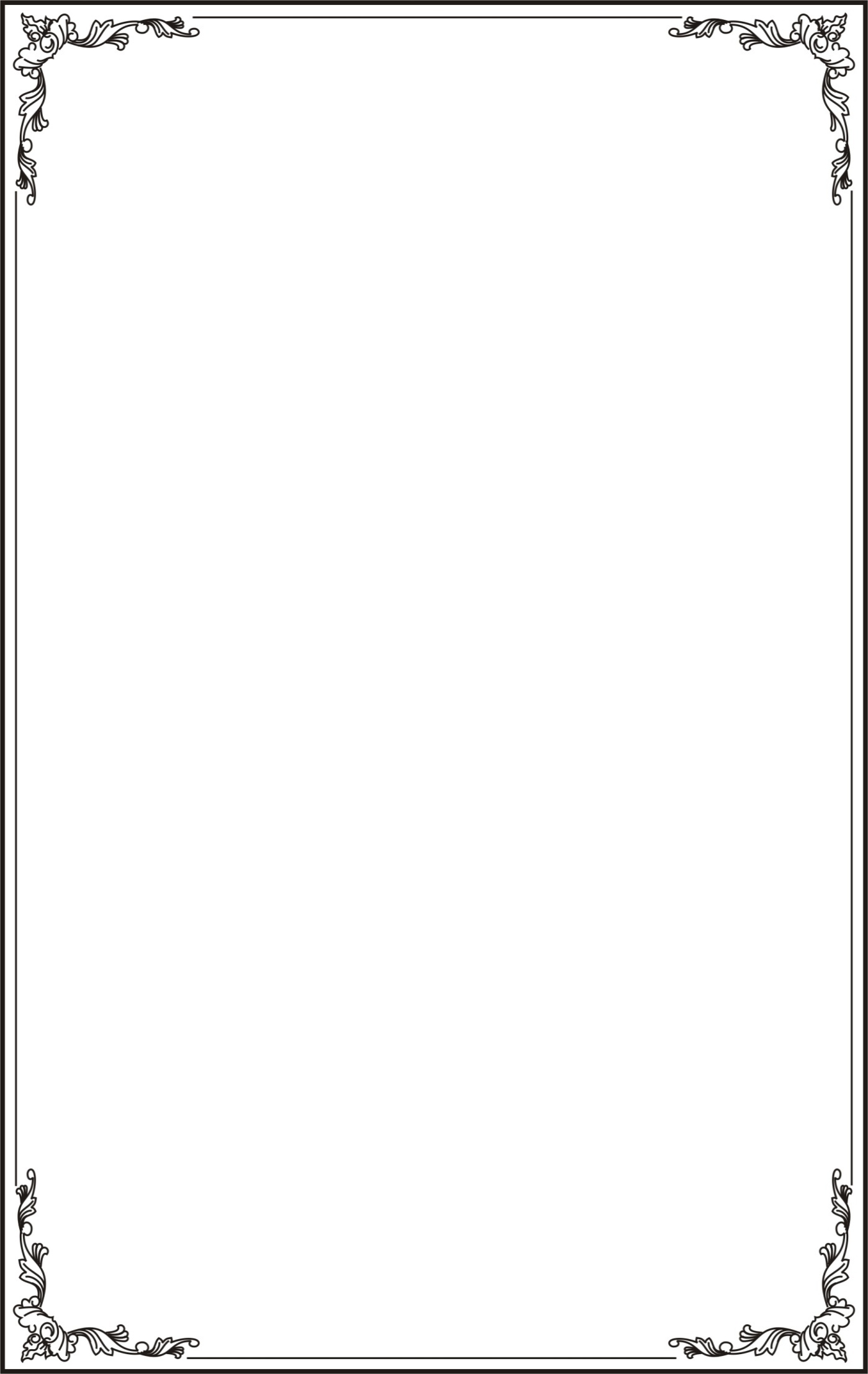
****TRƯỜNG ĐẠI HỌC KIÊN GIANG

**KHOA THÔNG TIN & TRUYỀN THÔNG**

--------------------------------



**BÁO CÁO**

**MÔN HỌC LẬP TRÌNH MẠNG**

**Đề tài:**

**VIẾT CHƯƠNG TRÌNH GROUP CHAT, VOICE CHAT + JAVA**

|  |
| --- |
| Nhóm 1 |
| Lâm Thiện Tính – 2006206072 |
| Quách Trùng Vương - 2006206091 |
| Trần Đào Quốc Đạt - 2006206098 |
| Võ Tuấn Kiệt - 2006206101 |

