A

Major Project Report

on

AI POWERED FAQ ASSISTANT FOR INSTANT ANSWERS

Submitted in Partial Fulfillment of
the Requirements for the Degree
of

Bachelor of Engineering

in

Computer Engineering

to

**Kavayitri Bahinabai Chaudhari
North Maharashtra University, Jalgaon**

Submitted by

**Nikita S. Hemnani
Kajal T. Patil
Nikita S. Patil
Sunidhi J. Chavan**

Under the Guidance of

Mr. Ashish T. Bhole

DEPARTMENT OF COMPUTER ENGINEERING
SSBT's COLLEGE OF ENGINEERING AND TECHNOLOGY,
BAMBHORI, JALGAON - 425 001 (MS)
2023 - 2024

SSBT's COLLEGE OF ENGINEERING AND TECHNOLOGY, BAMBHORI, JALGAON - 425 001 (MS)

DEPARTMENT OF COMPUTER ENGINEERING

CERTIFICATE

This is to certify that the major project entitled *AI POWERED FAQ ASSISTANT FOR INSTANT ANSWERS*
submitted by

Nikita S. Hemnani
Kajal T. Patil
Nikita S. Patil
Sunidhi J. Chavan

in partial fulfillment of the degree of *Bachelor of Engineering* in *Computer Engineering* has been satisfactorily carried out under my guidance as per the requirement of Kavayitri Bahinabai Chaudhari North Maharashtra University, Jalgaon.

Date: April 14, 2024
Place: Jalgaon

Mr. Ashish T. Bhole
Guide

Dr. Manoj E.Patil
Head

Dr. G. K. Patnaik
Principal

Contents

List of Tables

List of Figures

Acknowledgements

The success and final outcome of a project requires a lot of guidance and assistance from many people and it was extreme privilege to have it all along the completion of project. All Which we have done was only possible with such supervision and assistance and we would like to thank them and express deep gratitude and sincere thanks to all who helped us to complete the project successfully. We would also like to thank Prof. Dr. Girish K. Patnaik, Principal for providing us facilities to complete the work. Deep gratitude goes to Dr. Manoj E. Patil, Associate Professor and Head of Department and Mr. Ashish T. Bhole, Associate Professor and Project Guide for granting us the opportunity to conduct the project work as well as his valuable suggestions and guidance at the time of need.Also a great thanks to Mr. Akash D. Waghmare, Associate Professor and BE Project Incharge for his valuable suggestions and guidance.Sincerely thankful and fortunate enough to get constant encouragement, support and guidance from all teaching staffs of Computer Department of SSBT's COET which helped us in successfully completing the project work. Also would like to extend sincere esteems to all staff in laboratory for the timely support.

Nikita S. Hemnani
Kajal T. Patil
Nikita S. Patil
Sunidhi J. Chavan

Abstract

The motivation behind AI-based chatbots stems from the desire to create automated systems which simulate human-like conversations, providing convenient and efficient solutions for tasks such as customer support, information retrieval, and task automation. In a settled landscape, AI chatbots facilitate seamless customer interactions. Yet, a significant influx of user inquiries beyond the trained scope triggers system confusion. Consequently, customers experience inconsistencies and prolonged query resolution times, negatively impacting user experience. By deploying advanced training techniques, refining error-handling mechanisms, and expanding the chatbot's knowledge base, equilibrium is restored, and users enjoy improved engagement and efficient issue resolution. The approach behind AI-based chatbots involves utilizing natural language processing, machine learning, and large datasets to train models which understand and generate human-like responses, enabling effective communication and interaction with users. Creating an AI-based chatbot yields various results, including improved customer engagement, reduced workload for human agents, faster response times, 24/7 availability, and enhanced user experiences, all contributing to increased operational efficiency and user satisfaction. AI chatbots serve as versatile tools, enhancing customer experiences through personalized interactions, while also aiding businesses in optimizing the operations and support services. The evolving capabilities signify a pivotal shift in human-technology interaction.

Keywords: Artificial Intelligence, Machine Learning, Natural Language Processing (NLP), Chatbot, Food Website/Meal Ordering System.

Chapter 1

Introduction

Chatbots, a fusion of AI and human-like communication, redefine digital engagement by emulating natural conversations. Beyond basic responses, they excel in understanding context and user intent, enabling dynamic interactions. the adaptability allows customization for diverse purposes such as customer support, sales, information dissemination, and entertainment. Businesses leverage chatbots to automate tasks, enhance customer service, and provide round-the-clock support, elevating overall user experiences. In the ever-evolving tech landscape, chatbots stand out as versatile tools reshaping human-machine interactions.

The organization of chapter is as, the Section 1.1 describes the Background of the project. Motivation is described in the Section 1.2. The Section 1.3 describes the Problem Definition. The Scope of the project is described in the Section 1.4 .The Section 1.5 describes the Objectives of the project. The Selection of Life Cycle Model is described in the Section 1.6.

1.1 Background

Chatbots have a historical lineage dating back to the early days of computing, with pioneers like ELIZA in the mid-1960s, focused on basic text-based conversations. the early iterations operated on rudimentary keyword recognition and template-based responses, representing a foundational but limited approach. Progress in computer science and AI over the decades propelled chatbots into a new era, integrating natural language processing (NLP) and machine learning algorithms. Such evolution empowered chatbots to understand and produce human-like language, a shift hastened by the ascendancy of the internet and the ubiquitous adoption of digital platforms.Recognizing the potential, businesses embraced chatbots to enhance customer service, automate tasks, and improve user experiences on websites and social media.

As chatbot technology advances, it permeates diverse sectors like e-commerce, healthcare, finance, and customer support. The future promises even greater sophistication, with chatbots evolving into versatile virtual assistants. The persistent evolution in the field continues to blur the boundaries between human and machine communication, outlining a dynamic trajectory for the role of chatbots across various industries.

1.1.1 Study resources system and Technology

NLP encompasses a fusion of various aspects of machine learning, drawing from computer sciences, artificial intelligence, and linguistics. It primarily focuses on the intricate realm of communications and interactions between computers and human languages. Yet, the complexity of implementing the technology surpasses its initial appearance. Achieving flawless communication and understanding between users and systems demands a comprehensive grasp of all aspects involved. Given the nature of communication, even a seemingly simple word harbor multiple meanings and connotations due to the mood or feeling attached to it, influenced by the tone in which it was spoken. For instance, a word as straightforward as "what" serve various purposes and express different emotions in different scenarios[6]. The NLP toolkit is used for ambiguity disintegration between words, sentences – handling tokenization, handling POS, which is parts of speech tagging, which stands for parts of speech tag based on its context and definition. It's very useful in sense disambiguation. In seeing the word in its true content sense and the function of the word as in if it is an interjection, pronoun, conjunction, preposition or a verb, noun, adjective or adverb.

1.2 Motivation

The motivation behind AI-based chatbots stems from the desire to create automated systems which emulate human-like conversations, driven by a compelling vision of convenience and efficiency.The chatbots are designed to provide users with a seamless and accessible means of accessing information, seeking assistance, and automating tasks. the relentless 24/7 availability ensures assistance is just a message away, while scalability and cost-efficiency make them indispensable for organizations. By delivering consistent responses and personalized interactions, chatbots elevate user experiences. Furthermore, the data they generate offers valuable insights, making them a powerful tool for improving customer service and decision-making. In essence, AI-based chatbots represent a transformative force in redefining how we interact with technology, catering to the needs with unmatched efficiency and convenience.

1.3 Problem Definition

The Problem Definition gives information about the problem statement:
Problem Statement Title:AI-Powered FAQ Assistant for Instant Answers.
Organization: Artificial Intelligence, Machine Learning, Natural Language Processing.
Description: In a settled landscape, AI chatbots facilitate seamless customer interactions. Yet, a significant influx of user inquiries beyond the trained scope triggers system confusion. Consequently, customers experience inconsistencies and prolonged query resolution times, negatively impacting user experience. By deploying advanced training techniques, refining error-handling mechanisms, and expanding the chatbot's knowledge base, equilibrium is restored, and users. enjoy improved engagement and efficient issue resolution.
Complexity: Challenging.

1.4 Scope

The scope of implementing a chatbot on the food website encompasses a range of vital objectives. Firstly, it seeks to enhance the user experience by offering personalized menu assistance, streamlining the process for online orders and reservations, and delivering rapid customer support. A second key goal is to drive online sales and conversions, thus increasing revenue. To realize the objectives, the project will deliver several crucial components. the include a user-friendly chatbot interface equipped with natural language processing capabilities, and the establishment of a robust backend system for collecting and analyzing user data. In addition, the chatbot will be loaded with features such as menu exploration and recommendations, and special offers or promotions to engage and retain users.

1.5 Objectives

The objectives of the system are as follows:

- Menu Recommendation: The chatbot should offer personalized and enticing menu recommendations to enhance the user's dining experience, considering the preferences, dietary restrictions, and past orders.

- Efficient Order Placement: The chatbot aims to simplify the order placement process, guiding users through the menu, assisting with customizations, and seamlessly processing orders, thus increasing user satisfaction and order frequency.

- Order Tracking: The chatbot should provide real-time order tracking updates, allowing users to monitor the status of the orders, estimated delivery times, and ensuring transparency in the delivery process.

- Enhanced User Engagement: Encourage users to interact with the food website regularly by offering promotions, special offers, and engaging content, which fosters loyalty and repeat business.

1.6 Selection Of Life Cycle Model

The Waterfall approach was the first SDLC Model to be used widely in Software Engineering to ensure success of the project. In ”The Waterfall” approach, the whole process of software development is divided into separate phases. In a Waterfall model, typically, the outcome of one phase acts as the input for the next phase sequentially. The Figure 1.1 illustration shows the representation of different phases of Waterfall Model.

