**TRƯỜNG ĐẠI HỌC**

**CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG VIỆT-HÀN**

**Khoa Khoa Học Máy Tính**



**ĐỒ ÁN CƠ SỞ 4**

**XÂY DỰNG CHƯƠNG TRÌNH AUDIO/VIDEO CONFERENCING SỬ DỤNG CÔNG NGHỆ WEBRTC VÀ MÔ HÌNH CLIENT-SERVER**

Sinh viên thực hiện: **Nguyễn Phỉ Thuận**

Mã sinh viên: **22IT.B201**

Giảng viên hướng dẫn: **Ths. Dương Thị Mai Nga**

Đà Nẵng, tháng 11 năm 2024

**TRƯỜNG ĐẠI HỌC**

**CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG VIỆT-HÀN**

**Khoa Khoa Học Máy Tính**



**ĐỒ ÁN CƠ SỞ 4**

**XÂY DỰNG CHƯƠNG TRÌNH AUDIO/VIDEO CONFERENCING SỬ DỤNG CÔNG NGHỆ WEBRTC VÀ MÔ HÌNH CLIENT-SERVER**

Sinh viên: **Nguyễn Phỉ Thuận**

Mã sinh viên: **22IT.B201**

Giảng viên hướng dẫn: **Ths. Dương Thị Mai Nga**

Đà Nẵng, tháng 11 năm 2024

LỜI CẢM ƠN

Em xin gửi lời cảm ơn chân thành đến giáo viên hướng dẫn Ths.Dương Thị Mai Nga đã tận tình hỗ trợ và dẫn dắt em trong quá trình học tập đồ án cơ sở 4. Bằng sự am hiểu, kiến thức sâu rộng và kinh nghiệm giảng dạy của mình, cô đã giúp em định hướng, đề ra kế hoạch cụ thể và kiểm tra tiến độ công việc. Một lần nữa, em xin gửi lời cảm ơn sâu sắc đến cô Dương Thị Mai Nga.

*Sinh viên*

*Nguyễn Phỉ Thuận*

NHẬN XÉT CỦA GIẢNG VIÊN HƯỚNG DẪN

# 

# 

# 

# 

# 

# 

MỤC LỤC

[MỤC LỤC 5](#_Toc185864711)

[MỞ ĐẦU 7](#_Toc185864712)

[1. Giới thiệu 7](#_Toc185864713)

[2. Mục tiêu của đề tài 7](#_Toc185864714)

[3. Nội dung và kế hoạch thực hiện 7](#_Toc185864715)

[4. Bố cục báo cáo 7](#_Toc185864716)

[CHƯƠNG 1. CƠ SỞ LÝ THUYẾT 9](#_Toc185864717)

[1. Khái niệm về lập trình mạng 9](#_Toc185864718)

[2. Các mô hình mạng 9](#_Toc185864719)

[3. Các giao thức 13](#_Toc185864720)

[4. Sơ lược về một số công nghệ 15](#_Toc185864721)

[CHƯƠNG 2. PHÂN TÍCH THIẾT KẾ HỆ THỐNG 19](#_Toc185864722)

[1. Giới thiệu sơ lược về bài toán 19](#_Toc185864723)

[2. Phân tích thiết kế hệ thống 19](#_Toc185864724)

[CHƯƠNG 3. CHƯƠNG TRÌNH DEMO 32](#_Toc185864725)

[1. Thiết kế giao diện người dùng 32](#_Toc185864726)

[2. Chức năng chính 34](#_Toc185864727)

[KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN 39](#_Toc185864728)

[1. Kết luận 39](#_Toc185864729)

[2. So sánh mục tiêu 40](#_Toc185864730)

[3. Hướng phát triển 40](#_Toc185864731)

[TÀI LIỆU THAM KHẢO 41](#_Toc185864732)

[PHỤ LỤC 42](#_Toc185864733)

**DANH MỤC HÌNH ẢNH**

[Hình 1. Mô hình OSI 10](#_Toc186116737)

[Hình 2. Mô hình TCP/IP 12](#_Toc186116738)

[Hình 3. WebRTC 16](#_Toc186116739)

[Hình 4. Nền tảng ZegoCloud 16](#_Toc186116740)

[Hình 6. Node.js 17](#_Toc186116741)

[Hình 7. WebSocket 17](#_Toc186116742)

[Hình 8. Sơ đồ Use Case 21](#_Toc186116743)

[Hình 9. Sơ đồ Use-Case của quá trình đăng nhập 22](#_Toc186116744)

[Hình 10. Biểu đồ hoạt động của chức năng đăng nhập 23](file:///C:\Users\Admin\OneDrive\Desktop\dacS4.docx#_Toc186116745)

[Hình 11. Biểu đồ tuần tự chức năng đăng nhập 24](#_Toc186116746)

[Hình 12. Biểu đồ hoạt động chức năng đăng xuất 24](#_Toc186116747)

[Hình 13. Biểu đồ tuần tự chức năng đăng xuất 25](#_Toc186116748)

[Hình 14. Sơ đồ Use case của tham gia Room 26](#_Toc186116749)

[Hình 15. Biểu đồ hoạt động chức năng tham gia Room 26](#_Toc186116750)

[Hình 16. Biểu đồ hoạt động chức năng nhắn tin 28](#_Toc186116751)

[Hình 17. Biểu đồ tuần tự chức năng text message 29](file:///C:\Users\Admin\OneDrive\Desktop\dacS4.docx#_Toc186116752)

[Hình 18. Biểu đồ hoạt động chức năng video/call 30](#_Toc186116753)

[Hình 19. Biểu đồ tuần tự sự thiết lập và gọi audio-voice chat 31](#_Toc186116754)

[Hình 20. Đăng nhập 34](#_Toc186116755)

[Hình 21. Đăng ký 34](#_Toc186116756)

[Hình 22. Giao diện sau khi tham gia 36](#_Toc186116757)

[Hình 23. Giao diện tạo cuộc họp 36](#_Toc186116758)

[Hình 24. Chức năng chia sẻ màn hình 37](#_Toc186116759)

[Hình 25. Giao diện box chat 38](#_Toc186116760)

# MỞ ĐẦU

### Giới thiệu

Với sự phát triển nhanh chóng của công nghệ thông tin và truyền thông, các công cụ hội nghị trực tuyến (audio/video conferencing) đã trở thành một phần quan trọng trong cuộc sống và công việc hằng ngày. Các nền tảng như Zoom, Microsoft Teams hay Google Meet không chỉ giúp kết nối con người từ mọi nơi mà còn hỗ trợ làm việc nhóm từ xa và học tập trực tuyến hiệu quả. Vì vậy, việc nghiên cứu và phát triển một ứng dụng hội nghị trực tuyến bằng công nghệ WebRTC và mô hình Client-Server là cần thiết để hiểu rõ cách thức hoạt động và tận dụng những lợi ích của công nghệ truyền thông hiện đại.

### Mục tiêu của đề tài

* Xây dựng một ứng dụng hội nghị truyền hình (audio/video conferencing) thời gian thực sử dụng giao thức WebRTC kết hợp với nền tảng ZegoCloud. Ứng dụng phải đảm bảo tính ổn định, mượt mà và dễ sử dụng, hỗ trợ nhiều người tham gia cùng lúc.
* Sử dụng WebRTC để hỗ trợ truyền tải audio và video giữa các người dùng. Tận dụng nền tảng ZegoCloud theo mô hình Client-Server để quản lý kết nối, giảm thiểu vấn đề về NAT traversal và firewall.
* Cung cấp các tính năng cơ bản như:
  + Kết nối nhiều người dùng trong một phòng họp trực tuyến.
  + Bật/tắt camera và micro của từng người dùng.
  + Quản lý kết nối và xử lý người dùng tham gia/rời khỏi phòng.

### Nội dung và kế hoạch thực hiện

* Nghiên cứu tổng quan: Tìm hiểu giao thức WebRTC và cách thức hoạt động của nó trong việc truyền dữ liệu thời gian thực. Nghiên cứu mô hình Client-Server của ZegoCloud SDK và các API cần thiết để quản lý người dùng và phòng họp. So sánh mô hình Peer-to-Peer với mô hình Client-Server để thấy được ưu nhược điểm của từng loại.
* Phân tích và thiết kế hệ thống: Xác định các chức năng chính của ứng dụng. Thiết kế giao diện.
* Phát triển hệ thống: Tích hợp ZegoCloud để quản lý người dùng và phòng họp. Sử dụng WebRTC để truyền dữ liệu âm thanh và hình ảnh giữa các người dùng. Cài đặt STUN/TURN servers để đảm bảo kết nối ổn định qua NAT/firewall. Xây dựng chức năng bật/tắt camera và micro trong giao diện.

### Bố cục báo cáo

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

Chương 1: Khái niệm về lập trình mạng

Chương 2: Phân tích thiết kế hệ thống

Chương 3: Chương trình Demo

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

# CHƯƠNG 1. CƠ SỞ LÝ THUYẾT

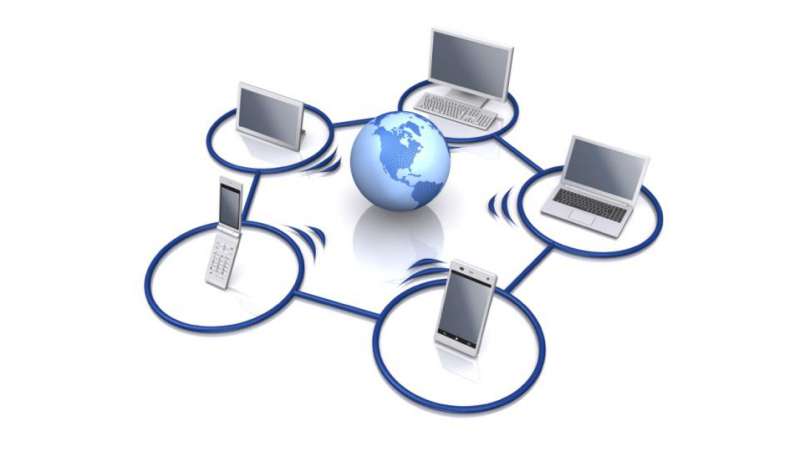
### Khái niệm về lập trình mạng

Lập trình mạng là quá trình xây dựng và phát triển các ứng dụng hoặc hệ thống có khả năng trao đổi dữ liệu thông qua mạng máy tính như LAN (Local Area Network), WAN (Wide Area Network), hoặc Internet. Các ứng dụng mạng này có thể là ứng dụng web, hệ thống truyền thông, trò chuyện, dịch vụ phát trực tuyến, hoặc các trò chơi trực tuyến.

Lập trình mạng là việc thiết lập giao tiếp giữa các thiết bị hoặc giữa các ứng dụng trên các thiết bị khác nhau. Để làm được điều này, lập trình viên cần tuân thủ các giao thức truyền thông để đảm bảo rằng dữ liệu được truyền đi một cách đúng định dạng và có thể hiểu được ở đầu nhận.

