ITERATION 2

1. Functional Requirements

S NO.	REQUIREMENTS
R1	The customer should be able to register on the system
R2	The admin (manager) should be able to login with valid credentials on the system
R3	The customer should be able to login with valid credentials on the system
R4	The chef should be able to login with valid credentials on the system
R5	The customer should be able to see the menu and choose the desired items
R6	The customer should be able to place an order from the available items in the menu
R7	The admin and chef should be able to view the orders placed by the customers
R8	The admin should be able to update menu, offers and deals for the day
R9	The chef should be able to update the status of the order
R10	The customer should be able to pay for the order using his card credentials
R11	The admin should be able to view the payment status for all the customers
R12	The admin, customer and chef should be able to log out of the system

2. Non- Functional Requirements

I. Maintenance and recoverability of the database containing all the login credentials and menu items.

II. Access permissions for the particular system information may only be changed by the system's data administrator.

3. **Use Cases (with actors)**

UC1 - Login/Logout functionality for every actor

I. Actor 1 :- Chef

UC2 - View current orders

UC3 - Update order status

II. Actor 2 :- Customer

UC4 - View available menu items

UC5 - Place order

UC6 - Make payment

III. Actor 3 :- Admin

UC7 - View Orders

UC8 - Add/Edit menu items

UC9 - View and manage payments

4. **User stories**

Actor 1 - Chef:

- As chef of the restaurant, I can check all the pending orders so that the pending orders can be completed in time.

- As chef of the restaurant, I can update the status of the order from pending to completed.

Actor 2 - Customer:

- As a customer, I can view available menu items so that I can decide my order without the ambiguity on the availability or non-availability of any food item on the menu.

- As a customer, I can place an order so that I do not have to wait for the waiter to do it for me.

- As a customer, I can make payment in any method so that I do not have to wait for the waiter to take the payment even though I have finished dining.

Actor 3 - Admin:

- As an admin, I can view all the orders so that I can keep a log of all orders which may or may not be needed in the future.

- As an admin, I can manage the menu for the restaurant without putting any load on the chef.

- As an admin, I can manage and view payments so that any kind of payment discrepancies can be resolved easily.

5. **Expanded Use-Case with Pre/Post Conditions**

UC1: Login/Logout Functionality for every actor

Precondition: This use case assumes that the actors (admin, chef, customer) have landed on

the welcome page of the system that requires login.	
Actor: Admin, Chef, Customer	**System: Login/Logout**
0. TUCBW the actor entering credentials to log into the system.	1. System logs the user in to their respective account once the correct credentials are entered.
Postcondition: The actors have successfully logged out of the system and again landed on the welcome page.	

UC2: View current orders

Precondition: This use case assumes that the chef has already logged in and is on the 'current orders' page.	
Actor: Chef	**System: Current Order Page**
1. The actor sees items listed as current orders placed by him.	0. System gives a listed view of all the current orders.
Postcondition: The actor has successfully logged out of the portal or is on another page.	

UC3: Update status of pending orders

Precondition: This use case assumes that the actor is logged in and needs to update the status of completed orders	
Actor: Chef	**System: Pending Orders Page**
0. The actor clicks on the 'Orders' tab and	1. When clicked on 'Orders' tab, the

checks the orders that are yet to be completed.	system screen is switched to a list which shows placed orders. 2. The chef can update the status of orders from 'processing' to 'completed'.
Postcondition: The actor has successfully updated the status of orders that were previously in processing state.	

UC4: View available menu items

Precondition: This use case assumes that the actor has logged in successfully and is on the main page.	
Actor: Customer	**System: Menu Page**
1. The actor sees a list of all the available menu items. 3. The actor selects items from the menu.	0. System screen is the Menu Page. 2. Screen allows the actor to select items in the menu list.
Postcondition: The actor has successfully navigated through the menu page and has selected item/items from the menu.	

UC5: Place order

Precondition: This use case assumes that the actor has successfully selected items to be ordered from the menu page.	
Actor: Customer	**System: Place order confirmation screen**
1. The actor places the order or cancels	0. System screen is navigated to the page

the order.	where the actor is supposed to either place or cancel the order.

Postcondition: If the actor has decided to cancel the order, the actor is navigated to the Menu Page. If the actor has decided to place an order, the actor is navigated to the 'Make Payment' page.

