Software Requirements Specification

for

Requirement Sage Chatbot

Version 1.0 approved

Date: November 30th, 2023

Table of Contents

Revision History

Name	Date	Reason for Changes	Version

1. Introduction

1.1 Purpose

The purpose of the RequirementSage chatbot is to automate the software requirements-gathering process for small-scale projects, with a specific focus on e-commerce websites. The product aims to revolutionize the traditional and manual methods of requirements collection by leveraging advanced Natural Language Processing (NLP) and Machine Learning (ML) technologies.

The purpose of RequirementSage is to address the following key objectives and business goals:

1.1.1 Automate Requirements Gathering

Streamline and automate the collection of software requirements to eliminate manual and time-consuming processes.

1.1.2 Improve Communication

Facilitate clear and effective communication between end users and software developers through natural language interactions.

1.1.3 Enhance Accuracy

Utilize Machine Learning to enhance requirement understanding, reducing the risk of miscommunication or incomplete requirements.

1.1.4 Learn and Adapt

Continuously learn from interactions to provide more accurate and relevant requirements gathering over time.

1.1.5 Ensure Data Security

Implement robust security measures to protect sensitive project information and maintain compliance with data privacy regulations.

1.2 Document Convention

1.2.1 Priority Levels

- **1.2.1.1 High Priority (HP)**

  Requirements that are critical for the core functionality and success of the chatbot.

- **1.2.1.2 Medium Priority (MP)**

  Important requirements that enhance the user experience and efficiency.

- **1.2.1.3 Low Priority (LP)**

  Desirable but non-essential features that can be considered for future enhancements.

1.2.2 Text Formatting (Bold Text)

Used for emphasizing key points and section headings. - *Italic Text:* Utilized for providing additional context or examples.

1.2.3 Naming Conventions

Acronyms and abbreviations are written in all capital letters (e.g., SRD for Software Requirement Document).

1.2.4 Document Sections

- **1.2.4.1 Functional Requirements (F)**

  Describes specific features and capabilities of RequirementSage.

- **1.2.4.2 Non-Functional Requirements (NF)**

  Outlines performance, security, and usability aspects.

- **1.2.4.3 Business Requirements (B)**

  Focuses on the broader business goals and considerations.

1.2.5 Requirement ID Format

Each requirement is identified by a unique alphanumeric code, starting with the category abbreviation (e.g., F-1.1.1 for a functional requirement).

1.2.6 Language Usage

The document uses clear and concise language, avoiding jargon whenever possible to ensure accessibility for all stakeholders.

1.2.7 Dependencies

Dependencies between requirements are indicated, specifying any prerequisites or relationships.

1.2.8 Assumptions

Assumptions made during the development process are clearly stated to provide context for certain requirements.

1.3 Intended Audience and Reading Suggestions

1.3.1 Intended Audience

The Software Requirements Specification (SRS) for RequirementSage is intended for a diverse audience involved in the development, implementation, and oversight of the chatbot project. The primary audience includes:

- **1.3.1.1 Developers**

  Those responsible for implementing the software, coding the chatbot functionalities, and integrating the NLP and ML technologies.

- **1.3.1.2 Project Managers**

  Individuals overseeing the project's planning, execution, and delivery. This includes those responsible for resource allocation, timeline management, and overall project coordination.

- **1.3.1.3 Quality Assurance/Testers**

  Testers responsible for validating the functionality, performance, and security aspects of the RequirementSage chatbot.

- **1.3.1.4 Documentation Writers**

  Individuals tasked with creating user manuals, system documentation, and training materials.

1.3.2 Reading Suggestions

For an effective understanding of the SRS, readers are encouraged to follow the suggested sequence below:

- **1.3.2.1 Overview Sections**
  - Begin with the "1. Abstract" to grasp the high-level goals and vision of RequirementSage.
  - Proceed to the "1.1 Purpose" to understand the overarching objectives and aspirations for the chatbot.

- **1.3.2.2 Project Proposal Summary**
    - Review the "Summary of Project Proposal" (Section 2) to gain insights into the scope, goals, and key features of RequirementSage.
- **1.3.2.3 Functional Requirements**
    - **1.3.2.3.1 Developers:** Focus on "3. Functional Requirements" for a detailed breakdown of the chatbot's capabilities and expected behaviors.
    - **1.3.2.3.1 Project Managers:** Pay attention to the scope of the project, business goals, and objectives outlined in "3. Functional Requirements."
- **1.3.2.4 Non-Functional Requirements**
    - **1.3.2.4.1 QA/Testers:** Explore "4. Non-Functional Requirements" to understand the criteria for performance, availability, security, and scalability testing.
    - **1.3.2.4.1 Developers:** Consider technical aspects related to the chatbot's architecture and capabilities.
- **1.3.2.5 Business Requirements**
    - Business Stakeholders: Examine "5. Business Requirements" to align the project with business goals, scalability considerations, and SEO strategies.
- **1.3.2.6 Document Conventions**
    - **Documentation Writers:** Refer to "6. Document Conventions" to understand the formatting and conventions used throughout the SRS.

By following this suggested sequence, each reader type can focus on the sections most relevant to their role, ensuring a comprehensive understanding of RequirementSage and its requirements.

1.4 Scope

1.4.1 Business Goal and Objectives

- **1.4.1.1** The primary objective is to automate the process of gathering requirements for users in the context of small-scale projects.
- **1.4.1.2** Reduce the effort and time required for the requirements-gathering phase for small-scale projects.
- **1.4.1.3** Summarize the requirements to enhance user satisfaction for small-scale projects.
- **1.4.1.4** Enable real-time generation of documentation to minimize user effort specifically for small-scale projects.

1.4.2 Project’s Goal and Objectives

- **1.4.2.1** Ensure that the chatbot has a fair understanding of human language and phrases for small-scale projects.
- **1.4.2.2** Enable the chatbot to understand synonyms, misspelled words, and context related to requirements for small-scale projects.
- **1.4.2.3** Recognize user intent accurately to tailor requirements for the specific situation in small-scale projects.

- **1.4.2.4** Enable the chatbot to categorize and organize entities related to each other for small-scale projects.

1.4.3 Project Deliverables

- **1.4.3.1** Provide clear instructions to users on how to interact with the chatbot during their first interaction. These instructions should be tailored to small-scale projects.
- **1.4.3.2** Develop predefined scripts as functional requirements to confirm and validate discovered and generated requirements for small-scale projects.
- **1.4.3.3** Continuously validate the chatbot's functionality during its development, training, and ongoing operations, with a focus on small-scale project requirements.

1.5 References

IEEE template for SRS

Project Proposal-Deliverable 1

Requirements Elicitation-Deliverable 2

Requirements Elaboration-Deliverable 3

2.Product Description

2.1 Product Perspective

RequirementSage is a standalone, innovative product that represents a new, self-contained solution for revolutionizing software requirements gathering. Unlike traditional methods, RequirementSage utilizes advanced Natural Language Processing (NLP) and Machine Learning (ML) technologies to automate and streamline the process of collecting comprehensive software requirements. This product is not a follow-on member of an existing product family, nor is it intended as a direct replacement for any specific existing system.

