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

SOFTWARE ENGINEERING

Project report

Restaurant POS 2.0

Lecturer: - Quản Thành Thơ
Student: - Đinh Hoàng Anh 1952553
- Nguyễn Minh Hùng 1952737
- Nguyễn Bá Minh Hưng 1952748
- Phạm Nhật Hoàng 1952703
- Phạm Hoàng Hải 1852020

HO CHI MINH CITY, SEPTEMBER 2021

Member list & Workload

No.	Fullname	Student ID	Problems	Percentage of work
1	Đinh Hoàng Anh	1952553	P1.2,P2.1	20%
2	Nguyễn Minh Hùng	1952737	P1.2,P2.2	20%
3	Nguyễn Bá Minh Hưng	1952737	P1.3,P2.2	20%
4	Phạm Nhật Hoàng	1952703	P1.3,P2.1	20%
5	Phạm Hoàng Hải	1852020	P1.1,P2.3	20%

Contents

1 Requirement elicitation

1.1 Task 1.1

1.1.1 Project Description

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. Even before the COVID-19 crisis, POS systems had gained traction across the industry. During the coronavirus pandemic, restaurants face greater peril than ever. Such systems are expected to increase business intelligence, reduce wasted effort and opportunity to scale to a large business. Our client have multiple restaurants and have a need to develop a responsive web-based POS system.

1.1.2 Project Scope

- The most basic function of this new system is to make order. It must allow customer to select food from menu and place an order, then calculate the amount and let customer make payment. To serve the widest range of customers, the system should be web-based and have QR code so that customer can open web page by scanning it, without the need to install any application. Furthermore, the system should be usable from a mobile device, a tablet device or a normal computer/laptop. The user can choose to pay by bank transfer, and allow customer to choose dine-in or takeaway option.

- POS system can handle orders. The system should allow manager to manage orders, process them and send them to kitchen.

- Another function of POS system is to manage menu. The manager should be able to add or remove option from the menu using the system. Also the system should generate reports for manager to keep track of restaurant performance.

- The final function of the system is maintenance. It should be easy for the technician to update the software and fix it in case of error.

- Furthermore, to reduce the risk of COVID exposure, the system should allow non-direct contact between Staff and Customers. Also our client has multiple restaurants, so the system should be extendable to use in multiple restaurants in the future.

1.1.3 Stakeholder

- **Customer:** People who use system to make food order.
- **Staff:** Receive the order, manage payment.
- **IT Staff:** Responsible for maintaining and upgrading the system.
- **Database Administrator:** Managing, modifying and securing the database.
- **Functional Manager:** Responsible for different departments of restaurant(finance, project management, marketing).
- **Owner:** Responsible for making important decisions.

1.1.4 Context diagram

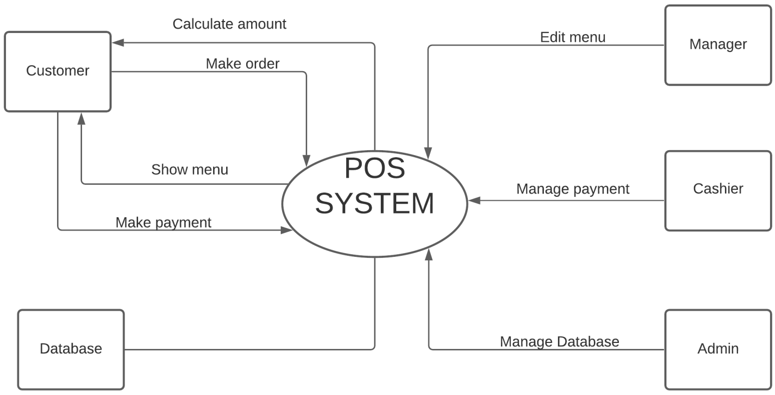


Figure 1: Context diagram for the Restaurant POS 2.0 system

1.2 Task 1.2

1.2.1 Functional requirements

- **View menu:** Display all the dishes and provide search engine for customer.
- **Food review:** A user can rate the dishes on a five-star scale.
- **Manage Profile:** Users can manage their account (edit profile, sign up, login, logout, change password, check payment history...).
- **Food Ordering:** Users can order food (A customer can choose to eat in, take away or delivered).

- **Data management:** an admin can manage all of data, (including customer's profile, modifying food list, check payment history...)
- **Payment function:** customers can add the payment method to pay for their order (cash, credit,...).
- **Food Recommendation:** Give the recommended food for each customer based on the rank of food and their previous ordering.
- **Table Reservation:** Customer can check available table and reserve table.

1.2.2 Non-functional requirements

Security requirements

- The system will deactivate automatically if user enter wrong password five times consecutively.
- Customer's information must be private.

Performance requirements

- The current transactions is about 300 orders per day.
- The response time should not exceed 4 seconds.

Usability requirements

- UI should be easy to use so that customers can finish ordering food in five minutes.
- Manager can check payment history in 5 clicks.
- The system should support blind person.
- The system should support new customers by online manual.

Availability requirements

- The website should be active 24/7 and be upgraded once per 3 months.

Environmental requirements

- The system should be implemented using Web technology and QR code, so customers will not have to install apps.
- The system should be usable from a mobile device, a tablet device or a normal computer/laptop.

Scalability requirements

- The system should be extendable to use in multiple restaurants in the future.

Localization requirements

- The system can switch among 3 popular languages (English, Vietnamese, French) and 3 unit currency (Dollar, VNĐ, Euro)

