

A Development of Inventory Management System for Sin Guan Leong Grocery Store

Low Hao Xian¹, Mazidah Mat Rejab^{1*}

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

DOI: <https://doi.org/10.30880/aitcs.2022.03.02.051>

Received 07 August 2022; Accepted 25 October 2022; Available online 30 November 2022

Abstract: Inventory management is a very important and common system used by every store in daily life to assist in managing their inventory. However, Sin Guan Leong's Inventory management system is currently manual, and this system is inconvenient for managing grocery stores. All product records are recorded by employees in handwriting and such documents will cause problems during detection and are difficult to keep. It will also cause human error due to misunderstandings between employees. Therefore, an inventory management system for Sing Guan Leong Grocery Store is needed by helping employees manage their groceries more effectively and efficiently. This system is designed based on an object oriented approach. The methodology used is the waterfall methodology. The user will be the administrator and worker in grocery store. There are five modules in the system namely login module, inventory module, cashier module, supplier module and report generation module. Reports generated by the system are able to help retail store owners determine which products are selling well so that owners can refill products when the products run out. Through the system, users can record all product movements in the system systematically and store them in the system. The system is expected to provide various types of modules to help employees in retail stores more easily manage products in stores. Lastly, functional requirements and user acceptance testing results show that the system fulfills the requirement. There are some improvements that can be done such as improving the security of the system.

Keywords: Grocery, Stock, Inventory, Inventory Management System, Barcode

1. Introduction

Sin Guan Leong Grocery is an old style Grocery store which has operated for many years. It is built in 7, Hala Kalui, Seberang Jaya, 13700 Perai, Pulau Pinang. The Grocery operates every day and there are thousands of products sold in the grocery. It is a grocery that sells many types of Goods such as foods, drinks, and other daily necessities products.

The process of managing the product in the store is manually. In the Grocery Store, all the items or products are managed manually by keeping the record of the product on papers, for example the product name, quantity and price. It required the worker to make a store check if they wanted to know the product remaining. Sometimes, the product quantity is not the same with the hand written records, it will be more or less compared to the hand written document. The workers in the store are hard to track the error on the papers if they do not do the store check. The duplicate information might be recorded and cause other workers to confuse the record. There is the same situation for when the in stock, where the product record will be added in the document, the unsystematic system will cause a lot of problems that are hard to solve. If a product is sold from the grocery, there will be a record by the worker there with hand written and make summaries at the end of the day using the hand written document. If a product is sold from the grocery, there will be a record by the worker there with handwritten and made summaries at the end of the day using the hand written document.

The duplicate or wrong information will cause an increase of workload to manage the product and goods. Since the records from the document are wrong, hence the mistake might happen to the worker and make a lot of problems such as wrong product name listed or wrong price listed.

Also, due to all the record documents being handwritten, the product is hard to track by the worker, it might cause a different item quantity record compared to the real product quantity in the grocery. The handwritten record for the shop is hard to keep, it needs a lot of places to keep and it will need the worker a lot of time to get the information from the document. The whole progress of the currently inventory management system are fully manual and request a lot of workload and time to complete,

The current management system used in the grocery is not convenient and not efficient, therefore a systematic inventory management system must be proposed to the grocery to make it easy to manage the product. Inventory management has become one of the key elements of the supply chain management and can greatly affect the performance of a business [1].

This article is organized into five sections. The first part is an introduction describing the context of the project. The second section describes the analysis of the relevant work. In the third section, the methodology is explained. The implementation and testing of this system is described in the fourth section. In the last section, a conclusion with some instructions for future employment is given.

2. Related Work

Three existing systems are investigated to examine the good features to be employed in the new system.

2.1 Inventory Management for Emart Grocery Shop

This is a web-based inventory management system developed by Chandra Sekhar et al [2]. The stakeholders of the Inventory management system are the workers in the Emart. The worker in the grocery shop can control the stock and know the remaining stock. Also, it can manage the details of customer, inventory, payment, purchasing and supplier. The worker can authorize the user and group of new incoming workers. All users must be logged in before using the features for the grocery system. All users can manage customers such as add, view and delete. It is also able to manage vendors and products which add, view and delete the information saved. It can also manage purchases and inventory.

2.2 Web Based Inventory Management System in LotteMart Solo Baru

LotteMart is an international company that started in South Korea and now the business has grown rapidly. The company now has 48 stores with two different types of store, wholesale and retail in Indonesia. Web based inventory management system in LotteMart Solo Baru is developed by ADONIS PALLAS SUTANTO [3]. The system consists of few stakeholders, which is the

administrator, manager and staff. The administrator has the highest access level which is able to access all the features such as create, read, update and delete users and delete and edit recorded data. Managers can access read transaction features and read general records. Staff can access the input goods arrival and departure feature, correcting arrival and departure features.

2.3 Supermarket Management System Using Barcode Technology for Star Supermarket in Adama City

This system is a web based system that manages the product in the supermarket. Supermarket management system using barcode technology is a web based system that automates the supermarket's sales activity [4]. It enables for managing the stock details, maintaining the records of the sales done for a specific period of time and monitoring the expiry status of the items from time to time. There will be four stakeholders in this system which are the administrator, salesman, supermarket manager and data encoder. The first function will enable the administrator to login by entering name and password. Administrators are able to create new users into the system by inserting information such as username and password. Also, the new user created can be modified and deleted by the admin. All users are able to create, edit and delete an item/category in the system. The system is also able to print stock reports and make summaries for the stock. If the user has a barcode scanner, the system is able to search the product and display it to the user. After that the salesman can print the bill to customers.

