

MyIndah Pharmacy Inventory Management System

Yuvadharshini Selvarathnam¹, Mohd Amin Mohd Yunus¹

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

*Corresponding Author: aminy@uthm.edu.my

DOI: <https://doi.org/10.30880/aitcs.2025.06.01.107>

Article Info

Received: 20 February 2025

Accepted: 18 June 2025

Available online: 30 June 2025

Keywords

Pharmacy inventory management, web-based system, waterfall methodology, Inventory processes, automated notification, project phases.

Abstract

The MyIndah Pharmacy Inventory Management System (MI-PIMS) is a web-based solution designed to revolutionize inventory management in pharmacies. The system aims to address issues such as time-consuming manual order processing, inefficient stock tracking, and systemic delays. It includes features like real-time visibility of stock levels, automated low-inventory notifications, seamless integration with point-of-sale terminals, and comprehensive reporting tools. The goal is to enhance operational efficiency, accuracy, and customer satisfaction while easing current inventory management challenges. The project utilizes advanced technology to address problems associated with traditional inventory procedures, employing a methodical approach with structured design and exact user acceptance testing to ensure a reliable and error-free system. MI-PIMS is designed to accommodate the unique needs of various retail sectors through scalability and customization, promoting improved operational agility and performance. The project is expected to transform inventory management, ushering in a new era of accessible, accurate, and efficient inventory data management.

1. Introduction

MyIndah Pharmacy, established on June 16, 2022, by Dato' Dr. Ilairaja A/L Sagadewan, aims to provide easy access to medical treatment, medications, and supplements for residents in the Lunas and Kulim areas. Currently, the pharmacy uses manual methods for daily operations like tracking inventory, managing purchase orders, updating stock, generating sales reports, and notifying staff about low stock levels. These manual processes are time-consuming and prone to errors, leading to inefficiencies, mistakes in stock counts, and delays in order processing. To address these issues, MyIndah Pharmacy is implementing the MyIndah Pharmacy Inventory Management System (MI-PIMS). This system will automate stock management processes, providing real-time information on inventory levels and reducing errors. MI-PIMS will help prevent overstocking and stockouts by maintaining optimal stock levels. The system will also integrate with point-of-sale terminals, ensuring seamless transactions and automatic inventory updates. This integration minimizes discrepancies between actual stock levels and recorded data, and the system's ability to generate accurate sales reports will improve financial management and strategic planning. Moreover, MI-PIMS will reduce reliance on paper-based records and manual data entry, saving time and labor costs while securely storing data. By enhancing operational efficiency and accuracy, MI-PIMS will transform the pharmacy's inventory management and daily operations, leading to improved business performance.

1.1 Problem Statement

MyIndah Pharmacy faces inventory management challenges. Manual processing of supplier orders causes delays, as there's no central system for tracking stock levels or identifying high-demand items. Staff waste time manually checking inventory, and the manager waits longer for orders because suppliers manually confirm stock. The lack of an inventory tracking system forces suppliers to physically visit the pharmacy to check stock levels, confirm inventory, and place orders. This manual process is time-consuming and error-prone, with discrepancies between recorded and actual stock levels often going unnoticed until the visit. This leads to delays in order placement and increases the risk of stockouts or overstock situations, highlighting the inefficiencies in the current inventory management approach. Limited visibility into stock levels is a big problem for pharmacies. When they don't have quick access to data about their products, it's hard to guess how much they'll need. This can lead to too much or insufficient stock, which messes up how things run and costs money. Without knowing exactly what they have at any time, pharmacies can't plan well, which causes issues with medicine stock and money management. Ensuring they can see their stock levels better is super important for pharmacies to manage things well and avoid these problems. MyIndah pharmacy needs to use a better system for keeping track of its stock at the pharmacy. This new system will make things easier for managers and help the pharmacy run better. It will save time, simplify managing stock, and fix the current issues. Plus, it will help the pharmacy use its resources better and prepare for future growth.

1.2 Objective

The objectives of developing the MI-PIMS are as follows:

- i. To design the MI-PIMS using the structured approach.
- ii. To develop an MI-PIMS by using waterfall methodology.
- iii. To test the function of an MI-PIMS using user acceptance testing successfully.

1.3 Scope Project

An extensive web-based system created to optimize and simplify an organization's inventory management processes is called the MyIndah Pharmacy Inventory Management System (MI-PIMS). This project encompasses the following key features and functionalities. The system includes three user roles which are manager, staff, and suppliers. On the other hand, the manager is responsible for managing the entire system. While the staff and supplier are involved in day-to-day operations related to pharmacy management. Table 1 shows the modules of the system.

Table 1 Modules system of MyIndah Pharmacy Inventory Management System

Modules	Description
Login	Users can enter the system by entering their username and password
Manage Stock	Add, edit, or delete stock
Manage Staff	Add, edit, or delete staff
Manage Supplier	Add, edit, or delete the supplier
Manage Profile	View and update profile information
Order Request	Add order for restocking inventory item
Manage Purchase Order	The supplier will accept or reject the incoming request
Sales List	Allow the manager and staff to process the transaction
Notification	Receive notifications for low stock levels, and pending orders alerts through the system
Payment	Allow manager to make payment once supplier has accepted the order
Manage Category	Add, edit, or delete categories
Generate Report	View sales report and revenue by day, week, year, and month

2. Related Work

This section discusses the current understanding of a computerized system and compares it with the proposed system through a literature review. The review aims to gather data and identify issues, while also serving as the project's source material. The literature review helps understand linked initiatives and solve important issues. CX Remedi Clinic Inventory System, Zoho Inventory & Trade Gecko Inventory and MyIndah Pharmacy Inventory Management System are the systems that are compared in this section.

2.1 Trade Gecko Inventory

Trade Gecko was a cloud-based inventory and order management tool for small and medium-sized businesses founded in Singapore. It optimized inventory, orders, and sales across platforms like Shopify and Amazon, and included features like multi-location tracking, real-time stock updates, and external interfaces. It was acquired by Intuit in 2020 and called QuickBooks Commerce before being deactivated in 2022. Users are now considering options such as Ordoro, Zoho Inventory, and Cin7.[3]

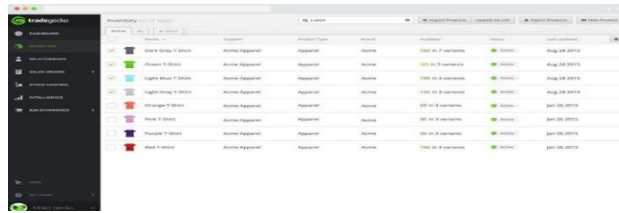


Fig. 1 Homepage for Trade Gecko Inventory

2.2 Zoho Inventory

Zoho Inventory offers various features tailored for small to medium-sized enterprises, simplifying inventory management [1]. Features include robust inventory monitoring with batch and serial number tracking, buy order, and backorder management. These features support our goals for the proposed Zoho Inventory System, which is to improve the accuracy and efficiency of inventory management. By making use of Zoho Inventory's capabilities, our system improves productivity, lowers error rates, and provides deeper insights into sales success. The Finance Suite's integration with Zoho guarantees smooth coordination with accounting and bookkeeping responsibilities, enabling users to maximize inventory management and maintain exact client information [7]. As shown in Fig. 2, the Homepage for Zoho Inventory provides a comprehensive and user-friendly interface that enhances these functionalities.

