



## **Group Assignment – Group 20**

TECHNOLOGY PARK MALAYSIA

CT038-3-2-ODJ

Object Oriented Development with Java

APU2F2305CS(CYB)/APD2F2305CS(DA)

**HAND OUT DATE: 30 June 2023**

**HAND IN DATE: 10 September 2023**

**WEIGHTAGE: 50%**

---

**LECTURER: Tan Li June**

No	Student Name	TP Number
1.	Soo Jiun Guan	TP068687
2.	Tan Wee Wen	TP069472

## Table of Contents

1.0 Introduction.....	7
2.0 Limitation of The System .....	8
3.0 UML Diagram.....	9
3.1 Use Case Diagram .....	9
3.1.1 Use Case Specification .....	10
3.2 Class Diagram .....	39
4.0 Sample Output.....	43
4.1 Login Page .....	43
4.2 Sales Manager .....	44
4.2.1 Sales Manager Menu .....	45
4.2.2 Item.....	47
4.2.2.1 Add Item.....	48
4.2.2.1.1 Add new item.....	48
4.2.2.1.2 Add Stock of existing Item .....	49
4.2.2.2 Edit Item .....	51
4.2.2.2.1 Edit name .....	52
4.2.2.2.2 Edit Stock.....	53
4.2.2.2.3 Edit Supplier Code .....	55
4.2.2.2.4 Edit Category .....	56
4.2.2.3 Delete Item .....	57
4.2.2.4 View Item .....	59
4.2.2.5 Search Item.....	60
4.2.2.6 Exit Item Menu .....	61
4.2.3 Supplier .....	62
4.2.3.1 Add Supplier.....	63
4.2.3.2 Edit Supplier .....	66

4.2.3.2.1 Edit Supplier Name .....	67
4.2.3.2.2 Edit Supplier Phone Number .....	68
4.2.3.2.3 Edit Supplier Email Address.....	69
4.2.3.2.4 Edit Supplier Address.....	70
4.2.3.3 Delete Supplier .....	71
4.2.3.4 View Supplier .....	73
4.2.3.5 Search Supplier.....	74
4.2.3.6 Exit Supplier Menu .....	75
4.2.4 Daily Item-wise Sales Entry.....	76
4.2.4.1 Generate Daily Sales .....	77
4.2.4.2: Edit Daily sales .....	80
4.2.4.2.1: Modify Sales Item .....	81
4.2.4.2.2: Remove Sales Item.....	83
4.2.4.2.3: Add New Item.....	85
4.2.4.3: View Daily Sales .....	87
4.2.4.4: Search Daily Sales .....	88
4.2.4.5 Delete Daily Sales.....	89
4.2.4.6: Exit Daily Item-wise Sales Menu .....	91
4.2.5 Purchase Requisition.....	92
4.2.5.1 Add Purchase Requisition .....	93
4.2.5.2: Edit Purchase Requisition .....	95
4.2.5.2.1 Edit Item Code .....	96
4.2.5.2.2 Edit Quantity .....	98
4.2.5.2.3 Edit Require Date.....	100
4.2.5.2.4 Edit Approved or Rejected PR.....	102
4.2.5.2.5 Edit Expired Purchase Requisition.....	104
4.2.5.3 Delete Purchase Requisition.....	106

4.2.5.4 View Purchase Requisition.....	108
4.2.5.4.1 Expired Status .....	109
4.2.5.5 Search Purchase Requisition .....	110
4.2.5.6 Exit Purchase Requisition .....	111
4.2.6 List of Purchase Order .....	112
4.2.7 Update Info.....	114
4.2.7.1 Edit Name .....	115
4.2.7.2 Edit Phone Number .....	116
4.2.7.3 Edit Email Address .....	118
4.2.7.4 Edit Password .....	120
4.2.8 View Info .....	123
4.2.9 Log Out .....	124
4.3 Purchase Manager .....	125
4.3.1 Purchase Manager Menu .....	125
4.3.2 List of Items .....	126
4.3.3 List of Suppliers .....	126
4.3.4 Display Purchase Requisition.....	126
4.3.5 Purchase Order .....	127
4.3.5.1 Create Purchase Order.....	128
4.3.5.2 Edit Purchase Order .....	131
4.3.5.2.1 Edit Item Code and Quantity .....	132
4.3.5.2.2 Edit Quantity .....	133
4.3.5.2.3 Edit Required Date .....	134
4.3.5.3 Delete Purchase Order.....	135
4.3.5.4 View Purchase Order .....	136
4.3.5.5 Search Purchase Order .....	136
4.3.6 List of Purchase Order .....	137

4.3.7 Update Info.....	137
4.3.8 View Own Info.....	137
4.3.9 Log Out .....	137
4.4 Admin .....	138
4.4.1. Admin Menu.....	138
4.4.2. Item Entry.....	139
4.4.3. Supplier Entry.....	139
4.4.4. Daily Item-wise Sales Entry.....	139
4.4.5. Purchase Requisition.....	139
4.4.6 Purchase Order Entry.....	139
4.4.7. Register New Staff.....	140
4.4.8. Update Staff's Info .....	141
4.4.9. Delete Existing Staff.....	141
4.4.10 Search Existing Staff.....	142
4.4.11 Log Out .....	143
4.5 Validation.....	144
4.5.1 User Authentication .....	144
4.5.2 Input Validation.....	145
4.5.3 Overstock Validation .....	147
4.5.4 Existence Validation.....	148
4.5.5: Choice Validation.....	149
5.0 Object Oriented Concept .....	150
5.1 Classes .....	150
5.2 Encapsulation .....	151
5.3 Inheritance.....	153
5.4 Polymorphism .....	154
5.5 Abstraction.....	158

6.0 Extra Feature .....	159
6.1 isStockLessThan5() .....	159
6.2 PrNeedToApprove() .....	160
6.3 refreshPRstatus() .....	162
6.4 UpdateInfo().....	163
6.5 deleteStaff().....	165
6.6 Search().....	166
6.7 Comment.....	168
7.0 Conclusion .....	170
8.0 References.....	171

## 1.0 Introduction

In today's fast-paced business world, it's essential for any company to handle its resources well in order to succeed. This is SIGMA SDN BHD (SSB), a growing wholesale distributor in Johor Bahru, Malaysia. They focus on selling everyday groceries, fresh fruit, and fresh foods. It's important for SSB to streamline and automate important processes like Purchase Orders (POs) to improve operating efficiency and accuracy as it continues to grow and serve more retailers across Malaysia.

Object-Oriented Programming (OOP) techniques will be used to make a strong Java application that meets SSB's unique needs as part of this coursework project. Through the use of this software, SSB hopes to achieve a number of important features that will help them better handle their purchasing and selling processes.

Major components to the project are secure user authentication, user registration with position task, managing items and suppliers, keeping track of daily sales operations, creating and showing purchase requests, creating and listing purchase orders.

We will talk more about the design and implementation of this Java app in the parts that follow. These sections will show how OOP principles are used to make a well-structured and effective system for SSB's business needs.

## 2.0 Limitation of The System

1. **Data Storage Limitation:** The system can only store a maximum of 999 records for each data type.
2. **Required Existing IDs:** When entering an ID, it must already exist in the system.
3. **Daily Sales Date Restriction:** Daily sales entries cannot be added for dates in the past.
4. **Non-Editable Purchase Requisition Status:** The status of a Purchase Requisition cannot be edited if it was expired.
5. **Purchase Requisition Status Update:** To update the Purchase Requisition status (only check if require date is overdate), the program needs to be restarted.
6. **Non-Interruptible Operations:** Once an operation is selected for processing, it cannot be halted mid-process. All required data must be filled out, and if data is no longer needed, it must be manually deleted.



## 3.0 UML Diagram

### 3.1 Use Case Diagram

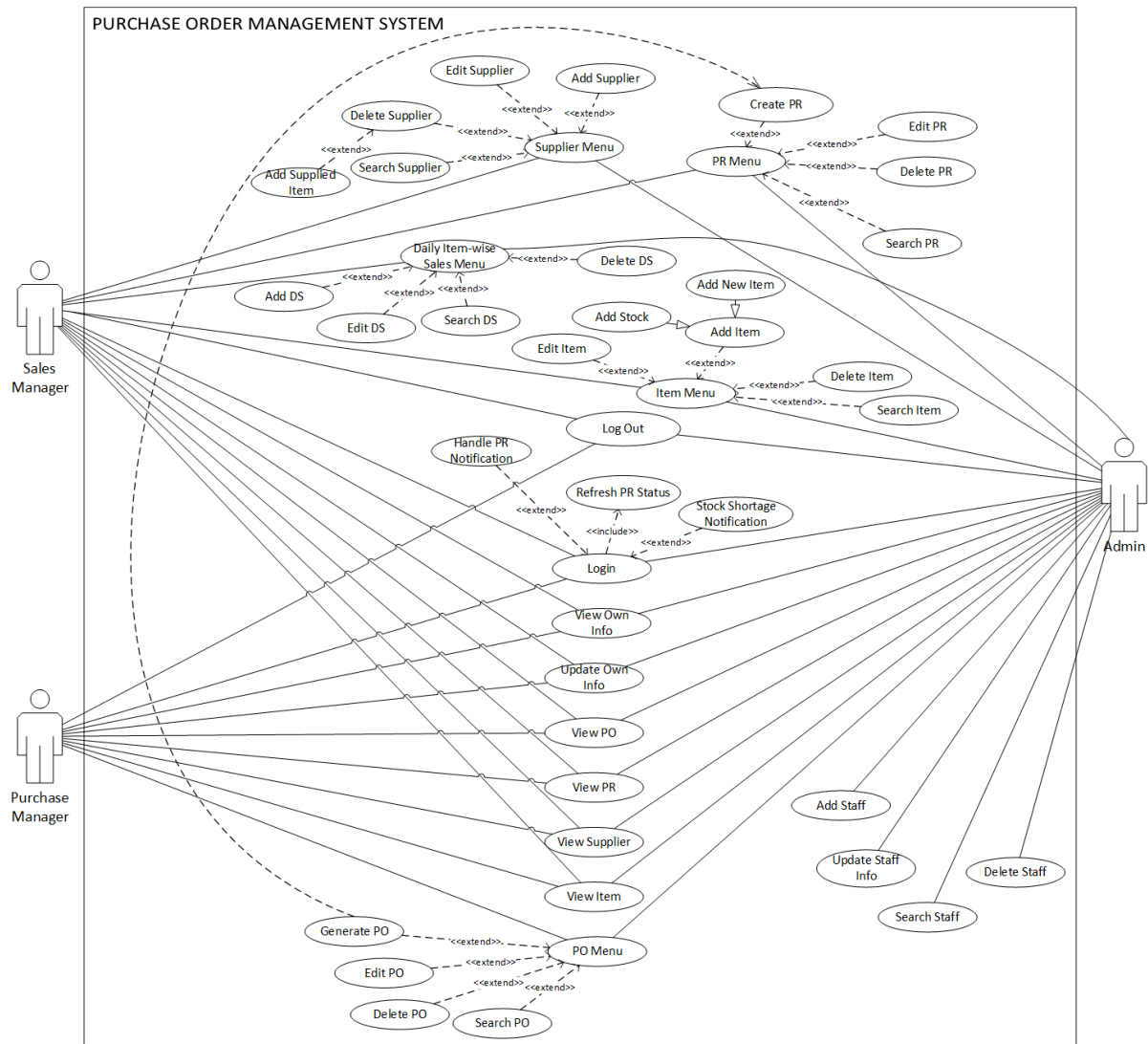


Figure 1: Use Case Diagram

### 3.1.1 Use Case Specification

Use Case	Login
Briefing Description	This use case allows the actor to access the functionality of the system.
Actors	Admin, Sales Manager, Purchase Manager
Preconditions	The actor's information needs to be registered by the admin in advance.
Main Flows	<ol style="list-style-type: none"><li>1. The use case begins when the actor opens the programs or logs out the system.</li><li>2. The system requests the actor to enter their username and password.</li><li>3. The system will validate the input username and password by searching in the file.</li><li>4. The system will give the actor different access to the system according to the role of the actor.</li></ol>
Alternative Flows	<ol style="list-style-type: none"><li>1. The actor doesn't remember the username or password and needs to contact admin to change it.</li><li>2. The system will perform validation using the file, and if it fails to validate the username and password, actors should consider checking for potential typographical errors in their username or password.</li></ol>

Use Case	Refresh PR Status
Briefing Description	This use case refreshes the status of the purchase requisition. For instance, the status of the purchase requisition will become “Expired” if its required date exceeds today's date.
Actors	Admin, Sales Manager, Purchase Manager
Preconditions	The program should be opened.
Main Flows	<ol style="list-style-type: none"> <li>1. The use case starts passively and checks today's date.</li> <li>2. The system will check whether any date of the purchase requisitions is expired.</li> <li>3. The status of the outdated purchase requisition will be changed to “Expired”.</li> </ol>
Alternative Flows	<ol style="list-style-type: none"> <li>1. If there is not any outdated purchase requisition in the file, it wouldn't change the content of the file.</li> </ol>

Use Case	Handle PR Notification
Briefing Description	This use case shows the notification in order to notify purchase manager in handling the purchase requisition.
Actors	Purchase Manager
Preconditions	<p>The system must log in by the purchase manager role.</p> <p>There is any purchase requisition need to be handled.</p>
Main Flows	<ol style="list-style-type: none"> <li>1. The use case begins after the sales manager login.</li> <li>2. The system checks whether purchase requisition that hasn't been processed yet exists in the file.</li> <li>3. It will show a notification that contains how much purchase requisition haven't been handled after the purchase manager log in.</li> </ol>
Alternative Flows	<ol style="list-style-type: none"> <li>1. If there is not any purchase requisition need to be process, it wouldn't show anything.</li> </ol>

Use Case	Stock Shortage Notification
Briefing Description	This use case shows the stock shortage notification of the specific item in order to prompt the sales manager to purchase more stock.
Actors	Sales Manager
Preconditions	The system must log in by the sales manager role. There is any stock of item less than 5.
Main Flows	<ol style="list-style-type: none"> <li>1. The use case begins after the actor login with sales manager role.</li> <li>2. The system checks whether items are in low stock.</li> <li>3. It will show a notification that contains the items code that its stock is less than 5.</li> </ol>
Alternative Flows	<ol style="list-style-type: none"> <li>1. If there is not any insufficient inventory in the file, it won't show anything.</li> </ol>

Use Case	View Own Info
Briefing Description	This use case allows actors to view their own basic information.
Actors	Admin, Sales Manager, Purchase Manager
Preconditions	The actor is logged into their user account.
Main Flows	<ol style="list-style-type: none"> <li>1. The use case starts when the actor chooses to view their own information.</li> <li>2. The system will retrieve their information in the file by using their username.</li> <li>3. The system presents their basic information in a user-friendly format.</li> </ol>
Alternative Flows	<ol style="list-style-type: none"> <li>1. The actor should click on enter key to proceed to the next function.</li> </ol>

Use Case	Update Own Info
Briefing Description	This use case allows sales manager and purchase manager to update their own basic information, excluding certain information like username and role, which is not accessible to regular staff. However, the admin is able to edit the entire information of himself/herself.
Actors	Admin, Sales Manager, Purchase Manager
Preconditions	The user is logged into the system with the appropriate role.
Main Flows	<ol style="list-style-type: none"> <li>1. The use case starts when the actor chooses to update the account information.</li> <li>2. The system will retrieve their information in the file by using their username.</li> <li>3. The system asks them to choose which part of the information they need to update and then requires them to enter or choose the changed content.</li> <li>4. The changed information will be changed to the file.</li> </ol>
Alternative Flows	<ol style="list-style-type: none"> <li>1. If the actor chooses or enter invalid option, the system will ask him/her to choose again.</li> </ol>

Use Case	View PO
Briefing Description	This use case allows the actor to view the list of purchase orders.
Actors	Admin, Sales Manager, Purchase Manager
Preconditions	The actor needs to login the system.
Main Flows	<ol style="list-style-type: none"> <li>1. The use case begins when the actor chooses to view the list of purchase orders.</li> </ol>

	2. The system will retrieve the valid purchase order in the file and display them to the actor.
Alternative Flows	1. If there is no purchase order in the file, it will print out a message than notify the actor there is no result found in the file.

Use Case	View PR
Briefing Description	This use case allows the actor to view the list of purchase requisitions.
Actors	Admin, Sales Manager, Purchase Manager
Preconditions	The actor needs to login the system.
Main Flows	<ol style="list-style-type: none"> <li>1. The use case begins when the actor chooses to view the list of purchase requisitions.</li> <li>2. The system will retrieve the valid purchase requisition in the file and display them to the actor.</li> </ol>
Alternative Flows	1. If there is no purchase requisition in the file, it will print out a message than notify the actor there is no result found in the file.

