

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

Bài tập lớn số 01

XÂY DỰNG ỨNG DỤNG CHAT  
THEO PROTOCOL DO NHÓM TỰ ĐỊNH NGHĨA

Giáo viên hướng dẫn:	Nguyễn Hồng Nam	
Sinh viên:	Nguyễn Thanh Tuấn	1613907
	Nguyễn Văn Sơn	1612980
	Trần Nhân Vũ	1614179
	Nguyễn Xuân Hiến	1652192

TP. HỒ CHÍ MINH, THÁNG 9/2018



## Mục lục

<b>I</b>	<b>Giai đoạn 1</b>	<b>3</b>
1	Nhiệm Vụ	4
2	Yêu Cầu	4
3	Tỉ lệ điểm	4
4	Định nghĩa và miêu tả các chức năng của ứng dụng	4
4.1	Đăng ký tài khoản	4
4.2	Đăng nhập	4
4.3	Gửi nhận tin nhắn	5
4.4	Gửi nhận tệp	5
4.5	Chat với nhiều liên hệ	5
4.6	Xem liên hệ hoạt động	5
4.7	Kết bạn với người dùng khác	5
4.8	Quản lý danh sách người dùng	5
5	Định nghĩa các giao thức sử dụng cho từng chức năng	5
<b>II</b>	<b>Giai đoạn 2:</b>	<b>8</b>
6	Miêu tả các chức năng thực tế của ứng dụng mà nhóm đã hiện thực	8
6.1	Các chức năng với vai trò người dùng	8
6.1.1	Đăng ký tài khoản	8
6.1.2	Đăng nhập	9
6.1.3	Gửi tin nhắn	10
6.1.4	Gửi tệp	10
6.1.5	Nhận tệp	11
6.1.6	Xem liên hệ đang hoạt động	12
6.2	Các chức năng với vai trò người quản lý	12
6.2.1	Tạo server phục vụ người dùng	12
6.2.2	Quản lý người dùng đang hoạt động và người dùng đã bị vô hiệu hóa tài khoản	13
7	Thiết kế chi tiết ứng dụng	13
7.0.1	SocketServer	13
7.0.2	Message	13
7.0.3	Database	13
7.0.4	SignUp	13
7.0.5	SocketClient	14
7.0.6	Message	14
7.0.7	Download	14
7.0.8	Upload	14
7.0.9	SignUp	14
8	Đánh giá kết quả hiện thực	14



<b>9 Các chức năng mở rộng của hệ thống ngoài các yêu cầu được qui định</b>	<b>14</b>
<b>10 Hướng dẫn sử dụng ứng dụng</b>	<b>14</b>
10.0.1 Hướng dẫn đăng ký tài khoản . . . . .	14
10.0.2 Hướng dẫn đăng nhập . . . . .	15
10.0.3 Hướng dẫn xem liên hệ đang hoạt động . . . . .	16
10.0.4 Hướng dẫn gửi tin nhắn . . . . .	16
10.0.5 Hướng dẫn gửi tệp tin . . . . .	17
10.0.6 Hướng dẫn nhận tệp tin . . . . .	18
10.0.7 Hướng dẫn tạo server chat . . . . .	18
<b>11 Mã nguồn chương trình (softcopy)</b>	<b>19</b>
<b>12 Ứng dụng chat (softcopy) được biên dịch từ mã nguồn</b>	<b>19</b>
<b>Tài liệu tham khảo</b>	<b>19</b>



## Phần I

# Giai đoạn 1

## Mở đầu

Báo cáo này đặc tả giao diện và giao thức truyền tin của ứng dụng cho Bài tập lớn số 1: Xây dựng ứng dụng chat trên mô hình Hybrid.

## 1 Nhiệm Vụ

Xây dựng ứng dụng chat theo protocol do nhóm tự định nghĩa, sử dụng giao thức TCP/IP.

## 2 Yêu Cầu

- Ứng dụng cho phép hai người sử dụng ở hai máy khác nhau có thể chat với nhau.
- Một người có thể chat với nhiều người khác nhau tại cùng một thời điểm.
- Ứng dụng 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 và quản lý danh mục người sử dụng đang online, quá trình chat được thực hiện trực tiếp giữa các client.
- Ứng dụng cho phép truyền tải file trong quá trình chat giữa hai người.
- Ứng dụng phải sử dụng Chat protocol đã được nhóm định nghĩa trước đó.

## 3 Tỷ lệ điểm

MSSV	Tên sinh viên	Tỷ lệ điểm
1613907	Nguyễn Thanh Tuấn	100 %
1612980	Nguyễn Văn Sơn	100 %
1614179	Trần Nhân Vũ	100 %
1652192	Nguyễn Xuân Hiến	100 %

## 4 Định nghĩa và miêu tả các chức năng của ứng dụng

### 4.1 Đăng ký tài khoản

- Ứng dụng cho phép người dùng đăng ký sử dụng. Thông tin đăng ký bao gồm tên tài khoản, tên người dùng và mật khẩu và nhập mật khẩu lần hai.
- Một người dùng có thể có một hoặc nhiều tài khoản.
- 

### 4.2 Đăng nhập

- Người sử dụng ứng dụng chat cần đăng nhập tài khoản của mình để xác minh danh tính và bắt đầu sử dụng ứng dụng. Tài khoản sẽ lưu kèm theo lịch sử chat, danh sách các bạn.
- Khi người dùng bắt đầu đăng nhập vào hệ thống, máy chủ kiểm tra thông tin đăng nhập. Nếu thông tin không hợp lệ thì báo lỗi tới người dùng. Nếu thông tin hợp lệ thì mở giao diện của phần mềm chat.

### 4.3 Gửi nhận tin nhắn

- Khi đang trong một cửa sổ chat với một người khác, người sử dụng có thể dùng bàn phím nhập thông tin dạng chữ để gửi tin nhắn cho người nhận.
- Cấu trúc tin nhắn hiển thị trên khung trò chuyện:

Tên người gửi + ">" + Tên người nhận + ":" + Nội dung tin nhắn

Ví dụ:

Taylor > Justin : How are you today, Justin?

Justin > Taylor: I'm good. Thanks. How about you?

- Việc gửi tin nhắn không thể hoàn tác.

### 4.4 Gửi nhận tệp

- Khi đang trong một cửa sổ chat với một người khác, người sử dụng có thể gửi một file có dung lượng giới hạn tới người nhận.
- Ứng dụng cho phép hai người dùng ở hai máy khác nhau có thể trao đổi tệp tin với nhau thông qua chức năng gửi tệp tin.
- Một người dùng có thể gửi tệp tin tới một người hoặc nhiều người dùng một lúc.
- Để gửi tệp tin đi, người dùng chọn vào mục đường dẫn để mở tệp tin được lưu trong máy tính, sau đó bấm nút gửi.
- Việc gửi tệp tin không thể hoàn tác.

### 4.5 Chat với nhiều liên hệ

- Người sử dụng có thể mở nhiều cửa sổ tương ứng với nhiều liên hệ khác nhau và có thể chat với nhiều người khác nhau tại cùng một thời điểm.

