

**ĐẠI HỌC BÁCH KHOA HÀ NỘI**  
**TRƯỜNG CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG**

---



**BÁO CÁO BÀI TẬP LỚN**  
**THỰC HÀNH LẬP TRÌNH MẠNG (IT4062)**  
**Đề tài: Ứng dụng chat**

Nhóm:	<b>8</b>
Họ Và Tên:	<b>Phạm Nhật Anh 20214987</b> <b>Trịnh Việt Anh 20214990</b>
Ngày thực hiện	<b>26/12/2024</b>
Giảng viên hướng dẫn:	<b>Th.S Tống Văn Vạn</b>

***Hà Nội, 12/2024***

## MỤC LỤC

<b>LỜI NÓI ĐẦU .....</b>	<b>3</b>
<b>PHÂN CÔNG THÀNH VIÊN TRONG NHÓM .....</b>	<b>4</b>
<b>I. GIỚI THIỆU ĐỀ TÀI.....</b>	<b>5</b>
1. Đặt vấn đề.....	5
2. Mục tiêu và phạm vi của đề tài .....	5
3. Kiến trúc ứng dụng .....	6
<b>II. TỔNG QUAN CHỨC NĂNG.....</b>	<b>7</b>
1. Biểu đồ usecase: .....	7
2. Luồng hoạt động của ứng dụng.....	8
3. Cấu trúc các thông điệp .....	9
<b>III. CÔNG NGHỆ SỬ DỤNG .....</b>	<b>11</b>
1. Thư viện sys/socket.h và các thư viện lập trình mạng .....	11
2. Qt Framework .....	12
<b>IV. KẾT QUẢ VÀ ĐÁNH GIÁ.....</b>	<b>12</b>
1. Xây dựng ứng dụng .....	12
2. Đánh giá.....	19

## LỜI NÓI ĐẦU

Ứng dụng chat đơn giản với socket là một giải pháp giao tiếp hiệu quả, được thiết kế để minh họa cách các hệ thống mạng tương tác thông qua giao thức TCP/UDP. Đây không chỉ là một ứng dụng phục vụ việc nhắn tin mà còn là cơ hội để người dùng khám phá và hiểu rõ hơn về cách giao tiếp giữa client và server trong môi trường mạng.

Với các tính năng cơ bản như nhắn tin 1-1, chat nhóm, và quản lý kết nối người dùng theo thời gian thực, ứng dụng mang lại trải nghiệm giao tiếp đơn giản, dễ sử dụng và trực quan. Ứng dụng sử dụng kỹ thuật socket để đảm bảo dữ liệu được truyền tải nhanh chóng, đồng thời hỗ trợ học tập và thực hành kỹ thuật lập trình mạng cho người dùng.

Tuy đã chuẩn bị và phát triển khá hoàn thiện nhưng không tránh khỏi những thiếu sót. Nhóm rất mong thầy và các bạn có thể góp ý để cho sản phẩm của chúng em được hoàn thiện hơn.

Xin chân thành cảm ơn! .

Phạm Nhật Anh  
Trịnh Việt Anh

## PHÂN CÔNG THÀNH VIÊN TRONG NHÓM

Họ và tên	MSSV	Tổng hợp công việc thực hiện
Trịnh Việt Anh	20214990	- Thiết kế ChatGroup, - Thêm bạn bè
Phạm Nhật Anh	20214987	- Thiết kế giao diện người dùng - Thiết kế chat 1-1

# I. GIỚI THIỆU ĐỀ TÀI

## 1. Đặt vấn đề

Trong thời đại công nghệ số phát triển mạnh mẽ, nhu cầu giao tiếp nhanh chóng và hiệu quả qua các ứng dụng nhắn tin ngày càng trở nên quan trọng. Các ứng dụng chat không chỉ phục vụ mục đích trao đổi thông tin mà còn đóng vai trò quan trọng trong việc kết nối cá nhân, nhóm làm việc và cộng đồng.

Tuy nhiên, việc xây dựng một ứng dụng chat hiệu quả đòi hỏi hiểu biết sâu sắc về giao thức mạng, cách thức trao đổi dữ liệu và quản lý kết nối. Đặc biệt, việc sử dụng socket trong lập trình mạng là một công cụ mạnh mẽ, giúp đảm bảo giao tiếp thời gian thực và quản lý dữ liệu một cách hiệu quả.

Do đó, việc phát triển một ứng dụng chat đơn giản sử dụng socket không chỉ mang lại giá trị thực tế trong việc liên lạc mà còn là một bài học quan trọng về lập trình mạng. Ứng dụng sẽ giúp người dùng, đặc biệt là các lập trình viên mới, hiểu rõ hơn về cách thức giao tiếp giữa client và server, quản lý kết nối và xử lý dữ liệu mạng.

## 2. Mục tiêu và phạm vi của đề tài

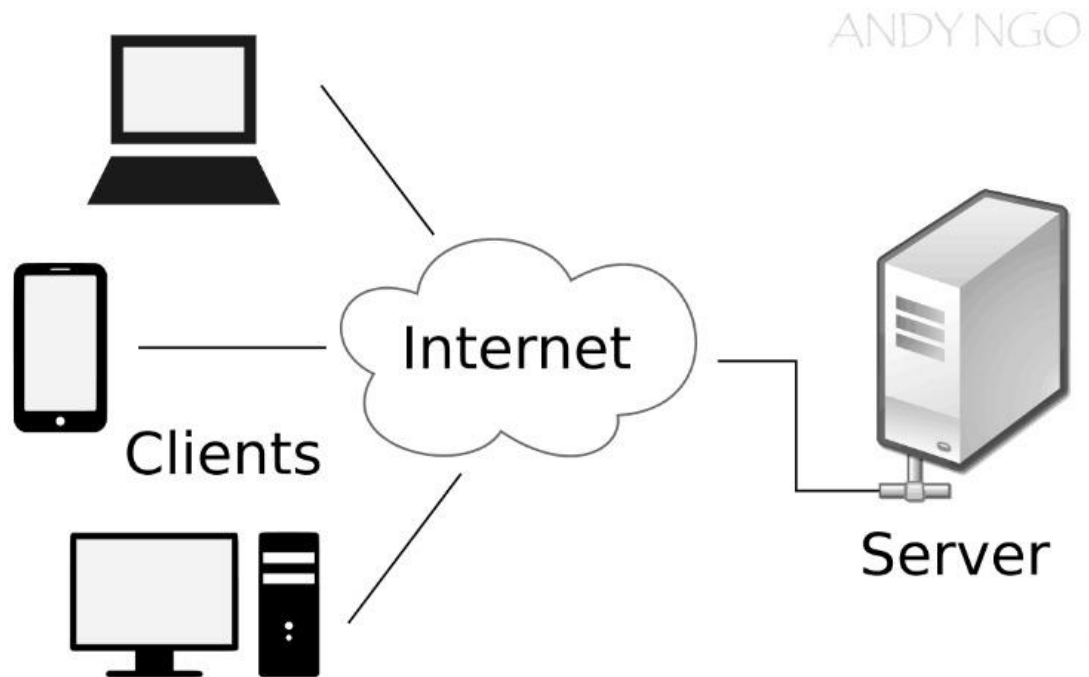
Trong đề tài này, nhóm mong muốn áp dụng các kiến thức đã học về lập trình mạng, đặc biệt là các khía cạnh như giao thức mạng, xử lý đa người dùng, và tương tác người dùng, để xây dựng một ứng dụng chat đơn giản với socket. Đây không chỉ là cơ hội để nhóm hiểu rõ hơn về giao tiếp mạng giữa client-server mà còn là nền tảng để tích lũy kinh nghiệm phát triển phần mềm, từ đó tạo ra các dự án hữu ích hơn trong tương lai.

