**ĐẠI HỌC QUỐC GIA HÀ NỘI**

**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ**

****

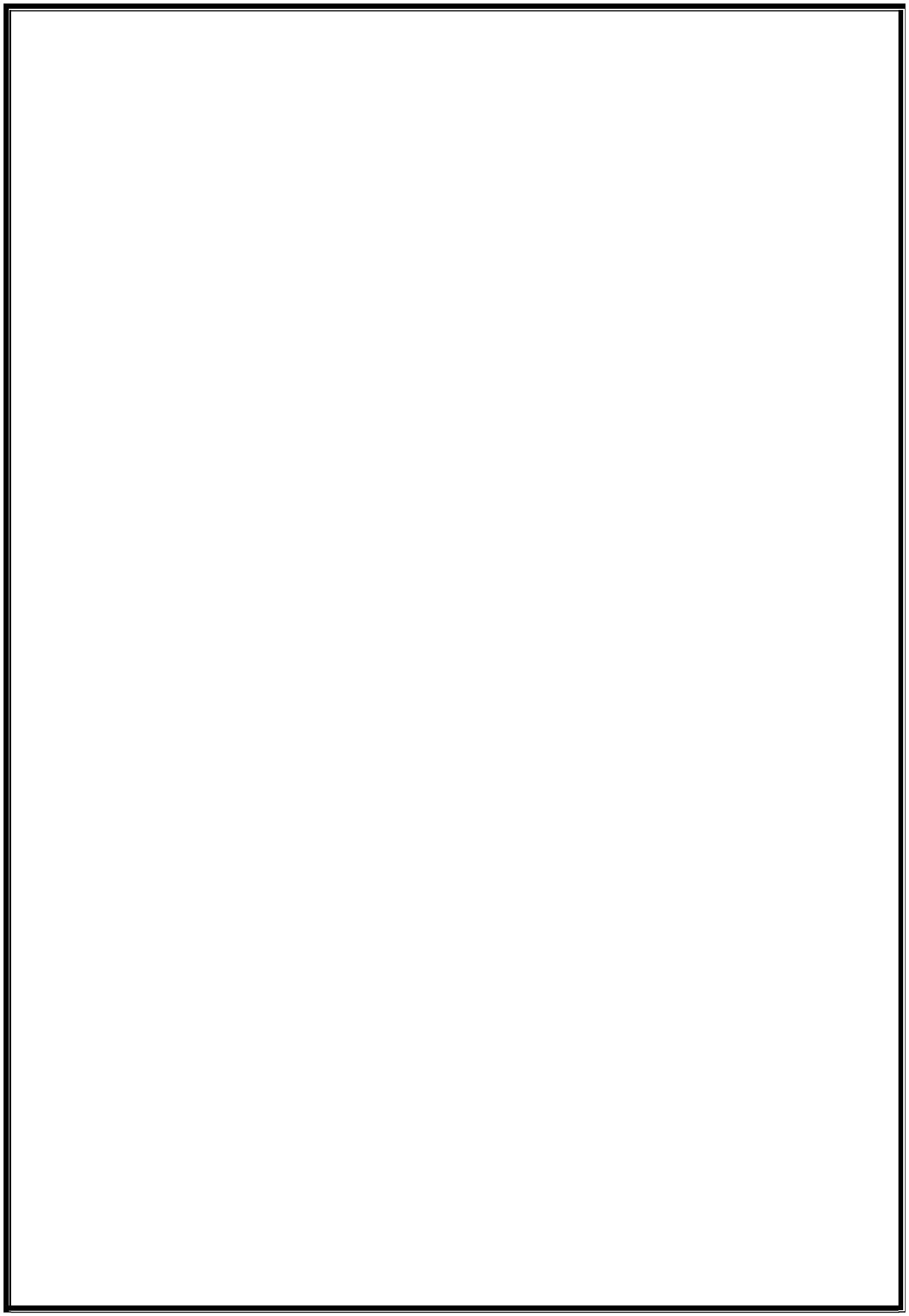
**LƯƠNG VĂN QUÝ**

**NGHIÊN CỨU ỨNG DỤNG WEBRTC CHO HỆ THỐNG E-LERNING**

**KHÓA LUẬN TỐT NGHIỆP ĐẠI HỌC HỆ CHÍNH QUY**

**Ngành: Khoa học máy tính**

**HÀ NỘI - 2019**

**ĐẠI HỌC QUỐC GIA HÀ NỘI**

**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ**

**LƯƠNG VĂN QUÝ**

**XÂY DỰNG HỆ THỐNG DẠY HỌC TRỰC TUYẾN**

**KHÓA LUẬN TỐT NGHIỆP ĐẠI HỌC HỆ CHÍNH QUY**

**Ngành: Khoa học máy tính**

**Cán bộ hướng dẫn: TS. Nguyễn Thị Hậu**

**Cán bộ đồng hướng dẫn:**

**HÀ NỘI - 2019**

**LỜI CẢM ƠN**

Khóa luận tốt nghiệp ngày được thực hiện tại Đại học Công nghệ - Đại học Quốc gia Hà Nội dưới sự hướng dẫn của TS. Nguyễn Thị Hậu. Xin được gửi lời cảm ơn sâu sắc đến cô Nguyễn Thị Hậu về những ý kiến quý báu định hướng cho khóa luận, liên tục quan tâm tạo điều kiện thuận lợi cho tôi trong suốt quá trình hoàn thành khóa luận tốt nghiệp.

Tôi cũng xin được gửi lời cảm ơn chân thành tới khoa Công nghệ thông tin đã tạo điều kiện cho tôi trong suốt quá trình làm khóa luận.

Do thời gian và kiến thức có hạn nên khóa luận không tránh khỏi những thiếu sót, rất mong nhận được những ý kiến góp ý quý báu của thầy cô phản biện.

Hà Nội, ngày tháng năm 2019

Sinh viên

Lương Văn Quý

**LỜI CAM ĐOAN**

Tôi Lương Văn Quý xin cam đoan khóa luận này hoàn toàn do tôi thực hiện dưới sự hướng dẫn của TS. Nguyễn Thị Hậu. Các nội dung, hình ảnh tham khảo đều được tôi trích dẫn đầy đủ. Nếu phát hiện có sự sao chép tôi xin hoàn toàn chịu trách nhiệm.

*Hà Nội, ngày … tháng … năm 2019*

Sinh viên

Lương Văn Quý

MỤC LỤC

[DANH MỤC CÁC TỪ VIẾT TẮT 7](#_Toc5893101)

[DANH MỤC HÌNH VẼ 8](#_Toc5893102)

[MỞ ĐẦU 10](#_Toc5893103)

[1. Đặt vấn đề 10](#_Toc5893104)

[2. Giới thiệu bài toán 10](#_Toc5893105)

[3. Phạm vi và mục tiêu của khóa luận 10](#_Toc5893106)

[4. Phương pháp và bố cục khóa luận 11](#_Toc5893107)

[CHƯƠNG 1. GIỚI THIỆU CÁC CÔNG NGHỆ ĐƯỢC SỬ DỤNG ĐỂ GIẢI QUYẾT BÀI TOÁN 12](#_Toc5893108)

[1.1. WebRTC 12](#_Toc5893109)

[1.2. Openvidu 15](#_Toc5893110)

[1.3. Pdf.js 18](#_Toc5893111)

[1.4. NodeJS 20](#_Toc5893112)

[1.5. MongoDB 20](#_Toc5893113)

[CHƯƠNG 2. PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG 24](#_Toc5893114)

[2.1. Mô hình tổng quan hệ thống 24](#_Toc5893115)

[2.2. Phân tích hệ thống 25](#_Toc5893116)

[2.2.1. Phân tích chức năng người dùng 25](#_Toc5893117)

[2.2.2. Phân tích các luồng sự kiện chính 26](#_Toc5893118)

[2.3. Thiết kế module chức năng 38](#_Toc5893119)

[2.3.1. Module lớp học ảo 38](#_Toc5893120)

[2.3.2. Module quản lý 51](#_Toc5893121)

[2.2. Thiết kế hệ thống server 61](#_Toc5893122)

[2.3. Thiết kế cơ sở dữ liệu 62](#_Toc5893123)

[2.4. Thiết kế giao diện 64](#_Toc5893124)

[CHƯƠNG 3. KIỂM THỬ HỆ THỐNG 69](#_Toc5893125)

[3.1. Điều kiện thử nghiệm 69](#_Toc5893126)

[3.1.1. Máy chủ 69](#_Toc5893127)

[3.1.2. Trình duyệt 69](#_Toc5893128)

[3.2. Kết quả kiểm thử 69](#_Toc5893129)

[CHƯƠNG 4. KẾT LUẬN CHUNG 70](#_Toc5893130)

[4.1. Các đóng góp của khóa luận 70](#_Toc5893131)

[4.2. Một số hướng phát triển tiếp theo 70](#_Toc5893132)

[4.2.1. Chia sẻ màn hình thời gian thực 70](#_Toc5893133)

[4.2.2. Tính năng KPI 70](#_Toc5893134)

[4.2.3. Tính năng trình chiếu các bài giảng trực quan 3D 71](#_Toc5893135)

[4.2.4. Thông báo tin nhắn về điện thoại, gửi thông báo qua email 71](#_Toc5893136)

[4.2.5. Khắc phục lỗi không hoạt động trên trình duyệt Firefox và IE 71](#_Toc5893137)

DANH MỤC CÁC TỪ VIẾT TẮT

|  |  |  |
| --- | --- | --- |
| STT | Từ viết tắt | Từ đầy đủ |
| 1 | WebRTC | Web Real-Time Communication |
| 2 | W3C | World Wide Web Consortium |
| 3 | ODM | Object Document Mapper |
| 4 | KPI | Key Performance Indicator |

DANH MỤC HÌNH VẼ

[Hình 1.1: Kiến trúc của WebRTC 13](#_Toc5890492)

[Hình 1.2: Cách thức hoạt động của WebRTC 14](#_Toc5890493)

[Hình 1.3: Kiến trúc OpenVidu Server 15](#_Toc5890494)

[Hình 1.4: Ví dụ về cách lưu trữ của MongoDB 21](#_Toc5890495)

[Hình 1.5: Ưu điểm của MongoDB so với sql 21](#_Toc5890496)

[Hình 1.6: Mô tả cách hoạt động của module mongoose 23](#_Toc5890497)

[Hình 2.1: Mô hình tổng quan hệ thống 24](#_Toc5890498)

[Hình 2.2: Biểu đồ tuần tự mô tả cách khởi tạo lớp học ảo 39](#_Toc5890499)

[Hình 2.3: Biểu đồ tuần tự mô tả cách một sinh viên join vào lớp học ảo 41](#_Toc5890500)

[Hình 2.4: Biểu đồ tuần tự mô tả cách kết nối giữa các client 43](#_Toc5890501)

[Hình 2.5: Mô tả chức năng thêm slide 44](#_Toc5890502)

[Hình 2.6: Biểu đồ tuần tự mô tả chức năng trình chiếu slide thời gian thực 45](#_Toc5890503)

[Hình 2.7: Biểu đồ tuần tự mô tả chức năng gửi tin nhắn văn bản 46](#_Toc5890504)

[Hình 2.8: Biểu đồ tuần tự mô tả chức năng bật bài kiểm tra 47](#_Toc5890505)

[Hình 2.9: Biểu đồ tuần tự mô tả chức năng kiểm tra 48](#_Toc5890506)

[Hình 2.10: Biểu đồ tuần tự mô tả chức năng chia sẻ video youtube 49](#_Toc5890507)

[Hình 2.11: Biểu đồ tuần tự mô tả chức năng điểm danh tự động 50](#_Toc5890508)

[Hình 2.12: Module quản lý truy cập 51](#_Toc5890509)

[Hình 2.13: Mô hình tổng quan quản lý truy cập hệ thống 52](#_Toc5890510)

[Hình 2.14: Biểu đồ tuần tự quá trình xác thực bằng email và mật khẩu 53](#_Toc5890511)

[Hình 2.15: Module quản lý giảng viên 54](#_Toc5890512)

[Hình 2.16: Mô tả chức năng thêm giảng viên mới 55](#_Toc5890513)

[Hình 2.17: Module quản lý sinh viên 56](#_Toc5890514)

[Hình 2.18: Mô tả chức năng thêm sinh viên mới 57](#_Toc5890515)

[Hình 2.19: Module phê duyệt khóa học 58](#_Toc5890516)

[Hình 2.20: Module chức năng quản lý khóa học 58](#_Toc5890517)

[Hình 2.21: Mô tả chức năng tạo khóa học mới 59](#_Toc5890518)

[Hình 2.22: Module quản lý buổi học 60](#_Toc5890519)

[Hình 2.23: Mô tả chức năng tạo quiz 61](#_Toc5890520)

[Hình 2.24: Cách thức hoạt động của hệ thống server 62](#_Toc5890521)

[Hình 2.25: Mô hình ER 63](#_Toc5890522)

[Hình 2.26: Giao diện trang login 64](#_Toc5890523)

[Hình 2.27: Giao diện trang tạo khóa học mới 65](#_Toc5890524)

[Hình 2.28: Giao diện trang danh sách khóa học 66](#_Toc5890525)

[Hình 2.29: Giao diện trang cập nhật thông tin cá nhân 66](#_Toc5890526)

[Hình 2.30: Giao diện lớp học ảo 67](#_Toc5890527)

[Hình 2.31: Giao diện quản lý khóa học của Quản trị viên 68](#_Toc5890528)

MỞ ĐẦU

## Đặt vấn đề

Giáo dục là một lĩnh vực luôn nhận được sự quan tâm của toàn xã hội. Từ xưa trên thế giới nói chung và Việt Nam nói riêng, quá trình dạy và học thường tập trung tại các trường học hoặc các cơ quan chuyên về đào tạo, phương pháp dạy học là thầy cô sẽ đứng lớp trực tiếp, trình bày bằng bảng, biểu đồ, slideshow,... người học cũng phải có mặt trực tiếp tại lớp học để ghi chép, tiếp thu những kiến thức được người dạy truyền đạt.

Ngày nay với sự phát triển chóng mặt của khoa học công nghệ, mạng internet càng ngày càng mở rộng đã thúc đẩy sự phát triển giáo dục và đào tạo. Các cơ quan, trường học hiện nay đều mong muốn có thể áp dụng các thành tựu công nghệ vào quá trình đào tạo để nâng cao kết quả dạy và học. Trong những năm gần đây, xuất hiện một loại hình đào tạo mới dựa trên internet đã đem lại những chuyển biến đáng kể trong việc dạy và học với những cái tên như “Online learning” (học trực tuyến), “virtual learning” (Học tập ảo), “E-Learning”,…

Được sự gợi ý từ Giảng viên hướng dẫn TS. Nguyễn Thị Hậu và mong muốn có thể áp dụng những kiến thức tích lũy được trong quá trình học tập vào giáo dục hiện nay, tôi đã tìm hiểu để xây dựng một hệ thống lớp học ảo đem đến trải nghiệm giáo dục mới cho việc dạy và học.

## Giới thiệu bài toán

Các hệ thống học tập trực tuyến hiện nay thường cần phải tốn một lượng lớn chi phí để xây dựng và duy trì hoạt động do các hệ thống hiện nay chủ yếu hoạt động hoàn toàn theo nguyên tắc client-server nên server thường phải chịu tải lớn.

Do đó, khóa luận này tập trung xây dựng một hệ thống lớp học ảo với đầy đủ các chức năng cơ bản, đồng thời xây dựng một hệ thống có thể vừa hoạt động theo mô hình client-server vừa có thể hoạt động theo mô hình peer-to-peer.

## Phạm vi và mục tiêu của khóa luận

Khóa luận tập trung xây dựng một hệ thống dạy và học trực tuyến bằng các công nghệ mới xuất hiện, giúp giảm chi phí xây dựng và duy trì hoạt động một hệ thống học tập trực tuyến.

Hệ thống được xây dựng hướng tới tính chất tương tác thời gian thực, giúp tạo ra môi trường dạy và học thân thiện, dễ sử dụng, dễ tương tác.

Đồng thời hệ thống còn giúp giảng viên dễ dàng quản lý lớp học và theo dõi mức độ chuyên cần, đánh giá khả năng tiếp thu của từng sinh viên.

Hệ thống được triển khai dựa trên các công nghệ như NodeJS, MongoDB, WebRTC, OpenVidu, PDFJS.

## Phương pháp và bố cục khóa luận

Khóa luận được chia thành 4 chương với các nội dung sau:

Chương 1 - Giới thiệu tổng quan về các công nghệ được sử dụng để triển khai hệ thống như NodeJS, MongoDB, WebRTC, OpenVidu, PDF.JS.

Chương 2 - Phân tích, thiết kế hệ thống: Chương này tập trung vào phân tích chức năng, phân tích luồng sự kiện, thiết kế module, cơ sở dữ liệu của hệ thống.

Chương 3 – Kiểm thử hệ thống

Chương 4 – Kết luận chung: Tổng kết lại các kết quả đạt được và đề xuất hướng phát triển trong tương lai.

