

Task 2: Analysis and Specify Software Quality Requirements

Security and Privacy protection

- The user's registration data such as username, password, phone number, home address can only be viewed and modified by the user. This is vital to comply with data privacy laws.
- All sensitive data including customer's personal details, payment information and communication between the customer's device and the backend system should be encrypted
- For reviews and rankings on a product, they should only be created by a user who purchased the product. Also, only the user who created the review should be able to modify or delete the review. Only registered users should be allowed to make reviews and give feedback. A response can only be given by the shop manager of the branch where the purchase was made by the customer.

Performance

- The application must load its initial interface in under 3 seconds on devices.
- For general user interactions the system should aim for a response time of under 2 seconds. These user interactions include searching for restaurants, booking tables, and pre ordering food.
- The application should minimise resource usage (CPU, memory and battery) on the customer's device by optimising background processes
- The system should be able to handle multiple simultaneous requests, for e.g. several users searching or booking at the same time, without a significant impact on performance.

Reliability

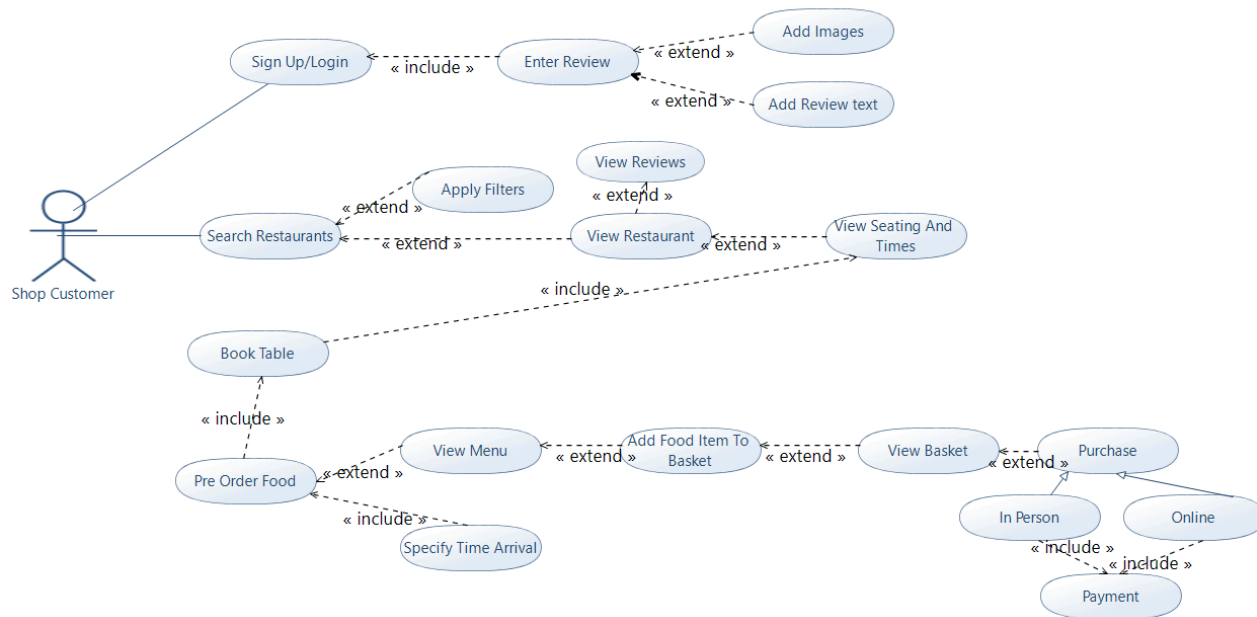
- The system should guarantee an uptime of 99% to ensure that customers can reliably search for restaurants, book tables and provide feedback. This ensures the application is available all the time for users (including peak usage times).
- The system should be fault tolerant and should have the ability to deal with and recover from failures like network issues, databases etc, without affecting the user's experience on the app.
- When a system failure occurs, recovery should on average take no more than 10 hours.
- Backup mechanisms should be implemented to ensure the continuation of services in the case of server failures or crashes.

Scalability

- The system should use a distributed database to handle large volumes of restaurant data and customer data efficiently.
- The system should be able to provide services to approximately 20 million users as customers of shoppers in UK nationally.
- During peak seasons such as Christmas, Easter and other holidays, the system should be able to process 3 million orders per hour.

Task 3:

Use Case Diagram



Activity Model

Name of Use Case: Pre Order Food

Actors: Customer- A customer is a client of the online mobile app who will utilise the system to search/view restaurants, view seating/time availability, book tables, pre order food and make payments.

