Detailed Use-case

Feature 1: Table Management

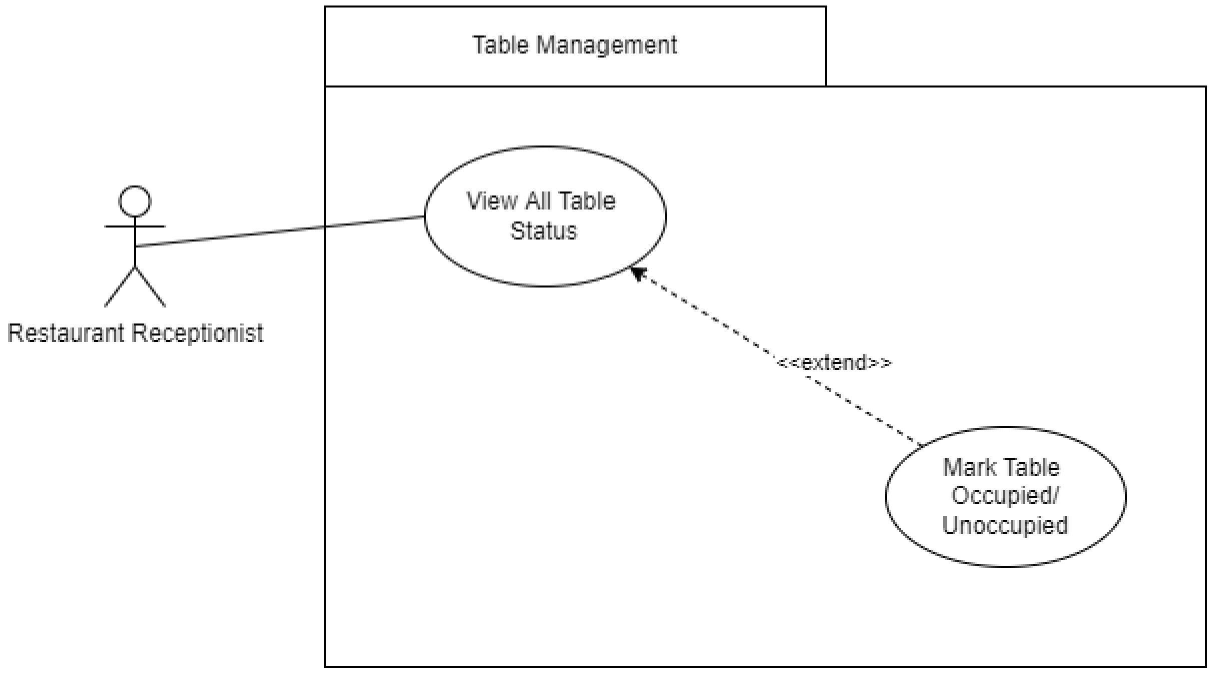


Table Management Use Case Diagram (Link image)

To further investigate Table Management Feature, we make Use Case Tables for that use case as demonstrated by Table UC-1.1, Table UC-1.2.

