

SOFTWARE ENGINEER

RESTAURANT POS 2.0 SYSTEM	0
CHANGELOG	1
1. CONTEXT OF PROJECT	4
2. USE-CASE DIAGRAM	7
3. FUNCTIONAL REQUIREMENT	8
3.1 USE-CASE DETAIL FOR TABLE RESERVATION	8
3.2 USE-CASE DETAIL FEEDBACK	9
3.3 USE-CASE DETAIL FOR ORDER	10
3.4 USE-CASE DETAIL FOR PAY BILL	12
3.5 USE-CASE DETAIL FOR MANAGE ORDER	13
3.6 USE-CASE DETAIL FOR GET PAYMENT	14
3.7 USE-CASE DETAIL FOR MANAGE FOOD MATERIAL	15
3.8 USE-CASE DETAIL FOR MANAGE ACCEPTED ORDER	16
3.9 USE-CASE DETAIL FOR MANAGE STAFF	17
3.10 USE-CASE DETAIL FOR EDIT MENU	18
3.11 USE-CASE DETAIL FOR VIEW REPORT	19
4. NON-FUNCTIONAL REQUIREMENT	20
4.1 FOR THE WHOLE SYSTEM	20
4.2 FOR EACH USE-CASE	20
4.2.1 View menu	20
4.2.2 Order	21
4.2.3 Pay by mobile wallet	21
4.2.4 Pay by card	21
4.3.5 View list order	21
4.3.6 Notify order	21
4.3.7 Change quantity of food	22
4.3.8 View report	22
4.3.9 Edit menu	22
5. ACTIVITY DIAGRAM	23
5.1 Activity diagram of order activity	23
5.2. Activity diagram of chef	24
5.3 Activity diagram of restaurant manager	25
6. CLASS DIAGRAM	28
6.1 Class diagram of table reservation	28

6.2 Class diagram of food ordering	28
6.3 Class diagram of pay bill	28
6.4 Class diagram of menu	29
7. SEQUENCE DIAGRAM	29
7.1 Sequence diagram of table reservation	29
7.2 Sequence diagram of feedback	30
7.3 Sequence diagram of food ordering	31
7.4 Sequence diagram of pay bill	33
7.5 Sequence diagram of manage food material	34
7.6 Sequence diagram of manager accepted order	35
7.7 Sequence diagram of manage staff	36
7.8 Sequence diagram of edit menu	37
7.9 Sequence diagram of view report	38
8. ARCHITECTURE APPROACH	39
9. IMPLEMENTATION DIAGRAM	42
9.1. Component diagram	42
9.2. Deployment diagram	44

1. CONTEXT OF PROJECT

- **Context:** Countries around the world are being greatly affected by the COVID pandemic, leading to economic decline. Places where many people gather, especially restaurants, will have trouble in business if they do not come up with solutions to prevent infection in the community

	Traditional restaurant with some fast food property
Customer	All people, families
Payment	Pay first; Option: cash; credit.
Food	<ul style="list-style-type: none">- Cook in restaurant- Raw material: update quantity at begin of day- Order meal in table
Role	Customer; Staff, Chef, Waiter,...
Table	Order meal, table reservation
Description	<ul style="list-style-type: none">- Customer come to restaurant choose available table to sit- Using QR code on the table to access menu through website, make order (not make payment yet)- Clerk get the order, confirm it to customer make payment- Order sent to kitchen

- **Desires:** Convenience, fast, avoiding direct contact between employees and customers are the top priorities of the project.
- **Team Development's scope:** 5 members:

- + Nguyễn Tân Đạt
- + Nguyễn Thọ Nam
- + Phạm Đức Trọng
- + Trần Duy Chánh
- + Nguyễn Thành An

- **Stakeholders:**

Stakeholders	Reason - Role
Customers	Use system to place an order
Chefs	Receive orders and make food
Receptionists	Receive payment, coordinate the customers
Waiters	Guide the customers to scan the QR code, confirm orders from customers, serve food for customers when receive finished notification from kitchen
IT staff	Set up, design and maintain the system
Food Administration	Ensure the restaurant's operating system meets the standards in food hygiene and safety
Restaurant Managers	Ensure the system works as intended, manage staff, manage finances and adjust menus.

- **Expected functions (user story):**

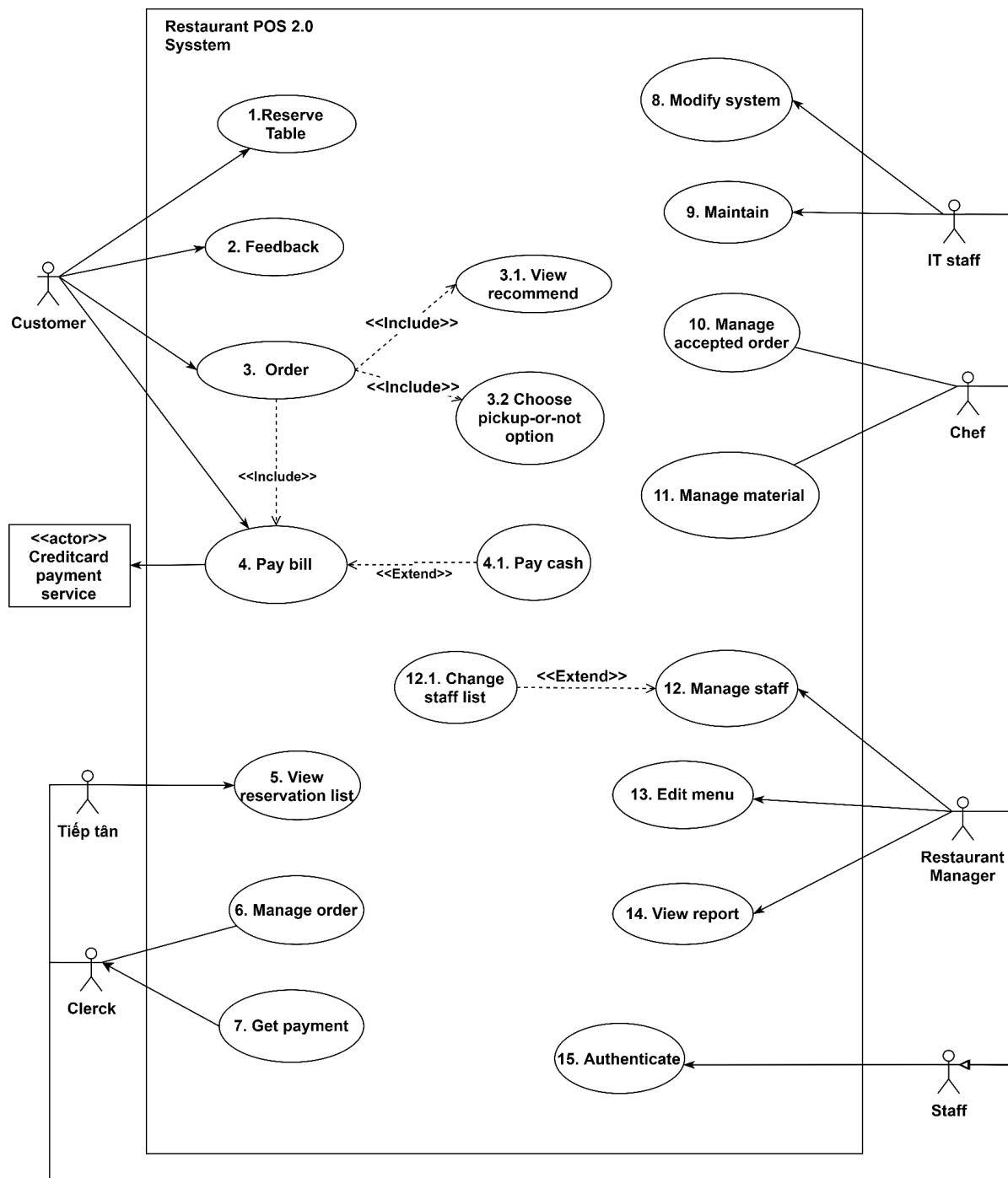
- + Customers:

- Booking table.
- Placing an order.
- Paying the bill.
- Rating and Sending feedback.
- + Waiters:
 - Managing order list.
 - Confirming the payments of customers.
- + Receptionists:
 - Managing the reservation list.
- + Chefs:
 - Managing the confirmed order list.
 - Managing the food warehouse.
- + Restaurant Managers:
 - Managing staffs.
 - Managing finances.
 - Managing the menu.
- + IT staff:
 - Maintaining the system.
 - Modifying the system.
- Estimated completion time: 4 month.

- **Project goal:**

- + Creating a safe workflow in restaurants during COVID pandemic.
- + Promoting the quickness and the convenience for customers and employees.
- + Turning the project into a trend in the future and become an essential part of restaurant development.

2. USE-CASE DIAGRAM



Hình 1: Use-case diagram

3. FUNCTIONAL REQUIREMENT

3.1 USE-CASE DETAIL FOR TABLE RESERVATION

Use Case ID:	01		
Use Case Name:	Table reservation		
Created By:	Nguyễn Tấn Đạt	Last Updated By:	Nguyễn Tấn Đạt
Date Created:	25/09/2021	Date Latest Updated:	25/09/2021
Actors	Customer		
Description:	The customer want to reserve table before come to the restaurant		
Trigger:	The customer click reserve button on order screen		
Preconditions:	1. Customer is in order menu		
Postcondition	1. Customer reserve table successfully 2. Their information sent to receptionist		
Normal Flow:	1. The system displays the list of free table 2. The customer chooses the available table to be reserve 3. The system displays menu for customer leave information 4. The customer fill his/her information 5. The customer confirm table reservation		
Exceptions:	<p>Exception 1: at step 4 (The customer don't fill necessary information) The system displays a inform and request retry step 4</p> <p>Exception 2: at step 5 (The customer cancel reservation) System turn back to order menu</p>		

Non Interactive functional requirement:

- The interface displays table positions and their status.
- The system records the reservation information into databases.

3.2 USE-CASE DETAIL FEEDBACK

Use Case ID:	02
Use Case Name:	Feedback
Created By: Nguyễn Tân Đạt	Last Updated By: Nguyễn Tân Đạt
Date Created: 25/09/2021	Date Last Updated: 25/09/2021
Actors:	Customer
Description:	Customer wants to send feedback of service to restaurant manager
Trigger:	Choose feedback button on the app screen
Preconditions:	The system has been accessed
Postconditions:	Feedback is sent to restaurant manager
Normal flow:	1. The system show a form for customer give feedback 2. The customer write down feedback 3. The customer confirm feedback
Alternative Flows	None
Exceptions flow:	Exception 1: at step 2 (Customer does fill all information) System notify and retry from step 2 Exception 2: at step 3 (The customer cancel feedback) The system turn back to main menu

Non Interactive functional requirement:

- The system records orders into the database.
- The Feedback icon always appears in the customer's interface.
- The Feedback will be shown directly in the manager's interface.

3.3 USE-CASE DETAIL FOR ORDER

Use Case ID	3		
Use Case Name	Food ordering		
Created By	Nguyen Thanh An	Last Updated By:	Nguyen Thanh An
Date Created	25/9/2021	Date Last Updated:	25/9/2021
Actor	Customer		
Description	Customer wants to place an order		
Trigger	Customer press “FOR CUSTOMER” button on Web app		
Preconditions	1. Customer places an order 2. Mobile devices have internet connection		
Postconditions	Customer orders successfully and wait for food		
Normal Flow	<ol style="list-style-type: none">1. Customer scans QR code or enter URL on Web browser2. Customer press “FOR CUSTOMER” button on Web app3. System displays the menu4. Customer choose dishes and quantity5. System display the list of ordered dishes and total prices6. Customer press the “PAYMENT” button on the screen after authenticating the final price7. Customer takes the bill		

Exceptions	<ol style="list-style-type: none"> 1. Customer does not have enough cash or money in credit card / digital wallet to pay <ul style="list-style-type: none"> - System notifies that customer does not have enough money to pay the bill - Cancel the bill or re-order 2. System will cancel the order if customer's device loss internet connection
Alternative Flows	<p>Customer wants to cancel the order (Before or when the clerk confirm the order)</p> <ul style="list-style-type: none"> - Customer press "CANCEL THE ORDER" button on the screen - Customer confirm to cancel the order - Web App returns to the Order Interface in Normal Flow.

