

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	4
Stakeholders	5
Project scope	5
Project Justification	5
User Story	6
General Feature of the Project	6
Assumptions	7
Project Context	8
Functional Requirements	10
Non-functional Requirements	10
Project's Use-Case Diagram	12
Work Assignment	13
Detailed Use-case	14
Feature 1: Table Management	14
Feature 2: Order System	17
Feature 3: Making Payment	20
Feature 4: View Statistics	25
Activity Diagram	28
Feature 1: Table Management	28
Feature 2: Order System	30
Feature 3: Making Payment	31
Feature 4: View Statistics	34
Sequence Diagram	36
Feature 1: Table Management	36
Feature 2: Order System (Make Order - Customer View)	38
Feature 2: Order System (Manage Order - Clerk View)	40
Feature 3: Making Payment	41
Feature 4: View Statistic	44
Class Diagram	46
Architecture design	50

General architecture	50
Component Diagram	52
User Interface and Screen of flow	53
Customer View	53
Clerk View	59
Receptionist View	62
Manager View	64
GitHub repository	66

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.

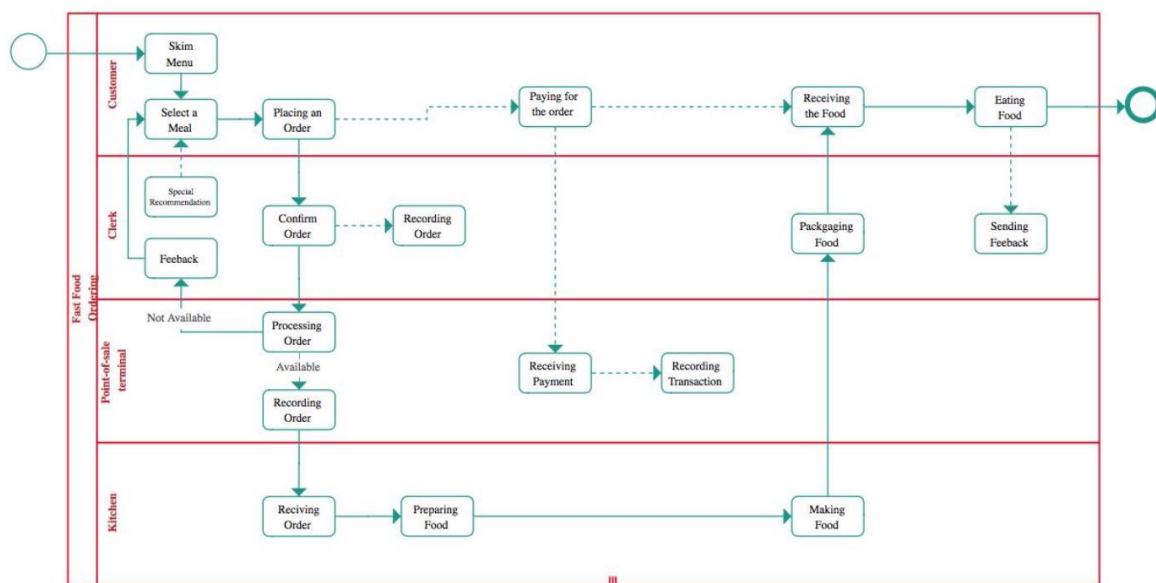


Figure 1: Customer-drawing workflow

Stakeholders

Stakeholder	Role	Description
Restaurant Owner	Product Owner	Who owns the application
Restaurant Manager	Restaurant Staff	Who is able to see the statistics on Order and Payment recorded in the Database
Restaurant Clerk	Restaurant Staff	Who is responsible for handling customer's orders and payments.
Restaurant Receptionist	Restaurant Staff	Who is responsible for keeping track of table status
Customer	Restaurant Customer	Who will use the application to 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 customer's submitted order
- I can confirm customer's physical payment, either by card or by cash

As the manager of the restaurant:

- I can view the restaurant's transaction (order and payment information)

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, card or online payment.

Feature 4: View Statistics

Daily orders and payments are recorded and this feature allows the restaurant manager to view the restaurant statistics (order and payment information).

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

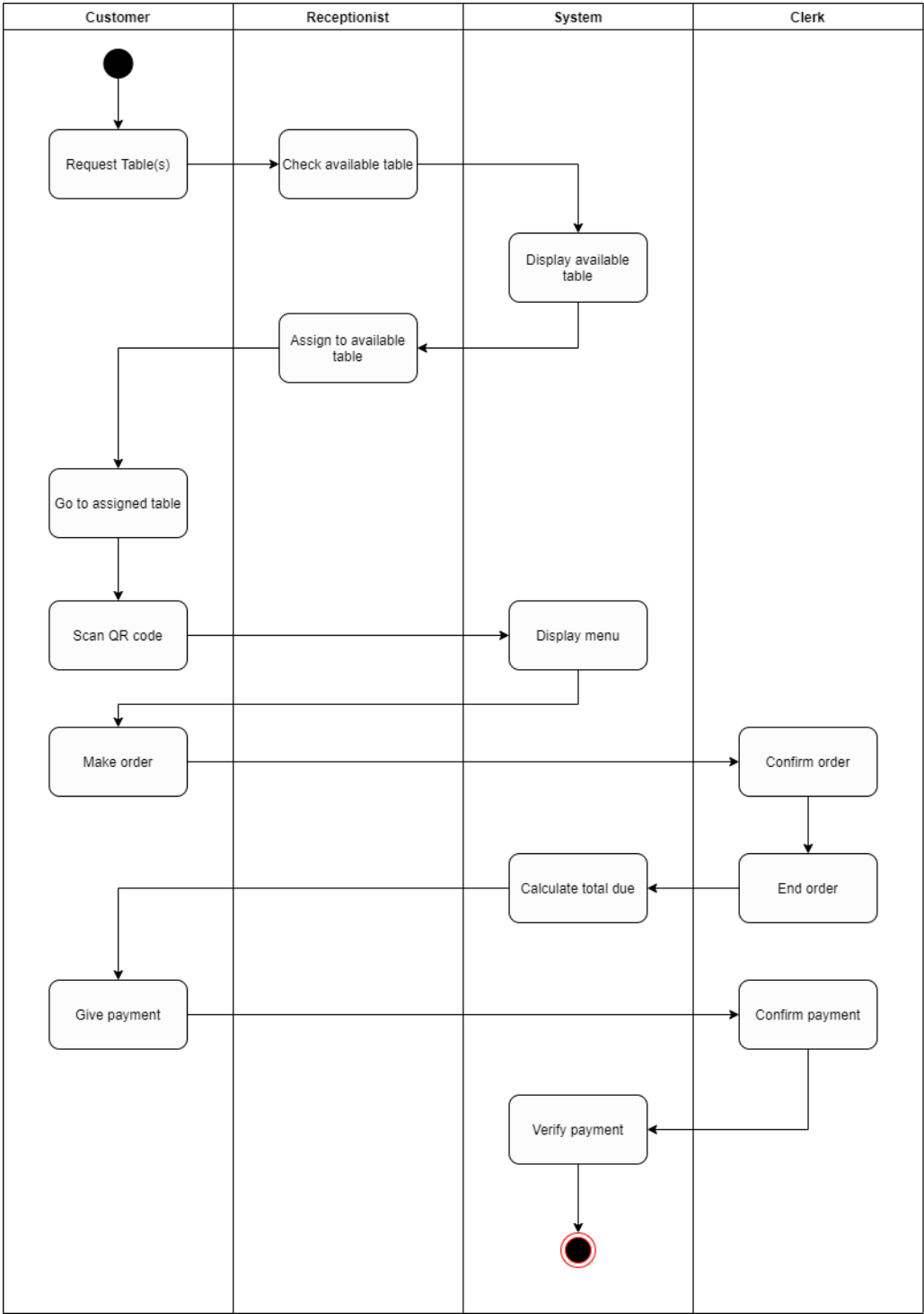
Business Model: Fine dining restaurant

Restaurant Customer: Upper Middle Class Customer

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

Dining service: Eat-in restaurant

Business Flow:



[Link image](#)

Functional Requirements

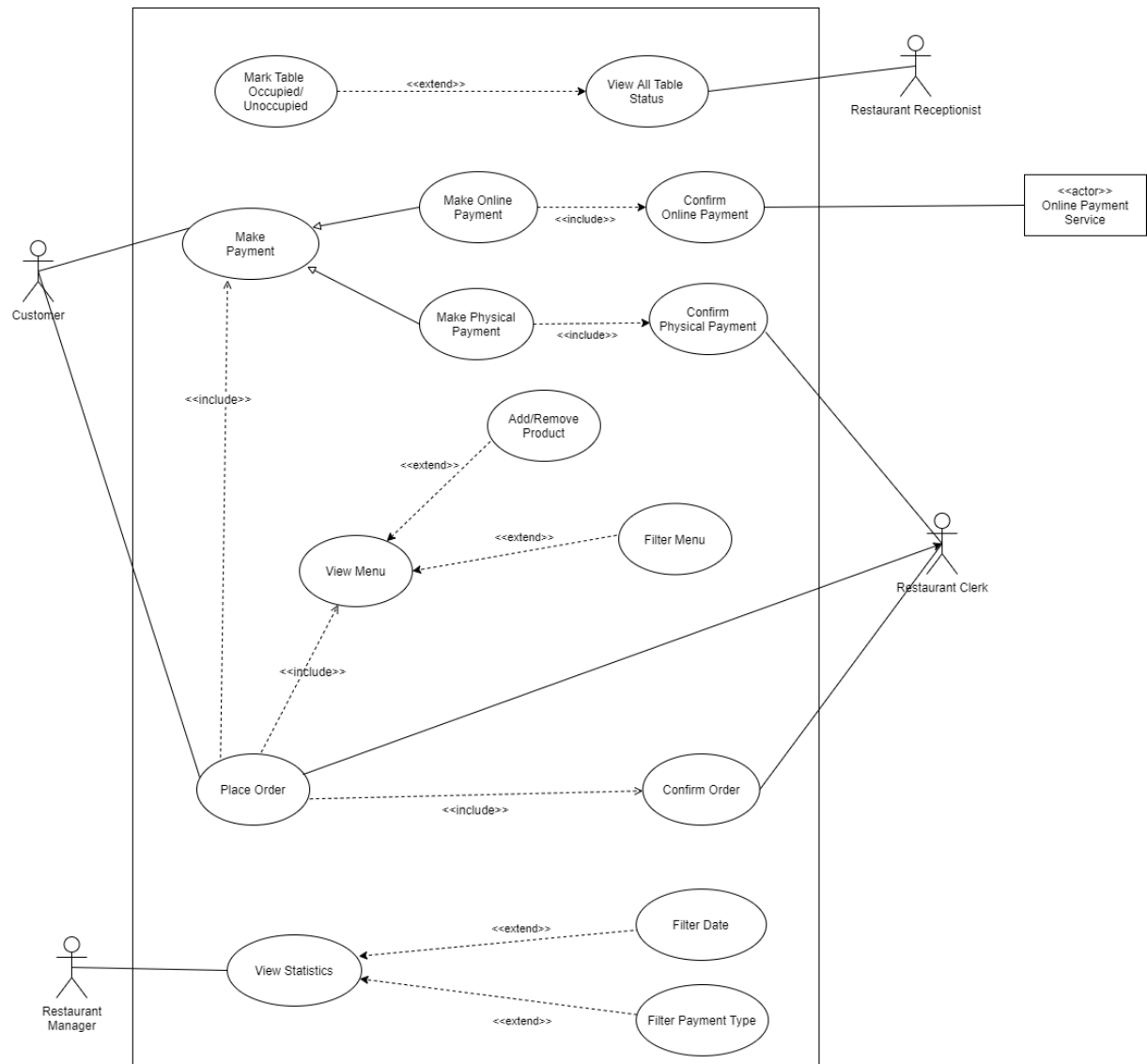
Requirement ID	Requirement	Priority	Comments
FR001	Restaurant Receptionist can view and update table status	Medium	
FR002	Customer can view the menu, make food orders and Clerk can confirm the food order	High	
FR003	Customer can make payment for their meal either by direct payment or third-party service	High	
FR004	Restaurant Manager can view all the transaction recorded by the System	Low	

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



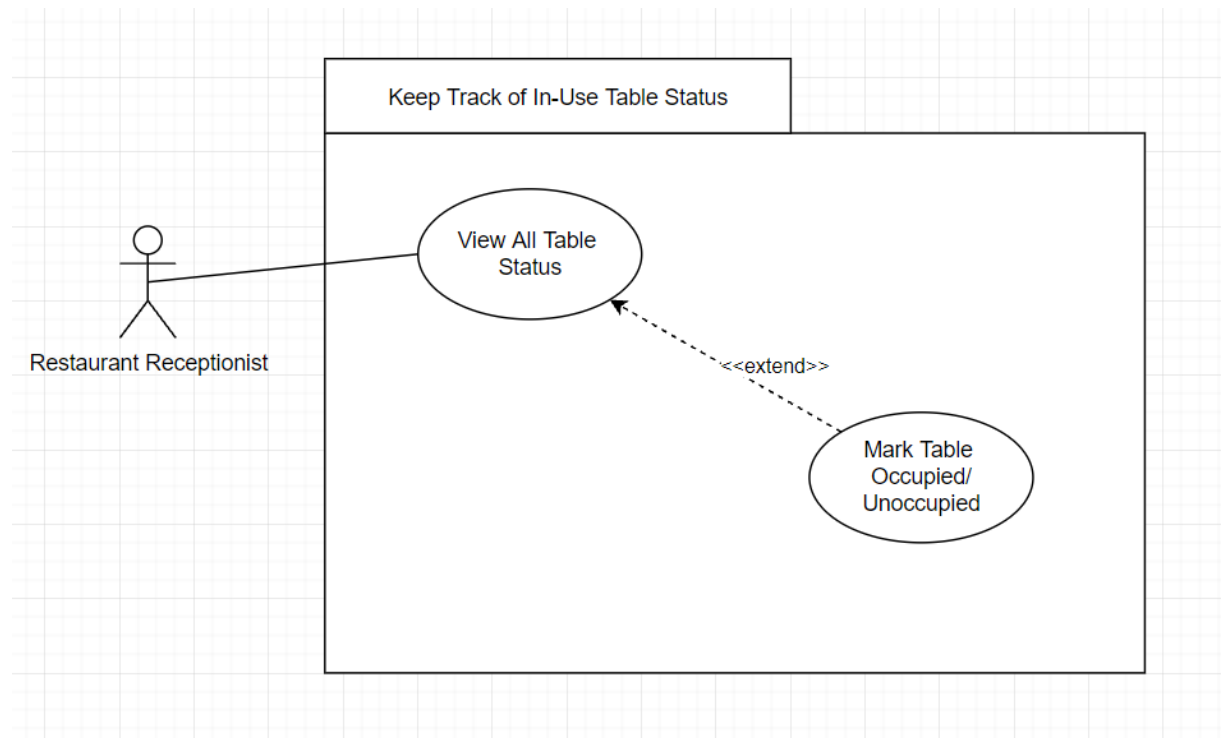
[Link image](#)

