

TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI

ĐỒ ÁN TỐT NGHIỆP

Vở bài tập trực tuyến

NGUYỄN VĂN ĐẠI

dai.nv172995@sis.hust.edu.vn

**Ngành Công nghệ thông tin và truyền thông
Chuyên ngành Hệ thống thông tin**

Giảng viên hướng dẫn: TS. Nguyễn Kiêm Hiếu

Chữ kí GVHD

Khoa: Công nghệ thông tin

Trường: Công nghệ thông tin và Truyền thông

HÀ NỘI, 06/2022

LỜI CẢM ƠN

Lời cảm ơn của sinh viên (SV) tới người yêu, gia đình, bạn bè, thầy cô, và chính bản thân mình vì đã chăm chỉ và quyết tâm thực hiện ĐATN để đạt kết quả tốt nhất, nên viết phần cảm ơn ngắn gọn, tránh dùng các từ sáo rỗng, giới hạn trong khoảng 100-150 từ.

TÓM TẮT NỘI DUNG ĐỒ ÁN

Sinh viên viết tóm tắt ĐATN của mình trong mục này, với 200 đến 350 từ. Theo trình tự, các nội dung tóm tắt cần có: (i) Giới thiệu vấn đề (tại sao có vấn đề đó, hiện tại được giải quyết chưa, có những hướng tiếp cận nào, các hướng này giải quyết như thế nào, hạn chế là gì), (ii) Hướng tiếp cận sinh viên lựa chọn là gì, vì sao chọn hướng đó, (iii) Tổng quan giải pháp của sinh viên theo hướng tiếp cận đã chọn, và (iv) Đóng góp chính của ĐATN là gì, kết quả đạt được sau cùng là gì. Sinh viên cần viết thành đoạn văn, không được viết ý hoặc gạch đầu dòng.

MỤC LỤC

CHƯƠNG 1. GIỚI THIỆU ĐỀ TÀI.....	1
1.1 Đặt vấn đề.....	1
1.2 Mục tiêu và phạm vi đề tài.....	1
1.3 Định hướng giải pháp.....	2
1.4 Bố cục đồ án	2
CHƯƠNG 2. KHẢO SÁT VÀ PHÂN TÍCH YÊU CẦU.....	4
2.1 Khảo sát hiện trạng	4
2.2 Tổng quan chức năng	5
2.2.1 Biểu đồ use case tổng quát	5
2.2.2 Biểu đồ use case phân rã Quản lý vở	6
2.2.3 Biểu đồ use case phân rã Quản lý bài tập.....	7
2.2.4 Biểu đồ use case phân rã Tương tác với bài tập	8
2.2.5 Quy trình nghiệp vụ	8
2.3 Đặc tả chức năng	11
2.3.1 Đặc tả use case Thêm vở.....	11
2.3.2 Đặc tả use case Thêm bài tập mới	11
2.3.3 Đặc tả use case Chia sẻ bài tập cho thành viên khác	12
2.3.4 Đặc tả use case Xem bài tập	12
CHƯƠNG 3. CÔNG NGHỆ SỬ DỤNG.....	13
3.1 .NET 6.0.....	13
3.2 Vue.js	13
3.3 MariaDB	16
3.4 MinIO Object Storage System.....	16
3.5 Docker.....	16

3.6 Kubernetes.....	17
CHƯƠNG 4. THỰC NGHIỆM VÀ ĐÁNH GIÁ	18
4.1 Thiết kế kiến trúc.....	18
4.1.1 Lựa chọn kiến trúc phần mềm	18
4.1.2 Thiết kế tổng quan.....	18
4.1.3 Thiết kế chi tiết gói	20
4.2 Thiết kế chi tiết.....	22
4.2.1 Thiết kế giao diện	22
4.2.2 Thiết kế lớp	26
4.2.3 Thiết kế cơ sở dữ liệu	31
4.3 Xây dựng ứng dụng.....	36
4.3.1 Thư viện và công cụ sử dụng	36
4.3.2 Kết quả đạt được	38
4.3.3 Minh họa các chức năng chính	38
4.4 Kiểm thử.....	38
4.5 Triển khai	38
CHƯƠNG 5. CÁC GIẢI PHÁP VÀ ĐÓNG GÓP NỔI BẬT.....	39
CHƯƠNG 6. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN	40
6.1 Kết luận	40
6.2 Hướng phát triển.....	40

DANH MỤC HÌNH VẼ

Hình 2.1	Biểu đồ use case tổng quan	5
Hình 2.2	Biểu đồ use case phân rã Quản lý vở	6
Hình 2.3	Biểu đồ use case phân rã Quản lý bài tập	7
Hình 2.4	Biểu đồ use case phân rã Tương tác với bài tập	8
Hình 2.5	Quy trình nghiệp vụ Đăng tải bài tập	9
Hình 2.6	Quy trình nghiệp vụ Chia sẻ bài tập	10
Hình 3.1	Vòng đời của một Vue Single-File Component	15
Hình 4.1	Kiến trúc phần mềm	18
Hình 4.2	Thiết kế tổng quan backend	19
Hình 4.3	Thiết kế tổng quan frontend	20
Hình 4.4	Thiết kế chi tiết gói xử lý nghiệp vụ đăng tải bài tập	21
Hình 4.5	Bố cục giao diện trên trình duyệt web của màn hình desktop	23
Hình 4.6	Thiết kế nút của ứng dụng	23
Hình 4.7	Vị trí thông điệp phản hồi	24
Hình 4.8	Minh họa thiết kế giao diện màn hình Thêm bài tập	25
Hình 4.9	Thiết kế lớp Excercise	26
Hình 4.10	Thiết kế lớp ExcerciseService	27
Hình 4.11	Thiết kế lớp EFCoreExcerciseRepository	28
Hình 4.12	Thiết kế lớp EFCoreSharingRepository	28
Hình 4.13	Biểu đồ tuần tự của use case Tạo bài tập	29
Hình 4.14	Biểu đồ tuần tự của use case Chia sẻ bài tập	30
Hình 4.15	Biểu đồ thực thể - liên kết	31
Hình 4.16	Thiết kế cơ sở dữ liệu	32

DANH MỤC BẢNG BIỂU

Bảng 2.1	Đặc tả use case Thêm vở	11
Bảng 2.2	Dữ liệu đầu vào của vở bài tập mới	11
Bảng 2.3	Đặc tả use case Thêm bài tập mới	11
Bảng 2.4	Dữ liệu đầu vào của bài tập mới	12
Bảng 2.5	Đặc tả use case Chia sẻ bài tập cho thành viên khác	12
Bảng 2.6	Đặc tả use case Xem bài tập	12
Bảng 4.1	Đặc tả màn hình mà ứng dụng hướng tới	22
Bảng 4.2	Bảng màu theo quy ước của Bootstrap	25
Bảng 4.3	Đặc tả màn hình Thêm bài tập	25
Bảng 4.4	Chi tiết bảng Users	32
Bảng 4.5	Chi tiết bảng Workbooks	33
Bảng 4.6	Chi tiết bảng Topics	33
Bảng 4.7	Chi tiết bảng Exercises	33
Bảng 4.8	Chi tiết bảng Answers	34
Bảng 4.9	Chi tiết bảng Sharings	34
Bảng 4.10	Chi tiết bảng Answers	35
Bảng 4.11	Chi tiết bảng Notifications	35
Bảng 4.12	Chi tiết bảng Votes	36
Bảng 4.13	Chi tiết bảng Tags	36
Bảng 4.14	Chi tiết bảng ExerciseTag	36
Bảng 4.15	Danh sách các công cụ sử dụng	37
Bảng 4.16	Danh sách các thư viện sử dụng cho backend	37
Bảng 4.17	Danh sách các thư viện sử dụng cho frontend	37

DANH MỤC THUẬT NGỮ VÀ TỪ VIẾT TẮT

Thuật ngữ	Ý nghĩa
API	Giao diện lập trình ứng dụng (Application Programming Interface)
EUD	Phát triển ứng dụng người dùng cuối(End-User Development)
GWT	Công cụ lập trình Javascript bằng Java của Google (Google Web Toolkit)
HTML	Ngôn ngữ đánh dấu siêu văn bản (HyperText Markup Language)
IaaS	Dịch vụ hạ tầng

CHƯƠNG 1. GIỚI THIỆU ĐỀ TÀI

1.1 Đặt vấn đề

Trong hoạt động học tập hằng ngày, học sinh cần phải giải quyết nhiều bài tập. Thông thường bài tập được ghi chú lại trong giấy, vở và được bảo quản thủ công. Ghi chú bài tập theo cách này cơ bản đáp ứng được nhu cầu học tập của cá nhân học sinh. Tuy nhiên, nếu chỉ lưu trữ bài tập bằng giấy vở thông thường, hoạt động chia sẻ bài tập giữa các học sinh với nhau vì cùng lúc chỉ có một học sinh khác được chia sẻ. Ngoài ra, khi học sinh khóa trước muốn truyền đạt lại các giấy vở ghi bài tập này cho học sinh khóa sau, việc tra cứu, ôn tập cũng gặp khó khăn vì số lượng bài tập lớn, chỉ có thể tra cứu thủ công và bất tiện trong bảo quản.

Từ năm 2017, kỳ thi Trung học phổ thông quốc gia được chuyển sang hình thức thi trắc nghiệm đối với hầu hết các môn học, trừ môn Ngữ văn. Thông qua hình thức thi trắc nghiệm, số lượng câu hỏi được tăng lên đáng kể. Số lượng câu hỏi lớn khiến đề bài có phạm vi kiến thức rộng và các dạng bài mới lạ. Thời gian làm bài lại gấp rút nên học sinh ít có điều kiện ghi chép bài tập của mình ra vở viết một cách rõ ràng mà chỉ ghi chú nhanh ra đề bài, giấy nháp... gây khó khăn trong việc tìm lại các bài giải cũ khi gặp các bài tập tương tự trong tương lai.

Với hoàn cảnh nêu trên, học sinh cần một cách thức mới để ghi chú và lưu trữ được số lượng lớn các bài tập của mình, cung cấp tra cứu, đem lại khả năng chia sẻ rộng rãi và một không gian thảo luận và trao đổi.

Giải quyết được vấn đề này, học sinh không chỉ có được một công cụ tiện lợi để lưu trữ, ghi chú, chia sẻ bài tập, khắc phục được nhược điểm của vở giấy thông thường. Nhờ tính năng chia sẻ và tương tác, học sinh có thể tham khảo được bài tập của nhau, tìm hiểu được nhiều bài tập và trao đổi trực tiếp về các bài hay bài khó.

1.2 Mục tiêu và phạm vi đề tài

Hiện nay chưa có hệ thống nào giúp học sinh lưu trữ, phân loại và chia sẻ bài tập.

ĐATN xây dựng ứng dụng web phục vụ cho nhóm các đối tượng là giáo viên, học sinh và những người dùng khác có nhu cầu lưu trữ, chia sẻ và trao đổi bài tập. ĐATN nhắm tới hai mục tiêu là (1) hỗ trợ người dùng lưu lại các bài tập đã giải với khả năng phân loại theo môn học, chủ điểm, dạng bài, độ khó..., (2) cung cấp tính năng chia sẻ và (3) đem đến khả năng tương tác cho phép tương tác với bài tập của người dùng khác.

1.3 Định hướng giải pháp

ĐATN Ứng dụng web Vở bài tập trực tuyến được thiết kế và xây dựng dựa trên việc phân tích nhu cầu lưu trữ, phân loại, chia sẻ và tương tác với bài tập của học sinh. Từ những phân tích đó, ĐATN thực hiện (i) xây dựng chức năng đăng tải và lưu trữ bài tập, (ii) hỗ trợ người dùng phân loại và hệ thống hoá bài tập, (iii) cung cấp khả năng tương tác phong phú giữa người dùng với bài tập.

ĐATN thực hiện phân quyền theo người dùng đối với bài tập và các cấp phân loại bài tập (gồm có Vở, Chủ đề, Nhân). Các quyền hạn đối với các đối tượng này được trình bày chi tiết tại Chương 5.

Để thực hiện các giải pháp trên, ĐATN sử dụng mô hình client-server với các công nghệ như sau: (i) phần server sử dụng framework .NET, (ii) phía client sử dụng framework Vue.js, (iii) cơ sở dữ liệu quan hệ được xây dựng bằng MariaDB và (iv) triển khai container hóa ứng dụng trên môi trường Docker và Kubernetes.

Các chức năng được xây dựng gồm có: (i) thêm sửa xoá bài tập, (ii) thêm sửa xoá chủ đề, (iii) thêm sửa xoá bài tập, (iv) đăng tải bài giải cho bài tập, (v) chia sẻ bài tập, (vi) chia sẻ vở, (vii), bình luận bài tập, (viii) bình luận bài giải và (ix) tương tác với bài tập.

1.4 Bố cục đồ án

Phần còn lại của đồ án được bố cục như sau:

Chương 2 trình bày về khảo sát bài toán thực tế và phân tích yêu cầu đối với ứng dụng web. Trong chương này, em khảo sát hiện trạng làm bài tập và chia sẻ bài tập của học sinh. Từ đó đưa ra những mô tả sơ lược về các chức năng cần phải có. Tiếp theo, em mô tả các use case đáp ứng các chức năng được nêu ra và khai quát hóa lại trong các quy trình nghiệp vụ. Sau đó, em đưa ra đặc tả cho các use case quan trọng và các đặc tả phi chức năng.

Trong Chương 3, em giới thiệu ngắn gọn về các công nghệ được sử dụng trong hệ thống. Các công nghệ này gồm có .NET, Vue.js, MariaDB, Docker và Kubernetes. Kubernetes là một công nghệ đáng chú ý, ĐATN trình bày thành một phần riêng trong Chương 5.

Chương 4 trình bày thực nghiệm quá trình xây dựng ứng dụng. Trong chương này, em trình bày các thiết kế, mô tả quá trình xây dựng ứng dụng, các kết quả đạt được, kiểm thử và triển khai. Các thiết kế gồm có thiết kế kiến trúc, thiết kế tổng quan hệ thống, thiết kế gói, thiết kế giao diện, thiết kế lớp và thiết kế cơ sở dữ liệu.

Chương 5 trình bày các giải pháp và đóng góp nổi bật của em trong ĐATN này nhằm đạt được các mục tiêu được đưa ra trong mục 1.2. Các giải pháp gồm: (1)

xây dựng chức năng lưu trữ bài tập, (2) xây dựng hệ thống phân loại bài tập với hai cấp độ là "Vở" và "Chủ đề", đánh nhãn cho bài tập, (3) xây dựng tính năng chia sẻ trên cấp độ vở và chia sẻ bài tập cùng với các tương tác đối với bài tập, (4) xây dựng tính năng tương tác giữa người dùng với bài tập và (5) vận dụng Kubernetes để triển khai hệ thống.

