

## Strawberry Stock and Order Management System

Nur Afrina Fasihah Ahmad Fadzil<sup>1</sup>, Hannani Aman<sup>1\*</sup>

<sup>1</sup> *Fakulti Sains Komputer dan Teknologi Maklumat,*

*Universiti Tun Hussein Onn Malaysia, Parit Raja, Batu Pahat, 86400, MALAYSIA*

\*Corresponding Author: [hanani@uthm.edu.my](mailto:hanani@uthm.edu.my)

DOI: <https://doi.org/10.30880/aitcs.2025.06.02.057>

### Article Info

Received: 15 June 2025

Accepted: 20 November 2025

Available online: 30 November 2025

### Keywords

Inventory Management, Barcode System, Agriculture Technology, Flask Framework, Cross-docking priority.

### Abstract

Agricultural and retail industries depend on efficient stock and order management to maintain efficient operations, reduce waste, and satisfy customer requests. The Strawberry Stock and Order Management System was created to reduce the drawbacks of manual inventory tracking and order processing, such as ineffective stock prioritization that causes spoiling and waste and order fulfillment delays, to improve operational efficiency at Mr. Khaw's strawberry farm in Cameron Highlands. To accelerate job execution, the system organizes its components as interactive objects by utilizing Object-Oriented Analysis and System Design (OOASD). Barcode-based inventory management, cross-docking order priority, real-time data processing with FIFO inventory updates, and comprehensive reports are some of the main features. This integration provides useful, data-based information to enhance well-informed decision-making, increases production capacity, and lowers human error. In the end, the system serves as an example of how contemporary technology may revolutionize agricultural operations by enhancing precision, effectiveness, and general output. To further increase the system's intelligence and accessibility, future works might concentrate on automated restock alerts, multilingual support, advanced analytics, mobile optimization, and external platform integration.

## 1. Introduction

Mr. Khaw Cheng Cheng is a seasoned strawberry farmer managing a 2.32-acre farm in Cameron Highlands that produces between 2,500 to 3,000 kilograms of strawberries per harvest. The farm currently relies on manual processes for managing inventory and processing customer orders. Stock quantities are tracked using handwritten notes, and order details are recorded manually, which increases the risk of human error and delays in operation.

The problem statement addressed in this project involves key challenges arising from the current manual system. These include inaccurate inventory tracking due to human error, inefficient stock prioritization that leads to spoilage and wastage, and delays in order processing that affect customer satisfaction. These issues highlight the need for a more efficient, automated solution tailored to the farm's operational environment.

In response, the objectives of this project are to design an automated stock tracking and order management system using an object-oriented approach and to develop the system using Python. Next, to test its functionality through black box testing. The goal is to reduce human error, improve inventory accuracy, prioritize stock using the FIFO method, and enhance productivity through real-time data processing and reporting. The remainder of

this paper is organized into sections discussing related works, methodology, system design, implementation and testing results, and final conclusions.

## 1.1 Scope

The goal of this project is to automate the management of orders and inventories at Mr. Khaw Cheng Cheng's strawberry farm. The administrator and the employees who work on regular basis are the primary users. Direct shipment of high-priority orders is prioritized by the system using the Cross-Docking approach and the First-In, First-Out (FIFO) method to minimize spoiling. Table 1 shows how its primary functions are arranged into five major modules.

**Table 1** System Module and Targeted User

System Module	Description	Targeted User
<b>Authentication and User Management module</b>	Allow users to log in and out of the system: <ol style="list-style-type: none"> <li>i. Allow users to log in using their user ID and password.</li> <li>ii. The administrator creates new users and provides them with credentials.</li> <li>iii. Administrator activates and disables employee accounts based on employment status.</li> </ol>	Administrator and employees
<b>Stock Management Module</b>	Allow users to monitor and track the stock flow: <ol style="list-style-type: none"> <li>i. Add new stock during stock check-in.</li> <li>ii. Update current stock during stock check-out.</li> <li>iii. Review the process history for stock checks in and out.</li> <li>iv. During stock checking, identify strawberries using specified attributes like barcodes and suppliers.</li> <li>v. Keep track of remaining stock using first-in, first-out (FIFO) technique.</li> </ol>	Administrator and employees
<b>Order Management Module</b>	Allow users to manage customer orders: <ol style="list-style-type: none"> <li>i. Create, view, and track orders.</li> <li>ii. Fulfilled orders by packing and scanning items for shipment.</li> </ol>	Administrator and employees
<b>Cross-docking Management Module</b>	Facilitate real-time stock movement directly from receiving to shipping without long-term storage: <ol style="list-style-type: none"> <li>i. Allow cross-docking for high-priority orders.</li> <li>ii. Track batch movement for improved storage and spoiling time using first-in, first-out (FIFO) technique.</li> <li>iii. Assign barcodes to crates maintaining sorted and weighed strawberries. Barcodes indicate the harvest date and grade for accurate tracking.</li> </ol>	Administrator and employees
<b>Generate Report Module</b>	Allow users to generate: <ol style="list-style-type: none"> <li>i. Inventory reports covering real-time stock levels and restocking needs</li> <li>ii. Order reports such as order fulfilment statuses and high-priority order tracking</li> <li>iii. Financial data report</li> </ol>	Administrator

## 2. Related Work

Barcode technology, combined with Cross-Docking and the FIFO (First-In, First-Out) inventory management technique, plays a crucial role in streamlining inventory control and order fulfillment processes. 3 system has been compared to address the advantages and disadvantages of a standard system [2][3][4]. This section reviews these systems to identify gaps that the proposed system aims to bridge.

### 2.1 Technology and Method support used in Inventory System

Modern inventory management systems, especially in agriculture, increasingly rely on barcode technology to improve the accuracy and efficiency of stock handling by automating product identification and tracking [5]. This approach enables real-time inventory updates across the supply chain, minimizing the risks of mismanagement, spoilage, and delays, while ensuring the quality and timely delivery of perishable products like strawberries [6].

Through the utilization of barcode scanning, the Cross-Docking method for direct stock transfers, and the First-In First-Out (FIFO) approach to guarantee that older stock is prioritized to reduce waste and improve order fulfillment efficiency, the suggested approach improves these capabilities. The Python Flask framework [7], which offers a real-time, lightweight, and modular web-based interface, is used in its development. Additionally, the system uses MySQL for data management, and Python's large library ecosystem facilitates scalability and maintainability. These technologies work together to provide Mr. Khaw's strawberry farm with a dynamic and dependable system.

### 2.2 Inventory Management Existing Systems Comparison

Various existing inventory systems featuring Enterprise Resource Planning (ERP) were reviewed [1], including AgriERP[2], MPRISE Agriware[3], and GoPalms[4].

