

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



**Mạng Máy Tính - Computer Network - 191**

---

**Assignment 1 Extra**

## **SOCKET CHAT APPLICATION**

---

GVHD: PHD. PHẠM TRẦN VŨ  
PHD. NGUYỄN MẠNH THÌN  
SINH VIÊN:  
HOÀNG VŨ TRỌNG THỤY - 1710321

# Mục lục

<b>1</b>	<b>Định nghĩa chức năng mở rộng</b>	<b>3</b>
1.1	Đăng nhập/đăng xuất . . . . .	3
1.2	Gửi tin nhắn đến Admin . . . . .	3
1.3	Admin gửi tin nhắn đến toàn bộ người dùng . . . . .	3
1.4	Xem trạng thái online offline của người trong list danh sách . . . . .	3
<b>2</b>	<b>Định nghĩa giao thức</b>	<b>3</b>
2.1	Giao thức giao tiếp client-server . . . . .	3
<b>3</b>	<b>Miêu tả chi tiết hiện thực chức năng</b>	<b>5</b>
3.1	Đăng nhập/đăng xuất ứng dụng. . . . .	5
3.2	Gửi tin nhắn văn bản từ web browser . . . . .	6
3.3	Thay đổi trạng thái hoạt động . . . . .	6
<b>4</b>	<b>Thiết kế chi tiết</b>	<b>6</b>
4.1	Kiến trúc tổng quát . . . . .	6
4.2	Lược đồ lớp . . . . .	7
<b>5</b>	<b>Đánh giá kết quả hiện thực</b>	<b>11</b>
5.1	Khởi động server . . . . .	11
5.2	Đăng nhập . . . . .	11
5.3	Gửi tin nhắn . . . . .	12
5.3.1	Gửi tin nhắn vào phòng chat chung . . . . .	12
5.3.2	Gửi tin nhắn riêng cho user khác . . . . .	16
5.4	Chuyển trạng thái hoạt động . . . . .	19
<b>A</b>	<b>Hướng dẫn sử dụng</b>	<b>21</b>

## Danh sách hình vẽ

1	Mô hình định nghĩa giao thức giao tiếp trong ứng dụng . . . . .	4
2	Kiến trúc tổng quát . . . . .	6
3	Lược đồ lớp package com.server . . . . .	7
4	Lược đồ lớp package com.protocols . . . . .	8
5	Lược đồ các lớp trong package com.lobby . . . . .	9
6	Lược đồ lớp package com.messenger . . . . .	10
7	Server CLI . . . . .	11
8	Server UI . . . . .	11
9	Màn hình Login . . . . .	12
10	Main Interface . . . . .	12
11	Send public message . . . . .	13
12	Message sent in community channel . . . . .	13
13	Send voice message . . . . .	14
14	Voice message sent . . . . .	14
15	Browse file . . . . .	15
16	File sent . . . . .	15
17	Send private message . . . . .	16
18	Enter private message . . . . .	16
19	Private message sent . . . . .	17
20	Browse directory private . . . . .	17
21	Change chat bubble's color . . . . .	18
22	Chat bubble's color changed . . . . .	18
23	Change state . . . . .	19
24	Thư mục app chứa artifacts để mở ứng dụng . . . . .	21
25	Cách mở messenger bằng terminal . . . . .	21
26	Cách mở server bằng terminal . . . . .	21

## 1 Định nghĩa chức năng mở rộng

### 1.1 Đăng nhập/đăng xuất

- Cho phép user nhập tên và nhập đường dẫn tới server và tên port để vào phòng chat. Tên user phải bắt đầu bằng chữ cái

### 1.2 Gửi tin nhắn đến Admin

- Cho phép user gửi tin nhắn đến Admin.
- Tin nhắn gửi đi là tin nhắn văn bản, emoji, sticker.

### 1.3 Admin gửi tin nhắn đến toàn bộ người dùng

- Cho phép Admin gửi tin nhắn đến một hoặc toàn bộ người dùng.
- Tin nhắn gửi đi là tin nhắn văn bản, emoji, sticker.

### 1.4 Xem trạng thái online offline của người trong list danh sách

- Admin có thể thấy ai đang online nếu người đó có tích xanh trước tên của mình, offline nếu màu đỏ ở bảng danh sách user và away nếu tích có màu vàng.
- User cũng có thể đổi trạng thái sử dụng của mình về online, away, busy.

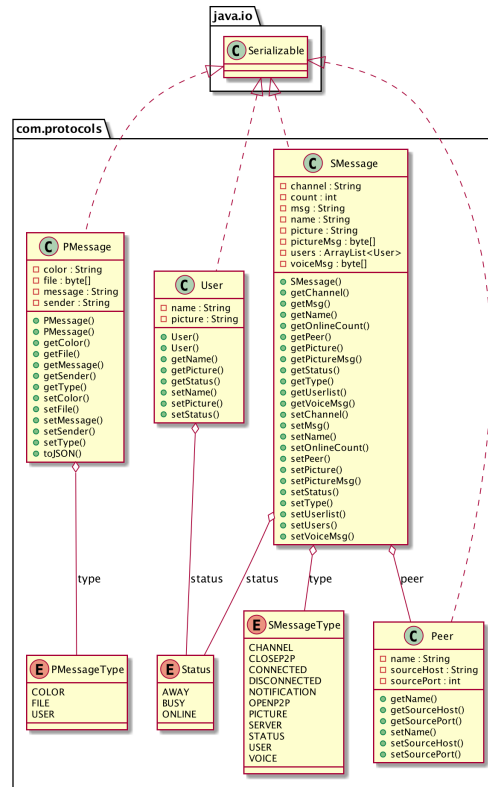
## 2 Định nghĩa giao thức

Phần mở rộng chủ yếu dùng giao thức sử dụng trong giao tiếp giữa client-server (dựa trên giao thức TCP).

### 2.1 Giao thức giao tiếp client-server

Client giao tiếp với Server để thực hiện các chức năng: thông báo đăng nhập/đăng xuất; thay đổi trạng thái hoạt động; gửi công khai/riêng tư tin nhắn, ảnh, âm thanh sử dụng giao thức TCP; thiết lập kết nối UDP trực tiếp với client khác.

PROTOCOLS's Class Diagram



PlantUML diagram generated by SketchIt! (<https://bitbucket.org/pmesmeur/sketchit/>)  
For more information about this tool, please contact philippe.mesmeur@gmail.com

Hình 1: Mô hình định nghĩa giao thức giao tiếp trong ứng dụng

Giao thức gồm các trường thông tin chính sau:

1. **name: String**. Thông tin về client (tên đăng nhập).
2. **type: SMessageType**. Chức năng yêu cầu server xử lý, gồm có:
  - DISCONNECTED
  - CONNECTED
  - STATUS
  - USER
  - SERVER
  - NOTIFICATION
  - VOICE
  - CHANNEL
  - PICTURE.
  - OPENP2P
  - CLOSEP2P

3. **channel: String.** Thông tin về **kênh** đang giao tiếp, gồm: **#Community** - tin nhắn gửi công khai đến toàn bộ người dùng, **Personal** - tin nhắn gửi riêng đến người dùng xác định (toàn bộ dựa trên giao thức TCP, tức là client - server - client).
4. **count: Integer.** Số lượng người dùng.
5. **status: Status.** Trạng thái hoạt động của người dùng: ONLINE, AWAY, BUSY.
6. **users: User.** Danh sách người dùng.
7. **picture: String.** Đường dẫn đến ảnh người dùng.
8. **msg: String.** Nội dung tin nhắn văn bản.
9. **pictureMsg: byte[].** Nội dung của ảnh được mã hoá theo chuẩn Base64.
10. **voiceMsg: byte[].** Nội dung của âm thanh được mã hoá.