GVHD: Nhan Thanh Nhã

**LỜI CẢM ƠN**

**------------------------------------------------------------------**

Chúng em xin chân thành cảm ơn Khoa Công Nghệ Thông Tin trường Đại Học Kiên Giang đã tạo điều kiện thuận lợi cho chúng em học tập và thực hiện đề tài này.

Chúng em xin bày tỏ lòng biết ơn sâu sắc tới thầy Nhan Thanh Nhã đã tận tình hướng dẫn chỉ bảo chúng em trong quá trình thực hiện đề tài.

Chúng em xin chân thành cảm ơn quý thầy cô trong khoa Công Nghệ Thông Tin đã tận tình giảng dạy, trang bị cho em nhưng kiến thức quý báu trong năm vừa qua. Chúng con xin chân thành cảm ơn ông bà, cha mẹ đã luôn động viên ủng hộ vật chất lẫn tinh thần trong suốt thời gian qua.

Chúng em xin cảm ơn sự quan tâm giúp đỡ và ủng hộ của các anh chị, bạn bè trong thời gian quá trình thực hiện khóa đề tài. Mặc dù đã cố gắng hoàn thành đề tài trong phạm vì và khả năng cho phép nhưng sẽ không tránh khổi những thiếu sót.

Chúng em mong nhận được sự thống cảm và góp ý của quý thầy cô và các bạn.

**LỜI CAM KẾT**

**-----------------**

Em đã đọc và hiểu về các hành vi vi phạm sự trung thực trong học thuật. Chúng em cam kết bằng danh dự cá nhân rằng nội dung tiểu luận/báo cáo này do em tự thực hiện và không vi phạm yêu cầu về sự trung thực trong học thuật.

**MỤC LỤC**

[**CHƯƠNG 1 GIỚI THIỆU CHUNG** 1](#_Toc130153185)

[**1.1 GIỚI THIỆU ĐỀ TÀI** 1](#_Toc130153186)

[**1.2 GIỚI THIỆU NGÔN NGỮ JAVA** 1](#_Toc130153187)

[**1.2.1 Đặc điểm của java** 1](#_Toc130153188)

[**1.2.2 Java Virtual Machine – JVM** 2](#_Toc130153189)

[**1.2.3 Java Development Kit – JDK** 2](#_Toc130153190)

[**1.2.4 Java Runtime Environment – JRE** 3](#_Toc130153191)

[**1.3 GIỚI THIỆU CHƯƠNG TRÌNH** 3](#_Toc130153192)

[**1.3.1 Ưu điểm** 3](#_Toc130153193)

[**1.3.2 Nhược điểm** 3](#_Toc130153194)

[**CHƯƠNG 2 THIẾT KẾ CHƯƠNG TRÌNH** 4](#_Toc130153195)

[**2.1 THIẾT KẾ CHƯƠNG TRÌNH** 4](#_Toc130153196)

[**2.1.1 CLIENT** 4](#_Toc130153197)

[**2.1.2 SEVER** 8](#_Toc130153198)

[**2.2 SƠ ĐỒ LỚP CHO HỆ THỐNG** 10](#_Toc130153199)

[**CHƯƠNG 3 DEMO CHƯƠNG TRÌNH** 11](#_Toc130153200)

[**3.1 DEMO GIAO DIỆN CHƯƠNG TRÌNH** 11](#_Toc130153201)

[**Tài liệu tham khảo** 15](#_Toc130153202)

# **DANH MỤC HÌNH**

[**HÌNH 2. 1 Code AudioMessage** 4](#_Toc127959685)

[**HÌNH 2. 2 Code Message** 4](#_Toc127959686)

[**HÌNH 2. 3 Code StringMessage** 5](#_Toc127959687)

[**HÌNH 2. 4 Code thu âm** 5](#_Toc127959688)

[**HÌNH 2. 5 Code dừng thu âm** 6](#_Toc127959689)

