**Software Design**

**Requirements Specification**

**for**

**Abbott & Sugar Project**

**Version 4**

**Prepared by:**

**Janis Garcia, Virginia Guadalupe Gonzalez, Ulises Gutierrez, James Hy, Charlie Martinez Dominguez, Michael Perez, Dylan Paul Tomasello, Omar Tovar, Theodore Tran, Thomas Yeung**

**Abbott & Shuger**

**September 23, 2023**

**Table of Contents**

Table of Contents................................................................................................................. pg. 2

Revision History................................................................................................................... pg. 3

1. Introduction................................................................................................................ pg. 4

1.1. Purpose........................................................................................................... pg. 4

1.2. Intended Audience and Reading Suggestions................................................ pg. 5

1.3. Product Scope................................................................................................ pg. 5

1.4. References...................................................................................................... pg. 7

2. Overall Description.................................................................................................... pg. 8

2.1. System Analysis…......................................................................................... pg. 8

2.2. Product Perspective........................................................................................... pg. 9

2.3. Product Functions........................................................................................... pg. 9

2.4. User Classes and Characteristics.................................................................... pg. 11

2.5. Operating Environment.................................................................................. pg. 14

2.6. Design and Implementation Constraints........................................................ pg. 15

2.7. User Documentation...................................................................................... pg. 16

2.8. Assumptions and Dependencies.................................................................... pg. 16

3. External Interface Requirements............................................................................... pg. 18

3.1. User Interfaces............................................................................................... pg. 18

3.2. Hardware Interfaces....................................................................................... pg. 18

3.3. Software Interfaces........................................................................................ pg. 19

3.4. Communications Interfaces........................................................................... pg. 21

4. Requirements Specification....................................................................................... pg. 23

4.1. Functional Requirements............................................................................... pg. 23

4.2. External Interface Requirements................................................................... pg. 23

4.3. Logical Database Requirements.................................................................... pg. 25

4.4. Design Constraints......................................................................................... pg. 27

5. Other Nonfunctional Requirements........................................................................... pg. 28

5.1. Performance Requirements............................................................................ pg. 28

5.2. Safety Requirements...................................................................................... pg. 28

5.3. Security Requirements................................................................................... pg. 28

6. Legal and Ethical Considerations.….......................................................................... pg. 30

Appendix A: Glossary........................................................................................................ pg. 31

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Date | Reason For Changes | Version |
| Janis Garcia | 9/3/2023 | Table of Contents/ Intro | 1 |
| Janis Garcia | 10/3/2023 | Overall Description/ External Interface Requirements | 2 |
| Janis Garcia | 11/3/2023 | Requirements Specification/ Other Nonfunctional Requirements | 3 |
| Janis Garcia | 12/7/2023 | Legal and Ethical Considerations/ Appendix/ Finishing touches/ last minute corrections | 4 |

**1. Introduction**

With all that we must worry about, we want to make the lives of university students like ourselves easier using AI. We wanted to use what we have learned in our time at Cal State LA. Our group may be a little different from the others in that we split our team into four smaller teams in order to tackle different strategies. Such as providing a chatbot as an alternative to an advisor, using databases and chaining prompts to figure out what your educational plan should look like, or creating a virtual education assistant to help you with your classes. Previously we had little experience with this AI so we spent the semester learning about language models and teaching them to each other so that next semester we can focus on a singular topic, a virtual assistant to help in different aspects of college life.

* 1. **Purpose**

The purpose of this Software Requirements Specification (SRS) document is to provide a comprehensive and detailed description of the intended functionality, features, and behavior of our software system. It will serve as a crucial document in the software development life cycle and acts as a foundation for the entire development process. We are on version 4. Our objectives of this SRS document include:

1. Communication Between Stakeholders:
   * Facilitate clear communication between various stakeholders, this would include developers, testers, and people in relation to the college or University.
2. Define System Functionality:
   * Clearly define the functional requirements of the software system, detailing what the system is expected to accomplish. This includes specific features, interactions, and overall system behavior.
3. Establish a Baseline for Development:
   * Provide a baseline or foundation for the development team by outlining the scope, objectives, and constraints of the project. This helps in managing expectations and setting the direction for the development process.
4. Aid in Project Management:
   * Assist project managers in planning and managing the software development process. The document helps in estimating timelines, allocating resources, and tracking progress against the defined requirements.
5. Facilitate Testing and Quality Assurance:
   * Provide a basis for creating test cases and conducting quality assurance activities. Testers refer to the SRS document to ensure that the developed software meets the specified requirements and functions as intended.
6. Support Maintenance and Upgrades:
   * Serve as documentation for future maintenance and upgrades. The document helps in understanding the existing system when modifications or enhancements are required, ensuring continuity in development efforts.
7. Risk Management:
   * Identify and address potential risks early in the development process. By clearly defining requirements, the SRS document allows stakeholders to anticipate challenges and plan mitigation strategies.

**1.2 Intended Audience and Reading Suggestions**

This report is intended for developers, university students, and by extension anyone in relation to Cal State LA or any other college or university, interested in implementing new strategies for improving university student lives. This SRS contains different strategies for improving students’ use of time using AI. It is not pertinent to read each section in order, each section is its own separate entity. Each section will be of different interest to different readers.

