ĐẠI HỌC ĐÀ NẴNG

TRƯỜNG ĐẠI HỌC BÁCH KHOA

KHOA CÔNG NGHỆ THÔNG TIN

Tel. (84-511) 3736949, Fax. (84-511) 3842771

Website: itf.dut.udn.vn, E-mail: cntt@dut.udn.vn



BÁO CÁO MÔN MẠNG MÁY TÍNH

**ĐỀ TÀI :**

**Xây dựng ứng dụng chat multiple client thông qua server với socket sử dụng giao thức TCP**

1. **GIẢNG VIÊN HƯỚNG DẪN : PGS.TS. Nguyễn Tấn Khôi**

1. **SINH VIÊN THỰC HIỆN:**

**Lý Thanh Hải MSSV : 102190061**

**Lê Thị Bình MSSV : 102190053**

**Nguyễn Hoàng Trường MSSV : 102190096**

**Nhóm: 19N10**

**Lớp : 19TCLC\_DT2**

Đà Nẵng, 12/12/2021

LỜI CẢM ƠN

Trước hết, chúng em chân thành cảm ơn các thầy cô của Khoa Công Nghệ Thông Tin trường đại học Bách Khoa Đà Nẵng đặc biệt là thầy Nguyễn Văn Nguyên đã tạo điều kiện thuận lợi cho chúng em trong suốt quá trình học.

* Chúng em xin chân thành cảm ơn tất cả quý thầy cô đã nhiệt tình giảng dạy.
* Chúng em cũng hết lòng biết ơn sự quan tâm và ủng hộ của gia đình và bạn bè. Đó chính là nguồn động viên tinh thần rất lớn để chúng em theo đuổi và hoàn thành đồ án này.
* Đặc biệt, chúng em vô cùng tri ân sự hướng dẫn tận tình và theo dõi sát sao đầy tinh thần trách nhiệm cùng lòng thương mến của thầy Nguyễn Văn Nguyên trong suốt quá trình em thực hiện đồ án PBL4 lần này.
* Cuối cùng em muốn gởi lời cảm ơn đến toàn bộ quý thầy cô của khoa Công Nghệ Thông Tin, những người có vai trò rất lớn trong suốt quá trình chúng em thực hiện đồ án.

Trong quá trình thực hiện đồ án, nhận thấy chúng em đã cố gắng hết sức nhưng vì kiến thức vẫn còn hẹn hẹp nên vẫn còn nhiều thiếu sót, mong thầy cô bổ sung để đồ án được hoàn thiện hơn.

Em xin chân thành cảm ơn!

**Lý Thanh Hải**

LỜI CAM ĐOAN

Chúng em xin cam đoan :

1. Nội dung trong đồ án này là do chúng em thực hiện dưới sự hướng dẫn trực tiếp của thầy Nguyễn Văn Nguyên.
2. Các tham khảo dùng trong đồ án đều được trích dẫn rõ ràng tên tác giả, tên công trình, thời gian, địa điểm công bố.
3. Nếu có những sao chép không hợp lệ, vi phạm, chúng em xin chịu hoàn toàn trách nhiệm.

**Lý Thanh Hải**

**NHẬN XÉT CỦA NGƯỜI HƯỚNG DẪN**

...................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................

NHẬN XÉT CỦA NGƯỜI PHẢN BIỆN

............................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................

MỤC LỤC

[CHƯƠNG 1: CƠ SỞ LÝ THUYẾT 15](#_Toc88894994)

[1.1. Giao thức TCP/IP 15](#_Toc88894995)

[1.2. Giao thức IP(Internet Protocol - Giao thức Liên mạng) 15](#_Toc88894996)

[1.3. Giao thức TCP(Transmission Control Protocol - Giao thức điều khiển truyền vận) . 17](#_Toc88894997)

[1.4. Mô hình client-server 18](#_Toc88894998)

[1.5. Socket trong C#. 20](#_Toc88894999)

[1.5.1. Socket là gì. 20](#_Toc88895000)

[1.5.2. Cơ chế hoạt động của socket. 20](#_Toc88895001)

[1.5.3. Mô hình truyền tin socket. 21](#_Toc88895002)

[1.6. KẾT CHƯƠNG 22](#_Toc88895003)

[CHƯƠNG 2 :PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG 23](#_Toc88895004)

[2.1. Phân tích và thiết kế hướng đối tượng. 23](#_Toc88895005)

[2.1.1. Sơ đồ usecase. 23](#_Toc88895006)

[2.1.2. Sơ đồ lớp. 24](#_Toc88895007)

[2.1.3. Cơ sở dữ liệu. 24](#_Toc88895008)

[2.1.4. Sơ đồ hoạt động. 25](#_Toc88895009)

[2.1.5. Sơ đồ tuần tự. 26](#_Toc88895010)

[2.2. Phân tích yêu cầu. 26](#_Toc88895011)

[2.3. KẾT CHƯƠNG 27](#_Toc88895012)

[CHƯƠNG 3: CHỨC NĂNG VÀ KẾT QUẢ 28](#_Toc88895013)

[3.1. MÔI TRƯỜNG CÀI ĐẶT 28](#_Toc88895014)

[3.2. CHỨC NĂNG VÀ KẾT QUẢ CỦA CHƯƠNG TRÌNH. 28](#_Toc88895015)

[3.2.1 Đăng nhập. 28](#_Toc88895016)

[3.2.2 Đăng ký. 30](#_Toc88895017)

[3.2.3 Đăng xuất. 32](#_Toc88895018)

[3.2.4 Tìm kiếm người dùng. 33](#_Toc88895019)

[3.2.5 Chức năng tạo nhóm. 34](#_Toc88895020)

[3.2.6 Chức năng nhắn tin. 36](#_Toc88895021)

[3.2.7 Chức năng gọi video. 38](#_Toc88895022)

[3.3. KẾT CHƯƠNG 40](#_Toc88895023)

**DANH SÁCH HÌNH ẢNH**

Hình 1.1.1. Sơ đồ TCP/IP.

Hình 1.2.1. Cấu trúc IP header.

Hình 1.3.1. Cấu trúc gói tin TCP.

Hình 1.4.1. Mô hình client-server.

Hình 1.5.1. Mô hình truyền tin của socket.

Hình 2.1.1.1. Sơ đồ usecase.

Hình 2.1.2.1. Sơ đồ lớp.

Hình 2.1.3.1. Cơ sở dữ liệu.

Hình 2.1.4.1. Sơ đồ hoạt động.

Hình 2.1.5.1. Sơ đồ tuần tự.

Hình 3.2.1.1. Giao diện đăng nhập.

Hình 3.2.2.1. Giao diện đăng ký.

Hình 3.2.4.1. Chức năng tìm kiếm người dùng.

Hình 3.2.5.1. Chức năng tạo nhóm.

Hình 3.2.6.1. Chức năng nhắn tin.

Hình 3.2.7.1. Chức năng chat video với người gọi.

Hình 3.2.7.2. Chức năng chat video với người nhận.

DANH SÁCH TỪ VIẾT TẮT

|  |  |
| --- | --- |
| **Từ viết tắt** | **Diễn giải** |
| IP | Internet Protocol |
| TCP | Transmission Control Protocol |
| MD5 | Message-Degist Algorithm 5 |
| SHA | Secure Hash Algorithm |
| API | Application Programming Interface |
| URL | Uniform Resource Locator |

MỞ ĐẦU

**1.**  **Tổng quan về đề tài**

Chúng ta đang sống trong thời đại 4.0, sự phát triển nhanh chóng của máy móc và thiết bị. Vậy nên con người có rất nhiều việc phải làm, họ rất bận rộn. Và nhu cầu giao tiếp với nhau càng nhiều để trao đổi về công việc của họ. Nhưng có lẽ chính vì chúng ta đang sống ở thời đại 4.0, nên chúng em nghĩ việc giao tiếp online cực kì quan trọng, giúp mọi người tiết kiệm được thời gian, và dễ dàng trao đổi với nhau hơn. Vì vậy chúng em đã quyết định tạo ra 1 ứng dụng nhỏ, có đầy đủ tính năng chat, call, call video, gửi hình ảnh, gửi file, để giúp mọi người có thể liên lạc tốt nhất. Chúng em nghĩ đề tài chúng em làm lần này rất quan trọng so với thời đại ngày nay, rất giúp ích cho mọi người. Tuy nhiên vẫn còn 1 số lỗi nhỏ như kết nối với nhau lâu, hoặc call với nhau vẫn còn hiện tượng giật lag, vì vậy chúng em sẽ cố gắng hết sức để có thể hoàn thành đề tài 1 cách tốt nhất.