Non Interactive functional requirement:

- The interface displays a dish on the menu that must contain both an image and a name.
- The display will update with the database regularly, providing the status of remaining food every 15 seconds..
- The system records orders into the database.
- The display will update with the database regularly, providing the number of remaining food every 1 minute

3.4 USE-CASE DETAIL FOR PAY BILL

Use Case ID:	4		
Use Case Name:	Pay Bill		
Created By:	Nguyễn Tấn Đạt	Last Updated By:	Nguyễn Tấn Đạt
Date Created:	22/06/2021	Date Latest Updated:	22/06/2021
Actors	Customer, Credit Card payment service		
Description:	The customer wants to pay for the order.		

Trigger:	The customer press payment button on Web app
Preconditions:	1. Customer has confirmed order
Postconditions:	1. Customer make payment successfully 2. Order is sent to clerk
Normal Flow:	1. Customer choose the payment button 2. System display total cost and requires payment 3. Customer makes payment 4. System sent information of the payment to payment service 5. Payment service confirm the payment of customer 6. System display the confirm order to customer
Alternative Flow:	<p><u>Alternative 1:</u> at step 1: 1a: Customer choose to pay by cash 1b: System inform the clerk to get the cash</p> <p><u>Alternative 2:</u> at step 3: 3a: The customer can press the cancel button on the interface. 3b: The touch screen returns to the menu interface.</p>
Exceptions:	<p><u>Exception 1:</u> at step 4 (Payment failed) 4.1 The system requests the customer to retry step 2 → 3 in normal flow until successful or the customer can cancel payment by card.</p> <p><u>Exception 2:</u> at step 6 (A credit card doesn't have enough money to pay) 6.1 The system notifies the credit card that it doesn't have enough money to pay, requests the customer to choose another payment type or cancel the payment.</p>

Non Interactive functional requirement:

- QR code displays on the screen in 60 seconds. After the 60s, the system updated the new one.
- Online payment service is compatible with Momo wallet, Air Pay, Zalo Pay and Payoo.
- Customer enters or scans the incorrect code, QR code, PIN code at most 5 times.
- The payment time is no more than 10 minutes.

3.5 USE-CASE DETAIL FOR MANAGE ORDER

Use Case ID:	06		
Use Case Name:	Manage order		
Created By:	Nguyễn Tấn Đạt	Last Updated By:	Nguyễn Tấn Đạt
Date Created:	22/06/2021	Date Latest Updated:	25/06/2021
Actors	Clerk		
Description:	The clerk want to manage the order of customer before sent it to chef		
Trigger:	1. Clerk click clerk button on his app 2. System receive order from customer		
Preconditions:	1. Clerk has login to his account in the app		
Postconditions:	1. The order is confirm 2. Order is sent to chef		
Normal Flow:	1. System display the order list of customer 2. Clerk choose the order to check 3. Clerk confirm the order 4. System sent information of the payment to payment service 5. Payment service confirm the payment of customer 6. System display the confirm order to customer		
Alternative Flow:	Alternative 1: at step 3: 3a: Clerk choose cancel the order of customer button 3b: System remove the order 3c: System return to step 1		
Exceptions:	Exception 2: at step 3 3.1: If the clerk don't choose any order, system show an inform and return to step 1		

Non Interactive functional requirement:

- List of orders updated regularly, every 15 seconds.

3.6 USE-CASE DETAIL FOR GET PAYMENT

Use Case ID	07		
Use Case Name	Change quantity of dishes		
Created By	Nguyễn Tấn Đạt	Last Updated By:	Nguyễn Tấn Đạt
Date Created	22/06/2021	Date Latest Updated:	25/06/2021
Actor	Clerk		
Description	Customer want to pay bill by cash, clerk will receive payment		
Trigger	The number of food changes is out of the chef's control.		
Preconditions	1. Customer choose pay by cash option 2. Order has been accept		
Postconditions	Payment is successfully completed		
Normal Flow	1. System inform clerk information of order 2. Clerk confirm the payment		
Alternative flows	<u>Alternative 1:</u> at step 2 2a: Clerk cancel the payment		
Exceptions	None		

3.7 USE-CASE DETAIL FOR MANAGE FOOD MATERIAL

Use Case ID	11		
Use Case Name	Manage food material		
Created By	Nguyễn Tấn Đạt	Last Updated By:	Nguyễn Tấn Đạt
Date Created	25/06/2021	Date Latest Updated:	25/06/2021
Actor	Chef		

Description	Chef want to update the quantity of food to inform which meal can be served
Trigger	1. Chef choose manage food button on his app 2. Chef confirm an order has been completed
Preconditions	1. Chef has been login to his staff account 2. The screen is displaying the system homepage
Postconditions	Quantity of food updated correctly
Normal Flow	<ol style="list-style-type: none"> 1. System show an table of information of available food 2. Chef change the quantity of them 3. Chef confirm quantity of food 4. System process and notify update quantity of food successfully
Alternative flows	Alternative 1: at step 2 2a: Jump to step 3
Exceptions	None

3.8 USE-CASE DETAIL FOR MANAGE ACCEPTED ORDER

Use Case ID:	11		
Use Case Name:	Manage accepted order		
Created By:	Nguyễn Tấn Đạt	Last Updated By:	Nguyễn Tấn Đạt
Date Created:	25/06/2021	Date Latest Updated:	25/06/2021
Actors	Chef		
Description:	As a chef, I want to see the list of orders and inform that which order has been done		
Trigger:	Chef requests to open a list of orders		
Preconditions:	1. The customer has ordered food and has paid the bill 2. The screen is displaying the system homepage		

PostConditions:	The chef views a list of accepted orders ordered from customers
Normal Flow:	<ol style="list-style-type: none"> 1. The chef chooses “View Dishes” on the screen of the system. 2. System show the list of order ordered from customer
Alternative Flow:	<p>Alternative 1: at step 2</p> <ol style="list-style-type: none"> 3a. he chef chooses the order on the screen 3b. The chef press button “complete” on the screen 3c. The system update the order state, notify the change applied, turn back to order list
Exceptions:	<p>Exception 1: * at any time</p> <p>Chef doesn't want to update information of order, he can choose “X” button on screen to cancel and turn back to order list</p>

Non Interactive functional requirement:

- Orders are sorted in ascending order by order number.
- Each order needs to display the order number, order time, the number of dishes ordered.

3.9 USE-CASE DETAIL FOR MANAGE STAFF

Use Case ID:	12		
Use Case Name:	Manage staff		
Created By:	Nguyễn Tấn Đạt	Last Updated By:	Nguyễn Tấn Đạt
Date Created:	25/06/2021	Date Latest Updated:	25/06/2021
Actors	Restaurant manager		
Description:	Restaurant manager want to add or remove staff of the restaurant		
Trigger:	Restaurant manager click “Manage staff” button on manager menu		
Preconditions:	Restaurant manager has been login to his account		

PostConditions:	Staff list is updated
Normal Flow:	<ol style="list-style-type: none"> 1. System display the list of staff 2. Manager choose the staff 3. System display new window with information of that employee 4. Manager change information of that staff 5. Manager confirm the change 6. System notify that staff information has been updated 7. System return to staff list
Alternative Flow:	<p>Alternative 1: at step 4</p> <ol style="list-style-type: none"> 4a: Manager choose “delete” button on the screen 4b: System display a new window ask confirm 4c: Manager choose “confirm” button on the screen 4d: System notify that staff information has been updated <p>Continue at step 7</p>
Exceptions:	<p>Exception 1: * at any time</p> <p>Manager doesn't want to change staff list, he can choose “X” button on screen to cancel and turn back to staff list</p>

3.10 USE-CASE DETAIL FOR EDIT MENU

Use Case ID:	13		
Use Case Name:	Edit menu		
Created By:	Nguyễn Tấn Đạt	Last Updated By:	Nguyễn Tấn Đạt
Date Created:	25/06/2021	Date Latest Updated:	25/06/2021
Actors	Restaurant manager		
Description:	Restaurant manager want to add or remove meal from menu of the restaurant		
Trigger:	Restaurant manager click “Edit menu” button on manager menu		
Preconditions:	Restaurant manager has been login to his account		
PostConditions:	Menu is updated		

Normal Flow:	<ol style="list-style-type: none"> 1. System display the list of meal in menu 2. Manager choose the meal to edit 3. System display new window with information of that meal 4. Manager choose “delete” button 6. System notify that meal has been deleted 7. System return to menu
Alternative Flow:	<p>Alternative 1: at step 2</p> <ol style="list-style-type: none"> 4a: Manager choose “add” button 4b: System display new window 4c: Manager add information off new meal and choose “confirm” button 4d: System notify that meal has been added and return to menu
Exceptions:	<p>Exception 1: * at any time</p> <p>Manager doesn't want to cancel the change of menu, he can choose “X” button on screen to cancel and turn back to staff list</p>

3.11 USE-CASE DETAIL FOR VIEW REPORT

Use Case ID:	14		
Use Case Name:	View report		
Created By:	Nguyễn Tấn Đạt	Last Updated By:	Nguyễn Tấn Đạt
Date Created:	25/06/2021	Date Latest Updated:	25/06/2021
Actors	Restaurant manager		
Description:	Restaurant manager want to see report of the restaurant		
Trigger:	Restaurant manager click “report” button on manager menu		
Preconditions:	Restaurant manager has been login to his account		
PostConditions:	Restaurant manager see report of restaurant		
Normal Flow:	<ol style="list-style-type: none"> 1. System display the new window with information of restaurant finance 2. Manager choose time on the screen and click “create” button 		

	3. System show the report of that month
Alternative Flow:	<u>none</u>
Exceptions:	<p>Exception 1: * at any time Manager doesn't want to cancel the change of menu, he can choose "X" button on screen to cancel and turn back to manager menu</p> <p>Exception 2: at step 2 Manager choose time is invalid, the system notify error and return to step 1</p>

Non Interactive functional requirement:

- Each time the manager views a report, the system will record the manager name, time, and content viewed. No one can edit this log information.
- System stores available reports of the last 1 month, 2 months. When the report is requested, it is not necessary to recalculate.
- The download file is in pdf format or in xlsx format.
- Report is automatically generated every weekend, every first day of months and years.

4. NON-FUNCTIONAL REQUIREMENT

4.1 FOR THE WHOLE SYSTEM

1. User interface with Vietnamese support, easy to use (8/10 people can use it in 5 minutes to get acquainted).
2. The system can receive up to 300 people per day.
3. Can operate at least 15 hours per day.
4. Comply with food hygiene and safety laws (55/2010/QH12).

5. The system can be easily upgraded and scale (deploy new features to the entire system in less than 1 week, and can be used in many restaurants).

4.2 FOR EACH USE-CASE

4.2.1 View menu

- Order interfaces should be easy to view (text size larger than 12 pt for mobile devices and 30 pt for tablet) and have white and red as main colors.

4.2.2 Order

- The system can accommodate simultaneously for up to 50 orders in the order list.
- The average time of order operation on the screen is less than 1 minute.

4.2.3 Pay by mobile wallet

- The barcode creation is in 30 seconds.
- Percentage of wrong bills in verification is less than 1%.
- Bar code is satisfied with ISO/IEC specification.

4.3.4 Pay by card

- Customers' credit and debit card information is encrypted before it is ever stored on database servers (use Tokenization).
- Follow PCI DSS (Payment Card Industry Data Security Standard).

Note: PCI DSS was put into place to protect consumers and businesses by creating a certain regulatory framework that provides

a universal standard for how to handle, use, and store credit card information.

4.3.5 View list order

- The time to collect orders is no more than 5 seconds.
- The rate of not viewing the order list although having available orders is less than 3%.

4.3.6 Notify order

- System responds to all actions of the chef and customer within 2s.
- The failure percentage of the system is less than 1%.

4.3.7 Change quantity of food

- The system updates changes in less than 10 seconds.
- The percentage of occurring errors in updating is no more than 3%.

