



ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
KHOA CÔNG NGHỆ THÔNG TIN

THỰC HÀNH MẠNG MÁY TÍNH

ĐỒ ÁN THỰC HÀNH 1

|Đề tài|

SOCKET: CHAT CONFERENCE

NGUYỄN QUỐC BẢO – 18120286

HÀ MINH CƯỜNG – 18120297

LỚP: 18_3



ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
KHOA CÔNG NGHỆ THÔNG TIN

THỰC HÀNH MẠNG MÁY TÍNH

ĐỒ ÁN THỰC HÀNH 1

|Đề tài|

SOCKET: CHAT CONFERENCE

|Giảng viên hướng dẫn|

Chung Thùy Linh

LỜI CẢM ƠN

Trong thời gian làm đồ án 1, em đã nhận được nhiều sự giúp đỡ, đóng góp ý kiến và chỉ bảo nhiệt tình của thầy cô, gia đình và bạn bè. Em xin gửi lời cảm ơn chân thành đến cô Chung Thùy Linh, giảng viên thực hành môn Mạng Máy Tính, Trường Đại học Khoa Học Tự Nhiên TP.HCM, người đã tận tình hướng dẫn, chỉ bảo em trong suốt quá trình làm tiểu luận. Cuối cùng, em xin chân thành cảm ơn gia đình và bạn bè, đã luôn tạo điều kiện, quan tâm, giúp đỡ, động viên em trong suốt quá trình học tập và hoàn thành bài đồ án 1.

MỤC LỤC

I. Thông tin của nhóm:.....	6
II. Đánh giá mức độ hoàn thành:.....	6
III. Kịch bản giao tiếp của chương trình:	7
IV. Kịch bản trao đổi thông điệp:	10
V. Môi trường lập trình và các framework hỗ trợ để thực thi ứng dụng:.....	10
VI. Hướng dẫn sử dụng tính năng của chương trình:	11
VII. Phân công công việc:.....	18
VIII. Tài liệu tham khảo:.....	19

DANH MỤC HÌNH ẢNH

Hình 1: Server tạo Socket.	7
Hình 2: Client tạo Socket và yêu cầu kết nối tới Server.	7
Hình 3: Trao đổi thông tin giữa Client và Server.	8
Hình 4: Kết thúc phiên làm việc.	8
Hình 5: Toàn bộ phiên làm việc.....	9
Hình 6: Giao diện Server khi chưa nhấn Listen.....	11
Hình 7: Server sau khi đã nhấn Listen.	11
Hình 8: Giao diện của Client.	12
Hình 9: Đăng kí thử với Username là ‘Doan’ và Pass là ‘123’.	12
Hình 10: Server báo là đã đăng kí thành công và hãy đăng nhập lại account.	13
Hình 11: Server đã có ‘Doan’.	13
Hình 12: Đăng nhập với tên không có trong File.	14
Hình 13: Đăng nhập với tên có trong File.	14
Hình 14: Server khi đăng nhập thành công.....	15
Hình 15: Giao diện của Client khi gửi một tin.....	15
Hình 16: Giao diện của Server khi gửi tin.	16
Hình 17: Giao diện khi có ‘Cuong’ LogIn và chat.	16
Hình 18: Giao diện Server khi Bao đăng xuất.	17
Hình 19: Giao diện của những Client còn lại khi Server đóng.....	17
Hình 20: Hoạt động trên Messenger	18
Hình 21: Hoạt động trên Messenger	19

I. Thông tin của nhóm:

- Nhóm gồm hai thành viên:

STT	Họ và tên	MSSV	Số điện thoại	Email
1	Nguyễn Quốc Bảo	18120286	0935742428	18120286@student.hcmus.edu.vn
2	Hà Minh Cường	18120297	0967968362	18120297@student.hcmus.edu.vn

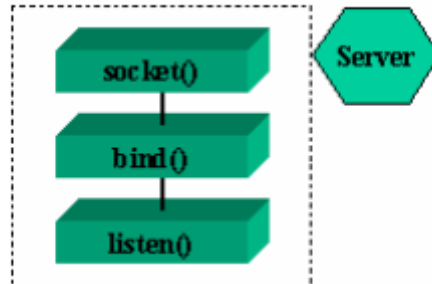
II. Đánh giá mức độ hoàn thành:

Yêu cầu	Mức độ hoàn thành
Cho phép nhiều client đăng nhập cùng một lúc. Số lượng client là không giới hạn.	Hoàn thành – 100%
Client cần đăng kí một username và password để đăng nhập. Nếu username đã được sử dụng, server sẽ yêu cầu client chọn username khác cho đến khi không bị trùng. Client sẽ sử dụng tên này trong toàn bộ quá trình chat. Tại giao diện server và client phải có mục thể hiện danh sách tất cả client đang kết nối đến.	Hoàn thành – 100%
Mọi thông điệp trao đổi của client phải được thể hiện trên màn hình của server và mọi client khác, dưới dạng “username: Thông điệp chat”.	Hoàn thành – 100%
Khi một client đăng nhập hoặc đăng xuất (hoặc thoát chương trình), phải có dòng thông báo trên màn hình server và mọi client.	Hoàn thành – 100%
Nếu server thoát, các client đều phải nhận biết là ngắt kết nối.	Hoàn thành – 100%
Tính năng chat riêng tư.	Chưa hoàn thành
Tính năng gửi file.	Chưa hoàn thành

III. Kịch bản giao tiếp của chương trình:

1. Giai đoạn 1:

- Server tạo Socket, gán số hiệu cổng và lắng nghe yêu cầu kết nối.



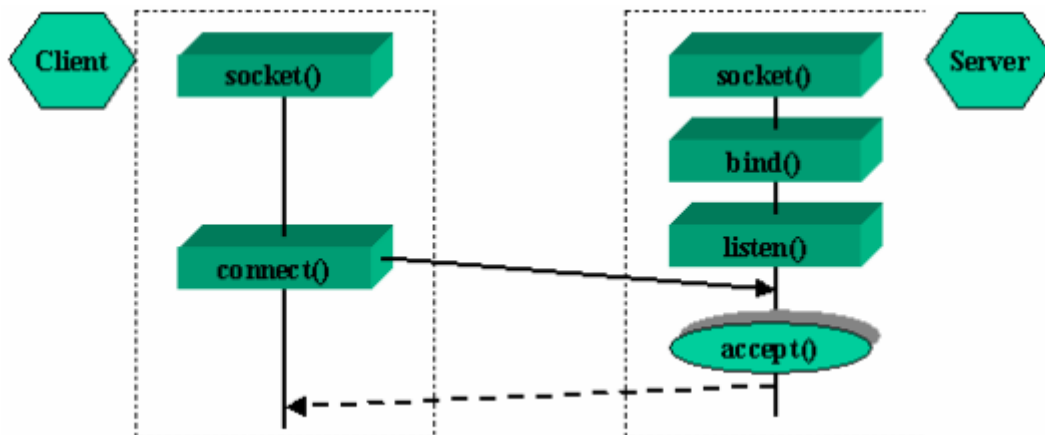
Hình 1: Server tạo Socket.

