

TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN &
TRUYỀN THÔNG VIỆT HÀN
Khoa Khoa Học Máy Tính



BÁO CÁO ĐỒ ÁN CƠ SỞ (4)
CHAT ROOM SỬ DỤNG MÔ HÌNH P2P

Sinh viên thực hiện: **Nguyễn Thành Công** **21IT469**

Giảng viên hướng dẫn: **ThS. Võ Ngọc Đạt**

Đà Nẵng, tháng 12 năm 2023

TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN &
TRUYỀN THÔNG VIỆT HÀN
Khoa Khoa Học Máy Tính



BÁO CÁO ĐỒ ÁN CƠ SỞ (4)
CHAT ROOM SỬ DỤNG MÔ HÌNH P2P

Sinh viên thực hiện: **Nguyễn Thành Công** **21IT469**

Giảng viên hướng dẫn: **ThS. Võ Ngọc Đạt**

Đà Nẵng, tháng 12 năm 2023

LỜI MỞ ĐẦU

Thời đại ngày nay chứng kiến sự bùng nổ của công nghệ thông tin, nơi mà mô hình ngang hàng (peer-to-peer - P2P) trở thành nguồn động viên mạnh mẽ đằng sau sự đổi mới và tạo ra những trải nghiệm tương tác độc đáo. Trong tình cảnh này, chủ đề "Chat Room sử dụng mô hình P2P" không chỉ là một dấu mốc quan trọng mà còn là hành trình khám phá những tiềm năng và động lực của sự kết nối ngang hàng trong không gian trò chuyện trực tuyến.

Thế giới liên kết của chúng ta đang trải qua một cuộc cách mạng trực tuyến, nơi mà sự giao tiếp không còn giới hạn bởi không gian và thời gian. Mô hình P2P, với khả năng kết nối trực tiếp giữa các đối tác mà không thông qua một trung tâm kiểm soát trung ương, mở ra những triển vọng mới đầy hứa hẹn. Trong ngữ cảnh này, đề tài này đặt ra nhiệm vụ khám phá và phát triển chat room sử dụng mô hình P2P, nhấn mạnh vào tính tương tác và sự linh hoạt mà mô hình này mang lại.

Điều này không chỉ đặt ra thách thức về việc xây dựng một không gian trò chuyện hiệu quả mà còn tập trung vào việc tận dụng sức mạnh của mô hình P2P để tạo ra một môi trường trò chuyện độc đáo và an toàn. Sự kết hợp giữa tính tự trị và sự linh hoạt của mô hình này hứa hẹn sẽ mang lại những trải nghiệm trò chuyện mới mẻ và khác biệt.

Với sự hỗ trợ tận tình của thầy ThS. **Võ Ngọc Đạt** trong khoa "Công nghệ thông tin và truyền thông Việt Hàn", người đã trực tiếp hướng dẫn, chỉ bảo và giúp đỡ. Em rất biết ơn vì đã nhận được sự chỉ dẫn và hỗ trợ tận tình của thầy trong suốt quá trình thực hiện đề tài này.

Một lần nữa em xin được chân thành cảm ơn!

[illegible]

MỤC LỤC

DANH SÁCH THUẬT NGỮ VIẾT TẮT	1
CHƯƠNG 1 TỔNG QUAN VỀ LẬP TRÌNH MẠNG	2
1.1 Tổng quan về mạng máy tính	2
1.1.1 Mạng máy tính là gì?	2
1.1.2 Mô hình mạng	2
1.1.3 Phân loại mạng	5
1.2 Mô hình mạng OSI	6
1.3 Giao thức truyền thông	7
CHƯƠNG 2 XÂY DỰNG ỨNG DỤNG CHAT ROOM SỬ DỤNG P2P VỚI GIAO THỨC TCP SOCKET	9
2.1 Tổng quan về giao thức TCP	9
2.2 Khái niệm về Socket	10
2.3 Sử dụng TCP socket	11
2.4 Mô hình P2P	12
2.4.1 P2P Chia Sẻ Tập (File Sharing):	12
2.4.2 P2P Truyền Thông (Communication):	12
2.4.3 P2P Tài Nguyên Phân Tán:	12
2.4.4 P2P Tài Chính (Peer-to-Peer Finance):	12
2.5 Ngôn ngữ và công cụ hỗ trợ	13
2.5.1 IntelliJ IDEA	13
2.5.2 Kotlin	13
CHƯƠNG 3: PHÂN TÍCH THIẾT KẾ HỆ THỐNG VÀ CÀI ĐẶT KẾT QUẢ THỰC NGHIỆM	15
3.1 Phân tích yêu cầu	15
3.1.1 Yêu cầu của người dùng	15
3.1.2 Yêu cầu chức năng	15
3.1.3 Yêu cầu phi chức năng	16
3.1.4 Yêu cầu hệ thống	17
3.1.5 Biểu đồ Usecase	17
3.1.6 Đặc tả UseCase	18
3.1.7 Biểu đồ Activity	20
3.1.8 Biểu đồ Sequence	20
3.2 Thiết kế	21
3.3 Cài đặt	21
3.4 Kết quả thực nghiệm	22
CHƯƠNG 4: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN	24
4.1 Kết luận	24
4.2 Hướng phát triển	24
TÀI LIỆU THAM KHẢO	25

DANH MỤC HÌNH

Hình 1: Mạng hình sao.....	2
Hình 2: Mạng tuyến tính.....	3
Hình 3: Mạng hình vòng.....	3
Hình 4: Mạng hình cây.....	4
Hình 5: Mạng hình lưới.....	4
Hình 6: Mạng không dây.....	4
Hình 7: Mạng phân tán.....	5
Hình 8: Mô hình mạng OSI.....	6
Hình 9: Quá trình bắt tay 3 bước.....	9
Hình 10: Socket trong mô hình TCP/IP.....	10
Hình 11: Socket trong mô hình ứng dụng.....	11
Hình 12: Mô hình P2P.....	12
Hình 13: Công cụ IntelliJ IDEA.....	13
Hình 14: Ngôn ngữ Kotlin.....	14
Hình 15: Biểu đồ UseCase.....	17
Hình 16: Biểu đồ Activity.....	20
Hình 17: Biểu đồ Sequence.....	20
Hình 18: Thiết kế phác thảo giao diện.....	21
Hình 19: Trang Master Server.....	22
Hình 20: Tạo phòng.....	22
Hình 21: Cửa sổ trò chuyện.....	22
Hình 22: Chuyển đổi ngôn ngữ.....	23

DANH MỤC BẢNG

<i>Bảng 1: Giao thức TCP truyền thống.....</i>	<i>10</i>
<i>Bảng 2: UseCase Gửi tin nhắn.....</i>	<i>18</i>
<i>Bảng 3: UseCase tạo phòng.....</i>	<i>19</i>
<i>Bảng 4: UseCase tham gia Chat Room.....</i>	<i>20</i>

DANH SÁCH THUẬT NGỮ VIẾT TẮT

STT	Từ viết tắt	Ý Nghĩa
1	P2P	Viết tắt của Peer-to-Peer – Mạng ngang hàng
2	OOA	Viết tắt của Object Oriented Analysis – Phân tích theo hướng đối tượng
3	CNTT	Công nghệ thông tin
4	UC	Use Case

CHƯƠNG 1 TỔNG QUAN VỀ LẬP TRÌNH MẠNG

1.1 Tổng quan về mạng máy tính

1.1.1 Mạng máy tính là gì?

Mạng máy tính hay hệ thống mạng (Computer Network hay Network System) là sự kết hợp các máy tính lại với nhau thông qua các thiết bị nối kết mạng và phương tiện truyền thông (giao thức mạng, môi trường truyền dẫn) theo một cấu trúc nào đó và các máy tính này trao đổi thông tin qua lại với nhau.

