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

AITCS

Vol. 6 No. 1 (2025) 343-361
https://publisher.uthm.edu.my/periodicals/index.php/aitcs

Inventory Management System with Two-Factor Authentication for Heng Huat Motor & Electrical Trading

Chia Jing Yan[1], Nur Ziadah Harun[1]*

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

*Corresponding Author: nurziadah@uthm.edu.my
DOI: https://doi.org/10.30880/aitcs.2025.06.01.019

Article Info

Received: 26 July 2024
Accepted: 18 June 2025
Available online: 30 June 2025

Keywords

Inventory Management System, 2FA, Time-Based One-Time Password, Web-Based System

Abstract

Inventory management system for Heng Huat Motor & Electrical Trading is a system designed to modernize and enhance its inventory control process. Since the current manual system lacks of inadequate security measures for safeguarding inventory data, it potentially leads to data loss and unauthorized access. To secure the system, role-based access control and two-factor authentication are implemented to restrict unauthorized access. Strong passwords and time-based one-time passwords add an extra layer of security. Each user has different access based on their role within the company to safeguard the sensitive information. With limited access control, the data are easily susceptible to potential breaches of confidentiality. An agile model is used to develop the system since it is flexible and adaptable. This web-based system is mainly written in PHP programming language and all data are stored in MySQL. This system simplifies inventory management by providing a comprehensive view of the product inventory and timely stock alerts. The product restocking and checkout process is enhanced by the use of barcode scanning to ensure accuracy in inventory level calculations. Barcodes are generated for each product and barcode scanners are used to simplify the data entry process. The system integrates with Google Authenticator app for two-factor authentication. During the first login, users need to scan a QR code to establish a connection between the Google Authenticator app and their user account so that users can input a one-time password provided by Google Authenticator app during each login. In short, this inventory management system with two-factor authentication helps Heng Huat to manage its inventory effectively.

1. Introduction

Effective inventory management involves efficiently monitoring stock levels and analyzing sales data. An in-store inventory management system eliminates the need for manual recording of every sale and new stock by automatically recording these transactions in the system [1]. Heng Huat Motor & Electrical Trading currently relies on a traditional manual inventory management method. With the current system, the inventory records mainly depend on memory. Any products that run low are manually recorded in a logbook for the purpose of restocking. Besides, all invoice statements received from suppliers are collected and stored by the owner in a separate file for record-keeping purposes. However, it is challenging for owners to organize the inventory efficiently in the existing system without a proper record-keeping system. Thus, a risk of forgetting certain

products and thus potentially leading to inventory obsolescence. The forgotten products may become less valuable after a long time as technology advances. At the same time, it also increases the storage costs. The current manual system also raises a significant risk of potential data loss since it is susceptible to human errors. An unintentional error may result in inaccurate data. Besides, the inadequate security measures for safeguarding inventory data makes inventory data vulnerable to unauthorized access. This may raise concerns about data breach.

The objective of this project is to design, develop and test a web-based inventory management system with Time-Based One-Time Password (TOTP) authentication for Heng Huat Motor & Electrical Trading. The system will be implemented with security elements including role-based access control, password hashing and two factor authentication for secure inventory management. In short, the proposed inventory management system provides Heng Huat an efficient control over its inventory and streamlines the inventory management process.

2. Literature Review

The literature review is presented by discussing various subjects relevant to the project including inventory management system, two-factor authentication, time-based one-time password, Google Authenticator and role-based access control. Three similar existing systems will be compared with the proposed system.

2.1 Inventory Management System

Inventory refers to the stock of goods or resources used in a business [2]. An inventory management system is a structured way to control an organization's stock of goods. Inventory modelling plays an important role in inventory management systems to derive an operating doctrine. The methodology for modelling inventory situations involves three simple steps [3]. Firstly, analyze the inventory situation carefully, documenting characteristics and underlying assumptions. Then, create the equation for the total annual relevant cost and lastly optimize the cost equation to determine the optimal order quantity and reorder point. It provides real-time tracking of inventory levels to support accurate demand forecasting. By monitoring the movement of all stocks in an organized manner, sales and remaining stock reports can be generated quickly and accurately. It enhances decision-making by providing insights into product movement. The automated system also decreases manual errors and streamlines the inventory management process.

2.2 Two-Factor Authentication

Two-Factor Authentication (2FA) is a security method that requires two different ways of identification to gain access to something [4]. 2FA adds an extra layer of security to the traditional SFA by using two different types of credentials. If one factor of the identification is compromised, the second factor can act as a backup to prevent unauthorized access of attackers. The second factor of authentication can be simply divided into three categories which are "something you know", "something you have" and "something you are" [5]. "Something you are" typically refers to password, PIN or security question which is usually used in SFA as the basic authentication method. The example of "Something you have" can be a one-time password sent to phone, hardware token or smart card. "Something you are" is considered as the most secure category since it involves biometric traits such as fingerprints, facial features, voices and retinas. This category provides extra safety since each person's physical characteristics are unique. In the proposed system, TOTP with Google Authenticator app is chosen for the second factor in 2FA. It provides a good balance between security and user convenience as users do not need specialized hardware or complex setup. However, users still risk to lose their accounts if their devices with authenticator are lost.

2.3 Time-Based One-Time Password

Time-Based One-Time Password (TOTP) is a time-based algorithm that employs the counter as a moving factor which was proposed by OATH members in 2008 and was formally accepted in 2011 [6]. The TOTP algorithm is a variation of the HMAC-based One-time Password (HOTP) algorithm that is computed using the common function but works based on time [7]. HOTP is an algorithm for generating OTP using the hash-based message authentication code. It relies on hash functions that convert message from an arbitrarily length into a sequence of bits with specific length. Since they are one-way functions, the hash value is calculated from the input and the reverse operation is prohibited.

Before creating a TOTP, both the client and the certifier need to synchronize their clocks by ensuring they have access to the current UNIX time [7]. They share a key or have information to create a key during the initial set up process. The shared key and the current time are used in TOTP algorithm to generate a unique TOTP code. Since the code is created based on the current time and the shared secret key, it becomes unique and can only be used once. These characteristics makes it more secure compared to traditional static passwords. TOTP codes are generated dynamically and remain valid only for a specific period. After this period, a new code will be generated

immediately, and the cycle will be repeated. The delay issue in OTP delivery is solved by generating TOTP codes on the client side. Authenticator app is used as the medium to deliver codes to users. TOTP creates a unique code which is synchronized between user's device and the system based on the current time. The TOTP mechanism working with authenticator app can efficiently verify that the user is associated with the phone. The continually changing security code which only works for a short time makes it challenging for attackers to sneak in and thus efficiently enhances resistance against unauthorized access.

2.4 Google Authenticator

Google Authenticator is a free mobile application specially used to provide security protection to users' accounts against attacks [8]. It is an authenticator app which is developed by Google to generate TOTP using a specific algorithm. It is primarily designed for Google accounts and services but also available for other platforms that support TOTP authentication. It operates offline for generating codes without requiring an internet connection. This increases reliability especially when having limited connectivity. However, users potentially lose access to the accounts linked to the app if without a proper backup. It is widely compatible with platforms and services that support TOTP for two-factor authentication. In short, Google authenticator focuses primarily on TOTP-based authentication.

2.5 Role-Based Access Control

Role-Based Access Control (RBAC) is a robust access control policy that regulates user permissions within an organization based on different roles. RBAC policy decides what actions can be taken by users in an organization by assigning specific functions associated with their roles [9]. This enables users to successfully access the data needed for their job duties while minimizing the risk of unauthorized access to sensitive information or engaging in unlawful operations [10]. In RBAC, different roles are defined, and each role represents a special set of responsibilities and functions. RBAC system will only allow access if the user's role has the required permissions for the requested action. Otherwise, access will be denied. RBAC provides a scalable approach to access control by allowing the modification of roles and the access rights as an organization evolves. It efficiently enhances security by ensuring that users only have the permission necessary for their roles so that any unauthorized access to sensitive information can be prevented.