### 3 Miêu tả chi tiết hiện thực chức năng

Trong phần mở rộng này, Admin sẽ sử dụng Java App để có thể tư vấn cho các người dùng sử dụng web browser.

Như vậy, muốn giao tiếp từ Java App đến Web browser cần mở websocket để dễ dàng giao tiếp với người dùng sử dụng web browser thông qua thư viện **socket.io**.

Server có thể phục vụ cùng lúc nhiều yêu cầu từ client, tạo nhiều thread, trong đó, 1 thread **master** chuyên ghi nhận các socket kết nối đến server và nhiều thread (các **handle**) xử lý các yêu cầu chức năng tương ứng với từng client trong quá trình giao tiếp.

Các chức năng yêu cầu được xác định dựa vào trường thông tin **type** của mỗi message gửi từ client. Message sẽ được truyền theo đường đi client-server-clients.

#### 3.1 Đăng nhập/đăng xuất ứng dụng.

User đăng nhập thông qua web browser (có thể trên PC, laptop, mobile).

Server cho phép người dùng sử dụng một tên đăng nhập để đăng ký vào Lobby của ứng dụng. Tên đăng nhập này là duy nhất.

Server sẽ lưu trữ các người dùng thông qua HashMap từ **username** đến user tương ứng. Nhờ đó trên mỗi **handle** chỉ cần lưu một trường thông tin là **username** vẫn có thể truy xuất thông tin về người dùng cũng như kiểm tra tên đăng nhập đã tồn tại hay chưa.

Khi người dùng đăng nhập/đăng xuất, một message có **kiểu** là **CONNECTED/DISCONNECTED** sẽ được gửi từ client đến server, server sẽ ghi nhận và thông báo đến toàn bộ **#Community** (cùng kiểu với message nhận từ client) để cập nhật danh sách trạng thái trên khung hiện thị của mỗi client.

### 3.2 Gửi tin nhắn văn bản từ web browser

Người dùng có thể giao tiếp được với Admin nhờ giao diện trên web browser. Phía dưới là Listener là khâu trung gian xử lý các gói tin gửi và nhận. Các listener này lưu trữ thông tin sessionId của client, từ đó có thể nhận cũng như chuyển đổi gói tin phù hợp cho ứng dụng hiện thị.

Nội dung tin nhắn được trích xuất tại Input của Web UI, đóng gói trong protocol message và gửi xuống Listener để xử lý. Các chức năng khác cũng sử dụng cách thức tương tự để giao tiếp.

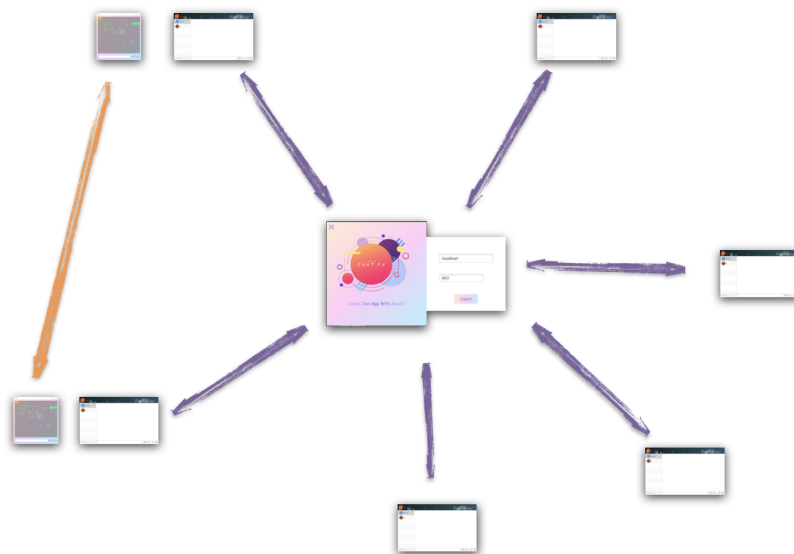
### 3.3 Thay đổi trạng thái hoạt động

Thông tin trạng thái hoạt động của người dùng được lưu tại trường thông tin **status**. trên mỗi message mỗi khi client gửi yêu cầu thay đổi trạng thái đến server.

Server sẽ ghi nhận và gửi đến toàn # Community (kể cả client gửi request) để cập nhật trạng thái hoạt động hiện thị trên khung lobby của mỗi client.

## 4 Thiết kế chi tiết

### 4.1 Kiến trúc tổng quát



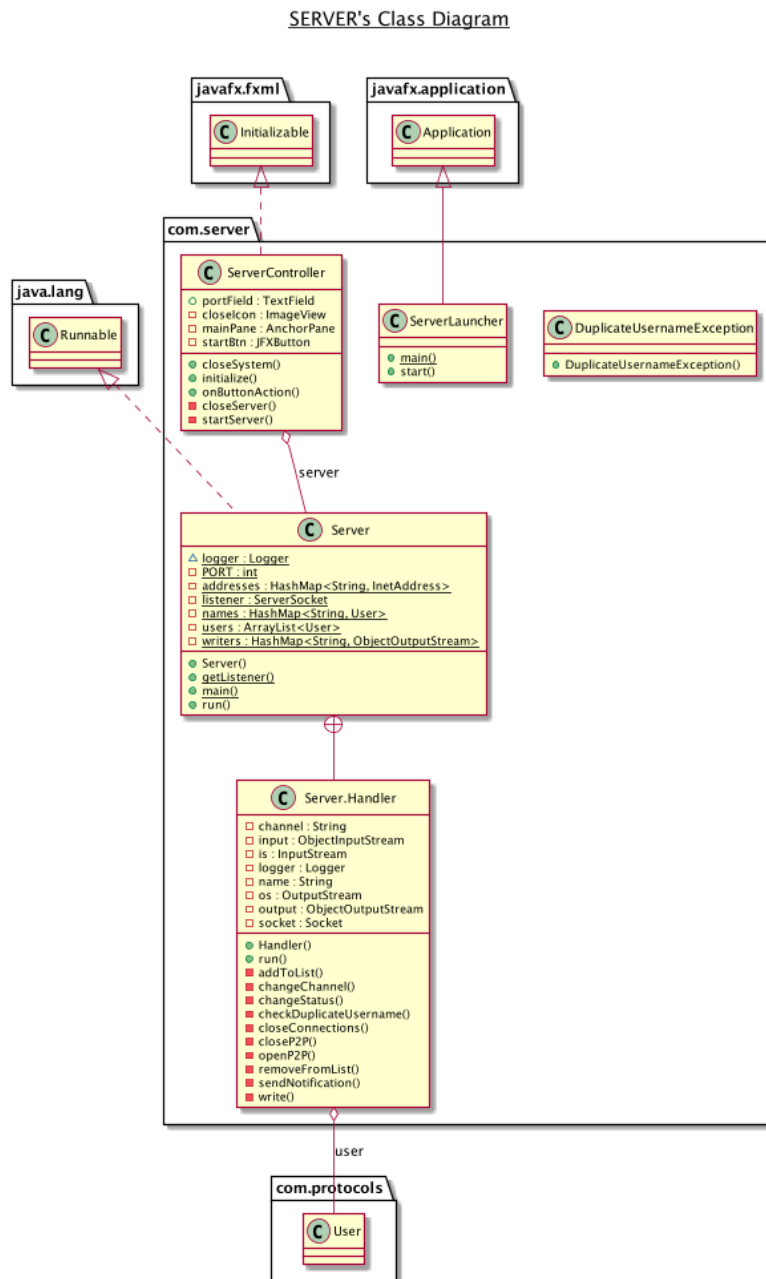
Hình 2: Kiến trúc tổng quát

Như đã trình bày ở phần trên, mỗi client sẽ đăng nhập và chuyển đến phòng chat chung. Nơi hiển thị tin nhắn đến cũng như thực hiện các chức năng của ứng dụng.