2.4 Inventory management system for Sin Guan Leong Grocery Store

There are few modules in the system which are login module, inventory module, and cashier module. For Login module, it allow user to log in, change password and log out from the system For Inventory module, it allow user to view all the stock information and all invoice and all details. Also can create, add, delete product, increase or decrease the product quantity, Product with low quantity will be easily recognize by worker to refill the product. All the product can be filter by category. For Cashier module, it can create an invoice or receipt during product sell, refund for the defect product, and print the summary for the product sell. All the products can be tracked by using the barcode number or the product name easily. For supplier module, it able user to add and edit the supplier name and other details, add the quantity of the product, and view all stock details and summary for the supplier. Another module will be report module, in this module, a report for the product can be show, such as product sale report.

Table 1 summarized the features comparison of 3 related systems and the proposed system. The proposed system has all the modules and features compared to the reference system. Compared to the existing system, the proposed system is using the barcode technology which enables users to enter products by using barcode scanning. Barcode, a printed series of parallel bars or lines of varying width that is used for entering data into a computer system [5].

Table 1: System Comparison

Features	A	B	C	D
Login	Yes	Yes	Yes	Yes
Manage Staff Information	No	Yes	Yes	Yes
Add/Search Inventory	Yes (manual input)	Yes (manual input)	Yes (Scan Barcode)	Yes (Scan Barcode)
Add Categories	No	No	Yes	Yes
Update Inventory	Yes	Yes	Yes	Yes
Display Inventory	No	No	No	Yes
Track Inventory product	Yes	Yes	Yes	Yes
Create Purchase Order	Yes	Yes	Yes	Yes
Create Low Product Quantity Reminder	No	No	Yes	Yes
Create Supplier Record	Yes	Yes	Yes	Yes
Generate Report	No	No	Yes	Yes
Usability platform	Web-based	Web-based	Web-based	Web-based

3. Methodology

This project begin from 1 September 2021 until 30 June 2022. The waterfall model is used in the system development. Table 2 present the activities and the deliverable outcomes in each phases during the software development life cycle in the project

Table 2: Software Development Activities and Deliverable for Phases in Waterfall Model

Phase	Activity	Deliverable
Planning	<ul style="list-style-type: none"> Identify background, problem, objective, scope of the project. Determine cost and possible risk. Determine project schedule. 	<ul style="list-style-type: none"> System proposal Gantt Chart
Analysis	<ul style="list-style-type: none"> Interview stakeholder. Gather and analyze system requirements. Determine development tools. 	<ul style="list-style-type: none"> System requirements UML use case diagram UML use case specification Class diagram To-be model Requirements traceability matrix (RTM)
Design	<ul style="list-style-type: none"> Design system architecture. Design database. Design user interface. 	<ul style="list-style-type: none"> Database Design Interface Design
Development	<ul style="list-style-type: none"> Coding and programming. 	<ul style="list-style-type: none"> System Prototype
Testing	<ul style="list-style-type: none"> Acceptance test from stakeholder. Repeat analysis, design and implementation for improvement. 	<ul style="list-style-type: none"> Test Cases Test Report Results Finalized Prototype

3.1 Requirement Analysis

This phase collects all of the necessary requirements for an interactive system or device and yields a requirements specification or document as its outcome. The requirement analysis helps in development process. Functional Requirement define the basic system behavior of the system. Table 3 shows the system functional requirements.

Table 3: Functional Requirements based on System Functional Module

Module	Description
Login Module	<ul style="list-style-type: none"> The system should allow user to login into the system using correct username and password The system should allow to alert the user for any invalid input The system should give respond when user successful login. The system should allow user to change password. The system should allow user to log out.

Inventory Module	<ul style="list-style-type: none"> • The system should allow user to view all the product information. • The system should allow user to create, add, modify, delete a product information • The system should allow user to filter the product by categories. • The system should remind the user when product is running low.
Cashier Module	<ul style="list-style-type: none"> • The system should allow user to generate an invoice and receipt when product is sell • The system should allow user to make a refund for a product.
Supplier Module	<ul style="list-style-type: none"> • The system should allow user to create and modify the supplier information • The system should allow user to create, modify and delete a stock in record.
Report Module	<ul style="list-style-type: none"> • The system should allow user to get a monthly report for all product sell. • The system should allow user to get a daily report for all product sell.

Table 3: (cont)

Non Functional requirement are also important during system development, it specify how the system should do it. Not only that, it is also will affect the user experience. Table 4 shows the system non-functional requirements.

Table 4: System Non-Functional Requirements

Requirements	Description
Performance	<ul style="list-style-type: none"> • The system should be usable at all times
Operational	<ul style="list-style-type: none"> • The loading time required for a website is no more than 1 minute
Security	<ul style="list-style-type: none"> • The system should be user friendly • The system information should be secured
Cultural and political	<ul style="list-style-type: none"> • The system should be able to work on any web browser

The user requirement defines the expectation function should be in the system. Table 5 shows the system user requirements.

Table 5: User Requirements

No.	User Requirements
1.	Users must be able to enter a valid username and password to login the system.
2.	Users must be able to change the password of the system.
3.	Users must be able to log out from the system.
4.	Users must be alert by the system when invalid input.
5.	Users must be able view all the product information.
6.	Users must be able to create, add, modify and delete a product information

Table 5: (cont)

7.	Users must be able to filter the product by the categories.
8.	Users must be able to generate the invoice and receipt when product sell.
9.	Users must be able to do refund on product.
10.	Users must be able to create, modify, and delete the supplier information.
11.	Users must be able to create, modify and delete a stock in record.
12.	Users must be able to view the daily and monthly product sell report.