**2. Mục đích và ý nghĩa của đề tài**

***2.1. Mục đích***

Giúp mọi người dễ dàng giao tiếp với nhau mà vẫn tiết kiệm thời gian, công sức, nhất là trong thời đại 4.0 ngày nay.

***2.2. Ý nghĩa***

Mang ý nghĩa quan trọng, giúp mọi người kết nối, gắn bó với nhau 1 cách thuận lợi, tạo ra không ít lợi ích cho mọi người.

**3. Phương pháp thực hiện**

Ngôn ngữ lập trình : C#.

Chương trình soạn thảo và build :  Visual studio.

Sử dụng Giao thức TCP/IP.

Sử dụng Socket trong C#.

**4. Bố cục của đồ án**

Đồ án bao gồm các nội dung sau:

1. Mở đầu.
2. Nội dung.

Chương 1: Trình bày và mô tả sơ lược về đồ án, cơ sở lý thuyết.

Chương 2: Trình bày và phân tích thiết kế hệ thống.

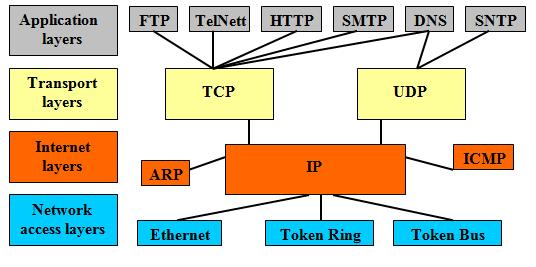
Chương 3: Trình bày và đánh giá, tổng kết về sản phẩm đồ án.

1. Kết luận và hướng phát triển.

# CHƯƠNG 1: CƠ SỞ LÝ THUYẾT

## Giao thức TCP/IP

TCP/IP là tên chung cho một tập hợp hơn 100 giao thức được sử dụng để kết nối các máy tính vào mạng, trong đó hai giao thức chính là TCP (Transmission Control Protocol) và IP (Internet Protocol). Trong phạm vi Internet, thông tin không được truyền tải như một dòng riêng biệt từ máy tính này tới máy tính khác. Thay vào đó, dữ liệu được chia thành những gói nhỏ gọi là packet. Các packet này được gửi trên mạng máy tính. Công việc của IP là chuyển chúng đến các máy tính ở xa. Tại trạm cuối, TCP nhận các packet và kiểm tra lỗi. Nếu một lỗi xuất hiện, TCP yêu cầu gói riêng biệt đó phải được gửi lại. Chỉ khi tất cả các packet đã nhận được là đúng, TCP sẽ sử dụng số thứ tự để tạo lại thông tin ban đầu.

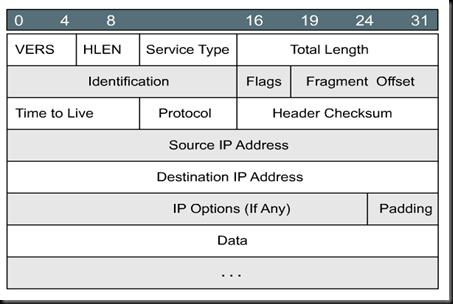


Hình 1.1.1. Sơ đồ TCP/IP.

## Giao thức IP(Internet Protocol - Giao thức Liên mạng)

Là một giao thức hướng dữ liệu được sử dụng bởi các máy chủ nguồn và đích để truyền dữ liệu trong một liên mạng chuyển mạch gói. Dữ liệu trong một liên mạng IP được gửi theo các khối được gọi là các gói (packet hoặc datagram). Cụ thể, IP không cần thiết lập các đường truyền trước khi một máy chủ gửi các gói tin cho một máy khác mà trước đó nó chưa từng liên lạc với. Giao thức IP cung cấp một dịch vụ gửi dữ liệu không đảm bảo (còn gọi là cố gắng cao nhất), nghĩa là nó hầu như không đảm bảo gì về gói dữ liệu.

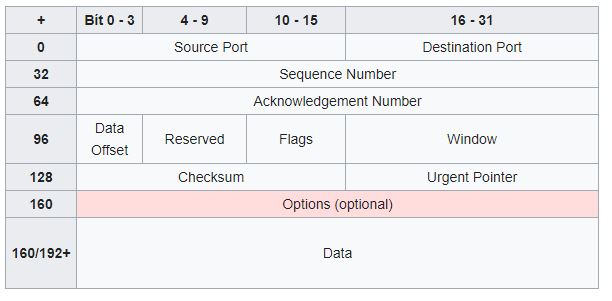
Gói dữ liệu có thể đến nơi mà không còn nguyên vẹn, nó có thể đến không theo thứ tự (so với các gói khác được gửi giữa hai máy nguồn và đích đó), nó có thể bị trùng lặp hoặc bị mất hoàn toàn. Nếu một phần mềm ứng dụng cần được bảo đảm, nó có thể được cung cấp từ nơi khác, thường từ các giao thức giao vận nằm phía trên IP. Các thiết bị định tuyến liên mạng chuyển tiếp các gói tin IP qua các mạng tầng liên kết dữ liệu được kết nối với nhau. Việc không có đảm bảo về gửi dữ liệu có nghĩa rằng các chuyển mạch gói có thiết kế đơn giản hơn. (lưu ý rằng nếu mạng bỏ gói tin, làm đổi thứ tự hoặc làm hỏng nhiều gói tin, người dùng sẽ thấy hoạt động mạng trở nên kém đi. Hầu hết các thành phần của mạng đều cố gắng tránh để xảy ra tình trạng đó. Đó là lý do giao thức này còn được gọi là cố gắng cao nhất. Tuy nhiên, khi lỗi xảy ra không thương xuyên sẽ không có hiệu quả đủ xấu đến mức người dùng nhận thấy được.) Giao thức IP rất thông dụng trong mạng Internet công cộng ngày nay. Giao thức tầng mạng thông dụng nhất ngày nay là IPv4; đây là giao thức IP  phiên bản 4. IPv6 được đề nghị sẽ kế tiếp IPv4: Internet đang hết dần địa chỉ IPv4, do IPv4 sử dụng 32 bit để đánh địa chỉ (tạo được khoảng 4 tỷ địa chỉ); IPv6 dùng địa chỉ 128 bit, cung cấp tối đa khoảng 3.4×1038 địa chỉ . Các  phiên bản từ 0 đến 3 hoặc bị hạn chế, hoặc không được sử dụng.



Hình 1.2.1. Cấu trúc IP header.