### 4.6 Xem liên hệ hoạt động

- Ứng dụng cho phép người dùng xem danh sách các người dùng khác đang hoạt động.

### 4.7 Kết bạn với người dùng khác

- Một người dùng có thể gửi lời mời kết bạn tới một hoặc nhiều người dùng khác. Nếu được chấp nhận hai người sẽ trở thành bạn bè và thấy tính trạng online của đối phương.

### 4.8 Quản lý danh sách người dùng

- Ứng dụng cho phép máy chủ quản lý danh sách tất cả người dùng đã đăng ký.
- Ứng dụng cho phép quản lý riêng danh sách người dùng đang hoạt động và người dùng không hoạt động.

## 5 Định nghĩa các giao thức sử dụng cho từng chức năng

Ứng dụng 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 và quản lý danh mục người sử dụng đang online, quá trình chat được thực hiện trực tiếp giữa các client.



Nội dung protocol	Mục đích sử dụng	Hướng sử dụng	Ghi chú
<SESSION_REQ> ClientID </SESSION_REQ>	Yêu cầu khởi tạo một chat session từ phía client. Client ID là id của người sử dụng phía client được yêu cầu	Client → Server	Được sử dụng một lần cho mỗi session.
<SESSION_NOACK/>	Từ chối yêu cầu chat từ một Client nào đó	Server → Client	Được sử dụng sau khi nhận yêu cầu chat từ phía client
<CHAT_MESS> Chat mes- senge </CHAT_MESS>	Gửi một chat messenge	Client ↔ Server & P2P	Được sử dụng bởi Client Chat mes- senge chỉ được gửi đi sau khi hai client đồng ý tạo kết nối. Tin nhắn này cũng sẽ được gửi lên server để lưu trữ.
<FILE_REQ> FileName </FILE_REQ>	Gửi một yêu cầu truyền file trong quá trình chat	Client → Server	Yêu cầu phải được cả hai chấp nhận. File này cũng sẽ được server tải lên để lưu trữ.
<FILE_NOACK_SER>	Server từ chối quá trình chuyển file	Server → Client	
<FILE_NOACK_CLI>	Client từ chối quá trình chuyển file	P2P	
<FILE_ACK_CLI>	Client chấp nhận quá trình chuyển file	P2P	
<FILE_BEGIN>	Bắt đầu chuyển file	Client ↔ Server & P2P	
<FILE_END>	Kết thúc quá trình chuyển file	Client ↔ Server & P2P	
<FILE_DATA> File data unit in binary </FILE_DATA>	Nội dung file cần chuyển. Trong trường hợp file lớn, file có thể cắt ra thành nhiều đoạn nhỏ và chuyển từng đoạn.	Client ↔ Server & P2P	
<SESSION_CLOSE/>	Đóng chat session.	P2P	
MAX_MSG_SIZE	1024 byte		Kích thước tối đa của một messenge, áp dụng cho nội dung chat và dữ liệu trong quá trình chuyển file.



Nội dung protocol	Mục đích sử dụng	Hướng sử dụng	Ghi chú
<LOGIN_REQ><USER_NAME> UserName </USER_NAME><PASSWORD> Password </PASSWORD></LOGIN>	Gửi yêu cầu đăng nhập vào tài khoản với user name và Password	Client → Server	
<LOGIN_NOACK>	Từ chối đăng nhập	Server → Client	
<LOGIN_ACK>	Được phép đăng nhập	Server → Client	
<LOGUP_REQ><USER_NAME> UserName </USER_NAME><PASSWORD> Password </PASSWORD></LOGUP_REQ>	Gửi yêu cầu tạo tài khoản với user name và Password	Client → Server	
<FRIEND_LIST_BEGIN>	Bắt đầu truyền danh sách bạn bè.	Server → Client	
<FRIEND_DATA> FriendName </FRIEND_DATA>	Một bạn bè được gửi cho Client.	Server → Client	
<FRIEND_LIST_END>	Kết thúc truyền danh sách bạn bè.	Server → Client	
<ONLINE_STATUS_REQ> name </ONLINE_STATUS_REQ>	Yêu cầu kiểm tra tình trạng online của một ID	Client → Server	
<ONLINE_STATUS_ACK> name </ONLINE_STATUS_ACK>	Trả về tình trạng online của một ID	Client → Server	
<ADD_FRIEND_REQ> FriendName </ADD_FRIEND_REQ>	Yêu cầu kết bạn với tài khoản khác	Client → Server	
<ADD_FRIEND_ACK> FriendName </ADD_FRIEND_ACK>	Yêu cầu kết bạn được chấp nhận	Server → Client	
<ADD_FRIEND_NOACK> FriendName </ADD_FRIEND_NOACK>	Yêu cầu kết bạn bị từ chối	Server → Client	



## Phần II

# Giai đoạn 2:

## 6 Miêu tả các chức năng thực tế của ứng dụng mà nhóm đã hiện thực

### 6.1 Các chức năng với vai trò người dùng

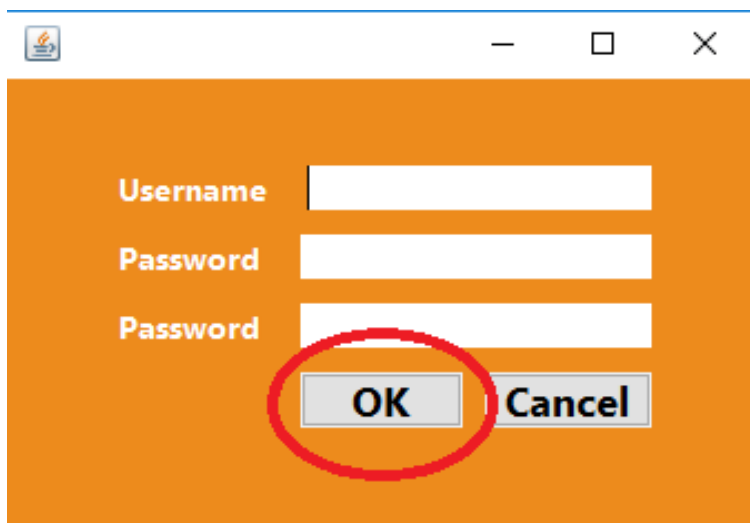
#### 6.1.1 Đăng ký tài khoản

- Để đăng ký tài khoản, người dùng mở ứng dụng, sau đó nhấn chọn mục SignUp và điền thông tin hợp lệ.



Hình 1: Giao diện người dùng

- Sau đó điền tên người dùng và mật khẩu vào giao diện SignUp.

