Ho Chi Minh University of Technology
Faculty of Computer Science & Engineering

SOFTWARE ENGINEERING

ASSIGNMENT

RESTAURANT POS 2.0

Lecturer
Dr. Quan Thanh Tho, Associate Professor

Students
Nguyen Bao Phuc - 1712674
Pham Ngoc Manh - 1813045
Lam Minh Vinh - 2120082
Pham Ngoc Hau - 1911136
Nguyen Huu Thuan - 1920060

Contents

1 Report 1

1.1 Introduction

Project context

During the COVID-19 crisis, people are paying more attention to their health and convenience, meanwhile, software engineering shows its myriad advantages by assisting and promising people unique ways of experiencing the outside environments. Concerning the restaurant business, an application can automate tedious and insecure tasks such as giving customer the menu, offline payment, reserving a table by phone,... can make both the restaurant staffs and customers easier and safer. This project aims to deliver to the restaurant's owner a useful application, which has the ability of automating the ordering and processing meals procedure without directly communicating between customers and restaurant staffs to reduce wasted effort and protect people from the coronavirus pandemic.

Project stakeholders

- **Customer:** Can be considered as the major user class of this system. In Restaurant POS 2.0, customers can select their desired meals via a friendly menu, order foods, pay for their invoice, and give back some feedback or surveys.
- **Clerk:** Provide their services to customers by confirming orders, processing customers' feedback, and serving meals to customers after being notified by the kitchen staff via the application.
- **Point-of-sale terminals:** Including third-party payment service, staffs that have a responsibility of processing customers' order before sending a request to the kitchen staffs.
- **Chef:** Receiving requests from others staff, making food, and calling clerk to serve the ordered meal to the customer.
- **Manager**: Manage system's operation as well as customer's feedbacks.

Project objectives

In this project, we try to obtain results corresponding to the objectives are as follows:

- Customer can check the menu and request for the item that they want.
- Customer can access the application via a QR code.
- Clerk can receive customers' request without meeting and asking customer directly.
- Restaurant staff can check and confirm customer orders.
- Kitchen staff can receive the customer's order after restaurant staff confirms it.
- Recommendation system suggests user to choose some trending or discounted meals.

Project scope

In terms of project scope, there are three main points:

- Implement a responsive web application for both computer and mobile phone.
- Design a database to persist user or food information.
- Implement a recommendation system that can suggest user to choose meal from a menu, which can highly increase user experiences. Some basic machine learning techniques will be used.

Technologies

- Platform: Responsive web application.
- Back-end: NestJS framework.
- Front-end: ReactJS framework.
- Database: MongoDB.

1.2 Functional Requirement

Functional Requirement

- Payment: The customer can choose the payment method to pay for their order.
- Table reservation: Clerk can check the status of the reservation, customers can choose table.
- Food Ordering: Customers can view food and select food that they want, clerks check the status of the ordering, and the chef receives orders to prepare food.
- Billing: Make bill for customer's order.
- Checkout: Check customer's order and know your money will pay, include scan item, calculate total.
- Alerts: Alert warning if the customer choose a reserved table, a payment error occurs,...
- Scan QR Code: The system generates a QR code, Customer can scan that QR code to log-in to the system.
- Customer Management: Customers send feedbacks about restaurant's service. Manager can view feedbacks and sale report for each day, week, month.

1.3 Non-Functional Requirement

Security

- Users can access the application without creating a username and password. For takeaway orders, users have to provide at least their name, phone number, email,....
- Restaurant staffs must have their unique username/password pair. And only can access the application by the correct username and password.

Maintainability

- The system operates 24/7, maintenance is 1 time / month.
- Each maintenance time is not more than 4 hours.
- Maintenance preservation must be notified at least 3 days.

Reliability

- Correctly connect customers with their orders.
- Calculate the correct amount the customer needs to pay.

Concurrency and capacity

- The current transactions is about 300 orders per day.
- Page transition time under 3 seconds.

