e-ISSN:2582-7219

INTERNATIONAL JOURNAL OF MULTIDISCIPLINARY RESEARCH IN SCIENCE, ENGINEERING AND TECHNOLOGY

Volume 7, Issue 4, April 2024

Impact Factor: 7.521

International Journal Of Multidisciplinary Research in Science, Engineering and Technology (IJMRSET)

| ISSN: 2582-7219 | www.ijmrset.com | Impact Factor: 7.521| Monthly Peer Reviewed & Referred Journal |

| Volume 7, Issue 4, April 2024 |

| DOI:10.15680/IJMRSET.2024.0704059 |

Components of Smart Chatbot Academic Model for A University Website

Prof. Jayshree Bhoj, Kenil Daslaniya , Suraj Narekar

Department of Computer Engineering, Sandip Institute of Technology and Research Centre Nashik, India

ABSTRACT: The websites have been popular in today's world, where people tend to search and get the information through browsing a website. Websites are making the searching information process easier toward people's daily life as compared to twenty years ago. Thus, most of the companies, institutions even government have built up their own websites including the universities. The use of websites could be various purposes based on the owner's intention to have it. However, the growing adoption of the websites has brought many problems in engaging the engagement between the university websites and the stakeholder. In order to increase the interactivity of the website, most website's owner has implemented an added technology to overcome it. One of the rise technologies currently is Chatbots.

In the ever-evolving landscape of higher education, the fusion of cutting-edge technology and pedagogy has become a hallmark of progress. One such revolutionary innovation that stands at the forefront of this transformation is the integration of Smart College Chatbots, a dynamic solution propelled by the prowess of Machine Learning (ML) and the versatility of Python programming. This comprehensive exploration delves into the multifaceted dimensions of Smart College Chatbots, unraveling the intricate tapestry of their development, implementation, and the transformative impact they promise to have on the student experience.

As an embodiment of technological innovation, Smart College Chatbots introduce a plethora of features poised to redefine how students engage with educational institutions. One standout characteristic is the ability to provide personalized assistance. By harnessing the power of ML, these chatbots analyze user behavior and historical data to tailor responses, offering a level of personalization that transcends the traditional one-size-fits-all approach to information dissemination. A chatbot (otherwise called a chatterbox, Bot, or Artificial Conversational Entity is a program that copies human discussions including content and communication in natural language utilizing artificial intelligence methods, for example, Natural Language Processing (NLP), picture and video processing, and voice analysis. A chatbot is a specialist that assists users in utilizing natural language. It was worked as an endeavor to trick people. Whether its queries about courses, exam schedules, or campus events, the Chabot's responses are finely tuned to meet the unique needs of each student. Beyond personalized assistance, the Smart College Chatbot promises 24/7 accessibility, breaking down temporal and spatial barriers. Students can seamlessly seek information, clarification, or support at any time, fostering a culture of continuous learning and accessibility. This continuous support aligns with the modern student's lifestyle, accommodating diverse schedules and time zones.

I. INTRODUCTION

At the core of Smart College Chatbots lies the intricate synergy between Machine Learning and Python programming. Machine Learning algorithms, specifically tailored for Natural Language Processing (NLP), empower these chatbots to comprehend and respond to user queries with a human-like intelligence. The adaptive nature of ML algorithms ensures that the chatbot continually refines its responses, learning from user interactions to provide an increasingly sophisticated and tailored experience over time. Python, celebrated for its readability, versatility, and an extensive array of libraries, serves as the engine driving the development of Smart College Chatbots. The language's agility allows for rapid prototyping, a crucial factor in the ever-evolving landscape of education. Leveraging libraries such as TensorFlow and PyTorch, developers can seamlessly implement intricate ML models, ensuring that the chatbot evolves in tandem with the dynamic needs of the academic environment

| ISSN: 2582-7219 | www.ijmrset.com | Impact Factor: 7.521| Monthly Peer Reviewed & Referred Journal |

| Volume 7, Issue 4, April 2024 |

| DOI:10.15680/IJMRSET.2024.0704059 |

II. FUNCTIONAL REQUIREMENTS

2.1 SYSTEM FEATURE

- User Interface: Design a user-friendly and intuitive interface for the chatbot. Ensure compatibility with various platforms
- Natural Language Processing (NLP): Implement NLP algorithms to enable the chatbot to understand and interpret user queries in natural language.
- Knowledge Base: Create a comprehensive knowledge base with accurate and up-to-date medical information, including symptoms, diseases, treatments, medications, and general health advice. Regularly update the knowledge base to reflect the latest medical advancements.
- Symptom Checker and Triage: Develop a symptom checker feature to help users assess their condition and severity. Provide recommendations for self-care or suggest seeking immediate medical attention when necessary.
- Scalability: Design the system with scalability in mind to accommodate a growing user base and increased functionalities.

