

ECE 316 - Operating Systems and Networking Laboratory

Final Project: Chat Server

Project overview: The chat server is responsible for providing a mechanism for the clients to communicate. This includes allowing clients to login so they can be identified by username and forwarding messages from one client to another. This also includes listing the usernames of the clients that are currently logged in to facilitate communication.

The chat application allows multiple clients to connect to a chat server. Connected clients can: log in with a username, list the clients that are already logged in, initiate conversations with other already logged in clients, and log out. The server must accept and maintain connections to all clients and relay chat messages between them.

The server will use threads in order to accommodate all the clients in parallel. The server will hold a public record of the names of the clients that are currently connected and the pairs of clients with active conversations. The record should only be accessible by one thread at a time and should be updated according to which user is connected and which is not.

The client: The client is separated in two applications/programs. The first, is the main communication terminal that is responsible for the communication with the server and for assigning and monitoring new conversations. The second client application is the connection terminal which is responsible for initiating a conversation assigned by the main communication terminal. All initiations of new conversations must be originated from the main communication terminal which will start a connection terminal (**new terminal**) for each new conversation.

Main communication terminal: The main communication terminal is responsible for connecting with the communication server, assigning new conversations and handing it over to one of its connection terminals. Also, it should monitor and keep a record about the client conversation pairs of its connection terminals.

The main communication terminal must be able to handle the following commands:

1. login <username> → it's the first command that the main communication terminal should send to the communication server. Only the first name is needed.
2. list → It asks the communication server to send the names of all the currently connected clients.
3. Initiate <name> → It starts a new connection terminal passing it all the information necessary (server port, current clients name, username of the second client in the conversation, etc.) for starting a conversation with the client indicated by the parameter "username".
4. active → Prints a list of all the connection terminals that he activated.
5. bye → It ends all its active connections and informs the communication server accordingly (log out).

Also, it should be able to receive a notification from the communication server for a new conversation initiated by another client. If such a notification is received, then it must start a new connection terminal passing it all the necessary information about the new conversation.

Each line must begin with a command. The parameters (if any) must be separated by a space and enclosed between <>. Note that, the commands should always be given as stated above. Commands in any other format should not be considered.

Connection terminal: The connection terminal is responsible of handling the conversation between two connection terminals of two different clients. It must connect with the communication server using the information given by the main communication terminal and be able to send and receive messages through the server to/from another connection terminal.

Also, it must be able to accept the console command “bye” that will terminate the current conversation by informing the main communication terminal and the communication server accordingly.

The messages sent and received must be in the following format:

- a) “To username : ”+ message to be sent
- b) “From username : ”+ received message

Communication Server: It has to assign a thread for every active conversation. It must hold a record of each client that is connected and for the active conversation pairs which only one thread must be able to modify at each time. When a main communication terminal is connected, the server must immediately ask the client to provide his login information. The login information must be correctly sent and received before any other command is initiated. When a connection terminal is connected it must add the new user pair of the conversation to the public record.

The server responses to each command from a main communication terminal are listed below:

1. login <username> : Adds the username to a public record of all the connected clients.
2. list : Sends the names of all the connected clients except the name of the one that sent the command.
3. bye : Terminates all of clients conversations and removes any information related to that client from the public record.

The communication server responses to each command from a connection terminal are listed below:

1. initiate <username> → Informs the main communication terminal of the user indicated by the “username” parameter to start a new conversation with the user that has sent the command.
2. bye → It terminates the current conversation between the first client that sent the command and the second client of the conversation. Also informs the second client about the termination of the conversation and removes the conversation from the public record.

Bonus – Group chat (50%): A client should be able to initiate a conversation consisting of multiple users. The command for this modification would be an extension of the initiate command mentioned above (e.g. initiate <username> <username> ...). Every message sent from any of the users in the conversation should be received from all the others stating the user that has transmitted it.

