

ĐẠI HỌC HUẾ
TRƯỜNG ĐẠI HỌC KHOA HỌC
KHOA CÔNG NGHỆ THÔNG TIN
📖

KHOÁ LUẬN
TỐT NGHIỆP ĐẠI HỌC

Đề tài:

**XÂY DỰNG WEBSITE LUYỆN NGHE
TIẾNG ANH**

Sinh viên thực hiện: VÕ XUÂN AN
Khóa: K43 - HỆ CHÍNH QUY

Huế, tháng 5 năm 2023

ĐẠI HỌC HUẾ
TRƯỜNG ĐẠI HỌC KHOA HỌC
KHOA CÔNG NGHỆ THÔNG TIN
📖

KHOÁ LUẬN
TỐT NGHIỆP ĐẠI HỌC
NGÀNH CÔNG NGHỆ THÔNG TIN

Đề tài:

XÂY DỰNG WEBSITE LUYỆN NGHE
TIẾNG ANH

Sinh viên thực hiện: VÕ XUÂN AN
Khóa: K43- HỆ CHÍNH QUY
Giáo viên hướng dẫn: ThS. TRẦN NGUYỄN PHONG

Huế, tháng 5 năm 2023

LỜI CẢM ƠN

Lời đầu tiên, em xin gửi lời cảm ơn chân thành đến quý Thầy Cô Khoa công nghệ thông tin, Trường Đại học Khoa học, Đại học Huế đã dạy dỗ, hướng dẫn tận tình và dẫn dắt em trong suốt quá trình học tập tại trường. Các Thầy Cô là nguồn động lực lớn nhất giúp em kiên trì theo đuổi đam mê của em cho đến bây giờ.

Đặc biệt, em xin gửi lời cảm ơn sâu sắc đến thầy TS. Trần Nguyên Phong, giảng viên hướng dẫn em trong suốt quá trình thực hiện bài khóa luận này. Trong quá trình làm khóa luận, thầy đã tận tình hỗ trợ, giải đáp các thắc mắc, đóng góp ý kiến để em có thể hoàn thành khóa luận tốt nghiệp một cách tốt nhất so với những gì đã đặt ra.

Em cũng xin gửi lời cảm ơn đến gia đình, bạn bè và những người thân yêu của em đã luôn bên cạnh, ủng hộ, nhắc nhở và động viên em.

Mặc dù đã em đã cố gắng hoàn thiện thật tốt khóa luận tốt nghiệp nhưng không thể nào tránh khỏi những sai sót, em rất mong nhận được sự cảm thông và góp ý, đánh giá của quý thầy cô và các bạn.

Em xin gửi đến tất cả mọi người lời chúc sức khỏe, hạnh phúc và thành công. Em xin chân thành cảm ơn!

Huế, tháng 5 năm 2023

Sinh viên

Võ Xuân An

DANH MỤC THUẬT NGỮ VÀ TỪ VIẾT TẮT

Chữ viết tắt	Tiếng Anh	Diễn giải
JSON	JavaScript Object Notation	Tiêu chuẩn định dạng dữ liệu
XML	Extensible Markup Language	Ngôn ngữ đánh dấu mở rộng
HTML	Hyper Text Markup Language	Ngôn ngữ đánh dấu siêu văn bản
NOSQL	Not Only Structured Query Language	Hệ thống cơ sở dữ liệu phi quan hệ
API	Application Programming Interface	Giao diện lập trình ứng dụng
URL	Uniform Resource Locator	Địa chỉ web
HTTP	Hypertext Transfer Protocol	Một giao thức truyền tải dữ liệu
BSON	Binary JSON	Một định dạng dữ liệu nhị phân
CRUD	Create, Read, Update, Xóa	Tạo, đọc, sửa, xóa
IMAP	Internet Message Access Protocol	Một giao thức được sử dụng để truy cập và quản lý email trên máy chủ email từ xa
SMTP	Simple Mail Transfer Protocol	Một giao thức mạng được sử dụng để truyền tải và gửi email qua internet
POP3	Post Office Protocol version 3	Một giao thức mạng được sử dụng để tải về và quản lý email từ một máy chủ email

OAuth2	Open Authorization 2.0	Một giao thức ủy quyền để quản lý quyền truy cập và chia sẻ thông tin giữa các ứng dụng
SSL/TLS/ UDP/ TCP	Secure Sockets Layer / Transport Layer Security/ User Datagram Protocol/ Transmission Control Protocol	Các giao thức truyền dữ liệu
NAT	Network Address Translation	Công nghệ chuyển đổi địa chỉ IP từ một dạng sang dạng khác
STUN	Session Traversal Utilities for NAT	Một giao thức mạng được sử dụng để giúp các thiết bị trong mạng có thể kết nối trực tiếp với các thiết bị khác qua Internet khi có NAT hoặc Firewall trong mạng.
TURN	Traversal Using Relays around NAT	Một giao thức mạng dùng để vượt qua các ràng buộc NAT
ICE	Interactive Connectivity Establishment	Một giao thức mạng được sử dụng trong việc thiết lập kết nối trực tiếp giữa các thiết bị trong mạng
SDP	Session Description Protocol	Một giao thức dùng để mô tả các thông tin về một phiên truyền thông trong các ứng dụng truyền thông thời gian thực

MỤC LỤC

PHẦN MỞ ĐẦU	1
1. Lý do chọn đề tài	1
2. Ý nghĩa thực hiện đề tài	1
CHƯƠNG 1: TỔNG QUAN VỀ ĐỀ TÀI VÀ CÁC CÔNG CỤ LIÊN QUAN	3
1.1. Tổng quan về đề tài	3
1.1.1. Đặt vấn đề	3
1.1.2. Đặc tả yêu cầu bài toán	3
1.1.3. Yêu cầu đối với sản phẩm	4
1.2. Các công cụ phục vụ nghiên cứu và thực hiện đề tài	4
1.2.1. Express	5
a. Giới thiệu về Express	5
b. Một số khía cạnh quan trọng của Express	5
1.2.2. React	6
a. Giới thiệu về React	6
b. Một số khía cạnh quan trọng của Express	7
1.2.3. RESTful API	8
a. Giới thiệu về API, REST, RESTful API	8
c. Ưu điểm của RESTful API	10
1.2.3. MongoDB	10
a. Giới thiệu về MongoDB	10
b. Một số khía cạnh quan trọng của MongoDB	10
1.3. Các công cụ hỗ trợ	11
1.3.1. WebRTC	11
a. Giới thiệu về WebRTC	11
b. Các thành phần của WebRTC	12
c. Cách WebRTC thiết lập kết nối ngang hàng	12
1.3.2. NodeMailer	14
CHƯƠNG 2: PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG	16
1.1. Phân tích và thiết kế hệ thống	16
1.1.1. Phân rã chức năng	16
1.1.2. Các tác nhân của hệ thống	16
1.1.3. Biểu đồ UseCase	17
1.1.3.1. Quản trị viên (Admin)	17

1.1.3.2. Người dùng (User)	21
1.1.4. Biểu đồ hoạt động.....	22
1.1.4.1. Đăng nhập	23
1.1.4.2. Đăng ký.....	24
1.1.4.3. Đặt lại mật khẩu	24
1.1.4.4. Video Call	25
1.1.4.5. Quản lý người dùng.....	26
1.1.4.6. Quản lý bài đăng	26
1.1.4.7. Quản lý chủ đề	26
1.1.4.8. Quản lý bài học	27
1.1.4.9. Quản lý câu hỏi	28
1.1.4.10. Làm bài	28
1.1.5. Biểu đồ lớp.....	29
1.2. Thiết kế giao diện	30
1.2.1. Giao diện đăng nhập.....	30
1.2.2. Giao diện đăng ký.....	30
1.2.3. Giao diện đặt lại mật khẩu	30
1.2.4. Giao diện thay đổi thông tin cá nhân.....	31
1.2.5. Giao diện video call.....	32
1.2.6. Giao diện bài đăng.....	33
1.2.7. Giao diện chủ đề.....	35
1.2.8. Giao diện bài học.....	35
1.2.9. Giao diện làm bài	35
1.3. Hoàn thiện sản phẩm	37
1.3.1. Đánh giá mức độ hoàn thiện	37
1.3.2. Hướng dẫn cài đặt sản phẩm.....	37
a. Môi trường cài đặt	37
b. Tiến hành cài đặt và chạy ứng dụng.....	38
PHẦN KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN	39
1. Kết luận.....	39
2. Hướng phát triển.....	39
TÀI LIỆU THAM KHẢO.....	40

PHẦN MỞ ĐẦU

1. Lý do chọn đề tài

Hiện nay, việc học tiếng anh đã và đang là một quan trọng và thiết yếu trong cuộc sống nó giúp ta nâng cao trình độ, mở rộng kiến thức và có thêm nhiều cơ hội trong công việc, học tập cũng như trong cuộc sống. Tuy nhiên, kỹ năng nghe vẫn là một thách thức đối với nhiều người học, đặc biệt là những người không có nhiều cơ hội giao tiếp bằng anh trong cuộc sống hằng ngày.

Sử dụng một website luyện nghe tiếng anh để luyện tập là một giải pháp tiện lợi, hiệu quả và linh hoạt cho những người muốn cải thiện kỹ năng nghe của mình. Với sự phát triển của công nghệ thông tin, việc tiếp cận tài nguyên học tiếng anh trực tuyến đã và đang trở nên dễ dàng hơn bao giờ hết.

Với mong muốn xây dựng một website luyện nghe tiếng anh miễn phí và dễ dàng sử dụng, website dành cho mọi đối tượng người học. Đặc biệt, website này sẽ hữu ích cho những người không có nhiều cơ hội giao tiếp và nghe nói tiếng anh trong cuộc sống hằng ngày... Với các lý do đó em chọn đề tài “***Xây dựng website luyện nghe tiếng anh***”. Em tin tưởng rằng đây sẽ là một đề tài thực tế và có thể giúp ích cho nhiều người.

2. Ý nghĩa thực hiện đề tài

Thực hiện xây dựng website luyện nghe tiếng anh giúp sinh viên nắm vững được các kiến thức và kỹ năng để phân tích, xây dựng một hệ thống website và đánh giá được năng lực của bản thân. Bên cạnh đó nó giúp người tiếng anh có cơ hội luyện tập và nâng cao kỹ năng nghe của mình một cách hiệu quả.

Cung cấp các tài liệu nghe đa dạng: website sẽ cung cấp cho người dùng các tài liệu nghe đa dạng từ các nguồn khác nhau, từ đó giúp người dùng đa dạng hóa phạm vi ngữ pháp, vốn từ, phản xạ phản ứng của mình

Cung cấp chức năng nhập liệu và trắc nghiệm: website sẽ giúp người học tiếng Anh có cơ hội luyện tập viết lại các đoạn văn bản nghe, đồng thời cung cấp cho người dùng các câu hỏi trắc nghiệm liên quan đến nội dung nghe. Điều này giúp người dùng có cơ hội vận dụng kiến thức của mình để hoàn thành bài tập.

Cung cấp tính năng video call: website sẽ cung cấp tính năng video call giữa hai người dùng để có thể học tập và giao tiếp với nhau, từ đó giúp người dùng có cơ hội luyện tập và phát triển kỹ năng nghe, nói và giao tiếp.

CHƯƠNG 1: TỔNG QUAN VỀ ĐỀ TÀI VÀ CÁC CÔNG CỤ LIÊN QUAN

1.1. Tổng quan về đề tài

1.1.1. Đặt vấn đề

Cùng với sự phát triển không ngừng của kinh tế và xã hội, tiếng anh ngày càng trở nên quan trọng hơn đối với con người. Tiếng anh được coi là ngôn ngữ giao tiếp quốc tế và là công cụ cần thiết để tham gia vào một số ngành nghề như du lịch, dịch thuật, kinh doanh, công nghệ thông tin, giáo dục...

Tuy nhiên, việc học tiếng anh luôn là một thách thức với nhiều người, đặc biệt là trong việc luyện nghe. Luyện nghe tiếng anh đòi hỏi sự cố gắng và thời gian dài hơn so với các kỹ năng khác như đọc, viết hay nói. Bởi vậy, nhiều người muốn tìm kiếm các giải pháp để có thể luyện nghe tiếng anh hiệu quả và tiết kiệm thời gian.

Trong bối cảnh này, việc xây dựng một website luyện nghe tiếng anh trở nên vô cùng cần thiết. Website này có thể giúp người học tiếng Anh luyện nghe một cách hiệu quả và tiện lợi, đồng thời cung cấp các tài liệu và bài học phong phú, đa dạng để giúp người học nâng cao kỹ năng nghe tiếng Anh của mình.

Với đề tài này em xin được trình bày cách thức xây dựng và quản lý website đầy đủ chức năng cần thiết, để cho dùng những người có nhu cầu cải thiện tiếng anh của mình dễ dàng tiếp cận và sử dụng.

1.1.2. Đặc tả yêu cầu bài toán

Để sử dụng được website người dùng hay người quản trị cần phải tạo tài khoản để thực hiện được các tác vụ trên website, nếu chưa có tài khoản có thể tiến hành đăng ký tài khoản, nếu người dùng hay người quản trị quên mật khẩu có thể tiến hành đặt lại mật khẩu, website sẽ gửi đến mail của tài khoản quên mật khẩu một liên kết để đặt lại mật khẩu, liên kết này chỉ có tác dụng trong vòng 15 phút kể từ khi mail được gửi đi.

Người quản trị trang web chọn lọc và đăng tải các bài nhập liệu, các bài tập với đầy đủ thông tin như: ảnh minh họa, audio, các lựa chọn của bài tập (nếu có), câu trả lời ...

Người dùng website có thể xem hoặc đăng những bài chia sẻ (blog) về tiếng anh, phương pháp học tiếng anh của mình, có thể xem và luyện tập những bài tập trên hệ thống, ngoài ra người dùng website còn có thể gọi video call với nhau, bằng cách này người dùng (bạn bè, thầy trò...) có thể tương tác trực tiếp với nhau để cùng nhau học tập.

1.1.3. Yêu cầu đối với sản phẩm

- Website phải có dung lượng không quá lớn, tốc độ xử lý nhanh
- Giao diện dễ nhìn, thân thiện với người dùng
- Bảo mật thông tin khách hàng, đảm bảo an toàn dữ liệu khi vận hành
- Website phải thực hiện được các chức năng như đã thiết kế, không thừa không thiếu

1.2. Các công cụ phục vụ nghiên cứu và thực hiện đề tài

Website được phát triển độc lập client và server với các công cụ hỗ trợ xây dựng và phát triển:

1.2.1. Express

a. Giới thiệu về Express

Express là một trong những framework web phổ biến nhất cho Node.js được phát triển để xây dựng các ứng dụng web và API. Nó giúp cho việc phát triển ứng dụng web với Node.js trở nên nhanh chóng và dễ dàng hơn. Express cung cấp một loạt các tính năng, middleware và công cụ giúp các nhà phát triển xây dựng các ứng dụng web và API một cách dễ dàng và hiệu quả.

Express cung cấp một cách tiếp cận rất linh hoạt để xử lý các yêu cầu HTTP và cung cấp các công cụ để quản lý session và cookie. Express cũng có khả năng tích hợp với nhiều loại cơ sở dữ liệu khác nhau và hỗ trợ middleware của bên thứ ba.

Với Express, việc xây dựng các ứng dụng web và API trở nên đơn giản hơn bao giờ hết. Nó cho phép các nhà phát triển tập trung vào việc xây dựng các tính năng của ứng dụng thay vì phải quan tâm đến các chi tiết thấp hơn của việc xử lý yêu cầu HTTP.

b. Một số khía cạnh quan trọng của Express

- Đơn giản và linh hoạt: Express được thiết kế để đơn giản và linh hoạt, cho phép bạn xây dựng các ứng dụng web phức tạp hoặc nhỏ gọn một cách dễ dàng. Nó cung cấp một cú pháp đơn giản và rõ ràng để xử lý các yêu cầu (request) và phản hồi (response) từ phía máy chủ.
- Middleware: Express sử dụng khái niệm middleware, cho phép bạn xử lý các yêu cầu trước khi chúng đến tới các xử lý cuối cùng. Điều này rất hữu ích khi bạn cần thực hiện các hoạt động chung như xác thực, ghi log, xử lý

lỗi... Express có sẵn nhiều middleware tích hợp sẵn và cũng cho phép bạn tự tạo middleware tùy chỉnh.

