CAMPUSGUIDE: AI DRIVEN COLLEGE CHATBOT

A PROJECT REPORT

submitted by

MUHAMMED AMAN TK
(KMC21CS031)

to

The APJ Abdul Kalam Technological University

In partial fulfillment of the requirements for the award of the Degree

Of

Bachelor of Technology
In
Computer Science and Engineering

Department of Computer Science and Engineering

KMCT COLLEGE OF ENGINEERING
KOZHIKODE
NOVEMBER 2024

DECLARATION

I undersigned hereby declare that the project report **“CAMPUSGUIDE: AI DRIVEN COLLEGE CHATBOT”**, submitted for partial fulfillment of requirements for the award of degree of Bachelor of Technology of the APJ Abdul Kalam Technological University, Kerala is a bonafide work done by me under supervision of **Ms.NAJIYA NASRIN K, Assistant Professor, Computer Science and Engineering, KMCT College of Engineering.** This submission represents my ideas in my own words and where ideas or words of others have been included, I have adequately and accurately cited and referenced the original sources. I also declare that I have adhered to ethics of academic honesty and integrity and have not misrepresented or fabricated any data or idea or fact or source in our submission. I understand that any violation of the above will be a cause for disciplinary action by the institute and/or the University and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been obtained. This report has not been previously formed the basis for the award of any degree, diploma or similar title of any other University.

Place: Kallanthode
Date: 07-11-24

MUHAMMED AMAN TK

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING KMCT COLLEGE OF ENGINEERING, KOZHIKODE

CERTIFICATE

This is to certify that the report entitled **“CAMPUSGUIDE: AI DRIVEN COLLEGE CHATBOT”** Submitted by **MUHAMMED AMAN TK (KMC21CS031)** to the APJ Abdul Kalam Technological University in partial fulfillment of the requirements for the award of the Degree of Bachelor of Technology in Computer science and engineering is a bonafide record of the project work carried out by them under our guidance and supervision. This report in any form has not been submitted to any other University or Institute for any purpose.

Project Guide	Project Coordinator	Head of The Department
Mrs.NAJIYA NASRIN K	**Mrs.NISHIDHA T**	**Dr.SREEKESH NAMBOODIRI**
Assistant Professor	Assistant Professor	Associate Professor
Dept of CSE	Dept of CSE	Dept of CSE

ACKNOWLEDGEMENT

Firstly, I would like to express my sincere gratitude to my principal **Dr. Sabiq P V** for providing such quality facilities in my esteemed institution for my project work.

I would like to express my sincere gratitude to **Dr. Sreekesh Namboodiri T, Associate Professor and Head of the Department, Computer Science and Engineering**, **KMCT College of Engineering** for being a great force behind all my efforts through timely advice, guidance, and words of encouragement.

I extend my sincere gratitude to my internal guide **Ms. Najiya Nasrin K, Assistant Professor, Computer Science and Engineering, KMCT College of Engineering** for all kinds of cooperation, guidance and help rendered throughout my seminar. And I also express my thanks to all teaching and nonteaching staff for their valuable advice and help to move forward with the project work.

I express my heartfelt gratitude to the project coordinator, **Ms. Nishida T, Assistant Professor, Computer Science and Engineering, KMCT College of Engineering** for her valuable direction and constant motivation which helped in doing the project with utmost dedication and presence of mind.

Above all, I thank the Almighty without whose grace my endeavors would not have been a success.

MUHAMMED AMAN TK

ABSTRACT

This report presents the development and functionality of "CampusGuide," an AI-powered chatbot designed to streamline access to academic information for students at KTU. By leveraging Natural Language Processing (NLP) and Artificial Intelligence (AI), CampusGuide enables users to instantly retrieve essential academic details, such as attendance records and exam scores, directly through a mobile-friendly application. This chatbot addresses the inefficiencies of traditional website navigation, which often requires students to browse multiple sites and repeatedly log in, thereby enhancing user experience and operational efficiency.

CampusGuide's responses are powered by the Rasa framework, an open-source conversational AI toolkit that ensures accurate and contextually relevant interactions. With Reinforcement Learning (RL)-based Rasa Core for managing conversation flow and Recurrent Neural Networks (RNN) for intent recognition, CampusGuide accurately interprets various query formats, providing precise, tailored responses. Student data is securely managed through a JSON database, with personalization achieved through structured data handling.

CampusGuide's user-friendly design includes integration with existing campus systems, ensuring accessibility and intuitive use across devices. The chatbot is available 24/7, minimizing the need for manual data retrieval and reducing administrative workload. This innovative system not only enhances the efficiency of academic data access but also establishes a scalable model for AI integration in educational support. Future developments could include enhanced campus navigation and broader system interoperability, setting new standards for academic information accessibility and campus connectivity.

CONTENTS

LIST OF FIGURES

CHAPTER 1
INTRODUCTION

1.1 General Background

In today's digital age, students increasingly rely on quick and easy access to academic information, yet many university systems remain fragmented and challenging to navigate. Traditional methods of obtaining academic data—such as scores, attendance records, and course information—often involve navigating multiple websites, repeated logins, and long wait times, which can frustrate students and disrupt their productivity. The CampusGuide project addresses these inefficiencies by introducing an AI-driven chatbot designed to serve as a centralized, responsive resource for students' academic needs.

CampusGuide combines advanced AI and NLP technologies to provide students with immediate access to critical information without the need to browse multiple platforms. By integrating with campus systems and delivering 24/7 availability, CampusGuide aims to enhance user experience and streamline academic data retrieval. This chatbot offers a single, accessible platform that caters to a wide range of student queries, from checking attendance to viewing exam results, freeing up administrative resources and allowing students to focus on their learning.

Beyond convenience, CampusGuide's AI-powered responses are personalized, adapting to the specific needs of each student by recognizing individual user intents and providing accurate, contextually relevant answers. The chatbot's ability to understand and respond to varied queries helps foster an inclusive and efficient academic environment, empowering students to manage their academic responsibilities with greater ease. In doing so, CampusGuide reflects a broader commitment to using AI in education, demonstrating how technology can break down traditional barriers to information access and support student success in meaningful ways.

