1. Use Cases Diagrams

Use Case ID: 1

Use Case Name: User logging in and starting a new chat.

Relevant requirements: 3.1.1.4, 3.1.5.7, 3.1.6.2, 3.1.6.1, 3.2.1, 4.1.3, 3.1.6.7, 3.1.6.26, 3.1.3.2, 3.1.6.5

Primary Actor: User

Pre-conditions: User has an existing username and password. There are other users with accounts.

Post-condition: User has successfully logged in and new chat was created.

Basic Flow or Main Scenario:

1. User enter's username and password

2. System verifies credentials

3. Status is changed to active

4. User selects option to create new chat

5. User inputs individual's username whom they wish to chat with.

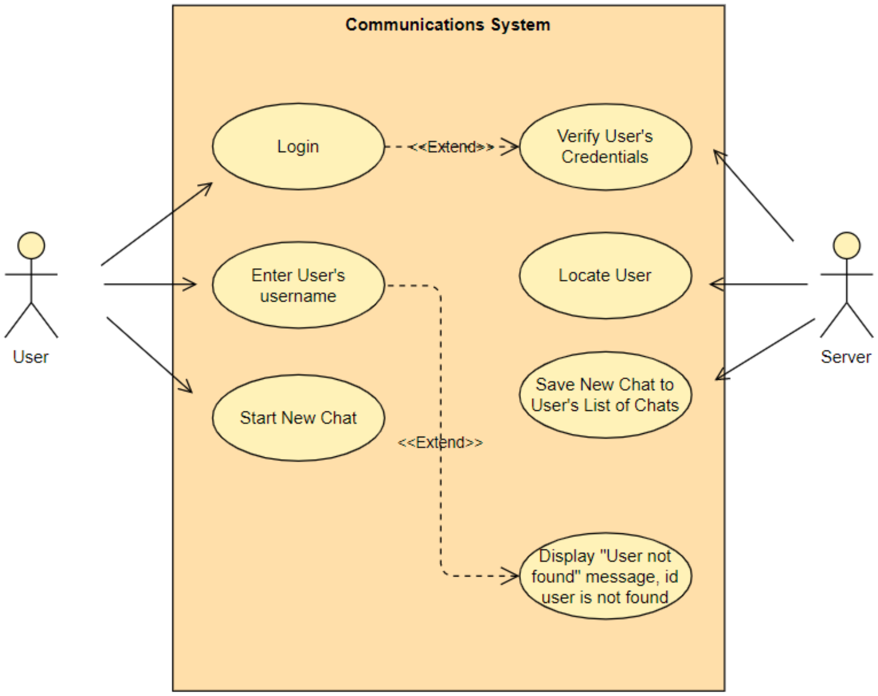
6 . If no chat already exists, a new chat is created.

Extensions or Alternate Flows:

1. User login is unsuccessful.

2. User is prompted to re-enter credentials.

Exceptions: If other user cannot be found, display message stating that the other user does not exist.


Communications System
Login
<<Extend>>
Verify User's Credentials
Enter User's username
Locate User
Save New Chat to User's List of Chats
Start New Chat
<<Extend>>
Display "User not found" message, id user is not found
User
Server

Use Case ID: 2

Use Case Name: User logging in and creating a new group

Relevant requirements: 3.1.1.4, 3.1.5.7, 3.1.6.2, 3.1.6.1, 3.2.1, 4.1.3, 3.1.3.3, 3.1.5.4, 3.1.5.5, 3.1.6.5

Primary Actor: User

Pre-conditions: User has an existing username and password. There are other users with accounts.

Post-condition: User has successfully logged in and created a group with multiple members.

