

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



Software Engineering

Assignment (Semester 211, Duration: 06 weeks)

Restaurant POS 2.0

Advisor: Assoc. Prof. Bui Hoai Thang

Class: CC03

Group LNUV: Hoang Nhu Ngoc - 1911697
Vo Khue Tam Uyen - 1852858
Nguyen Thuy Khanh Linh - 1952319
Pham Le The Vinh - 1915948
Nguyen Thanh Hieu - 1852369

HO CHI MINH CITY, SEPTEMBER 2021



Contents

1 Member list & Workload	2
2 Task 1: Requirement Elicitation	3
2.1 Task 1.1	3
2.1.1 Who are relevant stakeholders?	3
2.1.2 What are expected to be done?	3
2.1.3 What are the scope of the project?	3
2.2 Task 1.2	4
2.2.1 Describe all functional and non-functional requirements of the desired system	5
2.2.2 Draw a use-case diagram for the whole system	5
2.3 Task 1.3	6
2.3.1 Draw use-case diagram of one specific feature	6
2.3.2 Describe the use-case using a table format	7
3 Task 2: System Modelling	8
3.1 Task 2.1: Activity Diagram	8
3.2 Task 2.2: Sequence Diagram	10
3.3 Task 2.3: Class Diagram	11
4 Task 3: Architecture Sesign	13
4.1 Task 3.1: Architectural Approach	13
4.2 Task 3.2: Implementation Diagram	14
5 Task 4: Implementation – Sprint 1	19
5.1 Setting up - Create an online repository for version control.	19
5.2 Implement a Minimum Viable Product (MVP) for the menu screen	20
5.2.1 Menu Interface	20
5.2.2 Payment Interface	20
5.2.3 Bonus: Admin Interface	21
5.2.4 Bonus: API and Database	21
5.2.5 Instruction	22



1 Member list & Workload

No.	Full name	ID	Effort (%)
1	Hoang Nhu Ngoc	1911697	20
2	Vo Khue Tam Uyen	1852858	20
3	Nguyen Thuy Khanh Linh	1952319	20
4	Pham Le The Vinh	1915948	20
5	Nguyen Thanh Hieu	1852369	20



2 Task 1: Requirement Elicitation

2.1 Task 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?

2.1.1 Who are relevant stakeholders?

Stakeholders	Responsibilities
Customer	<ul style="list-style-type: none">– Order food and drinks with a suitable payment method for charging.– Reserve table for a specific time.– Send feedback and evaluation.
Clerk	<ul style="list-style-type: none">– See and confirm customers' orders and selected payment methods– Receive and collect feedback from customers.– Send orders to the kitchen without direct-contact.
Chef	<ul style="list-style-type: none">– See customers' confirmed orders to prepare them.– Manage the availability of dishes (Disable unavailable ones).
Manager	<ul style="list-style-type: none">– Edit menu (Add or remove dishes, update prices and discounts, etc.)– Record and confirm customers' table reservations.– Manage information of any order, reservation, customer at any time.
Accountant	<ul style="list-style-type: none">– See and check customer's payments (method, amount, discount,...).– Manage the income in a specific period of time.
Administrator	<ul style="list-style-type: none">– Maintain and revise technical documentation.– Oversee all day to day POS configurations and integrated applications– Ensure all the versions, policies are accurate and up to date.

2.1.2 What are expected to be done?

- Convenience in food ordering in COVID-19 time.
- Enhancing popularity through online advertisement.
- Effective menu and financial management (easy to edit: add new meal, remove old meal, etc., more accuracy in profit calculation).
- Better customer management.

2.1.3 What are the scope of the project?

1. Introduction

This POS System is being undertaken by HiHiLand restaurant system in order to support the management of food ordering and table reservation. It helps customers contact restaurant managers more easily, have more accurate information from the restaurant and make the ordering and reservation process more efficient and convenient.

2. Project Scope

The project will include food ordering, table reservation, member management, feedback receiving and will be published on many platforms and devices. Customers can register using Google account, Facebook account or phone number. On the server side, restaurant managers can manage all information about customers, orders, bills and earnings of each restaurant per period of time.



3. Project Deliverable

(a) Research Phase

- Customer research (What customers expect at online service, features of traditional food ordering that they want us to improve in POS system, what they expect to have when becoming members of restaurant,...)
- Elaboration on how online management works for all staffs of restaurant
- User's behaviour research (To develop friendly UI)

(b) System development phase

i. Front end

- The menu screen for customer
- Customer's services interfaces (Food Order, Table reservation, Payment, Ordering and Booking history, etc.)
- Restaurant management Interface (food and drinks management etc.)

ii. Back end

– Management

- + Customer Management
- + Order and Table reservation management.
- + Food and Drinks categories and availability.
- + Feedback and Evaluation receiving.

– Customers

- + Payment methods and processing
- + Ordering/Booking processing
- + Cart management
- + Feedback and Evaluation input

4. Project Acceptance Criteria

- The requirement elicitation must be done before 26th September 2021
- The system modelling must be done before 10th October 2021
- The project must have an architecture design before 17th October 2021
- A MVP release must be available before 4th November 2021

5. Project Exclusions

No support for full service sit down restaurant.

6. Project Constraints

- The project must not need any budget.
- The project must be completed before 2nd December 2021.

