

ĐẠI HỌC ĐÀ NẴNG
TRƯỜNG ĐẠI HỌC BÁCH KHOA
KHOA CÔNG NGHỆ THÔNG TIN

ĐỒ ÁN TỐT NGHIỆP
NGÀNH: CÔNG NGHỆ THÔNG TIN
CHUYÊN NGÀNH: CÔNG NGHỆ PHẦN MỀM

ĐỀ TÀI:
XÂY DỰNG WEBSITE
QUIZ HỌC TẬP - PIKI

Người hướng dẫn: **TS. BÙI THỊ THANH THANH**
Sinh viên thực hiện: **PHAN TIẾN ĐẠT**
Số thẻ sinh viên: **102200014**
Lớp: **20T1**

Đà Nẵng, 06/2024

**ĐẠI HỌC ĐÀ NẴNG
TRƯỜNG ĐẠI HỌC BÁCH KHOA
KHOA CÔNG NGHỆ THÔNG TIN**

ĐỒ ÁN TỐT NGHIỆP
NGÀNH: CÔNG NGHỆ THÔNG TIN
CHUYÊN NGÀNH: CÔNG NGHỆ PHẦN MỀM

ĐỀ TÀI:
**XÂY DỰNG WEBSITE
QUIZ HỌC TẬP - PIKI**

Người hướng dẫn: **TS. BÙI THỊ THANH THANH**
Sinh viên thực hiện: **PHẦN TIẾN ĐẠT**
Số thẻ sinh viên: **102200014**
Lớp: **20T1**

Đà Nẵng, 06/2024

TÓM TẮT

Tên đề tài: Xây dựng website quiz học tập - Piki

Sinh viên thực hiện: Phan Tiến Đạt

Số thẻ SV: 102200014

Lớp: 20T1

Tóm tắt nội dung đồ án:

Hệ thống là một nền tảng trực tuyến cho việc học tập và kiểm tra kiến thức thông qua các bài quiz. Mục tiêu chính của dự án là tạo ra một môi trường học tập trực tuyến tích hợp, nơi mà người dùng có thể tự kiểm tra kiến thức, tham gia vào các bài quiz được tạo bởi cộng đồng, và tạo ra các bài quiz cá nhân hoặc nhóm để chia sẻ với người khác nhằm nâng cao khả năng học tập và chia sẻ kiến thức.

NHIỆM VỤ ĐỒ ÁN TỐT NGHIỆP

Họ tên sinh viên: Phan Tiến Đạt Số thẻ sinh viên: 102200014
Lớp: 20T1 Khoa: Công nghệ thông tin Ngành: Công nghệ phần mềm

1. Tên đề tài đồ án:

Xây dựng website quiz học tập - Piki

2. Đề tài thuộc diện: ☐ Có ký kết thỏa thuận sở hữu trí tuệ đối với kết quả thực hiện

3. Các số liệu và dữ liệu ban đầu:

Không có.....
.....

4. Nội dung các phần thuyết minh và tính toán:

Mở đầu: Giới thiệu tổng quan đề tài, mục đích, ý nghĩa và yêu cầu đặt ra của đề tài, phương pháp thực hiện và bố cục của đồ án.

Cơ sở lý thuyết: Trình bày những cơ sở lý thuyết được áp dụng trong đề tài.

Phân tích và thiết kế hệ thống: Trình bày bản phân tích và thiết kế trong xây dựng hệ thống.

Triển khai và đánh giá kết quả: Trình bày cách triển khai, cài đặt vận hành hệ thống và đánh giá kết quả đạt được.

Kết luận: Rút ra kết quả, những bài học đạt được qua quá trình xây dựng hệ thống, những điểm hạn chế và hướng phát triển mở rộng.

5. Các bản vẽ, đồ thị (Ghi rõ các loại và kích thước bản vẽ):

Không có.....
.....
.....
.....

6. Họ tên người hướng dẫn: TS. Bùi Thị Thanh Thanh

7. Ngày giao nhiệm vụ đồ án:/...../2024

8. Ngày hoàn thành đồ án:/...../2024.

Đà Nẵng, ngày tháng năm 2024

Trưởng Bộ môn Công nghệ phần mềm

Người hướng dẫn

LỜI NÓI ĐẦU

Trong suốt thời gian học tập tại Trường Đại học Bách Khoa - Đại học Đà Nẵng và trong quá trình thực hiện đồ án tốt nghiệp, em đã nhận được rất nhiều sự giúp đỡ và hướng dẫn nhiệt tình từ quý Thầy, Cô khoa Công nghệ Thông tin. Em xin bày tỏ lòng biết ơn sâu sắc đến các Thầy, Cô đã tận tâm dạy dỗ, truyền đạt kiến thức chuyên môn và khơi gợi tinh thần học tập cho em. Nhờ có sự chỉ bảo tận tình đó, em mới có thể hoàn thành tốt đồ án tốt nghiệp của mình.

Em xin bày tỏ lòng biết ơn sâu sắc đến cô TS. Bùi Thị Thanh Thanh, người đã tận tình hướng dẫn và hỗ trợ em trong suốt quá trình thực hiện đồ án tốt nghiệp. Cô đã giúp em hiểu rõ mục tiêu của đồ án, đưa ra những lời khuyên hữu ích trong việc lựa chọn đề tài, và luôn sẵn sàng dành thời gian gặp gỡ để thảo luận và giải đáp thắc mắc. Sự hướng dẫn và giúp đỡ của cô đã đóng vai trò quan trọng trong việc em hoàn thành tốt đồ án này.

Trong suốt quá trình học tập và làm đồ án tốt nghiệp, Em không tránh khỏi những thiếu sót. Em mong nhận được những ý kiến đóng góp quý báu từ thầy cô và các bạn để em có thể cải thiện, hoàn thiện kết quả và tích lũy thêm nhiều kinh nghiệm hữu ích cho công việc tương lai.

Em xin chân thành cảm ơn!

CAM ĐOAN

Tôi xin cam đoan:

1. Báo cáo đồ án tốt nghiệp Tên đề tài: Xây dựng website quiz học tập - Piki là công trình nghiên cứu của chính cá nhân tôi dưới sự hướng dẫn trực tiếp của giảng viên TS. Bùi Thị Thanh Thanh.
2. Tôi đã tự đọc và nghiên cứu để làm nên báo cáo này và đảm bảo không sao chép ở bất cứ đâu.
3. Những lý thuyết trong luận văn đều được sử dụng tài liệu như tôi đã tham khảo ở phần tài liệu tham khảo đã có trong báo cáo.

Nếu có vi phạm, tôi xin chịu hoàn toàn trách nhiệm.

Sinh viên thực hiện

Phan Tiến Đạt

MỤC LỤC

TÓM TẮT

NHIỆM VỤ ĐỒ ÁN TỐT NGHIỆP

LỜI NÓI ĐẦU	i
CAM ĐOAN	ii
MỤC LỤC	iii
DANH MỤC HÌNH ẢNH	v
DANH MỤC BẢNG	vii
MỞ ĐẦU	1
1. Phân tích hiện trạng	1
2. Mục đích	1
3. Đối tượng khách hàng	2
4. Công nghệ phát triển	2
5. Bộ cục báo cáo và vai trò, nhiệm vụ	2
CHƯƠNG 1: CƠ SỞ LÝ THUYẾT	3
1.1 Công nghệ sử dụng	3
1.1.1 Typescript	3
1.1.2 NestJS Framework	3
1.1.3 MongoDB Atlas	4
1.1.4 AWS	7
1.1.5 Socket IO	7
1.2 RESTful API	8
1.2.1 Giới thiệu	8
1.2.2 Diễn giải các thành phần	8
1.2.3 RESTful	9
1.2.4 Authentication và dữ liệu trả về	9
1.2.5 Status code	10
1.2.6 HTTP Request	10
1.3 Kết chương	10

CHƯƠNG 2. PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG	11
2.1 Phân tích yêu cầu đề tài	11
2.1.1 Tóm tắt hoạt động của hệ thống mà dự án sẽ được ứng dụng	11
2.1.2 Đối tượng sử dụng	11
2.1.3 Mục đích của dự án	11
2.2 Xác định yêu cầu đề tài	12
2.3 Sơ đồ phân tích thiết kế hệ thống	13
2.3.1 Sơ đồ Use-case	13
2.3.2 Sơ đồ tuần tự	17
2.3.2 Sơ đồ lớp	19
2.4 Thiết kế cơ sở dữ liệu	20
2.5 Kết chương	26
CHƯƠNG 3. TRIỂN KHAI VÀ ĐÁNH GIÁ KẾT QUẢ	27
3.1 Xây dựng API	27
3.1.1 Cấu trúc thư mục	27
3.1.2 Cấu trúc module	28
3.1.3 Repository pattern	28
3.1.4 Socket IO realtime	29
3.1.5 Tài liệu Swagger	30
3.2 Giao diện chức năng người dùng	31
3.3 Giao diện chức năng cho quản trị viên	39
3.4 Triển khai	42
3.5 Kết chương	42
KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN	
TÀI LIỆU THAM KHẢO	

DANH MỤC HÌNH ẢNH

Hình 1. Cấu trúc module của NestJS -----	4
Hình 2. Ví dụ về cách hoạt động Socket IO -----	7
Hình 3. Mô hình RESTful API -----	8
Hình 4. Sơ đồ hoạt động RESTful API -----	9
Hình 5. Dữ liệu trả về của RESTful API -----	9
Hình 6. Usecase Người dùng -----	13
Hình 7. Usecase Admin -----	14
Hình 8. Usecase quản lý quiz -----	15
Hình 9. Usecase quản lý lớp học -----	16
Hình 10. Sơ đồ tuần tự chức năng đăng ký -----	17
Hình 11. Sơ đồ tuần tự chức năng đăng nhập -----	17
Hình 12. Sơ đồ tuần tự chức năng cập nhật thông tin cá nhân -----	18
Hình 13. Sơ đồ tuần tự chức năng tạo quiz -----	18
Hình 14. Sơ đồ tuần tự chức năng tạo quiz -----	19
Hình 15. Sơ đồ lớp -----	19
Hình 16. Thiết kế cơ sở dữ liệu -----	20
Hình 17. Cấu trúc thư mục -----	27
Hình 18. Các thuộc tính của module -----	28
Hình 19. Repository pattern -----	29
Hình 20. Socket room -----	29
Hình 21. Tài liệu Swagger -----	30
Hình 22. Giao diện trang đăng nhập -----	31
Hình 23. Giao diện trang đăng ký -----	31
Hình 24. Giao diện trang chủ -----	32
Hình 25. Giao diện tạo quiz -----	33
Hình 26. Giao diện kết quả tạo quiz -----	33
Hình 27. Chức năng kiểm tra quiz -----	34

Hình 28. Chức năng gán dấu sao -----	34
Hình 29. Chức năng xem lịch sử làm bài -----	34
Hình 30. Chức năng đánh giá quiz -----	35
Hình 31. Chức năng tạo thư mục -----	35
Hình 32. Chức năng thêm quiz vào thư mục -----	36
Hình 33. Chức năng tạo lớp học -----	36
Hình 34. Chức năng thêm quiz vào lớp học -----	37
Hình 35. Chức năng tìm kiếm -----	37
Hình 36. Giao diện cửa sổ truy cập nhanh -----	38
Hình 37. Chức năng thêm chức chỉ -----	38
Hình 38. Chức năng nhấn tin thời gian thực -----	39
Hình 39. Dashboard tài khoản -----	39
Hình 40. Chức năng quản lý học phần -----	40
Hình 41. Chức năng quản lý chủ đề -----	40
Hình 42. Chức năng vô hiệu hoá tài khoản -----	41
Hình 43. Chức năng quản lý chứng chỉ -----	41
Hình 44. Chức năng quản lý đánh giá -----	41
Hình 45. Deployment trên Railway -----	42

DANH MỤC BẢNG

Bảng 1. Đặc tả usecase quản lý quiz -----	16
Bảng 2. Đặc tả usecase quản lý lớp học-----	17
Bảng 3. Bảng Tài khoản -----	21
Bảng 4. Bảng Chứng chỉ -----	21
Bảng 5. Bảng Đáp án -----	22
Bảng 6. Bảng Lần làm-----	22
Bảng 7. Bảng Lớp học-----	22
Bảng 8. Bảng Danh sách lớp học -----	23
Bảng 9. Bảng Thư mục-----	23
Bảng 10. Bảng Câu hỏi-----	23
Bảng 11. Bảng Quiz -----	24
Bảng 12. Bảng Record Quiz-----	24
Bảng 13. Bảng Yêu cầu -----	25
Bảng 14. Bảng Chủ đề -----	25
Bảng 15. Bảng Đánh dấu -----	25
Bảng 16. Bảng liên hệ -----	25
Bảng 17. Bảng tin nhắn -----	26

MỞ ĐẦU

1. Phân tích hiện trạng

Ngày nay, công nghệ thông tin (CNTT) phát triển với tốc độ vượt bậc, ảnh hưởng sâu rộng đến nhiều lĩnh vực trong đời sống, bao gồm cả giáo dục. Hầu hết mọi người hiện nay đều sở hữu điện thoại di động hoặc máy tính, phục vụ cho việc học tập, công việc và giải trí. Việc tạo ra các trang web và ứng dụng học tập trực tuyến trở thành nhu cầu thiết yếu.

Internet không ngừng phát triển đã mang lại vô số cơ hội cho việc áp dụng công nghệ vào cuộc sống hàng ngày. Các ứng dụng học tập trực tuyến giúp người dùng tìm kiếm thông tin nhanh chóng, tiết kiệm thời gian và có thể sử dụng mọi lúc, mọi nơi. Điều này đã mở ra một phương thức học tập mới, linh hoạt và hiệu quả hơn. Trong lĩnh vực giáo dục, quảng bá khóa học và tài liệu học tập qua mạng Internet không chỉ giúp tiếp cận đông đảo người học mà còn tạo ra một cộng đồng học tập năng động và sáng tạo.

Chính vì những lợi ích thiết thực này, việc xây dựng các nền tảng học tập trực tuyến đã và đang phát triển mạnh mẽ trên toàn thế giới, góp phần thay đổi cách thức học tập truyền thống và nâng cao chất lượng giáo dục. Trong hoạt động giáo dục, các nền tảng học tập trực tuyến đóng vai trò quan trọng trong việc cung cấp kiến thức và kỹ năng cho người học, giúp họ tự chủ và linh hoạt hơn trong quá trình học tập.

Xuất phát từ nhu cầu thực tế của người học và những mong muốn của các cơ sở giáo dục, cùng với sự đồng ý của Trường Đại học Bách Khoa – Đại học Đà Nẵng và sự hướng dẫn tận tình của cô TS. Bùi Thị Thanh Thanh, em đã tiến hành thực hiện đề tài "Xây dựng website quiz học tập - Piki". Đề tài này không chỉ nhằm đáp ứng nhu cầu học tập của người dùng mà còn mang lại giải pháp tối ưu cho các tổ chức giáo dục trong việc quản lý và phát triển nội dung học tập.

Website là một nền tảng cho phép người dùng tạo, đăng tải và làm các bài kiểm tra trắc nghiệm (quiz) của nhau. Điều này không chỉ giúp người học tự đánh giá được mức độ hiểu biết của mình mà còn khuyến khích họ chia sẻ kiến thức, tạo ra một môi trường học tập tương tác. Với sự hỗ trợ nhiệt tình từ gia đình, bạn bè và thầy cô, em hy vọng rằng dự án này sẽ mang lại nhiều lợi ích cho cộng đồng học tập trực tuyến.

2. Mục đích

Hệ thống là một web app cho phép các người dùng có nhu cầu chia sẻ kiến thức, học tập, làm bài qua mạng miễn phí. Thông qua nền tảng này người dùng có thể thao tác đơn giản trên trình duyệt web để điều hướng đến những bộ quiz phù hợp yêu cầu.

3. Đối tượng khách hàng

Mọi đối tượng có nhu cập học tập qua mạng.

4. Công nghệ phát triển

- Quản lý và lưu trữ dự án: Github
- Ngôn ngữ: Typescript, Javascript
- Backend: NestJS
- Frontend: ReactJs
- Database: MongoDB Atlas
- Mã hoá & bảo mật: HTTPS Token

5. Bộ cục báo cáo và vai trò, nhiệm vụ

❖ Đề tài được thực hiện bởi hai sinh viên:

- Phan Tiến Đạt Mã số sinh viên: 102200014 Lớp: 20T1
- Trần Thị Ngà Mã số sinh viên: 102200025 Lớp: 20T1

❖ Vai trò, nhiệm vụ trong đề tài:

- Chọn và xác định đề tài
- Thiết kế tài liệu requirement
- Xây dựng nghiệp vụ đề tài
- Xây dựng cơ sở dữ liệu
- Lập trình API
- Deploy lên server

❖ Báo cáo trình bày chia thành 3 chương chính:

- Chương 1 - Cơ sở lý thuyết: Trình bày những cơ sở lý thuyết được áp dụng trong đề tài.
- Chương 2 - Phân tích và thiết kế hệ thống: Trình bày bản phân tích và thiết kế trong xây dựng hệ thống.
- Chương 3 - Triển khai và đánh giá kết quả: Trình bày cách triển khai, cài đặt vận hành hệ thống và đánh giá kết quả đạt được.

CHƯƠNG 1: CƠ SỞ LÝ THUYẾT

1.1 Công nghệ sử dụng

1.1.1 Typescript

TypeScript là một dự án mã nguồn mở được phát triển bởi Microsoft, nó có thể được coi là một phiên bản nâng cao của Javascript bởi việc bổ sung tùy chọn kiểu tĩnh và lớp hướng đối tượng mà điều này không có ở Javascript. TypeScript có thể sử dụng để phát triển các ứng dụng chạy ở client-side (Angular2) và server-side (NodeJS) [1].

TypeScript sử dụng tất cả các tính năng của của ECMAScript 2015 (ES6) như classes, modules. Không dừng lại ở đó nếu như ECMAScript 2017 ra đời thì mình tin chắc rằng TypeScript cũng sẽ nâng cấp phiên bản của mình lên để sử dụng mọi kỹ thuật mới nhất từ ECMAScript.

❖ Ưu điểm của Typescript:

- Dễ phát triển dự án lớn: Với việc sử dụng các kỹ thuật mới nhất và lập trình hướng đối tượng nên TypeScript giúp chúng ta phát triển các dự án lớn một cách dễ dàng.
- Nhiều Framework lựa chọn: Hiện nay các Javascript Framework đã dần khuyến khích nên sử dụng TypeScript để phát triển, ví dụ như AngularJS 2.0 và Ionic 2.0.
- Hỗ trợ các tính năng của Javascript phiên bản mới nhất: TypeScript luôn đảm bảo việc sử dụng đầy đủ các kỹ thuật mới nhất của Javascript, ví dụ như version hiện tại là ECMAScript 2015 (ES6).
- Là mã nguồn mở: TypeScript là một mã nguồn mở nên hoàn toàn có thể sử dụng mà không mất phí, bên cạnh đó còn được cộng đồng hỗ trợ.
- TypeScript là Javascript: Bản chất của TypeScript là biên dịch tạo ra các đoạn mã javascript nên bạn có thể chạy bất kì ở đâu miễn ở đó có hỗ trợ biên dịch Javascript. Ngoài ra bạn có thể sử dụng trọn lẫn cú pháp của Javascript vào bên trong TypeScript, điều này giúp các lập trình viên tiếp cận TypeScript dễ dàng hơn.

