LU-Connect: A multithreaded chat messenger that allows real-time communication between users.

Project Objective: The term project will focus on implementing a client-server architecture using concurrent programming (multithreading) to handle multiple concurrent connections efficiently. Students will gain hands-on experience in system design and implementation for concurrency handling in a distributed, i.e., networked systems.

Project Description: **LU-Connect** application is a chat messenger that enables users to send and receive text messages in real-time. The messenger will be built using a central server that manages multiple clients concurrently, ensuring seamless communication. The project will rely on multithreading for handling multiple users, and basic security mechanisms to maintain the integrity and privacy of messages.

Key Features of LU-Connect (Functional and Quality Requirements):

User Authentication

- Users can register and log in using a unique username and password.
- Authentication details will be stored securely in a database.

Connection and Messaging Features

- The server must restrict the number of concurrent client connections to three (3). Each client will be represented as a separate thread.
- Incoming connection requests beyond the allowed limit must be put on hold until a connection slot becomes available.
- The server should use a semaphore to manage active and waiting clients efficiently.
- The server must maintain a waiting queue and display estimated wait times for each client attempting to connect.
- Users must be able to send and receive text messages concurrently without blocking.
- Messages must include timestamps for both sent and received messages.
- Each new message must trigger a notification tone for the recipient.

File Transfer Features

- Users must be able to send and receive files concurrently while messaging.
- The only allowed file types are:

  .docx (Word Documents) or .pdf (PDF Files) or .jpeg (Images)

Any other file types must be restricted and rejected by the system.

Security and Useability Features (NFRs):

- All sent and received data (text messages & files) must be encrypted before storage on the server. (NFR: Security).
- Users should have the option to mute/unmute the notification tone. (NFR: Usability)

Use the programming language of your choice to implement the system. Multithreading, Semaphore implementation and pattern application must be explicit in the source code.

You are allowed to implement additional features such as voice chat, group chat etc. These are not mandatory implementations. However, implementing them will be incentivised, if they implement concurrent programming, during project assessment.