****EatzAI****

(A modern solution for effortless ordering and dining.)

**Software Requirements Specification**

(**Restaurant Chatbot Project**)

Version 1.0



**Group Id: F24PROJECT7B8AC**

**Supervisor Name : Abdullah Qamar    (abdullah.qamar@vu.edu.pk)**

**Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date (dd/mm/yyyy)** | **Version** | **Description** | **Author** |
| 15/11/2024 | 1.0 | The project focuses on developing a restaurant-specific chatbot using Google Dialogflow, leveraging Natural Language Processing (NLP) to automate essential customer interactions. The chatbot will streamline tasks such as table reservations, order placement, menu navigation, and customer support, enhancing the overall dining experience. By offering a conversational interface, it aims to improve operational efficiency, reduce human intervention, and provide a seamless, user-friendly solution for restaurant customers. | Bc200417404 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**Table of Contents**

1. [Scope (of the project)](#scope)
2. [Functional Requirements Non Functional requirements](#FRNFR)
3. [Use Case Diagram](#UCD)
4. [Usage Scenarios](#UCS)
5. [Adopted Methodology](#Adopted)
6. [Work Plan (Use MS Project to create Schedule/Work Plan)](#Gantt)

**SRS Document**

**Scope of Project:**

This project, titled "NLP Chatbot Development using Dialogflow," aims to develop a comprehensive chatbot system tailored for the restaurant industry. The chatbot will facilitate table reservations, order-taking, menu navigation, and customer support. It will provide an interactive, user-friendly experience that automates standard customer interactions and optimizes service delivery. The system will enhance operational efficiency by addressing repetitive queries and common service tasks.

**Project Overview:**

The chatbot system will leverage NLP technology through Dialogflow to interpret user inputs and generate accurate, context-aware responses. The primary functions include booking reservations, accepting food orders, assisting users with menu details, and answering frequently asked questions. Additionally, the system will integrate with existing restaurant management systems to ensure real-time updates and synchronization.

**Scope Boundaries**

The chatbot is designed to manage standard customer interactions but will not address complex customer complaints or personalized service management beyond predefined parameters. Payment processing and loyalty program integration are outside the scope of this project.

Functional and non Functional Requirements:

**Functional Requirements:**

### 1. ****Table Reservations****

* Users can reserve tables by providing details such as name, phone number, email, date, time, and party size.
* The chatbot confirms the reservation and provides a unique booking ID.

### 2. ****Order-Taking****

* Users can place orders by selecting items from the menu.
* The chatbot confirms the order, calculates the total cost, and logs the order for kitchen processing.

### 3. ****Menu Navigation****

* Users can request information about menu items, including prices, ingredients, and special offers.
* The chatbot can provide personalized recommendations based on user preferences or past orders.

### 4. ****Customer Support****

* The chatbot answers frequently asked questions, such as hours of operation, contact details, or directions.
* For complex inquiries, it directs the user to a human representative or collects contact details for follow-up.

