

TRƯỜNG ĐẠI HỌC BÁCH KHOA  
-ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH  
KHOA KHOA HỌC VÀ KỸ THUẬT MÁY TÍNH



BÁO CÁO BÀI TẬP LỚN SỐ 1

Môn: MẠNG MÁY TÍNH

Giảng viên hướng dẫn: BÙI XUÂN GIANG

Danh sách thành viên nhóm

Tên	Mã số sinh viên
Nguyễn Đăng Hà Nam	1710195
Nguyễn Thị Trúc Ly	1710187
Trần Hồ Minh Thông	1710314

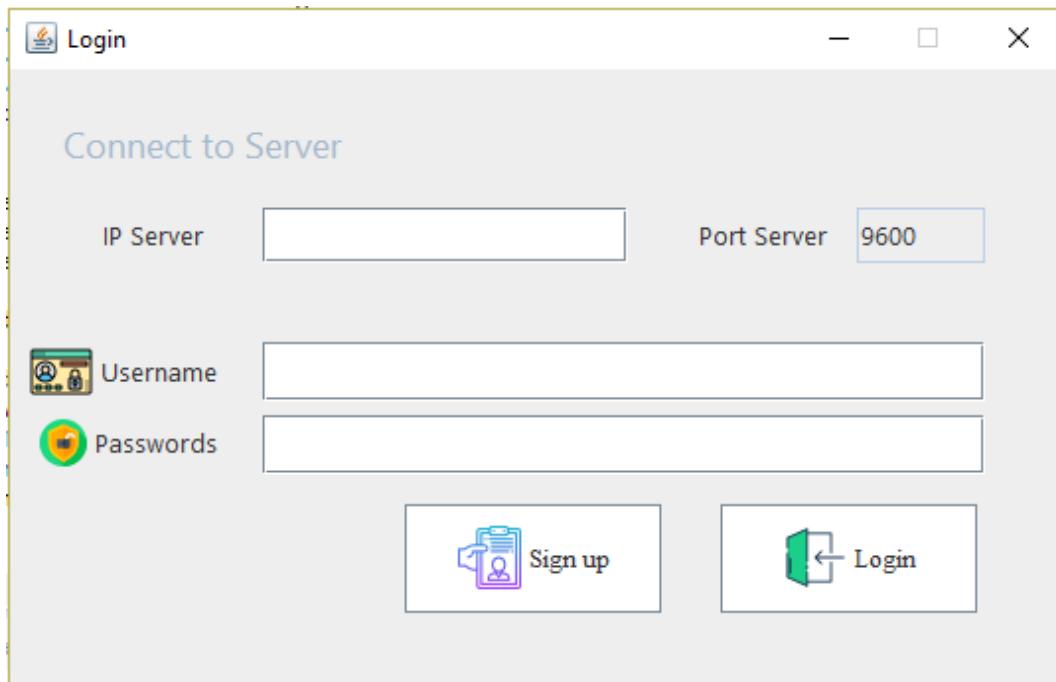
## Mục lục

1. Giới thiệu ứng dụng: .....	1
1.1. Chức năng đăng nhập: .....	1
1.2. Chức năng đăng ký tài khoản: .....	1
1.3. Chức năng gửi/ nhận tin nhắn riêng tư: .....	2
1.4. Gửi/ nhận tập tin riêng tư: .....	3
1.5. Gửi/ nhận tin nhắn trong cuộc trò chuyện nhóm (phần mở rộng): .....	4
2. Các giao thức được sử dụng: .....	5
3. Thiết kế ứng dụng: .....	8
3.1. Nền tảng công nghệ: .....	8
3.2. Các mô hình mạng máy tính: .....	8
3.3. Kiến trúc tổng thể: .....	9
3.4. Sơ đồ lớp: .....	11
3.5. Các lớp được sử dụng: .....	11
3.5.1 Các lớp vận hành ứng dụng ở máy khách: .....	11
3.5.2 Các lớp vận hành ứng dụng ở máy chủ: .....	12
3.5.3 Các lớp khác: .....	12
4. Hướng dẫn sử dụng: .....	12
4.1 Đối với quản trị viên: .....	12
4.2 Đối với người dùng: .....	12
5. Đánh giá kết quả hiện thực: .....	14
5.1 Kết quả đạt được: .....	14
5.2. Những điều chưa đạt được: .....	15
6. Tài liệu tham khảo: .....	15
References .....	15

