

CIT AI Tech Consulting

**Department of AI and DS
Coimbatore Institute of Technology**

Muti user Chat System (MCS) Software Test Document

Version: *1.0d*

Document Id: MCS-std-*1.0d*

Authors: Akshay Karthick M S

Bharath Kumar R

Dhanush Kumar K

Rohithraj N M

Reviewed By: Dr. Valliappan Raman (Head Of Department – AI & DS)

Date: 08.12.2023

Modification History

Table 1 Modification History

Date	Modifications	Reason	Version
29/11/2023	Initial Draft	-	0.1d
29/11/2023	Update test cases	Enhancement of functionalities	0.2d
29/11/2023	Incorporate feedback from review	Address feedback from review team	0.3d
30/11/2023	Add new test case scenario	Include additional test scenarios	0.4d
30/11/2023	Conduct UAT	Initiate User Acceptance Testing	0.5d
30/11/2023	Resolve UAT issues	Address issues identified during User Acceptance Testing	0.6d
01/12/2023	Modify test script for FT	Update test case for Functional Testing	0.7d
01/12/2023	Update test case based on FT	Reflect changes from Functional Testing	0.8d
02/12/2023	Approve final version for release	Document readiness for release	1.0d

Contents

1. Introduction	1
1.1 Document Purpose	1
1.2 About the Project.....	1
1.3 Document Scope	1
1.4 System Overview	1
2. Software Test Plan	2
2.1 User Acceptance Test Plan.....	2
2.2 Functional Test Plan	3
3. Software User Acceptance Test Plan.....	3
3.1 Test Case Design for “Chat Registration”.....	3
3.1.1 Test Case Specification and Design for “Registration based upon the valid username and password”.....	3
3.1.2 Test Cases Specification and Design for “Create a Public or Private chat session”.....	5
3.1.3 Test Cases Specification and Design for “To create an account”	6
3.2 Test Case Design for “creating private chat”.....	7
3.2.1 Test Case Specification and Design for “creating private chat”	7
Features to be tested: “Multimedia Sharing”	7
3.2.2 Test Case Specification and Design for “Creating private chat”	8
3.2.3 Test Case Specification and Design for “creating private chat”	9
3.3 Test Case Design for “Real-time messaging”	10
3.3.1 Test Case Specification and Design for “Real-time messaging”	10
Table 10	12
3.3.2 Test Case Specification and Design for “Real-time messaging”	12
Table 11	13
3.4 Test Case Design for “Roles and Permissions”	13
Features to be tested: “Administrator role assignment”	13
Features to be tested: “User role verification”	15
Features to be tested: “Role Modification”	16
Table 14	17
4. Software Functional Test Plan	18
4.1 Test Case Design for “Creating a Private or Public session”.....	18
4.1.1 Test Case Specification and Design for “create a session for chat system “.	18
4.1.2 Test on the number of users”	20
4.2 Test Case Design for “Video and voice chat”	22
4.2.1 Test Case Specification and Design for “Video and voice chat “.	22

4.2.2	Test Case Specification and Design for “Video and voice chat “.....	23
4.2.3	Test Case Specification and Design for “Video and voice chat “.....	25
4.3	Test Case Design for “Status information”	26
4.3.1	Test Case Specification and Design for “Status information “.....	26
4.3.2	Test Case Specification and Design for “Status information “.....	28
4.3.3	Test Case Specification and Design for “Status information “.....	29
Appendix A: Definitions, Acronyms, and Abbreviations		32
Appendix B: Contributions.....		33

1. Introduction

1.1 Document Purpose

This document presents the software test plan prepared by CIT AI Tech Consulting (henceforth referred to as “MCS”) developed for the Multimedia concern (henceforth referred to as “Client”). Specifically, the document details the user acceptance test plans and the functional test plans for the computer software to be developed by CIT AI Tech for the Client.

This document will be subject to formal and informal reviews by the CIT AI Tech development team and the Client, and will form the basis for ongoing testing of the software being developed by CIT AI Tech, to meet the requirements of the Client.

This document forms part of the main deliverable from the software testing design activity. It is intended to be a detailed documentation for the CIT AI Tech development team to continue the quality assurance tasks such as testing as required. Sections 2.1 and 3 constitute the user acceptance test plans, whereas Sections 2.2 and 4 constitute the functional test plans of testing the MCS.

1.2 About the Project

The aim of this project is to perform the ‘Software Test Design’ activity only. All tasks associated with this activity shall be performed based on the theoretical support provided as part of the ‘HIT1031 Introduction to Software Engineering’ course. It is assumed that other related or required activities involved in the development of the system shall be completed under different project(s).

1.3 Document Scope

The document covers the ‘Software Test Design’ activity for the computer software developed by CIT AI Tech for the Multimedia concern (Client). It includes user acceptance test plans and functional test plans. The scope is limited to testing activities, excluding software development and other related tasks. This document does not address activities beyond the testing domain or pertain to the completion of the system under different projects.

1.4 System Overview

The ‘Multi user Chat System’ (MCS) is a system that allows users to discuss various information’s and report facility usages in the community of the Client.

Document Overview

This document, the Software Test Document (STD), identifies the software test plan and the details of the testing to be carried out for the MCS.

This document has 5 major sections and 2 appendices:

- **Introduction** (Section 1) provides an overview of the entire MCS system and the MCS Software Test Document.
- **Software Test Plan** (Section 2) documents the approaches used in different stages of testing from functional testing to user acceptance testing.
- **Software User Acceptance Test Plan** (Section 3) describes the details of the user acceptance testing of the software.

- **Software Functional Test Plan** (Section 4) describes the details of testing the functionalities of the system to ensure the software behaves as described in SRS.
- **Traceability of Requirements** (Section 5) documents the related requirements per test cases in this document.
- **Terms and Definitions** (Appendix A) describes the general information that helps to understand this software test document.
- **Contribution** (Appendix B) documents the contributions of each.

2. Software Test Plan

This section describes the testing plans for testing the MCS software. In general, this includes the user acceptance test (UAT) and the functional testing. Moreover, it is assumed that the next level of testing is performed only when every test plan in the previous testing level has been carried out without any error.

2.1 User Acceptance Test Plan

This section documents a high level view of the user acceptance test plan and traces the functionalities under test against the SRS of the software. In particular, the user acceptance test plan addresses the user acceptance criteria in the SRS.

Table 2 User acceptance test plan

Functionality	Requirements in SRS	Description/Main Responsibilities	Feature to be tested	Test Spec. and Design
Registering an account	Section 3.1	To create an account	Create account in database	Section 3.1.1
			Privacy and security	Section 3.1.2
			Email verification	Section 3.1.3
Creating private chat	Section 3.2	To create a private chat	Multimedia Sharing	Section 3.2.1
			Real-time update	Section 3.2.2
			User interface	Section 3.2.3
Real-time messaging	Section 3.3	To send a message	File sharing	Section 3.3.1
			Edit message	Section 3.3.2
Roles and permissions	Section 3.4	To maintain channel servers	Administrator role assignment	Section 3.4.1
			User role verification	Section 3.4.2
			Role modification	Section 3.4.3

2.2 Functional Test Plan

This section documents a high level view of the functional test plan of the software. Moreover, it traces the functionalities under test against the SRS of the software.