2.2 OBJECTIVES:

The objectives of developing the components of a Smart Chatbot Academic Model for a University Website should align with addressing the identified challenges and improving the overall user experience. The following are the objectives of the Chatbot.

- Implement machine learning algorithms to personalize user interactions, learning from user preferences and behaviors to provide tailored responses.
- Implement advanced Natural Language Processing (NLP) algorithms to enable the chatbot to swiftly and accurately retrieve relevant information from the university's database.
- To develop interactive features within the chatbot, enabling users to engage in dynamic conversations, receive instant responses to queries, and access multimedia content.
- Implement machine learning algorithms to personalize user interactions, learning from user preferences and behaviors to provide tailored responses.
- Develop user education materials and campaigns to promote awareness and encourage the adoption of the chatbot for various university-related inquiries.

III. EXTERNAL INTERFACE REQUIREMENTS

3.1 USER INTERFACE

- Web Browser Compatibility**:** The chatbot interface should be compatible with major web browsers (Google Chrome, Mozilla Firefox, Safari, etc.) to ensure a consistent user experience.
- Web-based Communication: The chatbot will communicate with users and administrators through web-based interactions on the college website.

3.2 HARDWARE INTERFACE

- Internet Connectivity: Users and administrators need stable internet connectivity to access the chatbot.

3.3 SOFTWARE INTERFACE

- Programming Language: The chatbot is developed using Python, and it may have dependencies on specific Python libraries and frameworks (NLTK, Flask, etc.).
- Database Integration: Integration with a database system to store and retrieve information. The choice of database technology should be documented.

3.4 COMMUNICATION INTERFACE

- Secure Login: Administrators access the control panel through a secure login system.
- User Management: The control panel allows administrators to manage user data, access logs, and other relevant information.
- Knowledge Base Update: Administrators can update the chatbots knowledge base, add new responses, or modify existing ones through the control panel.

IV. NONFUNCTIONAL REQUIREMENTS

Nonfunctional requirements define the aspects of a system that are not related to specific behaviors or features but are critical for its overall performance, usability, security, and reliability. Here are the nonfunctional requirements for the Smart College Chatbot:

Performance:

- Response Time: The chatbot should provide responses within a reasonable time frame, typically under a few seconds, to ensure a smooth user experience.
- Concurrent User Handling: The system should be capable of handling multiple concurrent users without significant degradation in performance.

Scalability:

- Capacity Planning: The chatbot system should be scalable to accommodate a growing number of users and data without compromising performance.
- Load Balancing: Implement load balancing mechanisms to distribute user requests evenly across server resources.

Reliability:

- Uptime: The chatbot system should strive for high availability, with minimal downtime for maintenance or unexpected issues.
- Fault Tolerance: The system should be resilient to failures, with mechanisms in place to recover gracefully from errors or disruptions.

Scalability:

- Capacity Planning: The chatbot system should be scalable to accommodate a growing number of users and data without compromising performance.
- Load Balancing: Implement load balancing mechanisms to distribute user requests evenly across server resources

4.1 SAFETY REQUIREMENTS

While the primary focus of a chatbot system is on functionality and user experience, certain safety requirements should be considered to ensure ethical, secure, and responsible usage. Here are key safety requirements for the Smart College Chatbot:

- Data Privacy: Ensure that user data, including queries and personal information, is handled with strict privacy measures.
- Content Moderation: Implement content moderation mechanisms to filter inappropriate or offensive content in user queries and responses. Regularly update the moderation rules to adapt to emerging language trends
- Preventing Misuse: Implement measures to prevent the chatbot from being misused for malicious purposes. Monitor user interactions for any signs of abuse or inappropriate behaviour.
- User Consent: Obtain clear and informed consent from users regarding data collection and usage. Clearly communicate the chatbot's capabilities and limitations to users.
-

V. SYSTEM REQUIREMENTS

5.1 DATABASE REQUIREMENTS

The generated response from the chatbot exhibits a remarkable level of naturalness & resembling. Following are database requirements

5.1.1 Text classifiers

In chatbot development, text classification is a typical technique where the chatbot is educated to comprehend the intent of the user's input and reply appropriately. Text classifiers examine the incoming text and group it into intended categories after analysis. Certain intentions may be predefined based on the chatbots use case or domain. Text classification can be accomplished using deep learning models like Convolutional Neural Networks (CNN) or Recurrent Neural Networks (RNN), as well as machine learning methods like naive bayes and Support Vector Machines (SVM). Since the input text is linked to specified categories or intentions, these models are trained on labelled data, and to forecast the intent of the incoming text input, the model learns the patterns and features in the text data.

