case development	6 days	11/12/24, 8:00AM	11/19/24, 5:00PM
Develop test and evaluation requirements	5 days	11/20/24, 8:00AM	11/26/24, 5:00PM
Data mocking	3 days	11/27/24, 8:00AM	11/29/24, 5:00PM
Requirements feasibility	3 days	11/24/24, 8:00AM	11/27/24, 5:00PM
SRS preparation	3 days	11/27/24, 8:00AM	11/29/24, 5:00PM
Design Phase	28 days	12/2/24, 8:00AM	1/8/25, 5:00PM
Architectural design	15 days	12/2/24, 8:00AM	12/20/24, 5:00PM
Database design	13 days	12/23/24, 8:00AM	1/8/25, 5:00PM
User interface design	14 days	12/5/24, 8:00AM	12/24/24, 5:00PM
Implementation Phase	51 days	1/9/25, 8:00AM	3/20/25, 5:00PM
Implementation of System Prototype	30 days	1/9/25, 8:00AM	2/19/25, 5:00PM
Prototype Refinement Phase	21 days	2/20/25, 8:00AM	3/20/25, 5:00PM
Testing Phase	23 days	5/5/25, 8:00AM	6/4/25, 5:00PM
Functional Testing	7 days	5/5/25, 8:00AM	5/13/25, 5:00PM
User Acceptance Testing (UAT)	16 days	5/14/25, 8:00AM	6/4/25, 5:00PM