## 1. Giới thiệu ứng dụng:

### 1.1. Chức năng đăng nhập:

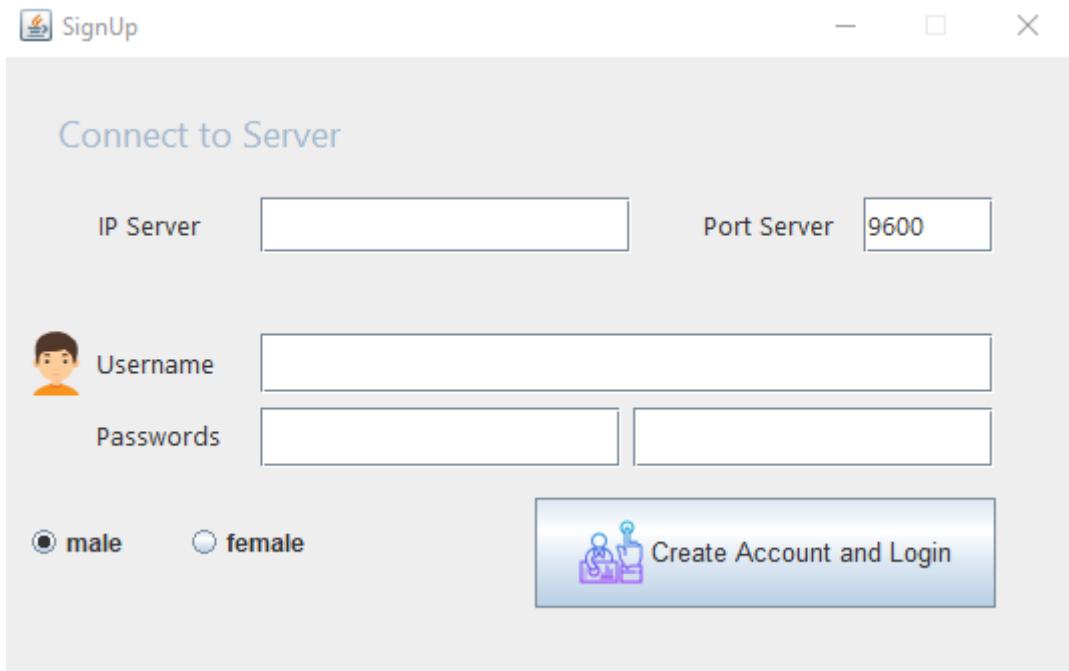
Chức năng này cho phép người dùng đăng nhập vào một tài khoản đã có sẵn với tên đăng nhập và mật khẩu(như hình 1). Nó đòi hỏi người dùng cần biết địa chỉ IP và port của máy chủ để đăng nhập.



Hình 1: Chức năng đăng nhập.

### 1.2. Chức năng đăng ký tài khoản:

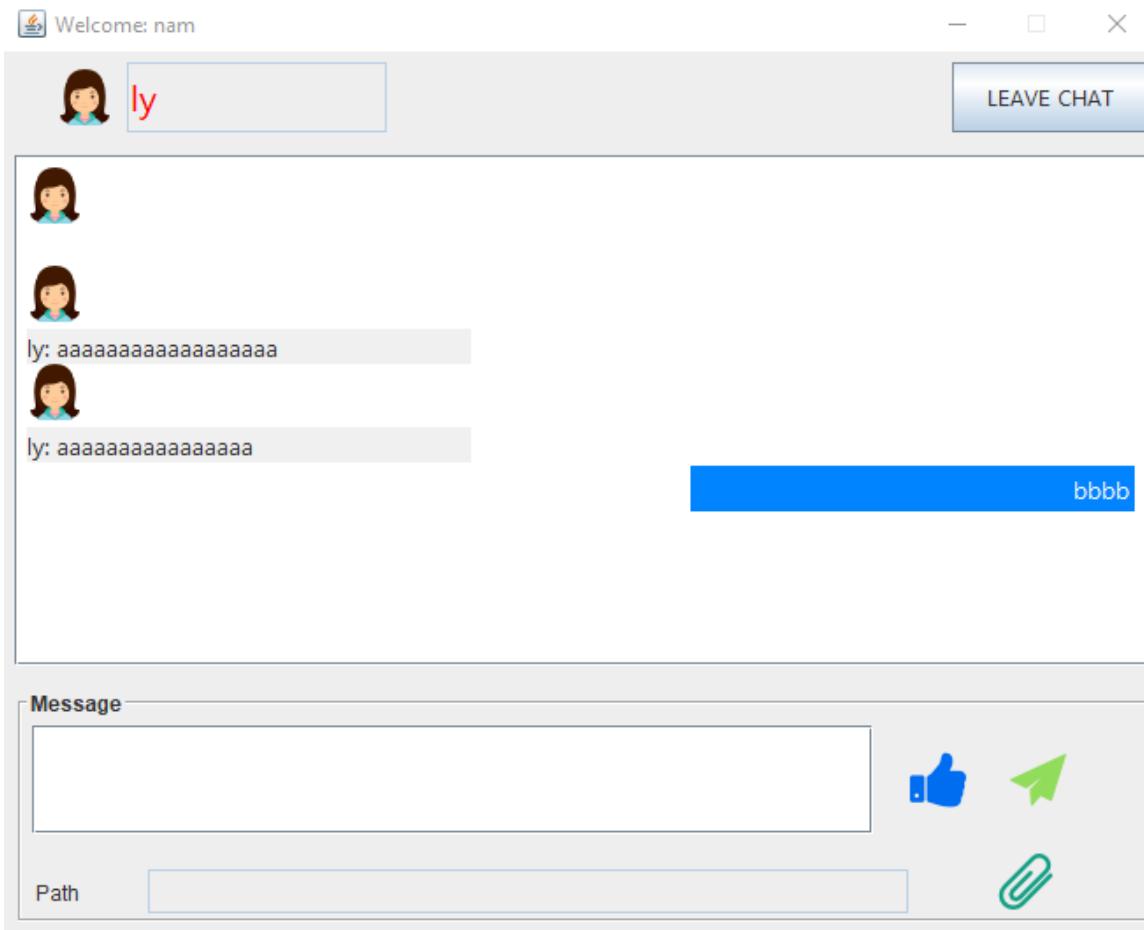
Chức năng này cho phép người dùng tạo một tài khoản mới, với điều kiện tên tài khoản mới không được trùng với bất cứ tên tài khoản nào đã có trước (như hình 2). Ngoài ra, người dùng cần biết địa chỉ IP và port của máy chủ để đăng ký tài khoản.