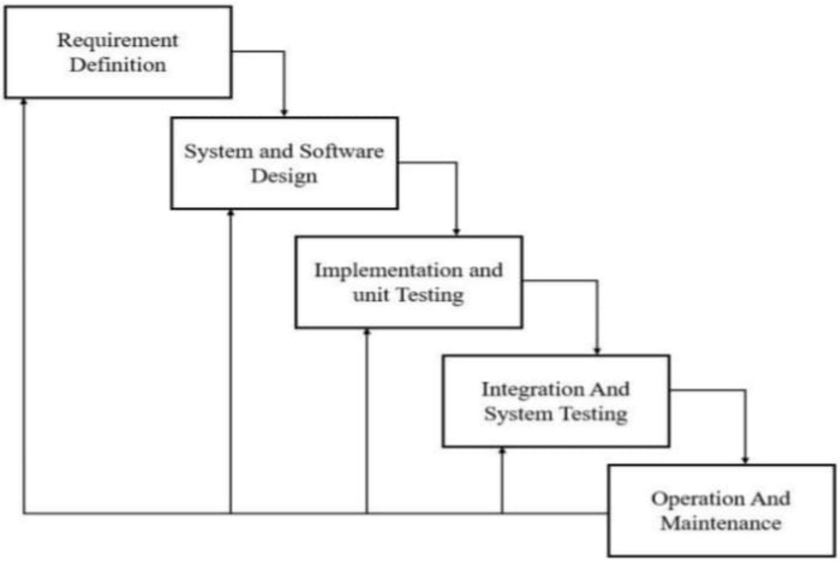
Requirement Definition
System and Software Design
Implementation and unit Testing
Integration And System Testing
Operation And Maintenance

Figure 1.1: Iterative Waterfall Model

The sequential phases in the Waterfall model are developed are captured in the phase and documented in a requirement specification document.

- The sequential phases in the Waterfall model are developed are captured in the phase and documented in a requirement specification document.

- System Design: The requirement specifications from the first phase are studied in the phase and the system design is prepared. The system design helps in specifying hardware and system requirements and helps in defining the overall system architecture.

- Implementation: With inputs from the system design, the system is first developed in small programs called units, which are integrated in the next phase. Each unit is developed and tested for its functionality, which is referred to as Unit Testing.

- Integration and Testing: All the units developed in the implementation phase are integrated into a system after testing of each unit. Post integration, the entire system is tested for any faults and failures.

- Deployment of system: Once the functional and non-functional testing is done; the product is deployed in the customer environment or released into the market.

- Maintenance: Some issues which come up in the client environment. To fix the issues, patches are released. Also, to enhance the product, some better versions are released. Maintenance is done to deliver the changes in the customer environment.

1.7 Organization of Report

Structuring the report involves a systematic approach to enhance readability and comprehension. It entails organizing the content in a manner which facilitates ease of understanding.
Chapter 1: Titled Introduction, describes Background, Motivation, Problem definition, Scope and Objective, Selection of Life cycle Model for Development.
Chapter 2: : Titled Project Planning and Management, which presents Feasibility Study, Risk Analysis, Project Scheduling, Effort Allocation, Cost Estimation.
Chapter 3: Titled Analysis, describes Requirement Collection And Identification, H/w and S/w Requirement (Data, Functional and Behavioral), Functional and non-Functional Requirements, Software Requirement Specification(SRS).
Chapter 4: Titled Design, presents System Architecture, Data Flow Diagram, UML Diagrams (Use case, Class, Sequence, Component, Deployment, State chart, Activity Diagram etc.).
Chapter 5: Titled Coding and Implementation, which presents Steps, Software and Hardware for Development in Detail, Modules in Project, Summary.
Chapter 6: Titled System Testing, Black Box Testing, White Box Testing, Manual/Automated Testing, Test Cases Identification and Execution, Summary.
Chapter 7: Titled Results and Discussion, Black Box Testing, White Box Testing, Manual/Automated Testing, Test Cases Identification and Execution, Summary.
Chapter 8: Titled Conclusion and Future Work about the Major Project.

1.8 Summary

In this Chapter, Introduction is discussed. In the next Chapter, Project Planning and Management is provided.

Chapter 2

Project Planning and Management

Project planning and management involve organizing, scheduling, and coordinating tasks to achieve specific goals within a defined timeframe and budget. It typically includes defining objectives, identifying resources, creating schedules, allocating tasks, monitoring progress, and adapting plans as needed to ensure successful completion. Effective communication, risk management, and stakeholder engagement are also crucial components of project management.

The organization of chapter is as, the Section 2.1 presents the Feasibility study. Risk Analysis is presented in the Section 2.2. The Section 2.3 presents Project Scheduling. Effort Allocation is described in the section 2.4 and the Section 2.5 describes Project Scheduling. Cost Estimation is described in the Section 2.6.

2.1 Feasibility Study

The feasibility study is carried out to test whether the proposed system is worth being implemented. A Feasibility study is a test of a system proposed regarding its workability, its impact on the organization's ability to meet user needs and effective use of resources. The key consideration involves in the feasibility study are:

- Economical feasibility
- Operational Feasibility
- Technical Feasibility

2.1.1 Economical Feasibility

The main factor in the feasibility study is the economic viability of the product. If the product is economically orderable, it will be utilized. Therefore, the project must result in cost savings. Establishing the cost-effectiveness of the proposed system is crucial.

Chatbots require minimal initial investment, operate round the clock without breaks, and handle numerous conversations simultaneously, reducing labor costs significantly. They provide consistent and accurate responses, leading to enhanced customer satisfaction and loyalty.

2.1.2 Operational Feasibility

Operational feasibility is the ability to utilize, support, and perform the necessary tasks of a system or program. It includes everyone who creates, operates or uses the system.

Regular menu updates demand a dynamic chatbot which swiftly adapt to changes, maintaining accuracy in customer interactions. The chatbot must be capable of addressing diverse customer needs.Thus, chatbot is operationally feasible.

2.1.3 Technical Feasibility

The assessment relies on an outline design of system requirements to ascertain if the company possesses the technical expertise for project completion. Technical feasibility includes evaluating the hardware, software, and other necessary technical components of the proposed system.

The portal is developed using technologies such as Python, Dialogflow and MySQL database. All the required hardware and software are easily available in the market. Hence, the system is technically feasible.
No additional hardware is necessary beyond a personal system.

2.2 Risk Analysis

Risk analysis and management are a series of steps which help a software team to understand and manage uncertainty. Many problems plague a software project. A risk embodies the possibility of encountering a problem, regardless of whether it manifests or not, . But, regardless of the outcome, it is really a good idea to identify it, assess its probability of occurrence, estimate its impact, and establish a contingency plan should the problem actually occur.

The risks incurred are:

- Misunderstandings and Miscommunications: Chatbots misinterpret user queries, leading to incorrect responses. Miscommunications, especially concerning recommendations, could lead to customer dissatisfaction or health risks.

- Technical Issues: Server downtimes, slow responses, or software glitches could disrupt user experience. Implementing regular maintenance and having backup systems mitigate risks.
- Dependency on Internet Connectivity: Both users and the business depend on a stable internet connection. Downtimes hinder the chatbot's functionality, impacting customer service and order processing.
- User Resistance: Some individuals favor traditional methods over chatbots, resulting in decreased acceptance. Mitigating such resistance accomplished by highlighting explicit advantages, educating users, and incorporating a user-friendly interface.

2.3 Project Scheduling

Software project scheduling is an activity which distributes estimated effort across the planned project duration by allocating the effort to specific software engineering tasks. It is important to note, however, which the schedule evolves over time.

In Table 2.1, a visual representation of the schedule is presented in the form of a chart. Additionally, Figure 2.1 provides an illustrative project scheduling chart, offering a visual overview of the timeline and highlighting key milestones in the project's execution.

Table 2.1: Schedule Representation Chart

Sr no.	Task	Start Date	Days of Completion
1	Selection of Project Title	03/08/2023	05
2	Gathering the information	05/08/2023	06
3	Analyze the information	14/08/2023	08
4	Discussion with Guide	22/08/2023	07
5	Made the Problem Statement	29/08/2023	02
6	Check Scope	01/09/2023	05
7	Check Feasibility	05/09/2023	07
8	Design UML Diagrams	12/09/2023	16
9	Present to Guide	29/09/2023	01
10	Coding and Implementation	15/11/2023	80
11	Testing	21/03//2023	3
12	Result	25/03/2023	1

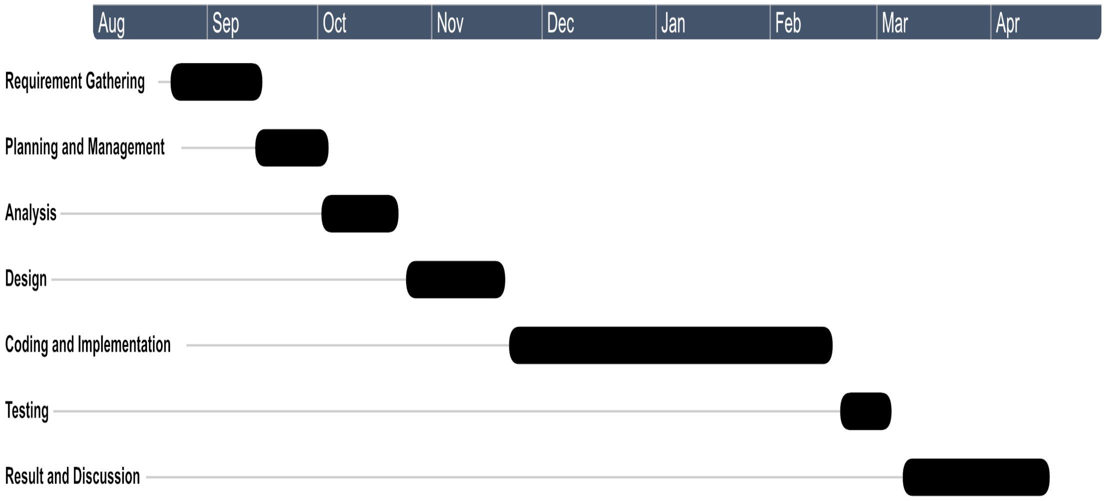

Figure 2.1: Project Scheduling Chart

2.4 Effort Allocation

Project is developed by a combination of the effort of the team, so the whole project is divided into modules and the number of modules is allotted to team members. After completion of each module, it's linked from one module to another module to form a complete project.Table 2.2 provides a comprehensive breakdown of effort allocation, detailing the distribution of resources across various tasks or activities.

