Coimbatore Institute of Technology

Department of AI and DS

Multiple Chat User System
Software Requirements Specification

Version: ***1.0d***

ADITHYA B
KATHIRESAN
GOWTHAM KUMAR
NITHIN SIVAKUMAR
NISHANTH

Date: 22.10.2023

Date	Modifications	Reason	Version
1/2/2006	Create a draft version of the document.	Adapted from various IEEE standards on software requirements specification	0.7d
15/2/2006	Add guidelines and faked examples as italics.	For students to follow	0.8d
1/3/2006	Minor amendments	As recommended by reviewer	0.9d
14/3/2006	Modify document content for the MCS	Adapted for Assignment 1 of Semester 1	1.0d
…			

Table 1.0 Modification History

Contents

1. Introduction

1.1 Document Purpose

This document presents the software requirements, as determined during initial analysis by creating multiple chat user system (MCS) which is more helpful for discussion and forums. Specifically, the document details the functional requirements and constraints for the computer software to be developed by CIT AI firm for the Client.

This document will be subject to formal/informal review by the CIT AI firm development team and the Client, and will form the basis for ongoing development of software, by CIT AI firm, to meet the requirements of the Client.

This document is the main deliverable output from the Requirements Elicitation Activity. It is intended to be a baseline, to supply sufficient requirements information to the Client to provide a foundation for subsequent software assessment and approval. Further, it also provides the development team with a basis for on-going software design.

1.2 About the Project

The aim of this project is to perform the 'Requirements Elicitation and Specification' activity. All tasks associated with this activity shall be performed based on the theoretical support provided as part of 'Software Engineering/Introduction to Software Engineering' course. It is assumed that other related/required activities involved in the development of the system shall be completed under different project(s).

1.3 Document Scope

In an SRS (Software Requirements Specification) document for a multi-user chat system, the "covered" section typically includes detailed specifications and requirements for the system's core functionality, such as user registration, chat room creation, real-time messaging, user authentication, and message storage.

The "not covered" section outlines aspects that the system will not address, such as specific hardware requirements, third-party integrations, and non-functional requirements like performance benchmarks, which might be addressed separately. This section clarifies the project's scope and ensures stakeholders understand the limitations of the chat system.

Terminology Used

All major definitions should be explained here. Add major terms and acronyms to the table below.

Term/Acronym	Definition/Description
Developers	The team responsible for the complete development of the software system.
API Key	A code or token used to authenticate access to a service or API.

DBMS - Database Management System	Software that manages and organizes data in a database.
QA - Quality Assurance:	The process of ensuring that the chat system meets quality and performance standards
SDK - Software Development Kit:	A set of tools and libraries for developing software applications for a specific platform.
UI - User Interface:	Refers to the visual components and design of the chat system that users interact with
UX - User Experience	Describes the overall quality of the user's interaction with the system, including ease of use and user satisfaction.

Table 2.0 Definitions, acronyms and abbreviations

1.4 Related Documents

- **Project Plan**: Details the timeline, resources, and milestones for the development process.

- **Design Specifications**: Documents that describe the architectural and design choices made for the chatbot, including flowcharts and system diagrams.

- **API Documentation**: If the chatbot interacts with other services or systems, the API documentation for those services may be referenced.

- **Data Models**: Documents outlining the structure and relationships of data used by the chatbot.

- **Testing Plan and Test Cases**: Details on how the chatbot will be tested, along with specific test cases.

- **User Interface (UI) Design Documents**: Describes the layout, visuals, and user experience design for the chatbot.

- **Project Charter**: Provides an overview of the project's goals, stakeholders, and high-level objectives.

1.5 Document Overview

This document, the Software Requirements Specification (SRS), identifies the software requirements for the project System Name.

This document has 6 major sections and *y* appendices:

- **Introduction** (Section 1) provides an overview of the entire SRS document, the project and the product being specified.

