# **CSE 4074 Programming Assignment**: Peer-to-peer Chatting

## Due Date: January 9th, Sunday, 23:59

In this assignment you will implement a P2P chatting application according to the specification given below.

### Protocol Description

Your P2P protocol will use the centralized index approach. It should be consist of two parts: a centralized 'registry' and a chatting application between users. Your protocol will implement the three key operations: join, search, and chat.

**Join:** The registry will listen to a TCP socket at a known port. A new user will contact the registry at this port and register his/her username with a newly defined password. All the usernames should be unique. Each time a user wants to use the chatting system, he/she must contact the registry again, login with his username and password and indicate its presence. Then registry will know that the user is online and stores the IP address of the user. When the user want to leave the chatting system, he/she sends a 'LOGOUT' message to the registry and it appears as an offline user. In addition, the registry also listens to a UDP socket and every 60 seconds the user issues a 'HELLO' message to the UDP port of the registry. If the registry has not received a HELLO message from a user in 200 seconds, the user is removed from the list of online users.

**Search:** A user looking for a peer to chat, contacts to the registry and searches his/her username. The registry looks for the name of the peer and returns the contact address of the user if he/she is online. The registry also sends a 'NOT FOUND' message if there is no user with the given username.

**Chat:** The user contacts the requested peer and sends a 'CHAT REQUEST' message. If the requested peer accepts, he/she sends an 'OK' reply message and they start chatting. Otherwise he/she sends a 'REJECT' message or a 'BUSY' message (if he/she is already chatting with another peer).

You have to define details of your protocol. The user interface need not be sophisticated, but should be usable. It can be text-based or can be graphical. Your program should also provide **log files** which include all the messages sent and received by the peers and the registry.

### Implementation

You can use any programming language of your choice (Java, Python, C, C++, etc). You should use only socket programming. Do not use any sophisticated libraries downloaded from the web. You will need to use multi-threaded server processes. Although we have covered implementation of single threaded servers in the lecture, there are tons of materials that you may refer for the description of multi-threaded server implementation, and also it is expected that you have some familiarity from the Operating Systems course.

### Bonus

You are expected to provide chatting between two peers. However, if your application provides group chatting, you will get up to 20 bonus points.

### Submission

The due date for this assignment is January 9th, 11.59pm. Deadline is strict, there will be no extension. It may take longer than expected, please start as soon as possible. You have to submit a zip file "your_name(s).zip" including all **commented** source codes, a **PROJECT REPORT** which contains **a project summary, your solution approach, any encountered problems and how you solved them, any unresolved issues, and a usage explanation. You should also give details of the protocol that is used by the peers and the registry.** You can submit your projects via google classroom.
Late submissions: 25% penalty after the deadline and another 25% for every 24h after the deadline.

*You can do this assignment **in groups of three students**. You are not allowed to exchange code snippets or anything across groups. Also you are not allowed to use any code from the Internet (github, etc). We will check your codes for plagiarism.*