

ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH

TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN  
KHOA MẠNG MÁY TÍNH VÀ TRUYỀN THÔNG



# ĐỒ ÁN CUỐI KỲ XÂY DỰNG HỆ THỐNG CHAT QUA MẠNG LAN

17520988

Nguyễn Hồng Sơn

17520915

Nguyễn Duy Phước

*Thành phố Hồ Chí Minh, năm 2019*



# LỜI CẢM ƠN

Thực tế luôn cho thấy, sự thành công nào cũng đều gắn liền với những sự hỗ trợ, giúp đỡ của những người xung quanh. Trong suốt thời gian từ khi bắt đầu làm bài đồ án đến nay, chúng em đã nhận được sự quan tâm, chỉ bảo, giúp đỡ của thầy cô, gia đình và bạn bè xung quanh.

Với tấm lòng biết ơn vô cùng sâu sắc, chúng em xin gửi lời cảm ơn chân thành đến quý thầy cô Trường Đại học Công nghệ Thông tin - ĐHQG HCM đã dùng những tri thức và tâm huyết của mình để có thể truyền đạt cho chúng em trong vốn kiến thức quý báu suốt thời gian qua.

Đặc biệt, chúng em xin chân thành cảm ơn ThS Trần Hồng Nghi đã tận tâm chỉ bảo, hướng dẫn chúng em qua từng buổi học. Chúng em cũng xin gửi lời cảm ơn tới giảng viên thực hành thầy Đồng Tiến Dũng. Nhờ có những lời hướng dẫn, chỉ bảo đó, bài đồ án chúng em đã hoàn thành một cách xuất sắc nhất. Một lần nữa, em xin gửi lời cảm ơn chân thành tới cô và thầy.

Bài đồ án là bài tập lớn dưới dạng tiểu luận này là lần đầu tiên chúng em thực hiện. Ban đầu còn bỡ ngỡ vì vốn kiến thức còn hạn hẹp. Do vậy, không tránh khỏi những thiếu sót, chúng em mong nhận được ý kiến đóng góp của quý Thầy, Cô để bài đồ án được hoàn thiện hơn.

Chúng em xin chân thành cảm ơn!

Thành phố Hồ Chí Minh, ngày 12 tháng 6 năm 2019

Nhóm sinh viên thực hiện



# MỤC LỤC

LỜI CẢM ƠN	1
MỤC LỤC	2
DANH MỤC HÌNH ẢNH	3
DANH MỤC TỪ VIẾT TẮT	3
I. GIỚI THIỆU TỔNG QUAN ĐỀ TÀI	4
1. Lý do chọn đề tài	4
2. Mục đích nghiên cứu	4
3. Đối tượng nghiên cứu	5
4. Phạm vi nghiên cứu	5
5. Phương pháp nghiên cứu	5
6. Bố cục đề tài	5
II. CƠ SỞ LÝ THUYẾT, CÔNG NGHỆ SỬ DỤNG	6
1. Cơ sở lý thuyết	6
2. Công nghệ sử dụng	7
III. PHÂN TÍCH THIẾT KẾ HỆ THỐNG ĐỀ TÀI	8
1. Về phía server	9
2. Về phía Client	10
IV. HIỆN THỰC ĐỀ TÀI, KẾT QUẢ THỰC HIỆN	11
1. Về phía chức năng đã thực hiện được	11
2. Về các chức năng chưa hiện thực thành công	23
V. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN	23

# DANH MỤC HÌNH ẢNH

1. Sơ đồ tổng quan hệ thống
2. Sơ đồ cơ sở dữ liệu
3. Giao diện chính của server
4. Public chat
5. Sơ đồ vận hành public chat
6. Tổng quan hiện thực hệ thống chat
7. Một số nhân sử dụng trong hệ thống
8. Giao diện màn hình đăng nhập
9. Lỗi đăng nhập
10. Xử lý logic đăng nhập
11. Form Users 's Data
12. Thêm dữ liệu User
13. Xóa dữ liệu User
14. Sửa dữ liệu User
15. Tìm kiếm thông tin User
16. Message Database

# DANH MỤC TỪ VIẾT TẮT

- |         |                           |
|---------|---------------------------|
| 1. CHAT | Trò chuyện                |
| 2. LINQ | Language Integrated Query |
| 3. SQL  | Structured Query Language |

## I. GIỚI THIỆU TỔNG QUAN ĐỀ TÀI

### 1. Lý do chọn đề tài

Hệ thống trò chuyện trực tuyến đầu tiên được gọi là Talkomatic, được tạo bởi Doug Brown và David R. Woolley vào năm 1973 trên Hệ thống PLATO tại Đại học Illinois. Nó cung cấp một số kênh, mỗi kênh có thể chứa tối đa năm người, với các thông báo xuất hiện trên tất cả các màn hình theo từng ký tự của người dùng khi họ được nhập. Talkomatic rất phổ biến đối với người dùng PLATO vào giữa những năm 1980. Vào năm 2014, Brown và Woolley đã phát hành phiên bản Talkomatic dựa trên web.

Hệ thống trực tuyến đầu tiên sử dụng lệnh "trò chuyện" thực tế đã được tạo ra cho The Source vào năm 1979 bởi Tom Walker và Fritz Thane của Dialcom, Inc.

Cuộc trò chuyện Internet xuyên Đại Tây Dương đầu tiên diễn ra giữa Oulu, Phần Lan và Corvallis, Oregon vào tháng 2 năm 1989.

Dịch vụ trò chuyện trực tuyến chuyên dụng đầu tiên được cung cấp rộng rãi cho công chúng là CompuServe CB Simulator vào năm 1980, được tạo bởi giám đốc điều hành CompuServe Alexander "Sandy" Trevor ở Columbus, Ohio. Tổ tiên bao gồm phần mềm trò chuyện mạng như "nói chuyện" UNIX được sử dụng trong những năm 1970.<sup>1</sup>

Dựa trên những cơ sở đó, với những kiến thức được tiếp xúc trong môn Lập trình mạng căn bản, nhóm sinh viên thực hiện đề tài “Hệ thống chat qua mạng LAN” nhằm thực hiện một số nhu cầu cơ bản là trò chuyện trong các hệ thống khép kín như Trường học, các văn phòng cỡ nhỏ, các phòng máy,... Sử dụng hệ thống trò chuyện nội bộ đảm bảo được thông tin trao đổi không bị truyền ra ngoài hay bị hack qua mạng Internet.

### 2. Mục đích nghiên cứu

Mặc dù hệ thống chat không còn xa lạ và hiện nay được ứng dụng rộng rãi nhưng đó chưa bao giờ là vấn đề không quan trọng đối với các tổ chức và doanh nghiệp.

Với đề tài này, mục đích lớn nhất là có thể tự xây dựng một hệ thống chat của riêng mình, đảm bảo được tính bảo mật và kiểm soát thông tin nội bộ

---

<sup>1</sup> Wikipedia: Online chat



trong phạm vi nhỏ lẻ. Hiểu được những yêu cầu kỹ thuật và các vấn đề gặp phải của một hệ thống chat. Từ đó, kiến nghị các giải pháp thay thế, bổ sung các điều kiện thực tế nhằm nâng cao hiệu quả sử dụng hệ thống chat.