# GIỚI THIỆU CÁC CÔNG NGHỆ ĐƯỢC SỬ DỤNG ĐỂ GIẢI QUYẾT BÀI TOÁN

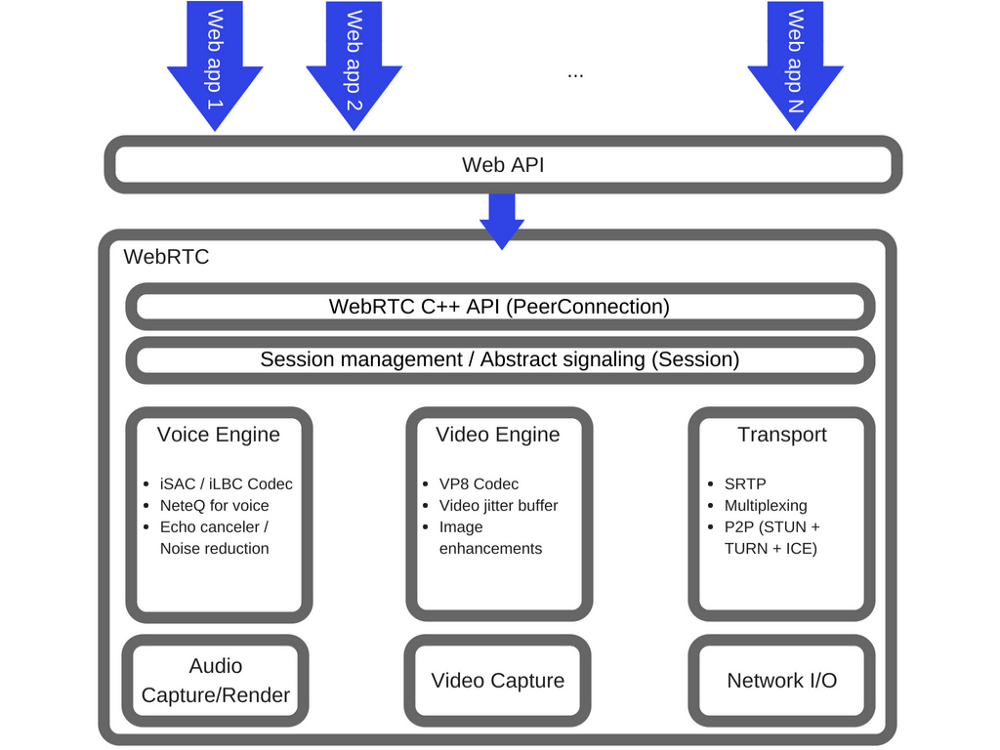
## WebRTC

Hệ thống Viduca đòi hỏi khả năng tương tác thời gian thực, để đạt được mục tiêu đó một công nghệ đã được sử dụng là WebRTC, không những mang đến khả năng tương tác thời gian thực, WebRTC còn giúp giảm tải phía Server do có cơ chế hoạt động peer-to-peer. Hệ thống lớp học ảo Viduca không được phát triển trực tiếp từ WebRTC mà sẽ được triển khai thông qua một nền tảng là OpenVidu. Nền tảng OpenVidu cho phép hệ thống vừa hoạt động theo mô hình client-server, vừa có thể hoạt động theo mô hình peer-to-peer. Do đó, trong giới hạn của khóa luận này, WebRTC chỉ được trình bày một cách tổng quan, không đi vào chi tiết kiến trúc, các giao thức được sử dụng,…

WebRTC (Web Real-Time Communication) là một dự án mã nguồn mở miễn phí cung cấp khả năng giao tiếp thời gian thực cho trình duyệt web và các ứng dụng di động thông qua các giao diện lập trình ứng dụng (API) đơn giản. Nó cho phép truyền tải audio, video hoạt động bên trong trang web bằng cách cho phép giao tiếp ngang hàng trực tiếp (peer-to-peer) loại bỏ yêu cầu phải cài đặt plugin hoặc tải xuống ứng dụng gốc. WebRTC được hỗ trợ bởi Google, Microsoft, Mozilla và Opera, WebRTC đang được chuẩn hóa thông qua World Wide Web Consortium (W3C) và nhóm Internet Engineering Task Force (IETF).

Sứ mệnh của nó là cho phép các ứng dụng thời gian thực phong phú, chất lượng cao được phát triển cho trình duyệt, nền tảng di động và thiết bị IoT, đồng thời cho phép tất cả chúng giao tiếp thông qua một bộ giao thức chung.

Kiến trúc của WebRTC được mô tả ở hình phía dưới:

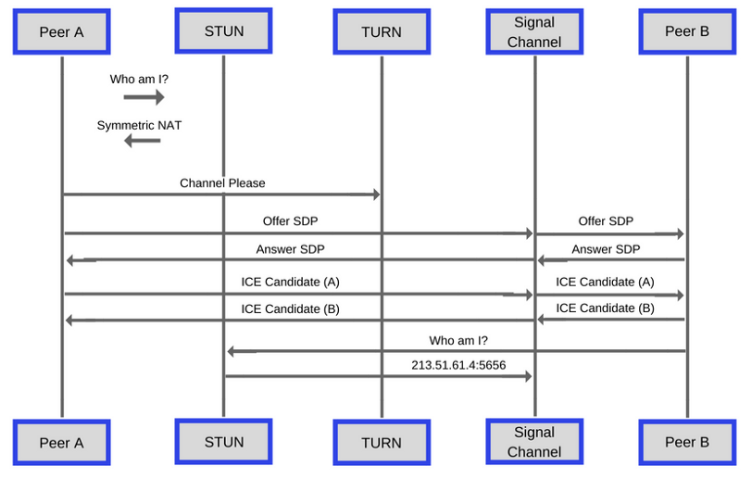


Hình 1.1: Kiến trúc của WebRTC

Trong kiến trúc WebRTC có ba lớp chính:

* Web API: Gồm các API dành cho những nhà phát triển Web như RTCPeerConnection, RTCDataChanel, MediaStream,…
* Session Management: Gồm các API dành cho những nhà phát triển trình duyệt.
* Overridable API: Gồm các API Audio Capture/Render, Video Capture, Network I/O, nhà phát triển trình duyệt có thể phát triển các API của riêng mình.

Dưới đây là hình vẽ mô tả cách thức hoạt động của WebRTC:



Hình 1.2: Cách thức hoạt động của WebRTC

Ở trên ta thấy có Peer A và Peer B, hình vẽ mô tả cách thức thiết lập một kết nối peer-to-peer giữa Peer A và Peer B. STUN và TURN là các server phục vụ cho việc trao đổi dữ liệu giữa Peer A và Peer B. Trong đó STUN có nhiệm vụ xác định các thông tin của các Peer để phục vụ cho việc thiết lập một kết nối peer-to-peer. Trong trường hợp không thể thiết lập một kết nối peer-to-peer từ Peer A tới Peer B, TURN server sẽ đứng giữa để trung chuyển các gói tin giữa Peer A và Peer B.

Bên trên đã giới thiệu một cách tổng quan về kiến trúc của WebRTC và cách thức WebRTC hoạt động. Dưới đây sẽ giới thiệu về OpenVidu, một nền tảng phát triển từ Kurento Server, được triển khai WebRTC, giúp việc phát triển và tích hợp các tính năng thời gian thực một cách đơn giản và nhanh chóng.

## Openvidu

Openvidu là một nền tảng mở nguồn mở giúp bổ sung các tính năng trao đổi dữ liệu thời gian thực như video, audio trong các ứng dụng web hoặc di động. OpenVidu rất dễ để tích hợp với các ứng dụng khác nhau.

OpenVidu được phát triển từ Kurento Server, tuy nhiên ở OpenVidu đã ẩn tất cả các hoạt động cấp thấp, giúp đơn giản hóa việc phát triển ứng dụng bằng cách cung cấp các API đơn giản, hiệu quả.

Bên dưới là kiến trúc của OpenVidu:

https://openvidu.io/img/docs/home/openvidu-new-architecture.png

Hình 1.3: Kiến trúc OpenVidu Server

OpenVidu gồm hai phần: OpenVidu Server và OpenVidu Browser:

* OpenVidu Browser: Đây là một thư viện được sử dụng ở phía Client, nó cho phép tạo một cuộc gọi video, gửi và nhận video,…
* OpenVidu Server: Đây là một ứng dụng phía Server, nó nhận các request từ OpenVidu Browser, thiết lập và quản lý các cuộc gọi video, audio, trao đổi dữ liệu,…

Để tích hợp OpenVidu vào ứng dụng chỉ cần triển khai một máy chủ OpenVidu và một file OpenVidu-browser.js phía người dùng. Tuy nhiên, để tăng tính bảo mật cho hệ thống, trong khóa luận này tôi sử dụng máy chủ NodeJS để khởi tạo kết nối an toàn giữa client đến máy chủ OpenVidu. Cụ thế như sau, để kết nối đến máy chủ OpenVidu ta cần khởi tạo một đối tượng OpenVidu bằng câu lệnh:

var OV = new OpenVidu(OPENVIDU\_URL, OPENVIDU\_SECRET);

Trong đó, OPENVIDU\_URL là địa chỉ của máy chủ OpenVidu, OPENVIDU\_SECRET là khóa bí mật cho phép kết nối đến máy chủ OpenVidu. Nếu chỉ đơn thuần sử dụng bằng cách khởi tạo ở phía client thì sẽ làm lộ khóa bí mật OPENVIDU\_SECRET. Do đó, tôi sử dụng một máy chủ NodeJS để khởi tạo một đối tượng OpenVidu kết nối đến máy chủ OpenVidu, sau đó mới tạo ra session và token trả về cho client. Như vậy ở phía client sẽ không biết được khóa bí mật OPENVIDU\_SECRET, đảm bảo an toàn cho hệ thống.

Để tạo ra một session mới, ta sử dụng câu lệnh:

OV.createSession().then(session => {

}).catch(error => {

});

Hàm này sẽ trả về một session mới, dựa vào session này có thể tạo ra nhiều token, mỗi token sẽ ứng với một người dùng. Session ở đây ta có thể hình dung là một lớp học, còn token chính là ứng với giảng viên và sinh viên trong lớp. Để tạo mới token cho một người dùng ta sử dụng câu lệnh:

session.generateToken(tokenOptions).then(token => {

}).catch(error => {

});

Sau khi token mới được tạo ra sẽ gửi cho phía client, ở phía client vẫn sử dụng file OpenVidu-Browser.js và khởi tạo đối tượng OpenVidu với câu lệnh:

OV = new OpenVidu();

Câu lệnh này ta không truyền tham số OPENVIDU\_URL (địa chỉ máy chủ OpenVidu) và OPENVIDU\_SECRET (khóa bí mật của máy chủ OpenVidu). Sau đó ta phải tạo mới một session ở phía client bằng câu lệnh:

session = OV.initSession();

Session này sẽ được sử dụng để quản lý các sự kiện liên quan đến OpenVidu ở phía client. Sau khi session ở phía client được khởi tạo, nó sẽ tạo một kết nối từ client tới máy chủ OpenVidu bằng câu lệnh:

session.connect(token).then(() => {

}).catch(error => {

});

Trong đó, token chính là token đã được tạo ra ở máy chủ NodeJS trả về cho client. Một số sự kiện đặc biệt được quản lý bằng session ở phía client như streamCreated (sự kiện phát sinh khi một người dùng mới kết nối với session), streamDestroyed (sự kiện phát sinh khi một người dùng thoát khỏi session).

Để bắt sự kiện có stream kết nối tới session ta dùng câu lệnh:

session.on('streamCreated', (event) => {

});

Để bắt sự kiện khi một người dùng thoát khỏi session ta dùng câu lệnh:

session.on('streamDestroyed', (event) => {

removeUserData(event.stream.connection);

});

Để gửi một message giữa các người dùng trong session, ta sử dụng câu lệnh:

session.signal({

data: 'My custom message',

to: [],

type: 'my-chat'

}).then(() => {

console.log('Message successfully sent');

}).catch(error => {

console.error(error);

});

Trong đó, data là text message mà người dùng muốn gửi, mảng to là danh sách các đối tượng sẽ được gửi tin nhắn đến, nếu để trống sẽ là là toàn bộ người dùng trong session, type là kiểu message do người lập trình tự đặt. Để bắt sự kiện có một message gửi đến, ta dùng câu lệnh:

session.on('signal:my-chat', (event) => {

console.log(event.data);

console.log(event.from);

console.log(event.type);

});

Trong đó, my-chat chính là kiểu message do người lập trình đặt, bằng câu lệnh này ta có thể thu được dữ liệu (data) do một người dùng đã gửi tới toàn bộ người dùng khác trong session.

## Pdf.js

Trong lớp ảo Viduca có tính năng trình chiếu slide thời gian thực, để triển khai tính năng này, hệ thống sử dụng Pdf.js.

Pdf.js là một nền tảng dựa trên tiêu chuẩn Web dùng để phân tích cú pháp và hiển thị các tệp Pdf.

Kiến trúc của Pdf.js gồm 3 lớp:

* Lớp core: Là lớp phân tích cú pháp và diễn giải các tệp Pdf, lớp này là nền tảng cho tất cả các lớp tiếp theo.
* Lớp display: Lớp này gồm các API cho phép truy xuất thông tin của tệp Pdf đã được lớp core phân tích.
* Lớp Viewer: Đây là lớp hiển thị file Pdf phía người dùng. Nó có thể được xây dựng lại cho phù hợp với từng ứng dụng.

Để sử dụng Pdf.js ta phải download file pdf.js ở trang <https://mozilla.github.io/pdf.js/> và import vào ứng dụng ở phía client. Sau khi import thì ta có thể sử dụng được đối tượng PDFJS – đây là một đối tượng global chứa các phương thức cơ bản như PDFJS.getDocument(URL).

PDFJS.getDocument() trả về một Promise có thể được sử dụng để đặt mã sẽ được thực thi khi PDF.js thực hiện tìm nạp tài liệu. Cuộc gọi lại thành công của Promise được thông qua một đối tượng có chứa thông tin về tài liệu PDF được tìm nạp. Hàm này được sử dụng như sau:

PDFJS.getDocument(url)

.then(function(pdf) {

return pdf.getPage(1);

})

.then(function(page) {

});

Trong đó url là đường dẫn tới file pdf. Hàm này trả về một đối số là pdf, đây là một đối tượng với các phương thức cơ bản cho việc hiển thị file.

* pdf.getPage(page\_number) được sử dụng để có được các trang riêng lẻ của một tài liệu pdf, page\_number là trang muốn lấy, hàm này trả về một đối tượng page.
* page.getViewport() trả về kích thước trang của tài liệu PDF.
* page.render() được sử dụng để render trang pdf lên màn hình.

## NodeJS

NodeJS là một mã nguồn được xây dựng dựa trên nền tảng Javascript V8 Engine, nó được sử dụng để xây dựng các ứng dụng web.  NodeJS cung cấp các thư viện phong phú ở dạng Javascript Module khác nhau giúp đơn giản hóa việc lập trình và giảm thời gian ở mức thấp nhất. Khi nói đến NodeJS thì phải nghĩ tới vấn đề Realtime. Realtime ở đây chính là xử lý giao tiếp từ client tới máy chủ theo thời gian thực. Do đó, NodeJs là một nền tảng phù hợp cho việc phát triển hệ thống lớp học ảo.

Các đặc trưng của Nodejs:

* **Không đồng bộ**: Tất cả các API của NodeJS đều không đồng bộ (none-blocking), nó chủ yếu dựa trên nền của NodeJS Server và chờ đợi Server trả dữ liệu về. Việc di chuyển máy chủ đến các API tiếp theo sau khi gọi và cơ chế thông báo các sự kiện của Node.js giúp máy chủ để có được một phản ứng từ các cuộc gọi API trước (Realtime).
* **Chạy rất nhanh**: NodeJ được xây dựng dựa vào nền tảng V8 Javascript Engine nên việc thực thi chương trình rất nhanh.
* **Đơn luồng nhưng khả năng mở rộng cao**: Node.js sử dụng một mô hình luồng duy nhất với sự kiện lặp. cơ chế tổ chức sự kiện giúp các máy chủ để đáp ứng một cách không ngăn chặn và làm cho máy chủ cao khả năng mở rộng như trái ngược với các máy chủ truyền thống mà tạo đề hạn chế để xử lý yêu cầu. Node.js sử dụng một chương trình đơn luồng và các chương trình tương tự có thể cung cấp dịch vụ cho một số lượng lớn hơn nhiều so với yêu cầu máy chủ truyền thống như Apache HTTP Server.
* **Không đệm**: NodeJS không đệm bất kì một dữ liệu nào và các ứng dụng này chủ yếu là đầu ra dữ liệu.
* **Có giấy phép**: NodeJS đã được cấp giấy phép bởi MIT License.

## MongoDB

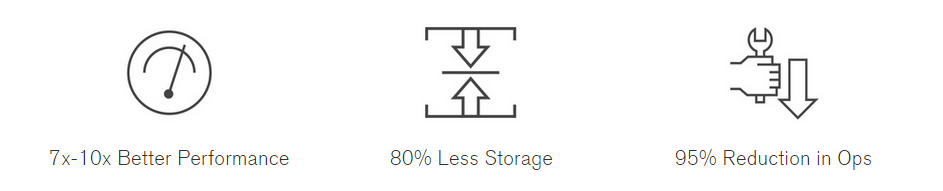
MongoDB là một trong những cơ sở dữ liệu mã nguồn mở NoSQL phổ biến nhất được biết bằng C++. Tính đến tháng 2/2015, MongoDB được xếp thứ 4 trong số các hệ thống cơ sở dữ liệu phổ biến nhất. Nó được phát triển bởi công ty 10gen sau này được biết đến với tên MongoDB Inc..

MongoDB là cơ sở dữ liệu hướng tài liệu, nó lưu trữ dữ liệu trong các document dạng JSON với schema động rất linh hoạt. Nghĩa là bạn có thể lưu các bản ghi mà không cần lo lắng về cấu trúc dữ liệu như là số trường, kiểu của trường lưu trữ. Tài liệu MongoDB tương tự như các đối tượng JSON.