*Hình 2: Chức năng đăng ký tài khoản mới.*

### 1.3. Chức năng gửi/ nhận tin nhắn riêng tư:

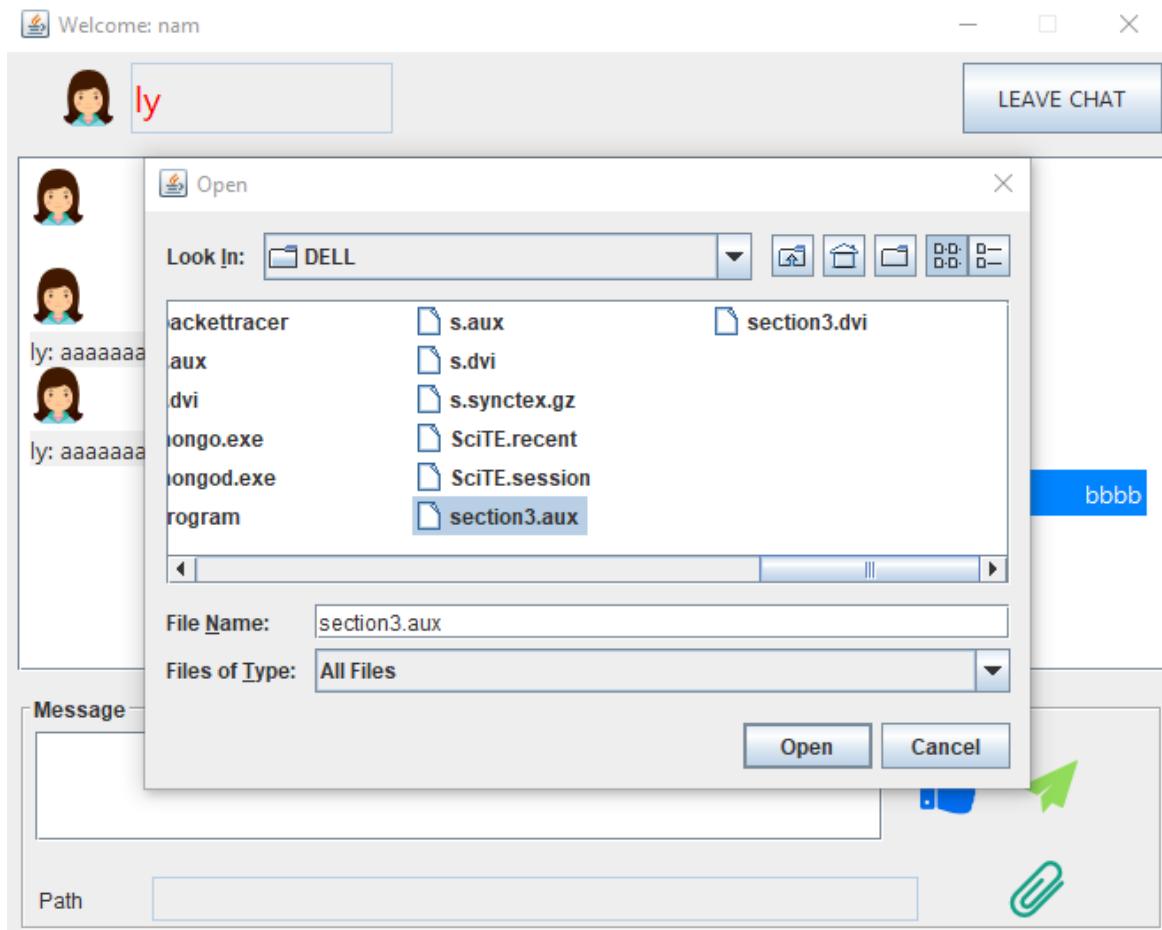
Chức năng này cho phép 2 người dùng gửi/ nhận tin nhắn với nhau theo mô hình peer-to-peer, với điều kiện cả hai người dùng đều phải đang trực tuyến (như hình 3). Chỉ có người gửi và người nhận nhìn thấy nội dung tin nhắn.



Hình 3: Chức năng gửi/ nhận tin nhắn riêng tư.

