**NLP Chatbot Development using Dialogflow**

**Software Requirements Specification**

Version 1.0



**Group Id: <F24PROJECT2A3F7>**

**Supervisor Name :< Abdullah Qamar >**

**Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date (dd/mm/yyyy)** | **Version** | **Description** | **Author** |
| Current date | 1.0 | Introduction of the project | Write student(s) id |
| 13/11/2024 | 1.0 | Initial SRS Document | Bc210412114 |
|  |  |  |  |
|  |  |  |  |

**Table of Contents**

[Scope of the Project: 1](#_Toc184059020)

[Functional and non Functional Requirements: 2](#_Toc184059021)

[Use Case Diagrams: 3](#_Toc184059022)

[Usage Scenarios: 6](#_Toc184059023)

[Adopted Methodology 9](#_Toc184059024)

[Work Plan 9](#_Toc184059025)

**SRS Document**

# Scope of the Project:

**Purpose:**

The project aims to design and develop a chatbot powered by Google Dialogflow to automate customer interactions for a restaurant. The chatbot will leverage Natural Language Processing (NLP) to understand user inquiries and provide accurate, context-aware responses. By automating routine tasks such as answering customer queries, managing reservations, and processing orders, the system will enhance both customer experience and operational efficiency.

**Intended Functionalities:**

Automating Customer Interaction: Without requiring human assistance, the chatbot will manage routine customer interactions including bookings, responding to inquiries, and offering assistance.

**Task Handling:**

For a Restaurant, the chatbot will manage table reservations, order-taking, menu navigation, and customer support.

**Seamless User Experience:** The system will interact conversationally with users, ensuring that their needs are addressed quickly and efficiently.

**Multi-Language Support:** The chatbot will support multiple languages to cater to a diverse user base.

**System Capabilities:**

Natural Language Processing (NLP): The chatbot will understand and interpret human language, using Google Dialogflow's advanced NLP capabilities to generate meaningful responses.

**Backend Integration:** The system will connect to a backend (PHP/Python) for storing and retrieving data, such as reservation details, order information, or user queries.

**Database Management:** The system will use MySQL to store user information, transaction logs, and system activities.

**What the System Will Accomplish:**

Automate repetitive tasks like responding to FAQs, processing orders, and handling reservations.

Improve response times by handling multiple users simultaneously with minimal latency.

Increase customer satisfaction by providing an always-available, interactive platform.

Enhance operational efficiency by reducing the need for human intervention in routine tasks.

**Limitations:**

**Complex Queries:** The system is not designed to handle highly complex or ambiguous queries that require deep expertise or decision-making.

**Medical Diagnoses:** In the Pharmacy Store case, the system will not provide any medical advice or diagnostic services—its role is limited to helping customers find products and set reminders.

**Advanced Customer Service:** The system will not replace human representatives for complex customer service cases that require escalation.

This project focuses on delivering a reliable, interactive chatbot to streamline customer-facing operations, allowing businesses to better serve their clients while maintaining efficiency.

# Functional and non Functional Requirements:

**Functional Requirements**

1. My project will allow customers to browse the restaurant’s menu, categorized into sections such as appetizers, main courses, desserts, and beverages.
2. It will enable customers to check table availability and make reservations by selecting the date, time, and party size, ensuring a smooth reservation process.
3. The chatbot will facilitate order placement.
4. My project will provide instant responses to FAQs, answering questions about the restaurant’s operating hours, location, and special events.
5. My project will integrate with a MySQL database to store and manage menu items, reservation details, and customer orders.
6. It will support multiple languages, catering to a diverse user base and ensuring an inclusive user experience.

**Non-Functional Requirements**

1. My project will be scalable, capable of handling multiple users simultaneously during peak hours without performance issues.
2. It will ensure high availability, operating 24/7 with a minimum uptime of 99.9%.
3. The chatbot will provide fast responses, with a response time of no more than 2 seconds for user queries.
4. My project will prioritize security, protecting customer data through HTTPS encryption and secure database storage.
5. It will offer language-specific responses, enabling smooth communication in multiple languages using Dialogflow.
6. The system will be reliable, with regular stress testing to minimize downtime during high-traffic periods.
7. My project will be easy to maintain, with a modular design that allows for quick updates to menu items, promotions, and features.
8. It will be accessible across multiple platforms, including the restaurant’s website, mobile app, and social media channels.
9. The chatbot will offer a simple and user-friendly interface, with interactive elements like quick reply buttons for common tasks.
10. My project will integrate analytics tools to track user interactions and gather insights for continuous improvement of services.