Hình 1.4: Ví dụ về cách lưu trữ của MongoDB

MongoDB là một cơ sở dữ liệu NoSQL hỗ trợ đa nền tảng, nó có thể chạy trên Windows, Linux và Mac...Nó hỗ trợ hầu hết các ngôn ngữ lập trình phổ biến như C#, Java, PHP, Javascript...và các môi trường phát triển khác nhau.



Hình 1.5: Ưu điểm của MongoDB so với sql

MongoDB lưu trữ dữ liệu trong các tài liệu thay vì các bảng. Có thể thay đổi cấu trúc các bản ghi (được gọi là document trong MongoDB) đơn giản chỉ cần thêm mới các trường hoặc xóa các trường có sẵn. Khả năng này của MongoDB giúp trình bày các quan hệ dạng thứ bậc, để lưu trữ mảng, và các cấu trúc phức tạp khác một cách dễ dàng. MongoDB cung cấp hiệu suất cao, tính sẵn có cao và dễ dàng mở rộng.

Trong khóa luận này tôi sử dụng module mongose để thao tác với cơ sở dữ liệu MongoDB. Mongoose là một ODM (Object Document Mapper), nghĩa là nó cho phép định nghĩa các object với một Schema được định nghĩa rõ ràng ánh xạ tới một Mongo Document. Để sử dụng được mongose ta cần install module mongose bằng câu lệnh:

npm install mongoose –save

Sau đó, sử dụng câu lệnh: var mongoose = require('mongoose'); để import module vào dự án. Để kết nối tới cơ sở dữ liệu MongoDB ta sử dụng câu lệnh: mongoose.connect('mongodb://URL/MongodbName', options).then(error => {});

Trong đó, tham số options có thể chứa username và password của cở sở dữ liệu MongoDB, nếu cần. Nếu có lỗi xảy ra khi kết nối đến cơ sở dữ liệu của MongoDB ta có thể xử lý lỗi trong phần then(error => {});

Để sử dụng module mongoose, ta phải định nghĩa một Schema. Sau đây là một Schema đơn giản:

var userSchema = mongoose.Schema({

firstName: String,

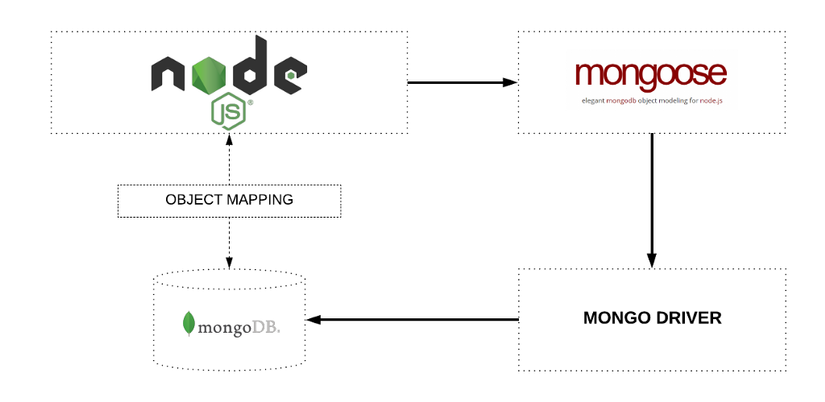
lastName: String

});

Trong đó, firstname và lastname là các thuộc tính của User có kiểu dữ liệu là chuỗi. Để tạo mới một model với Schema đã định nghĩa, ta dùng câu lệnh sau:

var User = mongoose.model('User', userSchema);

Trong đó, User là tên của Model, và userSchema chính là Schema được định nghĩa ở trên. Module mongoose giúp đơn giản hóa cách tương tác với cơ sở dữ liệu MongoDB. Hình bên dưới là mô tả cách mà module mongoose hoạt động:



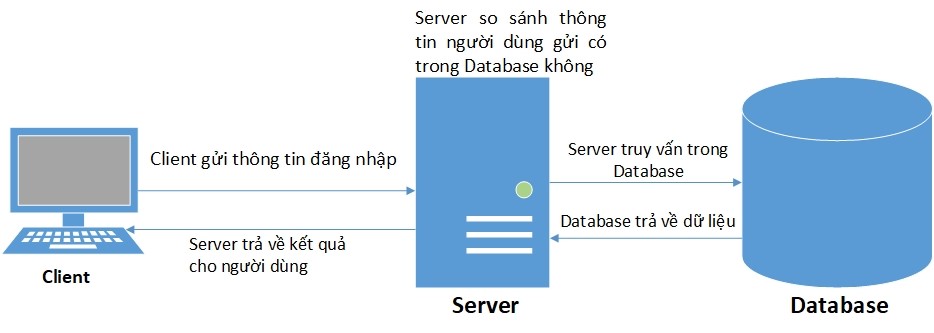
Hình 1.6: Mô tả cách hoạt động của module mongoose

Module mongoose sẽ thông dịch mã để có thể truy vấn giữa đối tượng và định dạng của cơ sở dữ liệu. Do đó, nó sẽ bị chậm hơn so với cách truy vấn trực tiếp bằng ngôn ngữ được hỗ trợ bởi cơ sở dữ liệu. Mặc dù vậy, Mongoose vẫn đem lại lợi ích đáng kể trong việc phát triển so với việc chỉ truy vấn chậm một chút. Điểm mạnh của Mongoose là nó giúp người phát triển có thể giữ tư duy như với các đối tượng của Javascript thay vì phải viết ngôn ngữ thuần túy dành riêng cho cơ sở dữ liệu, do đó giúp giảm thiểu chi phí phát triển và bảo trì. Ngoài ra, module mongoose còn cung cấp các tính năng để thực hiện xác thực và kiểm tra dữ liệu.

# PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG

Nội dung trình bày trong chương 3 gồm bốn phần, tập trung vào phân tích và thiết kế hệ thống. Phần đầu phân tích đối tượng sử dụng và các chức năng cần thiết cho mỗi đối tượng người dùng. Phần hai tập trung trình bày thiết kế các module chức năng của hệ thống, cách thức hoạt động của các chức năng trong mỗi module. Phần ba tập trung trình bày thiết kế hệ thống server, cách thức các server tương tác với nhau để các chức năng hoạt động ổn định. Phần cuối tập trung trình bày thiết kế cơ sở dữ liệu của cả hệ thống.

## Mô hình tổng quan hệ thống



Hình 2.1: Mô hình tổng quan hệ thống

Hệ thống lớp học ảo Viduca hoạt động trên nền tảng Web với mô hình Client-Server. Client sẽ gửi các request đến server, server sẽ tương tác với cơ sở dữ liệu, xử lý các yêu cầu người dùng gửi lên và phản hồi lại cho người dùng.

Ví dụ về tính năng đăng nhập:

Khi người dùng muốn đăng nhập để sử dụng hệ thống, người dùng sẽ phải điền thông tin email và mật khẩu vào form đăng nhập và ấn submit, các thông tin này sau đó sẽ được gửi đến server, server sau khi nhận được yêu cầu đăng nhập của người dùng sẽ truy vấn cơ sở dữ liệu tìm xem có email và mật khẩu trong cơ sở dữ liệu hay không. Nếu server kiếm tra thấy thông tin đăng nhập đúng thì sẽ tạo session và gửi lại cho người dùng để sử dụng hệ thống, nếu thông tin sai server sẽ gửi lại một lỗi cho người dùng yêu cầu đăng nhập lại.

## Phân tích hệ thống

### Phân tích chức năng người dùng

#### Các chức năng cho Giảng viên

Giảng viên là những người sẽ đứng lớp và tương tác trực tiếp với sinh viên trong lớp học ảo. Do đó giảng viên cần có các chức năng về quản lý lớp học như sau: chức năng tạo mới lớp học cho phép giảng viên tạo một lớp học mới và chờ quản trị viên phê duyệt, chức năng chỉnh sửa thông tin của lớp học cho phép giảng viên thay đổi thông tin khi cần thiết ví dụ như thay đổi mật khẩu của khóa học, chức năng xóa khóa học cho phép giảng viên xóa khóa học khi khóa học đã kết thúc hoặc không cần thiết nữa.

Giảng viên cũng cần có chức năng quản lý buổi học để quản lý chặt chẽ lộ trình học của khóa học với các chức năng cần có như tạo mới buổi học, thay đổi thông tin buổi học, thêm các tài liệu, slide, xem được danh sách điểm danh sinh viên nhờ chức năng điểm danh tự động của hệ thống….

Cuối cùng hệ thống phải cung cấp các công cụ dạy học trực tuyến để giảng viên có thể tương tác được với sinh viên. Các công cụ cần có như trình chiếu slide, truyền thông video, audio thời gian thực với chất lượng cao, chat văn bản hỏi đáp, công cụ tạo bài kiểm tra trực tuyến,…

#### Các chức năng cho sinh viên

Hệ thống cần cung cấp các chức năng cho phép sinh viên có thể khai thác thông tin và tương tác được với giảng viên. Sinh viên cần phải có các chức năng đăng ký khóa học, xem lộ trình học, thông tin khóa học, buổi học, tải xuống tài liệu, slide. Hệ thống cũng phải cung cấp cho sinh viên các công cụ để tương tác trực tiếp được với giảng viên như truyền thông video, audio trực tuyến thời gian thực, chat văn bản hỏi đáp, xem trình chiếu slide thời gian thực.

#### Các chức năng của quản trị viên

Quản trị viên là người dùng đặc biệt trong hệ thống với vai trò quản lý người dùng và thông tin. Hệ thống cần cung cấp cho quản trị viên các chức năng để quản trị người dùng như xem danh sách người dùng có trong hệ thống, thêm người dùng mới, chỉnh sửa thông tin người dùng, xóa người dùng khỏi hệ thống.

Với chức năng thêm người dùng mới hệ thống cũng cần cung cấp cho quản trị viên chức năng thêm mới người dùng hiệu quả và nhanh chóng như thêm bằng file dữ liệu hoặc thêm bằng form.

Quản trị viên cũng có nhiệm vụ kiểm soát các khóa học mới do giảng viên tạo ra, do đó hệ thống phải cung cấp cho quản trị viên các chức năng quản lý khóa học như xem danh sách các khóa học, xét duyệt các khóa học mới do giảng viên tạo ra, hay xóa khóa học khỏi hệ thống.

### Phân tích các luồng sự kiện chính

#### Đăng nhập hệ thống Viduca

|  |  |
| --- | --- |
| Tên Use Case | Đăng nhập |
| Tác nhân | Người dùng |
| Sự kiện kích hoạt | Người dùng click vào nút đăng nhập |
| Luồng sự kiện chính:   1. Người dùng nhập email và mật khẩu trên trang đăng nhập 2. Nhấn nút đăng nhập 3. Đăng nhập thành công, hệ thống chuyển hướng sang trang chủ. | |
| Luồng sự kiện phụ:   1. Đăng nhập thất bại: Yêu cầu đăng nhập lại. | |

#### Thêm mới giảng viên

|  |  |
| --- | --- |
| Tên Use Case | Thêm mới giảng viên |
| Tác nhân | Quản trị viên |
| Sự kiện kích hoạt | Người dùng click vào nút thêm mới |
| Luồng sự kiện chính:   1. Quản trị viên vào trang thêm giảng viên mới 2. Chọn file giảng viên hoặc nhập email vào form 3. Nhấn nút thêm mới 4. Thêm mới thành công, hệ thống thông báo đã thêm thành công. | |
| Luồng sự kiện phụ:   1. Thêm mới thất bại: Hệ thống yêu cầu nhập lại. | |

#### Thêm mới sinh viên

|  |  |
| --- | --- |
| Tên Use Case | Thêm mới sinh viên |
| Tác nhân | Quản trị viên |
| Sự kiện kích hoạt | Người dùng click vào nút thêm mới |
| Luồng sự kiện chính:   1. Quản trị viên vào trang thêm sinh viên mới 2. Chọn file sinh viên hoặc nhập email vào form 3. Nhấn nút thêm mới 4. Thêm mới thành công, hệ thống thông báo đã thêm thành công. | |
| Luồng sự kiện phụ:   1. Thêm mới thất bại: Hệ thống yêu cầu nhập lại. | |

#### Xóa Giảng viên

|  |  |
| --- | --- |
| Tên Use Case | Xóa giảng viên |
| Tác nhân | Quản trị viên |
| Sự kiện kích hoạt | Quản trị viên ấn vào nút xóa bên cạnh giảng viên |
| Luồng sự kiện chính:   1. Quản trị viên vào trang danh sách giảng viên 2. Ấn vào nút xóa bên cạnh giảng viên 3. Xóa thành công, hệ thống thông báo đã xóa thành công. | |
| Luồng sự kiện phụ:   1. Xóa thất bại: Hệ thống thông báo xóa thất bại. | |

#### Xóa sinh viên

|  |  |
| --- | --- |
| Tên Use Case | Xóa sinh viên |
| Tác nhân | Quản trị viên |
| Sự kiện kích hoạt | Quản trị viên ấn vào nút xóa bên cạnh sinh viên |
| Luồng sự kiện chính:   1. Quản trị viên vào trang danh sách sinh viên 2. Ấn vào nút xóa bên cạnh sinh viên 3. Xóa thành công, hệ thống thông báo đã xóa thành công. | |
| Luồng sự kiện phụ:   1. Xóa thất bại: Hệ thống thông báo xóa thất bại. | |

#### Phê duyệt khóa học

|  |  |
| --- | --- |
| Tên Use Case | Phê duyệt khóa học |
| Tác nhân | Quản trị viên |
| Sự kiện kích hoạt | Quản trị viên ấn vào nút phê duyệt bên cạnh khóa học |
| Luồng sự kiện chính:   1. Quản trị viên vào trang danh sách khóa học 2. Ấn vào nút phê duyệt bên cạnh khóa học 3. Phê duyệt thành công, hệ thống thông báo đã phê duyệt khóa học thành công. | |

#### Hủy phê duyệt khóa học

|  |  |
| --- | --- |
| Tên Use Case | Hủy phê duyệt khóa học |
| Tác nhân | Quản trị viên |
| Sự kiện kích hoạt | Quản trị viên ấn vào nút hủy phê duyệt bên cạnh khóa học |
| Luồng sự kiện chính:   1. Quản trị viên vào trang danh sách khóa học 2. Ấn vào nút hủy phê duyệt bên cạnh khóa học 3. Hủy phê duyệt thành công, hệ thống thông báo đã hủy phê duyệt khóa học thành công. | |

#### Xóa khóa học

|  |  |
| --- | --- |
| Tên Use Case | Xóa khóa học |
| Tác nhân | Quản trị viên |
| Sự kiện kích hoạt | Quản trị viên ấn vào nút xóa bên cạnh khóa học |
| Luồng sự kiện chính:   1. Quản trị viên vào trang danh sách khóa học 2. Ấn vào nút xóa bên cạnh khóa học 3. Xóa thành công, hệ thống thông báo đã xóa khóa học thành công. | |
| Luồng sự kiện phụ:   1. Xóa thất bại: Hệ thống thông báo xóa khóa học thất bại. | |

#### Tạo khóa học mới

|  |  |
| --- | --- |
| Tên Use Case | Tạo khóa học mới |
| Tác nhân | Giảng viên |
| Sự kiện kích hoạt | Giảng viên ấn vào nút Thêm mới |
| Luồng sự kiện chính:   1. Giảng viên vào trang tạo khóa học mới 2. Giảng viên nhập thông tin khóa học vào form. 3. Giảng viên ấn nút Thêm mới 4. Thêm thành công: Hệ thống thông báo đã thêm khóa học mới thành công. | |
| Luồng sự kiện phụ:   1. Thêm mới thất bại: Hệ thống thông báo lỗi cho người dùng. | |

#### Sửa thông tin khóa học

|  |  |
| --- | --- |
| Tên Use Case | Sửa thông tin khóa học |
| Tác nhân | Giảng viên |
| Sự kiện kích hoạt | Giảng viên ấn vào nút Cập nhật |
| Luồng sự kiện chính:   1. Giảng viên vào trang danh sách khóa học đã tạo 2. Giảng viên ấn vào khóa học 3. Giảng viên ấn vào nút sửa thông tin khóa học bên menu trái, hệ thống sẽ chuyển hướng sang trang sửa thông tin khóa học. 4. Giảng viên nhập thông tin cần sửa. 5. Giảng viên ấn nút Cập nhật. 6. Cập nhật thành công: Hệ thống thông báo đã cập nhật thông tin khóa học thành công. | |
| Luồng sự kiện phụ:   1. Sửa thông tin khóa học thất bại: Hệ thống thông báo lỗi cho người dùng. | |

#### Xóa khóa học

|  |  |
| --- | --- |
| Tên Use Case | Sửa thông tin khóa học |
| Tác nhân | Giảng viên |
| Sự kiện kích hoạt | Giảng viên ấn vào nút Cập nhật |
| Luồng sự kiện chính:   1. Giảng viên vào trang danh sách khóa học đã tạo 2. Giảng viên ấn vào khóa học 3. Giảng viên ấn vào nút xóa khóa học bên menu trái 4. Xóa thành công: Hệ thống thông báo đã xóa khóa học thành công. | |
| Luồng sự kiện phụ:   1. Xóa khóa học thất bại: Hệ thống thông báo lỗi cho người dùng. | |