[**HÌNH 2. 6 Code phát âm thanh** 6](#_Toc127959690)

[**HÌNH 2. 7 Code chứa dữ liệu âm thanh** 6](#_Toc127959691)

[**HÌNH 2. 8 Các kiểu dữ liệu của âm thanh** 7](#_Toc127959692)

[**HÌNH 2. 9 Phương thức kết nối client với sever** 7](#_Toc127959693)

[**HÌNH 2. 10 ObjectOuputStream và ObjectInputStream** 7](#_Toc127959694)

[**HÌNH 2. 11 Code kết nói Client với sever** 8](#_Toc127959695)

[**HÌNH 2. 12 Code gửi tin nhắn và nhận tin nhắn** 8](#_Toc127959696)

[**HÌNH 2. 13 Cổng sever dùng để kết nối client** 9](#_Toc127959697)

[**HÌNH 2. 14 Sơ đồ lớp** 10](#_Toc127959698)

[**Hình 3 1 Giao Diện Đăng Nhập** 11](#_Toc127959721)

[**Hình 3 2 Giao diện người dùng** 11](#_Toc127959722)

[**Hình 3 3 Gửi văn bản** 12](#_Toc127959723)

[**Hình 3 4 Thu âm thanh** 12](#_Toc127959724)

[**Hình 3 5 Dừng, gửi và nghe đoạn thu âm** 13](#_Toc127959725)

# **GIỚI THIỆU CHUNG**

## **GIỚI THIỆU ĐỀ TÀI**

Sự xuất hiện và phát triển mạnh mẽ của công nghệ thông tin và truyền thông đã, đang và sẽ là nhân tố thúc đẩy sự phát triển của thế giới nói chung và Việt Nam nói riêng. Có thể nói, công nghệ thông tin và truyền thông đã tác động tích cực đến hầu hết các ngành nghề trong xã hội, trong đó có lĩnh vực giáo dục, nơi tính hiệu quả trong cả quản lý và giảng dạy đã được chứng minh. Từ những nhận định đó nên nhóm đã chọn đề tài “Viết chương trình ứng dụng group chat, voice chat” để nghiên cứu.

## **GIỚI THIỆU NGÔN NGỮ JAVA**

Java là một nền tảng phát triển các ứng dụng phần mềm có vị trí rất lớn trong những năm cuối thế kỉ 20, đầu thế kỉ 21. Đánh dấu sự trưởng thành của mô hình lập trình hướng đối tượng, nó được coi là một nền tảng mang tính cách mạng trong ngành phần mềm. Mô hình máy ảo Virtual Machine đã cho phép các ứng dụng viết bằng Java có thể chạy trên nhiều hệ điều hành khác nhau.

### **Đặc điểm của java**

Java một ngôn ngữ độc lập với các nền tảng. Có nghĩa là mã code java có thể biên dịch và chạy trên các nền tảng khác nhau như Windows, Linux, Mac OS etc.

Cơ chế tự động gom rác thông minh và được phát triển liên tục. Các ứng dụng Java sẽ trở nên mạnh mẽ và nhanh nhẹn hơn khi có nhiều vùng nhớ được trả lại từ trình gom rác. Các lập trình viên không cần phải dọn rác thủ công vì nó tốn khá nhiều thời gian và đôi khi sẽ có những thiếu xót dẫn đến crash chương trình.

Bảo mật: Do Java phải biên dịch qua máy ảo Java (JVM) nên sẽ được bảo mật cao hơn khi mọi đối tượng phải qua JVM mới đến hệ điều hành.

Phân tán: Java hỗ trợ lập trình cho hệ thống phân tán như client-server, RMI… bằng Java Web, UDP, TCP

### **Java Virtual Machine – JVM**

Diagram

Description automatically generated

Như bạn thấy mỗi hệ điều hành chạy chương trình của nó với một mã máy khác nhau đó là mã bytecodes. Windows được biên dịch dưới dạng exe còn Linux là .ELF vì vậy hầu hết các phần mềm gặp khó khăn khi viết ra để sử dụng trên nhiều hệ điều hành.