- socket(): Server yêu cầu tạo một socket để có thể sử dụng các dịch vụ của tầng vận chuyển.
- bind(): Server yêu cầu gán số hiệu port cho socket.
- listen(): Server lắng nghe các yêu cầu kết nối từ các client trên cổng đã được gán.

➔ Server sẵn sàng phục vụ Client.

2. Giai đoạn 2:

- Client tạo Socket, yêu cầu thiết lập một nối kết với Server.

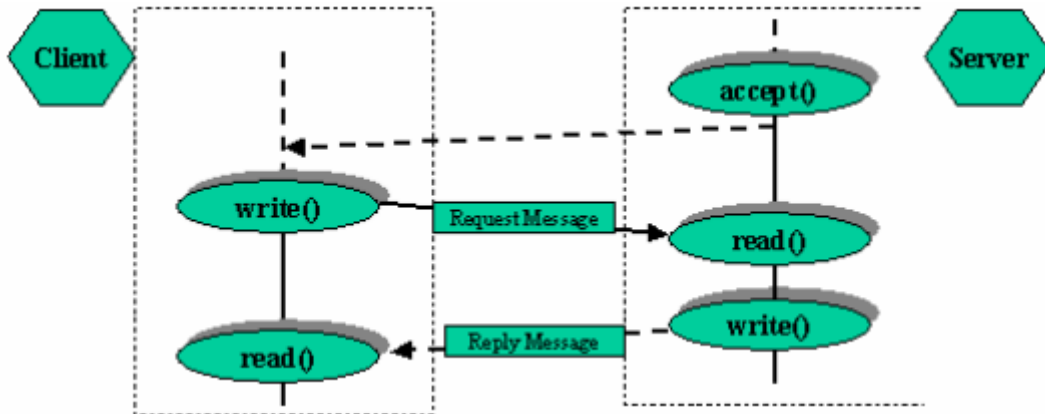


Hình 2: Client tạo Socket và yêu cầu kết nối tới Server.

- socket(): Client yêu cầu tạo một socket để có thể sử dụng các dịch vụ của tầng vận chuyển, thông thường hệ thống tự động gán một số hiệu cổng còn trống cho socket của Client.
- connect(): Client gửi yêu cầu nối kết đến server có địa chỉ IP và Port xác định.
- accept(): Server chấp nhận nối kết của client, khi đó một kênh giao tiếp ảo được hình thành, Client và server có thể trao đổi thông tin với nhau.

3. Giai đoạn 3:

- Trao đổi thông tin giữa Client và Server.

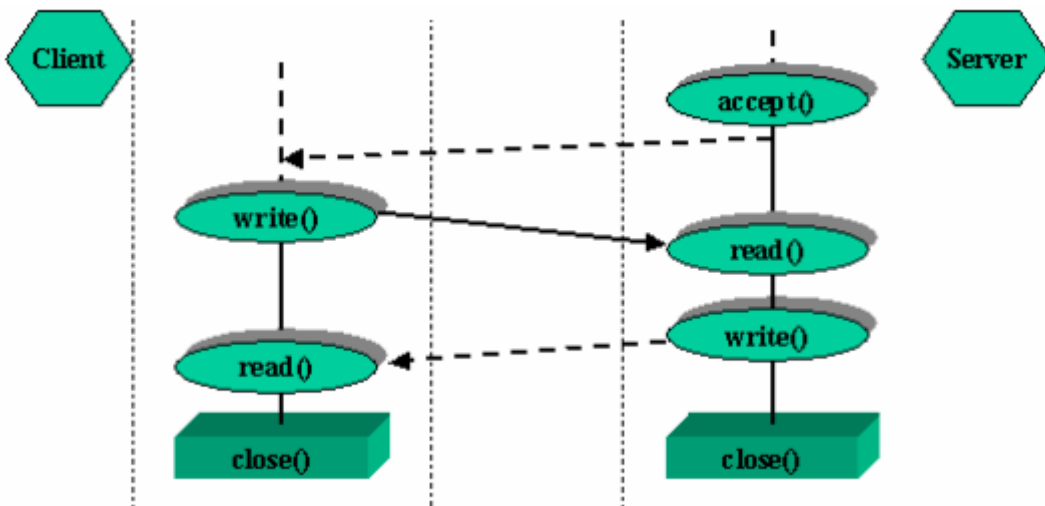


Hình 3: Trao đổi thông tin giữa Client và Server.

- Sau khi chấp nhận yêu cầu nối kết, thông thường server thực hiện lệnh read() để đợi cho đến khi có thông điệp yêu cầu (Request Message) từ client gửi đến.
- Server phân tích và thực thi yêu cầu. Kết quả sẽ được gửi về client bằng lệnh write().
- Sau khi gửi yêu cầu bằng lệnh write(), client chờ nhận thông điệp kết quả (ReplyMessage) từ server bằng lệnh read().

4. Giai đoạn 4:

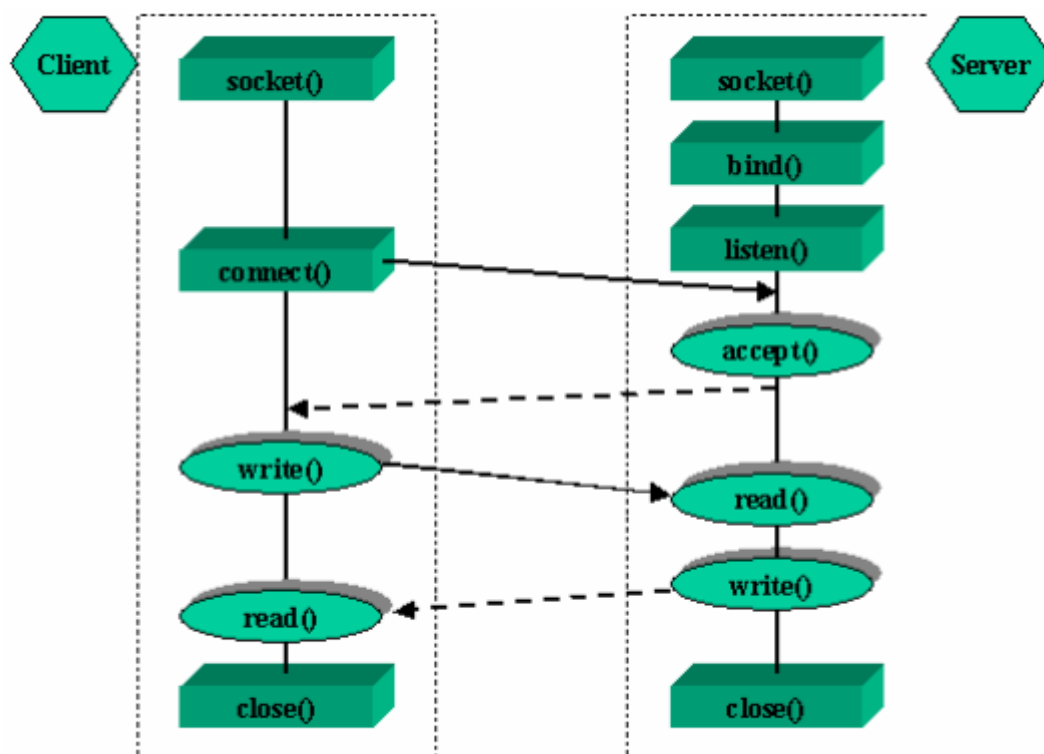
- Kết thúc phiên làm việc.



