**ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH**

**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN**

**-oOo-**

****

**ĐỒ ÁN CUỐI KỲ**

**MÔN: LẬP TRÌNH TRỰC QUAN (IT008.L13.PMCL - VN)**

**Đề tài: Xây dựng chương trình ChatUI**

**Giảng viên hướng dẫn:** Hoàng Văn Hà

**Sinh viên thực hiện:**

**Họ và tên**   **Mã số sinh viên**

Phùng Thanh Tú 19522455

Nguyễn Tiến Đạt 19521344

Vương Hoàng Trung 19522438

**TPHCM, 01/2021**

**Mở đầu**

Trong thời đại công nghệ ngày nay, việc ứng dụng công nghệ thông tin sẽ giúp công việc giải quyết một cách suôn sẻ và trôi chảy hơn. Việc trao đổi thông tin giữa các người dùng với nhau, hoặc người dùng muốn chia sẽ, trò chuyện với nhau là một nhu cầu cấp thiết.

Nhắc đến trao đổi thông tin, liên lạc thì không thể không nhắc đến những gã khổng lồ như Messenger của Facebook, Zalo của Việt Nam, và vô số những ứng dụng khác trên thị trường. Mỗi ứng dụng đều có điểm mạnh và yếu riêng của mình. Chung quy lại, chức năng chính của những ứng dụng này là để gửi nhận tin nhắn văn bản, hình ảnh hoặc tệp đính kèm. Một số ứng dụng sẽ có chức năng hội thoại, thậm chí là trò chuyện bằng video.

Điểm mạnh của ứng dụng trò chuyện trực tuyến đó là tính tiện lợi, nhanh chóng và tiết kiệm. Chỉ cần có internet, mọi thông tin hay tệp dữ liệu của bạn có thể gửi đến bất cứ đâu, bất cứ lúc nào với tốc độ cực kỳ nhanh. Không cần tốn tiền gửi thư, gửi bưu phẩm, hay phải chờ hàng tháng trời để nhận được thư và phản hồi.

**Lời cảm ơn**

Để thực hiện và hoàn thành tốt đồ án này, em đã nhận được sự giúp đỡ và hướng dẫn rất tận tình của các thầy cô thuộc Khoa Công Nghệ Phần Mền – Đại Học Công Nghệ Thông Tin. Em xin cảm ơn các thầy cô thuộc bộ môn chuyên ngành đã cung cấp cho chúng em các thông tin, kiến thức vô cùng quý báu và cần thiết trong suốt thời gian quá để em có thể thực hiện và hoàn thành đồ án của mình. Đặc biệt em xin chân thành cảm ơn thành thầy Hoàng Văn Hà người đã trực tiếp hướng dẫn chúng em trong thời gian thực hiện đồ án này.

Cuối cùng, xin chân thành cảm ơn các bạn trong ngành công nghệ thông tin đã ủng hộ, giúp đỡ, chia sẻ kiến thức, kinh nghiệm và tài liệu có được giúp chúng tôi trong quá trình nghiên cứu và thực hiện đề tài.

Do giới hạn về mặt thời gian và kiến thức cũng như kinh nghiệm thực tiễn nên đề tài không tránh khỏi những sai xót. Em rất mong nhận được sự thông cảm của quý thầy cô và mong đón nhận những góp ý của thầy cô và các bạn.

Em xin chân thành cảm ơn!

**NHẬN XÉT**

**(Của giảng viên hướng dẫn)**

**………………………………………………………………………………………………………………………………………………………………………………………………………………………………**

**………………………………………………………………………………………………………………………………………………………………………………………………………………………………**

**………………………………………………………………………………………………………………………………………………………………………………………………………………………………**

**………………………………………………………………………………………………………………………………………………………………………………………………………………………………**

**………………………………………………………………………………………………………………………………………………………………………………………………………………………………**

**………………………………………………………………………………………………………………………………………………………………………………………………………………………………**

**……………………………………………………………………………………………………………………………………………………………………………………………………………………………..**

**………………………………………………………………………………………………………………………………………………………………………………………………………………………………**

**………………………………………………………………………………………………………………………………………………………………………………………………………………………………**

HCM, ngày ….. tháng ….. năm 20…..

Giảng viên hướng dẫn

Hoàng Văn Hà

Contents

[Chương 1 : Giới thiệu đề tài 7](#_Toc61040368)

[1.1 Tổng quan 7](#_Toc61040369)

[1.1.1 Bối cảnh thực hiện đề tài 7](#_Toc61040370)

[1.1.2 Vấn đề cần giải quyết 7](#_Toc61040371)

[1.2 Phương pháp, kết quả 7](#_Toc61040372)

[1.2.1 Phương pháp 7](#_Toc61040373)

[1.2.2 Kết quả 7](#_Toc61040374)

[1.3 Cấu trúc đồ án 7](#_Toc61040375)

[Chương 2: Các kiến thức/công nghệ nền tảng 7](#_Toc61040376)

[2.1 TCP/IP 7](#_Toc61040377)

[2.1.1 Tổng quan 7](#_Toc61040378)

[2.1.2 Phương thức hoạt động của bộ giao thức TCP/IP 7](#_Toc61040379)

[2.2 C# 7](#_Toc61040380)

[2.2.1 C# là gì 7](#_Toc61040381)

[2.2.2 Đặc trưng của C# 7](#_Toc61040382)

[2.3 .Net 7](#_Toc61040383)

[2.3.1 Tổng quan về .Net 7](#_Toc61040384)

[2.3.2 .Net bao gồm những gì 7](#_Toc61040385)

[2.3.3 Lý do .Net trở thành ngôn ngữ phổ biến và được ứng dụng nhiều 7](#_Toc61040386)

[2.4 Socket 7](#_Toc61040387)

[2.4.1 Socket là gì 7](#_Toc61040388)

[2.4.2 Socket hoạt động như thế nào 7](#_Toc61040389)

[2.4.3 Phân loại Socket 7](#_Toc61040390)

[2.5 Windows Froms 7](#_Toc61040391)

[2.5.1 Tổng quan về Windows Froms 7](#_Toc61040392)

[2.5.2 Các thuộc tính cơ bản Windows Froms trong C# 7](#_Toc61040393)

[ToolBox 7](#_Toc61040394)

[Form 7](#_Toc61040395)

[Properties 7](#_Toc61040396)

[Chương 3: Xây dựng ứng dụng 7](#_Toc61040397)

[3.1 Phân tích/mô tả một vài qui trình /cách thức sử dụng các tính năng trong ứng dụng 7](#_Toc61040398)

[3.1.1 Xử lý đăng nhập 7](#_Toc61040399)

[3.1.2 Đăng ký 7](#_Toc61040400)

[3.1.3 Tạo Chatroom 7](#_Toc61040401)

[3.1.4 Tham gia vào Chatroom 7](#_Toc61040402)

[3.1.5 Mô hình luồng xử lý gửi tin nhắn 7](#_Toc61040403)

[3.2 Các thành phần/class trong ứng dụng 7](#_Toc61040404)

[3.2.1 Data persistence ( Lưu dữ liệu) 7](#_Toc61040405)

[3.2.2 Exception Handling (Xử lý ngoại lệ) 7](#_Toc61040406)

[3.2.3 Class Session 7](#_Toc61040407)

[3.2.4 Class Message 7](#_Toc61040408)

[3.2.5 Class TcpClient 7](#_Toc61040409)

[3.2.6 Class TcpServer 7](#_Toc61040410)

[3.2.7 Class Client 7](#_Toc61040411)

[3.2.8 Class ThreadedBindingList 7](#_Toc61040412)

[3.2.9 Class Chat 7](#_Toc61040413)

[3.2.10 Class Server 7](#_Toc61040414)

[3.2.11 Class ServerGUI và TextBoxStreamWriter 7](#_Toc61040415)

[Chương 4: Thử nghiệm đánh giá 7](#_Toc61040416)

[4.1 Môi trường thử nghiệm 7](#_Toc61040417)

[4.2 Kết quả thu được 7](#_Toc61040418)

[Chương 5: Kết luận và hướng phát triển 7](#_Toc61040419)

[5.1 Những hạn chế của ứng dụng 7](#_Toc61040420)

[5.2 Hướng phát triển 7](#_Toc61040421)

[5.3 Những kiến thức thú vị đã rút ra được 7](#_Toc61040422)

## 

# Chương 1 : Giới thiệu đề tài

## 1.1 Tổng quan

### 1.1.1 Bối cảnh thực hiện đề tài

Trong quá trình phát triển xã hội, công nghệ thông tin đã và đang phát triển mạnh mẽ, ngày càng thâm nhập sâu, trở thành một công cụ hỗ trợ đắc lực cho con người trong mọi lĩnh vực.

