**ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH**

**TRƯỜNG ĐẠI HỌC BÁCH KHOA**

**KHOA KHOA HỌC VÀ KỸ THUẬT MÁY TÍNH**

----------------------------------------

**MẠNG MÁY TÍNH**

**BÁO CÁO ASSIGNMENT 1**

**CHAT APPLICATION**

Giáo viên hướng dẫn: Bùi Xuân Giang

Thành viên nhóm :

1. Nguyễn Lê Hoàng Hiệu 1711355
2. Nguyễn Đức Quỳnh
3. Trần Văn Tâm 1713057

**TP. HCM THÁNG 11/2019**

**MỤC LỤC**

[**I. Các chức năng ứng dụng: 2**](#_Toc317)

[1. Login 2](#_Toc15244)

[2. Xem các user online 2](#_Toc17242)

[3. Chat 2](#_Toc7917)

[4. Truyền gửi file 2](#_Toc11695)

[**II. Định nghĩa các giao thức 2**](#_Toc28068)

[a) Giao thức client – server : 2](#_Toc14085)

[b) Giao thức P2P: 3](#_Toc5458)

[**III. Thiết kế chi tiết ứng dụng 4**](#_Toc2186)

[1. Công nghệ sử dụng 4](#_Toc3128)

[2. Lược đồ diagram 4](#_Toc4639)

[3. Các class chính 5](#_Toc4811)

[**IV. Hướng dẫn sử dụng 6**](#_Toc15282)

[1. Server 6](#_Toc8003)

[2. Client 6](#_Toc25428)

[**V. Đánh giá kết quả hiện thực 7**](#_Toc20609)

**VI. Tài Liệu tham khảo…………………………………………………………………8**

# Các chức năng ứng dụng:

## Login

Chức năng này cho phép client bất kì đăng nhập vào ứng dụng khi biết IP address và port của server. Và server này phải trong trạng trái Active.

## Xem các user online

Người dùng có thể xem danh sách tất cả những người đang online trên ứng dụng dưới dạng username ở cả client UI và server UI.

Thậm chí, user có thể xem được trạng thái online, offline của những bạn bè trong list danh sách của mình.

## Chat

Chức năng này cho phép 2 users ở 2 máy khác nhau tạo hội thoại riêng tư và gửi tin trực tiếp tới user khác.

Đặc biệt ,một user có thể nhắn tin trực tiếp với nhiều user trong cùng một thời điểm.

Hai user được quyền chat với nhau khi cả hai bên đều đồng ý chat.

App Chat được xây dựng theo mô hình lai giữa Client-server và P2P: hệ thống có một server trung tâm dùng cho việc đăng ký người sử dụng, quá trình chat được thực hiện trực tiếp giữa các client.

## Truyền gửi file

Chức năng này cho phép user gửi và nhận một số file với định dạng và kích thước (1MB) được quy định trong chat riêng tư với user khác khi được cho phép.

# Định nghĩa các giao thức

## Giao thức client – server :

|  |  |  |
| --- | --- | --- |
| Nội dung | Mục đích sử dụng | Ghi chú |
| <SESSION\_REG>clientInfo  </SESSION\_REG> | Yêu cầu tạo một chat session từ client. ClientInfo là thông tin của client. | Sử dụng một lần cho mỗi chat session |
| <PEER\_NAME>clientName  </PEER\_NAME> | Tạo một tên cho người dùng khi đăng nhập vào server | clientName là tên người dùng đăng kí |
| <PORT>clientPort</PORT> | Lấy port của client | clientPort là số hiệu Port của client. |
| <IP>clientIP</IP> | Lấy IP của client | ClientIP là địa chỉ IP của client |
| <SESSION\_KEEP\_ALIVE>  </SESSION\_KEEP\_ALIVE> | Nội dung yêu cầu do client gửi lên server. | Mỗi 10s, nếu client còn kết nối, thì sẽ gửi yêu lên bao gồm tên và trạng thái |
| <SESSION\_DENY /> | Từ chối client kết nối đến server | Từ chối khi người dùng đăng nhập thất bại |
| <SESSION\_ACCEPT>  </SESSION\_ACCEPT> | Chấp nhập client kết nối đến server | Chấp nhận khi người dùng đăng nhập thành công. |
| <PEER>PeerData</PEER> | Xác định thông tin của 1 client do server trả về. | Thông tin ở đây gồm name, port, IP. |

