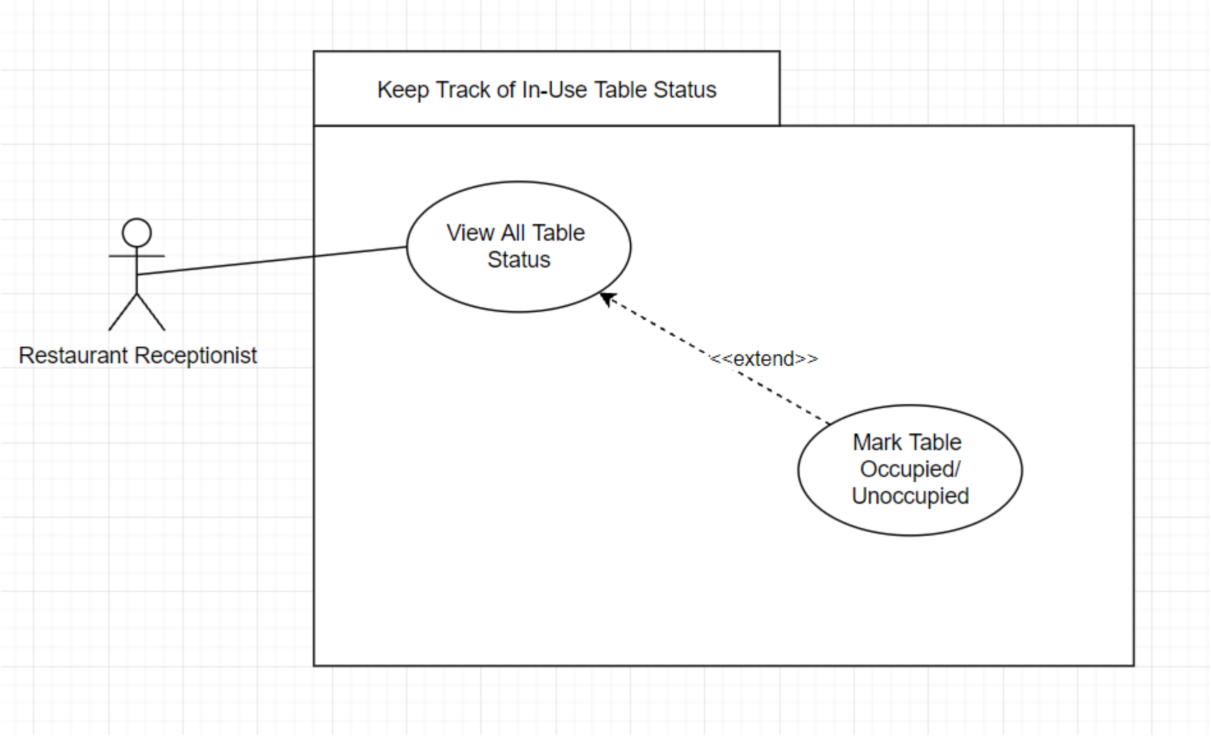
Work Assignment

Member	Feature
Trương Hoàng Phúc Nguyễn Kế Đạt	The Project’s Database
Nguyễn Phúc Thịnh	Table Management Feature
Đặng Quốc Thanh	View Statistics Feature
Huỳnh Đức Thịnh Nguyễn Diệu Ái	Order System Feature
Nguyễn Ngô Thanh Trúc	Making Payment Feature

Detailed Use-case

Feature 1: Table Management


Keep Track of In-Use Table Status
View All Table Status
Restaurant Receptionist
<<extend>>
Mark Table Occupied/ Unoccupied

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.

