**NLP Chatbot Development using Dialogflow**

**Design Document**

**Version 1.0**

****

**Group Id:** F24PROJECTB3B8C

**Supervisor Name :** *Abdullah Qamar*

**Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date (dd/mm/yyyy)** | **Version** | **Description** | **Author** |
| 20/12/2024 | 1.0 | This document outlines the design and architecture of the restaurant-based NLP chatbot, developed using Dialogflow for natural language processing and MySQL for data management. The chatbot is designed to assist customers with tasks such as making reservations, navigating the menu, placing orders, and providing customer support. The design includes the chatbot’s functional components, database schema, user interactions, and integration with the web interface built using HTML, CSS, and JavaScript. Additionally, it defines the system architecture, user interface design, and API integrations required to ensure seamless communication between the chatbot and the backend services. | **BC210414048** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**Table of Contents**

1. [Introduction of Design Document](#One)
2. [Entity Relationship Diagram (ERD)](#ERD)
3. [Sequence Diagrams](#Six)
4. [Architecture Design Diagram](#Seven)
5. [Class Diagram](#class)
6. [Database Design](#databasedesign)
7. [Interface Design](#interfacedesign)
8. [Test Cases](#testcases)

### 1.Introduction of Design Document

The **Design Document** serves as a comprehensive blueprint for the development phase, providing critical insights and specifications to ensure project alignment with defined requirements. This document includes

1. **Entity Relationship Diagram (ERD**)

The Entity Relationship Diagram (ERD), which outlines the system's data structure and relationships between entities, ensuring clarity in database design.

1. **Sequence Diagrams**

TheSequence Diagrams illustrate task sequences for each use case, capturing the system's interaction flow and facilitating better understanding of system behavior.

1. **Architecture Design** **Diagram**

The Architecture Design Diagram represents the system's tiered structure, defining components and their interactions for seamless integration.

1. **Class Diagram**

The Class Diagram provides a detailed representation of classes, attributes, and methods, forming the backbone of object-oriented design.

1. **Database Design**

The Database Design includes a logical schema, showing table structures and relationships for efficient data storage and retrieval.

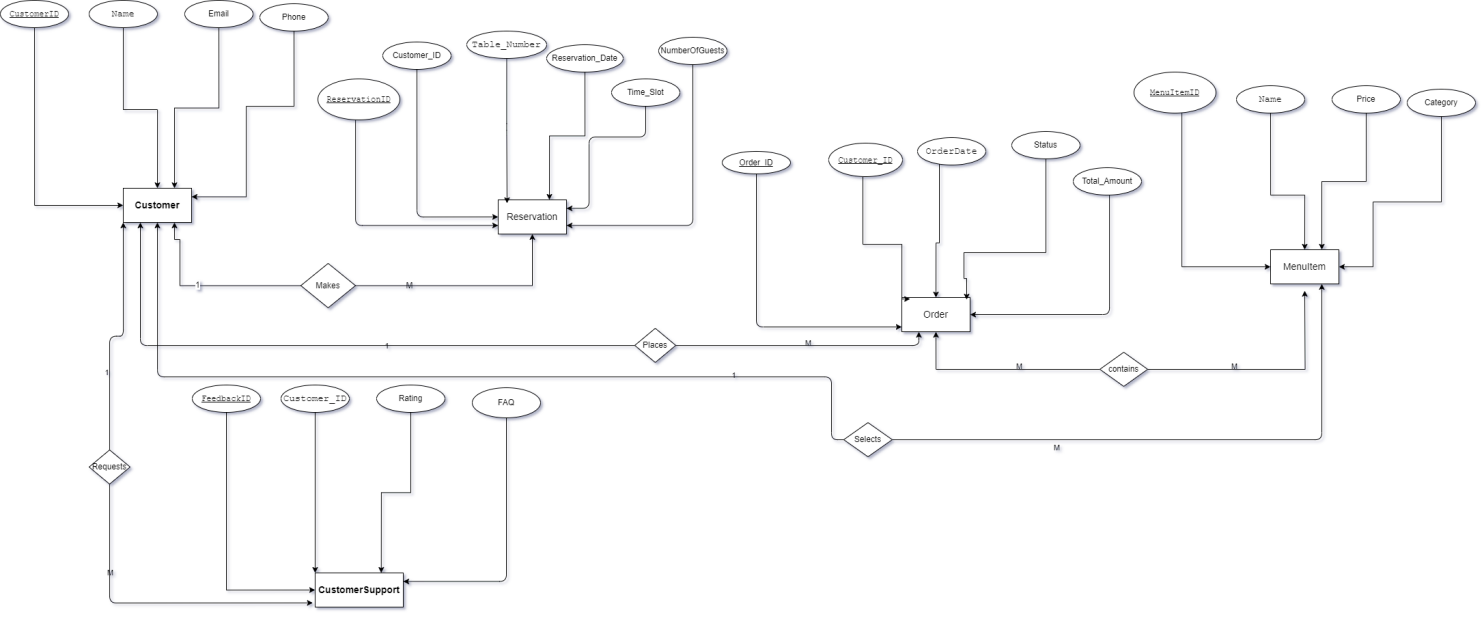
1. **Interface Design**

The Interface Design showcases the envisioned user interfaces, offering a glimpse into the application's functionality and user experience.