## Giao thức P2P:

Giao thức P2P được sử dụng để hai người chat với nhau. Ở mỗi client khi muốn  
chat với nhau sẽ thành lập một server.  
Ví dụ: client A muốn nói chuyện với client B thì ở client A và client B đều thành  
lập server, sau đó gửi port của các server đó cho người mình nói chuyện thông qua  
server trung tâm

|  |  |  |
| --- | --- | --- |
| Nội dung | Mục đích | Ghi chú |
| <CHAT\_REQ>  userName  </CHAT\_REQ> | Yêu cầu tạo chat riêng tư từ một user với user khác | userName là tên user tạo yêu cầu chat riêng tư |
| <CHAT\_DENY /> | Từ chối tạo chat session | Từ chối sau khi có yêu cầu tạo chat riêng tư |
| <CHAT\_ACCEPT /> | Chấp nhận tạo chat session | Chấp nhận sau khi có yêu cầu tạo chat riêng tư |
| <CHAT\_MSG>  Message  </CHAT\_MSG> | Gửi nội dung chat | Message được gửi khi user đồng ý tạo chat riêng tư |
| <PEER>  peerData  </PEER> | Xác định peer của user do server trả về | peerData là peer của user gồm name, port, IP |
| <FILE\_REQ>  fileName  </FILE\_REQ> | Yêu cầu gửi file | fileName là tên file muốn gửi |
| <FILE\_REQ\_NOACK /> | Từ chối yêu cầu gửi file | Sử dụng sau khi có yêu cầu gửi file |
| <FILE\_REQ\_ACK>  </FILE\_REQ\_ACK> | Chấp nhận yêu cầu gửi file | Sử dụng sau khi có yêu cầu gửi file |
| <FILE\_DATA\_BEGIN /> | Bắt đầu gửi file | Quá trình gửi file sau khi người gửi và nhận đồng ý |
| <FILE\_DATA>  fileData  </FILE\_DATA> | Nội dung file cần chuyển |  |
| <FILE\_DATA\_END /> | Kết thúc quá trình gửi file | Yêu cầu bởi người gửi file |
| <CHAT\_CLOSE /> | Kết thúc chat riêng tư | Sử dụng sau khi tạo chat riêng tư |