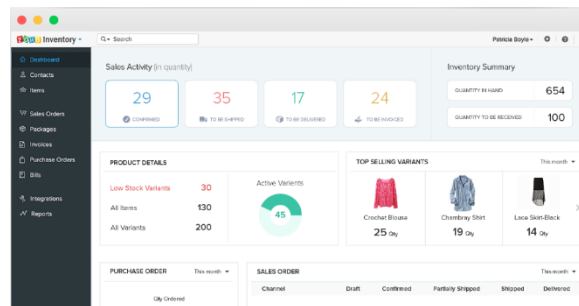


Fig. 2 Homepage for Zoho Inventory

2.3 CX Remedi Clinic Inventory System

The CX Remedi Clinic Inventory System streamlines clinic operations with appointment management, EHRs, billing, inventory, and reporting tools (REMEDi Medical Portal, 2024). It automates scheduling, maintains detailed patient records, and facilitates billing processes. The operation of the supply chain is made more efficient by inventory management, and performance is enhanced by reporting solutions that track key performance indicators [2]. The integration with other healthcare systems aims to improve productivity, accuracy, and patient happiness within clinics by ensuring smooth data transmission and compliance with laws like HIPAA. As shown in Fig. 3, the CX Remedi Clinic Inventory System Homepage provides an intuitive interface that effectively supports these functionalities.

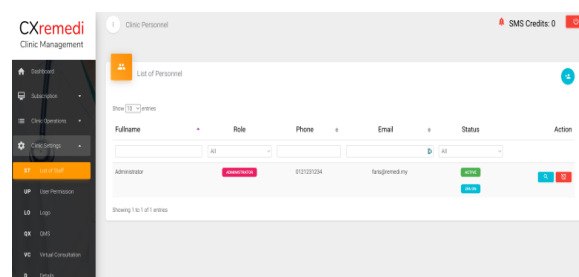


Fig. 3 Homepage for CX Remedi Clinic Inventory System

2.4 Comparison between related system

The My Indah Pharmacy Management System (MI-PMS) stands out from competitors like Trade Gecko, Zoho Inventory, and CX Remedi Clinic Inventory System with its extensive inventory management features. Unlike its competitors, MI-PIMS includes order requests, request management, stock control, staff and supplier management, sales monitoring, notifications, and report generation. Its user-friendly, web-based platform is accessible from anywhere. Utilizing PHP, HTML, CSS, JavaScript, and SQL, MI-PIMS ensures reliable performance and scalability. Table 2 compares MI-PIMS with existing systems.

Table 2 Comparison between the proposed system and the existing systems

Comparison/ System	Trade Gecko Inventory	Zoho Inventory	CX Remedi Clinic System	MI-PIMS
<i>Modules:</i>				
Login	Yes	Yes	Yes	Yes
Manage Stock	Yes	Yes	Yes	Yes
Manage Staff	Yes	No	Yes	Yes
Manage Supplier	No	No	Yes	Yes
Manage Profile	Yes	Yes	Yes	Yes
Order Request	Yes	Yes	Yes	Yes
Manage Purchase Order	No	Yes	No	Yes
Payment	No	No	No	Yes
Manage Category	Yes	Yes	Yes	Yes
Sales List	Yes	No	No	Yes
Notification	Yes	Yes	No	Yes
Generate Report	Yes	Yes	Yes	Yes

3. Methodology

Table 3 outlines the tasks for each stage of the waterfall model, which includes five phases: requirements and analysis, design, implementation, testing, and maintenance [4]. I have included a Gantt chart in the appendix to visually represent the sequential tasks and their timeframes within each phase. This chart acts as a roadmap for the project, helping team members and stakeholders understand task dependencies, durations, and the overall timeline.

Table 3 Software development activities task

Phases	Task/Activities	Deliverables	Tools
Requirement & Analysis	Identify system scope and requirement Analyze requirement: Data collection – interview client to identify system requirement Project timeline – Gantt Chart	i. Proposal ii. Gantt Chart	i. MS Word ii. TeamGantt.com
Design	Design user interface as per requirement Design database as per requirement Design Data Flow Diagram (DFD) and Entity Relationship Diagram (ERD)	i. Context Diagram ii. Data Flow Diagram iii. Entity Relationship Diagram iv. Class Diagram v. Schema table vi. Interface	i. Draw.io ii. PowerPoint

Table 3: Continued

Implementation	The designated design will be converted to actual code	i. To execute the system	i. Visual Studio Code ii. XAMPP Control Panel
System Testing	User acceptance testing	Test the functionalities and acceptance test	i. Visual Studio Code ii. XAMPP Control Panel
Maintenance	Involve correcting errors and upgrading due to current trends and needs	Make sure the system is ready to use by the stakeholder and ensures the quality of use which provides effectiveness, efficiency, satisfaction, reliability and secure	i. Visual Studio Code ii. XAMPP Control Panel

4. Analysis and Design

This section will discuss the analysis and design phases of the system, covering system requirement analysis, which includes both functional and non-functional requirements, database design, and implementation. The design phase involves detailing the components and interfaces to address these requirements and constraints.

4.1 System Requirement Analysis

The process of identifying the needs that a constructed system needs to meet and the expectations of users for the suggested system is known as requirement analysis. The system requirements include user requirements, system requirements, and functional and non-functional requirements. The functional needs and the proposed system's description are shown in Table 4, while the non-functional requirements and their descriptions are shown in Table 5.

Table 4 Functional Requirements

Modules	Description
Login	<ul style="list-style-type: none"> The system should allow users to log in with a username and password The system should verify user credentials against stored database data. The system should show alerts for incorrect login attempts. The system should direct users to appropriate pages upon successful login based on their roles (Manager, Staff, or Supplier).
Manage Stock	<ul style="list-style-type: none"> The system should allow users to add new stock items, including product name, description, quantity, expiration date, and price. The system should allow users to edit details of existing stock items. The system should allow users to delete stock items that are no longer available.
Manage Supplier	<ul style="list-style-type: none"> The system should allow users to add and maintain a database of suppliers, including company name, contact person, address, phone number, email, and contact details The system should allow users to edit supplier details to keep information up-to-date. The system should allow users to delete suppliers no longer in service.
Manage Staff	<ul style="list-style-type: none"> The system should categorize suppliers based on the products or services they provide The system should allow users to add and maintain a staff database, including their name, address, phone number, and email. The system should allow users to edit staff details to keep information up-to-date. The system should allow users to delete staff that are no longer employed.

Table 4: Continued

Manage Profile	<ul style="list-style-type: none"> The system should allow users to view personal profile information including name, phone number, and role. The system should allow users to update profile details such as email, phone number, and address.
Order Request	<ul style="list-style-type: none"> The system should provide a change password feature for security purposes. The system should allow users to place orders for restocking stock items, specifying product details, and quantities. The system should track the status of each pending order.
Manage Purchase Order	<ul style="list-style-type: none"> The system should allow suppliers to receive incoming requests from managers for order approvals or specific action. The system should allow suppliers to accept or reject requests The system should notify managers and staff of the request status (pending, accepted, rejected or order shipped). The system should track the history and status of all requests. The system should allow managers to view the status and details of their submitted requests in real time.
Sales List	<ul style="list-style-type: none"> The system should provide a user-friendly interface for managers to add item The system should allow managers and staff to process transactions efficiently, including calculating totals of sales. The system should track daily sales and transaction details. The system should integrate with inventory management to automatically update stock levels after each sale.
Notification	<ul style="list-style-type: none"> The system should send automated notifications for low stock levels and pending order alerts via email.
Payment	<ul style="list-style-type: none"> The system should allow the manager to initiate payment after the supplier has accepted the order. The system should prompt the manager to input card details (credit/debit card number, expiration date, CVV) or integrate with a payment gateway for secure card transactions. The manager should be able to track the status of the COD payment in the system The system should ensure that payment information, especially card details, is securely
Generate Report	<ul style="list-style-type: none"> The system should allow users to access and generate detailed reports on sales and revenue by day, week, month, and year. The system should allow users to export reports in PDF format.