## Giao thức TCP(Transmission Control Protocol - Giao thức điều khiển truyền vận).

Là một trong các giao thức cốt lõi của bộ giao thức TCP/IP. Sử dụng TCP, các ứng dụng trên các máy chủ được nối mạng có thể tạo các "kết nối" với nhau, mà qua đó chúng có thể trao đổi dữ liệu hoặc các gói tin. Giao thức này đảm bảo chuyển giao dữ liệu tới nơi nhận một cách đáng tin cậy và đúng thứ tự. TCP còn phân biệt giữa dữ liệu của nhiều ứng dụng (chẳng hạn, dịch vụ Web và dịch vụ thư điện tử) đồng thời chạy trên cùng một máy chủ. TCP hỗ trợ nhiều giao thức ứng dụng phổ biến nhất trên Internet và các ứng dụng kết quả, trong đó có WWW, thư điện tử và Secure Shell. Trong bộ giao thức TCP/IP, TCP là tầng trung gian giữa giao thức IP  bên dưới và một ứng dụng bên trên. Các ứng dụng thường cần các kết nối đáng tin cậy kiểu đường ống để liên lạc với nhau, trong khi đó, giao thức IP không cung cấp những dòng kiểu đó, mà chỉ cung cấp dịch vụ chuyển gói tin không đáng tin cậy. TCP làm nhiệm vụ của tầng giao vận trong mô hình OSI đơn giản của các mạng máy tính. Các ứng dụng gửi các dòng gồm các byte 8- bit tới TCP để chuyển qua mạng. TCP phân chia dòng byte này thành các đoạn (segment) có kích thước thích hợp (thường được quyết định dựa theo kích thước của đơn vị truyền dẫn tối đa (MTU) của tầng liên kết dữ liệu của mạng mà máy tính đang nằm trong đó). Sau đó, TCP chuyển các gói tin thu được tới giao thức IP để gửi nó qua một liên mạng tới mô đun TCP tại máy tính đích. TCP kiểm tra để đảm bảo không có gói tin nào bị thất lạc bằng cách gán cho mỗi gói tin một "số thứ tự" (sequence number). Số thứ tự này còn được sử dụng để đảm bảo dữ liệu được trao cho ứng dụng đích theo đúng thứ tự. Mô đun TCP tại đầu kia gửi lại "tin báo nhận" (acknowledgement) cho các gói tin đã nhận được thành công; một "đồng hồ" (timer) tại nơi gửi sẽ báo time-out nếu không nhận được tin báo nhận trong khoảng thời gian bằng một round-trip time (RTT), và dữ liệu (được coi là bị thất lạc) sẽ được gửi lại. TCP sử dụng checksum (giá trị kiểm tra) để xem có byte nào bị hỏng trong quá trình truyền hay không; giá trị này được tính toán cho mỗi khối dữ liệu tại nơi gửi trước khi nó được gửi, và được kiểm tra tại nơi nhận.



Hình 1.3.1 Cấu trúc gói tin TCP.

## Mô hình client-server

Mô hình được phổ biến nhất và được chấp nhận rộng rãi trong các hệ thống phân tán là mô hình client/server. Trong mô hình này sẽ có một tập các tiến trình mà mỗi tiến trình đóng vai trò như là một trình quản lý tài nguyên cho một tập hợp các tài nguyên cho trước và một tập hợp các tiến trình client trong đó mỗi tiến trình thực hiện một tác vụ nào đó cần truy xuất tới tài nguyên phần cứng hoặc phần mềm dùng chung. Bản thân các trình quản lý tài nguyên cần phải truy xuất tới các tài nguyên dùng chung được quản lý bởi một tiến trình khác, vì vậy một số tiến trình vừa là tiến trình client vừa là tiến trình server. Các tiến trình phát ra các yêu cầu tới các server  bất kỳ khi nào chúng cần truy xuất tới một trong các tài nguyên của các server. Nếu yêu cầu là đúng đắn thì server sẽ thực hiện hành động được yêu cầu và gửi một đáp ứng trả lời tới tiến trình client.

Mô hình client/server cung cấp một cách tiếp cận tổng quát để chia sẻ tài nguyên trong các hệ thống phân tán. Mô hình này có thể đƣợc cài đặt bằng rất nhiều môi trường phần cứng và phần mềm khác nhau. Các máy tính được sử dụng để chạy các tiến trình client/server có nhiều kiểu khác nhau và không cần thiết phải phân biệt giữa chúng; cả tiến trình client và tiến trình server đều có thể chạy trên cùng một máy tính. Một tiến trình server có thể sử dụng dịch vụ của một server khác. Mô hình truyền tin client/server hướng tới việc cung cấp dịch vụ.

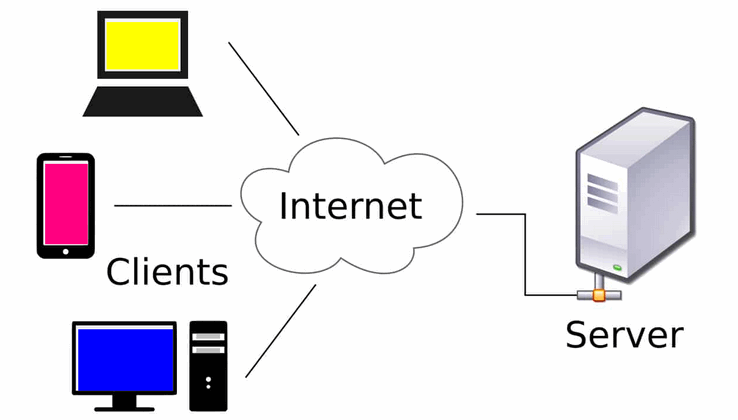
Quá trình trao đổi dữ liệu bao gồm:

1. Truyền một yêu cầu từ tiến trình client tới tiến trình server.

2. Yêu cầu được server xử lý.

3. Truyền đáp ứng cho client.

Mô hình truyền tin này liên quan đến việc truyền hai thông điệp và một dạng đồng bộ hóa cụ thể giữa client và server. Tiến trình server phải nhận thức được thông điệp được yêu cầu ở bước một ngay khi nó đến và hành động phát ra yêu cầu trong client phải được tạm dừng (bị phong tỏa) và buộc tiến trình client ở trạng thái chờ cho tớ khi nó nhận đƣợc đáp ứng do server gửi về ở bước ba. Mô hình client/server thường được cài đặt dựa trên các thao tác cơ bản là gửi (send) và nhận (receive).



*Hình 1.4.1. Mô hình client-server.*

Quá trình giao tiếp client và server có thể diễn ra theo một trong hai chế độ: bị phong tỏa (blocked) và không bị phong tỏa (non-blocked). Chế độ bị phong tỏa (blocked): Trong chế độ bị phong tỏa, khi tiến trình client hoặc server phát ra lệnh gửi dữ liệu (send), việc thực thi của tiến trình sẽ bị tạm ngừng cho tới khi tiến trình nhận phát ra lệnh nhận dữ liệu (receive). Tương tự đối với tiến trình nhận dữ liệu, nếu tiến trình nào đó (client hoặc server) phát ra lệnh nhận dữ liệu, mà tại thời điểm đó chưa có dữ liệu gửi tới thì việc thực thi của tiến trình cũng sẽ bị tạm ngừng cho tới khi có dữ liệu gửi tới. Chế độ không bị phong tỏa (non-blocked) Trong chế độ này, khi tiến trình client hay server phát ra lệnh gửi dữ liệu thực sự, việc thực thi của tiến trình vẫn được tiến hành mà không quan tâm đến việc có tiến trình nào phát ra lệnh nhận dữ liệu đó hay không. tương tự cho trường hợp nhận dữ liệu, khi tiến trình phát ra lệnh nhận dữ liệu, nó sẽ nhận dữ liệu hiện có, việc thực thi của tiến trình vẫn được tiến hành mà không quan tâm đến việc có tiến trình nào phát ra lệnh gửi dữ liệu tiếp theo hay không.

## Socket trong C#.

### Socket là gì.

Socket là giao diện lập trình ứng dụng mạng được dùng để truyền và nhận dữ liệu trên internet. Giữa hai chương trình chạy trên mạng cần có một liên kết giao tiếp hai chiều, hay còn gọi là two-way communication để kết nối 2 process trò chuyện với nhau. Điểm cuối (endpoint) của liên kết này được gọi là socket.

Một chức năng khác của socket là giúp các tầng **TCP** hoặc **TCP Layer** định danh ứng dụng mà dữ liệu sẽ được gửi tới thông qua sự ràng buộc với một cổng port (thể hiện là một con số cụ thể), từ đó tiến hành kết nối giữa client và server.

### Cơ chế hoạt động của socket.

Như đã đề cập trước đó, chức năng của socket là kết nối giữa client và server thông qua TCP/IP và UDP để truyền và nhận giữ liệu qua Internet. Giao diện lập trình ứng dụng mạng này chỉ có thể hoạt động khi đã có thông tin về thông số IP và số hiệu cổng của 2 ứng dụng cần trao đổi dữ liệu cho nhau.