CHƯƠNG 2. KHẢO SÁT VÀ PHÂN TÍCH YÊU CẦU

2.1 Khảo sát hiện trạng

Cùng với việc áp dụng thi trắc nghiệm trong kỳ thi Trung học phổ thông quốc gia, các bài tập hàng ngày cũng được chuyển gần như toàn bộ sang hình thức trắc nghiệm. Hình thức làm bài tập này đem đến cho học sinh số lượng lớn bài tập với nhiều dạng bài hay, kiến thức rộng. Tuy nhiên, thời gian làm mỗi bài tập bị rút ngắn đi đáng kể khi chỉ có từ 1 - 2 phút cho mỗi bài tập. Vì thời gian có hạn nên học sinh chỉ đủ thời gian ghi hết sức vắn tắt lời giải ra vở. Nhưng đề bài lại có số lượng lớn các bài tập và thường được in ra giấy riêng nên việc ghép đề bài với bài giải là cực kì khó khăn. Điều này dẫn tới hệ quả là học sinh gần như không hệ thống hóa lại kiến thức, các dạng bài đã làm và giảm hiệu quả của quá trình ôn tập về sau.

Bên cạnh đó, nhu cầu chia sẻ bài tập giữa giáo viên và học sinh, giữa học sinh với nhau cũng khó đáp ứng khi chỉ có giấy vở vật lý.

Tuy nhiên, theo khảo sát của ĐATN, hiện nay chưa có phần mềm nào giải quyết được hai nhu cầu này.

Để giải quyết được hai nhu cầu nêu trên, ĐATN đưa ra mô tả sơ lược các tính năng như sau:

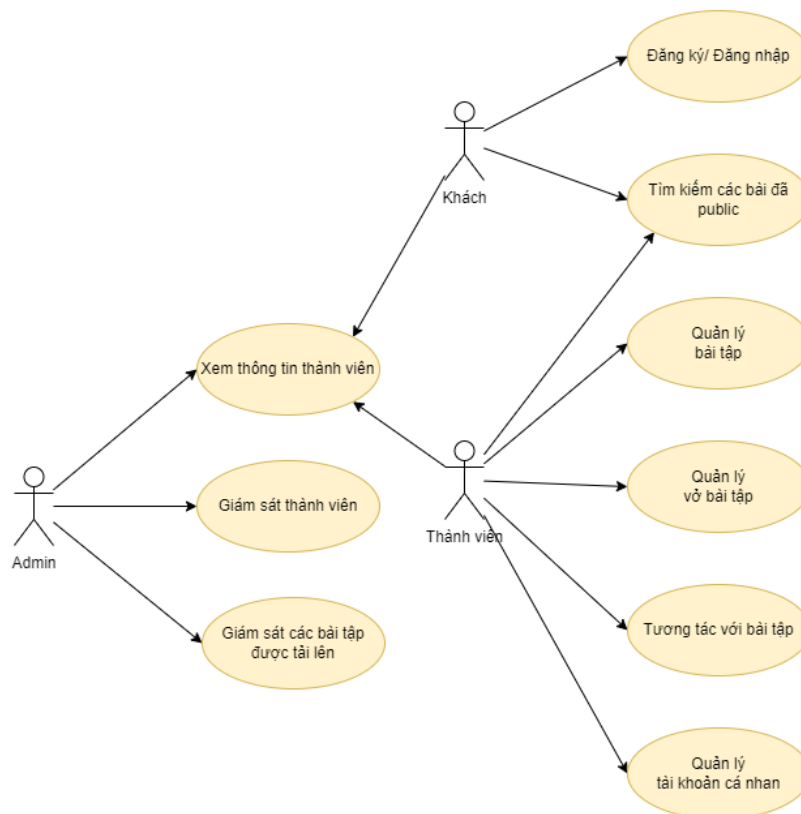
Lưu trữ bài tập: Tính năng này cho phép người dùng đăng tải bài tập của mình lên hệ thống với các thông tin đầu vào gồm văn bản, hình ảnh trực quan. Đồng thời, các bài giải cũng được đăng tải lên một cách dễ dàng.

Quản lý bài tập: Tính năng này hỗ trợ người dùng phân loại, sắp xếp bài tập của mình với nhiều mức phân loại.

Chia sẻ: Tính năng này đem lại cho người dùng khả năng chia sẻ các bài tập, nhóm bài tập đến một hay nhiều người dùng khác. Những người dùng được chia sẻ hoàn toàn có thể đóng góp bài giải của mình cho bài tập.

2.2 Tổng quan chức năng

2.2.1 Biểu đồ use case tổng quát

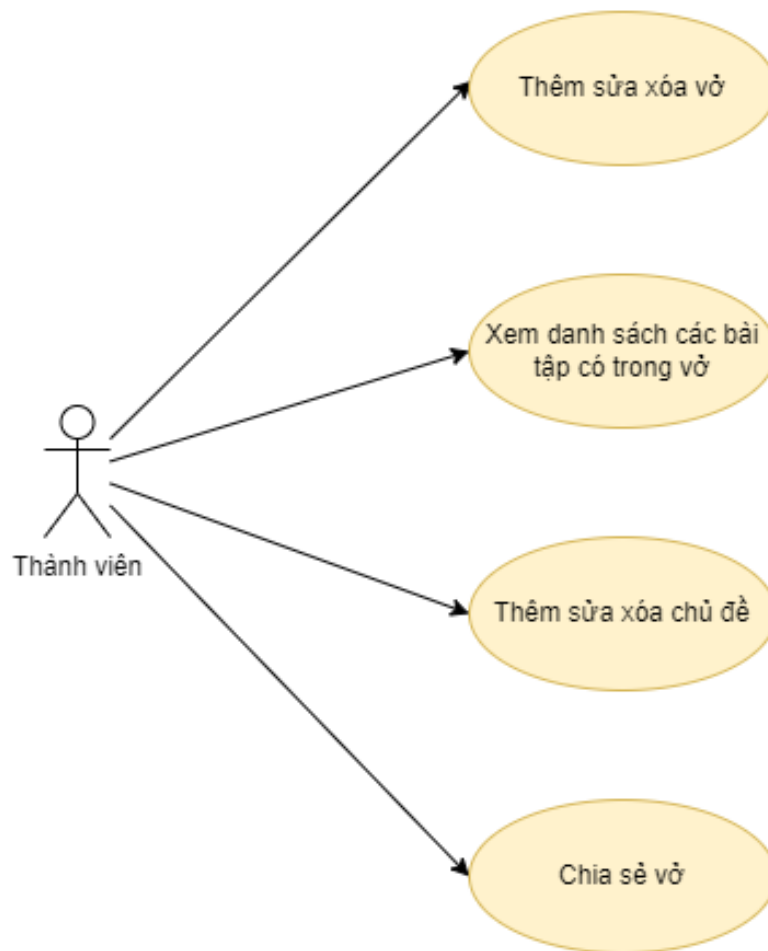


Hình 2.1: Biểu đồ use case tổng quan

Hình 2.1 mô tả biểu đồ use case tổng quan của ứng dụng web. Ứng dụng có 3 tác nhân chính là Khách, Thành viên và Quản trị viên. Khách là người dùng chưa đăng nhập vào ứng dụng. Thành viên là người dùng đã đăng nhập. Quản trị viên có vai trò giám sát và quản trị trong ứng dụng.

Khách có thể thực hiện các chức năng (i) Đăng ký/ Đăng nhập, (ii) Tìm kiếm các bài tập public. Sau khi đăng ký/ đăng nhập ứng dụng thì người dùng trở thành Thành viên. Thành viên có thể thực hiện các chức năng gồm có (i) Tìm kiếm các bài tập public, (ii) Quản lý bài tập, (iii) Quản lý vở bài tập, (iv) Tương tác với bài tập, (v) Tương tác với bài tập, (vi) Quản lý tài khoản cá nhân và (vii) Xem thông tin thành viên khác trong ứng dụng. Quản trị viên có thể thực hiện các chức năng (i) Xem thông tin thành viên trong ứng dụng, (ii) Giám sát thành viên và (iii) Giám sát các bài tập được tải lên.

2.2.2 Biểu đồ use case phân rã Quản lý vở

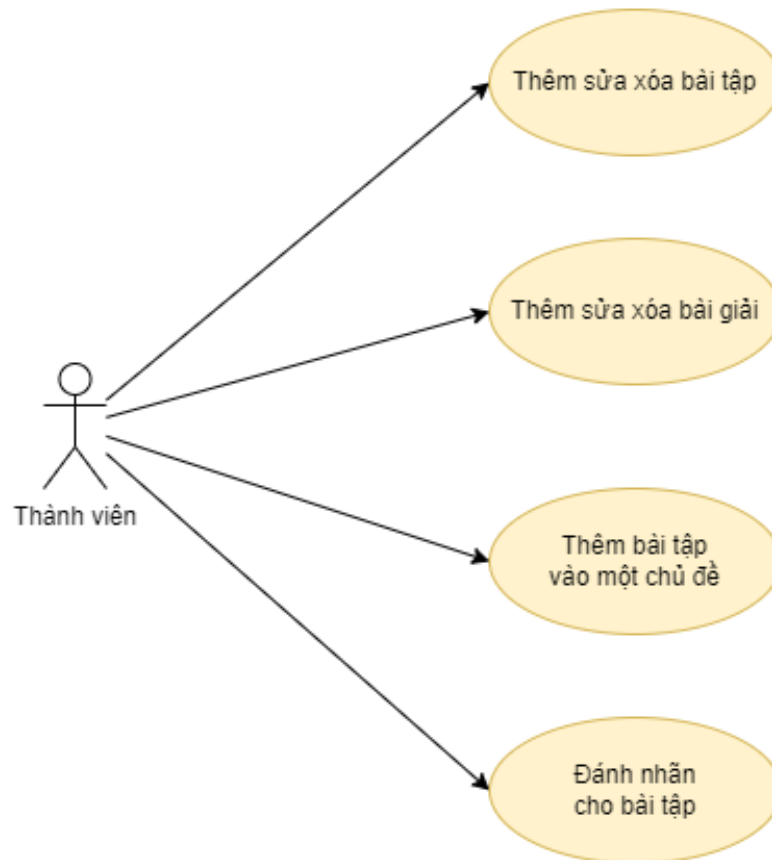


Hình 2.2: Biểu đồ use case phân rã Quản lý vở

Hình 2.2 ĐATN xây dựng lớp đối tượng Vở dựa trên vở viết thông thường. Vở bao gồm các bài tập mà thành viên sẽ đăng tải lên. Tên của Vở phản ánh tên của môn học.

Thành viên có thể thực hiện (1) thêm sửa xóa vở, (2) xem danh sách các bài tập có trong vở, (3) thêm sửa xóa các chủ đề trong vở và (4) chia sẻ vở cho thành viên khác.

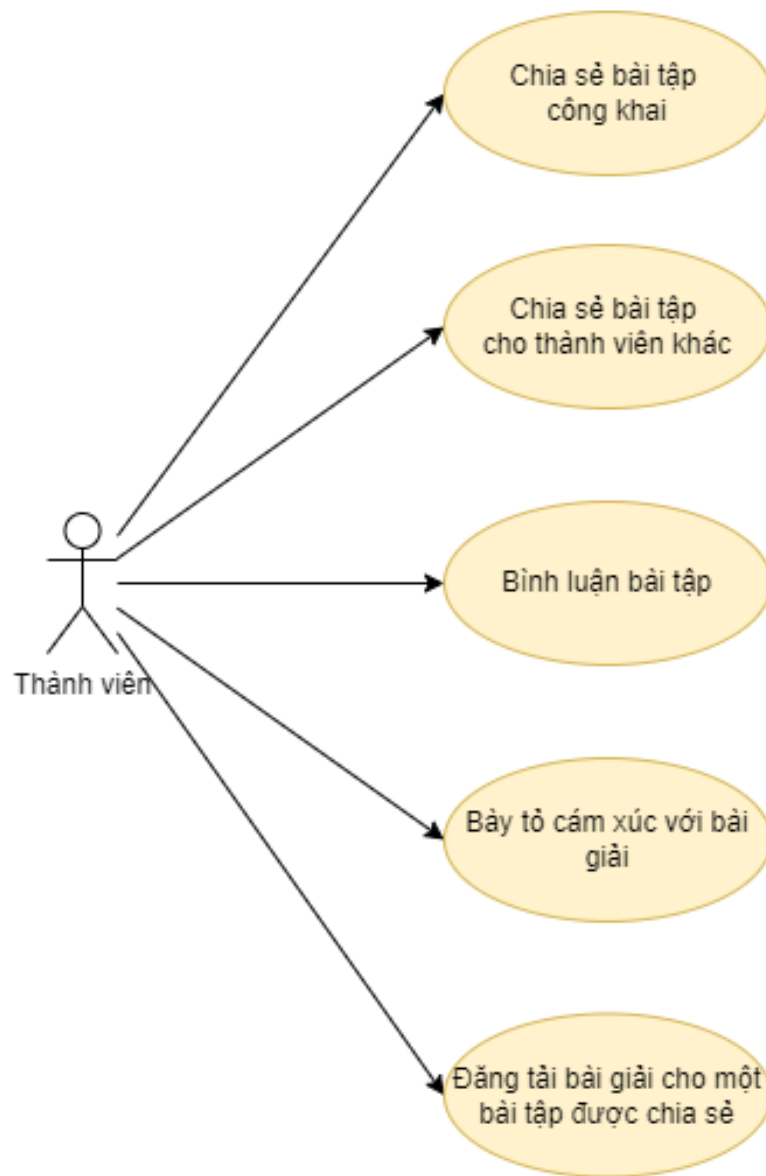
2.2.3 Biểu đồ use case phân rã Quản lý bài tập



Hình 2.3: Biểu đồ use case phân rã Quản lý bài tập

Hình 2.3 là biểu đồ use case phân rã Quản lý bài tập. Use case này được thực hiện bởi thành viên trong ứng dụng. Đối với bài tập, thành viên có thể thực hiện các chức năng gồm có (i) Thêm sửa xóa bài tập, (ii) Thêm sửa xóa bài giải, (3) Thêm bài tập vào một chủ đề, (4) Đánh nhãn cho bài tập.

