TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN

**Khoa Khoa Học Máy Tính**

Logo

**ĐỒ ÁN CƠ SỞ 4**

**Tìm hiểu và xây dựng ứng dụng hội thoại mô hình Client-Server hoạt động theo giao thức TCP/IP**

Sinh viên thực hiện: **NGUYỄN VĂN A**

Lớp: **19IT**

Giảng viên hướng dẫn: **ThS. NGUYỄN VĂN A**

Đà Nẵng, tháng 09 năm 2021

TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN

**Khoa Khoa Học Máy Tính**

LOGO

**ĐỒ ÁN CƠ SỞ**

**Tìm hiểu và xây dựng ứng dụng hội thoại mô hình Client-Server hoạt động theo giao thức TCP/IP**

Sinh viên thực hiện: **NGUYỄN VĂN A**

Lớp: **19IT**

Giảng viên hướng dẫn: **ThS. NGUYỄN VĂN A**

NHẬN XÉT CỦA GIẢNG VIÊN HƯỚNG DẪN

# 

# 

# 

# 

# 

# 

# 

# 

# 

Đà Nẵng, ngày tháng 12 năm 2021

**Giảng viên hướng dẫn**

**ThS. NGUYỄN VĂN A**

LỜI CẢM ƠN

Để hoàn thành đề tài này, em xin gửi lời cảm ơn chân thành đến thầy Nguyễn Văn A- giảng viên Khoa Khoa học máy tính – Trường Đại học công nghệ thông tin– Người hướng dẫn chính - đã tận tình chỉ bảo và hướng dẫn em hoàn thành đồ án.

Trong quá trình làm đồ án còn nhiều thiếu sót, em rất mong nhận được sự chỉ bảo, đóng góp ý kiến của các thầy cô để em có điều kiện bổ sung, khắc phục những hạn chế của bài đồ án cơ sở này.

Em xin trân trọng cảm ơn !

*Sinh viên*

# **Nguyễn Văn A**

MỤC LỤC

[DANH MỤC CÁC TỪ VIẾT TẮT 6](#_Toc67480301)

[DANH MỤC HÌNH VẼ 7](#_Toc67480302)

[MỞ ĐẦU 8](#_Toc67480304)

[1. Giới thiệu 8](#_Toc67480305)

[2. Mục tiêu của đề tài 8](#_Toc67480306)

[3. Phương pháp triển khai đề tài 8](#_Toc67480307)

[4. Kết cấu của đồ án 8](#_Toc67480308)

[Chương I. CƠ SỞ LÝ THUYẾT … 9](#_Toc67480309)

[I.1 Tổng quan về lập trình mạng 9](#_Toc67480310)

[I.1.1 Giới thiệu giao thức TCP/IP 10](#_Toc67480311)

[I.1.2. Giới thiệu về mô hình Client-Server 12](#_Toc67480312)

[I.2 Tổng quan về ngôn ngữ Java 13](#_Toc67480313)

[I.2.1 Giới thiệu 13](#_Toc67480314)

[I.2.2 Những đặc điểm cơ bản của Java 14](#_Toc67480315)

[I.2.3 Cơ chế Socket trong Java 14](#_Toc67480315)

[Chương II. PHÂN TÍCH THIẾT KẾ HỆ THỐNG… 17](#_Toc67480317)

[II.1. Phân tích yêu cầu bài toán 17](#_Toc67480318)

II.[2. Thiết kế mô hình hệ thống 18](#_Toc67480319)

[II.2.1 Xây dựng biểu đồ 19](#_Toc67480320)

[Chương III. THIẾT KẾ VÀ XÂY DỰNG ỨNG DỤNG … 21](#_Toc67480321)

[III.1 . Kết quả thực hiện 21](#_Toc67480322)

[Hình 7: Màn hình Đăng nhập, đăng ký tài khoản 25](#_Toc67480323)

[Hình 8: Chức năng gửi tin nhắn 26](#_Toc67480324)

[Hình 9: Chức năng gửi nhãn dán 27](#_Toc67480323)

[Hình 10: Chức năng gửi file 27](#_Toc67480324)

[Hình 11: Chức năng gửi ảnh và phóng xem ảnh 28](#_Toc67480323)

[Chương IV. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN 29](#_Toc67480325)

[IV.1. Kết luận 29](#_Toc67480326)

[IV.2. Hướng phát triển 29](#_Toc67480327)

[DANH MỤC TÀI LIỆU THAM KHẢO 30](#_Toc67480329)

# **DANH MỤC CÁC TỪ VIẾT TẮT**

|  |  |
| --- | --- |
| **VIẾT TẮT** | **NỘI DUNG** |
| TCP | Transmission Control Protocol |
| IP | Internet Protocol |

# **DANH MỤC HÌNH VẼ**

[Hình 1: Mô hình lập trình mạng 9](#_Toc60123646)

[Hình 2: Sơ đồ TCP/IP 10](#_Toc60123647)

[Hình 3: Mô hình Client-Server 12](#_Toc60123648)

[Hình 4: Mô hình truyền tin Socket 19](#_Toc60123649)

[Hình 5: Biểu đồ Use case người dùng hệ thống 23](#_Toc60123646)

[Hình 6: Biểu đồ Hoạt động đăng nhập 24](#_Toc60123647)

[Hình 7: Màn hình Đăng nhập, đăng ký tài khoản 25](#_Toc60123648)

[Hình 8: Chức năng gửi tin nhắn 26](#_Toc60123649)

[Hình 9: Chức năng gửi nhãn dán 27](#_Toc60123648)

[Hình 10: Chức năng gửi file 27](#_Toc60123649)

[Hình 11: Chức năng gửi ảnh và phóng xem ảnh 2](#_Toc60123648)8

# **MỞ ĐẦU**

1. Giới thiệu

Ngày này khi nói đến phát triển các ứng dụng phần mềm, đa số là người ta muốn nói đến chương trình có khả năng làm việc trong môi trường mạng tích hợp nói chung và mạng máy tính nói riêng. Từ các chương trình kế toán doanh nghiệp, quản lý, trò chơi, điều khiển... đều là các chương trình ứng dụng mạng.

Nhu cầu về công nghệ thông tin trong đời sống là đa dạng. Việc mở rộng các hệ thống truyền thông và ngày có nhiều máy vi tính kết nối vào mạng Internet. Với việc ứng dụng giao thức TCP/IP làm cho hệ thống mạng ngày càng rộng hơn và phát triển vượt bật. Vấn đề đề an ninh, bảo mật,…là một thế mạnh của giao thức này đem lại cho công nghệ truyền thông.

Bên cạnh việc phát triển của thư điện tử bằng nhiều dịch vụ khác nhau (gmail, yahoo, MSM trong mạng điện thoại di động…), việc nhắn tin và gửi flie trực tiếp trên Internet cũng là nhu cầu không thể thiếu.

Với thực tế như vậy, em thực hiện đề tài “**Tìm hiểu và xây dựng ứng dụng hội thoại mô hình Client-Server hoạt động theo giao thức TCP/IP**”.

2. Mục tiêu của đề tài

- Tìm hiểu các khái niệm về giao thức TCP/IP và Socket.

- Xây dựng ứng dụng hội thoại dựa trên giao thức TCP/IP và Socket.

3. Phương pháp triển khai đề tài

• Ngôn ngữ lập trình: Java.

• Chương trình soạn thảo và build : Eclipse.

• Sử dụng giao thức TCP/IP.

• Sử dụng Socket trong Java.

4. Kết cấu của đồ án

Chương 1: Cơ sở lý thuyết

Chương 2. Thiết kế và xây dựng hệ thống

Chương 3. Kết quả và hướng phát triển

# 

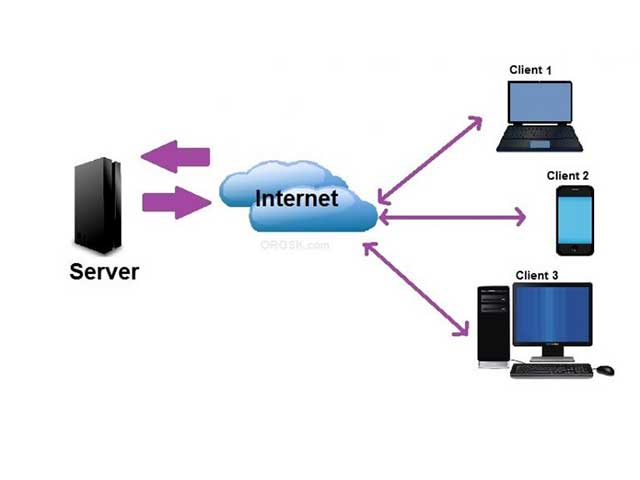
# **CHƯƠNG I: CƠ SỞ LÝ THUYẾT**

## I.1 Tổng quan về lập trình mạng

Mạng máy tính được phát triển nhằm mục đích nghiên cứu và phân tích quá trình giao tiếp. Nó hỗ trợ giao tiếp các hệ thống máy tính với nhau để trao đổi thông tin và tài nguyên. Cho nên, để thực hiện việc kết nối mạng bạn cần phải có một mạng riêng của nó.