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
-------------	--------

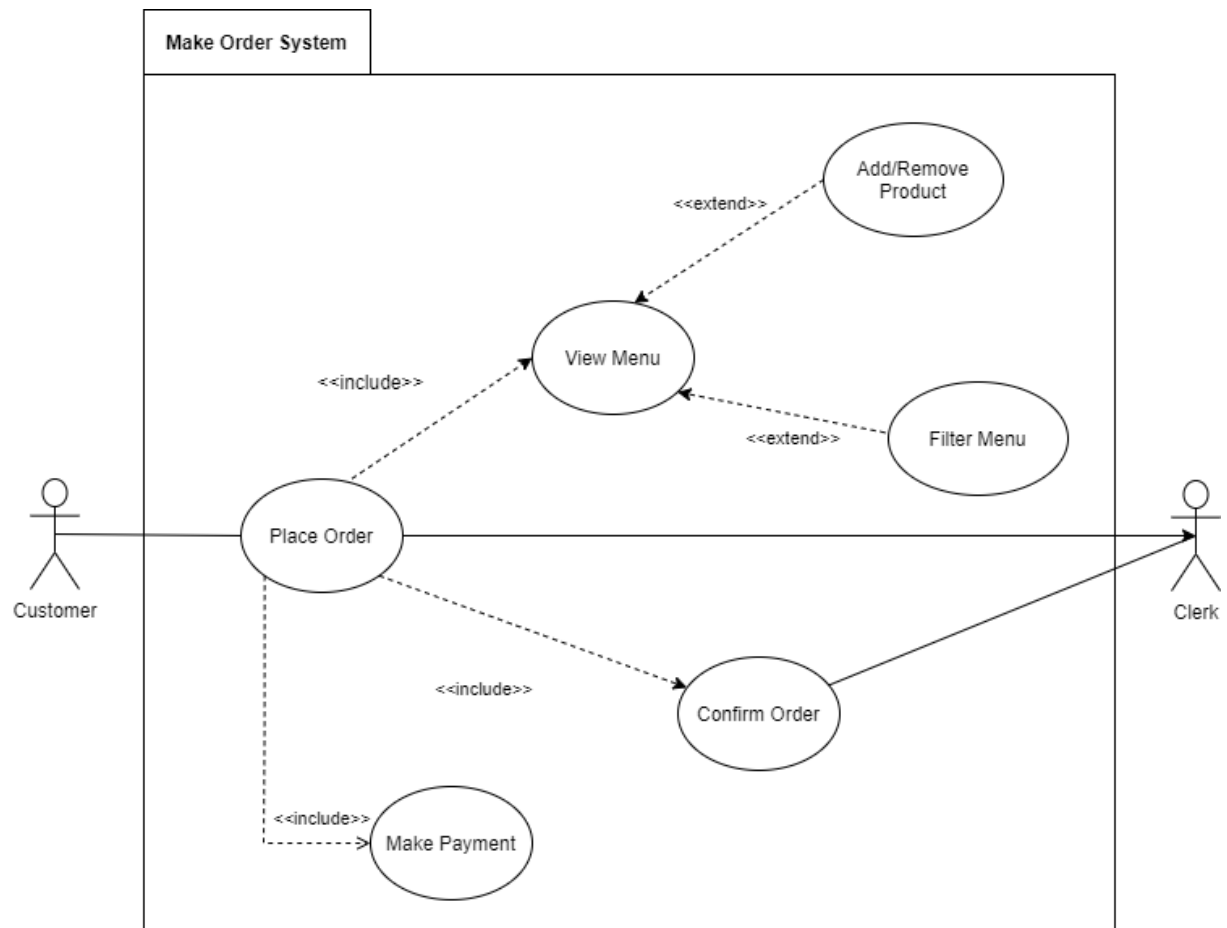
Use Case Name	View All Table Status
Description	Receptionist can view all table status in the restaurant (occupied or unoccupied)
Actor(s)	Restaurant Receptionist
Priority	Medium
Trigger	Receptionist chooses to View All Table Status
Precondition(s)	None
Postcondition(s)	System displays all table status
Basic Flow	<ol style="list-style-type: none"> 1. Receptionist chooses to View All Table 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	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	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).

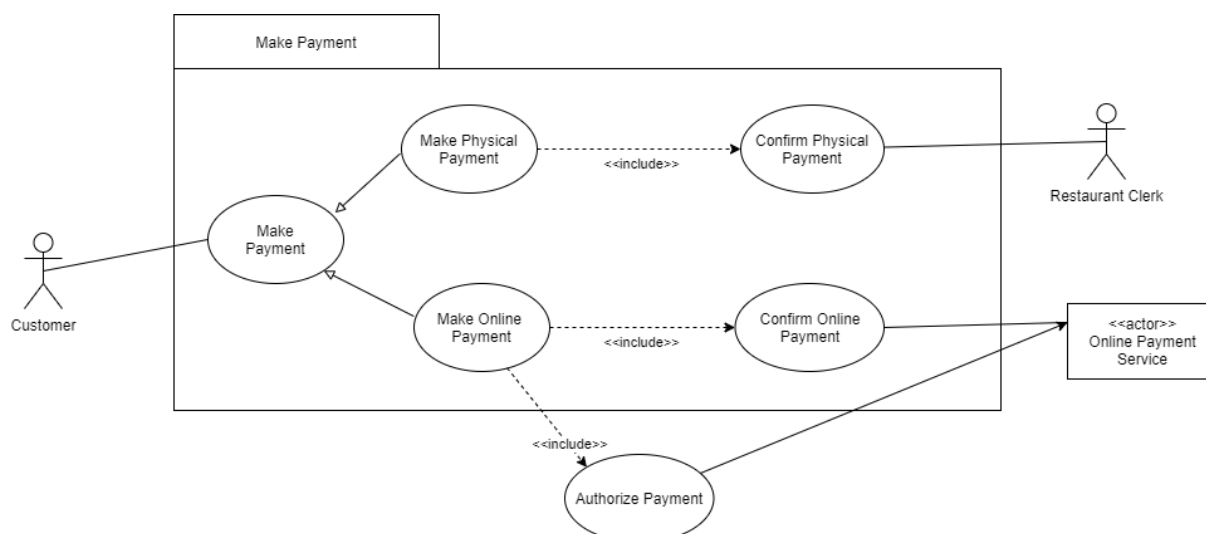
Use Case ID	UC-2.2
-------------	--------

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	<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 tab 2. System display the Pending Order 3. Clerk can views the Order's Information 4. Clerk confirms the Order 5. 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

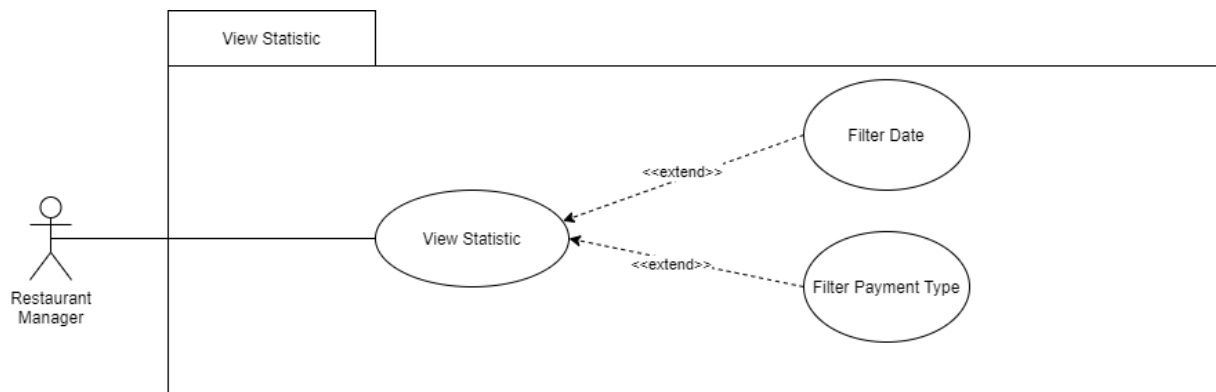
Basic Flow	<ol style="list-style-type: none"> 1. System gets the Orders's information 2. System calculates and display the bill 3. System sends Customer to the Third-party website to make payment 4. System sends Payment Information to Online Payment Service 5. Use Case continues at Confirm Online Payment 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 valid 2. System displays the validation to the Customer 3. 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 valid 3a. 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

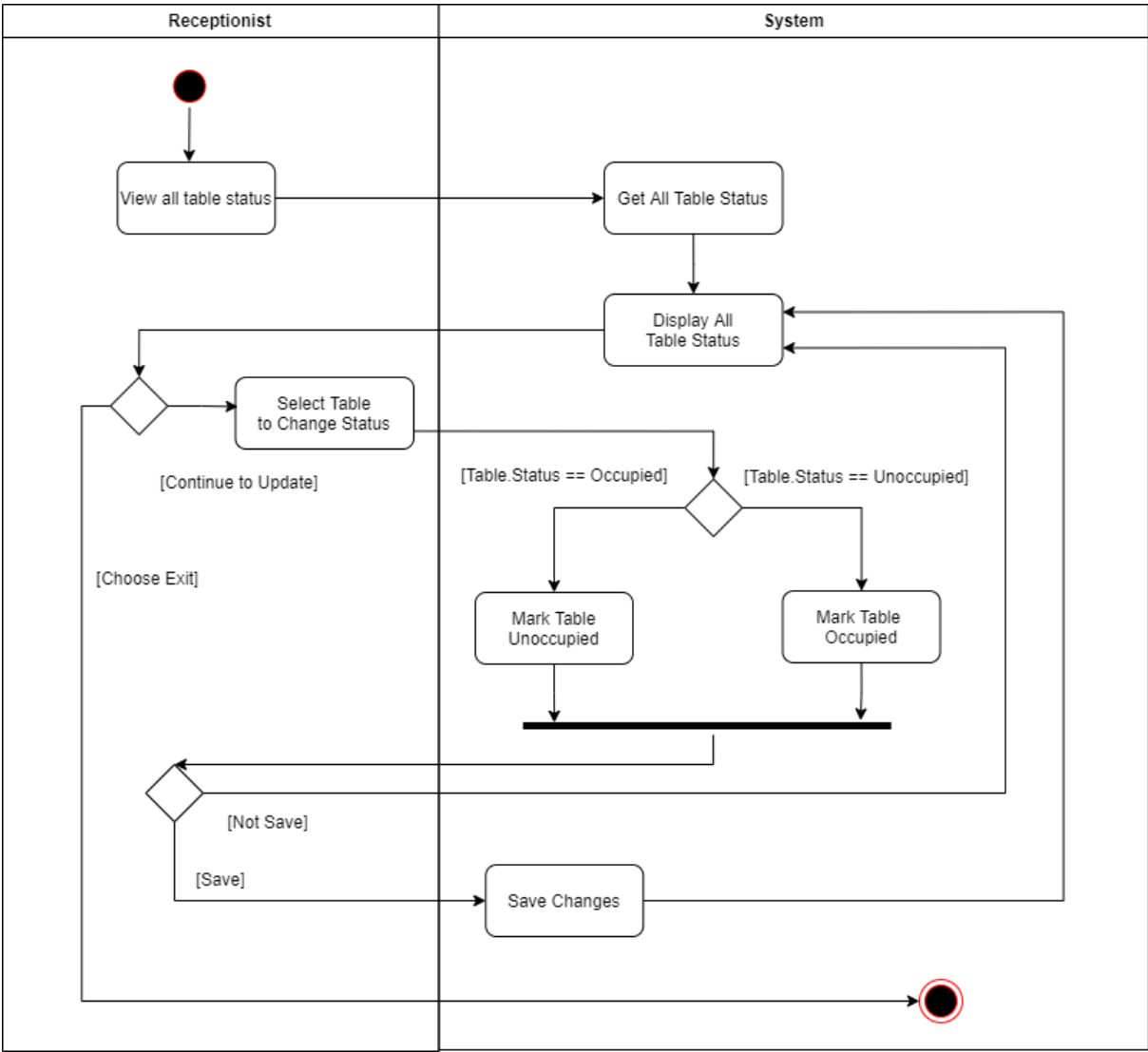
- Table UC-4.1: Table format for View Statistics use case