**1.3 Product Scope**

**1. Software Products Used:**

* **OpenAI:**
  + Utilized for powerful language models and natural language processing (NLP) capabilities.
* **Cohere:**
  + Integrated for efficient natural language processing (NLP) technologies, enhancing language understanding.
* **LLaMa by Meta:**
  + Leveraged for a universal language model that comprehends and generates text for various applications.
* **LangChain:**
  + Incorporated as a blockchain-based platform for advanced natural language processing and understanding.
* **Hugging Face:**
  + Utilized for a collaborative machine learning platform and access to pre-trained models for NLP tasks.
* **Colab (Google Colaboratory):**
  + Employed for a cloud-based platform with access to GPUs and TPUs, facilitating machine learning tasks.

**2. What the Software Will Do:**

* **Academic Assistance:** Provide academic guidance to students, assisting with course selection, degree planning, and academic resources.
* **Administrative Support:** Help students navigate administrative processes, including registration, enrollment, and accessing university services.
* **Information Retrieval:** Retrieve and provide accurate information on courses, campus events, and university policies.
* **Time Management:** Assist students in optimizing their time through scheduling reminders, deadlines, and study habit tips.
* **Problem Resolution:** Address common issues and queries faced by students, offering immediate solutions, or guiding them to relevant resources.
* **Integration with University Systems:** Seamlessly integrate with existing university systems and databases for real-time and accurate data retrieval.
* **User Engagement:** Foster engagement by creating a dynamic and interactive platform that encourages active participation and assistance-seeking.

**3. Benefits:**

* **Enhanced User Experience:** The virtual assistant provides a user-friendly and efficient interface, improving the overall experience for students.
* **Efficient Problem Resolution:** Quick and accurate responses aid in resolving issues and queries promptly, reducing user frustration.
* **Personalization:** Tailored responses based on individual user profiles, academic history, and preferences enhance the user experience.
* **Time Optimization:** Assistance in time management helps students optimize their schedules and deadlines, leading to better productivity.
* **Accessibility:** The virtual assistant caters to diverse learning preferences and needs, improving accessibility to critical university information.
* **Seamless Integration:** Integration with university systems ensures accuracy and relevance in the information provided.
* **Continuous Improvement:** Gathering user feedback allows for continuous improvement, ensuring the virtual assistant evolves to meet changing student needs.
* **Support for Decision Making:** The virtual assistant assists students in making informed academic decisions, improving overall academic performance.

We hope that by combining the capabilities of various software products, the virtual assistant will generate comprehensive solutions that address the diverse needs of university students, contributing to a more efficient, informed, and enjoyable educational experience.

**1.4 References**

1. “Building Generative AI Applications with Gradio.” *DeepLearning.AI*, www.deepLearning.ai/short-courses/building-generative-ai-applications-with-gradio/. Accessed 6 Dec. 2023.
2. “Generative AI for Everyone.” *DeepLearning.AI*, www.deeplearning.ai/courses/generative-ai-for-everyone/?utm\_campaign=genai4e-launch&utm\_medium=email&\_hsmi=280781425&\_hsenc=p2ANqtz-\_HjwSZ7YWPUM119AW7Wt3k8SVF8WZfPyI7dkRKCeXJ7zdoujKwLtKNQxbNLhZQZLFpIqBqWaf1SlAGoOFUmryueYeYfw&utm\_content=280781425&utm\_source=hs\_email. Accessed 6 Dec. 2023.
3. “How to Write Efficient Python Code: A Tutorial for Beginners.” *KDnuggets*, www.kdnuggets.com/how-to-write-efficient-python-code-a-tutorial-for-beginners. Accessed 6 Dec. 2023.
4. “Langchain Blog.” *LangChain Blog*, blog.langchain.dev/. Accessed 6 Dec. 2023.
5. LangChain. “Langchain Templates.” *LangChain Blog*, LangChain Blog, 10 Nov. 2023, blog.langchain.dev/langchain-templates/.
6. Muhammad Raza. (2023, November 19). *One liners python edition: Muhammad*. https://muhammadraza.me/2023/python-oneliners
7. *New Models and Developer Products Announced at DevDay*, openai.com/blog/new-models-and-developer-products-announced-at-devday. Accessed 6 Dec. 2023.
8. “Vector Databases: From Embeddings to Applications.” *DeepLearning.AI*, www.deepLearning.ai/short-courses/vector-databases-embeddings-applications/. Accessed 6 Dec. 2023.
9. “Welcome.” *Welcome - MindStudio Docs*, docs.youai.ai/youai/. Accessed 6 Dec. 2023.
10. Weng, Lilian. “Prompt Engineering.” *Lil’Log (Alt + H)*, 15 Mar. 2023, lilianweng.github.io/posts/2023-03-15-prompt-engineering/?ref=blog.langchain.dev%29.

**2. Overall Description**

**2.1 System Analysis**

**Goals**

**1.** Enhance the overall university experience for students by providing a user-friendly, efficient, and personalized virtual assistant/virtual assistant.

**2.** Offer academic guidance and support, aiding students in course selection, degree planning, and accessing relevant resources.

**3.** Streamline administrative processes by assisting students with tasks such as registration, enrollment, and accessing university services.

**4.** Facilitate quick and accurate retrieval of information on courses, campus events, and university policies.

**5.** Assist students in optimizing their time through scheduling reminders, deadlines, and providing study habit tips.

**6.** Address common issues and queries faced by students, offering immediate solutions, or guiding them to relevant resources.

**7.** Seamlessly integrate with existing university systems and databases for real-time and accurate data retrieval.

**8.** Foster user engagement by creating a dynamic and interactive platform that encourages active participation and assistance-seeking.