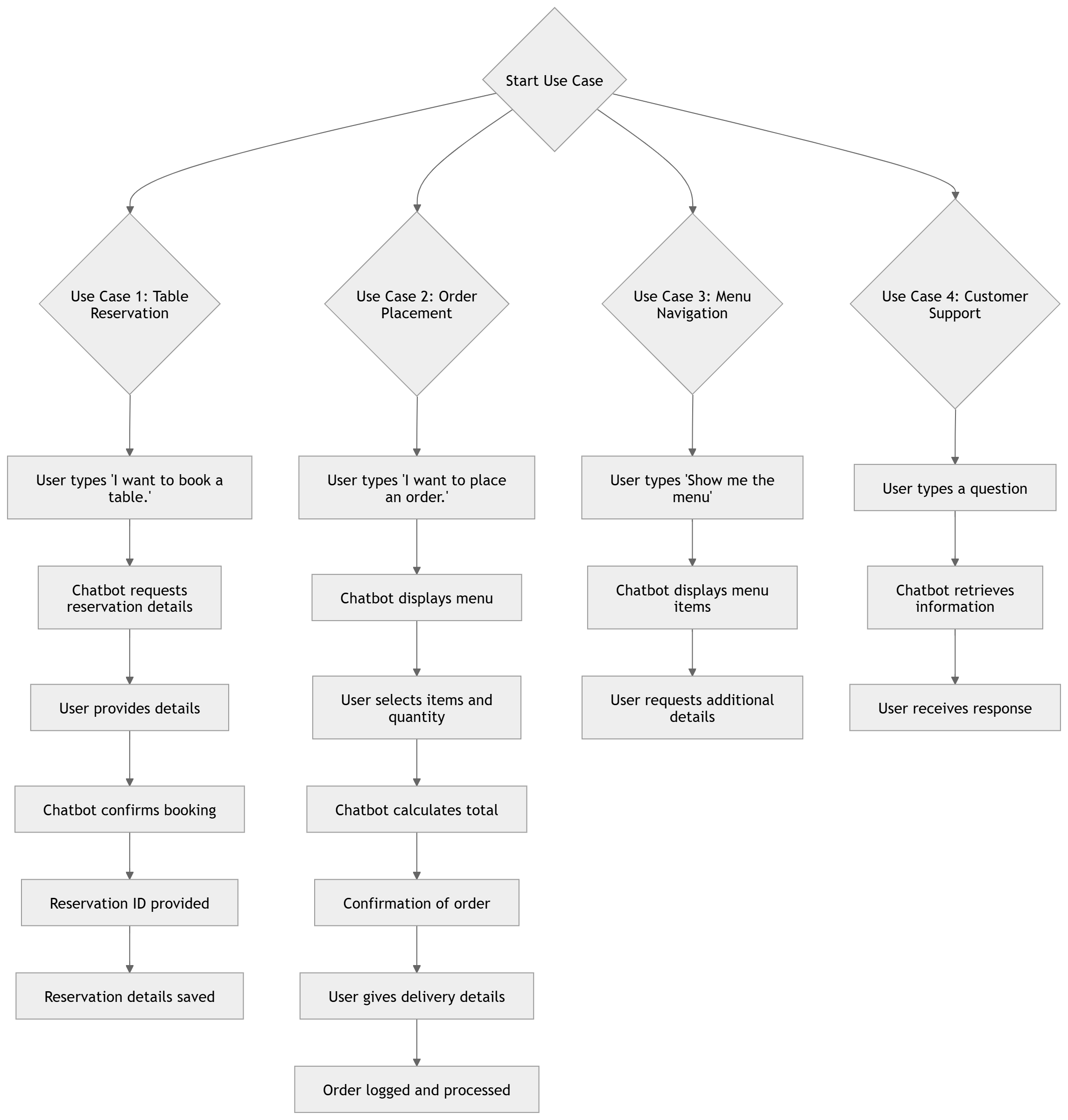
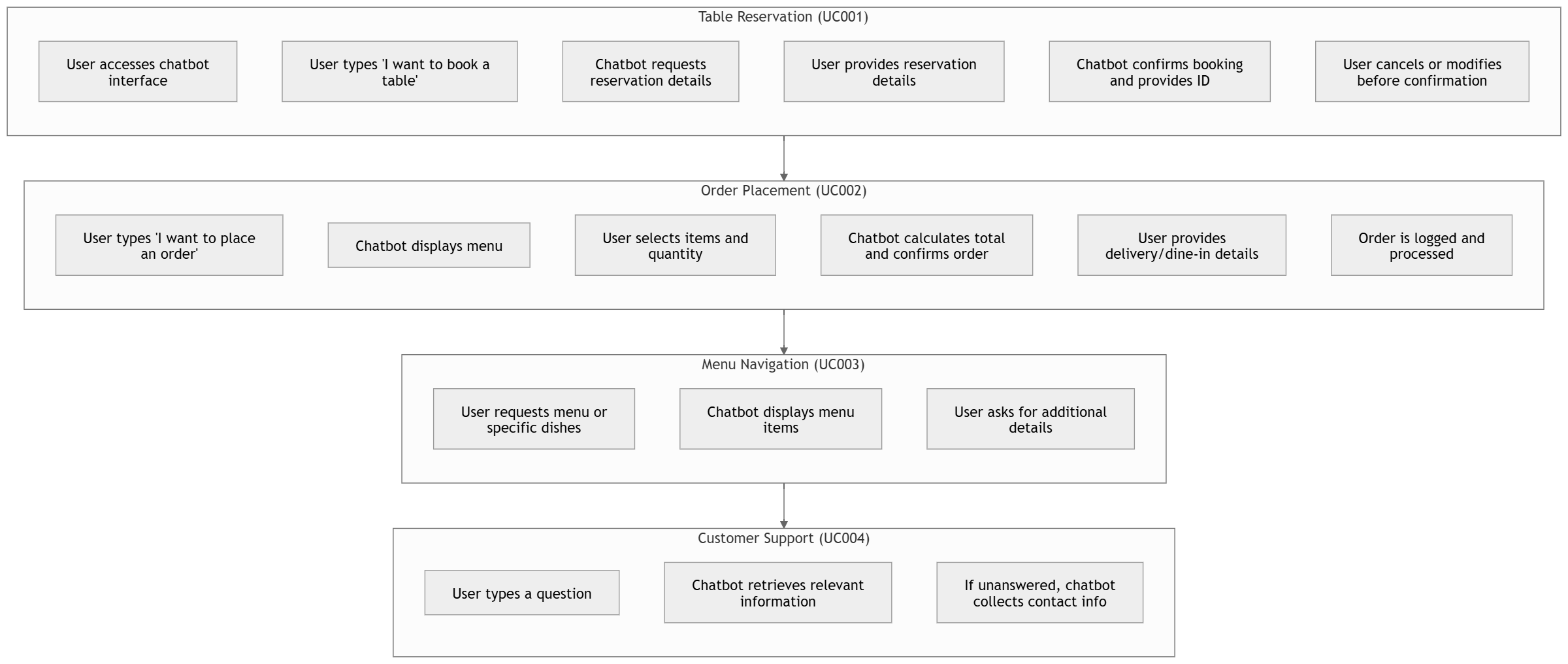
