



HO CHI MINH UNIVERSITY OF TECHNOLOGY
FALCUTY OF COMPUTER SCIENCE AND ENGINERRING

Software Engineering

Assignment 2

Task 1 & 2

Advisor: Quản Thành Thơ

Member:

- **Võ Phạm Hoài Nam (1952863)**
- **Võ Trần Minh Đạt (1913086)**
- **Huỳnh Kim Hưng (1952745)**
- **Trần Đức Nam (1952861)**
- **Nguyễn Quý Hải (2052974)**

Table of Contents

Table of Figures	1
Table of Contributions	1
History Log	2
Task 1.1.....	3
1. Context. Placeholders. Scope. Task.....	3
1.1. Context.....	3
1.2. Stakeholder	4
1.3. Scope	4
1.4. Task to be done.....	4
Task 1.2.....	5
2. Functional Requirements of online restaurant system.....	5
2.1. Customer	5
2.2. Clerk.....	5
2.3. POS terminal	6
2.4. Kitchen (or chefs).....	6
2.5. Restaurant manager.....	6
3. Nonfunctional Requirements of online restaurant system	7
4. Use-case Diagram for the whole system	8
Task 1.3.....	9
Task 2.1.....	10
Task 2.2.....	11
Task 2.3.....	12

Table of Figures

<i>Figure 1.</i> Use-case diagram of the whole system.....	8
<i>Figure 2.</i> Use-case diagram of Place order.....	9
<i>Figure 3.</i> Activity Diagram of the system.....	10
<i>Figure 4.</i> Sequence Diagram of the Place Order Use-case.....	11
<i>Figure 5.</i> Class diagram of the system.....	12

Table of Contributions

Name	Contributions
Võ Phạm Hoài Nam	<ul style="list-style-type: none"> • Describe Kitchen, Restaurant functional requirements. • Draw Customer, POS in use-case diagram of the whole system in Task 1.2. • Draw system class diagram in Task 2.3.
Trần Đức Nam	<ul style="list-style-type: none"> • Describe Customer, Clerk, POS terminal functional requirements. • Draw Clerk, Kitchen in use-case diagram of the whole system in Task 1.2. • Draw an activity diagram to capture Major (not all) functional requirements of the desired system in Task 2.1.
Huỳnh Kim Hưng	<ul style="list-style-type: none"> • Describe non-functional requirements i.n Task 1.2. • Describe view dishes use-case in Task 1.3. • Draw a sequence diagram in Task 2.2.
Võ Trần Minh Đạt	<ul style="list-style-type: none"> • Describe non-functional requirements i.n Task 1.2. • Describe view dishes use-case in Task 1.3. • Draw a sequence diagram in Task 2.2.
Nguyễn Quý Hải	<ul style="list-style-type: none"> • Describe non-functional requirements i.n Task 1.2. • Describe view dishes use-case in Task 1.3. • Draw a sequence diagram in Task 2.2.

History Log

Date	Action
26/09/2021	Perform task 1.1, 1.2, 1.3.
07/10/2021	Rewrite description in task 1.1 to be more clear in detail.
20:30, 07/10/2021	Adjust the whole system use case diagram in task 1.2: <ul style="list-style-type: none">• Re-draw the arrow to make it more clearly.• Adjust relationship between “View Dishes” use case and “Add to Cart” use case.• Kitchen will not need to login the system when working.• The orders will be confirmed by the clerk only.• Adding one more use case for the clerk, that is “Update the dishes data”.• Applying generalization relationship to the use case “Manage order” of the clerk.• Change the position of the actor in “Kitchen”.
21:30, 07/10/2021	Adjust the whole Use case Diagram(s) and Use case Scenario(s) in task 1.3.
14:00, 08/10/2021	Adjust the functional requirements in terms of Customer, Clerk, POS terminal and Kitchen in task 1.3.
09/10/2021	Perform task 2.1, 2.2, 2.3.

Task 1.1.