Entry Condition: The customer has selected a table to book. The system has displayed restaurants to search and seating arrangements so that the client can book the table.

Exit Condition:

- The client has pre ordered their food from the menu
- Payment is either made online or in-person.

Flow of Events:

1. The customer selects the menu item "Pre Order Food"
2. The system displays the menu to the user
3. The customer adds food items to their basket/order.
4. The customer clicks "finish order"
5. The basket/order is displayed to the customer.
6. The customer confirms their wish to proceed
7. If the customer has already registered, the customer enters username and password.
8. If the customer is new, the system displays a screen where the customer enters the following data if they are not registered: name, age, address, phone number and email.
9. The system gives the user the option to either pay online or in person
10. The system requests payment details if online payment is selected.
11. The customer enters payment information.
12. The system checks if the payment is valid.
13. If the payment is valid, reservation confirmation is shown
14. The customer exits the system.

Exceptions and alternative flow of events

2.a Not satisfied with any food options:

1.Customer exits the system

6.a.Customer Leaves Basket Empty:

1. The system displays a message to inform the customer that the basket is empty and they need to add food to their order.

System goes back to step(2)

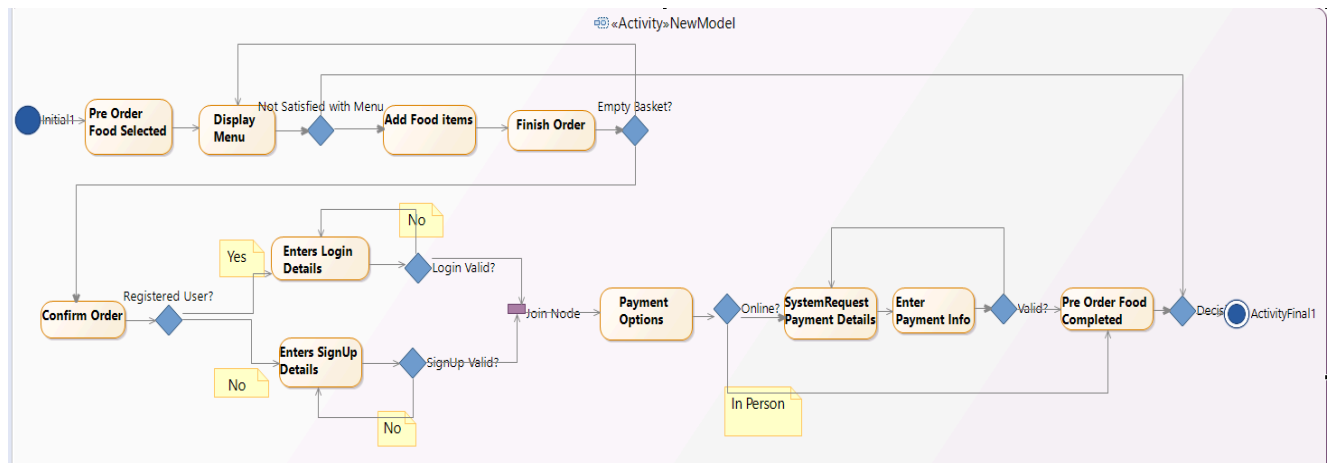
7.a. And 8.a. Invalid Customer Details:

1. The system displays a message to inform the customer that the signup/login isn't valid.

System goes back to step(7 or 8)