Use Case	View Supplier
Briefing Description	This use case allows the actor to view the list of suppliers.
Actors	Admin, Sales Manager, Purchase Manager
Preconditions	The actor needs to login the system.
Main Flows	<ol style="list-style-type: none"> <li>1. The use case begins when the actor chooses to view the list of the suppliers.</li> <li>2. The system will retrieve the valid supplier in the file and display them to the actor.</li> </ol>

Alternative Flows	<ol style="list-style-type: none"> <li>1. If there is no supplier in the file, it will print out a message than notify the actor there is no result found in the file.</li> </ol>
-------------------	---

Use Case	View Item
Briefing Description	This use case allows the actor to view the list of items.
Actors	Admin, Sales Manager, Purchase Manager
Preconditions	The actor needs to login the system.
Main Flows	<ol style="list-style-type: none"> <li>1. The use case begins when the actor chooses to view the list of the items.</li> <li>2. The system will retrieve the valid item in the file and display them to the actor.</li> </ol>
Alternative Flows	<ol style="list-style-type: none"> <li>1. If there is no item in the file, it will print out a message than notify the actor there is no result found in the file.</li> </ol>

Use Case	Item Menu
Briefing Description	This use case allows the actor to access the item menu functionality within the system. Item menu involves adding, editing, and deleting product information, including details such as item name, stock quantity, supplier code, and category.
Actors	Admin, Sales Manager
Preconditions	The actor is logged into the system with the appropriate role.
Main Flows	<ol style="list-style-type: none"> <li>1. The use case begins when the actor accesses the item menu feature.</li> </ol>

	<ol style="list-style-type: none"> <li>The system displays a menu that contains several actions related to items, such as adding, editing, deleting, and searching items.</li> <li>The system will ask the actor to select which action he/she would like to take.</li> </ol>
Alternative Flows	<ol style="list-style-type: none"> <li>If the invalid option is chosen by the actor, the system detects this and provides an error message.</li> <li>The actor is then asked to choose again from the menu.</li> </ol>

Use Case	Add Item
Briefing Description	This use case allows the actor to add the stock of specific items or add a new item into the file.
Actors	Admin, Sales Manager
Preconditions	The actor is logged into the system with the appropriate role.
Main Flows	<ol style="list-style-type: none"> <li>The use case begins when the actor chooses the add item option.</li> <li>The system presents the actor with two primary options to choose from, which are new item or stock addition.</li> </ol>
Alternative Flows	<ol style="list-style-type: none"> <li>The system will display error message and ask the actor to enter again in case of an inappropriate input option.</li> </ol>

Use Case	Add New Item
Briefing Description	This use case allows the actor to add new item into the file.



Actors	Admin, Sales Manager
Preconditions	The actor is logged into the system with the appropriate role.
Main Flows	<ol style="list-style-type: none"> <li>1. The use case begins when the actor chooses to add a new item rather than add stock.</li> <li>2. The system prompts the actor to input information about the new item.</li> <li>3. The information then will be stored into the file that contains all items of the system.</li> </ol>
Alternative Flows	<ol style="list-style-type: none"> <li>1. If the actor enters a non-numeric input, he/she needs to check his/her input and the system will be asked to re-enter.</li> </ol>

Use Case	Add Stock
Briefing Description	This use case allows the actor to add stock of the existing item into the file.
Actors	Admin, Sales Manager
Preconditions	The actor is logged into the system with the appropriate role. There is any item existing in the file.
Main Flows	<ol style="list-style-type: none"> <li>1. The use case begins when the actor chooses to increase the item's stock.</li> <li>2. The system prompts the actor to input the item code of the item he/she would like to add.</li> <li>3. Then, the system will check the existence of the input item code.</li> <li>4. It asks the actor to enter the quantity they want to add stock to the item.</li> <li>5. The data will subsequently be saved in the file.</li> </ol>
Alternative Flows	<ol style="list-style-type: none"> <li>1. If the actor enters a non-integer input or nonexistent item code, he/she needs to check</li> </ol>

	his/her input and the system will be asked to re-enter.
--	---

Use Case	Edit Item
Briefing Description	This use case allows the actor to edit information about the specific item.
Actors	Admin, Sales Manager
Preconditions	The actor is logged into the system with the appropriate role. There is any item existing in the file.
Main Flows	<ol style="list-style-type: none"> <li>1. The use case begins when the actor selects in editing item.</li> <li>2. The system checks whether any item exists in the file.</li> <li>3. The system requires the actor to enter the item code they wish to edit.</li> <li>4. It will check whether the item code exists or not.</li> <li>5. Then, it will ask the actor to enter the changed content.</li> <li>6. The system will store the new information of the specific item into the file.</li> </ol>
Alternative Flows	<ol style="list-style-type: none"> <li>1. The system will display an error message and ask the actor to enter again if the actor inputs an invalid item code.</li> </ol>

Use Case	Delete Item
Briefing Description	This use case allows the actor to edit delete the specific item.
Actors	Admin, Sales Manager

Preconditions	<p>The actor is logged into the system with the appropriate role.</p> <p>There is any item existing in the file.</p>
Main Flows	<ol style="list-style-type: none"> <li>1. The use case begins when the actor chooses to remove an existing item.</li> <li>2. The system checks whether any item exists in the file.</li> <li>3. The system requires the actor to enter the item code they wish to delete.</li> <li>4. It will check whether the item code exists or not.</li> <li>5. The system will remove the information of the specific item from the file.</li> </ol>
Alternative Flows	<ol style="list-style-type: none"> <li>1. The system will display an error message and ask the actor to enter again if the actor input an invalid item code.</li> </ol>

Use Case	Search Item
Briefing Description	This use case allows the actor to search the specific item.
Actors	Admin, Sales Manager
Preconditions	The actor is logged into the system with the appropriate role.
Main Flows	<ol style="list-style-type: none"> <li>1. The use case begins when the actor chooses to search for an existing item.</li> <li>2. The system requires the actor to enter the item code or name about the item they wish to know.</li> <li>3. The system will obtain the item data corresponding to the input code from the file.</li> <li>4. The system will show the result about the input item code.</li> </ol>

Alternative Flows	1. The system shows notifications that no found anything if the actor inputs an invalid item code.
-------------------	--

Use Case	Supplier Menu
Briefing Description	This use case allows the actor to access the supplier entry functionality within the system. Supplier menu involves adding, editing, and deleting supplier information, including details such as name, phone number, email, and address. In addition, the actor also can store the item supplied by the supplier into the file.
Actors	Admin, Sales Manager
Preconditions	The actor is logged into the system with the appropriate role.
Main Flows	<ol style="list-style-type: none"> <li>1. The use case begins when the actor accesses the supplier menu feature.</li> <li>2. The system displays a menu that contains several actions related to suppliers, such as adding, editing, deleting, and searching suppliers.</li> <li>3. The system will ask the actor to select which action he/she would like to take.</li> </ol>
Alternative Flows	<ol style="list-style-type: none"> <li>1. If the invalid option is chosen by the actor, the system detects this and provides an error message.</li> <li>2. The actor is then asked to choose again from the menu.</li> </ol>

Use Case	Add Supplier
Briefing Description	This use case allows the actor to add new supplier into the file.
Actors	Admin, Sales Manager

Preconditions	The actor is logged into the system with the appropriate role.
Main Flows	<ol style="list-style-type: none"> <li>1. The use case begins when the actor chooses to add new staff.</li> <li>2. The system prompts the actor to input the name of the new supplier.</li> <li>3. The system will search in the file to check whether the name exists in the file or not.</li> <li>4. Then, it will prompt the actor to enter more details about the new supplier.</li> <li>5. The all information then will be stored into the file that contains all suppliers of the system.</li> </ol>
Alternative Flows	<ol style="list-style-type: none"> <li>1. The system will display an error message and ask the actor to enter again if the actor inputs an invalid supplier code.</li> <li>2. It will ask the actor to enter again.</li> </ol>

Use Case	Edit Supplier
Briefing Description	This use case allows the actor to edit information about the specific supplier.
Actors	Admin, Sales Manager
Preconditions	<p>The actor is logged into the system with the appropriate role.</p> <p>There is any supplier existing in the file.</p>
Main Flows	<ol style="list-style-type: none"> <li>1. The use case begins when the actor selects in editing supplier.</li> <li>2. The system checks whether any supplier exists in the file.</li> <li>3. The system requires the actor to enter the supplier code they wish to edit.</li> </ol>

	<ol style="list-style-type: none"> <li>4. It will check whether the supplier code exists or not.</li> <li>5. Then, it will ask the actor to enter the changed content.</li> <li>6. The system will store the new information of the specific supplier into the file.</li> </ol>
Alternative Flows	<ol style="list-style-type: none"> <li>1. The system will display an error message and ask the actor to enter again if the actor input an invalid supplier code.</li> </ol>

Use Case	Delete Supplier
Briefing Description	This use case allows the actor to edit delete the specific supplier.
Actors	Admin, Sales Manager
Preconditions	<p>The actor is logged into the system with the appropriate role.</p> <p>There is any supplier existing in the file.</p>
Main Flows	<ol style="list-style-type: none"> <li>1. The use case begins when the actor chooses to remove an existing supplier.</li> <li>2. The system checks whether any supplier exists in the file.</li> <li>3. The system requires the actor to enter the supplier code they wish to delete.</li> <li>4. It will check whether the supplier code exists or not.</li> <li>5. The system will remove the information of the specific supplier from the file.</li> </ol>
Alternative Flows	<ol style="list-style-type: none"> <li>1. The system will display an error message and ask the actor to enter again if the actor input an invalid supplier code.</li> </ol>

Use Case	Search Supplier
Briefing Description	This use case allows the actor to search the specific supplier.
Actors	Admin, Sales Manager
Preconditions	The actor is logged into the system with the appropriate role.
Main Flows	<ol style="list-style-type: none"> <li>1. The use case begins when the actor chooses to search for an existing supplier.</li> <li>2. The system requires the actor to enter the supplier code or name of the supplier they wish to know.</li> <li>3. The system will obtain the supplier data corresponding to the input code from the file.</li> <li>4. The system will show the result about the input supplier code.</li> </ol>
Alternative Flows	<ol style="list-style-type: none"> <li>1. The system shows notifications that no found anything if the actor inputs an invalid supplier code.</li> </ol>

Use Case	Daily Item-wise Sales Menu
Briefing Description	This use case allows the actor to access the daily item-wise sales menu functionality within the system. Daily item-wise sales menu involves adding, editing, and deleting daily item-wise sales information, including details such as daily item-wise sales code, item name, sales quantity, and the date of the day.
Actors	Admin, Sales Manager
Preconditions	The actor is logged into the system with the appropriate role.
Main Flows	<ol style="list-style-type: none"> <li>1. The use case begins when the actor accesses the daily item-wise sales menu feature.</li> </ol>

	<ol style="list-style-type: none"> <li>2. The system displays a menu that contains several actions related to daily item-wise sales, such as adding, editing, deleting, and searching daily item-wise sales.</li> <li>3. The system will ask the actor to select which action he/she would like to take.</li> </ol>
Alternative Flows	<ol style="list-style-type: none"> <li>1. If the invalid option is chosen by the actor, the system detects this and provides an error message.</li> <li>2. The actor is then asked to choose again from the menu.</li> </ol>

Use Case	Add DS
Briefing Description	This use case allows the actor to add daily item-wise sales of the day into the file.
Actors	Admin, Sales Manager
Preconditions	The actor is logged into the system with the appropriate role.
Main Flows	<ol style="list-style-type: none"> <li>1. The use case begins when the actor chooses to add a new daily item-wise sales.</li> <li>2. The system will initially verify today's date and determine whether an entry for today exists or not.</li> <li>3. If not, the system will ask the actor to input the code of the items that were sold on that day.</li> <li>4. The system also performs validation to ensure that the input quantity does not exceed the available stock.</li> <li>5. The system saves the daily item-wise sales data in the file.</li> <li>6. It will ask the actor want to add another item to the same daily item-wise sales code or not.</li> </ol>



Alternative Flows	<ol style="list-style-type: none"> <li>1. The system shows notifications that no found anything if the actor inputs an invalid item code.</li> <li>2. The actor is then asked to choose again from the menu.</li> <li>3. If an entry for today's daily item-wise sales already exists, a notification message will be displayed, informing the user to access the edit function if they wish to make additional entries for today's sales.</li> </ol>
-------------------	---

Use Case	Edit DS
Briefing Description	This use case allows the actor to edit the existing daily item-wise sales.
Actors	Admin, Sales Manager
Preconditions	The actor is logged into the system with the appropriate role. There is entry for daily item-wise sales existing in the file.
Main Flows	<ol style="list-style-type: none"> <li>1. The use case begins when the actor chooses to edit the existing daily item-wise sales.</li> <li>2. The system checks whether any daily item-wise sales exist in the file.</li> <li>3. The system requires the actor to enter the daily item-wise sales code they wish to edit.</li> <li>4. It will check whether the daily item-wise sales code exists or not.</li> <li>5. The system will ask the actor to select which action he/she would like to take, including adding item, modifying, or removing the selected item.</li> <li>6. The system will validate whether the input quantity exceeds the available stock of the item or not.</li> </ol>

	7. The system will store the new information of the daily item-wise sales into the file.
Alternative Flows	<p>The system shows an error message notification if the actor inputs a nonexistent daily item-wise sales code and asks the actor to input again.</p> <p>If the actor selects an invalid option or enters a quantity exceeding the available stock, the system will request the user to re-enter the information.</p>

Use Case	Delete DS
Briefing Description	This use case allows the actor to delete the specific daily item-wise sales data.
Actors	Admin, Sales Manager
Preconditions	<p>The actor is logged into the system with the appropriate role.</p> <p>There is any daily item-wise sales data existing in the file.</p>
Main Flows	<ol style="list-style-type: none"> <li>1. The use case begins when the actor chooses to remove an existing daily item-wise sales.</li> <li>2. The system checks whether any purchase item-wise sales exist in the file.</li> <li>3. The system requires the actor to enter the daily item-wise sales code they wish to delete.</li> <li>4. It will check whether the daily item-wise sales data exists or not.</li> <li>5. The system will remove the information of the entered daily item-wise sales code from the file.</li> </ol>
Alternative Flows	<ol style="list-style-type: none"> <li>2. The system will display an error message and ask the actor to enter again if the actor input an invalid daily item-wise sales code.</li> </ol>

Use Case	Search DS
Briefing Description	This use case allows the actor to search the specific daily item-wise sales data.
Actors	Admin, Sales Manager
Preconditions	The actor is logged into the system with the appropriate role.
Main Flows	<ol style="list-style-type: none"> <li>1. The use case begins when the actor chooses to search for an existing daily item-wise sales data.</li> <li>2. The system requires the actor to enter the code of the daily item-wise sales data they wish to know.</li> <li>3. The system will obtain the item-wise sales data corresponding to the input code from the file.</li> <li>4. The system will show the result about the input daily item-wise sales code.</li> </ol>
Alternative Flows	<ol style="list-style-type: none"> <li>1. The system shows notifications that no found anything if the actor inputs an invalid daily item-wise sales code.</li> </ol>

Use Case	PR Menu
Briefing Description	<p>This use case allows the actor to access the purchase requisition menu functionality within the system.</p> <p>Purchase requisition menu involves adding, editing, and deleting purchase requisition information, including details such as purchase requisition code, item code, quantity, required date, supplier code of the required item, and the applied username.</p>
Actors	Admin, Sales Manager
Preconditions	The actor is logged into the system with the appropriate role.