Đặc điểm của lập trình mạng:

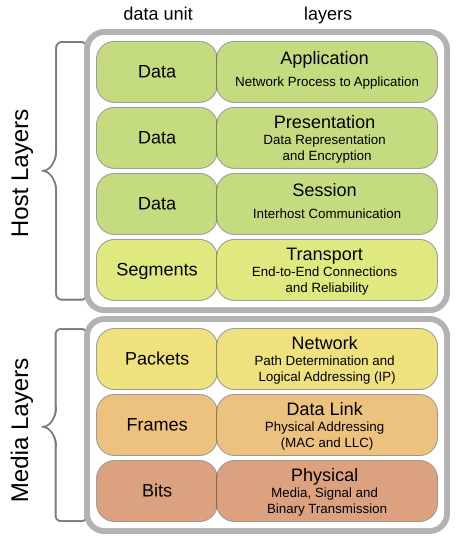
* Giao tiếp qua mạng: Các ứng dụng có khả năng gửi, nhận dữ liệu qua mạng giữa nhiều thiết bị.
* Tính phân tán: Dữ liệu được phân phối trên nhiều thiết bị và hệ thống khác nhau, có thể hoạt động đồng thời.
* Đồng bộ hoặc bất đồng bộ: Ứng dụng có thể xử lý dữ liệu theo thời gian thực (đồng bộ) hoặc theo cách không đồng bộ, tức là dữ liệu không cần được nhận ngay tức thì.
* Xử lý lỗi và bảo mật: Cần đảm bảo khả năng phát hiện và xử lý các lỗi (như lỗi kết nối) và bảo mật thông tin khi truyền qua mạng (sử dụng mã hóa, chứng thực).



### Các mô hình mạng

##### **Mô hình OSI (Open Systems Interconnection) – 7 tầng**

Mô hình OSI là một mô hình tham chiếu chuẩn được tổ chức ISO (International Organization for Standardization) phát triển nhằm tiêu chuẩn hóa việc truyền thông giữa các thiết bị trong mạng. Mô hình này chia quy trình truyền dữ liệu thành 7 tầng độc lập, giúp việc triển khai, xử lý lỗi và bảo trì hệ thống mạng trở nên dễ dàng hơn.



Hình 1. Mô hình OSI

Mỗi tầng trong mô hình OSI thực hiện một nhiệm vụ riêng và giao tiếp với tầng ngay trên và ngay dưới nó. Các tầng từ dưới lên bao gồm:

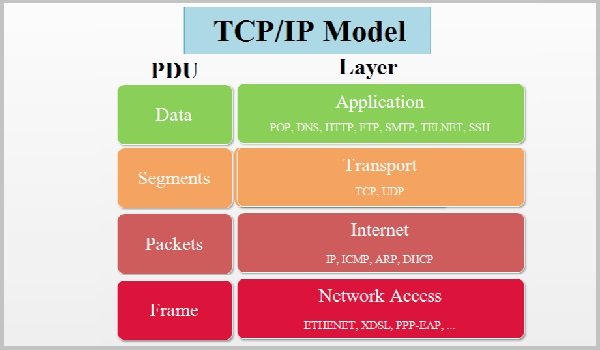
* Physical (Vật lý).
* Data Link (Liên kết dữ liệu).
* Network (Mạng).
* Transport (Giao vận).
* Session (Phiên).
* Presentation (Trình bày).
* Application (Ứng dụng).
  + - Tầng vật lý (Physical Layer):
      * Nhiệm vụ: Xử lý các tín hiệu điện tử, sóng vô tuyến, ánh sáng hoặc các phương tiện truyền vật lý khác để truyền dữ liệu thô giữa các thiết bị.
      * Chức năng chính: Truyền và nhận tín hiệu dạng bit 0 và 1. Quy định các tiêu chuẩn vật lý cho tốc độ truyền, điện áp, và đầu nối. Xử lý các vấn đề về đồng bộ tín hiệu.
    - Tầng Liên kết dữ liệu (Data Link Layer):
      * Nhiệm vụ: Tầng này chịu trách nhiệm truyền dữ liệu giữa hai thiết bị kết nối trực tiếp với nhau qua kênh vật lý, đồng thời phát hiện và sửa các lỗi phát sinh.
      * Chức năng chính: Điều khiển truy cập vào phương tiện truyền thông (MAC – Media Access Control). Phát hiện và sửa lỗi khung (frame) trong quá trình truyền. Tạo và xử lý các địa chỉ MAC cho thiết bị.
    - Tầng Mạng (Network Layer):
      * Nhiệm vụ: Tầng mạng chịu trách nhiệm định tuyến dữ liệu từ thiết bị nguồn đến thiết bị đích thông qua các mạng khác nhau. Nó thực hiện các tác vụ liên quan đến địa chỉ IP và chọn đường truyền tốt nhất.
      * Chức năng chính: Gán địa chỉ IP cho các thiết bị. Định tuyến gói dữ liệu qua nhiều mạng (routing). Phân mảnh và tái lắp ráp gói dữ liệu.
    - Tầng Giao vận (Transport Layer):
      * Nhiệm vụ: Đảm bảo dữ liệu được truyền đáng tin cậy từ đầu cuối này đến đầu cuối kia. Tầng này chia dữ liệu thành các gói nhỏ và kiểm tra xem các gói có đến đúng thứ tự không.
      * Chức năng chính: TCP: Đảm bảo truyền dữ liệu đầy đủ, có xác nhận và sắp xếp thứ tự. UDP: Truyền nhanh hơn nhưng không đảm bảo độ tin cậy. Kiểm soát luồng và xử lý lỗi.
    - Tầng Phiên (Session Layer):
      * Nhiệm vụ: Tầng này quản lý các phiên giao tiếp giữa hai thiết bị hoặc hai ứng dụng. Nó thiết lập, duy trì và chấm dứt các phiên kết nối khi cần thiết.
      * Chức năng chính: Thiết lập và duy trì phiên kết nối. Quản lý đồng bộ và tái kết nối trong trường hợp lỗi. Điều khiển hội thoại (đảm bảo dữ liệu được gửi đúng thứ tự giữa hai phía).
    - Tầng Trình bày (Presentation Layer):
      * Nhiệm vụ: Tầng này chịu trách nhiệm chuyển đổi và mã hóa dữ liệu giữa các định dạng khác nhau, đảm bảo dữ liệu được hiểu đúng ở đầu nhận.
      * Chức năng chính: Chuyển đổi định dạng dữ liệu (ví dụ: từ JSON sang XML). Mã hóa và giải mã dữ liệu. Nén và giải nén dữ liệu.
    - Tầng Ứng dụng (Application Layer):
      * Nhiệm vụ: Đây là tầng gần với người dùng nhất, cho phép các ứng dụng giao tiếp với mạng. Nó cung cấp các giao diện cho người dùng hoặc chương trình truy cập tài nguyên mạng.
      * Chức năng chính: Cung cấp dịch vụ cho các ứng dụng (ví dụ: duyệt web, gửi email). Quản lý các giao tiếp mạng cấp cao. Cung cấp cơ chế xác thực người dùng và dịch vụ.

**Lợi ích của mô hình OSI:**

* 1. Chuẩn hóa quy trình giao tiếp giữa các hệ thống và thiết bị.
  2. Giảm độ phức tạp bằng cách chia nhỏ hệ thống thành các tầng riêng biệt.
  3. Dễ dàng bảo trì và phát triển khi mỗi tầng hoạt động độc lập.
  4. Tăng khả năng tương thích giữa các thiết bị và phần mềm khác nhau.

##### **Mô hình TCP/IP**

Mô hình TCP/IP (Transmission Control Protocol/Internet Protocol) là kiến trúc mạng thực tế được sử dụng trên Internet và các mạng máy tính hiện đại. Nó định nghĩa cách dữ liệu được truyền từ máy tính này đến máy tính khác thông qua mạng lưới toàn cầu. Mô hình này bao gồm 4 tầng chính, mỗi tầng đảm nhận các chức năng riêng để truyền thông tin một cách hiệu quả.



Hình 2. Mô hình TCP/IP

Tổng quan về 4 tầng trong mô hình TCP/IP:

* + - 1. Application Layer (Tầng Ứng dụng):
         1. Nhiệm vụ: Đây là tầng cung cấp các giao diện trực tiếp cho người dùng và ứng dụng truy cập vào dịch vụ mạng, cho phép các ứng dụng giao tiếp với nhau thông qua mạng.
         2. Chức năng: Cung cấp dịch vụ cho các ứng dụng mạng. Định nghĩa các phương thức giao tiếp dữ liệu giữa ứng dụng và dịch vụ mạng.
      2. Transport Layer (Tầng Giao vận)
         1. Nhiệm vụ: Tầng này đảm bảo truyền dữ liệu đáng tin cậy giữa các thiết bị đầu cuối. Nó quản lý chia nhỏ dữ liệu thành các phân đoạn (segments), kiểm soát lỗi, và kiểm soát lưu lượng.
         2. Chức năng: Thiết lập và duy trì kết nối giữa hai thiết bị. Phân đoạn và tái lắp ráp dữ liệu. Quản lý kiểm soát lưu lượng và khắc phục lỗi.
      3. Internet Layer (Tầng Mạng)
         1. Nhiệm vụ: Tầng này chịu trách nhiệm định tuyến dữ liệu qua nhiều mạng để đến đúng đích. Nó sử dụng các địa chỉ IP để xác định nguồn và đích của các gói dữ liệu.
         2. Chức năng: Gán địa chỉ IP cho các thiết bị trong mạng. Định tuyến và phân phối gói dữ liệu đến đích. Quản lý phân mảnh dữ liệu để phù hợp với kích thước mạng.
      4. Network Access Layer (Tầng Liên kết mạng)
         1. Nhiệm vụ: Tầng này quản lý kết nối vật lý giữa các thiết bị trong cùng một mạng hoặc mạng cục bộ (LAN). Nó bao gồm việc truy cập vào các phương tiện truyền dữ liệu và đảm bảo dữ liệu được truyền thành công qua kênh vật lý.
         2. Chức năng: Điều khiển việc truyền và nhận dữ liệu qua cáp hoặc sóng vô tuyến. Phát hiện và xử lý lỗi liên kết dữ liệu. Đóng gói dữ liệu vào khung (frame) và truyền qua phương tiện vật lý.

- Ưu điểm của mô hình TCP/IP:

* Thực tiễn và được sử dụng rộng rãi: Là nền tảng của Internet và các mạng hiện đại.
* Tính linh hoạt cao: Hỗ trợ nhiều loại mạng và giao thức khác nhau.
* Khả năng định tuyến mạnh mẽ: Cho phép dữ liệu được truyền qua nhiều mạng khác nhau.
* Tương thích với nhiều thiết bị và ứng dụng: Hỗ trợ cả IPv4 và IPv6.
* Nhược điểm của mô hình TCP/IP:
  + - Không phải mô hình phân lớp hoàn toàn: Một số tầng gộp nhiều chức năng, gây khó khăn khi phát triển hệ thống phức tạp.
    - Bảo mật không được tích hợp: TCP/IP không cung cấp cơ chế bảo mật nội tại, cần thêm các giao thức bảo mật khác như TLS hoặc VPN.
    - Quản lý địa chỉ phức tạp: Với sự mở rộng của Internet, IPv4 đang gặp phải vấn đề cạn kiệt địa chỉ, buộc phải chuyển sang IPv6.

