

MÔN HỌC: MẠNG MÁY TÍNH
LỚP CỬ NHÂN TÀI NĂNG 2017

BÁO CÁO ĐỒ ÁN LẬP TRÌNH SOCKET

Nhóm thực hiện:

1712152 | Nguyễn Thị Mai Thanh

1712228 | Phạm Việt Nga

1712807 | Nguyễn Thị Minh Thùy

MỤC LỤC

| | |
|--|----|
| 1. YÊU CẦU ĐỒ ÁN | 2 |
| 2. KỊCH BẢN CHƯƠNG TRÌNH | 2 |
| 3. CẤU TRÚC GÓI TIN GỬI NHẬN..... | 3 |
| 3.1. Các gói tin gửi từ client lên server..... | 4 |
| 3.2. Các gói tin gửi từ server về client | 4 |
| 4. CẤU TRÚC CHƯƠNG TRÌNH | 5 |
| 4.1. Server | 5 |
| 4.2. Client | 6 |
| 5. CÁC HÀM CHỨC NĂNG | 6 |
| 5.1. Đăng ký tài khoản | 6 |
| 5.2. Đăng nhập: | 7 |
| 5.3. Chat nhóm..... | 7 |
| 5.4. Chat riêng tư:..... | 10 |
| 5.5. Gửi tập tin: | 11 |
| 5.6. Ngắt kết nối: | 12 |
| 6. GIAO DIỆN CHƯƠNG TRÌNH | 12 |
| 7. TỰ ĐÁNH GIÁ | 20 |
| 8. TÀI LIỆU THAM KHẢO | 21 |

1. YÊU CẦU ĐỒ ÁN

Xây dựng chương trình mô phỏng ứng dụng chat gồm 1 Server và nhiều Client với các chức năng:

- Đăng ký, đăng nhập tài khoản cho client.
- Chat nhóm giữa nhiều client.
- Chat riêng tư giữa 2 client.
- Gửi tập tin giữa 2 client khi chat riêng tư.
- Giao diện ứng dụng.

2. KỊCH BẢN CHƯƠNG TRÌNH

Server tạo một socket để chờ nhận kết nối từ các client. Mỗi khi có client mới kết nối, server sẽ tạo một tiến trình (thread) mới để lắng nghe và nhận tin nhắn từ client đó. Khi client không còn kết nối sẽ loại client đó ra khỏi danh sách. Khi nhận được gói tin từ thì thực hiện theo yêu cầu. Server sẽ quản lý mọi hoạt động của các client.

2.1. Chức năng đăng ký, đăng nhập tài khoản

Client mới kết nối tới server sẽ đăng ký tài khoản bằng cách gửi username, password cho server. Server sẽ kiểm tra trong cơ sở dữ liệu hiện tại. Nếu username chưa tồn tại, server sẽ cập nhật client vào dữ liệu hiện có và gửi thông báo đăng ký thành công tới client. Ngược lại, nếu tài khoản đã tồn tại, server sẽ gửi thông báo username không hợp lệ tới client và yêu cầu đăng ký lại.

Tương tự khi đăng kí, client gửi username, password cho server. Server sẽ kiểm tra trong cơ sở dữ liệu hiện tại. Nếu username chưa tồn tại, server sẽ đưa ra thông báo tài khoản chưa tồn tại và yêu cầu đăng kí. Ngược lại, nếu tài khoản đã tồn tại và password thỏa, server sẽ gửi thông báo đăng nhập thành công. Nếu password sai server sẽ gửi thông báo về client báo đã nhập sai password.

2.2. Chức năng chat nhóm

Client gửi yêu cầu tạo nhóm chat lên server kèm theo tên nhóm chat muốn tạo. Nếu tên nhóm chat không bị trùng với các nhóm chat hiện có, server sẽ tạo nhóm chat và thêm username gửi yêu cầu vào nhóm chat đồng thời gửi thông báo tạo nhóm chat thành công về cho client. Nếu nhóm chat đã tồn tại, server gửi thông báo tạo nhóm chat thất bại cho client để client thông báo cho người dùng.

Khi muốn thêm thành viên mới vào nhóm chat, client gửi yêu cầu kèm theo username của thành viên mới. Nếu username đã được đăng ký và chưa có trong nhóm chat, server sẽ thêm thành viên mới và gửi thông báo thêm thành viên thành công cho client, đồng thời gửi lệnh tạo nhóm chat cho thành viên mới, cũng như thông báo cho các thành viên khác trong nhóm là thành viên mới được thêm vào nhóm. Ngược lại, nếu username chưa được đăng ký hoặc đã tồn tại trong group chat, server gửi thông báo thêm thành viên thất bại cho client.

Khi một client gửi tin nhắn trong group chat thì tin nhắn sẽ được gửi lên server trước, sau đó server gửi tin nhắn cho tất cả những thành viên có trong nhóm chat đó.

Khi một client rời nhóm chat (bằng cách tắt cửa sổ nhóm chat), server sẽ loại thành viên đó ra khỏi nhóm chat và thông báo tới các thành viên còn lại. Nếu thành viên cuối cùng của nhóm chat rời khỏi nhóm, server sẽ xóa nhóm chat khỏi dữ liệu.

2.3. Chức năng chat riêng tư

Client gửi yêu cầu kèm theo username của thành viên muốn tạo chat riêng tư. Nếu username đã được đăng ký, server sẽ tạo chat riêng tư giữa 2 client và gửi thông báo đã tạo chat riêng tư thành công cho cả 2 client. Ngược lại, nếu username chưa được đăng ký, server sẽ gửi thông báo tạo chat riêng tư thất bại cho client gửi yêu cầu.

Khi một client gửi tin nhắn riêng tư thì tin nhắn sẽ được gửi lên server trước, sau đó server gửi tin nhắn cho người còn lại.

Khi một client thoát khỏi chat riêng tư (bằng cách đóng cửa sổ chat riêng tư), server sẽ thông báo cho client còn lại và xóa chat riêng tư giữa 2 client này.

2.4. Chức năng gửi tập tin giữa hai client khi chat riêng tư

Client gửi yêu cầu gửi tập tin trong quá trình chat riêng tư. Tập tin được chia thành nhiều phần và gửi từng phần lên server. Server nhận từng phần này và gửi cho client nhận. Sau khi nhận hết file, client nhận ghép các phần lại và ghi ra file. Server không lưu trữ file này.

2.5. Giao diện chương trình

Giao diện được xây dựng bằng các lớp hỗ trợ giao diện trên Qt. Chương trình nhận tương tác của người dùng qua giao diện client và tạo thành gói tin tương ứng để gửi cho server. Khi client nhận được một gói tin từ server, chương trình sẽ phân tách gói tin và hiện lên nội dung tương ứng trên giao diện client.

3. CẤU TRÚC GÓI TIN GỬI NHẬN

- Ý tưởng:
- Bên gửi sẽ gửi một gói tin với flag (cờ hiệu) tương ứng với yêu cầu cho bên nhận. Bên nhận dựa vào flag sẽ thực hiện yêu cầu của bên gửi.

Cấu trúc chung của các gói tin:

[FLAG][nội dung 1][nội dung 2]...[nội dung n]

- Server sẽ quản lý các client đã kết nối, đã ngắt kết nối; danh sách các nhóm chat, danh sách các phòng chat riêng tư...