4.3.8 View report

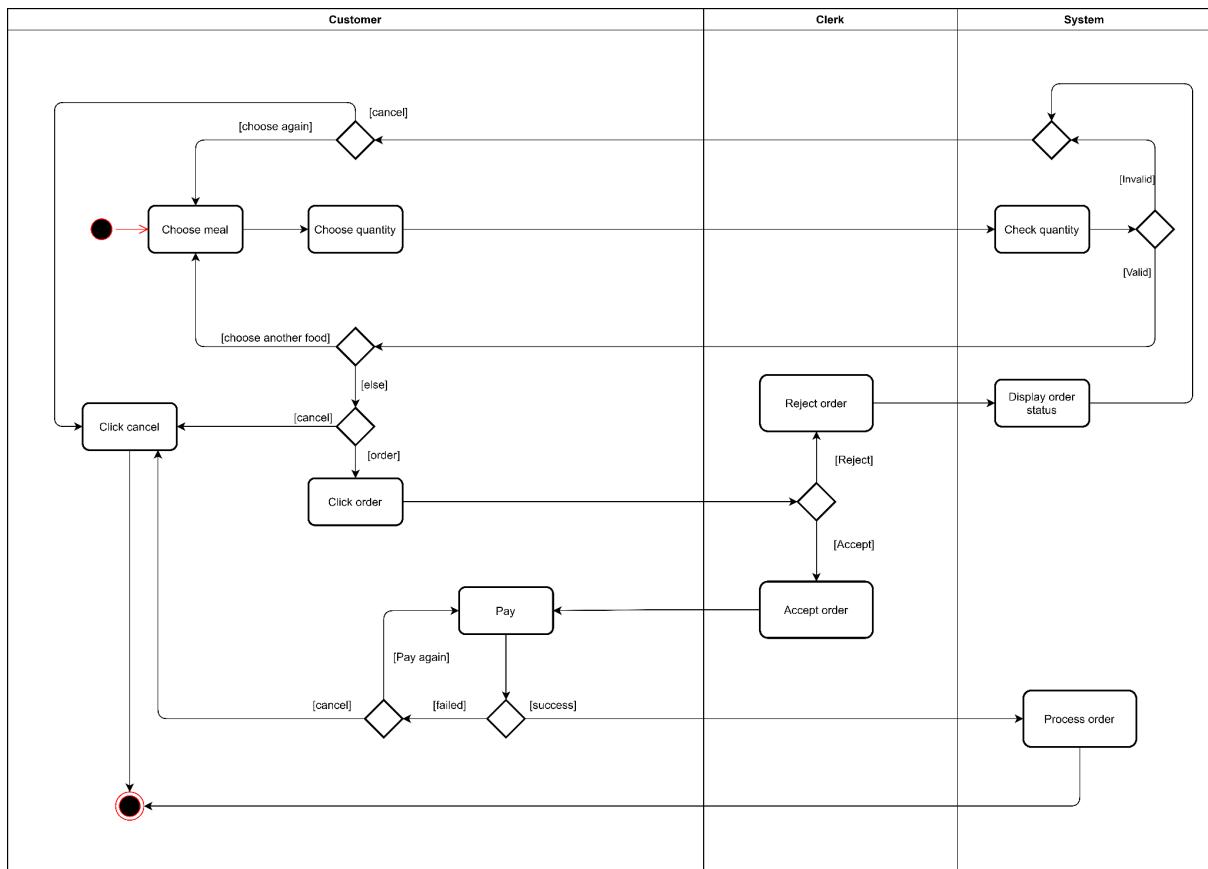
- Time to generate a report does not exceed 10 seconds.

4.3.9 Edit menu

- System responds to all actions of the manager within 1s.
- The failure percentage of the system is less than 2%.

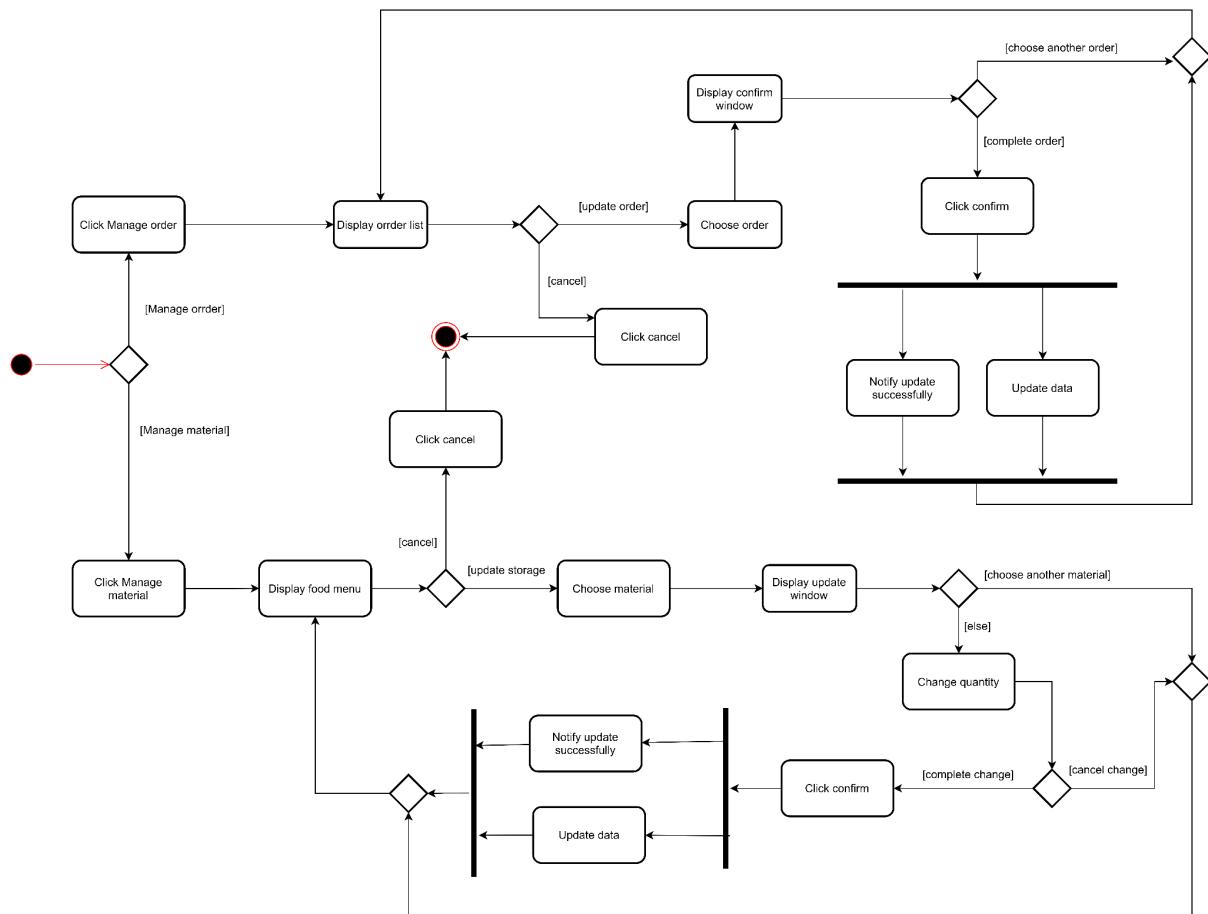
5.ACTIVITY DIAGRAM

5.1 Activity diagram of order activity



- Customer can choose meal and make order, pay the bill, the system then update the database
- However, the customer can interrupt the process to come back to previous activity
- When the customer want to quit, the click 'cancel' to end the activity