Main Flows	<ol style="list-style-type: none"> <li>1. The use case begins when the actor accesses the purchase requisition menu feature.</li> <li>2. The system displays a menu that contains several actions related to purchase requisition, such as adding, editing, deleting, and searching purchase requisition.</li> <li>3. The system will ask the actor to select which action he/she would like to take.</li> </ol>
Alternative Flows	<ol style="list-style-type: none"> <li>1. If the invalid option is chosen by the actor, the system detects this and provides an error message.</li> <li>2. The actor is then asked to choose again from the menu.</li> </ol>

Use Case	Add PR
Briefing Description	This use case allows the actor to create a purchase requisition.
Actors	Admin, Sales Manager
Preconditions	The actor is logged into the system with the appropriate role.
Main Flows	<ol style="list-style-type: none"> <li>1. The use case begins when the actor chooses to create a purchase requisition.</li> <li>2. The system prompts the actor to input information about the purchase requisition.</li> <li>3. The system will search in the file to check whether the item code exists in the file or not.</li> <li>4. Upon confirming the existence of the item code, the system retrieves the corresponding supplier code associated with the entered item code.</li> <li>5. The status of the purchase requisition will be changed to “Pending” in order to let the purchase manager or admin approve it.</li> </ol>

	<ol style="list-style-type: none"> <li>6. The all information will then be stored into the file that contains all purchase requisition of the system.</li> <li>7. It will ask the actor want to add another item to the same purchase requisition code or not.</li> </ol>
Alternative Flows	<ol style="list-style-type: none"> <li>1. The system shows notifications that no found anything if the actor inputs an invalid item code.</li> <li>2. The actor is then asked to choose again from the menu.</li> <li>3. If the actor enters a non-integer input or unformatted date, he/she needs to check his/her input and the system will be asked to re-enter.</li> </ol>

Use Case	Edit PR
Briefing Description	This use case allows the actor to edit the existing purchase requisition. If the purchase requisition has been approved, it will change its status to “Pending” after the actor modify it.
Actors	Admin, Sales Manager
Preconditions	The actor is logged into the system with the appropriate role. There is any purchase requisition existing in the file.
Main Flows	<ol style="list-style-type: none"> <li>1. The use case begins when the actor chooses to edit the existing purchase requisition.</li> <li>2. The system checks whether any purchase requisition exists in the file.</li> <li>3. The system requires the actor to enter the purchase requisition they wish to edit.</li> <li>4. It will check whether the purchase requisition code exists or not.</li> <li>5. The system will ask the actor to choose which part of the purchase requisition they would like to edit.</li> </ol>

	<ol style="list-style-type: none"> <li>6. It will require the actor to enter the changed content.</li> <li>7. The system will check whether the input item code exists or not and the format of the date.</li> <li>8. The</li> <li>9. The system will store the new information of the purchase requisition into the file.</li> </ol>
Alternative Flows	<ul style="list-style-type: none"> <li>. The system shows an error message notification if the actor inputs a nonexistent purchase requisition code and asks the actor to input again.</li> <li>. If the actor selects an invalid option or enters the date earlier than the day, the system will request the actor to re-enter the information.</li> </ul>

Use Case	Delete PR
Briefing Description	This use case allows the actor to edit delete the specific purchase requisition.
Actors	Admin, Sales Manager
Preconditions	<p>The actor is logged into the system with the appropriate role.</p> <p>There is any purchase requisition existing in the file.</p>
Main Flows	<ol style="list-style-type: none"> <li>1. The use case begins when the actor chooses to remove an existing purchase requisition.</li> <li>2. The system checks whether any purchase requisition exists in the file.</li> <li>3. The system requires the actor to enter the purchase requisition code they wish to delete.</li> <li>4. It will check whether the purchase requisition code exists or not.</li> <li>5. The system will remove the information of the entered purchase requisition code from the file.</li> </ol>

Alternative Flows	<ol style="list-style-type: none"> <li>1. The system will display an error message and ask the actor to enter again if the actor input an invalid purchase requisition code.</li> </ol>
-------------------	---

Use Case	Search PR
Briefing Description	This use case allows the actor to search the specific purchase requisition
Actors	Admin, Sales Manager
Preconditions	The actor is logged into the system with the appropriate role.
Main Flows	<ol style="list-style-type: none"> <li>1. The use case begins when the actor chooses to search for an existing purchase requisition.</li> <li>2. The system requires the actor to enter the code of the purchase requisition they wish to know.</li> <li>3. The system will obtain the information corresponding to the input code from the file.</li> <li>4. The system will show the result about the input purchase requisition code.</li> </ol>
Alternative Flows	<ol style="list-style-type: none"> <li>1. The system shows notifications that no found anything if the actor inputs a nonexistent purchase requisition code.</li> </ol>

Use Case	PO Menu
Briefing Description	This use case allows the actor to access the purchase order menu functionality within the system. Purchase order menu involves adding, editing, and deleting purchase order information, including details such as purchase order code, purchase requisition code, quantity, required date, supplier code of the required item, and the applied username.

Actors	Admin, Purchase Manager
Preconditions	The actor is logged into the system with the appropriate role.
Main Flows	<ol style="list-style-type: none"> <li>1. The use case begins when the actor accesses the purchase order menu feature.</li> <li>2. The system displays a menu that contains several actions related to purchase order, such as adding, editing, deleting, and searching purchase order.</li> <li>3. The system will ask the actor to select which action he/she would like to take.</li> </ol>
Alternative Flows	<ol style="list-style-type: none"> <li>1. If the invalid option is chosen by the actor, the system detects this and provides an error message.</li> <li>2. The actor is then asked to choose again from the menu.</li> </ol>

Use Case	Add PO
Briefing Description	This use case allows the actor to create a purchase order.
Actors	Admin, Purchase Manager
Preconditions	The actor is logged into the system with the appropriate role. There is any purchase requisition need to be handled.
Main Flows	<ol style="list-style-type: none"> <li>1. The use case begins when the actor chooses to create a purchase order.</li> <li>2. The system will initially check whether any purchase requisition needs to be handled in the file.</li> <li>3. The system will show the list of purchase requisition that are available for approval.</li> <li>4. It asks the actor to either approve or reject the purchase requisition.</li> </ol>



	<ol style="list-style-type: none"> <li>5. If the actor rejects the purchase requisition, it will ask the actor to leave a comment explaining the reason for rejection.</li> <li>6. The status of the purchase requisition will be changed according to the action taken by the actor.</li> <li>7. The all information will then be stored into the file that contains all purchase order of the system if the actor determines to approve the purchase requisition.</li> </ol>
Alternative Flows	<ol style="list-style-type: none"> <li>1. The system will display a message to notify the actor there is not any purchase requisition that needs to be approved.</li> <li>2. If the invalid option is chosen by the actor, the system detects this and provides an error message.</li> <li>3. The actor is then asked to choose again from the menu.</li> </ol>

Use Case	Edit PO
Briefing Description	This use case allows the actor to edit the existing purchase order. It means the actor can modify the purchase order without changing the status of the purchase requisition.
Actors	Admin, Purchase Manager
Preconditions	The actor is logged into the system with the appropriate role. There is any purchase order existing in the file.
Main Flows	<ol style="list-style-type: none"> <li>1. The use case begins when the actor chooses to edit the existing purchase order.</li> <li>2. The system checks whether any purchase order exists in the file.</li> </ol>

	<ol style="list-style-type: none"> <li>3. The system requires the actor to enter the purchase order code they wish to edit.</li> <li>4. It will check whether the purchase order code exists or not.</li> <li>5. The system will ask the actor to choose which part they would like to edit.</li> <li>6. It will require the actor to enter the changed content.</li> <li>7. The system will store the new information of the purchase requisition into the file.</li> </ol>
Alternative Flows	<ol style="list-style-type: none"> <li>1. The system shows an error message notification if the actor inputs a nonexistent purchase order code and asks the actor to input again.</li> <li>2. If the actor selects an invalid option or enters the required date earlier than the day, the system will request the user to re-enter the information.</li> </ol>

Use Case	Delete PO
Briefing Description	This use case allows the actor to delete the specific purchase order and change the status of the purchase requisition to “Pending”.
Actors	Admin, Purchase Manager
Preconditions	<p>The actor is logged into the system with the appropriate role.</p> <p>There is any purchase order existing in the file.</p>
Main Flows	<ol style="list-style-type: none"> <li>1. The use case begins when the actor chooses to remove an existing purchase order.</li> </ol>

	<ol style="list-style-type: none"> <li>2. The system checks whether any purchase order exists in the file.</li> <li>3. The system requires the actor to enter the purchase requisition code they wish to delete.</li> <li>4. It will check whether the purchase order code exists or not.</li> <li>5. The status of the purchase requisition contained within the purchase order is updated to "Pending."</li> <li>6. The system will remove the information of the entered purchase order code from the file.</li> </ol>
Alternative Flows	<ol style="list-style-type: none"> <li>1. The system will display an error message and ask the actor to enter again if the actor inputs an invalid purchase order code.</li> </ol>

Use Case	Search PO
Briefing Description	This use case allows the actor to search the specific purchase order.
Actors	Admin, Purchase Manager
Preconditions	The actor is logged into the system with the appropriate role.
Main Flows	<ol style="list-style-type: none"> <li>1. The use case begins when the actor chooses to search for an existing purchase order.</li> <li>2. The system requires the actor to enter the code of the purchase order they wish to know.</li> <li>3. The system will obtain the information corresponding to the input code from the file.</li> <li>4. The system will show the result about the input purchase order code.</li> </ol>
Alternative Flows	<ol style="list-style-type: none"> <li>1. The system shows notifications that no found anything if the actor inputs a nonexistent purchase order code.</li> </ol>

Use Case	Add Staff
Briefing Description	This use case allows admin to register new staffs.
Actors	Admin
Preconditions	The user must be logged in with the appropriate role.
Main Flows	<ol style="list-style-type: none"> <li>1. The system prompts admin to enter the new staffs' information like username, name, phone number, email address, password and their roles.</li> <li>2. Confirm password should also be entered to ensure the password is correct.</li> <li>3. The system will check whether the input username exists or not.</li> <li>4. The information of the new staff will be stored in the file.</li> </ol>
Alternative Flows	<ol style="list-style-type: none"> <li>1. If the confirm password doesn't match with the password, it will ask admin to input again.</li> </ol>

Use Case	Update Staff Info
Briefing Description	This use case allows admin to modify all information of the existing staffs.
Actors	Admin
Preconditions	The user must be logged in with the appropriate role.
Main Flows	<ol style="list-style-type: none"> <li>1. The use case starts when the admin chooses to update the staff's account information.</li> <li>2. The system prompts admin to enter the username of the staff he/she wishes to update.</li> <li>3. The system will retrieve the information of the input username in the file by using their username.</li> <li>4. The system asks admin to choose which part of the information he/she needs to update and then</li> </ol>

	<p>requires them to enter or choose the changed content.</p> <p>5. The changed information will be changed to the file.</p>
Alternative Flows	<p>1. If the actor chooses or enter invalid option, the system will ask him/her to choose again.</p>

Use Case	Delete Staff
Briefing Description	This use case allows admin to delete the information of the existing staffs.
Actors	Admin
Preconditions	The actor is logged into the system with the appropriate role.
Main Flows	<p>7. The use case begins when the actor chooses to remove an existing staff.</p> <p>8. The system requires the actor to enter the username of the staff they wish to delete.</p> <p>9. It will check whether the staff's username exists or not.</p> <p>10. The system will remove the information of the entered username from the file.</p>
Alternative Flows	<p>2. The system will display an error message and ask the actor to enter again if the actor inputs an invalid staff username.</p>

Use Case	Search Staff
Briefing Description	This use case allows the actor to search the specific staff.
Actors	Admin

Preconditions	The actor is logged into the system with the appropriate role.
Main Flows	<ol style="list-style-type: none"><li>1. The use case begins when the actor chooses to search for an existing staff.</li><li>2. The system requires the actor to enter the username of the staff they wish to know.</li><li>3. The system will obtain the information corresponding to the input code from the file.</li><li>4. The system will show the result about the input username.</li></ol>
Alternative Flows	<ol style="list-style-type: none"><li>1. The system shows notifications that no found anything if the actor inputs a nonexistent username.</li></ol>

### 3.2 Class Diagram

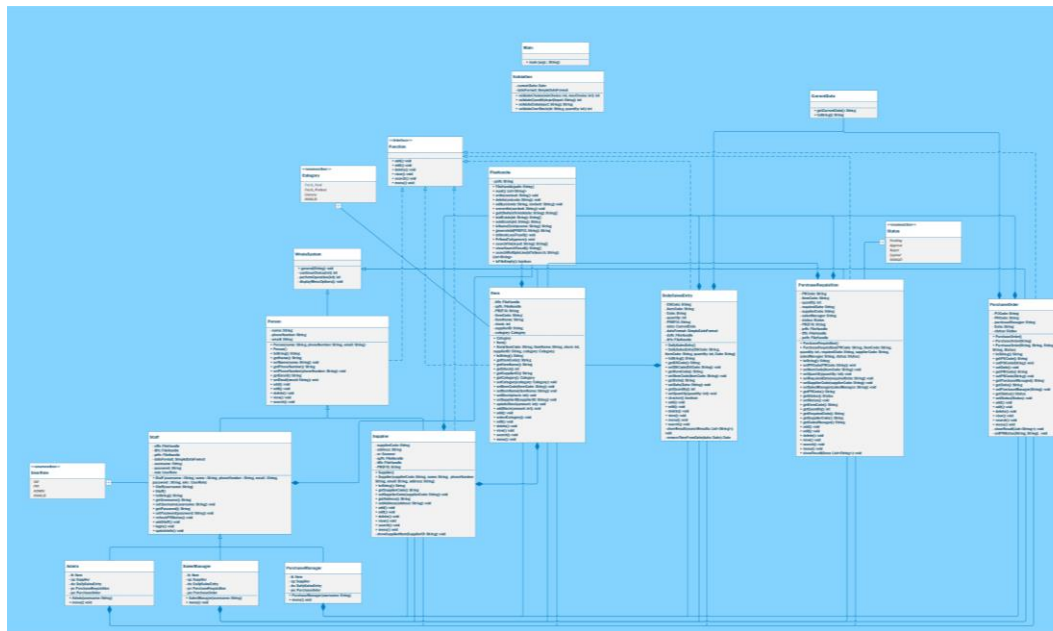


Figure 2: Class Diagram

## Classes

### Whole System:

- **Class Description:** Serves as a base class for various system entities.

**Person:**

- Class Description: Represents individuals in the system.
- Generalization Relationship: Person is a generalization of Staff and Supplier.

**Staff:**

- Class Description: Represents staff members.
- Generalization Relationship: Staff is a generalization of SalesManager, PurchaseManager, and Admin.
- Implementation Relationship: Implements Function interface.
- Composition Relationship: FileHandle is contained in Staff.

**Supplier:**

- **Class Description:** Represents suppliers.

- Generalization Relationship: Supplier is a generalization of Person.
- Composition Relationship: Item is contained in Supplier.

**SalesManager:**

- Class Description: Represents sales managers.
- Generalization Relationship: SalesManager is a generalization of Staff.
- Composition Relationship: Item, Supplier, DailySalesEntry, PurchaseRequisition, PurchaseOrder is contained in SalesManager

**PurchaseManager:**

- Class Description: Represents purchase managers.
- Generalization Relationship: PurchaseManager is a generalization of Staff.
- Composition Relationship: Item, Supplier, DailySalesEntry, PurchaseOrder is contained in PurchaseManager

**Admin:**

- Class Description: Represents administrators.
- Generalization Relationship: Admin is a generalization of Staff.
- Composition Relationship: Item, Supplier, DailySalesEntry, PurchaseRequisition, PurchaseOrder is contained in Admin.

**PurchaseOrder:**

- Class Description: Represents purchase orders.
- Implementation Relationship: Implements Function interface.
- Composition Relationship: FileHandle and CurrentDate is contained in PurchaseOrder.
- Composition Relationship: Purchase Order (PO) is contained in Sales Manager, Admin.

**PurchaseRequisition:**

- Class Description: Represents purchase requisitions.
- Implementation Relationship: Implements Function interface.



- Composition Relationship: PR is contained in PO.

**Item:**

- Class Description: Represents items.
- Implementation Relationship: Implements Function interface.
- Composition Relationship: Item is contained in Supplier.

**FileHandle:**

- Class Description: Handles file operations.
- Dependency Relationship: Used by various classes for file-related operations.
- Composition Relationship: FileHandle is contained in Staff, Item, Supplier, PO, PR, DailySalesEntry.

**CurrentDate:**

- Class Description: Provides the current date.
- Composition Relationship: CurrentDate is contained in DailySalesEntry and Purchase Order (PO).

**Validation:**

- Class Description: Handles input validation.
- Dependency Relationship: Used for input validation in various classes.

**Main:**

- Class Description: Main class for the application.
- Dependency Relationship: Creates a Staff object and invokes its login() method for application startup.

**Interface:****1. Function**

Function (Interface):

- Interface Description: Defines common methods for various system functionalities.

**Enumerations:**

## 1. UserRole

- Admin
- SALES MANAGER
- PURCHASE MANAGER
- INVALID
- Enum Description: Defines different user roles, including roles like Sales Manager, Purchase Manager, and Admin.

## 2. Category

- FRESH FOOD
- FRESH PRODUCE
- GROCERY
- INVALID
- Enum Description: Represents categories that items can belong to.

## 3. Status

- PENDING
- APPROVED
- REJECTED
- EXPIRED
- INVALID
- Enum Description: Represents various statuses, such as the status of Purchase Orders and Purchase Requisitions.

## 4.0 Sample Output

### 4.1 Login Page

```
*****  
PURCHASE ORDER MANAGEMENT SYSTEM  
*****  
  
Please enter your username:
```

*Figure 3: Login Page*

Once the system is opened, User is required to enter their unique username and password.

## 4.2 Sales Manager

```
1. 1111.
*****
PURCHASE ORDER MANAGEMENT SYSTEM
*****

Please enter your username: tan

Please enter your password: tan

Successfully logged in! Welcome back, tan.

!!!!!! The stock of ITEM003 is less than 5. !!!!!

!!!!!! The stock of ITEM004 is less than 5. !!!!!

!!!!!! The stock of ITEM001 is less than 5. !!!!!

*****
Sales Manager Menu
*****

1. Item Entry.
2. Supplier Entry.
3. Daily Item-wise Sales Entry.
4. Create Purchase Requisition.
5. Display Requisition.
6. List of Purchaser Orders.
7. Update Info.
8. View Own Info.
9. Log Out

Enter your choice: 8
```

Figure 4: Sales Manager Account Login

When the account is assigned to sale manager, the system will detect automatically and login to specific menu. After login successfully Sales Manager Account, the system will show which item's stock is less than 5 on the top of Sales Manager menu.