Hình 4: Kết thúc phiên làm việc.

- Các câu lệnh read(), write() có thể được thực hiện nhiều lần (ký hiệu bằng hình elipse).
- Kênh ảo sẽ bị xóa khi Server hoặc Client đóng socket bằng lệnh close().

➔ Như vậy toàn bộ tiến trình diễn ra như sau:



Hình 5: Toàn bộ phiên làm việc.

IV. Kịch bản trao đổi thông điệp:

1. Listen:

- Server bắt đầu hoạt động và đọc những account có trong file cũng như bắt đầu lắng nghe.

2. Đăng nhập:

- User nhấn đăng nhập phải được Server kiểm tra xem có tồn tại User đó hay không. Nếu có tồn tại thì Server sẽ báo là User đăng nhập thành công và đưa tên User vào khung Online Chat để hiển thị.

3. Đăng kí:

- Nếu đăng kí trùng với tên có trong file thì sẽ báo là đã tồn tại tên User đó rồi.
- Nếu đăng kí thành công thì Server sẽ nhờ người dùng đăng nhập lại tài khoản vừa đăng kí và thực hiện các thao tác tương tự đăng nhập.

4. Đăng xuất:

- Server sẽ loại bỏ thông tin của người dùng ra khỏi danh sách các User đang online.
- Server sẽ gửi 1 tin dưới dạng là “Tên User” LogOut vào khung chat chung.

5. Gửi tin nhắn chung:

- Tin nhắn sẽ được Client gửi cho Server và nội dung của tin nhắn sẽ được Server gửi cho tất cả các Client khác để cập nhật trong khung chat.

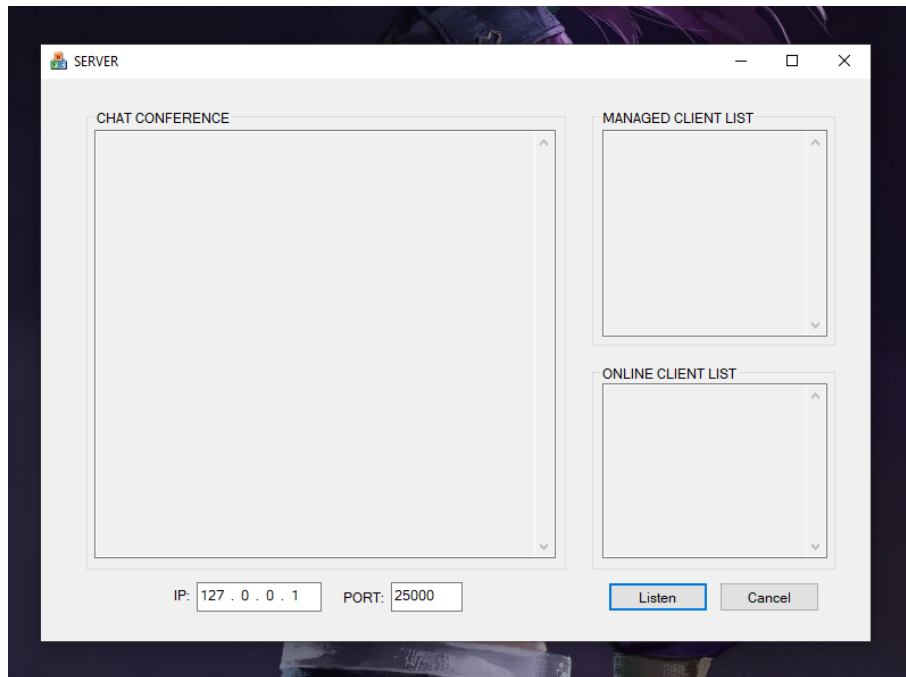
V. Môi trường lập trình và các framework hỗ trợ để thực thi ứng dụng:

- Phát triển ứng dụng dựa trên Demo: NonBlocking.
- Môi trường lập trình: Microsoft Foundation Class Library (MFC).
- Công cụ lập trình: Microsoft Visual Studio 2019.
- Môi trường hoạt động: Windows 10 Pro.

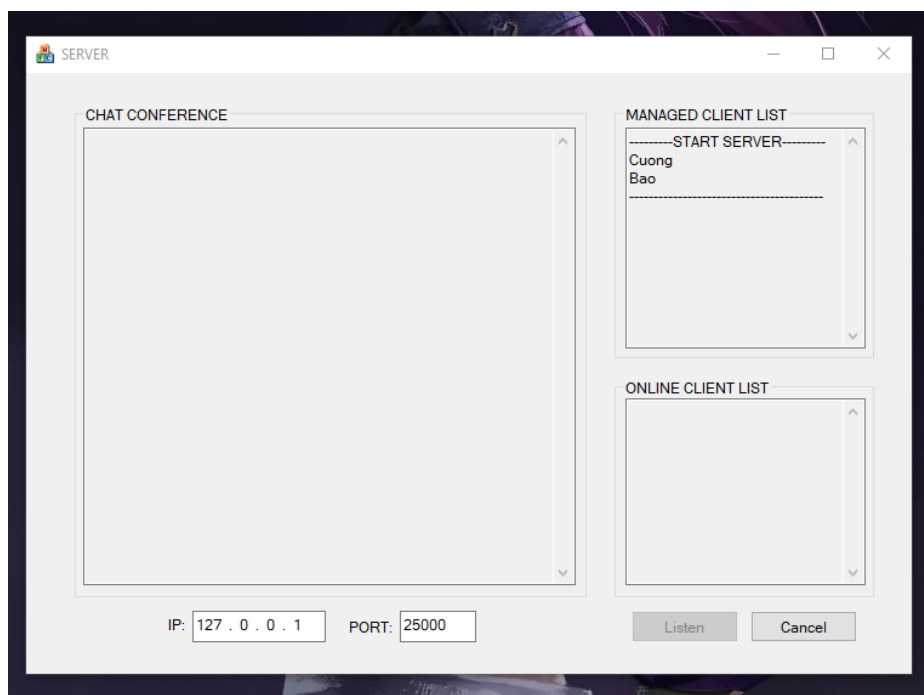
VI. Hướng dẫn sử dụng tính năng của chương trình:

1. Start Server:

- Mở giao diện của Server và nhấn vào nút Listen để start Server và Server sẽ lắng nghe cũng như hiển thị các Client đã đăng kí được lưu ở trong file Account.txt.