5.2. Activity diagram of chef



Chef can manage accepted order list or manage material quantity

5.3 Activity diagram of restaurant manager

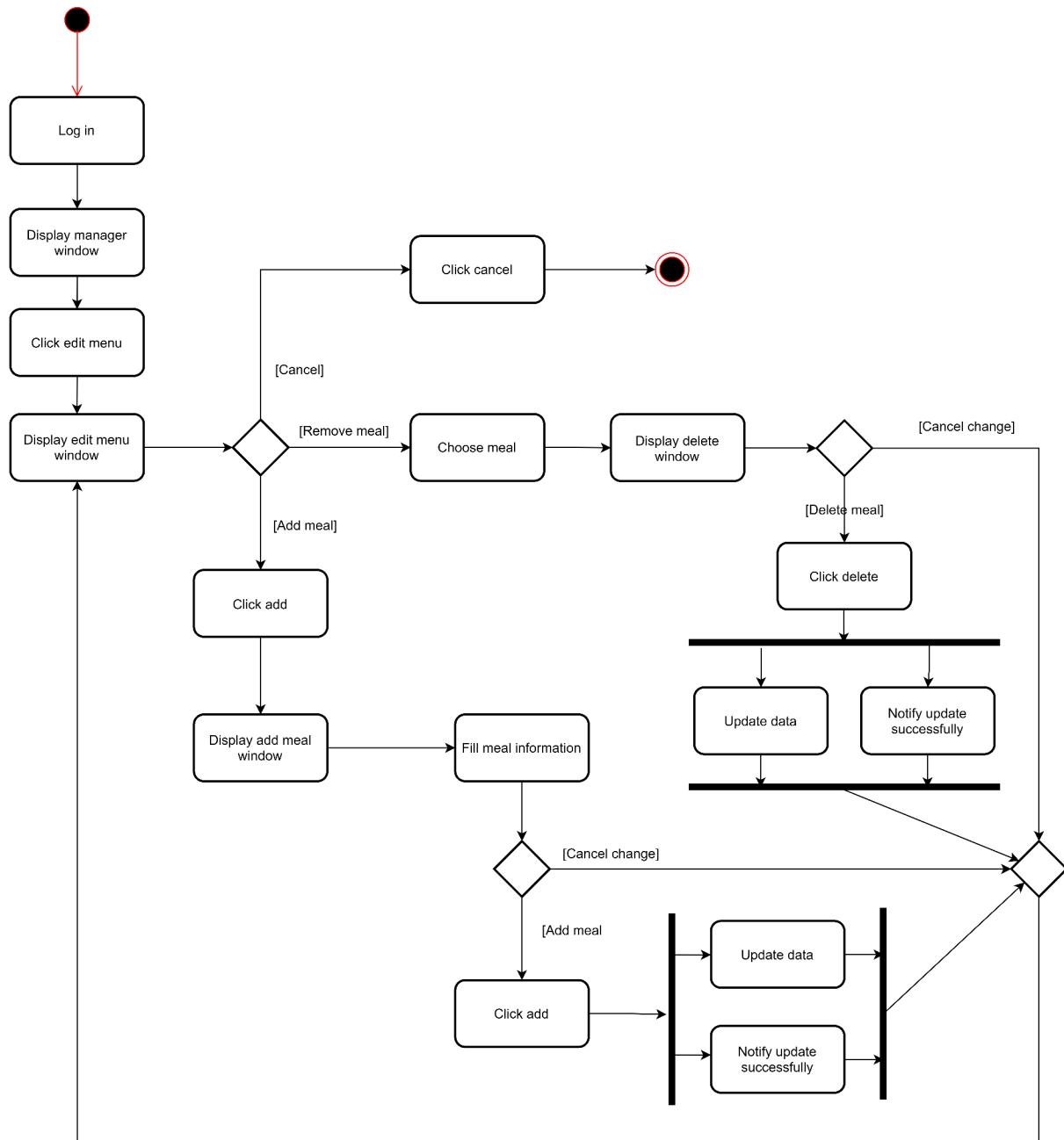
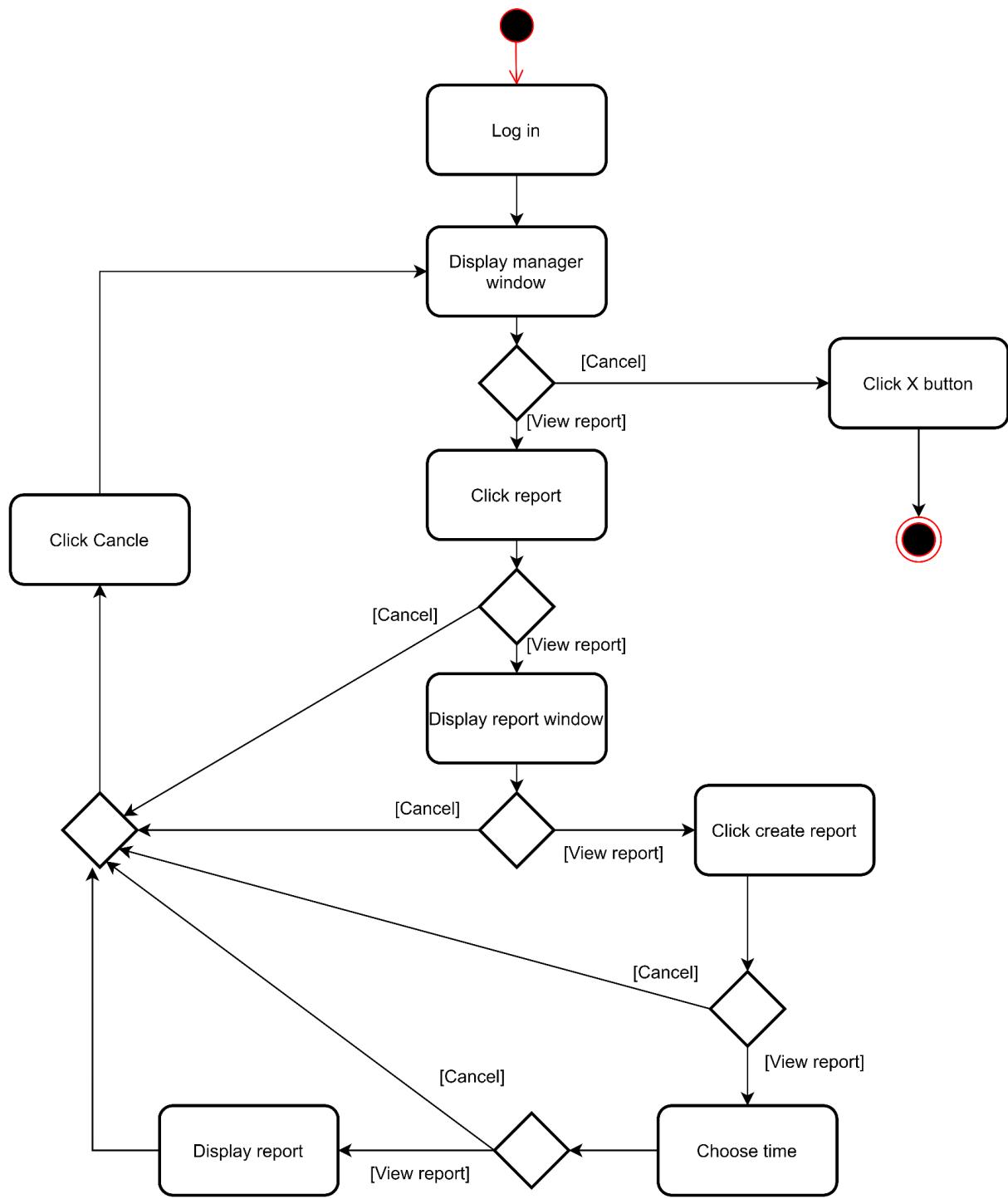
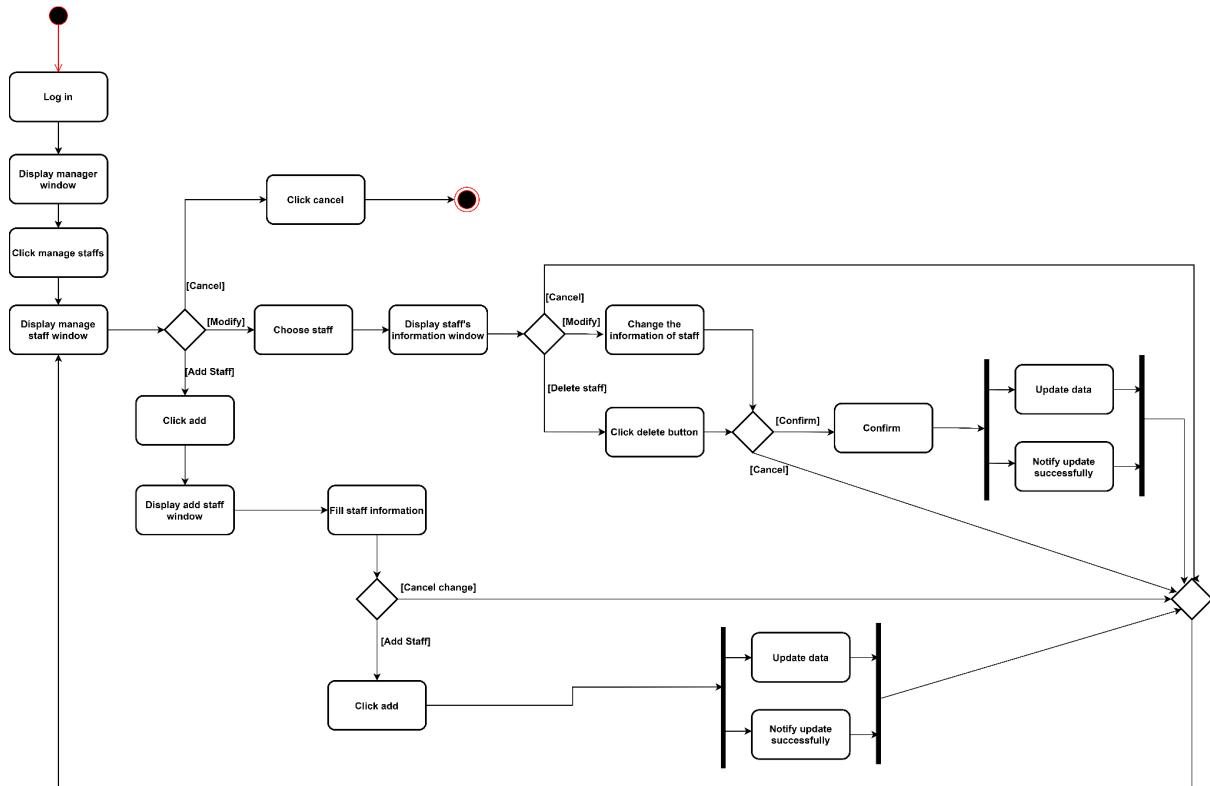


Diagram for edit menu activity



Create and view report of the restaurant



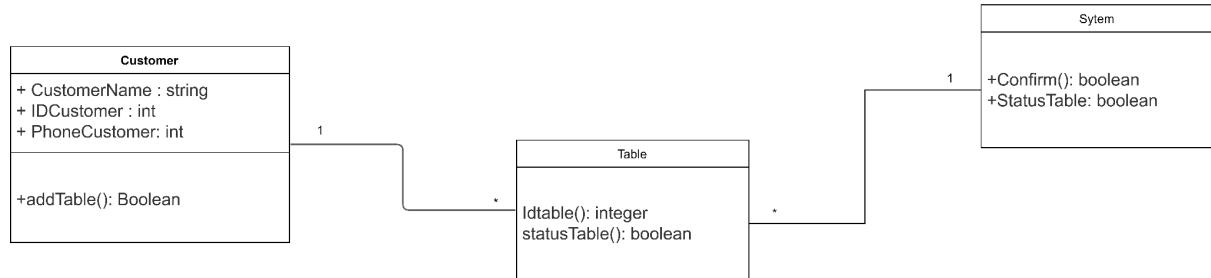
manage employee

6.CLASS DIAGRAM

6.1 Class diagram of table reservation

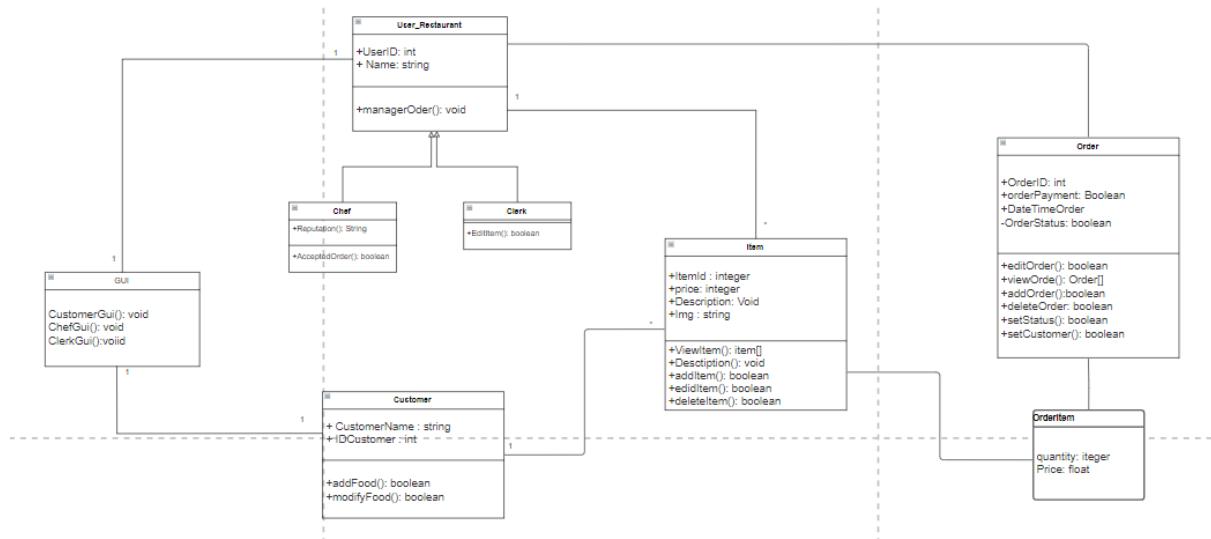
Customers view and select the table they want to book, the system will save and change the status of the table.

TABLE RESERVATION



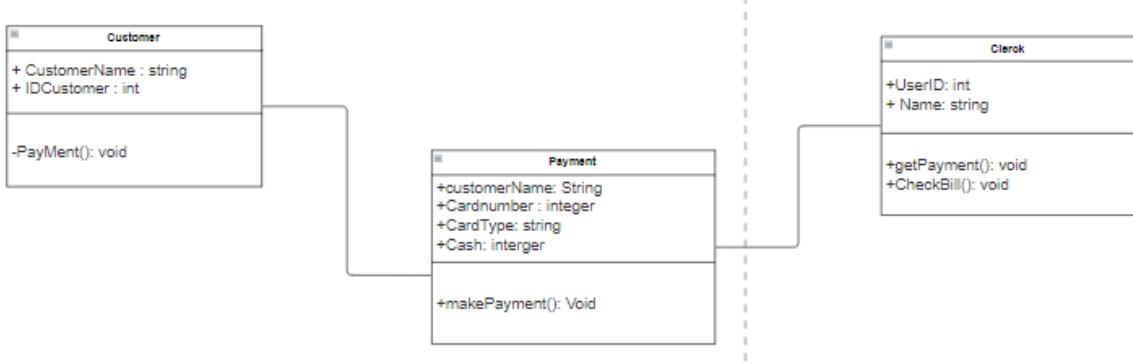
6.2 Class diagram of food ordering

Each dish will have a different item, in the item contains information about that dish. Customers choose the food item they want to order and send it to the system. There is a Gui for customers and staff, chefs to interact with each other.



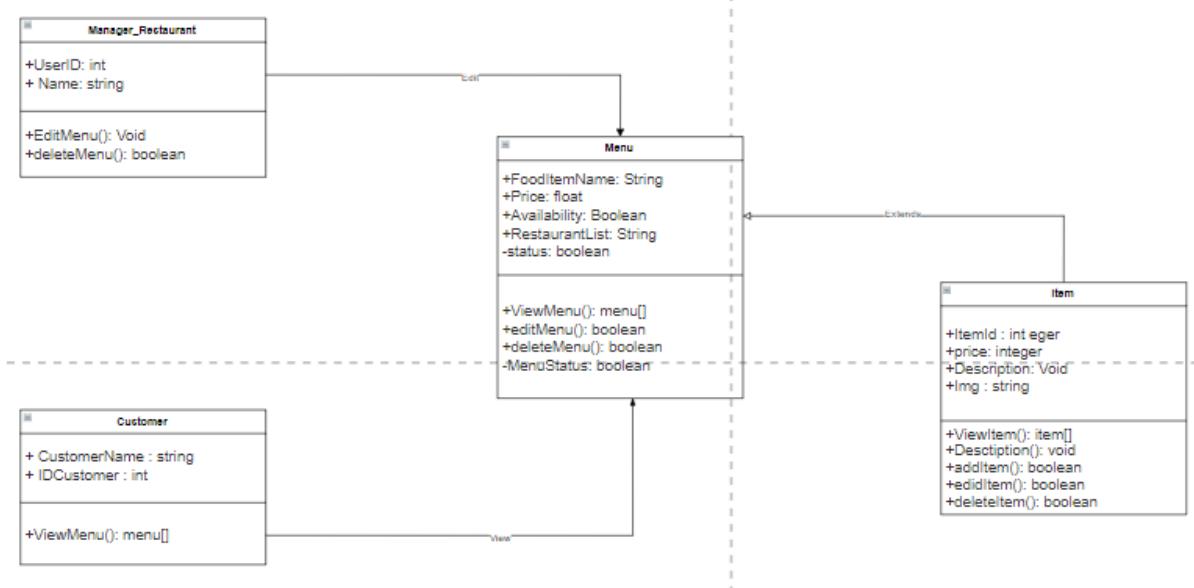
6.3 Class diagram of pay bill

Customers will make payment by card (if desired). In case of cash payment, the cashier will be contacted.



