

MỤC LỤC

LỜI CẢM ƠN.....	7
LỜI CAM ĐOAN	8
CHƯƠNG 1. GIỚI THIỆU.....	9
1.1. Đặt vấn đề.....	9
1.2. Các đóng góp của đồ án	9
1.3. Bố cục của đồ án.....	10
CHƯƠNG 2. KIẾN THỨC NỀN TẢNG.....	11
2.1. Cơ sở lý thuyết.....	11
2.1.1. Phân tích thiết kế hướng đối tượng.....	11
2.1.2. Cơ sở dữ liệu quan hệ	11
2.1.3. Ngôn ngữ được sử dụng	12
2.1.4. Công nghệ được sử dụng để phát triển hệ thống phía server	15
2.1.5. Công nghệ được sử dụng để phát triển phía client	19
2.1.6. Những đặc trưng của dự án thuộc lĩnh vực công nghệ thông tin	20
2.2. Công cụ sử dụng.....	22
2.2.1. Visual Studio 2017.....	22
2.2.2. SQL Server 2014	22
CHƯƠNG 3. PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG	23
3.1. Khảo sát hệ thống.....	23
3.1.1. Khảo sát hiện trạng hệ thống	23
3.1.2. Kết quả khảo sát về các hệ thống quản lý dự án trên thế giới.....	29
3.1.3. Kết luận	33
3.2. Yêu cầu cho hệ thống	34
3.3. Mô tả bài toán	35
3.4. Phân tích thiết kế.....	39
3.4.1. Xác định các tác nhân (Actor) và chức năng (Usecase)	39
3.4.2. Biểu đồ ca sử dụng (Usecase Diagram)	41
3.4.3. Biểu đồ trình tự.....	69
3.4.4. Biểu đồ hoạt động	92
3.4.5. Biểu đồ lớp	116
3.4.6. Thiết kế cơ sở dữ liệu vật lý	118

CHƯƠNG 4. XÂY DỰNG CHƯƠNG TRÌNH	123
 4.1. Xây dựng hệ thống	123
4.1.1. Giao diện chương trình	123
4.1.2. Một số mã code	132
 4.2. Cài đặt hệ thống	138
CHƯƠNG 5. KẾT LUẬN	139
TÀI LIỆU THAM KHẢO	141

DANH MỤC HÌNH ẢNH

Hình 2.1.3. a. Mô hình Entity Framework	17
Hình 2.1.3. b. Mô hình Repository.....	18
Hình 2.1.6. a. Mô hình quản lý dự án CNTT.....	21
Hình 3.1.2. a. Hình ảnh minh họa mô hình quản lý của Asana	31
Hình 3.3. a. Sơ đồ mô tả quy trình làm việc của quản lý dự án.....	36
Hình 3.3. b. Sơ đồ mô tả quá trình team lead tạo và gửi yêu cầu.....	36
Hình 3.3. c. Sơ đồ mô tả quá trình team member tạo và gửi yêu cầu.....	37
Hình 3.3. d. Sơ đồ mô tả quá trình những người theo dõi dự án tạo và gửi yêu cầu.....	37
Hình 3.3. e. Sơ đồ mô hình quản lý công việc.....	38
Hình 3.4.2. a. Biểu đồ usecase tổng quát.....	41
Hình 3.4.2. b. Biểu đồ usecase chi tiết đăng ký	42
Hình 3.4.2. c. Biểu đồ chi tiết usecase đăng nhập.....	43
Hình 3.4.2. d. Biểu đồ chi tiết usecase quản lý tài khoản cá nhân.....	44
Hình 3.4.2. e. Biểu đồ chi tiết usecase quản lý tài khoản trong hệ thống	45
Hình 3.4.2. f. Biểu đồ chi tiết usecase quản lý chức vụ trong ngành CNTT.....	47
Hình 3.4.2. g. Biểu đồ chi tiết usecase quản lý loại công việc.....	49
Hình 3.4.2. h. Biểu đồ chi tiết usecase quản lý dự án trong hệ thống.....	51
Hình 3.4.2. i. Biểu đồ chi tiết usecase quản lý dự án cá nhân	52
Hình 3.4.2. j. Biểu đồ chi tiết usecase quản lý module	54
Hình 3.4.2. k. Biểu đồ chi tiết usecase quản lý thành viên trong dự án	55
Hình 3.4.2. l. Biểu đồ chi tiết usecase kê tiền độ dự án	57
Hình 3.4.2. m. Biểu đồ chi tiết usecase quản lý yêu cầu cần xử lý.....	59
Hình 3.4.2. n. Biểu đồ chi tiết usecase quản lý công việc.....	60
Hình 3.4.2. o. Biểu đồ chi tiết usecase thực hiện công việc được giao	62
Hình 3.4.2. p. Biểu đồ chi tiết usecase quản lý yêu cầu cá nhân	63
Hình 3.4.2. q. Biểu đồ chi tiết usecase tìm kiếm	65
Hình 3.4.2. r. Biểu đồ chi tiết usecase xem	67
Hình 3.4.5. a. Biểu đồ lớp.....	116

<i>Hình 4.1.1. a. Giao diện đăng nhập.....</i>	123
<i>Hình 4.1.1. b. Giao diện đăng ký.....</i>	124
<i>Hình 4.1.1. c. Giao diện trang quản lý chức vụ</i>	124
<i>Hình 4.1.1. d. Giao diện trang quản lý loại công việc.....</i>	125
<i>Hình 4.1.1. e. Giao diện trang quản lý loại phương án giải quyết công việc.....</i>	125
<i>Hình 4.1.1. f. Giao diện trang quản lý dự án trong hệ thống của người QTHT</i>	126
<i>Hình 4.1.1. g. Giao diện trang quản lý tài khoản.....</i>	126
<i>Hình 4.1.1. h. Giao diện trang danh sách dự án người dùng tham gia.....</i>	127
<i>Hình 4.1.1. i. Giao diện thông tin dự án</i>	128
<i>Hình 4.1.1. j. Giao diện trang danh sách công việc.yêu cầu của người sử dụng</i>	128
<i>Hình 4.1.1. k. Giao diện trang thống kê công việc theo bảng Kanban.....</i>	129
<i>Hình 4.1.1. l. Giao diện trang thống kê công việc theo bảng Gantt.....</i>	130
<i>Hình 4.1.1. m. Giao diện trang quản lý thông tin dự án.....</i>	130
<i>Hình 4.1.1. n. Giao diện trang thống kê tiến độ dự án</i>	131
<i>Hình 4.1.1. o. Giao diện trang quản lý module trong dự án.....</i>	132

DANH MỤC BẢNG BIỂU

Bảng 3.1.1. a. Bảng câu hỏi phỏng vấn	24
Bảng 3.4.1. a. Bảng tác nhân và chức năng	39
Bảng 3.4.2. a. Kịch bản usecase đăng ký	42
Bảng 3.4.2. b. Kịch bản usecase đăng nhập.....	43
Bảng 3.4.2. c. Kịch bản usecase quản lý tài khoản cá nhân.....	44
Bảng 3.4.2. d. Kịch bản usecase quản lý tài khoản trong hệ thống	45
Bảng 3.4.2. e. Kịch bản usecase quản lý chức vụ trong ngành CNTT.....	47
Bảng 3.4.2. f. Kịch bản usecase quản lý loại công việc.....	49
Bảng 3.4.2. g. Kịch bản usecase quản lý dự án trong hệ thống.....	51
Bảng 3.4.2. h. Kịch bản usecase quản lý dự án cá nhân	52
Bảng 3.4.2. i. Kịch bản usecase quản lý module	54
Bảng 3.4.2. j. Kịch bản usecase quản lý thành viên trong dự án	56
Bảng 3.4.2. k. Kịch bản usecase thống kê tiến độ dự án.....	57
Bảng 3.4.2. l. Kịch bản usecase quản lý yêu cầu cần xử lý	59
Bảng 3.4.2. m. Kịch bản usecase quản lý công việc	60
Bảng 3.4.2. n. Kịch bản usecase thực hiện công việc được giao	62
Bảng 3.4.2. o. Kịch bản usecase quản lý yêu cầu cá nhân.	63
Bảng 3.4.2. p. Kịch bản usecase tìm kiếm.....	65
Bảng 3.4.2. q.Kịch bản usecase xem.....	68
Bảng 3.4.6. a. Bảng thông tin người dùng (UserInfo)	118
Bảng 3.4.6. b. Bảng chức vụ (JobRole).....	118
Bảng 3.4.6. c. Bảng dự án (Project)	119
Bảng 3.4.6. d. Bảng thành viên (RoleInProject)	119
Bảng 3.4.6. e. Bảng module (Module)	120
Bảng 3.4.6. f. Bảng loại công việc (TaskType).....	120
Bảng 3.4.6. g. Bảng công việc / yêu cầu (Task)	120
Bảng 3.4.6. i. Bảng nhật ký dự án (ProjectLog).....	121
Bảng 3.4.6. j. Bảng phương án giải quyết công việc (Solution).....	121
Bảng 3.4.6. k. Bảng loại giải quyết (ResolveType).....	122
Bảng 3.4.6. l. Bảng loại trạng thái (LookupStatus).....	122