### 3. Đối tượng nghiên cứu

Đối tượng nghiên cứu chính là các sinh viên đang theo học tại trường, các hệ thống chat mà sinh viên thường sử dụng cùng cơ chế hoạt động của chúng.

### 4. Phạm vi nghiên cứu

Trong khuôn khổ của một bài đồ án nhỏ, nhóm sinh viên thực hiện nhằm tới một số chức năng cơ bản nhất mà một hệ thống chat cần có. Thời gian và độ rộng của kiến thức không cho phép sinh viên xây dựng một hệ thống đầy đủ và có nhiều tính năng ưu việt như các hệ thống sẵn có trên thị trường.

### 5. Phương pháp nghiên cứu

Trong quá trình nghiên cứu và thực hiện đề tài, chúng em có sử dụng một số phương pháp nghiên cứu sau:

- Tham khảo cách xây dựng một hệ thống chat cơ bản qua các diễn đàn chia sẻ .
- Phân tích các chức năng của phần mềm được xây dựng sẵn, rút ra các cốt lõi cần có của một hệ thống.
- Tham khảo ý kiến của các thầy cô và các bạn khác thực hiện chung hướng đề tài.
- Sử dụng phương pháp suy luận, phân tích để xây dựng các chức năng mới.
- Và một số phương pháp khác.

### 6. Bố cục đề tài

Để phù hợp với mục tiêu, đối tượng, phạm vi nghiên cứu cũng như nội dung của đề tài, bố cục của đề tài như sau:

- Chương 1: Giới thiệu tổng quan đề tài.
- Chương 2: Cơ sở lý thuyết và công nghệ sử dụng.
- Chương 3: Phân tích thiết kế hệ thống.
- Chương 4: Hiện thực đề tài.
- Chương 5: Kết quả thực hiện và bài học rút ra.

- Chương 6: Kết luận và hướng phát triển.

## II. CƠ SỞ LÝ THUYẾT, CÔNG NGHỆ SỬ DỤNG

### 1. Cơ sở lý thuyết

Nhằm giải quyết vấn đề của đề tài là chat qua mạng, sinh viên nghiên cứu các phương pháp truyền và nhận gói tin an toàn thông qua giao thức mạng TCP/IP.

**Bộ giao thức TCP/IP**, (tiếng Anh: *Internet protocol suite* hoặc *IP suite* hoặc *TCP/IP protocol suite* - bộ giao thức liên mạng), là một bộ các giao thức truyền thông cài đặt chồng giao thức mà Internet và hầu hết các mạng máy tính thương mại đang chạy trên đó. Bộ giao thức này được đặt tên theo hai giao thức chính của nó là TCP (*Giao thức Điều khiển Giao vận*) và IP (*Giao thức Liên mạng*). Chúng cũng là hai giao thức đầu tiên được định nghĩa.

Như nhiều bộ giao thức khác, bộ giao thức TCP/IP có thể được coi là một tập hợp các tầng, mỗi tầng giải quyết một tập các vấn đề có liên quan đến việc truyền dữ liệu, và cung cấp cho các giao thức tầng cấp trên một dịch vụ được định nghĩa rõ ràng dựa trên việc sử dụng các dịch vụ của các tầng thấp hơn. Về mặt logic, các tầng trên gần với người dùng hơn và làm việc với dữ liệu trừu tượng hơn, chúng dựa vào các giao thức tầng cấp dưới để biến đổi dữ liệu thành các dạng mà cuối cùng có thể được truyền đi một cách vật lý.<sup>2</sup>

Giao thức TCP/IP có những ưu điểm vượt trội đó là truyền tin một cách tin cậy. Nhờ đó, những tin nhắn chat của các thành viên sử dụng hệ thống chat không bị mất mát.

Nhưng đối với phạm vi của đề tài không thể mở rộng sử dụng mạng internet, sinh viên sử dụng phương pháp kết hợp với hệ thống port, các địa chỉ IP localhost để kết nối các client là các process trong một máy với nhau.

Nhằm mục đích truyền gói tin mà hầu hết là các kí tự tin nhắn, sinh viên thiết lập một hệ thống sử dụng các stream để tạo nên socket.

Với mục đích là tối ưu hóa các phương tiện trên và tăng thêm tính bảo mật và dễ dàng quản lý các nội dung trò chuyện, sinh viên sử dụng mô hình Client - Server thay vì sử dụng mô hình Peer to Peer. Ưu điểm của mô hình Client - Server là máy con (đóng vai trò là máy khách) gửi một yêu cầu

---

<sup>2</sup> Wikipedia: TCP/IP

(request) để máy chủ (đóng vai trò người cung ứng dịch vụ), máy chủ sẽ xử lý và trả kết quả về cho máy khách.<sup>3</sup> Sự hoạt động của Client hoàn toàn phụ thuộc vào Server. Nhưng đối với mô hình, việc đòi hỏi có một Server mạnh để phục vụ số lượng lớn Client. Đây là vấn đề lớn, khó có thể xử lý một cách hiệu quả. Để đơn giản hơn, sinh viên sử dụng cơ chế đa luồng để phục vụ một số lượng có hạn Client.

Ngoài ra, để lưu trữ thông tin của các Client (sau đây gọi là user), sinh viên xây dựng một hệ thống lưu trữ nhỏ dựa trên nền tảng kỹ thuật LINQ. Sử dụng một số câu lệnh truy vấn SQL cơ bản để lưu dữ liệu đăng nhập và tin nhắn vào cơ sở dữ liệu đồng thời gọi dữ liệu từ đây.

## 2. Công nghệ sử dụng

C# (đọc là "C thăng" hay "C sharp" ("xi-sáp")) là một ngôn ngữ lập trình hướng đối tượng được phát triển bởi Microsoft, là phần khởi đầu cho kế hoạch .NET của họ. Tên của ngôn ngữ bao gồm ký tự thăng theo Microsoft nhưng theo ECMA là C#, chỉ bao gồm dấu số thường. Microsoft phát triển C# dựa trên C++ và Java. C# được miêu tả là ngôn ngữ có được sự cân bằng giữa C++, Visual Basic, Delphi và Java.

C# được thiết kế chủ yếu bởi Anders Hejlsberg kiến trúc sư phần mềm nổi tiếng với các sản phẩm Turbo Pascal, Delphi, J++, WFC.<sup>4</sup>

Lấy Window form làm đối tượng thiết kế chủ yếu, các thao tác của phần mềm đều dựa trên các hỗ trợ có sẵn của window form. Các tính năng sẵn có của .NET framework vô cùng hữu ích và tiện lợi. Một số thư viện đã sử dụng:

- System.Net;
- System.Net.Sockets;
- System.Text;
- System.Threading;
- System.Web;
- System.Linq;

Và một số thư viện khác.

