TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN VÀ  
TRUYỀN THÔNG VIỆT – HÀN  
KHOA KHOA HỌC MÁY TÍNH



**ĐỒ ÁN MÔN HỌC LẬP TRÌNH MẠNG**

**SỬ DỤNG WEBRTC ĐỂ XÂY DỰNG ỨNG DỤNG HỘI THOẠI TRỰC TUYẾN THEO MÔ HÌNH CLIENT/SERVER**

|  |  |
| --- | --- |
| **Sinh viên thực hiện:** | **Võ Chí Dũng – 21IT069** |
| **Lớp:**  **Giảng viên hướng dẫn:** | **21SE1**  **ThS. Nguyễn Thanh Cẩm** |
|  |  |

***Đà nẵng, ngày 25 tháng 09 năm 2023***

TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN VÀ  
TRUYỀN THÔNG VIỆT – HÀN  
KHOA KHOA HỌC MÁY TÍNH



**ĐỒ ÁN MÔN HỌC LẬP TRÌNH MẠNG**

**SỬ DỤNG WEBRTC ĐỂ XÂY DỰNG ỨNG DỤNG HỘI THOẠI TRỰC TUYẾN THEO MÔ HÌNH CLIENT/SERVER**

|  |  |
| --- | --- |
| **Sinh viên thực hiện:** | **Võ Chí Dũng – 21IT069** |
| **Lớp:**  **Giảng viên hướng dẫn:** | **21SE1**  **ThS. Nguyễn Thanh Cẩm** |
|  |  |

***Đà nẵng, ngày 25 tháng 09 năm 2023***

**LỜI CẢM ƠN**

Chúng em xin gửi lời cảm ơn chân thành tới nhà trường đã tận tình chỉ bảo, góp ý và tạo điều kiện cho em hoàn thành đề tài nghiên cứu Đồ án môn học “Lập trình mạng” một cách tốt nhất.

Em xin cảm ơn ThS. Nguyễn Thanh Cẩm đã nhiệt tình hướng dẫn em trong quá trình thực hiện báo cáo đề tài nghiên cứu.

Trong quá trình thực hiện đề tài nghiên cứu, bản thân chúng em đã cố gắng nỗ lực, tuy nhiên không tránh khỏi sai sót. Chúng em mong nhận được sự góp ý của thầy cô giáo, và từ phía hội đồng để đề tài nghiên cứu của chúng em được hoàn thiện hơn.

Chúng em xin chân thành cảm ơn!

**Sinh viên**

(Kí và ghi rõ họ tên)

NHẬN XÉT

(Giảng viên hướng dẫn)

….……………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………

**Giảng viên hướng dẫn**

(Kí và ghi rõ họ tên)

MỤC LỤC

[MỤC LỤC HÌNH ẢNH vi](#_Toc155849403)

[Chương 1 TỔNG QUAN ĐỀ TÀI 1](#_Toc155849404)

[1.1 Giới thiệu 1](#_Toc155849405)

[1.2 Lý do chọn đề tài 1](#_Toc155849406)

[1.3 Mục tiêu và đối tượng nghiên cứu 1](#_Toc155849407)

[1.4 Các nhiệm vụ cụ thể 1](#_Toc155849408)

[1.5 Phương pháp nghiên cứu 2](#_Toc155849409)

[1.6 Kết quả đặt được 2](#_Toc155849410)

[Chương 2 CƠ SỞ LÝ THUYẾT 3](#_Toc155849411)

[2.1 Sơ lược về lập trình mạng 3](#_Toc155849412)

[2.1.1 Khái niệm về lập trình mạng 3](#_Toc155849413)

[2.1.2 Một số mô hình mạng 3](#_Toc155849414)

[2.1.3 Một số mô hình ứng dụng mạng 9](#_Toc155849415)

[2.2 Công nghệ WebRTC 10](#_Toc155849416)

[2.2.1 Các đặc điểm chính của WebRTC 11](#_Toc155849417)

[2.2.2 Ứng dụng của WebRTC 11](#_Toc155849418)

[2.3 Công cụ lập trình VS Code 12](#_Toc155849419)

[2.3.1 Đặc điểm chính 12](#_Toc155849420)

[2.3.2 Extensions (Tiện ích mở rộng) 12](#_Toc155849421)

[2.3.3 Giao diện dễ sử dụng 13](#_Toc155849422)

[2.4 Ngôn ngữ lập trình Python 13](#_Toc155849423)

[2.4.1 Python là gì? 13](#_Toc155849424)

[2.4.2 Lợi ích Python mang lại: 14](#_Toc155849425)

[2.4.3 Ứng dụng của Python 14](#_Toc155849426)

[2.4.4 Lợi thế của Python 15](#_Toc155849427)

[2.5 Cơ sở dữ liệu MySQL 16](#_Toc155849428)

[Chương 3 PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG 18](#_Toc155849429)

[3.1 Biểu đồ Usecase 18](#_Toc155849430)

[3.2 Đặc tả Usecase 18](#_Toc155849431)

[3.3 Cơ sở dữ liệu 20](#_Toc155849432)

[3.4 Biểu đồ hoạt động 20](#_Toc155849433)

[3.5 Biểu đồ tuần tự 21](#_Toc155849434)

[Chương 4 TRIỂN KHAI DEMO 23](#_Toc155849435)

[4.1 Trang đăng nhập và đăng ký 23](#_Toc155849436)

[4.2 Trang chủ 24](#_Toc155849437)

[4.3 Nhắn tin 24](#_Toc155849438)

[4.4 Thoại trực tuyến 25](#_Toc155849439)

[KẾT LUẬN 27](#_Toc155849440)

[TÀI LIỆU THAM KHẢO 28](#_Toc155849441)

# MỤC LỤC HÌNH ẢNH

[Hình 2‑1 Mô hình OSI 3](#_Toc155849442)

[Hình 2‑2 Mô hình TCP/IP 5](#_Toc155849443)

[Hình 2‑3 Mô hình TCP/IP (1) 6](#_Toc155849444)

[Hình 2‑4 Tầng Ứng dụng 7](#_Toc155849445)

[Hình 2‑5 Tầng Giao vận 8](#_Toc155849446)

[Hình 2‑6 Tầng mạng 8](#_Toc155849447)

[Hình 2‑7 Mô hình Client/server 9](#_Toc155849448)

[Hình 2‑8 Giao tiếp máy chủ và máy khách 9](#_Toc155849449)

[Hình 2‑9 Mô hình Peer-to-peer 10](#_Toc155849450)

[Hình 2‑10 Visual Studio Code 12](#_Toc155849451)

[Hình 2‑11 Ngôn ngữ lập trình Python 13](#_Toc155849452)

[Hình 2‑12 Logo MySQL 16](#_Toc155849453)

[Hình 3‑1 Biểu đồ Usercase 18](#_Toc155849454)

[Hình 3‑2 Cơ sở dữ liệu 20](#_Toc155849455)

[Hình 3‑3 Biểu đồ hoạt động Đăng nhập 20](#_Toc155849456)

[Hình 3‑4 Biểu đồ hoạt động ca Đăng ký 21](#_Toc155849457)

[Hình 3‑5 Biểu đồ tuần tự ca Đăng ký 21](#_Toc155849458)

[Hình 3‑6 Biểu đồ tuần tự ca Đăng ký 22](#_Toc155849459)

[Hình 4‑1 Giao diện Đăng nhập 23](#_Toc155849460)

[Hình 4‑2 Giao diện trang Đăng ký 23](#_Toc155849461)

[Hình 4‑3 Giao diện trang chủ 24](#_Toc155849462)

[Hình 4‑4 Giao diện mục danh sách cuộc trò chuyện 24](#_Toc155849463)

[Hình 4‑5 Giao diện phần trò chuyện 25](#_Toc155849464)

[Hình 4‑6 Gửi file 25](#_Toc155849465)

[Hình 4‑7 Giao diện các nút chức năng trong cuộc trò chuyện 26](#_Toc155849466)

[Hình 4‑8 Giao diện cuộc gọi 26](#_Toc155849467)

# TỔNG QUAN ĐỀ TÀI

## Giới thiệu

Với nhu cầu sử dụng các thiết bị điện tử phổ biến như hiện nay thì việc liên lạc giữa người với người là vô cùng tất yếu, nó đóng một vai trò quan trọng trong cuộc sống. Không những giúp chúng ta trong việc liên lạc, mà nó còn giúp cho chúng ta giải quyết được những nhu cầu về mặt giải trí và những lợi ích xã hội khác. Càng ngày nó đã trở thành một công cụ hỗ trợ đắc lực cho con người trong mọi lĩnh vực.