3.2 System Analysis

Systems analysis is the process by which an individual (s) studies a system such that an information system can be analyzed, modeled, and a logical alternative can be chosen [6]. A UML use-case diagram shows the relationships among actors and use cases within a system [7]. Figure 1 shows the Use Case Diagram for the system.

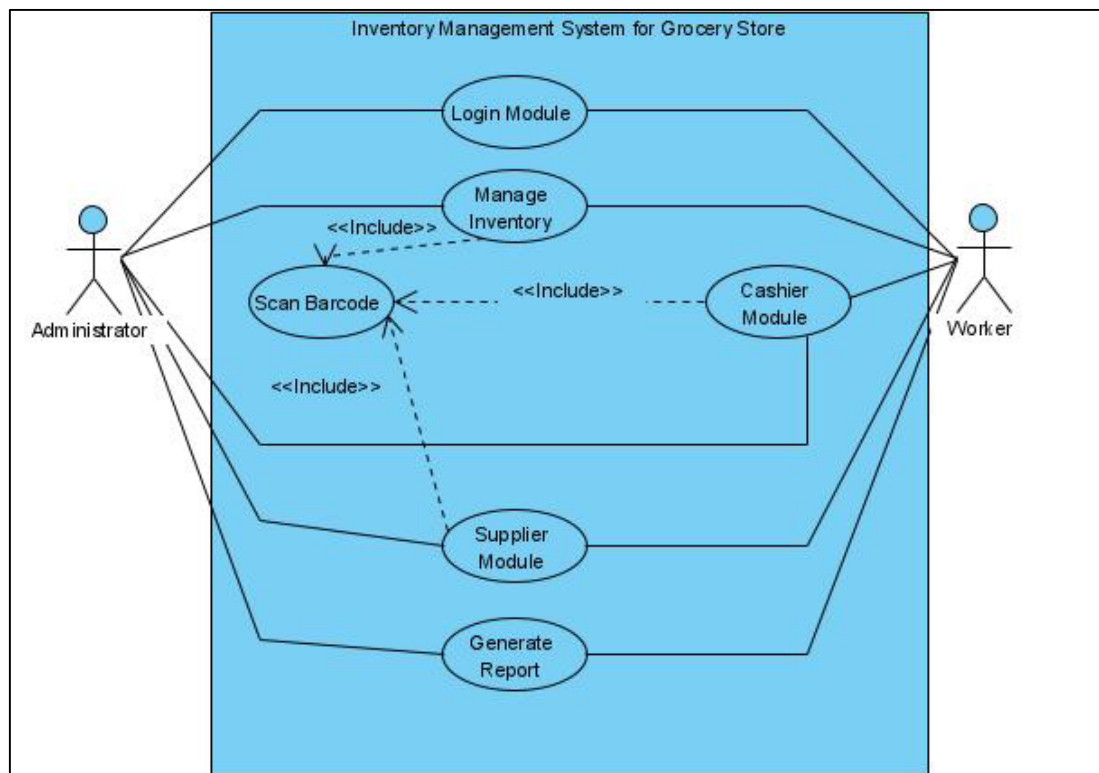


Figure 1: Use Case Diagram

According to Figure 1, the use case actor is the administrator and the worker which will interact with the system. There are 5 use cases interacting with the 2 actors. The association line between the actor and use case shows the relationship between the actor and use cases.

A class diagram is a static model that shows the classes and the relationships among classes that remain constant in the system over time [8]. Figure 2 shows the class diagram for the proposed system.

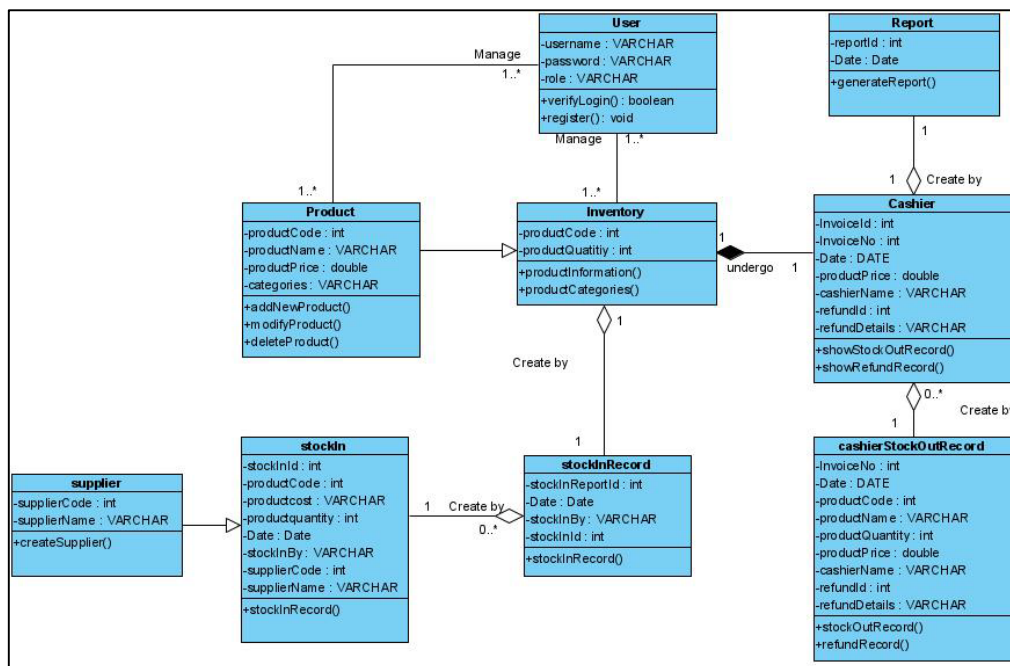


Figure 2: Class Diagram

The implementation of the proposed Inventory management system will modify the business model of the grocery store. The to-be model in Figure 3 presents the future inventory management process for the grocery store.