1.2.3 Use-case diagram for the whole system

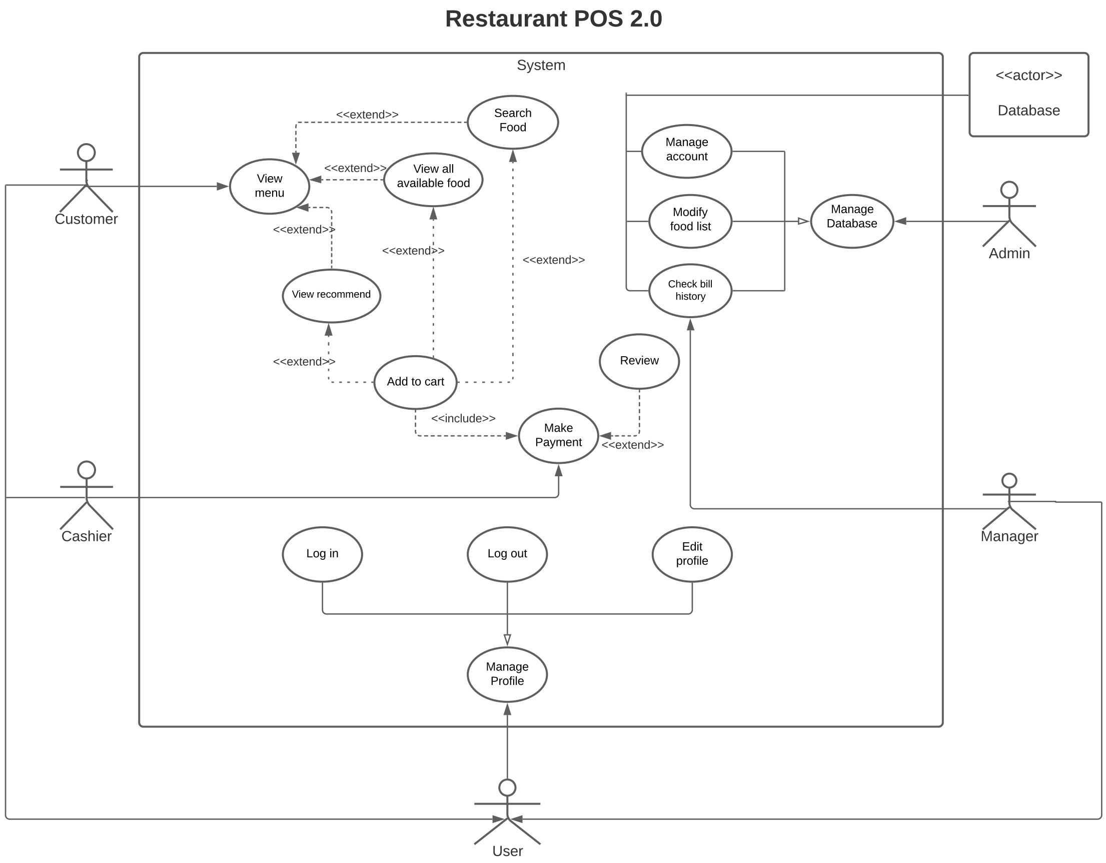


Figure 2: Use-case diagram for the Restaurant POS 2.0 system

1.3 Task 1.3

1.3.1 Feature: Checkout

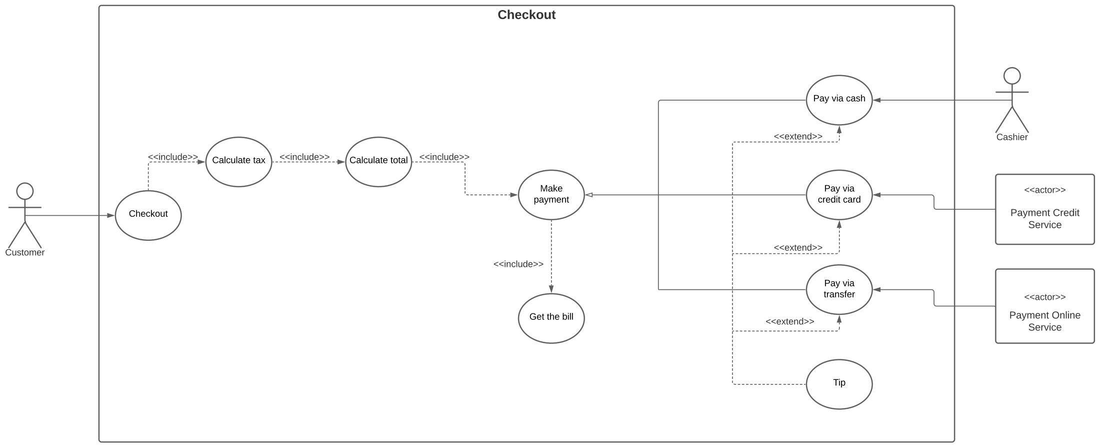


Figure 3: Use-case diagram for Checkout process

Use case name:	Checkout
Actors:	Customers
Description:	Customers can checkout by using one of following methods: - Scan QR code to transfer money. - Give cash to the cashier. - Use credit card.
Trigger:	Customers press "Checkout" button to do payment.
Preconditions:	- Customers' phone can connect to the Internet and access to the payment system. - Customers bring along cash to checkout. - Customers have credit card.
Post-conditions:	- The system verify that the payment finish successfully.
Normal flows:	1. Customers press "Checkout" button on the browser. 2. The system should calculate the tax and the total price. 3. The customers should receive the bill from the clerk. 4. The system displays all available payment methods. 5. The customers will choose the payment method. 6. The system displays the fill-in form. 7. The customers fill in the form and complete the payment.
Alternative flows:	- At step 4, customers choose appropriate method to checkout.

Table 1: Use case using table format

1.3.2 Feature: View menu

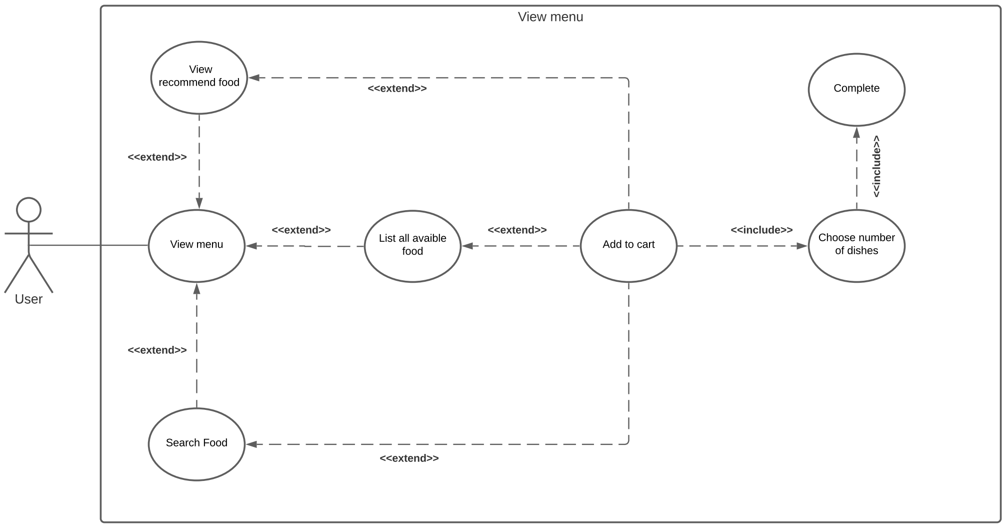


Figure 4: Use-case diagram for View-menu process

Use case name:	View Menu
Actor	Customer
Description	With view menu, customer can see restaurant's menu with recommendation, use search engine and add to cart.
Trigger:	Customer press "View Menu" button
Preconditions:	Customer device can connect to Internet and connect to the system
Post-conditions:	Customer receive order information
Normal flows:	1. Customer press "View Menu" button on the website. 2. System display options (List all available food, Search food, List recommend food). 3. Customer choose one option. 4. System display list of dishes according to customer's choice. 5. Customer choose dishes and input amount of each type. 6. System receive customers order and announce order successful then return to the main menu.
Alternative flows:	Step 3b: Customer choose "search food", then customer enters the key word Step 4b: System display list of dishes according to customer's keyword.