### Các giao thức

Các giao thức thường dùng:

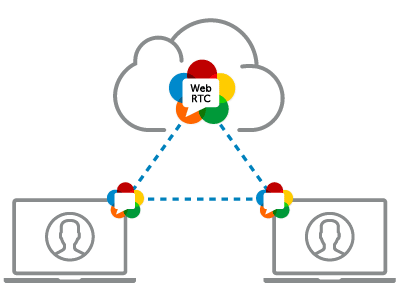
* HTTP/HTTPS: Giao thức truyền tải siêu văn bản dùng cho web.
* FTP: Giao thức truyền file giữa các máy tính.
* SMTP/IMAP/POP3: Các giao thức truyền và nhận email.
* WebSocket: Giao thức giúp truyền dữ liệu thời gian thực hai chiều giữa máy khách và máy chủ.
* UDP và TCP: Giao thức truyền tải dữ liệu giữa các thiết bị, TCP có tính tin cậy cao hơn UDP.

1. HTTP/HTTPS:
   * + HTTP (HyperText Transfer Protocol): Là giao thức cơ bản cho các trang web trên Internet. Nó cho phép máy khách (trình duyệt) và máy chủ web trao đổi thông tin thông qua các yêu cầu (request) và phản hồi (response).
     + HTTPS (HTTP Secure): Phiên bản bảo mật của HTTP, sử dụng TLS/SSL để mã hóa dữ liệu nhằm bảo vệ thông tin người dùng khi truyền qua mạng.
     + Ưu điểm: HTTPS cung cấp mã hóa để bảo mật thông tin. Được hỗ trợ bởi mọi trình duyệt và nền tảng.
2. FTP:
   * + FTP (File Transfer Protocol) là giao thức cho phép truyền tải tệp tin giữa hai thiết bị qua mạng. Nó thường được sử dụng để upload/download dữ liệu lên server hoặc chuyển tệp giữa máy chủ và máy khách.
     + Phiên bản mở rộng: SFTP (SSH File Transfer Protocol): Bảo mật hơn, sử dụng mã hóa SSH. FTPS: FTP kết hợp với SSL/TLS để mã hóa dữ liệu.
     + Ứng dụng: Quản trị viên sử dụng FTP để upload trang web lên máy chủ. Chuyển tệp tin lớn giữa các thiết bị trong mạng nội bộ hoặc qua Internet.
     + Nhược điểm: Phiên bản FTP gốc không mã hóa dữ liệu, dễ bị tấn công.
3. SMTP/IMAP/POP3:
   * + SMTP (Simple Mail Transfer Protocol): Giao thức dùng để gửi email từ máy khách đến máy chủ email và giữa các máy chủ email với nhau.
     + IMAP (Internet Message Access Protocol): Cho phép truy cập email từ nhiều thiết bị và đồng bộ hóa giữa máy chủ và máy khách. Email được lưu trên máy chủ, chỉ tải về khi người dùng mở.
     + POP3 (Post Office Protocol version 3): Tải toàn bộ email về máy khách và thường xóa bản sao trên máy chủ. Không đồng bộ tốt như IMAP, nhưng tiết kiệm dung lượng server.
     + Ứng dụng: SMTP để gửi email, IMAP/POP3 để nhận email trên các ứng dụng như Gmail, Outlook.
4. WebSocket:
   * + WebSocket là giao thức cho phép truyền thông hai chiều giữa máy khách và máy chủ thông qua kết nối liên tục. Không giống HTTP, WebSocket duy trì một kết nối mở để trao đổi dữ liệu thời gian thực mà không cần liên tục gửi request mới.
     + Ưu điểm: Hiệu quả cao cho các ứng dụng thời gian thực như chat, game online, hoặc truyền video. Giảm độ trễ do không phải liên tục thiết lập lại kết nối như HTTP.
     + Ứng dụng: Chatbots, ứng dụng nhắn tin, game nhiều người chơi. Truyền thông dữ liệu giữa các client và server trong thời gian thực.
5. UDP và TCP:
   * + TCP (Transmission Control Protocol):
       - TCP là giao thức đảm bảo độ tin cậy cao khi truyền dữ liệu. Nó kiểm soát lỗi bằng cách xác nhận (ACK) mỗi khi dữ liệu đến nơi và sắp xếp lại dữ liệu theo đúng thứ tự.
       - Đặc điểm: Truyền dữ liệu nguyên vẹn và đúng thứ tự. Có khả năng kiểm soát lỗi và đảm bảo dữ liệu được gửi lại nếu mất mát.
       - Ứng dụng: Trình duyệt web (HTTP), email (SMTP), chuyển file (FTP).
     + UDP (User Datagram Protocol)
       - UDP là giao thức nhẹ hơn và nhanh hơn TCP, không có cơ chế kiểm soát lỗi hoặc xác nhận. Điều này giúp UDP phù hợp cho các ứng dụng thời gian thực, nơi tốc độ quan trọng hơn tính toàn vẹn dữ liệu.
       - Đặc điểm: Không kiểm soát lỗi hoặc đảm bảo dữ liệu đến nơi. Phù hợp cho các ứng dụng streaming và truyền thông trực tiếp.
       - Ứng dụng: Voice over IP (VoIP), video conferencing, game online.

### Sơ lược về một số công nghệ

##### WebRTC (Web Real-Time Communication)

* Mô tả: WebRTC là một công nghệ mã nguồn mở cho phép truyền tải dữ liệu đa phương tiện thời gian thực (audio, video, dữ liệu) trực tiếp giữa các trình duyệt và thiết bị mà không cần cài thêm plugin.
* Đặc điểm:
  + Hỗ trợ truyền video/audio và chia sẻ màn hình.
  + Mô hình P2P giúp giảm tải cho máy chủ trung gian.
  + Dùng STUN và TURN server để hỗ trợ truyền dữ liệu qua tường lửa và NAT.
* Ứng dụng:
  + Hội nghị truyền hình, chat video (Google Meet, Zoom).
  + Streaming và chia sẻ dữ liệu trực tiếp.



Hình . WebRTC

##### ZegoCloud

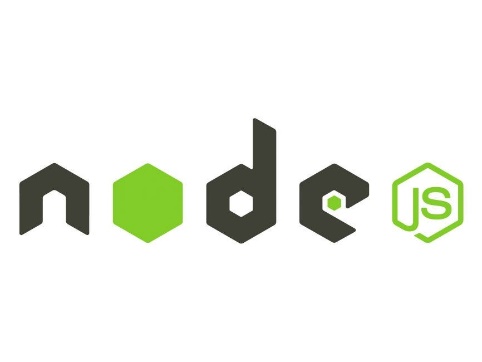
* Mô tả: là một nền tảng giao tiếp thời gian thực cung cấp các dịch vụ cho phép người dùng thực hiện các cuộc gọi video/audio đa phương tiện thông qua internet. ZEGOCLOUD cung cấp một bộ công cụ phát triển phần mềm (SDKs) mạnh mẽ để giúp các nhà phát triển dễ dàng xây dựng các ứng dụng có khả năng hỗ trợ hội nghị video/audio thời gian thực và streaming.
* Đặc điểm:
  + Chất lượng cuộc gọi cao, độ trễ thấp.
  + Tích hợp dễ dàng với nhiều nền tảng.
  + Chế độ đa người dùng.
  + Quản lý người dùng và phòng họp linh hoạt.
  + Hỗ trợ NAT Traversal và Firewall.
  + Chế độ livestream và chia sẻ màn hình.
  + Quản lý chất lượng video và âm thanh.
* Ứng dụng: Hội nghị truyền hình và giao tiếp nhóm, Giáo dục trực tuyến, Chăm sóc sức khỏe từ xa, Ứng dụng xã hội và giải trí.



Hình . Nền tảng ZegoCloud

##### Node.js

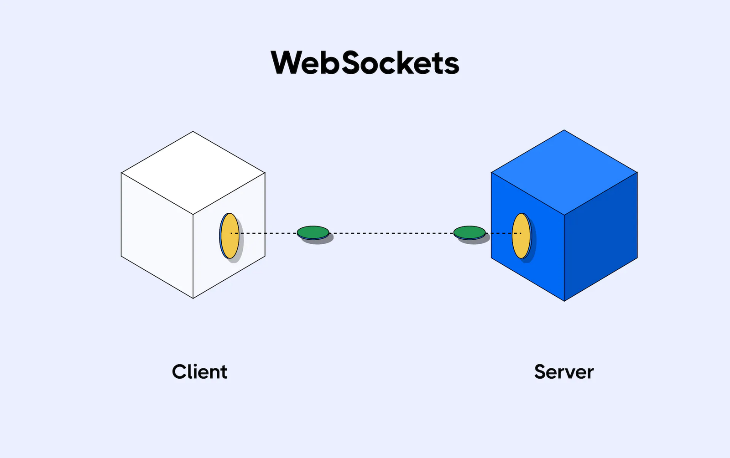
* Mô tả: Node.js là môi trường chạy JavaScript phía server giúp xây dựng các ứng dụng mạng với hiệu năng cao và khả năng xử lý không đồng bộ.
* Đặc điểm:
  + Phù hợp với các ứng dụng thời gian thực như chat và game.
  + Cung cấp thư viện mạnh mẽ cho việc triển khai WebSocket và HTTP server.
  + Khả năng xử lý nhiều kết nối đồng thời thông qua cơ chế event-driven.
* Ứng dụng: Dùng trong ứng dụng chat, API server, hoặc ứng dụng IoT.



Hình . Node.js

##### WebSocket

* Mô tả: WebSocket là một giao thức cho phép truyền dữ liệu hai chiều giữa máy khách và máy chủ trong thời gian thực qua một kết nối mở và liên tục.
* Đặc điểm:
  + Giảm độ trễ và tối ưu cho dữ liệu thời gian thực.
  + Thường được sử dụng trong trò chuyện trực tuyến, game nhiều người chơi, và streaming.
* Ứng dụng: Chatbots, bảng điều khiển trực tuyến, hệ thống báo cáo thời gian thực.



Hình . WebSocket

##### STUN và TURN Server

* STUN Server:
  + Mục đích: Giúp các thiết bị tìm ra địa chỉ IP công cộng của chúng khi hoạt động sau NAT.
  + Ứng dụng: Trong WebRTC để thiết lập kết nối P2P.
* TURN Server:
  + Mục đích: Truyền dữ liệu qua máy chủ trung gian khi kết nối P2P bị chặn bởi tường lửa hoặc NAT.
  + Ứng dụng: Đảm bảo kết nối trong các ứng dụng truyền thông thời gian thực (WebRTC, hội nghị truyền hình).

