|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | UC001 | | |
| Use Case Name: | Establish WebSocket Connection | | |
| Created By: | Sushmita/Krishu /Nam | Last Updated By: | Sushmita Nam  Krishu |
| Date Created: | 08/04/2023 | Date Last Updated: | 08/05/2023 |
| Actors: | User (WebSocket client),Server | | |
| Description: | This use case describes the process of establishing a WebSocket connection between the application’s WebSocket server and the clients. | | |
| Preconditions: | The user opens the application and wants to connect to the WebSocket server. | | |
| Postconditions: | The application successfully connects to the WebSocket server. | | |
| Normal Flow: | 1. The main method is called to start the application. 2. An instance of the App class is created with the server URI and port (e.g., ws://localhost:8081). 3. The connect method is called to initiate the WebSocket connection. 4. The on Open method is called when the connection is successfully established. A message is displayed on the console indicating that the connection is open. 5. The onMessage method is called when a message is received from the WebSocket server. The message is processed and logged using the Logger class. 6. If the received message contains a "client\_id", the client ID is extracted from the message and displayed on the console. 7. The onClose method is called when the WebSocket connection is closed. A message is displayed on the console indicating the reason for the closure. 8. The onError method is called if any error occurs during the WebSocket connection. The error is printed to the console. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | UC002 | | |
| Use Case Name: | Block User Message | | |
| Created By: | Sushmita/krishu /Nam | Last Updated By: | Sushmita  Nam  Krishu |
| Date Created: | 08/04/2023 | Date Last Updated: | 08/05/2023 |
| Actors: | User (Sender), Server | | |
| Description: | This use case represents the process of blocking a user in the web chat application. | | |
| Preconditions: | The user wants to block/ignore messages sent by another active user. | | |
| Postconditions: | The server receives a BlockUserMessage object from the sender and the sender will no longer receive messages from the blocked user. | | |
| Normal Flow: | 1. The user enters the name of an active user they want to block/ignore. 2. The client creates a BlockUserMessage object and sets the Type property to the blocked user's name or identifier, and the Text property to the content of the chat message. 3. The server processes the message and identifies the recipient based on the property. 4. The server prevents messages from the blocked user to be transmitted to the recipient. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | UC003 | | |
| Use Case Name: | Manage Chat Room | | |
| Created By: | Sushmita/Krishu/Nam | Last Updated By: | Sushmita  Nam  Krishu |
| Date Created: | 08/04/2023 | Date Last Updated: | 08/05/2023 |
| Actors: | User (Admin), Server | | |
| Description: | This use case describes the management of a chat room in the web chat application. It includes adding and removing users from the chat room and broadcasting messages to all users in the chat room. | | |
| Preconditions: | The user logged in to the application and wants to create/join in a chat room. | | |
| Postconditions: | The user has successfully created a new chat room or joined a currently active chat room. | | |
| Normal Flow: | 1. The ChatRoom class is created. 2. A chat room is initialized with a name, and an empty list of users. 3. The addUser method is called to add a user to the chat room. The user is added to the list of users. 4. The removeUser method is called to remove a user from the chat room. The user is removed from the list of users. 5. The broadcast method is called to send a message to all users in the chat room. The message is saved to the log using the Logger class. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | UC004 | | |
| Use Case Name: | Log Message with Timestamp | | |
| Created By: | Sushmita/Krishu/Nam | Last Updated By: | Sushmita  Nam  Krishu |
| Date Created: | 08/04/2023 | Date Last Updated: | 08/05/2023 |
| Actors: | Server, User | | |
| Description: | The Logger class is responsible for logging messages with timestamps to a JSON file. | | |
| Preconditions: | The server receives a login message from a user and the user wants to interact with the application. | | |
| Postconditions: | The application has recorded the actions the user has taken while being logged in. | | |
| Normal Flow: | 1. The saveToLog method is called with the message to be logged.  2. The current timestamp is generated using the LocalDateTime.now() method and formatted to the desired format using DateTimeFormatter.  3. A new JSON object is created with the message and timestamp properties.  4. The existing content of the log file, if any, is read.  5. The new log entry is added to the existing list of logs.  6. The list of logs is converted to a JSON array using Gson.  7. The JSON array is written back to the log file. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | UC005 | | |
| Use Case Name: | Send Login Message | | |