### 4.2.1 Sales Manager Menu

```
*****  
Sales Manager Menu  
*****  
  
1. Item Entry.  
2. Supplier Entry.  
3. Daily Item-wise Sales Entry.  
4. Create Purchase Requisition.  
5. Display Requisition.  
6. List of Purchaser Orders.  
7. Update Info.  
8. View Own Info.  
9. Log Out  
  
Enter your choice: .
```

*Figure 5: Sales Manager Menu*

The Sales Manager Menu has these operations:

1. Item Entry
2. Supplier Entry
3. Daily Item-wise Sales Entry
4. Create Purchase Requisition
5. Display Purchase Requisition
6. View List of Purchase Orders
7. Update Personal Account Information
8. View Personal Account Information
9. Log out

```
Enter your choice: 1
```

```
*****
```

```
Item Menu
```

```
*****
```

1. Add.
2. Edit.
3. Delete.
4. View.
5. Search.
6. Exit.

```
Enter your choice:
```

*Figure 6: Item menu*

Then if select Item Entry (Enter the number of the choice), it will access the Item entry page, that has:

1. Add Item
2. Edit Item
3. Delete Item
4. View all item information
5. Search item
6. Exit Item Entry Page

### 4.2.2 Item

Enter your choice: 1

\*\*\*\*\*

Item Menu

\*\*\*\*\*

1. Add.
2. Edit.
3. Delete.
4. View.
5. Search.
6. Exit.

*Figure 7: Item Menu*

### 4.2.2.1 Add Item

#### 4.2.2.1.1 Add new item

```
Enter your choice:
1

1. Add new item.
2. Add stock of existing item.

Enter choice: 1

Please enter item name:
Onion
Please enter the quantity (stock):
19
Please enter supplier code of this item:
SP003

ID not Found.
Enter again:
SP001

Select Category:
1. Fresh Food
2. Fresh Produce
3. Grocery

Enter the category number (1/2/3): 2

Congratulations! item added successfully!
```

*Figure 8: Add New Item*

After selecting "Add New Item," you will be prompted to enter the desired information. Once you provide this information, it will be successfully added to the database. If the ID is not existed, it will prompt to enter again.



**4.2.2.1.2 Add Stock of existing Item**

\*\*\*\*\*

1. Add.
2. Edit.
3. Delete.
4. View.
5. Search.
6. Exit.

Enter your choice:

1

1. Add new item.
2. Add stock of existing item.

Enter choice: 2

Please provide item code that you wish to add stock:

ITEM001

How much do you want to add:

43

*Figure 9: Add Stock*

If select “Add stock of existing item”, enter the item code that you want to add stock. For this example, the stock that is 19 originally, then add more 43. So the below picture shows now the stock of onion is 62.

```
-- -----
6. Exit.

Enter your choice:
4

*****
      Item Code: ITEM001
*****
      Item Name: Onion
      Quantity: 62
      Supplier Code: SP001
      Category: Fresh_Produce
*****
```

*Figure 10: Stock added in view item*

#### 4.2.2.2 Edit Item

```
*****
Item Menu
*****

1. Add.
2. Edit.
3. Delete.
4. View.
5. Search.
6. Exit.

Enter your choice:
2

Please provide item code that you wish to edit:
ITEM001

Please choose which component.
1. Item Name
2. Stock
3. Supplier ID
4. Category

Enter your choice:
```

*Figure 11: Edit Item*

After selecting “Edit”, the system will prompt you to enter the item that you want to edit. Then if the input ID exist, it will ask which component that you want to edit. The editable components have:

1. Item Name
2. Stock
3. Supplier Code
4. Category

#### 4.2.2.2.1 Edit name

```
Please choose which component.
1. Item Name
2. Stock
3. Supplier ID
4. Category

Enter your choice:
1

Enter changed content.
Cabbage

Congratulations! Item information updated successfully!

Do you want to continue this section? (yes/no):
no
```

Figure 12: Edit name of item

When select edit “Item Name”, System will prompt enter the new item name. Then the below picture shows the item name has been changed.

```
*****
      Item Code: ITEM001
*****
      Item Name: Cabbage
      Quantity: 62
      Supplier Code: SP001
      Category: Fresh_Produce
*****
```

Figure 13: Edited name of ITEM001

#### 4.2.2.2 Edit Stock

```
2. Edit.  
3. Delete.  
4. View.  
5. Search.  
6. Exit.  
  
Enter your choice:  
2  
  
Please provide item code that you wish to edit:  
ITEM001  
  
Please choose which component.  
1. Item Name  
2. Stock  
3. Supplier ID  
4. Category  
  
Enter your choice:  
2  
  
Enter changed content.  
76  
  
Congratulations! Item information updated successfully!  
  
Do you want to continue this section? (yes/no):  
NO
```

*Figure 14: Edit stock of item*

When select edit “Stock”, System will prompt enter the new quantity. Then the below picture shows the stock has been changed.

```
*****  
Item Code: ITEM001  
*****  
Item Name: Cabbage  
Quantity: 76  
Supplier Code: SP001  
Category: Fresh_Produce  
*****
```

*Figure 15: Edited Stock of ITEM001*

#### 4.2.2.2.3 Edit Supplier Code

```
Please provide item code that you wish to edit:
ITEM001

Please choose which component.
1. Item Name
2. Stock
3. Supplier ID
4. Category

Enter your choice:
3

Enter changed content.
SP002

Congratulations! Item information updated successfully!

Do you want to continue this section? (yes/no):
NO
```

*Figure 16: Edit Supplier Code*

When select edit “Supplier ID”, System will prompt enter the new Supplier Code. Then the below picture shows the supplier code has been changed.

```
*****
      Item Code: ITEM001
*****
      Item Name: Cabbage
      Quantity: 76
      Supplier Code: SP002
      Category: Fresh_Produce
*****
```

*Figure 17: Edited Supplier Code of ITEM001*

#### 4.2.2.2.4 Edit Category

```
Please provide item code that you wish to edit:
ITEM001

Please choose which component.
1. Item Name
2. Stock
3. Supplier ID
4. Category

Enter your choice:
4

Select Category:
1. Fresh Food
2. Fresh Produce
3. Grocery

Enter the category number (1/2/3): 1

Congratulations! Item information updated successfully!

Do you want to continue this section? (yes/no):
```

Figure 18: Edit Category

When select edit “Category”, System will prompt select the new item’s category. Then the below picture shows the category has been changed.

```
*****
      Item Code: ITEM001
*****
      Item Name: Cabbage
      Quantity: 76
      Supplier Code: SP002
      Category: Fresh_Food
*****
```

Figure 19: Edited Category of ITEM001



#### 4.2.2.3 Delete Item

```
*****  
Item Menu  
*****  
  
1. Add.  
2. Edit.  
3. Delete.  
4. View.  
5. Search.  
6. Exit.  
  
Enter your choice:  
3  
  
Please enter item code that you want to delete:  
ITEM001  
  
Congratulations! The item deleted successfully!  
  
Do you want to continue this section? (yes/no):  
NO
```

Figure 20: Delete Item

After selecting "Delete," the system will prompt you to enter the item code that needs to be deleted. Upon successful deletion, a notification confirming the deletion will be displayed. The picture below illustrates the result of searching for the deleted item code.

```
1. Add.  
2. Edit.  
3. Delete.  
4. View.  
5. Search.  
6. Exit.  
  
Enter your choice:  
5  
  
Please enter item code or name you want to search:  
ITEM001  
  
No results found.
```

*Figure 21: Search result*

#### 4.2.2.4 View Item

```
3. Delete.
4. View.
5. Search.
6. Exit.

Enter your choice:
4

*****
      Item Code: ITEM002
*****
      Item Name: cabbage
      Quantity: 10
      Supplier Code: SP002
      Category: Fresh_Produce
*****

*****
      Item Code: ITEM003
*****
      Item Name: oil
      Quantity: 0
      Supplier Code: SP002
      Category: Grocery
*****

*****
      Item Code: ITEM004
*****
      Item Name: minced meat
      Quantity: 0
      Supplier Code: SP003
      Category: Fresh_Food
```

Figure 22: View Item

After select “View”, the system will show all the item details.

#### 4.2.2.5 Search Item

```
*****

1. Add.
2. Edit.
3. Delete.
4. View.
5. Search.
6. Exit.

Enter your choice:
5

Please enter item code or name you want to search:
ITEM002

Item code: ITEM002
Name: cabbage
Stock: 10
Supplier code: SP002
catgeory: Fresh_Produce
```

Figure 23: Search Item

After selecting “Search”, the system will prompt to enter the item code that you want to search. Then It will display the specific item details.

#### 4.2.2.6 Exit Item Menu

```
*****
Item Menu
*****

1. Add.
2. Edit.
3. Delete.
4. View.
5. Search.
6. Exit.

Enter your choice:
6

*****
Sales Manager Menu
*****

1. Item Entry.
2. Supplier Entry.
3. Daily Item-wise Sales Entry.
4. Create Purchase Requisition.
5. Display Requisition.
6. List of Purchaser Orders.
7. Update Info.
8. View Own Info.
9. Log Out

Enter your choice: ||
```

Figure 24: Exit Item Menu

After Selecting “Exit”, it will return sales manager menu.

### 4.2.3 Supplier

```
9. Log Out

Enter your choice: 2

*****
Supplier Menu
*****

1. Add.
2. Edit.
3. Delete.
4. View.
5. Search.
6. Exit.

Enter your choice:
```

*Figure 25: Supplier Menu*

#### 4.2.3.1 Add Supplier

```
5. Search.
6. Exit.

Enter your choice:
1

Please enter supplier name:
SAYURAN
Please enter supplier phone number:
03-123456
Please enter supplier email address:
SAYURAN@email.COM
Please enter supplier address:
abc
Congratulations! Supplier added successfully!

Any items from this supplier to add?
1. Yes
2. No
Choose 1 or 2:
1

Enter name of the new item:
Chicken Breast

Select Category:
1. Fresh Food
2. Fresh Produce
3. Grocery

Enter the category number (1/2/3): 1

Congratulations! Item added successfully!
```

Figure 26: Add Supplier

After selecting "Add," you will be prompted to enter the desired information. Once you provide this information, it will be successfully added to the database.

```
Please enter supplier address:
abc
Congratulations! Supplier added successfully!

Any items from this supplier to add?
1. Yes
2. No
Choose 1 or 2:
1

Enter name of the new item:
Chicken Breast

Select Category:
1. Fresh Food
2. Fresh Produce
3. Grocery

Enter the category number (1/2/3): 1

Congratulations! Item added successfully!

Any items from this supplier to add?
1. Yes
2. No
Choose 1 or 2:
2

Do you want to continue this section? (yes/no):
no
```



```
Any items from this supplier to add?
1. Yes
2. No
Choose 1 or 2:
1

Enter name of the new item:
Chicken Breast

Select Category:
1. Fresh Food
2. Fresh Produce
3. Grocery

Enter the category number (1/2/3): 1

Congratulations! Item added successfully!
```

*Figure 27: Add Item after add supplier*

After successfully adding a supplier, the system will prompt you to inquire if there are any items associated with this supplier. Upon entering the item details, the added item will be included in the file, and the initial quantity will be automatically set to 0. If no any item need to be added, just select “No”.

```
Please enter item code or name you want to search:
ITEM001

Item code: ITEM001
Name: Chicken Breast
Stock: 0
Supplier code: SP003
catgeory: Fresh_Food
```

*Figure 28: Search result of new ITEM001*

This displays the search result for the newly added item after adding the supplier. The code of the newly added item is "ITEM001" because the original "ITEM001" had been deleted previously, as shown in the output of the "Item Delete" function.

#### 4.2.3.2 Edit Supplier

```
Supplier Menu
*****

1. Add.
2. Edit.
3. Delete.
4. View.
5. Search.
6. Exit.

Enter your choice:
2

Please provide supplier code that you wish to edit:
SP003

Please choose which component.
1.Name
2.Phone Number
3.Email address
4.Address
```

*Figure 29: Edit Supplier*

After selecting “Edit”, the system will prompt you to enter the supplier ID that you want to edit. Then if the input ID exist, it will ask which component that you want to edit. The editable components have:

1. Name
2. Phone Number
3. Email Address
4. Address

#### 4.2.3.2.1 Edit Supplier Name

```
Please provide supplier code that you wish to edit:
SP003

Please choose which component.
1.Name
2.Phone Number
3.Email address
4.Address

Enter your choice:
1

Enter changed content.
PPP

Congratulations! Supplier information updated successfully!

Do you want to continue this section? (yes/no):
NO
```

*Figure 30: Edit Supplier Name*

When select edit “Name”, System will prompt enter the new supplier’s name. Then the below picture shows the supplier name has been changed.

```
*****
      Supplier Code: SP003
*****

Name: PPP
Phone Number: 03-123456
Email: SAYURAN@email.COM
Address: abc
```

*Figure 31: Edited name of SP001*

#### 4.2.3.2.2 Edit Supplier Phone Number

```
Enter your choice:
2

Please provide supplier code that you wish to edit:
SP003

Please choose which component.
1.Name
2.Phone Number
3.Email address
4.Address

Enter your choice:
2

Enter changed content.
010-88888888

Congratulations! Supplier information updated successfully!

Do you want to continue this section? (yes/no):
NO
```

Figure 32: Edit Supplier Phone Number

When select edit “Phone Number”, System will prompt enter the new supplier’s phone number. Then the below picture shows the item name has been changed.

```
*****
Supplier Code: SP003
*****
Name: PPP
Phone Number: 010-88888888
Email: SAYURAN@email.COM
Address: abc
```

Figure 33: Edited Phone Number of SP003

#### 4.2.3.2.3 Edit Supplier Email Address

```
Enter your choice:
2

Please provide supplier code that you wish to edit:
SP003

Please choose which component.
1.Name
2.Phone Number
3.Email address
4.Address

Enter your choice:
3

Enter changed content.
PPP@email.com

Congratulations! Supplier information updated successfully!

Do you want to continue this section? (yes/no):
```

Figure 34: Edit Email Address

When select edit “Email Address”, System will prompt enter the new supplier’s email address. Then the below picture shows the supplier email address has been changed.

```
*****
Supplier Code: SP003
*****
Name: PPP
Phone Number: 010-88888888
Email: PPP@email.com
Address: abc
```

Figure 35: Edited Email Address of SP003

#### 4.2.3.2.4 Edit Supplier Address

```
Enter your choice:
2

Please provide supplier code that you wish to edit:
SP003

Please choose which component.
1.Name
2.Phone Number
3.Email address
4.Address

Enter your choice:
4

Enter changed content.
Jalan Besar Taman Kecil

Congratulations! Supplier information updated successfully!
```

*Figure 36: Edit Suppiler Address*

When select edit “Address”, System will prompt enter the new supplier’s email address. Then the below picture shows the address has been changed.

```
*****
      Supplier Code: SP003
*****
Name: PPP
Phone Number: 010-88888888
Email: PPP@email.com
Address: Jalan Besar Taman Kecil
```

*Figure 37: Edited Address of SP003*

#### 4.2.3.3 Delete Supplier

```
NO

*****
Supplier Menu
*****

1. Add.
2. Edit.
3. Delete.
4. View.
5. Search.
6. Exit.

Enter your choice:
3

Please enter supplier code that you want to delete:
SP003

Congratulations! The supplier deleted successfully!

Do you want to continue this section? (yes/no):
NO
```

Figure 38: Delete Supplier

After selecting "Delete," the system will prompt you to enter the supplier code that needs to be deleted. Upon successful deletion, a notification confirming the deletion will be displayed. The picture below illustrates the result of searching for the deleted supplier code.

```
Supplier Menu
*****
```

1. Add.
2. Edit.
3. Delete.
4. View. |
5. Search.
6. Exit.

```
Enter your choice:
5
```

```
Please enter supplier code or name you want to search: |
SP003
```

```
No results found.
```

```
Do you want to continue this section? (yes/no):
```

*Figure 39: Search Result of SP003*



#### 4.2.3.4 View Supplier

```

Supplier Menu
*****

1. Add.
2. Edit.
3. Delete.
4. View.
5. Search.
6. Exit.

Enter your choice:
4

*****
      Supplier Code: SP001
*****
Name: POP
Phone Number: 0123456677
Email: wee@gmail.com
Address: JALAN 123 TAMAN BUNGA
*****

*****
      Supplier Code: SP002
*****
Name: ppe
Phone Number: 0987612343546
Email: ppe2gmail.com
Address: ppe compony
Supplied item 1: ITEM002
Supplied item 2: ITEM003
Supplied item 3: ITEM005
*****

```

*Figure 40:View Supplier*

After selecting "View," the system will display all the supplier details and list the items that were supplied by the specific supplier.

#### 4.2.3.5 Search Supplier

```
Supplier Menu
*****

1. Add.
2. Edit.
3. Delete.
4. View.
5. Search.
6. Exit.

Enter your choice:
5

Please enter supplier code or name you want to search:
SP002

Supplier code: SP002
Name: ppe
Phone number: 0987612343546
Email: ppe2gmail.com
Address: ppe compony

Supplied item 1: ITEM002
Supplied item 2: ITEM003
Supplied item 3: ITEM005

Do you want to continue this section? (yes/no):
```

*Figure 41: Search Result of SP002*

After selecting "Search," the system will prompt you to enter the supplier code you want to search for. Once you provide the supplier code, the system will display the specific supplier details and list the items supplied by this supplier.

#### 4.2.3.6 Exit Supplier Menu

```
*****
Supplier Menu
*****

1. Add.
2. Edit.
3. Delete.
4. View.
5. Search.
6. Exit.

Enter your choice:
6

*****
Sales Manager Menu
*****

1. Item Entry.
2. Supplier Entry.
3. Daily Item-wise Sales Entry.
4. Create Purchase Requisition.
5. Display Requisition.
6. List of Purchaser Orders.
7. Update Info.
8. View Own Info.
9. Log Out

Enter your choice: |
```

Figure 42: Exit Supplier Menu

After Selecting “Exit”, it will return sales manager menu.