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.
--	---

Activity Diagram

Feature 1: Table Management



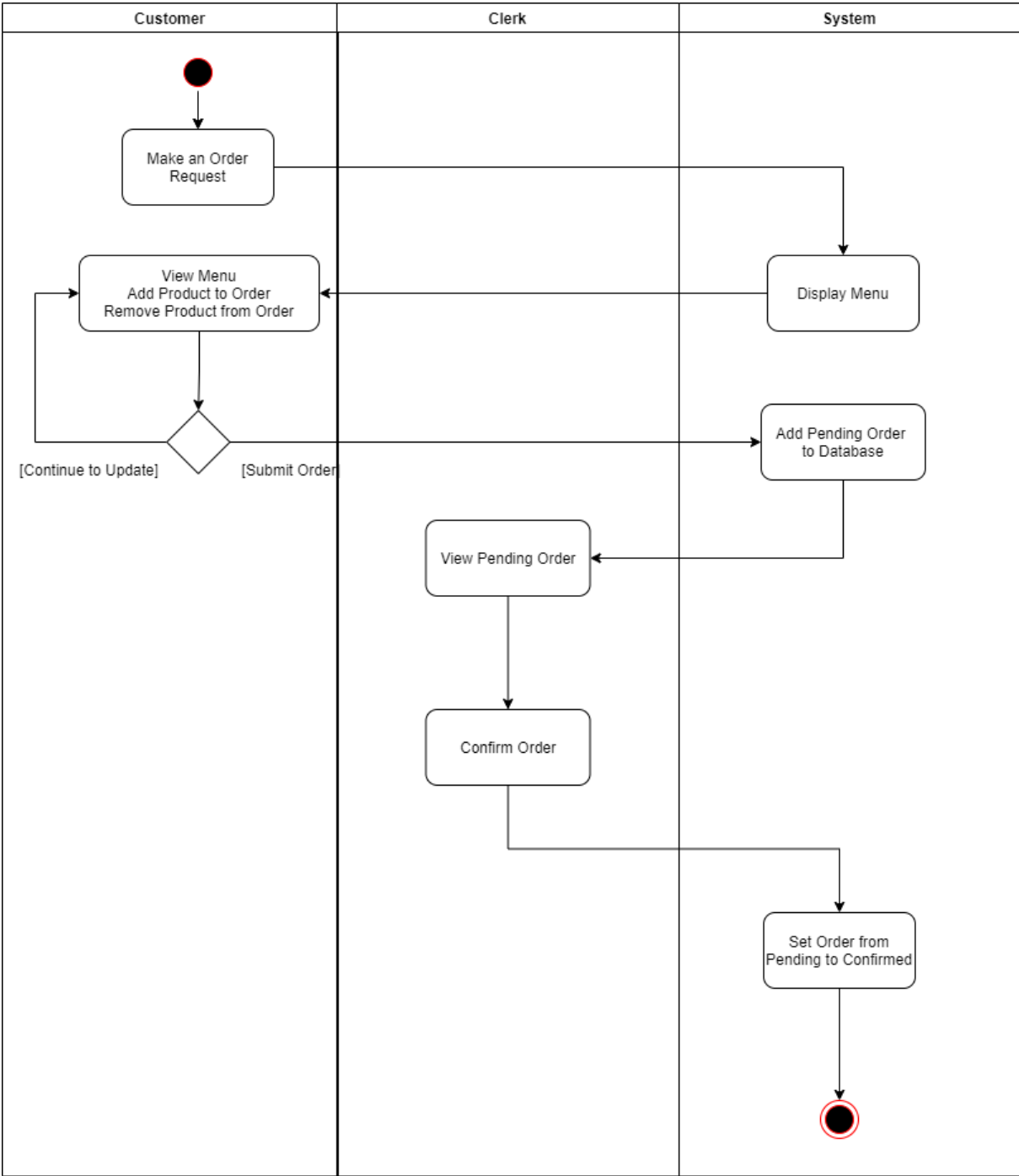
Visit this url to see the diagram better:

<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



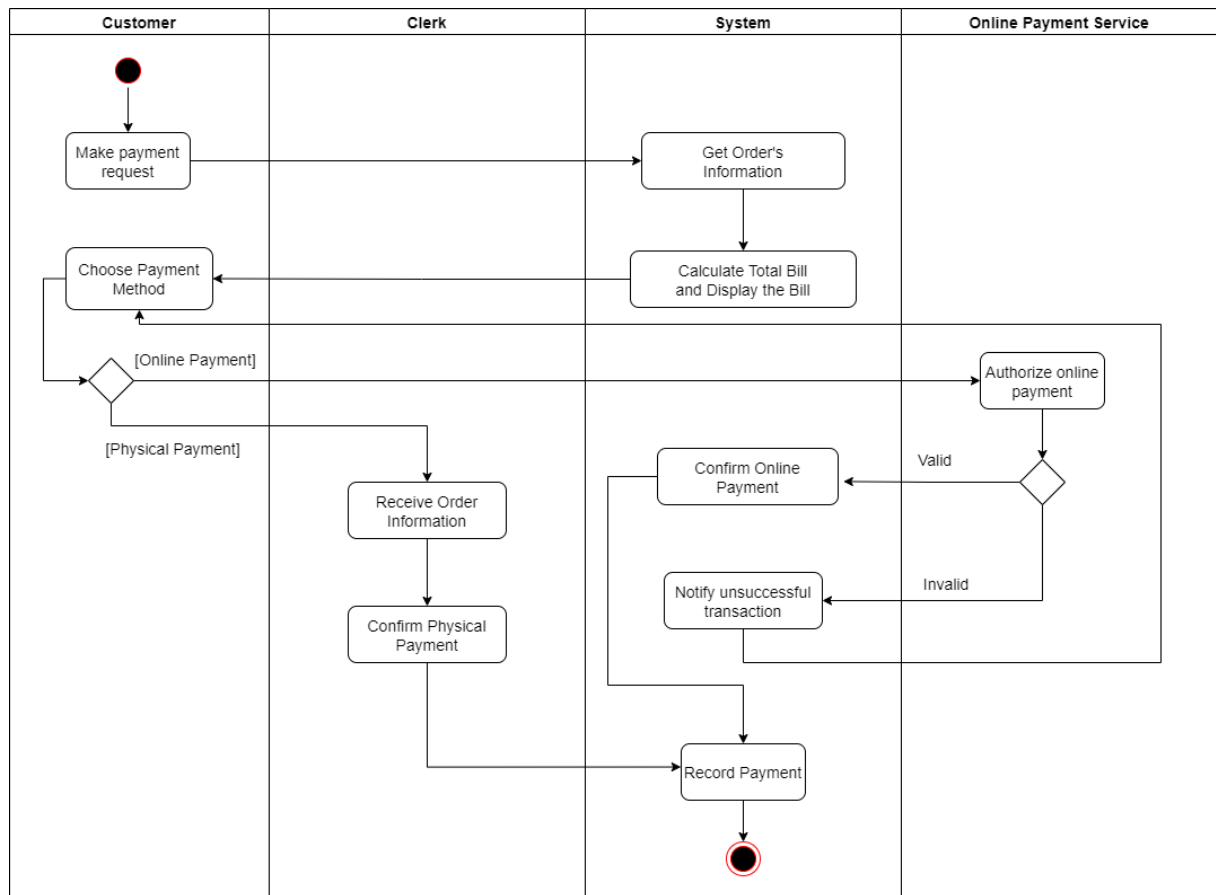
Visit this url to see the diagram better:

<https://drive.google.com/file/d/1CpaXqspHag24dzSCcwuvb97hEijfbXIn/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



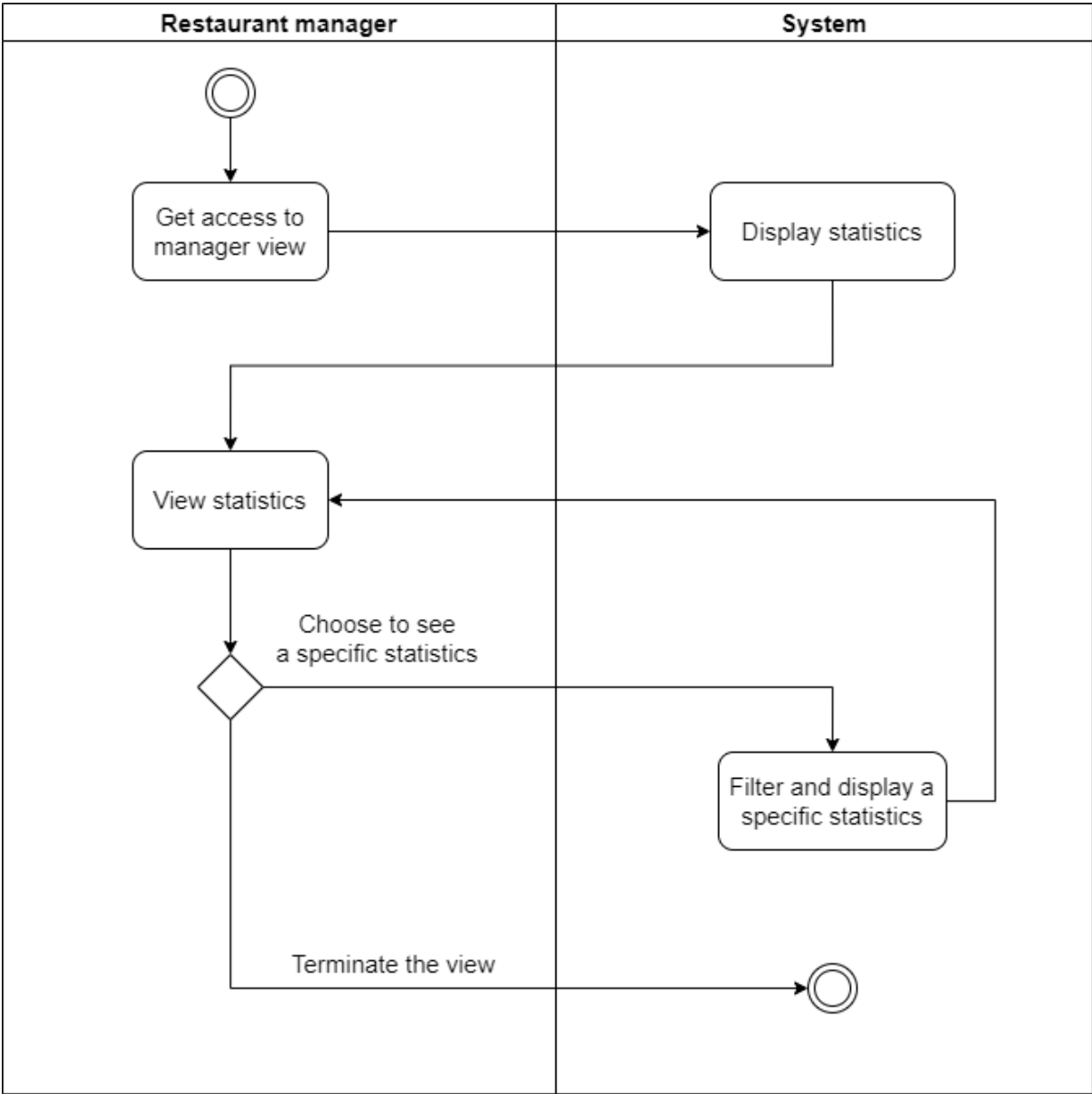
Visit this url to see the diagram better:

https://drive.google.com/file/d/1lr37Nj0nooYfDgLoF46xDQz8dfX_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:

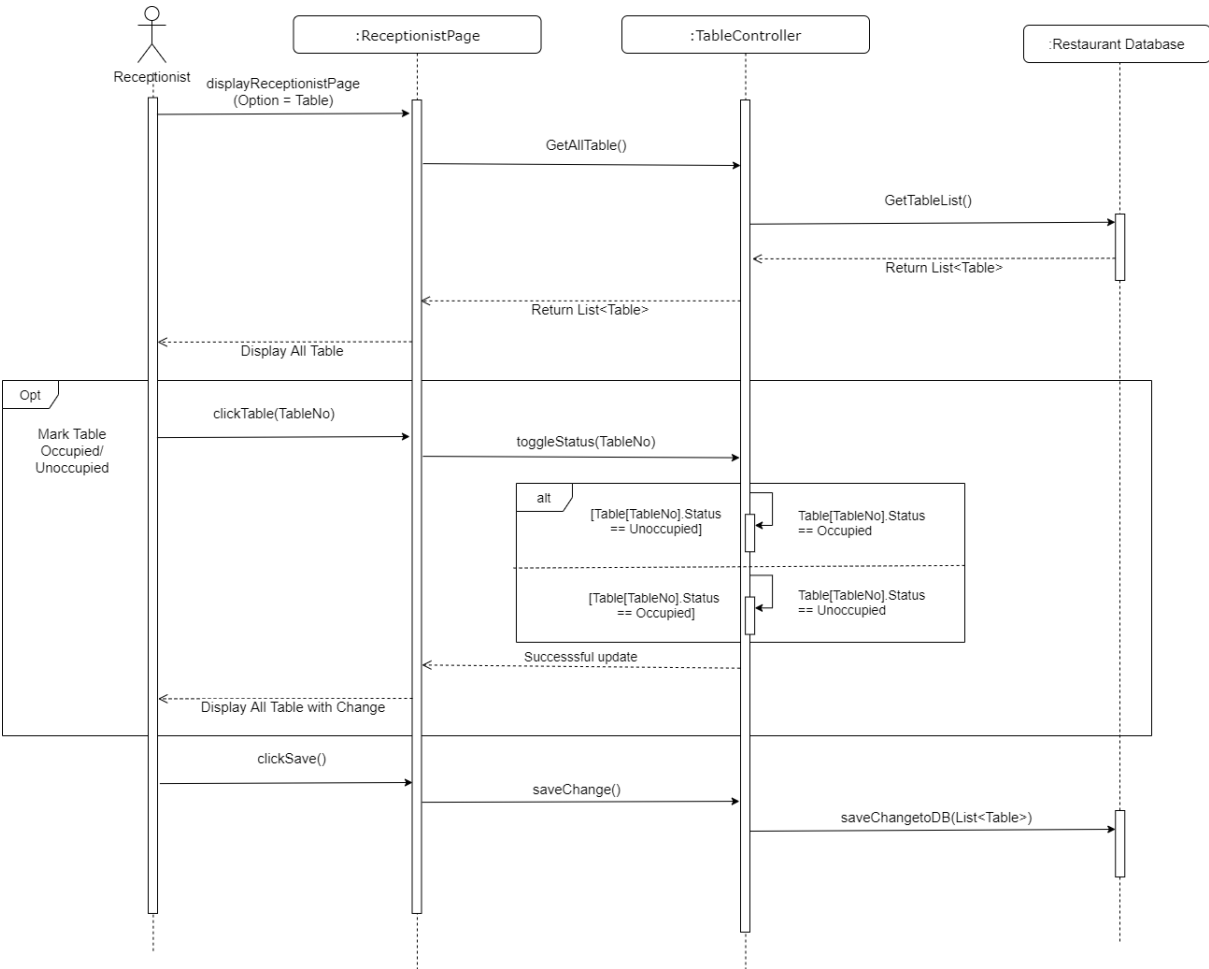
<https://drive.google.com/file/d/14oicHIqsBRyتهiDEQ8RdwLDw424o1XT/view?usp=sharing>

Above is the activity diagram of View Statistics Process

1. The diagram is divided into 2 swimlanes: Restaurant Manager and System
2. When the manager gets access to manager's view the system display the statics on view
3. The manager views statistics and terminates the window, the activity finishes.
4. If the manager chooses to see some specific statistics, the view shows the statistics after being filtered. The flow continues at step 3.