## Lý do chọn đề tài

Trong thời đại công nghệ ngày nay, sự tiện lợi và nhanh chóng của giao tiếp trực tuyến đang ngày càng trở nên quan trọng. Hội thoại trực tuyến, như là một phương tiện giao tiếp, không chỉ giúp mọi người liên kết mà còn mang lại những trải nghiệm tương tác động đến cảm xúc và sự hiểu biết. Đề tài "SỬ DỤNG WEBRTC ĐỂ XÂY DỰNG ỨNG DỤNG HỘI THOẠI TRỰC TUYẾN THEO MÔ HÌNH CLIENT/SERVER" được chọn để nghiên cứu và phát triển nhằm giải quyết nhu cầu ngày càng tăng về giao tiếp trực tuyến hiệu quả. Mục tiêu và đối tượng nghiên cứu

## Mục tiêu và đối tượng nghiên cứu

Mục tiêu của đề tài là phát triển một ứng dụng hội thoại trực tuyến đáp ứng đầy đủ các yếu tố cần thiết của một hệ thống giao tiếp hiện đại, như tốc độ, đồng bộ, an toàn, và tính năng đa dạng. Nghiên cứu sẽ tập trung vào việc tối ưu hóa trải nghiệm người dùng, cũng như đảm bảo tính bảo mật và ổn định của hệ thống.

Đối tượng nghiên cứu ở đây đó là:

* Ngôn ngữ lập trình Python
* Công cụ lập trình VS Code
* Cơ sở dữ liệu MySQL
* Giao thức Ipv4
* WebRTC

## Các nhiệm vụ cụ thể

* Thiết kế giao diện hội thoại: Xác định và phát triển giao diện người dùng thân thiện, dễ sử dụng và phản ánh đầy đủ chức năng của ứng dụng hội thoại.
* Xây dựng cơ sở hạ tầng hội thoại: Phát triển cơ sở hạ tầng để hỗ trợ truyền tải dữ liệu một cách nhanh chóng và đồng bộ, đồng thời đảm bảo tính bảo mật thông tin.
* Tích hợp tính năng tiện ích: Nghiên cứu và tích hợp các tính năng tiện ích như chia sẻ file, hình ảnh, video, và các tiện ích mở rộng khác để làm giàu trải nghiệm người dùng.

## Phương pháp nghiên cứu

* Nghiên cứu lý thuyết: Tìm hiểu các công nghệ và tiêu chuẩn hiện đại về hội thoại trực tuyến để áp dụng vào phát triển.
* Phân tích và thiết kế: Phân tích yêu cầu người dùng và thiết kế hệ thống dựa trên các yếu tố đã nghiên cứu.
* Phát triển và kiểm thử: Xây dựng ứng dụng, thực hiện kiểm thử để đảm bảo tính ổn định và an toàn

## Kết quả đặt được

Đề tài "SỬ DỤNG WEBRTC ĐỂ XÂY DỰNG ỨNG DỤNG HỘI THOẠI TRỰC TUYẾN THEO MÔ HÌNH CLIENT/SERVER" sẽ mang lại những giải pháp sáng tạo và hiệu quả cho việc giao tiếp trong môi trường trực tuyến. Tôi tin rằng kết quả của đề tài sẽ đáp ứng đầy đủ nhu cầu người dùng và đóng góp tích cực vào sự phát triển của lĩnh vực này...

# CƠ SỞ LÝ THUYẾT

## Sơ lược về lập trình mạng

### Khái niệm về lập trình mạng

Lập trình mạng là quá trình tạo ra các ứng dụng doanh nghiệp hoạt động trên mạng Internet. Trong quá trình này, người lập trình cần phát triển chương trình mạng nhằm thực hiện các chương trình trên nhiều máy tính khác nhau, tạo điều kiện cho các nút mạng chia sẻ dữ liệu một cách an toàn và hiệu quả.

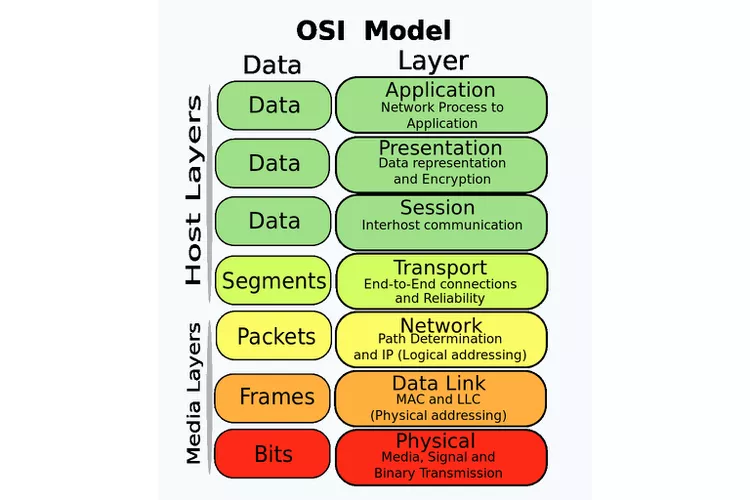
Một cách dễ hiểu hơn, lập trình mạng là việc sử dụng các công cụ lập trình để tạo ra chương trình giúp máy tính tương tác với nhau. Các công cụ này bao gồm ngôn ngữ lập trình và kiến thức về phân tích thiết kế hệ thống, hệ thống mạng, mô hình xây dựng chương trình ứng dụng mạng, cũng như kiến thức về cơ sở dữ liệu và nhiều khía cạnh khác.

### Một số mô hình mạng

1. Mô hình OSI

Còn được biết đến với tên gọi "Mô hình Tham chiếu 7 tầng OSI," là một khung làm việc có mục đích chính là giúp người sử dụng hiểu rõ hơn về cách thông tin được truyền tải giữa các máy tính. Gồm 7 tầng, mỗi tầng trong mô hình này có đặc tính sử dụng chức năng của tầng dưới nó, đồng thời chỉ cho phép tầng trên sử dụng các chức năng của nó.

Thực tế, Mô hình OSI chia nhỏ các hoạt động phức tạp của mạng thành các phần công việc đơn giản hơn, giúp tạo ra một cách nhìn tổng quan và dễ hiểu hơn về cơ chế hoạt động của mạng.

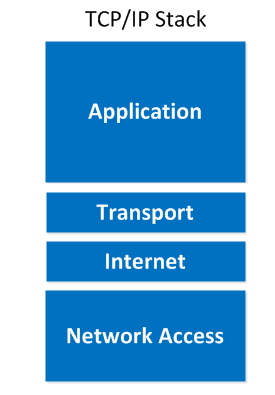


Hình 2‑1 Mô hình OSI

* Tầng 1: Tầng Vật lý (Physical Layer)
* Chức năng chính: Điều khiển việc truyền tải các bit trên đường truyền vật lý.
* Định nghĩa tín hiệu điện, trạng thái đường truyền, phương pháp mã hóa dữ liệu.
* Tầng 2: Tầng Liên kết Dữ liệu (Data-Link Layer)
* Đảm bảo truyền tải các khung dữ liệu (Frame) giữa hai máy tính có đường truyền vật lý nối trực tiếp với nhau.
* Cài đặt cơ chế phát hiện và xử lý lỗi dữ liệu nhận.
* Tầng 3: Tầng Mạng (Network Layer)
* Truyển các gói tin (packet) giữa hai máy tính bất kỳ trong mạng máy tính.
* Tầng 4: Tầng Vận chuyển (Transport Layer)
* Phân nhỏ các gói tin có kích thước lớn khi gửi và tập hợp chúng khi nhận.
* Đảm bảo tính toàn vẹn cho dữ liệu (không bị mất mát, không lặp và đúng thứ tự).
* Tầng 5: Tầng Giao dịch (Session)
* Quản lý phiên làm việc giữa các người sử dụng.
* Cung cấp cơ chế nhận biết tên và chức năng bảo mật thông tin qua mạng máy tính.
* Tầng 6: Tầng Trình bày (Presentation Layer)
* Đảm bảo các máy tính có kiểu định dạng dữ liệu khác nhau vẫn có thể trao đổi thông tin.
* Dịch dữ liệu từ định dạng riêng sang định dạng chung và ngược lại.
* Tầng 7: Tầng Ứng dụng (Application Layer)
* Cung cấp các ứng dụng truy xuất đến các dịch vụ mạng như Web Browser, Mail User Agent, hoặc các Program cung cấp các dịch vụ mạng như Web Server, FTP Server, Mail Server.

1. 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.



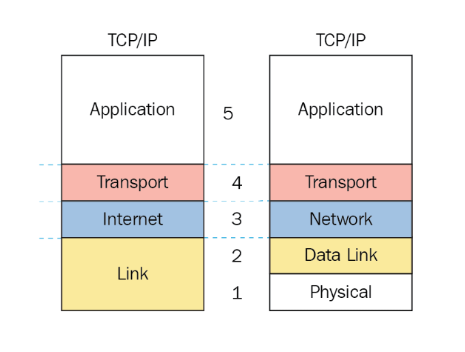
Hình 2‑2 Mô hình TCP/IP

* Cách thức hoạt động của mô hình TCP/IP

Dựa vào tên gọi, TCP/IP là sự kết hợp giữa hai giao thức. Trong đó, IP (Giao thức liên mạng) cho phép các gói tin được gửi đến đích đã định sẵn bằng cách thêm thông tin dẫn đường vào các gói tin, đảm bảo chúng đến đúng đích ban đầu. Giao thức TCP (Giao thức truyền vận) đóng vai trò kiểm tra và bảo vệ sự an toàn cho mỗi gói tin khi đi qua mỗi trạm. Trong quá trình này, nếu TCP phát hiện gói tin bị lỗi, một tín hiệu sẽ được truyền đi và yêu cầu hệ thống gửi lại một gói tin khác. Chức năng chi tiết của mỗi tầng trong mô hình TCP/IP sẽ làm rõ quá trình hoạt động này.

