

TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN &
TRUYỀN THÔNG VIỆT HÀN

KHOA KỸ THUẬT MÁY TÍNH VÀ ĐIỆN TỬ



**ĐỒ ÁN MÔN HỌC LẬP TRÌNH MẠNG
XÂY DỰNG ỨNG DỤNG CHAT
THEO MÔ HÌNH CLIENT – SERVER**

Sinh viên thực hiện:

Lê Quang Long

Nguyễn Thị Lâm Anh

Lớp:

20NS

Giảng viên hướng dẫn:

ThS. Nguyễn Thanh Cẩm

Đà Nẵng, tháng 11 năm 2022

TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN &
TRUYỀN THÔNG VIỆT HÀN

KHOA KỸ THUẬT MÁY TÍNH VÀ ĐIỆN TỬ



ĐỒ ÁN MÔN HỌC LẬP TRÌNH MẠNG
XÂY DỰNG ỨNG DỤNG CHAT
THEO MÔ HÌNH CLIENT – SERVER

Sinh viên thực hiện:

Lê Quang Long

Nguyễn Thị Lâm Anh

Lớp:

20NS

Giảng viên hướng dẫn:

ThS. Nguyễn Thanh Cẩm

Đà Nẵng, tháng 11 năm 2022

NHẬN XÉT CỦA GIẢNG VIÊN HƯỚNG DẪN

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

LỜI CẢM ƠN

Đây là năm thứ ba chúng em học tại trường Đại học Công nghệ thông tin và Truyền thông Việt-Hàn. Chúng em nghĩ mình thật may mắn khi được vào học tại trường Việt-Hàn. Cơ sở vật chất trường rất tốt, thầy cô giảng viên rất nhiệt tình, quan tâm, chăm lo cho sinh viên.

Để hoàn thành bài đồ án môn học này, bên cạnh sự nỗ lực của bản thân, chúng em đã nhận được nhiều sự chỉ dẫn, giúp đỡ của các thầy cô trường Đại học Công nghệ thông tin và Truyền thông Việt-Hàn, đặc biệt là sự hướng dẫn tận tình của ThS Nguyễn Thanh Cẩm và các thầy cô giảng dạy. Thầy đã tận tình hướng dẫn trong suốt quá trình em thực hiện báo cáo này. Em xin chân thành cảm ơn thầy đã hết lòng hướng dẫn để em hoàn thành bài báo cáo này.

Tuy nhiên, vì đây là đồ án cơ sở chúng em thực hiện một dự án cho bản thân, mặc dù tìm tòi nghiên cứu nhưng không tránh khỏi những thiếu sót. Chúng em rất mong nhận được sự thông cảm và góp ý của thầy cô.

MỤC LỤC

MỤC LỤC	III
DANH MỤC CÁC TỪ VIẾT TẮT	V
DANH MỤC HÌNH VẼ.....	VI
MỞ ĐẦU	1
1. Giới thiệu đề tài.....	1
2. Mục tiêu của đề tài	1
3. Nội dung và kế hoạch thực hiện	1
4. Bố cục báo cáo	2
CHƯƠNG 1. CƠ SỞ LÝ THUYẾT	3
1.1. Sơ lược về lập trình mạng	3
1.1.2. Lập trình mạng.....	3
1.1.3. Một số mô hình trong lập trình mạng	3
1.1.4. Một số giao thức trong lập trình mạng	8
1.2. Sơ lược về ngôn ngữ lập trình Java.....	15
1.2.1. Giới thiệu về ngôn ngữ lập trình Java	15
1.2.2. Đặc điểm của ngôn ngữ lập trình Java	15
CHƯƠNG 2. PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG	18
2.1. Mô tả về hệ thống.....	18
2.2. Xây dựng biểu đồ	18
2.2.1. Usecase Diagram	18
2.2.2. Class Diagram.....	20
2.2.3. Activity Diagram	20
2.2.4. Sequence Diagram	21
2.3. Cơ sở dữ liệu	21
2.4. Cấu trúc thư mục	23
CHƯƠNG 3. CHƯƠNG TRÌNH DEMO	24
3.1. Giao diện Server.....	24

3.2. Giao diện Client	26
-----------------------------	----

DANH MỤC CÁC TỪ VIẾT TẮT

UML	Unified Modeling Language
LTM	Lập trình mạng
TCP	Transmission Control Protocol
IP	Internet Protocol
P2P	Peer-To-Peer
HTTP	Hypertext Transfer Protocol
FTP	File Transfer Protocol

DANH MỤC HÌNH VẼ

Hình 2. 1 Usecase tổng quát.....	18
Hình 2. 2 usecase diagram client.....	19
Hình 2. 3 usecase diagraeme server	19
Hình 2. 4 class diagram	20
Hình 2. 5 Activity Diagram Send Mess	20
Hình 2. 6 Sequence Diagram Send Mess	21
Hình 2. 7 Danh sách các bảng của cơ sở dữ liệu	21
Hình 2. 8 Cấu trúc bảng messInfo.....	21
Hình 2. 9 Cấu trúc bảng user.....	22
Hình 2. 10 Cấu trúc code.....	23
Hình 2. 11 Cấu trúc lưu trữ	23
Hình 2. 12 Server ghi lại log và insert mess lên database.....	25
Hình 3. 1 Giao diện chính của server	24
Hình 3. 2 Chức năng thêm user.....	24
Hình 3. 3 Chức năng chat với user	25
Hình 3. 4 Giao diện đăng nhập.....	26
Hình 3. 5 Giao diện đăng ký	26
Hình 3. 6 Giao diện chat	27
Hình 3. 7 Chọn nơi lưu file	27
Hình 3. 8 Kết quả sau khi lưu file	27
Hình 3. 9 Chọn file để gửi.....	28
Hình 3. 10 Thư mục của server sau khi user gửi file	28

MỞ ĐẦU

1. Giới thiệu đề tài

Hiện nay, các chương trình hội thoại đã trở thành một chương trình phổ biến của một đại bộ phận người sử dụng Internet. Nó như một phương tiện hiệu quả để kết nối, giúp chia sẻ, trao đổi thông tin giữa mọi người với nhau. Mặc dù các ứng dụng để kết nối mọi người như mạng xã hội rất tiện ích nhưng nó cũng đem lại những cái hại như chúng ta mất quá nhiều thời gian vào mạng xã hội, bị ảnh hưởng bởi những nguồn tin tiêu cực và không chính xác... Chính vì vậy, em đã lựa chọn đề tài “Xây dựng ứng dụng Chat theo mô hình Client – Server”.

2. Mục tiêu của đề tài

Xây dựng ứng dụng Chat theo mô hình Client – Server

- Giao diện trực quan, dễ thao tác.
- Phù hợp với các nền tảng triển khai trên diện rộng, ít tốn kém khi mở rộng.
- Hệ thống dễ bảo trì và dễ phát triển.

3. Nội dung và kế hoạch thực hiện

Thời gian	Nội dung thực hiện
Từ 16-09-2022 đến 23-09-2022	Trao đổi với GVHD để thống nhất tên đề tài. Sau đó, tiến hành xây dựng đề cương chi tiết và tìm kiếm các tài liệu tham khảo cần thiết.
Từ 23-09-2022 đến 30-09-2022	Phân tích thiết kế hệ thống.
Từ 01-10-2022 đến 07-10-2022	Viết code cho các chức năng cơ bản của hệ thống.

Từ 08-10-2022 đến 30-11-2022	Kiểm thử và hoàn thiện các chức năng và giao diện của hệ thống.
Từ 01-12-2022 đến 07-12-2022	Hoàn thành đồ án, kiểm tra, sửa lỗi dự án và xây dựng tài liệu hướng dẫn sử dụng hệ thống. Nộp sản phẩm đồ án

4. Bố cục báo cáo

Sau phần Mở đầu, báo cáo được trình bày trong ba chương, cụ thể như sau:

Chương 1: Cơ sở lý thuyết.

Chương 2: Phân tích thiết kế hệ thống.

Chương 3: Chương trình Demo.

Cuối cùng là Kết luận, Tài liệu tham khảo và Phụ lục liên quan đến đề tài.

CHƯƠNG 1. CƠ SỞ LÝ THUYẾT

1.1. Sơ lược về lập trình mạng

1.1.2. Lập trình mạng

Lập trình mạng là nhiệm vụ cơ bản để phát triển các ứng dụng trong hệ thống doanh nghiệp, từ chương trình phần mềm quản lý như kế toán, nhân sự,... cho đến ứng dụng giải trí là trò chơi, điều khiển...

Lập trình mạng được xây dựng dựa trên công thức:

Lập trình mạng (LTM) = Kiến thức mạng truyền thông + Mô hình LTM + Ngôn ngữ LTM

Dựa theo công thức trên, ta có thể thấy có ba vấn đề chính cốt lõi là kiến thức mạng truyền thông, mô hình lập trình mạng và ngôn ngữ lập trình mạng.

Kiến thức mạng truyền thông là những kiến thức về mạng điện thoại di động, PSTN, hệ thống GPS, mạng như BlueTooth, WUSB, mạng sensor.... Nhất là cách sử dụng cũng như khai thác chúng.

Mô hình lập trình mạng là những kiến thức về cách xây dựng hệ thống mạng, mô hình xây dựng chương trình ứng dụng mạng, kiến thức về cơ sở dữ liệu...

Cuối cùng là ngôn ngữ lập trình mạng, đây là yếu tố quyết định xem các chương trình mạng của bạn sẽ hoạt động như thế nào. Có rất nhiều ngôn ngữ lập trình mạng khác nhau, tùy theo mục đích sử dụng.

