TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM

**TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG**

**KHOA CÔNG NGHỆ THÔNG TIN**

****

**ĐỒ ÁN CUỐI KÌ MÔN LẬP TRÌNH MẠNG CĂN BẢN**

**Xây dựng chương trình theo yêu cầu**

*Người hướng dẫn*: **TS BÙI QUY ANH**

*Người thực hiện*: **HUỲNH HỮU HIỆP – 51800677**

Lớp **: 18050402**

Khoá  **: 22**

**THÀNH PHỐ HỒ CHÍ MINH, NĂM 2020**

TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM

**TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG**

**KHOA CÔNG NGHỆ THÔNG TIN**

****

**ĐỒ ÁN CUỐI KÌ MÔN LẬP TRÌNH MẠNG CĂN BẢN**

**Xây dựng chương trình theo yêu cầu**

Người hướng dẫn: **TS BÙI QUY ANH**

Người thực hiện: **HUỲNH HỮU HIỆP**

Lớp **: 18050402**

Khoá  **: 22**

**THÀNH PHỐ HỒ CHÍ MINH, NĂM 2020**

LỜI CẢM ƠN

Tôi chân thành cảm ơn giảng viên hướng dẫn của tôi là thầy Bùi Quy Anh. Người đã hướng dẫn trên lớp cho tôi và truyền đạt khá nhiều kiến thức bổ ích cho tôi trong suốt quá trình học tập cũng như giải đáp tất cả các thắc mắc mà tôi gặp phải. Và hơn thế nữa thầy còn là người cung cấp cho tôi nguồn tài liệu bổ ích cho chúng tôi. Từ đó, đó là nguồn kiến thức để tôi có thể nắm vững và đọc thêm một số tài liệu mà thầy cung cấp để có thể hoàn thành bài báo cáo như ngày hôm nay một cách hoàn thiện nhất. Qua đây, mong thầy có những nhận xét, góp ý để bài báo cáo có thể hoàn thiện tốt nhất.

Một lần nữa tôi xin gửi lời cảm ơn chân thành nhất!

**ĐỒ ÁN ĐƯỢC HOÀN THÀNH**

**TẠI TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG**

Tôi xin cam đoan đây là sản phẩm đồ án của riêng tôi và được sự hướng dẫn của TS Bùi Quy Anh. Các nội dung nghiên cứu, kết quả trong đề tài này là trung thực và chưa công bố dưới bất kỳ hình thức nào trước đây. Những số liệu trong các bảng biểu phục vụ cho việc phân tích, nhận xét, đánh giá được chính tác giả thu thập từ các nguồn khác nhau có ghi rõ trong phần tài liệu tham khảo.

Ngoài ra, trong đồ án còn sử dụng một số nhận xét, đánh giá cũng như số liệu của các tác giả khác, cơ quan tổ chức khác đều có trích dẫn và chú thích nguồn gốc.

**Nếu phát hiện có bất kỳ sự gian lận nào tôi xin hoàn toàn chịu trách nhiệm về nội dung đồ án của mình.** Trường đại học Tôn Đức Thắng không liên quan đến những vi phạm tác quyền, bản quyền do tôi gây ra trong quá trình thực hiện (nếu có).

*TP. Hồ Chí Minh, ngày tháng năm*

*Tác giả*

*(ký tên và ghi rõ họ tên)*

*Huỳnh Hữu Hiệp*

PHẦN XÁC NHẬN VÀ ĐÁNH GIÁ CỦA GIẢNG VIÊN

**Phần xác nhận của GV hướng dẫn**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Tp. Hồ Chí Minh, ngày tháng năm

(kí và ghi họ tên)

**Phần đánh giá của GV chấm bài**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Tp. Hồ Chí Minh, ngày tháng năm

(kí và ghi họ tên)

TÓM TẮT

Bài báo cáo cuối kỳ sẽ nói đến việc xây dựng chương trình dựa trên việc ứng dụng lập trình Javaswing để xây dựng các chương trình cụ thể. Để làm rõ điều đó bài báo cáo này sẽ trình bày tổng quát một cách đầy đủ về việc xây dựng chương trình dựa trên nó. Bài báo cáo gồm ba chương sẽ giải thích một cách đầy đủ nhất. Chương đầu tiên sẽ giới thiệu tổng quát về TCP/IP , mô hình Client/Server và Socket bao gồm các khái niệm các gói tin và các chức năng chính và đặc biệt là mô hình Client/Server sẽ ảnh hưởng trực tiếp đến việc xây dựng chương trình dựa trên nó. Và phần sẽ là phần mô phỏng việc chương trình ứng dụng được xây dựng dựa trên mô hình Client/Server đó là: Xây dựng ứng dụng chat giữa các Client đến Server thông qua Socket bao gồm quá trình hướng dẫn sử dụng cũng như hiển thị giao diện khi chat với nhau, thêm đó là ứng dụng máy tính dựa trên mô hình Client/Server bằng cách Client gửi dữ liệu lên Server và Server trả lại kết quả tính toán. Không những thế để đồ án tốt hơn chương thứ hai tôi thêm vào đó ứng dụng truyền file từ Client lên Server thông qua ý tưởng cơ bản thông qua việc học tập gửi file hàng ngày trong cuộc sống. Chương cuối cùng là tổng kết lại nội dung của vấn đề các thành tựu cũng như hạn chế.

MỤC LỤC

[LỜI CẢM ƠN i](#_Toc57576425)

[PHẦN XÁC NHẬN VÀ ĐÁNH GIÁ CỦA GIẢNG VIÊN iii](#_Toc57576426)

[TÓM TẮT iv](#_Toc57576427)

[MỤC LỤC 1](#_Toc57576428)

[DANH MỤC CÁC BẢNG BIỂU, HÌNH VẼ, ĐỒ THỊ 4](#_Toc57576429)

[CHƯƠNG 1 – PHẦN BẮT BUỘC 6](#_Toc57576430)

[1.1 Tổng quan 6](#_Toc57576431)

[1.1.1 Giao thức TCP/IP 6](#_Toc57576432)

[1.1.2 Mô hình Client/Server 7](#_Toc57576433)

[1.1.3 Cơ chế Socket 11](#_Toc57576434)

[1.2 Yêu cầu 1: Xây dựng chương trình Chat giữa Client với Server theo yêu cầu 12](#_Toc57576435)

[1.2.1 Lý thuyết liên quan 12](#_Toc57576436)

[1.2.2 Biểu đồ tổng quát hệ thống 13](#_Toc57576437)

[1.2.3 Biểu đồ hoạt động của Server 15](#_Toc57576438)

[1.2.4 Biểu đồ hoạt động của Client 15](#_Toc57576439)

[1.2.5 Demo chương trình 16](#_Toc57576440)

[1.3 Yêu cầu 2: Xây dựng chương trình Client/Server thực hiện phép tính (cộng, trừ, nhân, chia) theo yêu cầu 29](#_Toc57576441)

[1.3.1 Xây dựng chương trình 29](#_Toc57576442)

[1.3.2 Vẽ và thiết kế hệ thống 29](#_Toc57576443)

[1.3.3 Demo chương trình 32](#_Toc57576444)

[CHƯƠNG 2 – PHẦN Ý TƯỞNG 38](#_Toc57576445)

[2.1 Ý tưởng 38](#_Toc57576446)

[2.2 Vẽ và thiết kế hệ thống 38](#_Toc57576447)

[2.2.1 Biểu đồ hoạt động phía Server 38](#_Toc57576448)

[2.2.2 Biểu đồ hoạt động phía Client 39](#_Toc57576449)

[2.3 Demo chương trình 39](#_Toc57576450)

[CHƯƠNG 3 – KẾT LUẬN 45](#_Toc57576451)

**DANH MỤC KÍ HIỆU VÀ CHỮ VIẾT TẮT**

**CÁC KÝ HIỆU**

**CÁC CHỮ VIẾT TẮT**

TCP/IP Transmission Control Protocol/Internet Protocol

HTTP Hyper Text Transfer Protocol

FTP File Transfer Protocol

LAN Local Area Network

ISDN Integrated Services Digital Network

DANH MỤC CÁC BẢNG BIỂU, HÌNH VẼ, ĐỒ THỊ

**DANH MỤC HÌNH**

[Hình 1-1: Giới thiệu giao thức TCP/IP. 6](#_Toc57661734)

[Hình 1-2: Mô hình Client/Server. 8](#_Toc57661735)

[Hình 1-3: Mô phỏng quá trình làm việc của Client/Server. 10](#_Toc57661736)

[Hình 1-4: Cơ chế Socket sử dụng trong Java. 11](#_Toc57661737)

[Hình 1-5: Màn hình chính giao diện chính. 16](#_Toc57661738)

[Hình 1-6: Cửa sổ của Start Client/Server. 17](#_Toc57661739)

[Hình 1-7: Giao diện để click vào Start Client. 17](#_Toc57661740)

[Hình 1-8: Giao diện chính của Server. 18](#_Toc57661741)

[Hình 1-9: Thông báo kết nối với số hiệu cổng. 19](#_Toc57661742)

[Hình 1-10: Giao diện của StartClient/Server. 20](#_Toc57661743)

[Hình 1-11: Nhập các thông số cần thiết cho Client. 21](#_Toc57661744)

[Hình 1-12: Thông báo kết nối thành công. 22](#_Toc57661745)

[Hình 1-13: Lỗi kết nối đến Server. 23](#_Toc57661746)

[Hình 1-14: Lỗi kết nối từ Client. 24](#_Toc57661747)

[Hình 1-15: Khung chat hiển thị bên phía Server đối với tên thứ nhất. 25](#_Toc57661748)

[Hình 1-16: Khung chat hiển thị bên phía Server đối với tên thứ hai. 26](#_Toc57661749)

[Hình 1-17: Giao diện đối với Client thứ nhất. 27](#_Toc57661750)

[Hình 1-18: Giao diện chat đối với Client thứ hai. 28](#_Toc57661751)

[Hình 1-19: Giao diện Menu chính để truy cập máy tính. 32](#_Toc57661752)

[Hình 1-20: Giao diện Client/Server của máy tính. 33](#_Toc57661753)

[Hình 1-21: Click to Server để mở Server. 33](#_Toc57661754)

[Hình 1-22: Thông báo kết nối từ chối. 34](#_Toc57661755)

[Hình 1-23: Click image to ... để mở giao diện tính toán. 35](#_Toc57661756)

[Hình 1-24: Giao diện chính thực hiện tính toán. 35](#_Toc57661757)

[Hình 1-25: Thông báo kết quả ra màn hình. 36](#_Toc57661758)

[Hình 1-26: Thông báo lỗi do nhập không đúng. 37](#_Toc57661759)

[Hình 2-1: Giao diện chính của Menu. 40](#_Toc57661760)

[Hình 2-2: Chọn FileServer để chạy chương trình. 40](#_Toc57661761)

[Hình 2-3: Cửa sổ hiển thị bên Server. 41](#_Toc57661762)

[Hình 2-4: Giao diện chọn FileClient để chạy chương trình. 42](#_Toc57661763)

[Hình 2-5: Giao diện bên phía Client. 42](#_Toc57661764)

[Hình 2-6: Giao diện để lựa chọn file phù hợp. 43](#_Toc57661765)

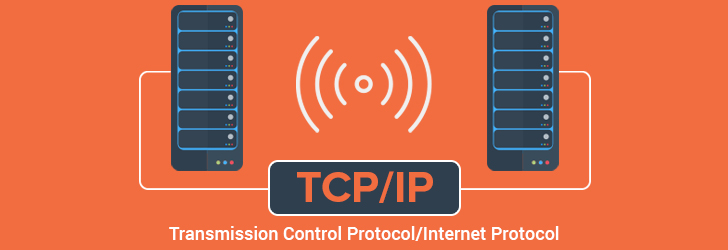
[Hình 2-7: Chọn một file bất kỳ để gửi. 44](#_Toc57661766)

[Hình 2-8:Thông báo hiển thị kết nối chấp nhận và nhận được File thành công. 44](#_Toc57661767)

**DANH MỤC BẢNG**

CHƯƠNG 1 – PHẦN BẮT BUỘC

1. Tổng quan
2. Giao thức TCP/IP



Hình 1-: Giới thiệu giao thức TCP/IP.

TCP/IP hoặc Transmission Control Protocol/Internet Protocol (Giao thức điều khiển truyền vận/giao thức mạng) là một bộ các giao thức trao đổi thông tin được sử dụng để kết nối các thiết bị mạng trên Internet. TCP/IP có thể được sử dụng như là một giao thức trao đổi thông tin trong một mạng riêng (intranet hoặc extranet). Toàn bộ bộ giao thức Internet - một tập hợp các quy tắc và thủ tục - thường được gọi là TCP/IP, mặc dù trong bộ cũng có các giao thức khác. TCP/IP chỉ định cách dữ liệu được trao đổi qua Internet bằng cách cung cấp thông tin trao đổi đầu cuối nhằm mục đích xác định cách thức nó được chia thành các gói, được gắn địa chỉ, vận chuyển, định tuyến và nhận ở điểm đến. TCP/IP không yêu cầu quản lý nhiều và nó được thiết kế để khiến mạng đáng tin cậy hơn với khả năng phục hồi tự động.

Có hai giao thức mạng chính trong bộ giao thức mạng phục vụ các chức năng cụ thể. TCP xác định cách các ứng dụng tạo kênh giao tiếp trong mạng. Ngoài ra, nó cũng quản lý cách các tin được phân thành các gói nhỏ trước khi được chuyển qua Internet và được tập hợp lại theo đúng thứ tự tại địa chỉ đến.