Máy ảo java được sinh ra với 3 mục đích chính:

* Dịch mã java ra mã máy chạy được trên các hệ điều hành khác nhau
* Tăng tốc độ
* Nâng cao độ bảo mật và tránh virus phá source code

JVM có 3 thành phần chính: ̀

* Class Loader: Tìm kiếm và load các file .class vào vùng nhớ của java dưới dạng bytecode
* Data Area : vùng nhớ hệ thống cấp phát cho Java Virtual Machine
* Execution Engine: chuyển các lệnh của JVM trong file .class thành các lệnh của máy, hệ điều hành tương ứng và thực thi chúng.

### **Java Development Kit – JDK**

Java Development Kit (JDK) là một trong ba gói công nghệ cốt lõi được sử dụng trong lập trình Java, cùng với JVM (Máy ảo Java - Java Virtual Machine) và JRE (Java Runtime Environment - Môi trường Java Runtime). Việc phân biệt giữa ba công nghệ này, cũng như hiểu được cách chúng kết nối với nhau là rất quan trọng.

JVM là thành phần Java nền tảng để chạy các chương trình.

JRE là phần on-disk của Java, phần tạo ra JVM.

JDK cho phép các developer tạo các chương trình Java, trong đó các chương trình có thể được JVM và JRE xử lý và chạy.

Các developer khi mới sử dụng Java thường dễ nhầm lẫn Java Development Kit với Java Runtime Environment. Để phân biệt, ta cần nhớ rằng JDK là một gói các công cụ phát triển phần mềm dựa trên Java, trong khi JRE là gói công cụ sử dụng để chạy Java code.

JRE có thể được sử dụng như một thành phần độc lập để chạy các chương trình Java, dù vậy, JRE vẫn là một phần của JDK. JDK đòi hỏi bộ công cụ JRE vì chạy các chương trình Java là một phần của việc phát triển chúng.

### **Java Runtime Environment – JRE**

JRE là một thành phần quan trọng của JDK. JRE là một môi trường để thực thi các chương trình java. Có nghĩa là khi máy tính của bạn có JRE mà không có JDK bạn chỉ có thể chạy chứ không biên dịch được một chương trình java.

## **GIỚI THIỆU CHƯƠNG TRÌNH**

Chương trình được tạo ra nhằm mục đích phục vụ cho nhu cầu trò chuyện online, các cuộc họp khẩn cấp, bạn bè cùng chơi game hay nhưng hội bạn lâu ngày muốn cùng với nhau nói chuyện hoặc đơn giản hơn là đưa 1 vài thông tin từ nơi này sang nơi khác.

### **Ưu điểm**

* Truy cập nhanh chóng
* Có thể truy cập nhiều người cùng lúc
* Hình thức truy cập đơn giản
* Giao diện thân thiện với người dùng

### **Nhược điểm**

* Bảo mật thấp
* Tính năng kém
* Chưa đáp ứng được nhìu nhu cầu khác nhau

# **THIẾT KẾ CHƯƠNG TRÌNH**

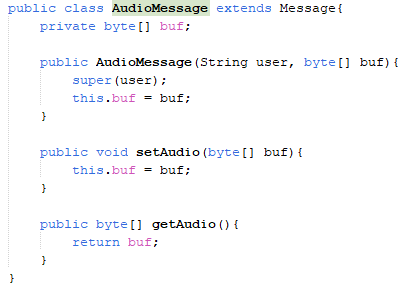
## **THIẾT KẾ CHƯƠNG TRÌNH**

### **CLIENT**

#### **MESSAGE TYPE**

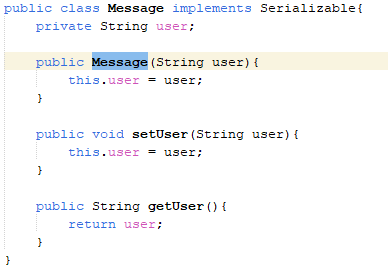
Dùng để định dạng kiểu dữ liệu âm thanh và kiểu dữ liệu chuỗi cho việc gửi tin nhắn bằng giọng nói và văn bản. Gồm có:

* Lớp AudioMessage: chứa dữ liệu kiểu âm thanh.



