

HO CHI MINH CITY UNIVERSITY OF TECHNOLOGY
FACULTY OF COMPUTER SCIENCE AND TECHNOLOGY



Mạng Máy Tính

Bài tập lớn số 1

Ứng dụng Chat

GVHD: Lê Bảo Thịnh
SV: Nguyễn Hữu Khang - 2011365
SV: Khuê Vĩ Lương - 2052589
SV: Trần Sách Nhật - 2014009

TP. HỒ CHÍ MINH, THÁNG 12/2022

Mục lục

1	Giới thiệu chung	2
2	Xác định các chức năng cụ thể của ứng dụng	2
2.1	Start server	2
2.2	Login	2
2.3	Register	2
2.4	Chatroom	3
2.4.1	Dashboard	3
2.4.2	Inbox	3
3	Xác định và mô tả các giao thức truyền thông được sử dụng cho từng chức năng	5
3.1	Mô tả sơ bộ	5
3.2	Mô tả chi tiết	5
3.2.1	Chức năng Login (Đăng nhập)	5
3.3	Chức năng Register (Đăng ký)	8
3.4	Chức năng gửi tin nhắn cho toàn bộ người đang online (thẻ Dashboard)	8
3.5	Chức năng gửi tin nhắn cho một người	8
3.6	Chức năng gửi file cho một cá nhân nào đó	8
4	Chi tiết thiết kế	9
4.1	Mô hình client-server và giao diện lập trình ứng dụng mạng socket	9
4.2	Usecase Diagram	11
4.3	Class Diagram	14
5	Kết quả đạt được và đánh giá	14
5.1	Kết quả đạt được	14
5.2	Đánh giá	15
6	Hướng dẫn sử dụng chung	15
7	Phân công công việc	17

1 Giới thiệu chung

Ứng dụng các kiến thức về các giao thức đã học trong bộ môn Mạng Máy Tính, nhóm chúng em đã thiết kế một ứng dụng Chat đơn giản.

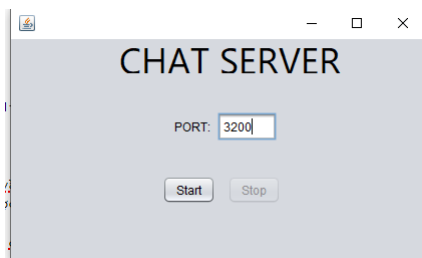
Giao thức sử dụng trong bài là: **TCP**.

Ngôn ngữ lập trình và giao diện: **Java, Java Swing**

2 Xác định các chức năng cụ thể của ứng dụng

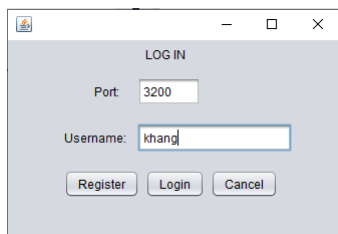
2.1 Start server

Muốn ứng dụng hoạt động được cần phải khởi động server. Điền Port là số nguyên, mặc định là 3200. Bấm start để khởi động server



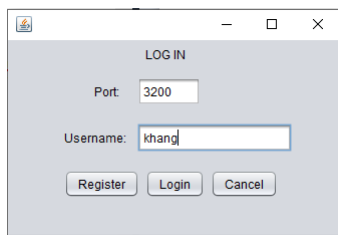
2.2 Login

Đây là màn hình Login của ứng dụng, người dùng để đăng truy cập ứng dụng bằng cách nhập username của mình

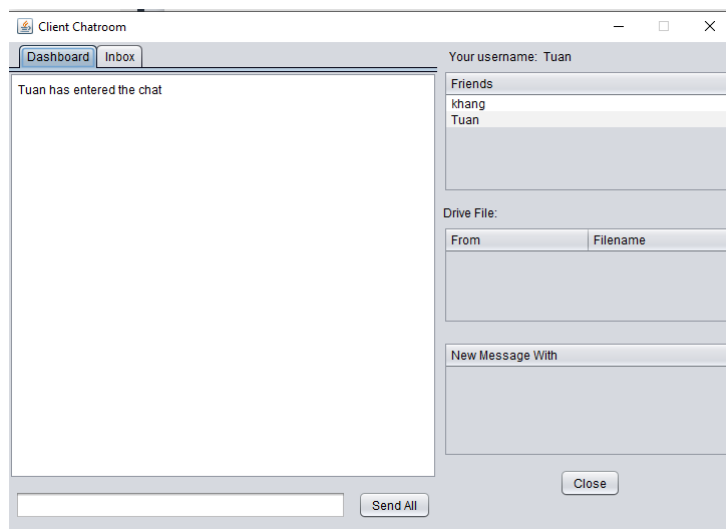


2.3 Register

Màn hình login cũng là màn hình register, người dùng nhập username sau đó chọn register là đã đăng ký được.



2.4 Chatroom



Sau khi truy cập thành công vào ứng dụng, người dùng sẽ thấy được giao diện như hình. Bên phải gồm có khung chat, 2 thẻ Dashboard, Inbox, khu vực nhập tin nhắn và nút gửi tin nhắn.

