HO CHI MINH UNIVERSITY OF TECHNOLOGY

FACULTY OF COMPUTER SCIENCE AND ENGINEERING

SOFTWARE ENGINEERING ASSIGNMENT

RESTAURANT POS

Instructor: Lê Đình Thuận

Group: SuperEngineers

Member List

Nguyễn Kế Đạt - 1913048

Nguyễn Diệu Ái - 1910032

Huỳnh Đức Thịnh - 1910563

Đặng Quốc Thanh - 1915083

Nguyễn Phúc Thịnh - 1910565

Trương Hoàng Phúc - 1914720

Nguyễn Ngô Thanh Trúc - 1910650

MỤC LỤC

Introduction

Point of sale (POS) or point of purchase is the time and place where a retail transaction is completed. At the point of sale, the merchant calculates the amount owed by the customer, indicates that amount, may prepare an invoice for the customer, and indicates the options for the customer to make payment. In restaurant business, POS systems often include table reservation, ordering food, alerts, billing, credit card processing and customer management.

The new POS system is requested to be developed based on a web-based system and shall implement the current business flow as described below.

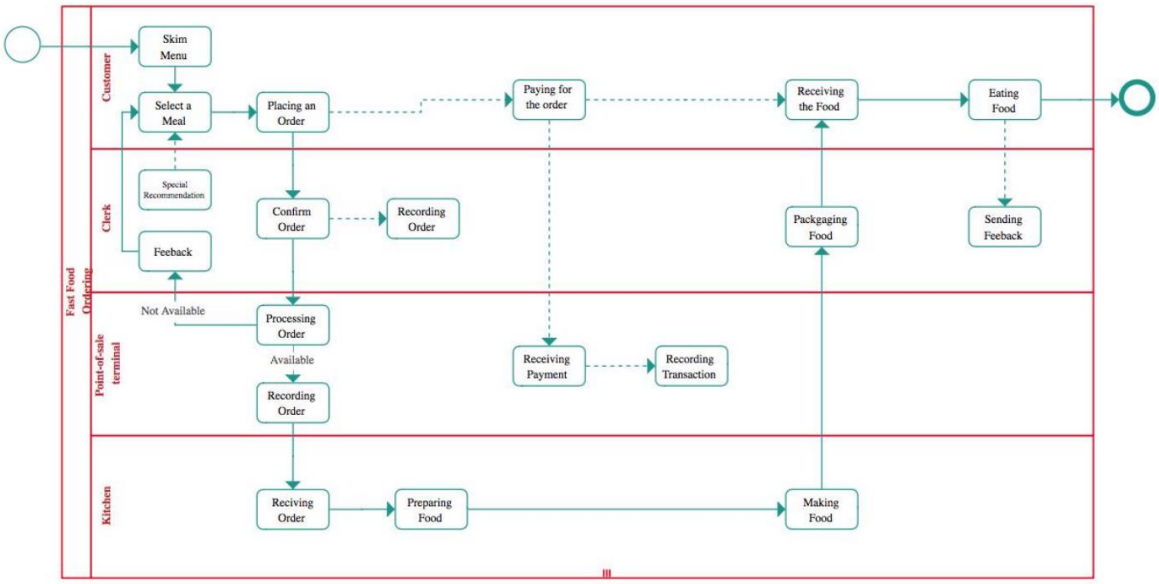


Figure 1: Customer-drawing workflow

Stakeholders

Stakeholder	Role	Description
Restaurant Owner	Product Owner	Who owns the application
Restaurant Manager	Restaurant Staff	Who is able to see the statistic on Order and Payment recorded in the Database
Restaurant Clerk	Restaurant Staff	Who is responsible for handling customer's issues and payment.
Kitchen Staff	Restaurant Staff	Who is responsible for complete the customer's order
Restaurant Receptionist	Restaurant Staff	Who is responsible for keeping track of table status
Customer	Restaurant Customer	Who will use the application to make reservations, order food and make payment.

Project scope

Project Justification

The primary goal of this Restaurant POS project is to provide a web-based application that automates many restaurant's processes in order to increase business intelligence, reduce wasted manpower and opportunity to scale to a large business.

User Story

As a customer of the restaurant:

· I can browse the restaurant menu and look at the various food options available in the restaurant along with the price for each item.

· I am able to select dishes from the menu and add wanted dishes to my order

· I can submit my order to inform the restaurant about my request

· I can make payment for my order, either by card, e-wallet or by cash

As a receptionist of the restaurant

· I can view all of the table status, if they are currently in-use or not

· I can change a table's status, from available to occupied or vice versa

As a clerk of the restaurant:

· I can view the customer's submitted orders

· I can confirm or remove customer's submitted order

· I can confirm customer's payment, either by card, e-wallet or by cash

As the manager of the restaurant:

· I can view the restaurant's order history

· I can view the restaurant's transaction (payment) history

General Feature of the Project

Feature 1: Table Management

Allowing restaurant receptionists to keep track of which tables are occupied currently and change their status from occupied to unoccupied when customers have finished their meal and vice versa when new customers arrived

Feature 2: Order System

Offering customers an interactive menu and indirect way of ordering food. Restaurant clerk can view and confirm the order. Kitchen staff can view and complete the order.

Feature 3: Making payment

Allowing customers to see their bills and send a check-out alert to the clerk. They can choose to pay by cash or online payment.

Feature 4: View Statistics

Daily transactions are recorded and this feature allows the restaurant manager to view the transaction statistics.

Assumptions

- Both the restaurant and the customers have access to the Internet when using the application
- Payment transactions are handled by the third-party service

Project Context

Business Model: Fine dining restaurant

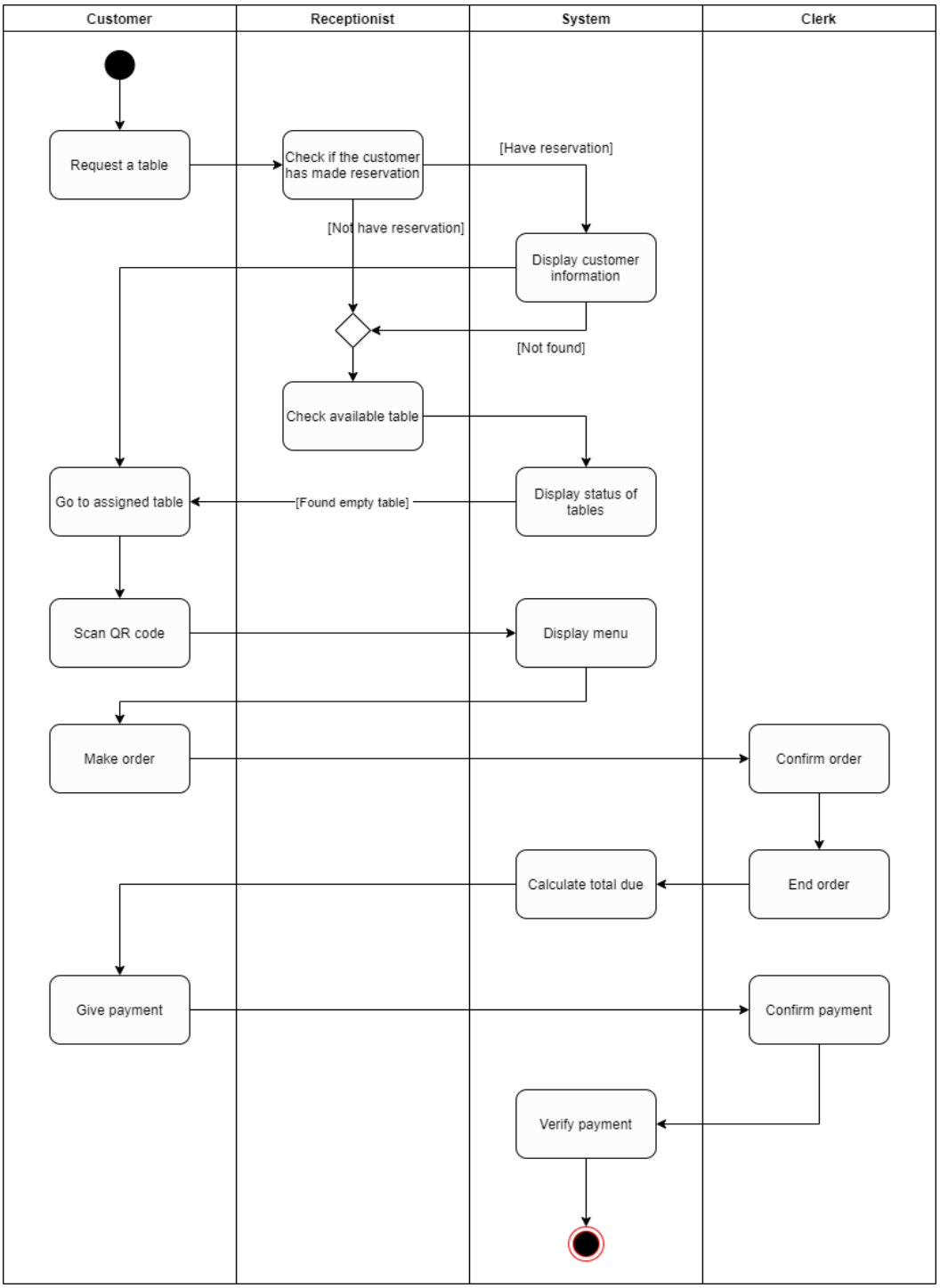
Restaurant Customer: Upper Middle Class Customer

Payment: Support cash payment and online payment (credit card, e-wallet)

Dining service: Eat-in restaurant

Tale service: Support customers making online reservations.