.NET framework đơn giản hóa việc viết ứng dụng bằng cách cung cấp nhiều thành phần được thiết kế sẵn, người lập trình chỉ cần học cách sử dụng và tùy theo sự sáng tạo mà gắn kết các thành phần đó lại với nhau. Nhiều công cụ được tạo ra để hỗ trợ xây dựng ứng dụng.NET, và IDE (*Integrated*

<sup>3</sup> Wikipedia: Client - Server

<sup>4</sup> Wikipedia: C#



*Development Environment*) được phát triển và hỗ trợ bởi chính Microsoft là Visual Studio.<sup>5</sup>

Ngoài những ưu điểm nói trên, sinh viên nhận thấy .NET là một framework là một thư viện đóng, có tính bảo mật và an toàn cao hơn rất nhiều so với các thư viện mở khác.

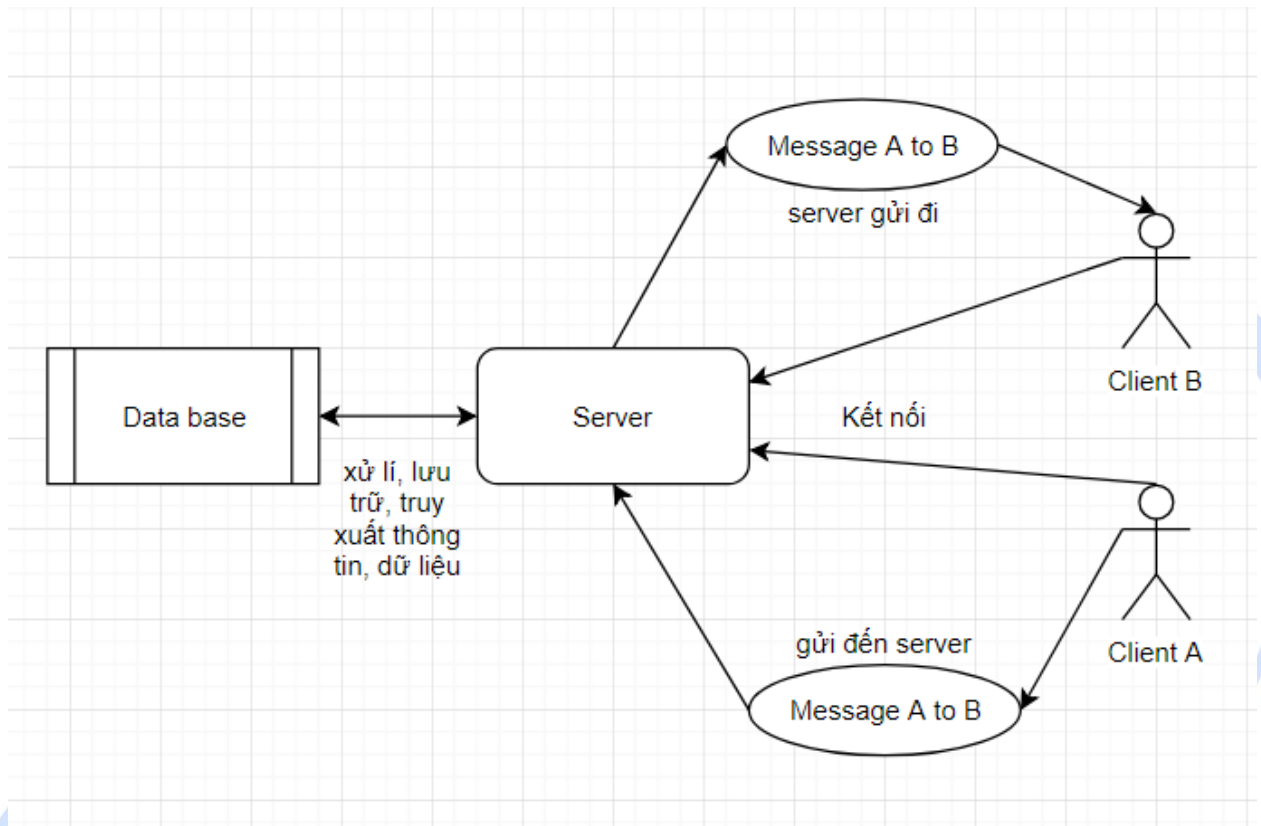
LINQ (Language Integrated Query, tạm dịch là ngôn ngữ truy vấn tích hợp) đưa ra 1 mô hình bền vững để hoạt động với các dạng nguồn dữ liệu và định dạng dữ liệu khác nhau. LINQ cho phép dùng các đoạn code đơn giản để truy vấn và chuyển đổi dữ liệu trong các tài liệu XML, cơ sở dữ liệu SQL, tập dữ liệu ADO.NET, các tập hợp .NET, và bất kỳ định dạng nào mà LINQ provider hỗ trợ.

### III. PHÂN TÍCH THIẾT KẾ HỆ THỐNG ĐỀ TÀI

Dựa vào các kiến thức nói trên, các công nghệ phù hợp đã chọn và các kiến thức được tiếp xúc trên lớp, cùng với những kiến thức khác đã tìm hiểu được thông qua mạng internet, sinh viên tiến hành xây dựng hệ thống bao gồm hai thành phần chính: Client và Server.

---

<sup>5</sup> Wikipedia: .NET Framework



**Hình 1. Sơ đồ tổng quan hệ thống**

## 1. Về phía server

Nắm vai trò vừa là đối tượng phục vụ các user, vừa là đối tượng quản lý các user đó, server có nhiều chức năng đa dạng với các mục đích khác nhau.

Đầu tiên và quan trọng nhất là vai trò trung gian và nhận kết nối từ các user. Để thực hiện được điều này, cần một kết nối luôn mở và các cấu trúc rẽ nhánh phù hợp để phân biệt các user khác nhau. Tham khảo các tài liệu, sinh viên tìm ra phương pháp đó là sử dụng các host name<sup>6</sup>. Dựa vào đó, sinh viên hiển thị các user khác nhau trên một bảng danh sách và quản lý chúng.

Đảm bảo hệ thống vẫn luôn mở và nhận kết nối, lại vừa đảm bảo tính khép kín và bảo mật, sinh viên xây dựng phương pháp xác thực cơ bản nhất là sử dụng Username – password. Sinh viên tiến hành xây dựng chức năng cho phép đăng ký từ phía user và tính năng thêm một user tại giao diện của chính server và lưu trữ nó trên server. Xử lý lưu trữ, nhận thấy nếu lưu trực tiếp cập

<sup>6</sup> Trong mạng máy tính, host name là nhãn được gán cho thiết bị được kết nối với mạng máy tính và được sử dụng để nhận dạng thiết bị trong các hình thức giao tiếp với các thiết bị khác nhau, như World Wide Web. Host name có thể là tên đơn giản bao gồm một từ hoặc một cụm từ hoặc chúng có thể được cấu trúc theo mẫu sẵn có.

username-password trên cơ sở dữ liệu của mình thì trong trường hợp bị đánh cắp dữ liệu, hoặc mất mát một phần vì kỹ thuật thì sẽ xảy ra các vấn đề về thông tin người dùng. Vì vậy, sinh viên nhắm tới sử dụng hash password<sup>7</sup> bằng phương thức được hỗ trợ ngay trong C# là MD5<sup>8</sup> sau đó mới lưu vào cơ sở dữ liệu. Bên cạnh đó, ngoài việc lưu “danh tính” các user, sinh viên nhận thấy cần phải lưu trữ tin nhắn giữa các user với nhau. Từ những điều đó, sinh viên xây dựng một sơ đồ cơ sở dữ liệu rất đơn giản.