Table 2: Use case using table format

2 System modelling

2.1 Activity diagram

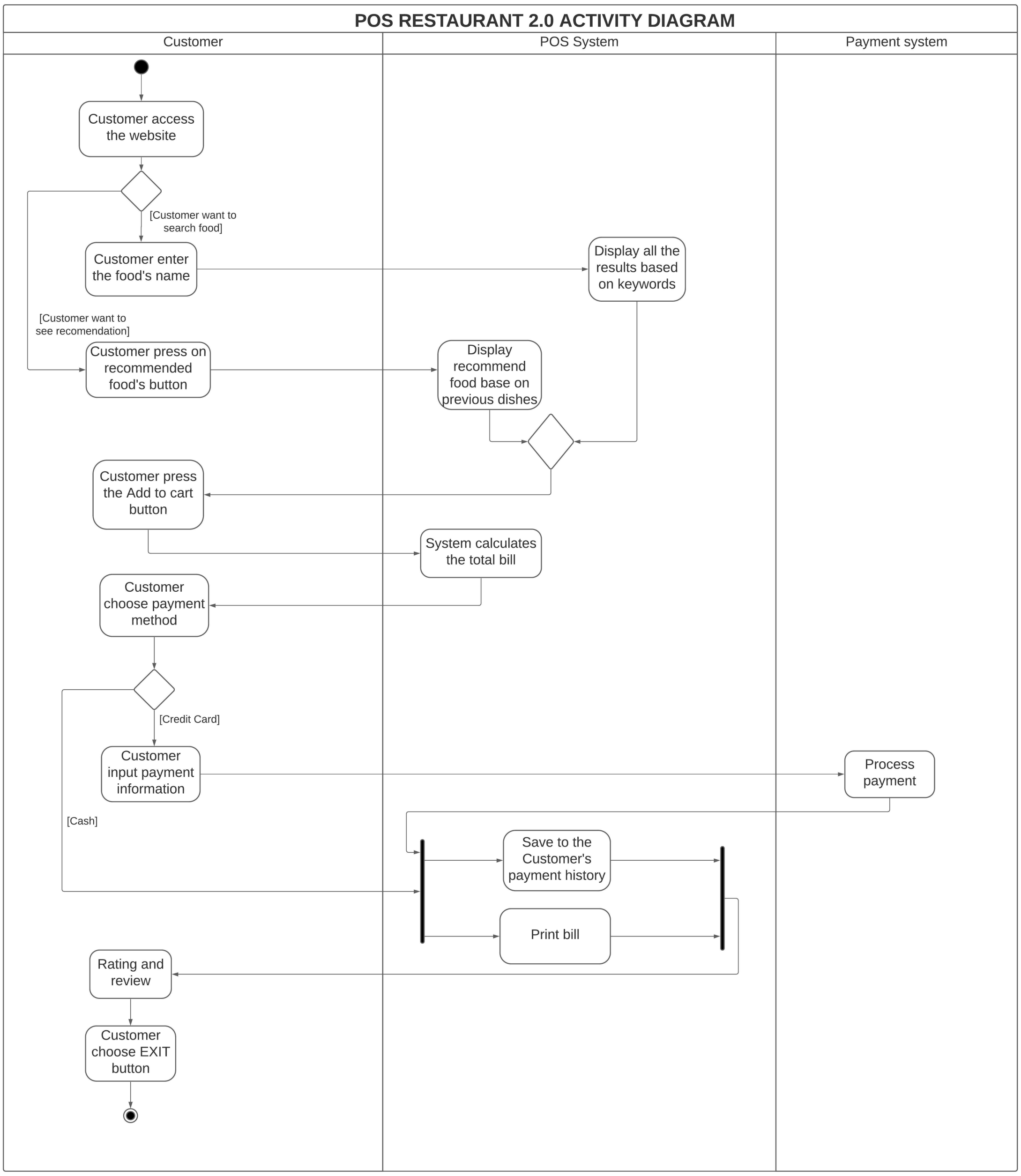

POS RESTAURANT 2.0 ACTIVITY DIAGRAM
Customer
POS System
Payment system
Customer access the website
[Customer want to search food]
Customer enter the food's name
Display all the results based on keywords
[Customer want to see recomendation]
Customer press on recommended food's button
Display recommend food base on previous dishes
Customer press the Add to cart button
System calculates the total bill
Customer choose payment method
[Credit Card]
Customer input payment information
Process payment
[Cash]
Save to the Customer's payment history
Print bill
Rating and review
Customer choose EXIT button