1.1.3. Một số mô hình trong lập trình mạng

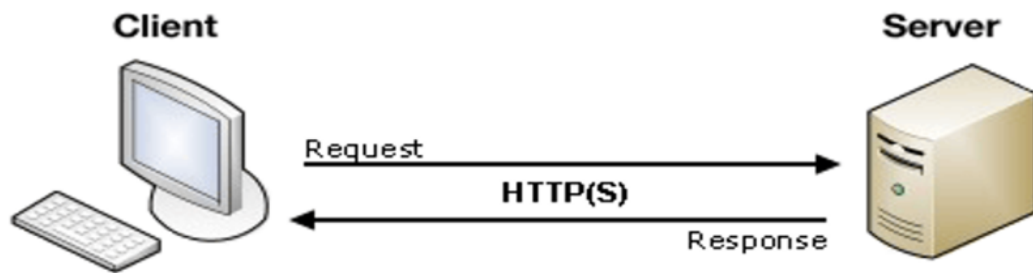
- *Mô hình hoạt động Client – Server*

Client server là mô hình mạng máy tính gồm có 2 thành phần chính đó là máy khách (client) và máy chủ (server). Server chính là nơi giúp lưu trữ tài nguyên cũng như cài đặt các chương trình dịch vụ theo đúng như yêu cầu của client. Ngược lại, Client bao gồm máy tính cũng như các loại thiết bị điện tử nói chung sẽ tiến hành gửi yêu cầu đến server.

Nguyên tắc hoạt động của Client Server

Trong mô hình Client – Server, server chấp nhận tất cả các yêu cầu hợp lệ từ mọi nơi khác nhau trên Internet, sau đó trả kết quả về máy tính đã gửi yêu cầu đó.

Máy tính được coi là máy khách khi chúng làm nhiệm vụ gửi yêu cầu đến các máy chủ và đợi câu trả lời được gửi về.



Để máy khách và máy chủ có thể giao tiếp được với nhau thì giữa chúng phải có một chuẩn nhất định, và chuẩn đó được gọi là giao thức. Một số giao thức được sử dụng phổ biến hiện nay như: HTTPS, TCP/IP, FTP, ...

Nếu máy khách muốn lấy được thông tin từ máy chủ, chúng phải tuân theo một giao thức mà máy chủ đó đưa ra. Nếu yêu cầu đó được chấp nhận thì máy chủ sẽ thu thập thông tin và trả về kết quả cho máy khách yêu cầu. Bởi vì Server – máy chủ luôn luôn trong trạng thái sẵn sàng để nhận request từ Client nên chỉ cần client gửi yêu cầu tín hiệu và chấp nhận yêu cầu đó thì Server sẽ trả kết quả về phía Client trong thời gian ngắn nhất.

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. Bởi vì, nguyên lý hoạt động của Client server là trao đổi dữ liệu giữa server và client ở 2 khu vực địa lý khác nhau. Trong quá trình trao đổi dữ liệu, khả năng thông tin mạng bị lộ là điều dễ xảy ra.

- Mô hình Peer to peer

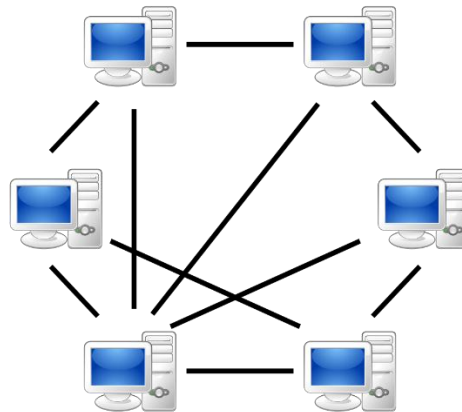
P2P – Peer-to-peer (mạng ngang hàng) là một mô hình mạng phi tập trung với các bên có các cấu trúc phiên giao tiếp giống nhau. Trong đó, mỗi nút hoạt động giống như một máy khách và máy chủ của hệ thống cho phép chia sẻ các phương tiện truyền thông với nhau dễ dàng hơn, nhanh chóng hơn.

P2P có nhiều tính năng như cung cấp môi trường tính toán song song, lưu trữ phân tán và định tuyến ẩn danh lưu lượng mạng. Bởi vì có khả năng chia sẻ phương tiện truyền thông nên P2P hay bị vi phạm bản quyền và vi phạm bản quyền phần mềm.

Hầu hết các ứng dụng P2P cho phép người dùng kiểm soát các thông số hoạt động như cho phép kết nối nhiều thành viên một lúc. Hay có hệ thống kết nối, dịch vụ cung cấp và các tài nguyên bảo vệ dành cho mạng.

Từ thời ARPANET các kiến trúc liên kết P2P đã được vận hành nhưng chưa được phổ biến. Cho đến cuối những năm 1990 mô hình truyền thông P2P và những lợi ích của nó mới thực sự được công nhận.

Mạng P2P vận hành như thế nào?



Thông thường, người dùng sẽ phải truy cập vào trình duyệt web của mình để tìm kiếm và tải xuống tệp mong muốn. Trang web lúc này giống như một máy chủ và máy tính sẽ giống như một máy khách chuyên nhận các dữ liệu. Hiểu đơn giản quá trình này giống như đường một chiều từ A đến B. Trong đó, vị trí của tệp tải xuống là điểm A còn máy tính chính là điểm B.

Còn với mạng P2P quá trình này sẽ được xử lý bằng quy trình khác. Trước tiên phần mềm P2P muốn vận hành được thì cần cài vào máy của người dùng. Khi đó những người dùng sẽ được đưa vào cùng một mạng ảo do P2P tạo ra.

Sau khi tải xuống, một tệp mạng ảo sẽ được nhận dưới dạng các bit đến từ các máy tính khác nhau đã được tạo sẵn. Đồng thời, máy tính nào yêu cầu thì sẽ nhận được dữ liệu từ máy tính của người dùng. Trường hợp này P2P hoạt động giống như đường hai chiều cho phép công việc truyền tải tệp được phân phối linh hoạt hơn.

Kiến trúc của P2P

Trong kiến trúc P2P thì mỗi máy tính sẽ có các nhiệm vụ và tính năng hoạt động như nhau. Các máy tính sẽ kết nối trực tiếp với nhau tạo thành một nhóm làm việc nhỏ tối đa 12 thiết bị có nhiệm vụ chia sẻ tệp, máy in và truy cập Internet. Vì vậy mà P2P được sử dụng phổ biến trong phạm vi làm việc nhỏ như gia đình, văn phòng hoặc trường học có các PC hoạt động như một máy trạm độc lập. Cho phép lưu trữ các dữ liệu trên ổ cứng riêng và có khả năng chia sẻ các dữ liệu đó tới các PC khác trên cùng một mạng.

- **Mô hình RMI**

RMI - Remote Method Invocation là một kỹ thuật cài đặt các đối tượng phân tán trong Java. RMI là một phần của bộ J2SDK và là hàm thư viện hỗ trợ các lời gọi phương thức từ xa và trả về giá trị cho các ứng dụng tính toán phân tán. Chúng ta giả sử rằng ngôn ngữ Java được sử dụng ở cả hai phía gọi và phía bên phương thức được gọi.

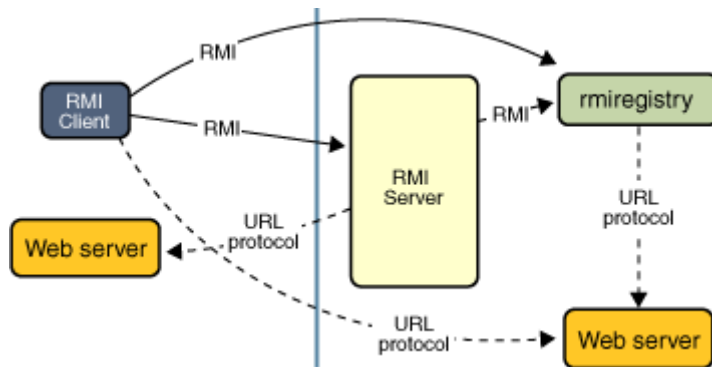
Đặc tính của RMI

- + RMI là mô hình đối tượng phân tán của Java, RMI giúp cho việc giao tiếp giữa các đối tượng phân tán trong môi trường internet trở nên dễ dàng hơn.
- + RMI là API bậc cao được xây dựng dựa trên lập trình Socket.
- + RMI không những cho phép chúng ta truyền dữ liệu giữa các đối tượng trên các hệ thống máy tính khác nhau, mà còn triệu gọi các phương thức trong các đối tượng ở xa (Remote Object).
- + Việc truyền dữ liệu giữa các máy khác nhau được xử lý một cách trong suốt bởi máy ảo Java (Java virtual machine).
- + Tương tự như mô hình Client/Server, RMI vẫn lấy/duy trì khái niệm của Client và Server, tuy nhiên cách tiếp cận (approach) của RMI linh hoạt hơn, mềm dẻo hơn so với mô hình Client/Server.
- + Một điều thuận lợi quan trọng nhất của RMI là nó cung cấp cơ chế callbacks, nó cho phép Server triệu gọi các phương thức ở Client.

RMI Architecture

- + Remote interface: Nên extend từ java.rmi.remote. Nó khai báo tất cả các phương thức mà Client có thể triệu gọi. Tất cả các method trong interface này nên throw RemoteException
- + Remote implementation: Được thực thi từ Remote interface và mở rộng từ UnicastRemoteObject. Triển khai các method được khai báo trong Interface tại đây. Nó là một Remote Object thực sự. Phát sinh hai lớp trung gian Stub và Skel.
- + Server class bao gồm:

- RMI registry: Bộ đăng kí này sẽ đăng kí một Remote object với Naming Registry. Giúp các Remote object được chấp nhận khi gọi các method từ xa.
- Các class thực thi trên server.
- Client class: Truy vấn trên tên Remote object trên RMI registry, thông qua stub để gọi các phương thức trên server.