# Thiết kế chi tiết ứng dụng

## Công nghệ sử dụng

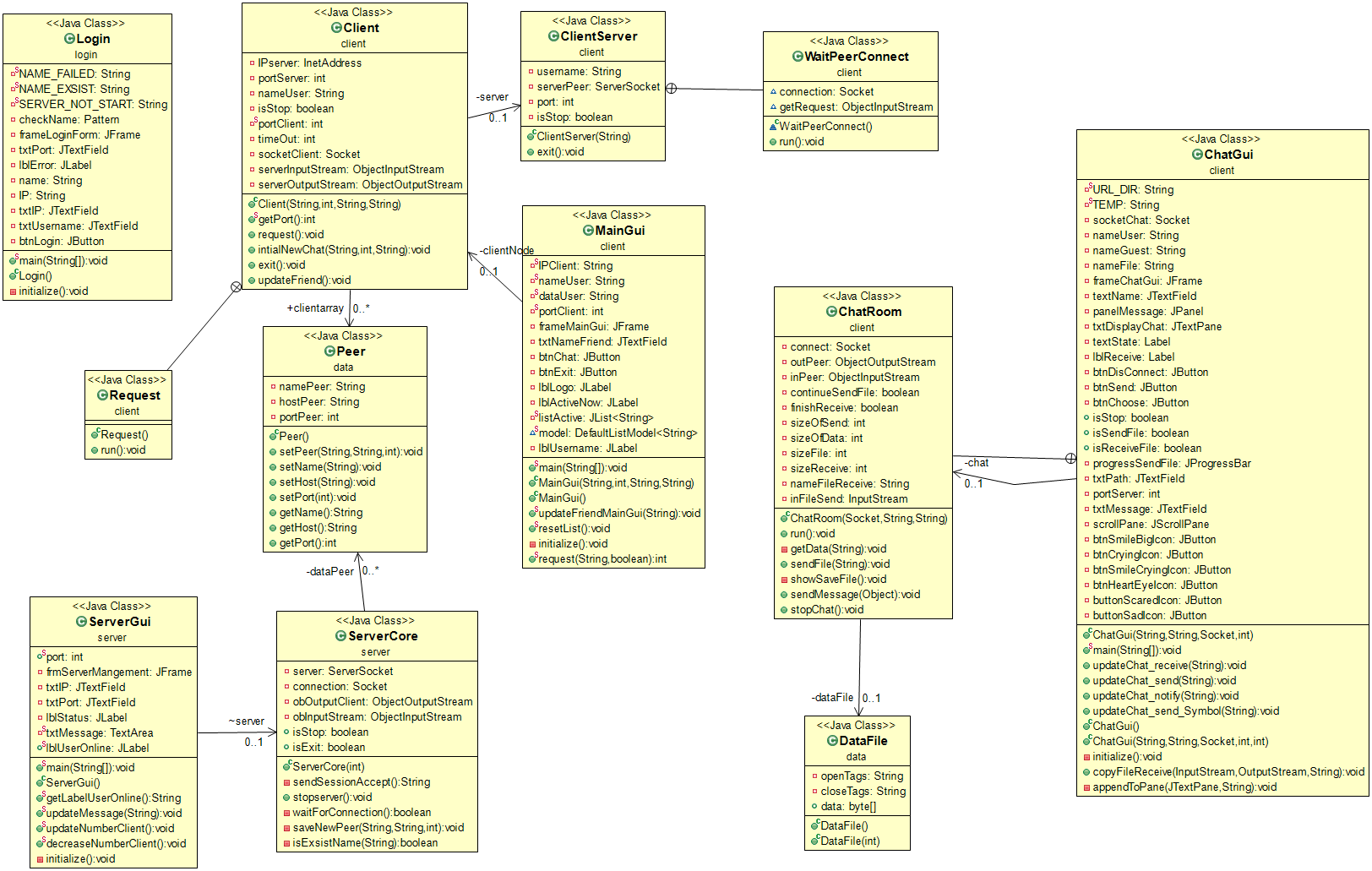
- TCP Socket: Một kĩ thuật dùng để hỗ trợ lập trình các ứng dụng giao tiếp qua mạng. TCP

Socket sử dụng Stream để thực hiện quá trình truyền dữ liệu của hai máy tính đã thiết lập

kết nối.

-Java Swing: Là một phần của Java Foundation Classes (JFC) được sử dụng để tạo các ứng dụng Window-Based.

## Lược đồ diagram



## Các class chính

**Server:**

- ServerGUI.java: sử dụng để thao tác với server như bật, tắt server. Ngoài ra còn hiển thị

các thông tin server như: IP, Port, số lượng người kết nối, các protocol mà client gửi lên.

- ServerCore.java: Đây là class để quản lí server, gồm đầy đủ thông tin, chức năng chính của server:

+ Tạo một SocketServer.

+ Quản lí danh sách các user đang online

+ Tạo một thread là WaitForConnect để xử lý các request kết nối từ user. Có thể là yêu cầu dăng nhập, thoát ứng dụng, lấy thông tin user đang online. Nếu đăng nhập thành công, trả về cho client danh sách user đang online, nếu không sẽ trả về một protocol từ chối đăng nhập.

**Client**

- MainGUI.java: GUI class để hiển thị danh sách user khác đang online, bắt sự kiện gửi yêu cầu chat của user.

- ChatGUI.java: Xử lí việc chat, gửi nhận File giữa hai user. Hiển thị nội dung chat giữa hai user.