Sequence Diagram

Feature 1: Table Management

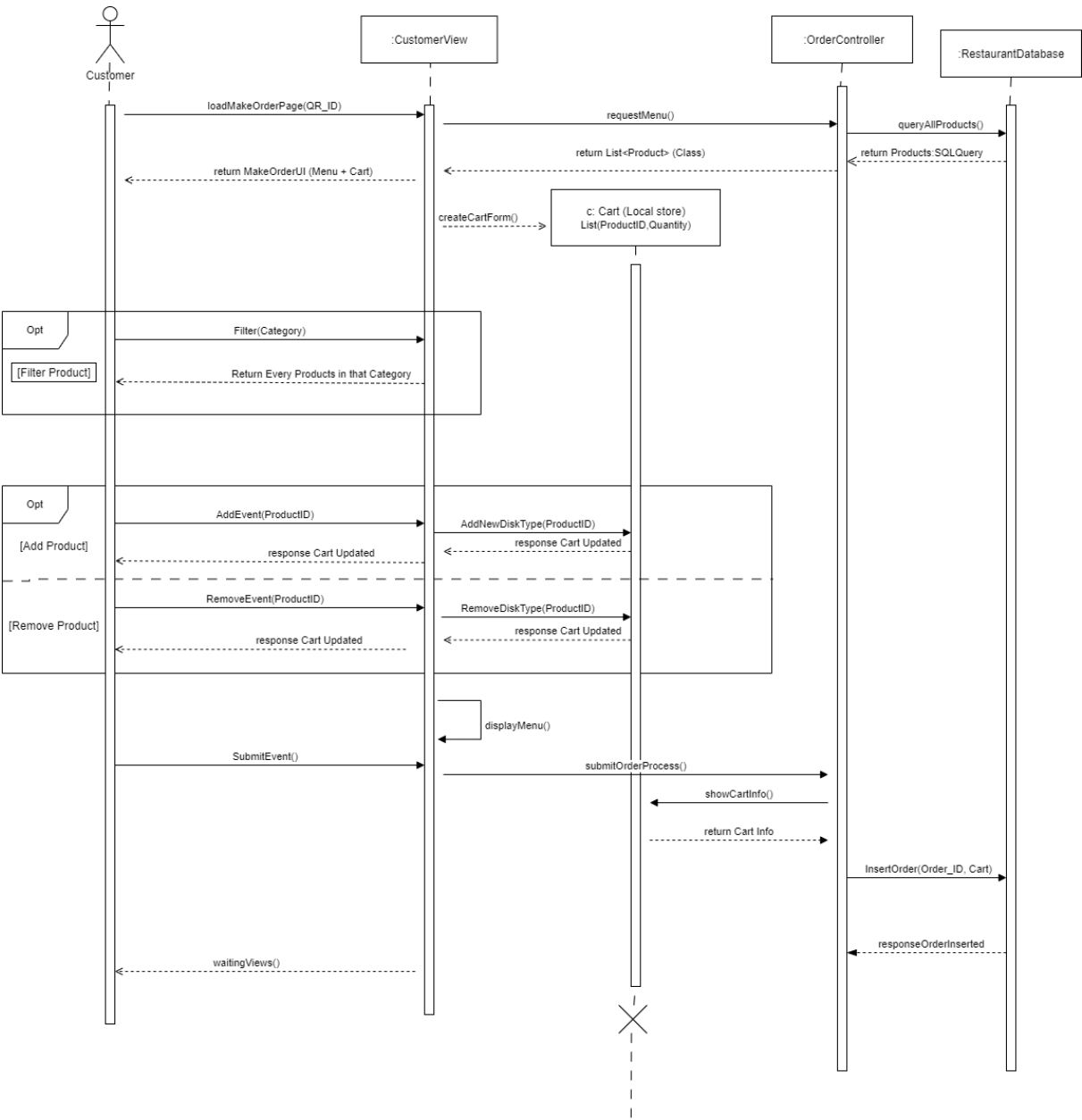


Visit this url to see the diagram better:

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)



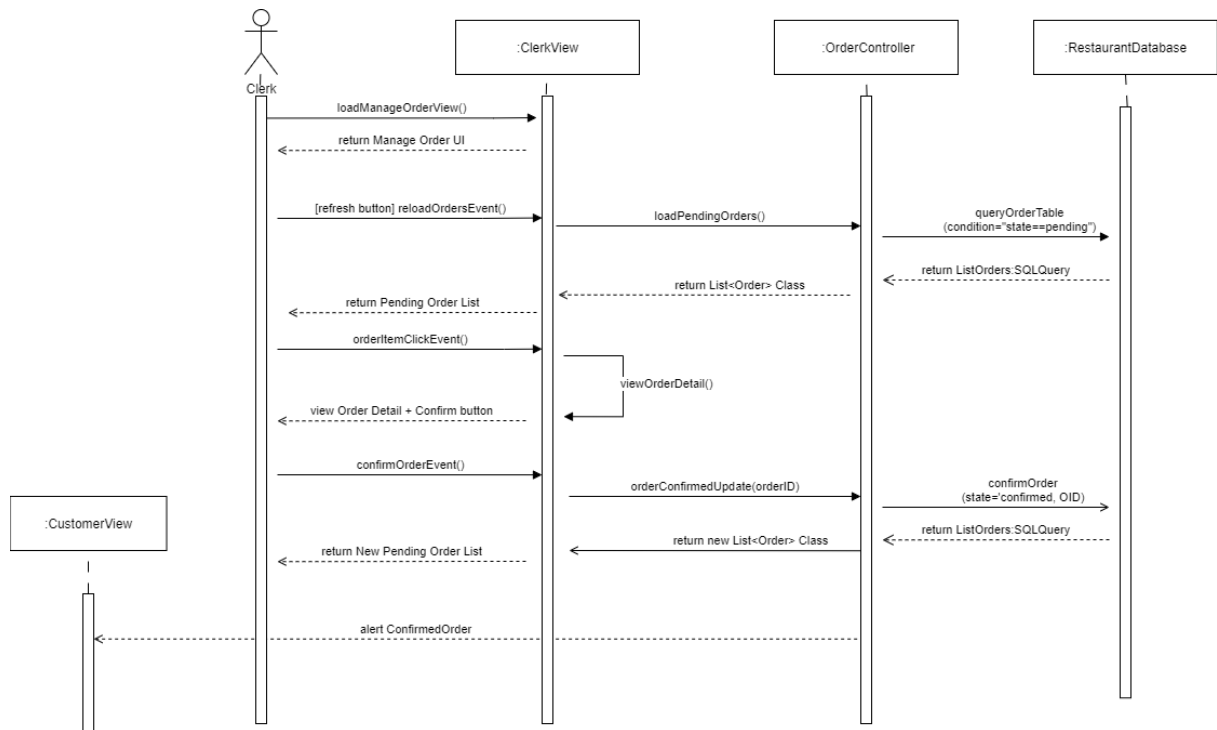
Visit this url, click on tab “Main Make Order” to see the diagram better:

https://drive.google.com/file/d/1c7ADQQkPwCfp_yUE6oIjMJ_xwM7F5mpg/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, filters dishes according to categories, clicks the product to view its details. Customers 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 2: Order System (Manage Order - Clerk View)



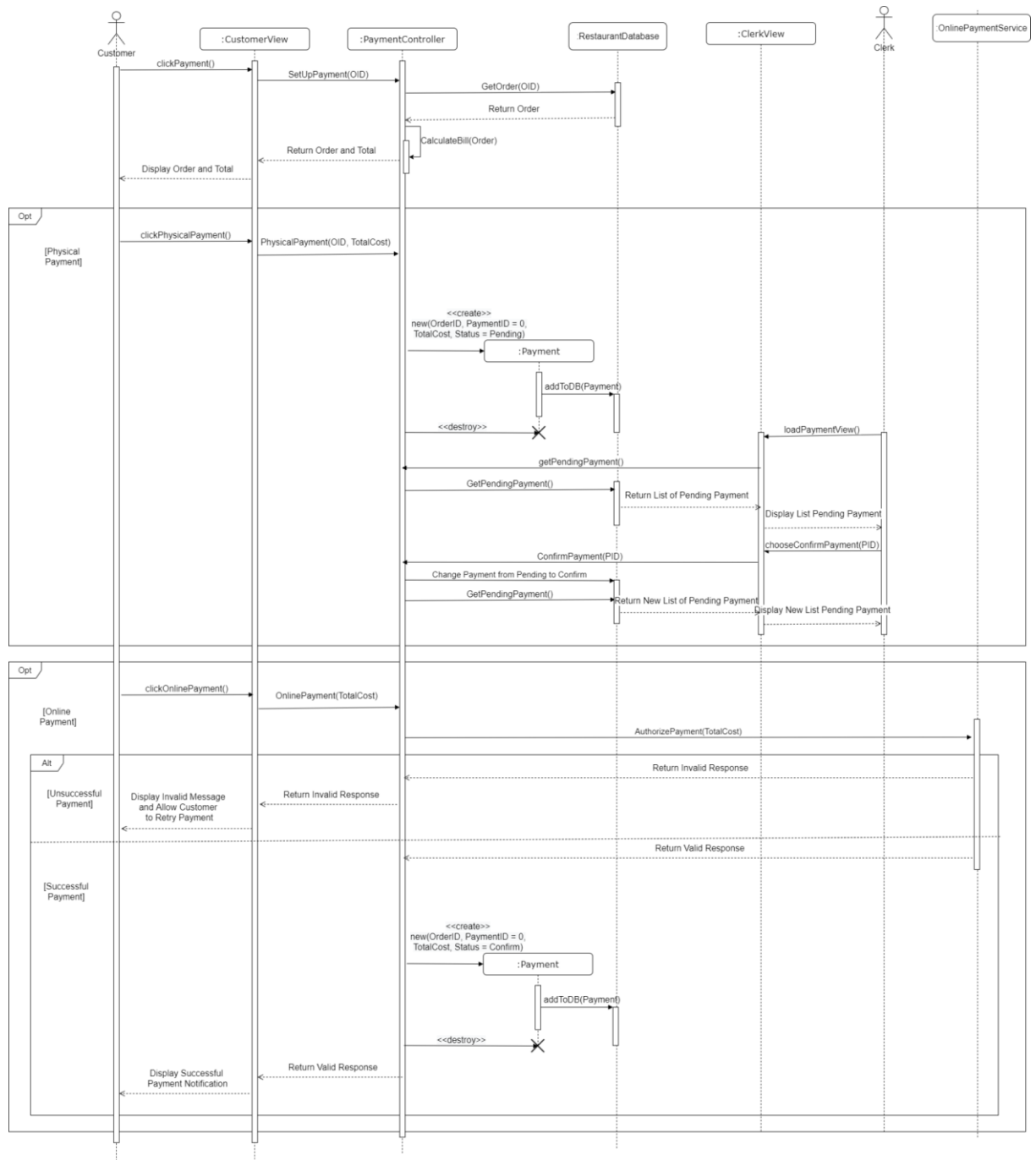
Visit this url, click on tab “Main Manage Order” to see the diagram better:

<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, and the clerk will refresh to get a new list of pending orders.

Feature 3: Making Payment



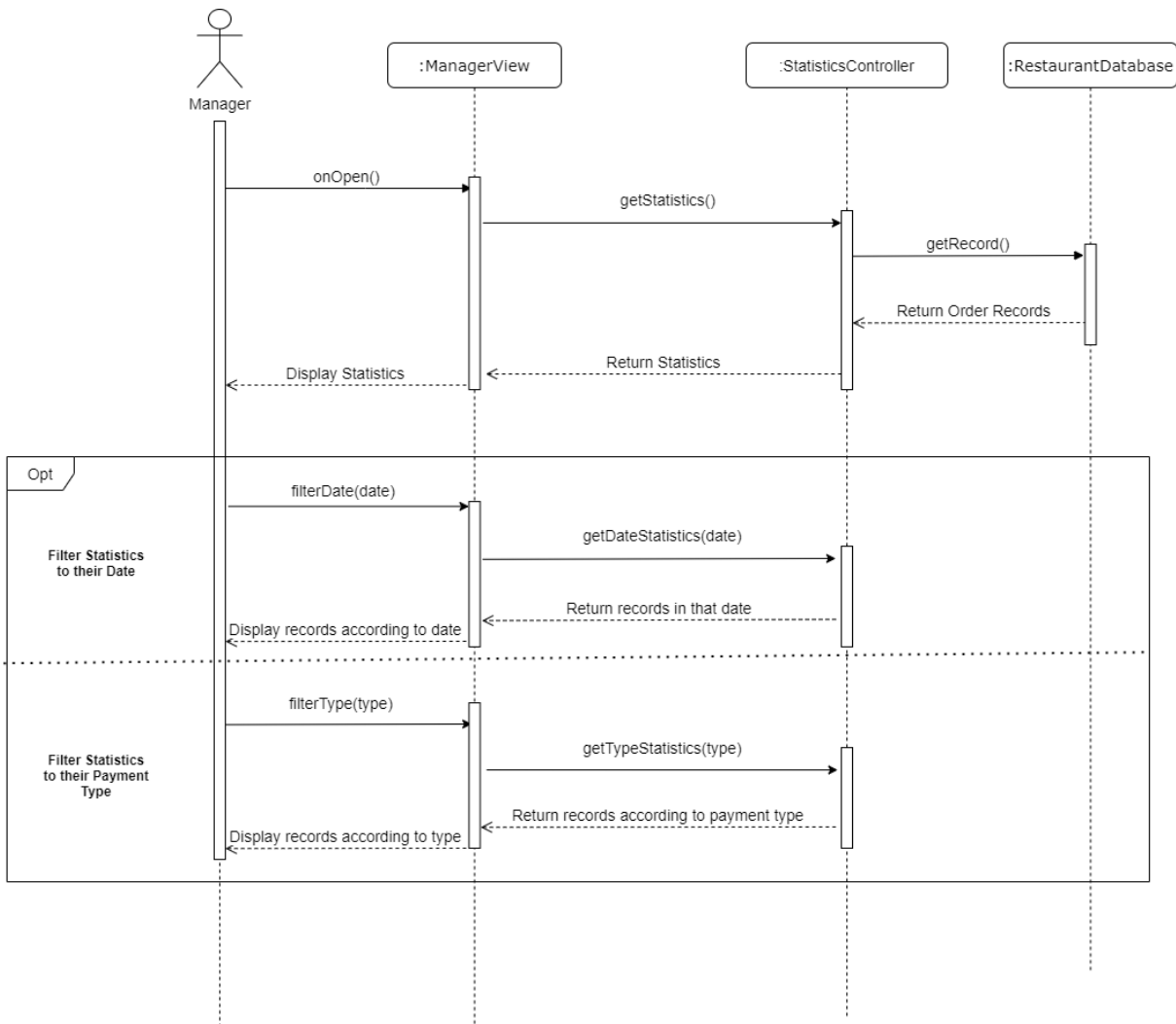
Visit this url to see the diagram better:

<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, “clickOnlinePayment()” 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



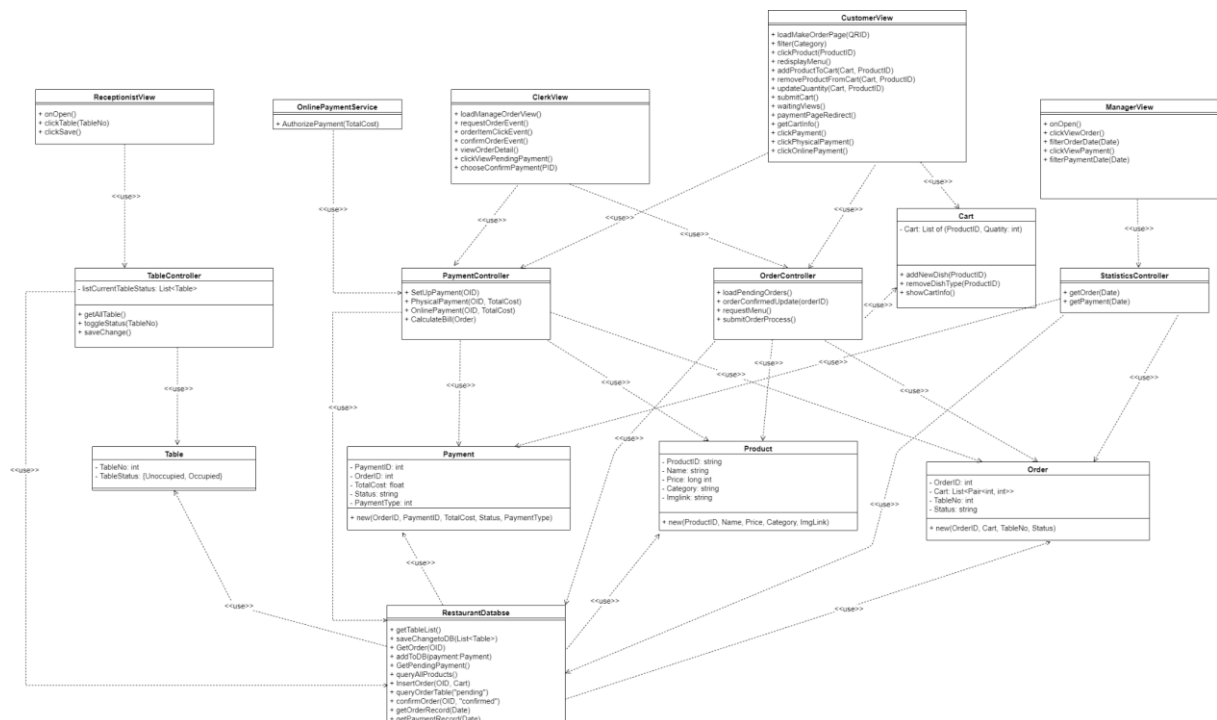
Visit this url to see the diagram better:

<https://drive.google.com/file/d/1RteZkRCepFY0qhjA46RbMybvXh4qLgQ/view?usp=sharing>

1. The manager gets access to the manager view.
2. The system loads the data (`getStatistics()`) to the manager view.
3. The Statistics Controller handles that request by sending a request to the Database.
4. The Database returns statistics to the Statistics Controller and Statistics Controller will send statistics to the manager view.
5. If the manager chooses to filter some statistics (`filterDate(date)`, `filterType(type)`), the filter request is handled by the Statistics Controller and displays the manager view.

Class Diagram

We have drawn the activity diagram representing the flow of the work inside the system. On the other hand, the sequence was designed specifically for the use case "Food Ordering". Now we establish the class diagram in order to have a closer look at how the system is organized. Below is how the class diagram representing the system:



Due to the amount of <<use>> can cause confusion when reading, below is the summary of each class as well as their relationship to each other:

- **ReceptionistView:** Receptionist's system UI, in charge of provide receptionist interface to use the Table Management feature
(Use class TableController)
- **ClerkView:** Clerk's system UI, in charge of provide clerk interface to view pending orders as well as pending payments
(Use class OrderController, PaymentController)

- **CustomerView:** Customer's system UI, in charge of provide customers interface to view restaurant's menu, make order as well as make payment for the order
(Use class OrderController, PaymentController, Cart)
- **ManagerView:** Restaurant's system UI, in charge of provide manager interface to view restaurant statistic, like the restaurant's order and payment history
(Use class StatisticController)
- **OnlinePaymentService:** An interface simulates the Online Payment Service, providing authorization and validation for online payment
(Use class PaymentController)
- **TableController:** A class provides functionality that handle service for Table Management
(Use class Table, Database)
- **PaymentController:** A class provides functionality that handle service for Payment Management
(Use class Table, RestaurantDatabase)
- **OrderController:** A class provides functionality that handle service for Order System
(Use class Order, Product, RestaurantDatabase, Cart)
- **PaymentController:** A class provides functionality that handle service for Making Payment
(Use class Order, Product, Payment, RestaurantDatabase)
- **StatisticController:** A class provides functionality that handle service for Viewing Statistic
(Use class Order, Payment, RestaurantDatabase)
- **RestaurantDatabase:** The class is in charge of query data from restaurant database, passing them as model class as arguments to other classes as well as functioning as an API for other classes to add, remove, update data in database
(Use class Order, Payment, Table, Product)

- Cart: A model class create when Customers are making their order as a way to store orders before add into the database
- Model class such as Table, Payment, Product, Order

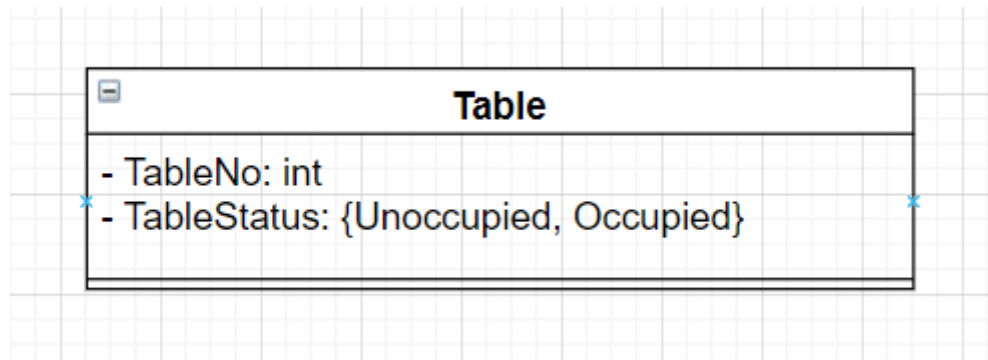
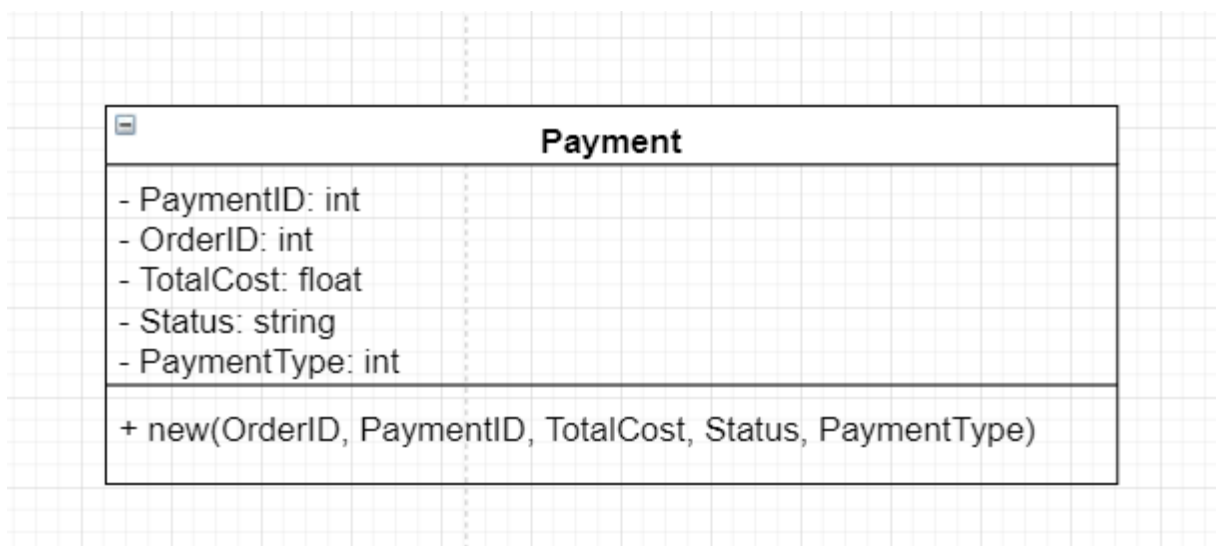
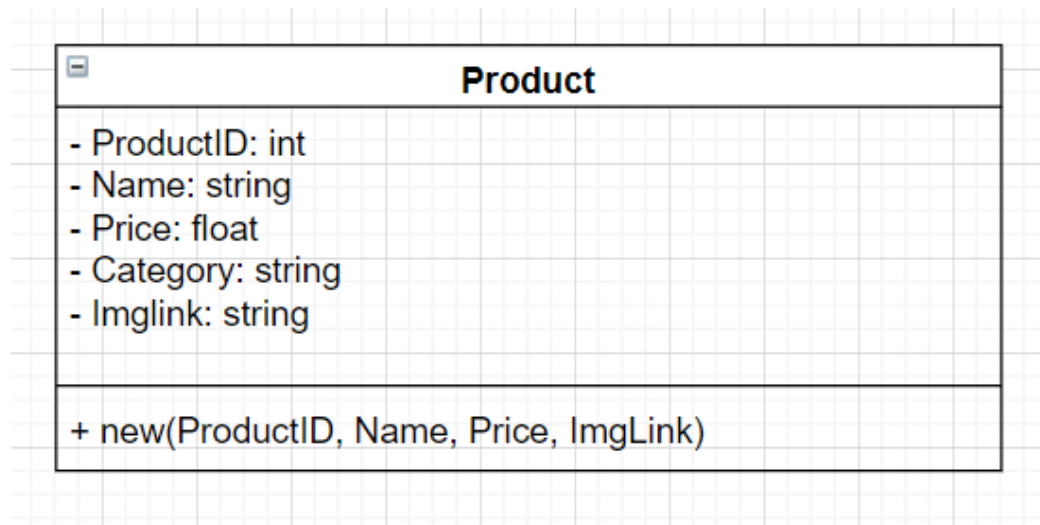