2.6 Study Of Existing System

In this section, three similar existing systems will be discussed respectively. The chosen existing inventory systems are Zoho Inventory, Odoo Inventory and Quickbooks Inventory.

2.6.1 Zoho

Zoho Inventory is a cloud-based solution which is designed to streamline the inventory management for different sizes of business. It is developed by Zoho Corperation [11]. The platform provides an efficient oversight of the stock, orders and supply chain processes as shown in Fig. 1. The data is encrypted using the 256-bit Advanced Encryption Standard (AES). Zoho's security framework is built based on OWASP standards. It distributes and maintains cloud space for organizations to ensure logical separation of each organization's service data and thus prevent any unauthorized access. The implementation of single sign-on (SSO) in Zoho streamlines the login process, provides robust access control and reduces the risk of password fatigue. Multi-factor authentication (MFA) is also offered during login sessions to restrict unauthorized access.

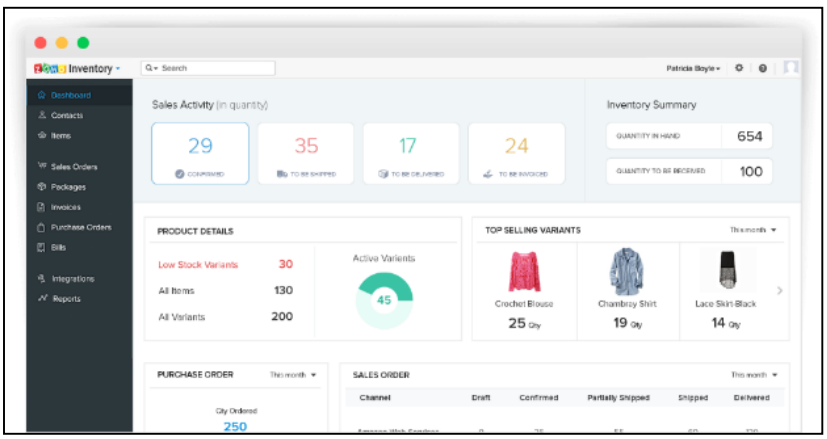


Fig. 1 *The dashboard interface of Zoho Inventory*

2.6.2 Odoo

Odoo is an open-source business management software tool that integrates and centralizes the structure and details of work [12]. It includes CRM, e-commerce, accounting, billing, manufacturing, warehouse, project management and inventory management. The data is stored in dedicated databases to ensure complete isolation between organizations. All login credentials are always transmitted securely via HTTPS. Strong password policies are also implemented in Odoo. State-of-the-art 256-bit SSL encryption is used to secure all communications to client instances. SSH encryption is used to secure all internal data communications between servers. Odoo encrypts all customer data including data in production and backups by using AES-128 or AES-256 encryption. This comprehensive approach makes Odoo with top security standards listed by the OWASP. Fig. 2 shows the dashboard interface of Odoo Inventory.

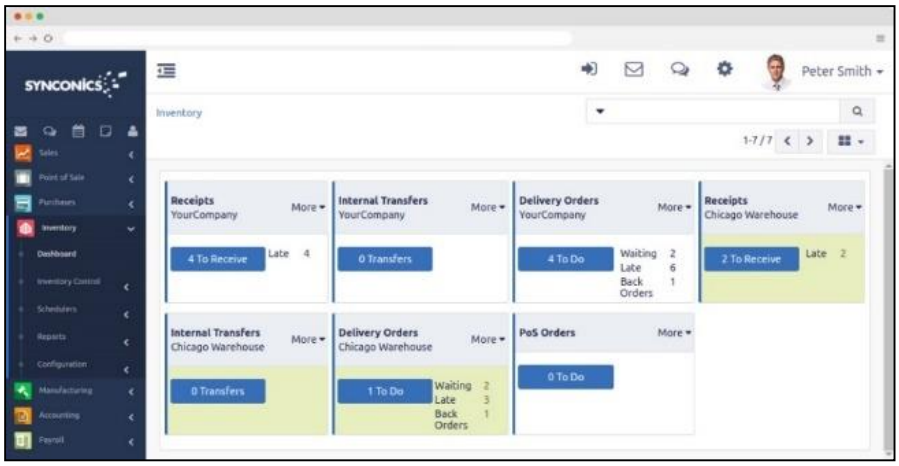


Fig. 2 *The dashboard interface of Odoo Inventory*

2.6.3 Quickbooks

QuickBooks is an accounting software package that extends its functionality to include inventory management services [13]. It is developed and marketed by Intuit. QuickBooks offers password-protected logins, firewall-protected servers and 128-bit SSL encryption to increase the security of data. Automatic back-ups are automatically done in QuickBooks so that data can be accessed from any computer which is connected to the internet if any unexpected accidents happen. QuickBooks relies on redundant servers and a self-correcting error detection program to ensure the service can remain unaffected if any server is unavailable. Fig. 3 shows the product and services interface of QuickBooks Inventory.

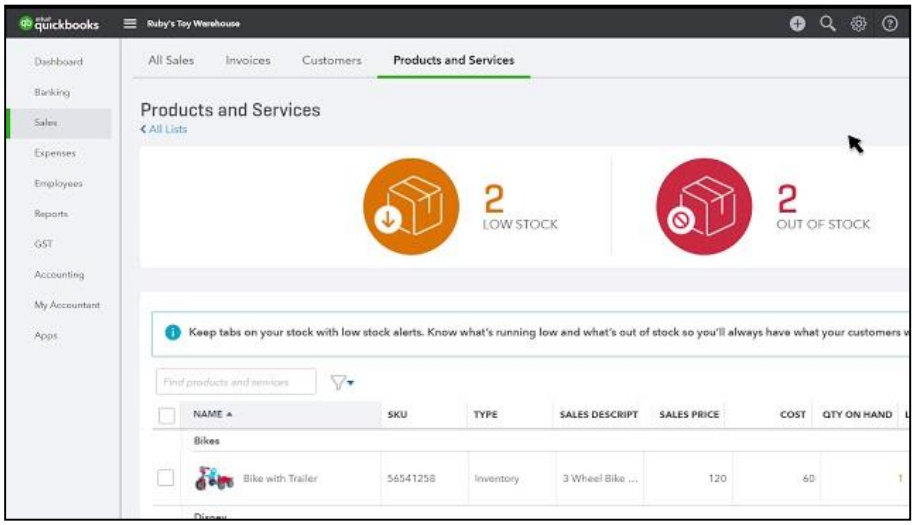


Fig. 3 *The products and services interface of QuickBooks Inventory*

Applied Information Technology and Computer Science Vol. 6 No. 1 (2025) p. 343-361

2.6.4 Comparison Between Existing System With Proposed System

Table 1 *Comparison between Existing System and Proposed System*

Features	Zoho	Odoo	QuickBooks	Proposed System
2FA (TOTP with Google Authenticator app)	Yes	Yes	No	Yes
Strong Password	Yes	Yes	Yes	Yes
RBAC	No	Yes	No	Yes
Data encryption	Yes	Yes	Yes	Yes
Single Sign-On	Yes	Yes	Yes	Yes
Register and Login	Yes	Yes	Yes	Yes
Product Management	Yes	Yes	Yes	Yes
Auto Reordering	Yes	No	Yes	No
Sales Reporting	Yes	Yes	Yes	Yes
Product Checkout	Yes	Yes	Yes	Yes
Barcode Item Tracking	Yes	Yes	Yes	Yes
Stock Alert Reminder	Yes	Yes	Yes	Yes

The systems have similar essential features and characteristics which are crucial for an inventory management system as shown in Table 1. The proposed system combines the security characteristics of the existing systems discussed including two-factor authentication and RBAC policy to enhance security and access control. However, the proposed system does not offer the auto reorder functionality which presents in discussed existing systems. All the discussed existing systems and the proposed system implement a registration and login system to ensure privacy and access control exclusively for internal workers. In the proposed system, 2FA will be implemented with strong password as first authentication factor and TOTP with Google Authenticator as second authentication factor.