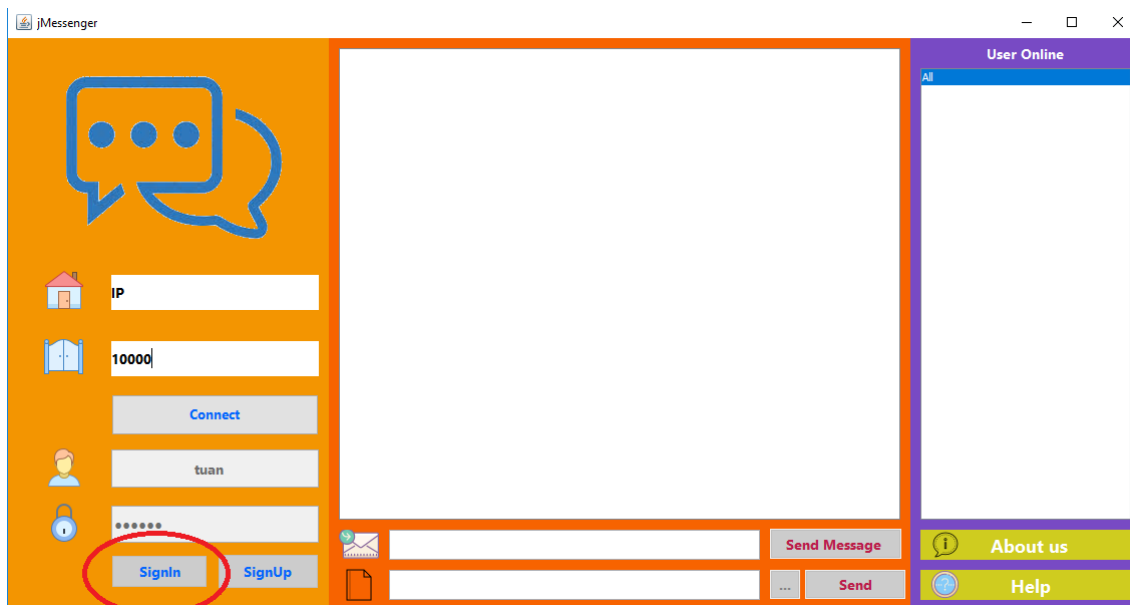


A screenshot of a login dialog box with an orange background. It contains three input fields: 'Username', 'Password', and 'Password'. Below the fields are two buttons: 'OK' and 'Cancel'. The 'OK' button is circled in red.

Hình 2: Giao diện đăng ký

### 6.1.2 Đăng nhập

- Để đăng nhập, người dùng mở ứng dụng, sau đó nhấn chọn mục SignIn và nhập thông tin hợp lệ và hai mục username và password.

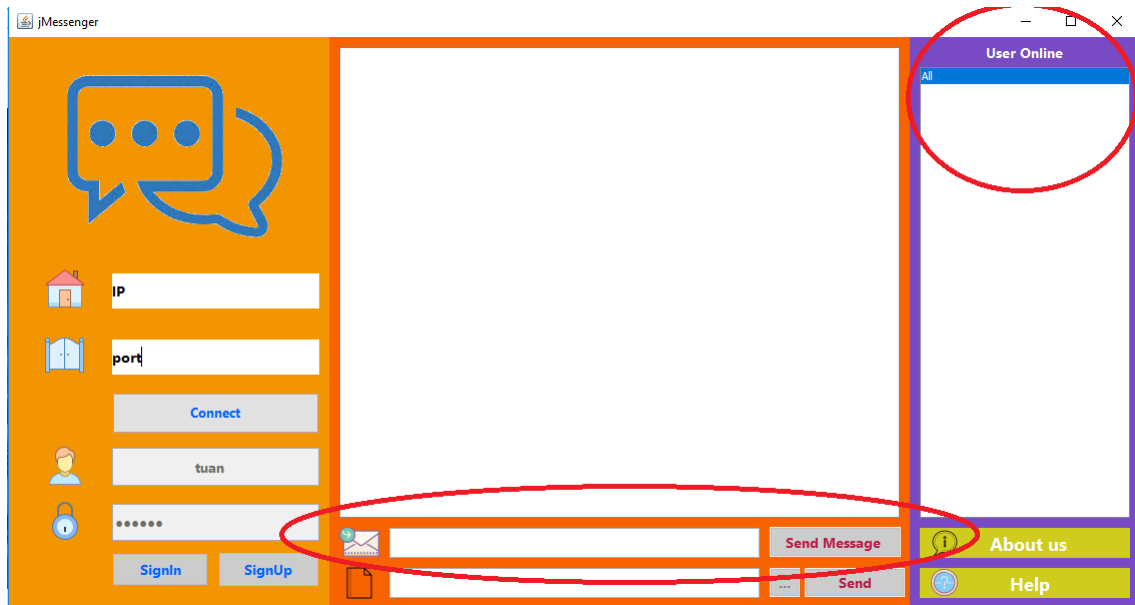


A screenshot of the jMessenger application interface. The left sidebar contains a chat icon, a home icon, a calendar icon, a user icon, and a lock icon. Below these are input fields for IP, a number (10000), a name (tuan), and a password (\*\*\*\*\*). The 'SignIn' button is circled in red. The main area is a large white rectangle. The right sidebar contains a 'User Online' section, an 'About us' button, and a 'Help' button.

Hình 3: Giao diện người dùng đăng nhập

### 6.1.3 Gửi tin nhắn

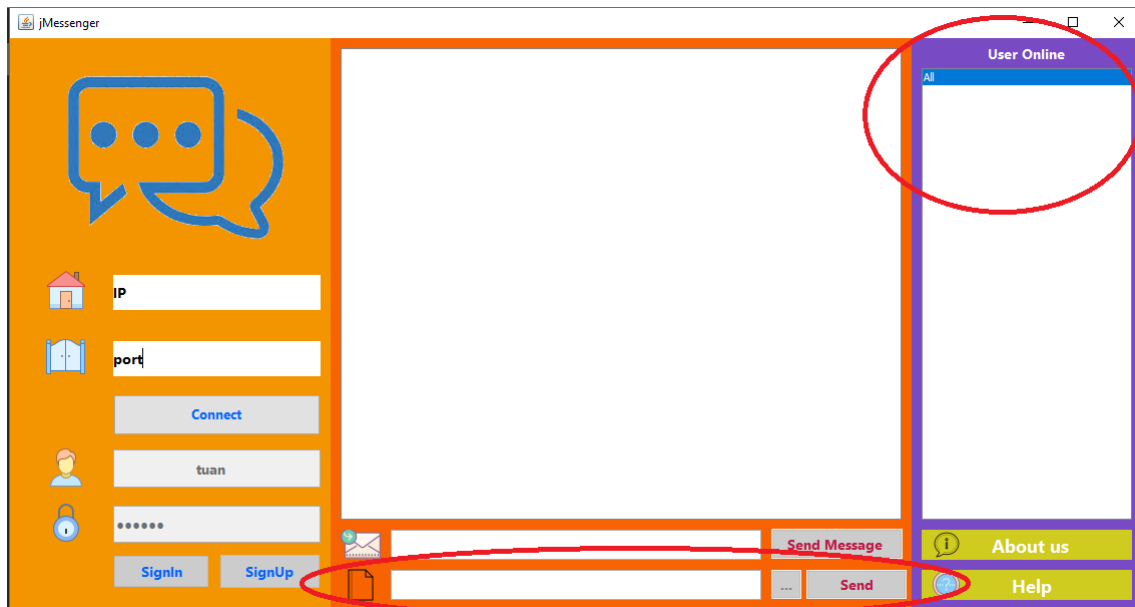
- Để gửi tin nhắn đi, người dùng chọn người cần gửi trong danh sách liên hệ đang hoạt động ở bên phải, sau đó nhập tin nhắn vào ô phía dưới và nhấn nút gửi tin nhắn.