Business Flow:



[Link image]

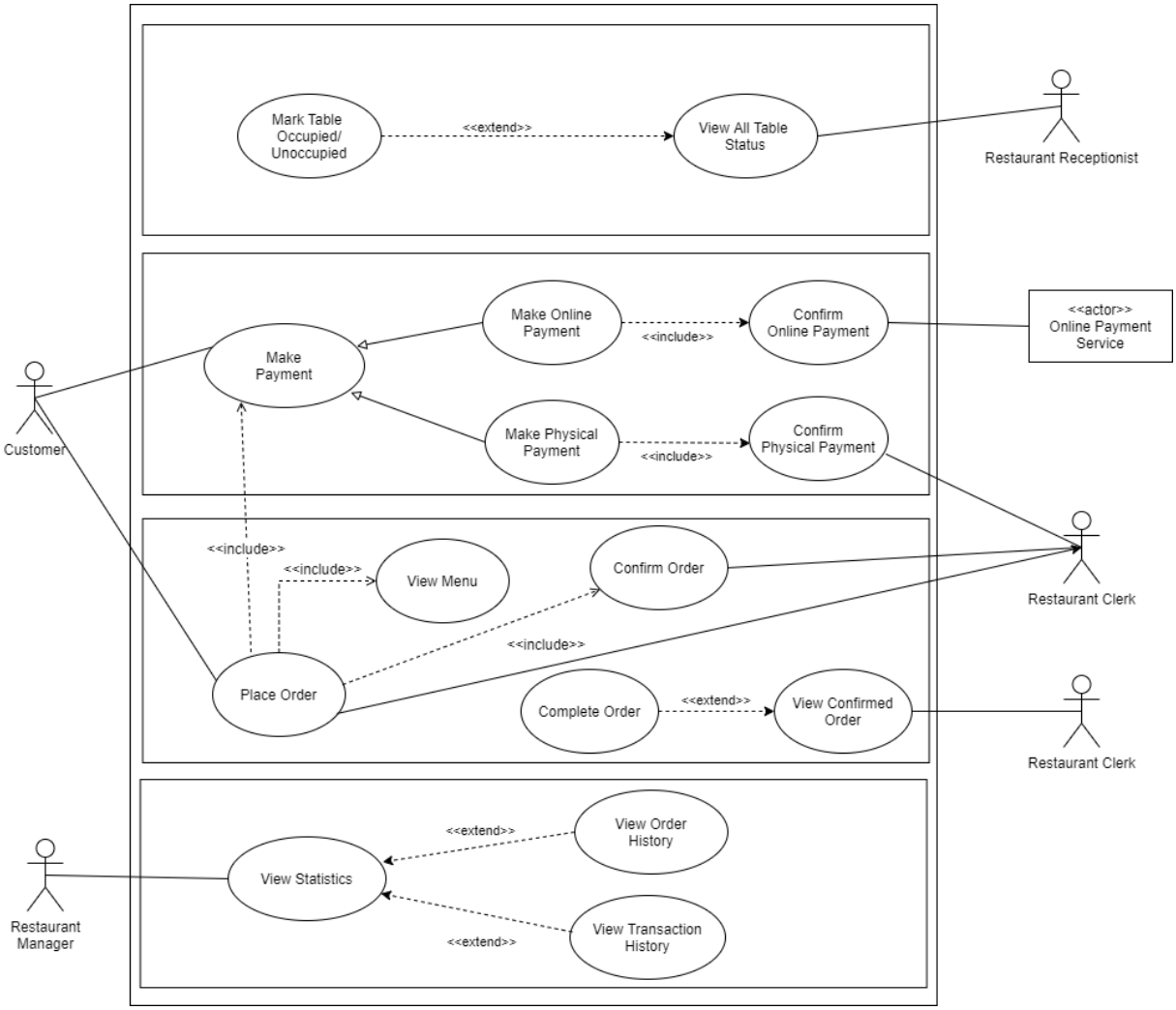
Functional Requirements

Requirement ID	Requirement	Priority	Comments
FR001	Customer can make reservations and Restaurant Receptionist can confirm the reservation	High	
FR002	Restaurant Receptionist can cancel Customer's Reservation according to Customer's Wishes	Medium	
FR003	Restaurant Receptionist can check if there are enough tables for non-reservation customer	High	
FR004	Restaurant Receptionist can view and update table status	High	
FR005	Customer can make food orders and Clerk can cancel the food order according to customers before confirming and sending to the kitchen	High	
FR006	Customer can make payment for their meal either by direct payment or third-party service	High	
FR007	Restaurant Manager can view all the Order and Transaction recorded by the System	Low	
FR008	Restaurant Manager can add, remove, update menu items in the Menu	Medium	

Non-functional Requirements

Requirement ID	Requirement	Comments
NFR001	The system is implemented using Web technology and QR code	
NFR002	The system should be used on mobile, table devices and computer/laptop	
NFR003	The system should be extendable for multiple restaurants in the future	
NFR004	The system can handle at least 300 orders per day	
NFR005	The system should allow non-direct contact between Clerks and Customers.	

Project's Use-Case Diagram



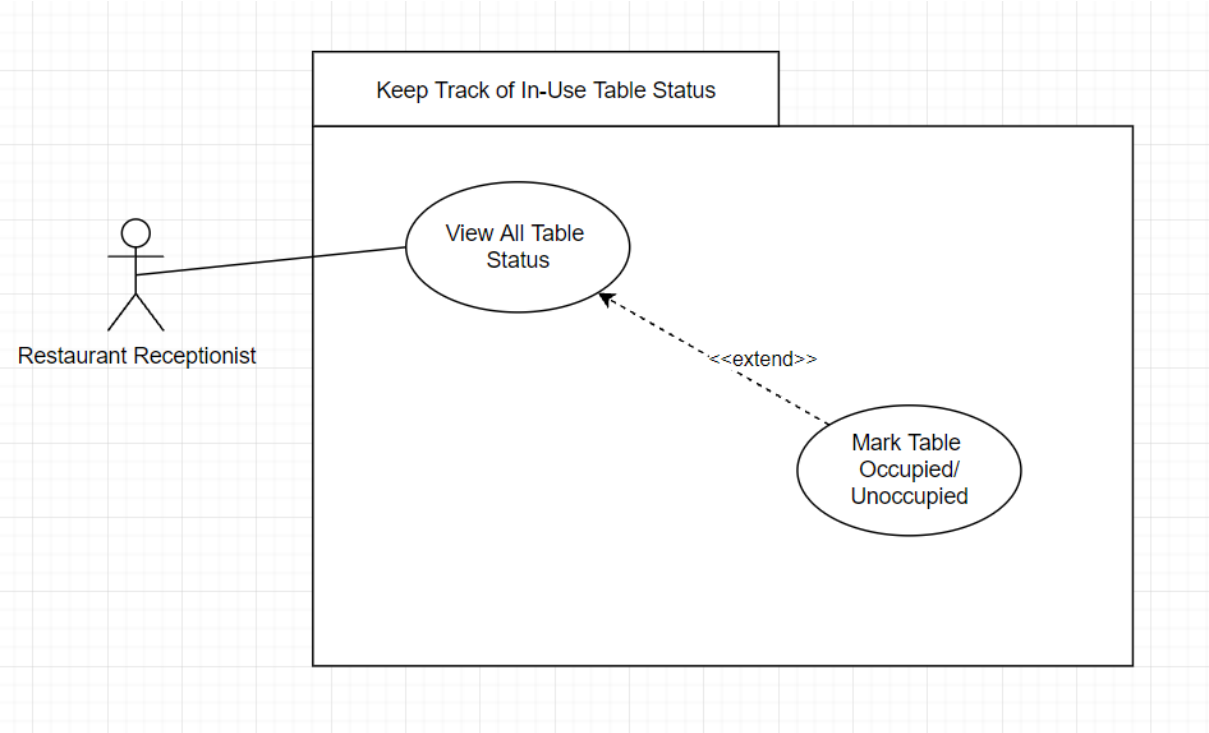
Link image

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



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.