Once the intent of the text input has been determined, the chatbot can produce a response or carry out the appropriate activities in accordance with the programmed responses or actions related to that intent. For instance, if the

| ISSN: 2582-7219 | www.ijmrset.com | Impact Factor: 7.521| Monthly Peer Reviewed & Referred Journal |

| Volume 7, Issue 4, April 2024 |

| DOI:10.15680/IJMRSET.2024.0704059 |

user wants to book a flight, the chatbot can request essential details, such as the destination, time of travel, and the number of passengers, before booking the flight as necessary.

Text classifiers are a crucial part of chatbot architecture because they allow the chatbot to comprehend user input and react accordingly based on the user's intent, resulting in a more tailored and meaningful conversation experience.

5.1.2 Natural Language Processing (NLP)

Natural Language Processing (NLP) is a subfield of artificial intelligence that enable computers to understand, interpret, and respond to human language. Applications for NLP include chatbots, virtual assistants, sentiment analysis, language translation, and many more.

The processing of human language by NLP engines frequently relies on libraries and, which are frequent words like "and," "the," and "is," can be safely eliminated.

Further, lemmatization and stemming are methods for condensing words to their root or fundamental form. While stemming entails truncating words to their root form, lemmatization reduces words to their basic form (lemma). For instance, "running" would be stemmed and lemmatized to "run." Part of Speech (POS) tagging is then employed, which entails classifying each word in the text according to its grammatical type, such as a noun, verb, adjective, etc. Understanding the grammatical structure of the text and gleaning relevant data is made easier with this information.

Named Entity Recognition (NER) is a crucial NLP task that involves locating and extracting specified data from user input, including names of individuals, groups, places, dates, and other pertinent entities. The chatbot or other NLP programs can use this information to interpret the user's purpose, deliver suitable responses, and take pertinent actions.

The chatbot responds based on the input message, intent, entities, sentiment, and dialogue context. This can involve selecting pre-defined responses from a set of templates, dynamically producing responses using text generation techniques like language models, or combining pre-defined templates with dynamic text generation to provide tailored responses. Natural language generation is the next step for converting the generated response into grammatical and human-readable natural language prose. This process may include putting together pre-defined text snippets, replacing dynamic material with entity values or system-generated data, and assuring the resultant text is cohesive. The chatbot replies with the produced response, displayed on the chat interface for the user to read and respond to. The chatbot may continue to converse with the user back and forth, going through the above-said steps for each input and producing pertinent responses based on the context of the current conversation.

5.1.3 Knowledge base

A knowledge base is a collection of data that a chatbot utilizes to generate answers to user questions. It acts as a repository of knowledge and data for the chatbot to deliver precise and accurate answers to user inquiries.

Installing the chatbot will determine the exact format of the knowledge base. It could be a database with organized data like facts, Frequently Asked Questions (FAQs), product details, or other pertinent data. The chatbot can run a query against this database to get the relevant data and utilize it to generate responses. A knowledge base could also include a library of pre-written responses or templates that the chatbot can utilize to create responses on the go. These pre-written replies can be created to address frequent user scenarios and queries. The following steps are often required for a chatbot's knowledge base to function:

Collecting essential data is the first stage in creating a knowledge base. Text files, databases, webpages, or other information sources create the knowledge base for the chatbot. After the data has been gathered, it must be transformed into a form the chatbot can understand. Tasks like cleaning, normalizing, and structuring may be necessary to ensure the data is searchable and retrievable.

The knowledge base's content must be structured so the chatbot can easily access it to obtain information. To do this, it may be necessary to organize the data using techniques like taxonomies or ontologies, natural language processing (NLP), text mining, or data mining.

The knowledge base must be indexed to facilitate a speedy and effective search. Various methods, including keyword-based, semantic, and vector-based indexing, are employed to improve search performance. The collected data may subsequently be graded according to relevance, accuracy, or other factors to give the user the most pertinent information. The chatbot explores the knowledge base to find relevant information when it receives a user inquiry. After retrieving the required data, the chatbot creates an answer based on the information found.

A knowledge base must be updated frequently to stay informed because it is not static. Chatbots can continuously increase the knowledge base by utilizing machine learning, data analytics, and user feedback. To keep the knowledge base updated and accurate, new data can be added, and old data can be removed. The knowledge base is

| ISSN: 2582-7219 | www.ijmrset.com | Impact Factor: 7.521| Monthly Peer Reviewed & Referred Journal |

| Volume 7, Issue 4, April 2024 |

| DOI:10.15680/IJMRSET.2024.0704059 |

connected with the chatbot's dialogue management module to facilitate seamless user engagement. The dialogue management component can direct questions to the knowledge base, retrieve data, and provide answers using the data.

