

ĐẠI HỌC QUỐC GIA TP. HỒ CHÍ MINH  
ĐẠI HỌC BÁCH KHOA TP. HỒ CHÍ MINH  
KHOA KHOA HỌC VÀ KỸ THUẬT MÁY TÍNH



# BÁO CÁO BÀI TẬP LỚN 1

**MÔN: MẠNG MÁY TÍNH 1**

**ĐỀ BÀI: XÂY DỰNG ỨNG DỤNG CHAT**

**GVHD: LỤC ĐỨC TUẤN**

**NHÓM: 01**

**THÀNH VIÊN:**

LƯU THIÊN AN 51300015

LÊ HẢI BĂNG 51300279

NGUYỄN PHÚC CƯỜNG 51300468

TRỊNH HOÀI CHƯƠNG 51300421

NGUYỄN TUẤN ANH 51300122

*Tp. Hồ Chí Minh, tháng 10/2015*

# MỤC LỤC

---

|   |           |
|---|-----------|
| <b>1. YÊU CẦU .....</b>   | <b>2</b>  |
| <b>2. THIẾT KẾ GIAO THỨC.....</b>                                 | <b>3</b>  |
| a) Kết nối Server .....   | 3         |
| b) Đăng kí User .....   | 3         |
| c) Gửi tin nhắn .....   | 4         |
| d) Thông báo trạng thái Server .....                              | 4         |
| e) Gửi File .....   | 4         |
| <b>3. XÂY DỰNG ỨNG DỤNG.....</b>                                  | <b>5</b>  |
| a) Sever cho phần quản lí các User .....                          | 5         |
| b) Client .....   | 5         |
| <b>4. PHÂN CÔNG CÔNG VIỆC .....</b>                               | <b>7</b>  |
| a) Server - Client kết nối với Server - Client đăng ký User ..... | 7         |
| b) Chat giữa 2 User .....   | 7         |
| c) Truyền tải File giữa 2 User .....                              | 7         |
| <b>5. HƯỚNG DẪN SỬ DỤNG .....</b>                                 | <b>8</b>  |
| a) Server .....   | 8         |
| b) Client .....   | 10        |
| <b>6. HẠN CHẾ – THIẾU SÓT – LỖI.....</b>                          | <b>13</b> |
| <b>7. TÀI LIỆU THAM KHẢO.....</b>                                 | <b>13</b> |

## 1. YÊU CẦU ỨNG DỤNG

---

- Ứng dụng cho phép 2 người ở 2 máy khác nhau có thể chat với nhau
- Một người có thể chat với nhiều người khác.
- Ứng dụng chat có được xây dựng theo mô hình P2P: vừa có chức năng chat server, đồng thời cũng có chức năng chat client.
- Ứng dụng cho phép truyền tải file trong quá trình chat giữa hai người.
- Cùng với ứng dụng chat theo mô hình P2P, hệ thống còn có một server trung tâm dùng cho việc đăng ký người sử dụng và quản lý danh mục người sử dụng đang online.
- Ứng dụng phải xây dựng theo giao thức **được định nghĩa bởi nhóm thực hành.**

## 2. THIẾT KẾ GIAO THỨC:

### a) Kết nối đến Server:

```
<CONNECT>
  <IP> IP address Server </IP>
  <PORT> Port </PORT>
</CONNECT>
```

### b) Đăng kí User:

```
<REGISTER>
  <PEER_NAME> Username </PEER_NAME>
  <IP> IP Address </IP>
  <PORT> Port number </PORT>
</REGISTER>
```

Lưu ý : Port number này dùng để tạo Server nhận các kết nối từ các User khác để chat

Sau đó, Server sẽ dò tìm danh sách User để xem có bị trùng tên, nếu không bị trùng tên, Server gửi cho User thông báo :

```
<USER_VALID>
  <PEER_NAME> Username </PEER_NAME>
  <SUCCESS> true </SUCCESS>
</USER_VALID>
```

+ Server gửi cho User đó list các User đang online:

```
<LIST_ONLINE>
  <PEER>
    <PEER_NAME> Username 1</PEER_NAME>
    <IP>IP address 1</IP>
    <PORT>port number 1</PORT>
  </PEER>
  <PEER>
    <PEER_NAME>Username 2</PEER_NAME>
    <IP>IP address 2</IP>
    <PORT>port number 2</PORT>
  </PEER>
  ...
  <PEER>
    <PEER_NAME>Username n</PEER_NAME>
    <IP>IP address n</IP>
    <PORT> port number n</PORT>
  </PEER>
</LIST_ONLINE>
```