*Hình 2: Sơ đồ cơ sở dữ liệu*

Ngoài ra, sinh viên nhận thấy server cũng cần có các chức năng trò chuyện với các user. Điển hình là các thông báo tới user các thông tin quan trọng sử dụng phương thức Broadcast<sup>9</sup>.

## 2. Về phía Client

Vì là một user bình thường, các tính năng xây dựng cho Client sẽ hạn chế. Đối với hệ thống trò chuyện mà sinh viên đang xây dựng, nhận thấy không cần quá nhiều chức năng cần cung cấp cho user nên user sẽ chỉ có các chức năng cơ bản sau:

- Đăng ký

<sup>7</sup> Hàm băm là bất kỳ hàm nào có thể được sử dụng để ánh xạ dữ liệu có kích thước tùy ý lên dữ liệu có kích thước cố định. Các giá trị được trả về bởi hàm băm được gọi là giá trị băm, mã băm, thông báo hoặc đơn giản là băm. Các hàm băm thường được sử dụng kết hợp với bảng băm, một cấu trúc dữ liệu phổ biến được sử dụng trong phần mềm máy tính để tra cứu dữ liệu nhanh chóng. Các hàm băm tăng tốc tra cứu bảng hoặc cơ sở dữ liệu bằng cách phát hiện các bản ghi trùng lặp trong một tệp lớn. Tính chất không thể đảo ngược của hàm băm giúp đảm bảo được tính toàn vẹn của dữ liệu khi có sự cố.

<sup>8</sup> Trong mật mã học, MD5 (viết tắt của tiếng Anh Message-Digest algorithm 5, giải thuật Tiêu hóa tin 5) là một hàm băm mật mã học được sử dụng phổ biến với giá trị Hash dài 128-bit. Là một chuẩn Internet (RFC 1321), MD5 đã được dùng trong nhiều ứng dụng bảo mật, và cũng được dùng phổ biến để kiểm tra tính toàn vẹn của tập tin. Một bảng băm MD5 thường được diễn tả bằng một số hệ thập lục phân 32 ký tự.

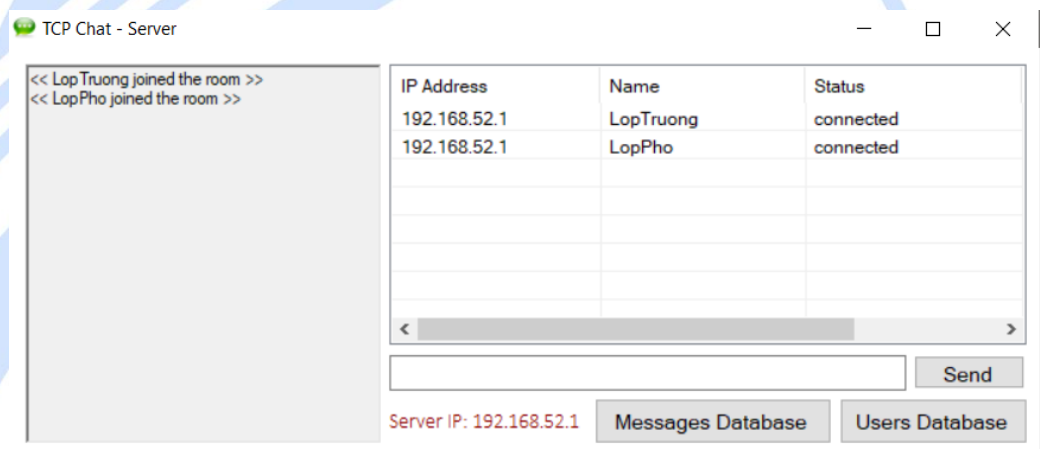
<sup>9</sup> Broadcast là thuật ngữ được sử dụng trong mạng máy tính để mô tả cách thức truyền tin được gửi từ 1 điểm đến tất cả các điểm khác trong cùng một mạng. Trong trường hợp này, Một gói broadcast chuyển đến tất cả những thiết bị tham gia trong một mạng cục bộ, mà không cần phải được quy định rõ ràng như một máy nhận.

- Đăng nhập
- Gửi tin và nhận tin nhắn chung trong kênh chat broadcast.
- Gửi và nhận tin nhắn riêng với các user khác và với server
- Gửi và nhận hình ảnh, icon

## IV. HIỆN THỰC ĐỀ TÀI, KẾT QUẢ THỰC HIỆN

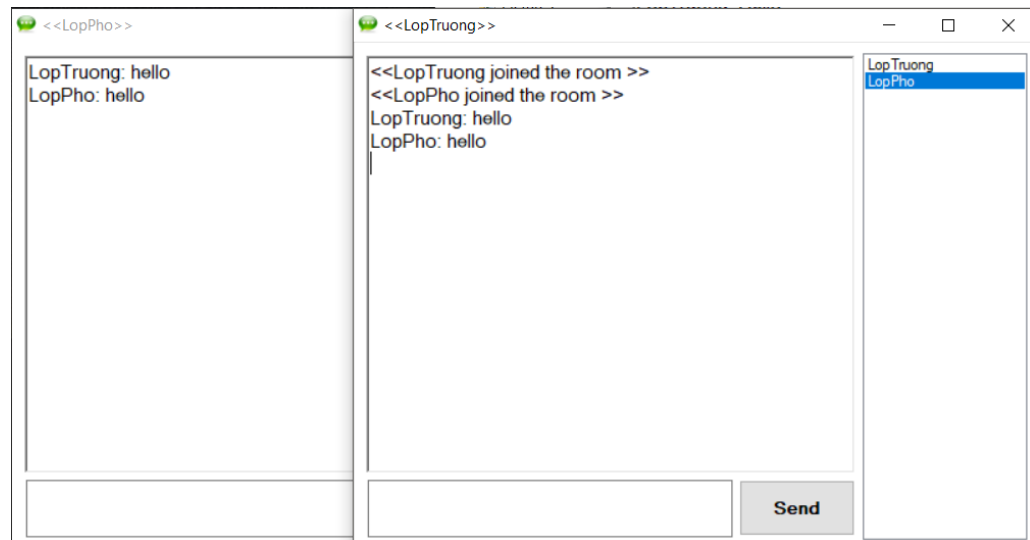
### 1. Về phía chức năng đã thực hiện được

Như đã đề cập ở trên, một số chức năng cơ bản của server đã được xây dựng thành công