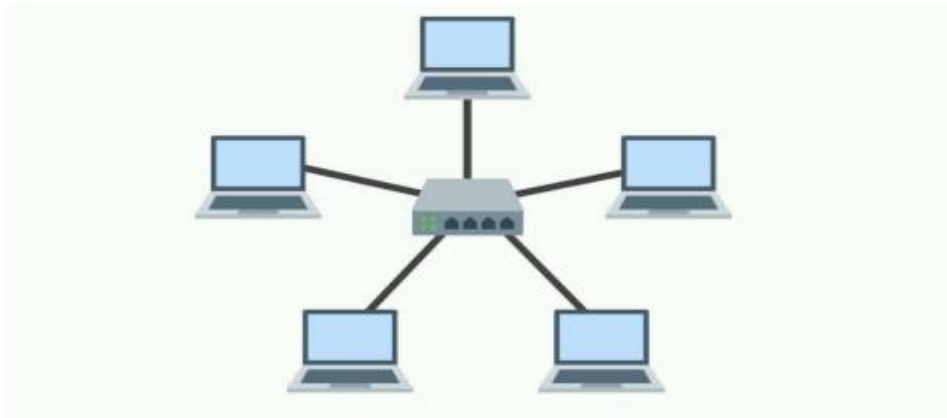
1.1.2 Mô hình mạng

1.1.2.1 Mạng hình sao (Star Network)

Có tất cả các trạm được kết nối với một thiết bị trung tâm có nhiệm vụ nhận tín hiệu từ các trạm và chuyển đến trạm đích. Tùy theo yêu cầu truyền thông trên mạng mà thiết bị trung tâm có thể là hub, switch, router hay máy chủ trung tâm. Vai trò của thiết bị trung tâm là thiết lập các liên kết Point – to – Point.

- Ưu điểm: Thiết lập mạng đơn giản, dễ dàng cấu hình lại mạng (thêm, bớt các trạm), dễ dàng kiểm soát và khắc phục sự cố, tận dụng được tối đa tốc độ truyền của đường truyền vật lý.

- Khuyết điểm: Độ dài đường truyền nối một trạm với thiết bị trung tâm bị hạn chế (bán kính khoảng 100m với công nghệ hiện nay).



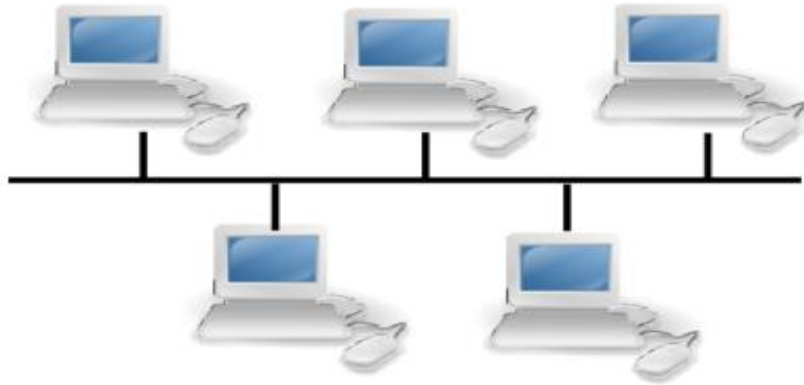
Hình 01: Mạng hình sao

1.1.2.2 Mạng tuyến tính (Bus Network)

Có tất cả các trạm phân chia trên một đường truyền chung (bus). Đường truyền chính được giới hạn hai đầu bằng hai đầu nối đặc biệt gọi là terminator. Mỗi trạm được nối với trục chính qua một đầu nối chữ T (Tconnector) hoặc một thiết bị thu phát (transceiver). Mô hình mạng Bus hoạt động theo các liên kết Point-to-Multipoint hay Broadcast.

- Ưu điểm: Dễ thiết kế và chi phí thấp.

- Khuyết điểm: Tính ổn định kém, chỉ một nút mạng hỏng là toàn bộ mạng bị ngừng hoạt động.

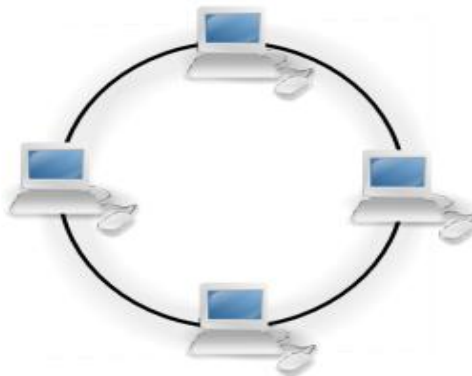


Hình 02: Mạng tuyến tính

1.1.2.3 Mạng hình vòng (Ring Network)

Tín hiệu được truyền đi trên vòng theo một chiều duy nhất. Mỗi trạm của mạng được nối với nhau qua một bộ chuyển tiếp (repeater) có nhiệm vụ nhận tín hiệu rồi chuyển tiếp đến trạm kế tiếp trên vòng. Như vậy tín hiệu được lưu chuyển trên vòng theo một chuỗi liên tiếp các liên kết Point-to-Point giữa các repeater.

- Ưu điểm: Mạng hình vòng có ưu điểm tương tự như mạng hình sao.
- Khuyết điểm: Một trạm hoặc cáp hỏng là toàn bộ mạng bị ngừng hoạt động, thêm hoặc bớt một trạm khó hơn, giao thức truy nhập mạng phức tạp.

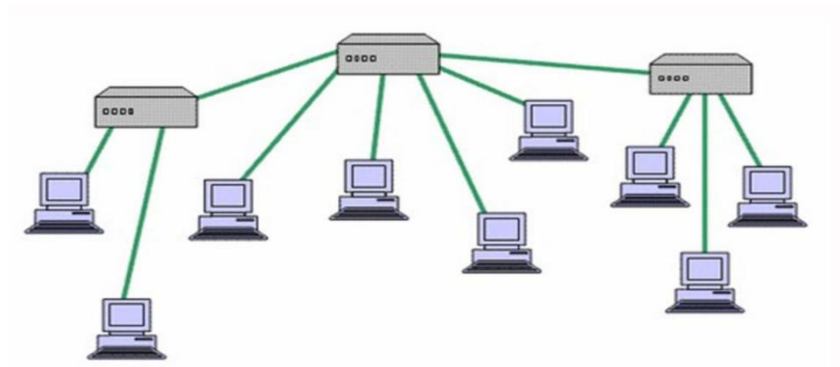


Hình 03: Mạng hình vòng

1.1.2.4 Mạng hình cây (Tree Network)

Mạng hình cây là sự kết hợp của mô hình mạng hình sao và mạng hình bus. Trong mô hình mạng này, các trạm được kết nối với nhau theo dạng hình cây. Mỗi trạm được kết nối với một thiết bị trung tâm, và các thiết bị trung tâm được kết nối với nhau theo dạng tuyến tính.

- Ưu điểm: Tính ổn định cao hơn mạng hình bus, dễ dàng mở rộng mạng
- Khuyết điểm: Chi phí cao hơn mạng hình bus



Hình 04: Mạng hình cây

1.1.2.5 Mạng hình lưới (Mesh Network)

Mạng hình lưới là mô hình mạng trong đó tất cả các trạm được kết nối trực tiếp với nhau. Điều này cho phép các trạm có thể truyền dữ liệu trực tiếp với nhau mà không cần thông qua thiết bị trung tâm.

- Ưu điểm: Tính ổn định cao, dễ dàng mở rộng mạng
- Khuyết điểm: Chi phí cao, khó quản lý

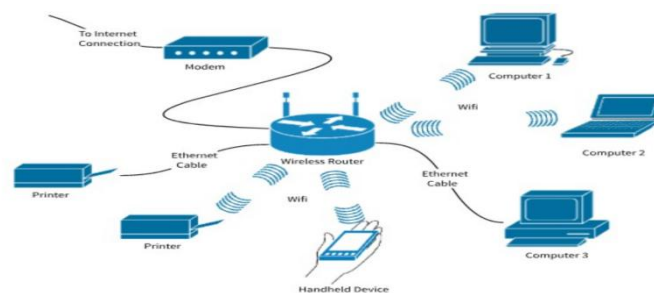


Hình 05: Mạng hình lưới

1.1.2.6 Mạng không dây (Wireless Network)

Mạng không dây là mô hình mạng trong đó các trạm được kết nối với nhau mà không cần sử dụng cáp. Các trạm truyền dữ liệu cho nhau bằng sóng vô tuyến.

- Ưu điểm: Dễ dàng lắp đặt và sử dụng, linh hoạt
- Khuyết điểm: Tốc độ truyền dữ liệu thấp hơn mạng có dây, dễ bị nhiễu

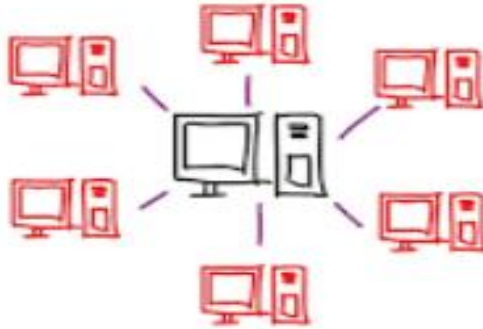


Hình 06: Mạng không dây

1.1.2.7 Mạng phân tán (Distributed Network)

Mạng phân tán là mô hình mạng trong đó các trạm có thể tự hoạt động độc lập với nhau. Các trạm có thể chia sẻ dữ liệu và tài nguyên với nhau thông qua một hệ thống phân tán.

