



IT-314 Software Engineering
DOCUMENTATION

Restaurant Management System
Group No.- 28

GROUP MEMBERS

Dev Vyas	202201453
Nisarg Vijaykumar Parmar	202201443
Pandya Stuti Hareshbhai	202201439
Mausam Kalpesh Kamdar	202201372
Rakshit Pandhi	202201426
Parmar Harshil Jaysukhlal	202201371
Patel Ridham Nareshkumar	202201430
Kansara Maulik Kamal	202201442
Vraj Gauravkumar Gandhi	202201425
Rit Rajendra Trambadia	202201424

Under the guidance of
Professor: Dr. Saurabh Tiwari

Index

1. Problem Statement
2. Solution
3. Key Features
4. Functional Requirements
5. Non-functional Requirements
6. Use Case Diagram
7. Use Cases
8. User Stories
9. Sprint Description
10. Class Diagram
11. Sequence Diagram
12. System Design
13. Object Design
14. Proposed Model for Development
15. Technology used
16. References

Problem Statement:

In today's fast-paced world, dining out often comes with the hassle of long waiting times, last-minute unavailability, and inefficient booking systems. Customers face difficulties in securing tables at their favorite restaurants, leading to frustration and lost dining experiences.

Restaurants, on the other hand, struggle to manage their seating arrangements efficiently, often missing opportunities to optimize their customer flow and revenue. There's a growing need for a seamless, technology-driven solution to bridge this gap between diners and restaurants.

Solution:

Fork and Feast is an innovative restaurant reservation system designed to streamline the dining experience for both customers and restaurants. Similar to the dine-in feature offered by platforms like Zomato, it provides an intuitive interface for users to explore restaurants, check real-time availability, and reserve tables instantly. Restaurants benefit from improved management tools, enhanced visibility, and a steady flow of patrons. With Fork and Feast, dining out becomes a hassle-free, enjoyable experience for all.

PURPOSE

Streamline reservation management: Simplifies the process of booking, modifying, and canceling table reservations for customers and restaurants.

Enhance customer experience: Provides a convenient platform for users to find, filter, and book restaurants based on their preferences.

Improve restaurant efficiency: Helps restaurants manage reservations, track table availability, and reduce overbooking.

We chose this project because it effectively addresses key issues faced by both customers and restaurants. For customers, the system simplifies the reservation process, provides real-time table availability, offers detailed menu information, and ensures confirmation with automated notifications. It also helps diners make informed choices through reliable reviews and rewards them with loyalty points. For restaurants, the system streamlines reservation management, prevents overbooking, and enhances communication through automated notifications. It also allows for easy menu updates. By tackling these challenges, the project improves overall dining experiences and operational efficiency, making it a valuable and impactful solution.

Key Features

Effortless Reservations: Provide a user-friendly platform that lets users select their preferred date, time, and seating effortlessly.

Enhanced Dining Experience: Offer descriptions and photos of dishes, along with filter options for dietary preferences, to enhance the dining experience.

Authentic Reviews: Provide honest reviews and ratings from real diners, allowing users to share and read the reality.

Real-Time Availability: Real-time updates on table availability and optimized seating arrangements to maximize capacity.

Notifications: notifications and reminders via email or SMS, keeping users informed about their reservations.

Coupons: Awards coupons to premium members

Functional Requirements

1. Dashboard Pages

- User Dashboard:
 - Display reservation history, upcoming reservations, and modification/cancellation options.
 - Provide a search and filter interface for browsing restaurants.
- Restaurant Dashboard:
 - Show pending, confirmed, and past reservations.
 - Allow restaurants to update table availability and view user reservation details.

2. User Registration

- Secure user registration with email verification.
- Login functionality using email and password or social media integration.
- Password recovery/reset mechanism.

3. Search and Filter Options

- Keyword-based search for restaurants (e.g., name, location).
- Filters for:
 - Cuisine type.
 - Location proximity.
 - Price ranges
- Dynamic availability check based on date, time, and party size.

4. Reservations

- Allow users to:
 - Select restaurant, date, time

- Confirm reservations if tables are available.
 - View and manage reservation details.
 - Modify reservations by changing date, time, or party size.
 - Delete reservations with confirmation.
- Restaurants can:
 - Approve or reject booking requests.
 - Update table availability dynamically.

5. Notifications

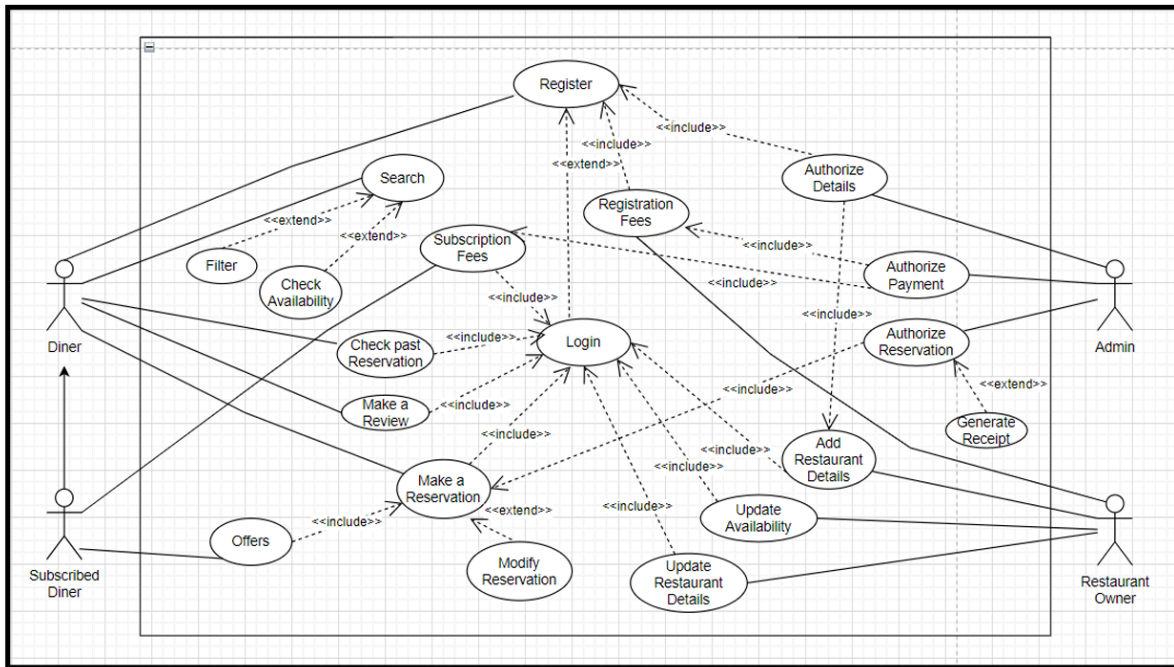
- Send email/SMS notifications for:
 - Booking confirmation or rejection.
 - Updates to reservation details.
 - Reminders for upcoming reservations.