- **Routing:** Express cung cấp một cách đơn giản để xác định các tuyến đường (routes) và xử lý các yêu cầu tới các tuyến đường đó. Bằng cách sử dụng các phương thức HTTP như GET, POST, PUT, DELETE, ... bạn có thể định nghĩa các hành động xử lý tương ứng với từng tuyến đường.
- **Template engine:** Express hỗ trợ sử dụng các template engine như EJS, Pug, Handlebars... để tạo các trang web động. Template engine giúp bạn tạo ra các giao diện người dùng linh hoạt và dễ dàng tích hợp dữ liệu vào trang web.
- **Xử lý lỗi:** Express cung cấp các công cụ để xử lý và quản lý lỗi. Bạn có thể xác định middleware đặc biệt để xử lý các lỗi xảy ra trong quá trình xử lý yêu cầu, từ đó giúp tăng cường độ tin cậy và hiệu suất của ứng dụng.
- **Hỗ trợ RESTful API:** Express thích hợp để xây dựng các RESTful API. Với Express, bạn có thể xử lý các yêu cầu API và trả về dữ liệu dưới dạng JSON, cho phép ứng dụng của bạn tương tác với các

1.2.2. React

a. Giới thiệu về React

React là một thư viện JavaScript phổ biến được sử dụng để xây dựng các ứng dụng web động, tương tác và dễ dàng bảo trì. Với React, người phát triển có thể xây dựng giao diện người dùng phức tạp và hiệu quả hơn bằng cách tách nhỏ giao diện thành các thành phần độc lập nhau. React cũng cung cấp một cách tiếp cận khác biệt với quản lý trạng thái ứng dụng thông qua việc sử dụng các component state và props, giúp cho việc phát triển và bảo trì ứng dụng trở nên dễ dàng hơn.

React còn có khả năng tái sử dụng code giữa các component khác nhau, tạo ra một sự nhất quán và giảm thiểu số lượng code lặp lại trong ứng dụng. Ngoài ra, React cũng có thể kết hợp với các thư viện và công cụ khác để cung cấp các tính năng bổ sung như routing, quản lý state, gọi API, và nhiều hơn nữa.

Do tính linh hoạt và khả năng tái sử dụng của nó, React đã trở thành một công nghệ phổ biến trong việc xây dựng các ứng dụng web hiện đại. Trong đề tài xây dựng website luyện nghe tiếng Anh, sử dụng React sẽ giúp cho việc tạo ra các thành phần giao diện tương tác, quản lý state và routing trở nên dễ dàng hơn, đồng thời cũng giúp tăng tính linh hoạt và khả năng bảo trì của ứng dụng.

b. Một số khía cạnh quan trọng của Express

- **Component-based:** React sử dụng mô hình phát triển dựa trên thành phần (component-based development). Một thành phần (component) trong React là một phần độc lập của giao diện người dùng, có thể được tái sử dụng và quản lý một cách độc lập. Điều này giúp tăng tính modular và khả năng bảo trì của ứng dụng.
- **Virtual DOM:** React sử dụng một cơ chế gọi là Virtual DOM để cải thiện hiệu suất và hiệu quả trong việc cập nhật giao diện người dùng. Virtual DOM là một bản sao của DOM (Document Object Model) và được React sử dụng để so sánh và xác định sự thay đổi trong giao diện. Khi có sự thay đổi, React sẽ cập nhật chỉ những phần tử cần thiết trên DOM thực tế, giúp tăng tốc độ và hiệu suất của ứng dụng.
- **JSX:** React sử dụng JSX (JavaScript XML) để tạo ra các thành phần giao diện người dùng. JSX là một phần mở rộng của JavaScript cho phép bạn kết hợp mã JavaScript với cú pháp HTML. Điều này giúp tạo ra mã nguồn dễ đọc, dễ hiểu và dễ duy trì.

- Unidirectional Data Flow: React sử dụng mô hình dữ liệu unidirectional flow để quản lý và cập nhật trạng thái của ứng dụng. Dữ liệu trong React luôn di chuyển theo một hướng duy nhất từ thành phần cha đến các thành phần con. Điều này giúp dễ dàng theo dõi và xác định nguồn gốc của dữ liệu và giúp tạo ra ứng dụng dễ bảo trì.

1.2.3. RESTful API

a. Giới thiệu về API, REST, RESTful API

API (Application Programming Interface) là một tập các quy tắc và cơ chế mà theo đó, một ứng dụng hay một thành phần sẽ tương tác với một ứng dụng hay thành phần khác. API có thể trả về dữ liệu mà bạn cần cho ứng dụng của mình ở những kiểu dữ liệu phổ biến như JSON hay XML.

REST (REpresentational State Transfer) là một dạng chuyển đổi cấu trúc dữ liệu, một kiểu kiến trúc để viết API. Nó sử dụng phương thức HTTP đơn giản để tạo cho giao tiếp giữa các máy. Vì vậy, thay vì sử dụng một URL cho việc xử lý một số thông tin người dùng, REST gửi một yêu cầu HTTP như GET, POST, DELETE... đến một URL để xử lý dữ liệu.

RESTful API là một tiêu chuẩn dùng trong việc thiết kế các API cho các ứng dụng web để quản lý các resource. RESTful là một trong những kiểu thiết kế API được sử dụng phổ biến ngày nay để cho các ứng dụng (web, mobile...) khác nhau giao tiếp với nhau.

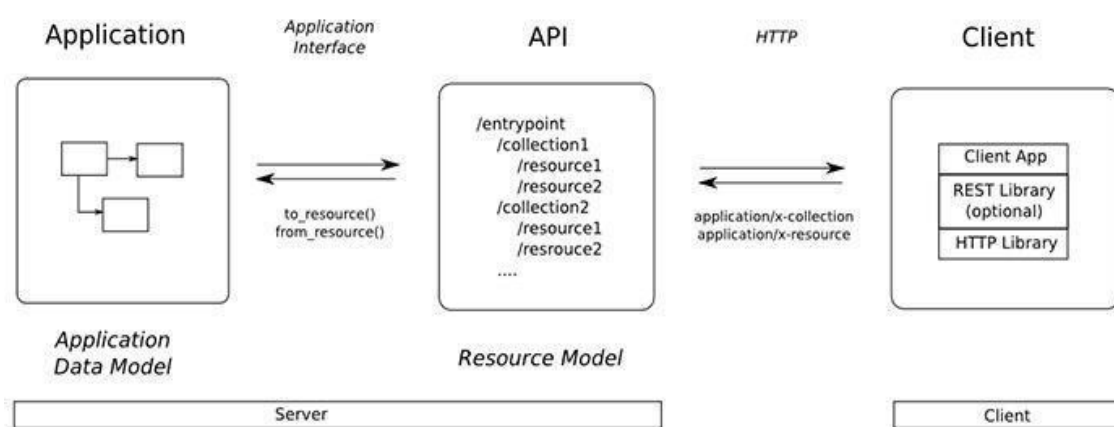
Chức năng quan trọng nhất của REST là quy định cách sử dụng các HTTP method (như GET, POST, PUT, DELETE...) và cách định dạng các URL cho ứng dụng web để quản lý các resource. RESTful không quy định logic code

ứng dụng và không giới hạn bởi ngôn ngữ lập trình ứng dụng, bất kỳ ngôn ngữ hoặc framework nào cũng có thể sử dụng để thiết kế một RESTful API.

b. Kiến trúc của RESTful API



Hình 1: Kiến trúc RESTful



Hình 2: Sơ đồ hoạt động của RESTful

REST hoạt động chủ yếu dựa vào giao thức HTTP. Các hoạt động cơ bản nêu trên sẽ sử dụng những phương thức HTTP riêng.

- GET (SELECT): Trả về một Resource hoặc một danh sách Resource.
- POST (CREATE): Tạo mới một Resource.

- PUT (UPDATE): Cập nhật thông tin cho Resource.
- DELETE (DELETE): Xóa một Resource.

Những phương thức hay hoạt động này thường được gọi là **CRUD** tương ứng với create (tạo), read (đọc), update (sửa), delete (xóa).

Hiện tại đa số lập trình viên viết **RESTful API** giờ đây đều chọn JSON là format chính thức nhưng cũng có nhiều người chọn XML làm format, nói chung dùng thế nào cũng được miễn tiện và nhanh.

c. Ưu điểm của RESTful API

Giúp cho ứng dụng được rõ ràng hơn, REST URL đại diện cho resource chứ không phải hành động. Dữ liệu được trả về với nhiều định dạng khác nhau như: xml, html, json... code đơn giản và ngắn gọn. REST chú trọng vào tài nguyên của hệ thống.

1.2.3. MongoDB

a. Giới thiệu về MongoDB

MongoDB là một hệ quản trị cơ sở dữ liệu (Database Management System - DBMS) phi quan hệ, mã nguồn mở. MongoDB sử dụng mô hình dữ liệu linh hoạt được gọi là "documents" thay vì các bảng và hàng như trong cơ sở dữ liệu quan hệ truyền thống. MongoDB được phát triển dựa trên kiến trúc NoSQL, đồng thời hỗ trợ mạnh mẽ cho việc mở rộng và phân tán dữ liệu.

b. Một số khía cạnh quan trọng của MongoDB

- Cấu trúc dữ liệu: MongoDB lưu trữ dữ liệu dưới dạng JSON-like document gọi là BSON (Binary JSON). Mỗi document có thể chứa các cặp khóa-giá trị và các tài liệu có thể khác nhau về cấu trúc.
- Linh hoạt: MongoDB cho phép bạn thay đổi cấu trúc dữ liệu mà không cần sửa đổi cấu trúc schema hoặc ngừng hoạt động. Điều này cho phép bạn dễ dàng thay đổi ứng dụng của mình theo thời gian và tương thích tốt với việc phát triển sản phẩm.
- Tốc độ và mở rộng: MongoDB hỗ trợ khả năng mở rộng theo chiều ngang (horizontal scaling) bằng cách phân tán dữ liệu trên nhiều máy chủ. Điều này giúp cải thiện hiệu suất và khả năng xử lý của hệ thống khi tải tăng.
- Truy vấn linh hoạt: MongoDB cung cấp một ngôn ngữ truy vấn phong phú và mạnh mẽ gọi là MongoDB Query Language (MQL). MQL hỗ trợ các truy vấn phức tạp, tìm kiếm văn bản đầy đủ và các phép toán đa truy vấn.
- Hỗ trợ replica set và sharding: MongoDB cung cấp khả năng sao chép dữ liệu tự động thông qua replica set. Điều này giúp cung cấp khả năng dự phòng và chịu lỗi cao. MongoDB cũng hỗ trợ phân mảnh dữ liệu (sharding) để phân tán dữ liệu trên nhiều máy chủ để tăng khả năng chịu tải.
- Hỗ trợ index: MongoDB hỗ trợ các loại index như single field, compound, multi-key và geospatial indexes. Indexing giúp tăng tốc độ truy vấn và cải thiện hiệu

1.3. Các công cụ hỗ trợ

1.3.1. WebRTC

a. Giới thiệu về WebRTC

WebRTC (Web Real-Time Communication) là một công nghệ mở được phát triển bởi Google nhằm hỗ trợ việc truyền thông thời gian thực (real-time communication) trên web. Nó cung cấp các giao thức và API để cho phép

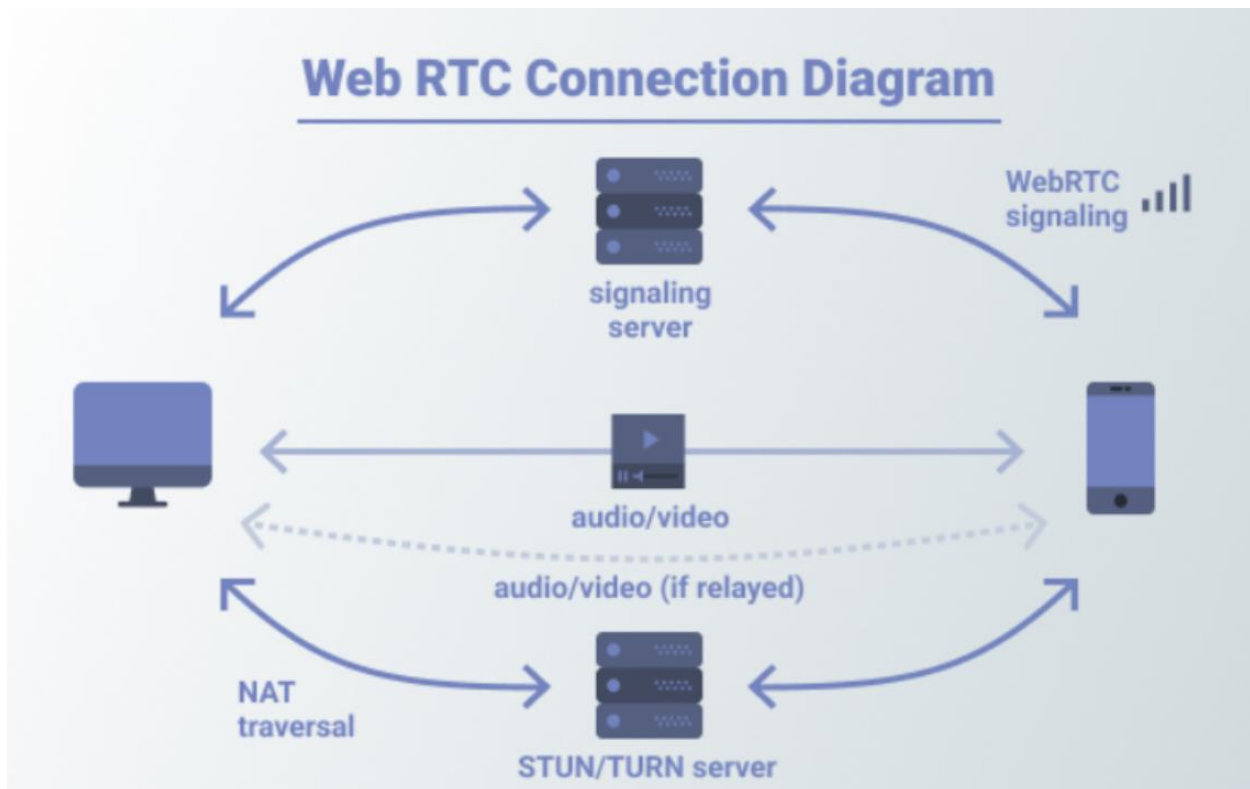
truyền dữ liệu âm thanh, video và dữ liệu trực tuyến giữa các trình duyệt web mà không cần thông qua các plugin hay ứng dụng bên ngoài.

b. Các thành phần của WebRTC

WebRTC bao gồm các thành phần chính sau đây:

- **RTCPeerConnection:** Đây là thành phần quan trọng nhất của WebRTC. Nó cho phép thiết lập và quản lý kết nối trực tiếp giữa các máy tính trong việc truyền thông qua mạng.
RTCPeerConnection hỗ trợ mã hóa, giải mã, điều khiển độ trễ, và các chức năng khác liên quan đến việc truyền dữ liệu âm thanh, video và dữ liệu.
- **RTCDataChannel:** Đây là một thành phần của WebRTC cho phép truyền dữ liệu trực tiếp giữa các trình duyệt. Nó giúp cho việc truyền tải các dữ liệu như tin nhắn văn bản, tệp tin và thông tin ứng dụng.
- **RTCSessionDescription:** Được sử dụng để mô tả thông tin về phiên kết nối như kiểu kết nối, các thông số và cấu hình liên quan.
- **getUserMedia:** API cho phép truy cập đến các thiết bị đa phương tiện như webcam và microphone của người dùng. Nó cho phép ứng dụng web lấy dữ liệu âm thanh và video từ người dùng để truyền đi hoặc xử lý.

c. Cách WebRTC thiết lập kết nối ngang hàng



Hình 3: Sơ đồ hoạt động của WebRTC

Để thiết lập một kết nối trên internet thì 2 máy phải biết IP của nhau và mở port để có thể gửi và nhận dữ liệu qua port đó, khi một máy tính nằm trong một mạng NAT private và public IP của nó là không giống nhau do đó để máy tính đó có thể kết nối với máy tính khác thì lúc này nó sẽ gửi một request tới STUN server, STUN server sẽ nhận request và trả về thông tin này cho máy tính.