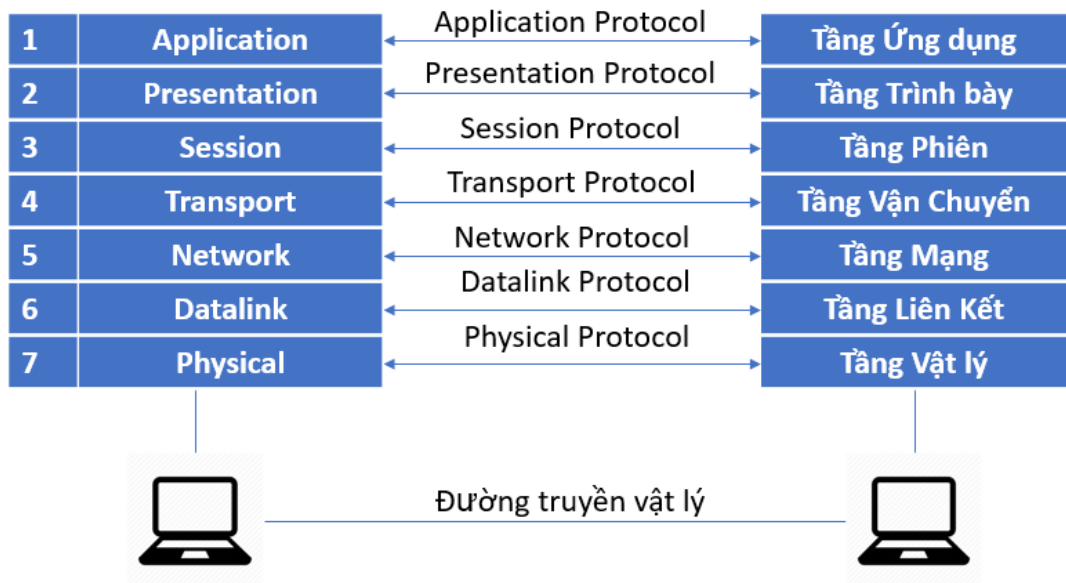


1.1.4. Một số giao thức trong lập trình mạng

- Mô hình OSI

Mô hình kết nối các hệ thống mở OSI là mô hình căn bản về các tiến trình truyền thông, thiết lập các tiêu chuẩn kiến trúc mạng ở mức Quốc tế, là cơ sở chung để các hệ thống khác nhau có thể liên kết và truyền thông được với nhau. Mô hình OSI tổ chức các giao thức truyền thông thành 7 tầng, mỗi một tầng giải quyết một phần hẹp của tiến trình truyền thông, chia tiến trình truyền thông thành nhiều tầng và trong mỗi tầng có thể có nhiều giao thức khác nhau thực hiện các nhu cầu truyền thông cụ thể.

Nguyên tắc định nghĩa các tầng hệ thống mở



Mô hình OSI tuân theo các nguyên tắc phân tầng như sau:

- + Mô hình gồm $N = 7$ tầng. OSI là hệ thống mở, phải có khả năng kết nối với các hệ thống khác nhau, tương thích với các chuẩn OSI.
- + Quá trình xử lý các ứng dụng được thực hiện trong các hệ thống mở, trong khi vẫn duy trì được các hoạt động kết nối giữa các hệ thống.
- + Thiết lập kênh logic nhằm mục đích thực hiện việc trao đổi thông tin giữa các thực thể.

Các giao thức trong mô hình OSI

Trong mô hình OSI có hai loại giao thức được sử dụng: giao thức hướng liên kết (Connection – Oriented) và giao thức không liên kết (Connectionless).

- + Giao thức hướng liên kết

Trước khi truyền dữ liệu, các thực thể đồng tầng trong hai hệ thống cần phải thiết lập một liên kết logic. Chúng thương lượng với nhau về tập các tham số sẽ sử dụng trong giai đoạn truyền dữ liệu, cắt/hợp dữ liệu, liên kết sẽ được hủy bỏ. Thiết lập liên kết logic sẽ nâng cao độ tin cậy và an toàn trong quá trình trao đổi dữ liệu.

- + Giao thức không liên kết

Dữ liệu được truyền độc lập trên các tuyến khác nhau. Với các giao thức không liên kết chỉ có giai đoạn duy nhất truyền dữ liệu.

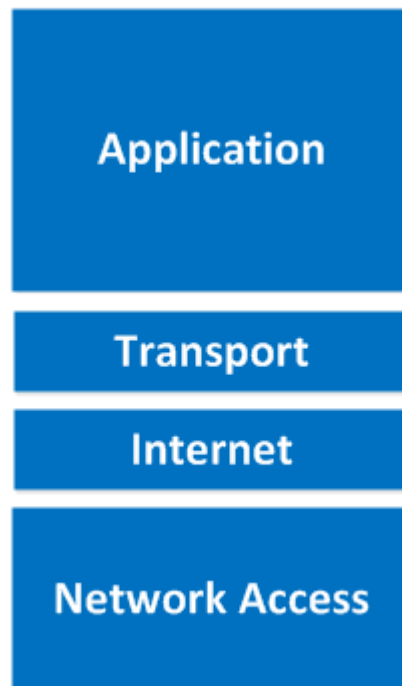
Vai trò và chức năng chủ yếu các tầng.

Tầng	Chức năng chủ yếu	Giao thức
7 – Application	Giao tiếp người và môi trường mạng	Ứng dụng
6 – Presentation	Chuyển đổi cú pháp dữ liệu để đáp ứng yêu cầu truyền thông của các ứng dụng	Giao thức Biến đổi mã
5 - Session	Quản lý các cuộc liên lạc giữa các thực thể bằng cách thiết lập, duy trì, đồng bộ hóa và hủy bỏ các phiên truyền thông giữa các ứng dụng	Giao thức phiên
4 – Transport	Vận chuyển thông tin giữa các máy chủ (End to End). Kiểm soát lỗi và luồng dữ liệu	Giao thức Giao vận
3 – Network	Thực hiện chọn đường và đảm bảo trao đổi thông tin trong liên mạng với công nghệ chuyển mạch thích hợp.	Giao thức mạng
2 – Data Link	Tạo/gỡ bỏ khung thông tin (Frames), kiểm soát luồng và kiểm soát lỗi.	Thủ tục kiểm soát
1 - Physical	Đảm bảo các yêu cầu truyền/nhận các chuỗi bit qua các phương tiện vật lý.	Giao diện DTE - DCE

- Mô hình TCP/IP

TCP/ IP (Transmission Control Protocol/ Internet Protocol - Giao thức điều khiển truyền nhận/ Giao thức liên mạng), là một bộ giao thức trao đổi thông tin được sử dụng để truyền tải và kết nối các thiết bị trong mạng Internet. TCP/IP được phát triển để mạng được tin cậy hơn cùng với khả năng phục hồi tự động.

TCP/IP Stack



Sự phát triển và hình thành của mô hình TCP/IP

Ý tưởng hình thành mô hình TCP/IP được bắt nguồn từ Bộ giao thức liên mạng trong công trình DARPA vào năm 1970. Trải qua vô số năm nghiên cứu và phát triển của 2 kỹ sư Robert E. Kahn và Vinton Cerf cùng sự hỗ trợ của không ít các nhóm nghiên cứu. Đầu năm 1978, giao thức TCP/ IP được ổn định hóa với giao thức tiêu chuẩn được dùng hiện nay của Internet đó là mô hình TCP/IP Version 4.

Vào năm 1975, cuộc thử nghiệm thông nối giữa 2 mô hình TCP/IP được diễn ra thành công. Cũng bắt đầu từ đây, cuộc thử nghiệm thông nối giữa các mô hình TCP/IP được diễn ra nhiều hơn và đều đạt được kết quả tốt. Cũng chính vì điều này, một cuộc hội thảo được Internet Architecture Broad mở ra, với sự tham dự của hơn 250 đại biểu của các công ty thương mại, từ đây giao thức và mô hình TCP/IP được phổ biến rộng rãi trên khắp thế giới.

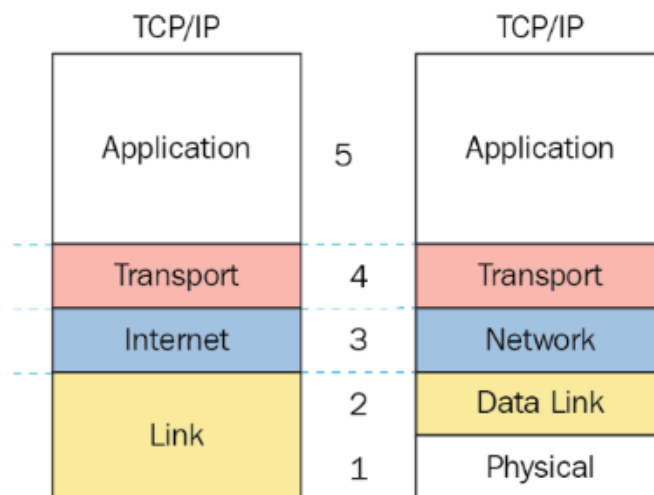
Cách thức hoạt động của mô hình TCP/IP

Phân tích từ tên gọi, TCP/IP là sự kết hợp giữa 2 giao thức. Trong đó [IP](#) (Giao thức liên mạng) cho phép các gói tin được gửi đến đích đã định sẵn, bằng cách thêm các thông tin dẫn đường vào các gói tin để các gói tin được đến đúng đích đã định sẵn ban đầu. Và

giao thức TCP (Giao thức truyền vận) đóng vai trò kiểm tra và đảm bảo sự an toàn cho mỗi gói tin khi đi qua mỗi trạm. Trong quá trình này, nếu giao thức TCP nhận thấy gói tin bị lỗi, một tín hiệu sẽ được truyền đi và yêu cầu hệ thống gửi lại một gói tin khác. Quá trình hoạt động này sẽ được làm rõ hơn ở chức năng của mỗi tầng trong mô hình TCP/IP.