##### 

# CHƯƠNG 2. PHÂN TÍCH THIẾT KẾ HỆ THỐNG

### Giới thiệu sơ lược về bài toán

* Trong thời đại công nghệ hiện nay, nhu cầu kết nối và giao tiếp từ xa ngày càng tăng cao. Các công cụ hội nghị trực tuyến như Zoom, Microsoft Teams, và Google Meet đã trở thành phần không thể thiếu trong công việc, học tập, và các cuộc họp cá nhân. Bài toán đặt ra là làm thế nào để xây dựng một chương trình hội nghị trực tuyến audio/video đơn giản, hiệu quả và có khả năng hoạt động mượt mà trên các môi trường mạng khác nhau.
* Mục tiêu bài toán
  + Bài toán cần giải quyết yêu cầu thiết kế và phát triển hệ thống hội nghị audio/video cho phép:
    - Người dùng có thể tham gia từ nhiều thiết bị và nhiều địa điểm khác nhau.
    - Duy trì kết nối âm thanh, video chất lượng cao với độ trễ thấp.
    - Cho phép các thành viên điều khiển camera, micro và nhắn tin trong thời gian thực.
    - Đảm bảo kết nối an toàn, ổn định ngay cả khi gặp phải tường lửa hoặc NAT.
* Công nghệ và giải pháp
  + Bài toán sẽ được giải quyết bằng cách sử dụng:
    - WebRTC: Một giao thức mã nguồn mở cho truyền tải âm thanh và video theo mô hình peer-to-peer (P2P).
    - Agora SDK: Giải pháp client-server cho phép điều phối và hỗ trợ kết nối khi P2P gặp vấn đề (như do NAT hoặc tường lửa).
    - JavaScript: Ngôn ngữ chính để xây dựng ứng dụng chạy trên trình duyệt web, giúp người dùng dễ dàng truy cập mà không cần cài đặt phần mềm.
* Thách thức của bài toán
  + Quản lý kết nối P2P: Khi có nhiều người tham gia, việc duy trì kết nối giữa các máy sẽ phức tạp và yêu cầu tối ưu hiệu năng.
  + Tương thích mạng: Một số mạng có tường lửa hoặc NAT có thể gây khó khăn cho kết nối P2P, cần đến TURN server để hỗ trợ.
  + Bảo mật: Đảm bảo dữ liệu âm thanh, video được mã hóa để tránh rủi ro nghe trộm hoặc bị tấn công.
  + Trải nghiệm người dùng: Cần đảm bảo độ trễ thấp và giao diện thân thiện để người dùng dễ dàng sử dụng.

### Phân tích thiết kế hệ thống

##### Yêu cầu của hệ thống

* + 1. Yêu cầu chức năng
    - Đăng nhập và tạo phòng họp: Người dùng có thể đăng nhập và khởi tạo một phòng hội nghị mới hoặc tham gia vào phòng có sẵn.
    - Hội nghị âm thanh và video: Hệ thống cho phép truyền tải âm thanh và video theo thời gian thực giữa các người dùng.
    - Quản lý người dùng trong phòng: Hiển thị danh sách người tham gia và cập nhật trạng thái khi có người vào hoặc rời phòng.
    - Điều khiển camera và micro: Người dùng có thể bật/tắt camera và micro trong quá trình hội nghị.
    - Nhắn tin văn bản thời gian thực: Cho phép người dùng gửi tin nhắn nhanh đến các thành viên trong phòng.
    - Thoát và đóng kết nối: Khi thoát phòng, kết nối sẽ được hủy và các tài nguyên được giải phóng.
    1. Yêu cầu phi chức năng
* Hiệu năng: Đảm bảo truyền tải video và âm thanh mượt mà với độ trễ thấp.
* Tính ổn định: Duy trì kết nối ổn định ngay cả khi mạng yếu, thông qua TURN server nếu cần.
* Bảo mật: Mã hóa dữ liệu âm thanh và video trong quá trình truyền.
* Khả năng mở rộng: Hỗ trợ nhiều người dùng trong cùng một phòng.

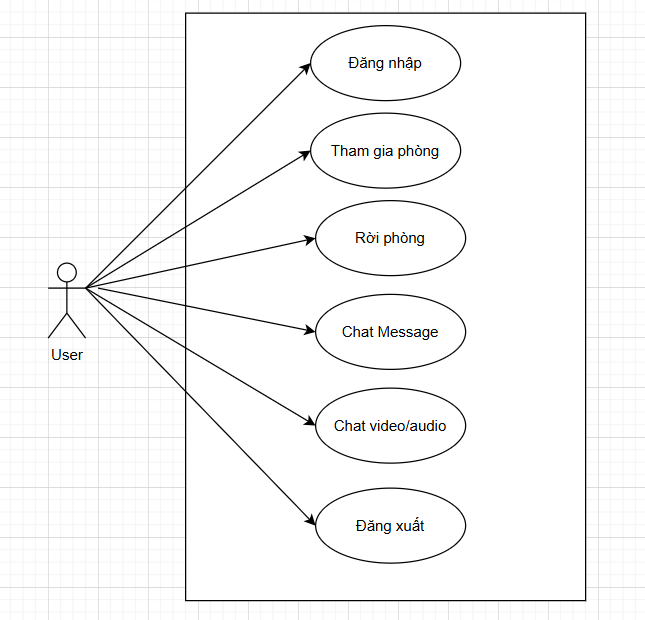
##### Kiến trúc hệ thống

* + 1. Mô hình kiến trúc
* Mô hình client-server với Agora làm server trung gian quản lý kết nối.
* Sử dụng WebRTC để truyền tải dữ liệu P2P giữa các người dùng nếu điều kiện mạng cho phép.
  + 1. Thành phần chính của hệ thống
* Client-side (Web App): Giao diện cho người dùng đăng nhập, tạo phòng và tham gia cuộc họp. Thực hiện gọi WebRTC API và kết nối với Agora RTM SDK.
* Server-side (Agora): Quản lý đăng nhập, phiên họp, và các kênh truyền dữ liệu. Điều phối kết nối thông qua STUN/TURN để duy trì truyền thông

##### Sơ đồ hệ thống

* + 1. Sơ đồ luồng dữ liệu
* Người dùng đăng nhập và gửi yêu cầu tạo/join phòng → Server Agora.
* Client nhận phản hồi và thiết lập kết nối WebRTC để truyền dữ liệu.
* Khi có thành viên mới tham gia, thông báo sự kiện được gửi đến tất cả client.
  + 1. Sơ đồ Use-Case

|  |  |
| --- | --- |
| Use Case | Mô tả |
| Đăng nhập | Người dùng đăng nhập vào hệ thống |
| Tạo phòng | Khởi tạo một phòng mới |
| Tham gia phòng | Người dùng vào một phòng đã có sẵn |
| Truyền âm thanh/video | Phát và nhận luồng âm thanh/video |
| Nhắn tin | Gửi tin nhắn và văn bản đến các thành viên |
| Điều khiển thiết bị | Bật/tắt camera và micro |
| Chia sẻ màn hình | Người dùng có thể chia sẻ màn hình của mình với những người tham gia khác. |
| Rời phòng | Người dùng thoát khỏi phòng. |



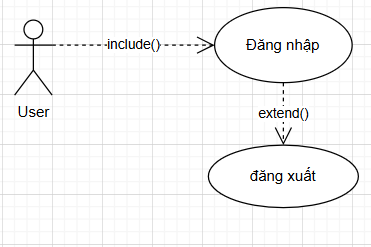
Hình . Sơ đồ Use Case

Mô tả Use-Case:

* Đăng nhập/Đăng ký: Người dùng cần đăng nhập hoặc đăng ký tài khoản để sử dụng ứng dụng.
* Tạo phòng: Người dùng có thể tạo một phòng hội nghị mới, nhận mã phòng để chia sẻ với người khác.
* Tham gia phòng: Người dùng nhập mã phòng hoặc nhấn vào liên kết để tham gia phòng hội nghị.
* Chia sẻ màn hình: Người dùng có thể chia sẻ toàn bộ màn hình hoặc một cửa sổ cụ thể cho những người tham gia khác.
* Bật/tắt camera và micro: Người dùng điều chỉnh các thiết bị video và âm thanh của mình.
* Gửi tin nhắn: Người dùng có thể gửi tin nhắn văn bản để giao tiếp ngoài giọng nói và video.
* Rời phòng: Người dùng thoát khỏi phòng khi không còn tham gia hội nghị.
  + 1. Chức năng
       1. Đăng nhập và đăng xuất

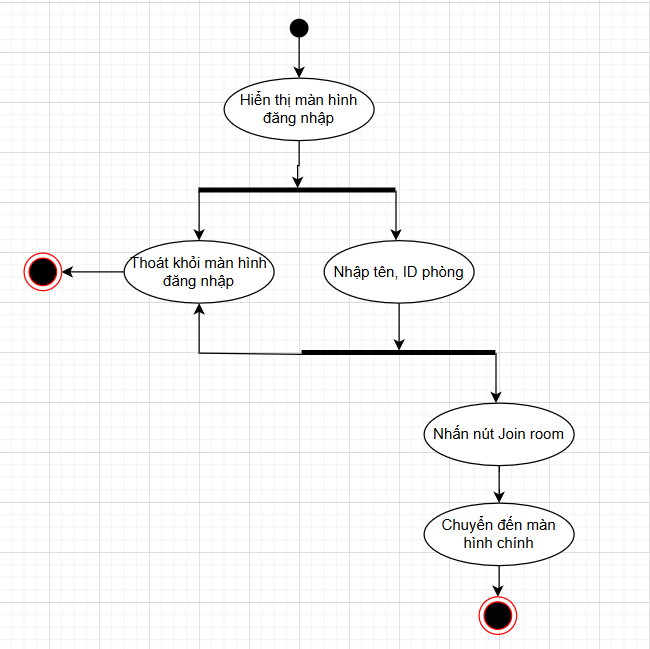
1. Sơ đồ use-case đăng nhập

* Người dùng nhập thông tin và mật khẩu của phòng cần tham gia.
* Nhấn nút “Tham gia phòng”.
* Đăng nhập thành công, hệ thống sẽ báo trong boxchat của room.
* Nhấn vào button rời room để rời phòng.
* Nhấn đăng xuất hoặc tắt trình duyệt sẽ đăng xuất.



Hình . Sơ đồ Use-Case của quá trình đăng nhập

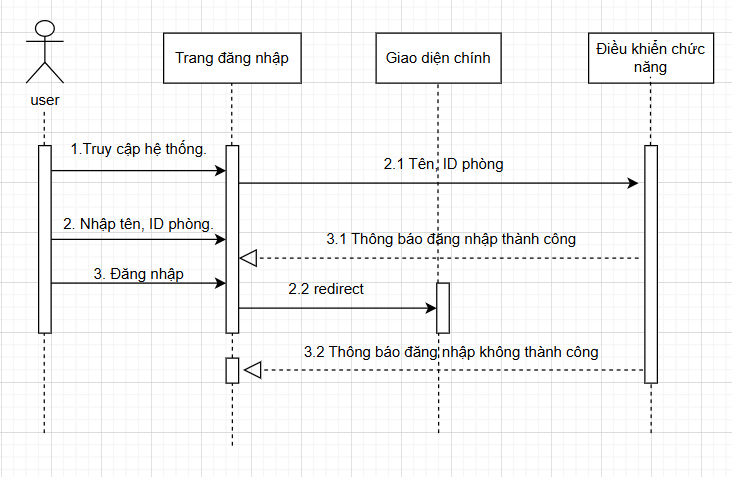
1. Biểu đồ hoạt động đăng nhập:

* Người dùng khởi động ứng dụng và chọn tùy chọn "Đăng nhập".
* Hệ thống yêu cầu nhập thông tin đăng nhập (tên người dùng và mật khẩu).
* Người dùng nhập thông tin và nhấn "Join Room".
* Hệ thống kiểm tra thông tin:
  + Nếu đúng, hệ thống xác thực và chuyển người dùng vào trang chủ.
  + Nếu sai, hiển thị thông báo lỗi và quay lại bước nhập thông tin.
* Người dùng truy cập các tính năng sau khi đăng nhập thành công.