3. Methodology

Agile development model is an iterative and incremental development approach which allows flexible adjustments to requirements based on customer needs [14]. It emphasizes the orderly delivery of specific system components rather than the complete system. Agile is chosen as the methodology in this project to adapt to changing business requirements including variations in demand and the introduction of new products. Agile effectively responds to these changes with regular iterations. It also encourages ongoing collaboration between development teams and the customer throughout the development cycle for effective feedback gathering. There are six phases included in Agile as shown in Fig. 4 which are plan, design, develop, test, deploy, review and launch. Table 2 shows the system development workflow.

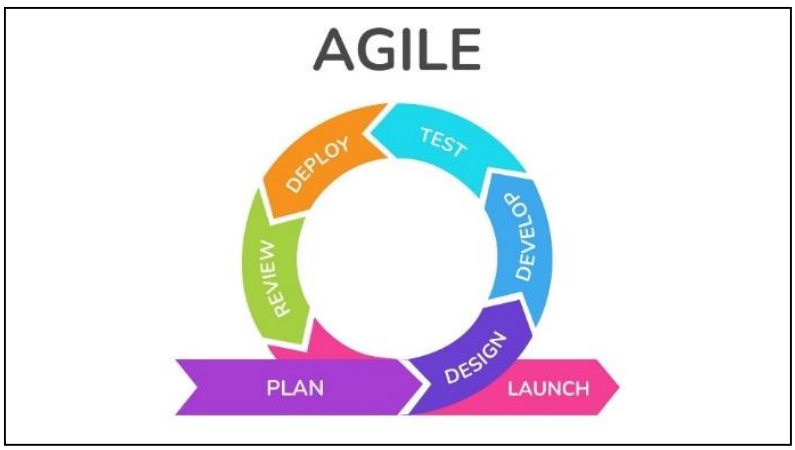


Fig. 4 *Agile Development Model*

Table 2 *System Development Workflow*

Phase	Task	Output
Plan	• Propose title of system • Determine problem statement, objective and scope • Interview with stakeholders • Identify users and system requirements • Construct work plan with Gantt chart • Review similar inventory management systems • Study relevant components to the system • List out software and hardware requirements	• Proposal • User and system requirements • Gantt chart • Literature review • Hardware and software requirements
Design	• Design interface of system by producing wireframes for inventory management system • Construct flowchart, entity relationship diagram	• Interface design • Backend design • Database design • Wireframe • Flowchart of system • Entity Relationship diagram
Develop	• Implement backend code using PHP programming language • Implement frontend code using HTML and CSS • Detect errors	• Complete developed system
Test	• Interface testing o Ensure that users can navigate through different sections of the system seamlessly. o Check the system's compatibility with various browsers and devices. o Ensure that system interfaces adapt to different screen sizes. • Functional testing o Ensure all functions run smoothly. o Check the interactions between modules. o Ensure the system meets end-users' requirements through User Acceptance Testing. • Security testing o Configuration and deployment management testing o Identity management testing o Authentication testing o Authorization testing o Session management testing o Input validation testing o Testing for error handling o Testing for weak cryptography o Business logic testing o Client-side testing	• Testing results • Final tested system with functionality test, security test and User Acceptance Test.
Deploy	• Deliver the system to customers • Installed the system in customers' devices	• Successful system release
Review	• Update the system regularly	• Users' feedback

• Add new features based on customers' requirements	• New feature requirements

4. System Analysis and Design

This section explains the system analysis and design of the proposed system. It covers requirement analysis, flowchart and sequence diagram.

4.1 System Requirements Analysis

Functional requirements are used to define the intended behavior of the proposed inventory management system including the services, tasks or functions that should be performed by the system [15]. It outlines the functions of each module in the system as a comprehensive guide in the development process. Table 3 shows the functional requirements of the proposed system.

Table 3 *Functional requirements of the proposed system*

Module	Functionality	User
Login	• The system should authenticate username and password for first layer authentication. • The system should validate user inputs to ensure they are in required format. • The system should prompt users to correct the input if invalid input is detected. • The system should authenticate TOTP for second layer authentication if username and password are valid. • The system should redirect users to the respective account after successful login.	• Administrator • Manager • Stork clerk • Cashier
User management	• The system should provide administrators with the ability to deactivate or create user accounts. • The system should provide administrators with the ability to assign roles for all users. • The system should allow users to modify account details.	• Administrator • Manager
Product management	• The system should allow users to add or delete products. • The system should allow users to modify the product details including product name, original price, supplier and quantity. • The system should allow administrators and managers to modify the product selling price. • The system should provide product search and filtering.	• Administrator • Manager • Stork clerk
Checkout	• The system should locate specific products through responding to barcode. • The system should accurately calculate the price of selected products. • The system should automatically delete the sold product from the inventory. • The system should save each transaction details as record.	• Administrator • Manager • Cashier
Reminder	• The system should send reminders to users if product quantity falls below the stock alert threshold.	• Administrator • Manager • Stork Clerk
Sales report	• The system should provide sales reports based on the transaction record.	• Administrator • Manager

Non-functional requirements are important to define the constraints that affect the overall behavior of the system. It ensures that the system can perform effectively while meeting the functional requirements. The non-functional requirements of the proposed inventory management system are discussed from four categories which

are operational, performance, security and usability. Table 4 shows the non-functional requirements of the proposed system.

Table 4 *Non-functional requirements of the proposed system*

Module	Functionality
Login	• The system should authenticate username and password for first layer authentication. • The system should validate user inputs to ensure they are in required format. • The system should prompt user to correct the input if invalid input is detected. • The system should authenticate TOTP for second layer authentication if username and password are valid. • The system should redirect users to the respective account after successful login.
User management	• The system should provide administrators with the ability to deactivate or create user accounts. • The system should provide administrators with the ability to assign roles for all users. • The system should allow users to modify account details.
Product management	• The system should allow users to add or delete products. • The system should allow users to modify the product details including product name, original price, supplier and quantity. • The system should allow administrators and managers to modify the product selling price. • The system should provide product search and filtering.
Checkout	• The system should locate specific products through responding barcode. • The system should accurately calculate the price of selected products. • The system should automatically delete the sold product from the inventory. • The system should save each transaction details as record.
Reminder	• The system should send reminders to users if product quantity falls below the stock alert threshold.
Sales report	• The system should provide sales reports based on the transaction record. • The system should provide inventory report.

4.2 Sequence Diagram

Sequence diagram is used to visualize the interactions and flow of messages between different objects within a system. Fig. 5 shows the sequence diagrams for administrator. Google authenticator is used to scan the QR code generated by system to get the shared secret key. The key is used for the TOTP generation. Once successful login system with correct username, password and valid TOTP, administrator is able to assign role, manage user, manage product, set stock alert reminder, check out and generate report. All accounts are created by the administrator.

Applied Information Technology and Computer Science Vol. 6 No. 1 (2025) p. 343-361

