GROUP 4

# Company Communications Systems

---

*Design Document*

Kai Abillar
Johnny Hoang
John Huynh
Bryan Madrigal
Rowwel Ponesto

# Table of Contents

# 1. Document Control

## 1.1. Revision History

| Date | Revision | Description | Author |
|---|---|---|---|
| 10/29/2024 | 1.0 | Document Creation | Kai Abillar |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |

## 1.2. References

- Software Requirements Specification Document v1.9
- Gantt Chart

# 2. Design Overview

## 2.1. Design Goals

1. System Architecture
   - Implement clear separation between communication, user management, chat, and UI components

2. Real-Time Communication

3. Role-Based Access Control
   - Differentiate between regular users and IT administrators

4. Persistent Data Management
   - Maintain chat histories and user information

5. Scalable Design
   - Support multiple concurrent users

## 2.2. Technical Constraints

1. Development Constraints
   - Use only standard Java (no external libraries)
   - No third-party frameworks or tools

2. Storage Constraints
   - All data must be stored in text files

3. Communication Constraints
   - Built-in Java networking capabilities only

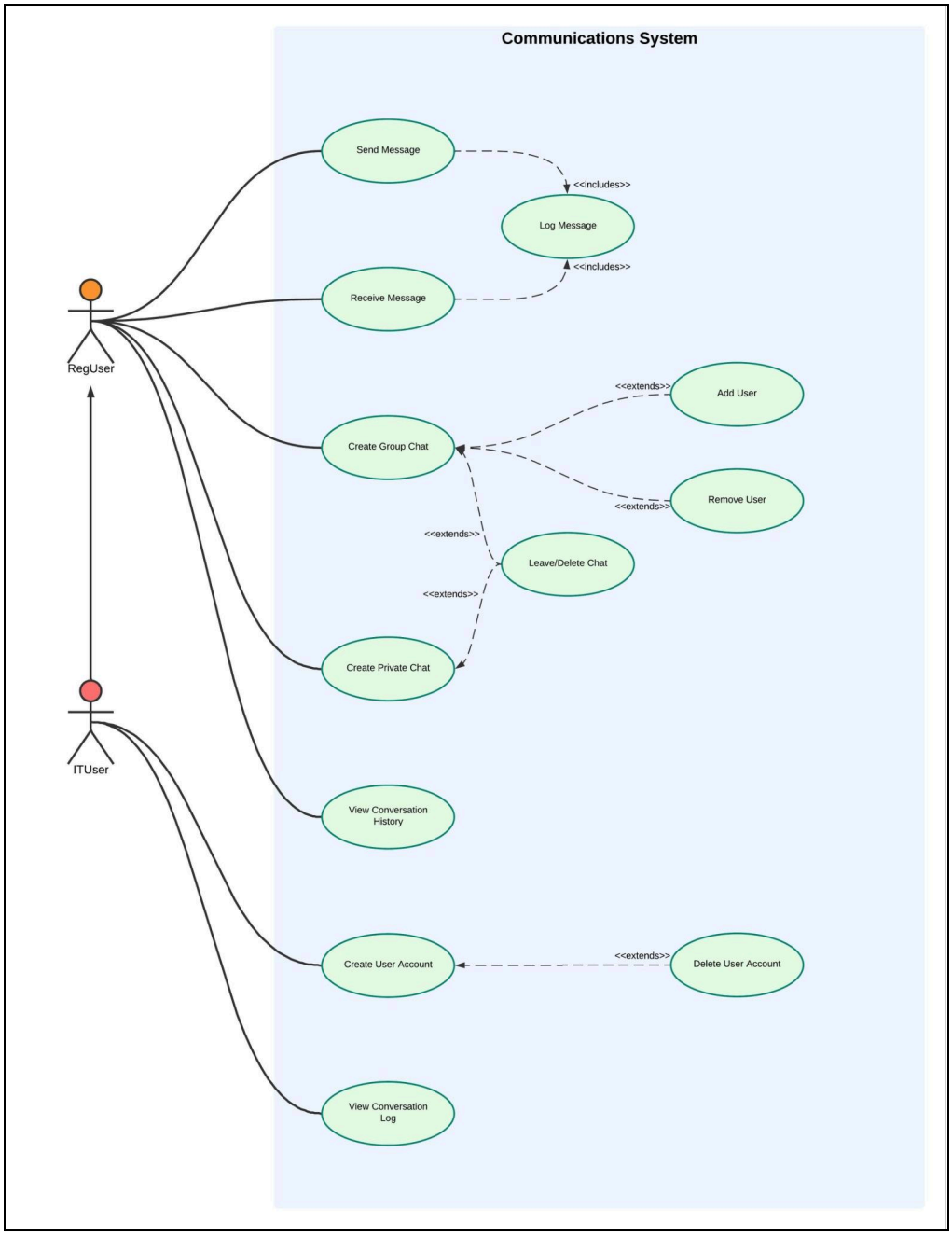
4. Security Constraints
   - Basic authentication using ID and PIN