3.1. Các gói tin gửi từ client lên server

| Yêu cầu từ client | FLAG | Nội dung |
|--|-----------------------|-------------------------------|
| Đăng kí tài khoản | [SIGN_UP] | [username][password] |
| Đăng nhập | [LOGIN] | [username][password] |
| Chat nhóm | | |
| Tạo nhóm chat | [CREATE_GROUP_CHAT] | [groupname] |
| Gửi tin nhắn trong nhóm chat | [GROUP_CHAT] | [groupname][sender][content] |
| Thêm thành viên vào nhóm chat | [ADD_USER] | [groupname][sender][new_user] |
| Tạo private chat với thành viên trong nhóm | [CREATE_PRIVATE_CHAT] | [sender][receiver] |
| Thoát khỏi nhóm chat | [END_GROUP_CHAT] | [groupname][sender] |
| Chat riêng tư | | |
| Tạo chat riêng tư | [CREATE_PRIVATE_CHAT] | [sender][receiver] |
| Gửi tin nhắn riêng tư | [PRIVATE_CHAT] | [sender][receiver][content] |
| Thoát khỏi chat riêng tư | [END_PRIVATE_CHAT] | [sender][receiver] |
| Gửi file | [SEND_FILE] | [receiver][filename][size] |

3.2. Các gói tin gửi từ server về client

| FLAG | Nội dung | Ý nghĩa |
|--------------------------------|---|---|
| Đăng ký & Đăng nhập | | |
| [SIGN_UP] | [SUCCESS] | Đăng kí thành công |
| | [FAIL] | Đăng kí thất bại |
| [LOGIN] | [SUCCESS] | Đăng nhập thành công |
| | [PASS_WRONG] | Nhập sai password |
| | [NOT_EXIST] | Username chưa được đăng kí |
| | [EXISTED] | Đã có người đăng nhập tài khoản |
| Chat nhóm | | |
| [CREATE_GROUP_CHAT] | [groupname][SUCCESS] | Yêu cầu tạo nhóm chat thành công |
| | [groupname][FAIL] | Yêu cầu tạo nhóm chat thất bại |
| [GROUP_CHAT] | [groupname][sender][content] | Gửi tin nhắn trong nhóm chat |
| [ADD_USER] | [groupname][new_user][SUCCESS] / [FAIL] | Gửi tin nhắn cho sender |
| [GROUP_CHAT_NEW_MEMBER] | [groupname][new_user] | Gửi cho các client trong nhóm chat tên của thành viên mới |

| | | |
|-----------------------|--|--|
| [END_GROUP_CHAT] | [groupname] | Gửi cho sender |
| [GROUP_CHAT_END_USER] | [groupname][sender] | Gửi cho các client còn lại trong nhóm chat tên thành viên rời nhóm |
| [USER_LIST] | [groupname][user1][user2]...[usern] | Gửi cho client danh sách các user trong nhóm chat |
| Chat riêng tư | | |
| [CREATE_PRIVATE_CHAT] | [sender][receiver] [SUCCESS] / [FAIL] | Yêu cầu tạo nhóm chat riêng tư thành công / thất bại |
| [PRIVATE_CHAT] | [sender][content] | Gửi tin nhắn trong chat riêng tư |
| [END_PRIVATE_CHAT] | [sender] | Gửi yêu cầu thoát khỏi chat riêng tư cho client nhận |
| [SERVER_SEND_FILE] | [sender][file name] [file size] | Thông báo bắt đầu quá trình truyền file từ server tới client. |

4. CẤU TRÚC CHƯƠNG TRÌNH

Chương trình được lập trình bằng ngôn ngữ C++ trên môi trường Qt Creator với framework hỗ trợ là Qt.

4.1. Server

Sử dụng lớp đối tượng **server** để quản lý các client, các thông tin trò chuyện và thực hiện việc nhận, xử lý, gửi gói tin đến các client.

- Các thuộc tính của server:
 - o Socket của server để kết nối với các client: **QTcpServer *Server**
 - o Danh sách các client kết nối: **vector<User> Clients**
 - o Danh sách các nhóm chat: **vector<GroupChat> g_chat**
 - o Danh sách các nhóm chat riêng tư: **vector<PrivateChat> p_chat**
 - o Một số thuộc tính khác phục vụ trong quá trình truyền nhận file: tên client gửi file, tên client nhận file, kích thước file, kích thước đã truyền/ nhận, biến kiểm tra có phải đang trong quá trình truyền nhận file không.
- Thông tin các client, nhóm chat, nhóm chat riêng tư được lưu trong các struct:
 - o Thông tin client: **User**
 - Tên đăng nhập và mật khẩu: **string username, password**
 - Tình trạng hoạt động của client: **bool online**
(true: đang kết nối/ false: ngắt kết nối)
 - Socket của client: **QTcpSocket* socket**
 - o Thông tin nhóm chat: **GroupChat**
 - Tên nhóm chat: **string group_name**
 - Danh sách các thành viên của nhóm chat: **vector<User> users**
 - o Thông tin chat riêng tư: **PrivateChat**
 - Tên 2 thành viên: **string client1, client2**

4.2. Client

Sử dụng lớp đối tượng **client** để thực hiện việc gửi nhận gói tin từ server và xử lý yêu cầu người dùng.

- Các thuộc tính chính của client:
 - o Socket của client để kết nối với server: **QTcpSocket *Client**
 - o Tên của client: **string username**
 - o Danh sách các nhóm chat client tham gia: **vector<group_chat*> groups**
 - o Danh sách các nhóm chat riêng tư mà client tham gia:
vector<private_chat*> private_rooms
 - o Một số thuộc tính khác phục vụ trong quá trình truyền nhận file: file cần ghi, nội dung file, kích thước file, kích thước đã truyền/ nhận, biến kiểm tra có phải đang trong quá trình truyền nhận file không.
- Mỗi nhóm chat và chat riêng tư là một lớp đối tượng riêng:
 - o Nhóm chat: **class group_chat**
 - Tên nhóm chat: **string name**
 - Tên client quản lý nhóm chat: **string clientName**
 - Socket của nhóm chat để gửi yêu cầu cho server: **QTcpSocket *groupSocket**
 - o Chat riêng tư: **class private_chat**
 - Tên client quản lý: **string clientName**
 - Tên client chat cùng: **string partner**
 - Socket của private chat để gửi yêu cầu cho server: **QTcpSocket *privateSocket**

5. CÁC HÀM CHỨC NĂNG

5.1. Đăng ký tài khoản

- Client:
 - Gửi gói tin: Khi người dùng nhấn nút “Sign up” client sẽ lấy username và password để đóng gói gói tin theo cấu trúc [SIGN_UP][username][password] và gửi yêu cầu đến server.
 - Nhận gói tin: Khi nhận gói tin client tách gói tin có dạng [SIGN_UP][content]. Nếu content là SUCCESS, có nghĩa là đăng kí thành công, thì client sẽ hiện thông báo cho người dùng. Nếu content là FAIL, có nghĩa là đăng kí thất bại, client sẽ báo cho người dùng biết username đã tồn tại và yêu cầu người dùng đăng kí lại.
- Server:

Thực hiện tách gói tin, sau đó, kiểm tra xem username đã tồn tại hay chưa. Nếu chưa thì thêm vào danh sách các client của server và lưu vào tập tin dữ liệu, sau đó gửi thông báo đến client [SIGN_UP][SUCCESS]. Ngược lại, gửi thông báo về cho client [SIGN_UP][FAIL].

5.2. Đăng nhập:

- Client:
 - Gửi gói tin: Khi người dùng nhấn nút “Log in” client sẽ lấy username và password để đóng gói gói tin theo cấu trúc [LOGIN][username][password] và gửi yêu cầu đến server.
 - Nhận gói tin: Khi nhận gói tin client tách gói tin có dạng [LOGIN][content]. Nếu content là SUCCESS, có nghĩa là đăng nhập thành công, thì client sẽ cho người dùng tạo nhóm chat / chat riêng tư. Nếu content là PASS_WRONG, có nghĩa là sai mật khẩu, client sẽ báo cho người dùng biết. Content là EXISTED, nghĩa là tài khoản này đang kết nối với server tại một địa chỉ khác, client sẽ thông báo cho người dùng biết. Nếu content là NOT_EXIST, tài khoản chưa được đăng kí, client sẽ gửi thông báo cho người dùng.
- Server:

Thực hiện tách gói tin, sau đó, kiểm tra xem username đã tồn tại hay chưa. Nếu chưa thì gửi thông báo về cho client [LOGIN][NOT_EXIST]. Nếu username đã tồn tại thì tiếp tục kiểm tra password. Nếu password đúng, thì kiểm tra trạng thái hoạt động. Nếu online = true, nghĩa là đã có người đăng nhập trước thì gửi tin cho client [LOGIN][EXISTED]. Nếu online = false thì cập nhật lại trạng thái hoạt động (online = true) và cập nhật lại socket, đồng thời gửi thông báo đến client [LOGIN][SUCCESS]. Ngược lại nếu password bị sai gửi tin về cho client [LOGIN][PASS_WRONG].