2.2.4 Biểu đồ use case phân rã Tương tác với bài tập

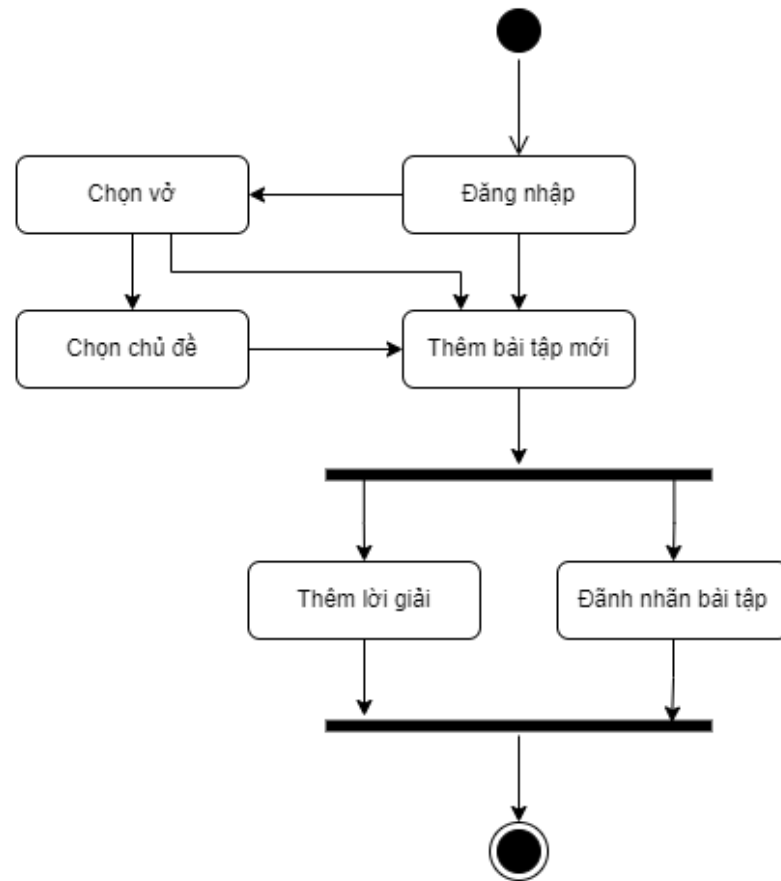


Hình 2.4: Biểu đồ use case phân rã Tương tác với bài tập

Hình 2.4 mô tả biểu đồ use case phân rã Tương tác với bài tập. Thành viên có thể thực hiện các chức năng gồm (1) chia sẻ bài tập công khai, (2) chia sẻ bài tập cho thành viên khác, (3) bình luận bài tập, (4) đăng tải bài giải cho một bài tập được chia sẻ và (5) bày tỏ cảm xúc với bài giải.

2.2.5 Quy trình nghiệp vụ

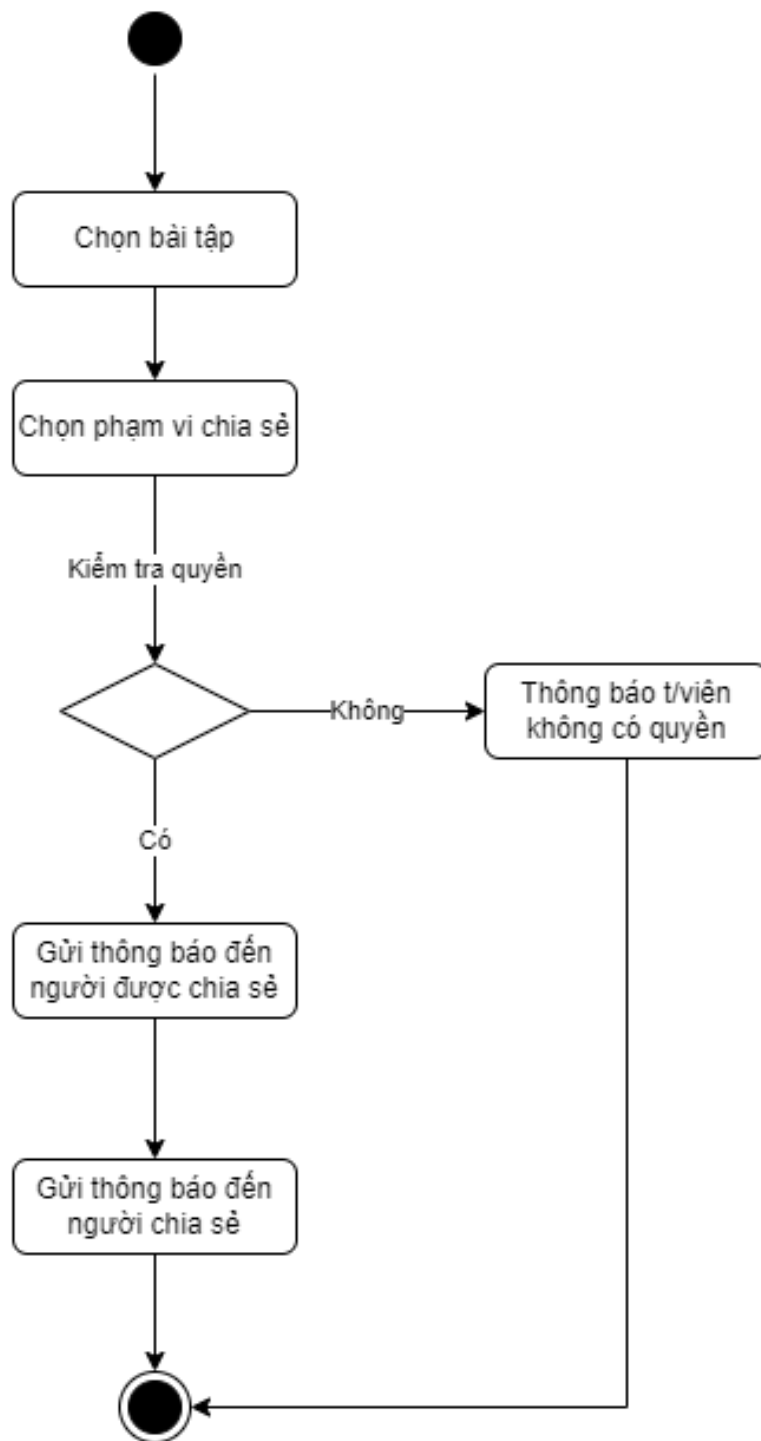
Quy trình nghiệp vụ Đăng tải bài tập



Hình 2.5: Quy trình nghiệp vụ Đăng tải bài tập

Hình 2.5 mô tả quy trình nghiệp vụ Đăng tải bài tập. Sau khi đăng nhập hệ thống, thành viên có thể tạo nhanh bài tập hoặc truy cập vào vở, vào chủ đề tạo. Sau khi bài tạo bài tập, thành viên có thể tiến hành thêm lời giải, đánh nhãn cho bài tập.

Quy trình nghiệp vụ Chia sẻ bài tập



Hình 2.6: Quy trình nghiệp vụ Chia sẻ bài tập

Hình 2.6 mô tả quy trình chia sẻ bài tập. Thành viên chọn bài tập cần chia sẻ, chọn phạm vi chia sẻ và xác nhận. Hệ thống kiểm tra lại quyền của thành viên đối với bài tập. Trường hợp thành viên không có quyền, hệ thống trả về thông báo lỗi cho người dùng. Ngược lại, hệ thống gửi thông báo đến thành viên được chia sẻ và thông báo chia sẻ thành công đến cho thành viên chia sẻ.

2.3 Đặc tả chức năng

2.3.1 Đặc tả use case Thêm vở

Mã use case	UC001	Tên use case	Thêm vở bài tập
Tác nhân	Thành viên		
Tiền điều kiện	Người dùng đã đăng nhập		
Luồng sự kiện chính	STT	Thực hiện bởi	Hành động
	1	Thành viên	Bấm nút thêm vở.
	2	Hệ thống	Hiển thị giao diện thêm vở mới.
	3	Thành viên	Điền thông tin tên vở và mô tả.
	4	Thành viên	Bấm nút lưu vở.
	5	Hệ thống	Kiểm tra dữ liệu đầu vào đã hợp lệ chưa.
	6	Hệ thống	Lưu trữ thông tin vở mới vào cơ sở dữ liệu.
	7	Hệ thống	Trả kết quả về cho thành viên qua thông báo trên màn hình.
	8	Thành viên	Nhận thông báo kết quả trên màn hình đồng thời hiển thị biểu tượng vở mới.
Luồng sự kiện thay thế	STT	Thực hiện bởi	Hành động
Hậu điều kiện	Có vở mới được hiển thị lên màn hình		

Bảng 2.1: Đặc tả use case Thêm vở

STT	Trường dữ liệu	Mô tả	Bắt buộc ?	Điều kiện hợp lệ	Ví dụ
1	title	Tiêu đề của vở	Có	Không chứa ký tự đặc biệt; Chứa ít nhất 3 ký tự.	Vở bài tập
2	description	Mô tả của vở	Không		Đây là phần mô tả

Bảng 2.2: Dữ liệu đầu vào của vở bài tập mới

2.3.2 Đặc tả use case Thêm bài tập mới

Mã use case	UC002	Tên use case	Thêm bài tập mới
Tác nhân	Thành viên		
Tiền điều kiện	Thành viên đã đăng nhập		
Luồng sự kiện chính	STT	Tác nhân	Hành động
	1	Thành viên	Bấm nút thêm bài tập.
	2	Hệ thống	Hiển thị giao diện thêm bài tập mới.
	3	Hệ thống	Truy vấn danh sách vở của thành viên; Hiển thị danh sách vở trong ComboBox.
	4	Thành viên	Chọn vở.
	5	Hệ thống	Truy vấn các chủ đề trong vở mà thành viên đã chọn; Hiển thị danh sách chủ đề trong ComboBox.
	6	Thành viên	Chọn chủ đề; Điền các thông tin của bài tập mới.
	7	Thành viên	Lưu bài tập.
	8	Hệ thống	Kiểm tra dữ liệu mới có hợp lệ không.
	9	Hệ thống	Lưu trữ bài tập vào cơ sở dữ liệu và trả kết quả về cho người dùng.
Luồng sự kiện thay thế	STT	Thực hiện bởi	Hành động
	4a	Hệ thống	Thông báo thành viên không có vở nào.
	6a	Hệ thống	Thông báo thành viên không có chủ đề nào.
Hậu điều kiện	Hiển thị danh sách bài tập, trong đó có đánh dấu bài tập mới tạo.		

Bảng 2.3: Đặc tả use case Thêm bài tập mới

STT	Trường dữ liệu	Mô tả	Bắt buộc ?	Điều kiện hợp lệ	Ví dụ
1	question	Nội dung câu hỏi	Không		Tính tỉ khối của nito so với oxy.
2	workbookId	ID của vở	Không		10f2594d-70b5-4d1b-ba7e-fb24a5608bf6
3	workbookTitle	Tên của vở	Không		Vở bài tập Hóa
4	topicId	ID của chủ đề	Không		aecc6da3-59f3-4f56-b01a-eef152927428
5	topicTitle	Tên của chủ đề	Không		Polyme
6	files	Danh sách file đính kèm	Không		

Bảng 2.4: Dữ liệu đầu vào của bài tập mới

2.3.3 Đặc tả use case Chia sẻ bài tập cho thành viên khác

Mã use case	UC003	Tên use case	Chia sẻ bài tập cho thành viên khác
Tác nhân	Thành viên		
Tiền điều kiện	Thành viên đã đăng nhập; Thành viên đã tạo bài tập trước đó hoặc có bài tập được chia sẻ		
Luồng sự kiện chính	STT	Thực hiện bởi	Hành động
	1	Thành viên	Bấm nút mở rộng trên bài tập
	2	Hệ thống	Xổ ra các tùy chọn thao tác với bài tập
	3	Thành viên	Bấm chọn chia sẻ
	4	Hệ thống	Hiển thị giao diện của tùy chọn chia sẻ
	5	Thành viên	Chọn chế độ chia sẻ với thành viên khác
	6	Hệ thống	Hiển thị input tìm kiếm thành viên
	7	Thành viên	Nhập username của thành viên cần chia sẻ đến
	8	Hệ thống	Tìm kiếm và hiển thị danh sách thành viên có username chứa các ký tự được nhập vào
	9	Thành viên	Chọn thành viên muốn chia sẻ
	10	Thành viên	Submit thông tin chia sẻ
	11	Hệ thống	Kiểm tra thông tin chia sẻ có hợp lệ không
	12	Hệ thống	Gửi thông báo đến thành viên được chia sẻ bài tập Lưu trữ thông báo này vào cơ sở dữ liệu
	13	Hệ thống	Thông báo lại kết quả cho thành viên thực hiện chia sẻ
	14	Thành viên	Nhận thông báo kết quả thực hiện hành động trên màn hình
Luồng sự kiện thay thế	STT	Thực hiện bởi	Hành động
	9a	Hệ thống	Thông báo không có thành viên nào có username chứa các ký tự đã nhập
Hậu điều kiện	Không có		

Bảng 2.5: Đặc tả use case Chia sẻ bài tập cho thành viên khác

2.3.4 Đặc tả use case Xem bài tập

Mã use case	UC004	Tên use case	Xem chi tiết bài tập
Tác nhân	Thành viên		
Tiền điều kiện	Thành viên đã đăng nhập		
Luồng sự kiện chính	STT	Thực hiện bởi	Hành động
	1	Thành viên	Click vào phần màn hình của nội dung bài tập
	2	Hệ thống	Kiểm tra bài tập có thỏa mãn một trong số điều kiện sau không: (i) thuộc sở hữu của thành viên (ii) được chia sẻ với thành viên không (iii) bài tập có ở trạng thái công khai
	3	Hệ thống	Nếu sai, kiểm tra xem vở chứa bài tập trên có được chia sẻ với thành viên không
	4	Hệ thống	Hiển thị thông tin bài tập ra màn hình
Luồng sự kiện thay thế	STT	Thực hiện bởi	Hành động
	4a	Hệ thống	Hiển thị thông báo lỗi thành viên không có quyền xem bài tập
Hậu điều kiện	Không có		

Bảng 2.6: Đặc tả use case Xem bài tập

CHƯƠNG 3. CÔNG NGHỆ SỬ DỤNG

3.1 .NET 6.0

.NET là một platform phát triển phần mềm miễn phí và mã nguồn mở. .NET được dùng để phát triển các phần mềm đa nền tảng, gồm Linux, Windows, MacOS và các hệ thống nhúng. .NET hỗ trợ bốn loại phần mềm gồm có:

- ASP.NET Core web app: Xây dựng các ứng dụng web.
- command-line/console app: Xây dựng các ứng dụng command-line/console.
- library: Xây dựng các thư viện lập trình.
- Universal Windows Platform app: Xây dựng các ứng dụng tương thích với các thiết bị cài đặt Microsoft Windows.

Các thư viện lập trình trong .NET được quản lý bởi NuGet package manager, cho phép nhà phát triển dễ dàng quản lý, cập nhật và bảo trì mã nguồn. .NET hỗ trợ container hóa phần mềm với Docker.