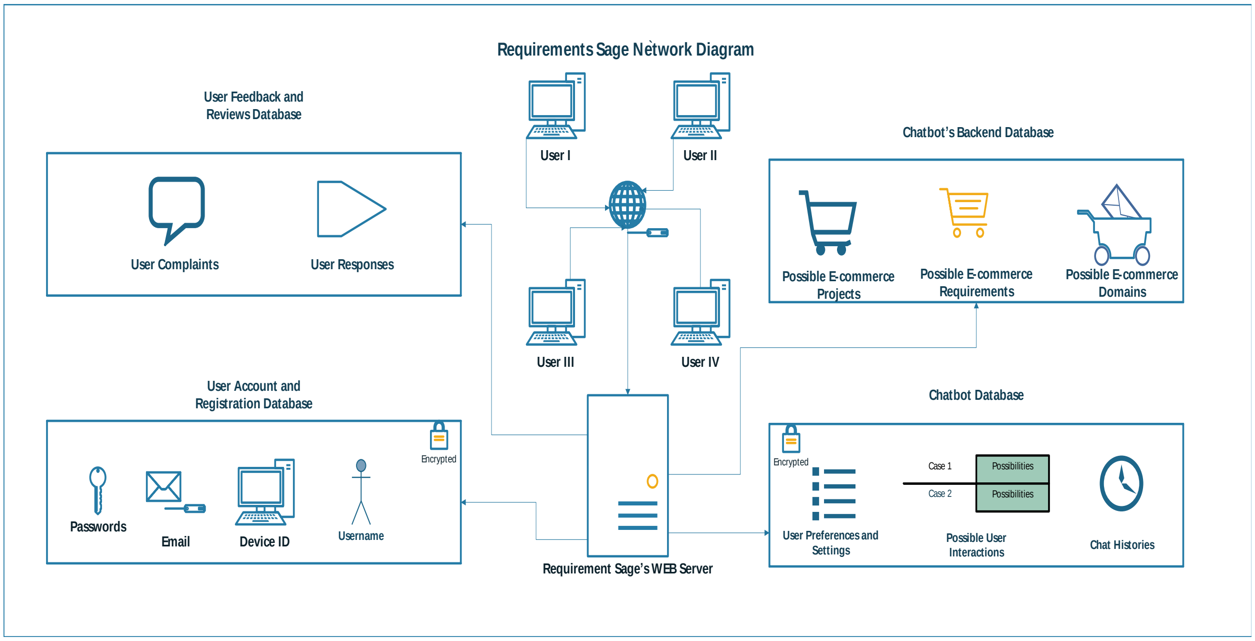
2.1.1 Origin and Context

The inception of RequirementSage is rooted in addressing the challenges associated with manual and time-consuming software requirements gathering processes, especially in the context of small-scale projects, with a primary focus on e-commerce websites. The need for a more efficient, user-friendly, and accurate solution led to the development of RequirementSage as an intelligent chatbot.

2.1.2 Relation to Larger Systems

RequirementSage is designed as a self-contained product dedicated to the specific task of software requirements gathering. While it may interact with external services, such as the OpenAI API for advanced NLP capabilities, it does not function as a component of a larger system. The boundaries of RequirementSage are well-defined, with a focus on providing a comprehensive yet specialized solution for stakeholders involved in small-scale software development projects.

2.1.3 System Architecture Overview


Requirements Sage Network Diagram
User Feedback and Reviews Database
User Complaints
User Responses
User I
User II
User III
User IV
Chatbot's Backend Database
Possible E-commerce Projects
Possible E-commerce Requirements
Possible E-commerce Domains
User Account and Registration Database
Encrypted
Passwords
Email
Device ID
Username
Requirement Sage's WEB Server
Chatbot Database
Encrypted
User Preferences and Settings
Case 1
Case 2
Possibilities
Possibilities
Possible User Interactions
Chat Histories

2.2 Product Functions

The key functions of RequirementSage are organized to provide a high-level summary of the major capabilities that the product must perform or enable users to perform. Detailed explanations and specifications are provided in Section 3. The functions can be summarized as follows:

2.2.1 User Account Management

This feature shall allow users to create and manage their accounts within the software. User account management shall be an essential part of gathering requirements because it can help identify and track user preferences and feedback.

2.2.2 Requirements Gathering via Chat

This shall be a core feature of the software. It shall enable users to interact with the software to discuss and specify their software requirements. This shall use Natural Language Processing

(NLP) for asking questions, providing information, and receiving clarifications from the user to ensure that the requirements are well-understood and documented.

2.2.3 Project Proposal Generation

This feature shall automatically generate a project proposal document. It shall include details like project scope, objectives, timelines, features, and other relevant information based on the input received during the requirements-gathering process.

2.2.4 Requirements Listing

This feature shall serve as the initial step in documenting software requirements. It shall involve compiling the list of all the identified software requirements during the requirements gathering phase and providing the user with that list.

2.2.5 Creation of Software Requirement Document (SRD)

This feature shall involve the comprehensive documentation of all the software requirements in a structured and detailed manner. It shall include a thorough description of each requirement and the relation between them and provide the user with that document.

2.2.6 User Feedback Collection

This feature shall focus on gathering feedback from the user and based on that feedback it shall give the updated version of the requirement listing and software requirement document.

2.3 User Classes and Characteristics for RequirementSage Chatbot

The anticipated user classes for RequirementSage Chatbot are derived from insights gathered through a survey targeting students involved in web development projects. These user classes are differentiated based on their roles, technical expertise, and specific needs in the software development process.

2.3.1 Web Developers

- **2.3.1.1 Frequency of Use**
  Regular engagement during project planning and software development stages.

- **2.3.1.2 Technical Expertise**
  High technical proficiency in web development.

- **2.3.1.3 Interactions**

Actively participate in gathering software requirements, utilizing the chatbot for efficient and accurate information extraction.

- **2.3.1.4 privilege Levels**
  Standard user privileges with a focus on requirements gathering and project utilization.

- **2.3.1.5 Importance**
  Primary user class, as web development is the most dominant domain among respondents.

2.3.2 Project Planners (Requirement Engineers)

- **2.3.2.1 Frequency of Use**
  Regular engagement during the initial stages of project planning.

- **2.3.2.2 Technical Expertise**
  Varied technical proficiency, focused on project management and planning.

- **2.3.2.3 Interactions**
  Utilize the chatbot for defining project scope, objectives, and initial requirements.

- **2.3.2.4 Privilege Levels**
  Standard user privileges with a focus on project planning functionalities.

- **2.3.2.5 Importance**
  Crucial for the effective utilization of RequirementSage during the planning phase.

2.3.3 Software Engineers (Students)

- **2.3.3.1 Frequency of Use**
  Periodic engagement for gathering and reviewing software requirements.

- **2.3.3.2 Technical Expertise**
  Moderate technical proficiency as students in software engineering.

- **2.3.3.3 Interactions**
  Actively participate in the requirements gathering process and utilize the chatbot for project insights.

- **2.3.3.4 Privilege Levels**
  Standard user privileges with a focus on utilizing gathered requirements.

- **2.3.3.5 Importance**
  Significant user class, representing the primary audience for the survey and aligned with the software engineering discipline.

2.4 Operating Environment

2.4.1 Hardware Platform

- **2.4.1.1** The chatbot is platform-agnostic, designed to run on standard hardware configurations commonly used for web and software applications.
- **2.4.1.2** It should be compatible with desktops, laptops, and servers with reasonable processing power and memory capacity to handle the chatbot's NLP and ML functionalities.

2.4.2 Operating System and Versions

- **2.4.2.1** RequirementSage is designed to be compatible with multiple operating systems, ensuring flexibility for users.

  - **2.4.2.1.1 Windows:** Windows 7 and above
  - **2.4.2.1.2 macOS:** macOS 10.14 (Mojave) and above
  - **2.4.2.1.3 Linux:** Ubuntu 18.04 LTS and above

2.4.3 Web Browsers

- **2.4.3.1** The chatbot is primarily accessed through web interfaces; therefore, compatibility with modern web browsers is essential.

  - **2.4.3.1.1** Google Chrome
  - **2.4.3.1.2** Mozilla Firefox
  - **2.4.3.1.3** Microsoft Edge
  - **2.4.3.1.4** Safari

2.4.4 Other Software Components

- **2.4.4.1 Database for Storing User Details**

  The chatbot may utilize a database system for storing user accounts, gathered requirements, and other relevant data. The compatibility with commonly used database systems, such as MySQL or MongoDB, is essential.

- **2.4.4.2 Database for Training Chatbot and Displaying existing requirements to the user**

  The chatbot utilizes a database on the backend to display already existing requirements to the user to select from them. This database also acts as the training database for the chatbot and is constantly updated with the new requirements.

2.4.5 Integration Points

- **2.4.5.1 Export Formats**

  The chatbot should support export functionality in standard formats, such as PDF, to ensure compatibility with common document viewing applications.