Và cùng với sự phát triển đó nhu cầu liên lạc, trao đổi thông tin giữa người với người ngày càng tăng cao và cần thiết. Dựa trên xu thế phát triển của xã hội cũng như nhu cầu thiết thực của con người, ứng dụng chat trò chuyện ra đời.

### 1.1.2 Vấn đề cần giải quyết

Nếu ai đã từng sử dụng Internet chắc sẽ không ít lần nghe hoặc sử dụng dịch vụ chat, đây là dịch vụ khá phổ biến hiện nay, nó cho phép bạn thiết lập các cuộc đối thoại thông qua máy vi tính với người dùng khác trên Internet. Sau khi bạn đã thiết lập được hệ thống này, những gì bạn làm trên máy tính của bạn như gõ chữ, nói chuyện, hình ảnh , truyền dữ liệu thì được hiển thị trên máy kia và ngược lại.

Dịch vụ chat còn đi vào lĩnh vực khác là ứng dụng trong một mạng của công ty có thể là mạng riêng của công ty đó hay mạng Internet. Nó giúp cho các quý giám đốc, những người quản lý không phải tốn nhiều công sức, thời gian khi cần thông báo việc gì đến nhân viên, việc đó có thể là quan trọng, không quan trọng, những vấn đề bí mật,cả đến những vấn đề riêng tư mà mà không sợ các đồng nghiệp khác hoặc cấp dưới biết....  
 Xuất phát từ lợi ích mà nó đem lại, với mong muốn tạo ra một ứng dụng chat để mọi người trao đổi thông tin với nhau cho nên nhóm chọn đề tài “ Xây dung chương trình LetsChat”

## 1.2 Phương pháp, kết quả

### 1.2.1 Phương pháp

Thực hiện nghiên cứu những công nghệ, công cụ cần sử dụng để thực hiện dự án. Lựa chọn những công nghệ tối ưu và hiện đại nhất để bắt kịp xu thế phát triển không ngừng của công nghệ hiện nay.  
Sau khi đã nắm bắt được công nghệ và hướng đi, bắt tay vào lập kế hoạch triển khai dự án, đứa ra những công việc cụ thể cần thực hiện. Tiến hành phân tích thiết kế, chỉ những rủi ro, tính khả thi cũng như bất khả thi của những công việc cần thực hiện.  
Sau khi phân tích, tiến hành thiết kế cơ sở dữ liệu, thiết kế giao diện và từng bước hoàn thiện, ghép nối dự án thành sản phẩm hoàn chỉnh.Sau khi có sản phẩm hoàn chỉnh, tiến hành thử nghiệm và chỉnh sửa, cập nhật những sai sót.  
Cuối cùng là tổng kết và đưa ra hướng phát triển trong tương lai.

### 1.2.2 Kết quả

Nghiên cứu và học tập được các công nghệ mới, được áp dụng để phát triển ứng dụng như Socket.NET, TCP/IP.  
Sau quá trình thực hiện, kết quả thu được là một ứng dụng trò chuyện trực tuyến với các chức năng:  
- Trò chuyện văn bản giữa hai cá nhận.  
- Trò chuyện văn bản giữa một nhóm gồm nhiều thành viên.

## 1.3 Cấu trúc đồ án

Nguyên cứu tổng quan: giới thiệu cơ sở nghiên cứu lý thuyết, những mô hình, công nghệ được áp dụng trong quá trình xây dựng dự án:  
- TCP/IP  
- C#  
- .NET  
- Socket  
- Windows Froms  
Triển khai xây dựng:

- Phân tích/mô tả một vài quy trình/cách thức sử dụng các tính năng trong ứng dụng  
- Các module/thành phần/class trong ứng dụng  
- Xây dựng giao diện

Kết luận và hướng phát triển:  
- Nhìn nhận, đánh giá những điểm được và chưa được trong quá trình thực hiện dự án. Rút kinh nghiệm lần sau.  
- Hoạch định ra hướng phát triển trong tương lai

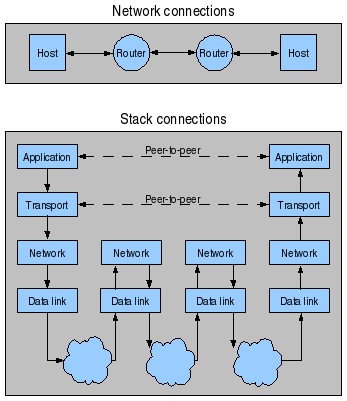
# Chương 2: Các kiến thức/công nghệ nền tảng

## 2.1 TCP/IP

### 2.1.1 Tổng quan

TCP/IP là bộ giao thức cho phép kết nối các hệ thống mạng không đồng nhất với nhau. Ngày nay TCP/IP được sử dụng rộng rãi trong mạng cục bộ cũng như mạng toàn cầu. TCP/IP được xem như giản lược của mô hình tham chiếu OSI với 4 tầng như sau:

* Tầng Liên Kết (Datalink Layer)
* Tầng Mạng (Internet Layer)
* Tầng Giao Vận (Transport Layer)
* Tầng Ứng Dụng (Application Layer)



*Hình 1 : Các tầng trong bộ giao thức TCP/IP*

* **Tầng liên kết:**

Tầng liên kết (còn được gọi là tầng liên kết dữ liệu hay tầng giao tiếp mạng) là tầng thấp nhất trong mô hình TCP/IP, bao gồm các thiết bị giao tiếp mạng và các chương trình cung cấp các thông tin cần thiết để có thể hoạt động, truy nhập đường truyền vật lý qua các thiết bị giao tiếp mạng đó.

* **Tầng Internet:**

Tầng Internet ( hay còn gọi là tầng Mạng) xử lý quá trình truyền gói tin trên mạng, các giao thức của tầng này bao gồm : IP ( Internet Protocol) , ICMP ( Internet Control Message Protocol) , IGMP ( Internet Group Message Protocol ).

* **Tầng giao vận:**

Tầng giao vận phụ trách luồng dữ liệu giữa 2 trạm thực hiện các ứng dụng của tầng trên, tầng này có 2 giao thức chính là TCP ( Transmisson Control Protocol) và UDP ( User Datagram Protocol ) .

- TCP cung cấp luồng dữ liệu tin cậy giữa 2 trạm, nó sử dụng các cơ chế như chia nhỏ các gói tin ở tầng trên thành các gói tin có kích thước thích hợp cho tầng mạng bên dưới, báo nhận gói tin, đặt hạn chế thời gian timeout để đảm bảo bên nhận biết được các gói tin đã gửi đi. Do tầng này đảm bảo tính tin cậy nên tầng trên sẽ không cần quan tâm đến nữa.

- UDP cung cấp một dịch vụ rất đơn giản hơn cho tầng ứng dụng . Nó chỉ gửi dữ liệu từ trạm này tới trạm kia mà không đảm bảo các gói tin đến được tới đích. Các cơ chế đảm bảo độ tin cậy được thực hiện bởi tầng trên tầng ứng dụng

* **Tầng ứng dụng:**

Là tầng trên của mô hình TCP/IP bao gồm các tiến trình và các ứng dụng cung cấp cho người sử dụng để truy cập mạng. Có rất nhiều ứng dụng được cung cấp trong tầng này, mà phổ biến là Telnet: sử dụng trong việc truy cập mạng từ xa, FTP ( File Transport Protocol ) dịch vụ truyền tệp tin., EMAIL : dịch vụ truyền thư tín điện tử. WWW ( Word Wide Web ).

### 2.1.2 Phương thức hoạt động của bộ giao thức TCP/IP



*Hình 2:* Quá trình đóng mở gói dữ liệu trong TCP/IP

Cũng tương tự như trong mô hình OSI, khi truyền dữ liệu , quá trình tiến hành từ tầng trên xuống tầng dưới, qua mỗi tầng dữ liệu được thêm vào thông tin điều khiển gọi là Header. Khi nhận dữ liệu thì quá trình xảy ra ngược lại, dữ liệu được truyền từ tầng dưới lên và qua mỗi tầng thì phần header tương ứng sẽ được lấy đi và khi đến tầng trên cùng thì dữ liệu không còn phần header nữa.



Hình 3: Cấu trúc dữ liệu trong TCP/IP

Hình trên cho ta thấy lược đồ dữ liệu qua các tầng. Trong hình ta thấy tại các tầng khác nhau dữ liệu được mang những thuật ngữ khác nhau:

* Trong tầng ứng dụng: dữ liệu là các luồng được gọi là stream.
* Trong tầng giao vận: đơn vị dữ liệu mà TCP gửi xuống gọi là TCP segment.
* Trong tầng mạng, dữ liệu mà IP gửi xuống tầng dưới gọi là IP Datagram.
* Trong tầng liên kết, dữ liệu được truyền đi gọi là frame.