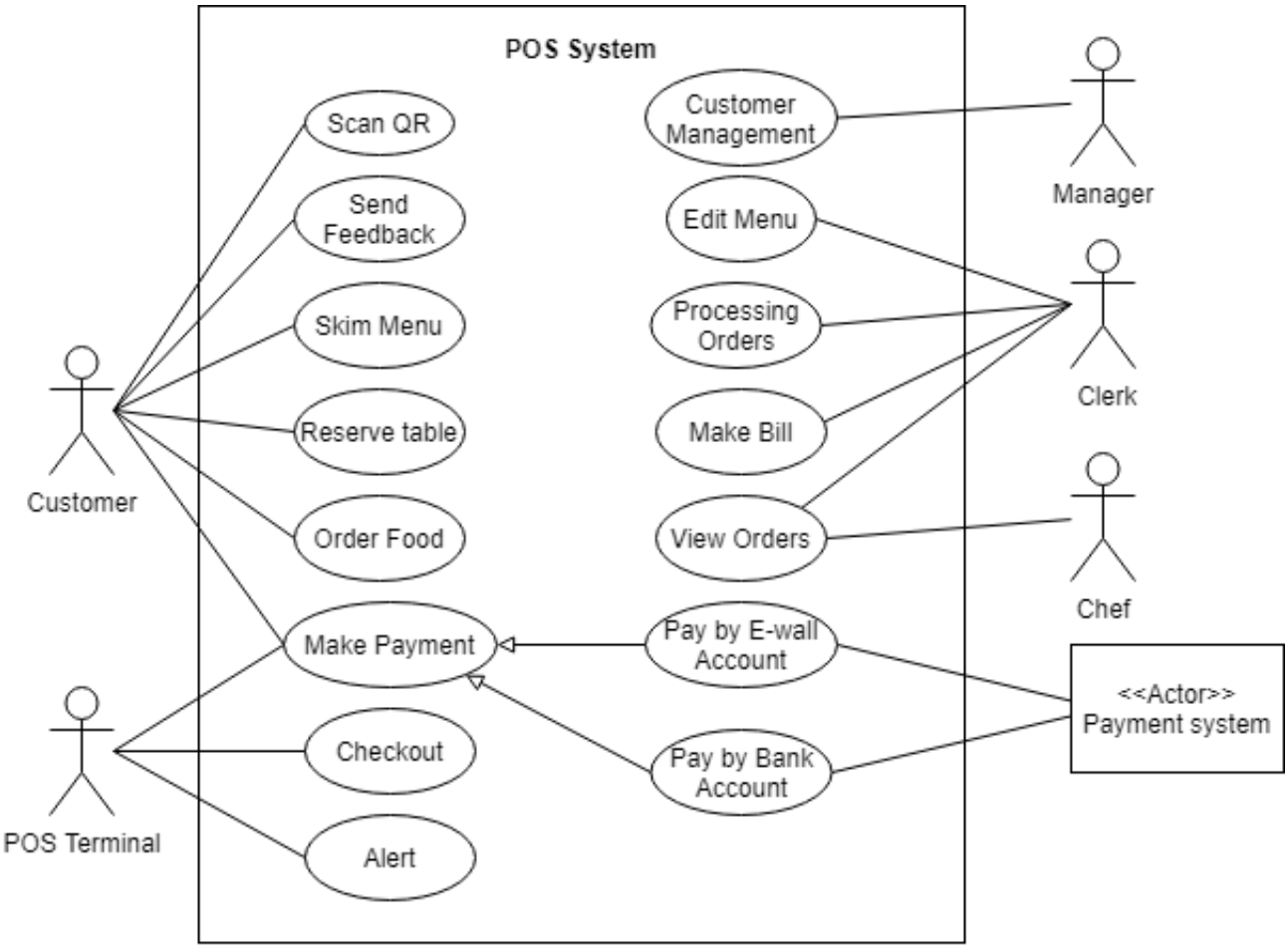
User interface

- The system should be usable from a mobile device, a tablet device or a normal computer/ laptop.
- Maximum 5 steps to complete a function.
- Up to 1 hour to use the system fluently.

Scalability

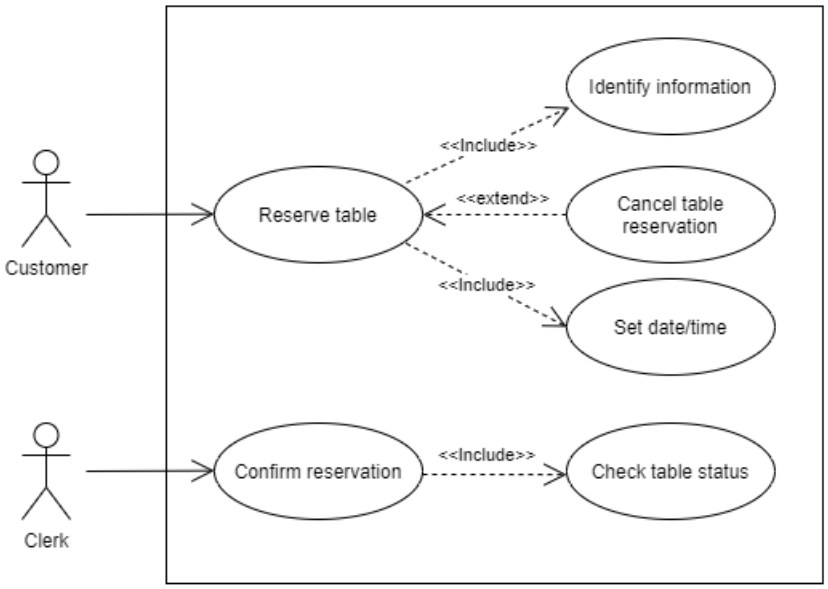
- The system should be extendable to use in multiple restaurants in the future.

