

BỘ CÔNG THƯƠNG  
TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI

---

ĐỒ ÁN TỐT NGHIỆP ĐẠI HỌC  
NGÀNH KỸ THUẬT PHẦN MỀM

XÂY DỰNG WEBSITE QUẢN LÝ CÔNG VIỆC  
CHO DOANH NGHIỆP CVS

CBHD: TS. Nguyễn Văn Tỉnh

Sinh viên: Vũ Anh Tú

Mã số sinh viên: 2020606774

Hà Nội – Năm 2024

## LỜI CẢM ƠN

Để đồ án tốt nghiệp được thực hiện thành công, em xin cảm ơn giảng viên hướng dẫn TS. Nguyễn Văn Tỉnh, Khoa Công nghệ thông tin, Trường đại học Công nghiệp Hà Nội đã chia sẻ những kiến thức quý báu, hướng dẫn tận tình và có những góp ý trong suốt quá trình thực hiện để tài để em có thể hoàn thành đồ án một cách tốt nhất.

Sau khi nghiên cứu đề tài và kết thúc khóa học, em đã học hỏi và tích lũy được kiến thức và kinh nghiệm từ giáo viên hướng dẫn để hoàn thiện và phát triển bản thân. Bên cạnh đó, đây cũng là cơ hội giúp chúng em nhận ra mình cần hoàn thiện thêm những gì để chuẩn bị cho một hành trình dài phía trước.

*Em xin chân thành cảm ơn thầy Nguyễn Văn Tỉnh đã giúp đỡ chúng em thực hiện đề tài này!*

Hà Nội, ngày 18 tháng 12 năm 2024

## MỤC LỤC

LỜI CẢM ƠN .....	i
MỤC LỤC.....	ii
DANH MỤC CÁC HÌNH ẢNH .....	iv
DANH MỤC BẢNG.....	vi
MỞ ĐẦU.....	1
Lý do chọn đề tài.....	1
Đối tượng, mục tiêu và phạm vi nghiên cứu .....	1
Ý nghĩa khoa học và thực tiễn của đề tài .....	2
Bố cục đề tài.....	3
CHƯƠNG 1. TỔNG QUAN VỀ NỘI DUNG NGHIÊN CỨU .....	5
1.1. Mô tả bài toán .....	5
1.2. Mô hình quy trình .....	6
1.3. Công nghệ áp dụng .....	7
1.3.1. ReactJS.....	7
1.3.2. NodeJs.....	9
1.3.3. Hệ quản trị cơ sở dữ liệu MySQL .....	10
CHƯƠNG 2. KHẢO SÁT VÀ PHÂN TÍCH THIẾT KẾ HỆ THỐNG.....	12
2.1. Khảo sát bài toán.....	12
2.2. Phân tích và thiết kế hệ thống.....	13
2.2.1. Biểu đồ Use case tổng quan.....	14
2.2.2. Biểu đồ lớp của hệ thống .....	15
2.2.3. Đặc tả các use case.....	15
2.2.4. Hiện thực hóa use case.....	33
2.2.5. Mô hình hóa dữ liệu.....	60
2.2.6. Thiết kế các bảng dữ liệu .....	62
2.2.7. Thiết kế giao diện .....	66
CHƯƠNG 3. TRIỂN KHAI, KẾT QUẢ VÀ KIỂM THỦ .....	71
3.1. Triển khai hệ thống .....	71
3.1.1. Phía back-end.....	71
3.1.2. Phía front-end.....	71

3.2. Giao diện của hệ thống .....	72
3.2.1. Giao diện phía người dùng .....	72
3.2.2. Giao diện phía quản trị.....	79
3.3. Kiểm thử chức năng hệ thống.....	81
3.3.1. Kiểm thử chức năng người dùng .....	81
3.3.2. Báo cáo kiểm thử .....	85
KẾT LUẬN.....	86
TÀI LIỆU THAM KHẢO .....	87

## DANH MỤC CÁC HÌNH ẢNH

Hình 1. 1. Các giai đoạn mô hình thác nước .....	6
Hình 1. 2. Thư viện ReactJS.....	7
Hình 1. 3. NodeJS Image.....	9
Hình 1. 4. Hệ quản trị cơ sở dữ liệu MySQL .....	10
Hình 2. 1. Biểu đồ usecase tổng quan của hệ thống.....	14
Hình 2. 2. Biểu đồ lớp của hệ thống.....	15
Hình 2. 3. Biểu đồ tuần tự use case Thêm tài khoản.....	33
Hình 2. 4. VOPC use case Thêm tài khoản.....	34
Hình 2. 5. Biểu đồ tuần tự use case Đăng nhập .....	35
Hình 2. 6. VOPC use case Đăng nhập.....	36
Hình 2. 7. Biểu đồ tuần tự use case Quên mật khẩu .....	37
Hình 2. 8. VOPC use case Quên mật khẩu .....	38
Hình 2. 9. Biểu đồ tuần tự use case Cập nhật thông tin tài khoản .....	39
Hình 2. 10. VOPC use case Cập nhật thông tin tài khoản .....	40
Hình 2. 11. Biểu đồ tuần tự use case Quản lý tài khoản (Admin) .....	41
Hình 2. 12. VOPC use case Quản lý tài khoản (Admin).....	42
Hình 2. 13. Biểu đồ tuần tự use case Tạo project .....	43
Hình 2. 14. VOPC use case Tạo project.....	44
Hình 2. 15. Biểu đồ tuần tự use case Chính sửa project .....	45
Hình 2. 16. VOPC use case Chính sửa project.....	46
Hình 2. 17. Biểu đồ tuần tự use case Xóa project .....	47
Hình 2. 18. VOPC use case Xóa project .....	48
Hình 2. 19. Biểu đồ tuần tự use case Tạo sprint.....	49
Hình 2. 20. VOPC use case Tạo sprint.....	49
Hình 2. 21. Biểu đồ tuần tự Chính sửa sprint.....	50
Hình 2. 22. VOPC use case Chính sửa Sprint .....	50
Hình 2. 23. Biểu đồ tuần tự use case Kết thúc sprint .....	51
Hình 2. 24. VOPC use case Kết thúc sprint .....	51
Hình 2. 25. Biểu đồ tuần tự use case Tạo task .....	52
Hình 2. 26. VOPC use case Tạo task .....	52
Hình 2. 27. Biểu đồ tuần tự use case Xem chi tiết/Chỉnh sửa task .....	53
Hình 2. 28. VOPC use case Xem chi tiết/chỉnh sửa task .....	54
Hình 2. 29. Biểu đồ tuần tự use case Search/filter task.....	55
Hình 2. 30. VOPC use case Search/filter task.....	55
Hình 2. 31. Biểu đồ tuần tự use case Xóa task.....	56
Hình 2. 32. VOPC use case Xóa task .....	56
Hình 2. 33. Biểu đồ tuần tự use case Comment task.....	57
Hình 2. 34. VOPC use case Comment task.....	57

Hình 2. 35. Biểu đồ tuần tự use case Chính sửa comment.....	58
Hình 2. 36. VOPC use case Chính sửa comment.....	58
Hình 2. 37. Biểu đồ tuần tự use case Xóa comment .....	59
Hình 2. 38. VOPC use case Xóa comment .....	60
Hình 2. 39. Mô hình cơ sở dữ liệu hệ thống.....	62
Hình 2. 40. Thiết kế bảng user .....	62
Hình 2. 41. Thiết kế bảng auth .....	63
Hình 2. 42. Thiết kế bảng project.....	63
Hình 2. 43. Thiết kế bảng project_type .....	63
Hình 2. 44. Thiết kế bảng sprint.....	64
Hình 2. 45. Thiết kế bảng task .....	64
Hình 2. 46. Thiết kế bảng status .....	64
Hình 2. 47. Thiết kế bảng action .....	65
Hình 2. 48. Thiết kế bảng comment .....	65
Hình 2. 49. Thiết kế bảng file.....	65
Hình 2. 50. Thiết kế giao diện trang chủ .....	66
Hình 2. 51. Thiết kế giao diện màn hình Đăng nhập .....	66
Hình 2. 52. Thiết kế giao diện màn hình Chi tiết project.....	67
Hình 2. 53. Thiết kế giao diện màn hình Thêm sprint .....	67
Hình 2. 54. Thiết kế giao diện màn hình Chi tiết sprint.....	68
Hình 2. 55. Thiết kế giao diện màn hình Chi tiết task .....	68
Hình 2. 56. Thiết kế giao diện màn hình Thêm task .....	69
Hình 2. 57. Thiết kế giao diện màn hình Quản lý tài khoản .....	69
Hình 2. 58. Thiết kế giao diện màn hình Thêm tài khoản.....	70
Hình 3. 1. Giao diện đăng nhập.....	72
Hình 3. 2. Giao diện reset mật khẩu.....	72
Hình 3. 3. Giao diện thay đổi mật khẩu .....	73
Hình 3. 4. Giao diện trang chủ .....	73
Hình 3. 5. Giao diện chi tiết sprint .....	74
Hình 3. 6. Giao diện chi tiết task .....	74
Hình 3. 7. Giao diện filter task .....	75
Hình 3. 8. Giao diện tạo project .....	76
Hình 3. 9. Giao diện form tạo sprint .....	77
Hình 3. 10. Giao diện form tạo task .....	78
Hình 3. 11. Giao diện Profile .....	79
Hình 3. 12. Giao diện quản lý người dùng.....	79
Hình 3. 13. Giao diện thêm user.....	80
Hình 3. 14. Giao diện xác nhận gmail.....	80
Hình 3. 15. Giao diện quản lý loại project .....	81

## **DANH MỤC BẢNG**

Bảng 2. 1. Mô hình hóa chức năng.....	13
Bảng 2. 2. Use case thêm tài khoản.....	15
Bảng 2. 3. Mô tả chi tiết use case Đăng nhập. ....	17
Bảng 2. 4. Mô tả chi tiết use case Đăng nhập .....	18
Bảng 2. 5. Mô tả chi tiết use case Cập nhật thông tin tài khoản .....	19
Bảng 2. 6. Mô tả chi tiết use case Quản lý tài khoản (Admin) .....	20
Bảng 2. 7. Mô tả chi tiết use case Tạo project .....	21
Bảng 2. 8. Mô tả chi tiết use case Chính sửa project .....	22
Bảng 2. 9. Mô tả chi tiết use case Xóa project.....	23
Bảng 2. 10. Mô tả chi tiết use case tạo sprint.....	24
Bảng 2. 11. Mô tả chi tiết use case Chính sửa sprint .....	24
Bảng 2. 12. Mô tả chi tiết use case Kết thúc sprint.....	25
Bảng 2. 13. Mô tả chi tiết use case Tạo task .....	26
Bảng 2. 14. Mô tả chi tiết use case Xem chi tiết/Chỉnh sửa task .....	27
Bảng 2. 15. Mô tả chi tiết use case search/filter Task .....	28
Bảng 2. 16. Mô tả chi tiết use case Xóa task.....	29
Bảng 2. 17. Mô tả chi tiết use case Comment task .....	30
Bảng 2. 18. Mô tả chi tiết use case Chính sửa Comment.....	30
Bảng 2. 19. Mô tả chi tiết use case Xóa Comment. ....	31
Bảng 3. 1. Danh sách test case chức năng hệ thống.....	81
Bảng 3. 2. Báo cáo kiểm thử .....	85

## MỞ ĐẦU

### Lý do chọn đề tài

Trong bối cảnh hiện nay, quản lý công việc một cách hiệu quả đã trở thành yếu tố quan trọng đối với mọi cá nhân và doanh nghiệp. Công việc ngày càng phức tạp và yêu cầu về năng suất, sự chuyên nghiệp càng cao. Việc quản lý thông tin và theo dõi tiến độ công việc không chỉ giúp tiết kiệm thời gian mà còn nâng cao chất lượng công việc. Đây là lý do em lựa chọn đề tài xây dựng một hệ thống website quản lý công việc.

- **Hỗ trợ quản lý công việc toàn diện:** Việc có một hệ thống quản lý công việc trực tuyến giúp người dùng dễ dàng theo dõi tiến độ công việc, phân chia nhiệm vụ, quản lý thời gian một cách trực quan và tiện lợi.
- **Tăng cường hiệu quả công việc:** Thay vì quản lý công việc thủ công qua sổ sách hoặc các công cụ truyền thống khác, một hệ thống trực tuyến giúp tối ưu hóa quy trình quản lý công việc, giúp người dùng nắm bắt thông tin nhanh chóng và kịp thời.
- **Giảm thiểu sai sót và quên công việc:** Với các tính năng nhắc nhở, theo dõi tiến độ công việc, phân bổ thời gian hợp lý, hệ thống quản lý công việc trực tuyến giúp giảm thiểu các sai sót, tăng tính chính xác, và nâng cao chất lượng công việc.

Dựa trên những lý do trên, em quyết định chọn đề tài xây dựng hệ thống website quản lý công việc cho doanh nghiệp CVS. Mục tiêu là tạo ra một công cụ hỗ trợ hiệu quả cho tổ chức trong việc quản lý công việc hàng ngày, góp phần giúp công việc trở nên dễ dàng và có tổ chức hơn.

### Đối tượng, mục tiêu và phạm vi nghiên cứu

† Đối tượng nghiên cứu bao gồm:

- Website quản lý công việc CVS Tasks (CVS-TM).
- Công nghệ: ReactJs, NodeJs.
- Hệ quản trị MySQL.

† Mục đích nghiên cứu:

- Xác định và làm rõ các vấn đề liên quan đến đề tài.
- Khảo sát, phân tích thiết kế hệ thống cho website.
- Cung cấp công cụ quản lý công việc hiệu quả: Xây dựng một nền tảng trực tuyến giúp người dùng quản lý công việc, theo dõi tiến độ và phân công nhiệm vụ một cách dễ dàng, từ đó cải thiện năng suất và chất lượng công việc.
- Nâng cao khả năng tổ chức và quản lý thời gian: Giúp người dùng tối ưu hóa việc phân bổ thời gian và nguồn lực, hạn chế tình trạng công việc bị quên hoặc trì hoãn.
- Hỗ trợ người dùng có thể truy cập và quản lý công việc mọi lúc, mọi nơi thông qua hệ thống website, giúp làm việc từ xa hoặc phối hợp nhóm hiệu quả hơn.
- Đề tài giúp nghiên cứu tìm hiểu và áp dụng các công nghệ mới trong việc phát triển một hệ thống quản lý trực tuyến, từ đó tích lũy thêm kiến thức và kinh nghiệm thực tế trong lĩnh vực công nghệ thông tin.

† Phạm vi nghiên cứu:

- Khảo sát về nhu cầu quản lý công việc của doanh nghiệp dựa trên các nhóm, đối tượng như nhân viên, quản lý, giám đốc từ đó đưa ra đánh giá.
- Nghiên cứu qua Internet: đánh giá và đưa ra các ý kiến khách quan về vấn đề.
- Thực hiện phân tích các chức năng chính cần thiết cho hệ thống.
- Lựa chọn công nghệ phù hợp để phát triển cũng như các công cụ cần thiết để xây dựng website.

### **Ý nghĩa khoa học và thực tiễn của đề tài**

#### **Ý nghĩa khoa học**

- Góp phần phát triển lĩnh vực quản lý công việc: Đề tài là cơ hội để nghiên cứu và áp dụng các mô hình quản lý công việc và quản lý dự án phổ biến, kết hợp lý thuyết và thực tiễn để đưa ra các giải pháp tối ưu.

- **Ứng dụng công nghệ vào phát triển phần mềm quản lý:** Đề tài giúp áp dụng và mở rộng kiến thức về công nghệ lập trình web, quản lý cơ sở dữ liệu, xây dựng giao diện người dùng, bảo mật thông tin và xử lý dữ liệu, từ đó tích lũy thêm kinh nghiệm trong lĩnh vực phát triển phần mềm quản lý.

## **Ý nghĩa thực tiễn**

- Hỗ trợ người dùng quản lý công việc hiệu quả.
- Tăng năng suất làm việc.
- Tạo điều kiện làm việc từ xa.
- Góp phần quá trình chuyển đổi số.

Một số kết quả thực nghiệm cần đạt được:

- Website đạt yêu cầu quản lý công việc cho một doanh nghiệp.
- Website giải quyết nhu cầu về quản lý công việc.
- Giúp quá trình quản lý công việc dễ dàng, thuận tiện, trực quan và linh hoạt hơn.
- Giao diện thân thiện, dễ sử dụng.

## **Bố cục đề tài**

Ngoài các phần Mở đầu, Kết luận, Tài liệu tham khảo và Phụ lục, báo cáo đồ án được bố cục thành bốn chương chính sau:

- Chương 1: Tổng quan về cơ sở lý thuyết và công nghệ sử dụng  
Giới thiệu tóm tắt về lịch sử ra đời, mục đích sử dụng và ưu điểm của các công nghệ được sử dụng trong quá trình thực hiện đề tài.

Tiến hành các khâu khảo sát, phân tích các yêu cầu của hệ thống, xây dựng các biểu đồ ca sử dụng, đặc tả chi tiết các ca sử dụng, thiết kế cơ sở dữ liệu và giao diện cho hệ thống.

- Chương 3: Kết quả và kiểm thử  
Mô tả môi trường thử nghiệm và hình ảnh kết quả ứng dụng đã được cài đặt trên môi trường thử nghiệm, mô tả các thao tác trên màn hình.

Trình bày kế hoạch, nội dung kiểm thử và kết quả thực hiện kiểm thử.

## CHƯƠNG 1. TỔNG QUAN VỀ NỘI DUNG NGHIÊN CỨU

### 1.1. Mô tả bài toán

Bài toán đặt ra là xây dựng một hệ thống website quản lý công việc, nhằm giúp người dùng quản lý các nhiệm vụ và dự án một cách khoa học, hiệu quả. Các yêu cầu cụ thể của bài toán bao gồm:

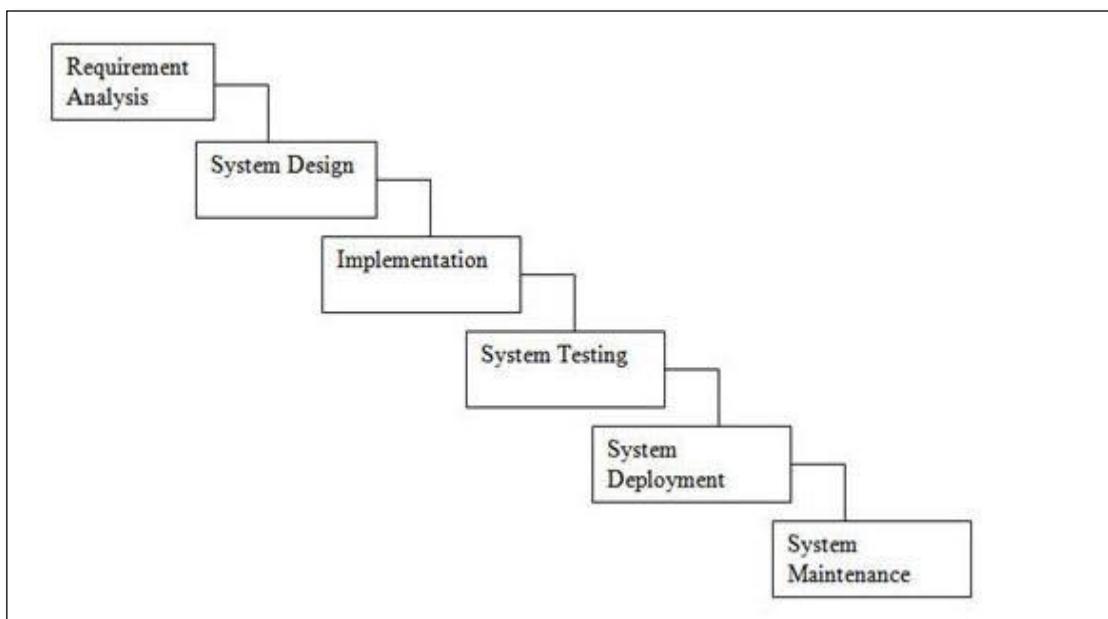
- Quản lý sprint:
  - Thêm, xóa, chỉnh sửa sprint: tạo ra sprint để quản lý các nhóm công việc theo giai đoạn.
- Quản lý công việc
  - Thêm, xóa, chỉnh sửa công việc: người dùng có thể tạo mới, xóa hoặc chỉnh sửa công việc. Mỗi công việc bao gồm các thuộc tính cơ bản như tên, mô tả, estimatetime.
  - Phân công công việc: người dùng có thể phân công công việc cho từng thành viên và theo dõi tiến độ của từng người.
- Theo dõi tiến độ công việc:
  - Trạng thái công việc: các công việc cần có các trạng thái như “To do”, “Inprogress”, “Ready for test”, “Done”, giúp người dùng theo dõi tiến độ một cách trực quan.
  - Cập nhật tiến độ: người dùng có thể cập nhật trạng thái và tiến độ hoàn thành công việc để nắm bắt thông tin kịp thời.
- Bảo mật và phân quyền người dùng:
  - Admin có thể thêm tài khoản cho người dùng để hoạt động và theo dõi công việc trong tổ chức.
  - Phân quyền người dùng: hệ thống có các mức phân quyền khác nhau ứng với các vị trí và hành động khác nhau.
  - Bảo mật: chỉ những người có quyền mới có thể chỉnh sửa một số thông tin.

Tóm lại, bài toán yêu cầu thiết kế và triển khai một hệ thống quản lý công việc trực tuyến với đầy đủ các tính năng lập kế hoạch, theo dõi, phân công và đánh giá công việc nhằm tối ưu hóa hiệu quả làm việc của cá nhân và nhóm.

## 1.2. Mô hình quy trình

Mô hình thác nước là một mô hình tuần tự, hoạt động phát triển phần mềm trong mô hình này được chia thành các giai đoạn khác nhau. Trong các giai đoạn gồm các mục tiêu/nhiệm vụ khác nhau.

Mô hình thác nước là giai đoạn đầu trong vòng đời phát triển phần mềm. Mô hình này được sử dụng rộng rãi, với mô hình này người phát triển cần phải thực hiện từng giai đoạn theo thứ tự một cách chỉnh chu và nghiêm ngặt bởi trong mô hình thác nước việc chuyển giai đoạn từ pha phát triển này sang pha khác sẽ diễn ra chỉ sau khi các pha trước đó đã kết thúc và hoàn toàn thành công, có nghĩa người phát triển sẽ không thể quay lui về pha trước đó hay nhảy vượt pha. Các giai đoạn lần lượt của mô hình thác nước được mô tả tại hình 1.1.



*Hình 1. 1. Các giai đoạn mô hình thác nước*

Mô hình Thác nước được sử dụng khi:

- Yêu cầu rõ ràng từ đầu và không cần cập nhật lại nhiều.
- Phù hợp với dự án, ứng dụng nhỏ.
- Không có hoặc ít có yêu cầu mà không hiểu hoặc không rõ ràng.

- Môi trường ổn định.
- Công nghệ được sử dụng đánh giá là ổn định.
- Nguồn lực con người đã sẵn sàng.

**Ưu điểm của việc sử dụng mô hình thác nước:**

- Đơn giản, dễ hiểu.
- Cân xong hoàn thành giai đoạn trước mới có thể đến giai đoạn sau nên dễ dàng cho việc bảo trì.
- Các tiêu chí đầu vào và đầu ra được xác định rõ ràng, do đó nó dễ dàng và có hệ thống để tiến hành chất lượng.

**Nhược điểm của việc sử dụng mô hình thác nước:**

- Không chấp nhận thay đổi yêu cầu cho nên không phù hợp với dự án có nhu cầu thay đổi thường xuyên.
- Khi có nhu cầu thay đổi ở giai đoạn trước thì khó có thể quay lại.
- Không phù hợp với dự án lớn.

Nếu yêu cầu là đơn giản, quy mô dự án nhỏ, mô hình thác nước sẽ mang lại kết quả tốt nhất. Cho nên sau khi xem xét thì mô hình thác nước này hoàn toàn phù hợp với bài toán đưa ra.

### **1.3. Công nghệ áp dụng**

#### **1.3.1. ReactJS**

**Khái niệm:**



*Hình 1. 2. Thư viện ReactJS*

ReactJS là một thư viện JavaScript mã nguồn mở được Facebook xây dựng và phát triển. ReactJS nổi bật với sự dễ dàng sử dụng và thuận tiện trong việc kết hợp với các thư viện javascript khác.

Trong ReactJS có khái niệm jsx, nó có cú pháp giống HTML để viết thành các thành phần của giao diện người dùng và các hành động xử lý sự kiện. Với jsx, ta có thể lồng các đoạn mã HTML vào trong javascript 1 cách thuần túy và nó sẽ được biên dịch để chạy được trên trình duyệt.

Đặc điểm nổi bật của React là nó sẽ so sánh sự thay đổi giá trị của lần render hiện tại và lần render trước đó để giảm thiểu ít nhất sự thay đổi trên DOM với công nghệ VIRTUAL DOM.

### **Hệ sinh thái của ReactJS được sử dụng trong bài làm:**

- Redux là một thư viện nổi bật trong việc quản lý dữ liệu trong một dự án React và được sử dụng rộng rãi. Do việc truyền dữ liệu trong React chỉ được thực hiện 1 chiều từ cha xuống con cho nên tránh việc truyền dữ liệu quá sâu gây mất hiệu năng và không cần thiết Redux đã cho phép ta thao tác với dữ liệu như 1 biến global để bất cứ đâu có thể truy xuất được biến đó, giúp cho việc thao tác với dữ liệu trở lên linh hoạt và dễ dàng hơn.
- React-router-dom là một thư viện giúp cho việc chuyển hướng giữa các trang trở lên linh hoạt. Với một SPA (Single page application) việc chuyển đổi giữa các trang thay vì phải tải lại cả trang thì ta chỉ cần thay đổi các thành phần để hiển thị điều này sẽ tốt cho trải nghiệm và tốc độ hiển thị sẽ nhanh hơn rất nhiều.
- Ant design là một thư viện UI giúp cho việc tổ chức thành phần một cách dễ dàng và nhanh chóng khi đã cung cấp cho người phát triển rất nhiều components và có thể tự điều chỉnh sao cho phù hợp với phong cách dự án.

### **Ưu điểm của ReactJS**

- Hiệu suất cao:
  - Sử dụng Virtual DOM để tối ưu hóa việc cập nhật giao diện.
  - Tăng tốc độ render và cải thiện hiệu suất.

- Component-Based Architecture:
  - Mã nguồn có thể tái sử dụng và dễ bảo trì.
  - Tạo và quản lý các component độc lập.
- Hệ sinh thái và cộng đồng phát triển mạnh:
  - Có một cộng đồng lớn, việc tìm kiếm tài liệu, thông tin trở lên nhanh chóng và dễ dàng.
  - Nhiều thư viện và công cụ hỗ trợ.
- Dễ học và sử dụng:
  - JSX giúp viết mã dễ đọc và dễ hiểu.
  - Cú pháp đơn giản và dễ tiếp cận với người đã làm quen với HTML và Javascript.
- Khả năng mở rộng và tích hợp:
  - Dễ dàng tích hợp với các công nghệ khác.
  - Khả năng mở rộng để phát triển các ứng dụng lớn và phức tạp.