DANH MỤC CỤM TỪ VIẾT TẮT & THUẬT NGỮ

stt	Thuật ngữ	Ý nghĩa
1	CNTT	Công nghệ thông tin
2	CSDL	Cơ sở dữ liệu
3	PM	Chức vụ quản lý dự án trong ngành công nghệ thông tin
4	Team Lead	Đóng vai trò lãnh đạo một phần dự án, tham gia trực tiếp vào việc phát triển phần mềm.
5	Team Member	Thành viên trong dự án
6	Repository	Là một loại mô hình triển khai code trong phát triển phần mềm
5	ACID	Nguyên tắc khi thiết kế cơ sở dữ liệu quan hệ
6	Watcher	Người theo dõi, có thể là khách hàng hoặc người đại diện bên khoán ngoài.
7	QTHT	Người quản trị hệ thống
8	MVC	Mô hình lập trình Model-View-Controller

LỜI CẢM ƠN

Trong thời gian nghiên cứu, học tập tại khoa Công nghệ thông tin trường Đại học Công nghệ Giao thông vận tải, được sự giúp đỡ quý báu của quý thầy cô giáo, các bạn sinh viên cùng các doanh nghiệp là đối tác của nhà trường đã giúp em hoàn thiện đồ án tốt nghiệp. Trước tiên cho phép em được bày tỏ lời cảm ơn sâu sắc nhất tới các thầy cô trong khoa Công nghệ thông tin. Đặc biệt là thầy TS. **Lê Chí Luận**, giảng viên khoa Công nghệ thông tin, trường Đại học Công nghệ Giao thông vận tải, người trực tiếp hướng dẫn, nhận xét, giúp đỡ em trong suốt quá trình thực hiện và hoàn thành đồ án.

Đồng thời em cũng xin chân thành cảm ơn Ban giám hiệu nhà trường, các thầy cô trong khoa Công nghệ thông tin cùng các phòng ban nhà trường đã tạo điều kiện tốt nhất cho em trong suốt thời gian học tập và làm đồ án tốt nghiệp này.

Tiếp theo, em xin gửi lời cảm ơn đến các thành viên lớp 66DCHT21, những người bạn đã luôn ở bên cạnh động viên, tạo điều kiện thuận lợi giúp em hoàn thành tốt đồ án, bên cạnh đó em xin cảm ơn gia đình đã tạo điều kiện về vật chất cũng như tinh thần, động viên giúp em hoàn thành đồ án này.

Trong quá trình thực hiện làm đồ án còn có rất nhiều sai sót, em mong nhận được sự giúp đỡ nhiệt tình của thầy cô cũng như các bạn để hoàn thiện một cách tốt nhất.

Em xin chân thành cảm ơn!

Hà Nội, ngày .. tháng .. năm 2019

Sinh viên thực hiện

Trần Lương Ngọc Trâm

LỜI CAM ĐOAN

Em xin cam đoan **đồ án này** là công trình của riêng em, các kết quả có tính độc lập riêng, không sao chép bất kỳ tài liệu nào và chưa công bố nội dung này ở đâu. Các tài liệu trong đồ án được sử dụng trung thực, nguồn trích dẫn có chú thích rõ ràng, minh bạch, có tính kế thừa, phát triển từ các tài liệu, các website.

Em xin hoàn toàn chịu trách nhiệm về lời cam đoan danh dự của em.

Hà Nội, ngày .. tháng .. năm 2019

Sinh viên thực hiện

Trần Lương Ngọc Trâm

CHƯƠNG 1. GIỚI THIỆU

1.1. Đặt vấn đề

Với sự phát triển theo hướng công nghiệp hóa ở nước ta hiện nay, ngành công nghệ thông tin cũng theo đó mà triển một cách mạnh mẽ. Hàng năm, số lượng nhân lực trong ngành công nghệ thông tin dần được tăng lên nhằm phục vụ cho các dự án trong lĩnh vực công nghệ. Điều này chứng tỏ rằng các dự án công nghệ thông tin hiện nay đang dần được tăng lên về số lượng cũng như về quy mô. Do đó, bài toán quản lý một dự án hay nhiều dự án phần mềm đang nhận được rất nhiều sự quan tâm đến từ các tổ chức trong lĩnh vực này.

Tuy ngành công nghệ thông tin của nước ta hiện tại đang phát triển một cách nhanh chóng nhưng tại một số công ty công nghệ thông tin mới thành lập vẫn đang quản lý dự án theo hướng sử dụng các văn bản một cách thủ công (sử dụng word, excel,...). Điều này gây ra những bất cập và khó khăn cho những người quản lý dự án do không thể nắm bắt cụ thể được vai trò và công việc của mỗi thành viên trong dự án, không thể thống kê được tiến độ của dự án. Do đó, những người quản lý dự án công nghệ thông tin khó có thể vạch ra các kế hoạch phát triển dự án một cách phù hợp để hoàn thành dự án theo đúng chỉ tiêu.

Qua tìm hiểu, khảo sát và xin ý kiến từ các thầy cô trong khoa, em đã quyết định chọn đề tài “THIẾT KẾ VÀ XÂY DỰNG HỆ THỐNG QUẢN LÝ DỰ ÁN CÔNG NGHỆ THÔNG TIN” với mục đích tạo ra một công cụ hỗ trợ công việc quản lý dự án công nghệ thông tin. Nhờ đó, giúp đẩy nhanh quá trình sản xuất trong các dự án các doanh nghiệp công nghệ thông tin có quy mô vừa và nhỏ, đồng thời tiết kiệm chi phí và tăng lợi nhuận một cách đáng kể.

1.2. Các đóng góp của đồ án

Qua thời gian tìm hiểu và xây dựng đồ án “THIẾT KẾ VÀ XÂY DỰNG HỆ THỐNG QUẢN LÝ DỰ ÁN CÔNG NGHỆ THÔNG TIN”, em đã có thể xây dựng thành công chương trình với nhiều chức năng như: Quản lý thông tin dự án, quản lý module trong dự án, quản lý thành viên trong dự án,... Ngoài ra còn có các chức năng quản lý yêu cầu, trích xuất thống kê ra file excel, tự động gửi email công việc, thống kê tiến độ dự án,... Từ đó nắm chắc được các vấn đề liên quan tới HTML, CSS, Jquery, SQL Server và Javascript, được trau dồi kỹ năng khảo sát và phân tích thiết kế hệ thống theo yêu cầu thực tế.

1.3. Bố cục của đồ án

Phần còn lại của đồ án bao gồm những kiến thức nền tảng được sử dụng, thông tin khảo sát, phân tích thiết kế và cách xây dựng chương trình. Cụ thể trong chương 1 là phần giới thiệu. Trong đó, chương này sẽ giới thiệu qua về đề tài, đặt vấn đề và các đóng góp của đồ án.

Tiếp theo đó, chương 2 sẽ giới thiệu về kiến thức nền tảng bao gồm: công nghệ được sử dụng để phát triển phía server và công nghệ được sử dụng để phát triển ứng dụng phía khách hàng. Bên server, em sẽ áp dụng các công nghệ phổ biến trên thế giới như: ASP.NET MVC, Entity Framework,... Còn về phía giao diện khách hàng, em sẽ sử dụng Bootstrap và JQuery.