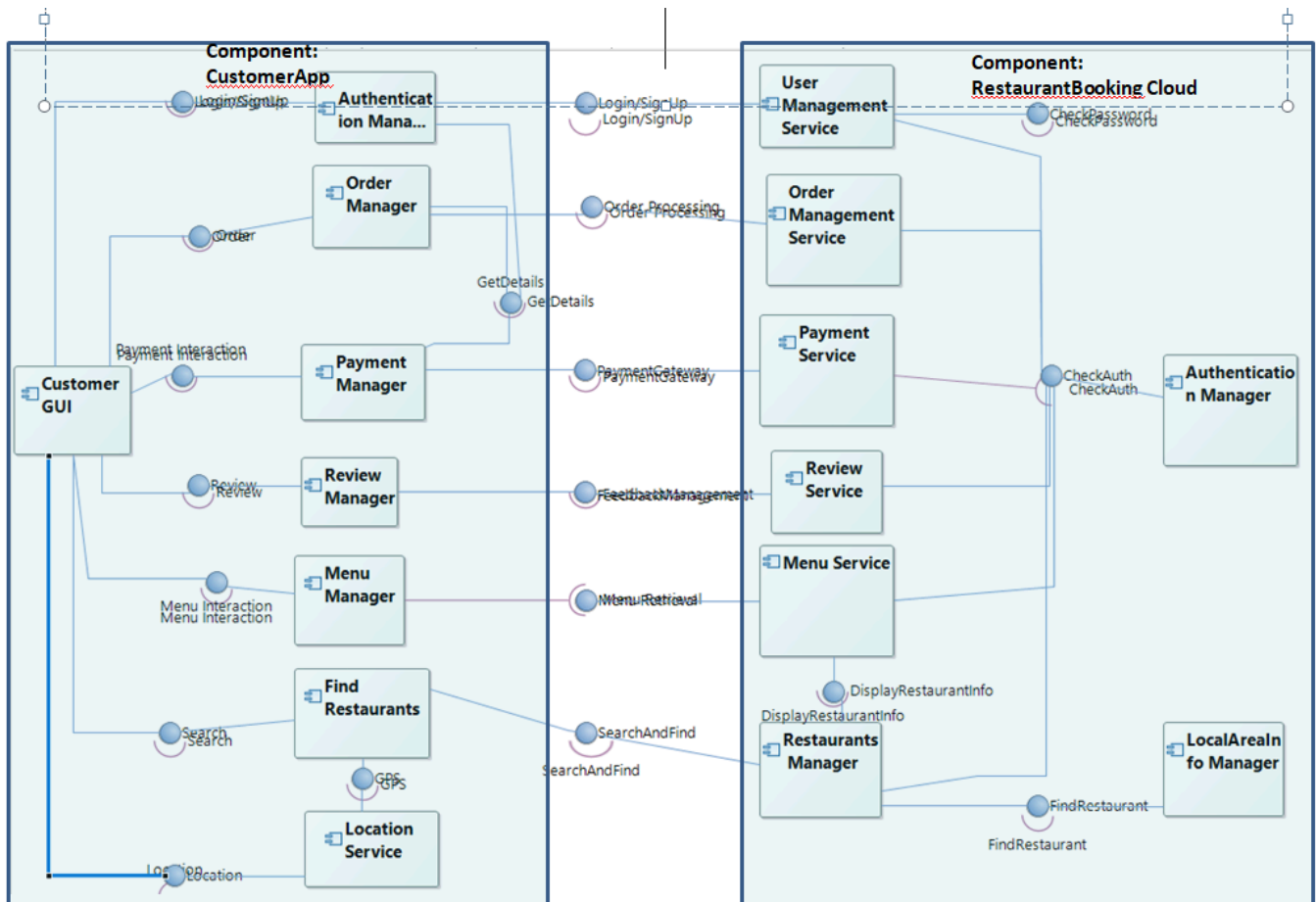
12.a.Payment Declined:

System goes back to step(10)

