

TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI

BÁO CÁO BÀI TẬP LỚN LẬP TRÌNH HƯỚNG ĐỒI TƯỢNG

Tên đề tài: Phát triển ứng dụng web quản lý dự án

GVHD: PGS.TS Lê Đức Hậu

Nhóm 11 : Lê Viết Đức Anh – 20226075
Nguyễn Duy Anh – 20226103
Hồ Tuấn Anh – 20226100
Lê Phan Anh – 20226102
Đào Kiên Cường – 20226104
Vũ Duy Dũng – 20226078
Nguyễn Tài Hoan – 20226106
Nguyễn Hữu Khải – 20200310
Đỗ Đình Trung – 20226128

Lớp : (LT) LTHĐT Mã lớp học: 151964

Hà Nội, 01/ 2025

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	1
1.4 Bố cục đồ án	2
Chương 2 Khảo sát và phân tích yêu cầu	4
2.1 Khảo sát hiện trạng	4
2.1.1 Khảo sát người dùng/khách hàng	4
2.1.2 Các hệ thống đã có	4
2.1.3 Các ứng dụng tương tự	5
2.1.4 Mô tả sơ lược các tính năng phần mềm quan trọng cần phát triển.....	6
2.2 Tổng quan chức năng.....	8
2.2.1 Biểu đồ use case tổng quan	8
2.2.2 Biểu đồ use case phân rã XYZ	9
Chương 3 Công nghệ sử dụng.....	12
3.1 Ngôn ngữ lập trình.....	12
3.2 Framework và thư viện	12
3.3 Cơ sở dữ liệu.....	12
3.4 Công cụ hỗ trợ phát triển	12
3.5 Công nghệ triển khai.....	13
3.6 Lý do chọn lựa công nghệ.....	13
Chương 4 Phát triển và triển khai ứng dụng	14
4.1 Thiết kế kiến trúc	14
4.1.1 Lựa chọn kiến trúc phần mềm	14
4.1.2 Thiết kế biểu đồ lớp.....	15

4.1.3 Thiết kế cơ sở dữ liệu	17
4.1.4 Thiết kế giao diện	20
4.2 Các kỹ thuật lập trình hướng đối tượng đã sử dụng	22
4.2.1 Dependency injection	22
4.2.2 Singleton pattern.....	22
4.2.3 Inheritance (Kế thừa).....	22
4.2.4 Polymorphisism (Đa hình)	23
4.2.5 Encapsulation (Đóng gói).....	23
4.2.6 Delegation (Ủy thác)	23
4.3 Xây dựng ứng dụng	24
4.3.1 Thư viện và công cụ sử dụng	24
4.3.2 Kết quả đạt được.....	25
4.4 Kiểm thử	Error! Bookmark not defined.
4.4.1 Kiểm thử module “Đăng nhập/Đăng ký”	Error! Bookmark not defined.
4.4.2 Kiểm thử module “Quản lý dự án, nhiệm vụ, comment”.....	Error! Bookmark not defined.
4.4.3 Kiểm thử module “Quản lý thành viên trong dự án”.....	Error! Bookmark not defined.
4.4.4 Kiểm thử module “Quản lý tiến độ dự án”.....	Error! Bookmark not defined.
4.4.5 Kiểm thử module “Thông báo”	Error! Bookmark not defined.
4.4.6 Kiểm thử module “Tạo thống kê”	Error! Bookmark not defined.
4.5 Triển khai	25
Chương 5 Kết luận và hướng phát triển	28
5.1 Kết luận.....	28
5.2 Hướng phát triển	28

Chương 1 Giới thiệu đề tài

1.1 Đặt vấn đề

Trong bối cảnh cạnh tranh khốc liệt và sự chuyển đổi nhanh chóng của nền kinh tế, việc quản lý dự án hiện nay đang trở thành một yếu tố then chốt để đảm bảo sự thành công trong kinh doanh. Dự án quản lý không chỉ đơn thuần là việc xử lý công việc hàng ngày, mà còn bao gồm quá trình lên kế hoạch, phân bổ tài nguyên, đầu tư con người và đạt được mục tiêu đã đề ra. Tuy nhiên, trong thực tế, nhiều doanh nghiệp đang đối mặt với những thách thức như quản lý thời gian, giao tiếp đội nhóm, và đảm bảo đầy đủ tiến độ, nhân viên doanh nghiệp cũng cần lịch cụ thể, thời gian biểu cho việc theo dõi các nhiệm vụ được giao. Từ nhu cầu thực tiễn và cấp bách ấy, chúng em đã chọn đề tài “phát triển web quản lý dự án”.

1.2 Mục tiêu và phạm vi đề tài

Hiện nay, cũng đã có rất nhiều web quản lý dự án được triển khai ví dụ như ProjectManager, Trello,... Nhưng để có thể có trải nghiệm đầy đủ các chức năng của trang web thì cần trả phí thường niên. Dự án của chúng em xây nên cũng dựa trên các trang web đó, bao gồm các tính năng cơ bản như thêm, sửa, xóa dự án, thêm, phân công nhiệm vụ,... Ngoài ra còn có các chức năng khác như nhắc nhở khi sắp tới deadline, comment trong dự án hay trong nhiệm vụ, báo cáo tiến độ của dự án,... mà không cần trả phí.

1.3 Định hướng giải pháp

a) Định hướng, phương pháp, công nghệ sử dụng:

Dự án sẽ được xây dựng dựa trên công nghệ web application với kiến trúc client-server. Frontend được phát triển bằng HTML, CSS, JS để tạo giao diện người dùng thân thiện và phản hồi nhanh. Backend sử dụng Spring Boot để xây dựng các API RESTful phục vụ logic nghiệp vụ và giao tiếp với cơ sở dữ liệu. Postman sẽ được sử dụng nhằm mục đích kiểm thử API được viết ở trên. Dữ liệu sẽ được lưu trữ và quản lý bằng PostgreSQL.

b) Mô tả giải pháp:

Giải pháp là xây dựng một hệ thống quản lý dự án miễn phí, dễ sử dụng, tập trung vào các tính năng cơ bản như thêm, sửa, xóa dự án, phân công nhiệm vụ. Ngoài ra, các tính năng mở rộng như nhắc nhở deadline, báo cáo tiến độ và tương tác trong nhiệm vụ (bình luận) sẽ được tích hợp. Người dùng có thể truy cập thông qua giao diện web, đảm bảo tính ổn định và trải nghiệm tốt.

c) Đóng góp chính và kết quả đạt được:

Đóng góp chính: Cung cấp một giải pháp quản lý dự án miễn phí, phù hợp với các nhóm nhỏ và vừa, không yêu cầu người dùng phải trả phí để sử dụng đầy đủ tính năng.

Kết quả đạt được: Hệ thống quản lý dự án cho phép quản lý công việc dễ dàng, tự động hóa nhắc nhở và cập nhật tiến độ. Sản phẩm hoàn thiện sẽ hỗ trợ các nhóm cải thiện hiệu quả làm việc và tăng cường khả năng phối hợp trong dự án.

1.4 Bố cục đồ án

Chương 2: Khảo sát và phân tích yêu cầu

Chương này giới thiệu tổng quan về hệ thống quản lý dự án, bao gồm biểu đồ Use Case tổng quan để mô tả các chức năng chính của hệ thống. Đặc tả chi tiết từng chức năng như quản lý dự án, phân công nhiệm vụ, nhắc nhở deadline ...

Chương 3: Công nghệ sử dụng

Chương này trình bày về các công nghệ được sử dụng để phát triển hệ thống, bao gồm HTML, CSS, JS cho giao diện người dùng, Spring Boot để xây dựng backend, PostgreSQL để quản lý cơ sở dữ liệu và các công cụ hỗ trợ khác như Postman, GitHub,... Các lý do chọn lựa và vai trò của từng công nghệ trong hệ thống cũng được nêu rõ.

Chương 4: Phát triển và triển khai ứng dụng

Chương này mô tả chi tiết về thiết kế kiến trúc hệ thống, từ tổng quan đến chi tiết các module. Phần này bao gồm thiết kế giao diện, luồng hoạt động, và cơ sở dữ liệu. Các bước xây dựng ứng dụng, kiểm thử từng chức năng và triển khai hệ thống lên môi trường thực tế cũng được trình bày rõ ràng.