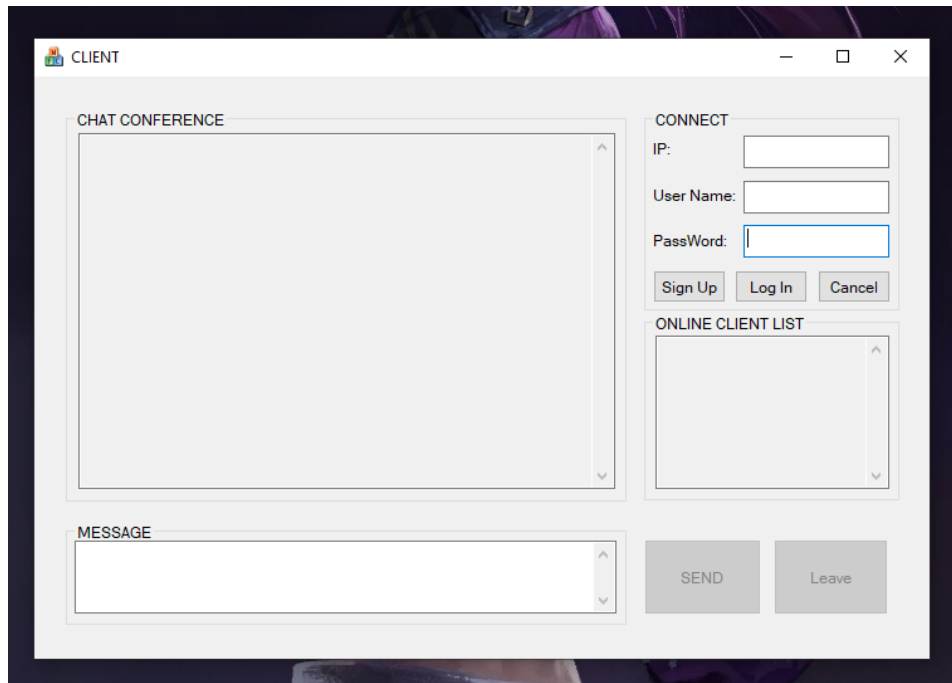
Hình 6: Giao diện Server khi chưa nhấn Listen.



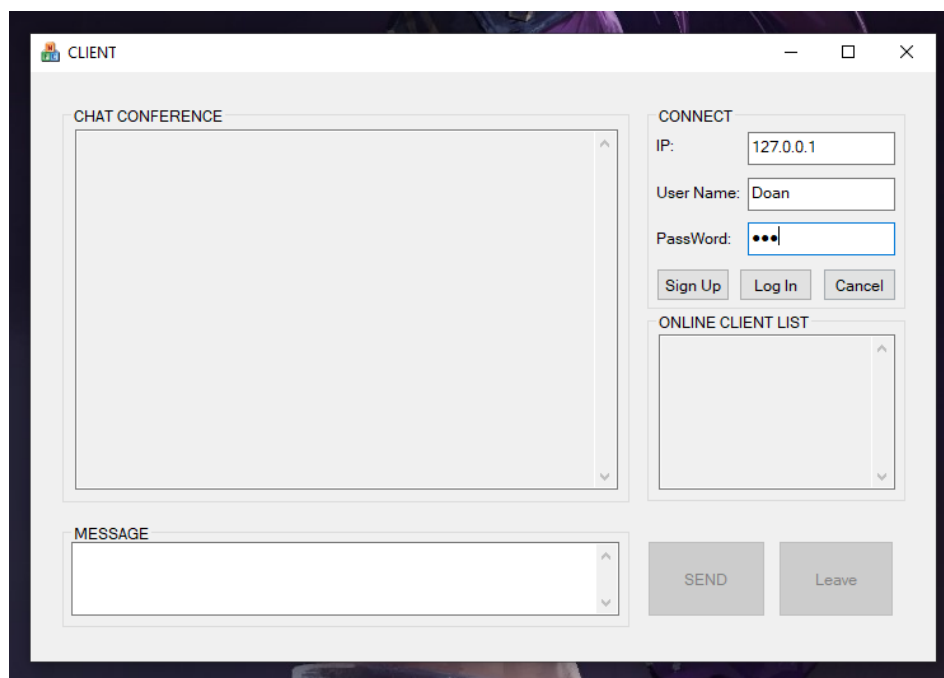
Hình 7: Server sau khi đã nhấn Listen.

2. Đăng kí (Sign Up):

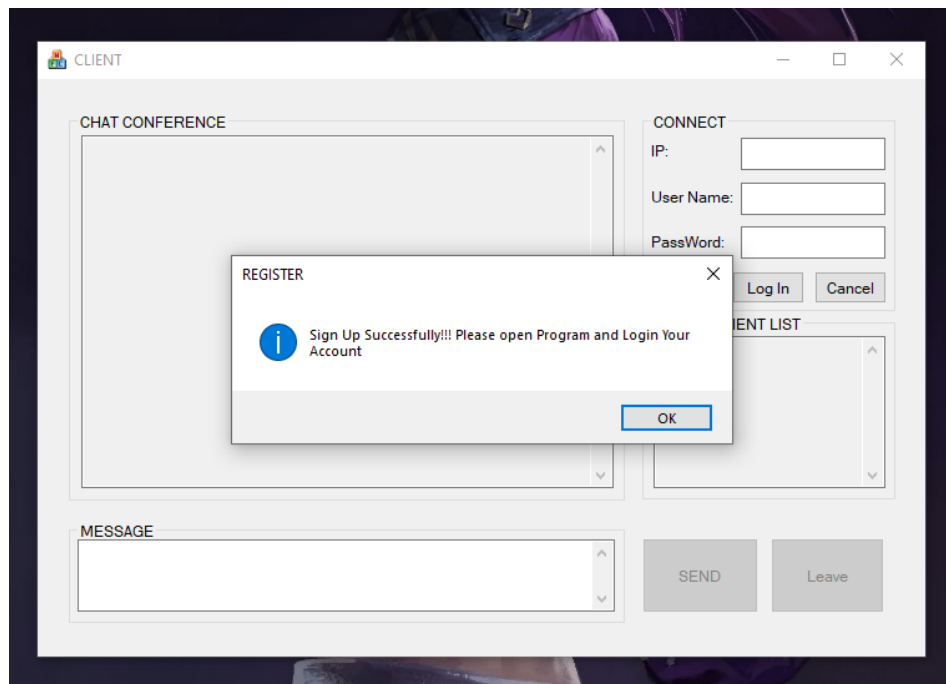
- Mở giao diện của Client, nhập địa chỉ IP, Username và Password muốn đăng kí và nhấn vào Sign Up để tiến hành đăng kí. Nếu tên Username đã tồn tại trong File thì sẽ thông báo đã tồn tại và Client phải tiến hành Sign Up lại.