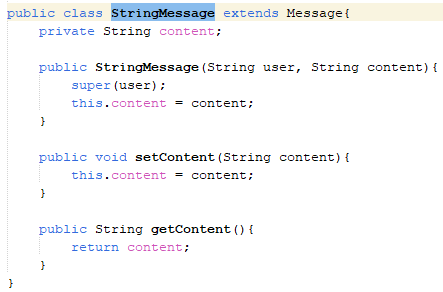
**HÌNH 2. 1 Code AudioMessage**

* Lớp Message: dùng để chứa thuộc tính người dùng, AudioMessage và StringMessage sẽ kế thừa thuộc tính người dùng này.



**HÌNH 2. 2 Code Message**

* Lớp StringMessage: chứa dữ liệu kiểu chuỗi.



**HÌNH 2. 3 Code StringMessage**

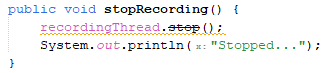
#### **CLIENT**

Dùng để kết nối với sever, hiển thị văn bản, giọng nói, định dạng âm thanh, thu âm, lưu trữ âm thanh, gửi dữ liệu về sever và lấy dữ từ sever hiển thị cho người dùng... Gồm có:

* Lớp AudioHandler: dùng thu âm thanh giọng nói từ ng dùng, dừng thu âm, chứa dữ liệu âm thanh, phát âm thanh vừa được thu.



**HÌNH 2. 4 Code thu âm**

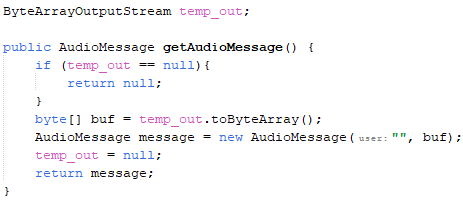


**HÌNH 2. 5 Code dừng thu âm**

Graphical user interface, text, application

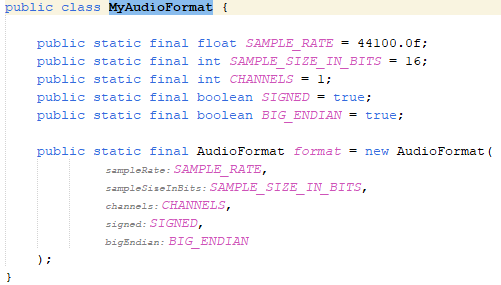
Description automatically generated

**HÌNH 2. 6 Code phát âm thanh**



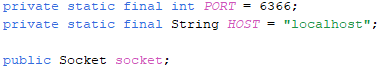
**HÌNH 2. 7 Code chứa dữ liệu âm thanh**

* Lớp MyAudioFormat: định dạng âm thanh(mẫu âm thanh/giây, độ sâu của mẫu âm thanh...)



**HÌNH 2. 8 Các kiểu dữ liệu của âm thanh**

* Lớp Client:
  + public Socket(String HostName, int PortNumber) : Phương thức này dùng để nối kết đến một server có tên là HostName, cổng là PortNumber. Nếu nối kết thành công, một kênh ảo sẽ được hình thành giữa Client và Server.
    - HostName: Địa chỉ IP hoặc tên logic theo dạng tên miền.
    - PortNumber: có giả trị từ 0 ..65535.



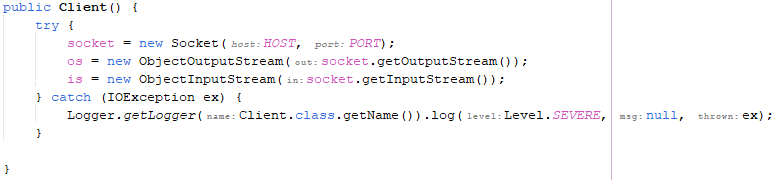
**HÌNH 2. 9 Phương thức kết nối client với sever**

* + public InputStream ObjectInputStream(): Phương thức này trả về InputStream nối với Socket. Chương trình Client dùng InputStream này để nhận dữ liệu từ Server gửi về.
  + Public OutputStream ObjectOutputStream(): Phương thức này trả về OutputStream nối với Socket. Chương trình Client dùng OutputStream này để gởi dữ liệu cho Server.



**HÌNH 2. 10 ObjectOuputStream và ObjectInputStream**

* Mở một socket nối kết đến server đã biết địa chỉ IP (hay tên miền) và số hiệu cổng.



**HÌNH 2. 11 Code kết nói Client với sever**

* Gửi dữ liệu về sever và lấy dữ liệu từ sever.