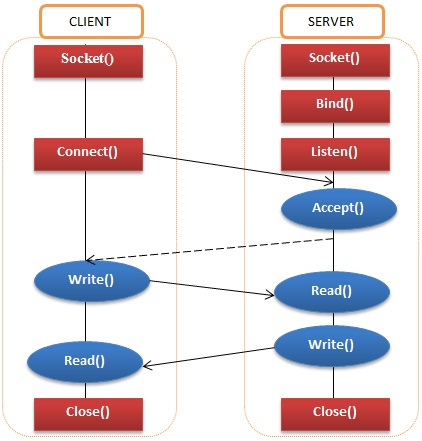
2 ứng dụng cần truyền thông tin phải đáp ứng điều kiện sau thì socket mới có thể hoạt động:

* 2 ứng dụng có thể nằm cùng trên một máy hoặc 2 máy khác nhau.
* Trong trường hợp 2 ứng dụng cùng nằm trên một máy, số hiệu cổng không được trùng nhau.

### Mô hình truyền tin socket.

Dựa trên giao thức TCP( Tranmission Control Protocol), stream socket thiết lập giao tiếp 2 chiều theo mô hình client và server. Được gọi là socket hướng kết nối.

Giao thức này đảm bảo dữ liệu được truyền đến nơi nhận một cách đáng tin cậy, đúng tuần tự nhờ vào cơ chế quản lý luồng lưu thông trên mạng và cơ chế chống tắc nghẽn.



*Hình 1.5.1. Mô hình truyền tin của socket.*

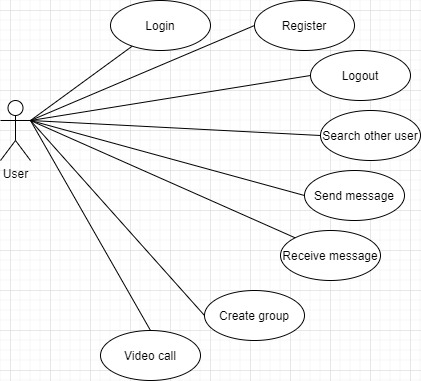
## KẾT CHƯƠNG

Chương này trình bày các lý thuyết căn bản để thực hiện đồ án.

# CHƯƠNG 2 :PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG

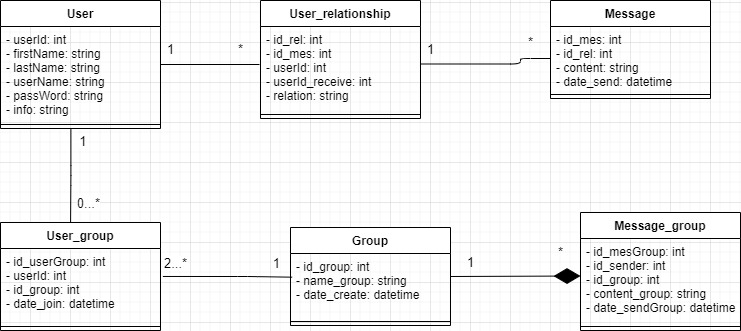
## 2.1. Phân tích và thiết kế hướng đối tượng.

### 2.1.1. Sơ đồ usecase.



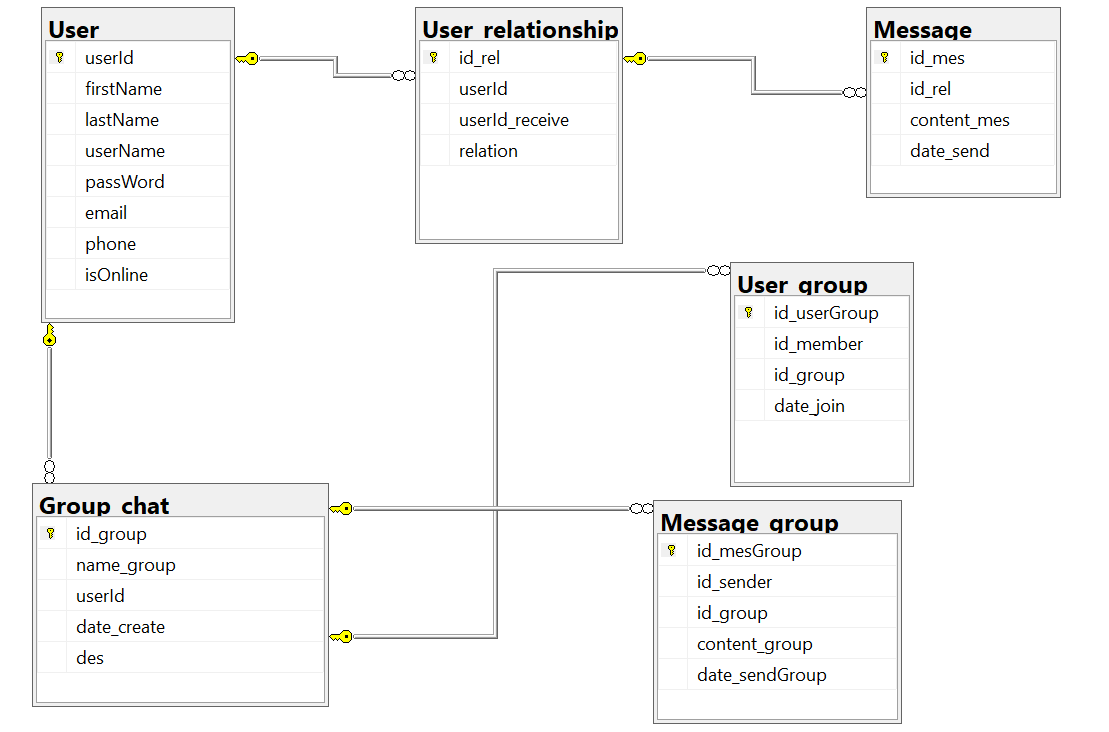
*Hình 2.1.1.1. Sơ đồ usecase.*

### 2.1.2. Sơ đồ lớp.



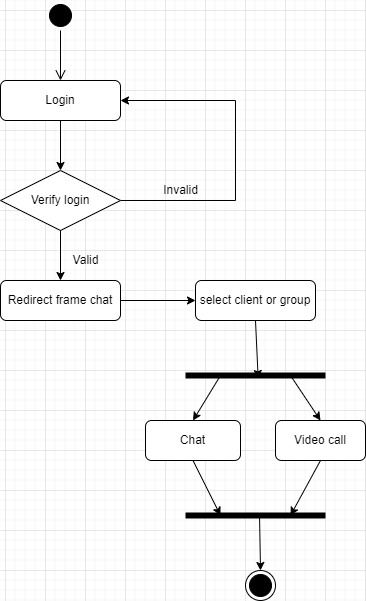
*Hình 2.1.2.1. Sơ đồ lớp.*

### 2.1.3. Cơ sở dữ liệu.



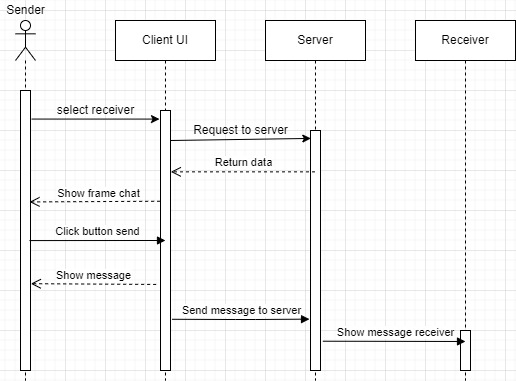
*Hình 2.1.3.1. Cơ sở dữ liệu.*

### 2.1.4. Sơ đồ hoạt động.



*Hình 2.1.4.1. Sơ đồ hoạt động.*

### 2.1.5. Sơ đồ tuần tự.



*Hình 2.1.5.1. Sơ đồ tuần tự.*

## 2.2. Phân tích yêu cầu.

- Yêu cầu của bài toán: Xây dựng mô hình Client – Server ứng dụng CHAT sử dụng socket và giao thức TCP/IP.

- Xây dựng chương trình Server:

 Tạo một TCP Socket và gắn vào một cổng.

Xây dựng một chương trình Server đa luồng để cho phép nhiều Client kết nối tới Server cùng lúc.

Mỗi luồng đảm nhận liên lạc với một Client riêng biệt. 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 client đăng nhập vào hệ thống thì Server sẽ thêm phiên làm việc cho client đó.