2.2 Task 1.2

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

2.2.1 Describe all functional and non-functional requirements of the desired system

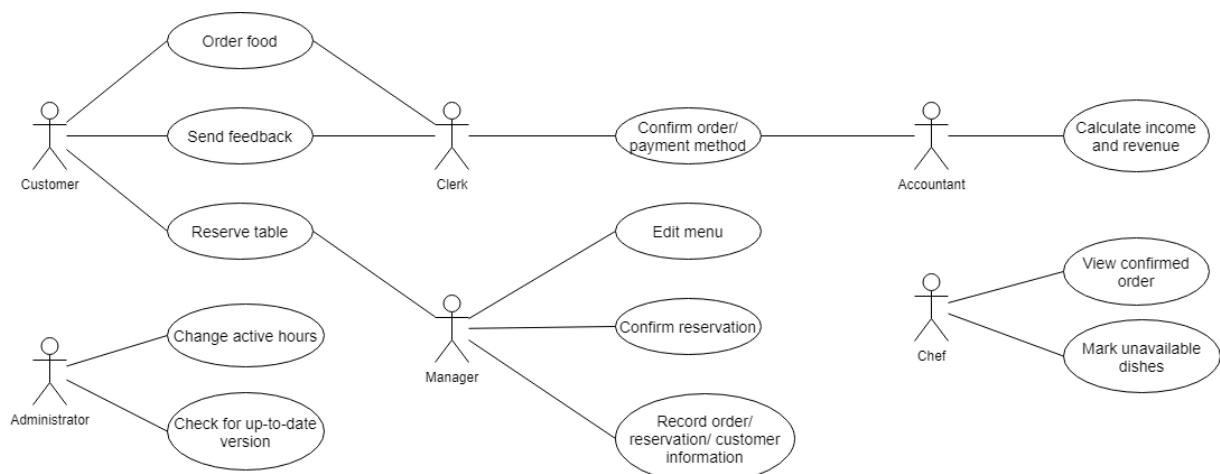
1. Functional requirements

- Customers can see the menu and select food and drinks from it.
- Customers can place an order.
- Customers can select payment methods and pay for the order.
- Customers can send feedback and evaluation.
- Customers can select and reserve tables.
- Clerk can confirm customers' orders.
- Manager can edit the menu, add or remove dishes and update prices.
- Manager can confirm and record table reservations.
- Manager can check detailed information about orders, reservations any time.
- Chef can see confirmed orders and mark the completed ones.
- Chef can mark the unavailable dishes.
- Accountants can calculate daily total income and generate income reports.
- IT Administrator can change active hours of the system.
- IT Administrator can check for up-to-date versions.

2. Non-functional requirements

- Allow non-direct contact between Clerks and Customers, Clerks and Chefs.
- Use Web technology and QR code
- Usable from a mobile, tablet or a computer and in variety of platforms.
- Expandable to use in multiple restaurants
- Availability: on from open time to close time of the restaurant

2.2.2 Draw a use-case diagram for the whole system

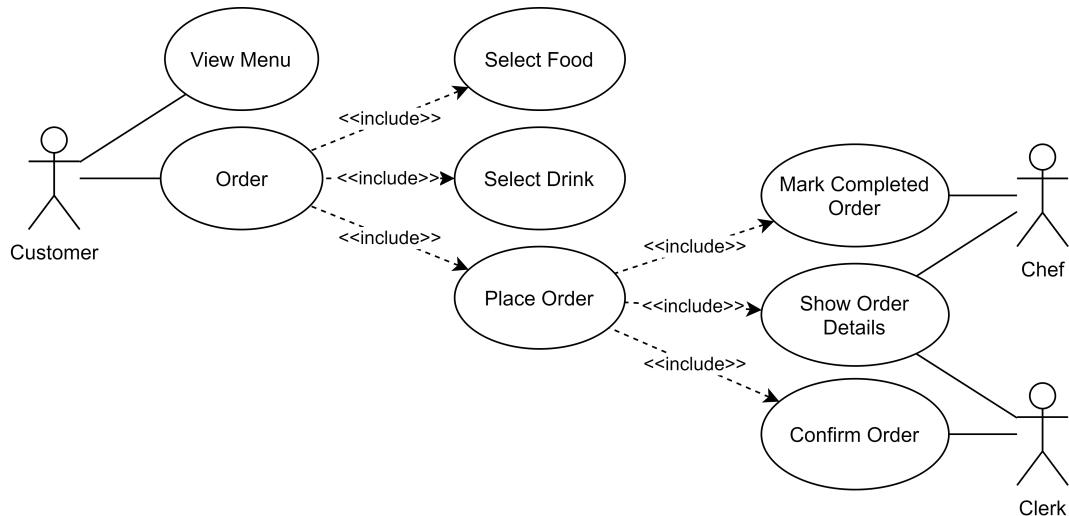


2.3 Task 1.3

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.