+ Đồng thời Server gửi cho các User đang online thông tin User vừa mới đăng nhập vào

```
<USER_ONLINE>
  <PEER_NAME> Username </PEER_NAME>
  <IP> IP address </IP>
  <PORT> port number </PORT>
</USER_ONLINE>
```

Nếu user đăng nhập bị trùng tên, Server gửi cho User thông báo :

```
<USER_EXIST>
  <PEER_NAME> Username </PEER_NAME>
  <SUCCESS> false </SUCCESS>
```

```
</USER_EXIST>
```

### c) Gửi tin nhắn:

User\_Sender sẽ tạo ra 1 socket kết nối vào Server của User\_receiver

User\_Sender gửi cho User\_Receiver thông tin:

```
<MESSAGE>
  <PEER_NAME> Peer sender </PEER_NAME>
  <CHAT> Content </CHAT>
</MESSAGE>
```

### d) Thông báo trạng thái cho Server:

d.1). Thông báo Server tình trạng online:

Định kỳ sau 15s, Client gửi cập nhật trạng thái cho biết mình vẫn còn online

```
<KEEP_ONLINE>
  <STATUS> ONLINE </STATUS>
</KEEP_ONLINE>
```

d.2). Thông báo Server tình trạng offline:

Trước khi client offline, client sẽ thông báo cho server theo định dạng sau

```
<KEEP_ONLINE>
  <STATUS> OFFLINE </STATUS>
</KEEP_ONLINE>
```

### e) Gửi file:

User\_Sender gửi một yêu cầu gửi file tới User\_Receiver

```
<FILE_REQ>
  <PEER_NAME> Peer sender </PEER_NAME>
  <FILE_NAME> filename </FILE_NAME>
</FILE_REQ>
```

User\_Receiver gửi lại User\_Sender thông báo đồng ý tải file:

+ Nếu đồng ý, User\_Receiver sẽ tạo ra 1 server với 1 port được cấp tự động (*new ServerSocket(0)*) và gửi cho User\_Sender

```
<FILE_RES>
  <PORT> Port number </PORT>
</FILE_RES>
```

+ Nếu không đồng ý : <FILE\_RES> false </FILE\_RES>

Nếu User\_Receiver đồng ý trao đổi file, User\_Sender bắt đầu gửi file:

```
<FILE_DATA> data </FILE_DATA>
```

Trường hợp file lớn, file có thể cắt ra thành các thành phần nhỏ rồi chuyển đi

Độ lớn tối đa của file: <MAX\_FILE\_SIZE> 1024 byte

Kết thúc quá trình chuyển file: <FILE\_END>

### 3. XÂY DỰNG ỨNG DỤNG:

---

#### a) Server cho phần quản lý các User:

Bao gồm các Class:

+ ServerGUI: giao diện của Server

+ Server:

- Tạo Server
- Lắng nghe kết nối từ Client
- Xử lý các Message mà Client gửi đến.
- Quản lý các Thread SocketServerThread bằng một mảng *SocketServerThread client[]*

+ SocketServerThread:

- Là một thread dùng để quản lý từng User kết nối tới Server. Khi 1 Client kết nối đến Server, Server sẽ tạo ra Thread này để tạo ra các kết nối nhận và gửi đến Client
- Nhận và gửi Message đến Client.
- Lưu Username, địa chỉ IP và Port của User

+ Message : Giao thức truyền tải thông tin

#### b) Client

- Class Message : Giao thức truyền tải thông tin

- Client kết nối đến Server :

+ Class ClientConnectGUI : Giao diện

+ Class PeerOnline : dùng để lưu thông tin Username, IP, Port của các User đang Online

+ Class Client :

- Tạo client
- Tạo luồng gửi và nhận kết nối đến Server.
- Nhận và gửi thông tin đến Server
- Xử lý các thông tin đến từ Server:
  - Thông báo đăng ký thành công
  - Nhận list User online từ Client và lưu lại vào mảng *PeerOnline listUser[]*

- Client chat:

+ Class ChatRoomGUI: Giao diện Chat room, hiển thị list các User đang online

+ Class ServerChat:

- Tạo ra một Server để lắng nghe các User khác kết nối vào
- Tạo ra các Thread để quản lý các User kết nối vào

+ Class SocketServerChatThread:

- Tạo luồng nhận và gửi đến ClientChat
- Mỗi SocketServerChatThread sẽ xử lý thông tin mà ClientChat của User\_Sender gửi đến
- Nhận tin nhắn từ User\_Sender
- Nhận yêu cầu tải file từ User\_Sender, phản hồi lại User\_Sender. Nếu đồng ý thì chọn nơi lưu file và download file từ User\_Sender

+ Class ClientChat:

- Kết nối đến ServerChat của User khác.
- Chỉ được gửi tin nhắn chat

- Gửi yêu cầu tải file, nhận phản hồi từ User\_Receiver. Nếu đồng ý tải file thì bắt đầu gửi file
  - Mỗi ChatMessenger sẽ có 1 ClientChat tương ứng để gửi tin nhắn đến User\_Receiver
- + Class ChatMessenger:
- Giao diện để chat
  - Lưu thông tin của User\_Receiver: *PeerOnline receiver*.
  - Tạo ra một ClientChat để gửi tin nhắn
  - Chọn file để gửi
- Gửi file:
- + Class Uploadfile: Tạo Socket liên kết với Server bên phía gửi và gửi file qua User\_Receiver.
- + Class Downloadfile: Tạo ra Server với port tự động. Nhiệm vụ của Server này là nhận file. Sau khi nhận xong, Server tự động đóng

## 4. PHÂN CÔNG CÔNG VIỆC :

### a) Server – Client kết nối với Server – Client đăng ký User :

Lưu Thiện An

Bao gồm các Class :

- Server :
  - Package Server
    - Server
    - SocketServerThread
    - ServerGUI
  - Package Protocol
    - Message
- Client:
  - Package Client
    - Client
    - ClientConnectGUI
    - PeerOnline
    - ChatRoomGUI

### b) Chat giữa 2 User:

Nguyễn Phúc Cường

Trịnh Hoài Chương

Lưu Thiện An (chỉnh sửa – tổng hợp)

Bao gồm các Class:

- Package Chat
  - ChatMessenger
  - ClientChat
  - ServerChat
  - SocketServerChatThread
- Package Client
  - ChatRoomGUI

### c) Truyền tải file giữa 2 User:

Lê Hải Bằng

Nguyễn Tuấn Anh

Lưu Thiện An (chỉnh sửa – tổng hợp)

Bao Gồm các Class

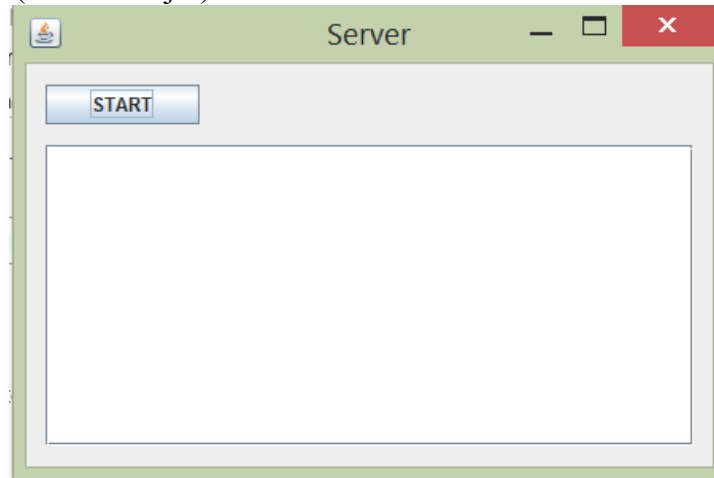
- Package File
  - UploadFile
  - DownloadFile
- Package Chat
  - ChatMessenger



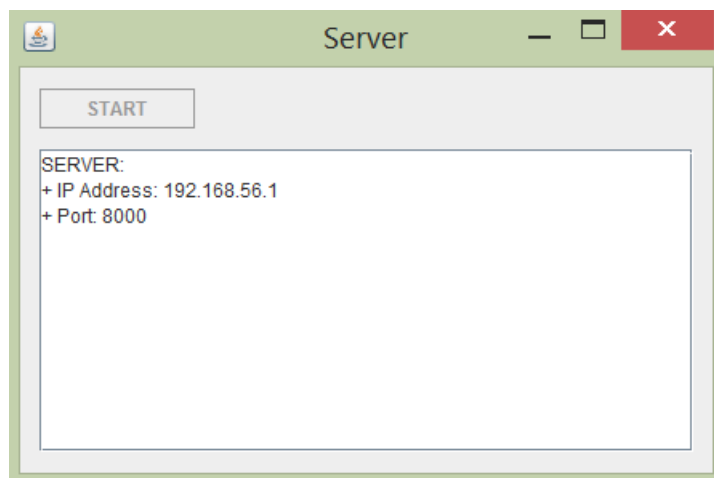
## 5. HƯỚNG DẪN SỬ DỤNG:

### a) Server:

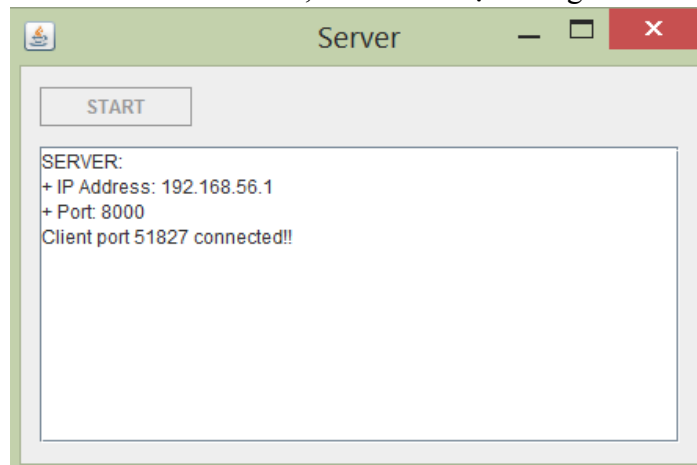
- Chạy Server trước (file Server.jar)



- Click button Start để chạy Server



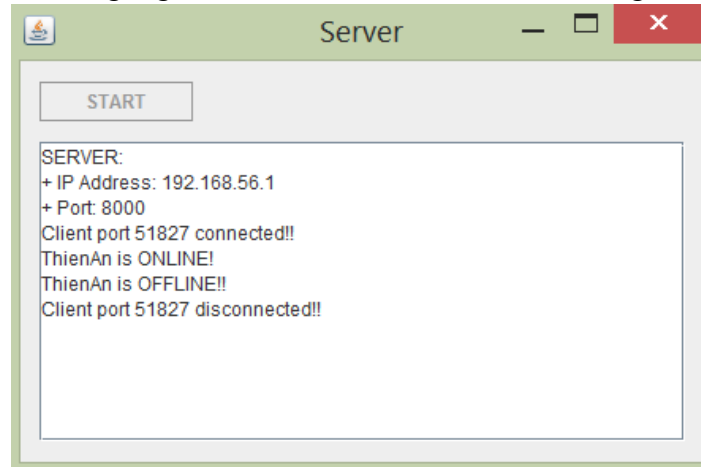
- Server đang lắng nghe các kết nối từ Client, mặc định port là 8000
- Khi có một Client kết nối vào Server, Server sẽ hiện thông báo.



- Khi user đăng nhập vào, Server hiện thông báo



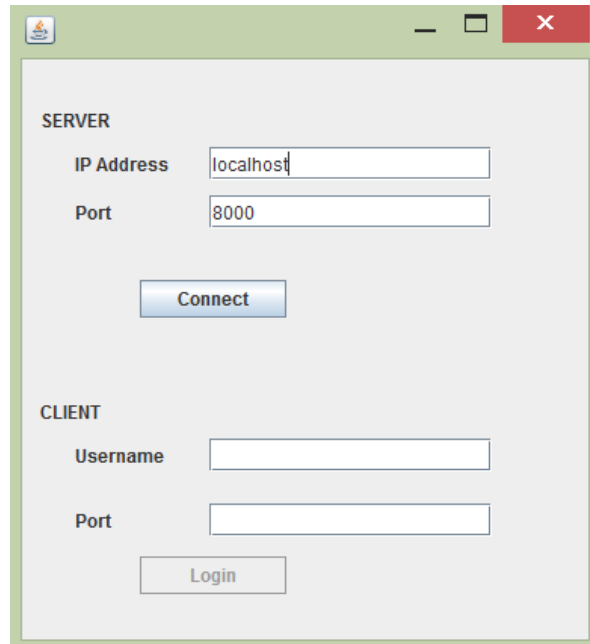
- Khi User offline và ngưng kết nối tới Server, Server hiện thông báo