Non-Functional Requirements

- **Performance:**
 - Support multiple concurrent users with no significant performance degradation.
 - Ensure real-time updates of table availability and reservation status.
- **Usability:**
 - Intuitive and user-friendly interface for both users and restaurants.
 - Mobile-responsive design for seamless usage across devices.
- **Security:**
 - Secure user data with encryption and HTTPS protocols.
 - Implement secure authentication and authorization mechanisms.
- **Scalability:**
 - Accommodate increasing numbers of users and restaurants as the platform grows.
- **Reliability:**
 - Ensure high system availability with minimal downtime (e.g., 99.9% uptime).
 - Robust error-handling mechanisms to manage unexpected inputs or system failures.
- **Maintainability:**
 - Modular architecture for easy updates and feature enhancements.
- **Localization:**
 - Provide multi-language support and currency conversion for global users.

Use Case diagram

➤ Use Case Diagram:



Use Cases:

Use Case 1: Login/Signup

Actor: Customer

Precondition:

1. The system is accessible and operational.

2. The customer has an internet connection and a device capable of accessing the application.

Main Flow:

1. The customer accesses the restaurant management application.
2. The customer selects the "Login" or "Signup" option.
3. For login, the customer enters their registered email/username and password.
4. The system validates the credentials. Upon successful validation, the customer gains access to their account.
5. For signup, the customer provides details such as name, email, phone number, and password.
6. The system verifies the provided details.
7. The system creates a new account upon successful verification.
8. The customer is redirected to their dashboard/home page.

Alternate Flow:

- 4.a. If the entered credentials are incorrect, the system displays an error message. And the system prompts the customer to retry or reset their password.
- 6.b. If the required fields are missing or the email is already registered, the system notifies the customer. The customer is requested to correct the errors.

Postconditions:

1. The customer gains access to their account or successfully creates a new one.
2. The system logs the customer's session for secure access.

Use Case 2: Reservation

Actor: Customer

Precondition:

1. The customer is logged into the system.
2. Restaurant details and availability data are up to date.

Main Flow:

1. The customer navigates to the "Reservations" section of the application.
2. The customer selects a restaurant and views its details (e.g., menu, location, ratings).
3. The customer checks table availability for the desired date and time.
4. The customer specifies the number of guests.
5. The customer confirms their reservation.
6. The system processes the reservation request and books the table.
7. The system generates a unique reservation ID.
8. A notification is sent to the customer confirming the reservation details.

Alternate Flow:

- 3.a. If no tables are available for the selected date/time, the system prompts the customer to choose another slot or restaurant.

6.a. If the system encounters an error (e.g., server issue), the reservation is not processed. The customer is notified to retry.

Postconditions:

1. The reservation is confirmed, and the details are stored in the system.
2. The customer receives a confirmation with all relevant details (e.g., date, time, number of guests, entry code).

Use Case 3: View and Modify Reservation

Actor: Customer

Precondition:

1. The customer has an active account.
2. The customer has existing reservations in the system.

Main Flow:

1. The customer navigates to the "My Bookings" section.
2. The system displays a list of active and past reservations.
3. The customer selects an active reservation to modify or cancel.
4. For modifications: The customer updates details (e.g., date, time, number of guests).
The system processes the changes and updates the records.
5. For cancellations: The customer confirms the cancellation. The system processes the cancellation and updates the records.

Alternate Flow:

4.a. Modification Error: If the desired slot is unavailable during modification, the system notifies the customer.

5.a. Cancellation Failure: If the cancellation cannot be processed (e.g., policy restrictions), the system alerts the customer.

Postconditions:

1. Reservation records are updated based on customer actions.
2. Notifications are sent for successful modifications or cancellations.

User Stories

User:

1. As a user, I want to log in or register so that I can access the features of the platform.

o Acceptance Criteria:

- Given that I am on the homepage, when I select "Login/Register", then I am prompted to enter my credentials or create a new account.
- If I enter valid credentials, I am logged in successfully. If I enter invalid credentials, I receive an error message.

2. As a user, I want to search by location, dish, or restaurant so that I can find options based on my preferences.

o Acceptance Criteria:

- When I enter a search query for a dish, location, or

restaurant, I am shown relevant results based on my search.

- If no results are available, a message informs me of the lack of availability.

3. As a user, I want to check the availability of the restaurant so that I can know if there is space for my reservation.

o Acceptance Criteria:

- When I select a restaurant, I am shown the available time slots and dates.
- If no availability is present, I receive a message stating that.

4. As a user, I want to make a reservation at a restaurant on my preferred date and time so that I can plan my visit accordingly.

o Acceptance Criteria:

- When I select an available slot, I can confirm the reservation with details including time, date, and number of guests.
- I receive a confirmation message once the reservation is made.

5. As a user, I want to modify my reservation so that I can adjust it if my plans change.

o Acceptance Criteria:

- Given that I have an existing reservation, I can update the date, time, or guest number, provided availability allows.
- If the reservation is successfully modified, I receive an updated confirmation.

6. As a user, I want to cancel my reservation if needed so that I am not charged or blocked from future bookings.

o Acceptance Criteria:

- I can select an option to cancel my reservation, and once confirmed, it is successfully canceled.
- I receive a message confirming the cancellation.

7. As a user, I want to receive updates on my reservation so that I am informed of any changes.

o Acceptance Criteria:

- If my reservation is modified by the restaurant, I receive a

notification with the updated details.

8. As a user, I want to subscribe to offers and discounts so that I can save on future bookings.

o Acceptance Criteria:

- When I subscribe, I am notified of current and future offers or discounts.
- If I unsubscribe, I stop receiving these updates.

9. As a user, I want to pay subscription fees so that I can unlock premium features or discounts.

o Acceptance Criteria:

- I am able to make a payment via the platform's available methods.
- Once payment is successful, I receive a confirmation and access to subscription benefits.

10. As a user, I want to receive a receipt after making a reservation so that I have proof of the transaction.

o Acceptance Criteria:

- After completing a reservation, I am sent a receipt via email or SMS with the details of my reservation.

11. As a user, I want to view my past reservations, so that I can keep track of where and when I've dined.

• Acceptance Criteria:

- o Users can access a "Reservation History" section.
- o The history displays previous reservations with details (restaurant, date, time, status).
- o Users can sort/filter history by date or restaurant.

12. As a user, I want to leave a review after dining at a restaurant, so that I can share my experience with others.

• Acceptance Criteria:

- o Users can leave a review after a completed reservation.
- o The review includes a rating (e.g., 1 to 5 stars) and optional

comments.

- o Users receive confirmation after successfully posting a review.

13. As a user, I want to view the restaurant's menu before making a reservation, so that I can decide if it meets my preferences.

- Acceptance Criteria:

- o Users can view a detailed menu on the restaurant's page.

- o The menu includes descriptions, prices, and dietary labels (e.g.,

- vegetarian, gluten-free).

- o Users can make a reservation directly from the menu page.

14. As a user, I want to receive an email or SMS confirmation after successfully paying for a reservation or subscription, so that I have a record of the transaction.

- Acceptance Criteria:

- o Users receive an automatic confirmation after payment is processed.

- o The confirmation includes payment details, reservation details, and a receipt.

- o Users can choose whether to receive confirmations via email or

- SMS in their profile settings.

Restaurant Owner:

1. As a restaurant owner, I want to register my restaurant on the platform

so that it is visible to users for reservations.

- o Acceptance Criteria:

- When I provide all required details for my restaurant, my registration is submitted for admin approval.

- Once approved, my restaurant appears on the platform for users to see.