## 4.2 Lược đồ lớp

Các package chính trong ứng dụng:

- **com.server** - Gồm các class để hiển thị thực SocketServer cùng GUI, trong đó class **Server.java** hiện thực để xử lý các yêu cầu chức năng chính.



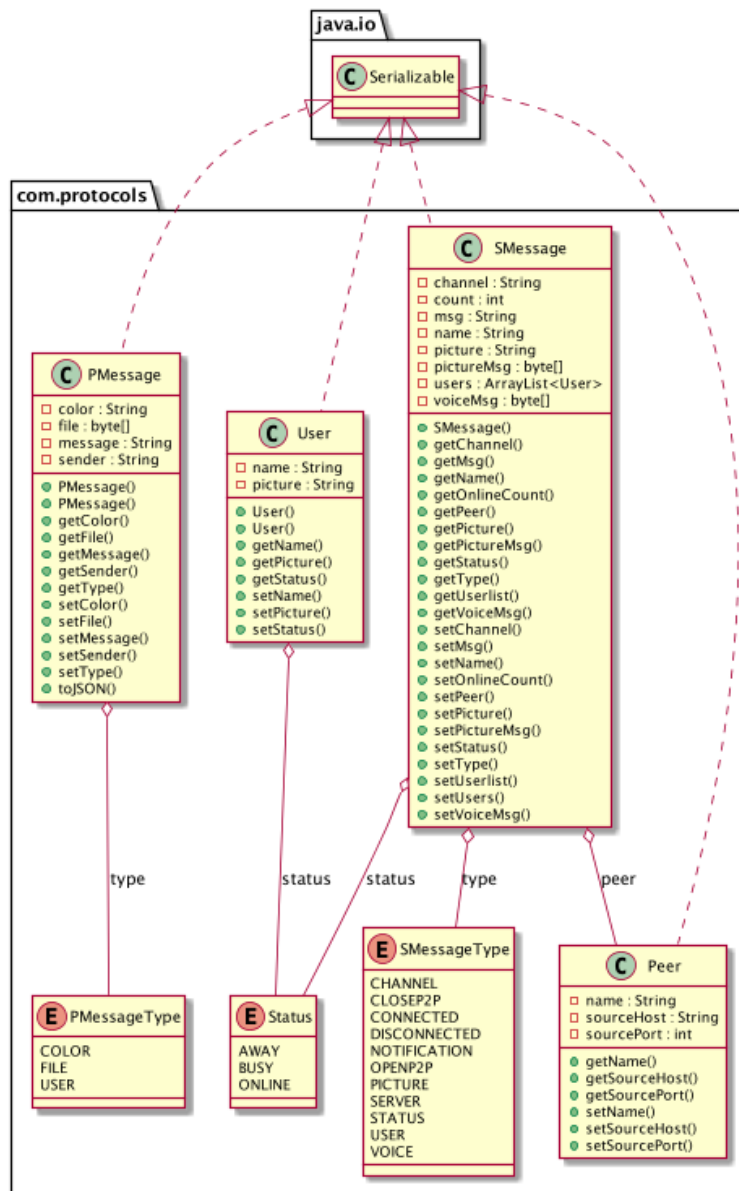
PlantUML diagram generated by SketchIt! (<https://bitbucket.org/pmesmeur/sketchit/>)  
For more information about this tool, please contact [philippe.mesmeur@gmail.com](mailto:philippe.mesmeur@gmail.com)

Hình 3: Lược đồ lớp package com.server



- **com.protocols** - Gồm các class định nghĩa các giao thức sử dụng trong ứng dụng, như: **SMessage**
  - giao thức giao tiếp giữa client-server; **PMessage** - giao thức giao tiếp client-client.

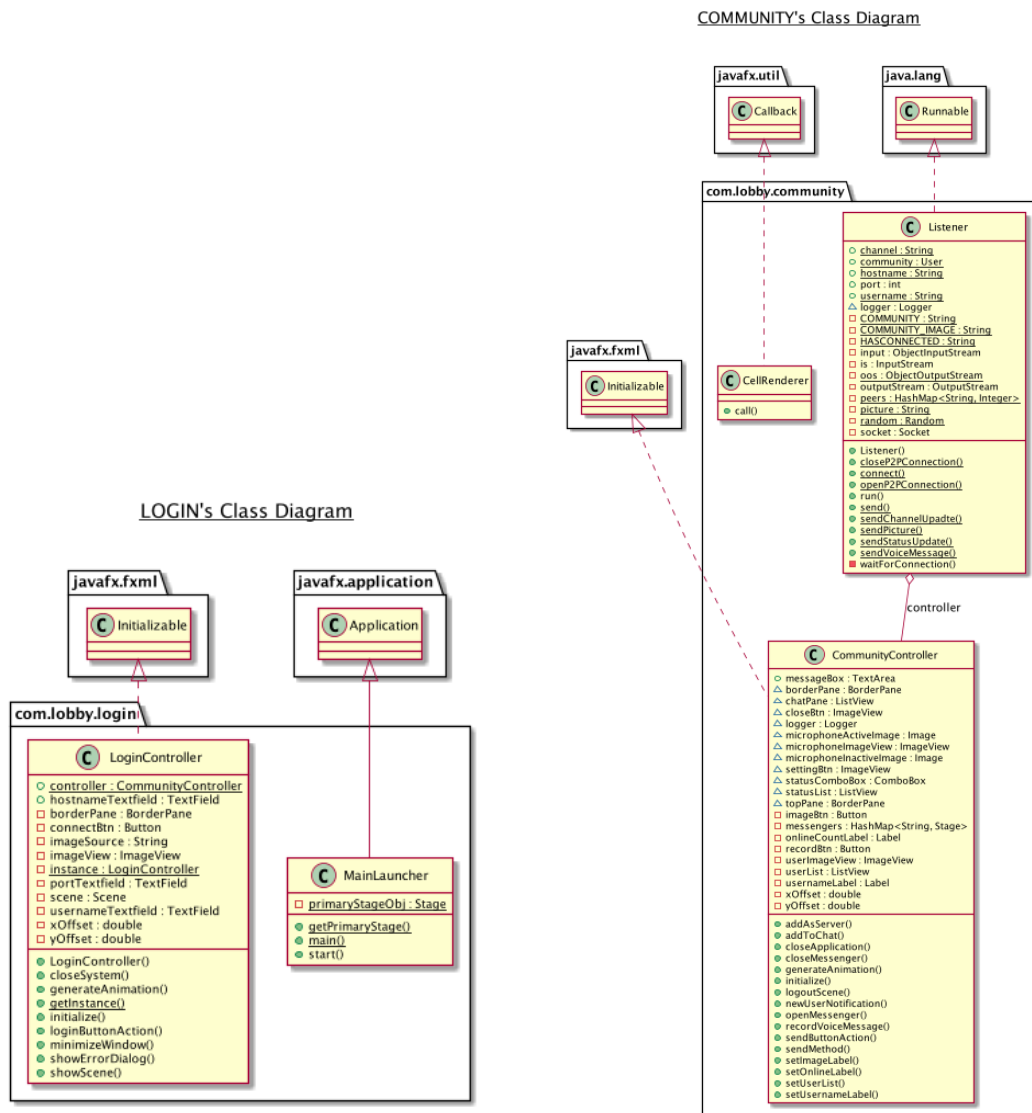
PROTOCOLS's Class Diagram



PlantUML diagram generated by SketchUML (<https://bitbucket.org/pmesmeur/sketchuml>)  
For more information about this tool, please contact philippe.mesmeur@gmail.com