A chatbot knowledge base generally functions by gathering, processing, organizing, and expressing information to facilitate effective search, retrieval, and response creation. It is an essential element that allows chatbots to offer users accurate and relevant information and continuously enhance their performance through continuous learning.

5.1.4 Dialogue management

A crucial part of a chatbot is dialogue management which controls the direction and context of the user's interaction. Dialogue management is responsible for managing the conversation flow and context of the conversation. It keeps track of the conversation history, manages user requests, and maintains the state of the conversation. Dialogue management determines which responses to generate based on the conversation context and user input. Let's explore the technicalities of how dialogue management functions in a chatbot.

- Input processing: The chatbot accepts inputs from the user, which may be speech or text. The task of retrieving pertinent data from the input, including intent, entities, and context, falls into the input processing module. Variables, such as dates, names, or locations, are the specific bits of information provided as the input, whereas intent represents the user's intention or purpose behind the input. Context is used to describe pertinent details from the conversation's earlier turns, which aids in preserving the dialogue's flow.
- Intent recognition: Inferring the user's intention based on their input is a crucial step in dialogue management for chatbots. Various methods can be employed to achieve this, including rule-based strategies, statistical approaches, and machine learning algorithms such as natural language understanding (NLU) models. NLU models utilize techniques like deep learning, recurrent neural networks, or transformers to process and understand the input text, enabling them to identify the intent behind the user's message.
- Dialogue state tracking: The chatbot updates its internal dialogue state, which depicts the current context and stage of the interaction, as soon as the intent is identified. The dialogue state contains any important context that must be retained to preserve discussion consistency, such as user choices, system answers, and other information. Dialogue state tracking is essential for managing the dynamic nature of discussions and delivering suitable responses based on the current state of the dialogue.
- Conversation policy: The dialogue policy decides the chatbot's action or response based on the current dialogue state. The conversation policy may be rule-based, in which case previously established rules determine the response, or it may be machine-learned using deep learning or reinforcement learning. The chatbot is trained using reinforcement learning to discover the best course of action based on a reward signal that denotes the effectiveness of the action taken. Large amounts of training data can be utilized to learn the dialogue policy using deep learning techniques like neural networks.
- Response generation: The chatbot generates a response to the user once the dialogue policy decides the proper action or reaction. Natural Language Generation (NLG) techniques or predefined templates can generate the response, with placeholders filled in with the relevant data from the dialogue state. The goal of NLG techniques is to produce text that sounds like human speech based on the state of the conversation and the desired response. These techniques can use template-based generation, rule-based generation, or machine learning-based generation using tools like recurrent neural networks or transformers.
- System response: The system responds by sending the generated response back to the user. Depending on the chatbot's capabilities and the user's desires, the system's answer may be in text, speech, or other modalities.
- User interaction and feedback: The chatbot keeps up the conversation with the user, taking in new information and changing the dialogue's status in response to the user's responses. The chatbot may also gather user feedback to improve over time by enhancing responses.

5.1.5 Machine learning models

The components of the chatbot architecture heavily rely on machine learning models to comprehend user input, retrieve pertinent data, produce responses, and enhance the user experience.

Collecting data is the initial step in creating an ML-based chatbot. A wide variety of inputs and outputs, including text dialogues, user questions, and related answers, can be included in this data. The ML algorithms are trained using this data to recognize and respond to various inputs and query types.

After data collection is pre-processed to eliminate noise and irrelevant information, the data is extracted with the required features or patterns from the text data. Tokenization, a technique that separates text into individual words or phrases and generates numerical representations of these words or phrases, such as word embeddings or bag-of-

| ISSN: 2582-7219 | www.ijmrset.com | Impact Factor: 7.521| Monthly Peer Reviewed & Referred Journal |

| Volume 7, Issue 4, April 2024 |

| DOI:10.15680/IJMRSET.2024.0704059 |

words representations, may be used in this situation. These features operate as inputs to the ML algorithms, assisting them in interpreting the meaning of the text.

The ML model needs to be trained after the data has been preprocessed and the features are extracted. Chatbots can use a variety of machine learning algorithms, including rule-based algorithms, traditional ML algorithms like naive bayes and decision trees, and more sophisticated ML algorithms like recurrent neural networks, convolutional neural networks, and transformer models. The model builds a mapping between user queries and responses during training by learning from the input attributes and associated outputs.

The ML model must be tested after training to gauge its effectiveness. A valid set of data—which was not used during training—is often used to accomplish this. The model's performance can be assessed using various criteria, including accuracy, precision, and recall. Additional tuning or retraining may be necessary if the model is not up to the mark. Once trained and assessed, the ML model can be used in a production context as a chatbot. Based on the trained ML model, the chatbot can converse with people, comprehend their questions, and produce pertinent responses. For a more engaging and dynamic conversation experience, the chatbot can contain extra functions like natural language processing for intent identification, sentiment analysis, and dialogue management.