5. Interface Constraints
   - Simple Java GUI implementation

# 3. Architectural Design

## 3.1. Use Case Analysis

### 3.1.1. Use Case Diagram



### 3.1.2. Use Case Specifications

<table>
<tr><th>USE CASE ID:</th><th>USE CASE NAME:</th><th>PRIMARY ACTORS:</th></tr>
<tr><td>1</td><td>Send Message</td><td>- Regular User<br>- I.T User</td></tr>
<tr><td colspan="3">RELEVANT REQUIREMENTS:<br><br>- Enable private/group chat creation<br>- Support async/sync messaging<br>- Provide real-time notifications<br>- Ensure instant message delivery<br>- Display chat history<br>- Log all messages with metadata (timestamp, sender ID, content)<br>- Store records in files<br>- Minimize message delivery delays</td></tr>
<tr><td colspan="2">PRE-CONDITION:<br><br>- User must be logged in<br>- Recipient(s) must be selected</td><td>POST-CONDITION:<br>- New chat created (if doesn't exist)<br>- Message delivered to receiver(s)<br>- Notification sent to receiver(s)<br>- Message logged in system records</td></tr>
<tr><td colspan="3">BASIC FLOW OR MAIN SCENARIO:<br>1. User chooses to send a private message or send a group message.<br>2. User identifies the receiver(s) of the message.<br>3. User sends a message.<br>4. Server receives the message.<br>5. Message is logged<br>6. Server sends the message to the receiver(s)<br>7. Receiver(s) receive the message<br>8. New chat is accessible for receiver and the sender</td></tr>
<tr><td colspan="3">EXTENSIONS:<br>● Server records: Sender, Receiver(s), TimeStamp, Message content<br>● Server sends delivery receipt<br>● Server sends read receipt when message is read</td></tr>
<tr><td colspan="2">EXCEPTIONS:<br>- Users can send a message to an employee account that no longer exists. The chat history between the active account and deleted account should be preserved</td><td>RELATED:<br>● UC2 - Receive Message<br>● UC4 - Log Message<br>● UC7 - Create Group Chat<br>● UC8 - Create Private Chat<br>● UC6 - View Conversation History</td></tr>
</table>

| USE CASE ID: | USE CASE NAME: | PRIMARY ACTORS: |
|---|---|---|
| 2 | Receive Message | - Regular User<br>- IT User |

**RELEVANT REQUIREMENTS:**

- Enable private and group chat creation
- Allow asynchronous messaging between users.
- Allow synchronous messaging between users.
- Users are notified of new messages in real time.
- Message sending and receiving are instantaneous
- Have a chat history to display previously sent messages
- System will log and archive all messages
- System records message time stamp
- System records message sender's name and id

| PRE-CONDITION: | POST-CONDITION: |
|---|---|
| - Message must be sent from another user | - Receiver receives a notification that a message was received.<br>- If the receiver is in chat then a read receipt must be sent back to the sender. |

**BASIC FLOW OR MAIN SCENARIO:**

1. Message is received by the receiver,
2. If the receiver in the chat the sender receives a read receipt from the sever and If the receiver is not in the chat then they receive a notification.
3. Receiver sees message

**EXTENSIONS:**

- Notification sent if user not in chat

| EXCEPTIONS: | RELATED: |
|---|---|
| - A deleted account should not be able to receive a message from an active employee.<br>- A user no longer in a chat should not receive any messages sent to that chat if they choose to leave the chat. | ● UC1 - Send Message<br>● UC4 - Log Message<br>● UC3 - Leave Chat<br>● UC6 - View Conversation History |

| USE CASE ID: | USE CASE NAME: | PRIMARY ACTORS: |
|---|---|---|
| 3 | Leave Chat | - Regular User<br>- IT User |

**RELEVANT REQUIREMENTS:**

- System will log and archive all messages
- System records message time stamp
- System records message sender's name and id
- System will record when a user joins and leaves the chat with the timestamp.

| PRE-CONDITION: | POST-CONDITION: |
|---|---|
| - User must be logged in<br>- User must be in the chat | - User removed from chat visibility<br>- User cannot send messages to chat<br>- Event logged in system |

**BASIC FLOW OR MAIN SCENARIO:**

1. User selects chat to leave
2. User leaves chat
3. Chat removed from user's view
4. Server removes user from chat
5. Server logs event (user name, date, time)

**EXTENSIONS:**

- There is only one user in the chat remaining. The chat is deleted.

| EXCEPTIONS: | RELATED: |
|---|---|
| - A user who left a chat shall no longer be able to read the chat history nor send messages. | ● UC1 - Send Message<br>● UC2 - Receive Message<br>● UC4 - Log Message |

| USE CASE ID: | USE CASE NAME: | PRIMARY ACTORS: |
|---|---|---|
| 4 | Log Message | - I.T User |

**RELEVANT REQUIREMENTS:**

- I.T can view the chat history of all employees.
- Allow asynchronous messaging between users.
- Allow synchronous messaging between users.
- Have a chat history to display previously sent messages
- System will log and archive all messages
- System records message time stamp
- System records message sender's name and id
- Store record in a file stored in computer

| PRE-CONDITION: | POST-CONDITION: |
|---|---|
| - Message must be sent by a user | - When the message is logged, we record the sender, receiver, timestamp of the message and the content of the message itself.<br>- The message will be stored in the chat history but also permanently stored as a chat log accessible by an I.T User |

**BASIC FLOW OR MAIN SCENARIO:**

1. A user sends a message to receiver(s)
    a. The message goes to the server.
2. Server logs data: timestamp, message content, sender, and receiver
    a. The logged data should then be accessible by an I.T User
3. The message is sent to the receiver(s) and appears in that specific chat history.