1.2 Existing System

Academic information systems today primarily depend on university websites and portals, requiring students to navigate numerous pages and log in multiple times. While these methods allow students to view essential information, the complex navigation and lack of centralization hinder the user experience, especially when students need quick access. Additionally, these systems typically rely on an internet connection and web-only interfaces, limiting accessibility for students who need mobile or offline access. For more specific or urgent queries, students often depend on administrative support, but this assistance can be delayed due to high staff workloads.

Another limitation is the inconsistency in user interfaces across different platforms, which can make it difficult for students to find information efficiently, especially if the interface is outdated or poorly organized. Furthermore, existing systems generally lack personalization, offering a one-size-fits-all approach that forces students to sift through irrelevant information to find what they need. While some schools have attempted to implement automated FAQ systems, these often fall short in providing real-time, contextually relevant responses, especially during peak periods like exams.

These challenges highlight the need for an AI-driven chatbot like CampusGuide, which provides students with a responsive, personalized, and centralized system for accessing academic data, improving both accessibility and user experience.

1.2.1 Disadvantages Of Existing System

- Complex Navigation and Multiple Logins
- Minimal Real-Time Support and Interaction
- Dependency on Human Assistance
- Slow Response Times and High Administrative Burden

1.3 Objective

The core objective of CampusGuide is to transform access to academic information for students, delivering a centralized, efficient solution for retrieving scores, attendance, and other essential data. Current university websites and portals, while useful, often involve complex navigation and fragmented information access, resulting in a time-consuming process for students. By leveraging advanced AI technology, CampusGuide simplifies the data retrieval experience, providing instant, accurate responses through Natural Language Processing (NLP) and real-time updates, thereby empowering students to access academic information without hassle.

CampusGuide also aims to enhance communication within the campus by reducing the dependency on administrative staff for routine inquiries. The chatbot offers an autonomous, user-friendly platform that answers common questions and resolves issues, streamlining the workload for campus administrators. This design fosters a more responsive academic environment, ensuring students have the resources they need without delays.

A critical goal of the system is to increase accessibility by offering 24/7 availability and mobile integration, enabling students to interact with CampusGuide anytime and from any device. This round-the-clock access ensures that students can find the information they need, whether during peak times like exams or when they cannot access a computer. By making academic resources available at any hour, CampusGuide strives to support student productivity and engagement throughout their academic journey.

1.4 Scope

A key component of CampusGuide's scope is its capacity to improve academic accessibility and foster campus engagement. By offering a centralized platform for retrieving academic information, CampusGuide enables students to independently access essential data, such as scores, attendance, and announcements, without navigating multiple websites or relying heavily on administrative staff. This streamlined access not only promotes student self-sufficiency but also strengthens the campus community by facilitating more efficient interactions. Additionally, by reducing the routine workload on administrative personnel, CampusGuide allows staff to focus on complex, value-adding tasks, creating a more responsive and connected academic environment.

From a technological perspective, CampusGuide is designed for scalability and flexibility. Built using the Rasa framework and Python, it is compatible across mobile and desktop platforms, with the ability to incorporate future AI advancements, such as voice recognition and multilingual support, broadening its accessibility to a diverse student base. Its cloud-based architecture also enables it to handle growing data and user demands, ensuring that CampusGuide can adapt to the evolving needs of the university.

Ultimately, CampusGuide envisions a future where accessing academic information is intuitive and seamless. With potential partnerships with other universities or educational technology providers, CampusGuide could set a benchmark for AI-driven academic support. By blending innovative technology with a student-centric approach, CampusGuide seeks to redefine information accessibility within educational institutions, making academic support more comprehensive and accessible for all.

CHAPTER 2

LITERATURE SURVEY

2.1 Designing an Interactive Chatbot for Educational Assistance using the Rasa Framework

Sujata Deshmukh, Xenus Gonsalves, 2023

This paper explores the development of an educational chatbot built using the Rasa framework to assist users with navigating basic queries related to institutional information, such as admission cut-offs and program details. Designed to serve students, teachers, and parents, the chatbot aims to reduce the need for manual intervention in answering routine questions. The authors highlight Rasa's ability to handle conversational flow and intent recognition effectively, which allows the chatbot to answer frequently asked questions, streamlining the information retrieval process on educational websites. However, the study identifies limitations, as the chatbot's main focus is on general inquiries, which does not fulfill the specific academic needs of individual students, such as accessing personalized data on grades or attendance.

The findings of this study demonstrate the potential of Rasa as an educational tool but highlight a gap in its capacity for student-specific academic data retrieval. The paper underlines the importance of advanced AI-driven chatbots for educational institutions, suggesting that future implementations should consider personalization to enhance user experience. This limitation in handling individual academic queries underscores the need for specialized systems, like CampusGuide, that cater specifically to personalized student information needs while maintaining real-time interaction capabilities.

2.2 Smart Chatbot for College Information Enquiry Using Deep Neural Network

Channabasamma, Lakshmi Soumya P, Nooka Swetha, Mukkari Indu, 2023

This research presents a chatbot that employs deep neural networks to handle general inquiries within a college setting, designed to assist students in finding essential information about the institution. The chatbot provides answers to FAQs and aids in campus navigation, helping students and visitors with directions and general college-related questions. By leveraging the neural network's capacity for improved accuracy in recognizing questions, the chatbot delivers contextually appropriate responses, enhancing the flow of information. However, the system is limited in scope, as it focuses only on general queries and lacks the capability to retrieve personalized academic data, such as attendance records or exam scores.

The limitations identified in this study highlight the gap between general information chatbots and the need for academic-focused systems that can provide individualized responses. While the chatbot successfully addresses campus navigation and basic inquiries, it does not offer the depth of service required for personalized academic assistance. This gap reveals the potential for solutions like CampusGuide, which are designed to meet specific academic needs by offering real-time, personalized data access to students, setting it apart from general inquiry systems.

2.3 VOID – An Assistant (Chatbot) for Education Query System.

Debaissh Chakraborty, Anjana Mishra, Vikash Kumar, Saloni Singh, Harshita Hani, 2022

VOID is an educational chatbot developed to assist with general information queries during high-demand periods, such as admissions. This chatbot is geared towards reducing the administrative workload by addressing basic education-related questions, allowing staff to focus on more complex tasks. The chatbot's primary function is to support potential students and other stakeholders with general queries about the institution, rather than providing personalized data for individual students. This functional design makes it suitable for handling routine questions but restricts its usability for students seeking specific academic information, such as grades or attendance.

