TSE2101
Final Report

for

Restaurant Management System

Version 3.0

Group Members:

Ng Kang Jie (Team Leader)	**1142700809**
Choo Jia Sheng	**1142700814**
Liew Soon Pang	**1142700808**
Nicholas Tan Yu Zhe	**1142701655**

Date of Submission: **23 September 2016**

Contents

CONTENTS **2**

REVISIONS **4**

1 PROJECT MANAGEMENT **5**

1.1 TEAM MEMBERS 5
1.2 PROJECT PLAN 5

2 SYSTEM OVERVIEW **7**

2.1 DESCRIPTION 7
2.2 ACTORS 7
2.3 ASSUMPTIONS AND DEPENDENCIES 7
2.4 USE CASE DIAGRAM 8

3 BASIC REQUIREMENTS **9**

3.1 ACTOR 1 9
3.2 ACTOR 2 19
3.3 ACTOR 3 29
3.4 ACTOR 4 37

4 SPECIFIC REQUIREMENTS **45**

4.1 CLASS DIAGRAMS 45
4.2 SEQUENCE DIAGRAMS 46

5 DATA DESIGN **58**

5.1 DATA DICTIONARY 58

6 ARCHITECTURE DESIGN **59**

6.1 SOFTWARE ARCHITECTURE 59
6.1.1 SUBSYSTEM 1: COUNTER SYSTEM 60
6.1.2 SUBSYSTEM 2: KITCHEN SYSTEM 61

7 INTERFACE DESIGN **62**

7.1 MAIN SCREENS 62
7.2 SUBSYSTEM 1 SCREENS 63
7.3 SUBSYSTEM 2 SCREENS 70

8 COMPONENT DESIGN **71**

8.1 MAIN COMPONENTS 71
8.1.1 COLLECT PAYMENT (COUNTER SYSTEM) 72
8.1.2 ENTER ORDER (COUNTER SYSTEM) 74
8.1.3 GENERATE REPORT (COUNTER SYSTEM) 75
8.1.4 PROCESS CANCELLATION (COUNTER SYSTEM) 76
8.1.5 PROCESS VOID (COUNTER SYSTEM) 77
8.1.6 ACCEPT ITEM (KITCHEN SYSTEM) 78
8.1.7 VOID ITEM (KITCHEN SYSTEM) 79
8.1.8 TERMINATE ITEM (KITCHEN SYSTEM) 80

9 DEPLOYMENT DESIGN **81**

9.1 DEPLOYMENT DIAGRAM 81
9.2 DESIGN 82

10 TEST DATA 83

10.1 LOGIN SCREEN 83
10.2 ADD NEW ITEM TO ORDER LIST 83
10.3 CANCEL ITEM FROM ORDER LIST 83
10.4 ADD ITEM TO MENU 83
10.5 DELETE ITEM FROM MENU 83
10.6 CASH PAYMENT 84

11 ACCEPTANCE TEST 85

13.1 CASE 1 85
13.2 CASE 2 85
13.3 CASE 3 85
13.4 CASE 4 86
13.5 CASE 5 86
13.6 CASE 6 87
13.7 CASE 7 87

12 SAMPLE SCREENS 88

14.1 COUNTER SYSTEM (MANAGER) 88

13 CONCLUSION 98

15.1 SUMMARY OF RESULTS 98
15.2 PROBLEMS ENCOUNTERED 98
15.3 LIMITATIONS AND FUTURE ENHANCEMENTS 98

Revisions

Version	Primary Author(s)	Description of Version	Date Completed
Version 1.0	Choo Jia Sheng Liew Soon Pang Ng Kang Jie Nicholas Tan Yu Zhe	Updated the documentations of assignment 1 and assignment 2.	21/09/16
Version 2.0	Choo Jia Sheng Liew Soon Pang Ng Kang Jie Nicholas Tan Yu Zhe	Added the additional parts required in final report.	22/09/16
Version 3.0	Choo Jia Sheng Liew Soon Pang Ng Kang Jie Nicholas Tan Yu Zhe	Standardized and checked the formatting of the documentation.	23/09/16

1 Project Management

1.1 Team Members

Team Leader:	Ng Kang Jie (Manager)
Programming Leader:	Nicholas Tan Yu Zhe (Customer)
Document Manager:	Liew Soon Pang (Waiter)
Quality Manager:	Choo Jia Sheng (Chef)

1.2 Project Plan

The software process model that we utilize is waterfall model. It is a systematic, sequential and classic life cycle. One phase has to be completed before proceeding to the next phase. The particular model has a restriction that it is difficult to answer to the changes from customer's requirements. The main activities for the part one of the project are as follows: -

- Determine the appropriate software process model
- Plan the project activities
- Identify major processes to be executed by the system
- Determine actors and their use cases/ functions
- List the assumptions and dependencies
- Construct use case diagram
- Describe use case for each actor and construct the Swimlane diagram
- Describe the classes and construct the class diagram
- Describe the sequence and construct the sequence diagram

The Gantt chart for the whole project is attached in the following page.

Gantt chart:

2 System Overview

This section is prepared by all the members together.

2.1 Description

The implementation of the Restaurant Management System is to bring a better and more fluent service flow. A customer will have a better dining experience and a staff will perform the tasks better. The manager can also utilize the system to generate sales reports. The intended users of the product are chef, customer, manager and waiter.

The system consists of Point of Sales (POS) system. The POS system is to enable the staffs to perform the daily activities, for instance, ordering of food and beverage, transmission of tasks to the kitchen and also computation of total bill. It could lower the service time and increase the order accuracy.

2.2 Actors

Manager: Enter order, process cancellation, process void, collect payment, generate report.
Customer: Make payment, drop order, make order, consume item, item dropped.
Waiter: Take order, serve item, facilitate cancellation, facilitate void.
Chef: Accept order, terminate item, void item, prepare item.

2.3 Assumptions and Dependencies

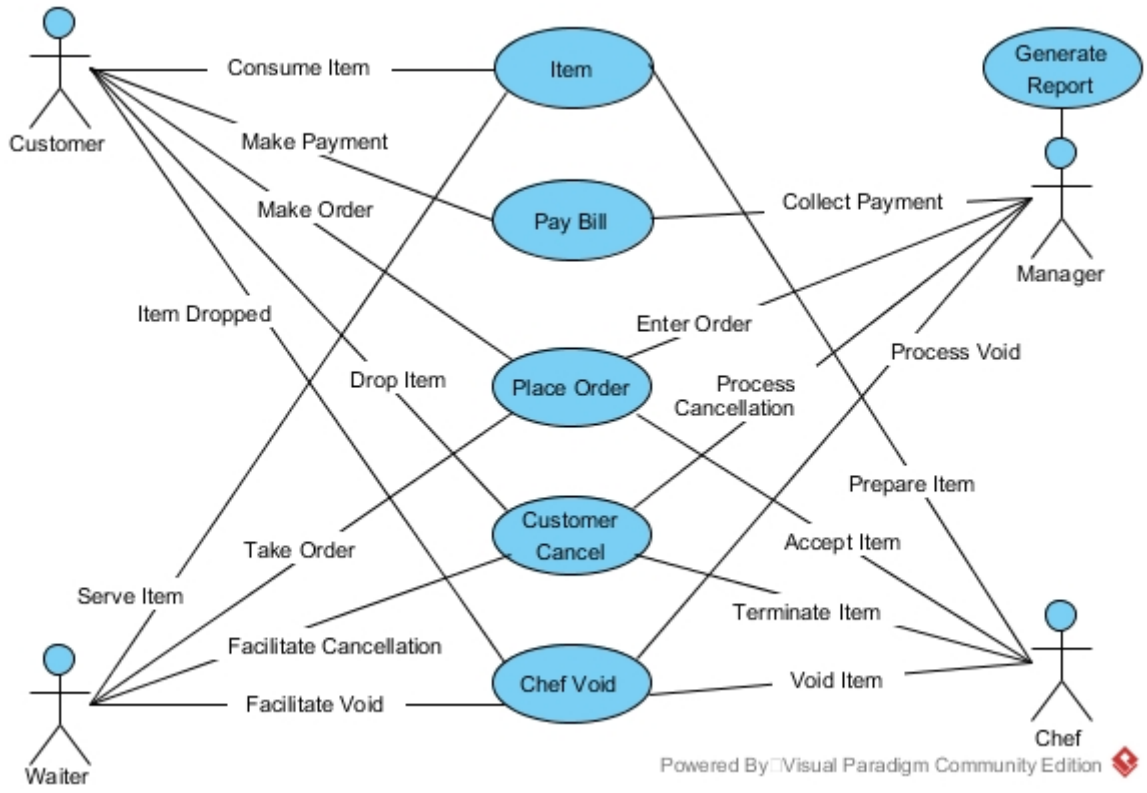
Assumptions:

1. The restaurant is a Japanese cuisine restaurant.
2. The staffs are able to communicate with the local communities fluently and can understand English.
3. The restaurant is a small restaurant that could only fit up to 24 tables.
4. The restaurant allows cash payment and card payment only.
5. The restaurant only accepts Ringgit Malaysia (RM).
6. The restaurant does not charge Goods and Services Tax (GST) and service tax.

Dependencies:

1. Stable electric supply is required for the system to run.
2. Maintenance will be performed quarterly on the system during non-operating hours. System downtime is expected to be minimal.

2.4 Use Case Diagram



3 Basic Requirements

3.1 Actor 1: Manager (Ng Kang Jie)

3.1.1 Use Case 1: Enter Order

Primary Actor: Manager

Description: Manager needs to enter order into the system.

Pre-conditions: Waiter had taken the order from customer.

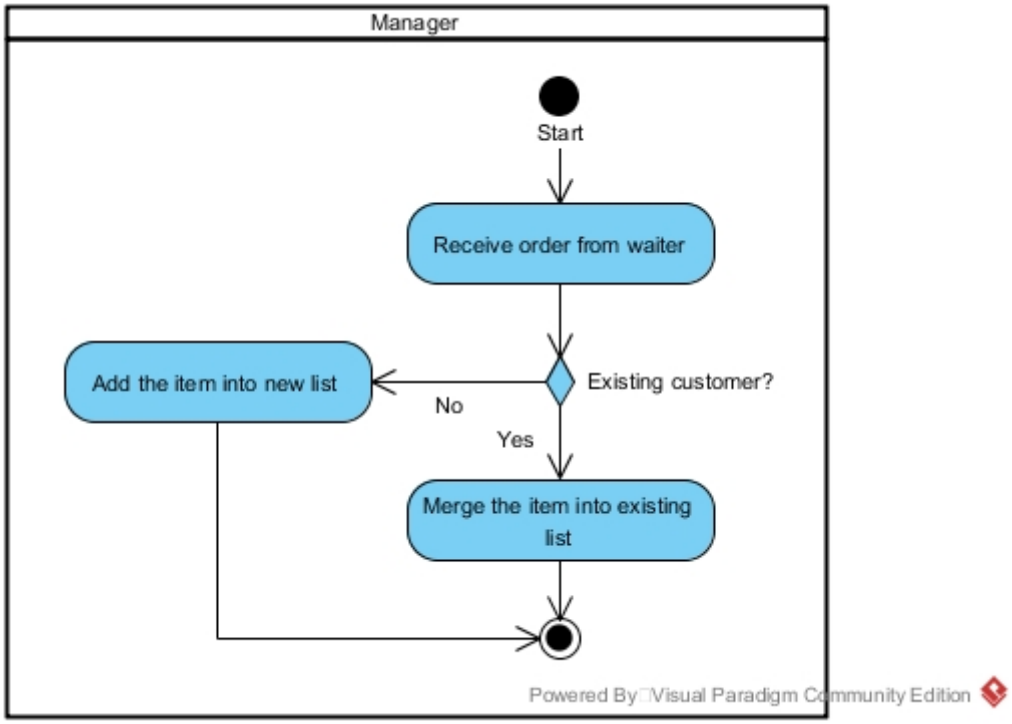
Flow of Events:
1. Manager receives order from waiter.
2. Manager checks if it is the existing customer.
3. Manager merges the item into existing list.

Alternative Flow of Events:
1. The order is from a new customer. The manager adds the item into a new list.

Post-condition: The chef is being notified about the order.

Assumption: The manager is familiar with the system. The manager has logged in into the system before the restaurant starts its operation.

Swimlane diagram for use case 1 is as below:



3.1.2 Use Case 2: Process Cancellation

Primary Actor: Manager

Description: Customer wants to cancel an item.

Pre-conditions: The item is not served yet.

Flow of Events:

1. Waiter facilitates the cancellation.
2. Manager asks chef the preparation status.
3. Chef checks the preparation status.
4. Chef updates manager on the preparation status.
5. Manager cancels the unprepared item from the system.
6. Chef is being notified about the cancellation.
7. Manager updates the waiter on the cancellation status.

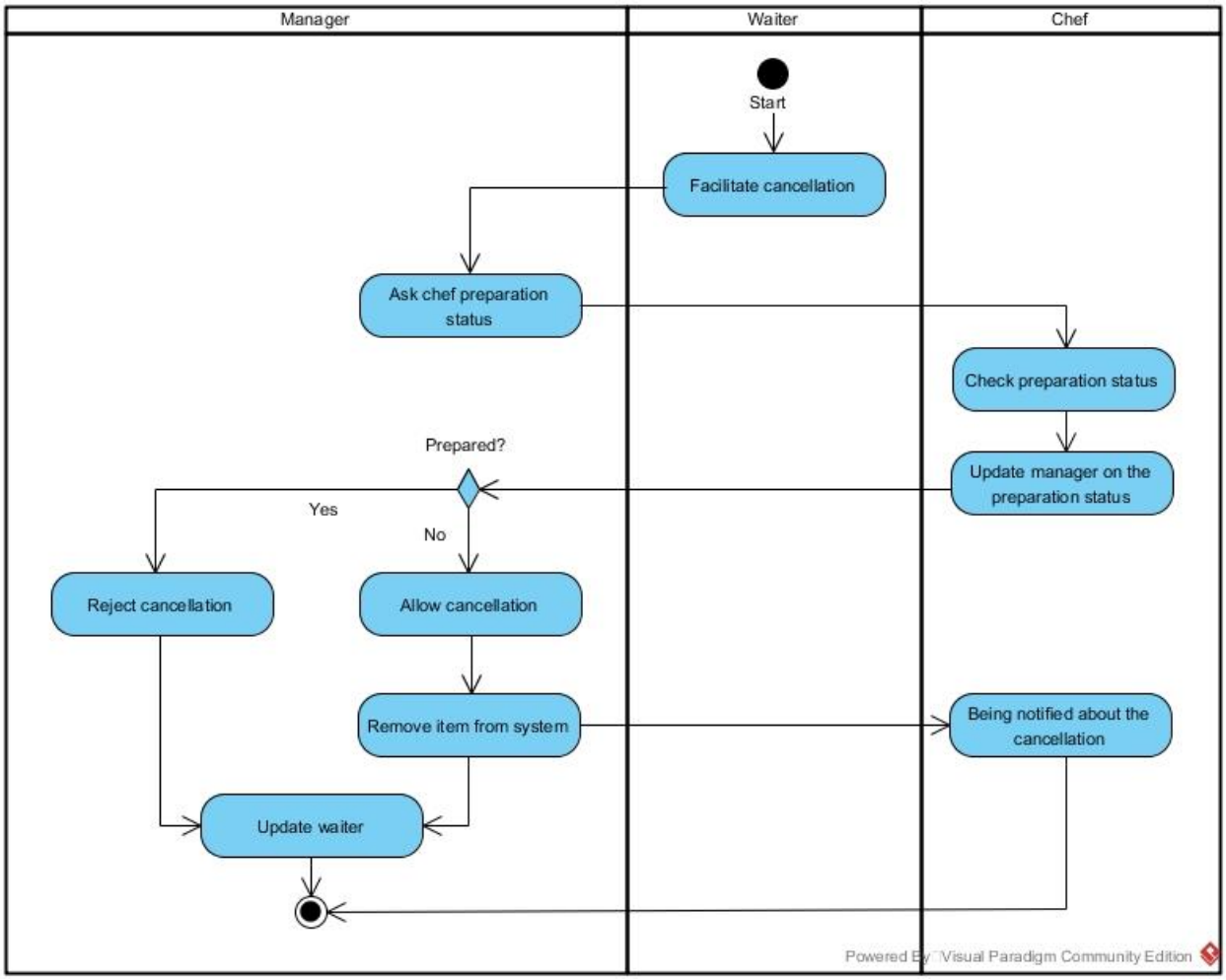
Alternative Flow of Events:

1. The cancellation is rejected because the item had been prepared.

Post-condition: Waiter updates the customer on the cancellation status.

Assumption: The manager is familiar with the system. The manager has logged in into the system before the restaurant starts its operation.

Swimlane diagram for use case 2 is as below:



3.1.3 Use Case 3: Process Void

Primary Actor: Manager

Description: Chef wants to void an item.

Pre-conditions: The ingredient is insufficient.

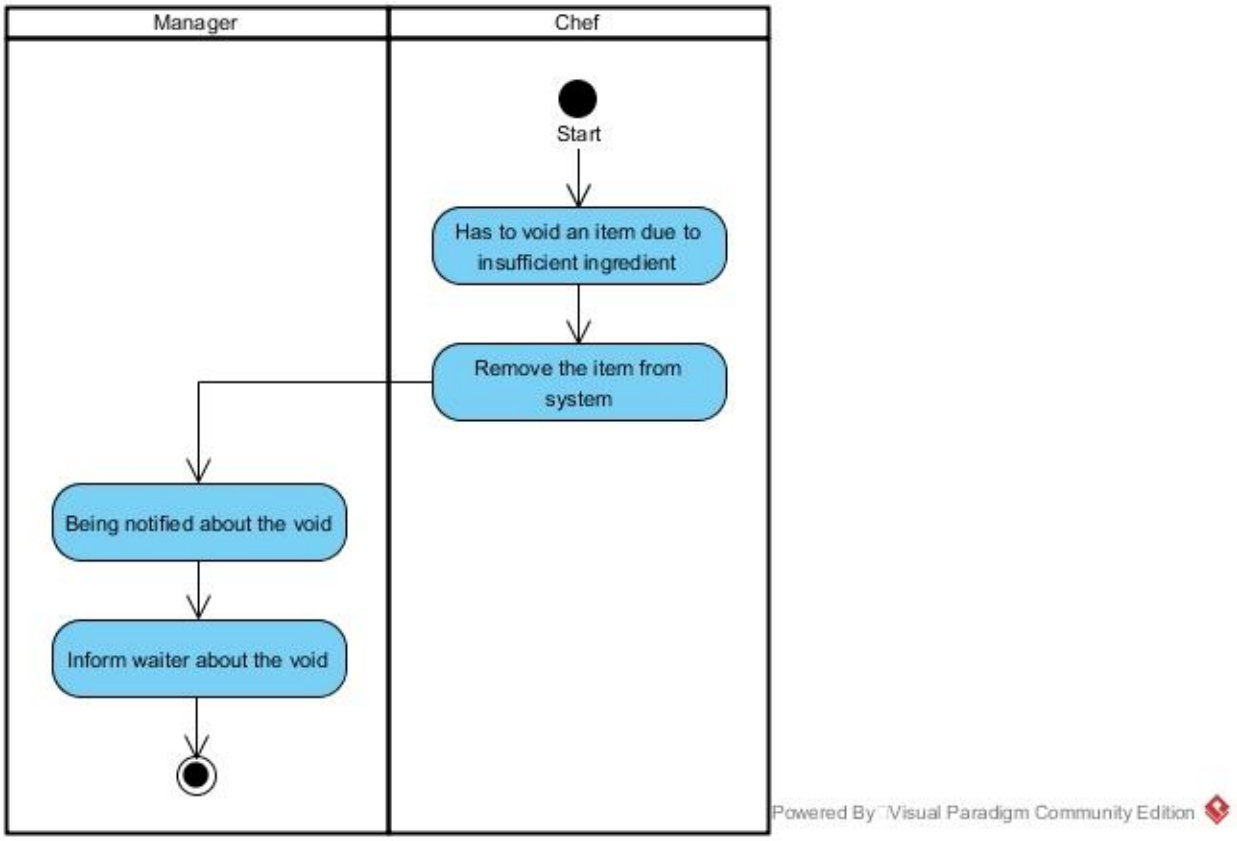
Flow of Events:

1. Chef has to void an item due to insufficient ingredient.
2. Chef removes the item from system.
3. Manager is being notified about the void.
4. Manager informs waiter about the void.

Post-condition: The waiter updates customer about the void.

Assumption: The manager is familiar with the system. The manager has logged in into the system before the restaurant starts its operation.

Swimlane diagram for use case 3 is as below:



3.1.4 Use Case 4: Collect Payment

Primary Actor: Manager

Description: Manager collects payment from the customer.

Pre-conditions: Customer wants to make payment.

Flow of Events:

1. Customer goes to the counter to make payment.
2. Manager selects the customer's bill.
3. Manager prints out bill.
4. Customer checks the bill.
5. Customer selects payment method.
6. Customer chooses to pay by cash.
7. Manager collects cash.
8. Manager returns the remaining balance and receipt.

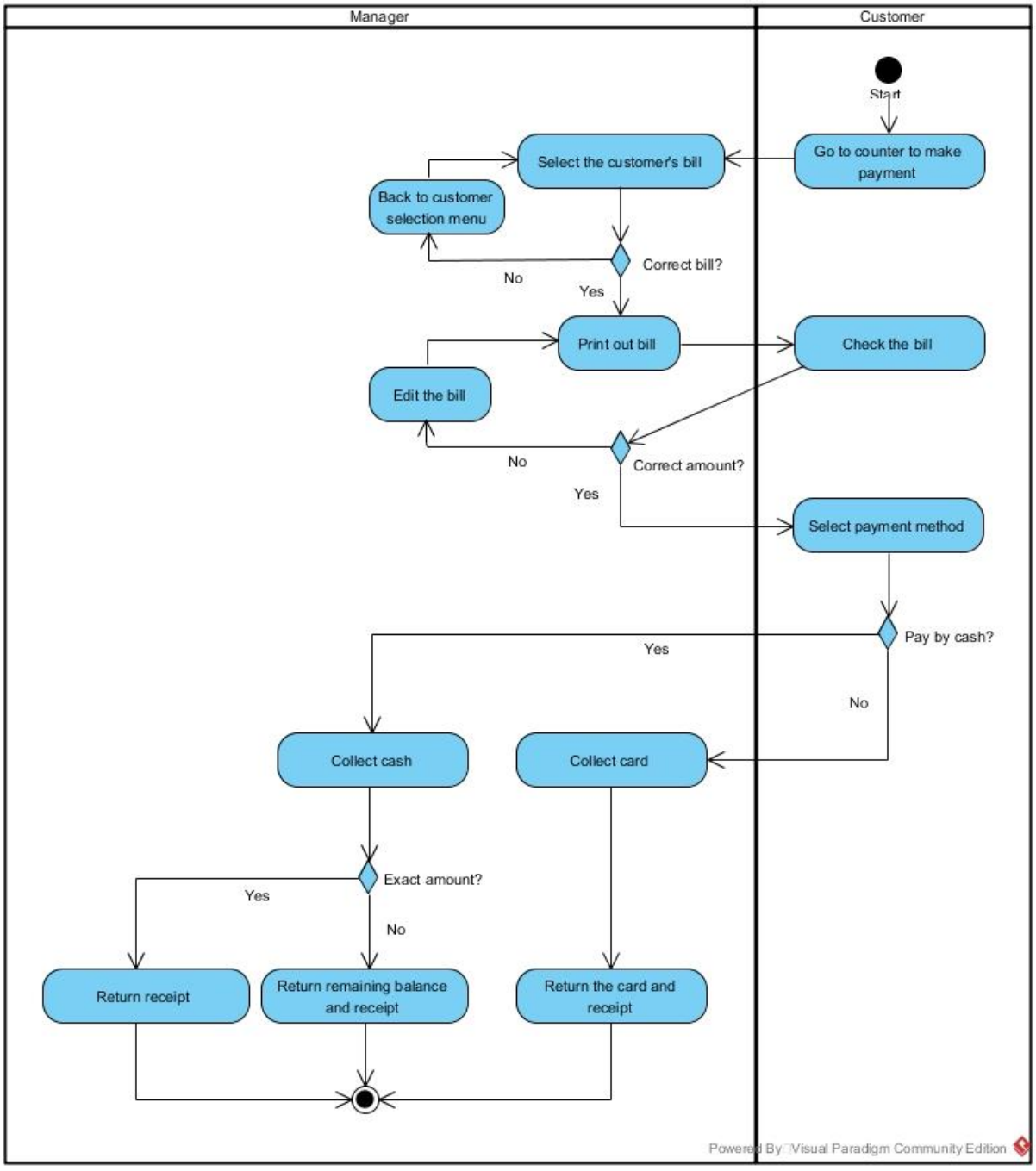
Alternative Flow of Events:

1. Manager selects the wrong customer and wants to go back to the customer selection menu.
2. Customer realizes that the invoice is incorrect and user needs to edit the bill.
3. Customer gives exact amount. Manager returns the receipt.
4. Customer chooses to pay by card.
5. Manager collects card.
6. Manager returns the card and receipt.