Đề tài này sẽ chú trọng vào phát triển một ứng dụng desktop thỏa mãn được một số chức năng sau:

- **Quản lý tài khoản:** Cung cấp tính năng đăng ký và đăng nhập, đảm bảo tính bảo mật và chính xác của thông tin người dùng.
- **Quản lý trạng thái người dùng:** Hiển thị danh sách các người dùng đang online, hỗ trợ việc kết nối và bắt đầu trò chuyện.
- **Gửi/nhận tin nhắn:** Cho phép người dùng nhắn tin trực tiếp (1-1) hoặc tham gia vào các nhóm chat. Tin nhắn được truyền qua giao thức socket để đảm bảo thời gian thực.
- **Quản lý nhóm chat:** Hỗ trợ tạo nhóm, thêm thành viên, rời nhóm, và gửi tin nhắn trong nhóm.
- **Quản lý kết nối:** Thông báo trạng thái kết nối của các người dùng (online/offline) và đảm bảo kết nối ổn định giữa client và server.

### 3. Kiến trúc ứng dụng

#### Mô hình Client-Server



Kiến trúc client-server là một mô hình cơ bản trong mạng máy tính, trong đó một máy chủ trung tâm (server) cung cấp dịch vụ và tài nguyên cho nhiều thiết bị client. Các client yêu cầu thông tin hoặc dịch vụ từ máy chủ, máy chủ xử lý yêu cầu và gửi phản hồi phù hợp.

Trong “Ứng dụng chat” sử dụng mô hình Client-Server, các nguyên tắc hoạt động cơ bản vẫn tuân theo quy trình chung của hệ thống Client-Server nhưng được điều chỉnh để phù hợp với yêu cầu gửi nhận tin nhắn và quản lý kết nối người dùng

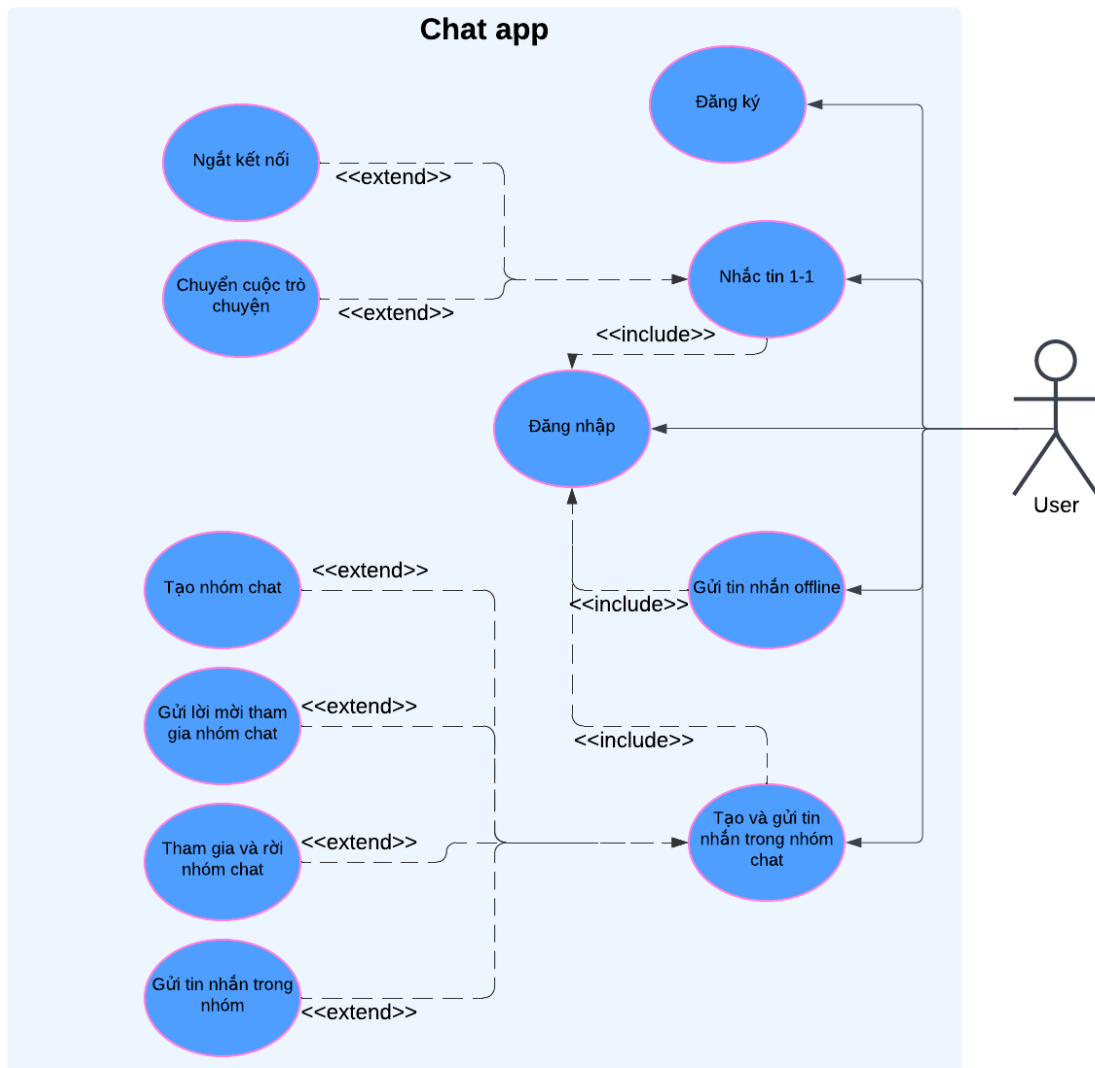
#### Nguyên tắc hoạt động

- **Client:** Mỗi người dùng sẽ chạy một ứng dụng client, có thể là giao diện đồ họa (GUI) hoặc ứng dụng dòng lệnh (CLI), cho phép họ đăng nhập, gửi và nhận tin nhắn.