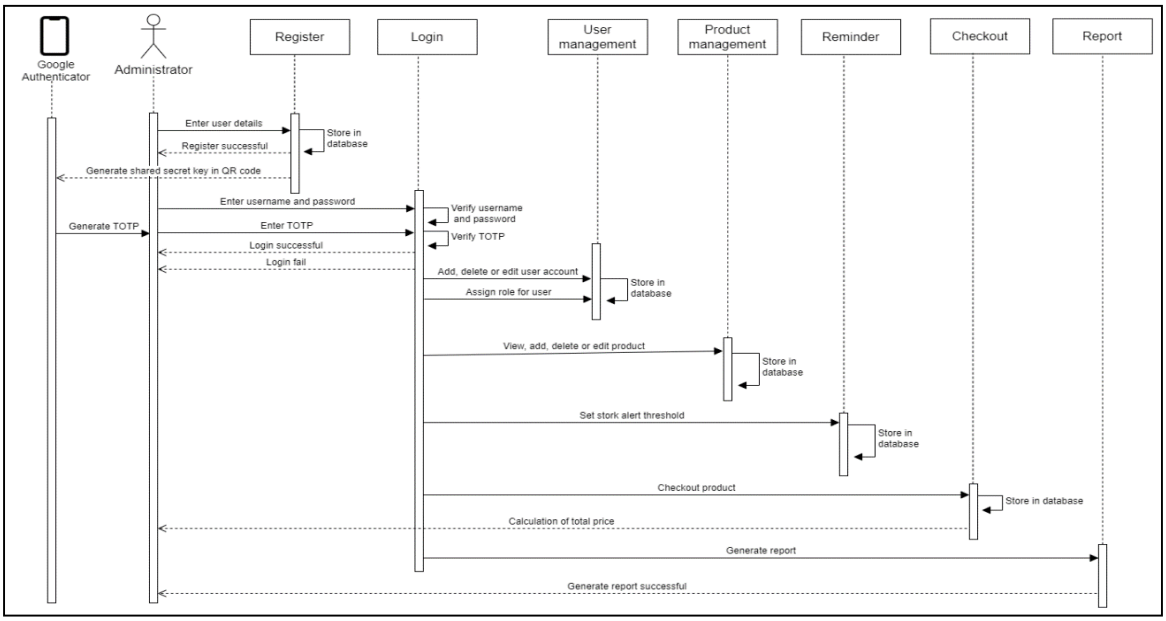


Fig. 5 *Sequence diagram for administrator*

Fig. 6 shows the sequence diagrams for manager. Manager has similar actions with administrator excluding create, edit and deactivate user account as well as assign role. Fig. 7 shows the sequence diagrams for stock clerk. Stock clerk can manage product and view stock alert reminder. Fig. 8 shows the sequence diagrams for cashier. After successfully login, cashier can view product and checkout product

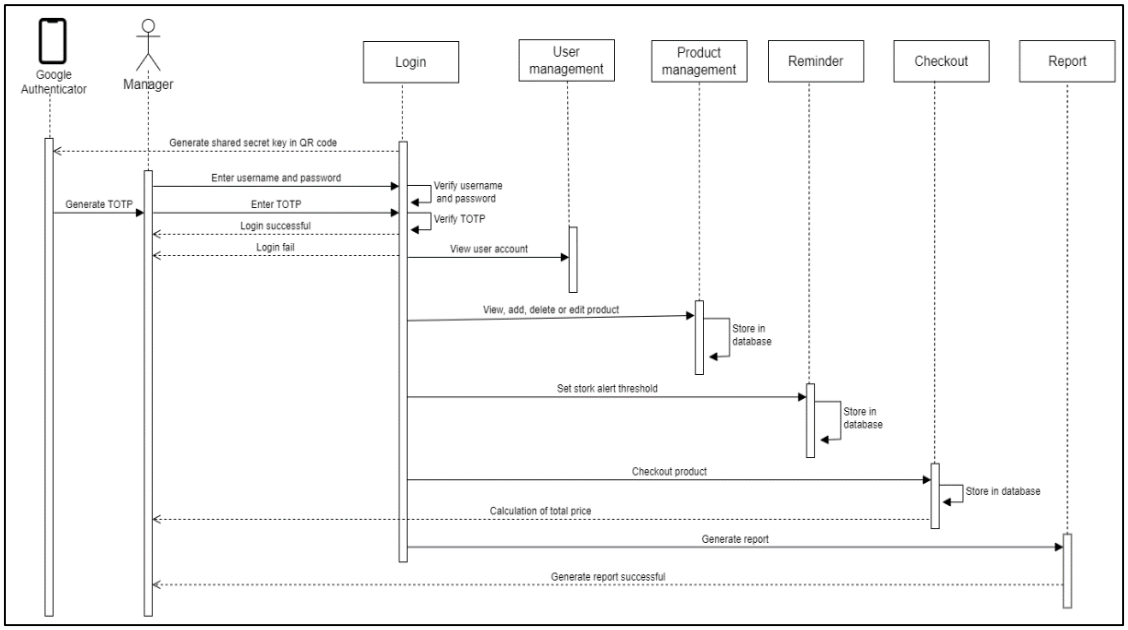


Fig. 6 *Sequence diagram for manager*

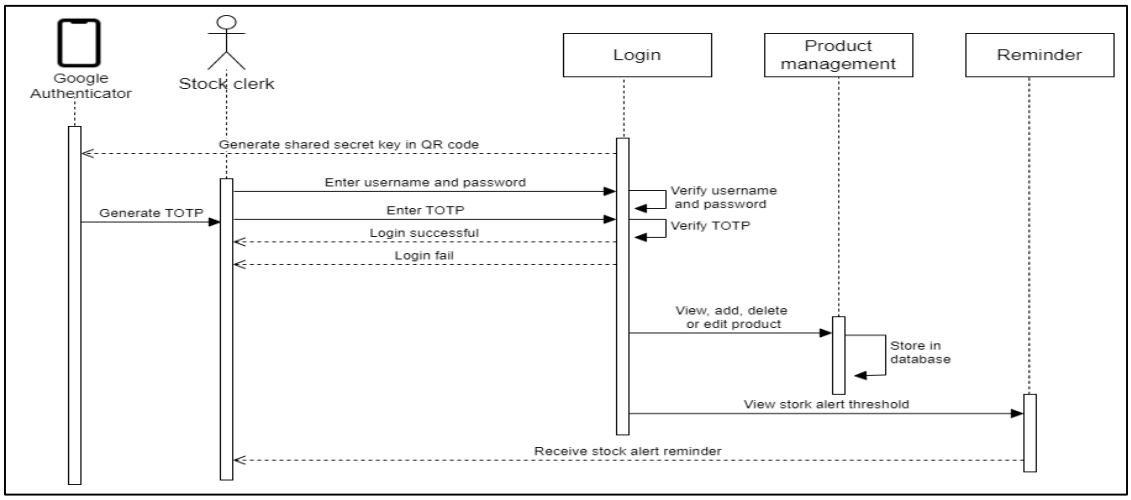


Fig. 7 *Sequence diagram for stock clerk*

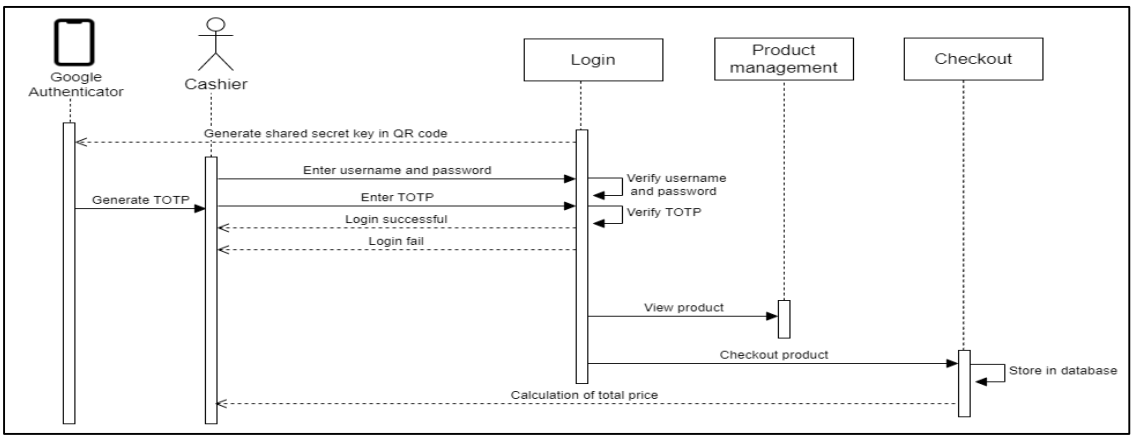


Fig. 8 *Sequence diagram for cashier*

4.3 System Flowchart

The system flowchart provides a clear representation to the flow of control and the order of operations in a system. It helps to understand the overall workflow of the system from beginning to end. Fig. 9 shows the flowchart of the proposed system. Once login to the system successfully, users are redirected to different panels based on their roles. Each role has different access privileges and can perform different actions. There are four characters in the diagram which are administrator, manager, stock clerk and cashier.

