

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



## **SOFTWARE ENGINEERING (C3001)**

---

**Assignment**

# **Restaurant POS 2.0**

---

Advisor:	Assoc. Prof. Bui Hoai Thang	
Students:	Ngo Nguyen Khoi Nguyen	1952544
	Nguyen Trong Tin	1752545
	Tran Nhat Tan	1852732
	Le Tien Loc	1952328
	Tran Tien Phat	1952386



## Contents

<b>1</b>	<b>Require elicitation</b>	<b>2</b>
1.1	Task 1.1 . . . . .	2
1.1.1	The context of project . . . . .	2
1.1.2	Stakeholders . . . . .	2
1.1.3	Scope of project . . . . .	2
1.2	Task 1.2 . . . . .	3
1.2.1	Functional requirements . . . . .	3
1.2.2	Non-functional requirements . . . . .	4
1.3	Task 1.3 . . . . .	4
<b>2</b>	<b>System modelling</b>	<b>6</b>
2.1	Task 2.1 . . . . .	6
2.1.1	Staffs can add, delete, edit products or change the restaurant information . . . . .	6
2.1.2	Users can view, add and order food . . . . .	7
2.1.3	Users can add and update their information . . . . .	8
2.1.4	Users can give the product feedback . . . . .	9
2.1.5	Employee Management . . . . .	10
2.2	Task 2.2 . . . . .	11
2.3	Task 2.3 . . . . .	12
<b>3</b>	<b>Architecture design</b>	<b>13</b>
3.1	Task 3.1 . . . . .	13
3.2	Task 3.2 . . . . .	13
<b>4</b>	<b>Implementation - Sprint 1</b>	<b>15</b>
4.1	Task 4.1 . . . . .	15
4.2	Task 4.2 . . . . .	15
4.3	Task 4.3 . . . . .	15
4.3.1	Customer side . . . . .	15
4.3.2	Manager side . . . . .	17
<b>5</b>	<b>Implementation - Sprint 2</b>	<b>20</b>



## 1 Require elicitation

### 1.1 Identify the context of this project. Who are relevant stakeholders? What are expected to be done? What are the scope of this project?

#### 1.1.1 The context of project

- Payment Option: Credit card
- Reservation: Customers can reserve table on the restaurant website, they can also pre-order food
- Food: At the beginning of the day, the kitchen will update the ingredients quantity. After that the system auto calculate the quantity of a meal to show the availability. However, during the day, the kitchen can update the ingredients or the meal manually (because there may be some problems that led to reduction of the ingredients)
- Menu, meal and price: can only be defined by the manager or the restaurant owner.

#### 1.1.2 Stakeholders

- End users:
  - Restaurant customers: Order food, reserve table, give feedback
  - Restaurant staffs: Manage receipts, list of food, payment, booking and customers service
- System managers: IT staffs implement and maintain the system
- System owners: Restaurant owners manage staffs, list of food, revenue
- External stakeholders

#### 1.1.3 Scope of project

- **Justification**
  - Managing restaurant can be easier by using POS system, which is developed based on web-platform.
  - POS system makes the reservation, payment more convenient and tracking profit more clearly.
  - The system, moreover, allows the restaurant owner to manage their staffs, food and revenue.
  - Being built in web-platform, the system guarantee the stability on several devices, security and convenience.
- **Description:** POS system provides basic functions such as food ordering, table reservation, online payment. It is almost the same as traditional restaurant but more flexible as everything is online.



- **Acceptance criteria:** In term of technology, the system is experienced and assess by the restaurant staffs. The project is considered to be complete if the rating point is more than 4 over fifth. In term of user experience, the project is considered to be complete if the user rating point is more than 4 over fifth.

- **Deliverable**

- Project manager has to provide the customer with documents related to implementation and maintaining the system such as SRS (system requirement), SDD (system design), STD (system testing document), some analysis about the system and the user manual.
- Project has to be handed over to the customer after testing and acceptance.
- Customer will be supported for implementation and maintaining for free in the first year.

- **Exclusions:** Some other functions such as Intelligent Recommend System, Data Analysis for the restaurant owner, Search Engine Optimization will not be supported.

## 1.2 Describe all functional and non-functional Requirements of the desired system. Draw a use-case diagram for the whole system