Basic Flow or Main Scenario:

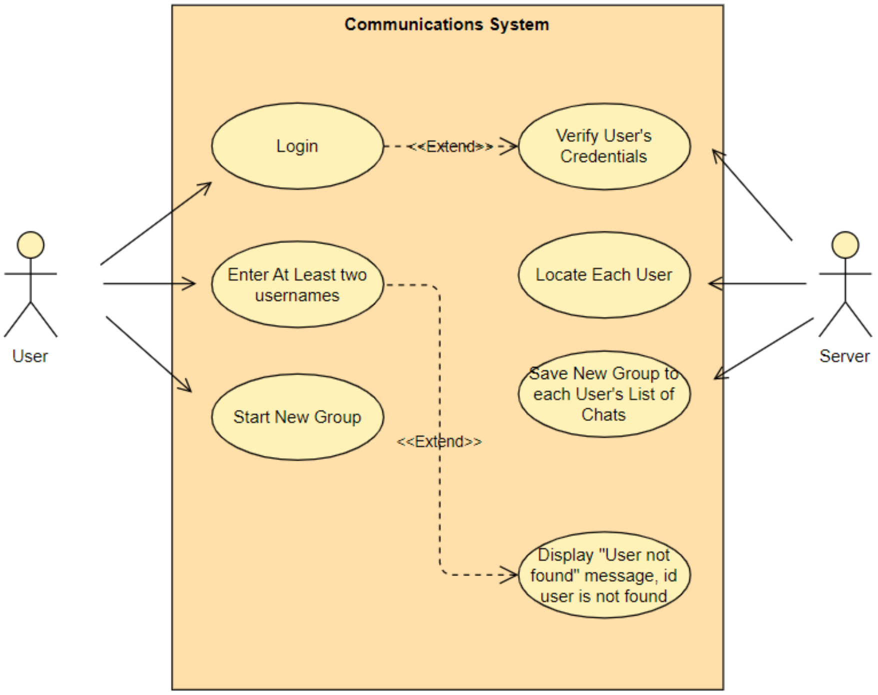
1. User enter's username and password
2. System verifies credentials
3. Status is changed to active
4. User selects option to create new group
5. User inputs other users to be added to group

Extensions or Alternate Flows:

1. User login is unsuccessful.
2. User is prompted to re-enter credentials.

Exceptions: If user cannot be found, display message stating the user does not exist.

Related Use Cases: User logging in and creating a new chat


Communications System
Login
<<Extend>>
Verify User's Credentials
Enter At Least two usernames
Locate Each User
Save New Group to each User's List of Chats
Start New Group
<<Extend>>
Display "User not found" message, id user is not found
User
Server

Use Case ID: 3

Use Case Name: User sending message in chat to inactive user

Relevant requirements: 3.1.1.4, 3.1.5.7, 3.1.6.2, 3.1.6.1, 3.2.1, 4.1.3, 3.1.2.5, 3.1.3.7, 3.1.5.8, 3.1.6.5

Primary Actor: User

Pre-conditions: User has an existing username and password. User that chat will be sent to is inactive.

Post-condition: User has logged in and message is sent and ready to be received when recipient logs in.

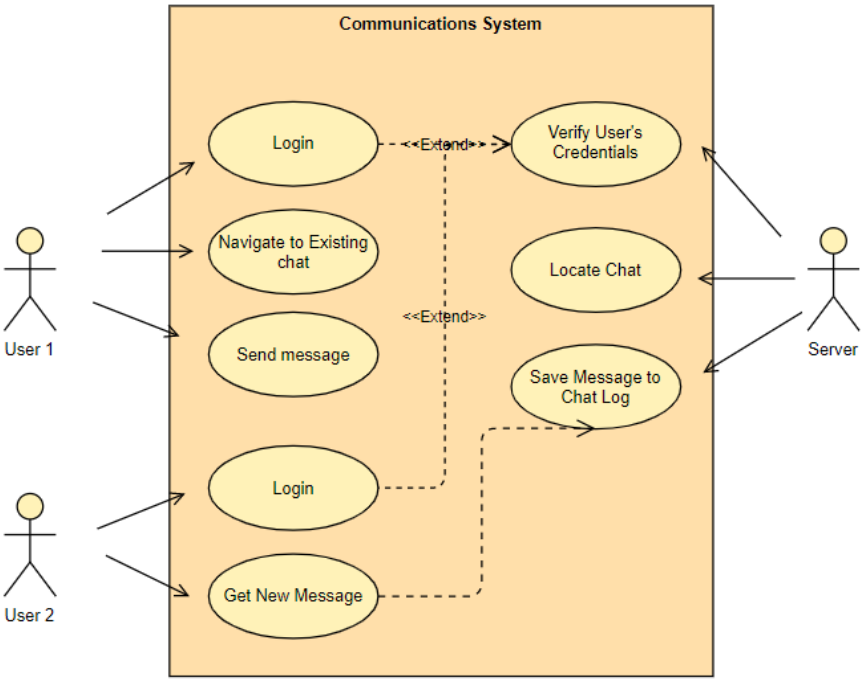
Basic Flow or Main Scenario:

1. User enter's username and password
2. System verifies credentials
3. Status is changed to active
4. User selects an existing chat with an inactive user.
5. User composes message with text only and sends it to inactive user.
6. Message is stored and sent to user who is inactive when they log in.

Extensions or Alternate Flows:

1. User enter's username and password
2. System verifies credentials
3. Status is changed to active
4. User composes message to be sent.
5. User is blocked by recipient.
6. Message is displayed explaining that the user is blocked.

Exceptions: If user cannot be found, display message stating the user does not exist.



Use Case ID: 4

Use Case Name: User sending message to active user.

Relevant requirements: 3.1.1.4, 3.1.5.7, 3.1.6.2, 3.1.6.1, 3.2.1, 4.1.3, 3.1.2.5, 3.1.3.7, 3.1.5.8 , 3.1.6.5

Primary Actor: User

Pre-conditions: User has an existing username and password. User that message will be sent to is active.

Post-condition: User logs in successfully and message is received immediately by recipient.

Basic Flow or Main Scenario:

1. User enter's username and password
2. System verifies credentials

3. Status is changed to active
4. User selects an existing chat with an active user.
5. User composes message with text only and sends it to active user.
6. Message is created and sent to its recipient.

Extensions or Alternate Flows:

1. User enter's username and password
2. System verifies credentials
3. Status is changed to active
4. User composes message to be sent.
5. User is blocked by recipient.
6. Message is displayed explaining that the user is blocked.

Exceptions: If user cannot be found, display message stating the user does not exist..

Related cases: User sending message to inactive user.

Use Case ID: 5

Use Case Name: User updates current status

Relevant requirements: 3.1.1.4, 3.1.5.7, 3.1.6.2, 3.1.6.1, 3.2.1, 4.1.3, 3.1.2.5, 3.1.3.4, 3.1.3.7, 3.1.5.8, 3.1.6.5

Primary Actor: User

Pre-conditions: User has an existing account.

Post-condition: User has successfully logged in and status has been changed.

Basic Flow or Main Scenario:

1. User enters their username and password.
2. Server verifies credentials and login is successful.
3. Status is updated to active once logged in.
4. User selects desired status.
5. New status is displayed only to users that do not have the user blocked.

Extensions or Alternate Flows:

1. User enters username and password
2. Display that credentials are not valid and request new login.

Exceptions: User has flagged a group chat and has been blocked by an admin.

Related Use Cases: Admin blocks a user from messaging in a group; User is flagged for spamming group chats

Use Case ID: 6

Use Case Name: Admin logs in and views chat logs.

Relevant requirements: 3.1.6.1, 3.1.6.5, 3.2.1, 3.1.5.1, 3.1.5.2, 3.1.5.9, 4.1.2

Primary Actor: Admin

Pre-conditions: Admin has an existing account. Some chat logs have been saved.

Post-condition: Admin has file with chat log history.

