

VIETNAM NATIONAL UNIVERSITY, HO CHI MINH CITY
UNIVERSITY OF TECHNOLOGY
FACULTY OF COMPUTER SCIENCE AND ENGINEERING



Software Engineering (CO3001)

ASSIGNMENT REPORT

Lecturer: Assoc. Prof. Quản Thành Thơ

Class : CC01

Students: Lê Đức Cầm - 1952588
 Cao Bá Huy - 1952713
 Ngô Minh Đại - 1913008
 Huỳnh Phước Thiện - 1952463

HO CHI MINH CITY, SEPTEMBER 2021



Changelog

No.	Date	Changes	Student
20	22/10/2021	Update the description about patterns	Cao Bá Huy
19	21/10/2021	Update the MVC model	Cao Bá Huy
18	19/10/2021	Add the MVC model	Lê Đức Cầm
17	18/10/2021	Add the comparison between 2 patterns	Lê Đức Cầm
16	17/10/2021	Add the description of architectural pattern	Lê Đức Cầm
15	16/10/2021	Update the implementation diagram	Huỳnh Phước Thiện
14	16/10/2021	Add the implementation diagram	Ngô Minh Đại
13	7/10/2021	Reformat the Class diagram	Cao Bá Huy
13	5/10/2021	Add some features to Activity diagram	Cao Bá Huy
12	4/10/2021	Add task 2.1 (Activity diagram)	Lê Đức Cầm
11	3/10/2021	Add task 2.3 (Class diagram)	Ngô Minh Đại
10	3/10/2021	Update the sequence diagram	Cao Bá Huy
9	1/10/2021	Add task 2.2 (Sequence diagram)	Huỳnh Phước Thiện
8	25/9/2021	Final checking	Cao Bá Huy
7	18/9/2021	Finish the use case description table	Huỳnh Phước Thiện
6	12/9/2021	Provide use-case diagram for the food order feature	Huỳnh Phước Thiện
5	6/9/2021	Provide use-case diagram for whole system	Cao Bá Huy , Lê Đức Cầm
4	3/9/2021	Finish the functional and non-functional requirements of the desired system	Lê Đức Cầm
3	2/9/2021	Finish the "Expectation" and "Scope of project" sections	Ngô Minh Đại
2	30/8/2021	Finish introduction, "Project context" and "Stakeholders of the project" sections added	Ngô Minh Đại
1	26/8/2021	Initialize the report file, Include template, add table content	Cao Bá Huy



Work assignment

Task	Description	Student
1.1	Identify the context of this project. Who are relevant stakeholders? What are expected to be done? What are the scope of the project?	Ngô Minh Đại
1.2	Describe all functional and non-functional requirements of the desired system. Draw a use-case diagram for the whole system	Lê Đức Cầm , Cao Bá Huy
1.3	Choose one specific feature, i.e. food ordering, table reservation, customer management. Draw its use-case diagram and describe the use-case using a table format	Huỳnh Phước Thiện
2.1	Draw an activity diagram to capture Major (not all) functional requirements of the desired system.	Lê Đức Cầm
2.2	Draw a sequence diagram for use-case in Task 1.3.	Huỳnh Phước Thiện
2.3	Draw a class diagram.	Ngô Minh Đại
2.4	Formatting and checking.	Cao Bá Huy
3.1	Describe an architectural approach that is used to implement the desired system	Cao Bá Huy , Lê Đức Cầm
3.2	Draw an implementation diagram for major functional requirements	Ngô Minh Đại, Huỳnh Phước Thiện

Contents

1	Introduction	4
2	Task 1.1	4
2.1	Project Context	4
2.2	The stakeholders of the project	4
2.2.1	Internal stakeholders	4
2.2.2	External stakeholders	5
2.3	The expectation	5
2.3.1	General Objectives	5
2.3.2	Specific Objectives	5
2.4	The scope of the project	5
2.4.1	Project objective	6
2.4.2	The deliverables	6
2.4.3	Technical requirements	6
2.4.4	Limits and exclusions	6
3	Task 1.2	6
3.1	Functional and non-functional requirements of desired system	6
3.1.1	Functional requirements for each actor:	6
3.1.2	Non-functional requirements for each actor:	8
3.2	Use Case Diagram for the desired system	9
4	Task 1.3	10
4.1	Use case diagram for food ordering feature	10
4.2	Use Case Description for food ordering feature	11
5	Task 2.1 : Activity diagram	12
6	Task 2.2 : Sequence diagram	13
7	Task 2.3 : Class diagram	14
8	Task 3.1 : Architecture pattern	15
8.1	Introduction about Model-View-Controller (MVC) and Microservices architecture pattern	15
8.2	Comparison	17
8.3	Why we choose these patterns	17
8.4	MVC diagram	19
9	Task 3.2 : Implementation diagram	20

1 Introduction

In the new era of electronic devices, it is really easy to do almost everything in our daily life when we apply the advances in the information technology field, especially in terms of software application. An inevitable and also important step in developing such a kind of software is the designing phase, so we need tools that help us do those works and use-case diagrams are introduced as one of the most effective tools to modelize and visualize our system. Our work below will consist of gathering and analyzing the context, the requirement in a specific situation then apply use-case diagram and description table to show the insights of our developing system.