#### 1.4. Gửi/ nhận tập tin riêng tư:

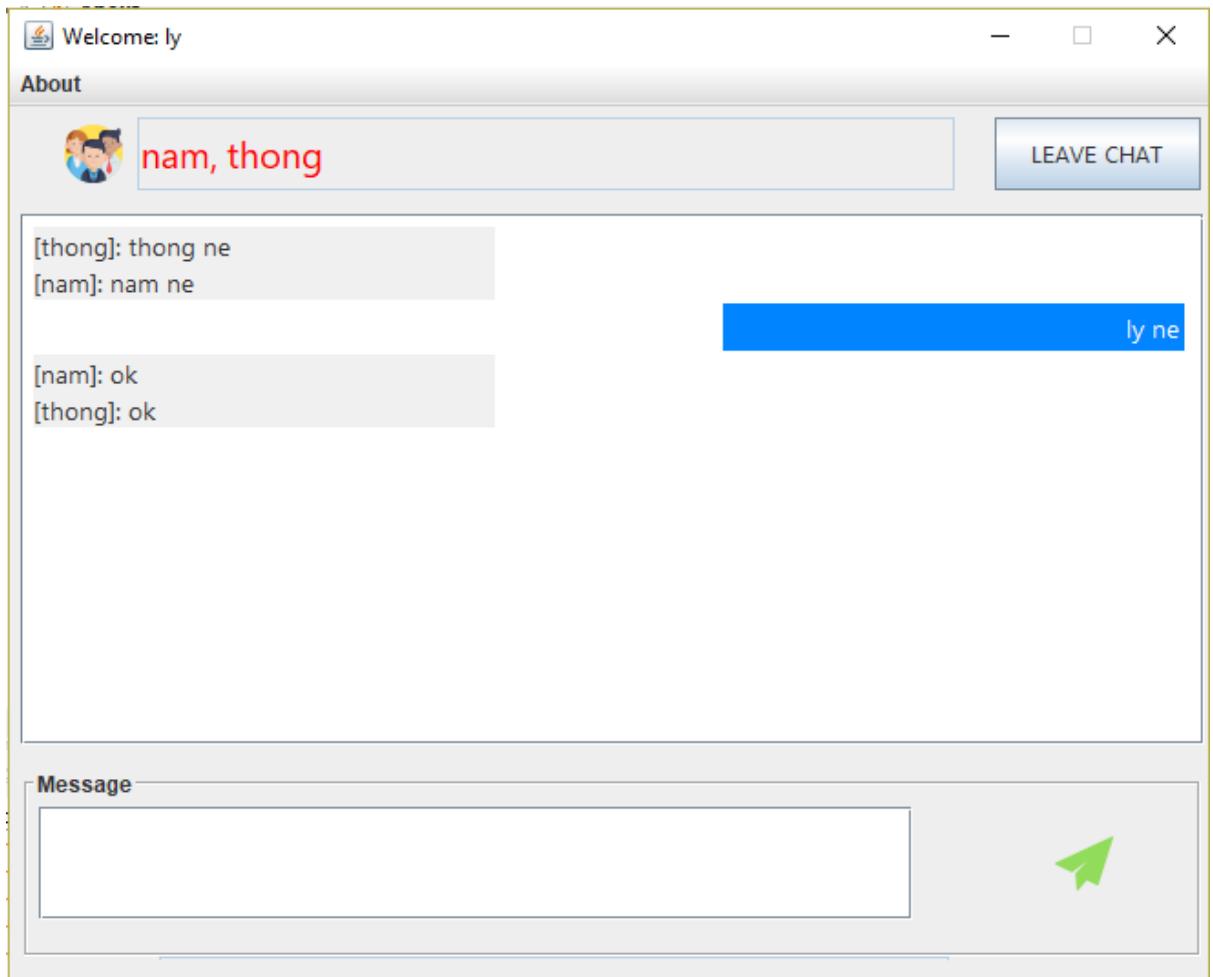
Chức năng này cho phép 2 người dùng gửi/ nhận file với nhau theo mô hình peer-to-peer, với điều kiện cả hai người dùng đều phải đang trực tuyến (như hình 4). Chỉ có người gửi và người nhận nhìn thấy tập tin.



Hình 4: Chức năng gửi/ nhận file riêng tư.

### 1.5. Gửi/ nhận tin nhắn trong cuộc trò chuyện nhóm (phần mở rộng):

Chức năng này cho phép 2 người dùng gửi/ nhận tin nhắn với nhau, với điều kiện có ít nhất hai người dùng đều phải đang trực tuyến (như hình 5). Mọi người trong cuộc trò chuyện nhóm nhìn thấy tin nhắn của nhau.



*Hình 5: Chức năng gửi nhận tin nhắn trong nhóm*

## 2. Các giao thức được sử dụng:

Nội dung	Mục đích	Ghi chú
<SESSION_REQ>clientID</SESSION_REQ>	Yêu cầu tạo một chat session từ client. clientID là id của người dùng	Sử dụng 1 lần cho mỗi chat session
<PEER_NAME>userName</PEER_NAME>	Tạo user khi người dùng đăng nhập vào server	userName là tên người dùng
numPort	Tạo port cho user	numPort là số hiệu port
<SESSION_KEEP_ALIVE></SESSION_KEEP_ALIVE>	Nội dung request do user gửi lên server	User request

		10s/lần bao gồm tên user và trạng thái
statusUser	Thông báo trạng thái user	statusUser là trạng thái user
<SESSION_DENY />	Từ chối người dùng kết nối tới server	Từ chối sau khi người dùng login
<SESSION_ACCEPT></SESSION_ACCEPT>	Chấp nhận người dùng kết nối tới server	Chấp nhận sau khi người dùng login
<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
stringIP	Định danh IP cho user	stringIP là IP của user
<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

peerData	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
MAX_MSG_SIZE	102400	Kích thước tối đa của một message(file) trong chat

SERVER_ONLINE	"RUNNING"	User đang online
SERVER_OFFLINE	"STOP"	User thoát khỏi chat session

### 3. Thiết kế ứng dụng:

#### 3.1. Nền tảng công nghệ:

Thư viện java.io hỗ trợ tạo, điều khiển và đóng socket, server socket.

Thư viện java.swing và java.awt hỗ trợ tạo, định dạng và xử lý sự kiện cho các đối tượng giao diện.

#### 3.2. Các mô hình mạng máy tính:

Mô hình server-client được sử dụng khi hiện thực các chức năng sau:

- Đăng ký tài khoản mới: Tên tài khoản, mật khẩu và chuỗi mật khẩu nhập lại sẽ được gửi từ máy người dùng (client) tới máy chủ (server). Sau khi đã kiểm tra dữ liệu gửi tới, nếu chúng hợp lệ, máy chủ sẽ lưu lại tài khoản này vào tập tin ‘account.txt’ (tập tin ghi lại trạng thái trực tuyến/ ngoại tuyến của tất cả các tài khoản đã được đăng ký), gửi tin nhắn chuyển sang cửa sổ trò chuyện tới máy người dùng và gửi tin nhắn cập nhật danh sách đang trực tuyến tới tất cả các máy tính của các người dùng đang trực tuyến khác. Ngược lại, máy chủ sẽ gửi các thông báo dữ liệu không hợp lệ tới máy người dùng.