Post-condition: The sales is recorded in the system.

Assumption: The manager is familiar with the system. The manager has logged in into the system before the restaurant starts its operation. The manager has sufficient change.

Swimlane diagram for use case 4 is as below:



3.1.5 Use Case 5: Generate Report

Primary Actor: Manager

Description: Manager wants to generate sales report.

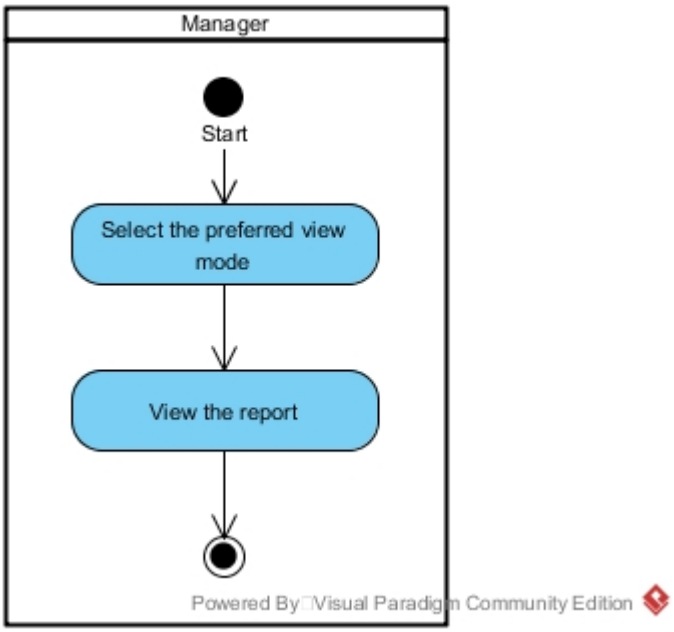
Pre-conditions: There is at least one sales record in the system.

Flow of Events:
1. Manager selects the preferred view mode.
2. Manager views the sales report.

Post-condition: The manager is aware of the sales performance.

Assumption: The manager is familiar with the system. The manager has logged in into the system before the restaurant starts its operation.

Swimlane diagram for use case 5 is as below:



3.2 Actor 2: Customer (Nicholas Tan Yu Zhe)

3.2.1 Use Case 6: Make Payment

Primary Actor: Customer

Description: Customer wants to pay the bill.

Pre-conditions: Customer had finished eating.

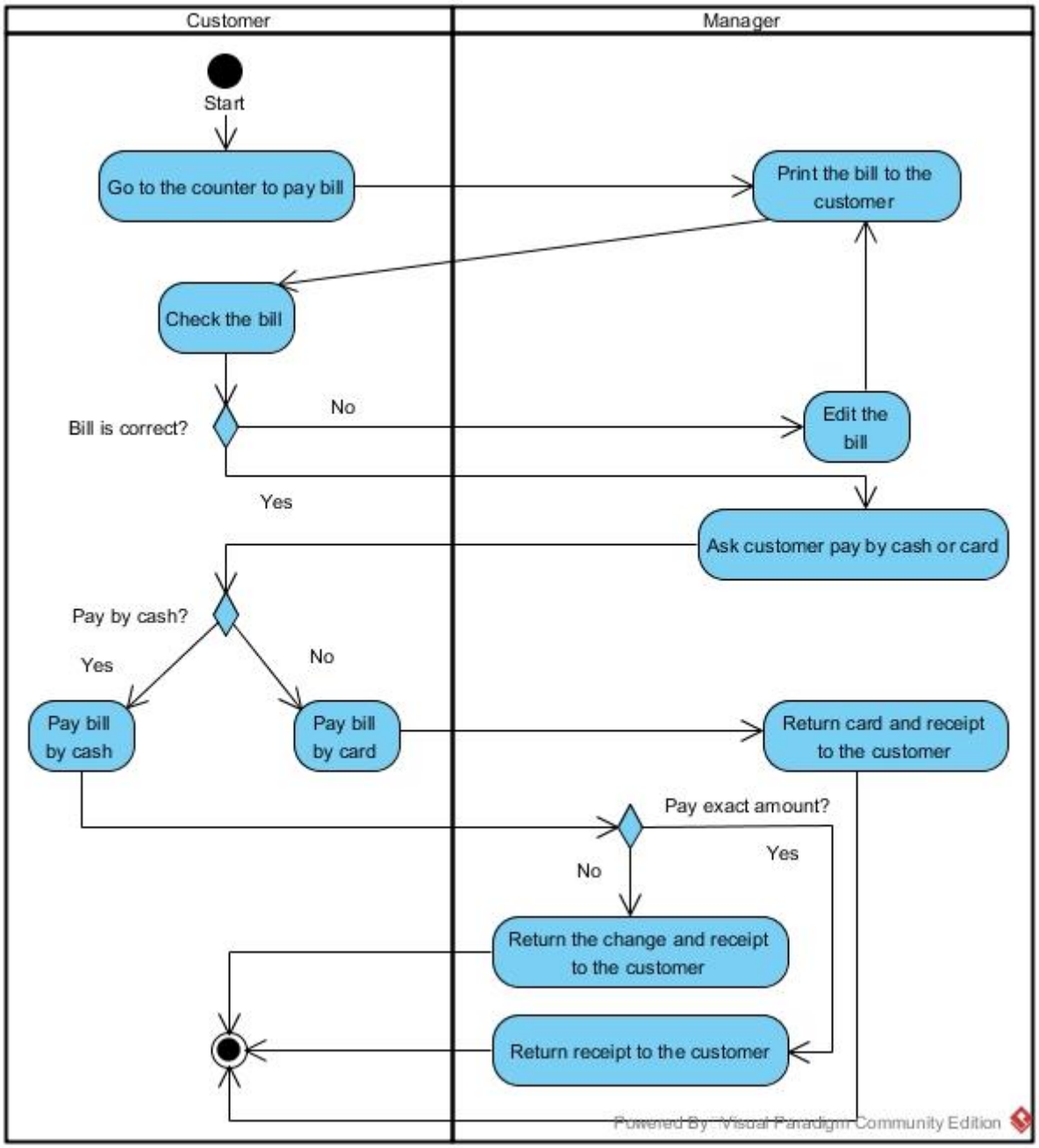
Flow of Events:
1. Customer goes to the counter to pay bill.
2. Manager prints the bill for the customer.
3. Customer checks the bill.
4. Manager asks customer whether to pay by cash or card.
5. Customer pays the bill by cash.
6. Manager returns the change and receipt to customer if the customer paid by cash.

Alternative Flow of Events:
1. Bill is wrong and manager edits the bill.
2. Customer pays the bill by card.
3. Customer pays the exact amount so manager only returns the receipt.
4. Manager returns the card and receipt to customer if the customer paid by card.

Post-condition: Customer leaves the restaurant.

Assumption: Manager has enough change.

Swimlane diagram for use case 6 is as below:



3.2.2 Use Case 7: Drop Item

Primary Actor: Customer

Description: Customer calls the waiter and cancels the item.

Pre-conditions: Customer realised that he/ she ordered the wrong item.

Flow of Events:

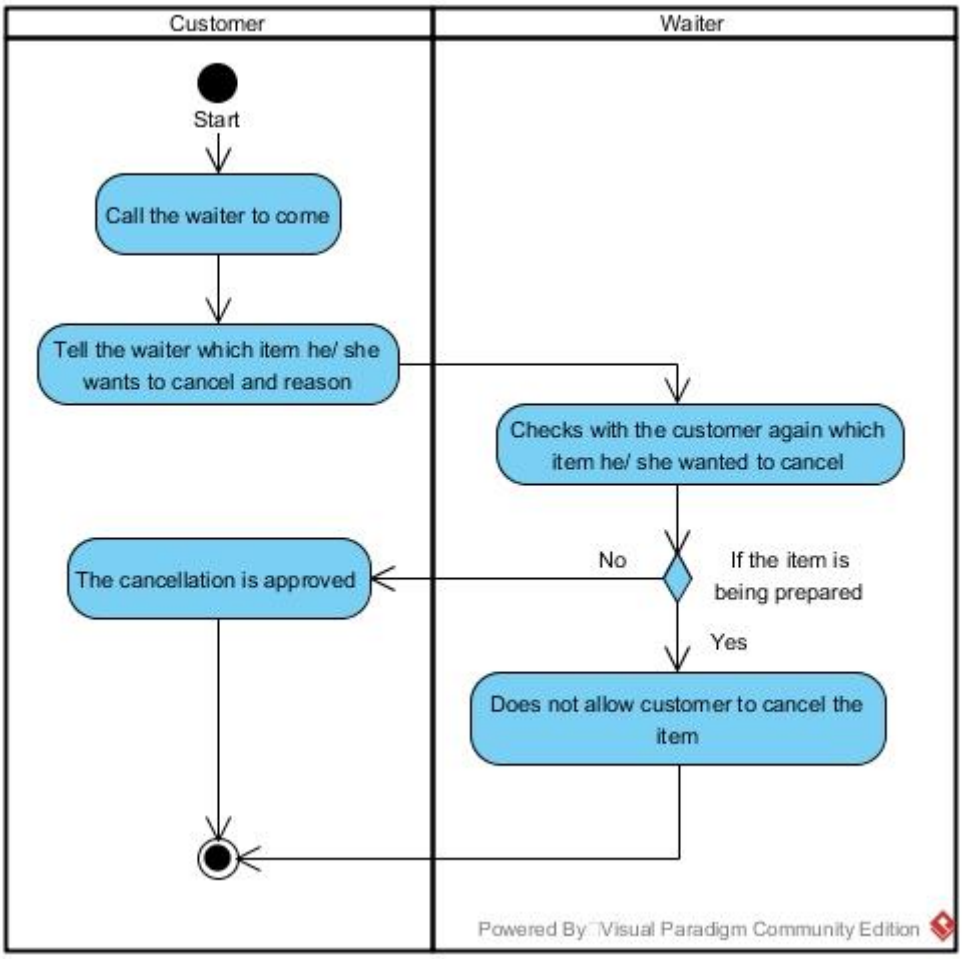
1. Customer calls the waiter to come.
2. Customer tells the waiter which item he/ she wants to cancel and reason.
3. Waiter checks with the customer again which item he/ she wanted to cancel.
4. Waiter checks whether the item is being prepared.
5. Food is not being prepared yet, the cancellation is approved.

Alternative Flow of Events: The item is already being prepared, waiter does not allow customer to cancel the item.

Post-condition: Manager removes the item from the order list.

Assumption: The item has not been sent to the customer yet.

Swimlane diagram for use case 7 is as below:



3.2.3 Use Case 8: Make Order

Primary Actor: Customer

Description: Customer calls the waiter to make order.

Pre-conditions: Customer had finished reading the menu.

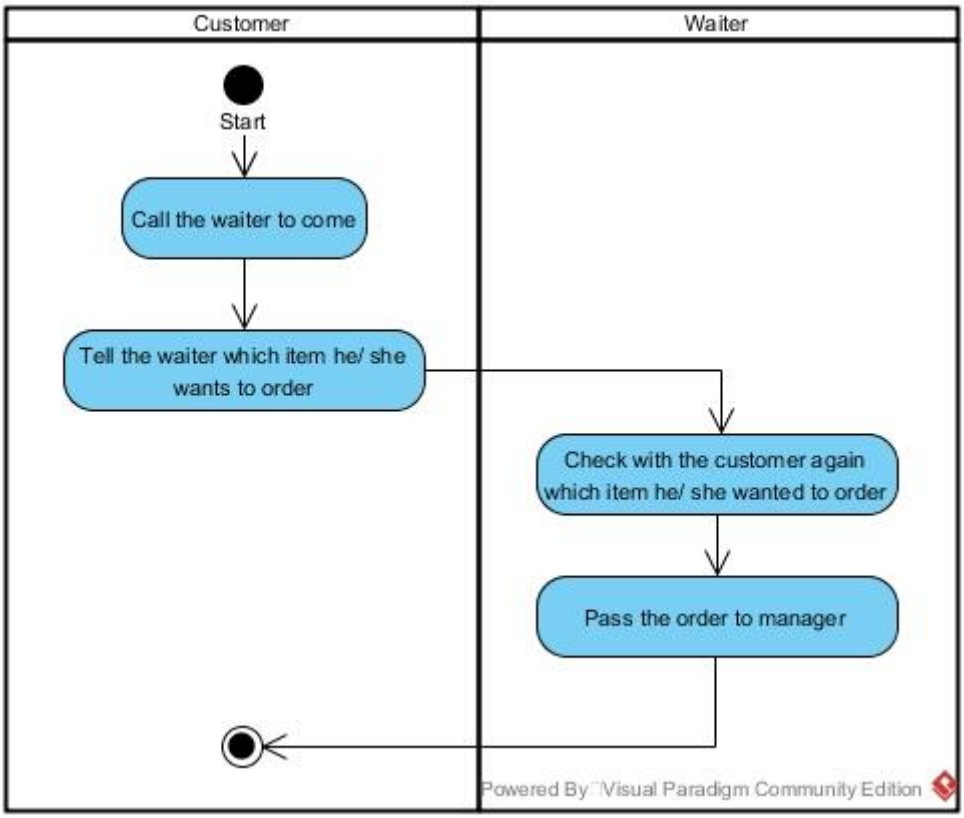
Flow of Events:

1. Customer calls the waiter to come.
2. Customer tells the waiter which item he/ she wants to order.
3. Waiter checks with the customer again which item he/ she had ordered.
4. Waiter passes the order to manager.

Post-condition: Manager enters order into the system.

Assumption: The menu is on the table.

Swimlane diagram for use case 8 is as below:



3.2.4 Use Case 9: Consume item

Primary Actor: Customer

Description: Customer eats the item.

Pre-conditions: Waiter had served the item to customer.

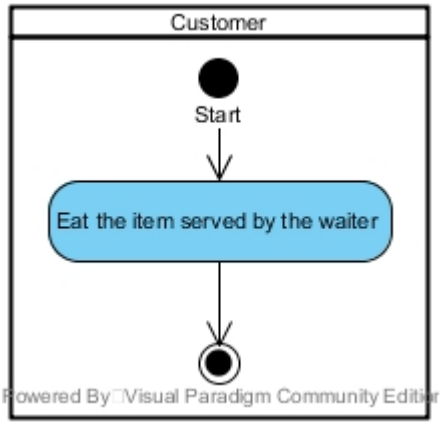
Flow of Events:

1. Customer eats the item served by the waiter.

Post-condition: Customer finished eating and proceeds to counter to pay bill.

Assumption: The item served is correct.

Swimlane diagram for use case 9 is as below:



3.2.5 Use Case 10: Item Dropped

Primary Actor: Customer

Description: The customer's item is being dropped due to insufficient ingredient.

Pre-conditions: Lack of ingredient for the item.

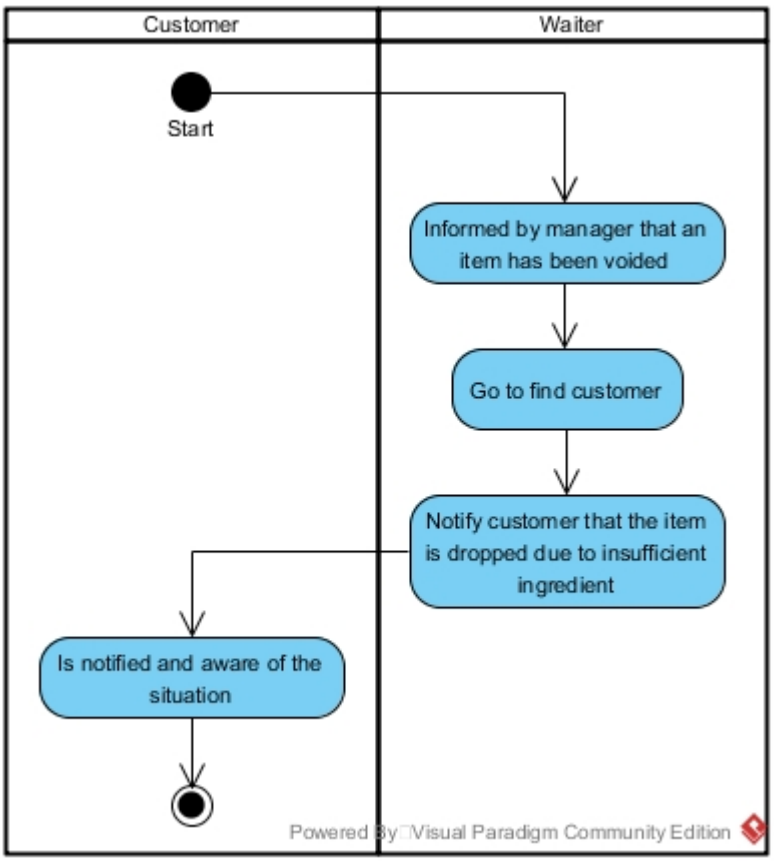
Flow of Events:

1. Waiter is informed by manager that an item has been voided.
2. Waiter goes to find customer.
3. Waiter notifies customer that the item of the customer is dropped due to insufficient ingredient.
4. Customer is notified and aware of the situation.

Post-condition: Waiter apologizes to the customer for the inconvenient caused.

Assumption: Customer agrees with the void.

Swimlane diagram for use case 10 is as below:



3.3 Actor 3: Waiter (Liew Soon Pang)

3.3.1 Use Case 11: Take Order

Primary Actor: Waiter

Description: The customer decides on what to order and the waiter will retrieve the order from the customer.

Pre-conditions: Customer had finished reading the menu.

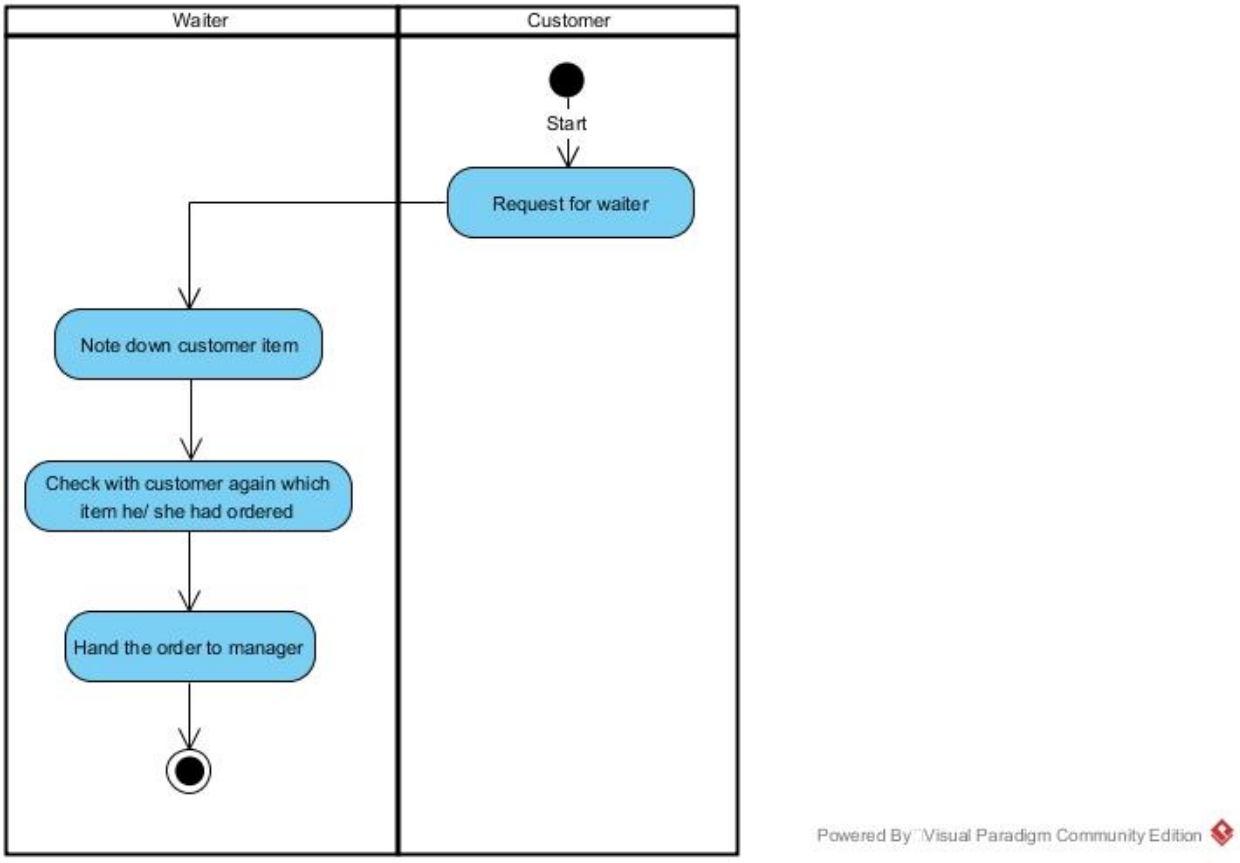
Flow of Events:

1. Customer requests for waiter.
2. Waiter notes down customer's item.
3. Waiter checks with the customer again which item they had ordered.
4. Waiter proceeds to hand the order to the manager.

Post-condition: Manager will enter the order into the system.

Assumption: The customer knows how to request for waiter.

Swimlane diagram for use case 11 is as below:



3.3.2 Use Case 12: Serve Item

Primary Actor: Waiter

Description: The waiter will serve the item to the customer.

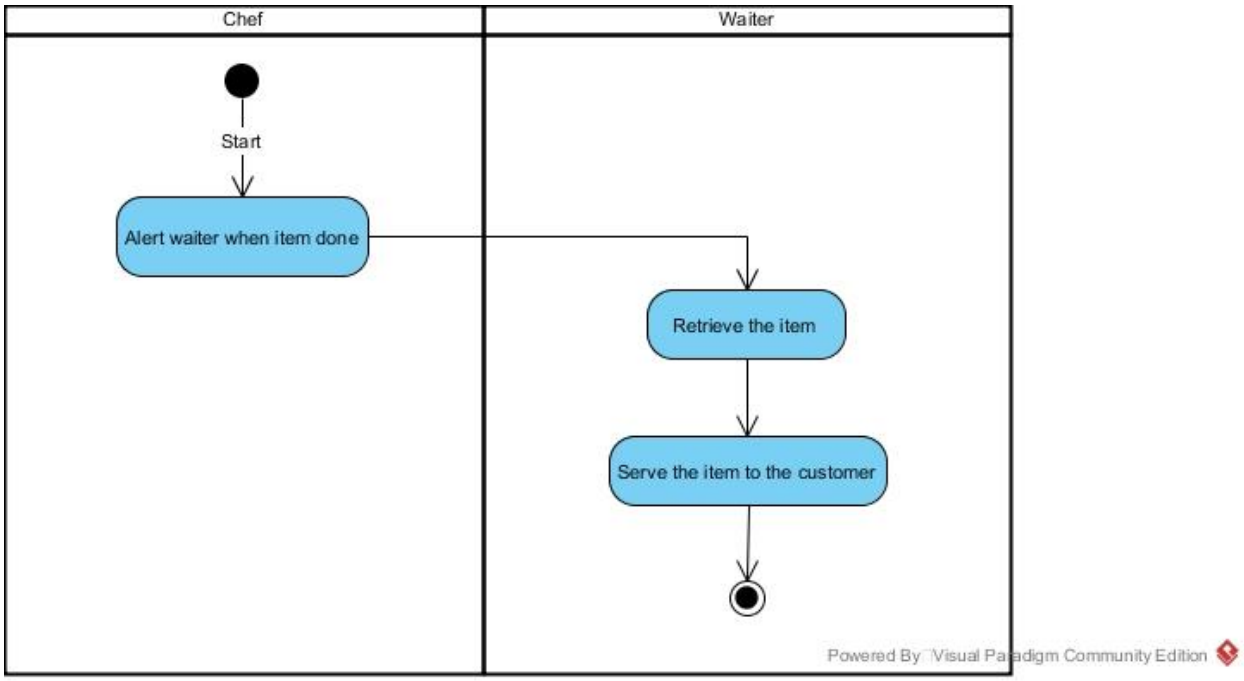
Pre-conditions: Waiter waits for the chef to give the signal when the item is ready.

Flow of Events:
1. The chef will alert the waiter after an item is done.
2. The waiter retrieves the item.
3. Waiter serves the item to the customer.

Post-condition: Customer eats the item.

Assumption: The item is prepared on time.

Swimlane diagram for use case 12 is as below:



3.3.3 Use Case 13: Facilitate Cancellation

Primary Actor: Waiter

Description: Waiter attends to the customer who needs help to cancel the item.

Pre-conditions: Customer realised that he/ she ordered the wrong item.

Flow of Events:

1. Customer calls the waiter.
2. Waiter notes down what to cancel.
3. Waiter checks with the customer again which item he/ she wanted to cancel.
4. Waiter checks with the manager to check if the item is being prepared.
5. The cancellation is approved.