6.4 Class diagram of menu

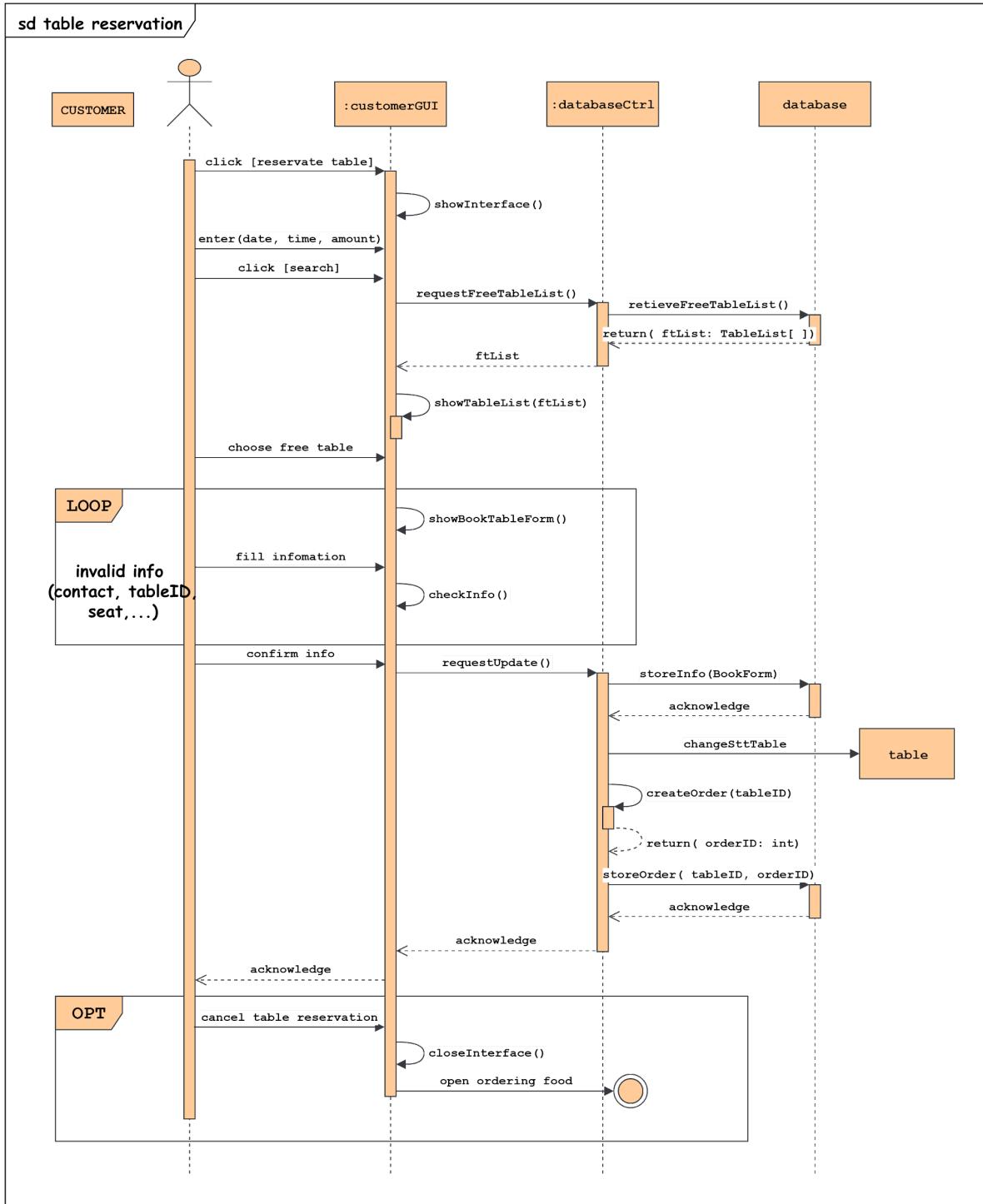
The restaurant's manager has the right to edit the menu (add, delete, edit). Customers can only view the menu and from there will order their food.



7. SEQUENCE DIAGRAM

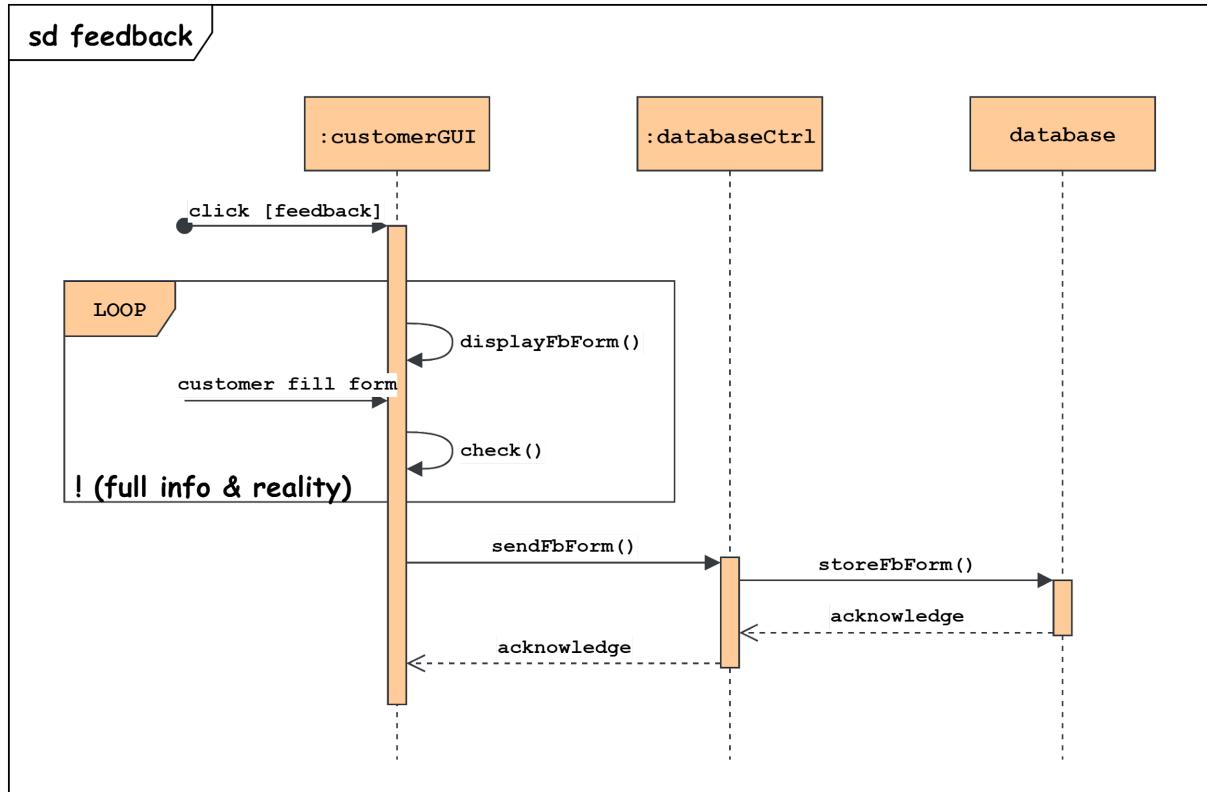
7.1 Sequence diagram of table reservation

This case main role is arranging the table, customers want to reserve before coming to the restaurant, notify and update the system.



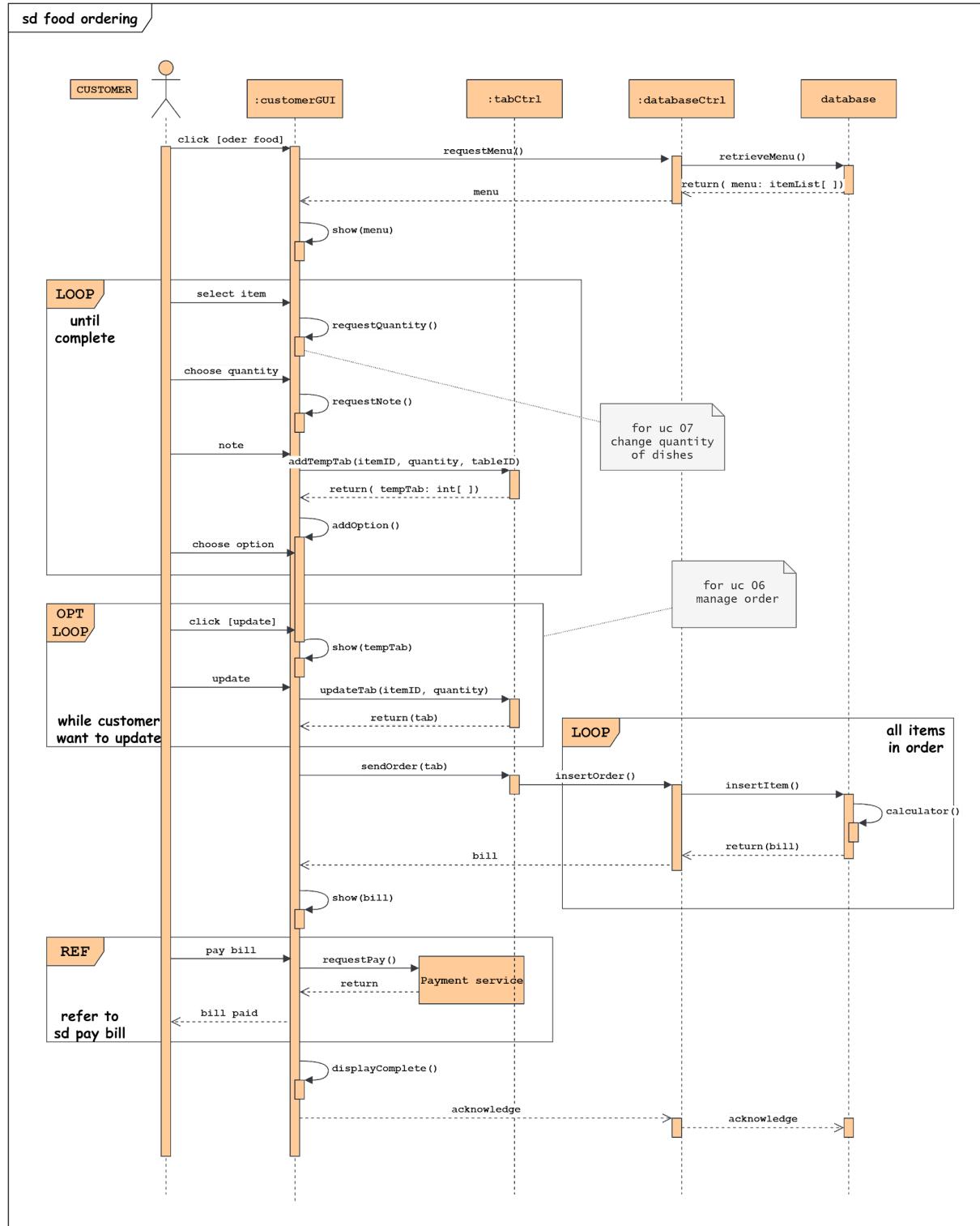
7.2 Sequence diagram of feedback

This diagram describes the situation when the customer wants to give feedback about the quality of the food and the service.