2.4.6 Network Requirements

- **2.4.6.1** RequirementSage operates in an environment with reliable internet connectivity. Continuous connectivity is crucial for real-time interactions, data storage, and potential integration with external services.

2.4.7 Security Considerations

- **2.4.7.1** The chatbot should adhere to security protocols for data transmission and storage, especially when interacting with sensitive project information.
- **2.4.7.2** Compliance with industry-standard encryption protocols for data security during transmission.

2.5 Limitations and Constraints

- ### 2.5.1 AI and NLP Focus on Small-Scale Projects

  The project deliberately concentrates on integrating Artificial Intelligence (AI) and Natural Language Processing (NLP) approaches tailored for small-scale projects, with a specific emphasis on the unique requirements of e-commerce scenarios.

- ### 2.5.2 Time Constraints and Emphasis on Efficiency

  Acknowledging the limited time available for development, training, and operations, the project places a strong emphasis on efficiency, particularly for small-scale projects. The condensed timeline may impact certain aspects of the project, making efficient processes crucial for successful implementation.

- ### 2.5.3 Restriction to E-commerce Websites

  The chatbot is designed exclusively for requirements gathering in the context of e-commerce websites. Attempts to use the chatbot for requirements unrelated to e-commerce may result in errors or unintended behavior. The system is specialized to cater specifically to the needs and language patterns associated with e-commerce projects.

- **2.5.4 Language Requirements**

  Development and user interaction language will be primarily in English. Consideration for multilingual support may be a future enhancement but is not part of the initial scope.

2.6 User Documentation

2.6.1 User Guideline Page

- **2.6.1.1** A dedicated user guideline page will be included in the documentation, providing users with a quick reference for common tasks and best practices.
- **2.6.1.2** The user guideline page will offer succinct instructions and tips for efficient use of RequirementSage.

2.7 Assumptions and Dependencies

2.7.1 Assumptions

- **2.7.1.1 Internet Connectivity**

  Assumed reliable and continuous internet connectivity for users interacting with RequirementSage, as the chatbot operates in real-time and relies on external services and systems.

- **2.7.1.3 E-commerce Focus**

  The chatbot's specialization for small-scale projects is assumed to align with the unique needs of e-commerce websites. Any deviation in project scope may impact the effectiveness of the chatbot.

2.7.2 Dependencies

2.7.2.1 External Libraries

RequirementSage may utilize external libraries for specific functionalities. The project is dependent on the availability and compatibility of these libraries i.e.

- Django
- NLP toolkit
- Tensorflow
- sklearn
- aiohttp==3.8.4
- aiosignal==1.3.1
- argon2-cffi==21.3.0
- argon2-cffi-bindings==21.2.0

- asgiref
- async-timeout==4.0.2
- attrs==23.1.0
- bcrypt==4.0.1
- certifi
- cffi==1.15.1
- charset-normalizer==3.1.0
- colorama==0.4.6
- DateTime==5.1
- Faker==18.4.0
- frozenlist==1.3.3
- idna==3.4
- multidict==6.0.4
- openai==0.27.6
- Pillow==9.5.0
- pyaes==1.6.1
- pycparser==2.2
- PySocks==1.7.1
- python-dateutil==2.8.2
- pytz==2023.3
- requests==2.29.0
- six==1.16.0
- sqlparse
- TgCrypto==1.2.5
- tqdm==4.65.0
- tzdata
- urllib3==1.26.15
- wincertstore==0.2
- yarl==1.9.2 ,
- zope.interface==6.0

2.7.2.2 Database System

Dependency on a selected database system (e.g., MySQL) for storing user accounts, gathered requirements, and other relevant data.

2.7.2.3 Backend Database and NLP Model

Dependency on the data of the backend database which stores the existing requirements and trains the NLP model

2.7.3 Development Environment

2.7.3.1 Development Tools

Assumed usage of standard development tools such as integrated development environments (IDEs) and version control systems.

2.7.3.2 Training Data Availability

The effectiveness of the chatbot's machine learning models assumes the availability of relevant and sufficient training data.

2.7.4 External Factors

2.7.4.1 Legal and Regulatory Compliance

Assumed adherence to legal and regulatory requirements related to data privacy, AI usage, and software development practices.

2.7.4.2 Customer Collaboration

Collaboration with the customer's organization is assumed for ongoing maintenance and adherence to their standards.

3. External Interface Requirements

3.1 User Interfaces

- **3.1.1 Chatbot Interface**
  - **3.1.1.1** The chatbot interface will feature a user-friendly design with a clean layout.
  - **3.1.1.2** Standard chat elements, including text input, send button, and chat history, will be present.
  - **3.1.1.3** GUI standards will align with modern conversational UI principles for intuitive interactions.
- **3.1.2 Web Interface**
  - **3.1.2.1** Users can access the chatbot via a web interface for requirements gathering.
  - **3.1.2.2** GUI will follow responsive design principles for a seamless experience across devices.
  - **3.1.2.3** Standard web navigation elements (e.g., menu, buttons) will be implemented.

3.2 Hardware Interfaces

- **3.2.1 Device Types**

- **3.2.1.1** RequirementSage will support standard computing devices (PCs, laptops).
- **3.2.1.2** Hardware interactions will be limited to typical input methods (keyboard, touch, etc.).

3.3 Software Interfaces

- **3.3.1 Database System**
  - **3.3.1.1** Interaction with the chosen database system (e.g., MySQL) for data storage.
  - **3.3.1.2** Definition of data items and messages exchanged for user accounts and gathered requirements.
- **3.3.2 NLP Model and Training Database**
  - **3.3.2.1** Integration with Backend Database and the NLP Model for natural language processing capabilities and storing/displaying requirements.
  - **3.3.2.2** Specification of data items and messages exchanged for language understanding.

3.4 Communications Interfaces

- **3.4.1 Chatbot Communication**
  - **3.4.1.1** Real-time communication between the chatbot and users.
  - **3.4.1.2** Use of standard messaging protocols for efficient data transfer.
  - **3.4.1.3** Security measures to protect user data during communication.
- **3.4.2 Web Interface Communication**
  - **3.4.2.1** Web interface communication using HTTP/HTTPS protocols.
  - **3.4.2.2** Secure data transfer and encryption for user privacy.
  - **3.4.2.3** Synchronization mechanisms for consistent interactions.

4. System Features

4.1 User Account Management

4.1.1 Description and Priority

- **Feature Name**

  User Account Management
- **Description**

  Allows users to create and manage accounts for tracking preferences and feedback.
- **Priority**

  Medium

4.1.2 Stimulus/Response Sequences

- **Stimulus**

User selects the option to register or manage their account.

- **Response**

  The software facilitates user account creation, login, profile modification and forgot password.

4.1.3 Functional Requirements

- **F-4.1.3.1: Registration**

Users shall be able to register with a unique username, email address, and password.

- **F-4.1.3.2: Login**

Registered users shall be able to log in using their username and password.

- **F-4.1.3.3: Profile Modification**

Users shall be able to change their email address and password.

- **F-4.1.3.4: Account logout**

The software shall provide an option for users to logout of their accounts.

4.2 Requirements Gathering via Chat

4.2.1 Description and Priority

- **Feature Name**
  Requirements Gathering via Chatbot
- **Description**
  Enables users to discuss and specify software requirements through natural language interaction.
- **Priority**
  High

4.2.2 Stimulus/Response Sequences

- **Stimulus**

User initiates a chat session for gathering software requirements by typing “Hello” or similar words.

- **Response**

The software processes user-provided input in English, identifies requirements, generates elicitation questions, captures and stores user responses, and provides assistance and suggestions during discussions.

4.2.3 Functional Requirements

- **F-4.2.3.1: User Initialization**

The chatbot shall respond appropriately when the user initiates a chat session.

- **F-4.2.3.2: Language Processing**

The chatbot shall process user-provided input in the English language.

- **F-4.2.3.3: Requirement Identification**

The software shall parse user input to identify requirements, questions, and clarifications.

- **F-4.2.3.4: Display Existing Requirements**

The chatbot shall display existing requirements to the user that are already in the database and matches with the user project’s requirements. Moreover, chatbot shall display them with the checkboxes so that user shall select the desired requirements.

