HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

**KHOA CÔNG NGHỆ THÔNG TIN 1**

**-----🙠🕮🙢 -----**



**BÁO CÁO BÀI TẬP LỚN**

**ĐỀ TÀI : XÂY DỰNG ỨNG DỤNG CHAT**

**MÔN : LẬP TRÌNH MẠNG**

**Danh sách thành viên nhóm :**

1. Phan Đức Mạnh - B15DCCN347

2. Hoàng Xuân Quyết - B15DCCN446

3. Nguyễn Minh Việt - B15DCCN644

4. Trịnh Văn Tuân - B15DCCN600

**Hà Nội, Tháng 10/2018**

Mục lục

1. Phát biểu bài toán

Ngày nay, nhu cầu về công nghệ thông tin trong đời sống là đa dạng. Việc mở rộng các hệ thống truyền thông và ngày có nhiều máy vi tính kết nối vào mạng Internet. Với việc ứng dụng giao thức TCP/IP làm cho hệ thống mạng ngày càng rộng hơn và phát triển vượt bậc. Vấn đề an ninh, bảo mật,…là một thế mạnh của giao thức này đem lại cho công nghệ truyền thông. Bên cạnh việc phát triển của thư điện tử bằng nhiều dịch vụ khác nhau(gmail, yahoo,MSM trong mạng điện thoại di động…), việc CHAT trực tuyến trên Internet cũng là nhu cầu không thể thiếu.

Với thực tế như vậy, nhóm chúng em đã nghiên cứu và xây dựng mô hình CHAT trong mạng LAN theo mô hình Client – Server sử dụng công nghệ Socket.

2. Phân tích thiết kế hệ thống

### 2.1 Phân tích yêu cầu

### Yêu cầu của bài toán: Xây dựng mô hình Client – Server ứng dụng CHAT

\*) Xây dựng chương trình phía Server

- Tạo một TCP Socket và gắn vào một cổng

- Xây dựng một chương trình Server đa tuyến (Threaded Server) để cho phép nhiều Client kết nối tới Server.Mỗi tuyến đảm nhận liên lạc với Client.

- - Chờ và lắng nghe yêu cầu kết nối từ Client chấp nhận kết nối và nhận Socket tương ứng.

- Truyền nhận thông tin qua các luồng nhận/gửi dữ liệu của socket.

- Khi một user login vào Server thì Server sẽ cập nhật user đó và gửi tới các Client đang kết nối

- Khi một user logout ra khỏi Server thì Server hiện thông báo user đó đã logout và gửi tới Client.

- Đóng kết nối.

\*) Xây dựng chương trình phía Client.

- Tạo một TCP Socket với địa chỉ IP và số cổng mà chương trình Server đang chạy

- Thiết lập kết nối tới Server trao đổi dữ liệu với Server.