Khi một client đăng xuất ra khỏi hệ thống thì Server sẽ đóng kết nối và xóa client đó

Đóng kết nối.

- Xây dựng chương trình Client:

Tạo một TCP client với địa chỉ IP và số cổng mà Server đang chạy.

Thiết lập kết nối tới Server.

Trao đổi dữ liệu(nhận/gửi) với Server.

Gửi/ nhận dữ liệu tới tất cả mọi người có trong phòng chat.

Đóng kết nối.

Thiết kế giao diện hiển thị khung chat phía Client.

- Sử dụng giao thức TCP/IP, Socket.

- Yêu cầu chức năng: Đáp ứng các chức năng cơ bản đăng nhập, đăng xuất, đăng ký, nhập tin nhắn, gửi tin nhắn, gửi dữ liệu ảnh, file, tìm kiếm người dùng khác để nhắn tin, tạo nhóm để có thể chat nhóm, video call bao gồm chia sẽ camera và âm thanh.

## 2.3. KẾT CHƯƠNG

Chương này trình bày những phân tích thiết kế hướng đối tượng nhằm giúp cho hệ thống hoạt động và phát triển theo đúng yêu cầu đồng thời với những yêu cầu bài toán và xây dựng cho hệ thống.

# CHƯƠNG 3: CHỨC NĂNG VÀ KẾT QUẢ

## . MÔI TRƯỜNG CÀI ĐẶT

- Sử dụng Visual Studio

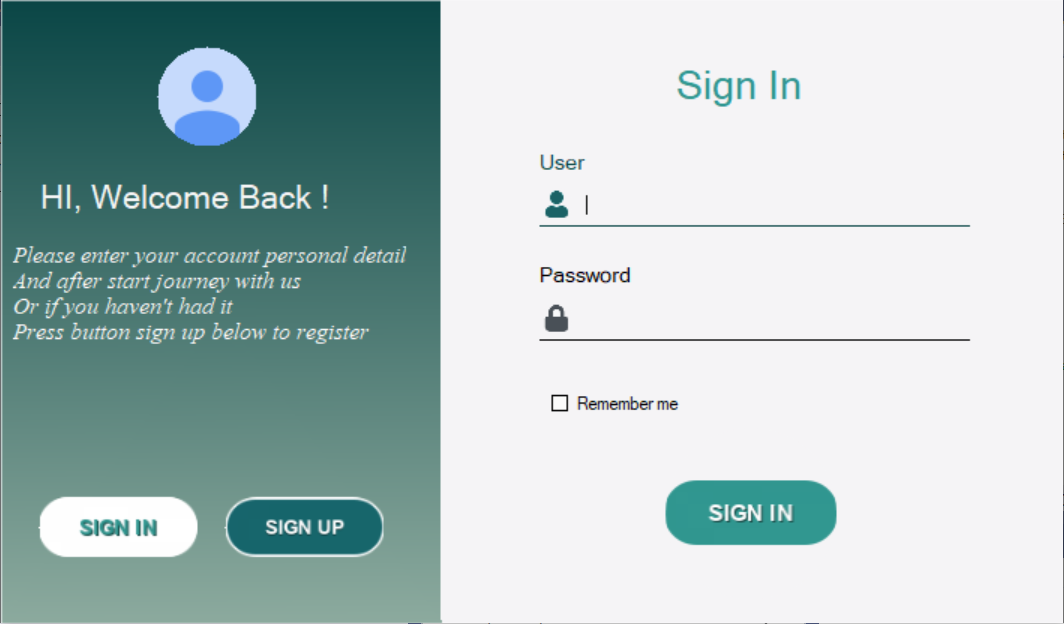
- Hệ điều hành window

|  |  |  |
| --- | --- | --- |
|  | Ưu điểm | Nhược điểm |
| Visual Studio | Dung lượng nhẹ, miễn phí.  Hỗ trợ đa ngôn ngữ lập trình, đa nền tảng, web, kho lưu trữ an toàn.  Tốc độ nhanh, nhỏ gọn và tiện lợi | Nếu tải bản Visual Studio Full Crack có thể làm máy tính nhiễm virus, nhiễm các chương trình độc hại |
| Hệ điều hành window | Độ phổ biến cao  Dễ dùng, dễ thích ứng  Dễ chuyển đổi giữa các hệ điều hành | Dễ bị tấn công bởi hacker  Dễ gặp lỗi sau mỗi bản cập nhật |

## . CHỨC NĂNG VÀ KẾT QUẢ CỦA CHƯƠNG TRÌNH.

### 3.2.1 Đăng nhập.

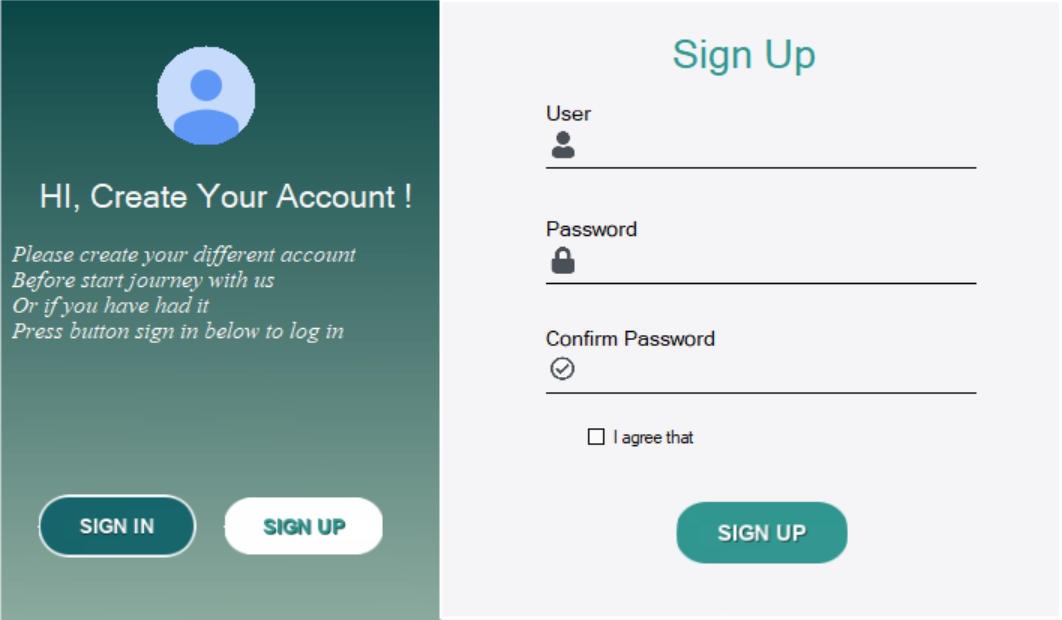
|  |  |  |  |
| --- | --- | --- | --- |
| **Giao diện** | Đăng nhập | | |
| **Mô tả** | Cho phép khách hàng đăng nhập vào hệ thống | | |
| **Truy cập** | Trang đăng nhập hiển thị khi khởi động chương trình | | |
| **Nội dung giao diện** | | | |
| **Thành phần** | **Kiểu** | **Dữ liệu** | **Mô tả** |
| Tài khoản | TextBox – String(100) |  | Trường dành cho khách hàng nhập tài khoản |
| Mật khẩu | TextBox – String(100) |  | Trường dành cho khách hàng nhập mật khẩu |
| Đăng nhập | Button |  | Đăng nhập vào hệ thống |
| **Các hành động trong giao diện** | | | |
| **Hành động** | **Mô tả** | **Thành công** | **Lỗi** |
| Đăng nhập | Khi admin kích nút đăng nhập, hệ thống sẽ kiểm tra Tài khoản với mật khẩu tương ứng có tồn tại trong hệ thống hay không, nếu có thì cho admin đăng nhập vào hệ thống, nếu không thì hiển thị thông báo “Tên đăng nhập hoặc mật khẩu không đúng!”.  Khi không nhập tài khoản hoặc mật khẩu thì hiển thị thông báo “Vui lòng nhập đủ thông tin”. | Vào màn chính của Tab hệ thống. | Hiện thông báo: “Tên đăng nhập hoặc mật khẩu không đúng!”  Hiện thông báo: “Vui lòng nhập đủ thông tin”. |