Chương 3 bao gồm thông tin khảo sát các ứng dụng quản lý dự án đã có trên thế giới và phân tích thiết kế theo phương pháp luận hướng đối tượng. Trong chương này, đồ án sử dụng các biểu đồ lớp, biểu đồ, trình tự và biểu đồ hoạt động nhằm có cái nhìn tổng quan và chi tiết hơn về hệ thống.

Áp dụng được các kiến thức được trình bày ở chương 3, trong chương 4 đồ án sẽ trình bày cách xây dựng chương trình và trình bày về sản phẩm hoàn thiện. Trong đó bao gồm giao diện chương trình, bố cục thư mục code. Và phần cuối là kết luận và hướng phát triển đề tài.

CHƯƠNG 2. KIẾN THỨC NỀN TẢNG

Chương 2 bao gồm kết quả tìm hiểu về các kiến thức nền tảng được áp dụng trong đồ án. Trong đó, phần cơ sở lý thuyết chiếm vai trò quan trọng nhất, sau đó sẽ là phần giới thiệu về công cụ được sử dụng để phát triển đồ án này.

2.1. Cơ sở lý thuyết

Để có thể xây dựng một website thì phân tích thiết kế, cơ sở dữ liệu và ngôn ngữ lập trình và mô hình lập trình là những phần không thể thiếu, vì vậy em đã nghiên cứu và tóm tắt những tài liệu và trọng điểm chính sau đây.

2.1.1. Phân tích thiết kế hướng đối tượng

Định nghĩa: Phân tích thiết kế hướng đối tượng là giai đoạn phát triển một mô hình chính xác và súc tích của vấn đề, có thành phần là các đối tượng và khái niệm đời thực, dễ hiểu đối với người sử dụng [2].

Ưu điểm: Một trong những ưu điểm quan trọng bậc nhất của phương pháp phân tích và thiết kế hướng đối tượng là tính tái sử dụng: Có thể tạo các thành phần (đối tượng) một lần và dùng chúng nhiều lần sau đó.

2.1.2. Cơ sở dữ liệu quan hệ

Cơ sở dữ liệu là một hệ thống các thông tin có cấu trúc, được lưu trữ trên các thiết bị lưu trữ nhằm thỏa mãn yêu cầu khai thác thông tin đồng thời của nhiều người sử dụng hay nhiều chương trình ứng dụng chạy cùng một lúc với những mục đích khác nhau.

Định nghĩa cơ sở dữ liệu quan hệ: Cơ sở dữ liệu quan hệ dựa trên mô hình dữ liệu quan hệ. Trong mô hình này, dữ liệu được tổ chức thành một hoặc nhiều bảng (hoặc "quan hệ") có các cột và hàng. Mỗi hàng được xác định bằng cách sử dụng một khóa duy nhất [2].

Những hàng này được gọi là bản ghi hoặc bộ dữ liệu. Các cột được gọi là các thuộc tính. Mỗi bảng/quan hệ biểu thị một "loại thực thể" (chẳng hạn như sản phẩm hoặc cá nhân). Các hàng biểu thị một cá thể của một loại thực thể. Các cột biểu thị các thuộc tính được gán cho cá thể đó.

Nguyên lý ACID của cơ sở dữ liệu quan hệ rất nổi tiếng, khiến nó tạo ra một sự khác biệt so với cơ sở dữ liệu truyền thống. Do đó, em quyết định sử dụng mô hình này trong việc thiết kế CSDL. ACID là từ viết tắt các chữ cái đầu của bốn từ tiếng Anh atomicity,

consistency, isolation, và durability. Chúng được coi là bốn thuộc tính quan trọng của một hệ quản trị cơ sở dữ liệu khi xử lý bất kỳ giao dịch nào. Nếu thiếu một trong những thuộc tính này thì tính toàn vẹn của cơ sở dữ liệu khó có thể được đảm bảo. Các tính chất ACID trong trường hợp này sẽ đảm bảo các giao dịch được thực hiện một cách đáng tin cậy:

- Tính nguyên tố (Atomicity). Một giao dịch có nhiều thao tác khác biệt thì hoặc là toàn bộ các thao tác hoặc là không một thao tác nào được hoàn thành..
- Tính nhất quán (Consistency). Một giao dịch hoặc là sẽ tạo ra một trạng thái mới và hợp lệ cho dữ liệu, hoặc trong trường hợp có lỗi sẽ chuyển toàn bộ dữ liệu về trạng thái trước khi thực thi giao dịch.
- Tính độc lập (Isolation). Một giao dịch đang thực thi và chưa được xác nhận phải bảo đảm tách biệt khỏi các giao dịch khác.
- Tính bền vững (Durability). Dữ liệu được xác nhận sẽ được hệ thống lưu lại sao cho ngay cả trong trường hợp hỏng hóc hoặc có lỗi hệ thống, dữ liệu vẫn đảm bảo trong trạng thái chuẩn xác [2].

Các cơ sở dữ liệu quan hệ phổ biến bao gồm Microsoft SQL Server, Oracle Database, MySQL và IBM DB2. Ngoài ra còn có các cơ sở dữ liệu quan hệ dựa trên đám mây như Dịch vụ cơ sở dữ liệu quan hệ của Amazon (RDS), Google Cloud SQL, IBM DB2, Cơ sở dữ liệu Microsoft Azure SQL và Dịch vụ đám mây cơ sở dữ liệu Oracle. Tuy nhiên, trong đề tài này, em sẽ sử dụng hệ quản trị Microsoft SQL Server [2].

2.1.3. Ngôn ngữ được sử dụng

a. HTML (*HyperText Markup Language*)

HTML (tiếng Anh, viết tắt cho HyperText Markup Language, hay là "Ngôn ngữ Đánh dấu Siêu văn bản", là một ngôn ngữ đánh dấu được thiết kế ra để tạo nên các trang web với các mẫu thông tin được trình bày trên World Wide Web. Cùng với CSS và JavaScript, HTML tạo ra bộ ba nền tảng kỹ thuật cho World Wide Web [1].

HTML được định nghĩa như là một ứng dụng đơn giản của SGML và được sử dụng trong các tổ chức cần đến các yêu cầu xuất bản phức tạp. HTML đã trở thành một chuẩn Internet do tổ chức World Wide Web Consortium (W3C) duy trì. Phiên bản chính thức mới nhất của HTML là HTML 4.01 (1999). Sau đó, các nhà phát triển đã thay thế nó bằng

XHTML. Hiện nay, HTML đang được phát triển tiếp với phiên bản HTML5 hứa hẹn mang lại diện mạo mới cho Web.

b. CSS

CSS (Cascading Style Sheets) là một ngôn ngữ dùng để định dạng cho các phần tử HTML, hạn chế tối thiểu việc làm rối mã HTML của trang Web bằng các thẻ quy định kiểu dáng (chữ đậm, chữ in nghiêng, chữ có gạch chân, chữ màu), khiến mã nguồn của trang Web được gọn gàng hơn, tách nội dung của trang Web và định dạng hiển thị, dễ dàng cho việc cập nhật nội dung. Do đó, CSS tạo ra các kiểu dáng có thể áp dụng cho nhiều trang Web, giúp tránh phải lặp lại việc định dạng cho các trang Web giống nhau [1].

c. Javascript

JavaScript, theo phiên bản hiện hành, là một ngôn ngữ lập trình thông dịch được phát triển từ các ý niệm nguyên mẫu. Ngôn ngữ này được dùng rộng rãi cho các trang web (phía người dùng) cũng như phía máy chủ (với Nodejs). Nó vốn được phát triển bởi Brendan Eich tại Hãng truyền thông Netscape với cái tên đầu tiên Mocha, rồi sau đó đổi tên thành LiveScript, và cuối cùng thành JavaScript. Giống Java, JavaScript có cú pháp tương tự C, nhưng nó gần với Self hơn Java. .js là phần mở rộng thường được dùng cho tập tin mã nguồn JavaScript [1].