Chức năng của các tầng trong mô hình TCP/IP

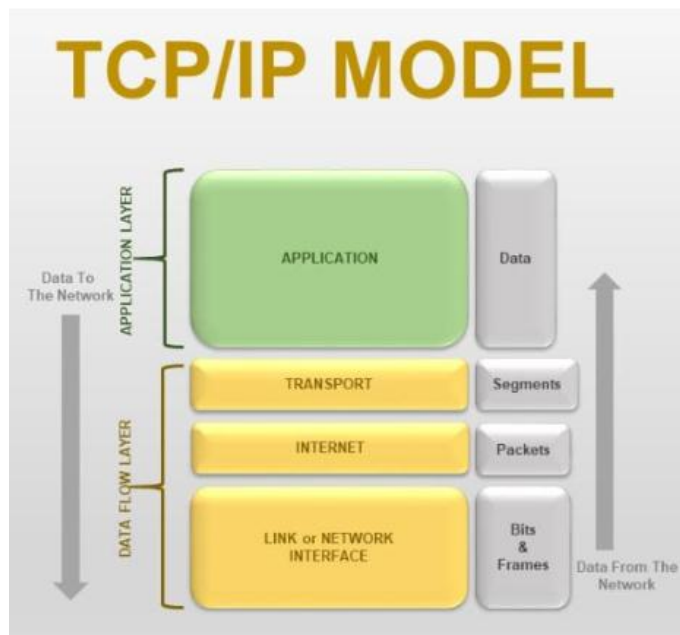
Một mô hình TCP/IP tiêu chuẩn bao gồm 4 lớp được chồng lên nhau, bắt đầu từ tầng thấp nhất là Tầng vật lý (Physical) → Tầng mạng (Network) → Tầng giao vận (Transport) và cuối cùng là Tầng ứng dụng (Application).



Tuy nhiên, một số ý kiến lại cho rằng mô hình TCP/IP là 5 tầng, tức các tầng 4 đến 2 đều được giữ nguyên, nhưng tầng Datalink sẽ được tách riêng và là tầng nằm trên so với tầng vật lý.

+ Tầng 4 - Tầng Ứng dụng (Application)

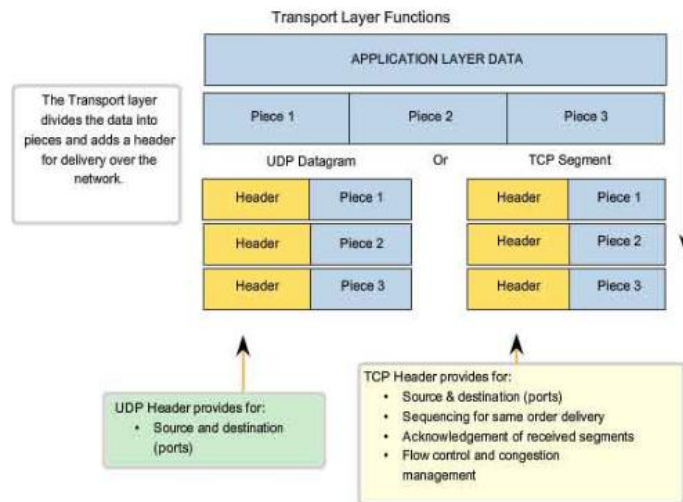
Đây là lớp giao tiếp trên cùng của mô hình. Đúng với tên gọi, tầng Ứng dụng đảm nhận vai trò giao tiếp dữ liệu giữa 2 máy khác nhau thông qua các dịch vụ mạng khác nhau (duyet web, chat, gửi email, một số giao thức trao đổi dữ liệu: SMTP, SSH, **FTP**,...). Dữ liệu khi đến đây sẽ được định dạng theo kiểu Byte nối Byte, cùng với đó là các thông tin định tuyến giúp xác định đường đi đúng của một gói tin.



+ Tầng 3 - Tầng Giao vận (Transport)

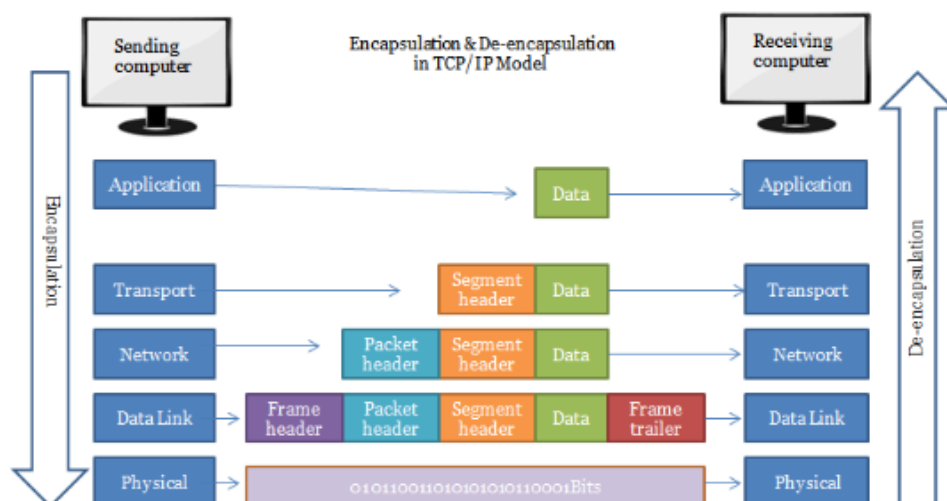
Chức năng chính của tầng 3 là xử lý vấn đề giao tiếp giữa các máy chủ trong cùng một mạng hoặc khác mạng được kết nối với nhau thông qua bộ định tuyến. Tại đây dữ liệu sẽ được phân đoạn, mỗi đoạn sẽ không bằng nhau nhưng kích thước phải nhỏ hơn 64KB. Cấu trúc đầy đủ của một Segment lúc này là Header chứa thông tin điều khiển và sau đó là dữ liệu.

Trong tầng này còn bao gồm 2 giao thức cốt lõi là TCP và UDP. Trong đó, TCP đảm bảo chất lượng gói tin nhưng tiêu tốn thời gian khá lâu để kiểm tra đầy đủ thông tin từ thứ tự dữ liệu cho đến việc kiểm soát vấn đề tắc nghẽn lưu lượng dữ liệu. Trái với điều đó, UDP cho thấy tốc độ truyền tải nhanh hơn nhưng lại không đảm bảo được chất lượng dữ liệu được gửi đi.



+ Tầng 2 - Tầng mạng (Internet)

Gần giống như tầng mạng của mô hình OSI. Tại đây, nó cũng được định nghĩa là một giao thức chịu trách nhiệm truyền tải dữ liệu một cách logic trong mạng. Các phân đoạn dữ liệu sẽ được đóng gói (Packets) với kích thước mỗi gói phù hợp với mạng chuyển mạch mà nó dùng để truyền dữ liệu. Lúc này, các gói tin được chèn thêm phần Header chứa thông tin của tầng mạng và tiếp tục được chuyển đến tầng tiếp theo. Các giao thức chính trong tầng là IP, ICMP và ARP.



+ Tầng 1 - Tầng Vật lý (Physical)

Là sự kết hợp giữa tầng Vật lý và tầng liên kết dữ liệu của mô hình OSI. Chịu trách nhiệm truyền dữ liệu giữa hai thiết bị trong cùng một mạng. Tại đây, các gói dữ liệu được đóng vào khung (gọi là Frame) và được định tuyến đi đến đích đã được chỉ định ban đầu.

1.2. Sơ lược về ngôn ngữ lập trình Java

1.2.1. Giới thiệu về ngôn ngữ lập trình Java

Java là một trong những ngôn ngữ lập trình hướng đối tượng. Nó được sử dụng trong phát triển phần mềm, trang web, game hay ứng dụng trên các thiết bị di động.

Java được khởi đầu bởi James Gosling và bạn đồng nghiệp ở Sun MicroSystem năm 1991. Ban đầu Java được tạo ra nhằm mục đích viết phần mềm cho các sản phẩm gia dụng, và có tên là Oak.

Java được phát hành năm 1994, đến năm 2010 được Oracle mua lại từ Sun MicroSystem.

Java được tạo ra với tiêu chí “Viết (code) một lần, thực thi khắp nơi” (Write Once, Run Anywhere – WORA). Chương trình phần mềm viết bằng Java có thể chạy trên mọi nền tảng (platform) khác nhau thông qua một môi trường thực thi với điều kiện có môi trường thực thi thích hợp hỗ trợ nền tảng đó.