Basic Flow or Main Scenario:

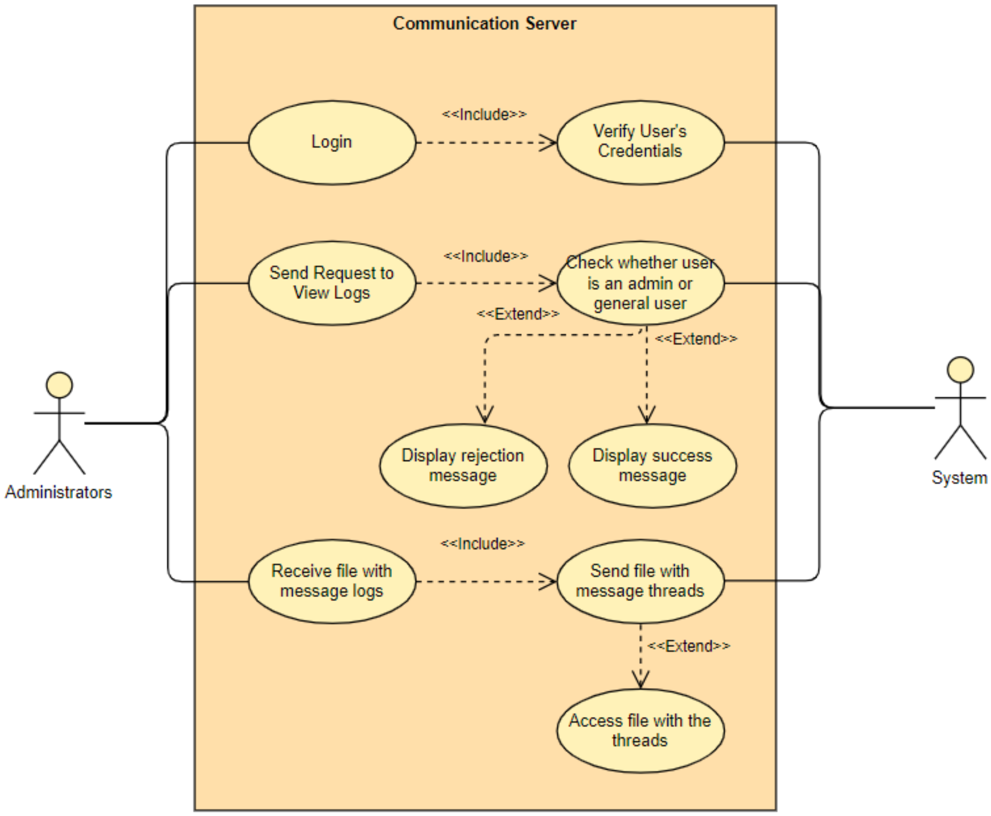
1. Admin enters username and password.
2. System verifies credentials.
3. Admin selects option to view chat logs.
4. System verifies that user is an administrator.
4. System retrieves requested log and returns it to the admin.

Extensions or Alternate Flows:

1. User enters username and password
2. System verifies credentials as user only
3. User is shown error message when trying to view the logs.

Exceptions: No chat logs exist, so the admin does not have anything to access.

Related Use Cases: n/a



Communication Server
Login
<<Include>>
Verify User's Credentials
Send Request to View Logs
<<Include>>
Check whether user is an admin or general user
<<Extend>>
<<Extend>>
Display rejection message
Display success message
Administrators
System
Receive file with message logs
<<Include>>
Send file with message threads
<<Extend>>
Access file with the threads

Use Case ID: 7

Use Case Name: User blocks another user

Relevant requirements: 3.1.1.4, 3.1.5.7, 3.1.6.2, 3.1.6.1, 3.2.1, 4.1.3, 3.1.2.5, 3.1.3.4, 3.1.3.7, 3.1.5.8, 3.1.6.5

Primary Actor: User

Pre-conditions: User has an existing account. Second user has an existing account.

Post-condition: Second user's status has been changed to blocked when viewed by first user.

Basic Flow or Main Scenario:

1. User enters their username and password.
2. Server verifies credentials and login is successful.
3. Status is updated to active once logged in.
4. User navigates to user that they want to block
5. User blocks individual and system changes their status to blocked.

Extensions or Alternate Flows:

n/a

Exceptions: If user is already blocked, display a message stating that they are.