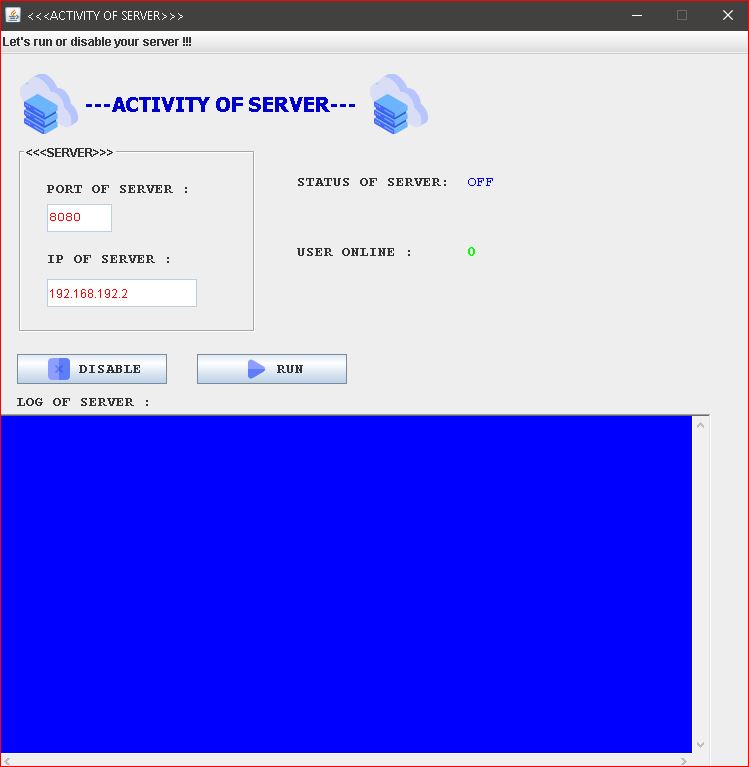
- ClientServer.java: Tạo và quản lí ServerSocket của mỗi user. Xử lí request chat gửi đến từ user khác.

- Client.java: Quản lí thông tin hiện tại của user. Nó sẽ tạo một thread để liên tục gửi request tới server để cập nhật danh sách user. Ngoài ra nó còn xử lí request chat tới user khác.

# Hướng dẫn sử dụng

## Server UI : Muốn chat với nhau, cần đăng nhập vào Server đang hoạt động trước

- Run file ServerGui.java sẽ được UI như hình dưới: (hình đã được thu lại)



**1.1 Turn on Server:**

- Click Button RUN để khởi động server, Status ban đầu là Off, sau khi RUN thì Status là RUNNING…

- Sô lượng người online, tức đã đăng nhập vào user ở Login UI với IP và Port đó sẽ được khởi tạo bằng 0; Nếu 1 ai đăng nhập vào, USER ONLINE sẽ tăng lên 1 và nếu 1 ai thoát ra thì sẽ giảm 1;

-Log of Server sẽ hiện những Protocol gửi tới Server và nó gửi đi cho các Client. Log hiển thị username đã đăng nhập; START SERVER; START ERROR; DISABLE SERVER.

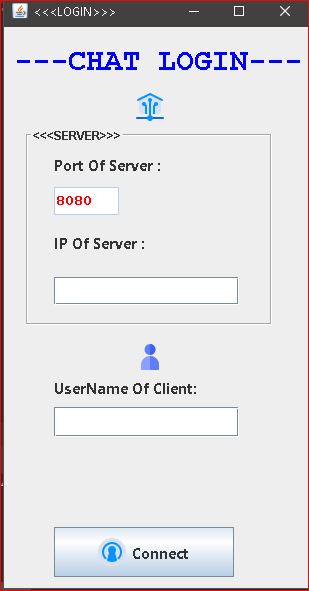
**1.2 Turn off Server:** 2 cách

- Click Button DISABLE để tắt Server, User online về 0, Status về OFF;

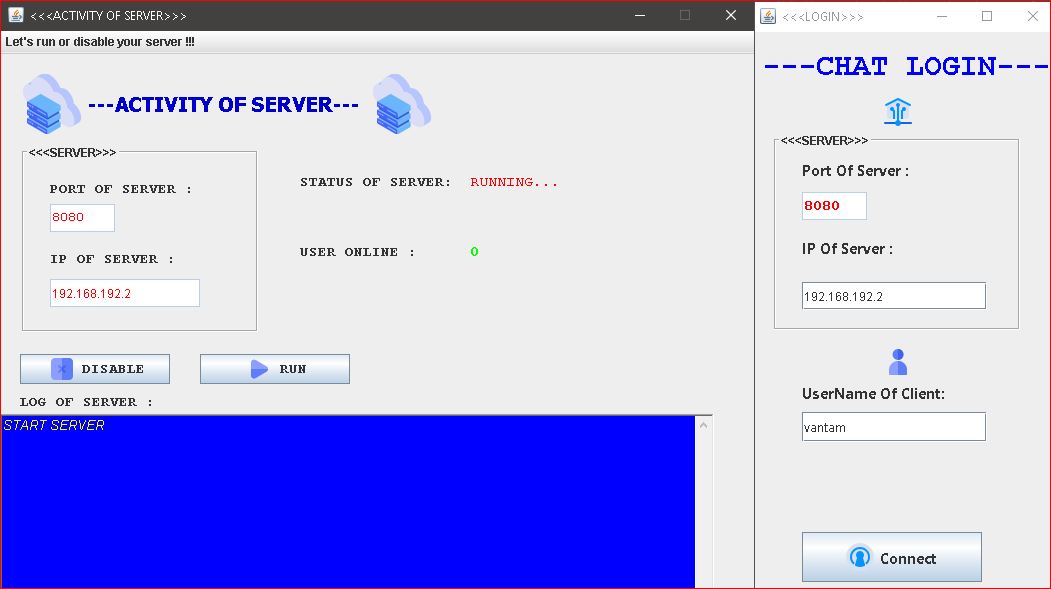
- Click vào Button “X” trên cùng bên phải window