UC6: Make payment

Precondition: This use case assumes that the actor has already placed the order and is navigated to the next screen i.e. 'Make Payment'	
Actor: Customer	**System: Make Payment screen**
1. The actor chooses a payment method. .3. The actor pays for the order. 5. The actor receives a receipt.	0. System screen is the payment page where the actor can choose payment methods. 2. System screen then navigates to payment gateway. 4. System screen after payment confirmation goes to the receipt screen.
Postcondition: The actor has now made the payment and can navigate back to the 'Menu Page'.	

UC7: View Orders

Precondition: This use case assumes that the actor has logged in successfully using valid credentials and is on the home page from where the actor can navigate to different screens.	
Actor: Admin	**System: View Orders screen**

1. The actor can view all the orders on this screen. 3. This actor can edit, or delete all the orders.	0. System screen contains a list of all (complete and pending) orders. 2. System screen also includes an edit button for the actor to edit or delete all the orders.
Postcondition: The actor has full read and write authority to alter the screen and navigate to other screens.	

UC8: Manage menu

Precondition: This use case assumes that the actor has logged in successfully using valid credentials and is on the home page from where the actor can navigate to different screens.	
Actor: Admin	**System: Menu Page**
1. The actor has read and write authority for the Menu Page. 3. The actor edits or deletes content from the Menu Page.	0. The system screen has the menu page. 2. The system screen has an edit button which helps in editing or deleting food items from the menu.
Postcondition: The actor has full read and write authority to alter the screen and navigate to other screens.	

UC9: View and manage payments

Precondition: This use case assumes that the actor has logged in successfully using valid credentials and is on the home page from where the actor can navigate to different screens.	
Actor: Admin	**System: Payment Page**
1. The actor can view all the payments	0. System screen has the payments list.

| made by customers. 4. The actor can manage payment in case of payment discrepancies.	2. System screen is different from the customer's payment screen. 3. System screen contains manage button, which helps the actor to regulate payment discrepancies.
Postcondition: The actor has full read and write authority to alter the screen and navigate to other screens.	

6. Requirements - Use Case Traceability Matrix

	Priority Weight	UC1	UC2	UC3	UC4	UC5	UC6	UC7	UC8	UC9
R1	3	x								
R2	3	x								
R3	3	x								
R4	3	x								
R5	2				x					
R6	2					x				
R7	1		x					x		
R8	1								x	

R9	1			x						
R10	3						x			
R11	3									x
R12	3	x								
Score	15	1	1	2	2	3	1	1	3