IP xác định bằng cách gán địa chỉ và định tuyến từng gói để đảm bảo nó đến đúng nơi. Mỗi gateway trên mạng kiểm tra địa chỉ IP này để xác định nơi chuyển tiếp tin nhắn.

Với IP, giao thức TCP chỉ mới có thể truyền dữ liệu chính xác từ máy này qua máy kia mà chưa thể truyền chính xác đến từng ứng dụng trên máy được. Hiện nay, các hệ thống máy thông thường hoạt động theo chế độ đa nhiệm, nghĩa là có nhiều ứng dụng chạy cùng một lúc và trong đó có thể có nhiều ứng dụng sử dụng dịch vụ mạng. Yêu cầu, khi máy chủ A truyền dữ liệu cho một ứng dụng u trên máy B thì trên máy B phải đảm bảo dữ liệu đó phải đến được ứng dụng U, chứ không phải ứng dụng. Để thực hiện điều đó thì máy chủ A khi truyền dữ liệu đi thì trên dữ liệu đó có một thành phần thông tin giúp máy B xác định được đúng ứng dụng U . Phần thông tin đó chính là địa chỉ port trên máy B, nó có thể hiểu là lỗ cắm ảo trên máy B mà ứng dụng u đã đăng ký để độc quyền sử dụng nhằn nhận dữ liệu từ máy chủ A.

Trên thực tế, địa chỉ port là một số nguyên 2 byte có giá trị từ 0 đến 65535. Nó có đặc điểm :

- Giá trị từ 0 đến 1023 là các cổng phổ biến dành cho các ứng dụng thông dụng như HTTP: 80, Mail : 25, FTP : 21, Telnet 23.... Các giá trị còn lại có thể được sử dụng linh hoạt.

- Mỗi cổng trong mỗi thời điểm được sử dụng cho tối đa là 1 ứng dụng.

1. Mô hình Client/Server

Server được hiểu là máy chủ, thường là một hệ thống máy lớn, có bộ xử lý mạnh, có khả năng hoạt động đáng tin cậy, có khả năng lưu trữ dữ liệu lớn, nó chuyên quản lý tài nguyên (chủ yếu là cơ sở dữ liệu), cung cấp các dịch vụ mạng cho các máy khách (Client) sử dụng. Bình thường nó chạy suốt thời gian thực và sẵn sàng chấp nhận các yêu cầu kết nối và các yêu cầu dịch vụ khác từ máy khách.

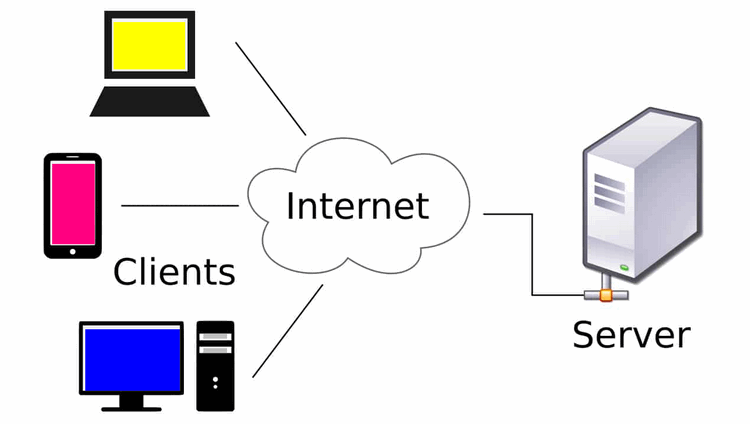
Client là máy khách, nó thường được sử dụng bởi người dùng cuối. Nó hoạt động dựa trên việc sử dụng dịch vụ mà máy server cung cấp để thực hiện các công việc mà người dùng cuối mong muốn.

Quy trình hoạt động của mô hình này lặp lại 2 quá trình như sau :

- Client gửi yêu cầu lên Server.

- Server nhận được yêu cầu thì sẻ xử lý thích hợp và trả lời lại Client.

Client server là mô hình mạng máy tính bao gồm 2 thành phần chính là máy khách (Client) và máy chủ (Server). Trong mô hình này, server là nơi lưu trữ tài nguyên, cài đặt các chương trình dịch vụ và thực hiện các yêu cầu của client. Client đón vai trò gửi yêu cầu đến Server. Client gồm máy tính và thiết bị điện tử nói chung.



Hình 1-: Mô hình Client/Server.

Mô hình Client/Server cho phép mạng tập trung các ứng dụng và chức năng tại một hoặc nhiều máu dịch vụ file chuyên dụng. Các máy này trở thành trung tâm của hệ thống. Hệ điều hành của Client/Server cho phép người dùng chia sẻ đồng thời cùng một tài nguyên, không quan trọng vị trí địa lý.

Mô hình web Client/Server là một mô hình nổi tiếng trong mạng máy tính, được áp dụng rất rộng rãi và là mô hình của mọi trang web hiện có. Một mô hình ngược lại là mô hình master-slaver, trong đó máy chủ (đóng vai trò ông chủ) sẽ gửi dữ liệu đến máy con (đóng vai trò nô lệ) bất kể máy con có cần hay không.

Mô hình Client/Server như sau: Client/Server là mô hình tổng quát nhất, trên thực tế thì một server có thể được nối tới nhiều server khác nhằm làm việc hiệu quả và nhanh hơn. Khi nhận được 1 yêu cầu từ Client, Server này có thể gửi tiếp yêu cầu vừa nhận được cho Server khác ví dụ như Database Server vì bản thân nó không thể xử lý yêu cầu này được. Máy server có thể thi hành các nhiệm vụ đơn giản hoặc phức tạp.

* Cách thức hoạt động của Client/Server

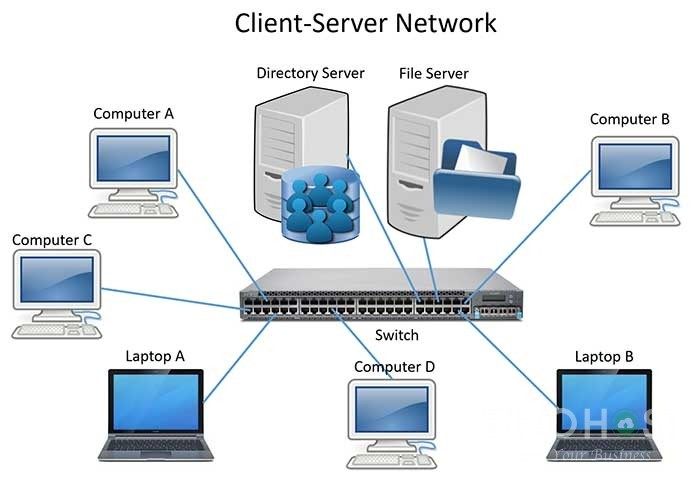
Toàn bộ quá trình giao tiếp giữa Server và Client phải dựa trên các giao thức chuẩn. TCP/IP, SAN (IBM), ISDN, OSI, X.25, LAN-to-LAN Netbios là những giao thức chuẩn được sử dụng phổ biến hiện nay. Client server là giải pháp phần mềm hiệu quả, giúp khắc phục tình trạng quá tải của hệ thống mạng. Bên cạnh đó, mô hình này còn vượt qua sự khác biệt trong cấu trúc vật lý và hệ điều hành của các hệ thống máy tính. Mô hình Client server gồm có 2 phần là Client & Server:

* Client

Client hay chính là máy khách, máy trạm – là nơi gửi yêu cầu đến Server. Nó tổ chức giao tiếp với người dùng, server và môi trường bên ngoài tại trạm làm việc. Client tiếp nhận yêu cầu của người dùng sau đó thành lập các query string để gửi cho Server. Khi nhận được kết quả từ Server, Client sẽ tổ chức và trình diễn những kết quả đó.

* Server

Server xử lý yêu cầu gửi đến từ client. Sau khi xử lý xong, server sẽ gửi trả lại kết quả, client có thể tiếp tục xử lý các kết quả này để phục vụ người dùng. Server giao tiếp với môi trường bên ngoài và client tại Server, tiếp nhận yêu cầu dưới dạng query string (xâu ký tự). Khi phân tích xong các xâu ký tự, Server sẽ xử lý dữ liệu và gửi kết quả về cho client.



Hình 1-: Mô phỏng quá trình làm việc của Client/Server.

* Ưu nhược điểm

- Ưu điểm

+ Client/Server có khả năng chống quá tải mạng.

+ Client/Server đảm bảo toàn vẹn dữ liệu khi có sự cố xảy ra.

+ Dễ dàng mở rộng hệ thống mạng.

+ Chỉ cần chung định dạng giao tiếp mà không cần chung nền tảng là có thể hoạt động được.

+ Client/Server cho phép tích hợp các kỹ thuật hiện đại như GIS, mô hình thiết kế hướng đối tượng,…

+ Với mô hình Client/Server, người dùng có thể truy cập dữ liệu từ xa, thực hiện các thao tác gửi, nhận file hay tìm kiếm thông tin đơn giản.

- Nhược điểm

+ Cần bảo trì, bảo dưỡng Server thường xuyên.

+ Khả năng bảo mật thông tin mạng là một hạn chế nữa của Client/Server.

1. Cơ chế Socket

Socket là một khái niệm để định vị một dịch vụ của một máy tính trên mạng khi kết hợp hai khái niệm IP và Port lại.

Socket có những nhiệm vụ như sau :

• Gắn một cổng trên máy.

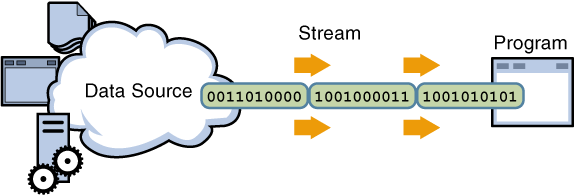
• Lắng nghe các kết nổi.

• Kết nối/Đóng kết nối đến máy tính ở xa qua cổng đã gắn.

• Gửi/nhận dữ liệu.

• Lắng nghe dữ liệu đến.

Đúng như tính chất của TCP chúng ta cần có liên kết 2 chiều trước khi Server và Client có thể trao đổi thông điệp với nhau. Ban đầu, phía Server tạo Socket được ràng buộc với một cổng (port number) để chờ nhận yêu cầu từ phía Client. Tiếp đến phía Client yêu cầu Server bằng cách tạo một Socket TCP trên máy kèm với địa chỉ IP và port number của tiến tình tương ứng trên máy Server. Khi client tạo Socket, Client TCP tạo liên kết với Server TCP và chờ chấp nhận kết nối từ Server. TCP cung cấp dịch vụ truyền dòng tin cậy và có thứ tự giữa Client và Server, giữa máy chủ và máy nhận chỉ có 1 địa chỉ IP duy nhất. Thêm vào đó, mỗi thông điệp truyền đi đều có xác nhận trả về.



Hình 1-: Cơ chế Socket sử dụng trong Java.

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.

* Các gói ServerSocket hỗ trợ

public Socket(int port) : Tạo cổng kết nối trên cổng port.

public accept():Chấp nhận một kết nối và trả về một đối tượng Socket mô tả nguồn đã được kết nối đến.

getInputStream() : Trả về một luồng mà có thể nhập dữ liệu vào.

getOutputStream() : Trả về một luồng ghi dữ liệu.

close() : Đóng kết nối khi không cần.

* Các gói Socket

public Socket(InetAddress address, int port): Kết nối tới một máy chủ thông qua địa chỉ IP và số hiệu cổng trên máy chủ đó.

public OutputStream getOutputStream(): Tạo luồng dữ liệu ra.

public InputStream getInputStream(): Tạo luồng dữ liệu vào.

public void Close(): Đóng kết nối.

* Các gói hỗ trợ khác

Gói Swing, awt.event: Tạo giao diện người dùng.

Gói Runnable, Thread: Hỗ trợ đa tiến trình.

Gói java.io: Cung cấp các lớp vào ra dữ liệu.

1. Yêu cầu 1: Xây dựng chương trình Chat giữa Client với Server theo yêu cầu
2. Lý thuyết liên quan

Xây dựng chương trình Chat giữa Client/Server với Socket. Socket là một 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 để kết nối với nhau. Điểm cuối (endpoint) của liên kết này được gọi là Socket, cho phép người dùng kết nối các máy tính truyền tải và nhận dữ liệu từ máy tính thông qua mạng. Vì vậy, hiểu đơn giản thì socket là thiết bị truyền thông hai chiều gửi và nhận dữ liệu từ máy khác. Để xây dựng được một ứng dụng chat bằng Socket thì cần phải hiểu rõ cách thức hoạt động của nó, có rất nhiều loại Socket, nhưng ở đây mình sẽ lấy Stream Socket.