- Thêm bạn bè với các user khác

- Chat riêng với các user khác

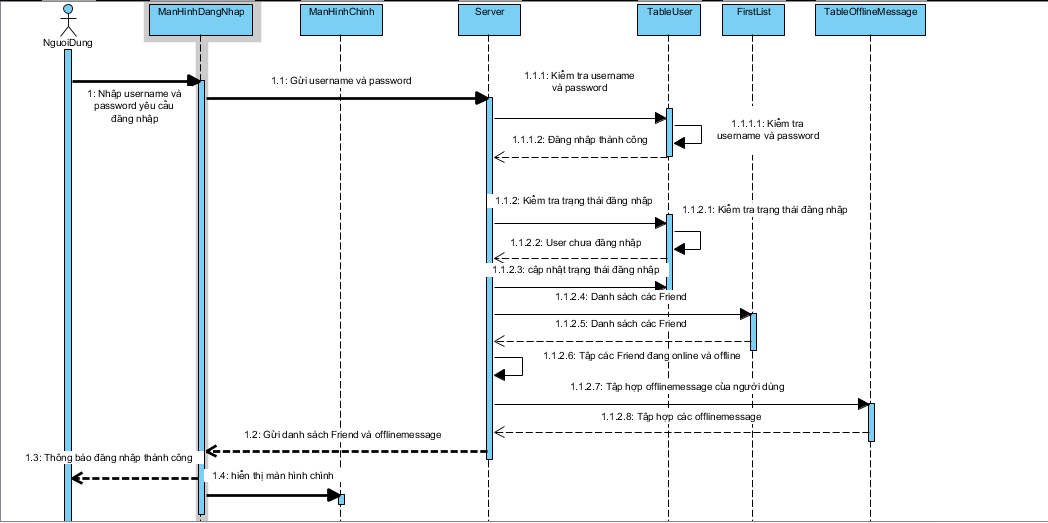
- Tạo group chat

- Gửi/ nhận thông điệp tới tất cả mọi người có trong phòng chat

- Đóng kết nối

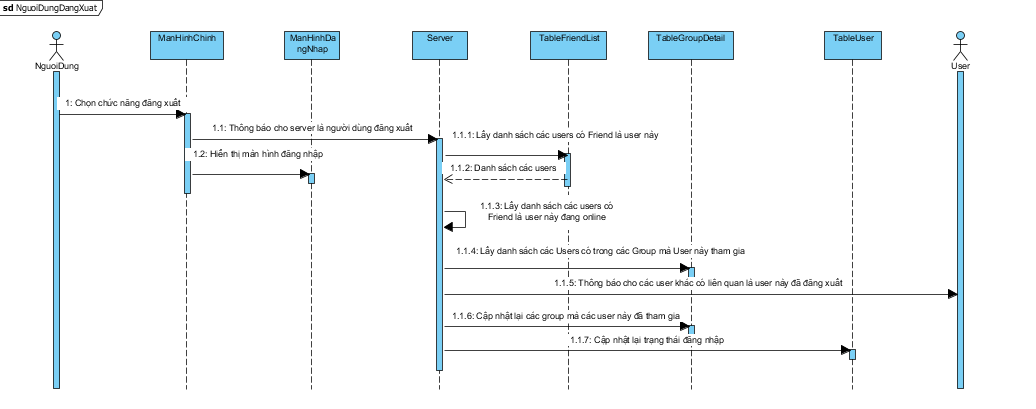
2.2 Phân tích và thiết kế

2.2.1. Mô hình xử lý đăng nhập



Khi một User yêu cầu đăng nhập hệ thống, Client sẽ gởi Username và Password cho Server. Server sẽ kiểm tra Username và Password có hợp lệ hay không. Sau đó, Server sẽ kiểm tra Username này đã đăng nhập chưa. Nếu đăng nhập thành công, Server sẽ lẫy danh sách các Friend đang Offline và Online của User kèm theo danh sách các tin nhắn Offline (nếu có) và gởi cho User và cập nhật lại trạng thái đăng nhập của User. Đồng thời cũng gởi thông báo đến các users khác có Friend là User này mới Online.

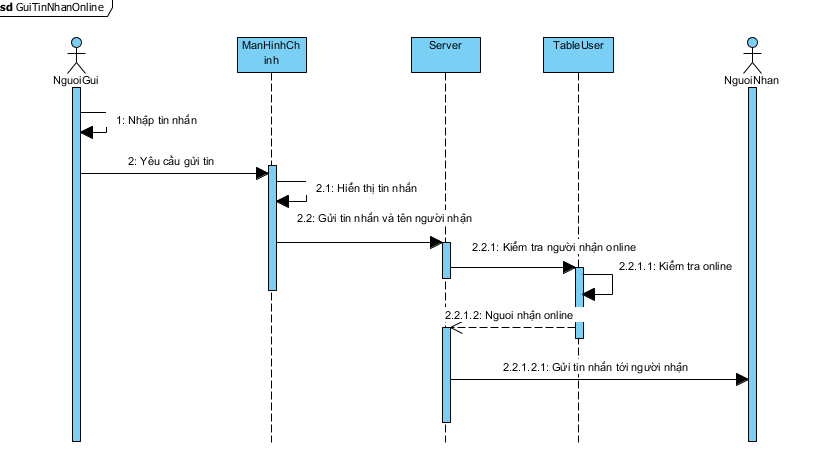
2.2.2. Xử lý đăng xuất



Khi một Ủser yêu cầu đăng xuất lại hệ thống, Client sẽ hiển thị lại màn hình đăng nhập và đồng thời gởi thông báo đang xuất đến Server. Server sẽ gởi thông báo đăng xuất tới các Users có Friend là User này. Bên cạnh đó, Server kiểm tra tất cả các Groups mà User này đang tham gia. Server sẽ gởi thông báo hủy đến các Group mà User này là người khởi tạo và sẽ gởi thông báo đăng xuất đến các Group mà User này chỉ tham gia với tư cách là thành viên. Cuôi cùng, Server sẽ cập nhật lại trạng thái đăng nhập của ser.