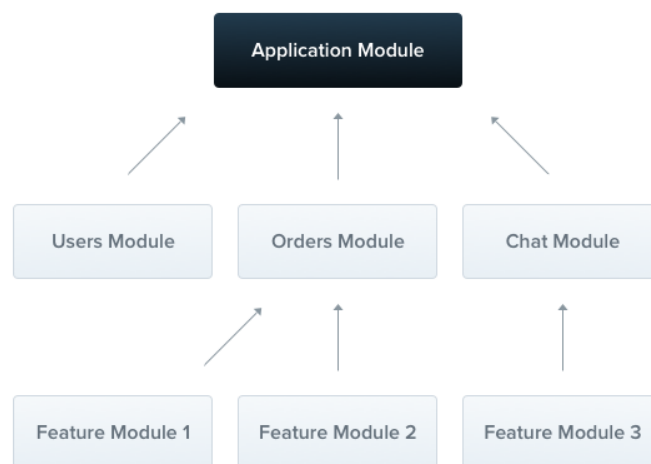
1.1.2 NestJS Framework

NestJS là một NodeJS framework dùng để phát triển server-side applications hiệu quả và có thể mở rộng. NestJS là sự kết hợp bởi OOP (Object Oriented Programming), FP(Functional Programming), FRP(Functional Reactive Programming) [2].

Nestjs hỗ trợ:

- NestJS tương thích với cả Typescript và Javascript thuần, default là Typescript.

- Design Pattern xuyên suốt NestJS là Dependency Injection, và cú pháp của NestJS rất giống Angular. Nên nếu bạn đã có kiến thức về Angular thì việc bắt đầu với NestJS sẽ khá dễ dàng.
 - NestJS có rất nhiều module hỗ trợ bạn, từ việc hot reload, logger cho đến GraphQL, Websocket rồi cQRS pattern, microservices,...
 - Module microservices của NestJS hỗ trợ đủ loại kết nối: RabbitMQ, gRPC, Kafka, Redis,... đây là một trong những điều mình rất thích, mình chỉ cần tập trung vào code cho phần business, còn infra thì NestJS đã implement sẵn chỉ việc dùng.
- Cấu trúc module của NestJs:



Hình 1. Cấu trúc module của NestJS

1.1.3 MongoDB Atlas

MongoDB là một cơ sở dữ liệu mã nguồn mở và là cơ sở dữ liệu NoSQL(*) hàng đầu, được hàng triệu người sử dụng. MongoDB được viết bằng C++ [3].

Ngoài ra, MongoDB là một cơ sở dữ liệu đa nền tảng, hoạt động trên các khái niệm Collection và Document, nó cung cấp hiệu suất cao, tính khả dụng cao và khả năng mở rộng dễ dàng.

NoSQL là 1 dạng CSDL mã nguồn mở không sử dụng Transact-SQL để truy vấn thông tin. NoSQL viết tắt bởi: None-Relational SQL, hay có nơi thường gọi là Not-Only SQL. CSDL này được phát triển trên Javascript Framework với kiểu dữ liệu JSON. (Cú pháp của JSON là “key:value”) NoSQL ra đời như là 1 mảnh vá cho những khuyết điểm và thiếu sót cũng như hạn chế của mô hình dữ liệu quan hệ RDBMS về tốc độ, tính năng, khả năng mở rộng, memory cache,...

❖ Các thuật ngữ được sử dụng trong MongoDB:

- `_id` – Là trường bắt buộc có trong mỗi document. Trường `_id` đại diện cho một giá trị duy nhất trong document MongoDB. Trường `_id` cũng có thể được hiểu là khóa chính trong document. Nếu bạn thêm mới một document thì MongoDB sẽ tự động sinh ra một `_id` đại diện cho document đó và là duy nhất trong cơ sở dữ liệu MongoDB.
- Collection – Là nhóm của nhiều document trong MongoDB. Collection có thể được hiểu là một bảng tương ứng trong cơ sở dữ liệu RDBMS (Relational Database Management System). Collection nằm trong một cơ sở dữ liệu duy nhất. Các collection không phải định nghĩa các cột, các hàng hay kiểu dữ liệu trước.
- Cursor – Đây là một con trỏ đến tập kết quả của một truy vấn. Máy khách có thể lặp qua một con trỏ để lấy kết quả.
- Database – Nơi chứa các Collection, giống với cơ sở dữ liệu RDMS chúng chứa các bảng. Mỗi Database có một tập tin riêng lưu trữ trên bộ nhớ vật lý. Một máy chủ MongoDB có thể chứa nhiều Database.
- Document – Một bản ghi thuộc một Collection thì được gọi là một Document. Các Document lần lượt bao gồm các trường tên và giá trị.
- Field – Là một cặp name – value trong một document. Một document có thể có không hoặc nhiều trường. Các trường giống các cột ở cơ sở dữ liệu quan hệ.
- JSON – Viết tắt của JavaScript Object Notation. Con người có thể đọc được ở định dạng văn bản đơn giản thể hiện cho các dữ liệu có cấu trúc. Hiện tại JSON đang hỗ trợ rất nhiều ngôn ngữ lập trình.
- Index – Là những cấu trúc dữ liệu đặc biệt, dùng để chứa một phần nhỏ của các tập dữ liệu một cách dễ dàng để quét. Chỉ số lưu trữ giá trị của một fields cụ thể hoặc thiết lập các fields, sắp xếp theo giá trị của các fields này. Index hỗ trợ độ phân tích một cách hiệu quả các truy vấn. Nếu không có chỉ mục, MongoDB sẽ phải quét tất cả các documents của collection để chọn ra những document phù hợp với câu truy vấn. Quá trình quét này là không hiệu quả và yêu cầu MongoDB để xử lý một khối lượng lớn dữ liệu.

❖ Các kiểu dữ liệu trong MongoDB:

- Chuỗi: Đây là kiểu dữ liệu được sử dụng phổ biến nhất để lưu giữ dữ liệu. Chuỗi trong MongoDB phải là UTF-8 hợp lệ.
- Số nguyên: Kiểu dữ liệu này được sử dụng để lưu một giá trị số. Số nguyên có thể là 32 bit hoặc 64 bit phụ thuộc vào Server của bạn.
- Boolean: Kiểu dữ liệu này được sử dụng để lưu giữ một giá trị Boolean (true/false).

- Double: Kiểu dữ liệu này được sử dụng để lưu các giá trị số thực dấu chấm động.
- Min/ Max keys: Kiểu dữ liệu này được sử dụng để so sánh một giá trị với các phần tử BSON thấp nhất và cao nhất.
- Mảng: Kiểu dữ liệu này được sử dụng để lưu giữ các mảng hoặc danh sách hoặc nhiều giá trị vào trong một key.
- Timestamp: Đánh dấu thời điểm một Document được sửa đổi hoặc được thêm vào.
- Object: Kiểu dữ liệu này được sử dụng cho các Document được nhúng vào.
- Null: Kiểu dữ liệu này được sử dụng để lưu một giá trị Null.
- Symbol: Kiểu dữ liệu này được sử dụng giống như một chuỗi
- Date : Kiểu dữ liệu này được sử dụng để lưu giữ date và time hiện tại trong định dạng UNIX time.
- Object ID: Kiểu dữ liệu này được sử dụng để lưu giữ ID của Document.
- Binary data: Kiểu dữ liệu này được sử dụng để lưu giữ dữ liệu nhị phân.
- Code: Kiểu dữ liệu này được sử dụng để lưu giữ JavaScript code vào trong Document.
- Regular expression: Kiểu dữ liệu này được sử dụng để lưu giữ Regular Expression.

❖ Lợi thế của MongoDB:

- Ít schema hơn: Vì schema được sinh ra là để nhóm các đối tượng vào 1 cụm, dễ quản lý. Ví dụ như tạo 1 schema tên là Students chẳng hạn thì chỉ có những gì liên quan đến student thì mới được cho vào schema này. Trong khi đó trong mongodb thì chỉ 1 collection ta có thể chứa nhiều document khác nhau. Với mỗi document thì số trường, nội dung, kích thước lại có thể khác nhau.
- Cấu trúc của một đối tượng rõ ràng.
- Không có các Join phức tạp.
- Khả năng mở rộng cực lớn: việc mở rộng dữ liệu mà không phải lo đến các vấn đề như khóa ngoại, khóa chính, kiểm tra ràng buộc, ... MongoDB cho phép thực hiện replication và sharding nên việc mở rộng cũng thuận lợi hơn.

MongoDB Atlas là cloud database của MongoDB được ra mắt vào năm 2016 chạy trên AWS, Microsoft Azure và Google Cloud Platform.

Dữ liệu trong mỗi Cluster ở Atlas được lưu trữ theo cơ chế Replication, với 3 nodes: 1 master (primary) và 2 slaves (secondary).

MongDb Atlas cho phép lưu và kết nối đến dữ liệu trên cloud đã được deploy, giúp đồng bộ dữ liệu đảm bảo bảo mật.

1.1.4 AWS

Amazon Simple Storage Service (Amazon S3) là một dịch vụ lưu trữ đối tượng với khả năng thay đổi theo quy mô, tính khả dụng của dữ liệu, độ bảo mật và hiệu năng hàng đầu trong ngành. Bạn có thể sử dụng Amazon S3 để lưu trữ và truy xuất dữ liệu với khối lượng bất kỳ, vào bất cứ thời điểm, nơi nào.

❖ S3 an toàn vì AWS cung cấp:

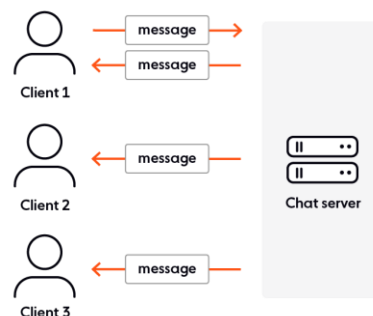
- Mã hóa dữ liệu mà bạn lưu trữ. Nó có thể xảy ra theo 2 cách là mã hoá ở client side và mã hoá ở server side
- Nhiều bản copy được duy trì để cho phespo phục hồi dữ liệu trong trường hợp dữ liệu bị hỏng.
- Versioning, trong đó mỗi bản chỉnh sửa đều được lưu trữ để sử dụng khi cần thiết.
- Có thể lưu trữ bất kì loại data với bất kì loại format nào. Trên S3 khi chúng ta nói về dung lượng, số lượng của đối tượng mà chúng ta có thể lưu trữ trên S3 là không giới hạn. Một đối tượng là thực thể cơ bản trong S3. Nó bao gồm dữ liệu, khóa và siêu dữ liệu (metadata).

1.1.5 Socket IO

Socket.IO là một thư viện mã nguồn mở trong Node.js được phát triển từ năm 2010. Mục đích nhằm tạo ra một môi trường giao tiếp thời gian thực và hai chiều giữa máy chủ và máy khách trên Internet. Nền tảng cho phép gửi và nhận dữ liệu một cách tức thì và tối ưu hóa việc sử dụng tài nguyên [4].

Công nghệ được tích hợp với nhiều ngôn ngữ khác nhau như PHP, ASP.NET, Node.js để tạo ra một môi trường giao tiếp hai chiều giữa máy chủ và máy khách.

Một cách phổ biến để chứng minh khả năng giao tiếp hai chiều mà Socket.IO cung cấp là một ứng dụng trò chuyện cơ bản. Với socket, khi máy chủ nhận được tin nhắn mới, nó sẽ gửi tin nhắn đó đến máy khách và thông báo cho họ, bỏ qua nhu cầu gửi yêu cầu giữa máy khách và máy chủ. Một ứng dụng trò chuyện đơn giản cho thấy cách thức hoạt động của tính năng này.

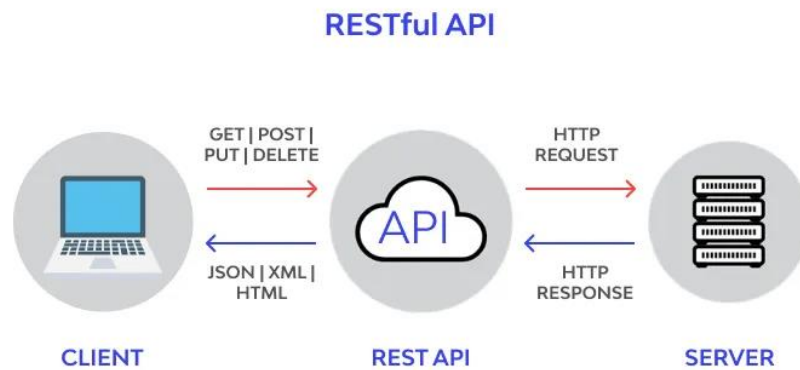


Hình 2. Ví dụ về cách hoạt động Socket IO

1.2 RESTful API

1.2.1 Giới thiệu

RESTful API là một tiêu chuẩn dùng trong việc thiết kế API cho các ứng dụng web (thiết kế Web services) để tiện cho việc quản lý các resource. Nó chú trọng vào tài nguyên hệ thống (tệp văn bản, ảnh, âm thanh, video, hoặc dữ liệu động...), bao gồm các trạng thái tài nguyên được định dạng và được truyền tải qua HTTP [5].



Hình 3. Mô hình RESTful API

1.2.2 Diễn giải các thành phần

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.

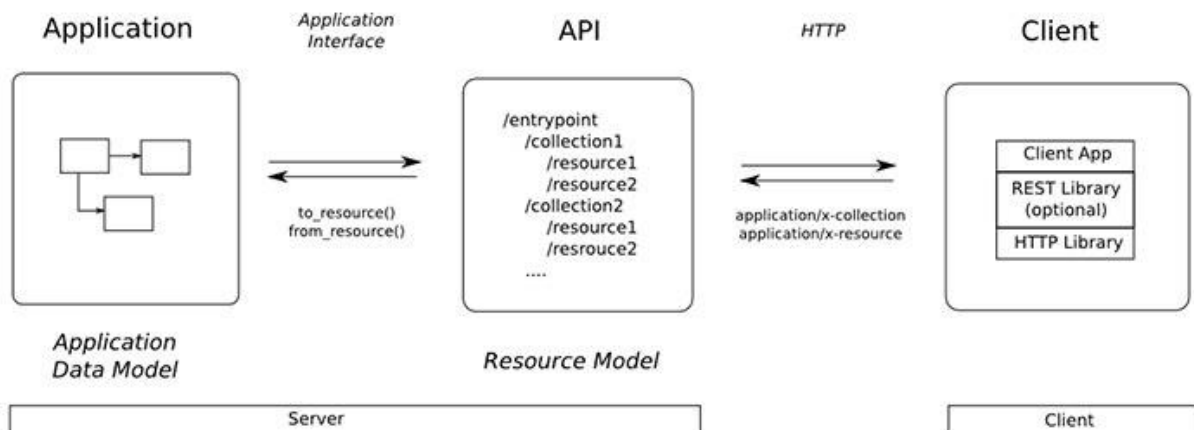
REST (REpresentational State Transfer) là một dạng chuyển đổi cấu trúc dữ liệu, một kiểu kiến trúc để viết API. Nó sử dụng phương thức HTTP đơn giản để tạo cho giao tiếp giữa các máy. Vì vậy, thay vì sử dụng một URL cho việc xử lý một số thông tin người dùng, REST gửi một yêu cầu HTTP như GET, POST,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

1.2.3 RESTful



Hình 4. Sơ đồ hoạt động RESTful API

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, Read, Update, Delete (Tạo, Đọc, Sửa, Xóa).

1.2.4 Authentication và dữ liệu trả về

RESTful API không sử dụng session và cookie, nó sử dụng một access_token với mỗi request. Dữ liệu trả về thường có cấu trúc như sau:

```

Pretty Raw Preview Visualize JSON
7  phone: null,
8  "is_active": true,
9  "role": 3,
10 "v": 0,
11 "refreshToken": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.
12   ADU",
13 "fcmToken": "",
14 "id": "6621259f7247e26733881534",
15 "expiresIn": "259200000",
16 "accessToken": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.
   E3NTczNzY1LCJ
  
```

Hình 5. Dữ liệu trả về của RESTful API

1.2.5 Status code

Khi chúng ta request một API nào đó thường thì sẽ có vài status code

- 200 OK – Trả về thành công cho những phương thức GET, PUT, PATCH hoặc DELETE.
- 201 Created – Trả về khi một Resource vừa được tạo thành công.
- 204 No Content – Trả về khi Resource xóa thành công.
- 304 Not Modified – Client có thể sử dụng dữ liệu cache.
- 400 Bad Request – Request không hợp lệ
- 401 Unauthorized – Request cần có auth.
- 403 Forbidden – Bị từ chối không cho phép.
- 404 Not Found – Không tìm thấy resource từ URI
- 405 Method Not Allowed – Phương thức không cho phép với user hiện tại.
- 410 Gone – Resource không còn tồn tại, Version cũ đã không còn hỗ trợ.
- 415 Unsupported Media Type – Không hỗ trợ kiểu Resource này.
- 422 Unprocessable Entity – Dữ liệu không được xác thực.
- 429 Too Many Requests – Request bị từ chối do bị giới hạn.

1.2.6 HTTP Request

HTTP request có tất cả 9 loại method (2 loại được sử dụng phổ biến nhất là GET và POST):

- GET: Được sử dụng để lấy thông tin từ server theo URI đã cung cấp.
- HEAD: Giống với GET nhưng response trả về không có body, chỉ có header.
- POST: Gửi thông tin tới sever thông qua các biểu mẫu http.
- PUT: Ghi đè tất cả thông tin của đối tượng với những gì được gửi lên.
- PATCH: Ghi đè các thông tin được thay đổi của đối tượng.
- DELETE: Xóa tài nguyên trên server.
- CONNECT: Thiết lập một kết nối tới server theo URI.
- OPTIONS: Mô tả các tùy chọn giao tiếp cho resource.
- TRACE: Thực hiện một bài test loop – back theo đường dẫn đến resource.

1.3 Kết chương

Trong chương này và phần mở đầu cung cấp cho người đọc về cơ sở lý thuyết đã được sử dụng để xây dựng hệ thống.

CHƯƠNG 2. PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG

2.1 Phân tích yêu cầu đề tài

2.1.1 Tóm tắt hoạt động của hệ thống mà dự án sẽ được ứng dụng

- Là một website cho phép các người dùng đăng và làm quiz của bản thân hoặc người dùng khác.
- Người dùng truy cập vào website có thể xem, tìm kiếm các quiz, lớp học và thư mục.
- Các quiz có thể được tìm kiếm theo chủ đề, dễ dàng tìm ra quiz phù hợp với nhu cầu cá nhân.
- Người dùng có thể xem chi tiết từng quiz, lớp học và thư mục
- Các hình thức làm quiz đều có thể được sử dụng.
- Người dùng có thể xem lại điểm số của quiz đã làm ở dạng bài kiểm tra.
- Người dùng có thể tạo ra các lớp học lưu trữ các bộ quiz tương tự thư mục, có các thành viên lớp học sẽ có hoặc không có quyền hạn sửa đổi các bộ quiz trong lớp, hoặc có thể điều chỉnh toàn cục dựa vào một cài đặt cho phép tất cả các thành viên trong lớp học chỉnh sửa bộ quiz.
- Người dùng có thể tạo ra các thư mục cá nhân lưu trữ các bộ quiz được thêm vào để dễ quản lý và truy cập.
- Người dùng có thể nhận tin thời gian thực
- Người dùng có thể để lại đánh giá quiz từ 1 đến 5 sao kèm lời nhận xét
- Quản trị viên có quyền duyệt các chứng chỉ mà người dùng đăng tải để duyệt cho các quiz thuộc chuyên môn của chứng chỉ đó.