2.3.1 Draw use-case diagram of one specific feature

Feature chosen: Order food





2.3.2 Describe the use-case using a table format

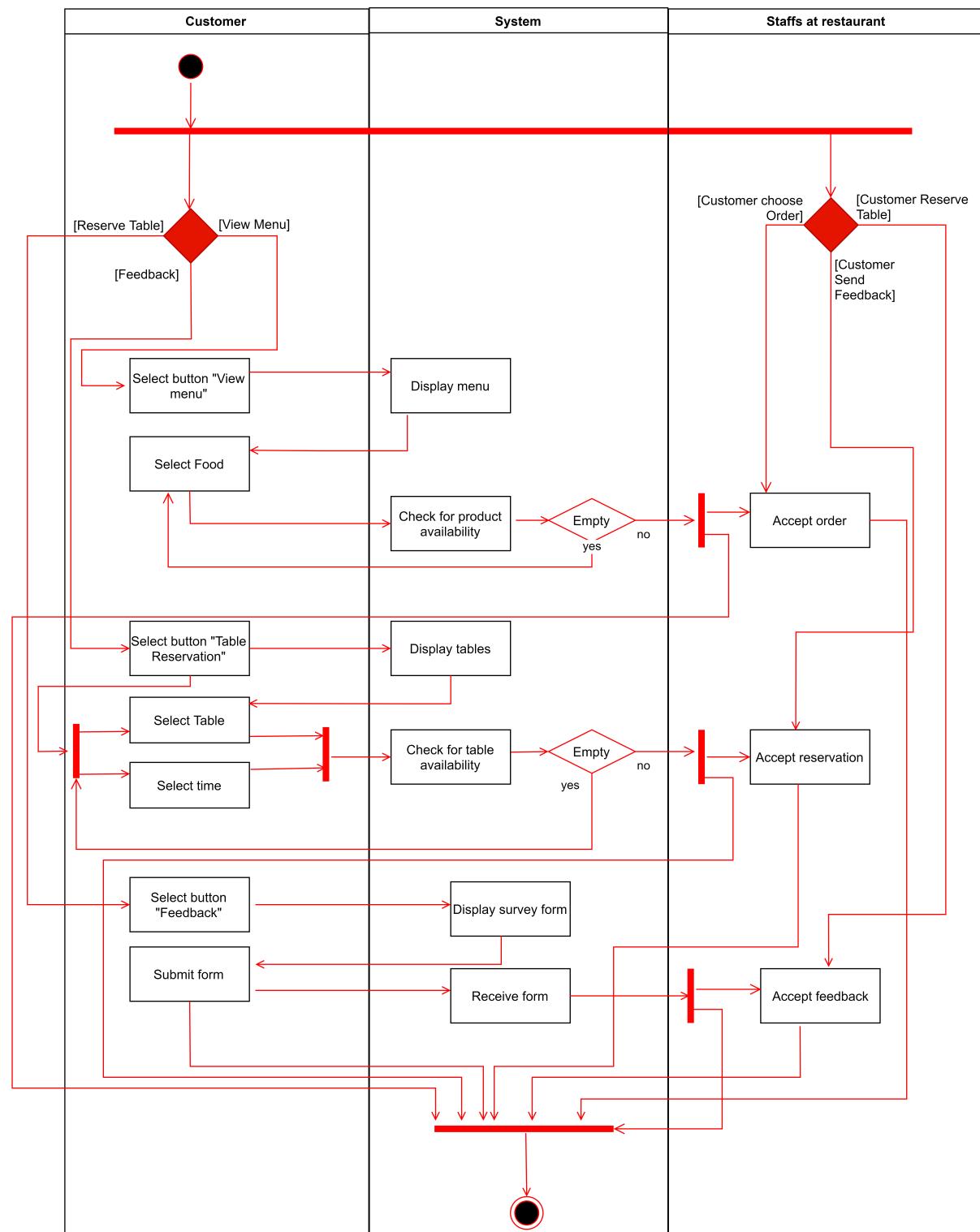
Use-case name	Order food
Actors	Customer, Chef, Clerk
Description	The customer wants to order food from the restaurant through the system
Preconditions	The customer is at the Menu web page of restaurant's website
Post-conditions	After the customer's order is completed, the system will reset and clean up the order cart.
Normal Flow	<ol style="list-style-type: none">1. On the Menu page, the system will show the detailed menu and special recommendations.2. Customer selects food from the menu or recommendations.3. Customer can recheck the order by clicking the food cart icon.4. The website directs customer to the order cart to see what they have ordered.5. Customer places the order by clicking the "Order" button.6. Clerk confirms the order and sends it to the chef.7. Chef receives the confirmed order and prepares it. After finishing, chef will mark the order completed.
Alternative Flow	<i>Alternative 1:</i> at step 3 3a. If the customer has already selected food and drinks, cart page will show all the selections and let customer place order. 3b. If the customer has not selected anything, the message is sent out and the "Order" button is disabled.
Exceptions	<i>Exception 1:</i> at step 2: 2a. If the food is unavailable, the system must disable it to prevent customer's selection from the Menu page.