#### 4.2.4 Daily Item-wise Sales Entry

```
*****  
Sales Manager Menu  
*****  
  
1.  Item Entry.  
2.  Supplier Entry.  
3.  Daily Item-wise Sales Entry.  
4.  Create Purchase Requisition.  
5.  Display Requisition.  
6.  List of Purchaser Orders.  
7.  Update Info.  
8.  View Own Info.  
9.  Log Out  
  
Enter your choice: 3  
  
*****  
Daily Item-wise Sales Menu  
*****  
  
1.  Generate.  
2.  Edit.  
3.  Delete.  
4.  View.  
5.  Search.  
6.  Exit
```

*Figure 43: Daily Item-wise Sales Menu*

#### 4.2.4.1 Generate Daily Sales

---

```
1. Generate.
2. Edit.
3. Delete.
4. View.
5. Search.
6. Exit
|
Enter your choice:
1
Please enter item code:
ITEM002
Please enter its quantity:
11

Input quantity is over stock.
Please enter again:
4

Congratulations! item added successfully!

Do you want to add another item? (yes/no)
YES
Please enter item code:
ITEM005
Please enter its quantity:
3

Congratulations! item added successfully!
```

Figure 44: Generate Daily Sales

After selecting "Generate," the system will prompt you to enter the sales item code and its quantity. If the input quantity is over the stock, the system will show the notification and prompt to enter again. It will then save this information into a file. If you need to add more items, you can enter "YES" when prompted by the continuation message.

-----	
DS001	2023-09-09
-----	
Item Code	Quantity
-----	-----
ITEM002	4
-----	-----
ITEM005	3
-----	-----

Figure 45: New Daily Sales has been added.

This indicates that the daily sales have been successfully generated, and it's important to note that the date for the daily sales is automatically set.

```

*****
Item Code: ITEM002
*****
Item Name: cabbage
Quantity: 10
Supplier Code: SP002
Category: Fresh_Produce
*****

```

Figure 48: Original stock of ITEM002

<pre> ***** Item Code: ITEM002 ***** Item Name: cabbage Quantity: 6 Supplier Code: SP002 Category: Fresh_Produce ***** </pre>	<pre> ***** Item Code: ITEM005 ***** Item Name: egg Quantity: 7 Supplier Code: SP002 Category: Fresh_Produce ***** </pre>
---	---

Figure 46: The stock of ITEM002 after sales

Figure 47: The stock of ITEM005 after sales

The stock will be updated when the daily sales is generated.

```
*****
Daily Item-wise Sales Menu
*****

1. Generate.
2. Edit.
3. Delete.
4. View.
5. Search.
6. Exit

Enter your choice:
1

Today's sales data has already been created.
If you need to make any changes or add new items, please use the edit function.
```

*Figure 49: Limited Daily Sale Entry*

If the current date daily sales is generated, the system will not allowed to generated again.  
You need go to edit function and add more item or edit item.

#### 4.2.4.2: Edit Daily sales

```
1. Generate.
2. Edit.
3. Delete.
4. View.
5. Search.
6. Exit

Enter your choice:
2

Please provide Daily Sales code that you wish to edit:
DS001
DS Lines for Editing:
```

-----	
DS001	2023-09-09
-----	
Item Code	Quantity
-----	-----
ITEM002	4
-----	-----
ITEM005	3
-----	-----

```
3. Add new Item

Please choose which DS line to edit (1-3):
```

Figure 50: Edit Daily Sales

After selecting “Edit,” the system will prompt you to enter the Daily Sales Code that you want to edit. Upon entering the existing daily sales code, it will display the specific daily sales details and ask if you want to add more items to this daily sales record. You will need to enter the line number to specify which item to edit. For this example, in DS001 “ITEM002” is the first line, you need to enter “1” to edit “ITEM002.”



**4.2.4.2.1: Modify Sales Item**

-----	
DS001	2023-09-09
-----	
Item Code	Quantity
-----	-----
ITEM002	4
-----	-----
ITEM005	3
-----	-----

3. Add new Item

Please choose which DS line to edit (1-3):

1

Select an action:

1. Modify Sales Item

2. Remove Sales Item

Enter your choice:

1

Enter the new Quantity: 6

Do you want to edit another item? (yes/no)

NO

*Figure 51: Modify Sales Item*

After selecting the specific item in the daily sales record that you wish to edit, the system will prompt you to enter the new quantity.

-----	
DS001	2023-09-09
-----	
Item Code	Quantity
-----	-----
ITEM002	6
-----	-----
ITEM005	3
-----	-----

*Figure 52: Edited Daily Sales*

The item quantity has been updated to 6 in the DS001.

```

*****
      Item Code: ITEM002
*****
      Item Name: cabbage
      Quantity: 4
      Supplier Code: SP002
      Category: Fresh_Produce
*****

```

*Figure 53: Edited Stock of ITEM002*

The stock of the ITEM002 has also been updated.

**4.2.4.2.2: Remove Sales Item**

Please provide Daily Sales code that you wish to edit:  
DS001  
DS Lines for Editing:

-----	
DS001	2023-09-09
-----	
Item Code	Quantity
-----	-----
ITEM002	6
-----	-----
ITEM005	3
-----	-----

3. Add new Item

Please choose which DS line to edit (1-3):  
2

Select an action:  
1. Modify Sales Item  
2. Remove Sales Item

Enter your choice:  
2

Sales Item removed.

Do you want to edit another item? (yes/no)  
NO

Figure 54: Remove Item in Daily Sales

After choosing the item that you want to remove then select “Remove Sales Item”, the specific item will be removed. The below picture shows the removed item daily sales.

-----	
DS001	2023-09-09
-----	
Item Code	Quantity
-----	-----
ITEM002	6
-----	-----

Figure 55: The removed item of DS001

```
*****  
      Item Code: ITEM005  
*****  
      Item Name: egg  
      Quantity: 10  
      Supplier Code: SP002  
      Category: Fresh_Produce  
*****
```

*Figure 56: The Stock of ITEM005 after removed*

After the item has been removable in the daily sales, the stock of the item will be updated.

**4.2.4.2.3: Add New Item**

Please provide Daily Sales code that you wish to edit:

DS001

DS Lines for Editing:

-----	
DS001	2023-09-09
-----	
Item Code	Quantity
-----	-----
ITEM002	6
-----	-----

2. Add new Item

Please choose which DS line to edit (1-2):

2

Enter Item Code: ITEM005

Enter Quantity: 8

Sales Item added.

Do you want to edit another item? (yes/no)

NO

*Figure 57: Add New Item in Daily Sales*

After selecting new item, the system prompt you to enter new item code and quantity. The below picture shows the new item added in the DS001.

-----	
DS001	2023-09-09
-----	
Item Code	Quantity
-----	-----
ITEM002	6
-----	-----
ITEM005	8
-----	-----

Type Enter to continue...

*Figure 58: New Added in DS001*

#### 4.2.4.3: View Daily Sales

```
*****
Daily Item-wise Sales Menu
*****

1. Generate.
2. Edit.
3. Delete.
4. View.
5. Search.
6. Exit

Enter your choice:
4

|-----|
|      DS001      2023-09-09      |
|-----|
| Item Code      | Quantity      |
|-----|-----|
| ITEM002        | 6              |
|-----|-----|
| ITEM005        | 8              |
|-----|-----|

Press Enter to continue
```

Figure 59: View Daily Sales

After selecting “View”, the system will display all daily sales.

#### 4.2.4.4: Search Daily Sales

```
*****
Daily Item-wise Sales Menu
*****

1. Generate.
2. Edit.
3. Delete.
4. View.
5. Search.
6. Exit

Enter your choice:
5

Please enter daily sales id you want to search:
DS001

|-----|
|      DS001      2023-09-09      |
|-----|
| Item Code      | Quantity      |
|-----|-----|
| ITEM002        | 6            |
|-----|-----|
| ITEM005        | 8            |
|-----|-----|
```

Figure 60: Search result of searching DS001

After selecting "Search," the system will prompt you to enter the daily sales code you want to search for.



#### 4.2.4.5 Delete Daily Sales

```
*****  
Daily Item-wise Sales Menu  
*****  
  
1. Generate.  
2. Edit.  
3. Delete.  
4. View.  
5. Search.  
6. Exit  
  
Enter your choice:  
3  
Please enter daily sales code that you want to delete:  
DS001  
Congratulations! The daily sales deleted successfully!
```

*Figure 61: Delete Daily Sales*

After selecting "Delete," the system will prompt you to enter the daily sales code that needs to be deleted. Upon successful deletion, a notification confirming the deletion will be displayed. The picture below illustrates the result of searching for the deleted daily sales code.

1. Generate.
2. Edit.
3. Delete.
4. View.
5. Search.
6. Exit

Enter your choice:

5

Please enter daily sales id you want to search:

DS001

Nothing found.

*Figure 62: Search result of searching DS001*

#### 4.2.4.6: Exit Daily Item-wise Sales Menu

```
*****
Daily Item-wise Sales Menu
*****

1. Generate.
2. Edit.
3. Delete.
4. View.
5. Search.
6. Exit

Enter your choice:
6

*****
Sales Manager Menu
*****

1. Item Entry.
2. Supplier Entry.
3. Daily Item-wise Sales Entry.
4. Create Purchase Requisition.
5. Display Requisition.
6. List of Purchaser Orders.
7. Update Info.
8. View Own Info.
9. Log Out
```

Figure 63: Exit Daily Item-wise Sales Menu

After Selecting “Exit”, it will return sales manager menu.

### 4.2.5 Purchase Requisition

```
*****
Sales Manager Menu
*****

1. Item Entry.
2. Supplier Entry.
3. Daily Item-wise Sales Entry.
4. Create Purchase Requisition.
5. Display Requisition.
6. List of Purchaser Orders.
7. Update Info.
8. View Own Info.
9. Log Out

Enter your choice: 4

*****
Purchase Requisition Menu
*****

1. Add.
2. Edit.
3. Delete.
4. View.
5. Search.
6. Exit.

Enter your choice:
1
```

*Figure 64: Purchase Requisition Menu*

#### 4.2.5.1 Add Purchase Requisition

```
3. Add.  
4. View.  
5. Search.  
6. Exit.  
  
Enter your choice:  
1  
Please enter its required date:  
2023-01-01  
  
Date is earlier than the current date. Please try again.  
20231231  
  
Invalid date format. Please enter the date in yyyy-MM-dd format.  
2023-09-18  
  
Please enter item code:  
ITEM001  
Please enter its quantity:  
6  
  
Congratulations! item added successfully!  
  
Do you want to add another item? (yes/no)  
YES  
  
Please enter item code:  
ITEM004  
Please enter its quantity:  
20  
  
Congratulations! item added successfully!  
  
Do you want to add another item? (yes/no)  
NO
```

Figure 65: Add Purchase Requisition

After selecting "Add," the system will prompt you to enter the required date in a specific format, ensuring that you cannot enter a date before the current date. Once the date is entered correctly, you can input the item code and quantity needed. If you have additional items to add, you can proceed to enter them without the need to enter the date again. The below

picture shows new purchase requisition is added, the sales manager name will be show on the top of table.

tan		
PR001	2023-09-18	Pending
Item Code	Quantity	Supplier Code
ITEM001	6	SP003
ITEM004	20	SP003

Figure 66: New Purchase Requisition is added

**4.2.5.2: Edit Purchase Requisition****Purchase Requisition Menu**

\*\*\*\*\*

1. Add.
2. Edit.
3. Delete.
4. View.
5. Search.
6. Exit.

Enter your choice:

2

Please provide PR code that you wish to edit:

PR001

PR Lines for Editing:

-----		
tan		
-----		
PR001	2023-10-09	Pending
-----		
Item Code	Quantity	Supplier Code
-----	-----	-----
ITEM002	23	SP002
-----	-----	-----
ITEM004	20	SP003
-----	-----	-----

3. Edit Required Date

Please choose which PR line to edit (1-3):

*Figure 67: Edit Purchase Requisition*

After selecting “Edit,” the system will prompt you to enter the Purchase Requisition Code that you want to edit. Upon entering the existing purchase requisition code, it will display the specific purchase requisition details. Then also you will need to enter the line number to specify which item to edit. For this example, in PR001 “ITEM001” is the first line, you need to enter “1” to edit “ITEM001.”





-----		
tan		
-----		
PR001	2023-10-09	Pending
-----		
Item Code	Quantity	Supplier Code
-----	-----	-----
ITEM002	4	SP002
-----	-----	-----
ITEM004	20	SP003
-----	-----	-----

Figure 69: Edited Item code of PR001

After item code edited successfully, the supplier code will be set the specific id.

#### 4.2.5.2.2 Edit Quantity

Please provide PR code that you wish to edit:

PR001

PR Lines for Editing:

-----		
tan		
-----		
PR001	2023-10-09	Pending
-----		
Item Code	Quantity	Supplier Code
-----	-----	-----
ITEM002	4	SP002
-----	-----	-----
ITEM004	20	SP003
-----	-----	-----

#### 3. Edit Required Date

Please choose which PR line to edit (1-3):

1

Choose which field to edit within this PR line:

1. Item Code

2. Quantity

Enter your choice:

2

Enter new Quantity:

12

Do you want to continue this section? (yes/no):

NO

Figure 70: Edit Quantity

After select edit quantity of the specific item, you will be prompted to enter the new quantity of this item. The below picture shows the edited quantity of ITEM002 in PR001.

-----		
tan		
-----		
PR001	2023-10-09	Pending
-----		
Item Code	Quantity	Supplier Code
-----	-----	-----
ITEM002	12	SP002
-----	-----	-----
ITEM004	20	SP003
-----	-----	-----

*Figure 71: Edited Quantity of ITEM002*

**4.2.5.2.3 Edit Require Date**

-----		
tan		
-----		
PR001	2023-10-09	Pending
-----		
Item Code	Quantity	Supplier Code
-----	-----	-----
ITEM002	12	SP002
-----	-----	-----
ITEM004	20	SP003
-----	-----	-----

**3. Edit Required Date**

Please choose which PR line to edit (1-3):

3

Enter new Required Date:

20230910

Invalid date format. Please enter the date in yyyy-MM-dd format.

2023-09-09

Date is earlier than the current date. Please try again.

2023-10-31

Do you want to continue this section? (yes/no):

NO

*Figure 72: Edit Require Date*

After selecting “Edit Require Date”, the system will prompt you to enter the new required date. The date will be validated if it is correct. The below picture shows the edited require date.

tan		
PR001	2023-10-31	Pending
Item Code	Quantity	Supplier Code
ITEM002	12	SP002
ITEM004	20	SP003

Figure 73: Edited Require Date of PR001

#### 4.2.5.2.4 Edit Approved or Rejected PR

Please provide PR code that you wish to edit:

PR001

PR Lines for Editing:

-----		
tan		
-----		
PR001	2023-10-09	Approve
-----		
Item Code	Quantity	Supplier Code
-----	-----	-----
ITEM002	23	SP002
-----	-----	-----
ITEM004	20	SP003
-----	-----	-----

#### 3. Edit Required Date

Please choose which PR line to edit (1-3):

1

Choose which field to edit within this PR line:

1. Item Code

2. Quantity

Enter your choice:

1

Enter new Item Code:

ITEM002

Figure 74: Edit Approved Purchase Requisition

If the Purchase Requisition is approved, if you select to edit the approved or rejected purchase requisition, the status will be changed to pending again. The below picture shows the changed status after edited.

-----		
tan		
-----		
PR001	2023-10-09	Pending
-----		
Item Code	Quantity	Supplier Code
-----	-----	-----
ITEM002	23	SP002
-----	-----	-----
ITEM004	20	SP003
-----	-----	-----

-----		
tan		
-----		
PR002	2023-09-12	Pending
-----		
Item Code	Quantity	Supplier Code
-----	-----	-----
ITEM001	4	SP003
-----	-----	-----
ITEM002	30	SP002
-----	-----	-----
ITEM003	12	SP002
-----	-----	-----

Do you want to continue this section? (yes/no):

|

Figure 75: Status Changed After Edited

**4.2.5.2.5 Edit Expired Purchase Requisition**

-----		
tan		
-----		
PR004	2023-09-08	Expired
-----		
Item Code	Quantity	Supplier Code
-----	-----	-----
ITEM001	15	SP002
-----	-----	-----
ITEM002	8	SP002
-----	-----	-----

*Figure 76: Status Expired*



```
*****
Purchase Requisition Menu
*****

1. Add.
2. Edit.
3. Delete.
4. View.
5. Search.
6. Exit.

Enter your choice:
2

Please provide PR code that you wish to edit:
PR004
This PR is already expired, please create the new Purchase Requisition

Do you want to continue this section? (yes/no):
NO
```

*Figure 77: Edit Status of Purchase Requisition is Expired*

If you want to edit the purchase requisition that the status is “Expired”, this operation is not allowed.