## 2.2 C#

### 2.2.1 C# là gì

C# (hay C sharp) là một ngôn ngữ lập trình đơn giản, được phát triển bởi đội ngũ kỹ sư của Microsoft vào năm 2000, trong đó người dẫn đầu là Anders Hejlsberg và Scott Wiltamuth.

C# là ngôn ngữ lập trình hiện đại, hướng đối tượng và nó được xây dựng trên nền tảng của hai ngôn ngữ mạnh nhất là C++ và Java.

C# được thiết kế cho Common Language Infrastructure (CLI), mà gồm Executable Code và Runtime Environment, cho phép chúng ta sử dụng các ngôn ngữ high-level đa dạng trên các nền tảng và cấu trúc máy tính khác nhau.

C# với sự hỗ trợ mạnh mẽ của .NET Framework giúp cho việc tạo một ứng dụng Windows Forms hay WPF (Windows Presentation Foundation), . . . trở nên rất dễ dàng.

### 2.2.2 Đặc trưng của C#

#### C# là một ngôn ngữ đơn giản

 C# dựng trên nền tảng C++ và Java nên ngôn ngữ C# khá đơn giản. Nếu chúng ta thân thiện với C và C++ hoậc thậm chí là Java, chúng ta sẽ thấy C# khá giống về diện mạo, cú pháp, biểu thức, toán tử và những chức năng khác được lấy trực tiếp từ ngôn ngữ C và C++, nhưng nó đã được cải tiến để làm cho ngôn ngữ đơn giản hơn. Một vài trong các sự cải tiến là loại bỏ các dư thừa, hay là thêm vào những cú pháp thay đổi.

#### C# là ngôn ngữ hiện đại

 C# chứa các đặt những đặc tính như là xử lý ngoại lệ, thu gom bộ nhớ tự động, những kiểu dữ liệu mở rộng, và bảo mật mã nguồn là những đặc tính được mong đợi trong một ngôn ngữ hiện đại.

#### **C# là một ngôn ngữ lập trình thuần hướng đối tượng** Lập trình hướng đối tượng(tiếng Anh: Object-oriented programming, viết tắt: OOP) là một phương pháp lập trình có 4 tính chất. Đó là tính trừu tượng (abstraction), tính đóng gói (encapsulation), tính đa hình (polymorphism) và tính kế thừa (inheritance). C# hỗ trợ cho chúng ta tất cả những đặc tính trên

#### C# là ngôn ngữ ít từ khóa C# là ngôn ngữ sử dụng giới hạn những từ khóa. Phần lớn các từ khóa được sử dụng để mô tả thông tin. Chúng ta có thể nghĩ rằng một ngôn ngữ có nhiều từ khóa thì sẽ mạnh hơn. Điều này không phải sự thật, ít nhất là trong trường hợp ngôn ngữ C#, chúng ta có thể tìm thấy rằng ngôn ngữ này có thể được sử dụng để làm bất cứ nhiệm vụ nào.

## 2.3 .Net

### 2.3.1 Tổng quan về .Net

.Net Framework là một khuôn khổ phần mềm chạy chủ yếu trên Microsoft Windows được phát triển bởi [Microsoft](https://www.microsoft.com/vi-vn). Nó có khả năng tương tác ngôn ngữ qua nhiều ngôn ngữ lập trình và bao gồm một thư viện lớp lớn có tên là Framework Class Library – viết tắt là FCL. .Net Framework thực thi các chương trình trong môi trường phần mềm có tên là Common Language Runtime – viết tắt là CLR. Đây là một máy ảo ứng dụng cung cấp các dịch vụ như: quản lý bộ nhớ, bảo mật và xử lý các ngoại lệ. Như vậy, .NET Framework được tạo ra từ 2 môi trường FCL và CLR.  
FCL cung cấp giao diện cho người dùng, kết nối cơ sở dữ liệu, truy cập dữ liệu, mật mã, truyền thông tin mạng, các thuật toán số và [phát triển ứng dụng web](https://mona.solutions/dich-vu/lap-trinh-web-app/) và [thiết kế web](https://mona.media/thiet-ke-website-tai-hcm/).

.Net Framework được tạo ra nhằm hoàn thành các mục tiêu sau:

+ Nhằm cung cấp cho người dùng một ngôn ngữ lập trình hướng đối tượng, tương thích hầu hết với các mã lưu trữ và thực hiện của đối tượng.

+ Nhằm giảm thiểu việc triển khai phần mềm và xung đột, thúc đẩy việc thực hiện mã an toàn, cũng như loại bỏ các vấn đề về hiệu suất của các môi trường kịch bản hoặc diễn giải bằng cách cung cấp môi trường thực thi mã.

+ Nhằm làm cho trải nghiệm của nhà phát triển nhất quán trên nhiều ứng dụng như: các ứng dụng dựa trên web, các ứng dụng dựa trên Windows.

+ .NET Framework có khả năng tích hợp với bất kỳ mã khác nên có thể xây dựng được tất cả thông tin liên lạc về các tiêu chuẩn ngành công nghiệp

### 2.3.2 .Net bao gồm những gì

.NET Framework bao gồm: thư viện lớp và thời gian chạy ngôn ngữ phổ biến.

**Thời gian chạy phổ biến** chính là nền tảng của .NET Framework, nó cung cấp các dịch vụ cốt lõi như: quản lý luồng và remoting, quản lý bộ nhớ, thúc đẩy tính mạnh mẽ và tính bảo mật. Nguyên tắc cơ bản của thời gian chạy đó chính là khái niệm về quản lý mã. Mã quản lý nhắm vào mục tiêu thời gian chạy, mã không quản lý không nhắm vào mục tiêu thời gian chạy.

.NET Framework bắt đầu thời gian chạy bằng mã được quản lý và có khả năng lưu trữ bằng những mã không được quản lý nhằm tạo ra một một trường phần mềm có cả 2 tính năng được quản lý và không được quản lý. .NET Framework ngoài chức ngăn cung cấp thời gian chạy cho máy chủ còn hỗ trợ phát triển thời gian chạy.  
**Thư viện lớp NET Framework**là tập hợp tất cả các loại tái sử dụng được tích hợp một cách chặt chẽ với thời gian chạy thông dụng. Đây chính là công cụ hướng đối tượng cung cấp các loại tính năng mà từ đó mã được quản lý được thực hiện. Điều này không chỉ làm cho .NET Framework dễ dàng sử dụng mà còn làm giảm thời gian kết hợp các tính năng.

Từ một thư viện hướng đối tượng của .NET Framework cho phép bạn hoàn thành các tác vụ như: Thu thập dữ liệu, quản lý chuỗi, kết hợp cơ sở dữ liệu và truy cập tệp.

.Net được ứng dụng rất nhiều trong thiết kế web và cụ thể đó là ASP.NET C# đây là một ngôn ngữ khó và phức tạp vì vậy chỉ những dịch vụ thiết kế web chuyên nghiệp mới co những nhân viên chuyên gia về .NET.

Web app là một mảng được .NET thực hiện rất nhiều đây là mảng mà các ngôn ngữ khác khó có thể làm được hoàn hảo và bảo mật bằng .net.

### 2.3.3 Lý do .Net trở thành ngôn ngữ phổ biến và được ứng dụng nhiều

Ở Việt Nam chỉ có các [Freelancer](http://freelancervietnam.vn/)lâu năm hoặc các công ty lớn mới sử dụng ngôn ngữ này còn những công ty thiết kế web rẻ tiền thì họ không đủ lực để sử dụng ngôn ngữ mạnh mẽ và phức tạp này. Tuy nhiên với khả năng tiếp cận và thích ứng tốt của các Lập trình viên, số lượng học viên và lập trình viên muốn [tay ngang tự học lập trình](https://www.youtube.com/watch?v=So0luoTtl_o) .NET để nâng cao khả năng cũng như tìm thêm cơ hội thử sức mới đang ngày một phổ biến hơn.

Với sự hỗ trợ đến từ Microsoft, .NET trở thành ngôn ngữ có tính bảo mật và nhất quán trong lập trình, hầu hết các lập trình viên cho đến công ty đều phải tuân theo một số quy định nhất định của Microsoft cho ngôn ngữ này nhằm đảm bảo quyền cũng như xây dựng nên tảng lâu dài cho tương lai.