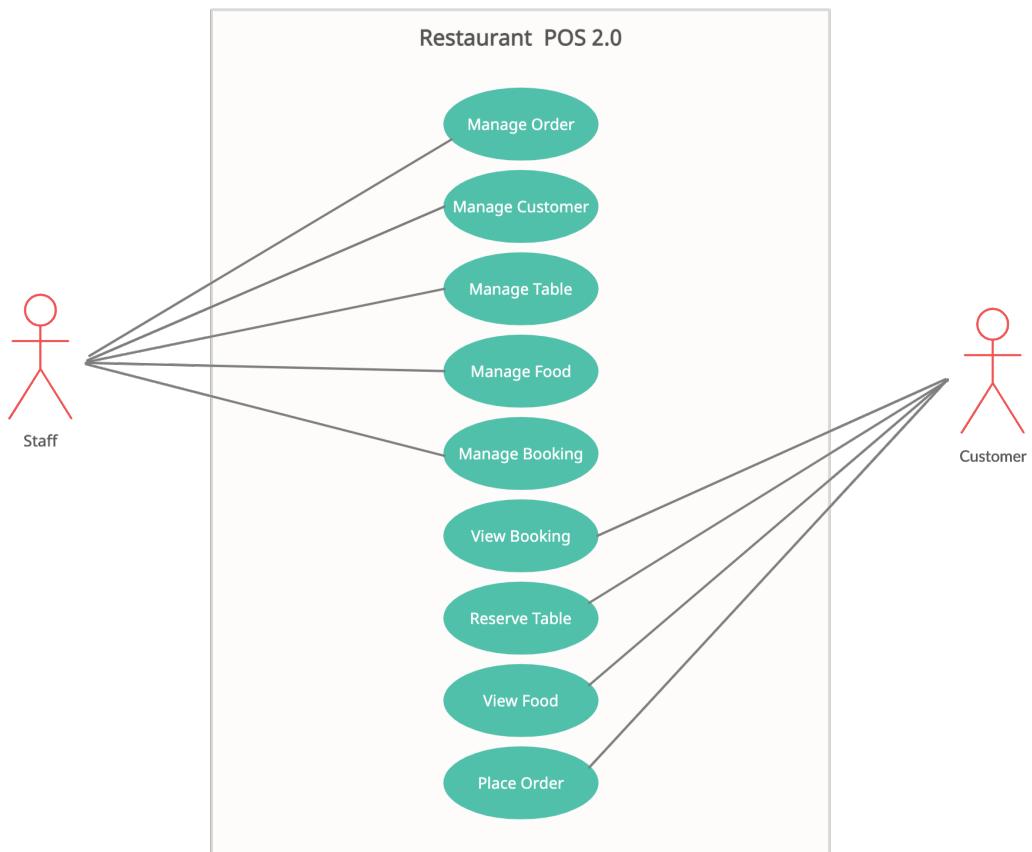
### 1.2.1 Functional requirements

- Customers are allowed to reserve tables before coming to the restaurant
- Customers shall select food from the menu
- Customers can confirm the order by a button on the app
- Customers are able to choose to pay by cash or credit card
- Customers receive information when the order is confirmed and ready
- Customers can indirectly contact the clerk
- Customers are able to update information, give feedback or cancel the transaction
- Clerk will receive orders after the customers click on the confirm button
- Clerk should send notifications to the customers if some problems occur
- Clerk can indirectly contact the customers
- Staffs can Add, Delete, Edit products or change the restaurant information
- The system allow the customers to add product to cart while they are exploring the restaurant
- The restaurant manager can manage daily/weekly/monthly report such as reports of feedbacks, order trends and income
- Restaurant owner are able to get the statistics of their restaurant
- Restaurant manager is able to add, edit, remove staffs
- Customer can order food without logging in

### 1.2.2 Non-functional requirements

- The application has user-friendly interface and easy to use for customers. In details, customers just need 4-5 minutes to get used to the application
- Restaurant clerks and managers can use the application after 1 training hour.
- The capability of transactions is 300 orders per day
- The application is able to load over 100 orders in just 2 - 3 seconds
- Customers' information must be also protected
- Restaurant data must be protected
- The application can be launched on any platform
- The application is extendable to use in multiple restaurants in the future

### 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





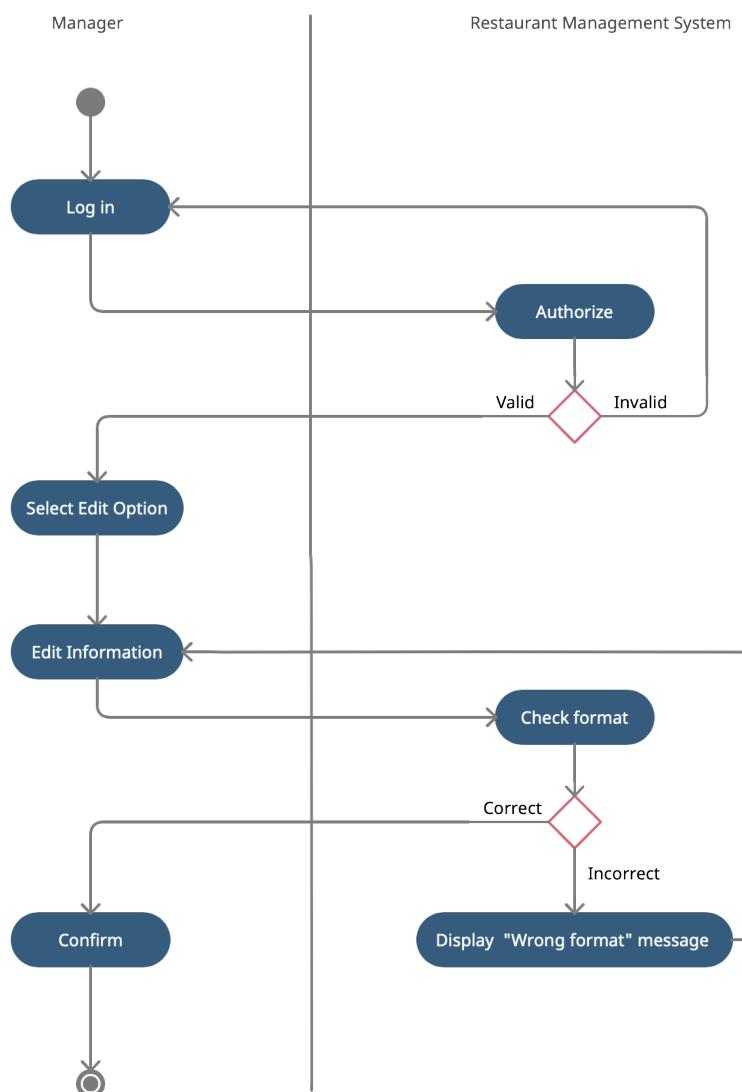
Use Case Name	Order Management
Use-case overview	For admin to manages all the bookings, food order or reserve table
Actor(s)	Admin (Primary)
Preconditions	User must be authenticated
Post conditions	All orders manages successfully
Success Scenario	1 - User select the option "Order Management" at the Management Page 2 - Website redirect to the Order Management Page and prompts to the user a detailed option menu. 3 - User select an option. 4 - User successfully manages the order.
Special Requirement:	1 - System response to the user within 30 seconds. 2 - Language internationalization text displayed.
Technology and Data Variations	1 - Technology and data variation is not applicable
Post conditions	All orders manages successfully
Open Issues:	Person has not the internet access