After deployment, an iterative procedure can continuously enhance the chatbot. The ML model can be further trained and improved by gathering user input and interactions. The chatbot can continuously learn from and adjust to human behavior thanks to this feedback loop, which helps it perform better over time.

Machine learning plays a crucial role in the technical operation of chatbots by enabling them to comprehend user inputs, produce suitable responses, and enhance their performance over time through continuous learning and adaptation.

VI. SYSTEM DESIGN

6.1 SYSTEM ARCHITECTURE:

Designing the system architecture for a smart college chatbot using Machine Learning (ML) and Python involves defining the components and their interactions.

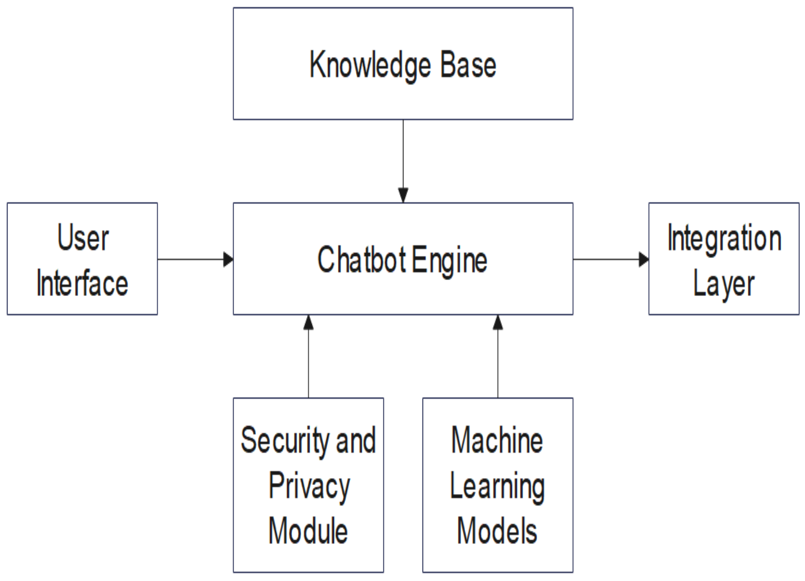


Fig 6.1: System Architecture of proposed system

User Interface (UI):
Description: The interface through which users interact with the chatbot.

Features:
Chat window for text-based communication.
Option for voice input/output if required.
User authentication for personalized interactions.

| ISSN: 2582-7219 | www.ijmrset.com | Impact Factor: 7.521| Monthly Peer Reviewed & Referred Journal |

| Volume 7, Issue 4, April 2024 |

| DOI:10.15680/IJMRSET.2024.0704059 |

Chatbot Engine:
Description: The core logic that processes user inputs, understands intents, and generates appropriate responses.

Features:
Natural Language Processing (NLP) module for understanding user queries.
Intent recognition for identifying user goals.
Dialog management for maintaining context in conversations.
Integration with ML models for continuous learning.

Machine Learning Models:
Description: ML models for tasks such as NLP, intent recognition, and user behaviour prediction.

Features:
Natural Language Processing (NLP) model for understanding and processing textual input.
Intent recognition model to identify the user's intent or purpose.
Entity recognition for extracting relevant information from user queries.
Recommender system for suggesting relevant information.

Knowledge Base:
Description: Repository of information that the chatbot can access to provide accurate and up-to-date responses.

Features:
Information about courses, admissions, campus facilities, events, etc.
Integration with university databases and APIs.
Regular updates to ensure information accuracy.

Integration Layer:
Description: Facilitates communication between the chatbot system and external systems, databases, and APIs.

Features:
Connects to university databases for real-time data retrieval.
Integrates with external APIs for additional information.
Ensures data consistency and integrity.

Security and Privacy Module:
Description: Ensures the security and privacy of user data and interactions.

Features:
Data encryption for sensitive information.
Compliance with data protection regulations.
Regular security audits and updates.