2.1.2 Đối tượng sử dụng

Có 2 đối tượng sử dụng cơ bản là người dùng và nhà quản trị:

- **Người dùng:** Qua website, người dùng có thể xem thông tin, lựa chọn quiz phù hợp với nhu cầu bản thân để làm quiz, học tập, rèn luyện
- **Nhà quản trị:** Nhà quản trị có toàn quyền quản lý các thư mục, khoá học và các bài quiz và cả quản lý mật khẩu của người dùng

2.1.3 Mục đích của dự án

- Đáp ứng nhu cầu học tập qua mạng, ở dạng trắc nghiệm – 1 loại bài kiểm tra rất phổ biến hiện nay.
- Thúc đẩy phát triển nhu cầu học tập online.
- Dễ dàng trao đổi các câu hỏi, câu trả lời, kiến thức qua các bài kiểm tra
- Việc quản lý các bộ đề trở nên dễ dàng.

2.2 Xác định yêu cầu đề tài

Xây dựng website quiz học tập – Piki, cho phép người dùng trao đổi kiến thức, làm các bài quiz được đăng tải.

❖ **Giao diện người dùng:**

- Giao diện bắt mắt không cầu kì và dễ sử dụng
- Giao diện đáp ứng đủ chức năng yêu cầu
- Có biểu mẫu đăng ký để trở thành người dùng
- Có danh mục quiz, lớp học và thư mục
- Landing page đầy đủ
- Có biểu mẫu liên hệ.
- Chức năng tìm kiếm.

❖ **Giao diện quản trị:**

- Đơn giản, dễ quản lý có đủ các mục:
- Quản lý tài quản.
- Quản lý quiz.
- Quản lý lớp học.
- Quản lý chứng chỉ
- Quản lý đánh giá

❖ **Về bảo mật**

- Quản lý User & Password của khách hàng an toàn, thông tin của khách hàng được bảo mật.

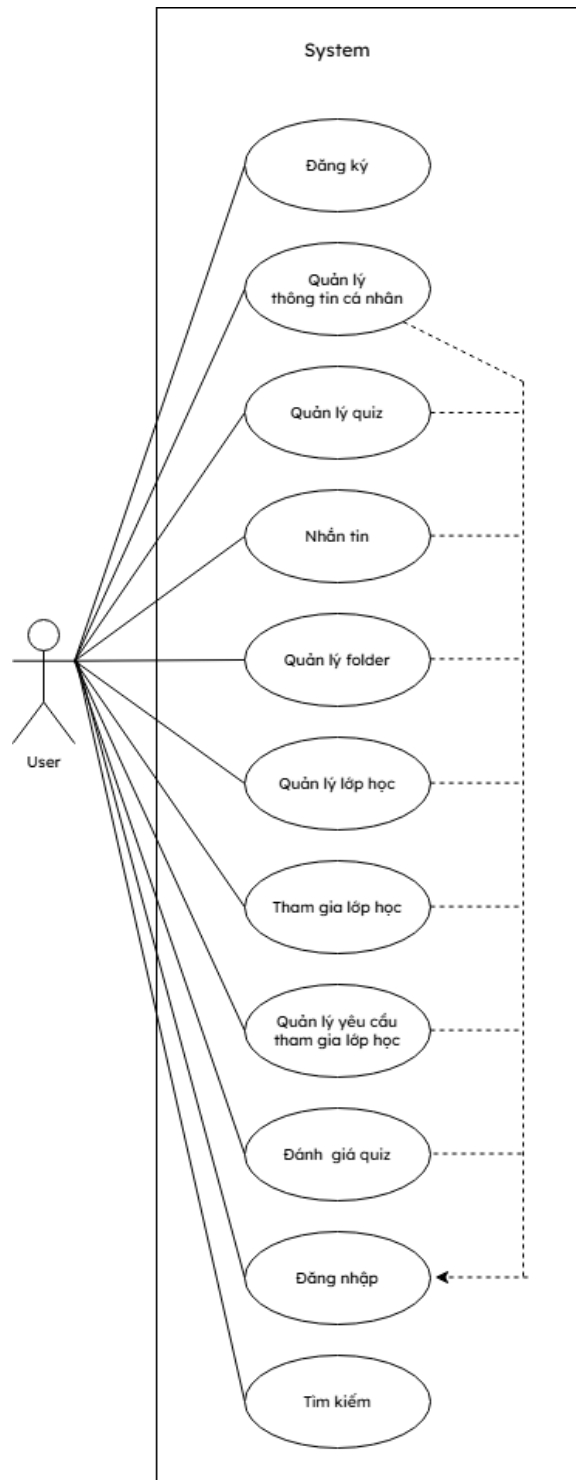
❖ **Về hệ thống:**

- Phần cứng: PC bộ vi xử lý pentium III, ram 128 trở lên, ổ cứng 10G trở lên.
- Hệ điều hành: Windown XP trở lên

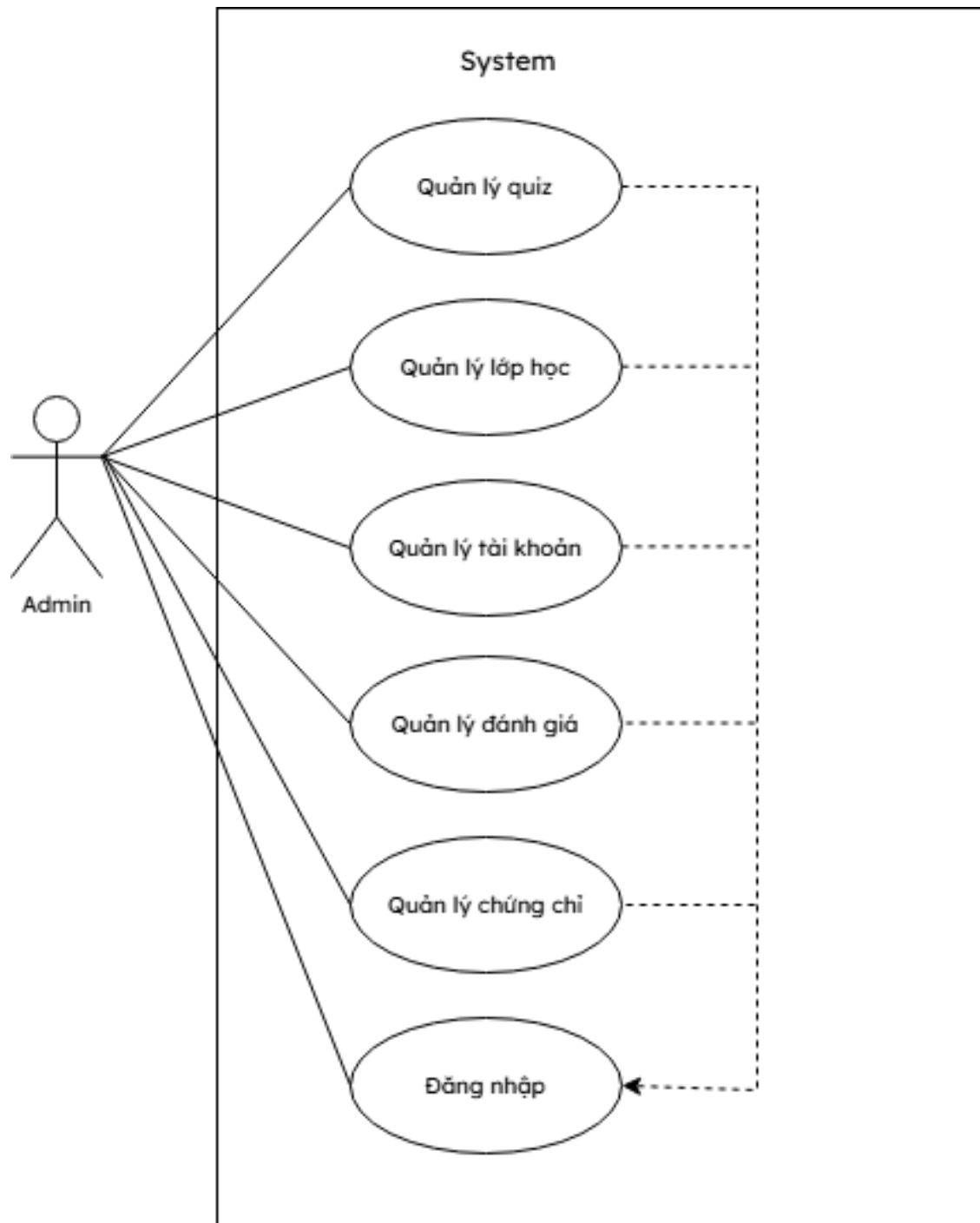
2.3 Sơ đồ phân tích thiết kế hệ thống

2.3.1 Sơ đồ Use-case

❖ Sơ đồ Use-case tổng quát

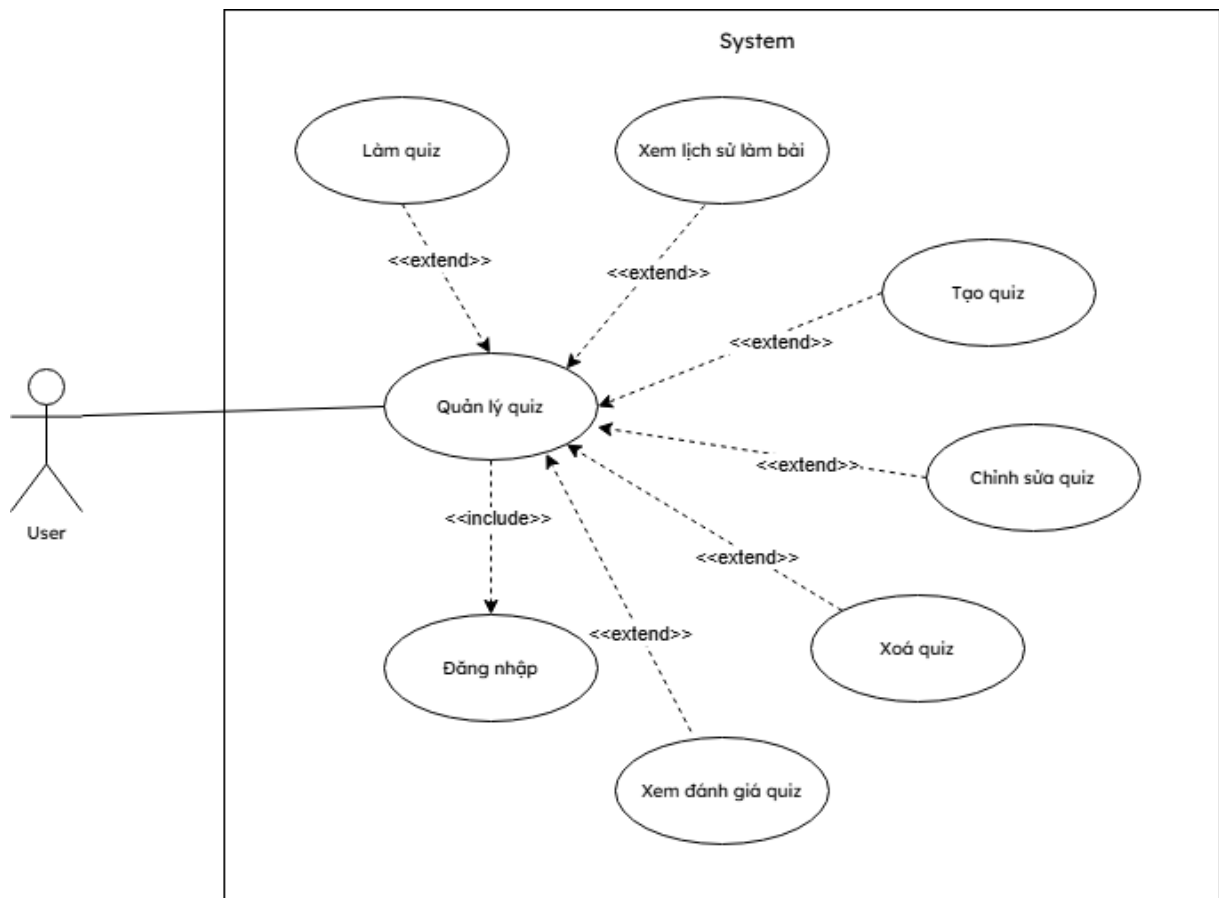


Hình 6. Usecase Người dùng



Hình 7. Usecase Admin

❖ Sơ đồ Usecase phân rã quản lý quiz



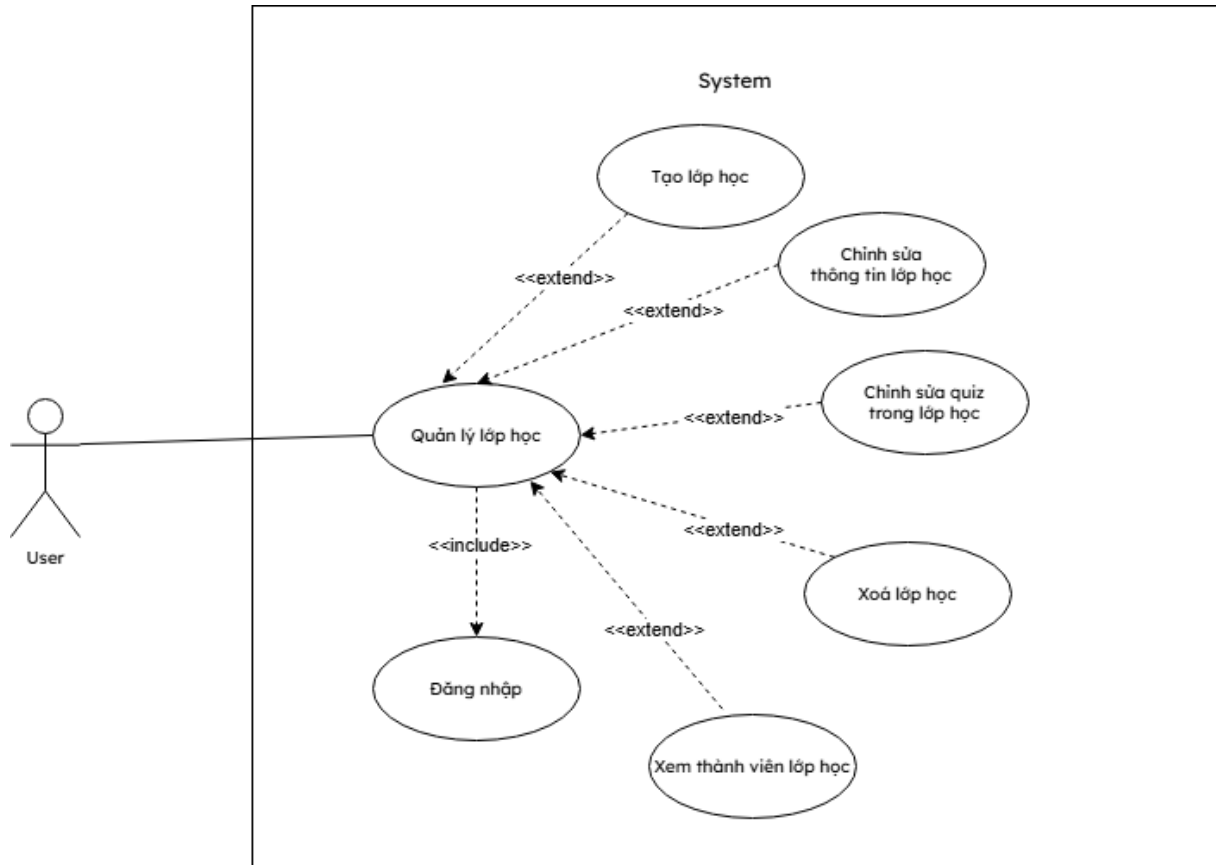
Hình 8. Usecase quản lý quiz

ID Usecase	UC01
Tên Usecase	Quản lý quiz
Tác nhân	Người dùng
Giả định	Người dùng đăng nhập thành công vào hệ thống
Điều kiện kích hoạt	Người dùng muốn quản lý quiz
Điều kiện tiên quyết	Người dùng có quyền hạn với quiz đó
Các bước thực hiện	1. Người dùng truy cập vào trang thông tin quiz. 2. Tiến hành thực hiện các chức năng xem thông tin, chỉnh sửa, hoặc xóa thông tin quiz 3. Hệ thống kiểm tra và hiển thị thông tin chi tiết sau tương tác tương ứng với chức năng được thực hiện

	4. Hệ thống cập nhật thông tin vào cơ sở dữ liệu.
Hậu điều kiện	Thao tác thành công trên quiz

Bảng 1. Đặc tả usecase quản lý quiz

❖ Sơ đồ Usecase phân rã quản lý lớp học



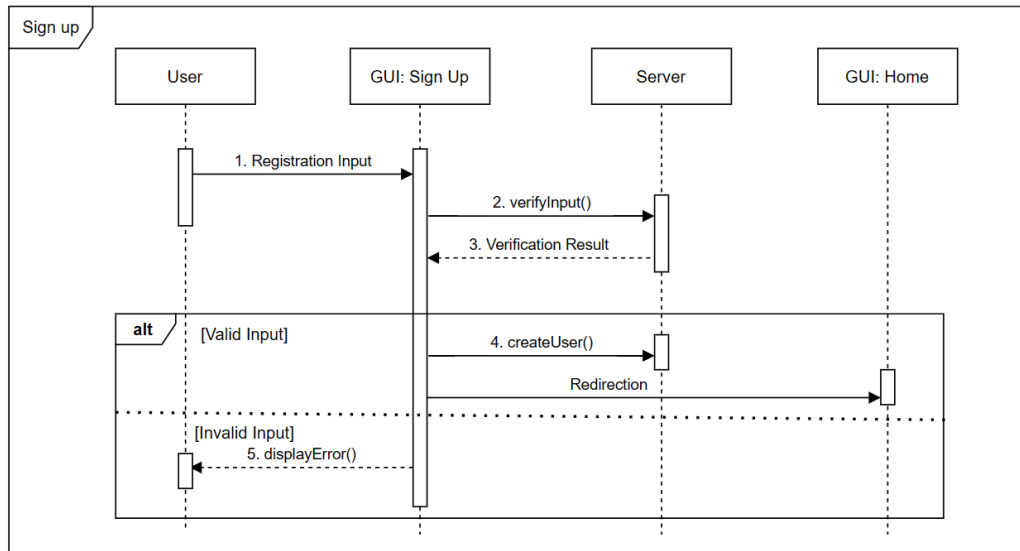
Hình 9. Usecase quản lý lớp học

ID Usecase	UC02
Tên Usecase	Quản lý lớp học
Tác nhân	Người dùng
Giả định	Người dùng đăng nhập thành công vào hệ thống
Điều kiện kích hoạt	Người dùng muốn quản lý lớp học
Điều kiện tiên quyết	Người dùng có quyền hạn với lớp học đó
Các bước thực hiện	1. Người dùng truy cập vào lớp học 2. Tiến hành thực hiện các chức năng xem thông tin, chỉnh sửa lớp học