2 Task 1.1

2.1 Project Context

Nowadays people live a completely different life compared with years ago since the arrival of technology. Technology is one of the main ways we change the world. Whenever people meet up together for several hours, they're going to require food, and beverages. This (Online Restaurant Management System) restaurant caters for parties like Birthday, Wedding, Anniversaries, Christening and Christmas parties. "The ultimate promise of technology is to make a world that we command by the push of a button."(Volker Grassmuck)

The advancement of technology has revolutionized the way we do things. It has provided us with means to improve our lives and maximize our sources, time and effort. With the use of these technologies many transactions have become faster, more efficient, accurate and effective to the management. Technology is important in today's world because it serves a variety of functions in many of the most important aspects in modern society like communication, education and business.

Furthermore, due to the COVID-19, the proposed Management and Booking System with Online Reservation aims to create a safe distance for the customer and it also helps the restaurant to organize data that computerizes the process of managing their transaction and recording inventory of their food and beverages.

2.2 The stakeholders of the project

2.2.1 Internal stakeholders

- **Project manager** : who will make sure the system satisfies all the requirements as well as ensure the project is completed on time and within budget.
- **Clerk** : Who will responsible for monitoring the orders . After the order is done , the clerk will package the food and deliver to the customers.
- **The kitchen staff** : who will receive the order and prepare the food.

- **System developers** : who are responsible for implementing the system and repairing any errors that may occur.
- **Restaurant owner** : who makes the requirements and supplies the resources.

2.2.2 External stakeholders

- **The customers**: who make the orders.
- **Suppliers**: who provide the ingredients for making food.

2.3 The expectation

2.3.1 General Objectives

This project is to develop an automated system using web-based and QR code technology as an online booking and arrangement which can replace the manual operation currently used by the traditional restaurants.

2.3.2 Specific Objectives

This project aims to attain the following:

- There will be no direct contact between clerk vs customers.
- To provide a fast, efficient and reliable system by managing the records of all their transactions.
- To evaluate the acceptability of the proposed system.
- To lessen paperwork, reduce human errors and make sure that the customer is satisfied with the service rendered.
- A computersystem that enhances the productivity of their catering services in terms of reservation and adds for other information.
- The system is internet. The system is intended for Administrator, Organizer, and customers. All users can view a list of menus through the website.

2.4 The scope of the project

The system includes: Booking, Billing, Inventory, Website, and Reservation. This system will change the manual system of the business in order for the user to work easier and provide faster transactions for the customer. It can produce an accurate and reliable report which includes print-outs of receipts for customer's payment and inventory of menus, beverages, etc.

2.4.1 Project objective

Increase business intelligence, wasted effort and opportunity to scale to a large business.
The system should support take-away options for customers.

2.4.2 The deliverables

In the first phase , we will need to determine the requirements of the system like what technologies we need to apply, some basic functions of the system like payment, menu bar and shopping list. Secondly, we need to deliver a diagram for modelling the major functional requirements for the whole system.

Thirdly, a implementation diagram for major functional requirements must be specified to to describe the overall architecture of the whole system.

Finally, we will transform all the diagrams to a complete application that should satisfy all the significant tasks for the customers as well as the staff.

2.4.3 Technical requirements

- The system should be implemented using Web technology . For example, HTML with CSS and JavaScript, Flutter, Django, etc.
- Customers are able to use the app from a mobile device, a tablet device or a normal computer or laptop.
- The system should also be accessed through a QR code so that customers do not need to install the app for ordering.

2.4.4 Limits and exclusions

- The system must be able to handle at least 300 orders per day.
- The system should be stable 24/7 .

3 Task 1.2

3.1 Functional and non-functional requirements of desired system

3.1.1 Functional requirements for each actor:

Customer

Access function : Customers can access directly to the restaurant's web service via a URL or scan the QR code using their phones , tablets or computers to access the online menu.

Ordering functions:

- When accessing the menu successfully, there will be some recommendations for the customer (best-seller or new dishes) .
- Customers can choose categories of food via a navigation bar.
- While viewing the menu, customers can click on a specific dish to see the detail of the dishes and they can choose to include side dishes.
- The customers can add the dishes they would like to have by clicking the “Buying” button then the customer can set the quantity of the food . Finally , the customer can add the dish to their cart by clicking the "Adding to cart" button.
- The customer can also modify the quantity of each dishes in their own cart .If the quantity of a specific item is set to 0, the system will automatically remove the item from their cart.

Payment functions :

- The customers will be able to see the cost of each chosen dish on the right-side cart and total payment of their orders that is displayed at the bottom of the cart.
- The customers can click the “Payment” button to view the available payment options such as by card ,by e-wallet or by internet banking .In case of using card or e-wallet , we are able to choose a specific available services (ex: Visa, Master Card, MOMO,ZaloPay , ...) .
- The app should provide a form for customer to fill in basic information for authentication purpose before they can click on “pay” button to start to the payment process or click on “cancel” button to cancel the transaction.

Other functions:

- When the system finishes confirming the order and payment methods, the customers will wait to receive their food .
- The customers can also give feedback if they want to by clicking the “Feedback” button and start filling in the feedback form .

Clerk

- The clerks can have the right to receive and monitor the orders through the system to keep track with the orders .
- Clerk prepare tables if the system informs that there are some table reservations .
- When the food is ready, the clerk will be responsible for packaging and delivering to the pick up area or shipper.