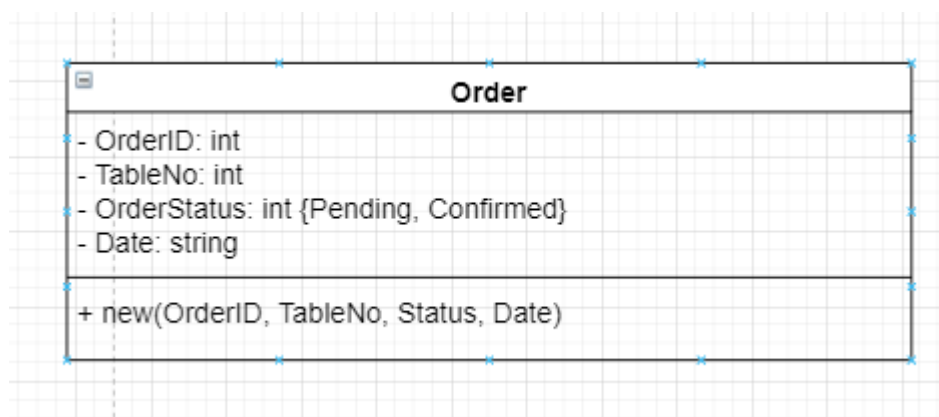
Table Class



Payment Class



Product Class



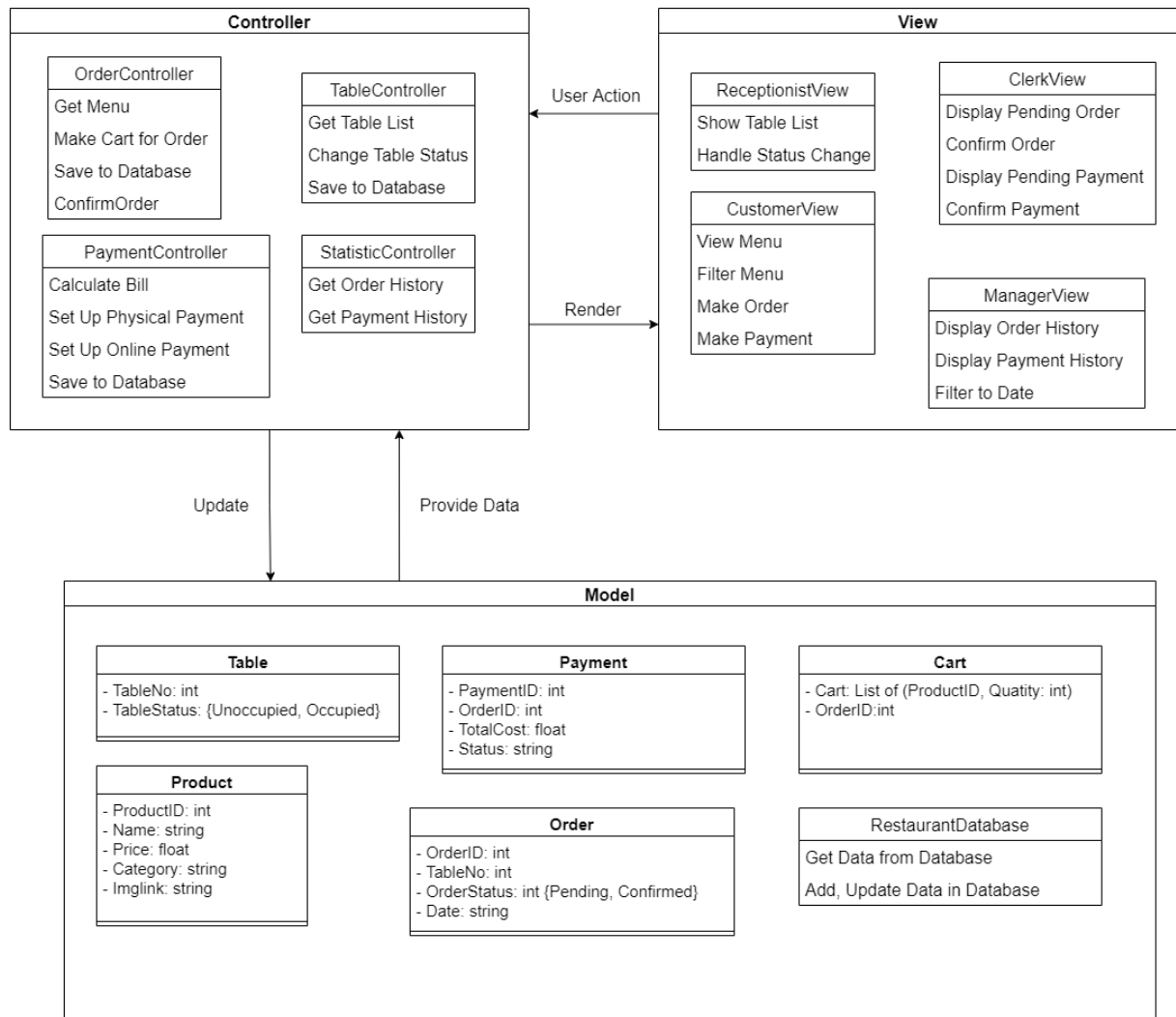
Order Class

Visit this url to see the diagram better:

<https://drive.google.com/file/d/11tW7szjQDztx29I3hLsIQsEodpMTL2ha/view?usp=sharing>

Architecture design

1. General architecture



In designing our application, our group chooses the MVC 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

View: Application has 4 different views:

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

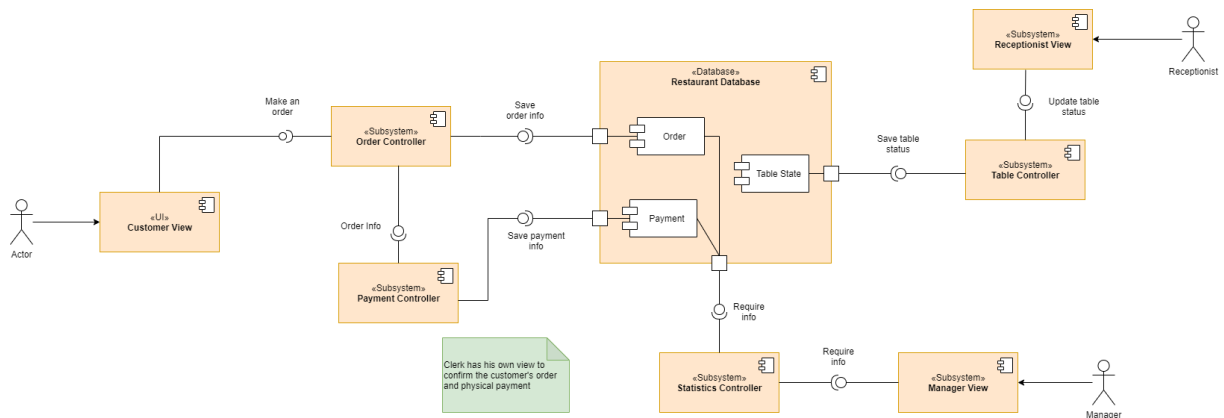
- +) Receptionist View: For the receptionist to use Table Management Feature
- +) Clerk View: For the clerk to receive and confirm pending physical payments and orders
- +) Manager View: For the manager to use View Statistic Feature

Controller: Application has 4 different Controller, each is responsible for each type of model and provide service and data queried from Model to different View

Model: Application has 5 type of Model (Table, Payment, Cart, Product, Order) and a Restaurant Database class to query data from / to Restaurant Database

- Table: this will have a Table Number as key, and Table Status to indicate whether that table is available or occupied
- Payment: this will have a Payment ID as key, as well a Order ID for the Order, the TotalCost for the value of the Payment, Status for Pending and Confirm Status
- Order: this will have a Order ID as key, Table No for which table the order is made, Order Status for Pending and Confirm Status, and Date for the date the order was made
- Product: this will have ProductID as key, the Name of the Product, Price of the Product, Category for which this product belongs to and ImgLink for Image of the Product
- Cart: this will have both ProductID and OrderID to store which Order have which Product with which quantity

2. Component Diagram



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

<https://drive.google.com/file/d/1t6Wo7eF9a5hLe2RV77ZTv24LiT3Dh821/view?usp=sharing>

Description of the component diagram:

The Order Controller and Payment Controller process order requests and payment requests, respectively, from the customer view.

The Table Controller manages the update table status from the receptionist view.

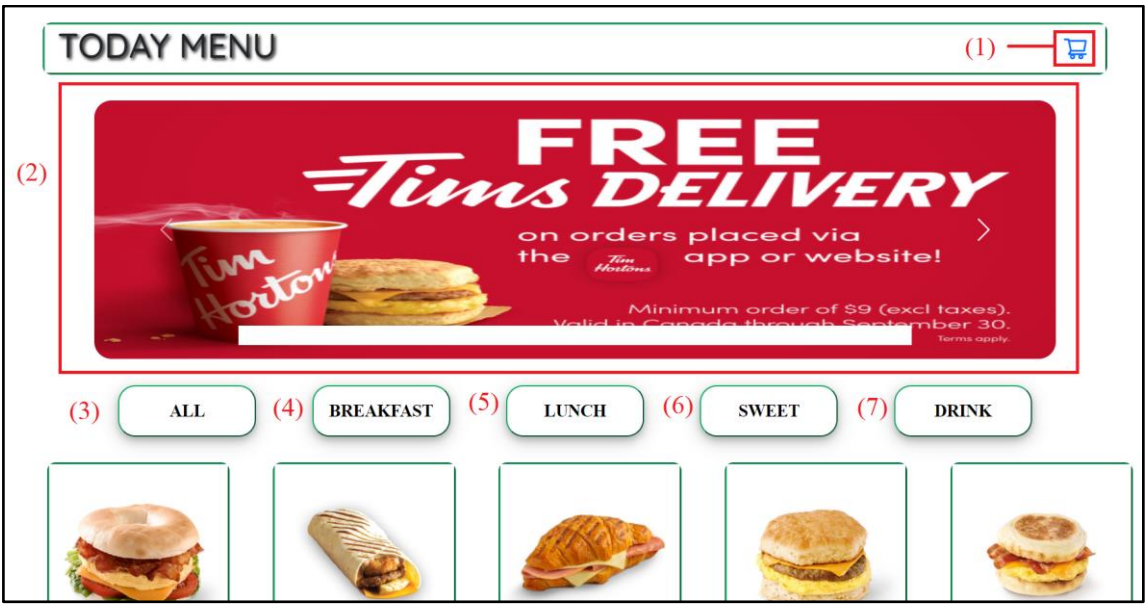
The Manager Controller manages the statistics request from the manager view.

The Database component contains order, payment and table status information.

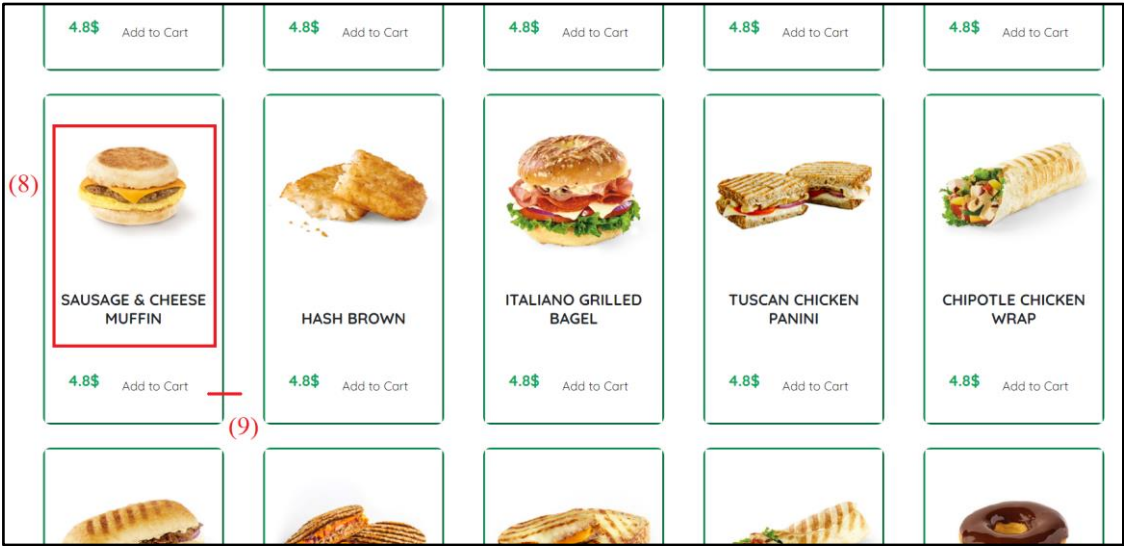
(Note that the Restaurant Clerk has his own view too, the role of Restaurant Clerk is to confirm order and physical payment which is included in the Order and Payment Component.)

User Interface and Screen of flow

Customer View



UI 1.1: Customer View start with Menu Page (1)











UI 1.2: Customer View start with Menu Page (2)

MY ORDER

BACK

Table No: 1

Food Names	Quantities	Price Per Unit
 HAM AND CHEESE CROSSIANT	- 2 +	4.80\$ 
 SAUSAGE EGG & CHEESE MUFFIN	- 1 +	4.80\$ 
 TUSCAN CHICKEN PANINI	- 1 +	4.80\$ 
 ITALIANO GRILLED BAGEL	- 1 +	4.80\$ 
Total Cost:	5	24.00\$

SUBMIT ORDER

UI 1.3: Customer View of Their Order

PAYMENT

Date: 11/7/2021

Time: 7:33:41 PM

Table No: 1

Total Cost: 38.40 \$

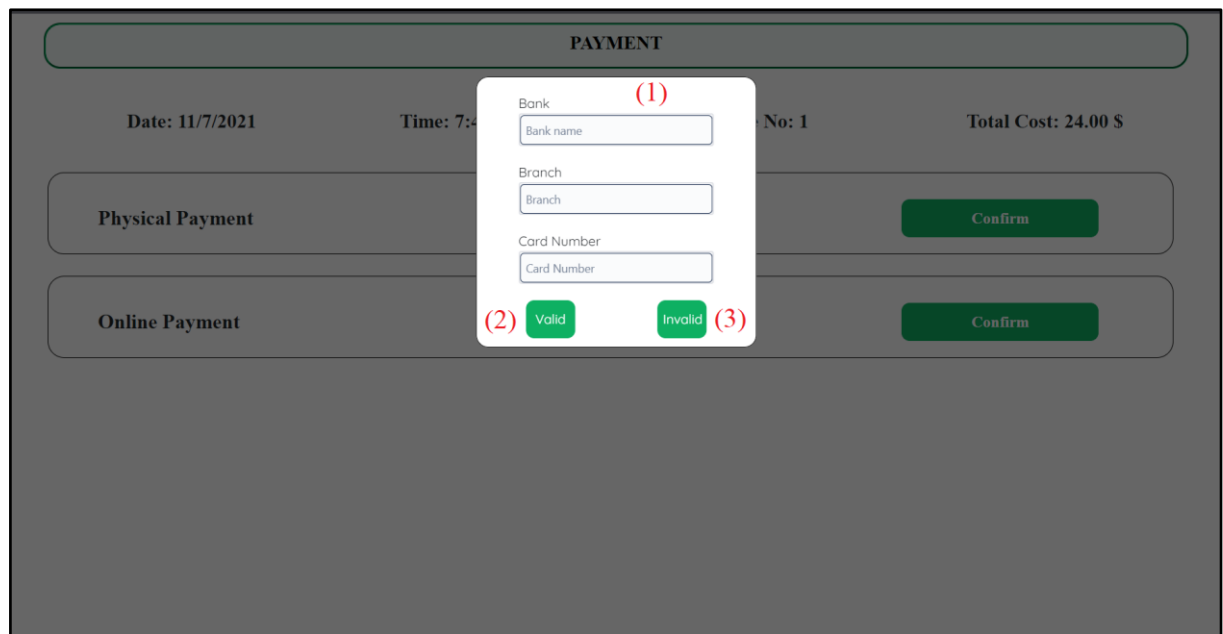
Physical Payment

Confirm

Online Payment

Confirm

UI 1.4: Customer View with Making Payment Feature



PAYMENT

Date: 11/7/2021 Time: 7:32:51 PM Table No: 1 Total Cost: 24.00 \$

Physical Payment Confirm

Online Payment (2) Valid Invalid (3) Confirm

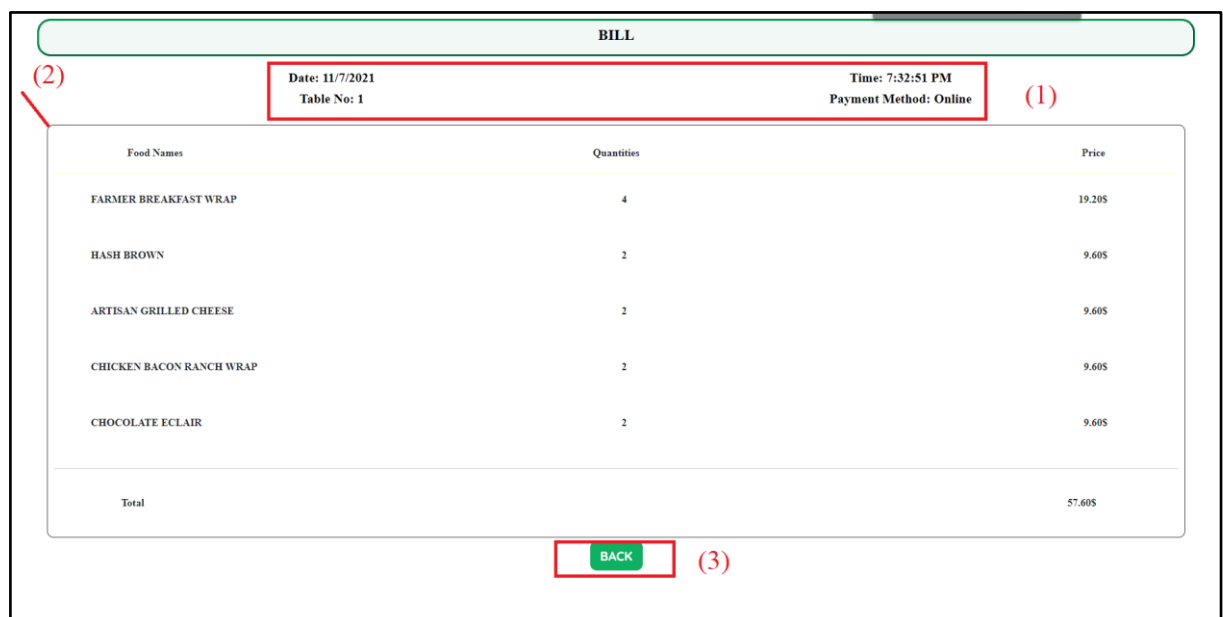
Modal (1):