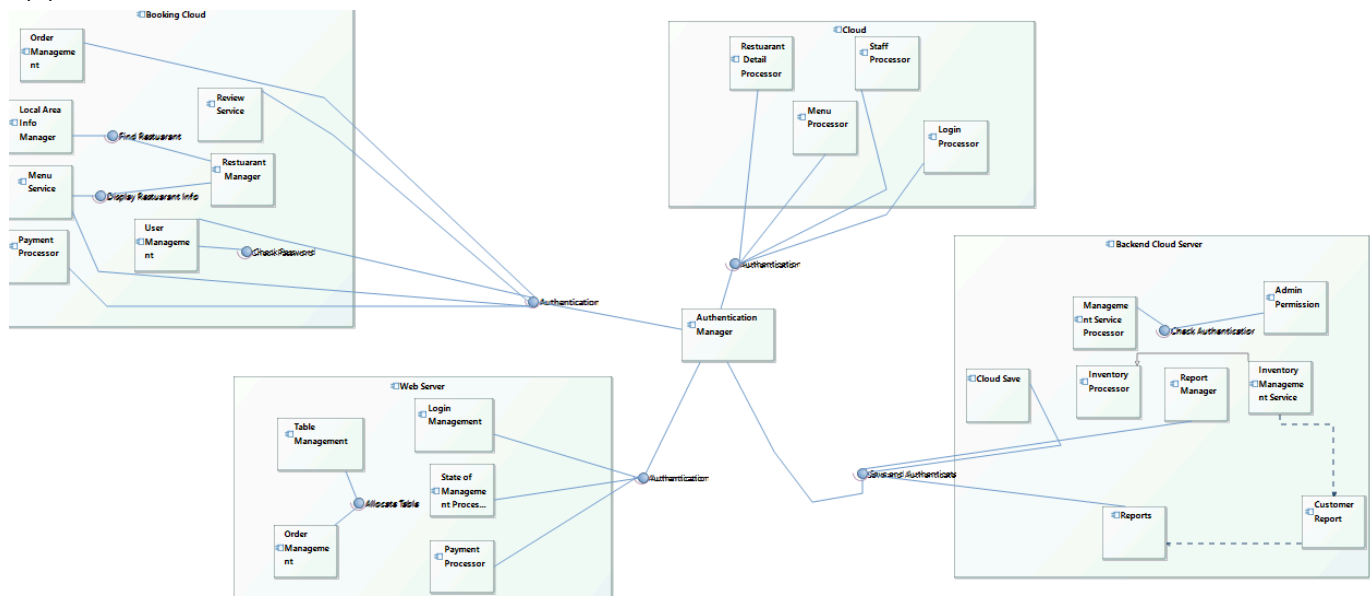


Task 4: Software Architectural Design

Architecture of the subsystem

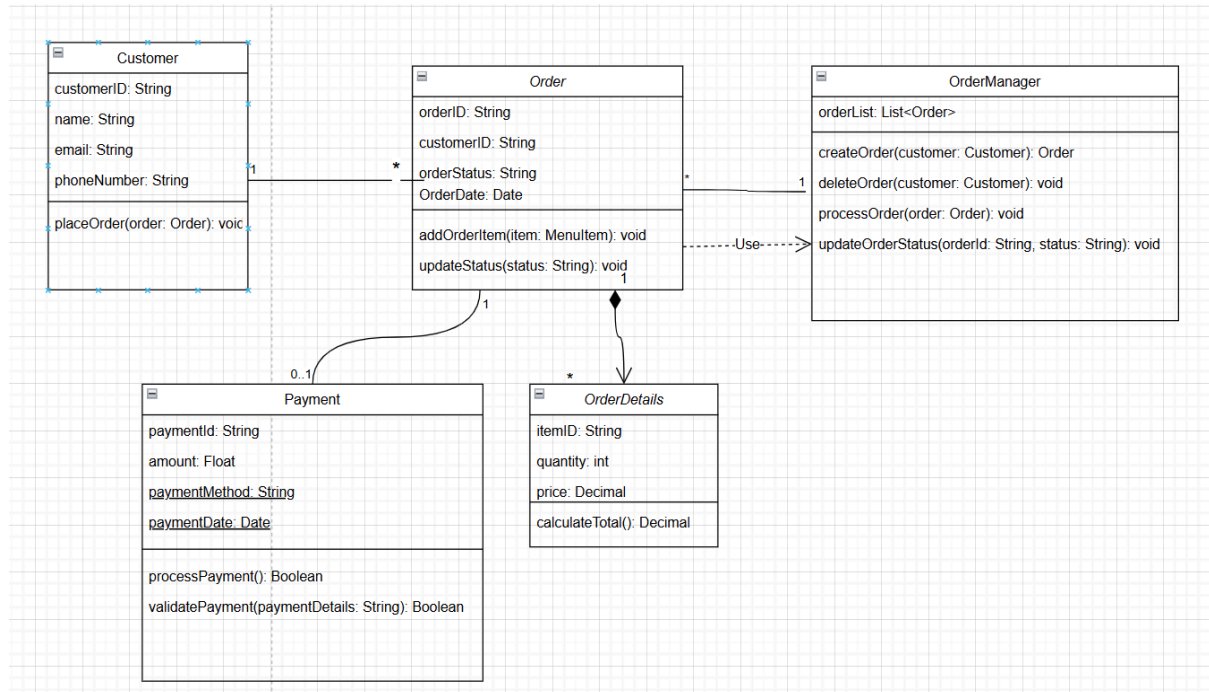


4(b):