Để xây dựng chương trình bao gồm các nội dung như sau:

* Xây dựng chương trình bên phía Server
* Tạo một TCP Socket và gắn một cổng nội bộ để kết nối
* Xây dựng chương trình Server đa tuyến để nhiều Client cùng lúc có thể kết nối đến Server
* Server luôn luôn chờ và lắng nghe kết nối từ các Client tương ứng
* Server chấp nhận kết nối và tạo ra các Socket với các Client tương ứng
* Cho phép chat giữa các Client với Server với nhau
* Thiết kế giao diện cho Server
* Xử lí một số công việc khác liên quan đến chương trình,..vv
* Xây dựng chương trình bên phía Client
* Tạo một TCP Socket với địa chỉ IP và số cổng cũng như tên mà chương trình Server đang chạy
* Thiết lập kết nối và chat với các thành viên khác thông qua Server
* Thiết kế giao diện cho Client
* Xử lí một số công việc khác liên quan đến chương trình,...vv

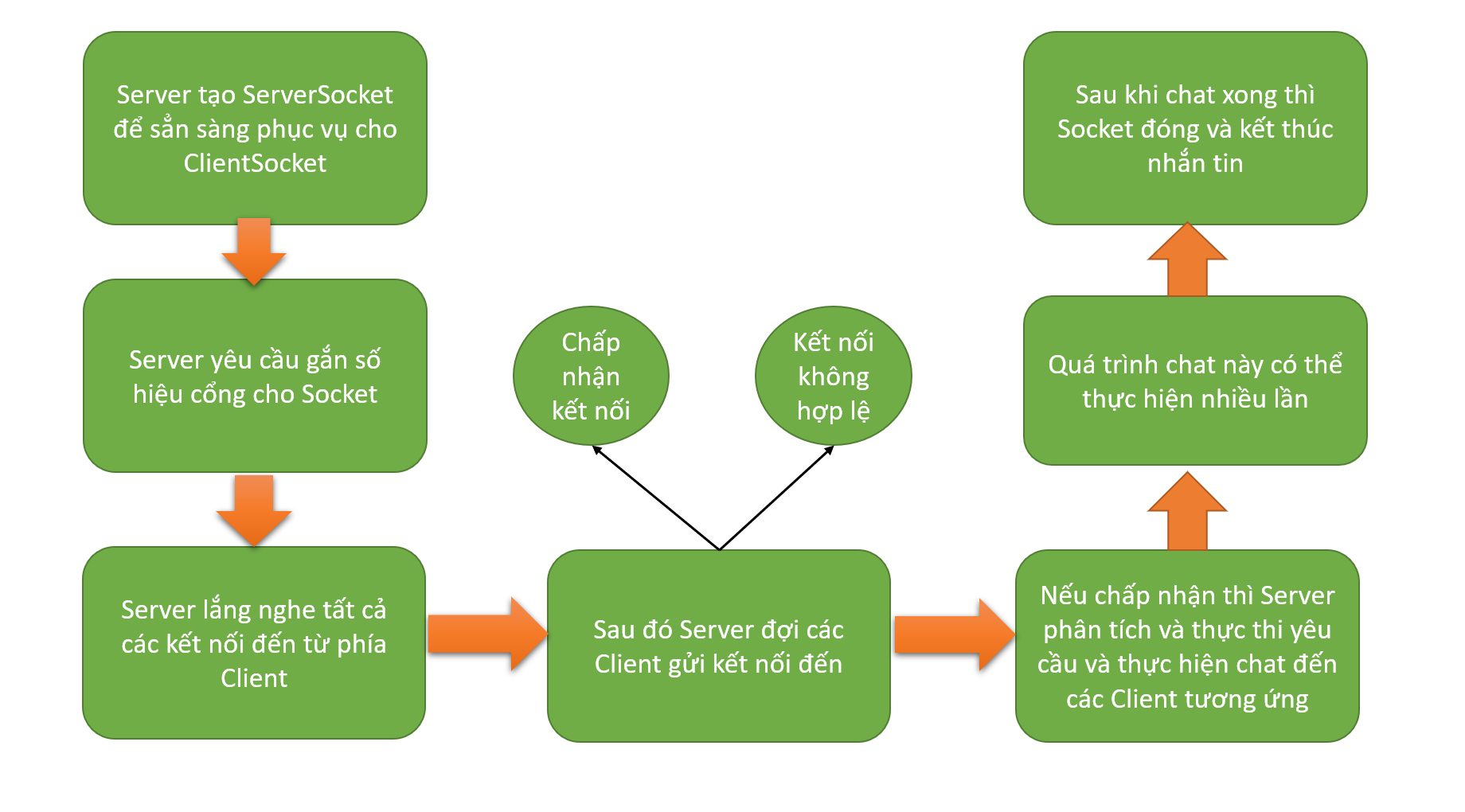
1. Biểu đồ tổng quát hệ thống

**Server**  **Client**

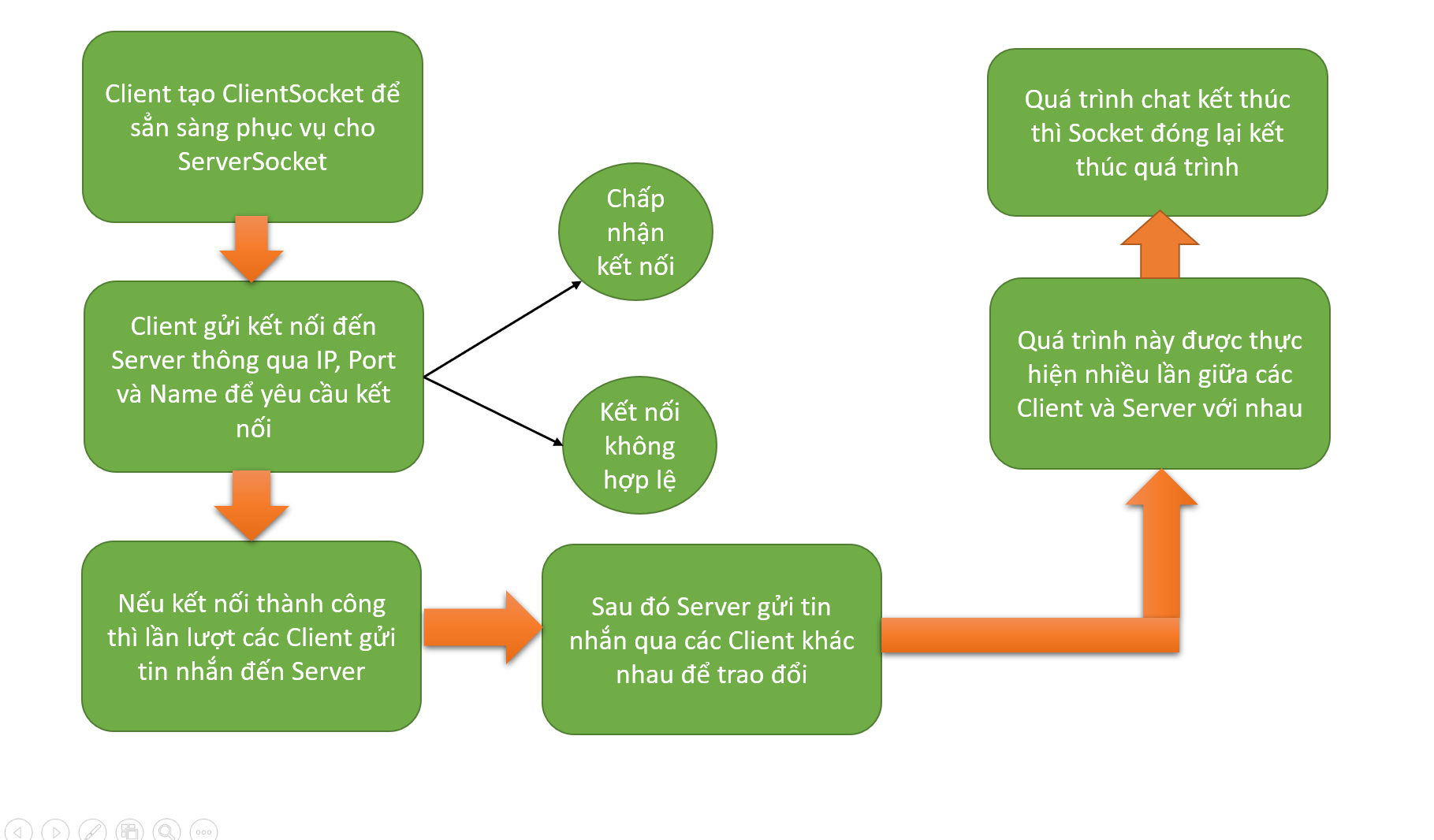
Thực hiện trao đổi tin nhấn qua lại

Chấp nhận kết nối để thực hiện chat

1. Biểu đồ hoạt động của Server

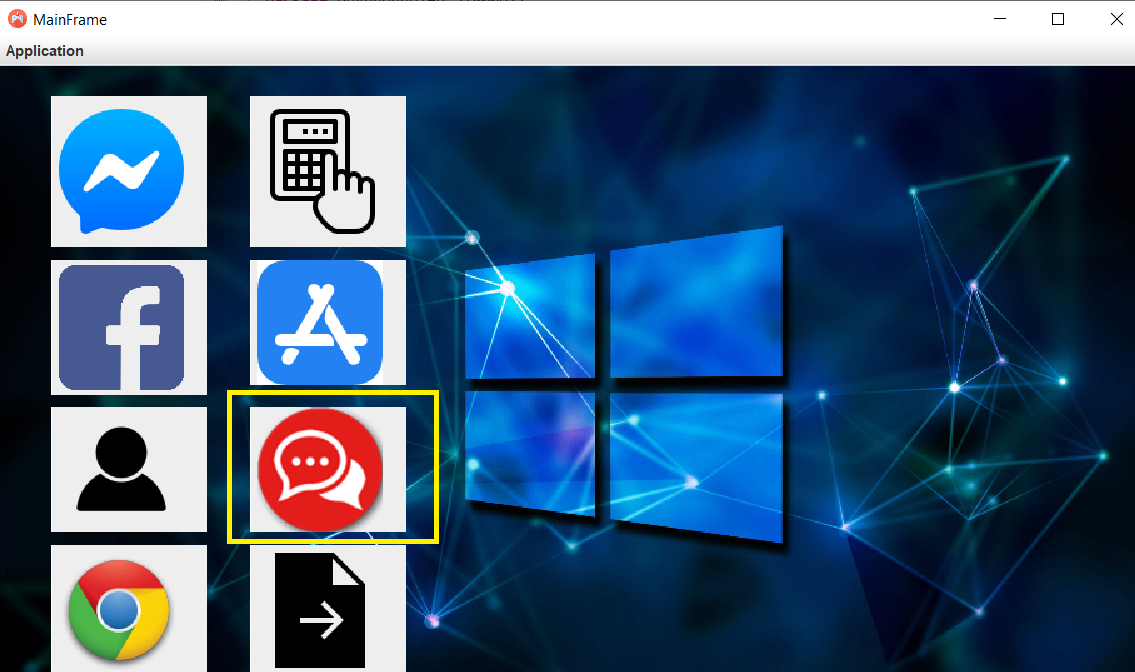


1. Biểu đồ hoạt động của Client



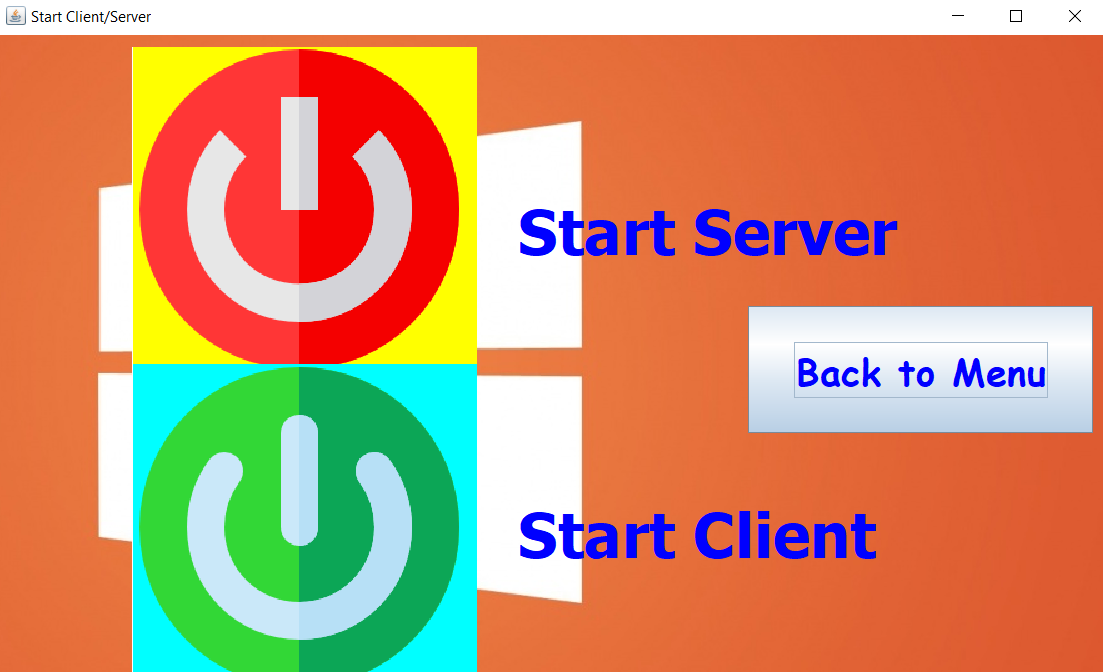
1. Demo chương trình

- Để mở chương trình chat giữa Client/Server đầu tiên Click vào biểu tượng “Chat” trên màn hình như hình bên dưới:



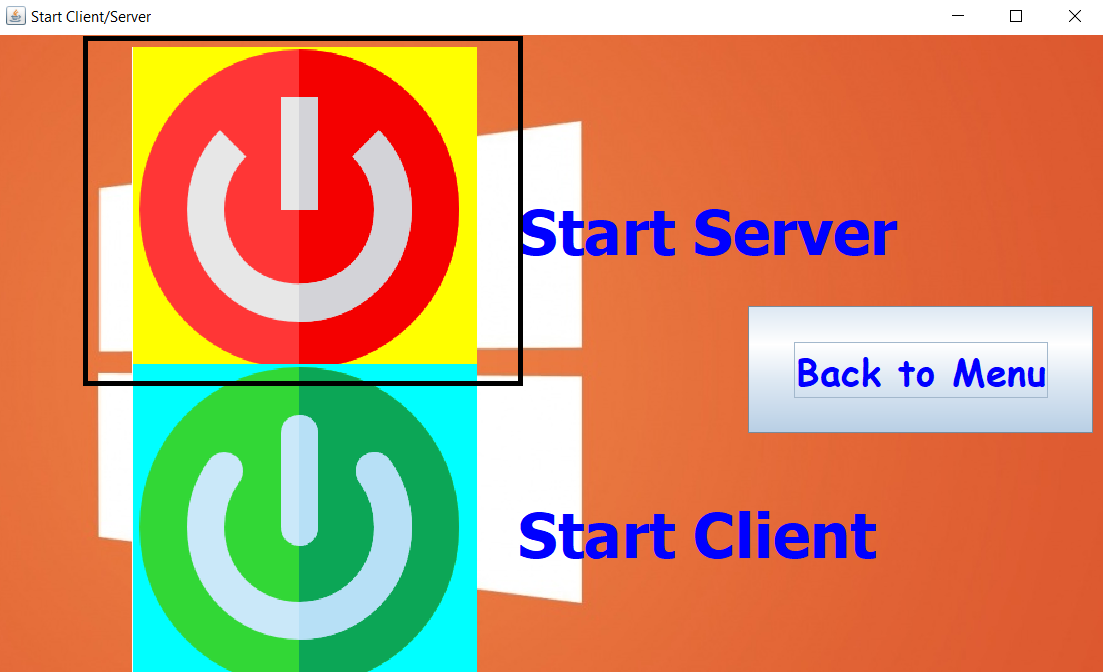
Hình 1-: Màn hình chính giao diện chính.