Hình 8: Giao diện của Client.



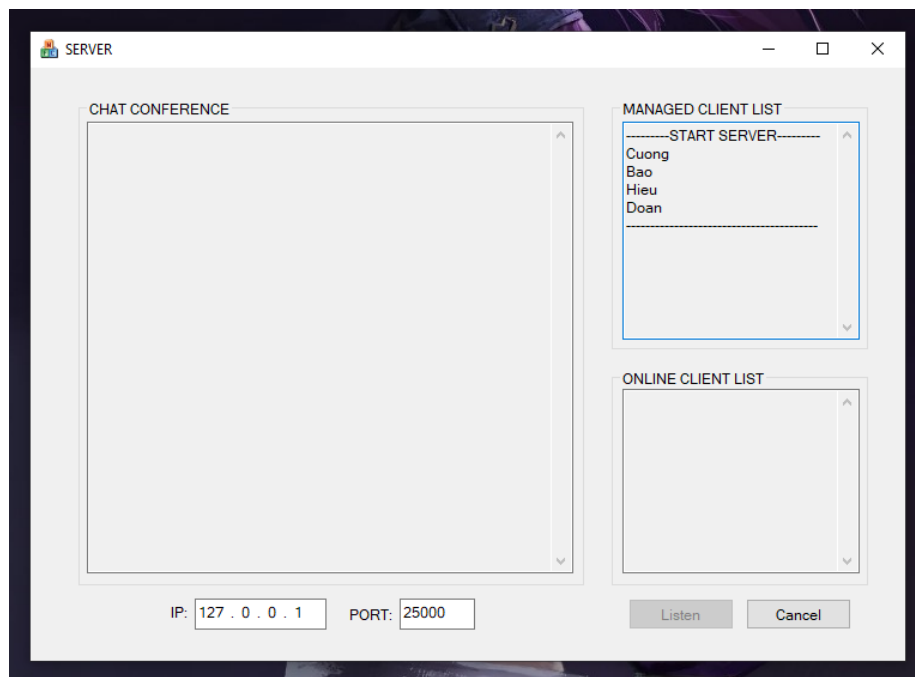
Hình 9: Đăng kí thử với Username là 'Doan' và Pass là '123'.



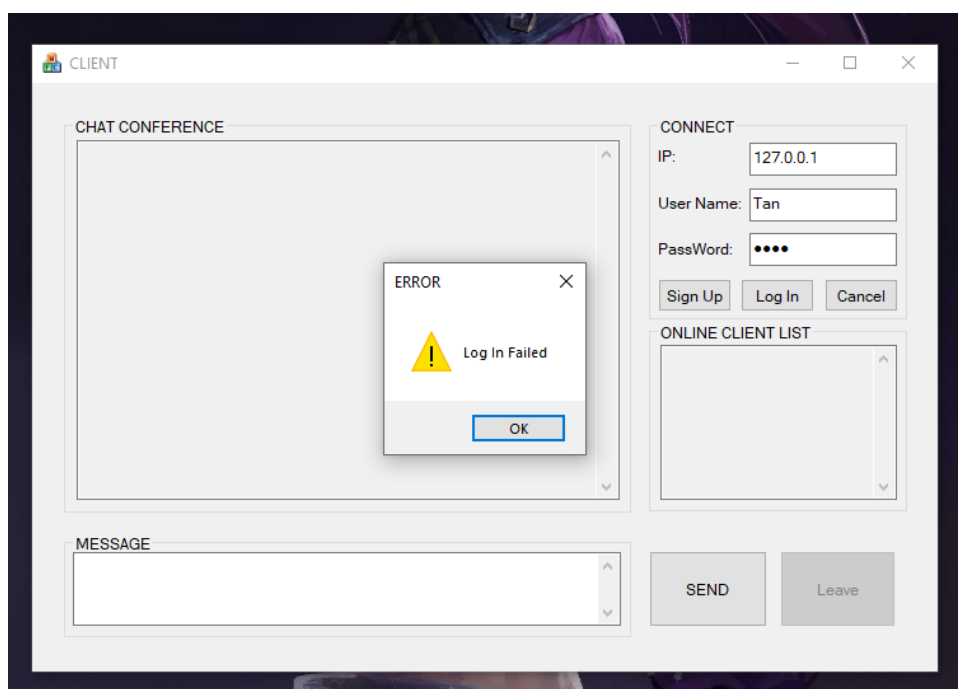
Hình 10: Server báo là đã đăng kí thành công và hãy đăng nhập lại account.

3. Đăng nhập (Log In):

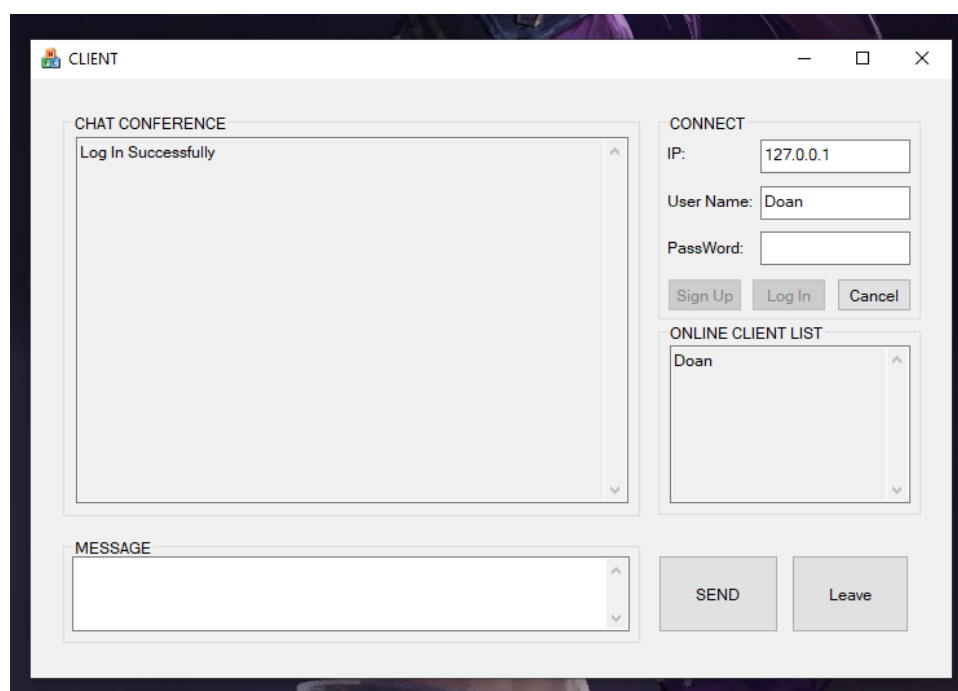
- Tại giao diện Client nhập IP, Username, Password và nhấn Log In. Nếu Username và Password trùng với Username và Password được lưu trữ trong File thì sẽ đăng nhập thành công còn ngược lại thì thông báo lỗi.
- Em sẽ đăng nhập với Username là 'Doan' và Password là '123' và một Username lạ khác.