Applied Information Technology and Computer Science Vol. 6 No. 1 (2025) p. 343-361

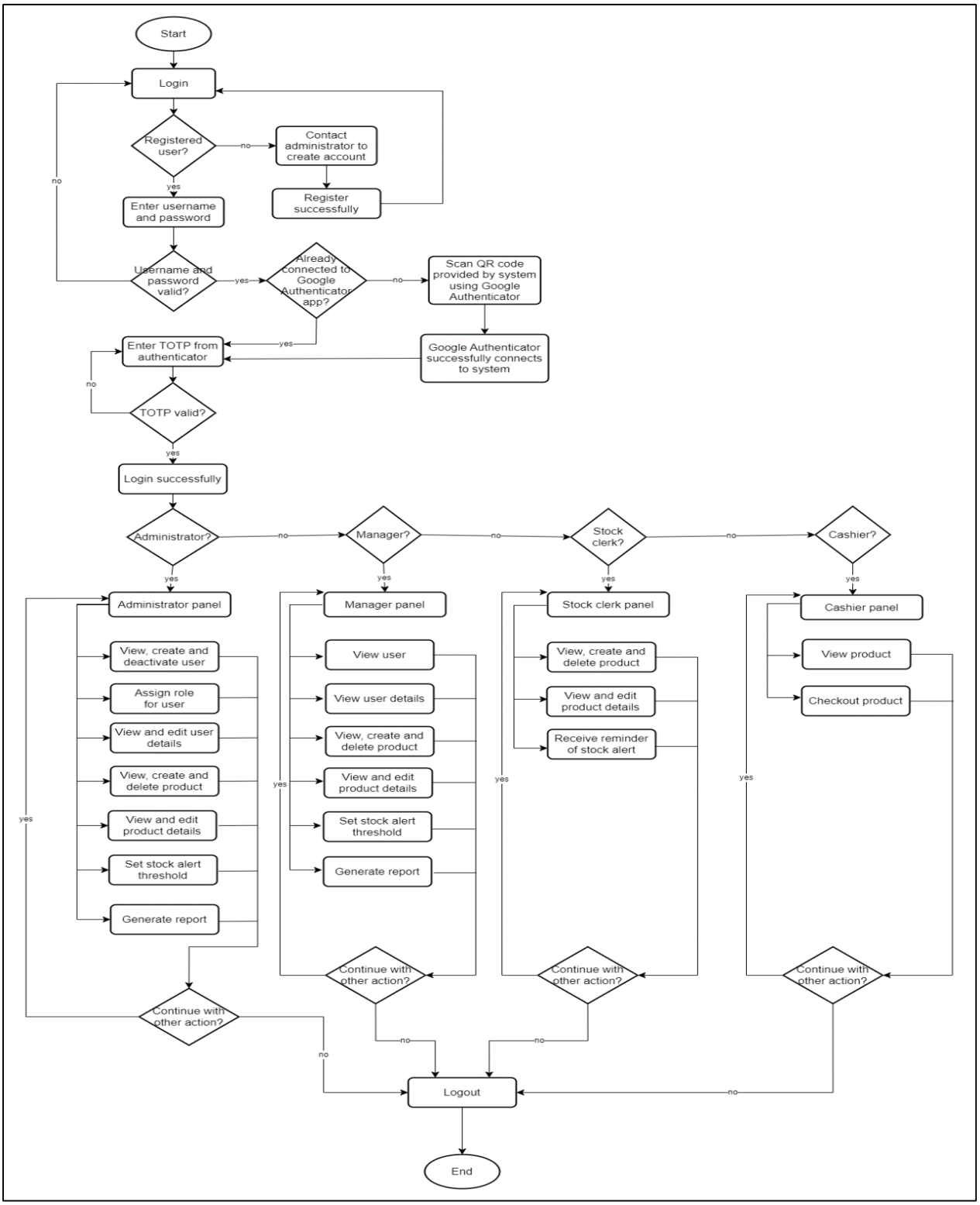


Fig. 9 *flowchart of the proposed system*

4.4 System Architecture

The system architecture is produced to organize and manage the structure and behavior of the proposed system as shown in Fig. 10. It provides a clear understanding of the system's structure. The system generates a secret key as QR code and shares it with Google authenticator. Users can scan the QR code provided by the system using Google authenticator to connect them. Google authenticator will generate TOTP to user for login. User with valid username, password and TOTP can login to the system successfully. The system able to manage inventory data, send stock alert reminder and generate sales report based on the inventory data. All gathered data within the system will be stored in a database.

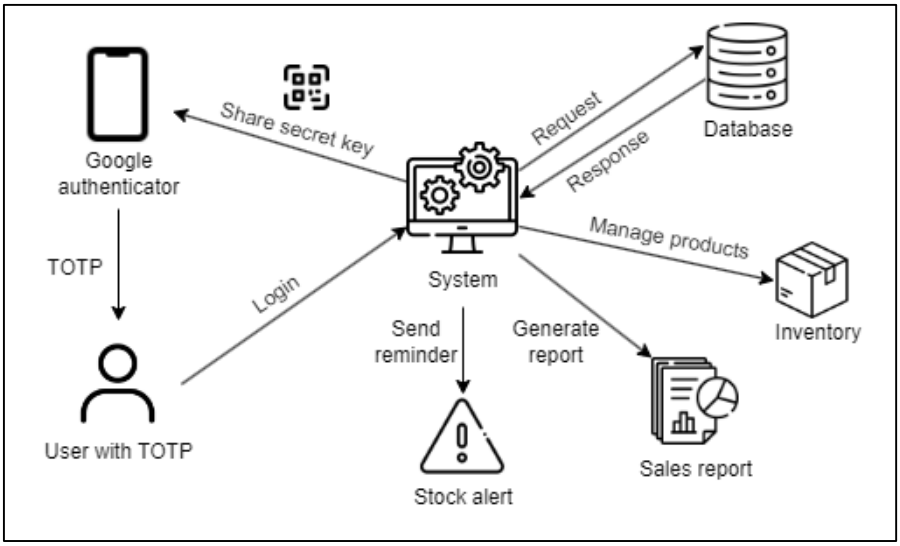


Fig. 10 *System architecture of the proposed system*

5. Implementation

This section explains the implementation of security modules of the proposed system.

5.1 Implementation of Security Module

In this section, the security modules implemented in the inventory management system for Heng Huat Motor & Electrical Trading are explained. There are four security modules implemented in the developed system which are multi-factor authentication, forgot password, input validation and role-based access control.

5.1.1 Implementation of Two-Factor Authentication (2FA)

In this inventory system, TOTP is implemented as part of 2FA during the login process. TOTP generates a unique temporary code to add an extra layer of security on top of traditional username and password verification. The Google2FA library is used to implement TOTP with Google authenticator. When a new user is created, a TOTP secret key is generated using the Google2FA library as shown in Fig. 11 and saved in the database.

```
if ($stmt = mysqli_prepare($link, $sql)) {
    // Generate a secret key for TOTP
    $ga = new PHPGangsta_GoogleAuthenticator();
    $secretKey = $ga->createSecret();
```

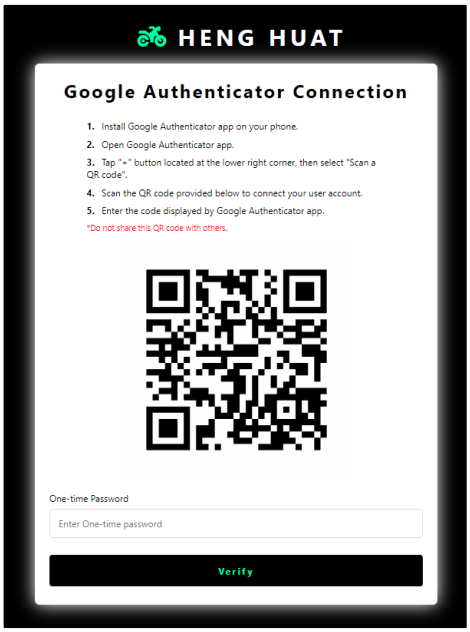
Fig. 11 *A secret key for TOTP is generated*