Figure 5: Activity Diagram

2.2 Sequence diagram

2.2.1 Sequence Diagram of View-Menu process

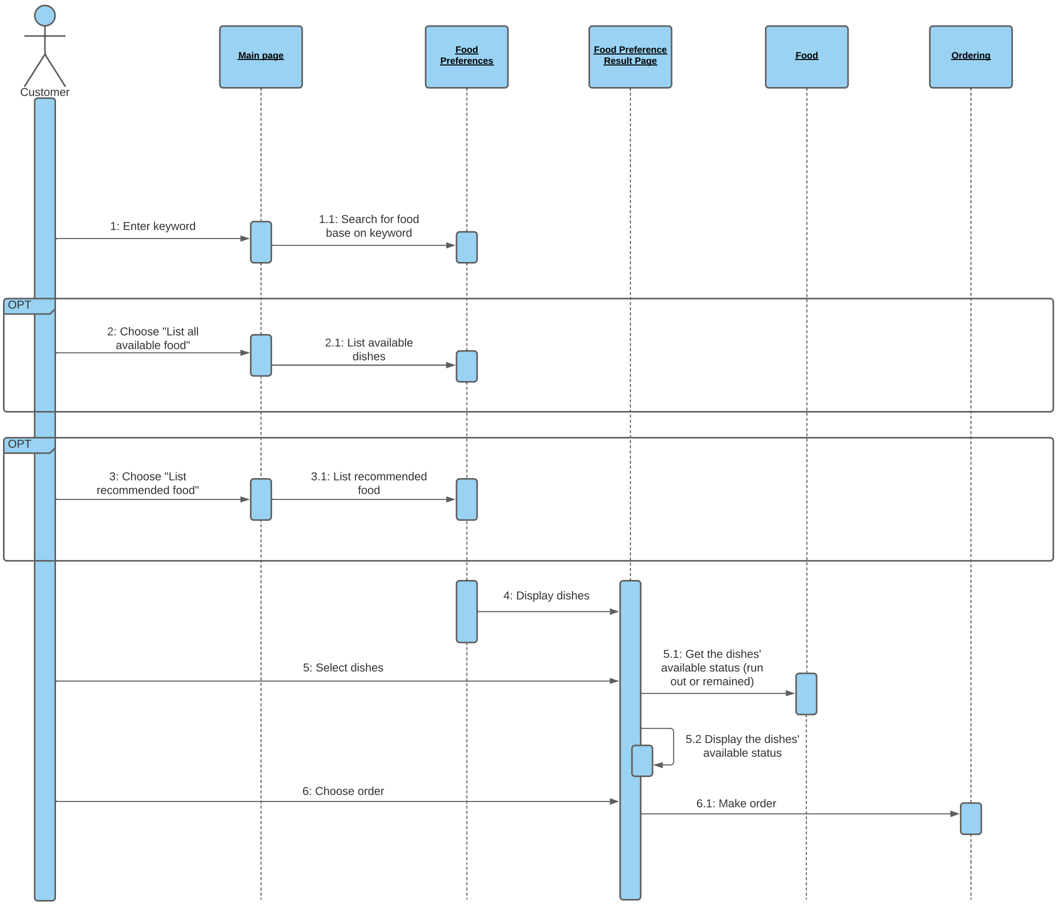

Customer
Main page
Food Preferences
Food Preference Result Page
Food
Ordering
1: Enter keyword
1.1: Search for food base on keyword
OPT
2: Choose "List all available food"
2.1: List available dishes
OPT
3: Choose "List recommended food"
3.1: List recommended food
4: Display dishes
5: Select dishes
5.1: Get the dishes' available status (run out or remained)
5.2 Display the dishes' available status
6: Choose order
6.1: Make order

Figure 6: Sequence Diagram of ViewMenu process

2.2.2 Sequence Diagram of Payment process

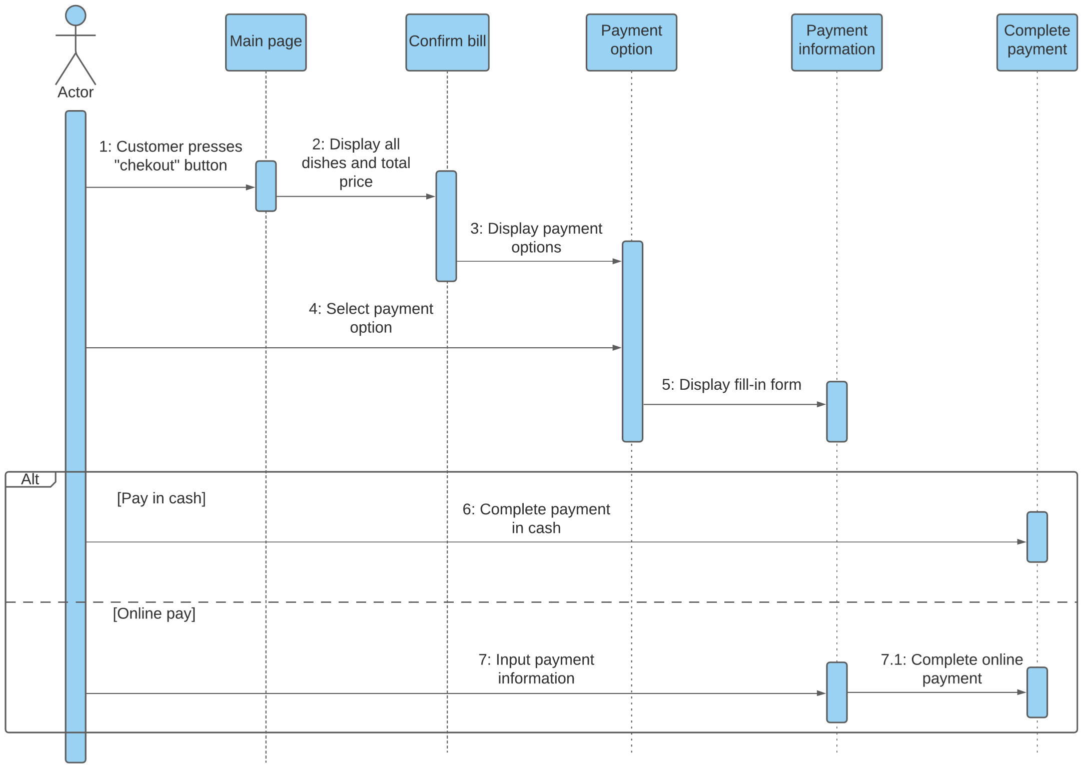

Actor
Main page
Confirm bill
Payment option
Payment information
Complete payment
1: Customer presses "chekout" button
2: Display all dishes and total price
3: Display payment options
4: Select payment option
5: Display fill-in form
Alt
[Pay in cash]
6: Complete payment in cash
[Online pay]
7: Input payment information
7.1: Complete online payment