3 Task 2: System Modelling

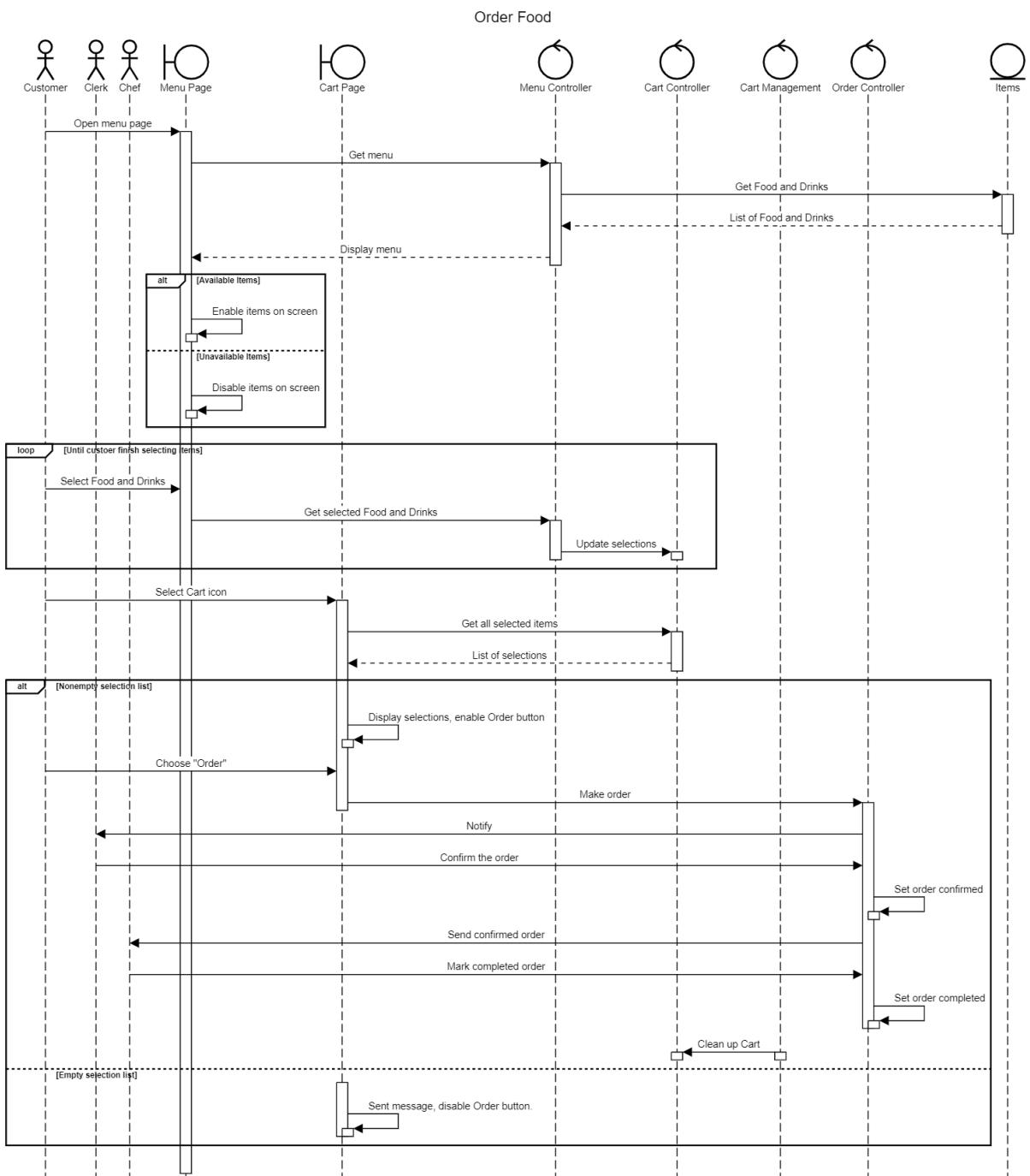
3.1 Task 2.1: Activity Diagram

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



3.2 Task 2.2: Sequence Diagram

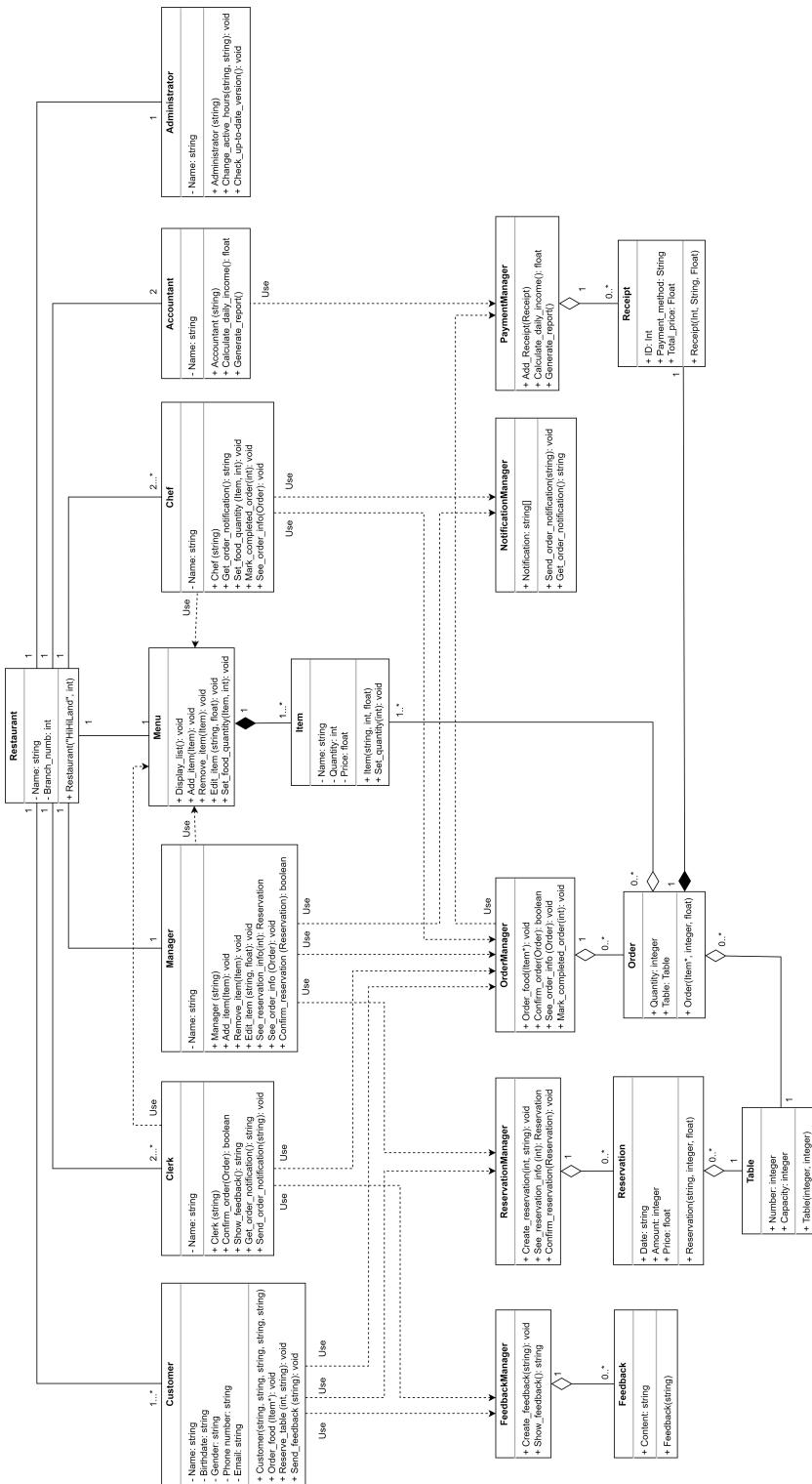
Draw a sequence diagram for use-case in Task 1.3.





3.3 Task 2.3: Class Diagram

Draw a class diagram.





4 Task 3: Architecture Sesign

4.1 Task 3.1: Architectural Approach

Describe an architectural approach you will use to implement the desired system

We decided to use **Layered Architecture** to organize the Restaurant POS System into multiple layers with related functionality associated with each layer. According to the functions of the system, there will be 4 layers:

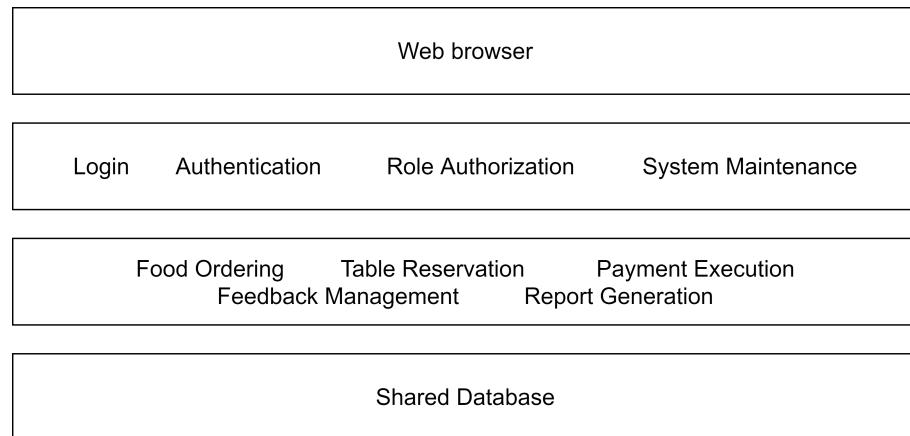
- The first layer will be the Web browser that the clients use to interact with the system.
- The second layer will include:

Component	Description
Login and Authentication	Helps the clients login to their respective accounts and verifies the identity of the clients.
Role Authorization	Allows customers and restaurant staff to have the correct functions that are allowed for them.
System Maintenance	Allows only system administrators to monitor and check the system activities and technical problems.

- The third layer will include 5 components of the system:

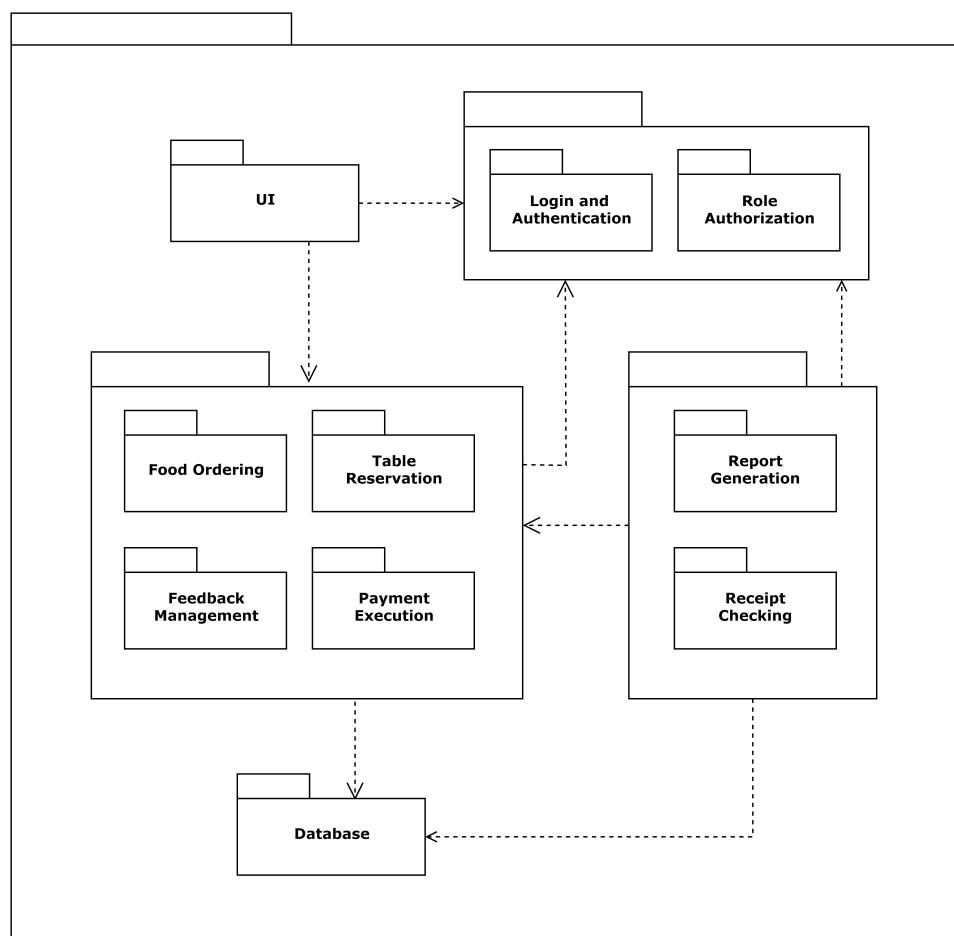
Component	Description
Food Order	<ul style="list-style-type: none">– Includes displaying menu, adding food into cart, creating order, notifying clerks and chefs about new orders.– Let only the manager edit the menu, change the items' price.
Table Reservation	<ul style="list-style-type: none">– Manages the tables in the restaurant and allow customers to select and reserve tables at a specific time.– Let the manager confirms and records the reservations.
Payment Execution	<ul style="list-style-type: none">– Processes all the payments, generates receipts for customers.– Records the daily income for report generation, if any.
Feedback Management	Collects all feedback and rating that were given by the customers and sends them to restaurant staff.
Report Generation	Allows managers and accountants to print out daily orders, income reports.

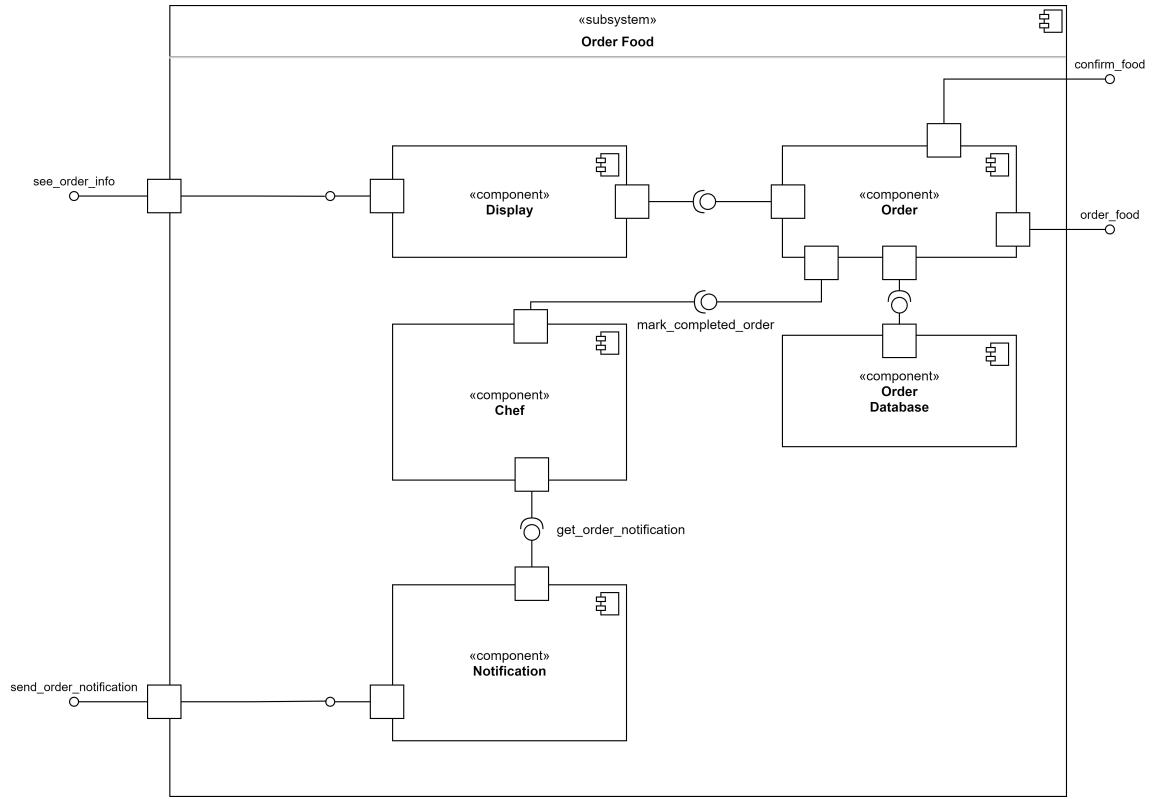
- The last layer will be for the shared system database that stores information about food and drinks, orders, tables, receipts and feedback.