Trên trình duyệt, rất nhiều trang web sử dụng JavaScript để thiết kế trang web động và một số hiệu ứng hình ảnh thông qua DOM. JavaScript được dùng để thực hiện một số tác vụ không thể thực hiện được với chỉ HTML như kiểm tra thông tin nhập vào, tự động thay đổi hình ảnh,... Ở Việt Nam, JavaScript còn được ứng dụng để làm bộ gõ tiếng Việt giống như bộ gõ hiện đang sử dụng trên trang Wikipedia tiếng Việt. Tuy nhiên, mỗi trình duyệt áp dụng JavaScript khác nhau và không tuân theo chuẩn W3C DOM, do đó trong rất nhiều trường hợp lập trình viên phải viết nhiều phiên bản của cùng một đoạn mã nguồn để có thể hoạt động trên nhiều trình duyệt. Một số công nghệ nổi bật dùng JavaScript để tương tác với DOM bao gồm Ajax và SPA.

d. C#

C# là một ngôn ngữ lập trình hướng đối tượng hiện đại được phát triển vào năm 2000 bởi Anders Hejlsberg tại Microsoft, với vị thế như là một đối thủ của Java. Java được tạo ra bởi một nhóm kỹ sư tại Sun (sau đó được mua lại bởi Oracle) vì họ không muốn Microsoft

thay đổi Java, do đó Microsoft chỉ còn cách tạo ra ngôn ngữ của riêng họ. C# đã phát triển nhanh chóng kể từ khi nó được ra mắt lần đầu tiên, với sự hỗ trợ to lớn từ Microsoft giúp nó đạt tạo được tiếng vang lớn. Và giờ đây nó trở thành một trong những ngôn ngữ lập trình phổ biến nhất trên thế giới. C# có rất nhiều ưu điểm nổi bật như:

- Gần gũi với các ngôn ngữ lập trình thông dụng như C++, Java, Pascal.
- Xây dựng dựa trên nền tảng của các ngôn ngữ lập trình mạnh nên thừa hưởng những ưu điểm của chúng.
- Cải tiến các khuyết điểm của C/C++ như con trỏ, hiệu ứng phụ,..
- Dễ dàng tiếp cận, dễ phát triển.

Nhược điểm lớn nhất của C# là chỉ chạy trên nền Windows và có cài .NET Framework. Thao tác đối với phần cứng yếu hơn so với ngôn ngữ khác, hầu hết phải dựa vào windows [1].

e. UML

UML là viết tắt của “Unified Modeling Language” – là ngôn ngữ mô hình hóa thống nhất, dùng để đặc tả, trực quan hóa và tư liệu hóa phần mềm hướng đối tượng, có phần chính bao gồm những ký hiệu hình học, được các phương pháp hướng đối tượng sử dụng để thể hiện và miêu tả các thiết kế của một hệ thống. Nó là một ngôn ngữ để đặc tả, trực quan hóa [2].

Diagram (Biểu đồ): Đồ thị biểu diễn đồ họa về tập các phần tử trong mô hình và mối quan hệ của chúng. Trong phiên bản mới nhất có 13 loại biểu đồ trong đó có 9 loại biểu đồ chính được sử dụng.

- ❖ *Use Case Diagram (Biểu đồ ca sử dụng):* Mô tả sự tương tác giữa các tác nhân ngoài và hệ thống thông qua các ca sử dụng.
- ❖ *Class Diagram (Biểu đồ lớp):* Mô tả cấu trúc tĩnh, mô hình khái niệm bao gồm các lớp đối tượng và các mối quan hệ của chúng trong hệ thống hướng đối tượng.
- ❖ *Object Diagram (Biểu đồ đối tượng):* Là một phiên bản của biểu đồ lớp và thường cũng sử dụng các ký hiệu như biểu đồ lớp.
- ❖ *Sequence diagram (Biểu đồ trình tự):* Thể hiện sự tương tác của các đối tượng với nhau.

- ❖ *Collaboration Diagram (Biểu đồ cộng tác)*: Tương tự như biểu đồ trình tự nhưng nhấn mạnh vào sự tương tác của các đối tượng trên cơ sở cộng tác với nhau bằng cách trao đổi các thông điệp để thực hiện các yêu cầu theo ngữ cảnh công việc.
- ❖ *State Diagram (Biểu đồ trạng thái)*: Thể hiện chu kỳ hoạt động của các đối tượng, của các hệ thống con và của cả hệ thống.
- ❖ *Activity Diagram (Biểu đồ hành động)*: Chỉ ra dòng hoạt động của hệ thống.
- ❖ *Component diagram (Biểu đồ thành phần)*: Chỉ ra cấu trúc vật lý của các thành phần trong hệ thống.
- ❖ *Deployment Diagram (Biểu đồ triển khai)*: Chỉ ra cách bố trí vật lý các thành phần theo kiến trúc được thiết kế của hệ thống.

2.1.4. Công nghệ được sử dụng để phát triển hệ thống phía server

a. ASP.NET Framework

ASP.NET MVC là một phần mềm mã nguồn mở của Microsoft, kết hợp các tính năng của kiến trúc MVC (Model-View-Controller). Hầu hết các ý tưởng và kỹ thuật được phát triển từ nền tảng Agile và các phần tốt nhất của nền tảng ASP.NET hiện có. [1]

ASP.NET MVC là một framework tuyệt vời hỗ trợ pattern MVC cho ASP.NET. Nếu bạn muốn hiểu ASP.NET MVC làm việc như thế nào, bạn cần phải có một sự hiểu biết rõ ràng về mô hình MVC. MVC là cụm từ viết tắt của Model-View-Controller, nó phân chia pattern của ứng dụng thành 3 phần - model, controller và view.

- Model giúp lưu trữ dữ liệu của ứng dụng và trạng thái của nó. Nó là một cơ sở dữ liệu hoặc cũng có thể chỉ là một tập tin XML.
- View được coi là một giao diện người dùng được sử dụng bởi khách truy cập trang web của bạn để nhìn thấy các dữ liệu. Các trang ASPX thường được sử dụng để hiển thị view trong các ứng dụng ASP.NET MVC.
- Controller chịu trách nhiệm xử lý các tương tác của người dùng với trang web. Nó được sử dụng để xác định loại view nào cần phải được hiển thị. Controller cũng được sử dụng cho mục đích giao tiếp với model.

Framework này là khá nhẹ và cung cấp khả năng kiểm thử, trong đó tích hợp với các tính năng hiện có của ASP.NET như xác thực (authentication) dựa trên membership và cả các master page.

b. Mô hình Repository kết hợp MVC

3-Layers là một kiến trúc kiểu client/server mà trong đó giao diện người dùng (UI-user interface), các quy tắc xử lý (BR-business rule hay BL-business logic), và việc lưu trữ dữ liệu được phát triển như những module độc lập, và hầu hết là được duy trì trên các nền tảng độc lập, và mô hình 3 tầng (3-tiers) được coi là một kiến trúc phần mềm và là một mẫu thiết kế. [3]

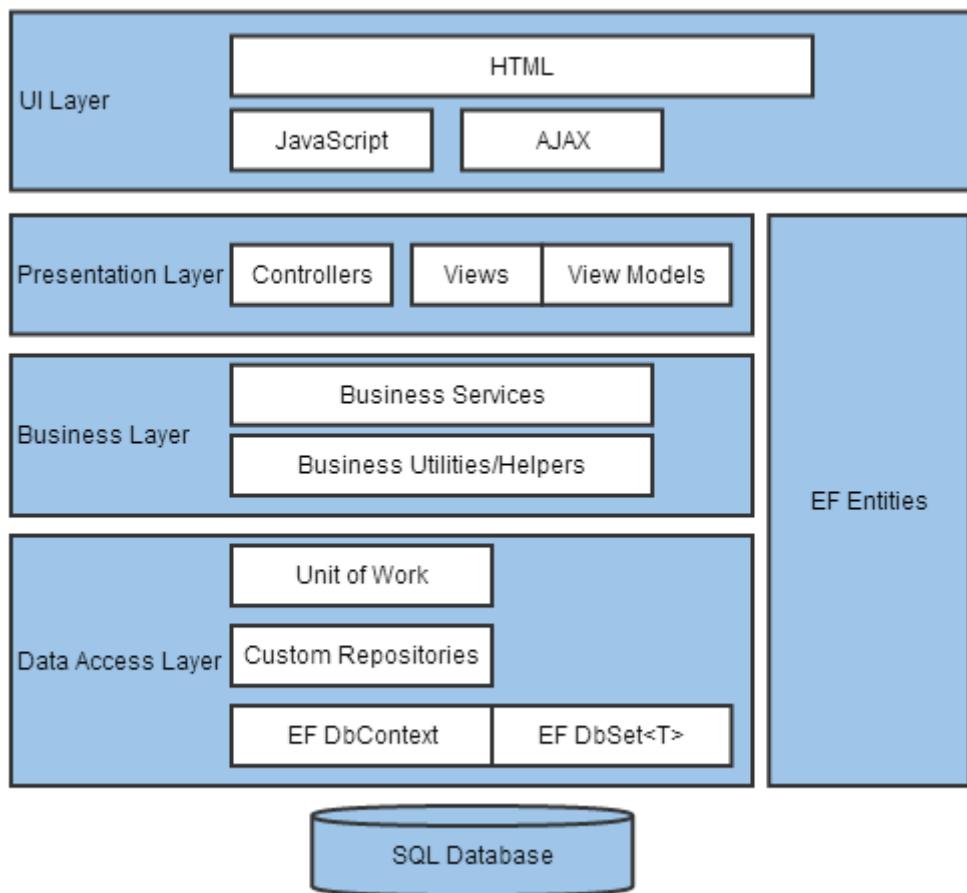
- Tầng Presentation: hiển thị các thành phần giao diện để tương tác với người dùng như tiếp nhận thông tin, thông báo lỗi, ...
- Tầng Business Logic: thực hiện các hành động nghiệp vụ của phần mềm như tính toán, đánh giá tính hợp lệ của thông tin, ... Tầng này còn di chuyển, xử lý thông tin giữa 2 tầng trên dưới.
- Tầng DAL: nơi lưu trữ và trích xuất dữ liệu từ các hệ quản trị CSDL hay các file trong hệ thống. Cho phép tầng Business logic thực hiện các truy vấn dữ liệu.