Table 5 Non-Functional Requirements

Modules	Description
Usability	<ul style="list-style-type: none"> The system's interface should be intuitive and user-friendly to reduce the learning curve for new users. The system should verify user credentials against stored database data. Provide clear error messages, tooltips, and prompts to help users
Performance	<ul style="list-style-type: none"> Standard queries should result in reports that are created and displayed within 5 seconds. The system should respond to user actions within 2 seconds for most operations.
Security	<ul style="list-style-type: none"> Encrypt user passwords for safe authentication.

4.2 System Analysis

A Data Flow Diagram (DFD) visually represents the data flow within an information system, illustrating its movement between internal processes, external entities, and data stores. The upcoming section dedicated to the MyIndah Pharmacy Inventory Management System (MI-PIMS) will present various diagrams, including the Context Diagram, DFD Level 0, and Entity Relationship Diagram (ERD).

4.2.1 Context Diagram

A context diagram is a visual representation of a system and its interactions with other components. It is beneficial for stakeholders who may not be familiar with the system or who need a quick overview of its overall functionality. Figure 5 illustrates the context diagram for the MI-PIMS. This diagram outlines the data input and output processes involving the administrator, staff, and supplier entities.

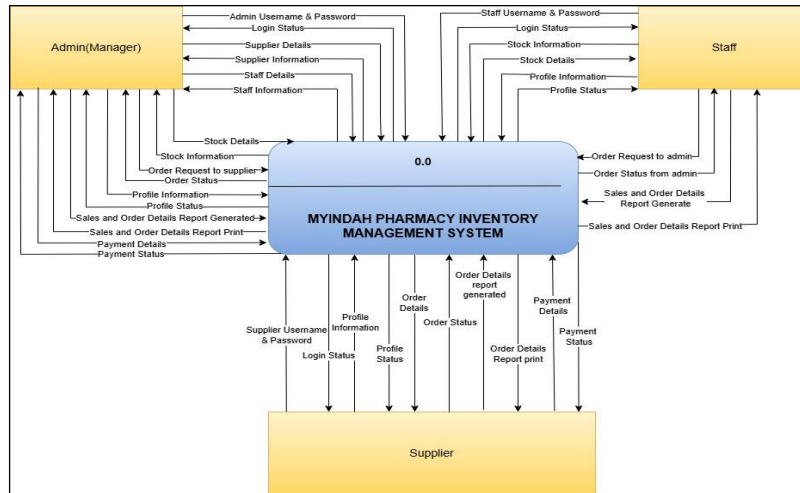


Fig. 4 Context Diagram (DFD CD)

4.2.2 DFD Level 0

Figure 6 shows the Level 0 Data Flow Diagram for the MI-PIMS. The data flows depicted in this diagram are described in detail in the accompanying Level 0 Data Flow Diagram. The relationships between each process and the entities at this level of the diagram are illustrated below the figure. Sample Data Flow Diagram Level 1 is provided in Appendix B.1

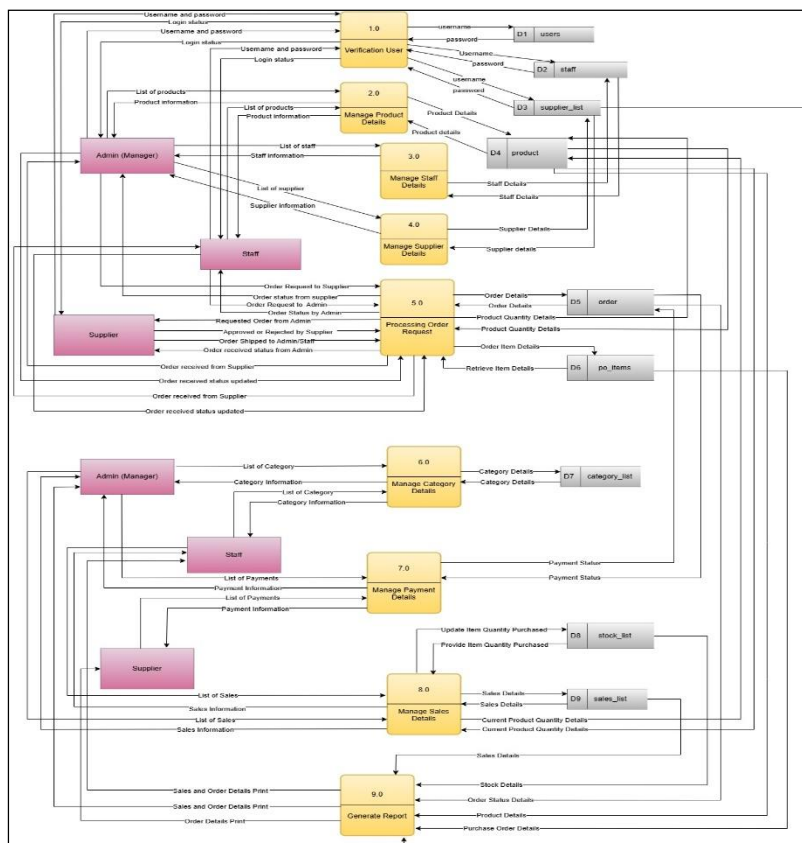


Fig. 5 Data Flow Diagram Level 0 (DFD 0)

4.2.3 Entity Relationship Diagram

An entity-attribute relationship diagram is a complete diagram that demonstrates the relationship between the two. An entity connection diagram is constructed to demonstrate how external entities, and the system are connected. The Entity Relationship Diagram below explains the relationship between the MI-PIMS and external entities such as suppliers, administrators, and staff [5].

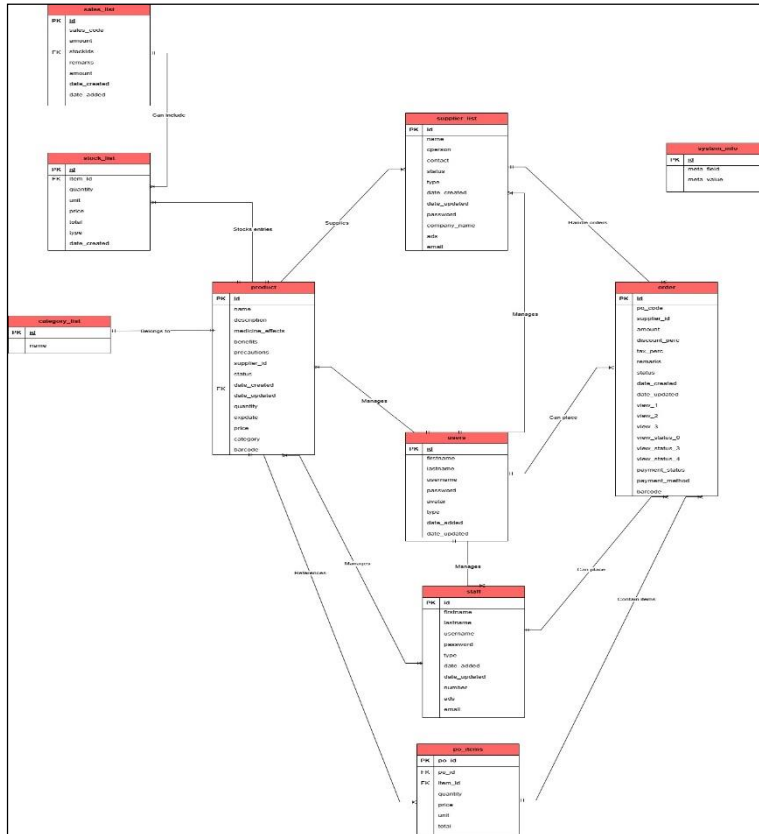


Fig. 6 ERD Diagram

4.3 Interface Design

Interface design is the art of creating user-friendly and visually appealing digital platforms like software, websites, and apps. It focuses on crafting intuitive and aesthetically pleasing interfaces that ensure a seamless user experience. Here are some graphical user interfaces (GUIs) from the prototype of the MyIndah Pharmacy Inventory Management System.