- **F-4.2.3.5: Question Generation**

The chatbot shall generate relevant requirement elicitation questions based on user-provided information.

- **F-4.2.3.6: Response Capture**

The software shall capture and store user responses to elicitation questions, associating them with specific requirements.

- **F-4.2.3.7: Record Maintenance**

The software shall maintain a record of gathered requirements.

- **F-4.2.3.8: Assistance and Suggestions**

The software shall provide assistance and suggestions to users during requirement discussions.

- **F-4.2.3.9: Confirmation and Finalization**

Users shall have the ability to confirm and finalize requirements in the chat.

- **F-4.2.3.10: Scope Handling**

Chatbot shall recognize if the project scope is supported by it or not. If not it shall aware the user with the scope of the project for which the chatbot can elicit requirements.

4.3 Project Proposal Generation

4.3.1 Description and Priority

- **Feature Name**

Project Proposal Generation

- **Description**

Automatically generates project proposals based on the gathered requirements.

- **Priority**

Medium

4.3.2 Stimulus/Response Sequences

- **Stimulus**

User requests the generation of a project proposal.

- **Response**

The software defines project scope, identifies objectives, lists features, and generates a project proposal document.

4.3.3 Functional Requirements

- **F-4.3.3.1: Scope Definition**

The software shall define the project scope based on the gathered requirements.

- **F-4.3.3.2: Objective Identification**

Project objectives shall be identified and documented based on the gathered information.

- **F-4.3.3.3: Feature Listing**

The software shall list project features and functionalities derived from the gathered requirements.

- **F-4.3.3.4: Document Generation**

The software shall generate a project proposal document, including scope, features, and requirements.

- **F-4.3.3.5: Export Functionality**

Users shall be able to export the project proposal document in PDF format.

4.4 Requirements Listing

4.4.1 Description and Priority

- **Feature Name**

Requirements Listing

- **Description**

Compiles a comprehensive list of identified software requirements.

- **Priority**

Medium

4.4.2 Stimulus/Response Sequences

- **Stimulus**

User confirms and finalizes requirements in the chat.

- **Response**

The software compiles a list of identified software requirements.

4.4.3 Functional Requirements

- **F-4.4.3.1: Comprehensive Compilation**

The software shall compile a comprehensive list of identified software requirements.

- **F-4.4.3.2: Export Functionality**

  Users shall be able to export the compiled list of requirements in PDF format.

- **F-4.4.3.3: Dynamic Updates**

  The software shall accommodate updates and changes to individual requirements, allowing users to add, modify, or remove requirements.

4.5 Creation of Software Requirement Document (SRD)

4.5.1 Description and Priority

- **Feature Name**

Creation of Software Requirement Document (SRD)

- **Description**

Creates a structured Software Requirement Document (SRD) based on gathered information.

- **Priority**

High

4.5.2 Stimulus/Response Sequences

- **Stimulus**

  User requests the creation of an SRD after finalizing requirements.

- **Response**

  The software generates a structured SRD with introduction, purpose, scope, requirements, dependencies, and other relevant sections.

4.5.3 Functional Requirements

- **F-4.5.3.1: Structured Document**

  The software shall create an SRD with structured sections.

- **F-4.5.3.2: Export Functionality**

  Users shall be able to export the SRD document in PDF format.

- **F-4.5.3.3: Dynamic Editing**

  Users shall be able to add, modify, or remove requirements and associated content in the SRD.

4.6 User Feedback Collection

4.6.1 Description and Priority

- **Feature Name**

User Feedback Collection

- **Description**

Gathers user feedback for continuous improvement.

- **Priority**

Medium

4.6.2 Stimulus/Response Sequences

- **Stimulus**

User provides feedback in the form of replies to the chat.

- **Response**

The software acknowledges and integrates user feedback into improving the responses using Reinforcement Learning.

4.6.3 Functional Requirements

- **F-4.6.3.1: Feedback Collection**

The chatbot shall record the user replies and search for key words to improve the responses.

- **F-4.6.3.2: Utilizing Feedback in Responses**

The chatbot shall use these feedbacks to check the scope of the project and possible.

5. Other Nonfunctional Requirements

5.1 Performance Requirements

5.1.1 Response Time

NF-5.1.1 The chatbot app shall respond to user inputs within 3 seconds to ensure a seamless and efficient user experience.

5.1.2 Concurrent Handling

NF-5.1.2 It should be able to handle concurrent interactions with multiple users without significant performance degradation.

5.2 Availability Requirements

5.2.1 Continuous Availability

NF-5.2.1 The chatbot app shall be available 24/7 to cater to user requests and provide assistance at any time.

5.3 Security Requirements

5.3.1 Data Security

NF-5.3.1 User data and project requirements gathered by the chatbot app shall be stored securely and protected from unauthorized access.

5.3.2 Encryption

NF-5.3.2 The chatbot app shall use encryption protocols to ensure the confidentiality of user data during transmission.

5.4 Scalability Requirements

5.4.1 Horizontal Scaling

NF-5.4.1 The chatbot app should be designed to scale horizontally to accommodate increased user demand during peak usage periods.

5.5 Accuracy Requirements

5.5.1 NLP Accuracy

NF-5.5.1 The chatbot app's natural language processing (NLP) capabilities shall have a minimum accuracy rate of 85% in understanding and interpreting user responses accurately.

5.5.2 Recommendation Accuracy

NF-5.5.2 It should consistently provide accurate recommendations and responses based on the gathered requirements.

5.6 User Interface Requirements

5.6.1 Intuitive Interface

NF-5.6.1 The chatbot app shall have an intuitive and user-friendly interface to facilitate easy and efficient interaction with users.

5.7 Data Retention and Privacy Requirements

5.7.1 Data Retention Policy

NF-5.7.1 User data collected by the chatbot app for requirement gathering shall be retained for a specified period and then automatically deleted to ensure compliance with data privacy regulations.

6. Business Requirements

6.1 Purpose

6.1.1 Automation of Requirement Discovery

B-6.1.1 The chatbot shall clearly automate the process of discovering requirements.

6.1.2 Streamlining Elicitation Process

B-6.1.2 The chat sets to streamline the process for elicitation of requirements.

6.2 User Satisfaction

6.2.1 Enhanced User Feedback

B-6.2.1 Provide an enhanced User Feedback system to better understand user needs.

6.2.2 Optimized Performance

B-6.2.2 The performance of the product should be properly optimized, so users will not get upset.

6.3 Time Considerations

6.3.1 Efficient Project Timeline

B-6.3.1 The project timeline should be as efficient as possible to counter our time constraints.

6.4 Security

6.4.1 User Management Security

B-6.4.1 The product's user management should have no loopholes to guarantee user safety.

6.4.2 User Awareness

B-6.4.2 The user must be aware of our security policies to avoid future problems.

6.5 Data Privacy

6.5.1 Agreement with Privacy Policy

B-6.5.1 Agreement shall be made with the user in accordance with our privacy policy.

6.5.2 Terms and Conditions Agreement

B-6.5.2 The user shall only use our product if they agree to our Terms and Conditions.

6.6 Business Scalability

6.6.1 Handling User Load

B-6.6.1 The system should handle a good sum of users at live time.

6.6.2 Performance During Peak Usage

B-6.6.2 Performance must not be degraded at times of peak usage of our hosting resources.

6.7 SEO (Search Engine Optimization)

6.7.1 Search Result Alignment

B-6.7.1 Search results must be according to popular keywords so the user can receive documentation according to the trend.

6.7.2 Free Source of Information

B-6.7.2 Source of Information must be free to use due to our budget constraints.

7. Other Requirements

7.1 Database Requirements

7.1.1 Data Storage

The chatbot app shall have a backend database to store user preferences, historical interactions, and relevant project data.

7.1.2 Data Updating

The backend database shall be update accordingly so that the chatbot could generate more accurate responses. This database shall be used to train the NLP model.

7.1.2 Database Security

Access to the database shall be restricted to authorized personnel only, with encryption protocols implemented for sensitive data.