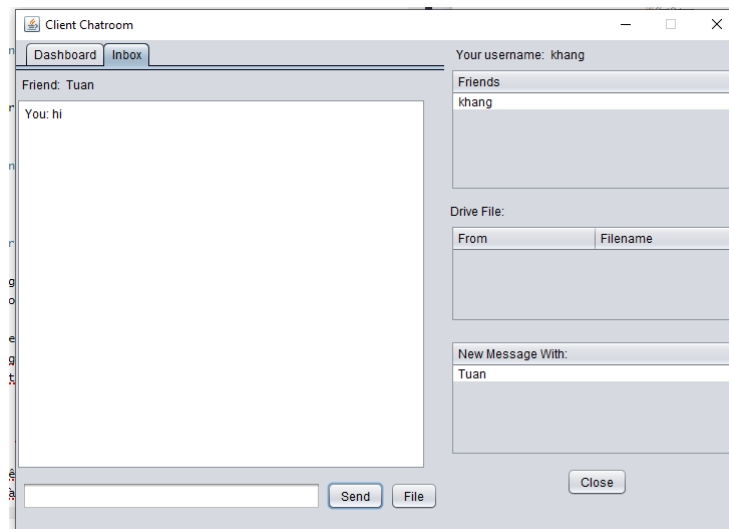
Bên trái gồm có tên người dùng (username), Số người đang online (Friends), Drive File khu vực chứa file mà người dùng nhận được từ người khác, và ô message with thông báo người dùng có tin nhắn mới từ một người dùng khác, nút close để đóng ứng dụng.

2.4.1 Dashboard

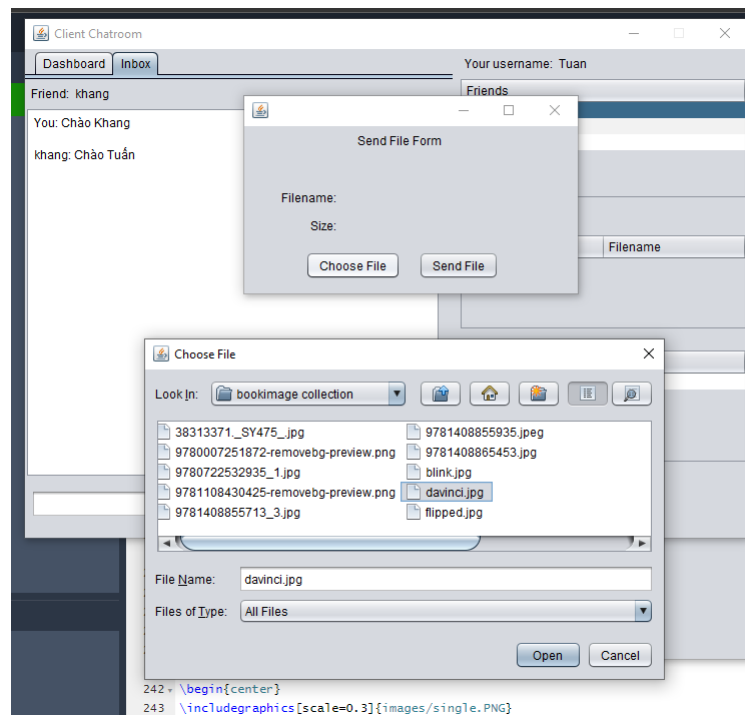
Đối với thẻ Dashboard, giao diện sẽ như hình trên với các chức năng đã được liệt kê.

2.4.2 Inbox

Với thẻ Inbox, khi người dùng bấm vào tên bất kì ở phần Friends, sẽ được chuyển sang thẻ này, thẻ này có thêm chức năng là gửi và truyền nhận file.



Khi chọn thẻ File, hệ thống sẽ pop-up lên một cửa sổ mới. Chọn thẻ Choose File để chọn file, rồi bấm Send File để gửi.



3 Xác định và mô tả các giao thức truyền thông được sử dụng cho từng chức năng

3.1 Mô tả sơ bộ

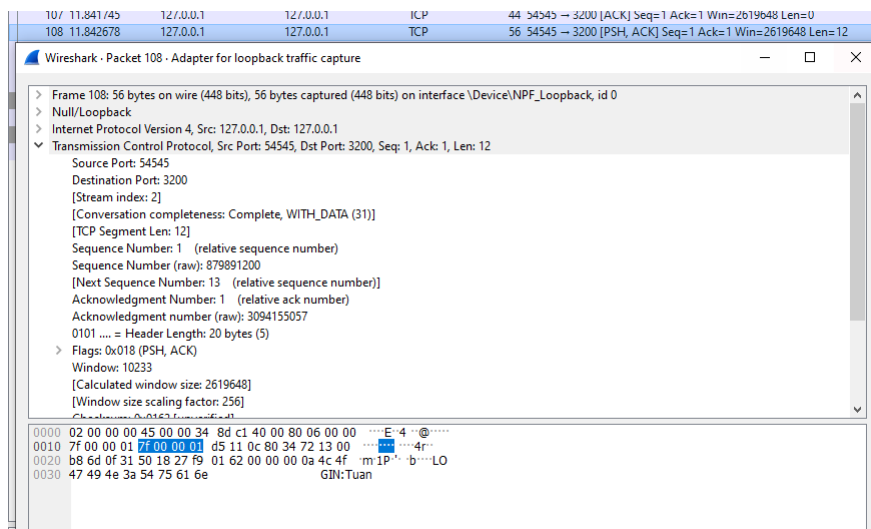
- Đối với các chức năng trong ứng dụng này chúng em đều sử dụng phương thức TCP làm phương thức chính.
- Chúng ta sẽ khởi động một tcpserver, sau đó tcp server này sẽ lắng nghe các hoạt động từ client.
- Các client và server phải kết nối chung một port.