1. **Test Cases**

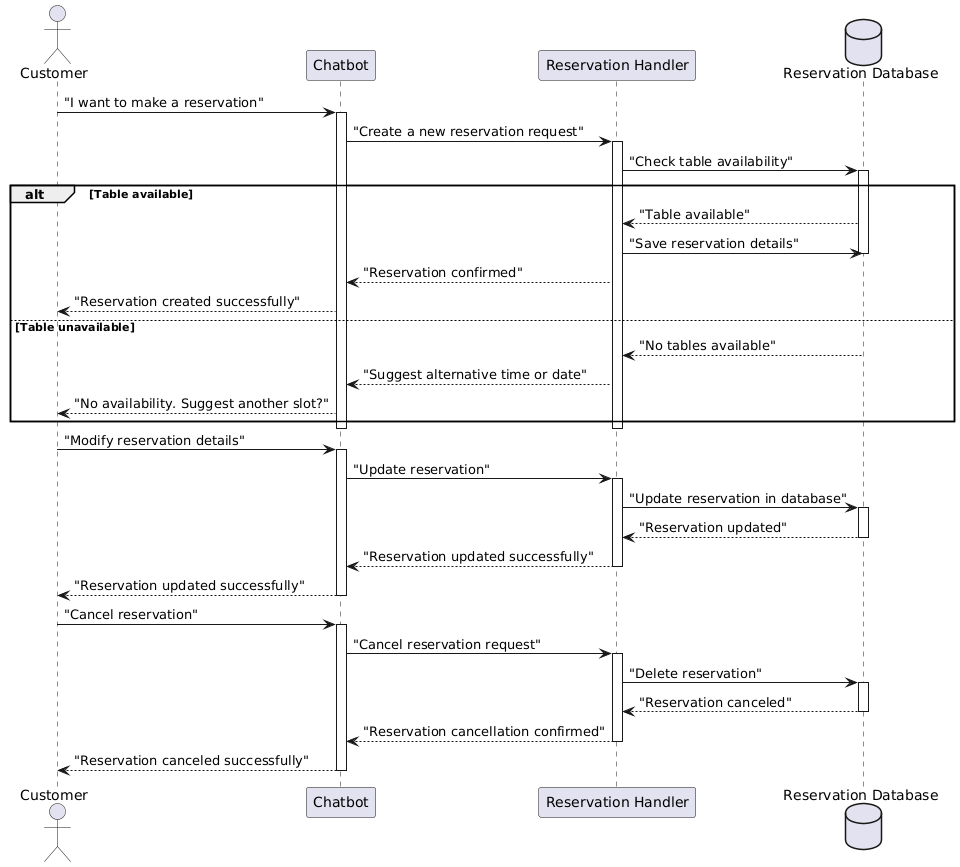
Test Cases detail scenarios for validating each use case, ensuring functionality and reliability.

Together, these elements make the Design Document a critical phase, bridging conceptual requirements from the SRS to tangible implementation, minimizing risks, and enhancing development efficiency.

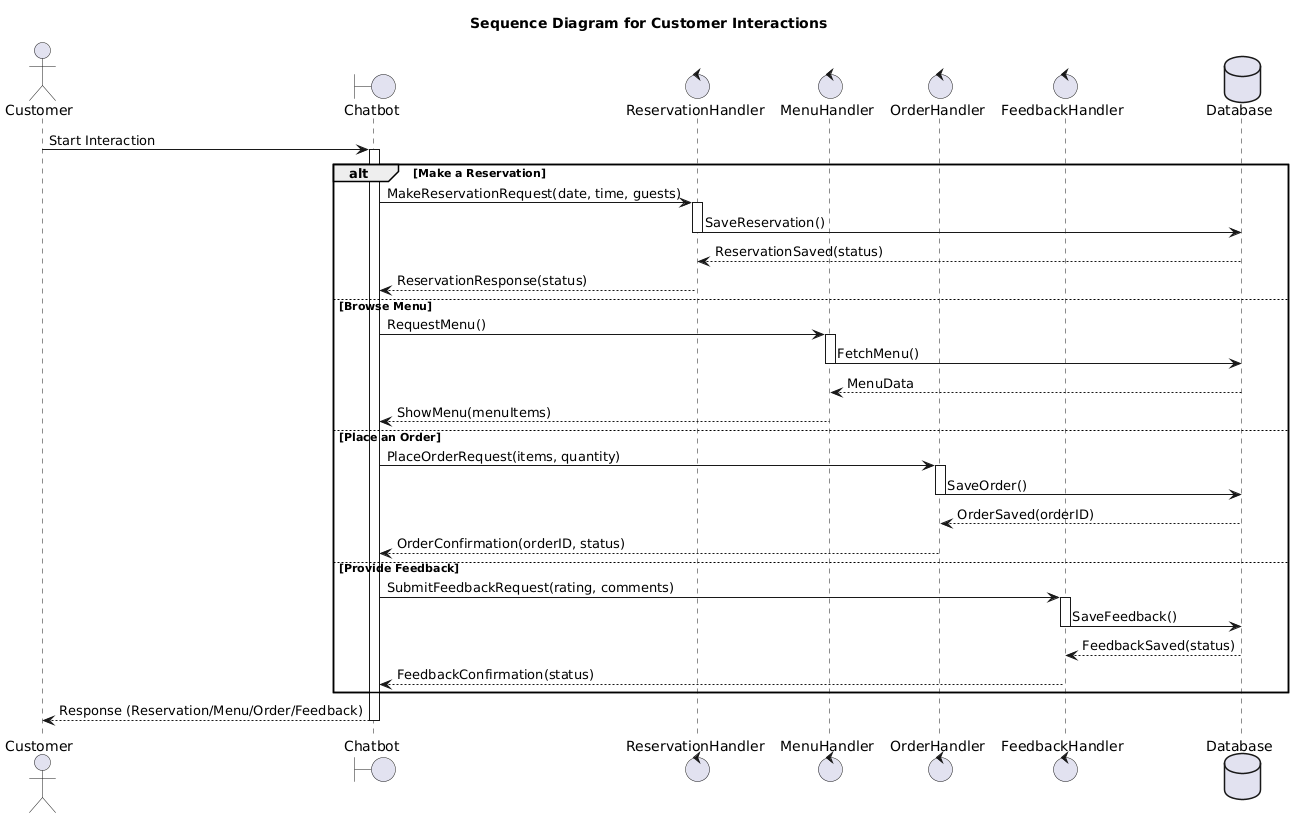
2.Entity Relationship Diagram (ERD) 

