

Software Design Specification

Group 7

Github Link:

<https://github.com/yoshiyahoo/Communication-App>

Revision History

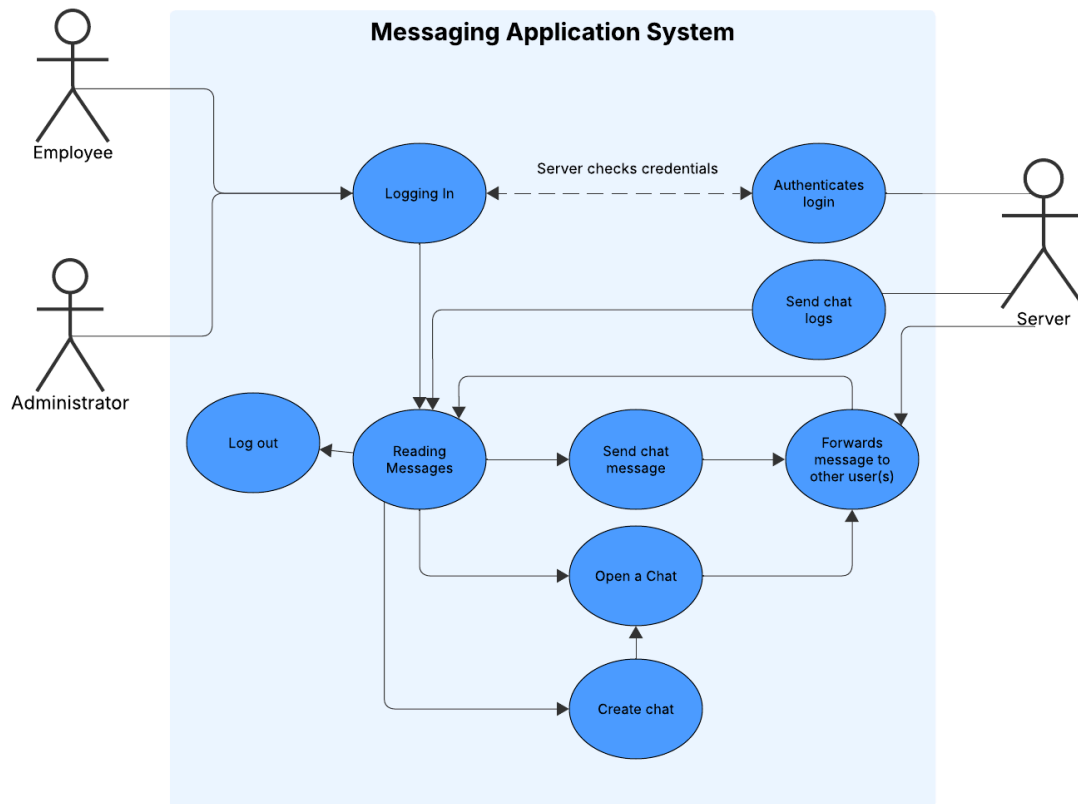
Date	Revision	Description	Author
3/13/2025	1.0	Initial Version	Josiah
3/18/2025	1.1	Imported unformatted use cases	Jacob
3/18/2025	1.2	Made a list of new classes/attributes/methods to put in the new class diagram	Adam
3/20/2025	1.3	Added Figma Design	Kamari
3/20/2025	1.3.1	Formatted Use Cases, Table of Contents, and Revision History	Josiah
3/24/2025	1.4	Made and added Class Diagram V2	Adam
4/04/2025	1.5	Made and added V3 Class Diagrams	Adam
4/05/2025	1.6	Added and edited some use cases	Anna
4/08/2025	1.6.1	Further edited Use Cases	Josiah
4/08/2025	1.7	Add an overview for V4 of class diagrams	Adam
4/08/2025	1.8	Imported use case diagram	Anna
4/08/2025	1.9	Added V5 class diagrams	Adam
4/09/2025	1.9.1	Added Use Case 3 and Slightly Changed GUI Concept	Kamari
4/09/2025	1.9.2	Added Sequence Diagram for Use Case 5	Adam
4/10/2025	1.9.3	Added Sequence Diagram for Use Case 1 and formatted document.	Josiah
4/10/2025	1.9.4	Added Sequence Diagram for Use Case 2 and updated Use Case Diagram	Anna
4/10/2025	1.9.5	Formatted the Class Diagrams Section	Adam
4/10/2025	1.9.6	Added Gantt Chart	Josiah
4/10/2025	1.9.7	Added Github Link	Josiah
5/2/2025	1.9.8	Added additional use case	Anna

5/6/25	1.9.9	Edited use cases and updated some sequence diagrams	Anna
--------	-------	---	------

Table Of Contents

Revision History	1
Table Of Contents	3
UML Case Diagram	4
Use Case Specification	5
Use Case 1: User Logs In	5
Use Case 2: User Sends a Message	6
Use Case 3: User Opens a Chat With Other Users	7
Use Case 4: User Creates Chat With Other Users	8
Use Case 5: User Logs Out	9
Use Case 6: User Search For Adding a Chat	10
UML Class Diagrams	11
Overview:	12
Core Client-Server-Database:	13
Client-Server:	14
Database:	15
Client:	16
UML Sequence Diagrams	17
Use Case 1: User Logs In	18
Use Case 2: Sending a Message	19
Use Case 3: Opening New Chat	20
Use Case 4: User Creates Chat with Other Users	21
Use Case 5: User Logs Out	22
Use Case 6: User Search For Adding a Chat	23
GUI Concept Diagram	24
Gantt Chart	25

UML Case Diagram



Use Case Specification

Use Case 1: User Logs In

Relevant Requirements:

3.1.2.3 - 3.1.2.5, 3.1.3.1 - 3.1.3.2, 3.2.1, 3.3.2, 4.1, 4.2.1, 4.3.3 - 4.3.4

Primary Actor: User

PreCondition: User is not logged in; Application is running

PostCondition: User is logged in and can start seeing chats; chat list and user list are loaded

Basic Flow of Main Scenario:

1. Application shows the login dialog
2. User enters username
3. User enters password
4. User clicks login button
5. Client sends login request to server
6. Server verifies credentials against the database
7. Server finds valid login information
8. Server returns a success response, the user's Account object, the list of allowed Chats, and the full user list
9. Client displays the main chat interface with the populated chat list and user pane

Alternative Flow: User Fails to Log In

1. User enters name
2. User enters password
3. Client sends login request to server
4. Server checks database for login information
5. Server can't find matching login information
6. Server sends login invalid confirmation to client
7. Client informs user login unsuccessful
8. Client can try to login again

Exceptions:

- Login Credentials Invalid
- Request timed out

Related Use Cases:

2, 3, 4, 5

Use Case 2: User Sends & Receives Messages

Relevant Requirements:

3.1.1.2, 3.1.2.2, 3.1.3.1, 3.1.3.3. - 3.1.3.6, 3.1.4.2 - 3.1.4.3, 3.2.2, 3.3.1, 3.3.4, 4.1.4, 4.2.1, 4.3.1. - 4.3.2.