Hình 4: Lược đồ lớp package com.protocols

- **com.lobby** - Gồm các class hiện thực phần front-end, cung cấp cách thức giao tiếp giữa người dùng và server.



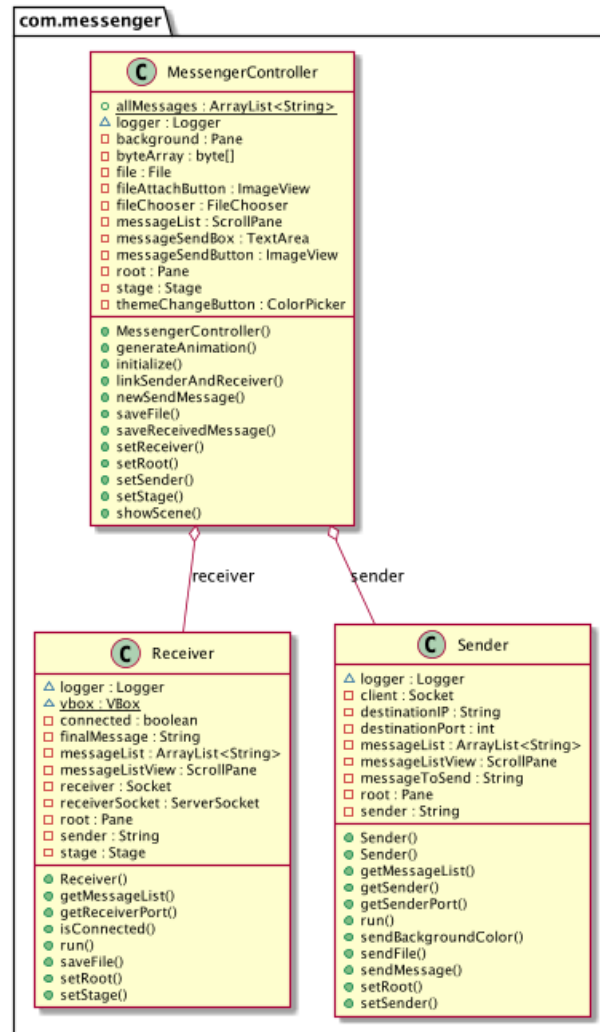
PlantUML diagram generated by SketchIt! (<https://bitbucket.org/pmesmeur/sketch.it>)  
For more information about this tool, please contact philippe.mesmeur@gmail.com

PlantUML diagram generated by SketchIt! (<https://bitbucket.org/pmesmeur/sketch.it>)  
For more information about this tool, please contact philippe.mesmeur@gmail.com

Hình 5: Lược đồ các lớp trong package com.lobby

- **com.messenger** - Gồm các class hiện thực giao tiếp trực tiếp giữa client-client.

### MESSANGER's Class Diagram



PlantUML diagram generated by SketchIt! (<https://bitbucket.org/pmameur/sketch.it>)  
For more information about this tool, please contact philippe.mesmeur@gmail.com

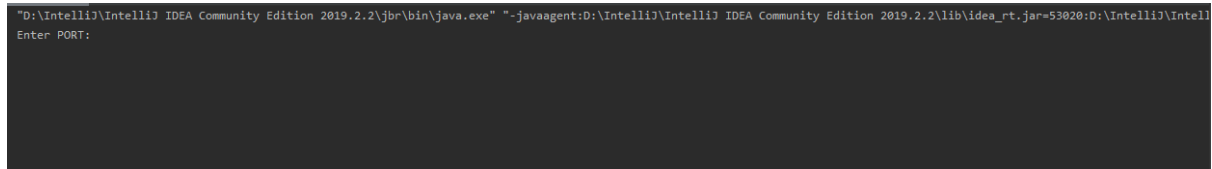
Hình 6: Lược đồ lớp package com.messenger

## 5 Đánh giá kết quả hiện thực

### 5.1 Khởi động server

Cách 1: sử dụng server CLI:

- Chạy class `com.server.Server`.

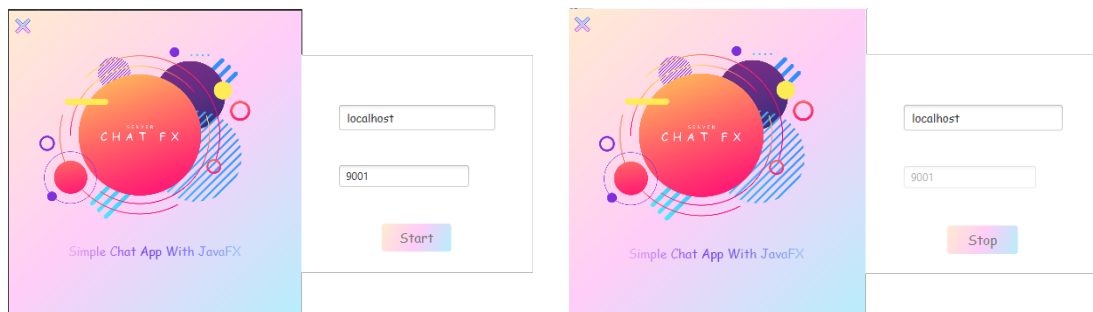


Hình 7: Server CLI

- Nhập tên port vào. VD: 9001.

Cách 2: sử dụng server UI:

- Chạy class `com.server.ServerController`.