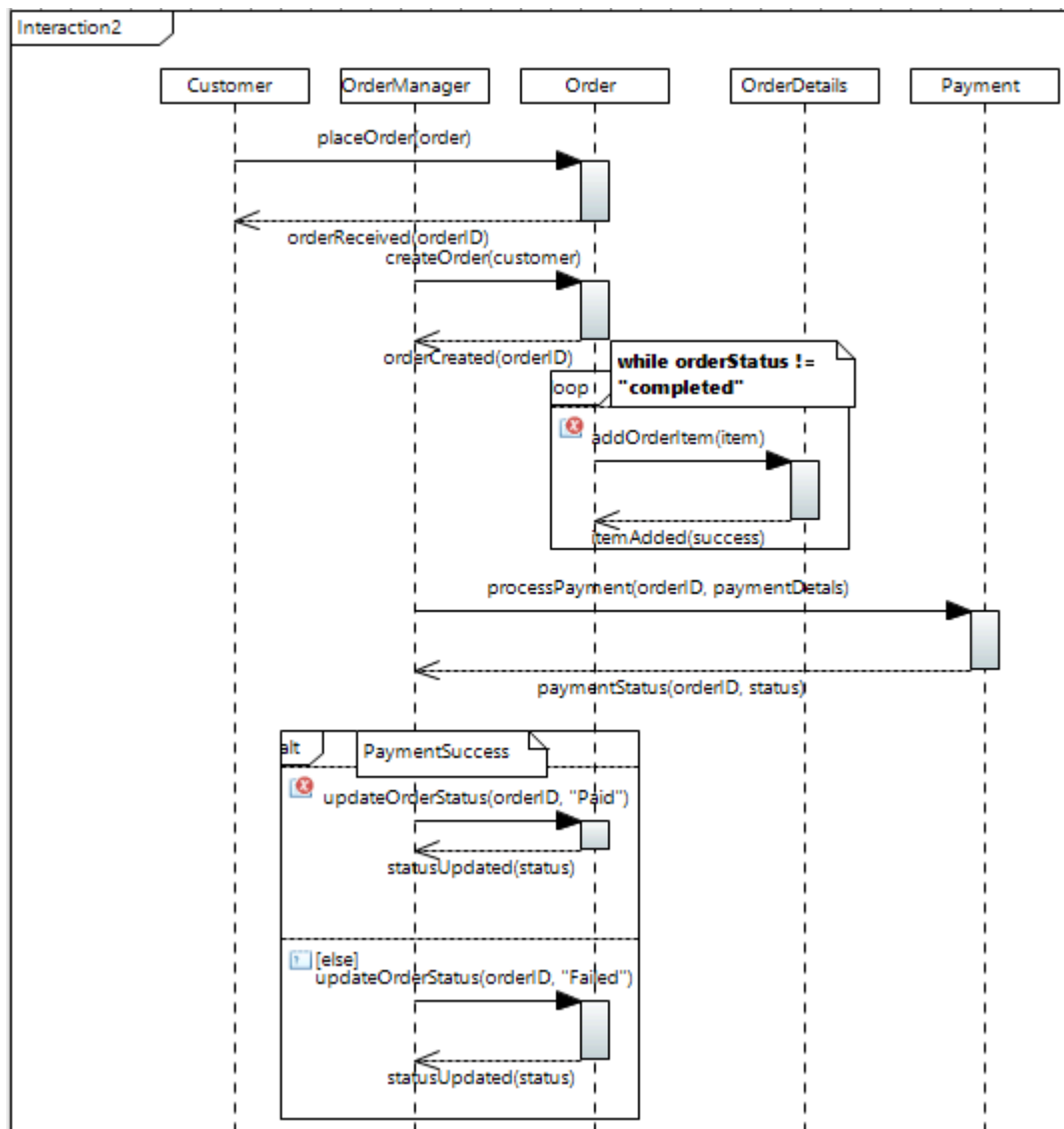


Task 5: Software Detailed Design

5(a): Structural Model- Class Diagram for Order Manager



5(b): Behaviour Model: Sequence Diagram



Report

Contribution to team organisation and project management

In our group project, I worked effectively and efficiently with all of my colleagues. As the meeting notes documentation manager, I took responsibility for ensuring that our team's project management process was well-organised and structured. My specific contributions included: Meeting Contribution: I prepared agendas before each team meeting, ensuring that all members knew what needed to be discussed. During meetings, I documented minutes, including attendance, progress updates, key decisions made, and action items assigned to team members. I uploaded the meeting agendas and minutes to our GitHub repository, maintaining a transparent record of our collaboration. I also facilitated communication between team members, ensuring that everyone was on the same page regarding task distribution and deadlines. Also, I reviews and provided constructive feedback on my colleagues' work to improve model accuracy.

Technical Contributions

I uploaded all of my work to the group's github repository and hence was able to keep track of my work and I could also view my group members' work. I developed UML diagrams (use case, activity diagrams, sequence diagrams) and class diagrams for software design.