### 4.2.5.3 Delete Purchase Requisition

```
*****
Purchase Requisition Menu
*****

1. Add.
2. Edit.
3. Delete.
4. View.
5. Search.
6. Exit.

Enter your choice:
3

Please enter purchase requisition code that you want to delete:
PR001

Congratulations! The purchase requisition deleted successfully!
```

*Figure 78: Delete Requisition*

After selecting "Delete," the system will prompt you to enter the purchase requisition code that needs to be deleted. Upon successful deletion, a notification confirming the deletion will be displayed. The picture below illustrates the result of searching for the deleted purchase

requisition code.

```
*****
Purchase Requisition Menu
*****

1. Add.
2. Edit.
3. Delete.
4. View.
5. Search.
6. Exit.

Enter your choice:
5

Please enter purchase requisition id you want to search:
PR001

No purchase requisition found.

Do you want to continue this section? (yes/no):
```

*Figure 79: Search result of searching PR001*

**4.2.5.4 View Purchase Requisition**

tan		
PR003	2023-09-10	Approve
Item Code	Quantity	Supplier Code
ITEM002	15	SP002
ITEM003	10	SP002

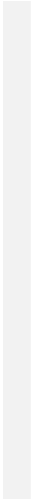
tan		
PR002	2023-09-12	Pending
Item Code	Quantity	Supplier Code
ITEM001	4	SP003
ITEM002	30	SP002
ITEM003	12	SP002

tan		
PR004	2023-09-08	Expired
Item Code	Quantity	Supplier Code
ITEM001	15	SP002
ITEM002	8	SP002

*Figure 80: View Purchase Requisition*

After selecting "View," the system will display all purchase requisitions.

#### 4.2.5.4.1 Expired Status



-----		
tan		
-----		
PR004	2023-09-08	Expired
-----		
Item Code	Quantity	Supplier Code
-----		
ITEM001	15	SP002
-----		
ITEM002	8	SP002
-----		

*Figure 81: Status of Purchase Requisition is Expired*

If the status is marked as "expired," it means that the purchase requisition had not been approved or rejected before the required date. The system monitors the current date, and when it detects that the current date has passed the required date, it automatically changes the status to "Expired" for those requisitions.

#### 4.2.5.5 Search Purchase Requisition

Enter your choice:

5

Please enter purchase requisition id you want to search:

PR002

-----		
tan		
-----		
PR002	2023-09-12	Pending
-----		
Item Code	Quantity	Supplier Code
-----	-----	-----
ITEM001	4	SP003
-----	-----	-----
ITEM002	30	SP002
-----	-----	-----
ITEM003	12	SP002
-----	-----	-----

Do you want to continue this section? (yes/no):

Figure 82: Search Result of PR002

After selecting "Search," the system will prompt you to enter the purchase requisition code you want to search for.

#### 4.2.5.6 Exit Purchase Requisition

```
*****
Purchase Requisition Menu
*****

1. Add.
2. Edit.
3. Delete.
4. View.
5. Search.
6. Exit.

Enter your choice:
6

*****
Sales Manager Menu
*****

1. Item Entry.
2. Supplier Entry.
3. Daily Item-wise Sales Entry.
4. Create Purchase Requisition.
5. Display Requisition.
6. List of Purchaser Orders.
7. Update Info.
8. View Own Info.
9. Log Out
```

Figure 83: Exit Purchase Requisition

After Selecting “Exit”, it will return sales manager menu.

## 4.2.6 List of Purchase Order

```

*****
Sales Manager Menu
*****

1.  Item Entry.
2.  Supplier Entry.
3.  Daily Item-wise Sales Entry.
4.  Create Purchase Requisition.
5.  Display Requisition.
6.  List of Purchaser Orders.
7.  Update Info.
8.  View Own Info.
9.  Log Out

Enter your choice: 6

*****
                PO Code: PO001
*****
PR Code: PR003
Date: 2023-09-10
Handled By: soo
*****
                Item section as below:
*****
Item Code 1: ITEM002
Quantity : 15
Required Date : 2023-09-10
Item Code 2: ITEM003
Quantity : 10
Required Date : 2023-09-10
*****

```

*Figure 84: List of Purchase Order*

When you select "List of Purchase Orders," the system will display all the purchase order details, including the Purchase Requisition code, required date, the name of the approving Purchase Manager, and the details of the items that need to be procured. This information



provides a comprehensive overview of the purchase orders in the system.

```
*****  
Sales Manager Menu  
*****  
  
1. Item Entry.  
2. Supplier Entry.  
3. Daily Item-wise Sales Entry.  
4. Create Purchase Requisition.  
5. Display Requisition.  
6. List of Purchaser Orders.  
7. Update Info.  
8. View Own Info.  
9. Log Out  
  
Enter your choice: 6  
  
Nothing in the file.
```

*Figure 85: List of Purchase Order If no Purchase Order in file*

If no purchase order in file, the system will display “Nothing in the file”.

## 4.2.7 Update Info

```
*****
Sales Manager Menu
*****

1.  Item Entry.
2.  Supplier Entry.
3.  Daily Item-wise Sales Entry.
4.  Create Purchase Requisition.
5.  Display Requisition.
6.  List of Purchaser Orders.
7.  Update Info.
8.  View Own Info.
9.  Log Out

Enter your choice: 7

Please choose which component.
1. Name
2. Phone Number
3. Email address
4. Password
```

Figure 86: Update Info Menu

```
*****
Username: tan
*****
Name: wee
Phone Number: 0123456789
Email: abc@example.com
Role: SM
*****
```

Figure 87: The original account information

#### 4.2.7.1 Edit Name

```
Please choose which component.
1. Name
2. Phone Number
3. Email address
4. Password

Enter your choice:
1

Enter changed content.
WEE

Congratulations! Staff information updated successfully!
```

Figure 88: Edit Name of username "tam"

When select edit "Name", system will prompt enter the new account name. Then the below picture shows the account name has been changed.

```
*****
Username: tan
*****
Name: WEE
Phone Number: 0123456789
Email: abc@example.com
Role: SM
*****
```

Figure 89: Edited name of username "tan"

#### 4.2.7.2 Edit Phone Number

```
*****  
Sales Manager Menu  
*****  
  
1.  Item Entry.  
2.  Supplier Entry.  
3.  Daily Item-wise Sales Entry.  
4.  Create Purchase Requisition.  
5.  Display Requisition.  
6.  List of Purchaser Orders.  
7.  Update Info.  
8.  View Own Info.  
9.  Log Out  
  
Enter your choice: 7  
  
Please choose which component.  
1. Name  
2. Phone Number  
3. Email address  
4. Password  
  
Enter your choice:  
2  
  
Enter changed content.  
012-2222222  
  
Congratulations! Staff information updated successfully!
```

Figure 90: Edit Phone Number of username "tan"

When select edit “Phone Number”, system will prompt enter the new user’s phone number. Then the below picture shows the phone number has been changed.

```
*****
      Username: tan
*****|
Name: WEE
Phone Number: 012-2222222
Email: abc@example.com
Role: SM
*****
```

*Figure 91: Edited Phone Number of username "tan"*

### 4.2.7.3 Edit Email Address

---

```
*****
Sales Manager Menu
*****

1. Item Entry.
2. Supplier Entry.
3. Daily Item-wise Sales Entry.
4. Create Purchase Requisition.
5. Display Requisition.
6. List of Purchaser Orders.
7. Update Info.
8. View Own Info.
9. Log Out

Enter your choice: 7

Please choose which component.
1. Name
2. Phone Number
3. Email address
4. Password

Enter your choice:
3

Enter changed content.
WEE@email.com

Congratulations! Staff information updated successfully!
```

Figure 92: Edit Email Address of username "tan"

When select edit “Email Address”, system will prompt enter the new user’s email address. Then the below picture shows the email address has been changed.

```
*****
Username: tan
*****
Name: WEE
Phone Number: 012-2222222
Email: WEE@email.com
Role: SM
*****
```

*Figure 93: Edited Email Address of username "tan"*

#### 4.2.7.4 Edit Password

```
*****
Sales Manager Menu
*****

1. Item Entry.
2. Supplier Entry.
3. Daily Item-wise Sales Entry.
4. Create Purchase Requisition.
5. Display Requisition.
6. List of Purchaser Orders.
7. Update Info.
8. View Own Info.
9. Log Out

Enter your choice: 7

Please choose which component.
1. Name
2. Phone Number
3. Email address
4. Password

Enter your choice:
4

Enter changed content.
abc

Congratulations! Staff information updated successfully!
```

Figure 94: Edit password of username "tan"

When selecting the "Edit Password" option, the system will prompt you to enter a new password for the user. The below picture shows "tan" attempt to log in again with the original password after making this change, it will not be allowed. Instead, "tan" uses the new password to log in successfully.



```
*****
PURCHASE ORDER MANAGEMENT SYSTEM
*****

Please enter your username: tan

Please enter your password: tan

Wrong password!

Please enter your username: tan

Please enter your password: abc

Successfully logged in! Welcome back, tan.

!!!!!! The stock of ITEM003 is less than 5. !!!!!
!!!!!! The stock of ITEM004 is less than 5. !!!!!
!!!!!! The stock of ITEM001 is less than 5. !!!!!

*****
```

*Figure 95: Login page after password changed*



### 4.2.8 View Info

```
*****
Sales Manager Menu
*****

1. Item Entry.
2. Supplier Entry.
3. Daily Item-wise Sales Entry.
4. Create Purchase Requisition.
5. Display Requisition.
6. List of Purchaser Orders.
7. Update Info.
8. View Own Info.
9. Log Out

Enter your choice: 8

*****
      Username: tan
*****
Name: WEE
Phone Number: 012-2222222
Email: WEE@email.com
Role: SM
*****
```

*Figure 96: View Account Information of username "tan"*

After selected “View Info”, the system will display the specific account details.

## 4.2.9 Log Out

```
*****
Sales Manager Menu
*****

1.  Item Entry.
2.  Supplier Entry.
3.  Daily Item-wise Sales Entry.
4.  Create Purchase Requisition.
5.  Display Requisition.
6.  List of Purchaser Orders.
7.  Update Info.
8.  View Own Info.
9.  Log Out

Enter your choice: 9

See you again,tan.

*****
PURCHASE ORDER MANAGEMENT SYSTEM
*****

Please enter your username: ||
```

---

Figure 97: Log Out

When selecting “Log Out”, the system will return to login page.

## 4.3 Purchase Manager

### 4.3.1 Purchase Manager Menu

```
Please enter your username: soo

Please enter your password: soo

Successfully logged in! Welcome back, soo.

!!!!!! There are 2 PR need to be handled !!!!!

*****
Purchase Manager Menu
*****

1. List of Items.
2. List of Supplier.
3. Display Requisition.
4. Generate Purchase Order.
5. List of Purchaser Orders.
6. Update Info.
7. View Own Info.
8. Log out.

Enter your choice:
```

*Figure 98: Purchase Manager Menu*

After login successfully Purchase Manager Account, the system will show whether any purchase requisition need to be handled on the top of Purchase Manager menu. The Purchase Manager menu has these operations:

1. View List of Item
2. View List of Supplier
3. Display Purchase Requisition
4. Generate Purchase Order
5. View List of Purchase Orders
6. Update Personal Account Information
7. View Personal Account Information

8. Log out

### **4.3.2 List of Items**

If you select the first option of the purchase manager menu, it will show list of the available items and then back to the purchase manager menu. The whole process same with the Heading 4.2.2.4.

### **4.3.3 List of Suppliers**

If you select the second option of the purchase manager menu, it will show list of available suppliers and then back to the purchase manager menu. The whole process same with the Heading 4.2.3.4.

### **4.3.4 Display Purchase Requisition**

If you choose the third option of the purchase manager menu, the system will show all existing purchase requisition regardless of its status and then back to the purchase manager menu.

### 4.3.5 Purchase Order

```
*****
Purchase Manager Menu
*****

1. List of Items.
2. List of Supplier.
3. Display Requisition.
4. Generate Purchase Order.
5. List of Purchaser Orders.
6. Update Info.
7. View Own Info.
8. Log out.

Enter your choice: 4

*****
Purchase Order Menu
*****

1. Add.
2. Edit.
3. Delete.
4. View.
5. Search.
6. Exit.

Enter your choice:
```

*Figure 99: Purchase Order Entry*

If you choose the fourth option of the purchase manager menu, the system will show the menu of the purchase order which enables you to exit the purchase order menu or take action on the purchase order, including generate, edit, delete, view, search and purchase order.

### 4.3.5.1 Create Purchase Order

```

*****
Purchase Order Menu
*****

1. Add.
2. Edit.
3. Delete.
4. View.
5. Search.
6. Exit.

Enter your choice:
1

|-----|
|               tan               |
|-----|
|   PR003       2023-09-19       Pending   |
|-----|
| Item Code    | Quantity    | Supplier Code |
| -----|-----|-----|
| ITEM003      | 90          | SP002        |
| -----|-----|-----|

|-----|
|               tan               |
|-----|
|   PR002       2023-09-18       Pending   |
|-----|
| Item Code    | Quantity    | Supplier Code |
| -----|-----|-----|
| ITEM001      | 90          | SP003        |
| -----|-----|-----|

Please enter PR code:
PR003

1.Approve    2.Reject    3.Exit

Enter your choice:
1

Congratulations! Purchase order added successfully!

Do you want to continue this section? (yes/no):
yes

```



```

|-----|
|               tan               |
|-----|
|   PR002       2023-09-18       Pending   |
|-----|
| Item Code    | Quantity    | Supplier Code |
|-----|-----|-----|
| ITEM001      | 90          | SP003        |
|-----|-----|-----|

Please enter PR code:
PR002

1.Approve    2.Reject    3.Exit

Enter your choice:
2

Why do u reject this purchase requisition:
The supplier has been deleted.

Do you want to continue this section? (yes/no):
no

```

*Figure 100: Create Purchase Order*

If you select the 1<sup>st</sup> option of the purchase order menu, the system will show which purchase requisition can be approved and then ask you to select which purchase order they would like to handle. If the purchase requisition has been approved by you, the system will show a message, then the purchase order will be generated and store in the file. Otherwise, it will ask you to leave a comment on why he/she reject this purchase requisition.

```

*****
                PO Code: P0001
*****
PR Code: PR003
Required Date: 2023-09-19
Approved Date: 2023-09-10
Approved By: soo
*****
                Item section as below:
*****
Item Code 1: ITEM003
Quantity : 90
*****

```

*Figure 101: Created PO After Approving*

-----		
tan		
-----		
PR003	2023-09-19	Approve
-----		
Item Code	Quantity	Supplier Code
-----	-----	-----
ITEM003	90	SP002
-----	-----	-----

-----		
tan		
-----		
PR002	2023-09-18	Reject
-----		
Item Code	Quantity	Supplier Code
-----	-----	-----
ITEM001	90	SP003
-----	-----	-----

*Figure 102: Handled Purchase Requisition*

### 4.3.5.2 Edit Purchase Order

```

*****
Purchase Order Menu
*****

1. Add.
2. Edit.
3. Delete.
4. View.
5. Search.
6. Exit.

Enter your choice:
2

Please provide PO code that you wish to edit:
PO001

*****
PO Code: PO001
*****
PR Code: PR003
Required Date: 2023-09-19
Approved Date: 2023-09-10
Approved By: soo
*****
Item section as below:
*****
Item Code 1: ITEM003
Quantity : 90
*****

```

*Figure 103: Edit Purchase Order*

If you select the 2<sup>nd</sup> option of the purchase order menu, it will prompt you to enter the purchase order code they wish to edit. After that, the system will show the details about the input purchase order code and then ask you to choose which item he/she wish to edit. The part of the chosen item also needs to be selected by you.

#### 4.3.5.2.1 Edit Item Code and Quantity

```

Please choose which item you want to edit (1-1):
1

Choose which field to edit within this PO line:
1. Item Code
2. Quantity
3. Required Date

Enter your choice:
1
Enter new Item Code:
ITEM005
Enter new Quantity:
80

Do you want to continue this section? (yes/no):

```

*Figure 104: Edit Item and Quantity*

After select edit item code, you will be prompted to enter the new item code and needed quantity of this item.

```

*****
                PO Code: PO001
*****
PR Code: PR003
Required Date: 2023-09-19
Approved Date: 2023-09-10
Approved By: soo
*****
                Item section as below:
*****
Item Code 1: ITEM005
Quantity : 80
*****

```

*Figure 105: Edited Purchase Order*

#### 4.3.5.2.2 Edit Quantity

```

Please choose which PO line to edit (1-2):
1

Choose which field to edit within this PO line:
1. Item Code
2. Quantity

Enter your choice:
2

Enter new Quantity:
100

Congratulations! The purchase order edited successfully!