Bank
Bank name

Branch
Branch

Card Number
Card Number

UI 1.5: Online Payment simulation with Valid and Invalid Option



BILL

(2) Date: 11/7/2021 Time: 7:32:51 PM Table No: 1 Payment Method: Online (1)

Food Names	Quantities	Price
FARMER BREAKFAST WRAP	4	19.20\$
HASH BROWN	2	9.60\$
ARTISAN GRILLED CHEESE	2	9.60\$
CHICKEN BACON RANCH WRAP	2	9.60\$
CHOCOLATE ECLAIR	2	9.60\$
Total		57.60\$

BACK (3)

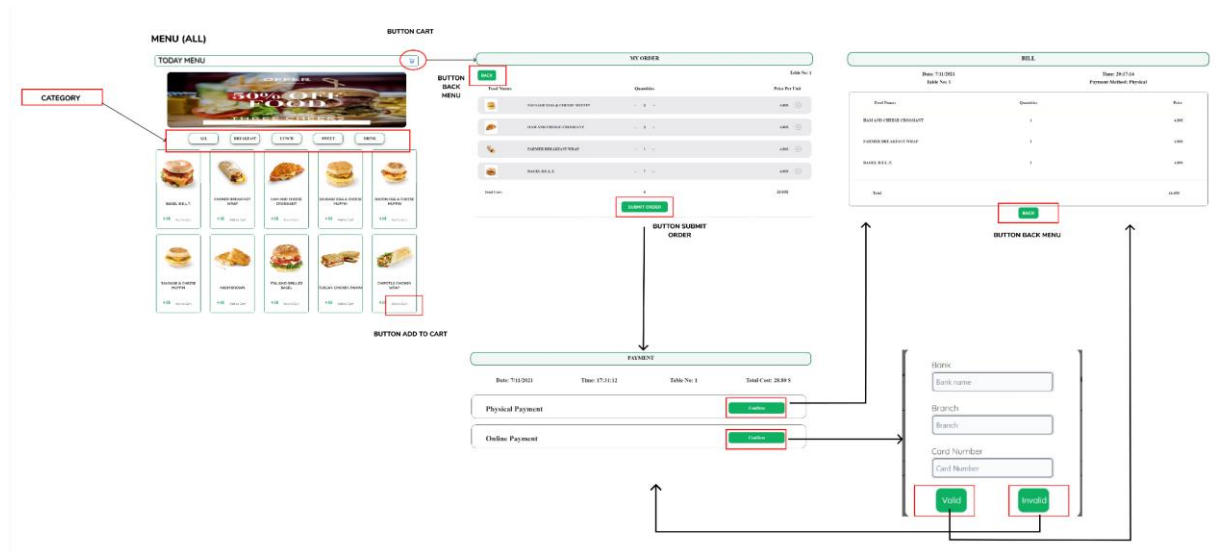
UI 1.6: System displays the bill for the Payment/ Order

Customer UI for Make Order/ Make Payment Description:

- **UI 1.1:** After scanning the QR code placed on the table, the customer device is presented with the restaurant's Menu Page (UI 1.1) viewing all available food and a banner slide for special deal promotion (2). Below the banner slide is the categories filter bar with multiple for different food types: All (3), Breakfast (4), Lunch (5), Sweet (6), Drink(7), clicking on each category will filter and show only the list of products that belong to that category.
- **UI 1.2:** Below the navigation bar is the menu itself viewing multiple cards for each product (UI 1.2). Each product card contains an image illustrating the food (8), the price and a button to add the product to cart (9). Customers can browse and add the desired food to the local cart store on their device. After being satisfied with their order, customers can click on the cart button placed on the top-right side of the page (1) to route to the View Order page (UI 1.3).
- **UI 1.3:** The view order page shows the table number (2) and customer current order state (3). Customers can either modify the quantity of the same product that they want (4) or remove product from the cart (5). Customers can click on the back button (1) to continue browsing the menu. Finally, customers will click on the Submit Order button (6) to finalize the order and routing to payment page (UI 1.4) and start the payment process.
- **UI 1.4:** In the Payment page (UI 1.4), there are some fields that show information about customer's detail (1). There is a Date which is the date of the customer's order. Another one is Time which is the time that a customer makes his/her order. There is also a Table No to show the table's id that the customer is using. And the last one is the total cost which shows the amount of money that the customer must pay.
- The customer can choose two options. Physical method or Online payment method to pay his/her bill. If the customer chooses Physical payment, he/she can press the confirm button (2) in the physical payment box, and the customer

- will send their payment request to the Clerk. If the customer chooses Online payment, they press the confirm button (3) in the online payment box, and the customer will receive a window that shows in the screen (UI 1.5).
- **UI 1.5:** The customer needs to fill in the blank including 3 fields Bank, Branch and Card number (1) to connect to his/her bank account. After filling the information, the Valid and Invalid Option is a simulation of the Online Payment Service, when the customer clicks on Valid, System registers the Payment as Valid and move to the Bill, while clicking on Invalid, System registers the Payment as Invalid and return customers back to the Payment Page, where Customer should choose a different payment method.
 - **UI 1.6:** In Bill page (UI 1.6), there are four information (1) provided below the Bill header. They are Date, Time, Table No, Payment Method. Date, Time and Table No are similar to Payment page, Payment Method is the method that customer chooses in Payment page. Below the information field is the summary table (2). The table provides several information about the order, these are Product Name , Quantities of each type Product, Price of each type Product and Total Payment. At the bottom of the page is Back button (3), which will return to the Menu page and reset all information in the cart when a customer presses it, used for making new orders.

Screen flow of Customer View (From View Menu, Make Order to Make Payment)



Visit this url to see the image better:

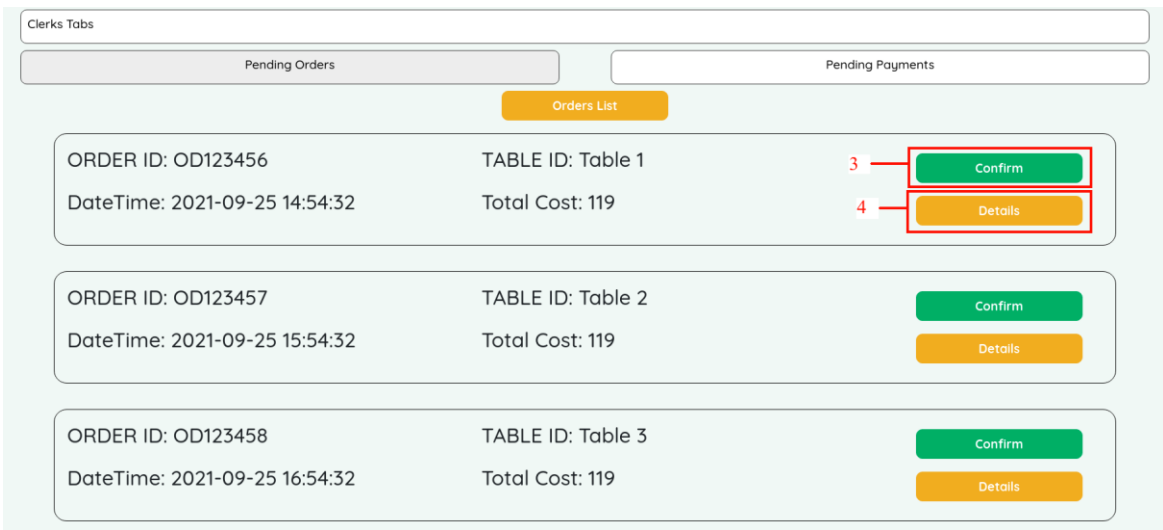
<https://drive.google.com/file/d/1h6qpXZUA6q0SNygHmns-KkHmQOn1zDQj/view?usp=sharing>

- Customers start as Menu Page (UI 1.1), can add Product to their Order and moves to My Cart Page by the Cart Button
- In My Cart Page, customer can go back to Menu Page through the Back Button, can change Quantity of current in-cart Product or clicks Submit Order for Clerk to Confirm, after their Order has been confirmed, customers go to Payment Page
- In Payment Page, customers chooses between Physical Payment and Online Payment, if they choose Online Payment, a pop up will appear, require input information, click the Valid Option simulated the Online Payment Service validates the Payment and customer moves to Bill Page, otherwise, they remain in Payment Page to choose a different payment option
- In the Bill Page, customers can go back to the Menu Page to make a new Payment

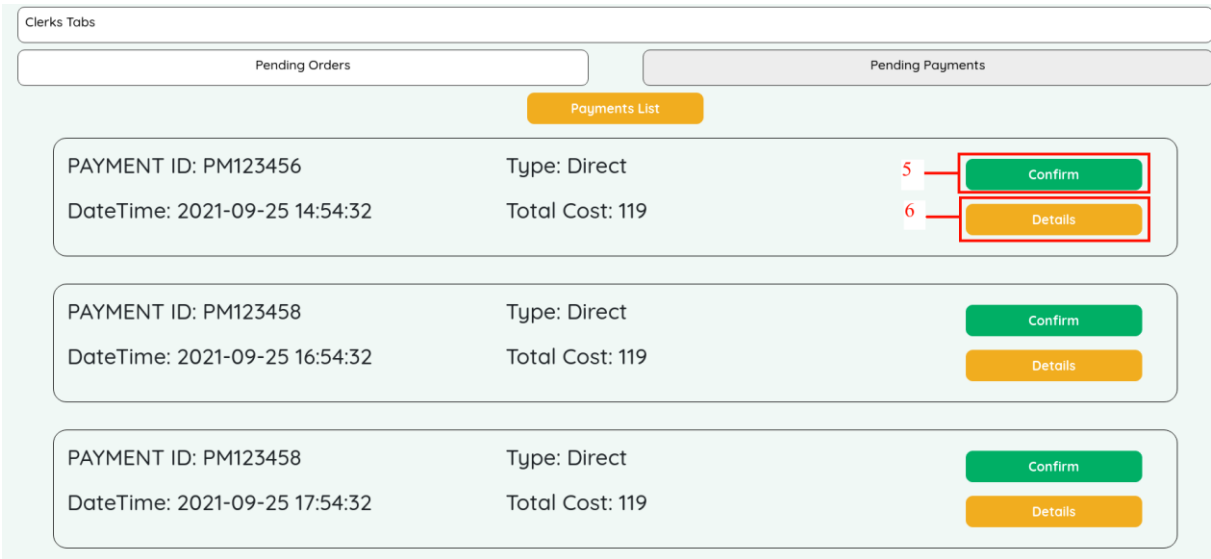
Clerk View



UI 2.1: Clerk View Load Page



UI 2.2: Clerk View Orders List



UI 2.3: Clerk View Payments List

Clerks Tabs

Pending Orders

Pending Payments





Order Detail

TABLE ID: 1

ORDER ID: OD123456

Date Time: 2021-09-25 14:54:32

Cart Info

Food Names	Quantities	Price Per Unit
 Spicy meatballs	1	19.99\$
 Spicy meatballs	1	19.99\$
 Spicy meatballs	4	19.99\$
 Spicy meatballs	1	19.99\$
Total Cost:		139.93\$

7

Confirm Customer Order

UI 2.4: Clerk View Order Detail

Clerks Tabs

Pending Orders

Pending Payments

Payment Detail

TABLE ID: 1

PAYMENT ID: PM123456

Date Time: 2021-09-25 14:54:32

Payment Info

Food Names	Quantities	Price
Spicy meatballs	1	19.99\$
Spicy meatballs	1	19.99\$
Spicy meatballs	4	79.96\$
Spicy meatballs	1	19.99\$
Total		139.93\$