- Đăng nhập: Tên tài khoản và mật khẩu sẽ được gửi từ máy người dùng (client) tới máy chủ (server). Sau khi đã kiểm tra dữ liệu gửi tới, nếu chúng hợp lệ, máy chủ sẽ cập nhật lại tập tin ‘account.txt’, gửi tin nhắn chuyển sang cửa sổ trò chuyện tới máy người dùng và gửi tin nhắn cập nhật danh sách đang trực tuyến tới tất cả các máy tính của các người dùng đang trực tuyến khác. Ngược lại, máy chủ sẽ gửi các thông báo dữ liệu không hợp lệ tới máy người dùng.

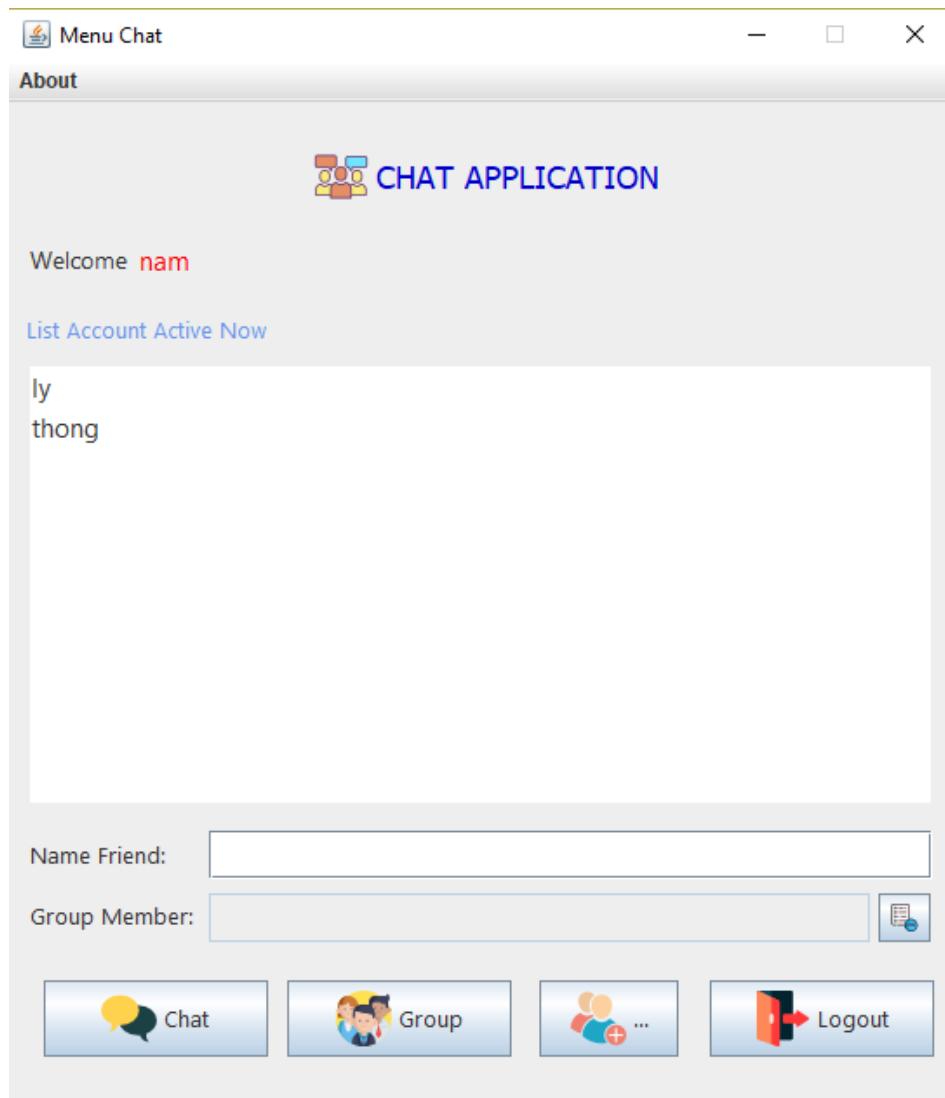
Mô hình peer-to-peer được sử dụng giúp các máy của người dùng giao tiếp với nhau: Mỗi máy người dùng có một server socket luôn được mở, người dùng khác tạo socket rồi kết nối vào server socket này để tạo kết nối

giữa hai máy. Các tin nhắn / tập tin giữa hai máy sẽ được gửi trực tiếp qua kết nối này, mà không qua máy chủ. Như vậy, mỗi máy người dùng sẽ giữ một server socket và một danh sách các socket kết nối tới các máy khác.