**Table 2** System comparison

Feature	Inventory system			
	MPRISE Agriware Inventory Management for Plant Growers	Inventory System for Agricultural Businesses with AgriERP	Inventory Management for Plantation Businesses at GoPalms	Strawberry Stock and Order Management System
Platform	Web-based	Web-based	Web-based	Web-based
Login	Yes	Yes	Yes	Yes
Manage Staff Information	No	No	No	Yes
Add Inventory	Yes (scan barcode)	Yes	Yes	Yes (scan barcode)
Update Inventory	Yes (scan barcode)	Yes	Yes	Yes (scan barcode)
Display Inventory	Yes	Yes	Yes	Yes
Track Inventory Flow	Yes	Yes	Yes	Yes (FIFO)
Request Stock Out	Yes	No	No	Yes
Generate Report	Yes	Yes	Yes	Yes
Generate barcode	Yes	No	No	Yes

Although current inventory systems have a strong emphasis on effective resource management and real-time tracking, they sometimes lack capabilities designed especially for perishable items like strawberries. To close this gap and preserve product freshness while reducing waste, the proposed approach combines barcode scanning with FIFO-based priority. Despite providing useful base for managing agricultural resources, inventory solutions such as MPRISE Agriware[2], AgriERP[3], and GoPalms[4] fall behind in addressing the difficulties related to perishable inventory. The proposed system's special features are designed to maximize the handling and fast distribution of perishable goods, guaranteeing increased precision and operational effectiveness.

### 3. Prototyping Model

Prototyping Model [8][9] was adopted to ensure iterative development and active feedback from stakeholders.

#### 3.1 System Development Workflow

Management System is systematically organized into distinct phases, ensuring that each step contributes to a well-structured and functional system. Each phase involves specific tasks and delivers clear outputs to maintain progress and alignment with the project objectives. Table 3 outlines the system development workflow, detailing the phases, tasks, and corresponding outputs.

**Table 3** *System development workflow*

Phase	Task	Output
<b>Planning</b>	i. Propose the project.	i. Project proposal.
	ii. Identify problem statement and objectives.	ii. Create Gantt chart.
	iii. Create a work plan.	iii. Write literature review.
	iv. Review relevant sources and online articles about the topic.	iv. Compare the developed system with existing ones.
	v. Analyse existing applications' features and functions.	
<b>Analysis</b>	i. Collect user requirements.	i. User requirements.
	ii. Analyze hardware and software needs.	ii. Determine hardware and software needs.
	iii. Identify relationships between processes, data, and entities.	iii. Identify functional and non-functional requirements.
		iv. Examples include context diagrams and entity relationship diagrams.
<b>Design</b>	i. Develop wireframes.	i. Interactive mock-ups.
	ii. Design the database schema.	ii. Database requirements.
<b>Implementation</b>	i. Create system modules.	i. Prototype.
	ii. Integrate the system.	ii. Identified and fixed bugs.
	iii. Connect with the database.	
	iv. Generate a simple system prototype for testing.	
<b>Testing: First Cycle (Functionality)</b>	i. Manage system testing.	i. Improving and correcting problems.
<b>Testing: Second Cycle (UAT)</b>	i. Identify opportunities for improvement.	ii. Reviewed the system.
		iii. Getting the system ready for use by users.

### 4. Results and Discussion

The results demonstrate the system's capability to streamline inventory tracking and order processing through real-time updates and barcode integration. This section discusses the system's performance, testing outcomes, and potential improvements

## 4.1 System Analysis and Design

Functional requirements are the system's visible behaviours under specific situations, as well as the actions that users are permitted to conduct when using the system. Table 4.1 displays the functional requirements for the proposed system.

**Table 4.1** *System functional requirements*

Module	Description
<b>Authentication and User Management module</b>	<ul style="list-style-type: none"> <li>• Users can log in and out with their registered username and password.</li> <li>• The system validates user credentials and returns an error message if they are invalid.</li> <li>• The administrator will be able to establish new user accounts and assign credentials.</li> <li>• The administrator will activate or disable user accounts based on their employment status.</li> </ul>
<b>Stock Management Module</b>	<ul style="list-style-type: none"> <li>• Users can add new stock during check-in and update current stock at checkout.</li> <li>• The system keeps track of all stock transactions, including stock-in and stock-out, as part of the transaction history.</li> <li>• Users can look for stock information using attribute such barcode.</li> <li>• The system generates barcodes for items that do not currently have one.</li> <li>• The system can show the remaining stock utilizing the FIFO (First-In, First-Out) method.</li> </ul>
<b>Order Management Module</b>	<ul style="list-style-type: none"> <li>• Users can generate orders.</li> <li>• Users can monitor orders.</li> <li>• Users can track orders.</li> <li>• Users can get precise order management information, such as shipment evaluations and packing specifications.</li> </ul>
<b>Cross-docking Management Module</b>	<ul style="list-style-type: none"> <li>• The system should enable for real-time stock movement from receiving to shipping places, eliminating the need for long-term storage.</li> <li>• The system uses the FIFO (First-In-First-Out) technique to decide which orders are high priority.</li> <li>• Users can sort stock using FIFO (First-In-First-Out) techniques.</li> <li>• Users can allocate stock using FIFO (First-In-First-Out) techniques.</li> <li>• The system tracks items for cross-docking based on attributes including barcodes, grades, and harvest dates.</li> </ul>
<b>Generate Report Module</b>	<ul style="list-style-type: none"> <li>• System generate detailed information on inventory stock levels and restocking needs.</li> <li>• The system creates order fulfilment reports depending on order status and priority.</li> <li>• The system generates financial reports based on transaction history, income, and expenses.</li> </ul>
<b>Authentication and User Management module</b>	<ul style="list-style-type: none"> <li>• Users can log in and out with their registered username and password.</li> <li>• The system validates user credentials and returns an error message if they are invalid.</li> <li>• The administrator will be able to establish new user accounts and assign credentials.</li> <li>• The administrator will activate or disable user accounts based on their employment status.</li> </ul>

Non-functional requirements are those features of a system that affect its efficiency and effectiveness for end users rather than its business functionality. Table 4.2 displays the developed system's non-functional needs.

**Table 4.2** *System non-functional requirements*

Requirements	Description
<b>Authentication</b>	Ensures that only authorized users can access the system via secure login.
<b>Role based control</b>	Assigns specific functionalities to users based on their roles
<b>Security</b>	Protects sensitive data using encryption and secure database connections.

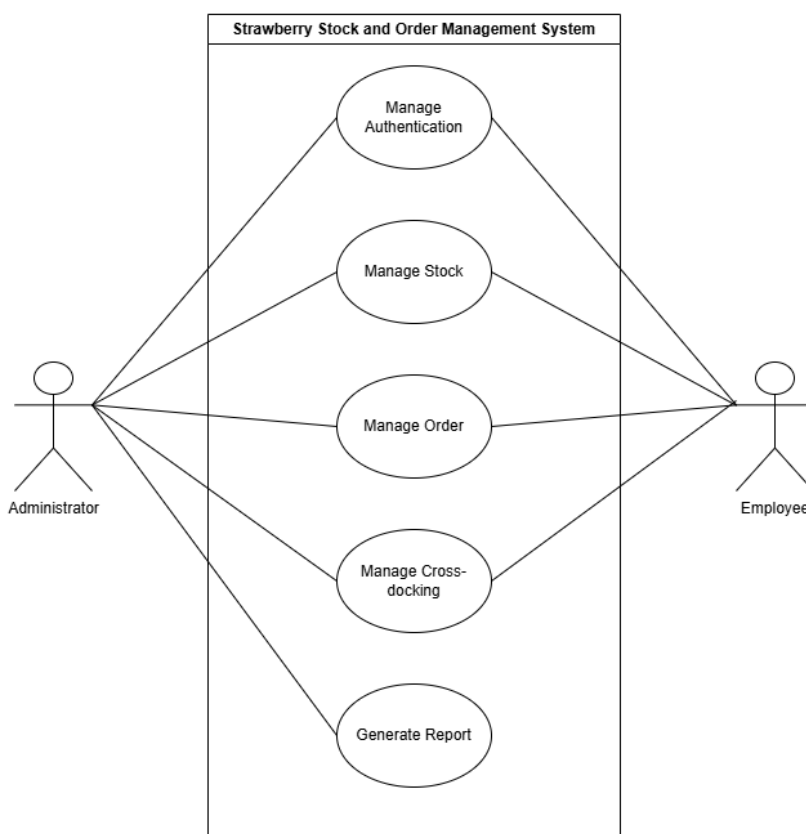
In user requirement analysis, system users' demands and expectations for system functionalities are considered and defined. The requirement is studied and identified from the system user's perspective. Table 4.3 displays the user requirements for the developed system.