Khi nói đến phát triển các ứng dụng phần mềm, đa số là người ta muốn nói đến chương trình có khả năng làm việc trong môi trường mạng tích hợp nói chung và mạng máy tính nói riêng. Từ các chương trình kế toán doanh nghiệp, quản lý, trò chơi, điều khiển…

Vấn đề lập trình mạng liên quan đến nhiều lĩnh vực kiến thức khác nhau. Từ kiến thức sử dụng ngôn ngữ lập trình, phân tích thiết kế hệ thống, kiến thức hệ thống mạng, mô hình xây dựng chương trình ứng dụng mạng, kiến thức về cơ sở dữ liệu… cho đến kiến thức truyền thông, các kiến thức các lĩnh vực liên quan khác như mạng điện thoại di động, PSTN, hệ thống GPS, các mạng như BlueTooth, WUSB, mạng sensor… đều là các chương trình ứng dụng mạng.



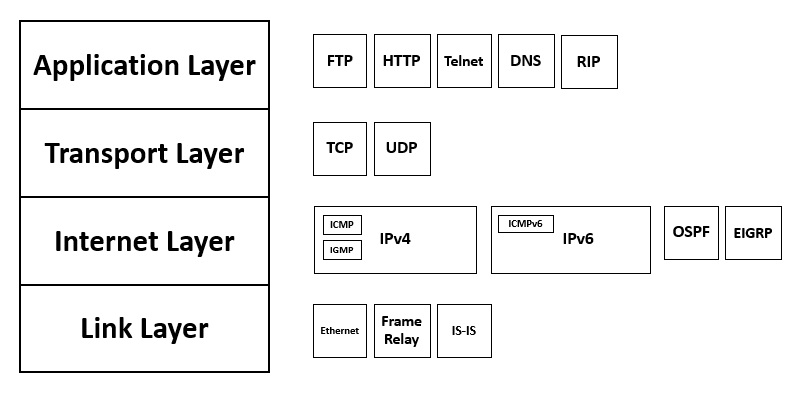
*Hình 1: Mô hình lập trình mạng*

**I.1.1 Giới thiệu giao thức TCP/IP**

TCP/IP là tên chung cho một tập hợp hơn 100 giao thức được sử dụng để kết nối các máy tính vào mạng, trong đó hai giao thức chính là TCP (Transmission Control Protocol) và IP (Internet Protocol).

Trong phạm vi Internet, thông tin không được truyền tải như một dòng riêng biệt từ máy tính này tới máy tính khác. Thay vào đó, dữ liệu được chia thành những gói nhỏ gọi là packet.

Các packet này được gửi trên mạng máy tính. Công việc của IP là chuyển chúng đến các máy tính ở xa. Tại trạm cuối, TCP nhận các packet và kiểm tra lỗi. Nếu một lỗi xuất hiện, TCP yêu cầu gói riêng biệt đó phải được gửi lại. Chỉ khi tất cả các packet đã nhận được là đúng, TCP sẽ sử dụng số thứ tự để tạo lại thông tin ban đầu.

**

*Hình 2: Sơ đồ TCP/IP*

- **Tầng truy cập mạng(Link Layer)**: Cung cấp giao diện tương tác với mạng vật lý. Format dữ liệu cho bộ phận truyền tải trung gian và tạo địa chỉ dữ liệu cho các tiểu mạng dựa trên địa chỉ phần cứng vật lý. Cung cấp việc kiểm tra lỗi trong quá trình truyền dữ liệu.

- **Tầng Internet(Internet Layer)**: Cung cấp địa chỉ logic, độc lập với phần cứng, để dữ liệu có thể lướt qua các tiểu mạng có cấu trúc vật lý khác nhau. Cung cấp chức năng định tuyến để giao lưu lượng giao thông và hỗ trợ việc vận chuyển liên mạng. Thuật ngữ liên mạng được dùng để đề cập đến các mạng rộng lớn hơn, kết nối từ nhiều LAN. Tạo sự gắn kết giữa địa chỉ vật lý và địa chỉ logic.

- **Tầng vận chuyển(Transport Layer)**: Giúp kiểm soát luồng dữ liệu, kiểm tra lỗi và xác nhận các dịch vụ cho liên mạng. Đóng vai trò giao diện cho các ứng dụng mạng.

- **Tầng ứng dụng(Application Layer)**: Cung cấp các ứng dụng để giải quyết sự cố mạng, vận chuyển file, điều khiển từ xa, và các hoạt động Internet. Đồng thời hỗ trợ Giao diện Lập trình Ứng dụng (API) mạng, cho phép các chương trình được thiết kế cho một hệ điều hành nào đó có thể truy cập mạng.

# **I.1.1.1 Giao thức TCP (Transmission Control Protocol)**

Giao thức TCP (Transmission Control Protocol) là giao thức hướng kết nối (connection-oriented), nó đòi hỏi thiết lập kết nối trước khi bắt đầu gửi dữ liệu và kết thúc kết nối khi việc gửi dữ liệu hoàn tất theo đúng thứ tự: thiết lập kết nối, truyền dữ liệu và kết thúc kết nối. Giao thức này đảm bảo chuyển giao dữ liệu tới nơi nhận một cách đáng tin cậy và đúng thứ tự. TCP còn phân biệt giữa dữ liệu của nhiều ứng dụng (chẳng hạn, dịch vụ Web và dịch vụ thư điện tử) đồng thời chạy trên cùng một máy chủ.

**I.1.1.2 Giao thức IP (Internet Protocol)**

IP là địa chỉ của một máy tính trên mạng, dựa vào địa chỉ IP giao thức TCP có thể truyền dữ liệu chính xác từ một máy này qua máy kia thông qua hệ thống mạng. Ở trên mạng, một máy tính sẻ có một địa chỉ IP khác nhau, từ địa chỉ IP có thể biết được máy nào trên mạng và ngược lại.

Dữ liệu trong một liên mạng IP được gửi theo các khối được gọi là các gói (packet hoặc datagram). Cụ thể, IP không cần thiết lập các đường truyền trước khi một máy chủ gửi các gói tin cho một máy khác mà trước đó nó chưa từng liên lạc với.

Giao thức IP cung cấp một dịch vụ gửi dữ liệu không đảm bảo, nghĩa là nó hầu như không đảm bảo gì về gói dữ liệu.

Gói dữ liệu có thể đến nơi mà không còn nguyên vẹn, nó có thể đến không theo thứ tự (so với các gói khác được gửi giữa hai máy nguồn và đích đó), nó có thể bị trùng lặp hoặc bị mất hoàn toàn. Nếu một phần mềm ứng dụng cần được bảo đảm, nó có thể được cung cấp từ nơi khác, thường từ các giao thức giao vận nằm phía trên IP.

Địa chỉ IP được chia thành 4 số giới hạn từ 0 - 255. Mỗi số được lưu bởi 1 byte - > IP có kích thước là 4byte, được chia thành các lớp địa chỉ. Có 3 lớp là A, B, và C. Nếu ở lớp A, ta sẽ có thể có 16 triệu điạ chỉ, ở lớp B có 65536 địa chỉ. Ví dụ: Ở lớp B chúng ta có tất cả các địa chỉ từ 132.25.0.0 đến 132.25.255.255.

**I.1.1.3 Cổng Port**

Với IP, giao thức TCP chỉ mới có thể truyền dữ liệu chính xác từ máy này qua máy kia mà chưa thể truyền chính xác đến từng ứng dụng trên máy được. Hiện nay, các hệ thống máy thông thường hoạt động theo chế độ đa nhiệm, nghĩa là có nhiều ứng dụng chạy cùng một lúc và trong đó có thể có nhiều ứng dụng sử dụng dịch vụ mạng. Yêu cầu, khi máy chủ A truyền dữ liệu cho một ứng dụng u trên máy B thì trên máy B phải đảm bảo dữ liệu đó phải đến được ứng dụng u, chứ không phải ứng dụng v.

Để thực hiện điều đó thì máy chủ A khi truyền dữ liệu đi thì trên dữ liệu đó có một thành phần thông tin giúp máy B xác định được đúng ứng dụng u. Phần thông tin đó chính là địa chỉ port trên máy B, nó có thể hiểu là lỗ cắm ảo trên máy B mà ứng dụng u đã đăng ký để độc quyền sử dụng nhằn nhận dữ liệu từ máy chủ A.

Trên thực tế, địa chỉ port là một số nguyên 2 byte có giá trị từ 0 đến 65535. Nó có đặc điểm :

- Giá trị từ 0 đến 1023 là các cổng phổ biến dành cho các ứng dụng thông dụng như http: 80, mail : 25, ftp : 21, Telnet 23.... Các giá trị còn lại có thể được sử dụng linh hoạt.

- Mỗi cổng trong mỗi thời điểm được sử dụng cho tối đa là 1 ứng dụng. Ví dụ như một ứng dụng nào đó đã sử dụng cổng 55 thì ứng dụng khác không thể sử dụng cổng 55 đó nữa chừng nào ứng dụng trước đó chưa có đóng cổng 55 lại.