- Ưu điểm: Tính ổn định cao, dễ dàng mở rộng mạng
- Khuyết điểm: Khó quản lý



Hình 07: Mạng phân tán

1.1.3 Phân loại mạng

1.1.3.1 LAN

LAN (local area network), hay còn gọi là "mạng cục bộ", là mạng tư nhân trong một toà nhà, một khu vực (trường học hay cơ quan chẳng hạn) có cỡ chừng vài km. Chúng nối các máy chủ và các máy trạm trong các văn phòng và nhà máy để chia sẻ tài nguyên và trao đổi thông tin.

LAN có 3 đặc điểm:

- Giới hạn về tầm cỡ phạm vi hoạt động từ vài mét cho đến 1 km.
- Thường dùng kỹ thuật đơn giản chỉ có một đường dây cáp (cable) nối tất cả máy. Vận tốc truyền dữ liệu thông thường là 10 Mbps, 100 Mbps, 1 Gbps, và gần đây là 100 Gbps.
- Ba kiến trúc mạng kiểu LAN thông dụng bao gồm:
 - + Mạng bus hay mạng tuyến tính. Các máy nối nhau một cách liên tục thành một hàng từ máy này sang máy kia. Ví dụ của nó là Ethernet (chuẩn IEEE 802.3).
 - + Mạng vòng. Các máy nối nhau như trên và máy cuối lại được nối ngược trở lại với máy đầu tiên tạo thành vòng kín. Thí dụ mạng vòng thẻ bài IBM (IBM token ring).
 - + Mạng sao.

1.1.3.2 MAN

MAN (metropolitan area network), hay còn gọi là "mạng đô thị", là mạng có cỡ lớn hơn LAN, phạm vi vài km. Nó có thể bao gồm nhóm các văn phòng gần nhau trong thành phố, nó có thể là công cộng hay tư nhân và có đặc điểm:

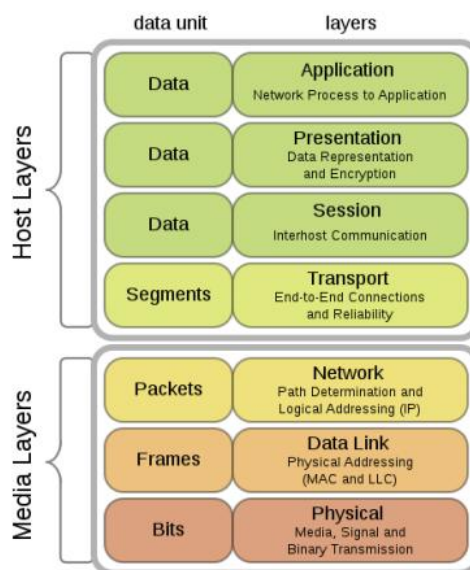
- Chỉ có tối đa hai dây cáp nối.
- Không dùng các kỹ thuật nối chuyên.
- Có thể hỗ trợ chung vận chuyển dữ liệu và đàm thoại, hay ngay cả truyền hình. Ngày nay người ta có thể dùng kỹ thuật cáp quang để truyền tín hiệu. Vận tốc có hiện nay thể đạt đến 10 Gbps.

1.1.3.3 WAN

WAN (wide area network), còn gọi là "mạng diện rộng", dùng trong vùng địa lý lớn thường cho quốc gia hay cả lục địa, phạm vi vài trăm cho đến vài ngàn km. Chúng bao gồm tập hợp các máy nhằm chạy các chương trình cho người dùng. Các máy này thường gọi là máy lưu trữ(host) hay còn có tên là máy chủ, máy đầu cuối. Các máy chính được nối nhau bởi các mạng truyền thông con (communication subnet) hay gọn hơn là mạng con (subnet). Nhiệm vụ của mạng con là chuyển tải các thông điệp (message) từ máy chủ này sang máy chủ khác.

1.2 Mô hình mạng OSI

Mô hình OSI (Open Systems Interconnection Reference Model, viết ngắn là OSI Model hoặc OSI Reference Model) - tạm dịch là Mô hình tham chiếu kết nối các hệ thống mở - là một thiết kế dựa vào nguyên lý tầng cấp, lý giải một cách trừu tượng kỹ thuật kết nối truyền thông giữa các máy vi tính và thiết kế giao thức mạng giữa chúng. Mô hình này được phát triển thành một phần trong kế hoạch Kết nối các hệ thống mở (Open Systems Interconnection) do ISO và IUT-T khởi xướng. Nó còn được gọi là Mô hình bảy tầng của OSI.



Hình 08: Mô hình mạng OSI

Tường trình các tầng cấp của mẫu hình OSI:

- Tầng vật lý (Physical Layer): Tầng vật lý định nghĩa tất cả các đặc tả về điện và vật lý cho các thiết bị. Trong đó bao gồm bố trí của các chân cắm(pin), các hiệu điện thế, và các đặc tả về cáp nối (cable).
- Tầng liên kết dữ liệu (Data-Link Layer): Tầng liên kết dữ liệu cung cấp các phương tiện có tính chức năng và quy trình để truyền dữ liệu giữa các thực thể mạng (truy cập đường truyền, đưa dữ liệu vào mạng), phát hiện và có thể sửa chữa các lỗi trong tầng vật lý nếu có.
- Tầng mạng (Network Layer): Tầng mạng cung cấp các chức năng và quy trình cho việc truyền các chuỗi dữ liệu có độ dài đa dạng, từ một nguồn tới một đích, thông qua một hoặc nhiều mạng, trong khi vẫn duy trì chất lượng dịch vụ (quality of service) mà tầng giao vận yêu cầu.
- Tầng giao vận (Transport Layer): Tầng giao vận cung cấp dịch vụ chuyên dụng chuyển dữ liệu giữa các người dùng tại đầu cuối, nhờ đó các tầng trên không phải quan tâm đến việc cung cấp dịch vụ truyền dữ liệu đáng tin cậy và hiệu quả.
- Tầng phiên (Session layer): Tầng phiên kiểm soát các (phiên) hội thoại giữa các máy tính. Tầng này thiết lập, quản lý và kết thúc các kết nối giữa trình ứng dụng địa phương và trình ứng dụng ở xa.
- Tầng trình diễn (Presentation layer): Lớp trình diễn hoạt động như tầng dữ liệu trên mạng. Lớp này trên máy tính truyền dữ liệu làm nhiệm vụ dịch dữ liệu được gửi từ tầng Application sang dạng Format chung. Và tại máy tính nhận, lớp này lại chuyển từ Format chung sang định dạng của tầng Application.
- Tầng ứng dụng (Application layer): Tầng ứng dụng là tầng gần với người sử dụng nhất. Nó cung cấp phương tiện cho người dùng truy nhập các thông tin và dữ liệu trên mạng thông qua chương trình ứng dụng. Tầng này là giao diện chính để người dùng tương tác với chương trình ứng dụng.

1.3 Giao thức truyền thông

- Giao thức truyền thông (Communication protocol) hay Giao thức liên mạng, Giao thức truyền thông, Giao thức trao đổi thông tin - trong công nghệ thông tin gọi tắt là giao thức (protocol) - là một tập hợp các quy tắc chuẩn dành cho việc biểu diễn dữ liệu, phát tín hiệu, chứng thực và phát hiện các lỗi dữ liệu. Đó là những việc rất cần thiết để gửi thông tin qua các kênh truyền thông, nhờ đó mà các thiết bị có thể kết nối và trao đổi thông tin với nhau.
- Các giao thức dùng cho việc truyền thông tin tín hiệu số trong mạng máy tính có nhiều tính năng để đảm bảo việc trao đổi dữ liệu một cách đáng tin cậy qua một kênh truyền thông không hoàn hảo.
- Có các mô hình giao thức mang tính lý thuyết, được ngành công nghệ thông tin tôn trọng một cách tương đối như mô hình OSI. Cũng có những giao thức được triển khai và ứng dụng rộng rãi như TCP, IP, HTTP, FTP, SMTP, POP3...