**Table 4.3** System user requirements

No.	User requirements
1.	Both the administrator and the employee must be able to enter the system using a valid username and password.
2.	The Administrator must be able to register the employee in the system by giving the necessary information.
3.	The Administrator will be able to view every employee details.
4.	The Administrator must be able to edit employees information and roles in the system.
5.	Both the administrator and the employee must be able to scan the barcode with a barcode scanner and input it into the system.
6.	Both the Administrator and the employee will be able to add new stock information.
7.	Both the administrator and the employees must be able to update current strawberry stock during stock check-in and check-out.
8.	Administrators and employees will be able to examine and search for current strawberry stock information.
9.	The Administrator will be able to receive notifications about low stock levels.
10.	The Administrator will be able to generate and view the monthly financial data report.

## 4.2 System Analysis

A UML diagram showing interactions between the system users which are Administrator and Employees, while the system use case manage authentication, manage stock, manage order, manage cross-docking and generate report.



**Fig. 1** Use Case Diagram of System

## 4.3 System Design

The flowchart illustrates the flowchart, class diagram, sequence diagram and activity diagram of the Strawberry Stock and Order Management System, detailing the sequence of actions and decision points for various functionalities. It provides a clear visual guide for understanding how users interact with the system based on their roles (see Appendix A).

## 5. Implementation and Testing

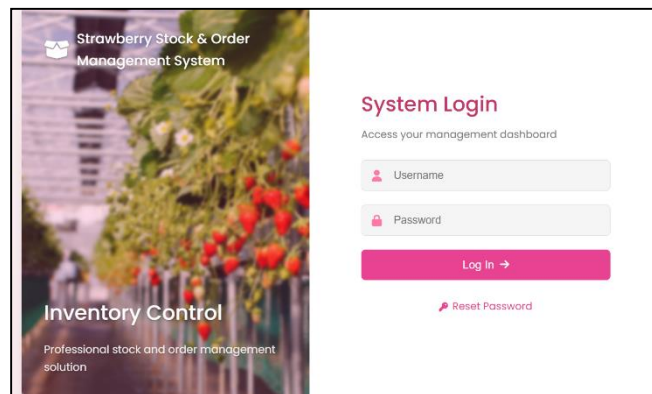
This section will discuss the implementation and testing of the system.

### 5.1 Implementation

The system development implementation and testing phases are covered in this chapter. The implementation covers where the various system interfaces for the main modules are displayed. The Python with flask framework, HTML and CSS programming languages were used to create the system from scratch.

#### 5.1.1 Manage Authentication Module

Logging into the system requires the user to enter their username and password. A redirect to the homepage occurs if the username and password match. It fails to log in otherwise.



**Fig. 2** Login Interface

Figure 2 displays the system's login interface, where users are asked to enter their login information. Both employees and administrators will find the interface easy to use because it is straightforward. If necessary, there is also the option to reset the password in order to recover the account.

#### 5.1.2 Manage Stock Module

Strawberry stock can be added, updated, and tracked using the Manage Stock module. This module keeps an accurate record of all stock-ins and stock-outs and applies FIFO logic to minimize spoiling. It is made to make sure that admin and employees can handle real-time stock activity with ease.

**Fig. 3** Check in Stock Interface

Figure 3 shows the Check in Stock Interface, where users can create or manually insert strawberry crates' barcodes. The form has sections for choosing the grade, entering the date of harvest, and stating the amount in kilograms. With its own barcode, the system makes sure that every incoming stock item is precisely documented and prepared for tracking. By keeping accurate records from the time produce enters the system, this function helps to ensure inventory freshness.

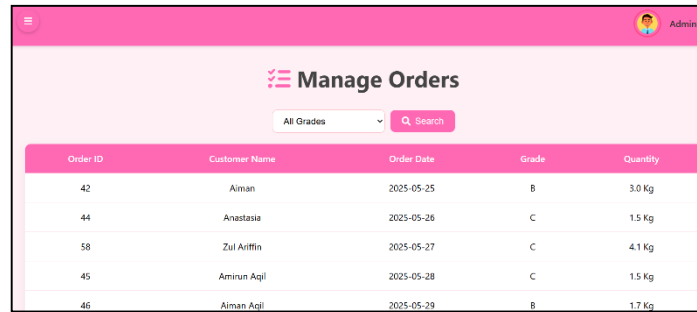
Order ID	Customer	Requested	Barcode	Harvest Date	Available
42	Aiman	3.0 kg	BR4908	2025-05-28	5.0 kg
44	Anastasia	1.5 kg	BR1163	2025-05-27	3.0 kg
45	Aminun Aqil	1.5 kg	BR1163	2025-05-27	3.0 kg

**Fig. 4** Stock Out Interface

Figure 4 shows the Stock Out Interface, which is utilized while fulfilling orders. The customer's name, order amount, and the system-recommended stock batch based on FIFO logic are among the comprehensive order details displayed. The interface minimizes spoilage and overstocking while guaranteeing precise stock assignment to complete each transaction. Batch selection for shipment and order review are made easier with each card-style layout.

### 5.1.3 Manage Order Module

Users can create, monitor, and handle customer orders using the Manage Order module. The system allows FIFO-based inventory assignment and verifies stock availability prior to order confirmation. This module offers transparent order status tracking and guarantees proper order processing.



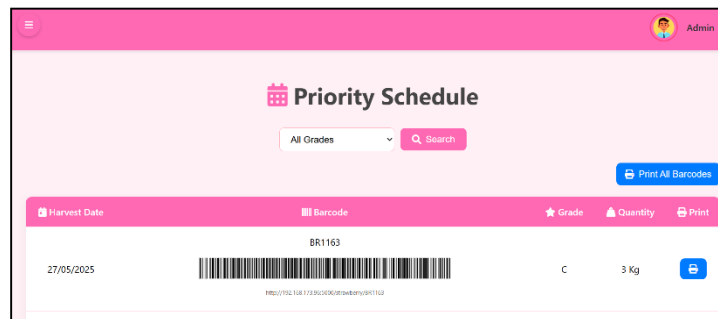
Order ID	Customer Name	Order Date	Grade	Quantity
42	Aiman	2025-05-25	B	3.0 Kg
44	Anastasia	2025-05-26	C	1.5 Kg
58	Zul Aiffin	2025-05-27	C	4.1 Kg
45	Amirun Aqil	2025-05-28	C	1.5 Kg
46	Aiman Aqil	2025-05-29	B	1.7 Kg



**Fig. 5 Admin Manage Order Interface**

Figure 5 shows how admin may monitor all incoming orders using the Admin Manage Order Interface. Important information including order ID, customer name, order date, grade, and quantity are provided in the list format. Admins can use the toolbar to search for specific entries or filter orders by grade. This simplified view facilitates effective order management and monitoring.