| ISSN: 2582-7219 | www.ijmrset.com | Impact Factor: 7.521| Monthly Peer Reviewed & Referred Journal |

| Volume 7, Issue 4, April 2024 |

| DOI:10.15680/IJMRSET.2024.0704059 |

6.2 DATA FLOW DIAGRAMS

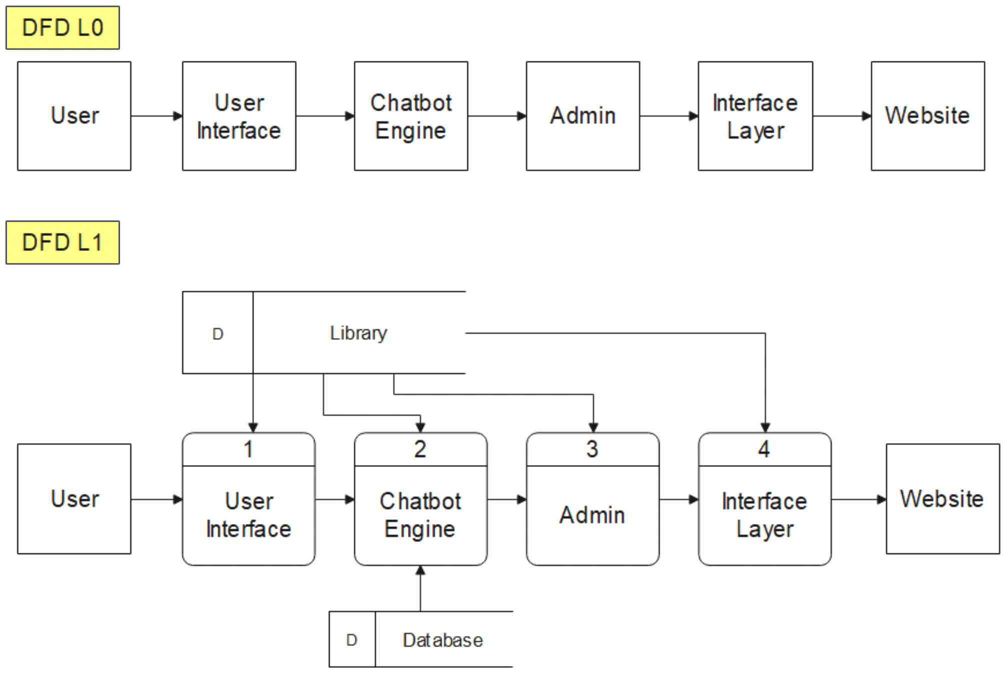


Fig 6.2: Data Flow diagram

- User interacts with the chatbot through the UI, providing queries or seeking information.
- The Chatbot Engine processes the user input using NLP and ML models.
- Relevant information is retrieved from the Knowledge Base and external systems through the Integration Layer.
- The chatbot generates a response based on the processed input and retrieved information.
- User profiles and preferences are managed through the User Management System.
- Feedback from users is collected through the Feedback and Improvement Module.
- The system continuously learns and improves through model updates and user feedback.

6.3 ENTITY RELATIONSHIP DIAGRAMS

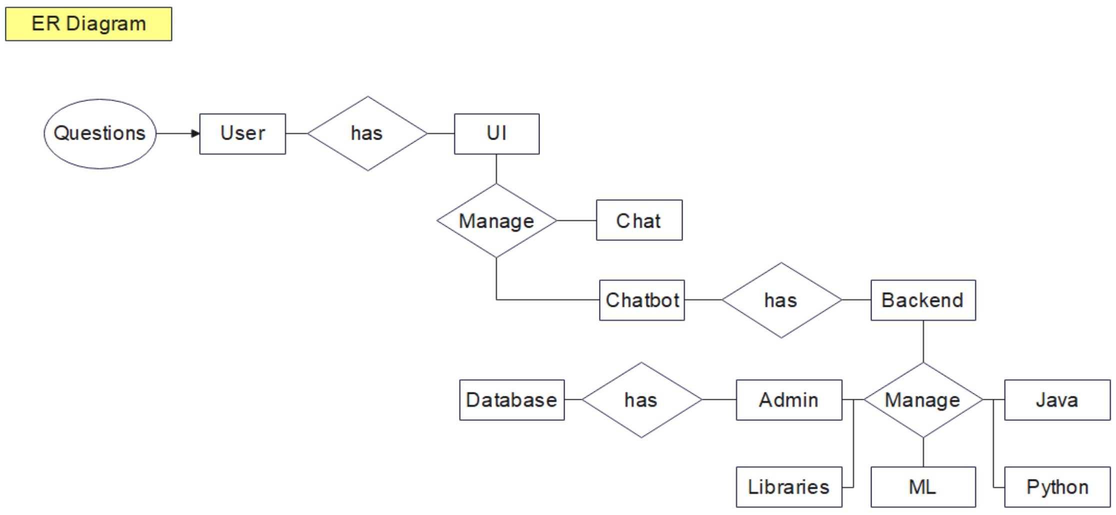


Fig 6.3: Entity Relationship Diagrams

| ISSN: 2582-7219 | www.ijmrset.com | Impact Factor: 7.521| Monthly Peer Reviewed & Referred Journal |