1. Context. Placeholders. Scope. Task.

1.1. Context

Dishy is the newest solution to the social distancing crisis for restaurants in the pandemic days. The user will be equipped with a tablet or may use his/her phone to order, pay the bills, and so much more, ensuring the utmost safety throughout the whole dining experience. The application will provide booking, ordering, and paying functions with user-friendly UI and highly performance to make sure to satisfy any foodaholic in the world.

Dishy will be applied in a traditional restaurant, called BK Food court. In detailed:

- The customer will make payment before receiving dishes;
- Payment Option: cash, credit;
- Table: The Customers are allowed to choose any available table.
- Take away: None
- Food: the food has already been processed, the manager enters the data about the dishes (such as name, quantity, description) at the beginning of the day (system check)
- Business Process:
 - Students come to the canteen to eat. They can choose any available table as they wish.
 - Students scan the QR code, which has already been provided on each table, to login the system and view the menu.
 - Student sent the order request to the system => Clerk check this request later.
 - Clerk confirms order => Student makes payment => (POS record transaction).
 - Clerk notified the kitchen to execute the order.
 - After finishing the order, the kitchen alerts the clerk to announce the customer later.
- Every time one order is successful payment, the clerks also update the data to corresponding dishes in the system.
- POS is responsible for only supporting the payment, record order (or transactions), and print bill.

1.2. Stakeholder

Stakeholders are defined as: Individuals and organizations who are actively involved in the project, or whose interests may be positively or negatively affected as a result of project execution or successful project completion. Basing on this definition, we easily has ability to realize that our restaurant online system has mainly three stakeholders:

- Customers: Those who order food from the restaurant through the online ordering system (web app).
- Restaurant owners: The owners of the online restaurant system or the restaurant itself.
- Restaurant Clerks: Who mainly log in the system to manage and control it.
- Kitchen: Who receives the order which was checked by the clerk, then executes that order.

1.3. Scope

The software/systems provides the following facilities to the customers:

- Customers are provided with up to date information on the products available.
- Provides information about the products in categories.
- Customers and clerks should be able to communicate in a non-direct manner through the system.

The objective of this software is to provide easy assistance to both the customer as well as the merchant with proper database and information. And furthermore, In the future, the system should be able to be used in multiple restaurants.

The system does not support take away.

1.4. Task to be done

- Identify correctly about the requirements of the system (using non-functional requirements and functional requirements).
- Draw some kind of key diagrams (such as use case diagram, activity diagram, class diagram, and sequence diagram) to have better views about the systems.
- Starting implements the systems by code.
- Testing to check whether the system satisfies the requirements or not. Re-implement the system if it does not meet the requirements.
- Operate and maintain the system.

Task 1.2.

2. Functional Requirements of online restaurant system

2.1. Customer

- The customer can approach or engage with the restaurant menu by scanning the table QR code or the website address via their mobile devices.
- After approaching the web app, the customers shall be able to view the categorized menu from the website.
- The display menu also shows the tags such as “best-seller”, “special combo”, “Big discount combo” to the customer when they login to the web app.
- The customer shall be able to dismiss the recommended or best seller food tags.
- The customer shall be able to search for the food as he/she wishes in the search engine.
- When the customers select some kind of dishes such as Hamburger, Beefsteak, they can also have ability to select the quantity and type in any special adjustments to the dish (example: extra vegetable)
- Whenever the customers make a selection or adjustment on their cart, the total price (including taxes if this meal has) will correspondingly change.
- When re-check the cart, customers can delete some dishes that they think are not necessary.
- After being satisfied with the chosen dishes, the customer can place an order.
- After the orders, which are viewed and confirmed by the restaurant clerk, are successful, the customer can move to the check-out step. There will be various payment methods for customers to freely choose from (cash, digital wallet, credit card).
- When the check-out step is complete, the customer is engaged to request an invoice .
- Even if the customer has a progressing order, they are completely engaged to make other orders whenever they wish.
- Customers can give personal feedback to the dishes as soon as they receive and try the taste of them.