- **Product Overview** (Section 2) describes the product in overview, addressing the hardware and software requirements, user characteristics, constraints, assumptions and dependencies, expressed in natural language.
- **Functional and Non-functional Requirements** (Section 3) describes the requirements needed for developing the above project.
- **User Interface requirements** (Section 4) describes the Interface details provided for the user.
- **Operating Environment** (Section 5) describes the hardware, software, etc.
- **Appendices** (Section 6) describes the project models, task and its outcomes, etc

2. Product Overview

2.1 Introduction

The Multi-User Chat Bot specifications for a dynamic, real-time communication system. This chat bot aims to facilitate seamless interactions among multiple users, providing a feature-rich, secure, and user-friendly platform for text and media-based conversations. This document serves as a comprehensive guide to its development, ensuring all stakeholders share a common understanding of the project's scope and objectives.

2.2 Business Services Supported

- **User Registration**: Allow users to create accounts with unique usernames and passwords.
- **User Authentication**: Verify user identity during login to ensure secure access.
- **User Profile Management**: Enable users to set up and update their profiles with information and avatars.
- **Real-Time Messaging**: Facilitate instant text and multimedia messaging between users.
- **Group Chat Creation**: Allow users to create and manage chat rooms for multiple participants.
- **Privacy Controls**: Implement settings for user privacy, including blocking and reporting options.
- **Notification System**: Notify users of new messages, mentions, and system updates.
- **Message History**: Store and retrieve chat messages for users to view past conversations.
- **File Sharing**: Support the sending and receiving of files and media within chats.

2.3 Product Characteristics

The software system specified in this document is a Multi-User Chat System, designed to enable real-time, text, and multimedia communication between multiple users. It facilitates the creation of chat rooms, user profiles, and seamless interaction while offering features like privacy controls, notifications, and message history

retrieval. The system ensures secure and dynamic communication for users across various platforms

2.3.1 .User Interface

The user interface of MCS is characterized by its user-friendly design. It includes:

Chat Room Selection: Users can easily choose chat rooms from a list of available options.
Chat Window: A main chat window displays messages, and a text entry box allows users to send messages.
Slash Commands: Users can utilize slash commands for various action within chat rooms.
Multimedia Integration: Integration of webcams for video conferencing and file sharing.

2.3.2 Operating Environment

The multi-user chat system operates in a versatile environment, compatible with common web browsers (e.g., Chrome, Firefox, Safari) and mobile applications on major platforms (iOS and Android). It relies on stable internet connectivity and standard hardware, making it accessible to a wide range of users across devices and operating systems.

2.3.3 Hardware Interfaces

The multi-user chat system requires standard hardware components such as computers, smartphones, and tablets, all equipped with internet connectivity. No specific hardware interfaces are necessary for its operation.

2.4 User Characteristics

This section shall describe the various types of users that are expected to use this system and what the minimum requirements are in terms of their knowledge and skills.

The users of this MCS system shall be:

- Multimedia Group employee
- Social Network Users
- Business Teams
- Customer Support,etc.

2.4.1 General User Characteristics

All users of the MCS can be assumed to have the following characteristics:

- **Usage Context**: Different purposes for using the system, such as social interaction, work collaboration, or learning.

- **Geographic Location**: Users from various regions, potentially spanning different time zones and culture

- **Privacy Concerns**: Users with varying levels of concern about data privacy and sharing personal information

2.4.2 Multimedia group Admin Characteristics

- **User Support**: Providing guidance and assistance to group members, including helping new users navigate the system and its features.
- **Decision-Making**: Admins may need to make decisions regarding user sanctions, content removal, or chat room modifications when necessary.
- **Adherence to Guidelines**: Admins should adhere to and enforce the chat system's guidelines, including maintaining a safe and respectful environment.
- **Security Awareness**: Awareness of potential security risks and measures to protect chat room data and user information.

2.4.3 Characteristics

For any additional user categories, characteristics will depend on the specific roles and requirements of these users. These may include, but are not limited to:

- **Role-Specific Knowledge:** Users in this category should possess knowledge specific to their role or responsibilities within the MCS.
- **Technical Expertise:** Depending on the role, users may require advanced technical skills for certain functionalities.
- **Collaboration Skills:** Users may need strong collaboration and teamwork skills to make the most of the MCS's features.

Domain Knowledge: Certain users may need domain-specific knowledge related to the subject matter of their discussions

2.5 General Constraints

General design/implementation constraints include:

- The software system will be developed on and run under Windows 2000.
- All code shall be developed to run on a Java Virtual Machine supporting Java 1.3.1 or above.
- Plain text files shall be used for all text-based data storage and reports.
- The documentation and code shall be in accord with relevant CIT AI firm documentation and style standards.

2.6 Priority of requirements

- **Security Requirements:** Ensuring the security of user data and conversations is often the highest priority.
- **User Authentication:** Implementing a robust user authentication system to control access to the chat system.
- **Real-time Messaging:** Enabling real-time messaging functionality, as it is a core feature of a chat system.
- **Scalability:** Designing the system to scale efficiently as the number of users and messages increases.
- **User Roles and Permissions:** Defining different user roles and permissions to control who can perform certain actions within the system.

- **User Interface (UI) and User Experience (UX):** Creating an intuitive and user-friendly interface to enhance the overall user experience.

3. Functional Requirements

- **1**. **User Registration and Authentication:**The system should allow users to create accounts.Users should be able to log in securely using usernames and password

- **2**. **Chat Rooms**:Users should be able to create chat rooms.Users can join or leave chat rooms.Chat rooms can have different access settings, such as public, private, or invite-onlyChat rooms can have a description and topic.

- **3. Real-Time Messaging**:Users in a chat room should be able to send real-time text messages. Messages should support basic formatting and emojis.Messages should be displayed in a chronological order.
- **4**.**Search and History:** Users should be able to search for past messages in a chat room.The system should maintain a history of messages for reference

- **5. Private Messaging:**Users should be able to initiate private, one-on-one chats with other users.Private messages should be secure and not visible to others in the chat room.

- **6. Multi-Platform Compatibility**: The chat system should be accessible on multiple platforms, including web, mobile apps, and desktop applications.

- **7. User Blocking and Reporting:**Users should be able to block or mute other users.Users should have the ability to report abusive behavior.

- **8. Presence and Typing Indicators:** The system should display indicators for when a user is typing and when users are present in a chat room.

- **9. Scalability:** The system should be able to handle a large number of concurrent users and chat rooms.

- **10. Backup and Recovery**: The system should regularly back up chat data and have a recovery plan in case of data loss.

3.1 Registering to the chat system

The system shall support registration with all information's

Function Name: Chat_Register

3.1.1 Trigger

- A person wants to register to entering to the chat system

3.1.2 Pre-Conditions

- The person is a employee of multimedia group
- Registration need to be selected

3.1.3 Post-Conditions

Either

- Entering personal information's for creating account

or

- Use existing account information

3.1.4 Business Rules Applicable

- Registration can be made at anytime
- Multiple accounts can be created

3.1.5 Data Manipulated

Type	Name	Format	Validation Rules
Input	Start time	hh:mm	Between 08:00 & 20:00
Input	Duration	Numeric	1, 2 or 3
Input	Chat ID	Numeric	1, 2 or 3
System	Password	Alpha-numeric	Max 8 characters
System	System date	dd/mm/yyyy	
System	System time	Hh:mm	

Table *3.0* Inputs required for Chat _Register

3.1.6 Normal Processing Procedure

- User enters to chat system
- User creates account
- User enters user id and password
- System validates login information
- User establishes the connection to server
- If user is available, system displays all available users
- User selects one of the available users.
- User start chatting by sending message
- System exits while user is offline

3.1.7 Variations

- If user is idle, system displays "Sorry, user is idle for long time".
- System prompts user to quit
- Reports Generated

No reports are generated for this function.

3.1.8 Exceptions

Exception	Error Message (Displayed/Generated/Logged)
Creating account with same name	Already name exists
Creating password with undefined symbols	Type your password correctly

Table *4.0* Exceptions during Chat_ register

3.2 Creating an Private session in chat

The system shall support establishing private session

Function Name: Chat_ Private

3.2.1 Trigger

- The private session is triggered when a user initiates a request for a one-on-one private conversation

3.2.2 Pre-Condition

- The involved users must be authenticated and logged into the system.

3.2.3 Post-Condition

- A private session is established between the specified users, allowing them to exchange messages exclusively.

3.2.4 Business Rules Applicable

- The system should enforce privacy rules to ensure that only the designated users have access to the private session.