.NET 6.0 là phiên bản mới nhất của .NET, được ra mắt vào tháng 11 năm 2021. .NET 6.0 có nhiều cải tiến. Trong đó, một số cải tiến nổi bật có thể đề cập đến gồm có:

- Tối giản hóa API: Giúp xây dựng các HTTP API với ít các dependency nhất có thể.
- Nâng cao hiệu năng của ASP.NET Core.
- Cải tiến ngôn ngữ lập trình C lên phiên bản C 10...

ĐATN sử dụng ASP.NET 6.0 để xây dựng backend hệ thống.

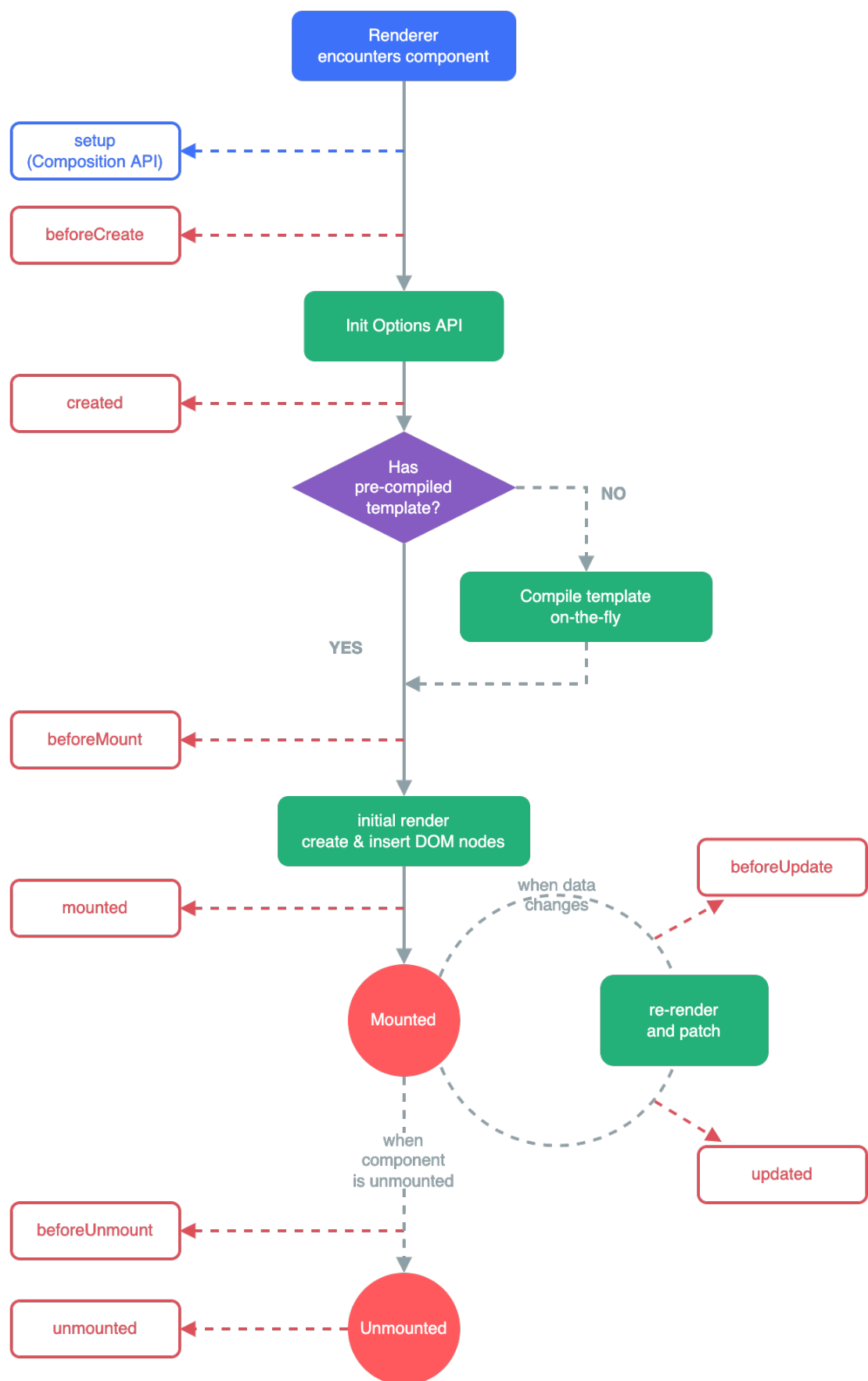
3.2 Vue.js

Vue.js (gọi tắt là Vue) là một JavaScript framework dùng để xây dựng giao diện người dùng. Vue được xây dựng trên cơ sở HTML, CSS và JavaScript. Vue có hai tính năng cốt lõi gồm có:

- Declarative Rendering: Vue mở rộng HTML bằng template syntax cho phép nhà phát triển khai báo các đoạn HTML và các đoạn HTML này được render bởi JavaScript mà HTML không bị trộn lẫn với JavaScript.
- Reactivity: Trạng thái của ứng dụng Vue là tập hợp các biến số và giá trị của chúng trong ứng dụng tại một thời điểm xác định. Vue tự động theo dõi trạng thái của ứng dụng để cập nhật lên DOM tương ứng trên giao diện.

Đối với tính năng Declarative Rendering đã đề cập bên trên, Vue cung cấp các Single-File Component, viết tắt là SFC, được biểu diễn bằng các file với đuôi mở rộng .vue. Theo đó, một SFC của Vue gồm 3 thành phần là HTML, JavaScript và CSS (hoặc có thể sử dụng SASS, LESS...). Phần HTML được khai báo bằng template syntax với các ràng buộc với phần JavaScript. Phần JavaScript định nghĩa là các trạng thái, các xử lý và các phương thức bám sát với vòng đời của component. Phần CSS có chỉ thị phạm vi cho phép các đoạn mã trong đó chỉ được áp dụng trong phạm vi của component. Các thành phần này được tách nhau tương đối rõ ràng, tiện lợi để phát triển và bảo trì mã nguồn.

Vue cụ thể hóa Reactivity bằng cách theo dõi vòng đời của một component và thực hiện các hành động được gắn với các sự kiện trong vòng đời của nó. Tính năng này được khai báo tại phần JavaScript của component. Sau đây là sơ đồ vòng đời của một component trong Vue SFC:



Hình 3.1: Vòng đời của một Vue Single-File Component

Vue là một Progressive Framework (đối lập với Monolithic Framework – framework đóng gói mọi thứ cần để xây dựng ứng dụng). Theo đó, trong quá trình xây dựng ứng dụng Vue, nhà phát triển có thể thêm dần dần các thành phần khác vào ứng dụng của mình. Một ứng dụng Vue mới được khởi tạo chỉ tập trung vào phần VIEW để hiển thị ra giao diện người dùng. Các thành phần khác có thể được thêm vào như Vue Router để thực hiện routing, Vuex để quản lý trạng thái ứng dụng. . .

Đồ án sử dụng Vue.js phiên bản Vue 3.0.0 để xây dựng giao diện người dùng.

3.3 MariaDB

MariaDB là một hệ quản trị cơ sở dữ liệu quan hệ mã nguồn mở, miễn phí, được phát triển từ hệ quản trị cơ sở dữ liệu quan hệ MySQL.

MariaDB kế thừa hầu hết các chức năng cơ bản cần thiết của MySQL. Do đó, MariaDB duy trì sự tương thích cao với MySQL. Bên cạnh đó, MariaDB phát triển thêm nhiều tính năng mới có sự nâng cấp hơn về cơ chế lưu trữ, tối ưu máy chủ.

DATN sử dụng MariaDB phiên bản 10.6.8 để xây dựng cơ sở dữ liệu quan hệ.

3.4 MinIO Object Storage System

MinIO là một Object Storage platform chuyên dùng để lưu trữ, truy xuất và tìm kiếm các *object*. Một *object* ở dạng binary, được hiểu là Binary Large Object (BLOB). Blob có thể là tệp ảnh, tệp âm thanh, tệp bảng tính hoặc có thể là đoạn mã thực thi ở dạng binary.

MinIO được sử dụng để lưu trữ tệp văn bản (word, PDF) hoặc hình ảnh được người dùng đăng tải lên hệ thống.

MinIO cung cấp các API tương tự như của Amazon S3 cloud storage service. MinIO tổ chức các objecy lại trong các bucket. Bucket tương tự như một thư mục (folder/directory) trong một filesystem. Bucket của MinIO cũng cung cấp các chức năng tương tự trên bucket Amazon S3.

MinIO là một phần mềm miễn phí. Nhà phát triển có thể dễ dàng cài đặt và tự quản lý mà không phụ thuộc vào bên thứ ba. Ngoài cách cài đặt trực tiếp lên máy chủ, MinIO cũng có Docker image đem đến khả năng triển khai trên môi trường container hoá.

3.5 Docker

Docker (hay Docker Engine) là một nền tảng cung cấp các phương pháp hiệu quả để xây dựng, triển khai và chạy các ứng dụng bằng cách sử dụng các container trên nền tảng ảo hóa.

Các containers cho phép lập trình viên đóng gói một ứng dụng với tất cả các

phần cần thiết, chẳng hạn như thư viện và các phụ thuộc khác, và gói tất cả ra dưới dạng một package. Bằng cách đó, nhờ vào container, ứng dụng sẽ chạy trên mọi máy Linux khác bất kể mọi cài đặt tùy chỉnh mà máy có thể có khác với máy được sử dụng để viết code.

Docker có nhiều ưu điểm khiến nó được ưa chuộng trong hoạt động phát triển và triển khai phần mềm:

- Tính dễ ứng dụng: Docker dễ tiếp cận với mọi người từ lập trình viên đến quản trị viên. Nó dễ dàng cài đặt và tận dụng lợi thế nhanh, gọn của container để build test nhanh chóng. Ứng dụng có thể được đóng gói từ một máy và chạy ở mọi nơi: trên máy cá nhân, private cloud hay public cloud.
- Ưu điểm về tốc độ: Docker container rất nhẹ và nhanh, người dùng có thể tạo và chạy docker container trong vài giây.
- Khả năng mở rộng linh hoạt: Người dùng có thể chia nhỏ ứng dụng của mình ra nhiều thành phần và đặt mỗi một thành phần đó vào trong container riêng. Ví dụ: Container này chạy app, container này chạy dịch vụ mysql và container khác nữa chạy Redis. Docker cung cấp cơ chế cho phép kết nối nhiều container với nhau để tạo thành ứng dụng và hỗ trợ scale, cập nhật các thành phần một cách độc lập với nhau.

ĐATN sử dụng Docker để đóng gói, triển khai backend và frontend của ứng dụng.

3.6 Kubernetes

Kubernetes là một mã nguồn mở được dùng để tự động triển khai hệ thống, scaling, quản lý các ứng dụng container. Kubernetes được phát triển bởi Google.

Kubernetes điều phối các container chạy trên nhiều server khác nhau (gọi là các worker node) với các cơ chế cân bằng tải. Người quản lý có thể dễ dàng thêm hoặc bớt một worker node vào cụm, cấu hình tự động tăng hoặc giảm số container theo nhu cầu sử dụng theo thời gian thực.

Với định hướng triển khai ứng dụng theo mô hình hiện đại, ĐATN vận dụng Kubernetes để triển khai backend và frontend.

CHƯƠNG 4. THỰC NGHIỆM VÀ ĐÁNH GIÁ

graphicx

4.1 Thiết kế kiến trúc

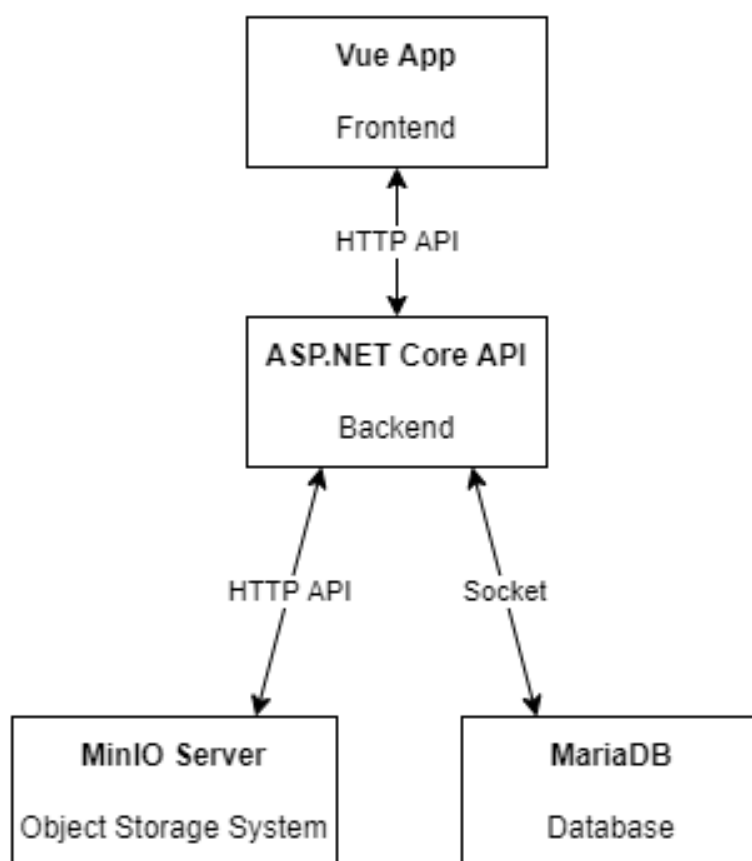
4.1.1 Lựa chọn kiến trúc phần mềm

DATN xây dựng phần mềm dựa trên kiến trúc client-server.

Client là một ứng dụng viết bằng framework Vue.js, đóng vai trò frontend. Server của phần mềm là một ứng dụng ASP.NET Core, đóng vai trò là backend. Client và server giao tiếp với nhau thông qua API.

Hệ thống sử dụng server MinIO để lưu trữ các file được người dùng tải lên.