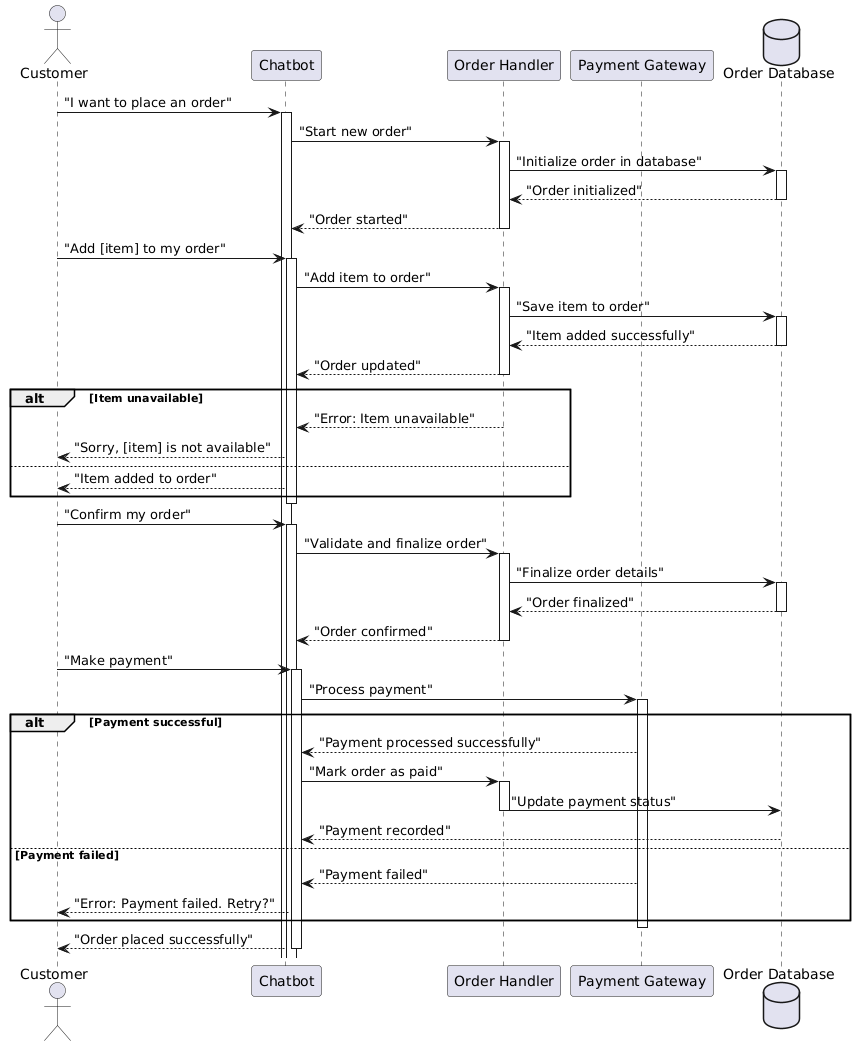
## Sequence Diagrams

## **Sequence Diagram of Reservation**

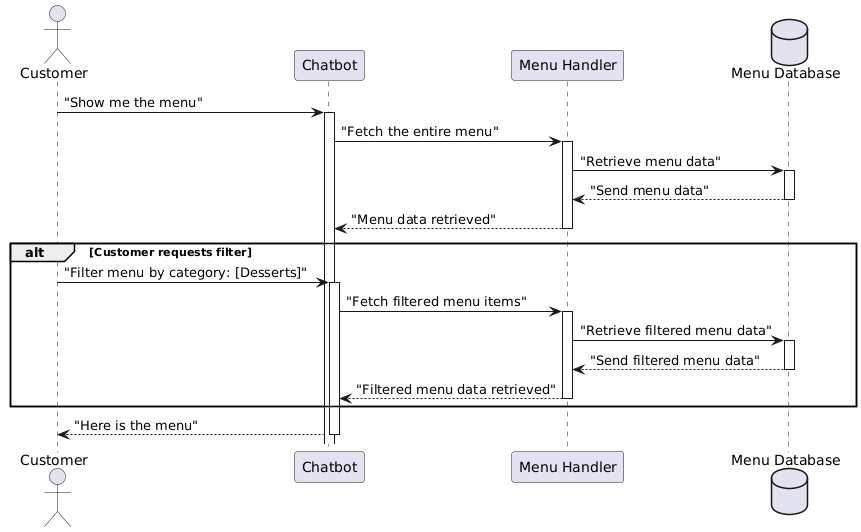
****

**Sequence Diagram of Customer**