The study on VOID underscores the utility of chatbots in minimizing repetitive tasks and highlights the efficiency gains in administrative support that can be achieved through automation. However, it also notes the limitations of general-purpose chatbots for ongoing, personalized student support. The lack of focus on individualized academic assistance in the VOID system showcases a gap that systems like CampusGuide aim to fill, as CampusGuide is specifically designed to provide tailored responses and access to academic data, thus enhancing the student experience with direct, real-time information retrieval.

2.4. Information Chatbot for College Management System Using Multinomial Naive Bayes

Guttikonda Geetha, Shaik Ahmad Shareef, Bogala Rahul Roy, Indurthi Deva Kumar, Godavarthi Sri Sai Vikas, 2021

This paper introduces a chatbot system for college management that utilizes the Multinomial Naive Bayes algorithm to handle general queries, particularly in scenarios where social distancing is necessary. Developed to assist with college-related management system queries, the chatbot provides responses to routine questions about institutional operations. The system aims to streamline communication by offering automated answers to frequently asked questions, minimizing the need for physical interactions between students and staff. However, the chatbot's scope is limited, as it focuses on general management-related queries rather than retrieving specific student academic data, such as attendance records or grades.

The authors emphasize the effectiveness of the Multinomial Naive Bayes algorithm in handling standard inquiries with high response accuracy, yet acknowledge the limitations of such rule-based approaches in dealing with personalized requests. While this system is beneficial for general management inquiries, it lacks the advanced AI-driven capabilities needed to address individual academic needs. This limitation highlights the need for more specialized solutions, like CampusGuide, that are designed to access and deliver specific academic data on demand, offering a more tailored and responsive experience for students who require immediate access to personal academic records.

CHAPTER 3

REQUIREMENT STUDY AND ANALYSIS

Requirement analysis is the first stage in the systems engineering process and software development process. Requirement analysis in systems engineering and software engineering encompasses those tasks that go into determining the needs or conditions to meet for a new or altered product, taking account of the possibly conflicting requirements of the various stakeholders, such as beneficiaries or users. Requirement analysis is critical to the success of a development project. Requirements must be actionable, measurable, testable, related to identified business needs or opportunities, and defined to a level of detail sufficient for system design. The primary goal of the system analyst is to improve the efficiency of the existing system and for that specification requirement is very essential. For the development of the new system, a preliminary survey of the existing system will be conducted. Investigation is done whether the upgrade of the system into an application program could solve the problems and eradicate the inefficiency of the existing systems.

3.1 Feasibility Study

Feasibility study is a test of system proposal according to its workability, impact of the organization, ability to meet and effective use of resources' focuses on the requisitions. What resources are available for a given candidate system ? Is it worth solving the problem ?.

Feasibility Study is an assessment of the practicality of a proposed project or system. It aims to objectively and rationally uncover the strengths and weaknesses of an existing business or proposed venture, opportunities and threats present in the natural environment, the resources required to carry through, and ultimately the prospects for success. In simplest terms, the two criteria to judge feasibility are cost required and value to be attained.

A project feasibility study is a comprehensive report that examines in detail the five frames of analysis of a given project. It also takes into consideration its four Ps, its risks and POVs, and its constraints. The goal is to determine whether the project should go ahead, be redesigned, or else abandoned altogether. Thus, the four feasibilities are:

- Operational feasibility.
- Technical feasibility.
- Economic feasibility.
- Behavioral feasibility.

3.1.1 Operational Feasibility

The platform requires only basic minimum hardware and software on the client side. It supports any operating system with any browser software installed. It is the measure of how well a proposed system solves the problems, and takes advantage of the opportunities identified during scope definition and how it satisfies the requirements identified in the requirements analysis phase of system development. To ensure success, desired operational outcomes must be imparted during design and development. These include such design-dependent parameters as reliability, maintainability, supportability, usability, producibility, disposability, sustainability, affordability and others. These parameters are required to be considered at the early stages of design if desired operational behaviors are to be realized.

Therefore, operational feasibility is a critical aspect of systems engineering that needs to be an integral part of the early design phases.

➢ **Integration with Educational Institutions:**

- Collaboration with Administrators: Work closely with campus administrators to ensure the chatbot meets the diverse needs of students and integrates with existing systems.
- Feedback Mechanisms: Establish channels for students and faculty to provide feedback on the chatbot’s performance, leading to iterative improvements.
- Training and Workshops: Offer training sessions for students and staff to familiarize them with the chatbot, promoting its use and enhancing user acceptance.

➢ **User Acceptance:**

- Intuitive Design: The chatbot's design prioritizes simplicity and accessibility, enabling users to navigate the system easily and increasing their comfort with the technology.
- Real-Time Responses: By providing immediate answers to academic queries, users will trust the chatbot as a reliable resource for information.
- Empowerment: The ability to access academic data independently boosts students' confidence and sense of autonomy, which is crucial for enhancing their educational experience.
- Security Features: Incorporating features such as data privacy and secure access will further increase user trust and acceptance of the chatbot.

3.1.2 Technical Feasibility

This assessment evaluates the technical feasibility of the CampusGuide chatbot based on an outline design of system requirements. It focuses on understanding the current technical resources available within the organization and their suitability for meeting the expected needs of the proposed system. This evaluation includes an analysis of the hardware, software, and technology stack required to implement the chatbot effectively.

➢ **Technology Stack:**

- Frontend Development: The chatbot interface can be developed using HTML, CSS, and JavaScript, ensuring a responsive design that works seamlessly across various web browsers and devices.

- Backend Development: The system will utilize the Python programming language with the Django framework to manage server-side logic and database interactions effectively.

- Database Management: JSON will be employed as the database management system to store user data, chat logs, and academic information securely.

- AI and NLP Integration: The chatbot will incorporate AI modules for natural language processing (NLP) to understand and respond to user queries accurately.
- Libraries such as Rasa and PyTorch can be utilized for building and training the chatbot's AI capabilities.