Dựa trên cơ sở của kiến trúc 3-layers, em tiến hành triển khai mô hình Repository kết hợp với MVC như sau:

- Tầng Presentation sẽ là project web ASP.NET. Trong đó bao gồm mô hình MVC. Mục đích của việc thiết kế này là giảm tải cho controller, tăng tốc độ request từ client. Ngoài ra, việc controller gắn chặt với view trong project web ASP.NET sẽ gây khó khăn trong việc Unit Test, nên khi đẩy phần code nghiệp vụ trong controller lên tầng Business sẽ có thể Unit Test một cách dễ dàng.
- Bổ sung Tầng Repository: Là lớp trung gian giữa tầng Business và tầng Data, giúp cho việc truy cập dữ liệu chặt chẽ và bảo mật hơn. Thông thường thì các phần truy xuất, giao tiếp với database nằm rải rác ở trong code, khi bạn muốn thực hiện một thao tác lên database thì phải tìm trong code cũng như tìm các thuộc tính trong bảng để xử lý. Điều này gây lãng phí thời gian và công sức rất nhiều. Với Repository design pattern, thì việc thay đổi ở code sẽ không ảnh hưởng quá nhiều công sức chúng ra chỉnh sửa.

c. Entity Framework

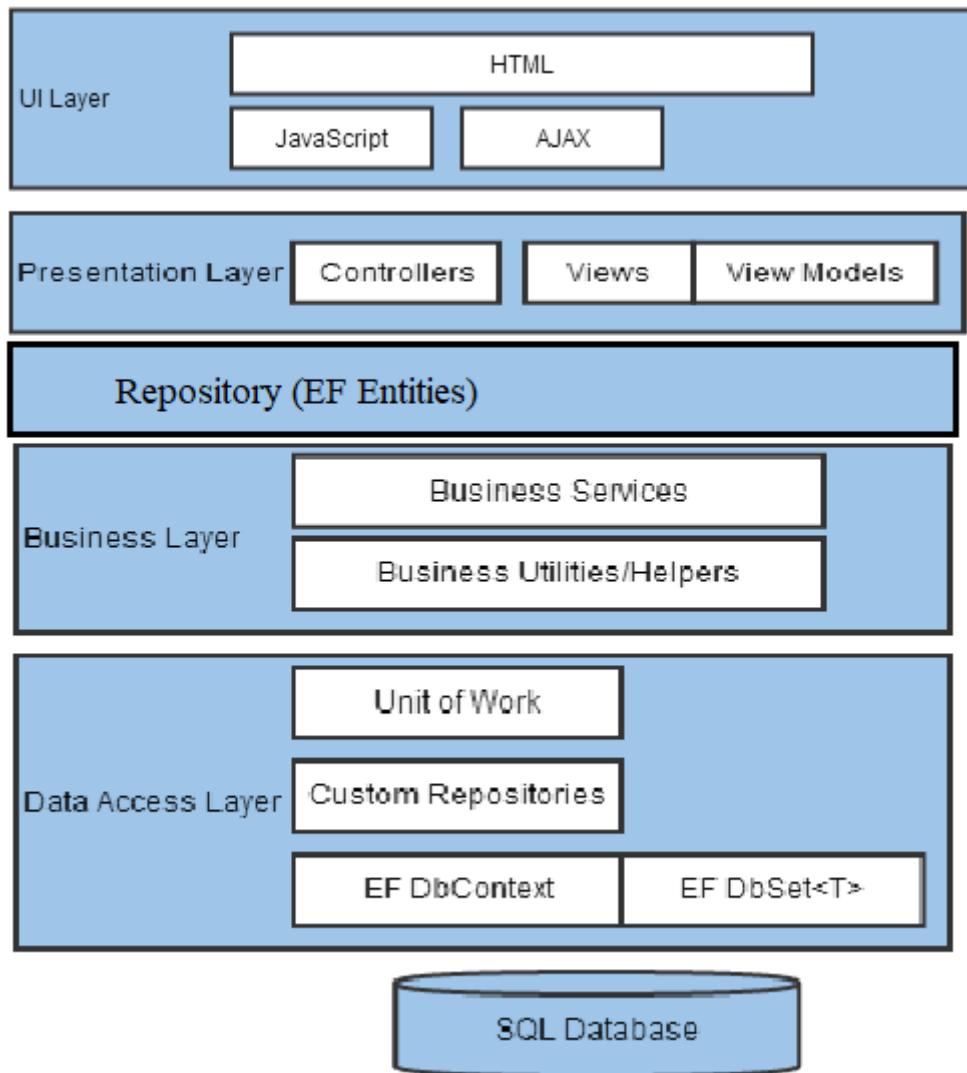
Entity Framework (EF) là một framework ánh xạ quan hệ đối tượng (ORM) dành cho ADO.NET, là 1 phần của .NET Framework. EF cho phép các nhà phát triển Web tương tác với dữ liệu quan hệ theo phương pháp hướng đối tượng đặc trưng. Lợi ích lớn nhất của EF là giúp lập trình viên giảm thiểu việc lập trình mã nguồn cần thiết để truy cập và tương tác với cơ sở dữ liệu. EF được Microsoft hỗ trợ phát triển lâu dài và bền vững, vì vậy EF là 1 framework mạnh nhất hiện nay để phát triển ứng dụng Web với sự hỗ trợ đồng đảo của các nhà phát triển Web [1].



Hình 2.1.3. a. Mô hình Entity Framework.

Trong hình trên, EF dường như có vị trí rất tự do, xuất hiện ở cả 3 tầng Data Access Layer, Business Layer, và cả Presentation Layer. Mô hình này xuất phát có thể là sự cầu thả của lập trình viên khi không thích tuân theo tiêu chuẩn quan hệ giữa các tầng hoặc cách giải thích khác là sự lỏng lẻo giúp cho việc lập trình thuận tiện hơn. Tuy nhiên lại gặp vấn đề khi muốn mở rộng dự án và sử dụng Framework khác thay cho EF sẽ khiến mất rất nhiều thời gian để sửa code. Lúc này, mô hình Repository phát huy tác dụng của nó. Code của EF sẽ

được viết trong tầng Repository và khi thay đổi framework khác ta chỉ việc sửa code trong tầng Repository.



Hình 2.1.3. b. Mô hình Repository.

d. Identity Sample 2.0

Identity Sample 2.0 là một pakage được cung cấp bởi Microsoft nhằm hỗ trợ quá trình xác thực và phân quyền người dùng trong nhiều loại project web sử dụng .NET Framework khác nhau như ASP.NET MVC, Web Forms, Web Pages, Web API, và SignalR. Identity Sample hỗ trợ:

- Xác định danh tính người dùng khi client sử dụng website
- Kiểm soát thông tin người dùng website
- Sử dụng Entity Framework để lưu trữ dữ liệu người dùng
- Có thể unit test

- Quản lý phân quyền
- Hỗ trợ claims-based authentication
- Cung cấp các phương thức đăng nhập bằng các tài khoản từ mạng xã hội khác
- Sử dụng OWIN CookieAuthentication để tạo cookie lưu trữ thông tin đăng nhập.