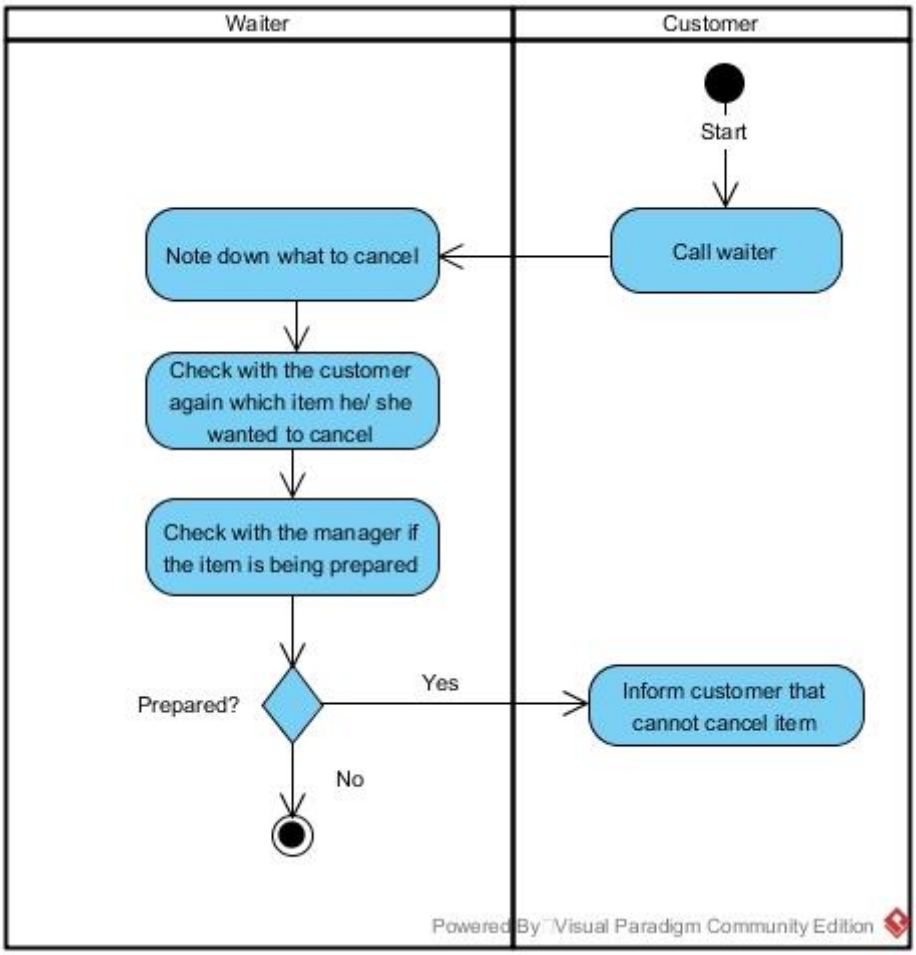
Alternative Flow of Events:

2. The cancellation is rejected because the item is being prepared.

Post-condition: Manager removes the item from the order list.

Assumption: Customer does not leave the restaurant.

Swimlane diagram for use case 13 is as below:



3.3.4 Use Case 14: Facilitate Void

Primary Actor: Waiter

Description: The chef voids an item due to lack of ingredient.

Pre-conditions: Lack of ingredient for the item.

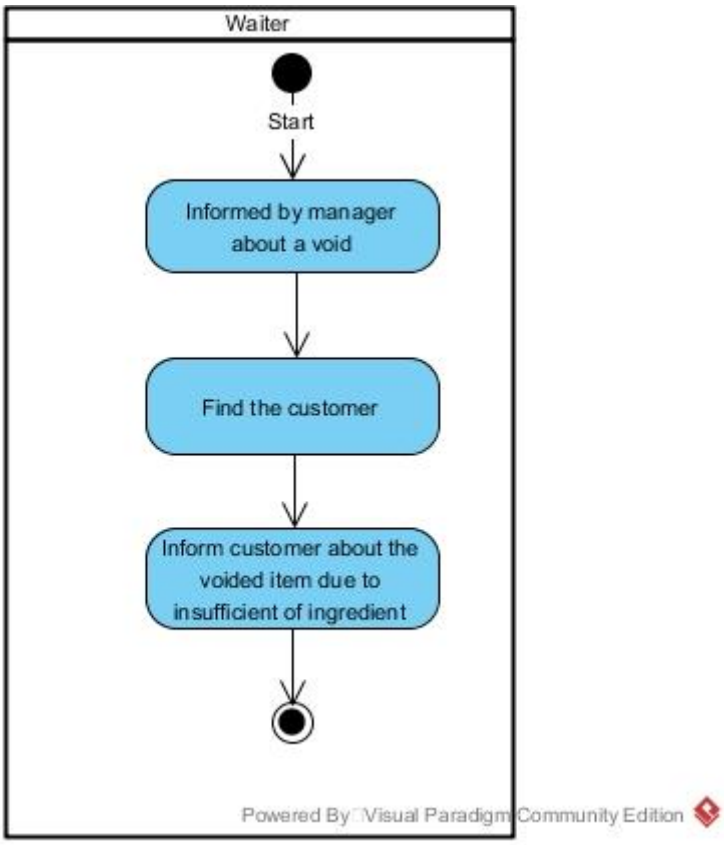
Flow of Events:

1. Waiter is informed by manager about a void.
2. Waiter finds the customer.
3. Waiter informs customer about the voided item due to insufficient of ingredient.

Post-condition: Waiter apologizes to the customer for the situation.

Assumption: Customer agrees with the void.

Swimlane diagram for use case 14 is as below:



3.4 Actor 4: Chef (Choo Jia Sheng)

3.4.1 Use Case 15: Accept Order

Primary Actor: Chef

Description: Chef accepts the order placed by manager.

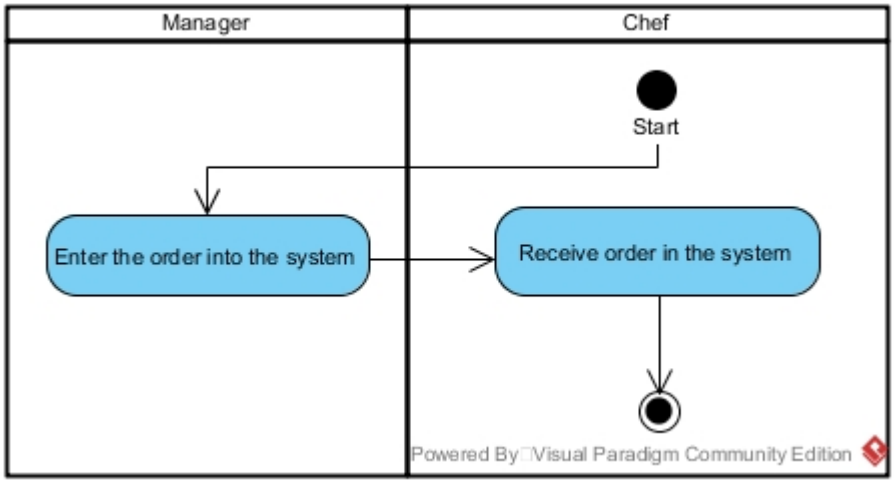
Pre-conditions: The system is connected for both kitchen system and counter system. Display of system is working fine, system is configured to accept the inputs.

Flow of Events:
1. Manager enters the order into the system.
2. Chef receives order in the system.

Post-condition: The accepted order will be added automatically into the bottom of order queue list.

Assumption: Chef is present in the kitchen.

Swimlane diagram for use case 15 is as below:



3.4.2 Use Case 16: Terminate Item

Primary Actor: Chef

Description: A notification will appear when an item is cancelled by the customer.

Pre-conditions: The system is connected for both kitchen system and counter system. Display of system is working fine, system is configured to accept the inputs.

Flow of Events:

1. Manager asks if an item had been prepared.
2. Chef updates manager on the preparation status.
3. The food has not been prepared yet, manager removes the item from the order list.
4. Chef is being notified on the cancellation.

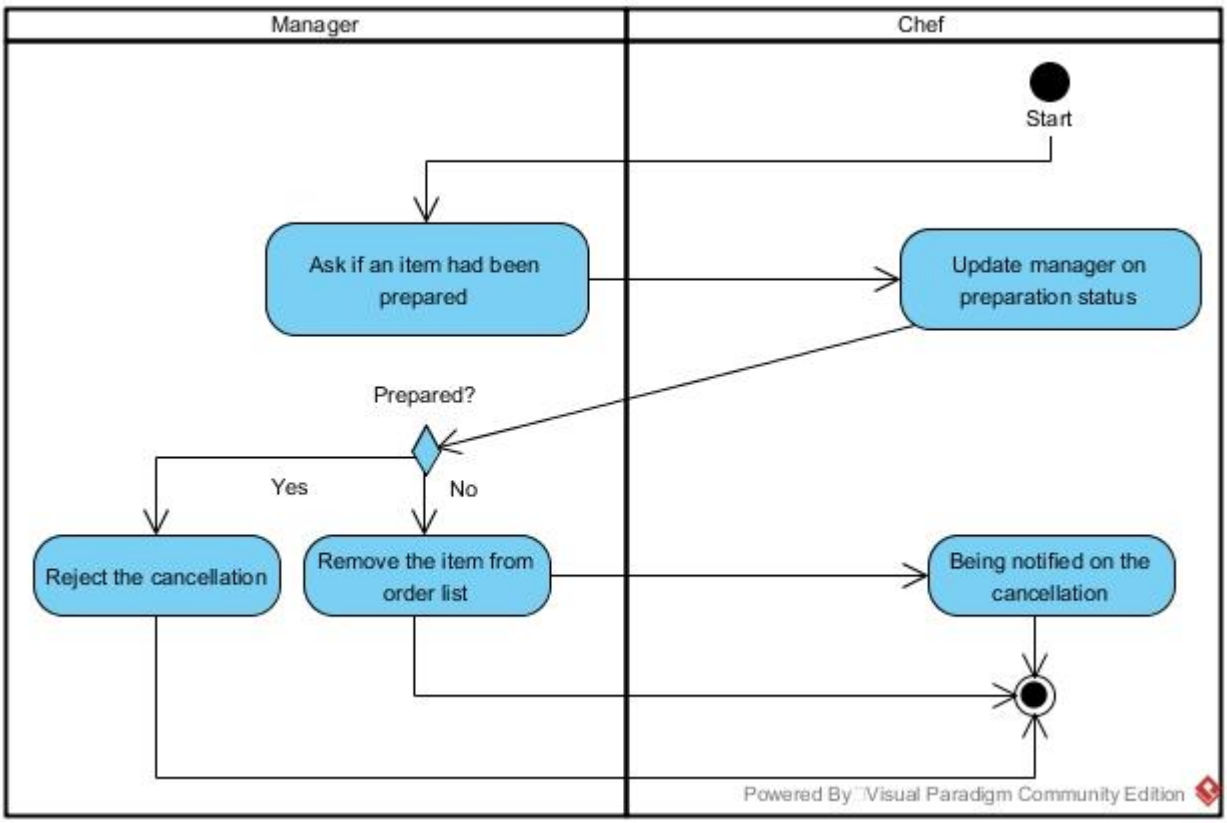
Post-condition: Manager will update waiter on the cancellation status.

Alternative Flow of Events:

1. The cancellation is rejected because the item had been prepared.

Assumption: Customer will not leave the restaurant even food is prepared and void is rejected

Swimlane diagram for use case 16 is as below:



3.4.3 Use Case 17: Void Item

Primary Actor: Chef

Description: Chef checks for the ingredient and voids the item with insufficient ingredient.

Pre-conditions: The system is connected for both kitchen system and counter system. Display of system is working fine, system is configured to accept the inputs.

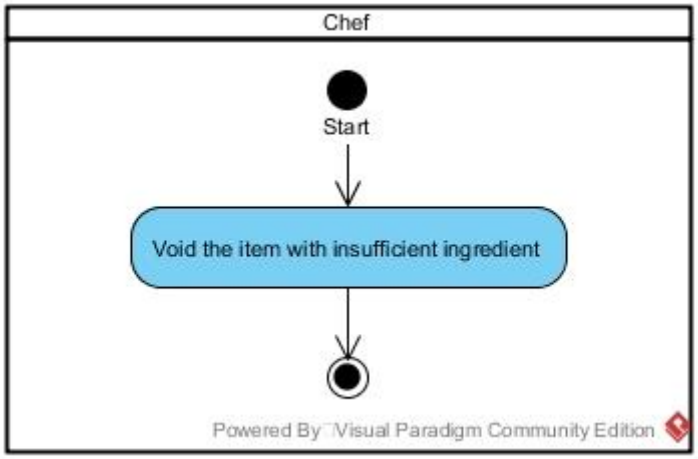
Flow of Events:

1. Chef voids the item with insufficient ingredient.

Post-condition: Manager will be notified on the void.

Assumption: The customer is still present in the restaurant.

Swimlane diagram for use case 17 is as below:



3.4.4 Use Case 18: Prepare Item

Primary Actor: Chef

Description: Chef prepares the food according to the queue of the order list.

Pre-conditions: There is an order in the queue list for the chef to prepare the item.

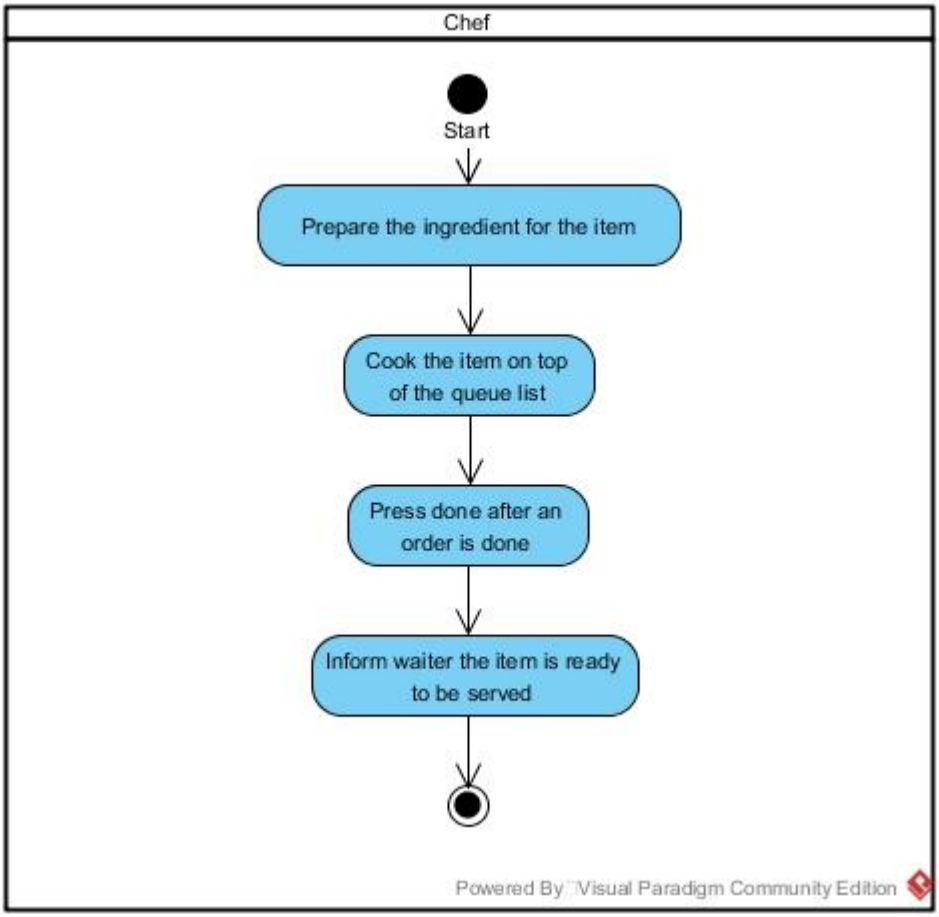
Flow of Events:

1. Chef prepares the ingredient of the item from fridge or storage room.
2. Chef starts to cook the item on top of the queue list.
3. After an order is done, chef will press done to remove it from the queue.
4. Chef will inform waiter that the item is ready to be served.

Post-condition: Waiter serves the item to customer.

Assumption: Chef is familiar with the input system and the ingredient of item is sufficient.

Swimlane diagram for use case 18 is as below:

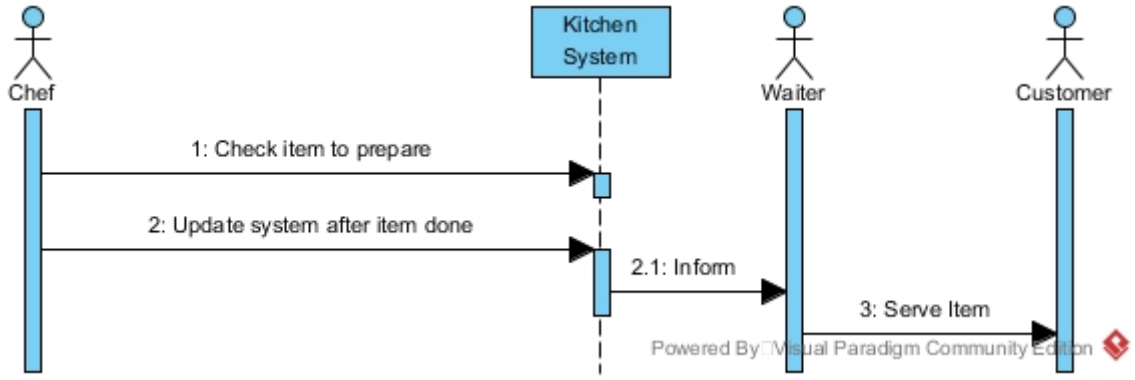


4 Specific Requirements

4.1 Class Diagrams (Prepared by all the members together)

4.2 Sequence Diagrams

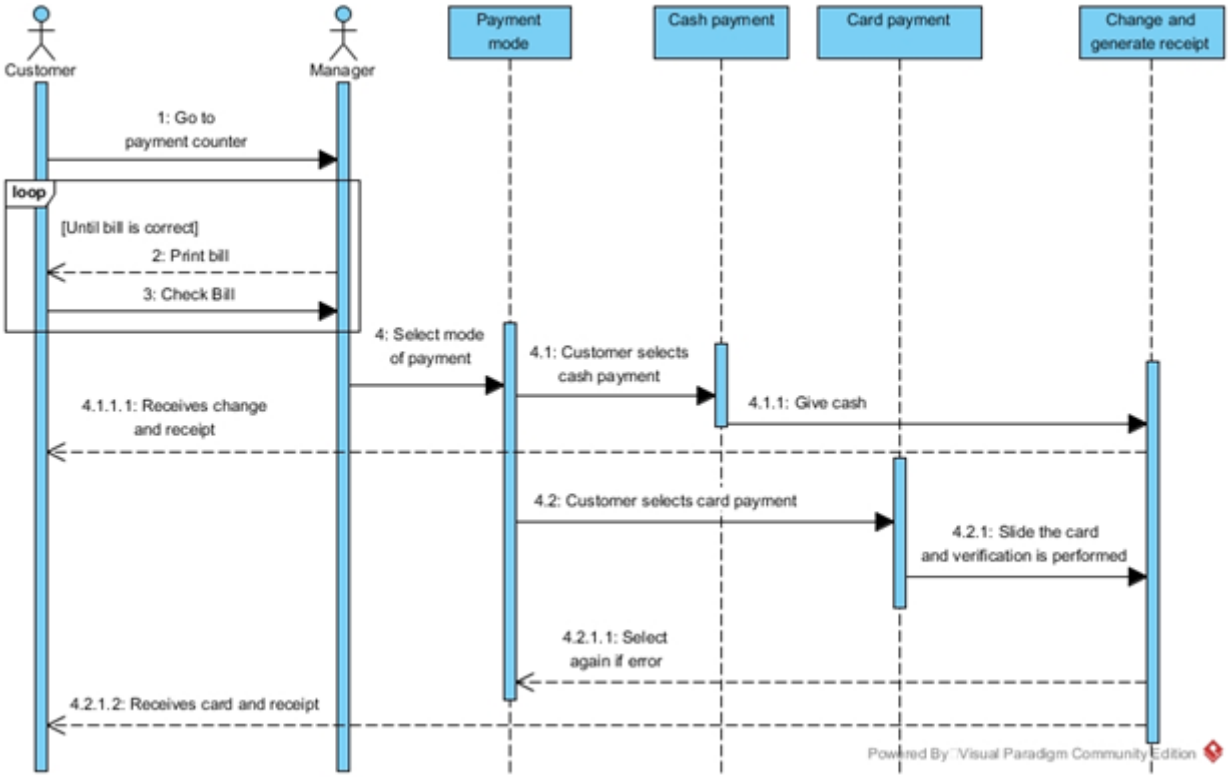
4.2.1 Use Case 1: Item (Liew Soon Pang)



Description:

1. Chef checks item to prepare from kitchen system.
2. Chef updates system after an item is done.
2.1. Chef informs waiter.
3. Waiter serves item to customer.

4.2.2 Use Case 2: Pay Bill (Nicholas Tan Yu Zhe)



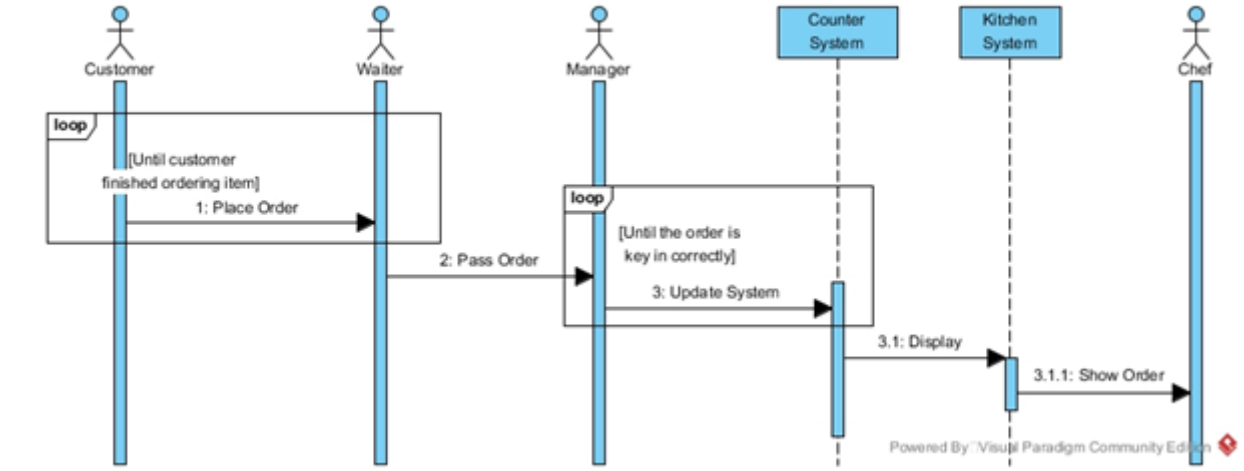
Description:

1. Customer goes to the payment counter, which a manager is in charge.
2. Until bill is correct, manager prints bill.
3. Until bill is correct, customer checks bill.
4. Manager lets the customer selects mode of payment.

4.1. The payment mode chosen by customer is cash payment.
4.1.1. Customer makes the payment by cash.
4.1.1.1. The customer receives change and receipt from manager.

4.2. The customer chooses to pay by card payment.
4.2.1. Card is slid and validation is performed.
4.2.1.1. Customer has to select other payment method if error occurred.
4.2.1.2. The customer receives card and receipt from manager.

4.2.3 Use Case 3: Place Order (Liew Soon Pang)



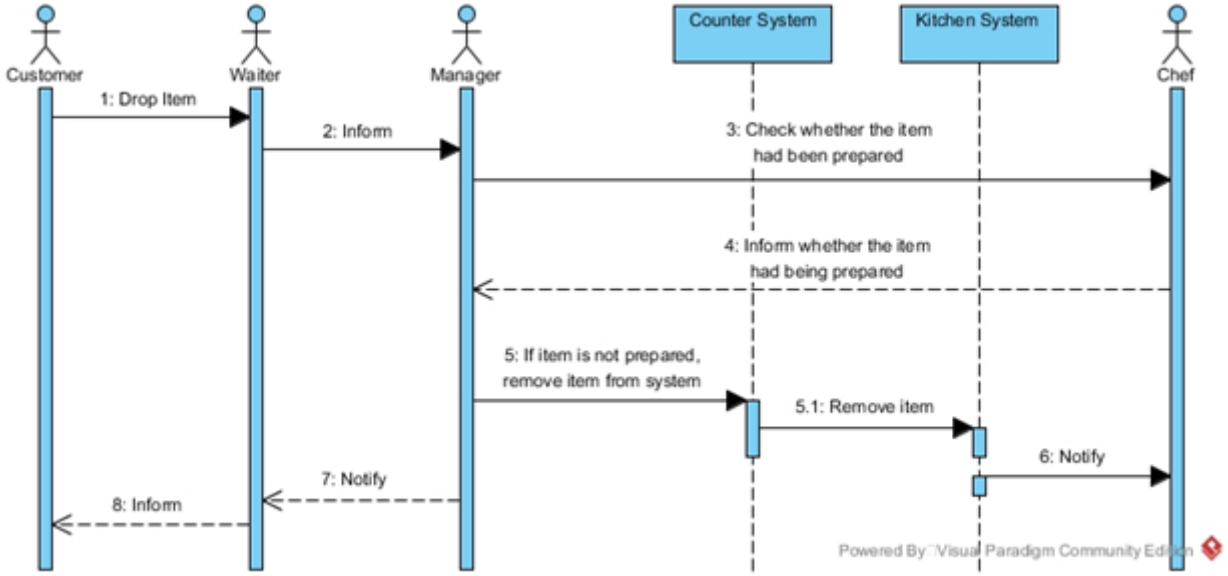
Description:

1. Until customer finished ordering item, customer places order with waiter.
2. Waiter passes order to manager.
3. Until the order is key in correctly, manager updates the counter system.