5.3. Chat nhóm

a) Tạo nhóm chat:

- Client:
 - Gửi gói tin: Khi người dùng nhấn nút tạo nhóm chat, client sẽ lấy tên nhóm chat để đóng gói gói tin theo cấu trúc [CREATE_GROUP_CHAT][groupname] và gửi yêu cầu đến server.
 - Nhận gói tin: Khi nhận gói tin client tách gói tin có dạng [CREATE_GROUP_CHAT][groupname][content]. Nếu content là SUCCESS, có nghĩa là yêu cầu tạo nhóm chat thành công, thì client sẽ tạo đối tượng nhóm chat mới và tạo cửa sổ chat nhóm mới. Nếu content là FAIL, có nghĩa là tên nhóm chat bị trùng, client sẽ báo cho người dùng biết.
- Server:

Thực hiện tách gói tin, sau đó, kiểm tra xem nhóm chat đã tồn tại hay chưa. Nếu chưa thì thêm vào danh sách nhóm chat của server và thêm client gửi yêu cầu vào danh sách thành viên của nhóm chat, sau đó gửi thông báo đến client [CREATE_GROUP_CHAT][groupname][SUCCESS]. Ngược lại, gửi thông báo về cho client [CREATE_GROUP_CHAT][groupname][FAIL].

b) Gửi tin nhắn:

- Client:
 - Gửi gói tin: Khi người dùng nhấn nút “Send” client sẽ lấy tin nhắn và đóng gói gói tin theo cấu trúc [GROUP_CHAT][groupname][sender][content] và gửi yêu cầu đến server.
 - Nhận gói tin: Khi nhận gói tin client tách gói tin có dạng [GROUP_CHAT][groupname][sender][content]. Client sẽ tìm kiếm nhóm chat trong danh sách nhóm chat và gửi yêu cầu cho nhóm chat in tin nhắn ra màn hình của nhóm chat đó với dạng “sender: content”, nếu là người gửi tin nhắn thì in tin nhắn với cấu trúc “You: content”.
- Server:

Thực hiện tách gói tin, sau đó tìm kiếm nhóm chat trong danh sách nhóm chat của server. Và thực hiện gửi gói tin đến tất cả các thành viên trong nhóm chat [GROUP_CHAT][groupname][sender][content].

c) Thêm thành viên:

- Client:
 - Gửi gói tin: Khi người dùng nhấn nút “Add user” client sẽ lấy username và đóng gói gói tin theo cấu trúc [ADD_USER][groupname][sender][new_user] và gửi yêu cầu đến server.
 - Nhận gói tin: Khi nhận gói tin client tách gói tin có dạng
 - * [ADD_USER][groupname][new_user][SUCCESS]/[FAIL]: client sẽ báo cho người dùng, nếu SUCCESS thì thêm thành viên mới vào danh sách các thành viên của nhóm chat.
 - * [GROUP_CHAT_NEW_USER][groupname][new_user]: hiện thông báo ra màn hình và thêm tên của thành viên mới vào danh sách các thành viên của nhóm chat.
 - * [CREATE_GROUP_CHAT][groupname][SUCCESS]: client tạo cửa sổ nhóm chat mới (server gửi tin cho thành viên mới).
 - * [USER_LIST][groupname][user1][user2]...[usern]: client in danh sách các thành viên nhóm chat ra màn hình.
- Server:

Thực hiện tách gói tin, sau đó, kiểm tra xem thành viên mới có online hay không và đã tồn tại hay chưa. Nếu chưa thì gửi thông báo đến client [ADD_USER][groupname][new_user][FAIL]. Ngược lại:

 - * Gửi thông báo về cho client gửi tin:
[ADD_USER][groupname][new_user][SUCCESS]
 - * Gửi tin cho thành viên mới: [CREATE_GROUP_CHAT][groupname][SUCCESS] và gửi danh sách các thành viên hiện có trong nhóm chat:
[USER_LIST][groupname][user1]...[usern]

* Gửi tin cho các thành viên còn lại của nhóm chat:

[GROUP_CHAT_NEW_MEMBER][groupname][new_user]

d) Chọn chat riêng tư:

- Client:
 - Gửi gói tin: Khi người dùng chọn thành viên muốn chat riêng và nhấn nút “Private chat” client sẽ đóng gói tin theo cấu trúc [CREATE_PRIVATE_CHAT][sender][receiver] và gửi yêu cầu đến server.
 - Nhận gói tin: Khi nhận gói tin client tách gói tin có dạng [CREATE_PRIVATE_CHAT][sender][receiver][content]. Nếu content là SUCCESS, có nghĩa là yêu cầu tạo chat thành công, thì client sẽ tạo đối tượng chat riêng tư mới và tạo cửa sổ chat mới. Nếu content là FAIL, có nghĩa là người dùng chưa tồn tại hoặc không online, client sẽ báo cho người dùng biết.
- Server:

Thực hiện tách gói tin, sau đó, kiểm tra xem người dùng đã tồn tại chưa hay có online hay không.

* Nếu không thỏa thì gửi thông báo về cho client gửi tin [CREATE_PRIVATE_CHAT][sender][receiver][FAIL]. Ngược lại, thêm chat riêng tư vào danh sách private chat của server và gửi thông báo về cho client [CREATE_PRIVATE_CHAT][sender][receiver]SUCCESS].

* Nếu thỏa các yêu cầu thì gửi tin cho client nhận:
[CREATE_PRIVATE_CHAT][receiver][sender][SUCCESS]

e) Thoát khỏi nhóm:

- Client:
 - Gửi gói tin: Khi người dùng tắt cửa sổ nhóm, client sẽ đóng gói tin theo cấu trúc [END_GROUP_CHAT][groupname][sender] và gửi yêu cầu đến server.
 - Nhận gói tin: Khi nhận gói tin client tách gói tin có dạng
 - * [END_GROUP_CHAT][groupname]: xóa tên nhóm chat ra khỏi danh sách nhóm chat của client và thông báo người dùng đã thoát khỏi nhóm chat.
 - * [GROUP_CHAT_END_USER][groupname][sender]: hiện thông báo ra màn hình.
 - * [USER_LIST][groupname][user1][user2]...[usern]: client in danh sách các thành viên nhóm chat ra màn hình.
- Server:

Thực hiện tách gói tin và gửi gói tin:

* Gửi thông báo về cho client gửi tin: [END_GROUP_CHAT][groupname]

* Gửi tin cho các thành viên còn lại của nhóm chat:
[GROUP_CHAT_END_USER][groupname][sender] và gửi danh sách các thành viên hiện có trong nhóm chat [USER_LIST][groupname][user1][user2]...[usern]

* Xóa username của sender khỏi danh sách thành viên nhóm chat, nếu nhóm chat không còn thành viên nào thì thực hiện xóa tên nhóm chat ra khỏi danh sách nhóm chat của server.

5.4. Chat riêng tư:

a) Tạo chat riêng tư:

- Client:
 - Gửi gói tin: Khi người dùng nhấn nút tạo chat riêng tư, client sẽ lấy tên user muốn chat để đóng gói gói tin theo cấu trúc [CREATE_PRIVATE_CHAT][sender][receiver] và gửi yêu cầu đến server.
 - Nhận gói tin: Khi nhận gói tin client tách gói tin có dạng [CREATE_PRIVATE_CHAT][sender][receiver][content]. Nếu content là SUCCESS, có nghĩa là yêu cầu tạo chat thành công, thì client sẽ tạo đối tượng chat riêng tư mới và tạo cửa sổ chat mới. Nếu content là FAIL, có nghĩa là người dùng chưa tồn tại hoặc không online, client sẽ báo cho người dùng biết.
- Server:

Thực hiện tách gói tin, sau đó, kiểm tra xem người dùng đã tồn tại chưa hay có online không.

* Nếu không thỏa thì gửi thông báo về cho client gửi tin [CREATE_PRIVATE_CHAT][sender][receiver][FAIL]. Ngược lại, thêm chat riêng tư vào danh sách private chat của server và gửi thông báo về cho client [CREATE_PRIVATE_CHAT][sender][receiver]SUCCESS].