Administration system

- After receiving the orders sent by the customer, the system will start processing the orders by checking if the dishes can be prepared or not. If the ingredients are enough, the terminal will record the order. Else, the terminal will send a notice to the customer about the shortage and ask the customer to modify their choices.
- If an order is successfully processed, the terminal will send a notification to the kitchen to inform them that the dishes have just been ordered.
- The terminal also receives information about the payment and records the transactions if they are valid.
- If a table reservation is valid, the system will inform clerk to prepare the table
- System records basic data of customer in database for future recommendation or give them discount.

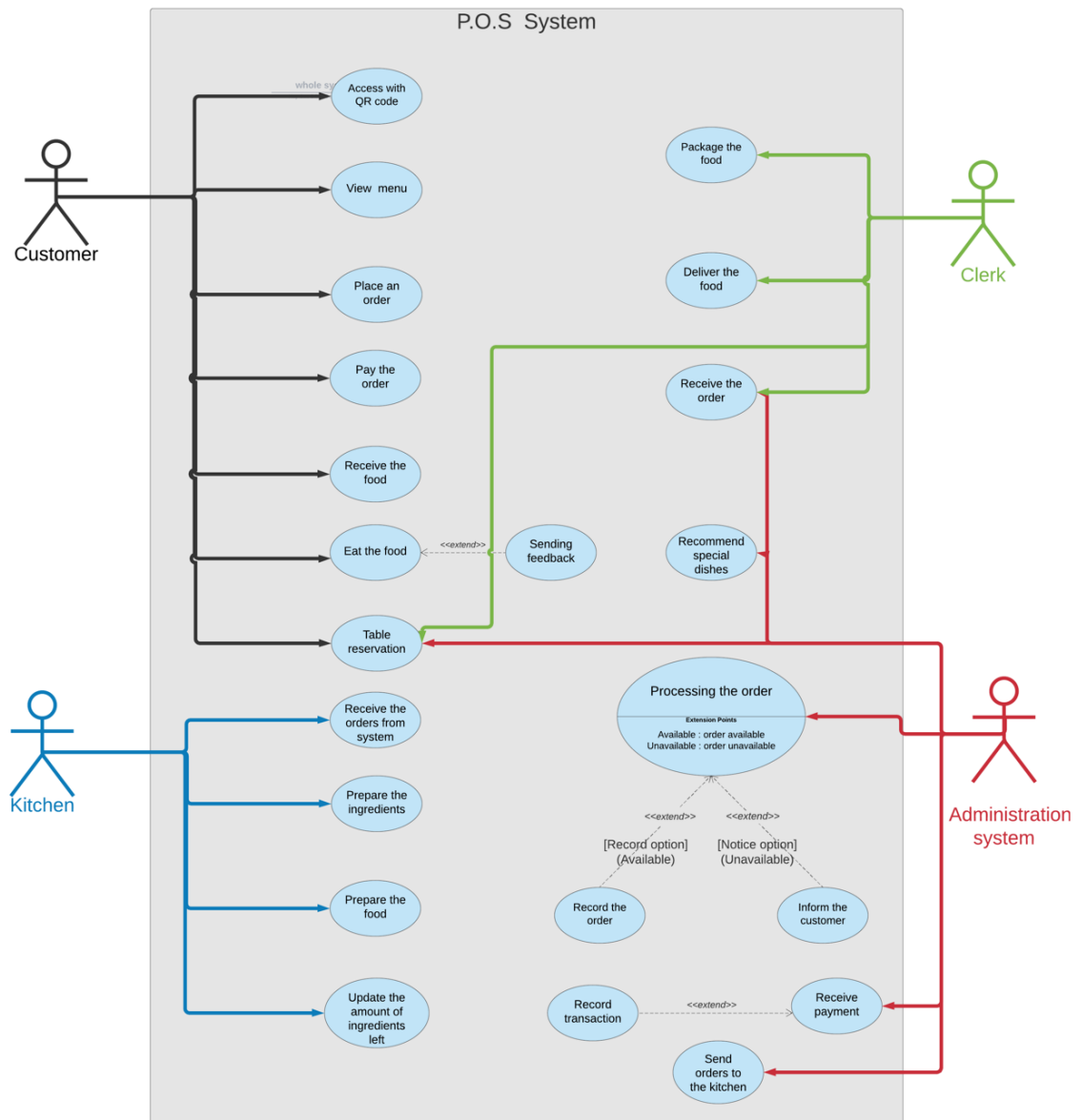
Kitchen

- The kitchen shall receive the orders from the system and prepare the necessary ingredients to prepare the dishes.
- The kitchen will start cooking if the ingredients are gathered enough before giving meals to the clerk
- Make notification to the system about amount of each type of dishes left after cooking.