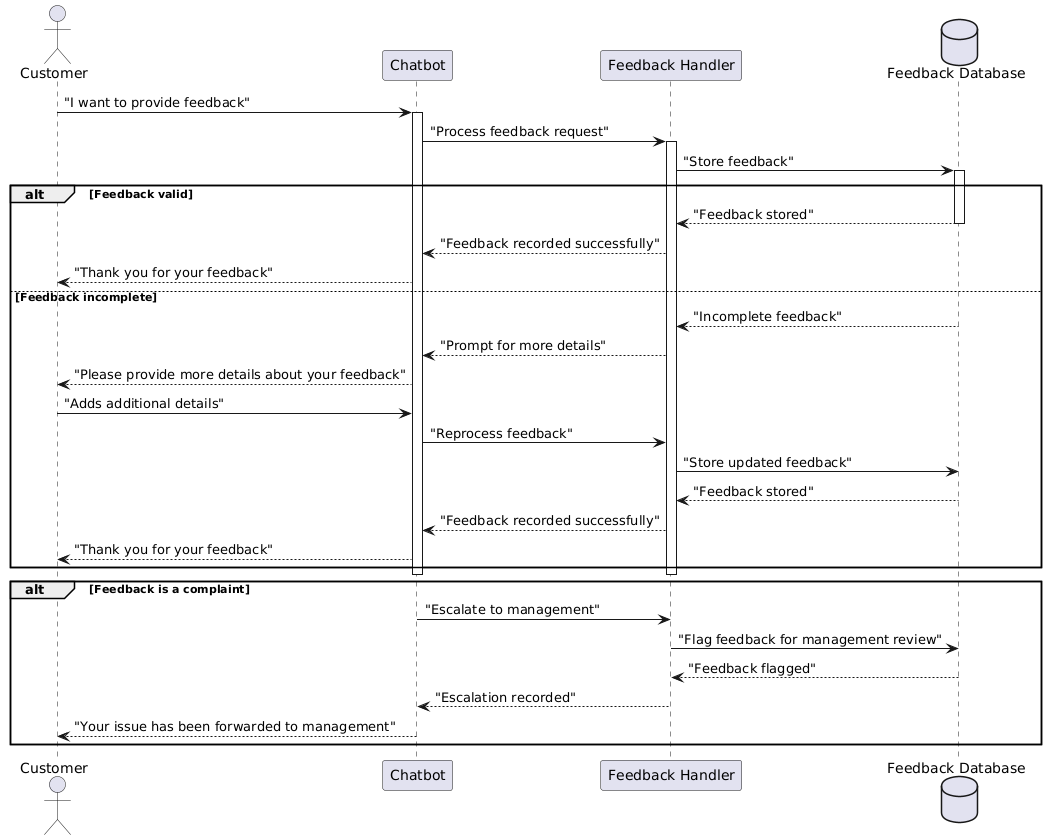


**Sequence Diagram of order** 

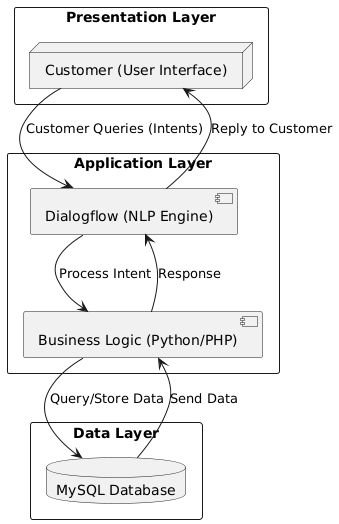
**Sequence Diagram of Menu**



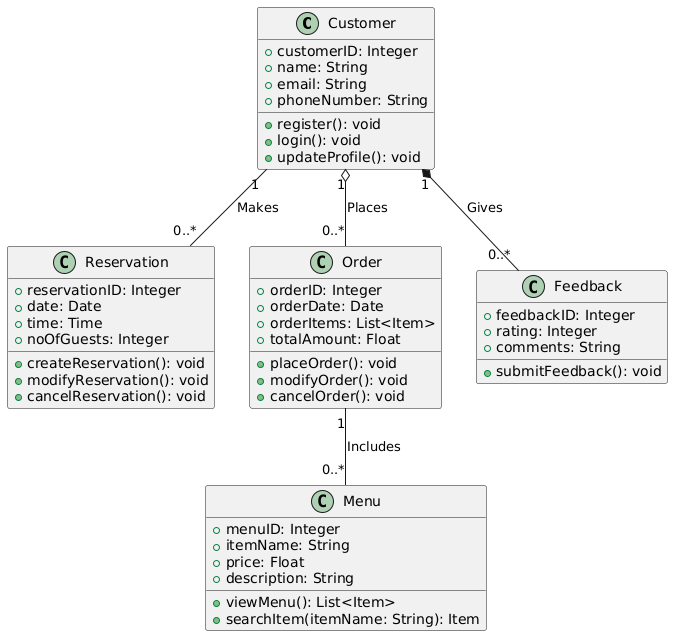