Mỗi máy tính sẽ tạo ra một bản mô tả SDP (Session Description Protocol) chứa thông tin về các multimedia của nó, thông tin này sẽ được chuyển cho máy tính kia để thiết lập kết nối. Thông thường ta sẽ tạo một Signaling Server để làm việc này, Signaling server có thể sử dụng Websocket, HTTP... miễn sao gửi thông tin này thành công

Khi hai máy biết thông tin về IP, SDP, các port có thể sử dụng của nhau mỗi máy tính sẽ tạo ra các ứng viên ICE đại diện cho các cặp IP/port mạng

mà nó có thể truyền qua các ICE này có thể bao gồm địa chỉ IP và cổng mạng của máy tính cũng như của các proxy hoặc NAT, nó sẽ thử thiết lập kết nối trên các đường truyền thông qua NAT được thể hiện trên các ICE được tạo cho đến khi thiết lập được kết nối hoặc thử hết các trường hợp (ưu tiên UDP > TCP > TLS)

Khi kết nối thành công máy tính sẽ biết được IP và cổng mạng có thể truyền được thông tin cho nhau, các thông tin này sẽ được sử dụng để thiết lập một kết nối ngang hàng peer to peer cho các file phương tiện

1.3.2. NodeMailer

Nodemailer là một thư viện mã nguồn mở được viết bằng Node.js cho phép gửi email từ máy chủ Node. Nó cung cấp các phương thức gửi email đơn giản, nhanh chóng và dễ sử dụng với nhiều cấu hình khác nhau. Điều này giúp cho các nhà phát triển web có thể tích hợp chức năng gửi email vào ứng dụng của họ một cách dễ dàng.

Nodemailer cung cấp cho người dùng khả năng gửi email thông qua các giao thức như SMTP, IMAP và POP3, cũng như hỗ trợ đính kèm file và xác thực email bằng OAuth2 và SSL/TLS.

Điểm mạnh của NodeMailer là nó cho phép người dùng tùy chỉnh và cấu hình email để phù hợp với nhu cầu sử dụng của họ. Với NodeMailer, người dùng có thể tùy chỉnh nội dung email, chủ đề, địa chỉ email người gửi và địa chỉ email người nhận. Ngoài ra, Nodemailer cũng cung cấp một bộ lọc để ngăn chặn các email spam.

Trong lĩnh vực lập trình web, NodeMailer được sử dụng phổ biến để gửi email xác nhận đăng ký tài khoản, thông báo đặt hàng, gửi mật khẩu mới và nhiều hơn nữa. Nó là một phần quan trọng trong các ứng dụng web hiện đại và giúp tăng tính bảo mật và tính tương tác của chúng.

CHƯƠNG 2: PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG

1.1. Phân tích và thiết kế hệ thống

1.1.1. Phân rã chức năng

- Xác thực:
 - Đăng nhập
 - Đăng ký
 - Khôi phục mật khẩu
 - Đăng xuất
- Quản lý tài khoản:
 - Chính sửa thông tin cá nhân
- Quản lý đề tài, bài học, câu hỏi
 - Thêm đề tài/ bài học/ câu hỏi
 - Xóa đề tài/ bài học/ câu hỏi
 - Sửa đề tài/ bài học/ câu hỏi
- Quản lý bài đăng
 - Thêm/ Xóa/ Sửa bài đăng
 - Bình luận bài đăng
- Quản lý bình luận
 - Xóa bình luận
- Video Call
 - Gọi video call
 - Chia sẻ màn hình
 - Nhấn tin

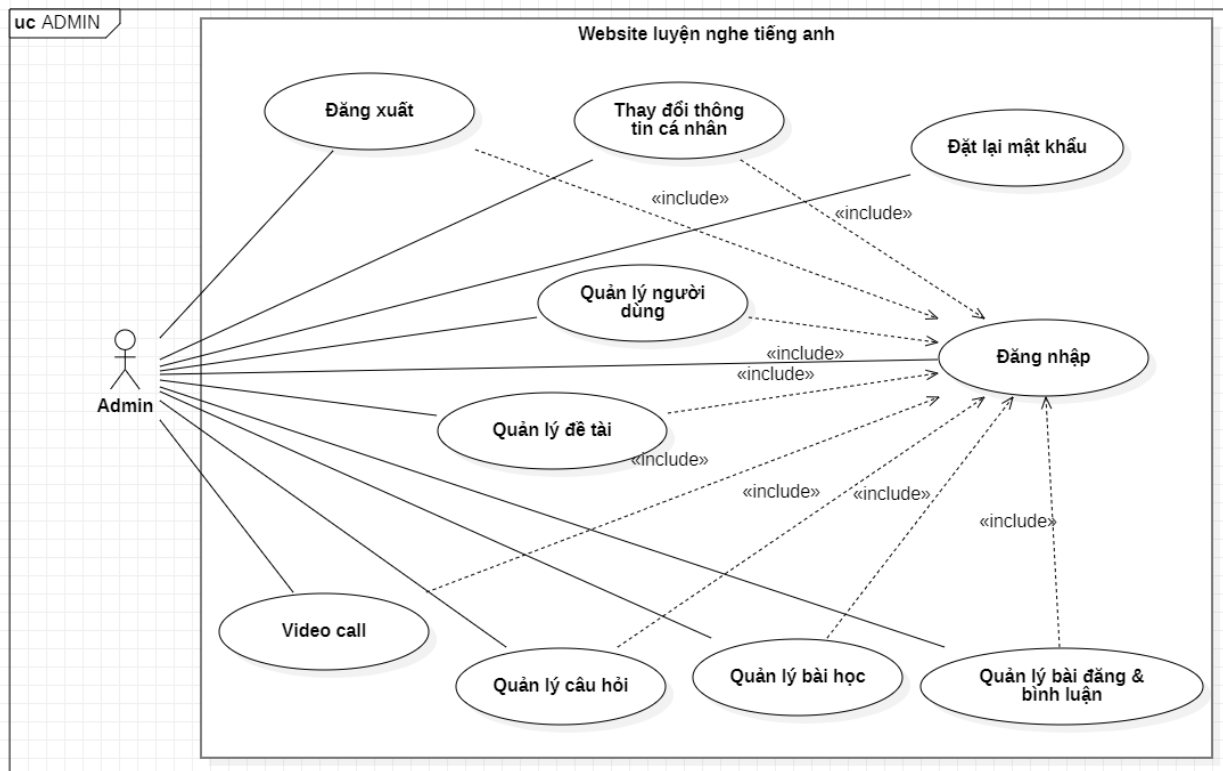
1.1.2. Các tác nhân của hệ thống

Tác nhân	Mô tả
----------	-------

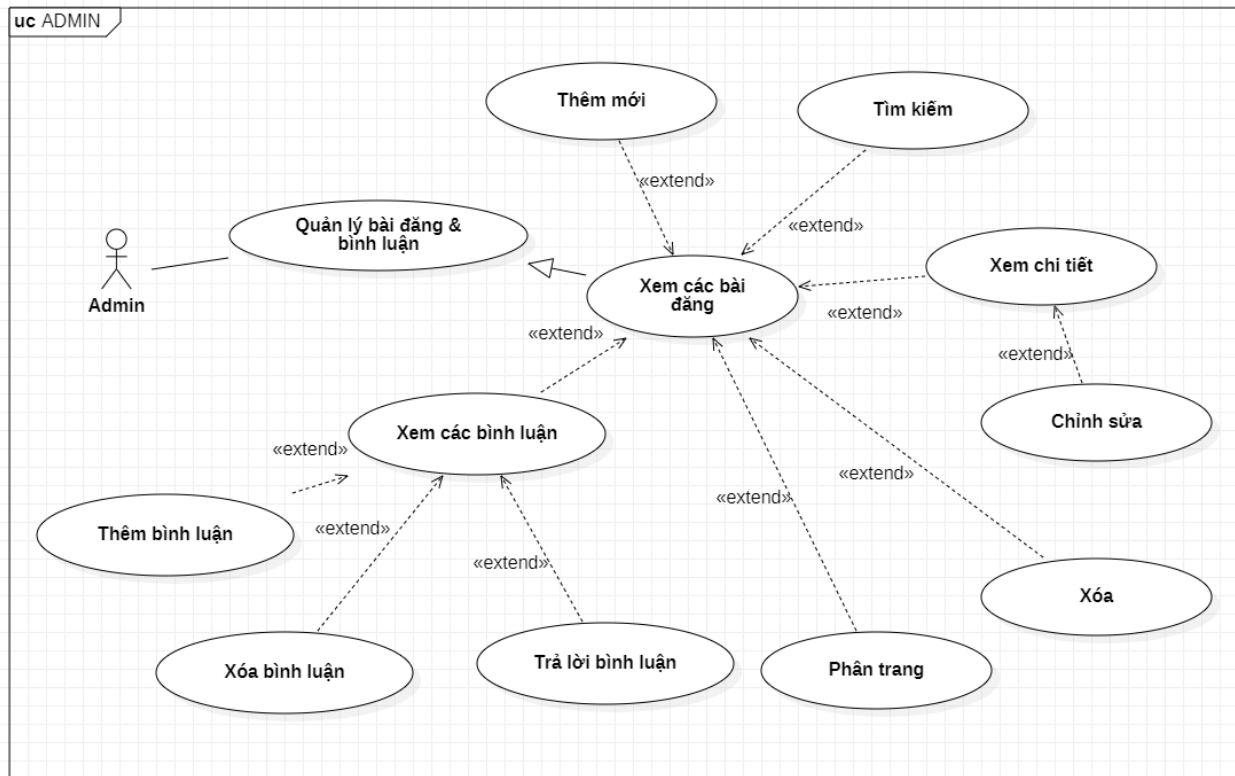
Admin (Người quản trị)	Có đầy đủ tất cả các tính năng của hệ thống như trong phần phân rã chức năng đã nêu trên
User (Người dùng)	Có thể đăng bài, chỉnh sửa bài viết của bản thân, bình luận, xem các chủ đề, bài học, làm các bài có trên hệ thống, có thể gọi video call, chỉnh sửa thông tin cá nhân.

1.1.3. Biểu đồ UseCase

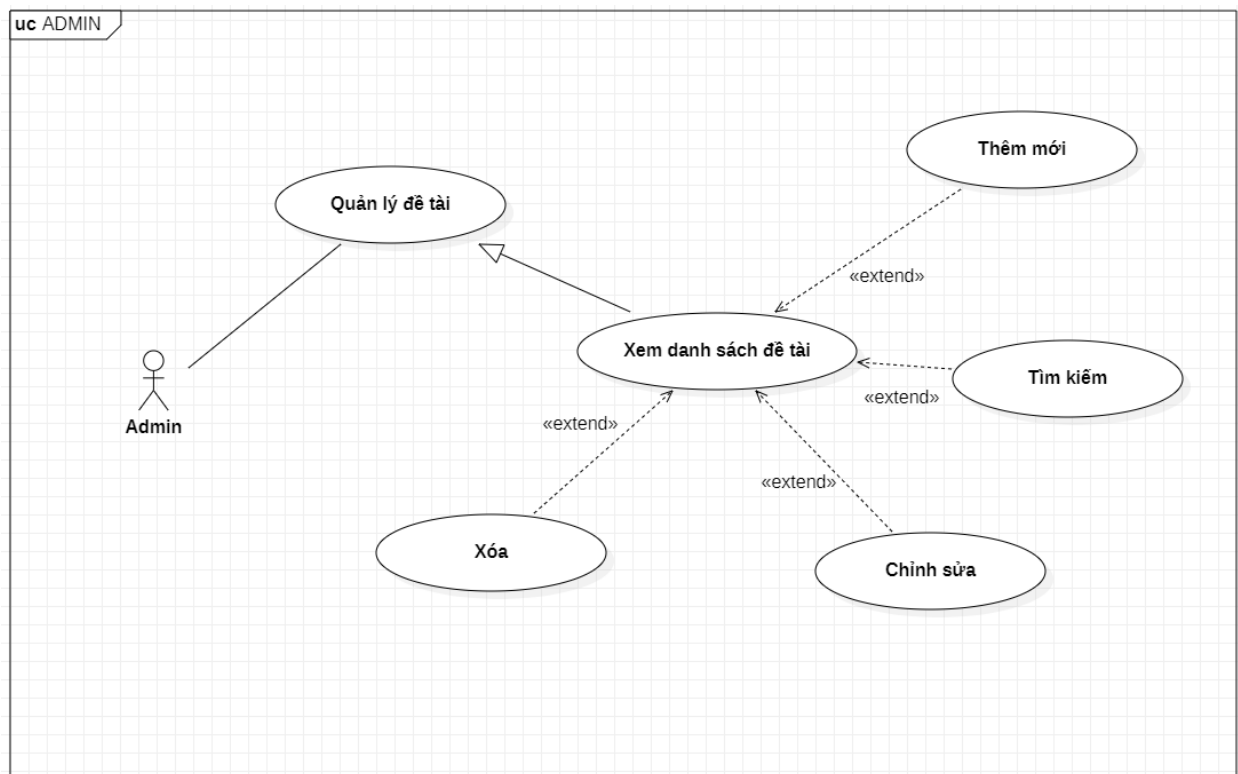
1.1.3.1. Quản trị viên (Admin)



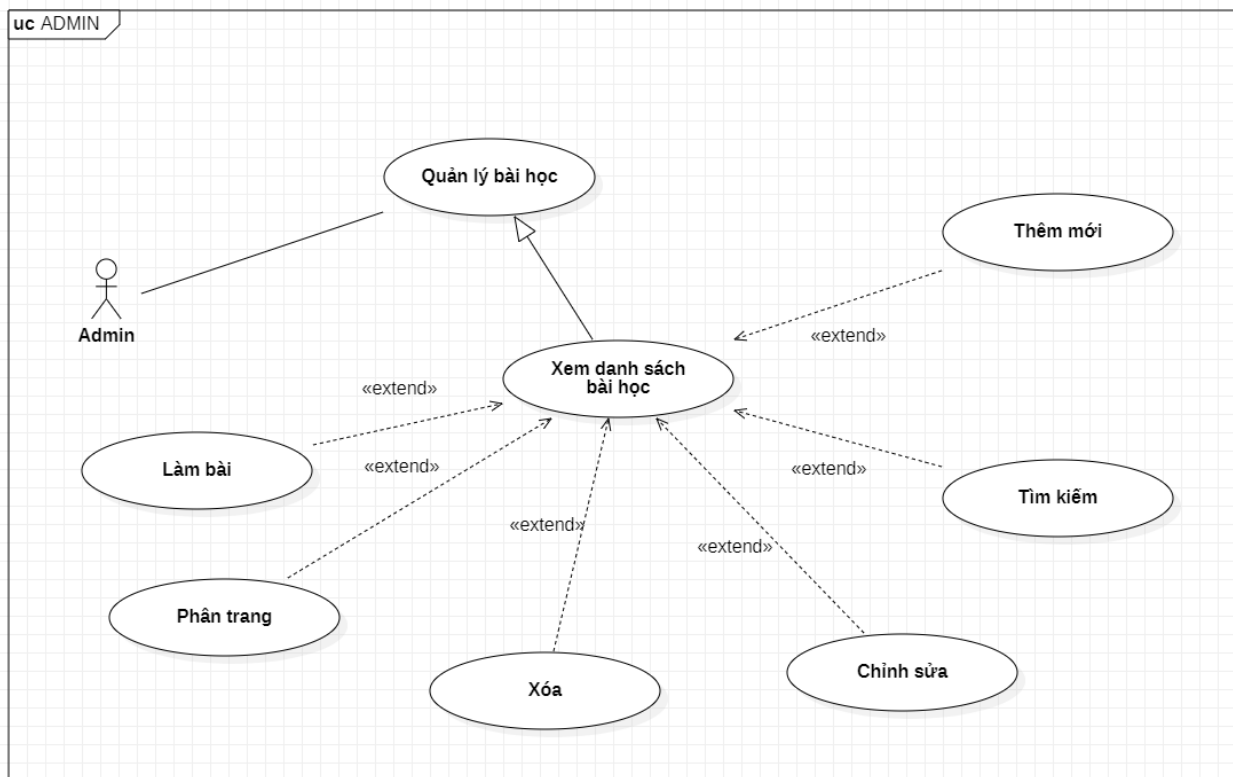
Hình 4: Biểu đồ Usecase - chức năng tổng quát của Admin