1.2.2. Đặc điểm của ngôn ngữ lập trình Java

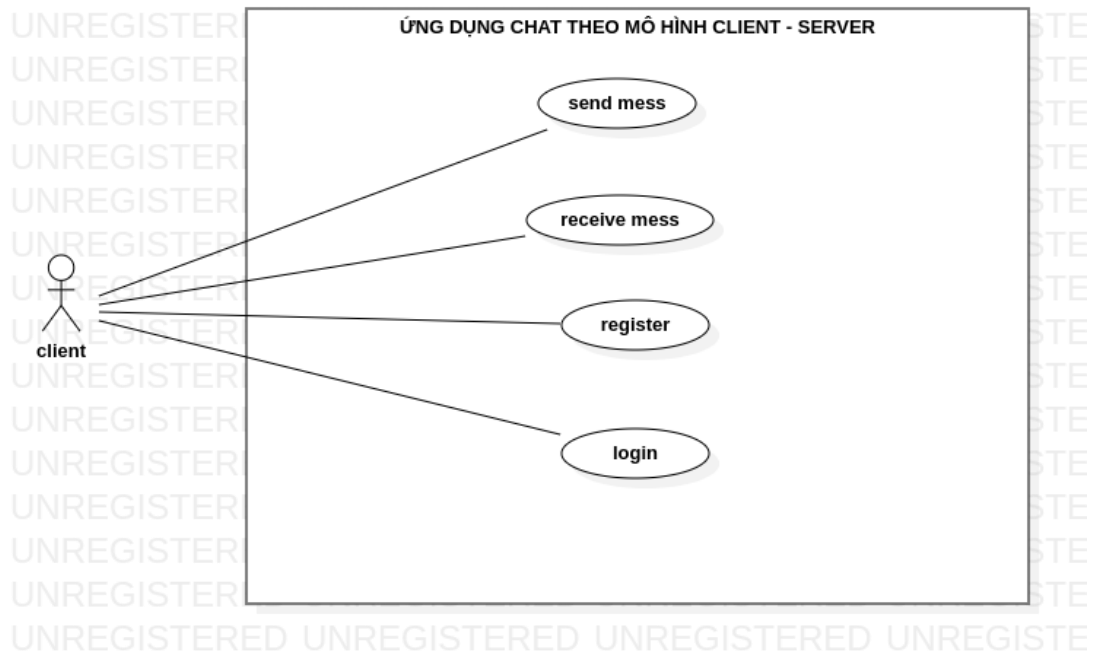
Là một ngôn ngữ lập trình hướng đối tượng vì vậy Java cũng có 4 đặc điểm chung sau đây:

- Tính trừu tượng (Abstraction): là tiến trình xác định và nhóm các thuộc tính, các hành động liên quan đến một thực thể đặc thù, xét trong mối tương quan với ứng dụng đang phát triển.
- Tính đa hình (Polymorphism): cho phép một phương thức có các tác động khác nhau trên nhiều loại đối tượng khác nhau. Với tính đa hình, nếu cùng một phương thức ứng dụng cho các đối tượng thuộc các lớp khác nhau thì nó đưa đến những kết quả khác nhau. Bản chất của sự việc chính là phương thức này bao gồm cùng một số lượng các tham số.

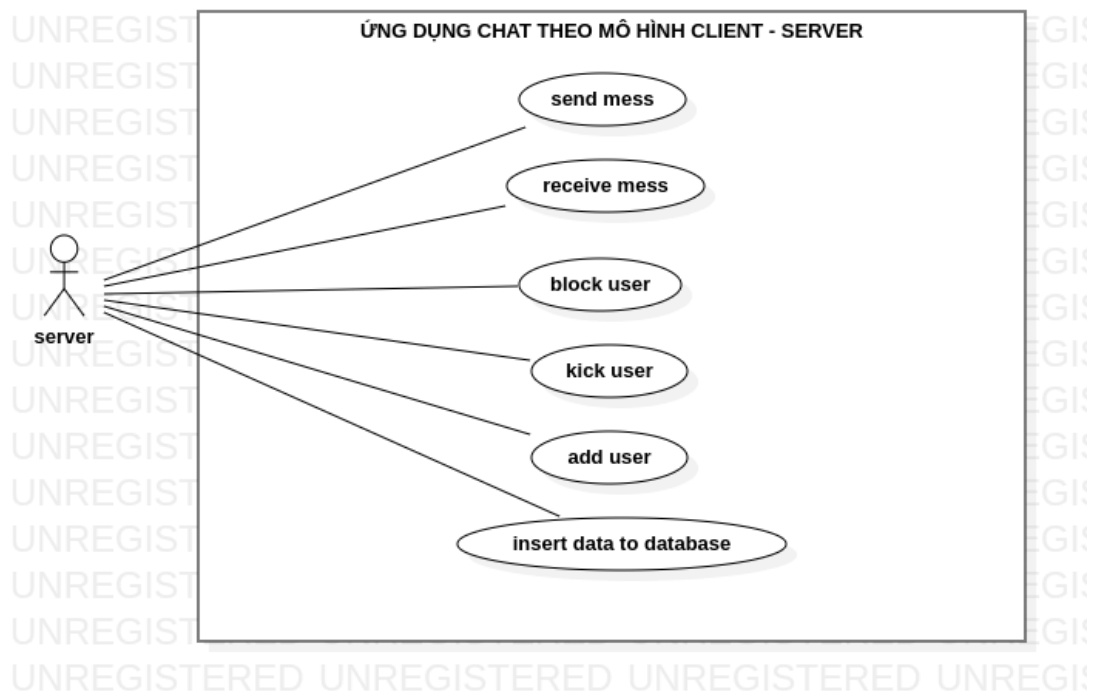
- Tính kế thừa (Inheritance): điều này cho phép các đối tượng chia sẻ hay mở rộng các đặc tính sẵn có mà không phải tiến hành định nghĩa lại.
- Tính đóng gói (Encapsulation): là tiến trình che giấu việc thực thi những chi tiết của một đối tượng đối với người sử dụng đối tượng ấy.
- Ngoài ra Java còn có một số đặc điểm sau:
- Độc lập nền (Write Once, Run Anywhere): Không giống như nhiều ngôn ngữ lập trình khác như C và C++, khi Java được biên dịch, nó không được biên dịch sang mã máy cụ thể, mà thay vào đó là mã bytecode chạy trên máy ảo Java (JVM). Điều này đồng nghĩa với việc bất cứ thiết bị nào có cài đặt JVM sẽ có thể thực thi được các chương trình Java.
- Đơn giản: Học Java thật sự dễ hơn nhiều so với C/C++, nếu bạn đã quen với các ngôn ngữ lập trình hướng đối tượng thì việc học Java sẽ dễ dàng hơn. Java trở nên đơn giản hơn so với C/C++ do đã loại bỏ tính đa kế thừa và phép toán con trỏ từ C/C++.
- Bảo mật: Java hỗ trợ bảo mật rất tốt bởi các thuật toán mã hóa như mã hóa một chiều (one way hashing) hoặc mã hóa công cộng (public key)...
- Thông dịch: Java là một ngôn ngữ lập trình vừa biên dịch vừa thông dịch. Chương trình nguồn viết bằng ngôn ngữ lập trình Java có đuôi *.java và được biên dịch thành tập tin có đuôi *.class sau đó được trình thông dịch thông dịch thành mã máy.
- Đa luồng: Với tính năng đa luồng Java có thể viết chương trình có thể thực thi nhiều task cùng một lúc. Tính năng này thường được sử dụng rất nhiều trong lập trình game.
- Hướng đối tượng: Hướng đối tượng trong Java tương tự như C++ nhưng Java là một ngôn ngữ lập trình hướng đối tượng hoàn toàn. Tất cả mọi thứ đề cập đến trong Java đều liên quan đến các đối tượng được định nghĩa trước, thậm chí hàm chính của một chương trình viết bằng Java (đó là hàm main) cũng phải đặt bên trong một lớp. Hướng đối tượng trong Java không có tính

đa kế thừa (multi inheritance) như trong C++ mà thay vào đó Java đưa ra khái niệm interface để hỗ trợ tính đa kế thừa.

- Hiệu suất cao: Nhờ vào trình thu gom rác (garbage collection), giải phóng bộ nhớ đối với các đối tượng không được dùng đến.
- Linh hoạt: Java được xem là linh hoạt hơn C/C++ vì nó được thiết kế để thích ứng với nhiều môi trường phát triển.