**EXTENSIONS:**

- Logged data is accessible by an I.T user

| EXCEPTIONS: | RELATED: |
|---|---|
| - Content of the message might be corrupted or incorrectly formatted. | - UC6 - View Conversation History<br>- UC5 - View Conversation Logs |

| USE CASE ID: | USE CASE NAME: | PRIMARY ACTORS: |
|---|---|---|
| 5 | View Conversation logs | - I.T User |

**RELEVANT REQUIREMENTS:**

- System Should know if user is I.T or Employee (have list of authorized I.T) and correctly gives extra access upon log-in.
- I.T can view the chat history of all employees.
- System will log and archive all messages
- System records message time stamp.
- System records message sender's name and id.
- Store records in a file stored in the computer.

| PRE-CONDITION: | POST-CONDITION: |
|---|---|
| - Users attempting to access conversation logs must be a designated I.T User.<br>- The System will determine whether the user has the credentials to access this data. | - The I.T User will be able to access the data stored in the logs. The data in the logs will be all conversations that occur between accounts. Group or private. |

**BASIC FLOW OR MAIN SCENARIO:**

1. User signs in as an I.T User
2. The System grants I.T User privileges
3. The user is able to access all conversation logs

**EXTENSIONS:**

- The I.T User will be able to delete the chat if it is still active but the conversation logs must not be deleted.

| EXCEPTIONS: | RELATED |
|---|---|
| - Conversation logs should never be deleted and should always be accessible by the I.T User. The I.T User might try to delete a chat that is not active and if this is the case then they must be told the chat does not exist. | - UC6 - View Conversation History |

| USE CASE ID: | USE CASE NAME: | PRIMARY ACTORS: |
|---|---|---|
| 6 | View Conversation History | - I.T User<br>- Regular User |

**RELEVANT REQUIREMENTS:**

- Message sending and receiving are instantaneous.
- Have a chat history to display previously sent messages.
- System will log and archive all messages .
- System records message time stamp.
- System records message sender's name and id.
- Store records in a file stored in the computer.

**PRE-CONDITION:**

- There must be at least one message for there to be a conversation history.
- All messages that were sent within the chat must be viewable to the users within the chat

**POST-CONDITION:**

- Users should be able to have access to old messages sent by themselves and others in a specific chat, whether private or group chat.

**BASIC FLOW OR MAIN SCENARIO:**

1. A user goes into an existing chat; private or group chat/
2. The user can access old messages between them and other members of the chat.

**EXTENSIONS:**

- Chat history is also accessible by the I.T User:
  - I.T User can access chats that are not a part of private or grouped and look at the conversation history of the users in the chat.
- Chat History is different from chat logs: The I.T User will have access to both, the chat history functions as the user facing chat history, where deleted messages can not be seen. In the chat logs however the I.T User has full access to the chat conversation history in addition to the chat logs that would include the deleted messages if any.

**EXCEPTIONS:**

- If a user deletes a chat then they lose access to the chat history. If the same user is added back to the chat then their prior activity and history they had access to is still gone.

**RELATED:**

- UC5 - View Conversation Logs

| USE CASE ID: | USE CASE NAME: | PRIMARY ACTORS: |
|---|---|---|
| 7 | Create Group Chat | - Regular User<br>- I.T User |

**RELEVANT REQUIREMENTS:**

- Enable private and group chat creation
- Allow asynchronous messaging between users.
- Allow synchronous messaging between users.
- Users are notified of new messages in real time.
- Message sending and receiving are instantaneous.
- There can be multiple group chats with the same Users.
- Initial creator of group chat is able to remove users from chat.
- I.T can remove an account from any group chat.

| PRE-CONDITION: | POST-CONDITION: |
|---|---|
| - There can be an existing chat with the same group members.<br>- The Sender must select multiple receivers for the message. | - The message sent by the Sender will be sent to the Receivers they specified.<br>- The message and its data will get logged by the server<br>- A new group chat will have been created |

**BASIC FLOW OR MAIN SCENARIO:**

1. User initiates group chat
2. User creates message
3. User selects multiple recipients
4. Message sent to all recipients
5. Server logs message
6. New chat created for all participants

**EXTENSIONS:**

N/A

| EXCEPTIONS: | RELATED: |
|---|---|
| - If a user deletes the group chat when there are only 3 members in the group chat there might be a conflict between the chat private or a group chat. | ● UC8 - Create Private Chat |

| USE CASE ID: | USE CASE NAME: | PRIMARY ACTORS: |
|---|---|---|
| 8 | Create Private Chat | - Regular User<br>- I.T User |

**RELEVANT REQUIREMENTS:**

- Enable private and group chat creation
- Allow asynchronous messaging between users.
- Allow synchronous messaging between users.
- Users are notified of new messages in real time.
- Message sending and receiving are instantaneous.
- There can only be one chat for a pair of users.

