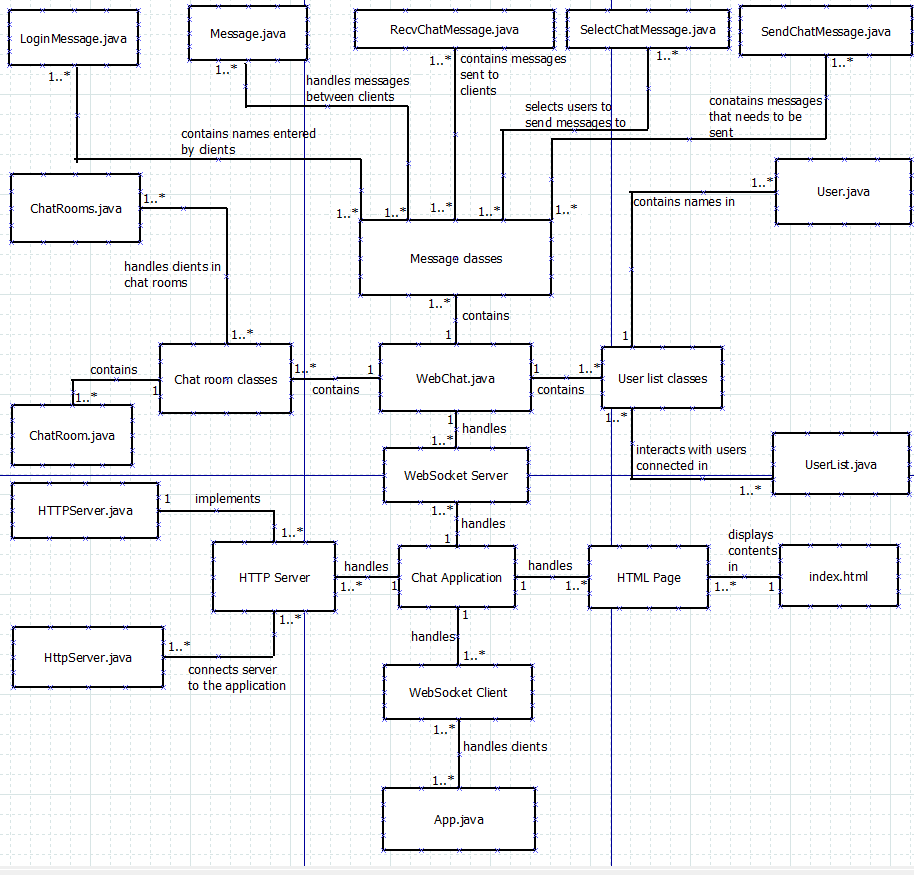
Iteration II – Group 4

Krishu Panta  
Nam Nguyen  
Sushmita Dhital

Software Design  
  
Introduction:   
  
Overall, this design separates the responsibilities into three main files: the WebSocket server, the HTTP server, and the HTML page. The WebSocket server handles the WebSocket connections and message broadcasting, the HTTP server serves the UI file, and the HTML/JavaScript page provides the user interface and interacts with the WebSocket server to send and receive messages.   
  
WebSocket Server:   
+ Responsibility: Establish and manage WebSocket connections, handle message broadcasting.   
+ Design considerations:   
- Implement WebSocket protocol to handle client-server communication.   
- Maintain a list of connected clients and their respective WebSocket connections.   
- Receive incoming messages from clients and broadcast them to all connected clients.   
- Handle disconnections and remove clients from the list.   
  
HTTP Server:   
+ Responsibility: Serve the HTML file that acts as the user interface for the chat app.   
+ Design considerations:   
- Implement an HTTP server to serve static files, specifically the HTML page for the chat app.   
- Listen on the specified server port number provided as a command line argument.   
- Serve the HTML file to clients when they request the app's URL.   
- Handle any other HTTP requests if needed.  
  
HTML Page:   
+ Responsibility: Provide the user interface for the chat app and handle user interactions.  
+ Design considerations: Design an HTML page with the following elements:   
- "Received Messages" area to display incoming messages from other clients or chatrooms.  
- Text box to enter the user’s name that the user wants to display to other clients and chatrooms.  
- Text box for the user to enter the name of the chatroom that the user wants to join.  
- Text box to enter messages that the user wants to send to other clients or chatrooms.   
- Port number field to specify the server port number (to be entered by the user).   
- Display area to show the assigned client port number.   
- Use Java to handle user interactions, such as sending messages to the WebSocket server and updating the UI based on received messages.   
- Connect to the WebSocket server using the assigned client port number.   
- Handle incoming messages from the WebSocket server and display them in the "Received Messages" area.   
- Send user-entered messages to the WebSocket server for broadcasting.  
  
  
  
Context Diagram:  
A diagram of a webdesign

Description automatically generated

Class Diagram:  
  
  
  
+ App.java class:  
A screenshot of a computer program

Description automatically generated  
  
+ ChatRoom.java class:  
A screenshot of a computer program

Description automatically generated  
  
+ HttpServer.java class:  
A screen shot of a computer program

Description automatically generated  
  
+ LoginMessage.java class:  
A screenshot of a computer screen

Description automatically generated  
  
+ Message.java class:  
A screenshot of a computer program

Description automatically generated  
  
+ RecvChatMessage.java class:  
A screenshot of a computer program

Description automatically generated  
  
+ SelecChatMessage.java class:   
A screenshot of a computer

Description automatically generated  
  
+ SendChatMessage.java class:  
A screenshot of a computer

Description automatically generated  
  
+ User.java class:  
A diagram of a computer program

Description automatically generated  
  
+ UserList.java class:  
A screenshot of a computer program

Description automatically generated  
  
+ WebChat.java class:  
A screen shot of a computer

Description automatically generated