➢ **Resource Requirements:**

- Programming Languages: Python for backend development, HTML/CSS/JavaScript
  for frontend development.
- Frameworks: Django for backend development and Rasa for AI processing.
- Database: JSON for data storage and management.
- Development Environment: A stable development environment set up on either Windows, Mac, or Linux operating systems.
- Hardware Requirements: The chatbot can be hosted on a server with basic specifications, such as:
- Processor: Intel i3 or equivalent.
- RAM: 4GB or more for optimal performance.
- Storage: Sufficient space to store the database and application files.

➢ **Technical Challenges:**

- Integration of AI Libraries: Implementing AI functionalities such as natural language understanding and intent recognition will require expertise in AI libraries and frameworks, which may pose a learning curve for the development team.

- Testing Across Platforms: Ensuring compatibility and performance across various devices and web browsers may present challenges, requiring thorough testing and potential adjustments in the codebase.

- Data Privacy and Security: Safeguarding user data and ensuring compliance with data protection regulations (e.g., GDPR) will be critical and may require additional resources for implementing security measures.

- Scalability: The system should be designed to handle increasing user loads as more students engage with the chatbot, necessitating a scalable architecture that can accommodate growth without performance degradation.

3.1.3 Economic Feasibility

The economic viability can be judged by total estimated cost of the project, financing of the project in terms of its capital structure.

➢ **Cost-Benefit Analysis:**

- Includes costs for software tools, licenses, and paying developers, testers, and designers to build the app.
- Covers ongoing expenses for hosting, updates, and support to keep the app running smoothly.

➢ **Return on Investment (ROI):**

- Growing user adoption and partnerships can generate revenue and boost the app's impact and value.

3.1.4 Behavioral Feasibility

Before the development of the Project, we need to study the feasibility of the successful execution of the and thus the following factors are considered for a feasibility study, need analysis, and provide the users information pertaining to the preceding requirement. Behavioral feasibility is a scale of how the proposed system solves the problems, to what extent it takes the advantage of the opportunities identified during scope definition and how much it satisfies the requirements identified in the requirements analysis phase of system development. The results show an enhanced level of performance compared to the existing method by using any other algorithm. It is easily operable. The software programs are more reliable.

3.2 Project Plan

Title	-	CampusGuide: AI Driven College Chatbot
Objective	-	AI chatbot for student information and support.
Starting Date	-	07-07-2024
Ending Date(Till Design)	-	01-11-2024
Project Guide	-	Ms.Najiya Nasrin K
Project Team	-	Abhinand M, Muhammad Aman TK, Neeraj CK, Sufail Salim

3.3 Proposed System

The proposed system for "CampusGuide" is an AI-powered chatbot that assists prospective students in obtaining real-time information about academic programs, campus facilities, and administrative processes. Utilizing natural language processing and machine learning, the chatbot provides accurate responses to student inquiries, streamlining communication and enhancing the overall user experience.

The chatbot operates 24/7, ensuring students can retrieve essential academic data such as scores and attendance at any time, thus reducing the need for manual searches across multiple platforms. It integrates seamlessly with existing campus systems, making it a valuable tool for both students and administrators.

To enhance usability, the system includes a user-friendly interface and personalized assistance features, allowing students to interact with the chatbot easily. Additionally, the system is structured into three main modules: Admin, User, and Support, which facilitate efficient management and quick responses to student queries. Prioritizing accessibility and user experience, CampusGuide aims to empower students by providing them with the information they need to navigate their academic journey confidently.

3.4 Specifications and Requirements

A software requirement specification lays out functional (SRS) is a description of a software system to be developed. It lays out functional and nonfunctional requirements, and may include a set of use cases that describes the user interactions that the software must provide. It should also provide a realistic basis for estimating product cost, risk and schedules.

3.4.1 Functional Requirements

- Provides real-time information about academic programs and campus facilities.
- Retrieves student-specific data such as scores and attendance.
- Facilitates 24/7 access to academic inquiries without human intervention.
- Integrates with existing campus systems for seamless information flow.

- Offers personalized assistance through natural language processing.
- Supports user-friendly interaction with an intuitive interface.
- Enables quick responses to student queries through structured modules (Admin, User, Support)

3.4.2 Non-Functional Requirements

- The application should be easy for users to navigate, ensuring a positive user experience.
- It should provide instant responses to user queries, minimizing any delays in information retrieval.
- The system should be reliable and maintain consistent performance, especially during peak usage times.
- User data must be securely stored and protected to maintain privacy and comply with data protection regulations.

3.4.3 System Requirements

- Needs Intel i5, 8GB RAM, 500GB storage; runs on Windows 8+.
- Needs Android 8+ with 4GB RAM; built with Dart and Flutter.
- Compatible with Windows and Android devices.
- Requires Intel i5 or equivalent processor, 8GB RAM, and 500GB storage; compatible with Windows 10 or higher.
- Compatible with both Windows and Android devices to ensure accessibility for all users.

3.4.4 User Requirements

- Provides clear and concise audio directions for users to facilitate easy navigation through the app.
- Accurately detects and identifies academic-related inquiries and provides relevant information.
- Offers fast access to support through video calls and real-time location tracking features for urgent inquiries.

3.4.5 Software Requirements

Flask:

For the backend framework of CampusGuide, Flask is chosen to support the web component, providing a versatile yet lightweight foundation. Flask, a micro-framework in Python, is well-suited for CampusGuide’s architecture as it facilitates fast, flexible development, allowing developers to tailor components to the project’s specific needs without imposing rigid structure. Flask’s modular nature is especially advantageous in projects like CampusGuide, where dynamic responses and ease of scaling are essential.Flask’s simplicity and ease of integration make it highly effective for CampusGuide’s requirement to communicate with MySQL, allowing smooth data retrieval and efficient storage of user information, such as student attendance and scores. This compatibility ensures that CampusGuide can handle queries and data operations swiftly, making student interactions seamless and immediate.