# Use Case Diagrams:

**Use Case Diagram Description: Restaurant Chatbot System**

The Restaurant Chatbot System is designed to streamline customer interaction with the restaurant, providing essential services such as menu browsing, table reservations, order placements, and frequently asked question (FAQ) handling through a conversational interface. The system also supports administrative functions, such as viewing reports, which are essential for restaurant management.

Actors:

1. Customer:

- Primary role: The customer interacts with the chatbot to perform various tasks related to the restaurant services.

- Use Cases:

- Menu Navigation: The customer uses the chatbot to browse the restaurant’s menu and explore available food and drink options.

- Table Reservation: The customer reserves a table for their desired time through the chatbot, allowing for easy scheduling of dining.

- Order Placement: After browsing the menu, the customer places an order for food or drinks via the chatbot.

- FAQ Handling: The customer can inquire about common questions related to the restaurant, such as operating hours, available promotions, or any other information facilitated by the chatbot.

2. Administrator:

- Primary role: The administrator manages and monitors the restaurant chatbot's performance, ensuring that the system works as expected and reviewing reports for insights into customer activity.

- Use Case:

- View Reports: The administrator accesses the system’s reports, which might include information about customer orders, reservations, or system usage, in order to make data-driven decisions.

**Use Cases and Relationships:**

- Menu Navigation: This use case enables customers to explore the restaurant’s offerings. It helps customers make informed decisions before proceeding to order or make a reservation.

- Table Reservation: Customers interact with the chatbot to book a table, ensuring they have a designated spot at the restaurant for their dining experience.

- Order Placement: After reviewing the menu, the customer can place their order through the chatbot. This use case ensures a seamless and quick ordering process, improving customer experience.

- FAQ Handling: The chatbot addresses frequent questions, which could be about the restaurant’s policies, location, or any other relevant information, providing quick responses to customer inquiries.

- View Reports: This use case is available exclusively to the Administrator. The reports might include data like total orders, peak times, popular menu items, and other analytics to help in business decision-making.

**Connections and Relationships:**

- Customer – Menu Navigation, Table Reservation, Order Placement, FAQ Handling

- The Customer interacts with all the core functions of the restaurant chatbot. The arrows denote that the customer directly engages with these use cases.

- Administrator – View Reports:

- The Administrator is connected only to the View Reports use case, as they are responsible for reviewing the performance and analytics of the chatbot and restaurant operations.

System Boundary:

The Restaurant Chatbot System is represented by the system boundary that encompasses all the use cases within the diagram. This boundary demarcates the interaction between external actors (Customer and Administrator) and the functionalities provided by the system.

Key Features and Functionalities:

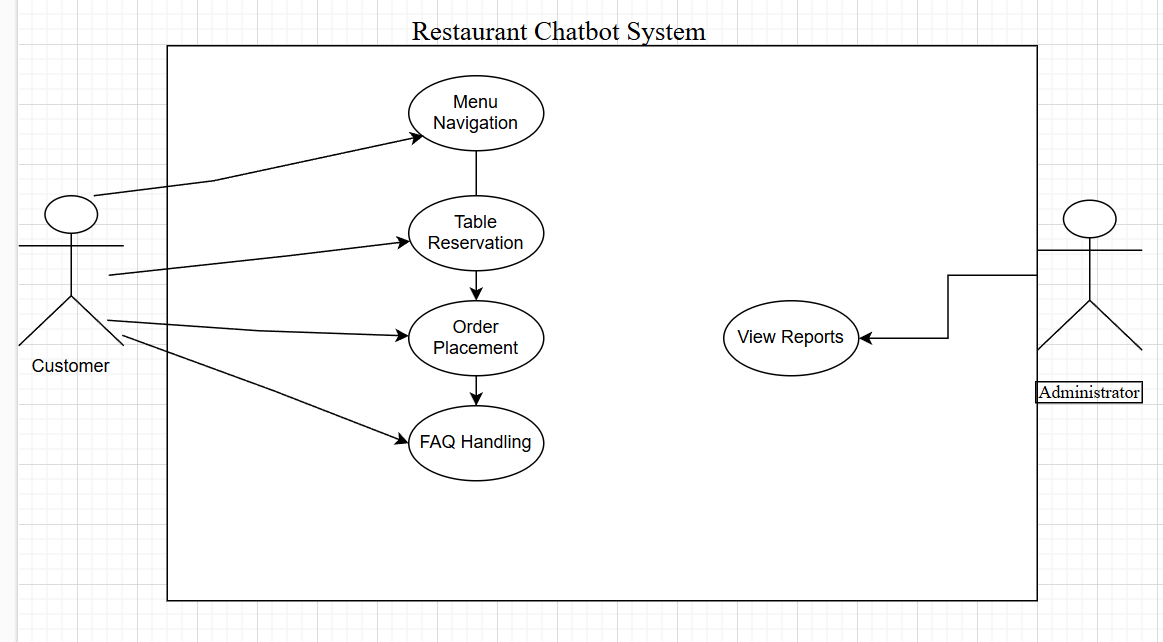
- Customer-Centric Services: The system primarily focuses on enhancing the customer experience through an interactive and accessible chatbot.