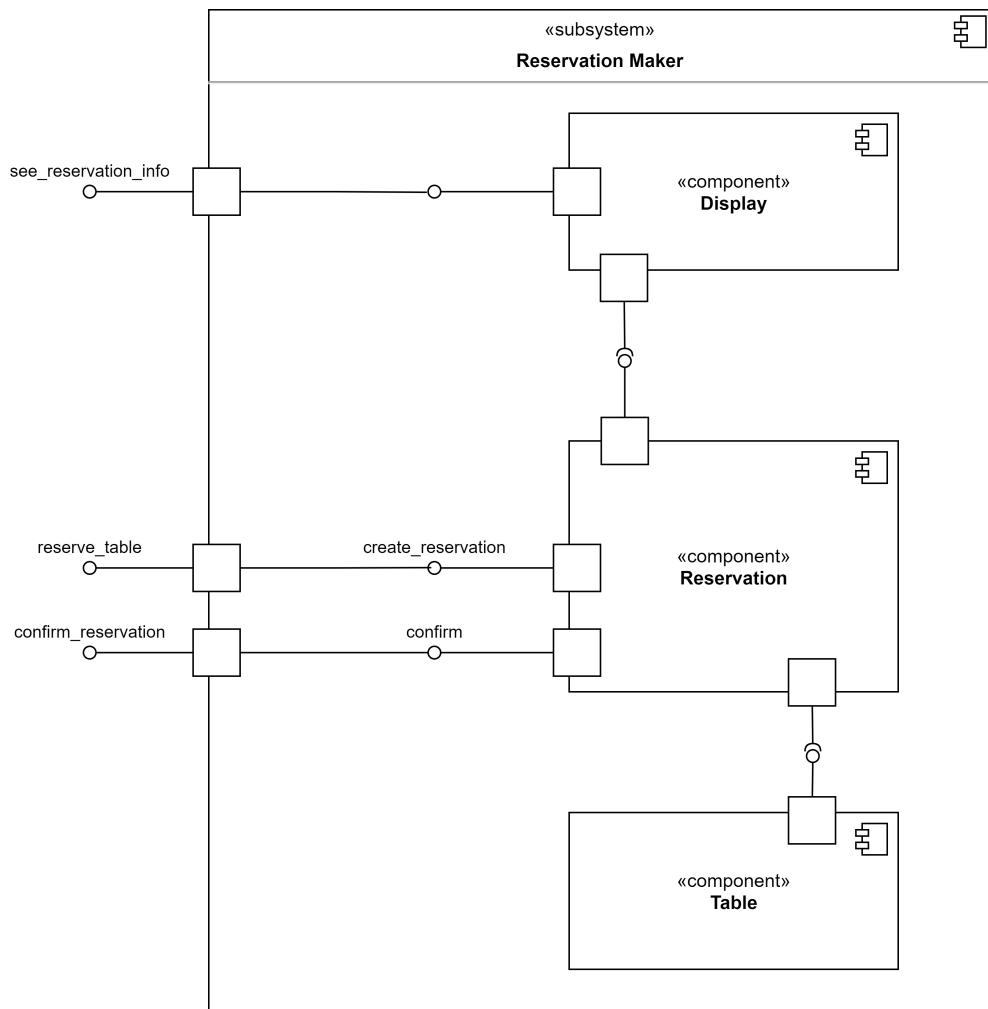


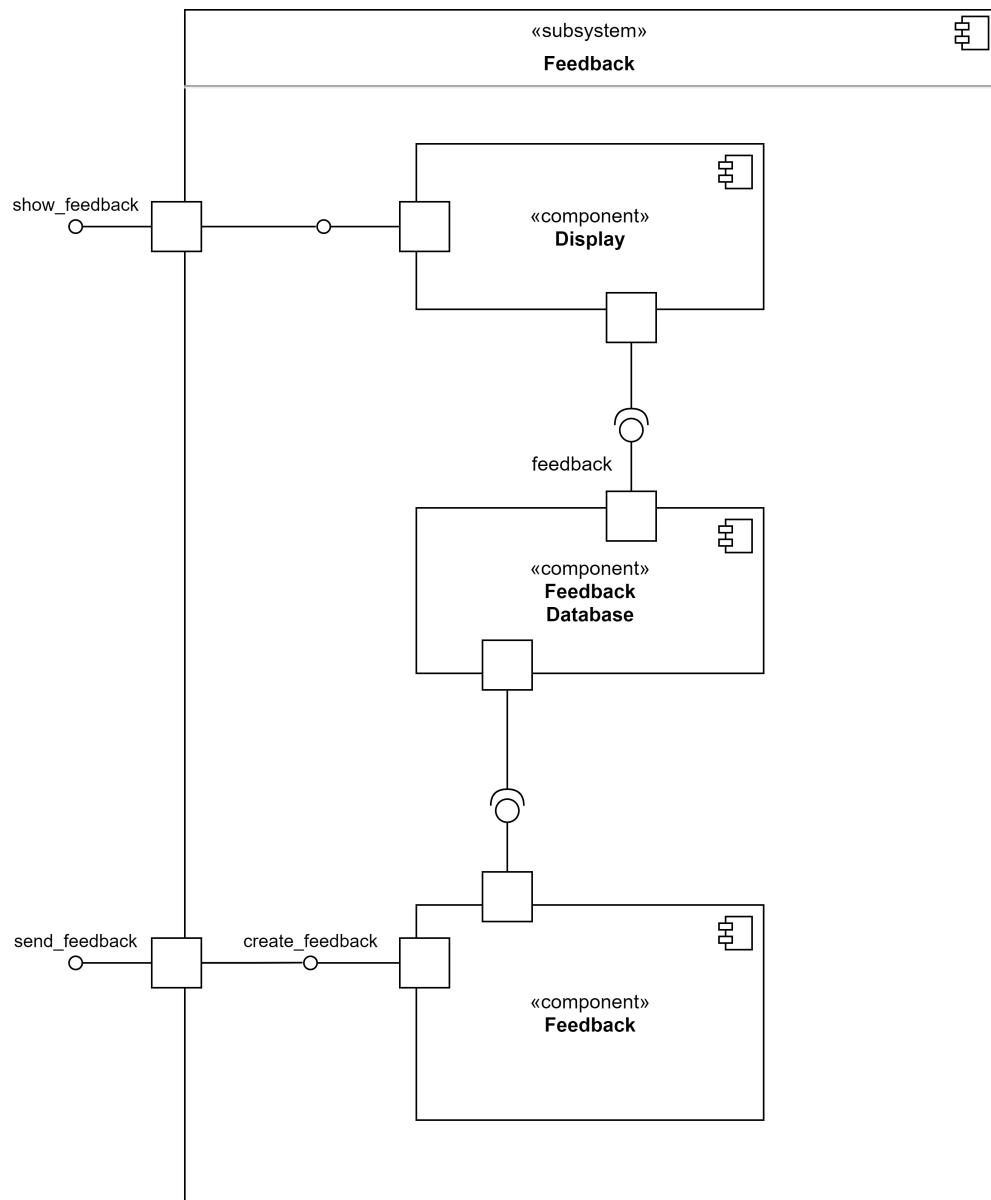
4.2 Task 3.2: Implementation Diagram

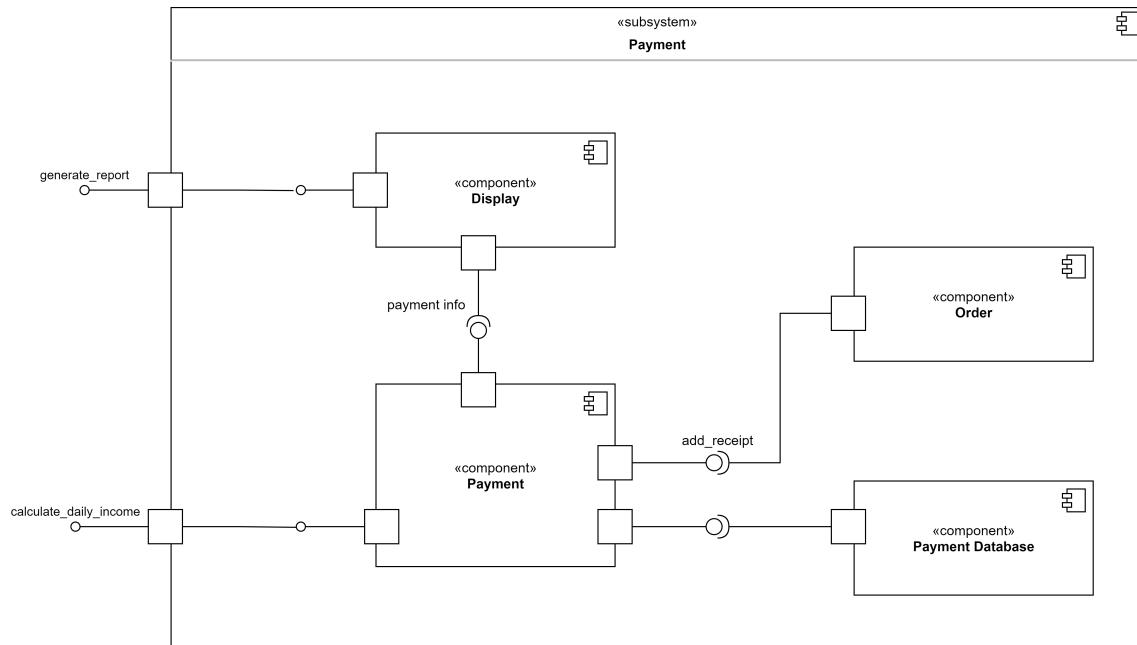
Draw an implementation diagram for Major (not all) functional requirements













5 Task 4: Implementation – Sprint 1

5.1 Setting up - Create an online repository for version control.

We decided to use **GitHub** to control our POS System Project's versions and updates.
Our GitHub link:

- Front: https://github.com/zBlueberry/POS_System.git
- API and Database: <https://github.com/tvnh321/Rest-API-For-SE-Assignment>

The screenshot shows the GitHub repository page for `zBlueberry / POS_System`. The repository is public. The main content area shows the `master` branch with 2 branches and 0 tags. A commit from `zBlueberry` dated 12-01-21 is listed, showing changes to `public`, `src`, `.gitignore`, `README.md`, `package-lock.json`, and `package.json`. The `README.md` file contains the text "Getting Started with Create React App". The right sidebar provides information about the repository, including "About" (no description), "Releases" (no releases published), "Packages" (no packages published), "Contributors" (zBlueberry and tvnh321), and "Languages" (JavaScript).