#### Xem danh sách sinh viên trong khóa học

|  |  |
| --- | --- |
| Tên Use Case | Xem danh sách sinh viên trong khóa học |
| Tác nhân | Giảng viên |
| Sự kiện kích hoạt | Giảng viên ấn vào nút Xem danh sách sinh viên trong khóa học |
| Luồng sự kiện chính:   1. Giảng viên vào trang danh sách khóa học đã tạo 2. Giảng viên ấn vào khóa học, hệ thống chuyển sang trang thông tin khóa học. 3. Giảng viên ấn vào nút xem danh sách sinh viên bên menu trái 4. Hệ thống hiển thị danh sách sinh viên trong khóa học. | |

#### Tạo buổi học mới:

|  |  |
| --- | --- |
| Tên Use Case | Tạo buổi học mới |
| Tác nhân | Giảng viên |
| Sự kiện kích hoạt | Giảng viên ấn vào nút Thêm mới |
| Luồng sự kiện chính:   1. Giảng viên vào trang danh sách khóa học đã tạo, hệ thống chuyển hướng giảng viên sang trang thông tin khóa học. 2. Giảng viên ấn nút tạo buổi học bên menu trái, hệ thống chuyển hướng giảng viên sang trang tạo buổi học mới. 3. Giảng viên nhập thông tin buổi học mới. 4. Giảng viên ấn nút Thêm mới. 5. Thêm thành công: Hệ thống thông báo đã thêm buổi học mới thành công. | |
| Luồng sự kiện phụ:   1. Thêm mới thất bại: Hệ thống thông báo lỗi cho người dùng. | |

#### Xem danh sách buổi học

|  |  |
| --- | --- |
| Tên Use Case | Xem danh sách buổi học |
| Tác nhân | Giảng viên |
| Sự kiện kích hoạt | Giảng viên ấn vào nút Xem danh sách buổi học |
| Luồng sự kiện chính:   1. Giảng viên vào trang danh sách khóa học đã tạo, hệ thống chuyển hướng giảng viên sang trang thông tin khóa học. 2. Giảng viên ấn nút Xem danh sách buổi học bên menu trái, hệ thống chuyển hướng giảng viên sang trang danh sách buổi học. 3. Hệ thống hiển thị danh sách buổi học. | |

#### Chỉnh sửa thông tin buổi học

|  |  |
| --- | --- |
| Tên Use Case | Sửa thông tin buổi học |
| Tác nhân | Giảng viên |
| Sự kiện kích hoạt | Giảng viên ấn vào nút sửa thông tin buổi học. |
| Luồng sự kiện chính:   1. Giảng viên vào trang danh sách khóa học đã tạo, hệ thống chuyển hướng giảng viên sang trang thông tin khóa học. 2. Giảng viên ấn nút Danh sách buổi học bên menu trái, hệ thống chuyển hướng giảng viên sang trang danh sách buổi học. 3. Giảng viên ấn vào buổi học cần sửa thông tin, hệ thống chuyển hướng sang trang thông tin buổi học. 4. Giảng viên ấn nút chỉnh sửa thông tin buổi học bên menu trái. 5. Giảng viên nhập thông tin buổi học mới. 6. Giảng viên ấn nút Thêm mới. 7. Thêm thành công: Hệ thống thông báo đã thêm buổi học mới thành công. | |
| Luồng sự kiện phụ:   1. Thêm mới thất bại: Hệ thống thông báo lỗi cho người dùng. | |

#### Xóa buổi học

|  |  |
| --- | --- |
| Tên Use Case | Xóa buổi học |
| Tác nhân | Giảng viên |
| Sự kiện kích hoạt | Giảng viên ấn vào nút xóa buổi học. |
| Luồng sự kiện chính:   1. Giảng viên vào trang danh sách khóa học đã tạo, hệ thống chuyển hướng giảng viên sang trang thông tin khóa học. 2. Giảng viên ấn nút Danh sách buổi học bên menu trái, hệ thống chuyển hướng giảng viên sang trang danh sách buổi học. 3. Giảng viên ấn vào buổi học cần xóa, hệ thống chuyển sang trang thông tin buổi học. 4. Giảng viên ấn vào nút xóa buổi học bên menu trái. 5. Xóa thành công: Hệ thống thông báo đã thêm buổi học mới thành công. | |
| Luồng sự kiện phụ:   1. Xóa thất bại: Hệ thống thông báo lỗi cho người dùng. | |

#### Tạo quiz mới

|  |  |
| --- | --- |
| Tên Use Case | Tạo quiz mới |
| Tác nhân | Giảng viên |
| Sự kiện kích hoạt | Giảng viên ấn vào nút Thêm mới. |
| Luồng sự kiện chính:   1. Giảng viên vào trang danh sách khóa học đã tạo, hệ thống chuyển hướng giảng viên sang trang thông tin khóa học. 2. Giảng viên ấn nút Danh sách buổi học bên menu trái, hệ thống chuyển hướng giảng viên sang trang danh sách buổi học. 3. Giảng viên ấn vào buổi học cần thêm quiz, hệ thống chuyển sang trang thông tin buổi học. 4. Giảng viên ấn vào nút thêm quiz bên menu trái, hệ thống chuyển hướng sang trang tạo quiz. 5. Giảng viên nhập thông tin quiz. 6. Giảng viên ấn nút Thêm mới. 7. Thêm mới thành công: Hệ thống báo thêm quiz mới thành công. | |
| Luồng sự kiện phụ:   1. Thêm mới thất bại: Hệ thống thông báo lỗi cho người dùng. | |

#### Gửi nhận tin nhắn chat realtime

|  |  |
| --- | --- |
| Tên Use Case | Gửi nhận tin nhắn |
| Tác nhân | Giảng viên, sinh viên |
| Sự kiện kích hoạt | Giảng viên, sinh viên ấn vào nút Send trong khung chat. |
| Luồng sự kiện chính:   1. Giảng viên, sinh viên vào lớp học. 2. Giảng viên, sinh viên nhập tin nhắn vào ô chat 3. Giảng viên, sinh viên ấn nút send. 4. Hiển thị tin nhắn trong khung chat. | |

#### Trình chiếu slide

|  |  |
| --- | --- |
| Tên Use Case | Trình chiếu slide |
| Tác nhân | Giảng viên |
| Sự kiện kích hoạt | Giảng viên ấn vào nút Submit |
| Luồng sự kiện chính:   1. Giảng viên vào lớp học. 2. Giảng viên ấn vào menu thêm slide, khung thêm slide hiện lên. 3. Giảng viên chọn file slide. 4. Giảng viên ấn nút Submit. 5. Slide hiển thị trong khung làm việc của Giảng viên và sinh viên. 6. Giảng viên điều khiển slide, slide của sinh viên cũng thay đổi theo. | |

#### Truyền thông video, audio

|  |  |
| --- | --- |
| Tên Use Case | Truyền thông video, audio |
| Tác nhân | Giảng viên, sinh viên |
| Sự kiện kích hoạt | Giảng viên, sinh viên vào lớp học |
| Luồng sự kiện chính:   1. Giảng viên, sinh viên vào lớp học. 2. Hệ thống khởi tạo session và token cho từng người dùng. 3. Video được render lên màn hình. | |

#### Kiểm tra

|  |  |
| --- | --- |
| Tên Use Case | Kiểm tra |
| Tác nhân | Giảng viên |
| Sự kiện kích hoạt | Giảng viên ấn Submit |
| Luồng sự kiện chính:   1. Giảng viên, sinh viên vào lớp học. 2. Giảng viên ấn vào nút thêm quiz trên thanh menu, khung set thời gian kiểm tra hiện lên. 3. Giảng viên nhập thời gian kiểm tra vào form. 4. Giảng viên ấn submit. 5. Câu hỏi được hiển thị trong màn hình làm việc của giảng viên và sinh viên. 6. Sinh viên chọn câu trả lời và ấn submit. 7. Khi hết thời gian, kết quả kiểm tra được hiển thị trên màn hình của giảng viên. | |

#### Điểm danh sinh viên

|  |  |
| --- | --- |
| Tên Use Case | Điểm danh sinh viên |
| Tác nhân | Sinh viên |
| Sự kiện kích hoạt | Sinh viên vào lớp học |
| Luồng sự kiện chính:   1. Sinh viên vào lớp học. 2. Hệ thống ghi nhận thời gian vào lớp của sinh viên. 3. Sinh viên rời khỏi lớp, đóng trình duyệt hoặc mất mạng, hệ thống ghi nhận thời gian sinh viên rời khỏi lớp. | |

## Thiết kế module chức năng

Hệ thống lớp học ảo gồm hai phần:

* Lớp học ảo: Cung cấp các công cụ dạy và học tập trực tuyến, hỗ trợ tương tác trực tiếp giữa giảng viên và sinh viên.
* Hệ thống quản lý: gồm các chức năng quản lý người dùng, quản lý khóa học, buổi học.

### Module lớp học ảo

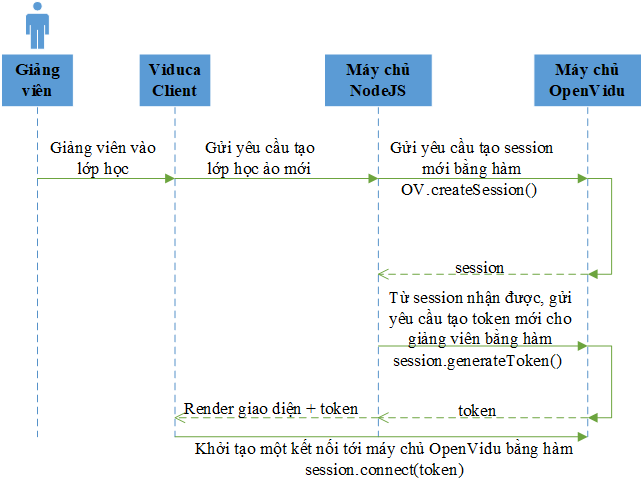
Module lớp học ảo gồm tập hợp các chức năng truyền thông video, audio thời gian thực, trình chiếu slide, chat văn bản, thêm quiz. Các chức năng này đều hoạt động dựa trên nền tảng OpenVidu, tạo ra sự tương tác thời gian thực giữa người dùng trong lớp học.

#### Chức năng truyền thông video, audio

Chức năng truyền thông video, audio trực tuyến thời gian thực được thực thi ngay khi lớp học ảo được khởi tạo.

*Giảng viên khởi tạo lớp học ảo*

Bên dưới là biểu đồ tuần tự mô tả cách một lớp học ảo được khởi tạo.



Hình 2.2: Biểu đồ tuần tự mô tả cách khởi tạo lớp học ảo

Khi giảng viên ấn nút vào lớp học, ở phía client sẽ gửi một yêu cầu tạo lớp học ảo đến máy chủ NodeJS, tại máy chủ NodeJS một đối tượng OpenVidu được tạo ra bằng câu lệnh:

var OV = new OpenVidu(OPENVIDU\_URL, OPENVIDU\_SECRET);

Trong đó, OPENVIDU\_URL là địa chỉ của máy chủ OpenVidu, OPENVIDU\_SECRET là khóa bí mật cho phép kết nối đến máy chủ OpenVidu. Sau khi đối tượng OV được khởi tạo, máy chủ NodeJS gửi yêu cầu tạo một session mới đến máy chủ OpenVidu bằng cách gọi hàm createSession():

OV.createSession().then(session => {

}).catch(error => {

});

Máy chủ OpenVidu sau khi nhận được yêu cầu, sẽ tạo ra một session mới và gửi thông tin session này cho máy chủ NodeJS. Máy chủ NodeJS nhận được session từ máy chủ OpenVidu sẽ tiếp tục gửi yêu cầu tạo token cho giảng viên bằng cách gọi hàm:

session.generateToken(tokenOptions).then(token => {

}).catch(error => {

});

Máy chủ OpenVidu sau khi nhận được yêu cầu tạo token, sẽ tạo ra một token mới và gửi lại cho máy chủ NodeJS. Khi máy chủ NodeJS nhận được token sẽ tiến hành render giao diện và trả về token cho client. Tóm lại ở phía máy chủ NodeJS ta thực hiện như sau:

OV.createSession().then(session => {

session.generateToken(tokenOptions).then(token => {

}).catch(error => {

});

}).catch(error => {

});

Ở phía client lúc này đã có được token, sẽ tiến hành kết nối tới máy chủ OpenVidu bằng cách gọi hàm:

session.connect(token).then(() => {

}).catch(error => {

});

Trong đó, session là biến được khởi tạo bằng câu lệnh:

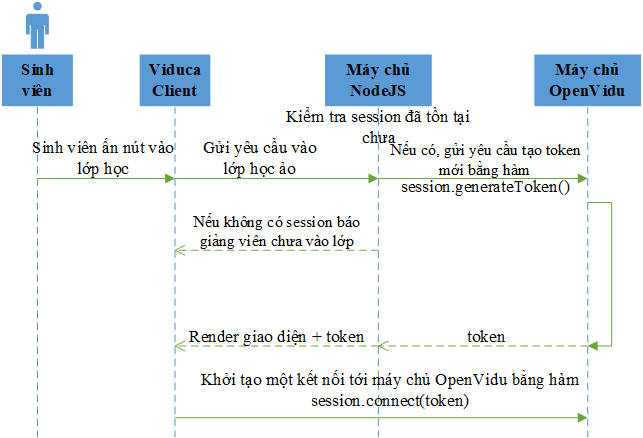
const OV = new OpenVidu();

session = OV.initSession();

Sau lời gọi hàm trên, ở phía client của giảng viên sẽ xin phép truy cập vào camera và microphone để lấy dòng video, audio, nếu người dùng cho phép thì local video, audio sẽ được render, đồng thời có một kết nối đến máy chủ OpenVidu, lúc này, nếu có một sinh viên mới vào lớp, tức session đã có thêm một stream thì video, audio của sinh viên này sẽ xuất hiện ở phía client của giảng viên.

*Sinh viên join vào lớp học ảo*

Dưới đây là biểu đồ tuần tự mô tả cách một sinh viên join vào một lớp học ảo do giảng viên tạo ra:



Hình 2.3: Biểu đồ tuần tự mô tả cách một sinh viên join vào lớp học ảo

Khi một sinh viên ấn nút vào lớp học, ở phía client của sinh viên sẽ gửi đến máy chủ NodeJS một yêu cầu join vào lớp, máy chủ NodeJS sẽ kiểm tra đã tồn tại session cho lớp học ảo này hay chưa, nếu chưa sẽ báo cho sinh viên là giảng viên chưa vào lớp, nếu đã tồn tại session, máy chủ NodeJS sẽ lấy ra session này và gửi một yêu cầu tạo token mới đến máy chủ OpenVidu bằng cách gọi hàm:

mySession.generateToken()

.then(token => {

});

Sau khi token được tạo ra, máy chủ OpenVidu sẽ gửi lại cho máy chủ NodeJS, máy chủ NodeJS nhận được token thì tiến hành render giao diện lớp học ảo kèm theo token trả về cho phía client. Ở phía client của sinh viên, sau khi nhận được token sẽ tiến hành tạo một kết nối tới máy chủ OpenVidu bằng cách gọi hàm:

session.connect(token).then(() => {

}).catch(error => {

});

Trong đó, session là biến được khởi tạo bằng câu lệnh:

const OV = new OpenVidu();

session = OV.initSession();

Sau lời gọi hàm trên, ở phía client của sinh viên sẽ xuất hiện local video, audio, đồng thời có một kết nối tới máy chủ OpenVidu. Lúc này, để render video, audio của giảng viên và các sinh viên khác trong lớp học thì client của sinh viên phải gọi hàm:

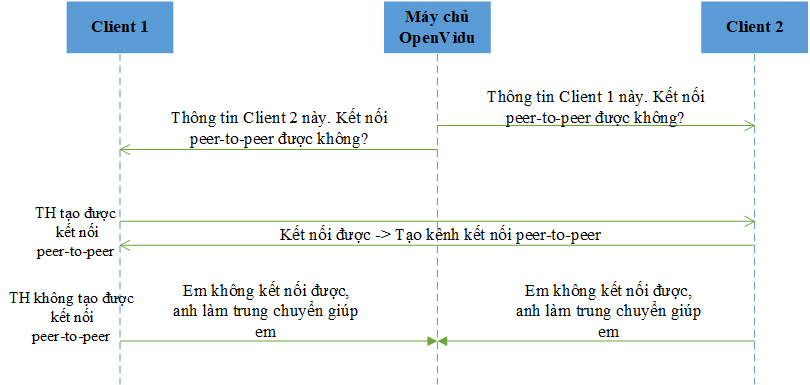
session.on('streamCreated', (event) => {

session.subscribe(event.stream, "remoteVideo");

});

Hàm trên sẽ trả về event chứa các stream của giảng viên và sinh viên khác, bên trong hàm là dòng code render các stream này lên màn hình.

Đến đây thì ta đã có một lớp học ảo với các stream kết nối đến cùng một session được quản lý trên máy chủ OpenVidu, phía client của giảng viên và client của sinh viên sẽ được máy chủ OpenVidu hướng dẫn để tạo ra một kết nối peer-to-peer. Trong trường hợp không thể thiết lập được một kết nối peer-to-peer thì máy chủ OpenVidu sẽ đứng ra làm máy chủ trung chuyển các gói tin giữa các client.