**HÌNH 2. 12 Code gửi tin nhắn và nhận tin nhắn**

### **SEVER**

#### **MESSAGE TYPE**

Giống với client.

#### **SEVER**

Server tạo Socket, gán số hiệu cổng và lắng nghe yêu cầu nối kết. Server sẵn sàng phục vụ Client.socket(): Server yêu cầu tạo một socket để có thể sử dụng các dịch vụ của tầng vận chuyển.

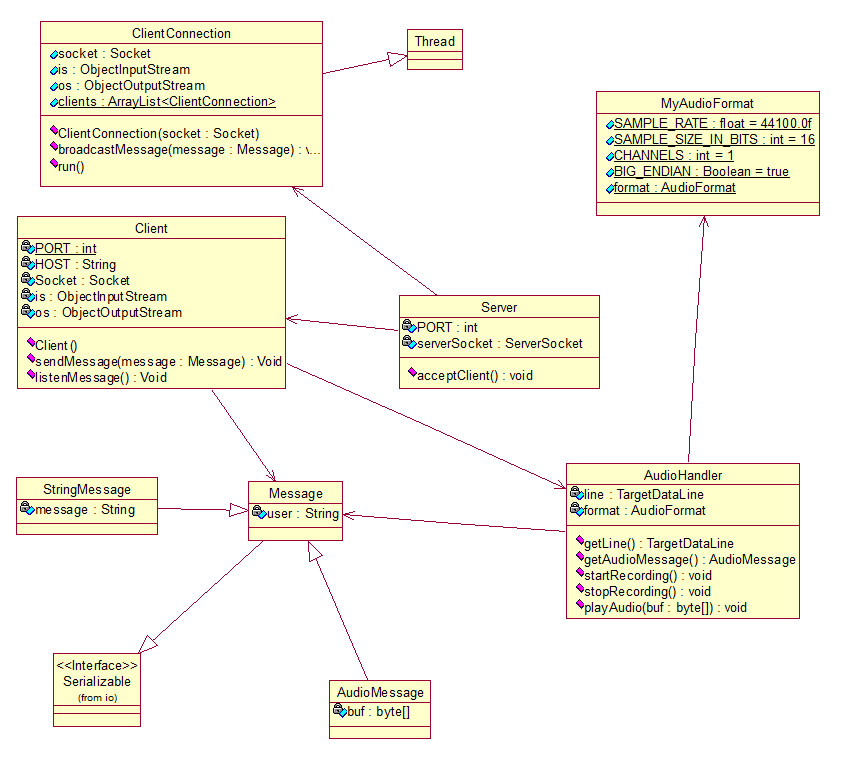
Server yêu cầu gán số hiệu cổng (port) cho socket.

Server lắng nghe các yêu cầu nối kết từ các client trên cổng đã được gán.



**HÌNH 2. 13 Cổng sever dùng để kết nối client**

## **SƠ ĐỒ LỚP CHO HỆ THỐNG**

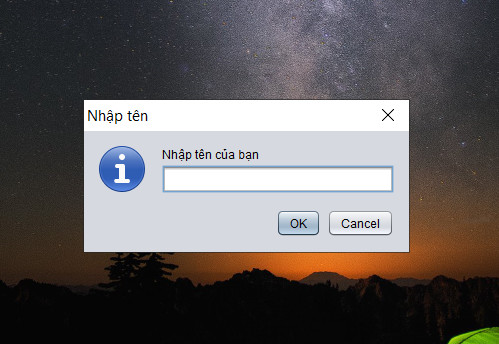


**HÌNH 2. 14 Sơ đồ lớp**

# **DEMO CHƯƠNG TRÌNH**

## **DEMO GIAO DIỆN CHƯƠNG TRÌNH**

Khởi chạy clien giao diện đăng nhập sẽ hiện ra.



**Hình 3 1 Giao Diện Đăng Nhập**

Sau khi khơi chạy thành công nếu muốn dừng sử sử dụng người dùng nhấn “ Cancel ” giao diện sẽ đóng lại.