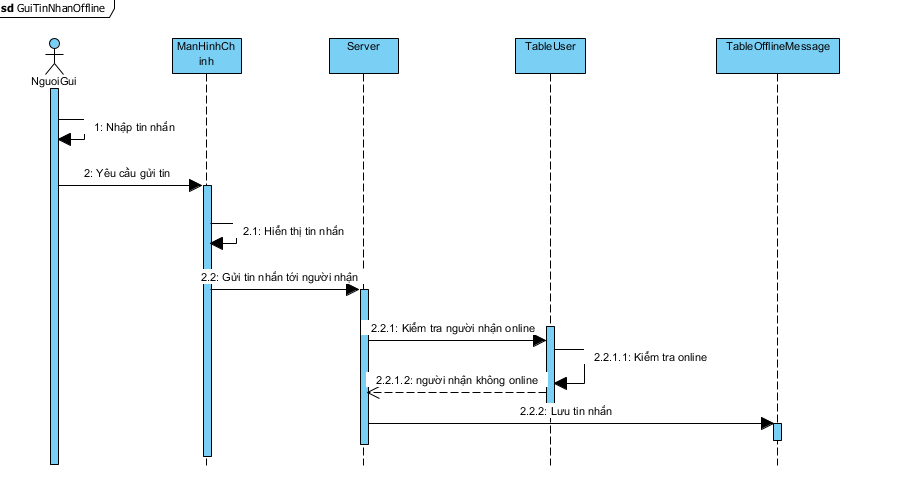
2.2.3. Các sử lý đối với việc gửi tin nhắn

###### + Gởi tin nhắn Online:



Khi người dùng nhập tin nhắn và nhắn phím Enter (hoặc button Senđ), Client sẽ hiển thị tin nhắn trên Tab Chat đồng thời gởi tin nhắn và tên người nhận lên cho Server. Server sẽ kiểm tra xem người nhận có phải đang Online hay không. Nếu người nhận đang Online, Server sẽ gởi tin nhắn và tên người gởi. Khi nhận được tin nhắn, Client sẽ hiển thị tin nhắn thông qua một Tab Chat (Tab này sẽ được mở nếu nó chưa có trước đó).

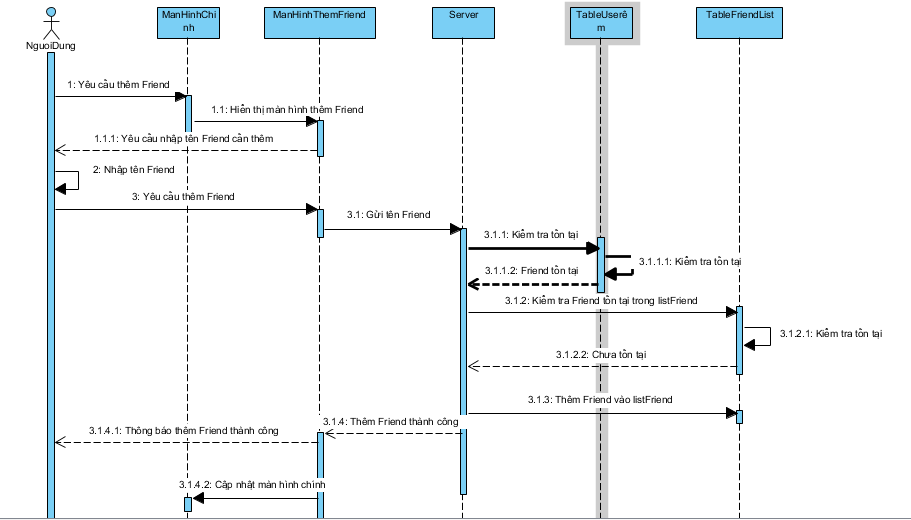
##### Gởi tin nhắn Offline:



Khi người dùng nhập tin nhắn và nhắn phím Enter (hoặc button Senđ), Client sẽ hiển thị tin nhắn trên Tab Chat đồng thời gởitin nhắn và tên người nhận lên cho Server. Server sẽ kiểm tra xem người nhận có phải đang Online hay không. Nếu người nhận đang Offline, Server sẽ lưu tin nhắn cùng tên người gởi vào bảng OfflineMessage để gởi cho người nhận ở lần đăng nhập tiếp theo.