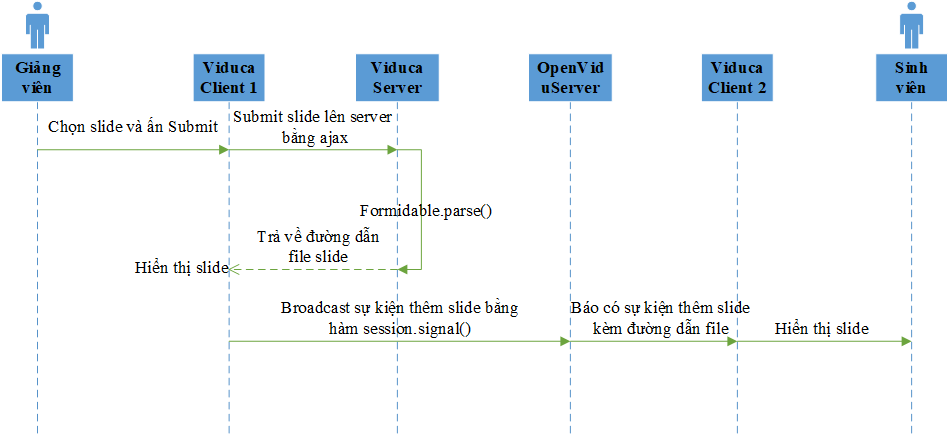


Hình 2.4: Biểu đồ tuần tự mô tả cách kết nối giữa các client

#### Chức năng trình chiếu slide

Chức năng trình chiếu slide giúp giảng viên dễ dàng trình bày bài giảng với sinh viên, giảng viên chỉ cần thêm file slide, ngay lập tức slide cũng sẽ hiển thị trên màn hình chính của sinh viên, khi giảng viên điều hướng slide thì slide tại máy của sinh viên cũng ngay lập tức thay đổi, việc trình chiếu thời gian thực này giúp tăng khả năng tương tác giữa giảng viên và sinh viên.

Trong chức năng trình chiếu slide thời gian thực gồm hai phần là thêm slide và điều khiển slide. Hình vẽ bên dưới là biểu đồ tuần tự mô tả phần thêm slide.



Hình 2.5: Mô tả chức năng thêm slide

Ở phần thêm slide, giảng viên sau khi chọn slide và ấn submit, slide sẽ được gửi đến server NodeJS bằng ajax. Ở phía server NodeJS, sau khi nhận được request ajax sẽ tiến hành sử dụng module formidable để parse request thành file có thể lưu trữ được trên server. Khi server lưu trữ xong slide nó sẽ gửi lại cho phía client của giảng viên đường dẫn đến file slide. Ở phía client của giảng viên, khi nhận được đường dẫn slide thì tiến hành render slide, đồng thời phát đi một sự kiện thêm slide đến tất cả sinh viên trong lớp bằng cách gọi:

session.signal({

data: “Đường dẫn slide”,

to: [],

type: 'send slide event'

});

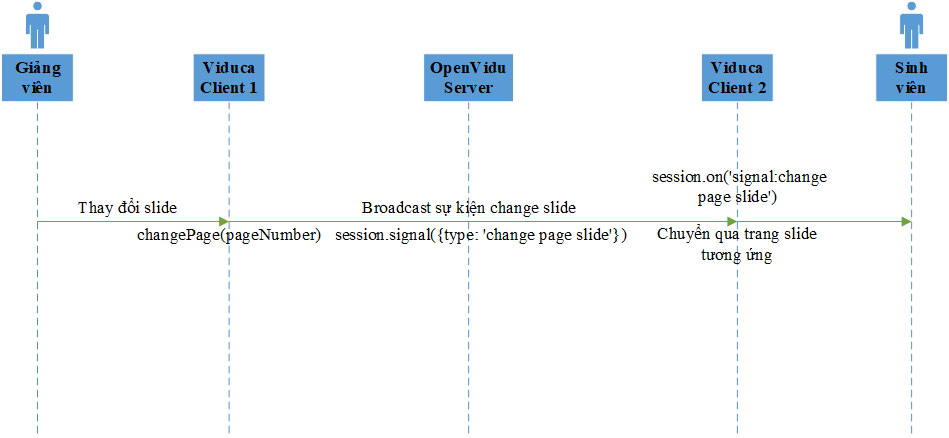
Sự kiện thêm slide này có trường data chứa đường dẫn đến file slide. Sau khi phía client của sinh viên nhận được sự kiện, nó sẽ tiến hành lấy đường dẫn đến file slide trong trường data để nạp và render slide lên giao diện:

session.on('signal:send slide event', event => {

showSlide(event.data);

});

Ở trên mới chỉ là phần thêm slide và hiển thị slide ở client, để có thể trình chiếu slide theo thời gian thực, tức giảng viên điều khiển đến slide nào thì slide ở phía client của sinh viên cũng thay đổi theo thì hệ thống cần triển khai một số kịch bản đặc biệt.



Hình 2.6: Biểu đồ tuần tự mô tả chức năng trình chiếu slide thời gian thực

Khi giảng viên thay đổi slide bằng cách ấn nút next, previous hoặc thay đổi số trang trong ô input thì phía client của giảng viên sẽ thay đổi slide ở local, đồng thời gửi một sự kiện “change slide event” đến tất cả client của sinh viên:

session.signal({

data: số trang slide,

to: [],

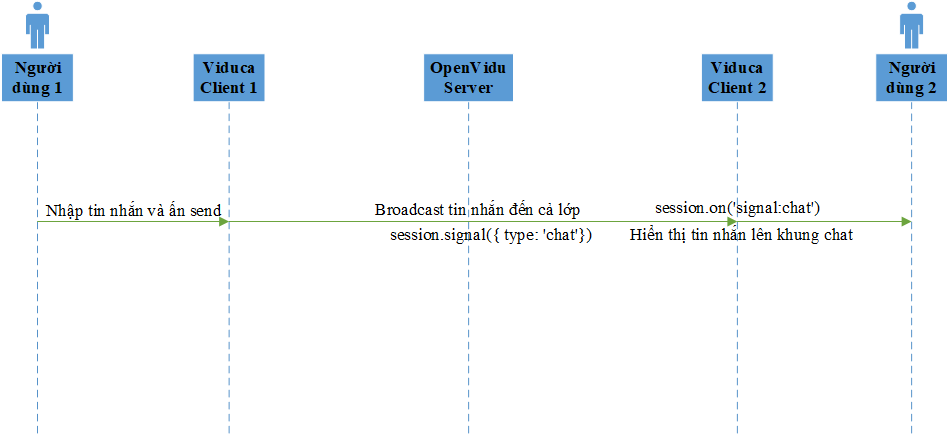
type: 'change page slide'

});

Ở phía client của sinh viên, sau khi nhận được sự kiện “change slide event” sẽ tiến hành thay đổi trang slide tương ứng.

#### Chức năng chat văn bản

Chức năng chat văn bản giúp sinh viên dễ dàng đặt câu hỏi với giảng viên và các bạn trong lớp, chức năng này cũng hoạt động trong thời gian thực, tức sinh viên gửi tin nhắn thì ngay lập tức tất cả thành viên trong lớp sẽ nhận được. Về cơ chế thì tính năng chat văn bản thời gian thực vẫn dựa trên nền tảng OpenVidu.



Hình 2.7: Biểu đồ tuần tự mô tả chức năng gửi tin nhắn văn bản

Khi một người dùng trong lớp học muốn gửi tin nhắn văn bản đến các người dùng khác trong lớp học bằng cách nhập tin nhắn và ấn nút send thì phía client của người dùng đó sẽ broadcast một sự kiện có type là chat đến toàn bộ lớp học:

session.signal({

data: messageData,

to: [],

type: 'chat'

});

Các client khác sau khi lắng nghe được sự kiện ‘chat’ sẽ ngay lập tức hiển thị tin nhắn lên khung chat:

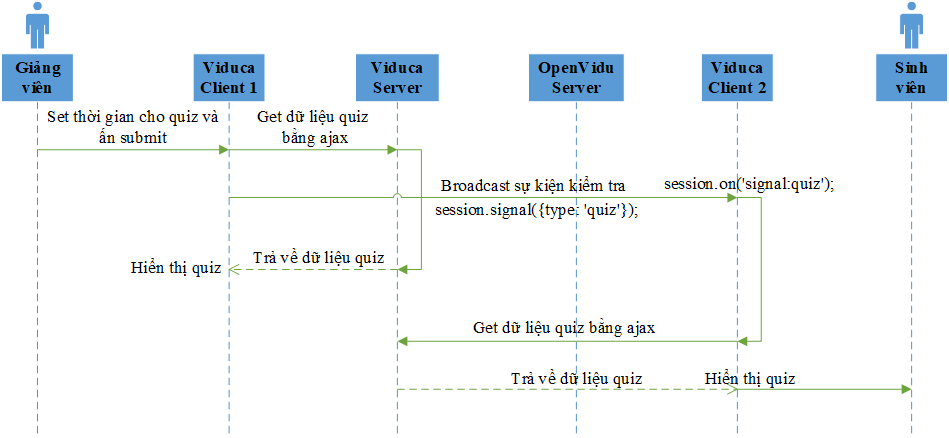
session.on('signal:chat', (event) => {

// *Hiển thị tin nhắn lên khung chat*

});

#### Chức năng kiểm tra

Chức năng kiểm tra gồm hai bước: Giảng viên set thời gian cho bài quiz và sinh viên làm bài. Hình bên dưới là biểu đồ tuần mô tả bước giảng viên set thời gian cho bài kiểm tra.



Hình 2.8: Biểu đồ tuần tự mô tả chức năng bật bài kiểm tra

Khi giảng viên set thời gian cho bài quiz và ấn submit, ở client của giảng viên sẽ gửi một get request ajax đến máy chủ NodeJS:

$.get(urlQuiz, function(data, status){

// Hiển thị quiz

});

Hàm trên trả về dữ liệu quiz và hiển thị ở local, đồng thời lúc này cũng đã broadcast sự kiện kiểm tra đến sinh viên trong lớp:

session.signal({

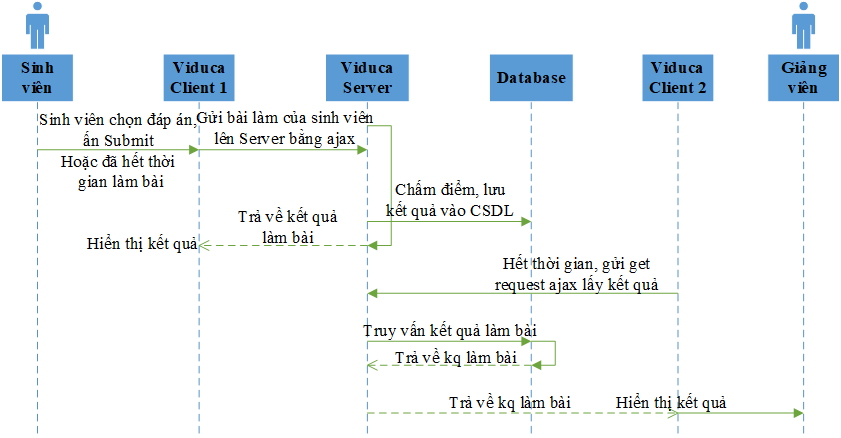
data: time,

to: [],

type: 'quiz'

});

Khi phía client của sinh viên nhận được sự kiện kiểm tra sẽ gửi một get request ajax đến máy chủ NodeJS để lấy dữ liệu quiz, sau khi lấy được dữ liệu quiz thì sẽ hiển thị lên màn hình đồng thời set thời gian làm bài. Sau đó sinh viên sẽ làm bài và ấn submit. Bên dưới là biểu đồ tuần tự mô tả chức năng kiểm tra.



Hình 2.9: Biểu đồ tuần tự mô tả chức năng kiểm tra

Khi sinh viên làm xong bài và ấn submit hoặc đã hết thời gian làm bài thì phía client của sinh viên sẽ gửi bài làm đến máy chủ Viduca bằng ajax:

$.ajax({

type: 'POST',

data: {

question: key,

answer: answersOfStudent[key]

},

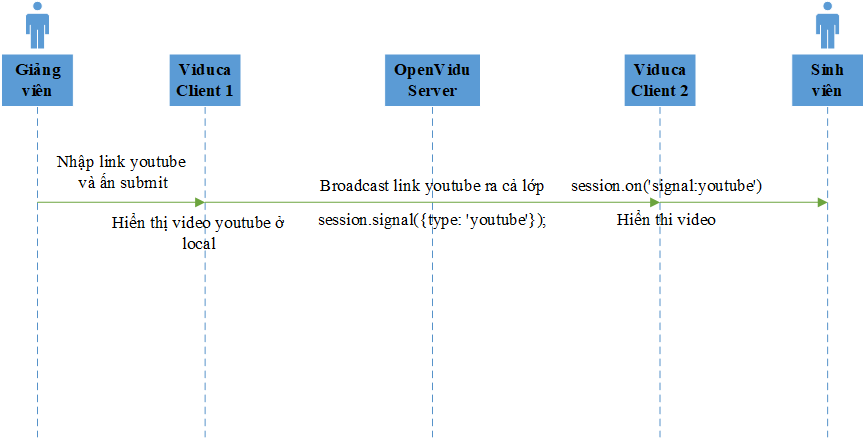
url: '/student/save-result-quiz',

});

Sau khi máy chủ Viduca nhận được bài làm của sinh viên sẽ tiến hành chấm điểm và lưu vào cơ sở dữ liệu, đồng thời trả về kết quả cho sinh viên. Khi hết thời gian làm bài, ở phía client của giảng viên sẽ gửi một get request ajax đến máy chủ Viduca để lấy kết quả và hiển thị lên màn hình của giảng viên. Giảng viên cũng có thể xem lại kết quả sau buổi học bằng cách vào menu “kết quả kiểm tra” của buổi học.

#### Chức năng chia sẻ video youtube

Chức năng chia sẻ video youtube giúp giảng viên chia sẻ những video giảng dạy cho sinh viên xem trực tiếp trong lớp học ảo. Bên dưới là biểu đồ tuần tự mô tả chức năng chia sẻ video youtube.



Hình 2.10: Biểu đồ tuần tự mô tả chức năng chia sẻ video youtube

Giảng viên nhập link youtbe và ấn submit, video sẽ ngay lập tức được hiển thị ở local, đồng thời client của giảng viên sẽ broadcast một sự kiện ‘youtube’ đến toàn bộ sinh viên trong lớp:

session.signal({

data: linkYoutube,

to: [],

type: 'youtube'

});

Sau khi client của sinh viên nhận được sự kiện sẽ tiến hành lấy link youtube và hiển thị video youtube lên màn hình.

session.on('signal:youtube', event => {

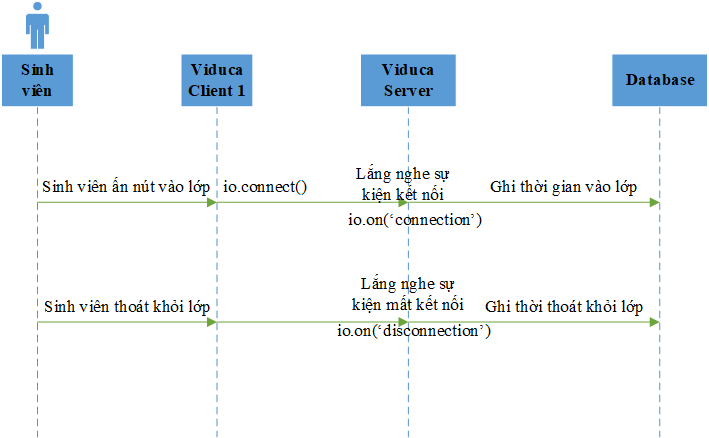
var linkYoutube = event.data;

//Hiển thị video youtube

});

#### Chức năng điểm danh tự động

Chức năng điểm danh tự động giúp giảng viên đánh giá được mức độ chuyên cần của từng sinh viên, chức năng này có thể theo dõi thời gian sinh viên ra vào lớp. Bên dưới là biểu đồ tuần tự mô tả chức năng điểm danh tự động:



Hình 2.11: Biểu đồ tuần tự mô tả chức năng điểm danh tự động

Khi sinh viên ấn nút “vào lớp học” thì ở client của sinh viên sẽ gửi đến máy chủ Viduca một sự kiện kết nối bằng hàm io.connect(). Ở máy chủ Viduca có hàm io.on(‘connection’) để lắng nghe xem có client nào vào lớp học hay không, ngay khi nhận được sự kiện connect từ phía client của sinh viên, máy chủ Viduca sẽ ghi lại thời gian sinh viên vào lớp vào trong cơ sở dữ liệu. Sau đó, nếu sinh viên thoát khỏi lớp học bằng cách đóng trình duyệt hoặc mất kết nối mạng, hay nguyên nhân nào đó làm phát sinh sự kiện disconnect thì máy chủ viduca sẽ ghi lại thời gian sinh viên thoát khỏi lớp học vào trong cơ sở dữ liệu. Giảng viên có thể xem lại các thông tin điểm danh này ở trong menu “kết quả điểm danh của buổi học”.

### Module quản lý

Module quản lý gồm các chức năng hỗ trợ cho việc vận hành hệ thống như quản lý người dùng, quản lý khóa học, buổi học. Mỗi người dùng sẽ có các quyền cụ thể trong việc sử dụng các chức năng này.