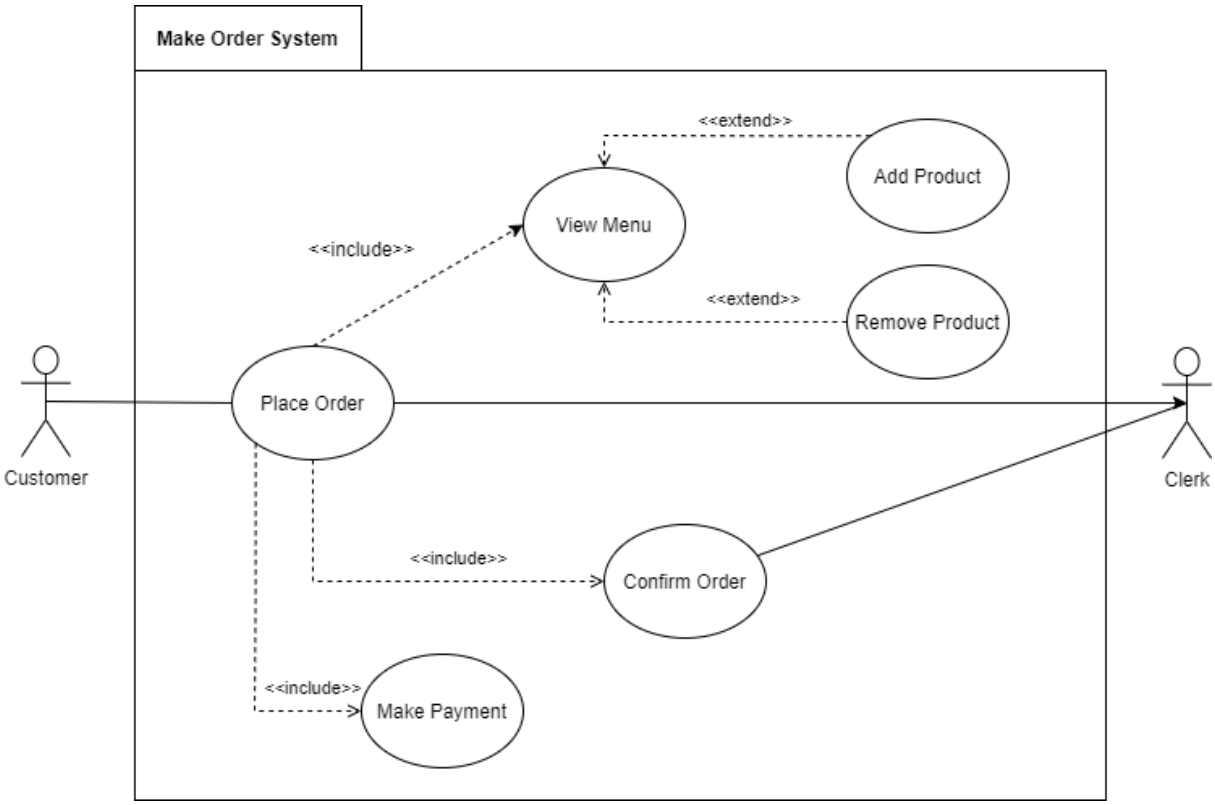
- Table UC-1.1: Table format for View All Table Status use case
- Table UC-1.2: Table format for Mark Table Occupied/ Unoccupied use case

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

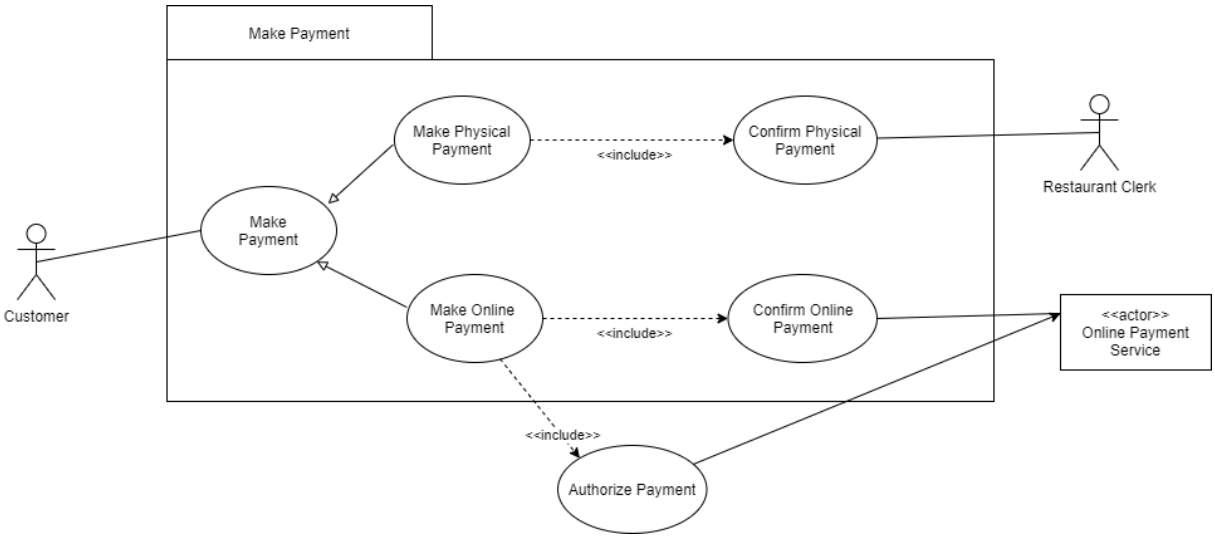
Use Case ID	UC-2.1
Use Case Name	Place Order
Description	Customers can order the food on the website of the restaurant
Actor(s)	Customer
Priority	High
Trigger	The Customer ask the System to place the order
Precondition(s)	Customer accessed the Order System through QR code
Postcondition(s)	The Order is 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).

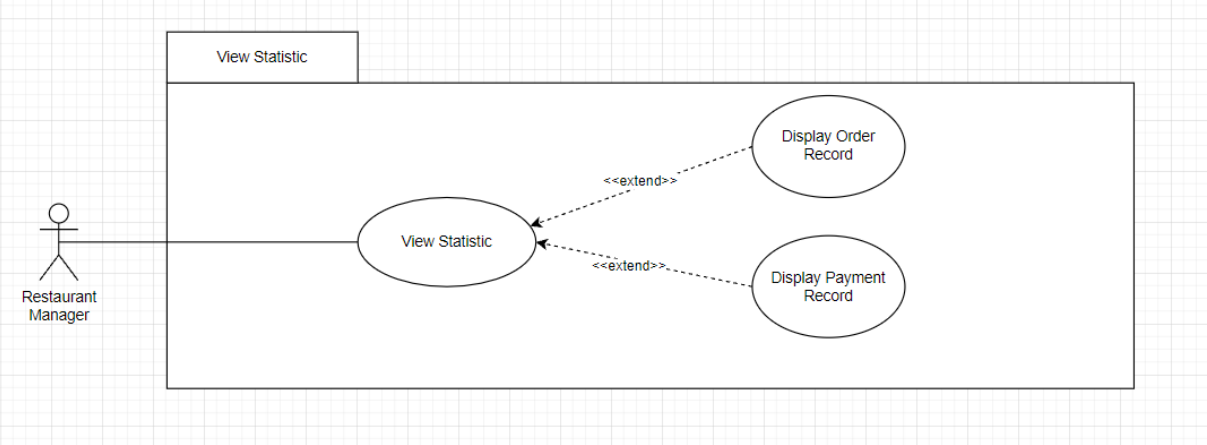
Use Case ID	UC-3.2
Use Case Name	Confirm Physical Payment
Description	After system sends a payment request to notify the Restaurant Clerk that customer has made payment, Receptionist confirms the payment is successful
Actor(s)	Restaurant Clerk
Priority	High
Trigger	Clerk receives a Payment Request from 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

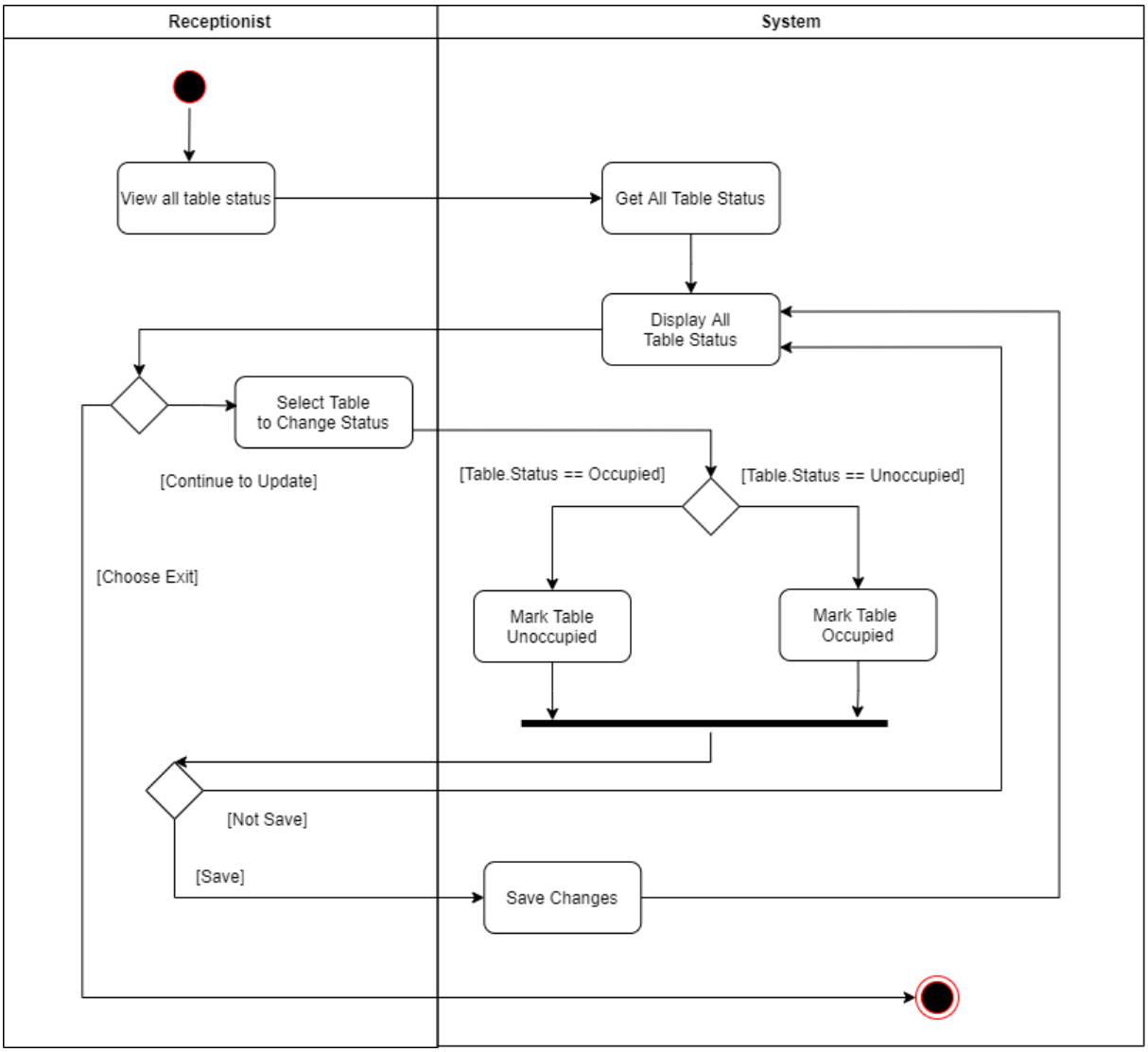
As we can see from the Table UC-4.1, the View Statistic use case does extend the Display Order Record and Display Payment Record use case. Therefore, we provide two use case tables for Display Order Record use case (Table UC-4.2) and Display Payment Record use case (Table UC-4.3)