**PRE-CONDITION:**

- There can not be an existing chat with the same person.
- The Sender must select one other user to become the receiver of the message

**POST-CONDITION:**

- The message sent by the Sender will be sent to the Receiver they specified.
- The message and its data will get logged by the server
- A new private chat will have been created

**BASIC FLOW OR MAIN SCENARIO:**

1. User initiates private chat
2. User creates message
3. User selects one recipient
4. Message sent to recipient
5. Server logs message
6. New chat created

**EXTENSIONS:**

- If the User attempts to create a private chat with another user that they already have a private chat with then, the message will be sent within that chat and there will not be a creation of a new private chat. There will only be a creation of a private chat if a private chat does not already exist between both users.

**EXCEPTIONS:**

- New chat not created for deleted user messages
- New chat created after leaving previous chat

**RELATED:**

- UC7 - Create Group Chat

<table>
<tr><th>USE CASE ID:</th><th>USE CASE NAME:</th><th>PRIMARY ACTORS:</th></tr>
<tr><td>9</td><td>Create User Account</td><td>- I.T User</td></tr>
<tr><td colspan="3">RELEVANT REQUIREMENTS:<br>3.1.2.1 Users should be allowed to log in using their issued id and pin, both of which are alphanumeric strings between 6 and 20 characters in length.<br>3.1.2.2 System should know if User is I.T. or Employee (have list of authorized I.T.) and correctly gives extra access upon log-in<br>3.1.2.6 I.T. can create new employee account</td></tr>
<tr><td colspan="2">PRE-CONDITION:<br>- Employee must be new to company<br>- HR must authorize account creation</td><td>POST-CONDITION:<br>- A new user account is created.</td></tr>
<tr><td colspan="3">BASIC FLOW OR MAIN SCENARIO:<br>1. HR requests account creation<br>2. IT accesses admin interface<br>3. IT creates account with valid ID and PIN</td></tr>
<tr><td colspan="3">EXTENSIONS:<br>● Handle duplicate ID cases<br>● Check for existing accounts</td></tr>
<tr><td colspan="2">EXCEPTIONS:<br>- The user creating the account must have Administrator privileges by way of possessing an I.T. account.</td><td>RELATED:<br>● UC10 - Delete User Account</td></tr>
</table>

| USE CASE ID: | USE CASE NAME: | PRIMARY ACTORS: |
|---|---|---|
| 10 | Delete User Account | - I.T User |

**RELEVANT REQUIREMENTS:**

- I.T can delete an employee account.

| PRE-CONDITION: | POST-CONDITION: |
|---|---|
| - An employee account must exist to be deleted | - The employee account is deleted<br>- The employee account is removed from all group chats and private chats.<br>- Any messages sent to a deleted user will not be logged. Therefore we need to throw an error whenever a user attempts to message a deleted user.<br>- No new user can use the unique ID of the deleted user. |

**BASIC FLOW OR MAIN SCENARIO:**

1. I.T user selects an account.
2. I.T user deletes the account.
3. Deleted users are removed from all chats.

**EXTENSIONS:**

- If any user attempts to message a deleted account then a receipt needs to be given to the sender letting them know the account is inactive.

| EXCEPTIONS: | RELATED: |
|---|---|
| - Check for an account's existence before confirming the deletion.<br>- Do not delete an inactive or already deleted account with the same unique ID | - UC9 - Create User Account |

| USE CASE ID: | USE CASE NAME: | PRIMARY ACTORS: |
|---|---|---|
| 11 | Add User | - I.T User<br>- Regular Users |

**RELEVANT REQUIREMENTS:**

1. Group chat members can add more users to the group chat.

| PRE-CONDITION: | POST-CONDITION: |
|---|---|
| - There must be an existing group chat<br>- The user being added must be an existing user.<br>- User added must not already be in the group chat | - Any group chat gets another user increasing group size by 1. |

**BASIC FLOW OR MAIN SCENARIO:**

1. Any user can add a new user to a group chat that they are a member of
    a. An I.T User can add a user to any group chat without needing to be a part of the group chat
2. All members of the group chat are notified of the users addition to the chat.

**EXTENSIONS:**

- If one user is trying to add an invalid user, let them know and throw an exception handler.
- I.T Users should always be notified if a user was added to a group chat

| EXCEPTIONS: | RELATED: |
|---|---|
| N/A | ● UC12 - Remove User from Chat |

<table>
<tr><th>USE CASE ID:</th><th>USE CASE NAME:</th><th>PRIMARY ACTORS:</th></tr>
<tr><td>12</td><td>Remove User</td><td>- I.T User<br>- Regular Users</td></tr>
<tr><td colspan="3">RELEVANT REQUIREMENTS:<br><br>1. Initial creator of group chat is able to remove users from chat.<br>2. I.T can remove an account from any group chat.</td></tr>
<tr><td colspan="2">PRE-CONDITION:<br>- The account that will be removed from a group chat must be in the group chat.</td><td>POST-CONDITION:<br>- The group chat will no longer contain the removed user.<br>- The removed user can not re-join unless they are added by user in the group chat</td></tr>
<tr><td colspan="3">BASIC FLOW OR MAIN SCENARIO:<br>1. User Regular or I.T User goes to a group chat<br>2. They are able to remove any account from the group chat.<br>a. The regular user only has the privilege to remove a user from the group chat if the user is the one who initialized the group chat. (i.e they sent the first message to all accounts in the group chat)<br>3. After removal the accounts in the group chat must be notified of the removal of the account.</td></tr>
<tr><td colspan="3">EXTENSIONS:<br>● Regardless what type of user removes a user, it should be logged so I.T can see If a user account is deleted, then this should be run in all group chats of the existing user, to remove the deleted account from the group chats.</td></tr>
<tr><td colspan="2">EXCEPTIONS:<br>N/A</td><td>RELATED:<br>● UC11 - Add User to Chat</td></tr>
</table>