### 1.3.2. NodeJs



*Hình 1. 3. NodeJS Image*

#### Node.js là gì?

Node.js là một nền tảng được xây dựng trên V8 JavaScript Engine. Node.js là một mã nguồn mở, được duy trì bởi cộng đồng lập trình viên trên toàn thế giới. Node.js không phụ thuộc vào bất kỳ hệ điều hành nào cho nên nó có thể chạy trên Windows, Linux, MacOS, điều này cho thấy sự linh hoạt cho các

nha phat trien khi phat trien ứng dung mà không cần thay đổi mã nguồn khi thay đổi nền tảng.

Là một môi trường thực thi Javascript. Các trình duyệt như Chrome, Firefox đã cung cấp môi trường để thực thi javascript thì ngoài trình duyệt Node.js đã cung cấp môi trường chạy trên máy chủ có thể thực thi ở bất kỳ môi trường máy tính nào khác.

Node.js được xây dựng trên V8, động cơ phát triển Javascript được phát triển bởi Google cho trình duyệt nên Node.js thực thi Javascript một cách nhanh và hiệu quả, nhanh chóng.

### **Node.js hoạt động như thế nào?**

Node.js hoạt động dựa trên một số nguyên tắc giúp cho việc xử lý các ứng dụng có nhiều hoạt động nhập xuất một cách hiệu quả mà không bị chặn. Với mô hình non-blocking I/O và event-driven, nghĩa là các hoạt động như đọc file, truy cập dữ liệu hoặc giao tiếp mạng được thực hiện mà không chặn tiến trình cho nên cho phép có nhiều yêu cầu cùng lúc mà không cần tạo nhiều luồng.

Đặc biệt trong Node.js là “Event loop”, đây là vòng lặp sự kiện ở đây Node.js tiếp tục lắng nghe sự kiện và thực hiện các hàm callback khi một sự kiện được kích hoạt. Với “Event loop” cho phép Node.js xử lý nhiều kết nối đồng thời mà không phải tạo ra nhiều luồng gây tốn chi phí quản lý.

#### **1.3.3. Hệ quản trị cơ sở dữ liệu MySQL**

##### **MySQL là gì?**



*Hình 1. 4. Hệ quản trị cơ sở dữ liệu MySQL*

MySQL là một hệ thống quản trị cơ sở dữ liệu mã nguồn mở (gọi tắt là RDBMS) được nhiều các nhà phát triển tin dùng. MySQL quản lý dữ liệu thông

qua các cơ sở dữ liệu, mỗi cơ sở dữ liệu có thể có nhiều bảng quan hệ chứa dữ liệu. Trong MySQL cũng có thể truy xuất dữ liệu qua các câu lệnh SQL.

MySQL là hệ quản trị cơ sở dữ liệu tốc độ cao, ổn định và dễ sử dụng, nó có thể hoạt động trên nhiều hệ điều hành cùng với nhiều hàm tiện ích mạnh mẽ. Bởi có tốc độ và tính bảo mật cao, MySQL thích hợp cho các ứng dụng có truy cập CSDL trên internet.

MySQL được sử dụng cho việc hỗ trợ Node.js, PHP,... là nơi lưu trữ dữ liệu trên các trang web.

### **Ưu điểm**

- Hiệu suất cao: MySQL có khả năng xử lý một lượng lớn dữ liệu với tốc độ nhanh và hiệu quả.
- Dễ sử dụng: Quá trình cài đặt và cấu hình MySQL khá đơn giản, và nó có giao diện thân thiện với người dùng.
- Mã nguồn mở: Là phần mềm mã nguồn mở, MySQL cho phép người dùng tùy chỉnh và tránh sự ràng buộc của nhà cung cấp.
- Khả năng mở rộng: MySQL có thể mở rộng linh hoạt để đáp ứng nhu cầu của các ứng dụng lớn.
- Bảo mật: MySQL cung cấp nhiều tính năng bảo mật để bảo vệ dữ liệu.
- Hỗ trợ cộng đồng: Có một cộng đồng lớn và năng động hỗ trợ, giúp người dùng dễ dàng tìm kiếm giải pháp cho các vấn đề gặp phải.

### **Nhược điểm**

- Vấn đề về hiệu suất với dữ liệu lớn: Mặc dù MySQL có hiệu suất cao, nhưng khi xử lý các tập dữ liệu cực lớn, nó có thể gặp phải một số vấn đề về hiệu suất.
- Hạn chế về tính năng: so với PostgreSQL hay Oracle, MySQL có thể thiếu một số tính năng nâng cao.

## CHƯƠNG 2. KHẢO SÁT VÀ PHÂN TÍCH THIẾT KẾ HỆ THỐNG

### 2.1. Khảo sát bài toán

Để linh hoạt trong quản lý, theo dõi và phân công công việc một cách trực quan và hiệu quả thì việc quản lý trên một nền tảng chung để theo dõi và hiển thị một cách chi tiết là rất cần thiết. Qua quá trình tham khảo ý kiến và khảo sát các công nghệ có chung đặc điểm quản lý công việc, dưới đây là kết quả chi tiết về các tính năng cần thiết:

- Tạo kế hoạch, đầu việc và phân công nhiệm vụ: Bao gồm việc tạo sprint và tạo / quản lý đầu việc trong sprint.
- Quản lý tiến độ công việc: Qua các trạng thái khác nhau và đánh dấu công việc đã hoàn thành.
- Thông báo và nhắc nhở: Thông báo đầy nhắc nhở giúp người dùng không bỏ lỡ các công việc hoặc đến hạn.
- Phân quyền và bảo mật: đảm bảo tính bảo mật, phân rõ các quyền truy cập khác nhau.

Không những vậy, ngoài một số chức năng cơ bản phải có ở trên hệ thống quản lý công việc cần phải dễ sử dụng. Từ đó để tối ưu và đơn giản hóa tránh những thao tác phức tạp em tập trung xây dựng những tính năng cơ bản và tối ưu về mặt trải nghiệm người dùng.

### **Yêu cầu chức năng của hệ thống**

Đây là một website quản lý công việc sẽ được phân ra 2 đối tượng chính là admin và người dùng. Có các chức năng sau:

#### **Người dùng:**

- Đăng nhập/đăng xuất.
- Tạo/chỉnh sửa/kết thúc sprint.
- Tạo/chỉnh sửa/xóa task.
- Comment task.
- Nhận thông báo.
- Filter/search task theo tên task, người được phân công.

- Chính sửa thông tin người dùng.

**Manager:**

- Đăng nhập/đăng xuất
- Quản lý project.

**Admin:**

- Đăng nhập/đăng xuất
- Quản lý tài khoản: tạo/chỉnh sửa/vô hiệu hóa.
- Quản lý project.
- Và toàn quyền thao tác với hệ thống.

**Yêu cầu phi chức năng của hệ thống**

- Website phải được trình bày sao cho dễ hiểu, giao diện dễ dùng và làm sao cho người dùng dễ thao tác và làm quen.
- Đồng thời Website còn phải có tính dễ nâng cấp, bảo trì, sửa chữa khi cần bổ sung, cập nhật những tính năng mới

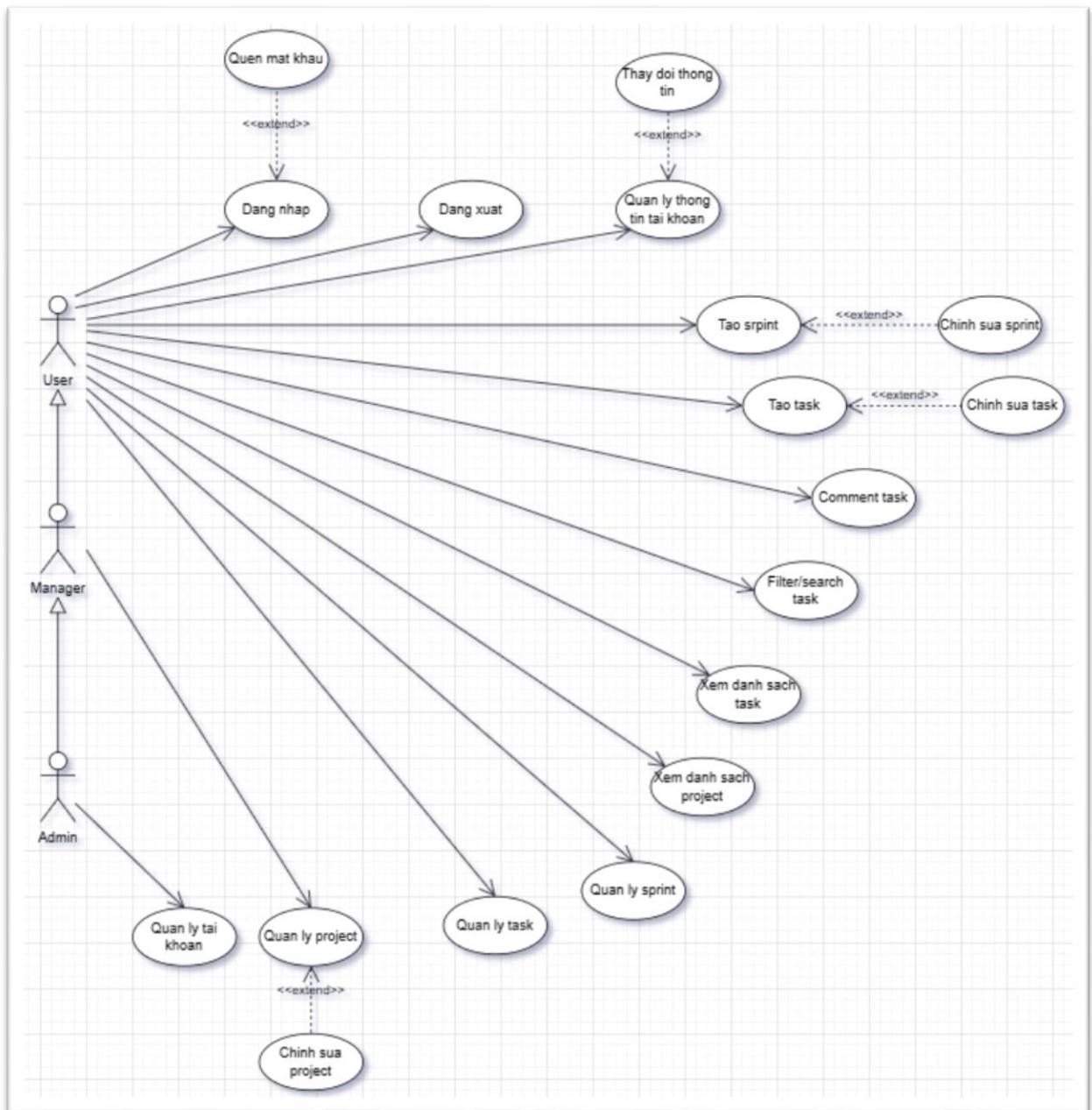
## 2.2. Phân tích và thiết kế hệ thống

**- Mô hình hóa chức năng:**

Bảng 2. 1. Mô hình hóa chức năng

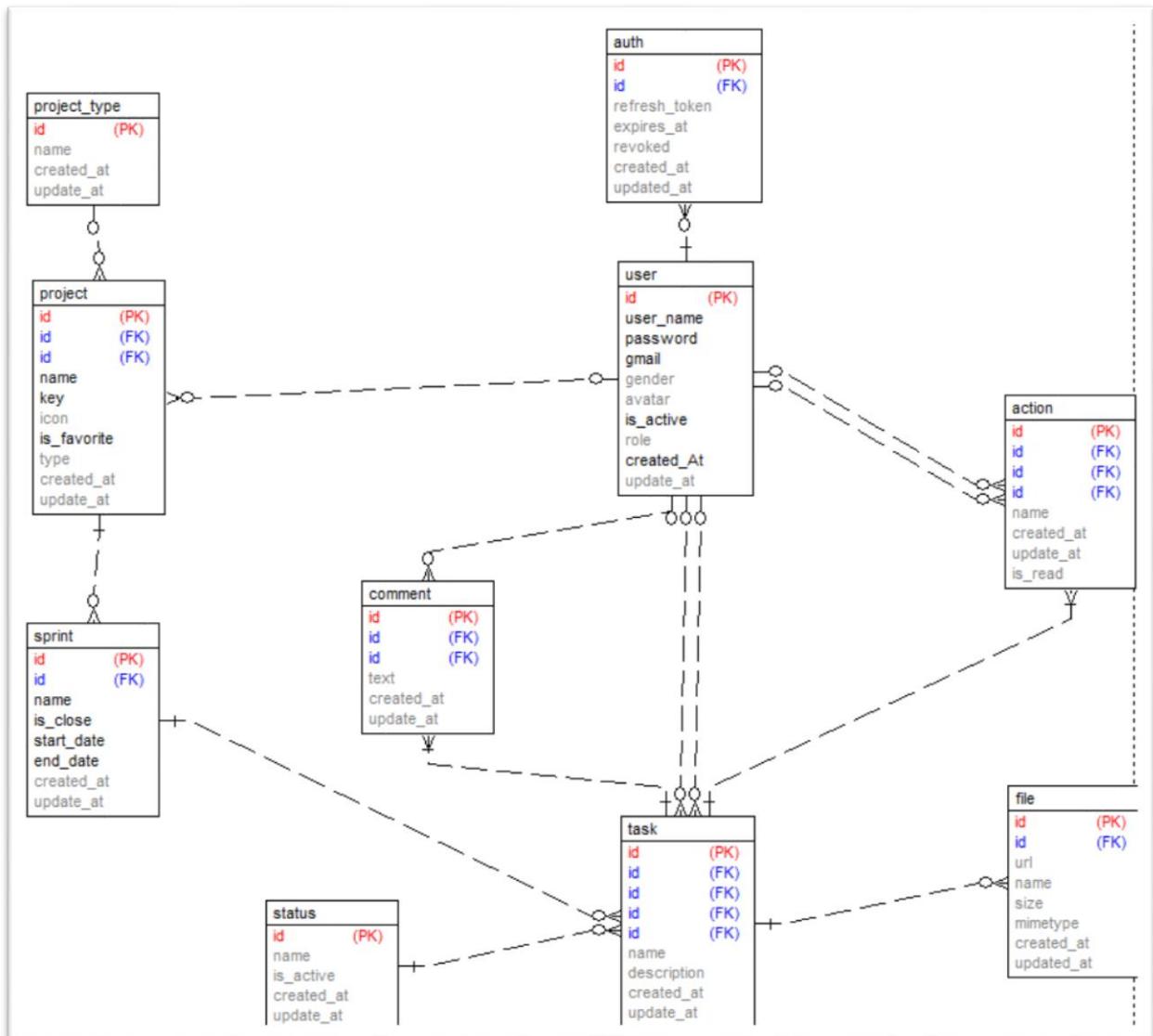
STT	Tên Actor	Mô tả
1	Admin (Quản trị)	Admin có toàn quyền tương tác với hệ thống, quản trị hệ thống, tạo project
2	Manager	Quản lý project.
3	User (Nhân viên)	User có thể đăng nhập, tạo và chỉnh sửa sprint/task, comment task, filter/search task theo tên hoặc người phân công.

### 2.2.1. Biểu đồ Use case tổng quan



Hình 2. 1. Biểu đồ usecase tổng quan của hệ thống

## 2.2.2. Biểu đồ lớp của hệ thống



Hình 2. 2. Biểu đồ lớp của hệ thống

## 2.2.3. Đặc tả các use case

### 2.2.3.1. Use case Thêm tài khoản (Admin)

Bảng 2. 2. Use case thêm tài khoản

Mô tả	Use case này cho phép admin thêm tài khoản người dùng vào hệ thống để thao tác trên website
Người thực hiện	admin

Tiền điều kiện	Role là admin
Luồng cơ bản	<p>1. Use case này bắt đầu khi admin click vào nút “Add user”.</p> <p>2. Hệ thống hiển thị màn hình nhập các thông tin để thêm người dùng.</p> <p>3. Admin nhập các thông tin theo yêu cầu của hệ thống và click vào nút “Thêm”. Hệ thống tự sinh tài khoản và mật khẩu đồng thời sinh gửi xác nhận vào email đã được nhập.</p> <p>4. Hệ thống lưu thông tin user vào bảng “user” và hiển thị lên màn hình thông báo đăng ký thành công. Use case kết thúc.</p>
Luồng rẽ nhánh	<p>1. Nếu nhập không hợp lệ một trong các trường dữ liệu thì hệ thống hiển thị thông báo lỗi, admin có thể nhập lại trường dữ liệu đó hoặc thoát khỏi màn hình thêm người dùng.</p> <p>2. Tại bước 3 trong luồng cơ bản nếu email đã được đăng ký thì hệ thống sẽ thông báo lỗi Email đã tồn tại.</p> <p>3. Không kết nối được cơ sở dữ liệu: tại bất kỳ thời điểm nào trong quá trình thực hiện use case nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị một thông báo lỗi và use case kết thúc.</p>
Hậu điều kiện	Nếu Use case thành công, hệ thống sẽ lưu dữ liệu mới của người dùng vào bảng “user”. Ngược lại hiển thị màn hình thông báo lỗi.

### 2.2.3.2. Use case Đăng nhập

Bảng 2. 3. Mô tả chi tiết use case Đăng nhập.

Mô tả	Use case này cho phép người dùng đăng nhập để sử dụng các chức năng của hệ thống.
Người thực hiện	người dùng
Tiền điều kiện	Đã có tài khoản của hệ thống
Luồng cơ bản	<p>1. Use case này bắt đầu khi người dùng truy cập trang web. Hệ thống hiển thị một màn hình đăng nhập.</p> <p>2. Người dùng nhập tài khoản và mật khẩu và nhấn “Đăng nhập”. Hệ thống kiểm tra thông tin chi tiết trong bảng “user” và hiển thị màn hình trang chủ lên màn hình. Use case kết thúc.</p>
Luồng rẽ nhánh	<p>1.Người dùng nhập sai thông tin tài khoản hoặc mật khẩu thì hệ thống hiển thị thông báo lỗi, khách hàng có thể nhập lại thông tin tài khoản hoặc thoát khỏi màn hình đăng nhập.</p> <p>2.Không kết nối được cơ sở dữ liệu: tại bất kỳ thời điểm nào trong quá trình thực hiện use case nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị một thông báo lỗi và use case kết thúc.</p>
Hậu điều kiện	Cho phép sử dụng các chức năng của hệ thống theo quyền của tài khoản đã đăng ký.

### 2.2.3.3. Use case Quên mật khẩu

Bảng 2. 4. Mô tả chi tiết use case Đăng nhập

Mô tả	Use case này cho phép người dùng lấy lại mật khẩu khi đã mất.
Người thực hiện	Người dùng
Tiền điều kiện	Không
Luồng cơ bản	<p>1.Use case này bắt đầu khi người dùng click vào nút “Quên mật khẩu” ở màn hình Đăng nhập.</p> <p>2.Hệ thống hiển thị màn hình quên mật khẩu.</p> <p>3.Người dùng nhập các thông tin theo yêu cầu của hệ thống và click vào nút “Đặt lại mật khẩu. Hệ thống tự sinh mật khẩu và gửi về email đã đăng ký.</p> <p>4.Hệ thống cập nhật thông tin mật khẩu mới của người dùng vào bảng “User” và hiển thị lên màn hình thông báo đặt lại mật khẩu thành công. Use case kết thúc.</p>

Luồng rẽ nhánh	<p>1.Nếu nhập không hợp lệ một trong các trường dữ liệu thì hệ thống hiển thị thông báo lỗi, người dùng có thể nhập lại trường dữ liệu đó hoặc thoát khỏi màn hình đăng ký.</p> <p>2.Tại bước 3 trong luồng cơ bản nếu email chưa được đăng ký thì hệ thống sẽ thông báo lỗi Email không tồn tại.</p> <p>3. Không kết nối được cơ sở dữ liệu: tại bất kỳ thời điểm nào trong quá trình thực hiện use case nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị một thông báo lỗi và use case kết thúc.</p>
Hậu điều kiện	<p>Nếu Use case thành công, hệ thống sẽ lưu dữ liệu mới của người dùng vào bảng “User”. Ngược lại hiển thị màn hình thông báo lỗi.</p>

#### 2.2.3.4. Use case Cập nhật thông tin tài khoản

Bảng 2. 5. Mô tả chi tiết use case Cập nhật thông tin tài khoản

Mô tả	Use case này cho phép người dùng cập nhật thông tin tài khoản
Người thực hiện	Người dùng
Tiền điều kiện	Đã đăng nhập
Luồng cơ bản	<p>1.Use case này bắt đầu khi khách hàng click “Settings”, hệ thống lấy thông tin người dùng từ bảng user và hiển thị lên màn hình.</p> <p>2.Người dùng điền thông tin chỉnh sửa và click “Save”. Hệ thống sẽ cập nhật thông tin người dùng vào bảng user và thông báo cập nhật thành công. Use case kết thúc</p>

Luồng rẽ nhánh	<p>1.Nếu người dùng nhập thông tin không đúng định dạng màn hình sẽ hiển thị lỗi và không cho phép cập nhật. Người dùng có thể chỉnh sửa lại thông tin hoặc rời khỏi màn hình cập nhật thông tin.</p> <p>2.Không kết nối được cơ sở dữ liệu: tại bất kỳ thời điểm nào trong quá trình thực hiện use case nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị một thông báo lỗi và use case kết thúc.</p>
Hậu điều kiện	Thông tin được cập nhật và update trong bảng “user”.

#### 2.2.3.5. Use case Quản lý tài khoản (Admin)

Bảng 2. 6. Mô tả chi tiết use case Quản lý tài khoản (Admin)

Mô tả	Use case này cho phép admin quản lý tài khoản người dùng có trong hệ thống
Người thực hiện	Admin
Tiền điều kiện	Role là admin và đã đăng nhập
Luồng cơ bản	<p>1.Use case này bắt đầu khi admin click “Accounts manager”. Hệ thống sẽ lấy tất cả thông tin các user có trong bảng “user” và hiển thị lên màn hình.</p> <p>2.Tại màn hình quản lý tài khoản, admin có thể chỉnh sửa thông tin user bằng cách double click vào ô thông tin tương ứng và chỉnh sửa, sau khi blur khỏi ô thông tin hệ thống sẽ cập nhật thông tin mới đã được chỉnh sửa.</p> <p>3.Use case kết thúc</p>

Luồng rẽ nhánh	<p>1.Trong quá trình cập nhật thông tin nếu không nhập đúng định dạng hệ thống sẽ không cập nhật thông tin và thông báo lỗi + hiển thị lại thông tin cũ.</p> <p>2.Không kết nối được cơ sở dữ liệu: tại bất kỳ thời điểm nào trong quá trình thực hiện use case nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị một thông báo lỗi và use case kết thúc.</p>
Hậu điều kiện	Thông tin được cập nhật trong bảng “user”

### 2.2.3.6. Use case Tạo project

Bảng 2.7. Mô tả chi tiết use case Tạo project

Mô tả	Use case này cho người dùng tạo project
Người thực hiện	Admin/Manager
Tiền điều kiện	Đã đăng nhập
Luồng cơ bản	<p>1.Use case này bắt đầu khi người dùng click “Create project”, màn hình sẽ hiển thị form người dùng sẽ nhập thông tin cần thiết và click “Create”</p> <p>2. Hệ thống sẽ tạo thêm project và lưu vào bảng “project” với thông tin đã được nhập và hiển thị thông báo thành công. Use case kết thúc</p>
Luồng rẽ nhánh	<p>1. Tại bước 1 nếu người dùng nhập thông tin không đúng định dạng hoặc thiếu hệ thống sẽ không cho phép người dùng tạo project và thông báo lý do. Người dùng có thể sửa nội dung đã điền hoặc rời khỏi màn tạo project.</p>

	2.Không kết nối được cơ sở dữ liệu: tại bất kỳ thời điểm nào trong quá trình thực hiện use case nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị một thông báo lỗi và use case kết thúc.
Hậu điều kiện	Project mới sẽ được tạo và lưu trong bảng “project”

#### 2.2.3.7. Use case Chính sửa project

Bảng 2.8. Mô tả chi tiết use case Chính sửa project

Mô tả	Use case này cho phép người dùng chỉnh sửa thông tin của project.
Người thực hiện	Người dùng
Tiền điều kiện	Là lead của project/admin
Luồng cơ bản	<p>1.Use case này bắt đầu khi người dùng click vào vào option setting, hệ thống sẽ lấy thông tin của project và hiển thị lên màn hình.</p> <p>2.Người dùng sẽ chỉnh sửa các trường cần thiết và click “save”. Hệ thống sẽ lưu lại thông tin project và bảng “project” và thông báo thành công. Use case kết thúc.</p>
Luồng rẽ nhánh	<p>1. Nếu người dùng nhập sai hoặc bỏ trống trường bắt buộc bắt kỳ thì hệ thống sẽ cảnh báo và không cho phép lưu. Người dùng có thể thoát hoặc chỉnh sửa lại thông tin.</p> <p>2.Không kết nối được cơ sở dữ liệu: tại bất kỳ thời điểm nào trong quá trình thực hiện use case nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị một thông báo lỗi và use case kết thúc.</p>
Hậu điều kiện	Project được update trong bảng “project”

### 2.2.3.8. Use case Xóa project

*Bảng 2. 9. Mô tả chi tiết use case Xóa project*

Mô tả	Use case này cho phép người dùng xóa project
Người thực hiện	Người dùng
Tiền điều kiện	Lead/admin
Luồng cơ bản	<p>1. Use case này bắt đầu khi người dùng click “Delete”, màn hình sẽ hiển thị màn hình xác nhận và yêu cầu nhập tên project để xóa. Người dùng nhập tên project và click nút “Delete”.</p> <p>2. Hệ thống sẽ xóa thông tin của project và các task, sprint thuộc project. Use case kết thúc</p>
Luồng rẽ nhánh	<p>1. Nếu không nhập đúng tên sẽ không xóa được project/bạn click “Cancel” thì thao tác xóa project sẽ được hủy bỏ.</p> <p>2. Không kết nối được cơ sở dữ liệu: tại bất kỳ thời điểm nào trong quá trình thực hiện use case nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị một thông báo lỗi và use case kết thúc.</p>
Hậu điều kiện	Thông tin project và các task, sprint trong project sẽ bị xóa khỏi CSDL

### 2.2.3.9. Use case Tạo sprint

Bảng 2. 10. Mô tả chi tiết use case tạo sprint

Mô tả	Use case này cho phép người dùng tạo sprint trong 1 project
Người thực hiện	User
Tiền điều kiện	Đã đăng nhập, có trong project
Luồng sự kiện chính	1. Use case này bắt đầu khi user click “Create sprint” 2. Màn hình hiển thị form nhập thông tin, người dùng nhập đầy đủ thông tin theo đúng định dạng và click “Create”. Use case kết thúc.
Luồng rẽ nhánh	1. Nếu người dùng nhập thiếu 1 trường thông tin hoặc sai định dạng hệ thống sẽ báo lỗi, lúc này người dùng sẽ sửa thông tin hoặc thoát khỏi màn tạo sprint. 2. Không kết nối được với cơ sở dữ liệu: tại bất kỳ thời điểm nào trong quá trình thực hiện use case nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị một thông báo lỗi và use case kết thúc.
Hậu điều kiện	Sprint được tạo và lưu trong bảng “sprint”.