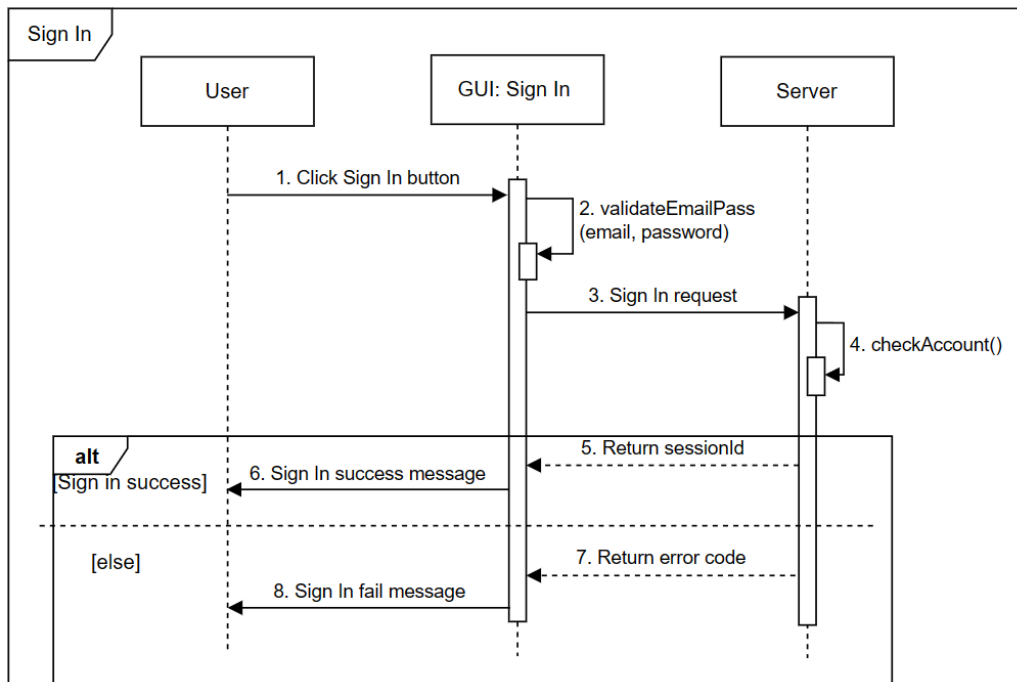
	3. Hệ thống kiểm tra và hiển thị thông tin chi tiết sau tương tác tương ứng với chức năng được thực hiện 4. Hệ thống cập nhật thông tin vào cơ sở dữ liệu.
Hậu điều kiện	Thao tác thành công trên lớp học

Bảng 2. Đặc tả usecase quản lý lớp học

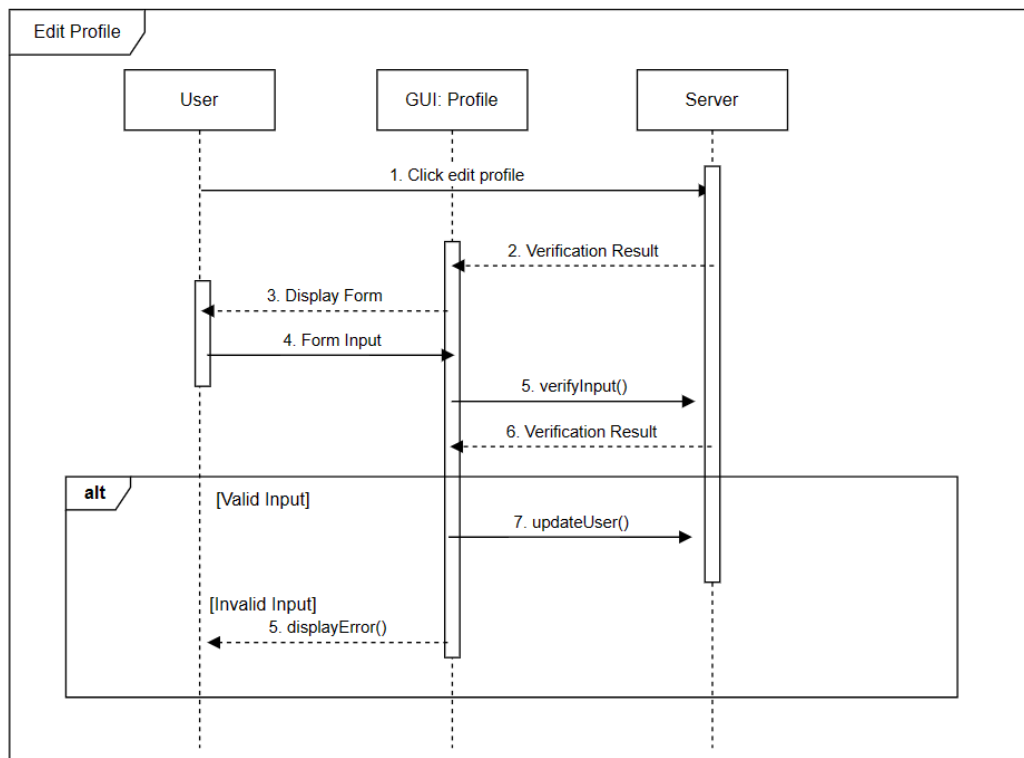
2.3.2 Sơ đồ tuần tự



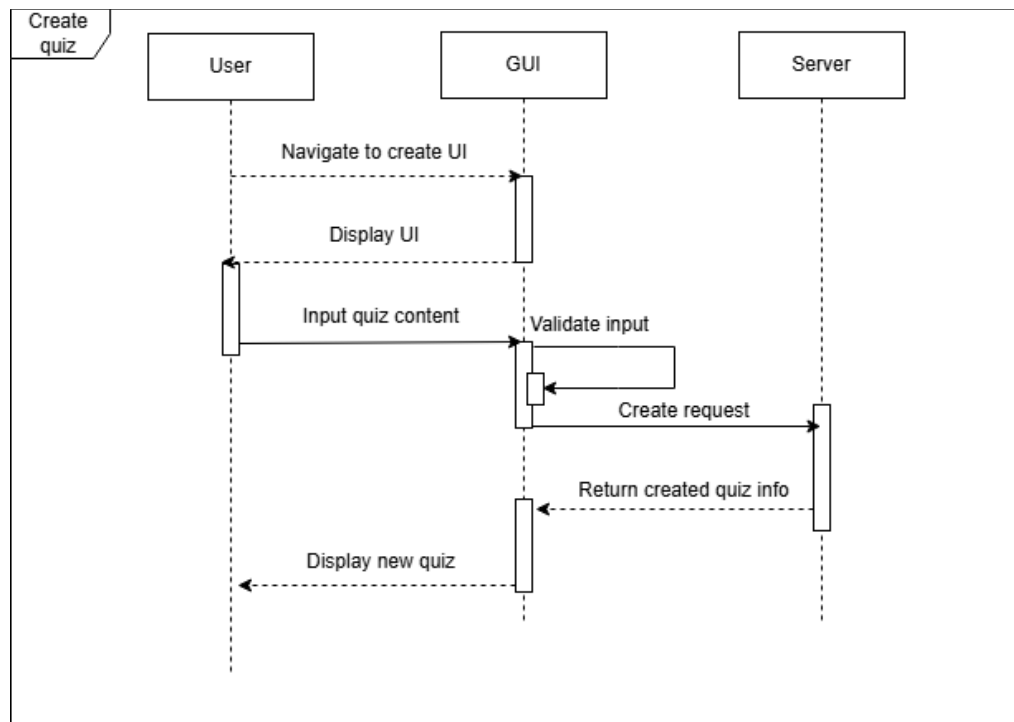
Hình 10. Sơ đồ tuần tự chức năng đăng ký



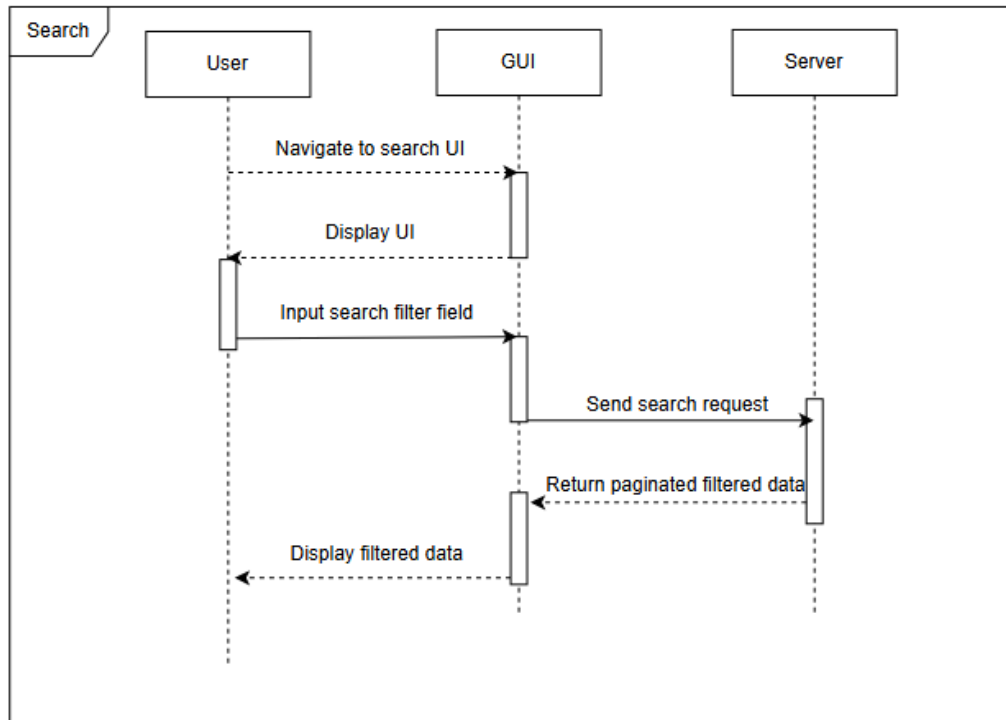
Hình 11. Sơ đồ tuần tự chức năng đăng nhập



Hình 12. Sơ đồ tuần tự chức năng cập nhật thông tin cá nhân

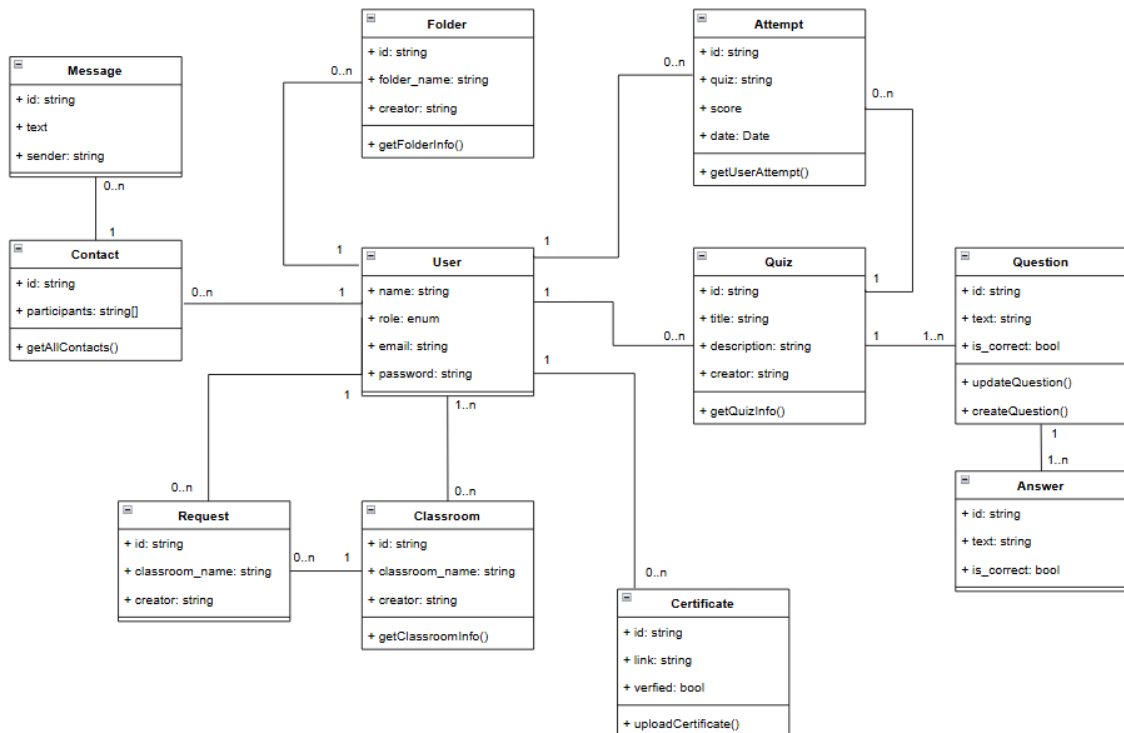


Hình 13. Sơ đồ tuần tự chức năng tạo quiz



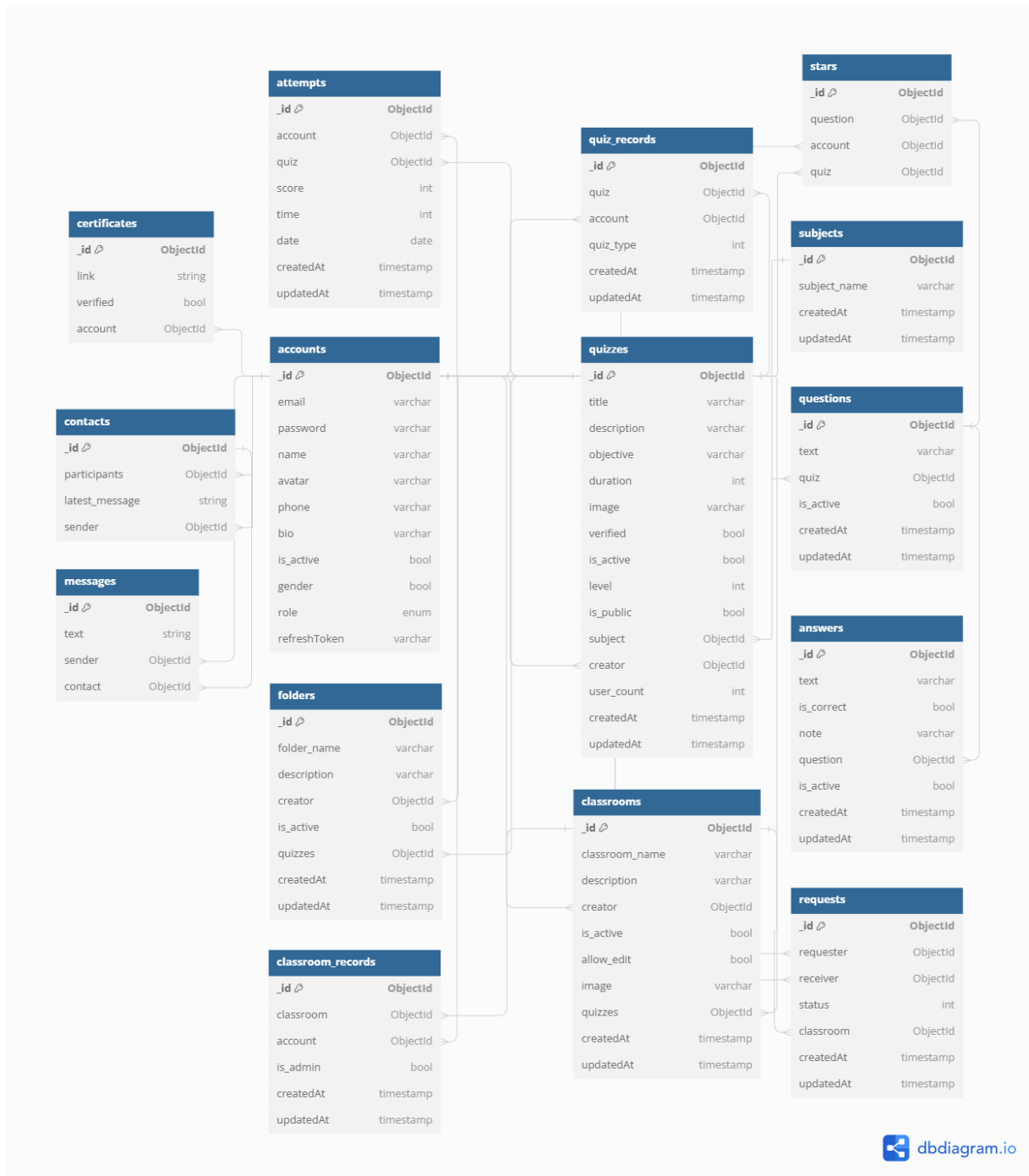
Hình 14. Sơ đồ tuần tự chức năng tạo quiz

2.3.2 Sơ đồ lớp



Hình 15. Sơ đồ lớp

2.4 Thiết kế cơ sở dữ liệu



Hình 16. Thiết kế cơ sở dữ liệu

Tên trường	Kiểu dữ liệu	Ràng buộc	Diễn giải
_id	ObjectId	Khóa chính	Mongo Id
email	String		Email đăng nhập
name	String		Tên hiển thị
password	String		Mật khẩu
phone	String		Số điện thoại
bio	String		Mô tả về bản thân
is_active	Boolean		Cờ xóa + true: Nếu đã bị xóa + false: Chưa bị xóa
gender	Boolean		Giới tính
role	Enum		Vai trò trong hệ thống (User/Admin)
refreshToken	String		RefreshToken

Bảng 3. Bảng Tài khoản

Tên trường	Kiểu dữ liệu	Ràng buộc	Diễn giải
_id	ObjectId	Khóa chính	Mongo Id
link	String		Link đến chứng chỉ
verified	Boolean		Trạng thái duyệt
account	String	Khoá ngoại	Tài khoản sở hữu

Bảng 4. Bảng Chứng chỉ

Tên trường	Kiểu dữ liệu	Ràng buộc	Diễn giải
_id	ObjectId	Khóa chính	Mongo Id
text	String		Mô tả đáp án
is_correct	Boolean		Đáp án đúng / sai
note	String		Note để lại của người tạo quiz
question	ObjectId	Khóa ngoại	Người cập nhật
is active	Boolean		Cờ xóa + true: Nếu đã bị xóa

			+ false: Chưa bị xóa
createdAt	TimeStamp		Ngày tạo
updatedAt	TimeStamp		Ngày cập nhật

Bảng 5. Bảng Đáp án

Tên trường	Kiểu dữ liệu	Ràng buộc	Diễn giải
_id	ObjectId	Khóa chính	Mongo Id
account	ObjectId	Khóa ngoại	Id tài khoản làm bài
quiz	ObjectId	Khóa ngoại	Id bài quiz
score	Number		Kết quả điểm số
time	Date		Thời gian làm bài
date	Boolean		Ngày làm bài
createdAt	TimeStamp		Ngày tạo
updatedAt	TimeStamp		Ngày cập nhật

Bảng 6. Bảng Lớn làm

Tên trường	Kiểu dữ liệu	Ràng buộc	Diễn giải
_id	ObjectId	Khóa chính	Mongo Id
classroom_name	String		Tên lớp học
description	String		Mô tả
creator	ObjectId	Khóa ngoại	Id chủ lớp học
allow_edit	Boolean		Cho phép thành viên quản lý quiz
is_active	Boolean		Cờ xóa + true: Nếu đã bị xóa + false: Chưa bị xóa
image	String		Ảnh bìa lớp học
quizzes	[ObjectId]		Các id của quiz
createdAt	TimeStamp		Ngày tạo
updatedAt	TimeStamp		Ngày cập nhật