3.2 Mô tả chi tiết

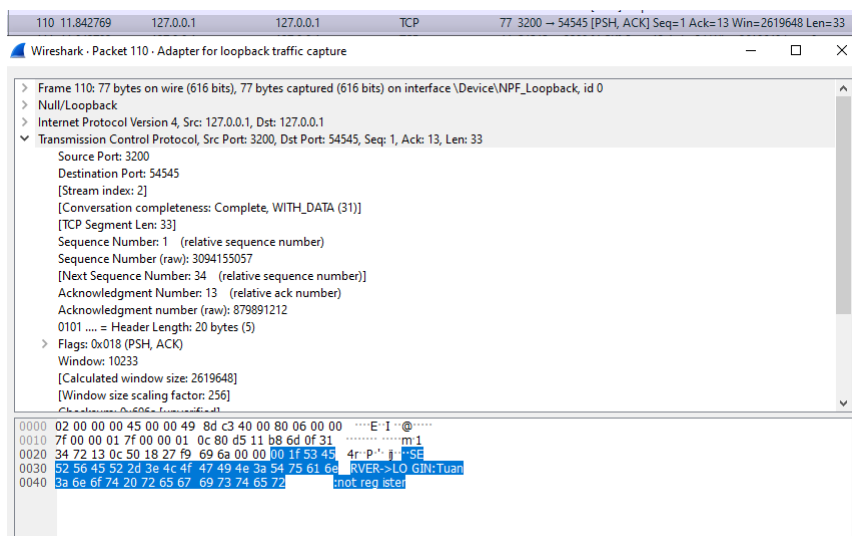
3.2.1 Chức năng Login (Đăng nhập)

- Trường hợp chưa đăng ký

Khi nhập vào một username chưa từng được đăng ký, phía client sẽ gửi một thông điệp có dạng như sau "LOGIN: username" đến phía server.

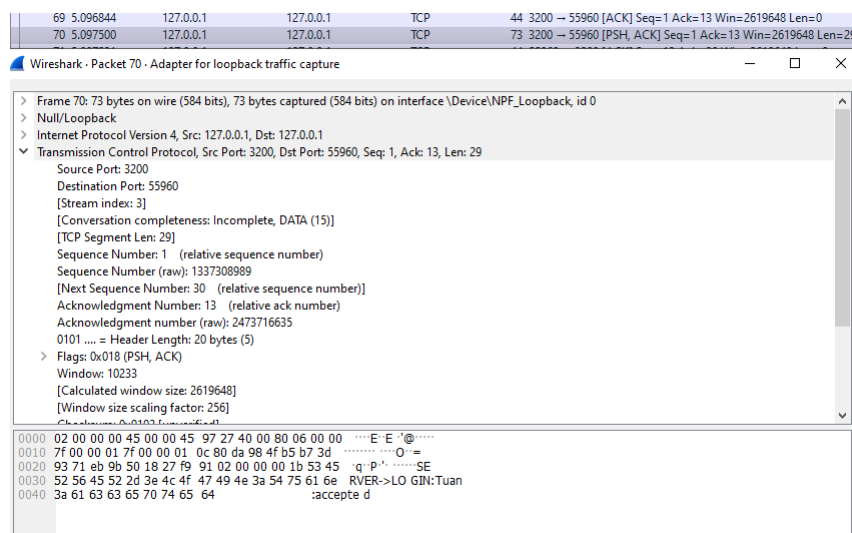


Tiếp theo phía client sẽ tiến hành kiểm tra xem username này đã được đăng ký chưa. Danh sách các username đã đăng ký được lưu trong mảng accountUser. Nếu trong mảng accountUser này không có tên username cần kiểm tra thì server sẽ đáp lại một tin nhắn có dạng "SERVER->LOGIN:" + user + ":not register". Sau đó phía client sẽ hiển thị thông báo đến người vừa nhập username "This username is not registered."