## Client LoginUI và ListUI: Để tạo 1 client, phải đăng nhập thông tin cho ServerUI trên

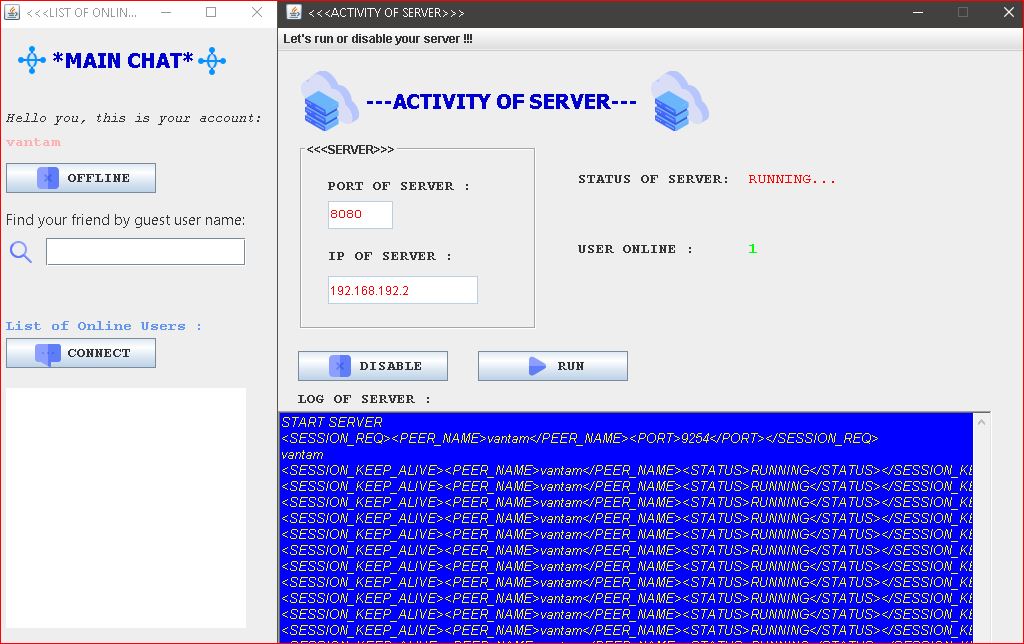
- Mở phần mềm lên có giao diện như hình: file Login.java



- Điền IP server center trùng với IP ở trên ServerUI và Username để đăng nhập(đảm bảo Server phải trong status RUNNING)