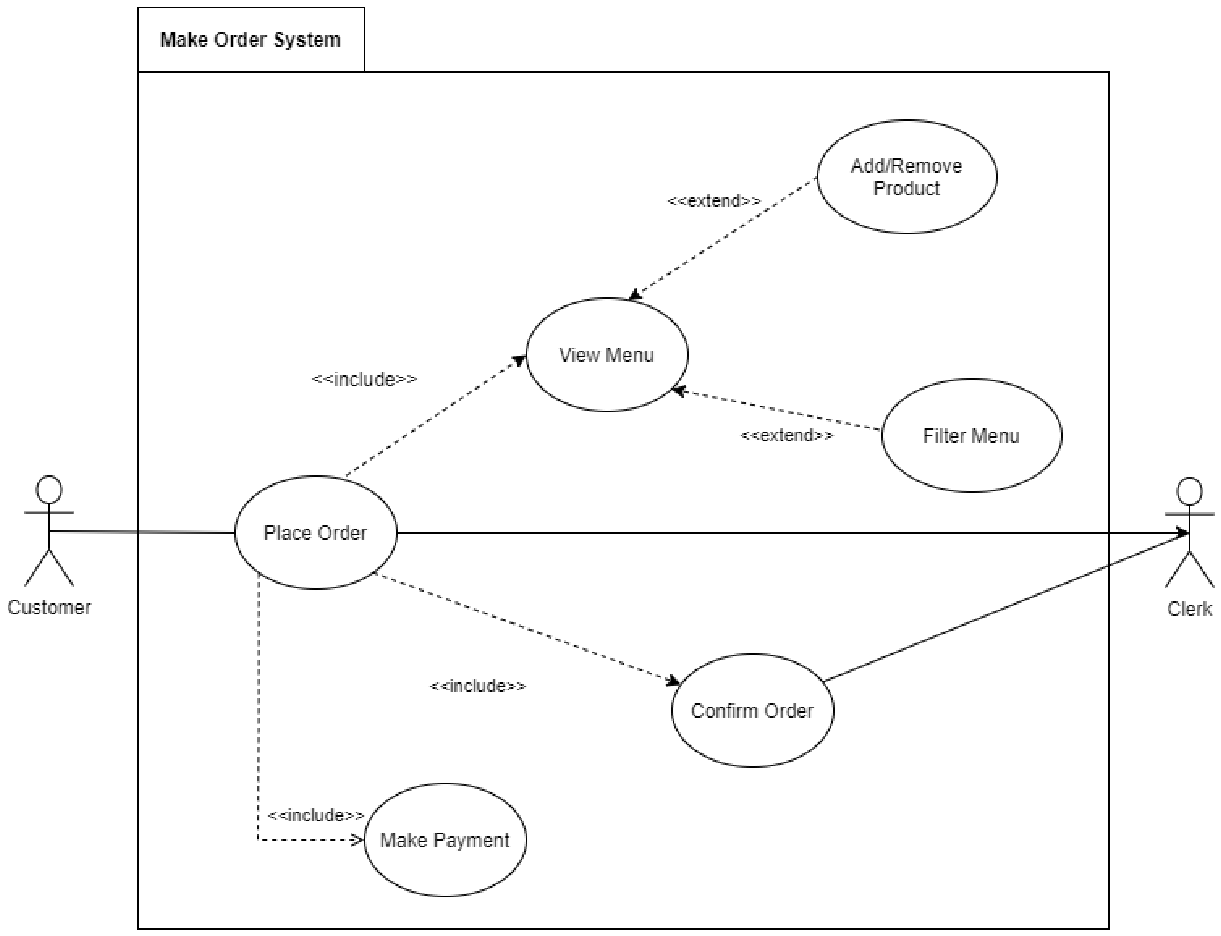
- Table UC-1.1: Table format for View All Table Status use case
- Table UC-1.2: Table format for Mark Table Occupied/ Unoccupied use case

Use Case ID	UC-1.1
Use Case Name	View All Table Status
Description	Receptionist can view all table status in the restaurant (occupied or unoccupied)
Actor(s)	Restaurant Receptionist
Priority	Medium
Trigger	Receptionist chooses to View All Table Status
Precondition(s)	None
Postcondition(s)	System displays all table status System saves changes to table status to the Database (if the receptionist decides to save changes)
Basic Flow	1. Receptionist chooses to View All Table Status 2. System get the status of all table status from Database 3. System displays the status of all table 4. Extend:: Mark Table Occupied/Unoccupied
Alternative Flow (Save Changes to Database)	5. Receptionist chooses to save 6. System updates the change to Database

As we can see from the Table UC-1.1, the View All Table Status use case does extend Mark Table Occupied/Unoccupied use case. Therefore, we provide use case table for Mark Table Occupied/Unoccupied use case (Table UC-1.2)

Use Case ID	UC-1.2
Use Case Name	Mark Table Occupied/Unoccupied
Description	Receptionist can mark a table occupied or unoccupied
Actor(s)	Restaurant Receptionist
Priority	Medium
Trigger	Receptionist chooses to update Table Status
Precondition(s)	Receptionist has viewed Table Status List
Postcondition(s)	System updates the table status (occupied or unoccupied) in the UI
Basic Flow (Update unoccupied table)	1. Receptionist selects a table need to be updated 2. System checks the selected table status as unoccupied 3. System updates the table status as occupied in Database
Alternative Flow (Update occupied table)	2a. System checks the selected table status as occupied 1. System updates the table as unoccupied in Database

Feature 2: Order System



Order System Use Case Diagram (Link image)

To further investigate Order System Feature, we make Use Case Tables for that use case as demonstrated by Table UC-2.1, Table UC-2.2, Table UC-2.3.