Primary Actor: User

PreCondition: User is logged in; a chat is selected and displayed

PostCondition: Message is delivered to participants; conversation updates in real time

Basic Flow of Main Scenario:

1. User types text in message field
2. User clicks send button (or presses Enter)
3. Client enqueues the Message for sending
4. OutgoingHandler thread pulls it and writes it to the server
5. Server saves the message, then pushes it into a broadcast queue
6. RqstHandler thread takes it, and for each online participant sends it
7. Each client's IncomingHandler receives it, enqueues it, and the GUI appends it to the conversation

Alternate Flow: Network Failure

1. User types in message
2. User clicks send button
3. Client fails to connect to server
4. App tries to pass the message to server
5. Message never appears in the chat window
6. User can try to resend message once connection is restored to recover

Exceptions:

- Server takes too long to process message
- Empty message
- Network failure on send
- Server error during broadcast

Related Use Cases:

1, 3, 4, 5

Use Case 3: User Views and Selects a Chat

Relevant Requirements:

3.1.2.1, 3.1.3.1, 3.1.3.3, 3.1.3.6, 3.1.4.2 - 3.1.4.3, 3.2.2, 3.3.4,

Primary Actor: User

PreCondition: User is logged in; chat list is populated

PostCondition: Selected chat's history and participants are displayed

Basic Flow of Main Scenario:

1. Client populates the left-hand chat list (admins see all chats; employees see only theirs)
2. Users clicks on a chat name
3. Client loads that chat's message history and participant list
4. Chat title, history, and users appear in the center and right panes

Exceptions:

- Chat list is empty
- Error fetching history or participants

Related Use Cases:

1, 2, 4, 5

Use Case 4: User Creates Chat With Other Users

Relevant Requirements: 3.1.1, 3.1.2.5, 3.1.2.2, 3.1.2.1, 3.1.3.5, 3.1.3.4, 3.1.3.3, 3.1.3.1, 3.2.2.1, 3.3.4, 3.3.3, 3.3.1.

Primary Actor: User

PreCondition: User is logged in; chat list is displayed

PostCondition: User has created group chat; new chat exists on server

Basic Flow of Main Scenario:

1. User Clicks the "+" button next to the chat list
2. New chat dialog appears, showing fields for chat name and user search
3. User types a name
4. As user types into "Add User", matching usernames filter in real time
5. User selects one or more participants from the results
6. User clicks "Create"
7. If the name field is blank, nothing happens (dialog stays open)
8. Otherwise, client sends the new-chat request to the server
9. Dialog closes
10. Client adds the chat name into the left-hand list but only if it wasn't already there

Alternate Flow- User Cancels:

1. Instead of Create, the user closes the dialog window
2. The dialog simply disappears
3. No request is sent and the chat list stays the same

Alternate Flow- Blank Chat Name:

1. In the New Chat dialog, user leaves the "Chat Name" field empty
2. Clicks Create
3. Client sees the blank and immediately returns, leaving the dialog open
4. No request is sent, and the chat list remains unchanged

Alternate Flow- Duplicate Chat Name:

1. User enters a chat name that already exists in the left-hand list
2. Clicks Create and the client still sends the request to the server
3. When updating the UI, it detects the name is already present and does not add a second entry
4. The dialog closes, and the chat list remains with a single entry for that name

Exceptions:

- Server times out

Related Use Cases:

1, 2, 3, 5, 6

Use Case 5: User Logs Out

Relevant Requirements:

3.1.2.6, 3.1.3.1, 3.2.2.2, 4.1.2, 4.1.4, 4.2.1, 4.3.4

Primary Actor: User

PreCondition: User is logged in

PostCondition: User is logged out; returned to login screen

Basic Flow of Main Scenario:

1. User presses logout button
2. GUI stops its I/O threads & closes the socket
3. Server sees the socket close (EOF) and removes that client automatically
4. GUI returns to the login screen

Exceptions:

- Logout request fails to send
- Server times out

Related Use Cases:

1, 2, 3, 4

Use Case 6: User Search for Adding Chat

Relevant Requirements:
3.2.2.4

Primary Actor: User

PreCondition: User is logged in and the new chat dialog is opened

PostCondition: A lists of users matching the search is shown

Basic Flow of Main Scenario:

1. User types a partial name
2. Client calls searchUserList(partialName)
3. Returns matching usernames to help start or join chats

Exceptions:

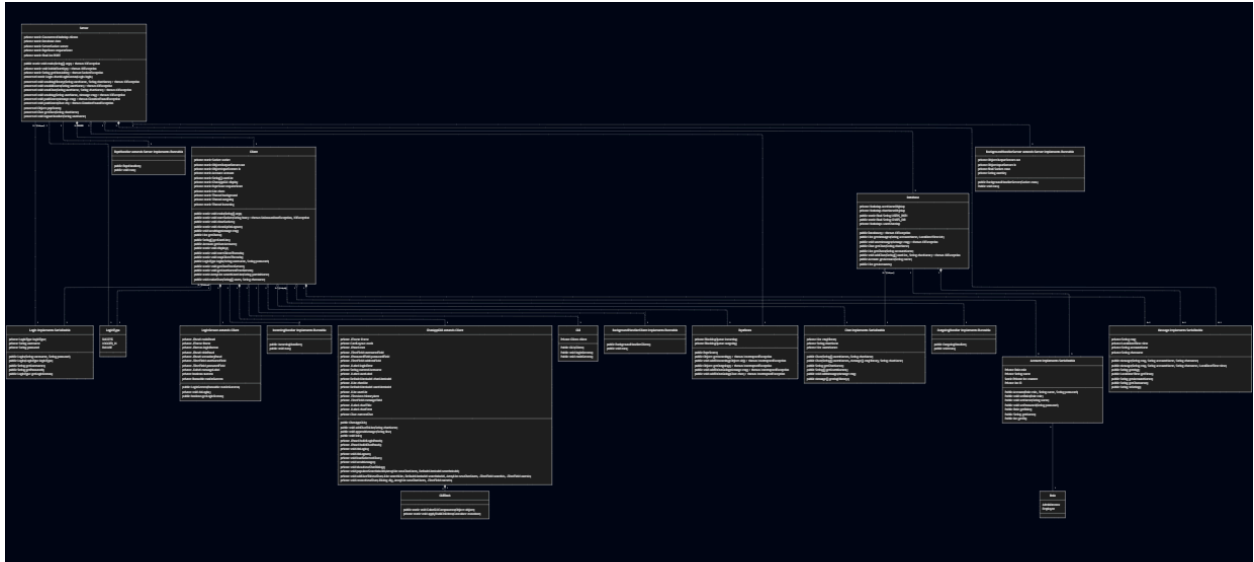
- Wrong input
- No matches found

Related Use Cases:

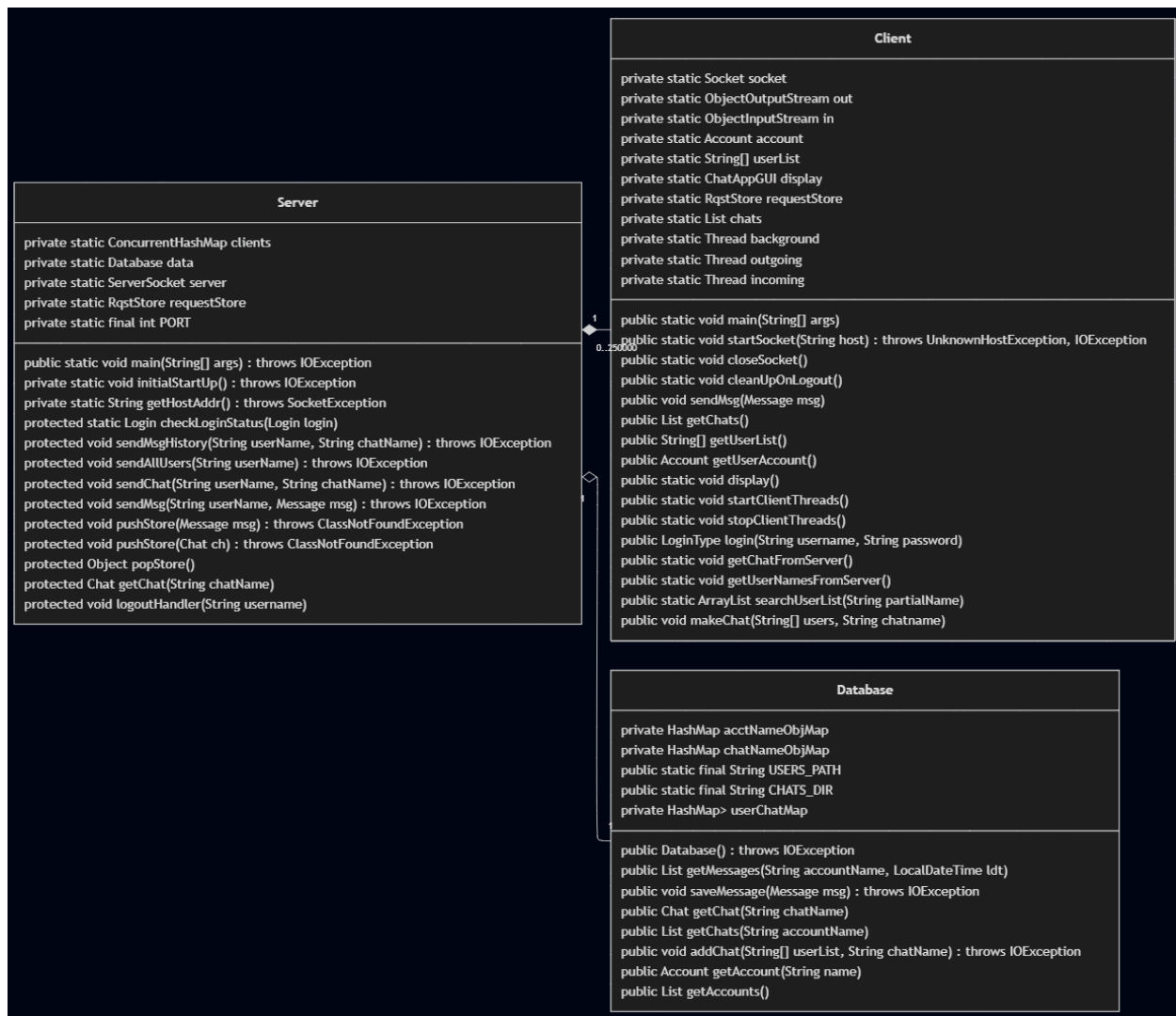
1, 3, 4

UML Class Diagrams

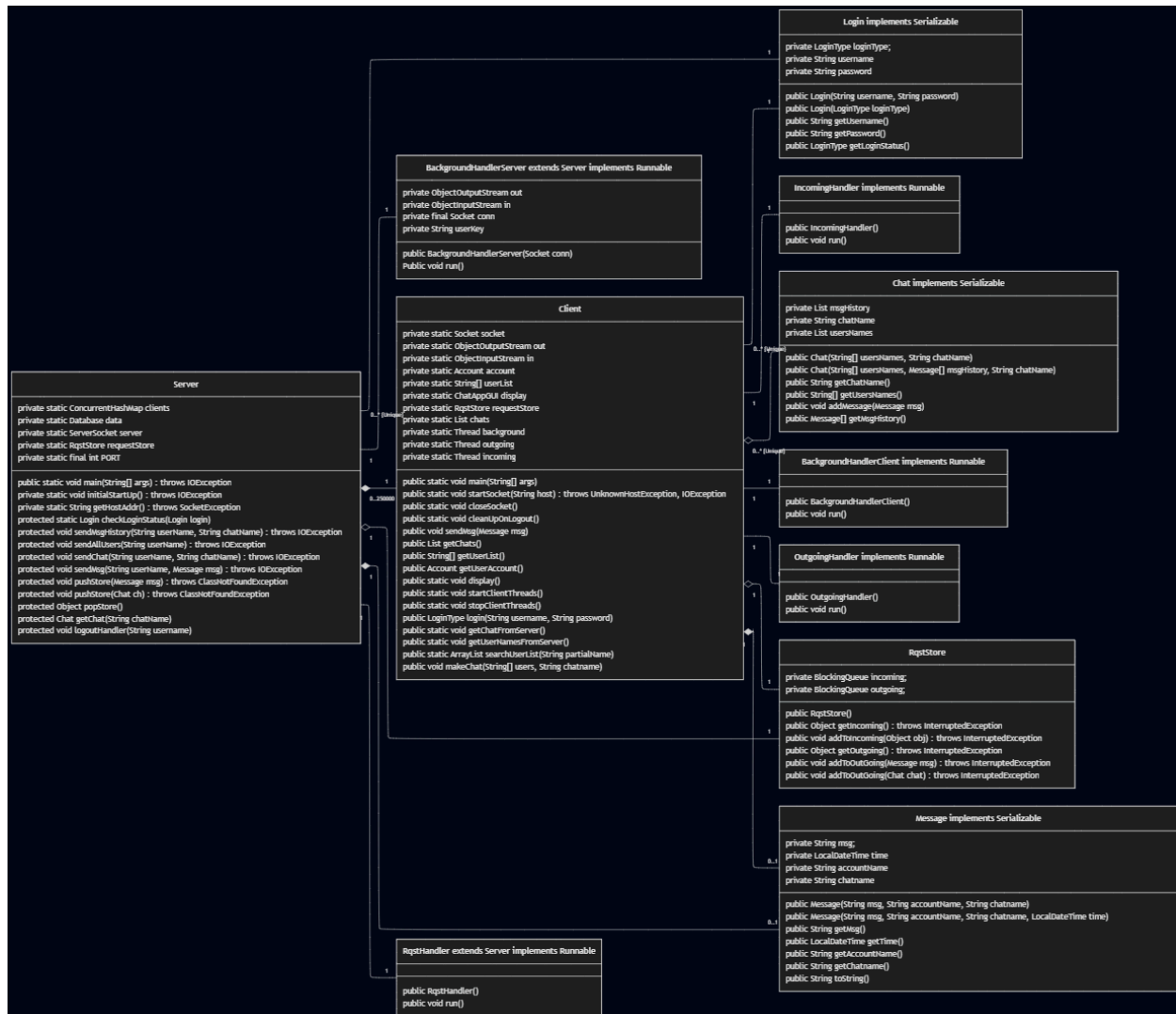