Figure 7: Sequence Diagram of Payment process

2.3 Class diagram

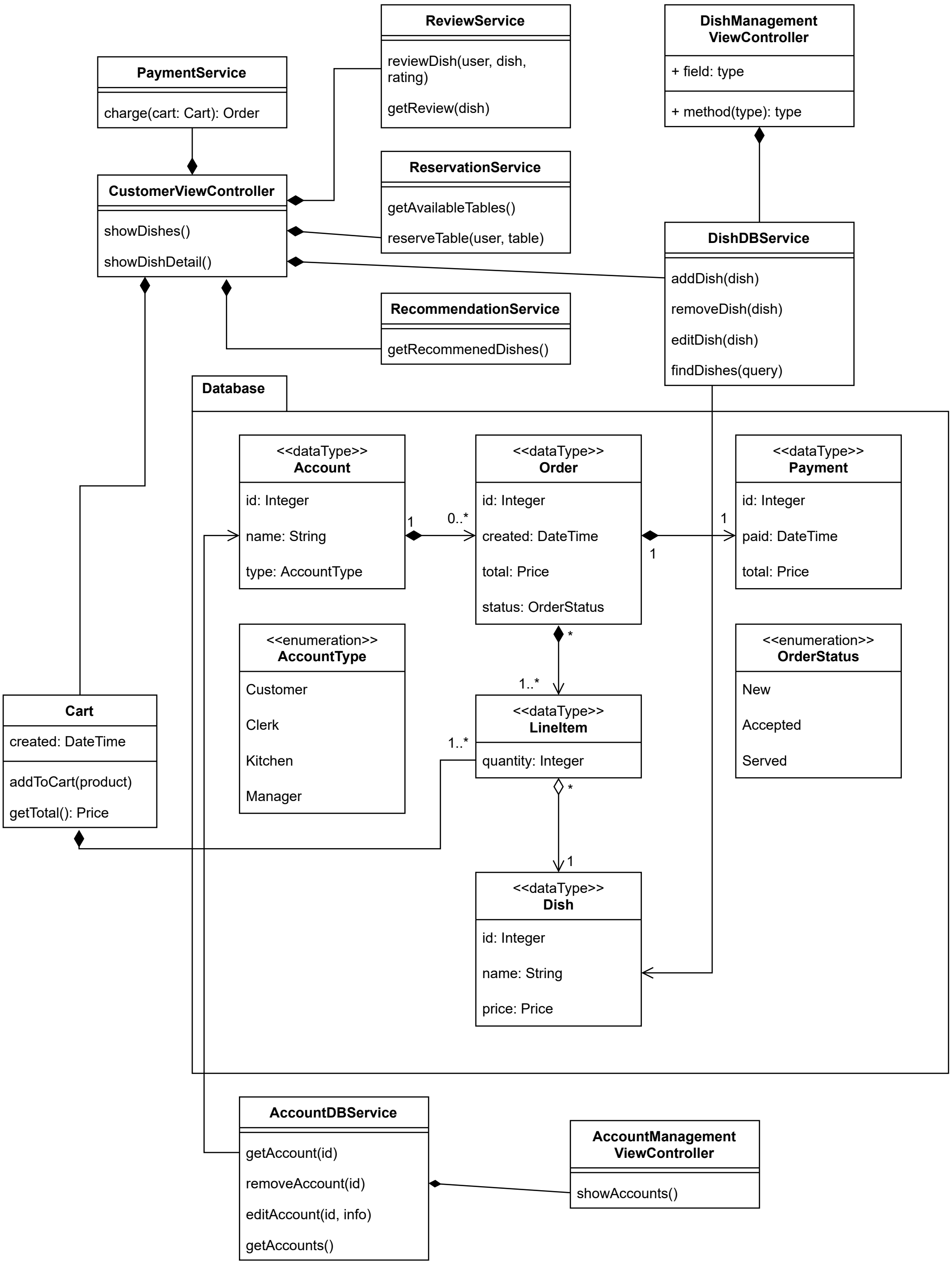


Figure 8: Class Diagram

3 Architectural design

3.1 Architectural approach

POS problems: The POS application can separate into three main components: **Database (Model), Web interface (View), User's interactive management (Controller**). Moreover, the display presented to the user frequently changes over time in response to user's action. The system needs to reflect data changes to all users in the way that they want to view them, while making it easy to make changes to the user interface.
From above reasons, we decide to choose the MVC design pattern approach, which also has many advantages corresponding to our demand.

Advantages of MVC in POS system

- **Faster Development Process:** MVC supports **rapid** and **parallel** development. If an MVC model is used to develop POS system then it is possible that one programmer can work on the view while the other can work on the controller to create the business logic of the system.
- **Ability To Provide Multiple Views:** In the MVC Model, we can create **multiple views** for a model. In POS system, each customers, database administrator and managers need to view the food in different ways.
- **The Modification Does Not Affect The Entire Model**: In POS system, the user interface tends to change frequently. It is obvious that you make frequent changes in your web application like changing type of food, amount, screen layouts. Moreover, adding a new type of view are very easy in the MVC pattern because the Model part does not depend on the views part. Therefore, any changes in the Model will not affect the entire POS system.

3.2 MVC Pattern for POS system

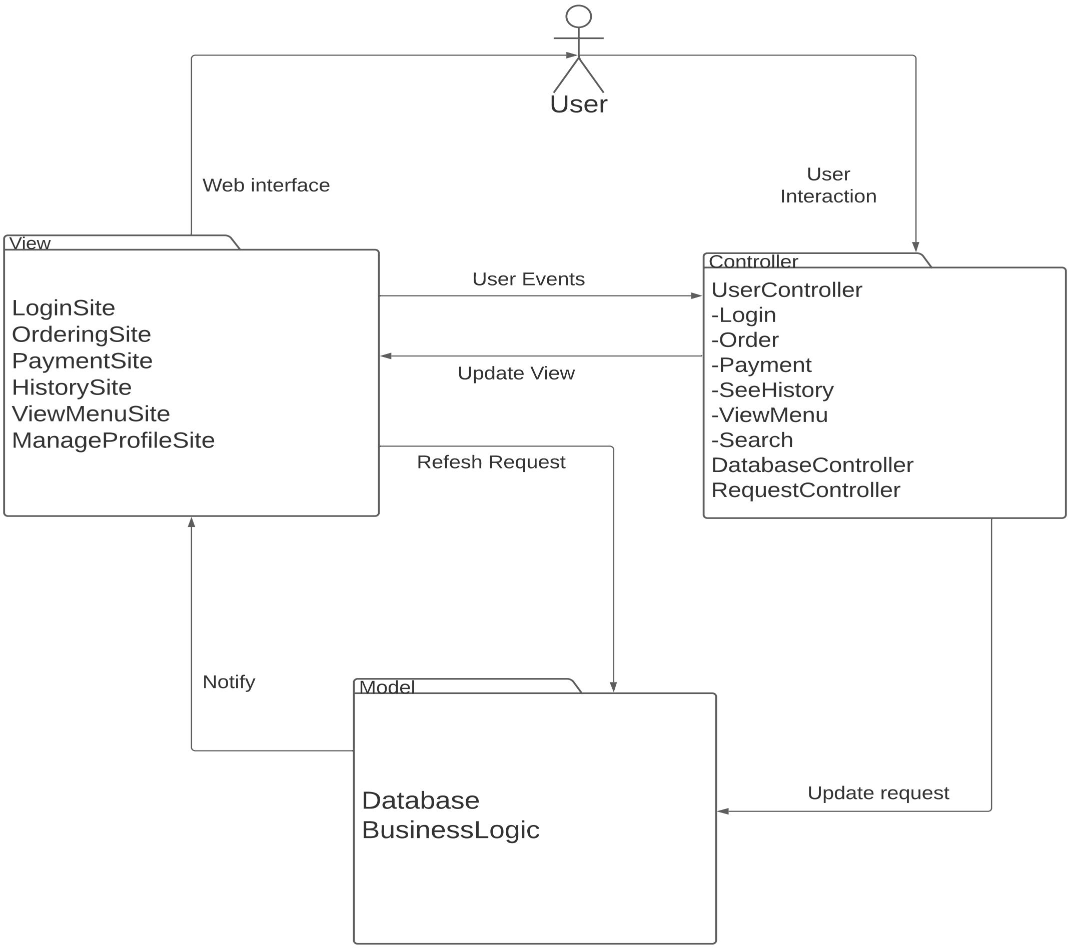


Figure 9: MVC Pattern for POS system

3.3 Deployment Diagram

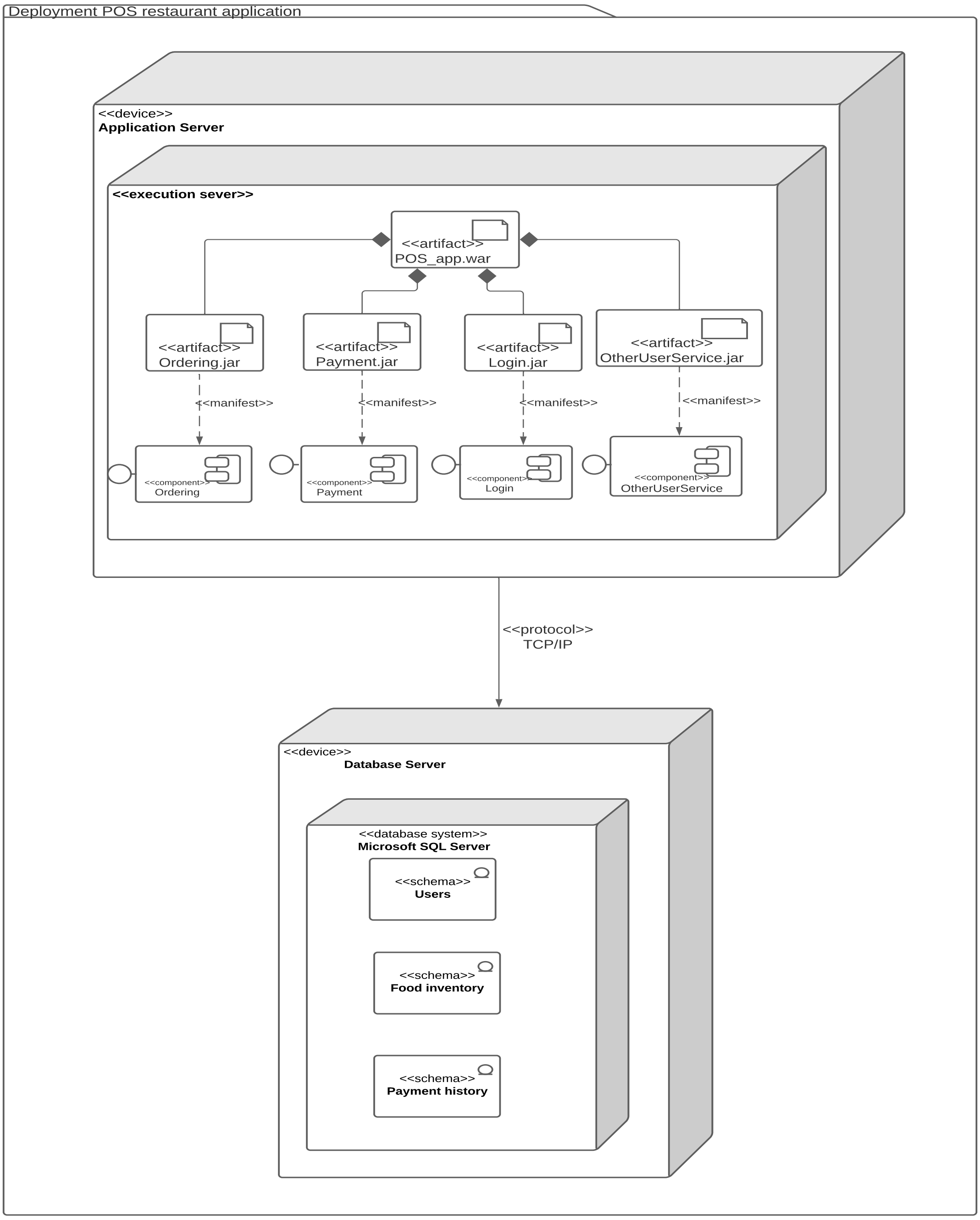


Figure 10: Deployment Diagram

3.4 Component Diagram

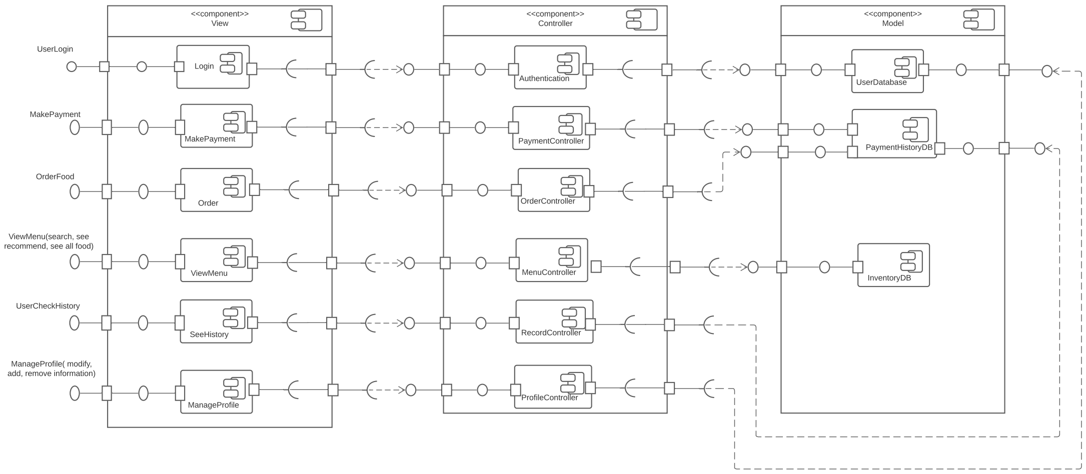
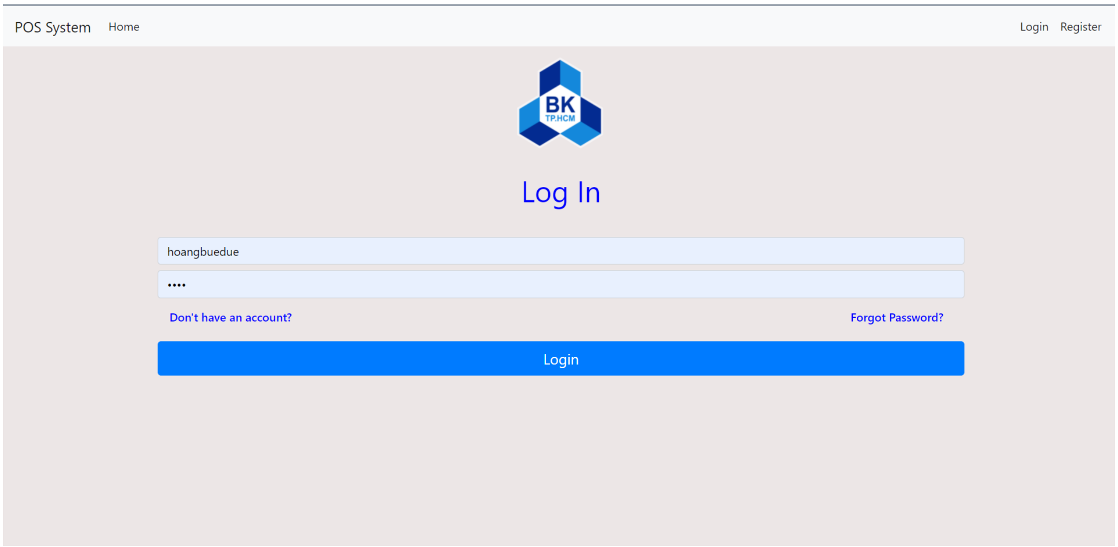


