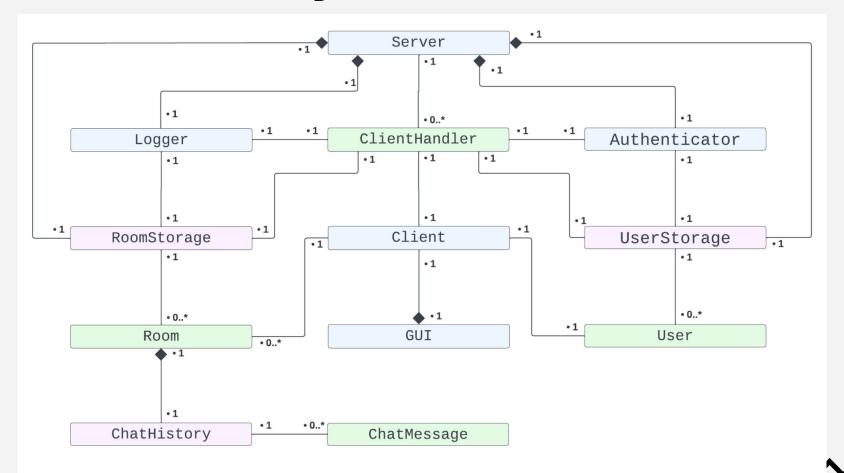
Design Presentation

Group #4 - Communications



Big Picture UML



Message

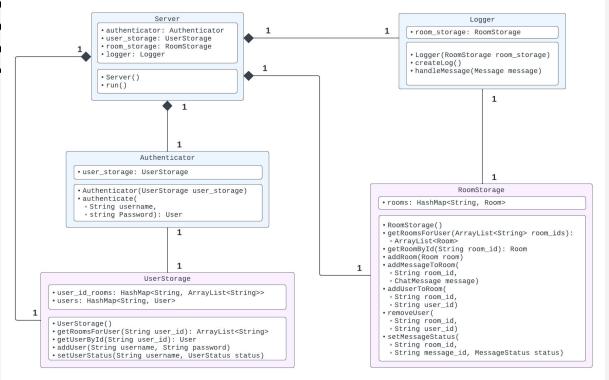
Passed back and forth between Server and Client

Message

- type: MessageType users: String[] · username: String · password: String login_status: String room id: String · user id: String · contents: String • rooms: ArrayList<Room>
- Message(MessageType type)
- setUsername(String username)
- setPassword(String password)
- setUsers(String[] users)
- setLoginStatus(String status) setRoomId(String room id)
- setUserId(String user id)
- setContents(String contents)
- setRooms(ArrayList<Room> room)
- getType(): MessageType
- getUsername(): String
- getPassowrd(): String
- getLoginStatus(): String
- getRoomId(): String getUserId(): String
- getContents(): String
- getRooms(): ArrayList<Room>

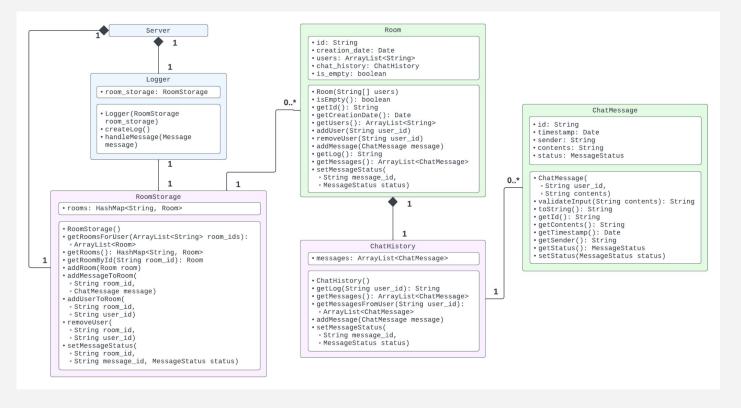
enum MessageType

- Login
- Logout
- NewChat
- · CreateRoom
- LeaveRoom
- AddToRoom
- ChangeStatus
- UpdateUserStatus



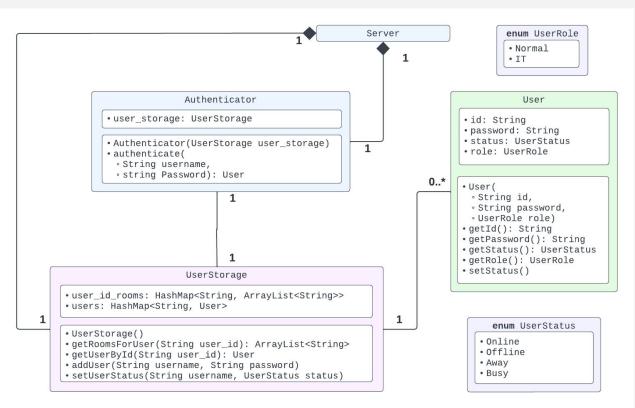
Server Interactions

- Server owns everything:
 - Authenticator
 - Logger
 - UserStorage
 - RoomStorage
- Authenticator has UserStorage and handles login
- Logger has RoomStorage and performs actions on rooms which are all logged