* Nếu thỏa các yêu cầu thì gửi tin cho client nhận:
[CREATE_PRIVATE_CHAT][receiver][sender][SUCCESS]

b) Gửi tin nhắn:

- Client:
 - Gửi gói tin: Khi người dùng nhấn nút “Send” client sẽ lấy tin nhắn và đóng gói gói tin theo cấu trúc [PRIVATE_CHAT][sender][receiver][content] và gửi yêu cầu đến server và in tin nhắn ra màn hình với dạng “You: content”
 - Nhận gói tin: Khi nhận gói tin client tách gói tin có dạng [PRIVATE_CHAT][sender][content]. Client sẽ tìm kiếm private chat trong danh sách nhóm chat riêng tư và gửi yêu cầu in tin nhắn ra màn hình với dạng “sender: content”.
- Server:

Thực hiện tách gói tin, sau đó tìm kiếm private chat trong danh sách nhóm chat riêng tư của server. Và thực hiện gửi gói tin đến thành viên còn lại [PRIVATE_CHAT][sender][content].

c) Thoát khỏi chat riêng tư:

- Client:
 - Gửi gói tin: Khi người dùng tắt cửa sổ chat riêng tư, client sẽ đóng gói gói tin theo cấu trúc [END_PRIVATE_CHAT][sender][receiver] và gửi yêu cầu đến server.
 - Nhận gói tin: Khi nhận gói tin client tách gói tin có dạng [END_PRIVATE_CHAT][sender]: xóa tên chat private ra khỏi danh sách private chat của client và đóng cửa sổ chat riêng tư.
- Server:

Thực hiện tách gói tin và gửi gói tin:

 - * Gửi tin cho thành viên còn lại của chat private [END_PRIVATE_CHAT][sender].
 - * Xóa chat private ra khỏi danh sách nhóm chat riêng tư của server.

5.5. Gửi tập tin:

- Client gửi:
 - Gửi gói tin: Khi người dùng ấn nút attach, client sẽ gửi gói tin cho server để bắt đầu quá trình gửi file: [SEND_FILE][receiver][file name][file size]
 - Sau khi gửi gói tin [SEND_FILE] lên server, client gửi đọc tập tin dưới dạng nhị phân và gửi từng phần nội dung lên server cho tới khi gửi hết gói tin (số byte đã gửi = kích thước tập tin)
- Server: Thực hiện tách gói tin và xử lý:
 - * Khi nhận được gói tin [SEND_FILE] từ client, server sẽ tách gói tin để lấy thông tin client nhận, sau đó server gửi gói tin [SERVER_SEND_FILE][sender][file name][file size] cho client nhận.
 - * Server bắt đầu nhận từng phần của tập tin do client gửi gửi lên và gửi từng phần này cho client nhận. Khi đã nhận và gửi hết tập tin (số byte đã gửi đi cho client nhận = kích thước tập tin) thì kết thúc quá trình truyền nhận tập tin.
- Client nhận:
 - * Khi nhận được gói tin [SERVER_SEND_FILE], client nhận tách gói tin để lấy thông tin người gửi, tên tập tin và kích thước. Sau đó, client nhận mở tập tin để chuẩn bị nhận nội dung.
 - * Client nhận nhận từng phần nội dung file do server gửi. Khi đã nhận hết tập tin (số byte đã nhận = kích thước tập tin) thì kết thúc quá trình nhận, client nhận ghi nội dung ra tập tin và đóng tập tin.

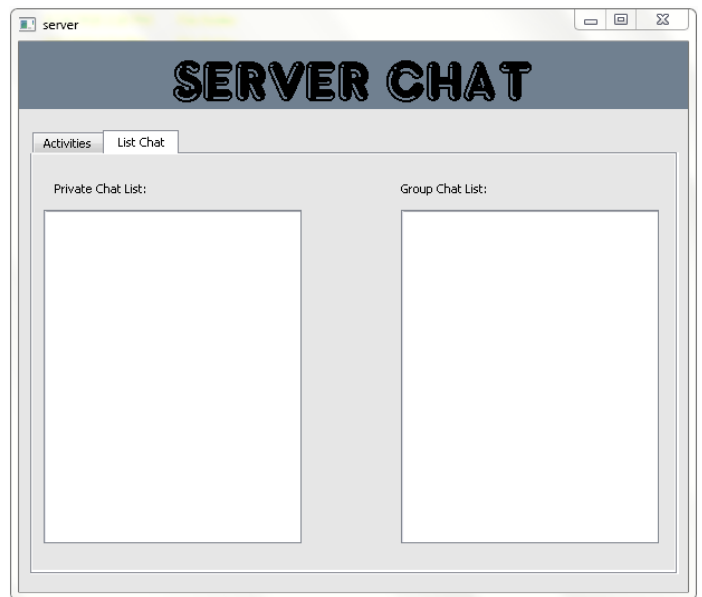
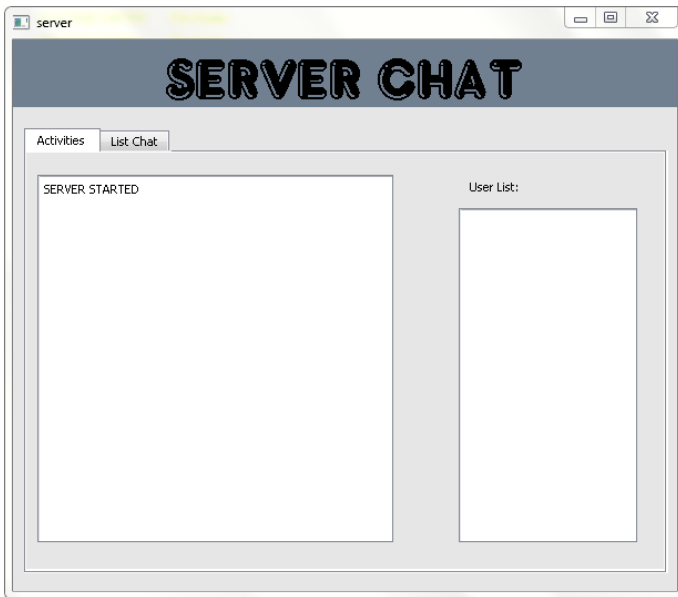
Nội dung file không được đóng gói thành gói tin hoàn chỉnh có cờ hiệu (flag). Server và client nhận biết message là nội dung gói tin khi biến fileTransferring dùng để nhận biết quá trình truyền nhận file có giá trị true. Khi kết thúc quá trình truyền nhận file, biến fileTransferring được gán lại giá trị false, các message sau đó được xử lý bình thường dựa vào flag.

5.6. Ngắt kết nối:

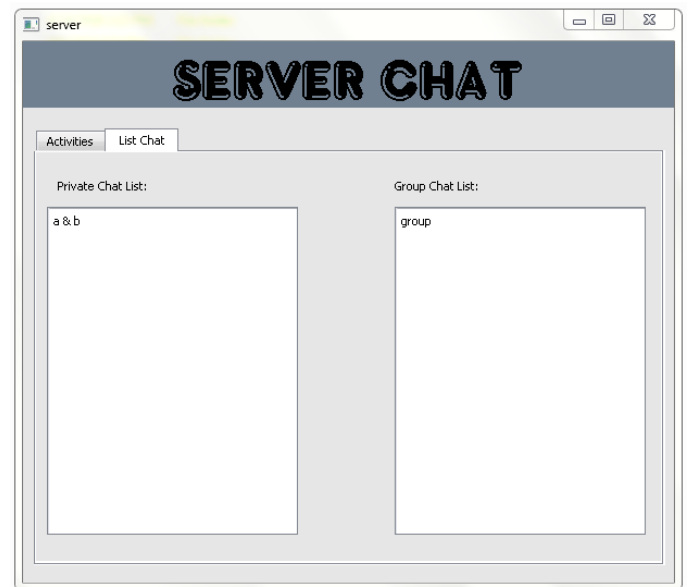
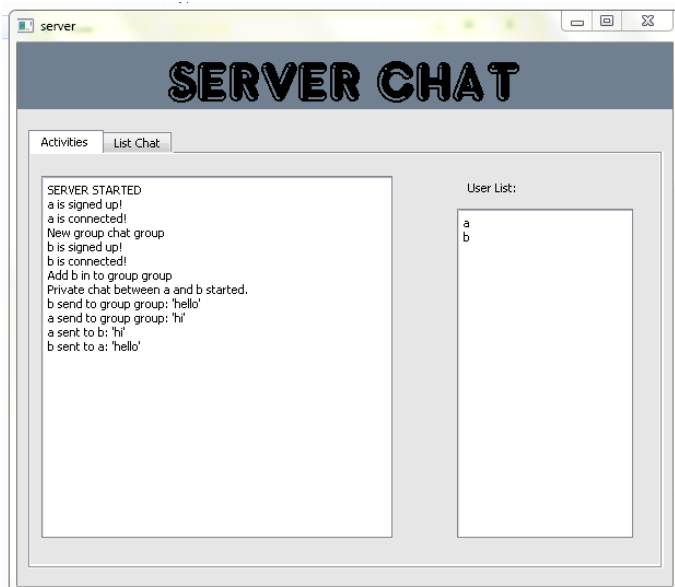
- Khi client ngắt kết nối:
 - Client đóng hết các cửa sổ chat
 - Server xóa tên client ngắt kết nối ra khỏi danh sách các nhóm chat và chat riêng tư (tuân thủ cấu trúc gói tin phía trên) và thông báo client ngắt kết nối ra màn hình
- Khi server ngắt kết nối:
Client thông báo cho người dùng là không thể kết nối với server.