7.2 Internationalization Requirements

7.2.1 Cultural Sensitivity

Language and responses generated by the chatbot should be culturally sensitive to ensure a positive user experience globally.

7.3 Legal Requirements

7.3.1 Data Privacy Compliance

The chatbot app must comply with international data privacy regulations, ensuring user data is handled in accordance with legal standards.

7.3.2 Terms of Service

Clear and comprehensive terms of service must be provided to users, outlining the obligations and rights of both the users and the product.

7.4 Reuse Objectives

7.4.1 Code Reusability

The development team should aim for modular and reusable code components to facilitate future updates and extensions.

7.4.2 Knowledge Base Reusability

The chatbot's knowledge base should be designed for easy updates and expansions, allowing the integration of new information without significant rework.

7.5 Accessibility Requirements

7.5.1 Accessibility Standards

The chatbot app should adhere to accessibility standards to ensure users with disabilities can interact seamlessly.

7.5.2 Assistive Technologies Compatibility

Compatibility with common assistive technologies, such as screen readers, should be considered for an inclusive user experience.

7.6 Documentation Requirements

7.6.1 User Guidelines

A comprehensive user guideline page should be included in the user documentation to assist users in effectively utilizing the chatbot app.

Appendix A: Glossary

- **Chatbot:** A computer program designed to simulate conversation with human users, especially over the Internet.
- **NLP (Natural Language Processing):** A field of artificial intelligence that focuses on the interaction between computers and humans using natural language.
- **ML (Machine Learning):** A subset of artificial intelligence that enables systems to automatically learn and improve from experience without being explicitly programmed.
- **API (Application Programming Interface):** A set of rules that allows one software application to interact with another.
- **UI (User Interface):** The point of interaction between the user and the chatbot, including screens, pages, and visual elements.
- **NLU (Natural Language Understanding):** A component of NLP that focuses on extracting meaning from human language.

Appendix B: Analysis Models

For class diagram, DFDs and ERD find the link attached with the name of **Requirements Elaboration-Deliverable 3** in the references section.

1. Use Cases

UC-1.1

Use Case ID:	UC-1		
Use Case Name:	Sign up	**Version No:**	1.0
End Objective:	Managing user accounts within the system		
Created by:	Anas Khan	**On (Date):**	March 18, 2023
Approved by:	Syed Huzaifa, Faizan Saleh	**On (Date):**	March 19, 2023
Actor:	User		
Pre-conditions			
User must open the system and be on the sign up screen			
Post-conditions			
System must appropriately manage the user account based on the performed actions			
Basic Flow			
Steps	**User Actions**	**System Actions**	

1	User entered name, email and password	
2		System shows check mark if name, email and password are valid
3	User clicked on sign up button	
4		System prompts “Account successfully Created” on screen and take the user to main interface of the application
Alternate Flow		
Steps	**User Actions**	**System Actions**
1.a	User entered name/email/password in wrong format	Prompt user to enter valid data
Go to Step 1		
1.b	User entered an easy password	Prompt user to enter a stronger password of at least 8 characters/digits
Go to Step 1		
3.a	User entered nothing in the input fields but clicked on Sign up button	System will mark the input fields red and prompt to Enter the name, email, and password
Go to Step 1		

UC-1.2

Use Case ID:	UC-2		
Use Case Name:	Log in	**Version No:**	1.0
End Objective:	System gets user logged in the application		
Created by:	Anas Khan	**On (Date):**	March 18, 2023
Approved by:	Syed Huzaifa, Faizan Saleh	**On (Date):**	March 19, 2023
Actor:	User		
Pre-conditions			
User must open the system and must be on the Log in page			
Post-conditions			
System must take the user to main interface if Log in is succeeded			
Basic Flow			
Steps	**User Actions**	**System Actions**	
1	User entered email and password		
2	User clicked on Log in button		
3		System takes the user to main interface of the application	

4	User clicked on Forgot Password link	
5		System takes user to account verification page
6	User clicked on Sign Up link	
7		System takes user to sign up page
Alternate Flow		
Steps	**User Actions**	**System Actions**
1.a	User entered wrong email address or password	Prompt user to enter correct password or click on forgot password in case password not remember
Go to Step 1 or Step 4		
1.b	User entered a new email which isn't in the database	Prompt user "Account not found, please enter correct email and password or sign up" message
Go to Step 1 or Step 6		
2.a	User entered nothing in the input fields but clicked on Sign up button	System will mark the input fields red and prompt to Enter the name, email, and password
Go to Step 1		

UC-1.3

Use Case ID:	UC-3		
Use Case Name:	Requirements Gathering via Chat	**Version No:**	1.0
End Objective:	Gathering software requirements through a chat session		
Created by:	Anas Khan	**On (Date):**	March 18, 2023
Approved by:	Syed Huzaifa, Faizan Saleh	**On (Date):**	March 19, 2023
Actor:	User		
Pre-conditions			
User must be logged into the system			
Post-conditions			
System must maintain a record of gathered requirements			
Basic Flow			
Steps	**User Actions**	**System Actions**	
1	User initiates a chat session by typing "Hello"	System acknowledges the user's initiation and prepares for input processing.	
2	User asks relevant questions	The software generates relevant requirement elicitation questions based on user-provided information	

3		System shows some existing requirements and features list related to the user's project scope in the form of checkboxes
4	User check the desired boxes	System narrow downs the checked requirements to gather more detail about them
3	User continues the discussion, and the system records details of the gathered requirements	The system maintains a record of gathered requirements in a structured format for future reference.
4	User interacts with the system based on provided assistance and suggestions.	The system offers contextual assistance and suggestions based on the ongoing discussion.
Alternate Flow		
Steps	**User Actions**	**System Actions**
1.	User provides input in a language other than English.	System prompts the user to use English for requirement gathering.
Go to Step 1		
2.	User asks something that is outside of the chatbot's scope	System apologize the user and aware them with the scope of the projects for which the user can generate requirements
Go to Step 2		
4.	User didn't check mark any box	System asks for more detail about the project idea from the user and generates a new list of possible requirements
Go to Step 4		

UC-1.4

Use Case ID:	UC-4		
Use Case Name:	Forgot Password	**Version No:**	1.0
End Objective:	System recovers the account password for user		
Created by:	Anas Khan	**On (Date):**	March 18, 2023
Approved by:	Syed Huzaifa, Faizan Saleh	**On (Date):**	March 19, 2023
Actor:	User		
Pre-conditions			
User must open the system, on login screen and must have application's user account			

Post-conditions		
System must recover the user's account and assign a new password to it		
Basic Flow		
Steps	**User Actions**	**System Actions**
1	User clicked on Forgot Password Link	
2		System asks for user's email associated with the account
3		System matches that email in the database, then send a recovery code to mail, takes the user to Forgot Password Window and ask for the recovery code
4	User entered the recovery code	
5		System checks the validity of code and takes to the new password window
6	User enter new password and confirm that password by entering again, then click on save password button	
7		System saves the password and takes user to the main interface
Alternate Flow		
Steps	**User Actions**	**System Actions**
2.a	User's email didn't match in the database	System prompt that "Account not found, please enter correct email or Sign up"
Go to Step 2		
4.a	User enter wrong recovery code	System Prompts "Check the recovery code and enter again"
Go to Step 4		
6.a	User entered a weak password	System Prompts to enter a strong password of at least 8 characters
6.b	New password didn't match with the confirm password	System prompts password not matched, please recheck
Go to Step 6		

UC-1.5

Use Case ID:	UC-5		
Use Case Name:	Project Proposal Generation	**Version No:**	1.0
End Objective:	Generating a project proposal based on gathered requirements		

Created by:	Anas Khan	**On (Date):**	March 18, 2023
Approved by:	Syed Huzaifa, Faizan Saleh	**On (Date):**	March 19, 2023
Actor:	User		
Pre-conditions			
Requirements gathering must be completed, and the system must have a record of gathered requirements			
Post-conditions			
System generates and provides the user with a project proposal document in PDF format			
Basic Flow			
Steps	**User Actions**	**System Actions**	
1	User expresses the intention to generate a project proposal.	System verifies that the requirements gathering process is completed and proceeds to the next step.	
2	User reviews the defined project scope, objectives, features, and functionalities.	System presents the compiled information for user review.	
3	User triggers the generation of a project proposal document.	System compiles all relevant information into a project proposal document.	
4	User exports the project proposal document saved in PDF format.	System provides an option for the user to export the project proposal document in PDF format.	
Alternate Flow			
Steps	**User Actions**	**System Actions**	
1.	User attempts to generate a project proposal without completing requirements gathering.	System prompts the user to complete the requirements gathering process before proceeding. The system guides the user back to the necessary steps.	