Hình . Biểu đồ hoạt động của chức năng đăng nhập

1. Biểu đồ tuần tự đăng nhập:

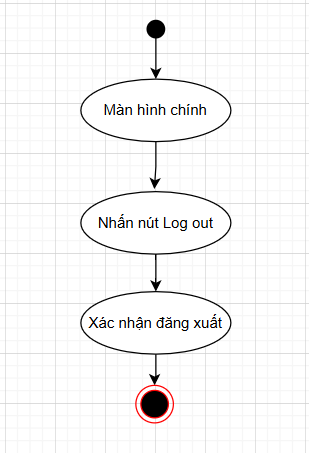
* Người dùng gửi yêu cầu đăng nhập (chứa thông tin tên người dùng và mật khẩu) đến Máy chủ xác thực.
* Máy chủ xác thực kiểm tra thông tin đăng nhập:
  + Nếu hợp lệ, Máy chủ trả về thông báo thành công và gửi người dùng tới Giao diện chính.
  + Nếu không hợp lệ, Máy chủ trả về lỗi và yêu cầu nhập lại.
* Người dùng có thể thao tác trên ứng dụng sau khi đăng nhập thành công.



Hình . Biểu đồ tuần tự chức năng đăng nhập

1. Biểu đồ hoạt động đăng xuất:

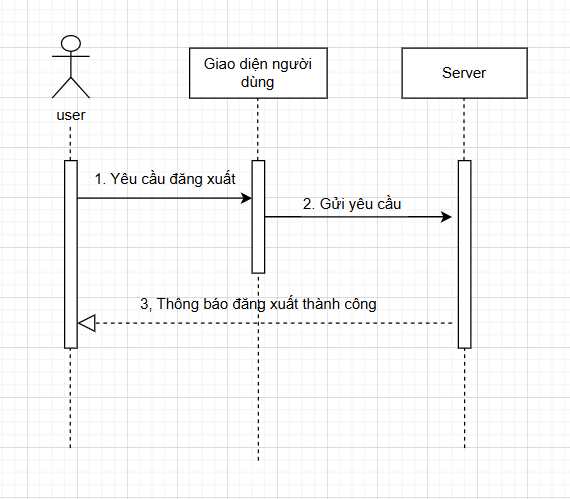
* Người dùng nhấn nút "Đăng xuất" trong giao diện ứng dụng.
* Hệ thống xác nhận và thực hiện đăng xuất.
* Người dùng quay về màn hình đăng nhập hoặc màn hình chính, tùy ứng dụng.



Hình . Biểu đồ hoạt động chức năng đăng xuất

1. Biểu đồ tuần tự đăng xuất:

* Người dùng nhấn nút "Đăng xuất".
* Máy chủ nhận yêu cầu đăng xuất và xác nhận đăng xuất.
* Máy chủ kết thúc phiên làm việc của người dùng và chuyển hướng Người dùng về Giao diện đăng nhập

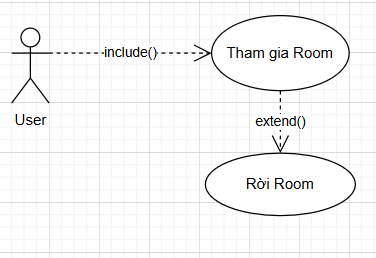


Hình . Biểu đồ tuần tự chức năng đăng xuất

* + - 1. Tham gia Room

1. Use case

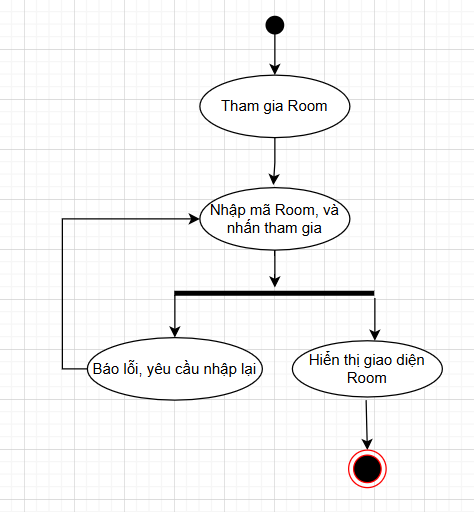
* Người dùng nhập mã Room.
* Hệ thống xác thực mã Room.
* Hệ thống cho phép người dùng truy cập vào Room nếu mã hợp lệ.



Hình . Sơ đồ Use case của tham gia Room

1. Biểu đồ hoạt động

* Bắt đầu: Người dùng truy cập vào giao diện chính và chọn "Tham gia Room".
* Hệ thống yêu cầu người dùng nhập mã Room.
* Người dùng nhập mã Room và nhấn "Tham gia".
* Hệ thống xác thực mã Room:
  + Nếu hợp lệ: Hệ thống cho phép người dùng vào Room và hiển thị giao diện Room.
  + Nếu không hợp lệ: Hệ thống hiển thị thông báo lỗi và yêu cầu nhập lại.
* Kết thúc: Người dùng đã tham gia Room thành công.



Hình . Biểu đồ hoạt động chức năng tham gia Room

1. Biểu đồ tuần tự

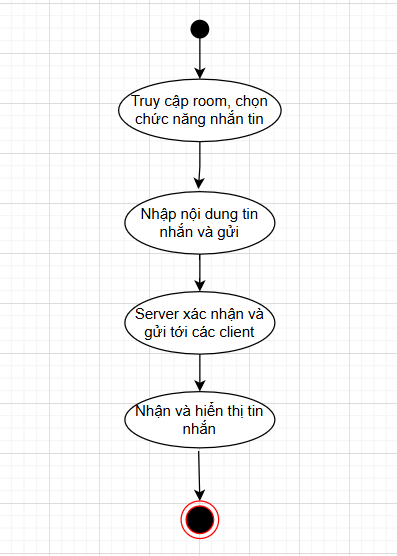
* Người dùng gửi yêu cầu tham gia Room đến Máy chủ.
* Máy chủ nhận mã Room và kiểm tra:
  + Nếu mã hợp lệ: Máy chủ gửi thông báo thành công và kết nối người dùng vào Room.
  + Nếu không hợp lệ: Máy chủ gửi thông báo lỗi và yêu cầu nhập lại mã Room.
* Người dùng nhận thông báo và tiến hành truy cập Room nếu thành công hoặc nhập lại mã nếu gặp lỗi.
  + - 1. Chức năng Message

1. Sơ đồ Use-Case

* Người dùng gửi tin nhắn trong room.
* Hệ thống chuyển tiếp tin nhắn đến người nhận.
* Người dùng nhận tin nhắn từ các thành viên khác trong room.

1. Biểu đồ hoạt động

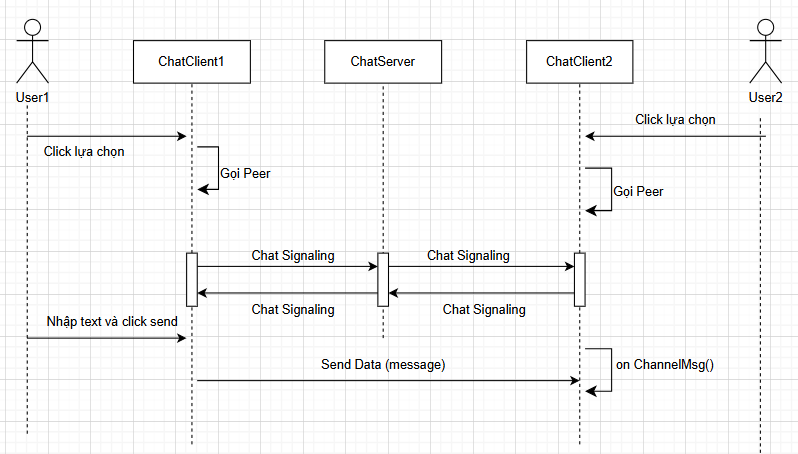
* Bắt đầu: Người dùng truy cập vào room và chọn chức năng nhắn tin.
* Người dùng nhập nội dung tin nhắn và gửi.
* Hệ thống xác nhận tin nhắn và gửi đến server để xử lý.
* Server chuyển tiếp tin nhắn đến các thành viên trong room.
* Các thành viên nhận tin nhắn và hiển thị trên giao diện của họ.
* Kết thúc: Tin nhắn đã được gửi và nhận thành công.



Hình . Biểu đồ hoạt động chức năng nhắn tin

1. Biểu đồ tuần tự

* Người dùng nhập tin nhắn và gửi đến Máy chủ.
* Máy chủ nhận tin nhắn và lưu vào cơ sở dữ liệu (nếu có).
* Máy chủ chuyển tiếp tin nhắn đến tất cả các Người nhận trong room.
* Người nhận nhận được tin nhắn từ server và hiển thị trên giao diện chat.
* Người gửi và Người nhận tiếp tục trao đổi tin nhắn lặp lại theo các bước trên.



Hình . Biểu đồ tuần tự chức năng text message

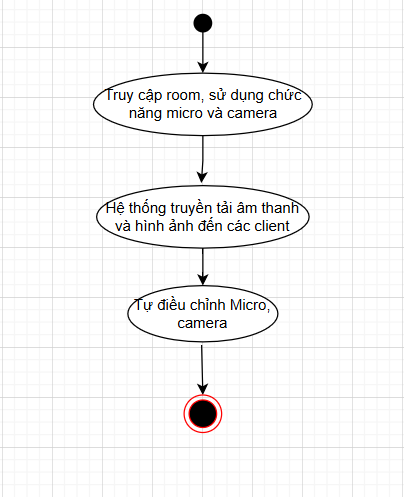
* + - 1. Chức năng audio-voice chat

a) Sơ đồ Use-Case

* Người dùng tham gia phòng chat với chức năng audio và video.
* Người dùng có thể bật/tắt micro và camera.

b) Biểu đồ hoạt động

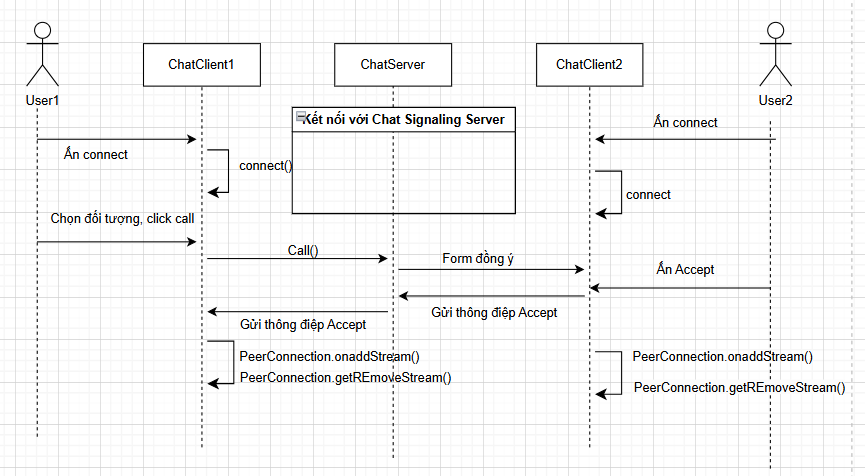
* Bắt đầu: Người dùng tham gia phòng chat.
* Người dùng bật/tắt micro và camera để chuẩn bị tham gia.
* Hệ thống thiết lập kết nối peer-to-peer (P2P) giữa người dùng và các thành viên khác trong phòng chat.
* Audio/Video Streaming: Hệ thống truyền tải âm thanh và hình ảnh của người dùng đến các thành viên khác và ngược lại.
* Người dùng có thể điều chỉnh micro và camera trong suốt cuộc trò chuyện.
* Kết thúc: Khi người dùng thoát phòng chat, hệ thống ngắt kết nối.