Table 3 System function test plan

Functionality	Requirements in SRS	Description	Feature to be tested	Test Spec. and Design
Create a Public or Private chat session	Section 3.1	Creating a Private or public chat room	While creating chat sessions, user can chat on both rooms.	Section 4.1.1
			User access	Section 4.1.2
Video and voice chat	Section 3.2	To create a video and voice chat system	Video call initiation	Section 4.2.1
			Audio quality	Section 4.2.2
			Call stability	Section 4.2.3
Status information	Section 3.3	To display status information	Automated status changes	Section 4.3.1
			Availability option	Section 4.3.2
			Setting and display	Section 4.3.3

3. Software User Acceptance Test Plan

This section describes the details of the plan for the user acceptance test (UAT). Some representatives from the Client are required for this testing session. The number of representative test cases should be subject to the mutual agreement between the Client and CIT AI Tech. The CIT AI Tech development team will demonstrate the functionalities of the MCS using the following test specifications.

3.1 Test Case Design for “Chat Registration”

This section describes the specifications of the test cases for testing how well the software handles the request of creating an account.

3.1.1 Test Case Specification and Design for “Registration based upon the valid username and password”

Features to be tested: Create account in database

3.1.1.1 User Registration:

Requirement: Users should be able to register for an account.

Expected Behaviour: The system should provide a registration form with fields for username, email, and password. Upon submission, a new user account should be created.

3.1.1.2 Input Validation:

Requirement: Validate user inputs during registration.

Expected Behaviour: The system should check for valid email formats, enforce strong password policies, and prevent common vulnerabilities like SQL injection.

3.1.1.3 Password Security:

Requirement: Ensure secure storage of passwords.

Expected Behaviour: Passwords should be hashed using a secure algorithm (e.g., bcrypt) before storage in the database.

3.1.1.4 Unique Usernames and Emails:

Requirement: Usernames and emails must be unique.

Expected Behaviour: The system should check for existing usernames and emails to avoid duplication during registration.

3.1.1.5 User Profile:

Requirement: Create a user profile upon successful registration.

Expected Behaviour: User profiles should store relevant information (e.g., username, profile picture) and be associated with the registered account.

3.1.1.6 Email Verification:

Requirement: Verify user email addresses for account activation.

Expected Behaviour: Users should receive an email with a verification link. Clicking the link should activate the account.

3.1.1.7 Notification:

Requirement: Notify users upon successful registration.

Expected Behaviour: Users should receive a confirmation message or notification upon successful account creation.

Test Case Design:

Test Id.	Inputs required	Environment needs	Output expected
3.1.1.1	Username	Accessible Registration Page	User Account Creation
	Email	Database Connectivity	Confirmation Message or Redirect
	Password	Secure Connection	Activation Email (if applicable)
3.1.1.2	Email input	Accessible Registration Page	Error Messages
	Password input	Validation Rules Configuration	Prevention of Vulnerabilities
3.1.1.3	Password	Secure Connection	Hashed Password
		Hashing Algorithm	No Storage of Plain Text Passwords
		Database Connectivity	Protection Against Password Exposure
3.1.1.4	Username	Access to Registration Page	Unique Inputs
	Email	Database Connectivity	Prevention of Duplication
3.1.1.5	Profile Information	User Account Creation	User Profile Creation
	Access to Profile Editing	Profile Page or Section	Access to Profile
		Database Connectivity	Profile Information Storage

3.1.1.6	User Email Address	Accessible Registration Page	Verification Email
		Email Service Integration	Verification Link Click
		Secure Connection	Activation Confirmation
3.1.1.7	User Account Creation	Notification System	Confirmation Message or Notification
		Access to Notification Platform	Clear Notification Content

Table 4

3.1.2 Test Cases Specification and Design for “Create a Public or Private chat session”

Features to be tested: Privacy and security

3.1.2.1 Secure Data Transmission:

Requirement: All data transmitted between users and the server should be encrypted.

Expected Behaviour: Use of secure protocols (e.g., HTTPS) to protect user data during transmission.

3.1.2.2 Password Protection:

Requirement: User passwords must be securely stored using strong encryption.

Expected Behaviour: Passwords are hashed and salted before storage to enhance security.

3.1.2.3 Access Controls:

Requirement: Users should have appropriate access controls based on their roles.

Expected Behaviour: Admins, moderators, and regular users should have different levels of access, and the system enforces these roles.

3.1.2.4 Privacy Settings:

Requirement: Users should have the ability to customize their privacy settings.

Expected Behaviour: Users can control who can see their online status, profile details, and other relevant information.

Test Case Design:

Test Id.	Inputs required	Environment needs	Output Expected
3.1.2.1	Data transmission scenario	Secure server setup	Successful Encryption
	Network simulator	Client-side configuration	Use of HTTPS
	Security testing tools	Security certificates	Secure Connection Establishment:
3.1.2.2	User Registration Form	Database Connection	Hashed and Salted Password
		Password Hashing Algorithm	Database Storage
3.1.2.3	User roles	Access control settings	Admin access
	Access permissions	User interface	Moderator access
	User accounts		Regular user access

3.1.2.4	User interaction	Application access	Customized privacy settings
	Selection of privacy option	Settings availability	Visibility control

Table 5

3.1.3 Test Cases Specification and Design for “To create an account”

Features to be tested: Email verification

3.1.3.1 User Registration:

Requirements: Users must register with a valid email address to create an account on the Multiple User Chat System.

Expected Behaviour: The system should prompt users to provide a unique and valid email address during the registration process.

3.1.3.2 Email Confirmation Link:

Requirements: After registration, an email containing a unique confirmation link should be sent to the provided email address.

Expected Behaviour: Users should receive an email with a confirmation link within a reasonable time frame. The link should be valid for a specific period.

3.1.3.3 Verification Link Validity:

Requirements: The confirmation link provided in the email should be valid for a limited time to ensure timely verification.

Expected Behaviour: Users clicking on the confirmation link within the specified time should be directed to a verification page.

3.1.3.4 Verification Page:

Requirements: The verification page should confirm successful email verification and activate the user's account.

Expected Behaviour: After clicking the confirmation link, users should be redirected to a page confirming successful verification, indicating that their account is now active.

Test Case Design:

Test Id.	Inputs required	Environment needs	Output Expected
3.1.3.1	Username	Internet connectivity	Successful registration
	Email address	Web browser	Unsuccessful registration
	Password		
3.1.3.2	User email address	Internet connectivity	Email sent
	Unique confirmation link	Email service configuration	Link validity
3.1.3.3	Confirmation link	Web server to host	Link validity check
	Timestamp	Secure https connection	Expired link
			Valid link
3.1.3.4	Verification token	Web server to host	User activation
	User id	Secure https connection	Login option

Table 6

3.2 Test Case Design for “creating private chat”

This section describes the specifications of the test cases for testing how the software handles the request of creating a private chat.

3.2.1 Test Case Specification and Design for “creating private chat”

Features to be tested: “Multimedia Sharing”

3.2.1.1 File Format Support:

Requirement: The chat system should support a variety of multimedia file formats, such as images, videos, and audio files.

Expected Behaviour: Verify that users can successfully share files in different formats without any loss of quality or functionality.

3.2.1.2 File Size Limitations:

Requirement: Define the maximum file size allowed for each type of multimedia file.

Expected Behaviour: Ensure that the system enforces file size limitations and provides appropriate error messages when users attempt to exceed these limits.

3.2.1.3 Upload and Download Speed:

Requirement: Multimedia files should upload and download at a reasonable speed.

Expected Behaviour: Test the upload and download speeds for various file sizes to ensure optimal performance.

3.2.1.4 Real-Time Updates:

Requirement: Users should receive real-time updates when someone shares a multimedia file in the chat.

Expected Behaviour: Confirm that users receive notifications promptly and that the multimedia content is visible without delay.