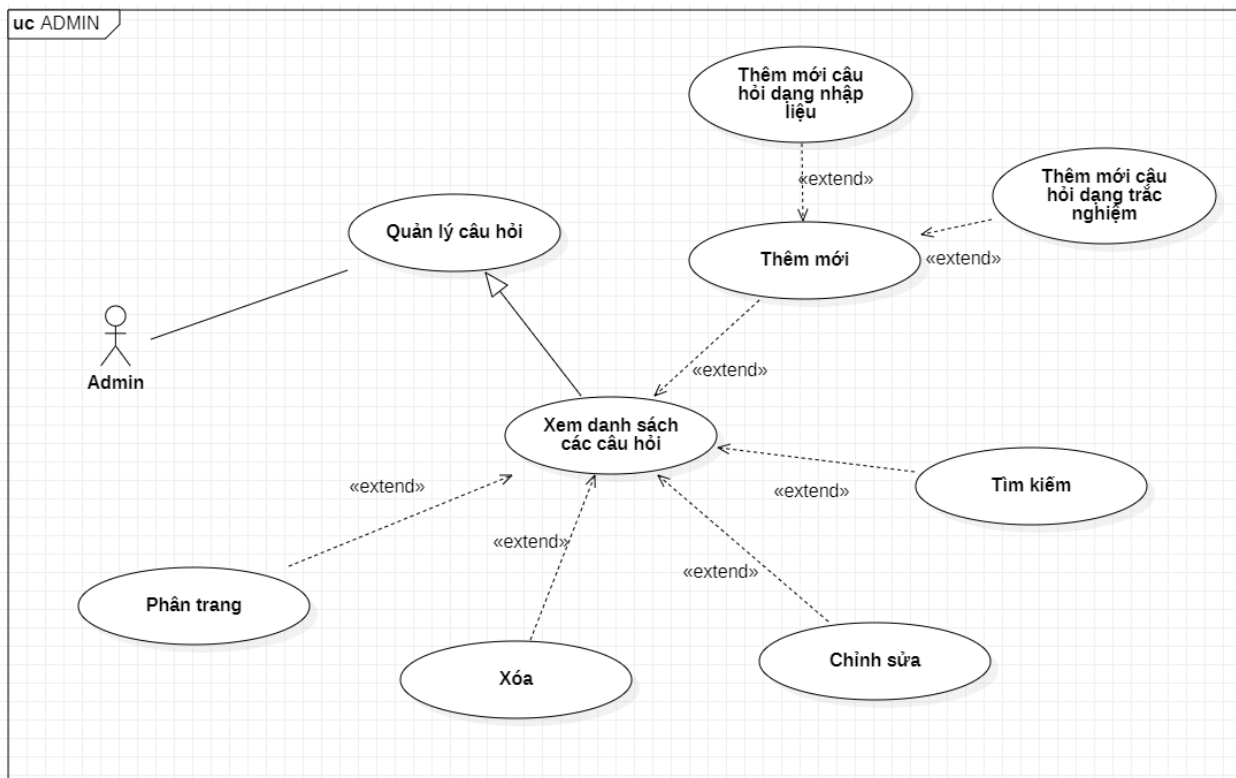
Hình 5: Biểu đồ Usecase - chức năng quản lý bài đăng, bình luận



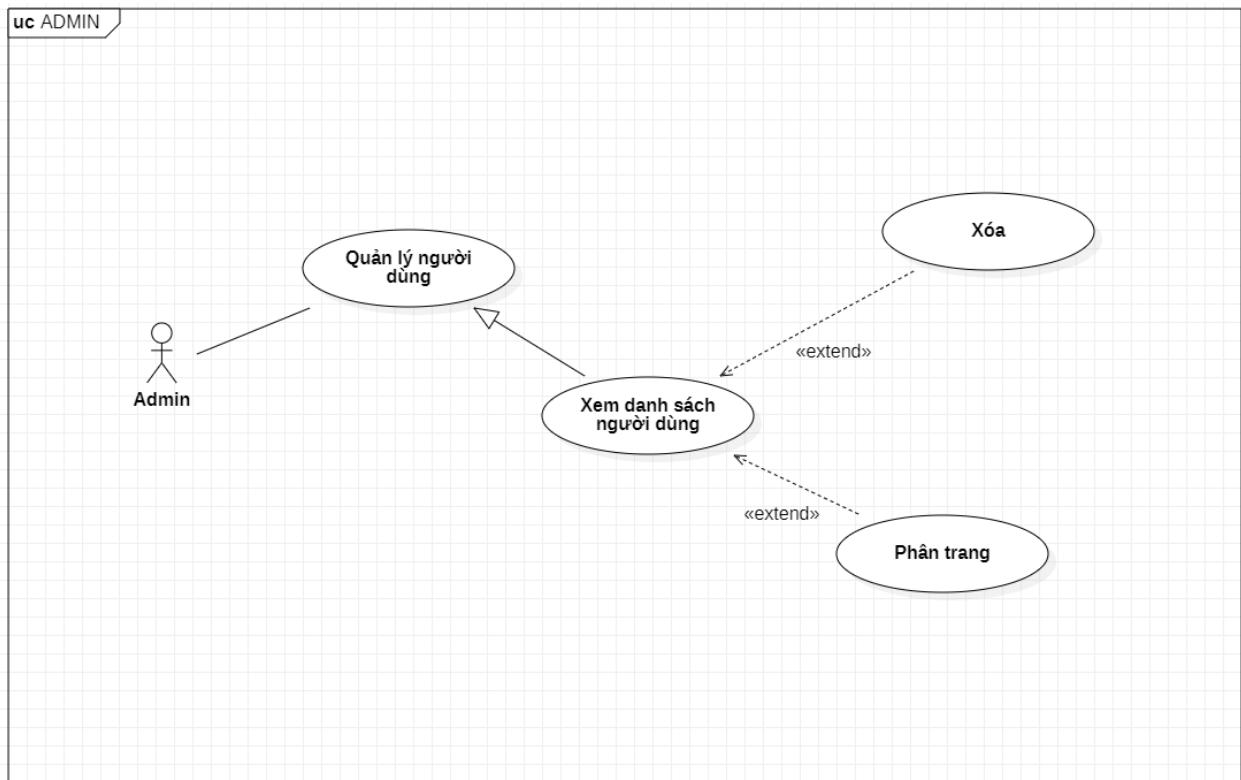
Hình 6: Biểu đồ Usecase - chức năng quản lý đề tài



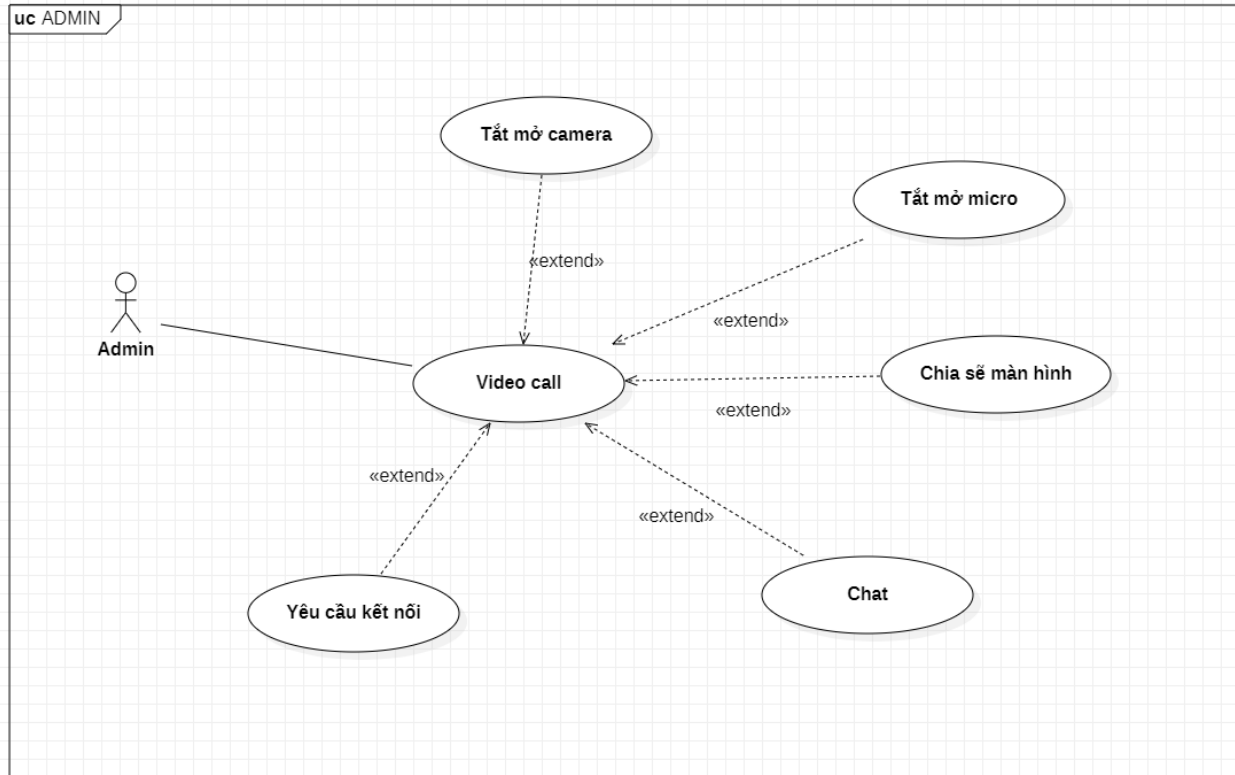
Hình 7: Biểu đồ Usecase - chức năng quản lý bài học



Hình 8: Biểu đồ Usecase - chức năng quản lý câu hỏi

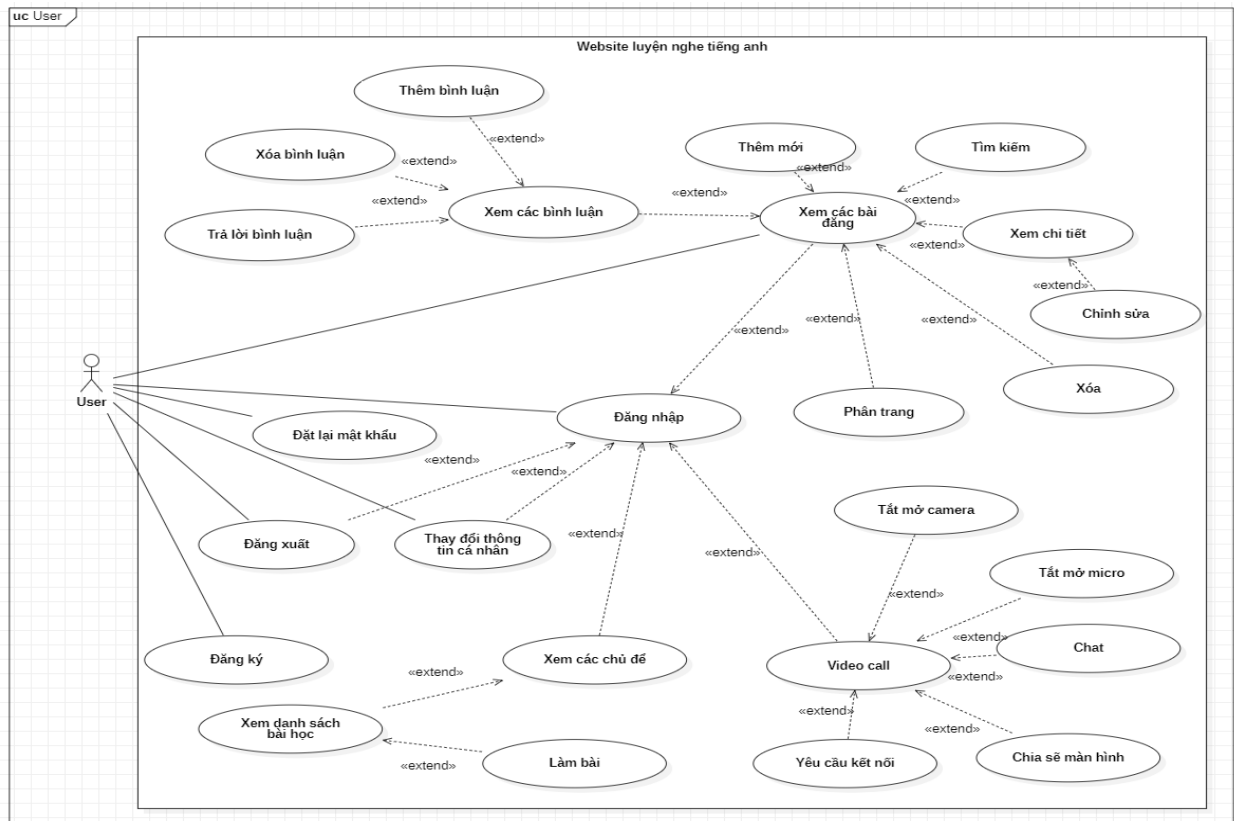


Hình 9: Biểu đồ Usecase - chức năng quản lý người dùng



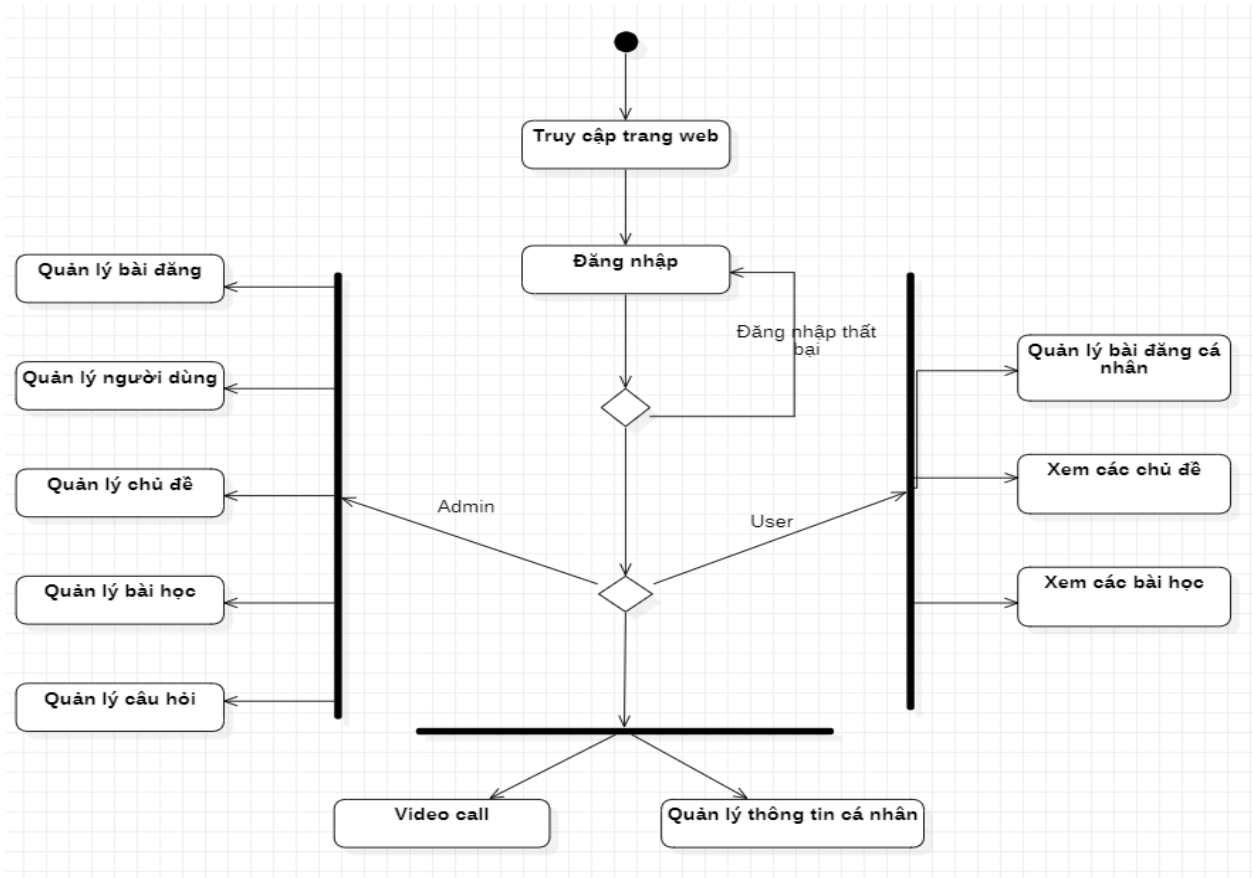
Hình 10: Biểu đồ Usecase - chức năng video call