| Volume 7, Issue 4, April 2024 |

| DOI:10.15680/IJMRSET.2024.0704059 |

6.4 UML DIAGRAMS

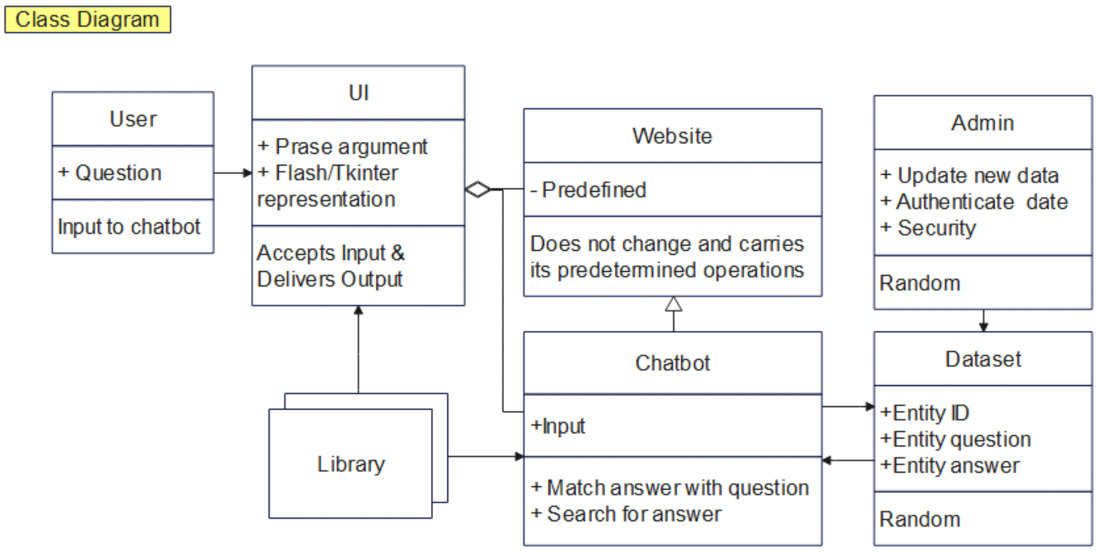


Fig 6.4: UML Class diagram

VII. CONCLUSION

In this project we will make a college specific chatbot system that can be custom fitted to education domain chatbot. This will make the user interactive more meaningful as it responds to the user queries very accurately as it is a domain specific chatbot system, and furthermore we had investigated our college chatbot system design stages and a few different techniques by which the precision of the chatbot system can be made much better.

The development of a Smart College Chatbot using ML and Python represents a forward-looking approach to improving the overall college experience. By harnessing the power of machine learning and natural language understanding, educational institutions can provide a more accessible, efficient, and user-centric interface for their students, faculty, and staff. As technology continues to advance, these smart chatbots are poised to play a pivotal role in shaping the future of education.

VIII. FUTURE SCOPE

The future scope of chatbots is dynamic and promising, with continuous advancements in technology and artificial intelligence. Here are some key areas of future development for chatbots:

- Integration with Emerging Technologies: Future Development: Chatbots will increasingly integrate with emerging technologies such as augmented reality (AR), virtual reality (VR), and mixed reality (MR) to provide more immersive and interactive user experiences.
- Multilingual and Cross-Platform Support: Future Development: Chatbots will evolve to support multiple languages seamlessly and operate across various platforms, ensuring a wider user reach and accessibility.