Test Case Design:

Test Id.	Inputs required	Environment needs	Output Expected
3.2.1.1	Test user	Test environment	Successful sharing and reception
	Multimedia files	Access to variety of multimedia files	Verification that the quality and functionality of files remain intact
3.2.1.2	Multimedia files of different file size	Test environment	Enforced file size limitations during upload
	Test user attempting to upload files exceeding the defined limits	Configuration settings defined maximum files sizes	Appropriate error messages displayed for user attempting to exceed file size limit
3.2.1.3	Multimedia files of various sizes	Test environment with varying network conditions (e.g., high-speed, low-speed).	Multimedia files upload and download at reasonable speeds

	Users initiating upload and download actions.	Monitoring tools to measure upload and download speeds	Consistent performance across different file sizes
3.2.1.4	Users sharing multimedia files in real-time	Test users monitoring for notifications.	Prompt notifications for users when multimedia files are shared.
	Test users monitoring for notifications.	Real-time messaging infrastructure	Immediate visibility of multimedia content without delay.

Table 7

3.2.2 Test Case Specification and Design for “Creating private chat”

Features to be tested: “Real-time update”

3.2.2.1 Real-time Messaging:

Requirement: Messages should be delivered to all users in real-time.

Expected Behaviour: When a user sends a message, it should appear immediately for all participants without any noticeable delay.

3.2.2.2 Message Order:

Requirement: Messages should be displayed in the order they are sent.

Expected Behaviour: Confirm that the messages are arranged chronologically, maintaining the correct order of when they were sent.

3.2.2.3 User Presence Updates:

Requirement: Users' online/offline status should be updated in real-time.

Expected Behaviour: When a user logs in or out, all other users should see the status change instantly.

3.2.2.4 Typing Indicators:

Requirement: Users should be able to see when others are typing in real-time.

Expected Behaviour: When a user starts typing, a visual indicator should appear for all participants, and it should disappear when they stop typing.

3.2.2.5. Read Receipts:

Requirement: Users should receive real-time notifications when their messages are read.

Expected Behaviour: When a user reads a message, the sender should receive a notification instantly.

Test Case Design:

Test Id.	Inputs required	Environment needs	Output Expected
3.2.2.1	User messages	Backend server with web socket support	Messages are instantly delivered to all users in real-time without noticeable delay.
	Websocket connection for real-time communication	Frontend client with WebSocket support	

3.2.2.2	User messages with timestamps	Backend server to store and retrieve messages in chronological order	Messages are displayed in the order they are sent, maintaining chronological order.
		Frontend client to display messages in the correct order	
3.2.2.3	User login/logout events	Backend server to update user presence status	Users' online/offline status is updated in real-time for all participants.
	WebSocket connection for real-time communication	Frontend client to handle real-time updates	
3.2.2.4	User typing events	Backend server to handle typing events	Visual indicators appear for all participants when a user starts typing, and they disappear when the user stops typing.
	WebSocket connection for real-time communication	Frontend client to display typing indicators in real-time	
3.2.2.5	Message read events	Backend server to handle read events	Users receive real-time notifications when their messages are read.
	WebSocket connection for real-time communication	Frontend client to display read receipts	

Table 8

3.2.3 Test Case Specification and Design for “creating private chat”

Features to be tested: “User interface”

3.2.3.1 User Registration and Login:

Requirement: Users should be able to register for an account and log in securely.

Expected Behaviour: Successful registration and login processes without errors.

Appropriate error messages for incorrect credentials.

3.2.3.2 User Profile:

Requirement: Users should be able to create and update their profiles.

Expected Behaviour: Successful profile creation and updates. Proper validation for profile information.

3.2.3.3 Contact Management:

Requirement: Users should be able to add, remove, and manage their contacts.

Expected Behaviour: Adding and removing contacts should reflect accurately.

Notifications for contact requests.

3.2.3.4 Chat Windows:

Requirement: Users should be able to open and close chat windows with individual or multiple users.

Expected Behaviour: Smooth opening and closing of chat windows. Real-time message updates.

3.2.3.5 Message Sending and Receiving:

Requirement: Users should be able to send and receive messages in real-time.

Expected Behaviour: Instant message delivery. Proper handling of large messages.

Test Case Design:

Test Id.	Inputs required	Environment needs	Output Expected
3.2.3.1	User details such as username, email, password	Database to store user registration information securely.	For Successful Registration: Confirmation message or redirection to the user's profile.
	For Login: Username/email and password	Secure connection (HTTPS) during registration and login.	For Successful Login: Redirect to the main chat interface. Appropriate error messages for incorrect credentials.
3.2.3.2	User details such as name, profile picture, status, etc.	Database to store and retrieve user profile information.	Confirmation message on successful profile creation/update. Proper validation for profile information (e.g., valid email format, image size).
3.2.3.3	Usernames or emails of contacts to be added or removed.	Database to store and manage user contacts.	Accurate reflection of added or removed contacts.
		Notification system for contact requests.	Notifications for contact requests and approvals/denials.
3.2.3.4	Selection of users for individual or group chats.	Real-time messaging infrastructure.	Smooth opening and closing of chat windows.
		Database for storing and retrieving chat history.	Real-time message updates in the chat window.
3.2.3.5	Text or multimedia content for messages.	Real-time messaging infrastructure.	Instant delivery and display of messages.
			Proper handling of large messages, with potential error messages for failures.

Table 9

3.3 Test Case Design for “Real-time messaging”

This section describes the specifications of the test cases for testing how the software handles real-time messages.

3.3.1 Test Case Specification and Design for “Real-time messaging”

Features to be tested: “File sharing”

3.3.1.1 File Upload:

Requirement: Users should be able to upload files of various types (documents, images, videos, etc.).

Expected Behaviour: Successful upload of files with different formats and sizes. Appropriate error messages for invalid file formats or size limits.

3.3.1.2 File Download:

Requirement: Users should be able to download files shared in the chat.

Expected Behaviour: Files should download correctly and completely. Proper validation for file integrity.

3.3.1.3 Real-time Updates:

Requirement: Users should receive real-time notifications for incoming files.

Expected Behaviour: Immediate notification when a file is shared. Accurate display of file information (sender, file name, timestamp).

3.3.1.4 Security:

Requirement: File transfers should be secure and encrypted.

Expected Behaviour: Encrypted transmission of files to ensure data security. Protection against potential vulnerabilities, such as file injection or malware.

3.3.1.5 File Management:

Requirement: Users should be able to manage shared files, including deleting or saving them.

Expected Behaviour: Accurate reflection of file status (deleted, saved). Proper handling of file deletion, ensuring it does not impact other users.

Test Case Design:

Test Id.	Inputs required	Environment needs	Output Expected
3.3.1.1	User selects a file from their local device.	Secure file upload API or mechanism.	Successful upload confirmation.
	Metadata such as file name, type, and size.	Server storage for uploaded files.	Appropriate error messages for invalid file formats or exceeding size limits.
3.3.1.2	User selects a file to download.	Secure file download API or mechanism.	Correct and complete download of the selected file.
		Access control to ensure only authorized users can download files.	Validation to ensure the integrity of the downloaded file.
3.3.1.3	File sharing action by a user.	Real-time messaging infrastructure.	Immediate notification to users when a file is shared.
		Notification system.	Accurate display of file information in real-time (sender, file name, timestamp).
3.3.1.4	File being transferred between users.	Secure file transfer protocols (e.g., HTTPS).	Encrypted transmission of files to ensure data security.
		Encryption mechanisms for file contents.	Protection against potential vulnerabilities such as file injection or malware.