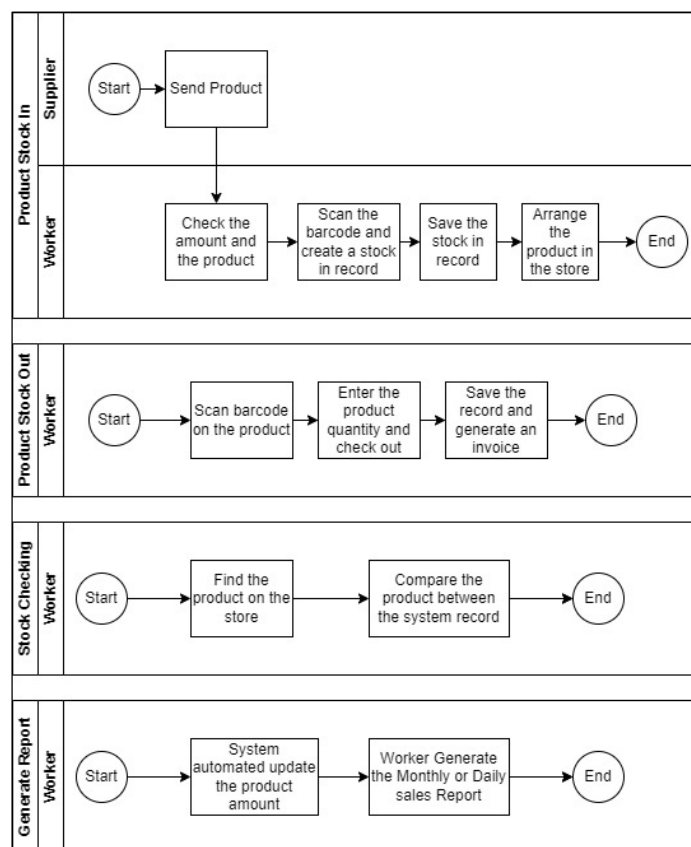


Figure 3: To-be Model

3.3 System Design

The system architecture use is client server architecture which the client is responsible for the presentation logic, whereas the server is responsible for the data access logic and data storage [9]. Figure 4 show the database design of the proposed system.

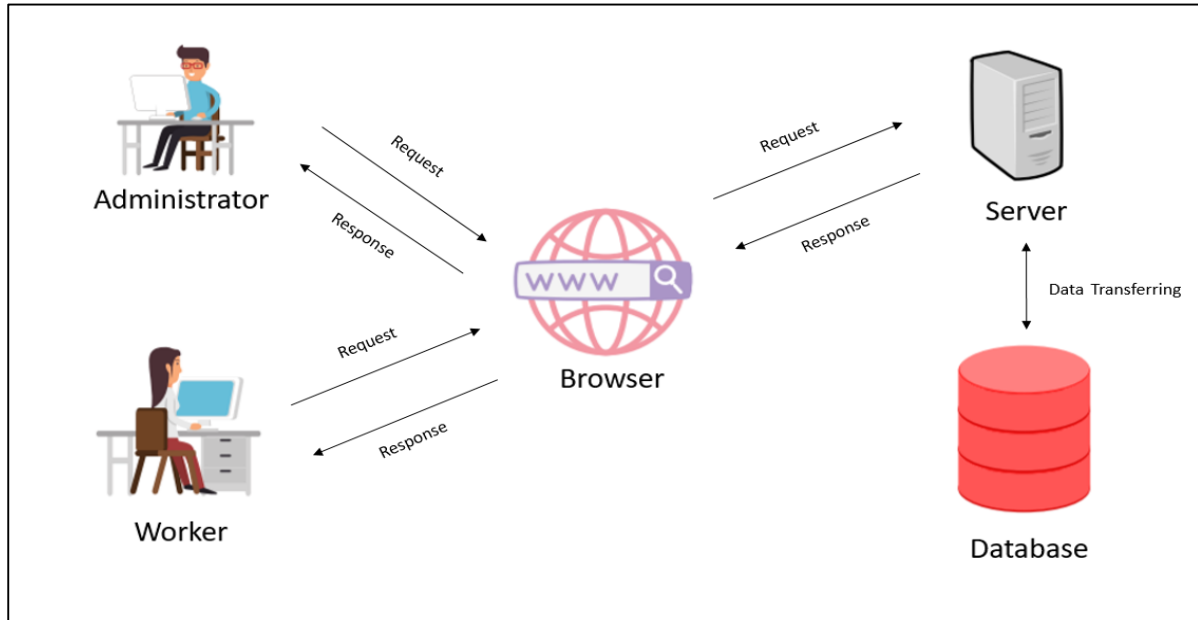


Figure 4: System Architecture

Database design refers to the activities that focus on the design of the database structure that will be used to store and manage end-user data [10]. Figure 5 shows the data schema of the proposed system.