**I.1.2 Giới thiệu mô hình Client – Server**

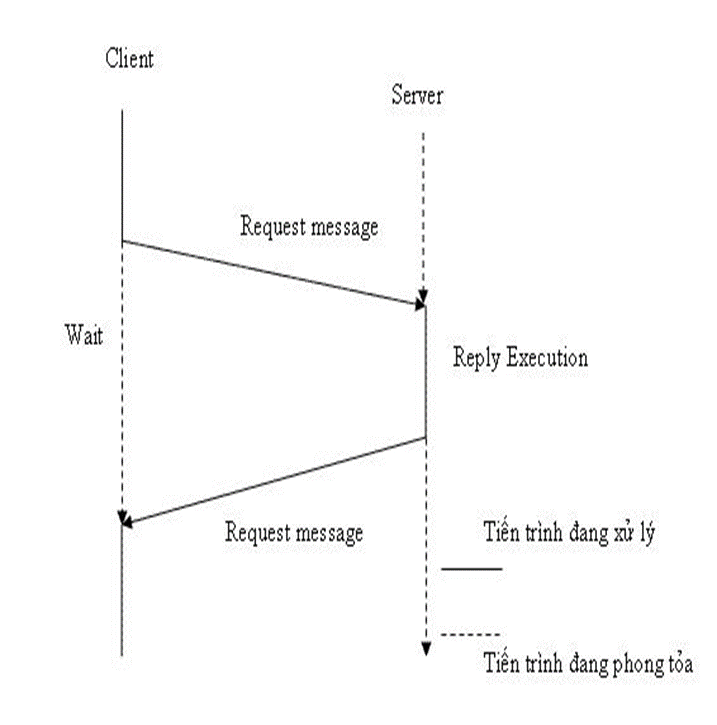
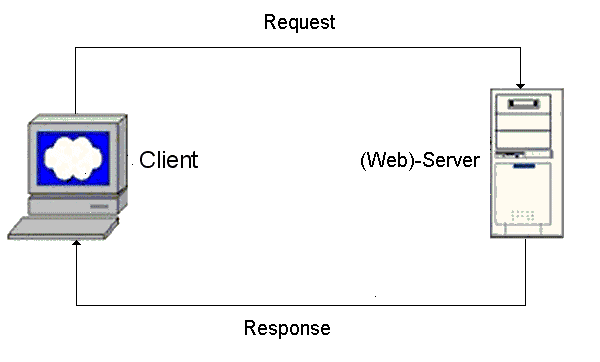
Server được hiểu là máy chủ, thường là một hệ thống máy lớn, có bộ xử lý mạnh, có khả năng hoạt động đáng tin cậy, có khả năng lưu trữ dữ liệu lớn, nó chuyên quản lý tài nguyên (chủ yếu là cơ sở dữ liệu), cung cấp các dịch vụ mạng cho các máy khách (client) sử dụng. Bình thường nó chạy suốt thời gian thực và sẵn sàng chấp nhận các yêu cầu kết nối và các yêu cầu dịch vụ khác từ máy khách.

Client là máy khách, nó thường được sử dụng bởi người dùng cuối. Nó hoạt động dựa trên việc sử dụng dịch vụ mà máy server cung cấp để thực hiện các công việc mà người dùng cuối mong muốn.

Quy trình hoạt động của mô hình này lặp lại 2 quá trình như sau :

- Client gửi yêu cầu lên server.

- Server nhận được yêu cầu thì sẻ xử lý thích hợp và trả lời lại client.



*Hình 3: Mô hình Client-Server*

Quá trình giao tiếp client và server có thể diễn ra theo một trong hai chế độ: bị phong tỏa (blocked) và không bị phong tỏa (non-blocked).

**Chế độ bị phong tỏa** (blocked):

Trong chế độ bị phong tỏa, khi tiến trình client hoặc server phát ra lệnh gửi dữ liệu (send), việc thực thi của tiến trình sẽ bị tạm ngừng cho tới khi tiến trình nhận phát ra lệnh nhận dữ liệu (receive).

Tương tự đối với tiến trình nhận dữ liệu, nếu tiến trình nào đó (client hoặc server) phát ra lệnh nhận dữ liệu, mà tại thời điểm đó chưa có dữ liệu gửi tới thì việc thực thi của tiến trình cũng sẽ bị tạm ngừng cho tới khi có dữ liệu gửi tới.

**Chế độ không bị phong tỏa** (non-blocked):

Trong chế độ này, khi tiến trình client hay server phát ra lệnh gửi dữ liệu thực sự, việc thực thi của tiến trình vẫn được tiến hành mà không quan tâm đến việc có tiến trình nào phát ra lệnh nhận dữ liệu đó hay không.

**I.2 Tổng quan về ngôn ngữ Java**

**I.2.1 Giới thiệu**

Java là một trong những ngôn ngữ lập trình hướng đối tượng. Nó được sử dụng trong phát triển phần mềm, trang web, game hay ứng dụng trên các thiết bị di động. Java được tạo ra với tiêu chí “Viết (code) một lần, thực thi khắp nơi” (Write Once, Run Anywhere – WORA). Chương trình phần mềm viết bằng Java có thể chạy trên mọi nền tảng (platform) khác nhau thông qua một môi trường thực thi với điều kiện có môi trường thực thi thích hợp hỗ trợ nền tảng đó.

Ngày nay Java được sử dụng với các mục đích sau:

• Phát triển ứng dụng cho các thiết bị điện tử thông minh, các ứng dụng cho doanh nghiệp với quy mô lớn.

• Tạo các trang web có nội dung động (web applet), nâng cao chức năng của server.

• Phát triển nhiều loại ứng dụng khác nhau: Cơ sở dữ liệu, mạng, Internet, viễn thông, giải trí,...

**I.2.2 Những đặc điểm cơ bản của Java**

• Đơn giản và quen thuộc

• Hướng đối tượng và quen thuộc

• Mạnh mẽ (thể hiện ở cơ chế tự động thu gom rác) và an toàn

• Kiến trúc trung lập, độc lập nền tảng và có tính khả chuyển

• Hiệu suất cao

• Phân tán

• Đa nhiệm

**I.2.3 Cơ chế Socket trong java**

**I.2.3.1 Giới thiệu về Socket**

Socket là một khái niệm để định vị một dịch vụ của một máy tính trên mạng khi kết hợp hai khái niệm IP và Port lại.

Để có thể nói chuyện được trên TCP thì chương trình client và chương trình server phải thiếp lập một đường truyền, và mỗi chương trình sẽ phải kết nối lại với socket là điểm cuối để kết nối, client và server muốn nói chuyện với nhau thì sẽ phải thông qua socket, mọi thông điệp sẽ phải đi qua socket. Chúng ta cứ mường tượng socket ở đây là một cái cửa mọi người muốn đi ra hay đi vào đều phải thông qua cái cửa này.

**I.2.3.2 Cơ chế Socket**

Một socket là một điểm cuối của thông tin hai chiều liên kết giữa hai chương trình đang chạy trên mạng. Những lớp socket được dùng để đại diện cho kết nối giữa một chương trình client và một chương trình server. Trong Java gói **com.corundumstudio.socketio** cung cấp hai lớp **Socket** và **SocketIOServer** để thực hiện kết nối giữa client và server.

* Socket có những nhiệm vụ như sau :

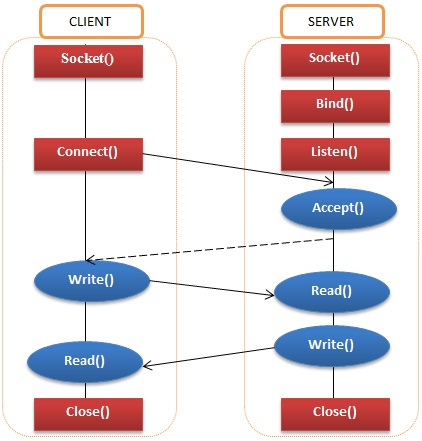
+ Gắn một cổng trên máy.

+ Lắng nghe các kết nổi.

+ Kết nối/Đóng kết nối đến máy tính ở xa qua cổng đã gắn.

+ Gửi/nhận dữ liệu.

+ Lắng nghe dữ liệu đến.



*Hình 4: Mô hình truyền tin Socket*

**Gói SocketIOServer:**

* Mở một cổng kết nối trên Server, sẵn sàng hoạt động trên cổng đó.

public SocketIOServer (int port) : Tạo cổng kết nối trên cổng port.

* Lắng nghe, chờ đợi các yêu cầu kết nối trên cổng đó và kết nối khi có yêu cầu.

public onConnect(SocketIOClient sioc):Chấp nhận một kết nối và trả về một đối tượng Socket mô tả nguồn đã được kết nối đến.

* Tạo luồng dữ liệu vào ra để có thể truyền dữ liệu trên cổng đến một socket ở máy khách. (Đây là chức năng quan trọng nhất).

onData () : Trả về một luồng mà có thể nhập dữ liệu vào.

sendEvent() : Trả về một luồng ghi dữ liệu.