Overview:



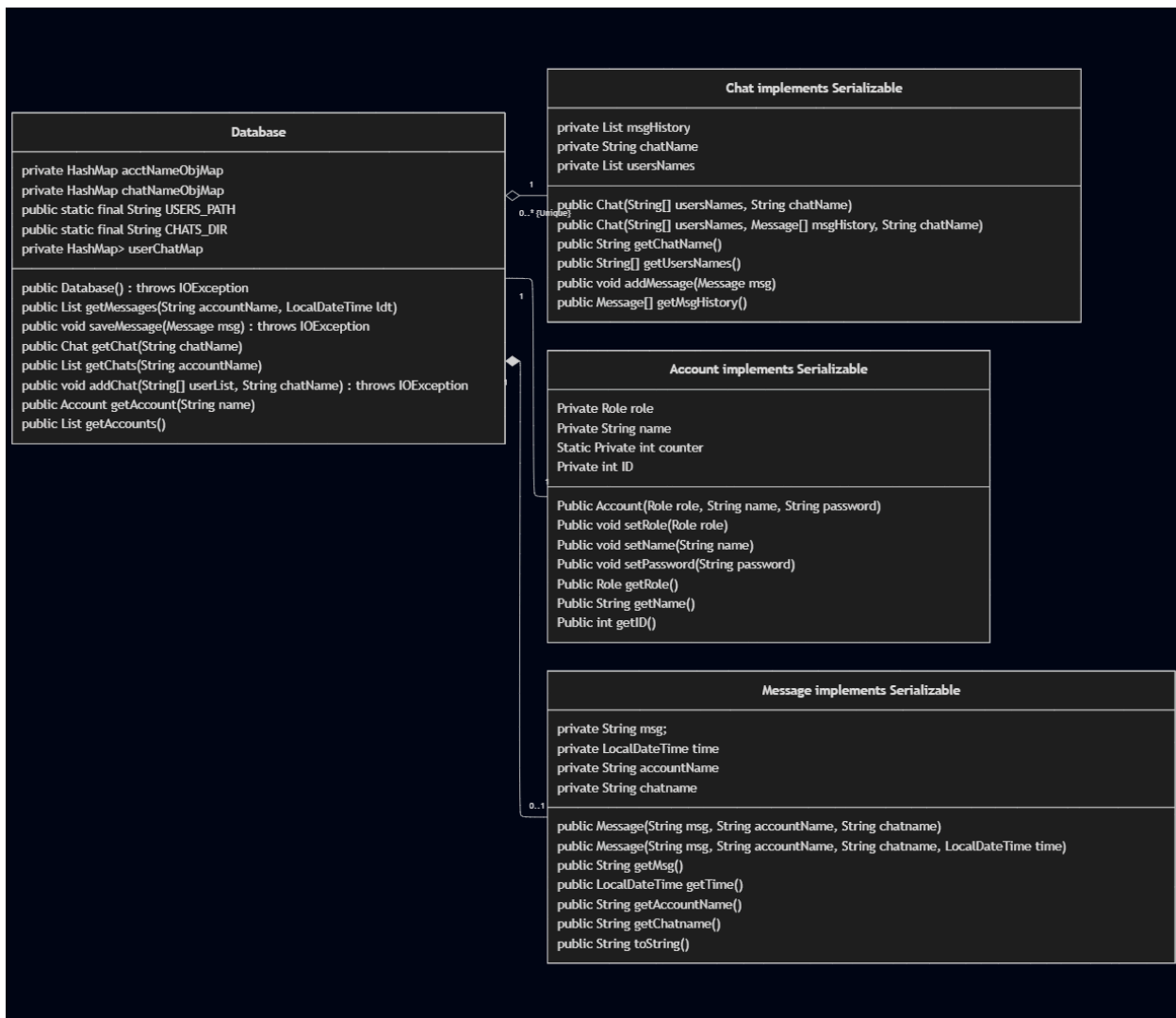
Core Client-Server-Database:



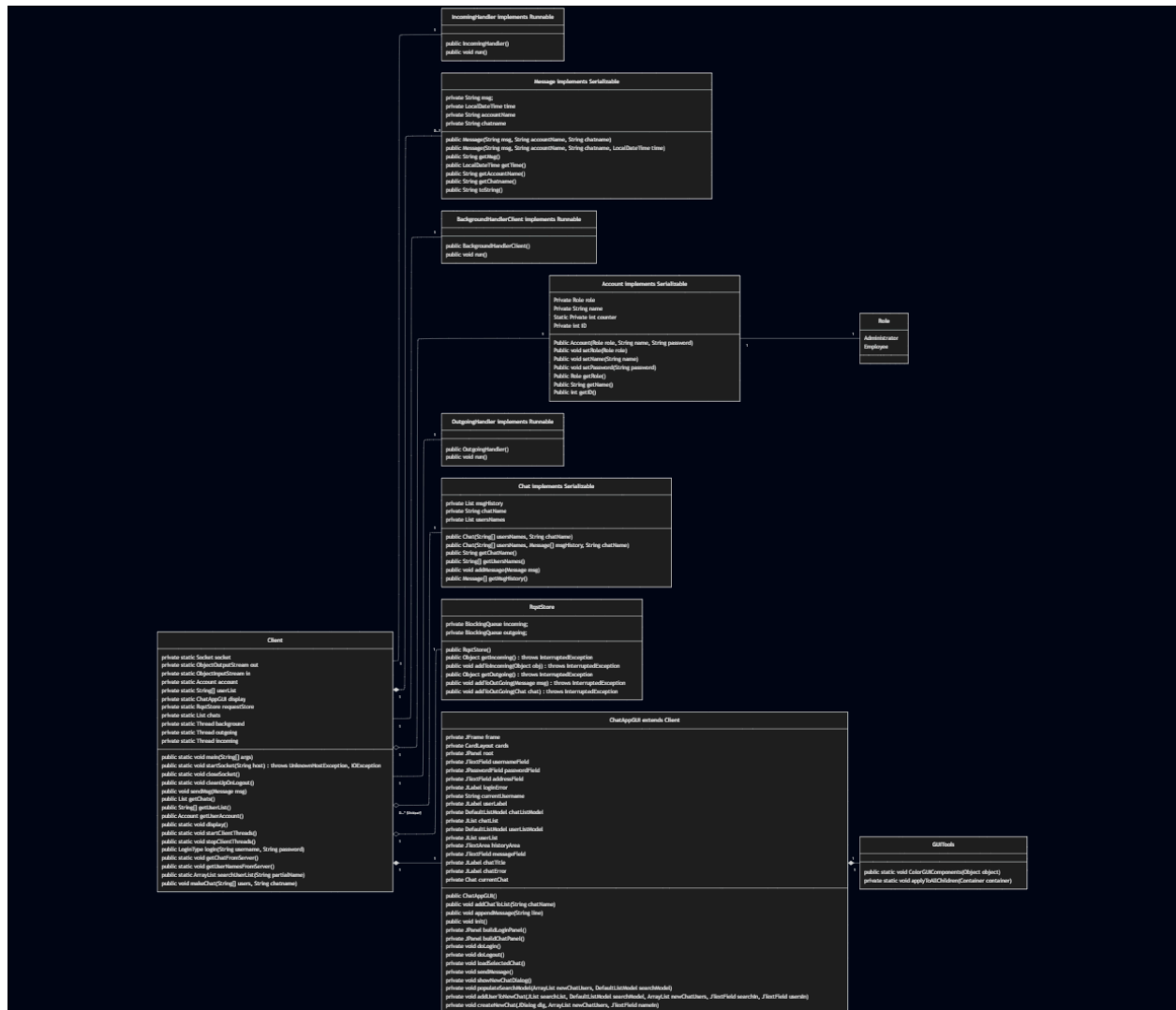
Client-Server:



Database:

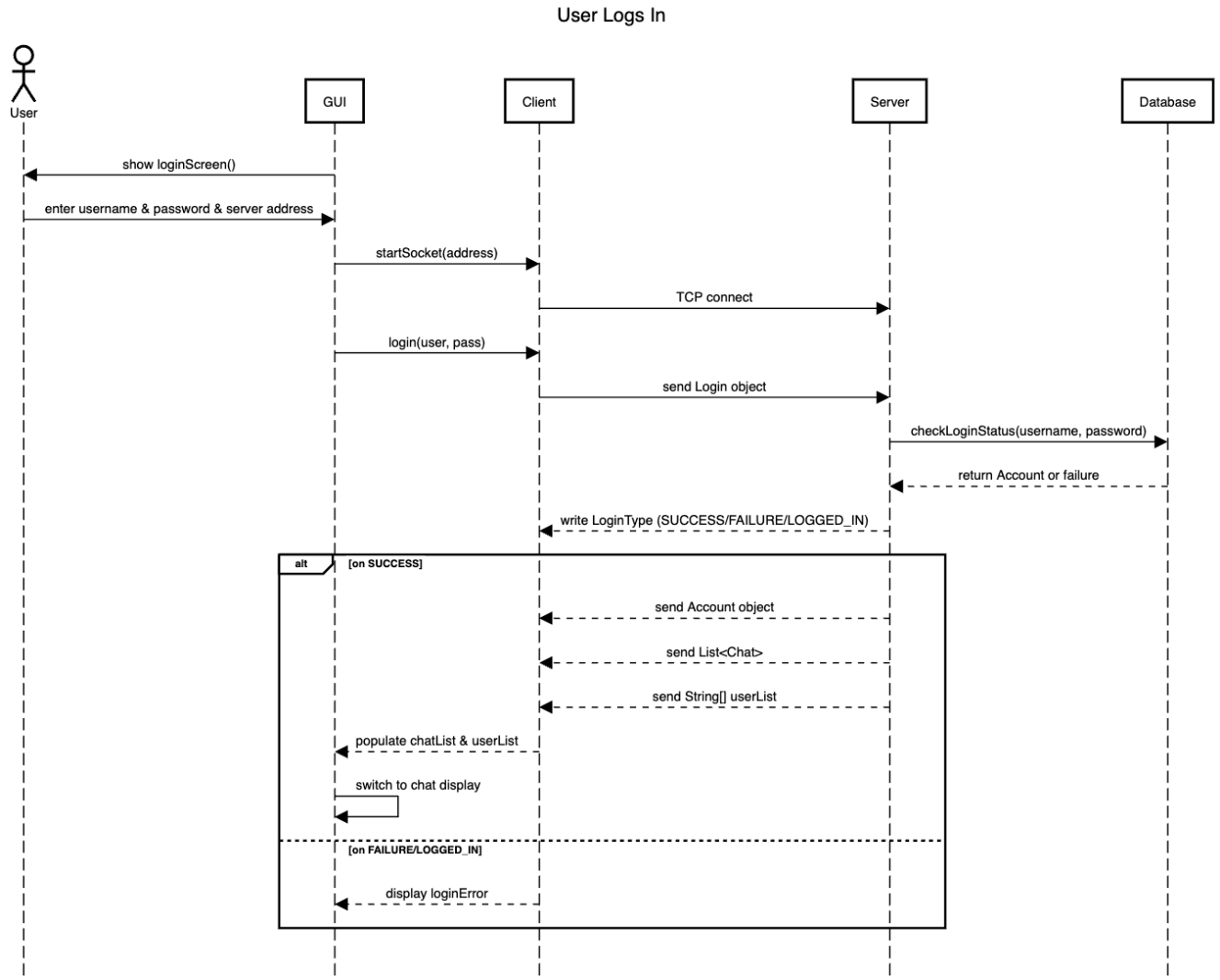


Client:

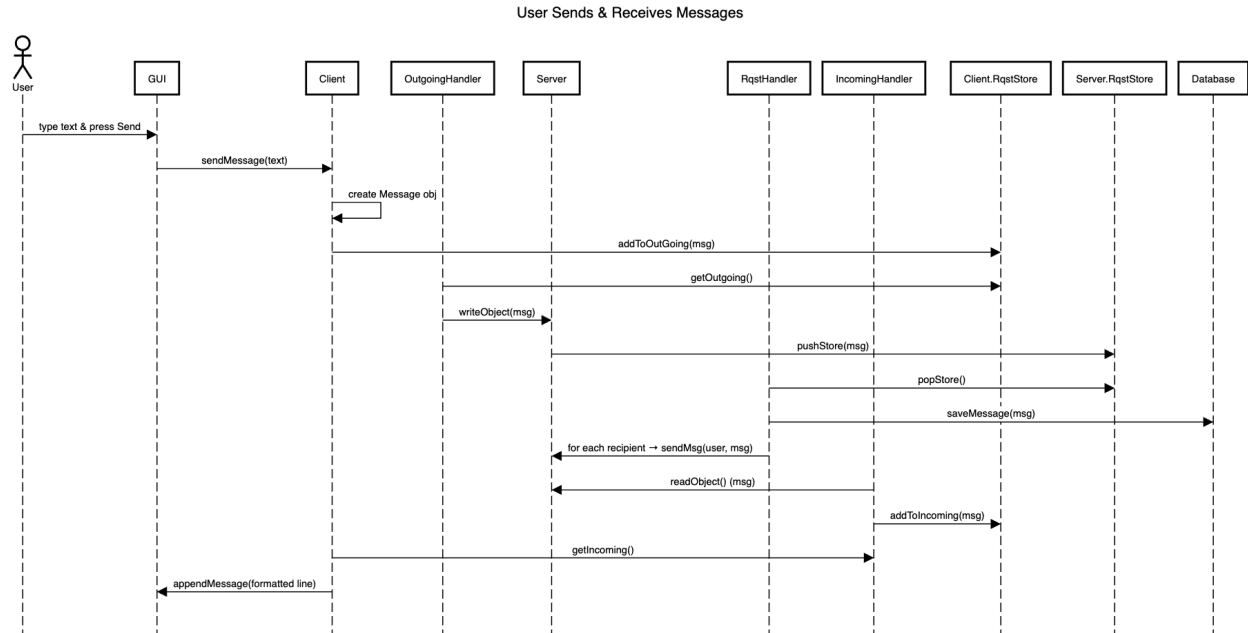


UML Sequence Diagrams

Use Case 1: User Logs In

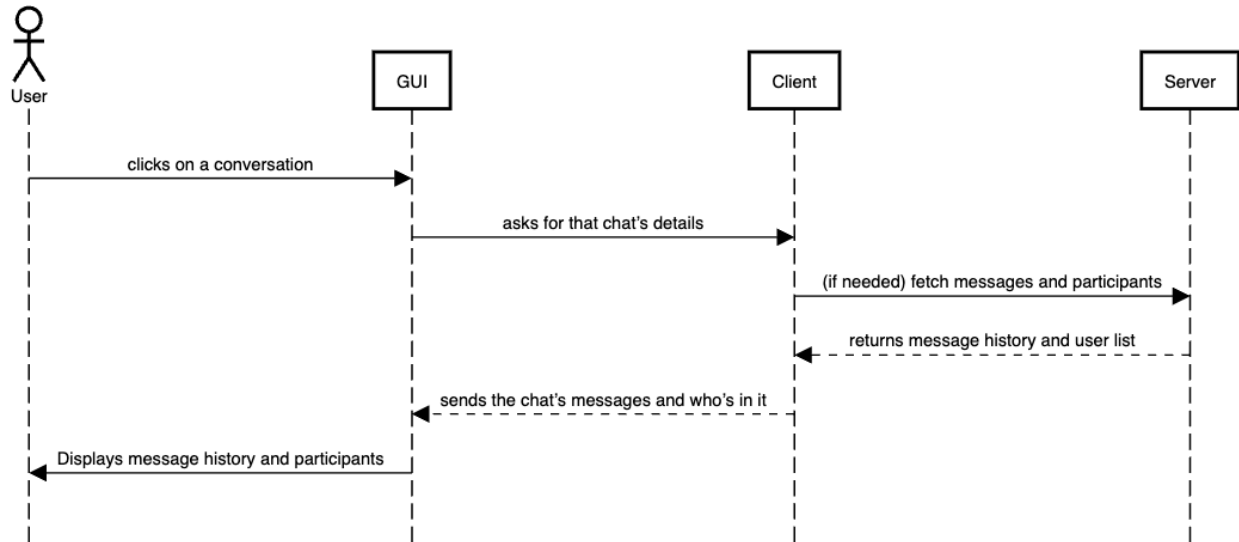


Use Case 2: Sending & Receiving a Message

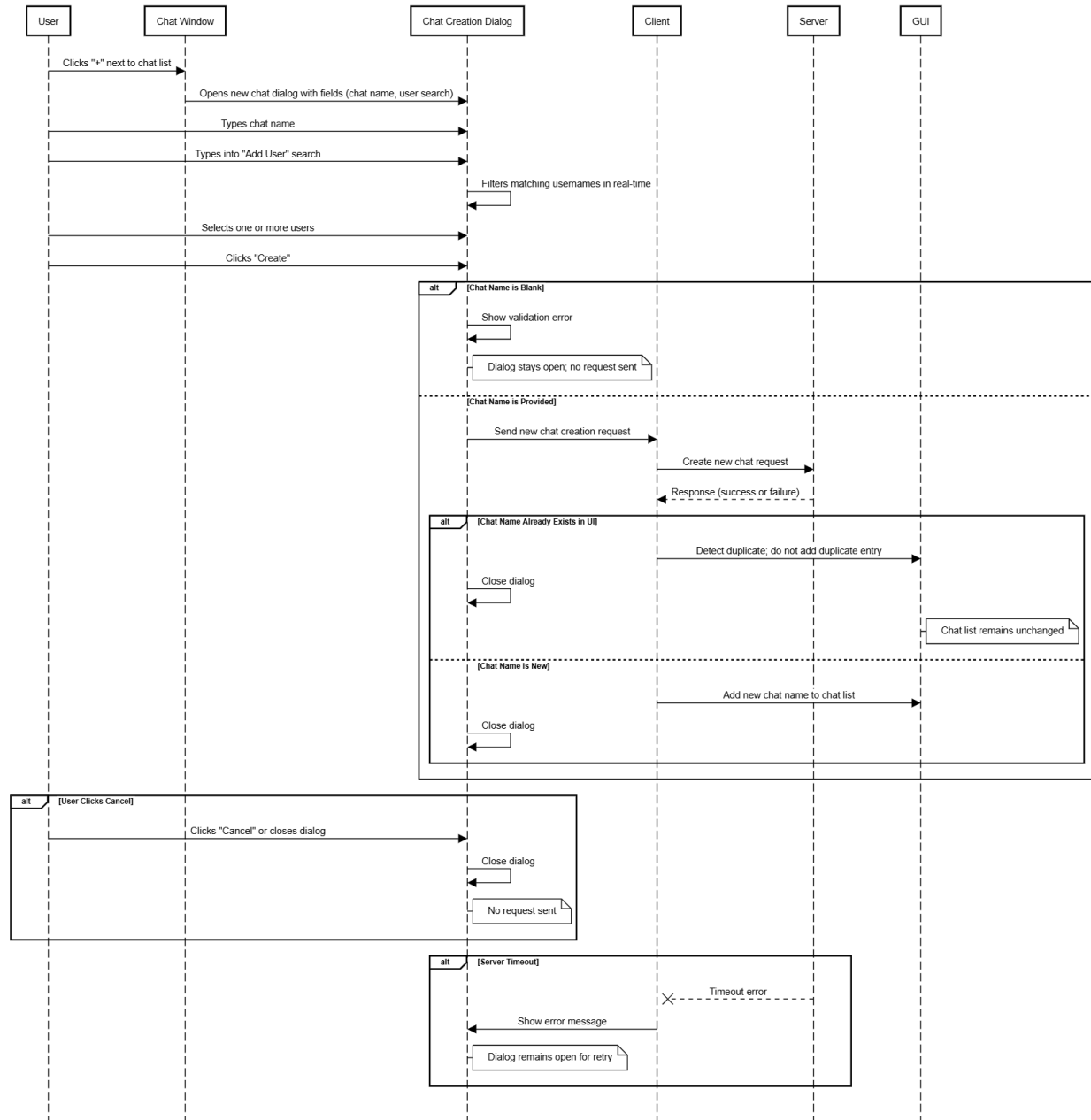