Modules	**Nikita(H)**	**Kajal**	**Nikita(P)**	**Sunidhi**
Gathering of information	✓	✓	✓	✓
Planning/Requirement Analysis	✓		✓	✓
Study of Included Stack and Frameworks	✓	✓	✓	✓
Selection Of Life Cycle model		✓	✓	
Planning and Management	✓	✓	✓	✓
Analysis and Design UML	✓	✓		✓
Coding and Implementation	✓	✓	✓	✓
Testing	✓	✓	✓	✓
Result	✓	✓	✓	✓

Table 2.2: Effort Allocation

2.5 Cost Estimation

Cost estimation in project management is the process of forecasting the financial and other resources needed to complete a project within a defined scope. Cost estimation accounts for each element required for the project—from materials to labor—and calculates a total amount, which determines a project's budget.

COCOMO Model

COCOMO (Constructive Cost estimation model), proposed by Boehm [1981]. Boehm postulated any software development project will be classified into one of the subsequent categories based on the development complexity: organic, semi-detached, and embedded. In order to classify a product into identified categories, Boehm not only considered the characteristics of the product but also those of the development team and development environment.

Boehm's definition of organic, semi-detached, and embedded systems are elaborates are:

1. **Organic Mode**
   - Relatively Small, Simple Software projects.
   - Small teams with good application experience work to a set of less than rigid require- ments.
   - Similar to previously developed projects
   - Relatively small and require little innovation.
2. **Semi Detached Mode:** Intermediate (in size and complexity) software projects in which teams with mixed experience levels must meet a mix of rigid and less than rigid requirements.
3. **Embedded mode:** Software projects must be developed within set of tight hardware, software and operational constraints.

Basic COCOMO Model: Basic COCOMO Model is good for a quick, early, rough order of magnitude estimate of software cost. It does not account for differences in hardware constraints, personal Quality and experience, use of modern tools and techniques, and other project attributes known to be a significant influence on software cost, which limits its accuracy. It gives an approximate estimate of the project parameters.

The basic COCOMO estimation model is given by the expressions.

Software project	a$_b$	b$_b$	c$_b$	d$_b$
Organic	2.4	1.05	2.5	0.38
Semi-detached	3.0	1.22	2.5	0.33
Embedded	3.6	1.20	2.5	0.32

Table 2.3: Cost estimation

The basic COCOMO estimation model is given by the expressions.

$$\textbf{Effort , E} = a_b \cdot (KLOC)^{b_b} KiloLinesofCode \tag{2.1}$$

$$\textbf{Tdev ,D} = c_b \cdot (E)^{d_b} months \tag{2.2}$$

Where, Equation 2.1 shows estimation of development of effort, expressed in person months (PM). Equation 2.2 shows Tdev is the estimated time to develop the software, expressed in months, KLOC is the estimated size of the software product expressed in Kilo Lines of Code.

KLOC = 6(approx.) Kilo Lines of Code
Software Project Type = Organic

Calculation:

$$\text{Efforts, } E = 2.4 * 6^{1.05}$$
$$\text{Efforts, E = 2.4 * 6.28}$$
$$\text{Efforts, E = 15.07 PM}$$
$$\text{Tdev,}D = 2.5 * 15.07^{0.38}$$
$$\text{Tdev, D = 2.5 * 3.99}$$
$$\text{Tdev, D = 9.97 months}$$

2.6 Summary

In this Chapter, Project Planning and Management is discussed.In the next Chapter, Analysis is described.

Chapter 3

Analysis

Analysis is the detailed examination of the project or the structure of the project. Analysis of the project is the process of breaking it into multiple tasks to simplify it. As Analysis simplify the project by dividing it into different parts, it is very important process in project development. The project management tool used to look at different process and decide what to do to make it better and more effective.

The organization of the chapter is as, the Section 3.1 describes Requirement Collection and Identification. Software Requirement Specification (SRS) is described in the Section 3.2

3.1 Requirement Collection and Identification

Requirement collection is the process which is used to gather, analyze, and documentation and reviews the requirements. Requirements describe what the system will do in place of how.In the current phase, the primary goal is to gain a comprehensive understanding of stakeholders' preferences, ensuring the needs are explicitly defined, articulated, and thoroughly understood. The primary goal is to establish a solid foundation for the project by outlining a clear set of criteria and guidelines which the resulting product or solution should fulfill to meet stakeholders' objectives and satisfaction.

3.1.1 Inception

In the inception phase of developing an AI-based chatbot, the focus lies on defining the fundamental purpose and objectives of the bot. Stakeholders collaborate to establish the bot's intended role—whether it's for customer service, information dissemination, or task automation. Identifying the target audience becomes pivotal, understanding the needs, preferences, and pain points. Furthermore, decisions regarding the platform and integration requirements, including the deployment location of the chatbot and its interaction with users, are outlined in the current phase.

3.1.2 Elicitation

Elicitation involves a comprehensive gathering of requirements from various stakeholders, ranging from customer service teams to end-users.Exploring the following stage involves thoroughly understanding stakeholders' preferences, ensuring explicit definition, articulation, and a comprehensive grasp of the needs.

3.1.3 Interview

Conducting interviews becomes a key component in understanding the nuanced requirements of stakeholders and users. the interviews offer a deeper understanding of the expectations surrounding the chatbot's functionalities, user interface preferences, language tone, and the specific types of queries users anticipate the chatbot handling.Direct engagement actively contributes to shaping the direction and features of the chatbot.

3.1.4 Brainstorming Sessions

During the following stage, the emphasis lies in promoting diverse thinking and innovation to shape the bot's functionalities, conversational flow, and overall personality. Moreover, it provides a space to explore a variety of AI models and technologies suitable for the chatbot's intended purposes.

Facilitated Application Specification Techniques (FAST): It is similar to brainstorming sessions. Team oriented approach occurs.

Quality Function Deployment (QFD): It incorporates the voice of the User. When deciding User requirements, next move to the Elaboration phase.

3.1.5 Elaboration

Elaboration involves refining and organizing the collected requirements into a comprehensive document which details the chatbot's functionalities and capabilities. The phase transforms raw information into structured user stories or scenarios, defining how users will interact with the chatbot across different scenarios and contexts.

3.1.6 Negotiation

The phase for AI based chatbot development is where the Users and developers discussed on financial and other commercial issues.During the following phase, discussions among users, developers, and other fund providers revolve around ranking study resource requirements, determining project risks, and finalizing the project cost.

3.1.7 Requirements Identification

It identifies different types of software requirements such as:

- **Data Domain:** It involves pinpointing essential data domains such as menu details, order preferences, and order tracking. the specifics are vital for creating a seamless user experience and integrating the chatbot effectively with existing systems, ensuring secure, personalized interactions for customers.
- **Function Domain:** The function domain for a chatbot on a food website includes key operations such as menu browsing, order placement, payment processing, delivery tracking, and customer support interactions. Understanding the functions is crucial during requirement identification, ensuring the chatbot performs tasks seamlessly, enhancing user experience, and meeting the specific operational needs of the food website.
- **Behavioral Domain:** It encompasses user interactions and engagement patterns. It involves understanding customer preferences, conversation styles, language nuances, and feedback mechanisms.

3.2 Software Requirement Specification (SRS)

It provides requirements, needs of project and those things which help to complete project. System requirement describe a system from a technical perspective, which describe the essential characteristics of the hardware and software which will meet those needs. It should specify the capabilities, capacities and characteristics of the system in both Qualitative and Quantitative terms.

3.2.1 Model Features

The Model features of the Project are as mentioned. Feature is any measurable input used in a predictive model.

- **Menu Browsing and Item Details:** Detailed menu listings with images, descriptions, prices, and nutritional information, allowing users to browse food items efficiently.
- **Order Placement and Customization:** User-friendly interfaces for placing orders, modifying items, specifying quantities, and adding special requests or dietary preferences.

- **Order Tracking and Status Updates:** Real-time order tracking features, allowing users to monitor the status of the orders, including preparation, dispatch, and delivery status.

- **Promotions and Discounts:** Integration of promotional offers, coupon codes, and loyalty program functionalities, enhancing user experience and customer retention.

- **Customer Support and FAQs:** Instant responses to frequently asked questions (FAQs) and seamless transfer to human customer support agents for complex queries, ensuring comprehensive user assistance.

3.2.2 Operating Environment

Operating environment of the project contains software and hardware environment.

- **Software Environment:** HTML, CSS, JavaScript, Python, Libraries like Fast API, NLTK, Spacy, MYSQL, Dialogflow.

- **Hardware Environment:** Intel i3 or higher processor, 4 GB RAM, 1 TB HDD, Desktop.

3.2.3 Assumption

Assumptions is an external factor for which the intervention is not responsible, but is very important for the realization of the results, the project purpose and the overall objective. It's outside direct intervention control, but vital for achieving a successful implementation. Some assumptions are.

- Stable Internet Connectivity: Users accessing the food website and chatbot are assumed to have stable and reliable internet connections to ensure uninterrupted service.