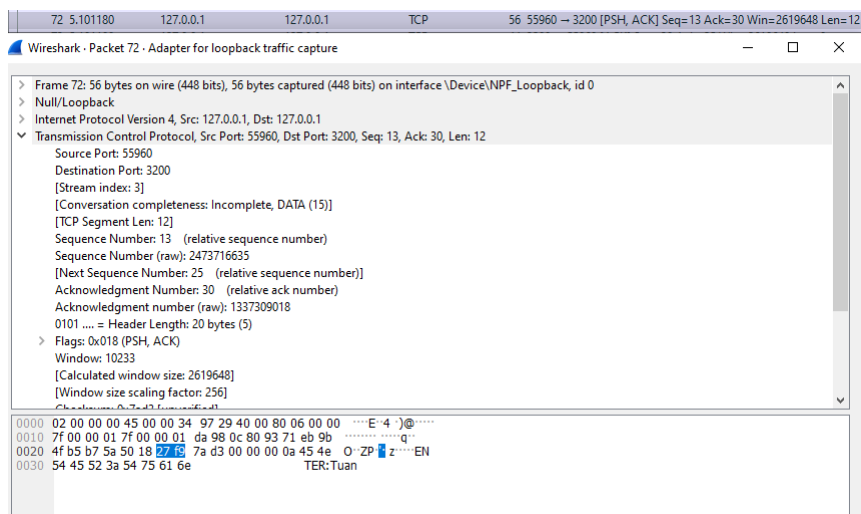


- Trường hợp đã đăng ký nhưng chưa đăng nhập

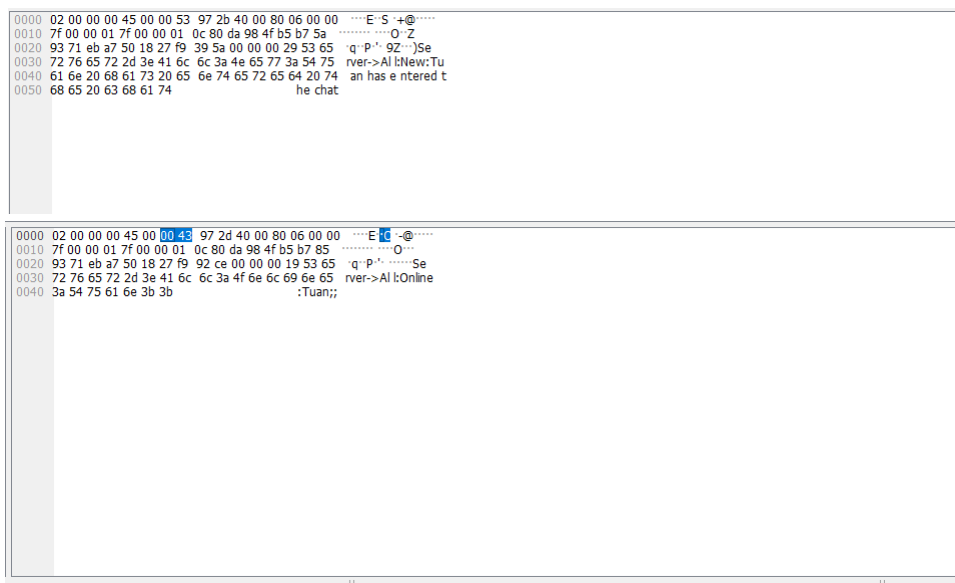
Khi nhập vào một username, phía client cũng gửi đi một thông điệp như ở trên và server cũng tiến hành đi kiểm tra như trên. Ngoài ra server còn kiểm tra thêm là người này có đang online không bằng cách tìm username trong mảng onlineUser. Nếu không có trong mảng này thì server tiến hành thêm username này vào mảng, thêm luồng của username này vào clientHandlers. Server gửi 1 thông điệp thông báo username đã đăng nhập thành công đến client "SERVER->LOGIN:" + user + ":accepted"



Sau khi nhận được thông điệp đăng nhập thành công, client gửi một thông báo dạng "ENTER:" + username" đến server.

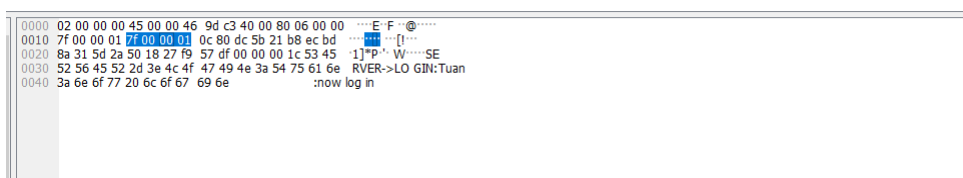


server sau đó tiếp tục gửi các thông điệp "Server->All:New: + username + " has entered the chat" và "Server->All:Online: + onlineList" để thông báo đến client username vừa vào phòng chat và danh sách tất cả username đang online. client sẽ lấy thông tin này để hiển thị lên giao diện.



- Trường hợp đã đăng ký và hiện tại đang đăng nhập (đang online)

Nếu người dùng hiện tại đang online mà lại bấm đăng nhập tiếp thì server sẽ gửi đi một thông điệp có dạng "SERVER->LOGIN: + user + :now login" để thông báo và sẽ không tạo luồng mới cho username này.



3.3 Chức năng Register (Đăng ký)

- **Trường hợp chưa từng đăng ký bao giờ**

Đối với người chưa từng đăng ký, đầu tiên client sẽ gửi đi một thông điệp "REGISTER: + username". Phía server tiếp nhận thông điệp này và tiến hành kiểm tra xem username này đã đăng ký chưa. Nếu chưa đăng ký thì phía server thêm username vào mảng accountUser rồi trả về thông điệp dạng "SERVER->REGISTER: + user + :register successful". Client tiếp nhận thông điệp rồi in ra giao diện thông báo đăng ký thành công.

- **Trường hợp đã từng đăng ký**

Đối với người đã đăng ký rồi mà lại đăng ký tiếp thì client sẽ trả về thông điệp "SERVER->REGISTER: + user + :user existed". Các bước kiểm tra như phía trên.

3.4 Chức năng gửi tin nhắn cho toàn bộ người đang online (thẻ Dashboard)

- **Trường hợp người dùng không nhập gì cả mà bấm gửi**

Với trường hợp này hệ phía client sẽ có hàm để kiểm tra, không có sự xuất hiện giao tiếp giữa client và server.

- **Trường hợp người dùng nhập tin nhắn và bấm gửi**

P phía client gửi một thông điệp dạng "PUBLIC:: + this.username + :: + messageInput", this.username là tên người gửi, messageInput là tin nhắn.

Server gửi tiếp thông điệp "PUBLIC:: + this.username + :: + messageInput" đến client, client sau khi nhận thông điệp này thì kiểm tra có phải là "PUBLIC" không rồi in ra chat table.

3.5 Chức năng gửi tin nhắn cho một người

- **Người dùng nhập tin và bấm gửi**

Khi người dùng bấm gửi, phía client gửi đi thông điệp có dạng "PRIVATE:: + this.username + -> + receiver + :: + messageInput" với this.username là tên người gửi, receiver là tên người nhận, messageInput là tin nhắn.

Sau đó Server tiếp tục gửi thông điệp trên cho cả 2 client là client gửi và client nhận. Các client sau khi nhận được thông điệp trên thì tiến hành tách chuỗi để lấy được tên người gửi (hoặc nhận) và tin nhắn để in ra giao diện.

3.6 Chức năng gửi file cho một cá nhân nào đó

- **Người dùng chưa chọn nhưng bấm gửi file** Không thể gửi được, client sẽ kiểm tra trước khi gửi, không có giao tiếp giữa client và server.