## 2 System modelling

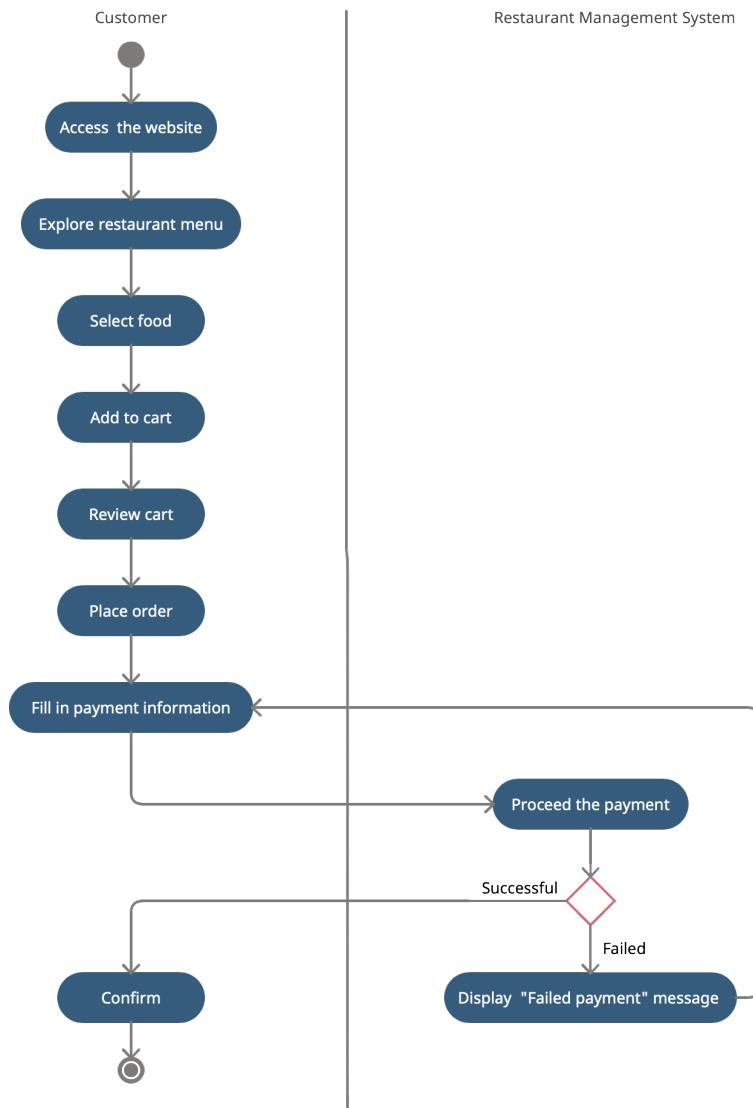
### 2.1 Task 2.1

**Require:** Draw an activity diagram to capture Major (not all) functional Requires of the desired system.

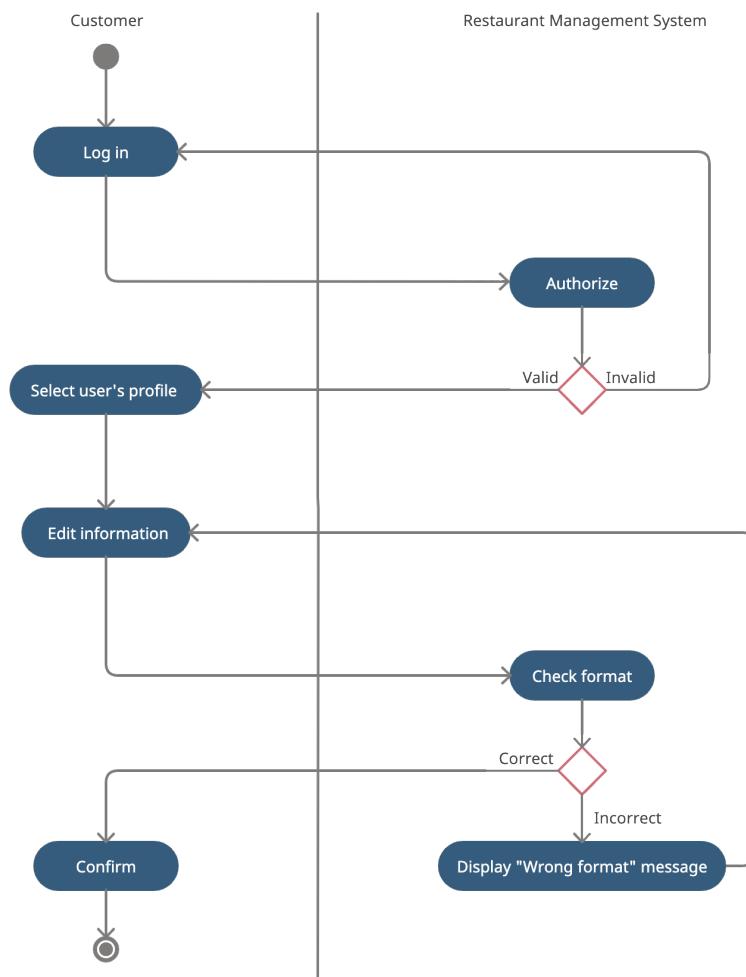