- Administrative Control: Through the View Reports use case, administrators can monitor the system’s performance, ensuring the chatbot is serving customers effectively.

**Additional Notes:**

- The use case diagram does not show direct interactions between Customer and Administrator since their responsibilities within the system are separate.

- The Customer is solely focused on interaction with the chatbot for service-oriented tasks (like reservations, ordering, and information retrieval), while the Administrator handles reporting and performance review tasks.



# Usage Scenarios:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Use Case Title** | **Use Case ID** | **Actions** | **Description** | **Alternative Paths** | **Pre-Conditions** | **Post-Conditions** | **Author** | **Exceptions** |
| **Menu Navigation** | UC-01 | 1. User opens the chatbot.  2. User selects “View Menu” option.  3. Chatbot displays menu categories.  4. User selects category.  5. Chatbot displays items. | Allows customers to browse the restaurant menu interactively by selecting categories such as appetizers, main courses, and desserts. | If the user does not specify a category, the chatbot can display the full menu. | User must initiate interaction and request menu information. | Menu details are displayed to the user. |  | If the menu database is unavailable, the chatbot notifies the user about the issue and suggests trying again later. |
| **Table Reservation** | UC-02 | 1. User selects “Reserve a Table.”  2. Chatbot prompts for date, time, and number of guests.  3. User provides details.  4. Chatbot confirms availability. | Enables users to book tables by specifying their preferred time and party size, ensuring convenience in making reservations. | If the requested time is unavailable, the chatbot suggests alternate slots. | Reservation data (availability) must be up-to-date in the database. | Reservation details are recorded, and the user receives a confirmation. |  | If the database cannot be accessed, the chatbot informs the user of a temporary issue and asks them to try again later. |
| **Order Placement** | UC-03 | 1. User selects “Place an Order.”  2. Chatbot asks for item selection.  3. User adds items to the cart.  4. User confirms the order. | Users can place orders for dine-in or takeout by interacting with the chatbot, including adding items, customizing orders, and confirming payment methods. | If the user adds unavailable items, the chatbot suggests alternatives. | The menu and order database should be functional and synced. | Order is logged in the system, and a confirmation is sent to the user and restaurant staff. |  | If there’s an issue with order logging or inventory mismatch, the chatbot apologizes and redirects users to customer support. |
| **FAQ Handling** | UC-04 | 1. User asks a question.  2. Chatbot uses NLP to match the query to an FAQ.  3. Chatbot provides an answer. | Provides instant responses to frequently asked questions about operating hours, location, parking, and special events. | If no matching FAQ is found, the chatbot prompts the user to rephrase or connect with customer support. | A list of FAQs and responses must exist in the chatbot’s knowledge base. | User receives an answer or is connected to a human support agent if needed. |  | If the FAQ database is unavailable, the chatbot apologizes and provides contact details for manual assistance. |

# Adopted Methodology

My project will follow the Prototyping Model to quickly develop a working prototype of the chatbot and gather early feedback from stakeholders.

* I will first gather requirements by identifying core functionalities such as menu navigation, table reservations, order management, and customer support.
* Next, a basic prototype of the chatbot will be developed using Google Dialogflow to demonstrate essential features.
* The prototype will be tested by potential users and stakeholders to gather feedback and identify areas for improvement.

My project will then adopt the Incremental Development Model to build and refine the chatbot in stages.

* Each feature will be developed and tested independently before being integrated into the main system.
* For example, I will first complete the table reservation feature, then move on to order management, followed by customer support and promotional notifications.