Login Interfaces: Users log into the system by entering their username and password through the interface in Fig 8. If both are correct, Admins, Staff, or Suppliers are redirected to the dashboard. An error message appears if either credential is incorrect, prompting the user to try again.



Fig. 7 Login interface

Manage Stock Interfaces: Figure 6(a) is an interface where the manager can add stock items within the system by inputting item details. Figure 6(b) displays list of stock and at the same time manager can update stock items.

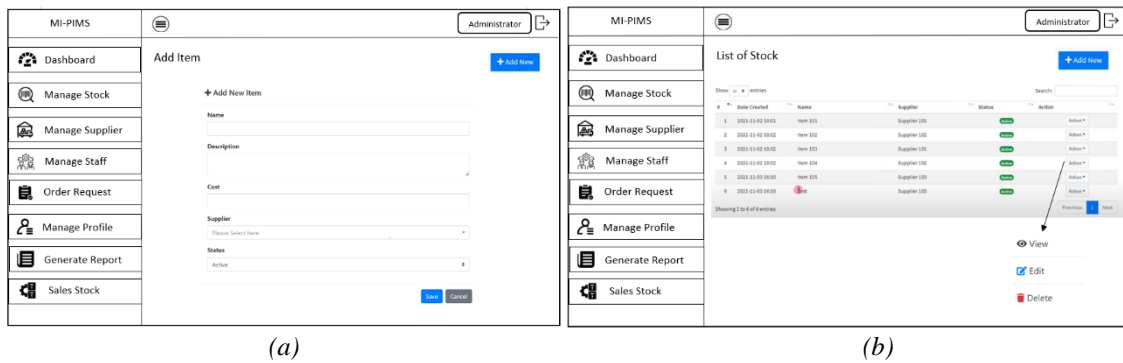


Fig. 8 (a) Add stock item; (b) List of stock interface

Manage Supplier Interfaces: The figures below show the interface where managers can create new suppliers by enter new supplier information such as name, contact information, address.

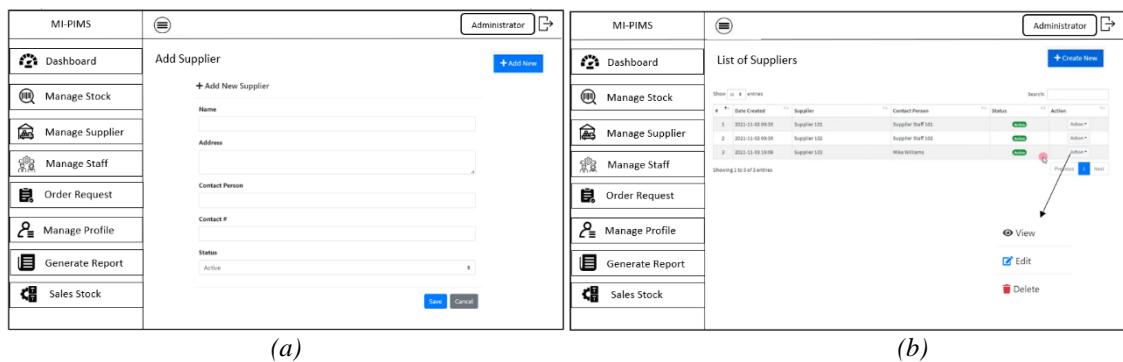


Fig. 9 (a) Add supplier interface; (b) List of supplier interface

5. Result and Discussion

The Results and Discussion section presents the findings of the study or project, interpreting them within the context of the research questions or objectives. It discusses the significance of these results, and their implications for the field, and suggests potential avenues for further research or application.

5.1 Implementation

The system uses HTML, CSS, JavaScript, Bootstrap, and jQuery for the front end, ensuring responsive and user-friendly interfaces. Bootstrap simplifies development with pre-built CSS classes. The back end is powered by PHP for server-side functionality, with CSV for Excel document generation and Apex Charts for interactive visualizations. This combination ensures efficient development and maintainable code.

5.1.1 User Login Interface

The login interface for manager, staff, and supplier is shown in Figure 9. Users need to enter a username and password into the input fields and select the roles. If the username, password and role are valid, the users will be allowed to log in to the system. If they are not valid, the system will display a message indicating that the username, password or role are incorrect.

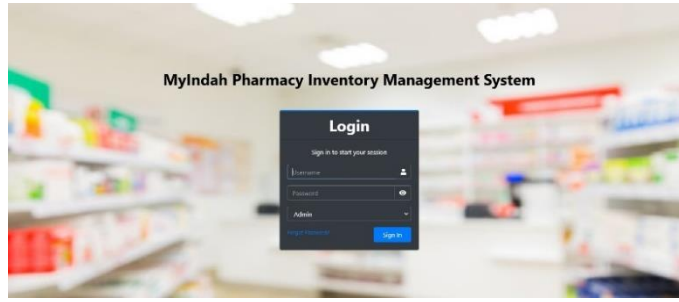


Fig. 10 Login interface

The PHP script in Figure 10(a) processes the login form submission by sanitizing and hashing the username, password, and role selections before comparing them to a MySQL database. If a match is found, a session is initiated; otherwise, an error message is displayed for incorrect credentials. Figure 10(b) shows the HTML code for the login form, which includes input fields for the username and password, a dropdown menu for role selection, a submit button labeled "Sign In," and a link for users who have forgotten their password.

```

<div class="input-group mb-3">
  <select name="login_type" class="form-control">
    <option value="1">Admin</option>
    <option value="2">Staff</option> <!-- Assuming 3 represents supplier login type -->
    <option value="3">Supplier</option> <!-- Assuming 3 represents supplier login type -->
  </select>
</div>
<div class="row">
  <div class="col-8">
    <a href="#" javascript:void(0)" class="forgot-password" onclick="flipForgotPassword()">Forgot Password</a>
  </div>
  <div class="col-4">
    <button type="submit" class="btn btn-primary btn-block">Sign In</button>
  </div>
</div>
</div>
// Check if form is submitted to post password
if ($SERVER['REQUEST_METHOD'] == 'POST' && isset($_POST['new_password']) && isset($_POST['username'])) {
  $username = $_POST['username'];
  $usertype = $_POST['user_type']; // 1 - Admin, 3 - Supplier
  $newpassword = $_POST['new_password'];

  // Validate if the password is not empty
  if (empty($username) || empty($newpassword)) {
    echo json_encode(['status' => 'error', 'message' => 'Username or new password cannot be empty']);
    exit;
  }

  // Check for Admin or Supplier
  if ($usertype == 1) {
    // For Admin users
    $sql = "SELECT * FROM users WHERE username = ?";
    $stmt = $conn->prepare($sql);
    $stmt->bind_param("s", $username);
    $stmt->execute();
    $result = $stmt->get_result();
  }
}
    
```

Fig. 11 (a)Front end for user login; (b)Back end for user login

5.1.2 Manage Stock Interface

The interface in Figure 12 shows a straightforward stock management page that presents a list of objects in tabular style. It contains critical information such as the Product ID, Product Name, Category, Quantity, and an Action column for tasks such as updating or deleting things. This user-friendly design ensures stock data is managed efficiently, giving users the tools to track, search, and alter inventory records.