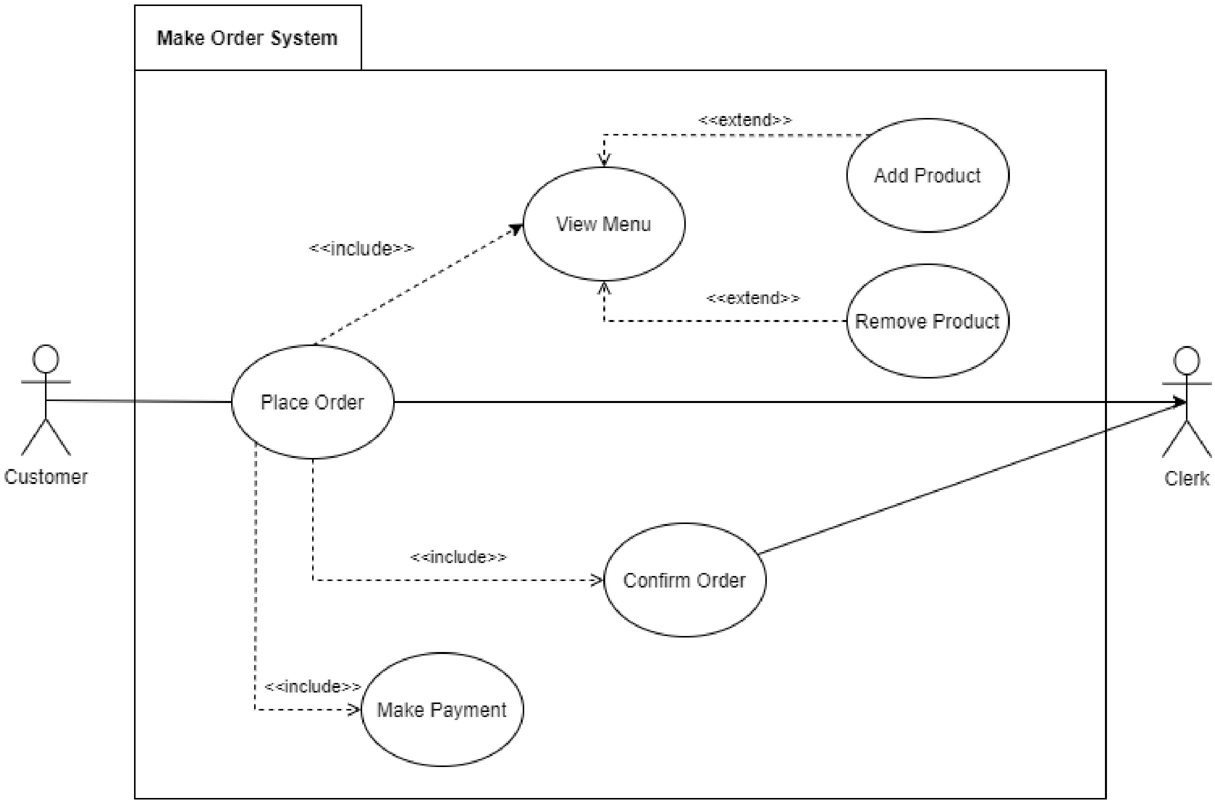
- Table UC-1.1: Table format for View All Table Status use case
- Table UC-1.2: Table format for Mark Table Occupied/ Unoccupied use case

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	High
Trigger	Receptionist chooses to View All Table Status
Precondition(s)	None
Postcondition(s)	System displays all table status
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 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	High
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)	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