i.	checkin (<u>id</u> , stockInId, productCode, productCosting, productQuantity, Date, username)
ii.	checkinreport (<u>checkInReportID</u> , stockInId, stockInDate, supplierId, date, username)
iii.	checkout (<u>invoiceId</u> , Date, cashierName, totalPrice, datetime)
iv.	checkoutlist (<u>productListId</u> , productCode, productQty, productPrice, invoiceId)
v.	Login (<u>ID</u> , username, password, role)
vi.	notification(<u>Id</u> , productCode, date)
vii.	productcategories(<u>categoriesId</u> , categories)
viii.	Refund (<u>refundId</u> , productCode, productQty, productPrice, cashierName, date, remark)
ix.	Report(<u>Id</u> , productCode, productQty, productPrice, reportId)
x.	Reportlist (<u>reportId</u> , datetime, username, start, end)
xi.	Supplier(<u>supplierId</u> , supplierName, supplierDetails)

Figure 5: Data Schema

A user interface, as recently described, is a collection of techniques and mechanisms to interact with something [11]. User Interface (UI) is the way to let human communicate and interact with a device or a system. The user interface for the proposed system is shown in Figure 6, Figure 7, Figure 8 and Figure 9.

Navigation Menu			
Inventory			
Product ID	Product Name	Product Qty	Actions
Barcode	Biscuit	13	<input type="button" value="Edit"/> <input type="button" value="Delete"/>

Figure 6: Manage Inventory Page

Navigation Menu				
Check In Record				
Check ID	Supplier Name	Date	By	Actions
Barcode	Biscuit	Date	Admin	<input type="button" value="Edit"/> <input type="button" value="Delete"/>

Figure 7: Manage Check In Page

Navigation Menu			
Refund			
Product ID	Product Name	Product Qty	Price
Barcode	Biscuit	2	RM 13.00

Reason

Total RM 13.00

Figure 8: Manage Refund Page

Navigation Menu				
Generate Report				
Product ID	Product Name	Product Qty	Price	Product Subtotal
Barcode	Biscuit	13	RM 4.00	RM 52.00

Figure 9: Generate Report Page

4. Results and Discussion

4.1 Implementation

The user login interface required the user to provide the correct user, password and role of the user to successfully log into the system. If one of the information is not matched. The login will be failed and the system will display an error message. Also, after login, the user can log out at any time by pressing the log out button at the navigation bar. The user login interface of the system is shown in Figure 10. The code segment for the login form is shown in Figure 11.

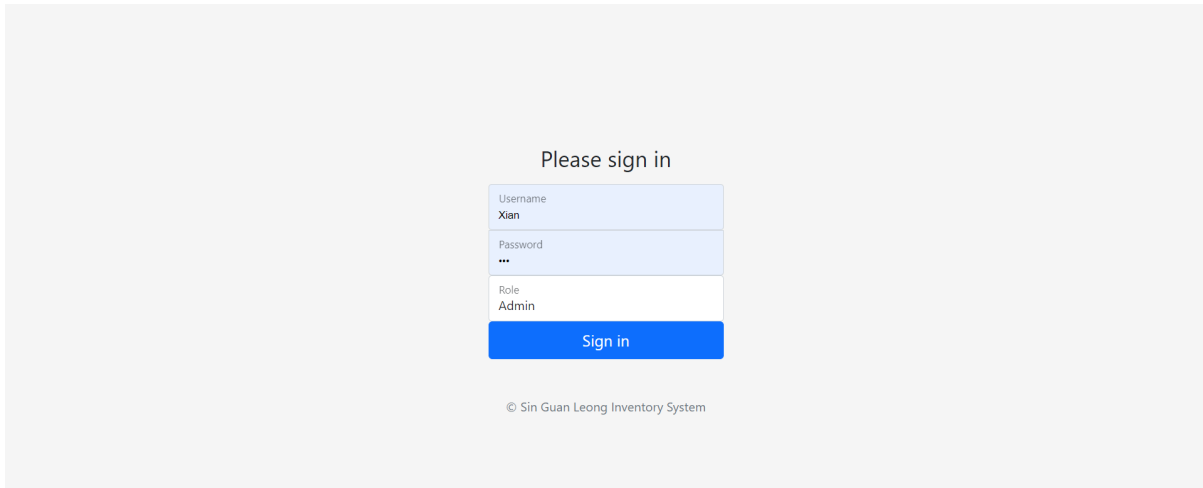


Figure 10: Login Interface

```
<body class="text-center">
  ...
  <main class="form-signin">
    <form method="POST">
      ...
      <h1 class="h3 mb-3 fw-normal">Please sign in</h1>
      ...
      <div class="form-floating">
        <input type="text" class="form-control" id="floatingInput" name="username" placeholder="Username" required>
        <label for="floatingInput">Username</label>
      </div>
      <div class="form-floating">
        <input type="password" class="form-control" id="floatingPassword" name="password" placeholder="Password" required>
        <label for="floatingPassword">Password</label>
      </div>
      <div class="form-floating">
        <select name="role" class="form-control" id="role" placeholder="Role">
          <option value="Admin">Admin</option>
          <option value="Worker">Worker</option>
        </select>
        <label for="role">Role</label>
      </div>
      ...
      <button class="w-100 btn btn-lg btn-primary" type="submit">Sign in</button>
      <p class="mt-5 mb-3 text-muted">&copy; Sin Guan Leong Inventory System</p>
    </form>
  </main>
</body>
```

Figure 11: Code Segment for Login Interface

In product module, user are able to create a new product and manage the inventory information. The user can also preview all product information in a listed table. The interface of create new product form is shown in the Figure 12. The code segment for create new product form is shown in Figure 13.