2.2. Clerk

- A clerk is able to log into the system with a specific role, which is distributed by the restaurant manager (or the owner of Foody).
- A clerk can log out the system.

- Whenever an order request is sent, the clerk is responsible for confirming the order.
- At the end of the ordering section (after the customers made payment for confirmed orders), the clerk will update the data to corresponding dishes.
- The clerk can access the feedback box to view and send the reply for the corresponding feedback.
- The clerk also can have ability to manage the customer order which can lead to two cases:
 - Case1: The order is valid then the clerk will confirm and notify the customer that the order is successful.
 - Case2: The order is not valid then the clerk can cancel the order and alert the customer that there are any problems in your order.

2.3. POS terminal

The data of the dishes (quantity, ingredients) on the system should be allowed to update manually and daily.

If the customer requests to print an invoice for the order, the POS can satisfy this requirement.

For every transaction made, it should be stored in the database.

POS supports several payment methods, such as: credit card, digital wallet.

2.4. Kitchen (or chefs)

After the order is successfully checked by the clerk, it will pass to the kitchen in which all the ingredients are prepared to cook.

2.5. Restaurant manager.

The manager (or sometimes owner) of the online system restaurant has the ability to access most of the system functions such as view and send feedback, query transactions record... etc. Additionally, the manager also can limit the access scope of each type of account in the system.

3. Nonfunctional Requirements of online restaurant system

Availability	The user should be aware of your restaurant application and be able to use it directly from the restaurant website or by scanning the QR code without download requirement to the device.
Screen Adaption	When the user uses a different type of mobile device, tablet device or normal computer, the system or application should render it's layout to different screen sizes. Along with automatic of Font size and image rendering.
Scalability	The system should be extendable to use in multiple restaurants in the future. In the other words, when the user data (caches , stored data etc) increases, the app should be capable of handling them without delay by optimising the way storage is done and accessed.
Performance	When any customer sends the request, the request should take no more than 3s to send the response to customers.
Use-ability	Users should be able to understand the flow of the web app easily. In the other words, users should be able to use the web with some simple guidelines.
Capacity	To make sure that there are not any mistakes happening on the transactions over the first year, the current transactions are about 300 orders per day.
Maintenance	Maintenance of the system will be conducted weekly. In detail, the system will be conducted during off-peak hours.
Security	Access will be controlled with usernames and passwords which are accessed only by the system authentication. On the other hand, databases should be reasonably secured to prevent leak or loss of confidential information such as credit card details from customers.