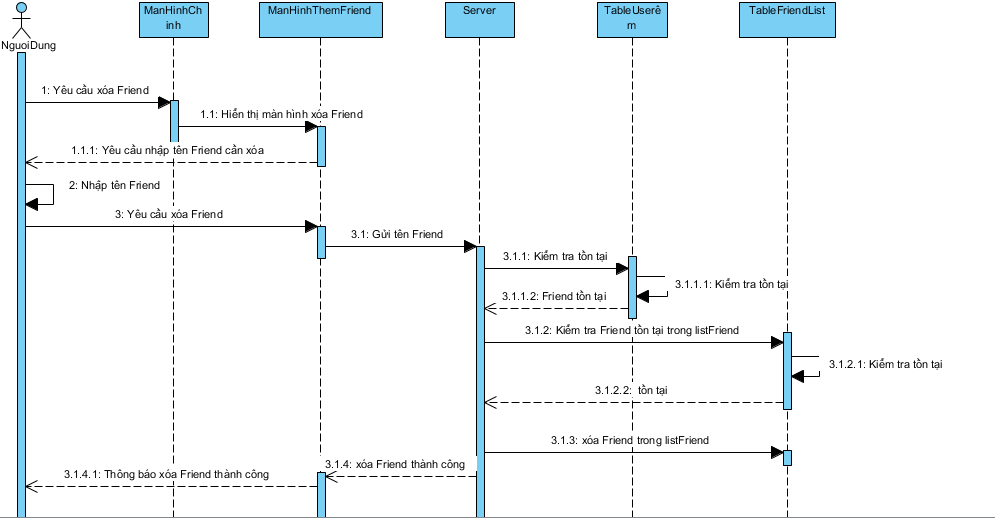
##### 2.2.4. Các xử lý đối với FriendList

###### + Thêm một Friend vào FriendList:



Khi người dùng chọn chức năng thêm một Friend vào FriendLIist, màn hình thêm FriendLIst sẽ được mở ra. Người dùng sẽ nhập tên của Friend và sau đó ClHient sẽ gởi tên Friend này lên cho Server. Trước tiên, Server sẽ kiểm tra Friend này có tồn tại hay không. Tiếp theo, sẽ kiểm tra Friend này đã được thêm vào FriendList trước đó hay chưa. Nếu Friend này chưa có trong FriendList, Server sẽ thêm Friend này vào FriendList của người dùng. Cuối cùng, Server sẽ gởi kết quả của công việc về cho Client. Dựa vào kết quả nhận được, Client sẽ thông báo cho người dùng biết là việc thêm thành công haythất bại (có hai nguyên nhân thất bại là Friend không tồn tại và Friend đã có trong FriendList rồi).

###### + Xóa một Friend ra khỏi FriendList:



Khi người dùng chọn chức năng xóa một Frlend vào FriendLIist, màn hình thêm FriendLIst sẽ được mở ra. Người dùng sẽ nhập tên của Friend và sau đó ChHient sẽ gởi tên Friend này lên cho Server. Trước tiên, Server sẽ kiểm tra Friend này có tồn tại hay không. Tiếp theo, sẽ kiểm tra Friend này đã được thêm vào FriendList trước đó hay chưa. Nếu Friend này đã có trong FriendList, Server sẽ xóa Friend này ra khỏi FriendList của người dùng. Cuối cùng, Server sẽ gởi kết quả của công việc về cho Client. Dựa vào kết quả nhận

##### được, Client sẽ thông báo cho người dùng biết là việc Xóa thành công

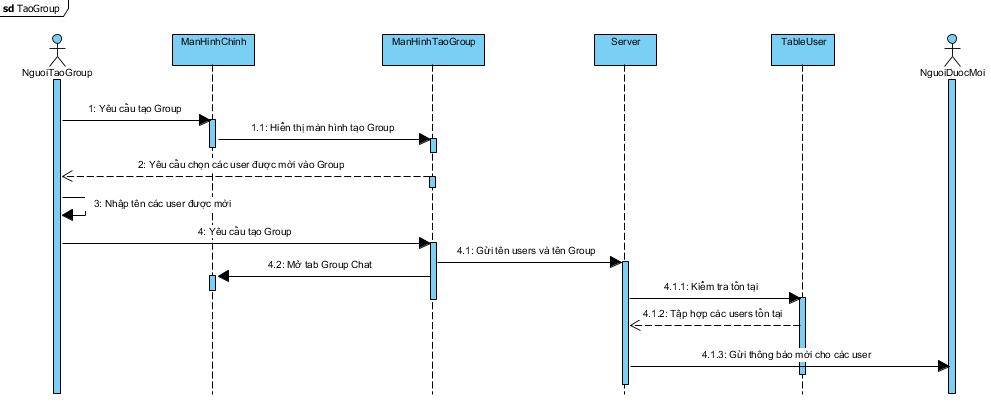
hay thất bại (có hai nguyên nhânthất bại là Friend không tôn tại và Friend chưa có trong FriendList) đồng thời cập nhật lại màn hình chính nếu cần.

