# CSNETWK

Machine Project

## Introduction

You are tasked to develop *De La Salle Usap (DLSU)*, a one-on-one chat application. This application allows clients to send and receive messages and files. DLSU allows privacy in such a way that a conversation is shared only between two (2) clients. These conversations are handled by a messaging server.

## Requirements

### General Requirements

- The programming language to be used is Java.
- The client-side of the application must be implemented using a Graphical User Interface (GUI). The design is up to the developers as long as the requirements for this project are met.
- Threading should be implemented as well. This ensures that the server will continuously run and listen while the clients are connected. Closing the server will only be done when at least one of the clients terminate the connection on their end.
- The following features must be supported:
    - Sending of messages
    - Receiving of message
    - Sending of text and/or image files
    - Receiving of text and/or image files

### Server

- For clients to use the application, they must connect first to the messaging server.
- The server should use TCP as its transport layer protocol.
- The server does not need a graphical user interface but should display information such as the IP address and port number while it is running.
- The server should support two (2) clients.

- The activities of the clients must be logged in the server. The logs must have the following information
    - Timestamp
    - Source
    - Destination
    - Activity
- There should be an option to save the logs to a text file after the clients logged out/disconnected from the server.
- Activities include: Client logging and out, sending and receiving messages, and sending and receiving files.

### Client

- When connecting to a server, the client will be asked first for the IP address and port number of the server. This should be done on the GUI. These information must not be hardcoded. Modification of the source code is also not allowed.
- As long as the connection is open, the client can send and receive messages and files. The connection will only be terminated if the client logs out or closes their chat window. There must be an option for the user to manually terminate their connection such as a log out button or a menu option. Additionally, simply closing the window will also terminate the connection.
- When sending a file, the GUI should allow the client to locate the directory and file on their local machine.
- When receiving a file, the client should be allowed to select a directory to save the file and have the option to rename it before the actual transfer happens.
- A client should receive a notification whenever the other client is disconnected or reconnected.
- If sending a message or file to a disconnected client, this should not be allowed by the server.
- Unless both clients terminate their respective connections, a disconnected client may resume their chat with the other client by logging in again.

## Submission

- The deadline of submission is **September 21, 2020 (M), 1159PM**.
- All source files and dependencies must be submitted through the Canvas assignment page.