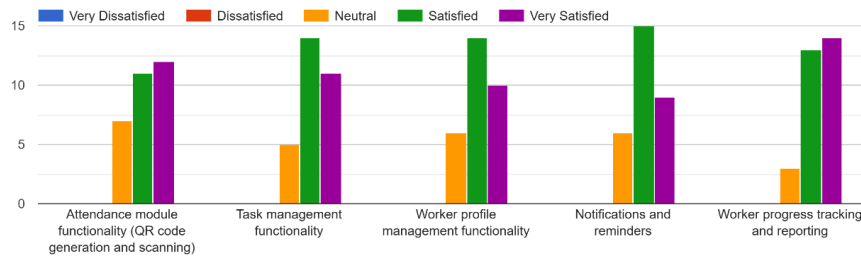
Fig. A.2 Project schedule and planning

Appendix C: User Acceptance Testing

1. User Interface & User Experience



2. System Functionality



Appendix D: Design Interface

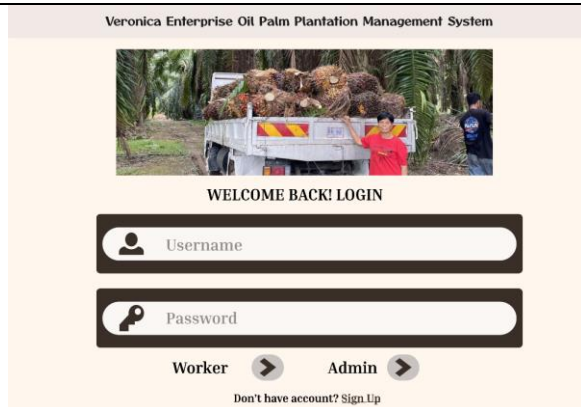


Fig D.1 Login interface



Fig D.2 Signup interface

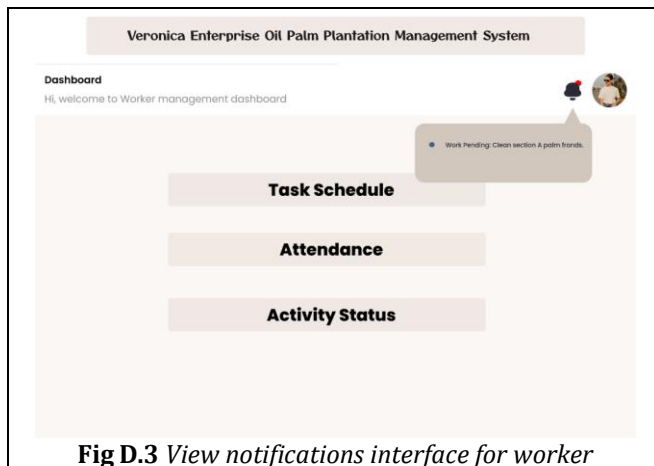


Fig D.3 View notifications interface for worker

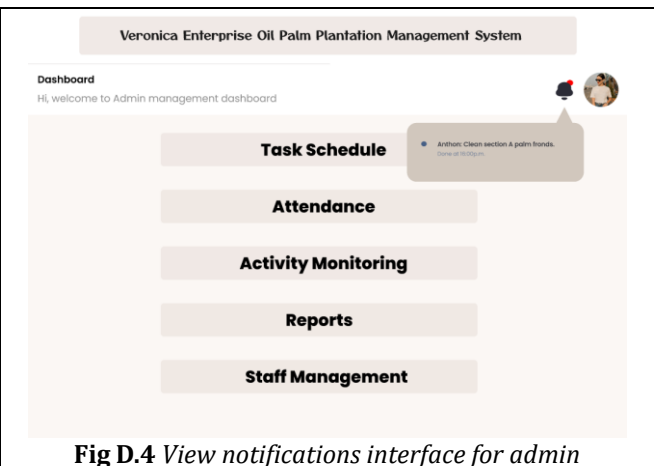


Fig D.4 View notifications interface for admin

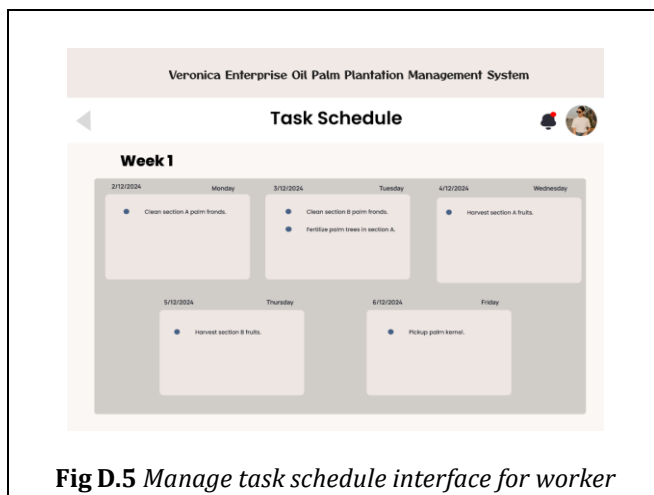


Fig D.5 Manage task schedule interface for worker

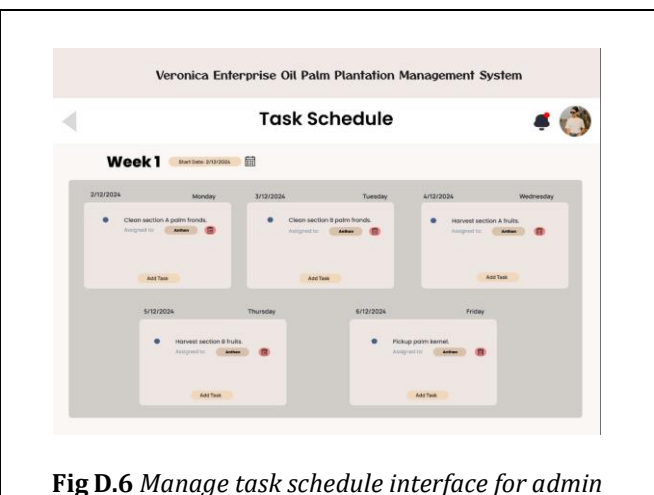


Fig D.6 Manage task schedule interface for admin