Chương 5: Kết luận và định hướng phát triển

Chương cuối cùng đánh giá những ưu, nhược điểm của hệ thống quản lý dự án. Các ưu điểm như tính miễn phí, dễ sử dụng, hỗ trợ quản lý hiệu quả được nhấn mạnh, đồng thời nêu các hạn chế cần khắc phục. Cuối cùng, chương trình bày định hướng phát triển hệ thống trong

tương lai, như phát triển thêm các chức năng, cải thiện giao diện, khắc phục các nhược điểm đã nêu trên.

Chương 2 Khảo sát và phân tích yêu cầu

2.1 Khảo sát hiện trạng

Trong giai đoạn khảo sát hiện trạng, mục tiêu chính là thu thập và phân tích các thông tin cần thiết về tình hình hiện tại của các hệ thống quản lý dự án, các yêu cầu của người dùng, cũng như những ứng dụng phần mềm hiện có trên thị trường. Dưới đây là ba nguồn chính mà chúng em sẽ khảo sát để đưa ra đánh giá về hệ thống phần mềm quản lý dự án.

2.1.1 Khảo sát người dùng/khách hàng

Người dùng là yếu tố quan trọng nhất trong việc thiết kế và phát triển phần mềm. Việc khảo sát người dùng giúp hiểu rõ các yêu cầu và mong muốn của họ đối với một hệ thống quản lý dự án. Phương pháp khảo sát có thể bao gồm các cuộc phỏng vấn, bảng câu hỏi, hoặc nhóm tập trung (focus groups). Các yêu cầu điển hình mà người dùng có thể yêu cầu từ một hệ thống quản lý dự án bao gồm:

Quản lý dự án: Tạo, sửa đổi và xóa dự án; theo dõi tiến độ và kết quả của các dự án.

Quản lý nhiệm vụ (Task Management): Phân công nhiệm vụ, thiết lập hạn chót, và theo dõi trạng thái công việc.

Theo dõi tiến độ: Các tính năng như biểu đồ Gantt, Kanban, hoặc các báo cáo chi tiết về tiến độ thực hiện nhiệm vụ và dự án.

Thống kê và báo cáo: Cung cấp các báo cáo định kỳ về tình hình tiến độ, hiệu suất của nhóm và dự án.

Bằng cách khảo sát người dùng, chúng em có thể nhận diện các tính năng cần thiết và mong muốn để phát triển hệ thống.

2.1.2 Các hệ thống đã có

Các hệ thống quản lý dự án hiện tại có thể cung cấp những thông tin quý giá về cách thức triển khai và các tính năng được sử dụng phổ biến trong ngành. Dưới đây là một số hệ thống nổi bật và một bảng so sánh ưu nhược điểm của từng hệ thống:

Hệ thống	Ưu điểm	Nhược điểm
Trello	<ul style="list-style-type: none"> - Giao diện trực quan, dễ sử dụng - Quản lý nhiệm vụ dễ dàng qua Kanban boards 	<ul style="list-style-type: none"> - Không phù hợp cho dự án phức tạp với nhiều tài nguyên. - Thiếu tính năng báo cáo và phân tích chi tiết.
Asana	<ul style="list-style-type: none"> - Hỗ trợ nhiều tính năng quản lý nhiệm vụ, theo dõi tiến độ - Báo cáo và phân tích tốt 	<ul style="list-style-type: none"> - Giao diện đôi khi quá phức tạp đối với người dùng mới. - Cần phải có một số kế hoạch trả phí để sử dụng tính năng nâng cao.
Jira	<ul style="list-style-type: none"> - Tính năng mạnh mẽ, phù hợp cho các dự án phần mềm phức tạp, Scrum/Kanban - Hỗ trợ tích hợp với các công cụ lập trình (Git, SVN, Jenkins, v.v.) 	<ul style="list-style-type: none"> - Giao diện và chức năng có thể gây khó khăn cho người dùng mới. - Cần cấu hình và đào tạo ban đầu khá phức tạp.

Nhìn chung, các hệ thống hiện tại đều có những điểm mạnh và hạn chế nhất định. Ví dụ, **Trello** rất dễ sử dụng nhưng thiếu tính năng báo cáo mạnh mẽ, trong khi **Jira** lại phù hợp cho các dự án phần mềm lớn nhưng khá phức tạp. Các hệ thống này đều thiếu tính năng tùy chỉnh chi tiết theo yêu cầu của người dùng, đặc biệt là đối với các dự án nhỏ hoặc nhóm nhỏ.

2.1.3 Các ứng dụng tương tự

Các ứng dụng tương tự như Basecamp, ClickUp, và Wrike cũng được khảo sát trong quá trình nghiên cứu. Các ứng dụng này cung cấp nhiều tính năng tương tự các hệ thống đã nêu, nhưng có sự khác biệt nhỏ về giao diện và phương thức tương tác với người dùng.

Basecamp: Dễ sử dụng nhưng hạn chế về tính năng quản lý thời gian và tài nguyên.

ClickUp: Rất mạnh mẽ trong việc tùy chỉnh các bảng công việc và tích hợp với các công cụ khác nhưng có thể gây rối mắt đối với người dùng mới.

Wrike: Cung cấp tính năng báo cáo và phân tích mạnh mẽ, nhưng giao diện không quá thân thiện với người dùng mới.

Phân tích các hệ thống và ứng dụng tương tự giúp nhận diện được các tính năng cần thiết cho phần mềm quản lý dự án, đồng thời chỉ ra những yếu điểm mà chúng ta có thể cải tiến.

2.1.4 Mô tả sơ lược các tính năng phần mềm quan trọng cần phát triển

Dựa trên khảo sát từ người dùng và các hệ thống hiện có, dưới đây là những tính năng quan trọng mà hệ thống quản lý dự án cần phát triển:

Quản lý dự án: Tạo, chỉnh sửa, xóa dự án, và phân công người tham gia.

Quản lý nhiệm vụ: Phân công nhiệm vụ, theo dõi tiến độ và thiết lập các mốc thời gian.

Quản lý tiến độ và báo cáo: Biểu đồ cột, báo cáo tình trạng tiến độ dự án.

Thông báo và nhắc nhở: Cảnh báo qua thông báo trong ứng dụng khi nhiệm vụ thay đổi trạng thái hoặc gần đến hạn.

Bằng cách phát triển các tính năng này, phần mềm quản lý dự án sẽ có thể đáp ứng được nhu cầu quản lý hiệu quả và linh hoạt, giúp người dùng dễ dàng theo dõi tiến độ, quản lý tài nguyên và tối ưu hóa quá trình làm việc nhóm.

Chúng em chia hệ thống các chức năng thành 5 module chính như sau:

Module 1: Các thao tác crud cơ bản

- Thêm/sửa/xóa dự án nvu comment
- Lọc và tìm kiếm dự án nvu
- Hiện thị tất cả dự án nvu

Module 2: Quản lý thành viên trong dự án

- Mời/xóa thành viên khỏi dự án
- Chấp nhận lời mời tham gia dự án
- Phân công nvu cho thành viên