4. Use-case Diagram for the whole system

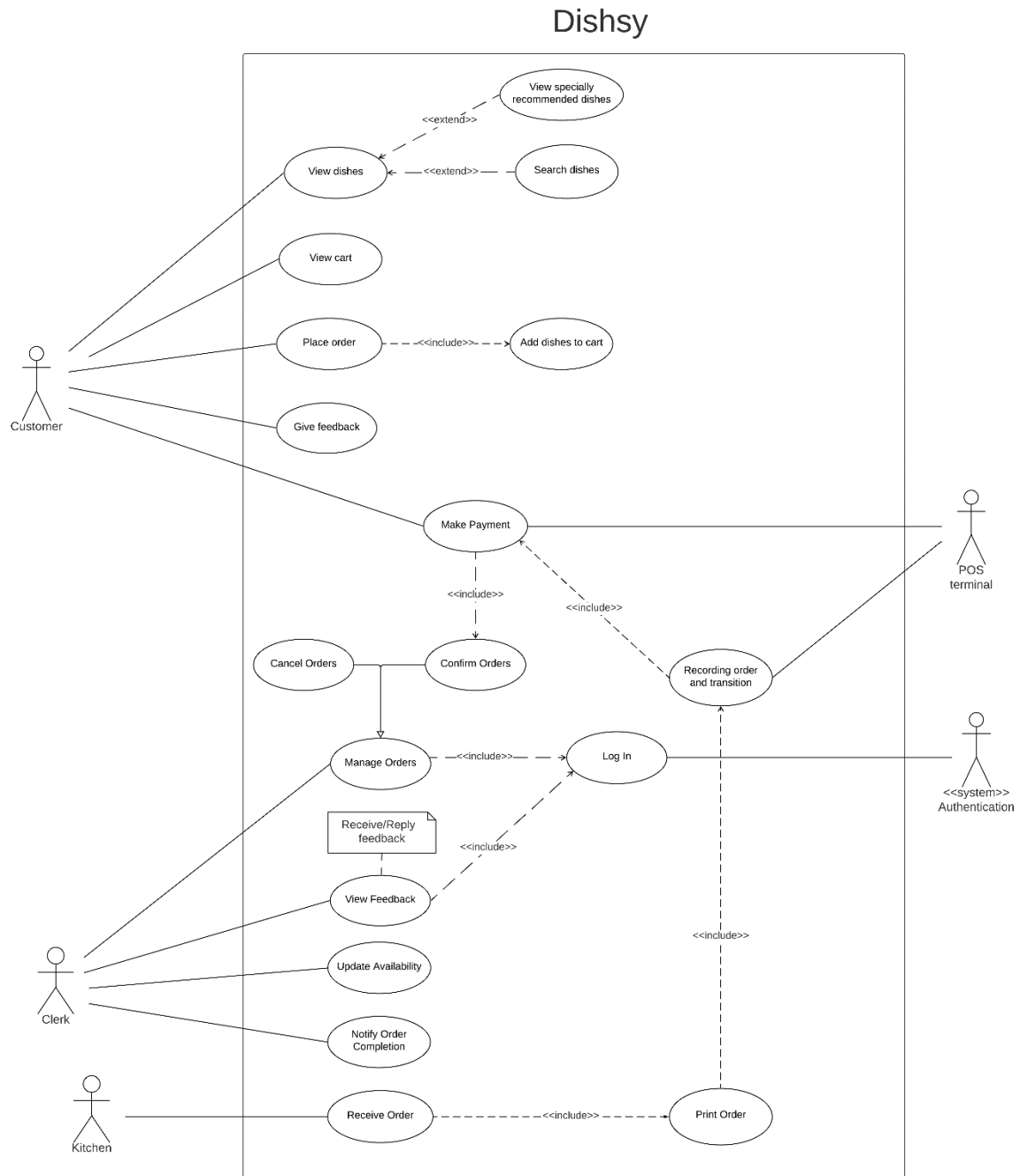


Figure 1. Use-case diagram of the whole system.

Task 1.3.

We choose food ordering feature to specify its use-case diagram and describe the use-case using a table format.