- Có nhiều giao thức được sử dụng để giao tiếp hoặc truyền đạt thông tin trên Internet, dưới đây là một số các giao thức tiêu biểu:

+ TCP (Transmission Control Protocol): thiết lập kết nối giữa các máy tính để truyền dữ liệu. Nó chia nhỏ dữ liệu ra thành những gói (packet) và đảm bảo việc truyền dữ liệu thành công.

+ IP (Internet Protocol): định tuyến (route) các gói dữ liệu khi chúng được truyền qua Internet, đảm bảo dữ liệu sẽ đến đúng nơi cần nhận.

+ HTTP (HyperText Transfer Protocol): cho phép trao đổi thông tin (chủ yếu ở dạng siêu văn bản) qua Internet.

+ FTP (File Transfer Protocol): cho phép trao đổi tập tin qua Internet.

+ SMTP (Simple Mail Transfer Protocol): cho phép gửi các thông điệp thư điện tử (e-mail) qua Internet.

+ POP3 (Post Office Protocol, phiên bản 3): cho phép nhận các thông điệp thư điện tử qua Internet.

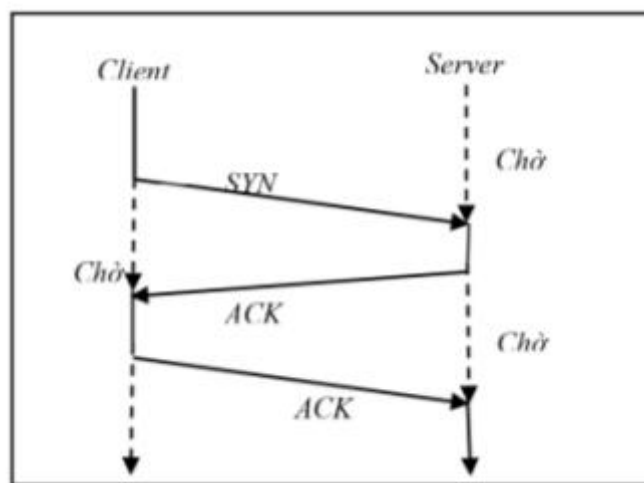
+ WAP (Wireless Application Protocol): cho phép trao đổi thông tin giữa các thiết bị không dây, như điện thoại di động.

CHƯƠNG 2 XÂY DỰNG ỨNG DỤNG CHAT ROOM SỬ DỤNG P2P VỚI GIAO THỨC TCP SOCKET

2.1 Tổng quan về giao thức TCP

Giao thức TCP là giao thức truyền thông có kết nối (connection oriented) và tin cậy (reliable). Truyền thông có kết nối có nghĩa là trước khi truyền dữ liệu, giao thức TCP gửi và giao thức TCP nhận phải thiết lập kênh truyền trước khi gửi/nhận dữ liệu và, trong suốt quá trình trao đổi dữ liệu hai giao thức gửi/nhận phải duy trì kênh truyền đã được thiết lập. Quá trình thiết lập kênh truyền được thực hiện cụ thể bởi một quá trình gọi là quá trình bắt tay ba bước (three-way handshake) như sau:

1. Trước hết giao thức TCP phía máy gửi phải yêu cầu thiết lập một kênh truyền bằng cách gửi một phân đoạn (segment) gọi là SYN đến giao thức TCP phía máy nhận.
2. Giao thức TCP phía máy nhận trả lại một phân đoạn (segment) để xác nhận đã nhận được thành công.
3. Giao thức TCP phía máy gửi tiếp tục gửi một phân đoạn ACK để xác nhận và sau đó là quá trình gửi dữ liệu.



Hình 09: Quá trình bắt tay 3 bước

Truyền thông tin cậy có nghĩa là nút gửi biết được gói tin đã được gửi đến đích hay không. Nếu không đến được, nút gửi phải gửi lại gói tin. Nếu gói tin đã được gửi thành công, nút gửi gửi gói tin tiếp theo. Cụ thể của quá trình này được thực hiện bằng cách mỗi gói tin được đánh một số thứ tự. Mỗi khi nhận được gói tin, nút nhận phải gửi một phản hồi xác nhận (acknowledgment) cho nút gửi để thông báo đã nhận được thành công gói tin.

Giao thức TCP sử dụng truyền thông dạng luồng dữ liệu hay dãy các bytes (byte-stream) liên tục trên kênh truyền, được mô tả như hình vẽ bên dưới:

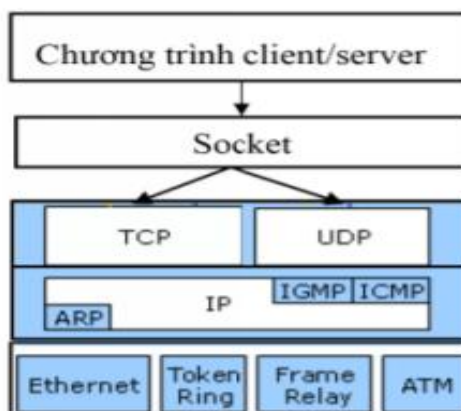
Thông tin điều khiển	Mô tả
Số cổng nguồn(Source Port)	Số hiệu cổng TCP của tiến trình gửi
Số cổng đích(Destination Port)	Số hiệu cổng TCP của tiến trình nhận
Số hiệu gói tin(Sequence Number)	Số thứ tự byte đầu tiên của gói tin TCP
Số hiệu xác nhận(Acknowledgment Number)	Số thứ tự của byte đầu tiên của gói tin TCP mà giao thức TCP gửi chờ nhận từ phía nhận
Kích thước bộ đệm(Window)	Kích thước hiện tại của bộ đệm TCP phía máy gửi dùng để chứa gói tin TCP nhận được.
Thông tin kiểm tra lỗi(TCP Checksum)	Dùng để kiểm tra tính toàn vẹn của phần đầu và dữ liệu của gói tin TCP.

Bảng 01: Giao thức TCP truyền thống

Để kết thúc kết nối, giao thức TCP cũng sử dụng cơ chế bắt tay 3 bước. Điều này đảm bảo cả hai giao thức TCP (gửi và nhận) đã kết thúc truyền dữ liệu và tất cả các dữ liệu đã được gửi và nhận thành công.

2.2 Khái niệm về Socket

Để xây dựng một chương trình theo mô hình client-server, cho phép truyền nhận dữ liệu qua mạng, người lập trình chỉ cần viết chương trình ở tầng ứng dụng và tìm cách giao tiếp với một trong hai giao thức (dịch vụ) ở tầng Vận chuyển là TCP hoặc UDP. Để làm được việc này, các hệ điều hành cung cấp một phương tiện cho phép ứng dụng người dùng giao tiếp được với các giao thức TCP và UDP một cách dễ dàng. Phương tiện đó được gọi là Socket, được biểu diễn như hình vẽ bên dưới:



Hình 10: Socket trong mô hình TCP/IP

Socket được xem như điểm giao tiếp của ứng dụng với mạng, do đó chương trình client/server sẽ thực hiện hai thao tác cơ bản sau:

- + Để truyền dữ liệu, chương trình chỉ cần gửi dữ liệu vào socket
- + Để nhận dữ liệu, chương trình chỉ cần đọc dữ liệu từ socket

Có hai loại socket: socket nối với giao thức TCP còn được gọi là TCP socket, và tương tự như vậy socket nối với giao thức UDP gọi là UDP socket. Một TCP socket được xem như là điểm cuối của kênh kết nối hai chiều liên kết giữa hai chương trình trên mạng.



Hình 11: Socket trong mô hình ứng dụng

Có thể nói một kênh kết nối giữa hai chương trình được thiết lập nhờ vào hai socket được gắn ở hai chương trình. Như hình vẽ trên, mỗi chương trình client và server đều có một socket được gắn vào nó để có thể truyền và nhận dữ liệu.

2.3 Sử dụng TCP socket

Như trình bày ở mục trên, việc xây dựng chương trình client-server sử dụng giao thức TCP của tầng Vận chuyển được thực hiện thông qua TCP socket. Do vậy, người lập trình chỉ cần viết mã lệnh để đọc và ghi dữ liệu giữa chương trình và TCP socket, sau đó TCP socket sẽ có nhiệm vụ liên lạc và gửi, nhận dữ liệu với TCP socket phía đầu còn lại.