UC-1.6

Use Case ID:	UC-6		
Use Case Name:	Requirements Listing	**Version No:**	1.0
End Objective:	Compiling and managing a comprehensive list of identified software requirements		
Created by:	Anas Khan	**On (Date):**	March 18, 2023
Approved by:	Syed Huzaifa, Faizan Saleh	**On (Date):**	March 19, 2023
Actor:	User		
Pre-conditions			
The requirements gathering process is completed.			
Post-conditions			
A comprehensive list of requirements is compiled and can be managed by the user.			
Basic Flow			
Steps	**User Actions**	**System Actions**	

1	Indicates the desire to compile a list of identified software requirements.	Verifies that the requirements gathering process is completed and proceeds to the next step.
2	User Reviews and confirms the comprehensive list of identified software requirements compiled from various sources.	Presents the compiled list of identified software requirements for user review.
3	Classifies requirements into different categories or types (e.g., functional requirements, non-functional requirements, and user-specific requirements).	Provides options for the user to classify requirements into predefined categories.
4	Updates and manages individual requirements, adding, modifying, or removing them from the compiled list.	Accommodates user updates and changes, allowing the addition, modification, or removal of individual requirements.
Alternate Flow		
Steps	**User Actions**	**System Actions**
1.	Chooses to cancel the update and return to the previous step.	Discards the changes made by the user and returns to the previous step.
2.	attempts to add a duplicate requirement.	Notifies the user that the requirement already exists and prompts for a unique identifier.
3.	Tries to delete a requirement that is linked to other parts of the system.	Informs the user that the requirement cannot be deleted due to existing dependencies.

UC-1.7

Use Case ID:	UC-7		
Use Case Name:	Creation of Software Requirement Document (SRD)	**Version No:**	1.0
End Objective:	Creating a structured Software Requirement Document (SRD) with relevant sections		
Created by:	Anas Khan	**On (Date):**	March 18, 2023
Approved by:	Syed Huzaifa, Faizan Saleh	**On (Date):**	March 19, 2023
Actor:	User		
Pre-conditions			
The requirements gathering process is completed.			
Post-conditions			
A structured SRD is created, and the user can export it.			
Basic Flow			
Steps	**User Actions**	**System Actions**	
1	Indicates the desire to create a Software Requirement Document (SRD).	Verifies that the requirements gathering process is completed and proceeds to the next step.	
2	Specifies the structure and content of the SRD, including sections like introduction, purpose, scope, requirements, dependencies, and any other relevant sections.	Guides the user through the process of defining the SRD structure, providing options for each section.	
3	Enters information into each section of the SRD based on their project requirements.	Captures and organizes the information provided by the user, populating the respective sections of the SRD.	
4	Reviews the content of the SRD and makes any necessary adjustments.	Allows the user to review and edit the content of each section before finalizing the SRD.	
Alternate Flow			
Steps	**User Actions**	**System Actions**	
1.	Chooses to modify an existing requirement.	Prompts the user to specify the modifications to the requirement.	

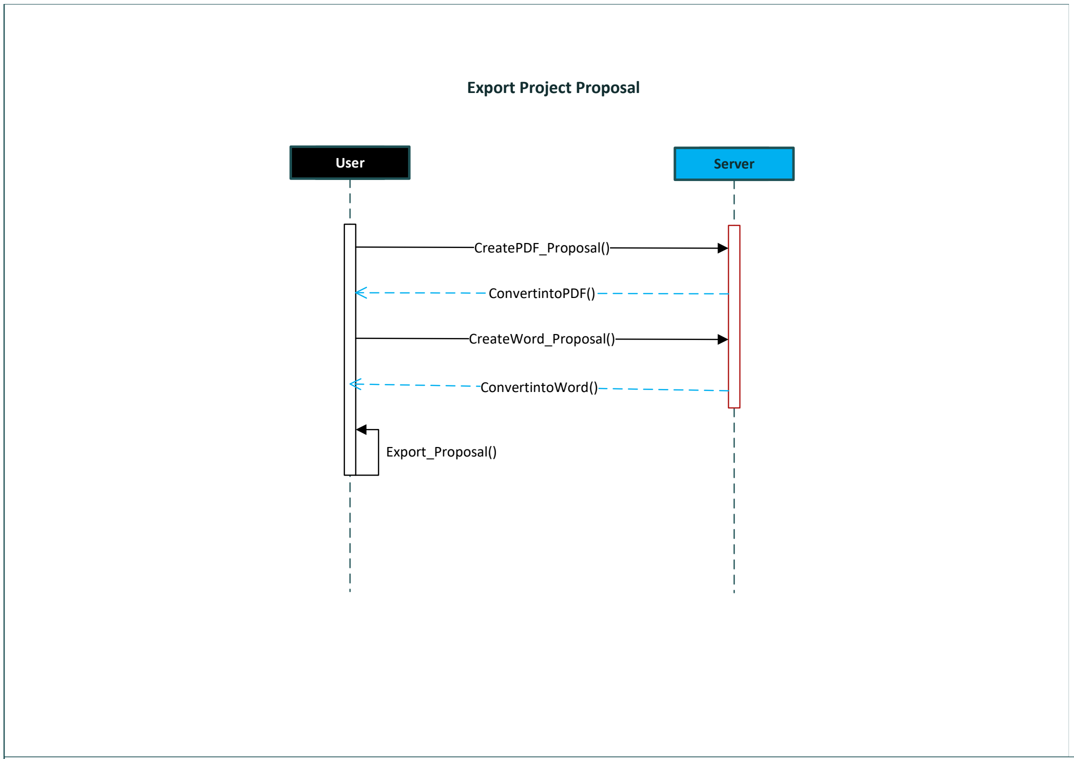
2.	Attempts to remove a requirement with dependencies.	Warns the user about existing dependencies and potential impacts.
3.	Users can provide feedback on the SRD content.	Implements a mechanism for users to submit feedback on the SRD content. If user feedback leads to SRD content changes, the software updates the SRD accordingly