#### Module quản lý truy cập

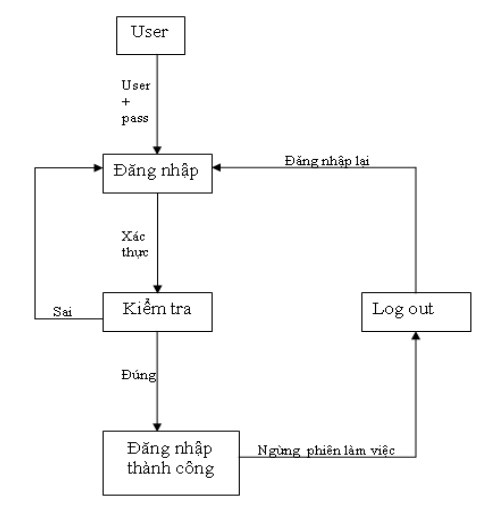
Module quản lý truy cập được mô tả ở sơ đồ bên dưới:



Hình 2.12: Module quản lý truy cập

Module quản lý truy cập gồm tập hợp các chức năng cho phép quản lý việc truy cập hệ thống của người dùng thông qua các cơ chế đăng nhập, đăng xuất với các quyền hạn cho phép, giúp người dùng dễ dàng trong việc khai thác thông tin và tương tác với hệ thống.

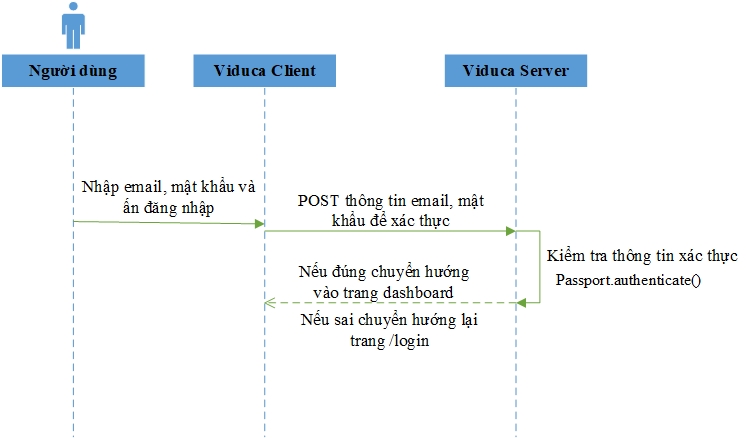
Người dùng sẽ sử dụng địa chỉ email và mật khẩu để truy cập hệ thống, hệ thống sẽ tạo ra cookie và session cho phép người dùng sử dụng hệ thống trong một khoảng thời gian nhất định. Mô hình tổng quan về quản lý truy cập hệ thống lớp học ảo Viduca được mô tả như hình bên dưới:



Hình 2.13: Mô hình tổng quan quản lý truy cập hệ thống

Khi người dùng truy cập vào hệ thống lớp học ảo Viduca, nếu chưa đăng nhập thì sẽ xuất hiện màn hình đăng nhập yêu cầu nhập địa chỉ email và mật khẩu.

Chức năng xác thực sẽ kiểm tra địa chỉ email người dùng nhập có trong hệ thống hay không, nếu có thì kiểm tra xem mật khẩu có đúng không, nếu thông tin email và mật khẩu đúng hệ thống sẽ tạo cookie và session cho người dùng sau đó cho phép người dùng sử dụng các chức năng tương ứng với quyền hạn có trong hệ thống. Nếu thông tin đăng nhập của người dùng sai, hệ thống sẽ trả lại màn hình đăng nhập và yêu cầu người dùng nhập lại thông tin đăng nhập.



Hình 2.14: Biểu đồ tuần tự quá trình xác thực bằng email và mật khẩu

Chức năng đăng xuất giúp người dùng thoát khỏi hệ thống khi muốn ngừng phiên làm việc. Khi người dùng sử dụng chức năng đăng xuất, hệ thống sẽ lập tức xóa session của người dùng và chuyển hướng người dùng về màn hình đăng nhập, lúc này, để sử dụng hệ thống người dùng bắt buộc phải đăng nhập lại.

Module quản lý truy cập còn có chức năng ghi lại thời gian tham gia lớp học của sinh viên, giúp giảng viên có những đánh giá khách quan về mức độ chuyên cần của sinh viên.

Chức năng đổi mật khẩu giúp người dùng thay đổi mật khẩu truy cập hệ thống, ngoài ra người dùng cũng có thể thay đổi các thông tin cá nhân khác.

Module quản lý truy cập được triển khai dựa trên module Passport.js và session trong express.

Module Passport.js sẽ hỗ trợ khả authentication của hệ thống, nó được thiết kế là một middleware hết sức linh hoạt cho khả năng tùy biến cao với rất nhiều kịch bản authenticate như sử dụng tài khoản Google, facebook, hoặc email, password trong Database. Đối với hệ thống lớp học ảo Viduca, chỉ sử dụng duy nhất kịch bản xác thực là dùng email và password trong database.

Để sử dụng module Passport.js ta phải require nó và thêm hai middleware của nó là passport.initialize() và passport.session();

#### Module quản lý giảng viên

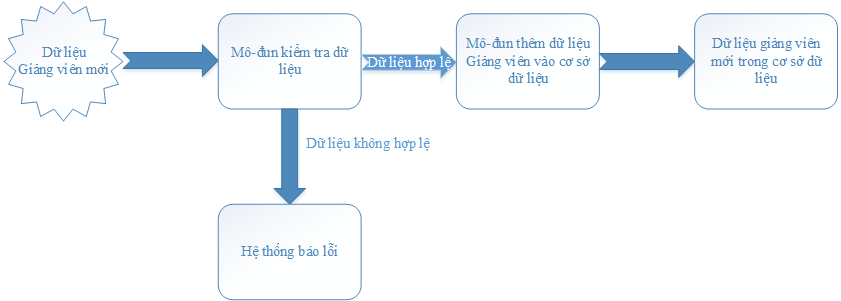
Module quản lý giảng viên được mô tả ở sơ đồ dưới đây:



Hình 2.15: Module quản lý giảng viên

Đây là module chức năng dành cho người quản trị viên. Module này là tập hợp các chức năng xem danh sách giảng viên, xem chi tiết thông tin giảng viên, chỉnh sửa thông tin giảng viên, xóa giảng viên khỏi hệ thống, thêm giảng viên mới.

Chức năng thêm mới giảng viên gồm hai chức năng nhỏ là thêm mới bằng file .csv và thêm mới bằng form. Với chức năng thêm mới giảng viên bằng file .csv, quản trị viên chỉ cần một file lưu trữ địa chỉ email của các giảng viên, sau đó submit file này lên server, server sẽ đọc các địa chỉ email và thêm vào cơ sở dữ liệu, mật khẩu đăng nhập mặc định của giảng viên là địa chỉ email. Với cách thêm mới bằng file, quản trị viên sẽ thêm mới được nhiều giảng viên cùng một lúc. Quản trị viên cũng có thể thêm mới giảng viên bằng cách nhập địa chỉ email của giảng viên vào form và ấn thêm mới, mật khẩu đăng nhập mặc định của giảng viên cũng là email, với cách này, quản trị viên chỉ có thể thêm từng giảng viên, do đó chỉ nên sử dụng khi cần thêm một số lượng ít giảng viên mới.



Hình 2.16: Mô tả chức năng thêm giảng viên mới

Chức năng xem danh sách giảng viên cho phép quản trị viên xem được tất cả các giảng viên đang có trong hệ thống, và trạng thái tài khoản của họ. Bên cạnh mỗi dòng chứa thông tin giảng viên có hai tác vụ quản lý là chỉnh sửa thông tin và chức năng xóa giảng viên khỏi hệ thống. Khi quản trị viên sử dụng tác vụ chỉnh sửa thông tin giảng viên, hệ thống sẽ tìm thông tin giảng viên theo id có trong cơ sở dữ liệu và hiển thị thông tin hiện tại trên trang sửa thông tin, giảng viên thay đổi thông tin và ấn nút cập nhật, thông tin giảng viên sẽ được server cập nhật lại trong cơ sở dữ liêu. Khi quản trị viên sử dụng chức năng xóa giảng viên, ngay lập tức server sẽ xóa giảng viên khỏi hệ thống.

#### Module quản lý sinh viên

Module quản lý sinh viên được mô tả ở sơ đồ dưới đây:



Hình 2.17: Module quản lý sinh viên

Tương tự như module chức năng quản lý giảng viên, module quản lý sinh viên cũng dành cho quản trị viên với các chức năng tương tự. Module này là tập hợp các chức năng xem danh sách sinh viên, xem chi tiết thông tin sinh viên, chỉnh sửa thông tin sinh viên, xóa sinh viên khỏi hệ thống, thêm sinh viên mới.

Chức năng thêm mới sinh viên gồm hai chức năng nhỏ là thêm mới bằng file .csv và thêm mới bằng form. Với chức năng thêm mới sinh viên bằng file .csv, quản trị viên chỉ cần một file lưu trữ địa chỉ email của các sinh viên, sau đó submit file này lên server, server sẽ đọc các địa chỉ email và thêm vào cơ sở dữ liệu, mật khẩu đăng nhập mặc định của sinh viên là địa chỉ email. Với cách thêm mới bằng file, quản trị viên sẽ thêm mới được nhiều sinh viên cùng một lúc. Quản trị viên cũng có thể thêm mới sinh viên bằng cách nhập địa chỉ email của sinh viên vào form và ấn thêm mới, mật khẩu đăng nhập mặc định của sinh viên cũng là email, với cách này, quản trị viên chỉ có thể thêm từng sinh viên, do đó chỉ nên sử dụng khi cần thêm một số lượng ít sinh viên mới.



Hình 2.18: Mô tả chức năng thêm sinh viên mới

Chức năng xem danh sách sinh viên cho phép quản trị viên xem được tất cả các sinh viên đang có trong hệ thống, và trạng thái tài khoản của họ. Bên cạnh mỗi dòng chứa thông tin sinh viên có hai tác vụ quản lý là chỉnh sửa thông tin và chức năng xóa sinh viên khỏi hệ thống. Khi quản trị viên sử dụng tác vụ chỉnh sửa thông tin sinh viên, hệ thống sẽ tìm thông tin sinh viên theo id có trong cơ sở dữ liệu và hiển thị thông tin hiện tại trên trang sửa thông tin, sinh viên thay đổi thông tin và ấn nút cập nhật, thông tin sinh viên sẽ được server cập nhật lại trong cơ sở dữ liêu. Khi quản trị viên sử dụng chức năng xóa sinh viên, ngay lập tức server sẽ xóa sinh viên khỏi hệ thống.

#### Module phê duyệt khóa học

Module phê duyệt khóa học được mô tả ở sơ đồ bên dưới:

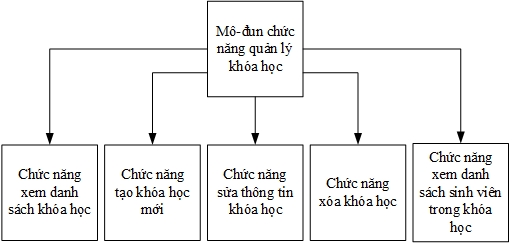


Hình 2.19: Module phê duyệt khóa học

Module chức năng này dành cho quản trị viên với các chức năng xem danh sách các khóa học đã tạo, trạng thái của các khóa học, duyệt hoặc hủy phê duyệt khóa học và chức năng xóa khóa học khỏi hệ thống. Khi giảng viên tạo mới khóa học, khóa học này sẽ không xuất hiện trong danh sách các khóa học, sinh viên cũng sẽ không thể đăng ký được vào các khóa học này. Quản trị viên sẽ xem xét và phê duyệt khóa học mới do giảng viên tạo ra, sau khi được phê duyệt, khóa học sẽ xuất hiện trong danh sách các khóa học, lúc này sinh viên có thể đăng ký tham gia khóa học.

#### Module quản lý khóa học

Module quản lý khóa học được mô tả ở sơ đồ bên dưới:



Hình 2.20: Module chức năng quản lý khóa học

Module chức năng này dành cho giảng viên với các chức năng tạo khóa học mới, sửa thông tin khóa học, xóa khóa học, xem danh sách sinh viên trong khóa học.

Chức năng tạo mới khóa học cho phép giảng viên thêm mới một khóa học với các trường thông tin cơ bản như tên khóa học, mô tả khóa học, ảnh đại diện cho khóa học (nếu giảng viên không chọn ảnh đại diện thì ảnh đại diện sẽ là ảnh mặc định), mật khẩu của khóa học (mật khẩu này được sử dụng trong trường hợp sinh viên muốn đăng ký tham gia khóa học). Sau khi tạo mới khóa học, giảng viên phải đợi quản trị viên xem xét phê duyệt khóa học, sau khi khóa học được phê duyệt, giảng viên có thể tạo các buổi học với các mốc thời gian cụ thể để sinh viên biết được lịch trình của khóa học.



Hình 2.21: Mô tả chức năng tạo khóa học mới

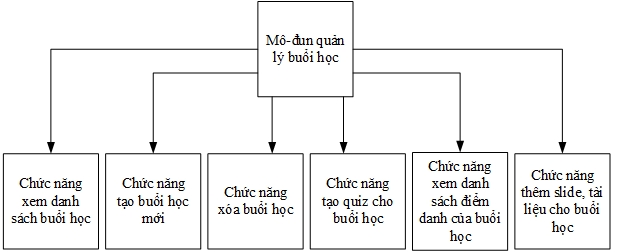
Chức năng sửa thông tin khóa học giúp giảng viên dễ dàng chỉnh sửa thông tin của khóa học đã tạo. Khi giảng viên sử dụng chức năng này, server sẽ truy vấn thông tin của khóa học trong cơ sở dữ liệu theo id và hiển thị trong trang chỉnh sửa, giảng viên chỉnh sửa thông tin xong ấn nút cập nhật, server sẽ cập nhật lại thông tin buổi học trong cơ sở dữ liệu.

Chức năng xem danh sách sinh viên trong khóa học cho phép giảng viên thấy được danh sách những sinh viên đăng ký tham gia khóa học, có được các thông tin của sinh viên để tiện liên lạc khi cần thiết.

Chức năng xóa khóa học giúp giảng viên xóa khóa học đã tạo khi khóa học không còn cần thiết nữa, sau khi giảng viên ấn nút xóa khóa học này, server sẽ ngay lập tức xóa khóa học khỏi hệ thống.

#### Module quản lý buổi học

Module quản lý buổi học được mô tả ở sơ đồ bên dưới:



Hình 2.22: Module quản lý buổi học

Cũng tương tự như module quản lý khóa học, module chức năng này cũng dành cho giảng viên với các chức năng tạo buổi học mới, sửa thông tin buổi học, xóa buổi học, thêm slide, tài liệu cho buổi học, tạo quiz cho buổi học, xem danh sách điểm danh của buổi học.

Chức năng tạo mới buổi học cho phép giảng viên thêm mới một buổi học với các trường thông tin cơ bản như tên buổi học, mô tả buổi học. Sau khi tạo mới buổi học, giảng viên có thể thêm slide, tài liệu, tạo quiz cho buổi học. Chức năng sửa thông tin buổi học giúp giảng viên dễ dàng chỉnh sửa thông tin của buổi học đã tạo. Khi giảng viên sử dụng chức năng này, server sẽ truy vấn thông tin của buổi học trong cơ sở dữ liệu theo id và hiển thị trong trang chỉnh sửa, giảng viên chỉnh sửa thông tin xong ấn nút cập nhật, server sẽ cập nhật lại thông tin buổi học trong cơ sở dữ liệu.

Chức năng tạo quiz là một chức năng đặc biệt trong module này, chức năng này cho phép giảng viên tạo các quiz với các loại câu hỏi khác nhau. Tại trang thêm quiz, giảng viên chọn loại câu hỏi và ấn vào thêm, hệ thống sẽ hiển thị các trường tương ứng với loại câu hỏi mà giảng viên đã chọn. Câu hỏi nhiều lựa chọn sẽ đưa ra cho sinh viên nhiều phương án trả lời, đáp án có thể là một hoặc nhiều phương án giúp sinh viên có cái nhìn tổng quan về bài học, tổng hợp được các kiến thức đã học. Dạng câu hỏi một lựa chọn sẽ có nhiều phương án nhưng sinh viên chỉ được chọn duy nhất một đáp án. Dạng câu hỏi điền đáp án giúp sinh viên tư duy và nhớ được các nội dung chính của bài học. Chức năng này giúp giảng viên đánh giá khả năng hiểu bài của sinh viên và có kết quả đánh giá khách quan cuối khóa học.



Hình 2.23: Mô tả chức năng tạo quiz

Chức năng xem danh sách điểm danh buổi học cho phép giảng viên xem danh sách những sinh viên đã tham gia vào buổi học hôm đó, thời gian sinh viên ở trong lớp học, giúp giảng viên có thể tổng hợp đánh giá mức độ chuyên cần của sinh viên vào cuối khóa học.

Chức năng xóa buổi học giúp giảng viên xóa buổi học đã tạo khi buổi học không còn cần thiết nữa, sau khi giảng viên ấn nút xóa buổi học này, server sẽ ngay lập tức xóa buổi học khỏi hệ thống.

## Thiết kế hệ thống server

Phần này tập trung trình bày thiết kế hệ thống server, cách thức các server tương tác với nhau để đảm bảo các chức năng hoạt động ổn định.

Hệ thống server gồm các server sau: server chạy ứng dụng trả về giao diện ứng dụng, thực hiện các tác vụ xem thêm xóa sửa, media server gồm turn server có nhiệm vụ trung chuyển các gói tin trong trường hợp không thể thiết lập một kết nối trực tiếp giữa các máy. STUN server có nhiệm vụ thiết lập kết nối peer-to-peer giữa các máy.