After verifying the username and password, a QR code is generated with the secret key as shown in Fig. 12 for first-time login users to scan with the Google Authenticator app on their phones and link the app to the inventory system. Once scanning the QR code, a connection is established between the Google Authenticator app and the system.

```php
// If the TOTP secret is found, generate the QR code
if ($totpSecret) {
    // Encode the TOTP secret into a URI format
    $otpAuthUri = sprintf('otpauth://totp/YourApp:%s?secret=%s', urlencode($username), $totpSecret);

    // Create a QR code renderer
    $renderer = new ImageRenderer(
        new \BaconQrCode\Renderer\RendererStyle\RendererStyle(400),
        new \BaconQrCode\Renderer\Image\SvgImageBackEnd()
    );

    // Generate the QR code image
    $writer = new Writer($renderer);
    $qrCode = $writer->writeString($otpAuthUri);
} else {
    // If TOTP secret is not found, handle the error
    echo "Error: TOTP secret not found for the user.";
}
```



(a) **(b)**

Fig. 12 *QR code generation with secret key (a) code; (b) interface*

Fig. 13 shows the process of validating the TOTP code. For subsequent login, the system retrieves the user's secret key from the database. It is used along with the current time to generate a TOTP code using the Google2FA library. This generated code is compared to the code entered by the user from the Google Authenticator app. If the TOTP code matches and is within the valid time period, the verification is successful. Once users successfully enter the TOTP code during their first login, their TOTP connection status is set to true in the database. So that users only need to enter the TOTP code from the Google Authenticator app on future logins after passing the username and password verification and the QR code will not be shown again.

```php
// Retrieve the TOTP secret for the logged-in user
$username = $_SESSION["username"];
$sql = "SELECT totp_secret_key FROM users WHERE username = ?";
$stmt = $link->prepare($sql);
$stmt->bind_param("s", $username);
$stmt->execute();
$stmt->bind_result($totpSecret);
$stmt->fetch();
$stmt->close();

// Validate the TOTP
$google2fa = new Google2FA();
if ($google2fa->verifyKey($totpSecret, $inputOTP)) {
    // TOTP is correct, update the totpConnection column
    $_SESSION["totp_authenticated"] = true;
    $sql = "UPDATE users SET totpConnection = ? WHERE username = ?";
    $stmt = $link->prepare($sql);
    $otpConnection = true; // Change to true to indicate TOTP connection
    $stmt->bind_param("ss", $otpConnection, $username);
    $stmt->execute();
    $stmt->close();

    echo "success";
} else {
    // TOTP is incorrect
    echo "Invalid code";
}
```

(a) **(b)**

Fig. 13 *TOTP validation (a) code; (b) interface of Google Authenticator app*

5.1.2 Implementation of Forgot Password

The "Forgot Password" feature allows users to securely reset their passwords if they forget them. This feature uses the PHPMailer library in PHP to send reset password emails with Gmail as the email service provider. When users click the "forget password" link on the login form, they are directed to the interface shown in Fig. 14.

Users are requested to enter their username and registered email address. The system checks the credentials against the database for a match.

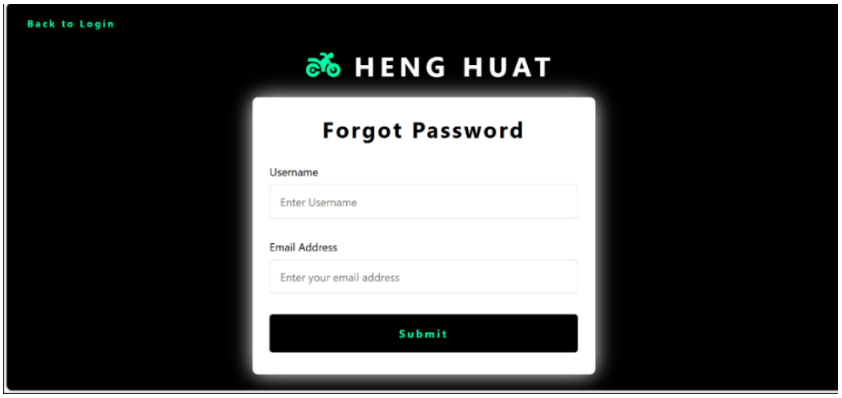


Fig. 14 *Forgot password interface*

Fig. 15 shows that a unique 16-byte cryptographic random token is generated and encoded to hexadecimal. The token is only given one hour expiry time from the current time. Both the token and token expiry are stored in the database.

```php
// Generate a unique token
$reset_token = bin2hex(random_bytes(16));
$reset_token_expiry = date("Y-m-d H:i:s", strtotime('+1 hour'));

// Update the database with the reset token and expiry
$sql = "UPDATE users SET reset_token = ?, reset_token_expiry = ? WHERE userId = ?";
```

Fig. 15 *Unique token is generated and stored*

An email containing the password reset link which includes the token as a parameter is sent from henghuat35@gmail.com to the user's email address as shown in Fig. 16. Fig. 17 shows the code for generating and sending the reset link.

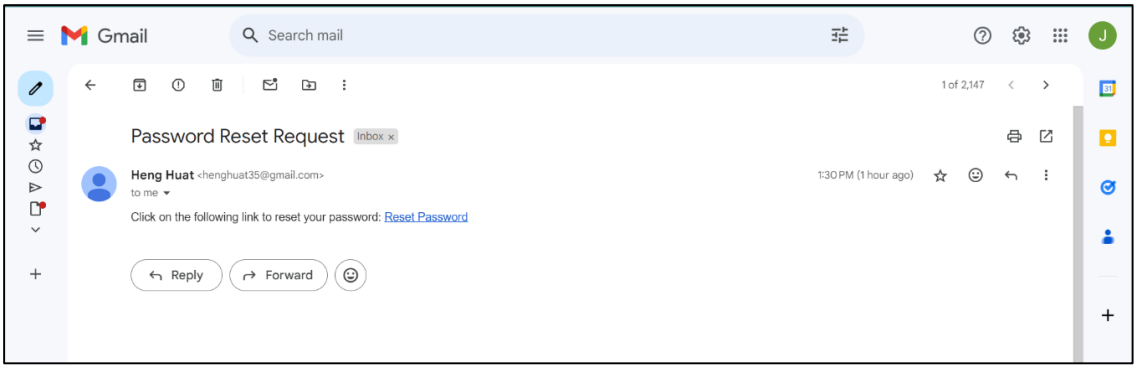


Fig. 16 *The reset link is sent to the user's email address*

```php
// Server settings
$mail->isSMTP();                                            // Send using SMTP
$mail->Host       = 'smtp.gmail.com';                       // Set the SMTP server to send through
$mail->SMTPAuth   = true;                                   // Enable SMTP authentication
$mail->Username   = 'henghuat35@gmail.com';                 // SMTP username
$mail->Password   = 'pfur tffu vxoc tyor';                  // SMTP password
$mail->SMTPSecure = PHPMailer::ENCRYPTION_STARTTLS;         // Enable TLS encryption; `PHPMailer::ENCRYPTION_SMTPS` en
$mail->Port       = 587;                                    // TCP port to connect to

// Recipients
$mail->setFrom('henghuat35@gmail.com', 'Heng Huat');
$mail->addAddress($email);     // Add a recipient

// Content
$mail->isHTML(true);                                  // Set email format to HTML
$mail->Subject = 'Password Reset Request';
$reset_link = "http://localhost/henghuat/resetPassword.php?token=" . $reset_token;
$mail->Body    = 'Click on the following link to reset your password: <a href="' . $reset_link . '">Reset Password</a>';
$mail->AltBody = 'Click on the following link to reset your password: ' . $reset_link;

$mail->send();
```