Hình 4: Giao diện người dùng gửi tin nhắn

### 6.1.4 Gửi tệp

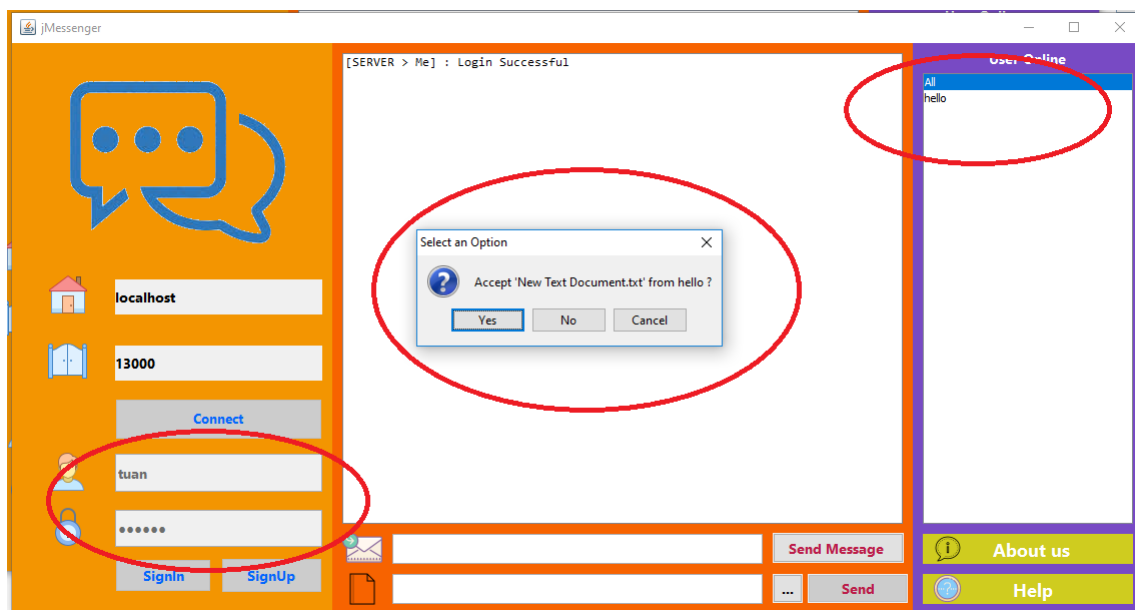
- Để gửi tệp đi, người dùng chọn người cần gửi trong danh sách liên hệ đang hoạt động ở bên phải, sau đó nhấn chọn để tải file lên và nhấn nút gửi tệp.



Hình 5: Giao diện người dùng gửi tệp

#### 6.1.5 Nhận tệp

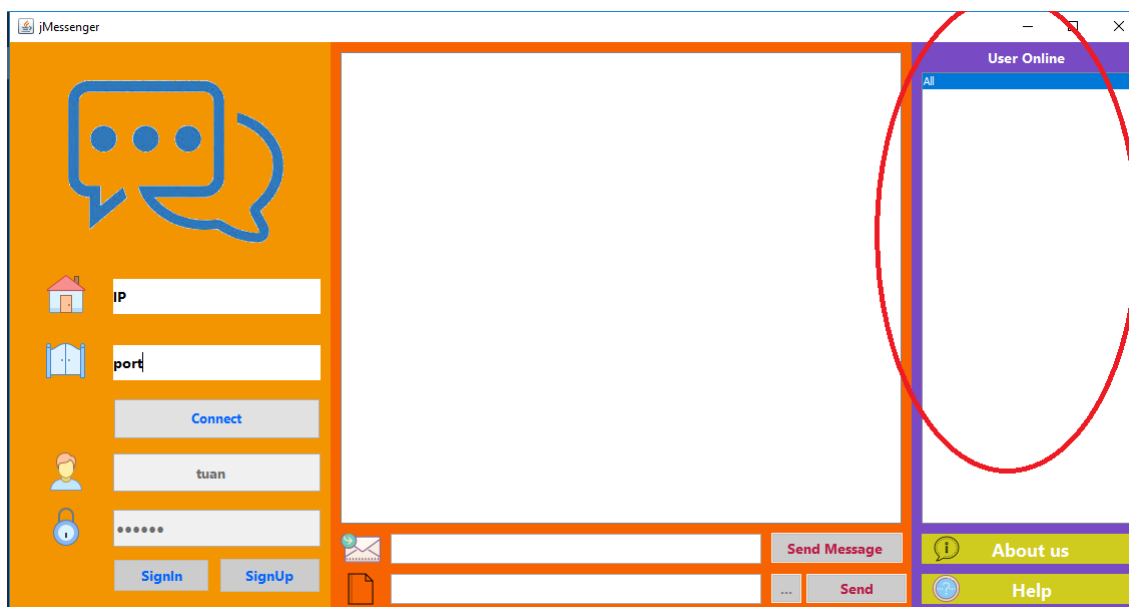
- Khi người dùng được một liên hệ khác gửi tệp, giao diện chat sẽ hiển thị một hộp thoại với tên của tệp được gửi đến, tên người gửi và các lựa chọn nhận tệp, không nhận tệp hay hủy.



Hình 6: Giao diện người dùng nhận tệp

### 6.1.6 Xem liên hệ đang hoạt động

- Để xem liên hệ đang hoạt động, người dùng xem ở mục User Online ở bên phải giao diện chính, được hiển thị dưới dạng danh sách.