* Chức năng của các tầng trong mô hình TCP/IP

Mô hình TCP/IP tiêu chuẩn bao gồm 4 hoặc 5 lớp được chồng lên nhau, bắt đầu từ tầng thấp nhất là Tầng Vật lý (Physical) → Tầng Mạng (Internet) → Tầng Giao vận (Transport) → Tầng Ứng dụng (Application). Mặc dù nhiều ý kiến cho rằng mô hình này có thể được xem xét với 5 tầng, trong đó Tầng Datalink được tách riêng và đặt ở trên tầng Vật lý.

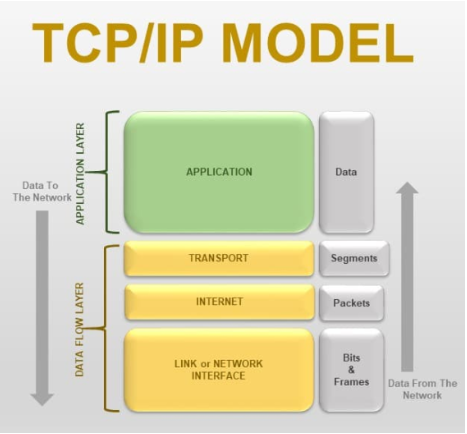


Hình 2‑3 Mô hình TCP/IP (1)

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)

Là tầng giao tiếp trên cùng của mô hình, đảm nhận vai trò giao tiếp dữ liệu giữa các máy thông qua các dịch vụ mạng khác nhau (duyệt web, chat, gửi email, v.v.). Dữ liệu tại đây được định dạng theo kiểu Byte nối Byte và đi kèm với thông tin định tuyến giúp xác định đường đi của một gói tin.

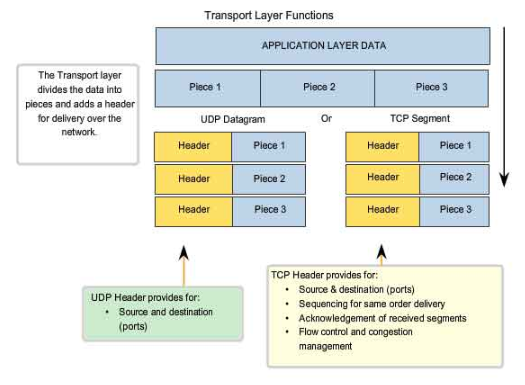


Hình 2‑4 Tầng Ứng dụng

* 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.

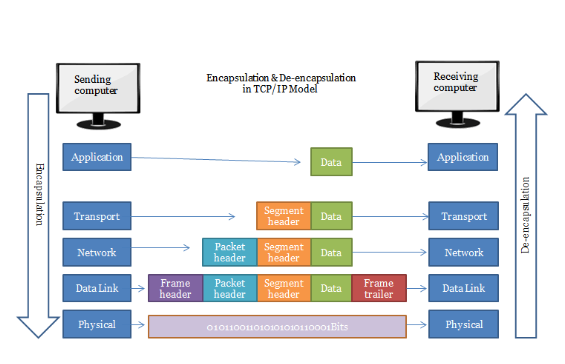
Trong tầng này còn bao gồm 2 giao thức cốt lõi là TCP và UDP. Trong đó, TCP đảm bảo chất lượng gói tin nhưng tiêu tốn thời gian khá lâu để kiểm tra đầy đủ thông tin từ thứ tự dữ liệu cho đến việc kiểm soát vấn đề tắc nghẽn lưu lượng dữ liệu. Trái với điều đó, UDP cho thấy tốc độ truyền tải nhanh hơn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



Hình 2‑5 Tầng Giao vận

* Tầng 2 - Tầng mạng (Internet)

Tương tự như tầng mạng của mô hình OSI, tầng này làm nhiệm vụ chuyển giao dữ liệu một cách logic trong mạng. Dữ liệu được đóng gói thành các gói (Packets) với kích thước phù hợp với mạng chuyển mạch, và thêm phần Header chứa thông tin của tầng mạng.



Hình 2‑6 Tầng mạng

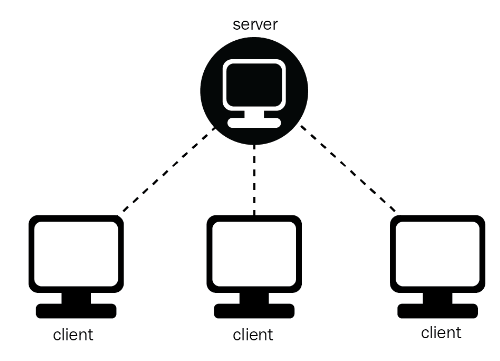
* Tầng 1 - Tầng Vật lý (Physical)

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.

### Một số mô hình ứng dụng mạng

* Mô hình Client/Server

Mô hình Client Server là mô hình mạng máy tính trong đó các máy tính con được đóng vai trò như một máy khách, chúng làm nhiệm vụ gửi yêu cầu đến các máy chủ. Để máy chủ xử lý yêu cầu và trả kết quả về cho máy khách đó.

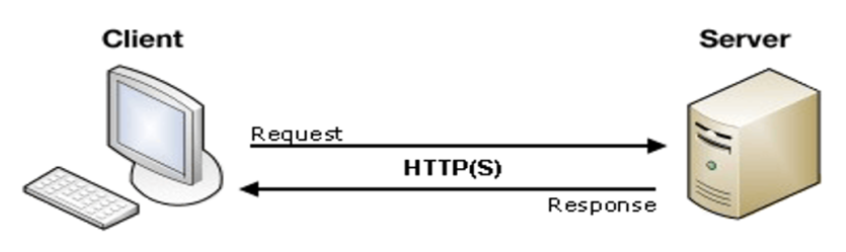


Hình 2‑7 Mô hình Client/server

* Nguyên tắc hoạt động

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.



Hình 2‑8 Giao tiếp máy chủ và máy khách

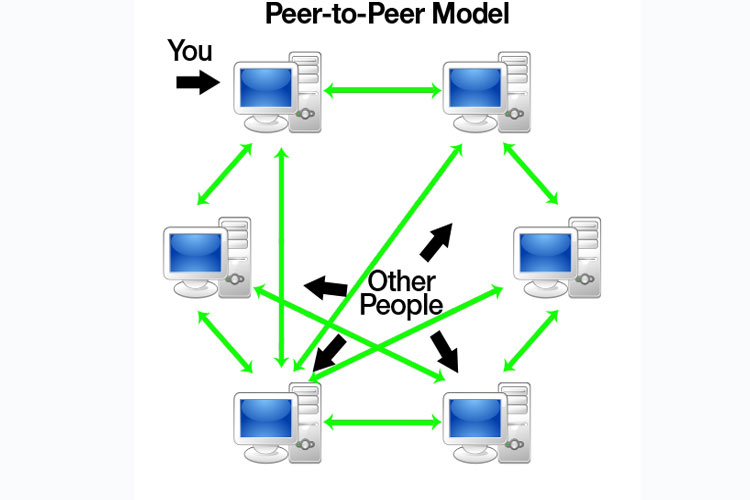
Để thiết lập giao tiếp giữa máy khách và máy chủ, cần tuân theo một chuẩn cụ thể, được biểu diễn thông qua giao thức. Giao thức là một bộ quy tắc được thiết lập để định rõ cách thông tin được trao đổi giữa các thiết bị. Các giao thức phổ biến bao gồm HTTPS, TCP/IP, FTP, và nhiều cái khác.

Khi máy khách muốn lấy thông tin từ máy chủ, nó phải tuân theo giao thức mà máy chủ đang sử dụng. Nếu yêu cầu của máy khách được chấp nhận, máy chủ sẽ xử lý yêu cầu đó, thu thập thông tin tương ứng và trả kết quả về cho máy khách. Do máy chủ luôn sẵn sàng nhận yêu cầu từ máy khách, quá trình này chỉ đòi hỏi máy khách gửi yêu cầu và chấp nhận kết quả trả về từ máy chủ, giữ cho hệ thống luôn trong trạng thái sẵn sàng để tương tác.

* Mô hình Peer to Peer

Mạng ngang hàng hay Peer to Peer (P2P) là một hệ thống máy tính kết nối với nhau thông qua Internet, và chia sẻ dữ liệu mà không cần máy chủ trung tâm. Các mạng máy tính ngang hàng sử dụng cấu trúc phân tán (phi tập trung).

Trong lĩnh vực tiền điện tử, người ta sử dụng P2P để chỉ những giao dịch tiền điện tử hoặc các tài sản kỹ thuật số thông qua mạng phi tập trung.



