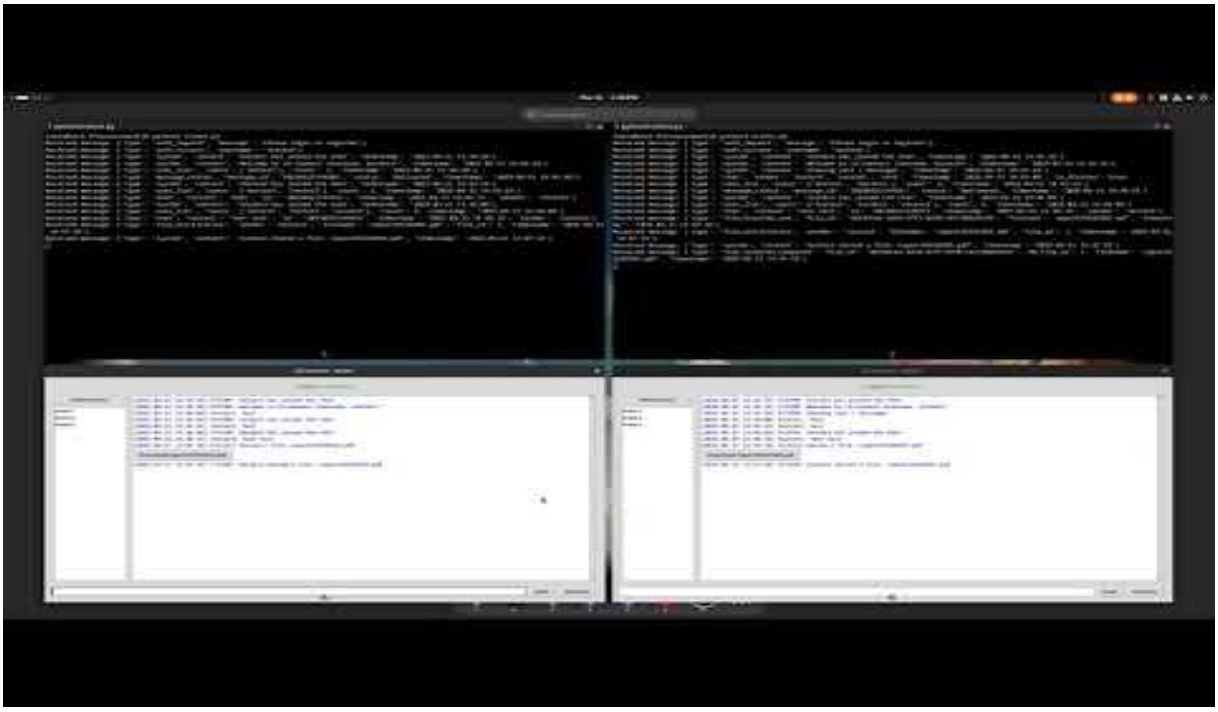
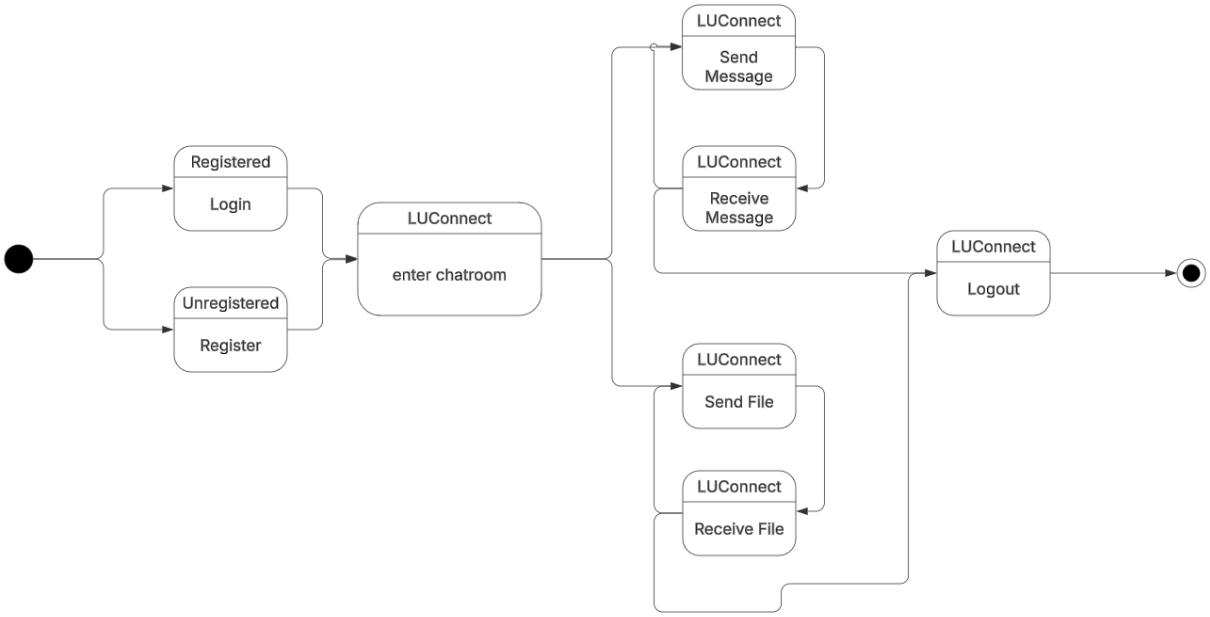
Parallel and Concurrent Systems Report

39226565

https://youtu.be/dOtUsAVtJvw



https://github.com/xnekr0/pcs-cw.git



Server:

Component	**Requester Interface**	**Provider Interface**
LUConnectServer	- Database Connection	- Socket and Client connection management

	- ClientHandler Interface	- Wait queue management - Message Broadcasting - Encryption/Decryption
AuthenticationHandler	- Database Records	- User registration - User login
DatabaseHandler	None	
ClientHandler	- Database Storage - Authentication	- Authentication Interface - Message sending/receiving - Command Processing

Client:

Component	**Requester Interface**	**Provider Interface**
LUConnectClient	- Socket Communication - Threading - JSON processing - File Operations	- Connection Management and Status Updates - Authentication - File Transfer - Receiving Messages - Sending Messages
ClientUI	- Client Logic - Input Processing	- User Interface

Pattern Used and Justification

The design pattern used was a semaphore pattern or a semaphore-based admission control pattern along with the client-server architecture. This is because the semaphore pattern is useful when trying to limit shared access to a resource, in this case only 3 clients at a time accessing the server. This is also useful for the organized wait queue system, so that rather than being denied access to the chatroom, the chatter would be in a waiting queue where they're told the estimated wait time and the amount of people in the queue, leading to a more comfortable experience.

Semaphore design pattern CodingDrills.

Available at: https://www.codingdrills.com/tutorial/design-patterns-tutorial/semaphore-pattern