**Major Technical Hurdles**

**1. Natural Language Processing (NLP):**

* **Challenge:** Developing robust NLP capabilities to ensure the virtual assistant understands and responds to user queries in a natural and contextually relevant manner.
* **Solution:** Utilize advanced NLP libraries and models from OpenAI, Cohere, and other sources, and conduct extensive testing to refine language comprehension.

**2. Integration with University Systems:**

* **Challenge:** Integrating the virtual assistant with diverse university systems and databases to ensure accurate and up-to-date information retrieval.
* **Solution:** Collaborate closely with university IT departments, use standardized APIs where available, and implement data synchronization mechanisms for real-time updates.

**3. User Authentication and Authorization:**

* **Challenge:** Implementing secure user authentication and authorization mechanisms, especially when dealing with sensitive student information.
* **Solution:** Utilize industry-standard authentication protocols, such as OAuth, and ensure compliance with university security policies to safeguard user data.

**4. Handling Multiturn Conversations:**

* **Challenge:** Developing a dialog management system that retains context across multiple user interactions for coherent and meaningful conversations.
* **Solution:** Implement a dialog management system that can understand and manage multiturn conversations, ensuring a seamless and user-friendly experience.

**5. Continuous Improvement and Adaptability:**

* **Challenge:** Establishing mechanisms for gathering user feedback and adapting the virtual assistant to evolving user needs.
* **Solution:** Implement feedback loops within the system, regularly analyze user interactions, and employ agile development methodologies to iteratively improve and enhance features.

**6. Ethical Considerations and Bias:**

* **Challenge:** Addressing potential biases in AI models and ensuring ethical data usage.
* **Solution:** Regularly audit and update training data, implement fairness and bias detection tools, and adhere to ethical guidelines in AI development.

**2.2 Product Perspective**

The development of our virtual assistant software for university students is driven by a specific focus on addressing the unique needs and challenges faced by students in their academic journey. While other educational platforms and virtual assistants exist, our motivation lies in providing a tailored solution that caters specifically to the academic requirements and lifestyle of university students. Unlike generic virtual assistants, our software is designed with features that offer targeted academic support. This includes functionalities such as course scheduling and personalized study recommendations. The motivation is to streamline and enhance the academic experience for university students.

**2.3 Product Functions**

1. **User Engagement:**
   * **Function:** Initiate and maintain interactive and engaging conversations with users.
   * **Purpose:** Enhance user interaction and foster a positive user experience.
2. **Natural Language Understanding:**
   * **Function:** Comprehend and interpret natural language input from users.
   * **Purpose:** Enable the virtual assistant to understand user queries, commands, and statements in a human-like manner.
3. **Information Retrieval:**
   * **Function:** Retrieve relevant information from databases or external sources based on user queries.
   * **Purpose:** Provide accurate and timely information on various topics, such as courses, events, and policies.
4. **Task Execution:**
   * **Function:** Perform specific tasks or actions based on user requests.
   * **Purpose:** Assist users in completing tasks, such as setting reminders, sending notifications, or accessing specific services.
5. **User Assistance and Support:**
   * **Function:** Address user inquiries, provide guidance, and offer solutions to common problems.
   * **Purpose:** Support users in navigating processes, overcoming challenges, and accessing resources.
6. **Multiturn Conversations:**
   * **Function:** Manage and maintain context across multiple interactions in a single conversation.
   * **Purpose:** Enable coherent and meaningful conversations by remembering previous user inputs and responses.
7. **User Authentication and Authorization:**
   * **Function:** Verify user identities and grant appropriate access permissions.
   * **Purpose:** Ensure data security, privacy compliance, and personalized interactions.
8. **Integration with External Systems:**
   * **Function:** Connect with external systems, databases, or APIs to fetch real-time information or perform actions.
   * **Purpose:** Enhance functionality by tapping into external resources and providing dynamic responses.
9. **User Personalization:**
   * **Function:** Tailor responses and recommendations based on user profiles, preferences, and historical interactions.
   * **Purpose:** Improve the user experience by delivering personalized and relevant information.
10. **Feedback Collection:**
    * **Function:** Gather user feedback on the virtual assistant's performance and user satisfaction.
    * **Purpose:** Facilitate continuous improvement by understanding user preferences and addressing concerns.
11. **Error Handling:**
    * **Function:** Detect and handle errors or misunderstandings in user input.
    * **Purpose:** Ensure a smooth user experience by providing informative error messages or seeking clarification.
12. **Contextual Understanding:**
    * **Function:** Understand the context of user queries and responses to maintain coherent conversations.
    * **Purpose:** Enhance the virtual assistant's ability to respond appropriately based on the ongoing conversation.
13. **Scheduling and Reminders:**
    * **Function:** Assist users in managing their time by setting reminders and scheduling events.
    * **Purpose:** Improve time management and help users stay organized.
14. **Security Measures:**
    * **Function:** Implement security protocols to protect user data and ensure secure interactions.
    * **Purpose:** Safeguard user information and maintain the integrity of the virtual assistant system.
15. **Learning and Adaptation:**
    * **Function:** Continuously learn from user interactions and adapt to evolving user needs.
    * **Purpose:** Enhance the virtual assistant's capabilities and responsiveness over time.