7. Sequence Diagrams

● **Login System**
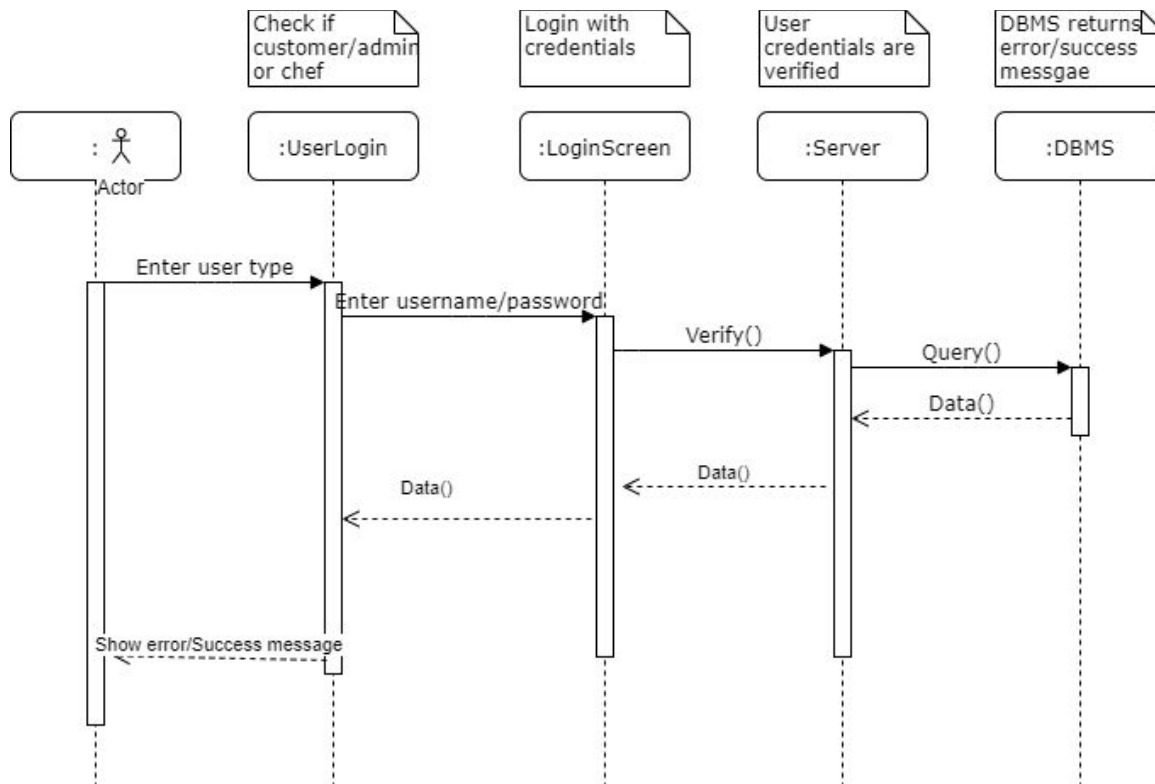
● **View menu items**
