SOFTWARE REQUIREMENT SPECIFICATION (SRS)

Version 1.0

<<Annotated Version>>

October 15,2024

AI-Powered Student Assistance Chatbot for College

Guide : G Ranjith Kumar

Team Details:

Batch No : 4

Yenugu Smitha (B201586)
Kale Kawya (B201472)
Nidugonda Sravani (B201478)

Batch No : 77

Md Azhar (B200489)
Enugandhula Nilesh (B201579)
Jagadam Shiva kumar (B200578)

Table of contents

1 Introduction

Purpose of the document.

The purpose of this document is to define the functional and non-functional requirements for the AI-Powered Student Assistance Chatbot. It serves as a guide for developers, stakeholders, and users, outlining the chatbot's features, performance expectations, and design specifications.

1.1 Product scope

The AI-powered chatbot will provide real-time, automated responses to student queries regarding academic services such as admissions, fees, scholarships, placements, and course information. It aims to improve efficiency by reducing manual interactions and enhancing the overall student experience.

1.2 Product value

The chatbot will provide a 24/7 self-service solution to students, faculty, and staff, reducing the need for in-person or phone support. It will streamline access to critical information and enhance user experience with voice and text interactions.

1.3 Intended audience

- **Students:** To inquire about academic services and campus-related information.
- **Faculty:** To get administrative information and support.
- **Staff:** For managing student queries.
- **Administrators:** To monitor and maintain the system.

1.4 Intended use

Students and staff will use the chatbot to get instant answers to their academic queries. The system will be available on the college portal and mobile app, ensuring ease of access across multiple devices.

1.5 General description

The system will be built using Natural Language Processing (NLP) for understanding queries and generating responses. It will interface with the student information system and other databases to provide accurate and real-time information.

2 Functional requirements

1. Student Query Resolution:

- The chatbot must understand and respond to student queries related to admissions, fees, courses, placements, and other academic services.
- It should leverage Natural Language Processing (NLP) to interpret user questions accurately and generate relevant answers.
- Support for multilingual communication is essential, enabling students to interact in both English and regional languages.

2. Multimodal Interaction:

- The chatbot should offer responses in both text and voice formats, providing flexibility for users based on their interaction preference.
- Voice recognition and text-to-speech capabilities should be included to facilitate voice-based communication.

3. Document Analysis:

- The chatbot should allow students to upload documents (e.g., admission forms, academic records) in formats like PDF or image files.
- It will analyze these documents using AI-based extraction techniques and provide feedback regarding missing or incomplete information.

4. Real-time Data Integration:

- The chatbot must integrate with the college's internal databases to fetch accurate, real-time information for answering queries.
- Data should be kept up-to-date, ensuring that students always receive the most current information available.

5. Contextual and Personalized Responses:

- The chatbot should remember past interactions within a session to provide context-aware responses and follow-up on related questions.
- It should offer personalized responses where applicable, such as specific course details based on the user's program or preferences.

3 External interface requirements

3.1 User interface Requirements

- **Simplicity:** The chatbot should have a clean and straightforward user interface, allowing users to easily select either text-based or voice-based interaction.
- **Mobile-Friendliness:** The UI should be responsive, meaning it works seamlessly on desktops, laptops, smartphones, and tablets.
- **Integration:** The chatbot will be integrated into the college's web portal, making it easily accessible to students.

3.2 Hardware interface Requirements

- **Device Support:** The system will support common devices like desktops, laptops, smartphones, and tablets.
- **Voice Interaction:** For voice-based features, the system will require access to the device's microphone for users to ask questions verbally.

3.3 Software interface Requirements

- **Database Integration:** The chatbot will connect to the college's database to retrieve information such as course details, fee structures, and admission deadlines.
- **NLP Libraries:** The chatbot will use simple and accessible NLP libraries (e.g., Hugging Face, or pre-trained models) to understand user queries in natural language. You can start with basic NLP processing that is easy to implement with libraries you are comfortable using, like Python's spaCy or NLTK.

3.4 Communication Interface Requirements

- **API Communication:** The system will use RESTful APIs to communicate with the backend, retrieving and sending data (e.g., answering user queries).
- **Real-Time Responses:** While WebSockets can be useful for real-time responses, since you're focusing on simplicity, you can stick with AJAX or simple HTTP requests for now. These will work fine for most chatbot interactions without requiring the complexity of WebSockets.

4 Non-functional requirements

4.1 Security

The system will encrypt sensitive data using HTTPS for secure communication, ensuring student information is protected during interactions.

4.2 Capacity

The chatbot will support up to 1000 concurrent users, ensuring smooth performance even during high-traffic periods like admissions or exam schedules.

4.3 Compatibility

The chatbot will work across major web browsers (Chrome, Firefox, Safari) and mobile platforms (iOS, Android) for wide accessibility.

4.4 Reliability

The system will aim for 99.9% uptime, ensuring continuous availability for students to access information 24/7 without interruptions.

4.5 Scalability

The system will scale efficiently to accommodate increasing user numbers and future feature expansions without major rework.

4.6 Maintainability

The chatbot will have modular components, making it easy to update individual parts without disrupting the entire system.

4.7 Usability

The interface will be user-friendly, providing intuitive navigation with clear instructions, supporting text and voice interactions for all users.

4.8 Other

The system should support voice interaction in multiple languages, including English and Telugu.

5 Definitions and acronyms

- **AI:** Artificial Intelligence – The capability of a machine to imitate human intelligence, such as understanding language and responding to questions.

- **NLP:** Natural Language Processing – A field of AI that focuses on the interaction between computers and humans through natural language.

- **UI**: User Interface – The space where users interact with a system, typically through text, buttons, or voice commands.

- **API:** Application Programming Interface – A set of rules and tools for building software and allowing different systems to communicate with each other.

- **HTTPS:** Hypertext Transfer Protocol Secure – A secure version of HTTP used for encrypted communication over the internet.

- **REST:** Representational State Transfer – An architectural style for designing networked applications, often used for web APIs.

- **OTP:** One-Time Password – A security feature that provides a time-sensitive password used for secure authentication (not used in this project but common in chatbot systems).

- **Concurrency:** The ability of the system to handle multiple tasks or users at the same time.