#### 5.1.4 Manage Cross-docking Module

High-priority strawberry batches can be processed quickly as a result to the Cross-docking module, which transfers them straight from receipt to shipping. It facilitates barcode tracking, improves batch traceability, and shortens storage times.



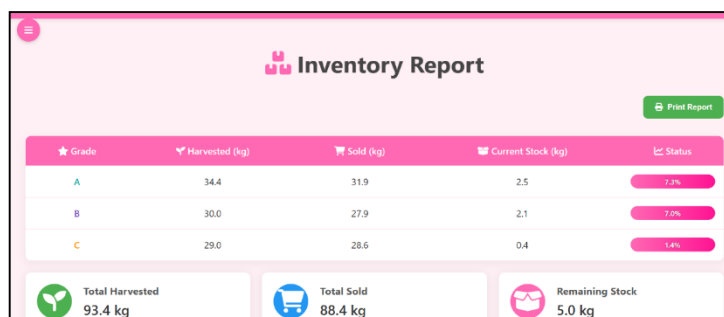
Harvest Date	Barcode	Grade	Quantity	Print
27/05/2025	BR1163 	C	3 Kg	

**Fig. 6 Priority Schedule Interface**

Figure 6 displays the Priority Schedule Interface, which assigns priority to batches of harvested strawberries before they are shipped. The harvest date, barcode, grade, quantity, and the ability to print barcode labels are all included in each entry. By assisting employees rapidly determining which batches should be given priority according to freshness, this module facilitates cross-docking processes. By allowing for instant crate identification and cutting down on storage time, it improves efficiency.




#### 5.1.5 Generate Report Module

This module gives admin the ability to create several reports such as inventory report, order fulfilment report and financial reports. Decision-making based on data and operational transparency are supported by these reports.



Grade	Harvested (kg)	Sold (kg)	Current Stock (kg)	Status
A	34.4	31.9	2.5	7.8%
B	30.0	27.9	2.1	7.0%
C	29.0	28.6	0.4	1.4%

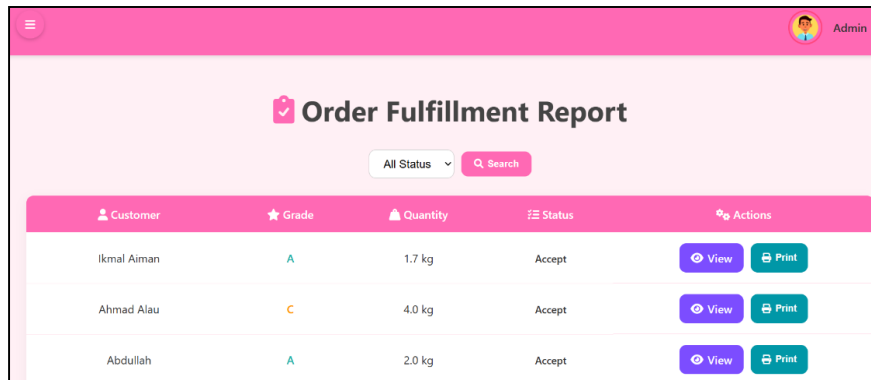
  

 Total Harvested 93.4 kg	 Total Sold 88.4 kg	 Remaining Stock 5.0 kg
--	---	---

**Fig. 7 Inventory Report Interface**

Figure 7 displays the Inventory Report Interface with a summary of the stock that has been harvested, sold, and remained behind. Along with visual stock status indicators to show current inventory levels, it displays

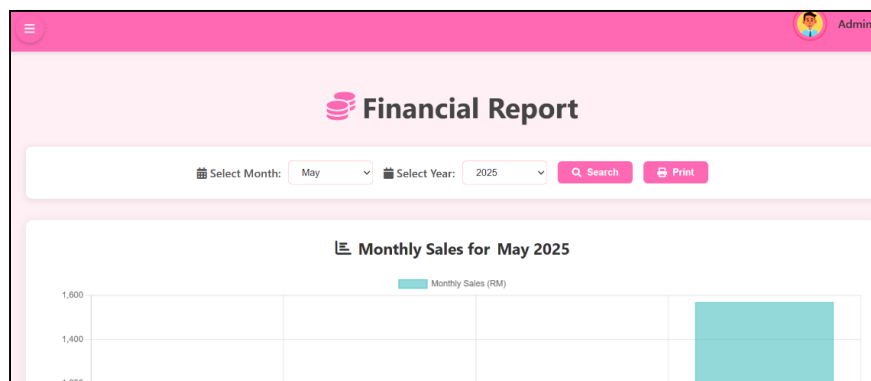
data by grade. This assists planning decisions and aids users in understanding stock trends. Quick statistics, including total harvested, sold, and available stock, are given at the bottom.



Customer	Grade	Quantity	Status	Actions
Ikmal Airman	A	1.7 kg	Accept	<a href="#">View</a> <a href="#">Print</a>
Ahmad Alau	C	4.0 kg	Accept	<a href="#">View</a> <a href="#">Print</a>
Abdullah	A	2.0 kg	Accept	<a href="#">View</a> <a href="#">Print</a>

**Fig. 8** Order Fulfillment Report Interface

Figure 8 shows all orders by customer, grade, amount, and status in the Order Fulfillment Report Interface. Users can view complete order details or print order records for documentation by using the action buttons. To make it simple to track accepted versus declined orders, the system filters orders according to their status. This view guarantees that every order is accounted for and facilitates performance analysis.



**Fig. 9** Financial Report Interface

Figure 9 summarizes monthly sales according to a chosen month and year, using the Financial Report Interface. The graph shows the total sales for the chosen time period in RM. Administrators are more capable of assessing revenue growth and make wise business decisions because of this. For printed or PDF reports, the interface offers a print option.

## 5.2 Testing

The system is put through testing to make sure it satisfies the requirements. Black box testing and user acceptance testing are the two forms of testing that are conducted. This section displays and discusses the testing's test cases and results.

### 5.2.1 Black Box Testing

The test cases are derived based on the functional requirements defined. Table 5 shows the test cases information. It focuses on input and output validation by testing the behavior of the system under different conditions. Each test case is designed to confirm whether the expected output is produced based on the given inputs.

**Table 5** Requirement Traceability Matrix

Software Requirement Specification	Test Case ID	Test Case Description	Expected Result	Test Result (Pass/Fail)
SRS_REQ_101, SRS_REQ_102	TEST_100_001	User enters valid username and password, system redirects to homepage.	System displays role-based homepage.	Pass
SRS_REQ_103	TEST_100_002	User enters invalid username/password.	System shows error message.	Pass
SRS_REQ_104	TEST_100_003	User clicks reset password without entering username.	System prompts to enter username.	Pass
SRS_REQ_201	TEST_200_001	Add new stock with valid barcode and data.	Stock successfully added and recorded.	Pass
SRS_REQ_301	TEST_300_001	Create an order with sufficient stock.	Order created and inventory updated.	Pass
SRS_REQ_401	TEST_400_001	Scan older crate for cross-docking.	Crate selected as high-priority item.	Pass
SRS_REQ_501	TEST_500_001	Generate report for selected month.	System generates and displays report.	Pass

The system operated as anticipated in every functional scenario, according to the results of all black box testing. The accuracy of system behaviour was confirmed by each test case, which was derived from the specified software requirements and generated the appropriate output based on the supplied inputs. All test cases were marked as "Pass," indicating that the login, stock management, order processing, cross-docking, and report creation modules had all undergone successful validation. Furthermore, an actual employee's User Acceptance Test (UAT) (see Appendix E) verified that the system is responsive, user-friendly, and meets user expectations. According to these findings, the system is operationally stable.