Để đơn giản hóa việc triển khai hệ thống, viduca cài đặt kurento-media-server, coturn server, openvidu trên cùng một máy chủ và gọi chung là media-server có nhiệm vụ thực hiện chức năng thiết lập kết nối peer-to-peer trong trường hợp khả dụng, nếu không sẽ chuyển qua vai trò là server trung gian trung chuyển các gói tin đa phương tiện giữa các máy tính.

Mô hình và cách thức hoạt động của các server được mô tả ở sơ đồ phía dưới:

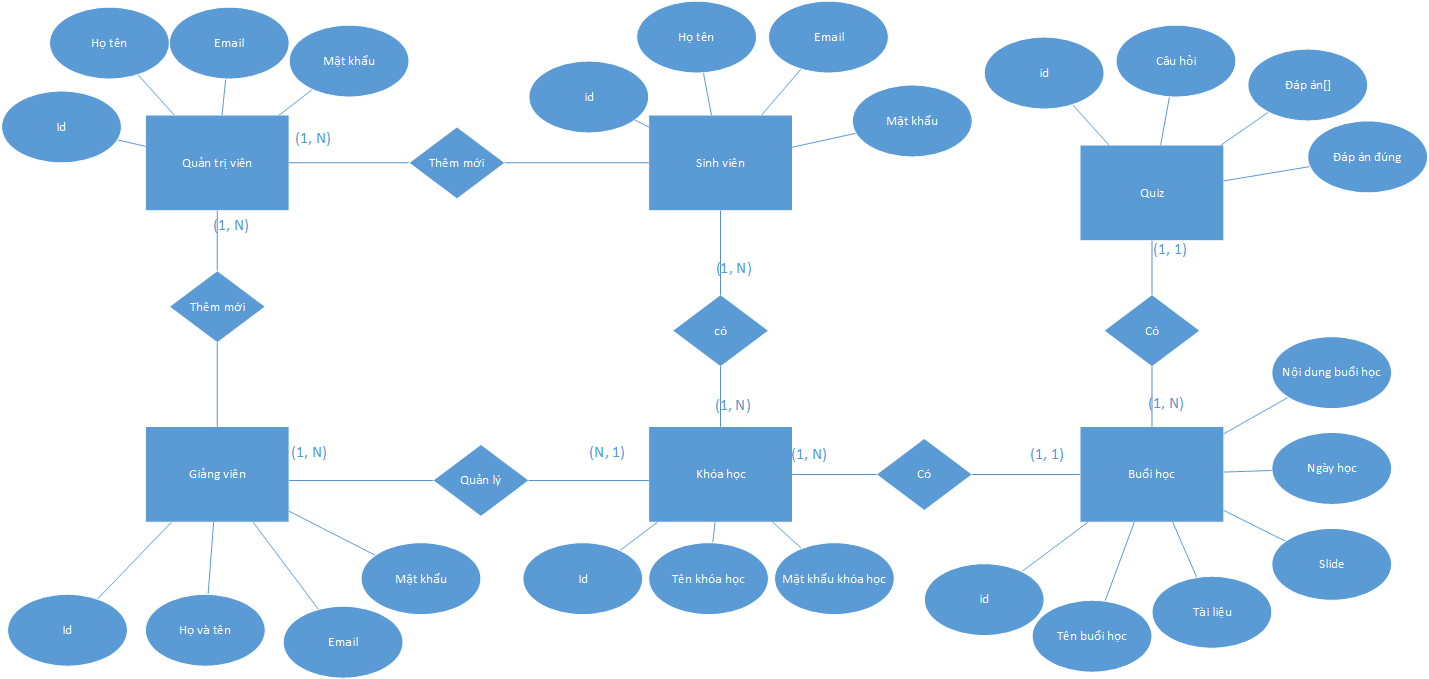


Hình 2.24: Cách thức hoạt động của hệ thống server

Khi người dùng đăng nhập hệ thống và sử dụng chức năng lớp học ảo, nếu session của lớp học ảo chưa được tạo một yêu cầu khởi tạo session sẽ được gửi đến server Nodejs, tại đây server sẽ tiến hành gửi một request đến media-server yêu cầu khởi tạo một session, media-server sẽ tạo ra một session mới và trả về sessionId cho server Nodejs, server Nodejs dựa vào sessionId tiến hành gửi một request đến media-server yêu cầu tạo một token mới cho người dùng tương ứng vào sessionId vừa tạo. Trong trường hợp session của lớp học đã tồn tại, server Nodejs sẽ chỉ cần gửi request yêu cầu media-server tạo một token mới cho người dùng tương ứng với session này.

## Thiết kế cơ sở dữ liệu

Phần này tập trung trình bày thiết kế cơ sở dữ liệu, các bảng dữ liệu, mối quan hệ giữa các bảng. Cơ sở dữ liệu của hệ thống viduca gồm các bảng cơ bản như sau: Bảng Users dùng để lưu thông tin của người dùng, bảng Courses dùng để lưu thông tin các khóa học, bảng Lessons để lưu thông tin các buổi học, bảng Quizs dùng để lưu thông tin các bài Quiz, bảng Attendances dùng để lưu danh sách điểm danh sinh viên. Các bảng dữ liệu được mô tả ở hình phía dưới:



Hình 2.25: Mô hình ER

Bảng User lưu trữ các trường thông tin cơ bản của người dùng như tên, email, password, quyền sử dụng hệ thống, trạng thái tài khoản, thời gian khởi tạo, thời gian chỉnh sửa. Hệ thống cho phép người dùng thay đổi các trường thông tin cá nhân của mình trong cơ sở dữ liệu như tên, password, riêng email và quyền sử dụng hệ thống người dùng không thể tự thay đổi.

Bảng Courses dùng để lưu thông tin các khóa học do giảng viên tạo ra, bảng này gồm các thông tin như tên khóa học, mô tả khóa học, id giảng viên hướng dẫn của khóa học, mảng chứa id sinh viên tham gia khóa học, mật khẩu khóa học, thời gian khởi tạo, thời gian chỉnh sửa. Trường lưu id của giảng viên có thể được sử dụng để tham chiếu đến bảng Users để trích xuất các thông tin khác của giảng viên trong trường hợp cần thiết, nó cũng giúp xác định xem giảng viên có quyền để thực thi các tác vụ với khóa học này hay không. Trường lưu mảng id của sinh viên cũng tương tự, được sử dụng để tham chiếu đến bảng Users để truy vấn thêm thông tin.

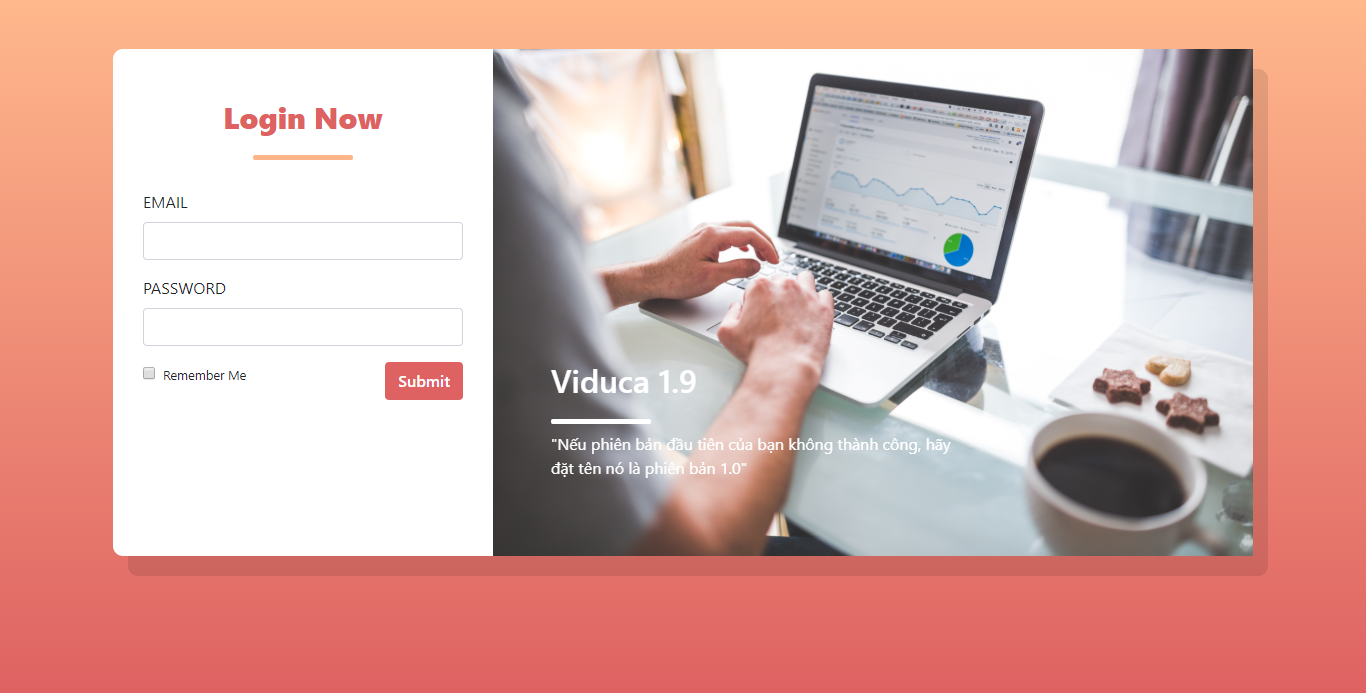
Bảng Lessons được sử dụng để lưu thông tin các buổi học do giảng viên tạo ra cho khóa học, bảng này gồm các trường thông tin cơ bản như tên buổi học, nội dung buổi học, id của khóa học, id của giảng viên hướng dẫn, ngày bắt đầu học, slide, tài liệu, thời gian khởi tạo, thời gian chỉnh sửa. Trường course trong bảng Lessons được sử dụng để tham chiếu đến bảng Courses lấy thêm thông tin trong trường hợp cần thiết, nó cũng được sử dụng trong trường hợp muốn truy vấn danh sách tất cả các buổi học của một khóa học. Trường instructor trong bảng Lesson cũng tham chiếu đến bảng Users giúp lấy thêm thông tin trong trường hợp cần thiết, và cũng giúp xác định xem giảng viên này có quyền sử dụng các tác vụ với buổi học này hay không.

Bảng Attendances được sử dụng để lưu danh sách điểm danh sinh viên với các trường thông tin cơ bản sau: trường lesson dùng để lưu id của buổi học được sử dụng để xác định buổi học mà sinh viên đã tham gia, trường student dùng để lưu id của sinh viên dùng để xác định những sinh viên nào đã tham gia buổi học, trường time là một mảng dùng để lưu mốc thời gian sinh viên vào lớp và mốc thời gian sinh viên thoát khỏi lớp.

## Thiết kế giao diện

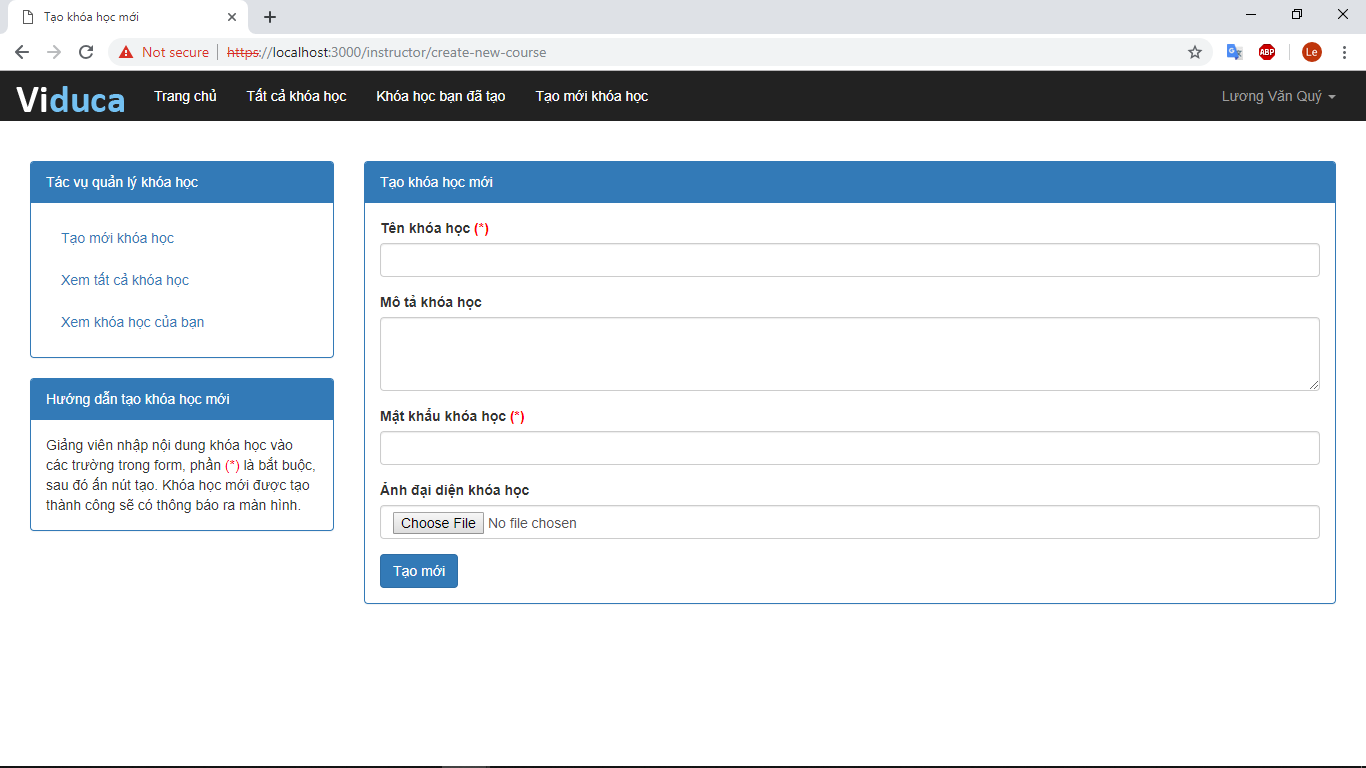
Hệ thống được xây dựng trên nền tảng Web, do đó giao diện được xây dựng bằng các ngôn ngữ web như HTML, CSS, Javascript. Ngoài ra, để tăng khả năng tương thích với các thiết bị di động khác, hiển thị với các kích thước màn hình khác nhau, giao diện còn được xây dựng bằng thư viện bootstrap. Dưới đây là một số màn hình giao diện cơ bản:

*Giao diện trang login*



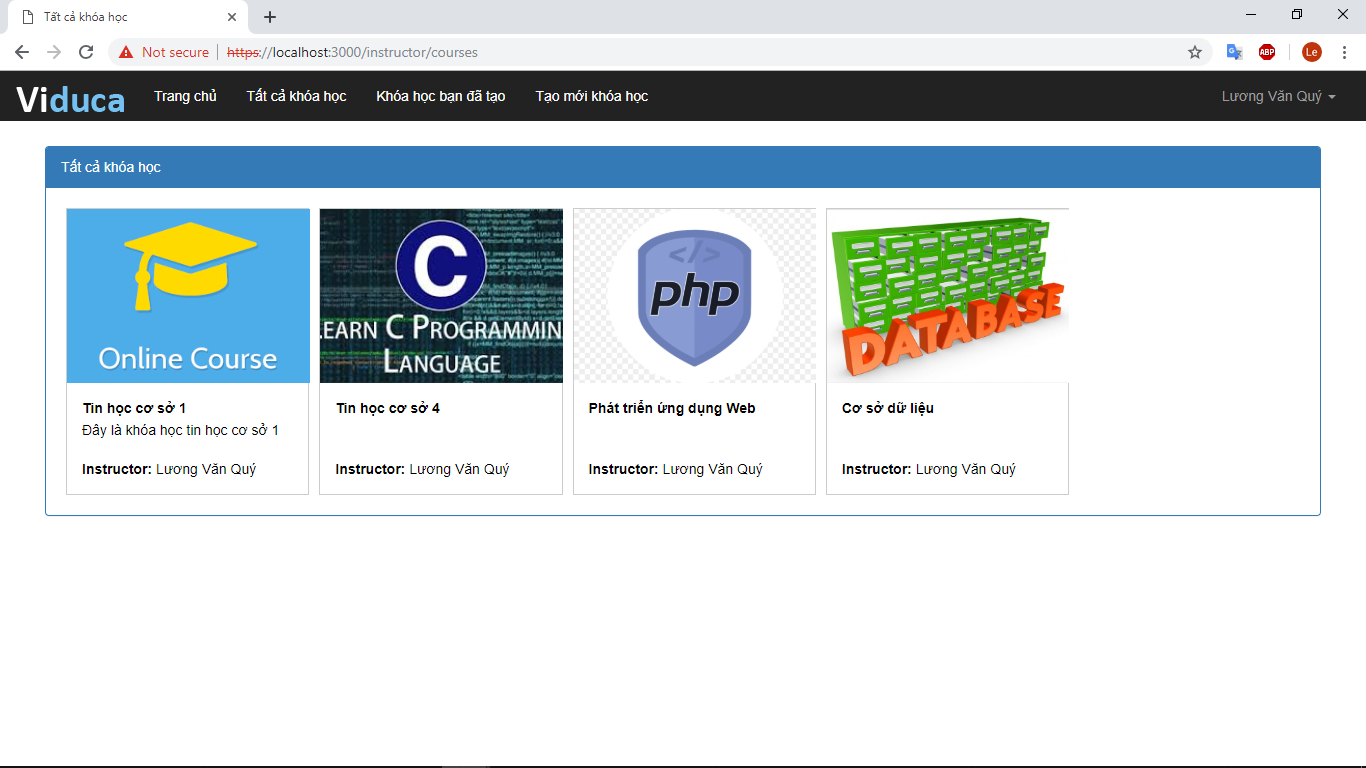
Hình 2.26: Giao diện trang login

*Giao diện trang tạo khóa học mới*



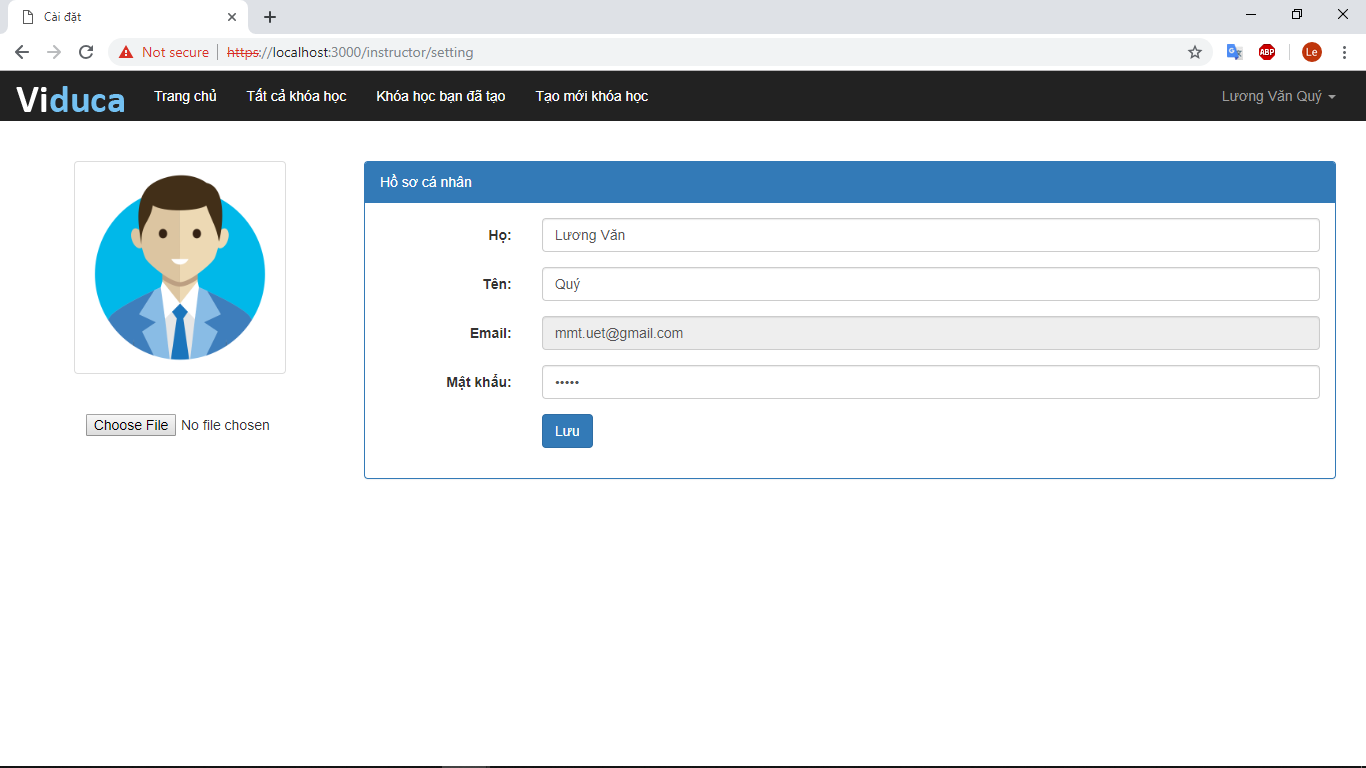
Hình 2.27: Giao diện trang tạo khóa học mới

*Giao diện trang khóa học*



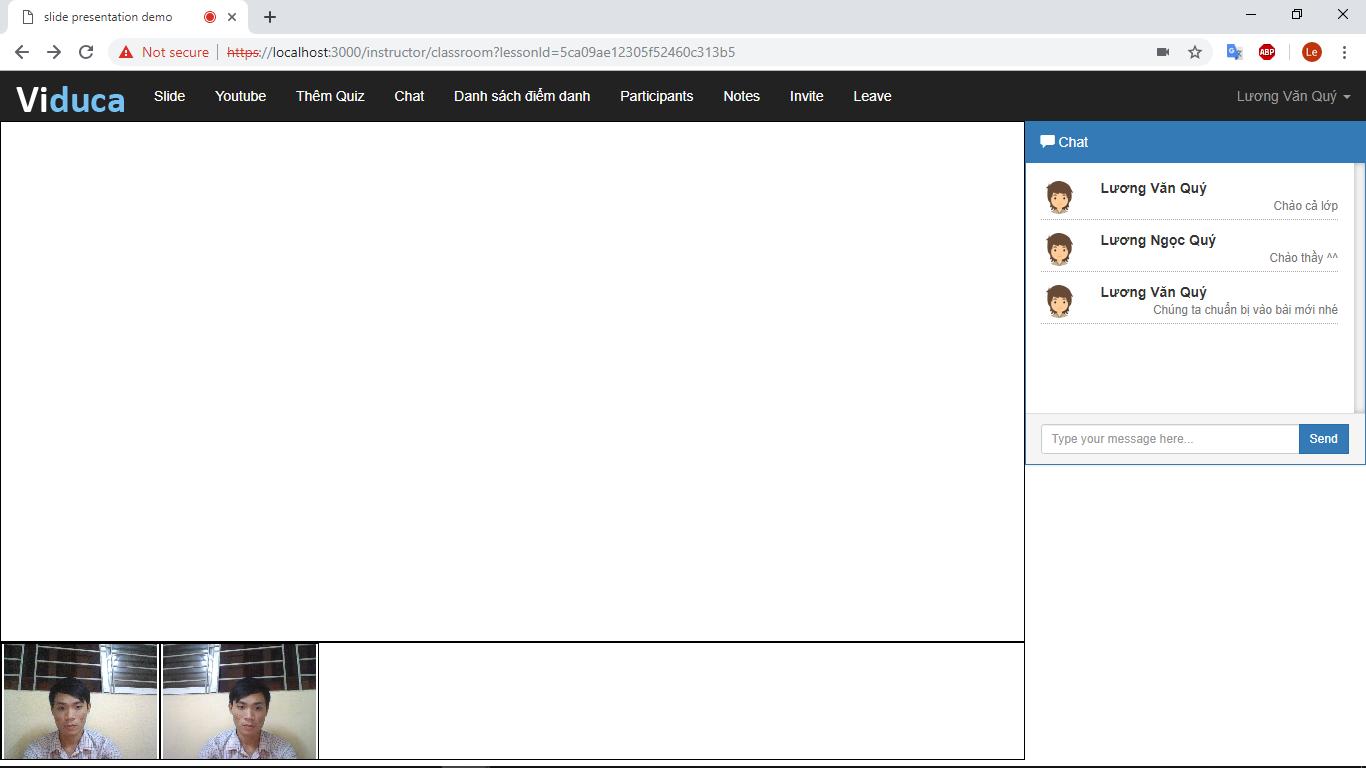
Hình 2.28: Giao diện trang danh sách khóa học

*Giao diện trang cập nhật thông tin*



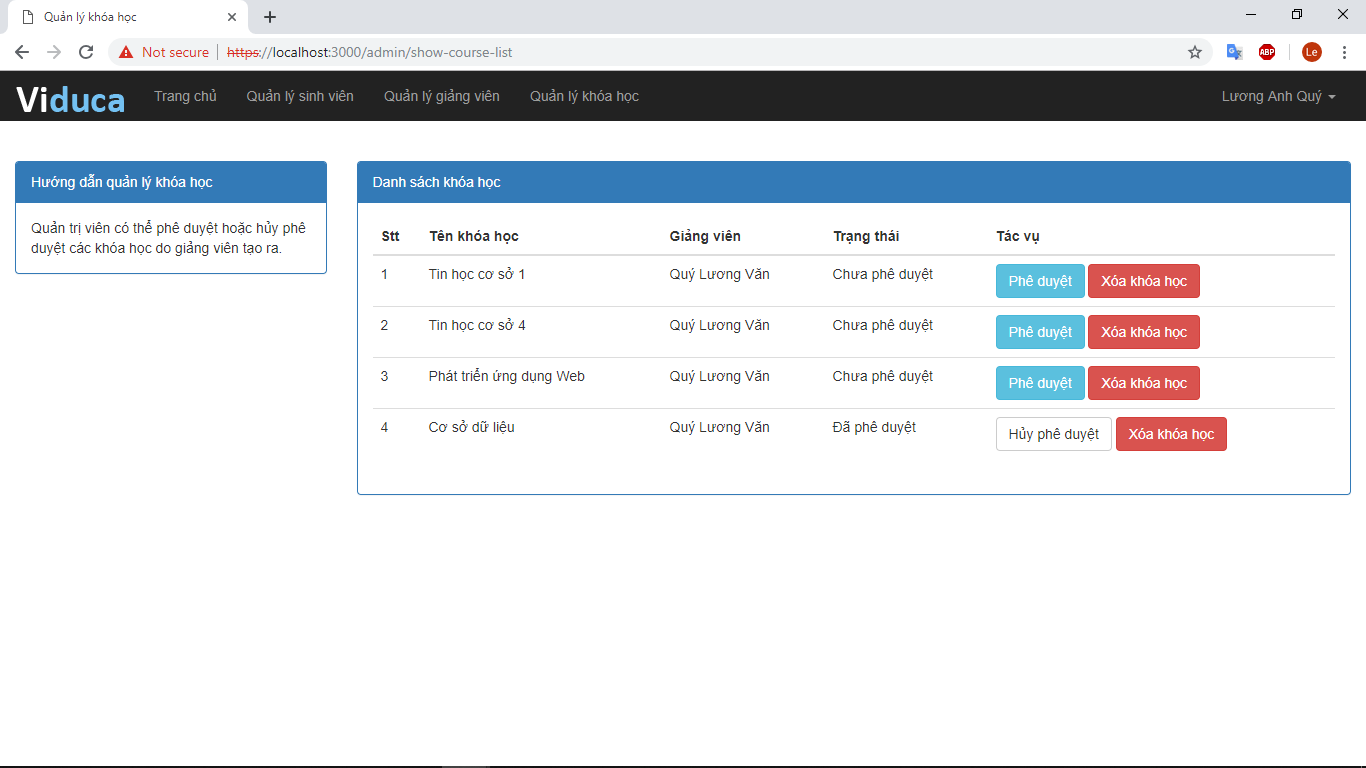
Hình 2.29: Giao diện trang cập nhật thông tin cá nhân

*Giao diện trang lớp học*



Hình 2.30: Giao diện lớp học ảo

*Giao diện quản lý khóa học của Quản trị viên*



Hình 2.31: Giao diện quản lý khóa học của Quản trị viên

# KIỂM THỬ HỆ THỐNG

## Điều kiện thử nghiệm

### Máy chủ

* Máy chủ Web: https cài đặt qua NodeJS phiên bản v10.15.3.
* Máy chủ OpenVidu phiên bản v2.8.0 cài đặt bằng docker.

### Trình duyệt

* Google chrome Version 73.0.3683.86 (Official Build) (64-bit)
* Firefox Quantum 66.0.3 (64-bit)
* Opera versions 15.0

## Kết quả kiểm thử

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Trình duyệt  Chức năng | Chrome | Firefox | Opera | IE |
| Truyền thông video, audio | **✓** | **🗶** | **✓** | **🗶** |
| Trình chiếu slide | **✓** | **🗶** | **✓** | **🗶** |
| Chat văn bản | **✓** | **🗶** | **✓** | **🗶** |
| Kiểm tra | **✓** | **🗶** | **✓** | **🗶** |
| Chia sẻ video youtube | **✓** | **🗶** | **✓** | **🗶** |
| Điểm danh tự động | **✓** | **🗶** | **✓** | **🗶** |
| Quản lý người dùng | **✓** | **✓** | **✓** | **✓** |
| Quản lý khóa học | **✓** | **✓** | **✓** | **✓** |
| Quản lý buổi học | **✓** | **✓** | **✓** | **✓** |
| Cập nhật thông tin | **✓** | **✓** | **✓** | **✓** |

Hiện tại tính năng lớp học ảo chỉ hoạt động tốt trên trình duyệt Chrome và Opera, rất có thể là do hiện tại hệ thống đang sử dụng chứng chỉ SSL tự ký nên bị chặn bởi trình duyệt Firefox và IE. Còn các tính năng như quản lý người dùng, quản lý khóa học, quản lý buổi học không bắt buộc phải sử dụng chứng chỉ SSL nên vẫn hoạt động tốt trên tất cả các trình duyệt.

# KẾT LUẬN CHUNG

## Các đóng góp của khóa luận

Với yêu cầu của đề tài khóa luận là xây dựng được một hệ thống học tập trực tuyến với các công cụ dạy học giúp tương tác trực tiếp giữa giảng viên với sinh viên, đồng thời vừa có thể hoạt động theo mô hình client-server vừa có thể hoạt động theo mô hình peer-to-peer để giảm tải cho server, khóa luận đã thu được một số kết quả sau:

* Tìm hiểu được những công nghệ như NodeJS, MongoDB, PDFJS, đặc biệt là các công nghệ mới WebRTC và OpenVidu.
* Sử dụng các công nghệ trên để triển khai thành công hệ thống với các công cụ cho phép tương tác thời gian thực như truyền thông video, audio, trình chiếu slide, chat văn bản, kiểm tra, điểm danh tự động.
* Xây dựng được hệ thống quản lý với các module quản lý người dùng, quản lý khóa học, quản lý buổi học.
* Kiểm thử các chức năng của hệ thống trên các loại trình duyệt khác nhau.
* Hệ thống có thể vừa hoạt động theo mô hình client-server vừa có thể hoạt động theo mô hình peer-to-peer.

## Một số hướng phát triển tiếp theo

Khóa luận đã đạt được một số kết quả nhất định, tuy nhiên còn nhiều tính năng nữa mà hệ thống của thể phát triển để đem đến sự trải nghiệm tốt hơn nữa cho người dùng. Trong thời gian tới, hệ thống có thể tiếp tục phát triển bổ sung thêm các tính năng như sau:

### Chia sẻ màn hình thời gian thực

Hiện tại, hệ thống học tập trực tuyến viduca vẫn chưa có tính năng này, tính năng này có thể sẽ rất hữu ích trong một số trường hợp như giảng viên muốn demo cho sinh viên xem một ứng dụng trên máy tính cá nhân hoặc cho sinh viên xem một file có kích thước lớn mà không cần phải tải lên hệ thống.

### Tính năng KPI

Tính năng này cho phép đặt ra KPI cho giảng viên và sinh viên trước khóa học. Giảng viên và sinh viên có thể xem xét mức độ tiến bộ, kế hoạch học tập đã đạt được chỉ tiêu hay chưa. Tính năng này sẽ thống kê lại một số tiêu chí như thời gian trong lớp học, số câu hỏi trả lời đúng… và mô hình hóa các dữ liệu này thành biểu đồ trực quan.

### Tính năng trình chiếu các bài giảng trực quan 3D

Đây là một trong những tính năng rất hữu ích đối với các môn học như hình học không gian, khoa học vũ trụ,… Tuy nhiên đây thực sự là một tính năng khó triển khai, cần một thời gian nghiên cứu dài hơi từ khâu làm sao để thiết kế được một bài giảng 3D đến việc làm sao để trình chiếu thời gian thực…

### Thông báo tin nhắn về điện thoại, gửi thông báo qua email

Tính năng này cho phép hệ thống tự động gửi thông báo về điện thoại hoặc gửi thông báo đến email của sinh viên và giảng viên trước khi buổi học sắp diễn ra.

Tính năng này cũng cho phép Quản trị viên gửi các tin nhắn thông báo đến sinh viên và giảng viên.

Sắp tới hệ thống có thể sẽ tiếp tục triển khai tính năng cần thiết này để tránh việc sinh viên và giảng viên quên lịch học.

### Khắc phục lỗi không hoạt động trên trình duyệt Firefox và IE

Hiện tại, tính năng lớp học ảo không hoạt động trên trình duyệt Firefox và IE có thể do hệ thống đang sử dụng chứng chỉ tự ký SSL. Sắp tới nếu có điều kiện tôi sẽ sử dụng chứng SSL do các cơ quan cung cấp để kiểm tra hệ thống trên trình duyệt Firefox và IE.

TÀI LIỆU THAM KHẢO

**TIẾNG ANH**

1. WebRTC: APIs and RTCWEB Protocols of the HTML5 Real-Time Web, Third Edition by Alan B Johnston, Daniel C Burnett
2. Web Development with Node and Express: Leveraging the JavaScript Stack 1st Edition by Ethan Brown
3. MongoDB 4 Quick Start Guide: Learn the skills you need to work with the world's most popular NoSQL database 1st Edition by Doug Bierer
4. HTML and CSS: Design and Build Websites 1st Edition by Jon Duckett
5. Learning jQuery 3 - Fifth Edition by Adam Boduch, Jonathan Chaffer, Karl Swedberg
6. Step By Step Bootstrap 3: A Quick Guide To Responsive Web Development Using Bootstrap 3 by Riwanto Megosinarso
7. OpenVidu Documentation: <https://openvidu.io/docs/home/>
8. Pdf.js wiki: <https://github.com/mozilla/pdf.js/wiki>