Figure 11: Component Diagram

4 UI Preview

4.1 Login Form

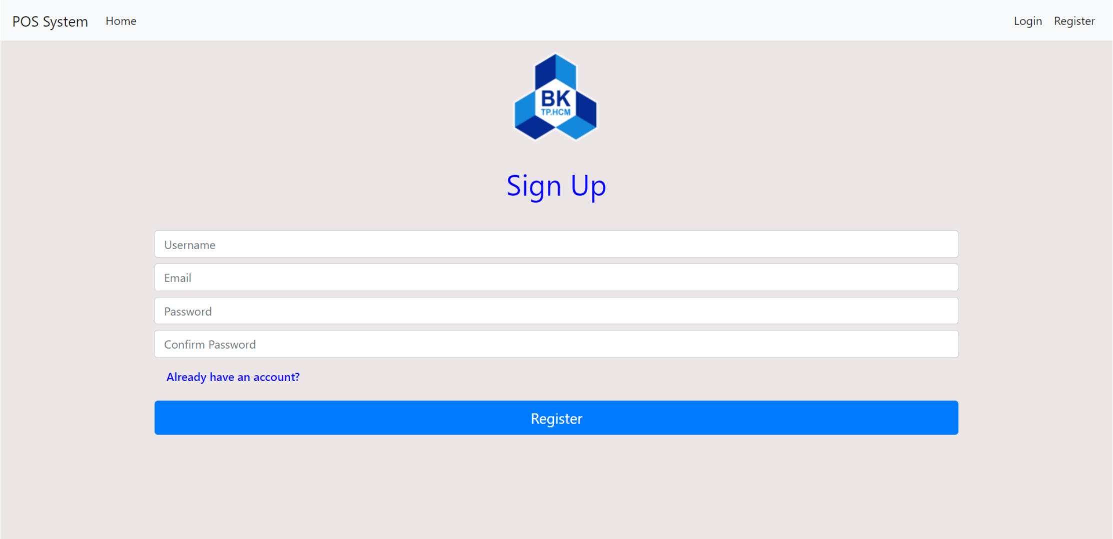


Log In

hoangbuedue

••••

4.2 Register Form



POS System Home Login Register

Sign Up

Username

Email

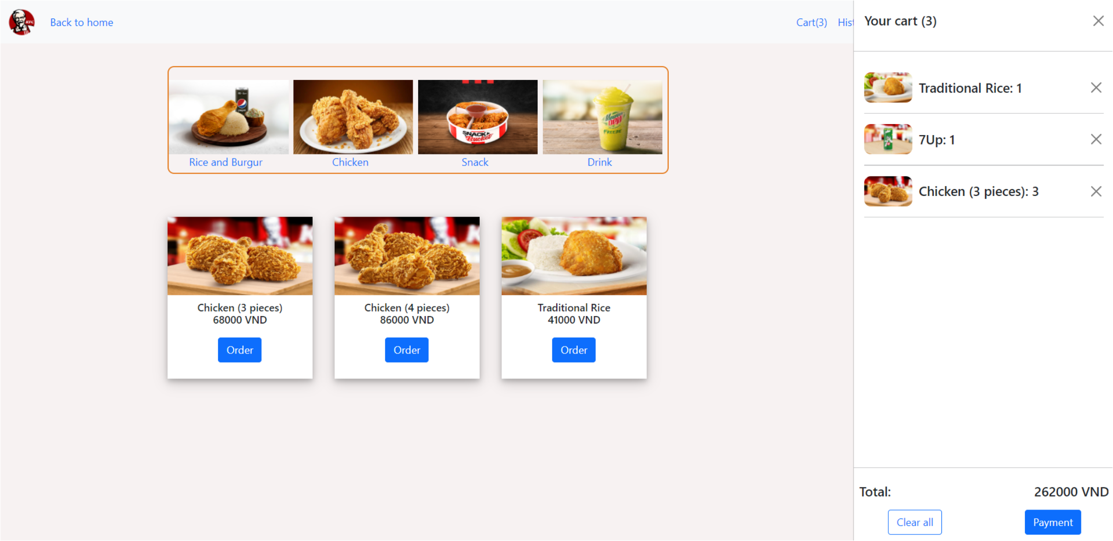
Password

Confirm Password

Already have an account?

Register

4.3 Main page



Back to home Cart(3) Hist

Rice and Burgur Chicken Snack Drink

Chicken (3 pieces)
68000 VND
Order

Chicken (4 pieces)
86000 VND
Order

Traditional Rice
41000 VND
Order

Your cart (3)