Hình . Biểu đồ hoạt động chức năng video/call

c) Biểu đồ tuần tự

* Người dùng yêu cầu tham gia phòng chat và gửi yêu cầu bật audio/video.
* Hệ thống thiết lập kết nối và xác nhận kết nối thành công.
* Người dùng bắt đầu gửi audio/video stream đến Máy chủ.
* Máy chủ tiếp nhận audio/video stream và chuyển tiếp đến các Người nhận.
* Người nhận nhận được audio/video stream và hiển thị trực tiếp trên giao diện.
* Người dùng có thể bật/tắt micro và camera, và các thay đổi này sẽ được cập nhật và thông báo cho Người nhận.
* Khi Người dùng rời phòng, Máy chủ ngắt kết nối và dừng chia sẻ audio/video với các Người nhận.



Hình . Biểu đồ tuần tự sự thiết lập và gọi audio-voice chat

# CHƯƠNG 3. CHƯƠNG TRÌNH DEMO

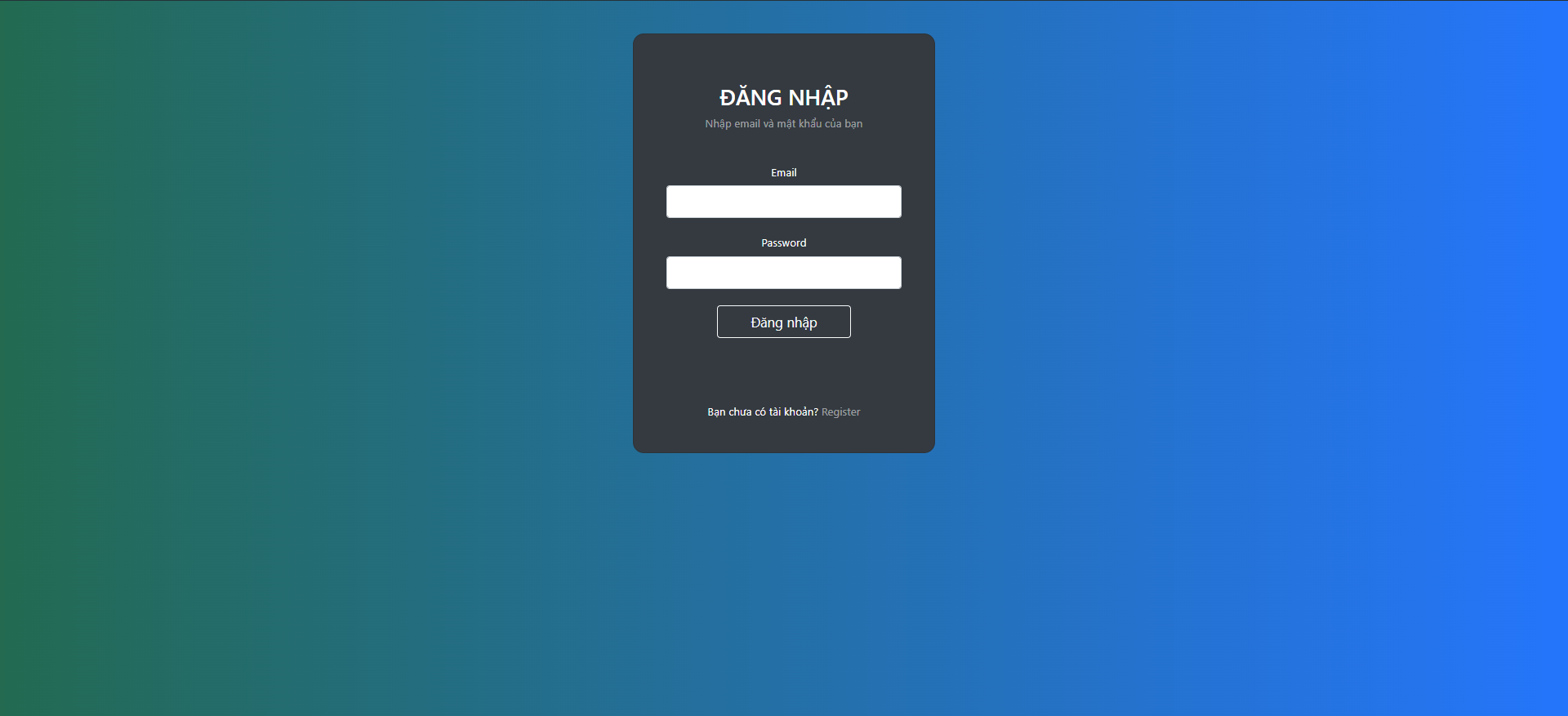
### 1. Thiết kế giao diện người dùng

- Màn hình chính: Giao hiện cho phép người dùng nhập tên và mã phòng hoặc tạo phòng mới. Phần giao diện rõ ràng và đơn gian để người dùng dễ thao tác.

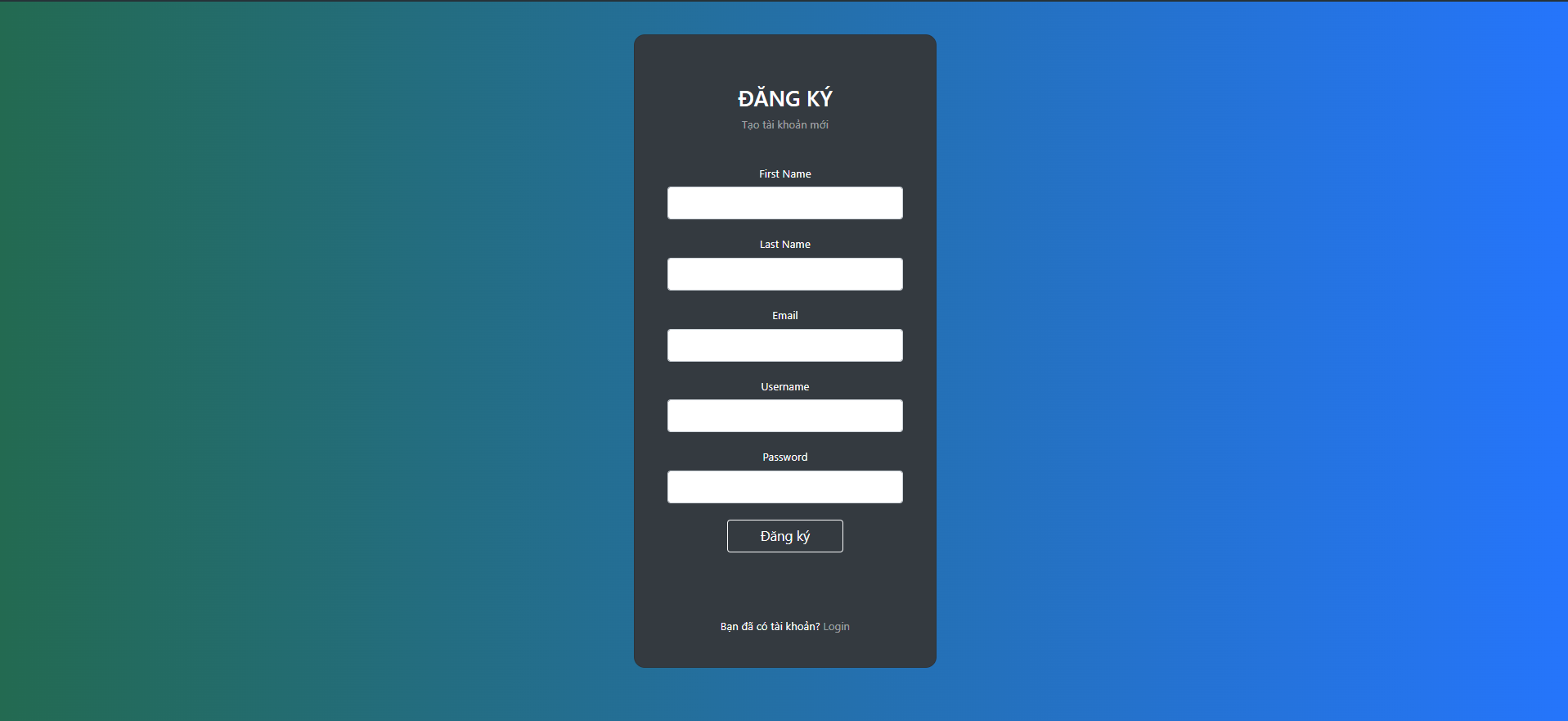
* Thanh nhập thông tin:
* Trường nhập tên người dùng: Cho phép người dùng nhập tên của mình.
* Trường mã phòng: Cho phép nhập mã để tham gia một phòng họp cụ thể.
* Nút "Tạo phòng mới": Nếu không muốn nhập mã phòng, người dùng có thể chọn tùy chọn để hệ thống tự động tạo phòng mới với mã phòng duy nhất.
* Thiết kế giao diện:
  + Tối giản và rõ ràng, chủ yếu dùng các tông màu thân thiện, như trắng và xanh, nhằm tạo cảm giác chuyên nghiệp nhưng dễ tiếp cận.
  + Các nút bấm lớn với chữ rõ ràng, dễ nhìn trên cả thiết bị nhỏ.
  + Hỗ trợ hướng dẫn nhanh (tooltip) khi người dùng di chuột hoặc nhấn giữ vào nút hoặc trường dữ liệu.
* Phòng họp trực tuyến:
* Khung video: Hiển thị luồng video từ camera của người dùng và các thành viên khác trong phòng.
  + Kích thước khung: Video người dùng chính nằm ở một khung lớn. Video của các thành viên khác được hiển thị dạng lưới nhỏ hơn.
  + Điều chỉnh linh hoạt: Hỗ trợ phóng to thu nhỏ từng khung video. Các video hiển thị dưới dạng "mặc định là mờ" nếu camera của người khác đang tắt.
  + Hiệu ứng chuyển động: Khi một thành viên mới bật camera, giao diện có hiệu ứng mờ dần hoặc thu phóng để thêm phần sinh động.
* Thanh công cụ điều khiển: Gồm các nút bật hoặc tắt camera, microphone, và thoát khỏi phòng. Mỗi nút này có các biểu tượng dễ nhận biết, cùng với hiệu ứng để báo hiệu trạng thái hiện tại.
  + Nút bật/tắt camera: Chuyển đổi trạng thái camera kèm biểu tượng rõ ràng, như biểu tượng camera hoặc đường chéo.
  + Nút bật/tắt microphone: Chức năng tương tự camera.
  + Nút thoát phòng: Được đặt cách biệt để tránh nhầm lẫn. Hỏi xác nhận trước khi rời khỏi phòng.
  + Nút chia sẻ màn hình: Hỗ trợ bật chức năng chia sẻ màn hình với giao diện thông báo người xem đang nhìn vào nội dung nào.
* Khung chat: Cho phép gửi và nhận tin nhắn trong thời gian thực qua RTM SDK.
  + Gửi/nhận tin nhắn: Giao diện dạng hộp thoại, cho phép gửi tin nhắn văn bản. Tin nhắn được hiển thị theo dạng bong bóng (bên phải là tin của mình, bên trái là của người khác).
  + Tích hợp biểu tượng cảm xúc. Thông báo khi ai đó gõ tin nhắn. Hiển thị danh sách người đang online trong phòng.