Use Case 3: User Views and Selects a Chat

User Views and Selects a Chat

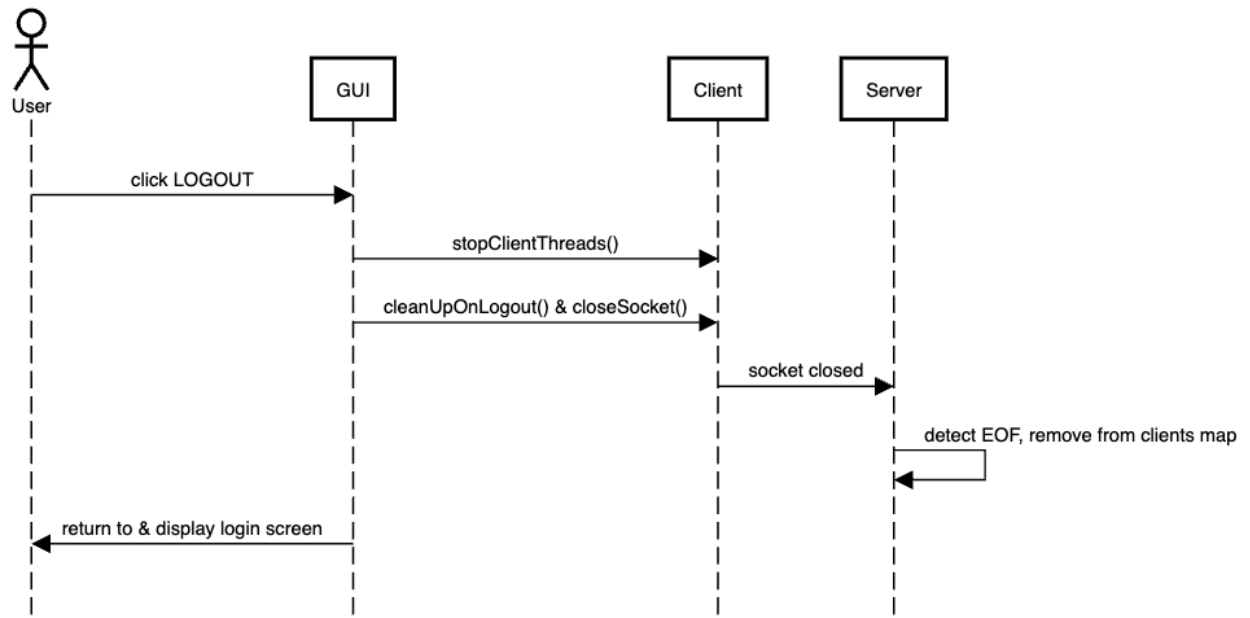


Use Case 4: User Creates Chat with Other Users

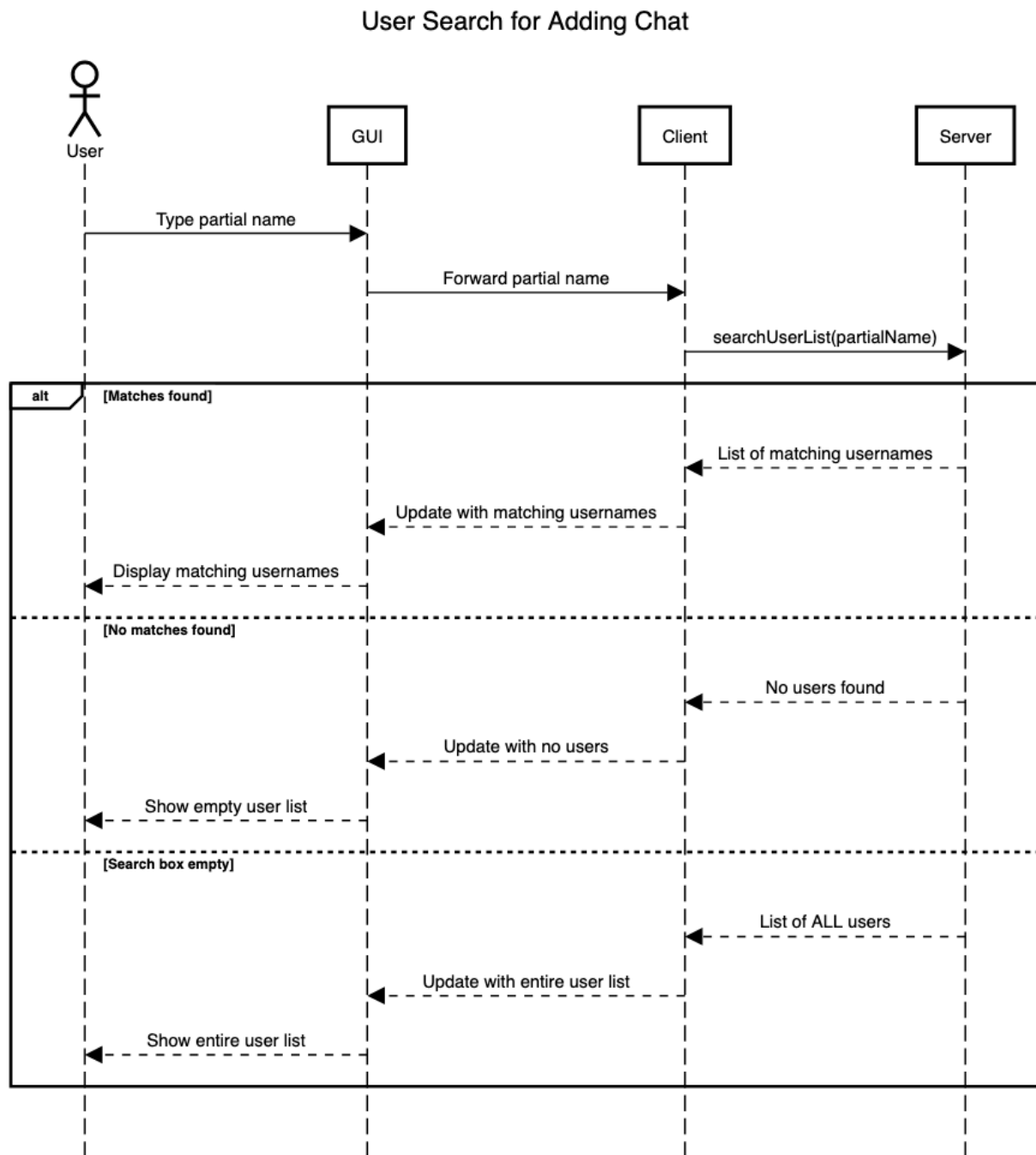


Use Case 5: User Logs Out