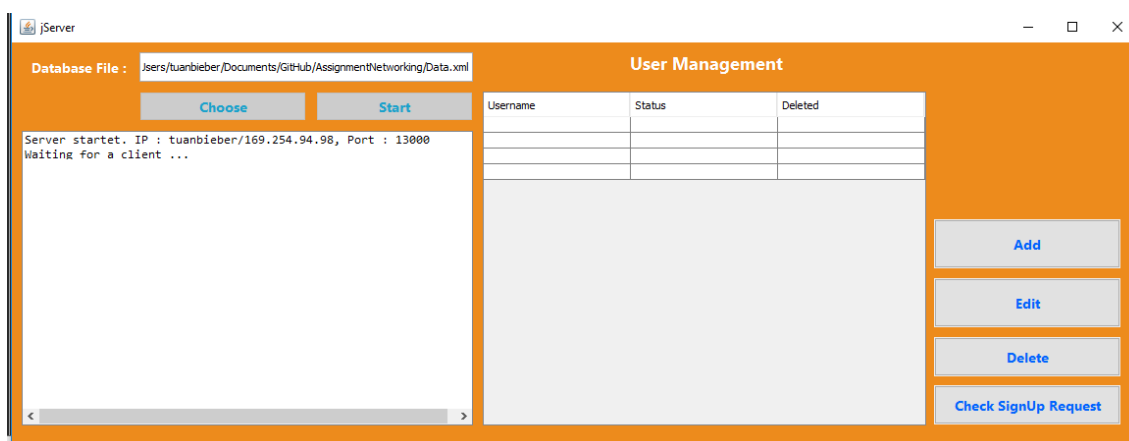


Hình 7: Giao diện người dùng xem danh sách liên hệ

## 6.2 Các chức năng với vai trò người quản lý

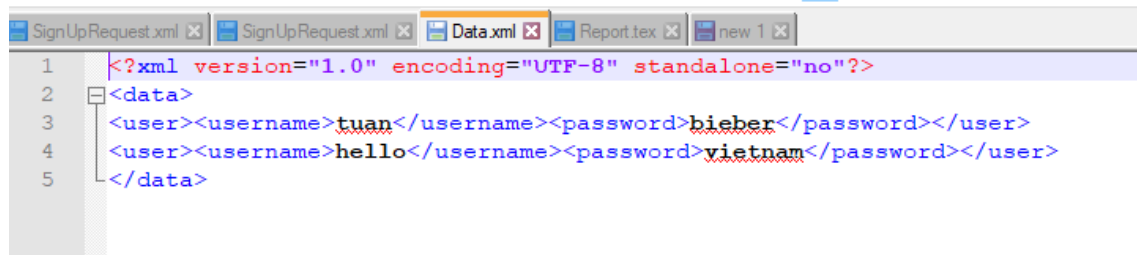
### 6.2.1 Tạo server phục vụ người dùng

- Để tạo một Server mới, trước tiên người quản lý cần tải lên file dữ liệu, file này bao gồm tên và mật khẩu của tất cả người dùng đã được kích hoạt trước đó (Không bao gồm tên và mật khẩu của người dùng nằm trong file đang chờ duyệt).



Hình 8: Giao diện người quản lý

- Mẫu file dữ liệu được lưu trên máy chủ:

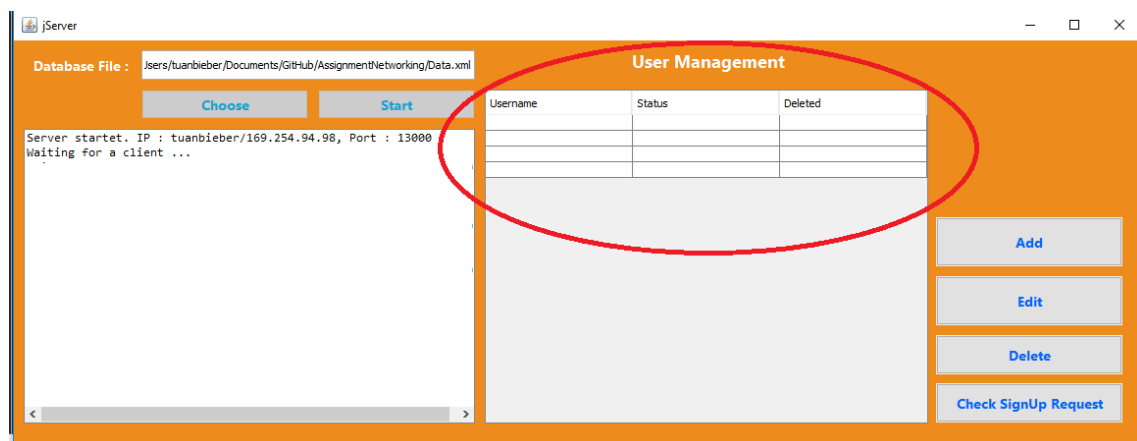


```
1 <?xml version="1.0" encoding="UTF-8" standalone="no"?>
2 <data>
3 <user><username>tuan</username><password>bieber</password></user>
4 <user><username>hello</username><password>vietnam</password></user>
5 </data>
```

Hình 9: Cấu trúc file dữ liệu được lưu trên máy chủ

## 6.2.2 Quản lý người dùng đang hoạt động và người dùng đã bị vô hiệu hóa tài khoản

- Quản lý nhìn vào bảng trên giao diện chính để xem người dùng nào đang hoạt động (status = 1), không hoạt động (status = 0) hoặc người dùng đã bị vô hiệu hóa tài khoản (deleted = 1)



Hình 10: Giao diện quản lý xem danh sách tất cả người dùng

## 7 Thiết kế chi tiết ứng dụng

- Những class chính trong mục ServerPart:

### 7.0.1 SocketServer

### 7.0.2 Message

### 7.0.3 Database

### 7.0.4 SignUp

- Những class chính trong mục UserPart:

#### 7.0.5 SocketClient

#### 7.0.6 Message

#### 7.0.7 Download

#### 7.0.8 Upload

#### 7.0.9 SignUp

### 8 Đánh giá kết quả hiện thực

- Nhóm đã tổ chức làm việc theo hai giai đoạn rõ ràng.
- Cơ bản đã hoàn thành được 80 phần trăm theo yêu cầu của đề bài.

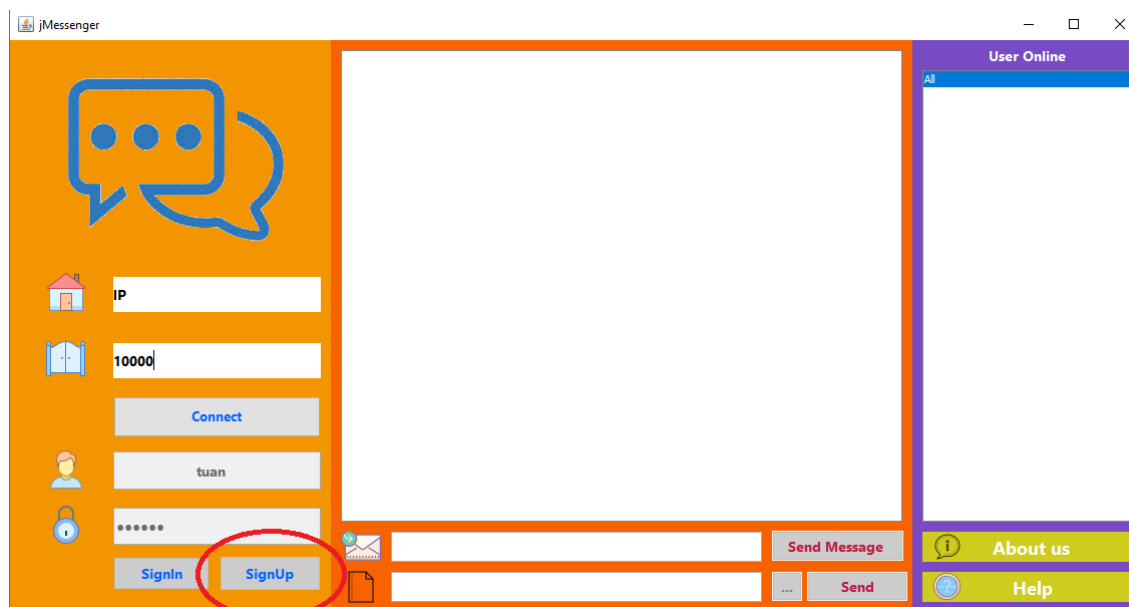
### 9 Các chức năng mở rộng của hệ thống ngoài các yêu cầu được qui định

- Nhóm không thực hiện thêm chức năng mở rộng.