### 2.2.3.10. Use case Chính sửa sprint

Bảng 2. 11. Mô tả chi tiết use case Chính sửa sprint

Mô tả	Use case này cho phép khách hàng hủy đơn hàng đã đặt
Người thực hiện	Khách hàng
Tiền điều kiện	Khách hàng đã đặt hàng trên hệ thống và người quản trị chưa duyệt đơn.

Luồng cơ bản	<p>1.Use case này bắt đầu khi người dùng click vào vào option setting, hệ thống sẽ lấy thông tin của sprint và hiển thị lên màn hình.</p> <p>2.Người dùng sẽ chỉnh sửa các trường cần thiết và click “save”. Hệ thống sẽ lưu lại thông tin sprint và bảng “sprint” và thông báo thành công. Use case kết thúc.</p>
Luồng rẽ nhánh	<p>1.Nếu người dùng nhập thiếu 1 trường thông tin hoặc sai định dạng hệ thống sẽ báo lỗi, lúc này người dùng sẽ sửa thông tin hoặc thoát khỏi màn tạo sprint.</p> <p>2.Không kết nối được với cơ sở dữ liệu: tại bất kỳ thời điểm nào trong quá trình thực hiện use case nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị một thông báo lỗi và use case kết thúc</p>
Hậu điều kiện	Không

### 2.2.3.11. Use case Kết thúc sprint

Bảng 2. 12. Mô tả chi tiết use case Kết thúc sprint

Mô tả	Use case này cho phép người dùng xóa sprint
Người thực hiện	Người dùng
Tiền điều kiện	Đã đăng nhập hệ thống
Luồng cơ bản	<p>1.Use case này bắt đầu khi người dùng click “Complete”, màn hình sẽ hiển thị màn hình xác nhận kết thúc sprint. Người dùng click nút “Accept”.</p> <p>2.Hệ thống sẽ cập nhật trạng thái của sprint và các task thuộc sprint. Use case kết thúc</p>

Luồng rẽ nhánh	1.Không kết nối được cơ sở dữ liệu: tại bất kỳ thời điểm nào trong quá trình thực hiện use case nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị một thông báo lỗi và use case kết thúc.
Hậu điều kiện	Trạng thái của sprint thay đổi

### 2.2.3.12. Use case Tạo task

Bảng 2. 13. Mô tả chi tiết use case Tạo task

Mô tả	Use case này cho người dùng tạo task
Người thực hiện	Người dùng
Tiền điều kiện	Đã đăng nhập, đã tạo sprint
Luồng cơ bản	<p>1.Use case này bắt đầu khi người dùng click “Create task”, màn hình sẽ hiển thị form và người dùng sẽ nhập các trường thông tin cần thiết và click “Create”.</p> <p>2.Hệ thống sẽ tạo task với thông tin người dùng đã nhập và lưu vào bảng “task” và thông báo thành công. Use case kết thúc</p>
Luồng rẽ nhánh	<p>1.Nếu người dùng nhập sai thông tin hoặc thiếu thông tin hệ thống sẽ không cho phép tạo task và hiển thị thông báo. Người dùng có thể hủy hoặc sửa thông tin để có thể tạo task tiếp.</p> <p>2.Không kết nối được cơ sở dữ liệu: tại bất kỳ thời điểm nào trong quá trình thực hiện use case nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị một thông báo lỗi và use case kết thúc.</p>
Hậu điều kiện	Tạo task thành công và task mới được lưu vào CSDL

### 2.2.3.13. Use case Xem chi tiết/Chỉnh sửa task

Bảng 2. 14. Mô tả chi tiết use case Xem chi tiết/Chỉnh sửa task

Mô tả	Use case này cho phép người dùng xem chi tiết/chỉnh sửa task
Người thực hiện	Người dùng
Tiền điều kiện	Người dùng đã đăng nhập hệ thống
Luồng cơ bản	<p>1. Use case này bắt đầu khi người dùng click vào 1 task. Hệ thống sẽ lấy thông tin đầy đủ của task và hiển thị lên màn hình.</p> <p>2. Tại màn hình chi tiết người dùng có thể chỉnh sửa các thông tin của task bằng cách chọn các thông tin được hiển thị lên</p> <p>3. Nếu có bất kỳ thay đổi nào hệ thống sẽ cập nhật thông tin lên CSDL. Use case kết thúc</p>
Luồng rẽ nhánh	<p>1. Tại bước 2 nếu người dùng chọn lựa chọn hoặc nhập thông tin không hợp lệ hệ thống sẽ tự động trở về trạng thái cũ của task.</p> <p>2. Không kết nối được cơ sở dữ liệu: tại bất kỳ thời điểm nào trong quá trình thực hiện use case nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị một thông báo lỗi và use case kết thúc.</p>
Hậu điều kiện	Nếu có chỉnh sửa, thông tin mới của task sẽ được cập nhật trong CSDL

#### 2.2.3.14. Use case search/filter Task

Bảng 2. 15. Mô tả chi tiết use case search/filter Task

Mô tả	Use case này cho phép người dùng tìm kiếm hoặc lọc task theo điều kiện xác định
Người thực hiện	Người dùng
Tiền điều kiện	Đã đăng nhập hệ thống
Luồng cơ bản	1. Use case bắt đầu khi người dùng nhập thông tin vào ô search/filter theo option, lúc này hệ thống sẽ dựa vào key seach để lấy dữ liệu trong CSDL và trả ra màn hình kết quả tương ứng. Use case kết thúc
Luồng rẽ nhánh	1.Không kết nối được cơ sở dữ liệu: tại bất kỳ thời điểm nào trong quá trình thực hiện use case nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị một thông báo lỗi và use case kết thúc.

Hậu điều kiện	Nếu tìm thấy kết quả màn hình sẽ hiển thị, còn không sẽ thông báo không có kết quả phù hợp
---------------	--

### 2.2.3.15. Use case Xóa task

Bảng 2. 16. Mô tả chi tiết use case Xóa task

Mô tả	Use case này cho phép người dùng xóa task
Tác nhân	Người dùng
Tiền điều kiện	Đã đăng nhập hệ thống
Luồng cơ bản	<p>1. Use case này bắt đầu khi người dùng click “Delete”. Màn hình sẽ hiển thị tooltip xác nhận xóa, người dùng click “Accept”.</p> <p>2. Hệ thống sẽ xóa task khỏi hệ thống và hiển thị thông báo. Use case kết thúc</p>
Luồng rẽ nhánh	<p>1. Nếu người dùng trong quá trình xóa chọn “Cancel”, quá trình xóa task sẽ bị hủy bỏ</p> <p>2. Không kết nối được cơ sở dữ liệu: tại bất kỳ thời điểm nào trong quá trình thực hiện use case nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị một thông báo lỗi và use case kết thúc.</p>
Hậu điều kiện	Nếu xóa task thành công, task sẽ bị xóa khỏi hệ thống và giao diện sẽ không hiển thị task bị xóa.

### 2.2.3.16. Use case Comment task

Bảng 2. 17. Mô tả chi tiết use case Comment task

Mô tả	Use case này cho phép người dùng bình luận trong task
Tác nhân	Người dùng
Tiền điều kiện	Đã đăng nhập hệ thống
Luồng cơ bản	<p>1. Use case này bắt đầu khi người dùng click “Add comment” tại màn hình chi tiết task. Người dùng sẽ nhập nội dung comment và click “Save”</p> <p>2. Hệ thống sẽ tạo comment với thông tin nhập từ người dùng và lưu vào bảng “comment”. Use case kết thúc</p>
Luồng rẽ nhánh	<p>1. Nếu người dùng trong quá trình nhập hủy bỏ hoặc để trống hệ thống sẽ không bị thay đổi gì.</p> <p>2. Không kết nối được cơ sở dữ liệu: tại bất kỳ thời điểm nào trong quá trình thực hiện use case nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị một thông báo lỗi và use case kết thúc.</p>
Hậu điều kiện	Nếu thêm comment thành công hệ thống sẽ lưu vào bảng “comment” và hiển thị thông báo

### 2.2.3.17. Use case Chính sửa comment

Bảng 2. 18. Mô tả chi tiết use case Chính sửa Comment

Mô tả	Use case này cho phép người dùng chỉnh sửa bình luận trong task
-------	---

Tác nhân	Người dùng
Tiền điều kiện	Đã đăng nhập hệ thống
Luồng cơ bản	<p>1. Use case này bắt đầu khi người dùng click “Edit comment” tại màn hình chi tiết task. Người dùng sẽ nhập nội dung comment và click “Save”</p> <p>2. Hệ thống sẽ cập nhật comment với thông tin nhập từ người dùng và lưu vào bảng “comment”. Use case kết thúc</p>
Luồng rẽ nhánh	<p>1. Nếu người dùng trong quá trình nhập hủy bỏ hoặc để trống hệ thống sẽ không bị thay đổi gì.</p> <p>2. Không kết nối được cơ sở dữ liệu: tại bất kỳ thời điểm nào trong quá trình thực hiện use case nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị một thông báo lỗi và use case kết thúc.</p>
Hậu điều kiện	Nếu edit comment thành công hệ thống sẽ lưu vào bảng “comment” và hiển thị thông báo

### 2.2.3.18. Use case Xóa comment

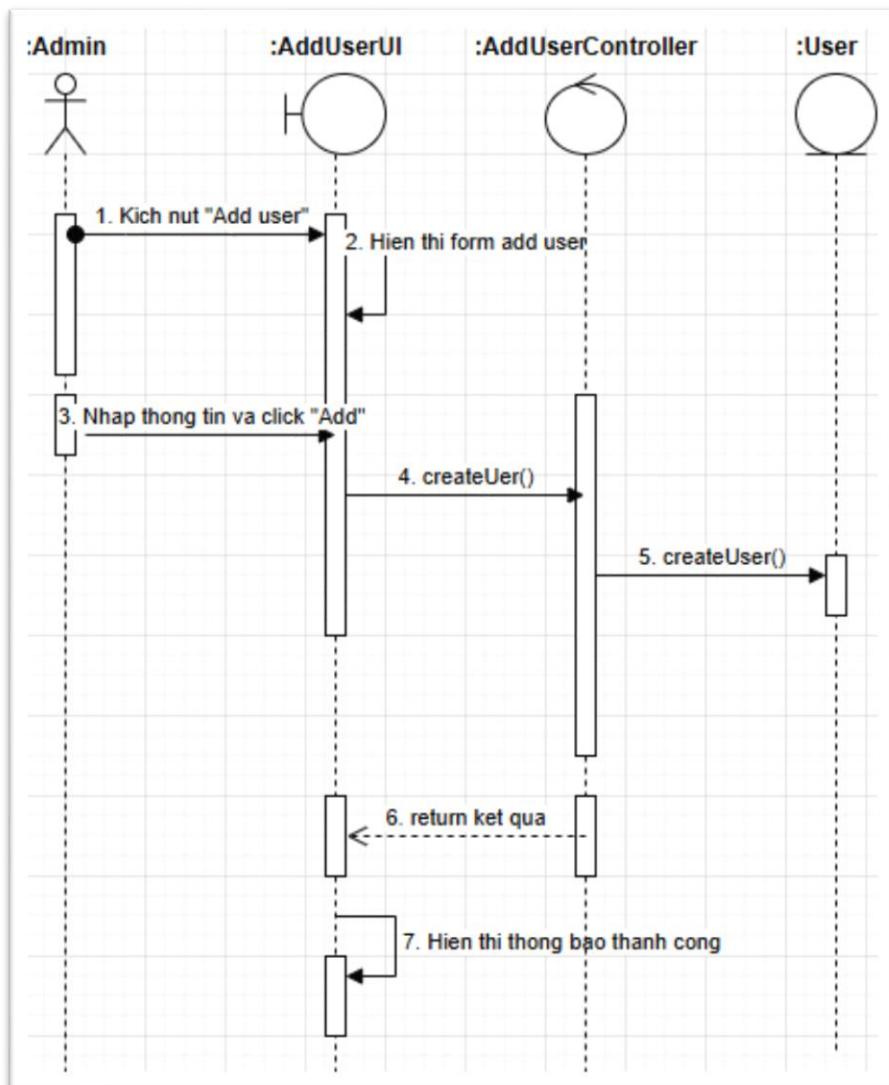
Bảng 2. 19. Mô tả chi tiết use case Xóa Comment.

Mô tả	Use case này cho phép người dùng xóa bình luận trong task
Tác nhân	Người dùng
Tiền điều kiện	Đã đăng nhập hệ thống

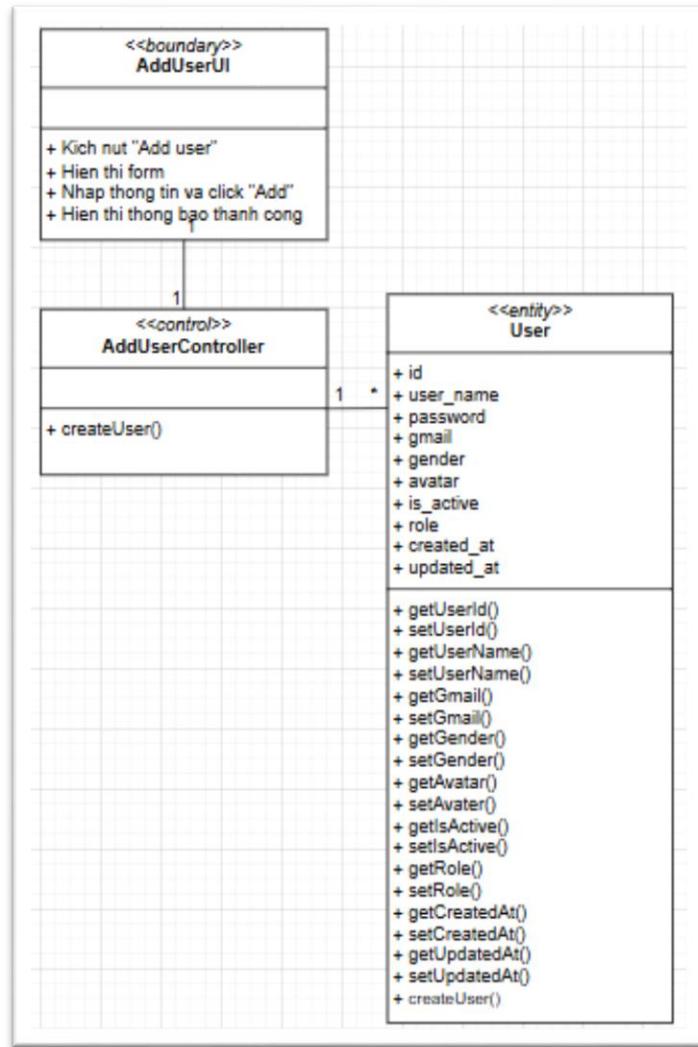
Luồng cơ bản	<p>1. Use case này bắt đầu khi người dùng click “Delete comment” tại khu vực comment. Màn hình sẽ hiển thị xác nhận. Người dùng click “Accept”</p> <p>2. Hệ thống sẽ xóa comment khỏi bảng “comment”. Use case kết thúc</p>
Luồng rẽ nhánh	<p>1. Nếu người dùng trong quá trình xác nhận xóa hủy bỏ hệ thống sẽ không bị thay đổi gì.</p> <p>2. Không kết nối được cơ sở dữ liệu: tại bất kỳ thời điểm nào trong quá trình thực hiện use case nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị một thông báo lỗi và use case kết thúc.</p>
Hậu điều kiện	Nếu xóa thành công, comment đó sẽ bị xóa trong CSDL