### 3.3. Kiến trúc tổng thể:

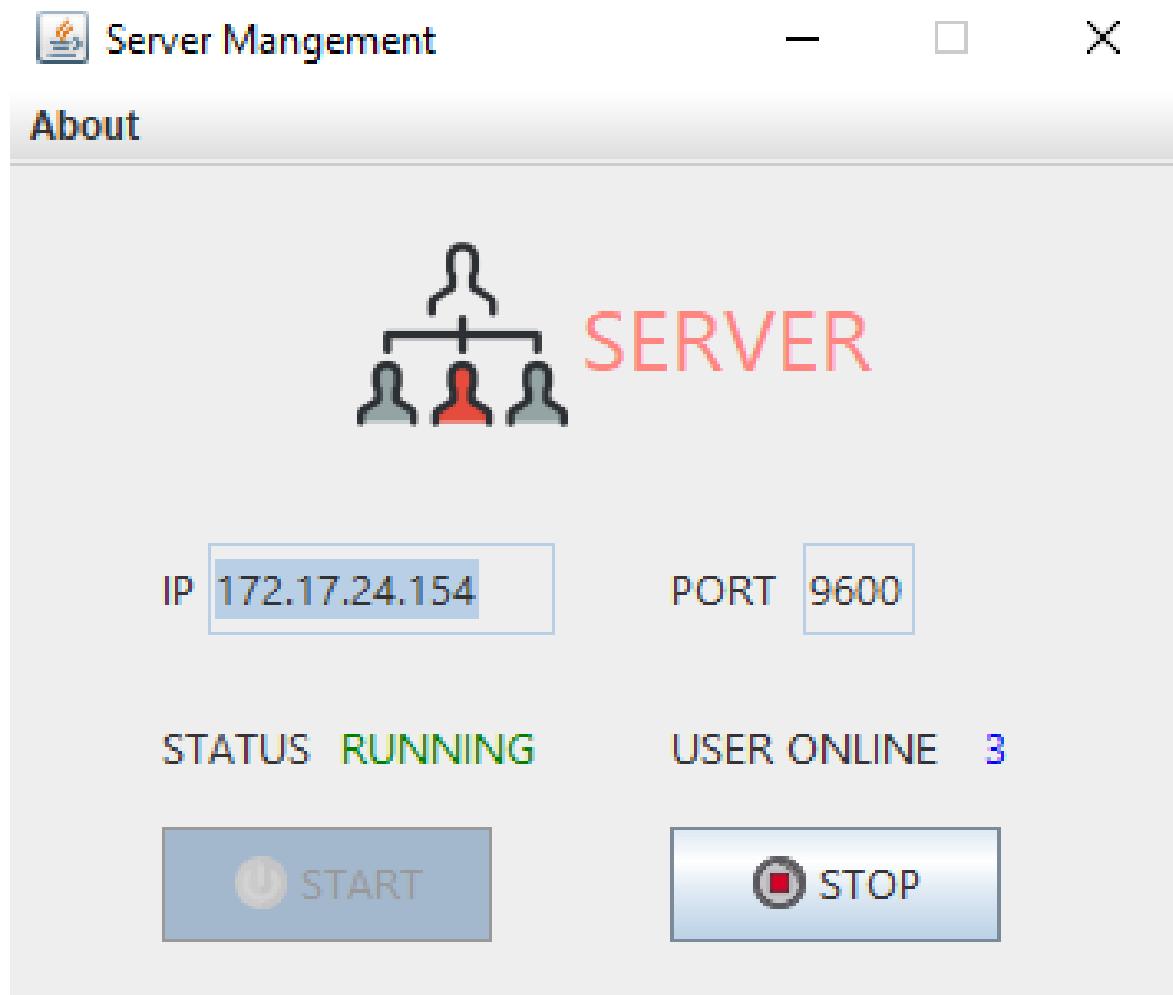
Các cửa sổ chính của ứng dụng:

- Cửa sổ ‘Login’: Đây là điểm nhập của chương trình ở phía người dùng (hình 1).
- Cửa sổ ‘SignUp’: Cửa sổ này được mở ra khi người dùng chọn nút ‘SignUp’ trong cửa sổ ‘Login’ (hình 2).
- Cửa sổ ‘MainGui’: Là cửa sổ chính từ phía người dùng (hình 6). Người dùng có thể nhìn thấy các danh sách các người dùng đang trực tuyến.



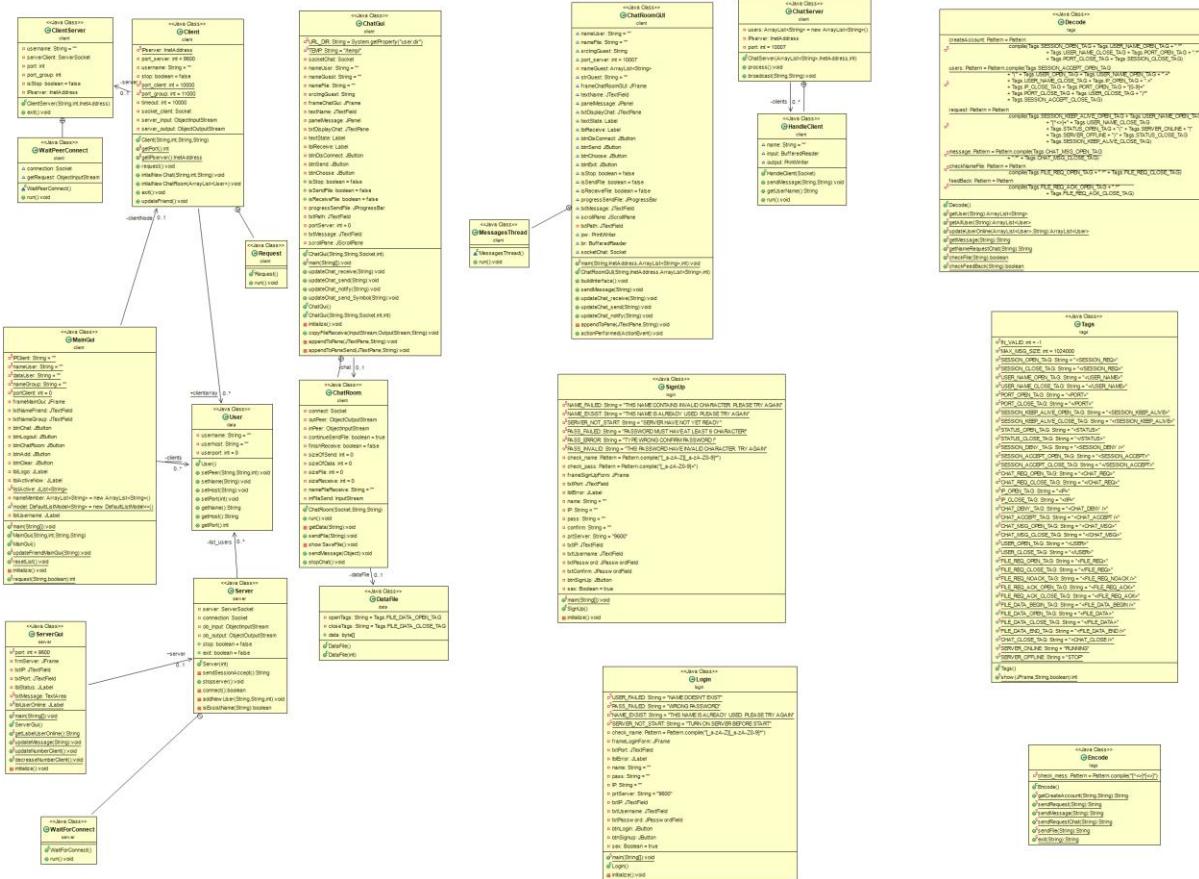
Hình 6: Cửa sổ ‘MainGui’