3.1.2 Non-functional requirements for each actor:

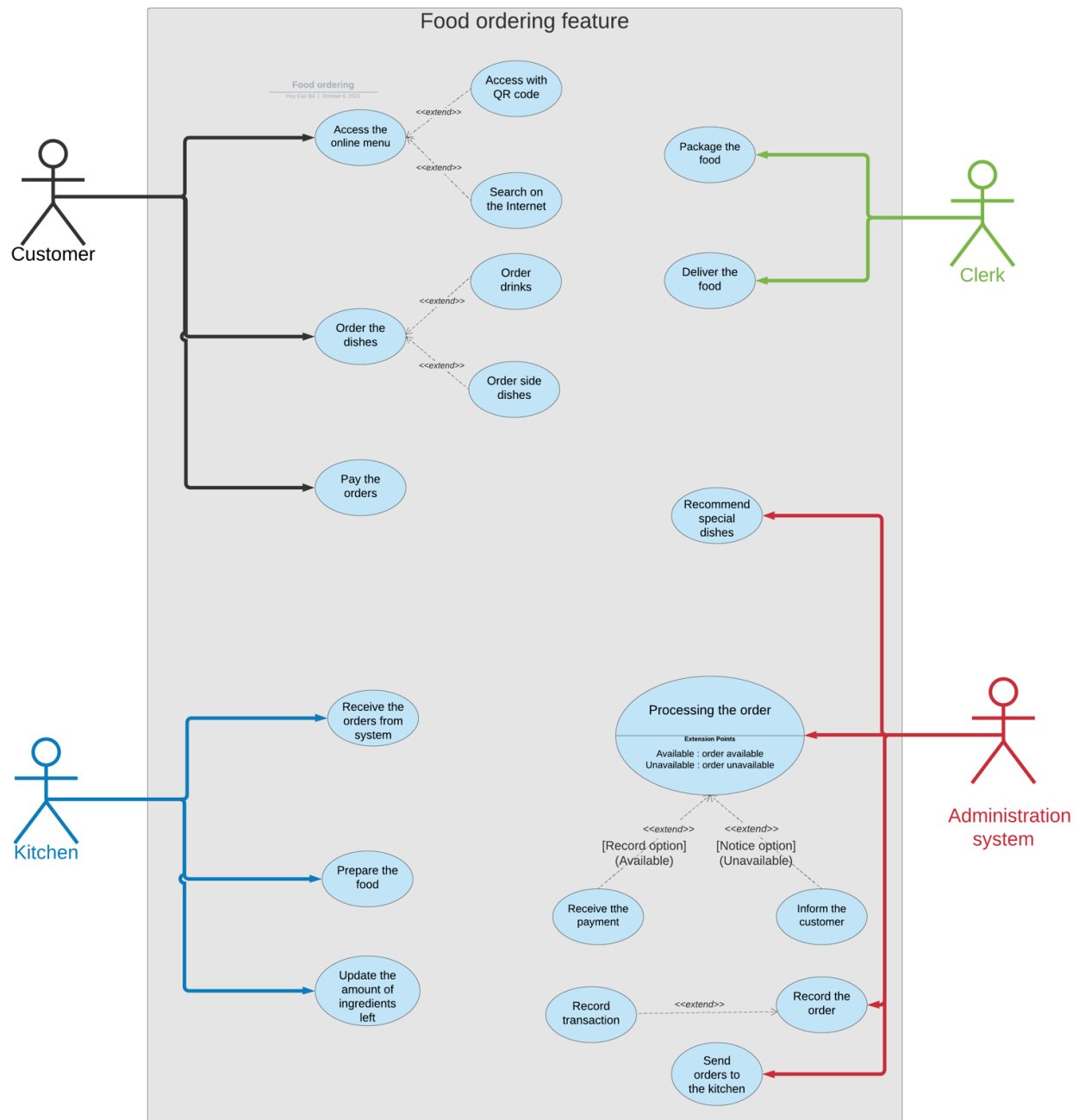
- The system should be able to handle at least 300 orders per day.
- The application must be usable from a mobile device, a tablet device or a normal computer/laptop.
- The system has the probability of unavailability of less than 5% in order to avoid order loss.
- The system must also have a fast response time of less than 3 seconds.
- User friendly - not too many buttons and information at a time. A normal order should take 5-7 clicking operations to be done.
- The system must guarantee not to leak any customer information outside.
- The system must be available 24/7.