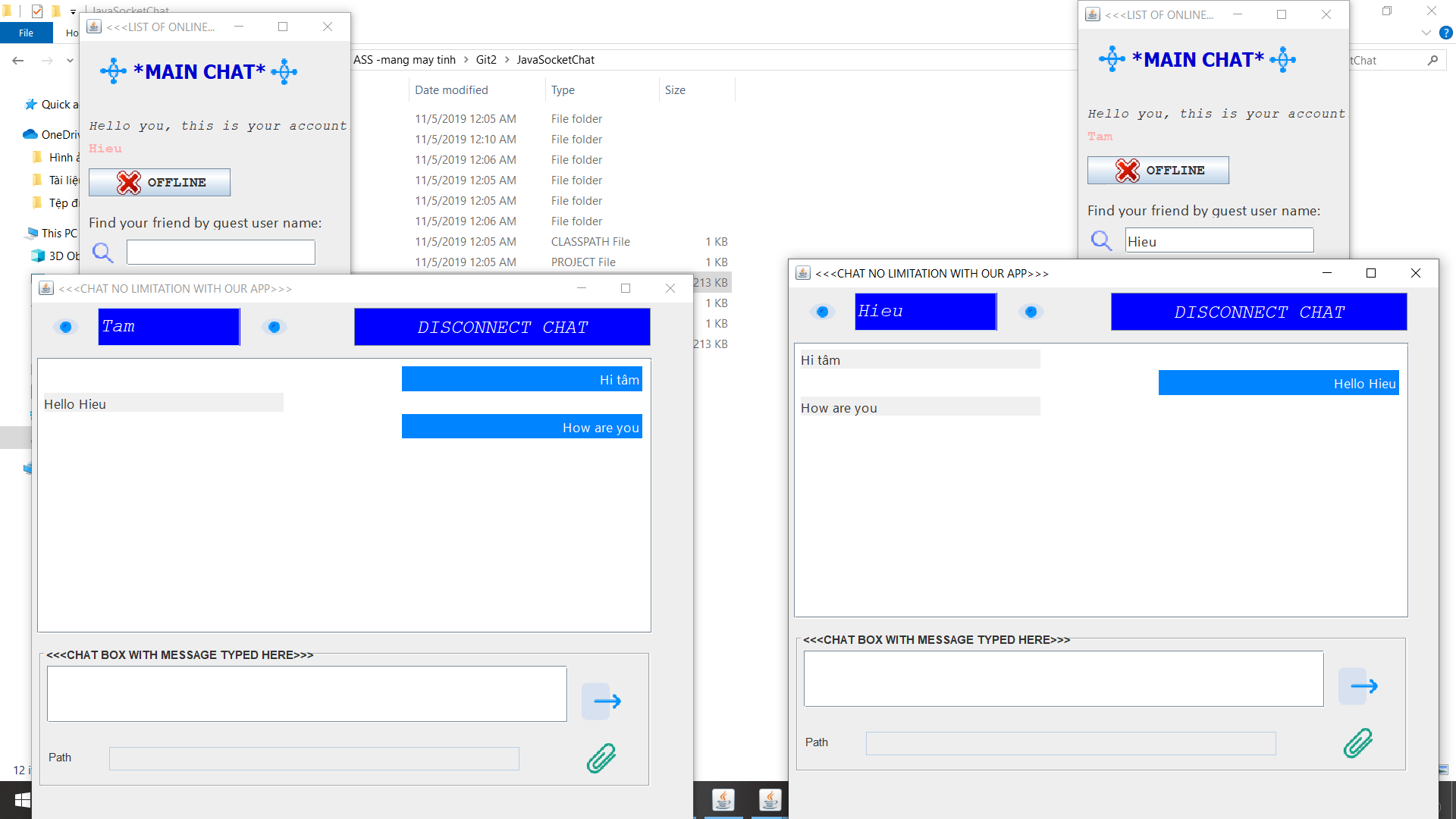
- Click Connect để kết nối server, log bên Server UI hiện tên username thì thành công



- Danh sách user online được hiển thị trong khung “List of Online User” của ListUI bởi file

- Click vào tên 1 user online sau đó chon button CONNECT để gửi yêu cầu chat đến user đó và đợi user bên đó đồng ý thì ChatUI sẽ hiện ra.

**3.ChatUI:**

Ở giao diện này ta có: 

- Tên user đang chat, khung message để nhập tin nhắn.

- Thanh Path để đính kèm file, Button  để chọn file đính kèm

- Bấm Button  send để gửi.

- Bấm button DISCONNECT để thoát khỏi khung chat.

- Bấm button Exit để thoát khỏi app.

# Đánh giá kết quả hiện thực

- Ứng dụng được xây dựng dựa trên mô hình kết hợp giữa client-server cho việc quản lí các user và P2P cho việc trò chuyện giữa hai user với nhau.

- Ứng dụng có các tính năng đơn giản như: chat giữa hai user, một lúc đồng thời chat với

nhiều user, gửi File trong quá trình chat.

# VI. Tài liệu tham khảo

Computer Networking: A Top-Down Approach (7th Edition)

<https://www.baeldung.com/a-guide-to-java-sockets>

<https://stackoverflow.com/questions/20947166/file-transfer-java-socket-programming>

Source Code: <https://github.com/toanbku/java-socket-chat>