2.1.5. Công nghệ được sử dụng để phát triển phía client

a. Bootstrap

Bootstrap là một framework cho phép thiết kế website responsive nhanh hơn và dễ dàng hơn

Bootstrap là bao gồm các HTML templates, CSS templates và Javascript tạo ra những cái cơ bản có sẵn như: typography, forms, buttons, tables, navigation, modals, image carousels và nhiều thứ khác. Trong bootstrap có thêm các plugin Javascript trong nó. Giúp cho việc thiết kế responsive của bạn dễ dàng hơn và nhanh chóng hơn [1].

b. Jquery

jQuery là một thư viện được xây dựng từ Javascript nhằm giúp lập trình viên xây dựng những chức năng có sử dụng Javascript trở nên dễ dàng hơn. jQuery được tích hợp nhiều module khác nhau từ module hiệu ứng cho đến module truy vấn selector. jQuery được sử dụng đến 99% trên tổng số website trên thế giới (trừ những website sử dụng JS Framework).

[1]

Các module chính của jQuery bao gồm:

- Ajax - xử lý Ajax
- Atributes - Xử lý các thuộc tính của đối tượng HTML
- Effect - xử lý hiệu ứng
- Event - xử lý sự kiện
- Form - xử lý sự kiện liên quan tới form
- DOM - xử lý Data Object Model
- Selector – xác định các đối tượng HTML

2.1.6. Những đặc trưng của dự án thuộc lĩnh vực công nghệ thông tin

Dự án công nghệ thông tin không giống như một dự án xây dựng bắt đầu bằng việc hình thành nên bức tranh toàn cảnh về mục tiêu cũng như sản phẩm được lên kế hoạch để hoàn thành. Dự án công nghệ thông tin giống như một con đường có nhiều ngã rẽ, chỉ một biến động nhỏ về cập nhật công nghệ hay thậm chí có tình huống công nghệ bị lạc hậu ngay trong quá trình triển khai dự án, các nhà quản lý bắt buộc phải thay đổi phạm vi, thêm mới các chức năng hay thậm chí thay đổi mục tiêu của mình đồng thời đảm bảo thời gian - tài chính đáp ứng duy trì hệ thống được đồng bộ.

a. Đặc điểm [5]

- Xuất hiện cùng sự ra đời của công nghệ thông tin như một điều tất yếu.
- Tuân theo quy trình PMI: Khởi tạo - Lập kế hoạch - Thực hiện - Theo dõi & Kiểm soát - Kết thúc Dự án
- Coi trọng Truyền thông: Tại sao cho rằng truyền thông là yếu tố cần được chú trọng trong quản lý dự án công nghệ thông tin? Bên cạnh việc kiểm soát thời gian, quản lý ngân sách, ứng biến rủi ro thì linh hoạt trong truyền thông cũng là điều mà bất cứ dự án nào cần coi trọng, đặc biệt với lĩnh vực CNTT. Tuy việc công nghệ thay đổi hàng ngày, hàng giờ đang mang lại những giá trị mới cho nhân loại nhưng chúng vốn tiềm ẩn nhiều rủi ro khiến các dự án CNTT gặp khó khăn, phải đổi hướng hay thậm chí là thất bại. Khi một yêu cầu thay đổi nhỏ về phần mềm hay phần cứng không được truyền đạt nhanh – dù tới các đối tượng liên quan thì hậu quả sẽ vô cùng nghiêm trọng. Trong khi những rắc rối vô hình từ vấn đề công nghệ gây ra thường khó kiểm soát thì yếu tố con người luôn ổn định hơn cả. Sự phối hợp truyền thông và xử lý tình huống của người quản lý luôn đóng vai trò giải quyết đáng kể các rủi ro mà đặc thù lĩnh vực mang lại
- Vòng đời triển khai ngắn: Không giống như những sản phẩm của dự án xây dựng cơ bản có thời gian sử dụng đến vài chục năm, sản phẩm của các dự án CNTT thường có vòng đời ngắn hơn khi và phải yêu tố lạc hậu về công nghệ, yêu cầu người dùng thay đổi. Như vậy một dự án CNTT có thể quay vòng nhiều lần để nâng cấp theo kịp sự phát triển
- Kết hợp đào tạo: Sau khi hoàn thiện sản phẩm cũng như đi vào bàn giao, những người thực hiện dự án phần mềm hay ứng dụng công nghệ thông tin còn phải lên kế hoạch, chuẩn bị tài liệu cũng như các phương pháp đào tạo người dùng. Một sản phẩm CNTT sẽ trở nên vô dụng khi nó không được sử dụng hiệu quả.

b. Phân loại quản lý dự án CNTT

- Dự án phần cứng: Triển khai lắp đặt, đưa trang thiết bị công nghệ, phần cứng vào hoạt động (VD: Dự án lắp đặt hệ thống xử lý nước thải.v.v.)
- Dự án phần mềm: Triển khai ứng dụng phần mềm vào hoạt động quản lý, an ninh, hoạt động sản xuất (VD: Dự án an ninh mạng, xây dựng cơ sở dữ liệu.....).

c. Quy trình của dự án CNTT [5].

- Khởi tạo dự án (Initiating): Giai đoạn này thực hiện việc định nghĩa một dự án mới hoặc một phát sinh (hoặc trộn lẫn) mới của một dự án có sẵn như: Xác định yêu cầu của dự án, mức độ ưu tiên của dự án, phân tích các yêu cầu đầu tư, phân công trách nhiệm cho các bộ phận triển khai.
- Lập kế hoạch dự án (Planning): Giai đoạn này yêu cầu thiết lập phạm vi công việc của dự án, điều chỉnh lại mục tiêu và xác định đường đi tới mục tiêu đó.
- Triển khai (Executing): Giai đoạn này thực hiện hoàn thành các công việc được xác định trong phần lập kế hoạch để đảm bảo các yêu cầu của dự án.
- Giám sát và kiểm soát (Monitoring & Control): Giai đoạn này yêu cầu việc theo dõi, rà soát và điều chỉnh lại tiến độ và khả năng thực hiện của dự án. Theo dõi các rủi ro, thay đổi, phát sinh trong quá trình thực hiện và có những đề xuất điều chỉnh kịp thời.
- Kết thúc (Closing): Giai đoạn này thực hiện để kết thúc tất cả các hoạt động của dự án để chính thức đóng lại dự án.



Hình 2.1.6. a. Mô hình quản lý dự án CNTT.

2.2. Công cụ sử dụng

2.2.1. Visual Studio 2017

Visual studio là một trong những công cụ hỗ trợ lập trình rất nổi tiếng nhất hiện nay của Microsoft và chưa có một phần mềm nào có thể thay thế được nó. Visual Studio được viết bằng 2 ngôn ngữ đó chính là C# và VB+. Đây là 2 ngôn ngữ lập trình giúp người dùng có thể lập trình được hệ thống một cách dễ dàng và nhanh chóng nhất thông qua Visual Studio [1].

Visual Studio là một phần mềm lập trình hệ thống được sản xuất trực tiếp từ Microsoft. Từ khi ra đời đến nay, Visual Studio đã có rất nhiều các phiên bản sử dụng khác nhau. Điều đó, giúp cho người dùng có thể lựa chọn được phiên bản tương thích với dòng máy của mình cũng như cấu hình sử dụng phù hợp nhất.

Bên cạnh đó, Visual Studio còn cho phép người dùng có thể tự chọn lựa giao diện chính cho máy của mình tùy thuộc vào nhu cầu sử dụng.

2.2.2. SQL Server 2014

SQL Server là một hệ thống quản lý cơ sở dữ liệu quan hệ, (relational database management system viết tắt là RDBMS), hỗ trợ một số lượng lớn các quy trình xử lý giao dịch, ứng dụng doanh nghiệp và ứng dụng phân tích trong các công ty IT. SQL Server là một trong 3 công nghệ dữ liệu dẫn đầu hiện nay cùng với Oracle Database và IBM's DB2 [4].