Additionally, Flask provides flexibility in integrating with various frontend technologies like HTML, CSS, and JavaScript. This enables CampusGuide to deliver a responsive, interactive user experience where backend processes, such as database access and NLP functions, work harmoniously with a smooth, visually engaging interface. By using Flask’s robust routing capabilities, CampusGuide ensures that user queries are quickly directed to the relevant response functions, maintaining responsiveness in user interactions.Moreover, Flask’s ecosystem includes support for several extensions and libraries, such as Flask-RESTful for building REST APIs. These APIs enable CampusGuide to communicate between backend and frontend components securely, manage data, and support the chatbot’s real-time response capabilities. Flask’s flexibility with add-ons for security features like authentication and authorization further strengthens CampusGuide’s ability to safeguard user data, which is crucial in a student-focused application.

Overall, Flask’s lightweight framework empowers CampusGuide with a foundation that is both scalable and adaptable, ensuring smooth functionality for users while supporting CampusGuide’s ability to grow and enhance its feature set as requirements evolve.

CHAPTER 4

SYSTEM DESIGN AND DEVELOPMENT

4.1 System Architecture

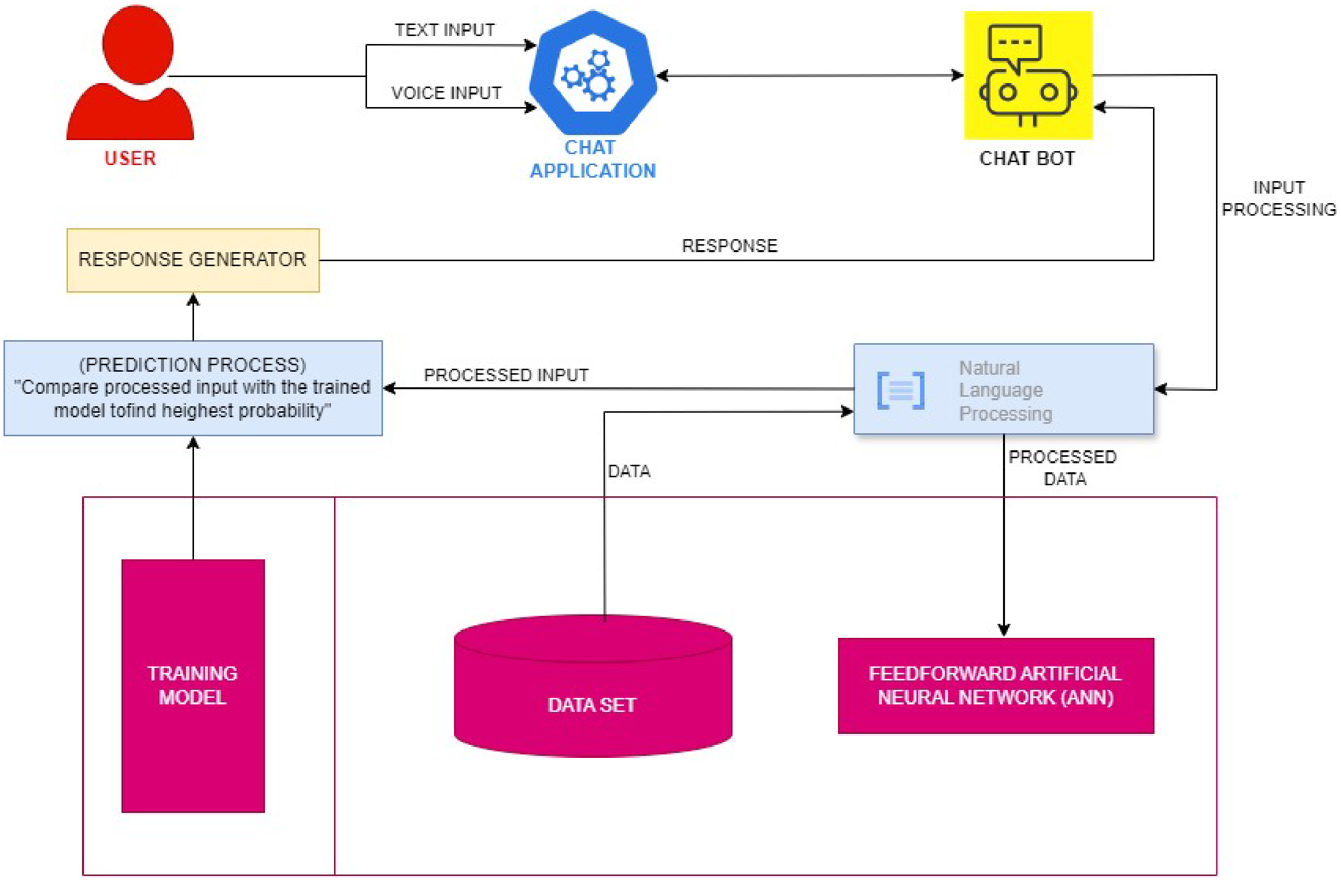
USER
TEXT INPUT
VOICE INPUT
CHAT APPLICATION
CHAT BOT
INPUT PROCESSING
RESPONSE GENERATOR
RESPONSE
(PREDICTION PROCESS) "Compare processed input with the trained model tofind heighest probability"
PROCESSED INPUT
Natural Language Processing
DATA
PROCESSED DATA
TRAINING MODEL
DATA SET
FEEDFORWARD ARTIFICIAL NEURAL NETWORK (ANN)

Fig 4.1 .System Architecture

- User provides input through text or voice to the chat application interface.
- Application receives the user input, converting voice to text using a speech recognition module if necessary.
- The raw input data is captured and prepared for initial processing by the system.
- Input undergoes validation to check for empty responses, prohibited content, or inappropriate language.

- Preprocessing of the input starts, involving steps like tokenization and normalization to make the text suitable for analysis.
- Extraneous elements such as excessive punctuation, special characters, or extra whitespace are removed to clean up the input data.
- Tokenization splits the input into individual words or subwords, breaking down sentences for easier analysis.
- Normalization steps, including lowercasing, stemming, or lemmatization, create a uniform text format, enhancing consistency.
- Each token is tagged with its Part-of-Speech (POS), identifying nouns, verbs, adjectives, etc., to help the model understand the sentence structure.
- Named Entity Recognition (NER) identifies key entities, such as names, dates, or locations, within the input for more accurate context.
- Dependency parsing examines grammatical relationships between words, allowing the model to interpret sentence context.
- Sentiment analysis may be applied to assess the user’s mood or attitude, aiding in generating a more suitable response.
- The processed input is fed into the NLP model, which interprets the meaning and intent behind the user's message.
- A trained machine learning model, often a neural network, predicts the most appropriate response based on learned conversation patterns.
- The final response is validated, then displayed as text or converted to audio for playback, completing the user interaction.