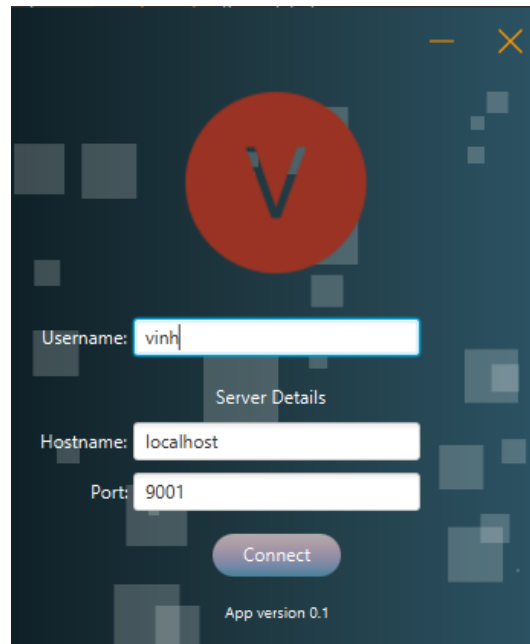
Hình 8: Server UI

- Nhập đường dẫn vào server ở ô localhost. Nhập port vào ô dưới rồi bấm nút Start để khởi động server.
- Để tắt server nhấn nút stop

### 5.2 Đăng nhập

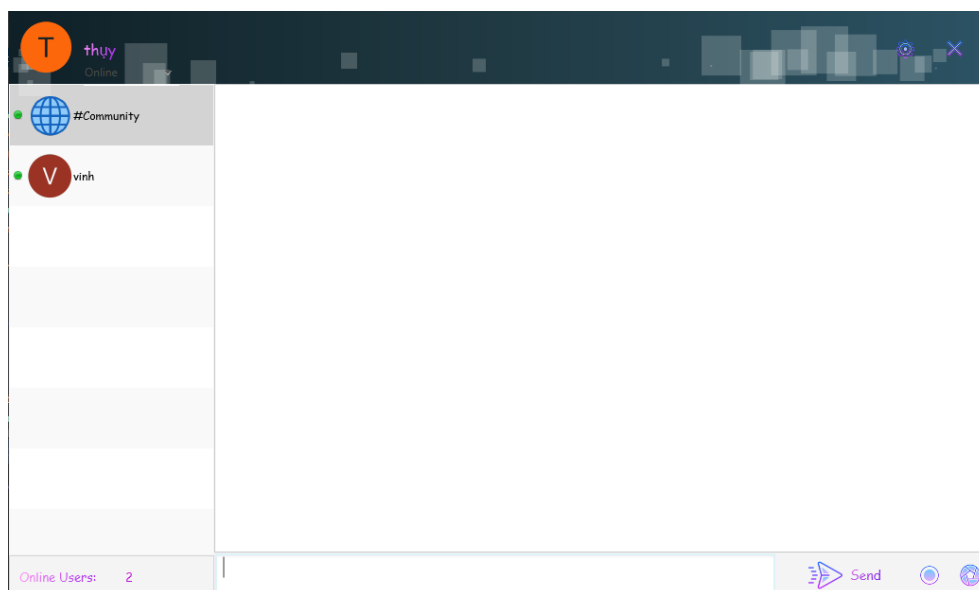
Chạy class `com.client.login.Mainlauncher`.

- Nhập tên người dùng vào ô username.
- Nhập đường dẫn tới server vào hostname.
- Nhập port kết nối.



Hình 9: Màn hình Login

Giao diện sau khi đăng nhập thành công vào server.