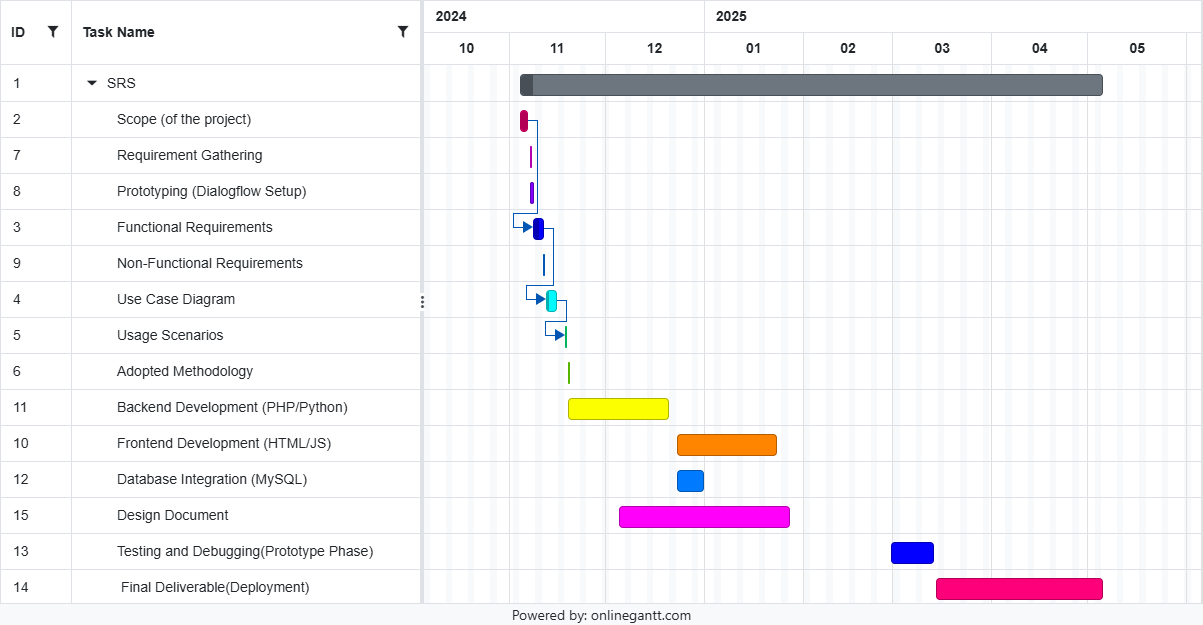
Usability testing will be conducted after each increment to ensure that each new feature integrates smoothly with the existing system.

Continuous testing and integration will ensure that the system remains stable and functional as new features are added.

My project will also involve regular user feedback after each development stage, helping to refine the chatbot’s functionality and user experience.

The final version of the chatbot will undergo comprehensive testing, including stress testing, to ensure reliability and performance during peak usage.

# Work Plan



**Project Phases and Milestones**

**SRS Development (October 2024 – May 2025)**

This overarching task spans the entire duration of the project, indicating that the SRS (Software Requirements Specification) document will be developed and refined throughout the project lifecycle.

**Scope of the Project (October 2024)**

The project scope is defined at the start to ensure clear objectives and boundaries for the chatbot development.

**Requirement Gathering (October 2024)**

Gathering functional and non-functional requirements, user needs, and system specifications.

**Prototyping (Dialogflow Setup) (October 2024)**

Initial chatbot prototype development using Google Dialogflow to test basic functionalities.

**Functional and Non-Functional Requirements (November 2024)**

Detailed documentation of the functional capabilities (e.g., menu navigation, reservations) and non-functional attributes (e.g., scalability, response time).

**Use Case Diagram (November 2024)**

Creating a UML diagram to visualize the system's interactions with users (customers and administrators).

**Usage Scenarios (November 2024)**

Outlining detailed interaction flows for various chatbot tasks like order placement and FAQ handling.

**Adopted Methodology (November 2024)**

Selecting and documenting the project development methodology (e.g., Prototyping and Incremental Development).

**Backend Development (PHP/Python) (December 2024 – February 2025)**

Developing backend logic to handle chatbot requests, connect to the database, and integrate with Dialogflow.

**Frontend Development (HTML/CSS/JS) (January 2025 – February 2025)**

Creating a user-friendly interface for interaction with the chatbot.

**Database Integration (MySQL) (December 2024 – February 2025)**

Setting up and integrating the database to manage menu items, reservations, and user interactions.

**Design Document (February 2025 – March 2025)**

Preparing a comprehensive design document outlining system architecture and workflows.

**Testing and Debugging (Prototype Phase) (March 2025)**

Conducting thorough testing of the prototype to identify and resolve bugs, ensuring system stability.

**Final Deliverable (Deployment) (April 2025 – May 2025)**

Deployment of the final chatbot system, making it accessible via the restaurant's platforms.