Cũng giống như các phần mềm RDBMS khác, Microsoft SQL Server được xây dựng bên trên lớp SQL - ngôn ngữ lập trình tiêu chuẩn hóa mà quản trị viên cơ sở dữ liệu (DBAs) và các chuyên gia CNTT sử dụng để quản lý cơ sở dữ liệu và truy vấn dữ liệu nằm bên trong. SQL Server thường gắn với Transact-SQL (T-SQL), một cài đặt SQL của Microsoft bổ sung một bộ chương trình mở rộng ngôn ngữ lập trình chuẩn.

CHƯƠNG 3. PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG

Trong chương này em sẽ tiến hành khảo sát hệ thống bằng phương pháp phỏng vấn và quan sát kết hợp tìm hiểu về Trello và Asana, hai công cụ quản lý dự án rất phổ biến trên thế giới, vì mỗi công cụ có một ưu điểm, nhược điểm và đặc trưng riêng. Từ đó, em có thể xác định được những yêu cầu cần thiết cho hệ thống. Hệ thống sẽ được phân tích và thiết kế bằng các loại biểu đồ usecase, biểu đồ lớp, biểu đồ trình tự và biểu đồ hoạt động.

3.1. Khảo sát hệ thống

Qua việc khảo sát hệ thống em đã biết được quy trình quản lý và vận hành một dự án công nghệ thông tin trong các công cụ quản lý dự án nổi tiếng trên thế giới như Trello và Asana. Theo đó, em sẽ có cái nhìn về cách xây dựng và phát triển các chức năng cần có.

3.1.1. Khảo sát hiện trạng hệ thống

3.1.1.1. Địa điểm khảo sát



Hình 3.1.1. a. Hình ảnh địa điểm cty FPT Software tại cơ sở Thành Công Building.

- Tên đơn vị: Cty FPT Software tại cơ sở tòa nhà Thành Công.
- Địa chỉ: 80 Dịch Vọng Hậu, Cầu Giấy, Hà Nội.
- Lĩnh vực kinh doanh: Gia công phần mềm, giải pháp hệ thống,...

3.1.1.2. Tiến hành khảo sát

❖ Đối tượng khảo sát: Người Team Lead team eLOG, thuộc dự án PETRONAS ROTF.

- Phương thức khảo sát: Phỏng vấn trực tiếp.

- Mục đích: Thu thập thông tin, nắm bắt được quy trình quản lý và nghiệp vụ chuyên môn của vai trò team lead trong dự án.

Bảng 3.1.1. a. Bảng câu hỏi phỏng vấn

Số thứ tự	Câu hỏi khảo sát	Câu trả lời
1	Chào anh, em đang thực hiện đồ án về đề tài xây dựng hệ thống quản lý dự án. Mong anh/chị có thể bớt chút thời gian cho em tìm hiểu một vài thông tin được không ạ?	Bạn muốn hỏi về vấn đề như thế nào?
2	Anh có thể cho em biết về vị trí team lead trong một dự án có vai trò như thế nào không ạ?	Trong một dự án, team lead sẽ có vai trò chỉ dưới PM, PM là người quản lý cấp cao nhất của dự án. Trong dự án, anh thường nhận các kế hoạch và chỉ đạo từ PM rồi sau đó phân chia công việc cho các member trong team của mình.
3	Vậy nếu như anh cần đề xuất ý tưởng của mình, báo cáo hay yêu cầu hỗ trợ thì anh sẽ liên hệ với ai ạ?	Nếu trong dự án, anh cần hỗ trợ từ các team khác, anh sẽ liên hệ với team lead của bên team đó, còn nếu cần sự hỗ trợ của bên ngoài thì anh sẽ liên hệ với PM. Sau đó, PM sẽ liên hệ với bên ngoài chứ anh không được trực tiếp liên hệ với họ.
5	Nếu thành viên trong team anh cần hỗ trợ hay đề xuất thì sẽ đề xuất với ai ạ?	Họ sẽ liên hệ trực tiếp với anh và không được liên hệ với những người ở team khác bởi vì anh không thể nắm bắt được thông tin công việc khi họ làm thế.
6	Ngoài công cụ quản lý dự án của cty	Có, anh có sử dụng công cụ trello và

	mình thì anh còn sử dụng công cụ nào bên ngoài không ạ?	asana. Hai công cụ đó rất nổi tiếng trên thế giới. Phần mềm quản lý dự án của cty anh cũng được phát triển dựa trên chúng.
7	Khi anh sử dụng các công cụ quản lý dự án, anh thường hay sử dụng những chức năng nào ạ?	Chủ yếu là các chức năng giao việc cho các thành viên trong team và chức năng duyệt yêu cầu của các thành viên đó. Ngoài ra anh cũng hay sử dụng các loại biểu đồ Kanban và Gantt để theo dõi tiến độ công việc của các thành viên team mình.
8	Anh có yêu cầu cần bổ sung nhằm cải thiện chất lượng của các công cụ đó không ạ?	Các công cụ trên thế giới rất nhiều nhưng rất chung chung, họ không làm sâu vào chuyên ngành CNTT, mỗi công cụ đều có những điểm mạnh, yếu riêng. Ngay cả công cụ của công ty anh cũng vậy, việc tạo một dự án rất phức tạp, anh cần một công cụ mở hơn, có thể cho phép mọi người tự do tạo và quản lý dự án mà không phải bó buộc vào tổ chức nào cả,

❖ Đối tượng khảo sát: Những người đang làm việc trong ngành CNTT

- Phương thức khảo sát: Gửi biểu mẫu câu hỏi qua mail.
- Số lượng khách hàng phản hồi: 59 người.
- Mục đích: thu thập thông tin và nhu cầu của người dùng đối với hệ thống

**Khảo sát về hệ thống quản lý dự án
CNTT**

Tôi đang thực hiện đồ án và đã tài xỉu dụng hệ thống quản lý dự án. Mong các bạn có thể bớt chút thời gian cho em tìm hiểu một vài thông tin về một vài hệ thống quản lý dự án trên thế giới

Quy mô đa số dự án của bạn gồm bao nhiêu người?

Dưới 100 người
 100 - 500 người
 500 - 1000 người
 Trên 1000 người

Bạn thường sử dụng Trello hay Asana để quản lý dự án CNTT của mình?

Trello
 Asana
 Khác
 Không sử dụng hệ thống

Nếu bạn sử dụng Trello, mức độ hài lòng của bạn là bao nhiêu?

Không hài lòng
 Có thể sử dụng được
 Hài lòng
 Rất hài lòng

Nếu bạn sử dụng Asana, mức độ hài lòng của bạn là bao nhiêu?

Không hài lòng
 Có thể sử dụng được
 Hài lòng
 Rất hài lòng

Bạn có yêu cầu gì cần bổ sung nhằm cải thiện chất lượng của Trello và Asana hay không?