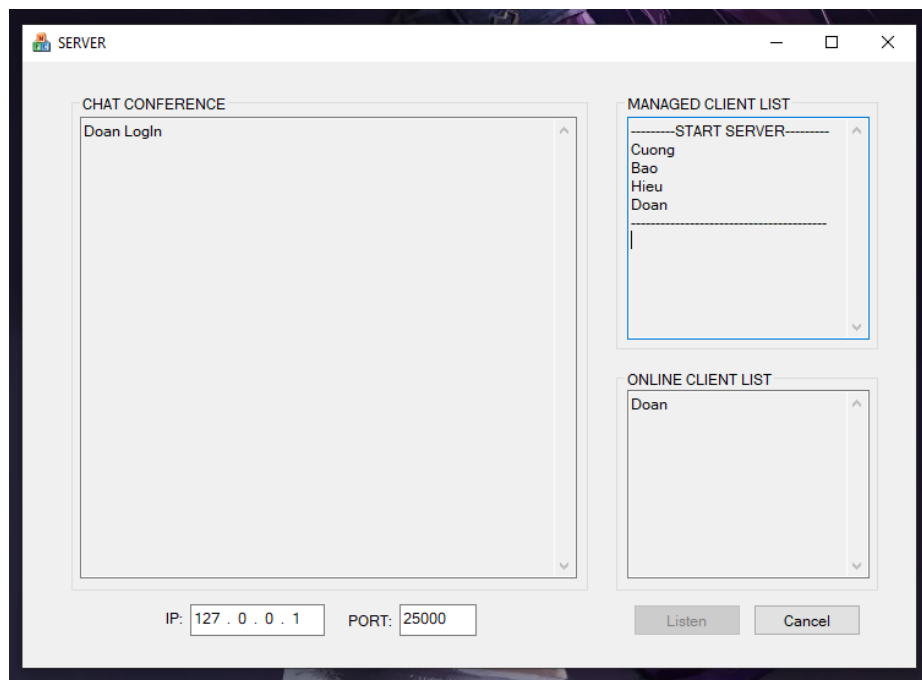
Hình 11: Server đã có 'Doan'.



Hình 12: Đăng nhập với tên không có trong File.



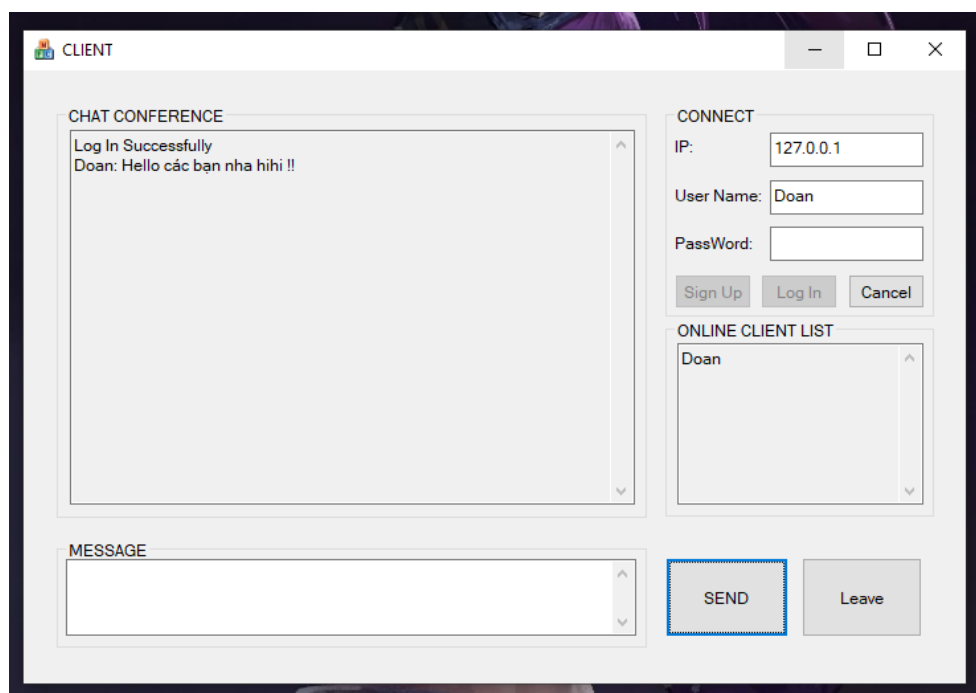
Hình 13: Đăng nhập với tên có trong File.



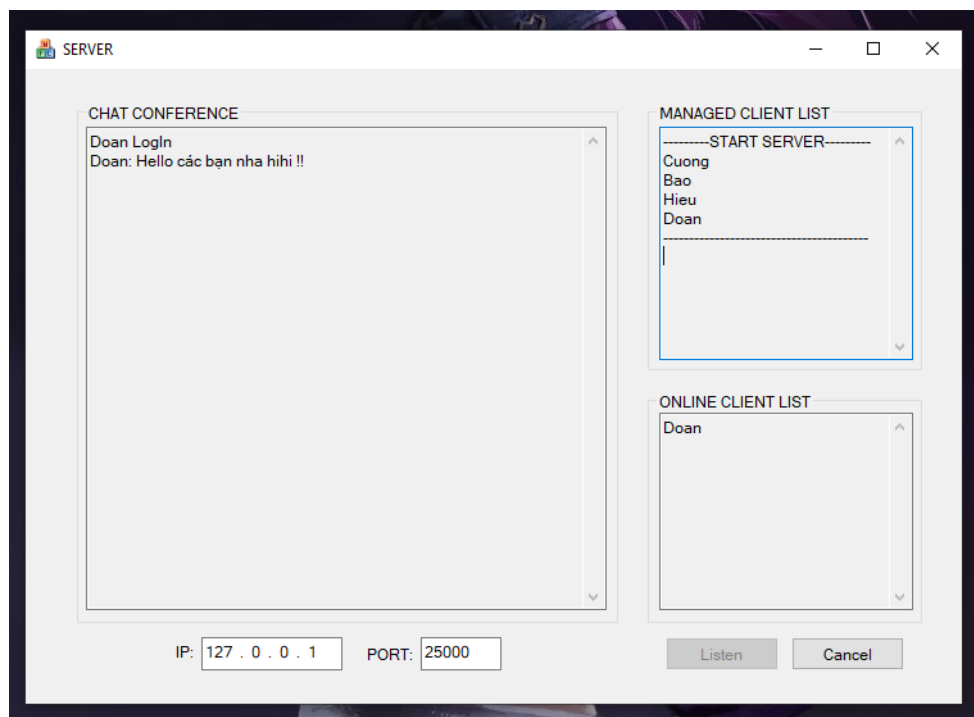
Hình 14: Server khi đăng nhập thành công.