6. GIAO DIỆN CHƯƠNG TRÌNH

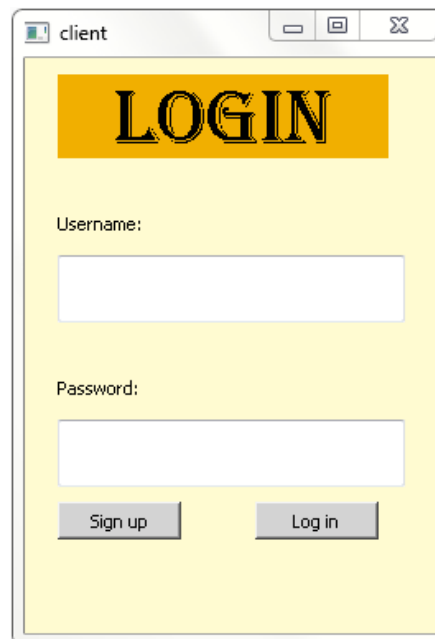
6.1. Giao diện của server:



- Server quản lý hoạt động client:

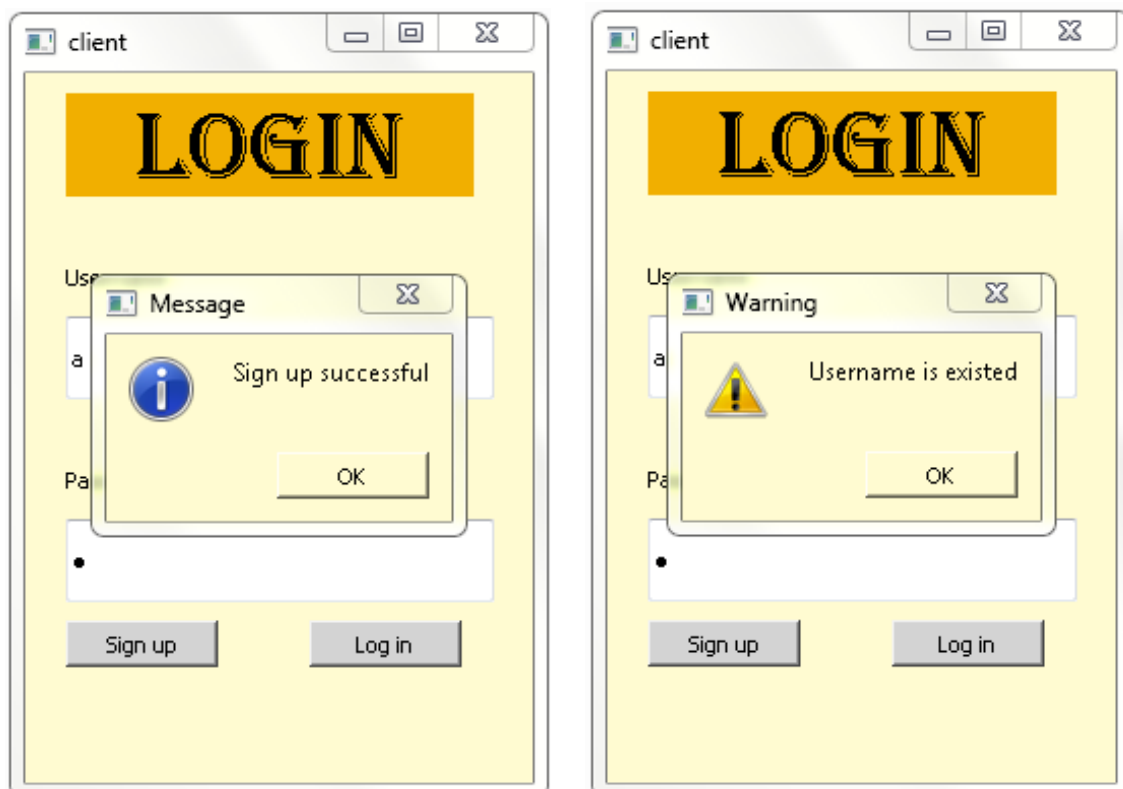


6.2. Giao diện đăng kí, đăng nhập:



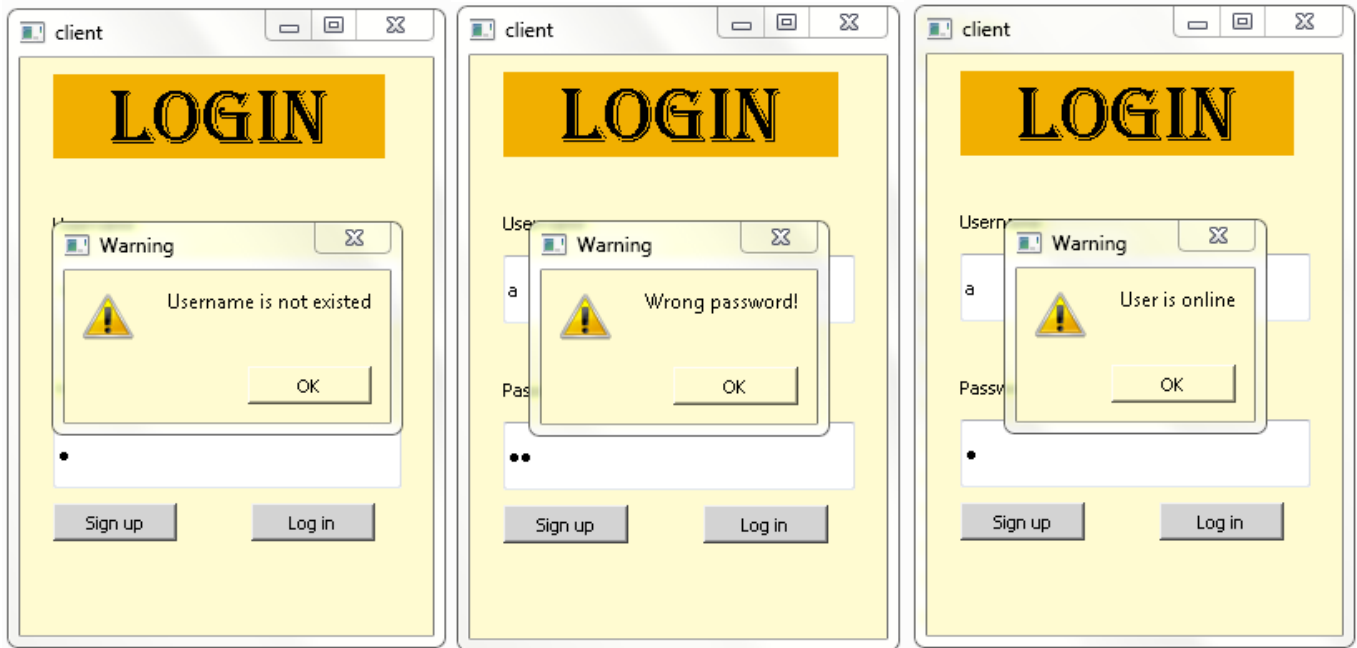
The screenshot shows a window titled 'client' with a yellow background. At the top, there is a large orange rectangle with the word 'LOGIN' in black, serif, all-caps font. Below this, the text 'Username:' is followed by a white text input field. Further down, the text 'Password:' is followed by another white text input field. At the bottom of the window, there are two gray buttons: 'Sign up' on the left and 'Log in' on the right.