Hình 2‑9 Mô hình Peer-to-peer

* Nguyên lý hoạt động

Để một máy tính tham gia vào mạng ngang hàng là kết nối Internet và phần mềm P2P. Các chương trình phần mềm P2P phổ biến bao gồm Kazaa, Limewire, BearShare, Morpheus và Acquisition. Các chương trình này kết nối với mạng P2P, như “Gnutella”, và cho phép máy tính truy cập hàng nghìn hệ thống khác trên mạng.

Sau khi kết nối với mạng, phần mềm P2P cho phép bạn tìm kiếm các tệp trên máy tính của người khác. Những người dùng khác trên mạng cũng có thể tìm kiếm tệp trên máy tính của bạn nhưng thường chỉ giới hạn trong một thư mục mà bạn đã chia sẻ.

## Công nghệ WebRTC

WebRTC, hay Web Real-Time Communication, là một công nghệ mạng cho phép truyền thông thời gian thực giữa các trình duyệt web và ứng dụng di động. Được phát triển bởi Internet Engineering Task Force (IETF) và World Wide Web Consortium (W3C), WebRTC mang đến khả năng thiết lập kết nối âm thanh, video và dữ liệu trực tiếp trên web mà không cần các plugin hay ứng dụng bên ngoài.

### Các đặc điểm chính của WebRTC

1. Thiết lập kết nối trực tiếp

WebRTC cho phép truyền thông trực tiếp giữa các thiết bị mà không cần thông qua máy chủ trung gian. Điều này giúp giảm độ trễ và tăng trải nghiệm người dùng trong các ứng dụng thời gian thực như video call hay trò chơi trực tuyến.

1. Âm thanh và video

WebRTC hỗ trợ truyền thông âm thanh và video chất lượng cao, với khả năng tự động điều chỉnh chất lượng tùy thuộc vào điều kiện mạng. Nó sử dụng các codec phổ biến như Opus cho âm thanh và VP8, VP9, hoặc H.264 cho video.

1. Dữ liệu trực tiếp:

Ngoài âm thanh và video, WebRTC cũng cho phép truyền dữ liệu trực tiếp giữa các peer. Điều này mở ra nhiều khả năng ứng dụng, từ truyền tệp đến chia sẻ thông tin trực tiếp giữa các trình duyệt.

1. Khả năng mở rộng và tương thích:

WebRTC được tích hợp sẵn trong nhiều trình duyệt hiện đại như Google Chrome, Mozilla Firefox, và Safari, tạo ra một tiêu chuẩn chung cho việc phát triển ứng dụng truyền thông trên web. Điều này giúp đảm bảo tính tương thích và khả năng mở rộng.

1. Bảo mật và quyền riêng tư:

WebRTC tích hợp các biện pháp bảo mật như mã hóa end-to-end để đảm bảo an toàn trong quá trình truyền thông. Người dùng có thể yên tâm về quyền riêng tư khi sử dụng các ứng dụng được xây dựng trên nền tảng này.

### Ứng dụng của WebRTC

1. Video call and conference

WebRTC đã mở đường cho việc phát triển các ứng dụng video call và conference trực tuyến, từ các ứng dụng nhóm đến các dịch vụ hội nghị trực tuyến.

1. Streaming media

Nền tảng này được sử dụng rộng rãi trong các dịch vụ trực tuyến như streaming video, âm nhạc và phát sóng trực tiếp.

1. Ứng dụng IoT (Internet of Things)

WebRTC có thể tích hợp vào các ứng dụng IoT, cho phép các thiết bị truyền thông với nhau một cách trực tiếp và hiệu quả.

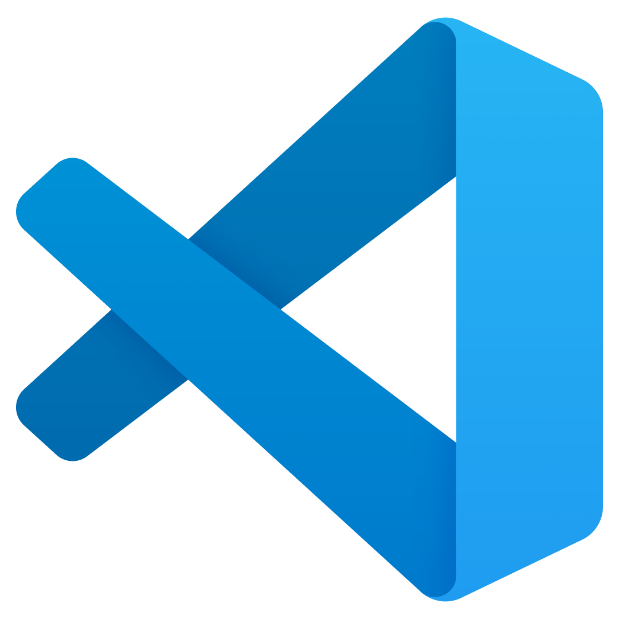
1. E-learning và hội thảo trực tuyến

Trong lĩnh vực giáo dục, WebRTC hỗ trợ việc tổ chức các lớp học trực tuyến, hội thảo và tương tác giữa giáo viên và học viên.

WebRTC không chỉ là một công nghệ, mà còn là một tiêu chuẩn mạng mở cho phép sự tương tác trực tuyến một cách thuận lợi và linh hoạt trên nền tảng web. Được đánh giá cao vì tính đơn giản và hiệu suất cao, WebRTC tiếp tục định hình cách chúng ta truyền thông trực tuyến và mở ra nhiều cơ hội mới cho việc phát triển ứng dụng web.

## Công cụ lập trình VS Code

Visual Studio Code (VS Code) là một trình soạn thảo mã nguồn mã mở phổ biến, được phát triển bởi Microsoft. VS Code có nhiều tính năng mạnh mẽ và linh hoạt, giúp người dùng lập trình hiệu quả và dễ dàng.



Hình 2‑10 Visual Studio Code

### Đặc điểm chính

* VS Code là một trình soạn thảo mã nguồn dựa trên nền tảng Electron, có đặc điểm tiện lợi và mở rộng với nhiều tích hợp. Cụ thể, VS Code có các đặc điểm chính sau:
* Hỗ trợ nhiều ngôn ngữ lập trình: VS Code hỗ trợ đa ngôn ngữ lập trình, bao gồm các ngôn ngữ phổ biến như JavaScript, Python, Java, C/C++,...
* Tích hợp với Git: VS Code tích hợp với Git, giúp người dùng quản lý mã nguồn hiệu quả.
* Tiện ích mở rộng: VS Code hỗ trợ một loạt các tiện ích mở rộng, giúp người dùng mở rộng tính năng của VS Code cho các nhu cầu cụ thể.
* Cộng đồng phát triển mạnh mẽ: VS Code có cộng đồng phát triển mạnh mẽ, cung cấp nhiều tài nguyên và hỗ trợ cho người dùng.

### Extensions (Tiện ích mở rộng)

Tiện ích mở rộng là một trong những tính năng nổi bật nhất của VS Code. Tiện ích mở rộng có thể được sử dụng để thêm các tính năng mới cho VS Code, chẳng hạn như:

* Tự động hoàn thành mã
* Hỗ trợ ngôn ngữ lập trình mới
* Tiện ích bổ sung cho các công cụ phát triển khác
* Tùy chỉnh giao diện của VS Code

### Giao diện dễ sử dụng

Giao diện của VS Code được thiết kế để đơn giản và trực quan, giúp người dùng dễ dàng sử dụng và tùy chỉnh. Giao diện của VS Code bao gồm các thành phần chính sau:

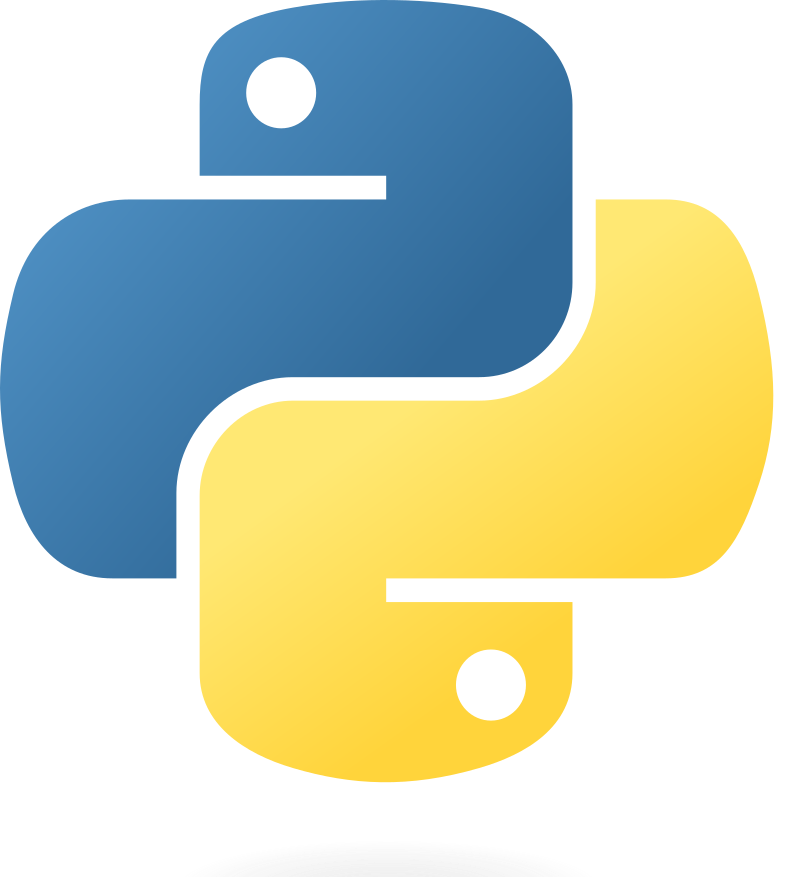
* Thanh menu: Chứa các menu để truy cập các tính năng của VS Code.
* Thanh công cụ: Chứa các nút để thực hiện các thao tác thường dùng.
* Khu vực mã: Khu vực để viết và chỉnh sửa mã.
* Thanh trạng thái: Cung cấp thông tin về trạng thái của VS Code.