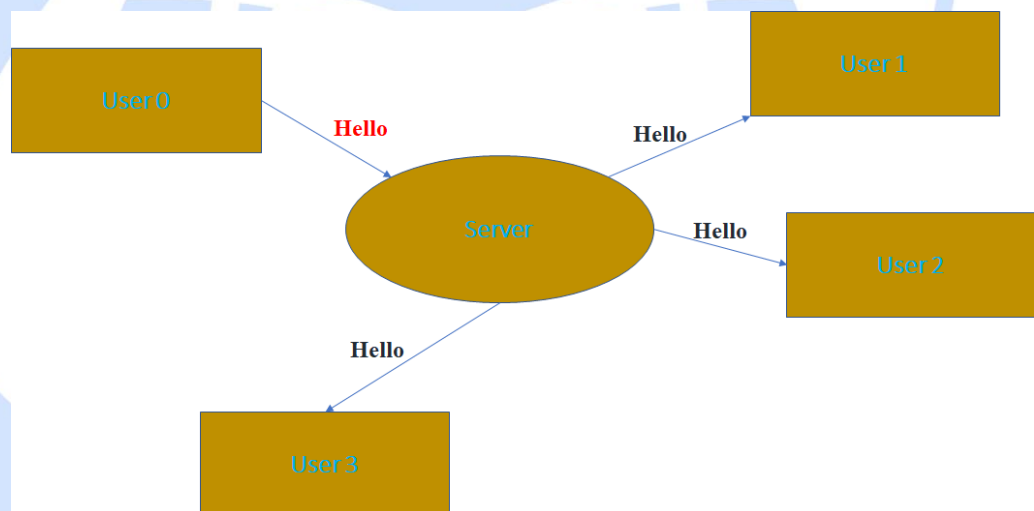
**Hình 3: Giao diện chính server**

Với mục tiêu đề tài là cung cấp một hệ thống trò chuyện bằng tin nhắn, sinh viên đã hoàn thành cơ bản được hệ thống trò chuyện bao gồm các tính năng: trò chuyện công khai trên toàn hệ thống (public chat), trò chuyện riêng với server, trò chuyện riêng với các user khác.



**Hình 4: Public chat**

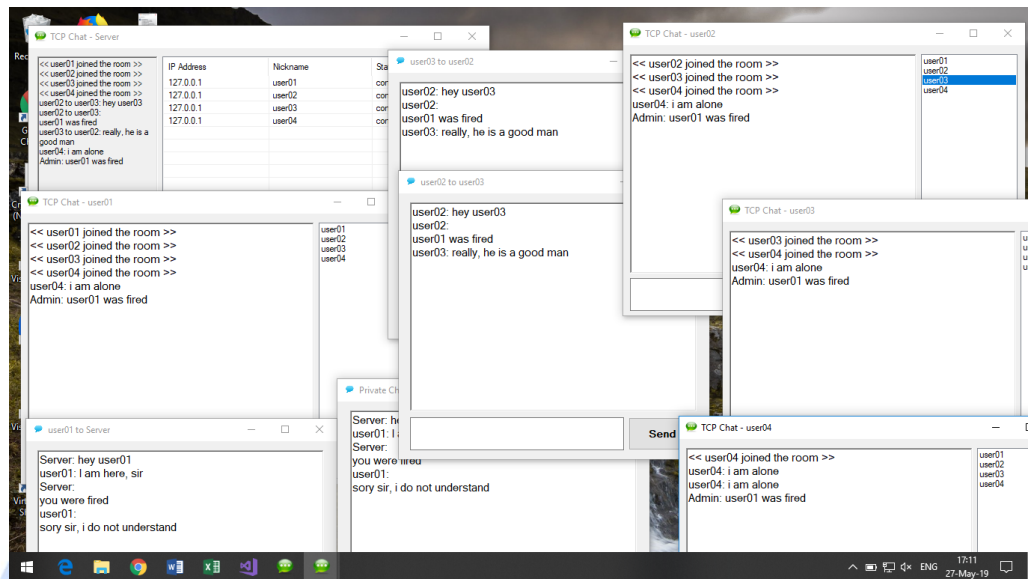
Trong tính năng public chat, các user gửi và nhận tin nhắn thông qua server. Tất cả các user có kết nối tới hệ thống đều sẽ nhận được tin nhắn từ bất kỳ user nào gửi. Khi một user chat public, thông tin đó sẽ được gửi tới server, server gửi broadcast tới tất cả các user.



**Hình 5: Sơ đồ vận hành public chat**

Bên cạnh đó, tính năng trò chuyện riêng tư giữa các user cũng được phát triển. Khung hội thoại trò chuyện riêng tư này là một khung riêng, hiển thị song song với khung chat public. Khi một user chat riêng tư với một user khác, cuộc trò chuyện này sẽ được hiển thị trên màn hình giao diện của server. Nếu đối tượng chat riêng tư là server thì đoạn trò chuyện này không hiển thị trên màn hình giao diện chính của server mà chỉ hiển thị trên khung chat riêng của server.





**Hình 6: Tổng quan hiện thực hệ thống chat**

Để hiện thực được những tính năng này, sinh viên tiến hành quy định một thư viện các nhãn dán đối với các thao tác khác nhau.

```
switch (command[0])
{
    case "Connect":
        ...
    case "Message":
        ...
    case "pMessage":
        ...
    case "pChat":
        ...
    case "pChatwithUser":
        ...
    case "pChatGroup":
        ...
    break;
}
```

**Hình 7: Một số nhãn sử dụng trong hệ thống**

Bằng việc sử dụng các nhãn để định nghĩa thao tác, hệ thống có thể dễ dàng được mở rộng thêm các tính năng dựa vào các nhu cầu thực tế.

Ngoài tính năng cơ bản là chat mà đề tài hướng tới, sinh viên còn xây dựng một số tính năng phụ mà sinh viên nhận thấy cần thiết đối với hệ thống. Đó là các tính năng đăng nhập từ phía user, tính năng lưu trữ và quản lý hệ cơ sở dữ liệu từ các user từ phía server.

Với truy vấn LINQ, ta có 3 tác vụ cơ bản:

- Kết nối với nguồn dữ liệu.
- Tạo truy vấn.
- Thực thi truy vấn.

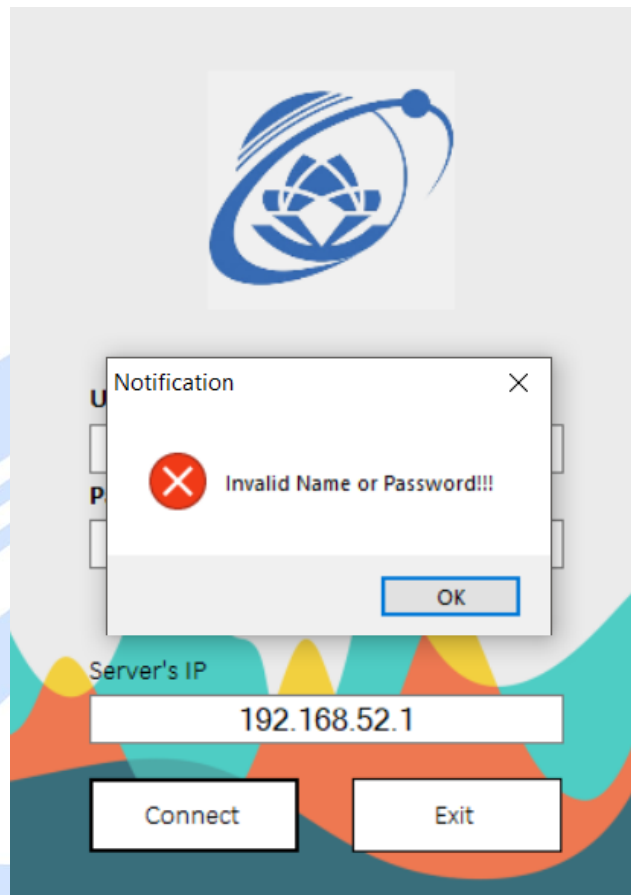
Đây là cốt lõi cho nhiều tác vụ đối với database của chương trình.