Do đó, các hoạt động của TCP socket sẽ bao gồm như sau:

1. TCP Socket phía server:

- Lắng nghe yêu cầu nối kết từ socket phía client
- Chấp nhận kết nối từ socket phía client
- Gửi dữ liệu nhận dữ liệu
- Đóng kết nối

2. TCP Socket phía client:

- Yêu cầu kết nối với socket phía server
- Gửi dữ liệu
- Nhận dữ liệu
- Đóng kết nối

Để TCP socket phía client có thể liên lạc được với TCP socket phía server thì TCP socket phía server phải được gắn với một cổng (port). Sau khi TCP socket

phía server được gán một công, nó sẽ chờ và lắng nghe yêu cầu tạo kênh kết nối từ TCP socket phía client gửi đến.

2.4 Mô hình P2P

Mô hình P2P là viết tắt của "Peer-to-Peer," có nghĩa là truyền thông hay truy cập trực tiếp giữa các thiết bị hoặc người dùng mà không thông qua một trung tâm trung gian. Trong lĩnh vực công nghệ, P2P thường được áp dụng vào nhiều lĩnh vực như chia sẻ tài nguyên, truyền thông và giao thức mạng.

Có nhiều loại mô hình P2P khác nhau, nhưng dưới đây là một số loại phổ biến:

2.4.1 P2P Chia Sẻ Tập (File Sharing):

Decentralized P2P: Các người dùng kết nối trực tiếp với nhau để chia sẻ tệp tin và không có trung tâm điều khiển.

Centralized P2P: Một trung tâm có thể tổ chức quy trình chia sẻ tệp tin nhưng vẫn giữ tính chất P2P trong việc kết nối giữa các người dùng.

2.4.2 P2P Truyền Thông (Communication):

VoIP (Voice over Internet Protocol) P2P: Gọi điện thoại hoặc truyền dữ liệu thoại trực tiếp giữa các người dùng mà không cần một máy chủ trung tâm.

Video Streaming P2P: Truyền video trực tiếp giữa các người dùng, giảm áp lực lên máy chủ trung tâm.

2.4.3 P2P Tài Nguyên Phân Tán:

P2P Computing: Sử dụng nguồn lực máy tính phân tán từ nhiều thiết bị để thực hiện các nhiệm vụ tính toán phức tạp.

P2P Storage: Lưu trữ dữ liệu trên nhiều thiết bị người dùng, tạo ra một hệ thống lưu trữ phân tán.

2.4.4 P2P Tài Chính (Peer-to-Peer Finance):

Cryptocurrency P2P: Giao dịch tiền điện tử trực tiếp giữa các bên mà không cần thông qua ngân hàng hoặc tổ chức tài chính trung ương.

P2P Lending: Cho vay và vay mượn trực tiếp giữa các cá nhân hoặc doanh nghiệp mà không thông qua ngân hàng truyền thống.



Hình 12: Mô hình P2P

2.5 Ngôn ngữ và công cụ hỗ trợ

2.5.1 IntelliJ IDEA

IntelliJ IDEA là môi trường phát triển tích hợp (IDE) rất phổ biến được phát triển bởi JetBrains. Nó được thiết kế chủ yếu cho Java nhưng cũng hỗ trợ nhiều ngôn ngữ lập trình bao gồm Kotlin, Groovy, Scala, v.v. Công cụ phát triển phần mềm này rất có lợi cho các lập trình viên vì với nó, họ có thể tạo, chỉnh sửa, gỡ lỗi và quản lý mã trong các ứng dụng khác nhau một cách hiệu quả.

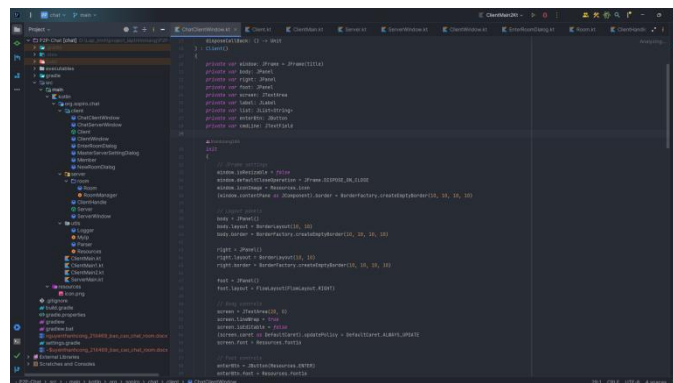
Các tính năng của IntelliJ:

a, Hỗ trợ mã thông minh: IntelliJ IDEA cung cấp khả năng hoàn thành, phân tích và đề xuất mã thông minh, giúp việc mã hóa nhanh hơn và không có lỗi. Nó cũng có thể hiểu ngữ cảnh mã của bạn và có thể cung cấp các đề xuất có liên quan.

b, Tái cấu trúc nâng cao: Các nhà phát triển có thể tự tin cấu trúc lại mã của mình một cách dễ dàng nhờ các công cụ mạnh mẽ để đổi tên, phương pháp trích xuất và tối ưu hóa quá trình nhập.

c, Kiểm tra mã: IDE liên tục kiểm tra mã của bạn để tìm các vấn đề tiềm ẩn, đưa ra các đề xuất và sửa chữa nhanh chóng để duy trì chất lượng mã và giảm thiểu lỗi.

d, Tích hợp kiểm soát phiên bản: Tích hợp liền mạch với các hệ thống kiểm soát phiên bản phổ biến như Git, SVN và Mercurial giúp đơn giản hóa việc cộng tác mã và theo dõi các thay đổi.



Hình 13: Công cụ IntelliJ IDEA

2.5.2 Kotlin

Kotlin là một ngôn ngữ lập trình đa năng, tương thích hoàn toàn với Java và chủ yếu được sử dụng cho phát triển ứng dụng Android. Dưới đây là một số điểm bạn có thể quan tâm khi sử dụng Kotlin trong dự án chatroom:

Tính Tương Thích với Java:

Kotlin hoàn toàn tương thích với mã nguồn Java, điều này có nghĩa là bạn có thể tích hợp mã nguồn Java vào dự án của mình và ngược lại. Điều này rất hữu ích nếu bạn đang làm việc với các thư viện hoặc mã nguồn hiện tại được viết bằng Java.

a, An Toàn và Hiệu Suất: Kotlin được thiết kế để cải thiện tính an toàn của mã nguồn và hiệu suất so với Java. Nó hỗ trợ các tính năng như kiểu dữ liệu không null và giúp giảm thiểu lỗi thời gian biên dịch.

b, Thư Viện và Framework: Kotlin có sẵn nhiều thư viện và framework hữu ích cho việc phát triển ứng dụng server-side và client-side, chẳng hạn như Ktor, một framework web nhẹ cho phát triển các dịch vụ web và ứng dụng.

c, Sử Dụng Coroutines: Kotlin có hỗ trợ cho coroutines, một cơ chế lập lịch nhẹ giúp quản lý luồng và xử lý bất đồng bộ một cách hiệu quả. Điều này có thể rất hữu ích trong các ứng dụng chatroom để xử lý sự kiện và tương tác đa luồng.

d, Community và Tài Nguyên: Kotlin có một cộng đồng lớn và năng động, cung cấp nhiều tài nguyên và hỗ trợ. Bạn có thể tìm kiếm thông tin, hỏi đáp, và chia sẻ kinh nghiệm trên các diễn đàn như Stack Overflow hoặc GitHub.



Hình 14: Ngôn ngữ Kotlin

CHƯƠNG 3: PHÂN TÍCH THIẾT KẾ HỆ THỐNG VÀ CÀI ĐẶT KẾT QUẢ THỰC NGHIỆM

3.1 Phân tích yêu cầu

3.1.1 Yêu cầu của người dùng

a, Quản lý Phòng Trò Chuyện:

Tạo và Tham Gia Phòng: Cho phép người dùng tạo và tham gia các phòng trò chuyện khác nhau.

Quản lý Người Dùng trong Phòng: Khả năng thêm, xóa và quản lý người dùng trong phòng trò chuyện.

b, Chức Năng Trò Chuyện:

Gửi Tin Nhắn: Cung cấp khả năng gửi và nhận tin nhắn giữa các người dùng trong phòng.

c, Hiệu Năng:

Tối Ưu Hóa Hiệu Năng: Đảm bảo hiệu năng hệ thống để tránh trễ và đảm bảo trải nghiệm người dùng mượt mà.

d, Thực Hiện API An Toàn:

API An Toàn: Nếu cung cấp API mở rộng, đảm bảo rằng chúng là an toàn và có thể kiểm soát được.

e, Giao Diện Người Dùng Thân Thiện:

Thiết Kế Giao Diện Người Dùng (UI/UX): Tạo một giao diện người dùng dễ sử dụng và thân thiện.

f, Kết nối P2P:

Quản lý Kết Nối: Tạo và duy trì kết nối P2P giữa các người dùng trong phòng trò chuyện.