### 10 Hướng dẫn sử dụng ứng dụng

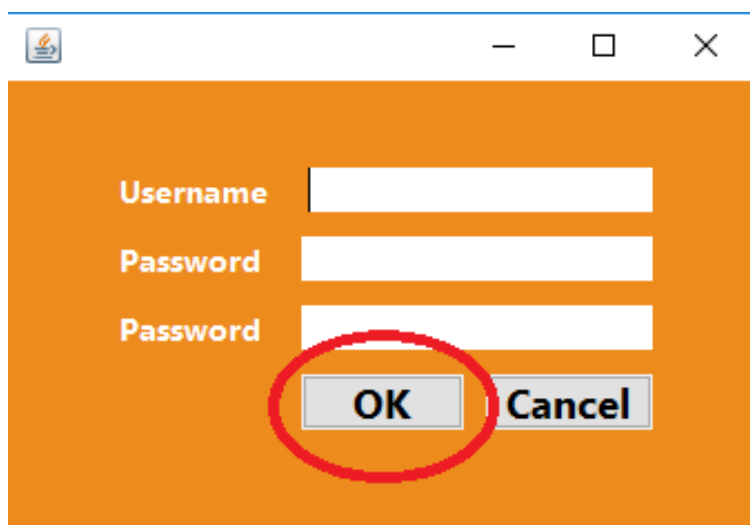
#### 10.0.1 Hướng dẫn đăng ký tài khoản

- Người dùng khởi động ứng dụng, sau đó bấm vào nút SignUp ở giao diện chính



Hình 11: Hướng dẫn đăng ký tài khoản

- Sau đó, nhập thông tin bao gồm tên người dùng và mật khẩu vào form đăng ký và nhấn chọn OK. Thông tin đăng ký sẽ được gửi về máy chủ và đợi người quản lý duyệt.

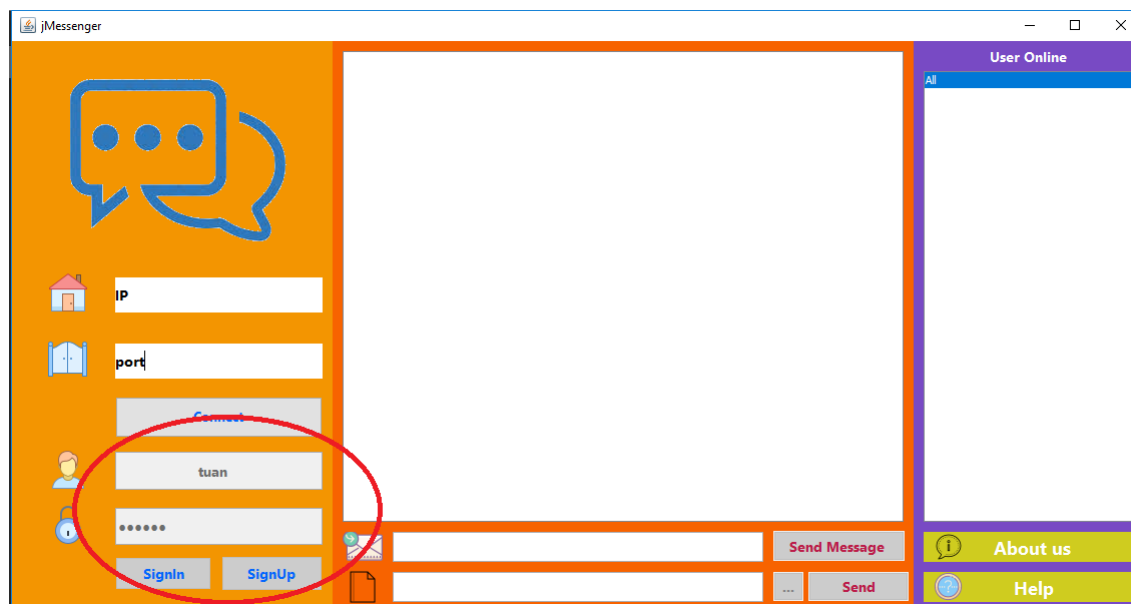


A registration form window with an orange background. It contains three input fields: 'Username', 'Password', and 'Password'. Below the fields are two buttons: 'OK' and 'Cancel'. The 'OK' button is circled in red.

Hình 12: Hướng dẫn điền thông tin đăng ký tài khoản

#### 10.0.2 Hướng dẫn đăng nhập

- Để đăng nhập người dùng ấn chọn nút SignIn sau khi đã nhập thông tin username và password. Nếu thông tin hợp lệ, người dùng sẽ được sử dụng các chức năng tiếp theo như xem liên hệ đang hoạt động, gửi tin nhắn, gửi tệp.



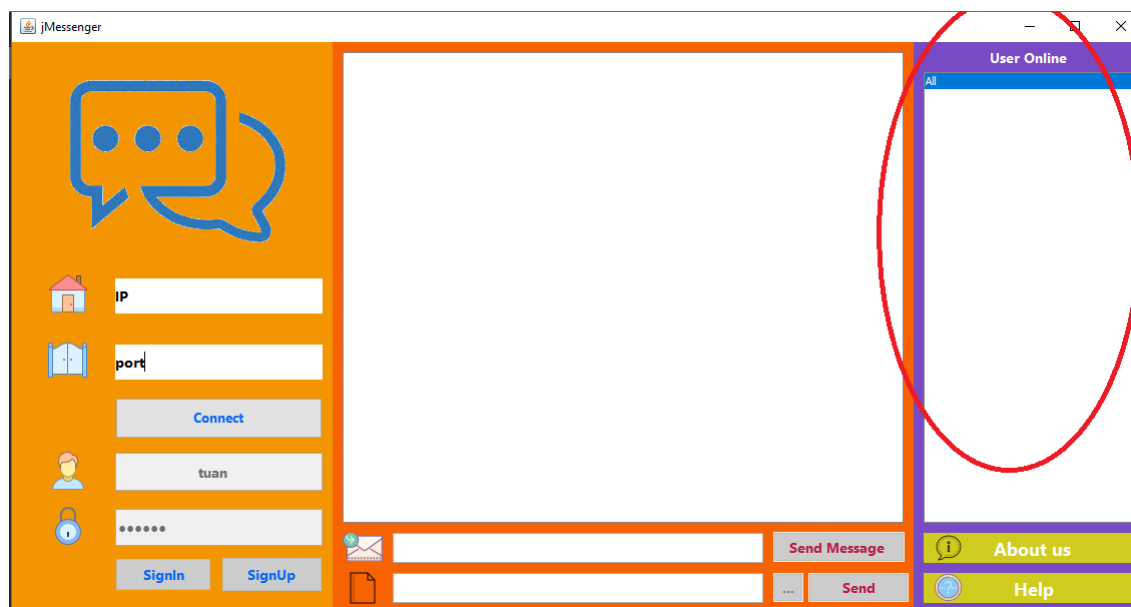
The jMessenger application interface. On the left, there is a sidebar with a chat icon and input fields for 'IP', 'port', 'username' (containing 'tuan'), and 'password' (masked with dots). Below these fields are 'SignIn' and 'SignUp' buttons. The 'SignIn' button is circled in red. The main area is a large white rectangle. On the right, there is a 'User Online' list. At the bottom, there are input fields for a message and a 'Send Message' button, and another input field for a file and a 'Send' button. The bottom right corner has 'About us' and 'Help' buttons.