4. Gửi tin nhắn chung (Public Chat):

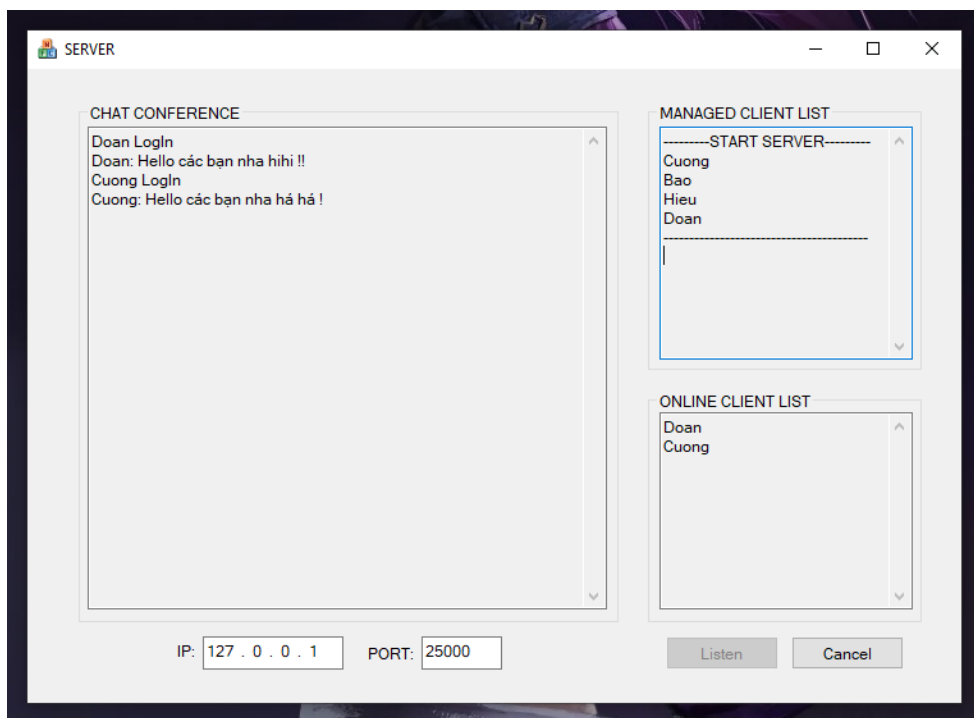
- Nhấn một tin trong khung MESSAGE và SEND. Sẽ gửi đến Server và Server sẽ gửi đến tất cả các Client khác.



Hình 15: Giao diện của Client khi gửi một tin.



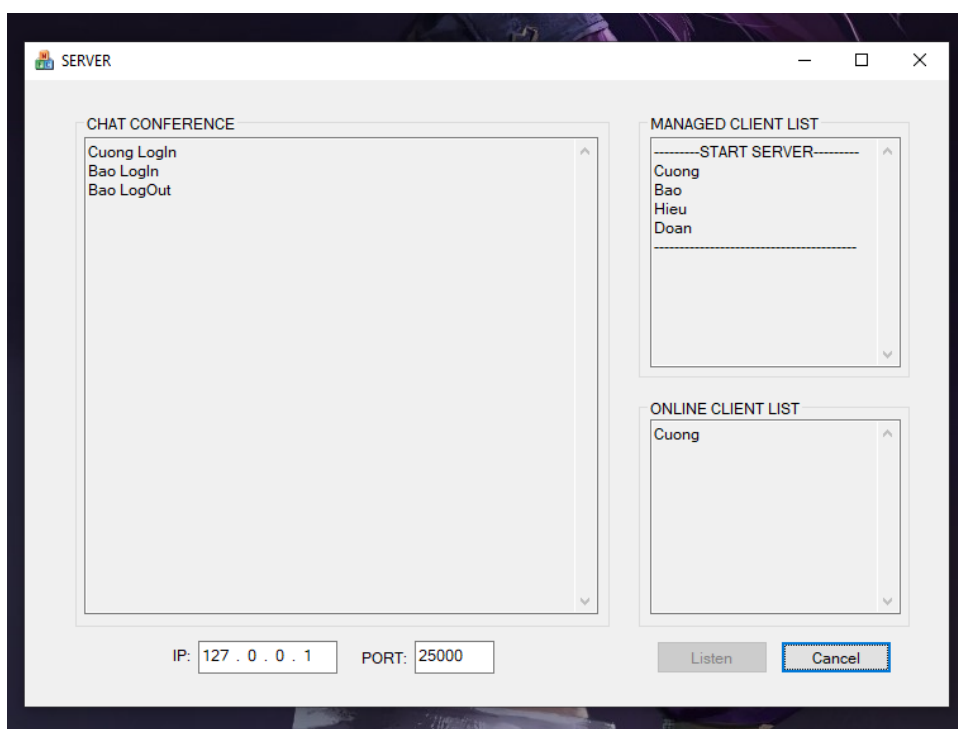
Hình 16: Giao diện của Server khi gửi tin.



Hình 17: Giao diện khi có 'Cuong' Login và chat.

5. Đăng xuất (Log out or Leave Chat Conference):