Tìm kiếm Người Dùng: Cơ chế để tìm kiếm và kết nối với người dùng khác trong mạng P2P.

3.1.2 Yêu cầu chức năng

a, Kết Nối Trực Tiếp:

Khả năng kết nối trực tiếp giữa các người dùng mà không thông qua máy chủ trung tâm.

b, Tạo và Tham Gia Phòng:

Khả năng tạo phòng chat mới và chia sẻ thông tin cần thiết (ví dụ: địa chỉ IP) để người khác có thể tham gia.

c, Trò Chuyện Văn Bản:

Gửi và nhận tin nhắn văn bản trong phòng chat.

d, Quản Lý Phòng:

Có thể thêm hoặc loại bỏ người dùng khỏi phòng.

Cung cấp tính năng chặn người dùng.

e, Giao Diện Người Dùng Đơn Giản:

Giao diện người dùng thân thiện và dễ sử dụng.

f, Tùy Chỉnh Cài Đặt:

Cho phép người dùng tùy chỉnh ngôn ngữ v.v.

g, Khả Năng Mở Rộng:

Hỗ trợ khả năng mở rộng để có thể xử lý nhiều phòng chat và người dùng đồng thời.

h, Hiệu Suất Cao:

Đảm bảo hiệu suất cao và thời gian đáp ứng ngắn.

3.1.3 Yêu cầu phi chức năng

a, Khả Năng Kết Nối Trực Tiếp:

Mô hình P2P nên hỗ trợ kết nối trực tiếp giữa các người dùng mà không thông qua máy chủ trung tâm.

Cho phép việc thiết lập các kết nối ngang hàng giữa các thiết bị.

b, Quản Lý Kết Nối và Đa Tuyển:

Có khả năng quản lý nhiều kết nối đồng thời giữa các người dùng.

Hỗ trợ chế độ đa tuyển để tối ưu hóa đường truyền.

c, Khả Năng Đồng Bộ Hóa Dữ Liệu:

Đảm bảo đồng bộ hóa tin nhắn và nội dung giữa các thiết bị người dùng.

Hỗ trợ đồng bộ hóa dữ liệu hiệu quả và đồng thời.

d, Chia Sẻ Đa Phương Tiện Trực Tiếp:

Cho phép chia sẻ đa phương tiện (hình ảnh, video) trực tiếp giữa các người dùng mà không thông qua máy chủ trung tâm.

e, Quản Lý Điểm Kết Nối:

Cung cấp cơ chế quản lý và bảo dưỡng các điểm kết nối (nodes) trong mạng P2P.

Tự động phát hiện và thích ứng với sự thay đổi trong mạng.

f, Tìm Kiếm Người Dùng Trực Tiếp:

Có chức năng tìm kiếm trực tiếp người dùng trong mạng P2P mà không cần thông qua máy chủ trung tâm.

g, Bảo Mật và Quyền Riêng Tư:

Hỗ trợ mô hình bảo mật phi tập trung để ngăn chặn sự can thiệp từ bên ngoài.

Đảm bảo quyền riêng tư của người dùng trong môi trường phân tán.

h, Ứng Phó Với Sự Mất Kết Nối:

Xử lý sự mất kết nối một cách linh hoạt và tự động để duy trì tính ổn định của cuộc trò chuyện.

3.1.4 Yêu cầu hệ thống

a, Kết Nối P2P An Toàn:

Bảo Mật Liên Kết: Đảm bảo rằng kết nối giữa các thiết bị là an toàn và được mã hóa.

Xác Thực Người Dùng: Thực hiện cơ chế xác thực mạnh mẽ để ngăn chặn truy cập trái phép.

b, Quản Lý Mạng và Kết Nối:

Hệ Thống Kết Nối Linh Hoạt: Xử lý các tình huống kết nối không ổn định hoặc mất kết nối.

Tìm Kiếm và Kết Nối Người Dùng: Cung cấp cơ chế để tìm kiếm và kết nối với người dùng khác trong mạng P2P.

c, Quản Lý Thông Điệp:

Xác Định và Theo Dõi Thông Điệp: Đảm bảo tính toàn vẹn và theo dõi thông điệp trong môi trường P2P.

Xử Lý Truyền Đa Thông Điệp: Hỗ trợ gửi và nhận đa thông điệp (multicast) một cách hiệu quả.

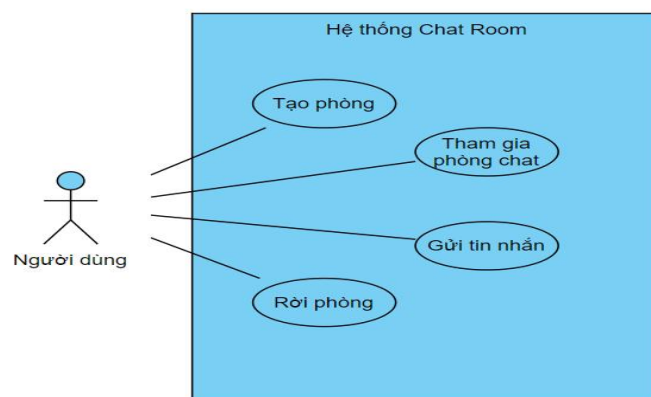
Quản Lý Quyền Riêng Tư: Cho phép người dùng kiểm soát thông tin được chia sẻ và quyền riêng tư của họ.

d, Quản Lý Phiên và Điều Hướng:

Quản Lý Phiên Kết Nối: Theo dõi và duy trì phiên kết nối giữa các thiết bị.

Điều Hướng Tin Nhắn: Xác định cách tin nhắn được định tuyến trong mạng P2P.

3.1.5 Biểu đồ Usecase



Hình 15: Biểu đồ UseCase

3.1.6 Đặc tả UseCase

a, Gửi Tin Nhắn

Tên UseCase	Gửi Tin Nhắn
Mô Tả	Mô tả quá trình một người dùng gửi tin nhắn trong một phòng chat.
Actor	Người Dùng
Điều Kiện Kích Hoạt	Người dùng đã nhập Ip vào hệ thống chat và tham gia vào một phòng chat cụ thể.
Tiền Điều Kiện	Người dùng đã đăng nhập vào hệ thống. Người dùng đã tham gia vào một phòng chat cụ thể.
Hậu Điều Kiện	Tin nhắn đã được gửi thành công và xuất hiện trong phòng chat.
Luồng Sự Kiện Chính	<ol style="list-style-type: none"> 1. Người dùng chọn tùy chọn "Gửi Tin Nhắn" trong giao diện phòng chat. 2. Hệ thống hiển thị ô nhập tin nhắn cho người dùng. 3. Người dùng nhập nội dung tin nhắn. 4. Người dùng xác nhận việc gửi tin nhắn. 5. Hệ thống hiển thị tin nhắn trong giao diện phòng chat cho tất cả các thành viên trong phòng.
Luồng Sự Kiện Rẽ Nhánh	- Nếu người dùng không nhập nội dung tin nhắn (bước 3) : Hệ thống hiển thị thông báo lỗi và yêu cầu người dùng nhập nội dung.

Bảng 02: Usecase gửi tin nhắn

b, Tạo phòng

Tên UseCase	Tạo phòng
Mô Tả	Mô tả quá trình một người dùng tạo một phòng chat mới.
Actor	Người Dùng
Điều Kiện Kích Hoạt	Người dùng đã vào hệ thống chat và muốn tạo một phòng mới.
Tiền Điều Kiện	Người dùng đã đăng nhập vào hệ thống.
Hậu Điều Kiện	Một phòng chat mới đã được tạo và hiển thị trong danh sách các phòng.
Luồng Sự Kiện Chính	<ol style="list-style-type: none"> 1. Người dùng chọn tùy chọn "Tạo Phòng" trong giao diện chat room.

	<p>2. Hệ thống hiển thị giao diện tạo phòng cho người dùng</p> <p>3. Người dùng nhập thông tin phòng (tên, mô tả, cài đặt khác nếu có).</p> <p>4. Người dùng xác nhận việc tạo phòng.</p> <p>5. Hệ thống tạo phòng mới và hiển thị nó trong danh sách phòng chat.</p>
Luồng Sự Kiện Rẽ Nhánh	- Nếu người dùng không nhập đủ thông tin cần thiết (bước 3) : Hệ thống hiển thị thông báo lỗi và yêu cầu người dùng nhập đủ thông tin.