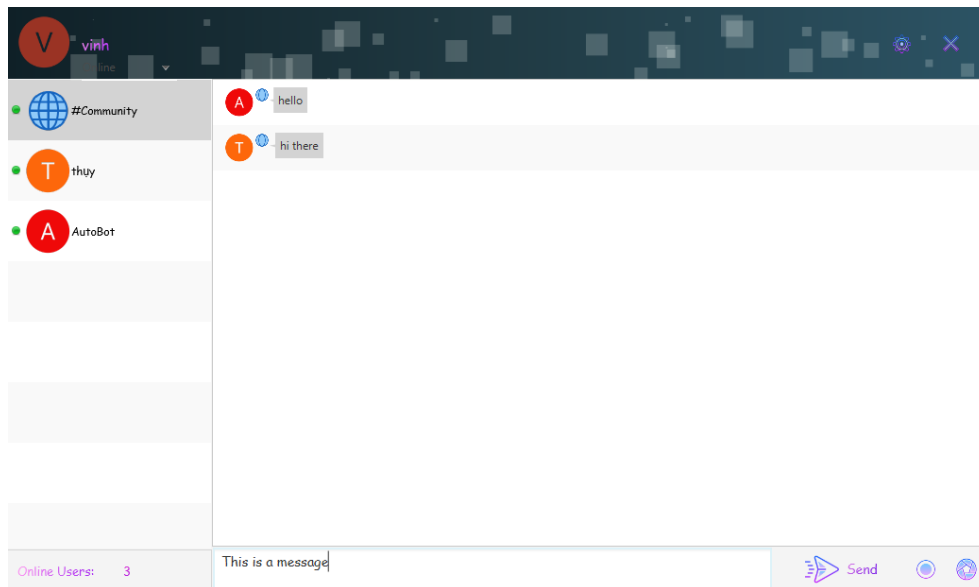
Hình 10: Main Interface

## 5.3 Gửi tin nhắn

### 5.3.1 Gửi tin nhắn vào phòng chat chung

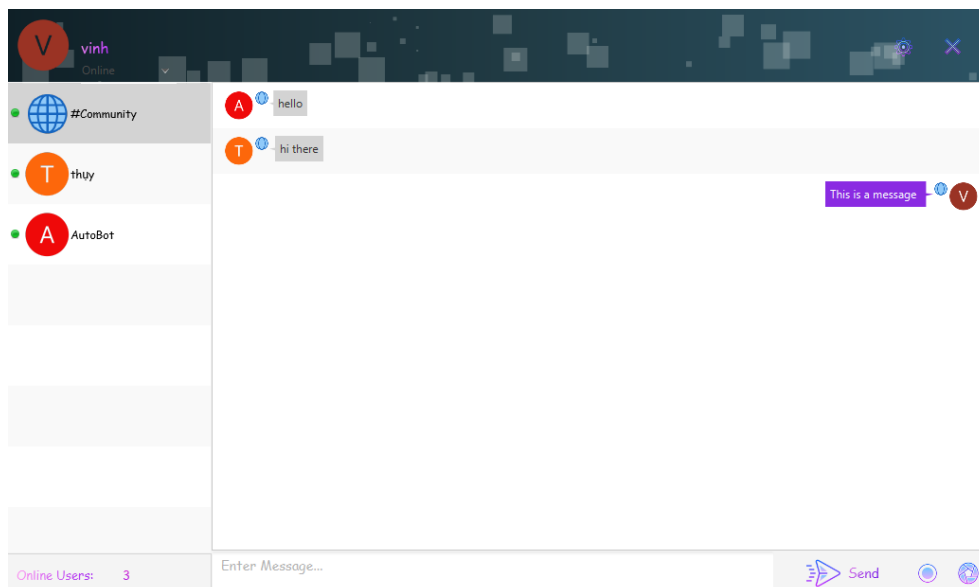
- Nhấn chọn kênh chat community

- Nhập tin nhắn từ bàn phím vào ô chat dưới cùng



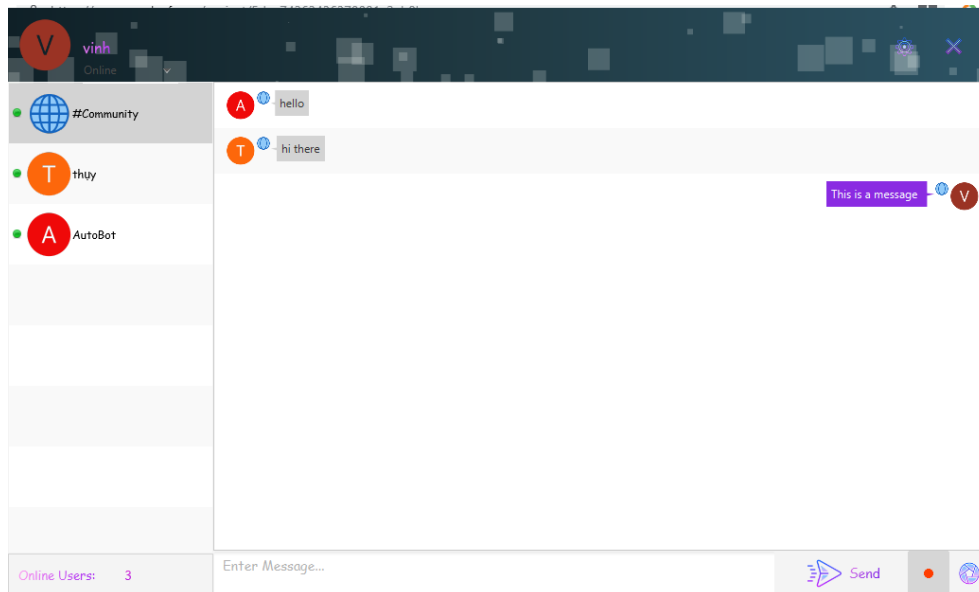
Hình 11: Send public message

- Bấm enter hoặc nhấn nút send cạnh ô chat sẽ được kết quả như hình



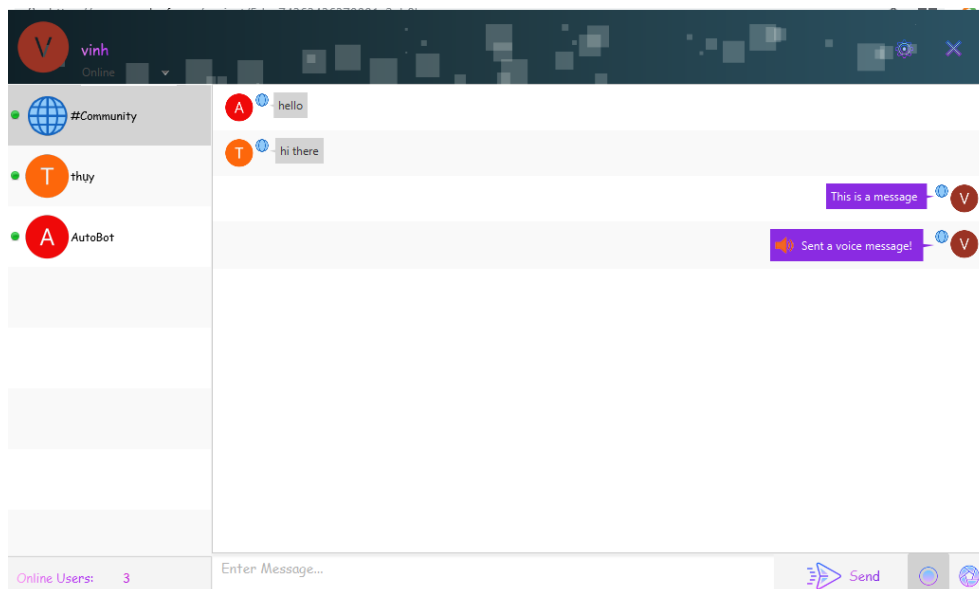
Hình 12: Message sent in community channel

- Để gửi tin nhắn thoại nhấn vào nút thu âm bên phải nút send và bắt đầu nói.



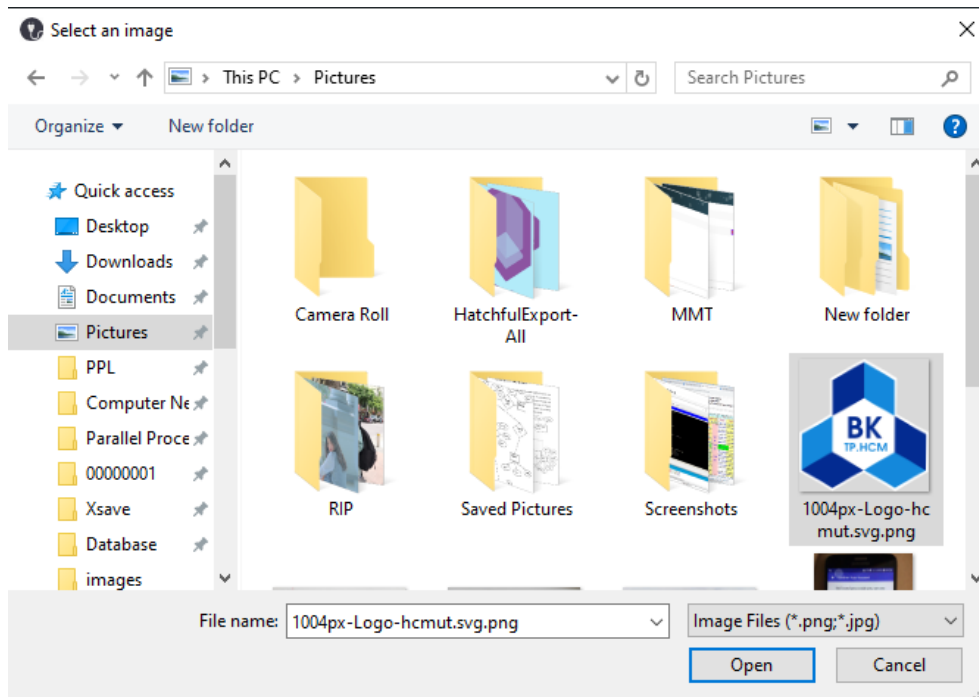
Hình 13: Send voice message

- Nhấn vào nút thu âm 1 lần nữa để gửi đi. Kết quả như hình sau.



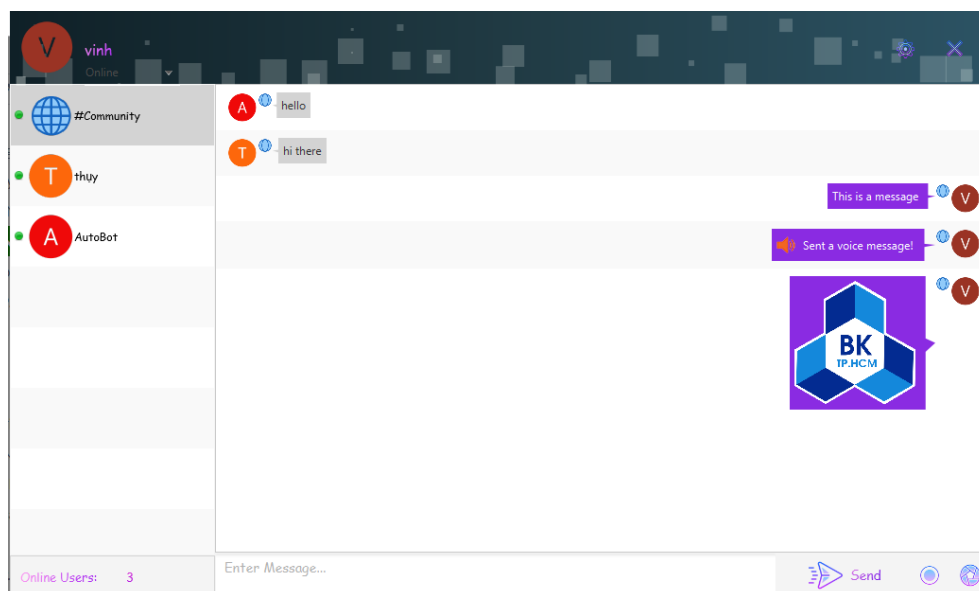
Hình 14: Voice message sent

- Để gửi file nhấn vào nút duyệt file bên phải nút thu âm để mở hộp thoại chọn thư mục.