**Sequence Diagram of Feedback**

****

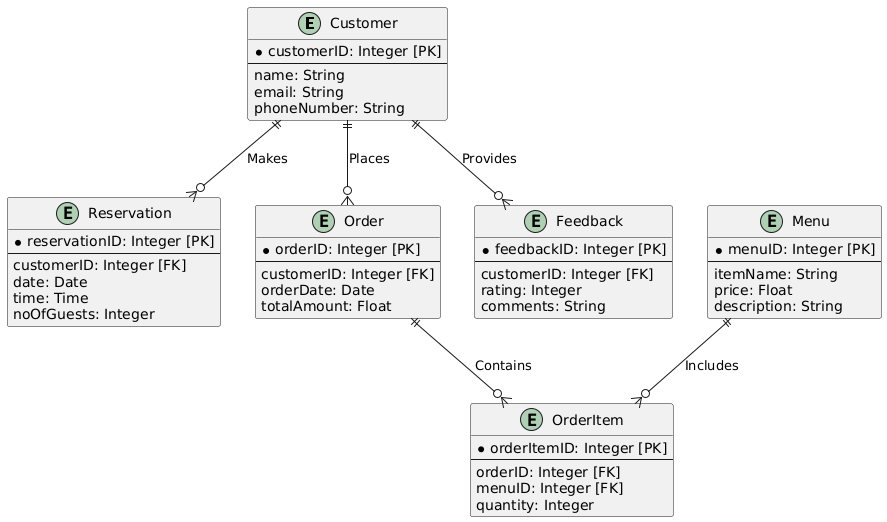
**4.Architecture Design Diagram**

****

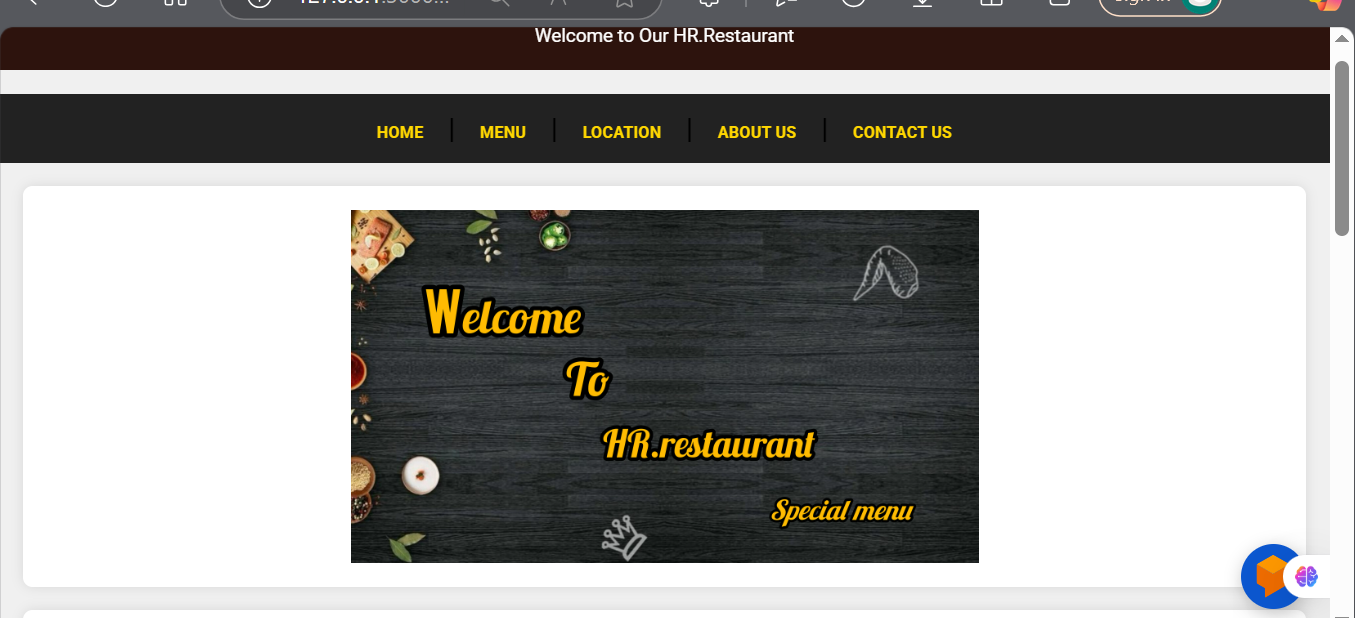