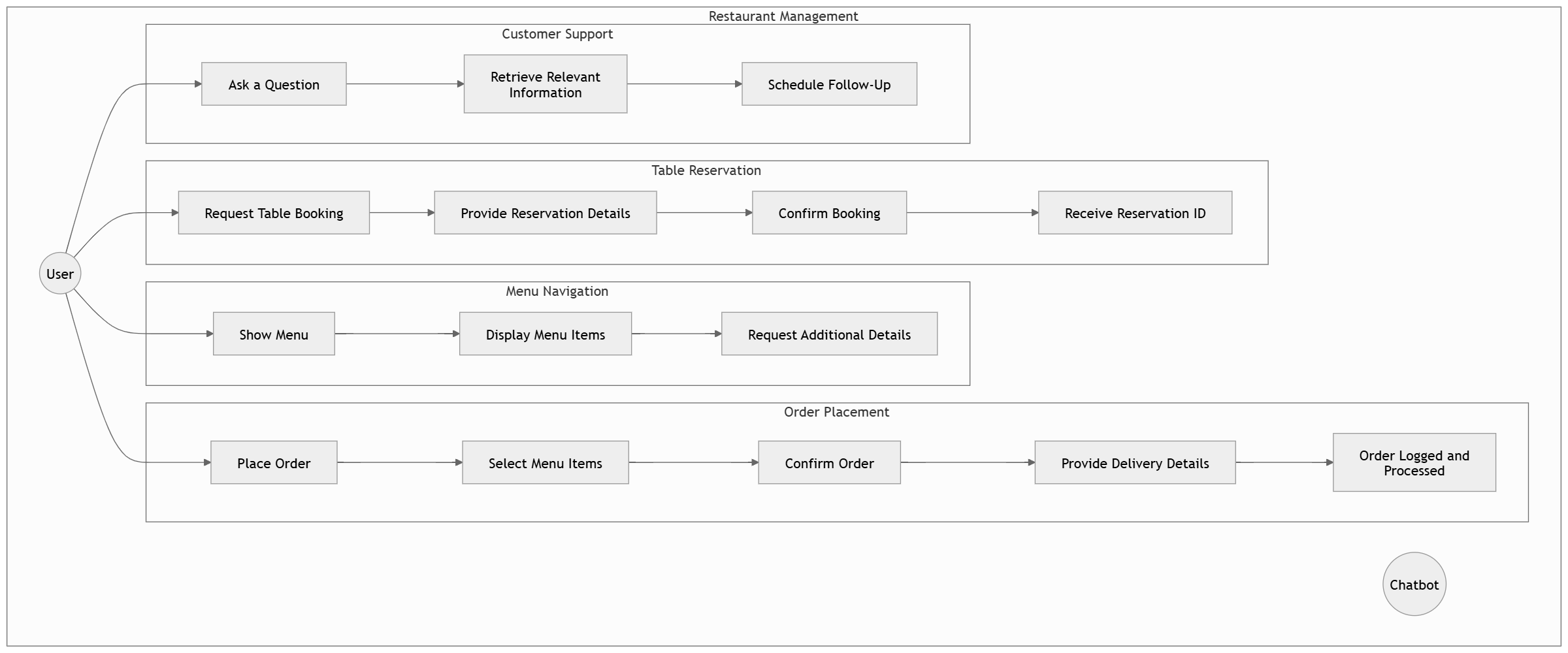
**Non-Functional Requirements:**