- Khi đó một giao diện khác sẽ hiện lên để tạo Client và Server:



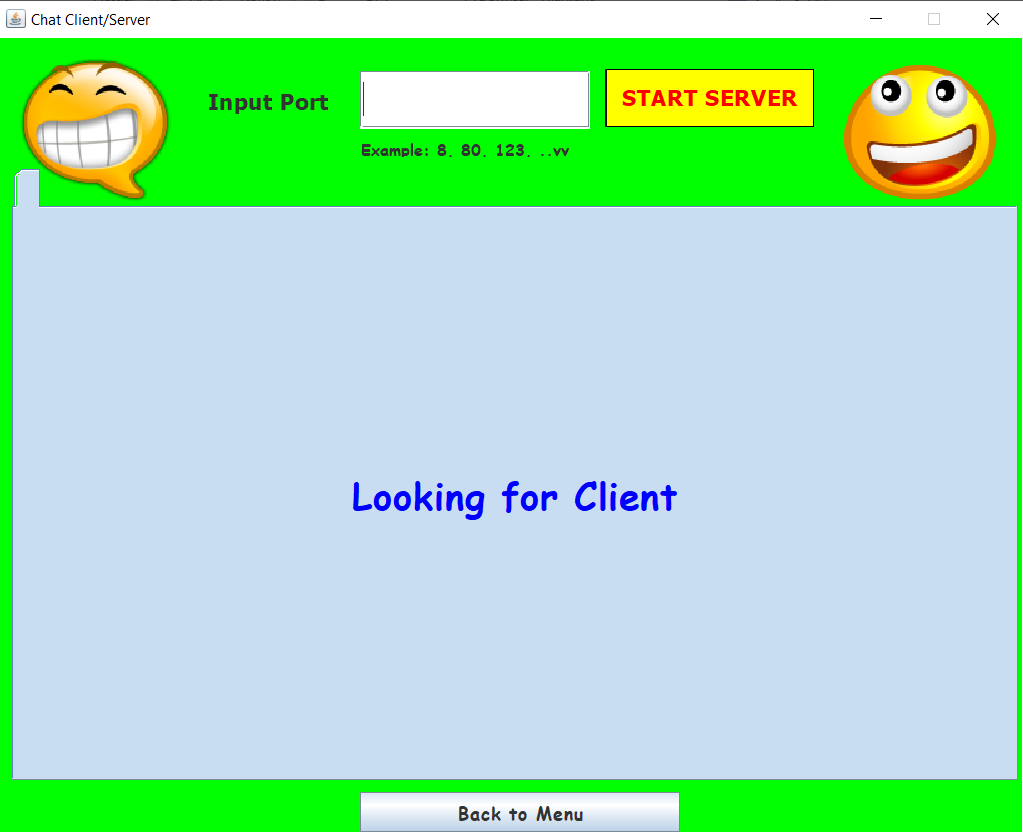
Hình 1-: Cửa sổ của Start Client/Server.

- Từ cửa sổ “Start Client/Server” Click vào icon “Start Server” để tạo kết nối cho Server đến Client



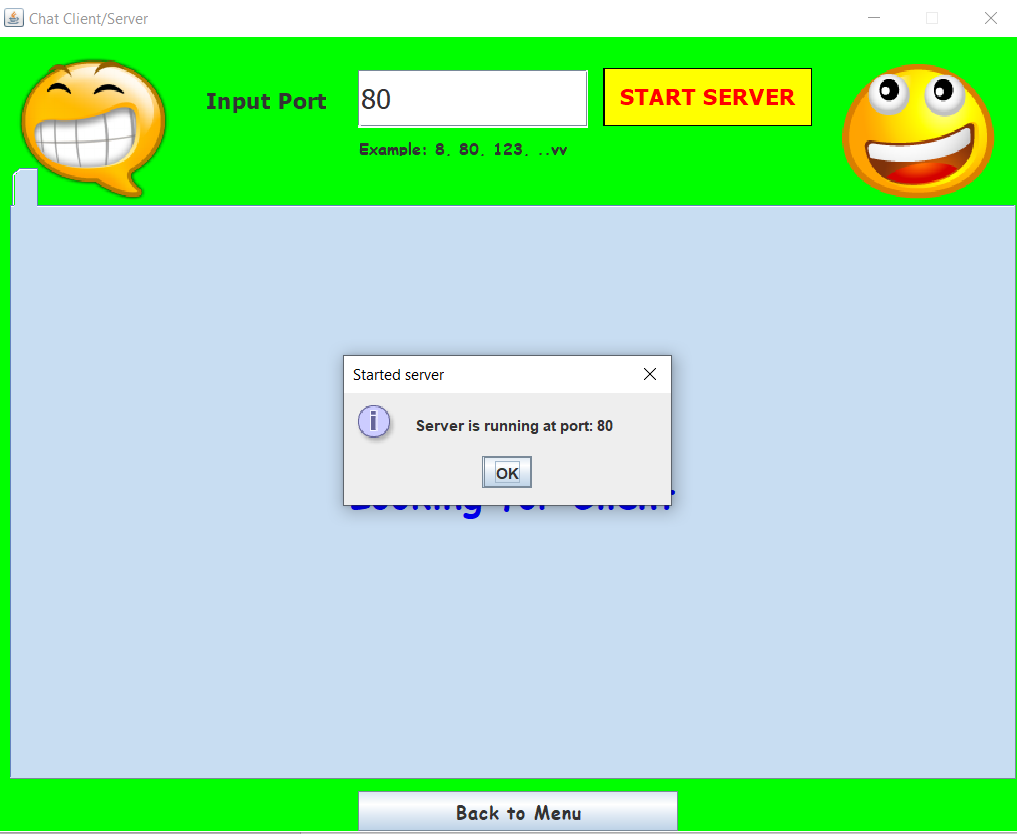
Hình 1-: Giao diện để click vào Start Client.

- Sau khi Click vào thì sẽ hiện ra một cửa sổ để nhập số hiệu cổng để kết nối đến Client



Hình 1-: Giao diện chính của Server.

- Sau đó nhập một cổng tùy ý để kết nối và Click “Start Server” thì Server thông báo số hiệu cổng trên đã được mở



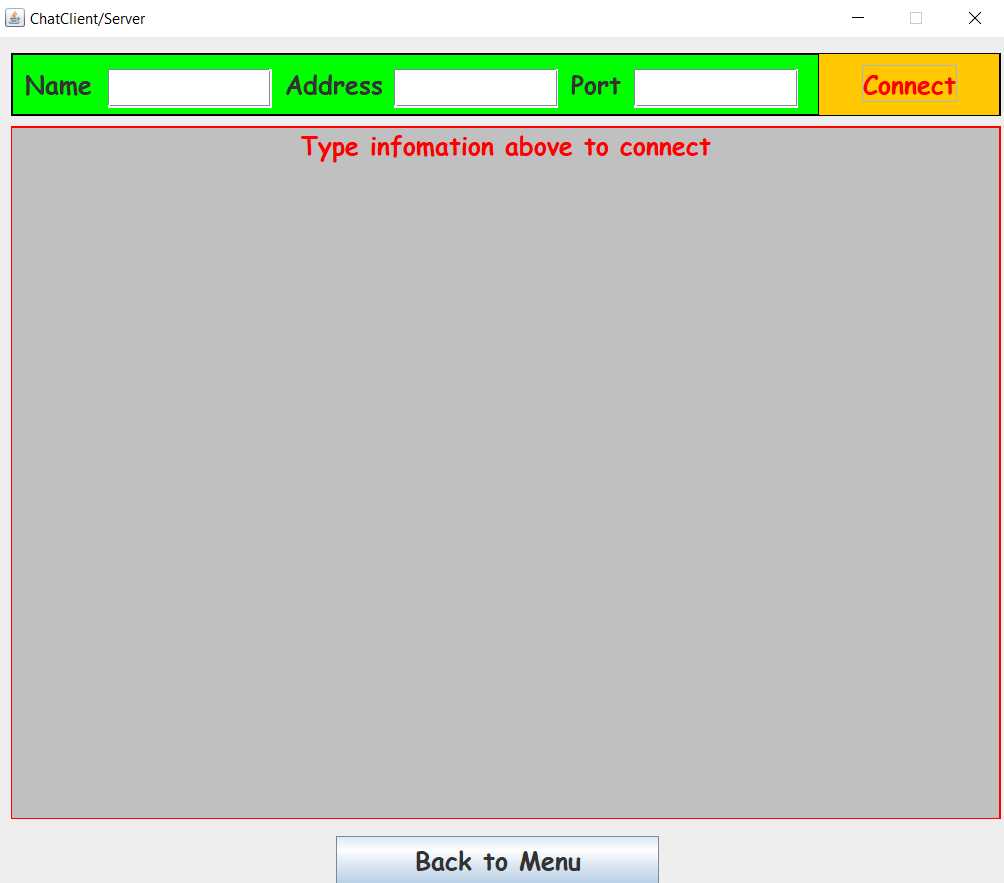
Hình 1-: Thông báo kết nối với số hiệu cổng.

- Trở lại màn hình “Start Client/Server” ta Click vào icon “Start Client” để mở giao diện Client:



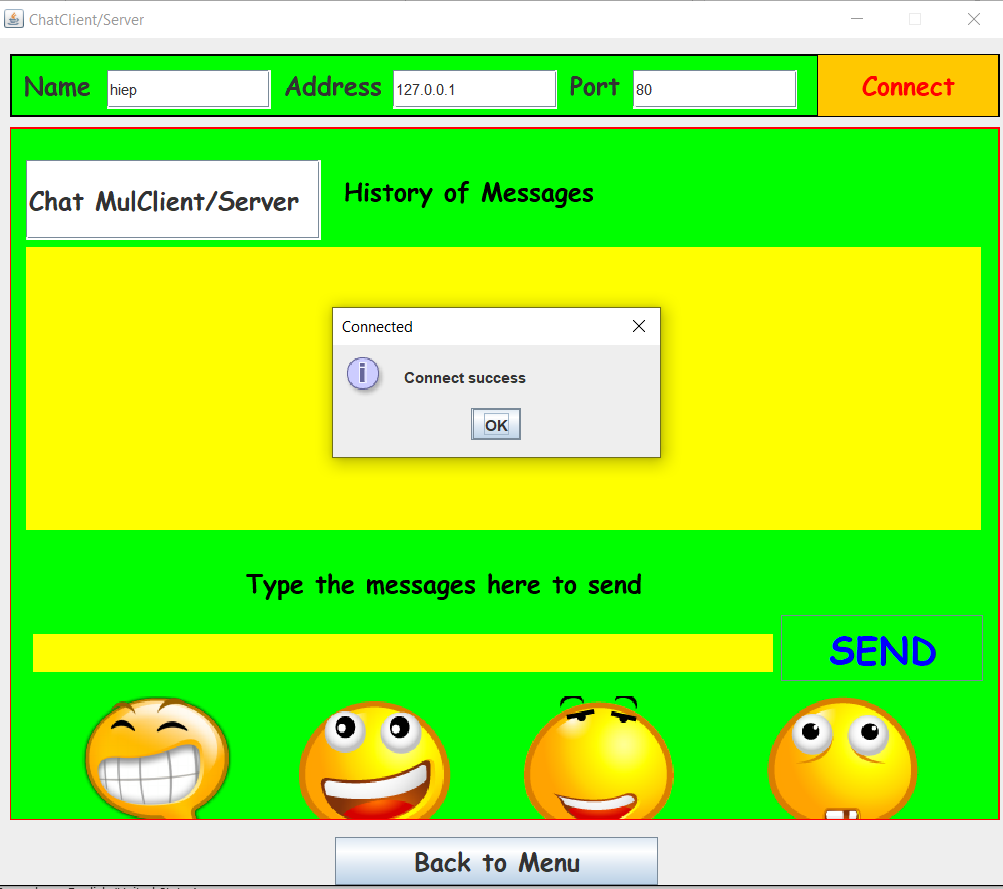
Hình 1-: Giao diện của StartClient/Server.