Visual Studio Code là một công cụ lập trình mạnh mẽ và linh hoạt, phù hợp với nhiều đối tượng người dùng. VS Code có nhiều tính năng nổi bật, bao gồm hỗ trợ nhiều ngôn ngữ lập trình, tích hợp với Git, tiện ích mở rộng và giao diện dễ sử dụng.

## Ngôn ngữ lập trình Python

### Python là gì?

Python là một ngôn ngữ lập trình được sử dụng rộng rãi trong các ứng dụng web, phát triển phần mềm, khoa học dữ liệu và máy học (ML). Các nhà phát triển sử dụng Python vì nó hiệu quả, dễ học và có thể chạy trên nhiều nền tảng khác nhau. Phần mềm Python được tải xuống miễn phí, tích hợp tốt với tất cả các loại hệ thống và tăng tốc độ phát triển.



Hình 2‑11 Ngôn ngữ lập trình Python

### Lợi ích Python mang lại:

* Các nhà phát triển có thể dễ dàng đọc và hiểu một chương trình Python vì ngôn ngữ này có cú pháp cơ bản giống tiếng Anh.
* Python giúp cải thiện năng suất làm việc của các nhà phát triển vì so với những ngôn ngữ khác, họ có thể sử dụng ít dòng mã hơn để viết một chương trình Python.
* Python có một thư viện tiêu chuẩn lớn, chứa nhiều dòng mã có thể tái sử dụng cho hầu hết mọi tác vụ. Nhờ đó, các nhà phát triển sẽ không cần phải viết mã từ đầu.
* Các nhà phát triển có thể dễ dàng sử dụng Python với các ngôn ngữ lập trình phổ biến khác như Java, C và C++.
* Cộng đồng Python tích cực hoạt động bao gồm hàng triệu nhà phát triển nhiệt tình hỗ trợ trên toàn thế giới. Nếu gặp phải vấn đề, bạn sẽ có thể nhận được sự hỗ trợ nhanh chóng từ cộng đồng.
* Trên Internet có rất nhiều tài nguyên hữu ích nếu bạn muốn học Python. Ví dụ: bạn có thể dễ dàng tìm thấy video, chỉ dẫn, tài liệu và hướng dẫn dành cho nhà phát triển.
* Python có thể được sử dụng trên nhiều hệ điều hành máy tính khác nhau, chẳng hạn như Windows, macOS, Linux và Unix.

### Ứng dụng của Python

Python thường được sử dụng để phát triển trang web và phần mềm, tự động hóa tác vụ, phân tích dữ liệu và trực quan hóa dữ liệu. Vì tương đối dễ học, Python đã được nhiều người không phải là lập trình viên như kế toán và nhà khoa học áp dụng cho nhiều công việc hàng ngày, chẳng hạn như tổ chức tài chính.

* Phân tích dữ liệu và học máy

Python đã trở thành một yếu tố chính trong khoa học dữ liệu, cho phép các nhà phân tích dữ liệu và các chuyên gia khác sử dụng ngôn ngữ này để thực hiện các phép tính thống kê phức tạp, tạo trực quan hóa dữ liệu, xây dựng thuật toán học máy, thao tác và phân tích dữ liệu cũng như hoàn thành các nhiệm vụ khác liên quan đến dữ liệu.

Python có thể xây dựng nhiều dạng trực quan hóa dữ liệu khác nhau, chẳng hạn như biểu đồ đường và thanh, biểu đồ hình tròn, biểu đồ 3D. Python cũng có một số thư viện cho phép các lập trình viên viết chương trình để phân tích dữ liệu và học máy nhanh hơn và hiệu quả hơn, như TensorFlow và Keras.

* Phát triển web

Python thường được sử dụng để phát triển back-end của trang web hoặc ứng dụng - những phần mà người dùng không nhìn thấy. Vai trò của Python trong phát triển web có thể bao gồm gửi dữ liệu đến và đi từ máy chủ, xử lý dữ liệu và giao tiếp với cơ sở dữ liệu, định tuyến URL và đảm bảo tính bảo mật. Python cung cấp một số khuôn khổ để phát triển web. Những cái thường được sử dụng bao gồm Django và Flask.

Một số công việc phát triển web sử dụng Python bao gồm kỹ sư phụ trợ, nhà phát triển Python, kỹ sư phần mềm và kỹ sư DevOps.

* Tự động hoá và phát triển phần mềm

Nếu bạn thấy mình thực hiện một nhiệm vụ lặp đi lặp lại, bạn có thể làm việc hiệu quả hơn bằng cách tự động hóa nó bằng Python. Quá trình viết code được sử dụng để xây dựng các quy trình tự động này được gọi là viết script. Trong thế giới mã hóa, tự động hóa có thể được sử dụng để kiểm tra lỗi trên nhiều tệp, chuyển đổi tệp, thực hiện phép toán đơn giản và loại bỏ các bản sao trong dữ liệu.

Python thậm chí có thể được sử dụng bởi những người mới bắt đầu để tự động hóa các tác vụ đơn giản trên máy tính—chẳng hạn như đổi tên tệp, tìm và tải xuống nội dung trực tuyến hoặc gửi email hoặc văn bản theo khoảng thời gian mong muốn. Trong phát triển phần mềm, Python có thể hỗ trợ các tác vụ như kiểm soát bản dựng, theo dõi lỗi và thử nghiệm. Với Python, các nhà phát triển phần mềm có thể tự động kiểm tra các sản phẩm hoặc tính năng mới. Một số công cụ Python được sử dụng để kiểm thử phần mềm bao gồm Green và Requestium.

### Lợi thế của Python

Python là một ngôn ngữ dễ đọc và đơn giản để hiểu cho các nhà phát triển chưa bao giờ viết code. Do đó, cộng đồng người dùng Python không ngừng phát triển và lớn mạnh. Có rất nhiều học giả và giáo sư trong cộng đồng người dùng Python. Vì vậy, khi xảy ra sự cố, nhà phát triển có thể tập trung vào vấn đề đó và nhận trợ giúp từ những người khác trong cộng đồng mà không phải lo lắng về sự phức tạp của ngôn ngữ.

Python là một ngôn ngữ lập trình miễn phí và mở. Giấy phép nguồn mở được OSI phê chuẩn mà Python được phát triển theo đó làm cho Python trở thành ngôn ngữ tự do sử dụng và phân phối, kể cả cho mục đích thương mại. Nó sẽ làm giảm chi phí của bạn để bảo trì. Trong khi các nhà phát triển có thể chia sẻ, sao chép và thay đổi nó. Đối với cộng đồng Python, đây là cơ hội để chia sẻ kiến thức với các chuyên gia cấp dưới.

Các lập trình viên nói rằng Python rất dễ sử dụng. Mặc dù khi xây dựng các ứng dụng hoặc trò chơi dành cho thiết bị di động, C++ hoặc bất kỳ ngôn ngữ kịch bản điển hình nào khác có thể dễ sử dụng hơn, nhưng Python sẽ tốt hơn để vì nó dễ dàng xây dựng các ứng dụng phía máy chủ, tự động hóa hệ thống xây dựng và thu thập dữ liệu thử nghiệm.

Python có nhiều lựa chọn thư viện và framework, đây là một trong những lợi thế lớn nhất của Python. Từ NumPy đến TensorFlow, thư viện Python được sử dụng cho mọi thứ từ trực quan hóa dữ liệu, học máy, khoa học dữ liệu, xử lý ngôn ngữ tự nhiên và phân tích dữ liệu phức tạp.

Nó sở hữu một thư viện lớn với khả năng quản lý bộ nhớ và thiết kế trống giúp tăng năng suất của nhà phát triển Python. Nhờ đó, các nhà phát triển có thể quản lý cơ sở dữ liệu, tài liệu, trình duyệt web; thực hiện kiểm tra đơn vị và nhiều chức năng khác. Ngoài ra, Python có thể được sử dụng để phát triển nhiều tác vụ, chẳng hạn như phát triển ứng dụng web và máy tính để bàn, phát triển các hệ thống tính toán phức tạp, hệ thống quản lý hỗ trợ cuộc sống, Internet of Things (IoT), trò chơi, v.v.