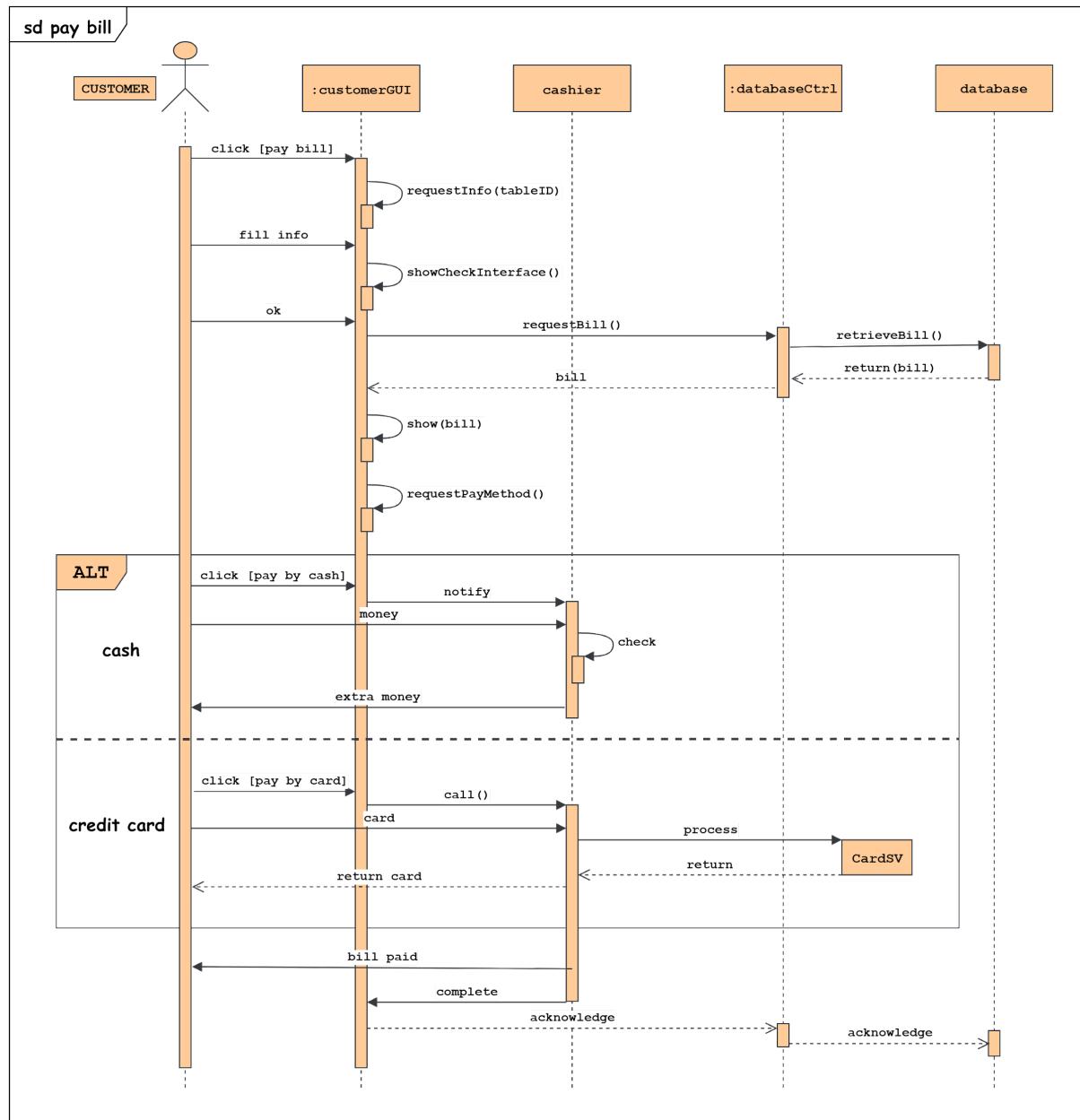
7.3 Sequence diagram of food ordering

This diagram describes the situation of making or modifying orders and saving them into the database and forward to the chef. Customers can do that by using GUI and the interface to communicate with the menu in the database. When there is a new order, GUI will save it in the database using a database controller. This helps keep the Low Coupling Principle because it just starts when users interact with it.



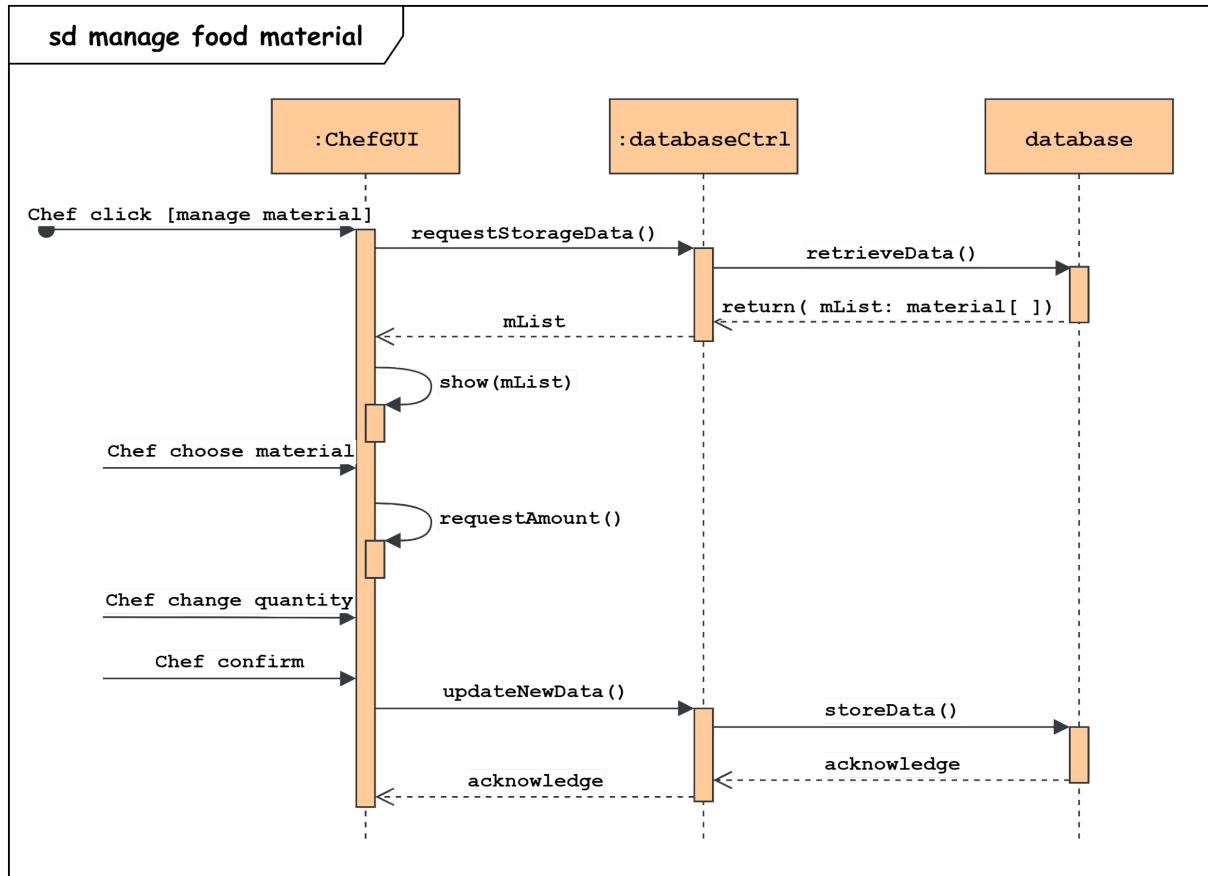
7.4 Sequence diagram of pay bill

This diagram describes the situation in which the customer pays the bill with two methods: pay by cash or pay by credit card.



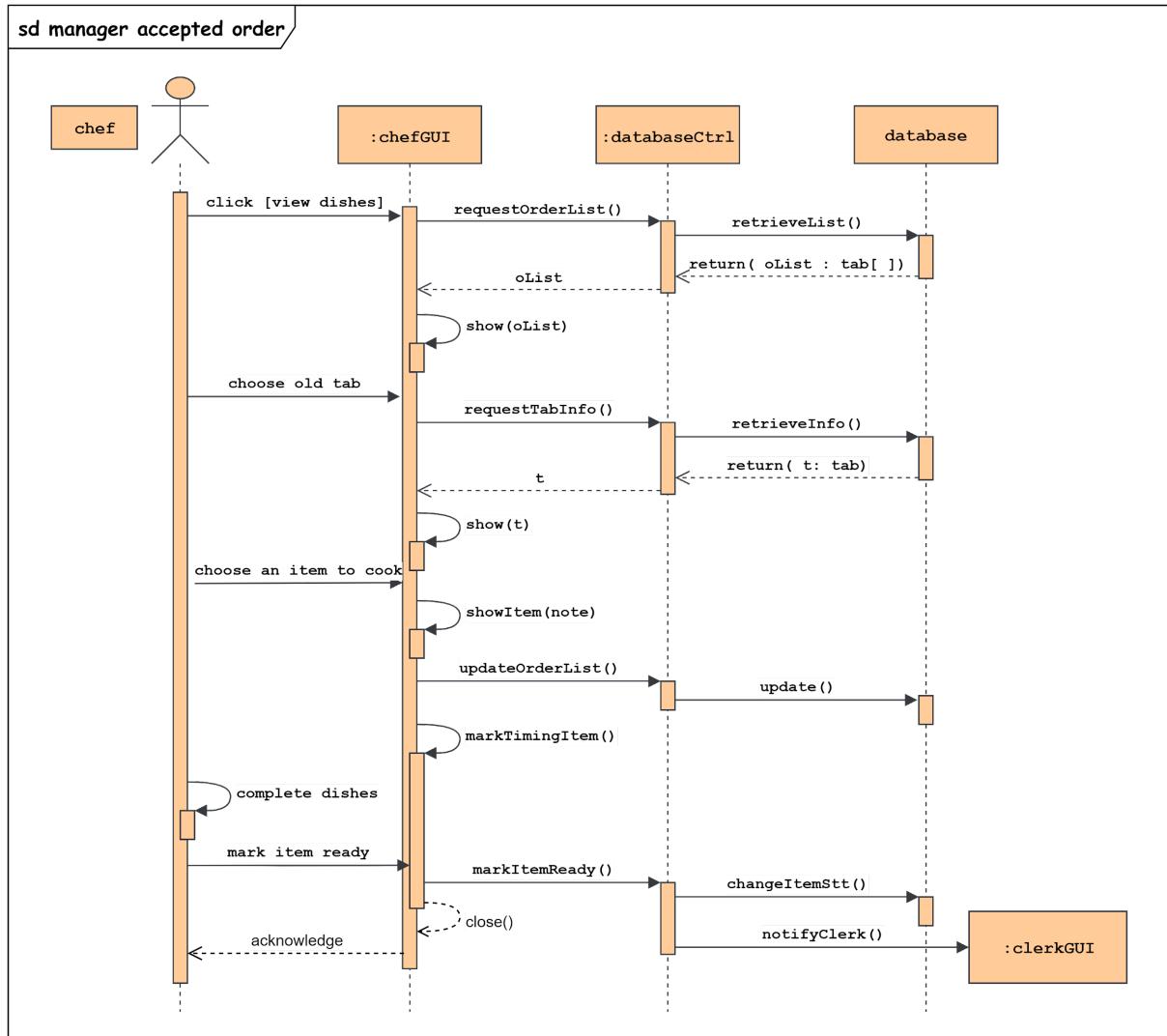
7.5 Sequence diagram of manage food material

This diagram describes the situation in which the chef changes the quantity of food ingredients whenever they want.