Module 3: Quản lý tiến độ dự án

- Thành viên cập nhật tiến độ cho nvu

- Hệ thống tự động kiểm tra nvu quá hạn
- Gửi lời nhắc tới thành viên

Module 4: Thông báo

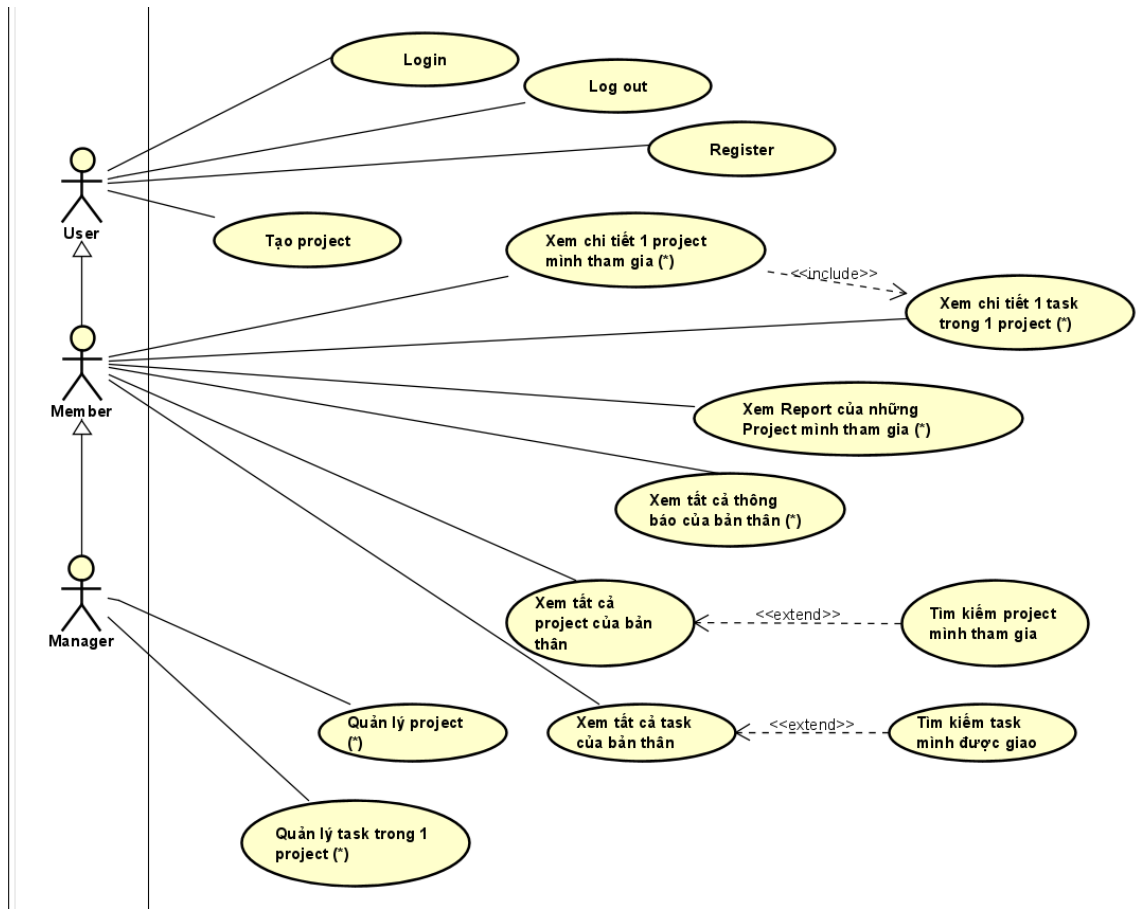
- Thông báo khi người dùng được phân công nvu
- Thông báo đến quản lý khi 1 thành viên hoàn thành nvu
- Thông báo tới quản lý khi người dùng đồng ý tham gia dự án

Module 5: Tạo thống kê

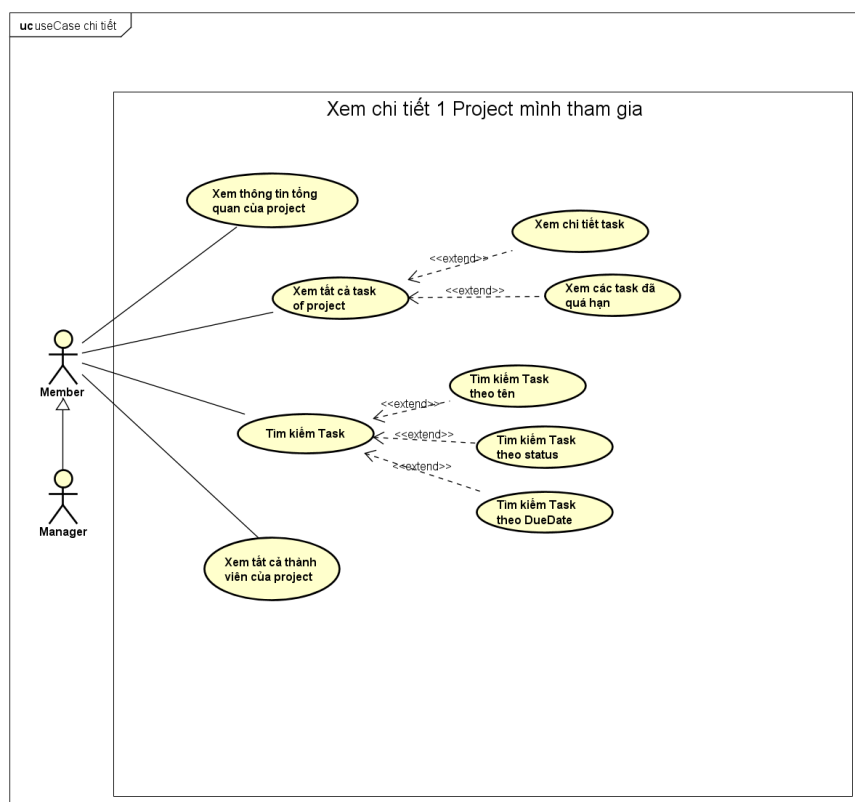
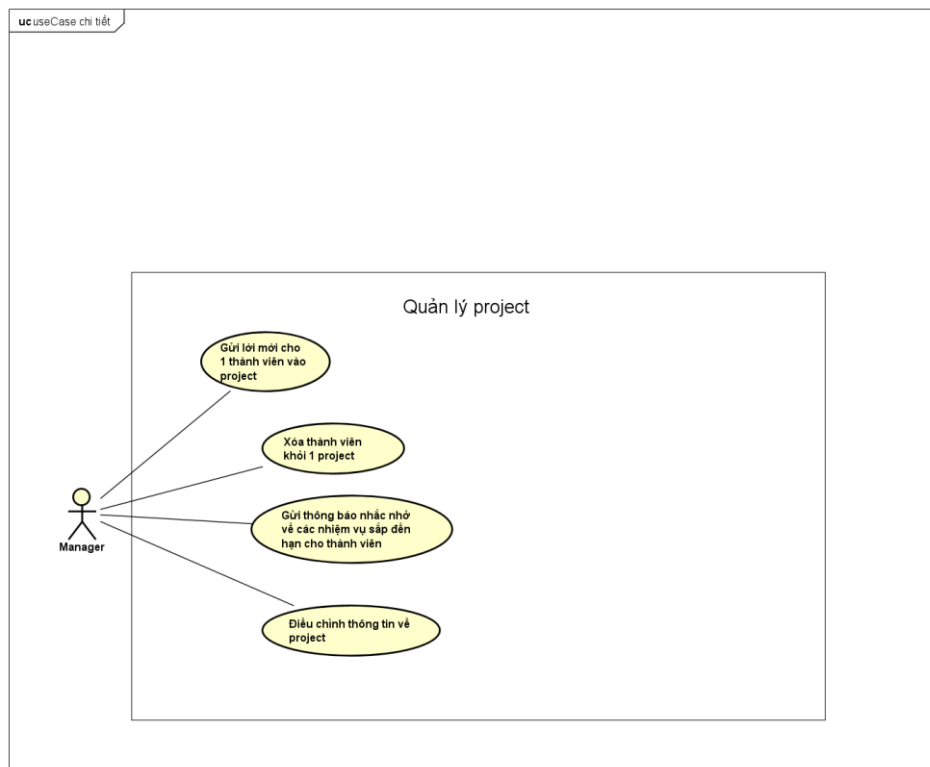
- Thống kê số nhiệm vụ 1 người dùng hoàn thành trong 7 ngày gần nhất
- Thống kê tỉ lệ hoàn thành nvu và số nvu hoàn thành của mỗi thành viên
- Thống kê về các nhiệm vụ có tỷ lệ hoàn thành thấp nhất/cao nhất