- Khi Click vào một cửa sổ khác hiện ra để nhập các thông số như tên , địa chỉ và số hiệu cổng để thực hiện chat với Server



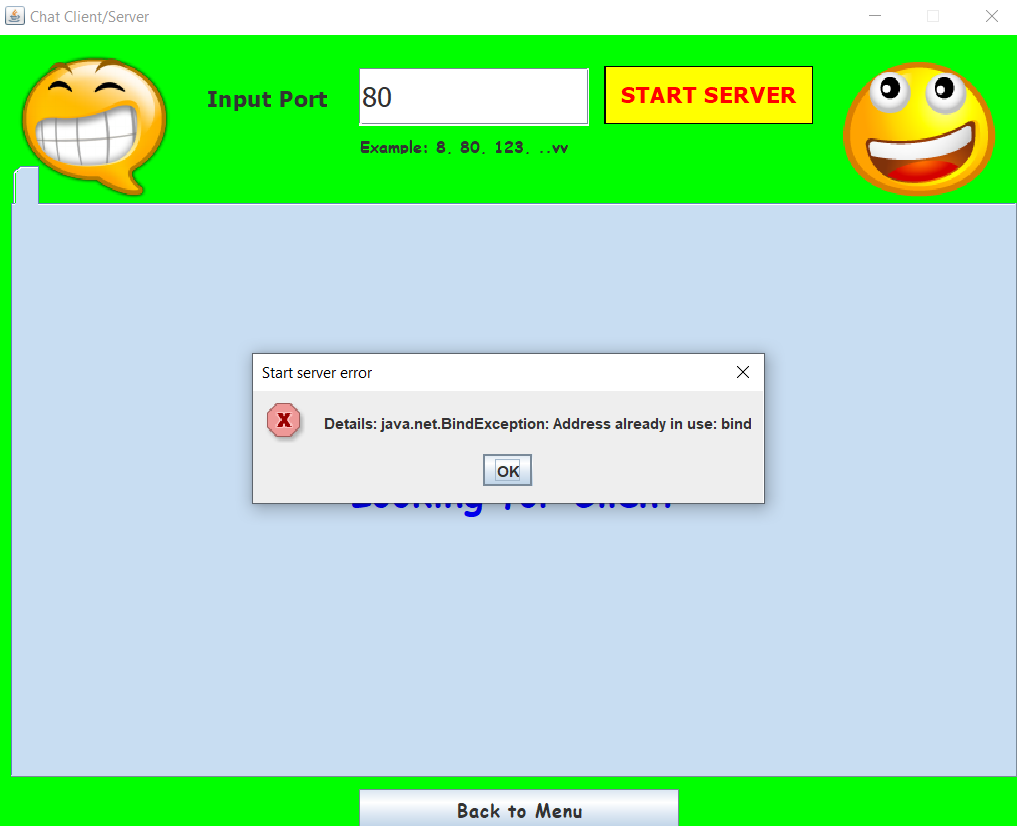
Hình 1-: Nhập các thông số cần thiết cho Client.

- Sau đó điền các thông số cần thiết và nhấn “Connect” để kết nối. Nếu đúng số cổng đã nhập trên Server thì hiện thông báo kết nối thành công



Hình 1-: Thông báo kết nối thành công.

- Nếu kết nối không thành công thì các thông báo lần lượt bên Client và Server sẽ hiện ra:



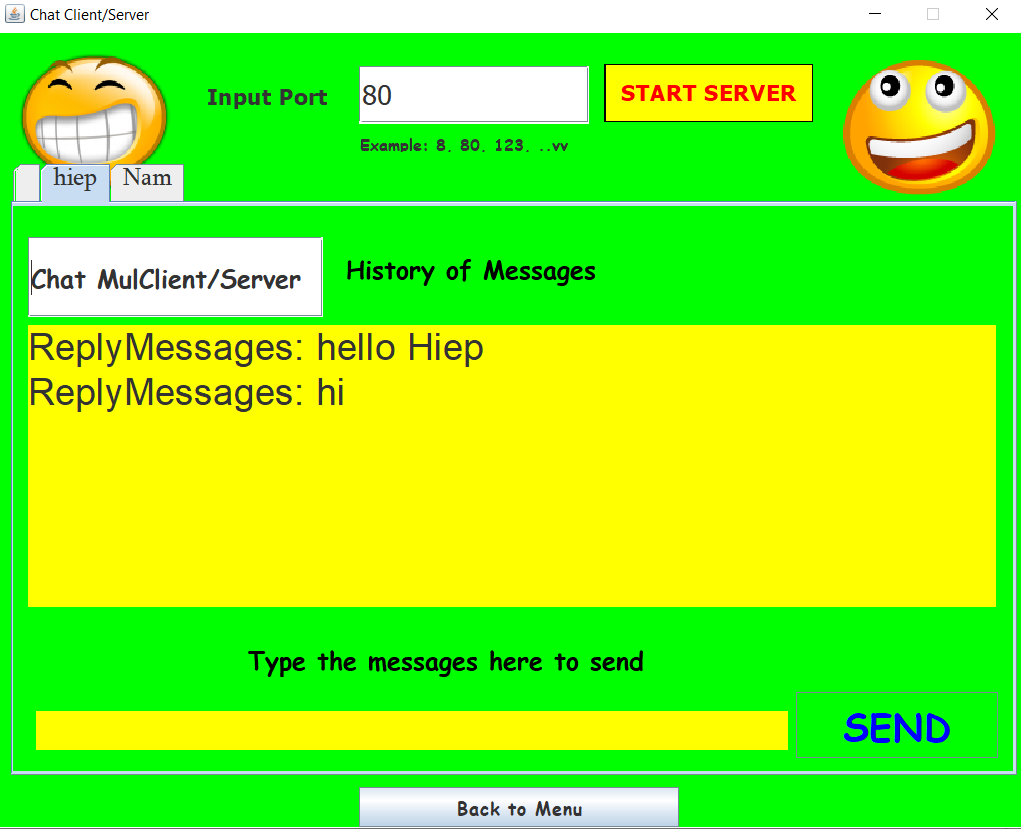
Hình 1-: Lỗi kết nối đến Server.



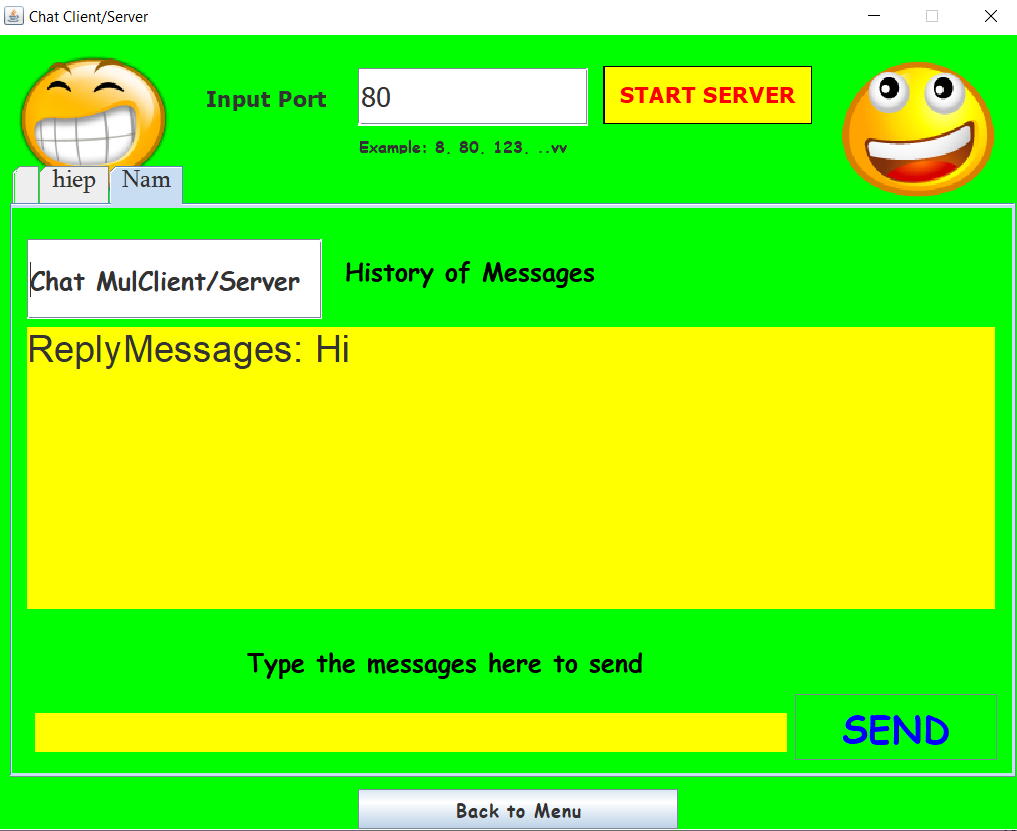
Hình 1-: Lỗi kết nối từ Client.

- Khi kết nối thành công thì chúng ta sẽ chat giữa các Client và Server với nhau thông qua giao diện trên màn hình:

- Bên phía Server sẽ hiển thị các thông tin:



Hình 1-: Khung chat hiển thị bên phía Server đối với tên thứ nhất.

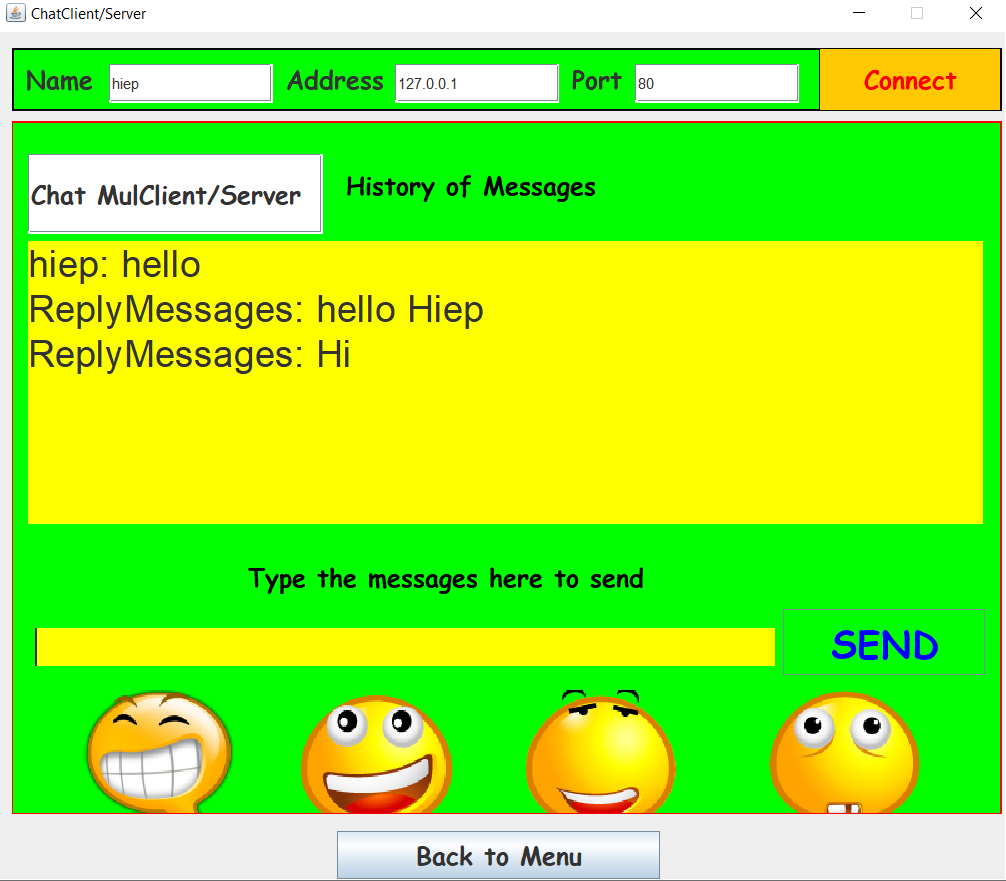


Hình 1-: Khung chat hiển thị bên phía Server đối với tên thứ hai.

- Bên phía các Client sẽ hiện như hình bên dưới:



Hình 1-: Giao diện đối với Client thứ nhất.



Hình 1-: Giao diện chat đối với Client thứ hai.