****

*Hình 3.2.1.1. Giao diện đăng nhập.*

### 3.2.2 Đăng ký.

|  |  |  |  |
| --- | --- | --- | --- |
| **Giao diện** | Đăng ký | | |
| **Mô tả** | Cho phép khách hàng đăng ký vào hệ thống | | |
| **Truy cập** | Trang đăng ký hiển thị khi nút đăng ký khi chưa có tài khoản được click | | |
| **Nội dung giao diện** | | | |
| **Thành phần** | **Kiểu** | **Dữ liệu** | **Mô tả** |
| Tài khoản | TextBox – String(100) |  | Trường dành cho khách hàng nhập tài khoản |
| Mật khẩu | TextBox – String(100) |  | Trường dành cho khách hàng nhập mật khẩu |
| Nhập lại mật khẩu | TextBox – String(100) |  | Trường dành cho khách hàng nhập lại mật khẩu |
| Đăng ký | Button |  | Đăng ký tài khoản |
| **Các hành động trong giao diện** | | | |
| **Hành động** | **Mô tả** | **Thành công** | **Lỗi** |
| Đăng ký | Khi khách hàng kích nút đăng ký, hệ thống sẽ kiểm tra tên tài khoản có trùng trong hệ thống chưa nếu chưa thì cho khách hàng đăng ký vào hệ thống, nếu rồi thì hiển thị thông báo “Tên đăng nhập đã tồn tại “ , nếu nhập không đủ số ký tự yêu cầu thì thông báo “Vui lòng nhập đủ số ký tự ” , nếu mật khẩu xác nhận không trùng mật khẩu thì hiển thị thông báo “Mật khẩu không trùng nhau”  Khi không nhập đủ thông tin trên form thì hiển thị thông báo “Vui lòng nhập đủ thông tin”. | Vào màn hình đăng nhập | Hiện thông báo: “Tài khoản đã tồn tại!” ,  “Số lượng ký tự không đủ ”  “Mật khẩu xác nhận không trùng mật khẩu ”  Hiện thông báo: “Vui lòng nhập đủ thông tin”. |

****

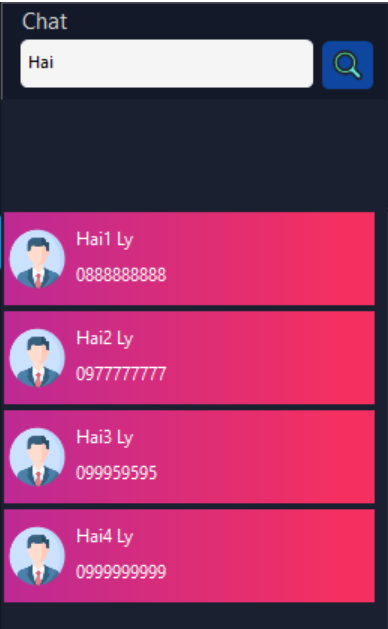
*Hình 3.2.2.1. Giao diện đăng ký.*

### Đăng xuất.

|  |  |  |  |
| --- | --- | --- | --- |
| **Giao diện** | Đăng xuất | | |
| **Mô tả** | Cho phép khách hàng đăng xuất | | |
| **Truy cập** | Từ màn hình chính khách chọn chức năng đăng xuất | | |
| **Các hành động trong giao diện** | | | |
| **Hành động** | **Mô tả** | **Thành công** | **Lỗi** |
| Đăng xuất | Khi khách hàng kích chức năng đăng xuất , khách hàng sẽ trở lại màn hình đăng nhập | Trả về màn hình đăng nhập | không |

### Tìm kiếm người dùng.

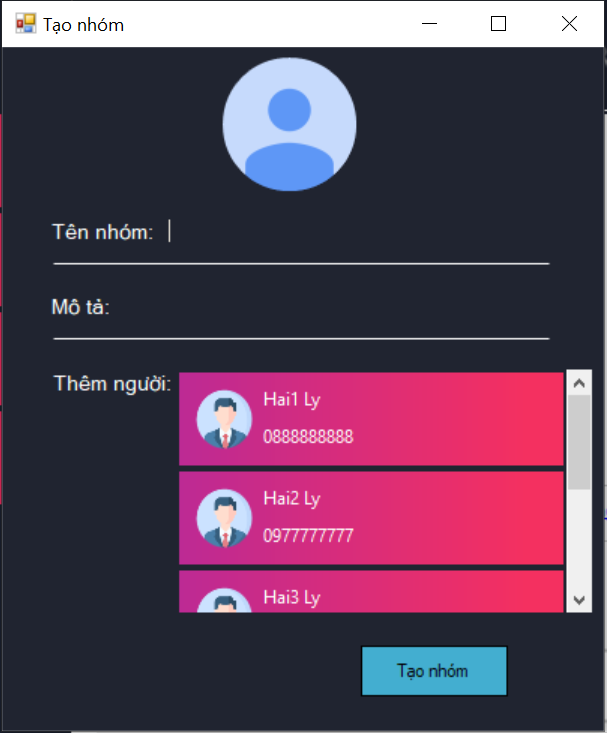
|  |  |  |  |
| --- | --- | --- | --- |
| **Giao diện** | Chính | | |
| **Mô tả** | Cho phép người dùng truy cập | | |
| **Truy cập** | Người dùng tìm kiếm trên thanh tìm kiếm | | |
| **Nội dung giao diện** | | | |
| **Thành phần** | **Kiểu** | **Dữ liệu** | **Mô tả** |
| Khung nhập thông tin | TextBox – String(100) |  | Trường dành cho khách hàng nhập thông tin tìm kiếm |
| Nút tìm kiếm | Button |  | Trường này cho phép khách hàng ấn nút để tìm kiếm |
| **Các hành động trong giao diện** | | | |
| **Hành động** | **Mô tả** | **Thành công** | **Lỗi** |
| Tìm kiếm | Khi người dùng nhập vào thông tin vào ô tìm kiếm sau đó nhấn nút tìm kiếm thì sẽ hiển thị những user có thông tin phù hợp | Hiện những người dùng. | Không hiện người dùng. |