Traditional Rice: 1

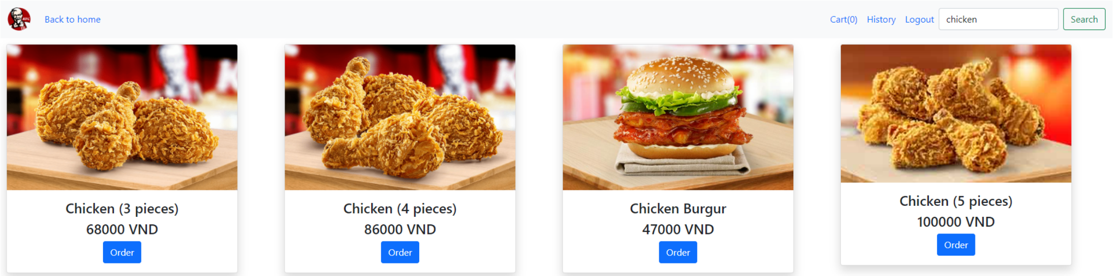
7Up: 1

Chicken (3 pieces): 3

Total: 262000 VND

Clear all Payment

4.4 Search Engine



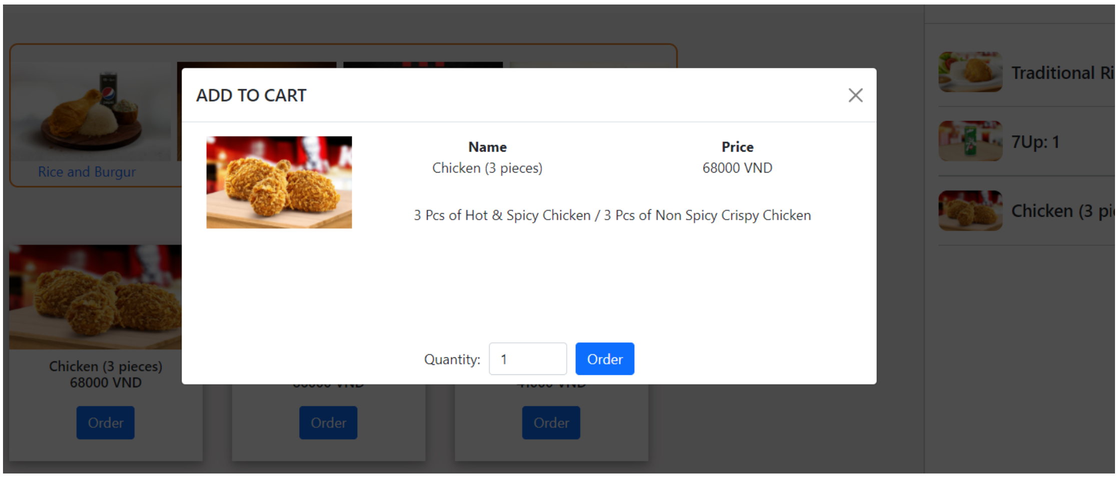
Chicken (3 pieces)
68000 VND

Chicken (4 pieces)
86000 VND

Chicken Burgur
47000 VND

Chicken (5 pieces)
100000 VND

4.5 Add to Cart



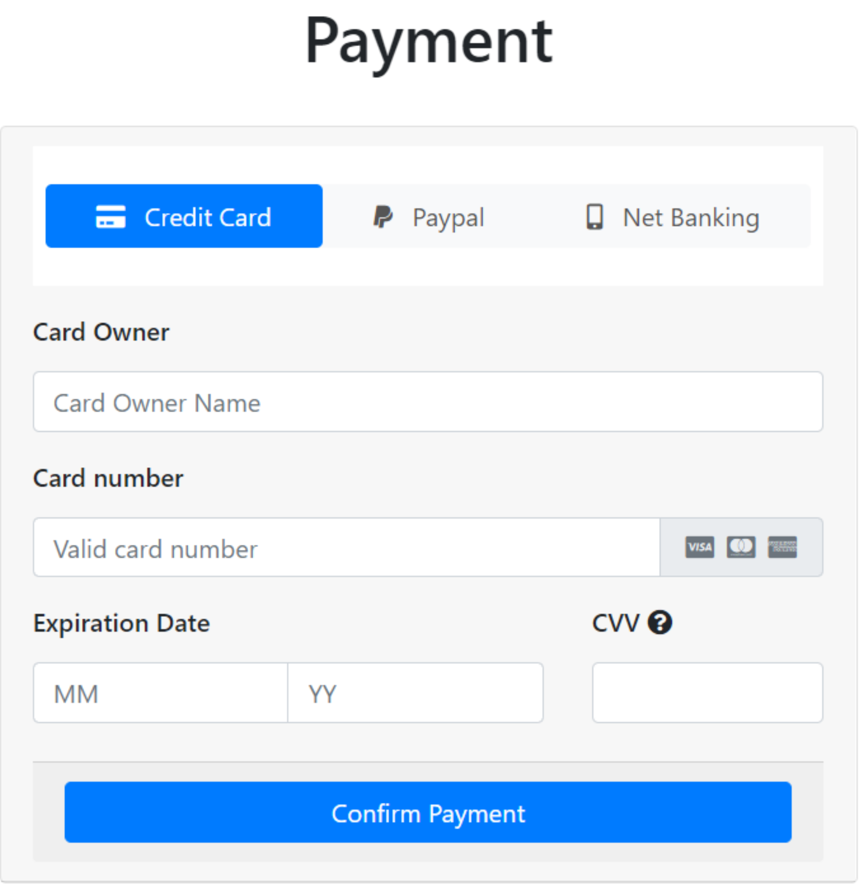
ADD TO CART

Name	Price
Chicken (3 pieces)	68000 VND

3 Pcs of Hot & Spicy Chicken / 3 Pcs of Non Spicy Crispy Chicken

Quantity: 1 Order

4.6 Payment



Payment

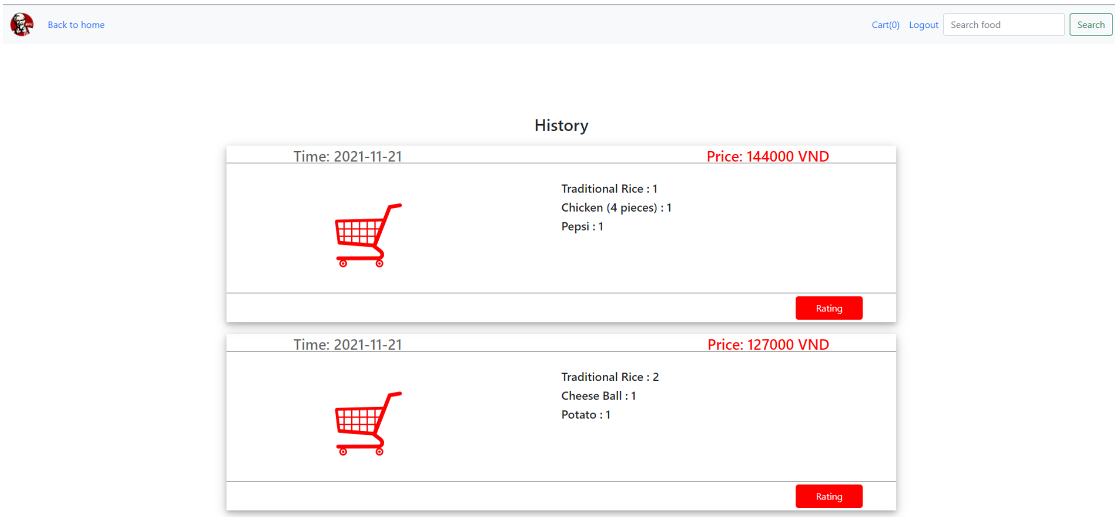
Card Owner: Card Owner Name

Card number: Valid card number

Expiration Date: MM YY

CVV

4.7 History



History

Time: 2021-11-21 Price: 144000 VND

Traditional Rice : 1
Chicken (4 pieces) : 1
Pepsi : 1

Time: 2021-11-21 Price: 127000 VND

Traditional Rice : 2
Cheese Ball : 1
Potato : 1

5 Source code and reference

Our **Github** repository link:
https://github.com/dinhhoanganh2001/SoftwareEngineering
Website link:
https://dinhhoanganh2001.herokuapp.com/
Reference

1. Functional and non-functional requirements https://qracorp.com
2. Deployment diagrams https://www.uml-diagrams.org
3. HTML, CSS https://www.w3schools.com
4. Flask library https://flask.palletsprojects.com/en/2.0.x
5. Search engine https://medium.com
6. Login https://www.digitalocean.com