```

*Figure 106: Edit Quantity*

After select edit quantity, you will be prompted to enter the quantity of the selected item.

```

*****
                PO Code: PO001
*****
PR Code: PR003
Required Date: 2023-09-19
Approved Date: 2023-09-10
Approved By: soo
*****
        Item section as below:
*****
Item Code 1: ITEM005
Quantity : 100
*****

```

*Figure 107: Edited Purchase Order*

#### 4.3.5.2.3 Edit Required Date

```
Please choose which PO line to edit (1-2):  
2
```

```
Enter new Required Date:  
2023-09-22
```

```
Congratulations! The purchase order edited successfully!
```

*Figure 108: Edit Required Date*

After select edit required date, you will be prompted to enter the required date of the purchase order.

```
*****  
PO Code: P0001  
*****  
PR Code: PR003  
Required Date: 2023-09-22  
Approved Date: 2023-09-10  
Approved By: soo  
*****  
Item section as below:  
*****  
Item Code 1: ITEM005  
Quantity : 100  
*****
```

*Figure 109: Edited Purchase Order*

### 4.3.5.3 Delete Purchase Order

```

*****
Purchase Order Menu
*****

1. Add.
2. Edit.
3. Delete.
4. View.
5. Search.
6. Exit.

Enter your choice:
3

Please enter purchase order code that you want to delete:
PO001

Congratulations! The purchase requisition deleted successfully!

Do you want to continue this section? (yes/no):
no

```

*Figure 110: Delete Purchase Order*

If you select the third option of the purchase order menu, it will prompt you to enter the purchase order code you wish to delete. After that, it will show a success message if the input purchase order code exists.

```

*****
Purchase Order Menu
*****

1. Add.
2. Edit.
3. Delete.
4. View.
5. Search.
6. Exit.

Enter your choice:
5

Please enter purchase order code or purchase requisition code you want to search:
PO001

No results found.

Do you want to continue this section? (yes/no): |
no

```

*Figure 111: Search whether the PO exists*

-----		
tan		
-----		
PR003	2023-09-19	Pending
-----		
Item Code	Quantity	Supplier Code
-----	-----	-----
ITEM003	100	SP002
-----	-----	-----

Figure 112: Purchase Requisition After Deleting Purchase Order

#### 4.3.5.4 View Purchase Order

If you select the 4<sup>th</sup> option of the purchase order menu, it will show list of the purchase orders. The whole process same with the Heading 4.2.6.

#### 4.3.5.5 Search Purchase Order

```
*****
Purchase Order Menu
*****

1. Add.
2. Edit.
3. Delete.
4. View.
5. Search.
6. Exit.

Enter your choice:
5

Please enter purchase order code or purchase requisition code you want to search:
PO001

No results found.

Do you want to continue this section? (yes/no): |
no
```

Figure 113: Search Purchase Order

If you select the 5<sup>th</sup> option of the purchase order menu, it will prompt you to enter the purchase order code or purchase requisition code of the purchase order you wish to search.



After that, it will show the result of the input code. If you enter a non-existent code, the system will display a message which is “No result found”.

#### **4.3.6 List of Purchase Order**

If you select the fifth option of the purchase manager menu, it will show list of the purchase orders. The whole process same with the Heading 4.2.6.

#### **4.3.7 Update Info**

If you select the sixth option of the purchase manager menu, it will enable you to update your own information. The whole process same with the Heading 4.2.7.

#### **4.3.8 View Own Info**

If you select the seventh option of the purchase manager menu, it will enable you to view his/her own information. The whole process same with the Heading 4.2.8.

#### **4.3.9 Log Out**

If you select the eighth option of the purchase manager menu, it will enable you to quit the system and return to the login page. The whole process same with the Heading 4.2.9.

## 4.4 Admin

### 4.4.1. Admin Menu

```
Please enter your username: admin

Please enter your password: admin

Successfully logged in! Welcome back, admin.

*****
Admin Menu
*****

1. Item Entry.
2. Supplier Entry.
3. Daily Item-wise Sales Entry.
4. Purchase Requisition Entry.
5. Purchase Order Entry.
6. Register New Staff.
7. Update Staff's Info.
8. Delete Existing Staff.
9. Search Existing Staff.
10. View own info.
11. Log out.

Enter your choice: |
```

*Figure 114: Admin Menu*

After login successfully Admin Account, it will show the admin menu. The Admin menu has these operations:

1. Item Entry
2. Supplier Entry
3. Daily Item-wise Sales Entry
4. Purchase Requisition Entry
5. Purchase Order Entry
6. Register New Staff
7. Update Staff's Info

8. Delete Existing Staff

9. Search Existing Staff

10. View Own Info

11. Log Out

#### **4.4.2. Item Entry**

If you select the 1<sup>st</sup> option of the admin menu, it will enable you to access functionality about item. The whole process same with the Heading 4.2.2.

#### **4.4.3. Supplier Entry**

If you select the 2<sup>nd</sup> option of the admin menu, it will enable you to access functionality about supplier. The whole process same with the Heading 4.2.3.

#### **4.4.4. Daily Item-wise Sales Entry**

If you select the 3<sup>rd</sup> option of the admin menu, it will enable you to access functionality about daily Item-wise sales. The whole process same with the Heading 4.2.4.

#### **4.4.5. Purchase Requisition**

If you select the 4<sup>th</sup> option of the admin menu, it will enable you to access functionality about purchase requisition. The whole process same with the Heading 4.2.5.

#### **4.4.6 Purchase Order Entry**

If you select the 5<sup>th</sup> option of the admin menu, it will enable you to access functionality about purchase order. The whole process same with the Heading 4.3.5.

#### 4.4.7. Register New Staff

```
*****
Admin Menu
*****

1. Item Entry.
2. Supplier Entry.
3. Daily Item-wise Sales Entry.
4. Purchase Requisition Entry.
5. Purchase Order Entry.
6. Register New Staff.
7. Update Staff's Info.
8. Delete Existing Staff.
9. Search Existing Staff.
10. View own info.
11. Log out.

Enter your choice: 6

Please enter name:
Tan Wee Wen
Please enter phone number:
0122222222
Please enter email:
tanweewen@email.com
Please enter username:
tan
Please enter password:
tan
Please enter confirm password:
tan

Select Role:
1. Sales Manager
2. Purchase Manager
3. Admin

Enter the role number (1/2/3): 1

Congratulations! The staff added successfully!
```

*Figure 115: Register New Staff*

If you select the 6<sup>th</sup> option of the admin menu, it will enable you to register a new staff by enter his/her personal information such as name, phone number, email address, username, and account password. After you input all information, it will show a success message.

```

*****
Username: tan
*****
Name: Tan Wee Wen
Phone Number: 0122222222
Email: tanweewen@email.com
Password: tan
Role: SM
*****

```

*Figure 116: Added Staff*

#### 4.4.8. Update Staff's Info

If you select the 7<sup>th</sup> option of the admin menu, it will enable you to update staff's or your own information. The rough process same with the Heading 4.2.7. However, in admin role, you are able to edit the staff's username and role, which are not available for sales manager and purchase manager.

#### 4.4.9. Delete Existing Staff

```

*****
Admin Menu
*****

1. Item Entry.
2. Supplier Entry.
3. Daily Item-wise Sales Entry.
4. Purchase Requisition Entry.
5. Purchase Order Entry.
6. Register New Staff.
7. Update Staff's Info.
8. Delete Existing Staff.
9. Search Existing Staff.
10. View own info.
11. Log out.

Enter your choice: 8

Please enter username of the staff you want to delete:
soo

Congratulations! The staff deleted successfully!

```

*Figure 117: Delete Existing Staff*

If you select the 8<sup>th</sup> option of the admin menu, it will enable you to delete a staff by enter his/her username. If you input an existing username, it will show a success message.

```
*****
Admin Menu
*****

1. Item Entry.
2. Supplier Entry.
3. Daily Item-wise Sales Entry.
4. Purchase Requisition Entry.
5. Purchase Order Entry.
6. Register New Staff.
7. Update Staff's Info.
8. Delete Existing Staff.
9. Search Existing Staff.
10. View own info.
11. Log out.

Enter your choice: 9

Please enter username or name you want to search:
soo

No results found.
```

*Figure 118: Search whether the staff exists*

#### 4.4.10 Search Existing Staff

```
*****
Admin Menu
*****

1. Item Entry.
2. Supplier Entry.
3. Daily Item-wise Sales Entry.
4. Purchase Requisition Entry.
5. Purchase Order Entry.
6. Register New Staff.
7. Update Staff's Info.
8. Delete Existing Staff.
9. Search Existing Staff.
10. View own info.
11. Log out.

Enter your choice: 9

Please enter username or name you want to search:
Tan Wee Wen
```

```
*****
      Username: tan
*****
Name: Tan Wee Wen
Phone Number: 0122222222
Email: tanweewen@email.com
Password: tan
Role: SM
*****
```

*Figure 119: Search Existing Staff*

If you select the 9<sup>th</sup> option of the admin menu, it will enable you to search a staff by enter his/her username. If you input an existing username, it will show all information of the staff.

#### **4.4.11 Log Out**

If you select the eleventh option of the admin menu, it will enable you to quit the system and return to the login page. The whole process same with the Heading 4.2.9.

## 4.5 Validation

### 4.5.1 User Authentication

```
*****
PURCHASE ORDER MANAGEMENT SYSTEM
*****

Please enter your username: low

Username doesn't exist! Please try again.

Please enter your username: admin

Please enter your password: nimda

Wrong password!

Please enter your username:
```

*Figure 120: Example of input wrong username or password*

User login credentials (username and password) validation is available for our system to ensure only authorized users can access the system. If you enter wrong username or password, it will display an error message and ask you to enter again.



## 4.5.2 Input Validation

```

*****
Item Menu
*****

1. Add.
2. Edit.
3. Delete.
4. View.
5. Search.
6. Exit.

Enter your choice:
1

1. Add new item.
2. Add stock of existing item.

Enter choice: 2

Please provide item code that you wish to add stock:
ITEM001
How much do you want to add:
gaq
Invalid input.
Please enter a valid integer for quantity:

```

Figure 121: Example of wrong numeric input when adding stock of item

```

Enter your choice:
1
Please enter its required date:
2023-09-32

Invalid date format. Please enter the date in yyyy-MM-dd format.
Enter:
2023-01-01

Date is earlier than the current date. Please try again.
Enter:
20230915

Invalid date format. Please enter the date in yyyy-MM-dd format.
Enter:

Invalid date format. Please enter the date in yyyy-MM-dd format.
Enter:
2023-09-15

```

Figure 122: Example of date validation when creating purchase requisition

Our system is able to check for valid data types (e.g., numeric, date) and ranges to prevent data entry errors.



### 4.5.3 Overstock Validation

```
*****
Daily Item-wise Sales Menu
*****

1. Generate.
2. Edit.
3. Delete.
4. View.
5. Search.
6. Exit

Enter your choice:
1
Please enter item code:
ITEM002
Please enter its quantity:
100

Input quantity is over stock.
Please enter again:
|
```

*Figure 123: Example of overstock validation when creating daily item-wise sales*

Our system is able to check whether the input quantity is over the stock to ensure that stock levels are updated correctly after sales.

#### 4.5.4 Existence Validation

```
*****
Supplier Menu
*****

1. Add.
2. Edit.
3. Delete.
4. View.
5. Search.
6. Exit.

Enter your choice:
2

Please provide supplier code that you wish to edit:
SP100

ID not Found.
Enter again:
```

*Figure 124: Example of enter non-existent ID when editing a supplier*

Our system can validate the existence of input ID for item, supplier, daily item-wise sales, purchase requisition, purchase order, and staff when editing, deleting, searching, referencing, or linking one entity to another within the system.

### 4.5.5: Choice Validation

```
*****
Admin Menu
*****

1. Item Entry.
2. Supplier Entry.
3. Daily Item-wise Sales Entry.
4. Purchase Requisition Entry.
5. Purchase Order Entry.
6. Register New Staff.
7. Update Staff's Info.
8. Delete Existing Staff.
9. Search Existing Staff.
10. View own info.
11. Log out.

Enter your choice: 0

Invalid choice. Please choose a valid option.
12

Invalid choice. Please choose a valid option.
2

*****
Supplier Menu
*****
```

Figure 125: Example of enter invalid choice

Our system is equipped to validate the existence of input choices whenever an operation selection is required.

## 5.0 Object Oriented Concept

### 5.1 Classes

```
public class Item extends WholeSystem implements Function{
    enum Category {
        Fresh_Food,
        Fresh_Produce,
        Grocery,
        INVALID
    }
    private final String PREFIX = "ITEM";
    FileHandle itfh = new FileHandle(path: "item.txt");
    FileHandle spfh = new FileHandle(path: "supplier.txt");
    private String itemCode;
    private String itemName;
    private int stock;
    private String supplierID;
    private Category category;
```

*Figure 126: class item*

In our program, classes such as "Staff", "FileHandle", "Item", etc. are utilized. Each class is analogous to a blueprint that defines a collection of objects with identical attributes, operations, relationships, and behaviours. These classes have the capacity to encapsulate attributes and methods pertaining to their respective entities. Furthermore, it simplifies complex systems by emphasising pertinent characteristics and suppressing other characteristics. This methodology promotes a modular and organised structure, thereby facilitating improved code management and maintainability.

## 5.2 Encapsulation

```
public class Person extends WholeSystem {
    private String name, phoneNumber, email;

    public Person(String name, String phoneNumber, String email) {
        this.name = name;
        this.phoneNumber = phoneNumber;
        this.email = email;
    }

    public Person() {
    }

    @Override
    public String toString() {
        return name + "," + phoneNumber + "," + email;
    }

    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }

    public String getPhoneNumber() {
        return phoneNumber;
    }

    public void setPhoneNumber(String phoneNumber) {
        this.phoneNumber = phoneNumber;
    }

    public String getEmail() {
        return email;
    }

    public void setEmail(String email) {
        this.email = email;
    }
}
```

Figure 127: Encapsulation example

Our code follows the encapsulation concept, which is another fundamental concept in object-oriented programming. This principle involves controlling direct access to some class members is done with access modifiers like "private". We effectively safeguard our data, fortify its security, and ensure its integrity. Anyone can interact with them by using the public

getter and setter method. This allows us can keep control over how data is viewed and changed. For instance, we can hide private data like username, password, and job, and only let authorized users access it through getters and setters in our "Staff" class.



## 5.3 Inheritance

```
public class SalesManager extends Staff{  
  
    public class Staff extends Person {  
  
        public class Person extends WholeSystem {
```

*Figure 128: four inheritance levels in our program*

The notion of inheritance is utilized to construct a hierarchical structure of classes, wherein classes such as Admin, SalesManager, and PurchaseManager inherit attributes and behaviors from a more general class called Staff. The presence of this inheritance relationship not only facilitates the process of code development by promoting the reuse of code but also guarantees the establishment of a consistent and structured class hierarchy. The utilization of inheritance enables each subclass to acquire shared functionality from its parent class, while also incorporating distinct attributes, hence promoting modularity and adaptability within the codebase. The enhancements and modifications implemented in the parent class are effectively transmitted to its child classes, hence augmenting the overall resilience of the software system. It is worth noting that some classes inherit attributes and methods from multiple parent classes. In our program, the “SalesManager” class is a subclass of the 'Staff' class, which is a subclass of the 'Person' class. Furthermore, the 'Person' class is a subclass of the 'WholeSystem' class. The degree of intricacy exhibited facilitates flexibility in delineating class associations and functionalities.

## 5.4 Polymorphism

```

public void menu() {
    while(true) {
        System.out.println(x: "\n*****");
        System.out.println(x: "Sales Manager Menu");
        System.out.println(x: "*****\n");
        System.out.println(x: "1.  Item Entry.");
        System.out.println(x: "2.  Supplier Entry.");
        System.out.println(x: "3.  Daily Item-wise Sales Entry.");
        System.out.println(x: "4.  Create Purchase Requisition.");
        System.out.println(x: "5.  Display Requisition.");
        System.out.println(x: "6.  List of Purchaser Orders.");
        System.out.println(x: "7.  Update Info.");
        System.out.println(x: "8.  View Own Info.");
        System.out.println(x: "9.  Log Out");
        System.out.print(s: "\nEnter your choice: ");
        int choice = validateChoice(minChoice:1,maxChoice:9);
        switch(choice) {
            case 1:
                it.menu();
                break;
            case 2:
                sp.menu();
                break;
            case 3:
                ds.menu();
                break;
            case 4:
                pr.menu();
                break;
            case 5:
                pr.view();
                break;
            case 6:
                po.view();
                break;
            case 7:
                Staff sf1 = new Staff(username: getUsername());
                sf1.updateInfo();
                break;
            case 8:
                Staff sf2 = new Staff(username: getUsername());
                sf2.updateInfo();
                break;
            case 9:
                System.out.println("\nSee you again," + getUsername() + ".\n");
                Staff sf3 = new Staff();
                sf3.login();
                break;
            default:
                System.out.println(x: "Please choose valid option!");
        }
    }
}

```

Figure 129: menu() in class SalesManager

```

public void menu() {
    while(true) {
        System.out.println(x: "\n*****");
        System.out.println(x: "Purchase Manager Menu");
        System.out.println(x: "*****\n");
        System.out.println(x: "1. List of Items.");
        System.out.println(x: "2. List of Supplier.");
        System.out.println(x: "3. Display Requisition.");
        System.out.println(x: "4. Generate Purchase Order.");
        System.out.println(x: "5. List of Purchaser Orders.");
        System.out.println(x: "6. Update Info.");
        System.out.println(x: "7. View Own Info.");
        System.out.println(x: "8. Log out.");
        System.out.print(s: "\nEnter your choice: ");
        int choice = validateChoice(minChoice:1,maxChoice:8);
        switch(choice) {
            case 1:
                it.view();
                break;
            case 2:
                sp.view();
                break;
            case 3:
                ds.view();
                break;
            case 4:
                po.menu();
                break;
            case 5:
                po.view();
                break;
            case 6:
                Staff sf1 = new Staff(username: getUsername());
                sf1.updateInfo();
            case 7:
                Staff sf2 = new Staff(username: getUsername());
                sf2.viewOwnInfo();
            case 8:
                System.out.println("\nSee you again," + getUsername() + ".\n");
                Staff sf3 = new Staff();
                sf3.login();
                break;
            default:
                break;
        }
    }
}