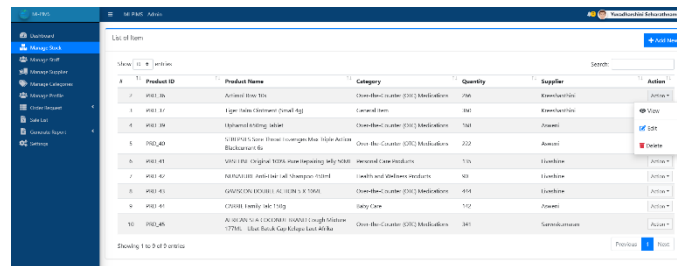


Fig. 12 Manage Stock interface

For the implementation in Figure 13(a), front-end code involves developing the user interface with HTML, CSS, and JavaScript. This includes building the table to display inventory items, providing a search box for filtering, and implementing pagination to move between pages. In Figure 13(b), back-end code uses PHP to perform server-side logic, such as obtaining, updating, deleting, and saving inventory data in a MySQL database, guaranteeing that the interface functions properly with real-time changes.

```

<div class="card-body">
  <div class="container-fluid">
    <div class="container-fluid">
      <table class="table table-bordered table-striped">
        <thead>
          <tr>
            <th>#</th>
            <th>Product ID</th>
            <th>Product Name</th>
            <th>Category</th>
            <th>Quantity</th>
            <th>Action</th>
          </tr>
        </thead>
        <tbody>
          <tr>
            <td>1</td>
            <td>ASPIRIN</td>
            <td>Aspirin 100mg Tablets</td>
            <td>Medicine</td>
            <td>100</td>
            <td>Update</td>
          </tr>
          <tr>
            <td>2</td>
            <td>PARACETAMOL</td>
            <td>Paracetamol 500mg Tablets</td>
            <td>Medicine</td>
            <td>200</td>
            <td>Update</td>
          </tr>
          <tr>
            <td>3</td>
            <td>IBUPROFEN</td>
            <td>Ibuprofen 400mg Tablets</td>
            <td>Medicine</td>
            <td>150</td>
            <td>Update</td>
          </tr>
          <tr>
            <td>4</td>
            <td>AMOXICILIN</td>
            <td>Amoxicillin 500mg Tablets</td>
            <td>Medicine</td>
            <td>120</td>
            <td>Update</td>
          </tr>
          <tr>
            <td>5</td>
            <td>CLOXACILIN</td>
            <td>Cloxacillin 500mg Tablets</td>
            <td>Medicine</td>
            <td>100</td>
            <td>Update</td>
          </tr>
          <tr>
            <td>6</td>
            <td>PENICILLIN</td>
            <td>Penicillin V 500mg Tablets</td>
            <td>Medicine</td>
            <td>80</td>
            <td>Update</td>
          </tr>
          <tr>
            <td>7</td>
            <td>ERYTHROMYCIN</td>
            <td>Erythromycin 250mg Tablets</td>
            <td>Medicine</td>
            <td>60</td>
            <td>Update</td>
          </tr>
          <tr>
            <td>8</td>
            <td>CLINDAMYCIN</td>
            <td>Clindamycin 300mg Tablets</td>
            <td>Medicine</td>
            <td>40</td>
            <td>Update</td>
          </tr>
          <tr>
            <td>9</td>
            <td>AZITHROMYCIN</td>
            <td>Azithromycin 500mg Tablets</td>
            <td>Medicine</td>
            <td>30</td>
            <td>Update</td>
          </tr>
          <tr>
            <td>10</td>
            <td>DILTIAZEM</td>
            <td>Diltiazem 120mg Tablets</td>
            <td>Medicine</td>
            <td>20</td>
            <td>Update</td>
          </tr>
        </tbody>
      </table>
    </div>
  </div>
</div>
</div>
<div class="card-body">
  <div class="container-fluid">
    <div class="container-fluid">
      <table class="table table-bordered table-striped">
        <thead>
          <tr>
            <th>#</th>
            <th>Product ID</th>
            <th>Product Name</th>
            <th>Category</th>
            <th>Quantity</th>
            <th>Action</th>
          </tr>
        </thead>
        <tbody>
          <tr>
            <td>1</td>
            <td>ASPIRIN</td>
            <td>Aspirin 100mg Tablets</td>
            <td>Medicine</td>
            <td>100</td>
            <td>Update</td>
          </tr>
          <tr>
            <td>2</td>
            <td>PARACETAMOL</td>
            <td>Paracetamol 500mg Tablets</td>
            <td>Medicine</td>
            <td>200</td>
            <td>Update</td>
          </tr>
          <tr>
            <td>3</td>
            <td>IBUPROFEN</td>
            <td>Ibuprofen 400mg Tablets</td>
            <td>Medicine</td>
            <td>150</td>
            <td>Update</td>
          </tr>
          <tr>
            <td>4</td>
            <td>AMOXICILIN</td>
            <td>Amoxicillin 500mg Tablets</td>
            <td>Medicine</td>
            <td>120</td>
            <td>Update</td>
          </tr>
          <tr>
            <td>5</td>
            <td>CLOXACILIN</td>
            <td>Cloxacillin 500mg Tablets</td>
            <td>Medicine</td>
            <td>100</td>
            <td>Update</td>
          </tr>
          <tr>
            <td>6</td>
            <td>PENICILLIN</td>
            <td>Penicillin V 500mg Tablets</td>
            <td>Medicine</td>
            <td>80</td>
            <td>Update</td>
          </tr>
          <tr>
            <td>7</td>
            <td>ERYTHROMYCIN</td>
            <td>Erythromycin 250mg Tablets</td>
            <td>Medicine</td>
            <td>60</td>
            <td>Update</td>
          </tr>
          <tr>
            <td>8</td>
            <td>CLINDAMYCIN</td>
            <td>Clindamycin 300mg Tablets</td>
            <td>Medicine</td>
            <td>40</td>
            <td>Update</td>
          </tr>
          <tr>
            <td>9</td>
            <td>AZITHROMYCIN</td>
            <td>Azithromycin 500mg Tablets</td>
            <td>Medicine</td>
            <td>30</td>
            <td>Update</td>
          </tr>
          <tr>
            <td>10</td>
            <td>DILTIAZEM</td>
            <td>Diltiazem 120mg Tablets</td>
            <td>Medicine</td>
            <td>20</td>
            <td>Update</td>
          </tr>
        </tbody>
      </table>
    </div>
  </div>
</div>
</div>
    
```