Bảng 03: Usecase tạo phòng

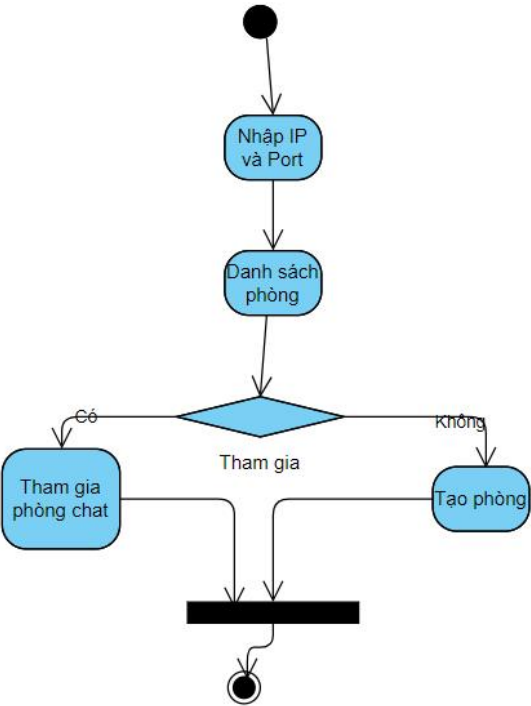
c, Tham gia Chat Room

Tên UseCase	Tham gia Chat Room
Mô Tả	Mô tả quá trình một người dùng tham gia vào một phòng chat.
Actor	Người Dùng
Điều Kiện Kích Hoạt	Người dùng đã vào hệ thống chat và muốn tham gia vào một phòng chat cụ thể.
Tiền Điều Kiện	Người dùng đã đăng nhập vào hệ thống.
Hậu Điều Kiện	Người dùng đã tham gia vào phòng chat và có thể xem và gửi tin nhắn trong phòng.
Luồng Sự Kiện Chính	<p>1. Người dùng chọn phòng chat mà họ muốn tham gia trong danh sách các phòng hiện có.</p> <p>2. Hệ thống kiểm tra xem người dùng có quyền tham gia vào phòng hay không.</p> <p>3. Nếu người dùng có quyền, hệ thống thêm người dùng vào phòng và hiển thị nội dung phòng chat.</p> <p>4. Người dùng có thể bắt đầu gửi và nhận tin nhắn trong phòng.</p>
Luồng Sự Kiện Rẽ Nhánh	- Nếu người dùng không có quyền tham gia vào phòng (bước 2): Hệ

	thống hiển thị thông báo lỗi và không thực hiện việc tham gia vào phòng chat.
--	---

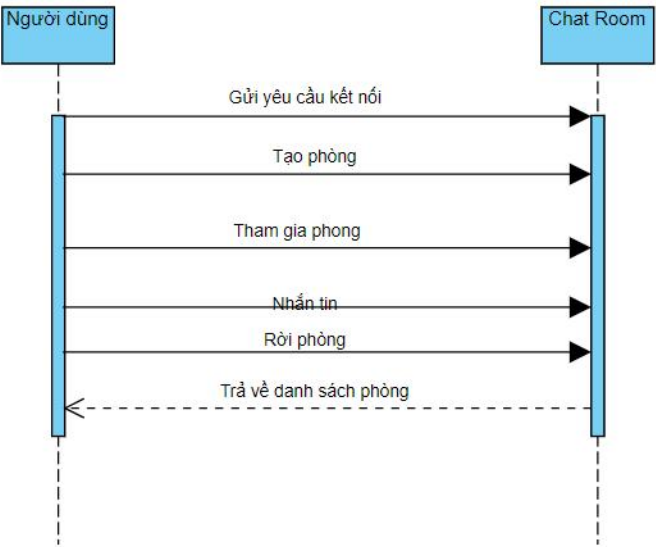
Bảng 04: Usecase tham gia Chat Room

3.1.7 Biểu đồ Activity



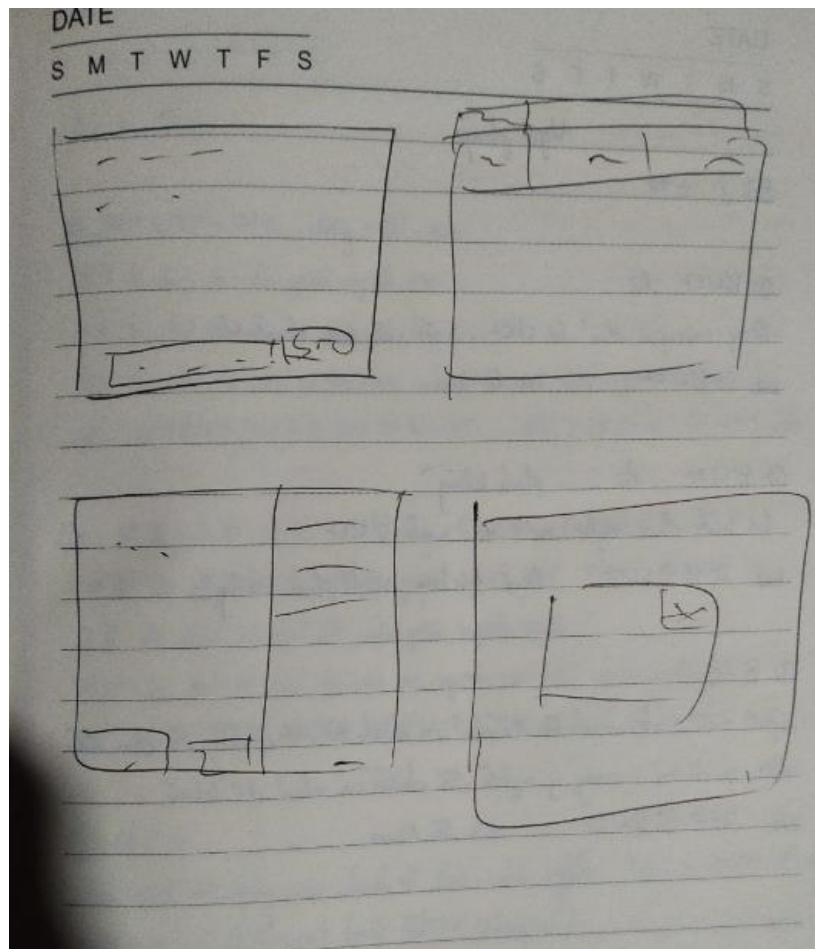
Hình 16: Biểu đồ Activity

3.1.8 Biểu đồ Sequence



Hình 17: Biểu đồ Sequence

3.2 Thiết kế



Hình 18: Thiết kế phác thảo giao diện

3.3 Cài đặt

Để triển khai ứng dụng chat room P2P sử dụng ngôn ngữ Kotlin trên hệ điều hành Windows, em đã sử dụng môi trường phát triển IntelliJ IDEA với plugin Kotlin. Điều này giúp em tạo ra một môi trường phát triển ổn định và linh hoạt cho quá trình phát triển ứng dụng.

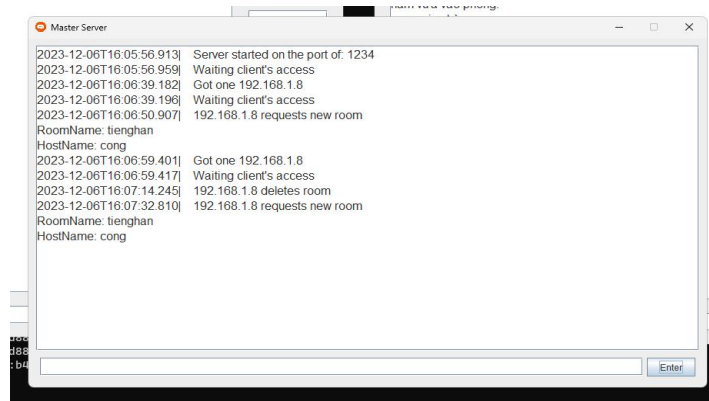
Đầu tiên, em đã tải và cài đặt IntelliJ IDEA từ trang chính thức để bắt đầu quá trình triển khai. Sau đó, em đã chuẩn bị dự án Kotlin bằng cách mở IntelliJ IDEA và tạo một dự án mới với ngôn ngữ lập trình Kotlin. Điều này giúp em có một nền tảng sẵn sàng để phát triển ứng dụng chat room.