* Đóng kết nối lúc không sử dụng.

close() : Đóng kết nối khi không cần

**Gói Socket:**

* Kết nối tới một máy chủ Server thông qua địa chỉ IP và số hiệu cổng trên máy chủ Server đó.

public IO.[socket](file:///F:\Tai%20lieu%20vi%20tinh\Javadoc\api\java\net\Socket.html#Socket(java.net.InetAddress, int))([InetAddress](file:///F:\Tai%20lieu%20vi%20tinh\Javadoc\api\java\net\InetAddress.html) address, int port)

* Tạo luồng dữ liệu vào ra.

public  socket.emit () ;

public  socket.on() ;

* Đóng kết nối Socket.

public void [Close()](file:///F:\Tai%20lieu%20vi%20tinh\Javadoc\api\java\net\InetAddress.html); Đóng kết nối

**Một số gói, lớp bổ trợ khác:**

1. Gói Swing, awt.event: tạo giao diện người dùng.
2. Gói Runnable, Thread: Hỗ trợ đa tiến trình.
3. Gói Java.io: Cung cấp các lớp vào ra dữ liệu.

# **CHƯƠNG II. PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG**

## II.1 Phân tích yêu cầu bài toán

* **Yêu cầu bài toán:** Xây dựng ứng dụng hội thoại mô hình Client-Server hoạt động theo giao thức TCP/IP.
* **Phân tích:**
* **Xây dựng chương trình bên Server:**

+ Tạo một TCP Socket và gắn vào một cổng.

+ Xây dựng một chương trình Server đa tuyến (Threaded Server) để cho phép nhiều Client kết nối tới Server.Mỗi tuyến đảm nhận liên lạc với Client.

+ Luôn luôn chờ và lắng nghe yêu cầu kết nối từ phía Client.

+ Chấp nhận kết nối và tạo ra các Thread(Socket) tương ứng.

+ Truyền nhận thông tin qua các luồng nhận/gửi dữ liệu của Socket.

+ Khi một user kết nối đến Server thì Server sẽ cập nhật User đó vào mảng user đã kết nối và gửi tới các Client đang kết nối.

+ Khi một user gửi thông điệp DISCONNECT đến Server thì Server sẽ gửi thông báo user đó đã thoát tới Client và xóa user đã gửi thông điệp DISCONNECT trong mảng user đã kết nối.

+ Thiết kế giao diện cho Server.

* **Xây dựng chương trình bên Client:**

+ Tạo một TCP Socket với địa chỉ IP và số cổng mà chương trình Server đang chạy

+ Thiết lập kết nối tới Server.

+ Trao đổi dữ liệu với Server.

+ Cập nhật các user khác vừa CONNECT/DISCONNECT.

+ Gửi nhận thông điệp với tất cả các Client khác trong phòng chat.

+ Chức năng xem lại lịch sử cuộc trò chuyện

+ Chức năng up file, download file, gửi emoji, gửi hình ảnh…

+ Thiết kế giao diện hiển thị khung chat phía Client.

## II.2 Thiết kế mô hình hệ thống

* ***Chức năng đăng nhập, đăng ký:***

Khi người dùng đang nhập/đăng ký tài khoản, người dùng nhập thông tin tài khoản và mật khẩu từ bàn phím kích vào nút đăng nhập/đăng ký thì Client sẽ gửi đến Server một đối tượng Messenger gồm tên tài khoản và mật khẩu đã nhập từ bàn phím. Khi Server nhận được đối tượng Messenger từ Client thì Server sẽ truy cập cơ sở dữ liệu và lấy bản ghi thông tin đăng nhập về để so khớp với thông tin mà Client gửi qua nếu khớp thì vào hộp thoại chat nếu không thì thông báo lỗi. hoặc ghi lại thông tin đăng ký lên cơ sở dữ liệu để dùng cho việc đăng nhập.

* ***Chức năng gửi tin nhắn:***

Khi người dùng nhập ký tự từ bàn phím và nhấn phím Enter hoặc kích vào biểu tượng Gửi thì Client sẽ gửi đến Server một đối tượng Messenger gồm tên người gửi và ký tự đã nhập từ bàn phím. Khi Server nhận được đối tượng Messenger từ Client thì Server sẽ nhân bản đối tượng Messenger và gửi đến các Client.

* ***Chức năng gửi nhãn dán:***

Khi người dùng vào biểu tượng mặt cười thì chương trình sẽ hiện thị lên danh sách nhãn dán có trong chương trình và người dùng sẽ click cái những nhãn dán đó thì sẽ gửi đến Server một đối tượng Messenger gồm tên người gửi và một mảng byte được chuyển từ nhãn dán. Khi Server nhận được đối tượng Messenger từ Client thì Server sẽ nhân bản đối tượng Messenger và gửi đến các Client.

* ***Chức năng gửi File:***

Khi người dùng click vào biểu tượng có hình tài liệu thì chương trình sẽ hiển thị cửu sổ FileDialog để người dùng chọn. Khi người dùng chọn file xong thì Client gửi đến Server một đối tượng Messenger gồm tên người gửi, tên file và một mảng byte được chuyển từ file đã chọn. Khi Server nhận được đối tượng Messenger từ Client thì Server sẽ nhân bản đối tượng Messenger và gửi đến các Client.

* ***Chức năng tải File:***

Khi người dùng click vào file đã được chia sẽ hiển thị trên hộp thoại tin nhắn thì chương trình sẽ hiện thị cửa sổ FileDialog để người dùng chọn vị trí để download file. Khi người dùng nhấn Save thì file sẽ được lưu vào vị trị mà người dùng đã chọn.

* ***Chức năng gửi hình ảnh và phóng to xem ảnh:***

Khi người dùng click vào biểu tượng hình ảnh trên giao diện chat thì chương trình sẽ hiển thị cửa sổ FileDialog người dùng bắt buộc phải chọn file có các đuôi PNG, JPG, GIF, BMP… là đuôi liên quan đến tệp hình ảnh, Khi người dùng chọn file ảnh xong thì Client gửi đến Server một đối tượng Messenger gồm tên người gửi, tên file ảnh và một mảng byte được chuyển từ file ảnh đã chọn.

Người dùng ấn vào file ảnh đã được chia sẽ trên hộp thoại chat thì ảnh sẽ được hiển thị ra một hộp thoại khác phóng to gấp 2 lần hình ảnh gốc.

* ***Chức năng hiển thị danh sách online:***

Khi có một người dùng nào đó đăng nhập vào hệ thống thì Server sẽ gửi danh sách người dùng đó đến các Client đang trong hệ thống để hiển thị người dùng mới lên danh sách trực tuyến, ngược lại nếu người dùng nào đó thoát khỏi hệ thống thì danh sách sẽ được Server gửi đến các Client để cập nhật lại một lần nữa

* ***Chức năng cập nhật ảnh đại diện:***

Khi người dùng cần thay đổi ảnh đại diện cho tài khoản của mình thì thì chương trình sẽ hiển thị cửa sổ FileDialog người dùng bắt buộc phải chọn file có các đuôi PNG, JPG, GIF, BMP… là đuôi liên quan đến tệp hình ảnh, Khi người dùng chọn file ảnh xong thì Client gửi đến Server một đối tượng Messenger gồm tên người gửi, tên file ảnh và một mảng byte được chuyển từ file ảnh đã chọn

sau đó server xử lí đẩy mảng byte vào cở sở dữ liệu để lưu trữ, tất cả người dùng khác đăng nhập vào thì sẽ được server gửi lại mảng byte đó sau đó chuyển đổi sang định dạng image để hiển thị

* ***Chức năng gửi voice chat:***

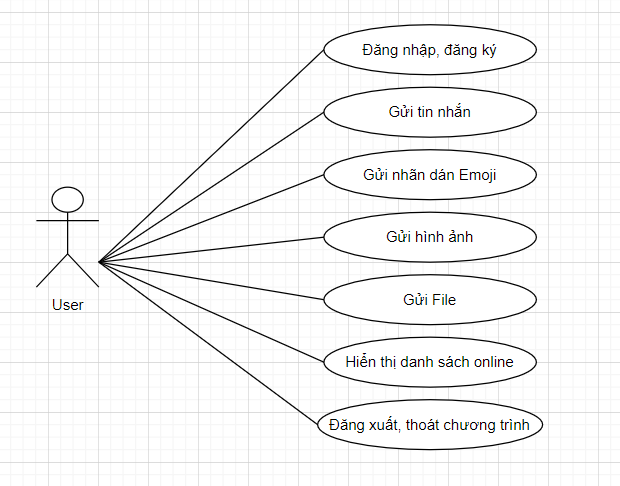
Chức năng gửi voice sử dụng thuật toán để mã hóa giọng nói sang dạng byte array gửi nhiều packet mỗi packet tối đa 2000 mảng byte từ client đến server

sau đó server gửi đến client đích dùng thuật toán mã hóa từ mảng byte sang dạng audio

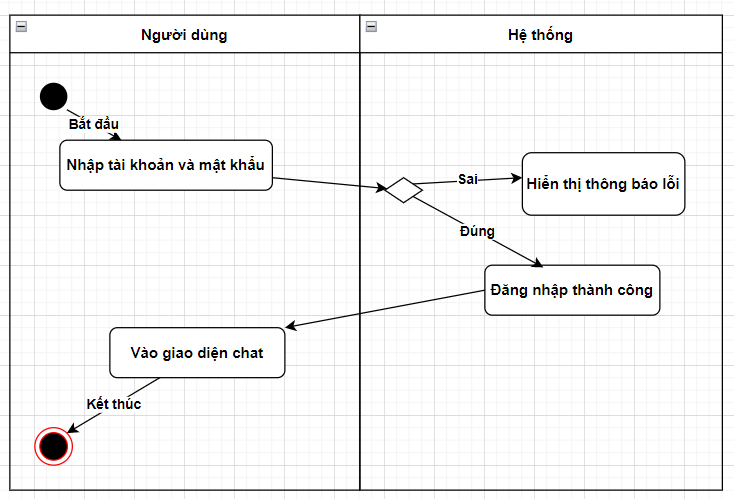
* ***Chức năng đăng xuất, thoát chương trình:***

Khi người dùng đăng xuất thì điều hướng trở lại màn hình đăng nhập, hoặc thoát chương trình.