REFERENCES

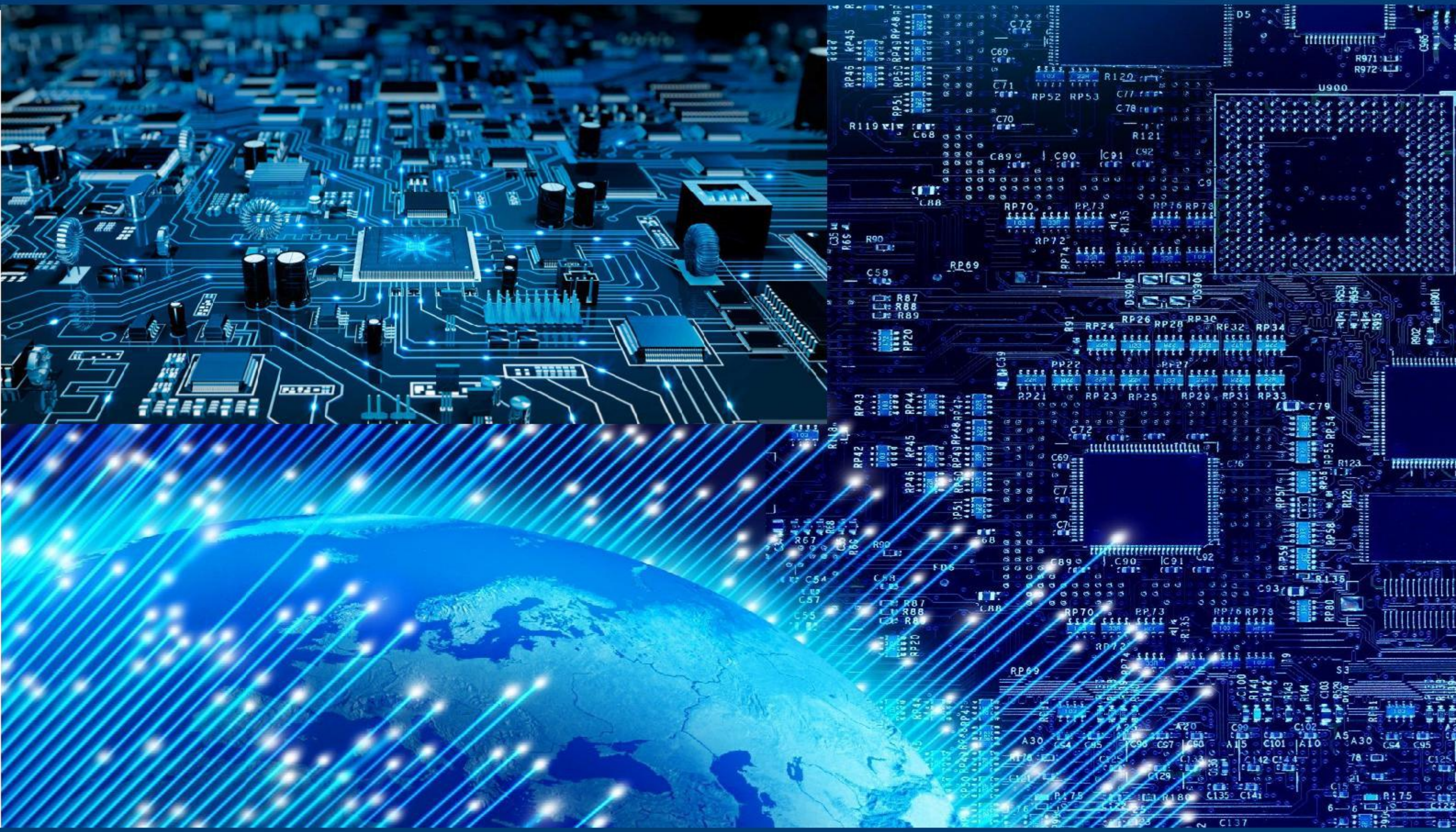
[1] C. Chakrabarti and G. F. Luger, “Artificial Conversations for Customer Service Chatterbots: Architecture, Algorithms, and Evaluation Metrics” Expert Systems with Applications, 2015.
[2] W. Maroengsit, T. Piyakulpinyo, K. Phonyiam, S. Pongnumkul, P. Chaovalit, T. Theeramunkong, “A Survey on Evaluation Methods for Chatbots” ACM International Conference Proceeding Series, 2019.
[3] J. WoolfAn, “Introduction to Chatbots” Josephwoolf.com, 2019. Online Source: An Introduction to Chatbots
[4] “What is a Chatbot and How to Use It for Your Business” Anadea, 2018 Online Source: Medium article
[5] B. A. Shawar and E. Atwell, “A Chatbot System as a Tool to Animate a Corpus” International Computer Archive Modern Medieval English Journal, 2005

| ISSN: 2582-7219 | www.ijmrset.com | Impact Factor: 7.521| Monthly Peer Reviewed & Referred Journal |

| Volume 7, Issue 4, April 2024 |

| DOI:10.15680/IJMRSET.2024.0704059 |

[6] Y. W. Chandra and S. Suyanto, "Indonesian Chatbot of University Admission Using a Question Answering System Based on Sequence-to-Sequence Model" Procedia Computer Science, 2019.
[7] R. Sutoyo, A. Chowanda, A. Kurniati, R. Wongso, "Designing an Emotionally Realistic Chatbot Framework to Enhance Its Believability with AIML and Information States" Published in: Procedia Computer Science, 2019.
[8] C. B. Ram Mohan, A. Babu Divi, A. Venkatesh, B. Sai Teja, "Chatbot for University Resource Booking" International Journal of Scientific Research in Computer Science, Engineering, and Information Technology, 2019.
[9] J. Singh, M. H. Joesph, K. Begum, A. Jabbar, "Rule-Based Chatbot for Student Inquiries" 2019.
[10] J. Porub, "Improving the User Experience of Electronic University Enrollment" 16th International Conference on Emerging eLearning Technologies, 2018.



INNO SPACE
SJIF Scientific Journal Impact Factor

INTERNATIONAL STANDARD SERIAL NUMBER INDIA

निस्केयर
NISCAIR