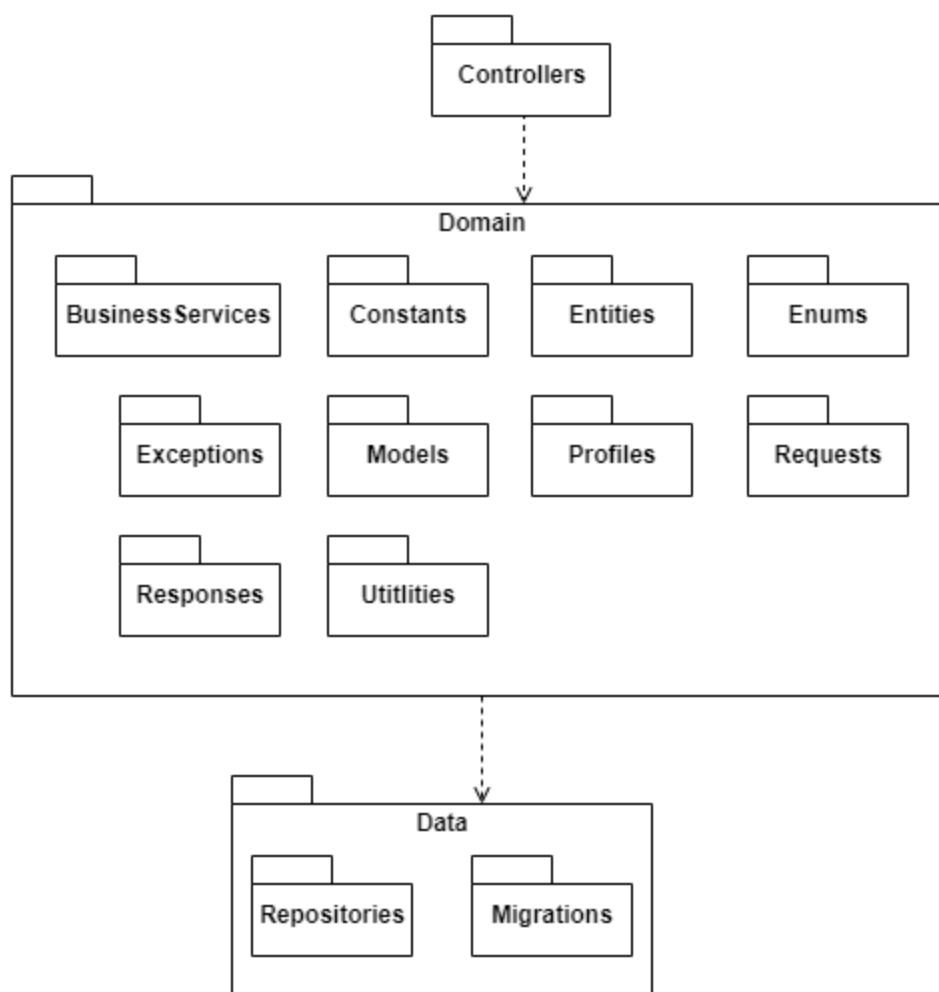
Kiến trúc phần mềm được mô tả như Hình 4.1 dưới đây:



Hình 4.1: Kiến trúc phần mềm

4.1.2 Thiết kế tổng quan

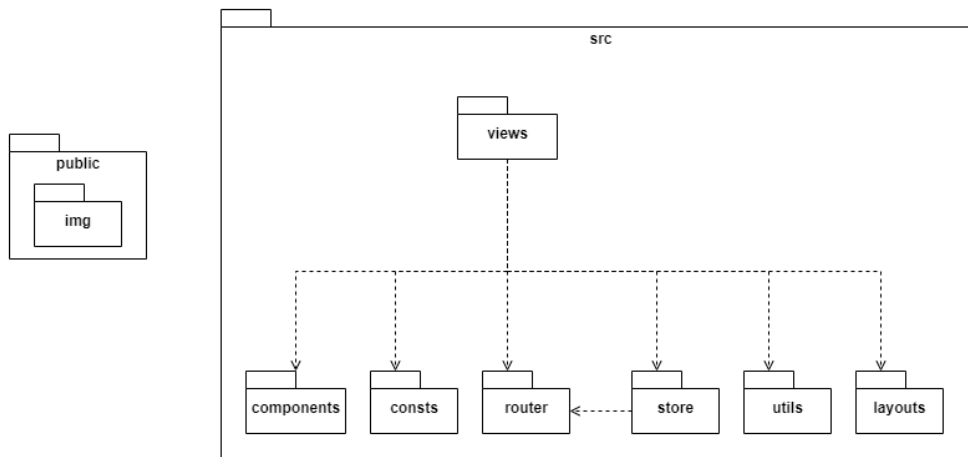
Thiết kế tổng quan của backend ứng dụng:



Hình 4.2: Thiết kế tổng quan backend

Hình 4.2 là biểu đồ gói mô tả thiết kế tổng quan của backend ứng dụng. Gói Controllers chứa các endpoint API của backend. Gói Domain chứa các gói con để xử lý nghiệp vụ của ứng dụng. Trong gói này có các gói nhỏ hơn gồm có (i) BusinessServices chứa logic nghiệp vụ, (ii) Constants chứa các hằng số, (iii) Entities chứa các thực thể ánh xạ đến bảng trong cơ sở dữ liệu quan hệ, (iv) Enums chứa các class lưu trữ các giá trị kiểu enum, (v) Exceptions chứa các exception, (vi) Models chứa các class định nghĩa dữ liệu được trả về cho người dùng, (vii) Profiles chứa các cấu hình chuyển đổi từ các class thực thể ở gói Entities sang các class trong gói Models, (viii) Requests chứa các class định nghĩa input người dùng nhập vào hệ thống, (ix) Response chứa các class định nghĩa output mà hệ thống trả về cho người dùng và (x) Utilities chứa các class dùng chung.

Thiết kế tổng quan của frontend ứng dụng:

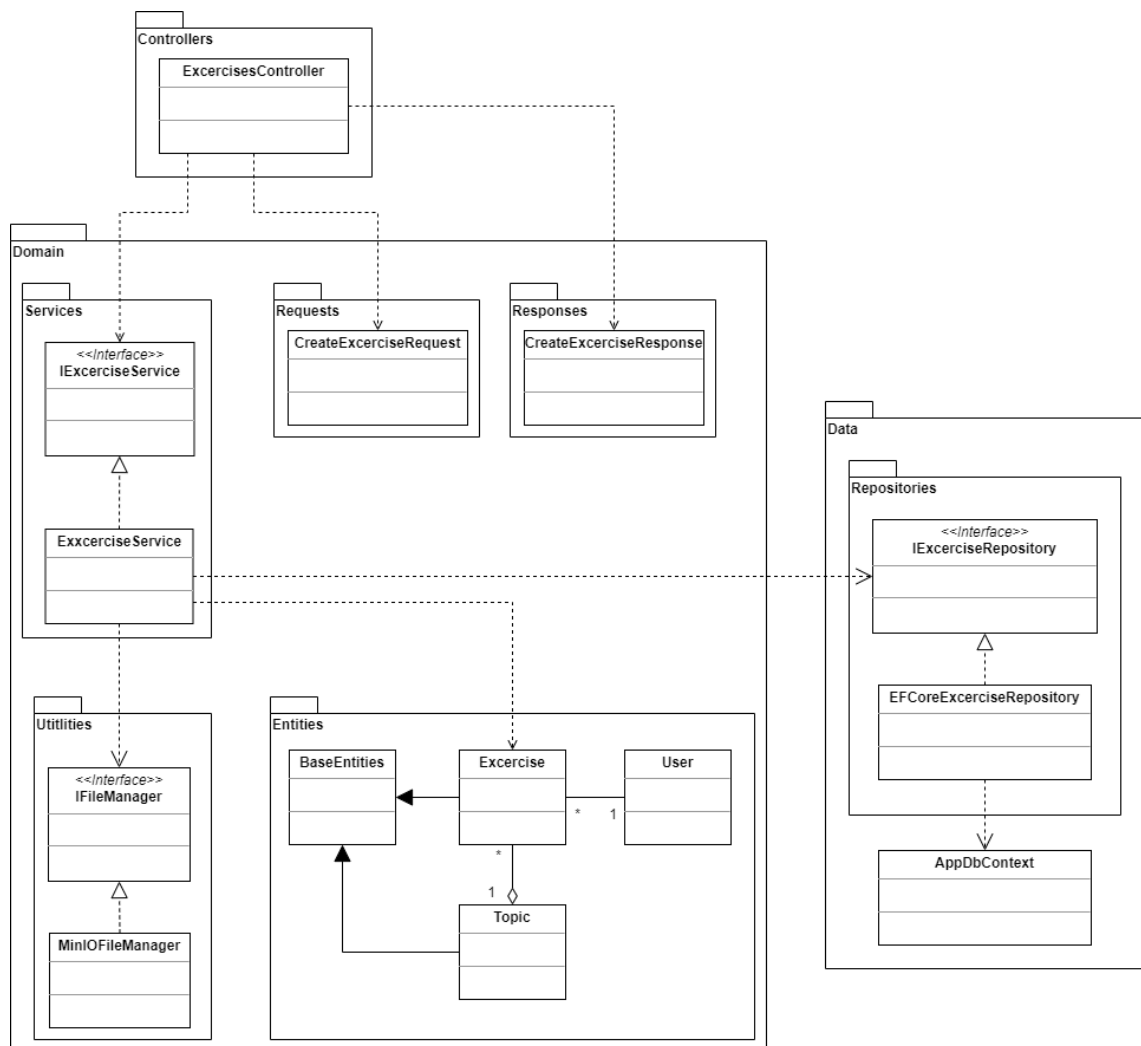


Hình 4.3: Thiết kế tổng quan frontend

Hình 4.3 là biểu đồ gói mô tả thiết kế tổng quan của frontend ứng dụng.. Gói public chứa các tài nguyên tĩnh của ứng dụng frontend như file index.html, các file ảnh... Gói src chứa mã nguồn hệ thống. Gói components chứa các Single-File Component dùng chung của ứng dụng. Gói consts chứa các hằng số và các đoạn văn bản cố định. Gói router chứa mã nguồn cấu hình routing URL. Gói store chứa các class kiểm soát trạng thái ứng dụng. Gói utils chứa các class, các hàm tiện ích dùng chung. Gói layouts chứa Single-File Component đóng vai trò xây dựng bố cục chung cho ứng dụng.

4.1.3 Thiết kế chi tiết gói

Dưới đây, em trình bày thiết kế chi tiết gói xử lý nghiệp vụ đăng tải bài tập phía backend ở Hình 4.4



Hình 4.4: Thiết kế chi tiết gói xử lý nghiệp vụ đăng tải bài tập

Vai trò của các lớp như sau:

- *ExercicesController* cung cấp API cho frontend để người dùng thực hiện nghiệp vụ. *ExercicesController* có quan hệ phụ thuộc vào interface *IExerciceService*.
- *CreateExerciseRequest* định nghĩa cấu trúc của input một bài tập mới.
- *CreateExerciseResponse* định nghĩa cấu trúc của phản hồi trả về.
- *IExerciceService* là interface định nghĩa hành vi lưu bài tập.
- *ExerciceService* implement interface *IExerciceService*, cài đặt cụ thể hành vi lưu bài tập. *ExerciceService* có quan hệ phụ thuộc vào interface *IExerciceRepository*, class entity *Exercise* và interface *IFileManager*.
- *IFileManager* là interface định nghĩa ra các hành vi xử lý upload, download file.
- *MinIOFileManager* implement interface *IFileManager*, cài đặt hành vi xử lý

upload, download file với MinIO Object Storage System.

- *BaseEntity* là class định nghĩa ra thuộc tính cơ bản của một entity.
- *Excercise* thừa kế *BaseEntity*, định nghĩa các thuộc tính của một entity *Excercise*, đại diện cho bài tập.
- *Topic* thừa kế *BaseEntity*, định nghĩa các thuộc tính của một chủ đề. *Topic* và *Excercise* có quan hệ một nhiều với *Excercise* hợp thành vào *Topic*.
- *User* định nghĩa các thuộc tính của một người dùng. Một *User* có thể có nhiều *Excercise*.
- *IExcerciseRepository* là interface định nghĩa hành vi truy vấn vào cơ sở dữ liệu.
- *EFCoreExcerciseRepository* implement interface *IExcerciseRepository*, cài đặt hành vi truy vấn vào cơ sở dữ liệu quan hệ theo cú pháp và cấu trúc của thư viện EF Core cho MySQL. *EFCoreExcerciseRepository* phụ thuộc vào *AppDbContext*.
- *AppDbContext* định nghĩa context để truy vấn cơ sở dữ liệu.

4.2 Thiết kế chi tiết

4.2.1 Thiết kế giao diện

Đặc tả thông tin về màn hình mà ứng dụng

Ứng dụng hướng tới hoạt động tốt trên trình duyệt web của hai loại thiết bị là desktop và mobile. Bảng mô tả thông tin màn hình ứng dụng hướng tới:

Tiêu chí	Desktop	Mobile	Chú thích
Độ phân dải màn hình	1366 × 768	1280 × 720	
Chiều rộng màn hình	1366px	320px - 480px	
Số lượng màu hỗ trợ	16777216	16777216	Đủ dải màu RGB

Bảng 4.1: Đặc tả màn hình mà ứng dụng hướng tới

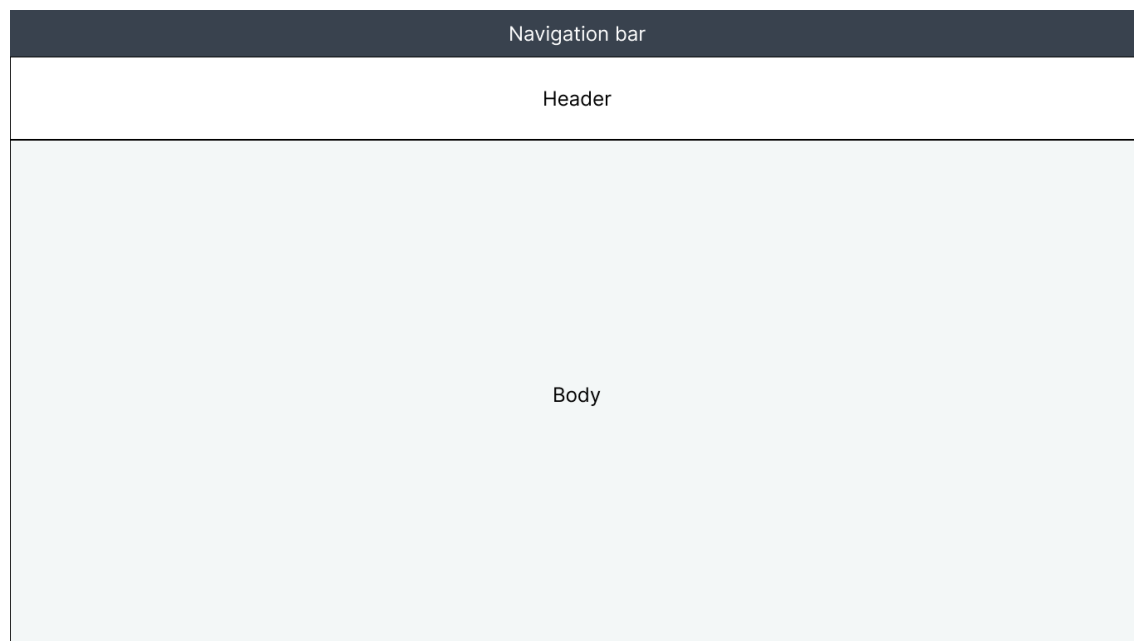
Chuẩn hóa thiết kế giao diện

Để đảm bảo thống nhất trong thiết kế giao diện trong phạm vi toàn bộ ứng dụng, ĐATN đã chuẩn hóa các yếu tố thiết kế giao diện gồm có: Bố cục, thiết kế nút, vị trí thông điệp phản hồi, màu sắc.