- **Người dùng chọn và gửi file**

Phía client gửi 2 thông điệp "FILE:: + this.username + -> + this.receiver" và "PRIVATE:: + this.username + -> + this.receiver + :: has sent file + filename". Bên cạnh đó client cũng gửi đi các biến fileNameBytes.length, fileNameBytes, fileContentBytes.length, fileContentBytes, là tên và nội dung file được lưu dưới dạng byte ở socket.

```

0000 02 00 00 00 45 00 00 3c c6 c6 40 00 80 06 00 00  ....E<@.....
0010 7f 00 00 01 7f 00 00 01 de a8 0c 80 e8 e7 7a fe  .........z
0020 bb da 74 93 50 18 27 f8 25 21 00 00 00 12 46 49  -tP'-'%l...FI
0030 4c 45 3a 3a 74 75 61 6e 2d 2d 3e 6b 68 61 6e 67  LE::tuan-->khang

0000 02 00 00 00 45 00 00 5d c6 c6 40 00 80 06 00 00  ....E]@.....
0010 7f 00 00 01 7f 00 00 01 de a8 0c 80 e8 e8 05 68  .........z
0020 bb da 74 93 50 18 27 f8 19 53 00 00 00 33 50 52  -tP'-'S...3PR
0030 49 56 41 54 45 3a 3a 74 75 61 6e 2d 2d 3e 6b 68  IVATE::tuan-->kh
0040 61 6e 67 3a 3a 20 68 61 73 20 73 65 6e 74 20 66  ang::ha s sent f
0050 69 6c 65 20 65 6e 74 65 72 43 68 61 74 2e 70 6e  ile ente rChat.pn
0060 67 g

```

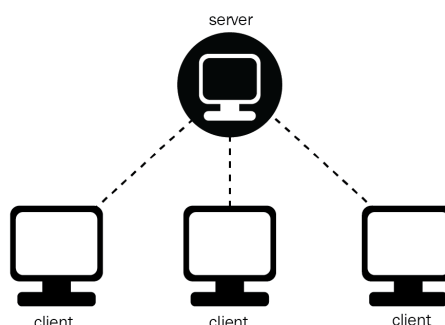
Server nhận được thông điệp từ client, server tiến hành gửi thông điệp "FILE:: + fromUser + "->" + receiver" đến người nhận. Server cũng gửi đến client nhận các dữ liệu của file như trên.

Client nhận (Receiver), tiếp nhận thông tin và đi kiểm tra xem file đó có nằm trong drive chưa, nếu chưa có thì tiếp nhận và đưa vào drive, nếu có rồi thì thông báo cho người dùng có muốn overwrite (ghi đè hay không).

4 Chi tiết thiết kế

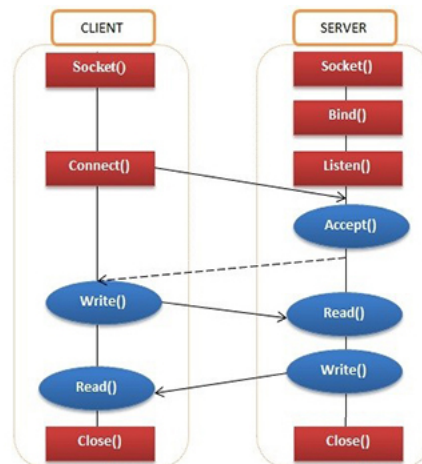
4.1 Mô hình client-server và giao diện lập trình ứng dụng mạng socket

- **Mô hình client-server:** Mô hình Client-Server (hay còn gọi là mô hình máy khách - máy chủ) là mô hình mà ở đây, máy khách là các máy tính, các thiết bị điện tử như máy in, điện thoại bàn, máy fax,... các máy khách này gửi yêu cầu đến server (hay còn được gọi là máy chủ). Máy chủ, xử lý các yêu cầu đó và trả về kết quả