- User Device Compatibility: Users are assumed to have compatible devices (such as smartphones, tablets, or computers) and modern web browsers to access the website and chatbot interfaces effectively.

- Basic User Proficiency: Users are assumed to have basic knowledge of online navigation and chatbot interactions, including understanding menu options and making selections.

- Accurate Menu and Pricing Information: It's assumed which the menu items, prices, descriptions, and availability displayed on the website and chatbot are accurate and regularly updated.

3.2.4 Functional Requirements

Functional requirements are specific and detailed descriptions of the functions, features, and capabilities a software system or application must have to fulfill its intended purpose. They outline the system's behavior, interactions, and operations, specifying what the software should do in response to different inputs or conditions. Functional requirements serve as the foundation for software design, development, and testing, ensuring which the final product meets user needs and expectations. The functional requirements of the project are as follows.

1. **Menu Assistance:** The chatbot must provide menu information, descriptions, and recommendations based on user preferences and dietary restrictions.

2. **Order Placement:** Users should be able to place food orders through the chatbot, specifying menu items, quantities, and customizations.

3. **Order Tracking:** The chatbot must offer real-time order tracking and providing status updates.

4. **Feedback Handling:** Collect and process user feedback and complaints, ensuring prompt responses and issue resolution.

5. **Promotions and Specials:**Promote and apply discounts, promotions, and special offers based on user profiles and order history.

3.2.5 Non-functional Requirements

Non-functional Requirement is mostly quality requirement. which stipulates how well the portal does, what it has to do. Other than functional requirements in practice, It would entail detail analysis of issues such as availability, security, usability and maintainability.
Non-functional requirements are as follows:

- Performance: The chatbot must respond promptly, with low latency, even during peak hours.

- Security: Ensure which user data, including payment information, is securely stored and transmitted. Implement measures to prevent unauthorized access.

- Scalability: The chatbot should handle a growing number of users and orders without degradation in performance.

- Reliability: Minimize downtime and ensure which the chatbot is available 24/7 to handle customer orders and inquiries.

- Usability: The chatbot interface should be user-friendly, with intuitive navigation and clear responses.
- Testing and Quality Assurance: Rigorous testing should be performed to identify and address any issues, including ensuring the chatbot understands a wide range of user inputs.

3.2.6 External Interfaces (User, Hardware, Software, Communication)

- Hardware interface
  The hardware requirement includes a system with following configurations:
  - Processor: Intel i3 processor
  - RAM: 4 GB
- Software interface
  the process is model based process. So we don't need internet connection which is required from user standpoint.
  - Backend: Python, Dialogflow
  - Frontend: HTML, CSS, JavaScript
  - Database: MySQL

3.3 Summary

In this Chapter, Analysis is discussed. In the next Chapter, Design is provided.

Chapter 4

Design

During the design phase of a project, designers gather requirements, conduct research, and conceptualize the product's design based on user needs and project goals. Creating visual interfaces, designing user experiences, and develop prototypes to test and refine the concepts. Feedback from stakeholders and users is collected and used to iterate on the designs until finalized for implementation. The phase is critical for ensuring the product meets user expectations, is intuitive to use, and aligns with the project's objectives, setting the stage for successful development and deployment.

The organization of the Chapter is as, The Section 4.1 describes the System Architecture. Data Flow Diagrams are presented in the Section 4.2. The Section 4.3 presents the UML Diagrams.

4.1 System Architecture

The System architecture refers to the overall design and organization of a complex system, which includes hardware, software, and various components interact with each other to achieve specific goals. It involves defining the system's structure, components, and interfaces, as well as the relationships between them. A good system architecture ensures the system is reliable, scalable, and maintainable, while meeting the functional and non-functional requirements. Figure.4.1 shows the architecture of the system.

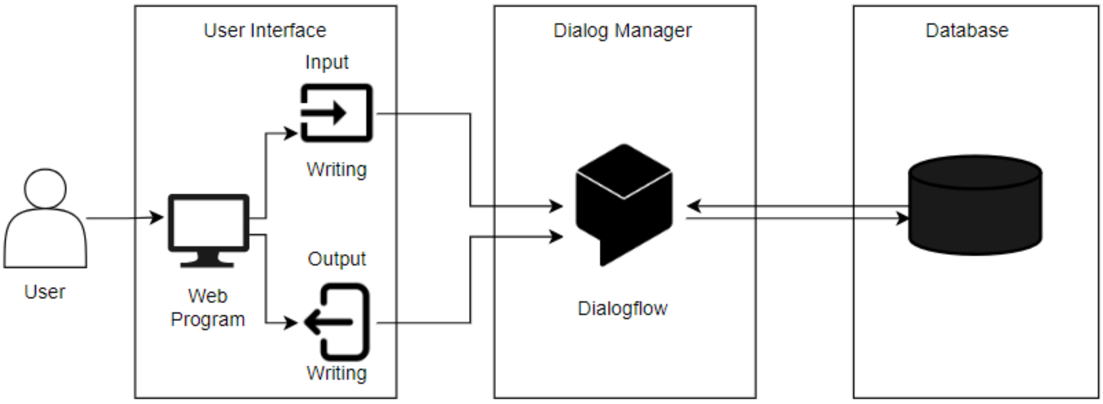

Figure 4.1: System Architecture

4.2 Data Flow Diagram

A data flow diagram (DFD) is a graphical representation of the 'flow' of data through an information system, modeling its process aspects. A DFD is often used as a preliminary step to create an overview of the system, which later be elaborated. DFDs also be used for the visualization of data processing (structured design). A DFD shows what kind of information will be input to and output from the system, where the data will come from and go to, and where the data will be stored.

4.2.1 Level 0 DFD

Level 0, also referred to as a context diagram, serves as an abstract representation of a system with a singular input and output. In the presented diagram, the system appears as a single process, visually enclosed within a bubble, demonstrating its interactions with external entities through incoming and outgoing arrows. At the foundational level, the emphasis is on presenting a comprehensive overview of the entire system's functionality, focusing on the flow of information from input (system) to output (user). Figure 4.2 enhances the concept by introducing the Data Flow Diagram (DFD) Level 0, offering a more detailed breakdown of the data processing and flow within the system.

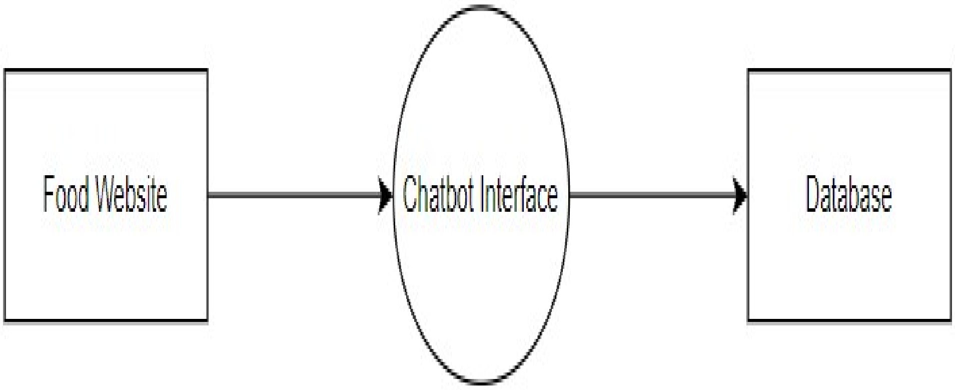

Figure 4.2: Data Flow Diagram - Level 0

4.2.2 Level 1 DFD

A Level 1 DFD, akin to an "exploded view" of the context diagram, elaborates on the main sub-processes constituting the overall system. It serves as a detailed breakdown of the Level 0 DFD, showcasing individual modules and the flow of data among them. Acting as a more specific representation, Level 1 DFD outlines basic processes and information sources, providing an intricate perspective of the Context Level Diagram. In the progression from Level 0, Figure 4.3 includes the Data Flow Diagram (DFD) of Level 1, emphasizing the system's main functions as it delves into its sub-processes.

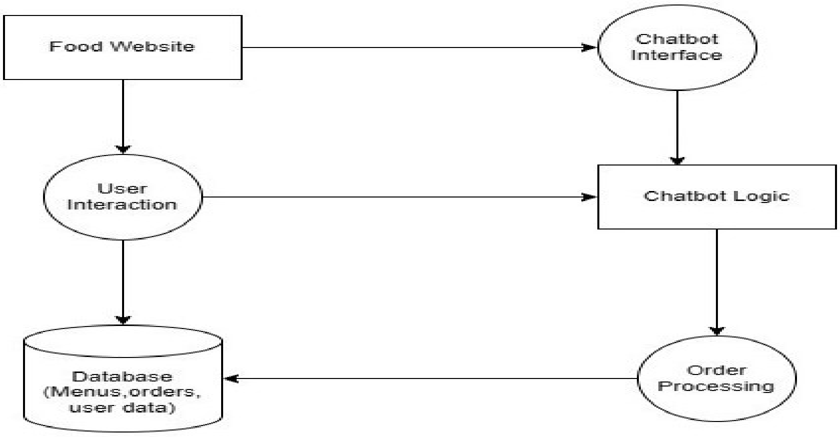

Figure 4.3: Data Flow Diagram - Level 1

4.2.3 Level 2 DFD

A Level 2 Data Flow Diagram (DFD) provides a more granular examination of the processes constituting an information system compared to a Level 1 DFD. The detailed representation is instrumental for planning and documenting the specific components of a system. By delving deeper into the intricacies of the processes, a Level 2 DFD offers a comprehensive view which aids in understanding the system's functionality. For a visual reference, please refer to Figure 4.4, which includes the Level 2 Data Flow Diagram, showcasing the nuanced details of the information system's structure and flow.