Thứ nhất, bố cục trang web được chuẩn hóa như sau:

Trên desktop, các màn hình chức năng tuân theo bố cục gồm ba phần theo thứ tự từ trên xuống dưới như sau: Navigation bar, Header và Body. Bố cục được minh

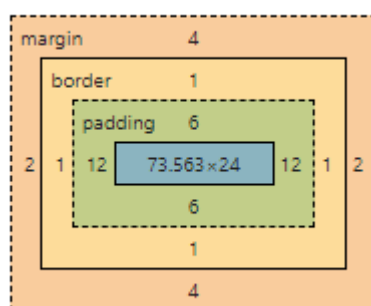
họa theo Hình 4.5 như sau:



Hình 4.5: Bố cục giao diện trên trình duyệt web của màn hình desktop

Phần **Navigation bar** là nơi chứa các nút điều khiển đến các màn hình chính của ứng dụng. Phần này có chiều cao 56px. Phần **Header** chứa thông tin mô tả chức năng cụ thể của màn hình hiện tại. Phần này có chiều cao 100px. Phần **Body** chứa nội dung chính của màn hình. Phần này có chiều cao tùy thuộc vào kích thước hiển thị của nội dung chứa bên trong.

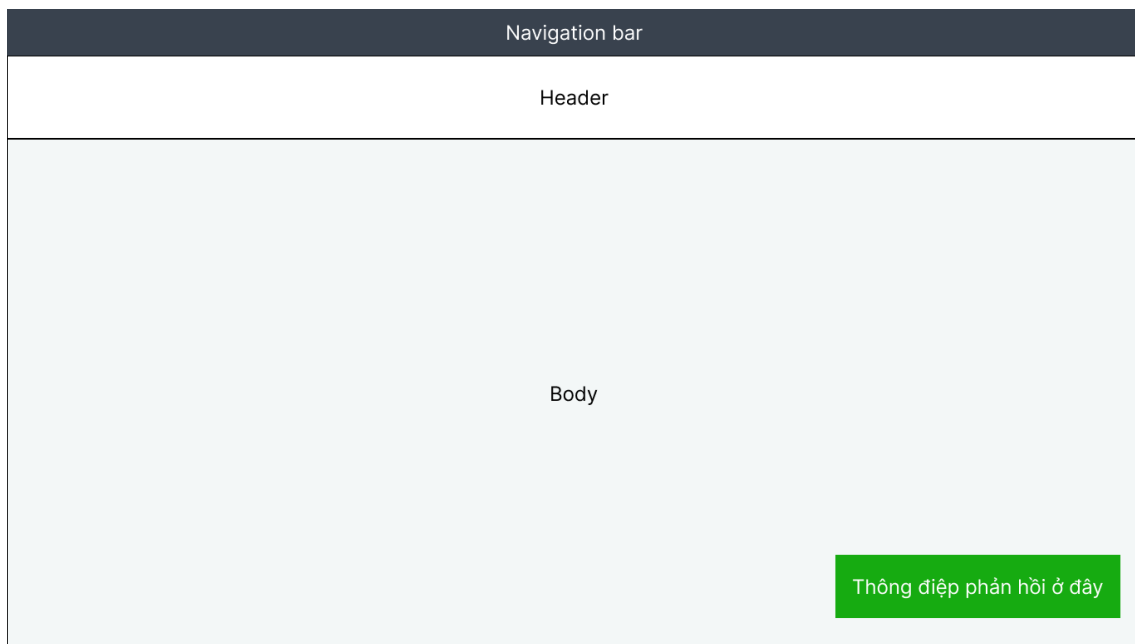
Thứ hai, các nút được thiết kế dựa trên nút của thư viện Bootstrap, được minh họa trong Hình 4.6:



Hình 4.6: Thiết kế nút của ứng dụng

Trong đó, chiều cao nút là 24px.

Thứ ba, các thông điệp phản hồi được bố trí xuất hiện ở góc dưới bên phải màn hình, thời gian xuất hiện là 3 giây.



Hình 4.7: Vị trí thông điệp phản hồi

Thư tự, màu sắc được sử dụng trong ứng dụng được quy ước như sau:

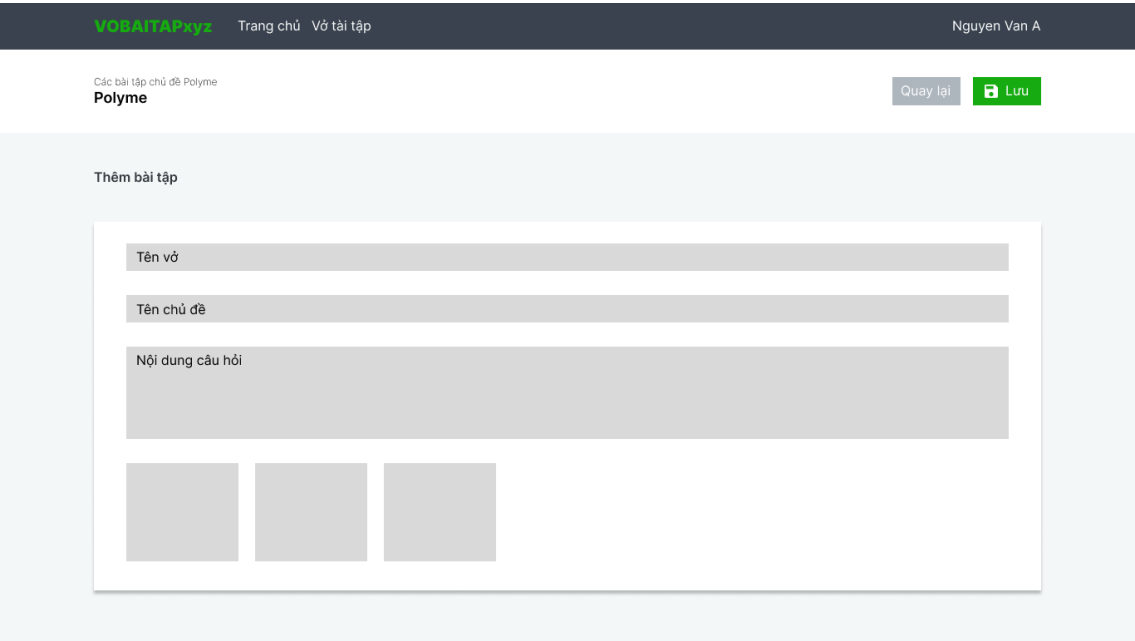
- Phần Navigation bar sử dụng màu nền đen với mã hexa là 39424E ;
- Phần Header sử dụng màu nền là màu trắng với mã hexa là FFFFFFF;
- Phần Body sử dụng màu nền xám nhé, mã hexa là F3F7F7;
- Các màu dùng cho thông điệp phản hồi gồm có màu xanh, màu đỏ và màu vàng. Các màu này được quy định theo quy ước màu của thư viện Bootstrap như ở Bảng . Màu xanh dùng cho các phản hồi thực hiện hành động thành công, màu đỏ được dùng cho các phản hồi hành động thực hiện xảy ra lỗi, màu vàng dùng cho các cảnh báo.
- Văn bản chỉ sử dụng hai màu là trắng và đen. Màu trắng được ưu tiên dùng cho phần giao diện có màu nền tối, màu đen được ưu tiên cho phần giao diện có màu nền sáng.

STT	Màu	Mã hexa
1	Xanh dương	#0D6EFD
2	Xám	#6C757D
3	Xanh lá	#198754
4	Đỏ	#DC3545
5	Vàng	#FFC107
6	Xanh nhạt	#0DACEF
7	Trắng	#F8F9FA
8	Đen	#212529

Bảng 4.2: Bảng màu theo quy ước của Bootstrap

Minh họa thiết kế giao diện

Hình 4.8 là minh họa thiết kế giao diện màn hình Thêm bài tập:



Hình 4.8: Minh họa thiết kế giao diện màn hình Thêm bài tập

Control	Operation	Function
Phần Navigation bar	Initial	Chứa các đường dẫn, nút chuyển hướng đến các phần khác nhau của ứng dụng
Phần Header	Initial	Chứa thông tin ngắn gọn về màn hình hiện tại (Thêm bài tập) và các nút thực hiện hành động
Nút "Quay lại"	Click	Quay về màn hình trước đó
Nút "Lưu"	Click	Lưu bài tập đang tạo
Form "Thêm bài tập"	Initial	Form dùng để nhập thông tin về bài tập

Bảng 4.3: Đặc tả màn hình Thêm bài tập

4.2.2 Thiết kế lớp

Dưới đây minh họa thiết kế của 4 lớp lần lượt là Excercise, ExcerciseService, EFCoreExcerciseRepository và EFCoreSharingRepository.

Excercise
+ Question: string + FileUrls: string + PublicMode: PublicModeEnum + AnswerCount: int + TopicId: Guid + TopicTitle: string + WorkbookId: Guid + WorkbookTitle: string + AuthorId: Guid + AuthorUserName: string
+ Excercise(string, string, PublicModeEnum, int, Guid, string, Guid, string, Guid, string): Excercise

Hình 4.9: Thiết kế lớp Excercise

Hình 4.9 mô tả thiết kế lớp Excercise. Đây là lớp thực thể, lưu các thuộc tính của lớp đại diện cho bài tập.

ExcerciseService
<ul style="list-style-type: none"> - _logger: ILogger - _mapper: IMapper - _fileManager: IFileManager - _workbookService: IWorkbookService - _excerciseRepository: IExcerciseRepository - _sharingRepository: ISharingRepository - _notificationRepository: INotificationRepository - _userRepository: IUserRepository - _notificationHubContext: IHubContext<NotificationHub> - _permissionService: IPermissionService
<ul style="list-style-type: none"> + ExcerciseService(ILogger<ExcerciseService> , IMapper, IFileManager, IWorkbookService, IExcerciseRepository, ISharingRepository, INotificationRepository, IUserRepository, IHubContext<NotificationHub>, IPermissionService): ExcerciseService + Index(Guid): List<ExcerciseViewModel> + CreateAsync(CreateExcerciseRequest, Guid, string): Task<Guid> + GetById(Guid, Guid): Task<ExcerciseViewModel> + GetStandaloneExcercises(Guid, Guid): List<ExcerciseViewModel> + GetExcercisesInTopic(Guid, Guid): List<ExcerciseViewModel> + Public(Guid, string): Task<Guid> + Public(Guid, string): Task<Guid> + GetPublicExcercises(IndexRequest): List<ExcerciseViewModel> + ShareExcerciseToUsers(Guid, string, Guid, List<string>, url): Task

Hình 4.10: Thiết kế lớp ExcerciseService

Hình 4.10 mô tả thiết kế lớp ExcerciseService. Lớp này có nhiệm vụ xử lý logic nghiệp vụ liên quan đến bài tập.

EFCoreExcerciseRepository
- _logger: ILogger - _appDbContext: AppDbContext
+ EFCoreExcerciseRepository(ILogger<EFCoreExcerciseRepository>, AppDbContext): EFCoreExcerciseRepository + GetAll(Guid): List<Excercise> + CreateAsync(Excercise): Task<Guid> + FindByIdAsync(Guid): Task<Excercise> + GetExcercisesInTopic(Guid, Guid): List<Excercise> + GetStandaloneExcercises(Guid, Guid): List<Excercise> + UpdateAsync(Excercise): Task + GetPublicExcercises(int, int): Task

Hình 4.11: Thiết kế lớp EFCoreExcerciseRepository

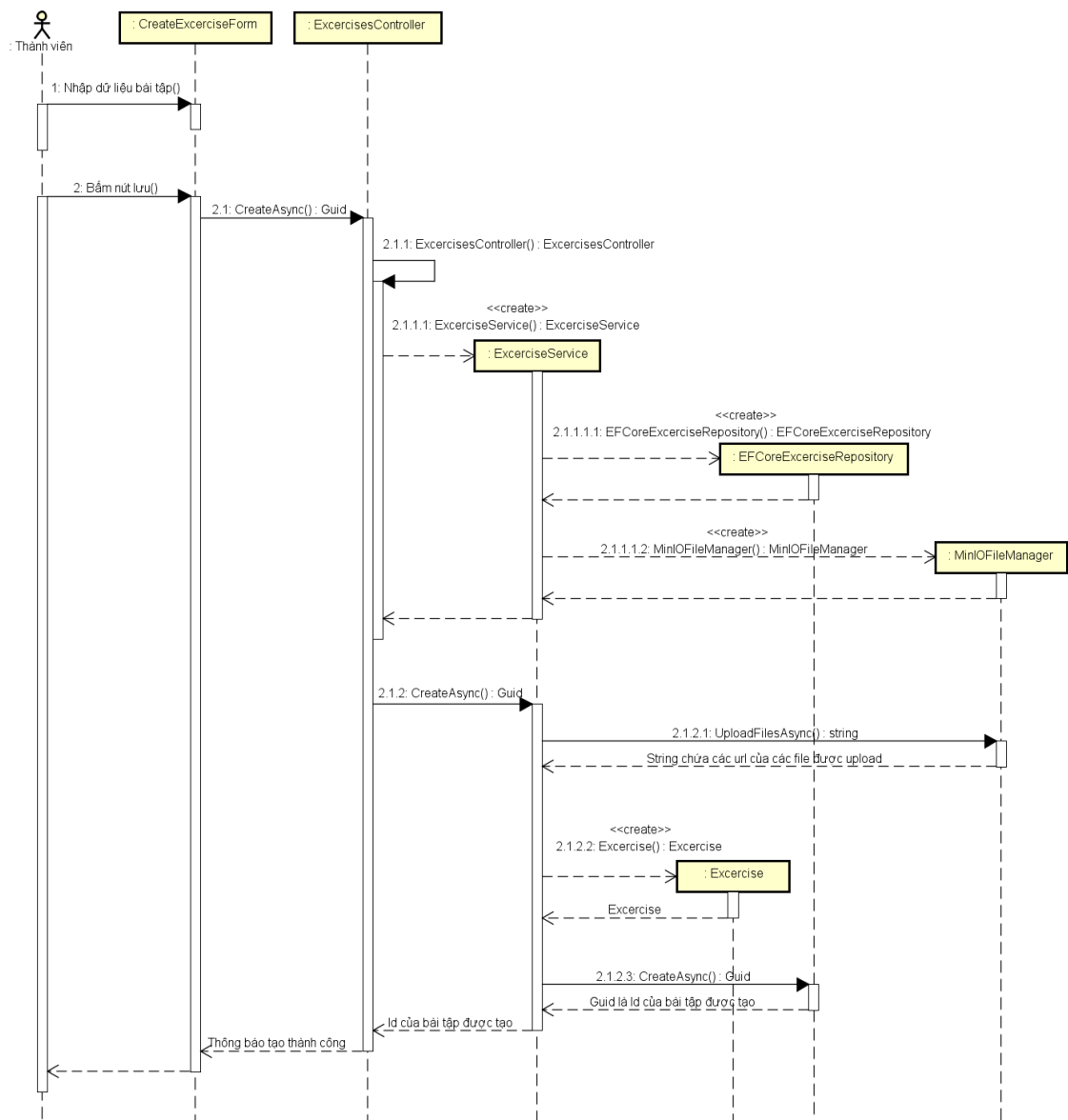
Hình 4.11 mô tả lớp EFCoreExcerciseRepository. Lớp này xử lý các truy vấn vào bảng Excercises trong cơ sở dữ liệu MariaDB, sử dụng thư viện trong EF Core Framework.