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

Activity Diagram

Feature 1: Table Management



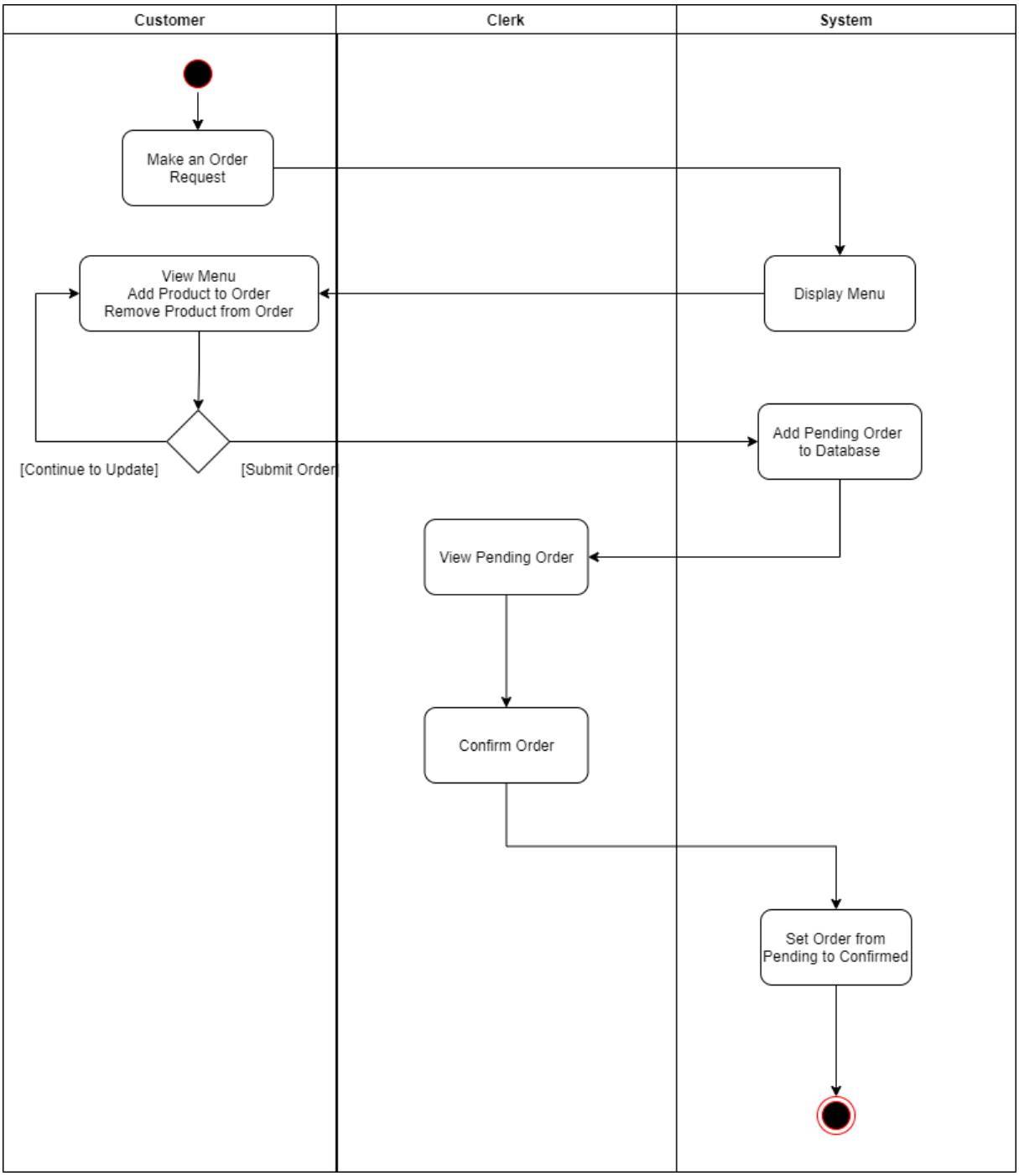
Visit this url to see the diagram better:

https://drive.google.com/file/d/1jxEeIkqhmeY-TZP2tCaWpxwTGcMlDgbn/view?usp=sharing

Above is the activity diagram and below is the description of Table management process:

1. The Process is divided into 2 swimlanes: Receptionist and the System
2. The Receptionist requests to view all table status in the Receptionist Tab
3. System request to get from database and display all table status on the screen.
4. Now, the Receptionist can choose to update one or many table's status by changing it from Occupied to Unoccupied or vice versa and save all changes to the database or not and exit, or the Receptionist just views the table status list and exit.

Feature 2: Order System



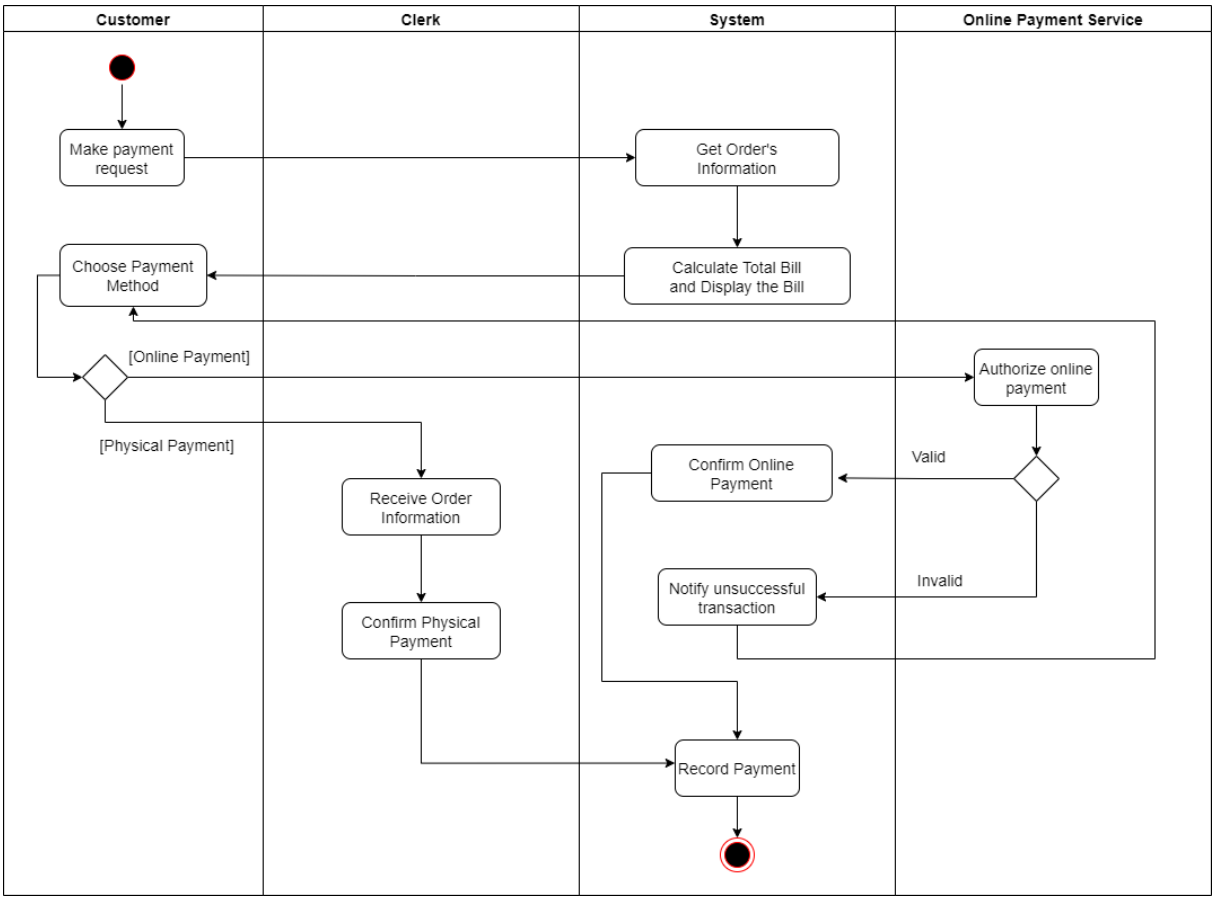
Visit this url to see the diagram better:

https://drive.google.com/file/d/1CpaXqspHaq24dzSCcwuvb97hEijfbXIn/view?usp=sharing

Above is the activity diagram of Order System Process

1. Order system activity diagram is divided into 3 swimlanes: Customer, Clerk and the System
2. First, after the customer accessed the table through QR code, a ‘Make Order’ request sent from the customer's device to the System, system will display the menu and create a temporary cart to cache the update (add, remove dish) that customer made to the order.
3. Customer submit the order, System will save the order into Order Table in the database as ‘pending’ state.
4. On the clerk side, when refreshing the ‘Order Management’ page, the clerk will view all the pending orders. Clerk can access the details of each order, then click the confirm button to set the state of the order to ‘confirmed’ and direct the customer to the payment page.