Bảng 7. Bảng Lớp học

Tên trường	Kiểu dữ liệu	Ràng buộc	Diễn giải
_id	ObjectId	Khóa chính	Mongo Id
classroom	ObjectId	Khóa ngoại	Id lớp học
account	ObjectId	Khóa ngoại	Id thành viên
is_admin	Boolean		Vai trò thành viên trong lớp
createdAt	TimeStamp		Ngày tạo
updatedAt	TimeStamp		Ngày cập nhật

Bảng 8. Bảng Danh sách lớp học

Tên trường	Kiểu dữ liệu	Ràng buộc	Diễn giải
_id	ObjectId	Khóa chính	Mongo Id
folder_name	String		Tên thư mục
description	String		Mô tả
creator	Objectid	Khóa ngoại	Id chủ thư mục
is_active	Boolean		Cờ xóa
quizzes	[Objectid]		Danh sách id các quiz
createdAt	TimeStamp		Ngày tạo
updatedAt	TimeStamp		Ngày cập nhật

Bảng 9. Bảng Thư mục

Tên trường	Kiểu dữ liệu	Ràng buộc	Diễn giải
_id	ObjectId	Khóa chính	Mongo Id
text	String		Mô tả câu hỏi
quiz	Objectid	Khoá ngoại	Id quiz của câu hỏi
is_active	Boolean		Cờ xóa + true: Nếu đã bị xóa + false: Chưa bị xóa
createdAt	TimeStamp		Ngày tạo
updatedAt	TimeStamp		Ngày cập nhật

Bảng 10. Bảng Câu hỏi

Tên trường	Kiểu dữ liệu	Ràng buộc	Diễn giải
_id	ObjectId	Khóa chính	Mongo Id
title	String	Khoá ngoại	Tiêu đề quiz
description	String		Mô tả về quiz
duration	Number		Thời gian làm bài / học bài
is_active	Boolean		Cờ xóa + true: Nếu đã bị xóa + false: Chưa bị xóa
image	String		Ảnh quiz
level	Enum		Mức độ khó/ dễ
is_public	Boolean		Quiz public / private
subject	ObjectId	Khoá ngoại	Id chủ đề
creator	ObjectId	Khoá ngoại	Id người tạo
createdAt	TimeStamp		Ngày tạo
updatedAt	TimeStamp		Ngày cập nhật

Bảng 11. Bảng Quiz

Tên trường	Kiểu dữ liệu	Ràng buộc	Diễn giải
_id	ObjectId	Khóa chính	Mongo Id
quiz	ObjectId	Khoá ngoại	Id quiz
account	ObjectId	Khoá ngoại	Id người làm quiz
quiz_type	[Enum]		Các loại quiz được làm
createdAt	TimeStamp		Ngày tạo
updatedAt	TimeStamp		Ngày cập nhật

Bảng 12. Bảng Record Quiz

Tên trường	Kiểu dữ liệu	Ràng buộc	Diễn giải
_id	ObjectId	Khóa chính	Mongo Id
receiver	ObjectId	Khoá ngoại	Id người nhận
requester	ObjectId	Khoá ngoại	Id người gửi
status	Enum		Trạng thái yêu cầu (Chờ, Chấp nhận, Từ chối)
classroom	ObjectId	Khoá ngoại	Id lớp học được yêu cầu

createdAt	TimeStamp		Ngày tạo
updatedAt	TimeStamp		Ngày cập nhật

Bảng 13. Bảng Yêu cầu

Tên trường	Kiểu dữ liệu	Ràng buộc	Diễn giải
_id	ObjectId	Khóa chính	Mongo Id
subject_name	String		Tên chủ đề
createdAt	TimeStamp		Ngày tạo
updatedAt	TimeStamp		Ngày cập nhật

Bảng 14. Bảng Chủ đề

Tên trường	Kiểu dữ liệu	Ràng buộc	Diễn giải
_id	ObjectId	Khóa chính	Mongo Id
question	ObjectId	Khoá ngoại	Id câu hỏi đánh dấu
account	ObjectId	Khoá ngoại	Id tài khoản đánh dấu
quiz	ObjectId	Khoá ngoại	Id quiz được đánh dấu
createdAt	TimeStamp		Ngày tạo
updatedAt	TimeStamp		Ngày cập nhật

Bảng 15. Bảng Đánh dấu

Tên trường	Kiểu dữ liệu	Ràng buộc	Diễn giải
_id	ObjectId	Khóa chính	Mongo Id
participants	[ObjectId]	Khoá ngoại	Id của các người trong liên hệ
latest_message	String		Tin nhắn mới nhất
sender	ObjectId	Khoá ngoại	Người gửi tin nhắn mới
createdAt	TimeStamp		Ngày tạo
updatedAt	TimeStamp		Ngày cập nhật

Bảng 16. Bảng liên hệ

Tên trường	Kiểu dữ liệu	Ràng buộc	Diễn giải
_id	ObjectId	Khóa chính	Mongo Id
text	String		Nội dung tin nhắn
sender	ObjectId	Khoá ngoại	Id người gửi
contact	ObjectId	Khoá ngoại	Id liên hệ
createdAt	TimeStamp		Ngày tạo
updatedAt	TimeStamp		Ngày cập nhật

Bảng 17. Bảng tin nhắn

2.5 Kết chương

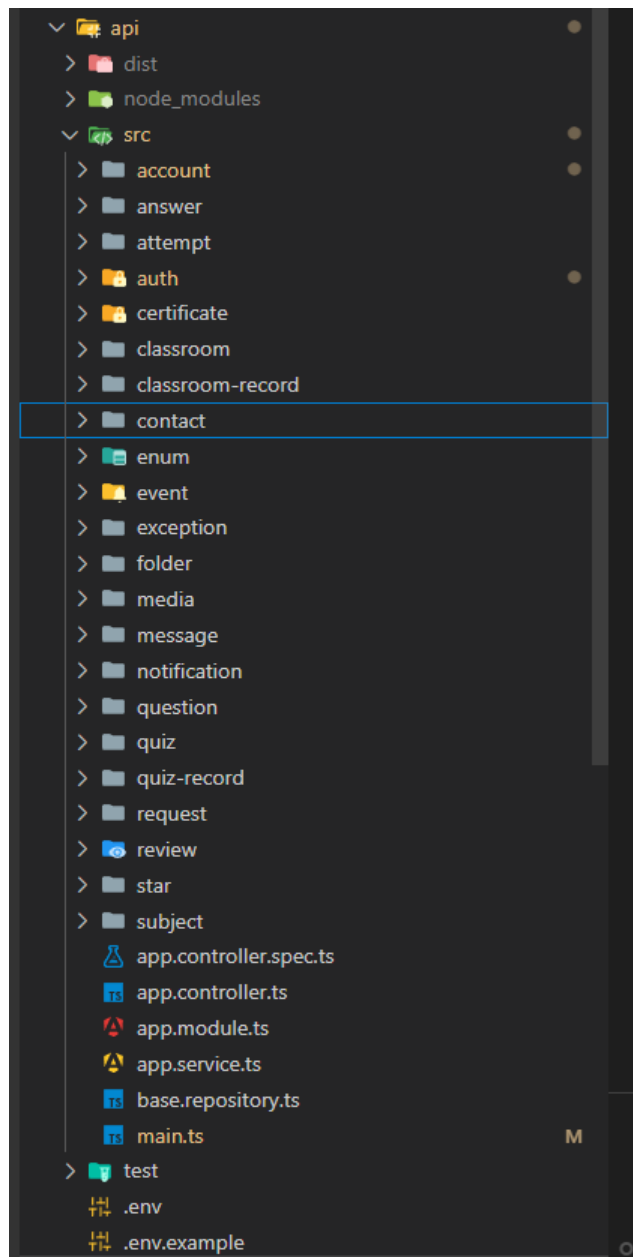
Trong chương này cung cấp cho người đọc cái nhìn tổng quan về nghiệp vụ của hệ thống. Đồng thời phân tích thiết kế hệ thống và mô hình cơ sở dữ liệu.

CHƯƠNG 3. TRIỂN KHAI VÀ ĐÁNH GIÁ KẾT QUẢ

3.1 Xây dựng API

3.1.1 Cấu trúc thư mục

Cấu trúc thư mục tuân theo cấu trúc thư mục của NestJS Framework. Mỗi thư mục sẽ tương ứng với một module hay Feature module.

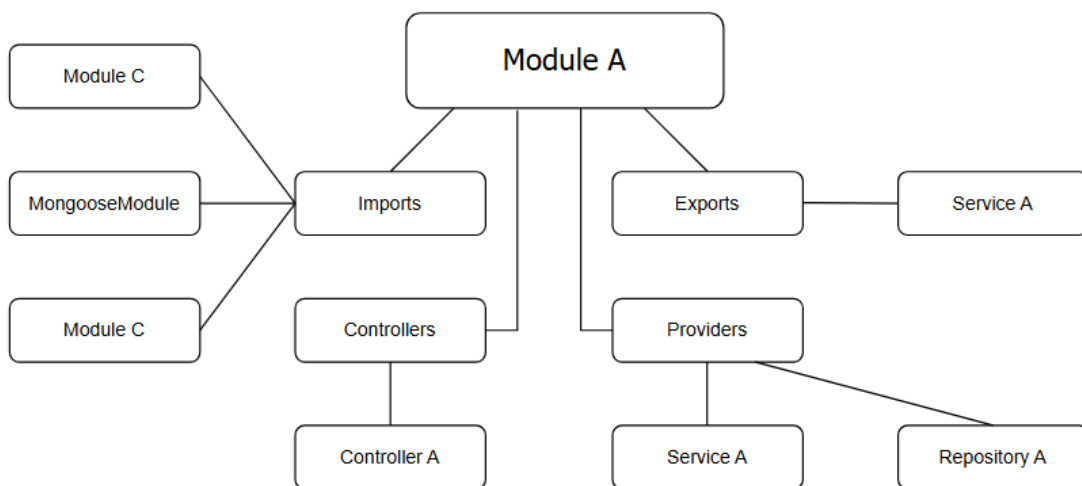


Hình 17. Cấu trúc thư mục

3.1.2 Cấu trúc module

Trong mỗi module có các thuộc tính:

- Imports - các module khác sẽ được import vào module này. Sau khi import vào, module này có thể sử dụng những gì mà các module kia đã exports ra.
- Controllers - các controller được định nghĩa trong module này, và cũng sẽ được khởi tạo bởi Nest injector.
- Providers - các providers sẽ được khởi tạo bởi Nest injector - Trình khởi tạo và có thể được chia sẻ trên toàn module này
- Exports - một bộ con các providers của module này sẽ mở ra cho các module khác có thể sử dụng khi chúng import module này.



Hình 18. Các thuộc tính của module

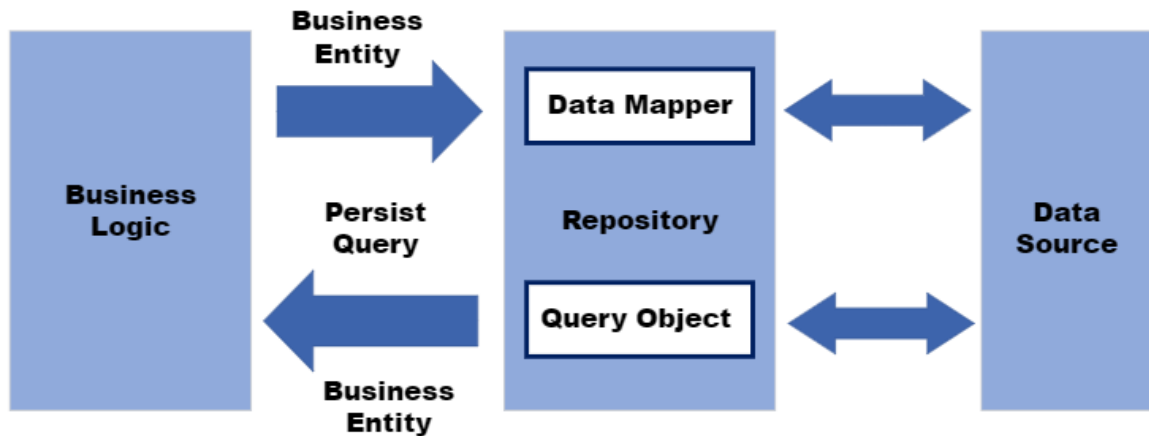
3.1.3 Repository pattern

Repository Pattern là lớp trung gian giữa tầng Business Logic và Data Access, giúp cho việc truy cập dữ liệu chặt chẽ và bảo mật hơn.

Với Repository design pattern, thì việc thay đổi ở code sẽ không ảnh hưởng quá nhiều công sức chỉnh sửa.

Api sử dụng database là MongoDB, việc sử dụng cấu trúc viết repository đem lại nhiều lợi ích trong việc query database. Viết một base repository rồi kế thừa lại các ở các module cần sử dụng thuận tiện và dễ hiểu hơn, giảm được sự phức tạp của code trong service.

Ví dụ để tạo một bản ghi chỉ cần gọi repository.create(dto)



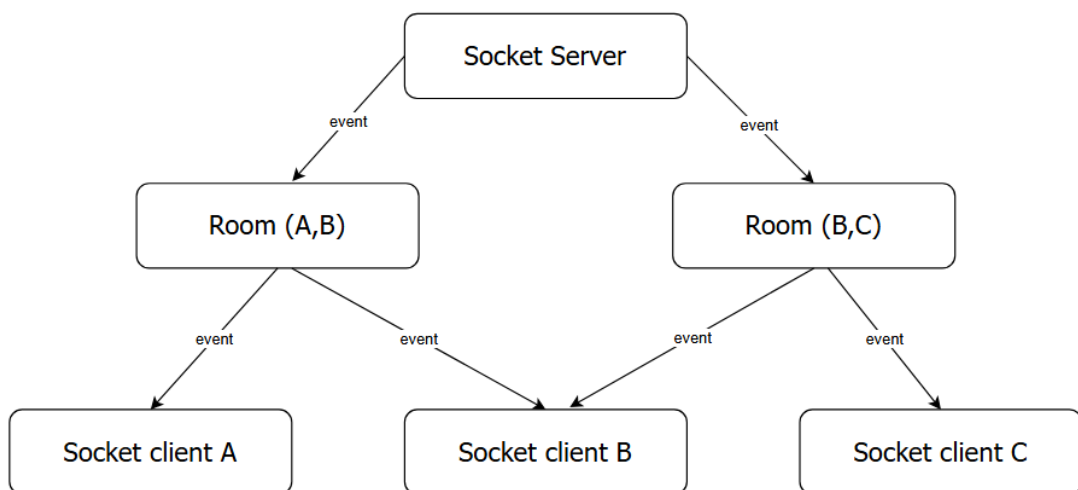
Hình 19. Repository pattern

3.1.4 Socket IO realtime

Đối với chức năng nhắn tin thời gian thực, công nghệ được sử dụng là Socket IO

Trong socket để nhận được event thông qua socket server, client cần vào một room nhất định nơi mà event được đẩy đến room đó.

Ví dụ, người dùng cần tham gia các room để nhận được event tin nhắn tương ứng với room đó. Điều đó được áp dụng bằng cách người dùng (client) sẽ được đẩy vào các room có tên là Id của các liên hệ của người dùng đó.



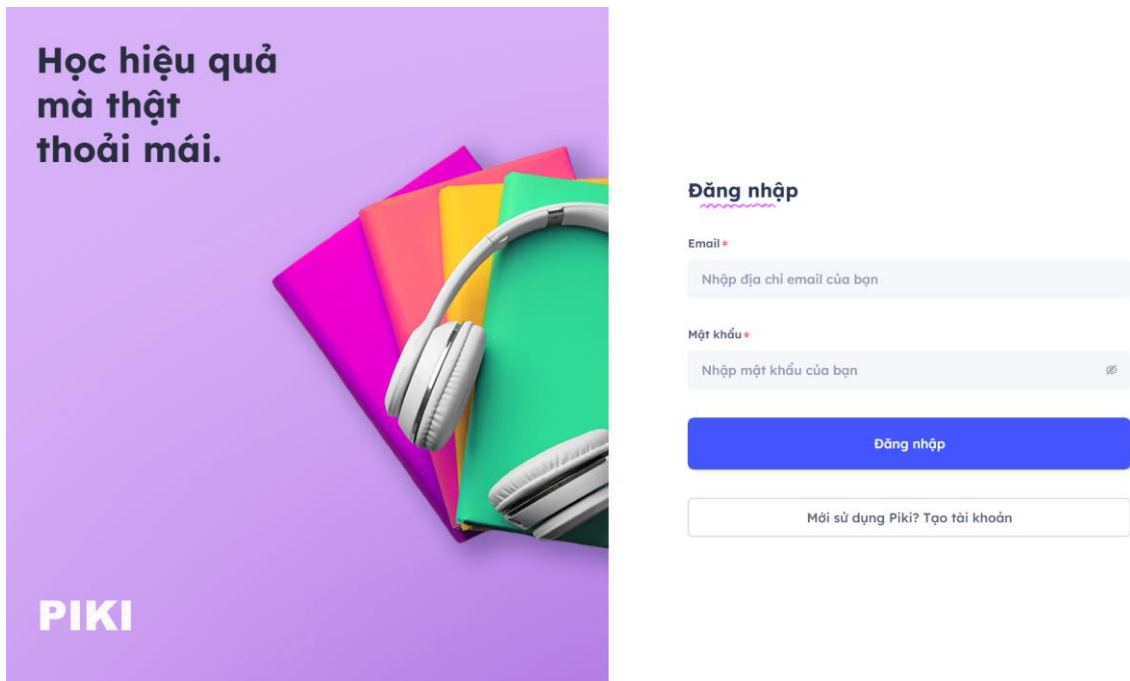
Hình 20. Socket room

3.1.5 Tài liệu Swagger

Piki 1.0 OAS 3.0	
default	^
GET /api	▼
auth	^
POST /api/auth/register	▼
POST /api/auth/login	▼
POST /api/auth/login/admin	▼
POST /api/auth/logout	▼
accounts	^
GET /api/accounts/profile	▼
PATCH /api/accounts/update	▼
GET /api/accounts/by-classroom/{id}	▼
GET /api/accounts/{id}	▼
PATCH /api/accounts/{id}	▼
GET /api/accounts/by-admin/{id}	▼
GET /api/accounts	▼
classrooms	^
POST /api/classrooms	▼
GET /api/classrooms	▼
POST /api/classrooms/join/{id}	▼
POST /api/classrooms/add-quiz	▼
PATCH /api/classrooms/{id}	▼
GET /api/classrooms/{id}	▼
DELETE /api/classrooms/{id}	▼
GET /api/classrooms/created	▼
GET /api/classrooms/joined	▼
GET /api/classrooms/by-quiz/{id}	▼
GET /api/classrooms/check-join/{id}	▼
GET /api/classrooms/by-account/{id}	▼
GET /api/classrooms/by-admin/all	▼
requests	^
GET /api/requests	▼
GET /api/requests/received	▼
GET /api/requests/sent	▼
PATCH /api/requests/{id}	▼
classrooms-records	^