## 2.2.4. Hiện thực hóa use case

### 2.2.4.1. Phân tích use case Thêm tài khoản (Admin)

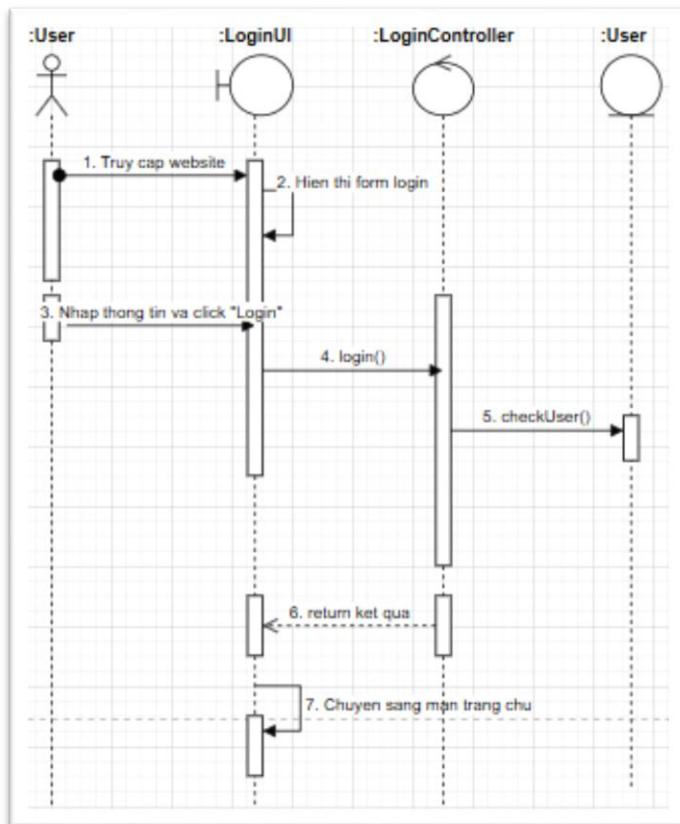


Hình 2. 3. Biểu đồ tuần tự use case Thêm tài khoản

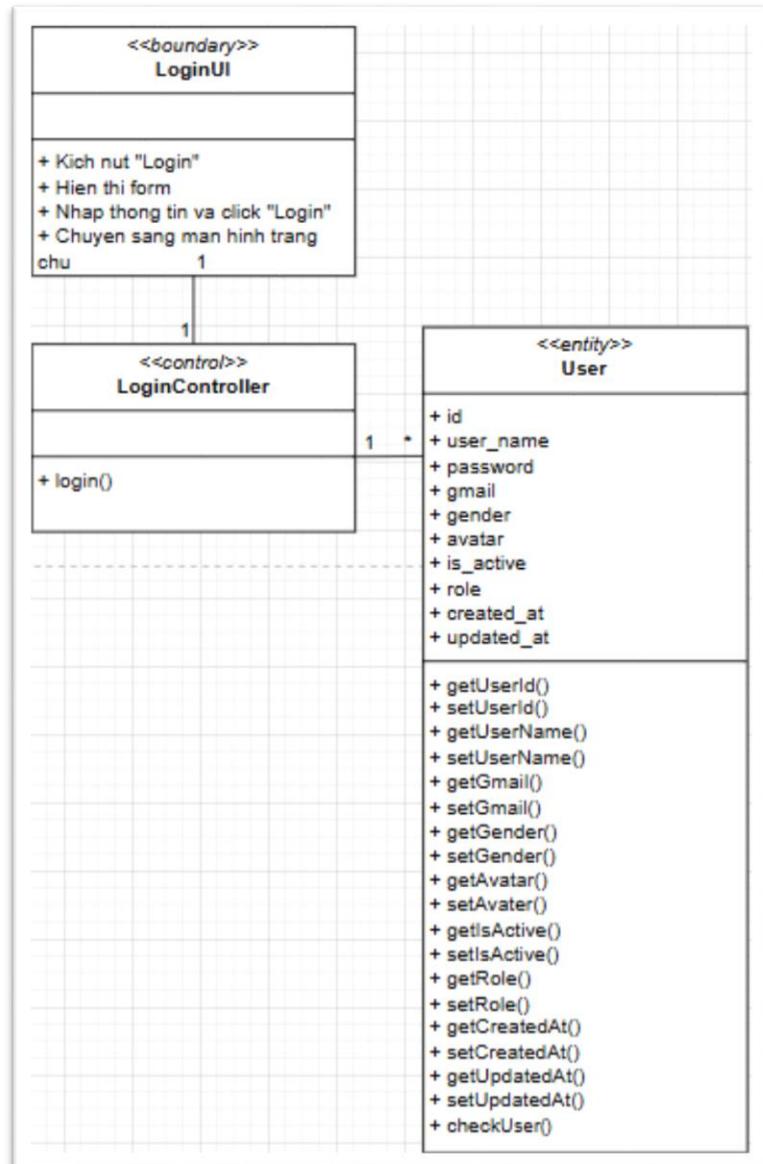


Hình 2. 4. VOPC use case Thêm tài khoản

#### 2.2.4.2. Phân tích use case Đăng nhập

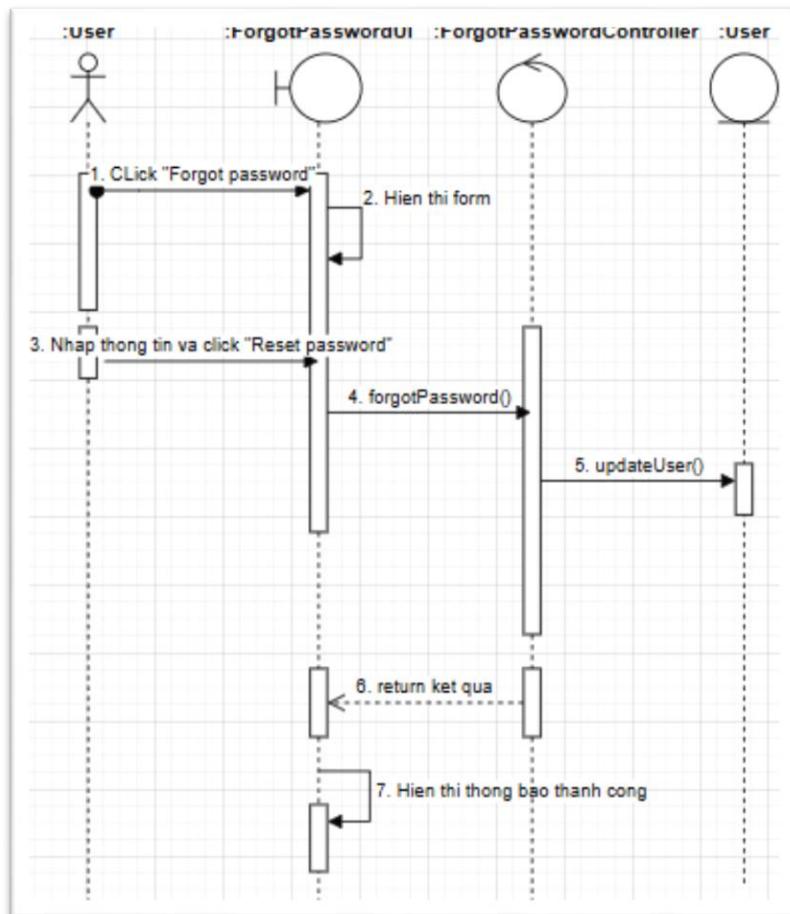


Hình 2. 5. Biểu đồ tuần tự use case Đăng nhập

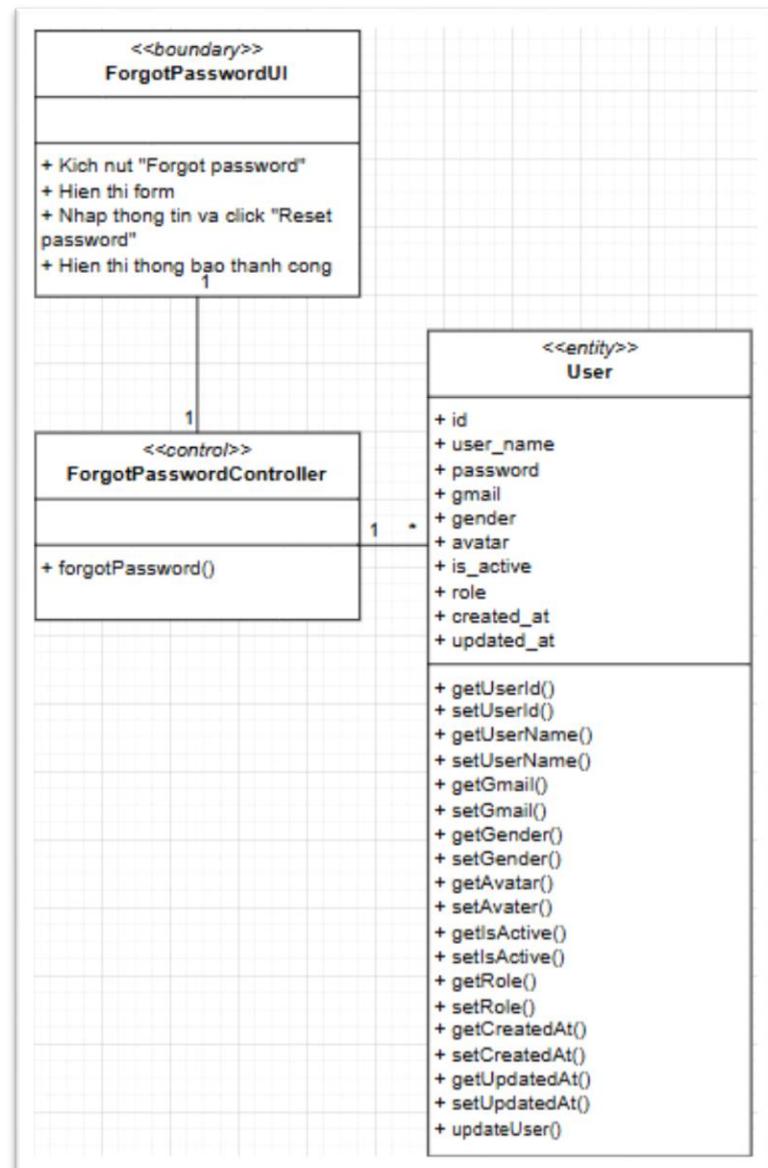


Hình 2. 6. VOPC use case Đăng nhập

### 2.2.4.3. Phân tích use case Quên mật khẩu

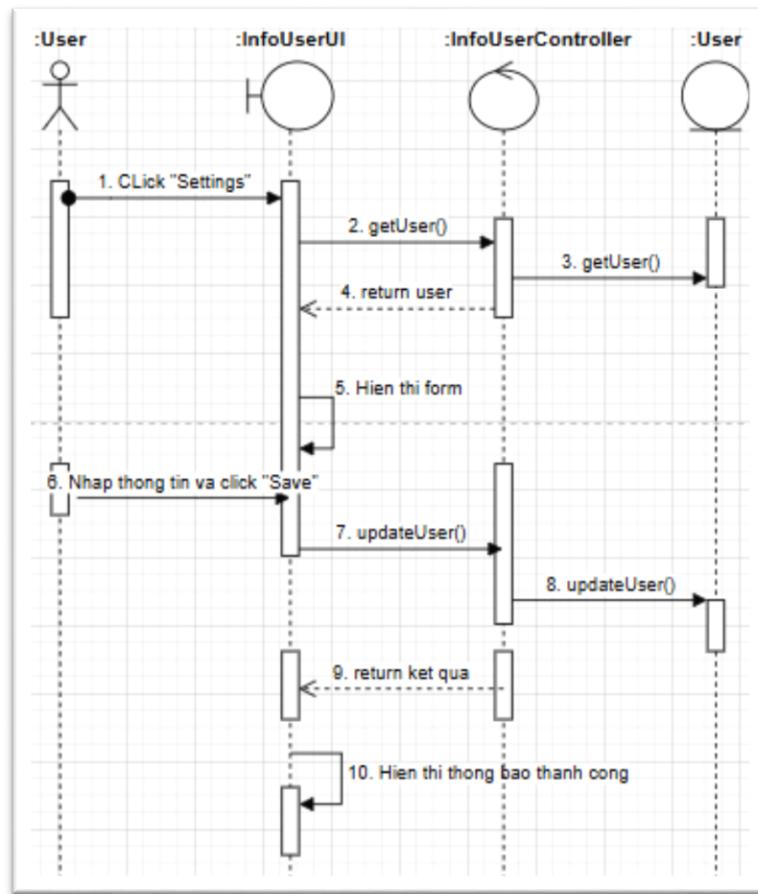


Hình 2. 7. Biểu đồ tuần tự use case Quên mật khẩu

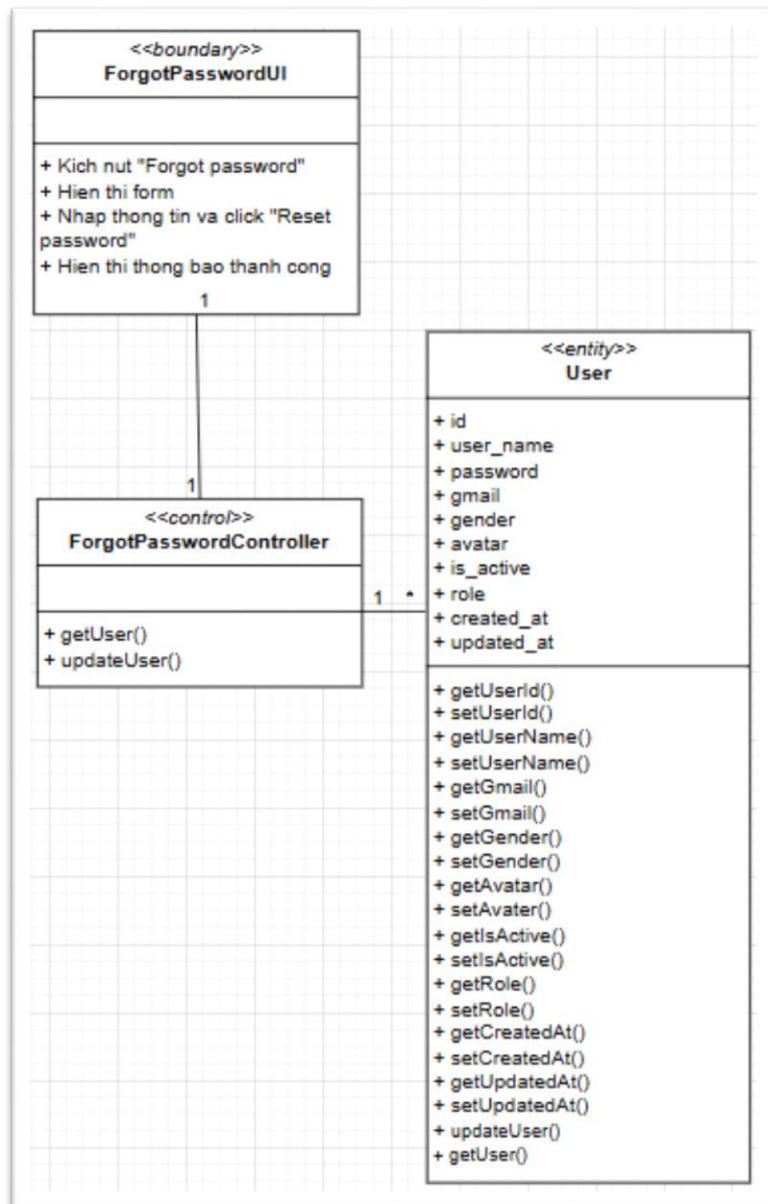


Hình 2. 8. VOPC use case Quên mật khẩu

#### 2.2.4.4. Phân tích use case Cập nhật thông tin tài khoản

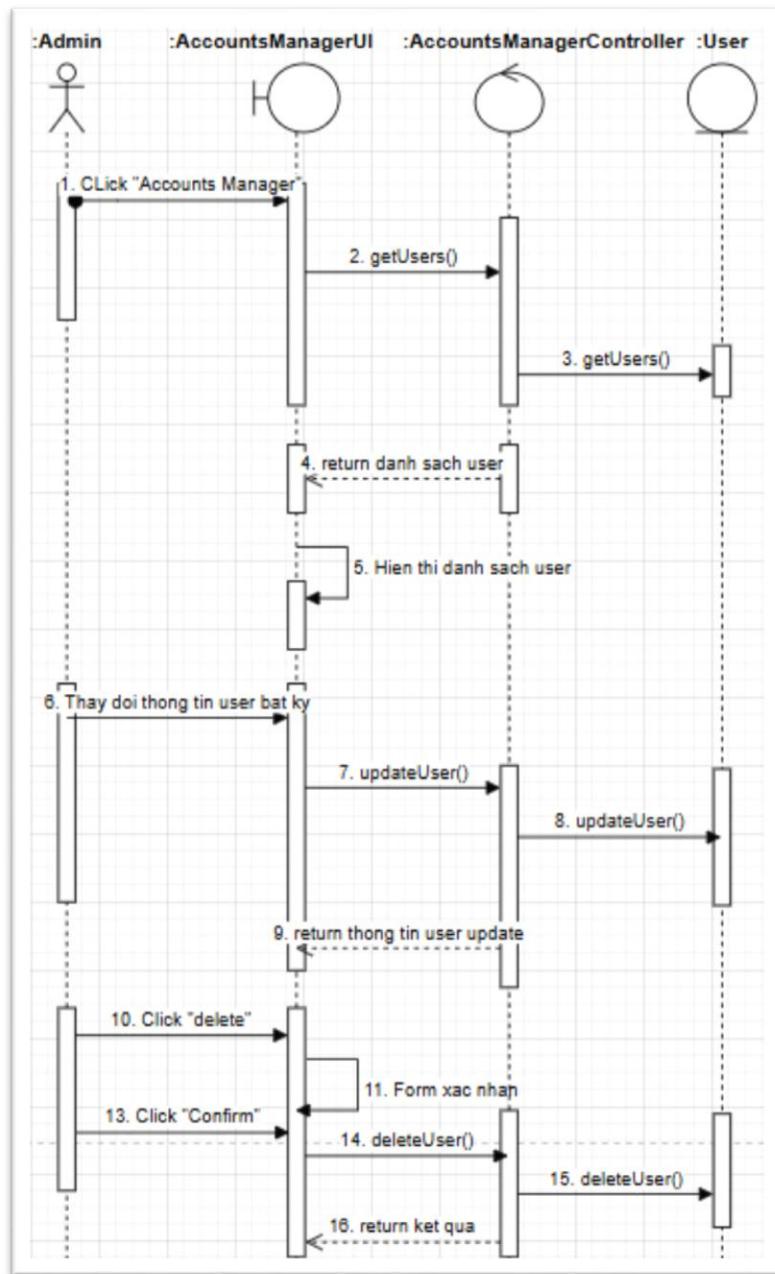


Hình 2. 9. Biểu đồ tuần tự use case Cập nhật thông tin tài khoản

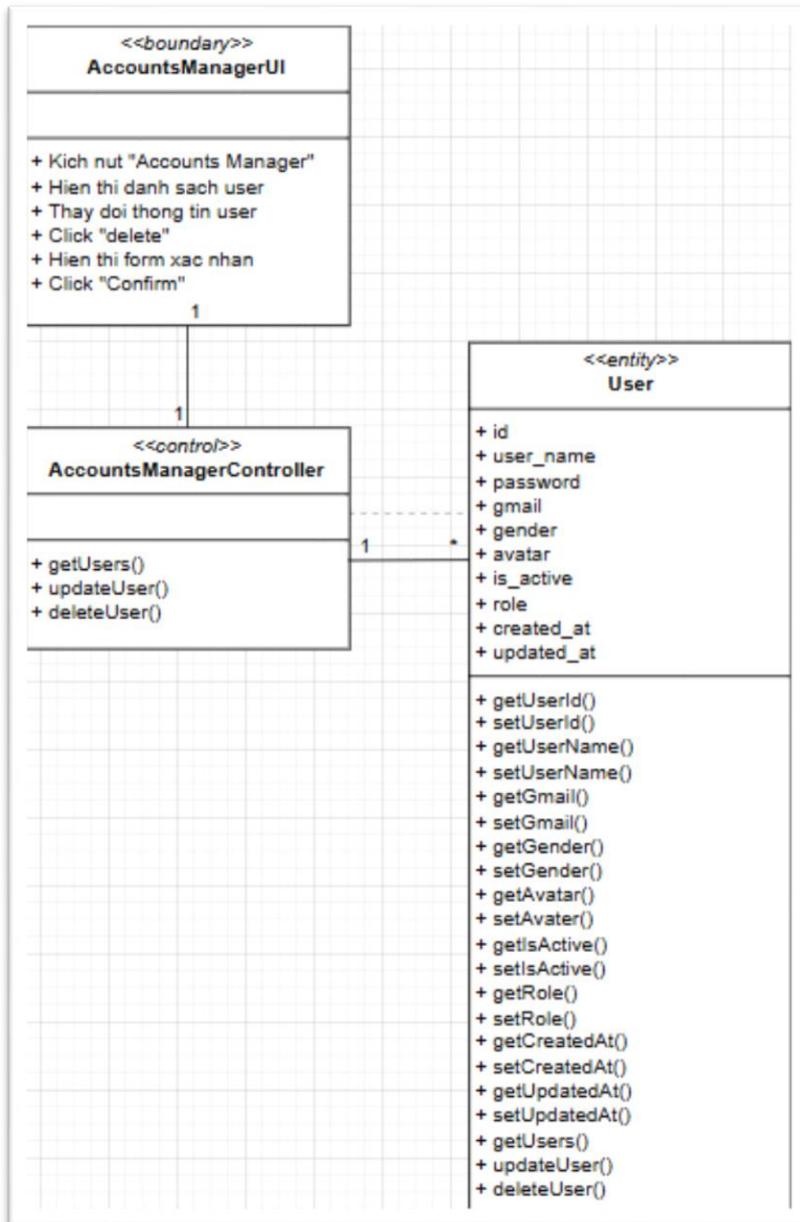


Hình 2. 10. VOPC use case Cập nhật thông tin tài khoản

#### 2.2.4.5. Phân tích use case Quản lý tài khoản (Admin)

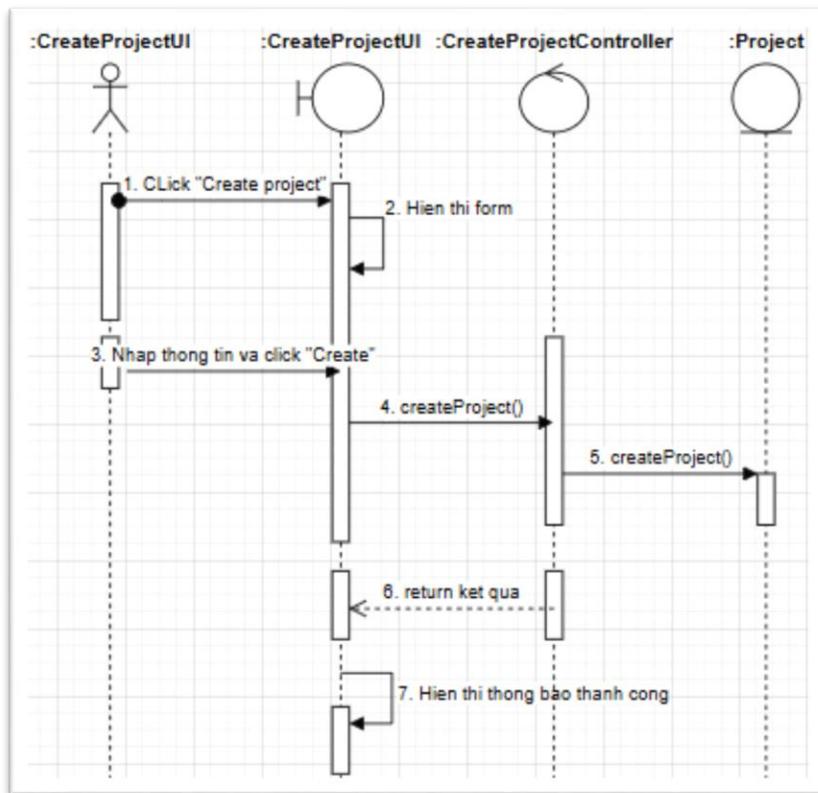


Hình 2. 11. Biểu đồ tuần tự use case Quản lý tài khoản (Admin)

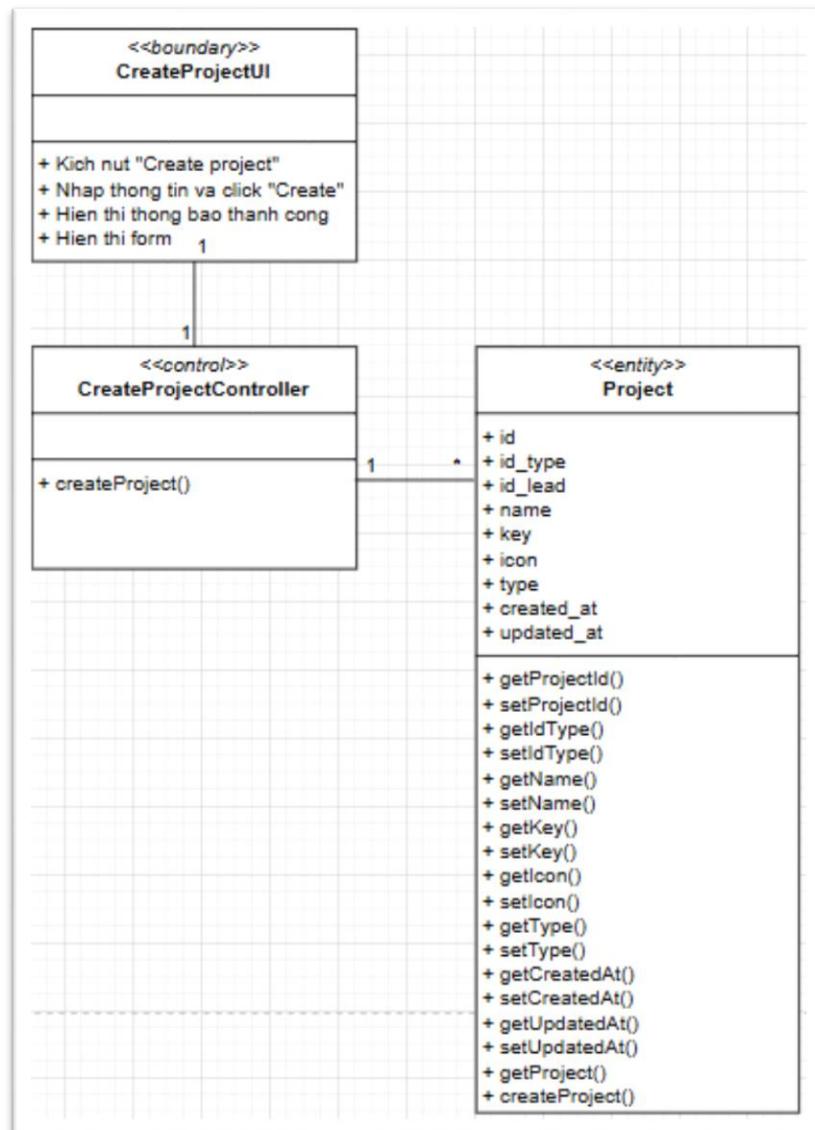


Hình 2. 12. VOPC use case Quản lý tài khoản (Admin)