## Cơ sở dữ liệu MySQL

MySQL là hệ quản trị cơ sở dữ liệu tự do nguồn mở (Relational Database Management System, viết tắt là RDBMS) phổ biến nhất thế giới và được các nhà phát triển rất ưa chuộng trong quá trình phát triển ứng dụng. Vì MySQL là cơ sở dữ liệu tốc độ cao, ổn định và dễ sử dụng, có tính khả chuyển, hoạt động trên nhiều hệ điều hành cung cấp một hệ thống lớn các hàm tiện ích rất mạnh. Với tốc độ và tính bảo mật cao, MySQL rất thích hợp cho các ứng dụng có truy cập CSDL trên internet. MySQL miễn phí hoàn toàn cho nên bạn có thể tải về MySQL từ trang chủ. Nó có nhiều phiên bản cho các hệ điều hành khác nhau: phiên bản Win32 cho các hệ điều hành dòng Windows, Linux, Mac OS X, Unix, FreeBSD, NetBSD, Novell NetWare, SGI Irix, Solaris, SunOS, …Actual result: Mô tả kết quả thực tế khi thực hiện kiểm thử trên môi trường của hệ thống. Actual result thường bao gồm ba giá trị: pass, fail và pending



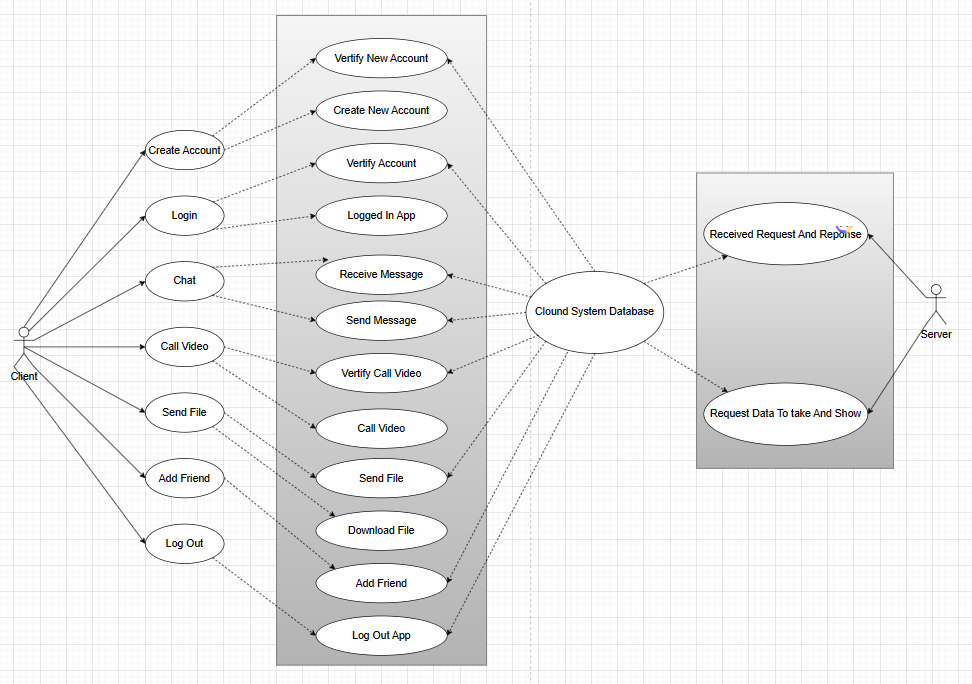
Hình 2‑12 Logo MySQL

MySQL có nhiều đặc điểm cuốn hút:

* Tốc độ: MySQL rất nhanh. Những nhà phát triển cho rằng MySQL là cơ sở dữ liệu nhanh nhất mà bạn có thể có.
* Dễ sử dụng: MySQL tuy có tính năng cao nhưng thực sự là một hệ thống cơ sở dữ liệu rất đơn giản và ít phức tạp khi cài đặt và quản trị hơn các hệ thống lớn.
* Giá thành: MySQL là miễn phí cho hầu hết các việc sử dụng trong một tổ chức.
* Hỗ trợ ngôn ngữ truy vấn: MySQL hiểu SQL, là ngôn ngữ của sự chọn lựa cho tất cả các hệ thống cơ sở dữ liệu hiện đại. Bạn cũng có thể truy cập MySQL bằng cách sử dụng các ứng dụng mà hỗ trợ ODBC (Open Database Connectivity -một giao thức giao tiếp cơ sở dữ liệu được phát triển bởi Microsoft).
* Năng lực: Nhiều client có thể truy cập đến server trong cùng một thời gian. Các client có thể sử dụng nhiều cơ sở dữ liệu một cách đồng thời. Bạn có thể truy cập MySQL tương tác với sử dụng một vài giao diện để bạn có thể đưa vào các truy vấn và xem các kết quả: các dòng yêu cầu của khách hàng, các trình duyệt Web…
* Kết nối và bảo mật: MySQL được nối mạng một cách đầy đủ, các cơ sở dữ liệu có thể được truy cập từ bất kỳ nơi nào trên Internet do đó bạn có thể chia sẽ dữ liệu của bạn với bất kỳ ai, bất kỳ nơi nào. Nhưng MySQL kiểm soát quyền truy cập cho nên người mà không nên nhìn thấy dữ liệu của bạn thì không thể nhìn được.
* Tính linh động: MySQL chạy trên nhiều hệ thống UNIX cũng như không phải UNIX chẳng hạn như Windows hay OS/2. MySQL chạy được các với mọi phần cứng từ các máy PC ở nhà cho đến các máy server.
* Sự phân phối rộng: MySQL rất dễ dàng đạt được, chỉ cần sử dụng trình duyệt web của bạn. Nếu bạn không hiểu làm thế nào mà nó làm việc hay tò mò về thuật toán, bạn có thể lấy mã nguồn và tìm tòi nó. Nếu bạn không thích một vài cái, bạn có thể thay đổi nó. Sự hỗ trợ: Bạn có thể tìm thấy các tài nguyên có sẵn mà MySQL hỗ trợ. Cộng đồng MySQL rất có trách nhiệm. Họ trả lời các câu hỏi trên mailing list thường chỉ trong vài phút. Khi lỗi được phát hiện, các nhà phát triển sẽ đưa ra cách khắc phục trong vài ngày, thậm chí có khi trong vài giờ và cách khắc phục đó sẽ ngay lập tức có sẵn trên Internet

# PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG

## Biểu đồ Usecase



Hình 3‑1 Biểu đồ Usercase

## Đặc tả Usecase

1. Đăng nhập

|  |  |
| --- | --- |
| Tên Use-Case | Đăng nhập |
| Actor chính | Người dùng |
| Tiền điều kiện | Người dùng chưa đăng nhập vào ứng dụng |
| Hậu điều kiện | Người dùng truy cập vào ứng dụng |
| Mô tả chung | Cho phép người dùng truy cập vào ứng dụng |
| Dòng sự kiện | B1: Người dùng đăng nhập bấm vào ứng dụng.  B2: Hệ thống hiển thị giao diện ứng dụng  B3: Người dùng điền thông tin đăng nhập và mật khẩu  B4: Người dùng bấm nút đăng nhập.  B5: Hệ thống kiểm tra thông tin vào hiển thị thông báo người dùng đăng nhập thành công  B6: Hệ thống chuyển người dùng vào hệ thống |
| Dòng sự kiện phụ | Nếu thông tin người dùng nhập không chính xác sẽ hiển thị thông báo |

1. Đăng xuất

|  |  |
| --- | --- |
| Tên Use-Case | Đăng xuất |
| Actor chính | Người dùng |
| Tiền điều kiện | Người dùng đã đăng nhập vào ứng dụng |
| Hậu điều kiện | Người dùng thoát khỏi ứng dụng thành công |
| Mô tả chung | Cho phép người dùng thoát khỏi ứng dụng |
| Dòng sự kiện | B1: Người dùng bấm vào Log out.  B2: Hệ thống hiển thị thông báo xác nhận  B3: Người dùng xác nhận có hoặc không  B4: Người dùng thoát ra giao diện đăng nhập |
| Dòng sự kiện phụ | Nếu người dùng xác nhận không sẽ đóng thông báo |