Use-case diagram



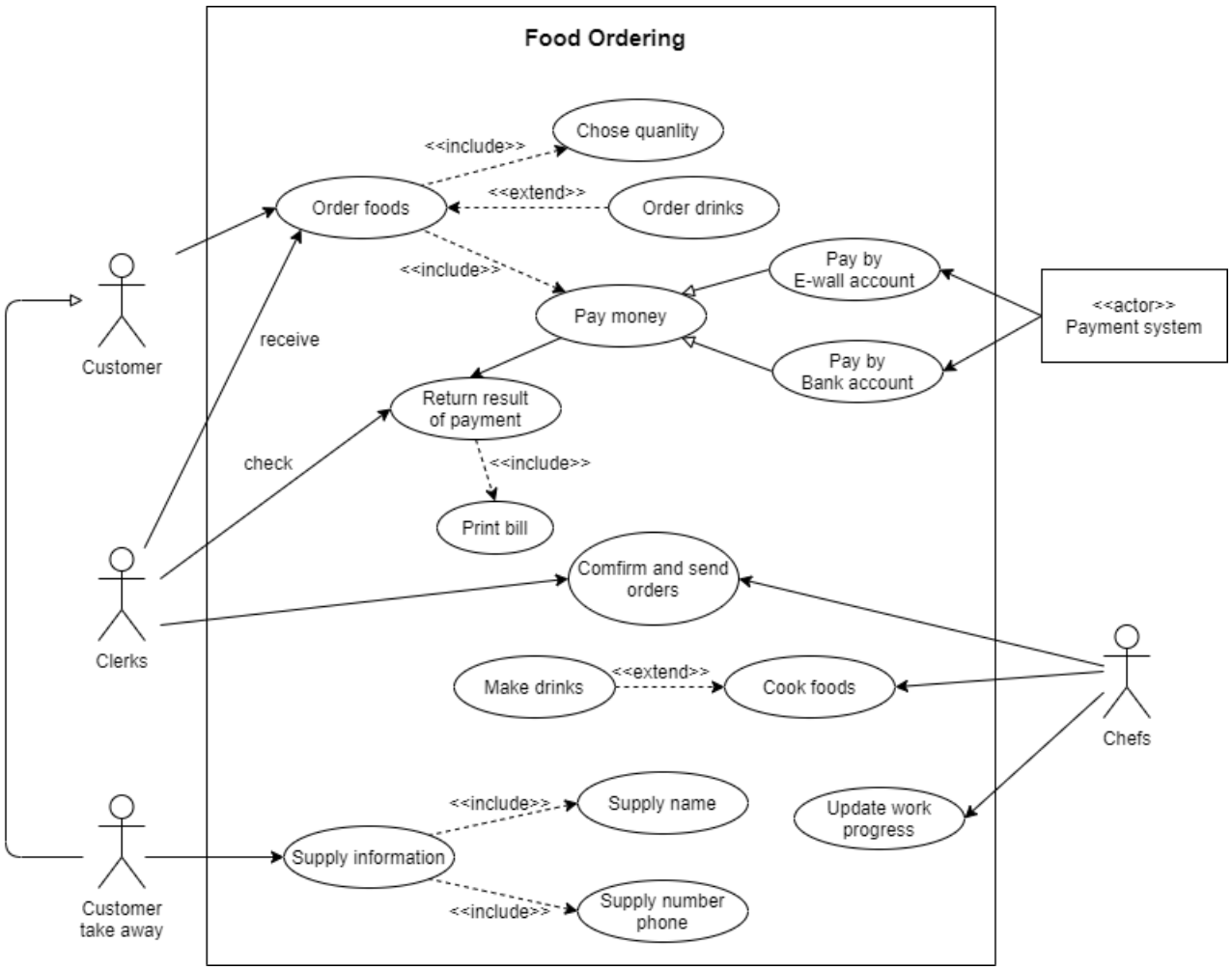
1.4 Use-case description

a. Table reservation



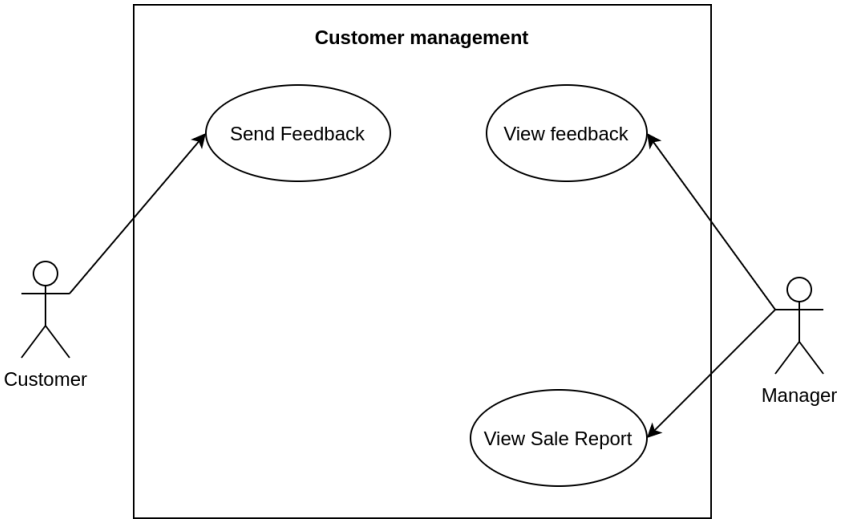
Name	Table reservation
Actor	Clerks and customers
Description	Clerk can check the status of the reserved table and confirm the identity of the customer. Customers can message, send notifications, and choose any seats via home page, they can also cancel the reservation.
Preconditions	Customers need to access to home page. Clerks need to have an account and log in first.
Normal flow	1. Customers go to home page of website by URL and clerks log in the website application. 2. In the section of table reservation, the website presents a page with many results in available seats. 3. Customers can choose the remaining seats they want 4. Customers have to identify their information (phones, names, ID, date-time). 5. Clerks can check the information of customers and the table they reserved, and confirm the result.
Exceptions	Exception 1: At step 5: Customer can cancel the table they reserved. Exception 2: At step 4: if customers have not identified all these information, the reservation cannot be executed. Exception 3: At step 3: If customers have chosen the reserved table, they cannot input the step 4. Exception 4 :At step 1, if Clerks have not logged into their account , they cannot operate the system.
Alternative flow	Alternative 1, Exception 2 : Website can notify the customers they have not identified their information yet. Alternative 2, Exception 3 : Website can notify the status of the table.

b. Food Ordering



Name	Food ordering
Actor	Clerks, customers, chefs and payment system
Description	Clerks receive orders, check payment, confirm orders and send orders to chefs. Customers chose food or drink, pay money, and get food. Chefs receive orders and cook food or make drinks. Payment system make transaction and return result to clerks.
Preconditions	Clerks have an account and login to home page. Customers access to home page. Chefs are ready. Payment system is available.
Normal flow	1. Customers click on Menu button to go to Menu page. 2. System gets and displays list of food. 3. Customers chose foods or drinks and quantity via Menu page. 4. Customers chose pay money via payment system by E-wall account or by Bank account. Payment system will make transaction and return result to clerks. 5. Clerks receive orders from customers, check result of payment system, confirm order. 6. System print bill and send orders to chefs. 7. Chefs receive orders, start cook foods or make drinks and update work progress. 8. Clerk mark the order as completed.
Exceptions	Exception 1: At step 3: If result of payment system is failed, customers will refuse orders.
Alternative flow	Alternative 1, Exception 1: Payment system notice failed transaction to customers and give reasons.