3.1. Counter system displays the order in kitchen system.

3.1.1. Kitchen system shows order to chef.

4.2.4 Use Case 4: Customer Cancel (Choo Jia Sheng)



Description:

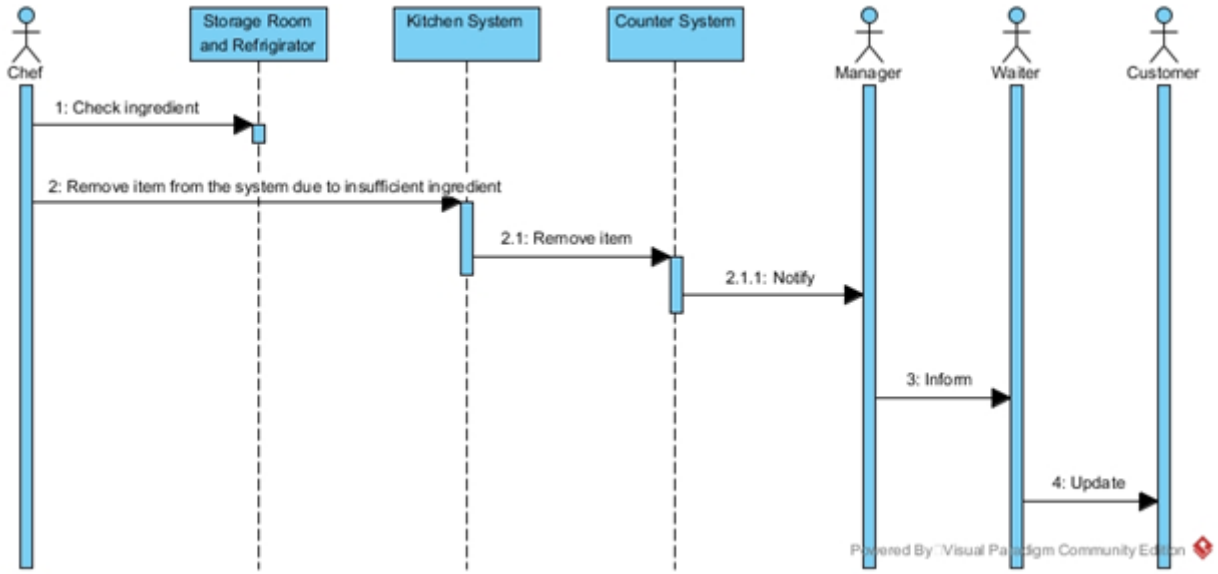
1. Customer wants to drop an item.
2. Waiter informs the manager regarding the customer's request.
3. The manager checks with chef whether the item had been prepared.
4. The chef informs manager whether the item had been prepared.
5. If item is not prepared, manager removes item from the system.

5.1. Counter system will remove item from kitchen system.

6 Kitchen system will notify chef on the cancellation.

7. The manager notifies waiter about the cancellation status.
8. The waiter informs customer.

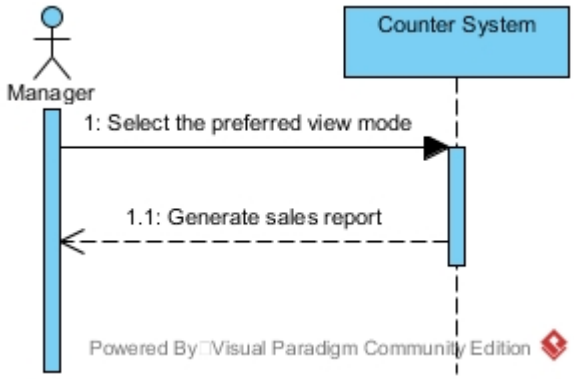
4.2.5 Use Case 5: Chef Void (Ng Kang Jie)



Description:

1. Chef checks ingredient in storage room and refrigerator.
2. The chef removes item from the system due to insufficient ingredient.
2.1. Kitchen system removes the item from counter system.
2.1.1. Counter system notifies manager on the void.
3. The manager informs waiter regarding the void.
4. The waiter updates customer.

4.2.6 Use Case 6: Generate Report (Ng Kang Jie)



Description:

1. Manager selects the preferred view mode.
1.1. The counter system generates sales report for manager.

5 Data Design

5.1 Data Dictionary

The data dictionary for the system is as below:

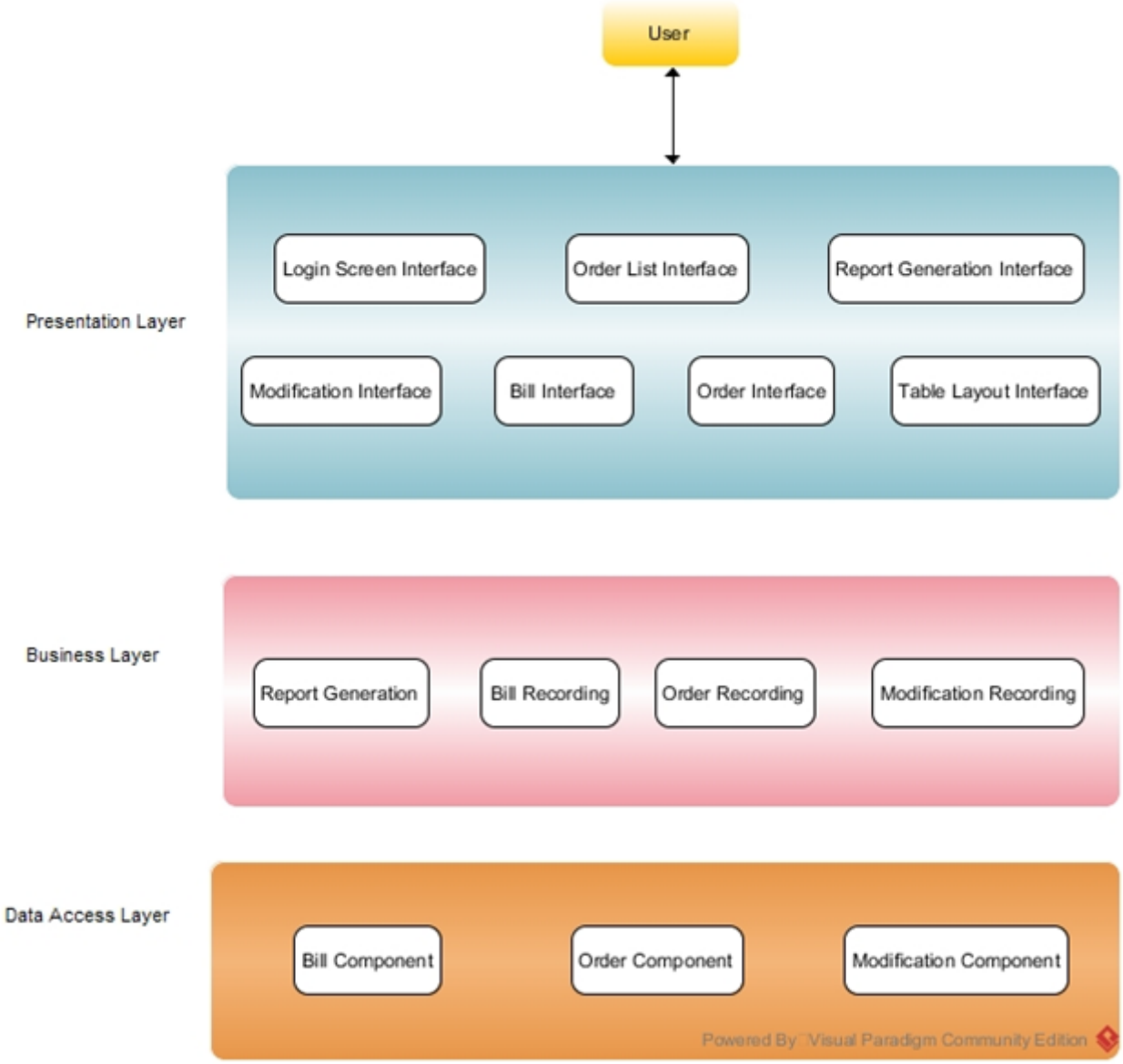
TABLE NAME	ATTRIBUTE NAME	DESCRIPTION	TYPE	FORMAT	PK/FK	FK REFERENCE TABLE
Order	orderID	Number of the order	INTEGER	10	PK	
Order	date	Date of the order	DATETIME	dd-mm-yyyy		
Order	tableNo	Table number of the order	INTEGER	10		
OrderDetail	orderID	Number of order	INTEGER	10	PK, FK	Order
OrderDetail	itemID	Store all the item	VARCHAR(20)	Xxxxxxxxxxxx		
OrderDetail	quantity	Total payment of the order	INTEGER	10		
Item	itemID	Item number	INTEGER	10	PK	
Item	itemName	Item name	VARCHAR(20)	Xxxxxxxxxxxx		
Item	itemType	Item type drink or food	VARCHAR(20)	Xxxxxxxxxxxx		
Item	itemPrice	Item price	DOUBLE	999.99		

6 Architecture Design

6.1 Software Architecture

First, the user enters the presentation layer where an interface of wide option choices is to be selected. Following the user selecting the option, it will bring the user to the business layer where work flow and logical operation comes in. After that, it moves to data layer which contains a group of entities. Finally, it reaches the end of the database.

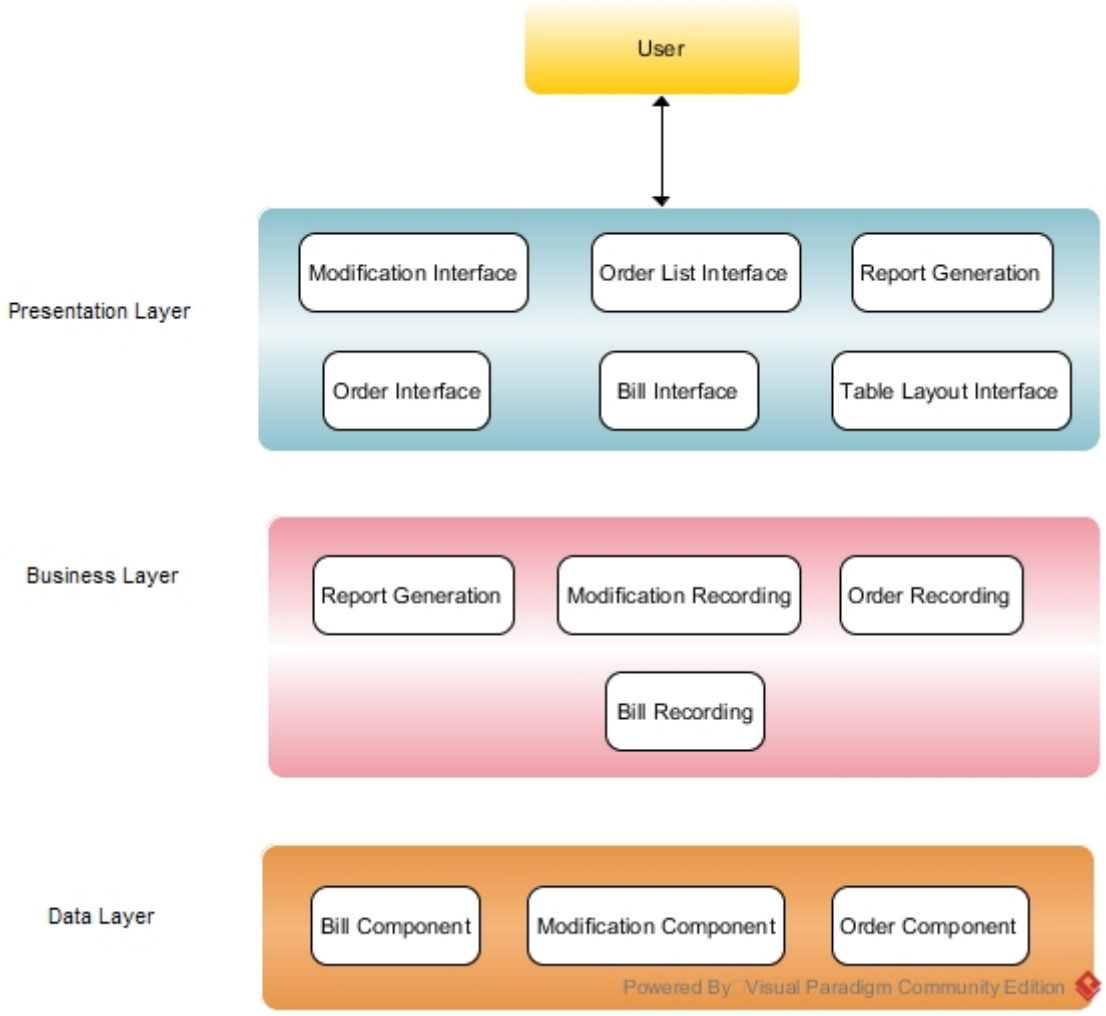
The architecture diagram for the system is as below:



6.1.1 Subsystem 1: Counter System

The architecture diagram that we use for the counter is multilayered architecture model. The user is able to interact with the presentation layer and is able to access business layer and data layer to return the result. For example, the user is able to click report generation and it will access the business layer to return the report to the user.

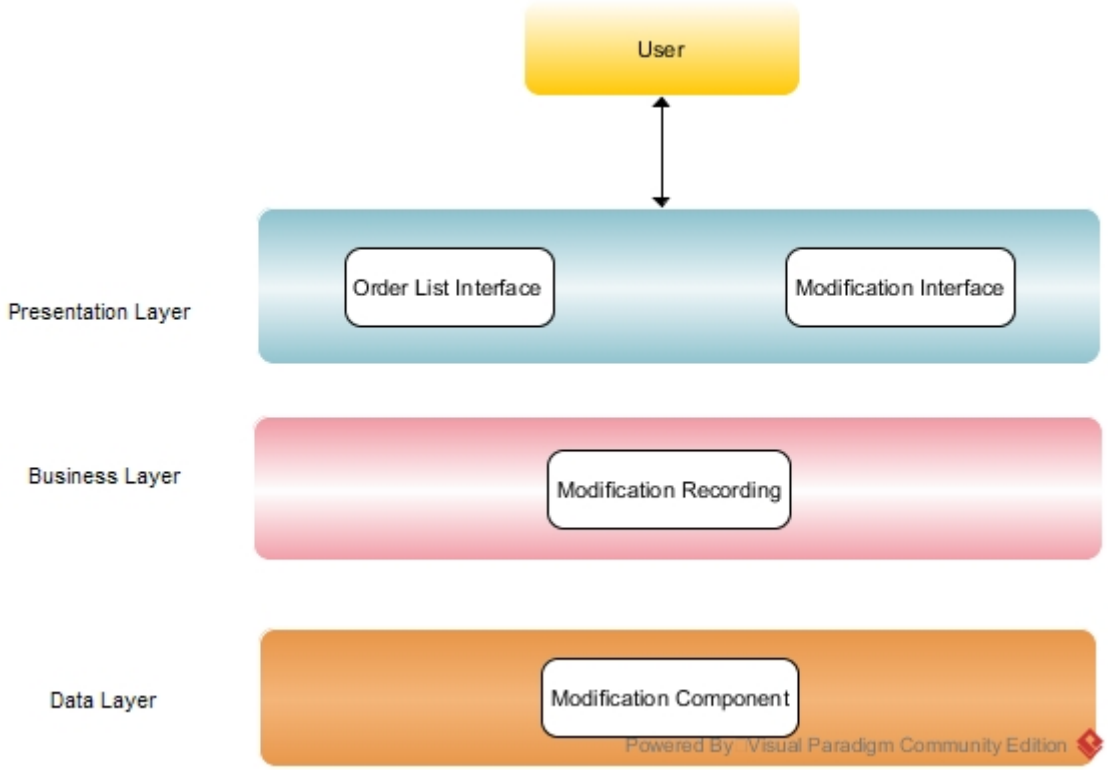
The architecture diagram for the subsystem is as below:



6.1.2 Subsystem 2: Kitchen System

The architecture diagram that we use for the kitchen is multilayered architecture model. The user is able to interact with the presentation layer and is able to access business layer and data layer to return the result. For example, the user is able to click void and access the business layer and data layer to retrieve which item to void and return result to the user.

The architecture diagram for the subsystem is as below:



7 Interface Design

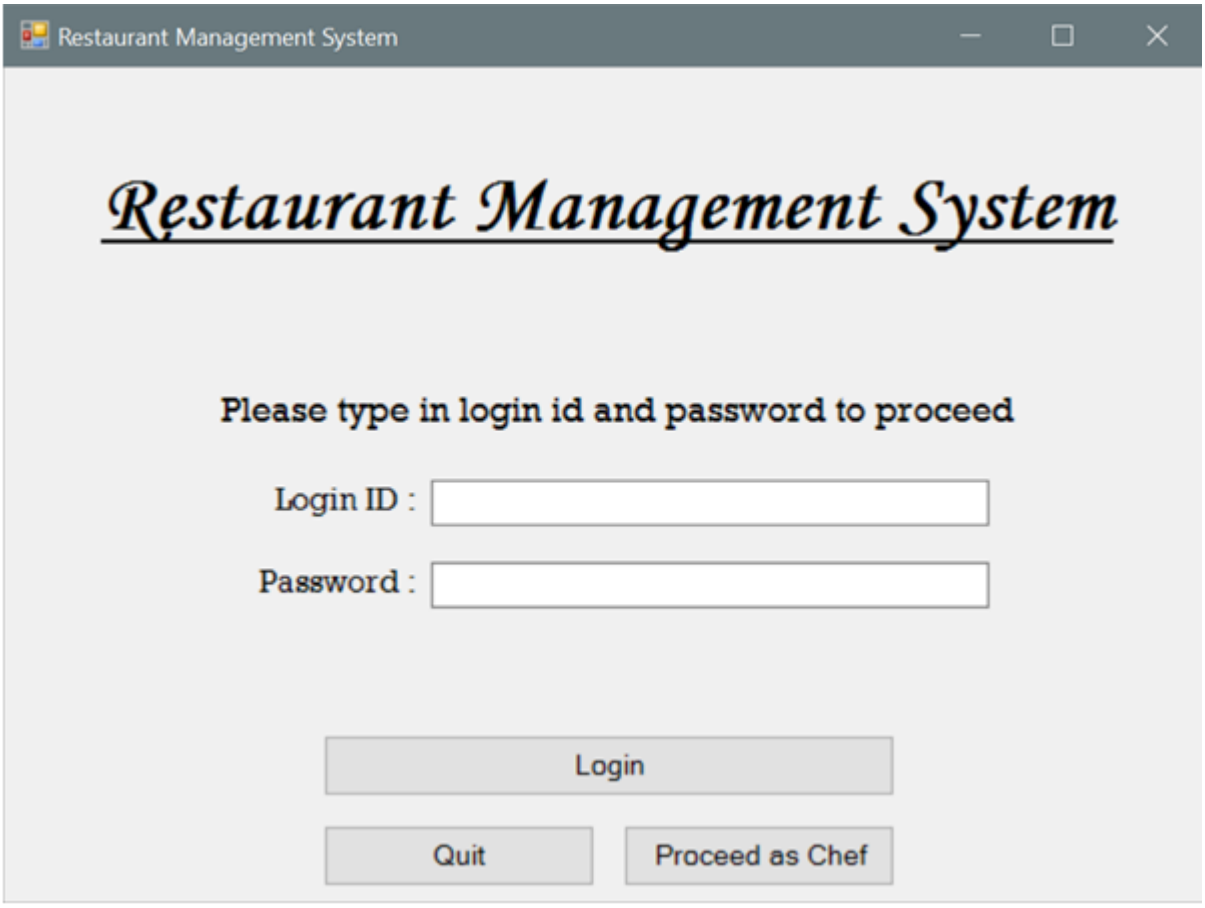
7.1 Main Screen

This is the main menu of the program. User must login in order to use the features of this program.

Click ‘Login’ button after entering the correct login ID and password to enter the program.

‘Quit’ button has the same function as ‘X’ on the top right of window, which it will exit the program.

‘Proceed as Chef’ button will open a new window which displays the kitchen monitor view.



7.2 Subsystem 1 Screens: Counter System

This is the system view after manager logged into his/her account.

'Log Out' button will let manager go back to the main menu.

Date on the top right corner shows the current date.

Button for the Table No. will update the list of ordered items and show all the items according to table number. Total will show the payable amount of a table.

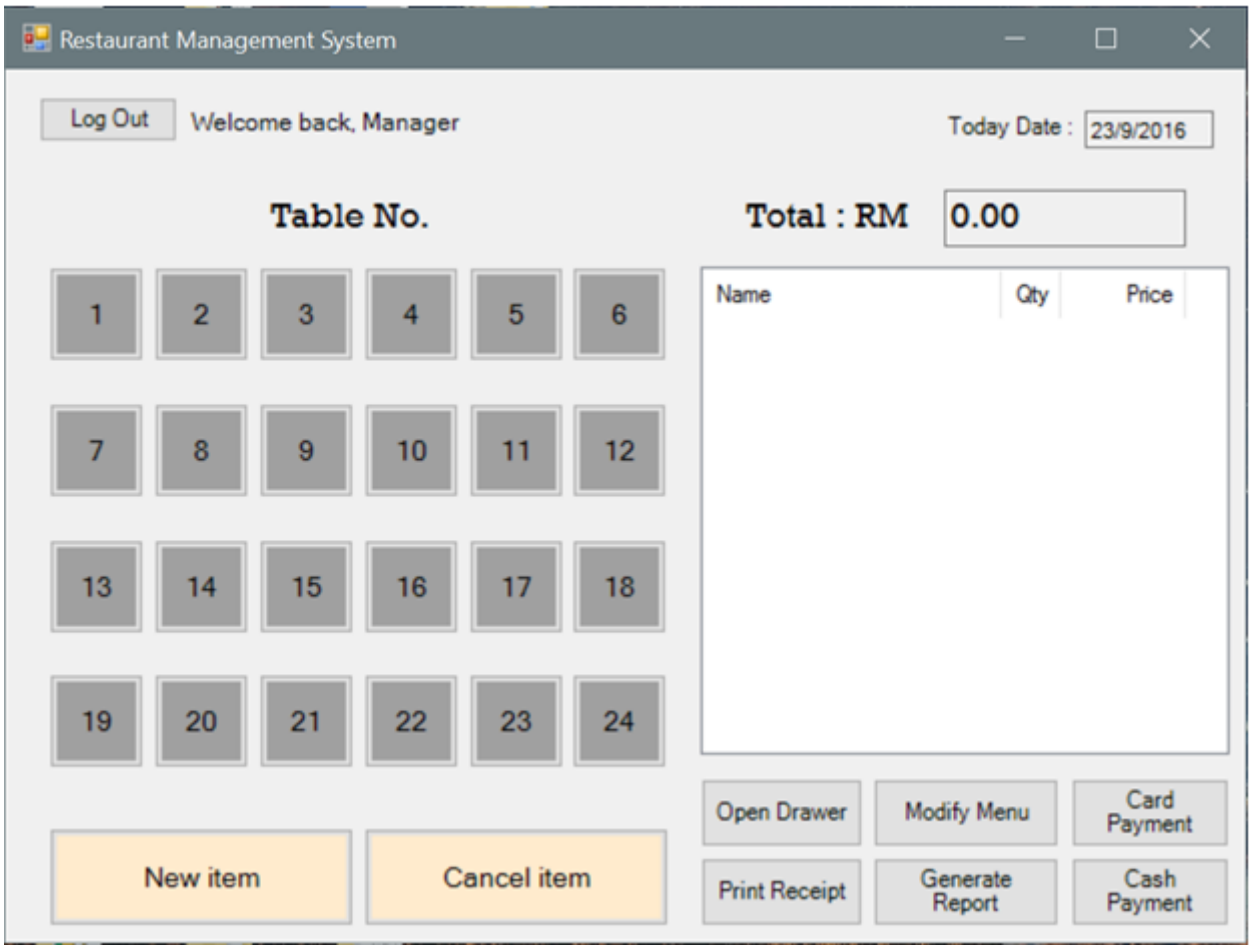
'New item' button will prompt a new small window for manager to add item to a table. 'Cancel item' button will prompt a new small window for manager to cancel item.

'Open Drawer' button will open the cash register for manager to collect money and give back the remaining balance.

'Print Receipt' button will print the receipt of the selected order according to table number.

'Modify Menu' button allows manager to add and remove item in the restaurant menu.

'Generate Report' button brings manager to the sales report page.