## 6. Conclusion

The Strawberry Stock and Order Management System successfully addresses the inefficiencies of manual inventory and order processing at Mr. Khaw's farm. The system increases efficiency, lowers human error, and guarantees on-time order fulfillment by utilizing barcode technology and real-time data processing. The successful outcomes of the black box testing procedure verify to the effective implementation of critical strategies like automated order prioritizing, Cross-Docking for high-priority order handling, and First-In, First-Out (FIFO) for stock rotation. The integration of a mobile application that allows field-level data access and input could be one of the future enhancements. Furthermore, demand forecasting and stock planning could be further optimized by incorporating advanced predictive analytics. The system might offer wise suggestions on inventory levels by examining past sales trends and seasonal patterns, assisting the farm in proactively managing supply and reducing losses. These improvements would guarantee long-term operating efficiency and improve the system's scalability.

## Acknowledgement

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

## Conflict of Interest

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

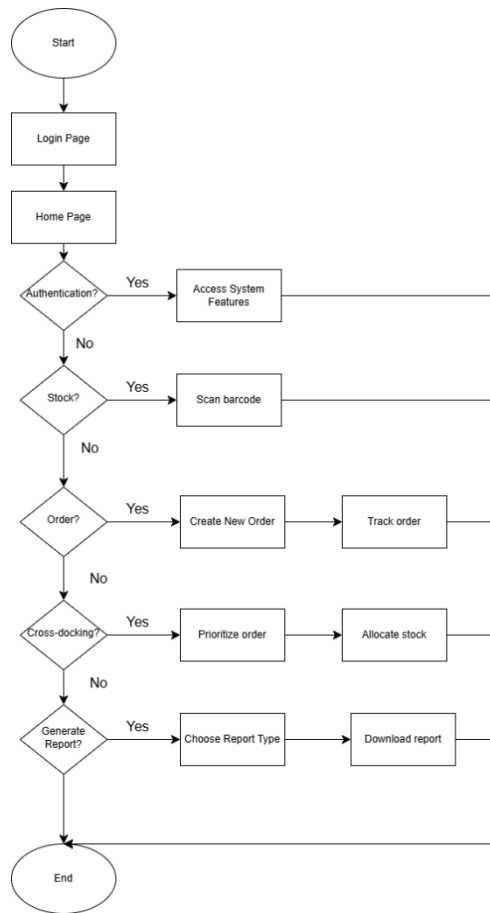
## Author Contribution

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

## References

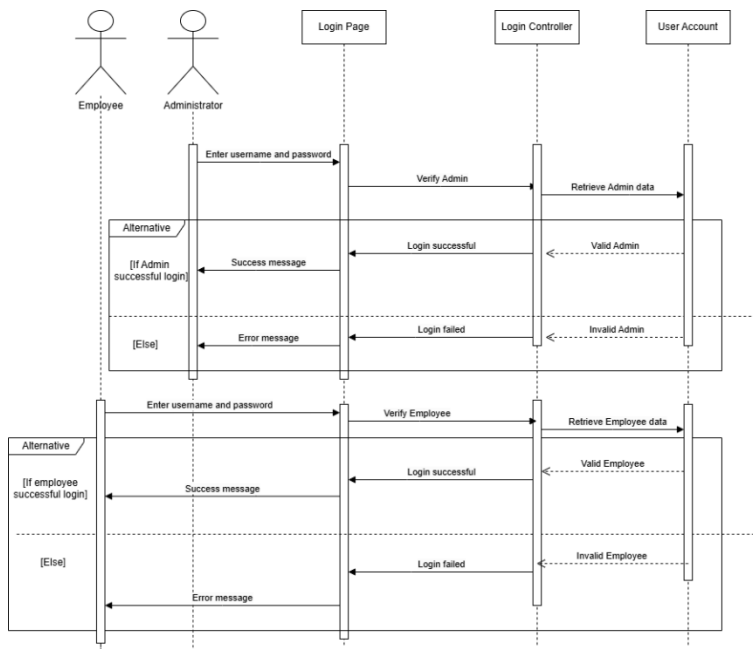
- [1] Bakker, N. (2024, August 30). Top Advanced features in modern ERP Systems (Explained for Beginners). News About Global E-invoicing. <https://www.storecove.com/blog/en/advanced-features-in-modern-erp-systems/>
- [2] *Plant Nursery Inventory Management | Mprise Agriware*. (n.d.). Mprise Agriware. <https://www.mprise-agriware.com/inventory-management>
- [3] *Farm Inventory Management*. (n.d.). AgriERP. <https://agrierp.com/product-features/farm-inventory-management/>
- [4] Gopalms Solutions – Manage Your Plantation Easily. (2022). Gopalms.com. <https://gopalms.com/>
- [5] Panganiban, E. B., & Bermusa, J. P. (2020). Simplified Barcode-Based Point of Sales and Inventory Management System with replenishment decision. *International Journal of Scientific and Technology Research*, 9(3), 597–600. <https://www.ijstr.org/final-print/mar2020/Simplified-Barcode-based-Point-Of-Sales-And-Inventory-Management-System-With-Replenishment-Decision.pdf>
- [6] Eood, P. (2024, November 28). Importance of barcodes in food processing and traceability. barkoder. <https://barkoder.com/blog/the-importance-of-barcodes-in-food-processing-and-traceability>
- [7] Dyouri, A. (2020, April 17). *How To Make a Web Application Using Flask in Python 3*. DigitalOcean. <https://www.digitalocean.com/community/tutorials/how-to-make-a-web-application-using-flask-in-python-3>
- [8] GeeksforGeeks. (2024, October 4). Prototyping model software engineering. GeeksforGeeks. <https://www.geeksforgeeks.org/software-engineering-prototyping-model/>
- [9] GeeksforGeeks. (2023, September 20). *Difference between Waterfall model and Prototype model*. GeeksforGeeks. <https://www.geeksforgeeks.org/difference-between-waterfall-model-and-prototype-model/>
- [10] Cwi\_Phoscreative. (2024, September 5). *Cross-Docking: The Basics (Definition, Types, Advantages) - CWI Logistics*. CWI Logistics. <https://www.cwi-logistics.com/news/the-basics-cross-docking/>