EFCoreSharingRepository
- _appDbContext: AppDbContext - _logger: ILogger
+ EFCoreSharingRepository(ILogger<EFCoreSharingRepository>, AppDbContext): EFCoreSharingRepository + CreateSharingAsync(Sharing): Task + Search(Guid, string, Guid): List<Sharing>

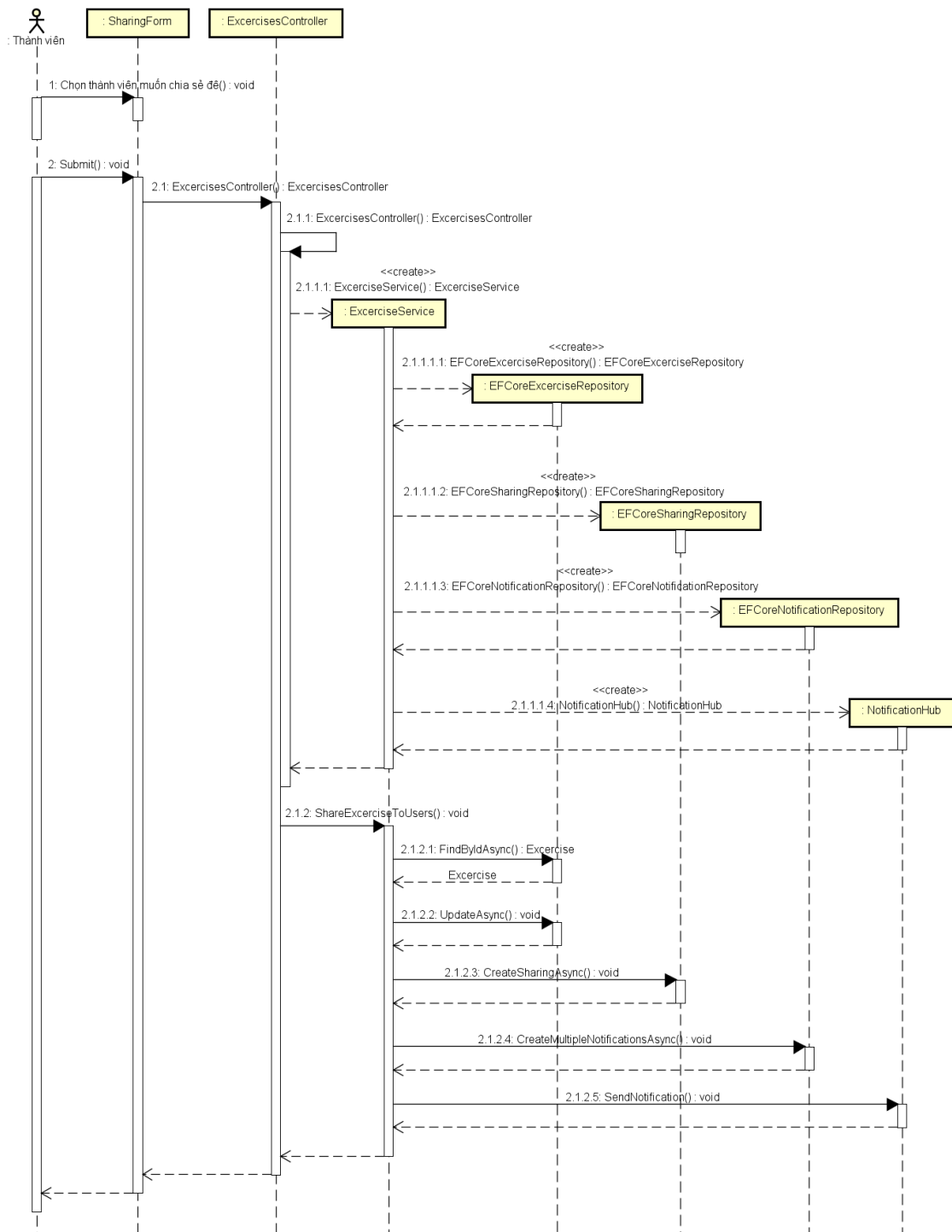
Hình 4.12: Thiết kế lớp EFCoreSharingRepository

Hình 4.12 mô tả lớp EFCoreSharingRepository. Lớp này xử lý các truy vấn vào các bảng lưu các thông tin chia sẻ, sử dụng thư viện trong EF Core Framework.

Sau đây, em mô tả mối quan hệ giữa các lớp trên qua các biểu đồ tuần tự.

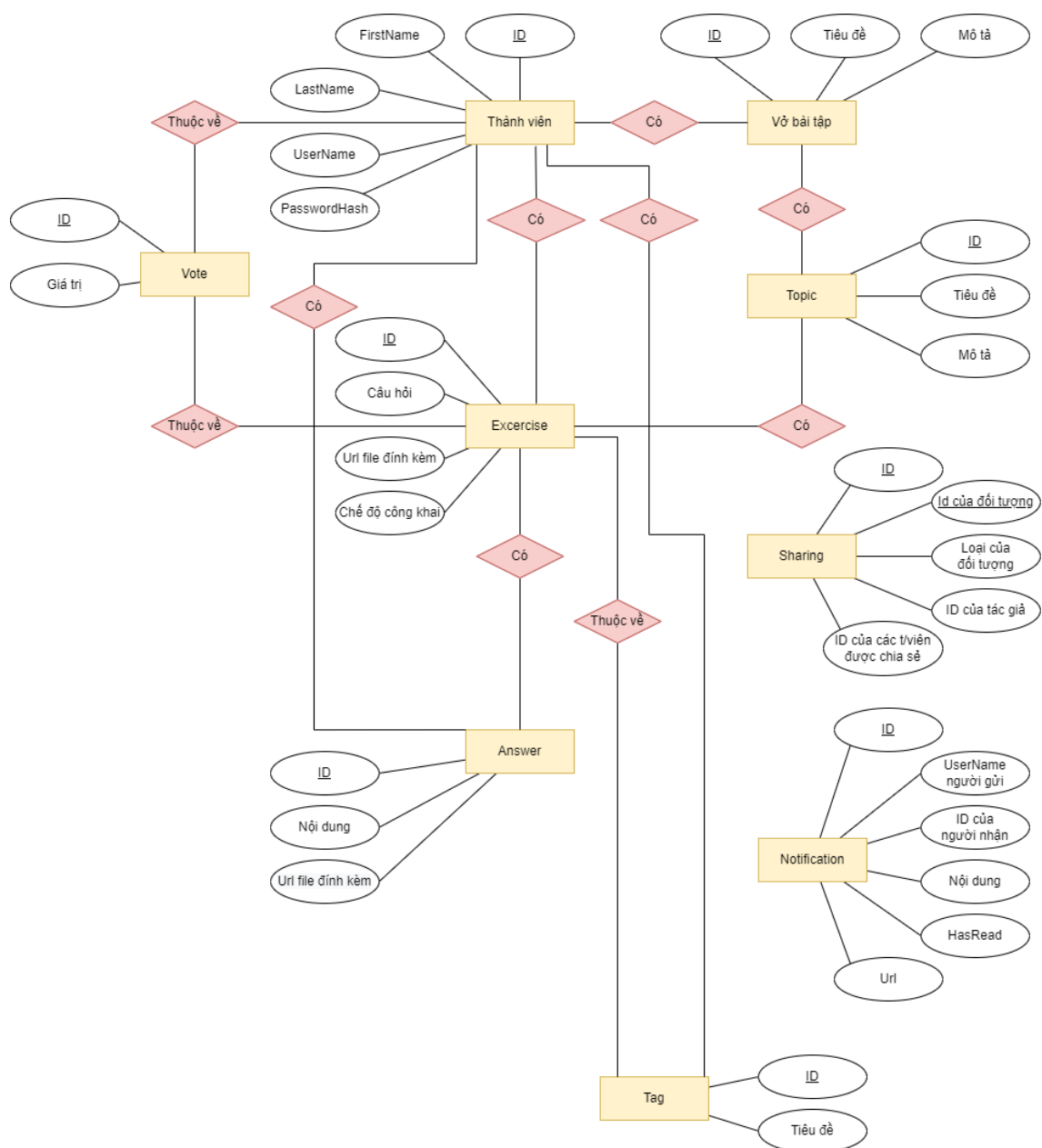


Hình 4.13: Biểu đồ tuần tự của use case Tạo bài tập

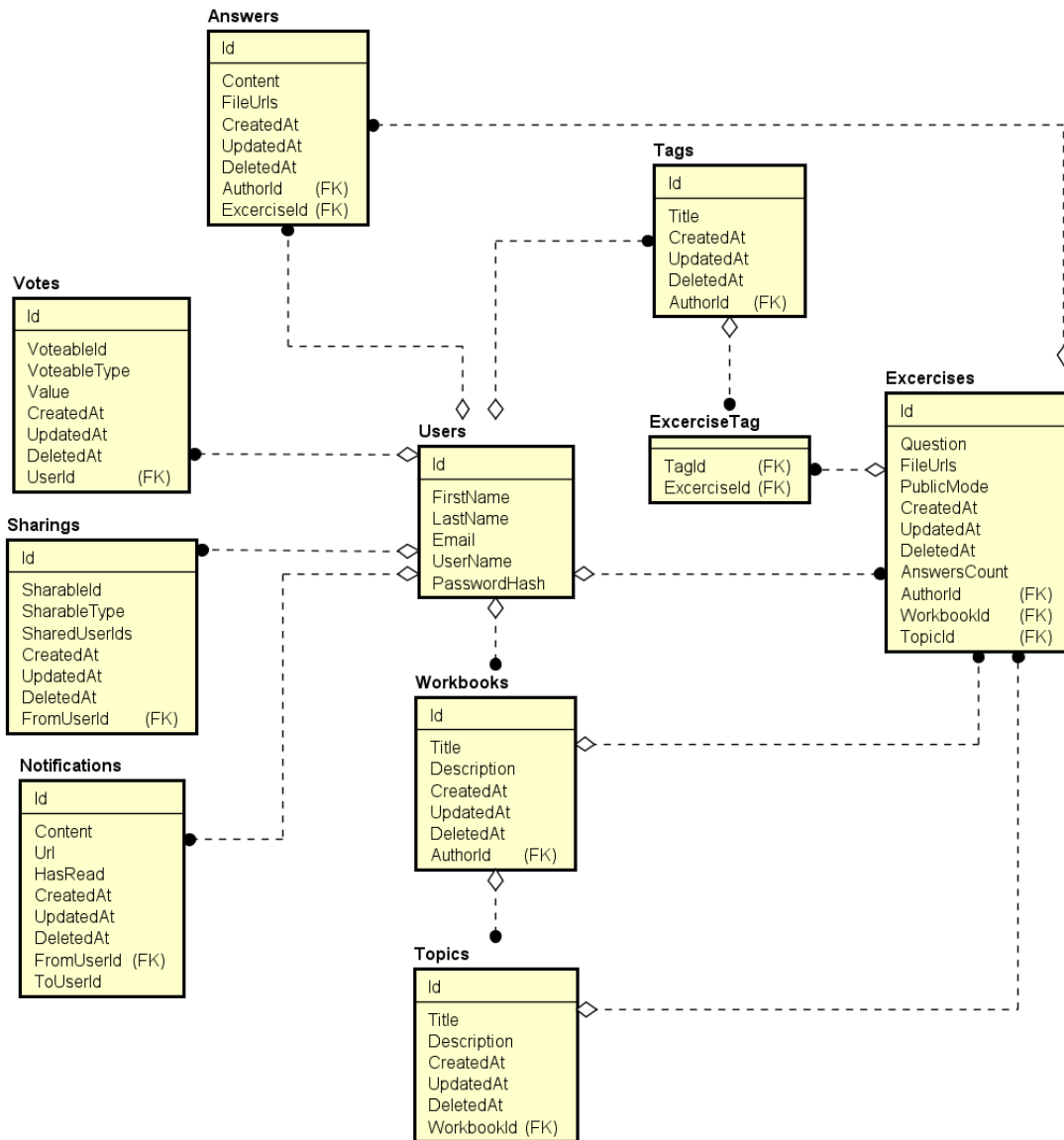


Hình 4.14: Biểu đồ tuần tự của use case Chia sẻ bài tập

4.2.3 Thiết kế cơ sở dữ liệu



Hình 4.15: Biểu đồ thực thể - liên kết

**Hình 4.16:** Thiết kế cơ sở dữ liệu

Bảng 4.4 mô tả thông tin bảng Users. Bảng này lưu dữ liệu của các thành viên.

STT	Tên trường	Kiểu dữ liệu	Key	Ý nghĩa
1	Id	CHAR(36)	Primary key	Id của bản ghi
2	FirstName	NVARCHAR(255)		Tên của thành viên
3	LastName	NVARCHAR(255)		Họ của thành viên
4	Email	VARCHAR(255)		Email của thành viên
5	UserName	VARCHAR(255)		Tên đăng nhập của thành viên
6	PasswordHash	VARCCHAR(1000)		Mã băm của mật khẩu

Bảng 4.4: Chi tiết bảng Users

Bảng 4.5 mô tả thông tin bảng Worksheets. Bảng này lưu dữ liệu của các vở bài tập.

Bảng 4.6 mô tả bảng Topics. Bảng này lưu thông tin các chủ đề.

