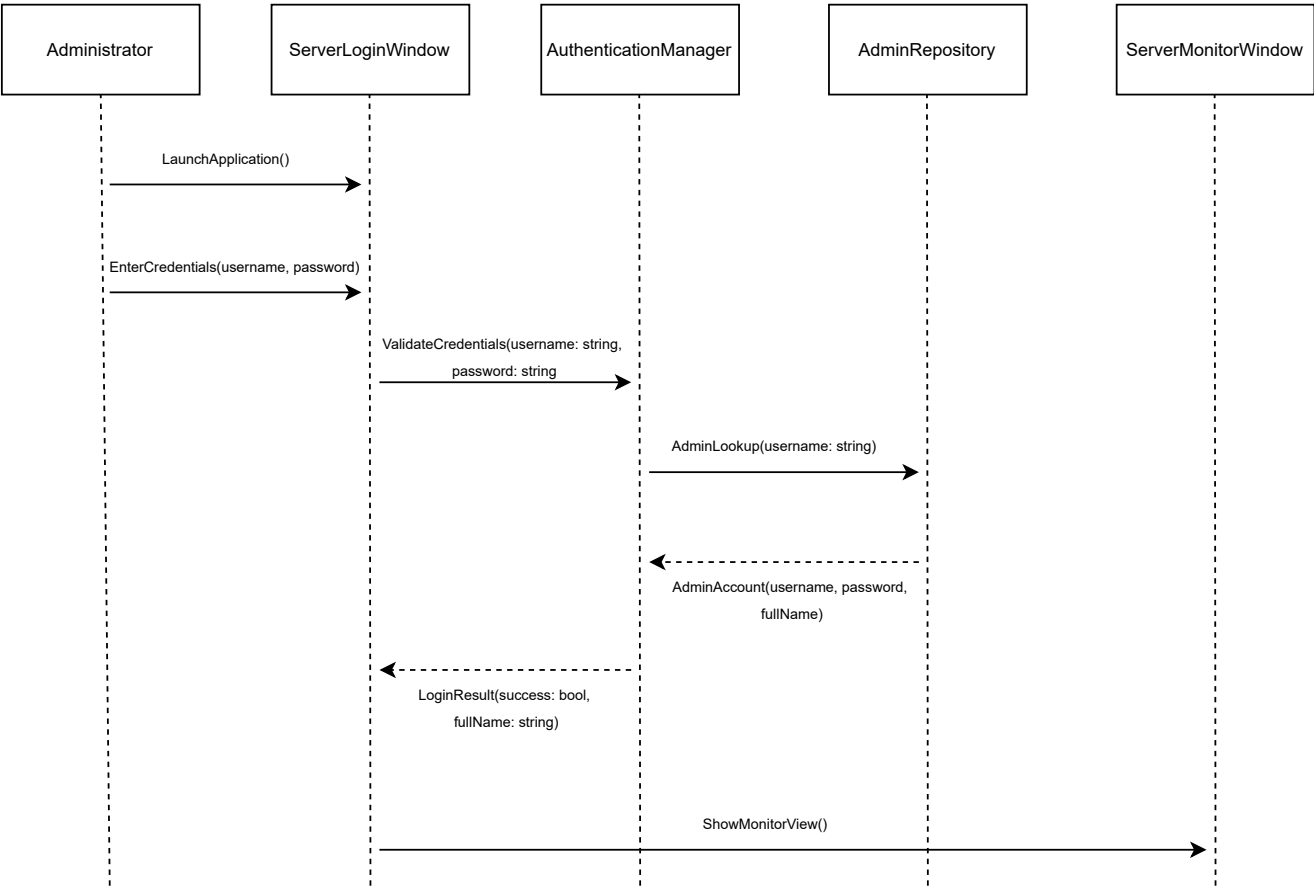
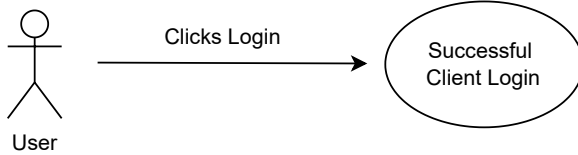