**5.Class Diagram**

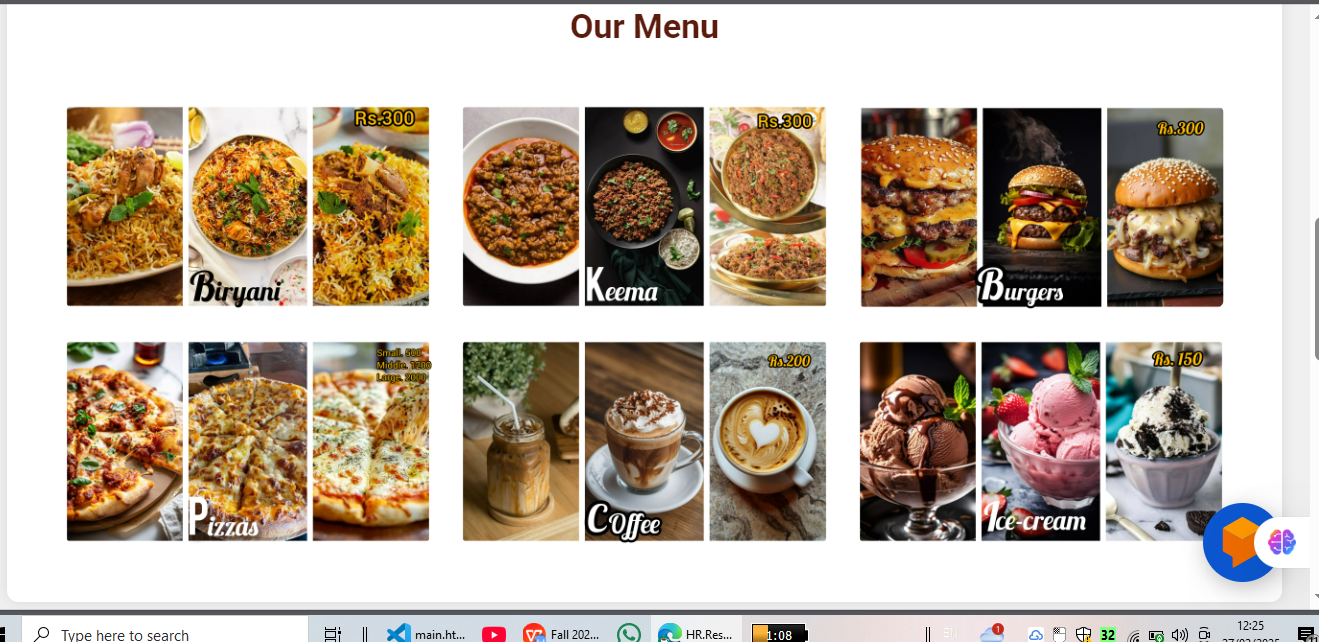


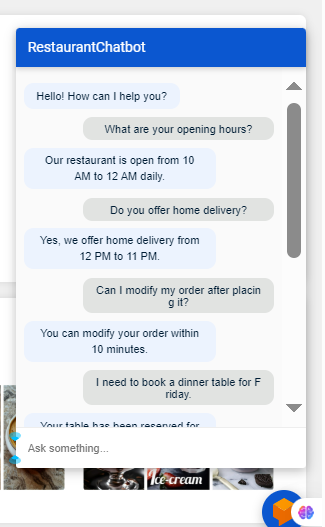
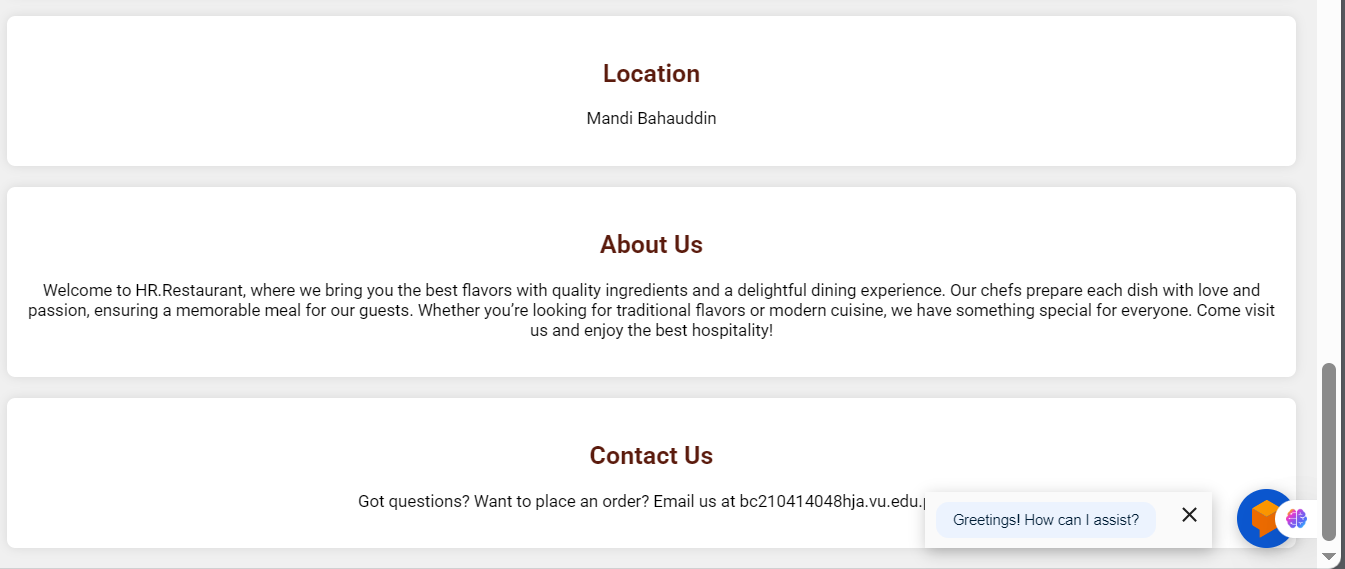
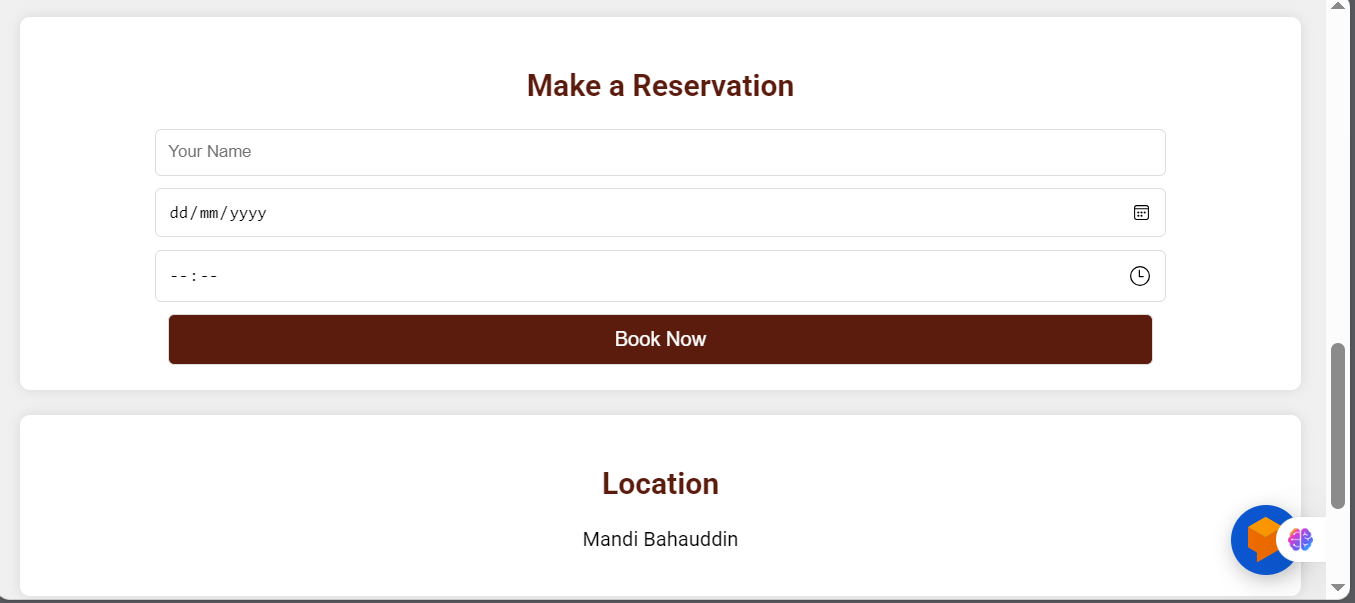
**6.Database Design**