c. Customer management

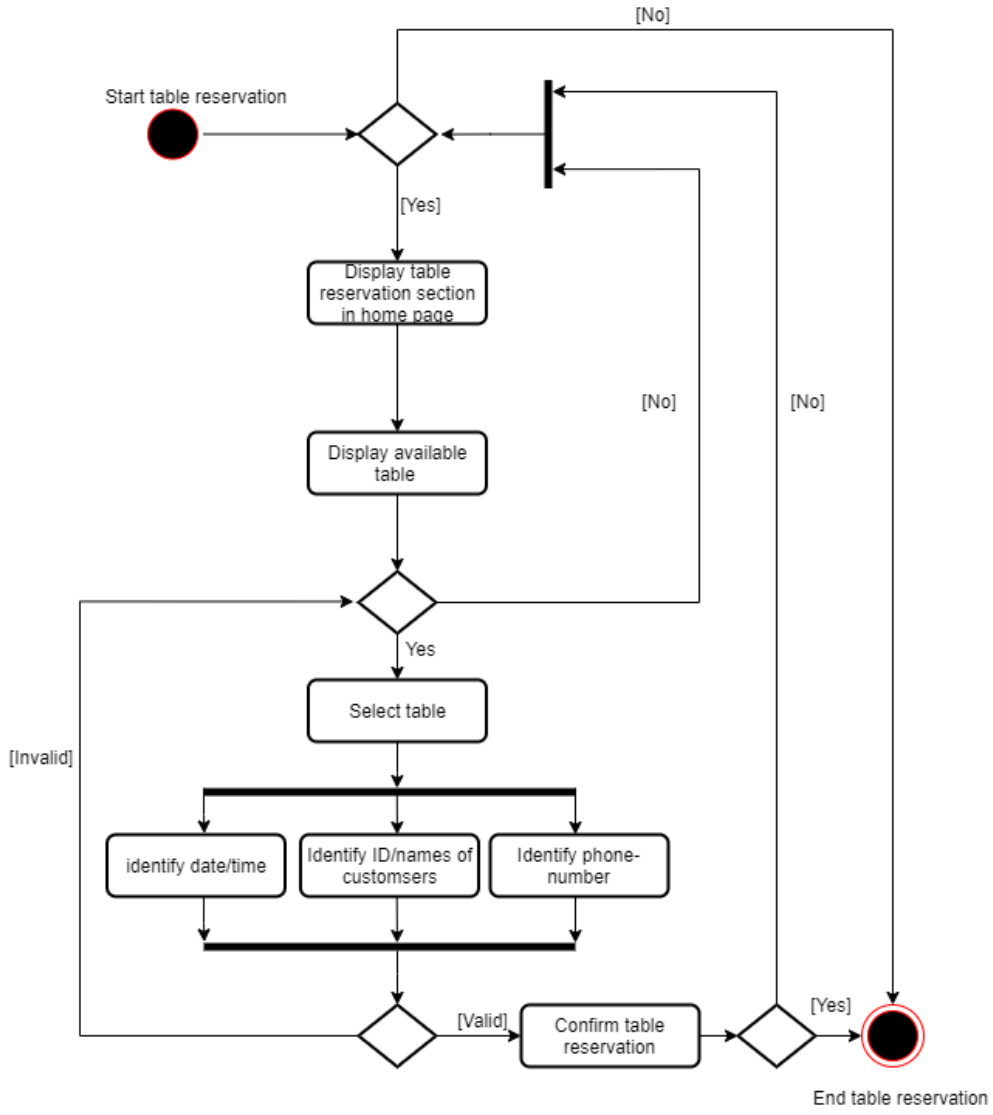


Name	**Food ordering**
Actor	Manager, customer
Description	Customers send feedbacks about restaurant's service. Manager can view feedbacks. Manage can view sale report for each day, week, month.
Preconditions	Managers have an admin account and login to home page. Customers access to the Feedback page.
Normal flow	1. Customers type their feedbacks in a feedback section. 2. System stores and sends feedback notification to manager. 3. Manager inspect customer's feedbacks in a customer feedback section. 4. Manager can view the sale report by clicking on the Sale report button. 5. System will calculate and send to report to manager as a pdf file.
Exceptions	Exception 1: At step 4: If the system does not have any sale record, manager can not view sale report.
Alternative flow	Alternative 1, Exception 1: The system can notify the sale status to the manager.

2 Report 2

2.1 Activity diagram

a. Table Reservation

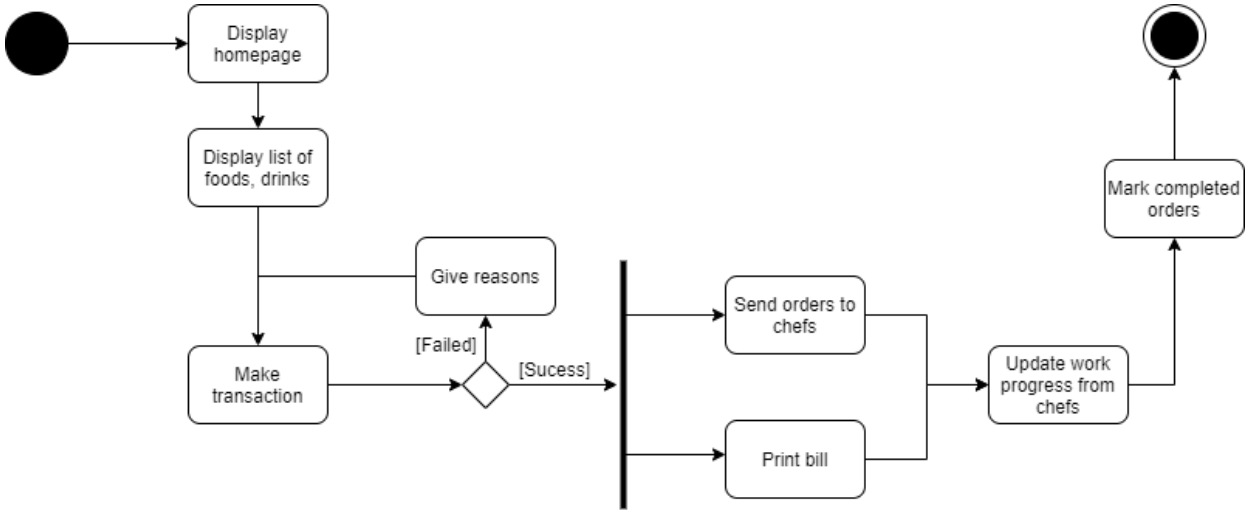


Description:

- In the first stage, there is a decision node that whether customers are able to access to table reservation or not, which means it can create the flow of output when they think reservation is not necessary.
- When customers access to table reservation section,the application show the available seats.
- Subsequently, a decision node 2 appears, if none of table is unavailable or customers unsatisfied with the remaining available seats, they can return to decision node 1.In contrast, they can select the table.
- After selecting the table, customers have to declare their information such as date/time, names, and phone-number.
- After that,decision note 3 will show that if one of these information is invalid or unidentified, the application will return the decision note 2 and system will show the notification about that.In contrast, they obviously confirm table reservation in the next step.

- After confirmation step, the existence of decision note 4 means that if customers accept the confirmation.If not,they will return decision note 1.

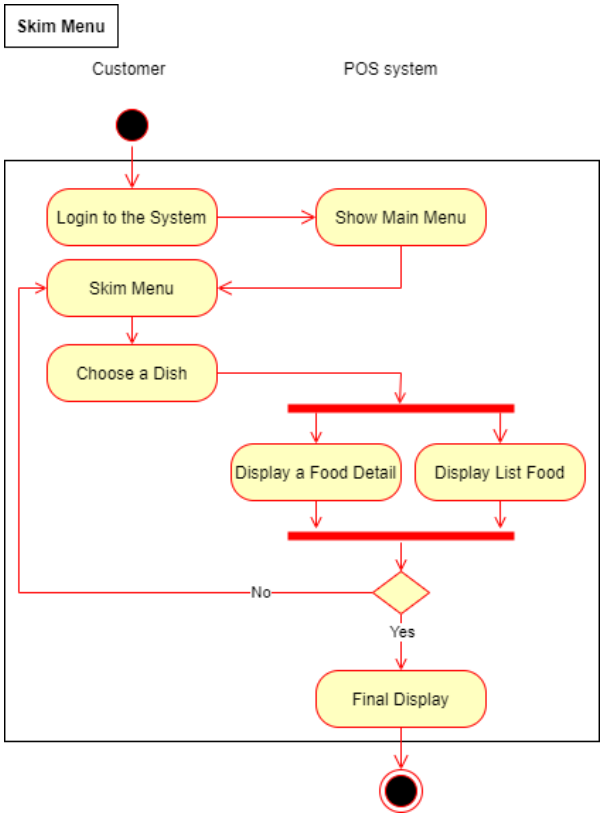
b. Food Ordering



Activity diagram about Food Ordering

Description: System will display homepage. After customer click on menu button, system will display list of foods and drinks to customer order. Then, customer choose pay money, system make a transaction. If transaction is fail, system will give reason for customer. Otherwise, system will print bill, send order to chefs. Update progress of order and mark completed when chefs done.