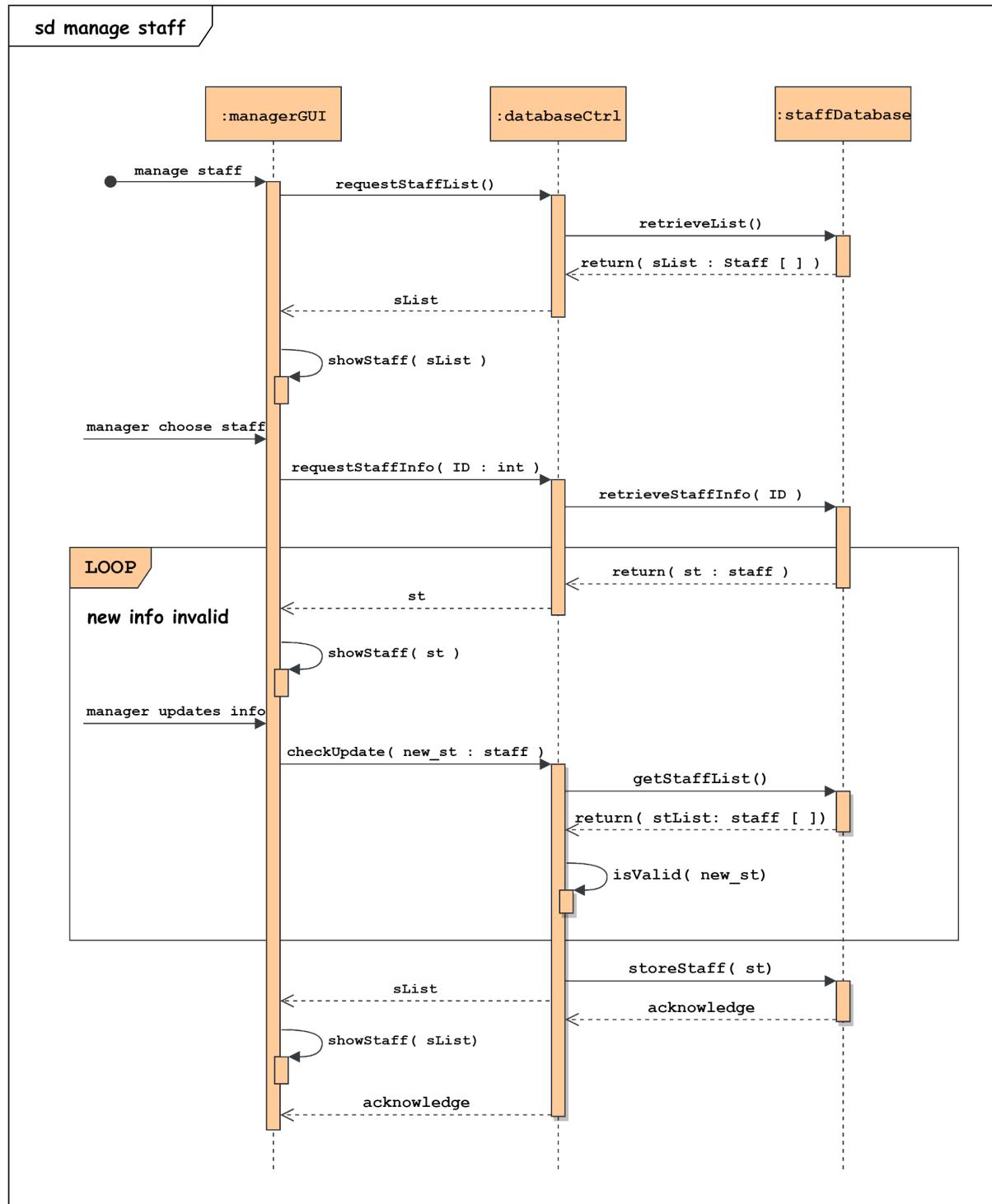
7.6 Sequence diagram of manager accepted order

This diagram describes the situation of the process of the chef taking the order until the food is ready to serve.



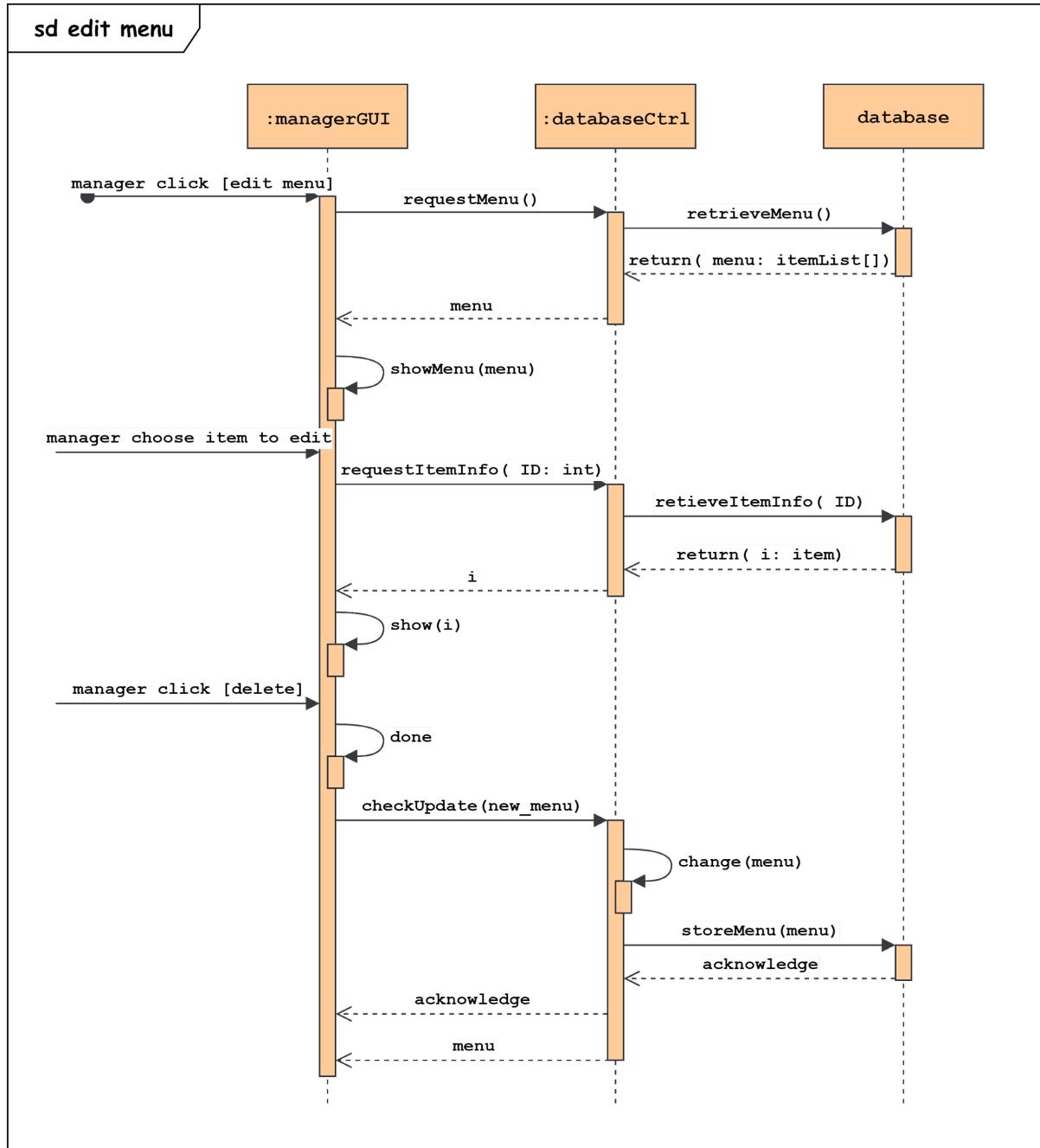
7.7 Sequence diagram of manage staff

This diagram describes the situation in which managers access staff-databases and manage information of employees.



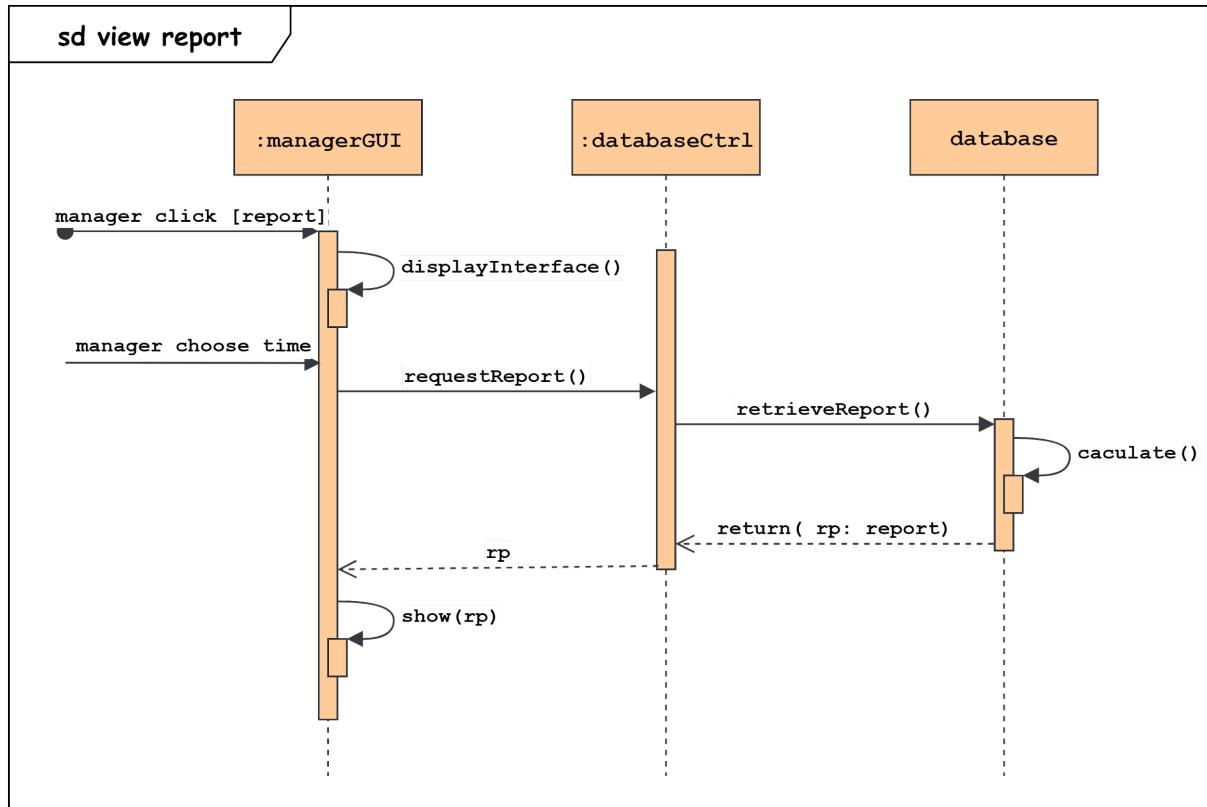
7.8 Sequence diagram of edit menu

This diagram describes the situation in which a restaurant manager wants to change the menu, add or delete meals (example for delete meal) . After that, a new menu will be updated in the database.



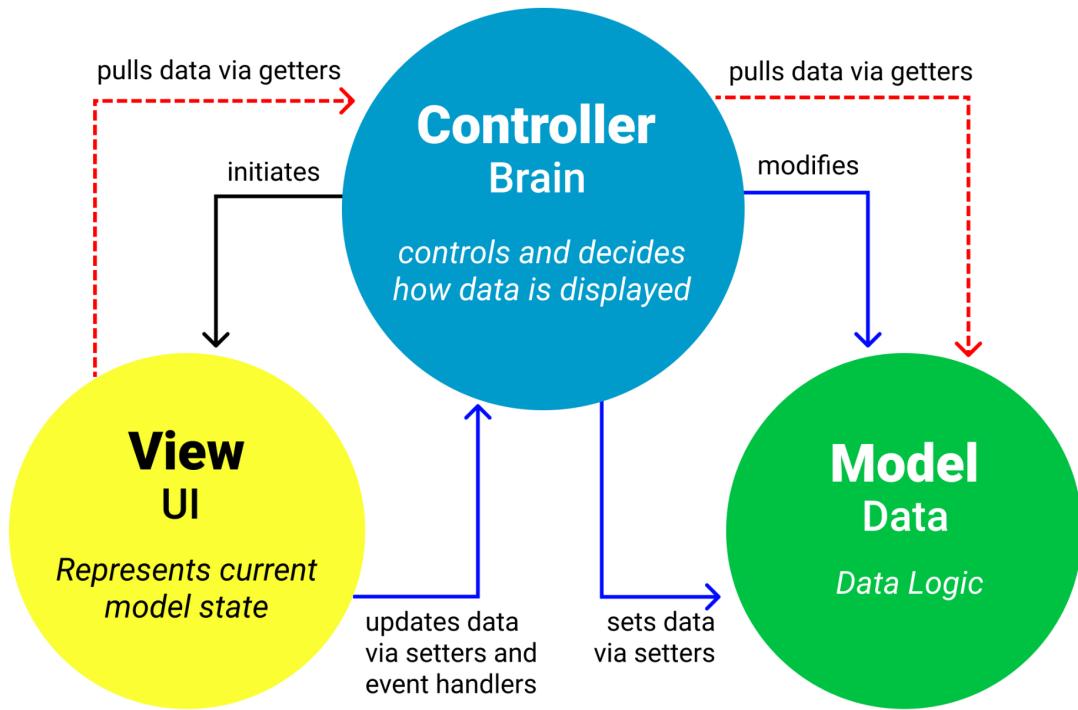
7.9 Sequence diagram of view report

This diagram describes the situation in which the manager wants to view reports whenever he wants.