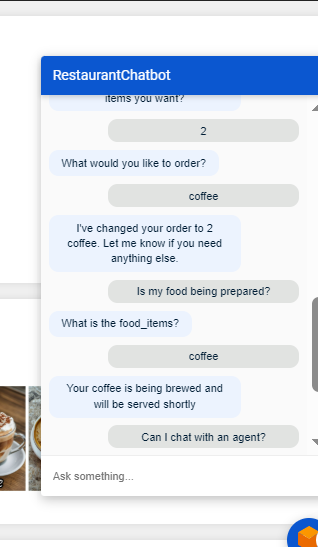
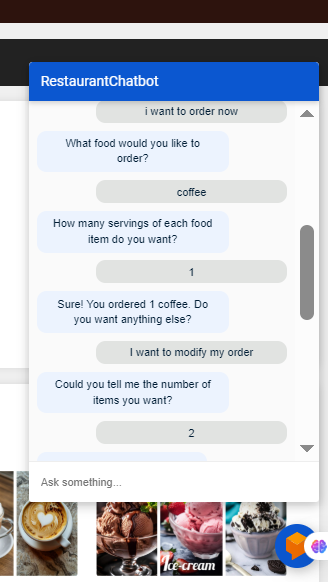


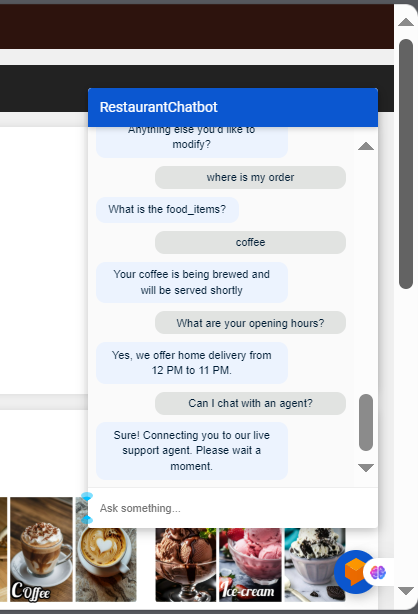
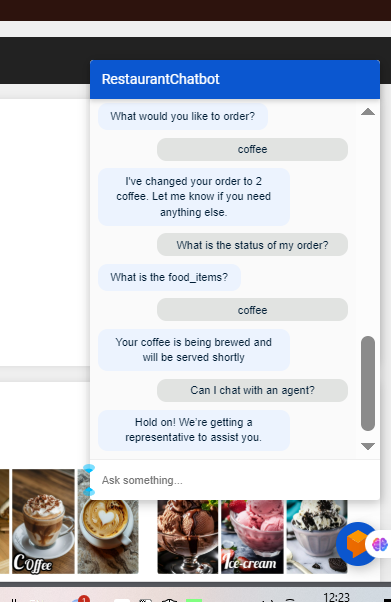
**7.Interface Design**