- Giao diện sự kiện tham gia/ rời khỏi: Khi có người tham gia hoặc rời khỏi phòng, giao diện sẽ hiển thị thông báo để người dùng biết số lượng người đang có mặt trong phòng.

* Khi có người tham gia: Hệ thống hiển thị popup nhỏ ở góc màn hình (vd: “A vừa tham gia phòng.”). Cập nhật danh sách người tham gia.
* Khi có người rời khỏi phòng: Popup tương tự xuất hiện với thông báo: “B đã rời khỏi phòng.” Giao diện điều chỉnh lại bố cục nếu cần, như sắp xếp lại khung video.
* Hệ thống thông báo tổng số người trong phòng trên thanh trạng thái.
* Khung thông báo trạng thái hệ thống: Giao diện cần cung cấp khung thông báo trạng thái toàn hệ thống nhằm hỗ trợ người dùng biết rõ hành động của họ và trạng thái hiện tại của phòng họp. Dưới đây là các yếu tố quan trọng:
  + Thanh trạng thái hệ thống:
    - Hiển thị thông báo như kết nối ổn định, đang đợi phản hồi từ server, hoặc có sự cố mạng.
    - Màu sắc được mã hóa, như màu xanh cho trạng thái bình thường và đỏ khi xảy ra lỗi.
  + Cảnh báo quyền truy cập: Nếu người dùng cố bật chức năng mà không có quyền, ví dụ bật camera khi hệ thống không được cấp quyền, một thông báo sẽ hiện ra kèm hướng dẫn khắc phục.
  + Hiển thị trạng thái người tham gia Bên cạnh mỗi tên người tham gia trong danh sách, trạng thái (online, mute, bật video...) sẽ được biểu diễn bằng các biểu tượng như đèn tín hiệu.



Hình . Đăng nhập



Hình . Đăng ký

### 2. Chức năng chính

- Tạo phòng và tham gia phòng: Cho phép người dùng nhập mã phòng hoặc tạo phòng mới. Sau khi phòng được tạo, mã phòng sẽ được cung cấp để chia sẻ với những người khác.

* Tính năng: Người dùng có thể nhập mã phòng để tham gia hoặc tạo phòng mới.
* Hoạt động: Sau khi một phòng mới được tạo, mã phòng duy nhất sẽ được cung cấp để chia sẻ với những người khác.
* Ứng dụng: Hỗ trợ tổ chức cuộc họp hoặc hội thảo trực tuyến, với mã phòng đóng vai trò như "chìa khóa" để truy cập phòng.

- Gọi video P2P: Sử dụng WebRTC, người dùng có thể thực hiện các cuộc gọi video trực tiếp với nhau. WebRTC sẽ quản lý việc truyền tải dữ liệu video và âm thanh trực tiếp, đảm bảo độ trễ thấp.

* Tính năng: WebRTC được sử dụng để triển khai tính năng gọi video peer-to-peer.
* Hoạt động:
  + WebRTC xử lý việc kết nối trực tiếp giữa các người tham gia, đảm bảo dữ liệu video và âm thanh được truyền tải nhanh chóng.
  + Sử dụng ICE (Interactive Connectivity Establishment) để xác định đường dẫn tối ưu giữa các máy.
* Lợi ích: Độ trễ thấp, chất lượng ổn định và tiết kiệm băng thông do không qua máy chủ trung gian.

- Nhắn tin tức thời: Agora RTM SDK hỗ trợ gửi và nhận tin nhắn tức thời giữa các thành viên trong phòng.

* Tính năng: Agora RTM SDK được sử dụng để hỗ trợ gửi và nhận tin nhắn tức thời trong phòng.
* Hoạt động:
  + Người dùng gửi tin nhắn văn bản đến các thành viên khác.
  + Tin nhắn có thể chứa nội dung văn bản hoặc dữ liệu bổ sung (như trạng thái).
* Ứng dụng: Thích hợp để giao tiếp, chia sẻ thông tin phụ trợ trong phòng họp.

- Bật/tắt camera và microphone: Các nút chức năng cho phép người dùng bật hoặc tắt camera và microphone, giúp điều chỉnh khả năng truyền phát video và âm thanh trong cuộc hội thoại.

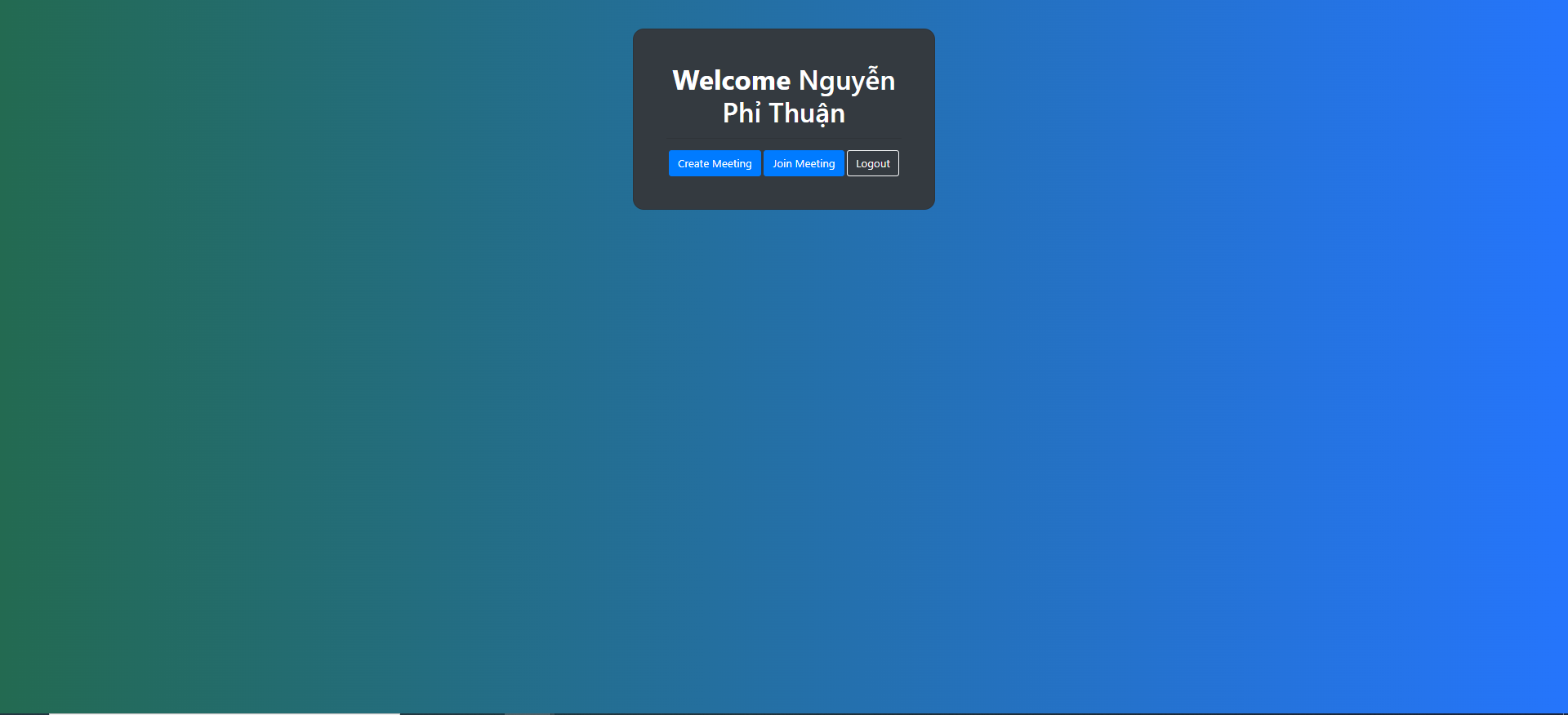
* Tính năng: Cung cấp các nút để người dùng điều khiển thiết bị đầu vào của mình.
* Hoạt động:
  + Camera và microphone có thể được kích hoạt hoặc vô hiệu hóa tùy theo trạng thái hiện tại của cuộc hội thoại.
  + Trạng thái hiển thị qua giao diện, giúp nhận biết trạng thái hiện tại (bật/tắt).
* Ứng dụng: Đảm bảo quyền riêng tư khi cần thiết.

- Chia sẻ màn hình: Chức năng chia sẻ màn hình giúp người dùng hiển thị nội dung màn hình của họ cho các thành viên khác trong phòng hội thoại, hỗ trợ tốt hơn cho các buổi thuyết trình, đào tạo, và làm việc nhóm từ xa.

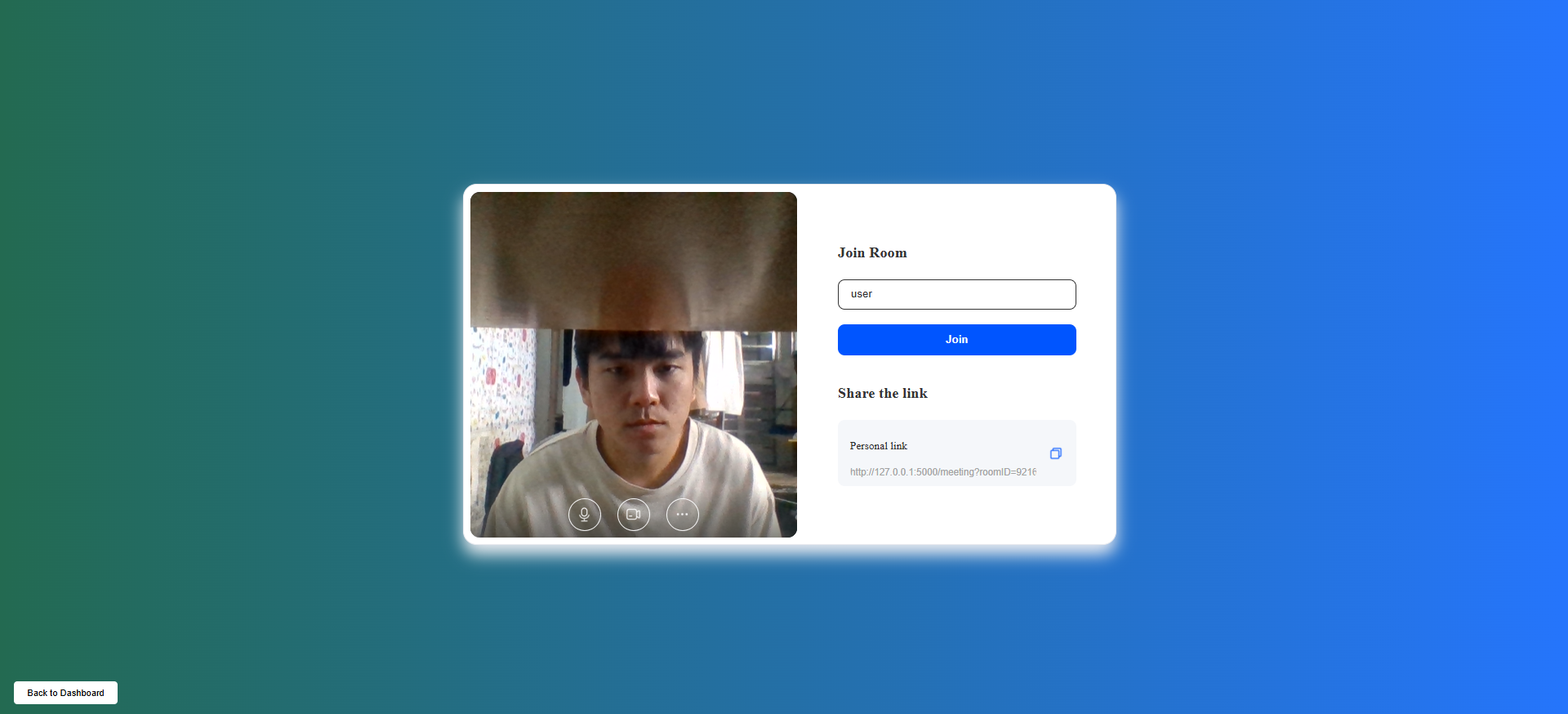
* Tính năng: Người dùng có thể chia sẻ nội dung trên màn hình của họ với các thành viên khác trong phòng.
* Hoạt động:
  + WebRTC hoặc các API hỗ trợ (như Screen Capture API) được sử dụng để truyền tải nội dung màn hình.
  + Người tham gia nhận video màn hình qua kết nối trực tiếp.
* Ứng dụng: Hữu ích trong việc thuyết trình, đào tạo từ xa, và làm việc nhóm.