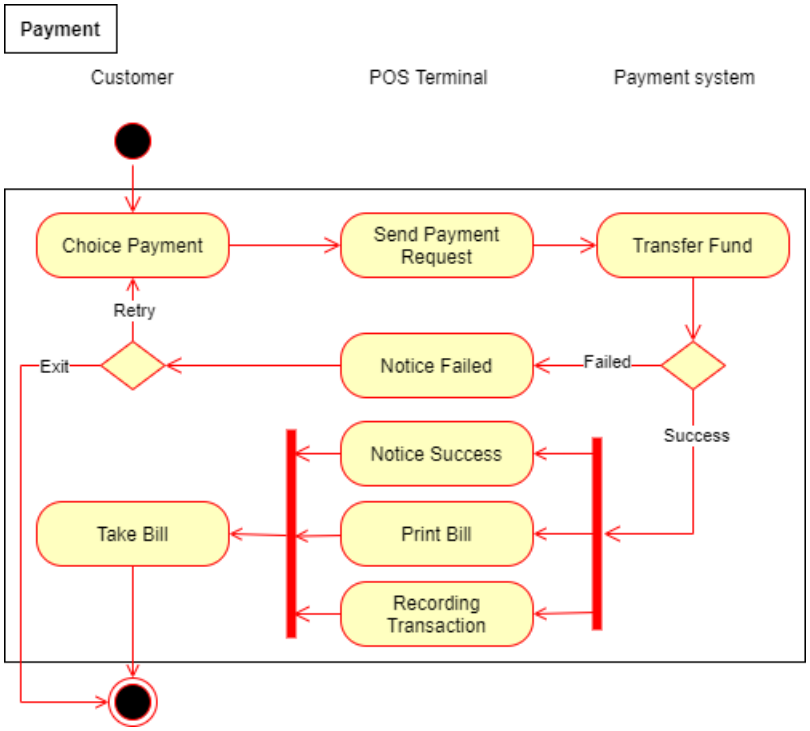
c. Skin Menu



Description:

- The customer logs into the system.
- The system displays the main menu.
- Customer can skim the menu and choose their favorite dishes.
- The system displays the customer's dishes according to the list of related dishes or the details of a single dish.
- If customer can't find the dish, they can skim the menu and choose another dish again.

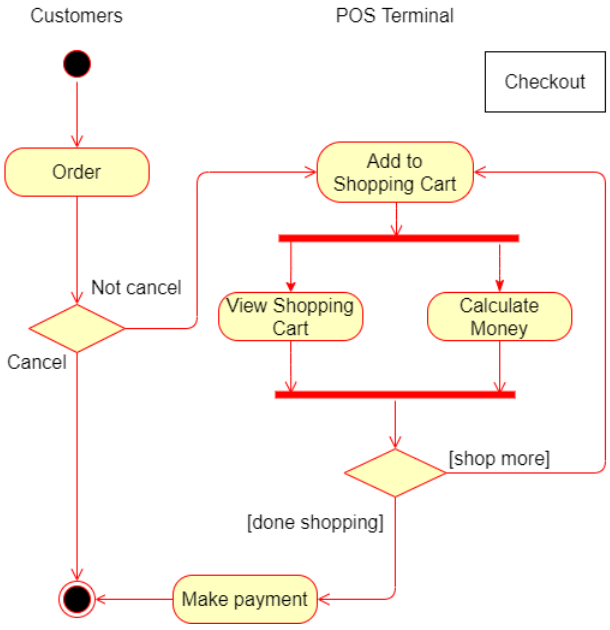
d. Payment



Description:

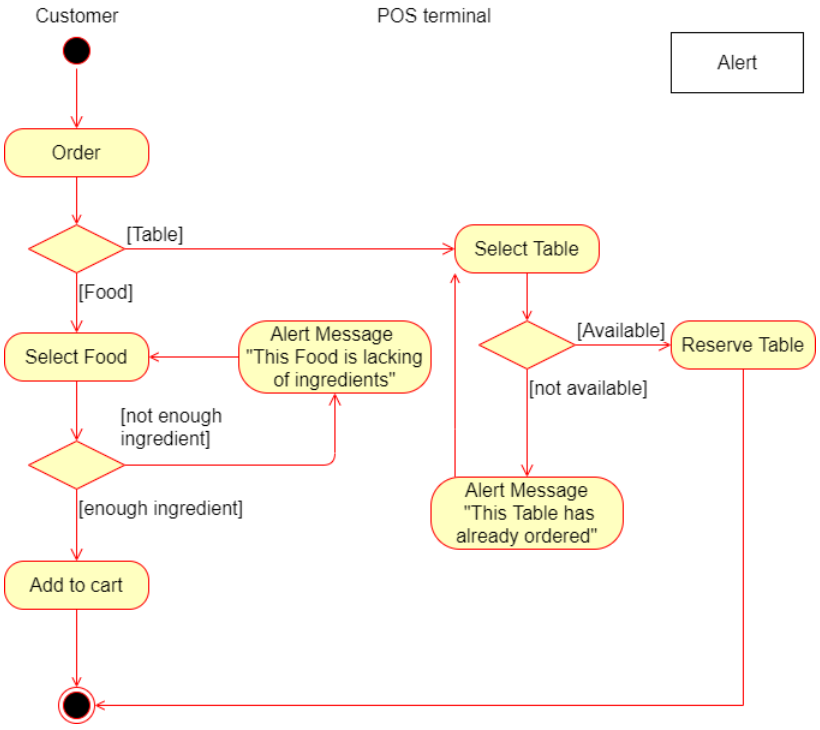
- Display the payment page.
- Customer choose payment gateway and transfer money.
- The system updates the payment process.
- Show paid if payment is successful, save the transaction and issue an electronic invoice to the customer.
- Show failed payment and ask customer to try again or exit.

e. Checkout



Description: When the customer orders something(food, table), the pos terminal will display the shopping cart and calculate the total amount. If the customer wants to choose more, the POS terminal will display the menu, and vice versa if the customer completes the purchase and presses pay, the pos terminal will display the payment system.