- Logger can create a log of messages from each room
- RoomStorage holds and performs actions on Rooms
- Messages are stored in a Room's ChatHistory



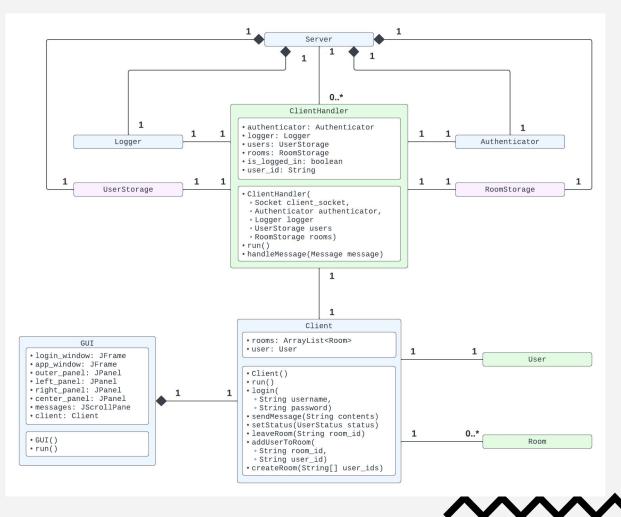


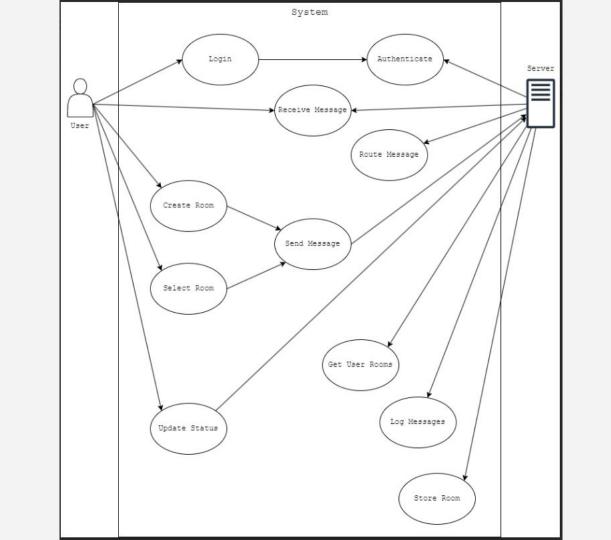
Authenticator, UserStorage & User

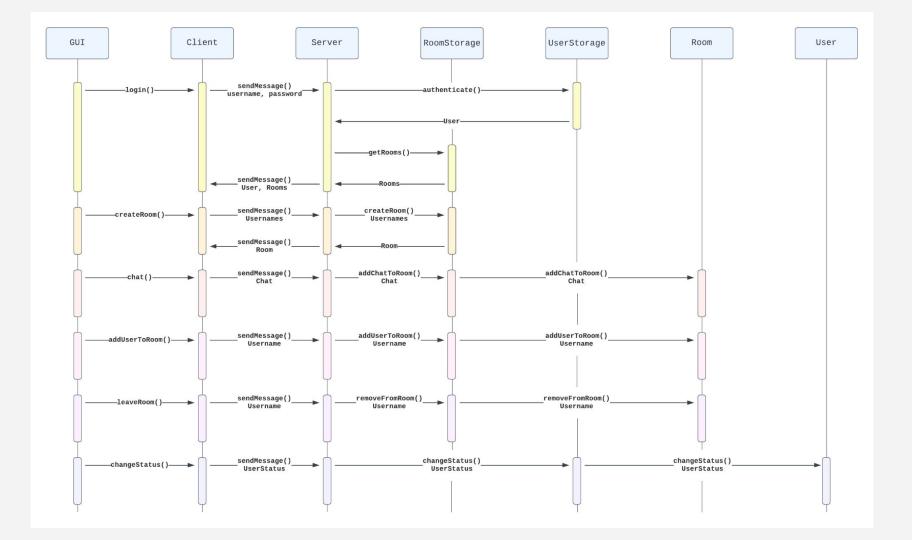
- Authenticator verifies user credentials
- UserStorage stores users by their usernames and performs actions on Users
- Usernames are also associated with a list of rooms the user is in
- Users have a status that they can update through the GUI

Client, Client Handler & GUI

- Each ClientHandler runs on a thread and handles a single client
- ClientHandler sends User and Rooms to the client
- Users perform actions through the GUI







Thank You!