*Hình 3.2.4.1. Chức năng tìm kiếm người dùng.*

### Chức năng tạo nhóm.

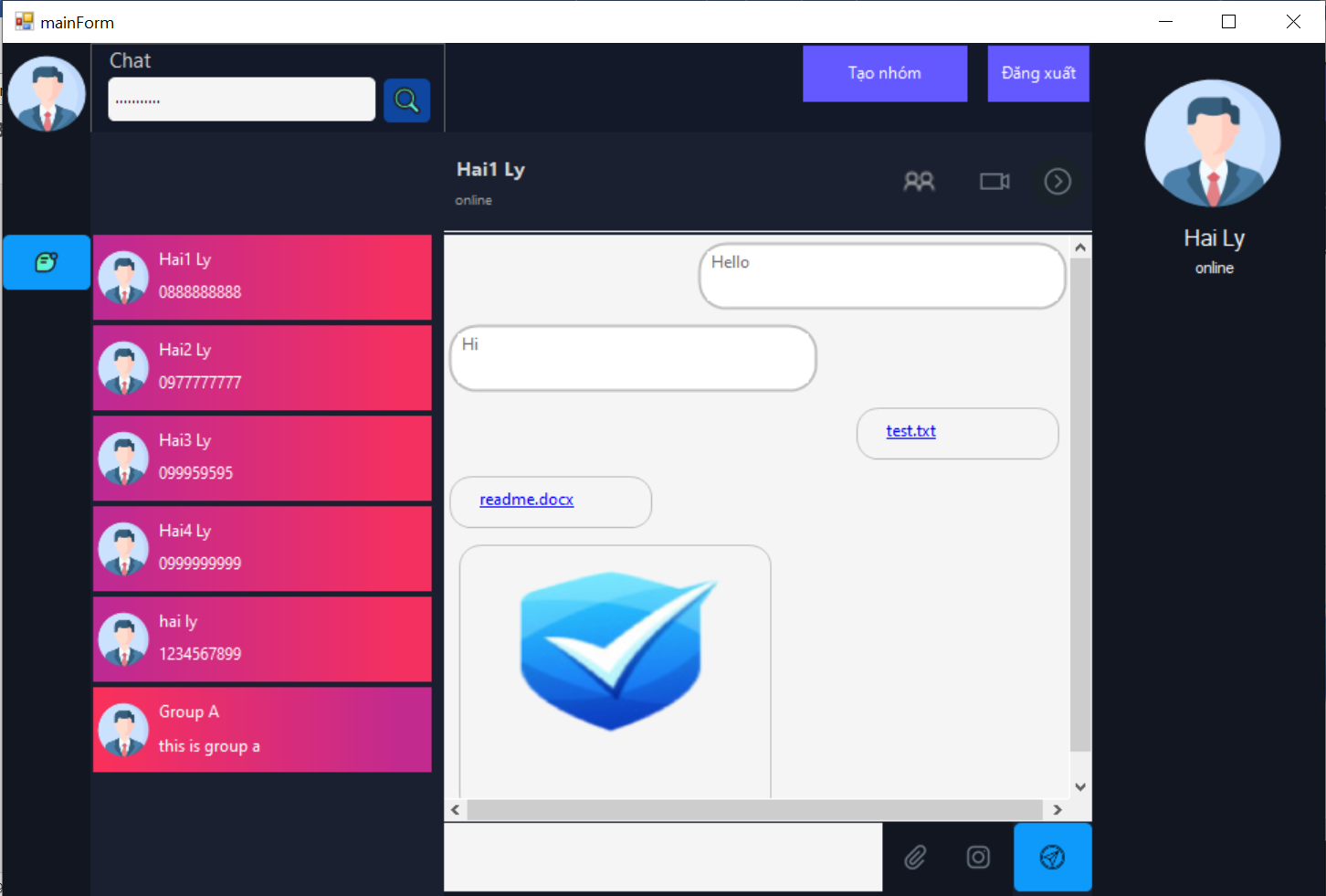
|  |  |  |  |
| --- | --- | --- | --- |
| **Giao diện** | Màn hình chính của người dùng | | |
| **Mô tả** | Cho phép người dùng tạo nhóm để có thể chat nhiều người với nhau | | |
| **Truy cập** | Chọn nút tạo nhóm | | |
| **Nội dung giao diện** | | | |
| **Thành phần** | **Kiểu** | **Dữ liệu** | **Mô tả** |
| Nút tạo nhóm | Button |  | Trường này cho phép tạo nhóm |
| **Các hành động trong giao diện** | | | |
| **Hành động** | **Mô tả** | **Thành công** | **Lỗi** |
| Tạo nhóm | Khi người dùng ấn nút tạo nhóm, hệ thống sẽ mở form tọa nhóm và người dùng có thể nhập các thông tin cần thiết và thêm người để có thể tạo nhóm | Tạo nhóm thành công và có thể chat | Hiện thông báo “một nhóm không thể ít hơn 3 người”.  Hiện thông báo “bạn chưa nhập thông tin đầy đủ” |



*Hình 3.2.5.1. Chức năng tạo nhóm.*

### Chức năng nhắn tin.

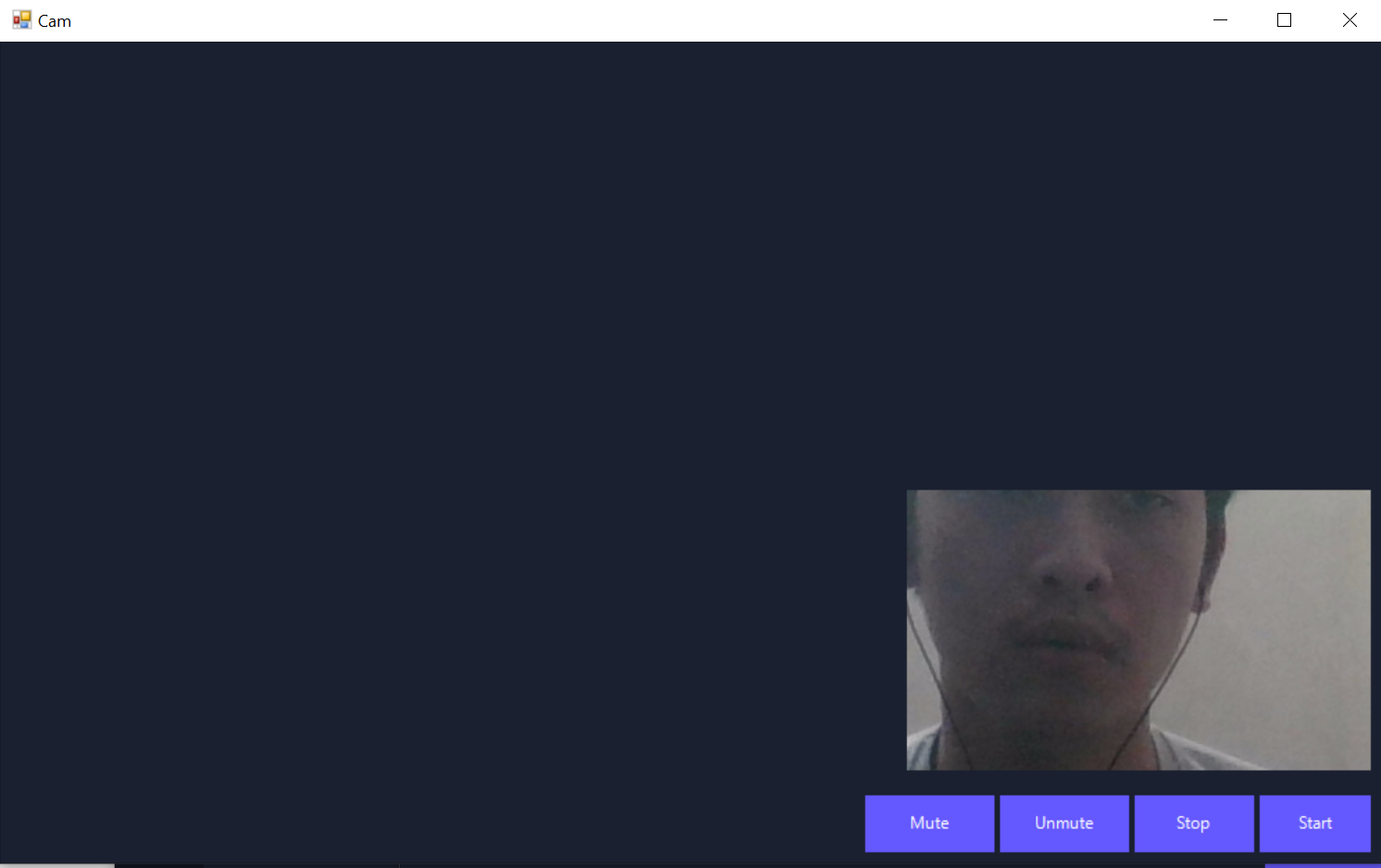
|  |  |  |  |
| --- | --- | --- | --- |
| **Giao diện** | Giao diện chính | | |
| **Mô tả** | Cho phép người dùng nhắn tin | | |
| **Truy cập** | Nhập thông tin và nhấn nút gửi | | |
| **Nội dung giao diện** | | | |
| **Thành phần** | **Kiểu** | **Dữ liệu** | **Mô tả** |
| Ô nhập thông tin | TextBox – String(100) |  | Trường dành cho khách hàng nhập tin nhắn |
| Nút gửi tin | Button |  | Gửi tin nhắn |
| Nút chọn file | Button |  | Chọn file để chia sẽ |
| Nút chọn ảnh | Button |  | Chọn ảnh để chia sẽ |
| **Các hành động trong giao diện** | | | |
| **Hành động** | **Mô tả** | **Thành công** | **Lỗi** |
| Nhắn tin | Khi người dùng chọn nhắn tin với client hoặc nhắn tin với nhóm sau đó họ nhập tin hoặc chọn file hoặc chọn ảnh và nhấn nút gửi | Tin nhắn sẽ được gửi đi. | Tin nhắn sẽ không được gửi đi. |