**b) Client:**

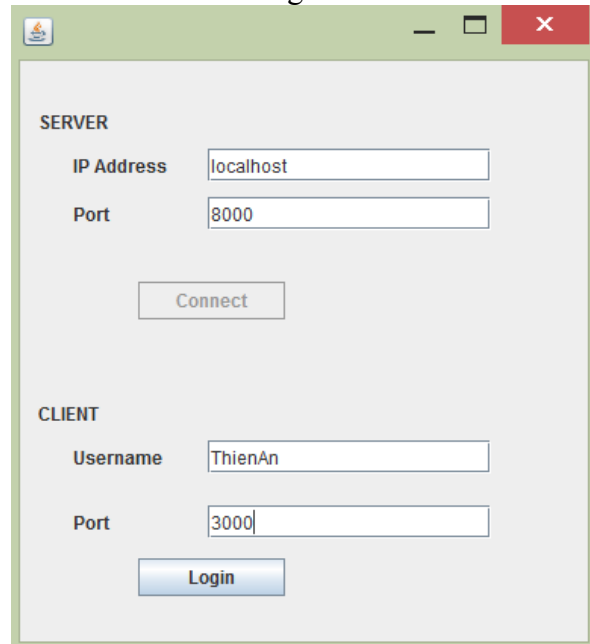
Chạy file Client.jar

Kết nối đến Server: Nhập địa chỉ IP và Port của Server (ở đây mặc định là localhost và port 8000) rồi click vào button Connect.



The screenshot shows a Java Swing window titled 'Client'. It contains two sections: 'SERVER' and 'CLIENT'. In the 'SERVER' section, the 'IP Address' field is filled with 'localhost' and the 'Port' field is filled with '8000'. Below these fields is a blue 'Connect' button. In the 'CLIENT' section, the 'Username' and 'Port' fields are empty. Below these fields is a grey 'Login' button.

Đăng ký làm User để chat: nhập vào Username và Port (Port này dùng để tạo Server nhận tin nhắn từ các User khác) rồi click vào button Login

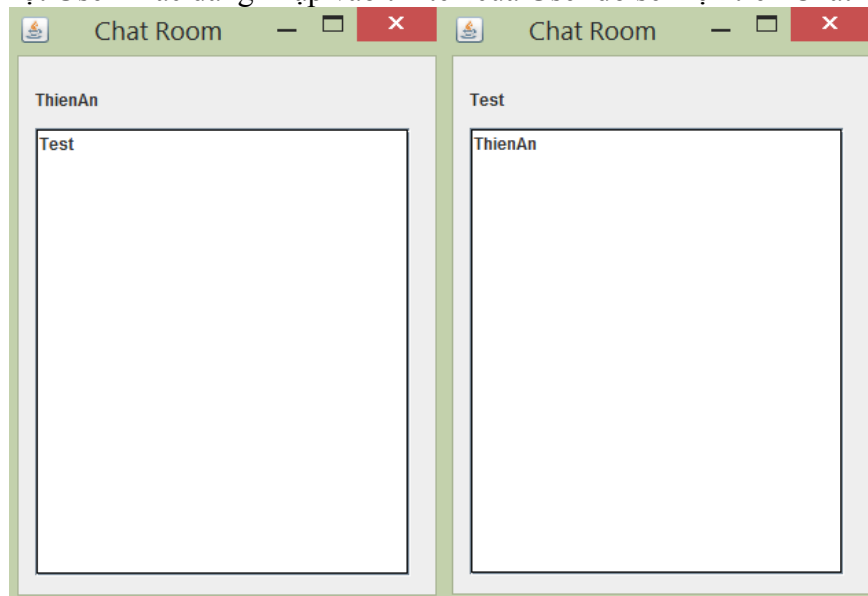


The screenshot shows the same 'Client' window. The 'SERVER' section is unchanged. In the 'CLIENT' section, the 'Username' field now contains 'ThienAn' and the 'Port' field contains '3000'. The 'Login' button is now highlighted in blue.

Login thành công sẽ chuyển đến khung Chat Room

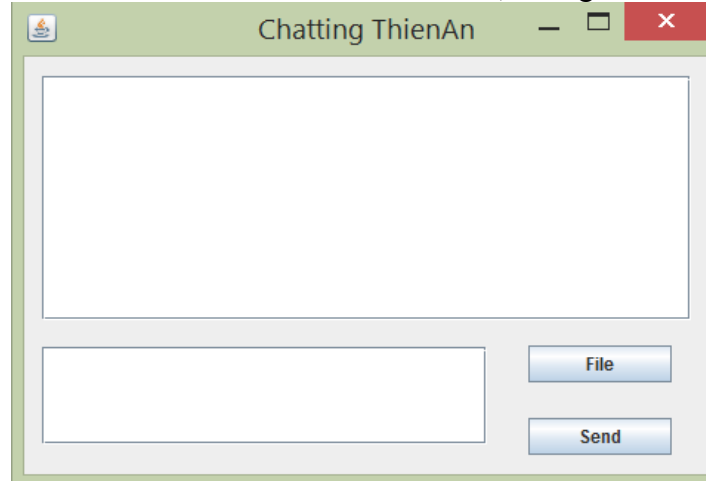


Lúc này, do chỉ có mình User ThienAn nên Chat Room sẽ trống  
Khi có thêm một User khác đăng nhập vào thì tên của User đó sẽ hiện trên Chat Room