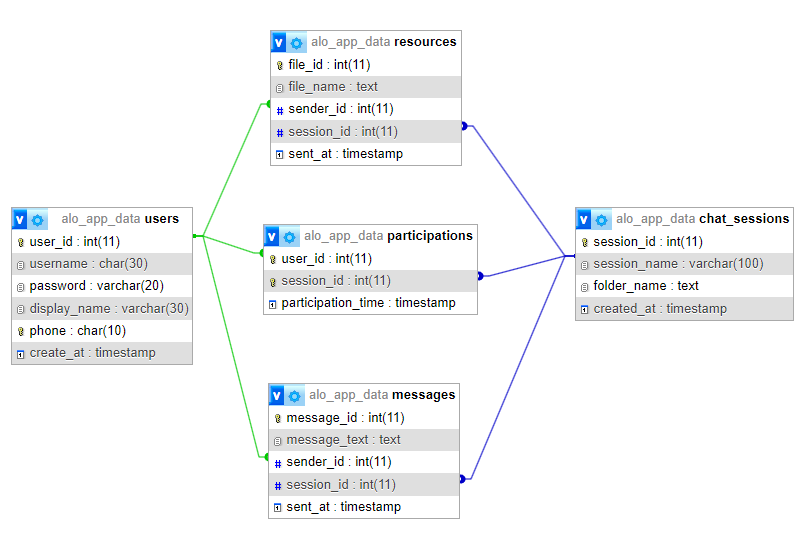
1. Đăng ký

|  |  |
| --- | --- |
| Tên Use-Case | Đăng ký |
| Actor chính | Người dùng |
| Tiền điều kiện | Người dùng chưa có tài khoản để truy cập vào ứng dụng |
| Hậu điều kiện | Người dùng tạo thành công tài khoản |
| Mô tả chung | Cho phép người dùng đăng ký tài khoản truy cập ứng dụng |
| Dòng sự kiện | B1: Người dùng bấm vào đăng ký  B2: Hệ thống chuyển người dùng tới giao diện đăng ký  B3: Người dùng điền thông tin  B4: Hệ thống xác nhận thông tin, nếu hợp lệ thì chuyển người dùng tới giao diện đăng nhập |
| Dòng sự kiện phụ | Nếu người dùng nhập sai thông tin thì xuất hiện thông báo |

1. Trò chuyện

|  |  |
| --- | --- |
| Tên Use-Case | Trò chuyện |
| Actor chính | Người dùng |
| Tiền điều kiện | Người dùng đã đăng nhập vào ứng dụng |
| Hậu điều kiện | Người dùng ở trong cuộc hội thoại |
| Mô tả chung | Cho phép người dùng nhắn tin với bạn bè |
| Dòng sự kiện | B1: Người dùng bấm vào tên của bạn bè  B2: Hệ thống sẽ hiển thị giao diện phòng chat |
| Dòng sự kiện phụ | Không có |