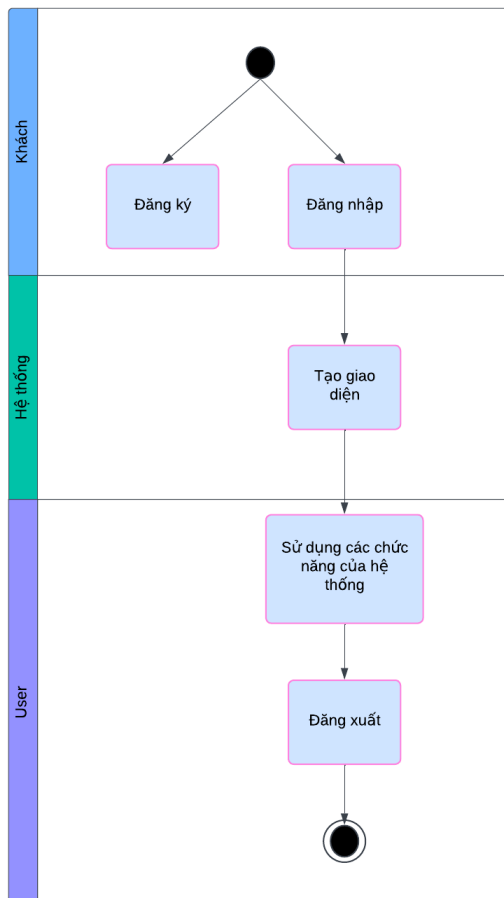
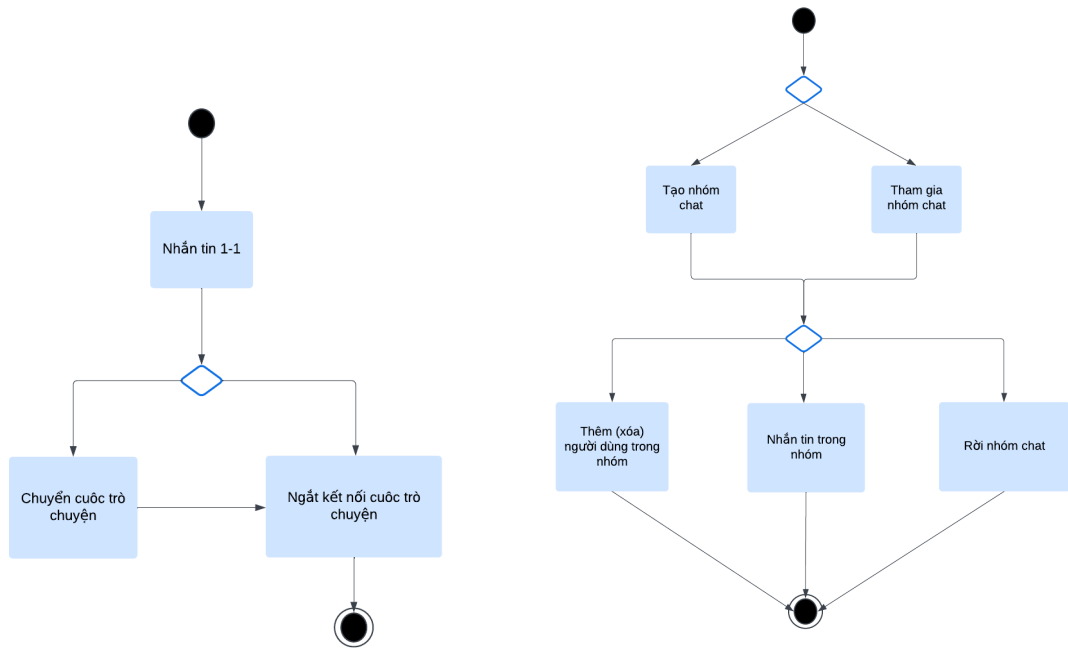
- **Server:** Server là trung tâm xử lý mọi yêu cầu từ client, chịu trách nhiệm nhận tin nhắn, chuyển tiếp tin nhắn tới người dùng đích và quản lý danh sách người dùng online.

## II. TỔNG QUAN CHỨC NĂNG

### 1. Biểu đồ usecase:



## 2. Luồng hoạt động của ứng dụng





### 3. Cấu trúc các thông điệp

#### CLIENT:

##### Client :

<b>- Đăng ký: {</b> "action": "signin", "username": "string", "password": "string" <b>}</b>	<b>- Gửi và nhận tin 1-1 : {</b> "action": "send_private_message", "sender_name": "string", "receiver_name": "string", "message": "string", <b>}</b>	<b>- Phản hồi nhắn tin 1-1 : {</b> "action": "accept_private_message" /"decline_private_message", "sender_name": "string", "receiver_name": "string", <b>}</b>
<b>- Đăng nhập: {</b> "action": "login", "username": "string", "password": "string" <b>}</b>	<b>- Yêu cầu nhắn tin 1-1 : {</b> "action": "request_private_message", "sender_name": "string", "receiver_name": "string", "message": "string", <b>}</b>	<b>- Ngắt kết nối: {</b> "action": "disconnect", "user_name": "string", "target_user_name": "string" <b>}</b>

<b>- Tạo nhóm chat: {</b> "action": "create_room", "creator_name": "string", "group_name": "string" <b>}</b> <b>- Gửi lời mời tham gia nhóm: {</b> "action": "invite_to_room", "user_name": "string", "group_name": "string" <b>}</b> <b>- Phản hồi lời mời tham gia nhóm: {</b> "action": "response_invite", "user_name": "string", "status": "int", <b>}</b>	<b>- Tham gia nhóm chat: {</b> "action": "join_room", "user_name": "string", "group_name": "string" <b>}</b> <b>- Rời nhóm chat: {</b> "action": "leave_room", "user_name": "string", "group_name": "string" <b>}</b> <b>- Xem người dùng nhóm chat: {</b> "action": "view_room_user", "user_name": "string", "group_name": "string" <b>}</b>	<b>- Gửi tin nhắn đến nhóm: {</b> "action": "send_message_to_room", "sender_name": "string", "group_name": "string", "message": "string", <b>}</b> <b>- Xóa người dùng khỏi nhóm: {</b> "action": "remove_member", "user_name": "string", "target_name": "string", "group_name": "string", <b>}</b>
--	--	--

<b>- Xem danh sách user online: {</b> "action": "view_online_users", "username": "string" <b>}</b> <b>- Xem danh sách friend request: {</b> "action": "view_friend_request", "username": "string", <b>}</b> <b>- Xem danh sách friend online: {</b> "action": "view_friend_request", "username": "string", <b>}</b>	<b>- Gửi lời mời kết bạn : {</b> "action": "send_friend_request", "sender_name": "string", "receiver_name": "string", <b>}</b> <b>- Phản hồi lời mời kết bạn : {</b> "action": "reponse_friend_request", "status": "int", "sender_name": "string", "receiver_name": "string", <b>}</b>	<b>- Hủy kết bạn : {</b> "action": "unfriend", "user_name": "string", "target_name": "string", <b>}</b> <b>- Ngắt kết nối: {</b> "action": "disconnect", "user_name": "string", "target_user_name": "string" <b>}</b>
--	--	--

#### SERVER:

**Server :****- Phản hồi đăng ký: {**

```
"action": "signin",
"status": "int",
"user_name": "string",
"message": "string",
}
```

**- Phản hồi đăng nhập: {**

```
"action": "login",
"status": "int",
"user_name": "string",
"message": "string",
}
```

**- Khi có lỗi xảy ra: {**

```
"action": "int" //depend on user's action
"message": "ERROR|string"
}
```

**- Hiển thị trạng thái friend: {**

```
"action": "view_friend_list",
"message": "FRIEND_LIST|
    user_name1 status,
    user_name2 status, .... "
}
```

**- Hiển thị danh sách người dùng online: {**

```
{
"action": "view_online_users",
"message": "ONLINE_USERS|
    username1": "string",
    "username2": "string", ... "
}
```

**- Hiển thị friend online: {**

```
"action": "view_friend_online",
"message": "ONLINE_FRIEND_LIST|
    user_name1,
    user_name2, .... "
}
```

**Server :****- Phản hồi tạo, tham gia phòng chat: {**

```
"action": "create_room"/"join_room",
"room_name": "string"
}
```

**- Phản hồi cập nhật phòng chat : {**

```
"message": "ROOM_UPDATE|
    message"
}
```

**- Hiển thị danh sách thành viên phòng chat: {**

```
"action": "view_room_users",
"message": "ROOM_USERS|
    username1": "string",
    "username2": "string", ... "
}
```

**- Hiển thị danh sách phòng chat: {**

```
{
"action": "view_user_rooms",
"message": "ROOM_LIST|
    room_name1": "string",
    "room_name2": "string" ... "
}
```

**- Hiển thị lời mời phòng chat: {**

```
"action": "send_room_invite",
"message": "INVITE|
    sender_name": "string",
    "room_name": "string" "
}
```

**- Hiển thị thông báo bị xóa khỏi phòng chat: {**

```
"action": "remove_member",
"message": "ROOM_REMOVE|
    message": "string" "
}
```

**- Hiển thị lời mời chat 1 1: {**

```
"action": "chat_request",
"message": "CHAT_REQUEST|
    sender_name": "string" "
}
```

**- Hiển thị phản hồi lời mời chat 1 1: {**

```
{
"action": "chat_request",
"message":
"CHAT_REQUEST_ACCEPTED|
    sender_name": "string" "
}
```