3.3.1.5	User actions to delete or save shared files.	Database for tracking file statuses (deleted, saved).	Accurate reflection of file status (deleted, saved).
		Access control to manage file actions.	Proper handling of file deletion, ensuring it does not impact other users.

Table 10

3.3.2 Test Case Specification and Design for “Real-time messaging”

Features to be tested: “Edit message”

3.3.2.1 Edit Message Functionality:

Requirement: Users should be able to edit messages they have sent within a specified time window.

Expected Behaviour: Successful editing of messages within the designated time frame. The original message content should be updated, and real-time updates should be reflected for other users in the chat. Appropriate error messages should be displayed for attempts to edit outside the allowed time window.

3.3.2.2 Edit Message Visibility:

Requirement: Real-time updates should be visible to all users in the chat when a message is edited.

Expected Behaviour: Instantaneous updates for all users in the chat when a message is edited. The edited message should clearly indicate that it has been modified. Users should be notified of the edit, especially if they are currently viewing the message.

3.3.2.3 Edit Message Permissions:

Requirement: Only the user who sent the original message should be allowed to edit it.

Expected Behaviour: Editing options should only be available to the user who sent the message. Attempts by other users to edit a message should result in an appropriate error message.

3.3.2.4 Editing Window:

Requirement: There should be a specified time window within which a user can edit their message.

Expected Behaviour: Users should only be able to edit messages within the defined time frame. Attempts to edit outside this window should be disallowed, and users should be notified accordingly.

3.3.2.5 Timestamps and Audit Trail:

Requirement: Edited messages should display both the original timestamp and the timestamp of the edit.

Expected Behaviour: Timestamps should accurately reflect the time of the original message and the time of the edit. The system should maintain an audit trail of edits for accountability and review.

Test Case Design:

Test Id.	Inputs required	Environment needs	Output Expected
3.3.2.1	User sends a message.	Multiple users in a chat system.	Real-time updates reflecting the edited message for other users.
	User attempts to edit the message within the specified time window.	Message editing functionality enabled.	Successful editing of the message.
3.3.2.2	User sends a message.	Multiple users in a chat system.	Instantaneous updates for all users in the chat.
	User or another user edits the message	Real-time messaging enabled.	Clear indication that the message has been modified.
3.3.2.3	User sends a message.	Multiple users in a chat system.	Editing options available only to the user who sent the message.
	Another user attempts to edit the message.	Message editing permissions configured.	Appropriate error message displayed for attempts by other users to edit the message
3.3.2.4	User sends a message.	Multiple users in a chat system.	Successful editing within the defined time frame.
	User attempts to edit the message within and outside the specified time window.	Message editing time window configured.	Attempts to edit outside the window disallowed.
3.3.2.5	User sends a message	Multiple users in a chat system.	Edited messages display both the original timestamp and the timestamp of the edit.
	User edits the message	Message timestamp and audit trail configured.	Timestamps accurately reflect the time of the original message and the time of the edit.

Table 11

3.4 Test Case Design for “Roles and Permissions”

This section describes the specifications of the test cases for testing how the software handles the roles and permissions of the user.

3.4.1 Test Case Specification and Design for “Roles and permission”**Features to be tested: “Administrator role assignment”****3.4.1.1 Role Assignment:**

Requirement: The system should allow designated administrators to assign the administrator role to other users.

Expected Behaviour: Designated administrators can assign the administrator role to other users via a user management interface. Upon successful assignment, the user immediately gains administrator privileges.

3.4.1.2 Administrator Privileges:

Requirement: Users with administrator roles should have additional privileges, such as managing users, moderating content, and performing administrative tasks.

Expected Behaviour: Users who receive the administrator role immediately gain associated privileges without reauthentication. The system clearly communicates the administrator privileges to the user.

3.4.1.3 Revocation of Administrator Role:

Requirement: Administrators should be able to revoke the administrator role from other users.

Expected Behaviour: Designated administrators can revoke the administrator role from users via a user management interface. Upon successful revocation, the user immediately loses administrator privileges.

3.4.1.4 User Notifications:

Requirement: Users being assigned or revoked the administrator role should receive notifications about the change in their status.

Expected Behaviour: Users affected by role changes receive clear and timely notifications about their new status, including whether they have been assigned or revoked the administrator role.

3.4.1.5 Security:

Requirement: Implement security measures to prevent unauthorized users from assigning or revoking administrator roles.

Expected Behaviour: Unauthorized attempts to assign or revoke administrator roles are blocked. The system provides clear error messages and takes appropriate security measures to safeguard against unauthorized actions.

Test Case Design:

Test Id.	Inputs required	Environment needs	Output Expected
3.4.1.1	Designated administrator credentials.	User management interface accessible to administrators.	Successful assignment of the administrator role to the specified user.
	User to whom the administrator role will be assigned.	List of users and their roles.	Immediate granting of administrator privileges to the user.
3.4.1.2	User who has been assigned the administrator role.	System with various administrative privileges defined.	User gains associated administrator privileges immediately after role assignment.
3.4.1.3	Designated administrator credentials.	User management interface accessible to administrators.	Successful revocation of the administrator role from the specified user.
	User from whom the administrator role will be revoked.	List of users and their roles.	Immediate loss of administrator privileges by the user.

3.4.1.4	User whose role is being changed (assigned or revoked).	Notification system integrated into the chat platform.	User receives a clear and timely notification about the change in their role status.
3.4.1.5	Unauthorized user attempting to assign or revoke administrator roles.	System with implemented security measures.	Unauthorized attempts to assign or revoke administrator roles are blocked. System provides clear error messages for unauthorized actions.

Table 12

3.4.2 Test Case Specification and Design for “Roles and permission”

Features to be tested: “User role verification”

3.4.2.1 Role Assignment:

Requirement: The system should allow administrators to assign different roles to users.

Expected Behaviour: Designated administrators can assign roles to users based on predefined roles in the system.

3.4.2.2 User Role Verification:

Requirement: The system should verify and display the assigned role for each user.

Expected Behaviour: When a user logs in, their assigned role should be verified, and the system should accurately display their role information.

3.4.2.3 Role-Based Access Control (RBAC):

Requirement: Users should only be able to access features and functionalities associated with their assigned role.

Expected Behaviour: Users can access only the features and functionalities corresponding to their assigned role. Attempts to access unauthorized functionalities result in appropriate error messages.

3.4.2.4 Role Modification:

Requirement: Administrators should be able to modify a user's role.

Expected Behaviour: Changes in a user's role by administrators should be reflected immediately, affecting the user's access to different functionalities.

3.4.2.5 Notification on Role Change:

Requirement: Users should receive notifications when their role is modified.

Expected Behaviour: Users are promptly notified when there is a change in their assigned role, providing transparency and awareness.

Test Case Design:

Test Id.	Inputs required	Environment needs	Output Expected
3.4.2.1	Designated administrator credentials.	Access to the role assignment interface in the admin panel.	Successful assignment of roles to users.

	List of predefined roles in the system.	User accounts with different roles.	System updates reflecting the changes in the assigned roles.
3.4.2.2	User credentials for login	User account with an assigned role.	Successful verification of the assigned role during the login process.
		Access to the login interface.	Accurate display of the user's role information.
3.4.2.3	User credentials for login.	User accounts with specific roles.	Successful access to functionalities associated with the assigned role
	Attempt to access functionalities beyond the assigned role.	System configured with role-based access control.	Appropriate error messages for attempts to access unauthorized functionalities.
3.4.2.4	Designated administrator credentials.	Access to the role modification interface in the admin panel.	Successful modification of a user's role.
	User credentials for whom the role will be modified.	User accounts with different roles.	Immediate reflection of the role change affecting the user's access.
3.4.2.5	User credentials.	Configured notification system.	Prompt notification to users when there is a change in their assigned role.
	Role modification by administrators.	User accounts with assigned roles.	Notification providing transparency and awareness of the role change.