**2.4 User Classes and Characteristics**

1. **University Students (main users):**
   * **Characteristics:**
     + Frequent users, interacting with the virtual assistant for academic guidance, course information, and administrative support.
     + Varied technical expertise, ranging from tech-savvy students to those less familiar with AI interfaces.
     + Diverse educational levels, including undergraduate and graduate students.
   * **Requirements:**
     + Access to personalized academic information.
     + Assistance with course planning and scheduling.
     + Support for administrative processes like enrollment and registration.
2. **University Faculty and Staff:**
   * **Characteristics:**
     + Moderate to frequent users, seeking assistance with administrative tasks or accessing information relevant to their roles.
     + Higher technical expertise compared to students, with varied educational levels.
   * **Requirements:**
     + Access to faculty-specific information and resources.
     + Assistance with administrative processes related to teaching and research.
3. **AI Researchers and Developers:**
   * **Characteristics:**
     + Infrequent users, exploring the virtual assistant's capabilities for research purposes.
     + High technical expertise in AI and programming.
   * **Requirements:**
     + Access to advanced features for experimentation.
     + Integration options with AI development tools.
4. **IT Administrators:**
   * **Characteristics:**
     + Infrequent but critical users, overseeing the integration, maintenance, and security aspects of the virtual assistant.
     + High technical expertise in system administration and security.
   * **Requirements:**
     + Access to system logs and security-related features.
     + Documentation for system integration and troubleshooting.
5. **University Administrators:**
   * **Characteristics:**
     + Infrequent users, utilizing the virtual assistant for high-level administrative queries and data retrieval.
     + Moderate technical expertise, with a focus on administrative processes.
   * **Requirements:**
     + Access to institutional data and reports.
     + Efficient handling of administrative inquiries.
6. **Prospective Students:**
   * **Characteristics:**
     + Infrequent users, exploring the university's offerings and seeking information about admission.
     + Varied technical expertise, with a focus on ease of use.
   * **Requirements:**
     + Access to general university information.
     + Guidance on the admission process.
7. **Parents or Guardians:**
   * **Characteristics:**
     + Infrequent users, interested in monitoring academic progress or accessing general information about their words.
     + Varied technical expertise, with an emphasis on user-friendly interactions.
   * **Requirements:**
     + Access to academic updates and general university information.
     + User-friendly interface for non-technical users.

**2.5 Operating Environment**

1. **Hardware Platform:**
   * **Description:** The virtual assistant will primarily operate in a cloud-based environment, utilizing the infrastructure provided by cloud service providers.
   * **Reasoning:** Cloud platforms offer scalability, flexibility, and accessibility, allowing the virtual assistant to handle variable workloads and ensuring availability.
2. **Operating System:**
   * **Description:** The virtual assistant's application and server components will be designed to operate independently of the end-users' operating systems.
   * **Reasoning:** This approach ensures compatibility with various devices and operating systems used by university students, faculty, and administrators.
3. **Development and Runtime Environment:**
   * **Description:** The development environment for building and testing the virtual assistant may involve Colab or other cloud-based development tools.
   * **Reasoning:** Cloud-based development facilitates collaboration among team members, and the runtime environment ensures consistent performance and availability.
4. **Programming Languages and Frameworks:**
   * **Description:** The virtual assistant may be developed using programming languages like Python and may leverage frameworks such as TensorFlow or PyTorch for machine learning and natural language processing tasks.
   * **Reasoning:** Python is widely used in AI and NLP development, and frameworks provide pre-built modules for efficient development.
5. **Database Systems:**
   * **Description:** The virtual assistant may interact with databases for information retrieval. Common database systems like MySQL, PostgreSQL, or cloud-based databases may be utilized.
   * **Reasoning:** Databases store and retrieve information, allowing the virtual assistant to provide accurate and up-to-date responses to user queries.
6. **External APIs and Services:**
   * **Description:** The virtual assistant may integrate with external APIs such as OpenAI, cohere, LLaMa by Meta, LangChain, and Hugging Face for advanced language processing and AI capabilities.
   * **Reasoning:** Leveraging external APIs enhances the virtual assistant's functionality and access to advanced language models.
7. **Web Interface and Integration:**
   * **Description:** The virtual assistant may have a web interface for user interactions and could integrate with university websites or portals for seamless access to institutional information.
   * **Reasoning:** A web interface provides a user-friendly interaction medium, while integration with university systems ensures real-time and accurate data retrieval.

**2.6 Design and Implementation Constraints**

**Regulatory Policies:** Adherence to university or organizational policies regarding data privacy, security, and ethical use of AI technologies. Development processes must align with established policies, requiring transparent data handling, secure authentication, and ethical considerations in the use of AI.

**Hardware Limitations:** Constraints related to the computational power and resources available on the selected cloud platform or server infrastructure. Developers must optimize code, manage memory efficiently, and design algorithms that operate within the hardware limitations to ensure optimal performance.

**Interfaces with Other Applications:** Integration with university databases, authentication systems, and external APIs for enhanced functionality. Developers need to design robust interfaces, handle data exchange seamlessly, and ensure compatibility with various external systems.

**Parallel Operation:** Consideration of concurrent user interactions and simultaneous processing of multiple requests. Developers need to implement parallel processing mechanisms to handle multiple user queries concurrently and maintain system responsiveness.

**Higher-Order Language Requirements:** Use of advanced language processing and understanding techniques to improve the virtual assistant's ability to comprehend user queries. Developers must be proficient in higher-order language processing technologies and integrate them effectively into the virtual assistant's architecture.

**Reliability Requirements:** Expectations for the virtual assistant to operate reliably, provide accurate information, and handle user queries without system failures. Developers need to implement robust error handling, conduct thorough testing, and ensure the virtual assistant's reliability under various usage scenarios.

**Safety and Security Considerations:** Implementation of security measures to safeguard user data, prevent unauthorized access, and protect against potential security threats. Developers must prioritize security features, implement encryption protocols, and conduct regular security audits to ensure the safety of user information.