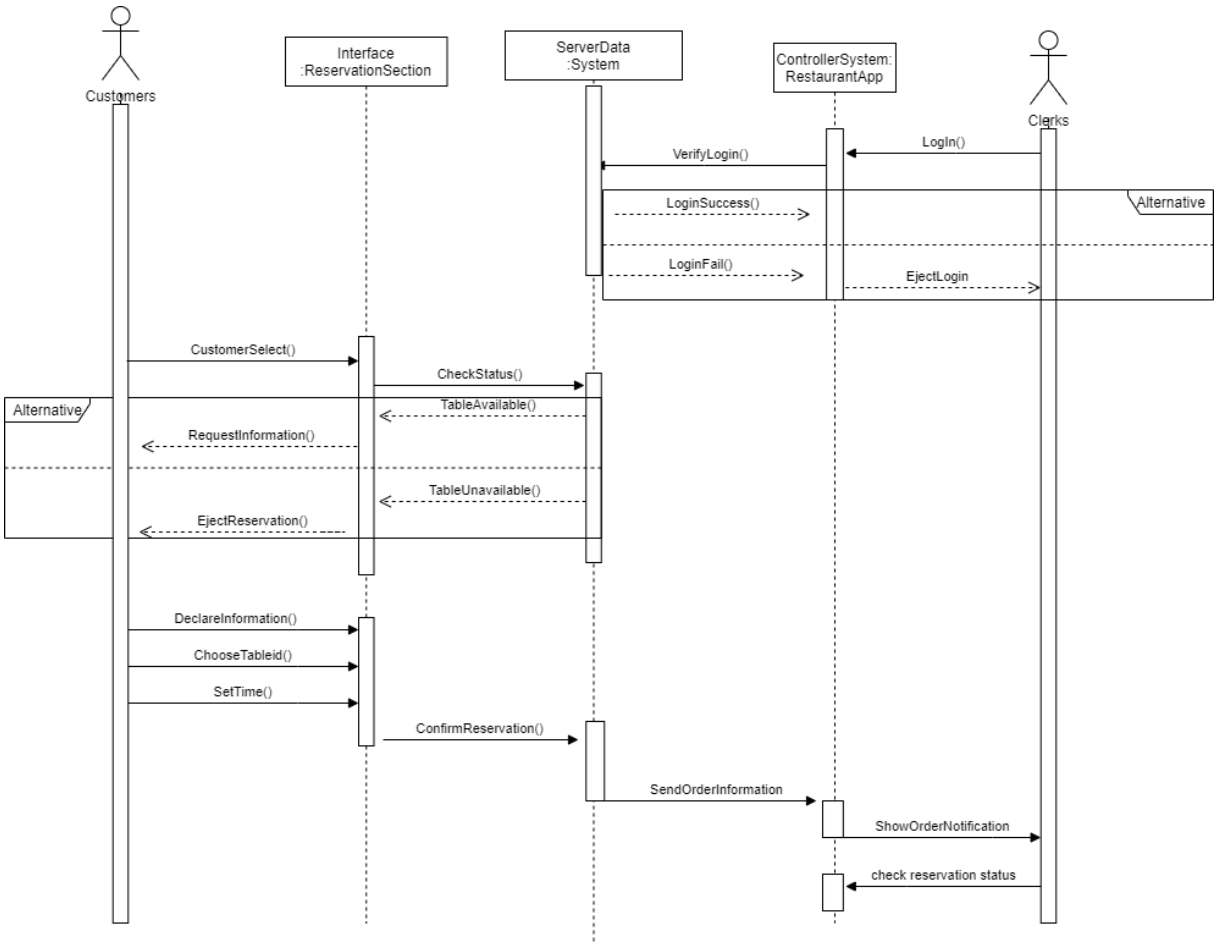
f. Alert



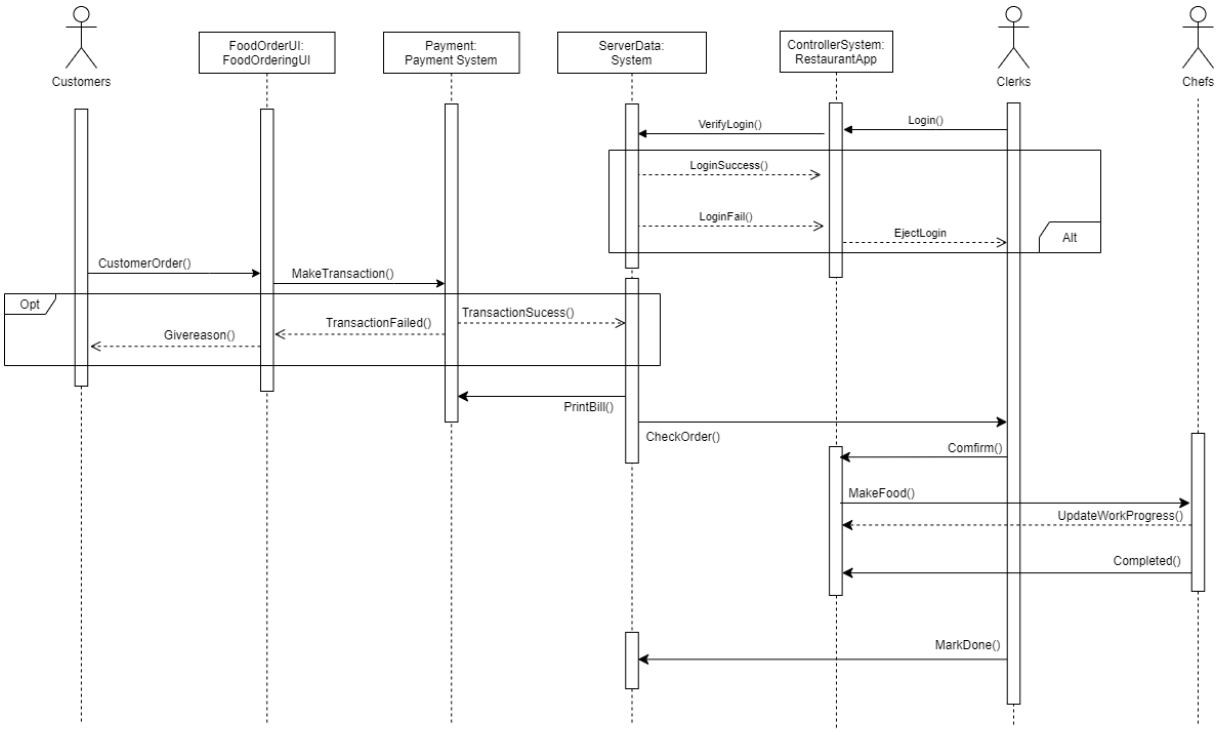
Description: When customer orders, if the ingredient of the food is not enough, or the table is not available. The POS system will alert about this error and require customer choose again.