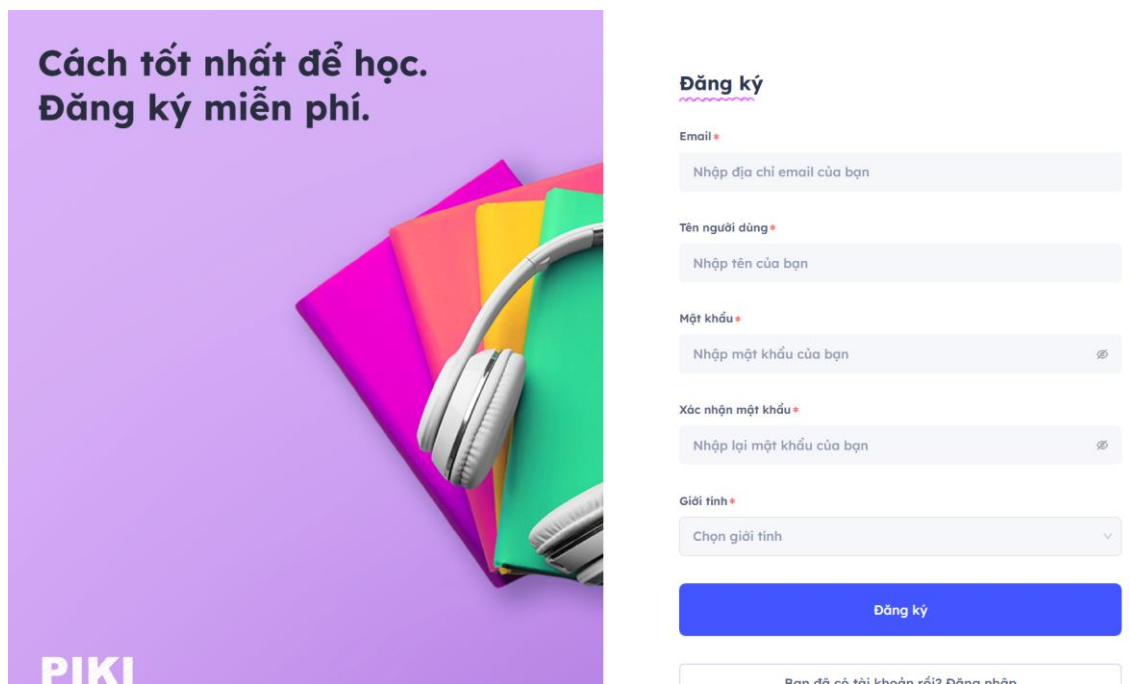
Hình 21. Tài liệu Swagger

3.2 Giao diện chức năng người dùng



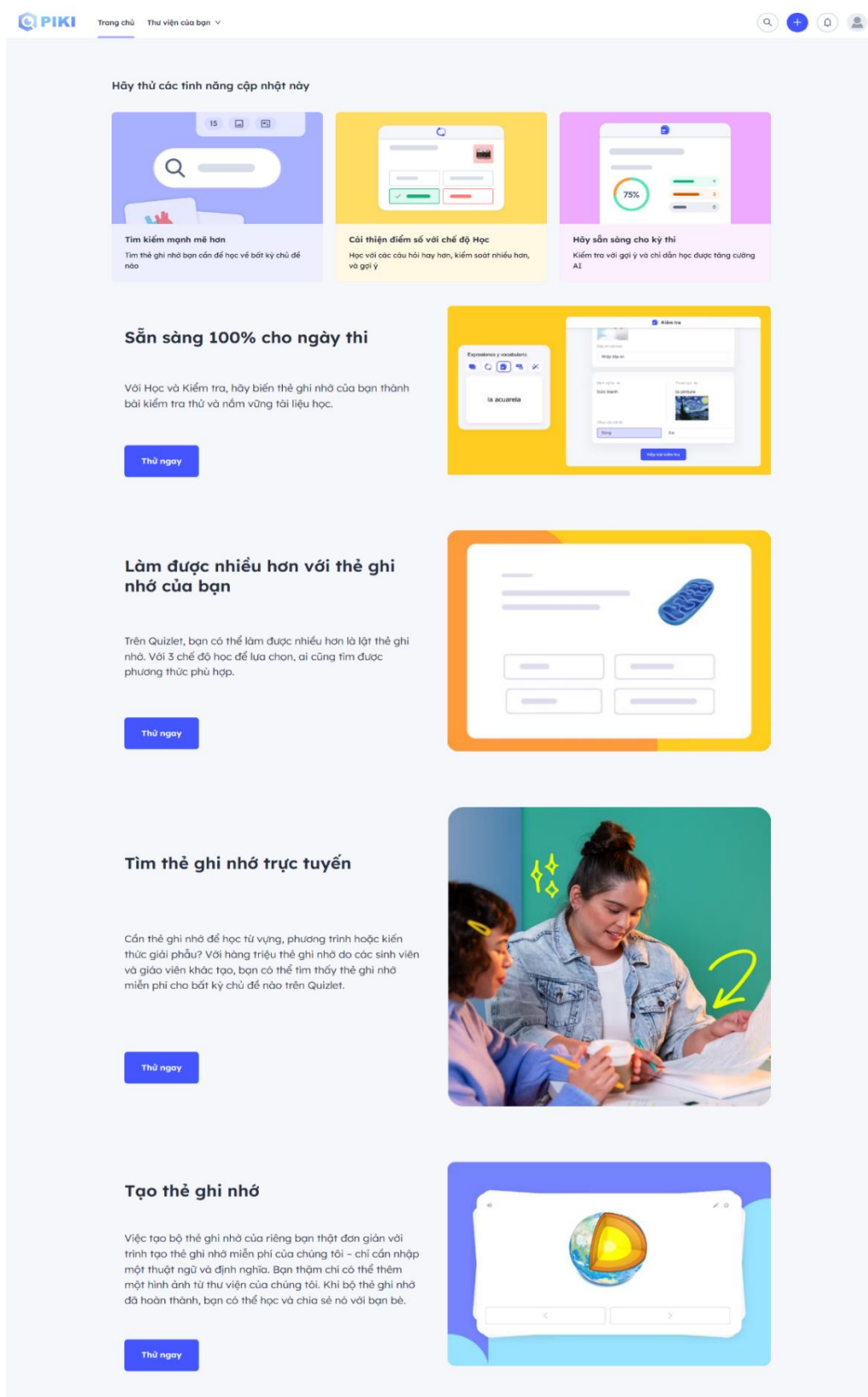
The login interface features a purple background on the left with the text "Học hiệu quả mà thật thoải mái." and the Piki logo. On the right, the "Đăng nhập" (Login) section includes an "Email" field with the placeholder "Nhập địa chỉ email của bạn", a "Mật khẩu" (Password) field with the placeholder "Nhập mật khẩu của bạn" and an eye icon, a blue "Đăng nhập" button, and a link "Mới sử dụng Piki? Tạo tài khoản" (New to Piki? Create account).

Hình 22. Giao diện trang đăng nhập



The registration interface features a purple background on the left with the text "Cách tốt nhất để học. Đăng ký miễn phí." and the Piki logo. On the right, the "Đăng ký" (Register) section includes an "Email" field with the placeholder "Nhập địa chỉ email của bạn", a "Tên người dùng" (Username) field with the placeholder "Nhập tên của bạn", a "Mật khẩu" (Password) field with the placeholder "Nhập mật khẩu của bạn" and an eye icon, a "Xác nhận mật khẩu" (Confirm password) field with the placeholder "Nhập lại mật khẩu của bạn" and an eye icon, a "Giới tính" (Gender) dropdown menu with the placeholder "Chọn giới tính", a blue "Đăng ký" button, and a link "Bạn đã có tài khoản rồi? Đăng nhập" (Already have an account? Login).

Hình 23. Giao diện trang đăng ký



Hình 24. Giao diện trang chủ

Tạo học phần mới

Tiêu đề*

Tiếng Anh Cơ bản

Mô tả

Tiếng Anh Cơ bản cho người nhập môn

Công khai

Thời gian bài test

15

 phút

Cấp độ

Dễ

Chủ đề*

Tiếng Anh

+ Thêm từ học phần đã tạo

Thẻ 1

How much [blank] it cost?

it

they

cause

go

+ Thêm đáp án/ định nghĩa

Hình 25. Giao diện tạo quiz

Vật lý cơ học

Thẻ ghi nhớ

Học

Kiểm tra

☆

Lực nào được đo bằng đơn vị Newton?