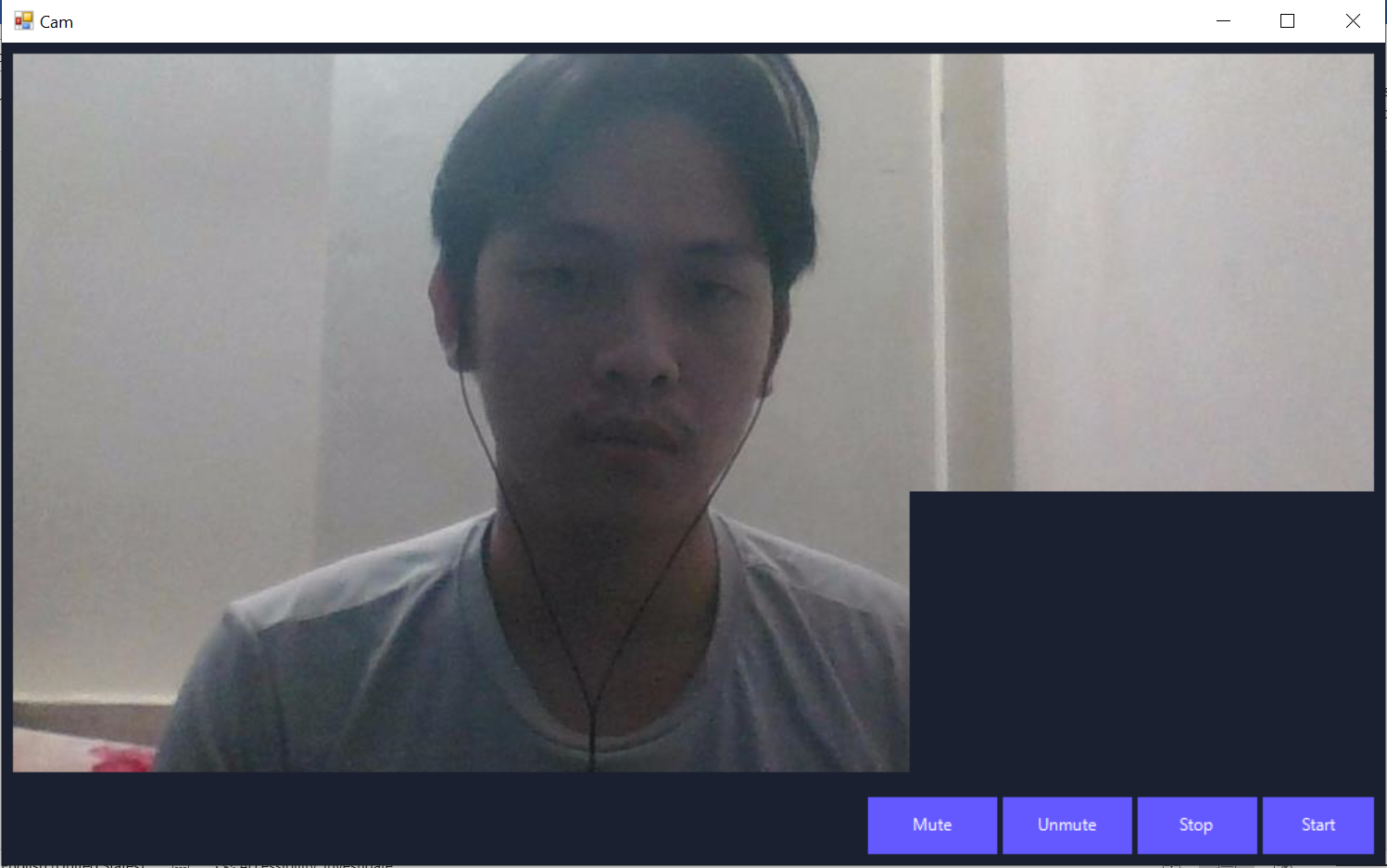
*Hình 3.2.6.1. Chức năng nhắn tin.*

### Chức năng gọi video.

|  |  |  |  |
| --- | --- | --- | --- |
| **Giao diện** | Giao diện chính | | |
| **Mô tả** | Cho phép người dùng gọi video với nhau | | |
| **Truy cập** | Ấn nút video call | | |
| **Nội dung giao diện** | | | |
| **Thành phần** | **Kiểu** | **Dữ liệu** | **Mô tả** |
| Nút mở form video | Button |  | Mở khung chat video |
| Nút mở cam | Button |  | Mở camera |
| Nút tắt cam | Button |  | Tắt camera |
| Nút mở mic | Button |  | Mở mic |
| Nút tắt mic | Button |  | Tắt mic |
| **Các hành động trong giao diện** | | | |
| **Hành động** | **Mô tả** | **Thành công** | **Lỗi** |
| Chat video | Khi người dùng chọn nút gọi video hệ thống sẽ mở khung chat video | Gọi thành công | Cuộc gọi không thành công |



*Hình 3.2.7.1. Chức năng chat video với người gọi.*



*Hình 3.2.7.2. Chức năng chat video với người nhận.*

## KẾT CHƯƠNG

Chương này trình bày, mô tả các chức năng và kết quả đạt được, đánh giá kết quả của sản phẩm.

KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

## 1. KẾT QUẢ ĐẠT ĐƯỢC

Trong thời gian tìm hiểu, nghiên cứu cơ sở lý thuyết và triển khai ứng dụng công nghệ, đồ án đã đạt được những kết quả sau:

*Về mặt lý thuyết,* đồ án đã đạt được tất cả những chức năng dự kiến ban đầu để làm. Giúp mọi người liên lạc, kết nối online với nhau.

*Về mặt thực tiễn ứng dụng*, đồ án đã đạt được kha khá những chức năng quan trọng như chat với nhau, call, call video, đã hoàn chỉnh được và không mắc lỗi cơ bản.

Tuy nhiên, đồ án còn tồn tại các vấn đề như sau:

* Vấn đề về giật lag, kết nối mạng ổn định nhưng chat với nhau còn bị lâu nhận tin nhắn

## HƯỚNG PHÁT TRIỂN

Một số số hướng nghiên cứu và phát triển của đề tài như sau:

* Chúng em đang nghiên cứu thêm về chat nhóm với nhau, giúp mọi người có thể làm việc theo team, dễ dàng thuận lợi hơn
* Và hơn nữa chúng em đang nghiên cứu để gửi thêm hình ảnh, sticker, các filter hay các file

TÀI LIỆU THAM KHẢO

**Tiếng Việt**

1. Phạm Minh Tuấn , bài giảng môn lập trình mạng, Đại học bách khoa Đà Nẵng.
2. Nguyễn Tấn Khôi , bài giảng môn mạng máy tính, Đại học bách khoa Đà Nẵng.
3. Trần Hồ Thủy Tiên , bài giảng môn lập trình trên hệ điều hành window, Đại học bách khoa Đà Nẵng.
4. Nguyễn Văn Hiệp, giáo trình lập trình mạng, Đại học bách khoa thành phố Hồ Chí Minh.
5. Đặng Hoài Phương, bài giảng môn lập trình C#, Đại học bách khoa Đà Nẵng.
6. Lê Thị Mỹ Hạnh , bài giảng môn phân tích và thiết kế hướng đối tượng , Đại học bách khoa Đà Nẵng.

**Tiếng Anh**

1. [Richard BLUM](https://www.google.com/search?hl=en&sxsrf=AOaemvLbxYwyGuyTvTxnvIZ3OEqxLcj4Ww:1638518627842&q=Richard+BLUM&stick=H4sIAAAAAAAAAOPgE-LRT9c3NDSuTK-oMjVU4gXz0qqqzJOT4ku0ZLKTrfST8vOz9cuLMktKUvPiy_OLsq0SS0sy8osWsfIEZSZnJBalKDj5hPruYGUEAOy-ZCNPAAAA&sa=X&ved=2ahUKEwjyrZLGlcf0AhXCJaYKHV16BskQmxMoAXoECDkQAw)(2002), C# network programing.
2. David Makofske, TCP/IP Sockets in C# practical guide for programmer.