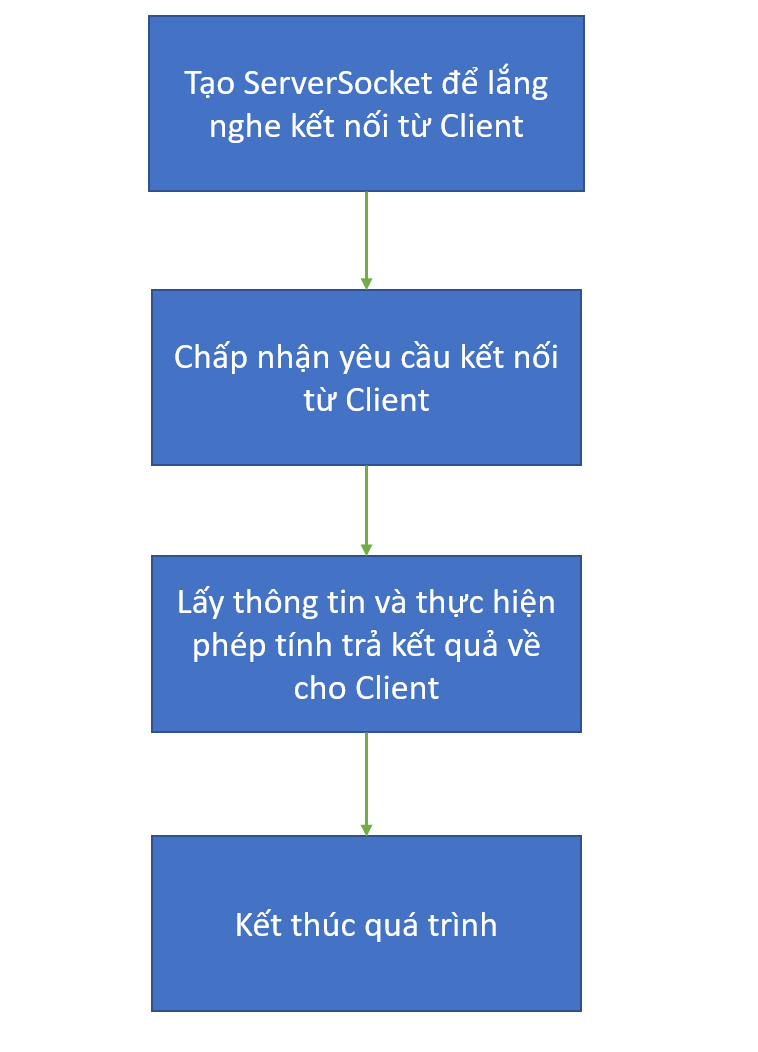
1. Yêu cầu 2: Xây dựng chương trình Client/Server thực hiện phép tính (cộng, trừ, nhân, chia) theo yêu cầu
2. Xây dựng chương trình

Xây dựng chương trình Client/Server cho phép thực hiện các phép tính toán cơ bản như cộng, trừ, nhân, chia dựa trên Socket. Chương trình được tạo gồm Client và Server. Đầu tiên Server và Client kết nối với nhau thông qua Socket được gắn số hiệu cổng. Sau đó, Client cho phép nhập số A và B cần thiết để lấy dữ liệu và gửi lên Server sao đó Server thực hiện tính toán và trả kết quả về Client và hiện kết quả cho Client. Các trường hợp không kết nối cũng như dữ liệu không đúng sẽ trả về các thông báo khác nhau ở chương trình demo bên dưới.

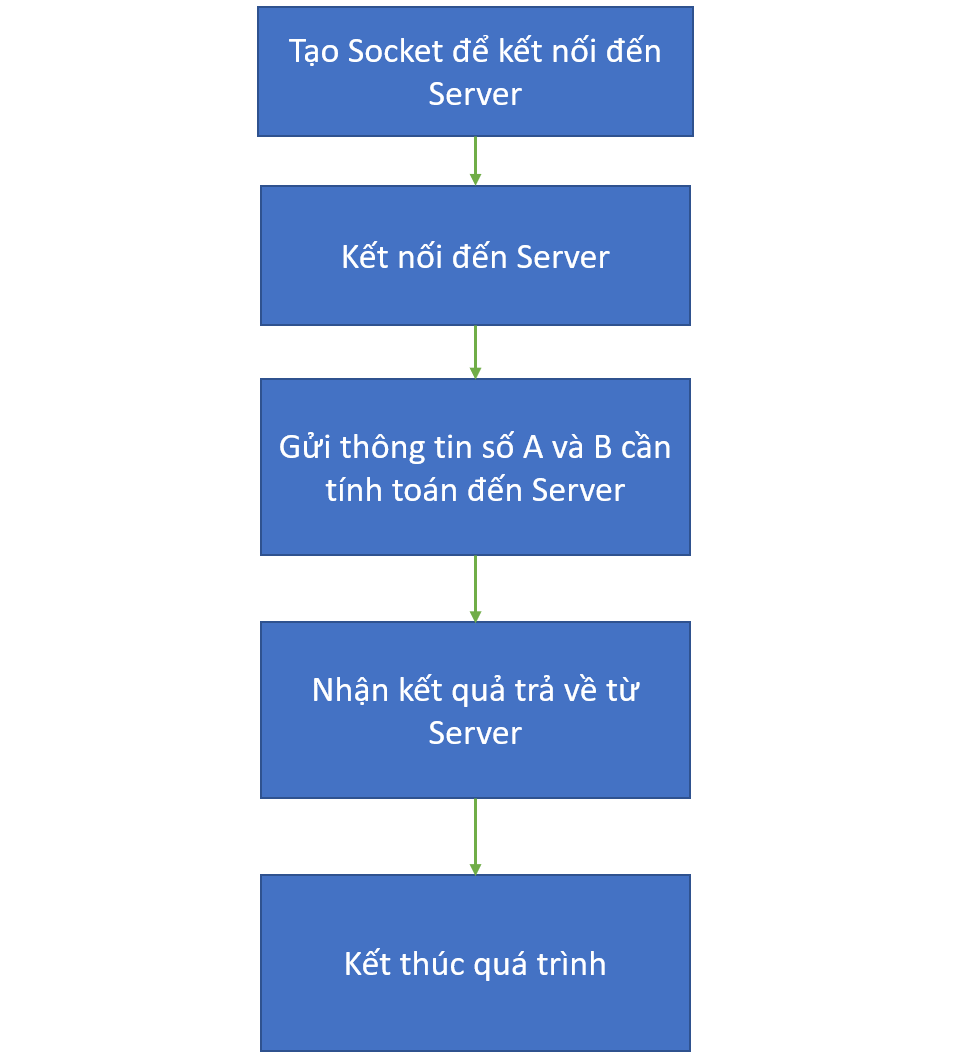
1. Vẽ và thiết kế hệ thống

Dưới đây mô hình hoạt động của máy tính thông qua hoạt động từ phía Client và từ phía Server:

* Server

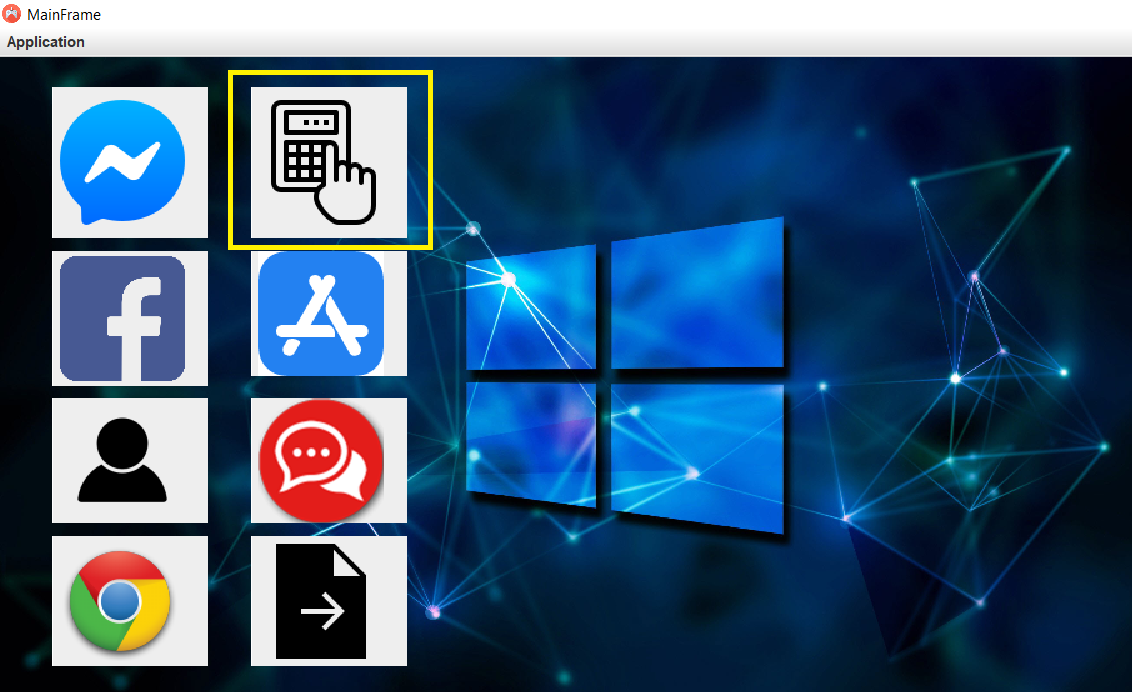


* Client



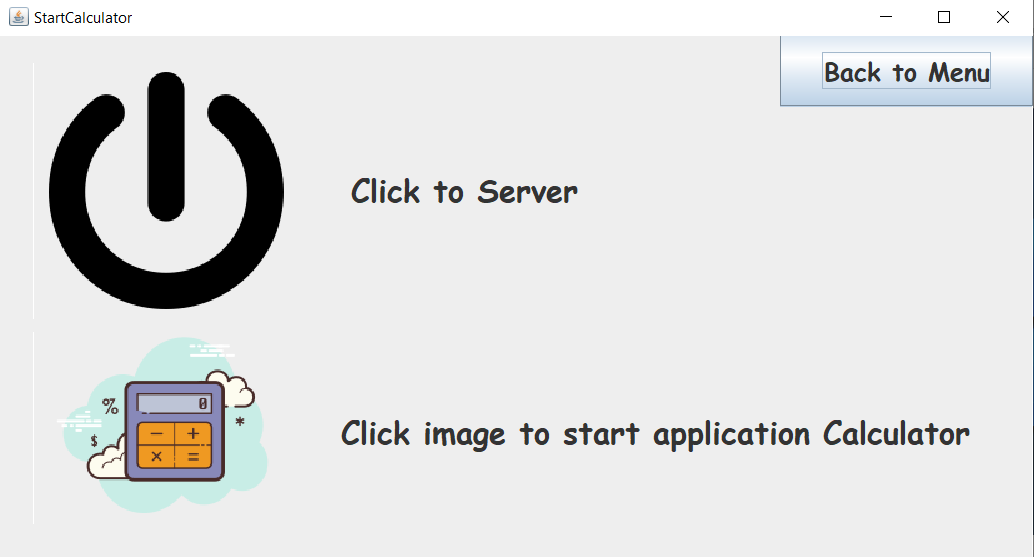
1. Demo chương trình

- Từ ứng dụng chính MainFame Click vào biểu tượng máy tính trên Menu để truy cập vào giao diện của máy tính:



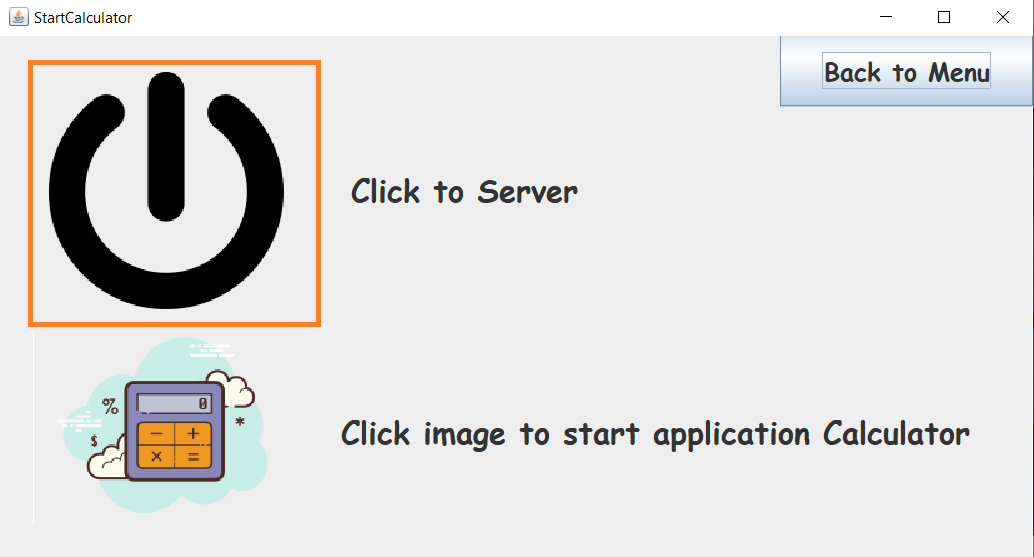
Hình 1-: Giao diện Menu chính để truy cập máy tính.

- Sau khi Click vào ta sẽ đến giao diện chính để bắt đầu tính toán bao gồm giao diện để mở File Client và Server:



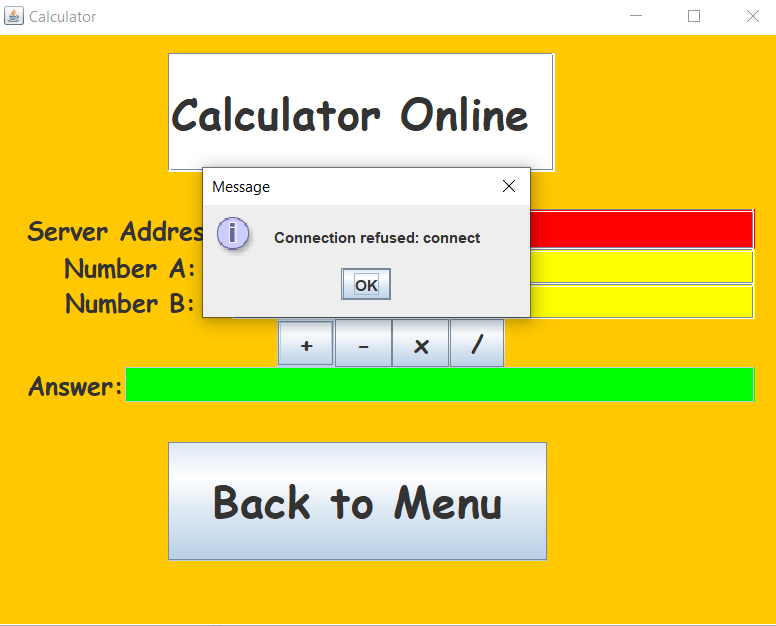
Hình 1-: Giao diện Client/Server của máy tính.