8. ARCHITECTURE APPROACH

MVC Architecture Pattern



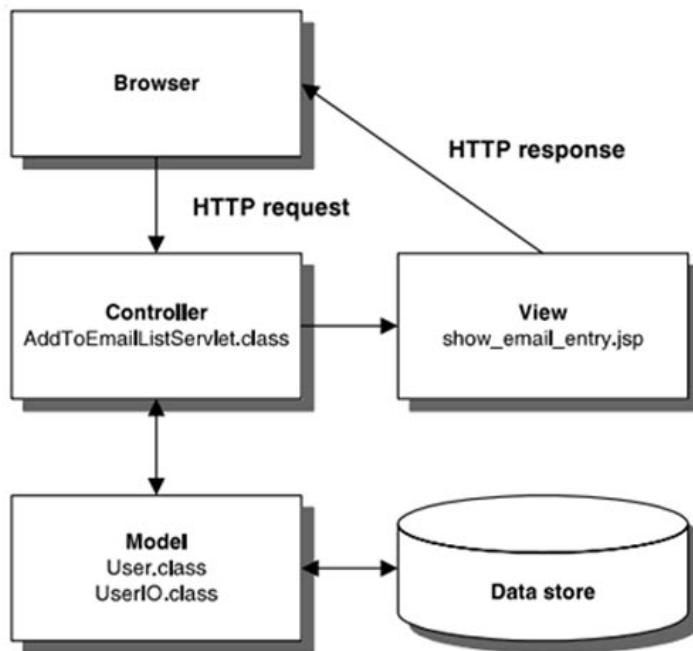
- **Description:** MVC Pattern is an architectural pattern that separates presentation and interaction from the system data. The system is structured into three logical components that interact with each other:
 - **Model component:** manages the system data and associated operations on that data. The Model component corresponds to all the data-related logic that the user works with. This can represent either the data that is being transferred between the View and Controller components or any other business logic-related data

- **View component:** defines and manages how the data is presented to the user. The View component is used for all the UI logic of the application
- **Controller component:** manages user interaction and passes these interactions to the View and the Model. Controllers act as an interface between Model and View components to process all the business logic and incoming requests, manipulate data using the Model component and interact with the Views to render the final output.

- **How the pattern work**

- First, the browser sends a request to the Controller. Then, the Controller interacts with the Model to send and receive data.
- The Controller then interacts with the View to render the data.
- Finally, the View will send its final presentation to the Controller and the Controller will send that final data to the user output.

The Model-View-Controller pattern



- **Advantages - Use-case:** because it separate application into module:

- It is easy to maintain the system
- We can expand the application easily
- Multiple developers can work simultaneously on the model, controller and views.
- Improve bandwidth because remove unnecessary information about UI from View component

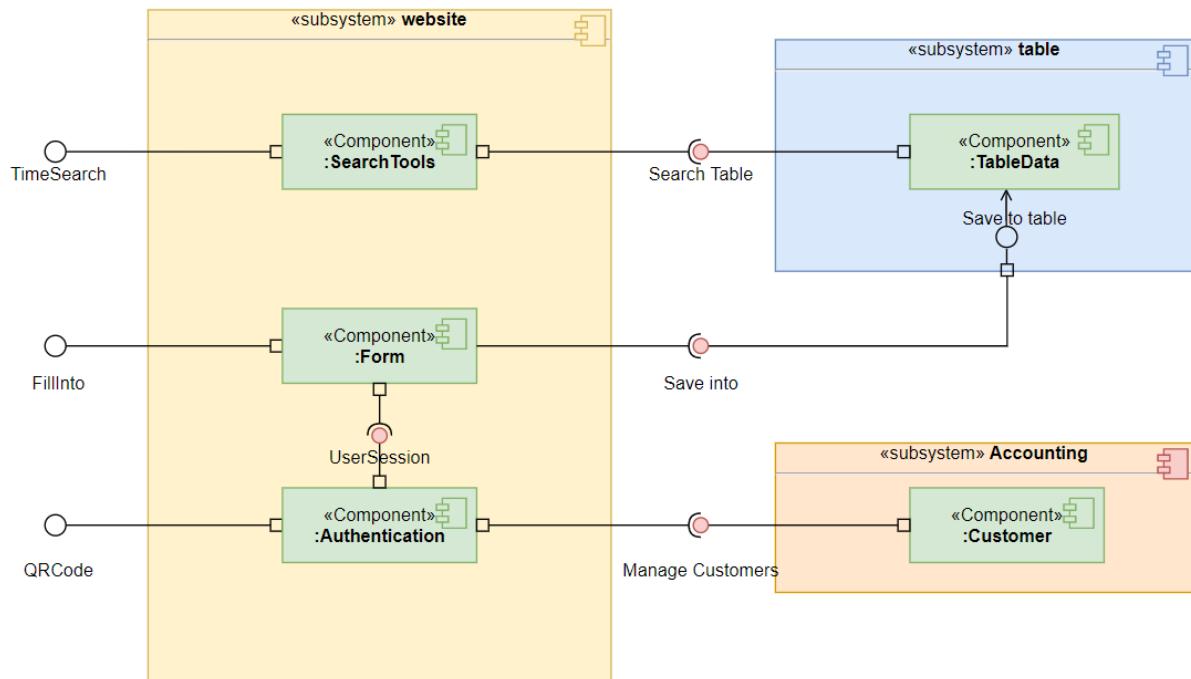
- **Disadvantages:**

- In small project, MVC make the project more complicated, more time-consumed and data transfer will be slower because it doesn't need to distinguish controller and model
- Knowledge on multiple technologies becomes the norm. Developers using MVC need to be skilled in multiple technologies.

9. IMPLEMENTATION DIAGRAM

9.1. Component diagram

9.1.1 Component diagram of table reservation



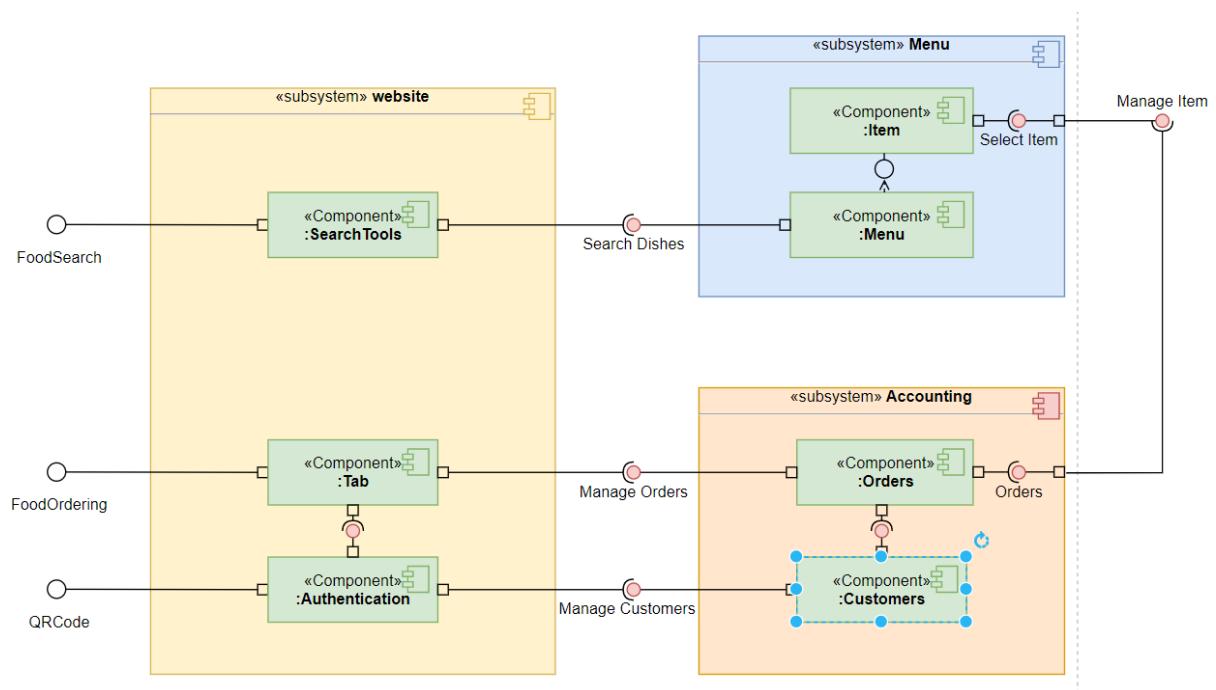
Website subsystem have three components related to Table Reservation

- SearchTools allow that enter day, time into the bar to find a free table in that time by requesting data from table component.
- Form use form interface to request information to apply and save into table database.
- Authentication allows customers to create an account, login, or logout and access by other platform.

Accounting subsystem help customers manage their account, registered orders.

Table subsystem help in searching, changing status and saving customer information to database.

9.1.2 Component diagram of food ordering

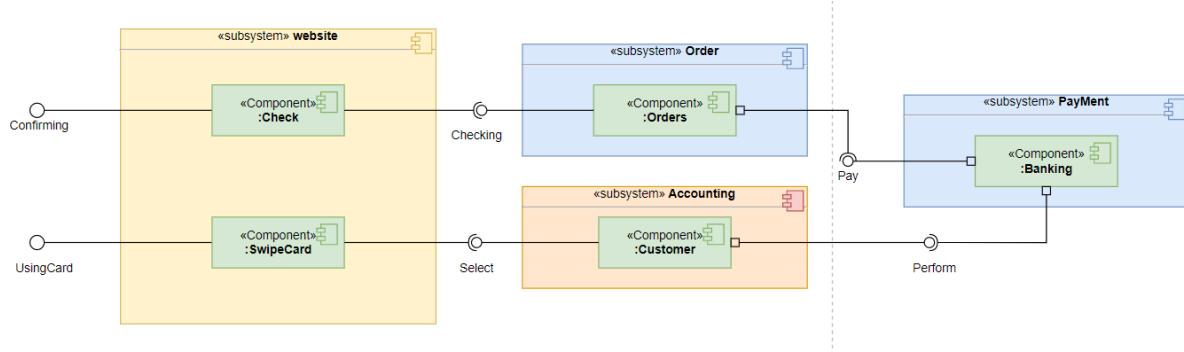


Website subsystem have three components related to Food Ordering:

- `SearchTools` allow that enter dishes name, check hot dishes by request data from menu component.
- `Tab` help customer save temporary dishes that they selected.

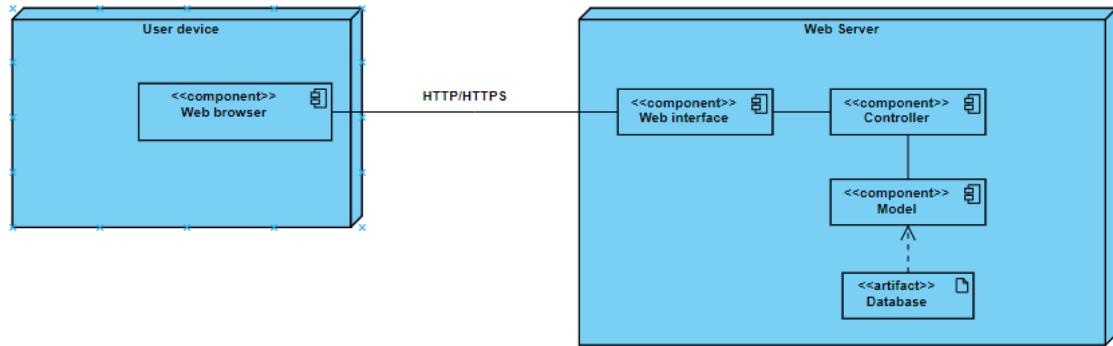
Menu subsystem where save information of item, menu and allow manager manage them (add, delete, update), allow chef check material in inventory and update food available.

9.1.3 Component diagram of pay bill



Payment service where receive total amount from bill of the restaurant and transfer it to a bank that customer selected, help customer payment by credit card.

9.2. Deployment diagram



In this project, we build all components of the application in one server because of size of project and because it focuses on usability. Other aspects like security are skipped due to the need in reality.

The app build with MVC pattern with three component:

- Web interface
- Controller
- Model process data (saved on Database file)

Customer can access into the system through the Web browser, the connection using HTTP/HTTPS protocol