Fig. 13 (a)Front end for manage stock; (b)Back end for manage stock

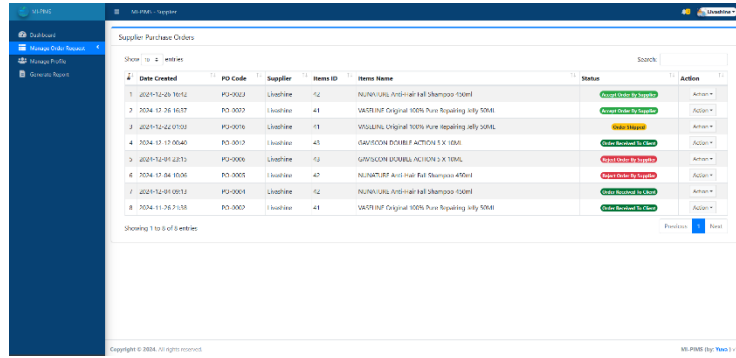


Fig. 16 Manage Purchase Order interface

Figure 17(a) shows the front-end code for purchase orders provides a visual interface for suppliers to interact with, presenting order requests and allowing them to accept, reject, or dispatch orders. It also let's check order history and status, and it uses HTML, CSS, and JavaScript to manage tables and communicate with users. Figure 17(b) shows the purchase order back-end code processes order data using server-side languages such as PHP, updating the database when orders are accepted, denied, or sent, and keeping track of order history.

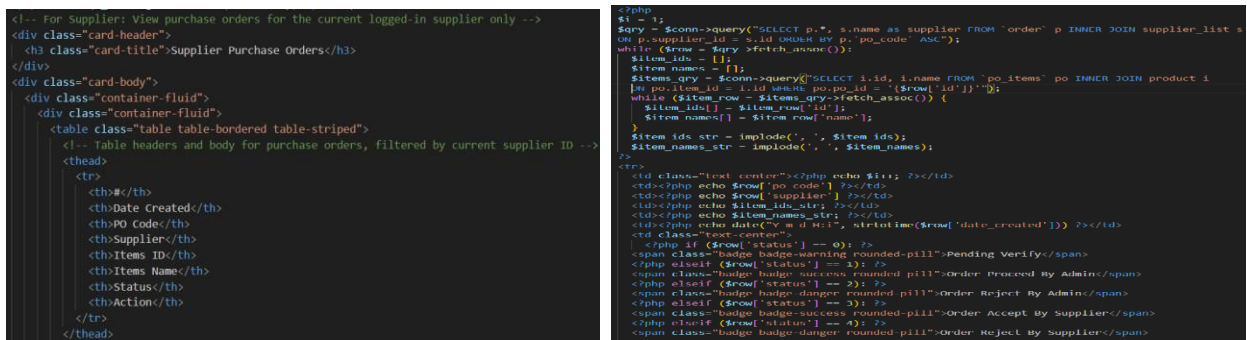


Fig. 17 (a)Front end for manage purchase order; (b)Back end for manage purchase order

5.1.5 Payment Interface

Figure 18 shows the payment options available once a provider accepts an order. It offers a list of pending payments and allows the manager to pay with cash or a card. When the payment is complete, the manager can obtain the receipt in Excel format.

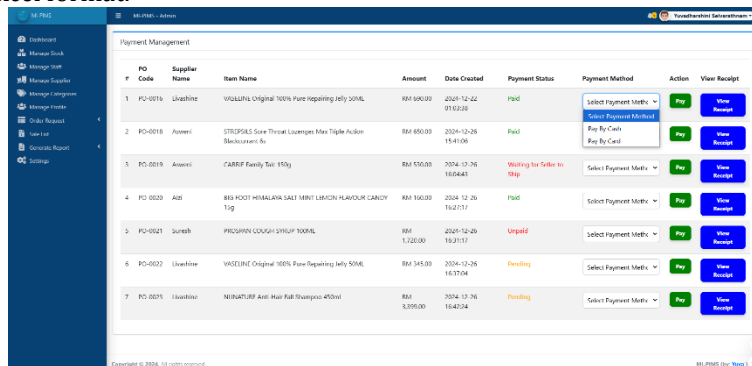


Fig. 18 Payment interface

The front-end code for payment in Figure 19(a) generates a user interface for managing payments, displaying payment options (cash or card), and including a button to download the receipt. The back-end code for payment in Figure 19 (b) processes the payment, updates the database, and generates the receipt through Excel.

5.2 Test Case

The testing phase is crucial for evaluating a system's performance and quality. It includes functional and user acceptance testing [10] to ensure the system meets user needs while improving reliability and usability (Jindal, 2016). Functional testing verifies that each module and function operates as expected based on functional requirements [9]. It identifies issues like incorrect calculations, missing features, or inconsistent user interfaces using test cases. Table 6 outlines the test cases for all modules.

Table 6 List of Test Cases

Test Cases	Description	Test Result (Pass/Fail)
TEST_100		
TEST_100_001	Verify that the manager can log in successfully.	PASS
TEST_100_002	Verify that staff can log in successfully.	PASS
TEST_100_003	Verify that the supplier can log in successfully.	PASS
TEST_100_004	Verify that the system displays an error message for a non-existent account.	PASS
TEST_100_005	Verify that the system displays an error message for an incorrect password.	PASS
TEST_100_006	Verify that the system displays a pop-up message in case of a system error during login	PASS
TEST_100_007	Verify that the login page is accessible.	PASS
TEST_200_001	Verify that a new item can be added with all required details	PASS
TEST_200_002	Verify that mandatory fields are validated when adding new item information.	PASS
TEST_200		
TEST_200_003	Verify that the information on existing items can be updated.	PASS
TEST_200_004	Verify that existing item information can be deleted.	PASS
TEST_200_005	Verify that item information can be searched	PASS
TEST_200_006	Verify the system behavior when adding new item information fails.	PASS
TEST_200_007	Verify the system behavior when updating item information fails	PASS
TEST_200_008	Verify the system behavior when deleting item information fails.	PASS
TEST_200_009	Verify the system behavior when no matching item information is found during a search.	PASS
TEST_300		
TEST_300_001	Verify that manager can navigate to the Manage Supplier module.	PASS
TEST_300_002	Verify that the manager can view a list of current suppliers.	PASS
TEST_300_003	Verify that a new supplier can be added with all required details.	PASS
TEST_300_004	Verify that mandatory fields are validated when adding new supplier information.	PASS
TEST_300_005	Verify that existing supplier information can be updated	PASS
TEST_300_006	Verify that existing supplier information can be deleted.	PASS
TEST_300_007	Verify that supplier information can be searched	PASS
TEST_300_008	Verify the system behavior when adding new supplier using same email	PASS
TEST_300_009	Verify the system behavior when adding new supplier information fails	PASS
TEST_300_010	Verify the system behavior when updating supplier information fails.	PASS
TEST_300_011	Verify the system behavior when deleting supplier list fails.	PASS
TEST_300_012	Verify the system behavior when no matching supplier information is found during a search.	PASS

