Final Project Guidelines

The final project is a hybrid P2P and client-server messenger application that enable two clients to send and receive messages between one and the other.
The developed project will implement the following functionality:

1. The project will include two apps: Messenger Server App (MSA) and Messenger Client App (MCA).
2. The communication between the MCA and MSA will be TCP based (control messages).
3. The communication between two MCA's will be UDP based (interaction between two clients).
4. To connect to the server the client will sign in using a username and a password (encryption is not required).
    - The client can create a new user with a new password (unless this user is already created).
    - To communicate with another client, the user will request to open a session with the required user name.
    - The list of usernames and passwords will be saved into a file in the server side.
5. Chat rooms (conference chat):
    - The client is able to open a chat room.
    - A chat room may be closed only by its creator.
    - Each client can enter a chat room or leave it.
    - Once a client enters a chat room it will receive all messages sent by all the other clients in the room, and all the clients will receive his messages.
    - In order not to overload the server, all client communication is P2P using UDP and does not pass through the server.
6. Messenger Server App CLI:
    - lu - list all users
    - lcu - list all connected users

- ls - list all sessions (two clients communicating)
- lr - list all rooms
- lru <room name> - list all users in this room
- x - shutdown

7. Messenger Client App CLI:
   - c <IP> - connect to the server in the given ip
   - lu - print the user list from the server
   - lcu - print the connected users list
   - lr - print all rooms
   - lru <room name> - print all users in this room
   - login <user> <password> - login with the user and password
   - register <user> <password> - register the new user with the given password and login the user.
   - o <username> - open a session with the user
   - or <room name> - enter a chat room

   user can be connected to only to one other user or chat room at a time, calling o or or commands will disconnect other open session.
   - s <message> - send a message
   - l - print the current status of the client (connected to “xyz”/not connected)
   - cs - disconnect the open session / exit from a room
   - d - disconnect from server
   - x - close the app

   The user will be notified in each change in the client state with a proper message printed to the screen.
8. Each incoming message will be printed in the format:
   >[Moshe] What's up?

The project can be programmed by a group of Maximum two students!!! No exceptions will be permitted, any student that can't find a partner will do the project on his own.

The project frontal exam (“הגנה”) date will be publish later on.

Project grading criterions:

- Functional completeness and correctness: 50%

- Classes design, code structure, clarity, documentation: 30%
- General evaluation: 20%

The project should be submitted to the moodle in a zip file containing the complete eclipse project.