3.2.5 Data Manipulated

Type	Name	Format	Validation Rules
User	User IDs	Alphanumeric	Valid user IDs of participants
Text	Message	String	Length and content validation

Table *5.0* Inputs required for BOOK_GOLF

3.2.6 Normal Processing Procedure

1. User selects another user to initiate a private session.
2. System validates user credentials and authorization.
3. Private session is created, and both users are notified.
4. Users can exchange messages exclusively within the private session.

3.2.7 Variations

- If a user is not online, the system may provide options to send an invitation or notify the user upon login.

3.2.8 Reports Generated

- A log of private sessions established, including user IDs, timestamps, and session duration.

3.2.9 Exceptions

Exception	Error Message (Displayed/Generated/Logged)
User Authentication Failure	"Invalid credentials. Please log in again."

Table *6.0* Exceptions during Chat_Privat

4. User Interface Requirements

1. User Registration:

The user registration page shall include fields for username, email, password, and confirmation of password.

The registration form should have validation for email format and password strength.

Upon successful registration, a confirmation message should be displayed to the user.

2. Chat Interface:

The chat interface shall display messages in a conversation format, with the user's messages on one side and the recipients on the other.

A text input field should be provided for users to enter their messages.

A "Send" button shall be available for sending messages.

3. Navigation Menu:

A navigation menu shall be present, allowing users to access various sections of the application, such as chat rooms, user settings, and help.

The navigation menu should remain consistent throughout the application.

4. Profile Management:

Users should be able to edit their profiles, including changing their profile picture and personal information.

Users can upload a profile picture, which will be displayed next to their messages in the chat.

4.1 Look and Feel Requirements

1. Colour Scheme:

The application shall use a colour scheme of blue and white to provide a clean and professional look.

The chat interface should have contrasting colours for user messages and system messages.

2. Font Style and Size:

The font used in the application shall be a Sans serif typeface for readability.

The font size should be consistent and user-friendly, ensuring legibility on various devices.

3. Icons and Graphics:

The application may use icons to represent different functions or actions, enhancing user experience.

Icons should be intuitive and easy to understand.

4.2 Usability Requirements

1. Ease of Use:

The chat interface shall be intuitive and easy for users to navigate.

Users should be able to start a chat, send messages, and access user settings with minimal effort.

2. Efficiency:

The application should load quickly, and messages should be sent and received in near real time.

Users should not experience significant delays in message delivery.

3. Consistency:

The application's layout and design should be consistent across different screens and sections.

The placement of buttons, menus, and navigation elements should remain uniform.

4. Accessibility:

The application shall be designed to be accessible to individuals with disabilities, complying with WCAG guidelines.

It should support screen readers and provide alt text for images.

5. User Feedback:

The application should provide feedback to users, such as message sent confirmations and error messages for unsuccessful actions.

User feedback should be clear and informative

5. Non-Functional requirements

- **Response Time**: The system shall respond to user requests within 2 seconds.

- **Scalability**: The system should be able to handle a 20% increase in user load without a significant performance degradation

- **Availability**: The system should be available 99.99% of the time, allowing for less than 1 hour of downtime per year.

- **Fault Tolerance**: The system should continue to operate correctly in the presence of hardware or software failures.

- **Authentication**: Users must authenticate using a two-factor authentication method.

- **Data Encryption**: All sensitive data in the system should be encrypted at rest and during transmission.

- **Access Control**: Only authorized personnel should have access to specific system functions or data.

- **Modularity**: The system should be designed with clear and independent modules to facilitate future updates.

- **Documentation**: Maintain up-to-date documentation for system architecture, code, and configurations.

- **Version Control**: Use version control for source code and configuration files to track changes

- **Accessibility**: The system should conform to WCAG 2.1 AA accessibility standards.

- **User Training**: Provide user training and support to ensure efficient system usage.

6. Operating Environment

The system specified shall operate on the hardware, software requirements outlined in this section.

6.1 Hardware

The system is expected to have the following minimum configuration:

- Pentium-200MHz (equivalent or above).
- 128 MB RAM.
- 20MB free hard disk space.
- VGA Monitor.

6.2 Software

The basic software required for effective operation shall be:

- Windows 2000 operating system.
- Java Virtual Machine capable of running Java v1.3.1 applications or above.