Feature 3: Making Payment



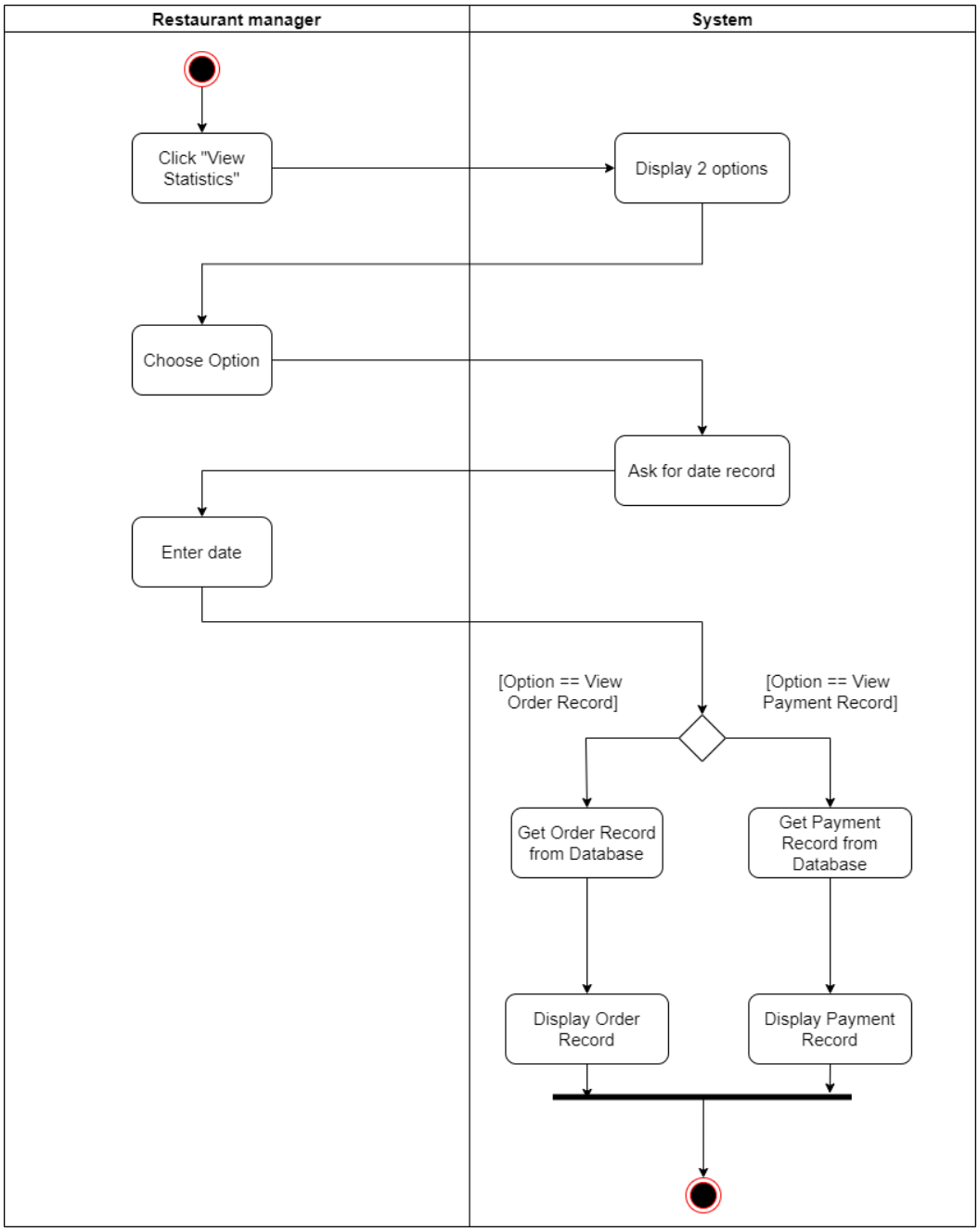
Visit this url to see the diagram better:

https://drive.google.com/file/d/1Ir37Nj0nooYfDgLoF46xDQz8dfX_BYkB/view?usp=sharing

Above is the activity diagram of Making Payment Process

1. The diagram is divided into 4 swimlanes: Customer, Clerk, System and Online Payment Service.
2. When the customers want to make their payment, they send a payment request to the system, the system then displays their bill and asks for the payment method they want to use.
    2.1.1. If customers choose to pay by physical payment method (cash, credit card), the clerk shall get notified about the payment. Clerk receives the payment and confirms it to the system. The transaction is recorded in the database.
    2.1.2. If customers choose to pay by online payment method, their payment is directed to online payment service. Payment service will verify their transaction and send back information to the system.
        2.1.2.1. If payment is failed, customers are navigated to choose the payment method page.
        2.1.2.2. If payment is made successful, the transaction is recorded in the database.

Feature 4: View Statistics



Visit this url to see the diagram better:

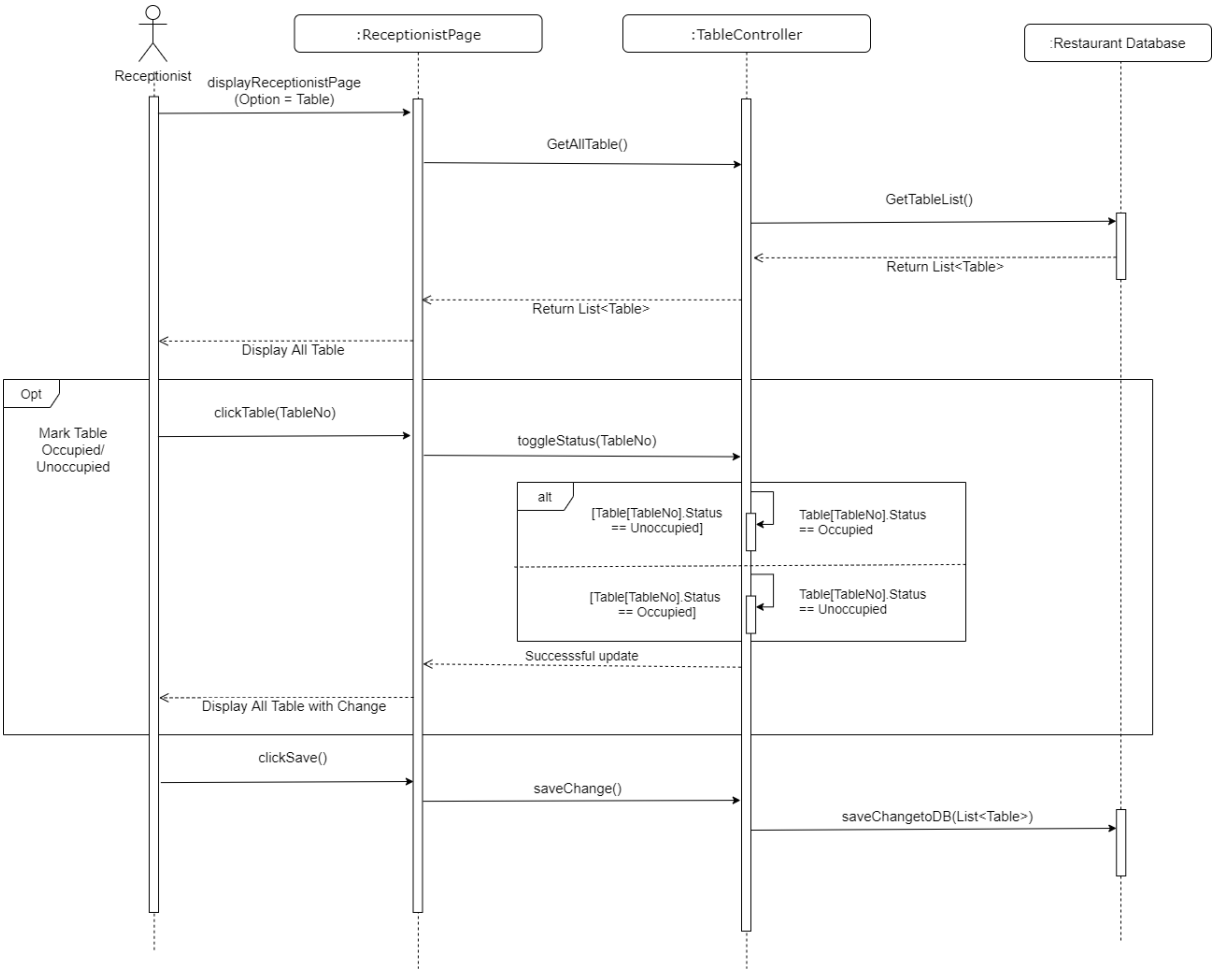
https://drive.google.com/file/d/17d7A3L6IpLX2O2GY98bVfQChMdwC0B3n/view?usp=sharing

Above is the activity diagram of View Statistics Process

1. The diagram is divided into 2 swimlanes: Customer and System
2. The manager chooses to view statistics
3. The system will display 2 options to view order record or payment record
4. The manager choose his/her option
5. The system will display a request date that the manager want to view
6. The manager enters the date or can leaves it blank
7. If the option is to view the order record, the system will display the order record on the screen. If the option is to view the payment record, the system will display the payment record on the screen.