3.2 Use Case Diagram for the desired system



4 Task 1.3

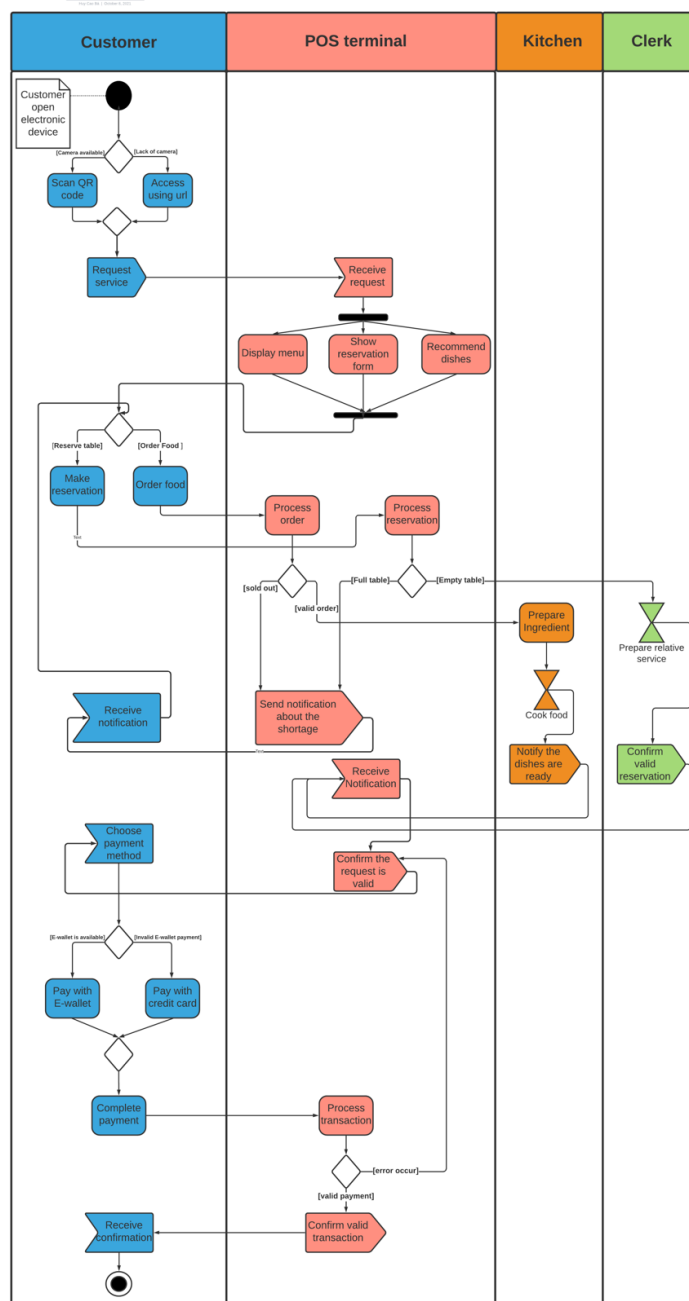
4.1 Use case diagram for food ordering feature



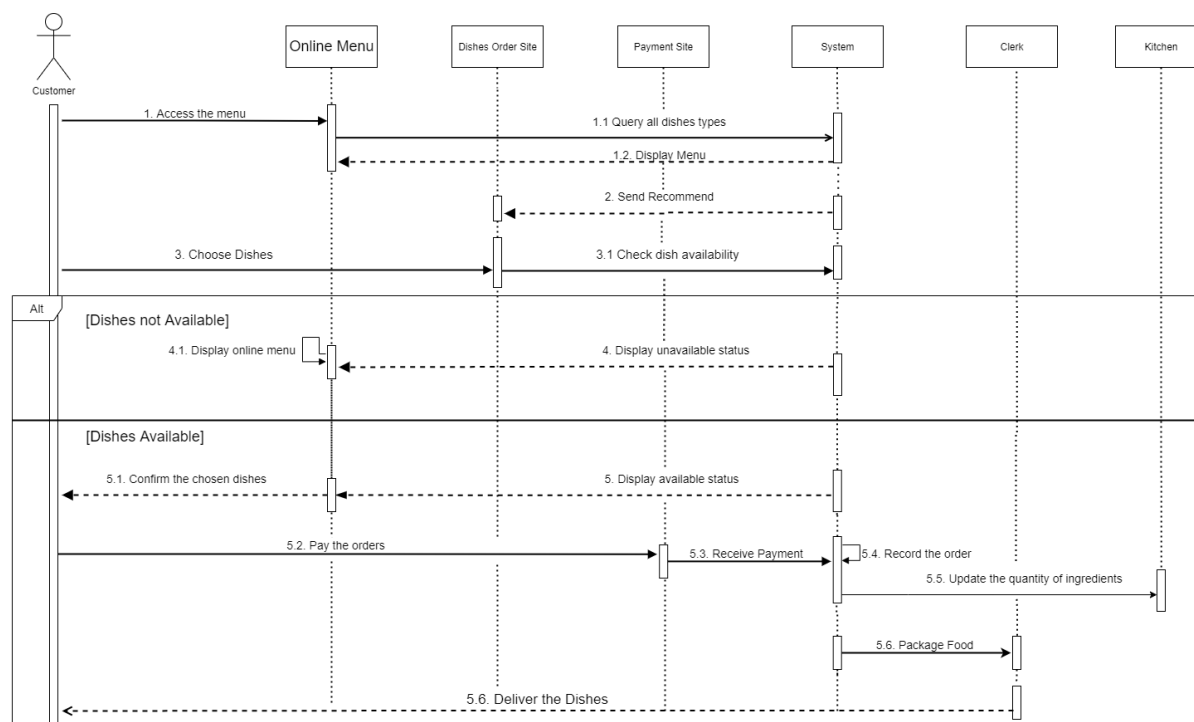
4.2 Use Case Description for food ordering feature

Name	Food Ordering
Actor	Customer, Kitchen Clerk and Administration System.
Description	With food ordering, customers can access menu digitally which helps drive safety, compliance, and green initiatives.
Preconditions	Customers need to access the restaurant website with QR code or search online.
Normal Flow	<ol style="list-style-type: none">1. Customers access the online menu.2. System recommends special dishes.3. Customers order the dishes.4. The Administration System checks the order availability. If the dishes are available then the system will proceed to receive the payment and record the order. Else, the system will inform the customers about the shortage and ask the customer to choose again.5. The Kitchen will receive the notifications about the order and prepare the food. After that the Kitchen will update the quantity of the ingredients.6. Clerk packages the food.7. Clerk delivers the food to the customer.
Exceptions	Exception at step 4: Customers will have to order another dish if their previous dish is not available.
Alternative Flow	At step 3, customers can choose another dishes.