Fig. 17 *Code for generating and sending the reset link*

When the user clicks on the reset password link, the token in the URL is validated to ensure the request is from the legitimate account owner and the token expiry is checked against the database as shown in Fig. 18. This process reduces the risk of malicious use of the password reset link. If the token is valid, the user is directed to reset password page to reset their password. The consistent minimum requirements must be achieved as when creating a password. Once the new password is validated, it is hashed and updated in the database. The user is then requested to login again with the new password.

```php
// Validate the token
$token = htmlspecialchars($_GET['token']);
$sql = "SELECT userId, reset_token_expiry FROM users WHERE reset_token = ?";
if ($stmt = mysqli_prepare($link, $sql)) {
    mysqli_stmt_bind_param($stmt, "s", $token);
    if (mysqli_stmt_execute($stmt)) {
        mysqli_stmt_store_result($stmt);
        if (mysqli_stmt_num_rows($stmt) == 1) {
            mysqli_stmt_bind_result($stmt, $userId, $reset_token_expiry);
            mysqli_stmt_fetch($stmt);
            $current_time = date("Y-m-d H:i:s");
            if ($current_time > $reset_token_expiry) {
                echo "The token has expired.";
                exit;
            }
        } else {
            echo "Invalid token.";
            exit;
        }
    } else {
        echo "Oops! Something went wrong. Please try again later.";
        exit;
    }
    mysqli_stmt_close($stmt);
} else {
    echo "Oops! Something went wrong. Please try again later.";
    exit;
}
```

Fig. 18 *Validate the token*

5.1.3 Implementation of Input Validation

Input validation is crucial in web-based systems to ensure data integrity and prevent security vulnerabilities. It verifies that user inputs are correctly formatted and within expected parameters to reduce the risk of attacks such as SQL injection and cross-site scripting (XSS). All input fields are marked as 'required' in HTML to prevent empty submissions. Fig. 19 shows the PHP code of the input validation for username. The system ensures username only contain letters, numbers and underscores with no spaces and it checks for existing usernames in the database to prevent duplicates. Similar input validation is implemented for product names by limiting them to contain only letters, numbers, and spaces.

```javascript
// Function to validate username
function validateUsername() {
  const usernameRegex = /^[a-zA-Z0-9_]+$/; // Username can only consist of letters, numbers, and underscores

  if (usernameRegex.test(usernameField.value)) {
    // Hide any previous error
    hideError(usernameField, usernameError);
    usernameField.classList.remove("invalid");
    // Check if the username already exists
    const username = usernameField.value;
    const xhr = new XMLHttpRequest();
    xhr.open("POST", "sameUsername.php", true);
    xhr.setRequestHeader("Content-type", "application/x-www-form-urlencoded");
    xhr.onreadystatechange = function () {
      if (xhr.readyState == 4 && xhr.status == 200) {
        const response = xhr.responseText;
        if (response === "exists") {
          // Username already exists, show error message
          showError(usernameField, usernameError, "Username already exists. Please choose a different username.");
        }
      }
    };
    xhr.send("username=" + username);
  } else {
    // Show error if criteria not met
    showError(usernameField, usernameError, "Username can only consist of letters, numbers, and underscores.");
    usernameField.classList.add("invalid");
  }
}
```

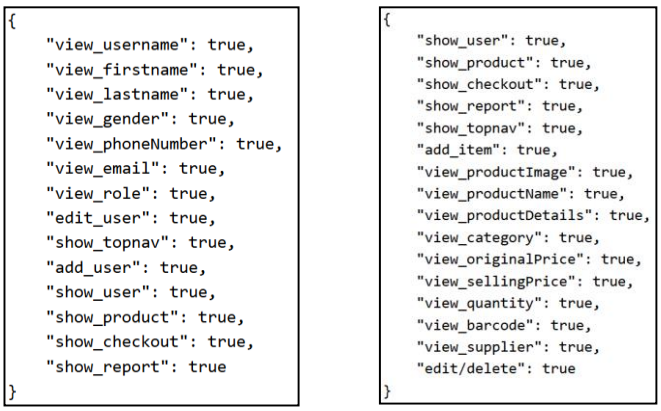
Fig. 19 *Input validation for username*

The system also checks the format of the password to ensure a strong password is used. Passwords length must be in between 10 to 20 characters and include at least one number, one lowercase letter, one uppercase letter and one special character. The system only allows letters and spaces in first and last names. Phone numbers are restricted to 10 or 11 digits and quantity contains only positive numbers. The system checks the email format to ensure only Gmail addresses are allowed by ensuring that the format ends with "@gmail.com". Product details can include letters, numbers, common symbols and spaces. Prices are restricted to numeric values. It is formatted to two decimal places and prefixed with "RM". Input validation is consistent across creating and editing users and products as well as for resetting and creating passwords.

5.1.4 Implementation of Role-Based Access Control (RBAC)

Role-Based Access Control (RBAC) assigns permissions to users based on their roles within Heng Huat. It ensures that users have access only to the resources necessary for their duties. In the database, the roles are saved in the roles table with roleId as a foreign key in the users table to assign roles to each user. The role_privileges table stores the actions each role can perform in JSON format for different pages using roledId as a foreign key. The data with no privileges is hidden for that role. The system has four roles which are admin, manager, stock clerk and cashier.

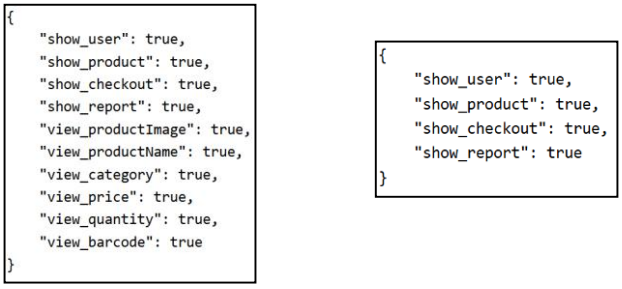
Fig. 20 shows the JSON code stored in privileges column in role_privileges table for admin in user management and product management. Admins have full access to user management and product management. For user management, they are allowed to view and edit user details such as username, first name, last name, gender, phone number and email. They also can create and deactivate the user accounts and assign roles. For product management, admins can view and edit product details such as product image, product name, product details, category, original price, selling price, quantity and supplier. They also can create and delete products. Similar privileges are assigned to managers excluding the ability in user management to create, edit and deactivate user accounts as well as assign roles to other users. Stock clerks have similar access in product management but no access to user management.



(a) **(b)**

Fig. 20 *JSON code for admin in (a) user management; (b) product management*

Fig. 21 shows the JSON code stored in privileges column in role_privileges table for admin in checkout and report. Admins have full access to checkout and report. For checkout, they are allowed to view product details such as product image, product name, category, selling price, quantity and barcode. Admin also can add products to the checkout list and confirm the transactions. In this checkout module, managers and cashiers have similar while stock clerks have no access. For report, admins can view the sales reports with filter by date. Managers have similar access while stock clerks and cashiers have no access.



(a) **(b)**

Fig. 21 *JSON code for admin in (a) checkout; (b) report*

6. Result and Discussion

This section presents the results of security test plan and user acceptance testing for each role.

6.1 Security Test Plan Result

The result of the security test plan is presented in Table 5 which indicates that the developed system has successfully passed the tests and performed as expected outcome.

Table 5 *Security test plan for the proposed system*

No.	Checklist	Result
1	The error message does not directly show which authentication data is incorrect such as "incorrect username" if user fails to login.	Pass
2	Enforce strong password policy for the password of user for login. The password should contain a minimum of 10 characters which consist of uppercase, lowercase, number and special character.	Pass
3	Verify that the password entered by user does not include SQL injection attack symbols such as '=' or '+'.	Pass
4	Ensure the password input length is restricted to a maximum of 20 characters.	Pass
5	Successful login is only allowed after passing both the username and password authentication and TOTP authentication.	Pass
6	Restrict user access privilege based on role.	Pass
7	Ensure password is obscured in the textbox.	Pass

6.2 User Acceptance Testing Result

User Acceptance Testing (UAT) was conducted by employees of Heng Huat Motor & Electrical Trading. The results of the tests for administrator, manager, stock clerk and cashier are presented in Table 6, Table 7, Table 8 and Table 9 respectively. The results show that the proposed system has successfully met the requirements of employees.