STT	Tên trường	Kiểu dữ liệu	Key	Ý nghĩa
1	Id	CHAR(36)	Primary key	Id của bản ghi
2	Title	NVARCHAR(255)		Tiêu đề vở
3	Description	NVARCHAR(255)		Mô tả vở
4	CreatedAt	DATETIME		Thời điểm tạo
5	UpdatedAt	DATETIME		Thời điểm cập nhật lần nhất
6	DeletedAt	DATETIME		Thời điểm xóa
7	AuthorId	CHAR(36)	Foreign key	Id của thành viên là chủ sở hữu

Bảng 4.5: Chi tiết bảng Workbooks

STT	Tên trường	Kiểu dữ liệu	Key	Ý nghĩa
1	Id	CHAR(36)	Primary key	Id của bản ghi
2	Title	NVARCHAR(255)		Tiêu đề của chủ đề
3	Description	NVARCHAR(255)		Mô tả chủ đề
4	CreatedAt	DATETIME		Thời điểm tạo
5	UpdatedAt	DATETIME		Thời điểm cập nhật lần nhất
6	DeletedAt	DATETIME		Thời điểm xóa
7	WorkbookId	CHAR(36)	Foreign key	Id của vở bài tập

Bảng 4.6: Chi tiết bảng Topics

Bảng 4.7 mô tả bảng Excercises. Bảng này lưu thông tin các bài tập.

STT	Tên trường	Kiểu dữ liệu	Key	Ý nghĩa
1	Id	CHAR(36)	Primary key	Id của bản ghi
2	Question	NVARCHAR(1000)		Nội dung câu hỏi ở dạng văn bản
3	FileUrls	VARCHAR(1000)		Danh sách url của các file được ghép thành chuỗi
4	PublicMode	SMALLINT		Trạng thái công khai của bài tập
5	AnswersCount	INT		Số lượng câu trả lời hiện có
4	CreatedAt	DATETIME		Thời điểm tạo
5	UpdatedAt	DATETIME		Thời điểm cập nhật lần nhất
6	DeletedAt	DATETIME		Thời điểm xóa
7	AuthorId	CHAR(36)	Foreign key	Id của thành viên là tác giả bài tập
8	WorkbookId	CHAR(36)	Foreign key	Id của vở chứa bài tập này
9	TopicId	CHAR(36)	Foreign key	Id của chủ đề chứa bài tập này

Bảng 4.7: Chi tiết bảng Excercises

Bảng 4.8 mô tả thông tin bảng Answers. Bảng này lưu dữ liệu các bài giải.

STT	Tên trường	Kiểu dữ liệu	Key	Ý nghĩa
1	Id	CHAR(36)	Primary key	Id của bản ghi
2	Content	NVARCHAR(1000)		Nội dung lời giải ở dạng văn bản
3	FileUrls	VARCHAR(1000)		Danh sách url của các file đính kèm được ghép thành chuỗi
4	CreatedAt	DATETIME		Thời điểm tạo
5	UpdatedAt	DATETIME		Thời điểm cập nhật lần nhất
6	DeletedAt	DATETIME		Thời điểm xóa
7	AuthorId	CHAR(36)	Foreign key	Id của thành viên là tác giả của câu trả lời
8	ExcerciseId	CHAR(36)	Foreign key	Id của bài tập

Bảng 4.8: Chi tiết bảng Answers

Bảng 4.9 mô tả thông tin bảng Sharings. Bảng này lưu dữ liệu các lượt chia sẻ.

STT	Tên trường	Kiểu dữ liệu	Key	Ý nghĩa
1	Id	CHAR(36)	Primary key	Id của bản ghi
2	SharableId	CHAR(36)		Id của đối tượng được chia sẻ
3	SharableType	VARCHAR(255)		Tên Class của đối tượng được chia sẻ
4	SharedUserIds	VARCHAR(1000)		Danh sách Id của các thành viên được chia sẻ
5	CreatedAt	DATETIME		Thời điểm tạo
6	UpdatedAt	DATETIME		Thời điểm cập nhật lần nhất
7	DeletedAt	DATETIME		Thời điểm xóa
8	FromUserId	CHAR(36)	Foreign key	Id của thành viên thực hiện chia sẻ

Bảng 4.9: Chi tiết bảng Sharings

Bảng 4.10 mô tả thông tin bảng Answers. Bảng này lưu thông tin của các lời giải.

STT	Tên trường	Kiểu dữ liệu	Key	Ý nghĩa
1	Id	CHAR(36)	Primary key	Id của bản ghi
2	Content	NVARCHAR(1000)		Nội dung lời giải ở dạng văn bản
3	FileUrls	VARCHAR(1000)		Danh sách url của các file đính kèm được ghép thành chuỗi
4	CreatedAt	DATETIME		Thời điểm tạo
5	UpdatedAt	DATETIME		Thời điểm cập nhật lần nhất
6	DeletedAt	DATETIME		Thời điểm xóa
7	AuthorId	CHAR(36)	Foreign key	Id của thành viên là tác giả của câu trả lời
8	ExcerciseId	CHAR(36)	Foreign key	Id của bài tập

Bảng 4.10: Chi tiết bảng Answers

Bảng 4.11 mô tả thông tin bảng Notifications. Bảng này lưu thông tin của các thông báo.

STT	Tên trường	Kiểu dữ liệu	Key	Ý nghĩa
1	Id	CHAR(36)	Primary key	Id của bản ghi
2	Content	VARCHAR(1000)		Nội dung của thông báo
3	Url	VARCHAR(1000)		Đường dẫn trở đến nội dung được thông báo
4	HasRead	BIT		Người nhận đã đọc thông báo hay chưa
5	ToUserId	CHAR(36)		Id của thành viên sẽ nhận thông báo
5	CreatedAt	DATETIME		Thời điểm tạo
6	UpdatedAt	DATETIME		Thời điểm cập nhật lần nhất
7	DeletedAt	DATETIME		Thời điểm xóa
8	FromUserId	CHAR(36)	Foreign key	Id của thành viên thực hiện chia sẻ

Bảng 4.11: Chi tiết bảng Notifications

Bảng 4.12 mô tả thông tin bảng Votes. Bảng này lưu thông tin của các lượt bình chọn (vote).

STT	Tên trường	Kiểu dữ liệu	Key	Ý nghĩa
1	Id	CHAR(36)	Primary key	Id của bản ghi
2	VotableId	CHAR(36)		Id của đối tượng được vote
3	VotableType	VARCHAR(255)		Tên class của đối tượng được vote
4	Value	SMALLINT		Giá trị của lượt vote: -1, 0, 1
5	CreatedAt	DATETIME		Thời điểm tạo
6	UpdatedAt	DATETIME		Thời điểm cập nhật lần nhất
7	DeletedAt	DATETIME		Thời điểm xóa
8	UserId	CHAR(36)	Foreign key	Id của thành viên thực hiện vote

Bảng 4.12: Chi tiết bảng Votes

Bảng 4.13 mô tả thông tin bảng Tags. Bảng này lưu thông tin các thẻ của người dùng tạo ra dùng để gắn vào các bài tập nhằm mục đích phân loại.

STT	Tên trường	Kiểu dữ liệu	Key	Ý nghĩa
1	Id	CHAR(36)	Primary key	Id của bản ghi
2	Title	VARCHAR(255)		Id của đối tượng được vote
3	CreatedAt	DATETIME		Thời điểm tạo
4	UpdatedAt	DATETIME		Thời điểm cập nhật lần nhất
5	DeletedAt	DATETIME		Thời điểm xóa
6	AuthorId	CHAR(36)	Foreign key	Id của thành viên sở hữu thẻ

Bảng 4.13: Chi tiết bảng Tags

Bảng 4.14 mô tả thông tin bảng ExcerciseTag. Bảng này là bảng trung gian của hai bảng Excercises và bảng Tags.

STT	Tên trường	Kiểu dữ liệu	Key	Ý nghĩa
1	ExcerciseId	CHAR(36)	Primary key	Id của bản ghi trong bảng Excercises
2	TagId	CHAR(36)		Id của bản ghi trong bảng Tags

Bảng 4.14: Chi tiết bảng ExcerciseTag

4.3 Xây dựng ứng dụng

4.3.1 Thư viện và công cụ sử dụng

Để thực hiện ĐATN, em sử dụng các công cụ và IDE như sau:

Mục đích	Công cụ	Địa chỉ URL
IDE lập trình	Visual Studio 2022	https://visualstudio.microsoft.com/vs/
Code Editor	Visual Studio Code	https://code.visualstudio.com/
SSH Server	Termius	https://termius.com/windows
Làm việc với database	MySQL Workbench	https://www.mysql.com/products/workbench/
Vẽ biểu đồ UML, thiết kế...	Astah Professional Diagrams.net	https://astah.net/products/astah-professional/ https://app.diagrams.net/
Thiết kế giao diện	Figma	https://figma.com
Quản lý mã nguồn	GitSCM Gitlab	https://git-scm.com/ https://gitlab.com

Bảng 4.15: Danh sách các công cụ sử dụng

Mục đích	Thư viện	Địa chỉ URL
AutoMapper extension	AutoMapper.Extensions.Microsoft. .DependencyInjection	https://www.nuget.org/packages/AutoMapper.Extensions.Microsoft.DependencyInjection/11.0.0
Nhận và xử lý bearer token	Microsoft.AspNetCore.Authentication. .JwtBearer	https://www.nuget.org/packages/Microsoft.AspNetCore.Authentication.JwtBearer/6.0.5
CORS middleware và policy	Microsoft.AspNetCore.Cors	https://www.nuget.org/packages/Microsoft.AspNetCore.Cors/2.2.0
Xác thực người dùng	Microsoft.AspNetCore.Identity	https://www.nuget.org/packages/Microsoft.AspNetCore.Identity/2.2.0
Quản lý database liên quan đến xác thực người dùng	Microsoft.AspNetCore.Identity. .EntityFrameworkCore	https://www.nuget.org/packages/Microsoft.AspNetCore.Identity.EntityFrameworkCore/6.0.5
Giao tiếp real-time trong ứng dụng	Microsoft.AspNetCore.SignalR	https://www.nuget.org/packages/Microsoft.AspNetCore.SignalR/1.1.0
Công cụ quản lý migration của Entity Framework Core	Microsoft.EntityFrameworkCore.Tools	https://www.nuget.org/packages/Microsoft.EntityFrameworkCore.Tools/6.0.5
Minio SDK dành cho .NET	Minio	https://www.nuget.org/packages/Minio/4.0.4
MySQL database provider	Pomelo.EntityFrameworkCore.MySql	https://www.nuget.org/packages/Pomelo.EntityFrameworkCore.MySql/6.0.1
Swagger tool để tài liệu hóa và test API	Swashbuckle.AspNetCore	https://www.nuget.org/packages/Swashbuckle.AspNetCore/6.3.1

Bảng 4.16: Danh sách các thư viện sử dụng cho backend

Mục đích	Thư viện	Địa chỉ URL
HTTP client	axios	https://www.npmjs.com/package/axios
Thư viện CSS	bootstrap	https://www.npmjs.com/package/bootstrap
Routing cho ứng dụng Vue.js	vue-router	https://www.npmjs.com/packages/vue-router
Spinner	vue-spinner	https://www.npmjs.com/packages/vue-spinner
Hiển thị thông báo	vue-toast-notification	https://www.npmjs.com/package/vue-toast-notification
Quản lý trạng thái ứng dụng Vue.js	vuex	https://www.npmjs.com/package/vuex
Editor input	@ckeditor/ckeditor5-vue	https://www.npmjs.com/package/ckeditor5
SignalR client	@microsoft/signalr	https://www.npmjs.com/package/@microsoft/signalr

Bảng 4.17: Danh sách các thư viện sử dụng cho frontend

4.3.2 Kết quả đạt được

4.3.3 Minh họa các chức năng chính

4.4 Kiểm thử

4.5 Triển khai

CHƯƠNG 5. CÁC GIẢI PHÁP VÀ ĐÓNG GÓP NỔI BẬT

Chương này có độ dài tối thiểu 5 trang, tối đa không giới hạn.¹ Sinh viên cần trình bày tất cả những nội dung đóng góp mà mình thấy tâm đắc nhất trong suốt quá trình làm ĐATN. Đó có thể là một loạt các vấn đề khó khăn mà sinh viên đã từng bước giải quyết được, là giải thuật cho một bài toán cụ thể, là giải pháp tổng quát cho một lớp bài toán, hoặc là mô hình/kiến trúc hữu hiệu nào đó được sinh viên thiết kế.

Chương này **là cơ sở quan trọng** để các thầy cô đánh giá sinh viên. Vì vậy, sinh viên cần phát huy tính sáng tạo, khả năng phân tích, phản biện, lập luận, tổng quát hóa vấn đề và tập trung viết cho thật tốt. Mỗi giải pháp hoặc đóng góp của sinh viên cần được trình bày trong một mục độc lập bao gồm ba mục con: (i) dẫn dắt/giới thiệu về bài toán/vấn đề, (ii) giải pháp, và (iii) kết quả đạt được (nếu có).

Sinh viên lưu ý **không trình bày lặp lại nội dung**. Những nội dung đã trình bày chi tiết trong các chương trước không được trình bày lại trong chương này. Vì vậy, với nội dung hay, mang tính đóng góp/giải pháp, sinh viên chỉ nên tóm lược/mô tả sơ bộ trong các chương trước, đồng thời tạo tham chiếu chéo tới đề mục tương ứng trong Chương 5 này. Chi tiết thông tin về đóng góp/giải pháp được trình bày trong mục đó.

Ví dụ, trong Chương 4, sinh viên có thiết kế được kiến trúc đáng lưu ý gì đó, là sự kết hợp của các kiến trúc MVC, MVP, SOA, v.v. Khi đó, sinh viên sẽ chỉ mô tả ngắn gọn kiến trúc đó ở Chương 4, rồi thêm các câu có dạng: “Chi tiết về kiến trúc này sẽ được trình bày trong phần 5.1”.

¹Trong trường hợp phần này dưới 5 trang thì sinh viên nên gộp vào phần kết luận, không tách ra một chương riêng rẽ nữa.

CHƯƠNG 6. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

6.1 Kết luận

Sinh viên so sánh kết quả nghiên cứu hoặc sản phẩm của mình với các nghiên cứu hoặc sản phẩm tương tự.

Sinh viên phân tích trong suốt quá trình thực hiện ĐATN, mình đã làm được gì, chưa làm được gì, các đóng góp nổi bật là gì, và tổng hợp những bài học kinh nghiệm rút ra nếu có.

6.2 Hướng phát triển

Trong phần này, sinh viên trình bày định hướng công việc trong tương lai để hoàn thiện sản phẩm hoặc nghiên cứu của mình.

Trước tiên, sinh viên trình bày các công việc cần thiết để hoàn thiện các chức năng/nhiệm vụ đã làm. Sau đó sinh viên phân tích các hướng đi mới cho phép cải thiện và nâng cấp các chức năng/nhiệm vụ đã làm.

.