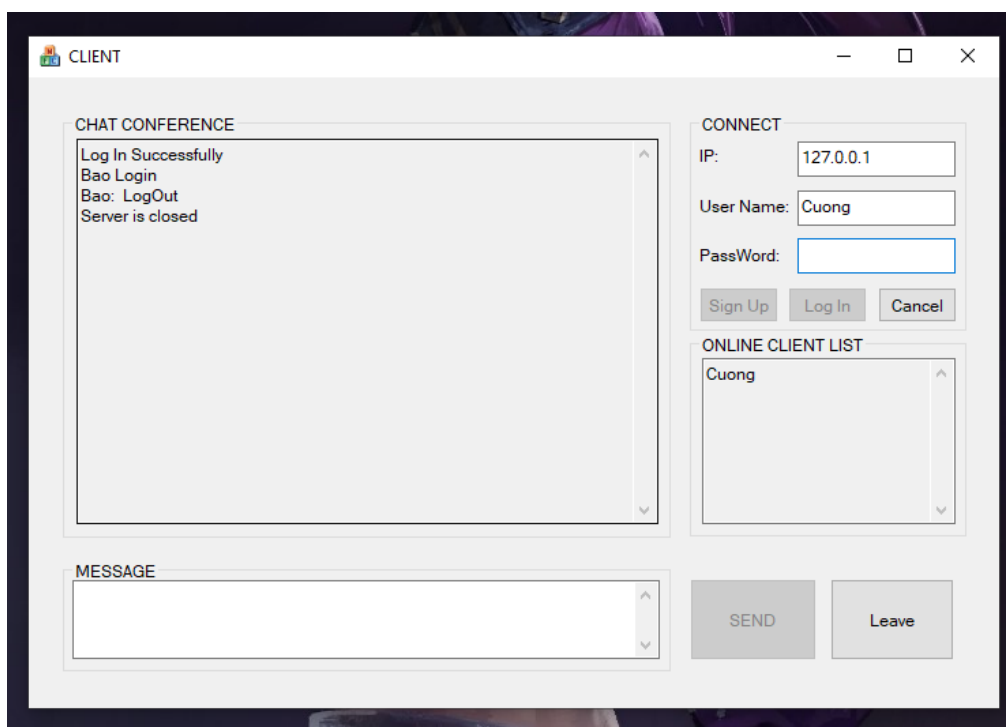
- Nhấn phím Leave hoặc Cancel trong giao diện Client để đăng xuất khỏi chương trình.



Hình 18: Giao diện Server khi Bao đăng xuất.

6. Close Server:

- Nhấn vào nút Cancel trong giao diện Server để đóng kết nối Server và các Client khác không thể hoạt động.



Hình 19: Giao diện của những Client còn lại khi Server đóng.

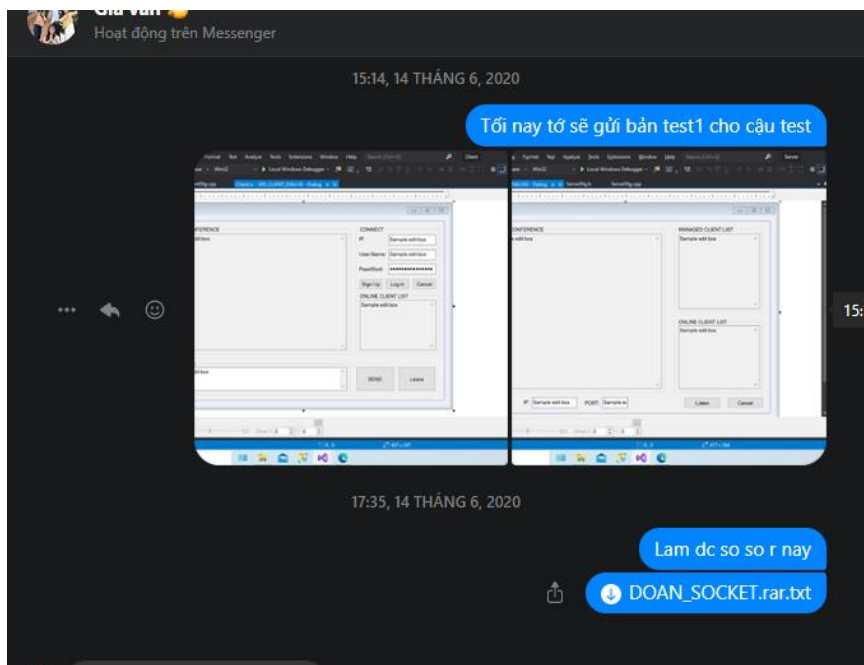
VII. Phân công công việc:

1. Bảng phân công công việc:

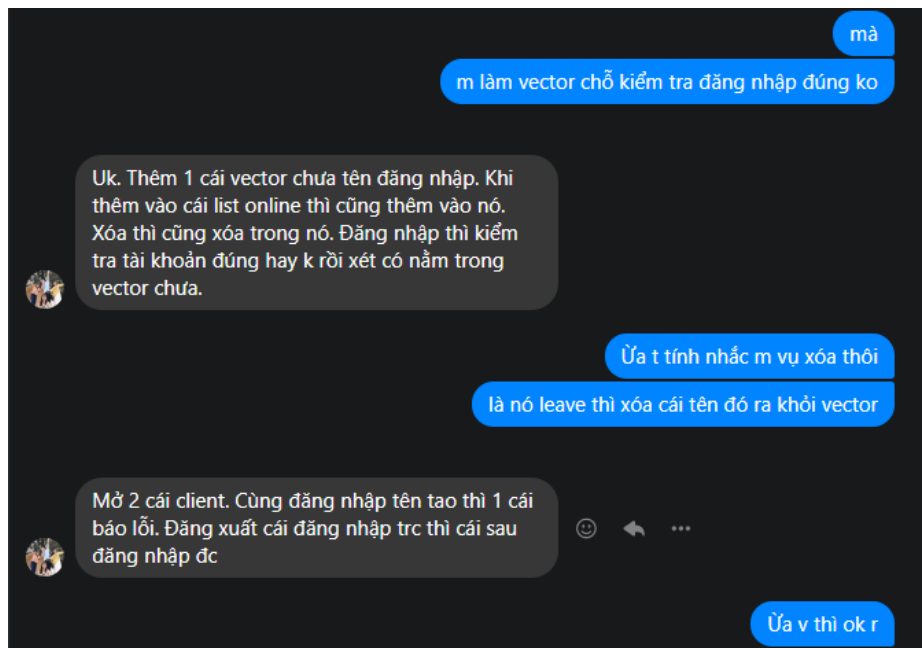
STT	Họ và tên	Công việc
1	Nguyễn Quốc Bảo	<ul style="list-style-type: none"> Kiểm tra người dùng có đăng nhập chưa. Nếu đã đăng nhập rồi thì thông báo là tài khoản này đã có người đăng nhập. Chức năng đăng nhập. Thông báo khi server đóng kết nối. Hiển thị những người đang online (khi đăng xuất thì xóa) Gửi file.
2	Hà Minh Cường	<ul style="list-style-type: none"> Giao diện Client và Server trên MFC, thiết kế Dialog. Chức năng đăng kí, nếu trùng thì thông báo trùng. Nội dung khi các client chat trên chat conference. Thông báo là client đó đăng nhập hay đăng xuất. Hiển thị những người đang online (khi đăng nhập thì thêm). Chat riêng tư. Viết báo cáo.

2. Hoạt động nhóm:

- Hình thức liên lạc: Messenger.



Hình 20: Hoạt động trên Messenger



Hình 21: Hoạt động trên Messenger

VIII. Tài liệu tham khảo:

- DemoSocket: NonBlocking.
- Tài liệu: CSocket.pdf, Socket.pdf, sockets Programming in C++.pdf.
- <http://www.hoangvancong.com/2017/03/03/mfc-socket-in-mfc/>.
- <https://stackoverflow.com/questions/13883329/mfc-c-socket-programming-how-to-connect-server-and-client>.