Sequence Diagram

Feature 1: Table Management



Visit this url to see the diagram better:

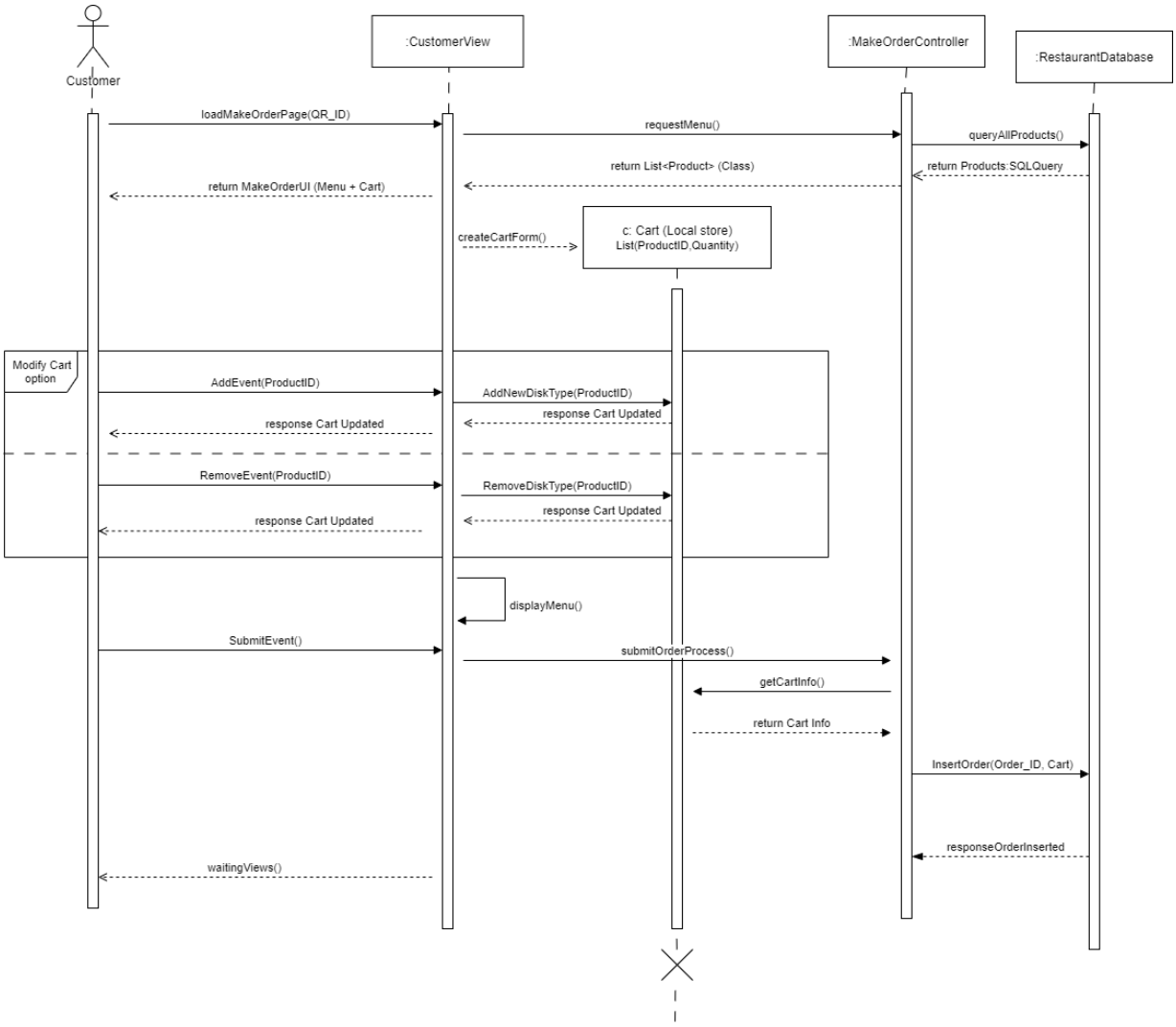
https://drive.google.com/file/d/1xDFNwIk9sSTJ1ehrfkcpwO0hmEh_2-Wk/view?usp=sharing

1. The receptionist opens the table option in the Receptionist Tab see all the table status. System sends a request to get the status of all table lists from Database by GetAllTable() and the system returns a table list (List<table>)).
2. A table status list will display on the screen to show all the tables with it's status.
3. Now, the receptionist can do as following:

   3.1 The receptionist presses the update button below a table, it's status will change from Occupied to Unoccupied or vice versa. After updating the table status successfully, the system will show the table list again with changes. After that, the receptionist clicks the save button (clickSave()) to save changes to the Database system or not. The receptionist can continue to update another table or can choose to exit.

   3.2 The receptionist just views all the table status and exit.

Feature 2: Order System (Make Order - Customer View)



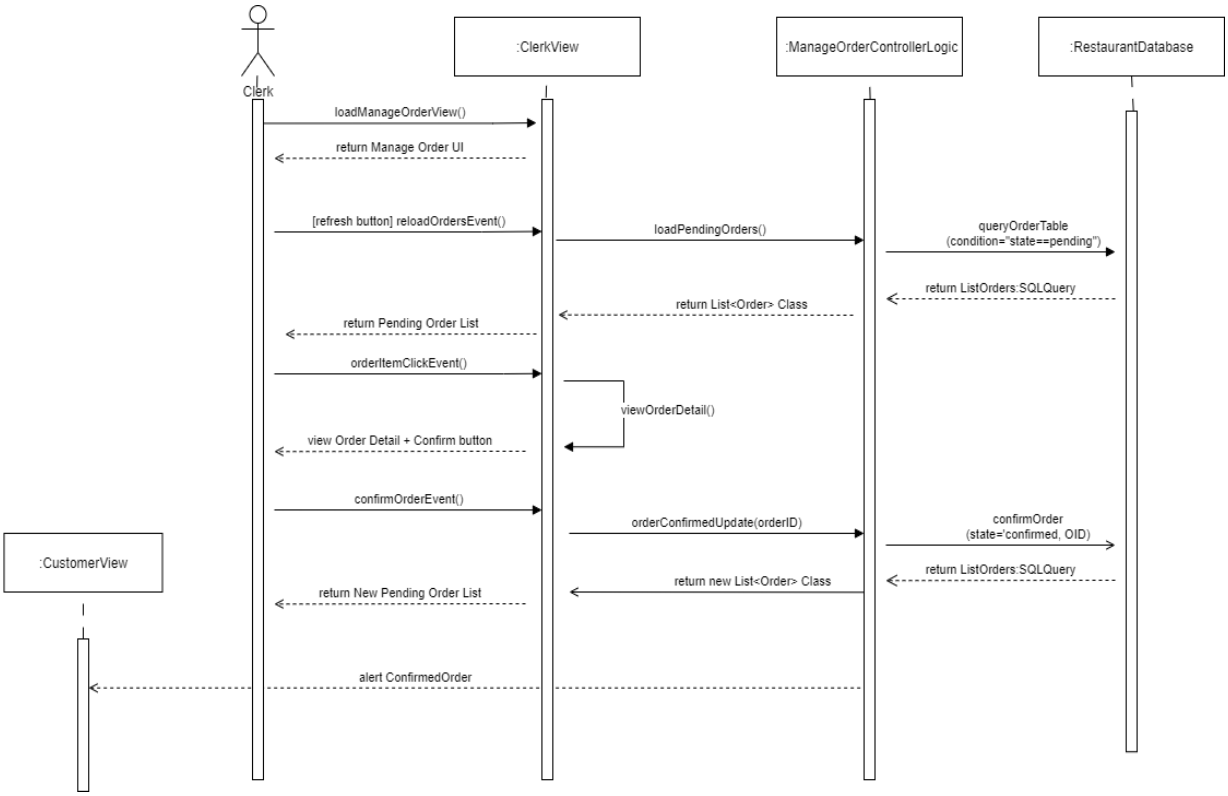
Visit this url, click on tab “Main Make Order” to see the diagram better:

https://drive.google.com/file/d/1cpehbG53OUSMxbfNWHnr81AZmXNcrOMO/view?usp=sharing

The sequence diagram above present first phase of make order process:

1. After the customer accesses the table through QR_code, MakeOrderView UI will appear at the customer screen. Before displaying, the view object will ask for Menu Info from the database.
2. A Cart object will be created and stored at customer local device to save the info about current order (disk added)
3. Customer views the menu, clicks the disk to view the details. Customer can set the quantities and click the add button to add the disk into the cart or remove added disk from the cart.
4. After getting the wanted cart, the customer clicks the submit button and an Order object will be created with information of Date time, Cart, QR of the table that made this order, state as ‘pending’ and inserted into the database.