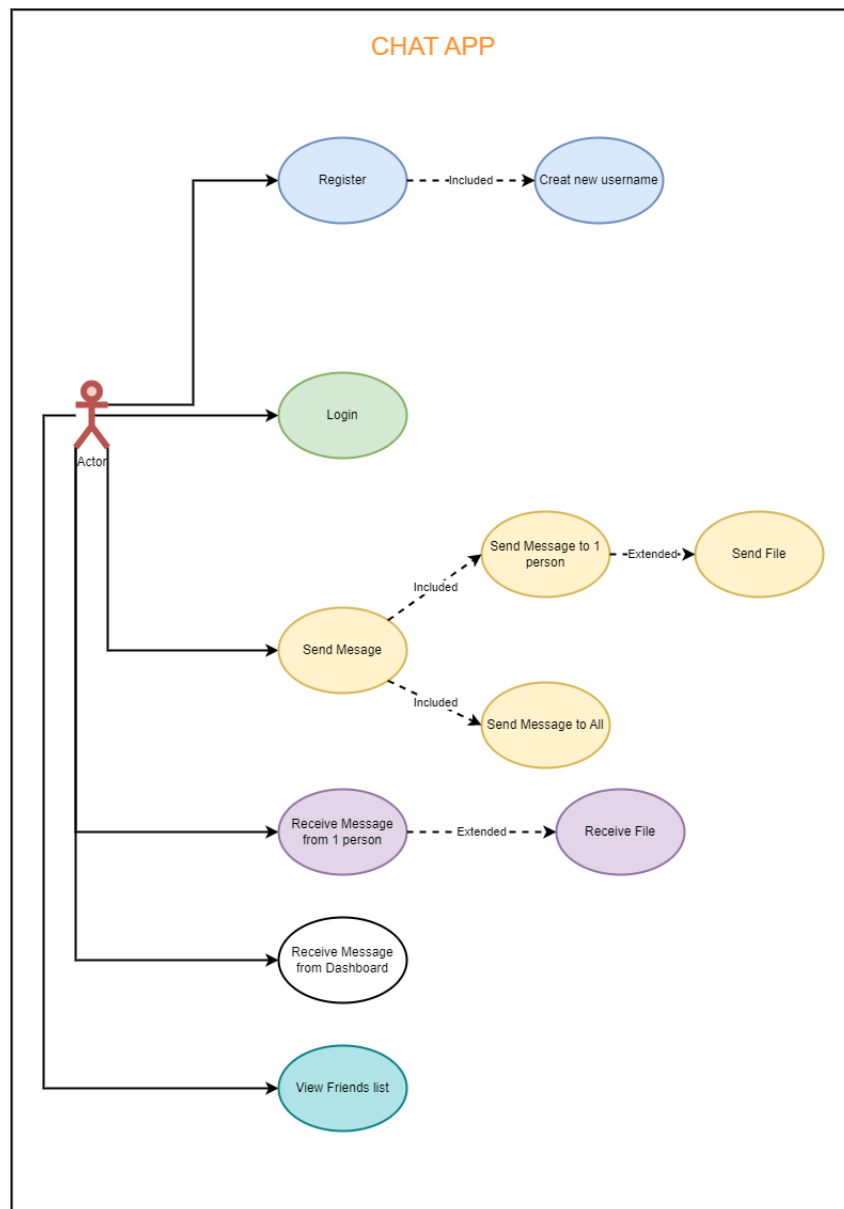


- **Socket:** Socket là một giao diện lập trình ứng dụng mạng được dùng để truyền và nhận dữ liệu trên internet. Giữa hai chương trình chạy trên mạng cần có một liên kết giao tiếp hai chiều để kết nối với nhau. Điểm cuối (endpoint) của liên kết này được gọi là socket, cho phép người dùng kết nối các máy tính truyền tải và nhận dữ liệu từ máy tính thông

qua mạng. Vì vậy, hiểu đơn giản thì socket là thiết bị truyền thông hai chiều gửi và nhận dữ liệu từ máy khác.



4.2 Usecase Diagram





Bảng 1: Usecase «Register»

Usecase	Register
Actors	User
Description	Đăng ký tài khoản để sử dụng app
Preconditions	Port của user phải trùng với port của server User chưa từng đăng ký
Postconditions	User có thể đăng nhập và sử dụng app
Normal flow	Tại trang login, user nhập tên mình tại ô nhập, sau đó bấm register
Alternative flow	không có
Exceptions	User không thể đăng ký bằng tên đã có người đăng ký trước, hệ thống sẽ hiện ra cảnh báo "This use name is existed. Please try another one". hệ thống sẽ trả về trang login nếu người dùng bấm ok

Bảng 2: Usecase «Login»

Usecase	Login
Actors	User
Description	Đăng nhập tài khoản để sử dụng app
Preconditions	Port của user phải trùng với port của server User đã đăng ký
Postconditions	User có thể sử dụng app
Normal flow	Tại trang login, user nhập tên mình tại ô nhập, sau đó bấm login
Alternative flow	không có
Exceptions	User chưa từng đăng ký. Hệ thống sẽ cảnh báo "This user name is not registered". Hệ thống sẽ trả về trang Login khi bấm ok

Bảng 3: Usecase «Send Message»

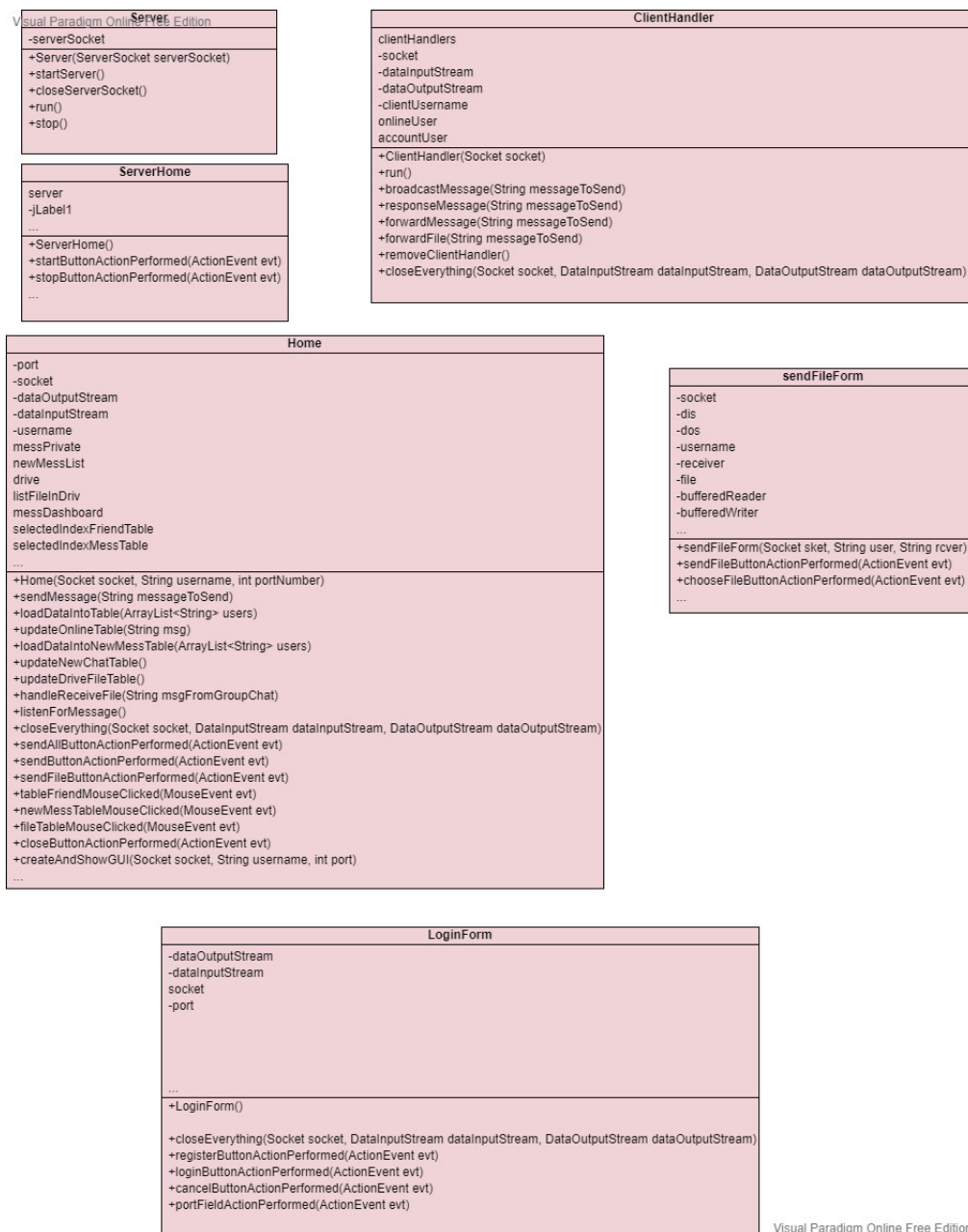
Usecase	Send Message
Actors	User
Description	User gửi tin nhắn
Preconditions	User đã đăng nhập thành công
Postconditions	User có thể gửi tin nhắn
Normal flow	Có hai mục để User chọn để nhắn tin là Dashboard và Inbox. Dashboard: User dùng để chat với tất cả người đang online. Inbox: User dùng để chat riêng với cá nhân. Tính năng chat riêng với cá nhân có thể gửi file được còn tính năng chat chung thì không thể gửi file
Alternative flow	không có
Exceptions	Không có