Table 13

3.4.3 Test Case Specification and Design for “Roles and permission”

Features to be tested: “Role Modification”

3.4.3.1 Role Assignment:

Requirement: The system should allow administrators to modify the roles assigned to users.

Expected Behavior: Administrators should be able to change the role of a user to another predefined role.

3.4.3.2 Permission Changes:

Requirement: Role modification should result in corresponding changes to the user's permissions.

Expected Behavior: When a user's role is modified, their permissions should be updated accordingly, granting or revoking access to specific features.

3.4.3.3 Role Hierarchy Modification:

Requirement: Administrators should be able to modify the hierarchy of roles.

Expected Behavior: Changes in the role hierarchy should reflect accurately in the permissions and privileges assigned to each role.

3.4.3.4 Notification to Users:

Requirement: Users affected by role modifications should receive notifications.

Expected Behavior: Users should be promptly notified of any changes in their roles and permissions, ensuring transparency.

3.4.3.5 Role Modification Logging:

Requirement: The system should log all role modifications.

Expected Behavior: A comprehensive log should be maintained, capturing details such as the user involved, the old role, the new role, and the timestamp of the modification.

Test Case Design:

Test Id.	Inputs required	Environment needs	Output Expected
3.4.3.1	Administrator credentials for authentication.	Access to the role modification functionality.	Successful modification of the user's role to the predefined role.
	User ID of the user whose role needs to be modified.	Predefined roles set up in the system.	Updated user role and corresponding permissions.
3.4.3.2	Administrator credentials for authentication.	Access to the role modification and permission management functionality.	Successful modification of the user's role and permissions.
	User ID of the user whose role and permissions need to be modified.	Predefined roles with associated permissions.	Confirmation message indicating that the role and permission changes were successful.
3.4.3.3	Administrator credentials for authentication.	Access to the role hierarchy modification functionality.	Successful modification of the role hierarchy.
	Existing role hierarchy details.	Existing roles and their hierarchical relationships.	Confirmation message indicating that the role hierarchy changes were successful.
3.4.3.4	Administrator credentials for authentication.	Access to the notification system.	Users affected by role modifications receive timely notifications.
	User ID of the user affected by the role modification.	Proper integration to trigger notifications.	Users affected by role modifications receive timely notifications.
3.4.3.5	Administrator credentials for authentication.	Access to the logging system.	Comprehensive log entries capturing role modification details.
	User IDs, old roles, new roles, and timestamps for role modifications.	Proper configuration for recording role modification events.	Each log entry includes information such as user ID, old role, new role, and timestamp.

Table 14

4. Software Functional Test Plan

This section describes the details of the test plan for the system testing which includes functional testing and performance testing.

4.1 Test Case Design for “Creating a Private or Public session”

This section describes the specifications of the test cases for testing the functionality of booking a tennis session.

4.1.1 Test Case Specification and Design for “create a session for chat system “.

Features to be tested: “While creating chat sessions, user can chat on both rooms”

4.1.1.1 Room Selection:

Requirement: Users should be able to select multiple rooms they want to join within a single chat session.

Expected Behavior: Users can pick multiple rooms to join in a single session.

4.1.1.2 Simultaneous Interaction:

Requirement: Users should be able to send and receive messages in multiple rooms simultaneously.

Expected Behavior: Users can chat in multiple rooms at once—sending and receiving messages concurrently.

4.1.1.3 Distinct Notifications:

Requirement: Notifications for new messages should be specific to each room to avoid confusion.

Expected Behavior: Notifications are room-specific, preventing confusion about where new messages arrive.

4.1.1.4 User Interface Clarity:

Requirement: Clear UI elements (panels, tabs, or indicators) should enable easy navigation between different rooms.

Expected Behavior: Clear UI elements facilitate easy navigation between rooms.

4.1.1.5 Synchronization:

Requirement: Messages sent or received in one room should not interfere with conversations in other rooms.

Expected Behavior: Messages in one room don't disrupt conversations in others, ensuring seamless interaction.

4.1.1.6 Moderation and Controls:

Requirement: Moderators should have tools to manage conversations, users, or content within each room independently.

Expected Behavior: Moderators have separate tools to manage each room's conversations, users, or content.

Test Case Design:

Test Id.	Inputs required	Environment needs	Output expected
4.1.1.1	Date = 02/12/2023 Start time = 7.00 Duration = 1	An interface that lists all available rooms with options (checkboxes, dropdowns, etc.) for users to select multiple rooms simultaneously.	Clear interface displaying rooms, enabling users to select multiple rooms at once easily.
		Backend infrastructure capable of handling multiple concurrent room connections for each user session.	Strong backend handling room data and user connections concurrently without issues.
		A real-time messaging system that supports parallel communication in multiple rooms without delays or conflicts.	Communication system allowing seamless interaction in multiple rooms simultaneously, ensuring instant message delivery without interference.
4.1.1.2	Date = 02/12/2023 Start time = 8.00 Duration = 1	A messaging infrastructure allowing parallel exchanges in various rooms without delays.	Users can chat in multiple rooms simultaneously, sending and receiving messages without delays or disruptions.
		An interface facilitating seamless navigation between rooms for simultaneous interaction.	Smooth UI experience allowing users to switch between rooms easily while staying updated in each conversation.
4.1.1.3	Date = 02/12/2023 Start time = 9.00 Duration = 1	System that delivers unique alerts for each room's new messages.	Users receive notifications unique to each room's new messages, avoiding confusion about the message source.
		Customizable options enabling users to tailor their notification experience for each room.	Users can customize how they receive notifications per room, ensuring a personalized experience.
4.1.1.4	Date = 02/12/2023 Start time = 10.00 Duration = 1	Interface elements designed for intuitive movement between rooms without confusion.	Clear UI elements facilitating effortless movement between rooms for users.
		Interface that ensures consistent navigation across various screen	A responsive interface allowing smooth navigation across

		sizes for seamless user experience.	devices, ensuring a consistent experience.
4.1.1.5	Date = 02/12/2023 Start time = 11:00 Duration = 1	Messaging system keeping conversations separate to prevent interference.	Messages in one room won't affect discussions in others.
		Updates without impacting conversations in other rooms.	Users engage seamlessly within each room without disruption from messages in other rooms.
4.1.1.6	Date = 02/12/2023 Start time = 12:00 Duration = 1	Integration of tools allowing independent management within individual rooms.	Moderators possess separate tools to manage each room's discussions, users, and content independently.
		System granting moderators control over conversations, users, and content within their assigned rooms.	Effective moderation within rooms without affecting other areas, ensuring focused control within specific rooms.

Table 15

4.1.2 Test on the number of users”

Features to be tested: “User access”

4.1.2.1 Authentication:

Requirement: Users should provide credentials (username/password, biometric data, etc.) to verify their identity.

Expected Behavior: Promptly report any suspicion of unauthorized access or compromised credentials.

4.1.2.2 Authorization:

Requirement: Define roles (admin, user, guest) and permissions (read, write, delete) for different user categories.

Expected Behavior: Avoid attempting to exceed authorized permissions or manipulate privileges.

4.1.2.3 Access Control Lists (ACLs)

Requirement: Specify what resources (files, data, features) each user or user group can access.

Expected Behavior: Report any discrepancies or inconsistencies in assigned access levels.