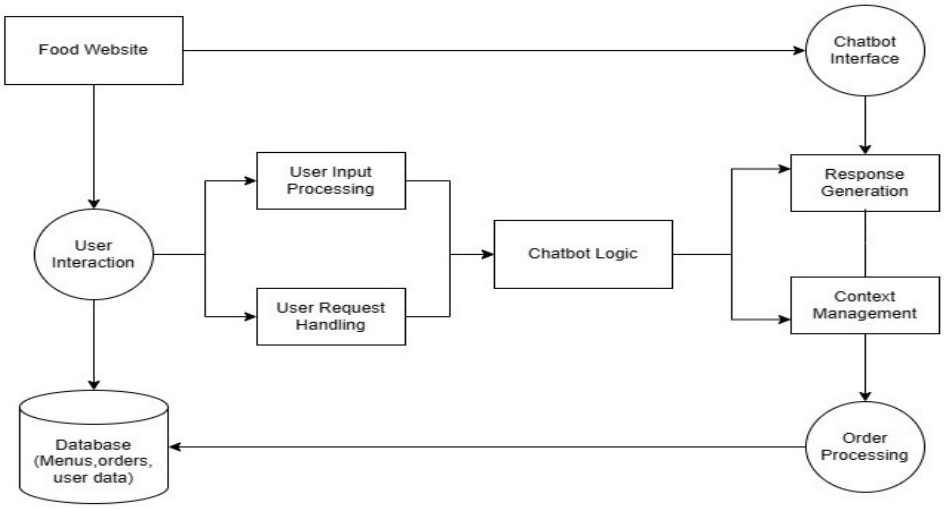

Figure 4.4: Data Flow Diagram - Level 2

4.3 UML Diagrams

The UML is a language for:

- Visualizing: The transient structures represented using the UML
- Specifying: The UML addresses the specification of all the important analysis, design, and implementation decisions which must be made in developing and deploying a software-intensive system
- Constructing: The UML is not a visual programming language, but its models directly connected to a variety of programming languages
- Documenting: The UML addresses the documentation of a system's architecture and all of its details

4.3.1 Use Case Diagram

Use case diagrams are a set of use cases, actors, and relationships. It represents the use case view of a system. A use case represents a particular functionality of a system. Hence, a use case diagram is used to describe the relationships among the functionalities and the internal/external controllers. The controllers are known as actors. The Figure 4.5 Shows the Use Case Diagram of the system.

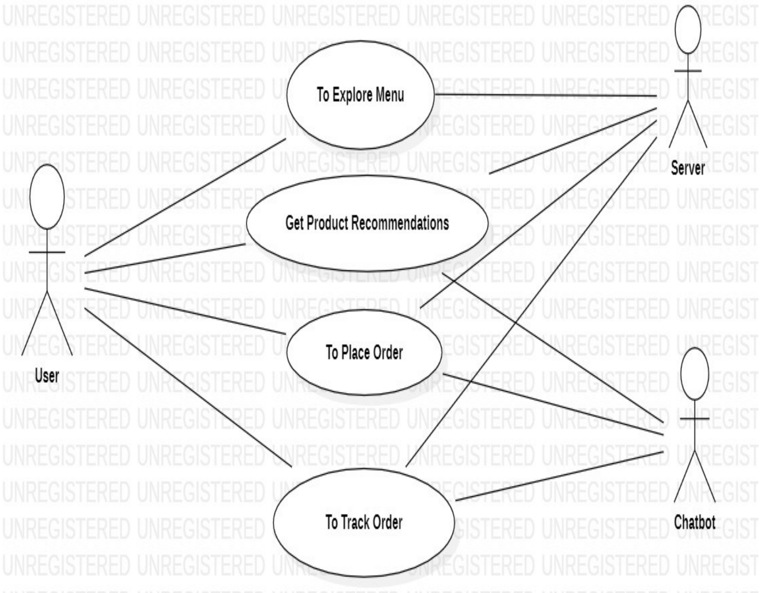

Figure 4.5: Use case Diagram for meal ordering system

4.3.2 Sequence Diagram

A sequence diagram is an interaction diagram. From the name, it is clear the diagram deals with some sequences, which are the sequence of messages flowing from one object to another. Interaction among the components of a system is very important from implementation and execution perspective. A sequence diagram is used to visualize the sequence of calls in a system to perform a specific functionality. The Figure 4.6 Shows the Sequence Diagram for use case to explore Menu in meal ordering system

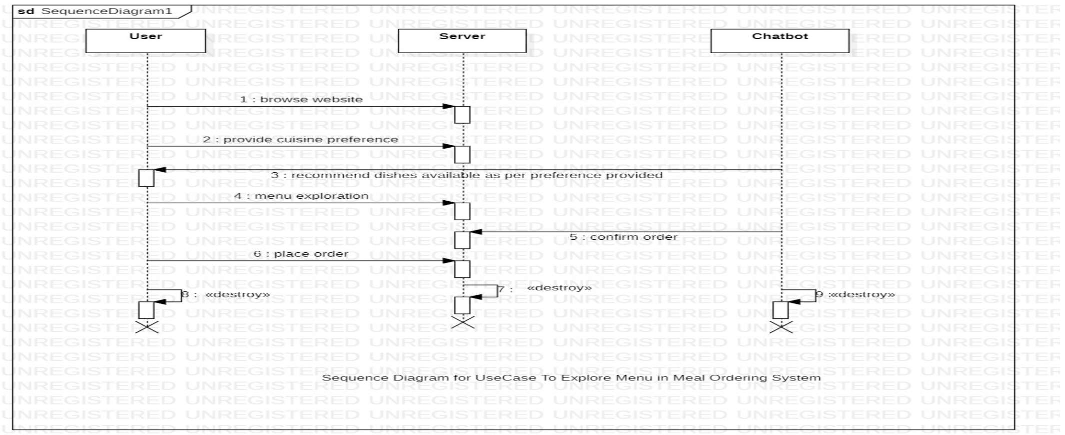
sd SequenceDiagram1
User
Server
Chatbot
1 : browse website
2 : provide cuisine preference
3 : recommend dishes available as per preference provided
4 : menu exploration
5 : confirm order
6 : place order
7 : «destroy»
8 : «destroy»
9 : «destroy»
Sequence Diagram for UseCase To Explore Menu in Meal Ordering System

Figure 4.6: Sequence Diagram 1

Figure 4.7 illustrates the sequence diagram depicting the use case for predicting recommendations.

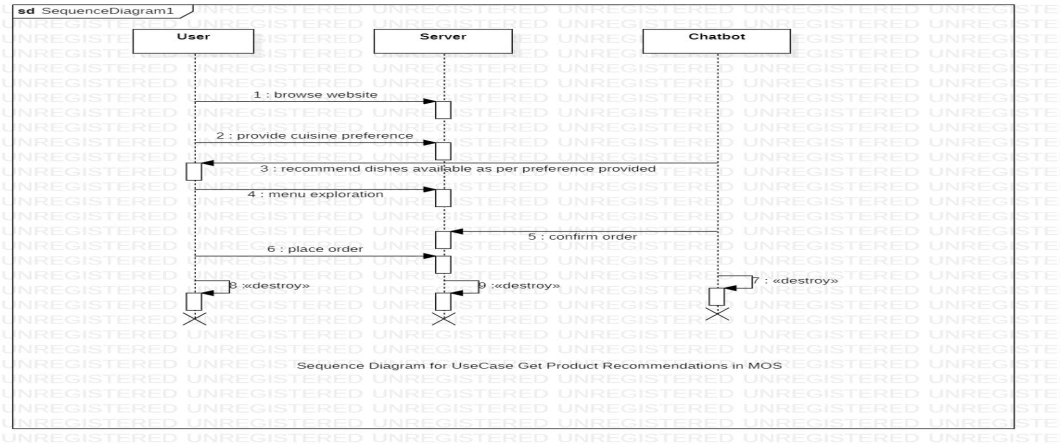
sd SequenceDiagram1
User
Server
Chatbot
1 : browse website
2 : provide cuisine preference
3 : recommend dishes available as per preference provided
4 : menu exploration
5 : confirm order
6 : place order
7 : «destroy»
8 :«destroy»
9 :«destroy»
Sequence Diagram for UseCase Get Product Recommendations in MOS

Figure 4.7: Sequence Diagram 2

Figure 4.8 presents the sequence diagram for tracking orders in the meal ordering system.

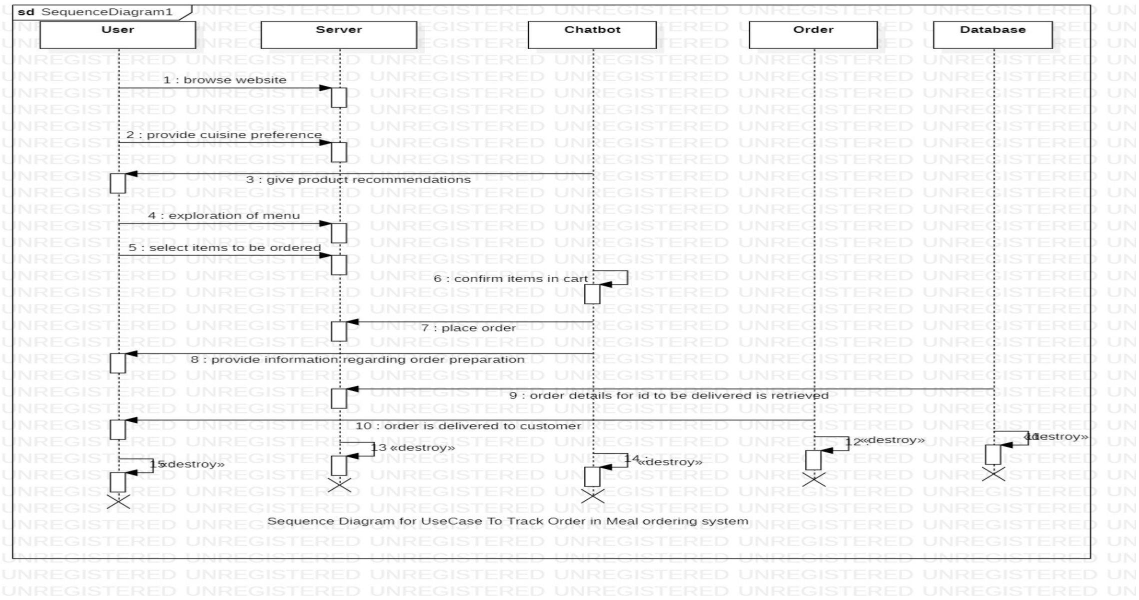
sd SequenceDiagram1
User
Server
Chatbot
Order
Database
1 : browse website
2 : provide cuisine preference
3 : give product recommendations
4 : exploration of menu
5 : select items to be ordered
6 : confirm items in cart
7 : place order
8 : provide information regarding order preparation
9 : order details for id to be delivered is retrieved
10 : order is delivered to customer
11«destroy»
12 :«destroy»
13 «destroy»
14 : «destroy»
15«destroy»
Sequence Diagram for UseCase To Track Order in Meal ordering system

Figure 4.8: Sequence Diagram 3

Figure 4.9 displays the sequence diagram illustrating the process of placing an order in the meal ordering system.