### Đăng nhập

The login interface consists of a central gray panel with a logo at the top. Below the logo are three input fields: 'User Name', 'Password', and 'Server's IP'. The 'Password' field is split into two parts. At the bottom of the panel are two buttons: 'Connect' and 'Exit'. The background of the application window features a colorful abstract design with blue, green, yellow, and orange shapes.

**Hình 8: Giao diện màn hình đăng nhập**

Với nguồn dữ liệu đã được tạo, việc đăng nhập được thực hiện bằng cách kiểm tra xem đã tồn tại chưa User Name và Hash Password (Password nhập vào sẽ được mã hóa MD5 và kiểm tra với Hash Password trong database). Nếu tồn tại đồng thời một cặp User Name và Hash Password trùng khớp thì đăng nhập hợp lệ. Ngược lại, nếu nhập sai tên đăng nhập hoặc mật khẩu chương trình sẽ báo lỗi và hiện thông báo như sau:



Hình 9. Lỗi đăng nhập

Đoạn code kiểm tra user có hợp lệ hay không:

```
private bool IsValidUser(string userName, string password)
{
    //hash MD5
    MD5 mh = MD5.Create();
    byte[] inputBytes = System.Text.Encoding.ASCII.GetBytes(tbPassword.Text); //plaintext to bytes

    byte[] hash = mh.ComputeHash(inputBytes);

    StringBuilder sb = new StringBuilder();

    for (int i = 0; i < hash.Length; i++)
    {
        sb.Append(hash[i].ToString("X2")); //X2 in hoa, x2 in thuong
    }

    //check
    UserDBDataContext context = new UserDBDataContext();
    var q = from p in context.USERS
            where p.UserName == tbName.Text
            && p.HashPassword == sb.ToString()
            select p;

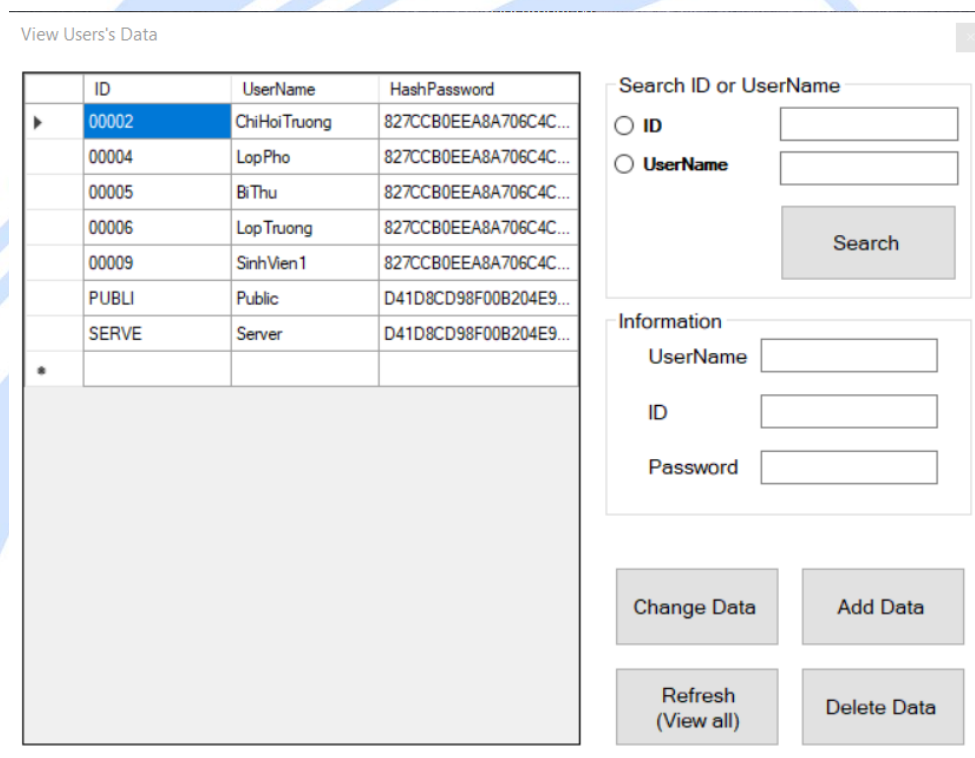
    if (q.Any())
    {
        return true;
    }
    else
    {
        return false;
    }
}
```

Hình 10: Xử lý logic đăng nhập

### Các thao tác truy xuất thông tin user

Việc truy xuất thông tin, bao gồm **Thêm, Xóa, Sửa, Tìm Kiếm** được thực hiện trên chương trình chạy Server, tức là người thực hiện sẽ ở quyền Admin. Để mở form UserDatabase, ta click vào nút “Users Database” ở góc phải dưới giao diện Server như **Hình 3**.

Ta được form như sau:



ID	UserName	HashPassword
00002	ChiHoiTruong	827CCB0EEA8A706C4C...
00004	LopPho	827CCB0EEA8A706C4C...
00005	BiThu	827CCB0EEA8A706C4C...
00006	LopTruong	827CCB0EEA8A706C4C...
00009	SinhVien1	827CCB0EEA8A706C4C...
PUBLI	Public	D41D8CD98F00B204E9...
SERVE	Server	D41D8CD98F00B204E9...

Search ID or UserName

☐ ID

☐ UserName

Search

Information

UserName

ID

Password

Change Data Add Data

Refresh (View all) Delete Data

**Hình 11: Form Users's Data**

#### **a. Thêm**

Để thêm thông tin, ta nhập vào đầy đủ các thông tin vào khung “Information”, sau đó nhấn “Add Data”. Thông tin được thêm thành công chương trình sẽ hiện lên thông báo như sau:

View Users's Data

ID	UserName	HashPassword
00004	LopPho	827CCB0EEA8A...
00005	BiThu	827CCB0EEA8A...
00006	LopTruong	827CCB0EEA8A...
00009	ChiHoiTruong	827CCB0EEA8A...
PUBLI	Public	D41D8CD98F00...
SERVE	Server	D41D8CD98F00...

Search ID or UserName

☐ ID

☐ UserName

Search

Information

UserName

ID

Password

Change Data

Add Data

Refresh (View all)

Delete Data

Success!

OK

Hình 12.1: Thêm thành công

Nhưng việc thêm không thể không được kiểm tra vì khi thêm có khả năng các dữ liệu trùng lặp, ở đây ID không thể được trùng lặp cũng như User Name là phải là độc nhất. Do đó, nếu thêm vào các thông tin bị trùng, chương trình sẽ hiện thông báo không thành công.