Feature 3: Order System (Manage Order - Clerk View)



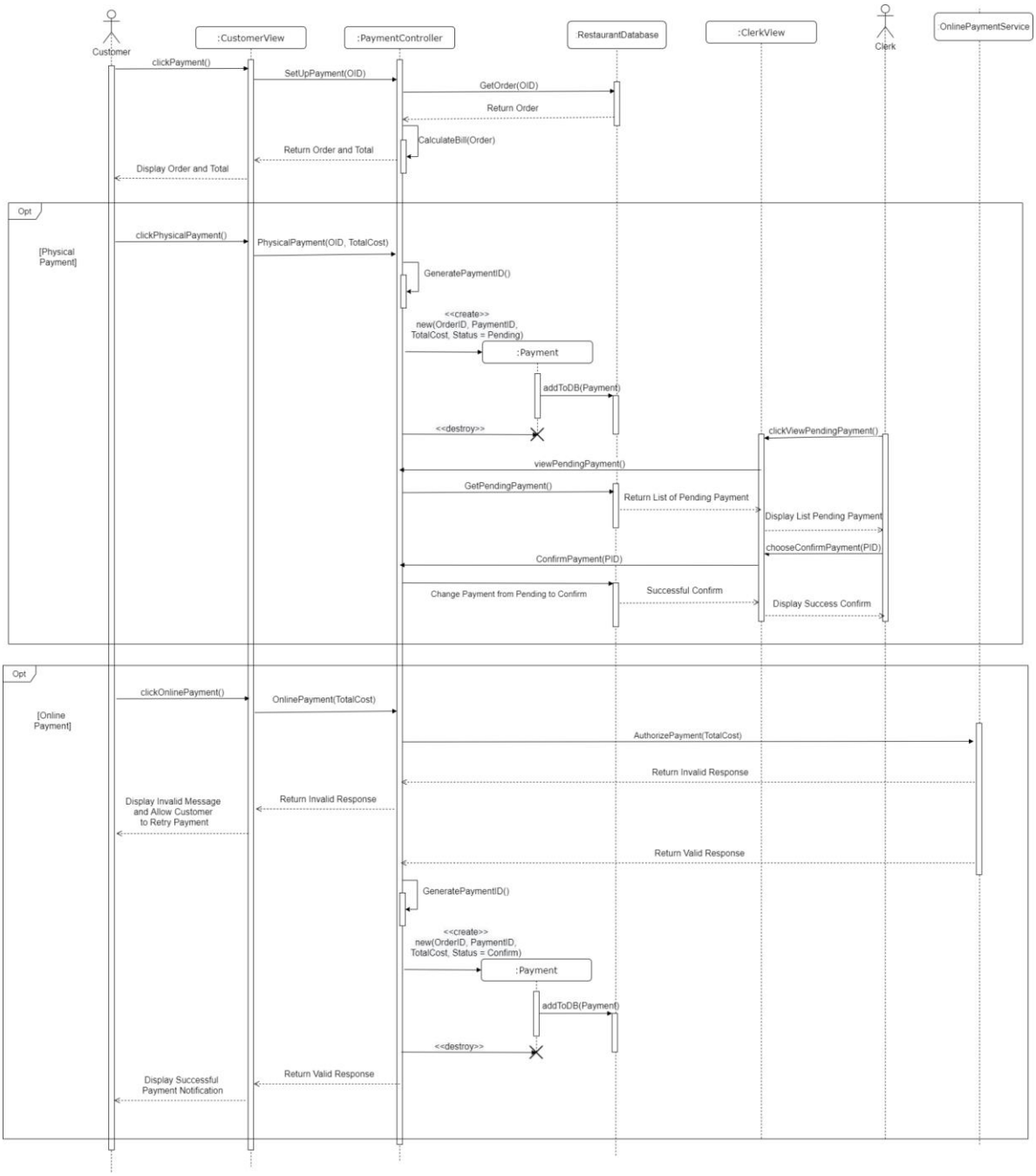
Visit this url, click on tab “Main Manage Order” to see the diagram better:

https://drive.google.com/file/d/1cpehbG53OUSMxbfNWHnr81AZmXNcrOMO/view?usp=sharing

The sequence diagram above present second phase of the make order process:

1. Clerk starts the special machine that is assigned to manage the order.
2. Clerk click the refresh button to load new pending orders from the database
3. Clerk view the detail the earliest order and click confirm, set the order’s state to ‘confirmed’
4. The customer accessing table that submits the confirmed order will be directed to the payment page, clerk will refresh to get a new list of pending orders.

Feature 3: Making Payment



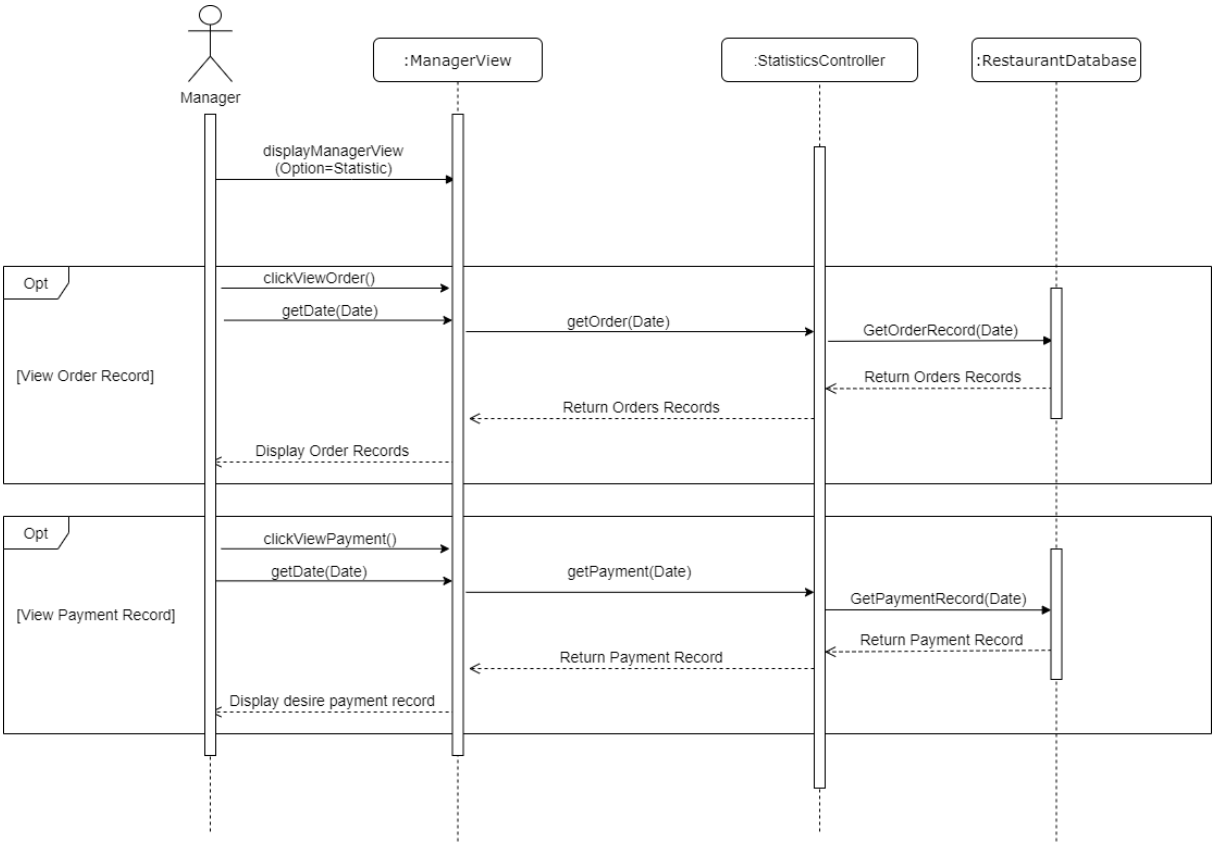
Visit this url to see the diagram better:

https://drive.google.com/file/d/1yNlu978lmTCuTrYxkh3oboYnmG2memGA/view?usp=sharing

1. Customers choose to make their payment, (clickPayment( ) is called and navigate them to the payment page) after receiving the request, payment controller retrieves the order information and calculates their bills then displays it.
2. Customer can choose two payment methods:
    2.1. If customers choose to make physical payment, “clickPhysicalPayment()” is called, , their payment request is sent to the payment controller to handle (PhysicalPayment(TotalCost)). Payment controller creates a payment object storing payment information and stores it to the database.
        2.1.1. Every time clerk view is refreshed by clerk, the view display payment pending list through method “getPendingPayment()” and return it to clerk view
        2.1.2. Clerk chooses to confirm a payment in the pending list, the payment status is changed from pending to confirmed. After that, a new pending payment list is displayed.
    2.2. If customers choose to make online payment, “clickOnlinelPayment()” is called, their payment request is sent to payment controller to handle (OnlinePayment(TotalCost)) Payment controller then send payment request to an online payment service (AuthorizePayment(TotalCost)), the service handles the transaction and returns back information to the system.
        2.2.1. If payment is invalid, customer view will navigate to the payment page, customer can choose to make online payment again or change to physical payment.

2.2.2. If payment is valid, payment object will be created and store their payment information to the database and customer view will display the successful payment notification

Feature 4: View Statistic



Visit this url to see the diagram better:

https://drive.google.com/file/d/15ROJtDalUv-EFI8x44Oql1AWT4vnk4DH/view?usp=sharing

1. The manager clicks the statistics button in the manager view.
2. Now, the manager can do as following:

2.1 The manager clicks to view the order record option displayed on the screen.

2.1.1 System displays dialog asks the Date

2.1.1.1 The manager enter the date that he/she want to view

2.1.1.1.1 System gets the orders made on that date from Database (by GetOrderRecord(date)) and displays those order records on the screen.

2.1.1.2 The manager leaves the date blank

2.1.1.2.1 System gets all of the orders made on that date from Database (by GetOrderRecord(date)) and displays those order records on the screen.