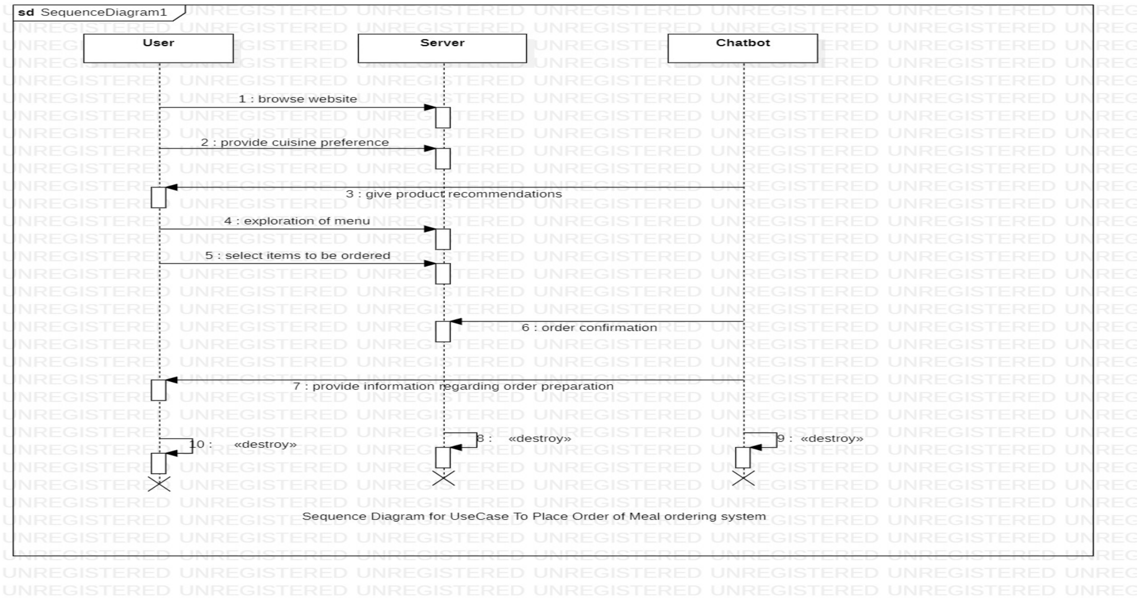
sd SequenceDiagram1
User
Server
Chatbot
1 : browse website
2 : provide cuisine preference
3 : give product recommendations
4 : exploration of menu
5 : select items to be ordered
6 : order confirmation
7 : provide information regarding order preparation
10 : «destroy»
8 : «destroy»
9 : «destroy»
Sequence Diagram for UseCase To Place Order of Meal ordering system

Figure 4.9: Sequence Diagram 4

4.3.3 Collaboration Diagram

Collaboration Diagram represents the interaction of the objects to perform the behavior of a particular use case or a part of use case. The designers use the Sequence diagram and Collaboration Diagrams to define and clarify the roles of the objects which perform a particular flow of events of a use case. Figure 4.10 depicts the collaboration diagram for the use case to explore the menu.

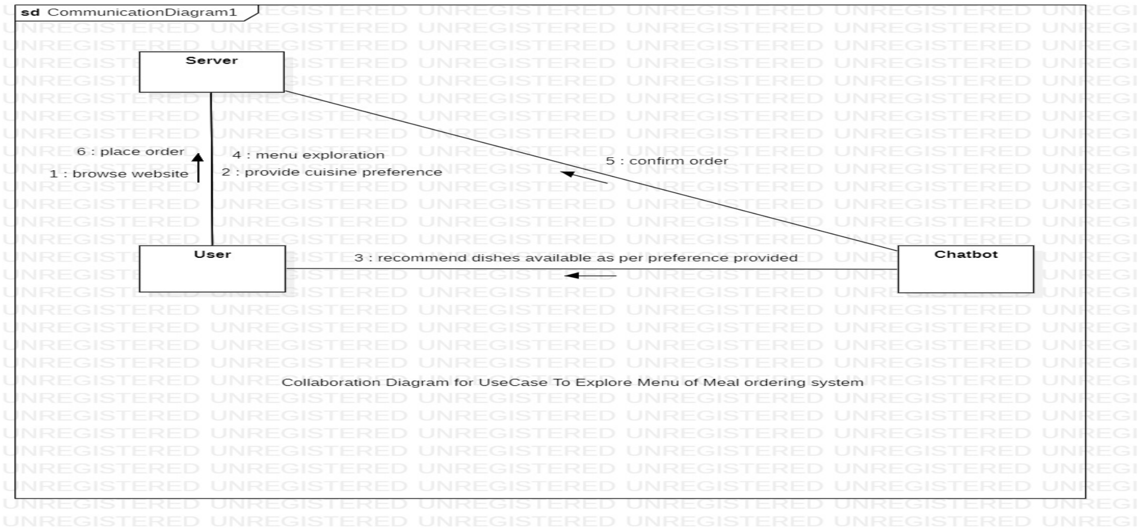
sd CommunicationDiagram1
Server
User
Chatbot
6 : place order
1 : browse website
4 : menu exploration
2 : provide cuisine preference
5 : confirm order
3 : recommend dishes available as per preference provided
Collaboration Diagram for UseCase To Explore Menu of Meal ordering system

Figure 4.10: Collaboration Diagram 1

Figure 4.11 showcases the collaboration diagram for the use case related to obtaining product recommendations.

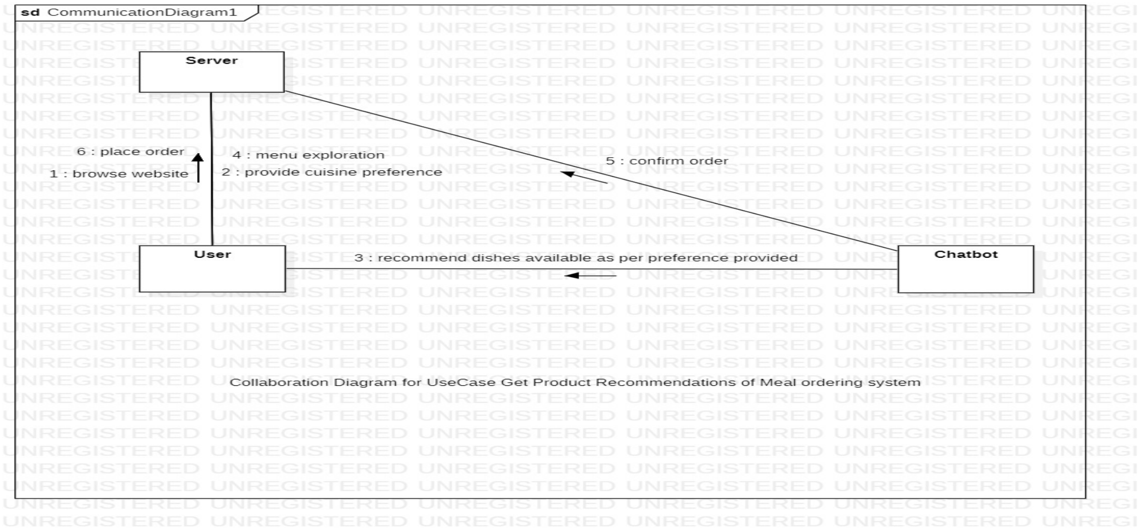
sd CommunicationDiagram1
Server
6 : place order
1 : browse website
4 : menu exploration
2 : provide cuisine preference
5 : confirm order
User
3 : recommend dishes available as per preference provided
Chatbot
Collaboration Diagram for UseCase Get Product Recommendations of Meal ordering system

Figure 4.11: Collaboration Diagram 2

Figure 4.12 illustrates the collaboration diagram for the use case focused on tracking orders in the meal ordering system.