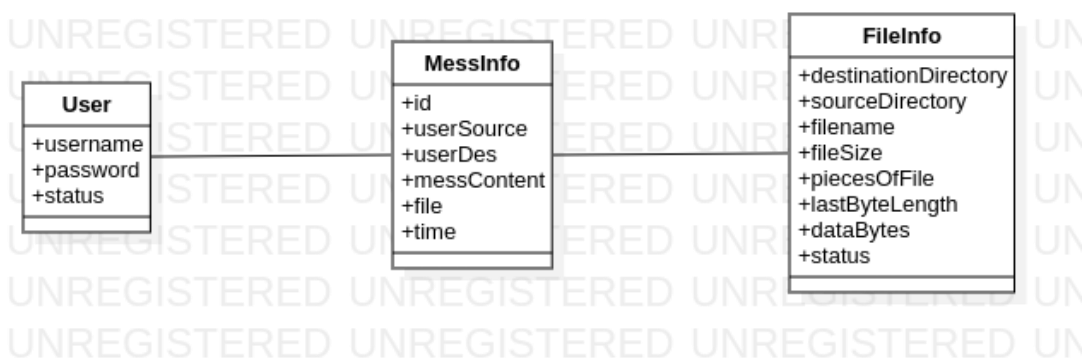


Hình 2. 2 usecase diagram client



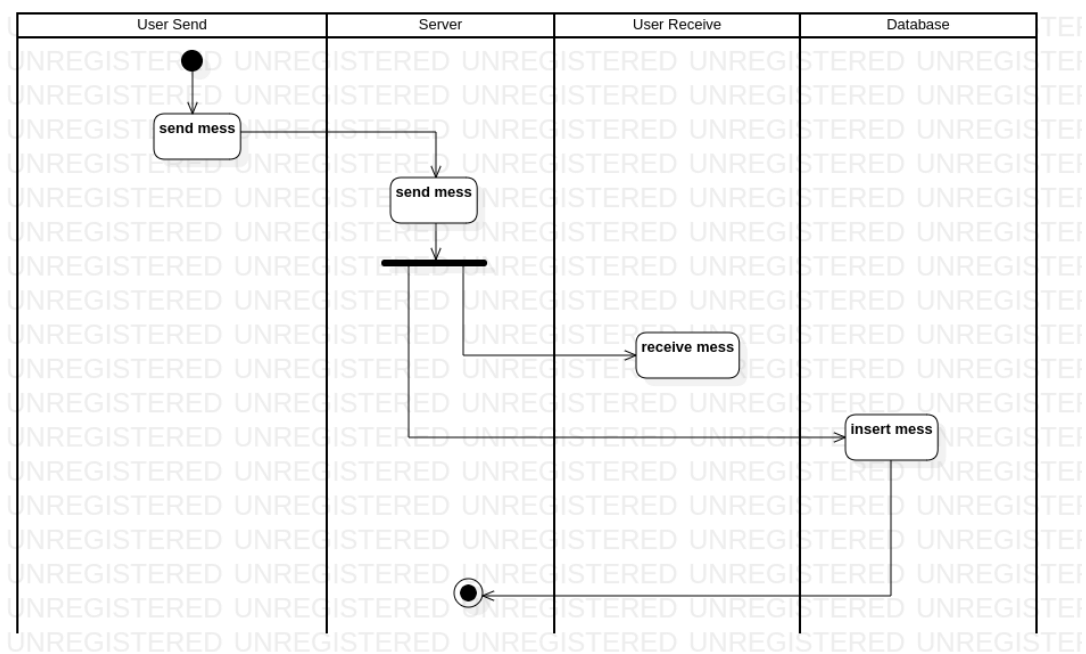
Hình 2. 3 usecase diagraphme server

2.2.2. Class Diagram



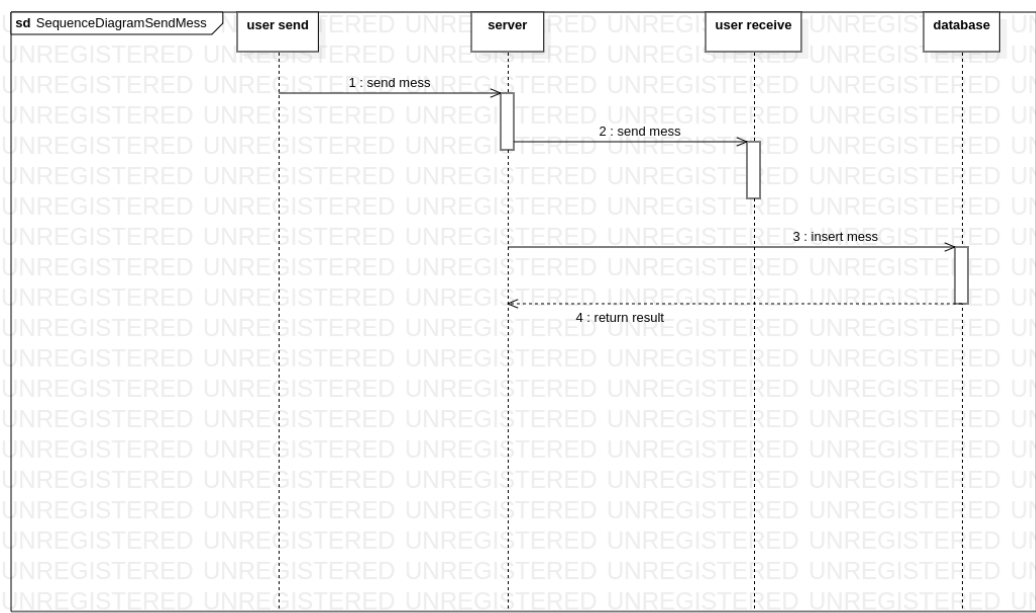
Hình 2. 4 class diagram

2.2.3. Activity Diagram



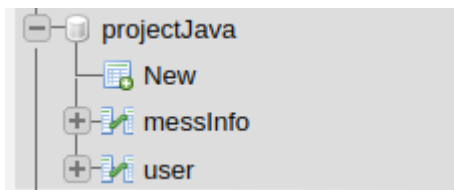
Hình 2. 5 Activity Diagram Send Mess

2.2.4. Sequence Diagram



Hình 2. 6 Sequence Diagram Send Mess

2.3. Cơ sở dữ liệu



Hình 2. 7 Danh sách các bảng của cơ sở dữ liệu

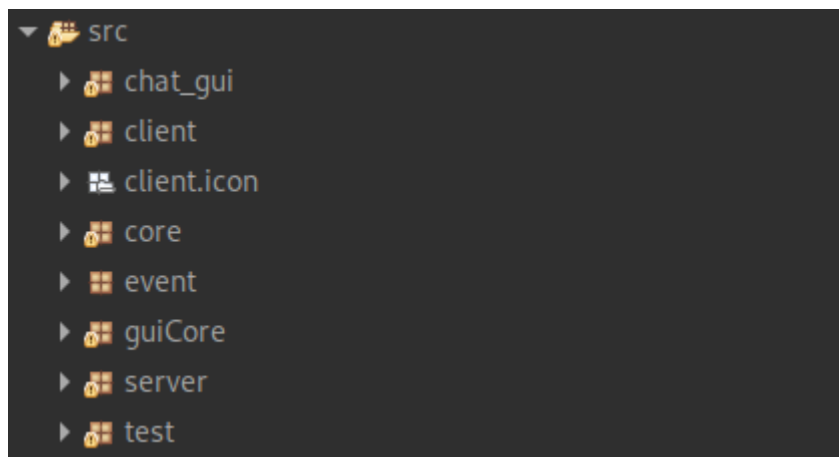
	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/>	1	id	int(11)			No	None		AUTO_INCREMENT	Change Drop More
<input type="checkbox"/>	2	userSource	varchar(50)	utf8mb4_unicode_ci		No	None			Change Drop More
<input type="checkbox"/>	3	userDes	varchar(50)	utf8mb4_unicode_ci		No	None			Change Drop More
<input type="checkbox"/>	4	messContent	text	utf8mb4_unicode_ci		No	None			Change Drop More
<input type="checkbox"/>	5	file	text	utf8mb4_unicode_ci		Yes	NULL			Change Drop More
<input type="checkbox"/>	6	time	datetime			No	None			Change Drop More

Hình 2. 8 Cấu trúc bảng messInfo

	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/>	1	username 🔑	varchar(50)	utf8mb4_unicode_ci		No	None			Change Drop More
<input type="checkbox"/>	2	password	varchar(50)	utf8mb4_unicode_ci		No	None			Change Drop More
<input type="checkbox"/>	3	status	int(11)			No	None			Change Drop More

Hình 2. 9 Cấu trúc bảng user

2.4. Cấu trúc thư mục



Hình 2. 10 Cấu trúc code

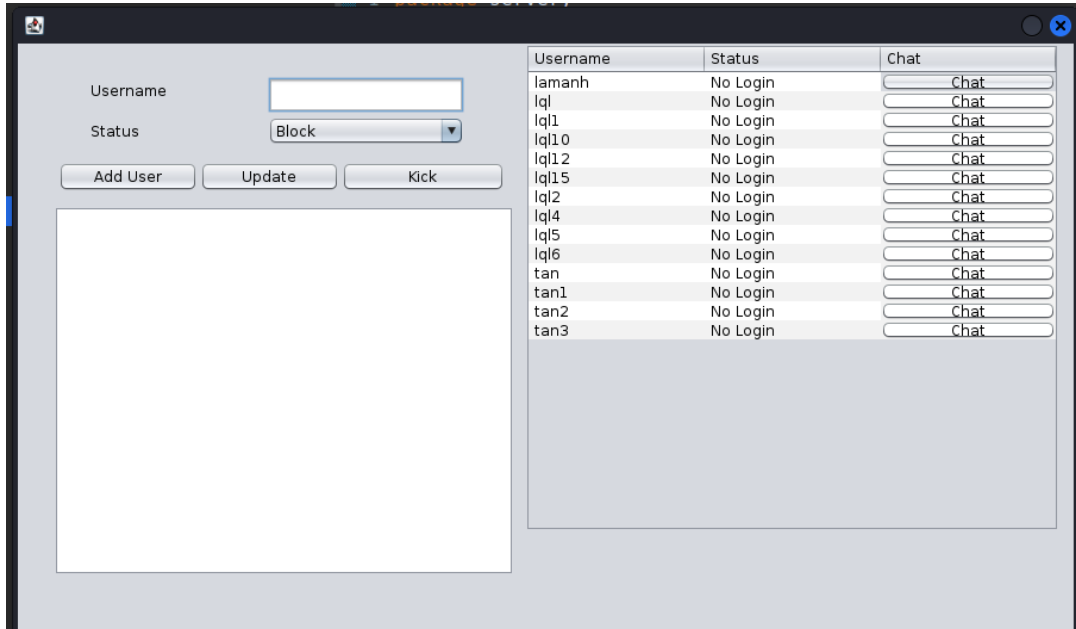
Cấu trúc lưu trữ bao gồm Server: Lưu trữ file khi được client gửi lên, Public nơi lưu trữ các file dùng để backup như .sql, chat_gui_swing là nơi lưu trữ source code.