2. As a restaurant owner, I want to update my restaurant details so that I

can ensure accurate information is displayed.

- o Acceptance Criteria:

- I can edit my restaurant's description, menu, photos, and other details as needed, and changes are saved successfully.

3. As a restaurant owner, I want to update my restaurant's availability so

that users know when reservations are possible.

- o Acceptance Criteria:

- I can manage availability by setting available time slots, and these changes are reflected immediately for users.

Admin:

1. As an admin, I want to authorize restaurant registrations so that only verified restaurants are available on the platform.

- o Acceptance Criteria:

- I can view all pending restaurant registrations and either approve or reject them with comments.

2. As an admin, I want to authorize reservations so that they comply with platform standards.

- o Acceptance Criteria:

- I can view and approve or reject reservations, with notifications sent to users and restaurant owners accordingly.

3. As an admin, I want to authorize payments so that all transactions are secure and legitimate.

- o Acceptance Criteria:

- I can monitor and approve payments made on the platform, ensuring successful transactions are recorded.

Sprint description

Sprint 1: Registration and Login

- Task: • Implement restaurant owner and customer registration (including basic validation).
- Implement login and reset password functionality for restaurant owners and customers.

Sprint 2: Restaurant Management

- Task: • Implement functionality for restaurant owners to add restaurant details (name, address, cuisine type, seating capacity, etc.).
- Implement functionality for restaurant owners to update restaurant details.
- Implement functionality for restaurant owners to delete restaurant details.
- Implement functionality for restaurant owners to view the list of their restaurants.

Sprint 3: Search and Filter Functionality

- Task: • Implement search functionality for customers to find restaurants based on various filters (location, cuisine, rating, etc.).
 - Integrate the search results with the reservation system.
 - Develop UI for search and filter options
- Function Type Count
Weighting

Sprint4: Reservation System

➤ Task:

- Implement seat availability checking functionality.
- Develop table booking functionality with date and time selection.
- Implement booking confirmation and reminders
- Allow customers to view and update their reservations.
- Implement reservation cancellation functionality
- Allow customers to write reviews.
- Implement nominal charge in order to reserve the table

Sprint 5: Customer Management & Premium Membership

➤ Task: 1. Develop Customer Profile Management:

- Features to update customer details.
- Ability to view reservation history.

2. Implement Premium Membership System:

- Fee payment for premium membership.
- Discount management for premium users.

3. Apply Discounts for Premium Users:

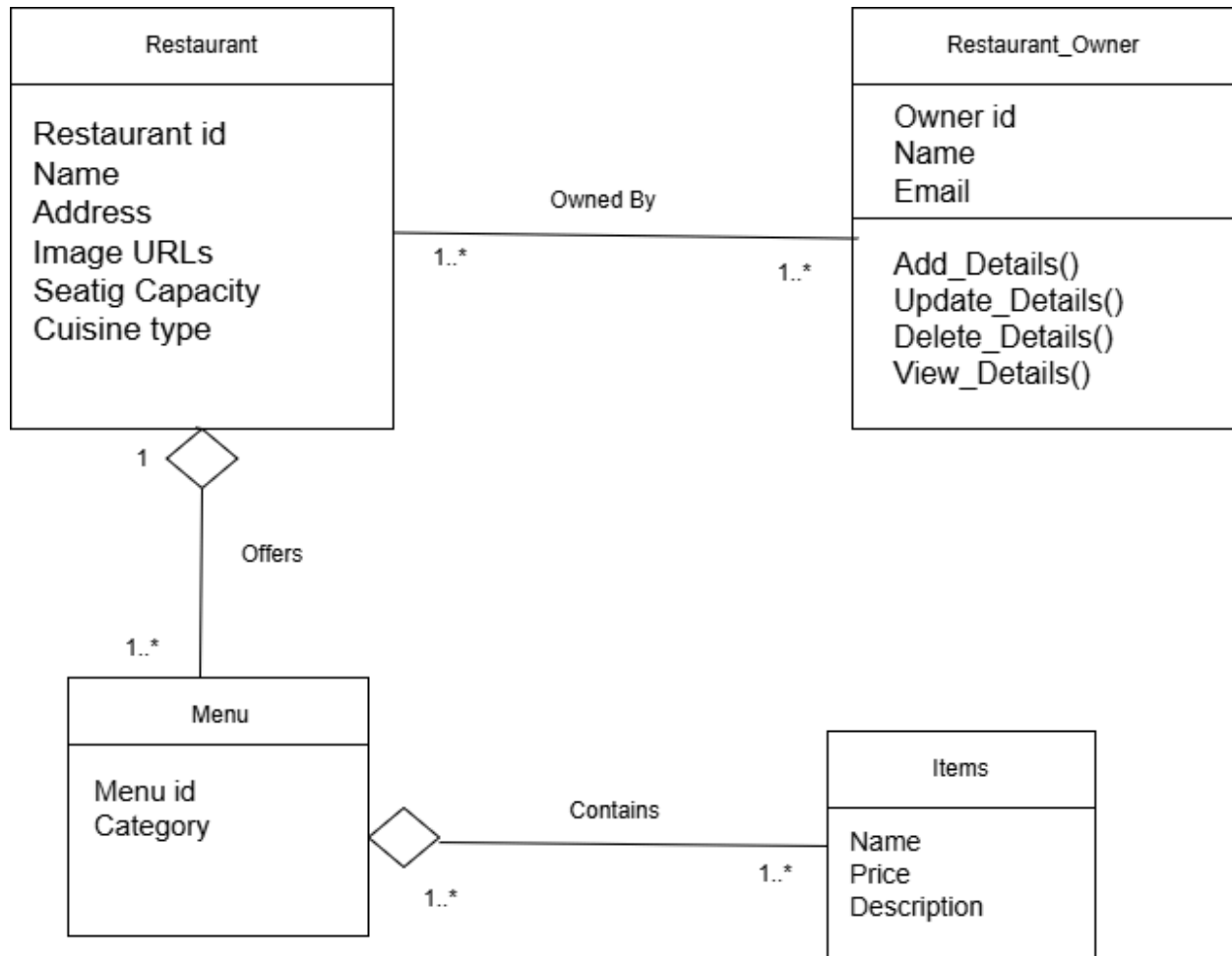
- Ensure that discounts are applied during the reservation process for premium members.