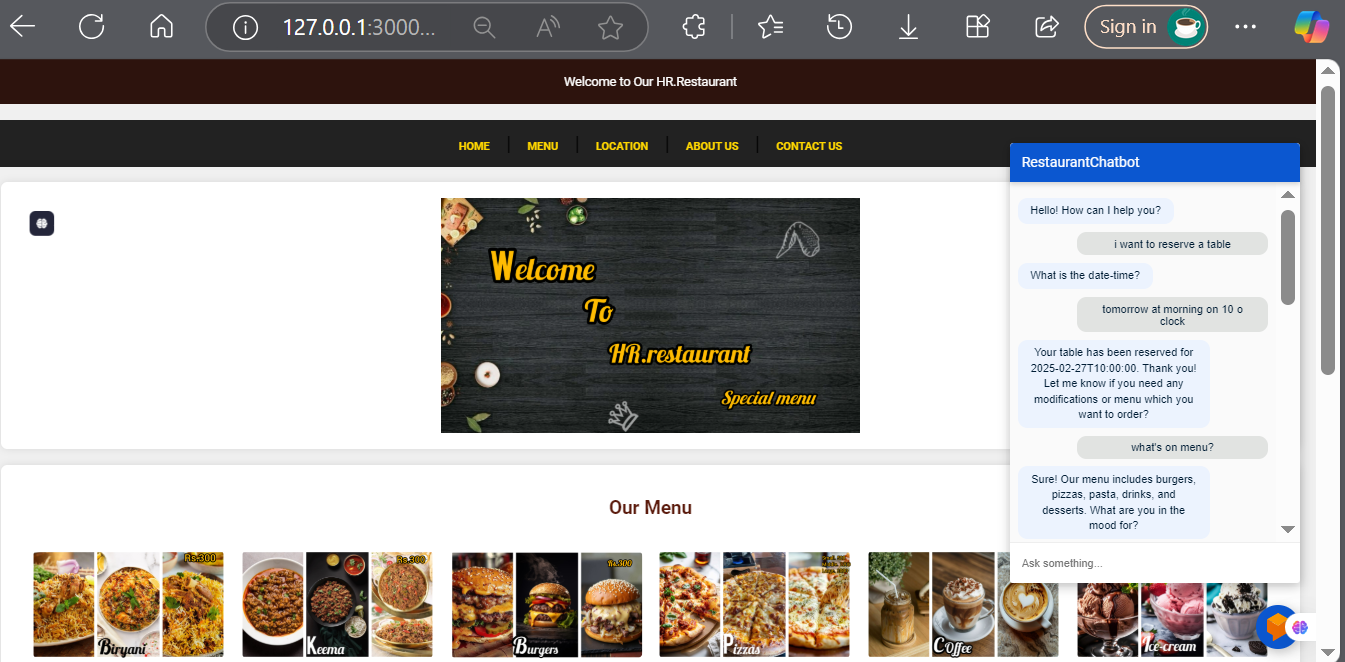












**8.Test Cases**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Test Case ID** | **Description** | **Input** | **Expectation** | **Output** | **Pass/Fail** | **Tested By** |
| **MR-01** | Verify reservation creation | "Book a table for 2 at 7 PM" | Reservation should be created successfully | Reservation ID generated | Pass/Fail | BC210414048 |
| **MR-02** | Validate reservation modification | "Change my reservation to 3 guests" | Reservation should update guest count | Updated reservation details | Pass/Fail | BC210414048 |
| **MR-03** | Check reservation cancellation | "Cancel my reservation" | Reservation should be removed | Confirmation of cancellation | Pass/Fail | BC210414048 |
| **MN-01** | Validate menu navigation | "Show me the menu" | Menu should be displayed | Menu items retrieved | Pass/Fail | BC210414048 |
| **MN-02** | Validate menu category selection | "Show me desserts" | Dessert category should be displayed | Dessert items listed | Pass/Fail | BC210414048 |
| **PO-01** | Verify order placement | "I want to order a coffee" | Order should be created successfully | Order ID generated | Pass/Fail | BC210414048 |
| **PO-02** | Validate order modification | "Add a sandwich to my order" | Order should update with new item | Updated order details | Pass/Fail | BC210414048 |
| **PO-03** | Check order tracking | "Where is my order?" | Order status should be displayed | Current order status retrieved | Pass/Fail | BC210414048 |
| **CS-01** | Validate FAQ response | "What are your opening hours?" | Correct response should be given | Accurate business hours displayed | Pass/Fail | BC210414048 |
| **CS-02** | Verify feedback submission | "I want to give feedback" | Chatbot should prompt for feedback | Feedback recorded | Pass/Fail | BC210414048 |
| **CS-03** | Check live chat support | "I need help with my order" | Chatbot should connect to support | Live chat initiated | Pass/Fail | BC210414048 |