- Table UC-2.1: Table format for Place Order use case
- Table UC-2.2: Table format for View Menu use case
- Table UC-2.2: Table format for Confirm Order use case

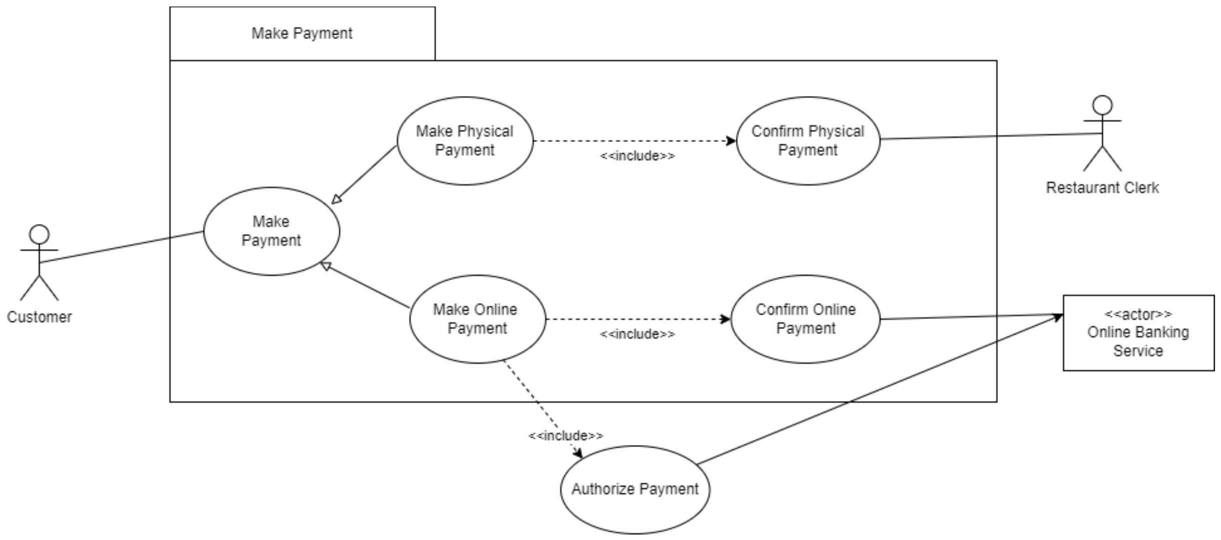
Use Case ID	UC-2.1
Use Case Name	Place Order
Description	Customers can order the food on the website of the restaurant
Actor(s)	Customer, Clerk
Priority	High
Trigger	Customers ask the System to place the order
Precondition(s)	Customers accessed the Order System through QR code or Restaurant Website
Postcondition(s)	The Order is processed by the System and is sent to Clerk to be confirm Customers move to Payment Process for the Order
Basic Flow	1. include::ViewMenu 2. include::ConfirmOrder

As we can see from the Table UC-2.1, the Place Order use case does include the View Menu use case with Confirm Order use case. Therefore, we provide two more use case tables for View Menu use case (Table UC-2.2) and Confirm Order use case (Table UC-2.3).