### Appendix A: Flowchart

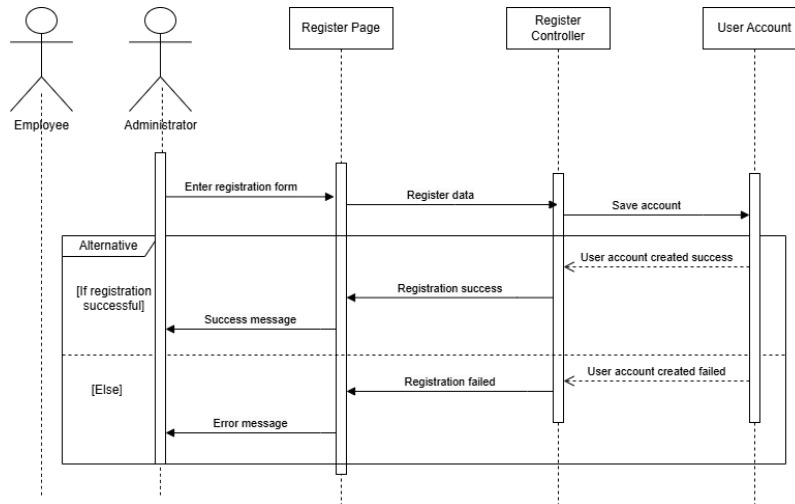


Flowchart diagram

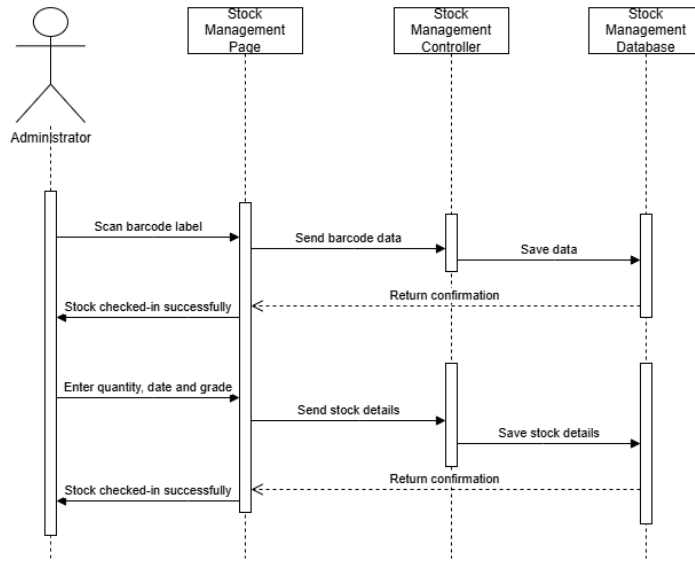
### Appendix B: Sequence Diagram



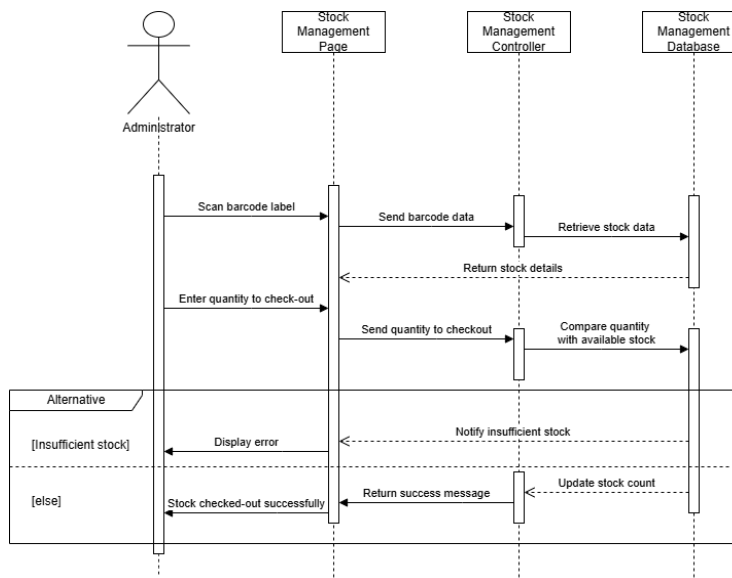
Login use case Sequence Diagram



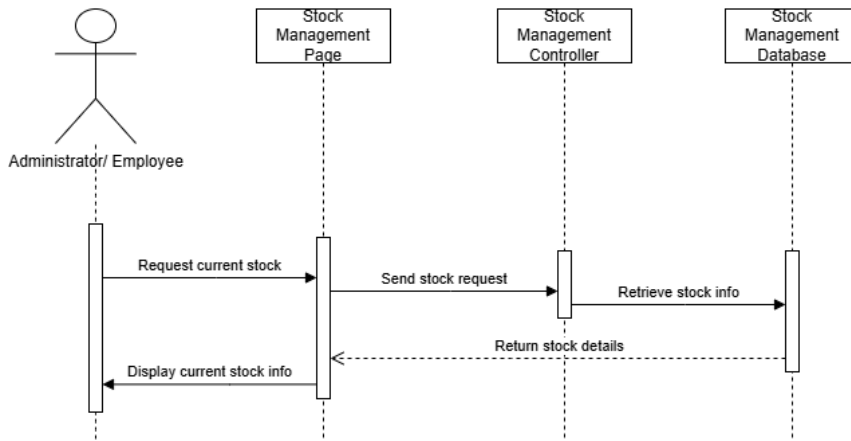
Register Use Case Sequence Diagram



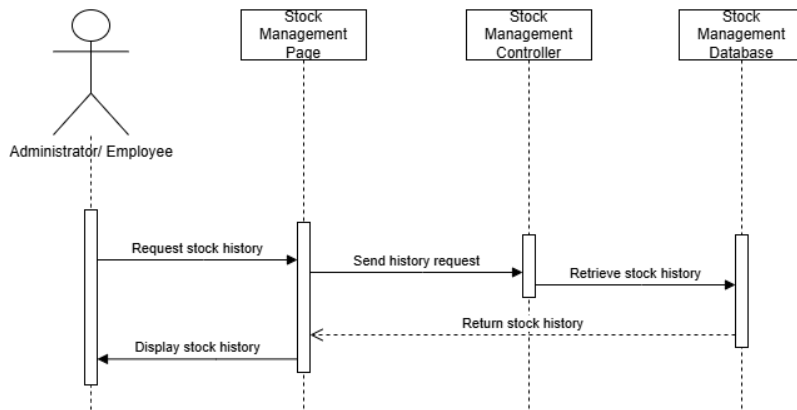
Manage Stock Use Case Sequence Diagram (Manage Stock Checked-in)



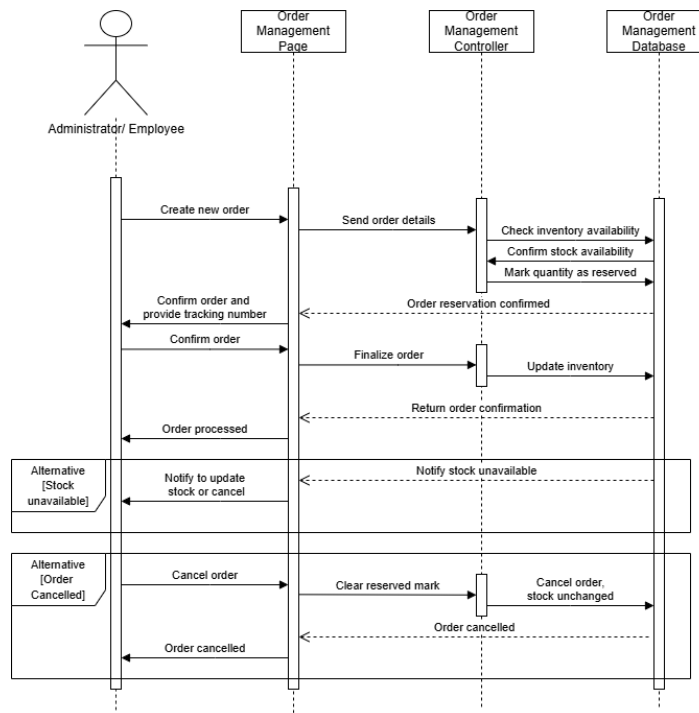
Manage Stock Use Case Sequence Diagram (Manage Stock Checked-out)