4.2 Activity Diagram

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. An activity diagram is a type of diagram used in Unified Modeling Language (UML).In theUnified Modeling Language, activity diagrams are intended to model both computational and organizational processes, as well as the data flows intersecting with the related activities. Although activity diagrams primarily show the overall flow of control, they can also include elements showing the flow of more data stores.These helps

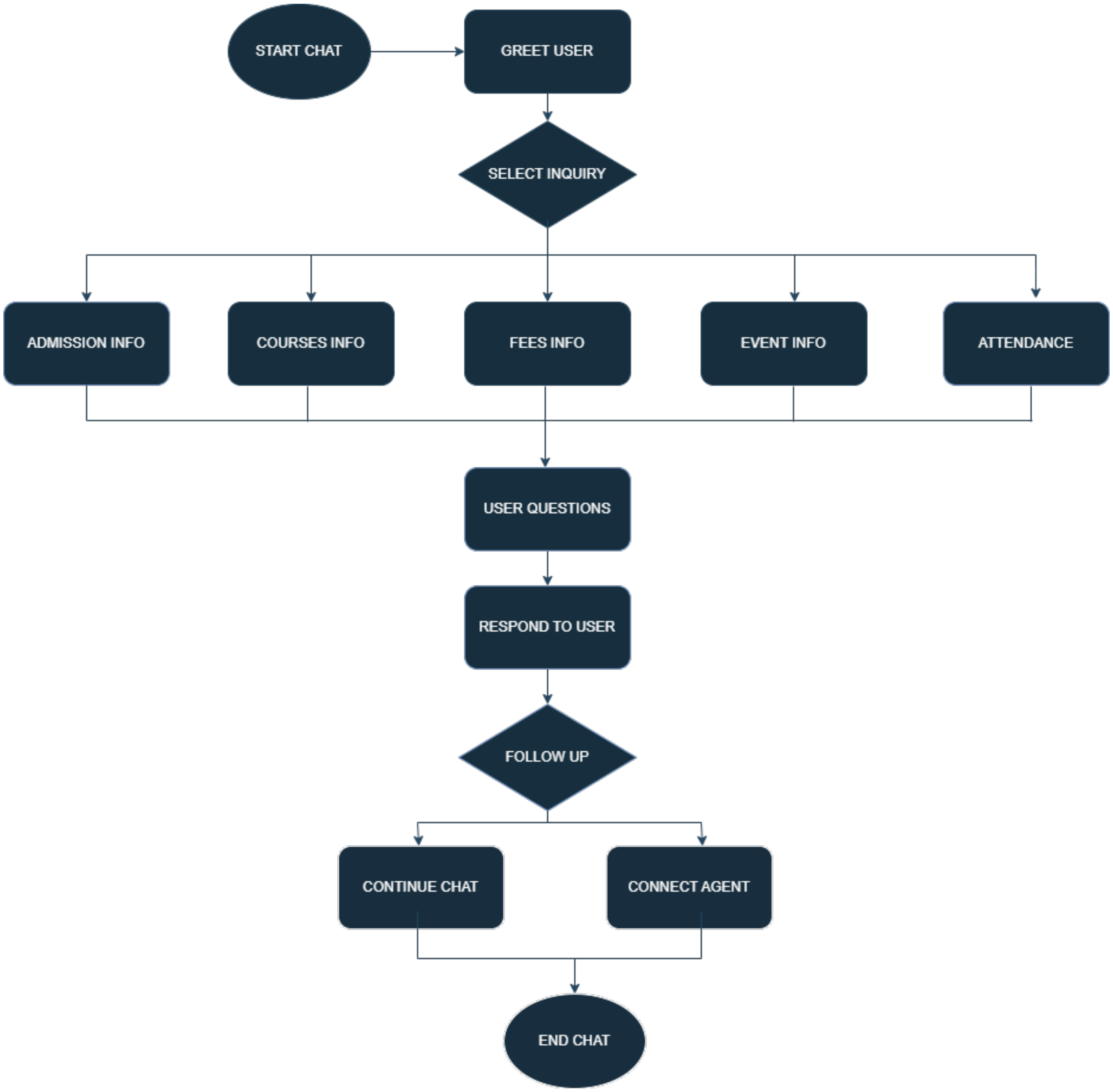

Fig 4.2 Activity Diagram

- The conversation begins when the user initiates a chat session, either by text or voice input.
- The chatbot greets the user with a welcoming message, setting a friendly and approachable tone.
- The chatbot prompts the user to choose from a list of topics to guide the conversation effectively.
- The user is presented with several predefined options to select, based on their interests or needs.
- “Information about the admission process” is an option that includes inquiries about application deadlines, requirements, and eligibility.
- “Details about available courses” provides information on subjects, curriculum, prerequisites, and course duration.
- Another option, “Information regarding fees and payment,” allows the user to inquire about tuition costs, payment plans, and deadlines.
- “Information about upcoming events” gives users details about academic or extracurricular events, including registration and schedules.
- The “Information about attendance records or policies” option allows students to check their attendance records or understand the attendance policies.
- After the user selects a topic, the chatbot invites them to ask specific questions within the chosen topic.
- The user enters their specific question, and the chatbot analyzes keywords and context to prepare a targeted response.
- The chatbot retrieves relevant information from its knowledge base or database, quickly processing the user’s question.
- Based on the topic and the specific query, the chatbot provides a clear and relevant response, aiming to answer accurately.
- After answering, the chatbot checks if the user has additional questions or needs further assistance, displaying a proactive approach.

- If the user has more questions on the same topic, the chatbot allows them to ask additional questions to deepen the conversation.
- For users seeking different information, the chatbot can guide them back to the main menu to select another topic.
- In situations where the user has complex or unique needs beyond the chatbot's scope, it suggests connecting them to a human agent.
- If the user agrees to the transfer, the chatbot initiates a seamless handover to a human representative for personalized assistance.
- If the user indicates that their questions are fully addressed, the chatbot acknowledges this and ends the conversation politely.
- The chatbot closes the session with a thank-you message, inviting the user to return with future questions and logging the interaction for learning purposes.