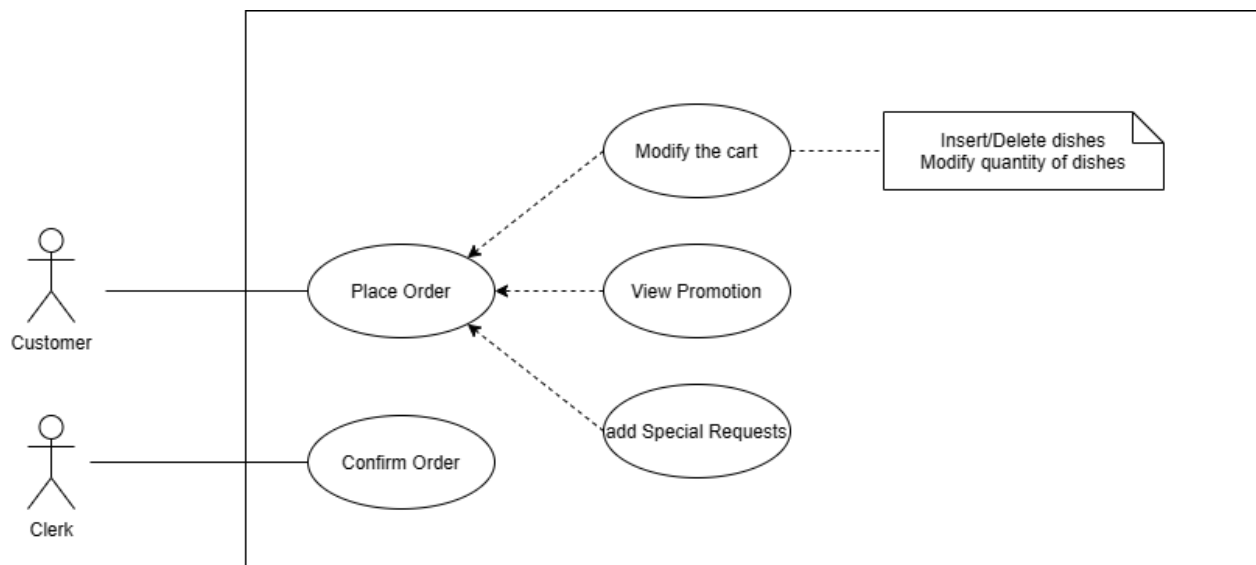


Figure 2. Use-case diagram of Place order

Use case name:	Place Order
Actor:	Customer
Description:	Customers can place orders, add special requests, view promotions, and the order can be recorded by POS and Clerk.
Pre-condition:	1. The Customer has a cart with at least 1 dish(es) and chooses to place the order.
Normal Flow	1. The system displays the cart again. 2. The Customer can review the cart and modify it (Insert/Delete dishes/modify quantity of dishes). 3. The Customer can add special requests. 4. The Customer can view promotion. 5. The system will send the Customer's order information to the Clerk if Customer confirms the order and chooses "Place order" button. 6. If the Clerk checks that the order is OK, the system will display the message "Order was successful" and navigate Customer to the Payment screen. 7. End the Use Case.
Alternative Flow	Alternative 1: At step 6. 6a. If the Clerk checks that the order has a problem (out of stock, etc), the system will display the message "Order was not successful" and back to step 1 in Normal Flow.
Exceptions:	Exception 1: At step 2. 2a. If the Customer cancels the order, the system navigates the Customer to the home screen. 2b. End the Use Case.

Task 2.1.

Draw an activity diagram to capture major functional requirements of the designed system.