Bảng 4: Usecase «Receive Message»

Usecase	Receive Message
Actors	User
Description	User nhận tin nhắn
Preconditions	User đã đăng nhập thành công, có ít nhất 2 người đang online trên hệ thống
Postconditions	User có thể nhận tin nhắn
Normal flow	Có hai mục để User chọn để nhận tin nhắn là Dashboard và Inbox. Dashboard: User dùng nhận tin từ tất cả người đang online trong phòng chat nhóm. Inbox: User dùng để nhận tin riêng từ cá nhân. Tính năng chat riêng với cá nhân có thể nhận được file được còn tính năng chat chung thì không nhận file
Alternative flow	không có
Exceptions	Không có

4.3 Class Diagram



5 Kết quả đạt được và đánh giá

5.1 Kết quả đạt được

- Sử dụng được Socket để xây dựng ứng dụng chat local hoạt động thông qua giao thức TCP

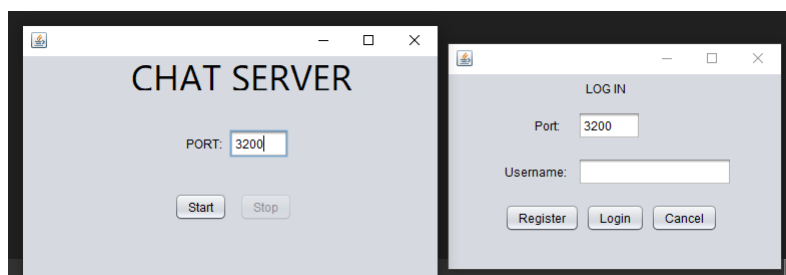
- Ứng dụng chat có 2 tính năng là chat đơn và chat chung
- Ứng dụng có khả năng hiển thị danh sách người online và gửi file.

5.2 Đánh giá

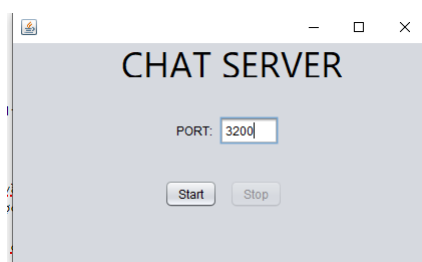
- Chức năng chat và gửi file chưa hiện thực được thông qua giao thức P2P
- Ứng dụng chưa có khả năng chat liên thiết bị
- Ứng dụng chưa có tính bảo mật do chưa có thiết kế password
- Nhóm cần học tập và nghiên cứu thêm để tạo được ứng dụng chat hoàn chỉnh

6 Hướng dẫn sử dụng chung

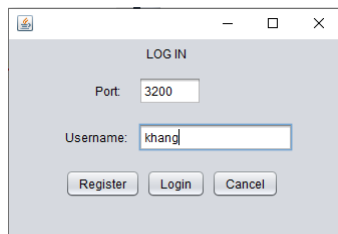
- **Bước 1:** Tải và cài đặt môi trường Java, sau đó tải source code nhóm đã nộp và tìm 2 file client.jar và server.jar. Trong đó server.jar là file dùng để khởi chạy server, client.jar là file dùng để khởi chạy các client.



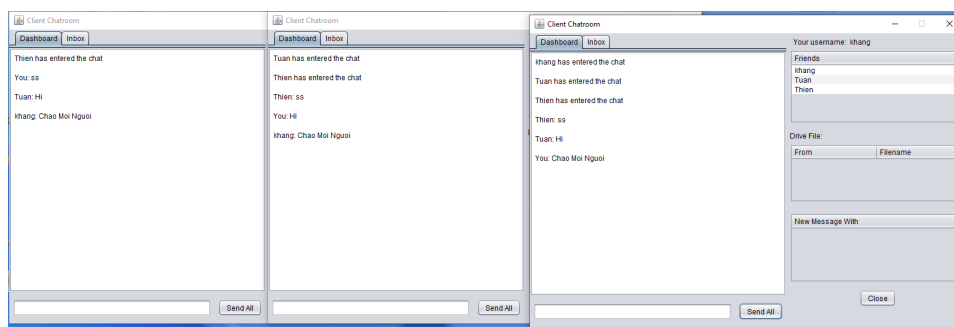
- **Bước 2:** Khởi động server bằng cách bấm start, Port mặc định là 3200 (có thể thay đổi). Nếu server khởi động thành công sẽ hiện chữ "Server start successfully".