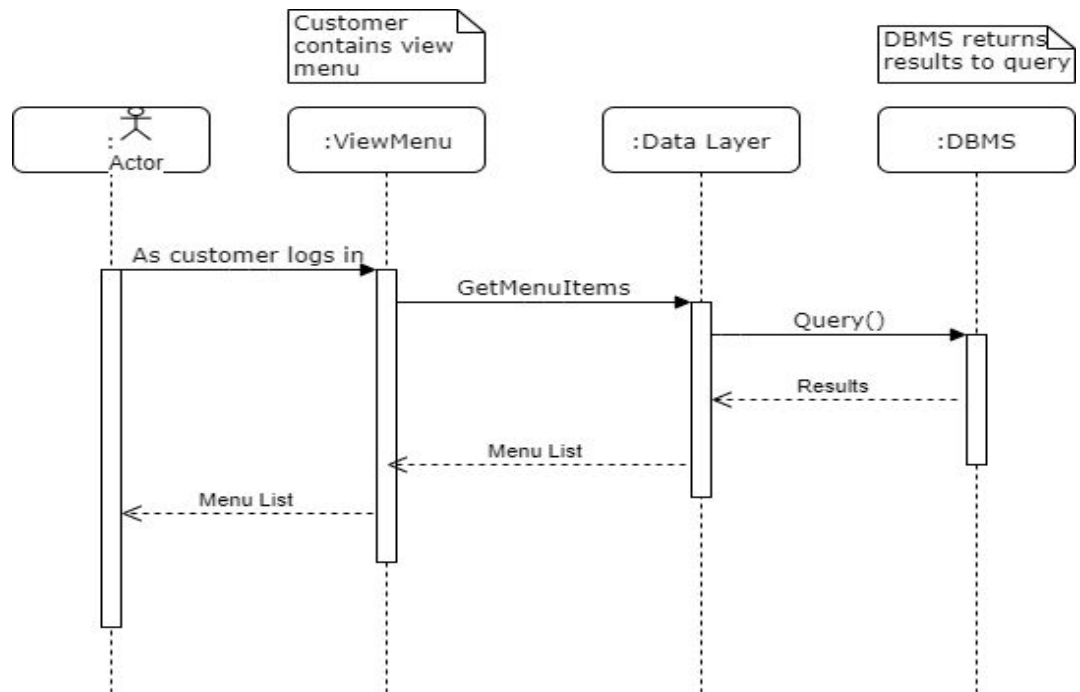
● **Place order**
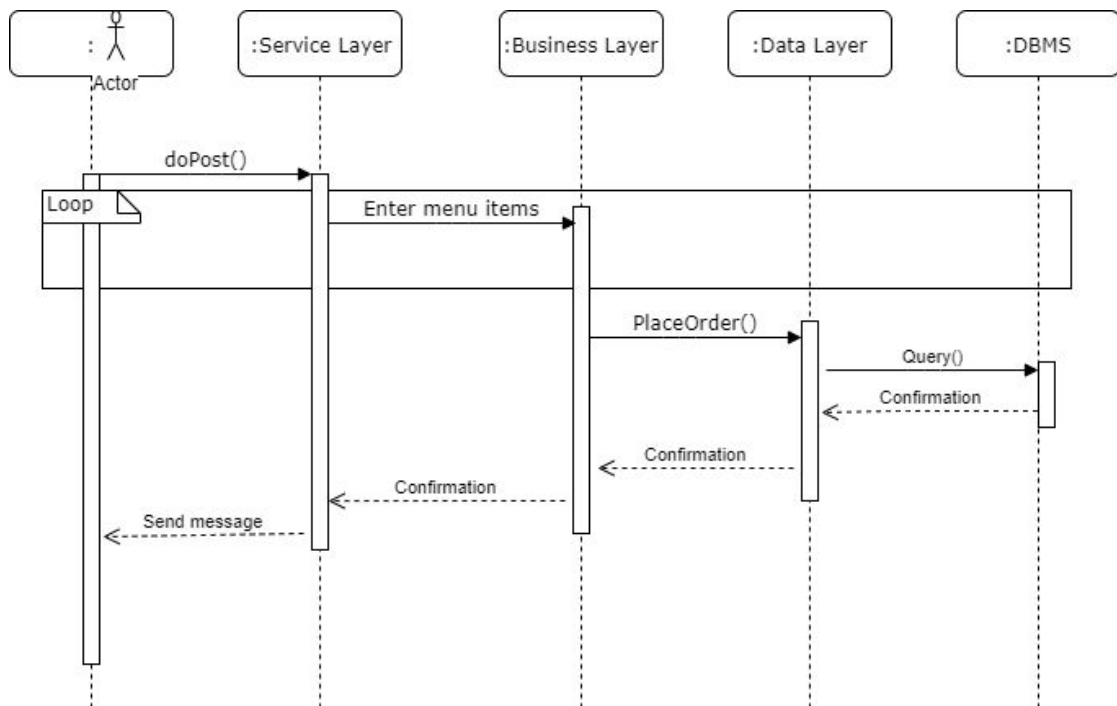
● **Update Order Status**