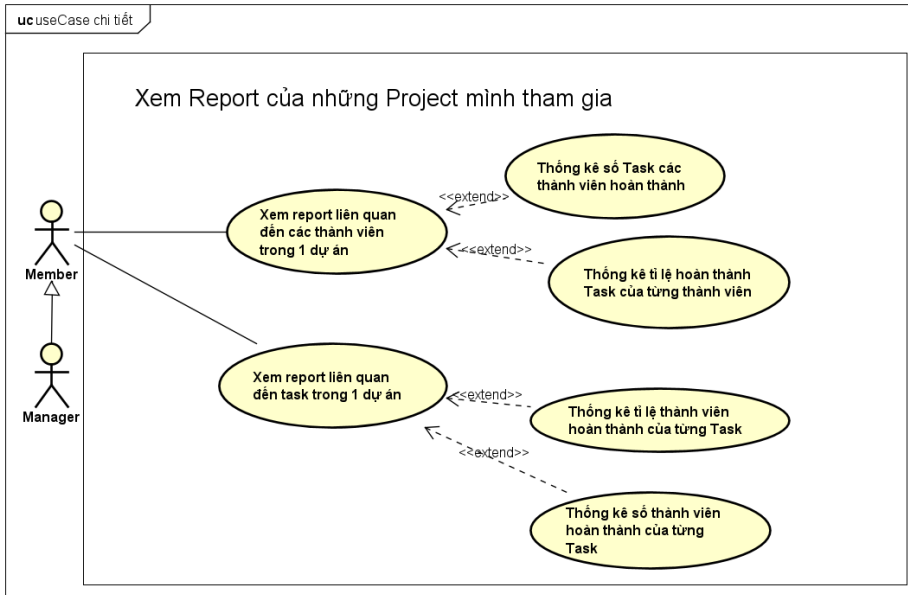
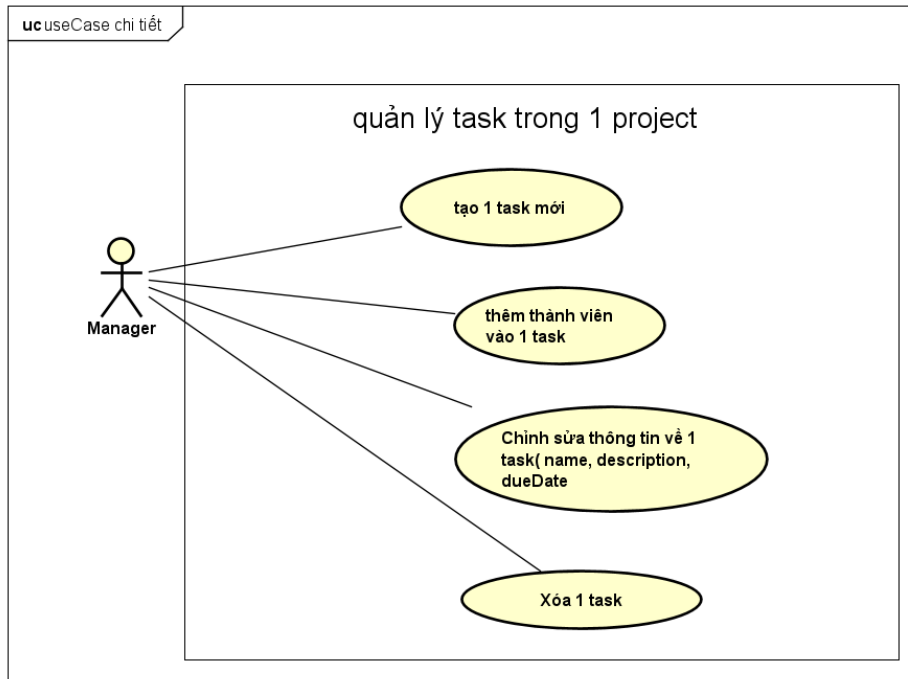
2.2 Tổng quan chức năng

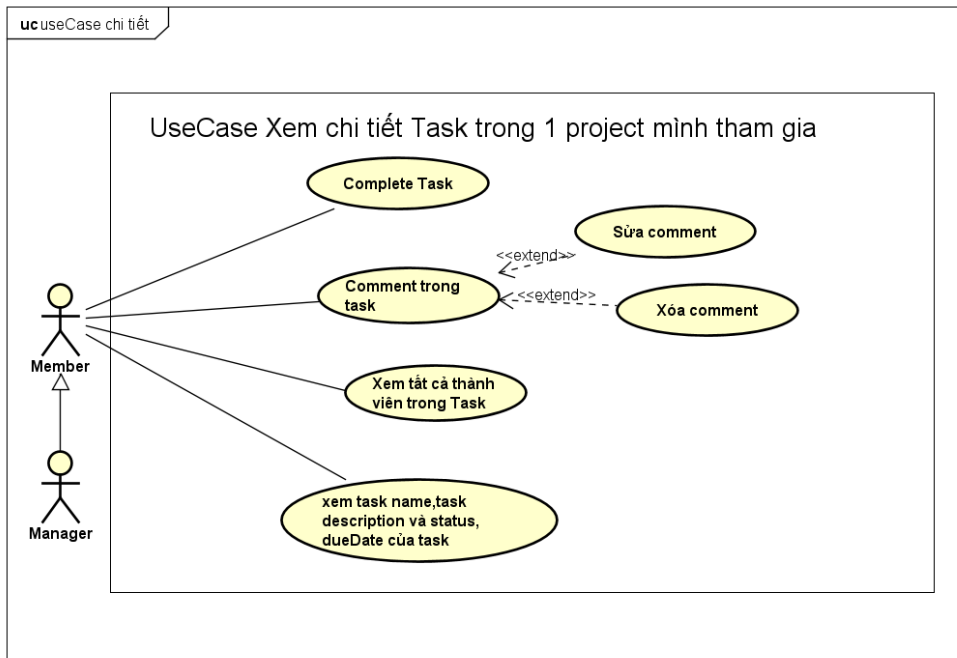
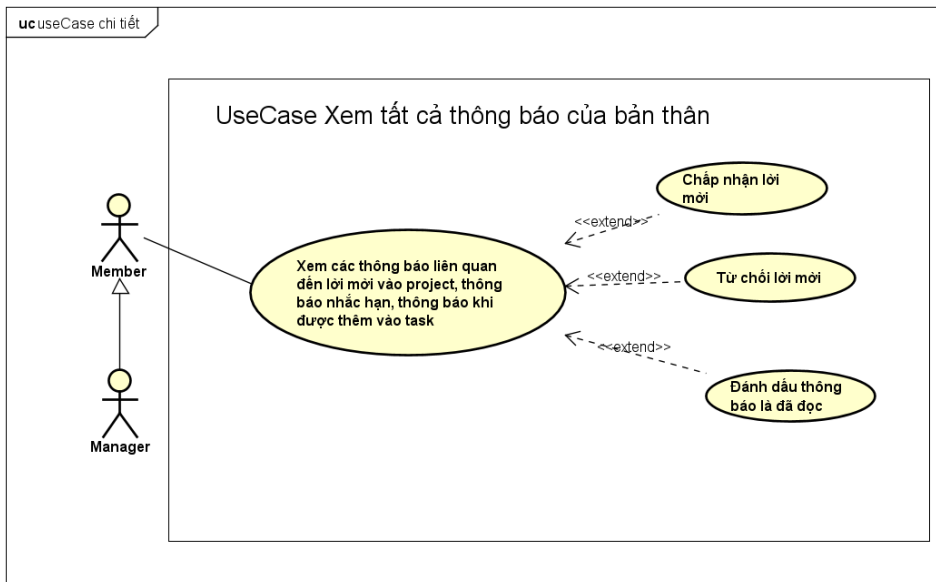
2.2.1 Biểu đồ use case tổng quan



2.2.2 Biểu đồ use case phân rã XYZ







Chương 3 Công nghệ sử dụng

Trong chương này, chúng em sẽ trình bày về các công nghệ, công cụ và nền tảng được sử dụng để phát triển hệ thống quản lý dự án. Việc lựa chọn các công nghệ này nhằm đảm bảo hệ thống có tính hiệu quả, dễ bảo trì và khả năng mở rộng trong tương lai.

3.1 Ngôn ngữ lập trình

Java: Được sử dụng để phát triển phần backend, Java cung cấp tính ổn định, hiệu năng cao và khả năng xử lý các yêu cầu nghiệp vụ phức tạp. Java còn hỗ trợ mạnh mẽ cho việc xây dựng các API RESTful thông qua các framework như Spring Boot.