Hình 15: Browse file

- Nhấn vào file cần chọn. Kết quả như hình sau.

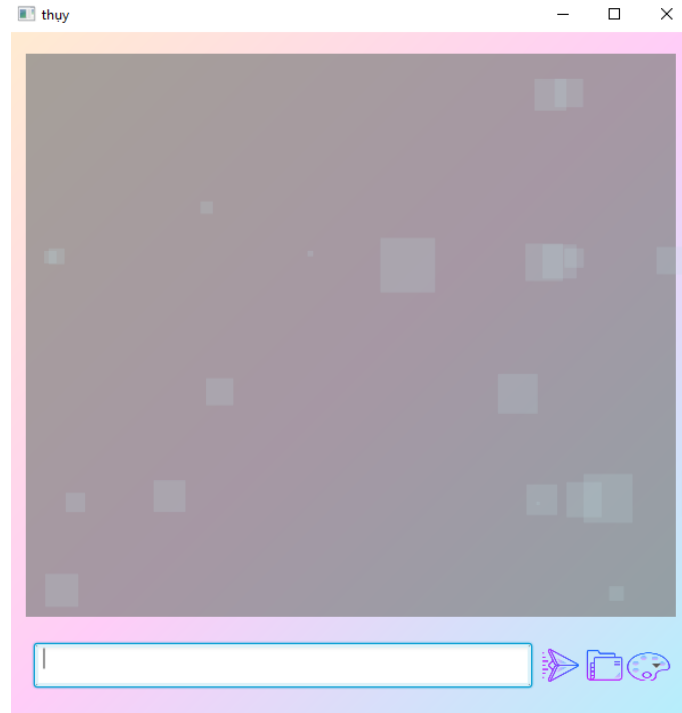


Hình 16: File sent



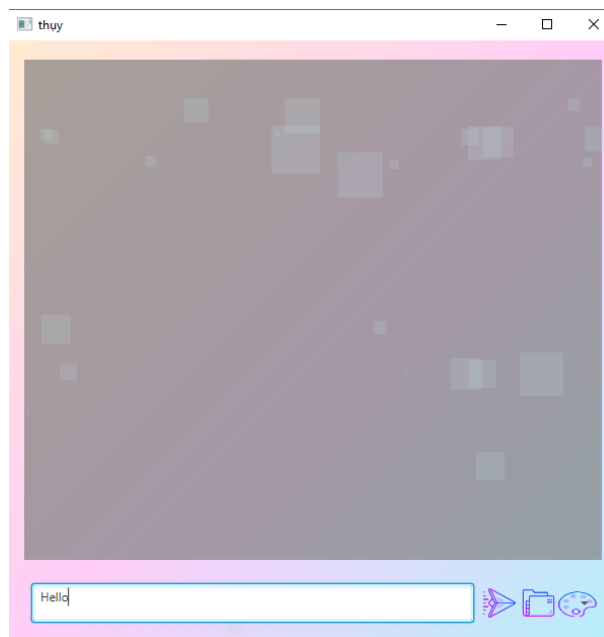
### 5.3.2 Gửi tin nhắn riêng cho user khác

- Nhấp chọn user để gửi tin nhắn đến.
- Cửa sổ chat mới hiện ra.



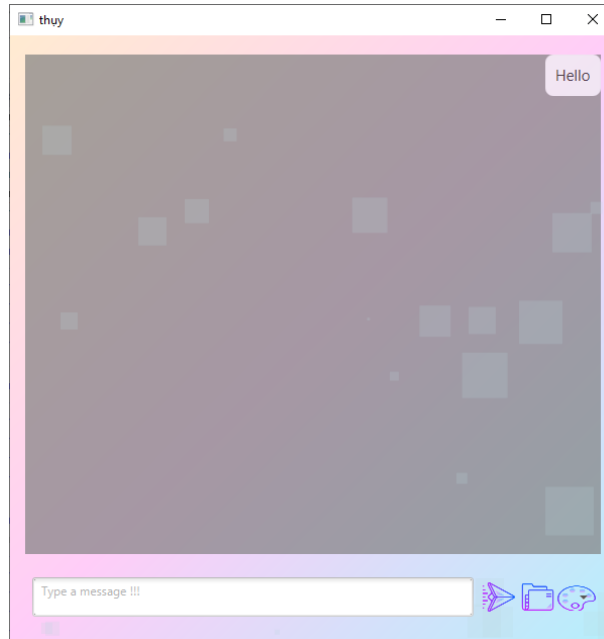
Hình 17: Send private message

- Nhập tin nhắn từ bàn phím vào ô chat dưới cùng.



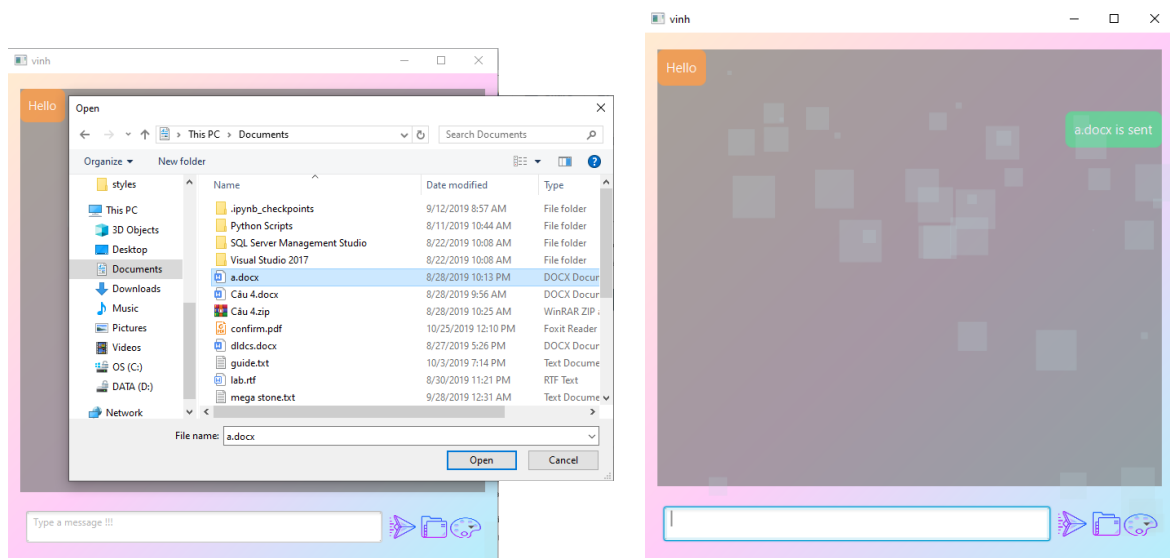
Hình 18: Enter private message

- Nhấn nút send ở bên cạnh ô chat hoặc nhấn phím enter để gửi tin.