- Sau đó mình không cần Click vào Server vì Server đã được mở ngay lúc mở Menu.



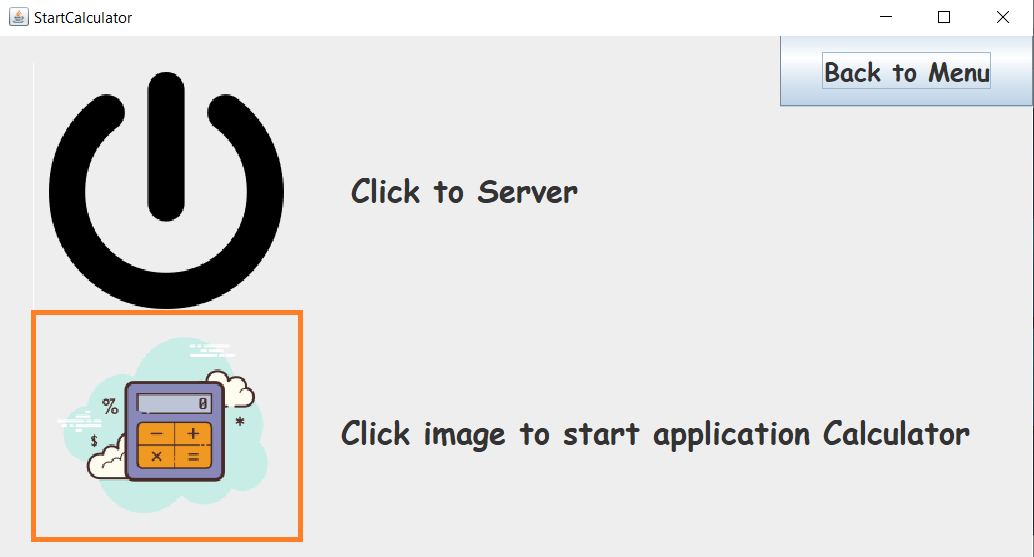
Hình 1-: Click to Server để mở Server.

- Nếu không mở Server thì kết nối bị từ chối.



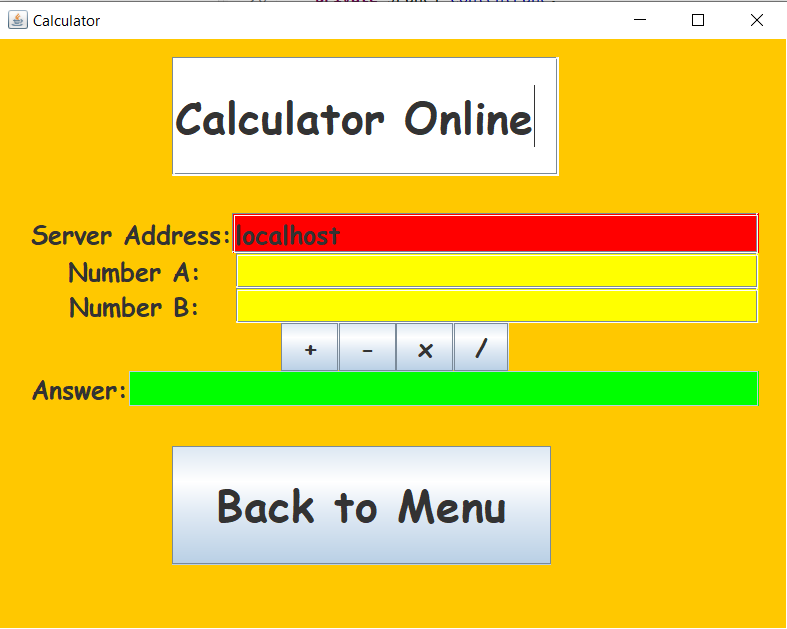
Hình 1-: Thông báo kết nối từ chối.

- Sau đó click vào icon máy tính để mở máy tính và thực hiện các phép tính:



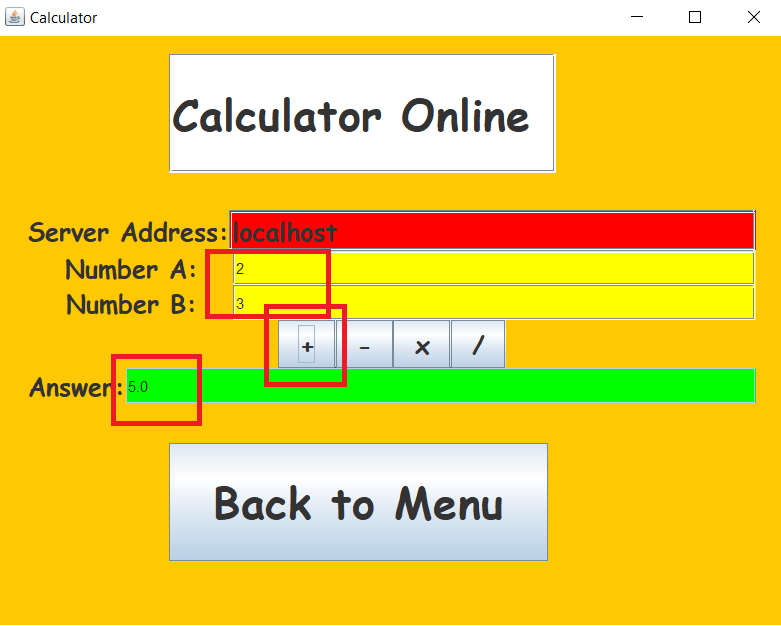
Hình 1-: Click image to ... để mở giao diện tính toán.

- Sau đó giao diện tính toán sẽ hiện ra.



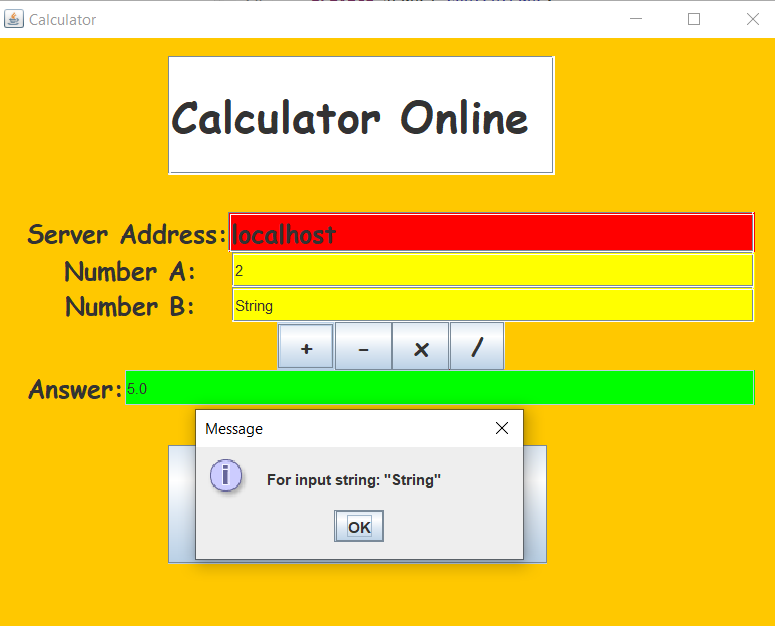
Hình 1-: Giao diện chính thực hiện tính toán.

- Ở đây tôi sẽ nhập hai số bất kỳ 2 và 3 và chọn nút “+” sẽ hiện kết quả như hình:



Hình 1-: Thông báo kết quả ra màn hình.

- Nếu bạn nhập không phải là số kết quả sẽ hiển thị như hình:



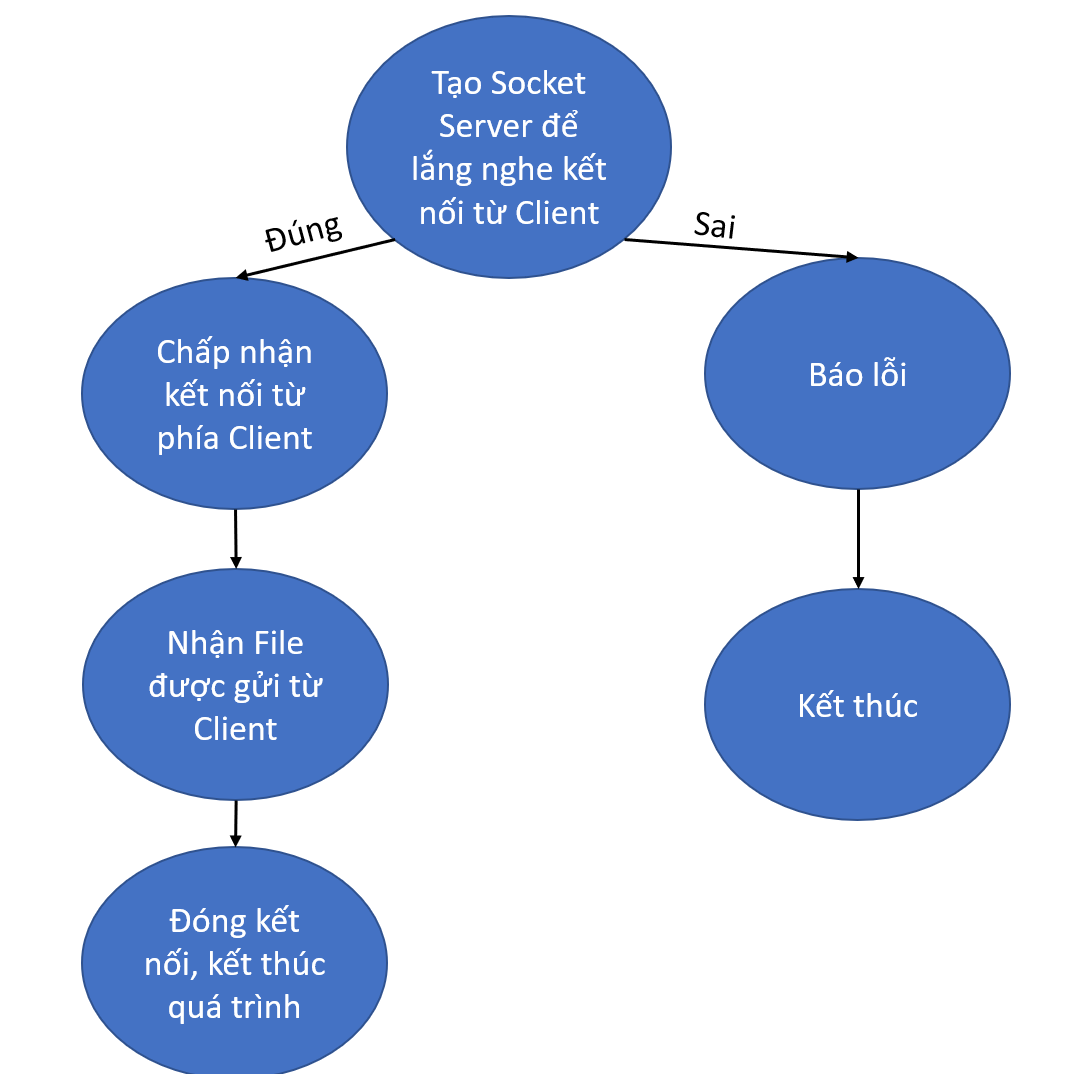
Hình 1-: Thông báo lỗi do nhập không đúng.