Figure 12: Create New Product Interface

```

<main class="form-signin">
  <form method="POST" class="container">
    <h1 class="h3 mb-3 fw-normal">Create Product</h1>
    <!-- Product Information -->
    <div class="form-floating">
      <input type="text" class="form-control" id="floatingInput" name="productCode" placeholder="Product Code" required>
      <label for="floatingInput">Product Code</label>
    </div>
    <div class="form-floating">
      <input type="text" class="form-control" id="floatingInput" name="productName" placeholder="Product Name" required>
      <label for="floatingInput">Product Name</label>
    </div>
    <div class="form-floating">
      <input type="currency" class="form-control mb-2" id="floatingInput" name="productPrice" placeholder="Product Price" required>
      <label for="floatingInput">Product Price</label>
    </div>
    <div class="form-floating">
      <input type="number" class="form-control" id="floatingInput" name="productQty" placeholder="Product Quantity">
      <label for="floatingInput">Product Quantity</label>
    </div>
    <!-- Product Categories -->
    <div class="form-floating">
      <select class="form-control" id="categories" name="categories">
        <option disabled selected-->Categories --</option>
        <?php
          $records = mysqli_query($con, "SELECT categoriesId, categories From productcategories"); // Use select query
          while($data = mysqli_fetch_array($records))
          {
            echo "<option value='". $data['categories'] ."'>". $data['categories'] . "</option>"; // displaying data in option menu
          }
        <?>
      </select>
      <label for="categories">Categories</label>
    </div>
    <button class="w-100 btn btn-lg btn-primary" type="submit">Create Product</button>
  </form>

```

Figure 13: Code Segment for Create New Product Interface

In Cashier module, the user can create an invoice record for check out process and refund record for any defect item return from customer. It also able to print a receipt for check out record. All for the record created can be view, edit and delete by user and all for the action are record by the system. Figure 14 shown the check out record interface. The code segment for check out record Interface is shown in Figure 15.

No. Invoice & Date Created	Cashier	Total Price	Actions
6 2022-06-08	Xian	RM 1,671.00	⋮
5 2022-06-08	Xian	RM 0.00	⋮
4 2022-06-05	Xian	RM 500.00	⋮
1 2022-06-05	Xian	RM 1,998.00	⋮
3 2022-05-30	Admin	RM 0.00	⋮
2 2022-05-30	Admin	RM 0.00	⋮
0 0000-00-00	Admin	RM 0.00	⋮

Figure 14: Check Out Record Interface

```

<?php
$sql = "SELECT invoiceId, Date, cashierName, totalPrice FROM checkout ORDER BY Date desc, invoiceId desc";
$result = $con->query($sql);
if ($result->num_rows > 0) {
    // output data of each row
    echo '<div class="table-responsive">'.
        '<table class="table text-nowrap" id="myTable">'.
        '<thead class="table-light">'.
        '<tr>'.
        '<th>No. Invoice & Date Created</th>'.
        '<th>Cashier</th>'.
        '<th>Total Price</th>'.
        '<th></th>'.
        '<th>Actions</th>'.
        '</tr>'.
        '</thead>';
    while($row = $result->fetch_assoc()){
        echo
        '<tbody>'.
        '<tr>'.
        '<td class="align-middle">'.
        '<div class="d-flex align-items-center">'.
        '<div class="ms-3 lh-1">'.
        '<h5 class="mb-1">'. $row['invoiceId'] . '</h5>'.
        '<p class="mb-0">'. $row['Date'] . '</p>'.
        '</div>'.
        '</div>'.
        '</td>'.
        '<td class="align-middle">'. $row['cashierName'] . '</td>'.
        '<td class="align-middle"><text>RM</text>.number_format( $row['totalPrice'],2) . '</td>'.
        '<td class="align-middle"></td>'.
        '<td class="align-middle">'.
    
```

Figure 15: Code Segment for Check Out Record Interface

Figure 16 shown the supplier stock-in interface. All record create by user will be display in this interface. The user able to edit the stock-in details by pressing the action button. The code segment for supplier stock-in interface is shown in Figure 17.