Table 6: Continued

TEST_400		
TEST_400_001	Verify that manager can navigate to the Manage Staff module.	PASS
TEST_400_002	Verify that the manager can view a list of current staff.	PASS
TEST_400_003	Verify that a new staff can be added with all required details.	PASS
TEST_400_004	Verify that mandatory fields are validated when adding new staff information.	PASS
TEST_400_005	Verify that existing staff information can be updated	PASS
TEST_400_005	Verify that existing staff information can be updated	PASS
TEST_400_006	Verify that existing staff information can be deleted.	PASS
TEST_400_007	Verify that staff information can be searched	PASS
TEST_400_008	Verify the system behavior when adding new staff using same email	PASS
TEST_400_009	Verify the system behavior when adding new staff information fails	PASS
TEST_400_010	Verify the system behavior when updating staff information fails.	PASS
TEST_400_011	Verify the system behavior when deleting staff list fails.	PASS
TEST_400_012	Verify the system behavior when no matching staff information is found during a search.	PASS
TEST_500		
TEST_500_001	Verify that the manager, staff, and supplier can navigate to the manage profile	PASS
TEST_500_002	Verify that the manager, staff, and supplier can modify profile details.	PASS
TEST_500_003	Verify that the manager can add a new admin and manage the admin profile	PASS
TEST_500_004	Verify the system behavior when adding new admin information fails.	PASS
TEST_500_005	Verify the system behavior when updating profile information fails.	PASS
TEST_500_006	Verify that only the authorized manager, staff, or supplier can access the managed profile	PASS
TEST_600		
TEST_600_001	Verify that the manager and staff can navigate to the order request module	PASS
TEST_600_002	Verify that the manager and staff can view a list of current order request	PASS
TEST_600_003	Verify that the staff can make an order request	PASS
TEST_600_004	Verify that the manager receives alerts about pending orders	PASS
TEST_600_005	Verify that the manager can accept or reject the order.	PASS
TEST_600_006	Verify that the staff are notified when the order is accepted or rejected	PASS
TEST_600_007	Verify that the manager and staff can modify order request details.	PASS
TEST_600_008	Verify that the manager and staff can delete an order request.	PASS
TEST_600_009	Verify that order request information can be search	PASS
TEST_600_010	Verify the system behavior when updating order request information fails.	PASS
TEST_700		
TEST_700_001	Verify that the supplier can navigate to the manage purchase order module	PASS
TEST_700_002	Verify that the supplier can view a list of current orders	PASS
TEST_700_003	Verify that the supplier receives alerts about pending orders	PASS
TEST_700_004	Verify that the supplier can accept or reject the order.	PASS
TEST_700_005	Verify that the manager and staff are notified when the order is accepted or rejected	PASS
TEST_700_006	Verify the system behavior when updating order status	PASS

Table 6: Continued

TEST_800		
TEST_800_001	Verify that the manager can access the payment section	PASS
TEST_800_002	Verify that the manager can select the cash payment method	PASS
TEST_800_003	Verify that the manager can select the card payment method.	PASS
TEST_800_004	Verify that the system allows the manager to input valid card details	PASS
TEST_800_005	Verify that the system rejects invalid card details.	PASS
TEST_800_006	Verify the system behavior when the payment by card transaction fails	PASS
TEST_800_007	Verify that the system provides a payment success message after card payment.	PASS
TEST_900		
TEST_900_001	Verify that the system displays the sales list page correctly.	PASS
TEST_900_002	Verify that the system shows all sales records accurately.	PASS
TEST_900_003	Verify that the supplier can search for a specific sales record.	PASS
TEST_900_004	Verify the system behavior when there are no sales records available.	PASS
TEST_900_005	Verify that clicking on a sales record shows detailed information about it.	PASS
TEST_1000		
TEST_1000_001	Verify the manager, staff or supplier can access the report generation module.	PASS
TEST_1000_002	Verify the system handles errors gracefully during report generation.	PASS
TEST_1000_003	Verify the manager, staff or supplier can use various filtering options to customize the report.	PASS
TEST_1100		
TEST_1100_001	Verify that the system sends a notification when a new order is placed.	PASS
TEST_1100_002	Verify that the system sends a notification when an order is accepted or rejected by the manager or supplier.	PASS
TEST_1100_003	Verify that the system sends an email containing the supplier's and staff's username and password upon account creation.	PASS

5.3 User Acceptance Testing

Figure 22(a) illustrates the login feedback, providing a fast and seamless experience with easy login, error alerts for incorrect credentials, and a simple password reset process. While the page is quick and accessible, two evaluators rated its usability and performance as "Agree." Similarly, Figure 22(b) shows the manage stock feature, allowing managers to efficiently add, update, and search for items, boosting productivity. Most users rated its performance as "Strongly Agree," though a few marked it as "Agree" for ease of use and efficiency. Sample evaluation questions are provided in Appendix C.1(a) and C.1(b).

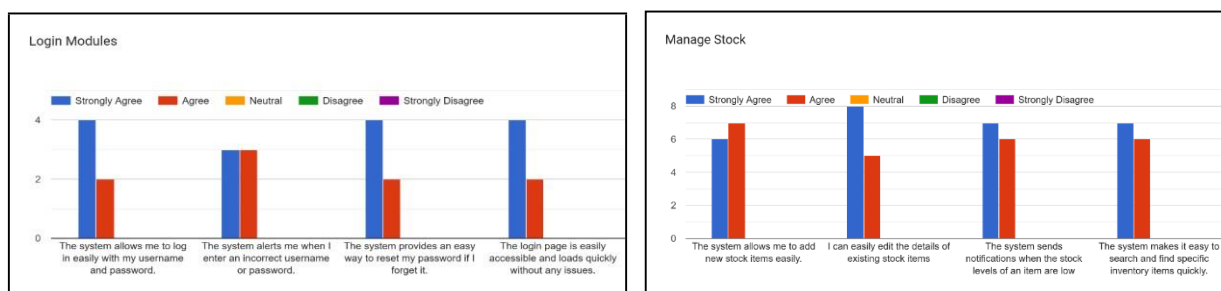


Fig. 22 (a)Feedback for Login module; (b)Feedback for Manage Stock module

The manage staff feedback in Figure 23(a) and manage supplier feedback in Figure 23(b) highlight efficient ways to organize information. Users can sort, edit, and access key details with ease. Both modules include a confirmation prompt to prevent accidental deletions and receive positive feedback, with users rating them "Strongly Agree" for ease of use and functionality. APPENDIX C.2(a) and APPENDIX C.2(b) show the sample questions used to evaluate system performance and usability.

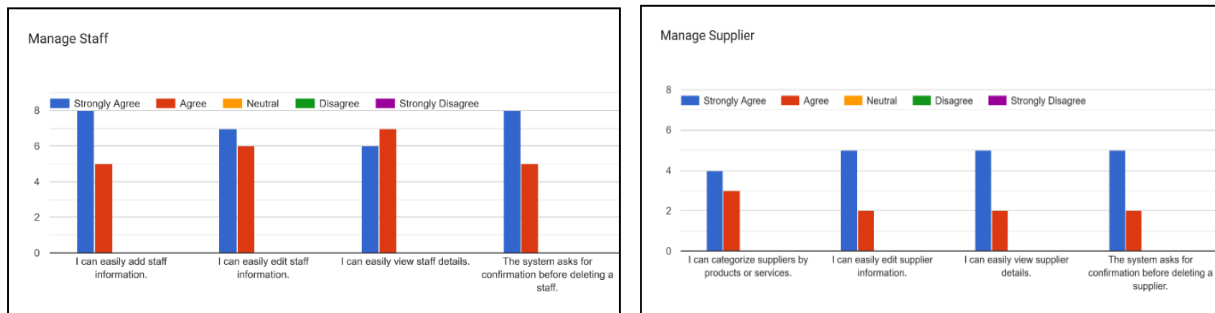


Fig. 23 (a)Feedback for Manage Supplier module; (b)Feedback for Manage Profile module

The order request feedback in Figure 24(a) streamlines order placement for stock restocking, providing confirmation and status updates. It is user-friendly and received positive feedback for efficiency. Similarly, the manage purchase order feedback in Figure 24(b) ensures transparency by maintaining complete purchase order records, including statuses. Users praised its simplicity and functionality, with most rating it "Strongly Agree" and some rating it "Agree." APPENDIX C.3(a) and APPENDIX C.3(b) show the sample questions used to evaluate system performance and usability.