#### 2.1.1 Staffs can add, delete, edit products or change the restaurant information



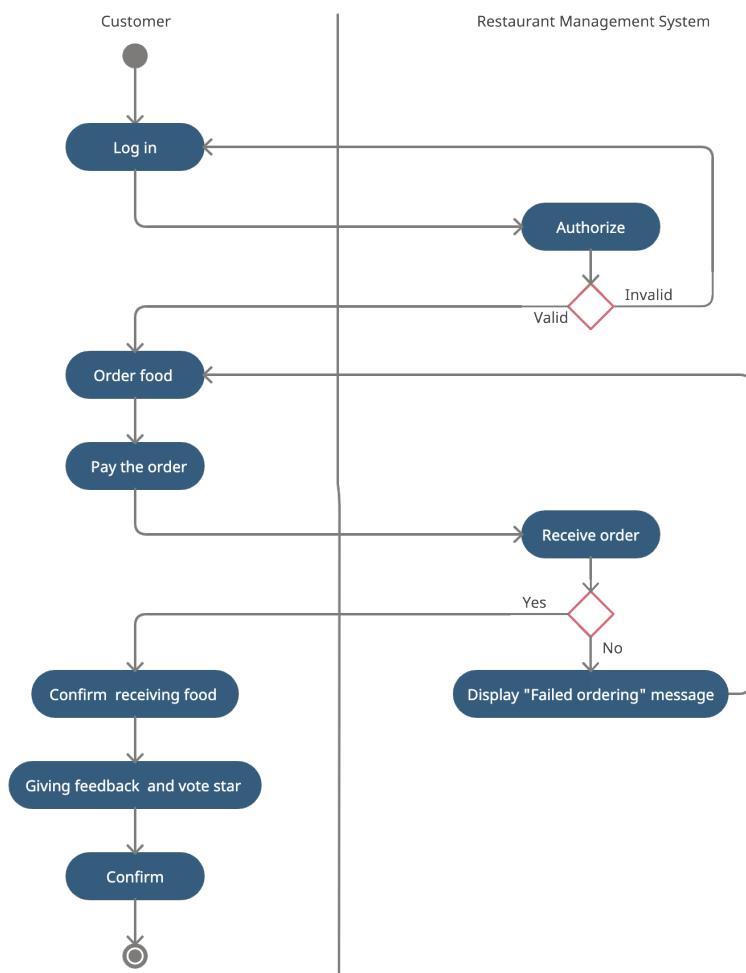
### 2.1.2 Users can view, add and order food



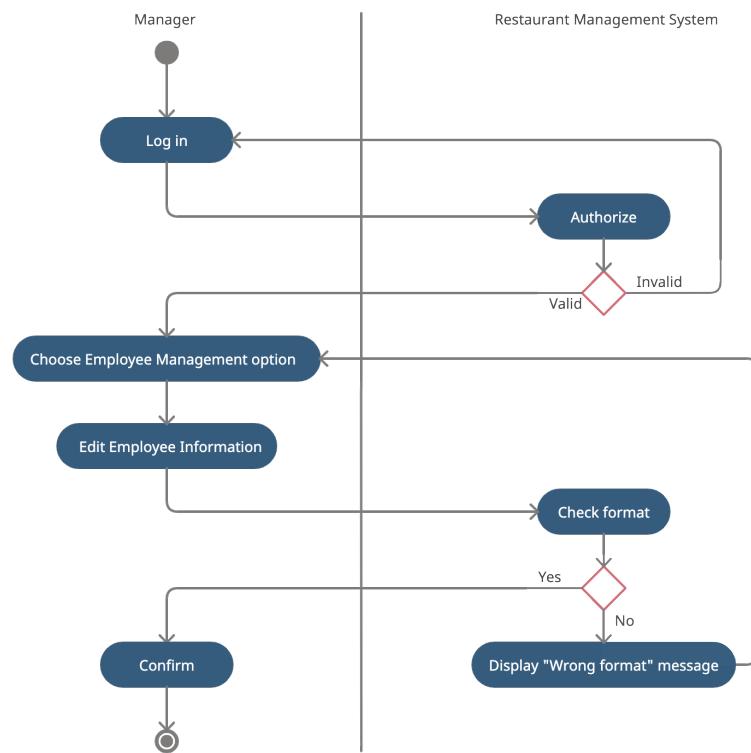
### 2.1.3 Users can add and update their information