2.2 The manager clicks to view the payment record option displayed on the screen.

2.2.1 System displays dialog asks the Date

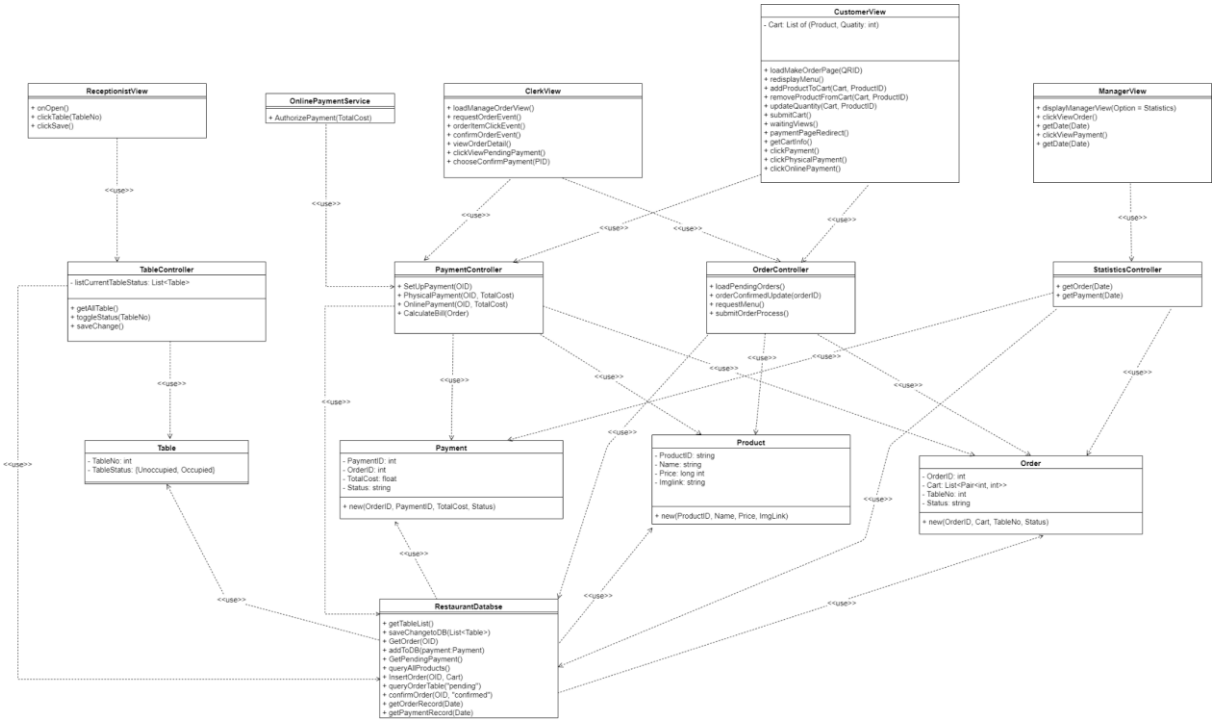
2.2.1.1 The manager enter the date that he/she want to view

2.2.1.1.1 System gets the payment made on that date from Database (by GetPaymentRecord(date)) and displays desired payment records on the screen.

2.2.1.2 The manager leaves the date blank

2.2.1.2.1 System gets all of the payment made on that date from Database (by GetPaymentRecord(date)) and displays desired payment records on the screen.

Class Diagram

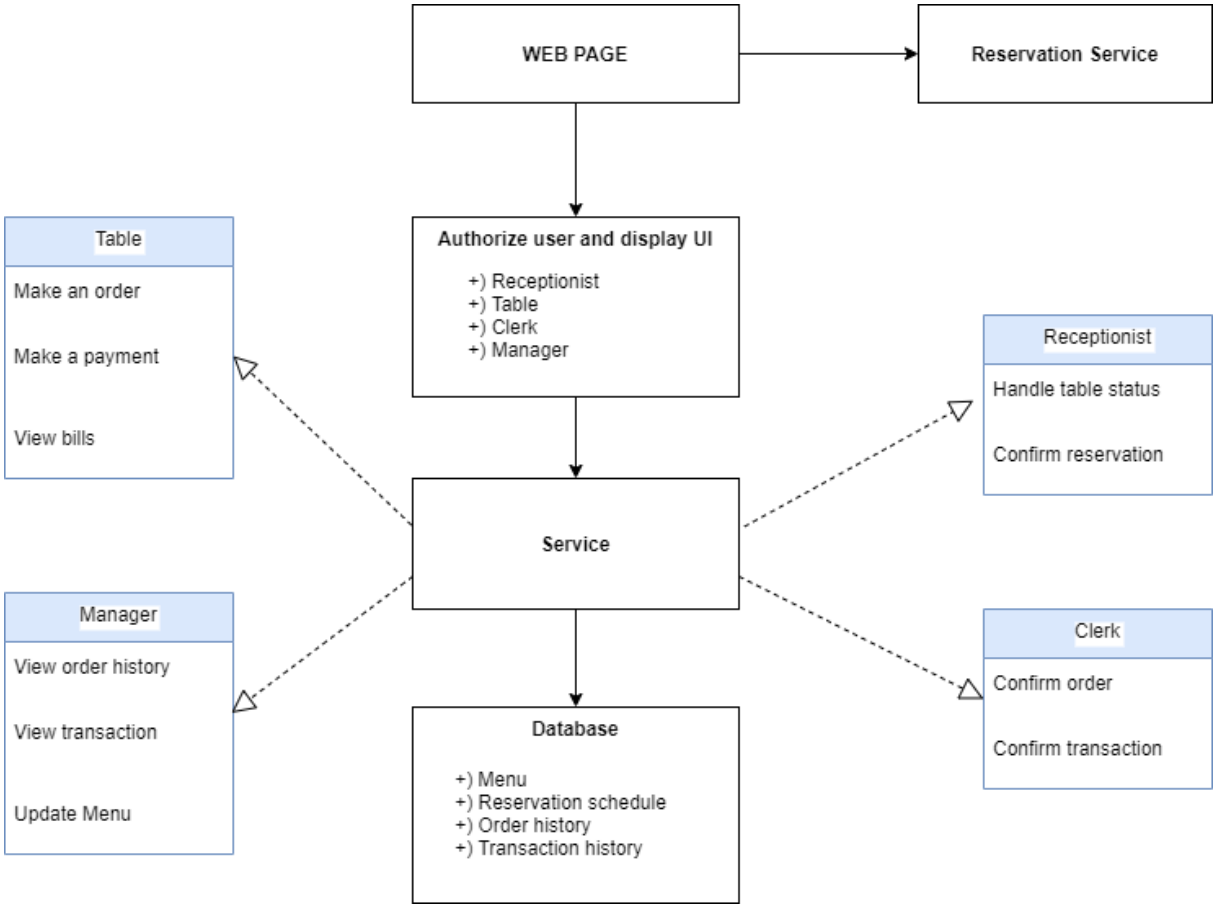


Visit this url to see the diagram better:

https://drive.google.com/file/d/1NRI6vUYea5q77U-lCA8HC1iPsOPAPfUf/view?usp=sharing

Architecture design

1. General architecture



Visit this url, click on tab “General Architecture” to see the diagram better:

https://drive.google.com/file/d/1t50r_K9CmGWyY3GQCv1Z_rmOE-OCvZEA/view?usp=sharing

Application has 5 different views:

+) Reservation View: For customers who want to make a reservation

+) Table View: For customers in the restaurant want to make orders and payment

+) Receptionist View: For the receptionist

+) Clerk View: For the clerk

+) Manager View: For the manager

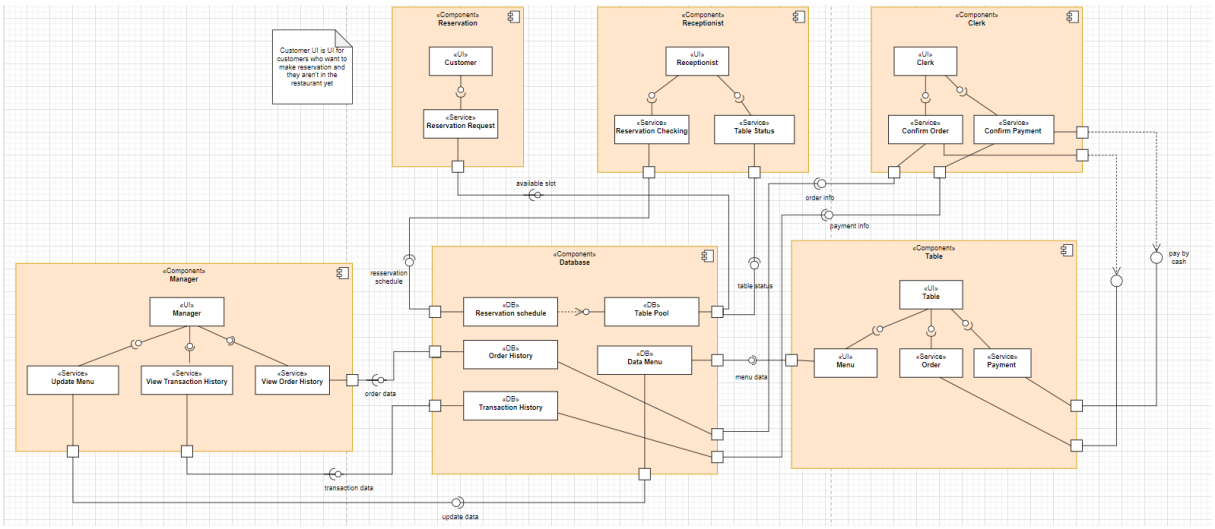
When customers make a reservation they shall make a reservation through reservation UI. They are not asked to authorize their identity to use the application.

For customers who are in the restaurant, the clerk, receptionist, manager have to authorize their identity.

After the authorization process, they can use their provided service.

The final layer is the database which stores information about menu, order history, transaction history and reservation schedule.

2. Component Diagram



Visit this url, click on tab “Component” to see the diagram better:
https://drive.google.com/file/d/1Y0-DoI8gLCqU2Zvw5cDXg_dXRwQ823L6/view?usp=sharing

There are 6 major components:

+) Reservation: handle reservation service

+) Receptionist: handle services that offer to the receptionist

+) Clerk: handle services that offer to the clerk

+) Table: handle services that offer to customers in the restaurant

+) Manager: handle services that offer to the manager

+) Database: store database of the system