#### 2.2.4.6. Phân tích use case Tạo project

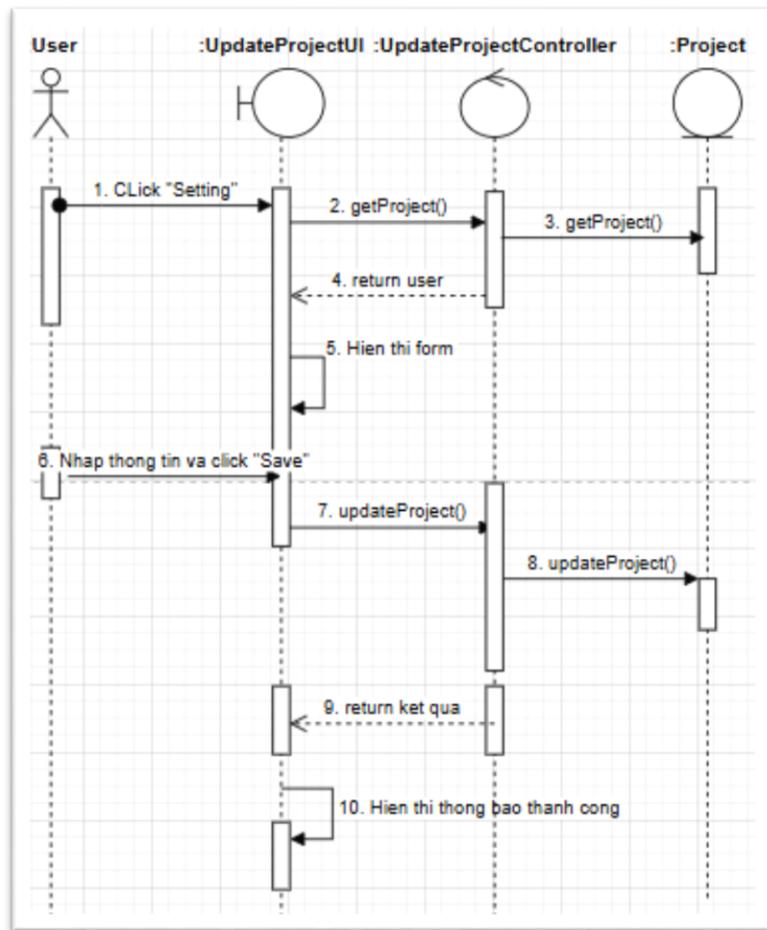


Hình 2. 13. Biểu đồ tuần tự use case Tạo project

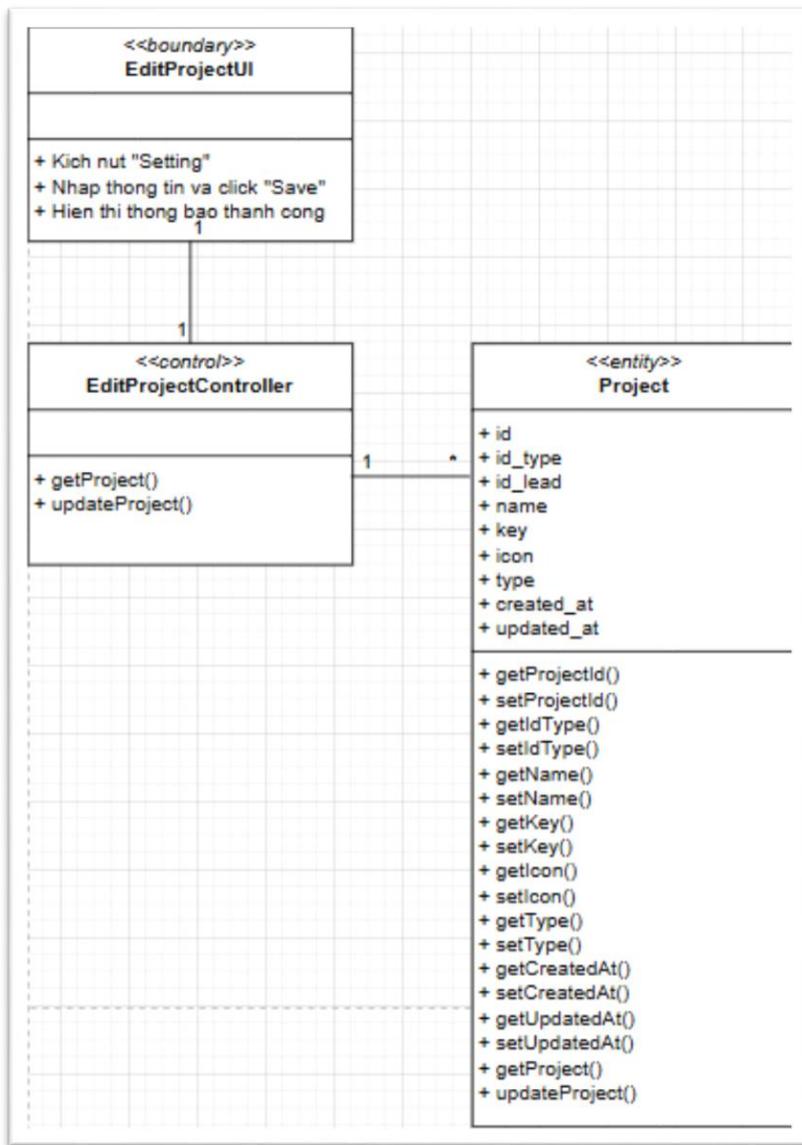


Hình 2. 14. VOPC use case Tạo project

### 2.2.4.7. Phân tích use case Chính sửa project

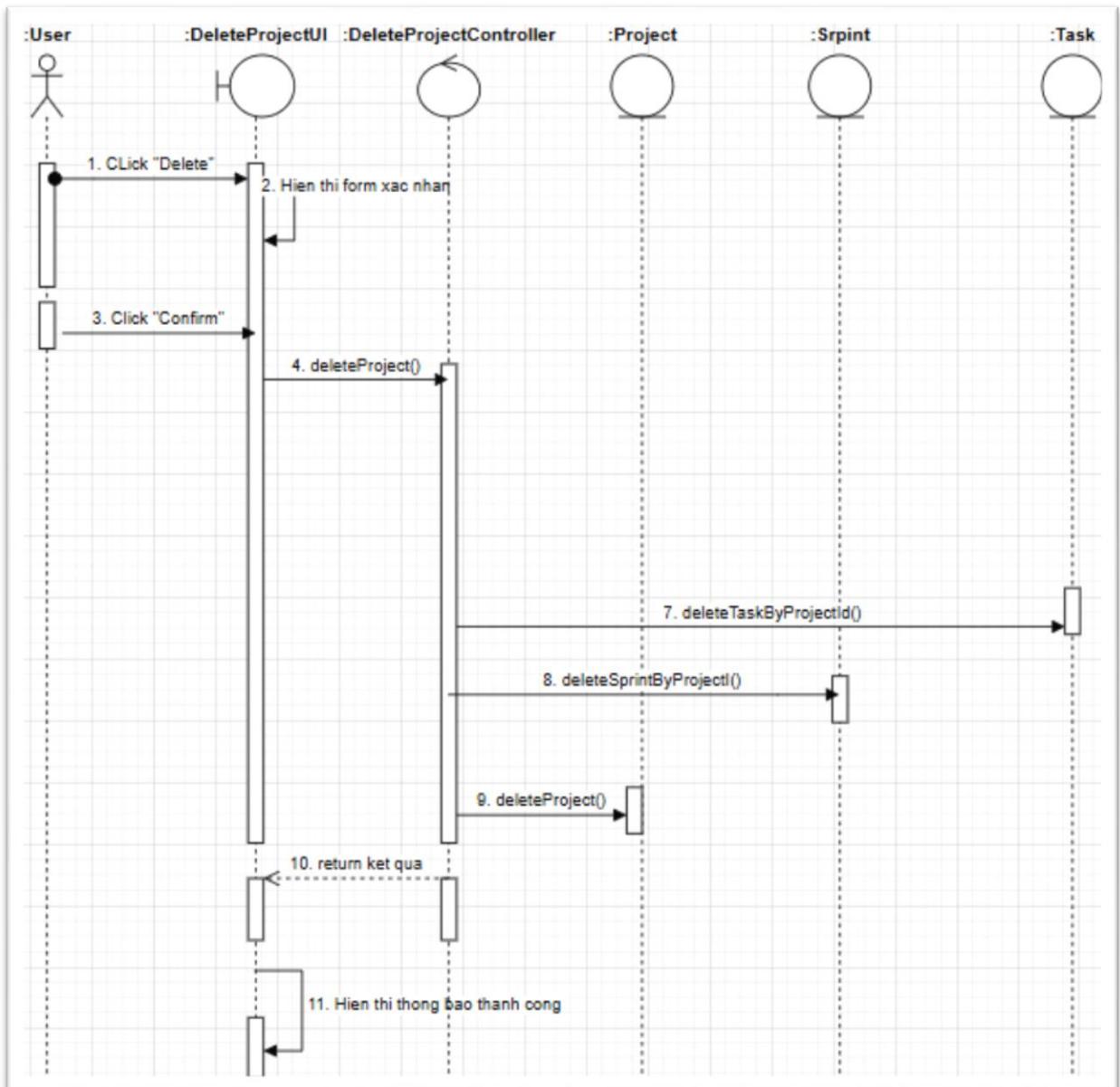


Hình 2. 15. Biểu đồ tuần tự use case Chính sửa project

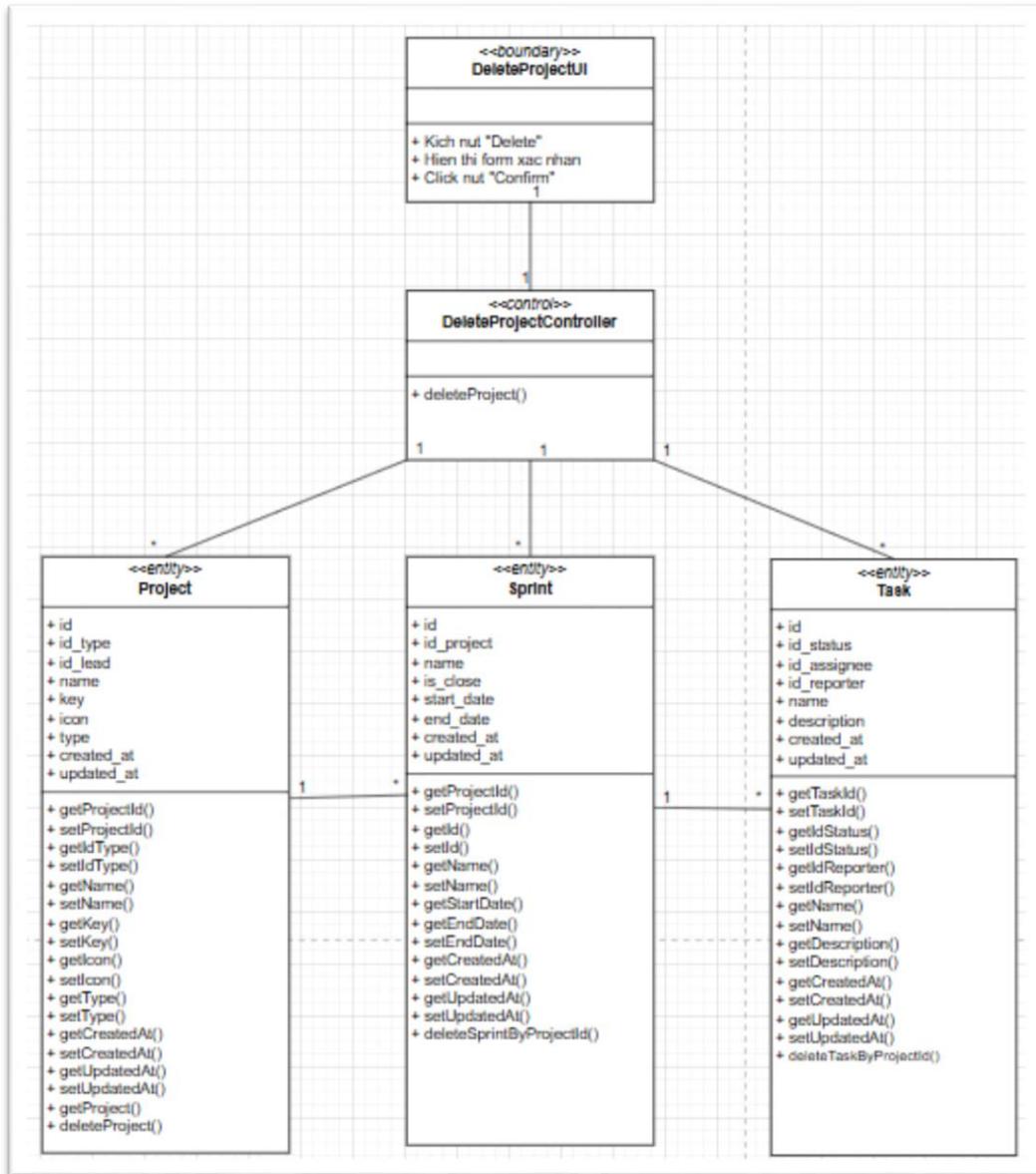


Hình 2. 16. VOPC use case *Chỉnh sửa project*

### 2.2.4.8. Phân tích use case Xóa project

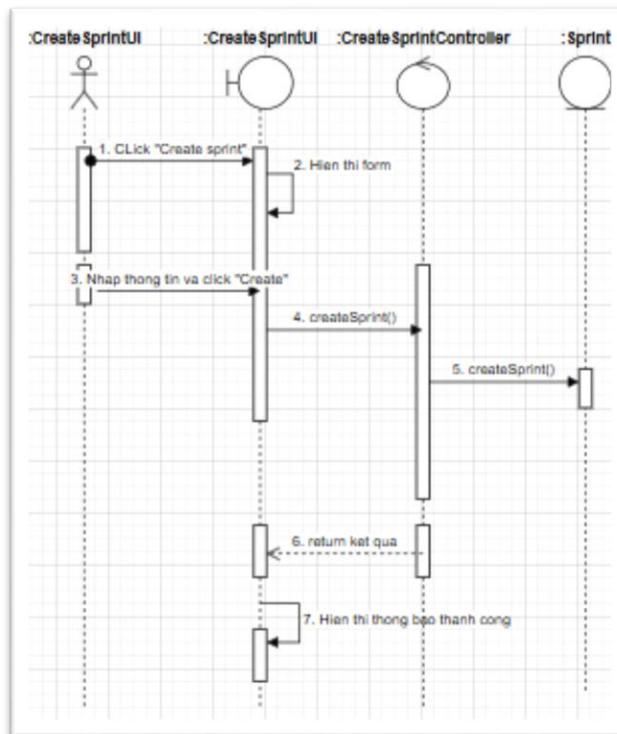


Hình 2. 17. Biểu đồ tuần tự use case Xóa project

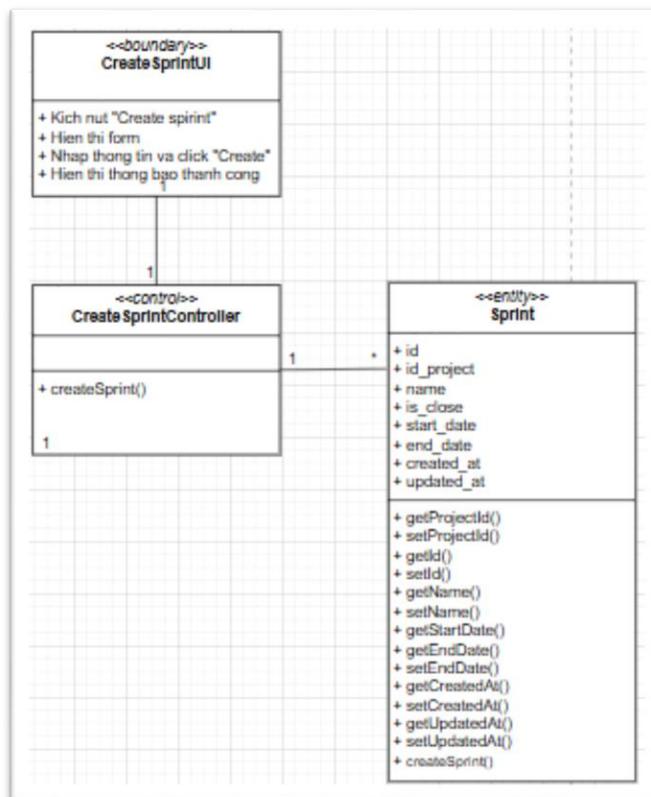


Hình 2. 18. VOPC use case Xóa project

### 2.2.4.9. Phân tích use case Tạo sprint

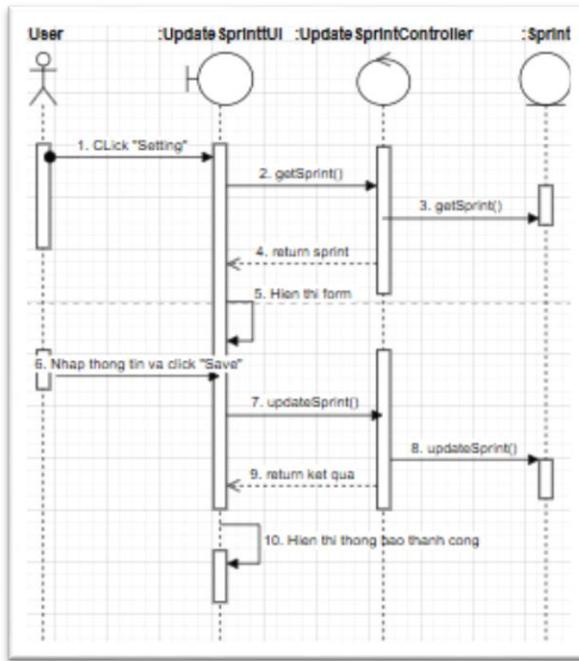


Hình 2. 19. Biểu đồ tuần tự use case Tạo sprint

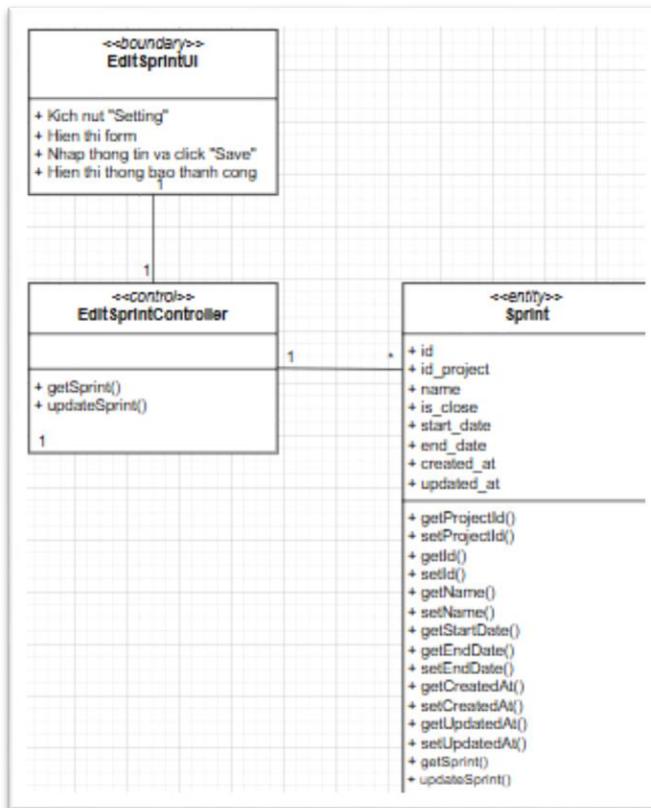


Hình 2. 20. VOPC use case Tạo sprint

#### 2.2.4.10. Phân tích use case Chính sửa sprint

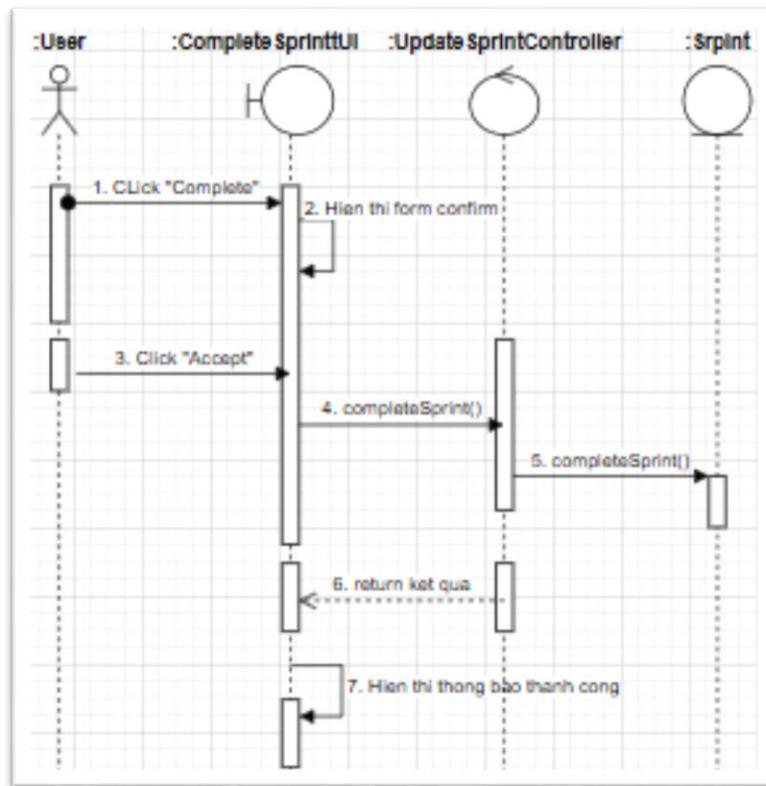


Hình 2. 21. Biểu đồ tuần tự Chính sửa sprint

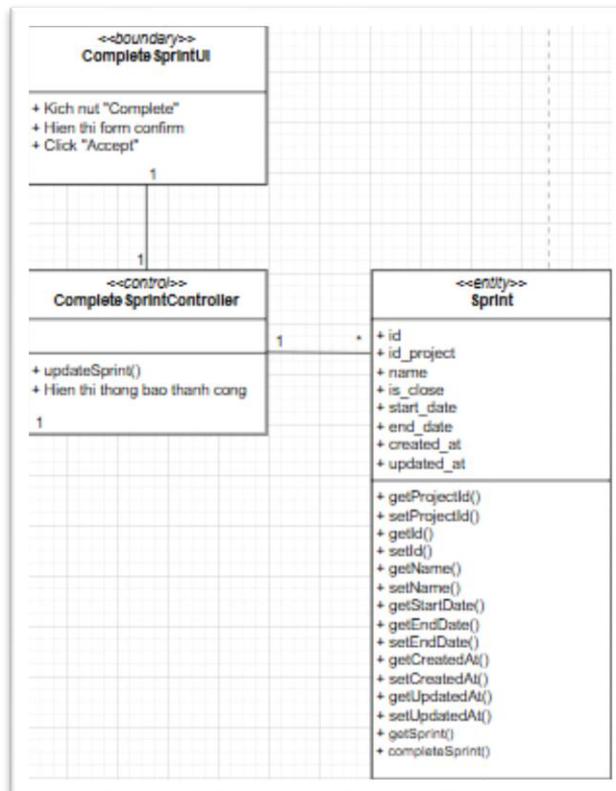


Hình 2. 22. VOPC use case Chính sửa Sprint

### 2.2.4.11. Phân tích use case Kết thúc sprint

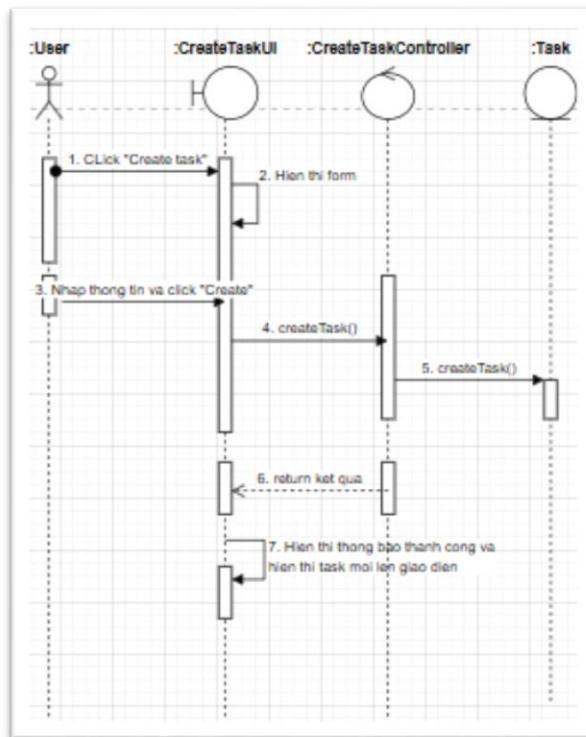


Hình 2. 23. Biểu đồ tuần tự use case Kết thúc sprint

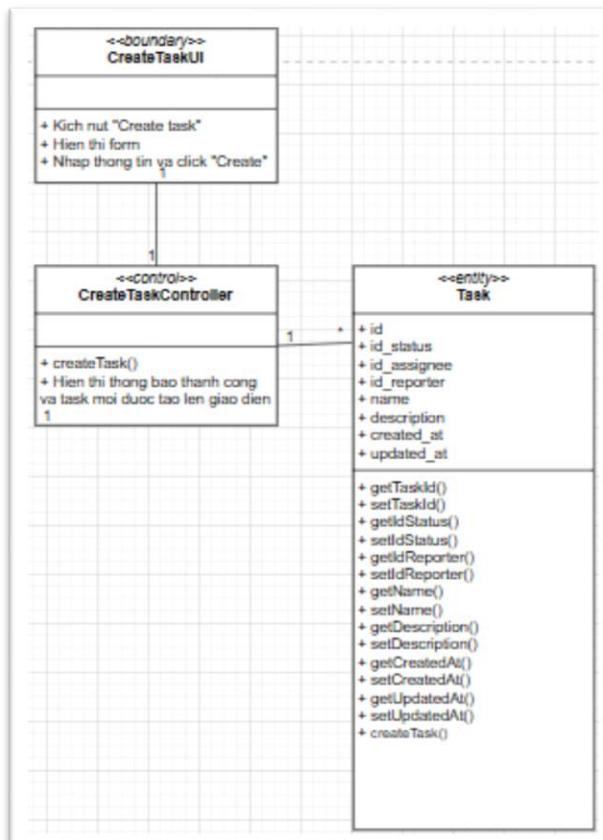


Hình 2. 24. VOPC use case Kết thúc sprint

### 2.2.4.12. Phân tích use case Tạo task

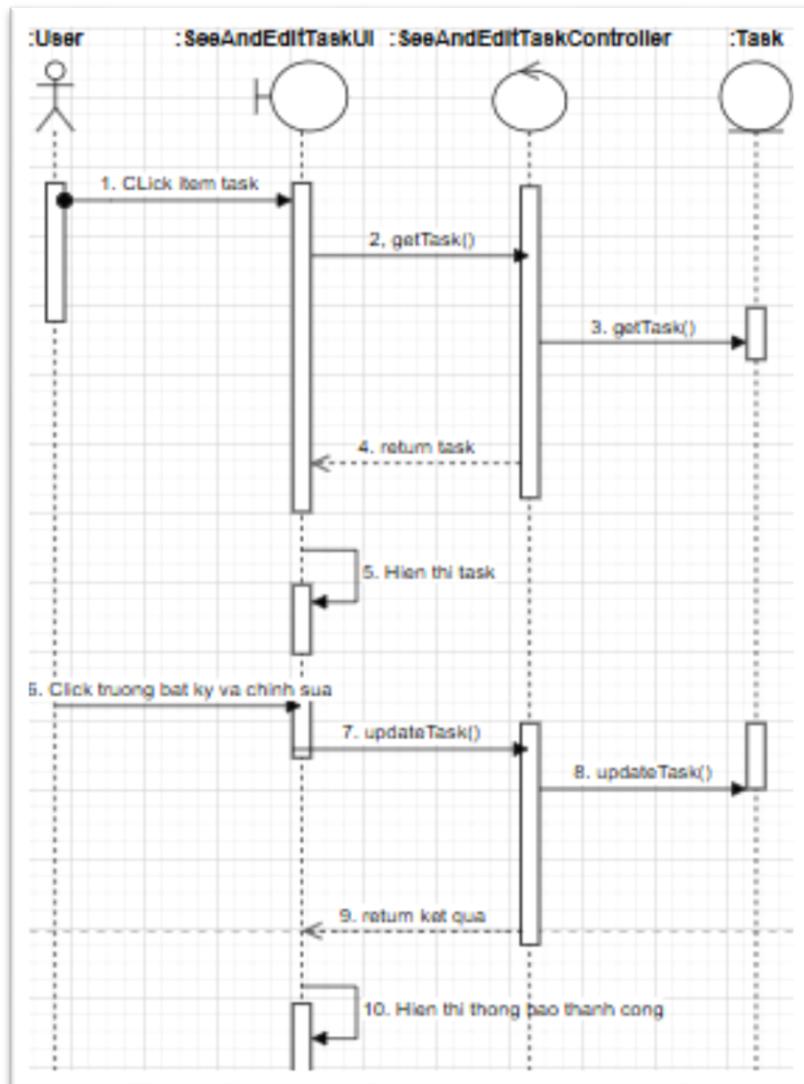


Hình 2. 25. Biểu đồ tuần tự use case Tạo task

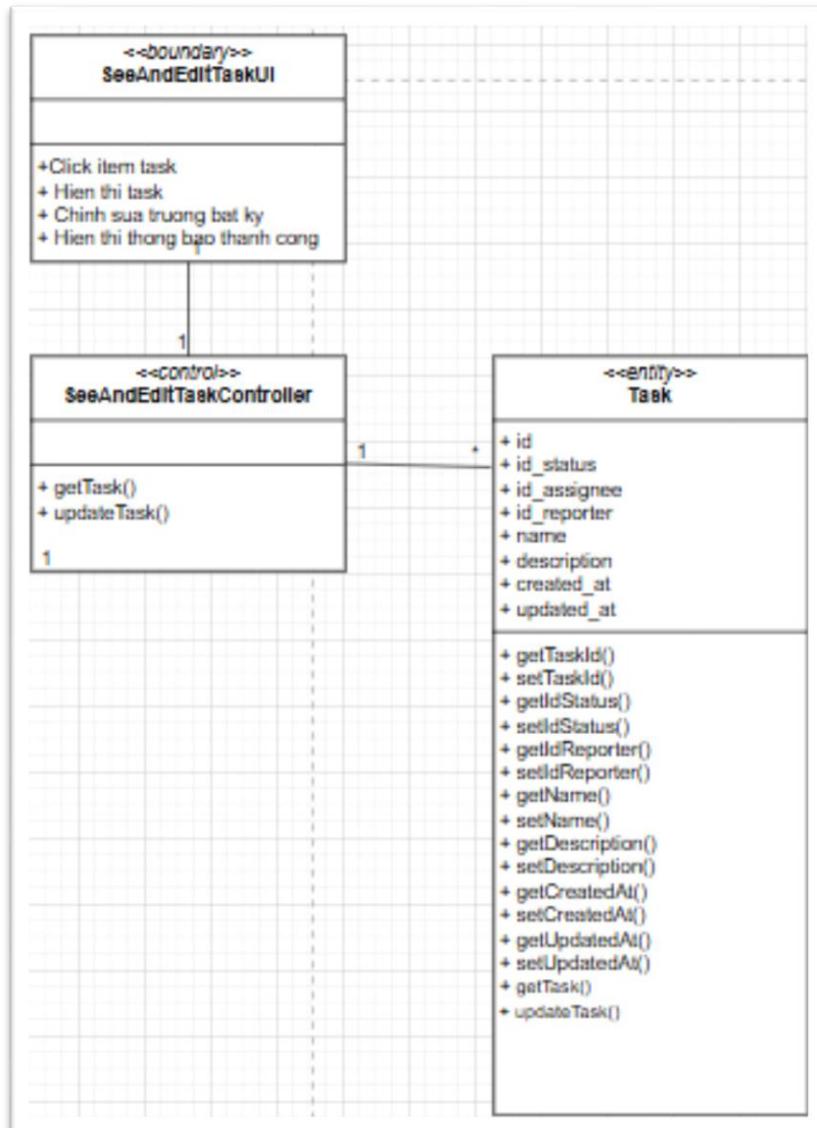


Hình 2. 26. VOPC use case Tạo task

#### 2.2.4.13. Phân tích use case Xem chi tiết/chỉnh sửa task

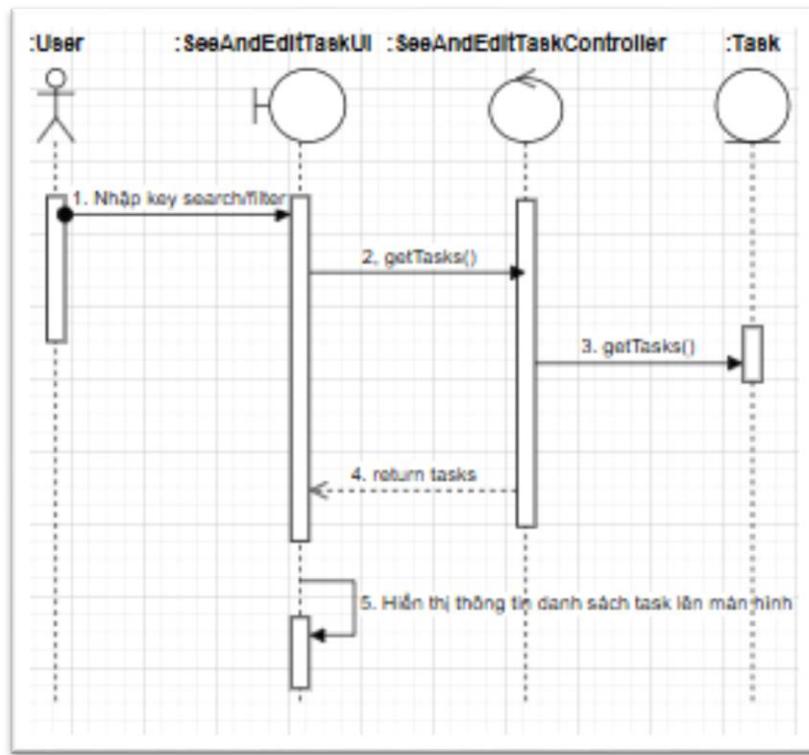


Hình 2. 27. Biểu đồ tuần tự use case Xem chi tiết/Chỉnh sửa task

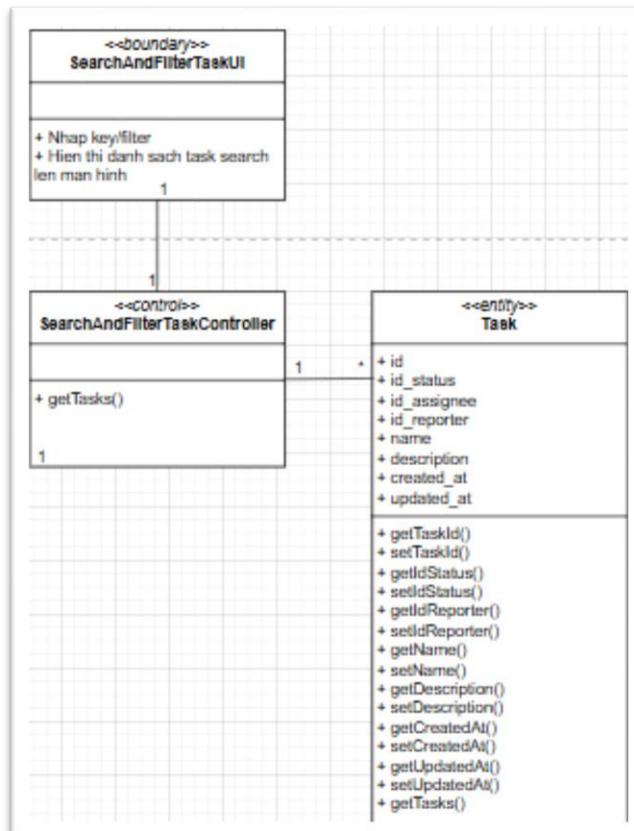


Hình 2. 28. VOPC use case Xem chi tiết/chỉnh sửa task

#### 2.2.4.14. Phân tích use case search/filter task

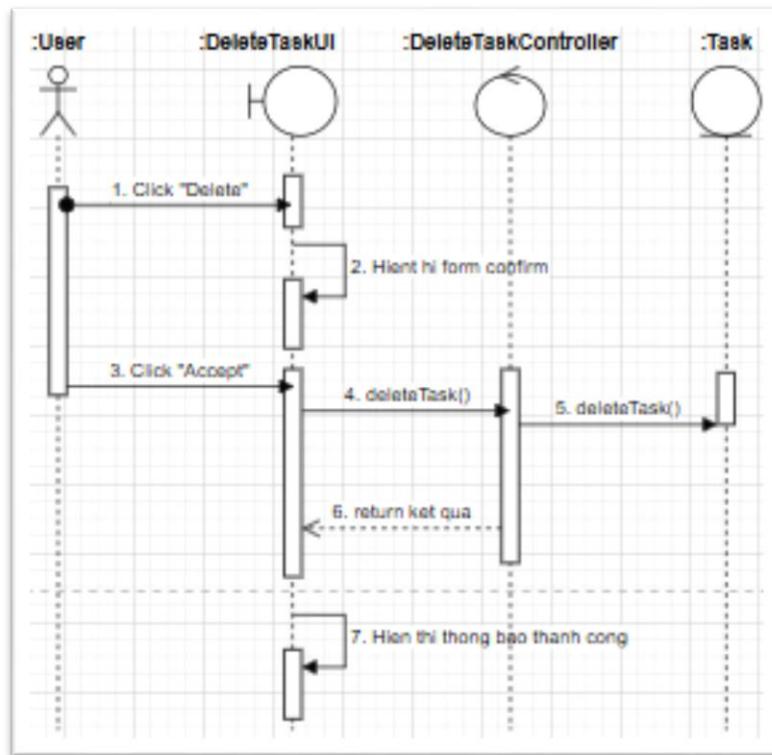


Hình 2. 29. Biểu đồ tuần tự use case Search/filter task

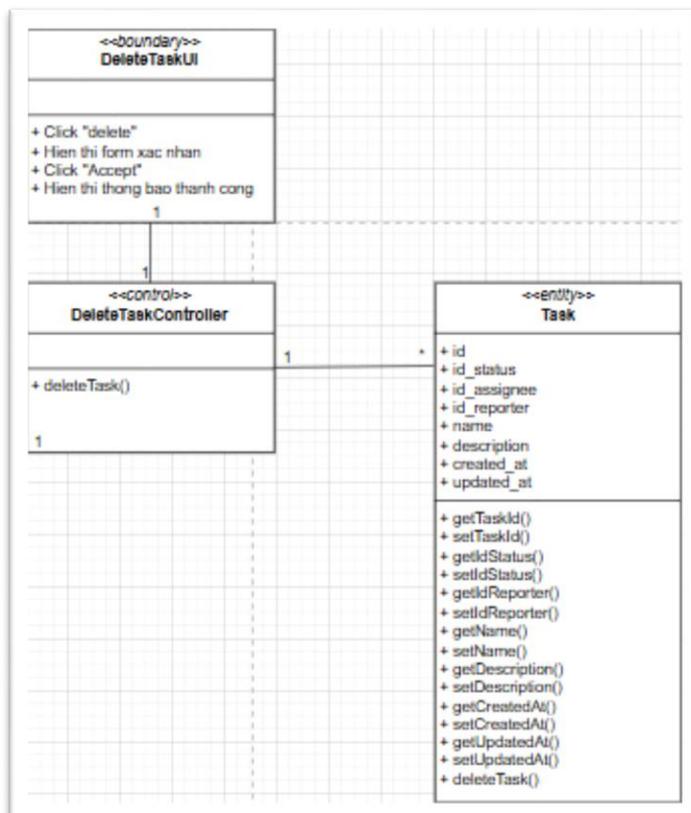


Hình 2. 30. VOPC use case Search/filter task

### 2.2.4.15. Phân tích use case Xóa task

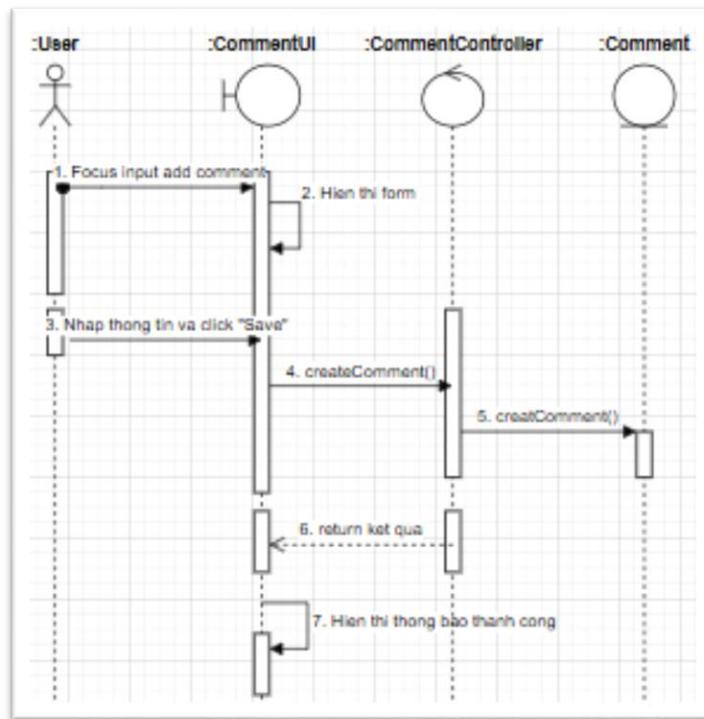


Hình 2. 31. Biểu đồ tuần tự use case Xóa task

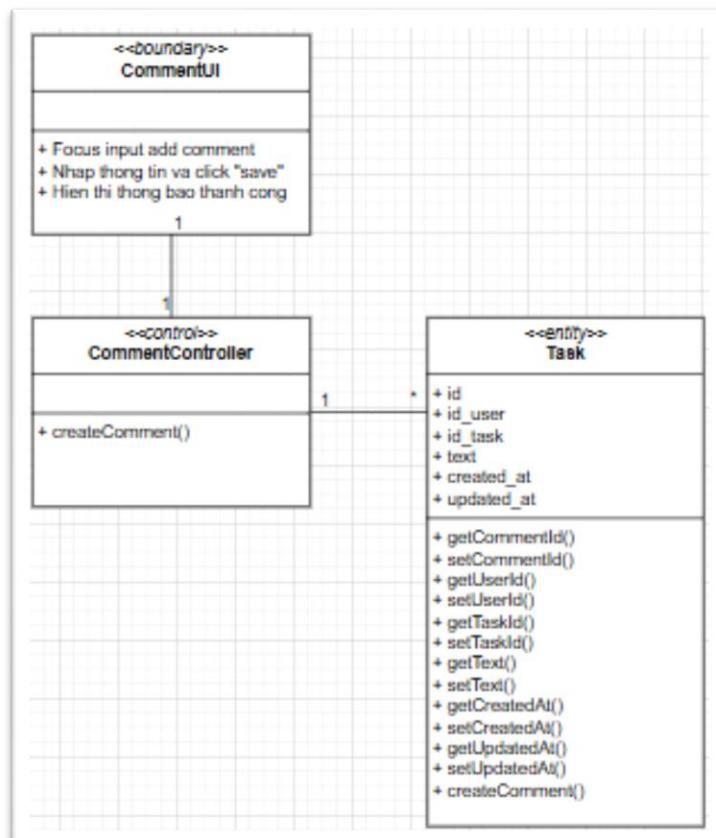


Hình 2. 32. VOPC use case Xóa task

#### 2.2.4.16. Phân tích use case Comment task

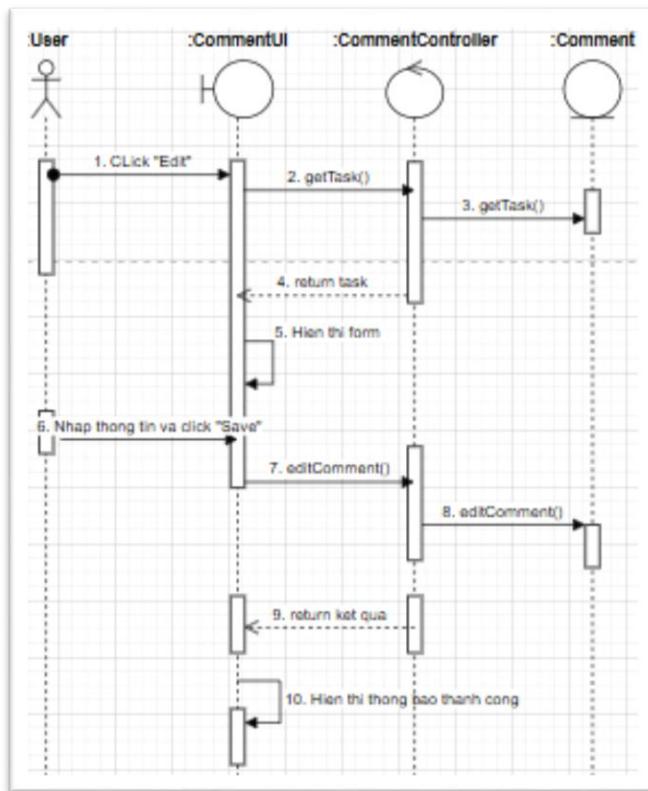


Hình 2. 33. Biểu đồ tuần tự use case Comment task

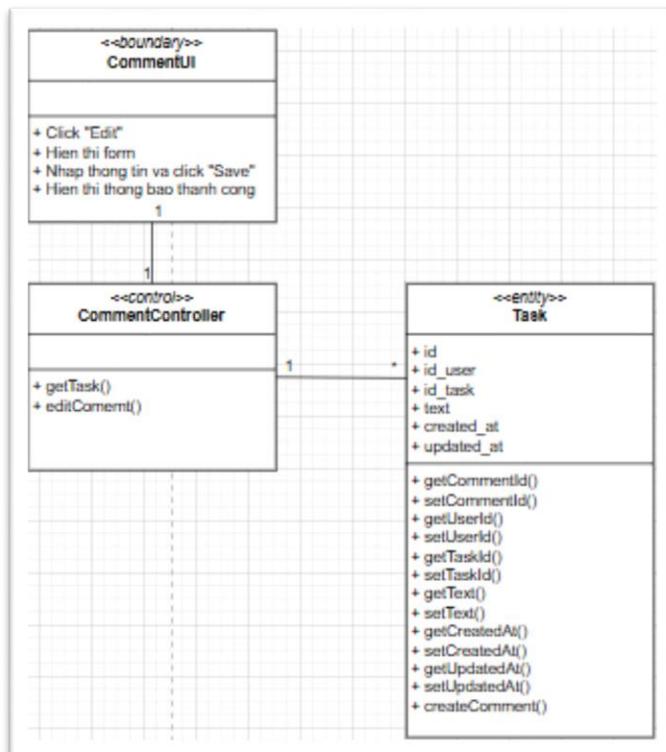


Hình 2. 34. VOPC use case Comment task

### 2.2.4.17. Phân tích use case Chính sửa comment

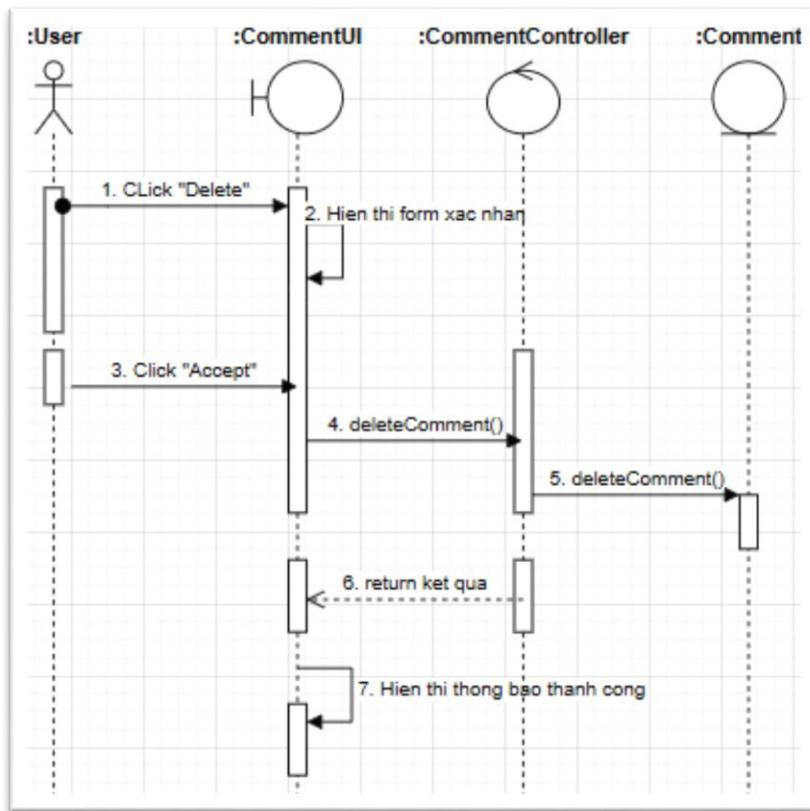


Hình 2. 35. Biểu đồ tuần tự use case Chính sửa comment

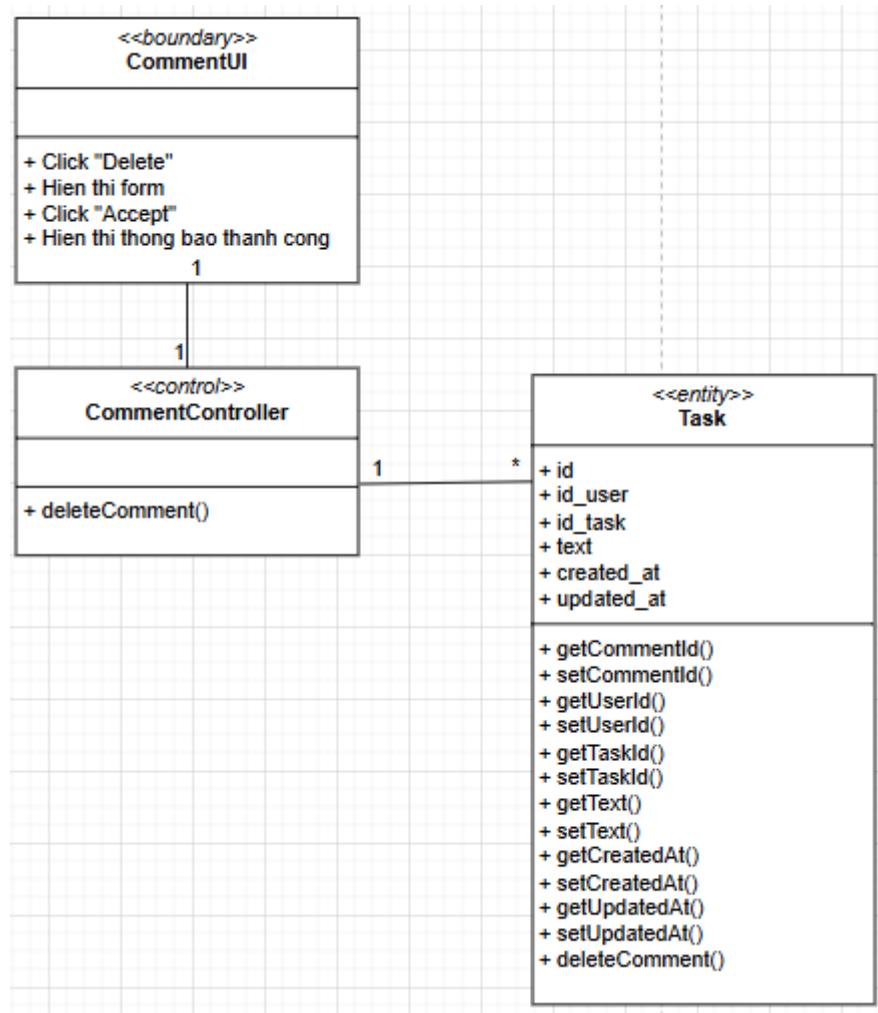


Hình 2. 36. VOPC use case Chính sửa comment

#### 2.2.4.18. Phân tích use case Xóa comment



Hình 2. 37. Biểu đồ tuần tự use case Xóa comment



Hình 2. 38. VOPC use case Xóa comment

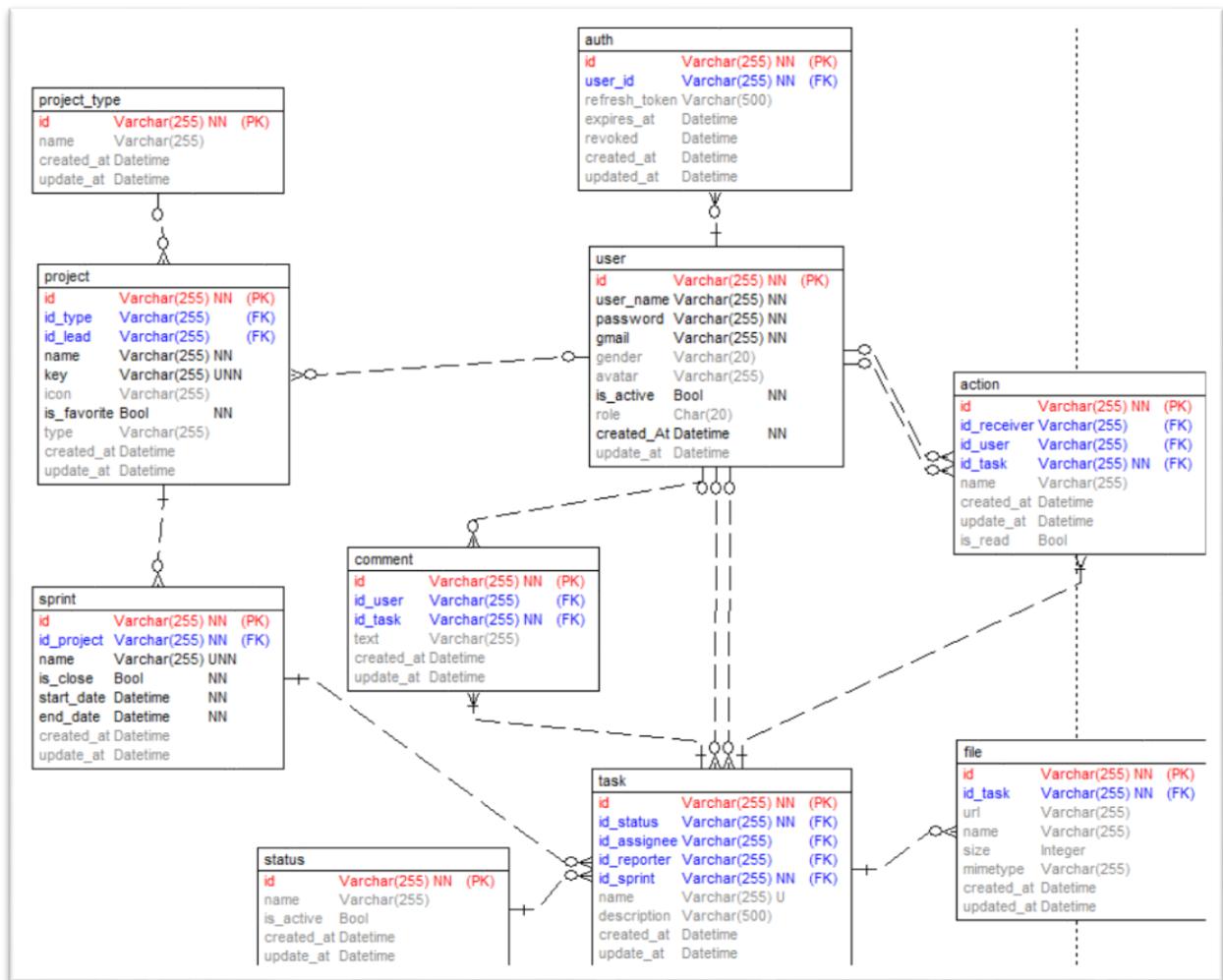
## 2.2.5. Mô hình hóa dữ liệu

### 2.2.5.1. Mô tả dữ liệu

- Để theo tác trên website, người dùng cần có tài khoản: nhập đầy đủ các thông tin theo yêu cầu của hệ thống. Sau khi đã tạo thành công tài khoản của người dùng sẽ được lưu trong cơ sở dữ liệu. Khi đã có tài khoản, người dùng có thể Đăng nhập hệ thống để thao tác. Hệ thống cần có bảng **user** để quản lý các người dùng sử dụng hệ thống: id, gmail, passwords, user\_name, avatar, birthday, gender, role, createdAt. Và chúng ta cần bảng **“auth”** để quản lý và phiên đăng nhập của user: id, user\_id, refresh\_token, expires\_at, revoked, created\_at, updated\_at.

- Bước đầu tiên để làm việc là cần tạo project từ đó sẽ có bảng “**project**”, mỗi project cần có các thông tin: id, id\_type, id\_lead, name, key, icon, is\_favorite, created\_at, update\_at.
- Mỗi project sẽ có các type riêng nên ta cần 1 bảng “**project\_type**” chứa các trường: id, name, created\_at, update\_at.
- Tiếp đến khi tao thắc với project ta cần tạo ra bảng “**sprint**” để lưu trữ các sprint để kiểm soát và ước lượng 1 khoảng thời gian cho một nhóm các task ta cần các trường: id, id\_project, name, is\_close, start\_date, end\_date, created\_at, update\_at.
- Trong 1 sprint sẽ có các đầu việc khác nhau, ta cần tạo ra bảng “**task**” để lưu trữ và mỗi đầu việc cần có một số thông tin: id, id\_status, id\_assignee, id\_reporter, id\_sprint, name, description, created\_at, update\_at.
- Chúng ta cần config các status để đồng nhất và cần bảng “**status**” để lưu trữ: id, name, is\_active, created\_at, update\_at.
- Để mạch lạch và ghi dấu lại chúng ta cần comment lại vào task khi thao tác chỉnh sửa trên task nên ta cần bảng “**comment**” với các trường sau: id, id\_user, id\_task, text, created\_at, update\_at.