> Ghi chú:

Chức năng này có thể được gọi khi người dùng nhấn phím Delete trong Listbox FriendLIst.

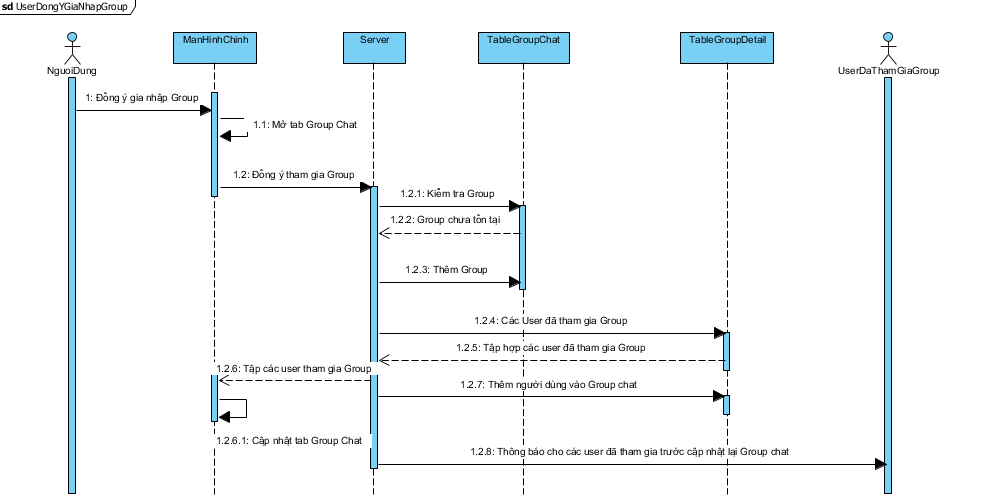
* + 1. Các xử lý đối với Group Chat

###### 2.2.5.1 Khi User tạo Group:



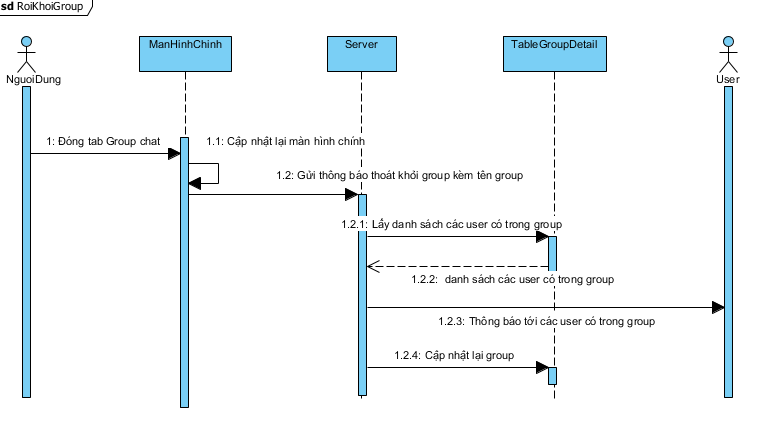
Khi người dùng chọn chức năng Invite Group thì màn hình tạo Group sẽ xuất hiện. Người dùng nhập danh sách các Users muốn mời và có thể nhập thêm Invite Message rồi nhắn Invite. Client sẽ tự phát sinh ra tên Group và sẽ gởi kèm nó với danh sách các Users muốn mời đồng thời cũng mở thêm một Tab Group Chat. Server sẽ lọc ra các sers đang Online trong danh sách các sers được mời. Sau đó, Server sẽ gởi lời mời kèm theo tên người mời đến các Users được mời.