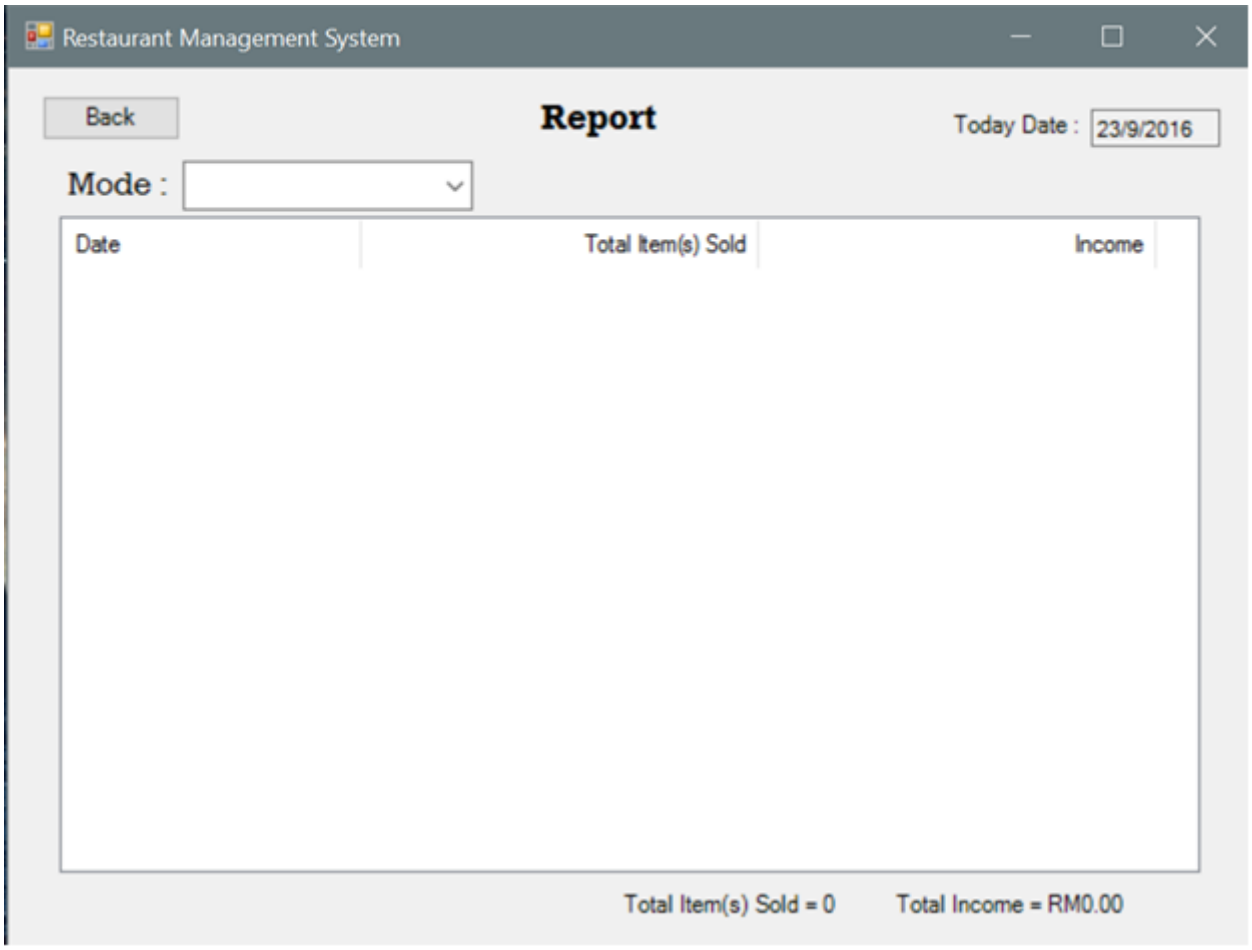
Back' button allows manager to return to the manager's main functions view.

Date on the top right corner show the current date.

Mode drop down list allows manager to select preferred view mode.

The list box will display the sales report.

Total item(s) sold and total income will be display at the right bottom of the window.

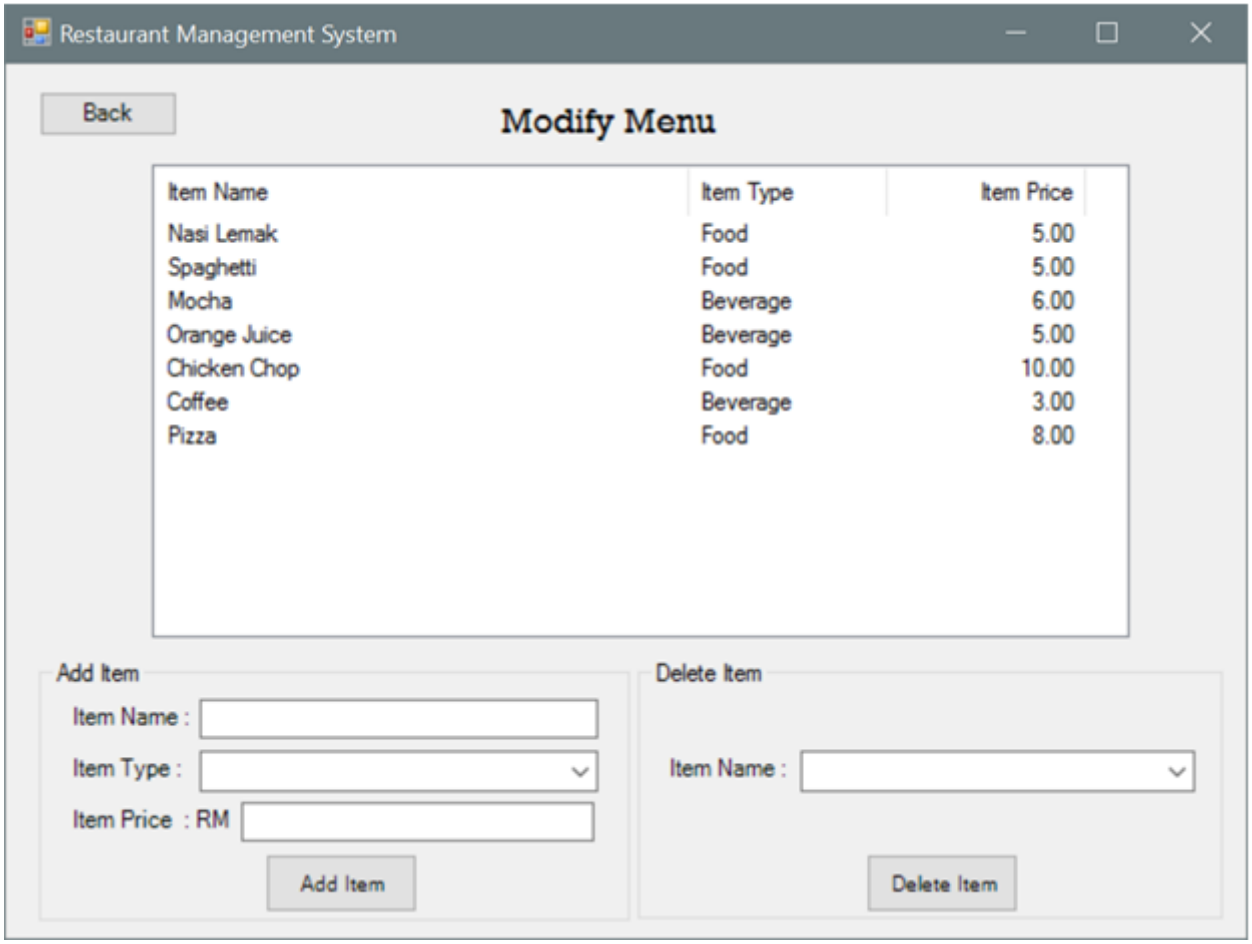


'Back' button allows manager to return to the manager's main functions view.

The item list will show all the items saved in the system.

'Add Item' button and the columns above allow manager to add new item into the list.

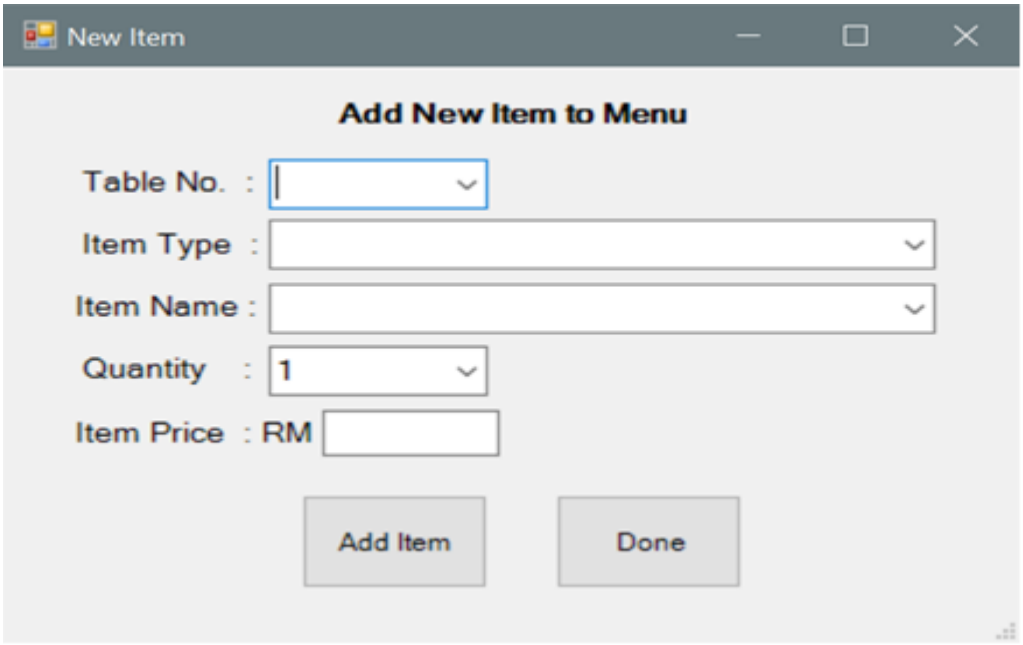
'Delete Item' button and the drop down list above the button allows manager to select item to delete from the list.



This is a small popup window which allows manager to add multiple items according to table number, item type, item name, quantity and default item price.

Press ‘Add Item’ after the details are correct to add orders to a table.

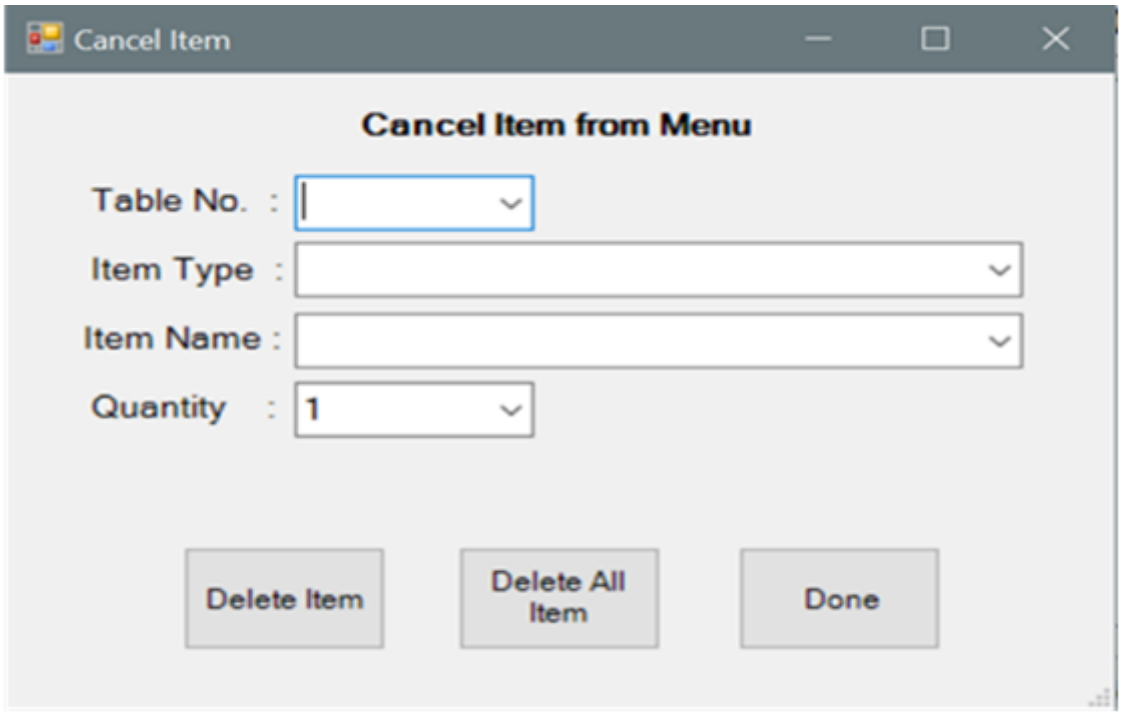
Press ‘Done’ button to close the window and return to manager’s main functions view.



This is a small popup window which allows manager to cancel either one or all items according to table number, item type, item name and quantity.

Press ‘Delete Item’ to delete a particular item or press ‘Delete All Item’ to delete all items of a particular table.

Press ‘Done’ button to close the window and return to manager’s main functions view.



This is a small popup window which allows manager to calculate the change to be returned to the customer with due amount and paid amount.

Due will show the total payable amount of a table.

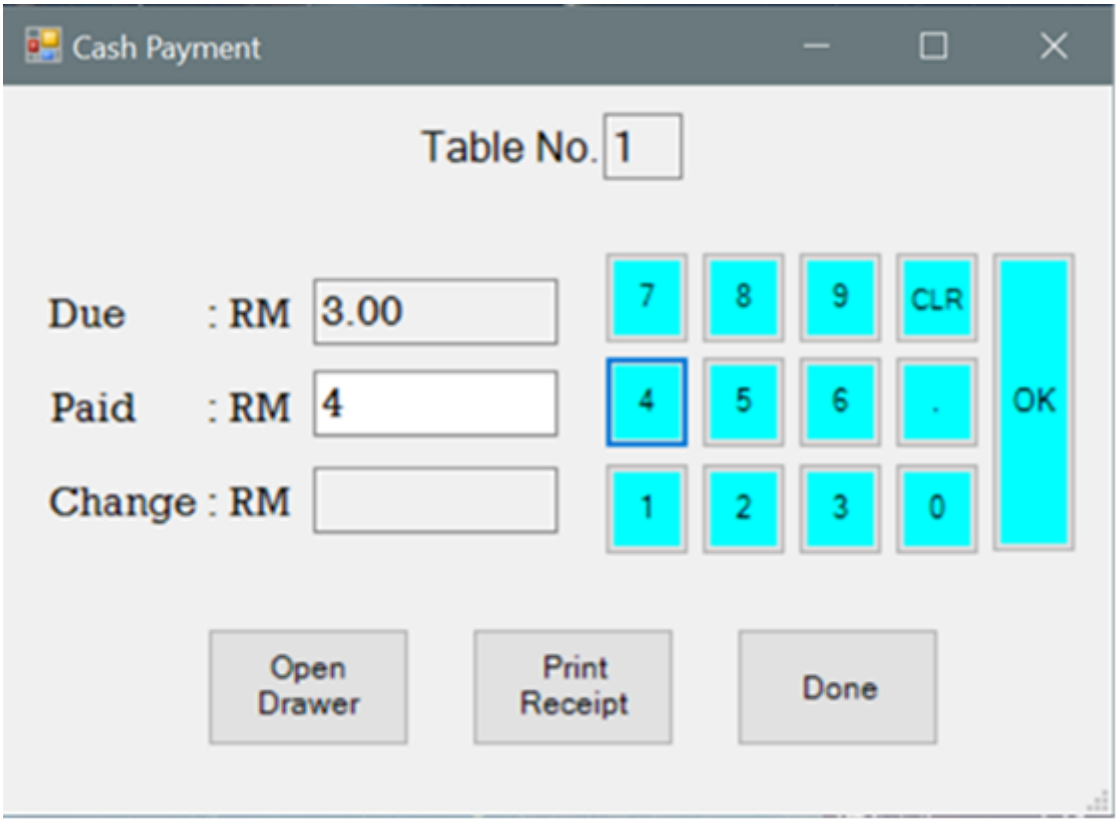
Paid allows manager to key in the amount a customer has paid.

Change will show manager how much change to be returned to the customer. Manager just needs to press 'OK' button after keying in the amount that the customer had paid to display how much change to return to the customer.

'Open Drawer' button will open the cash register for manager to collect money and give back the remaining balance.

'Print Receipt' button works as a backup button if the receipt is not printed automatically after a payment.

Press 'Done' button to close the window and return to manager's main functions view.



This is a small popup window which allows manager to swipe the customer card and check for card validity.

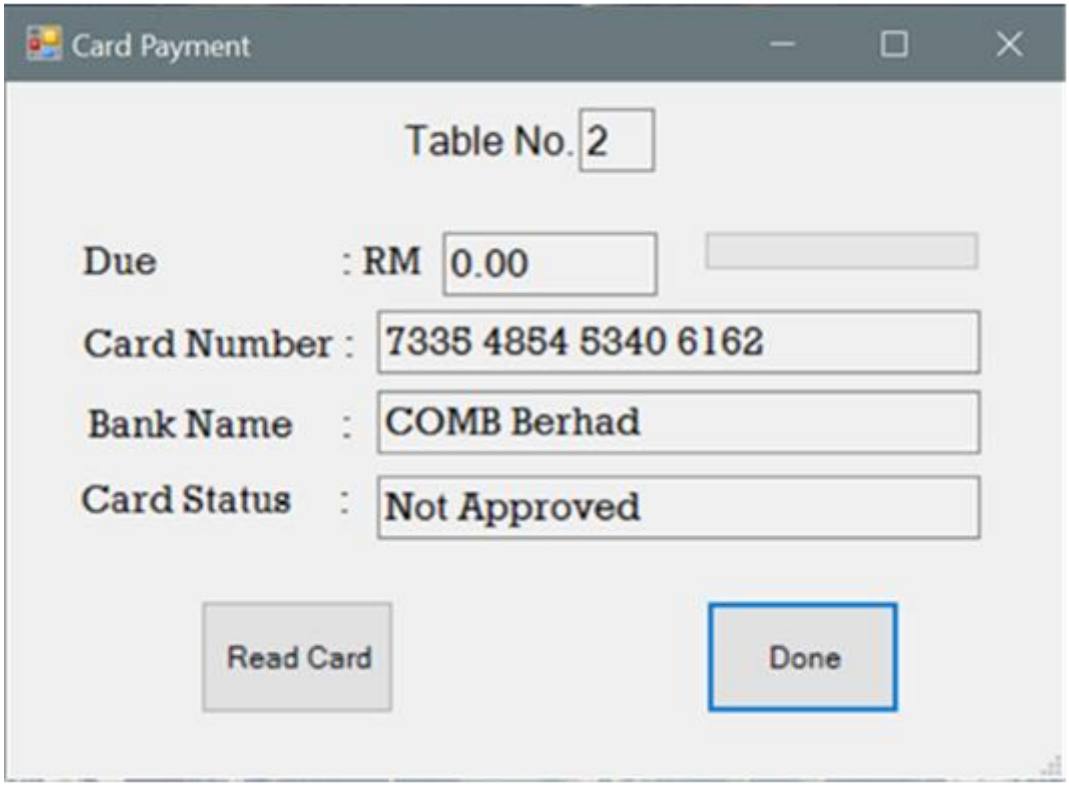
Scanning bar on top will run when card is processing through the bank.

Due will show the total payable amount of a table.

The information will be filled automatically when the process is done.

‘Read Card’ button serves as a backup button if the card does not be read on its own.

Press ’Done’ button to close the window and return to manager’s main functions view.



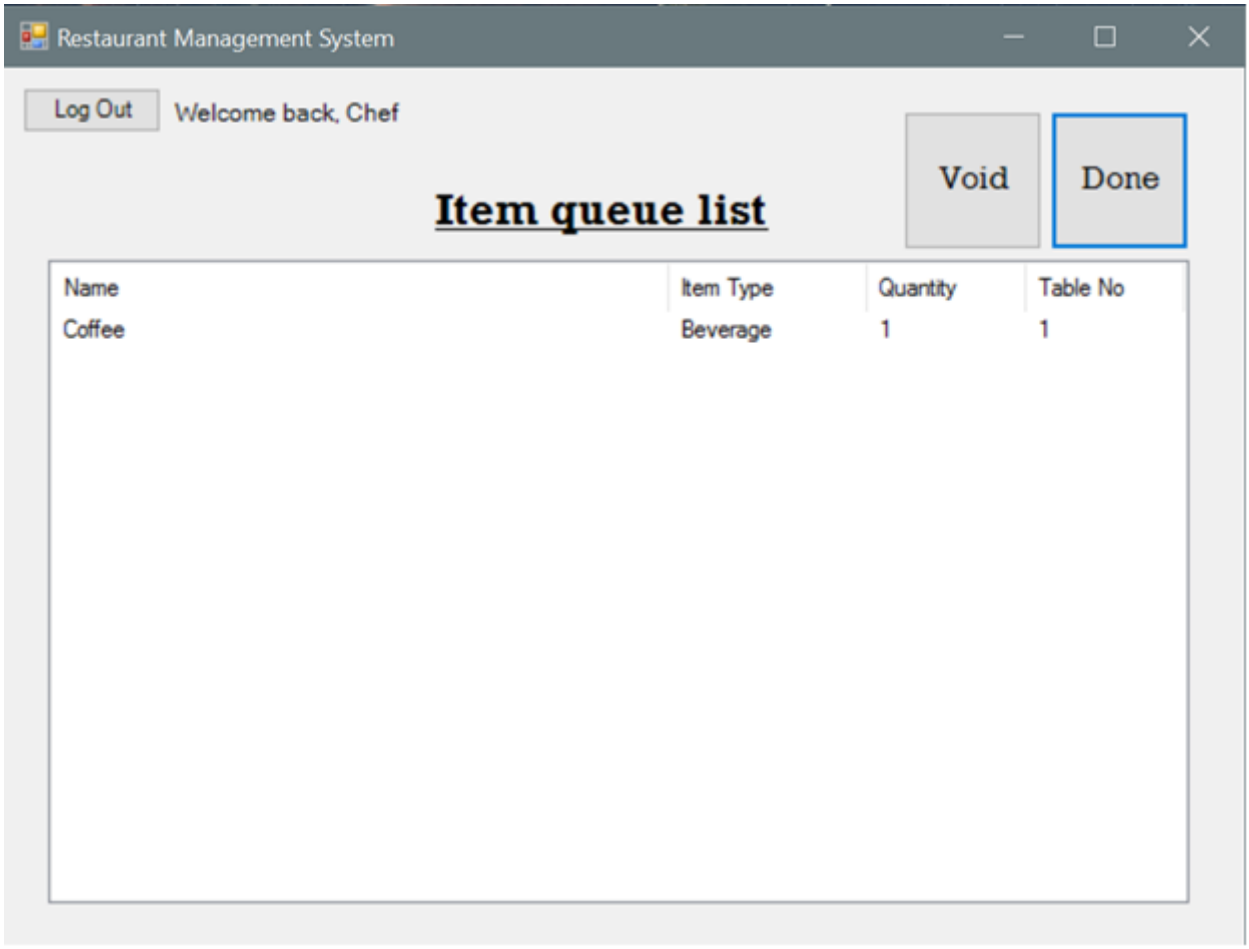
7.3 Subsystem 2 Screen: Kitchen System

This is the monitor view of chef in the kitchen which shows all the items that need to be prepared for the customers in a queue list.

'Log Out' button will close the current window.

'Void' button enables chef to void an item if the ingredient level for the item is insufficient.