- Để tạo thông báo ta cần thực hiện các action, mỗi action cần có các thông tin xác định và 1 bảng lưu trữ “**action**”: id, is\_user, id\_task, name, created\_at, update\_at.
- Tiếp đó ta có nhu cầu upload file đính kèm với task nên ta cũng cần 1 bảng “**file**” để lưu trữ các thông tin của file: id, url, name, size, minetype, created\_at, updated\_at.

### 2.2.5.2. Mô hình cơ sở dữ liệu hệ thống



Hình 2. 39. Mô hình cơ sở dữ liệu hệ thống

### 2.2.6. Thiết kế các bảng dữ liệu

#### 2.2.6.1. Bảng user

Attributes		Properties	Advanced	Indexes	Alternate Keys	Others	Relationship	Storage	Comments	Notes	DDL
1	PK	Key	Name	Column Name	Datatype	Not null	Unique	Description			
2			id	id	Varchar(255)						
3			user_name	user_name	Varchar(255)						
4			password	password	Varchar(255)						
5			gmail	gmail	Varchar(255)						
6			gender	gender	Varchar(20)						
7			avatar	avatar	Varchar(255)						
8			is_active	is_active	Bool						
9			role	role	Char(20)						
10			created_At	created_At	Datetime						
			update_at	update_at	Datetime						

Hình 2. 40. Thiết kế bảng user

### 2.2.6.2. Bảng auth

Attributes Properties Advanced Indexes Alternate Keys Others Relationship Storage Comments Notes DDL							
Key	Name	Column Name	Datatype	Not null	Unique	Description	
1	PK	id	Varchar(255)	Not null			
2	FK	id	Varchar(255)	Not null			
3		refresh_token	Varchar(500)				
4		expires_at	Datetime				
5		revoked	Datetime				
6		created_at	Datetime				
7		updated_at	Datetime				

Hình 2. 41. Thiết kế bảng auth

### 2.2.6.3. Bảng project

Attributes Properties Advanced Indexes Alternate Keys Others Relationship Storage Comments Notes DDL							
Key	Name	Column Name	Datatype	Not null	Unique	Description	
1	PK	id	Varchar(255)	Not null			
2	FK	id	Varchar(255)	Not null			
3	FK	id_lead	Varchar(255)	Not null			
4		name	Varchar(255)	Not null			
5		key	Varchar(255)	Not null	Unique		
6		icon	Varchar(255)				
7		is_favorite	Bool	Not null			
8		type	Varchar(255)				
9		created_at	Datetime				
10		update_at	Datetime				

Hình 2. 42. Thiết kế bảng project

### 2.2.6.4. Bảng project\_type

Attributes Properties Advanced Indexes Alternate Keys Others Relationship Storage Comments Notes DDL							
Key	Name	Column Name	Datatype	Not null	Unique	Description	
1	PK	id	Varchar(255)	Not null			
2		name	Varchar(255)				
3		created_at	Datetime				
4		update_at	Datetime				

Hình 2. 43. Thiết kế bảng project\_type

### 2.2.6.5. Bảng sprint

Attributes Properties Advanced Indexes Alternate Keys Others Relationship Storage Comments Notes DDL							
	Key	Name	Column Name	Datatype	Not null	Unique	Description
1	PK	id	id	Varchar(255)	Not null		
2	PK	id	id_project	Varchar(255)	Not null		
3		name	name	Varchar(255)	Not null	Unique	
4		is_close	is_close	Bool	Not null		
5		start_date	start_date	Datetime	Not null		
6		end_date	end_date	Datetime	Not null		
7		created_at	created_at	Datetime			
8		update_at	update_at	Datetime			

Hình 2. 44. Thiết kế bảng sprint

### 2.2.6.6. Bảng task

Attributes Properties Advanced Indexes Alternate Keys Others Relationship Storage Comments Notes DDL							
	Key	Name	Column Name	Datatype	Not null	Unique	Description
1	PK	id	id	Varchar(255)	Not null		
2	PK	id	id_status	Varchar(255)	Not null		
3	PK	id	id_assignee	Varchar(255)			
4	PK	id	id_reporter	Varchar(255)			
5	PK	id	id_sprint	Varchar(255)	Not null		
6		name	name	Varchar(255)		Unique	
7		description	description	Varchar(500)			
8		created_at	created_at	Datetime			
9		update_at	update_at	Datetime			

Hình 2. 45. Thiết kế bảng task

### 2.2.6.7. Bảng status

Attributes Properties Advanced Indexes Alternate Keys Others Relationship Storage Comments Notes DDL							
	Key	Name	Column Name	Datatype	Not null	Unique	Description
1	PK	id	id	Varchar(255)	Not null		
2		name	name	Varchar(255)			
3		is_active	is_active	Bool			
4		created_at	created_at	Datetime			
5		update_at	update_at	Datetime			

Hình 2. 46. Thiết kế bảng status

### 2.2.6.8. Bảng action

Attributes Properties Advanced Indexes Alternate Keys Others Relationship Storage Comments Notes DDL							
	Key	Name	Column Name	Datatype	Not null	Unique	Description
1	PK	id	id	Varchar(255)	Not null		
2	PK	id	id_receiver	Varchar(255)			
3	PK	id	id_user	Varchar(255)			
4	PK	id	id_task	Varchar(255)	Not null		
5		name	name	Varchar(255)			
6		created_at	created_at	Datetime			
7		update_at	update_at	Datetime			
8		is_read	is_read	Bool			

Hình 2. 47. Thiết kế bảng action

### 2.2.6.9. Bảng comment

Attributes Properties Advanced Indexes Alternate Keys Others Relationship Storage Comments Notes DDL							
	Key	Name	Column Name	Datatype	Not null	Unique	Description
1	PK	id	id	Varchar(255)	Not null		
2	PK	id	id_user	Varchar(255)			
3	PK	id	id_task	Varchar(255)	Not null		
4		text	text	Varchar(255)			
5		created_at	created_at	Datetime			
6		update_at	update_at	Datetime			