**Memory Constraints:** Limitations on the amount of memory available for data storage and processing. Developers need to optimize memory usage, manage data storage efficiently, and handle large datasets without exceeding memory constraints.

**Scalability Requirements:** The need for the virtual assistant to scale efficiently to handle increased user demand and growing data volumes. Developers must design the architecture to scale horizontally or vertically, considering factors such as load balancing and resource allocation.

**Compliance with Accessibility Standards:** Ensuring the virtual assistant is accessible to users with disabilities and complies with accessibility standards. Developers need to incorporate accessible design principles, provide alternative input methods, and test for compliance with accessibility guidelines.

**2.7 User Documentation**

A straightforward prompt upon launching the virtual assistant, guiding users on its functionalities, should serve as sufficient user documentation.

**2.8 Assumptions and Dependencies**

1. **Internet Connection:**
   * **Dependency:** A stable internet connection is required to access the model.
   * **Consideration:** Offline usage may not be possible, and any interruptions in internet connectivity can impact the ability to interact with the model.
2. **Availability:**
   * **Dependency:** Continued availability and support of the model.
   * **Consideration:** Changes in service, versions, or policies could impact the integration and functionality.
3. **API Rate Limits:**
   * **Dependency:** Adherence to rate limits.
   * **Consideration:** Frequent requests beyond the rate limits may result in service disruptions or additional costs.
4. **Model Updates:**
   * **Dependency:** Updates or new versions of the model.
   * **Consideration:** Changes in the model may affect the behavior and responses, and users need to adapt to new versions as necessary.
5. **Token Limits:**
   * **Dependency:** The model has a maximum token limit for each interaction.
   * **Consideration:** Long conversations may be truncated, and the need to manage interactions within the token limits.
6. **Costs:**
   * **Dependency:** Usage of the API is subject to associated costs.
   * **Consideration:** The need to be aware of pricing models, potential usage costs, and stay within allocated budgets.

**3. External Interface Requirements**

* 1. **User Interfaces**

The virtual assistant will have a user-friendly GUI featuring interactive elements for user input and displaying responses. The GUI will be accessible through standard web browsers, ensuring compatibility across most devices. The web-based interface allows users to interact with the virtual assistant from various platforms, including desktops, laptops, tablets, and smartphones. The interface will utilize responsive design, adapting to different screen sizes and resolutions. This ensures a consistent and optimized user experience across a variety of devices. A uniform layout across screens for a predictable user experience. Standardized navigation elements, such as headers and footers, will be maintained to enhance usability. The interface will conform to the accessibility standards outlined by the Americans with Disabilities Act (ADA). This includes features such as alternative text for images, keyboard navigation, and compatibility with screen. Clear and concise error messages indicating the nature of the issue and offering guidance for resolution. The design follows accessibility guidelines to ensure readability. Inclusion of progress indicators for tasks involving processing or data retrieval. Users will be informed about the system's status during such operations. These characteristics contribute to a user interface that is user-friendly and accessible.

**3.2 Hardware Interfaces**

1. **Supported Device Types:**
   * **Logical Description:** The virtual assistant software will support various device types, including desktop computers, laptops, tablets, and smartphones.
   * **Physical Implementation:** The software will be accessible through standard web browsers on these devices, ensuring a versatile and consistent user experience.
2. **Data and Control Interactions:**
   * **Logical Description:** The software will primarily involve data interactions, receiving user inputs in the form of text commands and providing responses.
   * **Physical Implementation:** Integration with device-specific APIs for capturing text input. Bi-directional communication for processing user commands and delivering responses.
3. **Communication Protocols:**
   * **Logical Description:** Standard communication protocols, such as HTTP/HTTPS, will be utilized for data exchange between the software and hardware components.
   * **Physical Implementation:** Integration with APIs, SDKs, or middleware that support HTTP/HTTPS for communication. Ensuring secure and efficient data transfer.
4. **Text Input Devices:**
   * **Logical Description:** The software will support text input devices such as keyboards for user interactions.
   * **Physical Implementation:** Standard integration with keyboard inputs. Providing a seamless text input experience for users.
5. **Network Infrastructure:**
   * **Logical Description:** The software relies on network infrastructure for communication between devices and servers.
   * **Physical Implementation:** Utilizing standard networking protocols to establish connections. Ensuring compatibility with different network configurations.

**3.3 Software Interfaces**

1. **OpenAI:**
   * **Name:** OpenAI
   * **Version Number:** 1.3.7
   * **Source:** OpenAI official website
   * **API Access:** Utilizing the OpenAI API for natural language processing tasks. The virtual assistant will send user queries to the OpenAI API, which will return responses generated by language models.
2. **Cohere:**
   * **Name:** Cohere
   * **Version Number:** 4.3.7
   * **Source:** Cohere official website
   * **API Access:** Integrating Cohere's natural language processing models through APIs. The virtual assistant will interact with Cohere's API to enhance language understanding and capabilities.
3. **LLaMa by Meta:**
   * **Name:** LLaMa (Language Model for Many Applications)
   * **Version Number:** LLaMA-2
   * **Source:** Meta official website
   * **API Access:** If available, integrating with LLaMa's API for advanced language understanding and generation. This may involve sending and receiving data through API calls.
4. **LangChain:**
   * **Name:** LangChain
   * **Version Number:** langchain 0.0.348
   * **Source:** LangChain official channels
   * **API Access:** Integrating with LangChain's blockchain-based platform for natural language processing. Communication with LangChain may involve utilizing APIs for language-related applications.
5. **Hugging Face:**
   * **Name:** Hugging Face
   * **Version Number:** v4.35.0
   * **Source:** Hugging Face official channels
   * **API Access:** Utilizing Hugging Face's library and model hub for access to pre-trained models, including those for natural language processing. Interaction with Hugging Face may involve API calls for model sharing and collaboration.
6. **Colab (Google Colaboratory):**
   * **Name:** Google Colaboratory
   * **Version Number:** N/A
   * **Source:** Google Colab official channels
   * **API Access:** Using Colab as a cloud-based platform for developing and running Jupyter notebooks. The virtual assistant may leverage GPUs and TPUs for machine learning tasks through Colab.
7. **MySQL Database Server:**