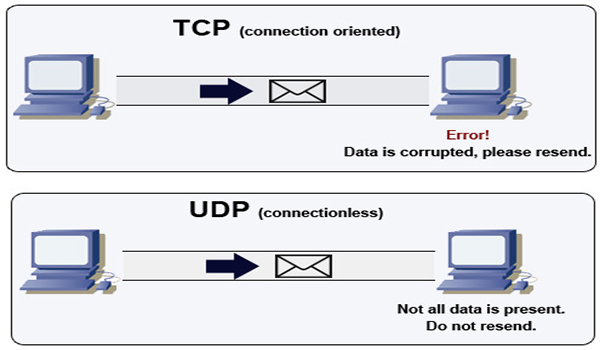
## 2.4 Socket

### 2.4.1 Socket là gì

**Socket là điểm cuối end-point trong liên kết truyền thông hai chiều (two-way communication) biểu diễn kết nối giữa Client – Server**. Các lớp Socket được ràng buộc với một cổng port (thể hiện là một con số cụ thể) để các tầng TCP (TCP Layer) có thể định danh ứng dụng mà dữ liệu sẽ được gửi tới  
**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.

### 2.4.2 Socket hoạt động như thế nào



Hình 4: Cách hoạt động của Socket  
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:

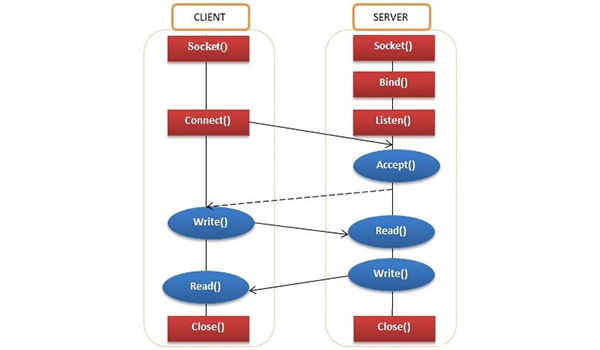
* 2 ứng dụng có thể nằm cùng trên một máy hoặc 2 máy khác nhau
* Trong trường hợp 2 ứng dụng cùng nằm trên một máy, số hiệu cổng không được trùng nhau.

### 2.4.3 Phân loại Socket

**Socket**được chia làm 4 phân loại khác nhau.

* Stream Socket
* Datagram Socket
* Websocket
* Unix socket

#### a. Stream Socket



Hình 5 : Hoạt động của Stream Socket

**Stream Socket** hay còn gọi là socket hướng kết nối, là socket hoạt động thông qua **giao thức TCP (Transmission Control Protocol)**. Stream Socket chỉ hoạt động khi server và client đã kết nối với nhau.

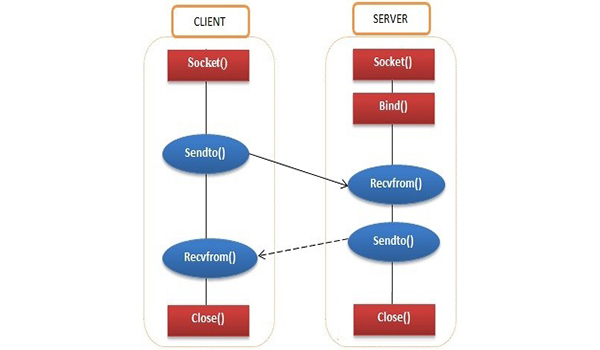
#### **Ưu điểm của Stream Socket là gì?**

* Dữ liệu truyền đi được đảm bảo truyền đến đúng nơi nhận, đúng thứ tự với thời gian nhanh chóng
* Mỗi thông điệp gửi đi đều có xác nhận trả về để thông báo cho người dùng thông tin về quá trình truyền tải.

#### Nhược điểm của Stream Socket là gì?

* Giữa máy chủ và máy nhận chỉ có 1 IP, nên khi kết nối, 1 máy phải chờ máy còn lại chấp nhận kết nối.

#### b. Datagram Socket



Hình 6: Hoạt động của Datagram Socket

Datagram **Socket** hay còn gọi là socket không hướng kết nối, là socket hoạt động thông qua**giao thức UDP ( User Datagram Protocol)**. Datagram Socket có thể hoạt động kể cả khi không có sự thiết lập kết nối giữa 2 máy với nhau.

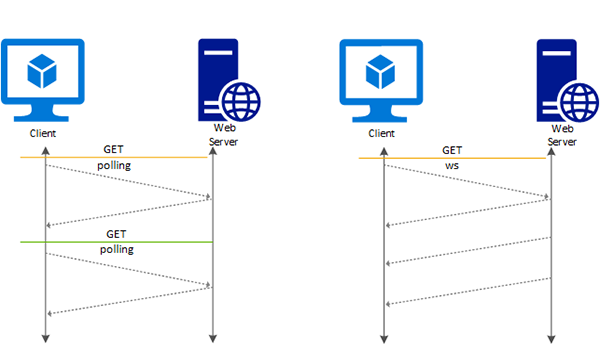
#### **Ưu điểm của Datagram Socket là gì?**

* Quá trình kết nối và truyền tải thông tin đơn giản, không cần thực hiện nhiều thao tác.
* Thời gian truyền tải dữ liệu cực nhanh.

#### **Nhược điểm của Datagram Socket là gì?**

* Quá trình truyền thông tin không đảm bảo tin cậy, thông tin có thể truyền sai thứ tự hoặc bị lặp.

#### c. Websocket



Hình 7: Hoạt động của Websocket

**Websocket** là công cụ hỗ trợ việc kết nối qua lại trên internet giữa client và server. Giúp diễn ra nhanh chóng và hiệu quả hơn thông qua việc sử dụng **TCP socket**. Không chỉ sử dụng riêng cho ứng dụng web, **Websocket** có thể áp dụng cho bất kì ứng dụng nào khác cần có sự trao đổi thông tin trên Internet.

#### **Ưu điểm của Websocket là gì?**

**Websocket** mang lại nhiều ưu điểm trong việc kết nối giữa client và server. Cụ thể như sau:

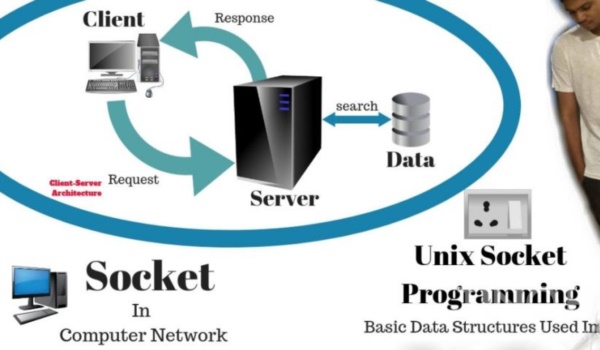
* Tăng tốc độ truyền tải thông tin giữa 2 chiều
* Dễ phát hiện và xử lý trong trường hợp có lỗi xảy ra
* Dễ dàng sử dụng, không cần cài đặt thêm các phần mềm bổ sung khác
* Không cần sử dụng nhiều phương pháp kết nối khác nhau

#### **Nhược điểm của Websocket là gì?**

Một số nhược điểm của **Websocket** mà bạn cần lưu ý khi sử dụng có thể kể đến như:

* Chưa hỗ trợ trên tất cả các trình duyệt
* Với các dịch vụ có phạm vi yêu cầu, Websocket chưa hỗ trợ hoàn toàn.

#### d. Unix Socket

  
 Hình 8: Hoạt động của Unix Socket  
**Unix socket** là điểm giao tiếp hỗ trợ trao đổi giữa các ứng dụng khác nhau ngay trên cùng máy tính. Mọi hoạt động Unix socket diễn ra ngay ở nhân của hệ điều hành. Nhờ vậy, tốc độ kết nối và truyền tải giữa các ứng dụng nhanh, nhẹ và hiệu quả hơn.

**Unix socket** có thể tránh được các bước kiểm tra hoặc routing. Vì thế mà quá trình truyền tin sẽ đảm bảo và dễ dàng hơn.

#### **Ưu điểm Unix socket là gì?**

**Unix socket** có nhiều ưu điểm vượt trội. Nó hỗ trợ cho việc truyền tải nhanh hơn và hiệu quả hơn. Trong đó, tiêu biểu phải kể đến các ưu điểm như:

* Tăng tốc độ truy cập MySQL lên đến 30-50%
* Giảm thời gian latency xuống, từ 60ms còn 5ms
* Tăng PostgreSQL lên  hơn 30%
* Tăng Redis lên 50%
* …

#### **Nhược điểm Unix socket là gì?**

Bên cạnh các ưu điểm nổi bật phía trên, **Unix socket** cũng còn tồn tại một số nhược điểm như sau:

* Trong trường hợp các ứng dụng nằm trên những máy chủ khác nhau, sẽ không thể kết nối bằng Unix socket.
* Vấn đề phân quyền giữa các tệp tin trên Unix socket đôi khi vẫn xảy ra, ảnh hưởng đến việc sử dụng và thao tác.

## 2.5 Windows Froms

### 2.5.1 Tổng quan về Windows Froms

•Windows Forms là cách cơ bản để cung cấp các thành phần giao diện (GUI components) cho môi trường .NET Framework

•Windows Forms được xây dựng trên thư viện Windows API

•Windows Forms cơ bản bao gồm

-Một Form là khung dùng hiển thị thông tin đến người dùng

-Các Control được đặt trong form và được lập trình để đáp ứng sự kiện

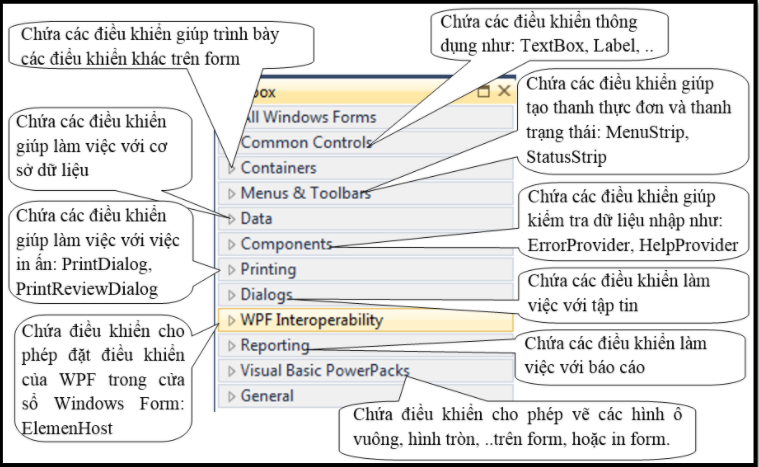
### 2.5.2 Các thuộc tính cơ bản Windows Froms trong C#

### ToolBox

**ToolBox** là nơi chứa các điều khiển để thiết kế giao diện, để mở cửa sổ **ToolBox**các bạn vào View | ToolBox (Ctrl + Alt + X).

C# cung cấp danh sách các **Component/Control** được liệt kê theo nhóm. Cho phép các bạn sử dụng thao tác kéo thả vào form để thiết kế giao diện cho chương trình.

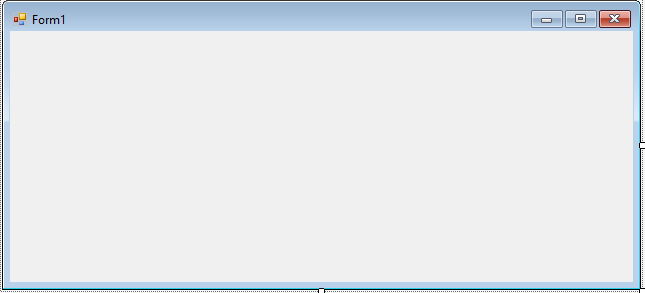
Trong ToolBox có các nhóm điều khiển để thiết kế giao diện như sau:



Hình 9: Các nhóm thiết kế giao diện

### Form

**Form**là vùng để thiết kế giao diện, ta chọn nhấn dữ chuột trái kéo điều khiển vào form hoặc double click vào điều khiển mà bạn muốn thiết kế. Các điều khiển còn được gọi là control hay component



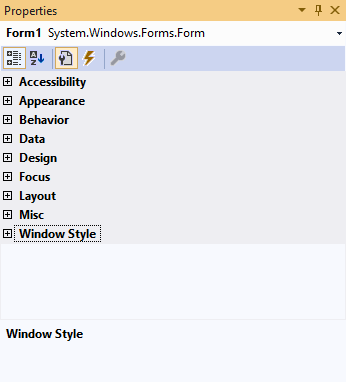
Hình 10: Giao diện Form

Form được gọi là control "chứa" (vì nó có thể được chứa trong các control khác).

### Properties

**Properties**là nơi thiết lập thuộc tính của các điều khiển, với mỗi điều khiển/ Control đều được cung cấp sẵn một danh sách các thuộc tính để các bạn có thể thiết lập.

Hầu hết giữa các điều khiển đều có những thuộc tính chung (giống nhau) và những thuộc tính riêng đặc trưng cho điều khiển đó.



Hình 11: Properties

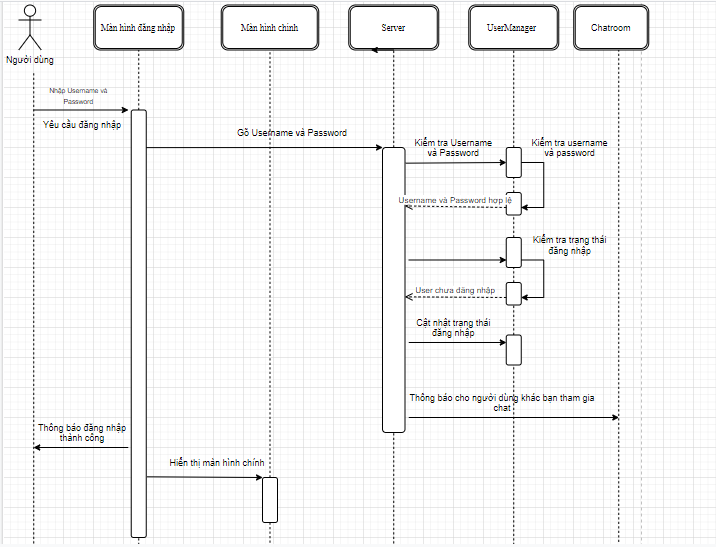
Với mỗi điều khiển, để xuất hiện cửa sổ thiết lập thuộc tính các bạn chỉ cần nhấp chuộc phải lên điều khiển đó và chọn properties trong menu hiện ra.

# Chương 3: Xây dựng ứng dụng

## 3.1 Phân tích/mô tả một vài qui trình /cách thức sử dụng các tính năng trong ứng dụng

### 3.1.1 Xử lý đăng nhập

\*Mô hình xử lý:



Hình 12: Mô hình xử lý đăng nhập

\* Mô tả qui trình xử lý:

Khi một User yêu cầu đăng nhập hệ thống, Client sẽ gởi Username và Password cho Server. Server sẽ kiểm tra xem Username và Password có hợp lệ hay không. Sau đó, Server sẽ kiểm tra Username này đã đăng nhập chưa. Sau khi xác nhận đúng thì Server chuyển đến Chat.

### 3.1.2 Đăng ký

Khi một User mới muốn đăng nhập hệ thống. Nhập các thông tin User và Password ở màn hình Login sau đó ấn Register . Server sẽ tạo một session để lưu trữ thông tin người dùng . Sau đó người dùng có thể sử dụng User và Password mình vừa tạo để đăng nhập vào chat.

### 3.1.3 Tạo Chatroom

Khi người dùng chọn chức năng Create a Chatroom thì màn hình tạo Room name sẽ xuất hiện. Sau đó nhập tên Chatroom mà bạn muốn .Client sẽ tự phát sinh ra tên Room. Chatroom mới sẽ nằm trong phần Select a chatroom.

### 3.1.4 Tham gia vào Chatroom

Người dùng ở màn hình chat, chọn chức năng Select a chatroom. Trong Select a chartroom, người dùng có thể chọn chatroom mà họ muốn để tham gia vô.

### 3.1.5 Mô hình luồng xử lý gửi tin nhắn

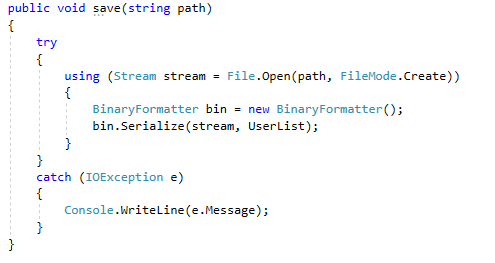
Khi người dùng nhập tin nhắn và nhấn phím Enter (hoặc sendButton), Client sẽ hiển thị tin nhắn trên tab Chat đồng thời gởi tin nhắn lên cho Server. Server sẽ gởi tin nhắn lên chatroom

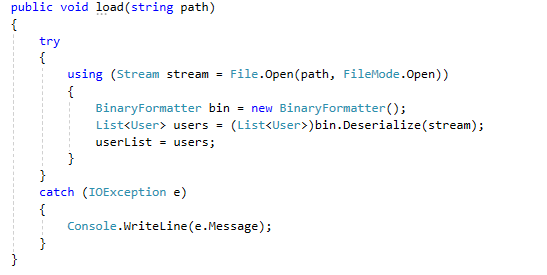
## 3.2 Các thành phần/class trong ứng dụng

### 3.2.1 Data persistence ( Lưu dữ liệu)

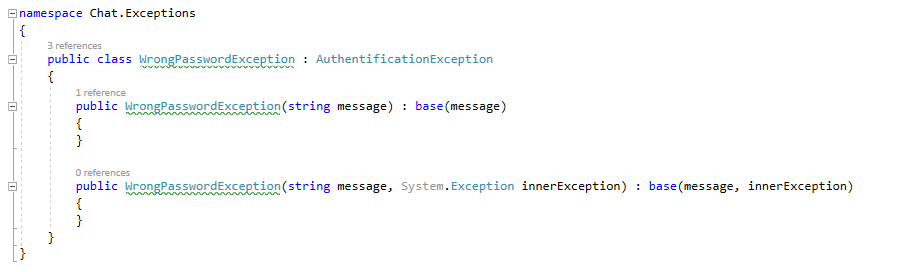
Lưu trữ dữ liệu để giữ ( danh sách Chatrooms và Users) trong static binary file (tệp nhị phân tĩnh).Sử dụng BinaryFomatter để lưu trạng thái hiện tại của lớp UserManager và ChatroomManager vô một tệp.

Việc sao lưu được thực hiện như sau:

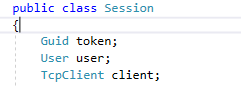


Khôi phục tệp cũng thực hiện tương tự:  


### 3.2.2 Exception Handling (Xử lý ngoại lệ)

Để xử lý các ngoại lệ xuất hiện trong subject, được thêm vào C# . Chúng đã thiết lập nên một class đặc biệt để đưa mỗi loại ngoại lệ vào trong đó.  
Mỗi lớp, rất đơn giản, kế thừa từ lớp cơ sở System.Exception:   
  


### 3.2.3 Class Session

Mỗi một người dùng sẽ được nhận dạng duy nhất thông qua vùng được liên kết với họ. Mỗi vùng chứa TcpClient của người dùng để truy cập vào Socket của họ, User của họ và mã nhận dạng có thể tìm thấy ở (Guid).  


### 3.2.4 Class Message

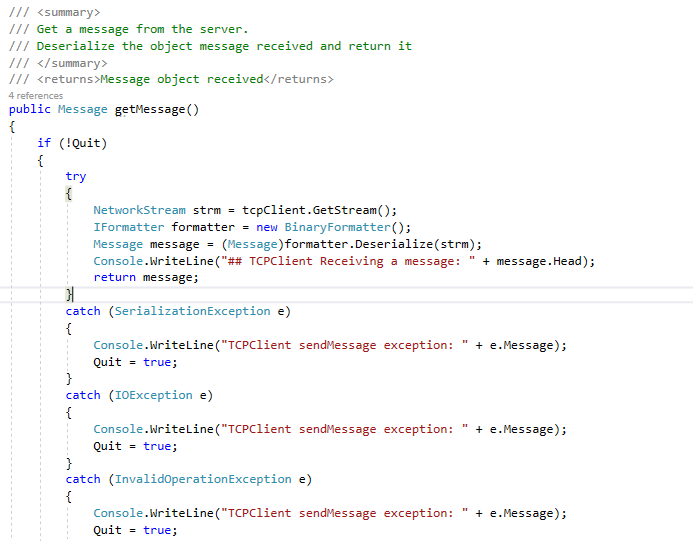
Theo chủ đề, Client và Server giao tiếp bằng cách trao đổi giữa các biến của Message Class. Lớp này chứa một tiêu đề, được chọn từ một danh sách được xác định trước:   
  
Ngoài tiêu đề, nó chứa một danh sách các chuỗi, chứa thông báo và thông tin được yêu cầu bởi Server và Client:  

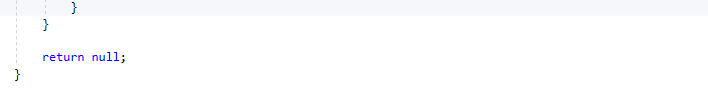

### 3.2.5 Class TcpClient

Lớp TcpClient cho phép Client của Client project kết nối tới Server.

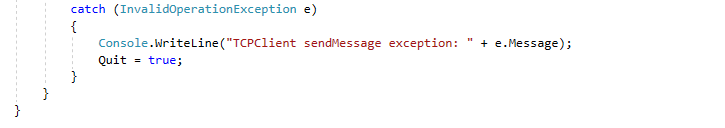
Nó khởi tạo một TcpClient được sử dụng (TcpClient này không phải là kiểu TcpClient dịnh dạng sẵn trong C#), và thu được khi kết nối với Server sử dụng IP và port(cổng) của nó.

Cuối cùng là nó cho phép bạn gửi tin và nhận tin nhắn. Để làm được điều đó, cần phải lấy luồng từ TcpClient và tuần tự hóa biến Message.





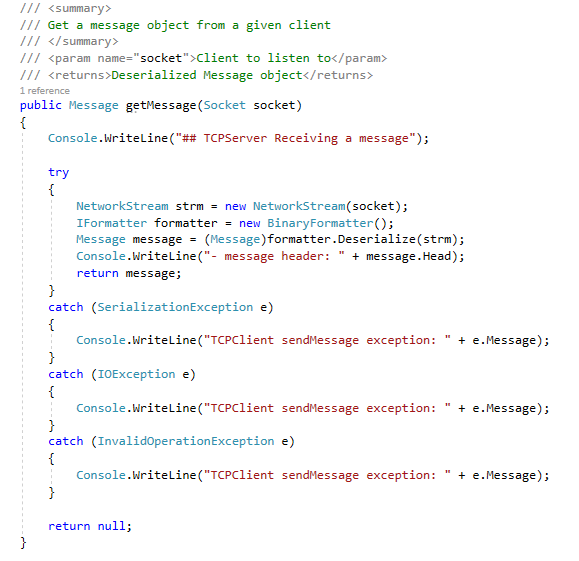


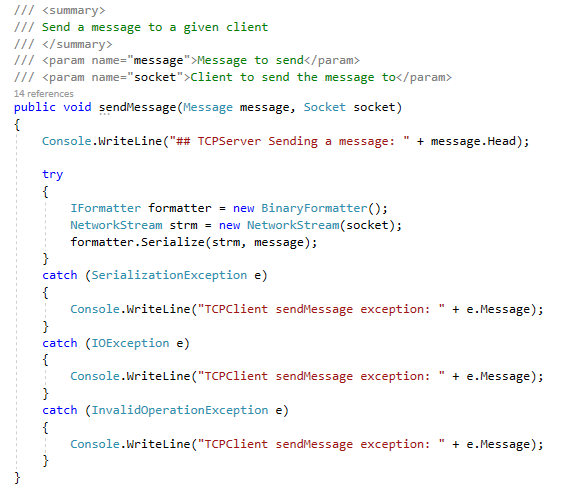


### 3.2.6 Class TcpServer

Class TcpServer khá tương tự nhưa TcpClient nhưng phức tạp hơn một chút. Do đó, TcpClient bị thay đổi thành TcpListener và xuất hiện 3 luồng (threads). Vấn đề nãy sẽ được nói đến ở Class Server.

Bạn có thể khởi động và dừng Server cũng như trước khi gửi hoặc nhận một tin nhắn.



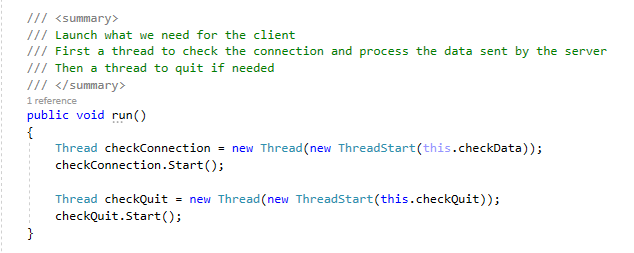


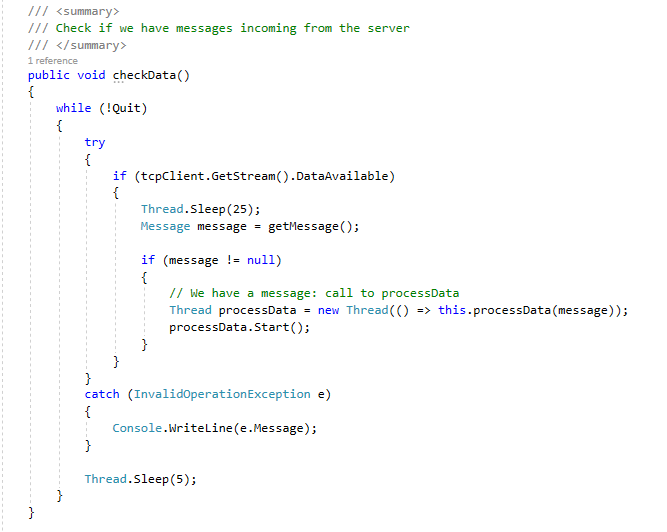
Mặc dù các chức năng của Class TcpServer tương tự Class TcpClient , dường như hai lớp không gắn kết với nhau để tạo ra một giao diện chung cho TcpClient và TcpServer bởi vì tín hiệu của chúng khác nhau. Server phải biết tại đó Socket(Client) nào gửi tin nhắn, cái nào không cần thiết ở bên phía Client nơi mà mọi người trò truyện với Server.

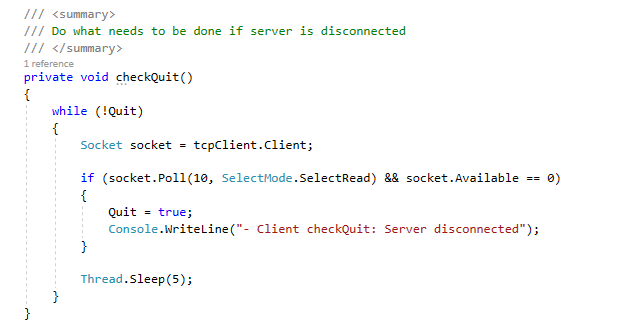
### 3.2.7 Class Client

Class Client được kế thừa từ TcpClient, nó bao gồm tất cả các logic của Client và được sử dụng trong Client WinForms.

Vai trò chính của Class Client là khởi chạy 2 luồng ( threads), checkConnection và checkQuit, kiểm tra sự xuất hiện các tin nhắn mới đến từ sever và kiểm tra xem Server vẫn còn online hay không.Miễn là người dùng vẫn mở cửa sổ Window và Server vẫn online thì nó vẫn tiếp tục hoạt động.







Để kiểm tra một tin nhắn có sẵn hay không, sử dụng getMessage được định nghĩa trong lớp cha, TcpClient. Nếu có một tin nhắn đến từ một hàm khác của lớp processData sẽ xử lý một cách thích hợp.





### 3.2.8 Class ThreadedBindingList

Để hiển thị danh sách, Client sử dụng ThreadedBindingList. Đây là một công cục class của BindingLists có thể được sử dụng trong một luồng. Chúng tôi chọn làm việc với BindingLists vì nó có vẻ cần thiết đối với chúng tôi để sử dụng danh sách cho phép "liên kết 2 chiều", nơi cập nhật danh sách từ Client class sẽ được WinForm nhìn thấy ngay lập tức mà không cần vòng lặp việc này sẽ xóa danh sách hiện tại sau mỗi X giây trước khi thay thế nó bằng một cái mới, đối với chúng tôi dường như là vô nghĩa khi các thành phần phù hợp tồn tại. Trong Client, những danh sách này được sử dụng để lưu trữ những người dùng được kết nối với phòng trò chuyện các phòng trò chuyện hiện có, cũng như các tin nhắn nhận được từ Server.

### 3.2.9 Class Chat

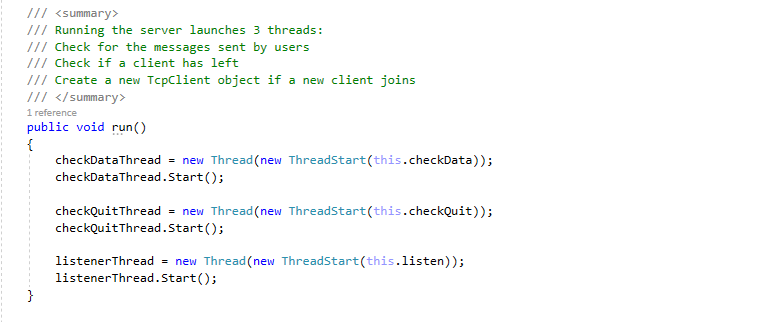
Đây là quan điểm chính của khách hàng. Các WinForms khác tồn tại (kết nối, nhận dạng) nhưng chúng không được quan tâm nhiều. Ba luồng (threads) được sử dụng cho:   
- Định kỳ truy xuất danh sách các phòng chat có sẵn bằng cách hỏi khách hàng gửi một tin nhắn LIST\_CR - Định kỳ truy xuất danh sách người dùng đã kết nối với phòng trò chuyện(chatroom) hiện tại bằng cách yêu cầu Client gửi LIST\_USERS tin nhắn  
 - Kiểm tra định kỳ xem Server vẫn được kết nối với boolean customer.Quit   
Phần còn lại của các chức năng được sử dụng để code hoạt động của các nút: nhấp vào nút để thêm phòng trò chuyện, quản lý combobox để chọn phòng trò chuyện, bấm để gửi tin nhắn.

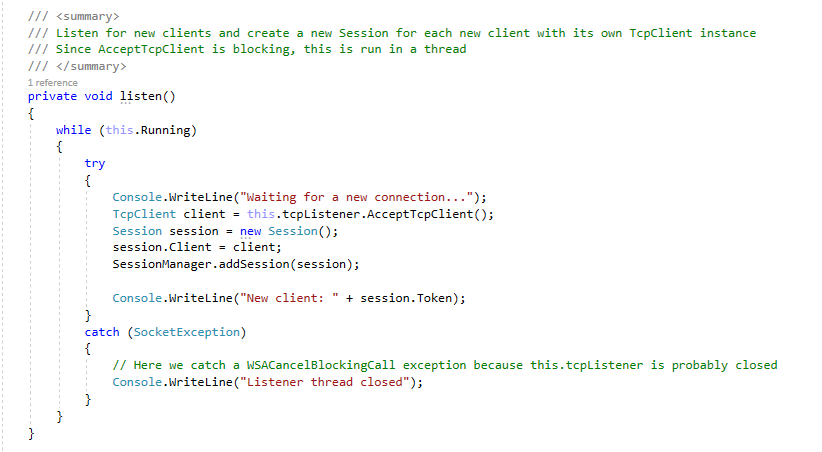
Các luồng có thời gian chờ 2 giây trong vòng lặp của chúng để tránh quá tải Server một cách không cần thiết, vì đây là dữ liệu không hoàn toàn cần thiết phải có trong thời gian thực.   
Như trong tất cả các WinForms của dự án, chúng tôi phải cẩn thận để gọi chuỗi của Main WinForm để sửa đổi các phần tử của nó vì chúng không thể sửa đổi từ một luồng bên ngoài.

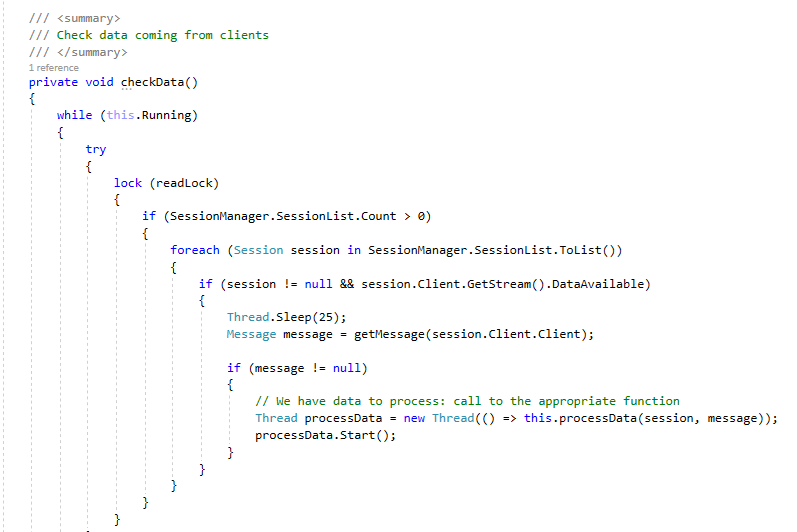
### 3.2.10 Class Server

Hoạt động của Server class tương tự như Client client ở trên. Nó cho phép bạn quản lý máy chủ bằng cách kế thừa từ TCPServer. Đầu tiên, trong hàm tạo của nó, chúng tôi khởi tạo tất cả Managers ( Users, Chatroom và Session) và chúng tôi khôi phục dữ liệu được lưu trữ trong tệp tĩnh, sau đó chúng tôi khởi chạy ba luồng (threads) để:  
 - Lắng nghe nếu chúng tôi có kết nối mới để tạo socket của họ với AcceptTcpClient và vùng của họ.  
 - Kiểm tra xem nếu có tin nhắn mới từ khách hàng

- Kiểm tra xem nếu khách hàng đã rời khỏi Server

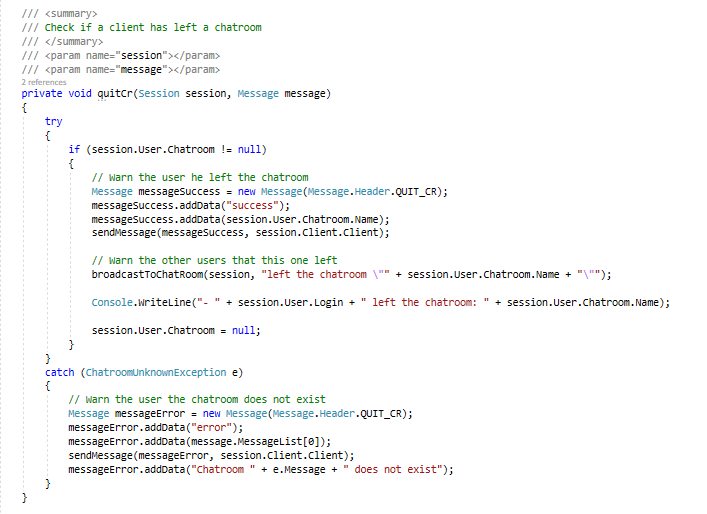


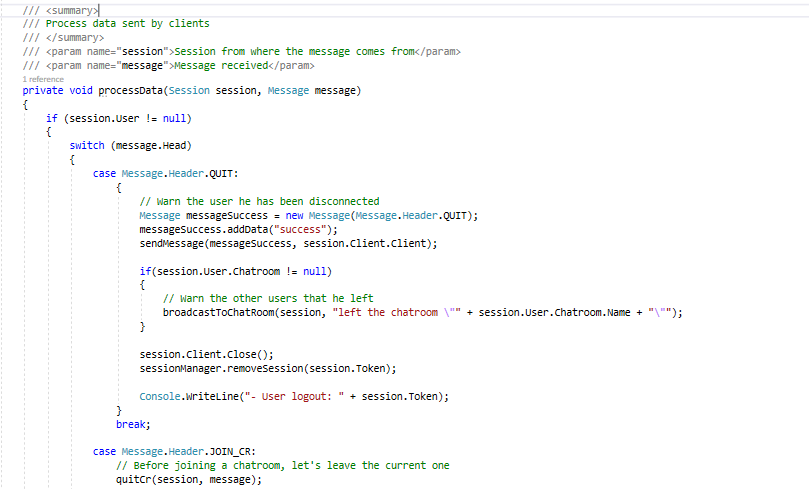






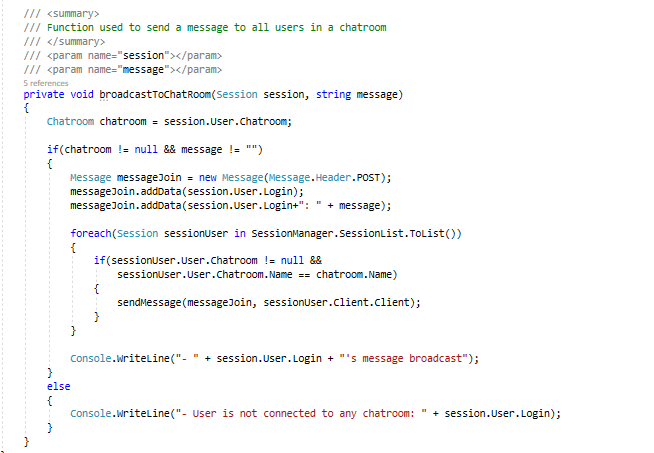








Hàm processData, phản hồi với Clienta, gửi tin nhắn trực tiếp đến chúng bằng sendMessage từ TCPServer hoặc gửi tin nhắn đến tất cả các Clients được kết nối với cùng một chatroom với chức năng broadcastToChatroom:



Do đó, chúng tôi sử dụng các vùng được lưu trữ trong SessionManager để xác định những người dùng mà chúng tôi cần áp dụng sendMessage.

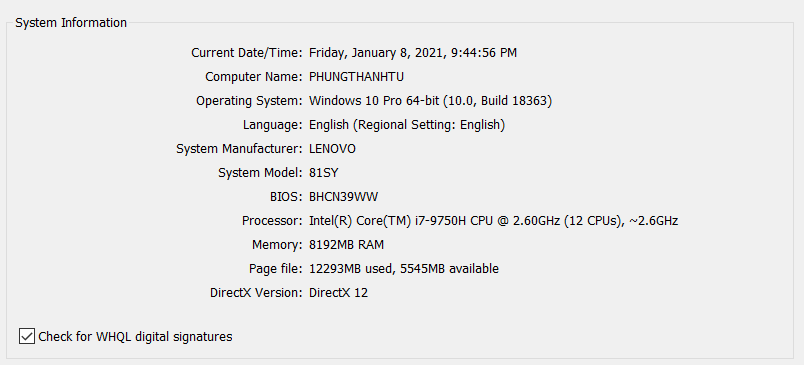
### 3.2.11 Class ServerGUI và TextBoxStreamWriter

Cuối cùng, Server project là dự án trò chuyện đơn giản nhất. Nó chỉ là một vấn đề đặt cổng để bạn muốn khởi động Server và khởi động hoặc dừng Server bằng cùng một nút. Tin nhắn từ Server được hiển thị trong textbox. Chúng tôi đã sử dụng lớp học TextBoxStreamWriter để chuyển hướng văn bản được gửi đến bảng điều khiển trong WinForm. Nó rất cần thiết hãy cẩn thận để văn bản này được gửi chính xác, vì nó có thể được gửi từ các luồng khác nhau.

# Chương 4: Thử nghiệm đánh giá

## 4.1 Môi trường thử nghiệm

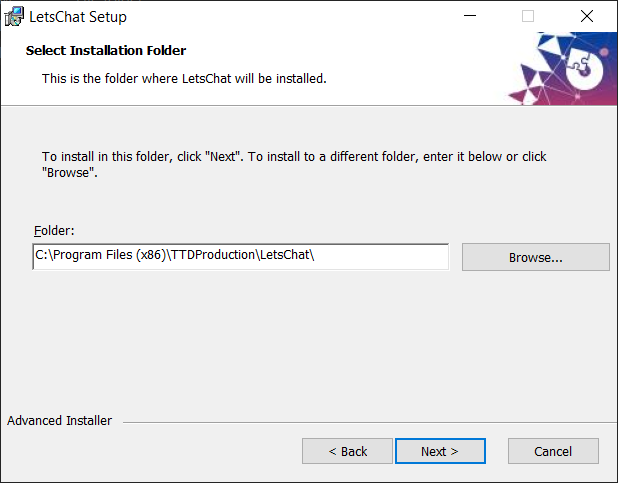
- Phần cứng máy tính :



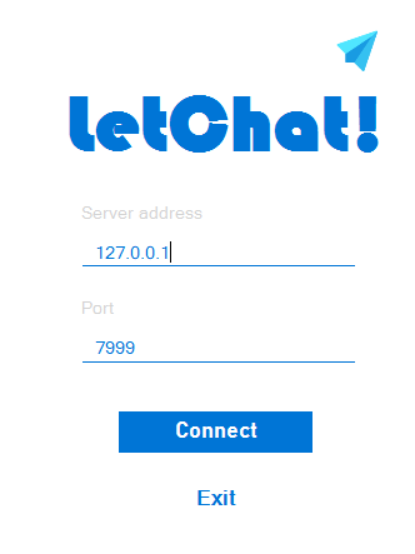
* + IDE : Visual Studio 2019 Build 16.8.2
  + Core : .NET Core 2.1 Runtime
  + Framework : .NET 4.5.2
  + Mạng Wifi FPT

## 4.2 Kết quả thu được

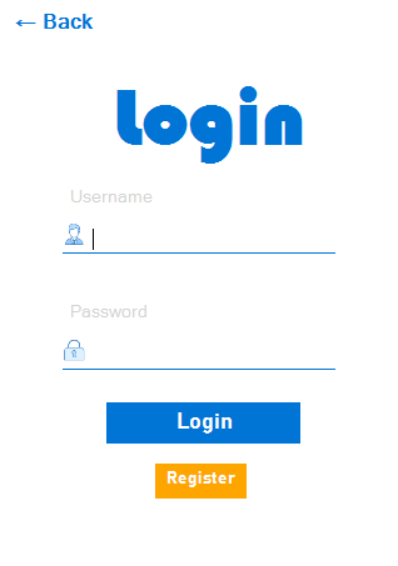
- Giao diện Installer



- Giao diện Connect Server



- Giao diện Login



- Giao diện Chat



# Chương 5: Kết luận và hướng phát triển

## 5.1 Những hạn chế của ứng dụng

- Giao diện chat chưa hoàn thiện, vẫn còn là dạng Textbox

- Chưa có chức năng Nghe/Gọi

- Chưa có chức năng gửi hình ảnh

- Chưa có chức năng thông báo đã gửi/ đã xem/ đang gõ

- Vẫn còn nhiều lỗi liên quan đến Thread và Socket

## 5.2 Hướng phát triển

- Thêm chức năng Gửi hình/nhạc/file

- Thêm chức năng Gọi

- Thêm chức năng Thông báo

- Thêm chức năng gửi icon

## 5.3 Những kiến thức thú vị đã rút ra được

- Lập trình mạng Socket

- Lập trình trực quan với WinForm

- Quản lí Session với Class Guid

- Sự dụng công cụ quản lí mã nguồn mở GitHub

## 5.4 Nguồn