UC-05: Successful Server (Administrator Login - Sequence Diagram



### UC-01: Successful Client Login



#### Actor sees, does, enters

- User launches the client application
- Sees login form with username and password fields
- Enters username and password
- Login button becomes enabled
- Clicks Login button

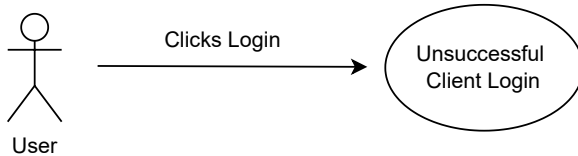
#### What the app does

- Formats credentials into a JSON login request
- Sends request to server via the WebSocket
- Server validates credentials using the .json
- Server sends success response with user's full name

#### Actor sees after

- Login form closes
- Main app view is displayed

### UC-02: Unsuccessful Client Login



#### Actor sees, does, enters

- User launches the client application
- Sees login form with username and password fields
- Enters username and password
- Login button becomes enabled
- Clicks Login button

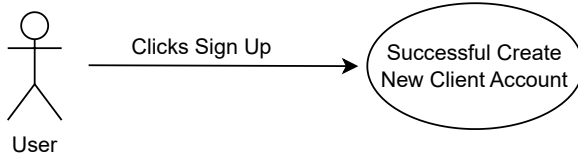
#### What the app does

- Formats credentials into a JSON login request
- Sends request to server via the WebSocket
- Server validates credentials using the .json
- Credentials are invalid, server sends failure response

#### Actor sees after

- Login form remains visible
- Error message is displayed indicating that login was unsuccessful

### UC-03: Successful Create New Client Account



#### Actor sees, does, enters

- User launches the client application
- Sees login form and clicks the sign-up button
- Registration form appears
- Enters full name, username, and password
- Submits the registration form

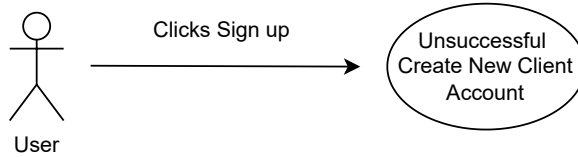
#### What the app does

- Sends account creation request to server
- Server checks that username is available
- Server creates new account in the .json
- Server sends success response with user's full name

#### Actor sees after

- Registration form closes
- User is logged in automatically
- Main application view becomes visible

#### UC-04: Unsuccessful Create New Client Account



##### Actor sees, does, enters

- User launches the client application
- Sees login form and clicks the sign-up button
- Registration form appears
- Enters full name, username, and password
- Submits the registration form

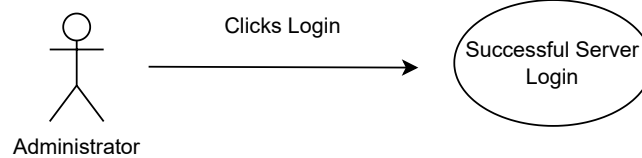
##### What the app does

- Sends account creation request to server
- Server checks that username is available
- Username is already taken, server sends failure request

##### Actor sees after

- Registration form stays visible
- Error message is displayed indicating username is unavailable

#### UC-05: Successful Server Login



##### Actor sees, does, enters

- Administrator launches the server application
- Sees login form with username and password fields
- Enters username and password
- Login button becomes enabled
- Clicks Login button

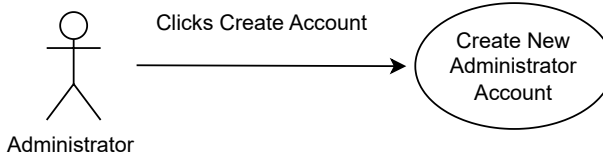
##### What the app does

- Passes credentials to the AuthenticationManager
- AuthenticationManager requests AdminRepository to look up the username
- AdminRepository reads .json and returns account data
- Authentication Manager compares passwords and confirms valid login

##### Actor sees after

- Login form closes
- Server monitoring dashboard is displayed
- Shows list of logged-in users and active chat rooms