#### 2.1.4 Users can give the product feedback

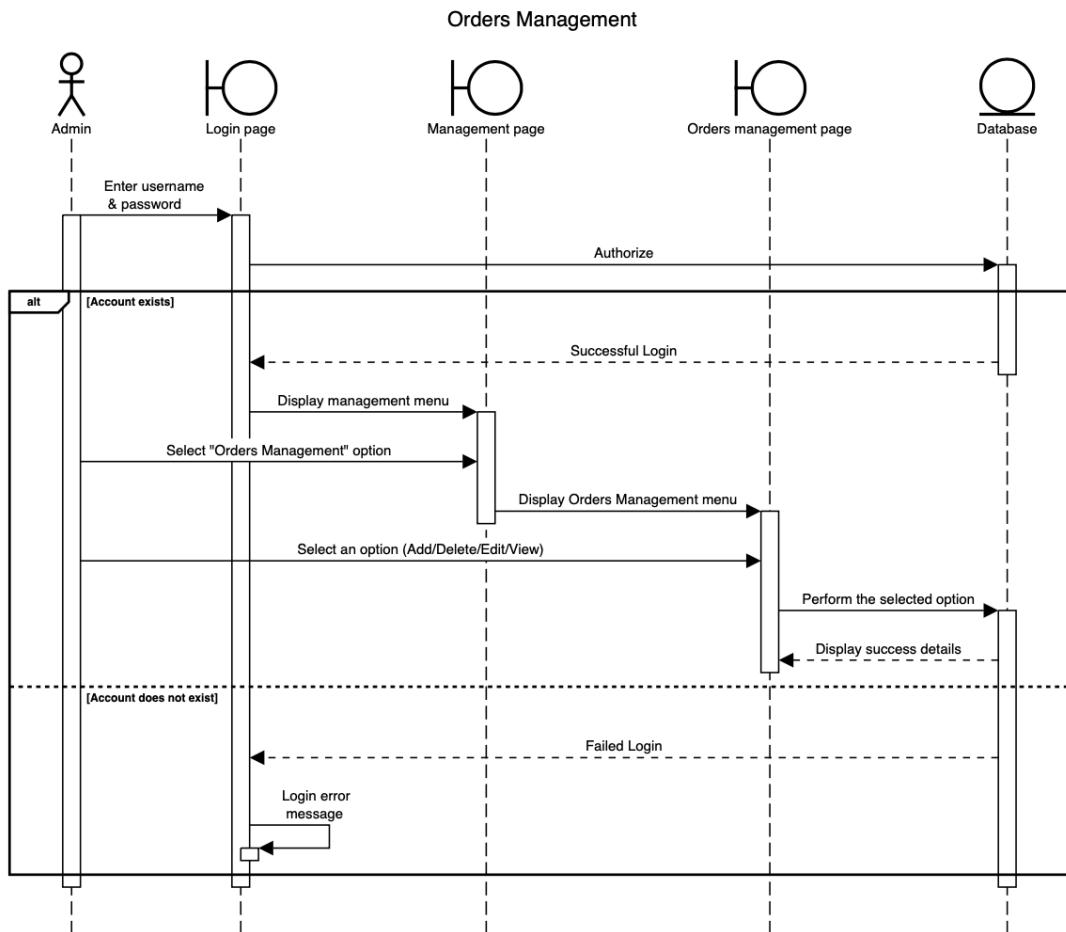


### 2.1.5 Employee Management



## 2.2 Task 2.2

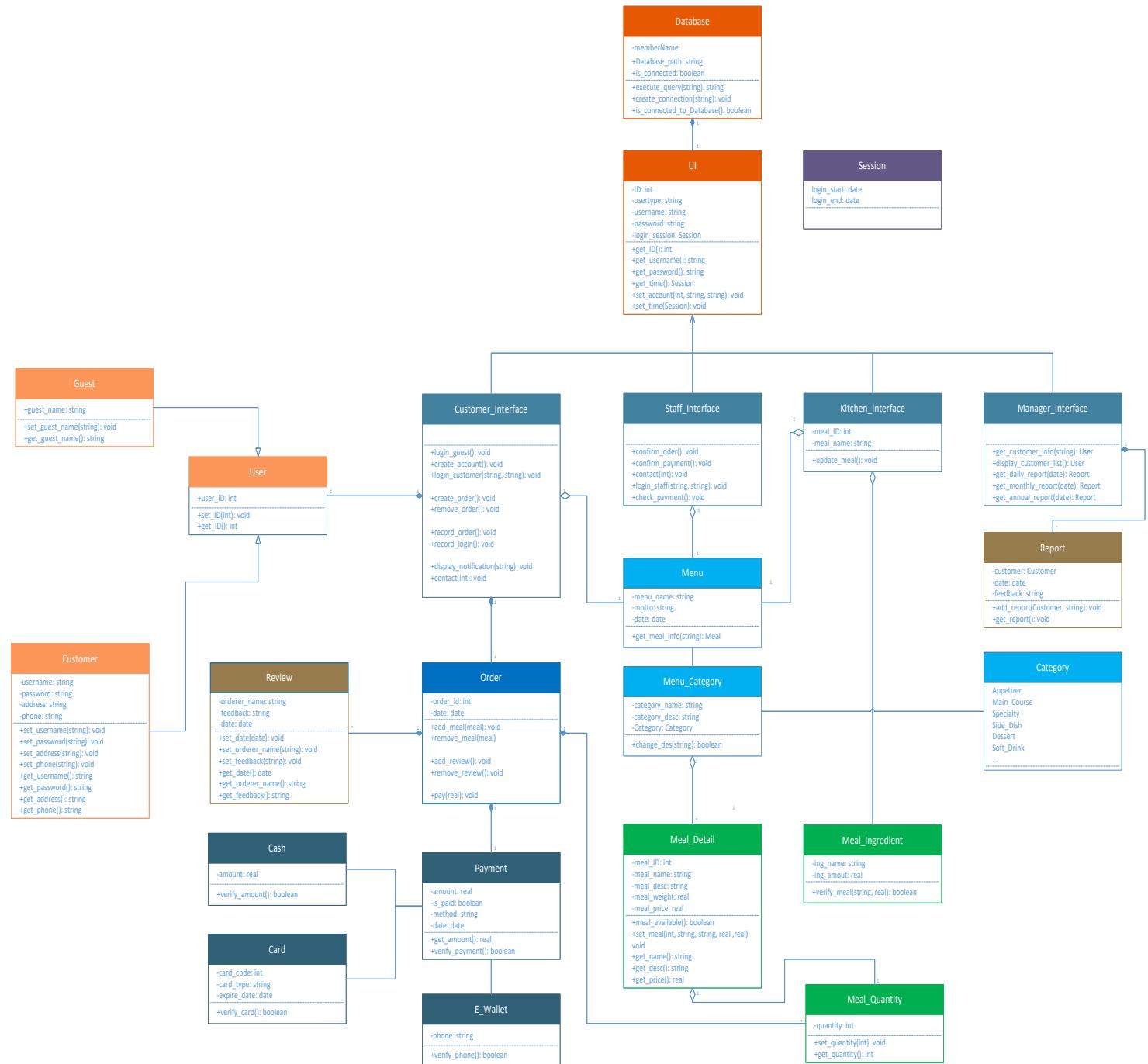
**Require:** Draw a sequence diagram for use-case in Task 1.3.



## 2.3 Task 2.3

**Require:** Draw a class diagram.

Note: If the text is too small, please zoom in to see it clearly because this figure is a .pdf file.