Hình 2. 48. Thiết kế bảng comment

### 2.2.6.10. Bảng file

Attributes Properties Advanced Indexes Alternate Keys Others Relationship Storage Comments Notes DDL							
	Key	Name	Column Name	Datatype	Not null	Unique	Description
1	PK	id	id	Varchar(255)	Not null		
2	PK	id	id_task	Varchar(255)	Not null		
3		url	url	Varchar(255)			
4		name	name	Varchar(255)			
5		size	size	Integer			
6		mimetype	mimetype	Varchar(255)			
7		created_at	created_at	Datetime			
8		updated_at	updated_at	Datetime			

Hình 2. 49. Thiết kế bảng file

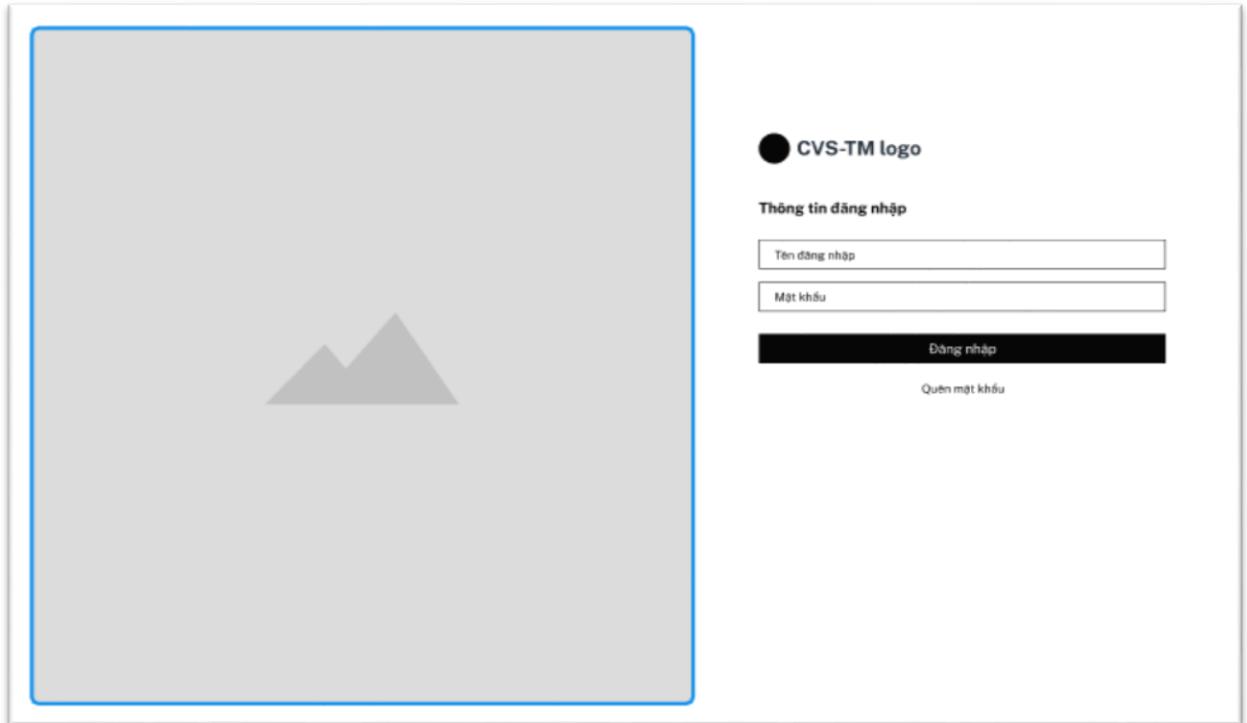
## 2.2.7. Thiết kế giao diện

### 2.2.7.1. Màn hình Trang chủ



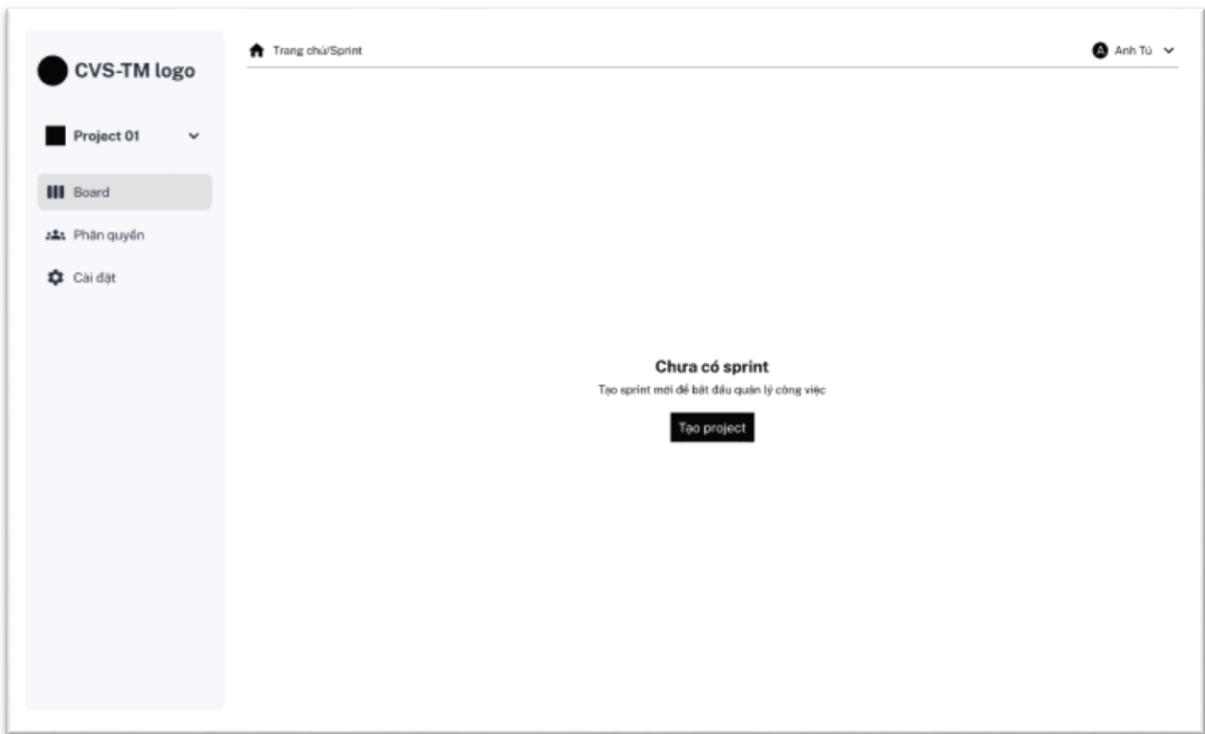
Hình 2. 50. Thiết kế giao diện trang chủ

### 2.2.7.2. Màn hình Đăng nhập



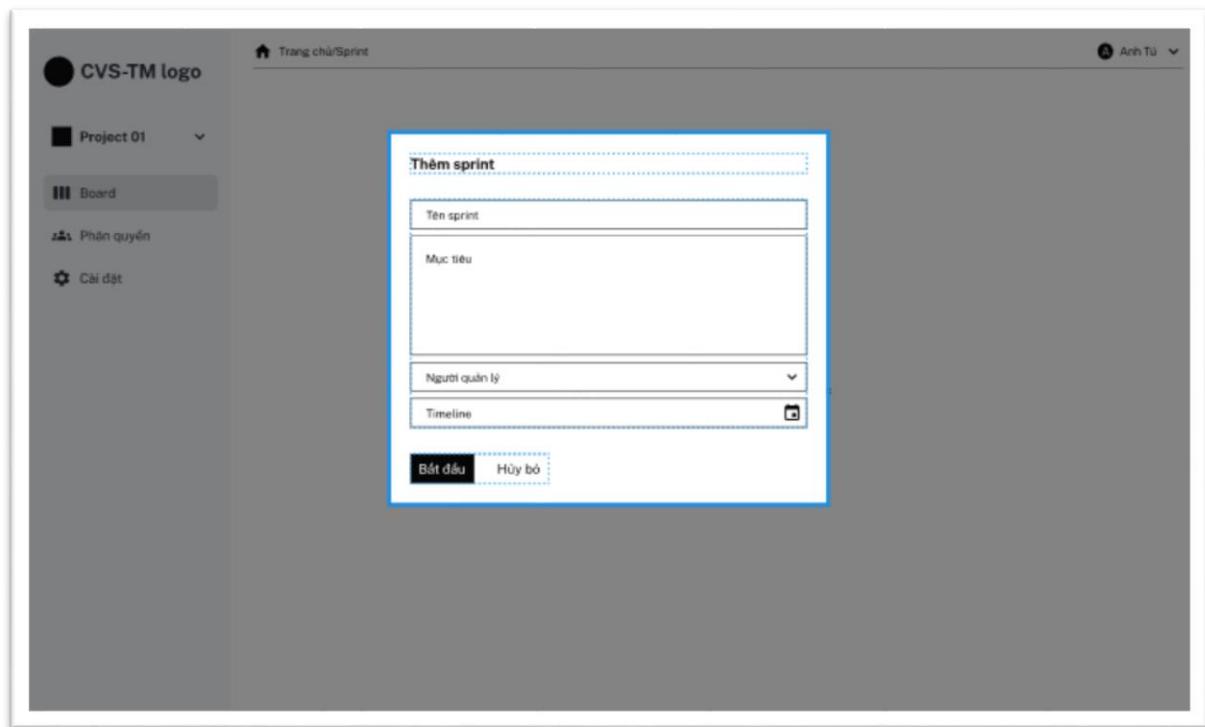
Hình 2. 51. Thiết kế giao diện màn hình Đăng nhập

### 2.2.7.3. Màn hình Chi tiết project



Hình 2. 52. Thiết kế giao diện màn hình Chi tiết project

### 2.2.7.4. Màn hình Thêm sprint



Hình 2. 53. Thiết kế giao diện màn hình Thêm sprint

### 2.2.7.5. Màn hình Chi tiết sprint

The screenshot shows the CVS-TM software interface. At the top, there is a header with the project logo, name, and a search bar. Below the header, the title "Sprint 01" is displayed, along with a status bar indicating "Kết thúc vào 22/12/2024" and a "Đóng sprint" button.

The main area is a board view with four columns: "To do", "In Progress", "Ready to test", and "Done". Each column contains several tasks, each with a brief description, a due date, and a progress indicator (green checkmark). The tasks are:

- To do:**
  - Người dùng muốn có option "Không hiển thị lại thông báo" khi truy cập fastdoc.vn và app.fastdoc.vn
  - Admin muốn tích hợp Smart align và Fastdoc
  - Lỗi tự động quay lại tổ chức mặc định khi đăng nhập trên Database
  - Admin muốn kéo rộng trường thông tin Căn cứ xử lý trên
- In Progress:**
  - Fastdoc bị lỗi server
  - Admin muốn thêm các giấy tờ dịch vụ công trên <https://doc.diginotes.online/>
  - Tối ưu tốc độ load của Bảng khi review tài liệu
  - Admin/Người sử dụng muốn search theo dữ liệu có trong file dữ liệu trả về
  - Admin muốn update review bảng đang nằm ngang (Giống như Hipot)
- Ready to test:**
  - Nâng cấp phiên bản
  - Admin muốn tích hợp Xử phạt hành chính vào <https://sohaatalieu.online/>
  - Admin muốn thêm 1 cột map danh mục hàng mục trong Báo giá xe
- Done:**
  - Admin muốn thêm trường mới ở Phát hành thẻ trên Dev.Database

Hình 2. 54. Thiết kế giao diện màn hình Chi tiết sprint

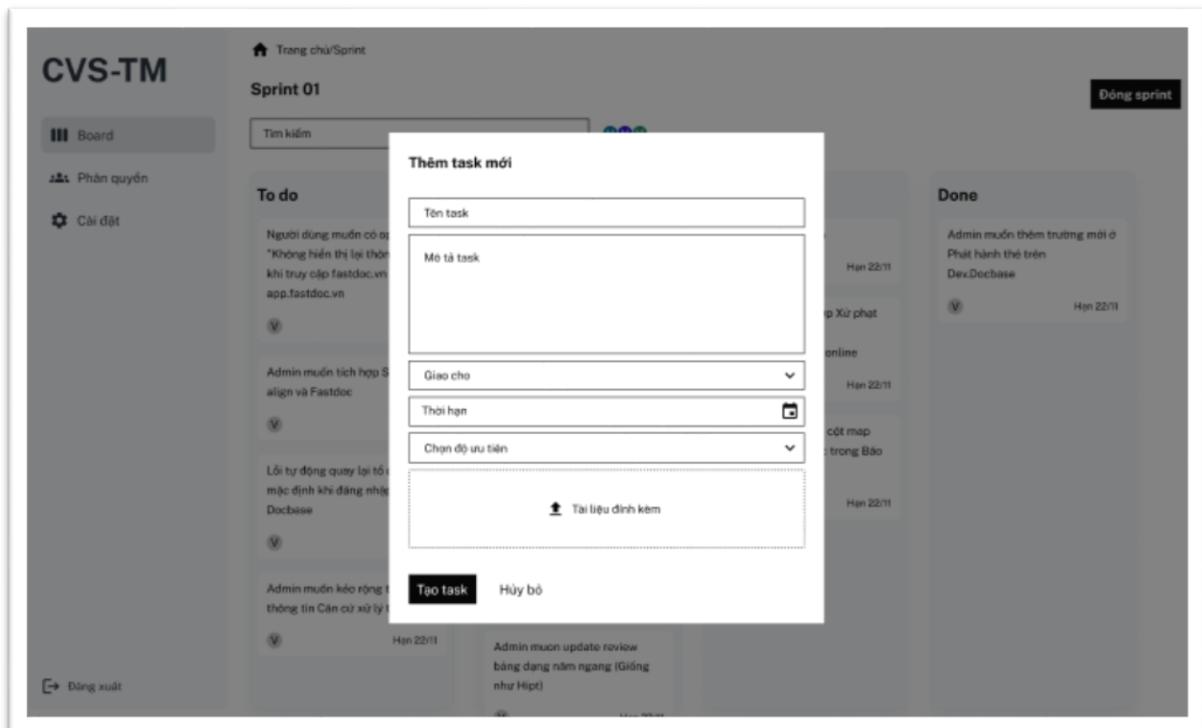
### 2.2.7.6. Màn hình Chi tiết task

The screenshot shows the CVS-TM software interface with a modal window open. The modal is titled "Tối ưu tốc độ load của Bảng khi review tài liệu" and is associated with the user "Đào Duy Anh". The modal contains a message about optimizing the table load speed during document review and a comment from the user.

The modal has a "Nhập comment" (Enter comment) input field and a "Gửi" (Send) button. The background of the main interface shows a board view with some tasks and a "Đóng sprint" button.

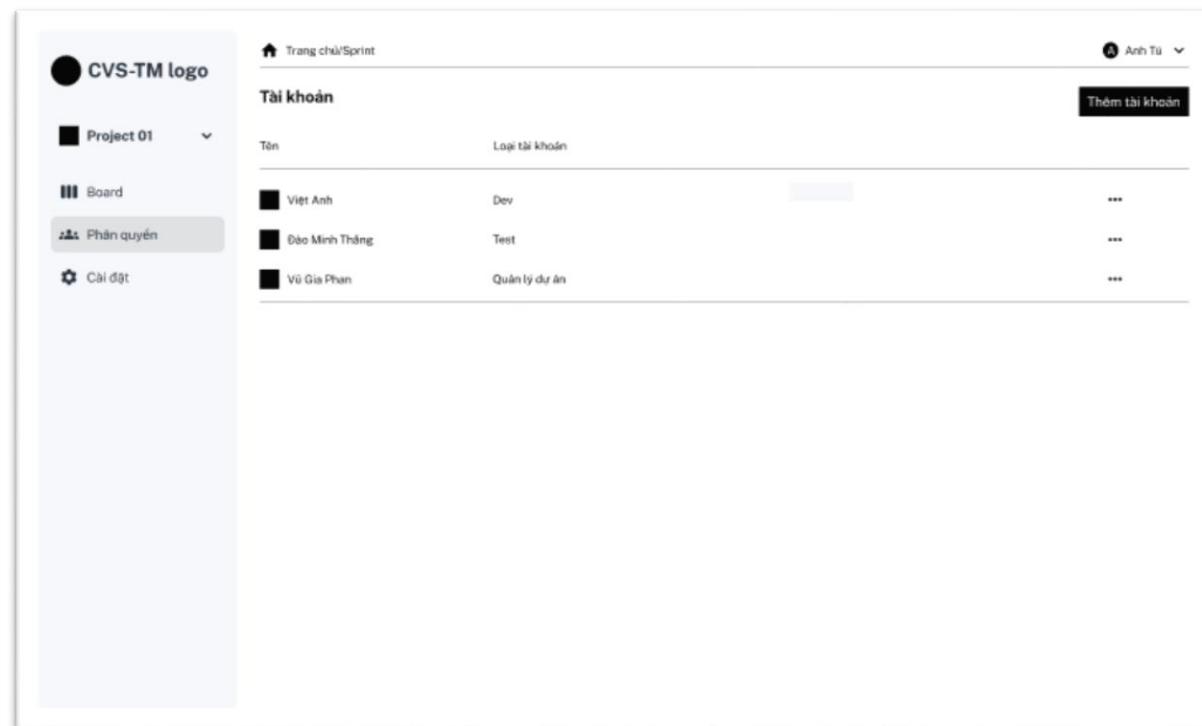
Hình 2. 55. Thiết kế giao diện màn hình Chi tiết task

### 2.2.7.7. Màn hình Thêm task



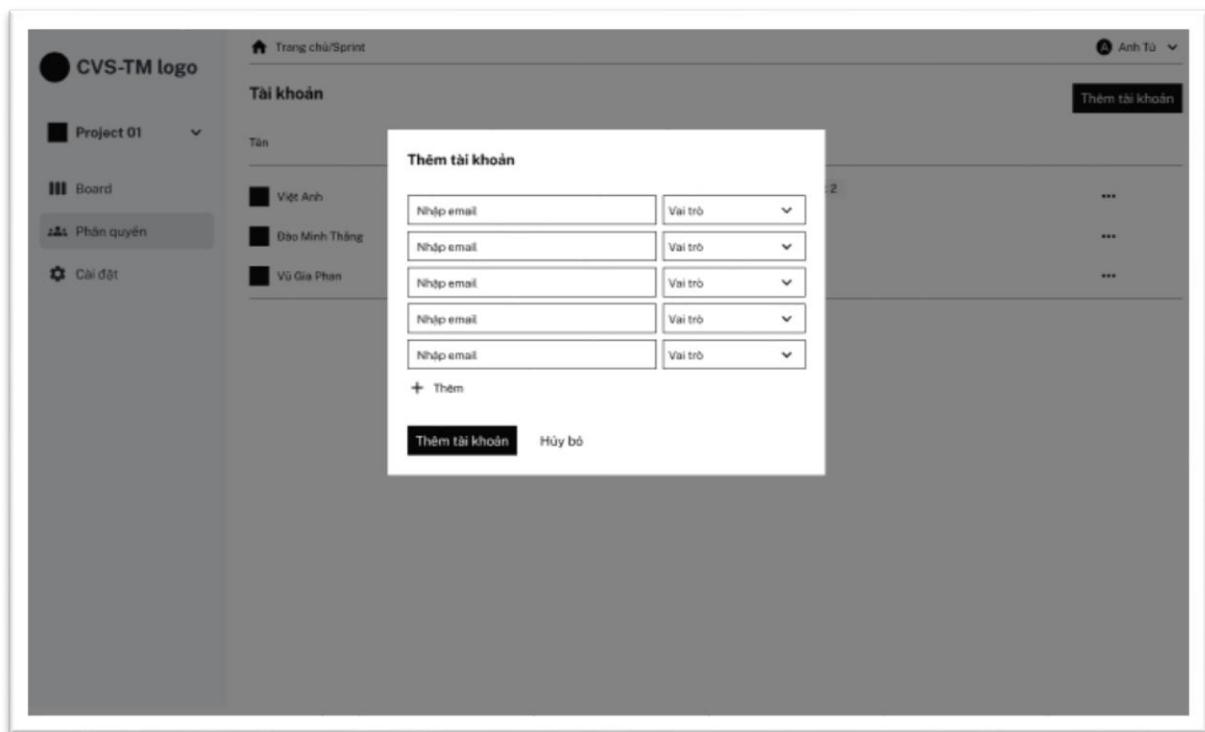
Hình 2. 56. Thiết kế giao diện màn hình Thêm task

### 2.2.7.8. Màn hình Quản lý tài khoản



Hình 2. 57. Thiết kế giao diện màn hình Quản lý tài khoản

### 2.2.7.9. Màn hình Thêm tài khoản



Hình 2. 58. Thiết kế giao diện màn hình Thêm tài khoản

## CHƯƠNG 3. TRIỂN KHAI, KẾT QUẢ VÀ KIỂM THỦ

### 3.1. Triển khai hệ thống

#### 3.1.1. Phía back-end

- Công nghệ sử dụng: Node.js, Express, MySQL.
- Cài đặt môi trường: node.js version 18.
- Cấu hình: bổ sung thêm file “.env” và nội dung đã được gửi đính kèm.
- Triển khai:
  - o Truy cập thư mục gốc của dự án.
  - o Cài đặt các thư viện có trong file “package.json”: sử dụng lệnh “npm install”,
  - o Chạy back-end: “npm run start”.

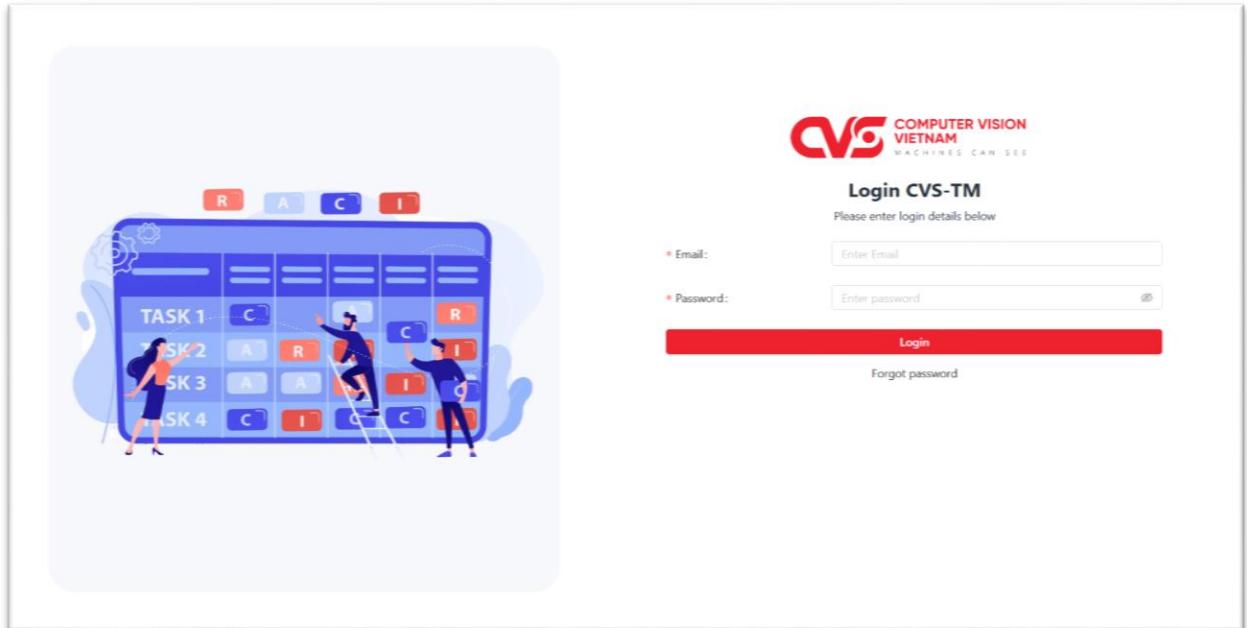
#### 3.1.2. Phía front-end

- Công nghệ sử dụng: ReactJS, SCSS, Vite-plugin, Yarn.
- Cài đặt môi trường: node.js version 18
- Cấu hình: bổ sung thêm file “.env” và nội dung đã được gửi đính kèm.
- Triển khai:
  - o Truy cập thư mục gốc của dự án
  - o Cài đặt các thư viện có trong file “package.json”: sử dụng lệnh “yarn install”.
  - o Chạy front-end: “yarn dev”.