**II.2.1 Xây dựng biểu đồ**

****

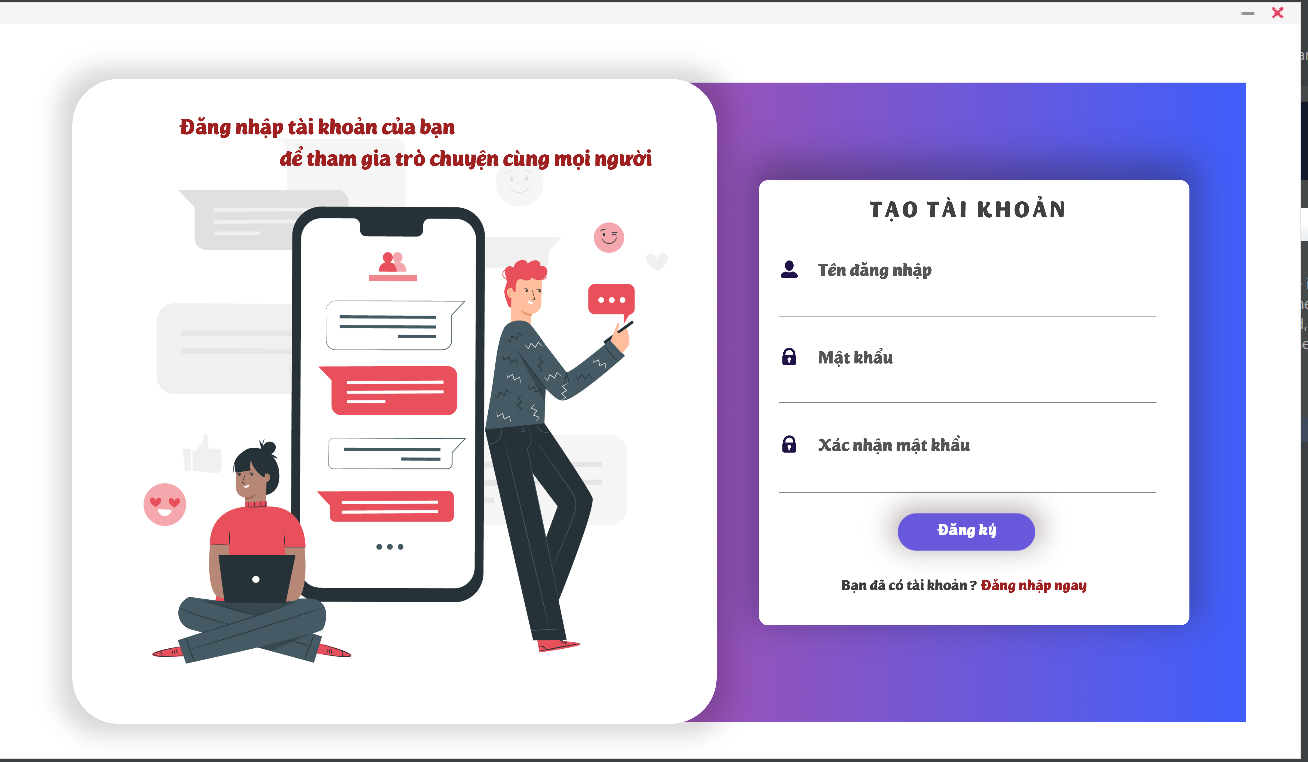
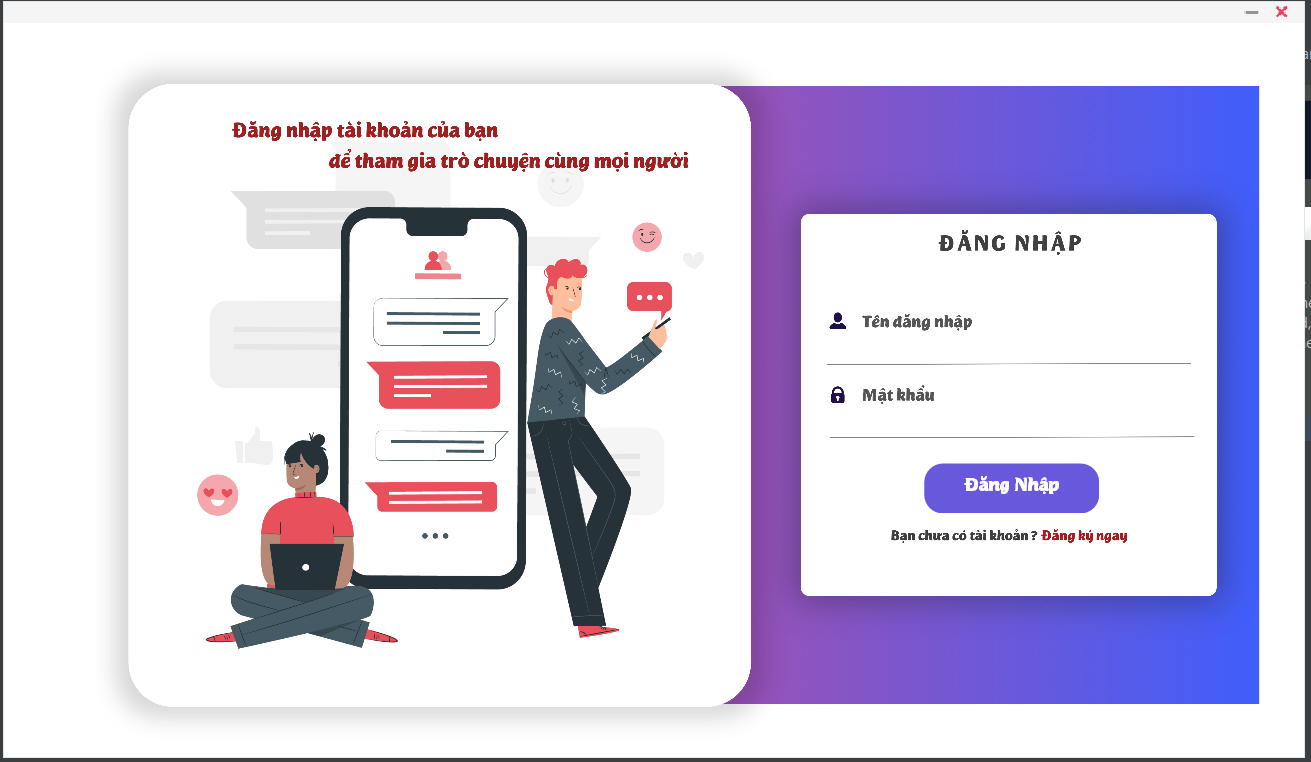
*Hình 5. Biểu đồ Use case người dùng hệ thống*