### 3 Architecture design

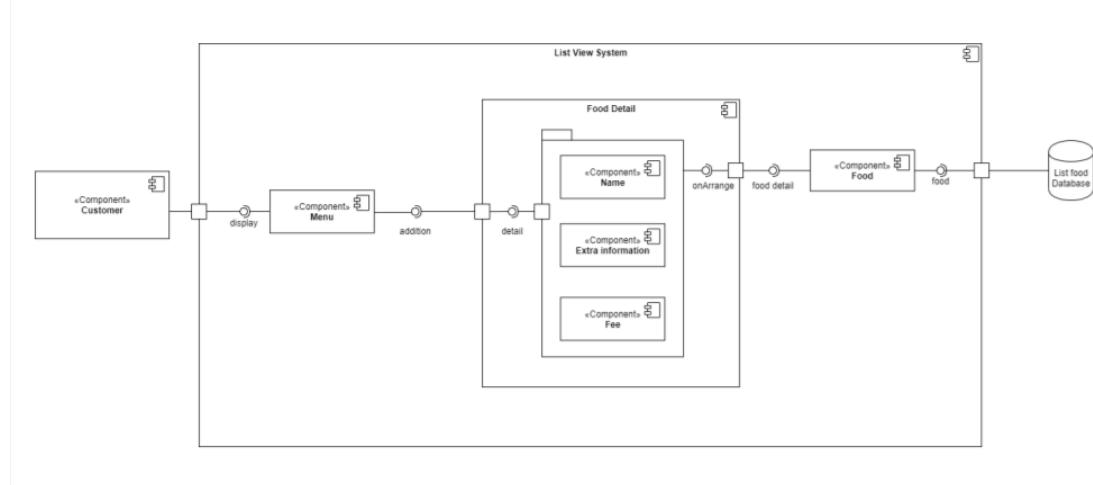
#### 3.1 Task 3.1

**Require:** Describe an architectural approach you will use to implement the desired system.

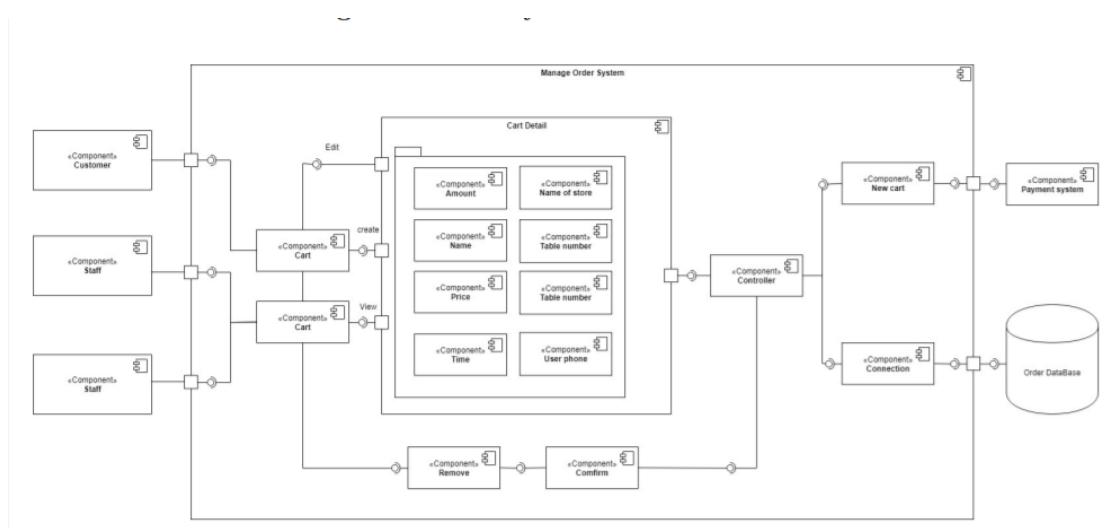
- Users access the website via the domain name owned by the restaurant. There are two sub-domains for 2 main types of users: staff members and customers. Users' devices can be laptop.
- The resources in host server include html, javascript and images to answer requests from users' browsers. All the inbound request will be redirected to https request. All the outbound request will be https response.
- The back-end service is built with 4 main services: list view, payment, order, reserve table. These services will request to a database cluster to get needed data then process this data and return result to requested client.