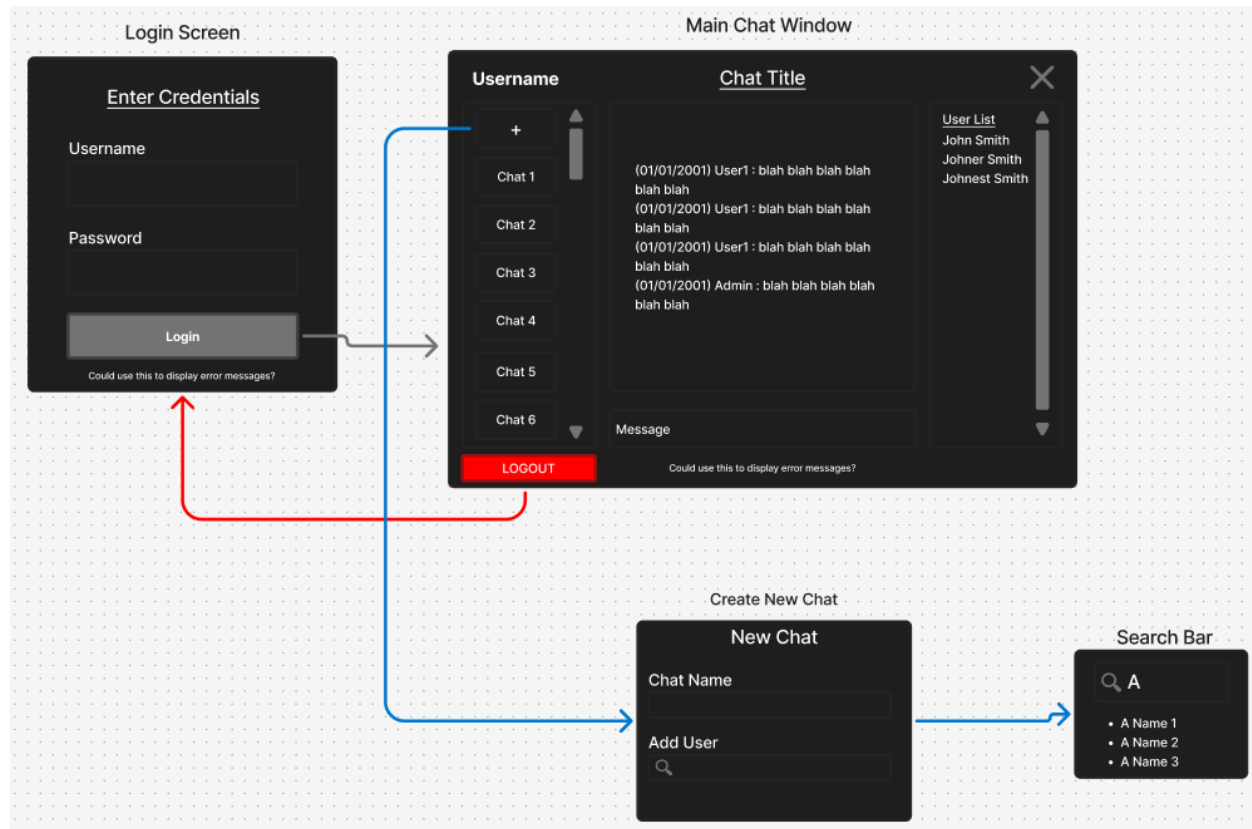
User Logs Out



Use Case 6: User Search For Adding Chat



GUI Concept Diagram



Figma Link:

<https://www.figma.com/board/9q77PtpLaQqwhea6qjYmXv/GUI-Concept?node-id=0-1&t=MMfCgX2ORltdWPO8-1>

Gantt Chart