- Đăng kí:

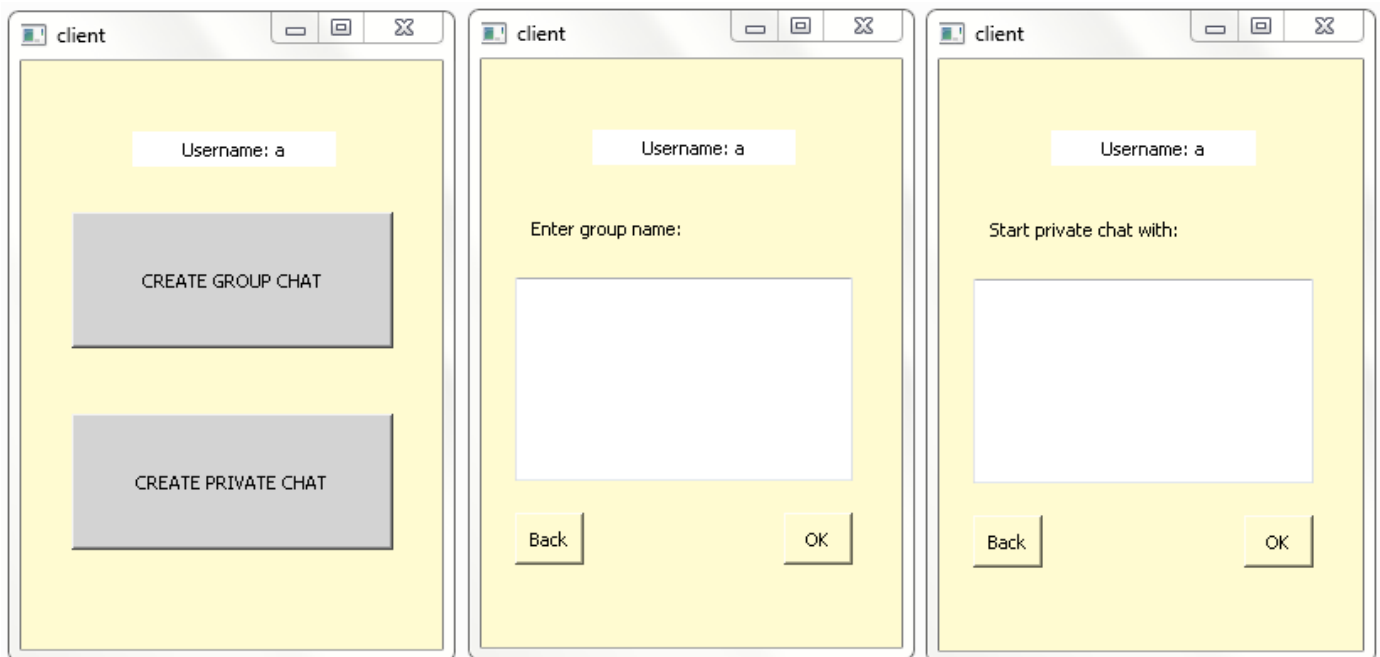


The two screenshots show the same 'client' window as above, but with different messages displayed in a small dialog box over the input fields. The left screenshot shows a 'Message' dialog box with a blue information icon and the text 'Sign up successful'. The right screenshot shows a 'Warning' dialog box with a yellow warning icon and the text 'Username is existed'. Both dialog boxes have an 'OK' button at the bottom.

- Đăng nhập:



- Tạo chat:



6.3. Giao diện chat nhóm:

The screenshot shows a window titled "Form" with a teal header bar labeled "GROUP CHAT". Below the header, there are two input fields: "Group name" with the value "group" and "Username" with the value "a". To the right of the "Username" field is an "ADD" button. Below these fields is a large white rectangular area for chat messages. To the right of this area is a vertical list of chat participants, currently showing "a (You)". Below the participant list is a "Private Chat" button. At the bottom of the window is a text input field and a "SEND" button.

- Thêm user vào nhóm chat:

This screenshot shows the same "GROUP CHAT" interface as before, but with a "Warning" dialog box open in the center. The dialog box has a yellow warning icon and the text "Add User Fail". There is an "OK" button at the bottom of the dialog. The background interface elements remain visible but are slightly dimmed.

This screenshot shows the same "GROUP CHAT" interface, but with a "Message" dialog box open in the center. The dialog box has a blue information icon and the text "Add User Successful". There is an "OK" button at the bottom of the dialog. The background interface elements remain visible but are slightly dimmed.

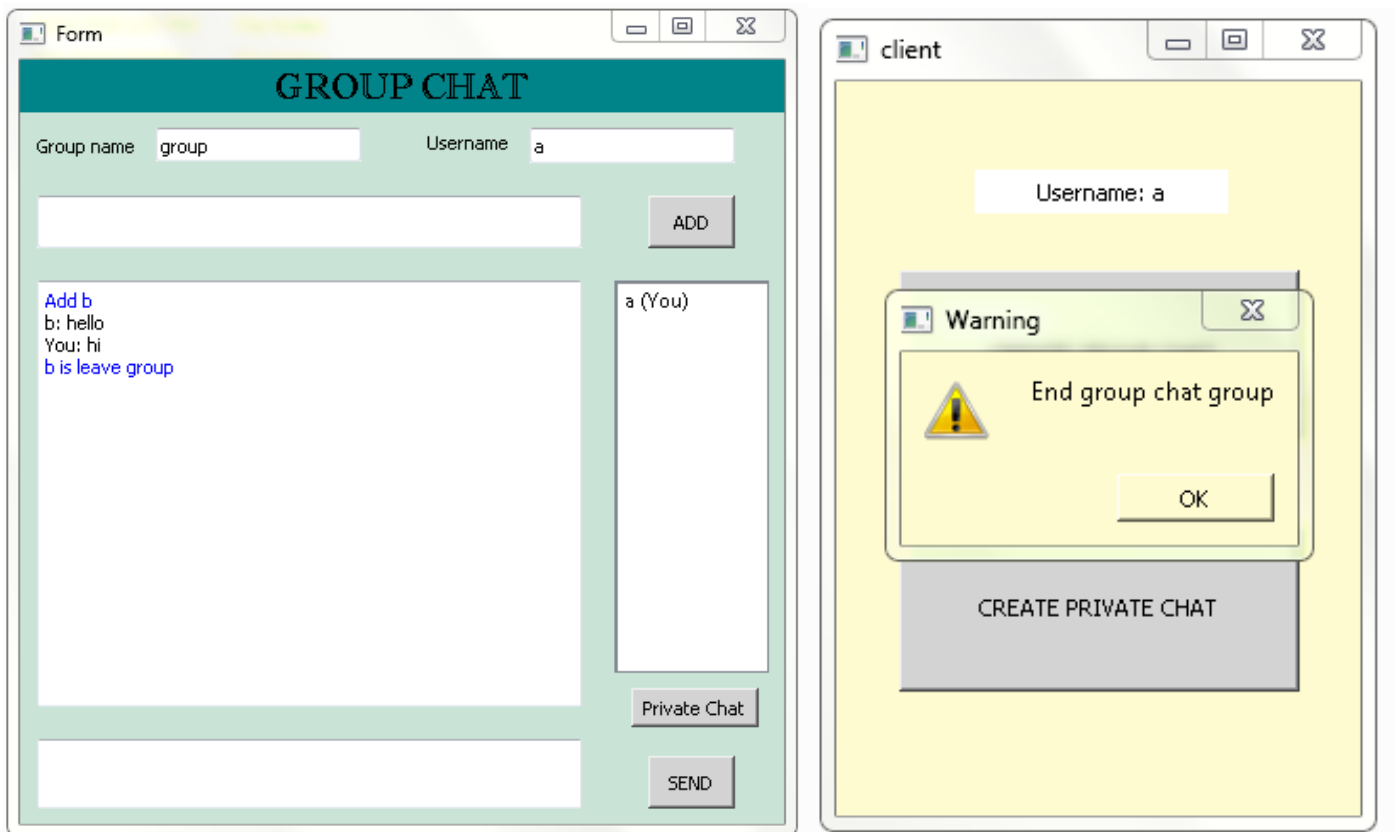
- Nhắn tin trong nhóm:

The screenshot shows a web application window titled "Form" with a teal header "GROUP CHAT". Below the header, there are two input fields: "Group name" with the value "group" and "Username" with the value "a". To the right of these fields is an "ADD" button. Below the input fields is a large text area containing the text "Add b", "b: hello", and "You: hi". To the right of this text area is a list of users: "a (You)" and "b". Below the list is a "Private Chat" button. At the bottom of the window is a "SEND" button.