Use Case ID	UC-2.2
Use Case Name	View Menu
Description	Customer views the Restaurant Menu and add /remove Product to their order
Actor(s)	Customer
Priority	High
Trigger	As the customer accesses the Customer's Order Tab, system displays the Menu and option to add product to Customer's Order
Precondition(s)	Customers accessed the Menu Page through QR code
Postcondition(s)	Customers see the Restaurant Menu and are able to add / remove Product to their order, and submit the Order
Basic Flow	1. Customers accessed the Menu Page 2. System gets the list of products in the Menu 3. System displays the Menu to Customer 4. Customers add or remove the food to their Order 5. Customers submit the Order 6. System adds the Order Information with Pending status 7. Use Case continues at Make Payment Use Case Extend Point: 4a. Filter Menu: Customers choose a category in the Filter section and System filters all of the products in that category

Use Case ID	UC-2.3
Use Case Name	Confirm Order
Description	Clerk confirms the customer’s order
Actor(s)	Clerk
Priority	High
Trigger	Customer submits a Pending Order and the Kitchen has finished customer’s Order
Precondition(s)	Customer placed an order and sent the Order information to Clerk
Postcondition(s)	Clerk confirms the Pending Order System changes the Order information to Confirmed Status
Basic Flow	1. Clerk clicks on Pending Order tab 2. System displays the Pending Order(s) 3. Clerk can views the Order’s Detail 4. Clerk confirms the Order 5. System changes the Order status to Confirmed

Feature 3: Making Payment



Making Payment Use Case Diagram (Link image)

To further investigate Making Payment Feature, we make table format for that use case as demonstrated by Table UC-3.1, Table UC-3.2, Table UC-3.3, Table UC-3.4

- Table UC-3.1: Table format for Make Physical Payment use case
- Table UC-3.2: Table format for Confirm Physical Payment use case
- Table UC-3.3: Table format for Make Online Payment use case
- Table UC-3.4: Table format for Confirm Online Payment use case

Use Case ID	UC-3.1
Use Case Name	Make Physical Payment
Description	After submitting their order, customers choose the physical payment option for their order using the payment feature (by Cash or Credit Card)
Actor(s)	Customer, Restaurant Clerk
Priority	High
Trigger	Customer selects 'Make Payment' Option and select 'Physical Payment' option
Precondition(s)	Customers have submitted their order
Postcondition(s)	System displays their bills System send the payment information to the Clerk
Basic Flow	1. System gets the Customer's Orders's information 2. System displays 2 option for payment: Physical and Online Payment 3. Customer chooses Physical Payment 4. System records the Payment as Pending Payment 5. System displays the bill to customers 6. include::Confirm Physical Payment

As we can see from the Table UC-3.1, the Make Physical Payment use case does include Confirm Physical Payment use case. Therefore, we provide the use case table for Confirm Physical Payment use case (Table UC-3.2).

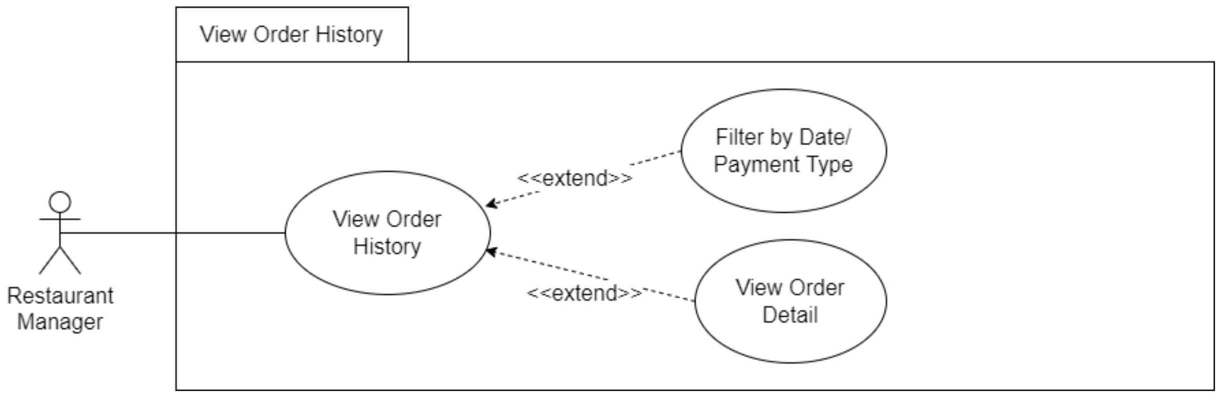
Use Case ID	UC-3.2
Use Case Name	Confirm Physical Payment
Description	After the system sends a payment request to notify the Restaurant Clerk that a customer has made a payment request, when the customers have finished their meal, the Clerk receives physical payment from the customers and confirms the payment.
Actor(s)	Restaurant Clerk
Priority	High
Trigger	Clerk receives a Payment Request from customer and Customers request to process the physical payment
Precondition(s)	There are pending physical payment requests.
Postcondition(s)	Customers pay off their bills by physical method. Clerk confirms the Payment
Basic Flow	1. Clerk views the Payment Information and requests physical payment from Customer (by cash or by credit card) and go to the table 2. Customers pay for the Order 3. Clerk confirms the Payment 4. System changes payment status in Database from Pending to Confirmed