1.1.3.2. Người dùng (User)



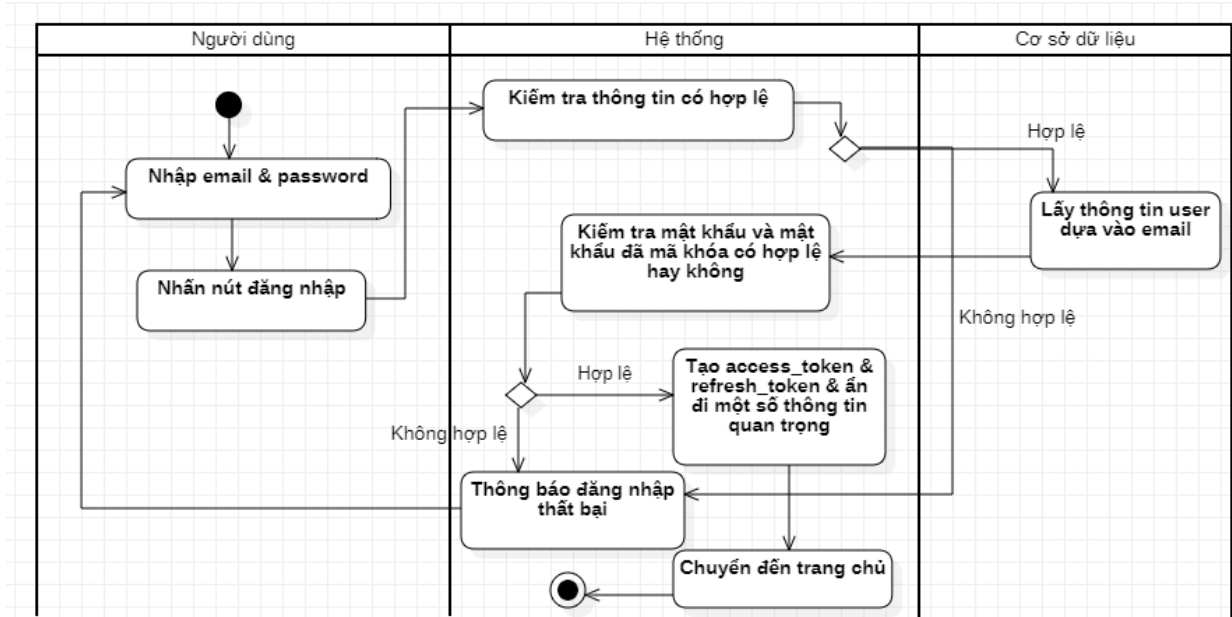
Hình 11: Biểu đồ Usecase - chức năng tổng quát của User (người dùng)

1.1.4. Biểu đồ hoạt động



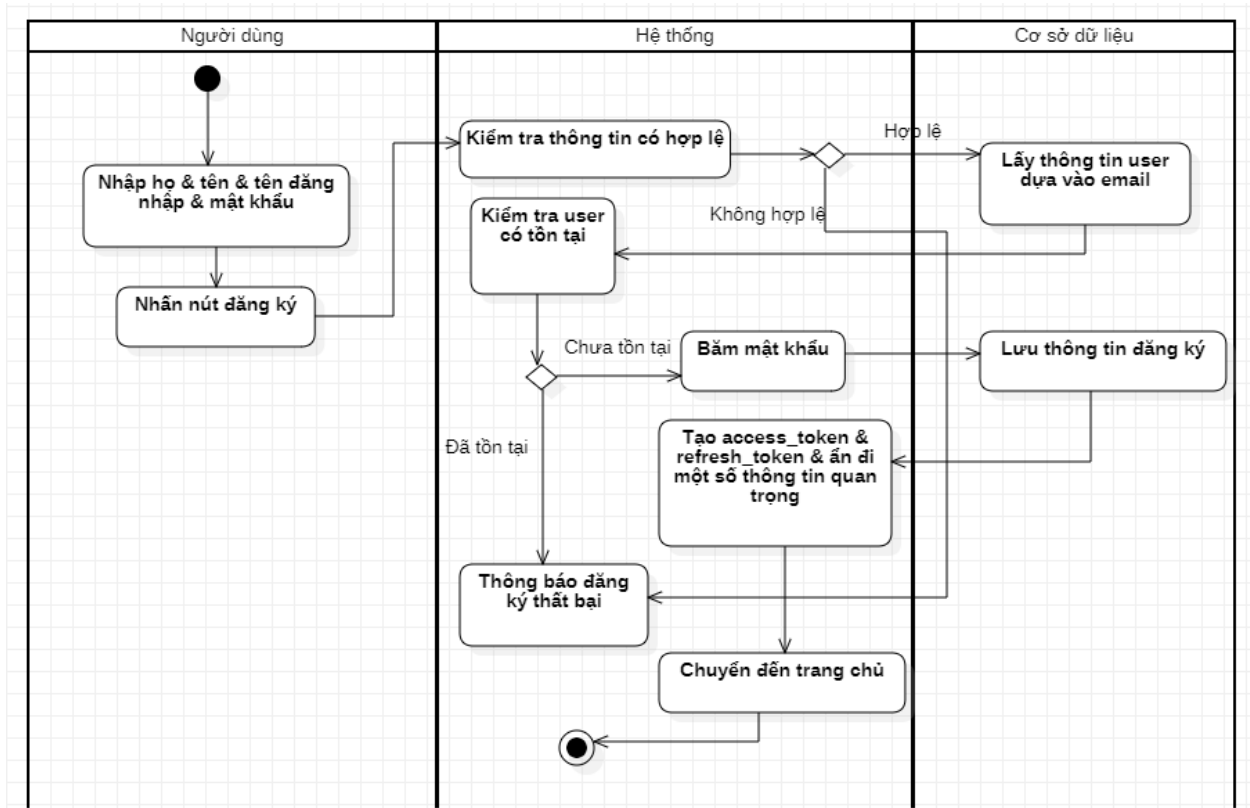
Hình 12: Biểu đồ hoạt động của hệ thống

1.1.4.1. Đăng nhập



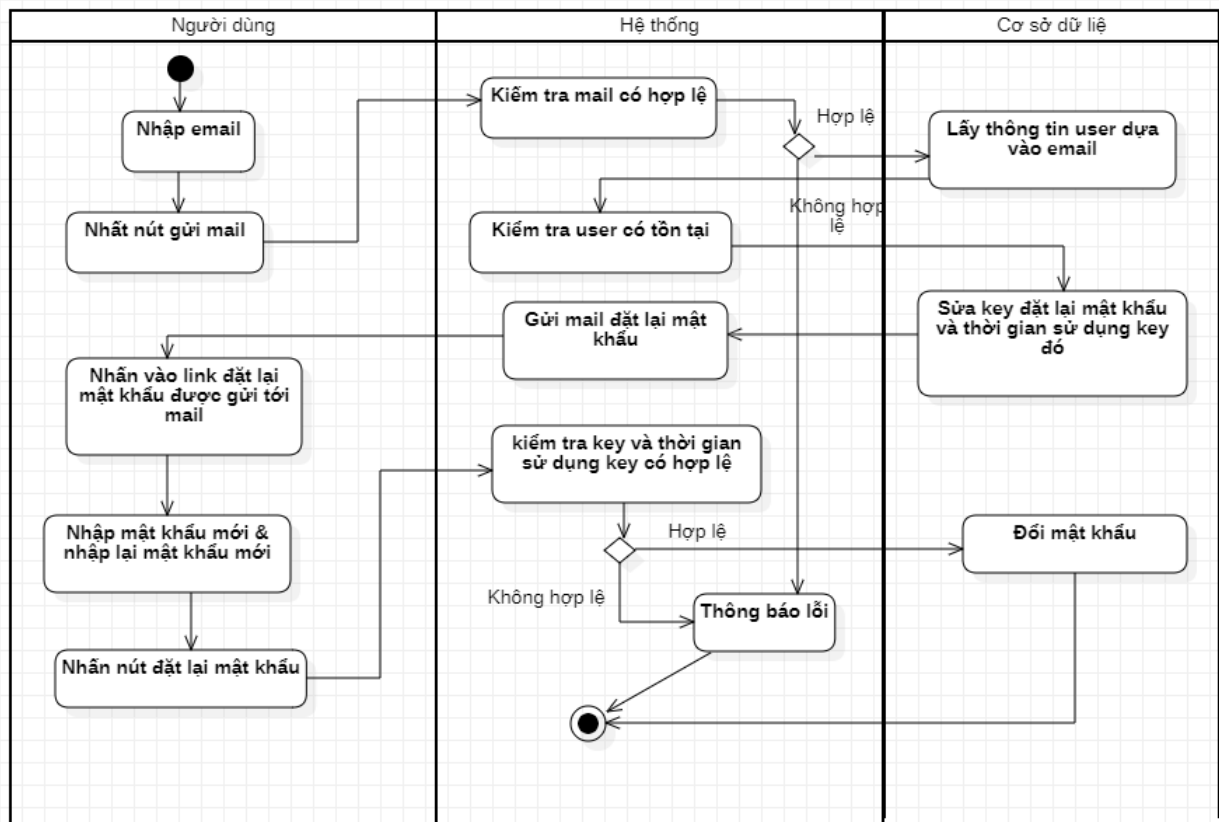
Hình 13: Biểu đồ hoạt động - đăng nhập

1.1.4.2. Đăng ký



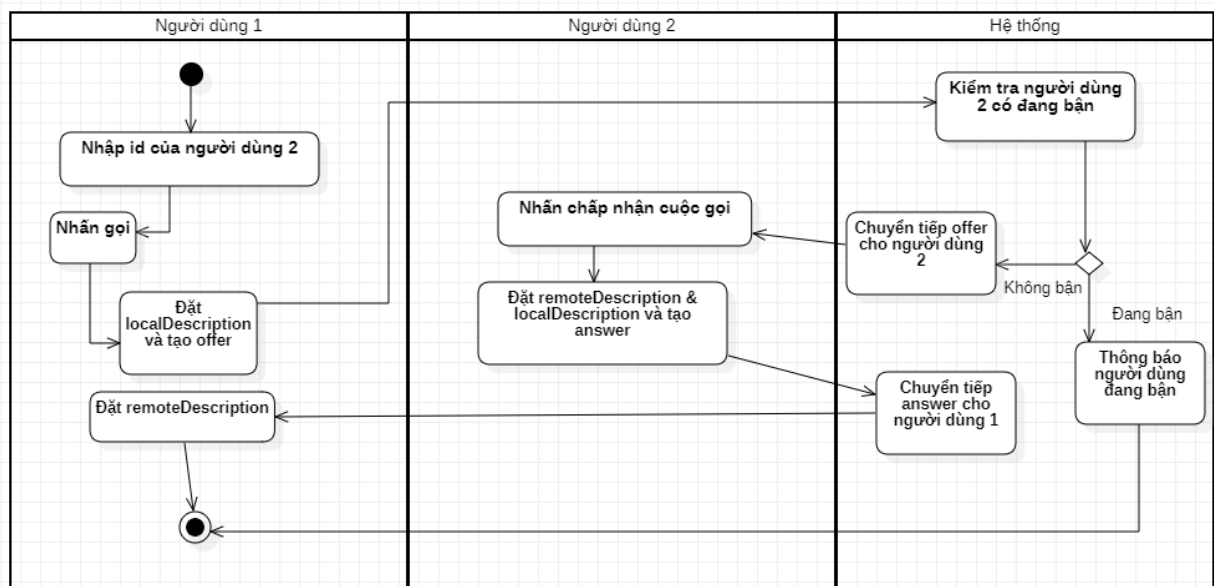
Hình 14: Biểu đồ hoạt động - đăng ký

1.1.4.3. Đặt lại mật khẩu



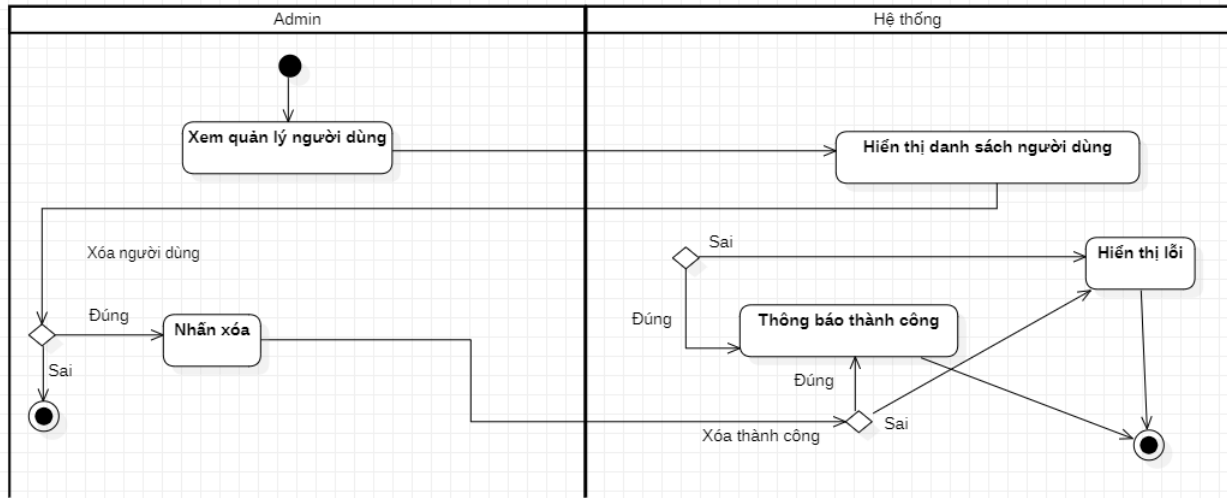
Hình 15: Biểu đồ hoạt động - đặt lại mật khẩu