* **Name:** MySQL Database Server
* **Version Number:** 8.0
* **Source:** The official MySQL website: https://www.mysql.com/
* **API Access:** Utilizing MySQL as a relational database server for data storage and retrieval. The virtual assistant may interact with MySQL databases to store, retrieve, and manipulate data.

**API Communication and Information Passing:**

* **Request-Response Mechanism:** The virtual assistant will send requests to external APIs (such as OpenAI, Cohere, LLaMa, LangChain, Hugging Face) through HTTP/HTTPS protocols.
* **Data Exchange Formats:** Information passed between the virtual assistant and external software will be in standard data exchange formats such as JSON. The virtual assistant will construct requests, send them to the respective APIs, receive responses in JSON format, and parse the information for appropriate use.
* **Authentication and Security:** API calls will be secured using authentication mechanisms provided by each API. Secure communication protocols (HTTPS) will be implemented to ensure the confidentiality and integrity of data exchanged between the virtual assistant and external software products.
* **Error Handling:** The virtual assistant will implement robust error-handling mechanisms to manage scenarios where API calls fail or return unexpected responses. This includes providing informative error messages to users when necessary.

**3.4 Communications Interfaces**

1. **User Input via Web Browser:**
   * **Requirement:** The virtual assistant must support user interactions initiated through standard web browsers on various devices.
   * **Message Formatting:** Input messages will be in plain text or voice commands, depending on the user's preference.
2. **API Communications:**
   * **Requirement:** The virtual assistant will communicate with external APIs, including OpenAI, Cohere, LLaMa, LangChain, and Hugging Face.
   * **Message Formatting:** Requests and responses will be formatted in JSON for consistency and ease of parsing.
3. **HTTP/HTTPS Protocols:**
   * **Requirement:** All communication between the virtual assistant and external servers or APIs will use HTTP/HTTPS protocols.
   * **Communication Standards:** Adherence to HTTP/HTTPS standards ensures secure and reliable data transfer.
4. **Data Transfer Rates:**
   * **Requirement:** Communication between the virtual assistant and external APIs should have minimal latency for a responsive user experience.
   * **Specification:** Data transfer rates should be optimized to minimize delays in processing user requests and delivering responses.
5. **Security and Encryption:**
   * Requirement: All communication channels, especially those involving user data and external APIs, must be secured.
   * Encryption Standards: HTTPS will be enforced to encrypt data in transit, ensuring the confidentiality and integrity of information exchanged.
6. **Error Handling and Feedback:**
   * **Requirement:** The virtual assistant must handle communication errors gracefully and provide informative feedback to users.
   * **Approach:** Clear error messages will be displayed, guiding users on how to proceed in case of communication failures or unexpected issues.
7. **User Authentication and Authorization:**
   * **Requirement:** If the virtual assistant involves user-specific information, it must implement secure user authentication and authorization.
   * **Standards:** Utilizing industry-standard authentication protocols to ensure the confidentiality and privacy of user data.
8. **Compliance with Privacy Regulations:**
   * **Requirement:** The virtual assistant must adhere to privacy regulations and data protection laws.
   * **Implementation:** Ensuring that user data is handled in accordance with relevant privacy standards, such as GDPR or regional data protection regulations.

**4. Requirements Specification**

* 1. **Functional Requirements**
* **Validity Checks on Inputs:** The system shall perform thorough validity checks on all inputs received, ensuring accuracy and relevance.
* **Exact Sequence of Operations:** The system shall define a precise sequence of operations to be executed, outlining the step-by-step processes to achieve specific tasks.
* **Responses to Abnormal Situations:** The system shall exhibit appropriate responses to abnormal situations, encompassing scenarios such as overflow, communication facility issues, and comprehensive error handling and recovery mechanisms.
* **Effect of Parameters:** The system shall account for the impact of parameters on its operations, ensuring that varying parameters result in expected and accurate outcomes.
* **Relationship of Outputs to Inputs:** The system shall establish a clear relationship between outputs and inputs, encompassing input/output sequences and specifying formulas for input to output conversion.
* **Sub-function Partitioning:** It may be suitable to further partition functional requirements into sub-functions or sub-processes, allowing for a more detailed breakdown of specific tasks. However, this partitioning does not inherently dictate the structure of the software design.