****

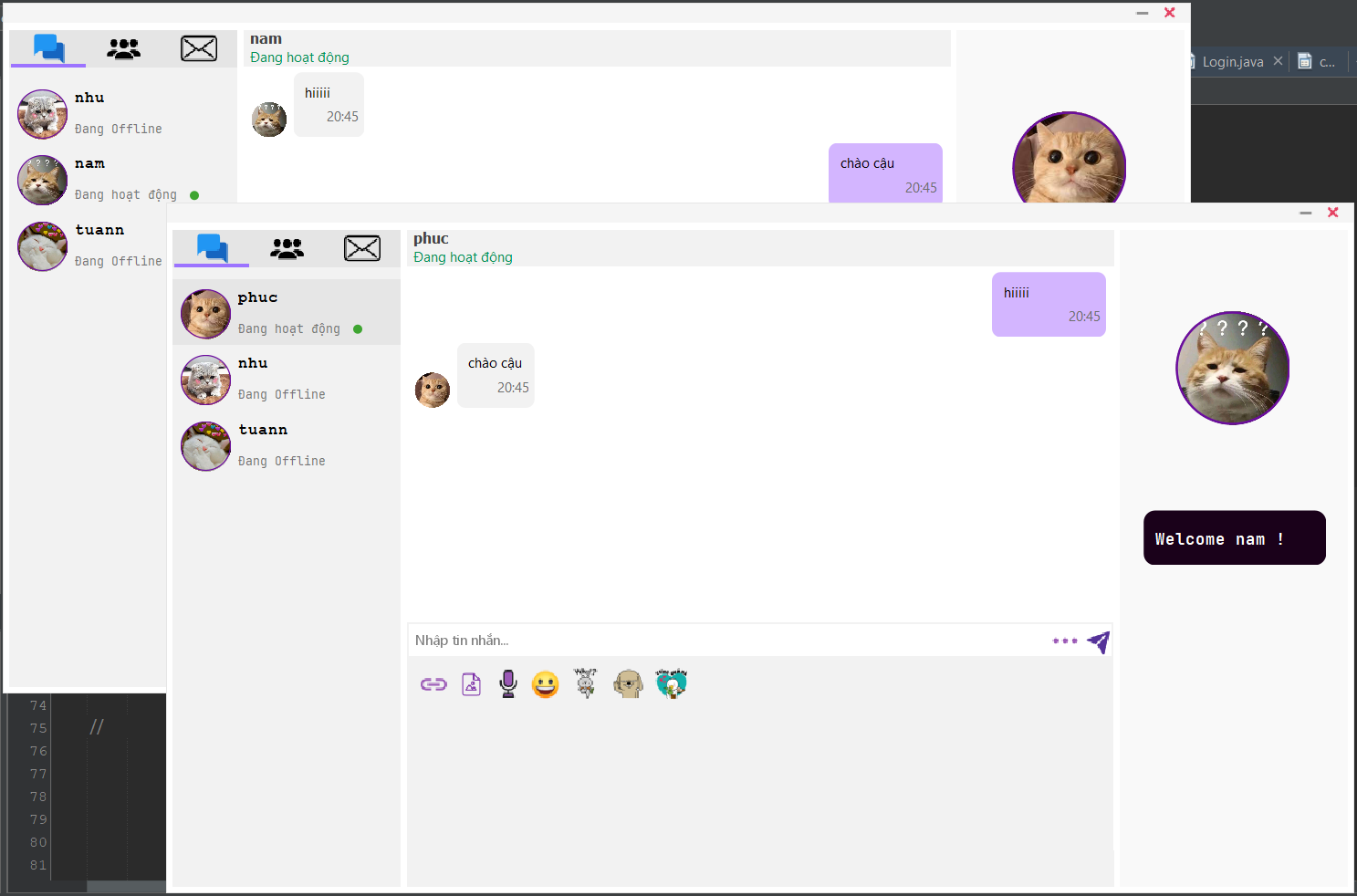
*Hình 6. Biểu đồ hoạt động đăng nhập*

# **CHƯƠNG III. THIẾT KẾ VÀ XÂY DỰNG ỨNG DỤNG**

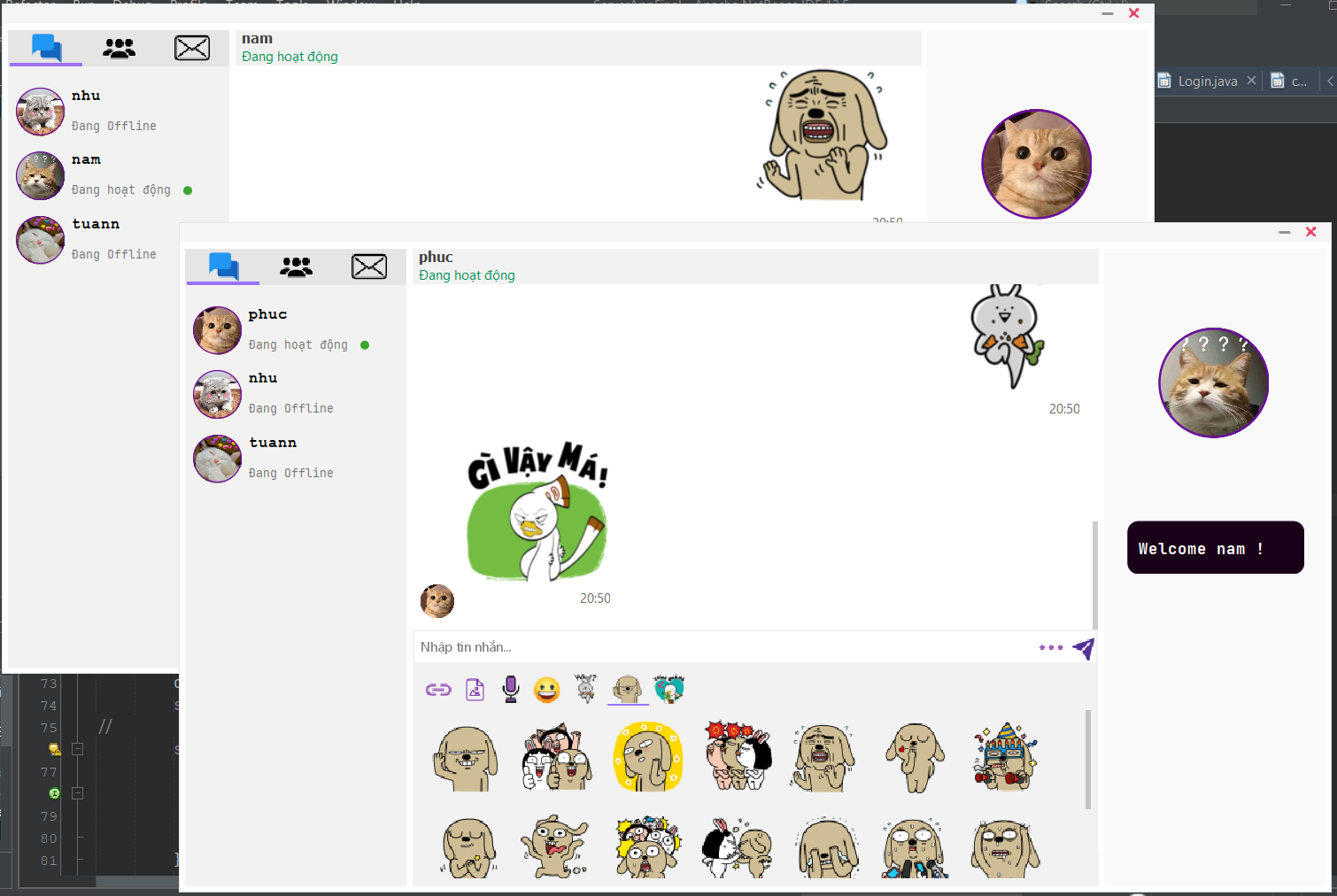
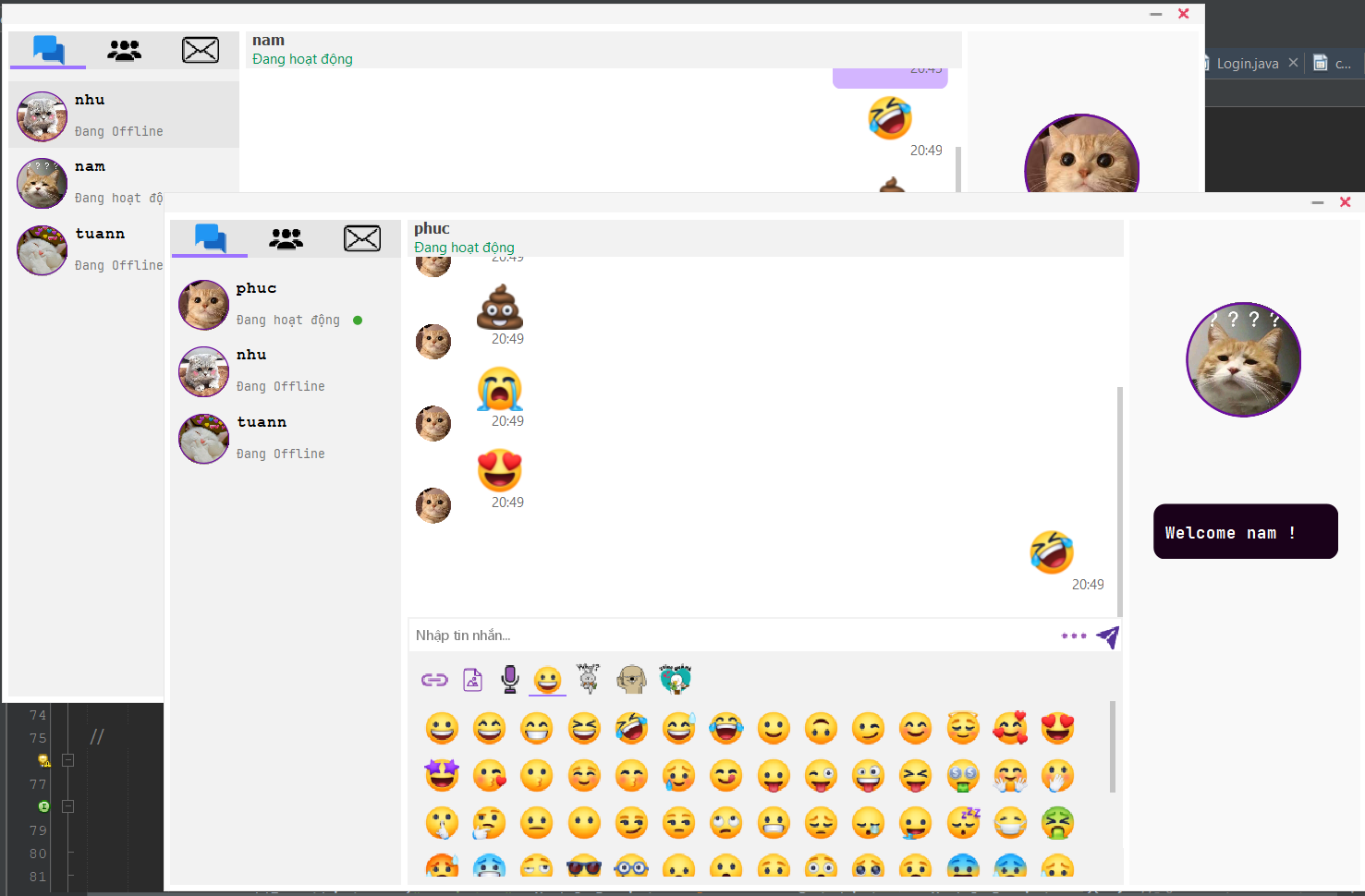
**III.1 Kết quả thực hiện**