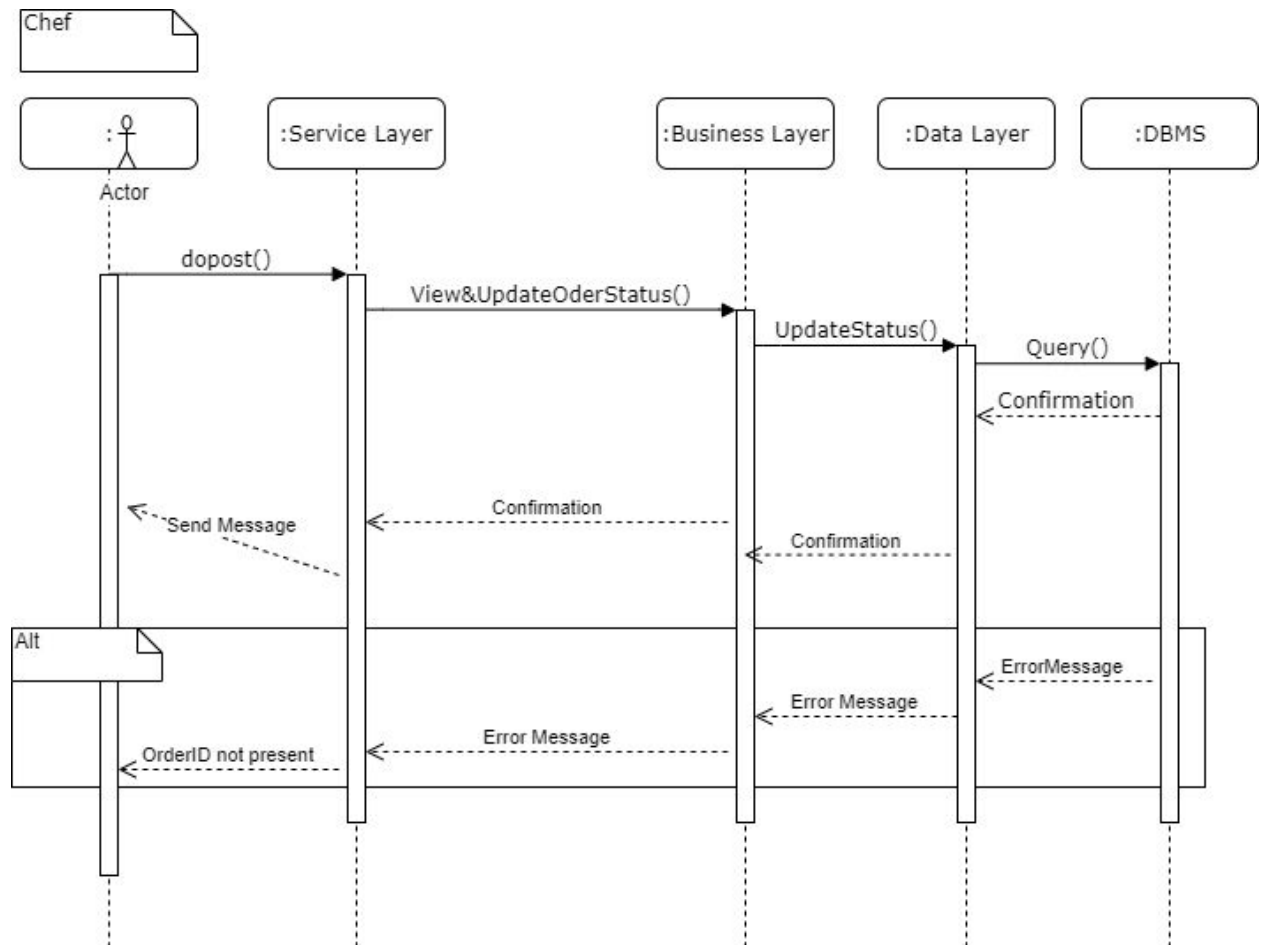
● **Edit Menu Items**
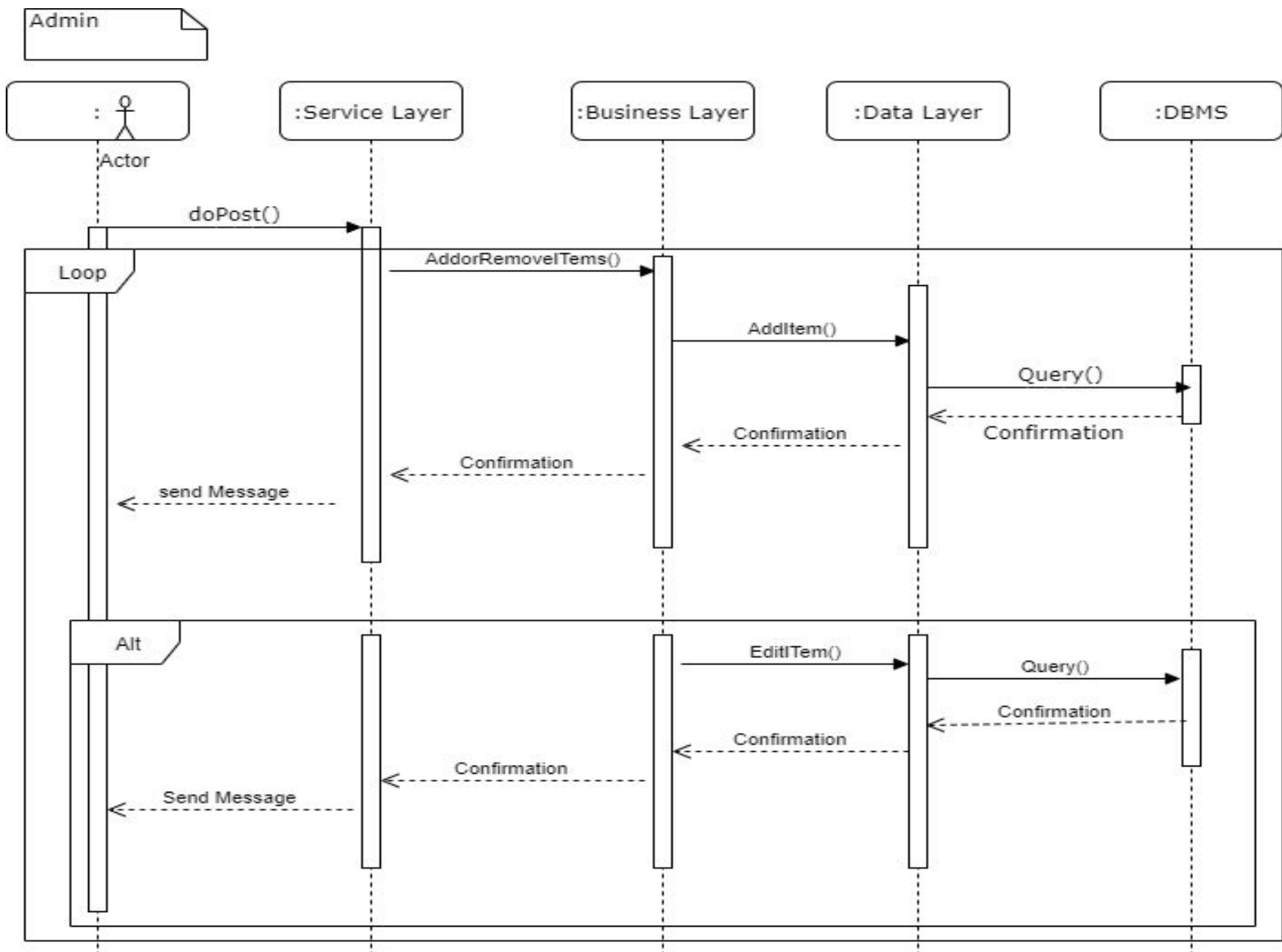