4. Develop UI for Premium Membership and Discount Management:

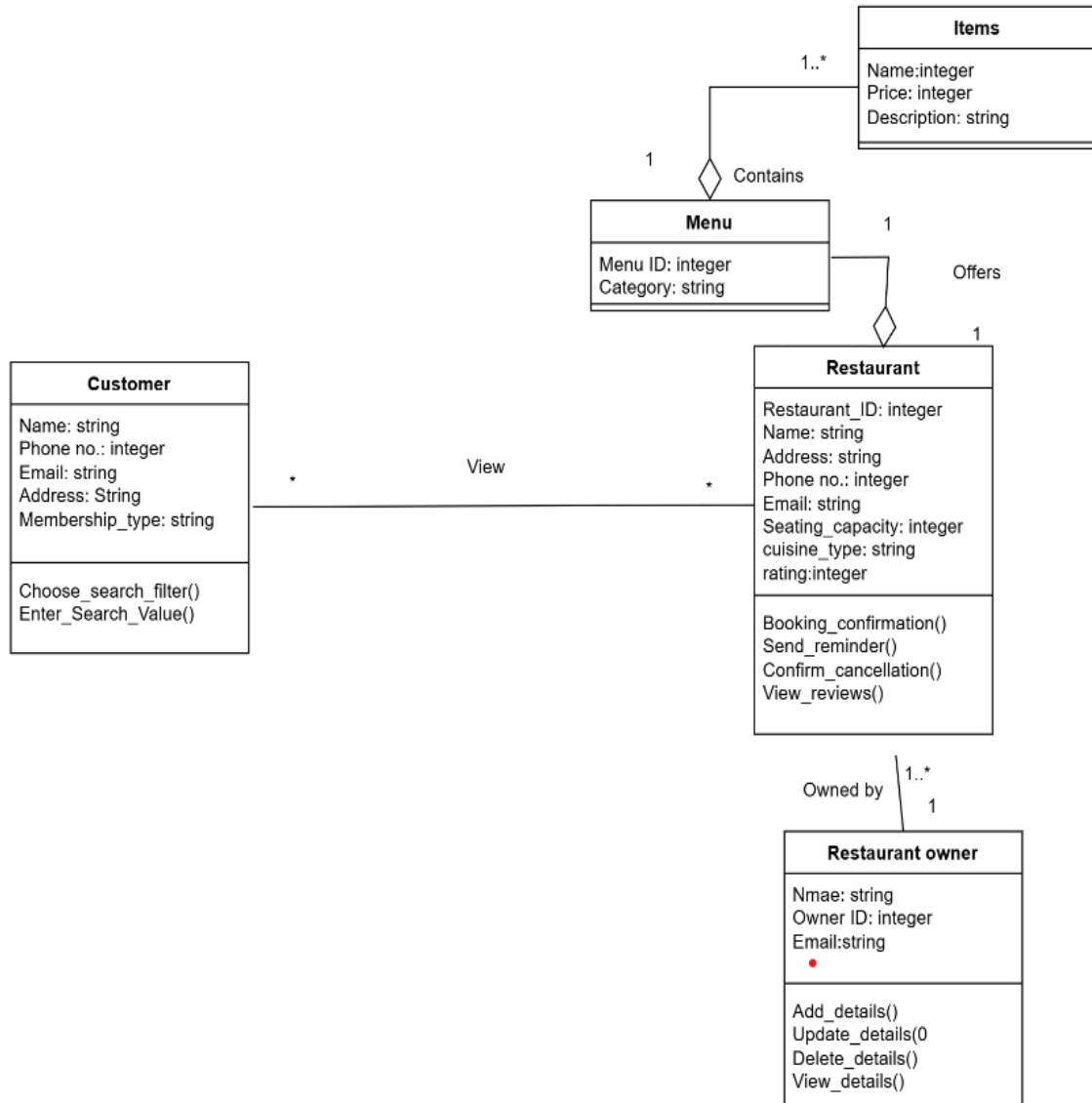
- Design and implement the user interface for managing premium membership and applying discounts.

Class Diagrams:

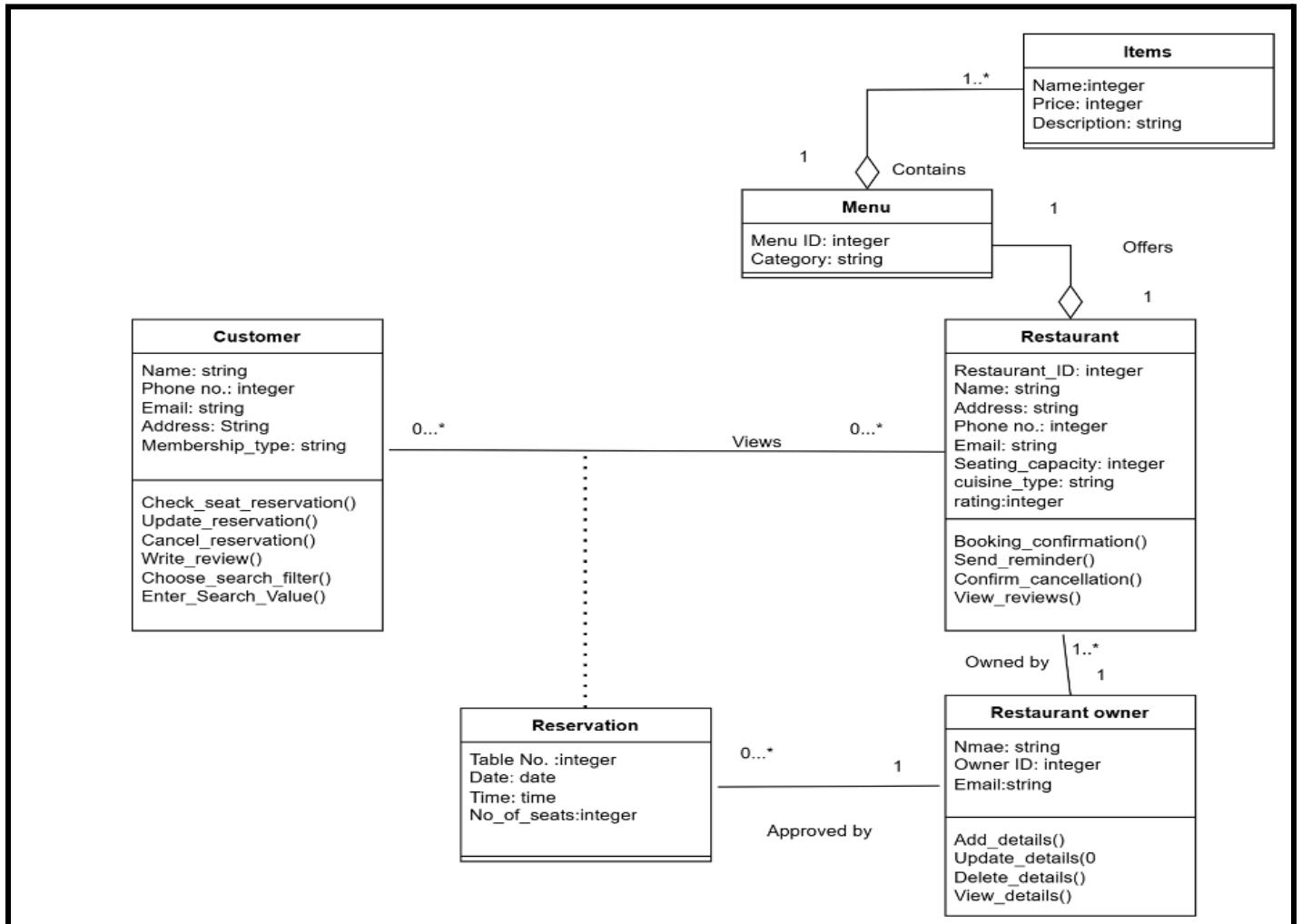
Sprint 2:



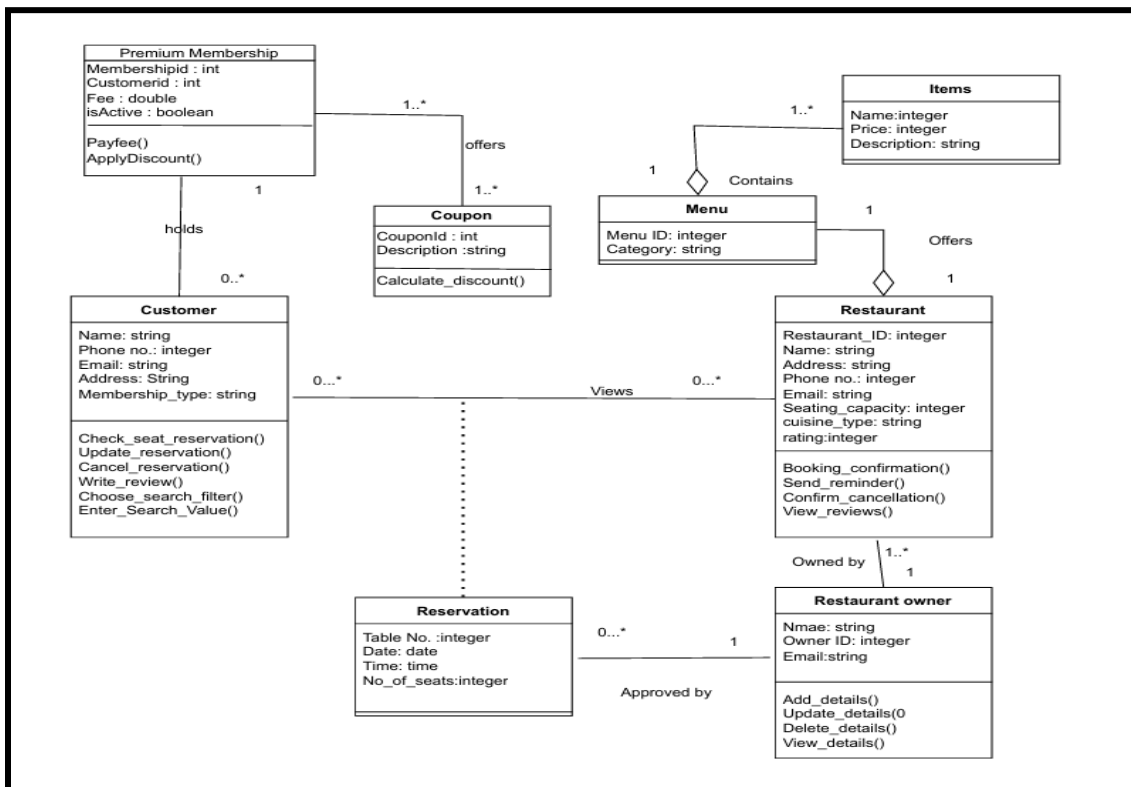
Sprint 3:



Sprint 4:

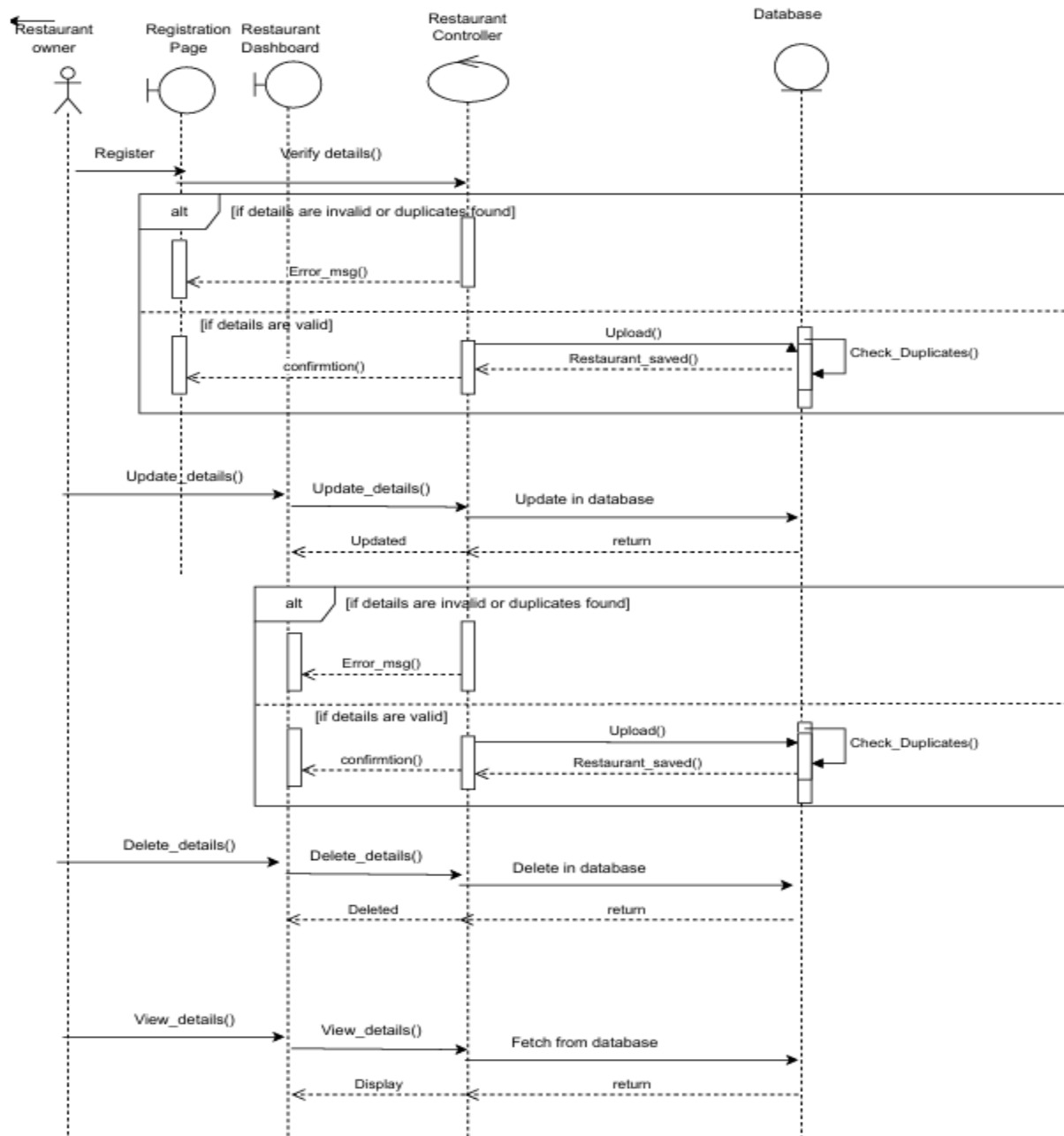


Sprint 5:

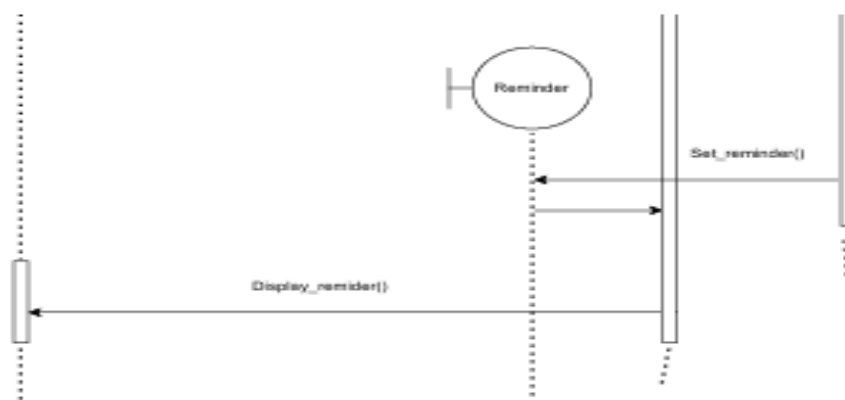
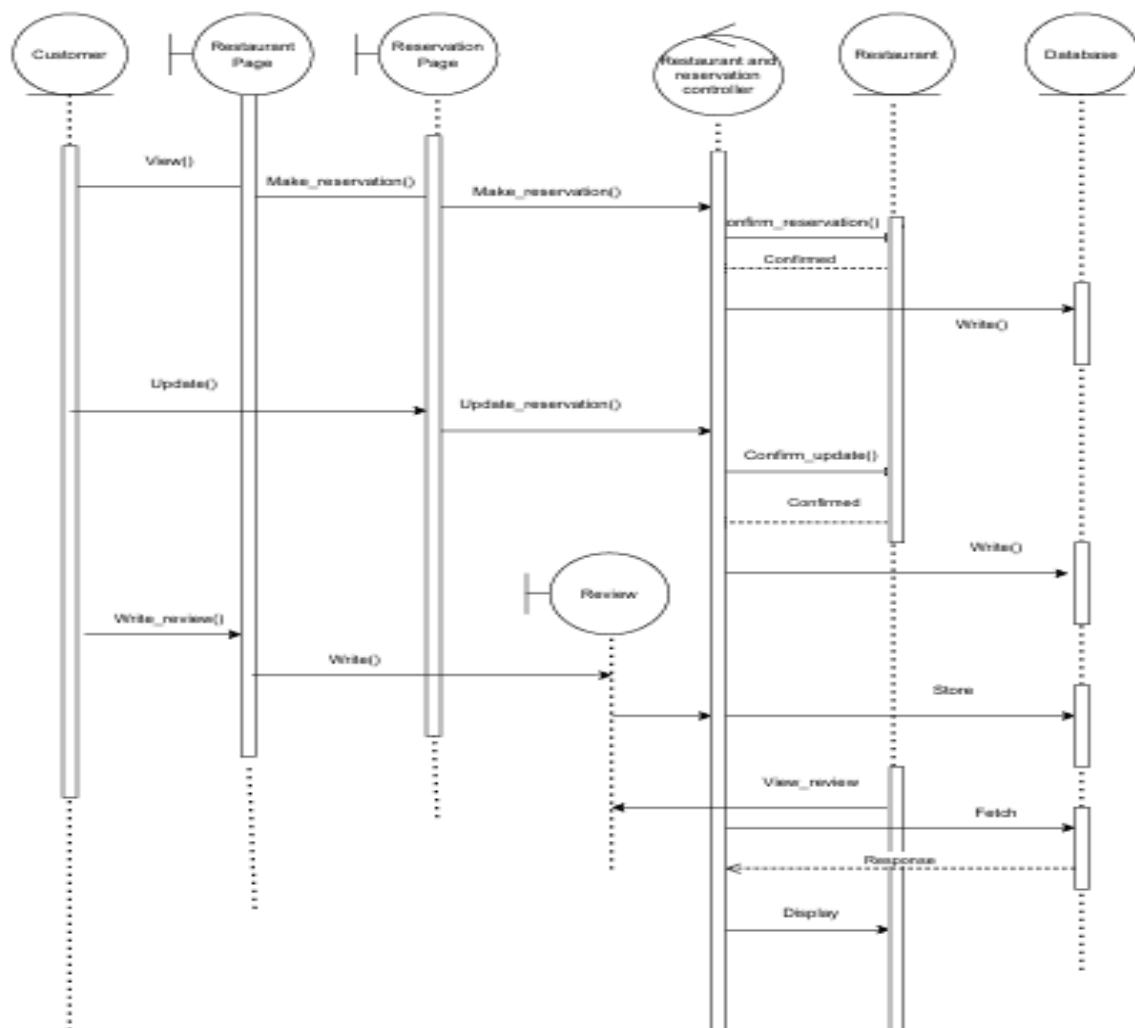


Sequence diagrams:

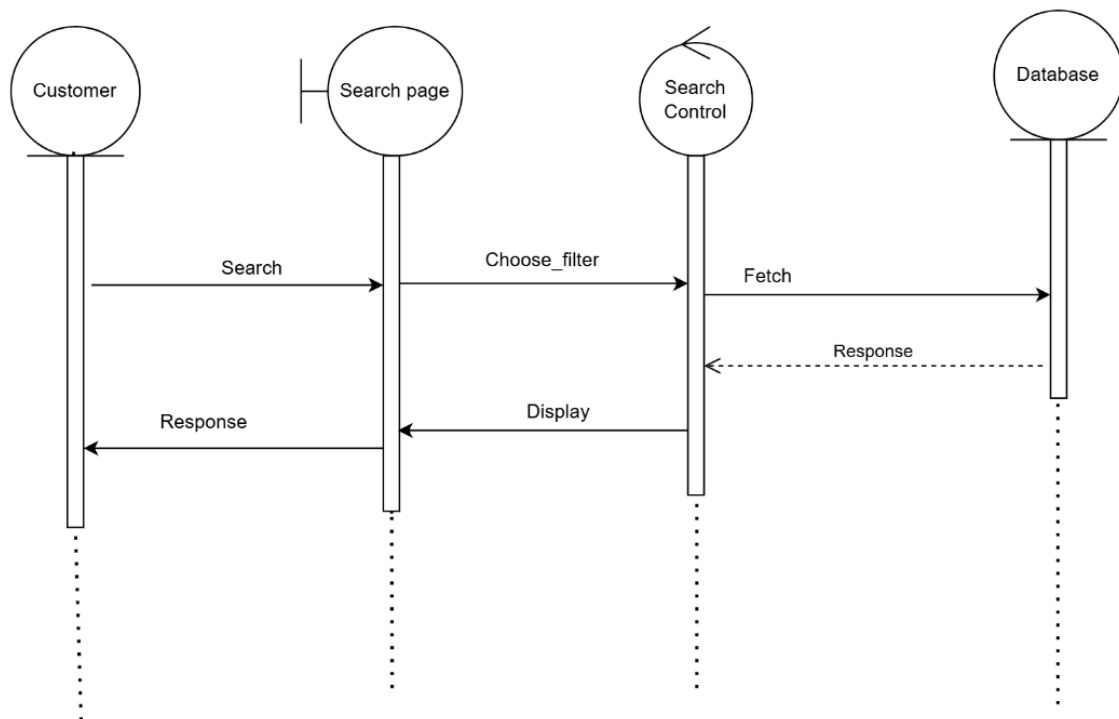
Sprint 2:



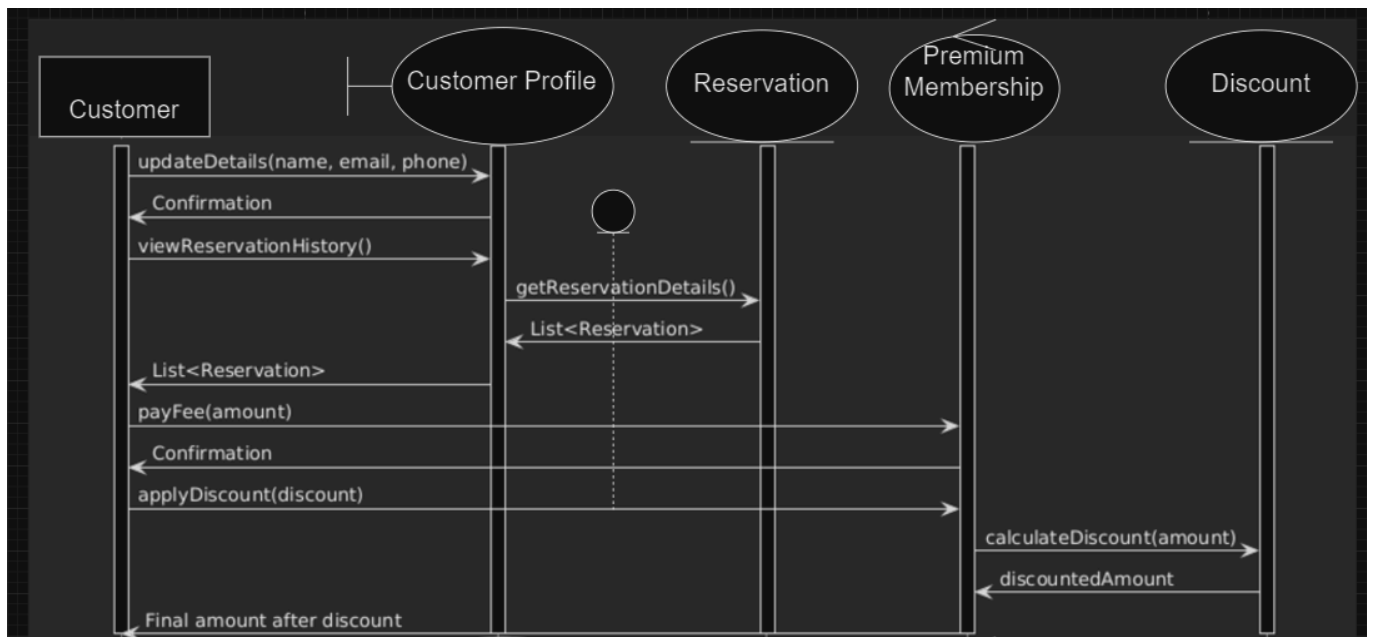
Sprint 3:



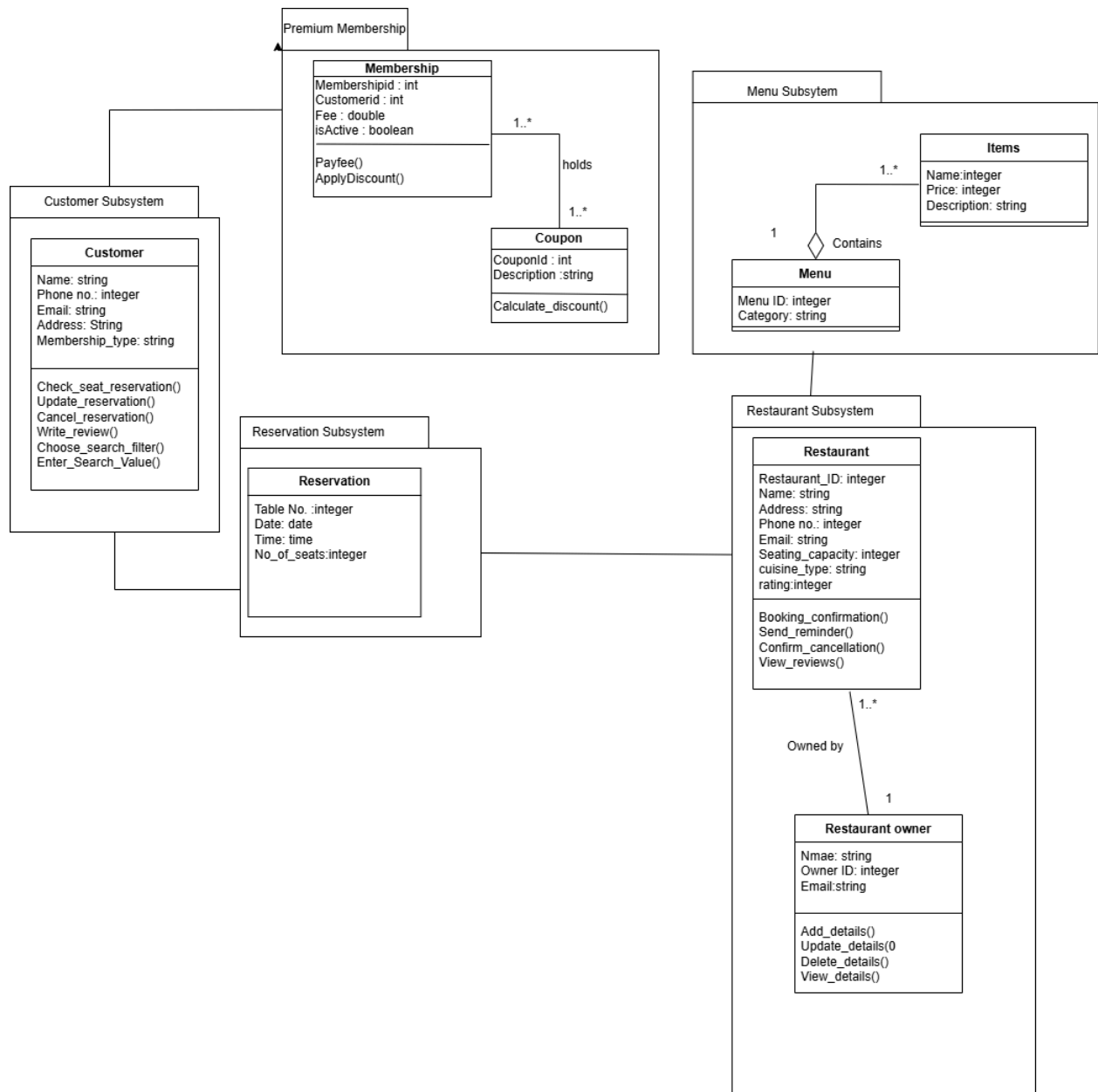
Sprint 4:



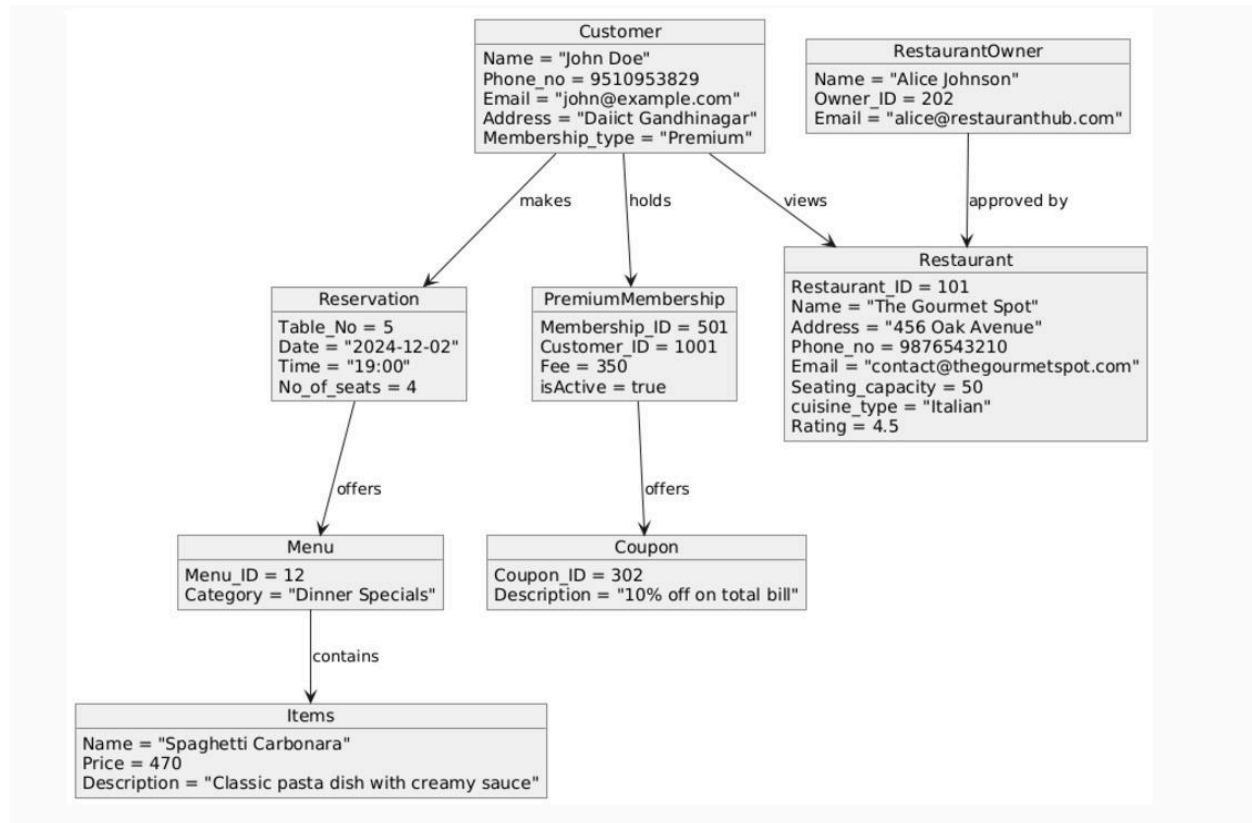
Sprint 5:



System Design:



Object Design:



Proposed Model for Development

1. Requirements Gathering

- **Stakeholder Collaboration:** Work closely with stakeholders, including restaurant owners, managers, and potential customers, to gather system requirements.
- **User Research:** Conduct interviews, surveys, and user testing to identify key features like dashboards, reservation management, search and filter options, and notifications.
- **Prototyping:** Create prototypes for the user and restaurant dashboards to validate initial ideas and gather feedback for improvement.

2. Sprint Planning

- **Task Breakdown:** Decompose requirements into manageable tasks such as dashboard design, search and filter functionality, reservation management, and notification system.
- **Prioritization:** Prioritize features based on stakeholder needs and system dependencies, focusing on high-priority components like user registration and reservation functionality first.
- **Sprint Organization:** Plan sprints lasting 2–4 weeks, with clear goals for each sprint, such as completing user registration in Sprint 1 and developing search and filter features in Sprint 2.

3. Development and Testing

- **Incremental Development:** Develop system features incrementally, starting with core functionalities like user registration and dashboards before adding advanced features like notifications and modification capabilities.
- **Continuous Testing:** Conduct unit testing for individual components and integration testing to ensure seamless interaction between system modules.

- **Code Reviews:** Regularly review code for quality assurance and to identify potential issues early.

4. Sprint Review

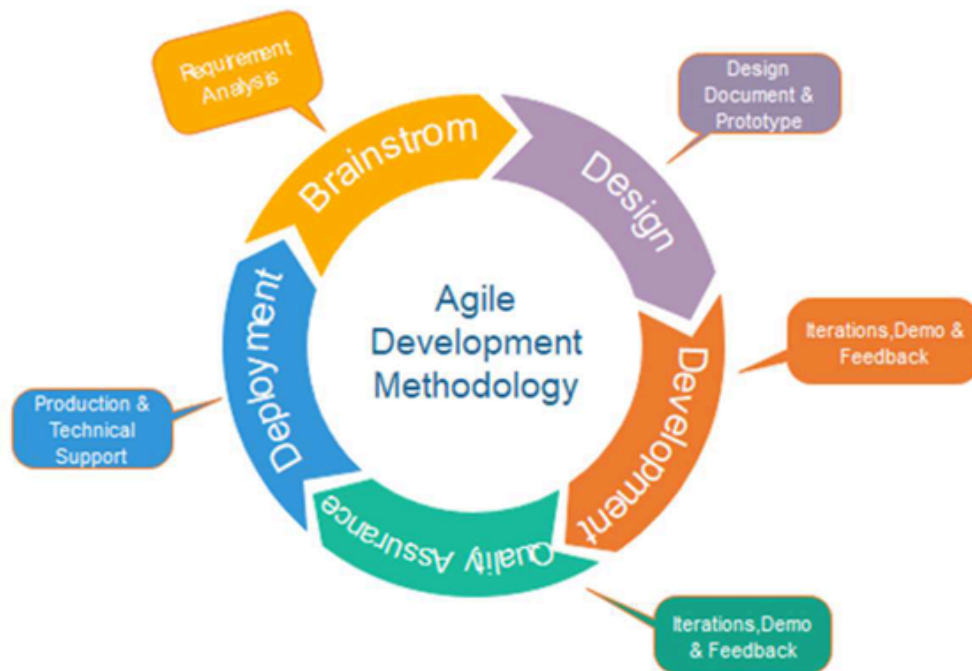
- **Stakeholder Feedback:** Present sprint deliverables to stakeholders (e.g., restaurant managers or end-users) and demonstrate system progress.
- **Iterative Feedback:** Collect feedback on usability, functionality, and design, and prioritize changes or enhancements for the next sprint.

5. Sprint Retrospective

- **Reflection:** Analyze the sprint to identify strengths and weaknesses in processes, tools, or team dynamics.
- **Improvements:** Implement changes to address challenges, such as adjusting workflows, refining communication, or adopting better tools.

6. Repeat

- **Iterative Development:** Continue the cycle of sprint planning, development, review, and retrospective. Use stakeholder feedback to refine the system incrementally.
- **Completion:** Repeat until all required features (e.g., dashboards, search, reservations, notifications, and cancellation/modification functionality) are fully implemented and meet stakeholder expectations



Technology Used

1. FrontEnd- HTML, CSS , ReactJS
2. Backend- NodeJS, MongoDB, ExpressJS
3. Testing- Mocha,Chai, JMeter, Selenium

References

1. StackOverflow - <https://stackoverflow.com/>
2. GeeksForGeeks - <https://www.geeksforgeeks.org/>
3. ChatGPT - <https://chat.openai.com/>