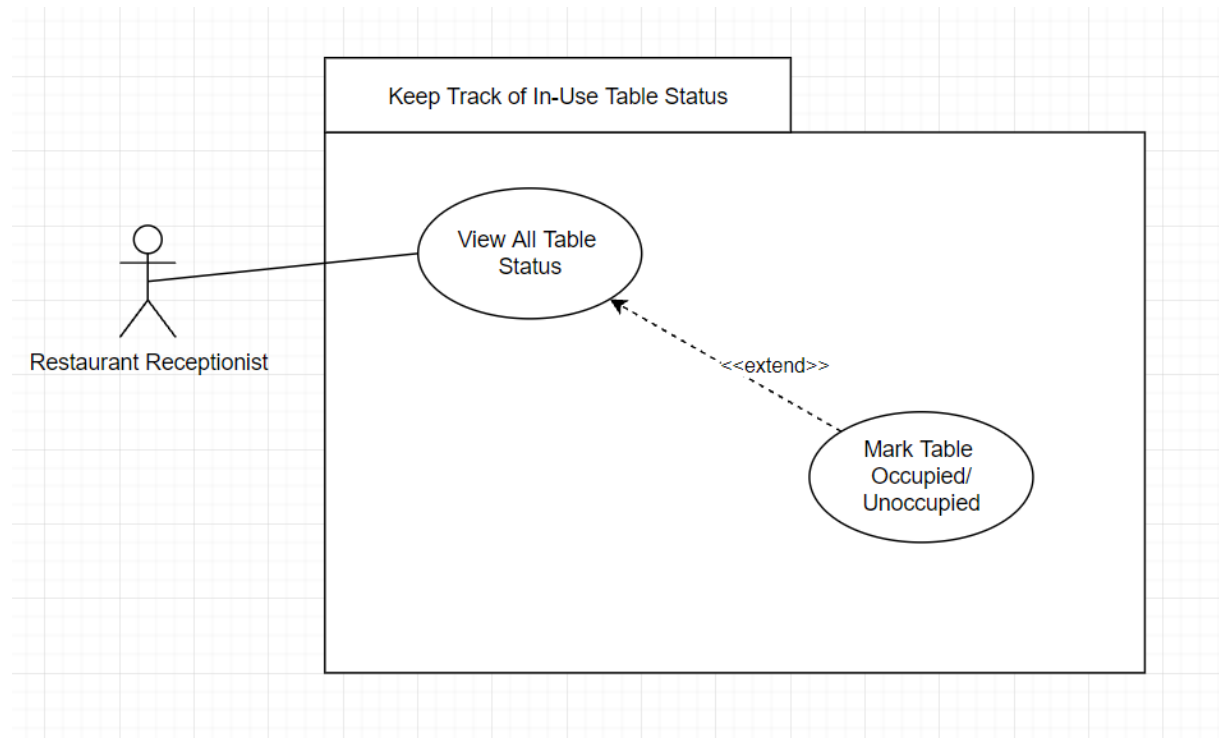


Detailed Use-case

Feature 1: Table Management



[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.

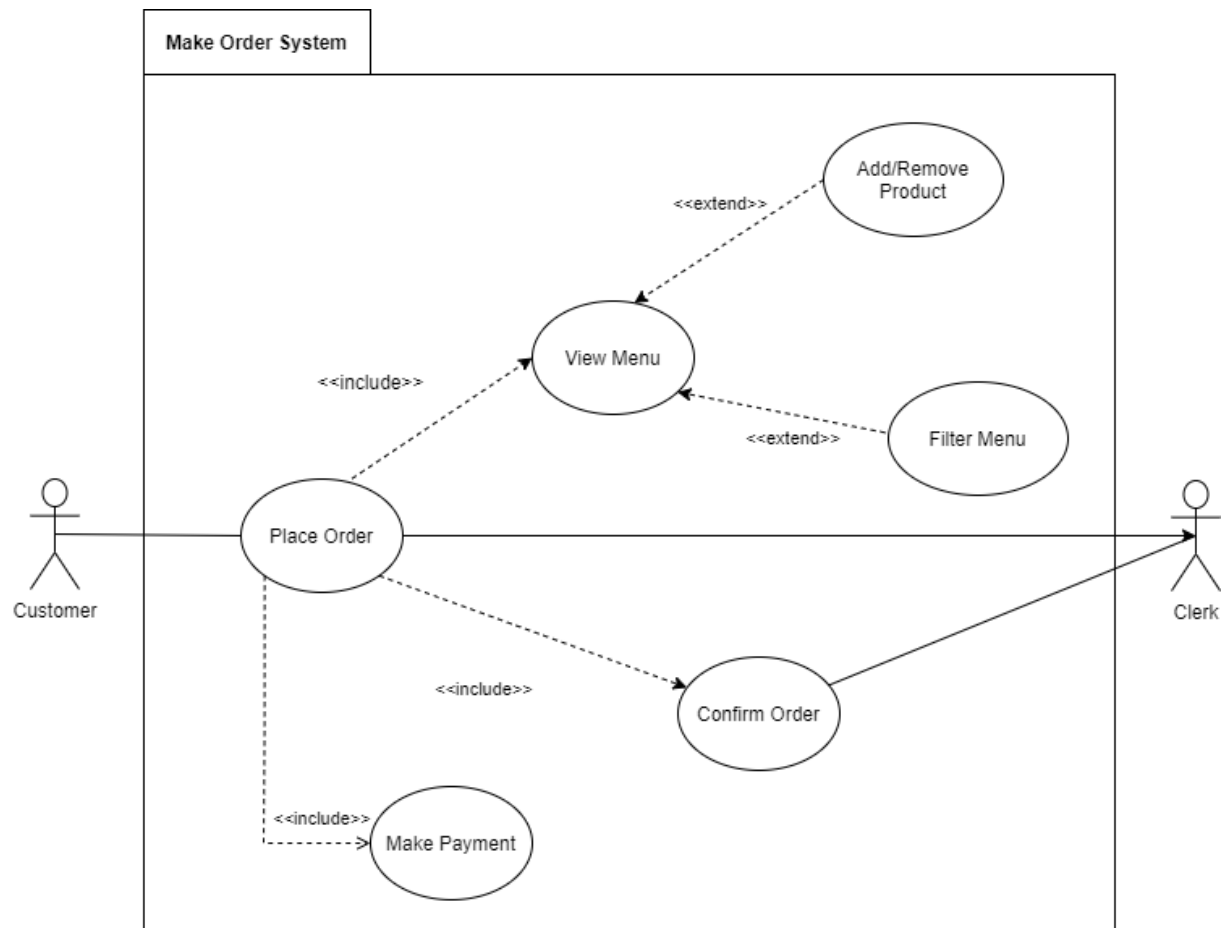
- Table UC-1.1: Table format for View All Table Status use case
- Table UC-1.2: Table format for Mark Table Occupied/ Unoccupied use case