#### 3.2 Task 3.2

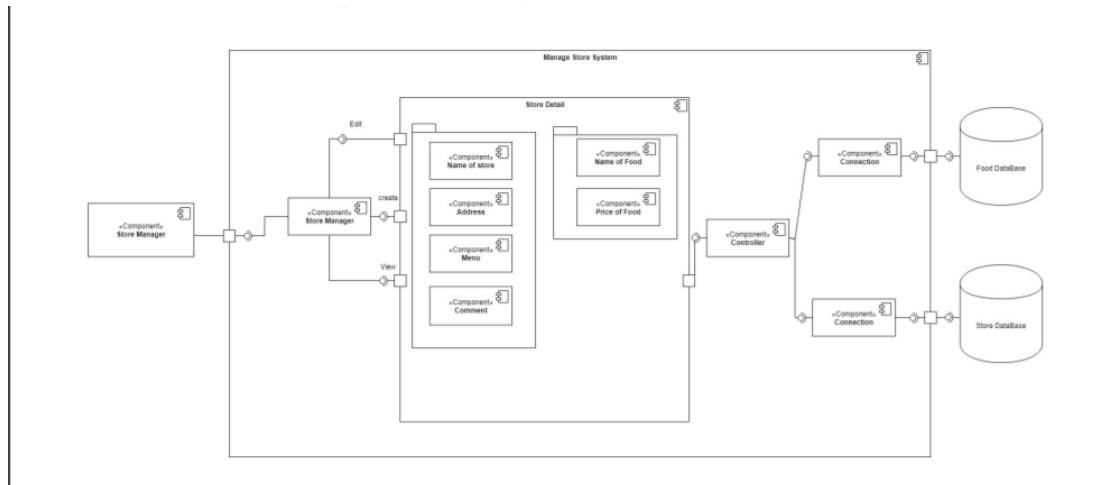
**Require:** Draw an implementation diagram for Major (not all) functional Requires.



List View



Manage Order



Manage Store

## 4 Implementation - Sprint 1

### 4.1 Task 4.1

**Require:** Setting up. The team creates an online repository (github, bitbucket, etc) for version control.

Github repository: <https://github.com/nhantantran/resPOS2.0-PHP>

### 4.2 Task 4.2

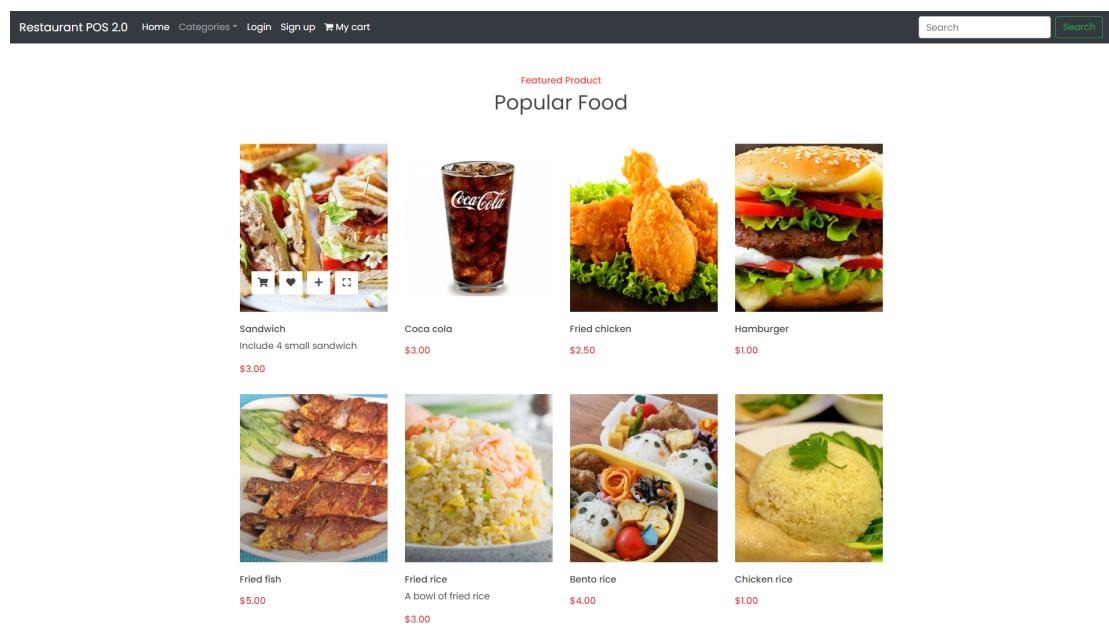
**Require:** Adding documents, materials and folders for Requirement, System modelling and Architectural design. Use the selected version control system to report the changes to these files/folders.