Table 6 *User Acceptance Testing result for administrator*

No.	Question	Test Result (Dissatisfied1- 5 Very Satisfied)				
		1	2	3	4	5
1	Administrator can connect Google Authenticator app with the user account.					√
2	Administrator can login account with valid username, password and TOTP.					√
3	Administrator can create new user account.					√
4	Administrator can assign role for all users.					√
5	Administrator can view user account details.					√
6	Administrator can edit user account details.					√
7	Administrator can delete user account.					√
8	Administrator can add product.					√
9	Administrator can view product added.					√
10	Administrator can edit product details.					√
11	Administrator can delete product.					√
12	Administrator can set stock alert threshold.					√
13	Administrator can checkout product.					√
14	Administrator can generate report.				√	

Table 7 *User Acceptance Testing form for manager*

No.	Question	Test Result (Dissatisfied1- 5 Very Satisfied)				
		1	2	3	4	5
1	Manager can connect Google Authenticator app with the user account.					√
2	Manager can login account with valid username, password and TOTP.					√
3	Manager can view user account details.					√
4	Manager can add product.					√
5	Manager can view product.					√
6	Manager can edit product details.					√
7	Manager can delete product.					√
8	Manager can set stock alert threshold.					√
9	Manager can checkout product.					√
10	Manager can generate report.				√	

Table 8 *User Acceptance Testing form for stock clerk*

No.	Question	Test Result (Dissatisfied1- 5 Very Satisfied)				
		1	2	3	4	5
1	Stock clerk can connect Google Authenticator app with the user account.					√
2	Stock clerk can login account with valid username, password and TOTP.					√
3	Stock clerk can add product.					√
4	Stock clerk can view product added.					√
5	Stock clerk can edit product details.					√
6	Stock clerk can delete product.					√
7	Stock clerk can receive stock alert reminder.					√

Table 9 *User Acceptance Testing form for cashier*

No.	Question	Test Result (Dissatisfied1- 5 Very Satisfied)				
		1	2	3	4	5
1	Cashier can connect Google Authenticator app with the user account.					√
2	Cashier can login account with valid username, password and TOTP.					√
3	Cashier can view product details					√
4	Cashier can add product to the checkout list.					√
5	Cashier can delete product from the checkout list.					√
6	Cashier can checkout product.					√

7. Conclusion

In conclusion, the inventory management system with two-factor authentication for Heng Huat Motor & Electrical Trading is complete. It successfully achieves objectives and goals based on project scope and user requirements.

This inventory management system has several advantages. It effectively helps Heng Huat Motor & Electrical Trading in systematically managing the inventory data with enhanced security measures such as implementing two-factor authentication with TOTP to significantly reduce the risk of unauthorized access. Its role-based approach also enhances security by ensuring that users only receive the minimum level of access necessary to perform their duties. Next, efficient product management and barcode scanning feature streamlines the product retrieval and management. The real-time inventory updates provided by the checkout module offer immediate visibility into inventory levels. Moreover, the reporting module provides analytical insights into product sales through data analysis and visualization. It helps Heng Huat to identify trends and capitalize on opportunities for growth.

While the inventory management system for Heng Huat Motor & Electrical Trading provides several advantages, there are also some disadvantages to consider. The implementation of two-factor authentication with TOTP may introduce complexity for users during the login process while enhancing security. This potentially leads to frustration adopting the system. Besides, the system's reliance on technology for barcode scanning and real-time inventory updates may pose a risk of system downtime which could disrupt operations. Moreover, the reporting and data analysis capabilities of the system are less flexible and thus limit Heng Huat to get useful insights from its inventory information. The lack of scalability of the proposed system also decreases the operational efficiency and limits the ability to effectively manage inventory as business expands.

There are some improvements that can be made to the system for future implementation to overcome its weaknesses. First, simplify the two-factor authentication process by using user-friendly methods such as biometric authentication to reduce complexity during login. Second, integrate advanced reporting and data analytics tools to provide dynamic control and deeper insights into inventory data. Third, design the system with scalability by using cloud-based infrastructure to easily scale resources based on inventory levels and business needs. Finally, the barcode scanning functionality can be improved by adding function to automatically checkout the products instead of just searching for the product ID.

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

The authors confirm contribution to the paper as follows: ***study conception and design:*** *J. Y. Chia, N.Z. Harun;* ***data collection:*** *J. Y. Chia, N.Z. Harun;* ***analysis and interpretation of results:*** *J. Y. Chia, N.Z. Harun;* ***draft manuscript preparation:*** *J. Y. Chia, N.Z. Harun. All authors reviewed the results and approved the final version of the manuscript.*

References

[1] A. Idlan Azmi, N. Selamat, and F. Sains Komputer dan Teknologi Maklumat, "Sales and Inventory System for Maperow Store," Applied Information Technology And Computer Science, vol. 3, no. 1, pp. 849–867, 2022, doi: 10.30880/aitcs.2022.03.01.057.

[2] D. Plinere and A. Borisov, "Case Study on Inventory Management Improvement," Information Technology and Management Science, vol. 18, Nov. 2015, doi: 10.1515/itms-2015-0014.

[3] E. E. Adam and R. J. Ebert, Production and Operation Management: Concepts, Models and Behaviour, 4th ed. Englewood Cliffs, New Jersey: Prentice Hall International, 1982.

[4] C. Acemyan, P. Kortum, J. Xiong, and D. Wallach, "2FA Might Be Secure, But It's Not Usable: A Summative Usability Assessment of Google's Two-factor Authentication (2FA) Methods," Proceedings of the Human Factors and Ergonomics Society Annual Meeting, vol. 62, pp. 1141–1145, Nov. 2018, doi: 10.1177/1541931218621262.

[5] M. Eldefrawy, K. Khan, K. Alghathbar, T.-H. Kim, and H. Elkamchouchi, "Mobile one-time passwords: Two-factor authentication using mobile phones," Security and Communication Networks, vol. 5, pp. 508–516, Nov. 2012, doi: 10.1002/sec.340.

[6] D. U. Balasta, S. C. Marie Pelito, M. R. Christopher Blanco, A. J. Alipio, K. E. Mata, and D. A. Michael Cortez, "Enhancement of Time-Based One-Time Password for 2-Factor Authentication," 2022. [Online]. Available: www.ijisrt.com563

[7] L. Lumburovska, J. Dobreva, S. Andonov, H. M. Trpcheska, and V. Dimitrova, "A Comparative Analysis of HOTP and TOTP Authentication Algorithms. Which one to choose?"

[8] E. Erdem and M. T. Sandıkkaya, "OTPaaS—One Time Password as a Service," IEEE Transactions on Information Forensics and Security, vol. 14, no. 3, pp. 743–756, 2019, doi: 10.1109/TIFS.2018.2866025.

[9] D. Ferraiolo and D. Kuhn, "Role-Based Access Controls," Nov. 2009.

[10] Q. Liu, H. Zhang, J. Wan, and X. Chen, "An Access Control Model for Resource Sharing Based on the Role-Based Access Control Intended for Multi-Domain Manufacturing Internet of Things," IEEE Access, vol. PP, p. 1, Nov. 2017, doi: 10.1109/ACCESS.2017.2693380.

[11] A. Shabdar, "Running Your Business on Zoho CRM," 2017, pp. 61–130. doi: 10.1007/978-1-4842-2904-0_4.

[12] A. Arvianto, Z. Rosyada, S. Saptadi, W. Budiawan, and Y. Demilda, "ERP ODOO IMPLEMENTATION IN SMALL RETAILERS," International Journal of Applied Science and Engineering Review, vol. 03, pp. 66–85, Nov. 2022, doi: 10.52267/IJASER.2022.3605.

[13] A. Schiff and J. Szendi, "Helping small business entrepreneurs avoid critical mistakes in QuickBooks accounting software," Entrepreneurial Executive, vol. 19, pp. 169–181, Nov. 2014.

[14] S. Sharma, D. Sarkar, and D. Gupta, "Agile Processes and Methodologies: A Conceptual Study," International Journal on Computer Science and Engineering, vol. 4, Dec. 2012.

[15] R. Malan, D. Bredemeyer, and B. Consulting, "Functional Requirements and Use Cases," 2001. [Online]. Available: http://www.bredemeyer.com