### 3.1.3. Primary Use Cases

1. Message Management
    - UC1: Send Message
    - UC2: Receive Message

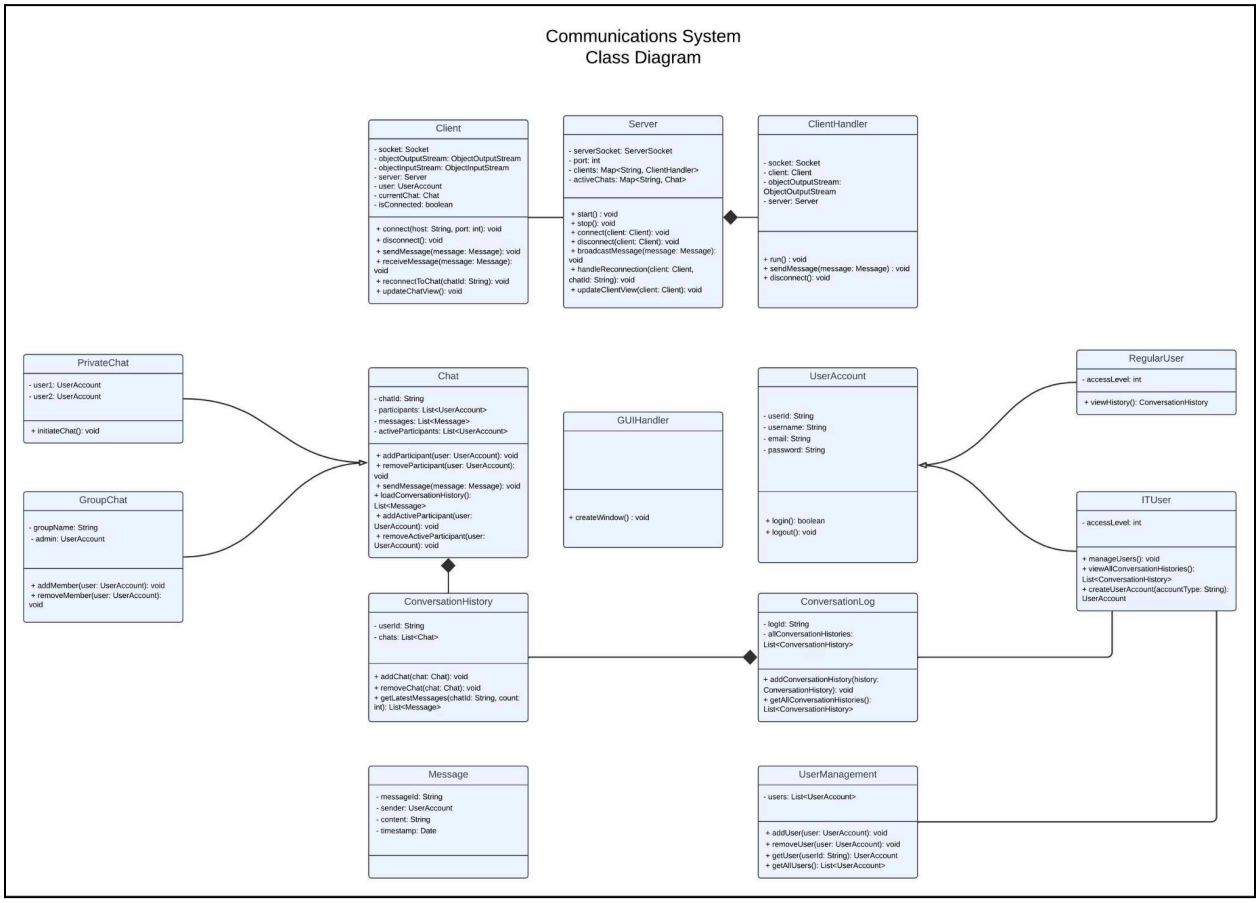
2. Chat Administration
    - UC7: Create Group Chat
    - UC8: Create Private Chat
    - UC3: Leave Chat

3. User Administration (IT Only)
    - UC9: Create User Account
    - UC10: Delete User Account
    - UC11/UC12: Add/Remove Chat Users

4. System Logging and History
    - UC4: Log Message
    - UC5: View Conversation Logs (IT)
    - UC6: View Conversation History

# 3.2. Class Architecture

## 3.2.1. Complete System Class Diagram



### 3.2.2. Module Breakdown

1. Communication Module Classes
    a. Server
    b. Client
    c. ClientHandler

2. User Management Module Classes
    a. UserAccount (Base Class)
    b. RegularUser
    c. ITUser
    d. UserManagement

3. Chat Module Classes
    a. Chat (Base Class)
    b. PrivateChat
    c. GroupChat
    d. Message

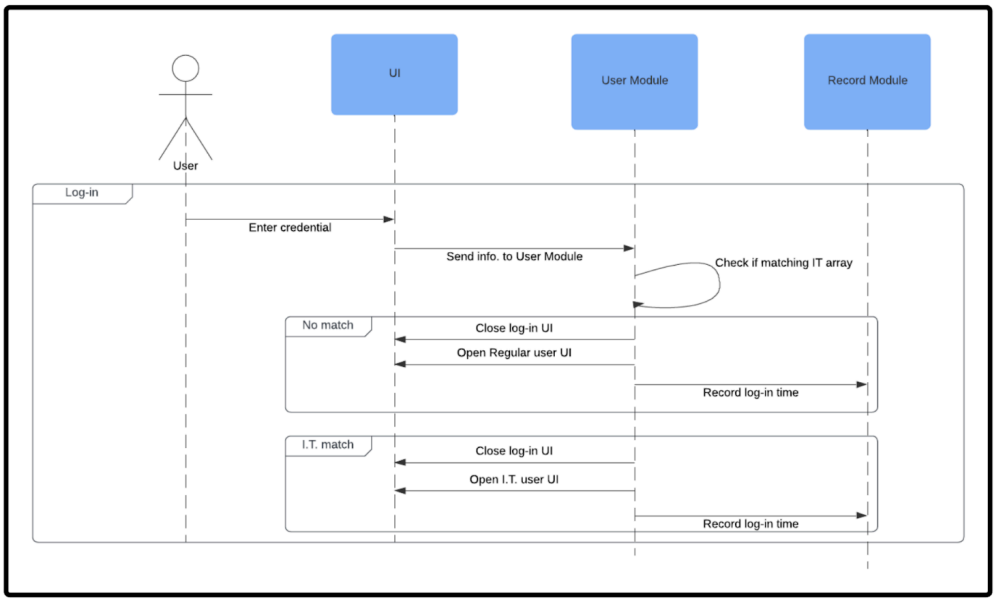
4. Record Module Classes
    a. ConversationHistory
    b. ConversationLog