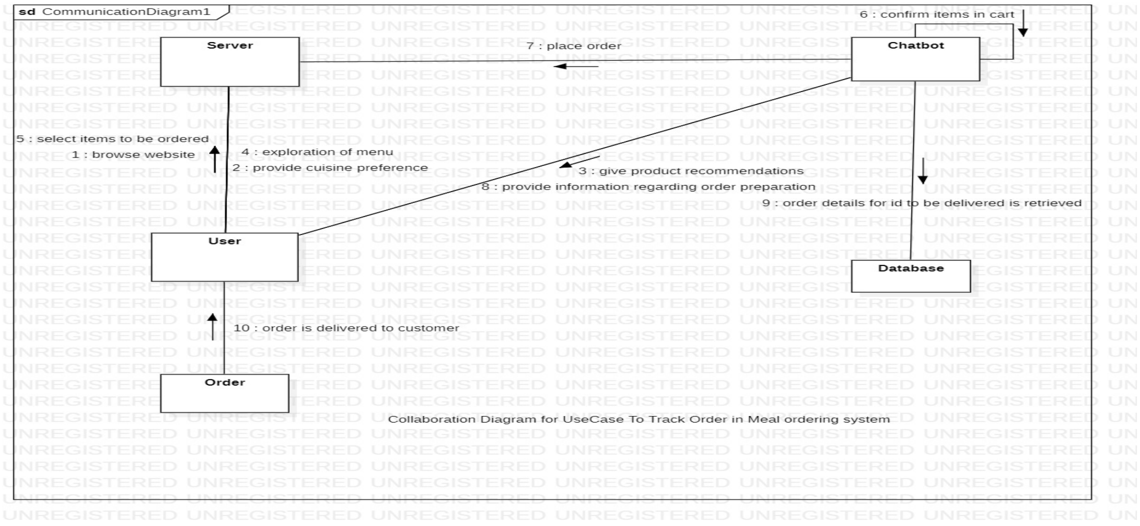
sd CommunicationDiagram1
6 : confirm items in cart
Server
7 : place order
Chatbot
5 : select items to be ordered
1 : browse website
4 : exploration of menu
2 : provide cuisine preference
3 : give product recommendations
8 : provide information regarding order preparation
9 : order details for id to be delivered is retrieved
User
Database
10 : order is delivered to customer
Order
Collaboration Diagram for UseCase To Track Order in Meal ordering system

Figure 4.12: Collaboration Diagram 3

Figure 4.13 displays the collaboration diagram for the use case involving placing an order.

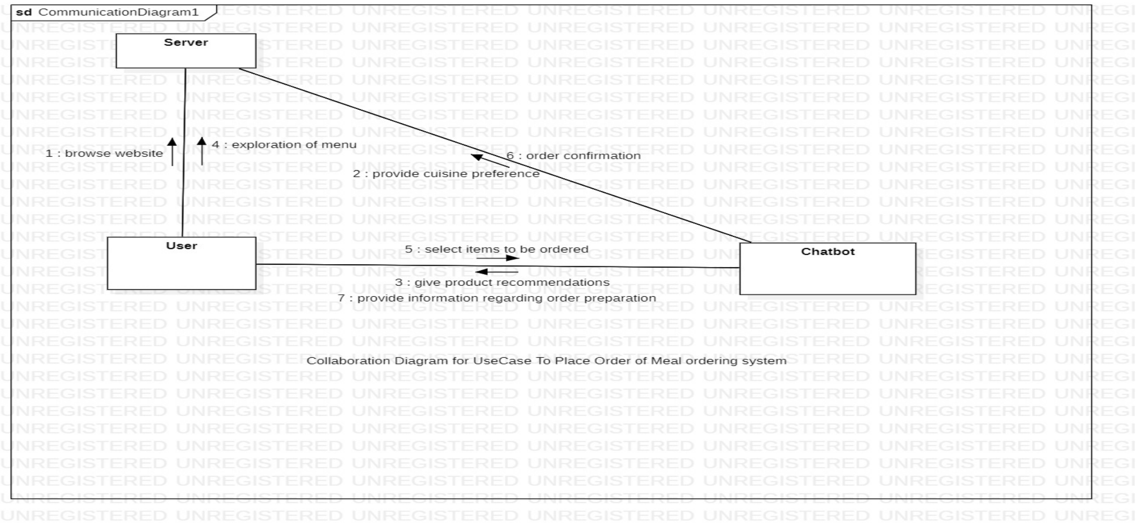
sd CommunicationDiagram1
Server
User
Chatbot
1 : browse website
4 : exploration of menu
6 : order confirmation
2 : provide cuisine preference
5 : select items to be ordered
3 : give product recommendations
7 : provide information regarding order preparation
Collaboration Diagram for UseCase To Place Order of Meal ordering system

Figure 4.13: Collaboration Diagram 4

4.3.4 Class Diagram

Class diagrams are the most common diagrams used in UML. Class diagram consists of classes, interfaces, associations, and collaboration. Class diagrams basically represent the object-oriented view of a system, which is static in nature. A class diagram represents the object orientation of a system. Hence, it is generally used for development purposes. The Figure 4.14 Shows the Class Diagram of the system.

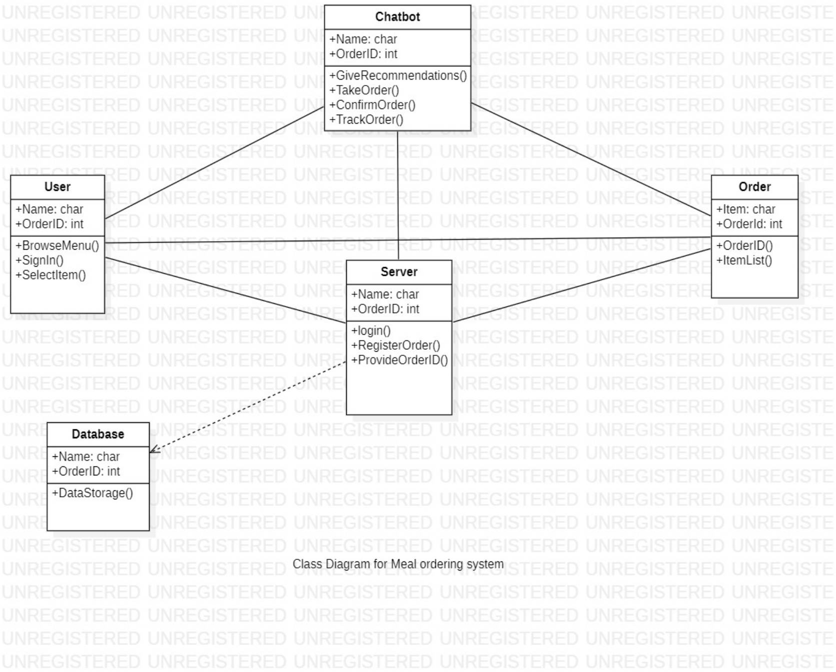

Figure 4.14: Class Diagram

4.3.5 Component Diagram

Component diagram represent a set of components and relationships. The components consist of classes, interfaces, or collaborations. Component diagrams represent the implementation view of a system. During the design phase, software artifacts (classes, interfaces, etc.) of a system are arranged in different groups depending upon the relationship. It is said component diagrams are used to visualize the implementation. The Figure 4.15 shows the representation of the Component Diagram.

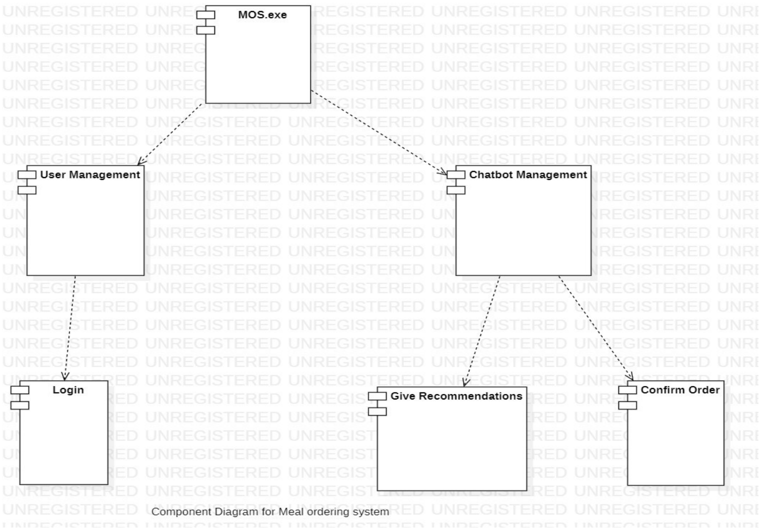

Figure 4.15: Component Diagram

4.3.6 Deployment Diagram

Deployment diagram are a set of nodes and the relationships. The nodes are physical entities where the components are deployed. Deployment diagrams are used for visualizing the deployment view of a system. It's generally used by the deployment team. The Figure 4.16 is the representation of Deployment Diagram.

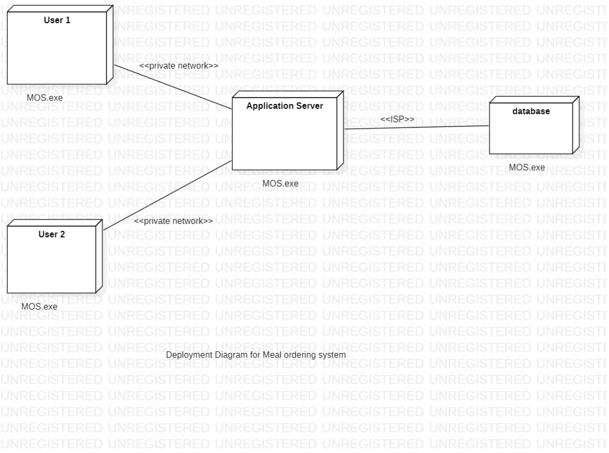

Figure 4.16: Deployment Diagram

4.4 Summary

In this Chapter, Design is discussed. In the next Chapter, Coding and Implementation is provided.

Chapter 5

Coding and Implementation

Coding and implementation are the stages where the project's blueprint is transformed into a functional reality. The phase involves writing the actual lines of code which make up the software or system being developed and putting them into action.Implementation goes hand in hand with coding, as it involves putting the code into action within the project environment. It includes tasks such as setting up databases, configuring servers, integrating with other systems, and deploying the software to production environments.

The organization of the Chapter is as, the Section 5.1 describes Algorithm/Steps. Requirements Collection and Identification are described in the section 5.2. The section 5.3 describes the Hardware and Software Requirements. Modules in Project are described in the Section 5.4

5.1 Required Software and Hardware for Development

The section details the software and hardware requirements necessary for developing and using the AI Powered FAQ assistant for instant answers.