JavaScript: Là ngôn ngữ chính để phát triển phần frontend. JavaScript kết hợp với HTML, CSS giúp xây dựng giao diện người dùng thân thiện và tương tác nhanh.

3.2 Framework và thư viện

Spring: Được sử dụng để phát triển backend, Spring hỗ trợ xây dựng các API RESTful nhanh chóng và tích hợp dễ dàng với cơ sở dữ liệu. Ngoài ra, Spring còn hỗ trợ quản lý các thành phần của ứng dụng một cách hiệu quả nhờ cấu trúc module rõ ràng.

3.3 Cơ sở dữ liệu

PostgreSQL: Hệ quản trị cơ sở dữ liệu quan hệ PostgreSQL được sử dụng để lưu trữ và quản lý dữ liệu của hệ thống. PostgreSQL có khả năng xử lý dữ liệu lớn, hỗ trợ mạnh mẽ các thao tác CRUD (Create, Read, Update, Delete) và dễ dàng tích hợp với Spring Boot.

3.4 Công cụ hỗ trợ phát triển

Postman: Được sử dụng để kiểm thử các API RESTful, đảm bảo rằng backend hoạt động đúng chức năng và dữ liệu truyền tải giữa frontend và backend chính xác.

GitHub: Là công cụ quản lý mã nguồn và hỗ trợ làm việc nhóm. GitHub giúp theo dõi lịch sử thay đổi của dự án và cho phép nhiều thành viên cùng phối hợp phát triển một cách hiệu quả.

IntelliJ Idea Ultimate Edition: Là môi trường phát triển được sử dụng để phát triển backend với Java. IntelliJ Idea cung cấp các tính năng mạnh mẽ để lập trình, kiểm thử, và gỡ lỗi.

Visual Studio Code: Môi trường phát triển có nhiều thư viện và extension đa dạng, là môi trường phát triển tích hợp (IDE) được sử dụng để phát triển phần frontend với HTML và CSS và cung cấp các tính năng mạnh mẽ như gợi ý mã, kiểm tra lỗi, tích hợp dễ dàng với các tiện ích mở rộng, giúp tăng hiệu quả lập trình và đảm bảo mã nguồn dễ bảo trì.

Eclipse: Eclipse được sử dụng như một môi trường phát triển tích hợp (IDE) cho việc lập trình backend bằng Java. Công cụ này cung cấp các tính năng mạnh mẽ như gợi ý mã, quản lý dự án và hỗ trợ debug, giúp tăng hiệu quả phát triển ứng dụng và giảm thiểu lỗi trong quá trình triển khai.

3.5 Công nghệ triển khai

Render: Render là nền tảng triển khai trực tuyến được sử dụng để triển khai hệ thống quản lý dự án. Render hỗ trợ tích hợp liên tục, cung cấp môi trường ổn định và khả năng mở rộng, đồng thời dễ dàng quản lý và tối ưu chi phí cho dự án.

Jenkins: Jenkins được sử dụng để tự động hóa các quy trình triển khai và kiểm thử, đảm bảo hệ thống luôn trong trạng thái sẵn sàng hoạt động. Công cụ này hỗ trợ tích hợp liên tục (CI) và triển khai liên tục (CD), giúp giảm thiểu rủi ro và tăng hiệu quả làm việc nhóm.

3.6 Lý do chọn lựa công nghệ

Tính ổn định: Các công nghệ như Java, Spring Boot và PostgreSQL được sử dụng phổ biến trong các hệ thống doanh nghiệp lớn, đảm bảo tính ổn định và bảo mật.

Dễ sử dụng và bảo trì: JS, HTML, CSS giúp dễ dàng phát triển và mở rộng giao diện.

Hiệu quả và tiết kiệm: Render và Jenkins cho phép triển khai nhanh chóng và tiết kiệm chi phí, phù hợp với các dự án sinh viên.

Chương 4 Phát triển và triển khai ứng dụng

4.1 Thiết kế kiến trúc

4.1.1 Lựa chọn kiến trúc phần mềm

Kiến trúc phần mềm : kiến trúc ba lớp MVC. Trong đó:

- M (Model): là nơi chứa những nghiệp vụ tương tác với dữ liệu bao gồm:
 - Lưu trữ trạng thái của ứng dụng trong các cấu trúc dữ liệu phù hợp hoặc cơ sở dữ liệu
 - Phản hồi những thay đổi của trạng thái
 - Xác thực và phân quyền người dùng
- V (View):
 - Hiển thị giao diện người dùng.
 - Gửi yêu cầu dữ liệu từ người dùng đến Controller
 - Nhận phản hồi từ Controller và cập nhật dữ liệu trên giao diện
- C (Controller):
 - Vận chuyển dữ liệu được gửi về từ View đến Model thông qua API
 - Kết xuất dữ liệu và thông báo từ Model và gửi phản hồi lên View thông qua API

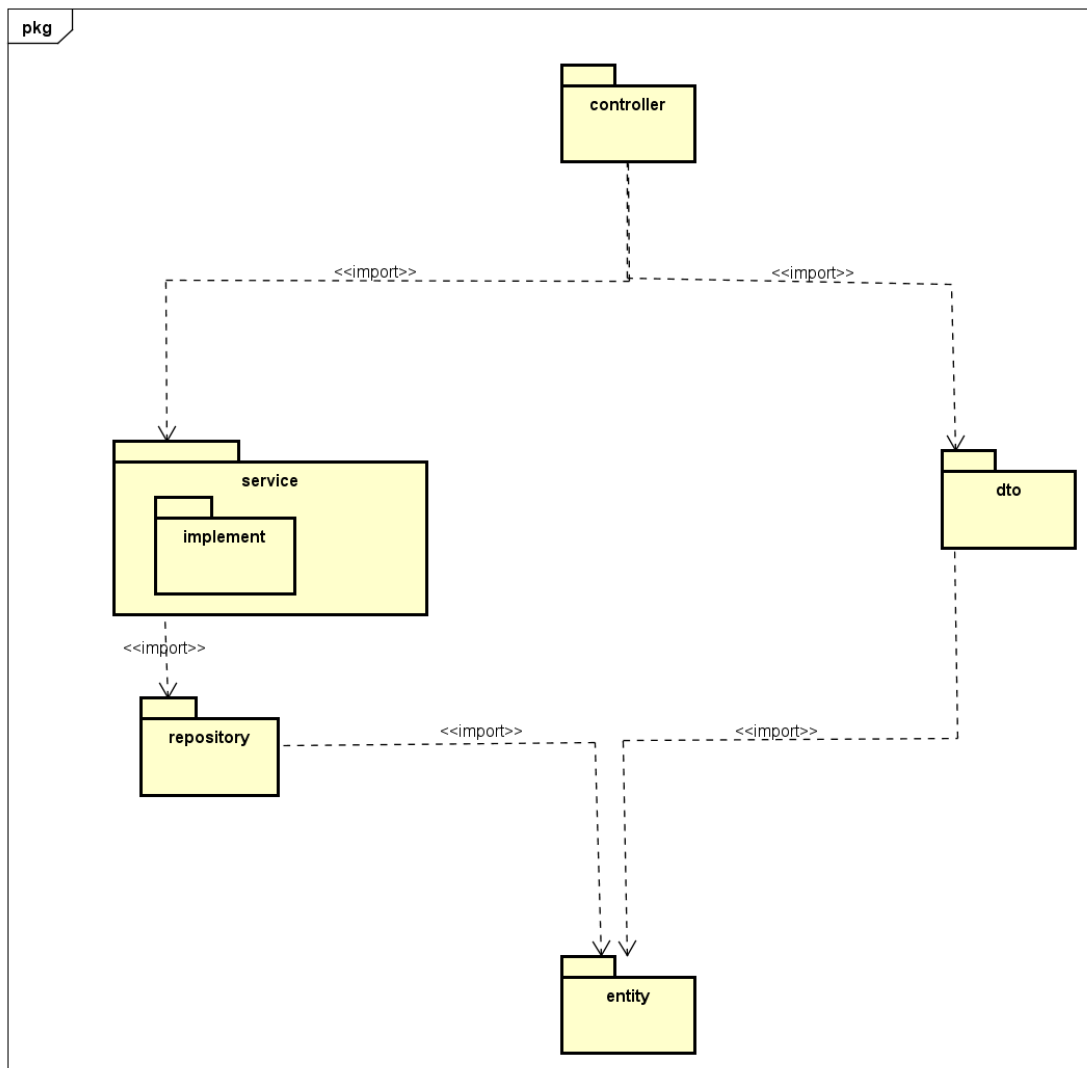
Triển khai trong ứng dụng Quản lý dự án:

- Model:
 - Dữ liệu liên quan tới thông tin người dùng: class User
 - Dữ liệu liên quan tới thông tin dự án: class Project
 - Dữ liệu liên quan tới thông tin nhiệm vụ: class Task
 - Dữ liệu liên quan tới bình luận, phản hồi: class Comment

- Dữ liệu liên quan tới thông báo: class Notification
- View:
 - Hiển thị giao diện thao tác cho người dùng
 - Bao gồm 3 màn hình chính: Home, chi tiết dự án và báo cáo.
 - Xử lý dữ liệu từ thao tác người dùng và gửi về controller.
 - Nhận dữ liệu từ controller để cập nhật dữ liệu cho người dùng.
- Controller:
 - Quản lý luồng điều khiển đối tượng người dùng: class UserController
 - Quản lý luồng điều khiển đối tượng dự án: class ProjectController
 - Quản lý luồng điều khiển đối tượng nhiệm vụ: class TaskController
 - Quản lý luồng điều khiển đối tượng bình luận, phản hồi: class CommentController
 - Quản lý luồng điều khiển đối tượng thông báo: class CommentController

4.1.2 Thiết kế biểu đồ lớp

Biểu đồ gói UML:



Chi tiết nhiệm vụ của từng package:

- Package entity: Chứa các lớp khai báo thuộc tính của đối tượng và ánh xạ đến các bảng trong cơ sở dữ liệu.
- Package repository: Chứa các giao diện khai báo các phương thức truy vấn dữ liệu trong database.
- Package dto: Chứa các lớp đóng gói data để chuyển giữa client – server, giúp giảm bớt lượng dữ liệu cần vận chuyển.
- Package service: Chứa các giao diện khai báo các phương thức nghiệp vụ như cập nhật, xử lý, thống kê dữ liệu. Package implement sẽ triển khai cụ thể từng phương thức được khai báo trong interface.
- Package controller: Chứa các lớp khai báo API endpoint cùng các phương thức chuyển đổi dữ liệu được phản hồi từ service sao cho phù hợp với kiểu dữ liệu yêu cầu từ phía client.

4.1.3 Thiết kế cơ sở dữ liệu

Mục tiêu thiết kế cơ sở dữ liệu:

- Đáp ứng yêu cầu quản lý dự án, bao gồm người dùng, dự án, nhiệm vụ, thông báo và các tương tác liên quan.
- Bảo đảm tính toàn vẹn dữ liệu, hỗ trợ khả năng mở rộng và tối ưu hoá hiệu suất.
- Phù hợp với hệ quản trị cơ sở dữ liệu PostgreSQL.

Biểu đồ thực thể liên kết (E-R Diagram)

- Các thực thể (Entities):
 - **Users (Người dùng)**: Lưu trữ thông tin người dùng như tài khoản, mật khẩu.
 - **Projects (Dự án)**: Quản lý thông tin dự án, bao gồm tên dự án, mô tả, và ngày tạo.
 - **Tasks (Nhiệm vụ)**: Theo dõi nhiệm vụ liên quan đến từng dự án, bao gồm trạng thái, ngày đến hạn.
 - **Comments (Bình luận)**: Lưu trữ bình luận của người dùng trên các nhiệm vụ.
 - **Notifications (Thông báo)**: Quản lý các thông báo liên quan đến dự án hoặc nhiệm vụ.
 - **Member_Task (Nhiệm vụ thành viên)**: Theo dõi trạng thái nhiệm vụ của từng thành viên.
- Các mối quan hệ (Relationships):
 - Một **user** có thể tham gia nhiều **projects**.
 - Một **project** có thể có nhiều **tasks**.
 - Một **task** có thể nhận nhiều **comments** từ các **users**.
 - Một **notification** có thể liên quan đến một **user** hoặc một **project/task**.
- Mô tả mối quan hệ:
 - **User và Projects**: Thông qua bảng trung gian **member_project**.
 - **Projects và Tasks**: Dự án chứa nhiều nhiệm vụ.
 - **Tasks và Comments**: Mỗi nhiệm vụ có thể có nhiều bình luận.
 - **Tasks và Notifications**: Nhiệm vụ tạo ra các thông báo liên quan.

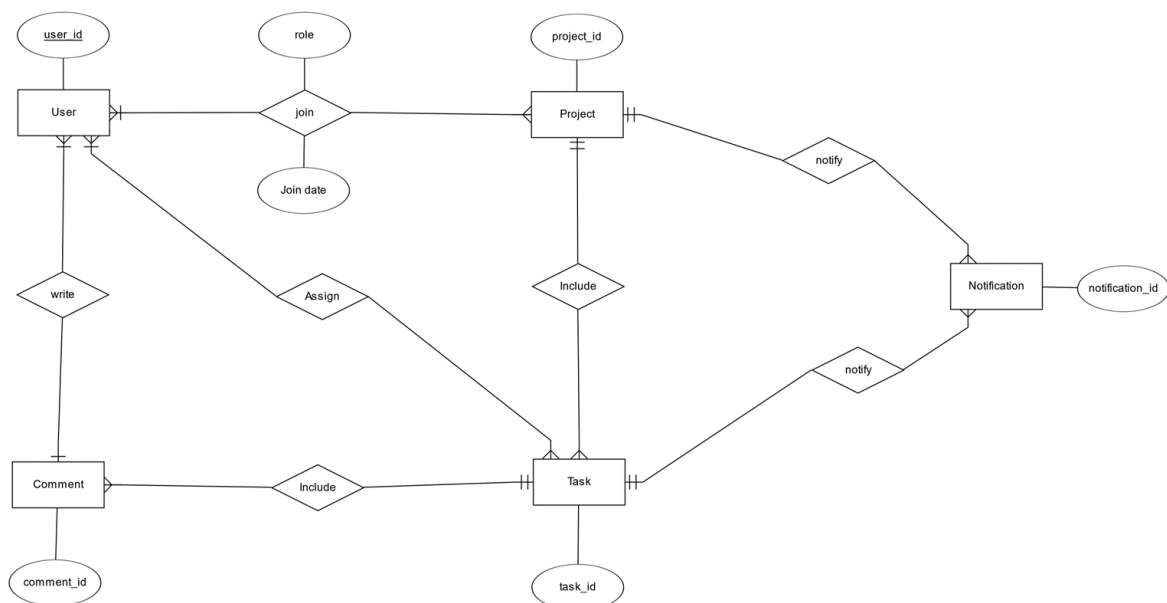
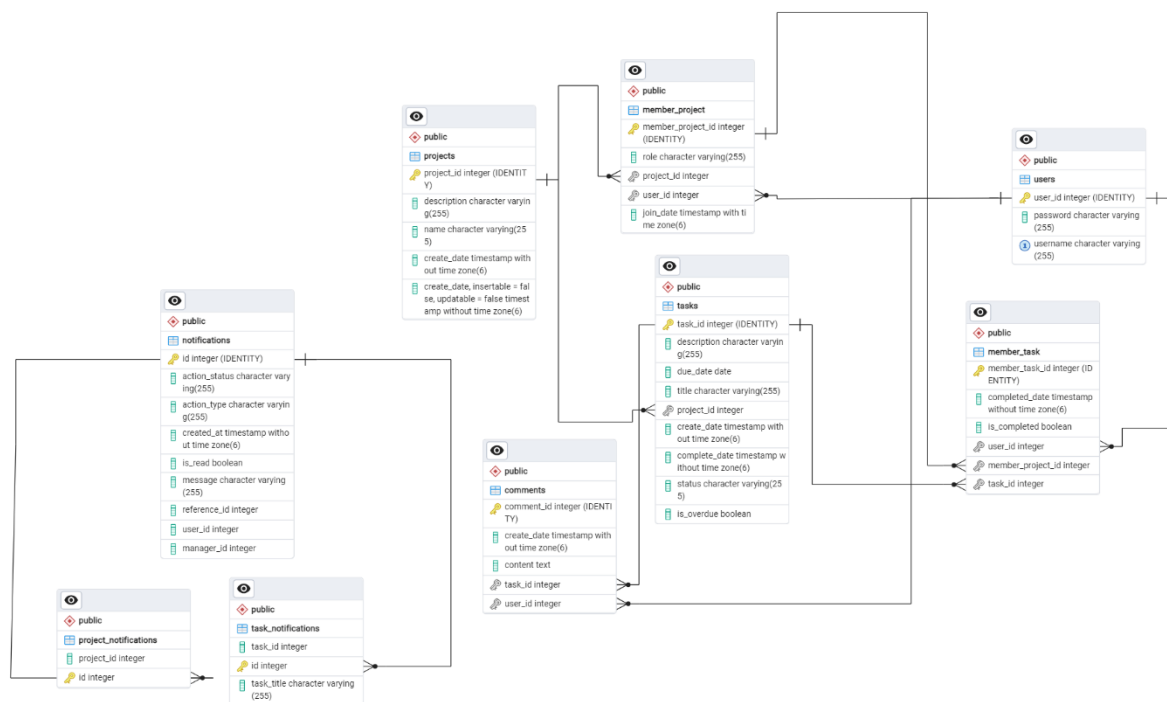