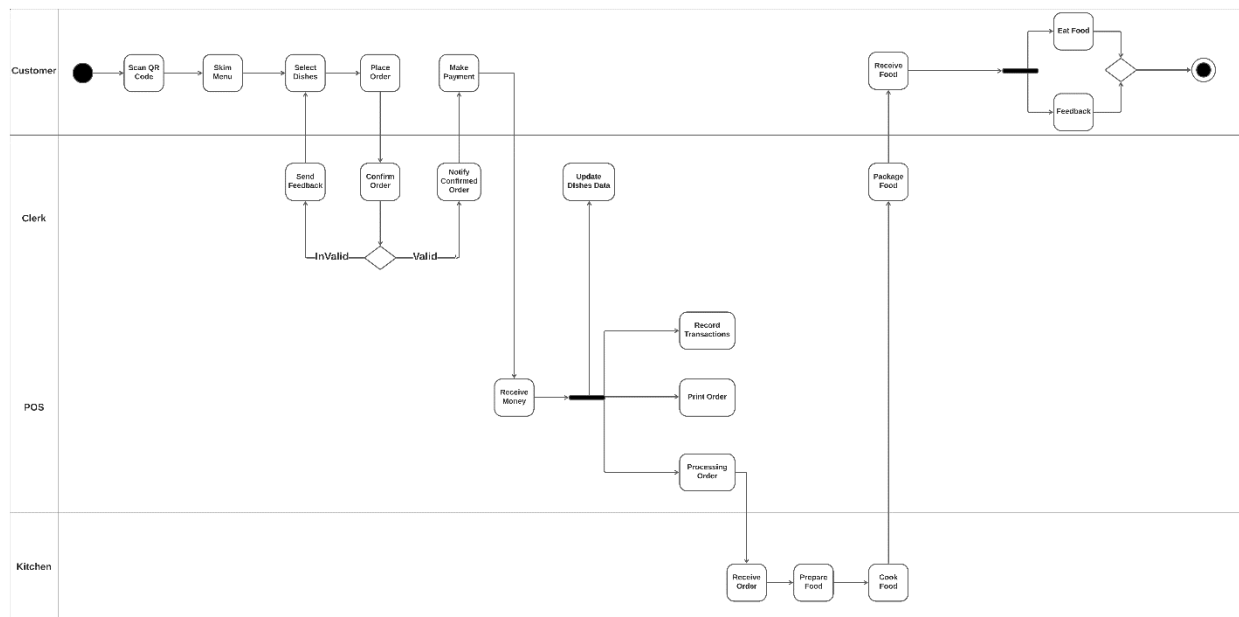


Figure 3. Activity Diagram of the system.

Task 2.2.

Draw a sequence diagram for use case in task 1.3. (Food ordering feature).