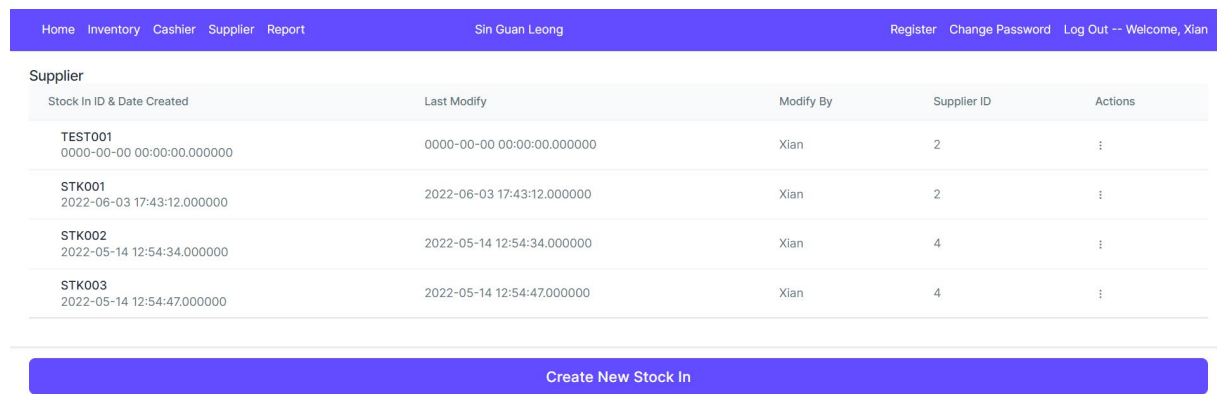


Figure 16: Supplier Stock-In Interface

```
<?php
$sql = "SELECT checkInReportID, stockInId, stockInDate, supplierId, date, username FROM checkinreport";
$result = $con->query($sql);
if ($result->num_rows > 0) {
    // output data of each row
    echo '<div class="table-responsive">'.
        '<table class="table text-nowrap" id="myTable">'.
            '<thead class="table-light">'.
                '<tr>'.
                    '<th>Stock In ID & Date Created</th>'.
                    '<th>Last Modify</th>'.
                    '<th>Modify By</th>'.
                    '<th>Supplier ID</th>'.
                    '<th>Actions</th>'.
                '</tr>'.
            '</thead>';
    while($row = $result->fetch_assoc()){
        echo
            '<tbody>'.
            '<tr>'.
                '<td class="align-middle">'.
                    '<div class="d-flex align-items-center">'.
                        '<div class="ms-3 lh-1">'.
                            '<h5 class="mb-1">'. $row['stockInId'] . '</h5>'.
                            '<p class="mb-0">'. $row['date'] . '</p>'.
                        '</div>'.
                    '</div>'.
                '</td>'.
                '<td class="align-middle">'. $row['date'] . '</td>'.
                '<td class="align-middle">'. $row['username'] . '</td>'.
                '<td class="align-middle">'. $row['supplierId'] . '</td>'.
                '<td class="align-middle">'.
                    '<div class="dropdown dropdown">'.
                        '<a class="text-muted text-primary-hover" href="#" role="button" id="dropdownTeamOne" data-bs-toggle="dropdown" aria-has
                            '<i class="icon-xxs" data-feather="more-vertical"></i>'.
                    '</a>'.
            '</tr>'.
        '</tbody>'.
    '</table>';
}
```

Figure 17: Code Segment for Supplier Stock-In Interface

In generate report module, the user able to create and generate a report for the sales. The report will be generated according the duration enter by user. The user can select the start and end date to generate the report in the range selected. Also, the report created can be print by user. The generate

report interface is shown in Figure 18. The user can print the report by pressing the print button. The code segment for report list interface is shown in Figure 19.

Product Code	Quantity	Average Sell Price	Total Price
1004	2098	RM 1.19	RM 2,498.00
1002	200	RM 2.50	RM 500.00
1111	100	RM 5.00	RM 500.00
9999	100	RM 1.00	RM 100.00
4897059680231	10	RM 5.00	RM 50.00
Total	2508		RM 3,648.00

Figure 18: Generate Report Interface

```

<?php
$sql = "SELECT * FROM report where reportId='$_REQUEST[reportId]'";
$result = $con->query($sql);
$totalPrice = 0;
$totalQty = 0;
if ($result->num_rows > 0) {
    while($row = $result->fetch_assoc()) {
        $averageSellPrice=$row['productPrice']/$row['productQty'];
        $averageSellPrice=number_format(round($averageSellPrice,2),2);
        $totalPrice=$totalPrice + $row['productPrice'];
        $totalQty= $totalQty + $row['productQty'];
        echo '<tr>'.
        '<th scope="row">'.$row['productCode'].'</th>'.
        '<td>'.$row['productQty'].'</td>'.
        '<td><text>RM</text>'.$averageSellPrice.'</td>'.
        '<td><text>RM</text>'.number_format($row['productPrice'],2).'</td>'.
        '</tr>';
    }
}
?>
<tr class="table-secondary">
<th>Total</th>
<td><?php echo $totalQty;?></td>
<td></td>
<td>RM<?php echo number_format($totalPrice,2);?></td>
</tr>
</tbody>
</table>
</body>
<button class="w-100 btn btn-lg btn-primary" onclick="window.print();return false;">Print</button>
</html>

```

Figure 19: Code Segment for Generate Report Interface

4.2 Testing

There will be two types of testing conducted which are the functional requirement testing and the user acceptance testing. Functional Testing is to validate the system against the functional requirements and specifications. The user acceptance testing is done with the expected user by using Google form. The test cases were created based on the Requirement Traceability Matrix (RTM). The RTM is shown in Table 6. This test case can help to trace the requirements of the system and make sure all requirements are being tested and passed.

Table 6: Requirement Traceability Matrix

Requirement	Test Case ID	Description	Test Result (Pass / Fail)
SRS_REQ_100	TEST_100	Login Module	
SRS_REQ_101	TEST_100_001	The system should allow users to log in to the system by providing a registered and valid username and password	Pass
SRS_REQ_102	TEST_100_002	The system should redirect the user to the respective page after users successfully login or change password.	Pass
SRS_REQ_103	TEST_100_003	The system should display an error message or a feedback to the users if login failed or change password failed	Pass
SRS_REQ_104	TEST_100_004	The system should allow users to log out from the system.	Pass
SRS_REQ_105	TEST_100_005	The system should allow users (Administrator) to register a new users	Pass
SRS_REQ_200	TEST_200	Manage Inventory	Test Result (Pass / Fail)
SRS_REQ_201	TEST_200_001	The system should allow the user to scan a barcode using a barcode scanner as input into the system.	Pass
SRS_REQ_202	TEST_200_002	The system should allow the user to create and modify product information into system.	Pass
SRS_REQ_203	TEST_200_003	The system should allow user to preview all product information.	Pass
SRS_REQ_204	TEST_200_004	The system should recognize the product information by using the barcode number input by user.	Pass
SRS_REQ_205	TEST_200_005	The system should allow users to delete the product from the system	Pass
SRS_REQ_206	TEST_200_006	The system should allow users to filter the product according the product categories.	Pass
SRS_REQ_207	TEST_200_007	The system should allow user to set the reminder to remind user when the product is running low amount.	Pass