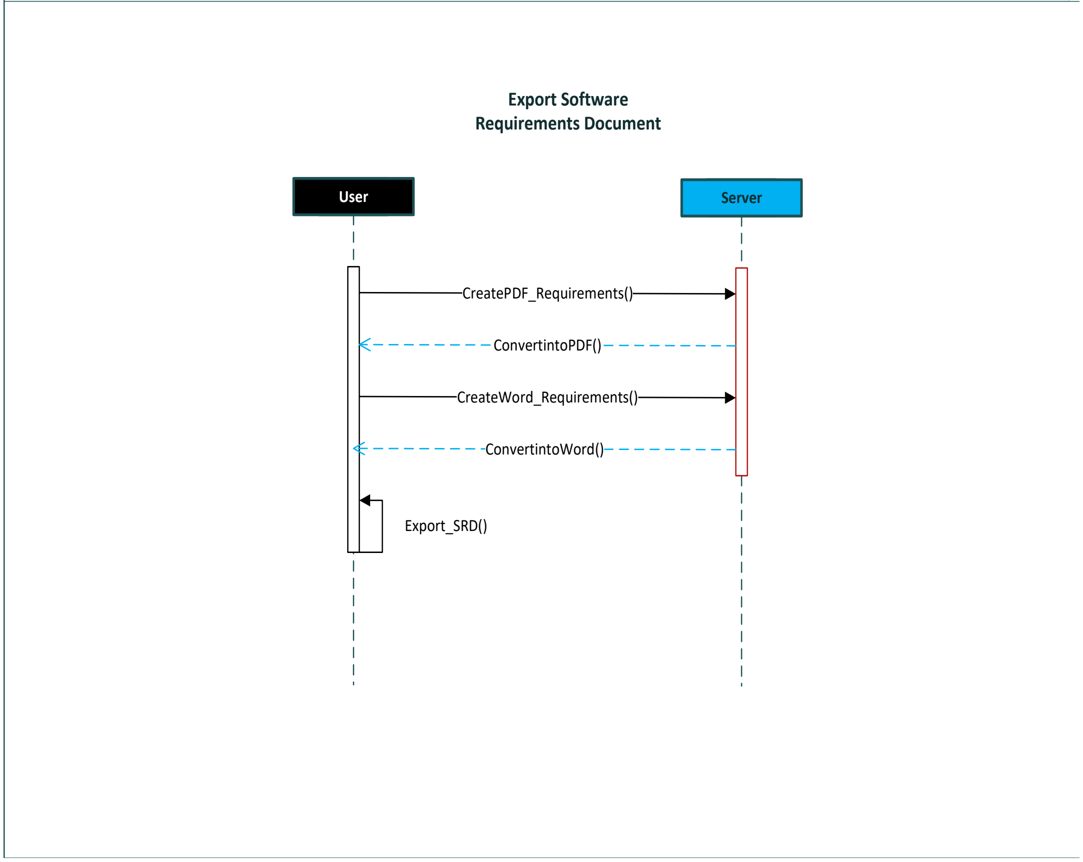
UC-1.8

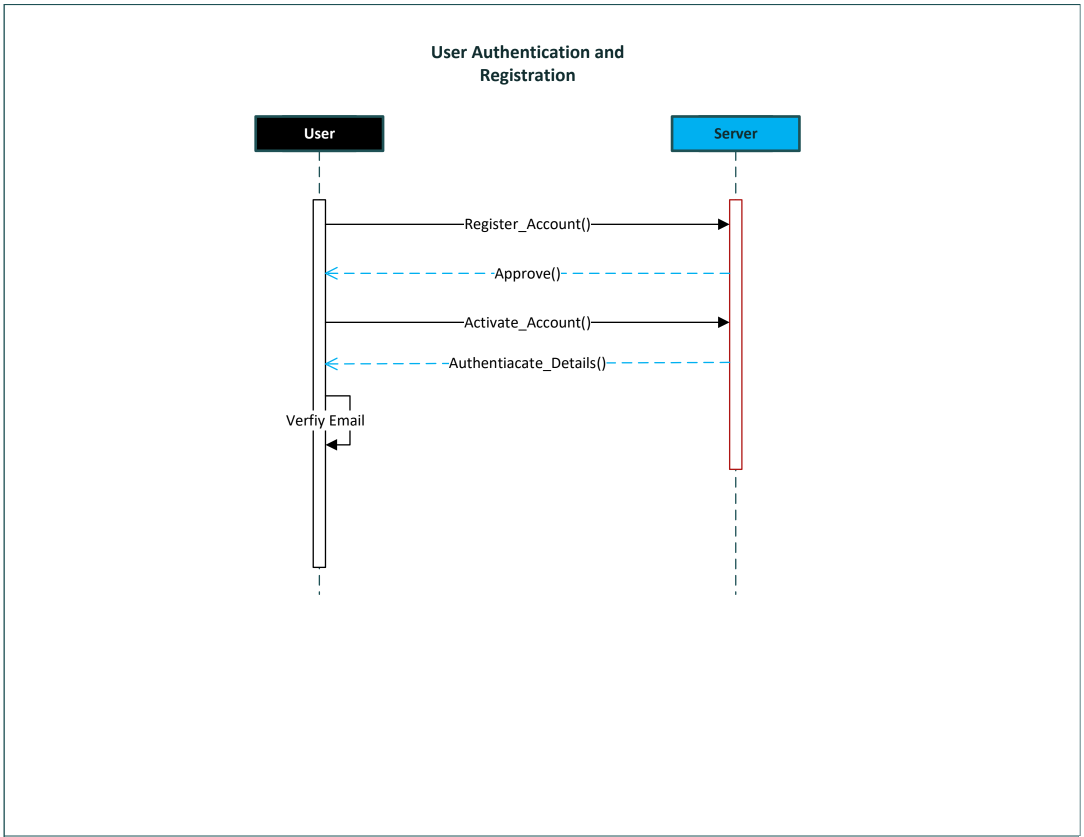
Use Case ID:	UC-8		
Use Case Name:	User Feedback Collection	**Version No:**	1.0
End Objective:	Collecting and incorporating user feedback on software requirements and the Software Requirement Document (SRD)		
Created by:	Anas Khan	**On (Date):**	March 18, 2023
Approved by:	Syed Huzaifa, Faizan Saleh	**On (Date):**	March 19, 2023
Actor:	User		
Pre-conditions			
The software requirements are defined. The SRD is created.			
Post-conditions			
User feedback is collected and integrated into the requirements listing and SRD. The user has access to the latest version of the SRD.			
Basic Flow			
Steps	**User Actions**	**System Actions**	
1	Indicates the desire to provide feedback on software requirements and the SRD.	Verifies that the software requirements are defined and an SRD is created.	
2	Provides feedback on specific software requirements or the overall SRD content.	Captures and records user feedback.	
3	Reviews the integrated user feedback within the requirements listing.	Presents the integrated user feedback within the requirements listing for the user to review.	
4	If user feedback leads to changes in requirements, the	Modifies the requirements listing based on user feedback.	

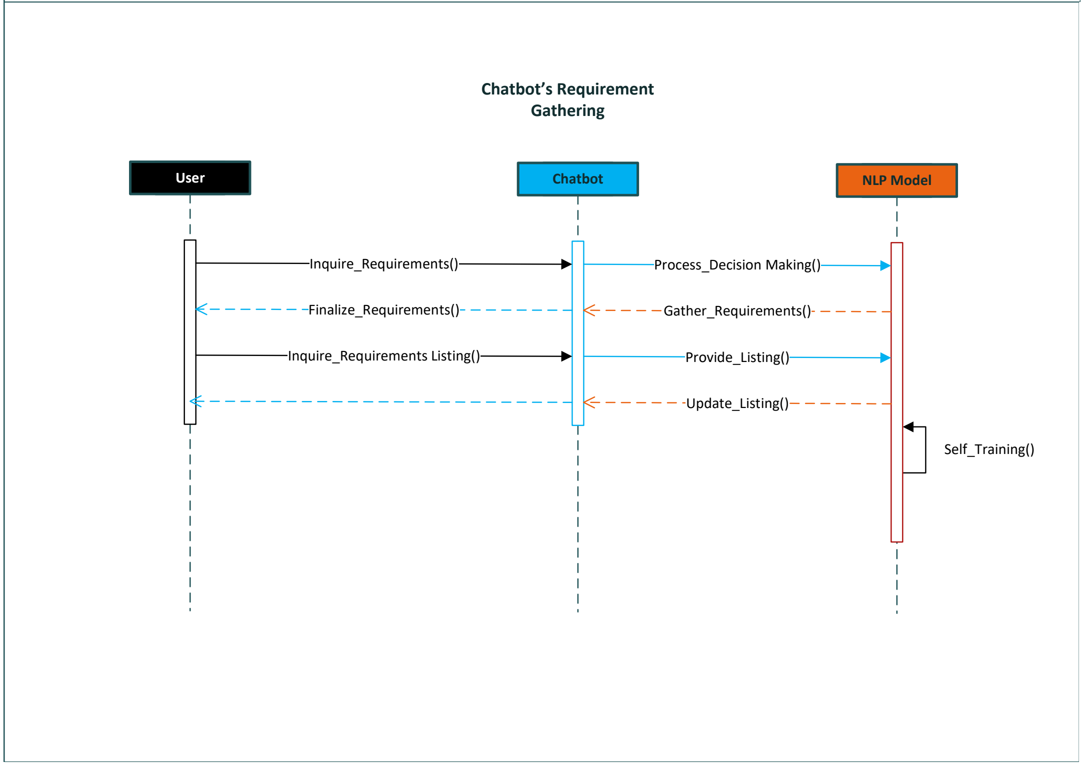
	software updates the requirements listing.	
5	If user feedback leads to revisions in the SRD, the software updates the SRD content.	Incorporates user feedback into the SRD, updating the content as necessary.
6	Confirms the completion of the feedback process and requests access to the latest version of the SRD.	Publishes and makes the latest version of the SRD accessible to the user.
Alternate Flow		
Steps	**User Actions**	**System Actions**
1.	Encounters difficulties in providing feedback and requests system assistance.	Provides guidance and assistance to help the user effectively communicate feedback.