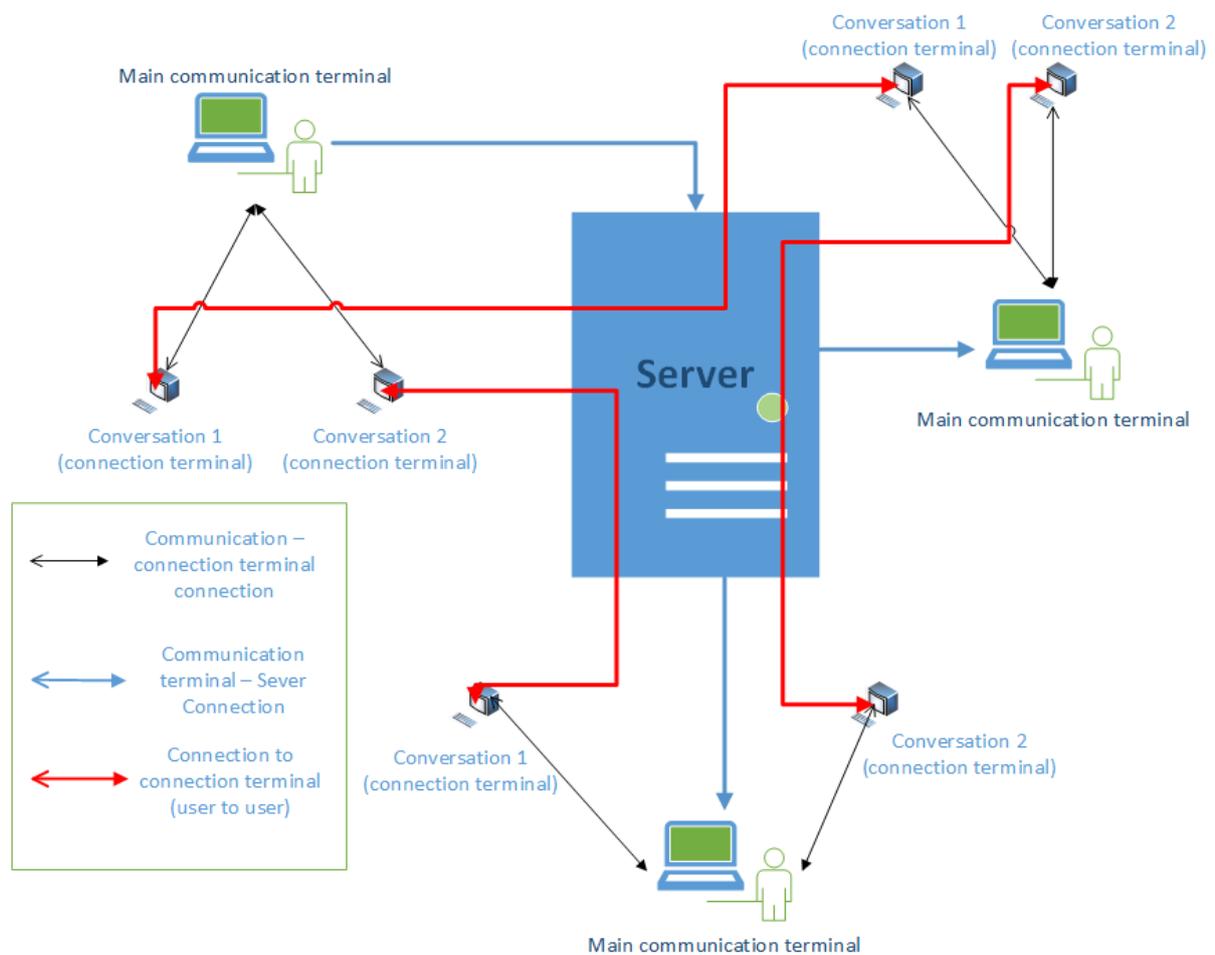
Grading: Completing the project 100%.

For completing the project without the main communication and connection terminals for the client but with only using one terminal(one program) for each client 75%.

Completing project and bonus 150%.



Diagram:



Deliverables: Your deliverable should include a detailed **report** (10 p. max.) and your **code**. The deliverables should be submitted electronically via email to lab teaching assistants (aanast01@ucy.ac.cy or ahadji29@ucy.ac.cy) prior the deadline (19/05/20 23:59). **Email subject line should only consist of "ECE316_2020_FinalProject"**. Naming format for the zip file: `lastName.zip` (or `lastName_lastName.zip` if a group). **Caution:** Remove the executables (.exe) from the files before you send them!

Examination: There will be two examination dates **07/05/20** and **20/05/20**. In the case a team prefers to be examined on 07/05 they should submit their project the **night before**. Please inform the teaching assistants about your preferred examination date before **05/05/20** in order to schedule the examination.