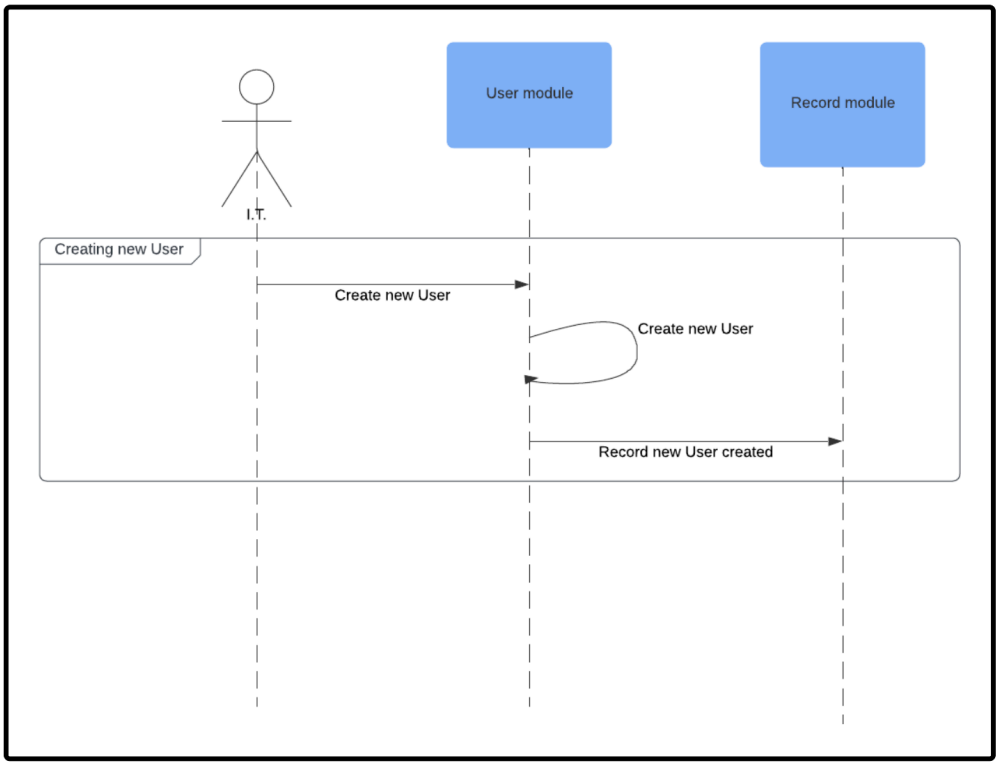
5. UI Module Classes
    a. GUIHandler

## 3.3. Dynamic Behavior

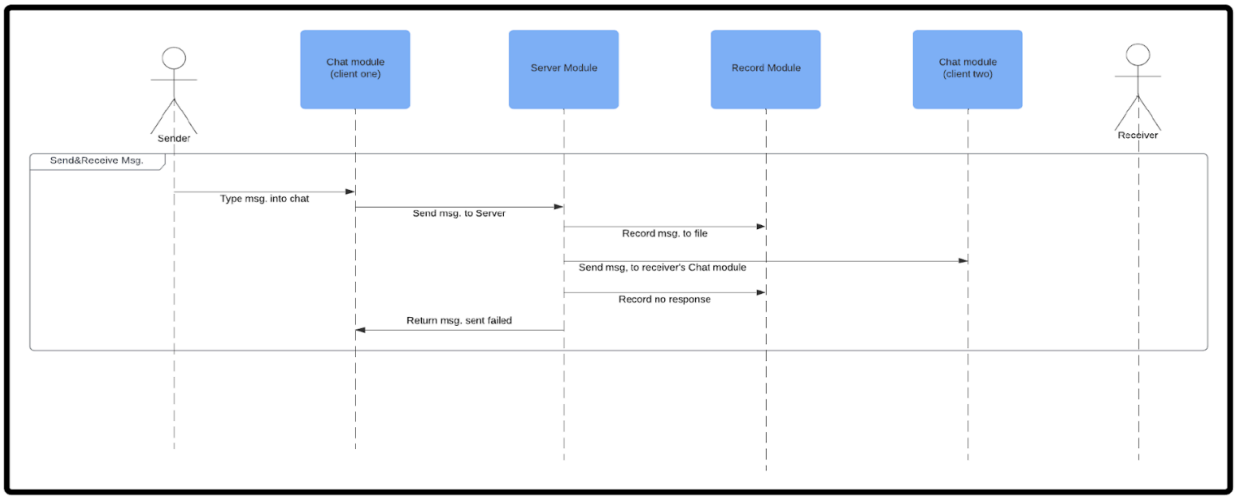
### 3.3.1. Sequence Diagrams

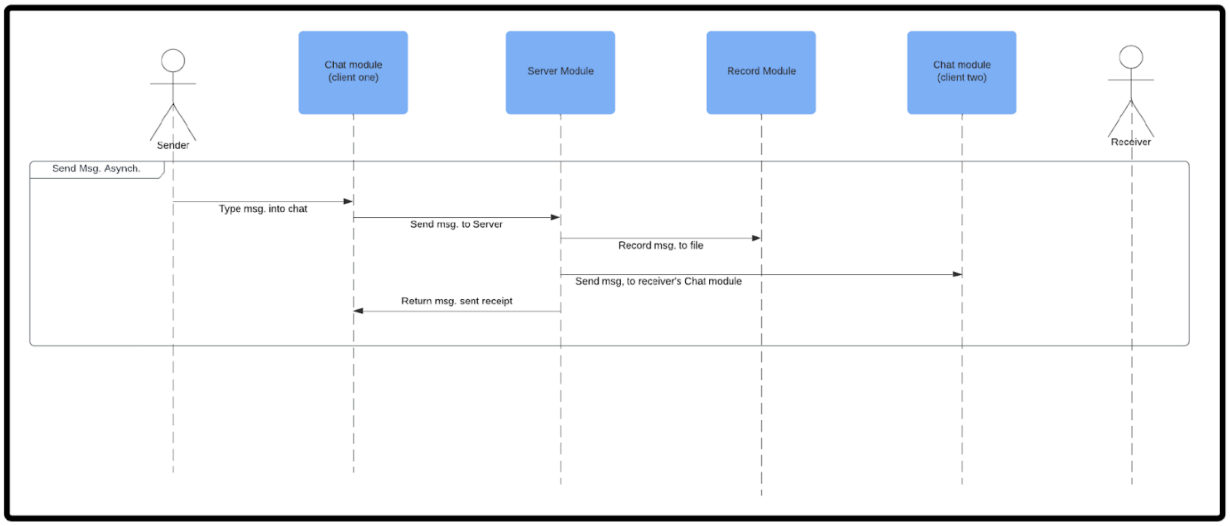
1. User Management

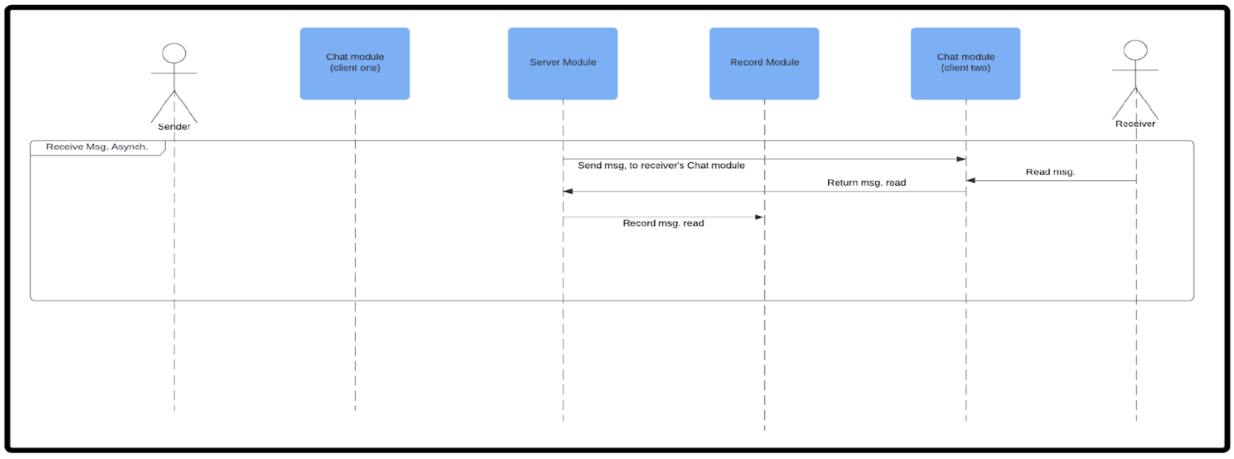