A. Lực hút

B. Lực đẩy

C. Lực lực

D. Lực trọng lượng

1 / 7

Chia sẻ

Chỉnh sửa

Nguyễn Thi

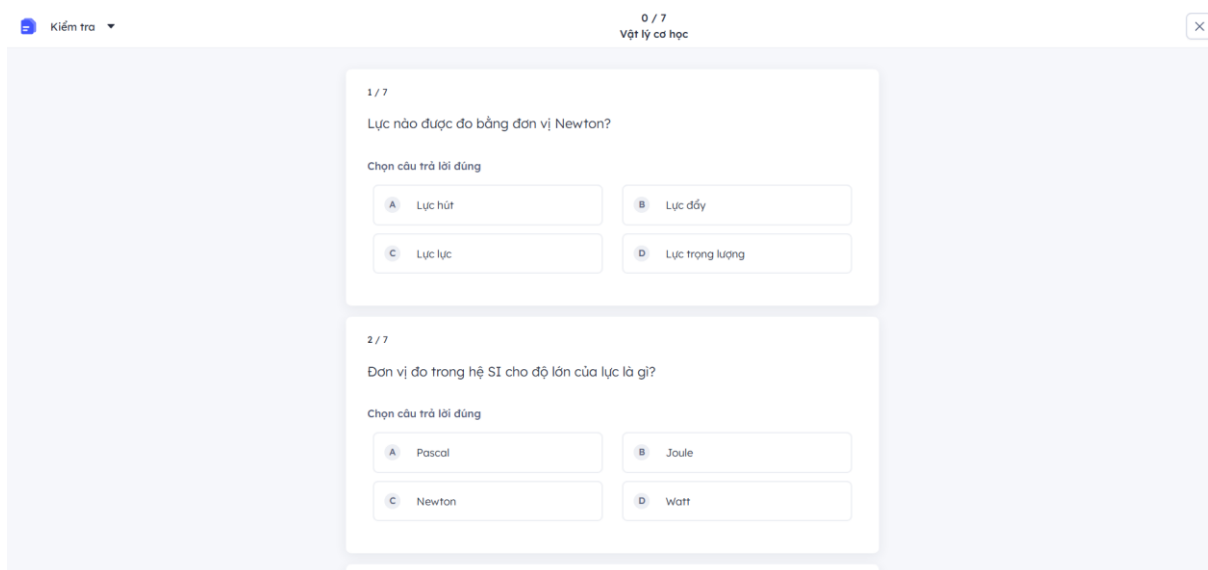
Ngày tạo: 17/06/2024

Hình 26. Giao diện kết quả tạo quiz

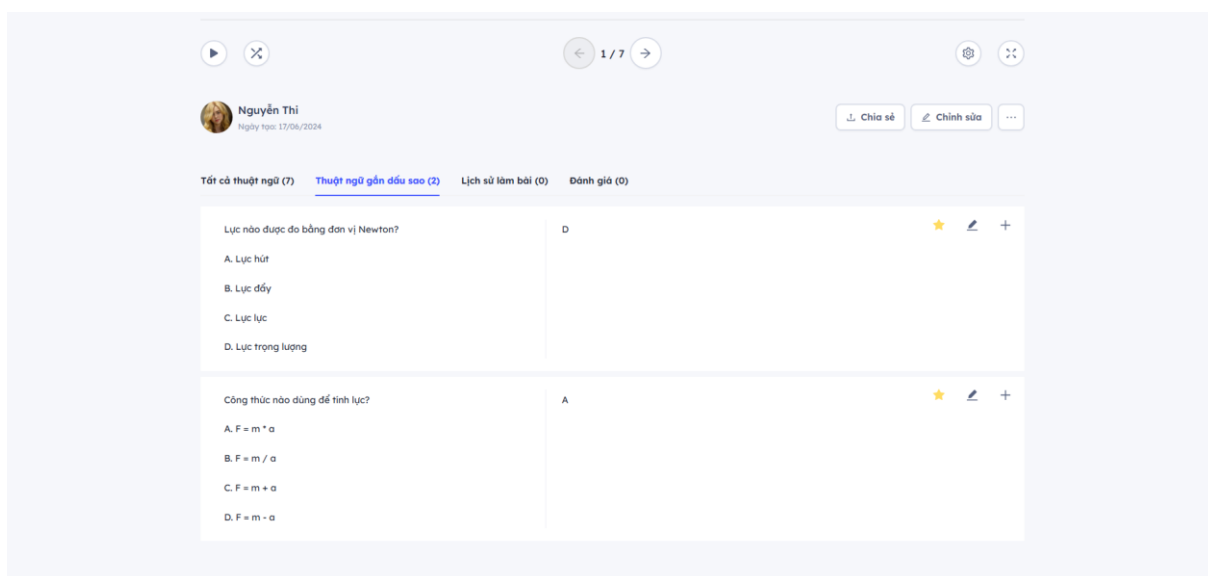
Sinh viên thực hiện: Phan Tiến Đạt

Giảng viên hướng dẫn: TS. Bùi Thị Thanh Thanh

33



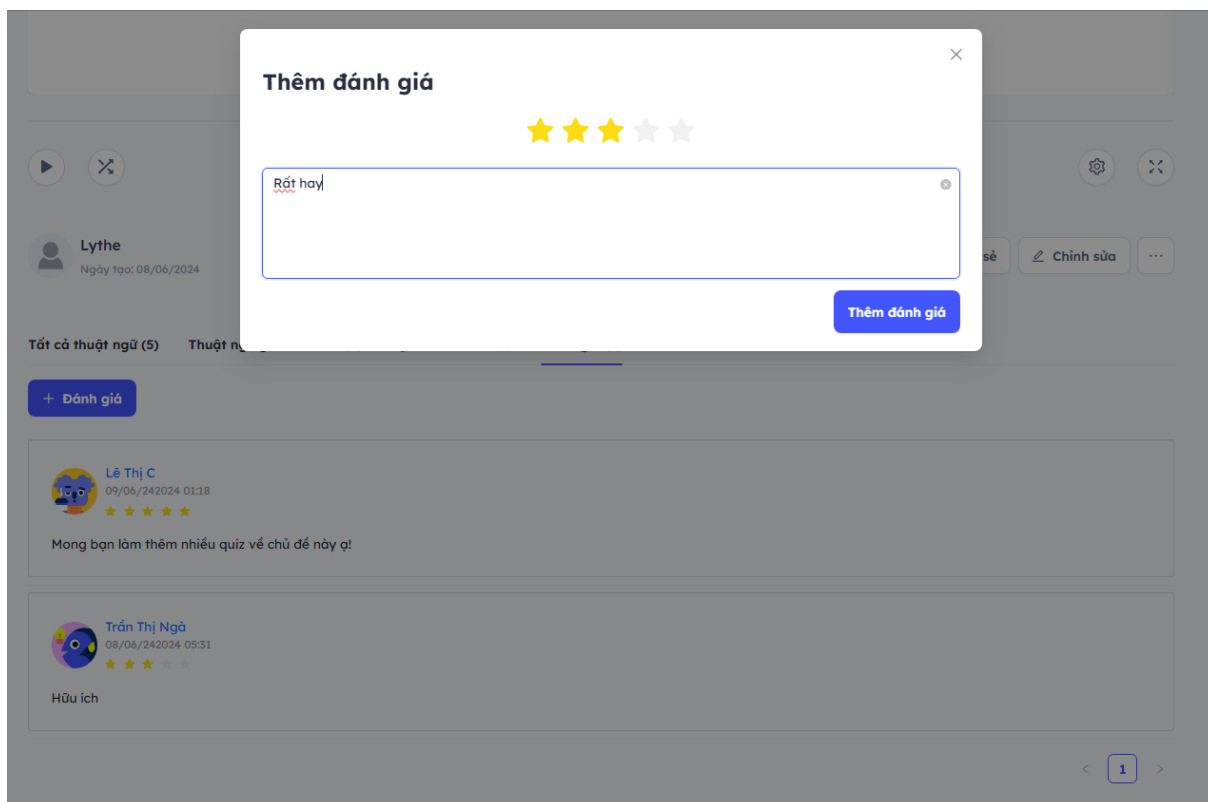
Hình 27. Chức năng kiểm tra quiz



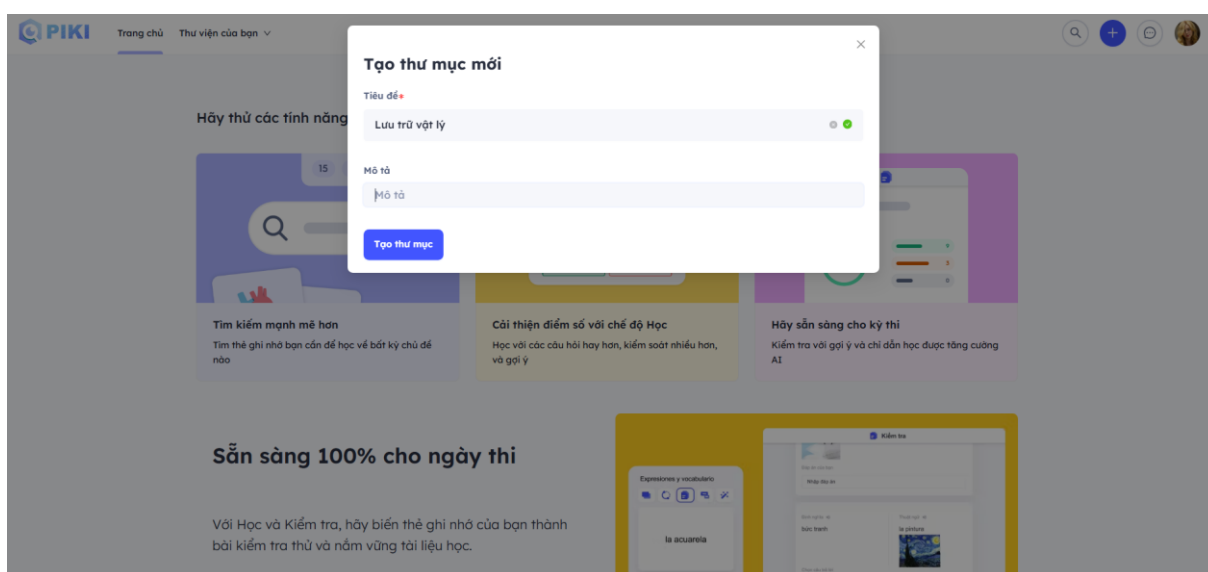
Hình 28. Chức năng gắn dấu sao



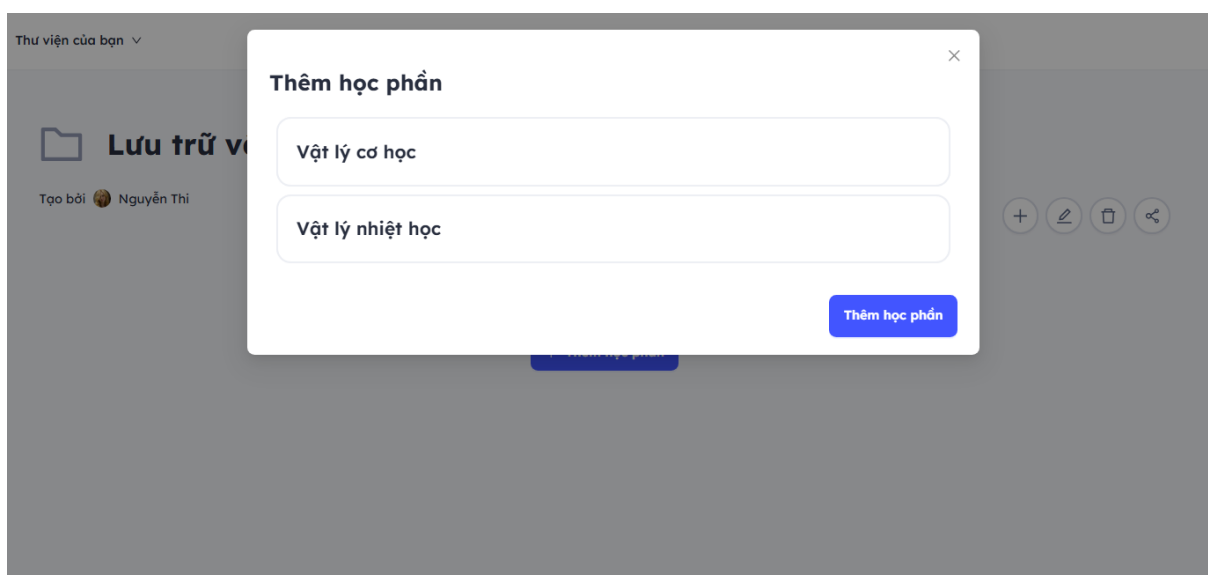
Hình 29. Chức năng xem lịch sử làm bài



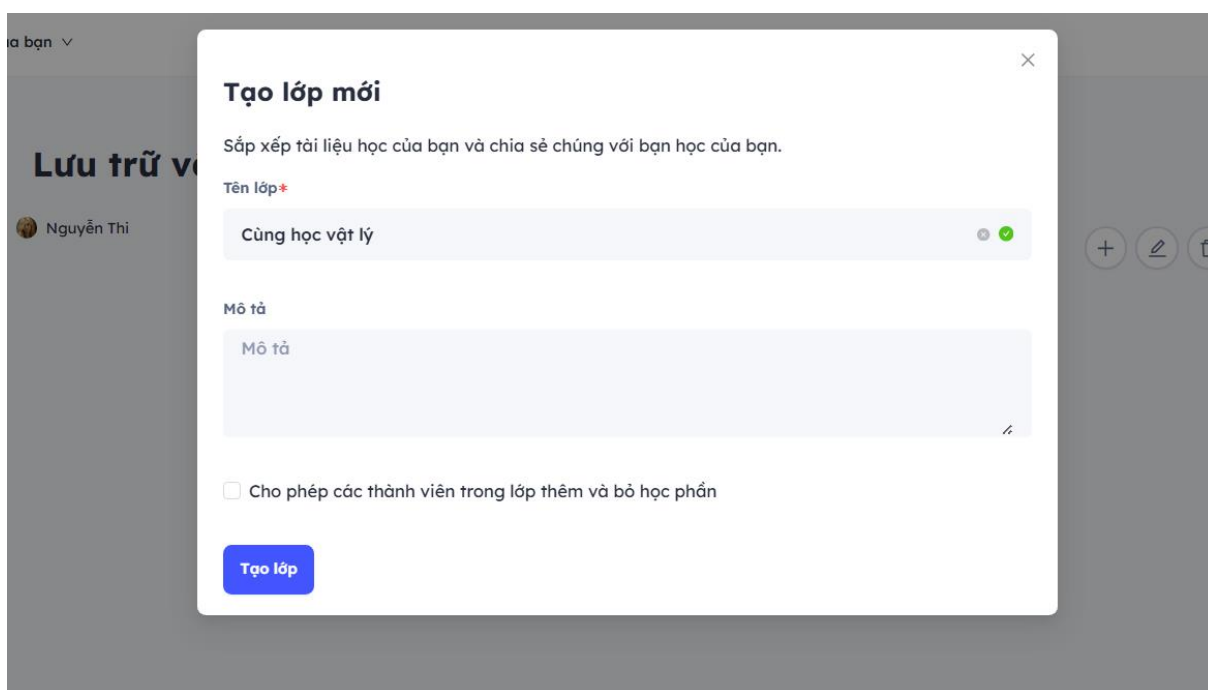
Hình 30. Chức năng đánh giá quiz



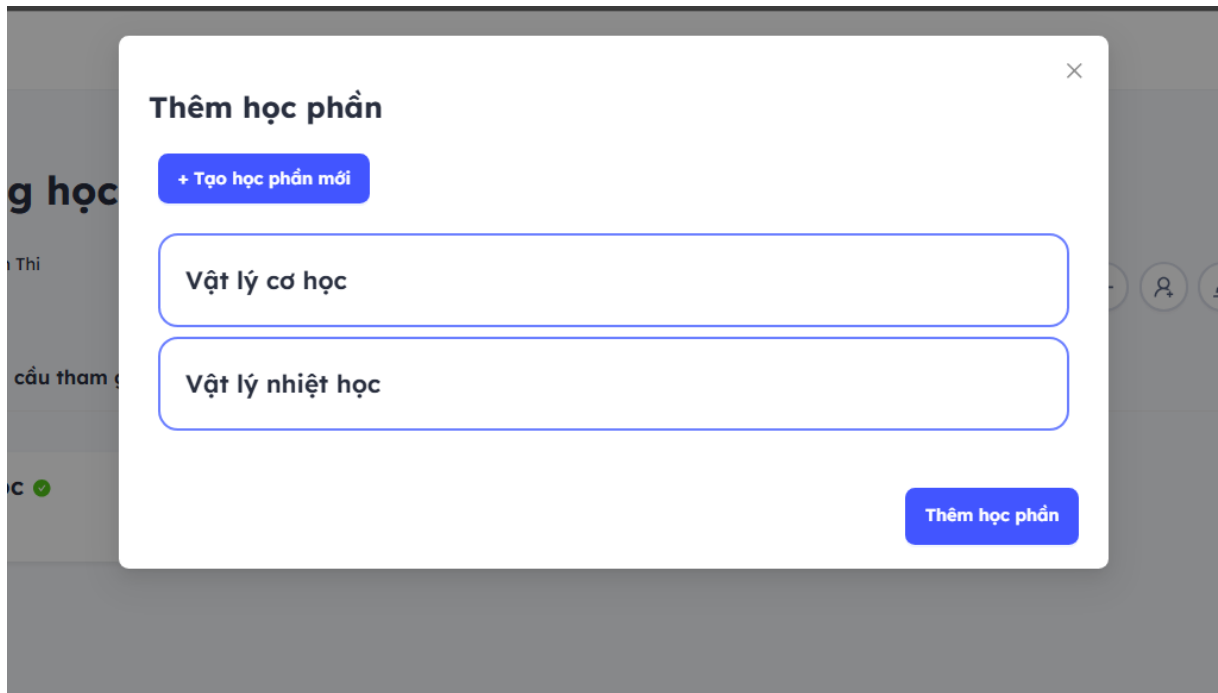
Hình 31. Chức năng tạo thư mục



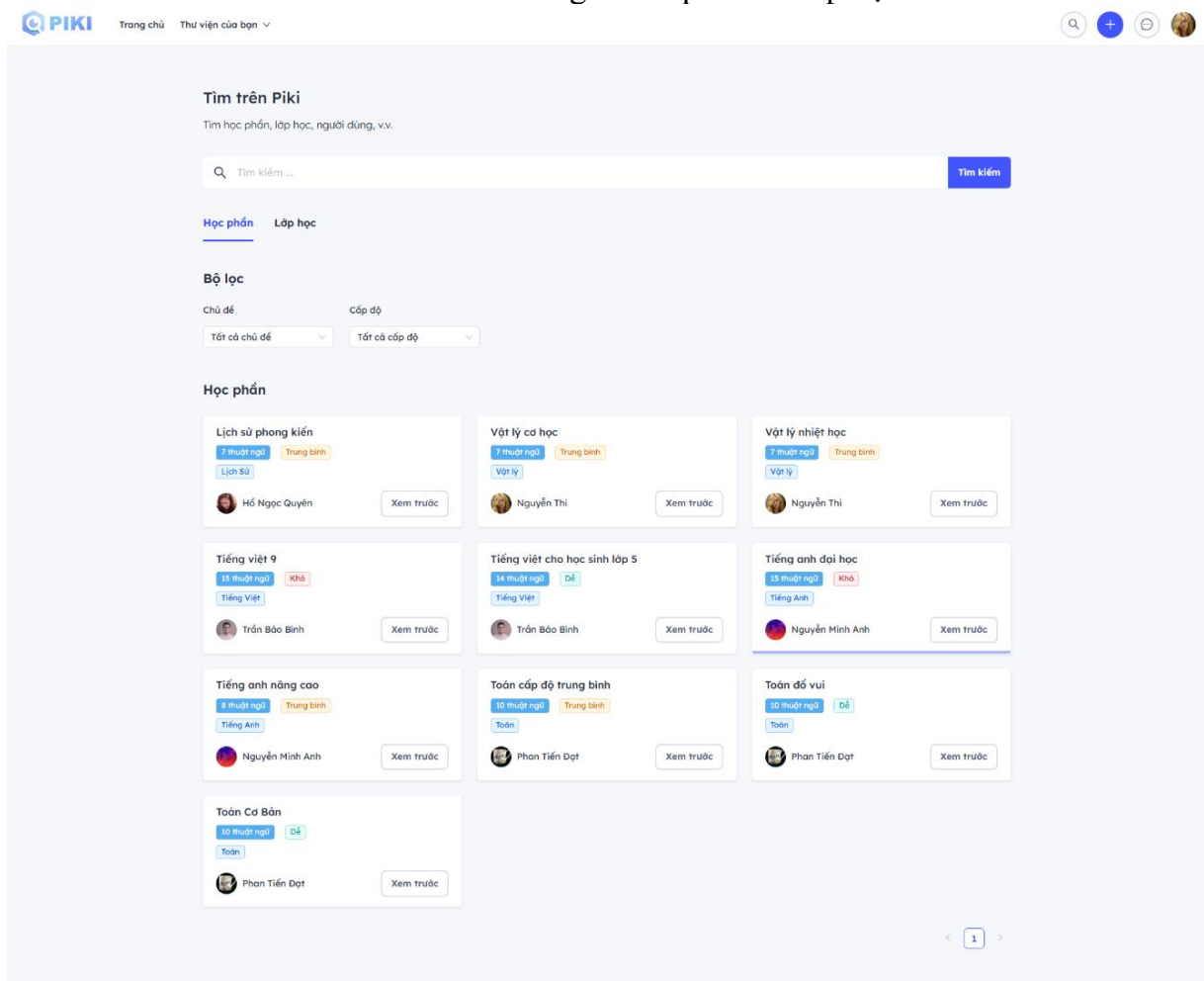
Hình 32. Chức năng thêm quiz vào thư mục



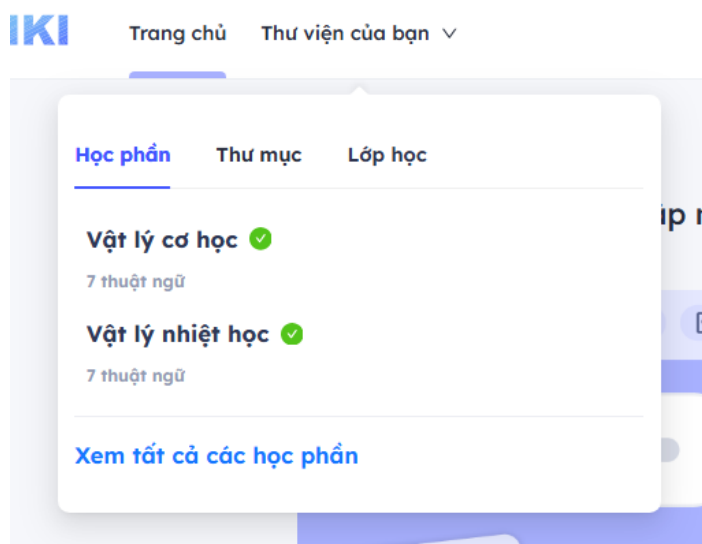
Hình 33. Chức năng tạo lớp học



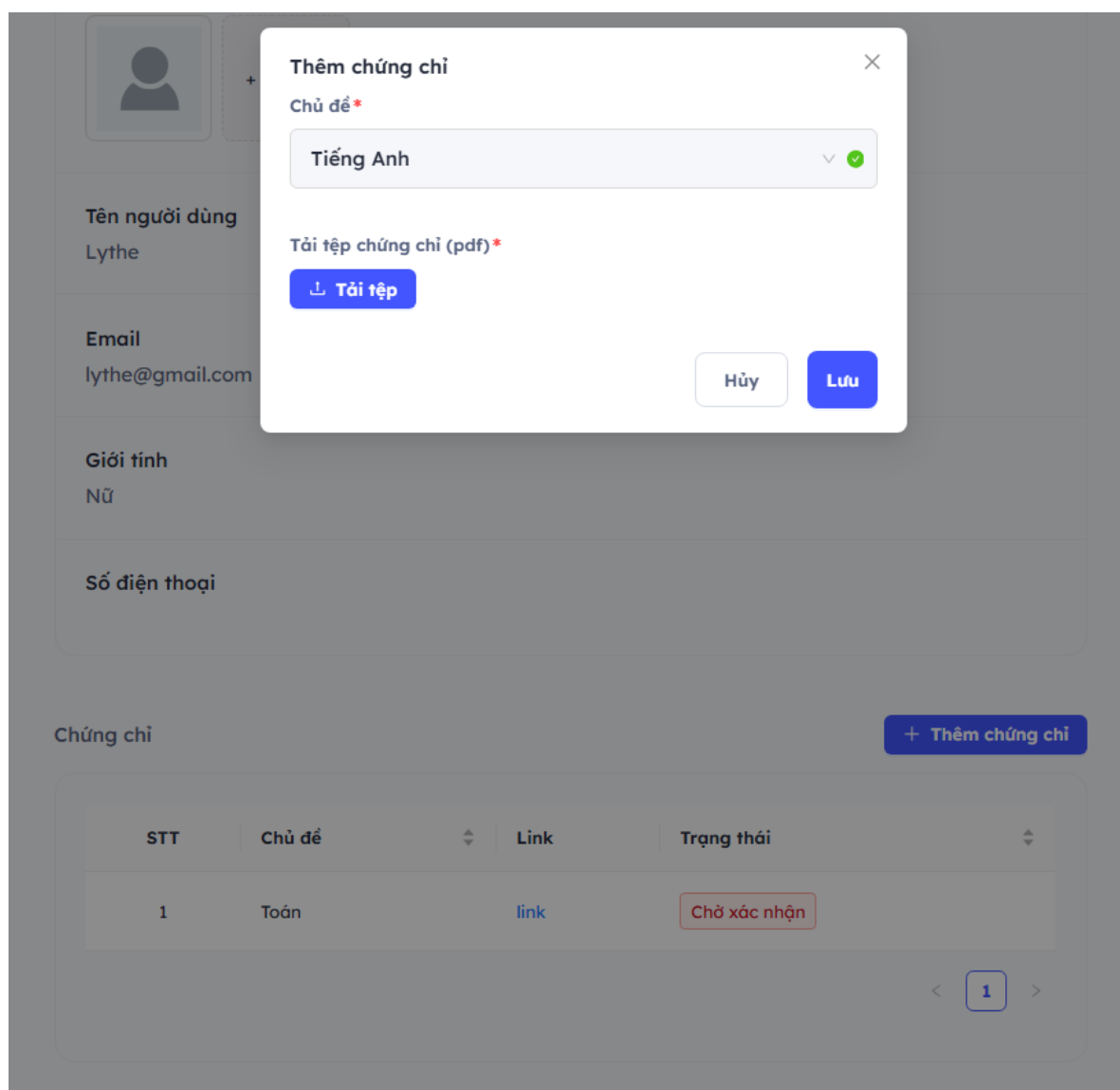
Hình 34. Chức năng thêm quiz vào lớp học