- Cửa sổ ‘ChatGui’: Đây là cửa sổ trò chuyện riêng tư giữa hai người dùng (hình 3). Nó được mở ra khi người dùng chọn tên người cần trò chuyện và nháy nút ‘Chat’ trong cửa sổ ‘MainGui’. Cửa sổ này in nội dung gửi đi bên phải và in nội dung nhận về bên trái khung trò chuyện.
- Cửa sổ ‘ServerGui’: Đây là cửa sổ chỉ được nhìn thấy ở máy chủ (hình 7). Nó hiển thị các thông tin server như: IP, port, số lượng người dùng đang trực tuyến, các giao thức mà máy khách gửi lên. Qua cửa sổ này, quản trị viên của hệ thống có thể bật, tắt server.



*Hình 7: Cửa sổ “ServerGui”*

### 3.4. Sơ đồ lớp:



### **3.5. Các lớp được sử dụng:**

### 3.5.1 Các lớp vận hành ứng dụng ở máy khách:

- MainGui.java: hiện thực cửa sổ ‘MainGui’.
  - ChatGui.java: Xử lý việc chọn tập tin trong máy mà người dùng muốn gửi đi và hiện thực cửa sổ ‘ChatGui’.
  - ClientServer.java: Tạo và quản lý ServerSocket của người dùng. Xử lý lời mời mở cuộc trò chuyện riêng tư/ nhóm từ các người dùng khác.
  - Client.java: Quản lý thông tin hiện tại của người dùng. Nó sẽ tạo một luồng để liên tục gửi yêu cầu cập nhật danh sách người dùng đang trực tuyến tới máy chủ. Ngoài ra nó còn có nhiệm vụ gửi lời mời mở cuộc trò chuyện riêng tư/ nhóm tới máy khác.
  - ChatRoomGUI.java: Xử lý việc thực thi chat giữa nhiều người trong cùng một group, thực hiện chat group, và hiện thực cửa sổ ‘ChatRoomGUI’.
  - ChatServer.java: Xử lý kết nối, tạo group chat với nhiều người.

### **3.5.2 Các lớp vận hành ứng dụng ở máy chủ:**

- ServerGui.java: Hiện thực cửa sổ “ServerGui”.
- Server.java: Quản lý máy chủ: tạo một SocketServer và một luồng là WaitForConnect để xử lý các yêu cầu kết nối từ người dùng như: đăng nhập, thoát ứng dụng, lấy danh sách các người dùng đang trực tuyến; cập nhật tình trạng trực tuyến/ ngoại tuyến của toàn bộ tài khoản và phát tán danh sách các người dùng đang trực tuyến.

### **3.5.3 Các lớp khác:**

- DataFile.java: Tạo một đối tượng File để gửi đi.
- Peer.java: Tạo một đối tượng Peer có các thuộc tính: IP, port, name.
- Tags.java: Định nghĩa các giao thức được sử dụng trong ứng dụng.
- Encode.java: Định nghĩa các phương thức người dùng để gửi yêu cầu lên máy chủ.
- Dedode.java: Giải mã các phương thức để lấy thông tin người dùng, port hay ip...

## **4. Hướng dẫn sử dụng:**

### **4.1 Đối với quản trị viên:**

- Bước 1: chạy tập tin ServerGui.java.
- Bước 2: chọn nút ‘START’ để khởi động ứng dụng trên máy chủ.
- Bước 3: chọn nút ‘STOP’ để dừng ứng dụng trên máy chủ.
- Bước 4: chọn nút ‘X’ trên thanh trạng thái để tắt ứng dụng trên máy chủ.