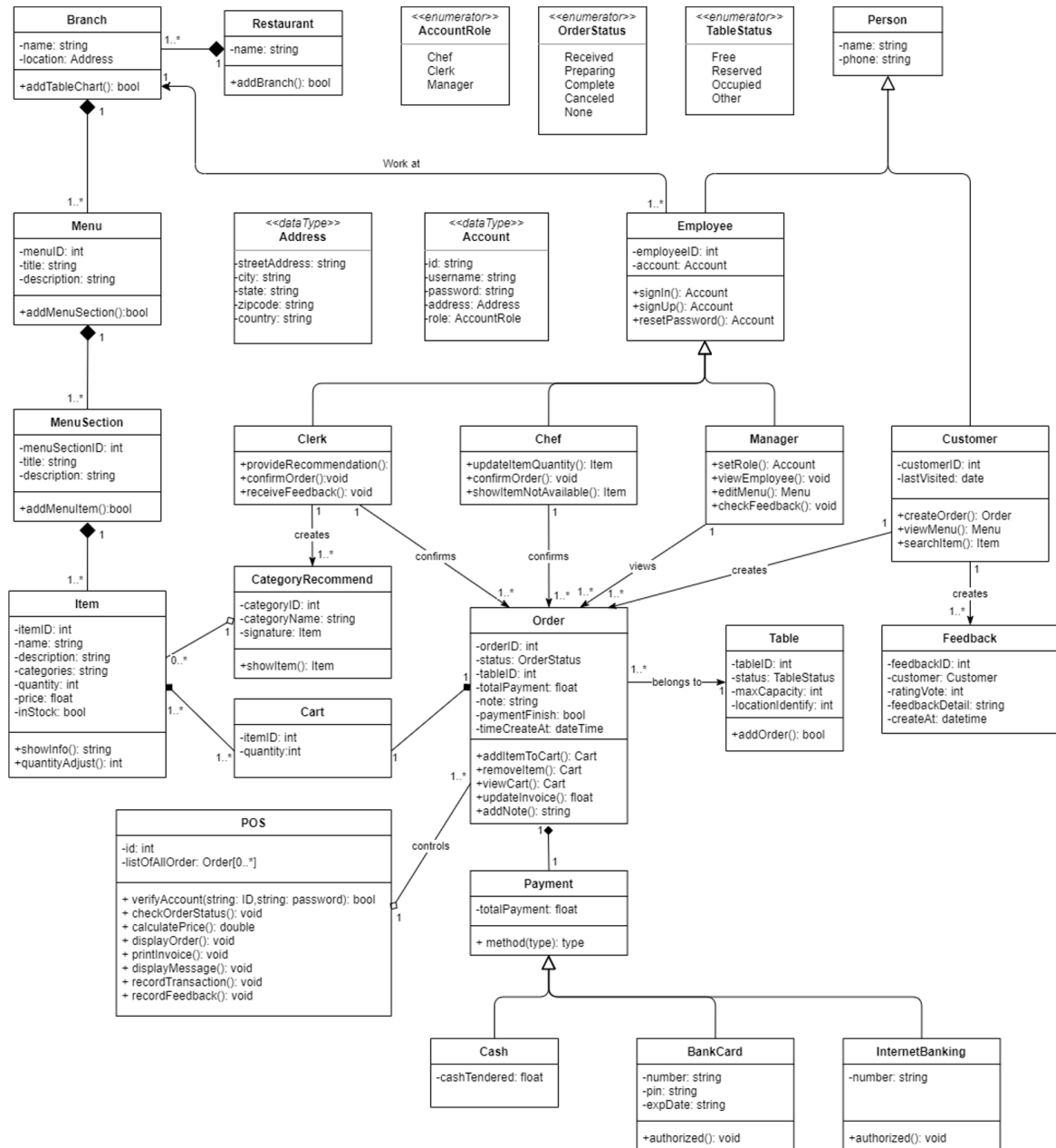
5 Task 2.1 : Activity diagram



6 Task 2.2 : Sequence diagram



7 Task 2.3 : Class diagram



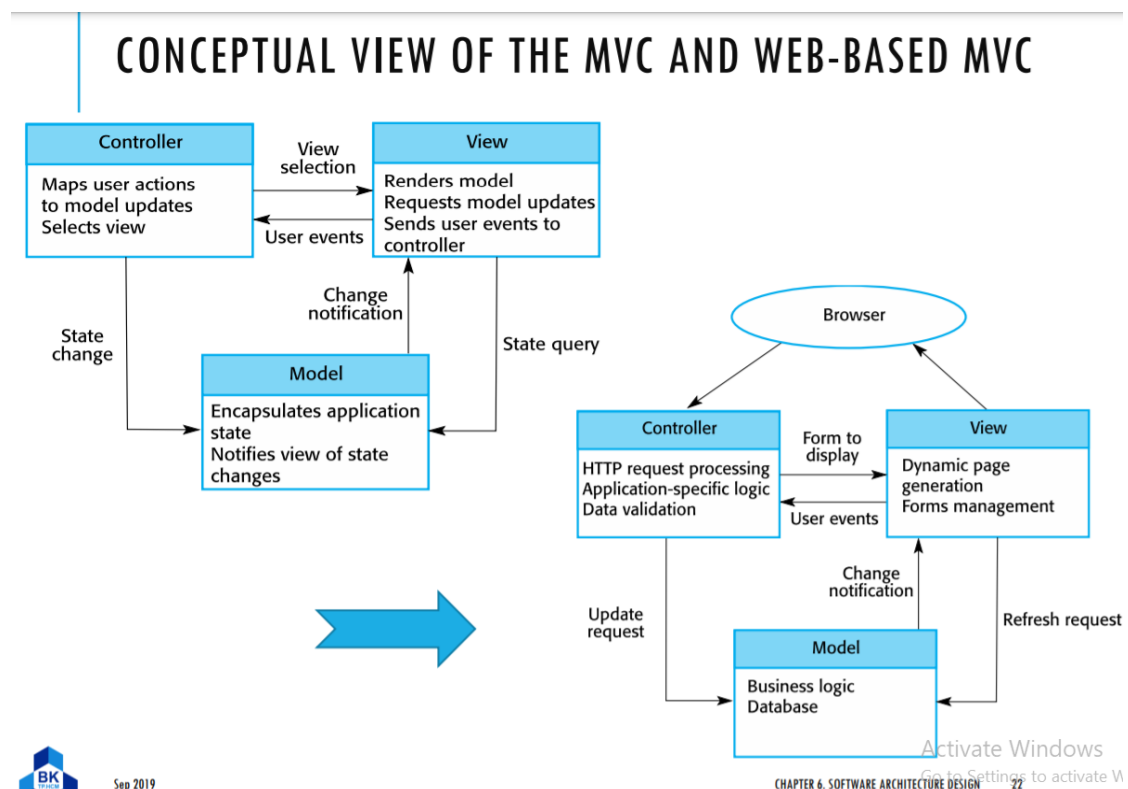
8 Task 3.1 : Architecture pattern

For our desired system we will use the “Model-View-Controller” and “Microservices” approach because there are many advantages of these approaches compared with other approaches like layered, client-server, master-slave, pipe-filter approach. . . and those 2 approach are really suitable for our system based on non-functional requirements. Let’s analyze it.

8.1 Introduction about Model-View-Controller (MVC) and Microservices architecture pattern

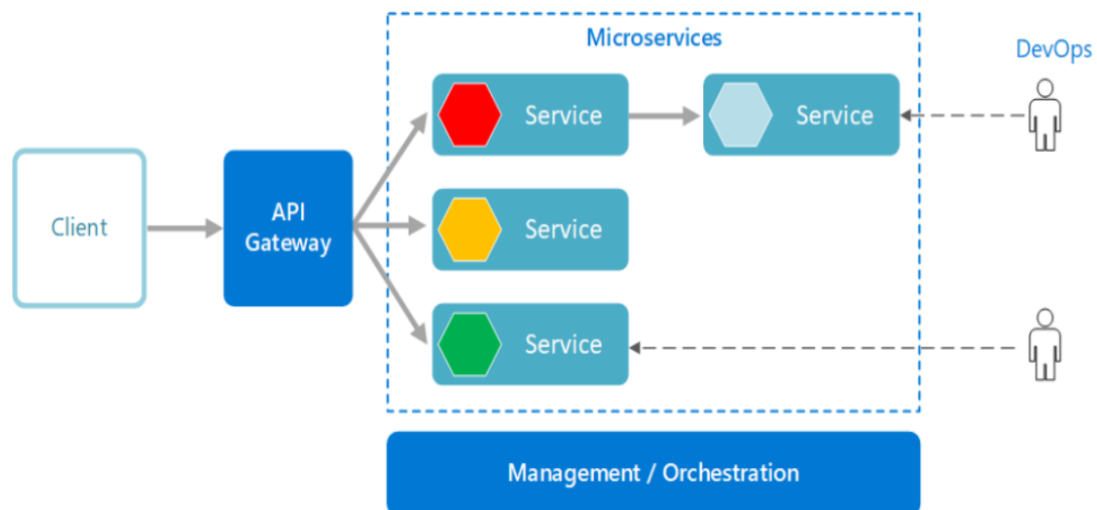
Model-View-Controller pattern

The Model-View-Controller (MVC) is a software design pattern that separates an application into three main logical components: the model, the view, and the controller. Each of these components are built to handle specific development aspects of an application. MVC is one of the most frequently used industry-standard web development frameworks to create scalable and extensible projects especially in developing user interfaces.



Microservices pattern

A microservices architecture consists of a collection of small, autonomous services. Each service is self-contained and should implement a single business capability within a bounded context. A bounded context is a natural division within a business and provides an explicit boundary within which a domain model exists. Below is the example of Microservice architecture pattern :



8.2 Comparison

Microservice pattern

- **Microservice:** Each service is small, independent, loosely coupled, and forms a separate codebase, which can be managed, deployed independently by a development team. Services are responsible for persisting their own data or external state and they can communicate with each other by using well-defined APIs.
- **Management/orchestration:** This component is responsible for placing services on nodes, identifying failures, rebalancing services across nodes, and so forth. Typically this component is an off-the-shelf technology such as Kubernetes, rather than something custom built.
- **API Gateway:** The API gateway is the entry point for clients. Instead of calling services directly, clients call the API gateway, which forwards the call to the appropriate services on the back end.