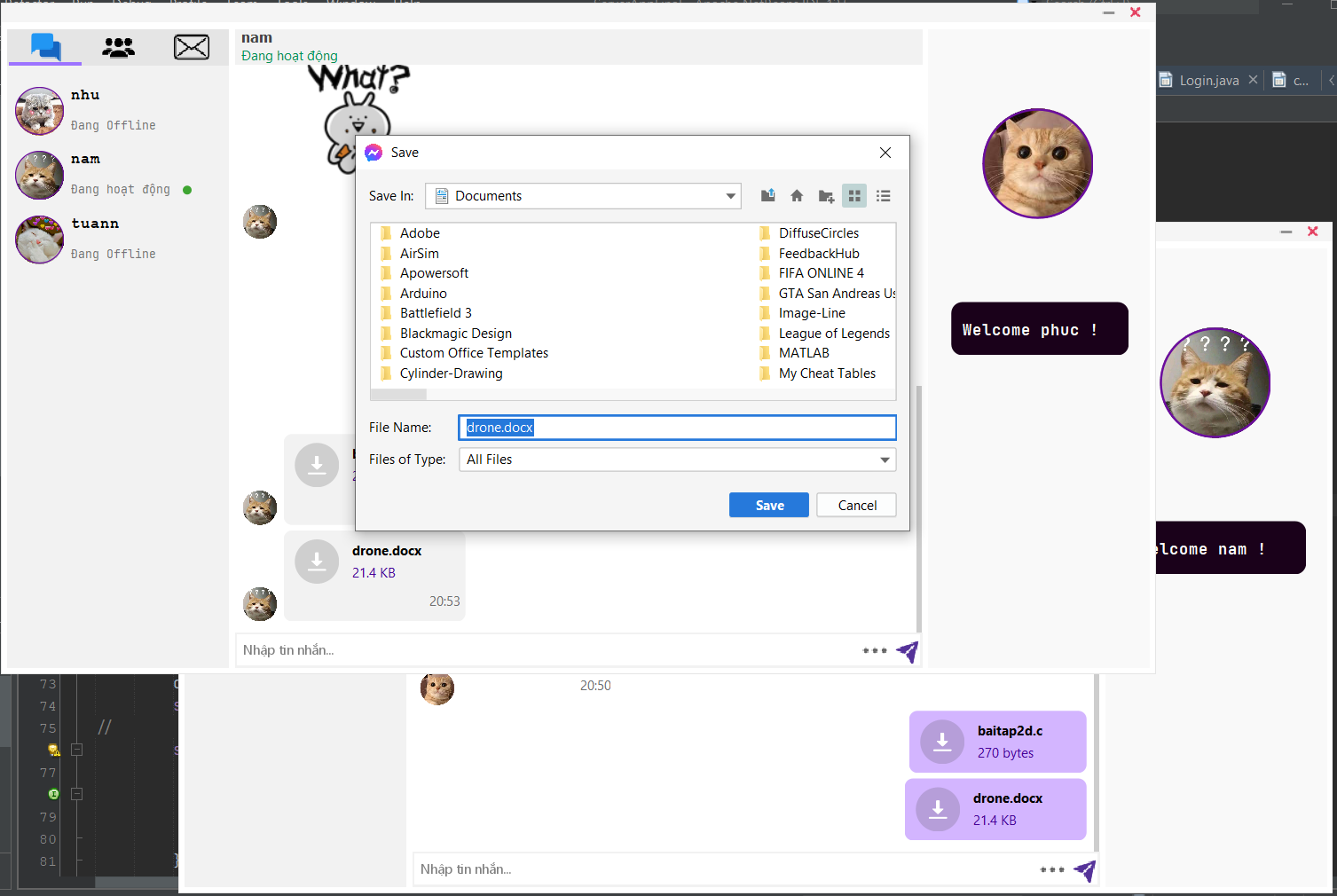
*Hình 7. Màn hình Đăng nhập, đăng ký tài khoản*



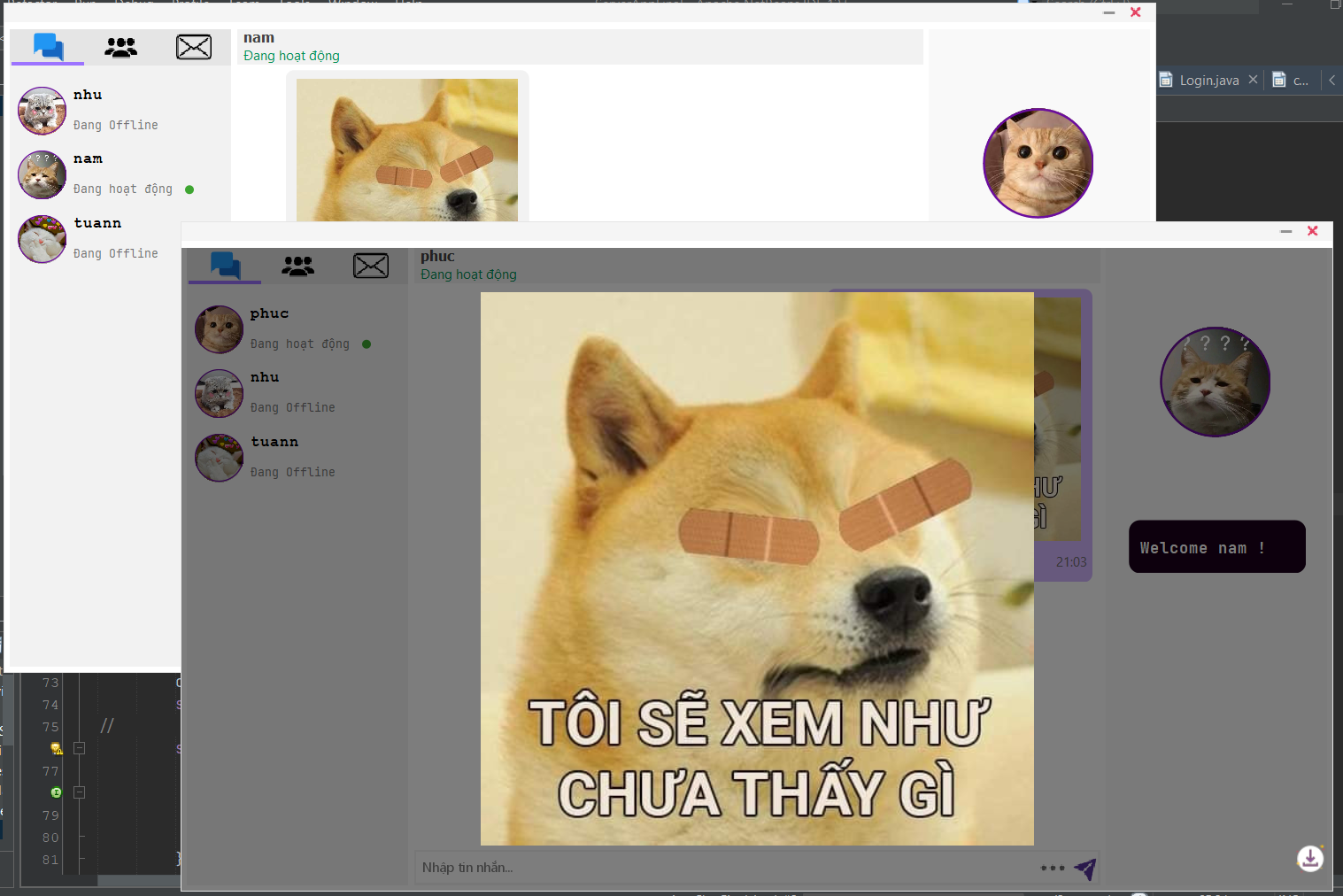
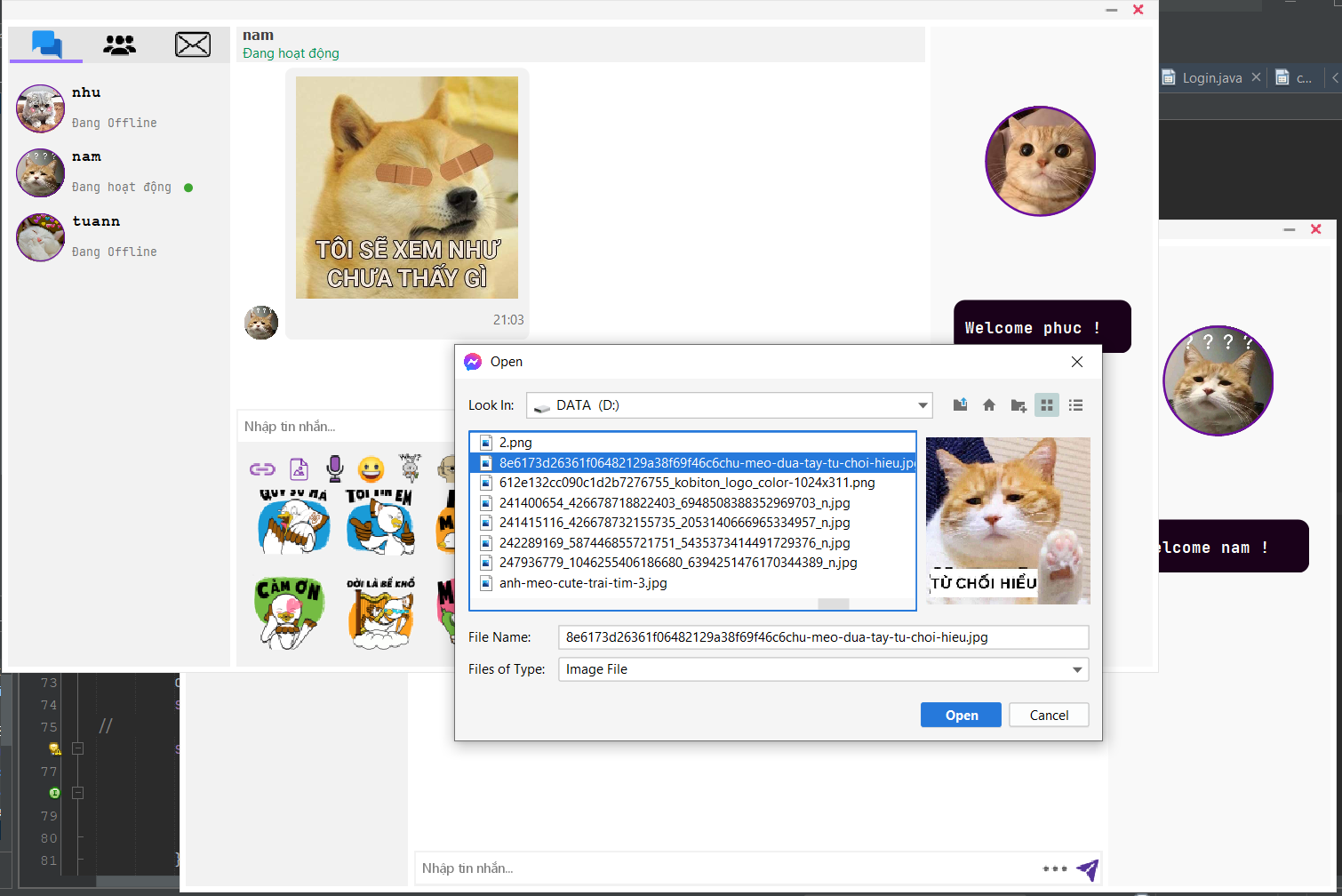
*Hình 8. Chức năng gửi tin nhắn*