5.1.1 Hardware Requirements

The following hardware assets are required for development and use of the AI Powered FAQ assistant for instant answers.

- A computer machine
- Intel i3 processor
- RAM 4GB
- Active and responsive server for managing data.
- Minimum Graphics Setup In Machine for Visualization Purpose.

5.1.2 Software Requirements

Software requirements are essential for the development and execution of the Chatbot application. The requirements serve as a guide for developers and users alike and provide insight into the system's nature and environment. The following software is required:

- Operating System: Windows operating system.
- The development of chatbot requires the use of Dialogflow.
- Backend development: Python.
- Frontend development: HTML, CSS and Javascript.
- Database : MySQL.

5.2 Modules in Project

The overall functionality of the chatbot is divided into 3 modules, which are detailed:

- **User interface of the Application:** The user interface of the application offers a seamless experience by integrating a visually appealing frontend displaying the menu of food items. Users effortlessly browse through the menu while also accessing the integrated chatbot for instant assistance with any queries. The interface prioritizes ease of use and responsiveness, ensuring a smooth navigation experience. With intuitive design and convenient access to information, users make informed decisions about food choices. Overall, user interface aims to provide a user-friendly platform where users explore menu and receive prompt assistance whenever needed, optimizing the experience with the application.
- **ML Model:**The Chatbot Model module is tasked with conducting conversation analysis, training, and response generation using natural language processing (NLP) techniques within the Dialogue flow platform. The module encompasses data preprocessing and intent identification to prepare conversational data for training. Additionally, it involves training and evaluating different dialogue models to select the most effective one. The Chatbot Model generates responses are seamlessly integrated into the user interface module. The primary objective is to provide accurate and contextually relevant answers to user queries, enhancing the conversational experience and ensuring efficient interaction with users.

- **Web sever and other services:**The Web Server and Other Services module manage the overall functioning of the application, specifically leveraging Fast API technology in context. The module comprises hosting the application on a Fast API-powered web server, storing and accessing data efficiently in a database, and facilitating communication with external services or data sources through APIs. Additionally, it implements robust security features such as user authentication and authorization to safeguard sensitive information. Deployment tools automate the process of deploying updates and ensuring smooth operation. The module is instrumental in delivering a seamless user experience and optimizing the performance and reliability of the application.

5.3 Summary

In this Chapter Coding and Implementation is discussed. In the next Chapter, Testing is provided.

Chapter 6

Testing

Errors and bugs do occur because of various factors like error in design of system, implementing the design of system, human error and much more, due to which testing becomes indispensable as application to be error free, for customer satisfaction and to develop the good product in market.

The organization of the Chapter is as, the first Section 6.1 describes the testing methodology of White Box Testing. Testing methodology of Black Box Testing is described in the Section 6.2. The section 6.3 presents Manual Testing methodology. Test Cases Identification and Execution is decribed in the Section 6.4.

6.1 White Box Testing

White Box Testing is testing of a software solution's internal structure, design, and coding. In such type of testing, the code is visible to the tester. It focuses primarily on verifying the flow of inputs and outputs through the application, improving design and usability, and strengthening security. White box testing is also known as Clear Box testing, Open Box testing, Structural testing, Transparent Box testing, Code-Based testing, and Glass Box testing.

- Internal security holes
- Broken or poorly structured paths in the coding processes
- The flow of specific inputs through the code
- Expected output
- The functionality of conditional loops
- Testing of each statement, object, and function on an individual basis.

After completion of testing it was found:

- No internal security holes detected during testing in Model/Code.
- No broken or poorly structured paths found in the coding processes.
- The flow of specific inputs through the code is working seamlessly.
- For Every process, Application is generating expected output.
- Testing of each statement, object, and function on an individual basis done successfully with expected outputs.

6.2 Black Box Testing

The goal of the technique is to find issues in the following categories,

- Incorrectly implemented Dialogflow, causing vulnerabilities.
- Errors in analysis of dataset and ML Model.
- Errors in data structures.
- Behavioral issues and Visualization issues.

After the Successful Testing it was found the:

- No incorrectly implemented Dialogflow causing vulnerabilities.
- ML model is correctly trained and gives a correct output.
- No errors found in data structures.
- Application no behavioral issues as well as No Visualization issues.

6.3 Manual Testing

Manual testing should never be avoided entirely as it is a continuous process requires human verification at regular intervals throughout the software development lifecycle. As a result, it's essential for teams to find the right balance between manual and automated tests.

6.4 Test Cases Identification and Execution

Test case is the set of inputs along with the output and some additional information like

- Test Case for analyzing data correctly
- Test Case of training and testing phase of output are same.

Table 6.1 shows the test cases with result.

Table 6.1: Test Cases with Result

Test	Test Cases	Expected Output	Actual Output	Result
1	Verifying the chatbot recommends food items based on user preferences.	The chatbot suggests spicy food options like Qasadia or Samosa.	Chatbot suggests food items based on user preferences successfully. .	Pass
2	If verifying the chatbot successfully processes a new order and provides an order ID.	The chatbot acknowledges the order and provides a unique order ID for tracking purposes.	The chatbot acknowledges the order and provides a unique order ID for tracking purposes.	Pass
3	Verifying the chatbot provides accurate tracking details for an order.	The chatbot retrieves the current status of the order (e.g., "Order received" or "In process").	The chatbot successfully retrieves and presents the current status of the order, ensuring transparency and convenience for users.	Pass
4	Verifying the chatbot handles invalid input gracefully.	The chatbot responds with an informative message, guiding the user on how to proceed.	The chatbot effectively communicates informative guidance to the user, facilitating smooth interaction and navigation.	Pass
5	Verifying the chatbot maintains responsiveness under different load conditions.	The chatbot responds promptly to user queries and orders without significant delays.	The chatbot responds promptly, ensuring efficient communication.	Pass

6.5 Summary

In this Chapter, Testing is described. In the next Chapter, Result is described.

Chapter 7

Result and Discussion

Results refer to the phase where the outcomes of the project are analyzed and interpreted. It involves comparing the actual results with the initial goals and objectives of the project, discussing any deviations or unexpected findings, identifying lessons learned, and determining future actions or improvements. It's an important step in evaluating the success of the project and informing decision-making for future endeavors.

The organization of the Chapter is as, the Section 7.1 represents the Result of the Project and the Section 7.2 represents Summary.

7.1 Result

In today's fast-paced digital world, where convenience is key, focusing on user-friendly solutions is crucial, particularly within the food industry. The project aims to redefine the way people experience food by introducing an innovative chatbot-based platform seamlessly integrated into an attractive and intuitive frontend interface.

The primary objective is to simplify the process of interacting with food services, providing users with a comprehensive solution to anticipates and meets the needs with exceptional ease. Whether it's browsing menus, receiving tailored recommendations, placing orders, or tracking the order progress, the platform strives to offer a complete and hassle-free experience, prioritizes user satisfaction over anything else.

Utilizing cutting-edge technology and thoughtful design principles, the platform not only streamlines the user journey but also enhances it, creating an engaging and interactive environment where users feel empowered and delighted at every step. Ultimately, the success of project hinges on its ability to seamlessly merge advanced technology with user-centric design, setting a new standard for digital culinary experiences.

The table 7.1 shows the Features and Results of the chatbot.

Table 7.1: Features and Results

Sr no.	Features	Result
1	To explore menu options	Present
2	Get product recommendations	Present
3	To place orders	Present
4	To track orders	Present
5	Small talk engagement	Present
6	Product recommendations based on menu categories	Present

7.2 Summary

In this chapter, Result is described. In the next chapter, Conclusion is presented

Chapter 8

Conclusion

8.1 Conclusion

Implementing a chatbot on a food website has transformative advantages, elevating user engagement and satisfaction. The chatbot's ability to deliver instant responses on menu options, order status, and general queries streamlines the user experience, simplifying decision-making with personalized recommendations based on preferences and past orders. This fosters customer retention by promoting convenience and efficiency, encouraging repeat visits. The 24/7 availability of immediate assistance enhances the overall user experience.

Collaborations for seamless ordering and tracking, coupled with ongoing feedback analysis, promise a chatbot continually evolves to meet user preferences, ensuring an ever-enhanced user journey.

Bibliography

[1] Wu, Eric Hsiao-Kuang, Chun-Han Lin, Yu-Yen Ou, Chen-Zhong Liu, Wei-Kai Wang, and Chi-Yun Chao. ”Advantages and constraints of a hybrid model K-12 E-Learning assistant chatbot.” Ieee Access 8 (2020): 77788-77801.

[2] Miklosik, Andrej, Nina Evans, and Athar Mahmood Ahmed Qureshi. ”The use of chatbots in digital business transformation: A systematic literature review.” IEEE Access 9 (2021): 106530-106539.

[3] Carlander-Reuterfelt, Daniel, Álvaro Carrera, Carlos A. Iglesias, Óscar Araque, Juan Fernando Sánchez Rada, and Sergio Muñoz. ”JAICOB: A data science chatbot.” IEEE Access 8 (2020): 180672-180680.

[4] Tamrakar, Rohit, and Niraj Wani. ”Design and development of CHATBOT: A review.” In Proceedings of International Conference On“Latest Trends in Civil, Mechanical and Electrical Engineering”. https://www. researchgate. net/publication/351228837. 2021.

[5] Sumit, Raj. ”Building chatbots with python.” Using Natural Language Processing and Machine Learning (2018).

[6] Nidhi Singh Kushwaha1, Pawan Singh.”Artificial Intelligence based Chatbot: A Case Study.” Journal of Management and Service Science, 2022, Vol. 02, Iss. 01, S. No. 004.

Index