6.3 Printer

The printer must be compatible with the computer system and must be capable of printing text files.

6.4 External Data Storage

All data that needs to be stored on external storage devices for security and protection reasons shall be clearly identified. Descriptions of all data files and methods of access to archive these data items shall be provided as part of the User Manual. It is expected that the technical operator of this system shall have the required expertise to transfer all relevant data files on to an external archive (e.g. Floppy Disk, CD and ZIP disk).

7. Acceptance Criteria

1. Security Measures:

The system must implement security measures to protect user data and conversations from unauthorized access. Security is paramount, especially in a communication system handling sensitive information.

2. Notification System:

Users should receive timely notifications for new messages and important events. An effective notification system is crucial for keeping users informed and engaged in the conversation

3. User Authentication:

The system must ensure that only authenticated users can access the chat system. This is crucial for maintaining the security and integrity of user interactions.

4. Scalability:

The chat system must be scalable to handle an increasing number of users and messages. Scalability ensures that the system can accommodate growth without compromising performance.

5. Realtime Messaging:

The chat system must facilitate realtime messaging with minimal latency. Realtime communication is a fundamental aspect of a chat system and directly impacts user experience

Glossary

1. Data Storage and Reports:

Storing all text-based data and reports within the MCS in plain text files for simplicity and portability.

2. Scalability:
The system's capability to efficiently handle an increasing number of users and messages.

3. User:
An individual operating or using the software system.

4. Chat Register:
The process of creating a user account to enter the chat system.

5. Requirement:
A condition or capability necessary for a user to solve a problem or achieve an objective.

6. Specification:
A document that precisely and verifiably outlines the requirements, design, behaviours, or other characteristics of a system or its components.

7. User Authentication:
The process of verifying and validating user credentials to control access to the chat system

8. Operating System Compatibility:
The system's ability to run under the Windows 2000 operating system.

9. Java Compatibility:
The software system developed using Java and running on a Java Virtual Machine (JVM) supporting Java version 1.3.1 or above.

10. Developers:
The individuals responsible for the comprehensive development of the software system.

11. Notification System:
The feature ensuring users receive timely notifications for new messages and important events

12. Chat Private:
Functionality supporting the establishment of private sessions for oneonone conversations in the chat system..

13. Realtime Messaging:
The feature of facilitating instant messaging with minimal latency.

14. Process:
A series of steps or actions undertaken to achieve a particular result.

15. Documentation and Coding Standards:

Adherence to relevant COIMBATORE INSTITUTION OF TECHNOLOGY AI firm documentation and style standards for the documentation and code.

Appendices:

Appendix A: External Context Model

1. Network: The communication infrastructure supporting the chat system. It includes the channels through which users connect, exchange messages, and interact with the system. This could involve internet connectivity, servers, and other networking components.

2. Users: These are the individuals who actively engage with the chat system, creating accounts, participating in conversations, and utilizing the system's features.

3. External Systems: Refers to the integration points where the chat system connects with databases or external APIs. This could involve fetching user data, integrating with third-party services, or accessing additional information to enhance the chat experience.

Appendix B: Internal Context Model

1. User Data Management: This module handles the collection and secure storage of user data. It includes information such as user preferences, chat histories, and personal details. Security measures are implemented to protect sensitive user information.

2. Integration Layer: Responsible for enabling communication with external systems and databases. It facilitates the exchange of data between the chat system and external entities, ensuring seamless integration and access to relevant information.

3. Chatbot Engine: The core of the chat system, managing the functionality and responses of chatbots. It interprets user inputs, processes requests, and generates appropriate responses, creating a conversational experience.

4. User Interface Module: This module is responsible for managing the interaction between users and the system. It includes the design and functionality of the user interface, allowing users to navigate, send messages, and access various features.

Appendix C: Use Case Analysis Notes

1.User Interaction with Chatbots

Description: Users interact with multiple chatbots simultaneously.
Actors: User, Chatbots, System
Flow:

1. User engages with chatbots.
2. System handles interactions seamlessly.

2.Automation of Chatbot Tasks

Description: Chatbots automate tasks like answering questions or processing orders.