8. Use-Case Diagram
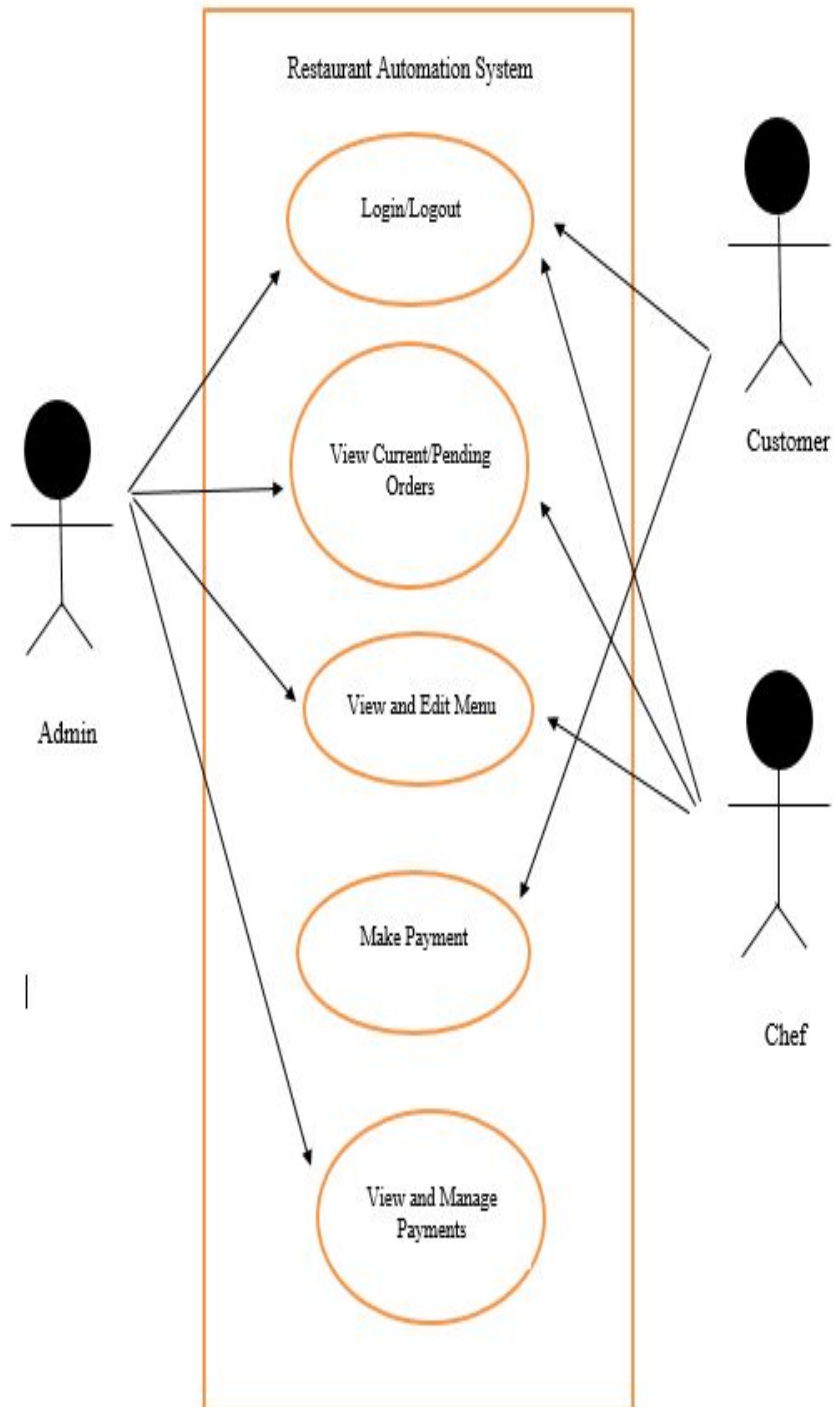
9. Class Diagrams

● **Design Class Diagram**