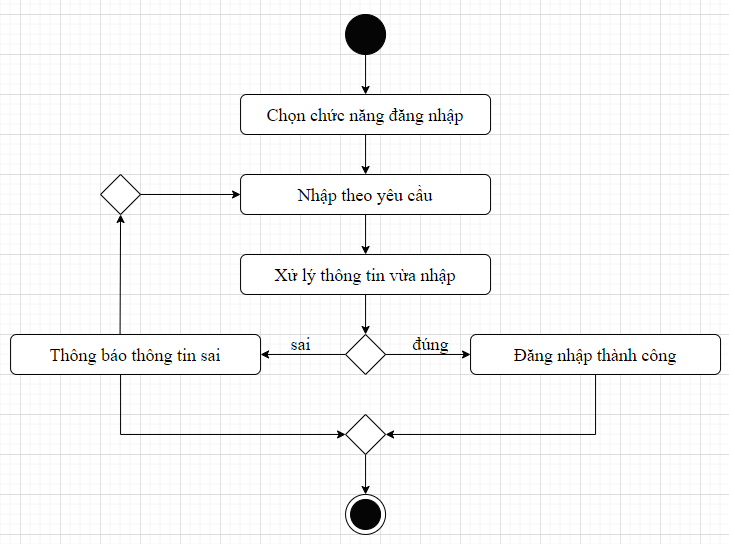
## Cơ sở dữ liệu



Hình 3‑2 Cơ sở dữ liệu

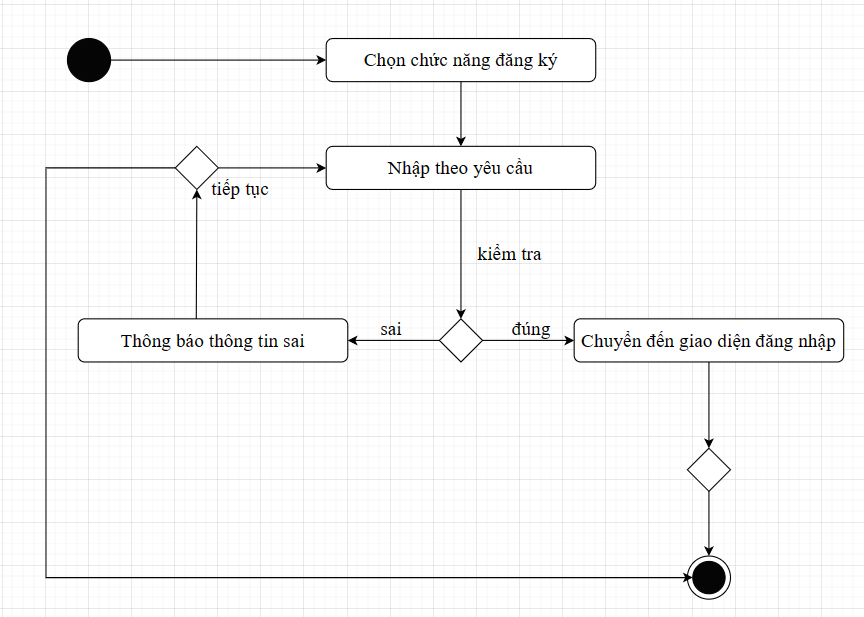
## Biểu đồ hoạt động

1. Đăng nhập



Hình 3‑3 Biểu đồ hoạt động Đăng nhập

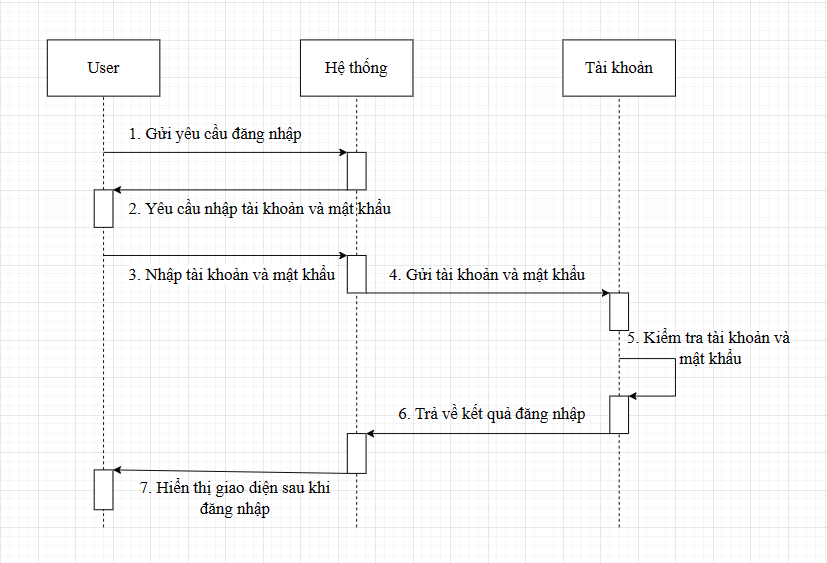
1. Đăng ký



Hình 3‑4 Biểu đồ hoạt động ca Đăng ký

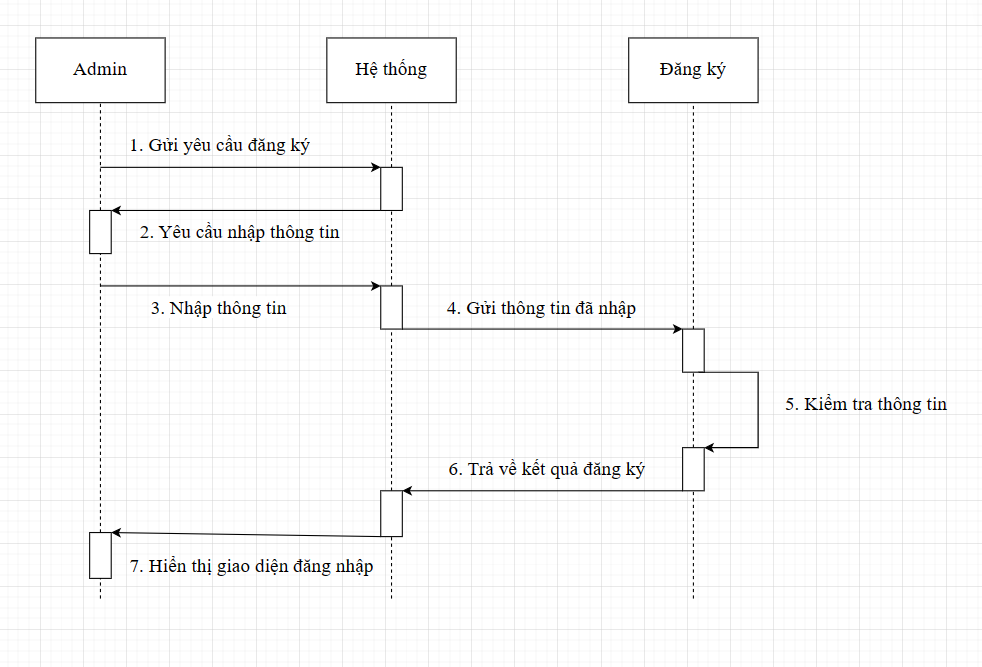
## Biểu đồ tuần tự

1. Đăng nhập



Hình 3‑5 Biểu đồ tuần tự ca Đăng ký

1. Đăng ký

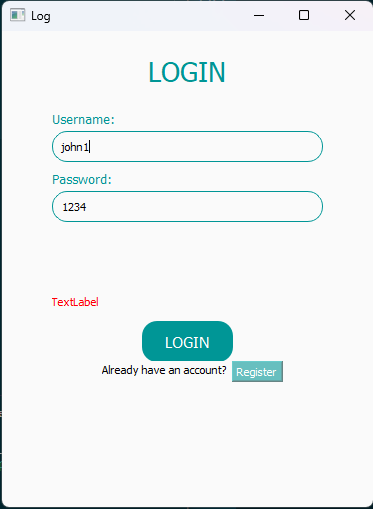


Hình 3‑6 Biểu đồ tuần tự ca Đăng ký

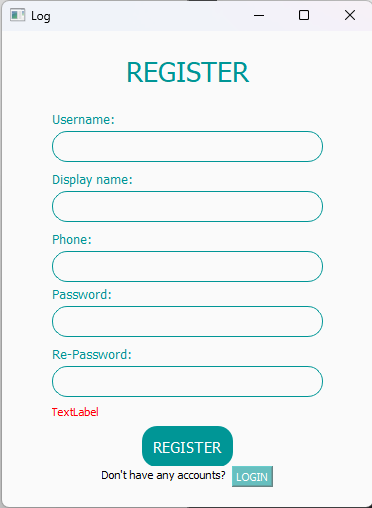
# TRIỂN KHAI DEMO

## Trang đăng nhập và đăng ký

* Nhập thông tin trường Username và Password nhấn nút “LOGIN” để đăng nhập (nếu như đã có tài khoản).
* Nếu chưa có tài khoản thì bấm vào nút “Register” ở trang Loging để chuyển sang giao diện Register. Nhập các thông tin trường nhập và bấm nút “REGISTER” để đăng ký tài khoản.

****

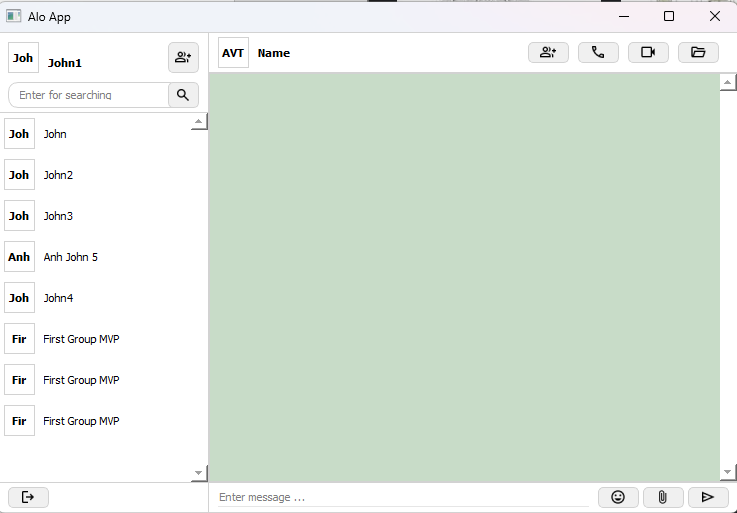
Hình 4‑1 Giao diện Đăng nhập

****

Hình 4‑2 Giao diện trang Đăng ký

## Trang chủ

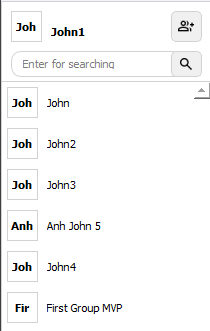
* Sau khi đăng nhập vào hệ thống thì giao diện trang chủ như hình sau:



Hình 4‑3 Giao diện trang chủ

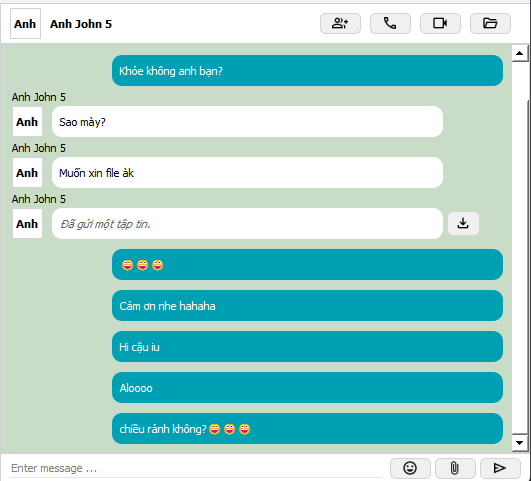
## Nhắn tin

* Bấm vào cuộc trò chuyện trong danh sách như hình sau để bắt đầu cuộc trò chuyện:



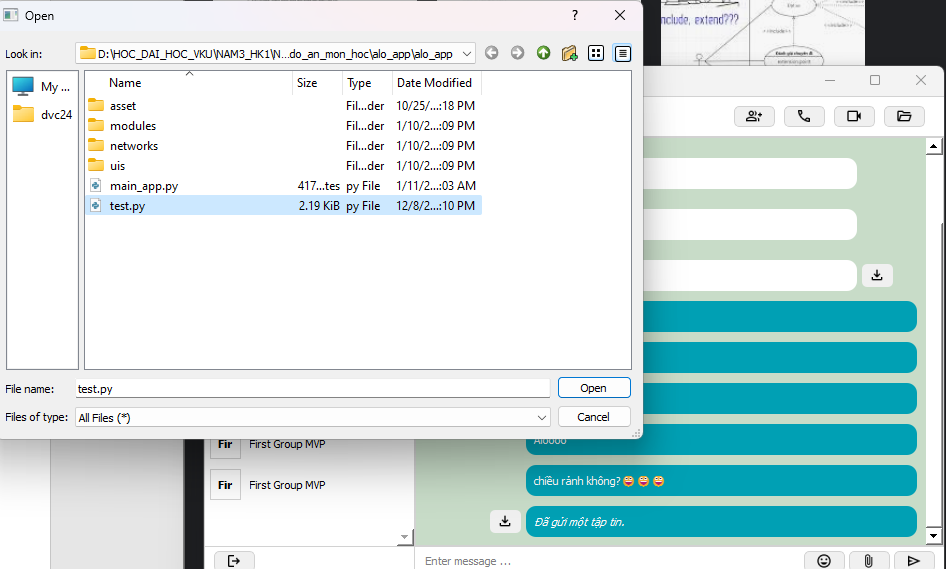
Hình 4‑4 Giao diện mục danh sách cuộc trò chuyện

* Màn hình cuộc trò chuyện hiển thị như hình sau:



Hình 4‑5 Giao diện phần trò chuyện

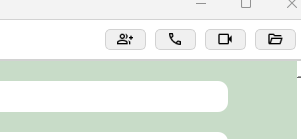
* Nhập thông tin vào mục nhập sau đó bấm nút gửi để gửi tin nhắn (có thể thêm emojis bằng cách bấm nút hình mặt cười để mở kho emojis).
* Để gửi file thì nhấn vào nút “Liên kết” để mở hộp thoại thư mục. Đi đến vị trí file cần gửi và gửi file:



Hình 4‑6 Gửi file

## Thoại trực tuyến

* Trong màn hình trò chuyện, bấm nút gọi như hình dưới đây để tiến hành gọi điện cho đối phương trong cuộc trò chuyện hiện tại:



Hình 4‑7 Giao diện các nút chức năng trong cuộc trò chuyện

* Cuộc trò gọi thoại sẽ được thể hiện như hình sau đây:



Hình 4‑8 Giao diện cuộc gọi

# KẾT LUẬN

1. **Kết luận**

Trong đồ án môn học này chúng em đã vận dụng các kiến thức đã học vào đề tài: “SỬ DỤNG WEBRTC ĐỂ XÂY DỰNG ỨNG DỤNG HỘI THOẠI TRỰC TUYẾN THEO MÔ HÌNH CLIENT/SERVER”, qua đó đạt được một số kết quả sau:

* Tạo ra được một sản phẩm có tính ứng dụng
* Hiều và ứng dụng các kiến thức vào đồ án: Mô hình mạng Client/server, WebRTC, ...
* Nâng cao kĩ năng viết và trình bày báo cáo

1. **Hướng phát triển**

Đồ án của chúng em vẫn còn những hạn chế nên chúng em đưa ra những hướng nghiên cứu phát triển trong tương lai như sau:

* Nghiên cứu sâu hơn về các kiến thức đã và chưa áp dụng có liên quan tới đồ án.
* Phát triển thêm về độ phức tạp cho chương trình.

# TÀI LIỆU THAM KHẢO

[1] Begin to Code with Python (Rob Miles)

[2] Computer Networks: A Systems Approach (Larry Peterson & Bruce Davie)

[3] <https://wikipedia.org>

[4] <https://www.lifewire.com/osi-model-reference-guide-816289>

[5] <https://codelearn.io/sharing/tim-hieu-ve-mo-hinh-client-server>

[6] <https://cryptoleakvn.com/mang-ngang-hang-peer-to-peer-p2p-la-gi-nhung-dieu-can-biet-ve-p2p/>