**4.2 External Interface Requirements**

* **Name of item:**
  + - User input interface
    - Database interface
    - Chatbot output interface
* **Description of purpose:**
  + - User input interface - To receive questions and queries from users.
    - Database interface - For accessing CSULA course details, department information, schedules, and calendars (do we?).
    - Chatbot output interface - The chatbot responses generated in response to user inputs.
* **Source of input or destination of output:**
  + - User input interface - Input from users.
    - Database interface - Input from the system’s database, output as data reference in chained prompt instruction to the LLM.
    - Chatbot output interface - Displayed to the users on the chatbot interface as system responses.
* **Valid range, accuracy and/or tolerance:**
  + - User input interface - Accepts natural language in the English language.
    - Database interface
    - Relational database - Must handle structured query language with high accuracy.
    - Vector database - Vector search using cosine similarity with a score higher than 0.75.
    - Chatbot output interface - Variable based on the context of the interaction, providing relevant course details, or campus information.
* **Units of measure:**
  + - User input interface - Measured in terms of response time (milliseconds).
    - Database interface - Data throughput measured in Mbps and recalls in queries per second.
    - Chatbot output interface - N/A
* **Timing:**
  + - User input interface - Real-time processing of users’ inputs.
    - Database interface - Scheduled backups, real-time data queries.
* **Relationships to other inputs/outputs:**
  + - User input interface - Interacts with the chatbot engine to generate appropriate responses. May trigger backend processes for more complex queries.
    - Database interface - Relies on user inputs for data queries, or similarity search. Query results are sent to the model for final completion as output back to the user.
    - Chatbot output interface - Connected to the input interactions from users, influenced by the chatbot engine’s processing.
* **Screen formats/organization:**
  + - Responsive design compatible with various screen sizes.
    - Intuitive layout for ease of navigation.
    - Accessibility features.
* **Window formats/organization:**
  + - Modular window design for easy customization.
    - Consistent style across different modules for a unified look.
    - Pop-up notifications for important alerts.
* **Data formats:**
  + - JSON for data interchange.
    - SQL for database queries.
* **Command formats:**
  + - Text commands through chat interface.
* **End messages:**
  + - Confirmation messages upon task completion.
    - Error messages without troubleshooting tips.
    - Feedback prompts for user satisfaction and improvement.

**4.3 Logical Database Requirements**

1. **Types of Information Used by Various Functions:**
   * *Student Information:* Personal details, academic records, enrollment status, and course history.
   * *Course Information:* Details about courses, curriculum, prerequisites, and scheduling.
   * *Administrative Information:* University policies, announcements, and administrative procedures.
   * *User Authentication Data:* Secure storage of user credentials and access permissions.
2. **Frequency of Use:**
   * Frequent access: Student information, course details, and administrative procedures.
   * Moderate access: User authentication data and less frequently accessed administrative policies.
   * Infrequent access: Historical data and archived information.
3. **Accessing Capabilities:**
   * *Student Access:* Read access to personal academic records and course information.
   * *Administrative Access:* Write access for authorized personnel to update policies and announcements.
   * *User Authentication Access:* Highly restricted access with a focus on encryption and security protocols.
4. **Data Entities and Their Relationships:**
   * *Entities:* Students, Courses, Instructors, Administrative Policies.
   * *Relationships:* Student enrollment in courses, Instructor assignment to courses, Dependencies between policies and student interactions.
5. **Integrity Constraints:**
   * *Academic Integrity:* Ensuring data accuracy in student records and grades.
   * *Referential Integrity:* Maintaining consistency in relationships, e.g., linking student records with enrolled courses.
   * *Security Constraints:* Enforcing access controls and encryption mechanisms to protect sensitive information.
6. **Data Retention Requirements:**
   * *Student Records:* Retained for the duration of enrollment and a defined period after graduation.
   * *Course Information:* Kept for historical analysis and academic planning purposes.
   * *Administrative Data:* Retained based on legal and institutional requirements.
7. **Security and Encryption:**
   * *Data Encryption:* Use of encryption algorithms to safeguard sensitive information during transmission and storage.
   * *Access Controls:* Role-based access controls to ensure authorized access to different types of data.
   * *Audit Trails:* Logging mechanisms to track changes, access, and system activities for accountability.
8. **Scalability:**
   * *Handling Growth:* Database design to accommodate an increasing volume of student and course data over time.
   * *Performance Optimization:* Regular optimization to ensure efficient data retrieval, especially during peak usage periods.
   1. **Design Constraints**

The virtual assistant's design must accommodate the hardware constraints of the target devices. This includes considerations for processing power, memory, and storage capacity. The software should be optimized to run efficiently within these limitations. The virtual assistant must be designed to work seamlessly across various platforms, considering differences in system architectures and dependencies.Constraints related to network conditions must also be considered, like internet connectivity. The virtual assistant should be designed to operate efficiently under different network speeds and potential connectivity disruptions. Design constraints may be imposed by accessibility standards to ensure that the virtual assistant is usable by individuals with disabilities. Compliance with accessibility guidelines such as WCAG (Web Content Accessibility Guidelines) may be necessary. The virtual assistant must adhere to relevant legal and regulatory standards, especially those pertaining to data privacy and protection. The design must account for scalability requirements to accommodate potential growth in user base and feature enhancements. The virtual assistant should be able to scale efficiently without compromising performance. Constraints may arise from third-party services or APIs integrated into the virtual assistant. The design should consider the limitations and specifications of these external components to ensure seamless interoperability. Considering and addressing these design constraints is crucial for ensuring the successful development and deployment of the virtual assistant, meeting both technical and regulatory requirements.

**5. Other Nonfunctional Requirements**

**5.1 Performance Requirements**

**Status:** To Be Determined (TBD) We have yet to determine the number of transactions, tasks, and amount of data can be processed and in what amount of time.