Đối với quản lý phụ thuộc và xây dựng dự án, em đã sử dụng Gradle. Bằng cách này, em có thể quản lý các dependencies cần thiết cho ứng dụng một cách dễ dàng và linh hoạt.

Với các bước trên, em đã thành công trong việc cài đặt môi trường phát triển và chuẩn bị dự án cho quá trình phát triển ứng dụng chat room P2P trên IntelliJ IDEA, sử dụng ngôn ngữ Kotlin trên hệ điều hành Windows.

3.4 Kết quả thực nghiệm

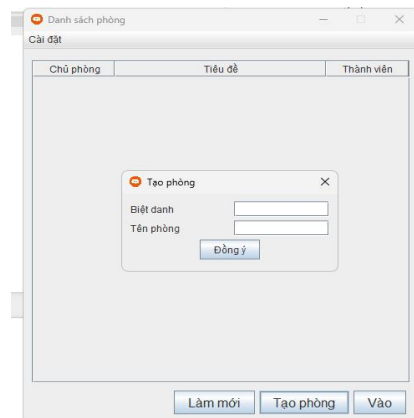
a, Trang Master Server



Hình 19: Trang Master Server

Trang này hiển thị trạng thái của máy chủ. Nó là một trang điều khiển cung cấp thông tin về các kết nối máy khách hiện tại, các phòng đã tạo và các sự kiện mạng gần đây.

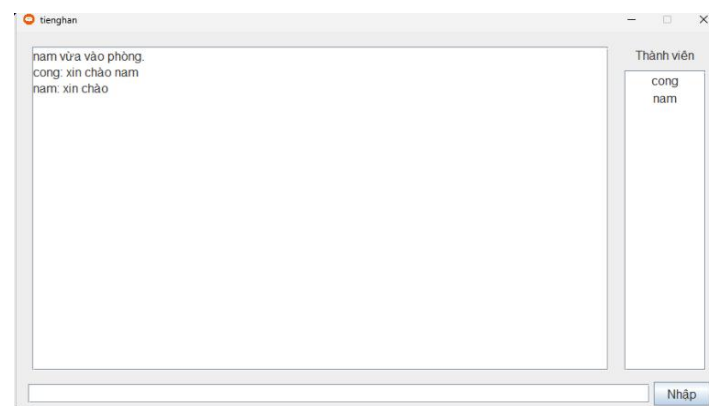
b, Tạo phòng



Hình 20: Tạo phòng

Trang này cho phép khách hàng tạo ra phòng chat room, và những người kết nối vào server đều có thể thấy được phòng đó

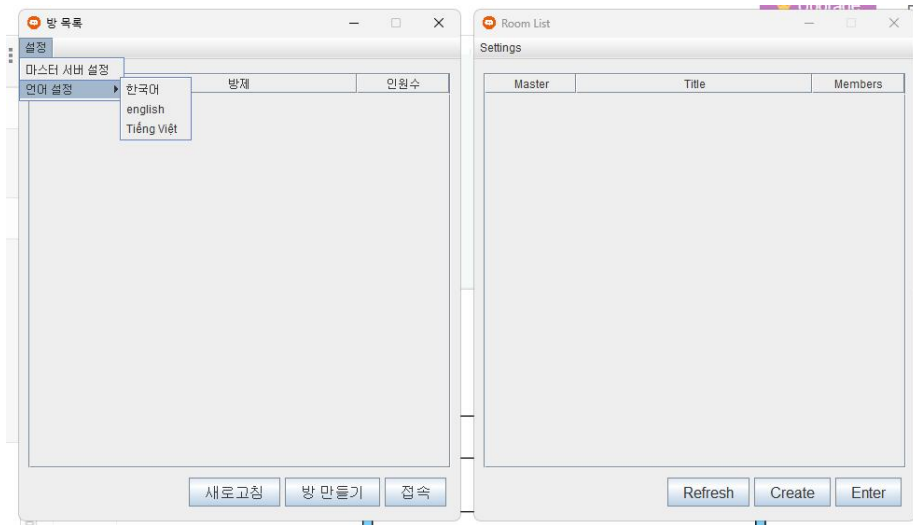
c, Cửa sổ trò chuyện



Hình 21: Cửa sổ trò chuyện

Trang này là một cửa sổ trò chuyện trực tuyến. Nó cho phép hai hoặc nhiều người dùng trò chuyện với nhau trong thời gian thực.

d, Chuyển đổi ngôn ngữ



Hình 22: Chuyển đổi ngôn ngữ

Trong trang này nói về tính năng chuyển đổi ngôn ngữ giúp người dùng tương tác hiệu quả với nhau bằng nhiều ngôn ngữ khác nhau.

CHƯƠNG 4: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

4.1 Kết luận

Trong quá trình thực hiện đề tài về xây dựng ứng dụng chat room sử dụng P2P qua giao thức TCP socket, em đã đạt được những kết quả quan trọng và tích lũy được nhiều kiến thức có giá trị. Dưới đây là tổng hợp những điểm đáng chú ý và những bài học quan trọng:

+ *Triển khai giao thức P2P qua TCP socket*: Em đã thành công trong việc xây dựng và triển khai giao thức P2P sử dụng TCP socket, giúp thiết lập kết nối trực tiếp giữa các người dùng và chuyển đổi dữ liệu một cách hiệu quả.

+ *Hiểu rõ về chat room*: Em đã hiểu rõ về cơ chế hoạt động của các phòng chat và cách quản lý người dùng, tin nhắn, cũng như xử lý sự kiện trong môi trường P2P.

+ *Đảm bảo tính bảo mật và ổn định*: Em đã chú trọng đến vấn đề bảo mật và ổn định trong ứng dụng, đảm bảo rằng thông tin của người dùng được bảo vệ và trải nghiệm sử dụng lành mạnh.

Tuy nhiên, những hạn chế còn tồn tại:

+ *Khả năng mở rộng*: Hiện tại, hệ thống chưa thực sự linh hoạt trong việc mở rộng quy mô và số lượng người dùng. Em cần thêm nghiên cứu và phát triển để đảm bảo khả năng mở rộng của ứng dụng.

+ *Hiệu suất*: Có thể cần tối ưu hóa hiệu suất của ứng dụng, đặc biệt là trong trường hợp số lượng người dùng đồng thời lớn.

4.2 Hướng phát triển

Dựa trên những kết quả đã đạt được và nhận thức từ những hạn chế, em đặt ra những hướng phát triển sau đây:

+ *Mở rộng chức năng chat*: Bổ sung các tính năng như gửi file, hỗ trợ hình ảnh và video, tạo phòng chat riêng, để làm phong phú trải nghiệm người dùng.

+ *Tối ưu hóa mô hình mạng P2P*: Nghiên cứu và triển khai các cải tiến trong mô hình mạng P2P để cải thiện khả năng mở rộng và hiệu suất.

+ *Phát triển ứng dụng di động*: Tích hợp ứng dụng với nền tảng di động, như Android và iOS, để tạo điều kiện thuận lợi cho người dùng sử dụng trên các thiết bị di động.

+ *Bảo mật và quản lý người dùng*: Nâng cao hệ thống bảo mật, áp dụng các biện pháp an toàn thông tin và quản lý người dùng một cách chặt chẽ.

Với những hướng phát triển này, em tin rằng ứng dụng chat room sử dụng P2P TCP socket sẽ ngày càng hoàn thiện và đáp ứng tốt hơn nhu cầu người dùng, đồng thời tạo ra trải nghiệm giao tiếp mới mẻ và thú vị.

TÀI LIỆU THAM KHẢO

- [1] <https://growupcareer.com/2020/04/14/>.
- [2] <https://vdodata.vn/tim-hieu-ve-mo-hinh-client-server-va-client-server-la-gi/>.
- [3] <https://www.semtek.com.vn/mo-hinh-client-server/>.
- [4] <https://www.bkns.vn/so-sanh-mo-hinh-osi-va-tcp-ip.html>.
- [5] <https://www.totolink.vn/article/149-mo-hinh-tcp-ip-la-gi-chuc-nang-cua-cac-tang-trong-mo-hinh-tcp-ip.html>.
- [6] <https://viettuts.vn/lap-trinh-mang-voi-java/giao-thuc-tcp-ip>.
- [7] <https://viblo.asia/p/tim-hieu-giao-thuc-tcp-va-udp-jvEla11xlkw>.
- [8] <https://bizflycloud.vn/tin-tuc/p2p-la-gi-20211112141719259.htm>