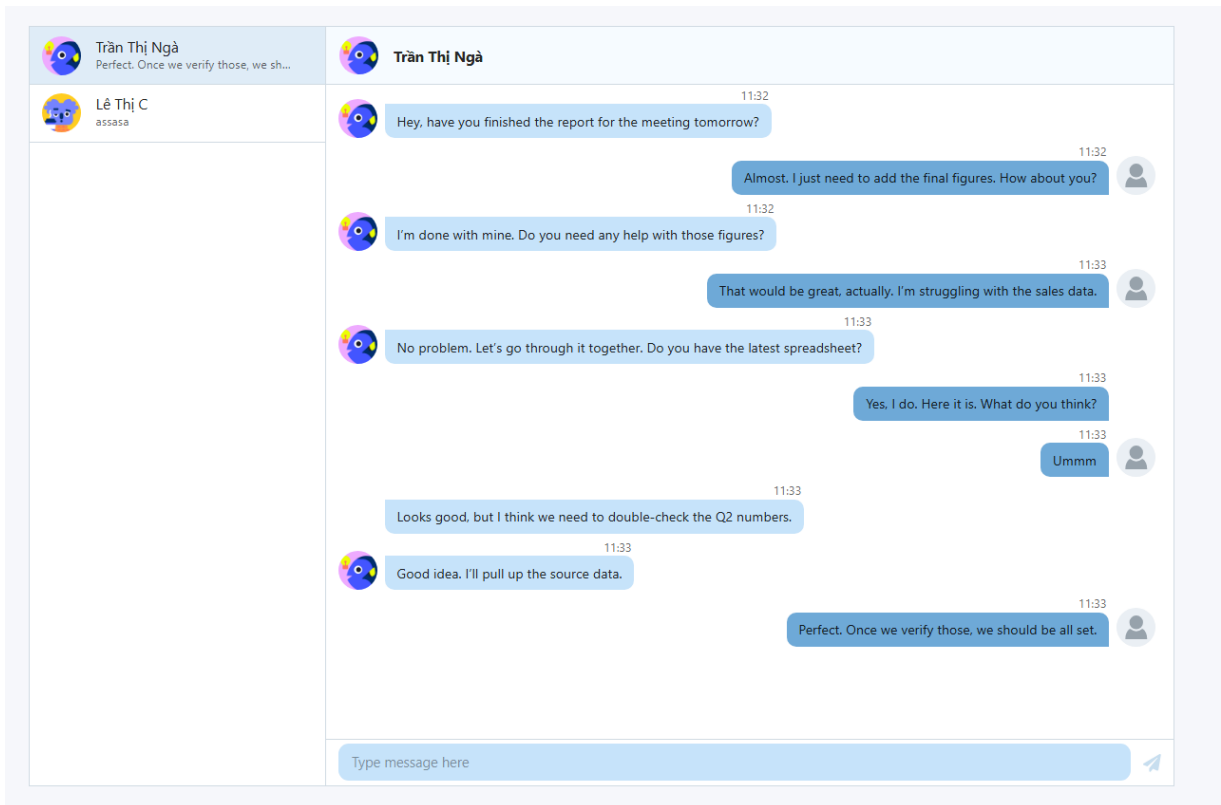
Hình 35. Chức năng tìm kiếm



Hình 36. Giao diện cửa sổ truy cập nhanh

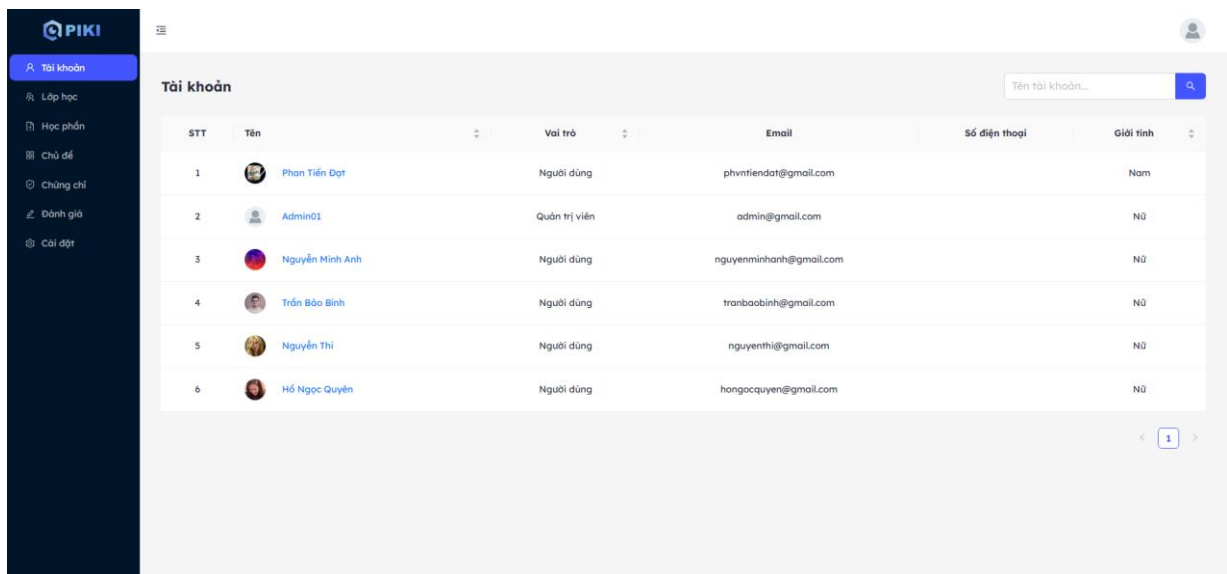


Hình 37. Chức năng thêm chứng chỉ

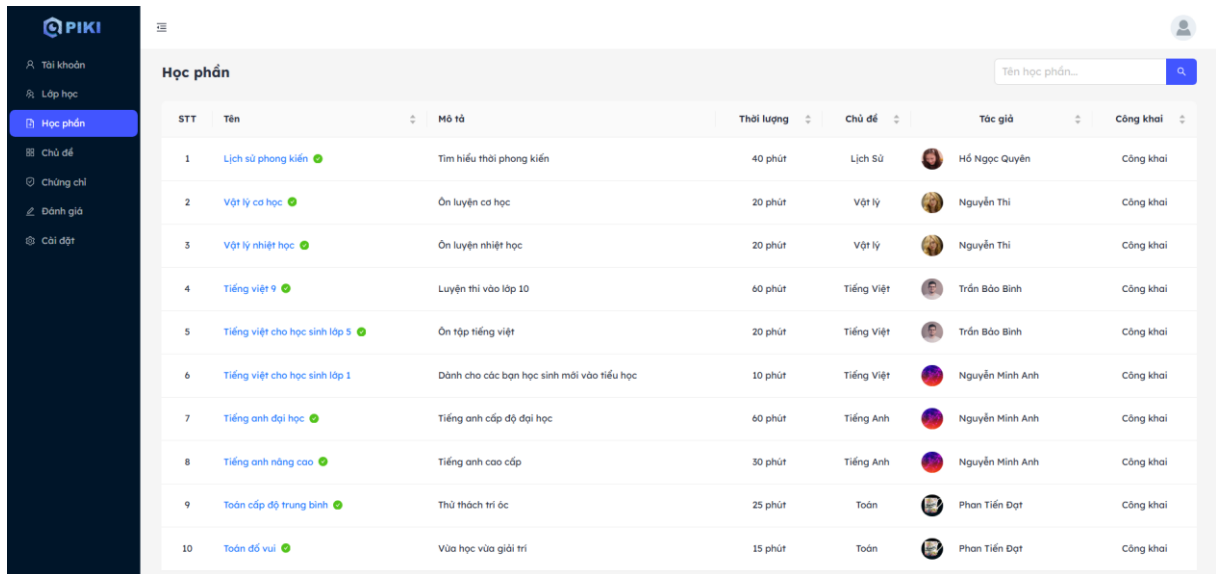


Hình 38. Chức năng nhắn tin thời gian thực

3.3 Giao diện chức năng cho quản trị viên

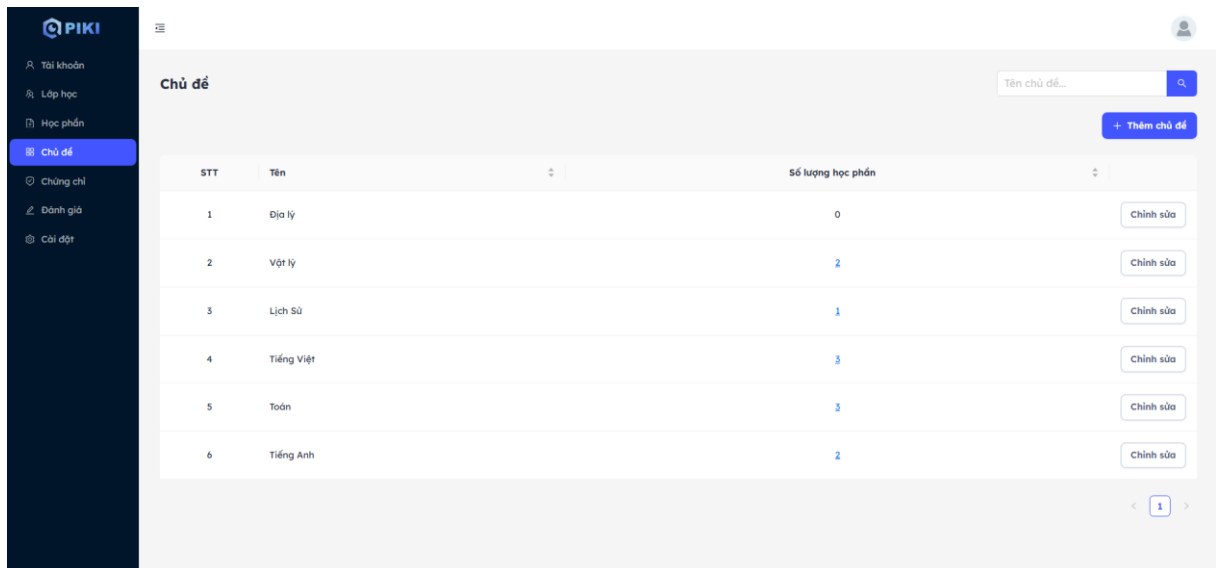


Hình 39. Dashboard tài khoản



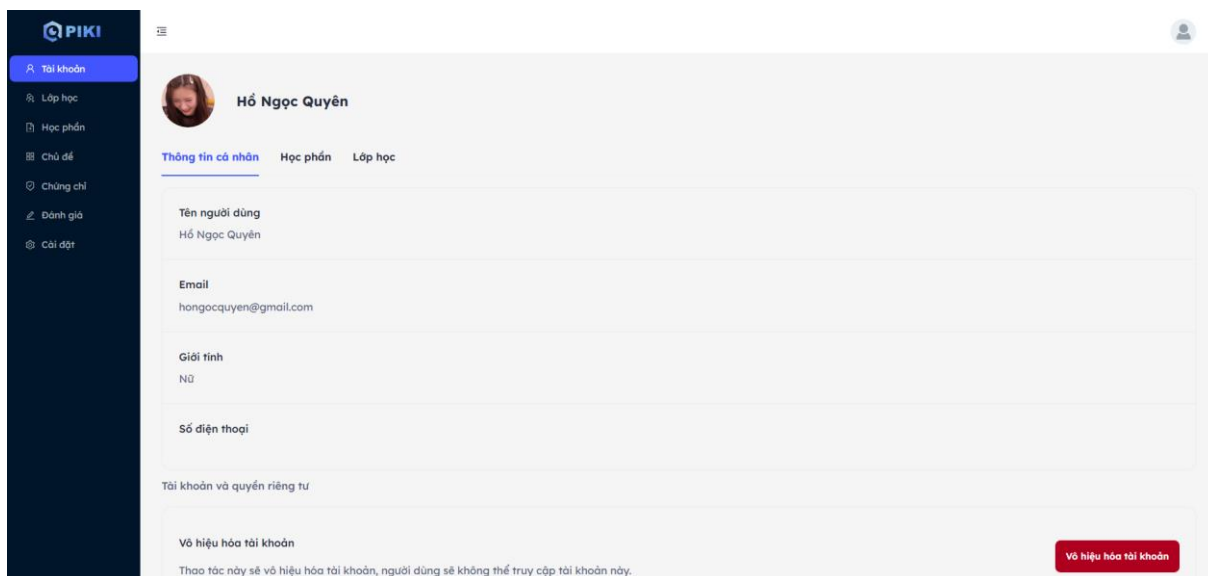
STT	Tên	Mô tả	Thời lượng	Chủ đề	Tác giả	Công khai
1	Lịch sử phong kiến	Tìm hiểu thời phong kiến	40 phút	Lịch Sử	Hồ Ngọc Quyền	Công khai
2	Vật lý cơ học	Ôn luyện cơ học	20 phút	Vật lý	Nguyễn Thị	Công khai
3	Vật lý nhiệt học	Ôn luyện nhiệt học	20 phút	Vật lý	Nguyễn Thị	Công khai
4	Tiếng Việt	Luyện thi vào lớp 10	60 phút	Tiếng Việt	Trần Bảo Bình	Công khai
5	Tiếng Việt cho học sinh lớp 5	Ôn tập tiếng Việt	20 phút	Tiếng Việt	Trần Bảo Bình	Công khai
6	Tiếng Việt cho học sinh lớp 1	Dành cho các bạn học sinh mới vào tiểu học	10 phút	Tiếng Việt	Nguyễn Minh Anh	Công khai
7	Tiếng anh đại học	Tiếng anh cấp độ đại học	60 phút	Tiếng Anh	Nguyễn Minh Anh	Công khai
8	Tiếng anh năng cao	Tiếng anh cao cấp	30 phút	Tiếng Anh	Nguyễn Minh Anh	Công khai
9	Toán cấp độ trung bình	Thử thách trí óc	25 phút	Toán	Phan Tiến Đạt	Công khai
10	Toán đố vui	Vừa học vừa giải trí	15 phút	Toán	Phan Tiến Đạt	Công khai

Hình 40. Chức năng quản lý học phần

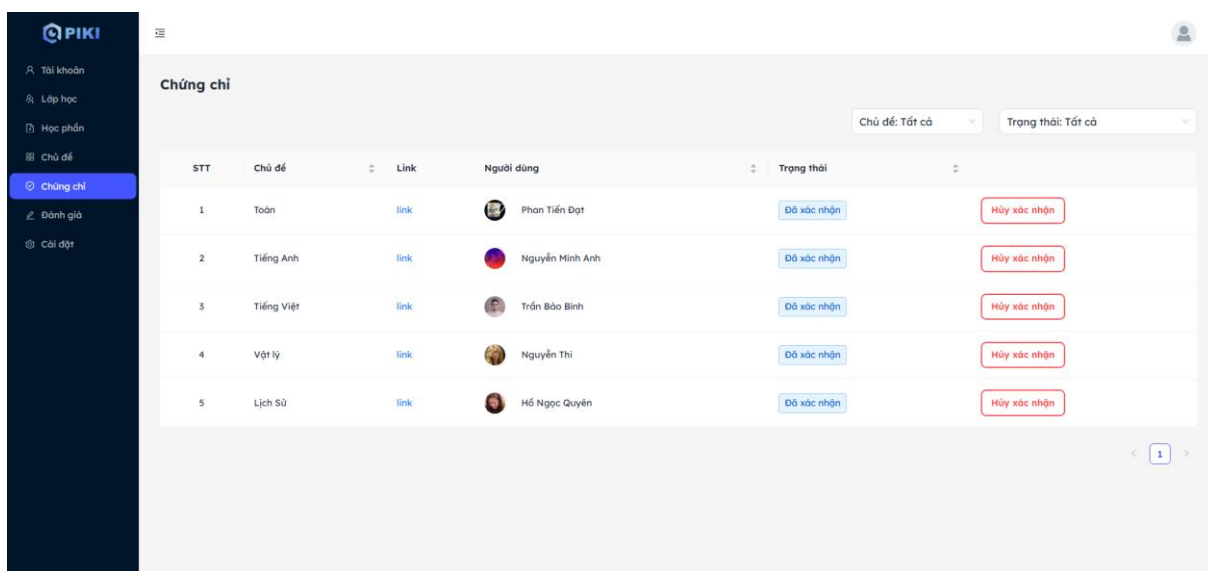


STT	Tên	Số lượng học phần
1	Địa lý	0
2	Vật lý	2
3	Lịch Sử	1
4	Tiếng Việt	3
5	Toán	3
6	Tiếng Anh	2

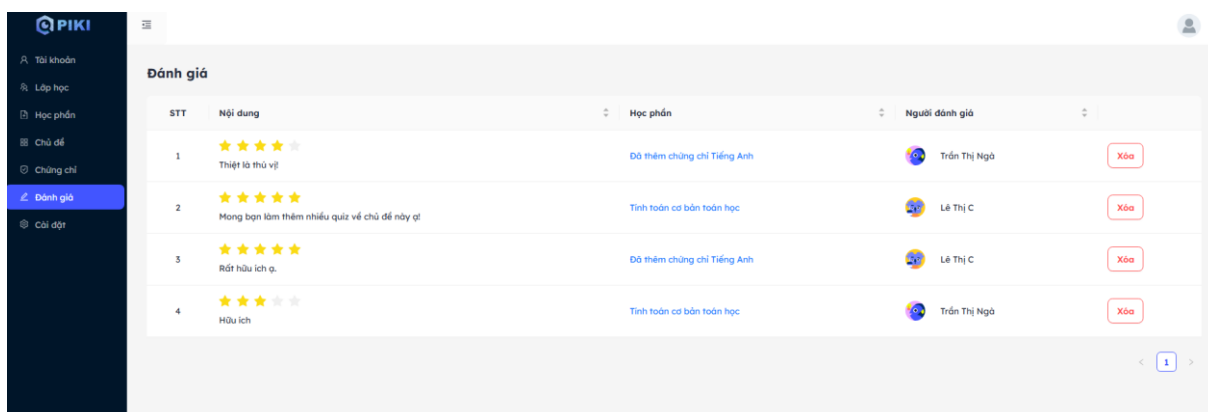
Hình 41. Chức năng quản lý chủ đề



Hình 42. Chức năng vô hiệu hoá tài khoản



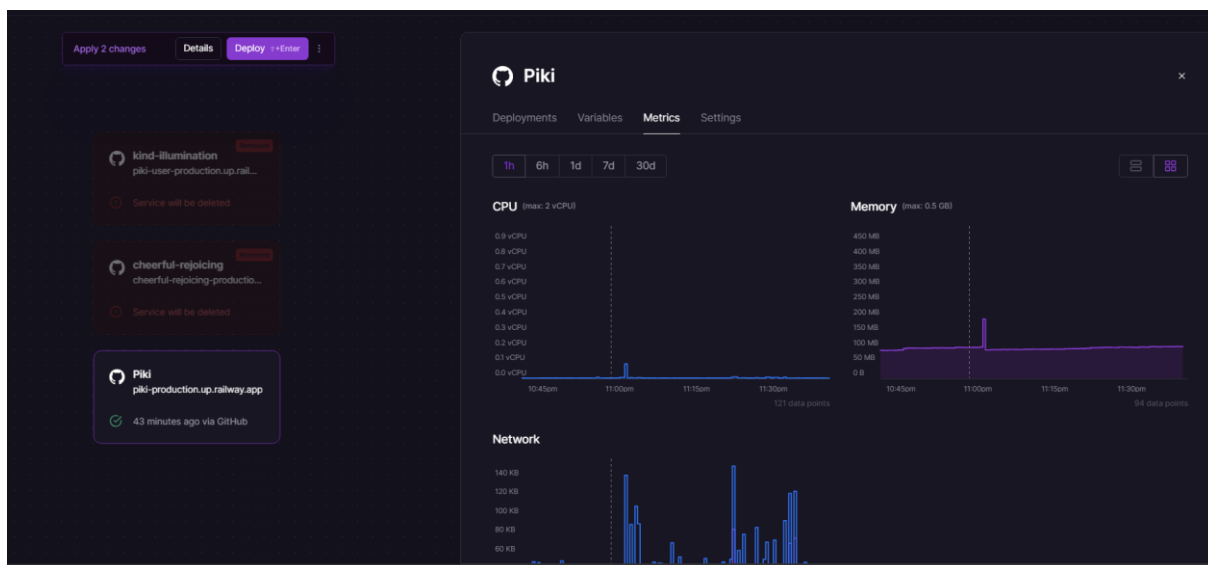
Hình 43. Chức năng quản lý chứng chỉ



Hình 44. Chức năng quản lý đánh giá

3.4 Triển khai

- Công cụ: Railway
- Tài nguyên: 512Mb Ram, 1Gb Shared Disk, 2 vCPU



Hình 45. Deployment trên Railway

3.5 Kết chương

Trong chương này mô tả được các triển khai trong quá trình làm đề tài cũng như kết quả đạt được.

KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

Kết quả đạt được

Trong quá trình nghiên cứu tìm hiểu kiến thức, lý thuyết và triển khai ứng dụng đồ án đã đạt được những kết quả sau:

- Trong quá trình làm và hoàn thành đồ án đã giúp em hiểu thêm cũng như củng cố nắm vững quy trình làm website
- Ứng dụng được RESTful API để triển khai backend ứng dụng
- Xử lý nghiệp vụ
- Website đạt được yêu cầu chính và hoạt động theo yêu cầu đặt ra.

Hạn chế

Ngoài những kết quả đã đạt được trong quá trình thực hiện đề tài. Hệ thống còn một số hạn chế:

- Chưa kiểm thử đầy đủ ứng dụng
- Ứng dụng vẫn ở quy mô nhỏ
- Thời gian hạn chế và kinh nghiệm thực tế, thiếu khảo sát nên chương trình chưa thực sự đáp ứng được nhu cầu thực tế sử dụng

Hướng phát triển

- Cập nhật tính năng cho phép người dùng xem lại chi tiết lịch sử bài làm

TÀI LIỆU THAM KHẢO

- [1] B. T. Huyen, "Tìm hiểu TypeScript và kiến thức cơ bản". [Trực tuyến]
<https://viblo.asia/p/tim-hieu-typescript-va-kien-thuc-co-ban-PmeRQpnyGoB>
[Truy cập ngày 17/6/2024].
- [2] N. T. Tung, "Tìm hiểu về NestJS (Phần 1)". [Trực tuyến]
<https://viblo.asia/p/tim-hieu-ve-nestjs-phan-1-3P0lP0ymlox> [Truy cập ngày
13/6/2024].
- [3] Aurey, "Tìm hiểu về MongoDB". [Trực tuyến]
<https://viblo.asia/p/tim-hieu-ve-mongodb-4P856ajGLY3> [Truy cập ngày
13/6/2024].
- [4] N. Lien, "Socket IO là gì? Hướng dẫn cách dùng Socket IO cơ bản". [Trực
tuyến]
<https://fptshop.com.vn/tin-tuc/danh-gia/socket-io-la-gi-172216> [Truy cập ngày
14/6/2024].
- [5] "RESTful API là gì? Cách thiết kế RESTful API". [Trực tuyến]
<https://topdev.vn/blog/restful-api-la-gi/> [Truy cập ngày 16/6/2024].