1. **Response Time:** The chatbot should respond to user inputs within **10 to 30 seconds** to maintain a smooth interaction flow.
2. **Scalability:** The system should be able to handle interactions with up to **50 concurrent users,** which is sufficient for a small-scale deployment.
3. **Security:** Basic security measures should be in place to protect user data, such as contact details, ensuring no unauthorized access. For this project, data encryption and password protection can be implemented.
4. **Availability:** The chatbot should be accessible 24/7 via web and mobile platforms.

**Assumptions and Constraints:**

* Users do not need to log in to interact with the chatbot.
* The restaurant’s menu and table availability data are maintained by an admin.
* The chatbot will only handle common restaurant operations (reservations, orders, menu navigation, and FAQs).

Use Case Diagram(s):



Usage Scenarios:

### ****Use Case 1: Table Reservation****

**Use Case ID:** UC001

* **Actors:** User, Chatbot
* **Preconditions:**
  + User accesses the chatbot interface via web or mobile.
  + Table availability data is pre-configured.
* **Main Flow:**
  + User types "I want to book a table."
  + Chatbot requests reservation details (name, phone number, date, time, and party size).
  + User provides the details.
  + Chatbot confirms the booking and provides a reservation ID.
* **Alternative Paths:**
  + User cancels or modifies the reservation before confirmation.
* **Postconditions:**
  + Reservation details are saved, and the user receives confirmation.

### ****Use Case 2: Order Placement****

**Use Case ID:** UC002

* **Actors:** User, Chatbot
* **Preconditions:**
  + The restaurant’s menu is pre-loaded into the system.
* **Main Flow:**
  + User types "I want to place an order."
  + Chatbot displays the menu and prompts the user to select items.
  + User selects items and specifies the quantity.
  + Chatbot calculates the total and confirms the order.
  + User provides delivery or dine-in details.
  + Order is logged and sent for processing.
* **Alternative Paths:**
  + Chatbot suggests alternative items if a selected item is unavailable.
* **Postconditions:**
  + Order details are logged, and the user receives an order confirmation.

### ****Use Case 3: Menu Navigation****

**Use Case ID:** UC003

* **Actors:** User, Chatbot
* **Preconditions:**
  + Menu details (dish names, descriptions, prices, etc.) are available.
* **Main Flow:**
  + User types "Show me the menu" or queries about specific dishes.
  + Chatbot displays the requested menu items.
  + User asks for additional details, such as ingredients or recommendations.
* **Alternative Paths:**
  + Chatbot suggests popular dishes or daily specials based on user preferences.
* **Postconditions:**
  + User receives the requested menu details or recommendations.

### ****Use Case 4: Customer Support****

**Use Case ID:** UC004

* **Actors:** User, Chatbot
* **Preconditions:**
  + FAQ database is pre-loaded.
* **Main Flow:**
  + User types a question, such as "What are your operating hours?"
  + Chatbot retrieves and provides the relevant information.
* **Alternative Paths:**
  + If the question is not in the database, the chatbot collects user contact information for follow-up.
* **Postconditions:**
  + User receives a response or follow-up is scheduled.

Adopted Methodology

For this project, we will use a **Hybrid Approach** by combining **Waterfall** and **Agile** methodologies. This mix allows us to start with a clear plan and adapt as we build and improve the chatbot.

### ****Phase 1: Planning (Waterfall)****

1. **Define Requirements**: We will list all the features and tasks the chatbot needs to perform, like table reservations and order placement.
2. **Document the Scope**: Create detailed documentation so everyone on the team knows what we are building and why.

This phase ensures we start with a clear understanding of the project.

### ****Phase 2: Development (Agile)****

1. **Work in Small Steps (Sprints)**: Break the work into smaller parts (called sprints). Each sprint will focus on building one or two features, like table booking or menu navigation.
2. **Start with Core Features**: Begin by building the most important features first, such as reservations and order processing.
3. **Collect Feedback**: After completing each sprint, we will test the chatbot and get feedback. This will help us make improvements before moving to the next sprint.

This phase lets us build the chatbot step-by-step while making changes along the way.

### ****Why Use This Hybrid Approach?****

1. **Waterfall Helps Plan Better**: In the beginning, we use Waterfall to create a roadmap. This helps us stay organized and ensures nothing important is missed.
2. **Agile Adds Flexibility**: Later, Agile lets us work flexibly, so we can adapt to new ideas or solve problems as we go.

This approach keeps us focused but flexible, making it easier to create a chatbot that works well and meets user needs.

Work Plan (Use MS Project to create Schedule/Work Plan/Task Sheet)

Link:

<https://docs.google.com/spreadsheets/d/1txVrSf2JnRMNp01zcrsgGPpG6lDne7lDnGP4x9_s42k/edit?usp=sharing>

|  |  |  |  |
| --- | --- | --- | --- |
| **Phase** | **Duration** | **Start Date** | **End Date** |
| Requirement Analysis | 1 Week | 11/27/2024 | 12/4/2024 |
| Design and Prototyping | 2 Weeks | 12/5/2024 | 12/19/2024 |
| Development - Phase 1 | 3 Weeks | 12/20/2024 | 1/2/2025 |
| Development - Phase 2 | 3 Weeks | 1/2/2025 | 1/23/2025 |
| Testing and Deployment | 2 Weeks | 1/24/2025 | 2/7/2025 |