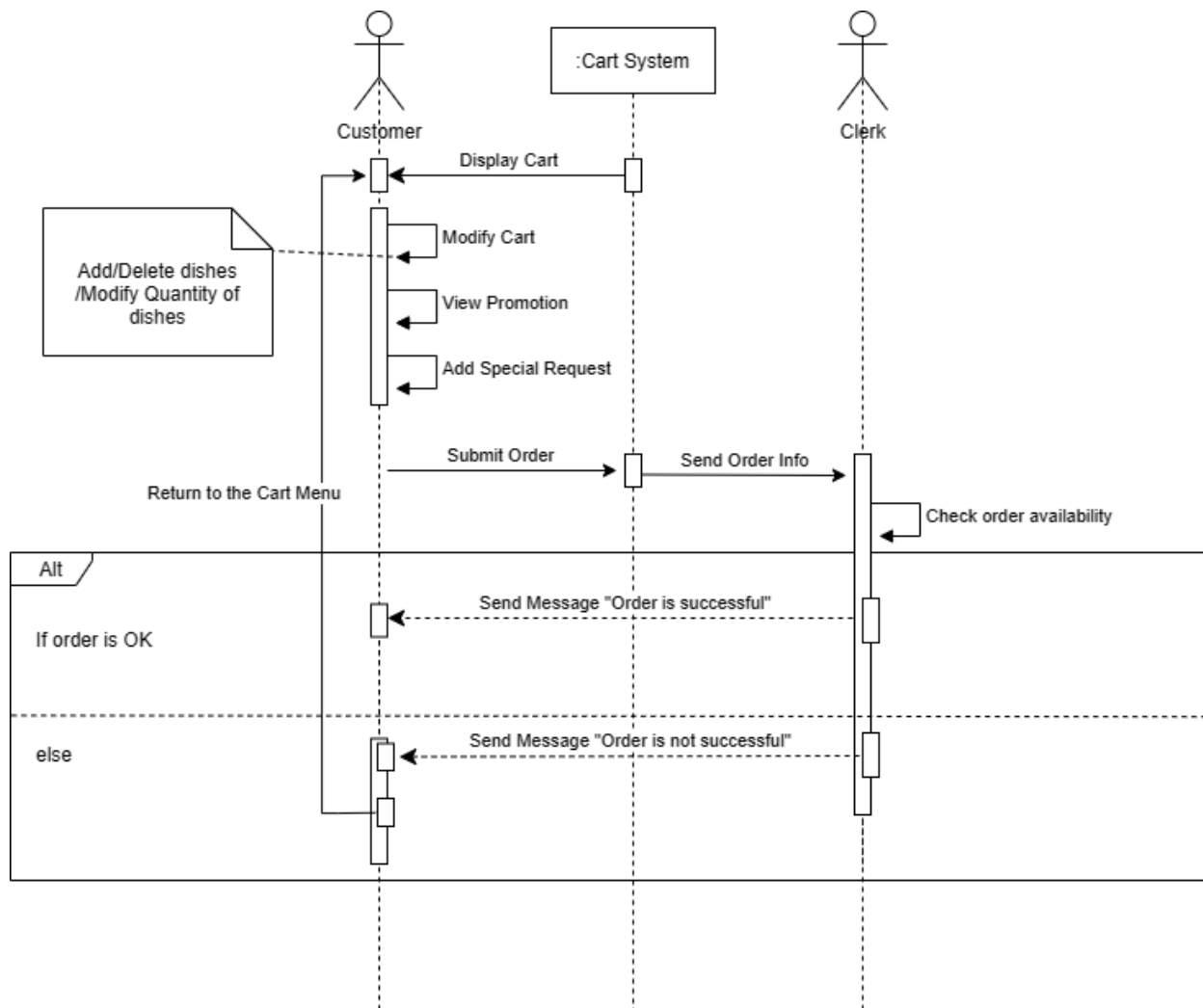


Figure 4. Sequence Diagram of the Place Order Use-case

Task 2.3.

Draw a class diagram to the whole system.

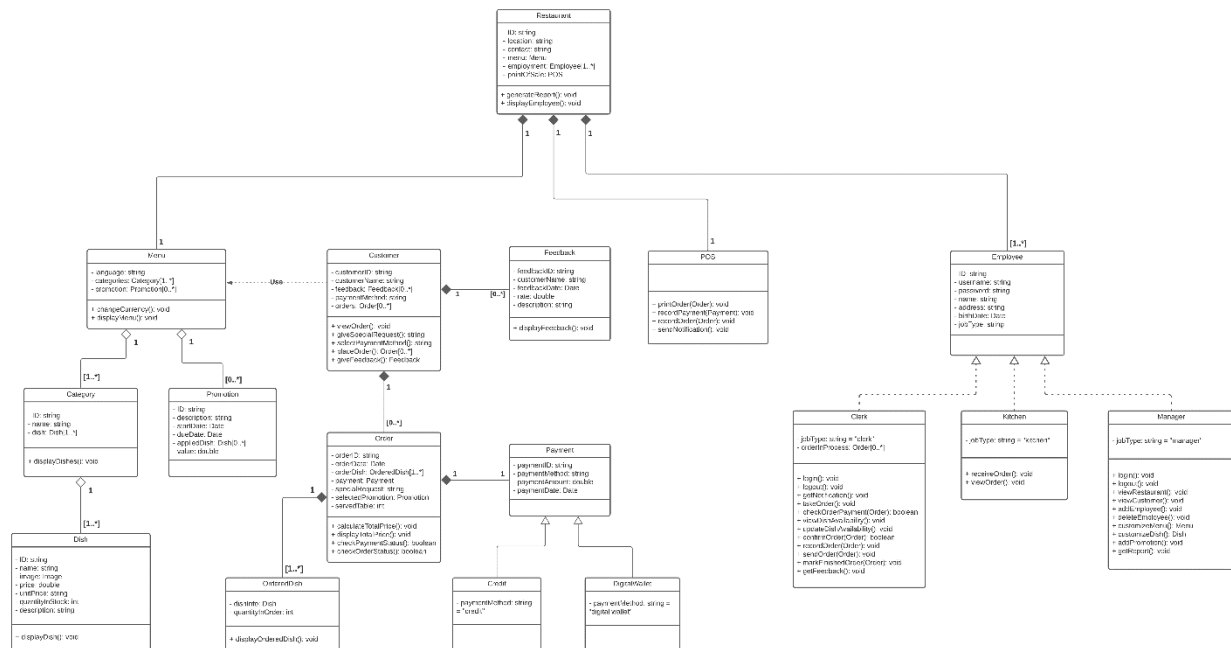


Figure 5. Class diagram of the system.

In this diagram, all getter and setter methods for the private attributes will not be listed for simplicity. We just specify the main methods.