1.1.4.4. Video Call



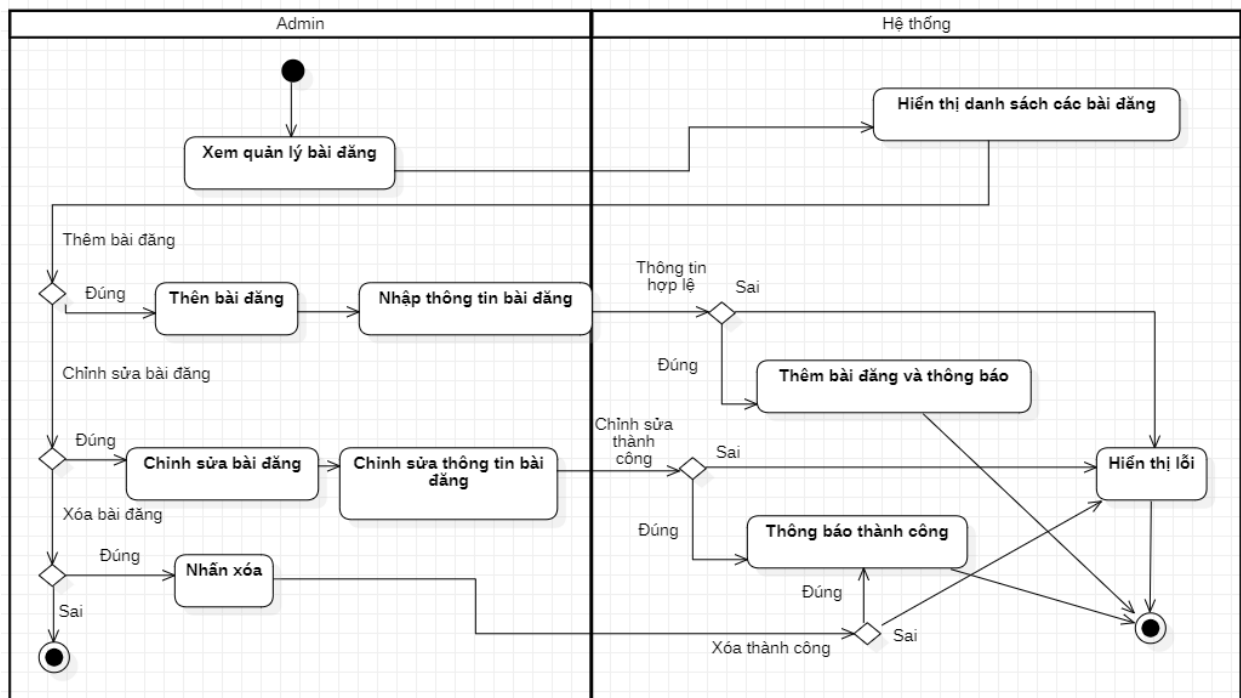
Hình 16: Biểu đồ hoạt động - video call

1.1.4.5. Quản lý người dùng



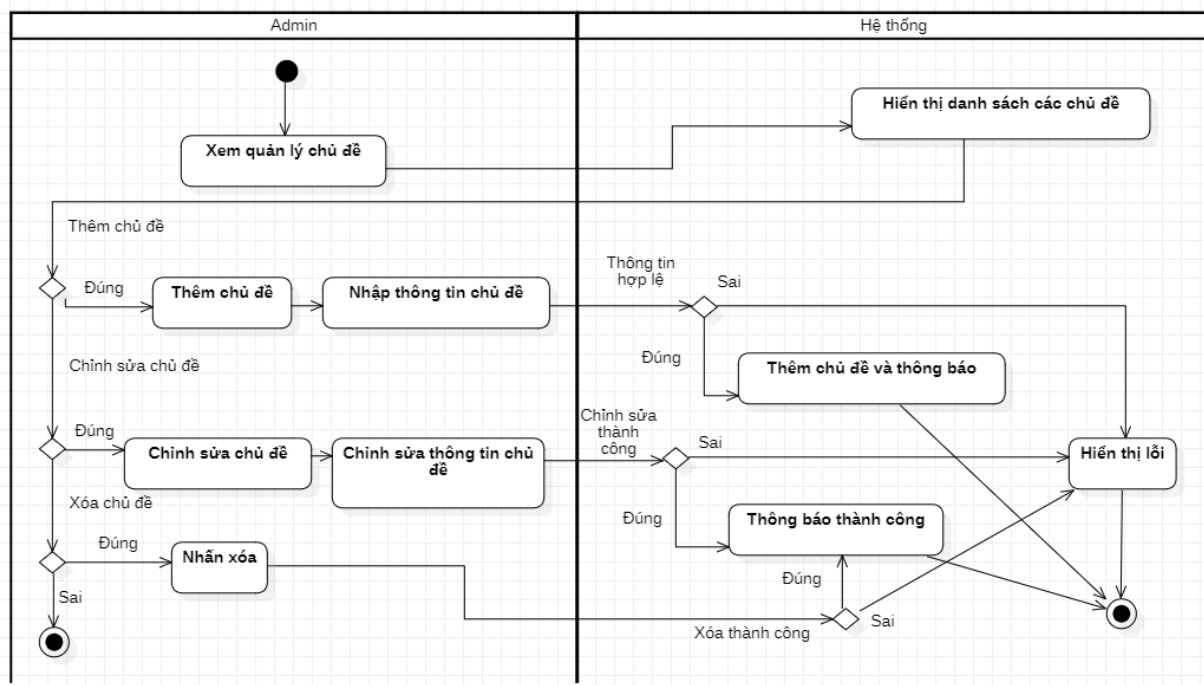
Hình 17: Biểu đồ hoạt động – quản lý người dùng

1.1.4.6. Quản lý bài đăng



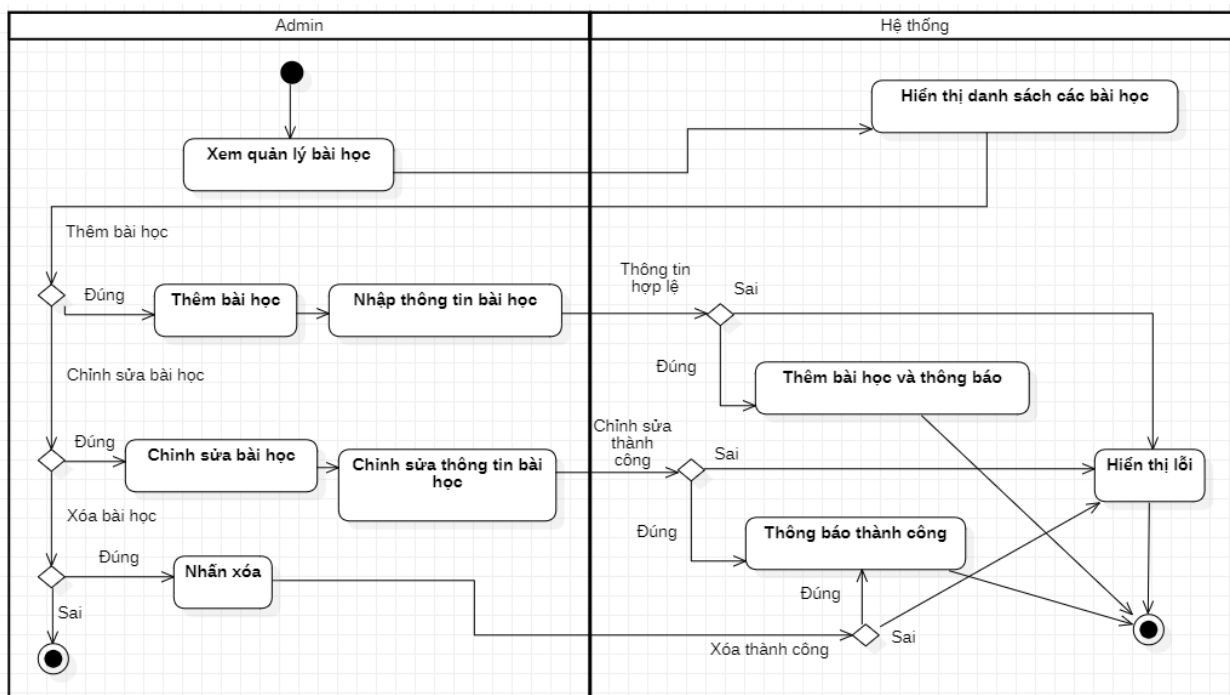
Hình 18: Biểu đồ hoạt động – quản lý bài đăng

1.1.4.7. Quản lý chủ đề



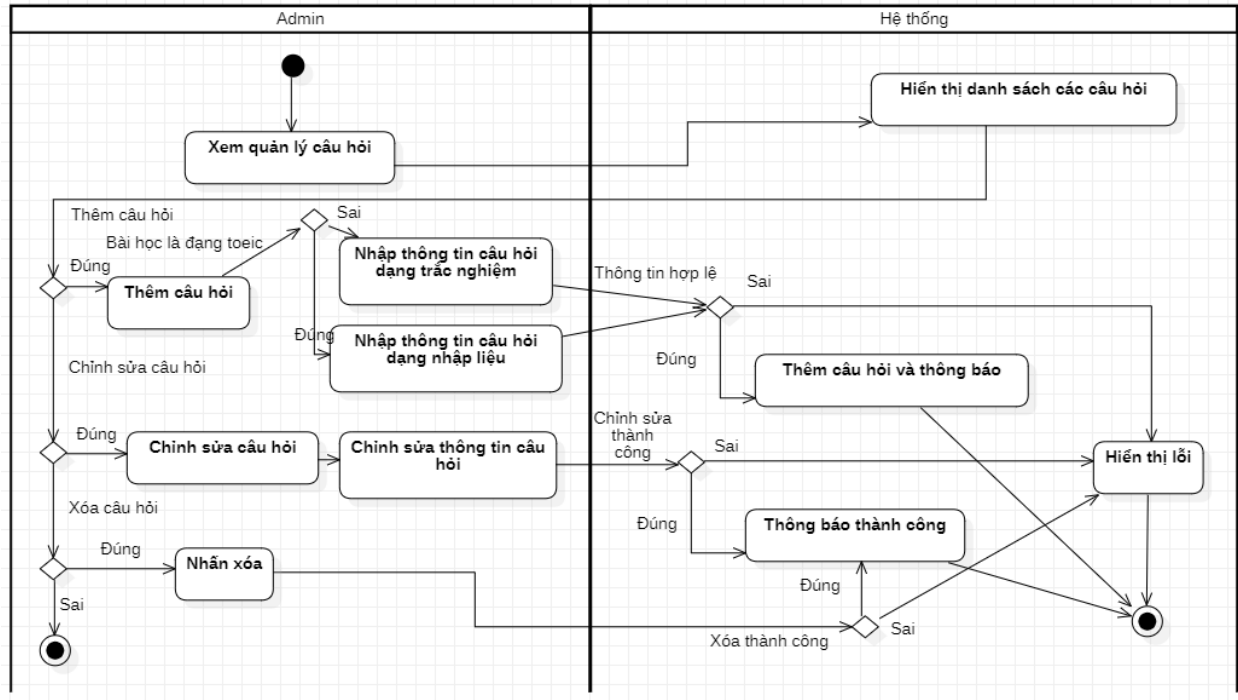
Hình 19: Biểu đồ hoạt động – quản lý chủ đề

1.1.4.8. Quản lý bài học



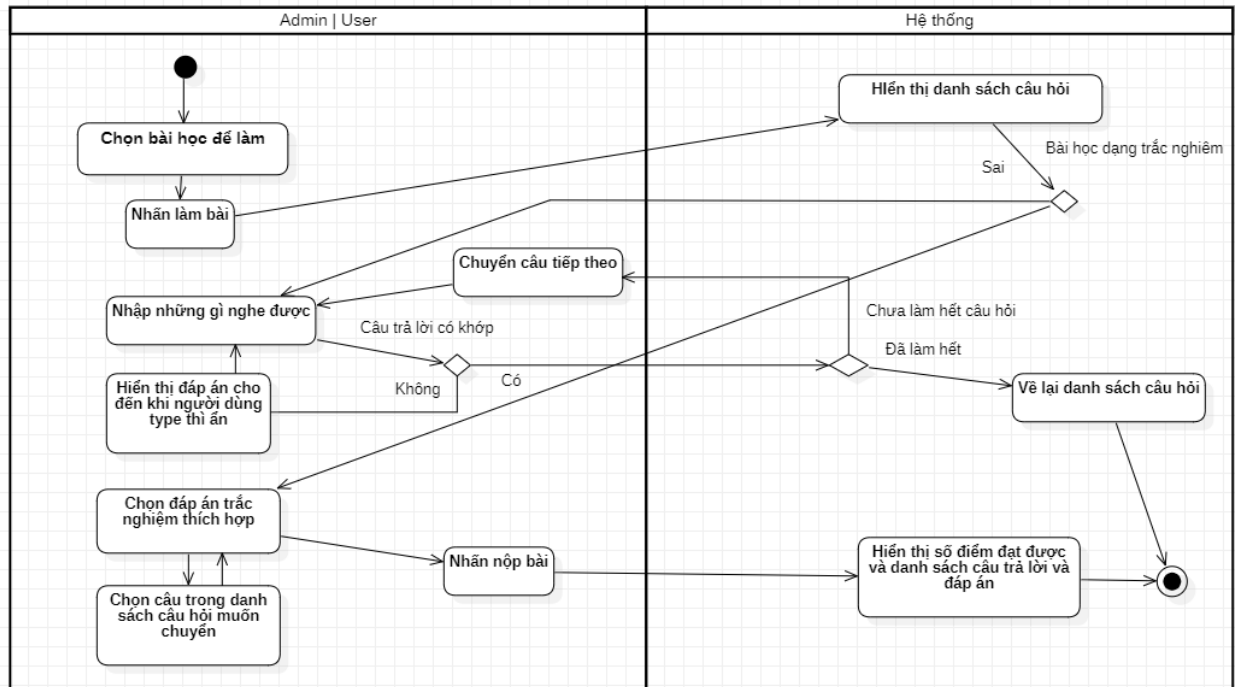
Hình 20: Biểu đồ hoạt động – quản lý bài học

1.1.4.9. Quản lý câu hỏi



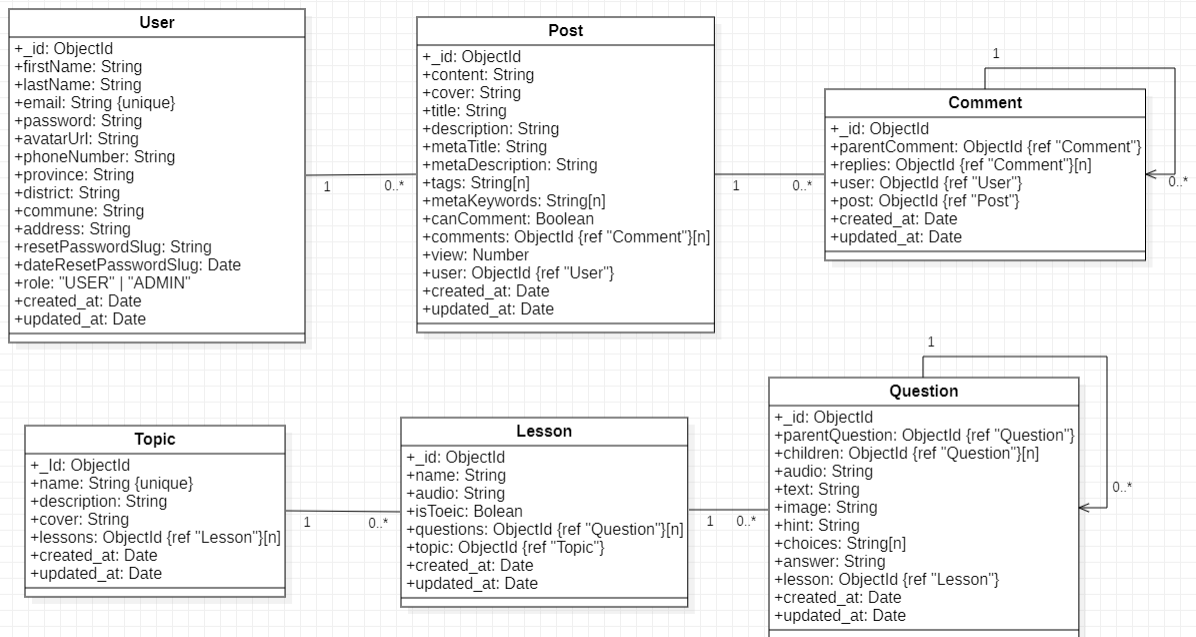
Hình 21: Biểu đồ hoạt động – quản lý câu hỏi