Figure 4-1: Thiết kế cơ sở dữ liệu

- Bảng **Users**:

Tên trường	Kiểu dữ liệu	Ràng buộc
user_id	Integer	PRIMARY KEY
username	Varchar(255)	UNIQUE, NOT NULL
password	Varchar(255)	NOT NULL

- Bảng **Projects**:

Tên trường	Kiểu dữ liệu	Ràng buộc
project_id	Integer	PRIMARY KEY
name	Varchar(255)	NOT NULL
description	Text	
create_date	Timestamp	

- Bảng **Task**:

Tên trường	Kiểu dữ liệu	Ràng buộc
task_id	Integer	PRIMARY KEY
project_id	Integer	FOREIGN KEY
title	Varchar(255)	NOT NULL
due_date	Date	

- Bảng **comments**:

Tên trường	Kiểu dữ liệu	Ràng buộc
------------	--------------	-----------

comment_id	Integer	PRIMARY KEY
task_id	Integer	FOREIGN KEY
content	Text	NOT NULL
create_date	Date	

- Các bảng phụ trợ:
 - o **Member_Project**: Quản lý vai trò của người dùng trong dự án.
 - o **Member_Task**: Quản lý trạng thái hoàn thành nhiệm vụ của từng thành viên.
 - o **Notifications**: Quản lý thông báo liên quan đến nhiệm vụ hoặc dự án.

Giải thích và lý do thiết kế

- E-R Diagram: Biểu đồ E-R cung cấp cái nhìn tổng quan về cấu trúc dữ liệu và mối quan hệ giữa các thực thể.
- Phân chia bảng:
 - o Các thực thể chính được biểu diễn thành các bảng riêng biệt để đảm bảo tính toàn vẹn dữ liệu.
 - o Các bảng trung gian như **member_project** và **member_task** giúp quản lý mối quan hệ nhiều-nhiều.
- Hiệu suất:
 - o Sử dụng khóa chính và khóa ngoại để liên kết dữ liệu chặt chẽ.
 - o Tạo chỉ mục cho các trường thường xuyên truy vấn như **username**, **project_id**, và **task_id**.
- Tính mở rộng: Cấu trúc cơ sở dữ liệu được thiết kế linh hoạt, có thể dễ dàng thêm các thực thể hoặc thuộc tính mới khi cần thiết.

4.1.4 Thiết kế giao diện

Mục tiêu:

- Giao diện của ứng dụng hướng tới việc mang lại trải nghiệm người dùng thân thiện, dễ sử dụng và phù hợp với các đối tượng mục tiêu.
- Các yếu tố thiết kế đảm bảo tính nhất quán, dễ nhận biết và hoạt động tốt trên các độ phân giải màn hình máy tính phổ biến.

Thông tin về màn hình

- Độ phân giải hỗ trợ:
 - o 1920x1080 (Full HD) là độ phân giải chính.
 - o Tương thích với các độ phân giải thấp hơn như 1600x900.
- Kích thước màn hình:
 - o Thiết kế tương thích với màn hình máy tính 14inch trở lên.
- Số lượng màu sắc:
 - o Ứng dụng hỗ trợ hiển thị trên màn hình 16.7 triệu màu (24-bit color depth).
 - o Đảm bảo tương phản và độ rõ ràng cho các màn hình chỉ hỗ trợ 16-bit màu.

Nguyên tắc và chuẩn hoá trong thiết kế giao diện

- Thiết kế nút bấm:
 - o Nút có kích thước tối thiểu 24x24 pixel để đảm bảo người dùng quan sát tốt và tương tác tốt khi dùng chuột.
 - o Sử dụng hiệu ứng hover hoặc nhấn để phản hồi hành động của người dùng.
- Thiết kế điều khiển:
 - o Các mẫu nhập liệu có khung viền rõ ràng, kích thước đủ lớn để nhập liệu đầy đủ và thoải mái.
 - o Thanh điều khiển cố định phía bên trái màn hình để chọn các chức năng.
- Vị trí hiển thị các thông điệp phản hồi:
 - o Các thông báo được hiển thị dưới dạng các toastify hoặc các alert, sử dụng màu sắc để phân biệt các trạng thái:
 - Màu đỏ cho thông báo lỗi.
 - Màu xanh cho thông báo thành công.
- Phối màu:
 - o Tông màu chính: #EBDFD7 (Hồng phấn nhẹ), #000 (Đen).
 - o Tông màu phụ: #F2EAE5 (Trắng kem).
 - o Tông màu nhấn mạnh: #E65F2B (Cam đất).
- Font chữ:
 - o Sử dụng font chữ Opens San, cỡ chữ tối thiểu 12px để dễ đọc.
 - o Tất cả tiêu đề đều in đậm.

Hình ảnh minh hoạ giao diện

- Trang đăng nhập/đăng ký:
 - o Form nhập liệu gồm các trường username, password (Đối với giao diện đăng nhập) và username, password, confirm password (Đối với giao diện đăng ký).
 - o Nút “Đăng nhập”, “Đăng ký” và các nút chức năng nằm ở dưới cùng, căn giữa.

4.2 Các kỹ thuật lập trình hướng đối tượng đã sử dụng

4.2.1 Dependency injection

Trong hệ thống này, các lớp service và controller đang sử dụng Dependency Injection để quản lý phụ thuộc. Điều này giúp cải thiện khả năng mở rộng và bảo trì mã nguồn.

Ví dụ:

- Các service như `ProjectService`, `UserService`, `TaskService` có thể được tiêm vào Controller thông qua constructor hoặc annotation như `@Autowired` trong Spring Framework.
- `UserRepository`, `ProjectRepository`, `TaskRepository` được sử dụng trong các service, có thể được tiêm vào thông qua Spring Data JPA.

Điều này giúp code tách biệt và dễ kiểm thử hơn.

4.2.2 Singleton pattern

Singleton thường được áp dụng cho các service hoặc component có tính chất toàn cục và cần quản lý trạng thái duy nhất.

- Trong sơ đồ, các lớp Service như `ProjectService`, `UserService` có thể được định nghĩa là **singleton bean** trong Spring, nhờ vào annotation `@Service`.
- Các lớp repository như `UserRepository`, `ProjectRepository` cũng có thể được Spring quản lý như singleton instance.

Mẫu thiết kế này giúp tiết kiệm tài nguyên và tránh việc khởi tạo nhiều đối tượng không cần thiết.

4.2.3 Inheritance (Kế thừa)

Kế thừa là một trong những nguyên lý quan trọng của lập trình hướng đối tượng, giúp tái sử dụng mã nguồn và mở rộng chức năng.