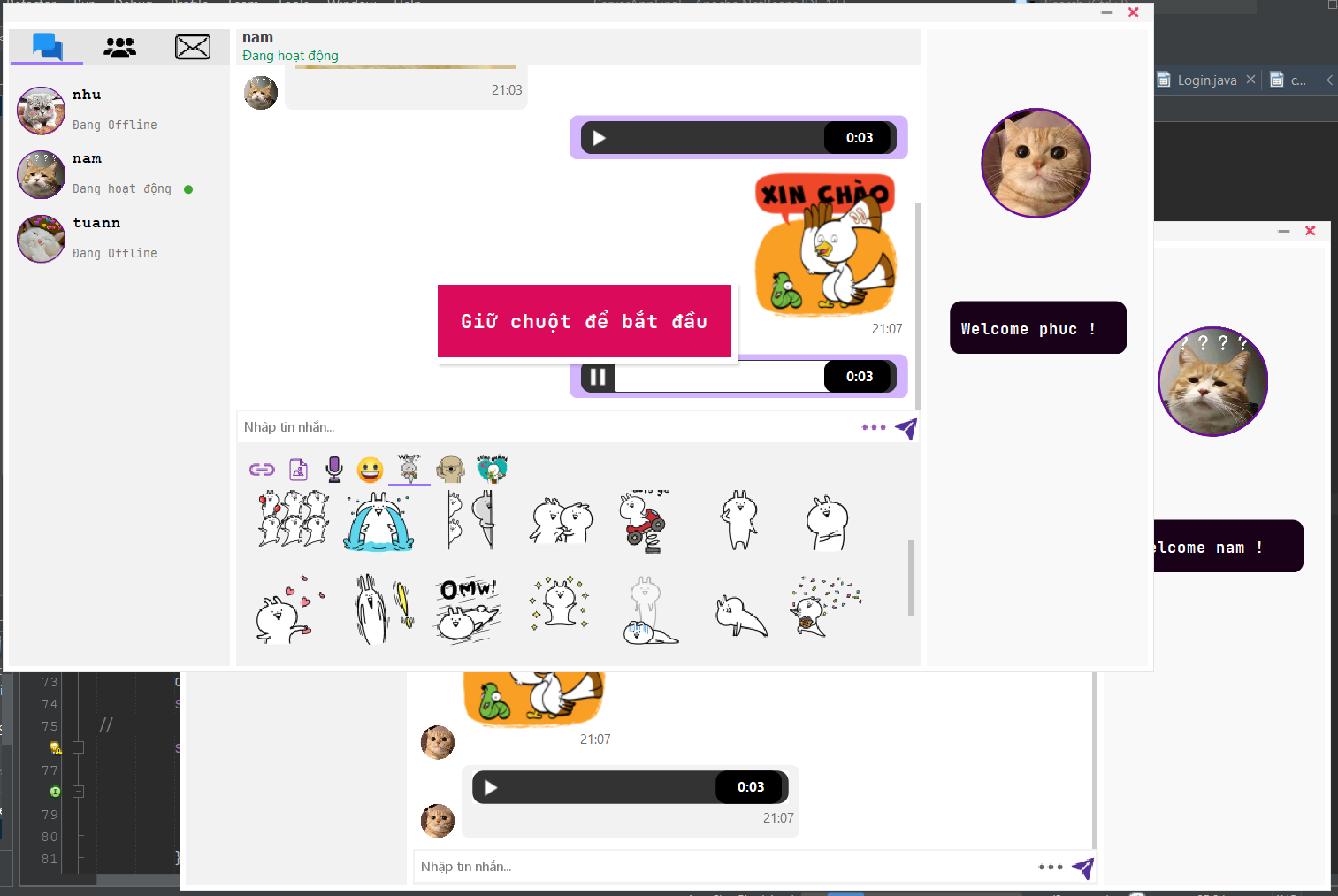
*Hình 9. Chức năng gửi nhãn dán*



*Hình 10. Chức năng gửi file*



*Hình 11. Chức năng gửi ảnh và phóng xem ảnh*



*Hình 12. Chức năng gửi voice chat*

**CHƯƠNG IV: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN**

**IV.1 Kết luận**

Sau một thời gian tập trung triển khai đề tài, em đã hoàn thành được đồ án “**Tìm hiểu và xây dựng ứng dụng hội thoại mô hình Client-Server hoạt động theo giao thức TCP/IP**” với giao diện đẹp, các thao tác sử dụng dẽ dàng, thân thiện với người dùng. Khi hoàn thành xong ứng dụng, em có thêm những kỹ năng nền tảng để xây dựng được một ứng dụng bằng ngôn ngữ Java, áp dụng các kiến thức trong môn lập trình mạng đã được học vào việc phát triển ứng dụng, phần mềm. Bên cạnh đó, cũng mở rộng được kiến thức chuyên ngành mới nhờ việc tìm hiểu một số kiến thức lập trình để áp dụng vào việc phát triển ứng dụng này. Nhờ vậy, kỹ năng tự học và vận dụng các kỹ năng mới của em được nâng cao hơn.

**• Những vấn đề tồn tại**

* **ssssx**Chức năng làm được còn hạn chế.
* Chương trình chat còn đơn giản.
* Hệ thống xử lí nghiệp vụ chưa tối ưu
* Chỉ thực hiện được việc kiểm tra trong kết nối giữa Client-Server.

**IV.2 Hướng phát triển**

* Hoàn thiện giao diện đẹp hơn.
* Xây dựng hệ cơ sở dữ liệu lớn hơn trong việc lưu trữ thông tin lịch sử trò chuyện của người dùng.
* Phát triển call video, và thêm một vài chức năng như thu hồi tin nhắn, tìm kiếm tin nhắn…
* Phát triển trên mạng rộng ngoài LAN.

**DANH MỤC TÀI LIỆU THAM KHẢO**

1. 2013-Advanced Network Programming - Principles and Techniques-1
2. <https://www.youtube.com/watch?v=8e-Q5KeVTPw&t=871s>

**x**

x