All of necessary documents, materials are uploaded. If you want to have a closer view on our project's requirements, idea and designs, you can see the reports in the **designs** folder.

Whenever there are updates or changes in files or folders, GitHub will record them as **commits**. Each will include the account and the time that change was made. Thanks to this support of GitHub, we can manage the version and control our code easier.

The screenshot shows the commit history for the `master` branch on November 30, 2021. There are three commits:

- Merge pull request #2 from tvnh321/master by zBlueberry committed yesterday. This commit is verified and has a hash of `d690e4a`.
- Fix by tvnh321 committed yesterday. This commit has a hash of `4cb772b`.
- Update 30-11-21 by zBlueberry committed 2 days ago. This commit has a hash of `380c29d`.



5.2 Implement a Minimum Viable Product (MVP) for the menu screen

5.2.1 Menu Interface

The layout contains:

- Item selections on the left with food selection tab above. Each item has Item Name, Price and Add to Cart Button.
- Cart view on the right where people can add and remove items.
- Total Price of the Cart
- Payment Button which will leads to the Payment Page
- Admin Page Button for accessing Admin Page

The screenshot displays a menu interface for cupcakes. At the top, there are tabs for 'Doughnut', 'Cupcake' (which is selected), and 'Drinks'. Below the tabs, there are six cupcake items arranged in two rows of three. Each item includes a small image, the name, price, and an 'Add to cart' button.

Item	Description	Price	Action
Blueberry Cupcake	With blueberry filling and swirls.	39,000 VND	Add to cart
Strawberry Cupcake	With strawberry filling and swirls.	39,000 VND	Add to cart
Lemon Cupcake	With lemon zest and swirls.	45,000 VND	Add to cart
Black Cupcake	With dark chocolate frosting.	30,000 VND	Add to cart
Pink Cupcake	With pink frosting.	30,000 VND	Add to cart
Matcha Cupcake	With matcha green tea frosting.	45,000 VND	Add to cart

To the right, a 'Your Cart' section shows the current contents:

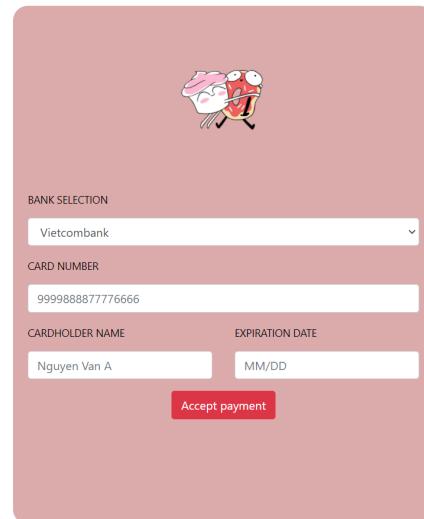
Item	Quantity	Price	Action
Blueberry Cupcake	2	78,000	Delete
Strawberry Cupcake	2	78,000	Delete
Lemon Cupcake	3	135,000	Delete

Total: **291,000 VND** [VAT 10% = 15,000 VND]

[PAYMENT](#) [GO TO ADMIN PAGE](#)

5.2.2 Payment Interface

The layout is a form for entering payment method.



5.2.3 Bonus: Admin Interface

Because this is just a MVP website. We implemented a button below Payment for easier access. In this page, Items can be add, update or remove from the Database using API.

Food code	Food name	Image	Description	Action
8	Blueberry Cupcake		Great Flavors from Blueberry and Vanilla !!!	<button>Update</button> <button>Delete</button>
9	Strawberry Cupcake		Strawberry Goodness in a Cupcake !!!	<button>Update</button> <button>Delete</button>
10	Lemon Cupcake		For Times that we need a little Sour and Sweet at ...	<button>Update</button> <button>Delete</button>
11	Black Cupcake		Feeling Edgy ? We have a cupcake for you !!!	<button>Update</button> <button>Delete</button>
12	Pink Cupcake		Black Cupcake's Sister, they are the complete oppo...	<button>Update</button> <button>Delete</button>
13	Matcha Cupcake		For All Matcha Lovers Out There !!!	<button>Update</button> <button>Delete</button>
14	Crème De La Creme		Custard, Caramelised Hard Crack	<button>Update</button> <button>Delete</button>
15	Rosewater Mist		Rosewater Glaze, Pitaschio	<button>Update</button> <button>Delete</button>

5.2.4 Bonus: API and Database

Even though API and Database are not needed for the MVP website, we implemented it so that the Admin Page can be usable.



We use Django REST Framework for API and SQLite for Database

The API includes:

- GET request for retrieving Menu Data for **Menu and Admin Page**
- POST request for creating Menu Item for **Admin Page**
- PUT request for editing Menu Item for **Admin Page**
- DELETE request for removing Menu Item for **Admin page**

5.2.5 Instruction

First, you can clone our project from GitHub to your computer. Then change direction into project directory.

Back-End

- Install Python on your local computer.
- Using cmd, run:

```
pip install -r requirements.txt
```

- Then run:

```
python manage.py runserver
```

Front-End

- Install Nodejs on your local computer.
- Install react-scripts using cmd:

```
npm install react-scripts
```

- Now, you can run the web by cmd:

```
npm start
```

- The browser will open: <http://localhost:3000/> and load the page for you.