2.2 Sequence diagram

a. Table reservation

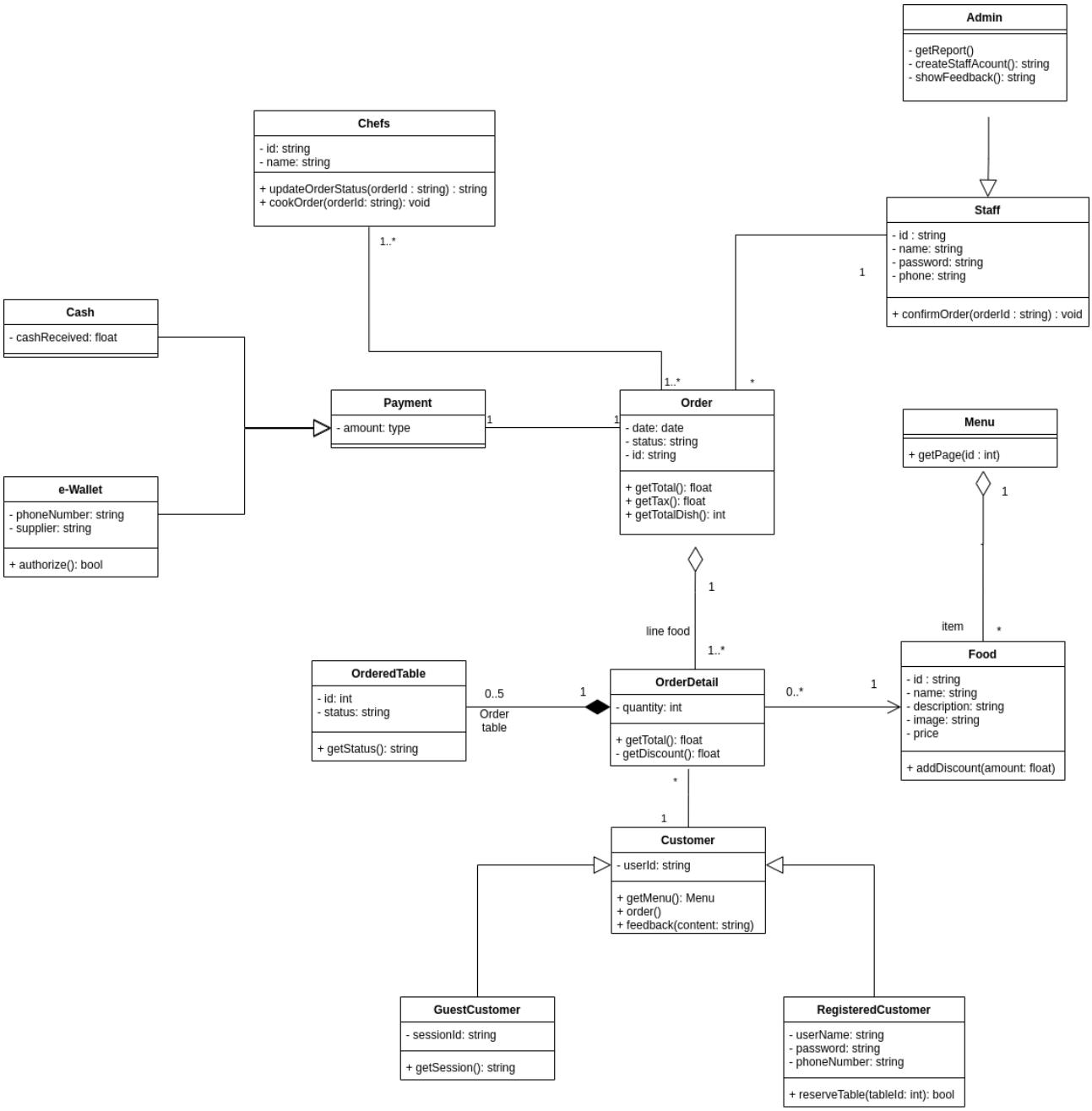


b. Food Ordering



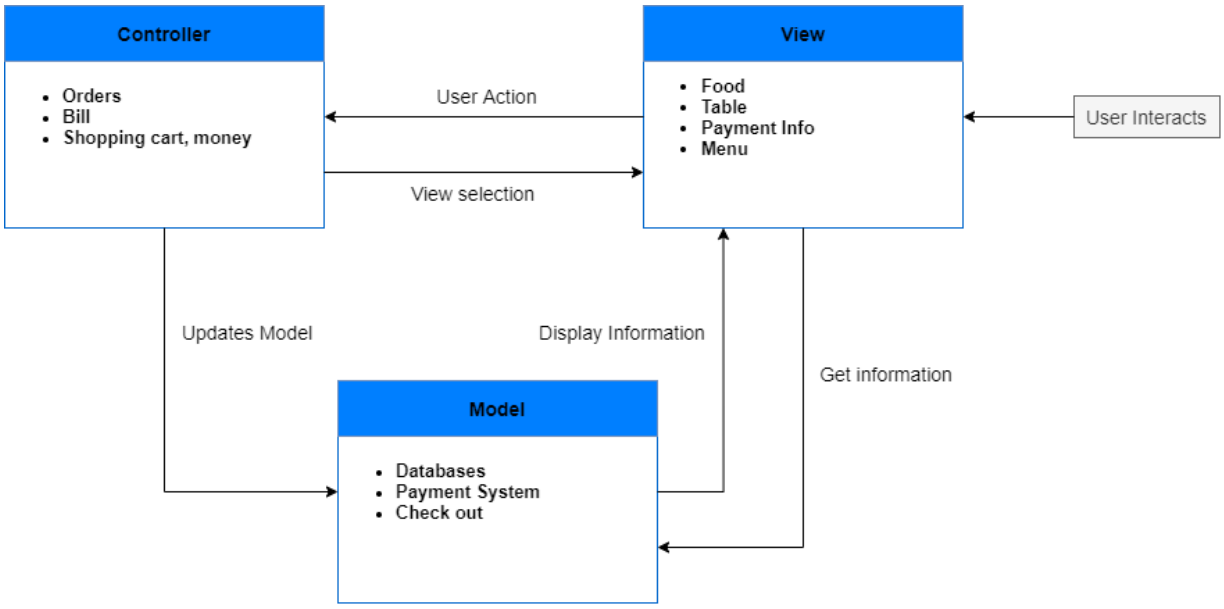
Sequence diagram about Food Ordering

2.3 Class diagram



3 Report 3

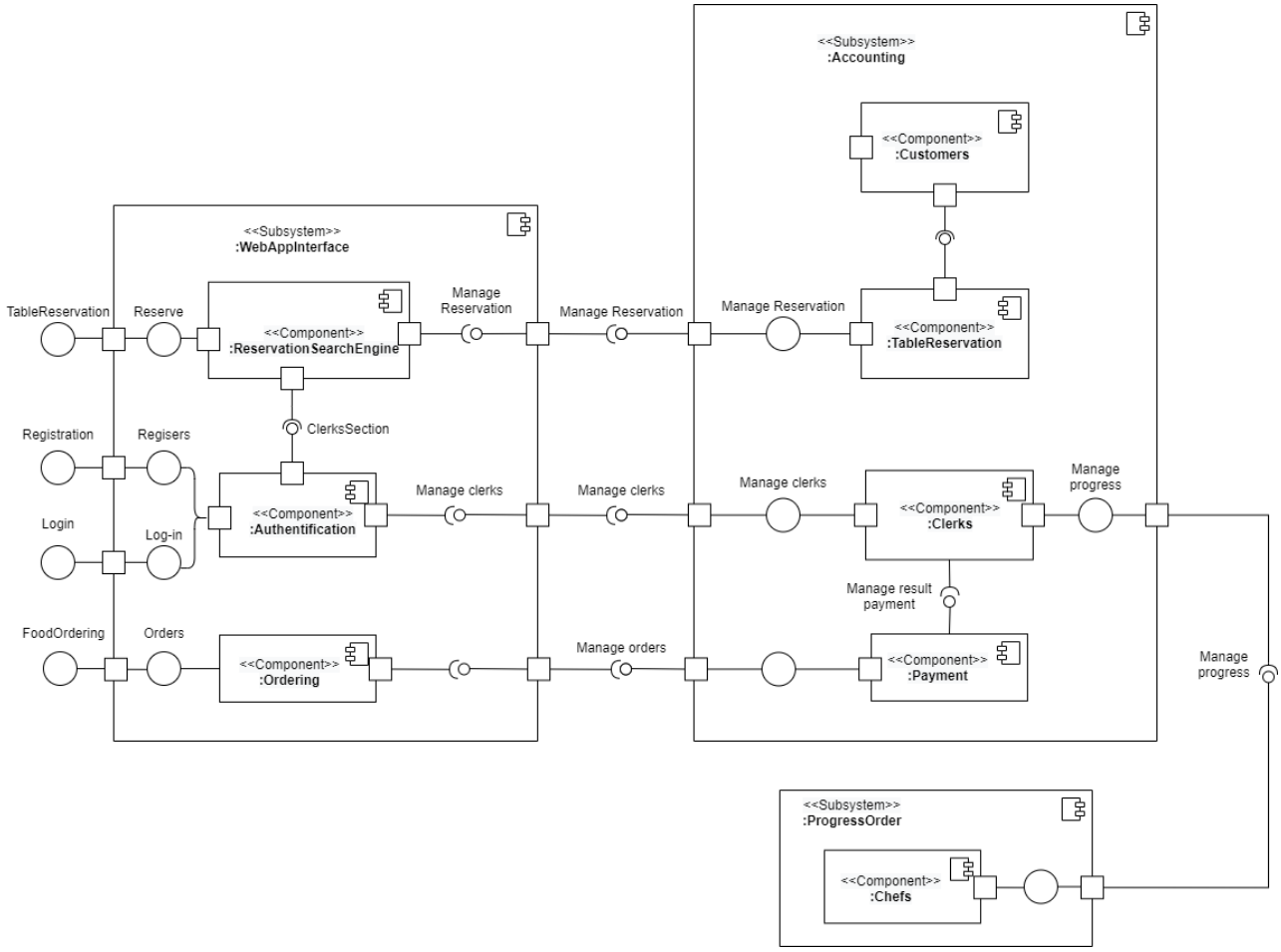
3.1 Describe architectural approach (MVC)



Description:

- The system will display the homepage. When a user interacts with the system, the action of user will send required to retrieve the data from the model and send the request to the controller if any new action is updating from the data. The controller collects the action from View and sends information updated to Model for process and manipulation, then sends the update to View.
- Controller: NestJS framework (Back-end). View: ReactJS framework (Front-end). Modal: MongoDB (Database).
- For example, Customer skims menu to choose the food, View send a request to the Model (database) to display food information, then the customer chooses the food, the action will send the Controller, the Controller will update the information to the Model (check out: add shopping cart, money) and the Model sends update information to the View.

3.2 Implementation diagram (Component Diagram)



Description:

- WebAppInterface subsystem contains components related to ReservationSearchEngine, Authentication, and Ordering. Search Engine component allows to search or browse items by table reservation such as status (reserved or unreserved), Identity (table numbering). Authentication component allows clerks to create account, login, or logout and binds customer to some account. Ordering component allows customer order foods, drinks, and manage that orders.
- Accounting subsystem provides three interfaces - Manage Reservation, Manage Clerks, and Manage Orders. TableRevervation component allows Customer check and manage table reservation. Clerks component is used in oder to manage clerks on restaurant. Finally, Payment component manage ordering of Customer and return result of transaction to Clerks.
- ProgressOrder subsystem provides an interface Mangage Progress used by Clerks component in Accounting subsystem to update progress status.