Hình 13: Hướng dẫn đăng nhập



### 10.0.3 Hướng dẫn xem liên hệ đang hoạt động

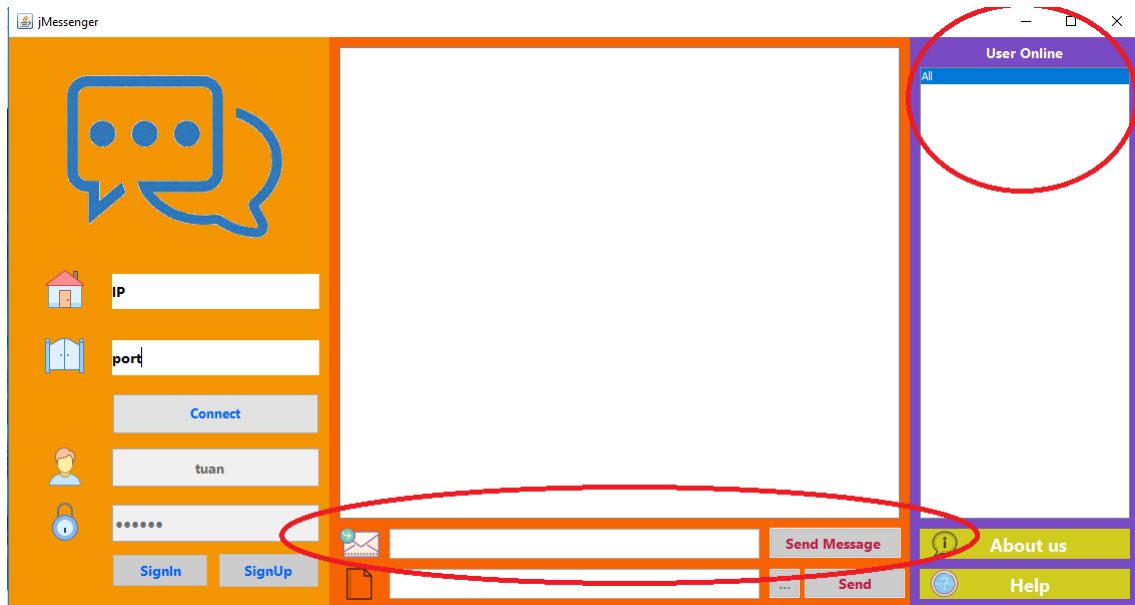
- Để xem các liên hệ đang hoạt động, trước tiên, người dùng phải đăng nhập thành công vào hệ thống, sau đó nhìn vào danh sách User Online nằm ở bên phải giao diện chính.



Hình 14: Hướng dẫn xem danh sách liên hệ đang hoạt động

### 10.0.4 Hướng dẫn gửi tin nhắn

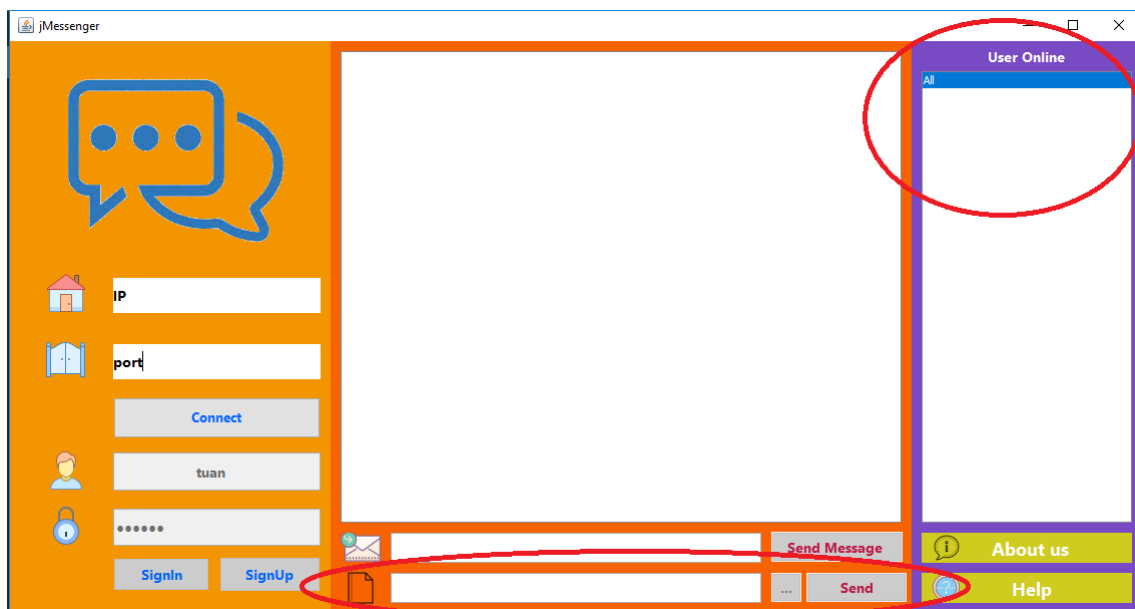
- Để gửi tin nhắn đi, người dùng chọn người cần gửi trong danh sách liên hệ đang hoạt động ở bên phải, sau đó nhập tin nhắn vào ô phía dưới và nhấn nút gửi tin nhắn.