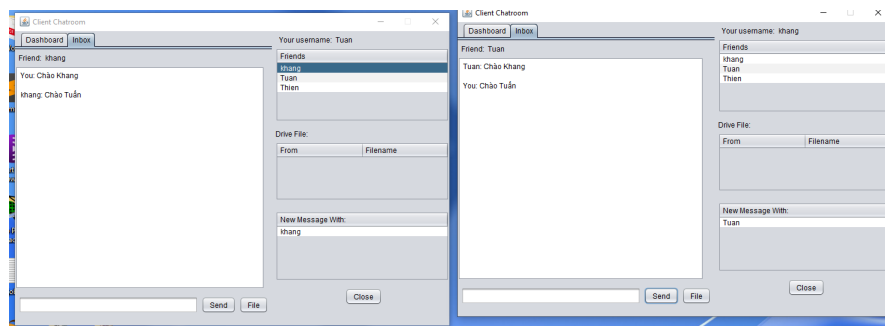
- **Bước 3:** Đăng nhập hoặc đăng ký client. Nếu đã có tên đăng nhập thì gõ vào username rồi bấm Login, còn chưa có và hệ thống báo chưa đăng ký thì nhập username rồi bấm register để đăng ký



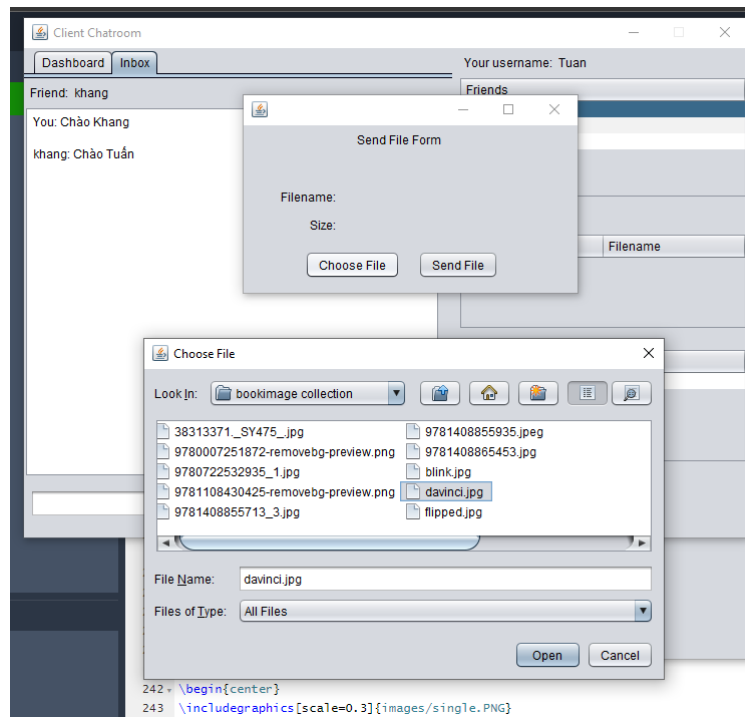
- **Bước 4:** Tham gia chat. Có hai kiểu chat là chat phòng và chat cá nhân bạn bè.
 - **Chat Phòng:** Khi chat phòng ta chọn thẻ Dashboard, tất cả người đang online đều nhận được tin nhắn, chat phòng (Dashboard) chỉ có thể chat văn bản.



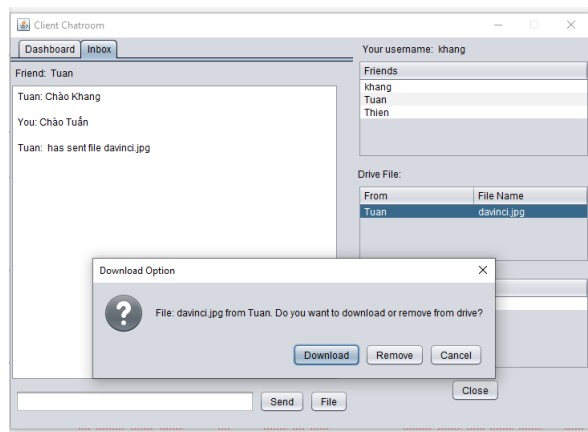
- **Chat cá nhân bạn bè:** Khi muốn chat cá nhân ta chọn tên người mình muốn chat ở bên phải mục Friends, rồi tiến hành chat với người đã được chọn.



- Gửi file trong phần chat cá nhân. Để gửi file ta bấm nút file gần cạnh ô nhập tin nhắn rồi tiến hành chọn file.



- File mà người nhận nhận được sẽ được liệt kê tại Drive File. Người nhận cần nhấp chọn để tải về hoặc xóa khỏi Drive File.



7 Phân công công việc

Tên	MSSV	Công Việc
Nguyễn Hữu Khang	2011365	Hiện thực ứng dụng
Khưu Vĩ Lương	2052589	Liệt Kê, Thiết kế các chức năng cho ứng dụng, trợ giúp phần hiện thực
Trần Sách Nhật	2014009	Viết báo cáo, trợ giúp phần hiện thực ứng dụng, kiểm tra ứng dụng



Tài liệu

- [1] Codelearn “link: <https://codelearn.io/sharing/lap-trinh-ung-dung-chat-noi-bo-don-gian-voi-socket>”, *Lập trình ứng dụng chat nội bộ đơn giản với socket*