**- Phản hồi ngắt kết nối: {**

```
"message": "DISCONNECT|string",
}
```

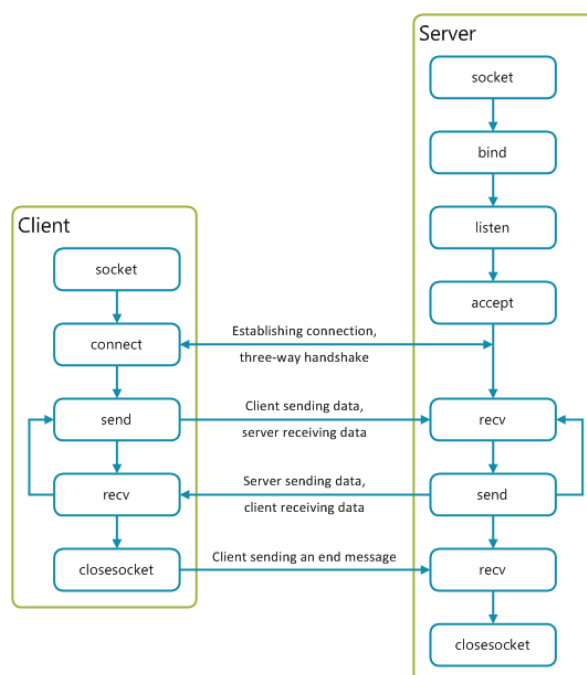
**- Hiển thị lời mời kết bạn: {**

```
"action": "view_friend_request",
"message": "FRIEND_REQUEST|
    username1": "string",
    "username2": "string", ... "
}
```

### III. CÔNG NGHỆ SỬ DỤNG

Để xây dựng ứng dụng này, nhóm sẽ tạo ra hai chương trình, một dành cho phía server và một dành cho phía client. Ở phía server, nhóm sẽ sử dụng ngôn ngữ lập trình C, cùng với thư viện *sys/socket.h* để tạo một tiến trình server chạy trên hệ thống. Ở phía client sẽ sử dụng *Qt Framework* với nền tảng là ngôn ngữ C++ để xây dựng giao diện người dùng, kết hợp với thư viện *sys/socket.h* để hỗ trợ giao tiếp với server thông qua TCP socket.

#### 1. Thư viện *sys/socket.h* và các thư viện lập trình mạng



*sys/socket.h* là một thư viện trong ngôn ngữ lập trình C được sử dụng để thực hiện các thao tác liên quan đến việc làm việc với socket, một cơ chế cơ bản trong truyền thông mạng. Thư viện này cung cấp các hàm và cấu trúc dữ liệu để tạo, quản lý, và giao tiếp qua các kết nối mạng sử dụng các giao thức như TCP hoặc UDP.

## 2. Qt Framework



Qt là một framework phát triển ứng dụng đa nền tảng mạnh mẽ và linh hoạt, được viết bằng C++ và quản lý bởi Qt Company. Nổi bật với khả năng "viết một lần, chạy mọi nơi," Qt cho phép ứng dụng hoạt động trên nhiều hệ điều hành như Windows, Linux, macOS, Android và iOS mà không cần sửa đổi mã nguồn đáng kể. Một trong những điểm sáng của Qt là mô hình "signal và slot," mang đến cách quản lý sự kiện và tương tác giữa các thành phần trong ứng dụng một cách hiệu quả và linh hoạt.

Bên cạnh đó, Qt còn cung cấp một loạt thư viện mạnh mẽ như Qt Widgets để xây dựng giao diện truyền thống, Qt Quick cho giao diện động, Qt Network hỗ trợ mạng, và Qt Multimedia để xử lý âm thanh, video. Các công cụ đi kèm, đặc biệt là Qt Creator, giúp lập trình viên tăng năng suất trong việc phát triển và gỡ lỗi ứng dụng. Qt cũng hỗ trợ đa ngôn ngữ, dễ dàng tích hợp dịch thuật cho các dự án quốc tế hóa.

Với giấy phép LGPL, Qt là sự lựa chọn lý tưởng cho cả các dự án mã nguồn mở lẫn thương mại, mang đến sự linh hoạt và tự do trong phát triển phần mềm mà không phải lo lắng về việc chia sẻ mã nguồn. Chính những ưu điểm này đã giúp Qt trở thành công cụ phát triển phổ biến trong cả cộng đồng mã nguồn mở và ngành công nghiệp phần mềm.