1.1.4.10. Làm bài



Hình 22: Biểu đồ hoạt động – làm bài

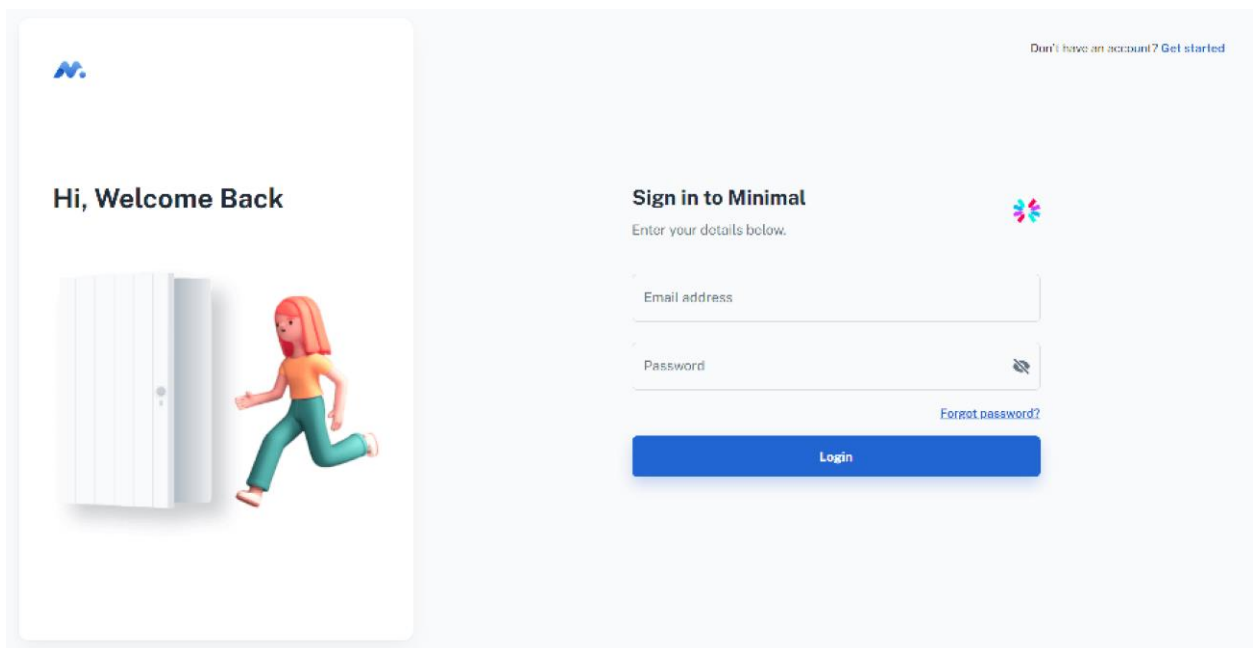
1.1.5. Biểu đồ lớp



Hình 23: Biểu đồ lớp

1.2. Thiết kế giao diện

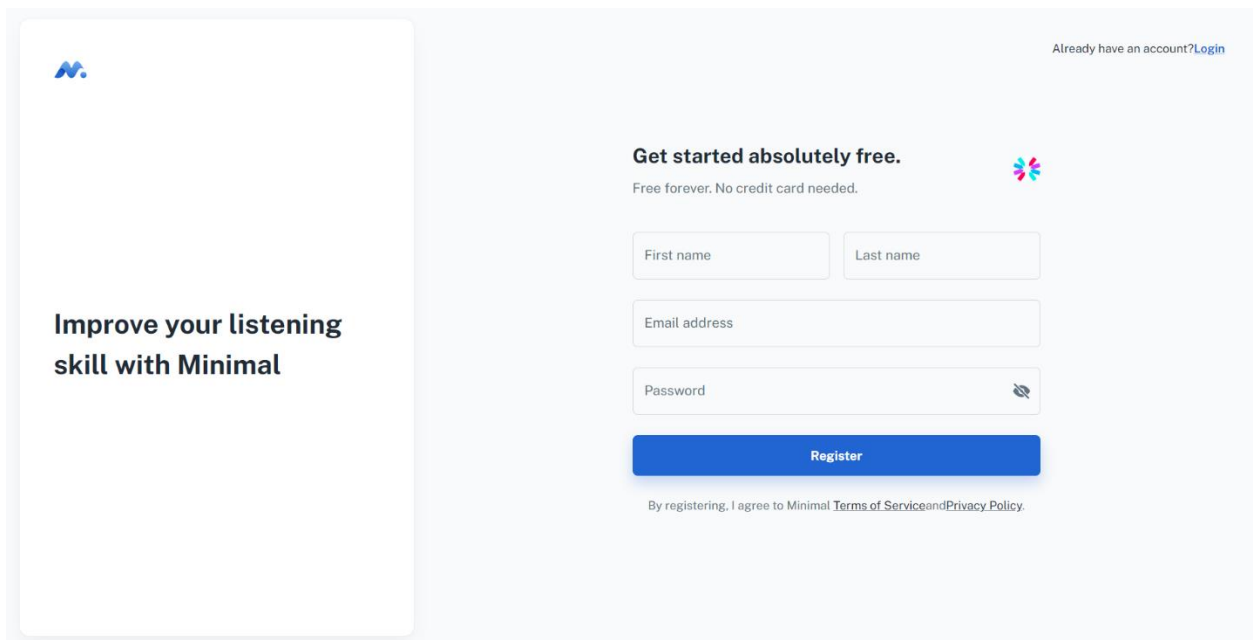
1.2.1. Giao diện đăng nhập



The login interface is divided into two main sections. On the left, a white card features the Minimal logo at the top, followed by the text "Hi, Welcome Back". Below this is an illustration of a woman with red hair, wearing an orange shirt and green pants, running towards a white door. On the right, a light blue background contains the heading "Sign in to Minimal" and the instruction "Enter your details below.". There are two input fields: "Email address" and "Password" (with a toggle icon). A "Forgot password?" link is positioned below the password field. A prominent blue "Login" button is at the bottom.

Hình 24: Giao diện – đăng nhập

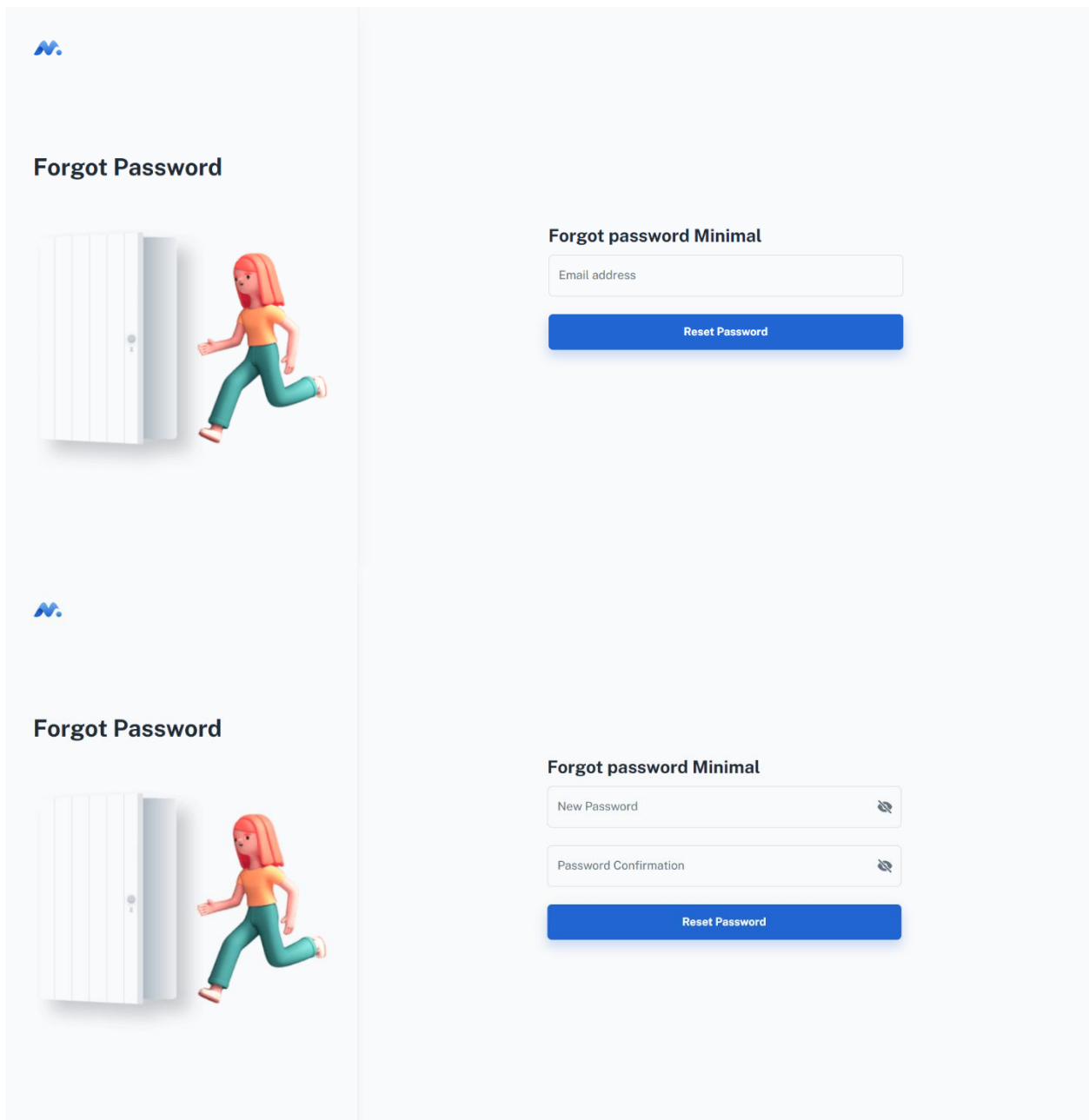
1.2.2. Giao diện đăng ký



The registration interface is divided into two main sections. On the left, a white card features the Minimal logo at the top, followed by the text "Improve your listening skill with Minimal". On the right, a light blue background contains the heading "Get started absolutely free." and the instruction "Free forever. No credit card needed.". There are four input fields: "First name", "Last name", "Email address", and "Password" (with a toggle icon). A prominent blue "Register" button is at the bottom. Below the button, a small line of text states: "By registering, I agree to Minimal Terms of Service and Privacy Policy."