Fig. 24 (a)Feedback for Order Request module; (b)Feedback for Manage Purchase Order module

6. Conclusion

The MyIndah Pharmacy Inventory Management System (MI-PIMS) successfully met its overall objectives. Firstly, a structured design approach was used to ensure clarity, efficiency, and consistency in the system's architecture. Secondly, both front-end and back-end technologies were employed in the development of the web application, resulting in a responsive and user-friendly interface. Lastly, the system completed user acceptance testing to confirm it meets user requirements and performs reliably under various conditions. These accomplishments emphasize the system's reliability, usability, and adherence to its design methodology.

MI-PIMS enhances efficiency by automating tasks like inventory tracking and data management, reducing manual errors, and supporting better decision-making. The system provides an easy-to-use platform for pharmacy staff, with features like real-time tracking and detailed reports to help with decision-making. However, it has some limitations, including occasional inaccuracies in tracking, potential security risks if users do not log out, and vulnerabilities to cyber-attacks. Enhancements such as stronger security features and more accurate tracking could further improve its reliability and performance.

In summary, MI-PIMS demonstrates the project's success in automating processes and centralizing inventory data to improve management efficiency and user experience. It addresses challenges, particularly overcoming the inefficiencies of manual systems, and highlights strengths in improving operational efficiency. The results of user acceptance testing offer valuable insights for future system improvements.

Acknowledgment

The authors express their gratitude to the Faculty of Computer Science and Information Technology at Universiti Tun Hussein Onn Malaysia for their invaluable assistance and motivation during the study's conduct.

Conflict of Interest

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

Author Contribution

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

The authors confirm contribution to the paper as follows: **study conception and design:** Yuvadharshini A/P Selvarathnam, Mohd Amin Bin Mohd Yunus; **data collection:** Yuvadharshini A/P Selvarathnam, Mohd Amin Bin Mohd Yunus; **analysis and interpretation of results:** Yuvadharshini A/P Selvarathnam, Mohd Amin Bin Mohd Yunus; **draft manuscript preparation:** Yuvadharshini A/P Selvarathnam, Mohd Amin Bin Mohd Yunus. All authors reviewed the results and approved the final version of the manuscript.

References

- [1] Zoho Corporation Pvt Ltd. (n.d.). Online inventory management software | Zoho Inventory. <https://www.zoho.com/inventory/>
- [2] Remedi. (n.d.). Remedi - Clinic Management System Malaysia. <https://remedi.my/>
- [3] The, "Ordoro," *Ordoro.com*, 2020. <https://www.ordoro.com/tradegecko-inventory>
- [4] Hooray, L. (2024, May 29). *What is waterfall methodology? Here's how it can help your project management strategy.* Forbes Advisor. <https://www.forbes.com/advisor/business/what-is-waterfall-methodology/>
- [5] Fergusson, K. (2023, August 18). *Entity Relationship Diagrams with draw.io.* draw.io. <https://drawio-app.com/blog/entity-relationship-diagrams-with-draw-io/>
- [6] Visual Studio Code - Code editing. Redefined. (2021, November 3). <https://code.visualstudio.com/>
- [7] Zoho | Cloud Software Suite for businesses. (n.d.). Zoho. <https://www.zoho.com/>
- [8] GeeksforGeeks. (2024, October 4). *What is DFD (Data Flow Diagram)?* GeeksforGeeks. <https://www.geeksforgeeks.org/what-is-dfd-data-flow-diagram/>
- [9] "What Is Test Case? How To Write Test Cases In Software Testing?," *Testsigma Blog*, Jun. 28, 2023. <https://testsigma.com/guides/what-is-test-case/>
- [10] A. Gillis, "What is User Acceptance Testing (UAT)?," *SearchSoftwareQuality*, Mar. 2022. <https://www.techtarget.com/searchsoftwarequality/definition/user-acceptance-testing-UAT>

Appendix A

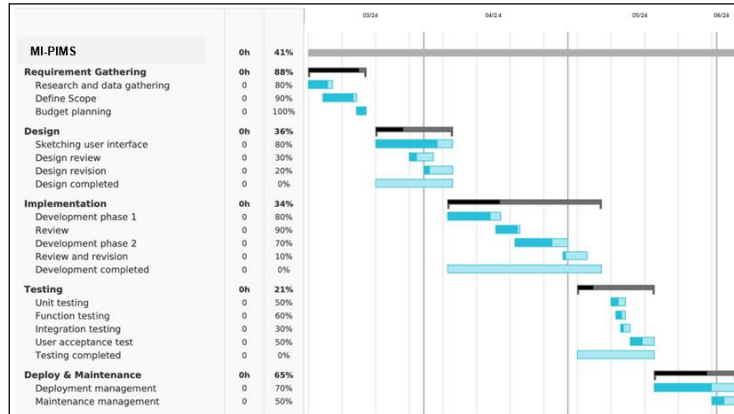


Fig A.1 Gantt Chart

Appendix B

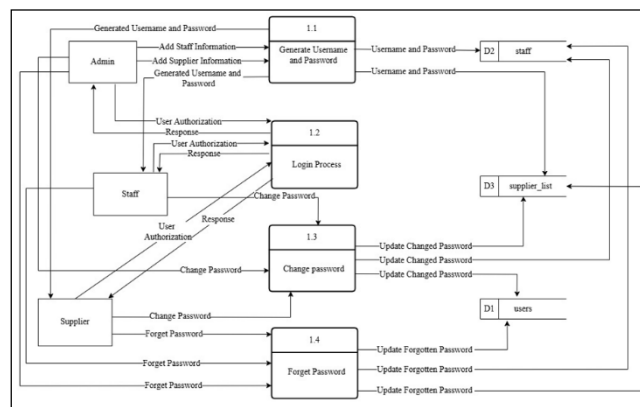


Fig B.1 Data Flow Diagram Level 1 Process 1.0 Login

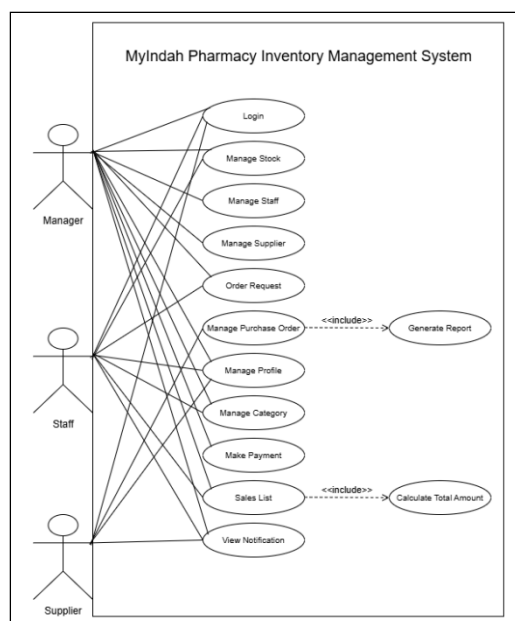


Fig B.2 Use Case Diagram