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

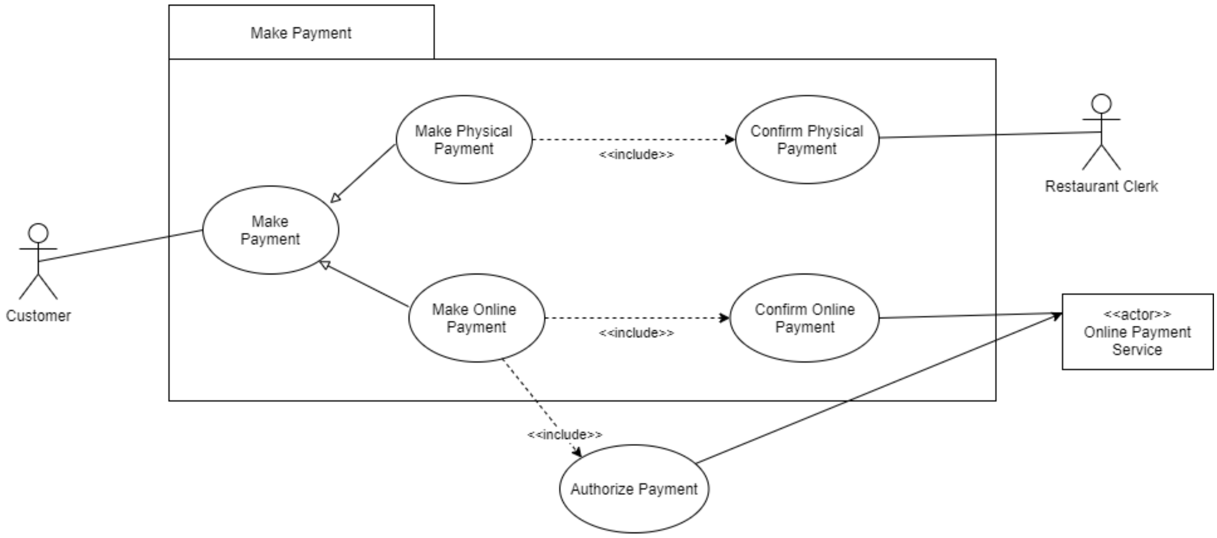
Use Case ID	UC-2.1
Use Case Name	Place Order
Description	Customers can order the food on the website of the restaurant
Actor(s)	Customer
Priority	High
Trigger	The Customer ask the System to place the order
Precondition(s)	Customer accessed the Order System through QR code
Postcondition(s)	The Order is processed by the System and is sent to Clerk to be Confirm
Basic Flow	1. include::ViewMenu 2. include::ConfirmOrder 3. 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	1. Customer clicks on the "View Menu" button 2. System gets the list of products in the Menu 3. System displays the Menu to Customer 4. Customer adds or remove the food to their Order 5. Customer submits the Order 6. System adds the Order Information with Pending status

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	1. System display the Pending Order 2. Clerk views the Order’s Information 3. Clerk confirms the Order 4. System changes the Order status to Confirmed 5. System notify customer about the Confirmation

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 has Orders that haven't not been paid
Postcondition(s)	Restaurant Clerk is notified by the payment request
Basic Flow	1. System gets the Customer's Orders's information 2. System calculates and display the bill 3. System sends Payment Information to Clerk 4. 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).