8

Confirm Customer Payment

UI 2.5: Clerk View Payment Detail

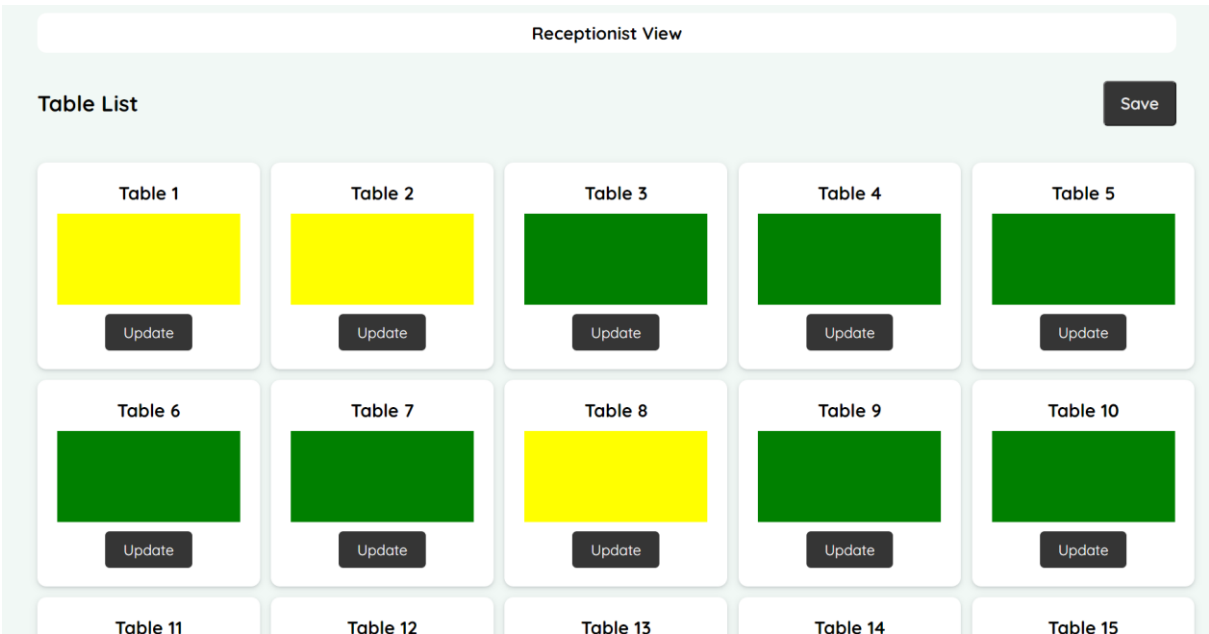
Clerk Manage Orders/Payments's UI Description:

- **UI 2.1:** When starting up the view on the browser, the clerk is presented only with the navigation bar (UI 2.1) with two options, click on Pending Orders

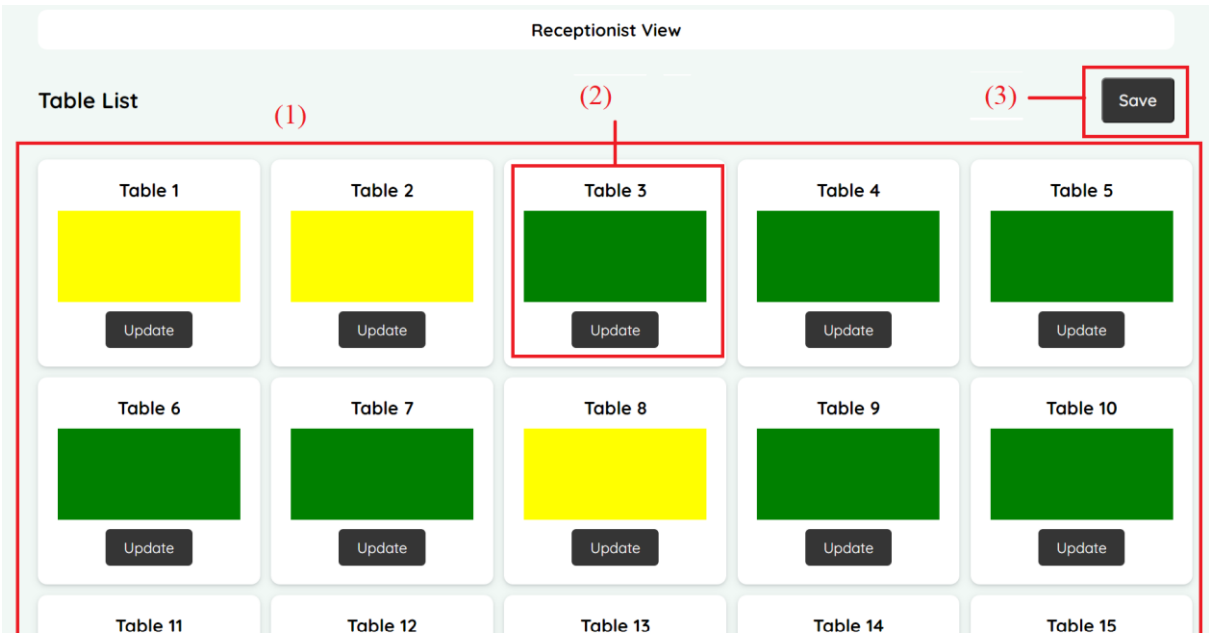
navigation link (1) views list of pending orders waiting to be confirmed (UI 2.2), the same with Pending Payments navigation link (2) views list of pending payments (UI 2.3).

- **UI 2.2 + UI 2.3:** On the Orders List page (UI 2.2), there are couples of items presented for each pending order, clicking on the confirm button (3) will send the accept update to backend and the item will disappear from the screen and clicking on detail button (4) will route to the Order Detail Page (UI 2.4). The same with Payments List Page (UI 2.3), clicking confirm button (5) to accept the payment and clicking detail button (6) to view Payment Detail (UI 2.5).
- **UI 2.4:** On the Order Detail page (UI 2.4) the details about the list of products ordered by the customer are presented with the prices and the total cost of the order, clicking the 'Confirm Customer Order' (7) will send an update request to backend. Clerk now will be routed back to the Orders List page (UI 2.2) and the confirm order is disappeared.
- **UI 2.5:** On the Payment Detail page (UI 2.5) the bill of the direct payment is presented. After a customer pays the bill, the clerk will click the 'Confirm Customer Payment' to confirm payment and will be routed back to the Payments List page (UI 2.3) and the confirmed payment will disappear.

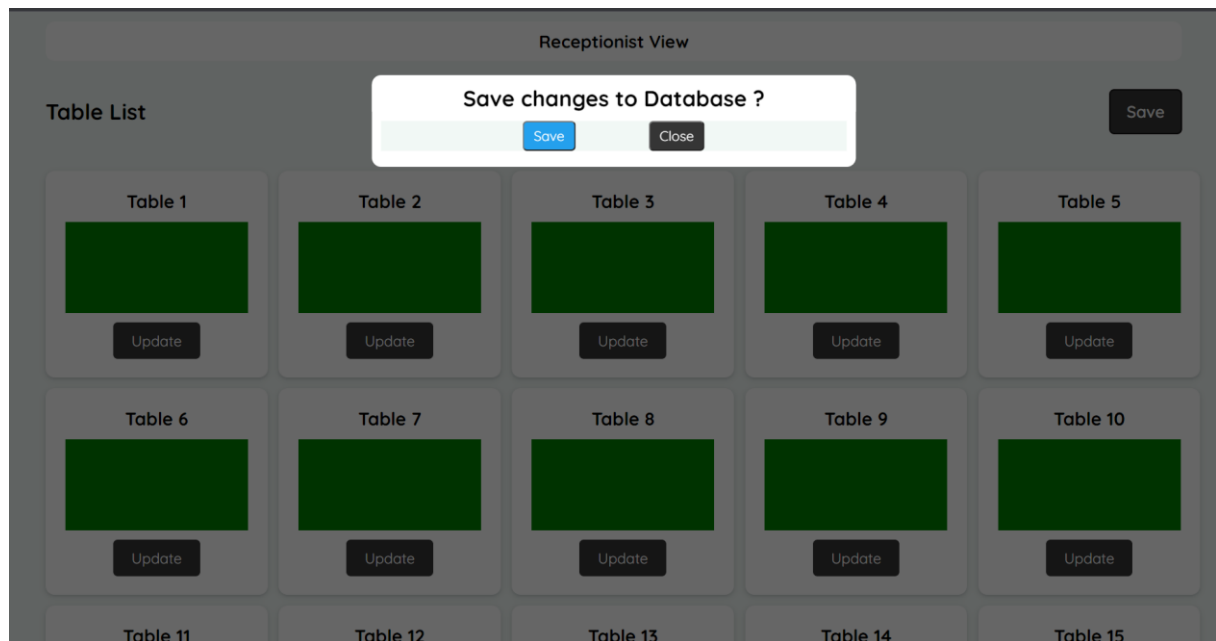
Receptionist View



UI 3.1: Receptionist view with Feature Table Management



UI 3.2: Table Management annotation



UI 3.3: Save Changes to Database popup

Feature Table Management's UI Description:

- **UI 3.1:** When start up the page, the System will execute the procedure to get 25 table's status from the Restaurant Database and display them to the Receptionist
- **UI 3.2:** Each table is displayed as each card (Object 1) in a grid, inside each card contain table number, table status represented by a colored box (green as available and yellow as occupied) and a "Update" button to change its status
- When receptionist presses "Update" button in a selected table, the colored box toggles its (from yellow to green and vice versa) and the system saves that change to its local storage
- **UI 3.3:** hen receptionist has finished their update, the receptionist clicks the Save Button (2) , which pops up a modal asking to confirm the choice and if confirm, the changes will be updated to the Restaurant Database

Manager View

Restaurant Statistic

Payment Type Filter

All

Date Filter

☐ Filter Both

Filter

ORDER ID	Table	Payment	Total (USD)	Date
1	1	online	45	10/10/2021
2	10	online	145	10/1/2021
3	4	physical	90	10/10/2021
4	5	online	545	1/3/2021
5	7	physical	68	7/5/2021
6	1	online	45	10/10/2021

1

2

UI 4.1: Manager view screen

Restaurant Statistic

Payment Type Filter

All

Date Filter

☐ Filter Both

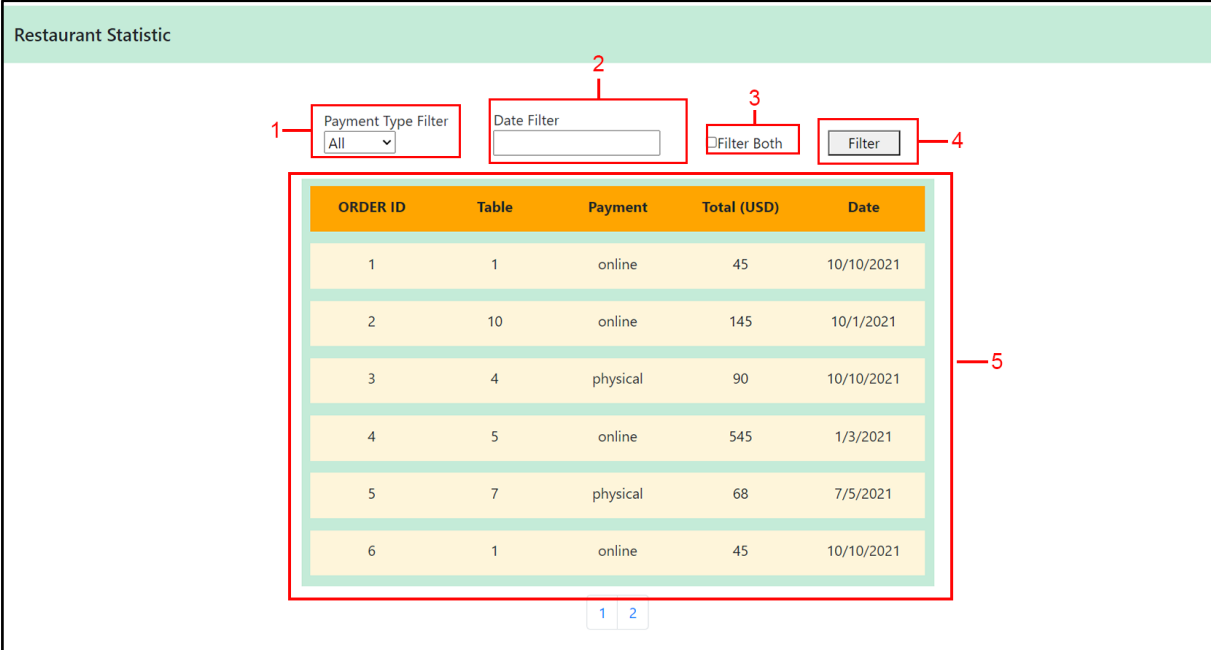
Filter

ORDER ID	Table	Payment	Total (USD)	Date															
<div><div>Thông tin chi tiết</div><div><div>Close</div><table><thead><tr><th>Product</th><th>Quantity</th><th>Total</th></tr></thead><tbody><tr><td>Pizza America</td><td>2</td><td>190</td></tr><tr><td>Chines Noodle</td><td>4</td><td>70</td></tr><tr><td>Cocacola</td><td>1</td><td>40</td></tr><tr><td>White Coffee</td><td>3</td><td>50</td></tr></tbody></table></div></div>					Product	Quantity	Total	Pizza America	2	190	Chines Noodle	4	70	Cocacola	1	40	White Coffee	3	50
Product	Quantity	Total																	
Pizza America	2	190																	
Chines Noodle	4	70																	
Cocacola	1	40																	
White Coffee	3	50																	
4	5	online	545	1/3/2021															
5	7	physical	68	7/5/2021															
6	1	online	45	10/10/2021															

1

2

UI 4.2: Pop-up screen



The screenshot shows a web interface titled "Restaurant Statistic". It features a filter section at the top with a "Payment Type Filter" dropdown (labeled 1) set to "All", a "Date Filter" input field (labeled 2), a "Filter Both" checkbox (labeled 3), and a "Filter" button (labeled 4). Below the filters is a data table (labeled 5) with the following data:

ORDER ID	Table	Payment	Total (USD)	Date
1	1	online	45	10/10/2021
2	10	online	145	10/1/2021
3	4	physical	90	10/10/2021
4	5	online	545	1/3/2021
5	7	physical	68	7/5/2021
6	1	online	45	10/10/2021

At the bottom of the table are two pagination buttons labeled "1" and "2".

UI 4.3: Manager view annotation

Feature View Statistic's UI Description:

- When start up the page, the System will execute the procedure to get all order and payment record from the Restaurant Database and display them to the Manager in pages
- Each Record is displayed as each row in the Data Table (5), inside row contains Order ID, Table of the Order, Payment Type, Total Bill and Date
- When Manager clicks a row in the data table, system displays a modal showing the order detail about that order/ payment
- Manager can filter rows by date by choosing a day in the Date Picker (2) and presses the filter button (4), payment method by using Payment Type Filter (1) and presses the filter button (4) or both by choosing input for (1) and (2), check the filter both box (3) and presses the filter button (4)

GitHub repository

For further detail, please visit the work at our GitHub repository

<https://github.com/remsoakawaii1/CNPM>