CHƯƠNG 2 – PHẦN Ý TƯỞNG

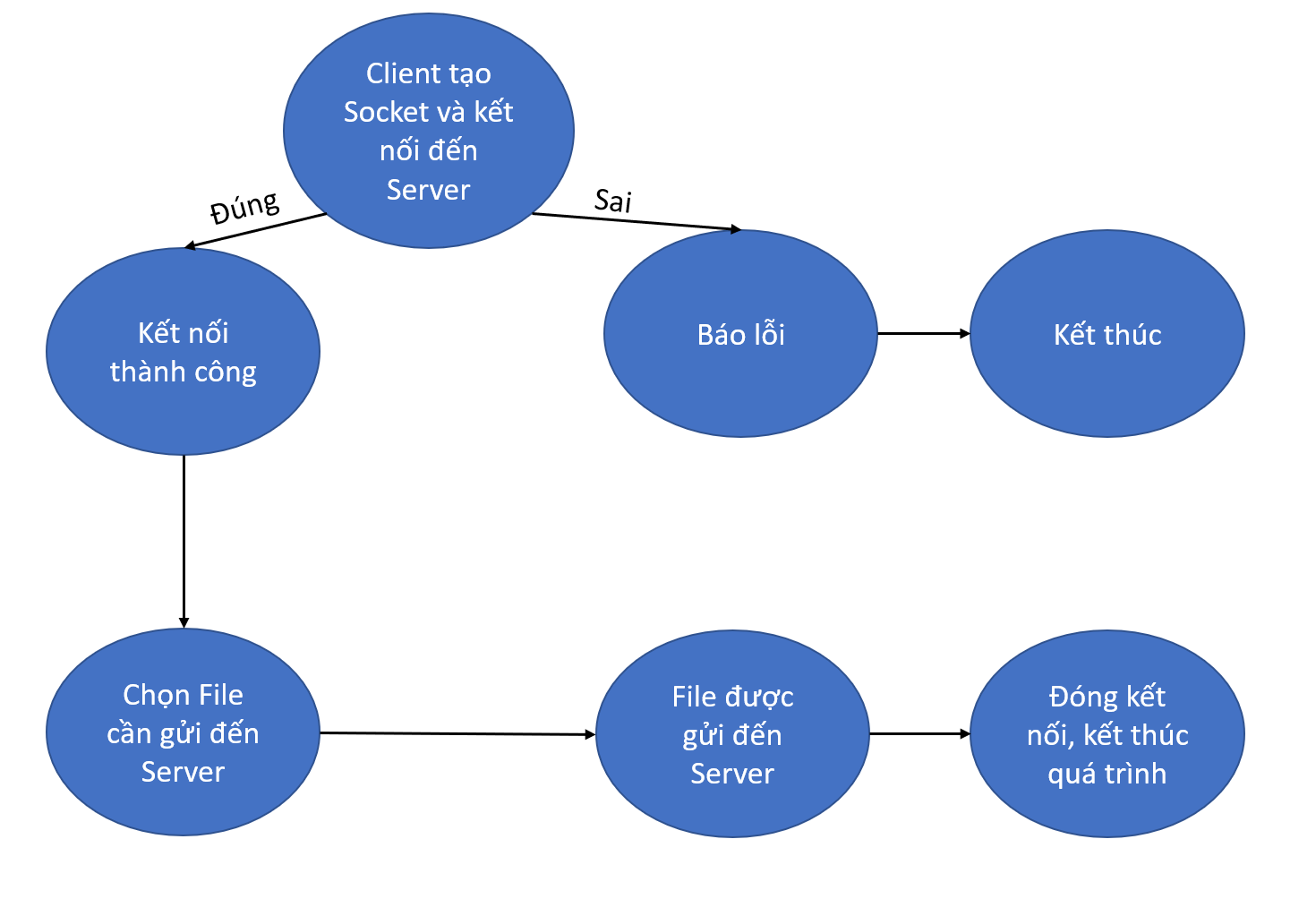
1. Ý tưởng

Trao đổi dữ liệu thực chất là quá trình trao đổi giữa hai chương trình trên máy tính nó tương tự như một máy tính được gán nhãn là Server và một máy gán nhãn là Client để trao đổi file với nhau. Ý tưởng thực hiện đề tài này là quá trình truyền một file bất kỳ dựa trên phương thức JFileChooser trong JavaSwing từ phía Client để gửi lên Server lúc đó Server nhận được file thông qua cơ chế kết nối của Socket dựa trên số cổng đã được gán trên cả Client và Server.

1. Vẽ và thiết kế hệ thống
2. Biểu đồ hoạt động phía Server

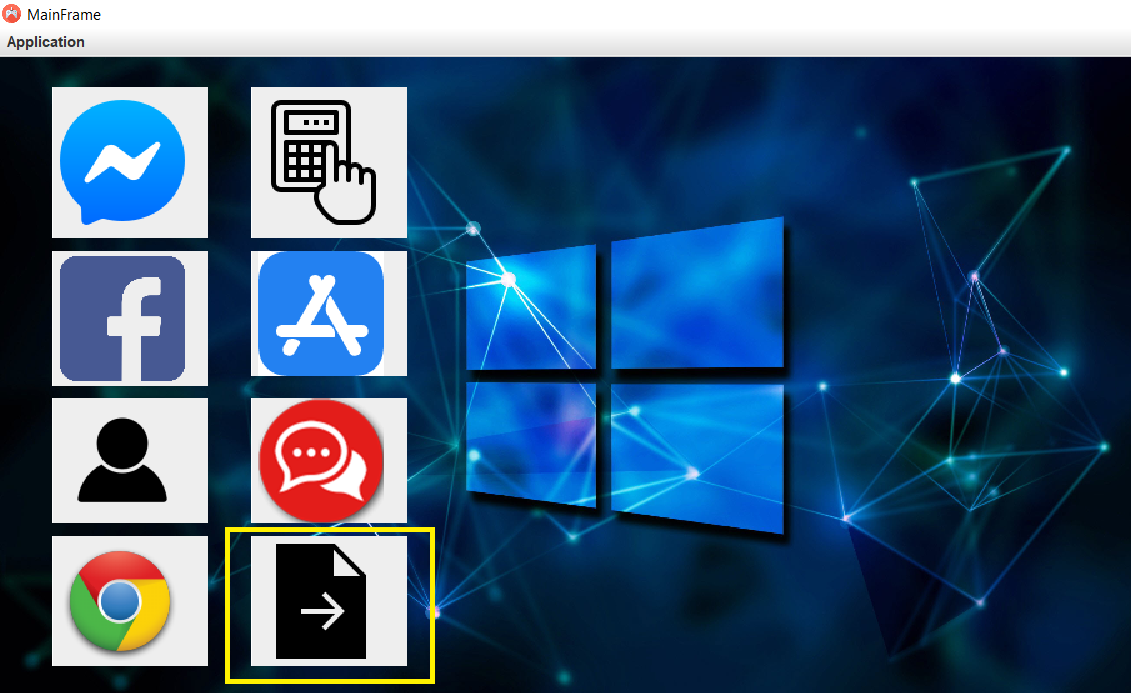


1. Biểu đồ hoạt động phía Client



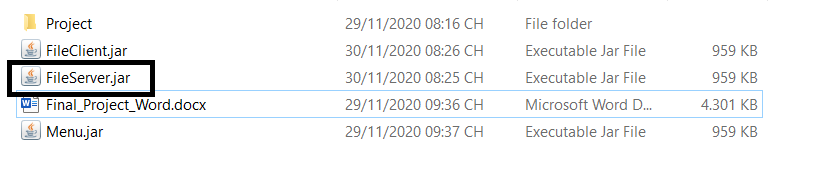
1. Demo chương trình

- MainFrame chứa giao diện Send file chỉ là mô phỏng do chưa tạo luồng thành công nên mình sẽ chạy bằng mở file.jar:



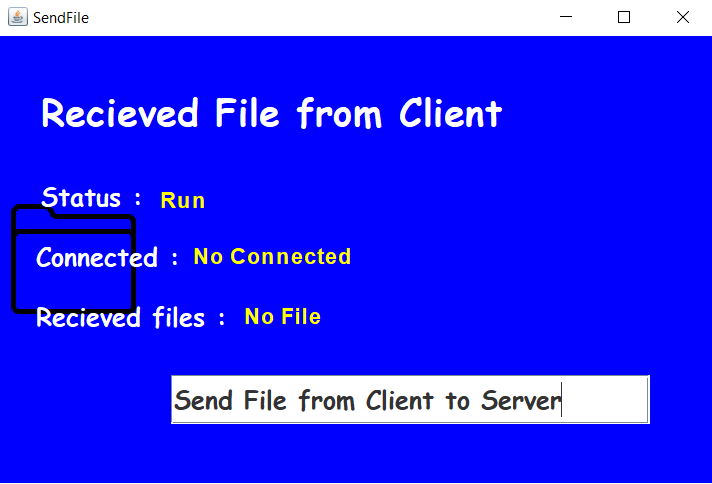
Hình 2-: Giao diện chính của Menu.

- Đầu tiên chạy file FileServer.jar sẽ hiển thị giao diện như hình:



Hình 2-: Chọn FileServer để chạy chương trình.

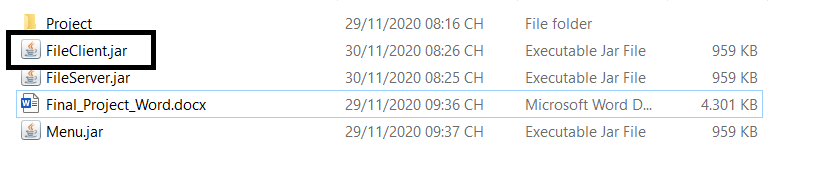
- Một cửa sổ sẽ hiện ra bên phía Server:



Hình 2-: Cửa sổ hiển thị bên Server.

🡪Ở đây nó bao gồm trạng thái đang chạy và chưa kết nối cũng như chưa nhận File từ Client.

- Sau đó Click vào file FileClient.jar để chạy phía Client để gửi file thích hợp:



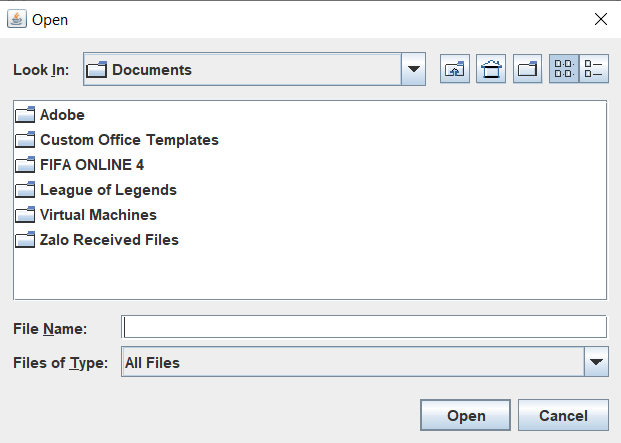
Hình 2-4: Giao diện chọn FileClient để chạy chương trình.

- Sau đó Click vào “Browse file” để chọn file thích hợp:



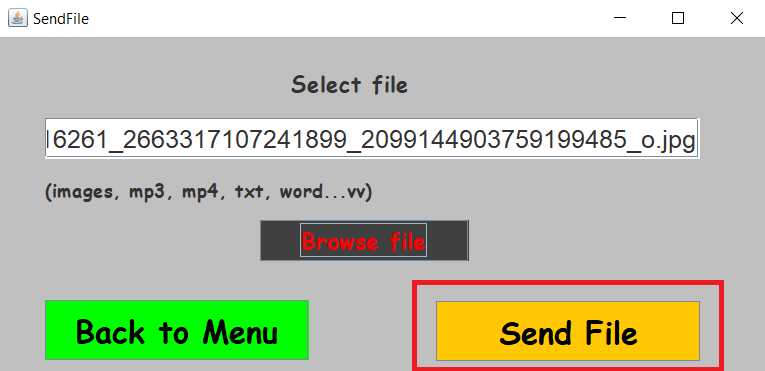
Hình 2-5: Giao diện bên phía Client.

- Sau đó chọn file thích hợp để gửi.



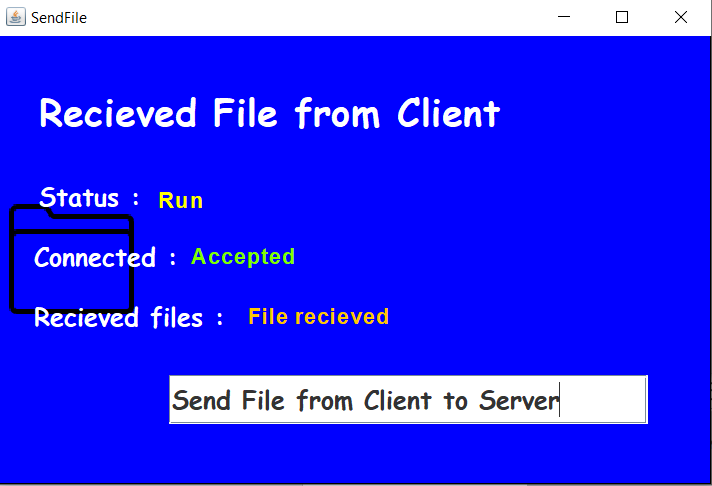
Hình 2-6: Giao diện để lựa chọn file phù hợp.

- Sau khi chọn File xong Click vào nút Send để gửi file lên Server. Ở đây mình chọn một hình ảnh bất kỳ:



Hình 2-7: Chọn một file bất kỳ để gửi.

- Khi đó giao diện bên phía Server sẽ báo hiển thi File thành công như hình bên:



Hình 2-8:Thông báo hiển thị kết nối chấp nhận và nhận được File thành công.

CHƯƠNG 3 – KẾT LUẬN

Về lý thuyết, đồ án đã trình bày một cách khái quát về giao thức TCP/IP và lập trình Socket và lập trình Socket trong Java đặc biệt và tầm quan trọng cũng như vai trò của mô hình Client/Server và các nội dung liên quan.

Về ứng dụng đồ án, đã phân tích các cơ chế cũng như cách thức hoạt động của các ứng dụng từ phía Client cho đến Server một cách khá hoàn chỉnh, có thể chạy cũng như tham khảo một cách dễ dàng, giao diện dễ nhìn và có thể làm các ứng dụng tương tự.

* Thành tựu đạt được

- Khái quát được cơ sở của vấn đề bao gồm: TCP/IP, Client/Server, Socket.

- Mô phỏng được ứng dụng: cách hoạt động của ứng dụng, cách chạy chương trình thực hiện được giữa Client với Server.

- Có kiến thức cơ bản về lập trình Socket, Client/Server.

- Có thể áp dụng các ứng khác trong tương lai.

* Hạn chế

- Còn khá nhiều tính năng chưa được tính hợp khó áp dụng vào thực tiễn ngay phải trải qua chỉnh sửa.

- Ứng dụng chưa hoàn thiện nhất do vấn đề liên quan ảnh hưởng.

**TÀI LIỆU THAM KHẢO**

**Tiếng Việt**

1. <https://quantrimang.com/kien-thuc-ve-giao-thuc-mang-tcp-ip-48>. [Accessed 5 11 2020].
2. <https://vdodata.vn/tim-hieu-ve-mo-hinh-client-server-va-client-server-la-gi/>. [Accessed 1 11 2020].
3. <https://topdev.vn/blog/socket-la-gi-websocket-la-gi/>. [Accessed 6 11 2020].

**Tiếng Anh**

1. <https://www.javatpoint.com/socket-programming>. [Accessed 10 11 2020].