Ví dụ ở đây ta có 2 User Test và ThienAn

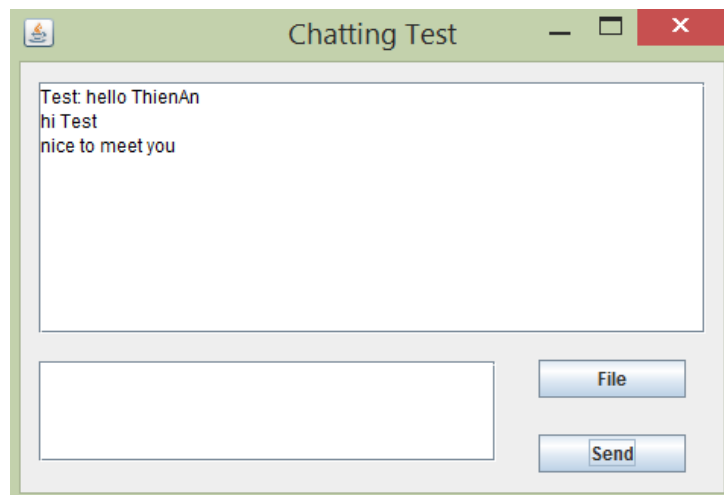
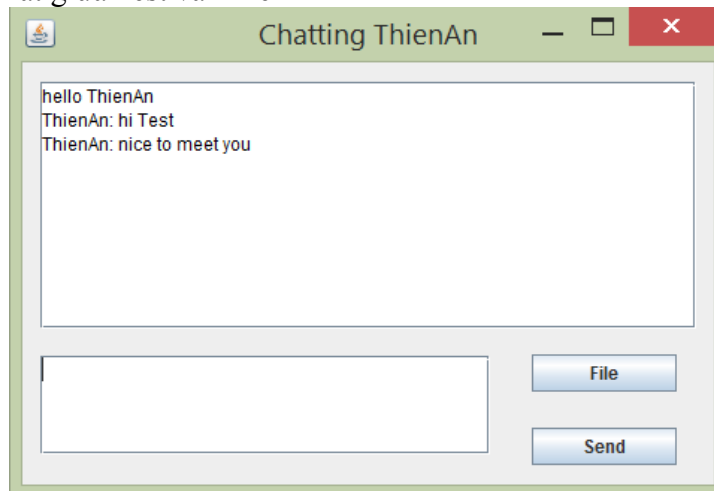
Muốn Chat với User nào thì Double Click vào tên User đó, khung Chat sẽ hiện ra



Button Send: Gửi tin nhắn

Button File: Chọn file và gửi file

Thử một vài đoạn chat giữa Test và ThienAn



## 6. HẠN CHẾ – THIẾU SÓT – LỖI

---

Trong quá trình thực hiện, do nhóm bị vướng phải một số lỗi nên thời gian làm kéo dài hơn dự kiến. Vì thế, một số chức năng nhóm em chưa thể thực hiện:

- Đăng nhập bằng username – password
- Lưu lịch sử tin nhắn
- Sử dụng Timeout để quản lý việc User mất kết nối đột ngột
- Giao diện không được đẹp

## 7. TÀI LIỆU THAM KHẢO:

---

Code có tham khảo trên một số trang web sau:

<http://ksec.info/threads/xay-dung-ung-dung-chat-bang-java.586/>  
[http://www.tutorialspoint.com/java/java\\_multithreading.htm](http://www.tutorialspoint.com/java/java_multithreading.htm)  
[http://www.tutorialspoint.com/java/java\\_networking.htm](http://www.tutorialspoint.com/java/java_networking.htm)  
<http://fahdshariff.blogspot.com/2012/10/java-find-available-port-number.html>  
<http://www.javaworld.com/article/2076864/java-concurrency/building-an-internet-chat-system.html>  
<http://mrbool.com/file-transfer-between-2-computers-with-java/24516>  
<http://www.java2s.com/Code/Java/Network-Protocol/FindsalocalnonloopbackIPv4address.htm>