| Created By: | Sushmita/Krishu/Nam | Last Updated By: | Sushmita  Nam  Krishu |
| Date Created: | 08/04/2023 | Date Last Updated: | 08/05/2023 |
| Actors: | User, Server | | |
| Description: | The LoginMessage class is used to send login information from the client to the server. | | |
| Preconditions: | The user wants to log in to the web chat application. | | |
| Postconditions: | The server receives the login message with the user's name. | | |
| Normal Flow: | 1. The user provides their name during the login process.  2. The client creates a LoginMessage object and sets the Name property to the user's name.  3. The client sends the LoginMessage object to the server. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | UC006 | | |
| Use Case Name: | Send Message | | |
| Created By: | Sushmita/Krishu/Nam | Last Updated By: | Sushmita  Nam  Krishu |
| Date Created: | 08/04/2023 | Date Last Updated: | 08/05/2023 |
| Actors: | User | | |
| Description: | The Message class is used to transmit messages between the server and the clients. | | |
| Preconditions: | The user wants to send a message to another user in the web chat | | |
| Postconditions: | The server receives the message and processes it.  The message is sent to the specified recipient. | | |
| Normal Flow: | 1. The user composes a message and specifies the recipient's ID. 2. The client creates a Message object and sets the from property to the sender's ID, the property to the recipient's ID, and the content property to the message text. 3. The client sends the Message object to the server. 4. The server processes the message and identifies the recipient based on the property. 5. The server sends the message to the specified recipient. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | UC007 | | |
| Use Case Name: | Receive Chat Message | | |
| Created By: | Sushmita/Krishu/Nam | Last Updated By: | Sushmita  Nam  Krishu |
| Date Created: | 08/04/2023 | Date Last Updated: | 08/05/2023 |
| Actors: | User, Server | | |
| Description: | The RecvChatMessage class is used to represent the messages received by the client from the server in the web chat application. | | |
| Preconditions: | The user is connected to the web chat application and has an active WebSocket connection. | | |
| Postconditions: | The user receives and displays the chat message from the server. | | |
| Normal Flow: | 1. The server sends a RecvChatMessage object to the client containing information about the chat message received. 2. The client receives the RecvChatMessage object. 3. The client extracts the list of online users (Users), the list of available chatrooms (Chatrooms), and the text content of the message (Text) from the RecvChatMessage object. 4. The client updates the user interface to display the list of online users and available chatrooms. 5. If the Text property is not empty, the client displays the received message in the chat interface. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | UC008 | | |
| Use Case Name: | Select Chat Message | | |
| Created By: | Sushmita/Krishu/Nam | Last Updated By: | Sushmita  Nam  Krishu |
| Date Created: | 08/04/2023 | Date Last Updated: | 08/05/2023 |
| Actors: | User, Server | | |
| Description: | The SelectChatMessage class is used to represent the message sent by the client to the server when the user wants to select a chatroom to join. | | |
| Preconditions: | The user is connected to the web chat application and has an active WebSocket connection. | | |
| Postconditions: | The server receives the SelectChatMessage object from the client.  The server processes the request and updates the user's chatroom membership accordingly. | | |
| Normal Flow: | 1. The user selects a chatroom from the list of available chatrooms in the user interface. 2. The client creates a SelectChatMessage object and sets the Name property to the name of the selected chatroom. 3. The client sends the SelectChatMessage object to the server through the WebSocket connection. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | UC009 | | |
| Use Case Name: | Send Chat Message | | |
| Created By: | Sushmita/Krishu/Nam | Last Updated By: | Sushmita Nam  Krishu |
| Date Created: | 08/04/2023 | Date Last Updated: | 08/05/2023 |
| Actors: | User, Server | | |
| Description: | The SendChatMessage class is used to represent the message sent by the client to the server when the user wants to send a chat message to a chatroom or another user. | | |
| Preconditions: | The user is connected to the web chat application and has an active WebSocket connection. | | |
| Postconditions: | The server receives the SendChatMessage object from the client.  The server processes the message and broadcasts it to the specified chatroom or user | | |
| Normal Flow: | 1. The user enters a chat message in the user interface. 2. The client creates a SendChatMessage object and sets the From property to the user's name or identifier, and the Text property to the content of the chat message. 3. The client sends the SendChatMessage object to the server through the WebSocket connection. | | |

Graphical Use Case Table:A diagram of a software company

Description automatically generated with medium confidence