MVC pattern

- **Model:** The Model component corresponds to all the data-related logic that the user works with. This can represent either the data that is being transferred between the View and Controller components or any other business logic-related data. For example, a Customer object will retrieve the customer information from the database, manipulate it and update its data back to the database or use it to render data.
- **View:** The View component is used for all the UI logic of the application. For example, the Customer view will include all the UI components such as text boxes, dropdowns, buttons, effects, ... that the final user interacts with.
- **Controller:** Controllers act as an interface between Model and View components to process all the business logic and incoming requests, manipulate data using the Model component and interact with the Views to render the final output. For example, the Customer controller will handle all the interactions and inputs from the Customer View and update the database using the Customer Model. The same controller will be used to view the Customer data.

8.3 Why we choose these patterns

MVC pattern

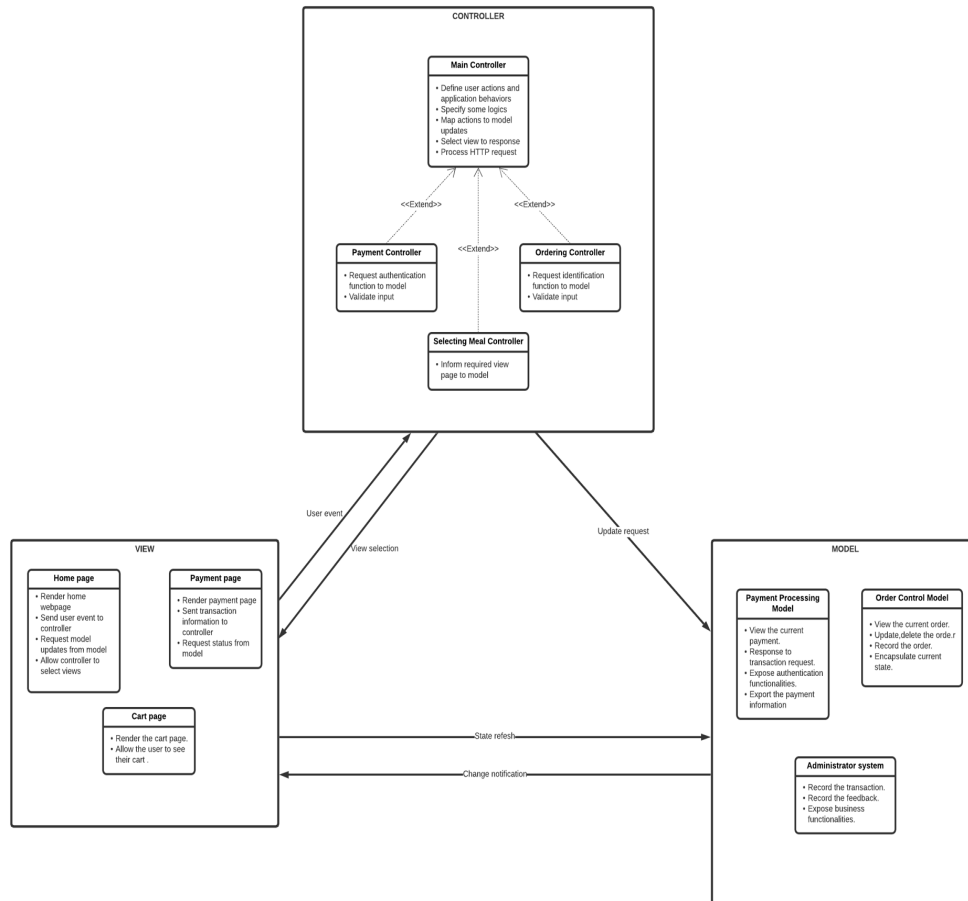
- MVC model is used for current development and deployment when our customer's restaurant need a change to be operated in the pandemic period (chosen architecture pattern)
- MVC model ensures fast development process which is the key point for our customer to quickly reoperate their restaurant.
- MVC model provide multiple views: it separates data and business logic from the display
-> limit code duplication

- The MVC model makes it easy to plan and maintain our system so for any modification or upgrade it will not affect the whole system.
- In the MVC model, code is not too complex and can be reused so when a restaurant has new features, functions or in case another customer wants to use our product for their restaurant, it is really convenient for the developing team. This model is a well-tested architecture so we can be sure our system or even demo may actually work.

Microservice pattern

- Microservice model can be used for future development when our customer's restaurant chain become big enough with maybe over million of users per day (optional architecture pattern)
- MS model helps us to optimize sizing and easy to integrate with third-party services. For example, when our customer has a big chain of restaurants, and they decide to add a live-stream feature into the available system or maybe real-time chatting and reviewing online features...
- Each microservice has its own database so it reduces the damages when there is a loss of data, an error in operation or a hacking event occurs. You can imagine what will happen if a large restaurant chain in the FnB industry loses their customer information like social security number, phone, payment-card number or transaction details...
- It allows us to make appropriate, service-specific decisions and enables companies to optimize resources for development and applications.

8.4 MVC diagram



9 Task 3.2 : Implementation diagram