Hình 15: Hướng dẫn gửi tin nhắn

#### 10.0.5 Hướng dẫn gửi tệp tin

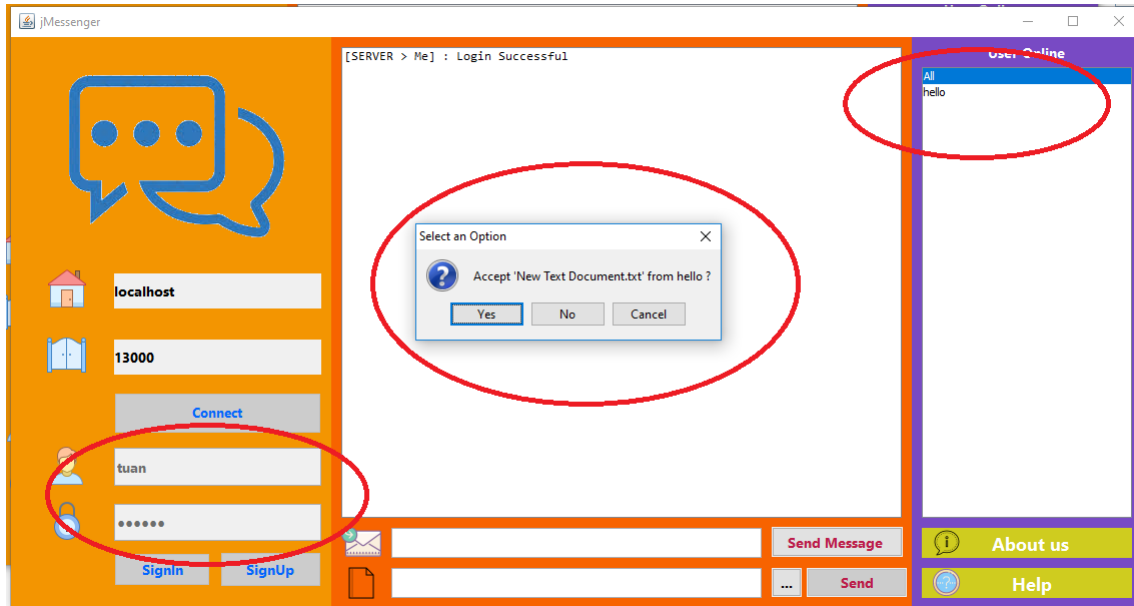
- Để gửi tệp đi, người dùng chọn người cần gửi trong danh sách liên hệ đang hoạt động ở bên phải, sau đó nhấn chọn để tải file lên và nhấn nút gửi tệp.



Hình 16: Hướng dẫn gửi tệp tin

### 10.0.6 Hướng dẫn nhận tệp tin

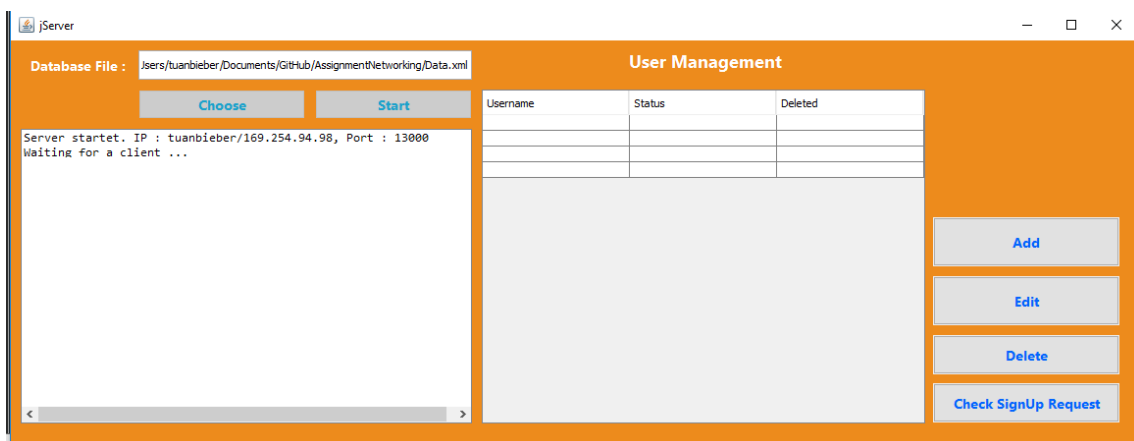
- Khi người dùng được một liên hệ khác gửi tệp, giao diện chat sẽ hiển thị một hộp thoại với tên của tệp được gửi đến, tên người gửi và các lựa chọn nhận tệp, không nhận tệp hay hủy.



Hình 17: Hướng dẫn nhận nhận tệp

### 10.0.7 Hướng dẫn tạo server chat

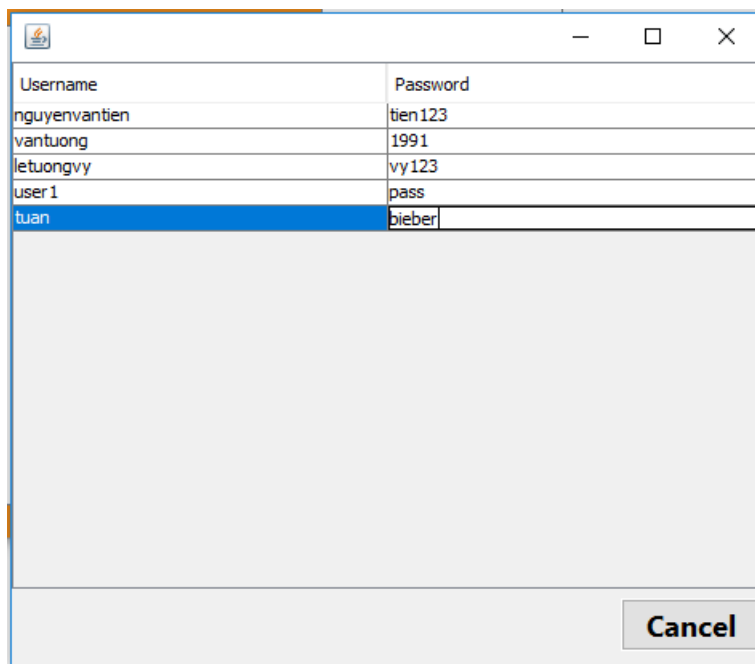
- Để tạo một Server mới, trước tiên người quản lý cần tải lên file dữ liệu, file này bao gồm tên và mật khẩu của tất cả người dùng đã được kích hoạt trước đó (Không bao gồm tên và mật khẩu của người dùng nằm trong file đang chờ duyệt).



Hình 18: Hướng dẫn tạo server

#### 10.0.8 Hướng dẫn xem danh sách đăng ký đang chờ được duyệt

- Để xem danh sách người dùng đã đăng ký tài khoản nhưng chưa được duyệt, người quản lý ấn vào nút "Check SignUp Request", form chứa thông tin người dùng đang chờ duyệt sẽ hiện lên.



Username	Password
nguyenvantien	tien123
vantuong	1991
letuongvy	vy123
user1	pass
tuan	bieber

Cancel

Hình 19: Hướng dẫn xem danh sách người dùng đăng ký

## 11 Mã nguồn chương trình (softcopy)

- Mã nguồn chương trình của nhóm được lưu trữ tại trang Github
- Đường dẫn: <https://github.com/GiaDiBaby/AssignmentNetworking>

## 12 Ứng dụng chat (softcopy) được biên dịch từ mã nguồn

### Tài liệu

- [1] Codeproject  
”<<https://www.codeproject.com/Articles/524120/A-Java-Chat-Application>>”
- [2] ChatAppJava  
“<https://youtu.be/Q-wzRVgqM-M>”,
- [3] GpCoder  
“<https://gpcoder.com/3679-xay-dung-ung-dung-client-server-voi-socket-trong-java/>”,



- [4] <http://tech.agu.edu.vn>  
“<http://tech.agu.edu.vn/clbtinhoc/index.php?threads/java-huong-dn-vit-chuong-trinh-chat-client-server-don-gian.512/>”,
- [5] [github.com](https://github.com)  
“<https://github.com/Drakirus/java-chat>”,