Hình 25: Giao diện – đăng ký

1.2.3. Giao diện đặt lại mật khẩu

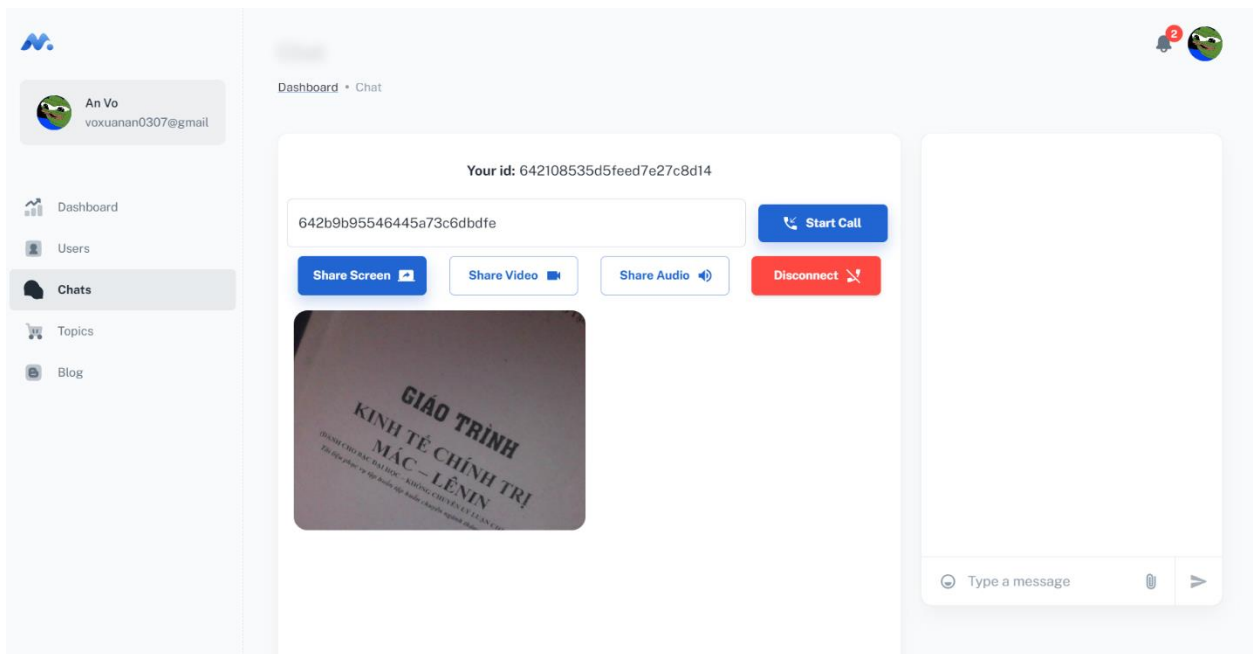


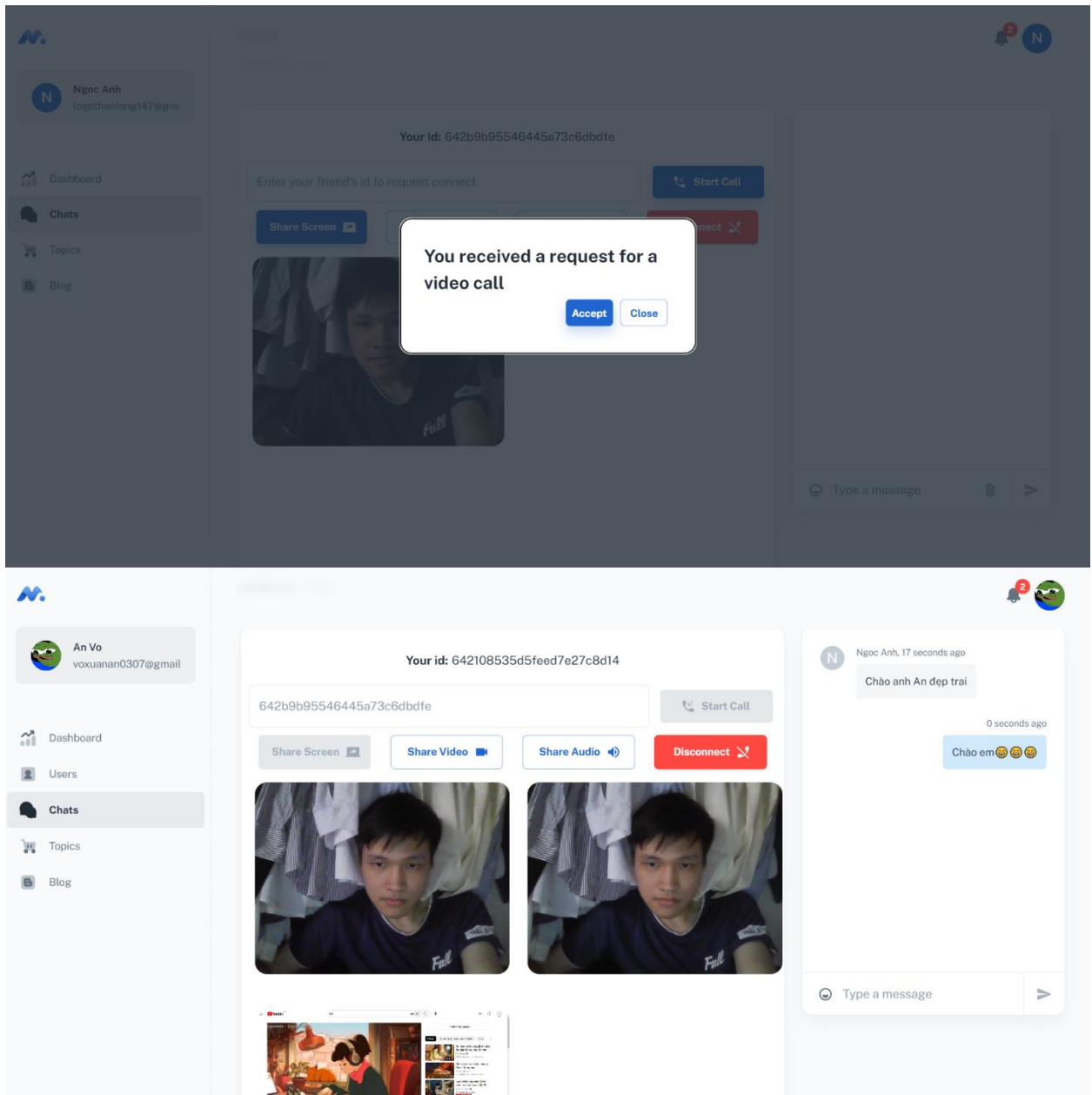
Hình 26-27: Giao diện – đăng nhập

1.2.4. Giao diện thay đổi thông tin cá nhân

Hình 28: Giao diện – thay đổi thông tin cá nhân

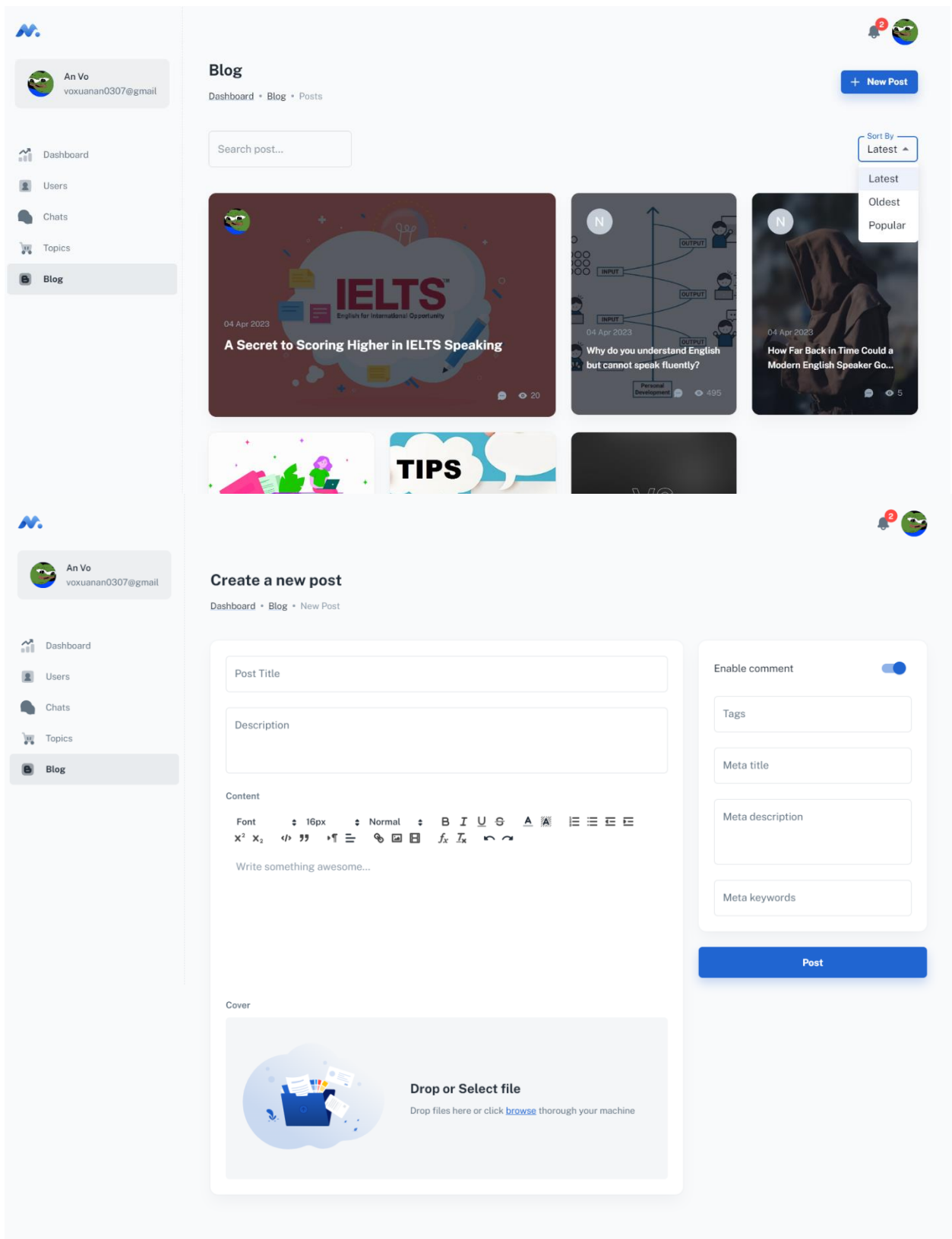
1.2.5. Giao diện video call





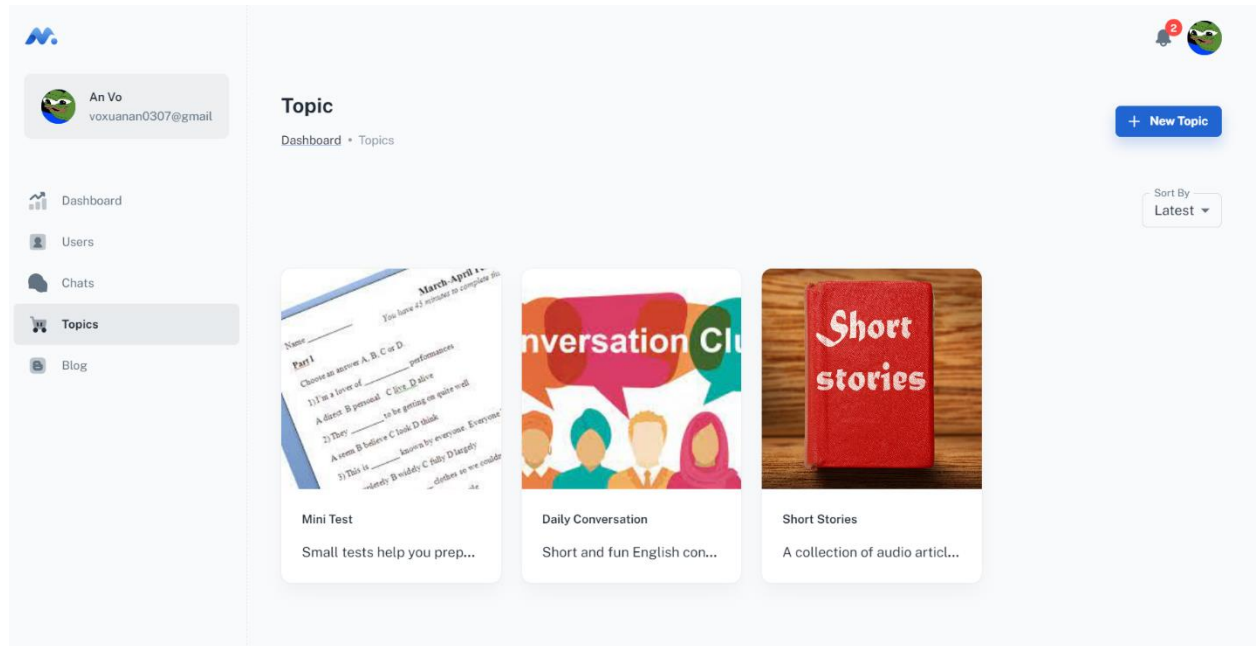
Hình 29-31: Giao diện – video call

1.2.6. Giao diện bài đăng



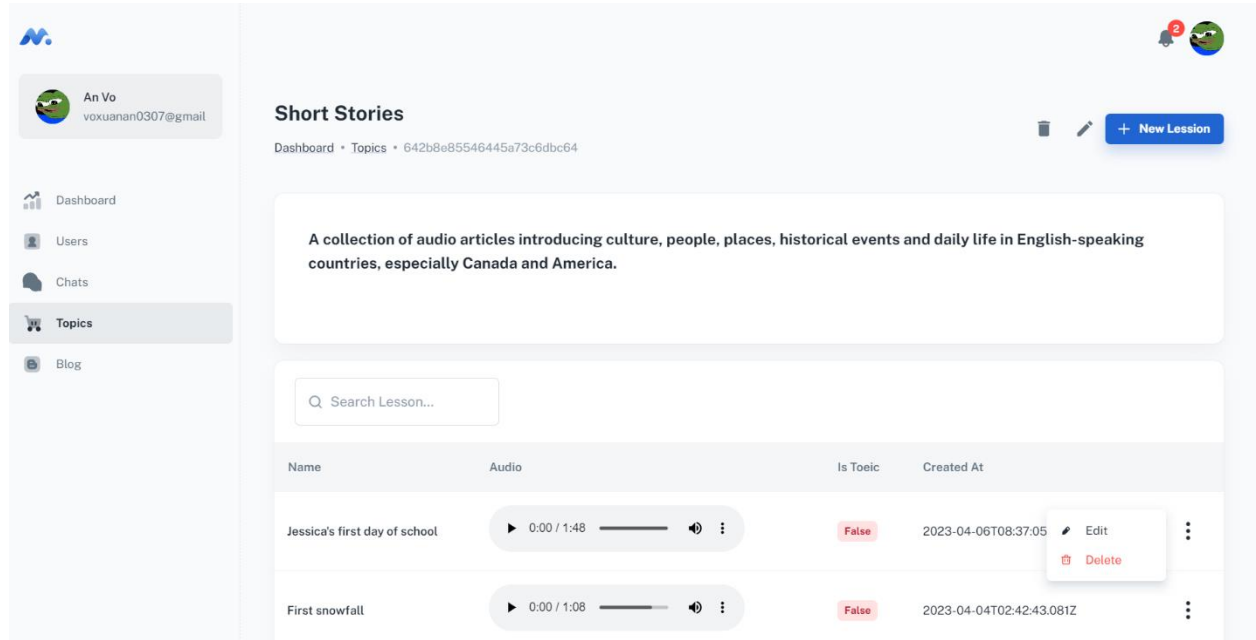
Hình 32-33: Giao diện – bài đăng

1.2.7. Giao diện chủ đề




Hình 34: Giao diện – chủ đề


1.2.8. Giao diện bài học



Hình 35: Giao diện – bài học

1.2.9. Giao diện làm bài





An Vo

voxuanan0307@gmail

Dashboard

Users

Chats

Topics


Blog


First snowfall

Dashboard • Topics • Lesson • 642b8ea3546445a73c6dbc6b

← 10/20 →

▶ 0:00 / 0:03





Answer


i put on my batman and mitten


Skip

Check


▲ Correct Answer:


i put on my (b)hat(man) and mittens



 Pronunciation ⓘ

Click Here to show





An Vo

voxuanan0307@gmail

Dashboard

Users


Chats

Topics


Blog


TOEIC mini test (1)

Dashboard • Topics • Lesson • 642ba04b546445a73c6dbe8c



▶ 0:00 / 0:19





☐ (A)

☒ (B)

☐ (C)

☐ (D)

Clear

Submit Answer

Parent Question Info

Question Palette

← 3/20 →

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

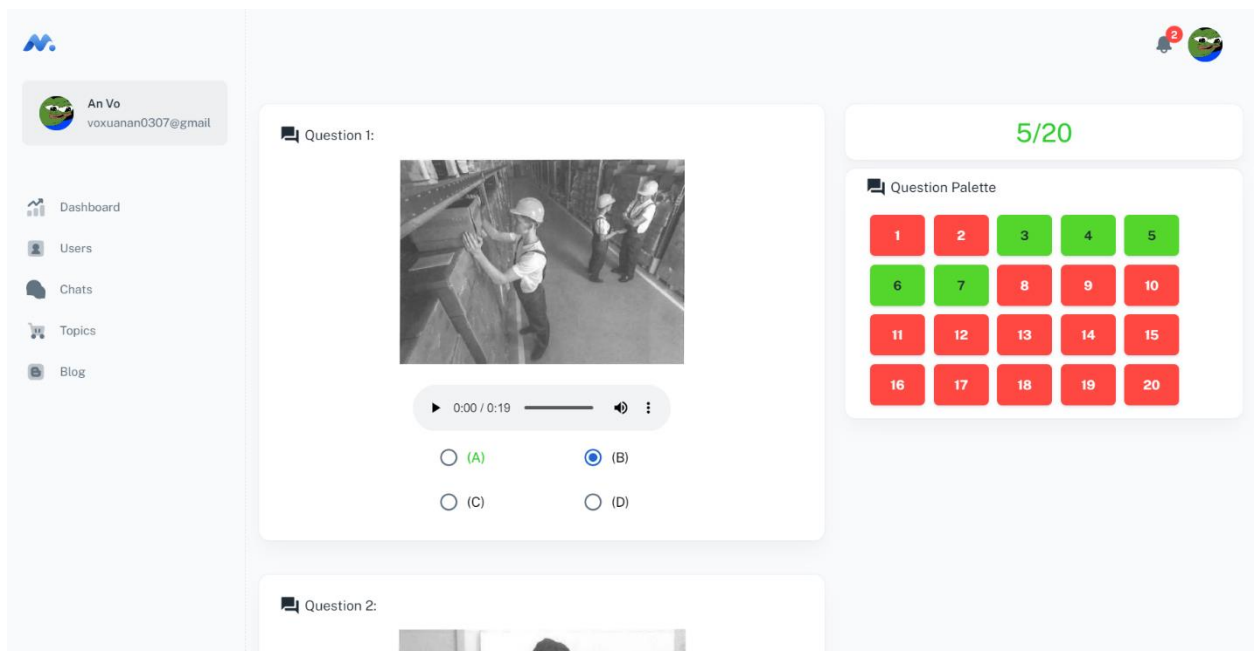
17

18

19

20

36



Hình 36-38: Giao diện – làm bài

1.3. Hoàn thiện sản phẩm

1.3.1. Đánh giá mức độ hoàn thiện

Sản phẩm đã hoàn thiện đầy đủ một số chức năng sau: Đăng ký, đăng nhập, đặt lại mật khẩu, các phân quản lý như quản lý người dùng, quản lý đề tài, quản lý bài học, quản lý câu hỏi, quản lý bài đăng, quản lý bình luận, và phần video call

Tuy nhiên vẫn còn tồn tại một số lỗi như khi đang video call có chia sẻ màn hình và disconnect dẫn tới frame chia sẻ màn hình bị đen, mặc dù không gây ảnh hưởng nhiều nhưng cần xử lý để sản phẩm hoàn thiện hơn

1.3.2. Hướng dẫn cài đặt sản phẩm

a. Môi trường cài đặt

Sản phẩm được phát triển trên nền tảng web, với server – backend sử dụng ngôn ngữ Node và front- end sử dụng React nên các môi trường cài đặt cần thiết như sau: Node, Npm, MongoDB (có thể dùng mongo atlas để thay thế)

b. Tiến hành cài đặt và chạy ứng dụng

Sau khi cài đặt môi trường và có mã nguồn chương trình, ta tiến hành cài đặt như sau:

- Front-end:
 - Mở terminal và nhập command “npm install” để cài đặt các package cần thiết
 - Nhập “npm run start” để chạy front-end
- Back-end:
 - Copy file .env.example thành file .env và thay đổi một số thông tin nếu cần thiết
 - Mở terminal và nhập command “npm install” để cài đặt các package cần thiết
 - Nhập command “npm run dev” để chạy môi trường development

PHẦN KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

1. Kết luận

- Nắm bắt được các kiến thức cơ bản của Express, React
- Xây dựng được website luyện nghe tiếng anh bằng Express, React
- Trang bị thêm được khả năng phân tích, xây dựng và phát triển một hệ thống phần mềm có tính thực tế.
- Có thêm kinh nghiệm trong việc sử dụng các ngôn ngữ lập trình để phát triển ứng dụng.
- Tìm hiểu và áp dụng các ngôn ngữ, thư viện mới, phù hợp với dự án

2. Hướng phát triển

- Làm phần dashboard cho ứng dụng
- Lưu thông tin bài làm của người dùng thành một bảng lịch sử làm bài, thêm trường thời gian để các câu hỏi dạng trắc nghiệm được trong một khoảng thời gian nhất định và cache câu trả lời để khi có sự cố về đường truyền vẫn có thể tiếp tục làm bài
- Làm thêm bảng vocabulary lưu thông tin cách đọc (Giọng US, UK) nghĩa của từ vựng, cách dùng, khi dùng...
- Cache, lazy load ở phía front-end
- Cho phép truyền file, focus vào 1 frame trong chế độ video call
- Sử dụng các thẻ meta để config chia sẻ bài đăng (blog) lên các trang mạng xã hội

TÀI LIỆU THAM KHẢO

- [1]. Nguyễn Mậu Hân, Giáo trình Phân tích và thiết kế các hệ thống thông tin.
- [2]. Nguyễn Văn Ba (2005), *phát triển hệ thống hướng đối tượng với UML và C++*, Nhà xuất bản Đại học Quốc gia Hà Nội.
- [3]. Trần Thị Phương Loan (5/2022), Khóa luận tốt nghiệp đại học “Tìm hiểu Spring Boot và xây dựng website cho thuê nhà”, Đại học Khoa học, Đại học Huế
- [4]. Hussein Nasser. (2020). WebRTC Crash Course [Video]. YouTube
<https://www.youtube.com/watch?v=FExZvpVvYxA> truy cập ngày 20/5/2023
- [5]. What is WebRTC and When to Use it? Truy cập từ: <https://os-system.com/blog/what-is-webrtc-and-when-to-use-it/> truy cập ngày 20/5/2023