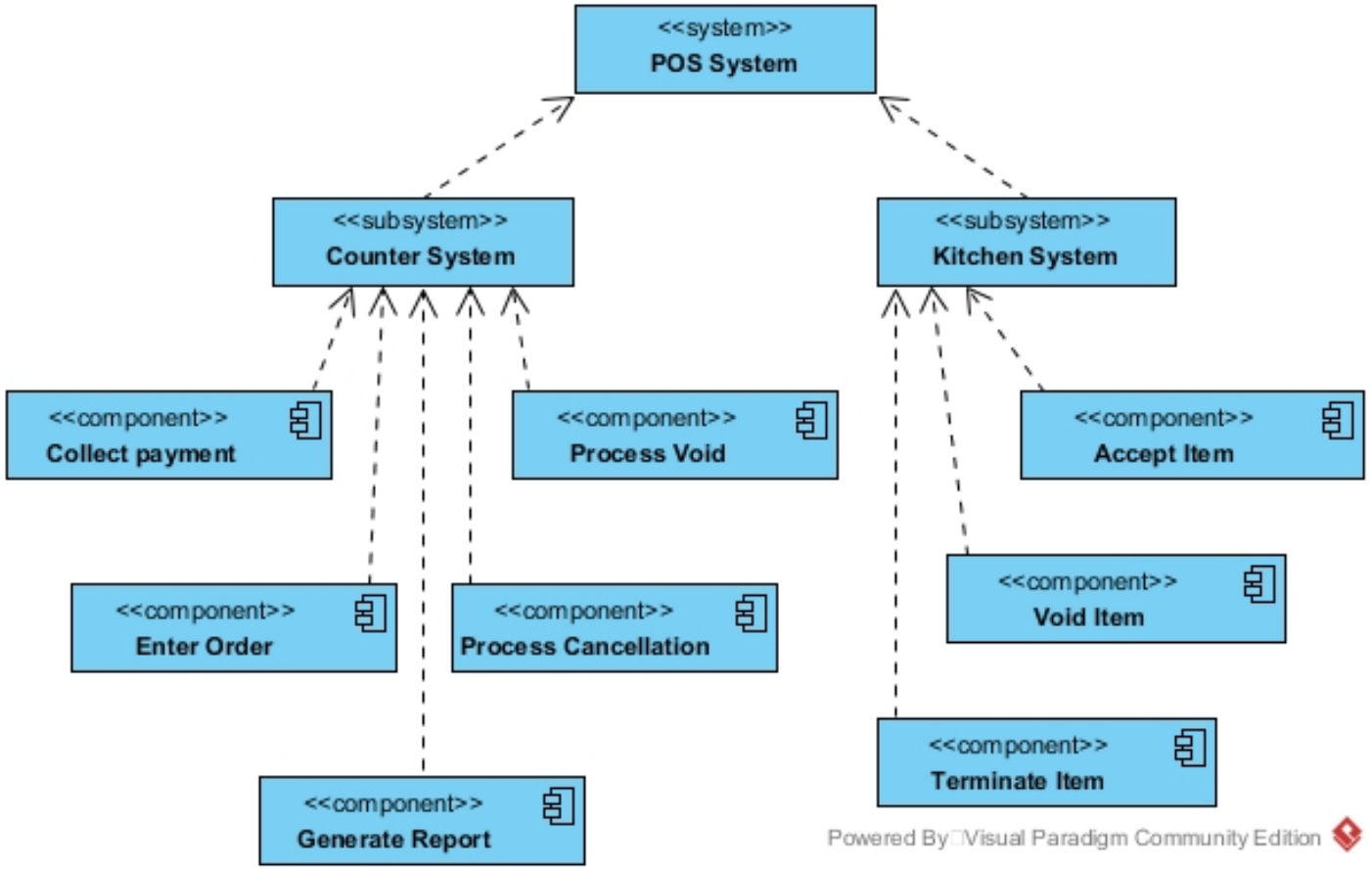
'Done' button allows chef to delete an item after the preparation is done and ready to serve.



8 Component Design

8.1 Main Components

The Restaurant Management System is a POS system with 2 subsystems, namely Counter System and Kitchen System. There are five components in the Counter System and three components in the Kitchen System. The system's component diagram is as below:



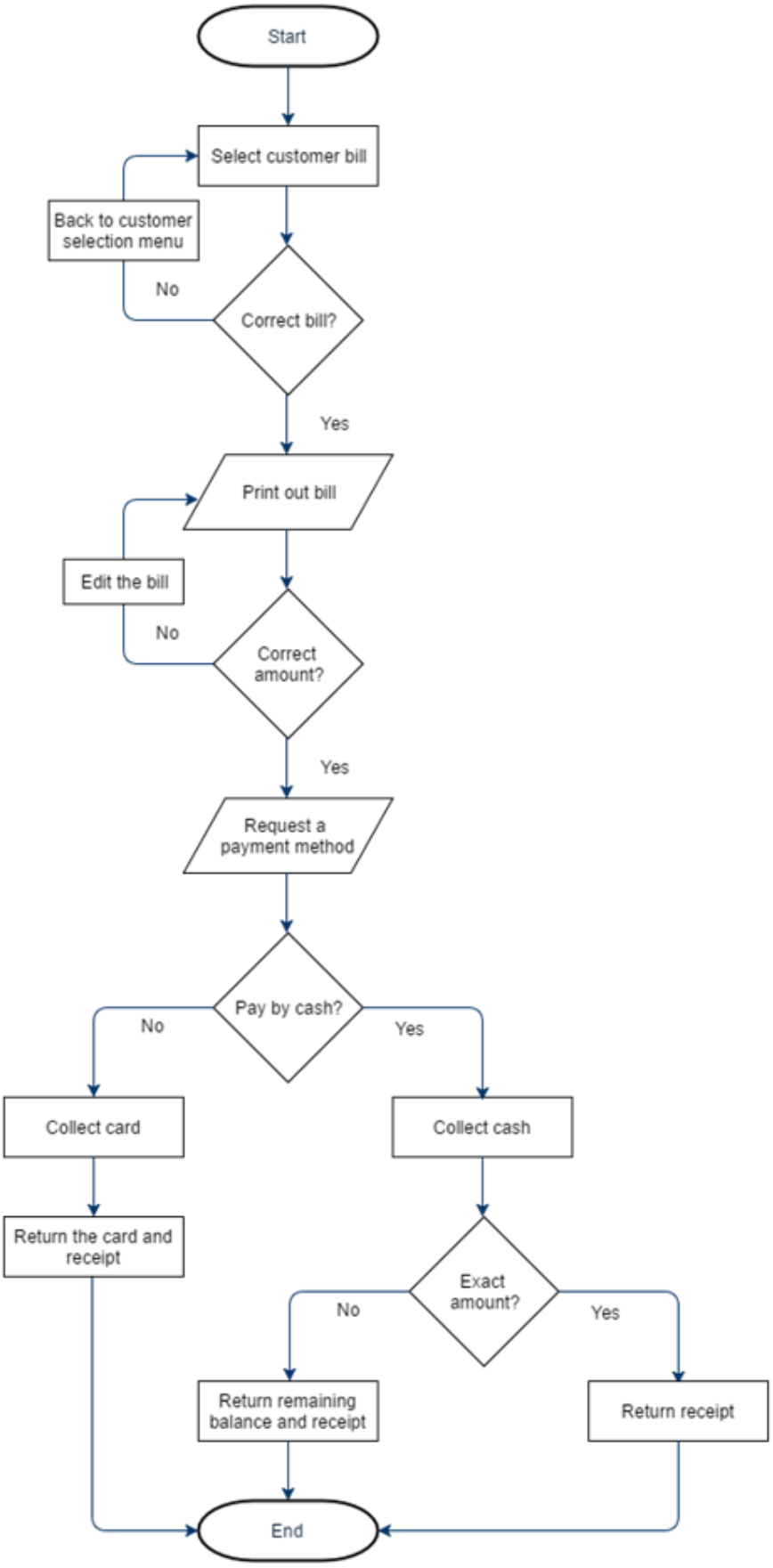
8.1.1 Collect Payment (Counter System)

This component is for manager to collect payment from the customers.

The pseudocode for the component is as below:

```
Start
Select customer bill
If correct bill is selected
        Print out bill
Else
        Back to customer selection menu
Endif
If it is the correct amount
        Request a payment method
Else
        Edit the bill
Endif
If pay by cash
        Collect cash
        If it is an exact amount
                Return receipt
        Else
                Return remaining balance and receipt
        Endif
Else
        Collect card
        Return the card and receipt
Endif
End
```

The flowchart for the component is as below:



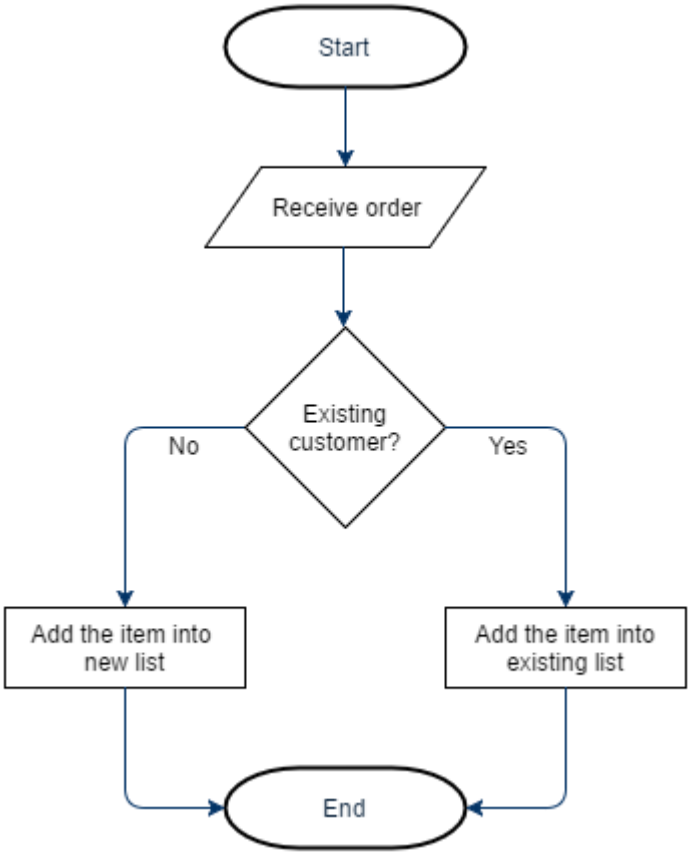
8.1.2 Enter order (Counter System)

This component is for manager to enter order into the system.

The pseudocode for the component is as below:

```
Receive order
If it is existing customer
        Add the item into existing list
Else
        Add the item into new list
Endif
End
```

The flowchart for the component is as below:



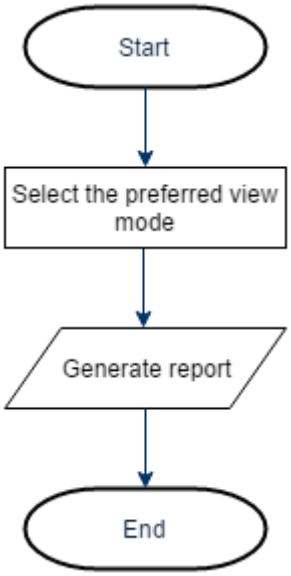
8.1.3 Generate Report (Counter System)

This component is for manager to generate sales report.

The pseudocode for the component is as below:

```
Start
Select the preferred view mode
Generate report
End
```

The flowchart for the component is as below:



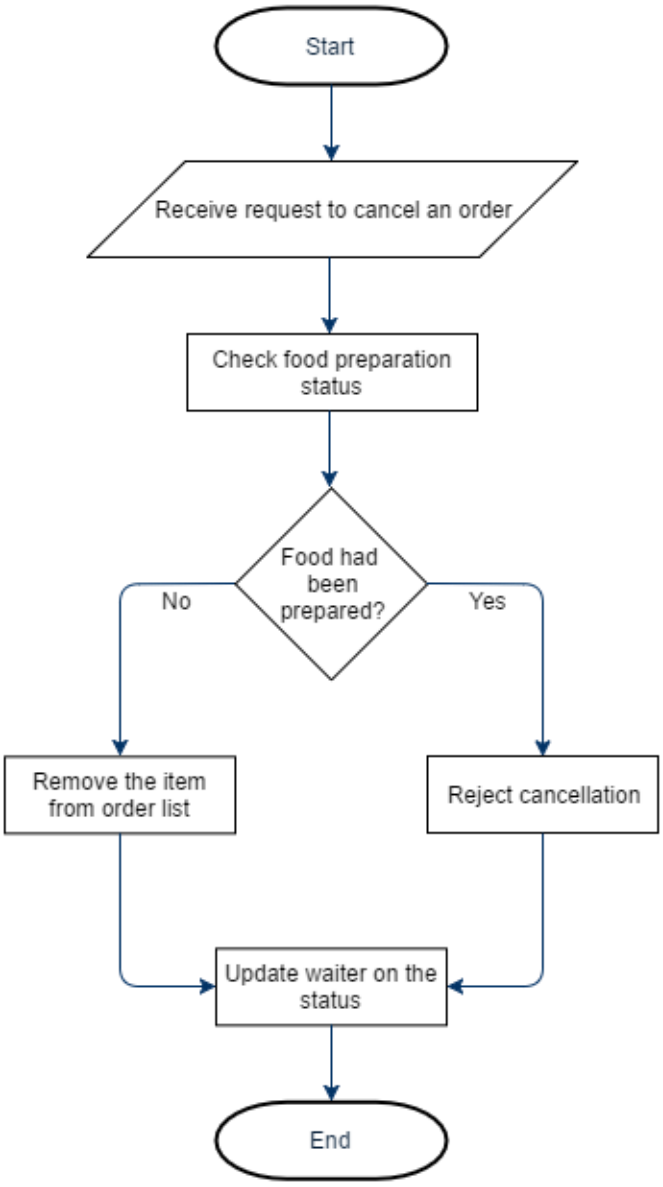
8.1.4 Process Cancellation (Counter System)

This component is to notify manager on the cancellation status.

The pseudocode for the component is as below:

```
Start
Receive request to cancel an order
Check food preparation status
If food had been prepared
        Reject cancellation
Else
        Remove the item from order list
Update waiter on the status
Endif
End
```

The flowchart for the component is as below:



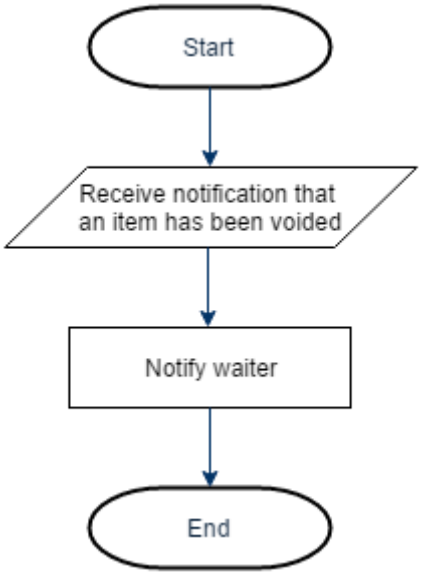
8.1.5 Process Void (Counter System)

This component is to notify manager that an item has been voided.

The pseudocode for the component is as below:

```
Start
Receive notification that an item has been voided
Notify waiter
End
```

The flowchart for the component is as below:



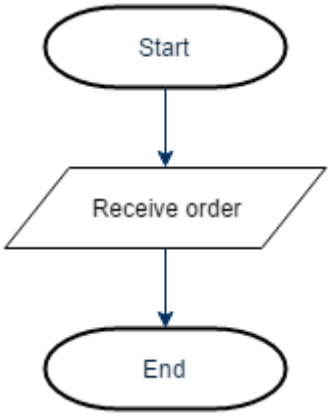
8.1.6 Accept Item (Kitchen System)

This component is for chef to receive orders placed by the manager.

The pseudocode for the component is as below:

```
Start
Receive order
End
```

The flowchart for the component is as below:



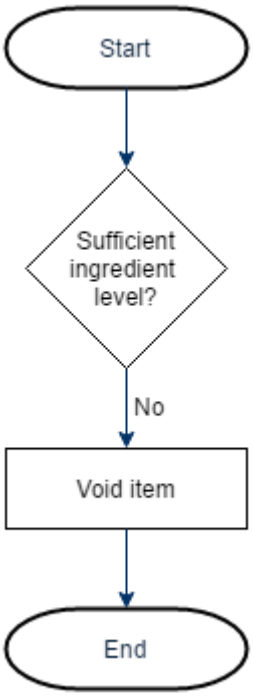
8.1.7 Void Item (Kitchen System)

This component is for chef to void an item when the ingredient level is insufficient.

The pseudocode for the component is as below:

```
Start
If ingredient level is insufficient
        Void item
Endif
End
```

The flowchart for the component is as below:



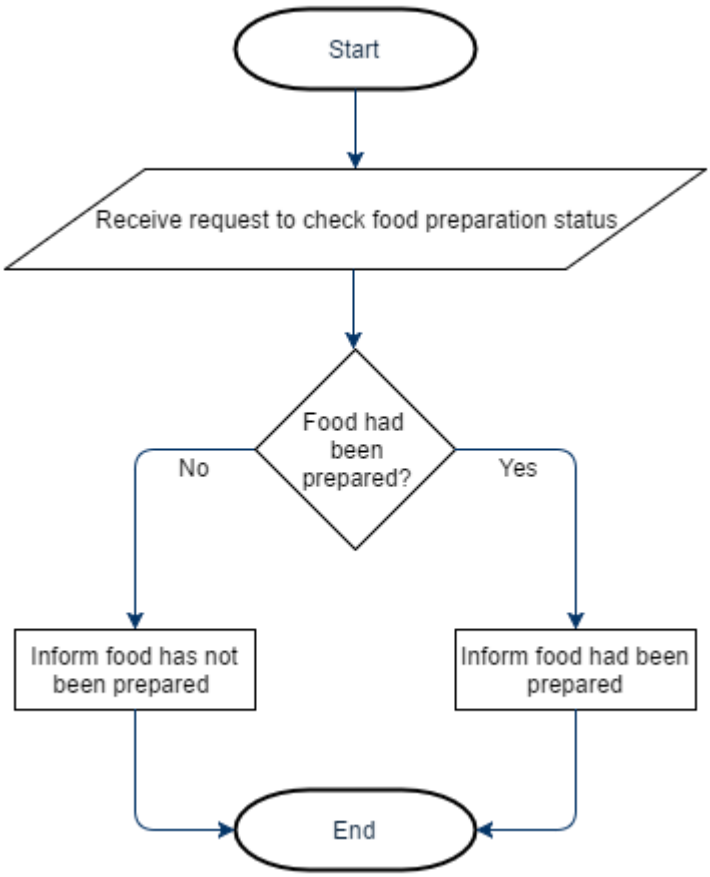
8.1.8 Terminate Item (Kitchen System)

This component is for chef to check if an item has been prepared when a customer wants to cancel the item.

The pseudocode for the component is as below:

```
Start
Receive request to check food preparation status
If the food had been prepared
        Inform food had been prepared
Else
        Inform food has not been prepared
Endif
End
```

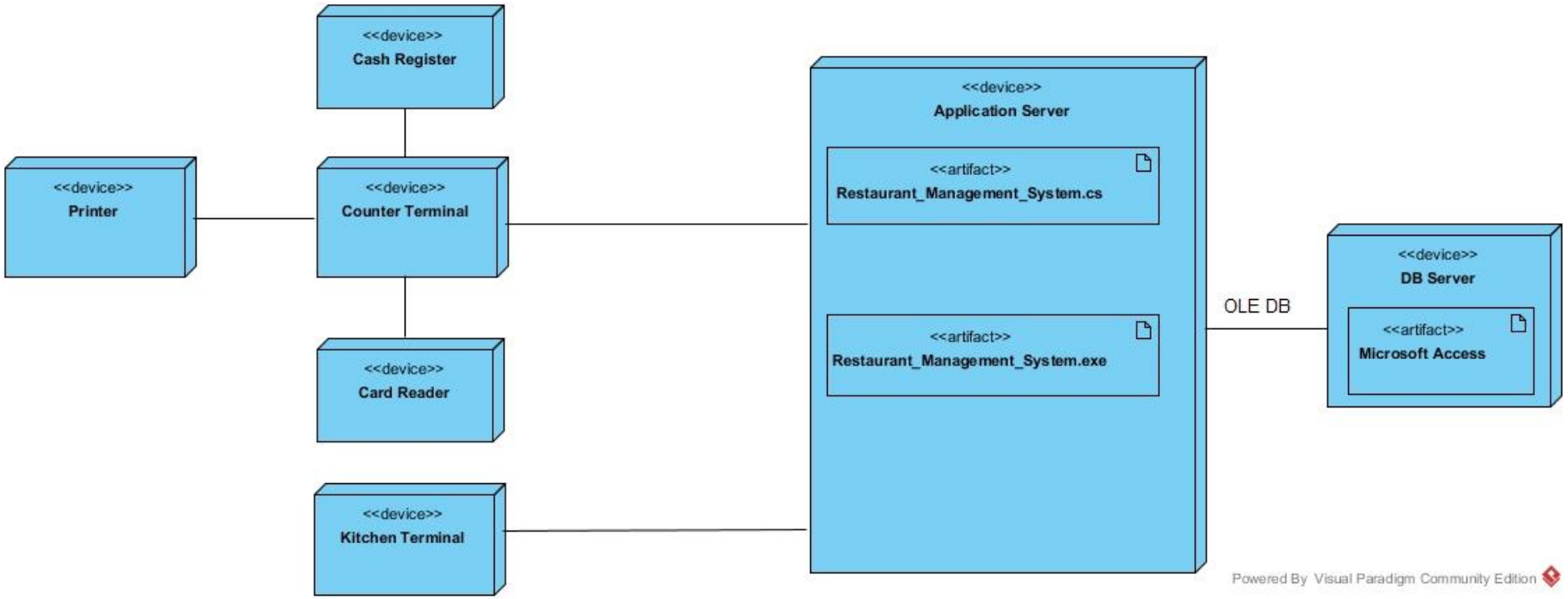
The flowchart for the component is as below:



9 Deployment Design

9.1 Deployment Diagram

The deployment diagram for the system is as below:



9.2 Design

The main components of the deployment design are explained below:

1. Counter Terminal

A device where the Manager enters commands or data for a computer system and also to display the collected output.

i) Printer

To print the receipt of the customer after the customer pays the bill.

ii) Card Reader

To check whether the debit or credit card is applicable and sufficient to pay for the bill. If sufficient then the amount will be deducted from the debit or credit card.

iii) Cash Register

To return the balance amount to the customer after the customer pays the bill.

2. Kitchen Terminal

A device where the Chef enters commands or data for a computer system and also to display the collected output.

3. Application server

Component-based product that exists in the middle layer of a centric architecture. Supplies middleware services. (State maintenance, data access and persistence)

4. DB server

Computer program that supplies database services to other computers or computer programs.

10 Test Data

10.1 Login Screen

Login Username	Login Password	Validity
admin	admin	Valid
Admin	admin	Invalid
Abc	1	Invalid

10.2 Add New Item to Order List

Table No	Item Type	Item Name	Quantity	Item Price (RM)	Validity
1	Food	Pizza	1	8	Valid
2	Beverage	Mocha	2	6	Valid
Abc	Food	Nasi Lemak	1	5	Invalid
3	Beverage	Orange Juice	2	abc	Invalid

10.3 Cancel Item from Order List

Table No	Item Type	Item Name	Quantity	Validity
1	Food	Chicken Chop	2	Valid
2	Beverage	Coffee	3	Valid
Abc	Food	Spaghetti	6	Invalid
5	Beverage	Orange Juice	Abc	Invalid

10.4 Add Item to Menu

Item Name	Item Type	Item Price (RM)	Validity
Lamb Chop	Food	12	Valid
Coke	Beverage	2.50	Valid
Mushroom soup	Food	Abc	Invalid
Pepsi	Beverage	C	Invalid

10.5 Delete Item from Menu

Item Name	Validity
Pizza	Valid
12345	Invalid

10.6 Cash Payment

Paid	Validity
20.00	Valid
10.00	Valid
	Invalid

11 Acceptance Test

11.1 Case 1: Login Page

Step	Criteria	Fulfill?	Remark
1.	The manager is allowed to enter his or her credentials into the system.	Yes / ~~No~~	
2.	If username and password is correct, the login will be successful.	Yes / ~~No~~	
3.	The manager cannot get access to the system if he or she use the wrong username and password.	Yes / ~~No~~	

11.2 Case 2: New Item (Manager only)

Step	Criteria	Fulfill?	Remark
1.	User selects table no to state the order of a particular table.	Yes / ~~No~~	
2.	User can select the item type to add into order.	Yes / ~~No~~	
3.	User can select the item name to add into order.	Yes / ~~No~~	
4.	User can choose the quantity of the item needed.	Yes / ~~No~~	
5.	If no input is entered and the user presses add item, a warning message will be shown.	Yes / ~~No~~	

11.3 Case 3: Cancel Item (Manager only)

Step	Criteria	Fulfill?	Remark
1.	User clicks the cancel item option.	Yes / ~~No~~	
2.	User selects the table no.	Yes / ~~No~~	
3.	User selects the item type.	Yes / ~~No~~	
4.	User selects the item name.	Yes / ~~No~~	
5.	User can proceed to cancel the item when all the requirements on top are filled.	Yes / ~~No~~	

11.4 Case 4: Modify Menu (Manager Only)

Step	Criteria	Fulfill?	Remark
1.	User selects modify menu to add new item to the menu.	Yes / ~~No~~	
2.	User fills in the new item details by adding the new item name, item type and price.	Yes / ~~No~~	
3.	User can choose to delete any added item.	Yes / ~~No~~	
4.	If the price is not a decimal, a warning message will be shown.	Yes / ~~No~~	

11.5 Case 5: Card Payment (Manager Only)

Step	Criteria	Fulfill?	Remark
1.	The customer is allowed to pay by card.	Yes / ~~No~~	
2.	The manager will read the card using an external machine.	Yes / ~~No~~	
3.	The system then will show the card details such as card number, bank name and card status.	Yes / ~~No~~	
4.	Manager have to continue to press done if the card status is approved in order to complete the payment.	Yes / ~~No~~	

11.6 Case 6: Generate Report (Manager Only)

Step	Criteria	Fulfill?	Remark
1.	User selects the mode to view the report.	Yes / ~~No~~	
2.	User views the report sorted by daily.	Yes / ~~No~~	
3.	User views the report sorted by monthly.	Yes / ~~No~~	
4.	User views the report sorted by yearly.	Yes / ~~No~~	
5.	User views the report sorted by quantity	Yes / ~~No~~	