Server	4.0 KiB	folder	11/25/2022
Public	4.0 KiB	folder	11/25/2022
chat_gui_swing	4.0 KiB	folder	11/09/2022

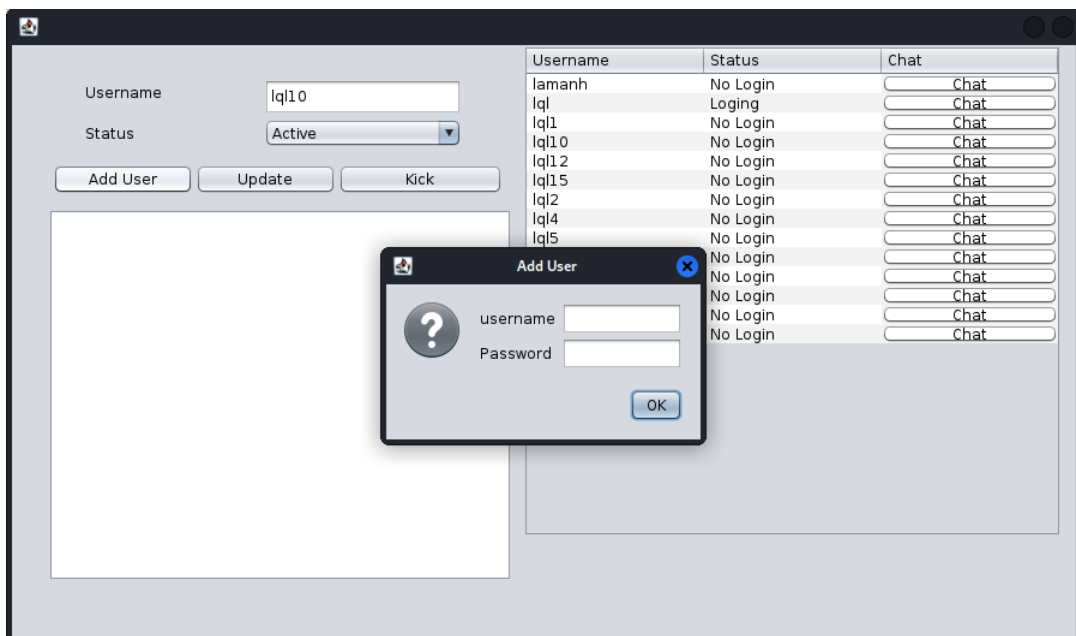
Hình 2. 11 Cấu trúc lưu trữ

CHƯƠNG 3. CHƯƠNG TRÌNH DEMO

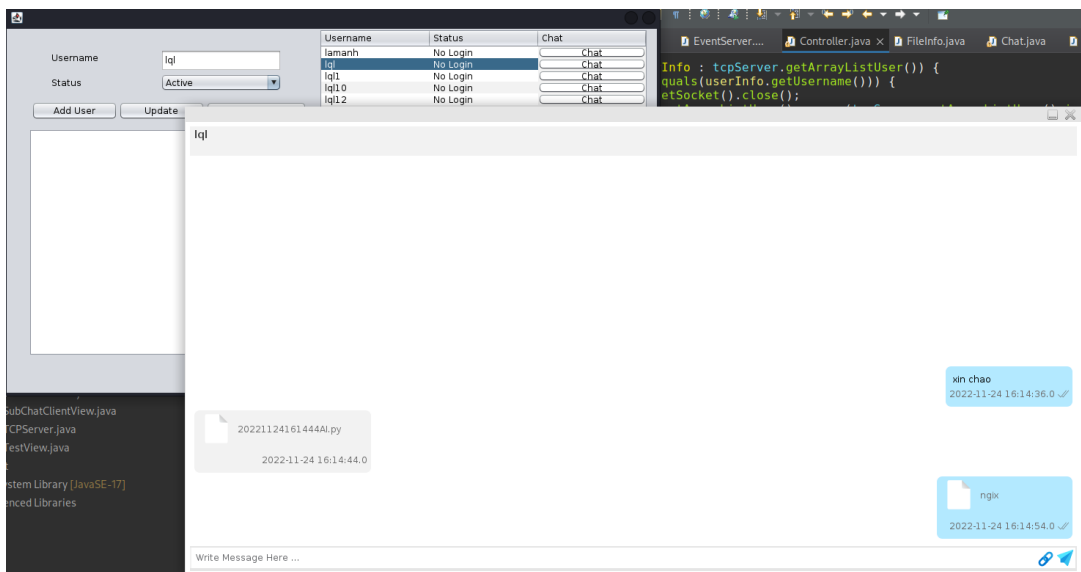
3.1. Giao diện Server



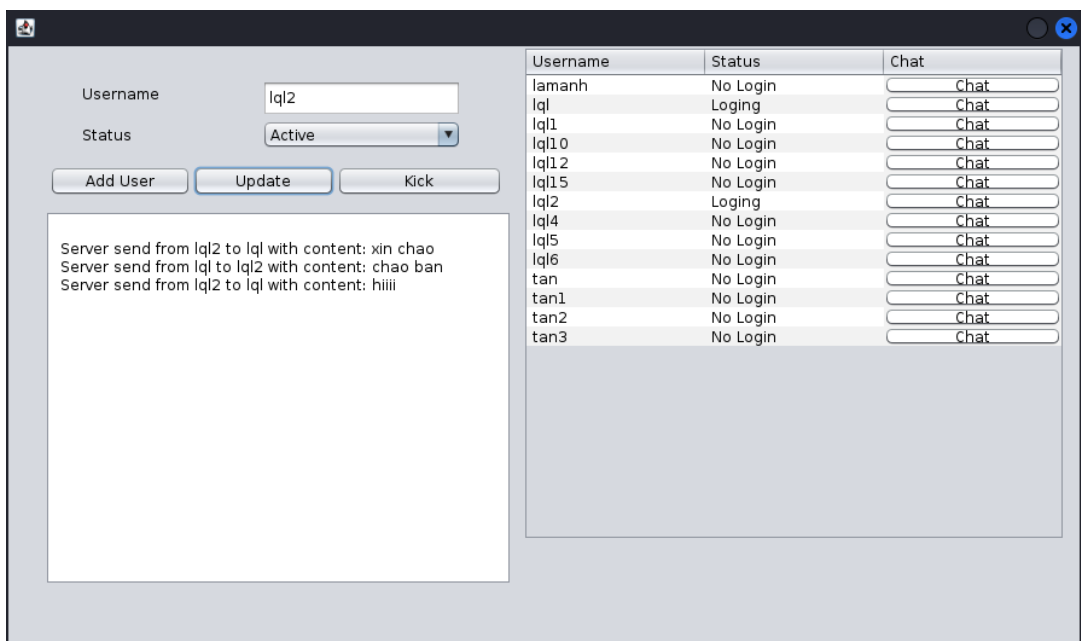
Hình 3. 1 Giao diện chính của server



Hình 3. 2 Chức năng thêm user

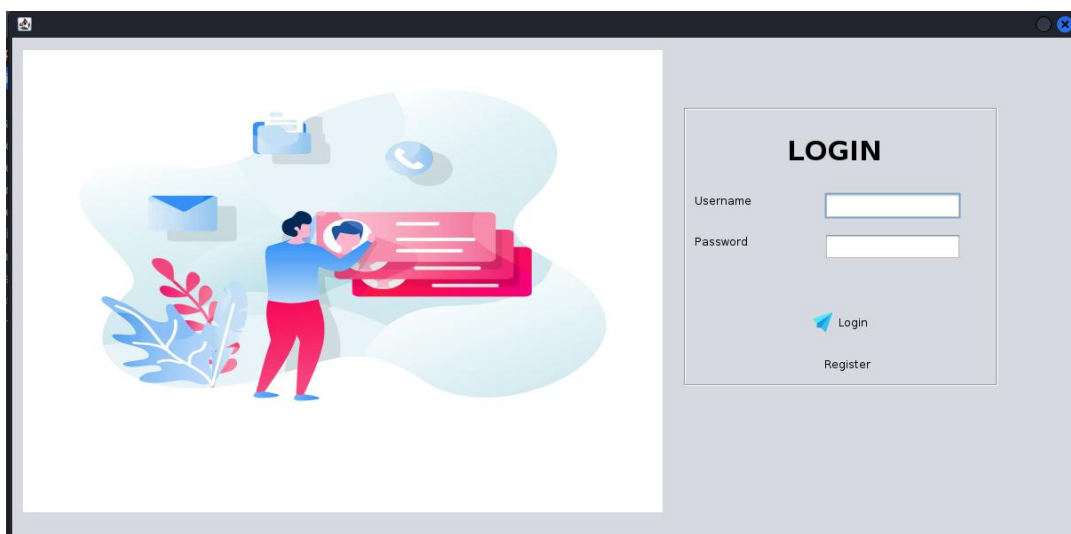


Hình 3. 3 Chức năng chat với user



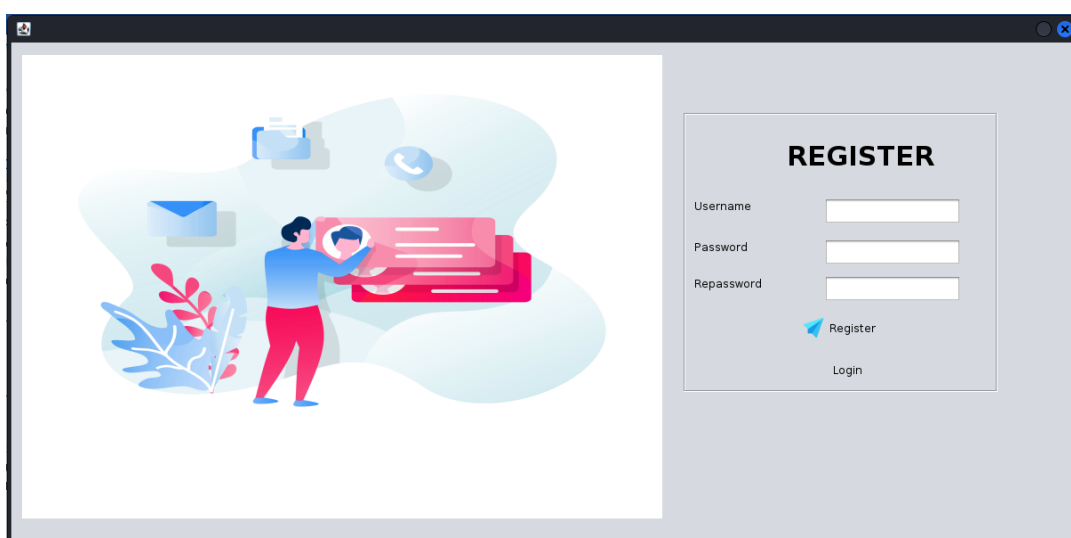
Hình 2. 12 Server ghi lại log và insert mess lên database