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
Basic Flow	<ol style="list-style-type: none">1. Receptionist chooses to View All Table Status2. System get the status of all table status from Database3. System displays the status of all table4. Extend:: Mark Table Occupied/Unoccupied5. Receptionist chooses to save6. 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) successfully
Basic Flow (Update unoccupied table)	<ol style="list-style-type: none">1. Receptionist selects a table need to be updated2. System checks the selected table status as unoccupied3. System updates the table status as occupied in Database
Alternative Flow (Update occupied table)	<ol style="list-style-type: none">2a. System checks the selected table status as occupied<ol style="list-style-type: none">1. System updates the table as unoccupied in Database

Feature 2: Order System



[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

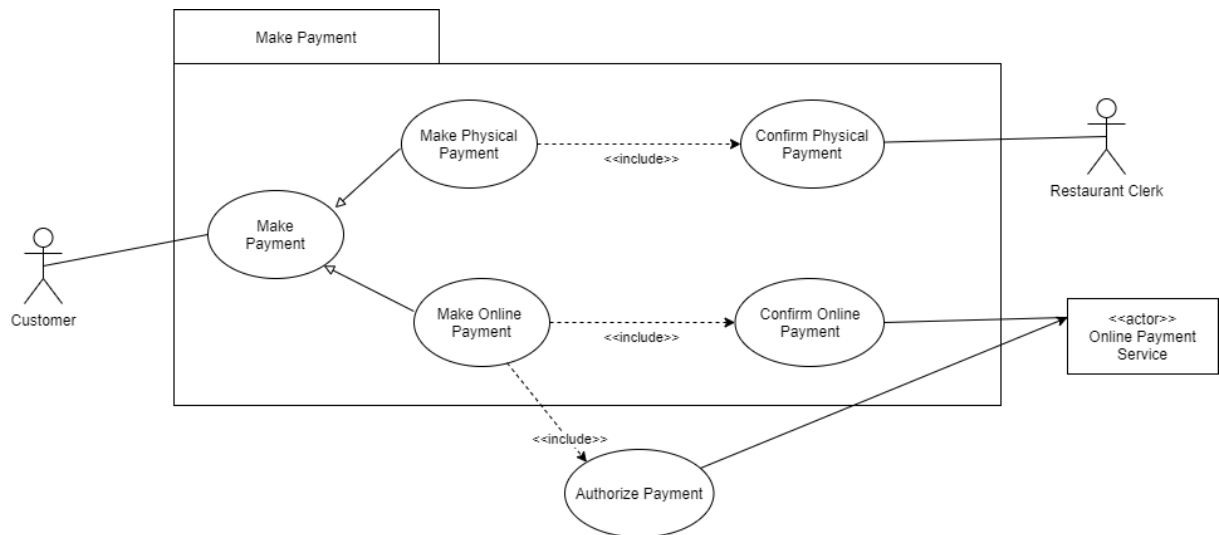
Use Case ID	UC-2.1
Use Case Name	Place Order
Description	Customers can order the food on the website of the restaurant
Actor(s)	Customer, Clerk
Priority	High
Trigger	The Customers ask the System to place the order
Precondition(s)	Customers accessed the Order System through QR code
Postcondition(s)	The Order is processed by the System and is sent to Clerk to be confirm Customers move to Payment Process for the Order
Basic Flow	<ol style="list-style-type: none">1. include::ViewMenu2. include::ConfirmOrder3. Use Case continues at Make Payment Use Case