#### UC-06: Creates New Administration Account



##### Actor sees, does, enters

- Administration launches the client application
- Clicks the option to create a new account
- Registration form appears
- Enters full name, username, and password
- Submits the registration form

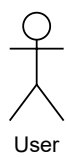
##### What the app does

- Checks that the username is not already taken in the .json
- Creates a new administrator account
- Stores the new account in the .json

##### Actor sees after

- Registration form closes
- Confirmation that the account was created successfully

### UC-07: Synchronize Client State After Login



Logs in Successfully

Synchronize Client State After Login

#### Actor sees, does, enters

- The user has just logged in successfully
- Main application view currently loading

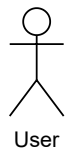
#### What the app does

- Client request the user's contact list from the server
- Server sends back contact list with an online/offline status
- Client updates the local state with server data

#### Actor sees after

- Contact list is displayed with current online/offline status
- Main application is fully loaded and ready to use

### UC-08: Add Contact



Clicks Add Account

Add Contact

#### Actor sees, does, enters

- The user is in the main application view
- Selects the option to add a contact
- Enters the username of the contact to add

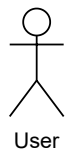
#### What the app does

- Sends add contact request to the server
- Server checks if the contact username exists in the .json
- Contact exists, server adds them to the user's contact list
- Server sends confirmation back to the client

#### Actor sees after

- New contact appears in the contact list
- Contact's online/offline status is displayed

### UC-09: Remove Contact



Clicks Remove Contact

Remove Contact

#### Actor sees, does, enters

- The user is in the main application view
- Selects a contact from the contact list
- Clicks the option to remove the contact

#### What the app does

- Sends remove contact request to server
- Server removes the contact from the user's contact list
- Server sends confirmation back to the client

#### Actor sees after

- Contact is removed from the contact list
- Contact list updates to reflect the change

#### UC-10: One-to-One Chat



Clicks Start Chat

One-to-One Chat

##### Actor sees, does, enters

- The user is in the main application window
- Selects an online chat from the contact list
- Clicks start chat
- Types a message and clicks send

##### What the app does

- Client sends chat request to the server
- Server creates a chat session between the two users
- Messages are routed through the server to the other user
- Messages are timestamped and tagged with sender's username

##### Actor sees after

- Chat window opens showing conversation history
- Messages appear with timestamps and the names of the sender
- User can continue sending and receiving messages

#### UC-11: Add Participant to Chat



Clicks Add to Chat

Add Participant to Chat

##### Actor sees, does, enters

- The user is in an active one-to-one chat
- Selects the option to add a participant
- Chooses an online contact that is in common with both chat participants

##### What the app does

- Client sends add participant to server
- Server verifies the third user is online and in common with both participants
- Server adds the third user to the chat session
- Server notifies all participants of the updated chat

##### Actor sees after

- Chat window updates to show the new participant
- All three users can now send and receive messages

#### UC-12: Participants in Multiple Chats



Start another chat

Participants in Multiple Chats

##### Actor sees, does, enters

- The user is already in an active chat
- Selects another online contact from the contact list
- Starts a new chat session with that contact

##### What the app does

- Client sends new chat request to server
- Server creates an additional chat
- Server manages multiple active chat sessions for the user
- Messages are routed to the correct chat session

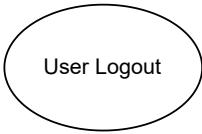
##### Actor sees after

- A new chat window opens alongside the existing one
- User can switch between and participate in both chats at the same time

**UC-13: User Logut**



Clicks Logut



- Actor seeds, does, enters**
  - The user is in the main application window
  - Clicks the logout option
- What the app does**
  - Client sends logout request to the server
  - Server removes user from active users list
  - Server updates the user's status to offline for all contacts
  - Server closes any active chat sessions involving the user
  - WebSocket connection gets lost
- Actor see after**
  - Main application view closes
  - Login form is displayed again
  - Other users see this user's status change to offline

UC-01: Successful Client Login