Use Case ID	UC-3.3
Use Case Name	Make Online Payment
Description	After submitting their order, customers choose the online payment option for their order using the payment feature (by online payment service)
Actor(s)	Customer, Online Payment Service
Priority	High
Trigger	Customer selects 'Make Payment' Option and select 'Online Payment' option
Precondition(s)	Customers have submitted their order
Postcondition(s)	Online Payment Service is notified by the payment request
Basic Flow	1. System gets the Orders's information 2. System displays 2 option for payment: Physical and Online Payment 3. Customer chooses Online Payment 4. System sends Customer to the Online Payment Service to make payment 5. System sends Payment Information to Online Payment Service 6. Include::Confirm Online Payment

As we can see from the Table UC-3.3, the Make Physical Payment use case does include Confirm Online Payment use case. Therefore, we provide the use case table for Confirm Online Payment use case (Table UC-3.4).

Use Case ID	UC-3.4
Use Case Name	Confirm Online Payment
Description	Online Payment Service sends a Payment Validation to confirm the Online Payment
Actor(s)	Online Payment Service
Priority	High
Trigger	System receives Payment Validation from Online Payment Service
Precondition(s)	A customer has made an online payment request for the Order
Postcondition(s)	Customer finishes the Online Payment Process System records the Online Payment
Basic Flow (Online Payment)	1. Online Payment Service sends Payment Validation as valid 2. System displays the validation to the Customer 3. System records the Payment in Database as Confirmed 4. System display the bill to the Customer
Exception Flow (Online Payment is not successful)	2a. The Validation that System receives from Online Payment Service is not valid ● System return Customer back to Select Payment Step

Feature 4: View Order History



View Order History Use Case Diagram (Link image)

To further investigate View Order History Feature, we make table formats for that use case as demonstrated by Table UC-4.1, Table UC-4.2

- Table UC-4.1: Table format for View Order History use case
- Table UC-4.2: Table format for View Order Detail use case

Use Case ID	UC-4.1
Use Case Name	View Order History
Description	Restaurant Manager can view the restaurant order and payment record in Database, can filter the data by date or by payment type
Actor(s)	Restaurant Manager
Priority	Low
Trigger	None
Precondition(s)	The manager accesses to the manager's view to see the restaurant's Order History
Postcondition(s)	System displays the Order Records or the Payment Records
Basic Flow	1. Restaurant Manager gets access to the manager's view 2. System gets list of Orders and Payments from Database 3. System displays the list of Order and Payment 4. Extend:: View Order Detail Extend:: Filter by Date / Payment Type Extend:: Filter by Date / Payment Type ● The manager chooses the date or type to filter. ● The system handles the filter request. ● Use case continues at step 3.

As we can see from the Table UC-4.1, the View Order History use case does extend the View Order Detail use case. Therefore, we provide the use case table for View Order Detail use case (Table UC-4.2).

Use Case ID	UC-4.2
Use Case Name	View Order Detail
Description	Restaurant Manager can view the restaurant Order's Detail of a specific order
Actor(s)	Restaurant Manager
Priority	Low
Trigger	Restaurant Manager clicks one of the orders displayed in Manager View
Precondition(s)	The manager access to the manager's view to see the restaurant Orders
Postcondition(s)	System displays the Order's Detail, including the products in the Order
Basic Flow	1. Restaurant Manager clicks one of the orders displayed 2. System gets Order Detail with the Order ID in database 3. System displays the Order Detail