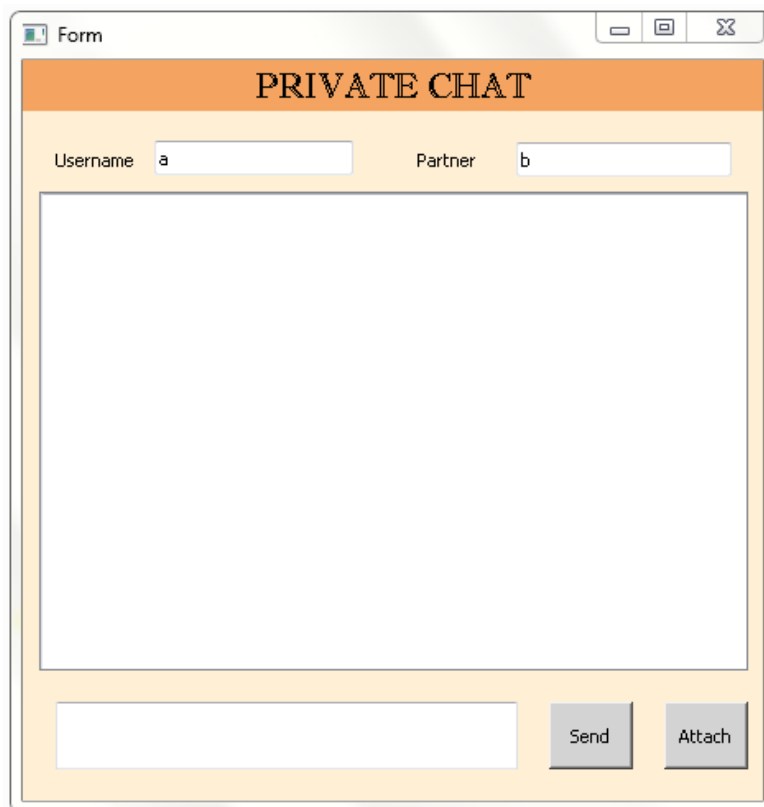
- Tạo chat riêng tư:

The screenshot shows the same web application window as before, but with a "Private Chat" dialog box open. The dialog box has a title bar "Private Chat" and a question mark icon. The text inside the dialog box says "Create Private Chat with b?". There are two buttons: "Yes" and "No". The "b" user in the list on the right is highlighted.


- Thoát khỏi group chat:



6.4. Giao diện chat riêng tư:

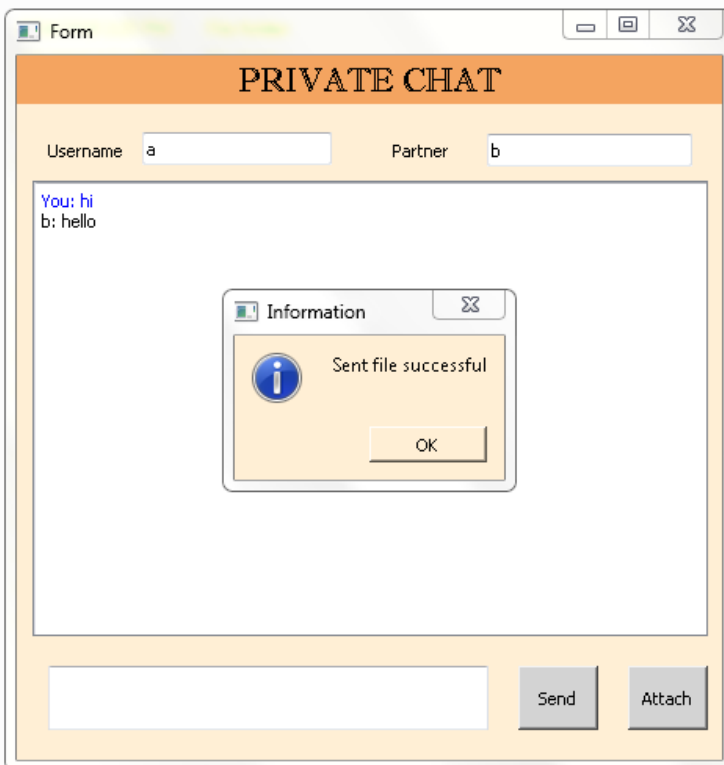


- **Nhắn tin:**

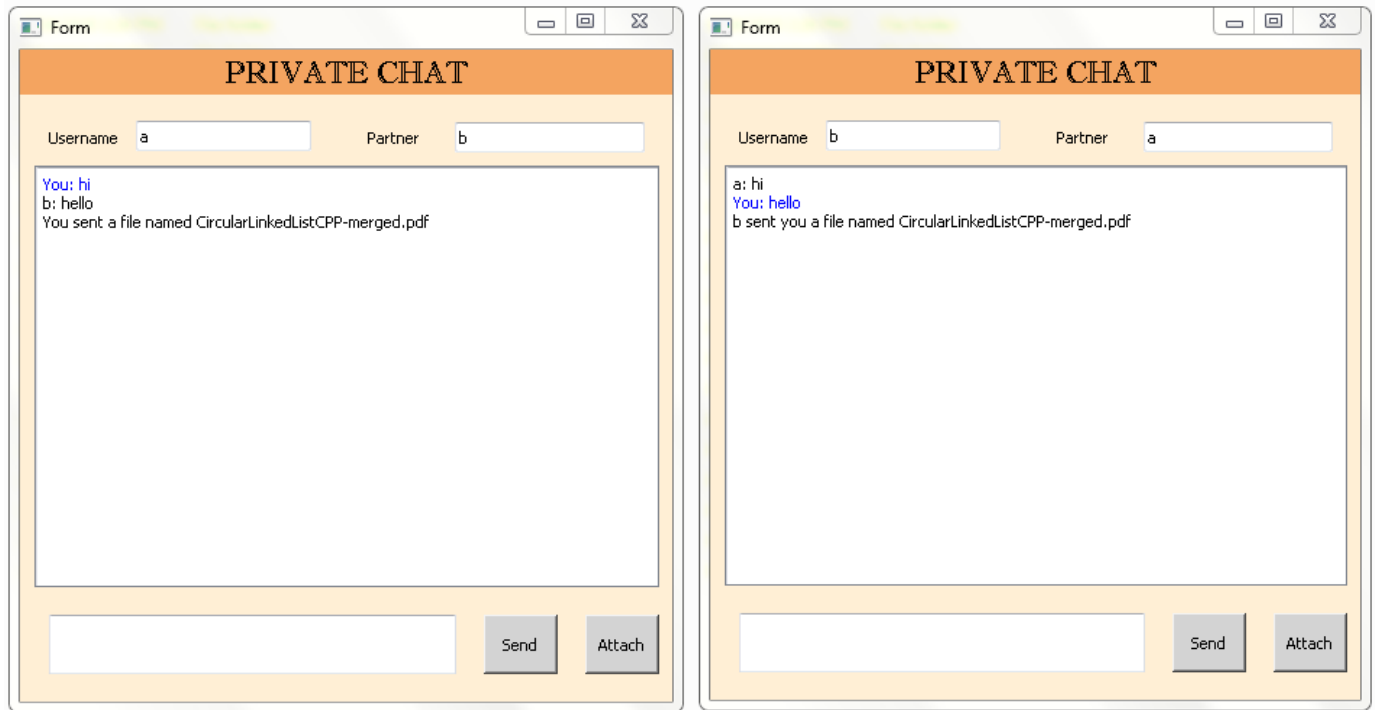


The screenshot shows a window titled "Form" with a sub-header "PRIVATE CHAT". Below the header, there are two input fields: "Username" with the value "a" and "Partner" with the value "b". The main area of the window is a large text box containing the message history: "You: hi" and "b: hello". At the bottom of the window, there is a text input field for sending new messages, and two buttons labeled "Send" and "Attach".