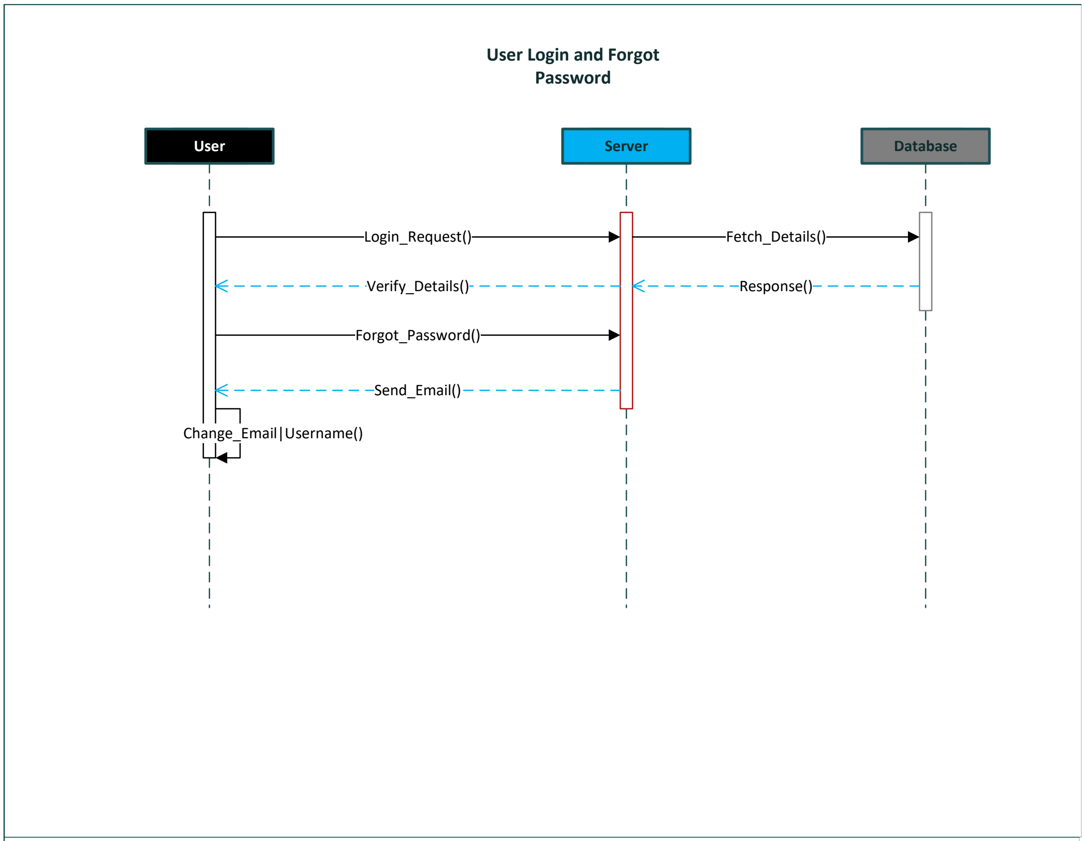
2. Sequence Diagrams

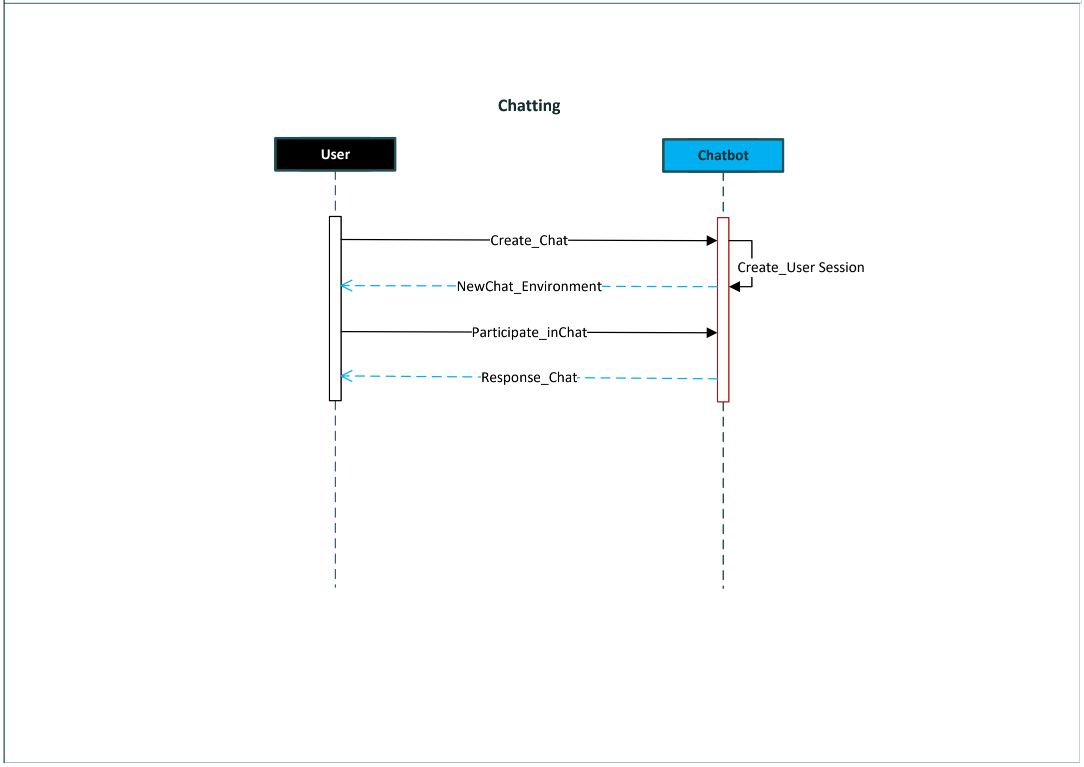

Export Project Proposal
User
Server
CreatePDF_Proposal()
ConvertintoPDF()
CreateWord_Proposal()
ConvertintoWord()
Export_Proposal()


Export Software
Requirements Document
User
Server
CreatePDF_Requirements()
ConvertintoPDF()
CreateWord_Requirements()
ConvertintoWord()
Export_SRD()


User Authentication and Registration
User
Server
Register_Account()
Approve()
Activate_Account()
Authentiacate_Details()
Verfiy Email


Chatbot's Requirement Gathering
User
Chatbot
NLP Model
Inquire_Requirements()
Process_Decision Making()
Finalize_Requirements()
Gather_Requirements()
Inquire_Requirements Listing()
Provide_Listing()
Update_Listing()
Self_Training()


User Login and Forgot Password
User
Server
Database
Login_Request()
Fetch_Details()
Verify_Details()
Response()
Forgot_Password()
Send_Email()
Change_Email|Username()


Chatting
User
Chatbot
Create_Chat
Create_User Session
NewChat_Environment
Participate_inChat
Response_Chat

Appendix C: To Be Determined List

TBD-1 Increase Scope of the Chatbot

Description: Increasing the scope of the chatbot from E commerce specific to accommodate Content Management Systems requirements too.

Reason for TBD: Will be decided after the detailed feedback of the working of the chatbot on the E-commerce project's requirements elicitation.

TBD-2 Prioritize Project Requirements

Description: Adding additional feature of prioritizing the project's requirements once they are elicited.

Reason for TBD: Will be decided after the detailed feedback of the working of the chatbot on the requirements elicitation phase.