4.3 Data Flow Diagrams

LEVEL 0

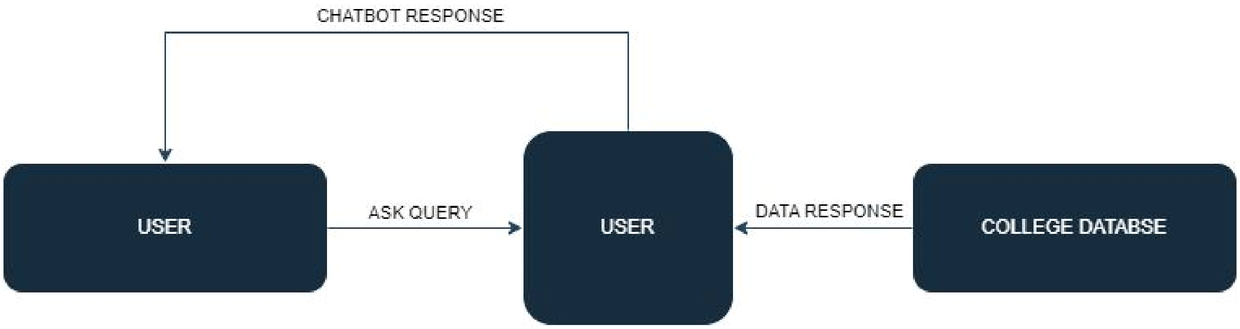

Fig 4.3.1 DFD Level 0

LEVEL 1.1

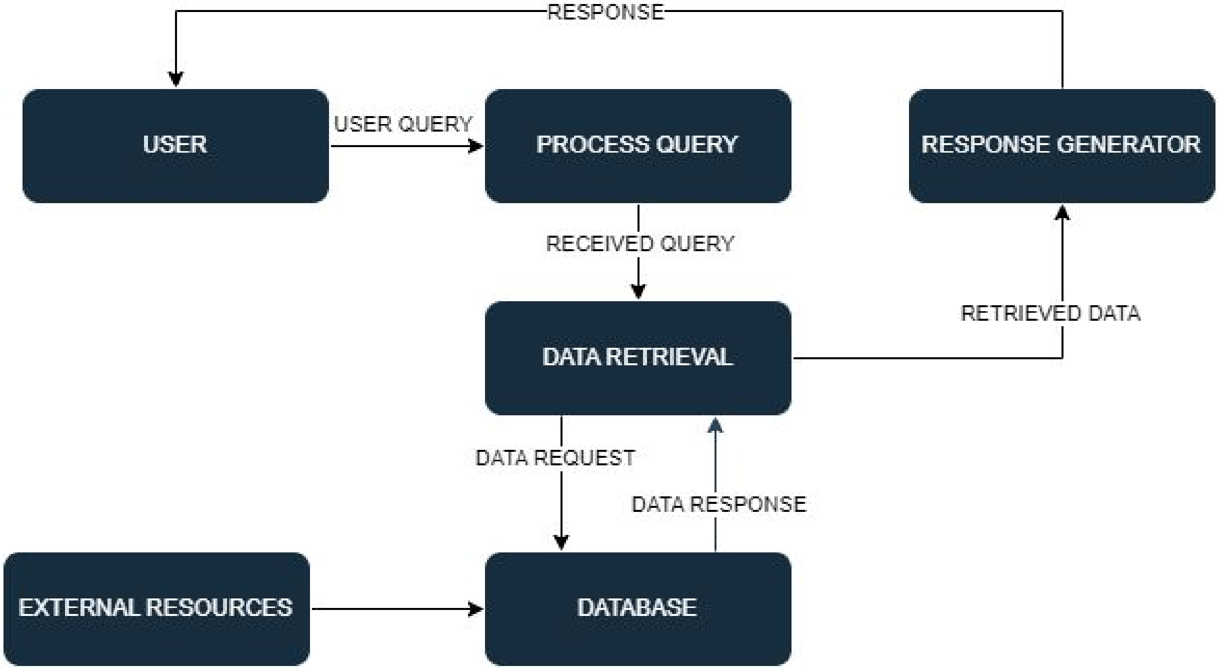

Fig 4.3.2 DFD Level 1

LEVEL 1.2

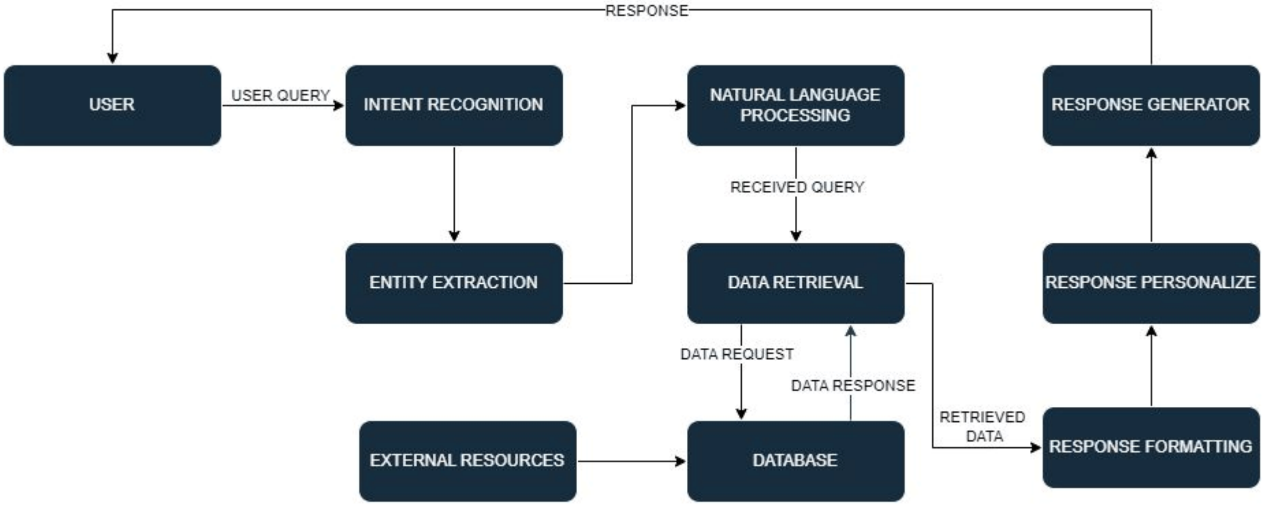

Fig: 4.3.3 DFD Level 2

4.4 Flow Chart

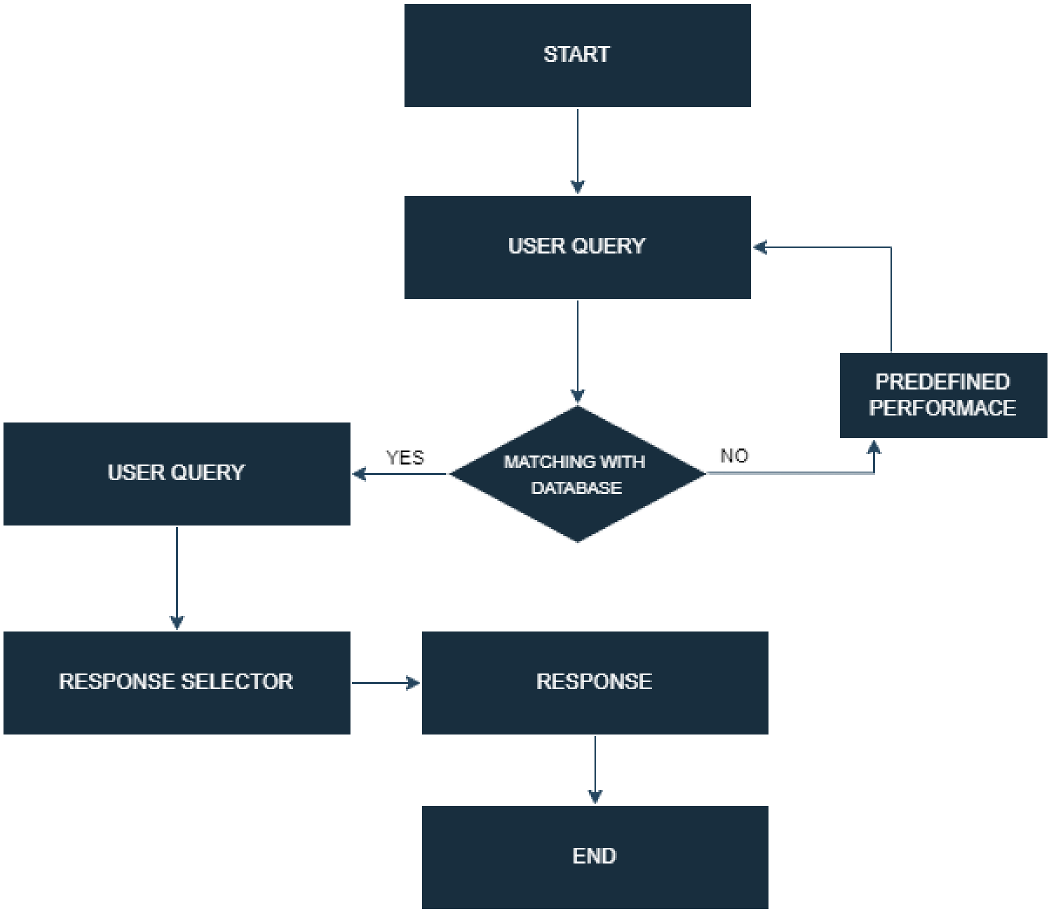

Fig 4.5 FlowChart

REFERENCES

[1] M R. Parkar, Y. Payare, K. Mithari, J. Nambiar and J. Gupta, ”AI And Web-Based Interactive College Enquiry Chatbot,” 2021 13th International Conference on Electronics, Computers and Artificial Intelligence (ECAI), Pitesti, Romania, 2021, pp. 1-5, doi:10.1109/ECAI52376.2021.9515065.

[2] Chiara Valentina Misischia, Flora Poecze, Christine Strauss, Chatbots in customer service: Their relevance and impact on service quality, Procedia Computer Science, Volume 201, 2022, Pages 421-428, ISSN1877 0509

[3] B. R. Ranoliya, N. Raghuwanshi and S. Singh, ”Chatbot for university related FAQs,” International Conference on Advances in Computing, Communications and Informatics (ICACCI), Udupi, 2017, pp.1525 1530.

[4] N. P. Patel, D. R. Parikh, D. A. Patel and R. R. Patel, ”AI and Web-Based Human-Like Interactive University Chatbot (UNIBOT),”2019 3rd International conference on Electronics, Communication and Aerospace Technology (ICECA), Coimbatore, India, 2019, pp. 148-150, doi: 10.1109/ICECA.2019.8822176.