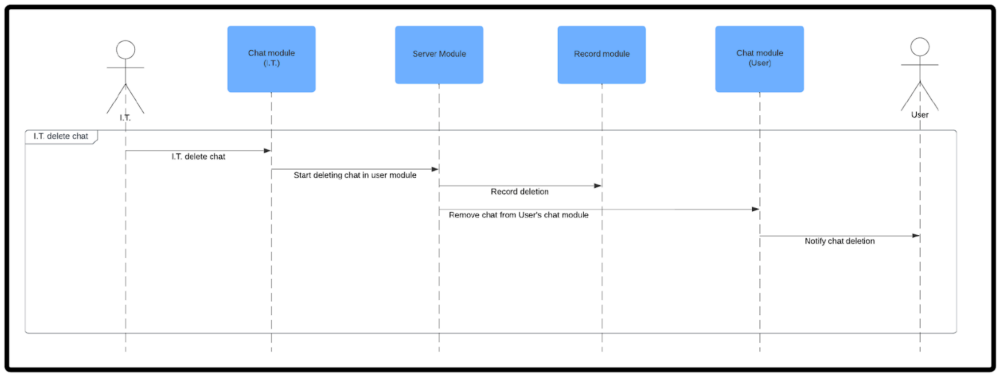
2. Message Operations

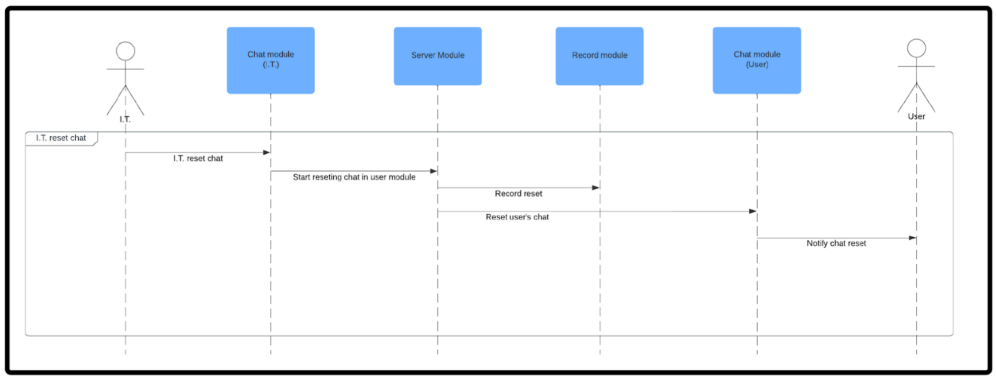


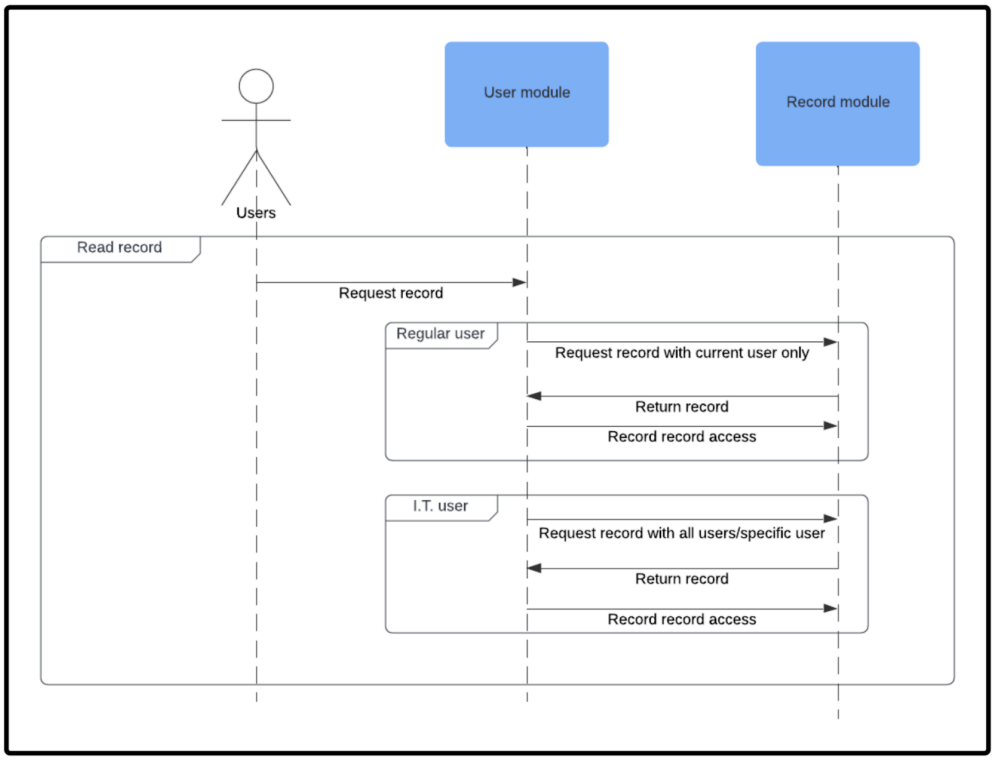



3. History & Record Management



# 4. Interface Specifications

## 4.1. GUI Components

## 4.2. Communication Interfaces

# 5. Data Design

## 5.1. File Structure