### 4.3 Task 4.3

**Require:** Implement a Minimum Viable Product (MVP) for the menu screen in Figure 2 and demonstrate the result. MVP means that do the least to be able to demonstrate. That means at this stage, no need for a database to store all menu items, customers, etc. Data can be hard coded in code files.

#### 4.3.1 Customer side

Customers are able to order food without logging in. The user interface is as follow



Restaurant main menu

Customers can use category food filter by manipulating the **Category** button of the header bar



The screenshot shows the homepage of the Restaurant POS 2.0 system. At the top, there is a navigation bar with links for Home, Categories, Login, Sign up, and My cart. A search bar is also present. On the left, a vertical dropdown menu lists categories: Bread, Chicken, Drink, Fast food, Fish, and Rice. The main content area features a "Featured Product" section titled "Popular Food" with three items: Sandwich, Fried chicken, and Hamburger. Each item has a small image, a name, a description, and a price. The "Sandwich" item includes a note: "Include 4 small sandwich".

Product	Description	Price
Sandwich	Include 4 small sandwich	\$3.00
Fried chicken		\$2.50
Hamburger		\$1.00

Fast food filter

After clicking **Add to cart** button, customer can check their current cart by navigating **My cart** button, user can also delete a food from their cart by clicking the **Bin** button. Finally, they input their information and press **Check out** button

The screenshot shows the customer cart page. At the top, there is a "Continue Shopping" link and a "Search" bar. The main area is divided into two sections: "Shopping cart" on the left and "Card details" on the right. The "Shopping cart" section lists four items: Sandwich, Coca cola, Fried chicken, and Hamburger, each with a quantity of 1 and a price of \$3, \$3, \$2.5, and \$1 respectively. The "Card details" section is a form for payment, including fields for Card type, Name on card, Card number, Expiry date, CVV, Full name, Address, and Phone number. Below the form, there is a summary table with Subtotal (\$9.5), Shipping (\$0.00), and Total(incl. taxes) (\$9.5). A "Checkout" button is located at the bottom right of the form.

Customer cart



User can also search for food by using search form on the navigation bar



Featured Product  
Popular Food



Coca cola  
\$3.00



Search function

#### 4.3.2 Manager side

The main menu of the manager is quite the same as those of customer, but the **delete** button is added. Manager can delete a product from the menu by clicking delete button



The screenshot shows a web browser window titled "Admin Panel" with the URL "localhost/restaurant\_pos/admin/manage.php". The page displays a grid of six food items under the heading "Popular Food". Each item has a "Delete" button. The items are:

- Big chicken: A very big chicken, \$2.00
- Sandwich: Include 4 small sandwich, \$3.00
- Coca cola: \$3.00
- Fried chicken: \$2.50
- Grilled fish: \$3.00
- Rice: \$2.00

Admin menu

Manager is able to add new product or add new category

The screenshot shows a web browser window titled "Admin Panel" with the URL "localhost/restaurant\_pos/admin/addingFood.php". It contains two main sections:

- Adding a new food:** A form with fields for Item ID, Price, Category (set to Bread), Item name, Description, and a link to the item image. A blue "Add food" button is at the bottom.
- Adding a new category:** A form with a field for Category name and a blue "Add category" button.

Adding new food

Manage can also see all orders. The orders are in descending order by time, the oldest order is on top of the page. When an order is done, manage click **Done** button to delete it



The screenshot shows a web browser window titled "cốc cốc" displaying the "Admin Panel". The URL is "localhost/restaurant\_pos/admin/orders.php". The page lists two orders:

**Order 772016:**  
Name: Tan  
Address: 497 Hoa Hao, Q10  
Phone number: +84989325783  
Date created: 2021-12-07 12:08:17

**Shopping cart:**

	Coca cola	2
--	-----------	---

**Done**

**Order 558163:**  
Name: Hang  
Address: 497 Hoa Hao, Q10  
Phone number: +84989325783  
Date created: 2021-12-07 12:08:36

**Shopping cart:**

	Sandwich	1
	Fried chicken	3

**Done**

Viewing orders



## 5 Implementation - Sprint 2

**Require:** Implement the MVP for screen showing in Figure 2 and Figure 3.