ID	UserName	HashPassword
00002	ChiHoiTruong1	827CCB0EEA8A706C4C...
00004	LopPho	827CCB0EEA8A706C4C...
00005	BiThu	827CCB0EEA8A706C4C...
00006	LopTruong	827CCB0EEA8A706C4C...
00009	ChiHoiTruong	827CCB0EEA8A706C4C...
PUBLI	Public	D41D8CD98F00B204E9...
SERVE	Server	D41D8CD98F00B204E9...

Search ID or UserName

☐ ID

☒ UserName

Search

Information

UserName

ID

Password

Change Data

Add Data

Refresh (View all)

Delete Data

Duplicated data!

OK



### Hình 12.2.1. Thêm không thành công

View Users's Data

ID	UserName	HashPassword
00002	ChiHoiTruong1	827CCB0EEA8A706C4C...
00004	LopPho	827CCB0EEA8A706C4C...
00005	BiThu	827CCB0EEA8A706C4C...
00006	LopTruong	827CCB0EEA8A706C4C...
00009	ChiHoiTruong	827CCB0EEA8A706C4C...
PUBLI	Public	D41D8CD98F00B204E9...
SERVE	Server	D41D8CD98F00B204E9...

Search ID or UserName

☐ ID

☒ UserName

Search

Information

UserName

ID

Password

Change Data Add Data

Refresh (View all) Delete Data

Duplicated data!

OK

### Hình 12.2.2. Thêm không thành công

#### b. Xóa

Để thực hiện thao tác xóa, ta chọn một hàng (item), sau đó nhấn "Delete". ID của hàng được chọn sẽ được tham chiếu tới và được remove khỏi datasource.

View Users's Data

ID	UserName	HashPassword
00002	ChiHoiTruong1	827CCB0EEA8A706C4C...
00004	LopPho	827CCB0EEA8A706C4C...
00005	BiThu	827CCB0EEA8A706C4C...
00006	LopTruong	827CCB0EEA8A706C4C...
00009	ChiHoiTruong	827CCB0EEA8A706C4C...
PUBLI	Public	D41D8CD98F00B204E9...
SERVE	Server	D41D8CD98F00B204E9...

Search ID or UserName

☐ ID

☒ UserName

Search

Information

UserName

ID

Password

Change Data Add Data

Refresh (View all) Delete Data

Caution!

Delete selected data?!

Yes No

### Hình 13.1: Thông báo trước khi Xóa

View Users's Data

ID	UserName	HashPassword
00004	LopPho	827CCB0EEA8A706C4C...
00005	BiThu	827CCB0EEA8A706C4C...
00006	LopTruong	827CCB0EEA8A706C4C...
00009	ChiHoiTruong	827CCB0EEA8A706C4C...
PUBLI	Public	D41D8CD98F00B204E9...
SERVE	Server	D41D8CD98F00B204E9...

Search ID or UserName

☐ ID

☒ UserName

Search

Information

UserName

ID

Password

Change Data Add Data

Refresh (View all) Delete Data

Hình 13.2: Sau khi xóa thành công

c. *Sửa*

Với thao tác Sửa, ta sửa trực tiếp thông tin trên một hàng trong bảng. Có một lưu ý nhỏ là ta không thể sửa trường ID vì nó là khóa chính. Tương tự như thao tác Thêm, các dữ liệu không thể trùng nhau sẽ được kiểm tra trùng lặp khi sửa.

View Users's Data

ID	UserName	HashPassword
00002	ChiHoiTruong1	827CCB0EEA8A706C4C...
00004	LopPho	827CCB0EEA8A706C4C...
00005	BiThu	827CCB0EEA8A706C4C...
00006	LopTruong	827CCB0EEA8A706C4C...
00009	SinhVien1	827CCB0EEA8A706C4C...
PUBLI	Public	D41D8CD98F00B204E9...
SERVE	Server	D41D8CD98F00B204E9...

Search ID or UserName

☐ ID

☒ UserName

Search

Information

UserName

ID

Password

Change Data Add Data

Refresh (View all) Delete Data

Hình 14.1.1 Thao tác sửa

View Users's Data

ID	UserName	HashPassword
00002	ChiHoiTruong1	827CCB0EEA8A706C4C...
00004	LopPho	827CCB0EEA8A706C4C...
00005	BiThu	827CCB0EEA8A706C4C...
00006	LopTruong	827CCB0EEA8A706C4C...
00009	SinhVien1	827CCB0EEA8A706C4C...
PUBLI	Public	D41D8CD98F00B204E9...
SERVE	Server	D41D8CD98F00B204E9...

Search ID or UserName

☐ ID

☒ UserName

Search

Information

UserName

ID

Password

Change Data

Add Data

Refresh (View all)

Delete Data

Success!

OK

Hình 14.1.2 Sửa thành công

View Users's Data

ID	UserName	HashPassword
00002	ChiHoiTruong	827CCB0EEA8A706C4C...
00004	LopPho	827CCB0EEA8A706C4C...
00005	BiThu	827CCB0EEA8A706C4C...
00006	LopTruong	827CCB0EEA8A706C4C...
00009	ChiHoiTruong	827CCB0EEA8A706C4C...
PUBLI	Public	D41D8CD98F00B204E9...
SERVE	Server	D41D8CD98F00B204E9...

Search ID or UserName

☐ ID

☒ UserName

Search

Information

UserName

ID

Password

Change Data

Add Data

Refresh (View all)

Delete Data

Hình 14.2.1 Sửa bị lặp

View Users's Data

ID	UserName	HashPassword
00002	ChiHoiTruong	827CCB0EEA8A706C4C...
00004	LopPho	827CCB0EEA8A706C4C...
00005	BiThu	827CCB0EEA8A706C4C...
00006	LopTruong	827CCB0EEA8A706C4C...
00009	ChiHoiTruong	827CCB0EEA8A706C4C...
PUBLI	Public	D41D8CD98F00B204E9...
SERVE	Server	D41D8CD98F00B204E9...

Search ID or UserName

☐ ID

☒ UserName

Search

Information

UserName

ID

Password

Change Data

Add Data

Refresh (View all)

Delete Data

Fail!

OK

Hình 14.2.2 Sửa không thành công

#### d. Tìm kiếm

Ở thao tác này, chương trình có 2 lựa chọn là tìm kiếm bằng ID hoặc User Name. Muốn tìm kiếm bằng một trong 2 trường trên thì ta click vào radio button bên trái tương ứng với lựa chọn mình muốn.

View Users's Data

ID	UserName	HashPassword
00002	ChiHoiTruong	827CCB0EEA8A706C4C...