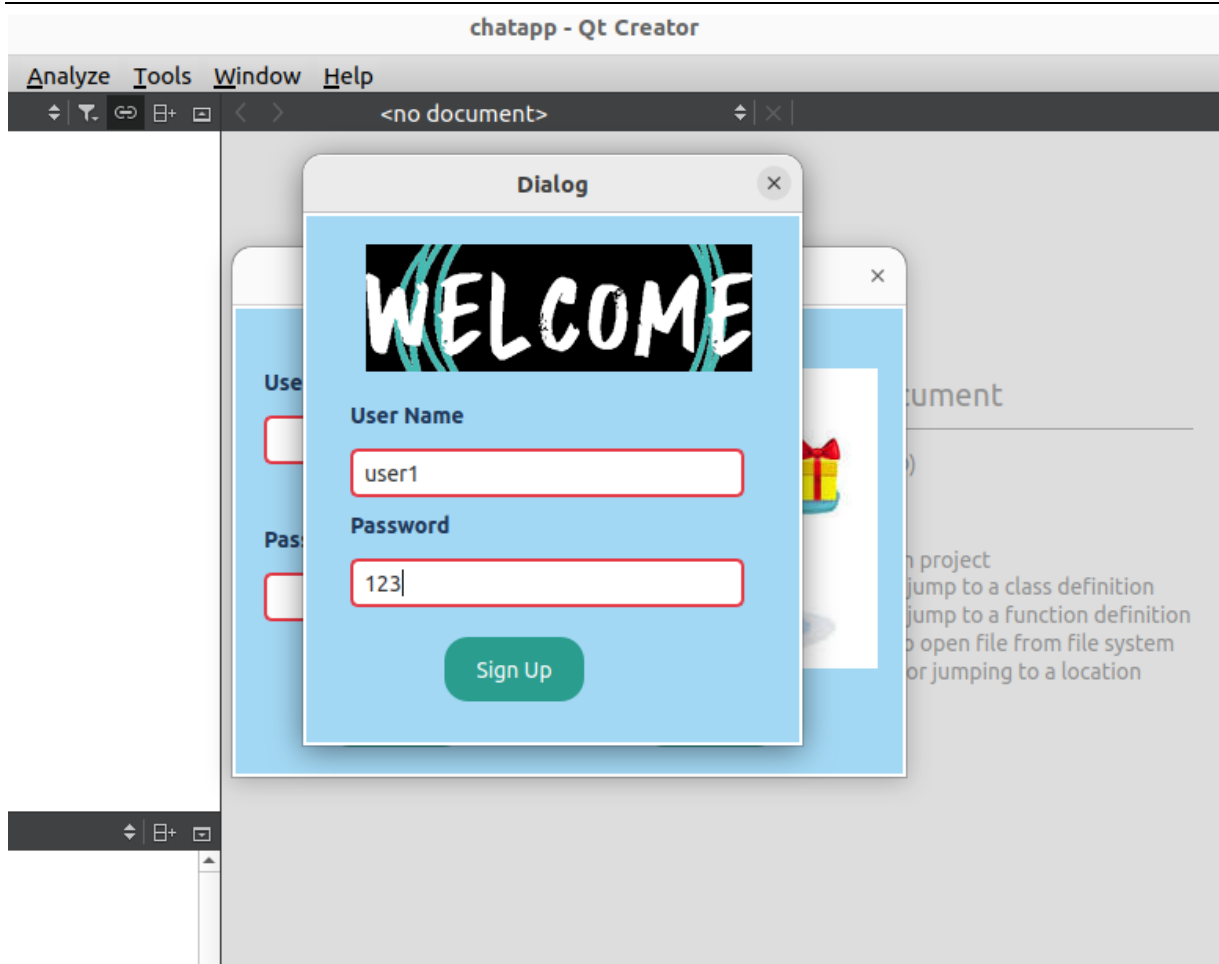
## IV. KẾT QUẢ VÀ ĐÁNH GIÁ

### 1. Xây dựng ứng dụng

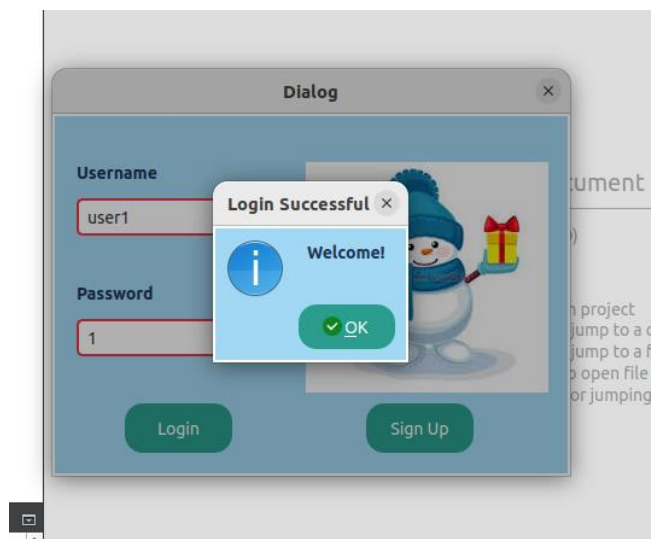
Sau quá trình xây dựng ứng dụng, nhóm đã thu được ứng dụng sản phẩm với các chức năng đã đề ra.

Một số hình ảnh demo của ứng dụng:

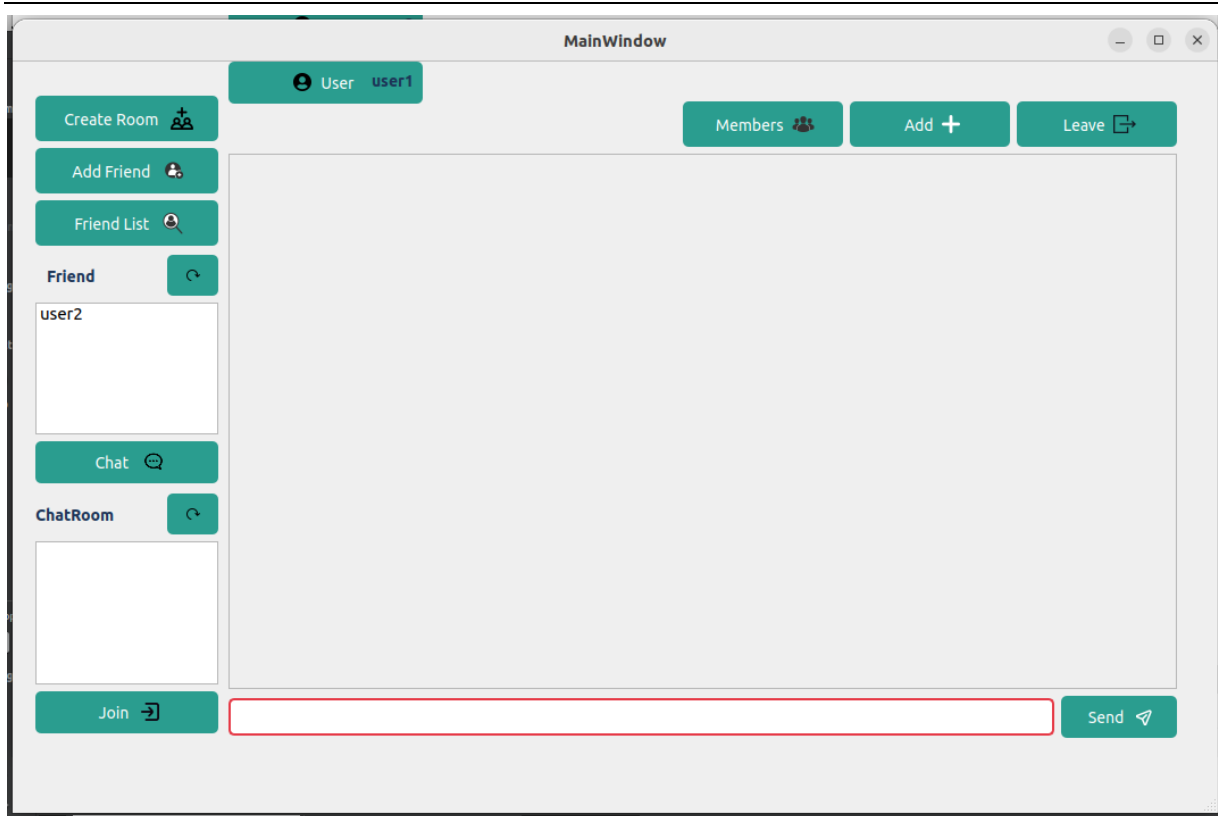
Trang đăng ký:



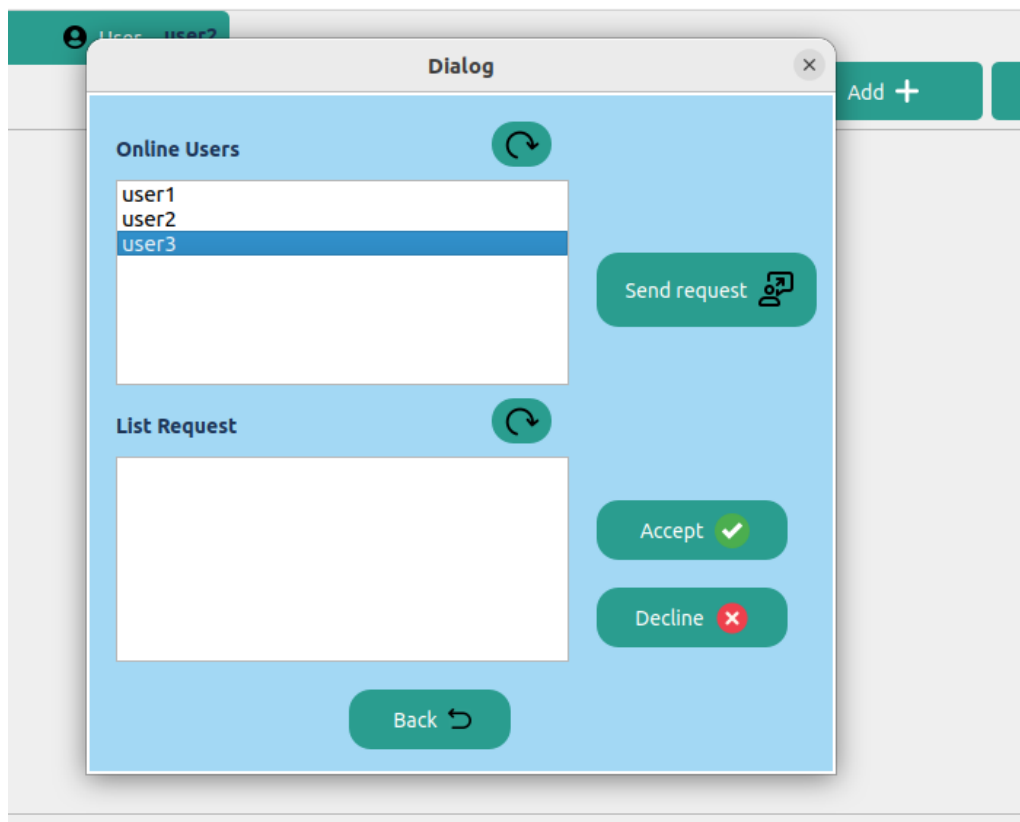
Trang đăng nhập:



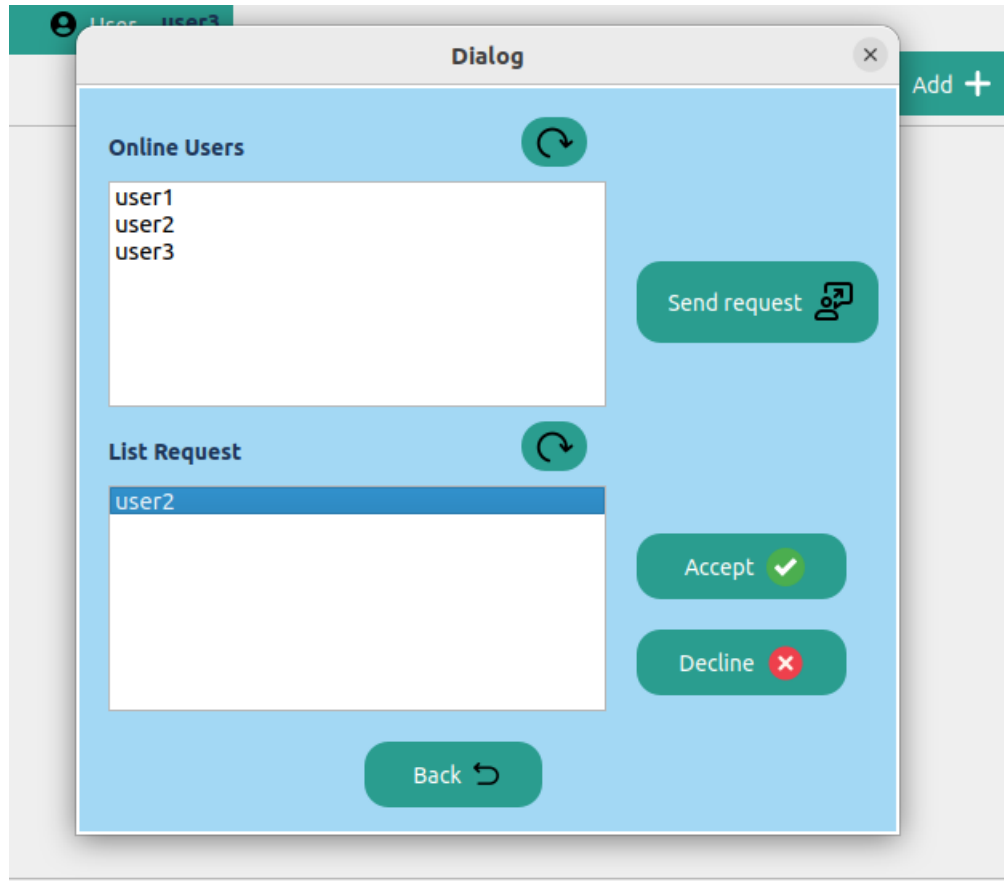
Màn hình chính:



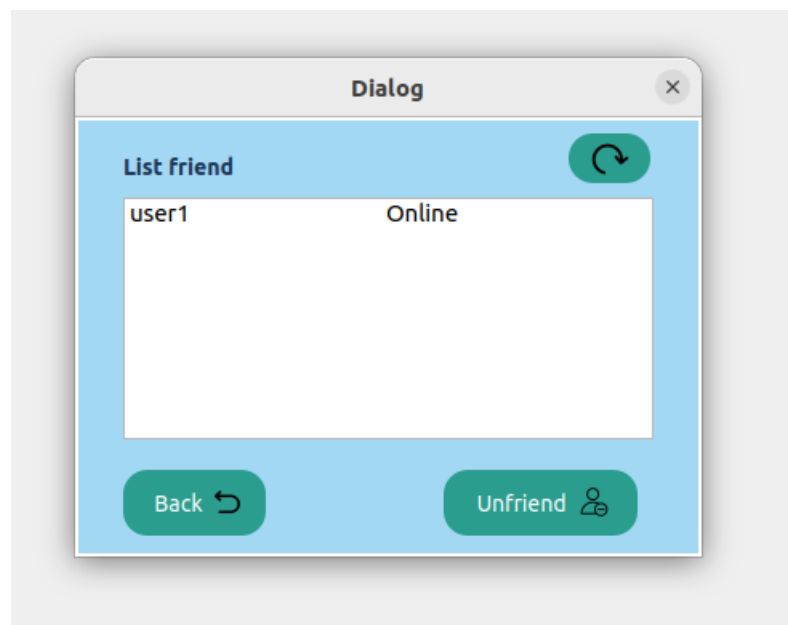
Xem danh sách user online và gửi lời mời kết bạn:



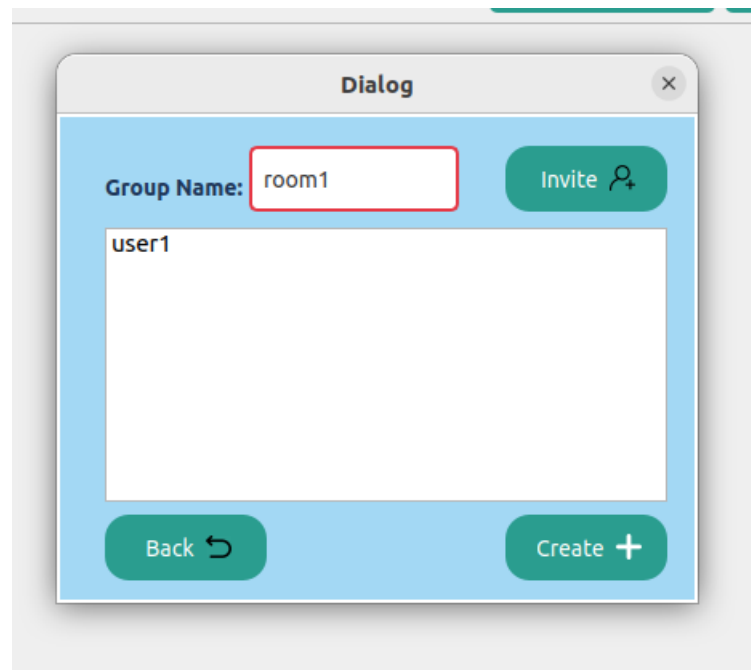
Xem lời mời kết bạn và đồng ý:



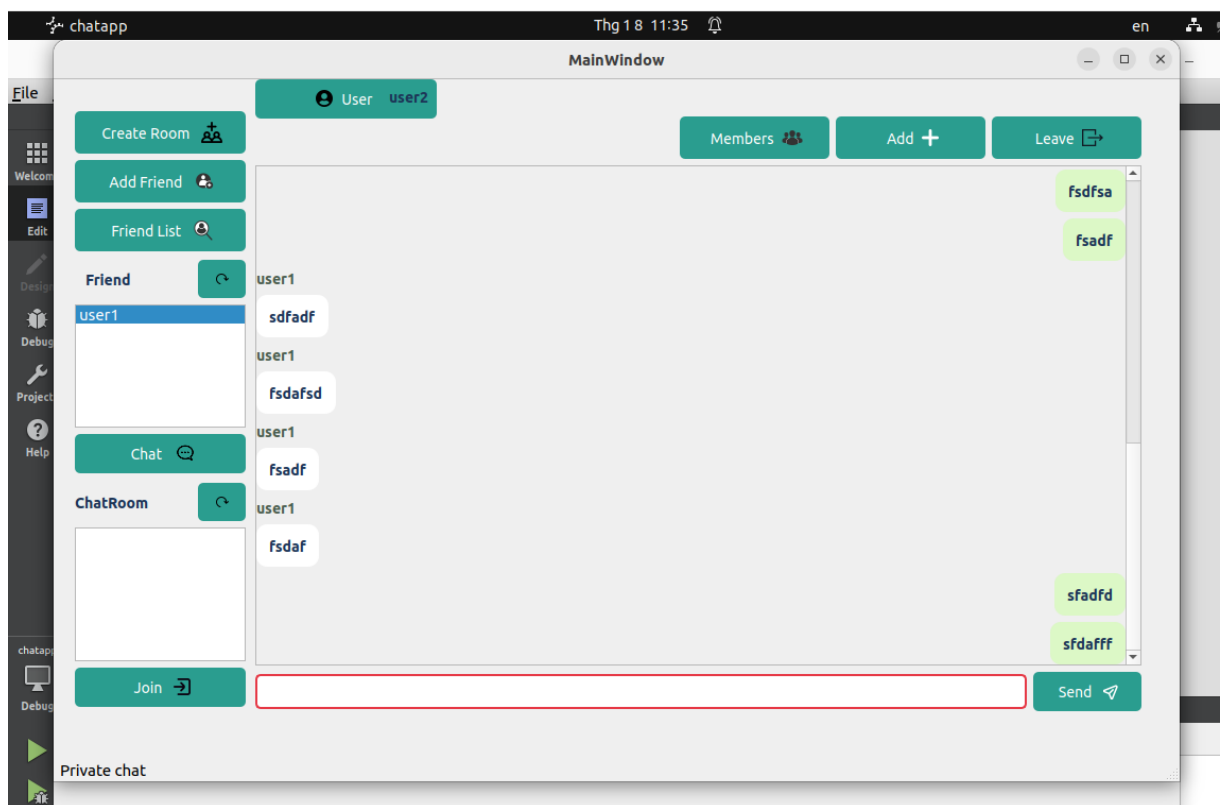
Xem danh sách bạn bè và trạng thái:



Tạo nhóm:

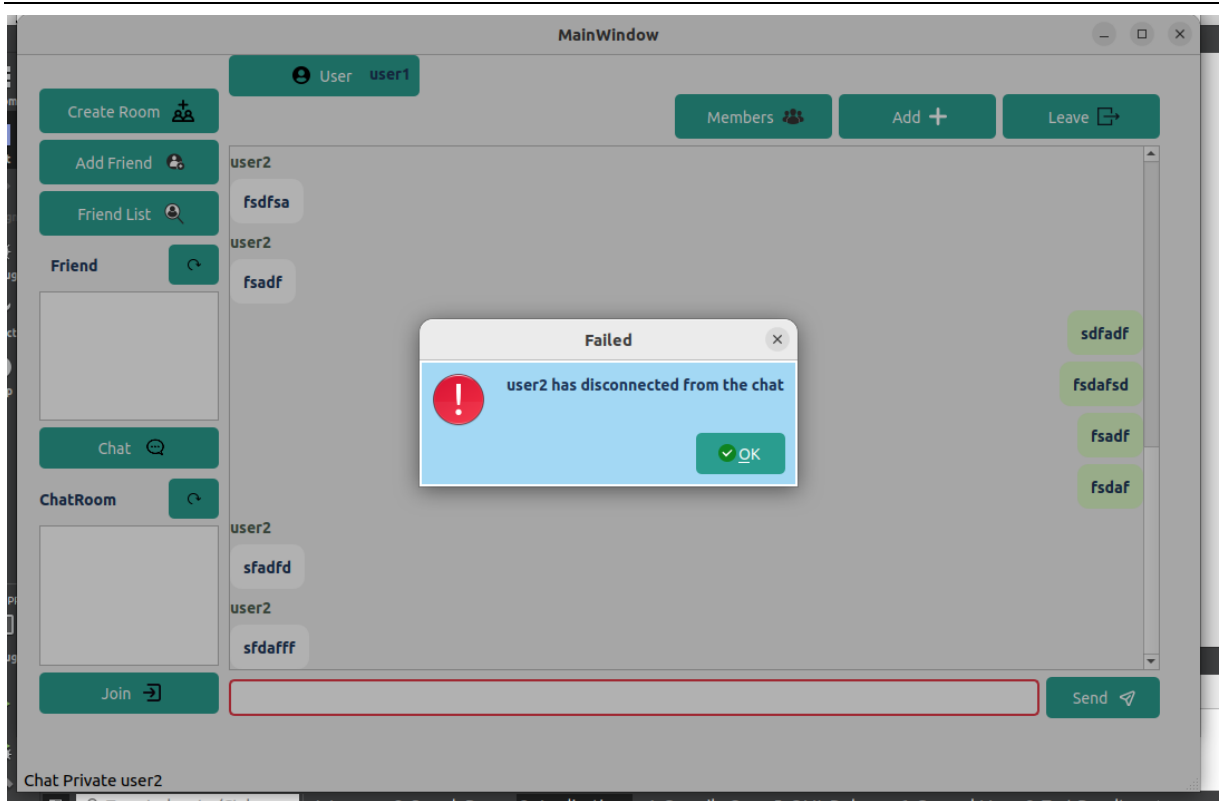


Gửi tin nhắn 1-1:

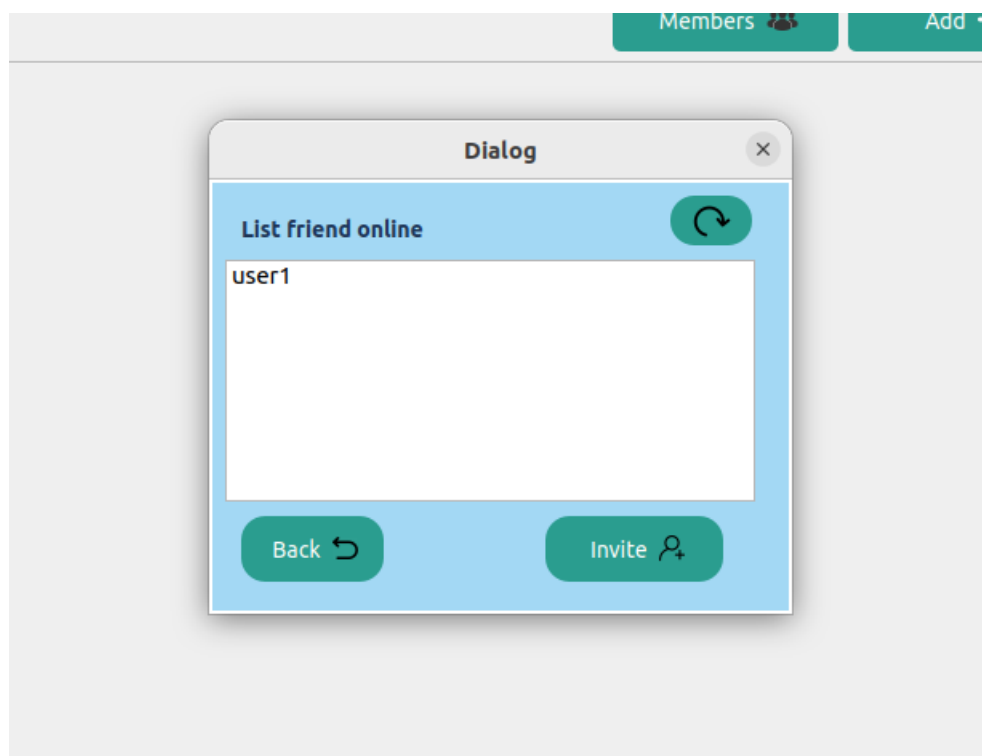


Người dùng ngắt kết nối khỏi chat:

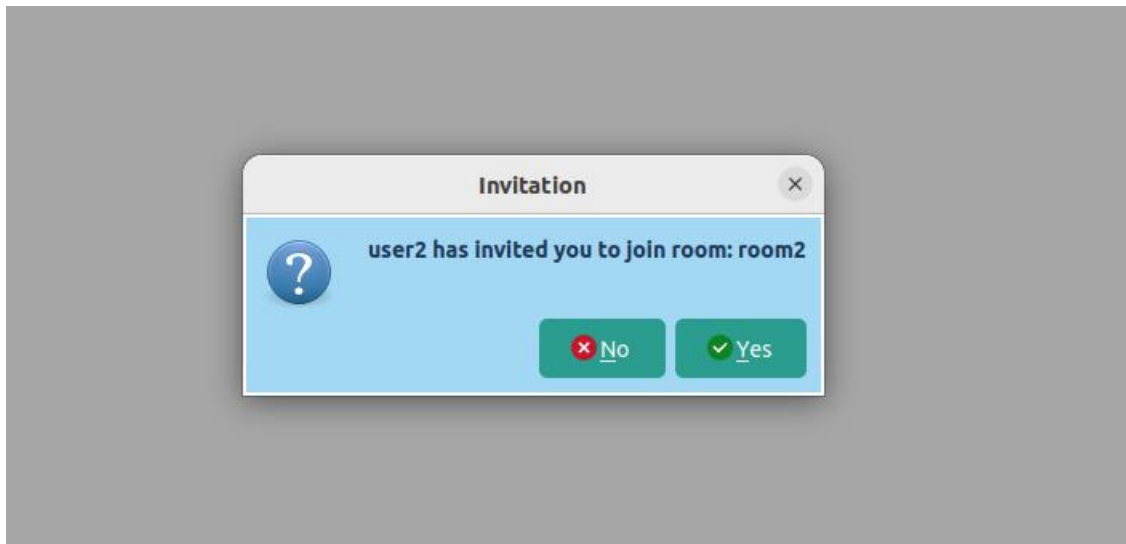




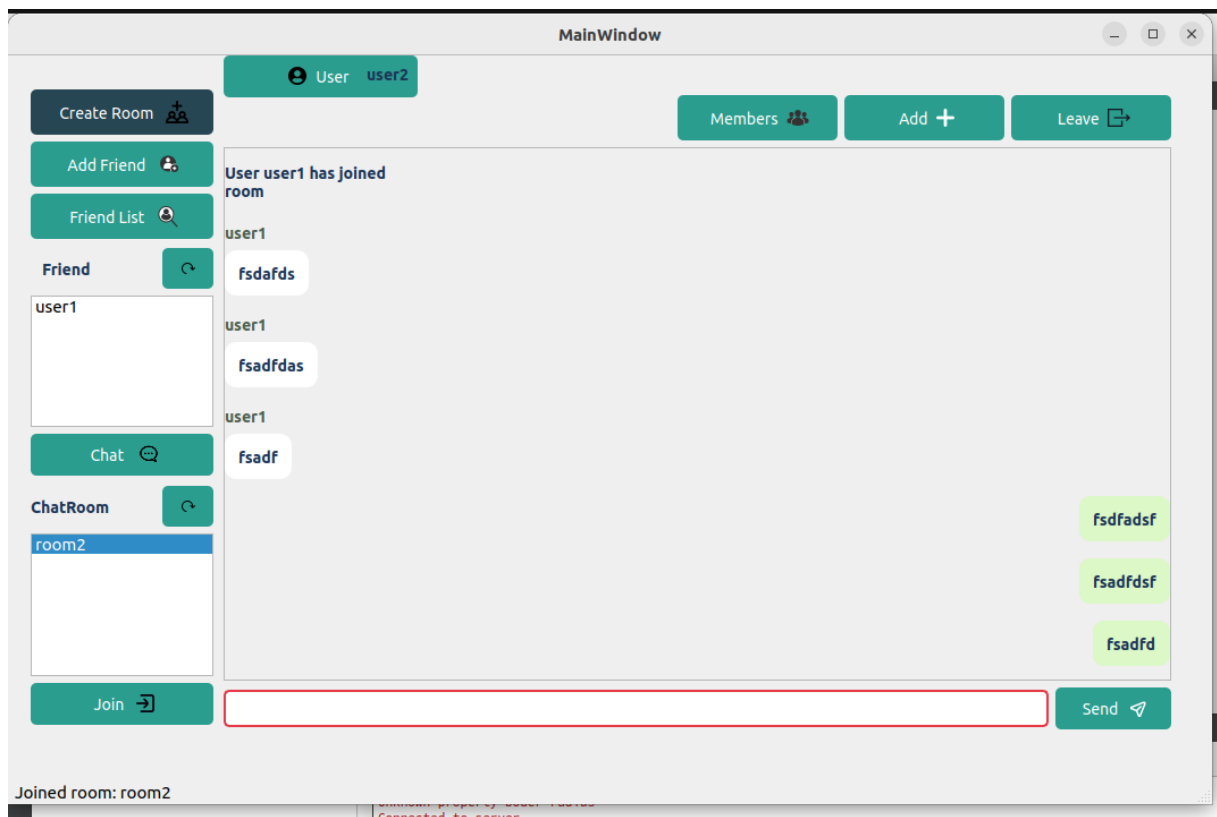
Gửi lời mời tham gia nhóm:



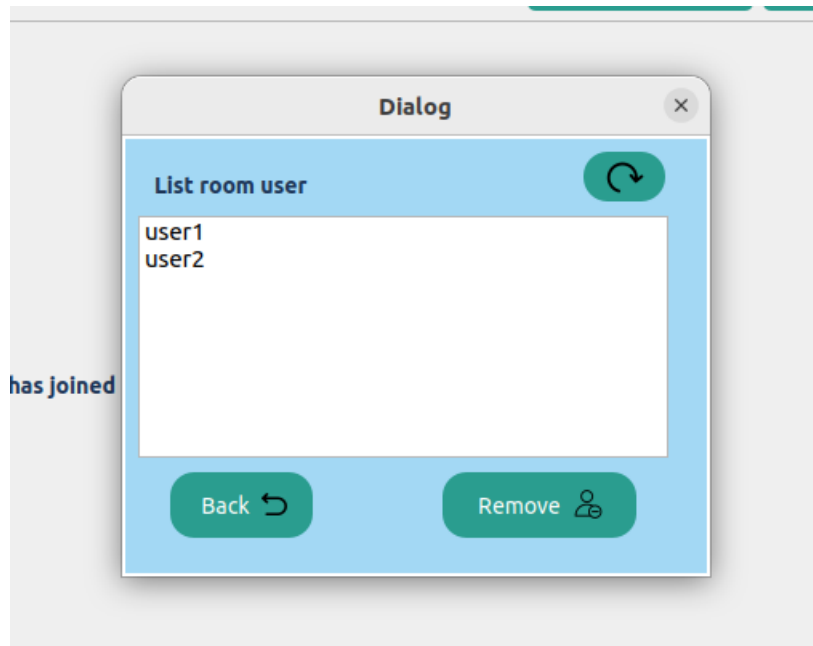
Đồng ý lời mời tham gia nhóm:



Gửi tin nhắn trong nhóm chat:



Xem danh sách member trong nhóm và xóa member khỏi nhóm:



## 2. Đánh giá

Ứng dụng chat đơn giản là một công cụ giao tiếp cơ bản nhưng hiệu quả, cung cấp các chức năng thiết yếu như gửi và nhận tin nhắn theo thời gian thực. Giao diện thường được thiết kế tối giản, dễ sử dụng, phù hợp với mọi đối tượng người dùng. Tuy nhiên, các tính năng nâng cao như bảo mật tin nhắn, gửi file, hoặc hỗ trợ nhóm chat có thể còn hạn chế. Đây là lựa chọn tốt cho những ai cần một giải pháp nhanh gọn, không phức tạp, nhưng có thể cần cải tiến để cạnh tranh với các ứng dụng chat phổ biến hiện nay.

## TÀI LIỆU THAM KHẢO

- [1] TS. Tống Văn Vạn (2023). *Slide bài giảng môn Thực hành Lập trình mạng*