- Thoát khỏi phòng: Khi người dùng rời khỏi phòng, họ sẽ được đưa trở lại màn hình chính. Các thành viên khác trong phòng sẽ nhận được thông báo về việc này.

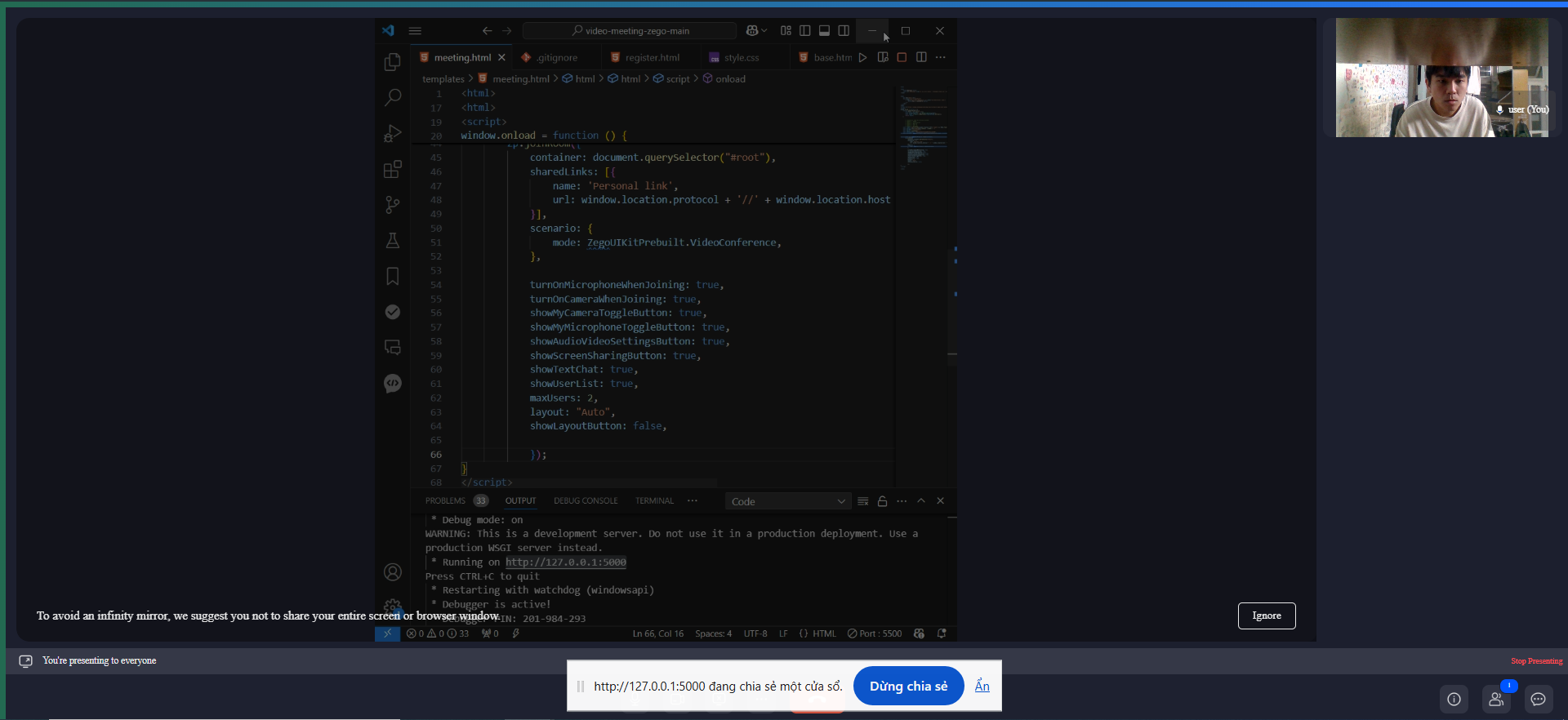
* Tính năng: Người dùng thoát khỏi phòng họp trực tuyến.
* Hoạt động:
  + Người dùng được chuyển trở lại màn hình chính khi thoát.
  + Thông báo được gửi đến các thành viên khác để cập nhật số lượng người tham gia.
* Ứng dụng: Đảm bảo quá trình rời phòng diễn ra một cách rõ ràng và có quản lý trạng thái.



Hình . Giao diện sau khi tham gia



Hình . Giao diện tạo cuộc họp



Hình . Chức năng chia sẻ màn hình



Hình . Giao diện box chat

# KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

### 1. Kết luận

Qua quá trình phát triển và thử nghiệm, ứng dụng hội nghị truyền hình trực tuyến đã đạt được các mục tiêu chính đã đề ra, bao gồm các chức năng như gọi video, nhắn tin, quản lý người dùng, và chia sẻ màn hình. Kết quả cho thấy:

* Khả năng kết nối ổn định: Ứng dụng đã đáp ứng tốt yêu cầu kết nối video và audio ổn định giữa nhiều người dùng, với độ trễ chấp nhận được, đảm bảo chất lượng giao tiếp thời gian thực.
* Khả năng mở rộng: Kiến trúc WebRTC và Agora RTM được áp dụng giúp dễ dàng mở rộng và tích hợp thêm các dịch vụ khác mà không ảnh hưởng đến hiệu suất.
* Trải nghiệm người dùng: Giao diện người dùng thân thiện, dễ sử dụng, với các tính năng như bật/tắt camera, mic, và chia sẻ màn hình linh hoạt.

Tuy nhiên, trong quá trình phát triển, một số hạn chế đã được ghi nhận, chẳng hạn như phụ thuộc vào kết nối mạng và chất lượng đường truyền của người dùng, gây ảnh hưởng đến trải nghiệm nếu mạng không ổn định.

Sau khi hoàn thành đề tài, em đã đạt được nhiều kiến thức và kỹ năng quan trọng trong cả lý thuyết và thực nghiệm:

* Về lý thuyết:
  + Đã nắm vững các khái niệm cốt lõi về giao thức truyền thông WebRTC và các thành phần của Agora RTM SDK, cùng các kỹ thuật kết nối mạng P2P cho truyền video và audio. Điều này giúp hiểu sâu hơn về mô hình truyền thông thời gian thực và cơ chế quản lý băng thông.
  + Học cách phân tích và xây dựng một ứng dụng giao tiếp thời gian thực dựa trên các nguyên tắc truyền thông cơ bản, giúp hệ thống hoạt động ổn định và hiệu quả.
* Về thực nghiệm:
  + Đã ứng dụng các kiến thức lý thuyết để xây dựng một ứng dụng hội nghị truyền hình với các chức năng như gọi video, nhắn tin, và chia sẻ màn hình. Việc triển khai giúp rèn luyện khả năng xử lý lỗi và tối ưu hóa hiệu suất trong môi trường thực tế.
  + Thực hành triển khai và cấu hình WebRTC và Agora SDK cho các ứng dụng thực tế, cũng như tinh chỉnh giao diện người dùng để tạo ra trải nghiệm thân thiện hơn.

### 2. So sánh mục tiêu

So với mục tiêu ban đầu, đề tài đã đạt được hầu hết các yêu cầu đề ra, bao gồm việc thiết lập kết nối video và audio ổn định, khả năng nhắn tin tức thời, và chia sẻ màn hình. Tuy nhiên, vẫn còn một số hạn chế do:

* Thời gian và kinh nghiệm hạn chế: Một số lỗi nhỏ trong quá trình phát triển (như các vấn đề về băng thông khi chia sẻ màn hình) chưa được giải quyết triệt để. Do giới hạn về thời gian và kinh nghiệm, các giải pháp tối ưu hóa chưa được thực hiện đầy đủ.
* Điều kiện phát triển: Việc triển khai trong môi trường thử nghiệm có sự khác biệt so với môi trường thực tế, dẫn đến một số lỗi khó dự đoán trước.

### 3. Hướng phát triển

Để nâng cao hơn nữa chức năng và khả năng sử dụng của ứng dụng:

* Tích hợp tính năng ghi âm và ghi hình cuộc gọi: Cho phép người dùng lưu lại các phiên họp quan trọng để xem lại khi cần thiết.
* Cải thiện tối ưu hóa băng thông: Bằng cách áp dụng các thuật toán nén và điều chỉnh chất lượng video động dựa trên tốc độ mạng, có thể đảm bảo chất lượng video ổn định hơn trong môi trường mạng yếu.
* Phát triển ứng dụng đa nền tảng: Đưa ứng dụng lên các nền tảng khác như di động và máy tính bảng để tăng sự tiện lợi cho người dùng.
* Bảo mật nâng cao: Tích hợp các phương thức mã hóa mới nhằm đảm bảo tính bảo mật và quyền riêng tư của người dùng trong các cuộc hội thoại.
* Ứng dụng trí tuệ nhân tạo (AI): Tự động ghi chú cuộc họp hoặc nhận diện khuôn mặt để theo dõi sự tham gia của các thành viên.

# TÀI LIỆU THAM KHẢO

1. WebRTC Official Documentation: <https://webrtc.org>

2. Agora RTM SDK Documentation: <https://docs.agora.io>

3. Mozilla Developer Network (MDN) - WebRTC API : https://developer.mozilla.org/en-US/docs/Web/API/WebRTC\_API

4. Giáo trình lập trình mạng: 2013- Advanced Network Programming – Principles and Techniques-1.pdf

5. https://youtu.be/QsH8FL0952k?si=tUHQdJEBn-QuW3bR

# PHỤ LỤC

x

x

x