Search ID or UserName

☒ ID

☐ UserName

Search

Information

UserName

ID

Password

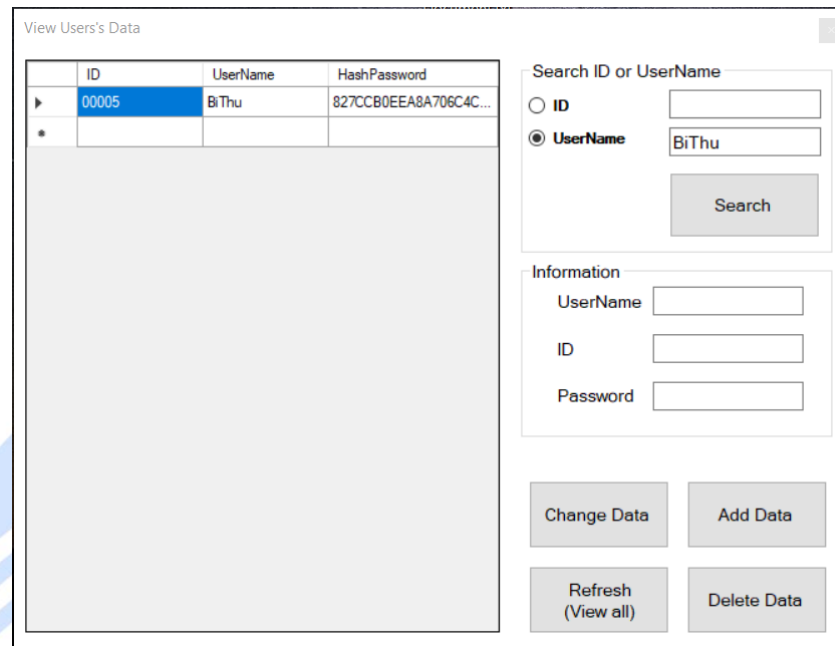
Change Data

Add Data

Refresh (View all)

Delete Data

Hình 15.1: Tìm kiếm bằng ID



ID	UserName	HashPassword
00005	BiThu	827CCB0EEA8A706C4C...
...		

Search ID or UserName

☐ ID

☐ **UserName**

Search

Information

UserName

ID

Password

Change Data Add Data

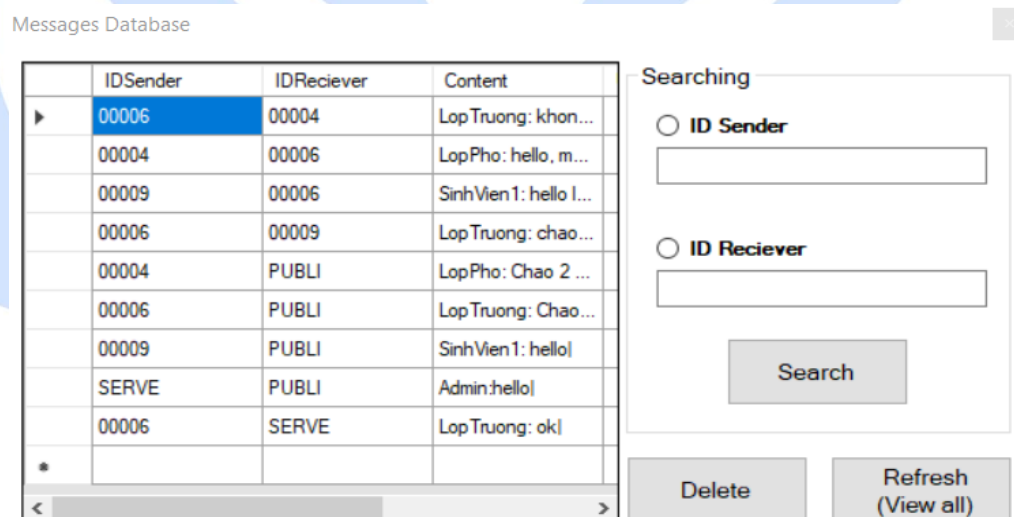
Refresh (View all) Delete Data

Hình 15.2 Tìm kiếm bằng Username

Sau khi tìm kiếm xong, ta chỉ cần nhấn nút “Refresh”, listview sẽ được load lại và trang sẽ về trạng thái đầy đủ thông tin của User.

### Lưu tin nhắn vào Database

Trước tiên, để quản lý tin nhắn một cách trực quan hơn, chương trình có 1 form để hiển thị các dữ liệu này.



IDSender	IDReceiver	Content
00006	00004	LopTruong: khon...
00004	00006	LopPho: hello, m...
00009	00006	SinhVien1: hello l...
00006	00009	LopTruong: chao...
00004	PUBLI	LopPho: Chao 2 ...
00006	PUBLI	LopTruong: Chao...
00009	PUBLI	SinhVien1: hello!
SERVE	PUBLI	Admin:hello!
00006	SERVE	LopTruong: ok!
...		

Searching

☐ ID Sender

☐ **ID Receiver**

Search

Delete Refresh (View all)

Hình 16: Message Database

Ở đây chúng ta cũng có chức năng Xóa và Tìm Kiếm tương tự như các thao tác ở dữ liệu người dùng. Đáng chú ý là việc lưu tin nhắn vào cơ sở dữ liệu.



Để lưu tin nhắn, vì tất cả các tin nhắn đều được gửi về Server, cho nên ta chỉ cần cho một vài dòng thực hiện nhiệm vụ lưu khi Server nhận được một gói tin nào đó. Vấn đề là làm sao để phân biệt đâu là tin nhắn của ai cho ai, public hay private.

Để thực hiện việc đó, sinh viên sử dụng những Label cố định để xác định đâu là tin nhắn public, và đâu là tin nhắn private; ID của người gửi và người nhận sẽ là một phần của dữ liệu được gửi đi, nên sau khi nhận được ID, ta chỉ cần kiểm tra xem nó có tồn tại trong cơ sở dữ liệu chưa, nếu đã hợp lệ thì tự động tạo một hàng mới lưu tin nhắn vào.

## **2. Về các chức năng chưa hiện thực thành công**

Trong phạm vi của đề tài, sinh viên có đề ra một số chức năng. Nhưng vì khả năng và thời gian còn hạn chế, sinh viên chưa thể hiện thực thành công các tính năng này. Đó là các tính năng:

- Đăng ký từ phía user.
- Gửi hình ảnh và icon.
- Tải lên lịch sử tin nhắn từ phía user và server.

Tuy vậy, nhìn chung khả năng phát triển của các tính năng này là rất cao và có tính khả thi.

## **V. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN**

Như vậy, qua kiến thức đã được tìm hiểu và nghiên cứu trong khuôn khổ môn học, sinh viên đã xây dựng được một hệ thống chat khá hoàn thiện đảm bảo các tính năng yêu cầu cơ bản. Sinh viên nắm rõ được phương thức hoạt động của lập trình socket, sử dụng nó để hiện thực đề tài. Ngoài ra, sinh viên cũng tìm hiểu thêm các thư viện, các tính năng ngoài phạm vi môn để có thể xây dựng nên hệ thống này.

Về hướng phát triển, tùy theo các tính năng mà người dùng (end user) yêu cầu, hệ thống này có thể dễ dàng mở rộng được. Các tính năng chưa hiện thực được sẽ có thể mở rộng được. Mặc dù vậy, cần hiểu rõ phương thức, các luồng tín hiệu, thông tin mà hệ thống sử dụng để mở rộng nó.