[5] Manoj Joshi , Rakesh Kumar Sharma, 2020, An Analytical Study and Review of open Source Chatbot framework, RASA, INTERNATIONAL JOURNAL OF ENGINEERING RESEARCH AND TECHNOLOGY (IJERT) Volume 09, Issue 06 ,June 2020.

[6] Delphin Lydia,J Sangeetha Priya, Dhivya Jai Sree G,Vidhya, Vi-jayameena S, Chatbot for College Management System, International Journal for Research in Applied Science and Engineering Technology (IJRA), Volume 9, Issue 7, July 2021.

[7] Attigeri, G., Agrawal, A. and Kolekar, S., 2024. Advanced NLP Models for Technical University Information Chatbots: Development and Comparative Analysis. *IEEE Access*.

[8] Patel, D., Shetty, N., Kapasi, P. and Kangriwala, I., 2023. College enquiry chatbot using conversational AI. *International Journal for Research in Applied Science & Engineering Technology (IJRASET)*, *11*(5).

[9] Bieletzke, S., 2023. Ai-chatbot-integration in campus-management-systems. In *EDULEARN23*

[10] *Proceedings* (pp. 3574-3583). IATED.

[11] Assayed, S.K., Alkhatib, M. and Shaalan, K., 2023, May. Artificial intelligence based chatbot for promoting equality in high school advising. In *2023 4th International Conference on Intelligent Engineering and Management (ICIEM)* (pp. 1-4). IEEE.