Actors: Chatbots, User

Flow:

1. Chatbots efficiently automate specified tasks.

3.Creating a Private Session in Chat

Description: Users initiate private conversations.

Actors: User, System

Flow:

1. User selects another user for a private session.
2. System validates user credentials.
3. Private session established.

4.Registering to the Chat System

Description: Users create an account to access the chat system.

Actors: User, System

Flow:

1. User initiates the registration process.
2. User provides necessary information.
3. System validates information.
4. User gains access to the chat system.

5. Realtime Messaging

Description: Realtime messaging functionality in the chat system.

Actors: User, System

Flow:

1. Users experience realtime messaging.
2.

6.Integration with External Systems

Description: System integrates with external databases, systems, and APIs.

Actors: System, External Systems

Flow:

1. System connects with external systems.
2. Realtime data exchange occurs.

7.Security Measures Implementation

Description: System implements security measures for user data protection.

Actors: System, User

Flow:

1. Security measures ensure confidentiality and integrity

8.User Data Management

Description: System collects and manages user data securely.

Actors: System, User

Flow:

1. System collects user data.
2. Data stored securely and accessible to authorized entities.

9. Scalability of the System

Description: System scales horizontally to handle increased users and messages.

Actors: System, User

Flow:

1. System accommodates growth without performance compromise.

10.Error Handling in Chatbots

Description: Chatbots gracefully handle errors.

Actors: Chatbots, User

Flow:

1. Chatbot encounters errors.
2. Clear responses provided to the use

Tasks Undertaken

Task Name	Description	Technique(s) Used
Requirements	Requirements needed for the development of MCS	Online sources ,Book based on the topics.
Draft Creation	The first draft version of the document.	Adapted from various IEEE standards on software requirements specification
Modify Content	Modifying content for the MCS as per Assignment	Assignment requirements

Table *51.0* Tasks undertaken in the development of this document

Task Outputs

Task Name	Outputs	Section(s)
Document Overview	Introduction	1.0
	Scope	1.3
	About project	1.2
Requirements	Functional and Non-functional	3.0, 5.0
Appendices	Contributions & outcomes.	Last section.

Appendix E: Contributions

This section contains the names of all the contributors to this document. The sections that each contributor has worked on shall be described in the table below.

Contributor Name	Sections Worked On
ADITHYA B	INTRODUCTION AND DOCUMENT OVERVIEWS.
KATHIR	OVERVIEWS.
GOWTHAM	CHARACTERISTICS OF PRODUCT.
NITHIN SIVAKUMAR	REQUIREMENTS.
NISHANTH	APPENDICES

Table *52.0* Document Contributions

Appendix F: Meeting Agendas/Minutes

This section includes agendas and minutes from all the various meetings that team members have attended as part of this project.

ID	Meeting Date/Time	Apologies
1	20.10.2023/9pm-12pm	none
2	21.10.2023/1pm-6pm	none
3	22.10.2023/11am-3pm & 7pm-12pm	none

Table *53.0* Meeting dates/times

Appendix G: Problem Investigation Reports

1. Issue: Integration Challenges with External Systems

Date Reported: 21.10.2023

Description: Faced difficulties in establishing seamless integration with external databases, causing delays in realtime data exchange.

Impact: Slowed down overall system performance and responsiveness.

Resolution: Collaborated with external system providers, identified compatibility issues, and necessary adjustments to ensure smooth integration.

2. Issue: User Authentication Failures

Date Reported: 21.10.2023

Description: Users experienced difficulties in authenticating, leading to login failures.

Impact: Impacted user access and system security.

Resolution: Identified and fixed a bug in the authentication process, ensuring users could log in without issues.

3. Issue: Unanticipated Scalability Challenges

Date Reported: 22.10.2023

Description: System faced performance issues as user interactions increased beyond initial expectations.

Impact: System slowdown during peak usage times.

Resolution: Implemented horizontal scaling strategies, optimizing system architecture to handle a larger user base without compromising performance.

Lessons Learned:

Lesson 1: Optimization for scalability are critical during system design.

Lesson 2: Common communication with external system providers is essential for addressing integration challenges.

Lesson 3: Swift response to authentication issues is crucial for maintaining user trust and **system security**.