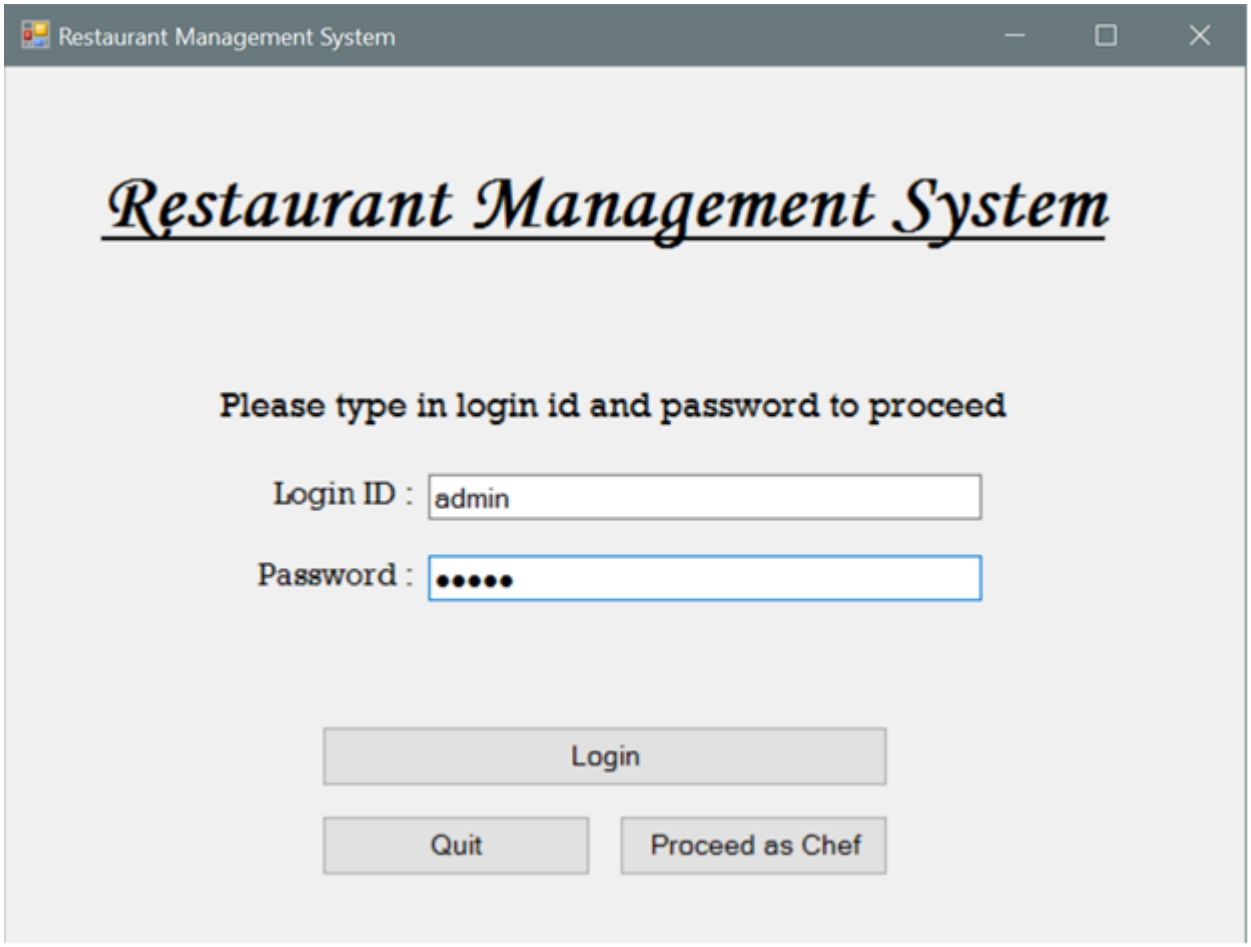
11.7 Case 7: Cash Payment (Manager only)

Step	Criteria	Fulfill?	Remark
1.	User clicks the table of the customer whom going to pay on the screen.	Yes / ~~No~~	
2.	The total due is shown to the customer through the hardware.	~~Yes~~ / No	It will show to the customer only if the hardware is connected.
3.	User keys in the amount given by the customer.	Yes / ~~No~~	
4.	The user then clicks the open drawer button to get the change.	~~Yes~~ / No	It will prompt that the hardware device is missing.
5.	The amount of balance to be returned is shown in the system.	Yes / ~~No~~	
6.	User prints the receipt for the customer and gives back the remaining change.	~~Yes~~ / No	It will prompt that the hardware device is missing.

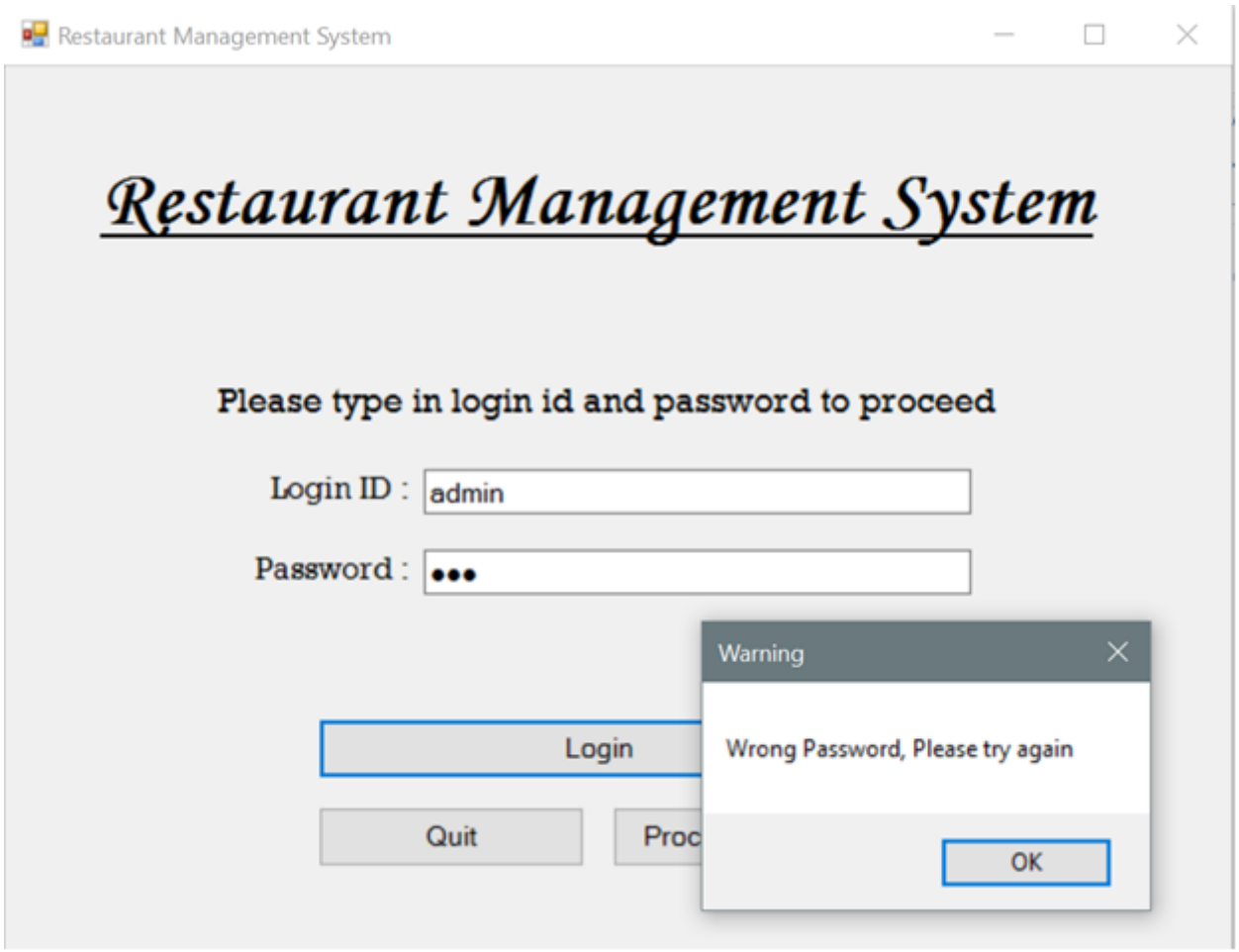
12 Sample Screens

12.1 Counter System (Manager)

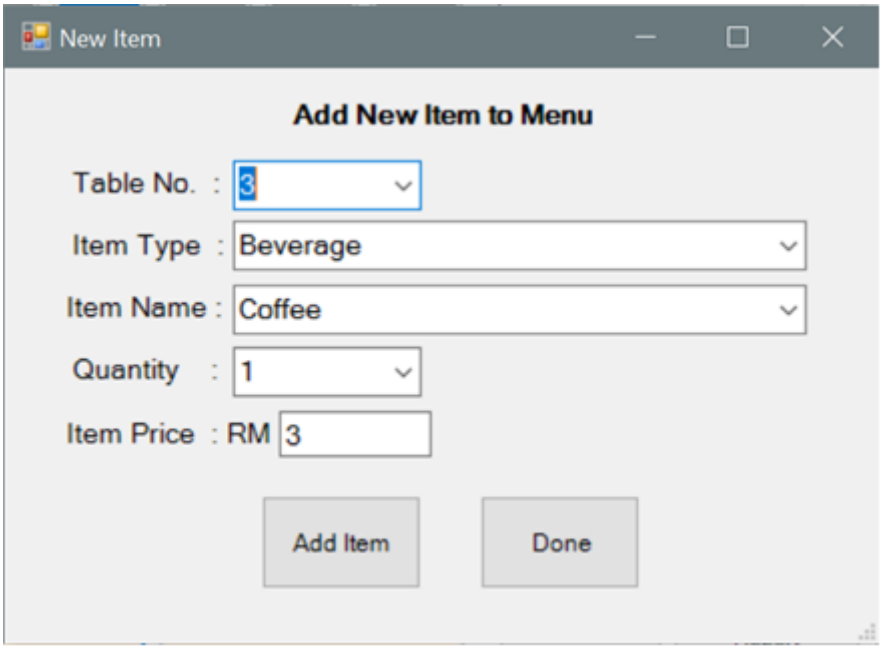
Manager can log in using his or her credentials into system. If the username and password is correct then the login will be successful:

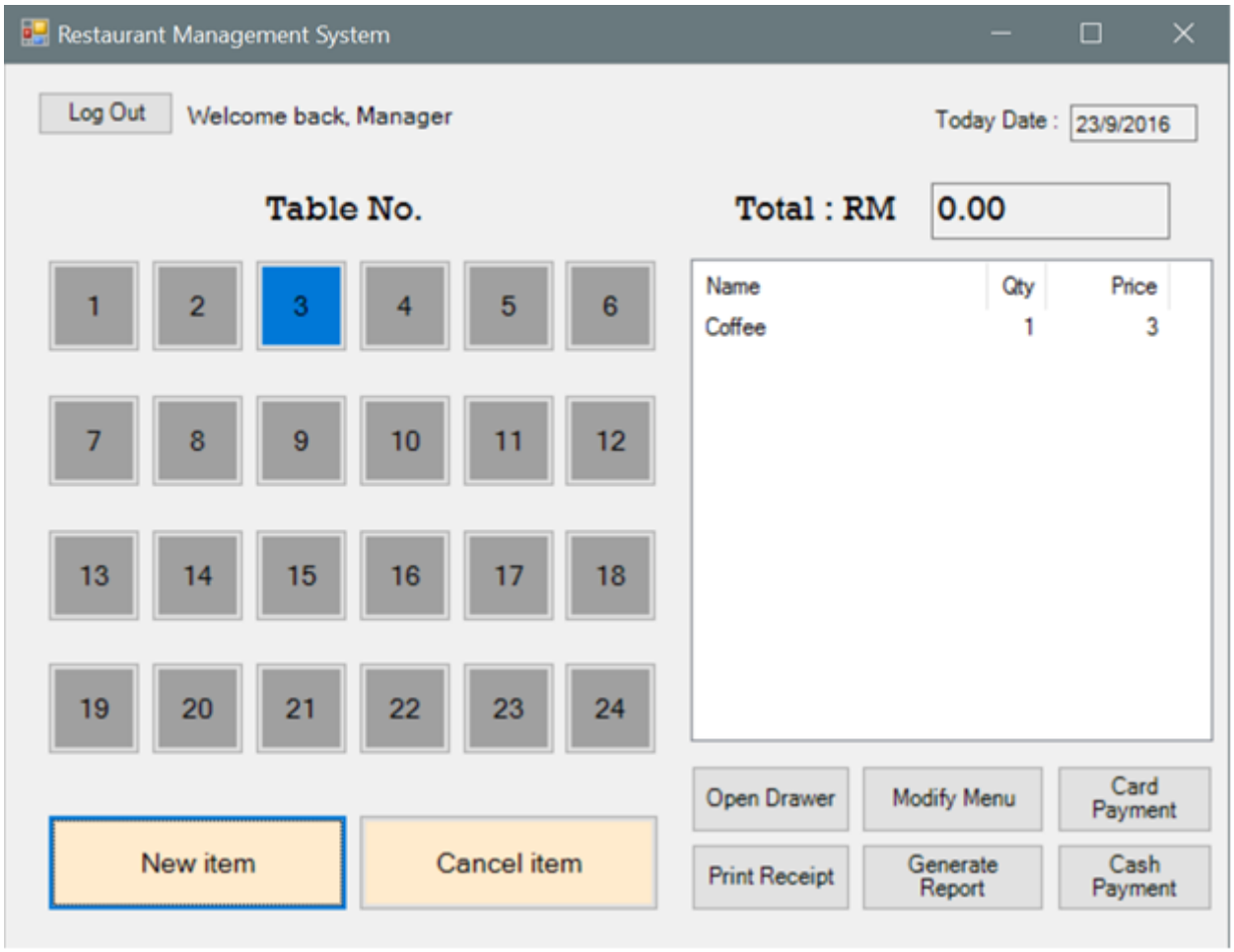


An error message will be prompted when the username and the password entered is wrong:

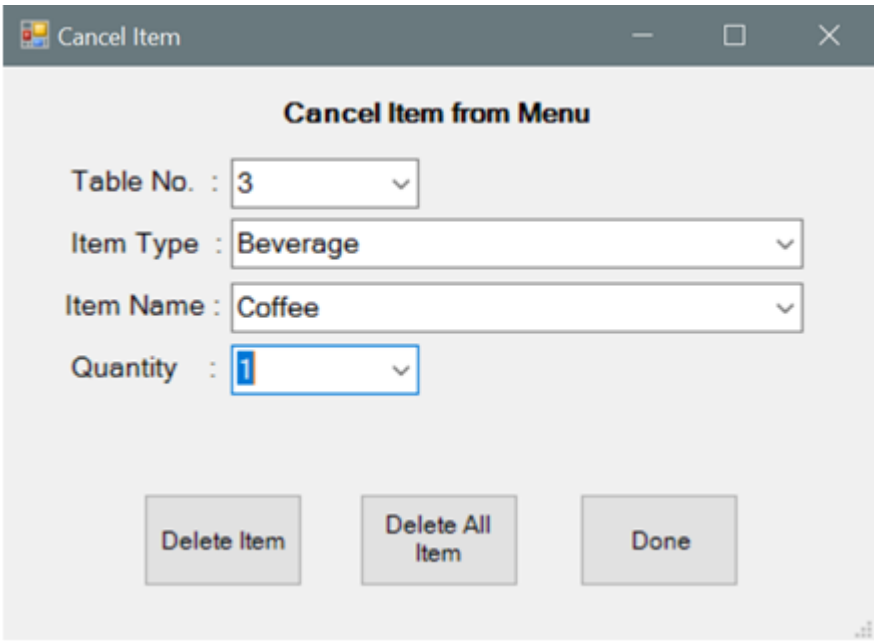


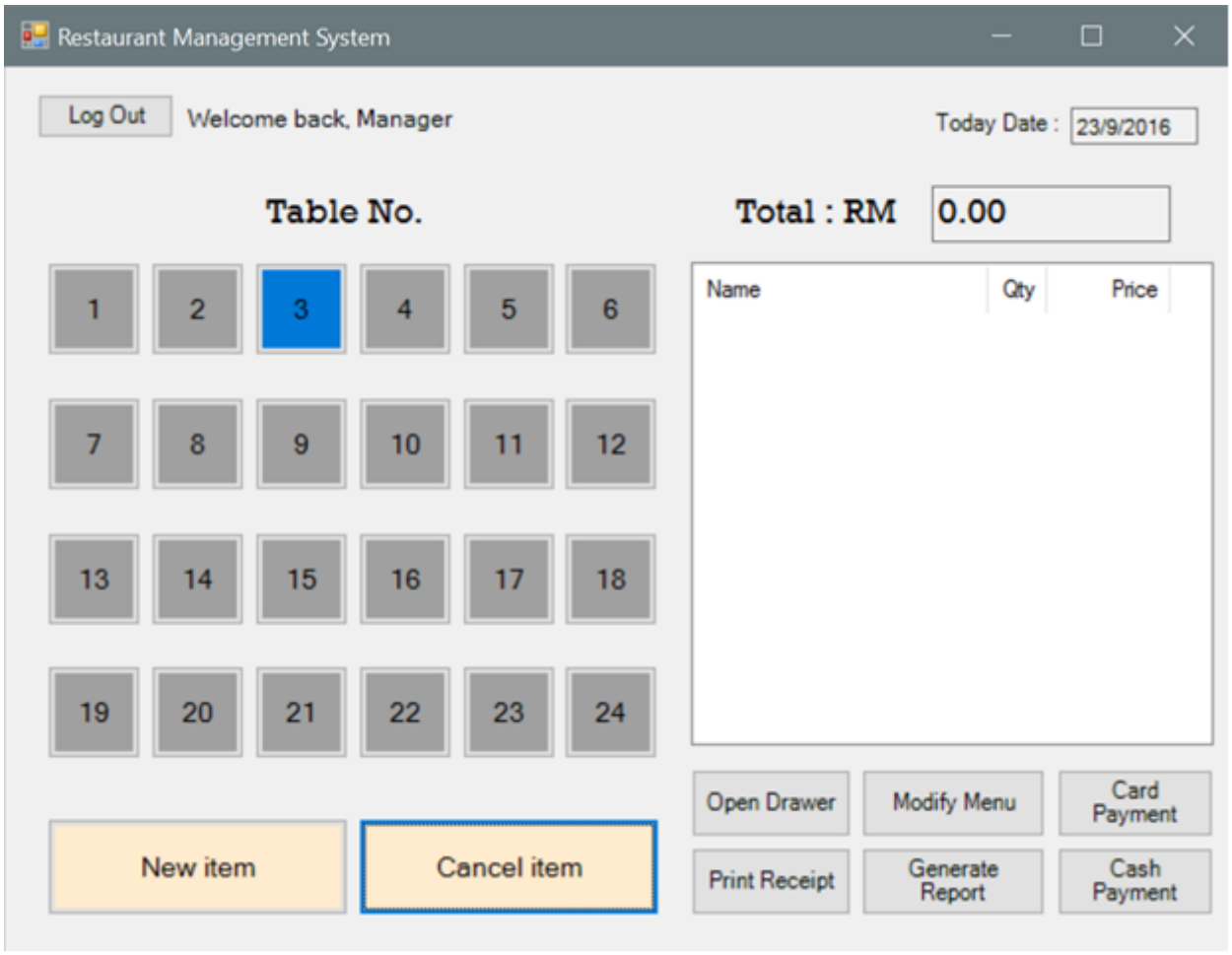
Manager can add in new item to the menu when a customer orders food or beverage:



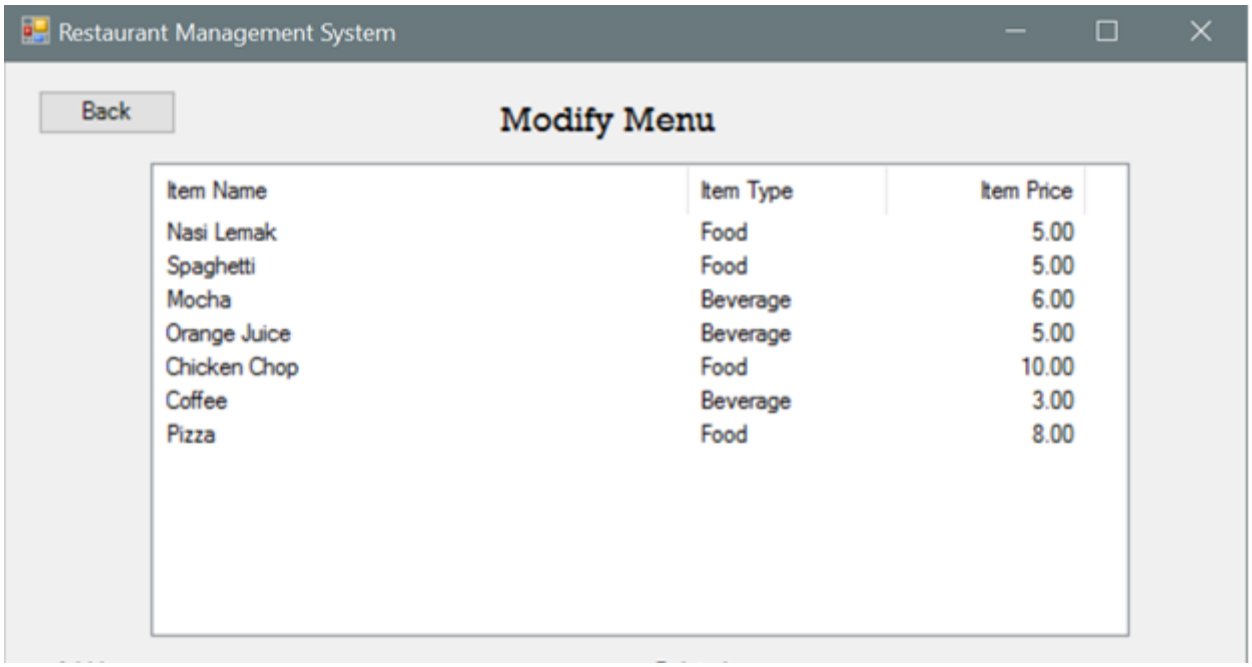


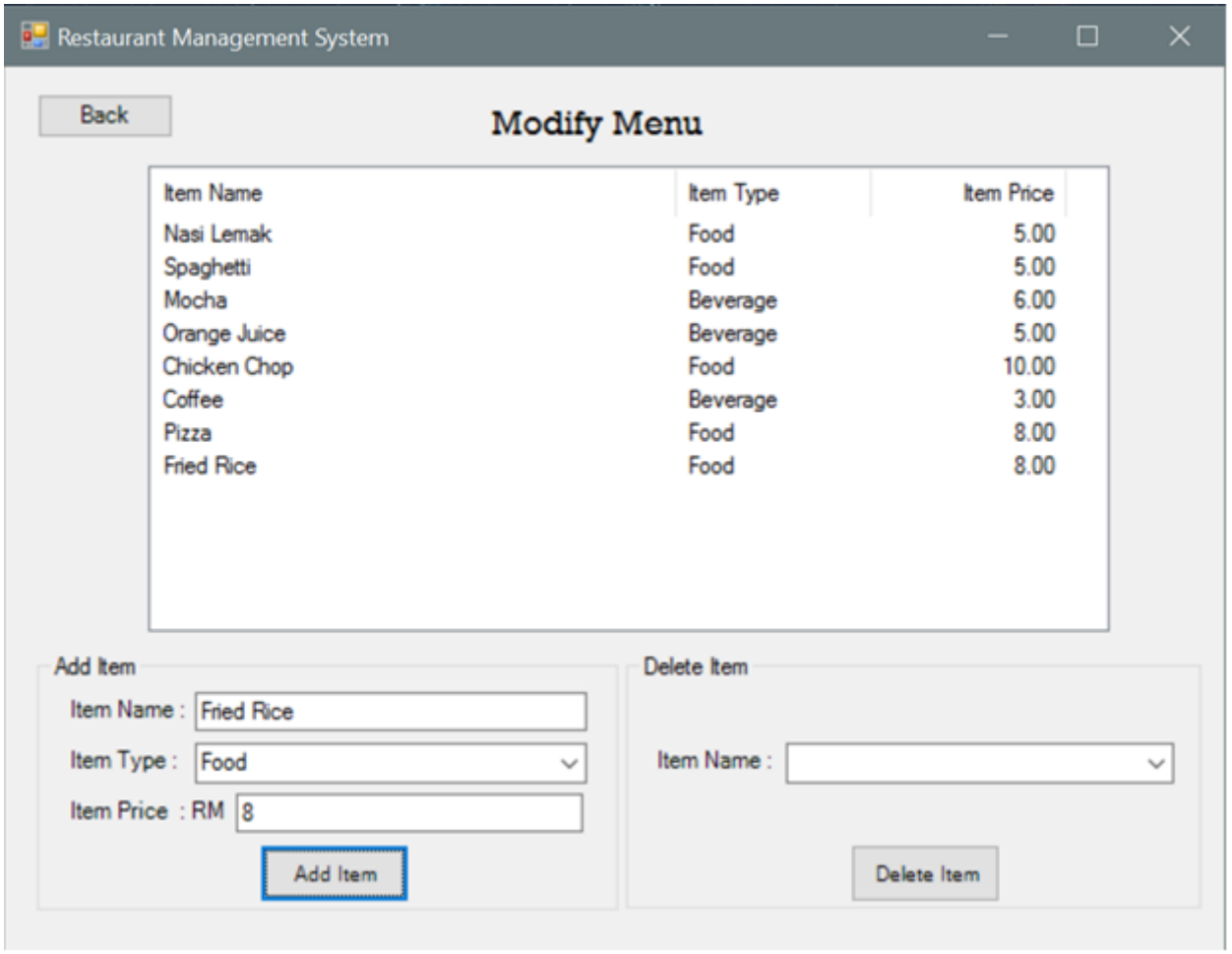
Manager can cancel and item from the menu, if the customer has sudden change in mind:



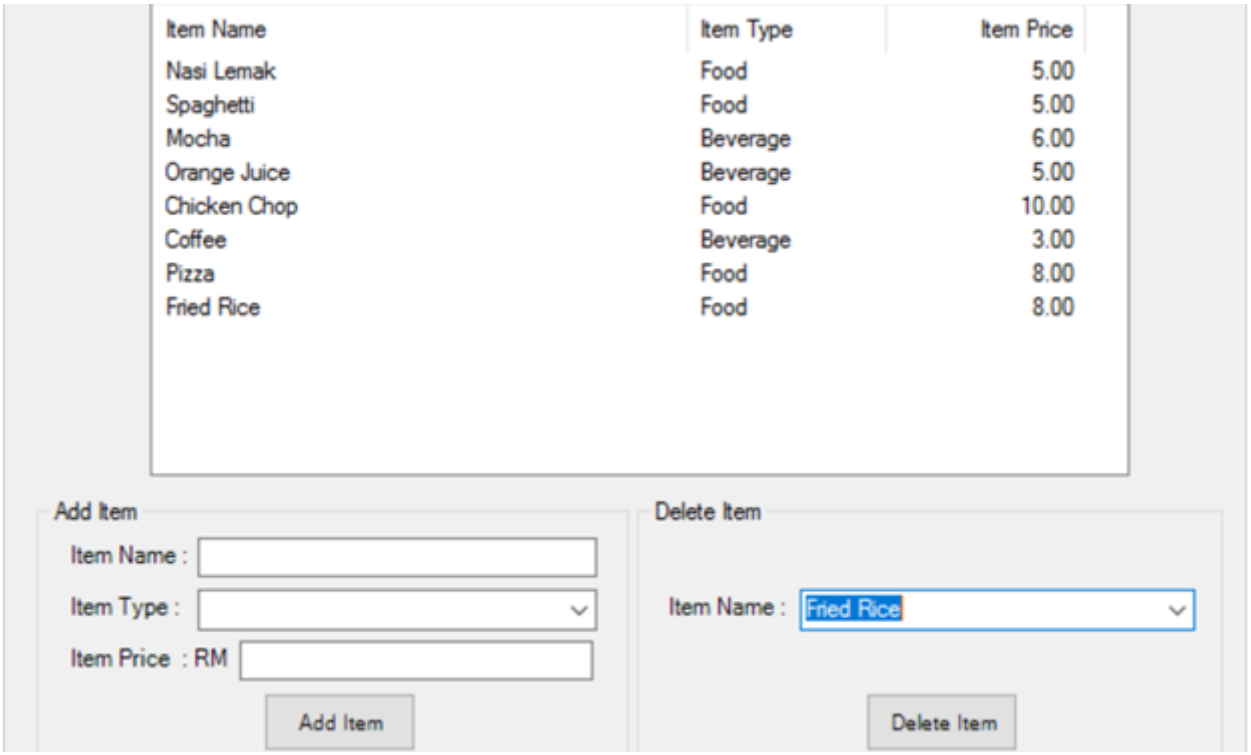


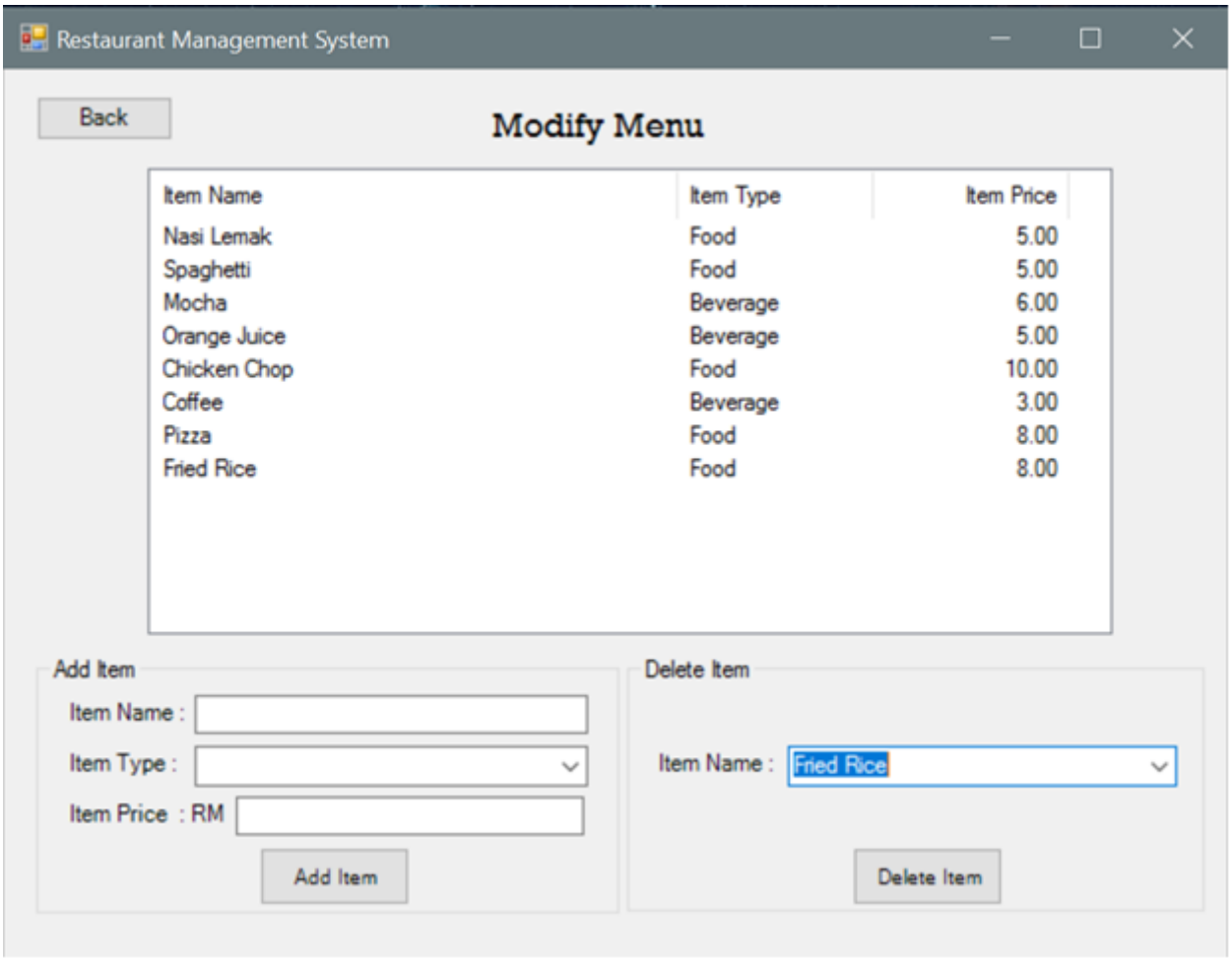
Manager can create new items in the modify menu by entering the item name, item type and item price:



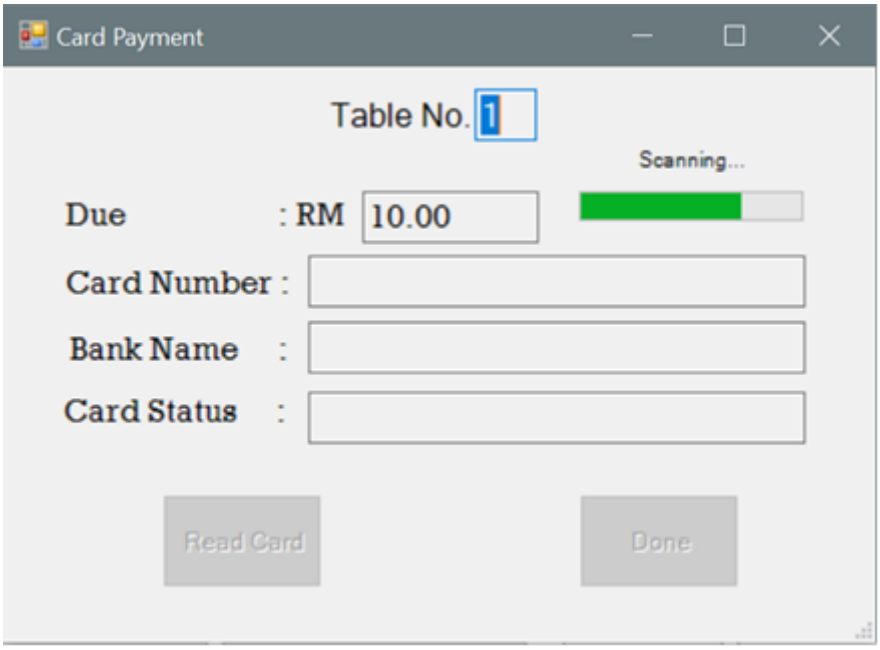


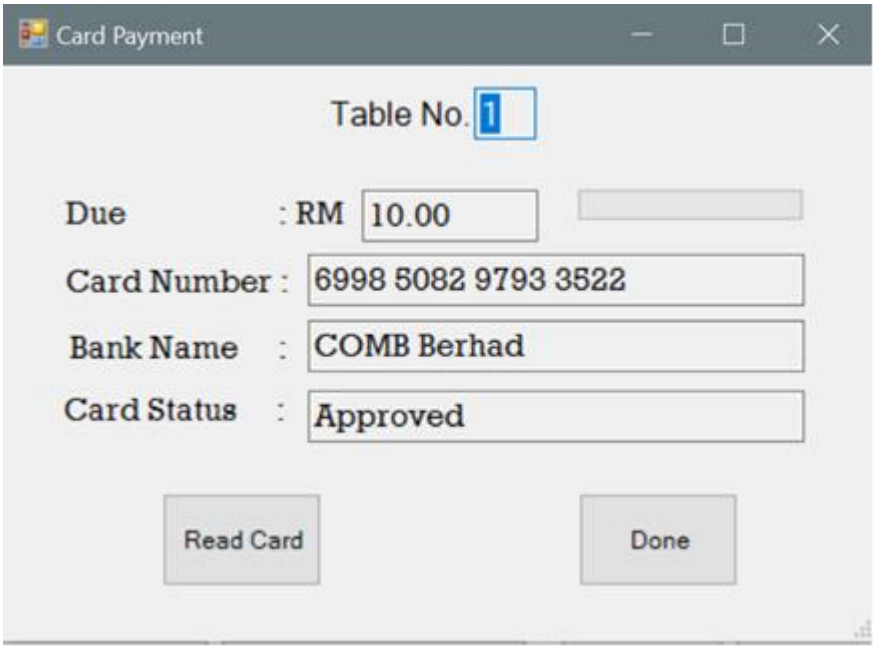
Manager can delete an item by choosing the item name:



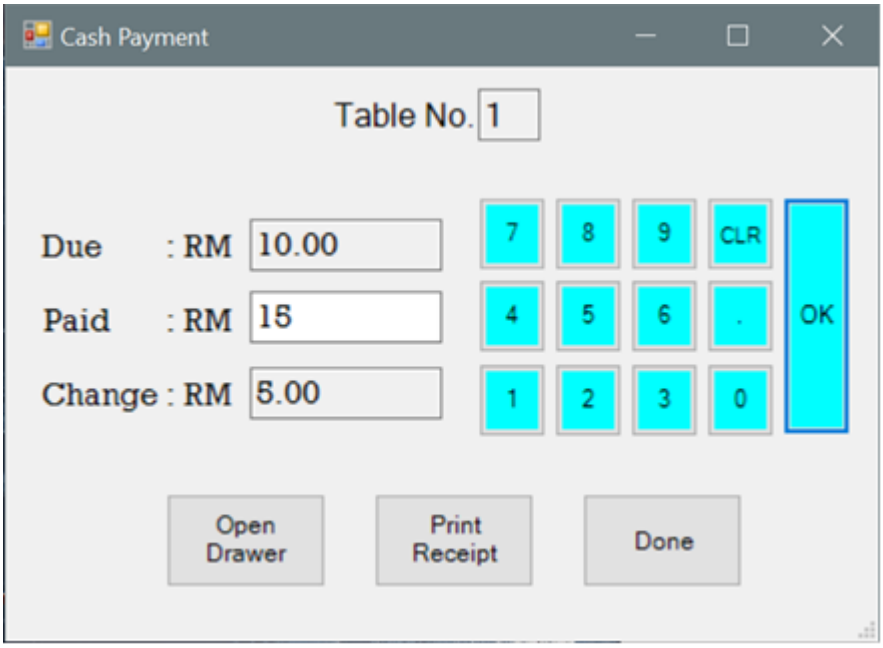


Customer can make payment by using their card:

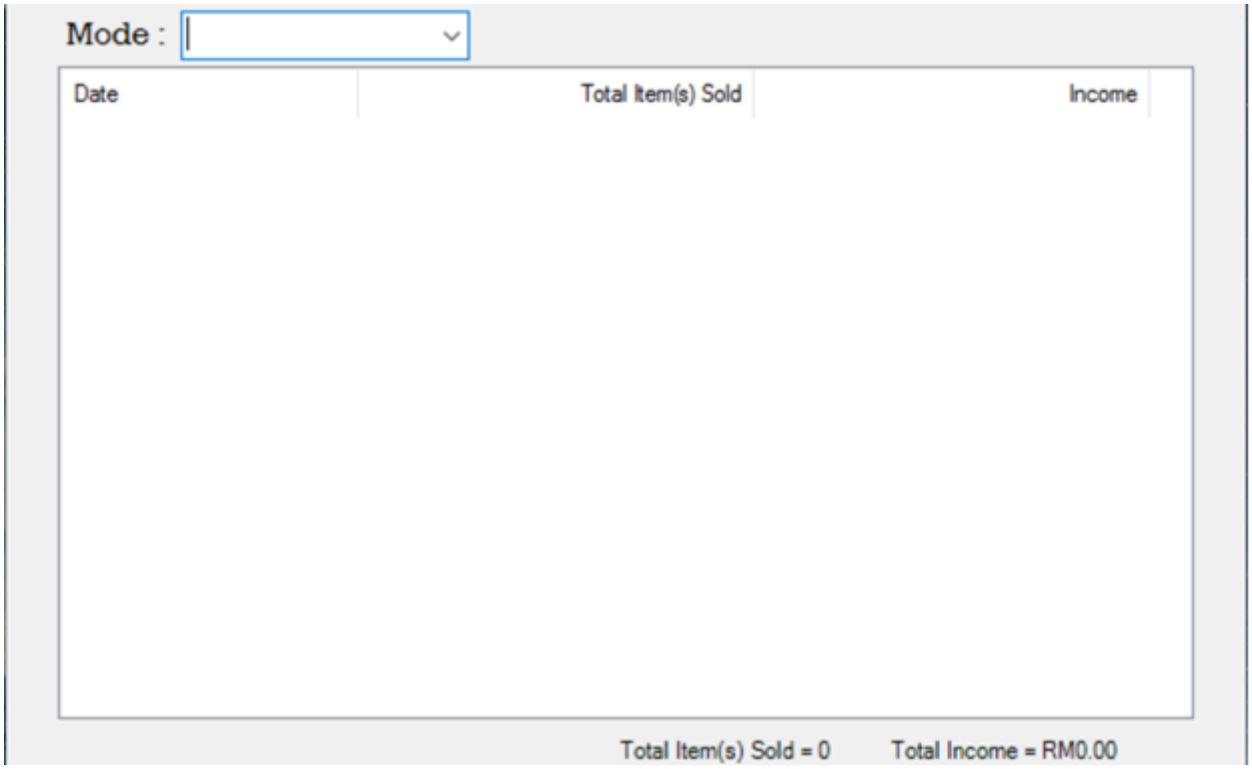


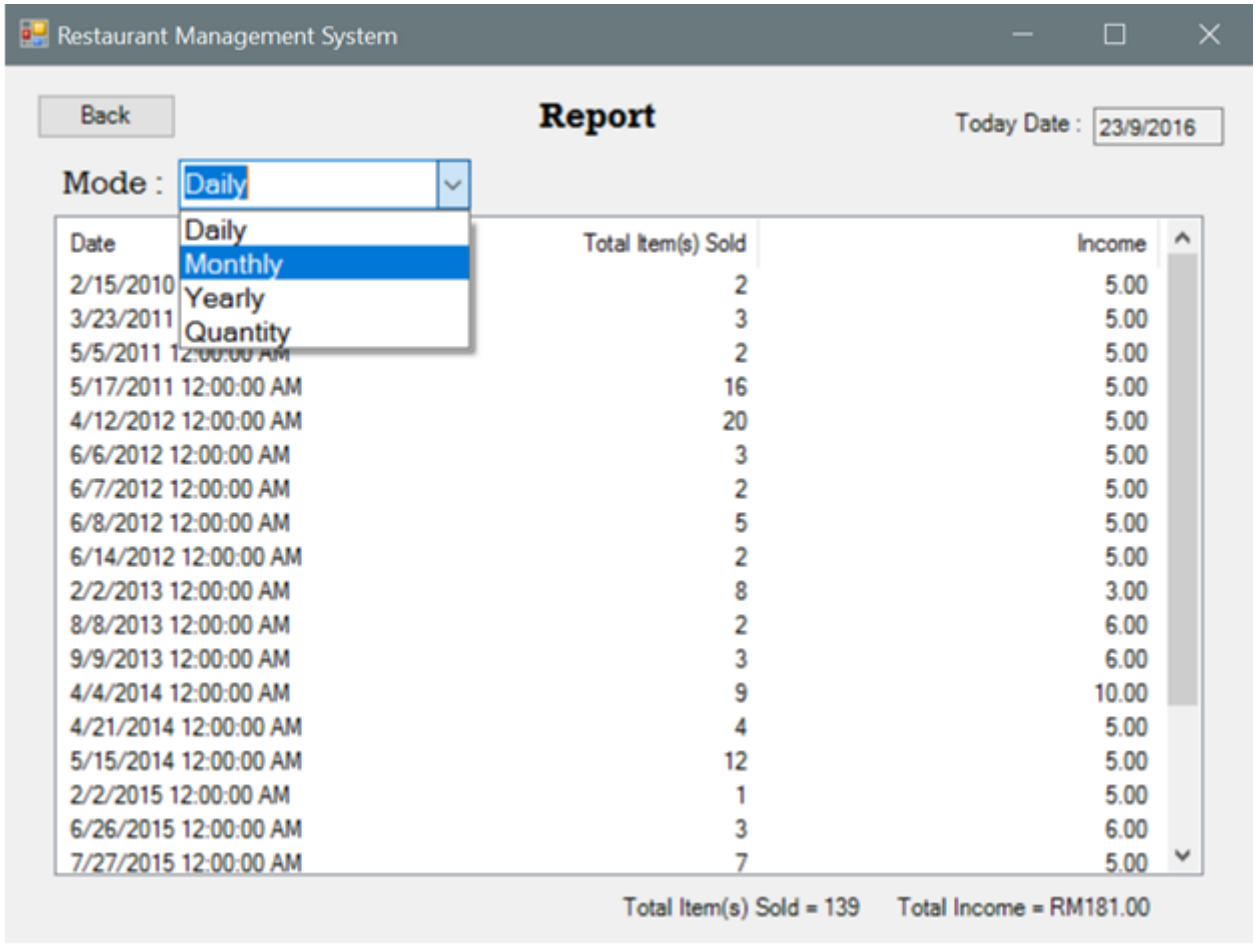


Customer can also make payment by using cash:

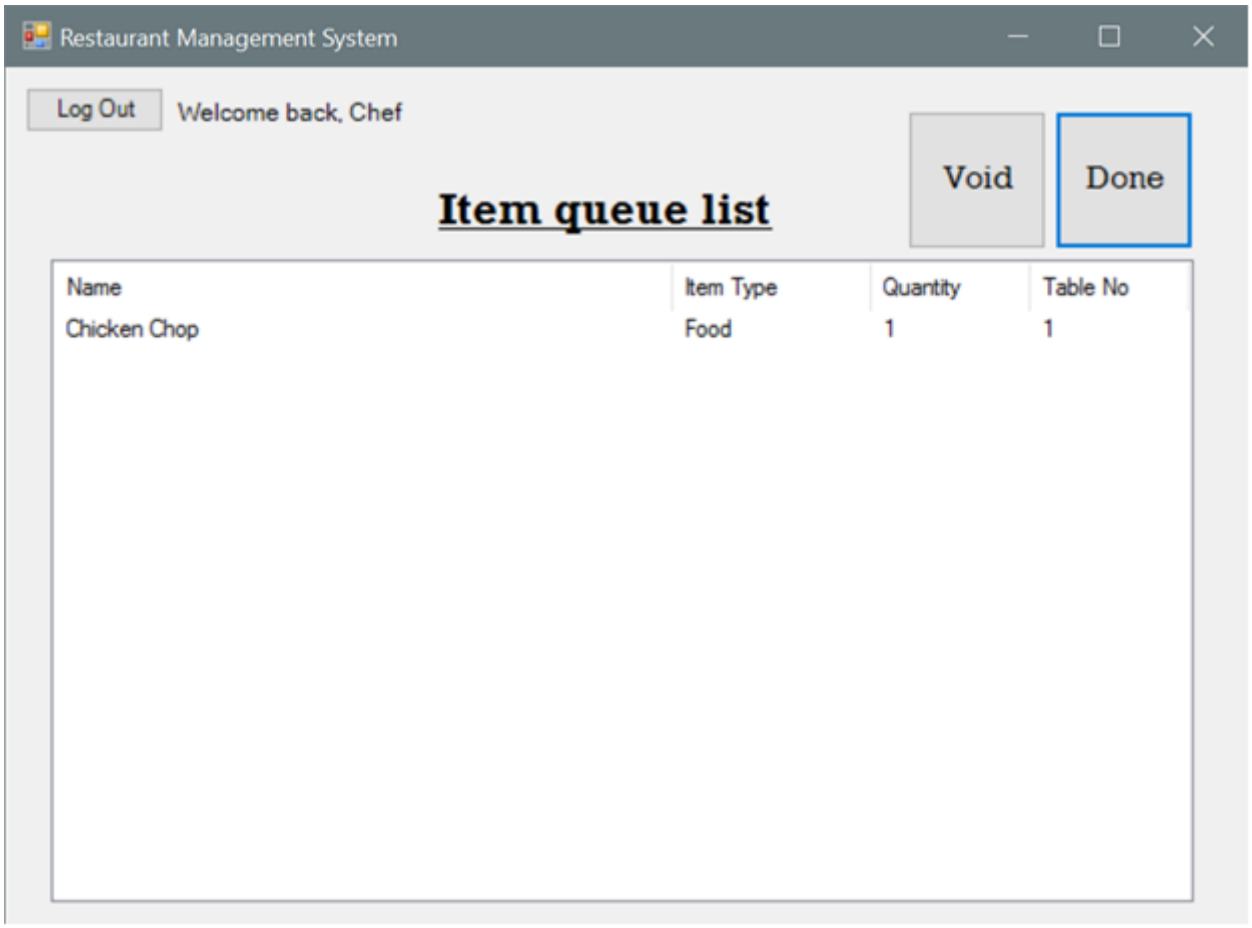


Manager can see his restaurant's report by daily, monthly, yearly and quantity:





Chef uses item queue list in order to keep track of all the items ordered:



13 Conclusion

13.1 Summary of Results

In the process of this project we have built a Restaurant Management System (RMS) that achieved to perform all the basic functions of a working Point of Sale System that the market now has to offer.

We researched and planned carefully in doing software design analysis and software specification in order to bring out the full potential of the RMS. Although faced with a number of constraints we managed to achieve it.

The system is further enforced with error prevention functions in order to reduce the chances of it crashing or causing errors. The screen for the manager and the chef is separated, so the manager or the chef would not mix it up.

In conclusion, we tested and ran the system through many trials until all its functions are made to be errorless and fully functional.

13.2 Problems Encountered

During the process of the project, we were faced with problems at each phase:

At the early stages of requirement specification, the information we had were insufficient for providing us with enough knowledge and experience to build the system. To solve this problem we decided to have group meetings each week to discuss and do research about it. In the end all was made understandable.

At the stage of software design process, a lot of discussion is made between our group members to decide on what we should add into the system. In the end we decided on a standard design following the examples we found. After that we just represented it in the shape of UML models.

Finally, during development phase, we were faced with difficulties in utilizing the development software as it is hard to use due to the complicated interfaces.

13.3 Limitations and Future Enhancements

The current working system even though it is able to perform all the basic functions, but it still has limitations such as the inability to produce hardcopy receipt using the printer device because of no driver programs. Lack of card reader device also makes our system pay by card method to be not fully useable. Furthermore, the system can only be showed using normal PCs. Finally, the platform compatibility is an issue since the system is able to run on windows platform only.

In future, we would like to spend more time and collect enough model to improve the system even further by making it easy to use, platform independent, deployable, and finally to transform it into a fully working system that can help many people.