### **4.2 Đối với người dùng:**

- Bước 1: Chạy tập tin Login.java để mở cửa sổ đăng nhập.
- Bước 2: Điền tên địa chỉ IP của máy chủ, tên đăng nhập và mật khẩu vào và chọn nút ‘Login’. Lưu ý các lỗi sau:

Chuỗi thông báo	Lý do
NAME DOESN'T EXIST	Tên đăng nhập sai
WRONG PASSWORD	Mật khẩu sai

TURN ON SERVER BEFORE START	Ứng dụng trên máy chủ không được khởi động
THIS NAME IS ALREADY USED. PLEASE TRY AGAIN	Tài khoản cần đăng nhập hiện đang trực tuyến

Trong trường hợp chưa có tài khoản, có thể chọn nút ‘SignUp’ để mở cửa sổ ‘SignUp’. Trong cửa sổ này, điền tên tài khoản mới và mật khẩu, nhập lại mật khẩu ở textbox bên cạnh textbox tạo mật khẩu, chọn giới tính rồi nhấn ‘Create Account and Login’. Lưu ý các lỗi sau:

Chuỗi thông báo	Lý do
THIS NAME CONTAINS INVALID CHARACTER. PLEASE TRY AGAIN	Tên tài khoản chứa ký tự lạ
THIS NAME IS ALREADY USED. PLEASE TRY AGAIN	Tên tài khoản đã tồn tại
SERVER HAVE NOT YET READY	Ứng dụng trên máy chủ không được khởi động
PASSWORD MUST HAVE AT LEAST 6 CHARACTER!	Mật khẩu ngắn hơn 6 ký tự
TYPE WRONG CONFIRM PASSWORD !	Mật khẩu tái nhập không trùng khớp
THIS PASSWORD HAVE INVALID CHARACTER. TRY AGAIN	Mật khẩu chứa ký tự lạ

- Bước 3: Chọn tên người dùng muốn trò chuyện trong cửa sổ ‘MainGui’ và nhấn vào nút ‘Chat’ để mời tạo cuộc trò chuyện. Hệ thống sẽ hỏi người được mời có muốn khởi tạo một cuộc trò chuyện riêng tư hay không, nhấn ‘yes’ để bắt đầu hoặc nhấn ‘no’ để từ chối. Nếu được chấp nhận, cửa sổ ‘ChatGui’ sẽ được mở ra. Lưu ý các lỗi sau:

Chuỗi thông báo	Lý do
Invalid UserName	Chưa chọn tên tài khoản để chat
Friend is not found. Please wait to update your list friend	Tên tài khoản bạn chọn để chat không có trong danh sách
This software doesn't support chat yourself function	Lỗi chat với chính mình

- Bước 4: Người dùng gõ nội dung tin nhắn vào textbox và nút  để gửi.

- Bước 5: Người dùng chọn nút  để chọn tập tin đính kèm. Khi cửa sổ chọn tập tin được mở ra, người dùng chọn tập tin cần gửi và nhấn ‘Open’. Chọn



tiếp  để gửi tới người nhận. Về phía người nhận, ứng dụng sẽ thông báo và hỏi người nhận có muốn lưu tập tin vào máy hay không. Nếu muốn, người nhận sẽ chọn ‘Yes’, và chọn địa chỉ lưu tập tin từ hộp thoại vừa mở lên.

- Bước 6: Người dùng chọn ‘LEAVE CHAT’ để thoát khỏi cuộc trò chuyện. Ứng dụng sẽ hiện hộp thoại để xác nhận, chọn ‘Yes’ để dừng cuộc trò chuyện. Cửa sổ ‘ChatGui’ bị đóng lại và người còn lại sẽ nhận  được thông báo.

- Bước 7: Để tạo cuộc trò chuyện nhiều người, nhấn vào  để thêm thành viên vào cuộc trò chuyện. Có thể xóa tất cả thành viên vừa thêm vào bằng nút .

- Bước 8: Chọn  ‘Group’ để tạo cuộc trò chuyện với các thành viên mình vừa chọn. Hệ thống sẽ hỏi người được mời có muốn khởi tạo một cuộc trò chuyện nhóm hay không, nhấn ‘yes’ để bắt đầu hoặc nhấn ‘no’ để từ chối. Nếu được chấp nhận, cửa sổ ‘ChatRoomGUI’ sẽ được mở ra. Lưu ý lỗi sau:

Chuỗi thông báo	Lý do
Group invalid username	Chưa chọn tên tài khoản để chat nhóm

- Bước 9: Chọn ‘Logout’ để đăng xuất khỏi hệ thống. Ứng dụng sẽ hiện hộp thoại để xác nhận, chọn ‘Yes’ để đăng xuất.

## 5. Đánh giá kết quả hiện thực:

### 5.1 Kết quả đạt được:

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

- Ứng dụng có các tính năng đơn giản như: trò chuyện riêng giữa hai người dùng, một lúc đồng thời trò chuyện với nhiều người dùng, gửi tập tin trong quá trình trò chuyện.

- Ứng dụng có tính năng mở rộng: chat nhóm, cuộc trò chuyện với nhiều người cùng tham gia.

- Link source code: <https://tinyurl.com/y3833rub>

### 5.2. Những điều chưa đạt được:

- Mã nguồn còn chưa tối ưu cho ứng dụng.
- Ứng dụng còn có thể thêm các tính năng như: gọi video...

## 6. Tài liệu tham khảo:

### References

- [1] Jim Kurose, Keith Ross, COMPUTER NETWORKING - A Top-Down Approach, New York: Addison-Wesley, 2013.
- [2] G. Coder, "GP Coder," Kiến thức, 26 03 2018. [Trực tuyến]. Available: <https://bit.ly/36C42WP>. [Đã truy cập 28 10 2019].
- [3] S. Barua, "geeksforgeeks.org," webSite, [Trực tuyến]. Available: <https://bit.ly/33ksLNq>. [Đã truy cập 28 10 2019].
- [4] N. L. D. Lai, Computer Networks Lecture, Ho Chi Minh, 2016.

---

Hết