2.2.5.2 Khi User đồng ý gia nhập Group:



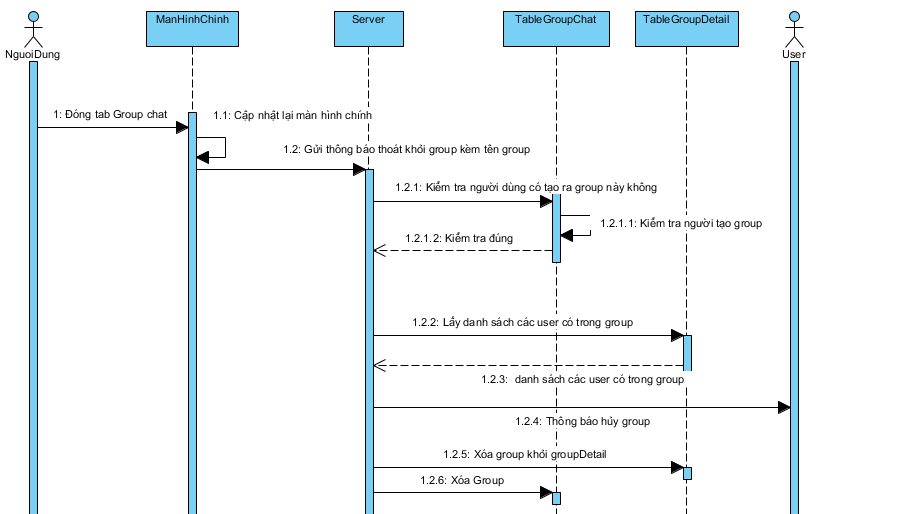
Khi người dùng đồng ý lời mời gia nhập nhóm, Client sẽ gởi thông báo đồng ý cho Server. Server sẽ kiểm tra xem Group này đã được lưu hay chưa (Group chưa được lưu nếu chưa có người dùng nào đồng ý tham gia). Sau đó, Server sẽ lưu người dùng này vào GroupDetail. Cuối cùng, Server sẽ gởi tên tất cả các Users đã tham gia Group cho người dùng, đồng thời cũng gởi thông báo đến cho các Users trong Group là người dùng này đã đồng ý gia nhập nhóm.

2.2.5.3 Khi User thoát khỏi Group



Khi một người dùng tắt Tab Group Chat, Client sẽ gởi thông báo thoát khỏi Group cho Server. Server trước tiên kiểm tra người dùng này có phải là người đã tạo Group hay không. Nếu không phải, Server sẽ gởi thông báo tới các Users khác trong Group. Cuối cùng, Server sẽ cập nhật lại GroupDetail.

2.2.5.4 Khi User hủy Group:

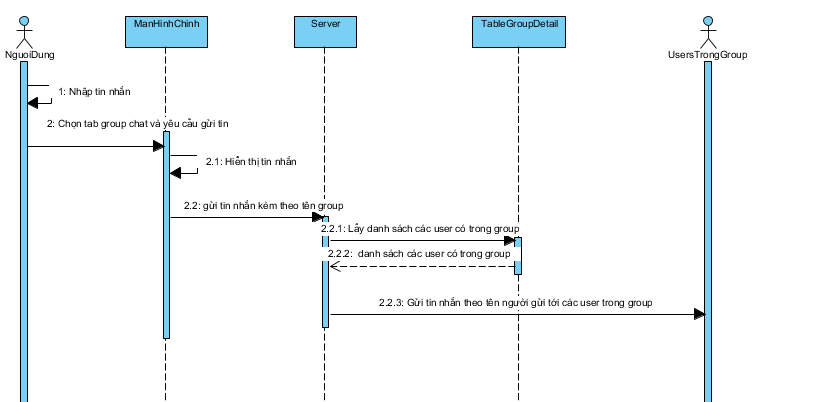


Khi một người dùng tắt Tab Group Chat, Client sẽ gởi thông báo thoát khỏi Group cho Server. Server trước tiên kiểm tra người dùng này có phải là người đã tạo Group hay không. Nếu đúng, Server sẽ gởi thông báo hủy Group tới các Users khác trong Group. Cuối cùng, Server sẽ xóa Group này khỏi GroupDetail và GroupChat.

Ghi chú:

Khi nhận được thông báo hủy Group từ Users, Client sẽ tắt Tab Group Chat tương ứng đồng thời thông báo cho người dùng.