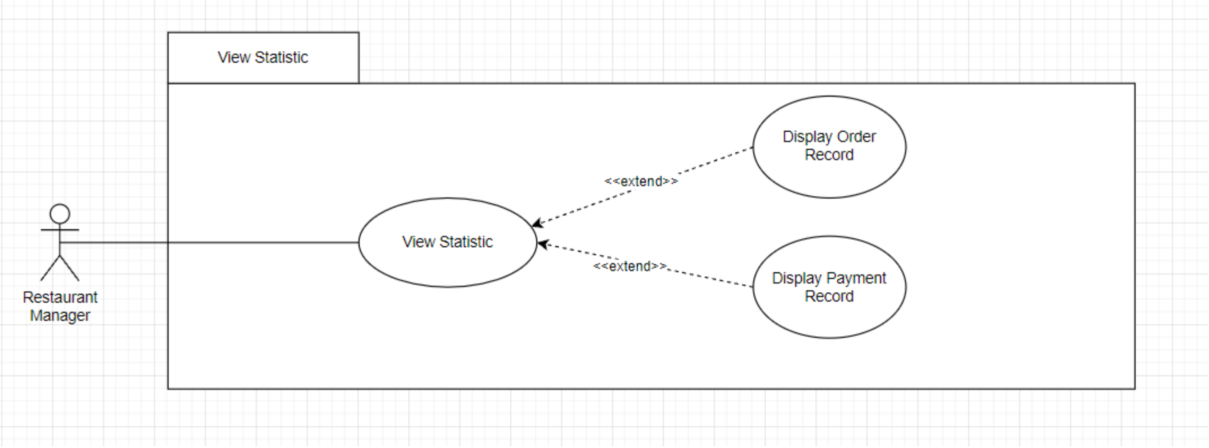
Use Case ID	UC-3.2
Use Case Name	Confirm Physical Payment
Description	After system sends a payment request to notify the Restaurant Clerk that customer has made payment, Receptionist confirms the payment is successful
Actor(s)	Restaurant Clerk
Priority	High
Trigger	Clerk receives a Payment Request from customer
Precondition(s)	Customer has a payment in request of confirming (by Physical method or by third party service)
Postcondition(s)	Customer finishes the physical transaction System records the transaction
Basic Flow	1. Clerk views the Payment Information and requests transaction from Customer (by cash or by credit card) 2. Clerk confirms the Payment 3. 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	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 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)	Customer has made an online payment for the Order
Postcondition(s)	Customer finishes the Payment Process System records the 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
Exception Flow (Online Payment is not successful)	2a. The Validation that System receives from Online Payment Service is not valid 1. System return Customer back to Select Payment Step

Feature 4: View Statistics



Link image

To further investigate View Statistics Feature, we make table formats for that use case as demonstrated by Table UC-4.1, Table UC-4.2, Table UC-4.3

- Table UC-4.1: Table format for View Statistics use case
- Table UC-4.2: Table format for Display Order Record use case
- Table UC-4.3: Table format for Display Payment Record 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	Restaurant Manager clicks “View Statistic” option
Precondition(s)	None
Postcondition(s)	System displays the Order Record or the Payment Records
Basic Flow	1. Restaurant Manager clicks “View Statistic” option 2. System displays two options: “View Order” and “View Payment” 3. Extend :: Display Order Record, Display Payment Record

As we can see from the Table UC-4.1, the View Statistic use case does extend the Display Order Record and Display Payment Record use case. Therefore, we provide two use case tables for Display Order Record use case (Table UC-4.2) and Display Payment Record use case (Table UC-4.3)

Use Case ID	UC-4.2
Use Case Name	Display Order Record
Description	Restaurant Manager view All Order Record in the Database in a specific date or all of them
Actor(s)	Restaurant Manager
Priority	Low
Trigger	Restaurant Manager clicks “View Order” option
Precondition(s)	Restaurant Manager has accessed the “View Statistics” Tab
Postcondition(s)	System displays the Order Records in a specific date or all of them
Basic Flow	1. Restaurant Manager clicks “View Order” option 2. System displays dialog asks the Date 3. Restaurant Manager enters the Date want to view Order Record and clicks “View” 4. System gets all of the orders made in that date from Database 5. System displays the orders
Alternative Flow	3a. Restaurant leaves the Date field blank and clicks “View” 1. System gets all of the order made in that date from Database 2. System displays the orders

Use Case ID	UC-4.3
Use Case Name	Display Payment Record
Description	Restaurant Manager view All Payment Record in the Database in a specific date or all of them
Actor(s)	Restaurant Manager
Priority	Low
Trigger	Restaurant Manager clicks "View Payment" option
Precondition(s)	Restaurant Manager has accessed the "View Statistics" Tab
Postcondition(s)	System displays the Payment Records in a specific date or all of them
Basic Flow	1. Restaurant Manager clicks "View Payment" option 2. System displays dialog asks the Date 3. Restaurant Manager enters the Date want to view Payment Record and clicks "View" 4. System gets all of the payments made in that date from Database 5. System displays the payments
Alternative Flow	3a. Restaurant Manager leaves the Date field blank and clicks "View" 3. System gets all of the payments made in that date from Database 4. System displays the payments