**5.2 Safety Requirements**

**Data Privacy and Protection:**

* Requirement: The virtual assistant must ensure the privacy and protection of user data to prevent unauthorized access, loss, or misuse.
* Safeguards: Implement robust encryption mechanisms, secure user authentication, and adhere to data protection regulations such as GDPR.
* Actions to Prevent: Unauthorized access, data breaches, and any form of data manipulation must be prevented through rigorous security measures.

. **User Interaction Safety:**

* Requirement: Ensure the virtual assistant's interface is designed with safety in mind to prevent user confusion or unintentional errors.
* Safeguards: Provide clear instructions, error messages, and user prompts. Implement intuitive design principles.
* Actions to Prevent: Mitigate the risk of user errors or confusion through user-friendly design and informative feedback.

**Safety Certifications:**

* Requirement: Obtain and maintain necessary safety certifications relevant to the virtual assistant's domain.
* Safeguards: Follow industry-specific safety standards and certification processes.
* Actions to Prevent: Ensure that the virtual assistant meets all safety certification requirements to guarantee its suitability for use.

**5.3 Security Requirements**

1. **User Authentication Security:**
   * Requirement: Ensure secure user authentication to prevent unauthorized access to sensitive information.
   * Safeguards: Implement strong password policies, two-factor authentication, and periodic security audits.
   * Actions to Prevent: Unauthorized access to user accounts and the potential compromise of personal information must be prevented through stringent authentication measures.
2. **Network Security:**
   * Requirement: Protect against potential network vulnerabilities to prevent data interception or unauthorized access during data transmission.
   * Safeguards: Use secure communication protocols (e.g., HTTPS), employ firewalls, and regularly update security measures.
   * Actions to Prevent: Mitigate the risk of data interception, unauthorized access, or network attacks through proactive security measures.

**6. Legal and Ethical Considerations**

The virtual assistant has to navigate various legal and ethical considerations. When it comes to Data Privacy and Protection, handling sensitive student information requires following data protection laws like GDPR. User Authentication and Authorization involve securing user-specific information, aligning with legal requirements to prevent unauthorized access and ethically maintaining user privacy and the integrity of educational data. Legal compliance is crucial for Privacy Regulations, ensuring the avoidance of legal consequences and showcasing a commitment to ethical user privacy rights. Pursuing Safety Certifications is seen as an ethical responsibility to meet industry-specific standards, emphasizing accountability, and ensuring user safety. Adherence to Accessibility Standards is both a legal mandate and an ethical responsibility, promoting inclusivity and equal access to educational resources. Network Security, a legal necessity, prevents data breaches and unauthorized access, aligning with ethical principles to protect users and communication channels. Ethical Use of AI involves transparent communication, avoiding bias, and responsible AI practices, fostering user trust, and aligning with fairness and accountability principles. Ethical Communication, vital for user understanding and trust, includes clear instructions and feedback, aligning with transparency and user-centric design principles.

**Appendix A: Glossary**

1. **API (Application Programming Interface):** A set of protocols and tools for building software applications, facilitating communication between different software systems.
2. **Accessibility Standards:** Guidelines and requirements ensuring that digital products and services are accessible to people with disabilities.
3. **Assumptions:** Statements considered to be true without direct evidence, forming the basis for project planning and decision-making.
4. **Baseline:** A reference point that outlines the scope, objectives, and constraints of a project, providing a foundation for development and management.
5. **Cloud-Based Environment:** An operational environment where computing resources, including storage and processing power, are hosted on remote servers accessed via the internet.
6. **Compliance:** Adherence to established standards, regulations, and policies relevant to the development and use of a software system.
7. **Costs:** The financial expenses associated with the usage of software, services, or APIs, including considerations of budgeting and pricing models.
8. **Data Privacy:** The protection of sensitive and personal information from unauthorized access, **use, or disclosure.**
9. **Dependencies:** Relationships or connections between different components or aspects of a project, where changes in one may affect others.
10. **Ethical Considerations:** The examination of moral principles and standards in the development and use of technology, ensuring responsible and fair practices.
11. **Machine Learning:** A subset of artificial intelligence that enables systems to learn and improve from experience without explicit programming.
12. **Memory Optimization:** The efficient use of computer memory to enhance system performance and prevent resource exhaustion.
13. **Natural Language Processing (NLP):** A field of artificial intelligence focused on enabling computers to understand, interpret, and respond to human language.
14. **Parallel Processing:** The simultaneous execution of multiple tasks or processes to improve efficiency and system performance.
15. **Project Management:** The process of planning, organizing, and overseeing the execution of a project, including estimating timelines, allocating resources, and tracking progress.
16. **Quality Assurance:** Activities and processes implemented to ensure that a product or service meets specified requirements and standards.
17. **Reliability:** The ability of a system to consistently perform its intended functions with accuracy and dependability.
18. **Security Protocols:** Procedures and measures implemented to safeguard data, prevent unauthorized access, and protect against security threats.
19. **Software Requirements Specification (SRS):** A comprehensive document outlining the intended functionality, features, and behavior of a software system, serving as a foundation for the development process.
20. **Stakeholders:** Individuals or groups with an interest or involvement in the software development process, including developers, testers, and university personnel.
21. **Scalability:** The capability of a system to handle increasing workloads or demands by adapting and expanding its resources.
22. **Token Limits:** Maximum limits on the number of tokens (units of text) allowed in a single interaction or conversation.
23. **User Experience (UX):** The overall experience a user has while interacting with a product or system, encompassing usability, accessibility, and satisfaction.
24. **User Feedback:** Information provided by users about their experience, preferences, and opinions regarding a product or service.
25. **User Interface (UI):** The point of interaction between a user and a computer system, including screens, pages, and visual elements.