Table 6: (cont)

SRS_REQ_300	TEST_300	Cashier Module	
SRS_REQ_301	TEST_300_001	The system should allow users to create check out record for the product.	Pass
SRS_REQ_302	TEST_300_002	The system should allow users to create refund record for a product.	Pass
SRS_REQ_303	TEST_300_003	The system should keep the record information for the check out and refund progress.	Pass
SRS_REQ_304	TEST_300_004	The system should allow users to view the check out and refund record.	Pass
SRS_REQ_400	TEST_400	Supplier Module	Test Result (Pass / Fail)
SRS_REQ_401	TEST_400_001	The system should allow users to create stock in record for the product.	Pass
SRS_REQ_402	TEST_400_002	The system should allow users to create new supplier.	Pass
SRS_REQ_403	TEST_400_003	The system should allow users to modify the stock in record.	Pass
SRS_REQ_404	TEST_400_004	The system should allow users to delete the stock in record.	Pass
SRS_REQ_405	TEST_400_005	The system should allow users to view the stock in record.	Pass
SRS_REQ_406	TEST_400_006	The system should allow users to modify the supplier information.	Pass
SRS_REQ_407	TEST_400_007	The system should keep the process history record for the stock in.	Pass
SRS_REQ_500	TEST_500	Generate Report	
SRS_REQ_501	TEST_500_001	The system should allow users to generate the report on the daily or monthly check out data.	Pass
SRS_REQ_502	TEST_500_002	The system should allow users to print the generated report	Pass

The overall test cases for the functional requirement is shown in Table 7. There are total of 25 test cases are been test according each module respectively and the test results are stated in the table.

Table 7: Overall Test Case Result

Test Case ID	Total Test Cases	Total Passed
TEST_100	5	5
TEST_200	7	7
TEST_300	4	4
TEST_400	7	7
TEST_500	2	2
	25	25

The user acceptance testing is done with the expected user by using Google form. The users have been provided 3 aspects which are functionality of the system, usability of the system and the user interface design of the system. Users can rate those aspects by the scale 1 to 5 which represents very unsatisfied to very satisfied. There are 3 people who test the system. One of the users tested the system as Administrator and two of the users tested the system as a worker. Figure 17 shows the overall user satisfaction on each aspect of the proposed system.



Figure 17: Overall User Satisfaction on Proposed System

From Figure 5.31, all of the three users are very satisfied with the functionality of the system. This shows that the functional requirements meet the needs of the grocery store and help improve the management of inventory. For usability, two users are very stratified with the system usability, this shows that the system is easy to use and user-friendly. The user interface design is very stratified from two users, it shows that the user interface is simple, consistent and provides good informative feedback.

5. Conclusion

The inventory system for the Sin Guan Leong Grocery Store was successfully developed. All of the phases determined in methodology are successfully done in the expected schedule. The system meets all the requirements and objectives. The user acceptance testing and functional testing are passed. The purpose to develop the system is to provide a complete systematic and reliable system with less possibility of any errors. It also makes the management for the grocer more easy and efficient. Even the system has some limitations that can't be achieved such as the system is not available when there is no internet access but the suggestions are provided for further improvement.

Acknowledgment

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

References

- [1] Priniotakis, G., & Argyropoulos, P. (2018, December). Inventory management concepts and techniques. In *IOP conference series: Materials Science and Engineering* (Vol. 459, No. 1, p. 012060). IOP Publishing.
- [2] Chandra Sekhar. (2021, July). Inventory Management for Emart Grocery Shop. *International Journal of All Research Education and Scientific Methods (IJARESM)* (Volume 9)
- [3] PALLAS SUTANTO, A. D. O. N. I. S. (2019). Web Based Inventory Management System in LotteMart Solo Baru. *Web Based Inventory Management System in LotteMart Solo Baru*, 1, 2–4.
- [4] AYELE, D. N. M. M. G. (2015). *Supermarket Management System Using Bar code Technology* (Doctoral dissertation).
- [5] Britannica, T. Editors of Encyclopaedia (2019, October 24). barcode. Encyclopedia Britannica. <https://www.britannica.com/technology/barcode>
- [6] Bidgoli, H. (2003). *Encyclopedia of Information Systems: E-J [E-book]*. Amsterdam University Press.
- [7] UML Use-Case Diagrams. (2005). *The Elements of UMLTM 2.0 Style*, 33–46. <https://doi.org/10.1017/cbo9780511817533.005>
- [8] Dennis, A., Wixom, B., & Tegarden, D. (2015). *Systems Analysis and Design: An Object-Oriented Approach with UML* (5th ed.). Wiley.
- [9] Dennis, A. *Systems Analysis and Design* (5th ed.). Wiley. p. 309, 2012
- [10] Coronel, C., Morris, S., & Rob, P. (2009). *Database Systems: Design, Implementation, and Management (Management Information Systems)* (9th ed.). Cengage Learning.
- [11] Galitz, W. O. (2002). *The Essential Guide to User Interface Design: An Introduction to GUI Design Principles and Techniques* (2nd ed.). Wiley.