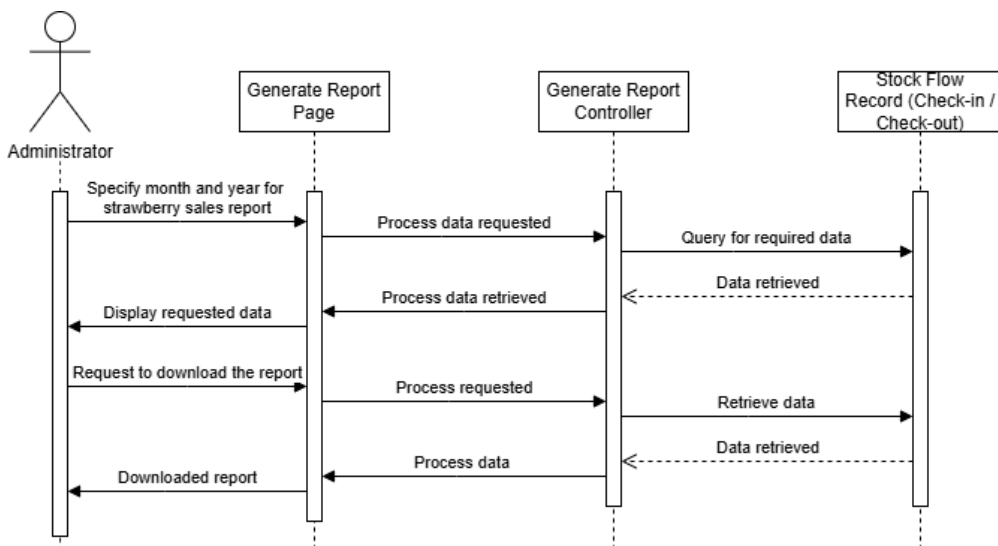
Manage Stock Use Case Sequence Diagram (View current stock)



Manage Stock Use Case Sequence Diagram (View stock history)