```

Figure 130: menu() in class PurchaseManager

```
public void menu() {
    while(true) {
        System.out.println(x: "\n*****");
        System.out.println(x: "Admin Menu");
        System.out.println(x: "*****\n");
        System.out.println(x: "1. Item Entry.");
        System.out.println(x: "2. Supplier Entry.");
        System.out.println(x: "3. Daily Item-wise Sales Entry.");
        System.out.println(x: "4. Purchase Requisition Entry.");
        System.out.println(x: "5. Purchase Order Entry.");
        System.out.println(x: "6. Register New Staff.");
        System.out.println(x: "7. Update Staff's Info.");
        System.out.println(x: "8. Delete Existing Staff.");
        System.out.println(x: "9. Search Existing Staff.");
        System.out.println(x: "10. View own info.");
        System.out.println(x: "11. Log out.");
        System.out.print(s: "\nEnter your choice: ");
        int choice = validateChoice(minChoice:1,maxChoice:11);
        switch(choice) {
            case 1:
                it.menu();
                break;
            case 2:
                sp.menu();
                break;
            case 3:
                ds.menu();
                break;
            case 4:
                pr.menu();
                break;
            case 5:
                po.menu();
                break;
            case 6:
                Staff sf1 = new Staff();
                sf1.addStaff();
                break;
            case 7:
                Staff sf2 = new Staff(username: getUsername());
                sf2.updateInfo();
                break;
            case 8:
                Staff sf3 = new Staff(username: getUsername());
                sf3.deleteStaff();
                break;
            case 9:
                Staff sf4 = new Staff();
                sf4.searchStaff();
                break;
            case 10:
                Staff sf5 = new Staff(username: getUsername());
                sf5.viewOwnInfo();
                break;
            case 11:
```

```
        System.out.println("\nSee you again," + getUsername() + ".\n");
        Staff sf6 = new Staff();
        sf6.login();
    default:
        break;
    }
}
```

Figure 131: menu() in class Admin

Polymorphism is a concept that involves the utilization of class inheritance to achieve the idea of "multiple forms" (*Polymorphism in Java - Javatpoint*, n.d.). Inheritance from a common superclass enables the dissemination of comparable attributes and approaches across multiple classes, hence facilitating their sharing. In our program, there is a superclass referred to as "Staff" which functions as a general representation of employees within our system. The subclasses inside the Staff class cover a range of responsibilities, including "Admin", "Sales Manager", and "Purchase Manager", each characterized by distinct implementations. This demonstrates the role of polymorphism in enabling the implementation of a unified task, such as displaying the menu in many ways depending on the specific staff function inside the system.

## 5.5 Abstraction

```
public abstract class WholeSystem {  
    Scanner sr = new Scanner(System.in);  
    public abstract void add();  
    public abstract void edit();  
    public abstract void delete();  
    public abstract void view();  
    public abstract void search();  
}
```

*Figure 132: abstraction in class WholeSystem*

In object-oriented programming, abstraction is the concept of hiding the complex implementation details of an object and showing only the necessary features of the object. Abstract classes and methods are utilized in order to attain abstraction. In our code, we have implemented an abstract class named "WholeSystem". Within this class, a set of abstract methods including add(), edit(), delete(), view(), and search() have been declared. The aforementioned methods are stated in the abstract class without including any implementation within the class itself. This requires that every specific subclass of WholeSystem must furnish implementations for these methods.

## 6.0 Extra Feature

### 6.1 isStockLessThan5()

Source Code

```
public void isStockLessThan5(){  
    for (String line : read()) {  
        String[] component = line.split(regex: ",");  
        if (Integer.parseInt(component[2]) < 5) {  
            System.out.println("\n!!!!!! The stock of " + component[0] + " is less than 5. !!!!!");  
        }  
    }  
}
```

Figure 133: Source Code of isStockLessThan5()

This feature is designed to remind sales managers to purchase items when the stock falls below 5. The message will be displayed after any sales manager logs in.

**Output:**

```
Please enter your password: tan  
  
Successfully logged in! Welcome back, tan.  
  
!!!!!! The stock of ITEM003 is less than 5. !!!!!  
  
!!!!!! The stock of ITEM004 is less than 5. !!!!!  
  
!!!!!! The stock of ITEM001 is less than 5. !!!!!  
  
.....
```

Figure 134: Output of isStockLessThan5()

## 6.2 PrNeedToApprove()

```
public void PrNeedToApprove() {  
    int count = 0;  
    for (String line : read()) {  
        String[] component = line.split(regex: ",");  
        if (component[6].equals(anObject: "Pending")) {  
            count++;  
        }  
    }  
    if(count != 0){  
        String linkingVerb;  
        if(count == 1){  
            linkingVerb = "is";  
        }  
        else{  
            linkingVerb = "are";  
        }  
        System.out.println(redColorCode + "\n!!!! There " + linkingVerb +  
            " " + Integer.toString(count) + " PR need to be handled !!!!!" + resetColorCode);  
    }  
}
```

Figure 135: Source Code of PrNeedToApprove()

The purpose of this feature is to serve as a reminder for Purchase Managers to assess and address any outstanding purchase requisitions. A notification will be presented by the system to prompt the purchase manager to either approve or reject requisitions that necessitate attention. This message is intended to be displayed to Purchase Managers upon their successful login to the system.



**Output:**

```
Please enter your username: soo

Please enter your password: soo

Successfully logged in! Welcome back, soo.

!!!!!! There are 2 PR need to be handled !!!!!

*****
Purchase Manager Menu
*****

1. List of Items.
2. List of Supplier.
3. Display Requisition.
4. Generate Purchase Order.
5. List of Purchaser Orders.
6. Update Info.
7. View Own Info.
8. Log out.

Enter your choice:
```

*Figure 136: Output of PrNeedToApprove()*

## 6.3 refreshPRstatus()

```

public void refreshPRstatus() {
    List<String> prData = prfh.read();
    for (int i = 0; i < prData.size(); i++) {
        String[] pr = prData.get(index: i).split(regex: ",");
        try {
            if(pr[6].equals(anObject: "Pending") && dateFormat.parse(pr[3]).compareTo(anotherDate: currentDate) < 0) {
                pr[6] = "Expired";
                prData.set(index: i, element: String.join(delimiter:",", elements: pr));
            }
        } catch (ParseException ex) {
            System.out.println(x: "\nThere are some problems while parsing the date.\n");
        }
    }
    prfh.overwrite(content: "");
    for (String dsCode : prData) {
        prfh.write(content: dsCode);
    }
}

```

Figure 137: Source Code of refreshPRstatus()

This feature is designed to update the status of purchase requisitions automatically when their required date has expired. It provides convenience for Purchase Managers, as they don't need to manually check for requisitions with expired dates, the system will handle this status update automatically.

### Output:

-----		
tan		
-----		
PR004	2023-09-08	Expired
-----		
Item Code	Quantity	Supplier Code
-----		
ITEM001	15	SP002
-----		
ITEM002	8	SP002
-----		

Figure 138: Output of refreshPRstatus()

## 6.4 UpdateInfo()

```

public void updateInfo() {
    String Username = getUsername();
    String idToEdit;
    String content = null;
    int count = 0;
    int choice;
    if (Username.equals(anObject: "admin")) {
        count = 1;
    }
    if(count == 1) {
        System.out.println(x: ""
            \n\u001b[34mHint: Enter ur username if you would like to edit yourself.\u001b[0m
            Please provide username of the staff that you wish to update: "");
        idToEdit = sr.nextLine();
    }
    else{
        idToEdit = getUsername();
    }
    String[] staffToUpdate = sfh.isIdExists(id: idToEdit);
    Staff sf = new Staff();
    if (staffToUpdate != null) {
        System.out.println(x: "\nPlease choose which component.");
        System.out.println(x: "1. Name");
        System.out.println(x: "2. Phone Number");
        System.out.println(x: "3. Email address");
        System.out.println(x: "4. Password");
        if (count == 1) {
            System.out.println(x: "5. Role");
            System.out.println(x: "6. Username");
        }
        System.out.println(x: "\nEnter your choice: ");
        if(count == 1){
            choice = validateChoice(minChoice:1,maxChoice:6);
        }
        else{
            choice = validateChoice(minChoice:1,maxChoice:4);
        }
        if(choice == 5){
            sf.selectRole();
        }
        else{
            System.out.println(x: "\nEnter changed content.");
            content = sr.nextLine();
        }
        if (choice == 1) {
            staffToUpdate[1] = content;
        } else if (choice == 2) {
            staffToUpdate[2] = content;
        } else if (choice == 3) {
            staffToUpdate[3] = content;
        } else if (choice == 4) {
            staffToUpdate[4] = content;
        } else if (choice == 5 && count == 1) {
            UserRole userRole = sf.getRole();
            String userRoleString = userRole.toString();
            staffToUpdate[5] = userRoleString;
        } else if (choice == 6 && count == 1) {
            staffToUpdate[0] = content;
        } else {
            System.out.println(x: "\nInvalid choice.\n");
        }
        sf.setUsername(staffToUpdate[0]);
        sf.setName(staffToUpdate[1]);
        sf.setPhoneNumber(staffToUpdate[2]);
        sf.setEmail(staffToUpdate[3]);
        sf.setPassword(staffToUpdate[4]);
        sf.setRole(role:UserRole.valueOf(staffToUpdate[5]));
        sfh.edit(unicode: idToEdit, content: sf.toString());
        System.out.println(x: "\nCongratulations! Staff information updated successfully!\n");
    }
}

```

Figure 139: Source Code of UpdateInfo()

This feature is designed to allow any staff account to update their information as needed. Additionally, it permits the admin to change the unique username associated with an account, providing flexibility in managing user data and access.

**Output:**

```
7.  Update Info.
8.  View Own Info.
9.  Log Out

Enter your choice: 7

Please choose which component.
1. Name
2. Phone Number
3. Email address
4. Password
```

*Figure 140: Output of UpdateInfo()*

## 6.5 deleteStaff()

```
public void deleteStaff(){
    System.out.println(x: "\nPlease enter username of the staff you want to delete: ");
    String[] usernameToDelete = sfh.isIdExists(id: sr.nextLine());
    System.out.println(x: "\nCongratulations! The staff deleted successfully!\n");
    sfh.delete(usernameToDelete[0]);
}
```

Figure 141: Source Code of deleteStaff()

This feature is designed for administrators to remove employees who are no longer employed by the company. Its purpose is to maintain data accuracy and the overall efficiency of the system by ensuring that there is no unnecessary or outdated information in the database.

### Output:

```
*****
Admin Menu
*****

1. Item Entry.
2. Supplier Entry.
3. Daily Item-wise Sales Entry.
4. Purchase Requisition Entry.
5. Purchase Order Entry.
6. Register New Staff.
7. Update Staff's Info.
8. Delete Existing Staff.
9. Search Existing Staff.
10. View own info.
11. Log out.

Enter your choice: 8

Please enter username of the staff you want to delete:
soo

Congratulations! The staff deleted successfully!
```

Figure 142: Output of deleteStaff()

## 6.6 Search()

```

public void searchStaff() {
    System.out.println(x: "\nPlease enter username or name you want to search: ");
    String[] output = sffh.showSearchResult();
    if (output != null) {
        System.out.println(x: "\n\n*****");
        System.out.println("    Username: " + output[0]);
        System.out.println(x: "*****");
        System.out.println("Name: " + output[1]);
        System.out.println("Phone Number: " + output[2]);
        System.out.println("Email: " + output[3]);
        System.out.println("Password: " + output[4]);
        System.out.println("Role: " + output[5]);
        System.out.println(x: "*****");
    }
    else{
        System.out.println(x: "\nNo results found.\n");
    }
}

```

Figure 143: Search fuction for staff searching

```

@Override
public void search() {
    System.out.println(x: "\nPlease enter item code or name you want to search: ");
    String[] output = itfh.showSearchResult();
    if (output != null) {
        System.out.println("\nItem code: " + output[0] + "\nName: " + output[1]
            + "\nStock: " + output[2] + "\nSupplier code: " + output[3] + "\ncatgeory: " + output[4] + "\n");
    }
    else{
        System.out.println(x: "\nNo results found.\n");
    }
}

```

Figure 144: Search Function for Item searching

```

public void search() {
    System.out.println(x: "\nPlease enter supplier code or name you want to search: ");
    String[] output = spfh.showSearchResult();
    if (output != null) {
        System.out.println("\nSupplier code: " + output[0] + "\nName: " + output[1]
            + "\nPhone number: " + output[2] + "\nEmail: " + output[3]
            + "\nAddress: " + output[4] + "\n");
        showSuppliedItem(output[0]);
    }
    else{
        System.out.println(x: "\nNo results found.\n");
    }
}

```

Figure 145: Search Function for supplier searching

```

public void search() {
    System.out.println(x: "\nPlease enter daily sales id you want to search: ");
    String searchCode = sr.nextLine();
    List<String> searchResults = dsfh.searchMultipleLine(idToSearch: searchCode);
    showResult(searchResults);
}

```

Figure 146: Search Function for daily sales searching

```

@Override
public void search() {
    System.out.println(x: "\nPlease enter purchase requisition id you want to search: ");
    String searchCode = sr.nextLine();
    List<String> searchResults = prfh.searchMultipleLine(idToSearch: searchCode);
    showResult(lines: searchResults);
}

```

Figure 147: Search Function for Purchase Requisition Searching

```

@Override
public void search(){
    System.out.println(x: "\nPlease enter purchase order code or purchase requisition code you want to search: ");
    String[] output = pofh.showSearchResult();
    if (output != null) {
        System.out.println("\npurchase order code: " + output[0]
            + "\nHandled purchase requisition code: " + output[1] + "\nDate: "
            + output[2] + "\nHandled By: " + output[3] + "\nStatus: " + output[4] + "\n");
    }
    else{
        System.out.println(x: "\nNo results found.\n");
    }
}

```

Figure 148: Search Function for Purchase Order Searching

This feature is designed to provide convenience to users when they need to retrieve specific information using a unique ID.

## 6.7 Comment

```
public class Comment {
    private String prCode;
    private String comment;
    Scanner sr = new Scanner(System.in);
    FileHandle cmfh = new FileHandle(path: "comment.txt");

    @Override
    public String toString() {
        return prCode + "|" + comment;
    }

    public String getPrCode() {
        return prCode;
    }

    public void setPrCode(String prCode) {
        this.prCode = prCode;
    }

    public String getComment() {
        return comment;
    }

    public void setComment(String comment) {
        this.comment = comment;
    }

    public Comment(String prCode) {
        this.prCode = prCode;
    }

    public void writeComment() {
        System.out.println(x: "\nWhy do u reject this purchase requisition: ");
        setComment(comment: sr.nextLine());
        cmfh.write(content: toString());
    }
}
```

This feature is intended to provide comments or justifications when a Purchase Manager rejects an order request. Its purpose is to inform Sales Managers about the reasons for the rejection, ensuring the process is transparency and clarity.



**Output:**

tan		
PR002	2023-09-18	Pending
Item Code	Quantity	Supplier Code
ITEM006	90	SP003

Please enter PR code:

PR002

1.Approve    2.Reject    3.Exit

Enter your choice:

2

Why do u reject this purchase requisition:

The quantity of ITEM006 is too many.

Do you want to continue this section? (yes/no):

NO

Figure 149: Reject PR002 and Give the commentt

tan		
PR002	2023-09-18	Reject
Item Code	Quantity	Supplier Code
ITEM006	90	SP003

Rejected By: admin

Reason: The quantity of ITEM006 is too many.

Figure 150: Output of Sales Manager viewing Purchase Requisition

## 7.0 Conclusion

The system facilitates the login process for several roles, wherein each role is issued a unique username by the administrator. This ensures that users can only access capabilities that are pertinent to their designated role inside the system. Conversely, it restricts access to individuals who lack authorization.

In addition, our system provides a diverse array of actions and functionalities that enable users to effectively oversee multiple facets of the system. These include the ability to add, amend, delete, view, and search for products, suppliers, daily item-wise sales, purchase requisitions, and purchase orders. The actions and features provided enable users to efficiently oversee inventories, suppliers, and procurement processes within the system, serving as the fundamental framework for their operational activities. The incorporation of functionalities such as data addition, editing, deletion, viewing, and searching contributes to increased efficiency and facilitates data-informed decision-making, therefore serving as a crucial factor in the wholesale distributor's journey towards achieving success.

Administrators possess the authority to oversee and regulate personnel within the system, encompassing the tasks of introducing new staff members, modifying staff information, and eradicating staff records. In our system, the storage of invalid input in the file is prevented due to the presence of numerous validations. These validations serve the purpose of safeguarding against the entry of inaccurate information by the user.

Lastly, individuals can securely conclude their sessions by logging out, discontinuing access to their accounts, and thwarting any unwanted usage. This process plays a crucial role in upholding the system's highest security measures and user access control.

## 8.0 References

*Polymorphism in Java - javatpoint.* (n.d.). [www.javatpoint.com](http://www.javatpoint.com).

<https://www.javatpoint.com/runtime-polymorphism-in-java>