4.1.2.4 Multi-factor Authentication (MFA):

Requirement: Implement an additional layer of security, requiring users to provide multiple forms of verification.

Expected Behavior: Cooperate with the MFA process by providing the required authentication factors promptly and accurately.

4.1.2.5 Logging and Monitoring:

Requirement: Keep records of user access, failed login attempts, and system activities to track any suspicious behavior.

Expected Behavior: Report any irregularities noticed in logs or suspicious activities observed while using the system.

Test Case Design:

Test Id.	Inputs required	Environment needs	Output Expected
4.1.2.1	Date = 02/12/2023 Start time = 13:00 Duration = 1	Robust authentication methods and encryption for secure storage.	Efficient user verification with minimal friction.
		Scalable, user-friendly interface with stringent security protocols.	Immediate reporting of suspicions or unauthorized access.
4.1.2.2	Date = 02/12/2023 Start time = 14:00 Duration = 1	Role-based system defining permissions for different user categories.	Strict adherence to assigned permissions, avoiding unauthorized actions.
		Granular control and secure assignment mechanisms for permissions.	Consistent access control based on roles, facilitating efficient access reviews.
4.1.2.3	Date = 02/12/2023 Start time = 15:00 Duration = 1	Detailed resource-specific access definitions for users/groups.	Accurate resource access as per assigned permissions.
		Secure assignment and auditing mechanisms for access control.	Prompt reporting and correction of access discrepancies.
4.1.2.4	Date = 02/12/2023 Start time = 16:00 Duration = 1	Integration of diverse authentication methods for layered security.	Prompt and accurate provision of authentication factors by users.
		User-friendly interface and compliance with security standards.	Improved security resilience against unauthorized access.
4.1.2.5	Date = 02/12/2023 Start time = 17:00 Duration = 1	Detailed logging of user access and system activities.	Identification and recording of irregularities in system logs.
		Real-time monitoring tools and secure data storage.	Prompt reporting and resolution of suspicious activities.

Table 16

4.2 Test Case Design for “Video and voice chat”

"Developing a real-time video and voice chat system for seamless and immersive communication experiences across various devices and platforms."

4.2.1 Test Case Specification and Design for “Video and voice chat “.

Features to be tested: “Video call initiation”

4.2.1.1 User Interface:

Requirement: Compatibility across different devices and platforms.

Expected Behavior: Intuitively accessible video call initiation with cross-device compatibility.

4.2.1.2 Contact Selection:

Requirement: Capability to select contacts or participants for the video call from a contact list or by entering their details.

Expected Behavior: Efficient selection from contacts or manual entry for participant invitation.

4.2.1.3 Call Invitation:

Requirement: Functionality to send invitations or call requests to selected contacts, indicating the intention to start a video call.

Expected Behavior: Prompt, clear, and informative requests sent to chosen contacts.

4.2.1.4 Permission Handling:

Requirement: Integration with device permissions to access the camera and microphone for video and audio transmission.

Expected Behavior: Immediate granting of camera and microphone access as required.

4.2.1.5 Connectivity and Bandwidth Management:

Requirement: Ensuring stable internet connectivity and the ability to manage bandwidth effectively for smooth video call transmission.

Expected Behavior: Ensuring stable connections and managing bandwidth for smooth calls.

Test Id.	Inputs required	Environment needs	Output Expected
4.2.1.1	Date = 02/12/2023 Start time = 17:00 Duration = 1	Robust authentication methods and encryption for secure storage.	Efficient user verification with minimal friction.
		Scalable, user-friendly interface with stringent security protocols.	Immediate reporting of suspicions or unauthorized access.
4.2.1.2	Date = 02/12/2023 Start time = 18:00 Duration = 1	Role-based system defining permissions for different user categories.	Strict adherence to assigned permissions, avoiding unauthorized actions.

		Granular control and secure assignment mechanisms for permissions.	Consistent access control based on roles, facilitating efficient access reviews.
4.2.1.3	Date = 02/12/2023 Start time = 19:00 Duration = 1	Detailed resource-specific access definitions for users/groups.	Accurate resource access as per assigned permissions.
		Secure assignment and auditing mechanisms for access control.	Prompt reporting and correction of access discrepancies.
4.2.1.4	Date = 02/12/2023 Start time = 20:00 Duration = 1	Integration of diverse authentication methods for layered security.	Prompt and accurate provision of authentication factors by users.
		User-friendly interface and compliance with security standards.	Improved security resilience against unauthorized access.
4.2.1.5	Date = 02/12/2023 Start time = 21:00 Duration = 1	Detailed logging of user access and system activities.	Identification and recording of irregularities in system logs.
		Real-time monitoring tools and secure data storage.	Prompt reporting and resolution of suspicious activities.

Table 16

4.2.2 Test Case Specification and Design for “Video and voice chat “.

Features to be tested: “Audio quality”

4.2.2.1 Echo Cancellation:

Requirement: Implementation to prevent echoes caused by audio feedback.

Expected Behavior: System actively identifies and eliminates echo feedback during calls, ensuring no audible reverberation or repetition of audio.

4.2.2.2 Low Latency:

Requirement: Ensuring minimal delay in audio transmission for real-time conversation. -

Expected Behavior: Real-time audio transmission with minimal delay, allowing for smooth, natural conversation without noticeable pauses or lags.

4.2.2.3 Consistency Across Devices:

Requirement: Maintaining consistent audio quality across various devices and platforms.

Expected Behavior: Uniform audio quality experienced across various devices (phones, computers, tablets) and platforms, maintaining clarity and reliability.

4.2.2.4 Adaptive Bitrate:

Requirement: Dynamic adjustment of audio quality based on available bandwidth for uninterrupted communication.

Expected Behavior: Dynamic adjustment of audio quality based on available bandwidth, ensuring uninterrupted communication despite varying internet speeds.

4.2.2.5 Support for HD Audio:

Requirement: Capability to handle high-definition audio formats for enhanced clarity. -

Expected Behavior: Capable of handling high-definition audio formats, providing superior clarity and richness in sound quality during communication sessions.

Test Id.	Inputs required	Environment needs	Output Expected
4.2.2.1	Date = 03/12/2023 Start time = 7:00 Duration = 1	Robust authentication methods and encryption for secure storage.	Efficient user verification with minimal friction.
		Scalable, user-friendly interface with stringent security protocols.	Immediate reporting of suspicions or unauthorized access.
4.2.2.2	Date = 03/12/2023 Start time = 8:00 Duration = 1	Role-based system defining permissions for different user categories.	Strict adherence to assigned permissions, avoiding unauthorized actions.
		Granular control and secure assignment mechanisms for permissions.	Consistent access control based on roles, facilitating efficient access reviews.
4.2.2.3	Date = 03/12/2023 Start time = 9:00 Duration = 1	Detailed resource-specific access definitions for users/groups.	Accurate resource access as per assigned permissions.
		Secure assignment and auditing mechanisms for access control.	Prompt reporting and correction of access discrepancies.
4.2.2.4	Date = 03/12/2023 Start time = 10:00 Duration = 1	Integration of diverse authentication methods for layered security.	Prompt and accurate provision of authentication factors by users.
		User-friendly interface and compliance with security standards.	Improved security resilience against unauthorized access.
4.2.2.5	Date = 03/12/2023 Start time = 11:00 Duration = 1	Detailed logging of user access and system activities.	Identification and recording of irregularities in system logs.
		Real-time monitoring tools and secure data storage.	Prompt reporting and resolution of suspicious activities.

Table 17

4.2.3 Test Case Specification and Design for “Video and voice chat “.

Features to be tested: “Call stability”

4.2.3.1 Reliable Connectivity:

Requirement: Implementation to prevent echoes caused by audio feedback.

