Group 4: Communications

Class Diagrams

Revision History

Date	**Revision**	**Description**	**Authors**
9/19/2022	1.0	Initial Version	Jared Patterson
9/26/2022	1.1	Added the Class Diagrams	Aimee Diaz
10/30/2022	2.0	Updated With More Specific Design	Jared Patterson

Note: This UML is currently lax about constructors, setters, getters. Those may be implied as existing, where necessary.
Note: Members are assumed to be private; methods, unless otherwise specified, are assumed to return void.
Note: Members shown as arrays might be better as lists, or some other data structure. For now the array typing simply indicates that multiple of something may be stored.

Note: Multiple threads might have reason to write to a member in Server, so members in Server might need to be made static in implementation. A synchronization scheme might also be necessary during implementation.
Note: In a distant future implementation, it may make sense to organize things such that loading and saving to files happens on a more granular level.

Server
Members: User[] users ChatRoom[] chatRooms
Methods: +main() -load(String usersFilename, String chatRoomsFilename) -save(String usersFilename, String chatRoomsFilename) -startNewClientHandler(Socket clientSocket)

Client
Members: GUI gui Socket serverSocket
Methods: +main() -startGUI() -chatMessageNotification()

Note: The passing down of serverSocket simplifies the interplay of socket IO and GUI functionality. It may also come to pass during implementation that we want to pass state down as well, for simplicity in updating it (this is not yet accounted for in the UML).
Note: Client.main() cannot call wait(), and would have to busy-wait for GUI stuff to happen before resuming control and starting a new part of the GUI. I think it makes sense to start GUI.run() as a thread, which can wait() and can handle all things appropriately without busy-waiting. Also, Client.main() would be free to listen for incoming chat messages.

GUI
Members:
Socket serverSocket
UserData currentUser
enum State
State state
Methods:
+run()
-doChatRoomListView()
-doChatRoomView()
-doUserListView()
-doLoginView()

Note: JFrame in the views is not a comprehensive list of what shall be used for UI. It is only indicative of a start.
Note: Most methods will be called by JButtons, and input and output will be [gotten from]/[shown by] appropriate library UI objects.
Note: Methods like ChatRoomView.sendMessage() that seem like they should take arguments will get what they need from UI objects.

ChatRoomListView
Members:
Socket serverSocket
ChatRoomDescription[] crds
JFrame frame
Methods:
+run()
-requestChatRoomDescriptions()
-openChatRoom()

ChatRoomView
Members: Socket serverSocket ChatRoomData crd JFrame frame
Methods: +run() -requestChatRoomData() -refreshForNewMessage() -sendMessage()

UserListView
Members: Socket serverSocket UserData[] userData JFrame frame
Methods: +run() -requestUserData() -requestChatRoomDescriptions() -formNewChatRoom() -addUsersToChatRoom()

LoginView
Members: Socket serverSocket JFrame frame
Methods: +run() -sendLoginRequest() -sendLoginRequestAsIT()

Note: User has a clientSocket. This simplifies the distribution of ChatMessages within a ChatRoom.

ClientHandler
Members: User user
Methods: +run() -sendLoginResponse() -sendChatRoomDescriptions() -sendChatRoomData() -sendUserData() -distributeNewChatMessage()

Note: One socket per user makes the assumption a user will not login on multiple devices at once.
Note: IDs are meant to be unique and help handle similar Users (e.g. they have the same name).

User
Members: Socket clientSocket String name String username String password ID id boolean isIT ChatMessageData[] pendingMessages
Methods: +getUserData() : UserData

UserData
Members: String name ID id boolean isIT
Methods:

Note: IDs are meant to be unique and help handle similar ChatRooms (e.g. they have the same users, but are two different instances). Also, it probably makes sense to not allow two instances of a DM.
Note: It may be nice to add a group name member to ChatRoom.

ChatRoom
Members: User[] users ChatMessage[] messages ID id
Methods: +isDM() : boolean +getChatRoomData() : ChatRoomData +getChatRoomDescription() : ChatRoomDescription

ChatRoomData
Members: UserData[] users ChatMessage[] messages ID id
Methods: +isDM() : boolean

ChatRoomDescription
Members: Users[] users ChatMessage mostRecentMessage ID id
Methods:

Note: It may be nice to add a time sent member to ChatMessage.
Note: Sent by both client and by server. Received by both client and by server.

ChatMessage
Members: String senderName ID senderID ID groupID String contents
Methods:

Note: Sent by client, received by server.

LoginRequest
Members: String username String password
Methods:

Note: Sent by server, received by client.

LoginResponse
Members: UserData whoLoggedIn
Methods: +wasLoginSuccessful() : boolean

Note: Sent by client, received by server.

Request
Members: enum Type Type requestType
Methods:

Note: Sent by server, received by client.

ChatRoomDescriptionsResponse
Members: ChatRoomDescriptions[] crds
Methods:

Note: Sent by server, received by client.

ChatRoomDataResponse
Members: ChatRoomData crd
Methods:

Note: Sent by server, received by client.

UserListResponse
Members: UserData[] userList
Methods:
