**TRƯỜNG ĐẠI HỌC PHENIKAA**

**KHOA KHOA HỌC CƠ BẢN**

⸎⸎⸎⸎⸎

**Logo, company name

Description automatically generated**

**BÀI TẬP LỚN KẾT THÚC HỌC PHẦN MÔN LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG**

Đề tài: Lập chương trình truyền file và chat cho giao thức TCP sử dụng java

Giảng viên: Trần Đăng Hoan

NHÓM 5

Thành viên nhóm:

Hoàng Gia Bảo – 20010845

Nguyễn Nam Khánh – 20010869

Bùi Đức Tuấn Anh – 20010838

Nguyễn Đăng Trường Giang – 20010855

Hà Nội, Tháng 6 Năm 2022

MỤC LỤC

TỔNG QUAN ĐỀ TÀI

CHƯƠNG I Tổng quan về đề tài

1.Tổng quan về ngôn ngữ Java

2. Giao thức TCP/IP

3. Mô hình Client/Server

4. Cơ chế socket trong Java

5. Server hoạt động đa tuyến

6. Môi trường và công cụ lập trình

7. Ứng dụng OOP Java vào đề tài

CHƯƠNG II Thiết kế và xây dựng mô hình bài toán

1. Phân tích yêu cầu

2. Phân tích chức năng

3. Thiết kế chương trình

4. Xây dựng các chức năng

CHƯƠNG III Triển khai và đánh giá kết quả

1. Môi trường triển khai

2. Kết quả các chức năng của chương trình

3. Kết quả và hướng phát triển

**TỔNG QUAN VỀ ĐỀ TÀI**

1. Bối cảnh và lý do thực hiện đề tài

Ngày nay, nhu cầu về công nghệ thông tin trong đời sống là đa dạng.Việc mở rộng các hệ thống truyền thông và ngày càng có nhiều máy vi tính kết nối vào mạng Internet. Với việc ứng dụng giao thức TCP làm cho hệ thống mạng ngày càng rộng hơn và phát triển vượt bậc. Vấn đề đề an ninh, bảo mật,…là một thế mạnh của giao thức này đem lại cho công nghệ truyền thông. Bên cạnh việc phát triển của thư điện tử bằng nhiều dịch vụ khác nhau(gmail, yahoo trong mạng điện thoại di động…), việc chat trực tuyến và gửi flie trực tiếp trên internet cũng là nhu cầu không thể thiếu. Với thực tế như vậy, nhóm chúng em đã nghiên cứu và xây dựng chương trình truyền file cho giao thức TCP

1. Giao thức TCP tồn tại kênh giao tiếp ảo giữa hai quá trình, dữ liệu sẽ được gửi đi theo chế độ bảo đảm, có kiểm tra lỗi, truyền lại gói tin lỗi hay mất, bảo đảm thứ tự truyền đến của các gói tin, có dữ liệu chính xác, tốc độ truyền tin chậm, thích hợp cho các ứng dụng cần độ chính xác cao.
2. Nói một cách đơn giản là chúng em sẽ tạo những giao diện để cho client kết nối tới sever máy chủ, sau khi kết nối vào sever thành công thì các user có thế thực hiện chat và truyền file vs nhau, sau đó có thể kết thúc giao thức chat và đóng kết nối tới sever
3. Đề tài nhóm em sẽ sử dụng ngôn ngữ lập trình hướng đối tượng trong Java để tạo ra các đối tượng client, sever với những chức năng đăng nhập kết nối và cho phép các đối tượng thực hiện chat vs nhau.
4. Ngoài ra tạo giao diện đơn giản, bố cục gọn gàng với đầy đủ chức năng nêu trên
5. Phương thức triển khai đề tài

Ngôn ngữ lập trình: Java(OOP)

Chương trình soạn thảo và build: Eclipse IDE

Sử dụng giao thức TCP

Sử dụng Socket trong Java

1. Kết cấu của bài tập lớn

Gồm ba chương:

Chương 1: Tổng quát lý thuyết

Chương 2: Thiết kế và xây dựng mô hình bài toán

Chương 3: Triển khai và đánh giá kết quả

**CHƯƠNG 1:Tổng quan lý thuyết**

1.Tổng quan về ngôn ngữ Java

- Java là một ngôn ngữ lập trình hướng đối tượng do Sun MicroSystem đưa ra vào giữa thập niên 90.

- Chương trình viết bằng ngôn ngữ lập trình java có thể chạy trên bất kỳ hệ thống nào có cài máy ảo Java (Java Virtual Machine). Ngày nay Java được sử dụng với các mục đích sau:

- Phát triển ứng dụng cho các thiết bị điện tử thông minh, các ứng dụng cho doanh nghiệp với quy mô lớn.

- Tạo các trang web có nội dung động (web applet), nâng cao chức năng của server.

- Phát triển nhiều loại ứng dụng khác nhau: Cơ sở dữ liệu, mạng, Internet, viễn thông, giải trí,... Về hướng đối tượng

- Mạnh mẽ (thể hiện ở cơ chế tự động thu gom rác – Garbage Collection) và an toàn.

- Kiến trúc trung lập, độc lập nền tảng và có tính khả chuyển (Portability).

- Hiệu suất cao.

- Máy ảo (biên dịch và thông dịch).

- Phân tán.

- Đa nhiệm: Ngôn ngữ Java cho phép xây dựng trình ứng dụng, trong đó nhiều quá trình có thể xảy ra đồng thời. Tính đa nhiệm cho phép các nhà lập trình có thể biên soạn phần mềm đáp ứng tốt hơn, tương tác tốt hơn và thực hiện theo thời gian thực.

2. 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.

Table

Description automatically generated

Hình 1: Sơ đồ TCP/IP

2.1 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 thường 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. Phiên bản 5 được dùng làm giao thức dòng (stream) thử nghiệm. Còn có các phiên bản khác, nhưng chúng thường dành là các giao thức thử nghiệm và không được sử dụng rộng rãi.

Địa chỉ IP được chia thành 4 số giới hạn từ 0 - 255. Mỗi số được lưu bởi 1 byte - > IP có kích thước là 4byte, được chia thành các lớp địa chỉ. Có 3 lớp là A, B, và C. Nếu ở lớp A, ta sẽ có thể có 16 triệu điạ chỉ, ở lớp B có 65536

địa chỉ. Ví dụ: Ở lớp B chúng ta có tất cả các địa chỉ từ 132.25.0.0 đến 132.25.255.255. Phần lớn các địa chỉ ở lớp A là sở hữu của các công ty hay của tổ chức. Một ISP thường sở hữu một vài địa chỉ lớp B hoặc C. Ví dụ: Nếu địa chỉ IP của bạn là 132.25.23.24 thì bạn có thể xác định ISP của bạn là ai.

3. Mô hình Client/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.

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:

+ Truyền một yêu cầu từ tiến trình client tới tiến trình server

+ Yêu cầu được server xử lý

+ Truyền đáp ứng cho client

1. Cơ chế socket trong Java

4.1. Khái quát về Socket

Như chúng ta đã biết kết nối URLs và URL cung cấp cho chúng ta một cơ cấu để truy xuất vào các tài nguyên trên Internet ở một mức tương đối cao, nhưng đôi khi chương trình của chúng ta lại yêu cầu một giao tiếp ở tầng mạng mức thấp.Ví dụ khi chúng ta viết một ứng dụng client-server. Trong một ứng dụng client-server thì phía server sẽ cung cấp một số dịch vụ, như: xử lí cơ sở dữ liệu, các yêu cầu bên phía client đưa ra, sau đó sẽ gửi lại cho phía client. Sự giao tiếp như vậy gọi là tin cậy bởi vì dữ liệu sẽ không bị mất mát, sai lệch trong quá trình truyền, server gửi cho client thông điệp gì thì phía client sẽ nhận được thông điệp nguyên như vậy. Giao thức TCP sẽ cung cấp cho chúng ta một cách thức truyền tin cậy. Để có thể nói chuyện được trên TCP thì chương trình client và chương trình server phải thiếp lập một đường truyền, và mỗi chương trình sẽ phải kết nối lại với socket là điểm cuối để kết nối, client và server muốn nói chuyện với nhau thì sẽ phải thông qua socket, mọi thông điệp sẽ phải đi qua socket. Chúng ta cứ mường tượng socket ở đây là một cái cửa mọi người muốn đi ra hay đi vào đều phải thông qua cái cửa này.

* 1. Cơ chế Socket

Socket là một điểm cuối của thông tin hai chiều liên kết giữa hai chương trình đang chạy trên mạng. Những lớp socket được dùng để đại diện cho kết nối giữa một chương trình client và một chương trình server. Trong Java gói Java.net cung cấp hai lớp Socket và ServerSocket để thực hiện kết nối giữa client và server. Thông thường thì server sẽ chạy trên một máy đặc biệt và có một socket giới hạn trong 1 Portnumber đặc biệt. Phía client: client được biết hostname của máy mà server đang chạy và  port number mà server đang lắng nghe. Để tạo một yêu cầu kết nối client sẽ thử hẹn gặp server ở trên máy của server thông qua port number. Client cũng cần xác định chính nó với server thông qua local port number.

A picture containing diagram

Description automatically generated

Hình 2: Server “lắng nghe” yêu cầu kết nối của client

Nếu mọi thứ tốt đẹp thì server sẽ đồng ý kết nối. khi đồng ý kết nối thì server sẽ tạo ra một socket mới để nói chuyện với client và cũng tạo ra một socket khác để tiếp tục lắng nghe.

Chart, diagram, line chart

Description automatically generated

Hình 3: Server đồng ý kết nối và tiếp tục lắng nghe.

Một socket là một điểm cuối của thông tin hai chiều liên kết giữa hai chương trình đang chạy trên mạng. Những lớp socket được dùng để đại diện cho kết nối giữa một chương trình client và một chương trình server. Thực hiện kết nối giữa client và server.

Diagram

Description automatically generated

Hình 4: Mô hình truyền tin socket

Table

Description automatically generated with low confidence

Hình 5: Mô hình tương tác giữa client/server qua socket TCP

* Một socket có thể thực hiện bảy thao tác cơ bản:

Kết nối với một máy ở xa (ví dụ, chuẩn bị để gửi và nhận dữ liệu)

Gửi dữ liệu

Nhận dữ liệu

Ngắt liên kết

Gán cổng

Nghe dữ liệu đến

Chấp nhận liên kết từ các máy ở xa trên cổng đã được gán

A picture containing graphical user interface

Description automatically generated

Hình 6: Mô hình kết nối giữa Client/Server qua socket TCP

5. Server hoạt động đa tuyến(multi threading)

Để cho phép chương trình server kết nối và phục vụ được nhiều chương trình client tại một thời điểm, chúng ta cần phải xây dựng chương trình server là chương trình đa tuyến. Trong đó, mỗi tuyến (thread) ở chương trình server sẽ giữ một kết nối với một chương trình client và đảm nhận việc truyền nhận dữ liệu với client đó. Khi có một chương trình client gửi yêu cầu kết nối đến chương trình server thì chương trình server phải tạo ra một tuyến (thread) để kết nối và trao đổi dữ liệu với client đó.

6. Môi trường và công cụ cài đặt

Eclipse IDE là gì?

Eclipse IDE là một môi trường phát triển tích hợp (IDE) cho Java và các ngôn ngữ lập trình khác như C, C ++, PHP, và Ruby ... Môi trường phát triển được cung cấp bởi Eclipse bao gồm các công cụ phát triển Java Eclipse (JDT) cho Java, Eclipse CDT cho C/C ++, và Eclipse PDT cho PHP, và một số thứ khác.

Ban đầu Eclipse được tạo ra để phát triển Java IDE, tuy nhiên Eclipse Foundation hiện đang phát triển một loạt các công cụ phát triển hỗ trợ nhiều ngôn ngữ lập trình:  C/C++, PHP, Javascript, Python, Rust. Tuy nhiên nó được mọi người biết và sử dụng nhiều nhất để phát triển các ứng dụng Java.

Eclipse là ứng dụng mã nguồn mở và miễn phí, có nghĩa là bạn có thể sử dụng nó miễn phí và tham khảo mã nguồn của nó nếu cần. Ngày nay, Eclipse là IDE được sử dụng rộng rãi nhất để phát triển các ứng dụng Java, với hàng triệu lập trình viên sử dụng mỗi ngày.

JDK là gì?

[Java Development Kit](https://bizflycloud.vn/tin-tuc/jdk-la-gi-2018121318321223.htm) (JDK) là một trong ba gói công nghệ cốt lõi được sử dụng trong lập trình Java, cùng với JVM (Máy ảo Java - Java Virtual Machine) và JRE (Java Runtime Environment - Môi trường Java Runtime). Việc phân biệt giữa ba công nghệ này, cũng như hiểu được cách chúng kết nối với nhau là rất quan trọng.

- JVM là thành phần Java nền tảng để chạy các chương trình.

- JRE là phần on-disk của Java, phần tạo ra JVM.

- JDK cho phép các developer tạo các chương trình Java, trong đó các chương trình có thể được JVM và JRE xử lý và chạy.

Các developer khi mới sử dụng Java thường dễ nhầm lẫn Java Development Kit với Java Runtime Environment. Để phân biệt, ta cần nhớ rằng JDK là một gói các công cụ phát triển phần mềm dựa trên Java, trong khi JRE là gói công cụ sử dụng để chạy Java code.

JRE có thể được sử dụng như một thành phần độc lập để chạy các chương trình Java, dù vậy, JRE vẫn là một phần của JDK. JDK đòi hỏi bộ công cụ JRE vì chạy các chương trình Java là một phần của việc phát triển chúng.

7. Ứng dụng OOP Java vào đề tài

Có bốn tính chất của lập trình hướng đối tượng trong Java

Tính đóng gói (encapsulation) và che giấu thông tin (information hiding): Tức là trạng thái của đối tượng được bảo vệ không cho các truy cập từ code bên ngoài như thay đổi trạng thái hay nhìn trực tiếp. Việc cho phép môi trường bên ngoài tác động lên các dữ liệu nội tại của một đối tượng theo cách nào là hoàn toàn tùy thuộc vào người viết mã. Đây là tính chất đảm bảo sự toàn vẹn, bảo mật của đối tượng trong Java, tính đóng gói được thể hiện thông qua phạm vi truy cập (access modifier). Ngoài ra, các lớp liên quan đến nhau có thể được gom chung lại thành package.

Tính kế thừa là khả năng cho phép ta xây dựng một lớp mới dựa trên các định nghĩa của một lớp đã có. Lớp đã có gọi là lớp cha, lớp mới phát sinh gọi là lớp con và đương nhiên kế thừa tất cả các thành phần của lớp cha, có thể chia sẻ hay mở rộng các đặc tính sẵn có mà không phải tiến hành định nghĩa lại.

Tính đa hình (polymorphism):Khi một tác vụ được thực hiện theo nhiều cách khác nhau được gọi là tính đa hình. Đối với tính chất này, nó được thể hiện rõ nhất qua việc gọi phương thức của đối tượng. Các phương thức hoàn toàn có thể giống nhau, nhưng việc xử lý luồng có thể khác nhau. Nói cách khác: Tính đa hình cung cấp khả năng cho phép người lập trình gọi trước một phương thức của đối tượng, tuy chưa xác định đối tượng có phương thức muốn gọi hay không. Đến khi thực hiện (run-time), chương trình mới xác định được đối tượng và gọi phương thức tương ứng của đối tượng đó. Kết nối trễ giúp chương trình được uyển chuyển hơn, chỉ yêu cầu đối tượng cung cấp đúng phương thức cần thiết là đủ.

Trong Java, chúng ta sử dụng nạp chồng phương thức (method overloading) và ghi đè phương thức (method overriding) để có tính đa hình.

Nạp chồng (Overloading): Đây là khả năng cho phép một lớp có nhiều thuộc tính, phương thức cùng tên nhưng với các tham số khác nhau về loại cũng như về số lượng. Khi được gọi, dựa vào tham số truyền vào, phương thức tương ứng sẽ được thực hiện.

Ghi đè (Overriding): là hai phương thức cùng tên, cùng tham số, cùng kiểu trả về nhưng thằng con viết lại và dùng theo cách của nó, và xuất hiện ở lớp cha và tiếp tục xuất hiện ở lớp con. Khi dùng override, lúc thực thi, nếu lớp con không có phương thức riêng, phương thức của lớp cha sẽ được gọi, ngược lại nếu có, phương thức của lớp con được gọi.

Tính trừu tượng là một tiến trình ẩn các chi tiết triển khai và chỉ hiển thị tính năng tới người dùng. Tính trừu tượng cho phép bạn loại bỏ tính chất phức tạp của đối tượng bằng cách chỉ đưa ra các thuộc tính và phương thức cần thiết của đối tượng trong lập trình. Tính trừu tượng giúp bạn tập trung vào những cốt lõi cần thiết của đối tượng thay vì quan tâm đến cách nó thực hiện. Trong Java, chúng là sử dụng abstract class và abstract interface để có tính trừu tượng.

**CHƯƠNG 2: Thiết kế và xây dựng mô hình bài toán**

1.Phân tích yêu cầu

1.1 Yêu cầu của bài toán: Xây dựng mô hình Client – Server ứng dụng chat

1.2 Xây dựng chương trình Server

- Tạo một TCP Socket và gắn vào một cổng

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

- Chờ và lắng nghe yêu cầu kết nối từ Client.

- Chấp nhận kết nối và nhận Socket tương ứng.

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

- Khi một user login vào Server thì Server sẽ cập nhật user đó thì server tự động hiển thị user.

- Khi một user logout ra khỏi Server thì Server tự động xóa user.

- Đóng kết nối.

1.3 Xây dựng chương trình Client.

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

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

- Cập nhật các user khác vừa login/logout.

- Gửi/ nhận thông điệp với user có trong phòng chat.

- Đóng kết nối

1.4 Thiết kế giao diện hiển thị khung chat phía Client.

Yêu cầu giao diện:

- Chat trực tiếp giữa Client với Client thông qua Server khi được kết nối.

- Trao đổi dữ liệu với nhau nhờ vào Stream nhập và Stream xuất.

Phân tích các chức năng:

- Chức năng của chương trình được thể hiện qua chương trình Client và Server.

- Chức năng chat trong cùng 1 mạng

- Chuỗi ký tự nhập từ bàn phím.

- Ấn mũi tên để gửi chuỗi ký tự vừa nhập để giao tiếp với nhau giữa Client và Client thông qua Server.

1.5 Xây dựng chương trình Client: nối kết đến Server trong mạng để trao đổi thông tin và truyền tải dữ liệu.

1.6 Xây dựng chương trình Server đa tuyến để kết nối với nhiều Client: xử lý các yêu cầu kết nối: lắng nghe yêu cầu nối kết của Client.

Xử lý các thông điệp yêu cầu từ Client: chấp nhận hoặc từ chối kết nối.

2. Phân tích chức năng

- Chức năng của chương trình được thể hiện qua chương trình Client và Server.

- Chức năng Chat trong mạng WAN:

- Chuỗi ký tự nhập từ bàn phím.

- Ấn nút “Send” để gửi chuỗi ký tự vừa nhập để giao tiếp với nhau giữa Client và Server.

- Xây dựng chương trình Client kết nối đến Server trong mạng để trao đổi thông tin và truyền tải dữ liệu.

- Xây dựng chương trình Server đa tuyến để kết nối với nhiều Client:

- Xử lý các yêu cầu nối kết: lắng nghe yêu cầu nối kết của Client.

- Xử lý các thông điệp yêu cầu từ Client: chấp nhận hoặc từ chối kết nối.

3. Thiết kế chương trình

Thiết kế giao diện

Graphical user interface, application

Description automatically generated

Hình 7: Giao diện server

Graphical user interface, application

Description automatically generated

Hình 8: Giao diện Login cho phép client kết nối tới sever

IP Address Server: địa chỉ IP của Server (mặc định là địa chỉ localhost).

Port : số hiệu cổng kết nối

Your Name: tên user để kết nối

4. Xây dựng các chức năng

Xây dựng chương trình Client(sử dụng lớp java.net.Socket):

- Mở một socket nối kết đến Server đã biết địa chỉ IP/localhost và số hiệu cổng(8080).

- Lấy Stream nhập và Stream xuất được gán với socket.

- Trao đổi dữ liệu với Server nhờ vào các Stream nhập và Stream xuất. Dùng giao thức TCP/IP để kiểm tra dữ liệu trao đổi với Server.

- Xây dựng chương trình Server đa tuyến (Server phải luôn giữ kết nối) xử lý các yêu cầu kết nối:

- Lắng nghe yêu cầu kết nối. Chấp nhận một yêu cầu kết nối.

- Tạo kênh giao tiếp ảo với các Client. Chờ nhận phân tích và xử lý yêu cầu từ Client.

- Gửi thông điệp trả lời cho Client.

**CHƯƠNG 3: Triển khai và đánh giá kết quả**

1. Môi trường triển khai

Hệ điều hành Microsoft Windowns 10,11

Môi trường lập trình: Eclipse IDE, JDK

Mạng WAN, LAN

2. Kết quả các chức năng của chương trình

Graphical user interface, application, Word

Description automatically generated

Hình 9: Kết nối lỗi tới sever

Graphical user interface, application

Description automatically generated

Hình 10: Chạy chương trình server

Graphical user interface, text, application, chat or text message

Description automatically generated

Hình 11: Giao diện chat giữa hai client

3. Kết quả và hướng phát triển

- Những kết quả đạt được:

1. Màn hình Server với đủ thông tin: config, đóng / mở server, danh sách client đang kết nối.

2. Màn hình Client đủ thông tin về các chức năng cơ bản của chương trình: danh sách server để kết nối và thông tin của server nếu đang kết nối, danh sách các user đang online.

3. Cho phép quản lý danh sách server cho client như: thêm, xóa, sửa và ghi nhớ dạng file config.

4. Đăng ký chat user (đăng ký từ ứng dụng client).

5. Khung chat với đầy đủ thông tin người chat, nội dung chat.

6. Cho phép lựa chọn ENTER là gửi nội dung chat hoặc xuống dòng (cho phép nội dung chat nhiều dòng).

7. Cho phép sử dụng emoji đơn giản .

8. Cho phép chat với nhiều user (chia nhiều tab).

9. Cho phép gởi file trong khi chat.

- Hướng phát triển

Hoàn thiện giao diện đẹp và phù hợp với người dùng.

Mở rộng ứng dụng trong chương trình Chat như :

+ Gửi voice chat, call video

+ Xây dựng hệ cơ sở dữ liệu quản lý thông tin đăng nhập ( username, password …)

+ Phát triển mạnh trên mạng WAN,LAN.