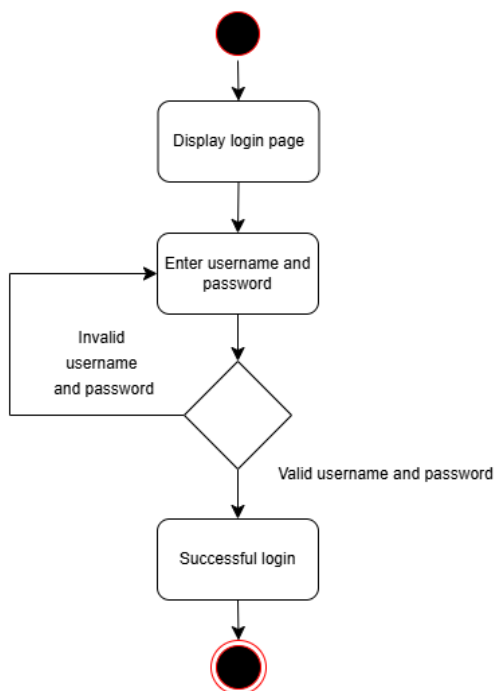


Order Management Use Case Sequence Diagram

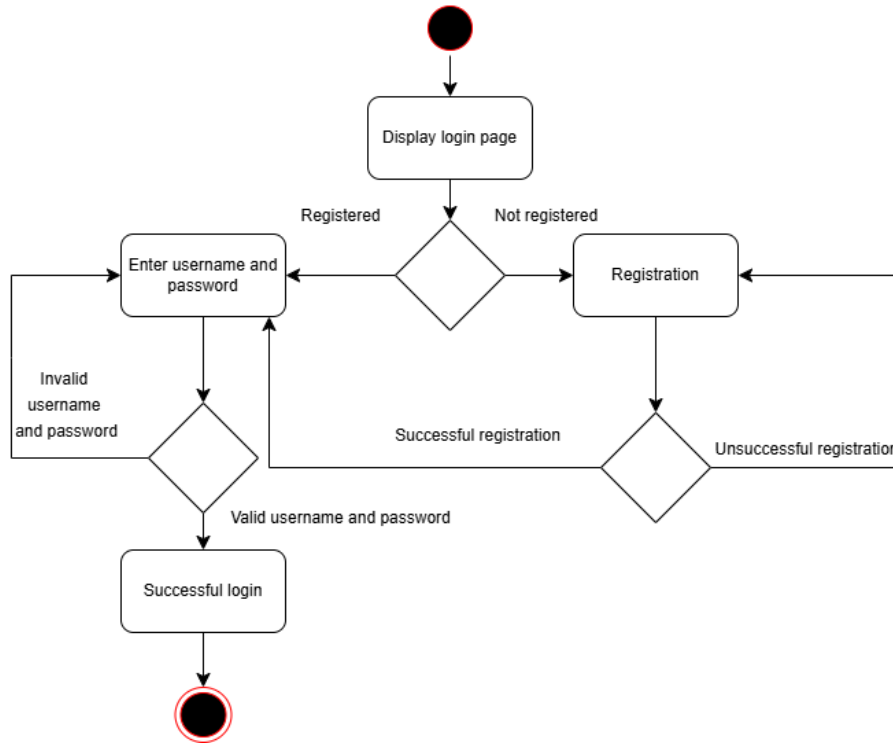


Generate Report Use Case Sequence Diagram

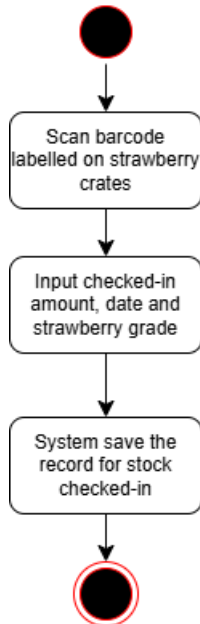
### Appendix C: Activity Diagram



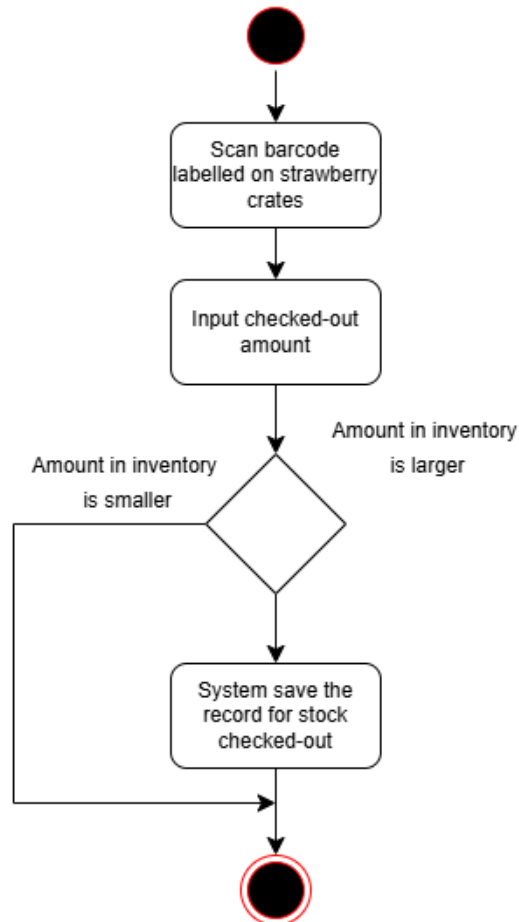
Login Use Case Activity Diagram



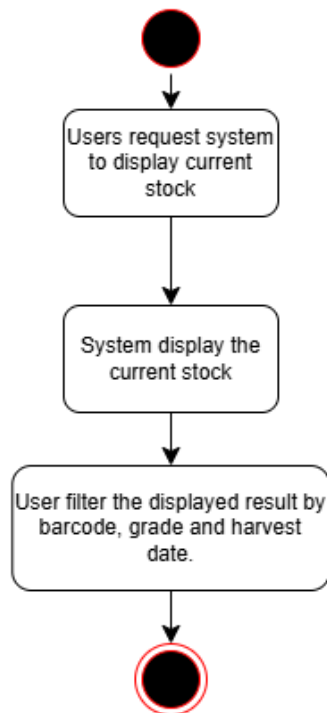
Register Use Case Activity Diagram



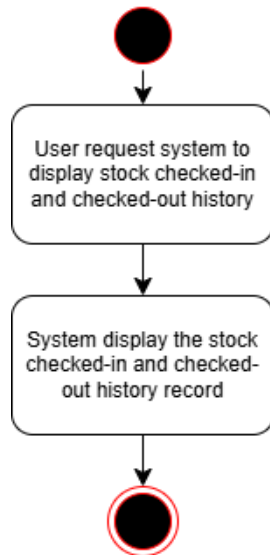
Manage Stock Use Case Activity Diagram (Manage Stock Checked-in)



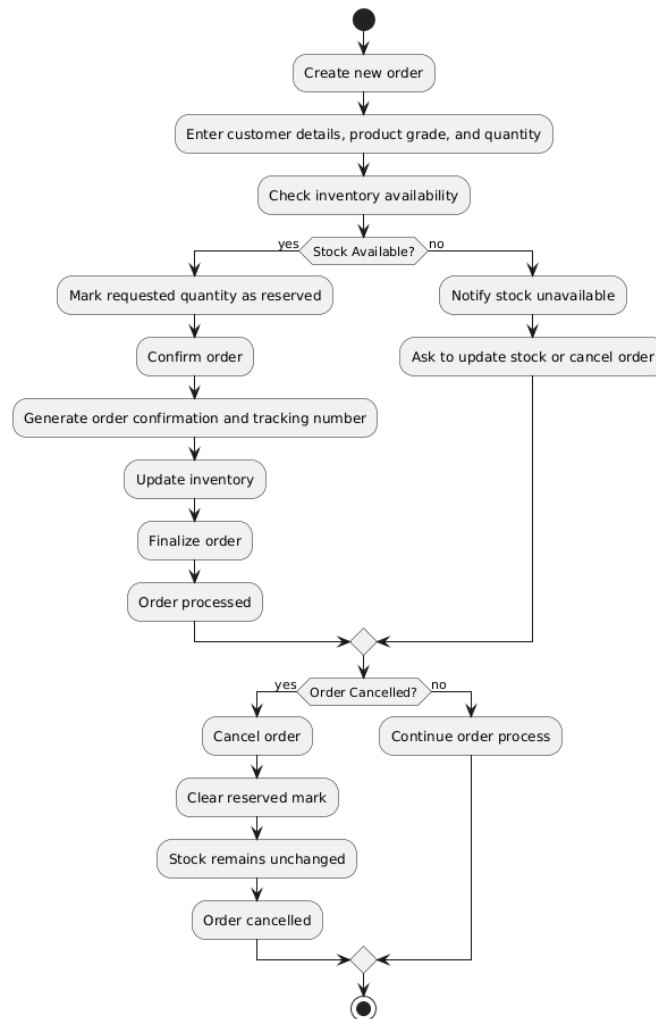
Manage Stock Use Case Activity Diagram (Manage Stock Checked-out)



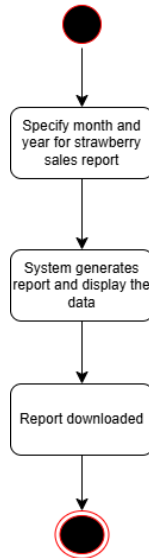
Manage Stock Use Case Activity Diagram (View current stock)



Manage Stock Use Case Activity Diagram (View stock history)

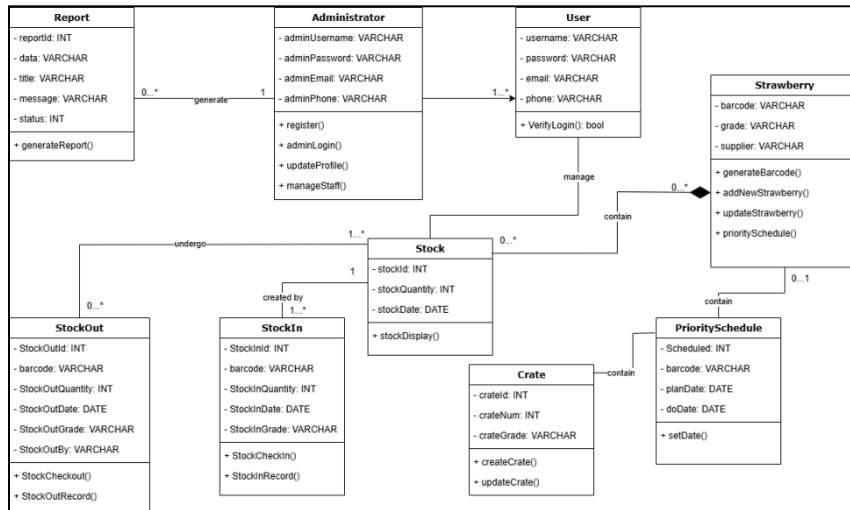


Order Management Use Case Activity Diagram



Generate Report Use Case Activity Diagram

Appendix D: Class Diagram



Class Diagram

Appendix E: User Acceptance Testing

User Acceptance Test (UAT) Form				
Project Name		Strawberry Stock and Order Management System		
Tested By		T. H. K. A. F. I.		
Role		Administrator / Employee		
Date		27/6/2025		
Key Test Scenarios				
No.	Module	Scenario	Expected Result	Pass/Fail
1	Login	Login with valid credentials	Redirects to dashboard	Pass
2	Stock Management	Add new stock	Stock is added and visible	Pass
3	Stock Management	Check-out stock within limit	Updates stock correctly	Pass
4	Order Management	Place an order with enough stock	Order is confirmed and tracked	Pass
5	Report Generator	Generate monthly report	Report is downloaded successfully	Pass
User Feedback				
Please rate the following statements:				
Statement	Rating	Comment (Optional)		
The system is easy to use.	<input checked="" type="checkbox"/> Very Easy <input type="checkbox"/> Easy <input type="checkbox"/> Difficult <input type="checkbox"/> Very Difficult			
Navigation is intuitive.	<input checked="" type="checkbox"/> Very Easy <input type="checkbox"/> Easy <input type="checkbox"/> Difficult <input type="checkbox"/> Very Difficult			
System features meet my job needs.	<input type="checkbox"/> Very Easy <input checked="" type="checkbox"/> Easy <input type="checkbox"/> Difficult <input type="checkbox"/> Very Difficult			
System is responsive and not laggy.	<input checked="" type="checkbox"/> Very Easy <input type="checkbox"/> Easy <input type="checkbox"/> Difficult <input type="checkbox"/> Very Difficult			
Sign-Off				
Tester Name		T. H. K. A. F. I.		
Signature				
Date		27-6-2025		

User Acceptance Testing