Nếu muốn tiếp tục sử dụng người dùng cần nhập tên sử dụng của bạn rồi nhấn “ Ok ” màn hình sẽ chuyển sang giao diện chat

Graphical user interface

Description automatically generated

**Hình 3 2 Giao diện người dùng**

Text

Description automatically generated

**Hình 3 3 Code show tin nhắn**

Người dùng gửi tin nhắn bằng cách soạn tin nhắn cần nhắn vào ô “ xin chào mọi người “ sau đó nhấn gửi văn bản.

Graphical user interface, application

Description automatically generated

**Hình 3 4 Gửi văn bản**

Graphical user interface, text, application, email

Description automatically generated

**Hình 3 5 Code gửi tin bằng văn bản**

Khi cần dùng chat voice thì người dùng có thể nhấn nút “ thu âm “.

Graphical user interface

Description automatically generated

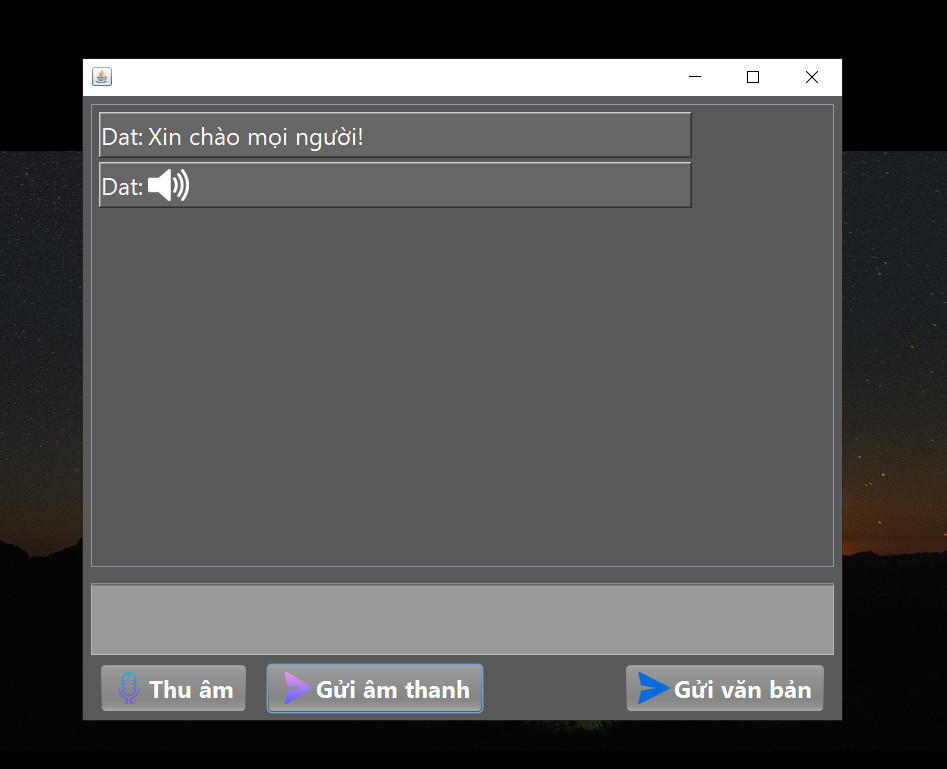
**Hình 3 6 Thu âm thanh**

Graphical user interface, text, application

Description automatically generated

**Hình 3 7 Code thu âm thanh**

Sau khi thu âm xong người dùng nhấn “ Dừng “ sau đó nhấn “ gửi âm thanh “ để gửi đoạn thu âm vừa rồi.



**Hình 3 8 Dừng, gửi và nghe đoạn thu âm**

Graphical user interface, text, application

Description automatically generated

**Hình 3 9 Code dừng thu âm, gủi âm thanh**

Người nhận chỉ cần nhấn vào icon cái loa thì cỏ thể nghe đc đoạn thu âm.

**Hướng phát triển của ứng dụng**

* Thêm các chức năng nghe gọi trực tiếp giữa 2 người
* Thêm tính năng tham gia phòng chat bằng link
* Thêm tính năng trưởng phòng chat ( có thể thêm người và xóa người dùng trong group chat)

# **Tài liệu tham khảo**

Tập bài giảng môn lập trình mạng trường Đại học Kiên Giang năm học 2020 - 2021 do thầy Nhan Thanh Nhã phụ trách giảng dạy.