### 3.2. Giao diện của hệ thống

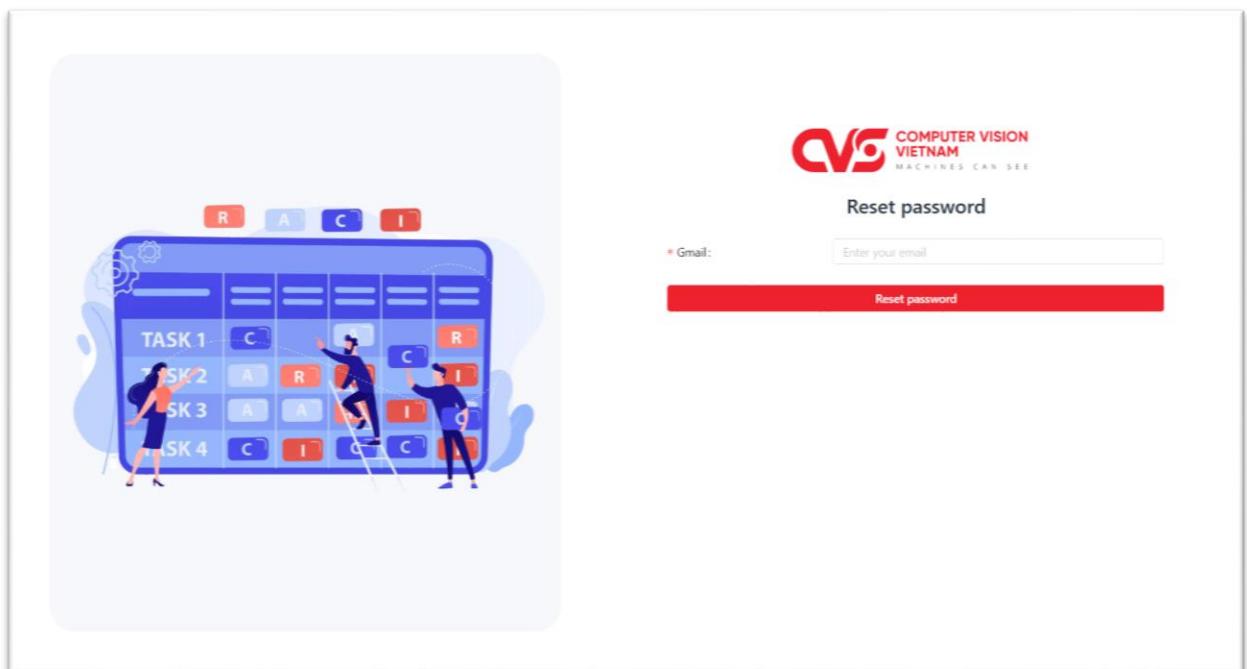
#### 3.2.1. Giao diện phía người dùng

##### 3.2.1.1. Giao diện Đăng nhập



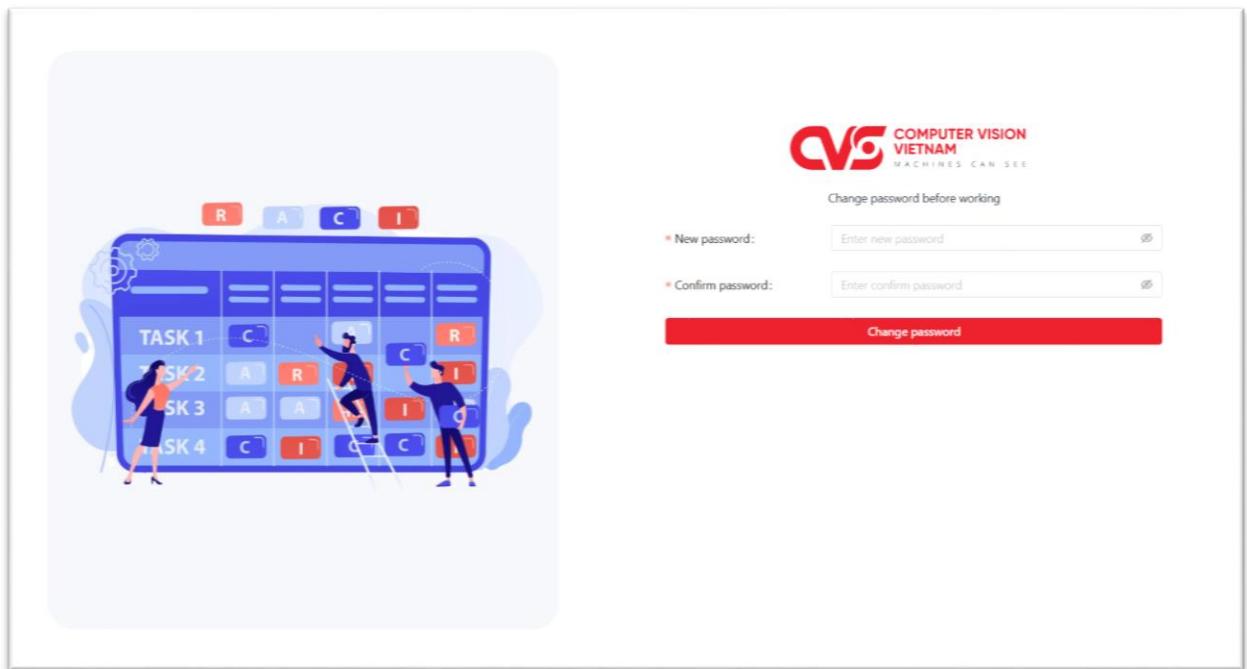
*Hình 3. 1. Giao diện đăng nhập*

##### 3.2.1.2. Giao diện Reset password



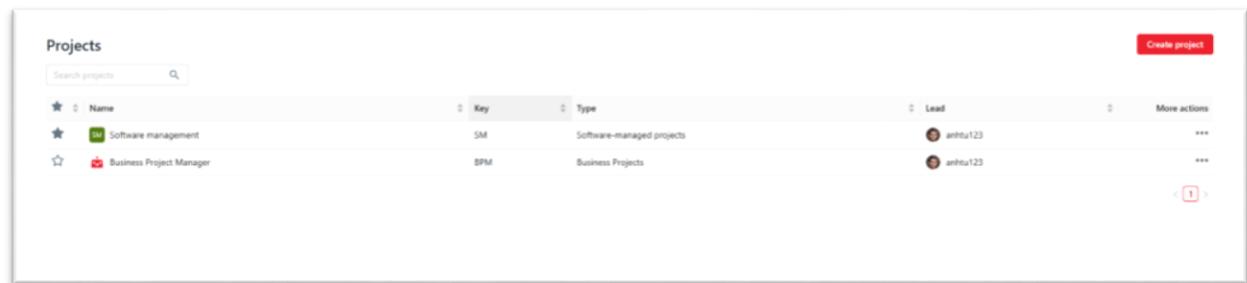
*Hình 3. 2. Giao diện reset mật khẩu*

### 3.2.1.3. Giao diện Thay đổi mật khẩu



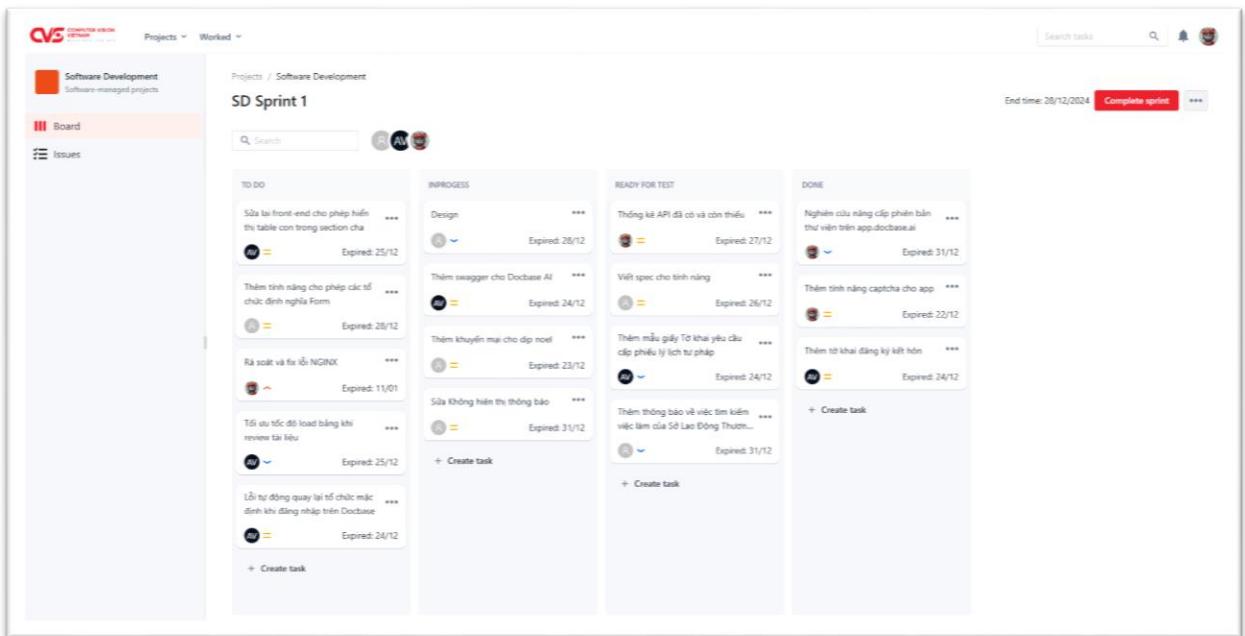
Hình 3. 3. Giao diện thay đổi mật khẩu

### 3.2.1.4. Giao diện Trang chủ



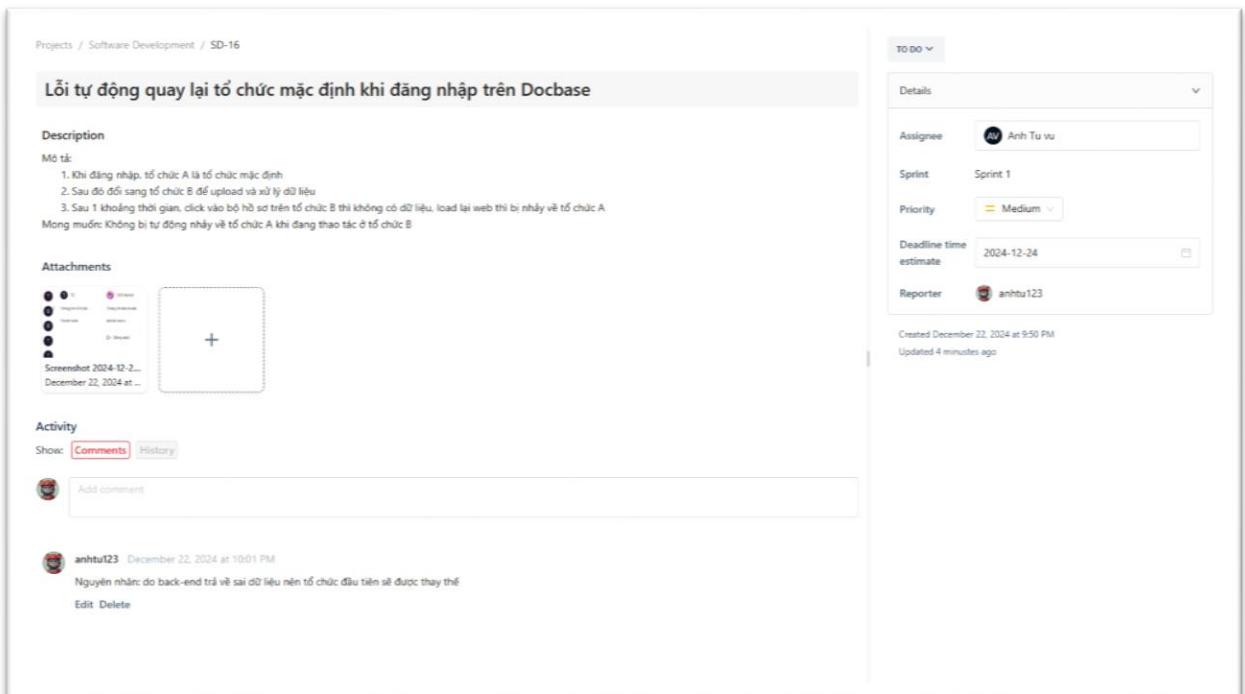
Hình 3. 4. Giao diện trang chủ

### 3.2.1.5. Giao diện Chi tiết sprint



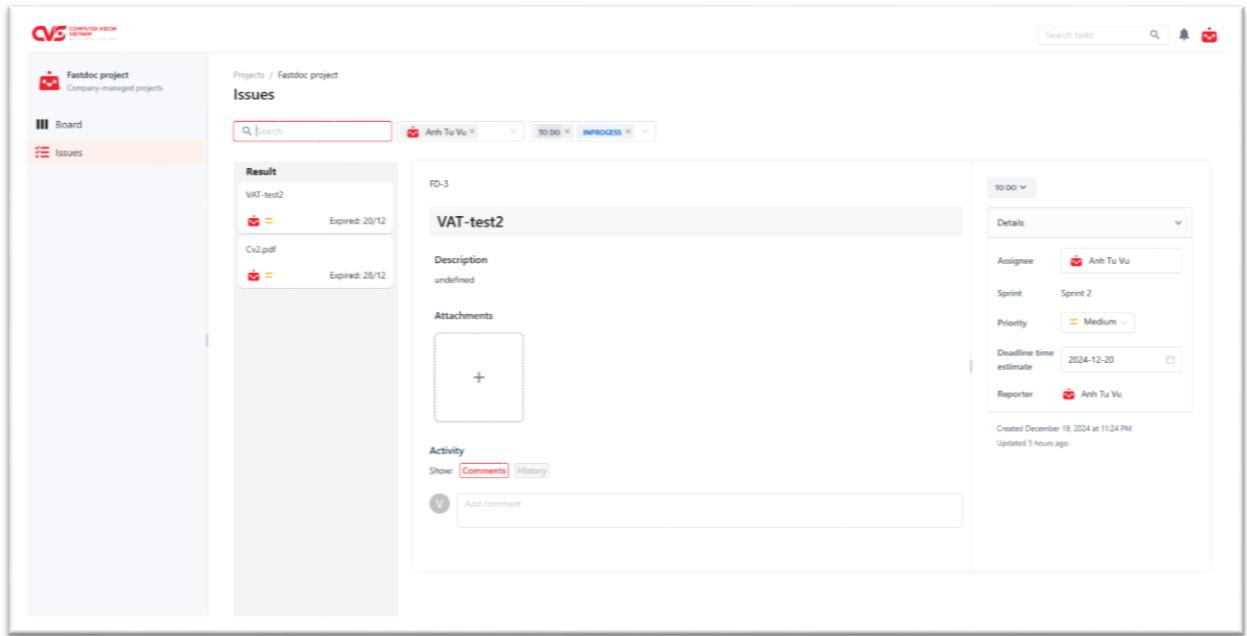
Hình 3. 5. Giao diện chi tiết sprint

### 3.2.1.6. Giao diện Chi tiết task



Hình 3. 6. Giao diện chi tiết task

### 3.2.1.7. Giao diện filter task



Hình 3. 7. Giao diện filter task

### 3.2.1.8. Giao diện Form tạo project

The screenshot shows a modal dialog titled "Create project". At the top right is a close button (X). In the center is a dashed square placeholder labeled with a plus sign (+) and the word "Icon". Below this are four input fields: "Project name", "Project key", "Company-managed projects" (with a dropdown arrow), and "Description" (with a text area and a scroll bar). At the bottom are two buttons: a red "Create" button and a white "Cancel" button.

Hình 3. 8. Giao diện tạo project

### 3.2.1.9. Giao diện Form tạo sprint

The screenshot shows a modal window titled "Create sprint". Inside, a message says "Required fields are marked with an asterisk \*". There are two date input fields: "Start date" and "End date", each with a "Select date" button. Below the inputs are two buttons: a red "Create" button and a white "Cancel" button.

Hình 3. 9. Giao diện form tạo sprint

### 3.2.1.10. Giao diện Form tạo task

**Create task** X

Required fields are marked with an asterisk \*

\* Task name:

Description:

\* Assignee: A anhtu123

\* Expried date:  calendar icon

Priority:  ▼









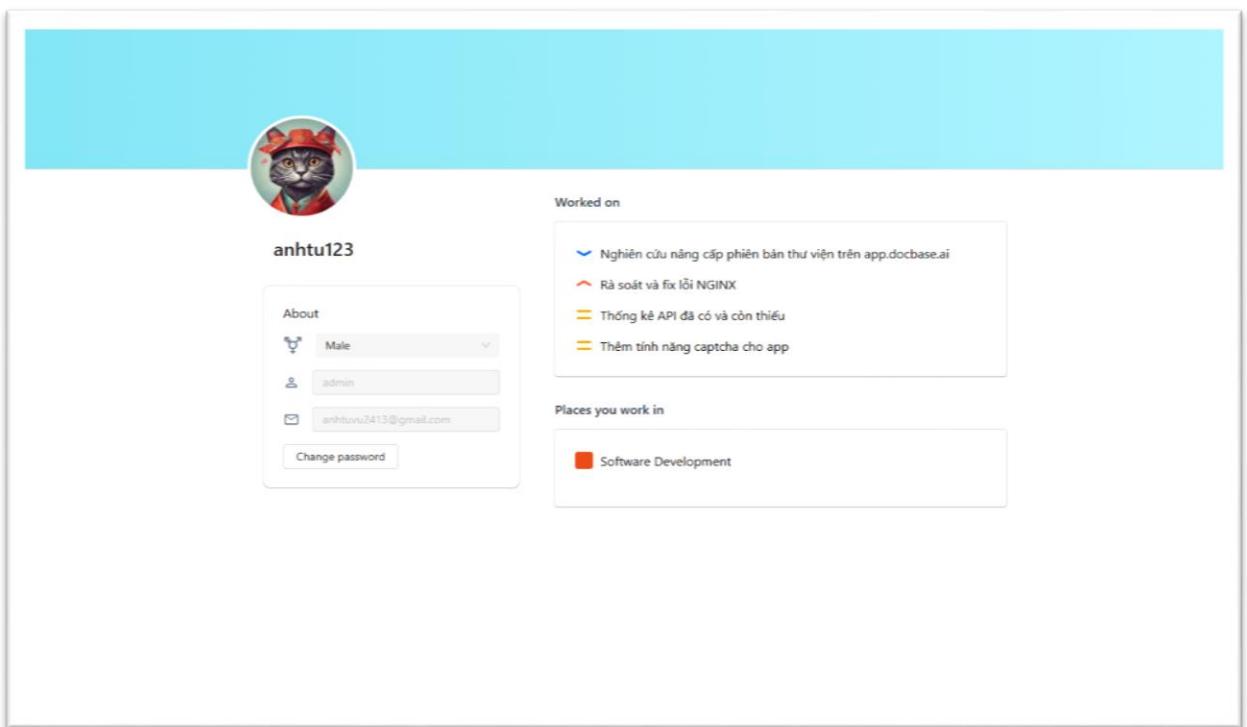


↑ **Attachments**

Create Cancel

Hình 3. 10. Giao diện form tạo task

### 3.2.1.11. Giao diện Profile



Hình 3. 11. Giao diện Profile

### 3.2.2. Giao diện phía quản trị

#### 3.2.2.1. Giao diện Quản lý người dùng

		Projects / Accounts							
		Search users		Status				Create user	
Role assignment		Name	Gmail	Role	Status		More actions		
Project type		Anh Tu vu	anhtu24.dev@gmail.com	Admin	On		***		
		anhtu123	anhtu2413@gmail.com	Admin	On		***		

Hình 3. 12. Giao diện quản lý người dùng

### 3.2.2.2. Giao diện Form thêm người dùng

Create user

User name

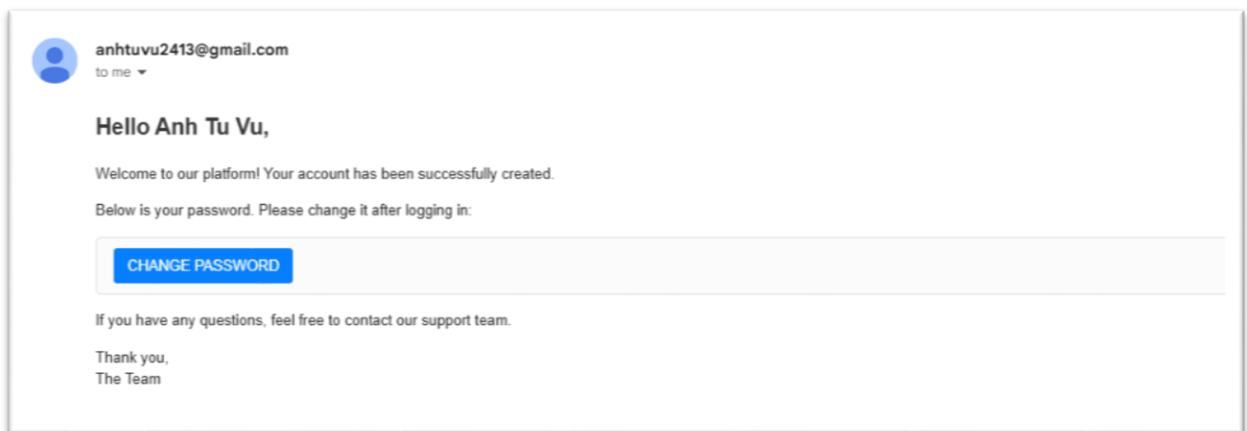
Gmail

Select a role

Create Cancel

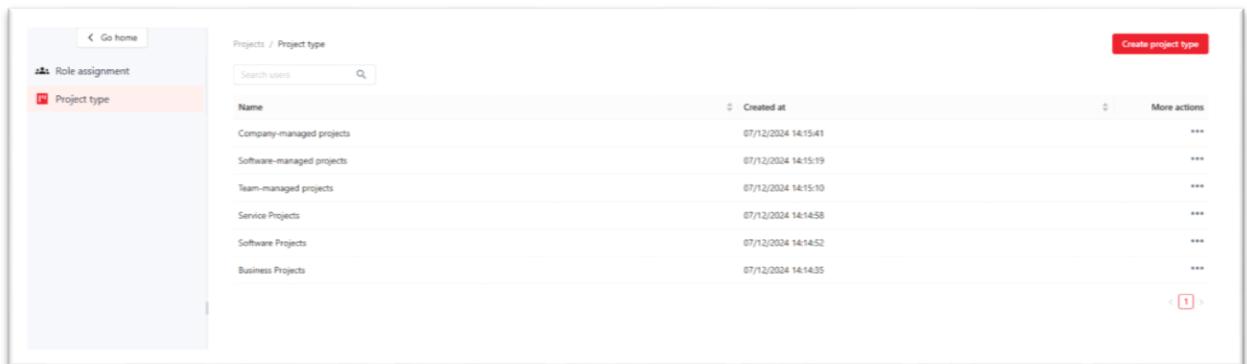
Hình 3. 13. Giao diện thêm user

### 3.2.2.3. Xác nhận user



Hình 3. 14. Giao diện xác nhận gmail

### 3.2.2.4. Giao diện quản lý loại project



Hình 3. 15. Giao diện quản lý loại project

## 3.3. Kiểm thử chức năng hệ thống

### 3.3.1. Kiểm thử chức năng người dùng

Bảng 3. 1. Danh sách test case chức năng hệ thống

STT	Case	Đầu vào	Đầu ra mong muốn	Kết quả
1	Kiểm tra chức năng đăng nhập	Nhập tài khoản & mật khẩu hợp lệ	Đăng nhập thành công, chuyển sang trang Trang chủ.	Pass
		Nhập tài khoản & mật khẩu không hợp lệ	Đăng nhập không thành công	Pass
2	Kiểm tra chức năng tạo tài khoản	Nhập thông tin đăng ký hợp lệ	Thành công và gửi mail tới gmail đã nhập	Pass
		Nhập thông tin gmail bị trùng	Hệ thống thông báo đăng ký không hợp lệ	Pass

3	Kiểm tra chức năng đăng xuất	Đăng xuất tài khoản	Đăng xuất khỏi tài khoản, chuyển sang trang đăng nhập	Pass
4	Quên mật khẩu	Nhập gmail đã đăng ký	Thông báo thành công và gửi mail xác nhận về hòm thư	Pass
		Nhập sai gmail	Không tìm thấy gmail.	Pass
5	Thay đổi mật khẩu sau khi xác nhận (Lần đầu hoặc reset mật khẩu)	Nhập thông tin mật khẩu mới và xác nhận lại mật khẩu	Thay đổi thành công và về trang đăng nhập	Pass
		Nhập sai định dạng hoặc không trùng	Báo lỗi và không cho phép thay đổi mật khẩu	Pass
6	Kiểm tra chức năng đăng xuất	Click đăng xuất	Xóa token của user khỏi hệ thống, trả về trang login	Pass
7	Kiểm tra chức năng thay đổi thông tin user	Thực hiện thay đổi thông tin user	Cập nhật trên hệ thống.	Pass
8		Nhập đầy đủ thông tin bắt buộc (Có	Tạo thành công dự án và hiển thị lên danh sách dự án	Pass

	Kiểm tra chức năng tạo project	thêm bỏ thông tin không bắt buộc)		
		Nhập thiếu thông tin hoặc bị trùng key	Thông báo lỗi và không cho phép tạo	Pass
9	Kiểm tra chức năng tạo sprint	Nhập ngày bắt đầu và kết thúc	Tự động sinh các thông tin cần thiết và hiển thị thông tin lên màn hình, chuyển sang màn hình chi tiết task	Pass
10	Kiểm tra chức năng hoàn thành sprint	Chuyển trạng thái sprint	Chuyển trạng thái sprint và trở về màn hình active sprint	Pass
11	Kiểm tra chức năng tạo task	Nhập thông tin cần thiết để tạo task	Tạo thành công và hiển thị task lên giao diện	Pass
12	Kiểm tra chức năng xóa task	Chọn option xóa task và xác nhận	Xóa khỏi hệ thống và xóa task khỏi giao diện hiển thị	Pass
13	Xem chi tiết task	Click vào 1 task bất kỳ	Lấy tất cả thông tin task và chuyển sang màn hình chi tiết task	Pass
14	Kiểm tra chức năng thay đổi thông tin task (assignee,	Thay đổi thông tin	Cập nhật thông tin trên hệ thống và hiển thị lên giao diện.	Pass

	priority, estimate time,...)			
15	Kiểm tra chức năng comment trong task	Nhập comment và click save	Lưu thông tin comment và hệ thống và hiển thị lên giao diện	Pass
16	Kiểm tra chức năng comment	Click edit và sửa thông tin, click save	Lưu thông tin mới của comment và hiển thị lên giao diện	Pass
		Click delete và xác nhận	Xóa thông tin comment khỏi hệ thống và trên giao diện	Pass
17	Kiểm tra chức năng upload file đính kèm trong task	Click icon thêm file đính kèm và thực hiện chọn file	Lưu thông tin file và hiển thị lên màn hình	Pass
18	Kiểm tra chức năng kéo thả status và vị trí của task	Thực hiện kéo task tới vị trí hoặc status mong muốn và thả	Lưu thông tin vị trí/status mới của task và hiển thị lên màn hình	
19	Kiểm tra chức năng search task	Nhập search key	Hệ thống trả ra những kết quả phù hợp và hiển thị lên giao diện	Pass

### 3.3.2. Báo cáo kiểm thử

*Bảng 3. 2. Báo cáo kiểm thử*

Người thực hiện	Vũ Anh Tú
Ngày báo cáo	18/12/2024
Tổng số trường hợp kiểm thử	19
Số trường hợp chưa kiểm thử	0
Số trường hợp kiểm thử đạt	19
Số trường hợp kiểm thử không đạt	0

## KẾT LUẬN

† Những kết quả đạt được

○ Về công nghệ:

- Hiểu được quá trình thiết kế một website thực tế, đi từ những bước cơ bản: khảo sát, phân tích, thiết kế, thực hiện, kiểm thử, ...
- Xây dựng thành công website bán điện thoại di động □ Năm được các kiến thức xây dựng phần mềm.
- Sử dụng thành thạo các công cụ hỗ trợ: Postman, Webstorm, Draw.io, CASE studio 2

○ Về chương trình:

- Cho phép doanh nghiệp theo dõi và quản lý công việc...
- Cho phép người dùng theo dõi chéo và trao đổi công việc.

† Hướng phát triển

- Kết nối và quản lý công việc ở mọi nơi với thiết bị và internet.

## TÀI LIỆU THAM KHẢO

- [1]. Nguyễn Thị Thanh Huyền, Ngô Thị Bích Thúy và Phạm Kim Phượng, *Giáo Trình Phân Tích Thiết Kế Hệ Thống*, Nhà xuất bản Giáo dục Việt Nam, (2011).
- [2]. Nguyễn Trung Phú (chủ biên), Trần Thị Phương Nhung và Đỗ Thị Minh Nguyệt, *Giáo Trình Thiết Kế Web*, Nhà xuất bản Thông Kê, (2019).
- [3]. Ethan Brown (2014), *Web Development with Node and Express*, Published by O'Reilly Media Inc.
- [4]. Trang chủ react: <https://react.dev/learn> (Lần truy cập cuối: 11/12/2024)
- [5]. Trang chủ node.js: <https://nodejs.org/en> (Lần truy cập cuối: 18/12/2024)
- [6]. Trang chủ express: <https://expressjs.com> (Lần truy cập cuối: 18/12/2024)
- [7]. Antd: <https://ant.design/> (Lần truy cập cuối: 19/12/2024)