Related Use Cases: n/a

User Case ID: 8

Use Case Name: User is flagged for spamming group chats

Relevant requirements: 3.1.1.4, 3.1.5.7, 3.1.6.2, 3.1.6.1, 3.2.1, 4.1.3

Primary Actor: System

Pre-conditions: User has an existing account and is logged in.

Post-condition: User status is changed to blocked and they can no longer message the chat.

Basic Flow or Main Scenario:

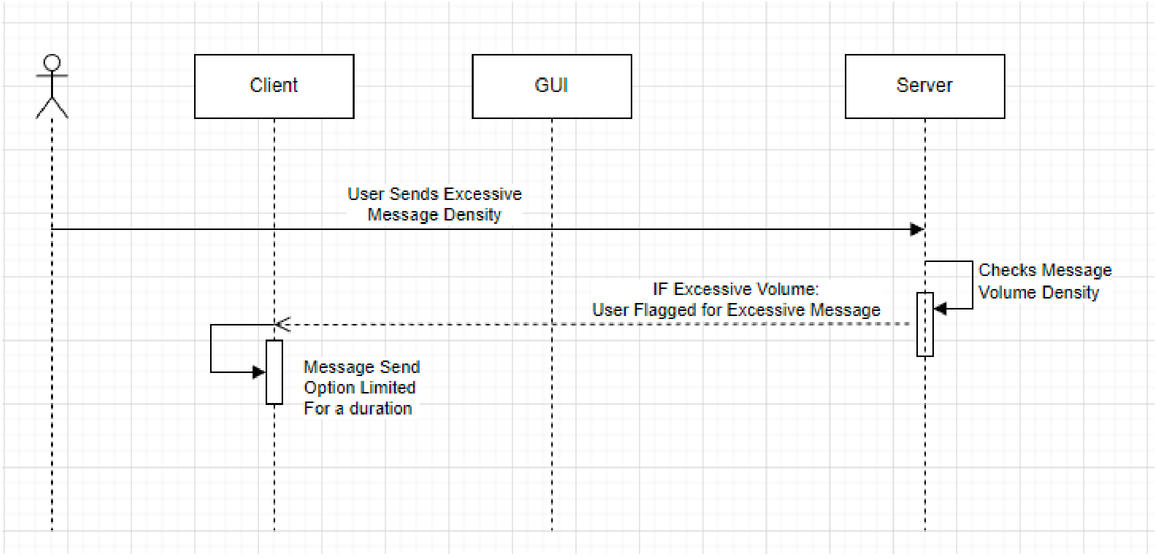
1. User enters their username and password.
2. Server verifies credentials and login is successful.
3. Status is updated to active once logged in.
4. User messages chat beyond the reasonable limit of messages per minute.
5. System flags user’s account
6. System updates user status to blocked and displays message.

Extensions or Alternate Flows:

1. User enters their username and password.
2. Server verifies credentials and login is successful.
3. Status is updated to active once logged in.
4. System flags user for spamming chat
5. Admin changes user status to blocked

Exceptions: If user status changes to away before being blocked, user’s status will be changed to blocked when changing it back to active

Related Use Cases: n/a


Client
GUI
Server
User Sends Excessive Message Density
Checks Message Volume Density
IF Excessive Volume: User Flagged for Excessive Message
Message Send Option Limited For a duration

User Case ID: 9

Use Case Name: User changes their password.

Relevant requirements: 3.1.1.4, 3.1.5.7, 3.1.6.2, 3.1.6.1, 3.2.1, 4.1.3

Primary Actor: User

Pre-conditions: User has an existing account.

Post-condition: User has a new password

Basic Flow or Main Scenario:

1. User enters their username and password.
2. Server verifies credentials and login is successful.
3. Status is updated to active once logged in.
4. User selects option to change password
5. User enters new password
6. System checks length and updates password.

Extensions or Alternate Flows:

1. User enters their username and password.
2. Server verifies credentials and login is successful.
3. Status is updated to active once logged in.
4. User selects option to change password
5. User enters new password
6. System finds that password exceeds the maximum password length
7. System requests new passwords.

Exceptions: If user enters existing password when trying to change password, password will not change.

Related Use Cases: n/a