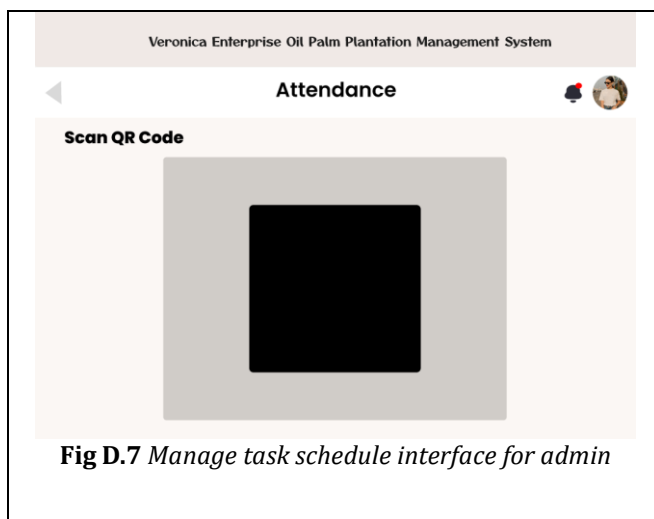


Fig D.7 Manage task schedule interface for admin

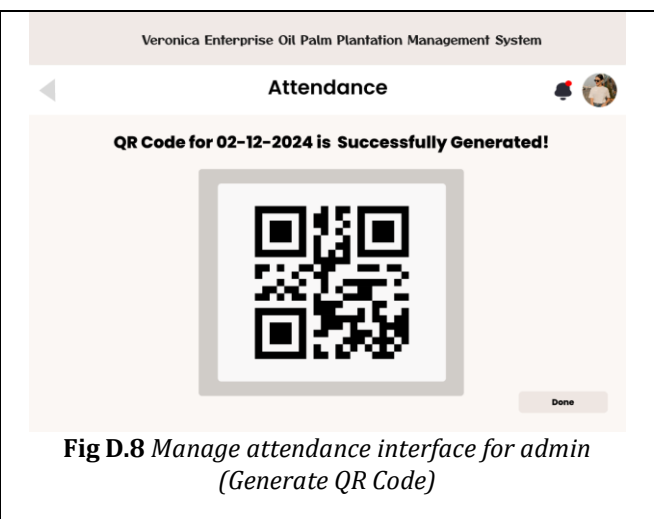


Fig D.8 Manage attendance interface for admin (Generate QR Code)



Fig D.9 Manage attendance interface for admin
(Generate QR Code)

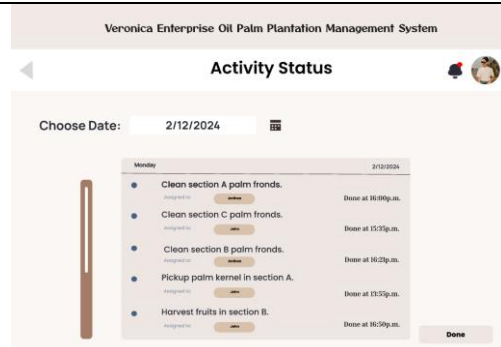


Fig D.10 Manage activity interface for admin



Fig D.11 Edit profile interface



Fig D.12 Generate reports interface



Fig D.13 Attendance Reports interface if admin wants a report on worker's attendance

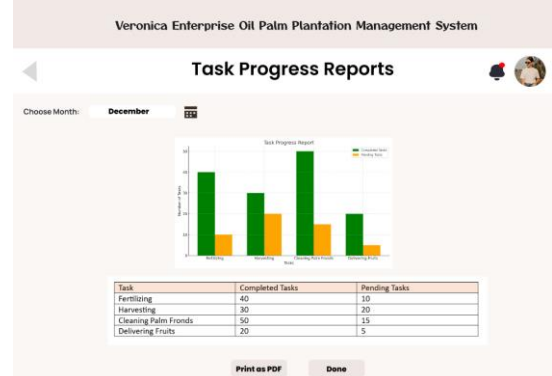


Fig D.14 Task Progress Reports interface if admin wants a report on worker's task progress