Expected Behavior: System actively identifies and eliminates echo feedback during calls, ensuring no audible reverberation or repetition of audio.

4.2.3.2 Redundancy Measures:

Requirement: Ensuring minimal delay in audio transmission for real-time conversation. -

Expected Behavior: Real-time audio transmission with minimal delay, allowing for smooth, natural conversation without noticeable pauses or lags.

4.2.3.3 Load Balancing:

Requirement: Maintaining consistent audio quality across various devices and platforms.

Expected Behavior: Uniform audio quality experienced across various devices (phones, computers, tablets) and platforms, maintaining clarity and reliability.

4.2.3.4 Packet Loss Prevention:

Requirement: Dynamic adjustment of audio quality based on available bandwidth for uninterrupted communication.

Expected Behavior: Dynamic adjustment of audio quality based on available bandwidth, ensuring uninterrupted communication despite varying internet speeds.

4.2.3.5 Robust Server Infrastructure:

Requirement: Capability to handle high-definition audio formats for enhanced clarity. -

Expected Behavior: Capable of handling high-definition audio formats, providing superior clarity and richness in sound quality during communication sessions.

Test Id.	Inputs required	Environment needs	Output Expected
4.2.3.1	Date = 03/12/2023 Start time = 12:00 Duration = 1	A microphone or multiple microphones capturing sound.	Clear audio transmission without audible reverberation or repetition, ensuring seamless conversations.
		Speakers or headphones through which sound is played..	Instant identification and elimination of echo feedback, maintaining uninterrupted and smooth audio quality.
4.2.3.2	Date = 03/12/2023 Start time = 13:00 Duration = 1	Role-based system defining permissions for different user categories.	Strict adherence to assigned permissions, avoiding unauthorized actions.
		Granular control and secure assignment mechanisms for permissions.	Consistent access control based on roles, facilitating efficient access reviews.

4.2.3.3	Date = 03/12/2023 Start time = 14:00 Duration = 1	Detailed resource-specific access definitions for users/groups.	Accurate resource access as per assigned permissions.
		Secure assignment and auditing mechanisms for access control.	Prompt reporting and correction of access discrepancies.
4.2.3.4	Date = 03/12/2023 Start time = 15:00 Duration = 1	Integration of diverse authentication methods for layered security.	Prompt and accurate provision of authentication factors by users.
		User-friendly interface and compliance with security standards.	Improved security resilience against unauthorized access.
4.2.3.5	Date = 03/12/2023 Start time = 16:00 Duration = 1	High-performance servers for processing high-definition audio formats.	Superior clarity and richness in sound quality during communication sessions.
		Reliable, high-capacity network infrastructure for seamless transmission of high-definition audio data.	Immersive and engaging audio experience due to enhanced audio capabilities.

4.3 Test Case Design for “Status information”

"Developing a real-time Status information chat system for seamless and immersive communication experiences across various devices and platforms."

4.3.1 Test Case Specification and Design for “Status information “.

Features to be tested: “Automated status changes”

4.3.1.1 Clear Requirements:

Requirement: These could include time-based triggers, user actions, system events, or data thresholds.

Expected Behavior: Changes should occur promptly upon meeting the triggering conditions.

4.3.1.2 Audit Trail:

Requirement: Maintain a log or history of status changes, including the reason for the change and who or what triggered it.

Expected Behavior: Status changes should accurately reflect the current state based on the predefined criteria.

4.3.1.3 Customization:

Requirement: Allow for customization of triggers and statuses based on user needs.

Expected Behavior: The system should consistently apply the rules for status changes without deviation.

4.3.1.4 Communication:

Requirement: Notify relevant stakeholders when a status change occurs.

Expected Behavior: The automated system should be scalable to handle changes in data volume or user load.

4.3.1.5 Permissions and Security:

Requirement: Implement appropriate access controls to ensure that only authorized personnel or systems can initiate status changes.

Expected Behavior: Implement mechanisms to handle errors or exceptions gracefully, such as fallback procedures or notifications for failed changes.

Test Id.	Inputs required	Environment needs	Output Expected
4.3.1.1	Date = 03/12/2023 Start time = 17:00 Duration = 1	Reliable system clock or time-tracking mechanism.	Status changes occur precisely at the designated times, reflected in the system logs or interface.
		User interface elements or APIs to track and initiate status changes based on user actions.	Immediate status change following the user action, with confirmation provided to the user.
4.3.1.2	Date = 03/12/2023 Start time = 18:00 Duration = 1	Role-based system defining permissions for different user categories.	Strict adherence to assigned permissions, avoiding unauthorized actions.
		Granular control and secure assignment mechanisms for permissions.	Consistent access control based on roles, facilitating efficient access reviews.
4.3.1.3	Date = 03/12/2023 Start time = 19:00 Duration = 1	Detailed resource-specific access definitions for users/groups.	Accurate resource access as per assigned permissions.
		Secure assignment and auditing mechanisms for access control.	Prompt reporting and correction of access discrepancies.
4.3.1.4	Date = 03/12/2023 Start time = 20:00 Duration = 1	Integration of diverse authentication methods for layered security.	Prompt and accurate provision of authentication factors by users.
		User-friendly interface and	Improved security resilience against unauthorized access.

		compliance with security standards.	
4.3.1.5	Date = 03/12/2023 Start time = 21:00 Duration = 1	Error handling protocols or libraries within the system's code.	Graceful handling of errors or exceptions without system crashes or data corruption.
		Fallback procedures or alternative pathways if a status change fails.	Fallback procedures initiated automatically if a status change encounters issues.

Table 18

4.3.2 Test Case Specification and Design for “Status information “.

Features to be tested: “Availability option”

4.3.2.1 Flexibility:

Requirement: Implementation to prevent echoes caused by audio feedback.

Expected Behavior: System actively identifies and eliminates echo feedback during calls, ensuring no audible reverberation or repetition of audio.

4.3.2.2 User Control:

Requirement: Ensuring minimal delay in audio transmission for real-time conversation. -

Expected Behavior: Real-time audio transmission with minimal delay, allowing for smooth, natural conversation without noticeable pauses or lags.

4.3.2.3 System Compatibility:

Requirement: Maintaining consistent audio quality across various devices and platforms.

Expected Behavior: Uniform audio quality experienced across various devices (phones, computers, tablets) and platforms, maintaining clarity and reliability.

4.3.2.4 Clear Accessibility:

Requirement: Dynamic adjustment of audio quality based on available bandwidth for uninterrupted communication.

Expected Behavior: Dynamic adjustment of audio quality based on available bandwidth, ensuring uninterrupted communication despite varying internet speeds.

Test Id.	Inputs required	Environment needs	Output Expected
4.3.2.1	Date = 04/12/2023 Start time = 7:00 Duration = 1	A settings or configuration interface within the application..	Accessible menus or sections where users can modify feature availability settings.
		User-specific controls to manage feature availability.	Mechanisms to distinguish and apply individual user preferences regarding feature activation.
4.3.2.2	Date = 04/12/2023 Start time = 8:00 Duration = 1	Ensuring the availability option is consistent across	Synchronized availability settings across mobile, web, and other platforms.

		various devices and platforms.	
		Instant application of availability changes without system downtime.	Immediate implementation of feature availability adjustments upon user action.
4.3.2.3	Date = 04/12/2023 Start time = 9:00 Duration = 1	Clear feedback to users when changing availability settings.	Confirmation messages or indicators notifying users when a feature availability setting is altered.
		Integration with the system's architecture and UI.	Seamless incorporation of the availability option within the existing user interface without causing disruption.
4.3.2.4	Date = 04/12/2023 Start time = 10:00 Duration = 1	Version-specific adaptability to maintain availability option functionality across updates.	Ensuring the availability option remains functional and consistent with different versions or updates of the application.
		Tracking and recording of availability changes for user behavior analysis.	Logging feature availability modifications for analytical purposes or user behavior assessment

Table 19

4.3.3 Test Case Specification and Design for “Status information “.

Features to be tested: “Setting and display”

4.3.3.1 User Accessibility:

Requirement: Options/settings should be easily accessible within the user interface. -

Expected Behavior: Users can quickly locate and access the settings or options without confusion.