Hình 19: Private message sent

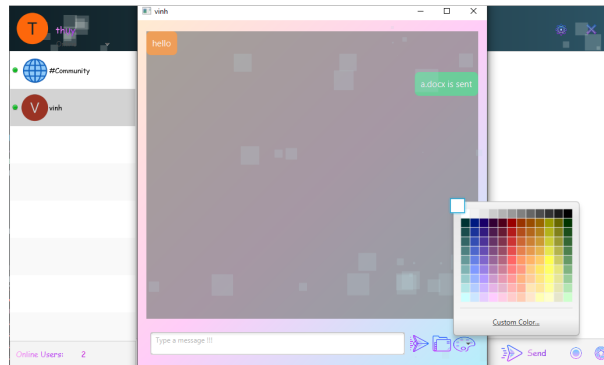
- Để gửi tệp tin nhấn nút hình tệp tin cạnh nút send để mở hộp thoại chọn thư mục.



Hình 20: Browse directory private

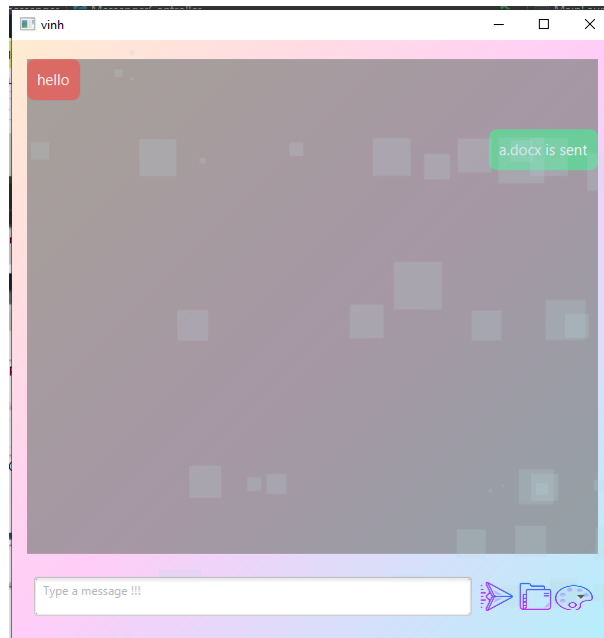
- Nhấn open để gửi

- Để đổi màu bóng chat nhấn chọn nút bảng màu nằm bên phải nút chọn file và chọn màu.



Hình 21: Change chat bubble's color

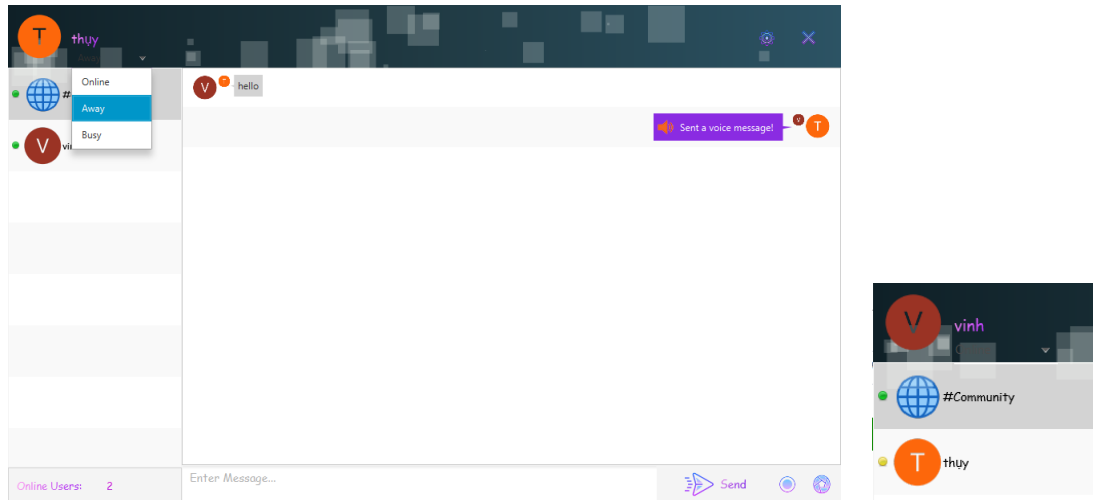
- Kết quả như hình sau.



Hình 22: Chat bubble's color changed

## 5.4 Chuyển trạng thái hoạt động

Nhấn vào dấu mũi tên ở dưới username góc trên cùng bên trái để mở bảng chuyển trạng thái. Sau



Hình 23: Change state

khi đổi trạng thái, dấu tích ở bên cạnh username sẽ đổi màu đối với các user khác.

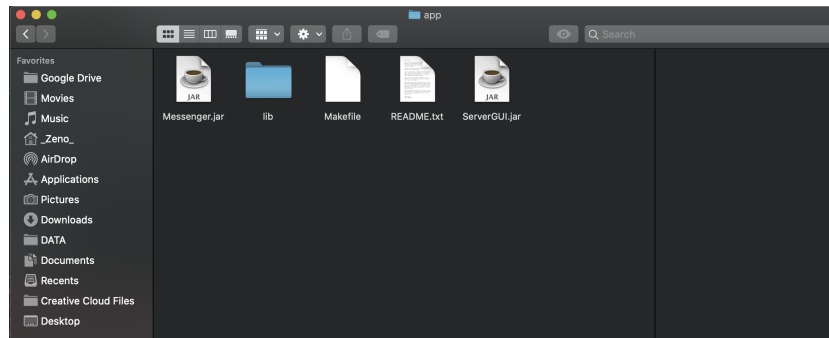


## Tài liệu

- [1] GPCoder *Xây dựng ứng dụng Client-Server với Socket trong Java.*  
last visited Oct 28th, 2019.
- [2] KeepToo *JavaFX UI design youtube channel .*  
Last seen Oct 25th, 2019.
- [3] <https://docs.oracle.com/javase/tutorial/>.  
Last visited Oct 28th, 2019.

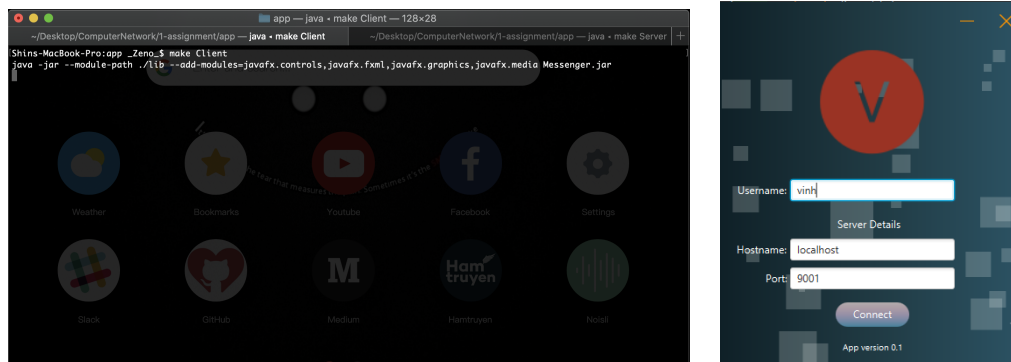
## A Hướng dẫn sử dụng

Thư mục *app/* (như hình dưới) chứa các artifacts (file .jar) để mở ứng dụng.



Hình 24: Thư mục app chứa artifacts để mở ứng dụng

Mở terminal trở đến thư mục hiện tại và chạy hai dòng lệnh để khởi động ứng dụng **make Client**



Hình 25: Cách mở messenger bằng terminal

Nếu muốn máy tính là một server, chạy dòng lệnh sau: **make Server**



Hình 26: Cách mở server bằng terminal