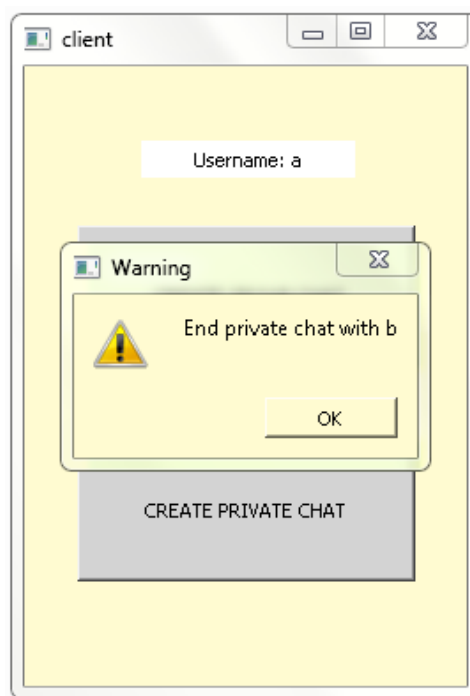
- **Gửi file:**



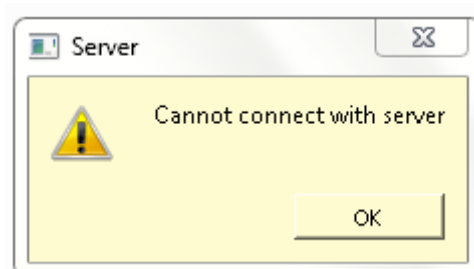
The screenshot shows the same "PRIVATE CHAT" window as before, but with an additional "Information" dialog box displayed in the center. The dialog box has a title bar "Information" and a close button. It contains a blue information icon and the text "Sent file successful". Below the text is an "OK" button. The chat window's message history and input fields remain visible in the background.

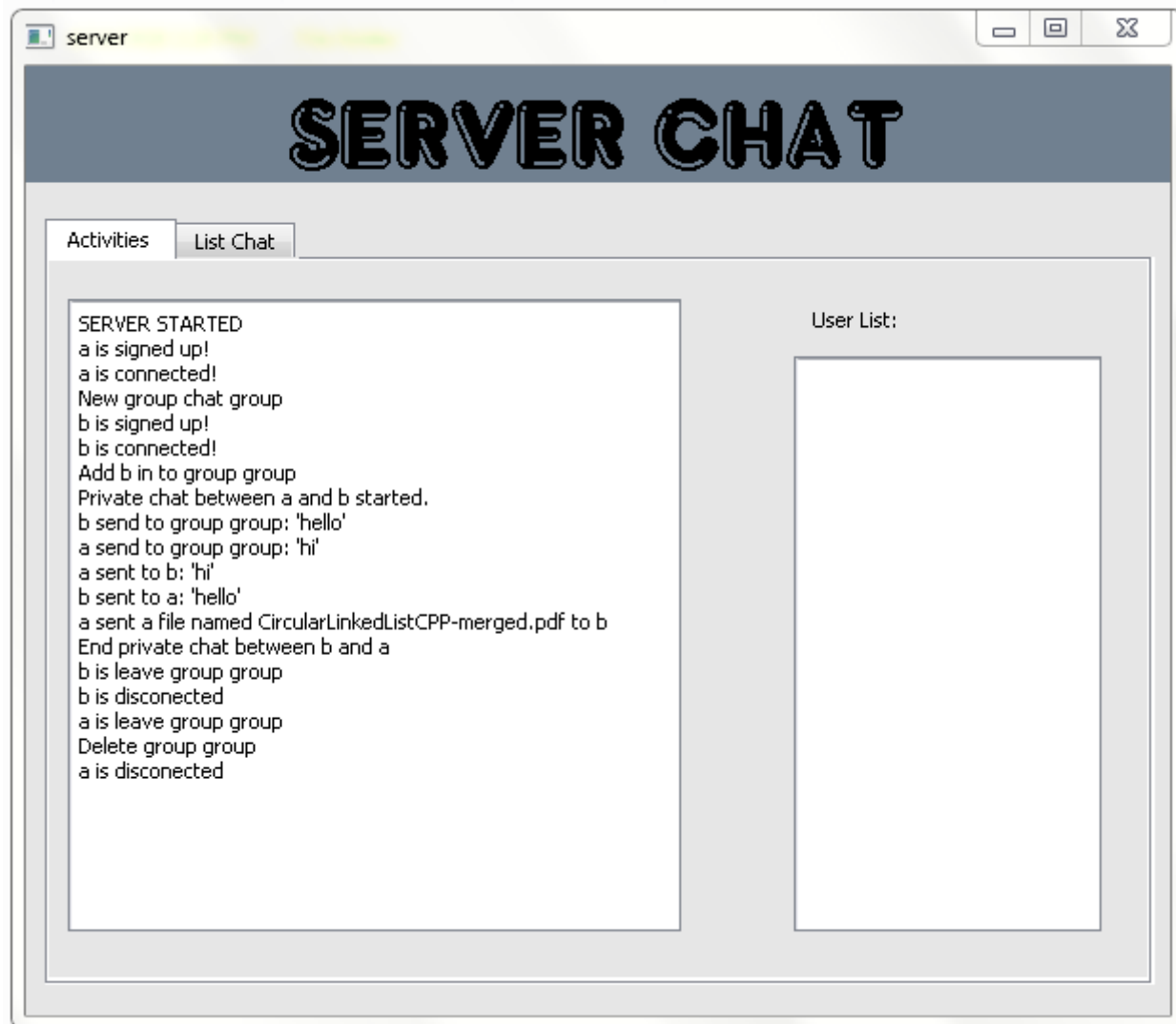


- Thoát chat riêng tư:



6.5. Khi server/ client ngắt kết nối:





7. TỰ ĐÁNH GIÁ

7.1. Mức độ hoàn thành đồ án

| Chức năng | Mức độ hoàn thành | Chưa hoàn thành |
|----------------------------|-------------------|---|
| Đăng ký và đăng nhập | 100% | |
| Chat nhóm | 100% | |
| Chat riêng tư | 100% | |
| Gửi file khi chat riêng tư | 40% | <ul style="list-style-type: none"> - Chỉ gửi được file có kích thước nhỏ. Một số file khi gửi bị mất dữ liệu nên không gửi tới đích được. - Chưa có thông báo khi gửi file thất bại. Chưa cho người nhận chọn chỗ lưu file. |
| Giao diện chương trình | 100% | |
| Tổng cộng: | 85% | |

7.2. Đóng góp của các thành viên

| Thành viên | Khối lượng công việc | Công việc |
|--|----------------------|---|
| 1712152 Nguyễn Thị Mai Thanh | 35% | <ul style="list-style-type: none">- Load thông tin client từ file, lưu thông tin client xuống file- Thiết kế cấu trúc các gói tin- Xử lý khi client/ server ngắt kết nối- Cài đặt chức năng chat nhóm: tạo chat riêng tư khi đang chat nhóm, xóa nhóm chat, in danh sách các thành viên trong nhóm chat- Thiết kế giao diện |
| 1712228 Phạm Việt Nga | 35% | <ul style="list-style-type: none">- Khởi tạo các socket, thực hiện kết nối client với server- Cài đặt hàm lắng nghe, gửi nhận tin giữa server, client và phân tách các gói tin- Cài đặt chức năng chat riêng tư: tạo, gửi tin nhắn và kết thúc chat- Cài đặt chức năng truyền nhận tập tin |
| 1712807 Nguyễn Thị Minh Thùy | 30% | <ul style="list-style-type: none">- Mã hóa và giải mã mật khẩu khi client đăng ký/ đăng nhập, kiểm tra các điều kiện đăng ký/ đăng nhập- Kiểm tra trạng thái của client và cập nhật danh sách client- Cài đặt chức năng chat nhóm: tạo nhóm chat, thêm thành viên, gửi tin nhắn |

8. TÀI LIỆU THAM KHẢO

- Các bài viết về Qt trên trang web <https://doc.qt.io/> và diễn đàn <https://forum.qt.io/>
- Demo Chat_1Server_1Client, DemSoNT_1Server_NhieuClient
- Bài viết về cách gửi nhận file: <https://stackoverflow.com/questions/30288385/how-to-send-a-file-in-qt>