As we can see from the Table UC-2.1, the Place Order use case does include 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)	Customer accessed the Order System through QR code
Postcondition(s)	Customer can see the Restaurant Menu and are able to add / remove Product to their order
Basic Flow	<ol style="list-style-type: none"> 1. Customers click on the "View Menu" button 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 <p>Extend Point:</p> <p>4a. Filter Menu: Customers choose a category in the Filter section and System filters all of the products in that category</p>

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
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	<ol style="list-style-type: none">1. Clerk clicks on Pending Order tab2. System display the Pending Order3. Clerk can views the Order's Information4. Clerk confirms the Order5. System changes the Order status to Confirmed

Feature 3: Making Payment



[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 finishing their meal, customer makes Physical payment 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)	Customer's order has not been paid
Postcondition(s)	Restaurant Clerk is notified by the payment request
Basic Flow	<ol style="list-style-type: none">1. System gets the Customer's Orders's information2. System calculates and display the bill3. System sends Payment Information to Clerk4. Use Case continues at Confirm Physical Payment Use Case

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).

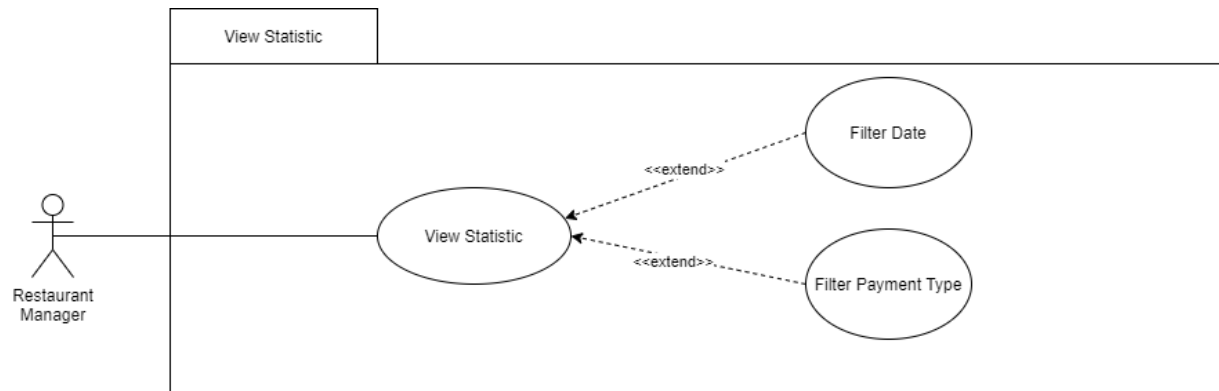
Use Case ID	UC-3.2
Use Case Name	Confirm Physical Payment
Description	After system sends a payment request to notify the Restaurant Clerk that a customer has made payment, the Clerk receives payment from customer and confirms the payment is successful
Actor(s)	Restaurant Clerk
Priority	High
Trigger	Clerk receives a Payment Request from customer
Precondition(s)	There are some physical payment requests.
Postcondition(s)	Customers pay off their bills by physical method. System records the transaction
Basic Flow	<ol style="list-style-type: none">1. Clerk views the Payment Information and requests transaction from Customer (by cash or by credit card)2. Clerk confirms the Payment3. System records the transaction in Database

Use Case ID	UC-3.3
Use Case Name	Make Online Payment
Description	After finishing their meal, customer makes payment 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)	Customer has Orders that haven't not been paid
Postcondition(s)	Online Payment Service is notified by the payment request
Basic Flow	<ol style="list-style-type: none"> 1. System gets the Orders's information 2. System calculates and display the bill 3. System sends Customer to the Third-party website to make payment 4. System sends Payment Information to Online Payment Service 5. Use Case continues at Confirm Online Payment Use Case

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 for the Order
Postcondition(s)	Customer finishes the Payment Process System records the Payment
Basic Flow (Online Payment)	<ol style="list-style-type: none">1. Online Payment Service sends Payment Validation as valid2. System displays the validation to the Customer3. System records the Payment in Database
Exception Flow (Online Payment is not successful)	<ol style="list-style-type: none">2a. The Validation that System receives from Online Payment Service is not valid3a. System return Customer back to Select Payment Step

Feature 4: View Statistics



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

- Table UC-4.1: Table format for View Statistics use case

Use Case ID	UC-4.1
Use Case Name	View Statistics
Description	Restaurant Manager can view the restaurant order and payment record in Database
Actor(s)	Restaurant Manager
Priority	Low
Trigger	None
Precondition(s)	The manager access to the manager's view to see the restaurant statistics
Postcondition(s)	System displays the Order Record or the Payment Records
Basic Flow	<ol style="list-style-type: none">1. Restaurant Manager gets access to the manager's view2. System gets Order and Payment Statistics from Database3. System displays the list of Statistics.
Alternative Flow (Filter statistics according to date or type)	<ol style="list-style-type: none">4. The manager chooses the date or type to filter.5. The system handles the request. Use case continues at step 3.