4.3.3.2 Intuitive Interface:

Requirement: The interface should be user-friendly and intuitive.

Expected Behavior: Clear and straightforward menus or interfaces that guide users through setting and display configurations.

4.3.3.3 Customization Capability:

Requirement: Users should have the ability to customize settings according to their preferences.

Expected Behavior: The system allows users to set personalized configurations that cater to their needs.

4.3.3.4 Visibility and Clarity:

Requirement :Display options should be clear and visible.

Expected Behavior: The chosen settings or display preferences should be prominently visible in the interface.

4.3.3.5 Consistency Across Devices:

Requirement: Ensuring settings are consistent across different devices or platforms.

Expected Behavior: Settings applied on one device/platform reflect uniformly across others.

Test Id.	Inputs required	Environment needs	Output Expected
4.3.3.1	Date = 04/12/2023 Start time = 11:00 Duration = 1	User testing and feedback to ensure the location of settings is intuitive for most users.	A clearly marked and easily identifiable settings icon or button accessible from any page/screen within the application.
		Implement straightforward navigation paths, such as a clickable settings icon visible on all pages/screens.	Users can access the settings with minimal clicks or interactions, following an intuitive and straightforward path from the main interface.
4.3.3.2	Date = 04/12/2023 Start time = 12:00 Duration = 1	Conduct user research, surveys, or usability tests to understand user preferences and behaviors.	An interface structured with a logical hierarchy, leading users through settings in a step-by-step manner, preventing confusion or disorientation.
		Establish a design system with consistent elements (colors, icons, fonts) to create familiarity for users.	Use of intuitive labels, buttons, and interactive elements that clearly convey their purpose, ensuring users can effortlessly configure settings without ambiguity.
4.3.3.3	Date = 04/12/2023 Start time = 13:00 Duration = 1	Detailed resource-specific access definitions for users/groups.	Accurate resource access as per assigned permissions.
		Secure assignment and auditing mechanisms for access control.	Prompt reporting and correction of access discrepancies.
4.3.3.4	Date = 04/12/2023 Start time = 14:00 Duration = 1	Design elements that place display options prominently, considering visual hierarchy and importance.	Display options stand out visibly from other interface elements.

		Employ contrasting colors, proper typography, and appropriate sizing to make display options stand out.	Display options remain consistently visible across different sections or screens.
4.3.3.5	Date = 04/12/2023 Start time = 15:00 Duration = 1	Implement a robust user account system where settings and preferences are stored and synced across devices for each user.	Settings changes made on one device reflect seamlessly across all synchronized devices.
		Utilize emulators or real-device testing methodologies to verify consistent behavior.	Settings synchronization occurs reliably and without delays.

Table 20

5. Requirements Traceability:

Requirement Traceability Matrix									
Project Name:	Multiple-user Chat System								
Test Case	Test 1	Test 2	Test 3	Test 4	Test 5	Test 6	Test 7	Test 8	Test 9
Requirements									
Registering to the chat system	X	X	X						
Creating an private session in chat	X	X	X						
Real – Time Messaging			X		X				
User Profile Management			X			X	X		
File Sharing				X				X	
Edit and Delete Messages						X			X
Video and Voice Chat				X				X	
Status Information		X			X		X		
Roles and Permission			X			X			

Appendix A: Definitions, Acronyms, and Abbreviations

This section describes any general information that helps to understand this software test document. This section contains an alphabetical listing of all acronyms, abbreviations, and their meanings as used in this document and a list of any terms and definitions needed to understand this document.

Table 21 Definitions, Acronyms, and Abbreviations

Term/Acronym	Definition/Description
Functional Testing	Functional testing is a type of software testing that assesses the system's compliance with specified requirements by evaluating its functions and features.
Performance Testing	The activity of testing the software as a whole to verify that its performance (e.g. response time) matches its SRS.
Requirement	A condition or capability needed by a user to solve a problem or achieve an objective.
Specification	A document that prescribes, in a complete, precise, verifiable manner, the requirements, design, behaviours or other characteristic of a system or system components.
Software Requirements Specification (SRS)	It is a technical oriented software requirements document.
Software Test Document (STD)	A document that describes test plans and test case specifications for testing the software.
System Testing	The testing of the software application as a whole to see the software works according to its specification. This can be further divided into functional testing and performance testing.
User Acceptance Testing (UAT)	User Acceptance Testing (UAT) is the final phase of software testing where end users evaluate the system to ensure it meets their requirements and functions as intended before it is officially deployed.
Multipleuser Chat	A software application that enables multiple users to engage in real-time text-based communication within a shared digital environment.
Load Testing	Assessing the performance and stability of the multiuser chat system under anticipated user loads to identify and address potential bottlenecks.
Concurrency Testing	Testing the multiuser chat system's ability to handle multiple users simultaneously without compromising performance or causing data integrity issues.
Scalability Testing	Evaluating the system's capacity to handle a growing number of users, ensuring that it can scale appropriately as user numbers increase.
Security Testing	Assessing the multiuser chat system for vulnerabilities and ensuring that adequate security measures are in place to protect user data and prevent unauthorized access.
Integration Testing	Verifying the seamless interaction between different components of the multiuser chat system to ensure they function cohesively as a unified application.
Regression Testing	Repeatedly testing the multiuser chat system after changes or updates to ensure that existing functionalities remain unaffected.

Usability Testing	Evaluating the user interface and overall user experience of the multiuser chat system to ensure it is intuitive and user-friendly.
Error Handling	Testing the system's ability to gracefully handle and recover from unexpected errors or input from users.
Collaborative Features	Functionalities that enable users to collaborate effectively within the multiuser chat system, such as file sharing, group chats, and multimedia content sharing.
Message Encryption	Implementing security measures to encrypt messages exchanged within the multiuser chat system, ensuring confidentiality and data integrity.
User Roles and Permissions	Defining and testing different levels of access and permissions for users within the multiuser chat system to maintain data security and control user actions.
Chat History Management	Assessing the system's ability to store and retrieve chat histories accurately, ensuring that users can access past conversations as needed.
Notification Testing	Verifying that users receive timely and accurate notifications for events such as new messages, mentions, or system updates within the multiuser chat system.
Accessibility Testing	Ensuring that the multiuser chat system is accessible to users with disabilities, meeting compliance standards and providing an inclusive user experience.
Real-time Communication Testing	Evaluating the system's capability to facilitate instant and synchronous communication among users in the multiuser chat system.
Scenarios Testing	Creating and testing real-world usage scenarios to assess the multiuser chat system's performance, reliability, and user experience in practical situations.

Appendix B: Contributions

Table 22 Document Contributions

Contributor Name	Sections Worked On
Akshay Karthick M S	Introduction, Test Plan Review Document
Bharathkumar R	User Acceptance Test Plan, Appendix
Dhanush Kumar K	Functional Test Plan
Rohithraj N M	Software Test Plan, UAT, Requirement Traceability