Ví dụ từ sơ đồ:

- `ProjectNotification` và `TaskNotification` là các lớp con mở rộng quan hệ giữa `Notifications`, giúp tổ chức dữ liệu theo hướng kế thừa.

Điều này giúp giảm trùng lặp code và tăng khả năng mở rộng.

4.2.4 Polymorphism (Đa hình)

Đa hình có thể xuất hiện trong cách các lớp interface và abstract class được sử dụng.

Ví dụ:

- `Repository` có thể sử dụng các interface như `JpaRepository` trong Spring, giúp đa hình hóa các thao tác CRUD mà không cần viết lại code.
- `NotificationController` có thể xử lý nhiều loại thông báo khác nhau dựa trên kiểu dữ liệu hoặc logic runtime, ví dụ gửi email, thông báo trong hệ thống, v.v.

Đa hình giúp hệ thống linh hoạt hơn khi mở rộng.

4.2.5 Encapsulation (Đóng gói)

Đóng gói giúp bảo vệ dữ liệu bên trong các đối tượng và chỉ cho phép truy cập thông qua các phương thức xác định.

Ví dụ:

- Các lớp `User`, `Project`, `Task`, `Comment` đều có các thuộc tính `private`, giúp ẩn dữ liệu và chỉ truy cập thông qua getter/setter.
- Các lớp `Service` chỉ cung cấp một số phương thức `public`, giấu đi logic xử lý bên trong.

Điều này giúp bảo mật dữ liệu và giảm nguy cơ lỗi do thay đổi trực tiếp dữ liệu bên ngoài.

4.2.6 Delegation (Ủy thác)

Ủy thác (delegation) giúp phân chia trách nhiệm giữa các lớp, tránh việc một lớp phải xử lý quá nhiều logic.

Ví dụ:

- ProjectController ủy thác xử lý nghiệp vụ cho ProjectService, thay vì thực hiện trực tiếp.
- ProjectService có thể ủy thác công việc lấy dữ liệu cho ProjectRepository.
- TaskService có thể sử dụng UserRepository để kiểm tra thông tin người dùng trước khi giao task.

Điều này giúp mã nguồn rõ ràng hơn, dễ bảo trì và mở rộng.

4.3 Xây dựng ứng dụng

4.3.1 Thư viện và công cụ sử dụng

Bảng 1 Danh sách thư viện và công cụ sử dụng

Mục đích	Công cụ	Địa chỉ URL
IDE lập trình	Eclipse Oxygen 64 bit	http://www.eclipse.org/
IDE lập trình	IntelliJ IDEA Ultimate 64 bit	https://www.jetbrains.com/
IDE lập trình	Visual Studio Code 64 bit 1.96.2	https://code.visualstudio.com/
Kiểm thử API	Postman 64 bit	https://www.postman.com/
Quản trị cơ sở dữ liệu quan hệ	PostgreSQL 64 bit 16	https://www.postgresql.org/
Triển khai ứng dụng	Render	https://render.com/
Triển khai ứng dụng	Jenkins 2.491	https://www.jenkins.io/

4.3.2 Kết quả đạt được

Web quản lý dự án đã hoàn thành, đảm bảo các tính năng cũng như giao diện. Với phần backend được phát triển bởi Java với Spring Boot, code được viết bằng IntelliJ Ultimate và Eclipse. Frontend được xây dựng bằng Javascripts, HTML, CSS, được code bằng IDE lập trình VSCode. Web được triển khai bằng Jenkins.

Bảng 2 Thống kê về backend

Thông số backend	Giá trị
Tổng số dòng code	2806 dòng
Tổng số gói	9 gói
Tổng số file	63 file
Dung lượng mã nguồn	683 kb

Bảng 3 Thống kê về frontend

Thông số	Giá trị
Tổng số dòng code	3227 dòng
Tổng số file	26 file
Dung lượng mã nguồn	6.17 MB

4.4 Triển khai

Mô hình triển khai: Mô hình client-server giao tiếp thông qua giao thức HTTPS

Máy chủ triển khai: Ubuntu Server 22.04 LTS

Cấu hình chi tiết:

- CPU core: 1
- Ram: 4GB
- Dung lượng ổ cứng: 30 GB
- Máy chủ web: Apache Tomcat

Kết quả thử nghiệm:

Performance Test Report - Dec 28, 2024 (#15)

Open in Postman

Postman collection: OOP-PMS
Report exported on: Dec 28, 2024, 11:50:14 (GMT+7)

Test setup

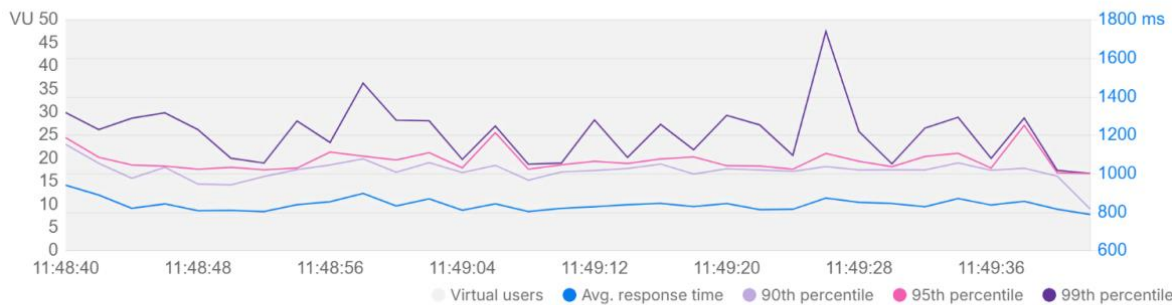
Virtual users	Start time	Load profile
50 VU	Dec 28, 11:48:35 (GMT+7)	Fixed
Duration	End time	Environment
1 minute	Dec 28, 11:49:42 (GMT+7)	New Environment

1. Summary

Total requests sent	Throughput	Average response time	Error rate
1,608	23.92 requests/second	842 ms	0.00 %

1.1 Response time

Response time trends during the test duration.



1.2 Throughput

Rate of requests sent per second during the test duration.

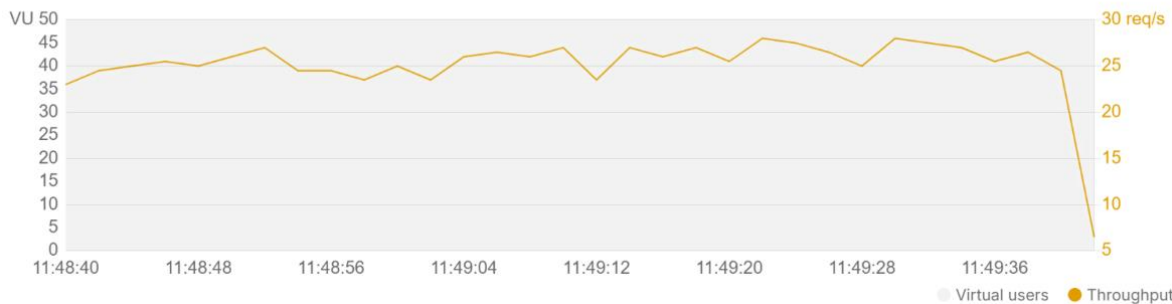


Figure 4-20: Kết quả thử nghiệm với 50 người dùng ảo truy cập

Chương 5 Kết luận và hướng phát triển

5.1 Kết luận

Phần lớn các chức năng trong phần mềm gần như đáp ứng yêu cầu đề án. Trong quá trình làm đề án còn gặp khá nhiều khó khăn trong lập trình, viết báo cáo cũng như quản lý, sắp xếp thời gian làm việc nhóm. Bên cạnh đó vẫn còn một số lỗi nhóm cần khắc phục như giao diện còn thô sơ, các lỗi tiềm ẩn trong quá trình thực thi, chưa kiểm tra được hiệu suất web, cần cải thiện thêm chức năng để tăng tương tác với người dùng, ...

5.2 Hướng phát triển

Trong tương lai, nhóm có ý định phát triển ứng dụng thành ứng dụng đa nền tảng. Nhóm cũng cần cải thiện giao diện thân thiện với người dùng hơn và mở rộng thêm nhiều chức năng khác như cho phép up load file hay tạo báo cáo cho dự án.