3.2. Giao diện Client



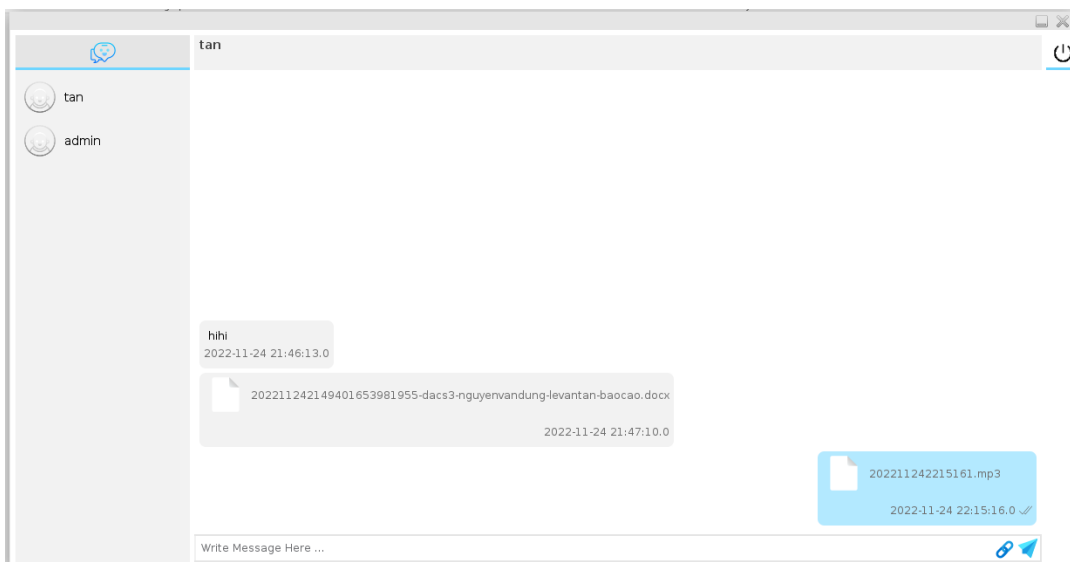
The screenshot shows a web application window with a light gray background. On the left, there is a large illustration of a person in a blue shirt and red pants interacting with a red login card. The card has a white circle with a person icon. Above the card are icons for a folder, a phone, and an envelope. On the right, there is a white box with the title "LOGIN" in bold. Below the title, there are two input fields labeled "Username" and "Password". At the bottom of the box, there are two buttons: "Login" with a blue arrow icon and "Register".

Hình 3. 4 Giao diện đăng nhập

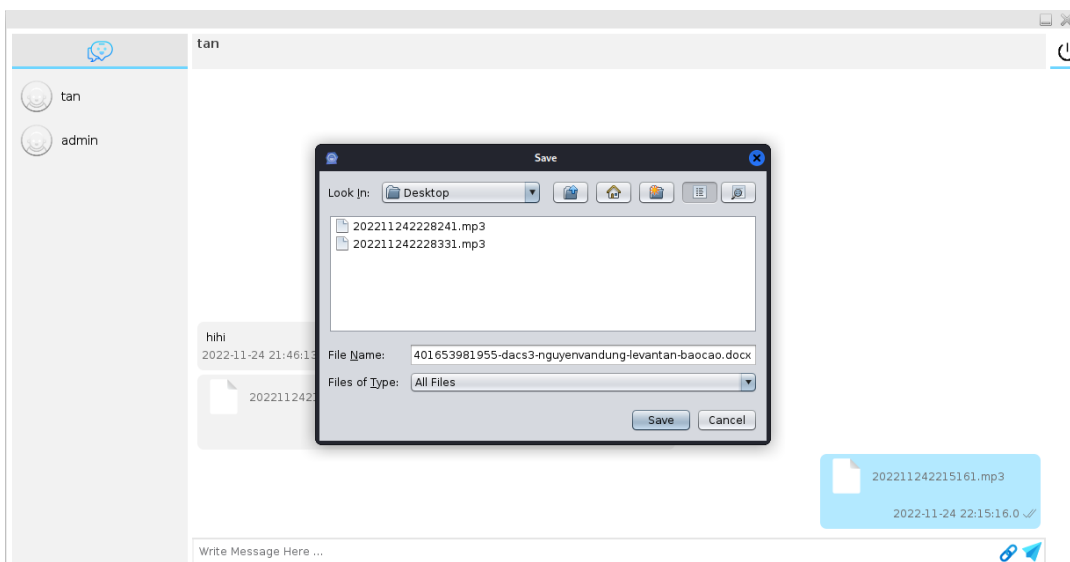


The screenshot shows a web application window with a light gray background. On the left, there is a large illustration of a person in a blue shirt and red pants interacting with a red login card. The card has a white circle with a person icon. Above the card are icons for a folder, a phone, and an envelope. On the right, there is a white box with the title "REGISTER" in bold. Below the title, there are three input fields labeled "Username", "Password", and "Repassword". At the bottom of the box, there are two buttons: "Register" with a blue arrow icon and "Login".

Hình 3. 5 Giao diện đăng ký



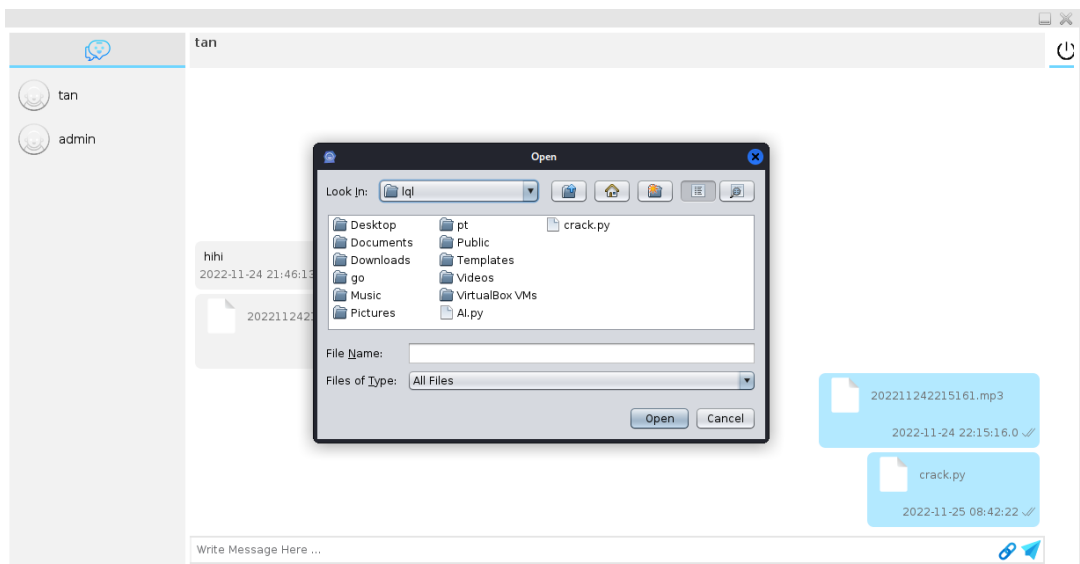
Hình 3. 6 Giao diện chat



Hình 3. 7 Chọn nơi lưu file

	20221125084150202211242149401653981955-dacs...	7.2 MiB	Microsoft Word Document	Today
	202211242228331.mp3	357.6 KiB	MP3 audio	Yesterday
	202211242228241.mp3	357.6 KiB	MP3 audio	Yesterday

Hình 3. 8 Kết quả sau khi lưu file



Hình 3. 9 Chọn file để gửi

20221125084222crack.py	673 bytes	Python script	Today
1.mp3	357.6 KiB	MP3 audio	Yesterday
202211242215161.mp3	357.6 KiB	MP3 audio	Yesterday
crack.py	673 bytes	Python script	Yesterday
202211242149401653981955-dacs3-nguyenvandung...	7.2 MiB	Microsoft Word Document	Yesterday
nginx	678 bytes	plain text document	Yesterday
20221124161444Al.py	483 bytes	Python script	Yesterday

Hình 3. 10 Thư mục của server sau khi user gửi file

KẾT LUẬN

1. Kết quả đạt được

1.1 Ưu điểm

- Thực hiện được các chức năng chính đặt ra bao gồm gửi file, mess, lưu dữ liệu lên database.
- Cải thiện được về mặt giao diện.

1.2 Hạn chế

- Giao diện vẫn chưa được tối ưu
- Chỉ thực hiện được các tính năng chat cơ bản

2. Hướng nghiên cứu

Phát triển hơn về mặt giao diện nhằm có thể cải thiện cảm nhận của người dùng cũng như là thêm các chức năng như gọi, call video, ...

TÀI LIỆU THAM KHẢO

[1] stackoverflow.com

[2] Advanced Network Programming - Principles and Techniques-1.pdf

[3] [w3schools.com](https://www.w3schools.com)

[4] viettuts.vn/java-swing

[5] [tutorialspoint.com](https://www.tutorialspoint.com)

PHỤ LỤC