2.2.5.5 Khi User gởi tin nhắn tới các thành viên trong Group:



Khi người dùng nhập tin nhắn và nhấn button Send trong Tab Group Chat, Client sẽ gởi tin nhắn và tên Group cho Server. Dựa vào tên Group nhận được, Server sẽ tìm tất cả các Users đã tham gia Group và gởi tin nhắn kèm theo tên người gởi đến các Users này.

* + 1. Thiết kế dữ liệu

Gồm có các bảng Users, FriendList, OfflineMessage, GroupChat, GroupDetail để lưu trử các thông tin vê user đê đăng nhập, quan hệ giữa các user, lưu tin nhăn offline của các user và lưu tạm thời thông tin các nhóm Chat.

##### Chuẩn hóa dữ liệu:

> Users (Username, Password, State, FirstName,LastName, Email, Address)

> FriendList(UserID,FriendID)

> OfflineMessage(UserID,ToUserID, Message,CreateDate)

> Message(MessageID,FromUserID, ToUserID,content, CreateDate,groupID)

> GroupChat(GroupID, UserID, GroupName, CreateTime)

> GroupDetail(GroupID, ToUserID)

###### + Table Users

> Chức năng: Lưu thông tin về User

> Danh sách các thuộc tính:

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Tên thuộc tính | Kiểu dữ liệu | Diễn giải |
| 1 | username | Nvarchar(45) | Tên đăng nhập |
| 2 | Password | Nvarchar(45) | Mật khâu đăng nhập |
| 3 | State | int (1) | Trạng thái của Ủser |
| 4 | FirstName | Nvarchar(45) | Tên |
| 5 | LastName | Nvarchar(45) | Họ |
| 6 | Email | Nvarchar(45) | Email của Ủser |
| 7 | Address | Nvarchar(45) | Địa chỉ của User |

Table FriendList

> Chức năng: Lưu thông tin về quan hệ giữa các User

>\_ Danh sách các thuộc tính

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Tên thuộc tính | Kiểu dữ liệu | Diễn giải |
| 1 | UserID | Int (4) | ID của người nhận, Khóa chính |
| 2 | FriendID | Int (4) | ID của Friend, Khóa chính |

###### Table OfflineMessage

> Chức năng: Lưu các tin nhắn offline của User

> Danh sách các thuộc tính:

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Tên thuộc tính | Kiểu dữ liệu | Diễn giải |
| 1 | FromUserId | Nvarchar(45) | Tên đăng nhập |
| 2 | toUserID | Nvarchar(45) | Mật khâu đăng nhập |
| 3 | message | Text | Trạng thái của Ủser |
| 4 | createdDate | DateTime | Tên user |

###### Table Message

> Chức năng: Lưu các tin nhắn đã gửi

> Danh sách các thuộc tính:

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Tên thuộc tính | Kiểu dữ liệu | Diễn giải |
| 1 | messagesId | Int(11) | Id của tin nhắm |
| 2 | fromUserId | Int(11) | ID người gửi |
| 3 | toUserId | Int(11) | ID người nhận |
| 4 | content | Text | Nội dung tin nhắn |
| 5 | createdDate | DateTime | Thời gian gửi |
| 6 | groupId | Int(11) | Group gửi |

###### Table GroupChat

> Chức năng: Lưu tạm thời thông tin của các nhóm Chat

> Danh sách các thuộc tính:

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Tên thuộc tính | Kiểu dữ liệu | Diễn giải |
| 1 | GroupID | Int (11) | Tự động tăng, Khóa chính |
| 2 | UserID | Int (11) | ID của User tạo nhóm |
| 3 | GroupName | varchar (45) | Tên của nhóm |
| 4 | CreateDate | datetime | Thời đim tạo nhóm |

###### Table GroupDetail

> Chức năng: Lưu thông tin chi tiết của các nhóm Chat

> Danh sách các thuộc tính

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Tên thuộc tính | Kiểu dữ liệu | Diễn giải |
| 1 | GroupID | Int (11) | ID của Group, Khóa chính |
| 2 | ToUserID | Int (11) | User nhận, khóa chính |

1. Khó khăn

Client liên tục gửi requet để nhận tin nhắn

Biện pháp: khi server nhận được thông điệp gửi tin từ clientA -> clientB

* + 1. Server lưu vào db
  + 2. Đồng thời nếu client B có trong list online thì gửi tin nhắn đó cho client B do luồng nhận luôn được mở