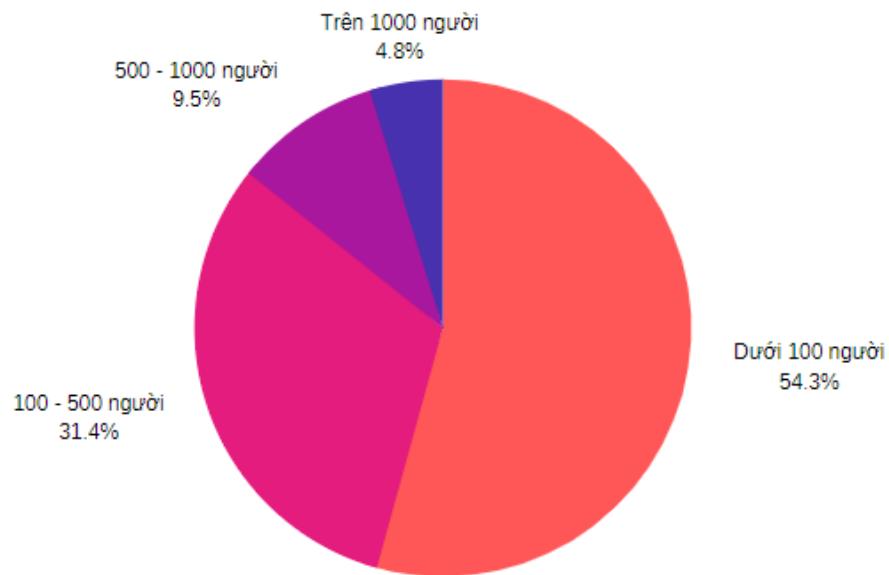
Câu trả lời của bạn

GỬI

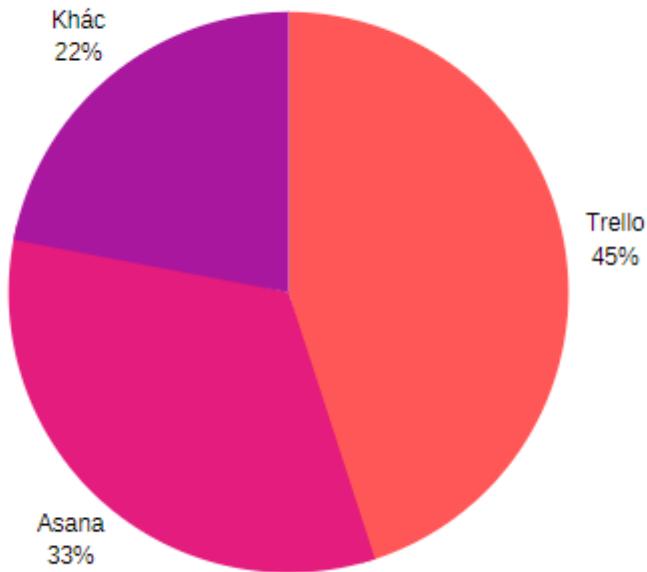
Không bao giờ gửi mật khẩu thông qua Google Biểu mẫu.

Hình 3.1.1. b. Biểu mẫu khảo sát.

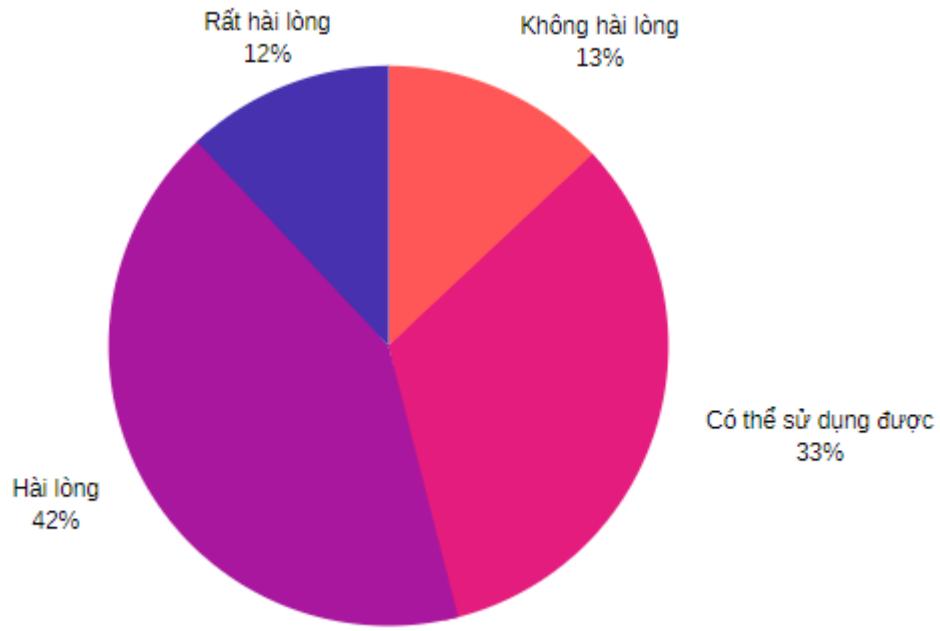
- Thống kê phản hồi của khách hàng



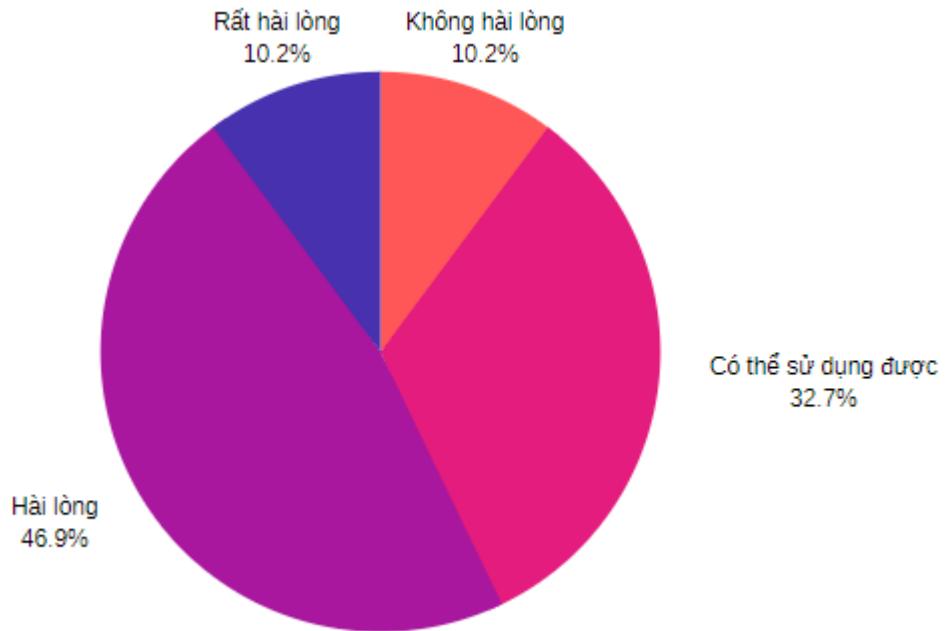
Hình 3.1.1. c. Kết quả cho câu hỏi “Quy mô đa số dự án của ban gồm bao nhiêu người”.



Hình 3.1.1. d. Kết quả cho câu hỏi “Bạn thường sử dụng Trello hay Asana để quản lý dự án CNTT của mình?”



Hình 3.1.1. e. Kết quả cho câu hỏi “Nếu bạn sử dụng Trello, mức độ hài lòng của bạn là bao nhiêu?”.



Hình 3.1.1. f. Kết quả cho câu hỏi “Nếu bạn sử dụng Asana, mức độ hài lòng của bạn là bao nhiêu?”

3.1.2. Kết quả khảo sát về các hệ thống quản lý dự án trên thế giới

Trong phần này, Dựa vào các khảo sát trên và qua việc sử dụng thử, quan sát hai hệ thống quản lý nổi tiếng trên thế giới là Trello và Asana, em sẽ trình bày kết quả tìm hiểu của mình.

a. Trello

Trello là công cụ quản lý dự án vận hành theo phương pháp Kanban. Nguyên lí của phương pháp này là trực quan hoá công việc thành một bảng thông tin, gồm các cột tương ứng với trạng thái công việc To do, Doing, hoặc Done. Mục đích của việc phân loại là để giới hạn số “công việc đang tiến hành”, điều này giúp cho cả nhóm có thể làm việc tập trung hơn, tránh lãng phí thời gian khi phải di chuyển qua lại giữa các công việc khác nhau.

Ở Trello, mỗi “thé” đại diện cho một nhiệm vụ. Các thẻ được phân loại thành các danh sách, mỗi danh sách đại diện cho một tình trạng công việc hoặc một tính chất khác biệt (cần hoàn thành, đã hoàn thành...). Các cột được xếp trong một “bảng”, mỗi “bảng” là một dự án, hoặc một luồng công việc cụ thể.

Trello quản lý công việc dựa trên thẻ (card-based), các bình luận, trao đổi, ghi chú sẽ xảy ra xung quanh “thé” này. Trello hoạt động rất tốt nếu bạn tổ chức dự án theo chiều dọc (theo các giai đoạn khác nhau của dự án), bạn có thể di chuyển một thẻ (công việc) từ giai đoạn này sang giai đoạn tiếp theo.

Ví dụ: Bạn có thể sử dụng Trello cho việc quản lý các công việc của phòng Di trú các “Công việc cần làm” đến “Công việc đang làm” và cuối cùng là “Công việc đã làm”

Qua thực tế sử dụng, Trello được khuyến nghị sử dụng cho các team từ 3 - 10 người, với đặc thù công việc theo dạng tuần tự, cần tập trung vào sự đơn giản, cộng tác hiệu quả,

Ưu điểm của Trello

- Dễ sử dụng

Giao diện làm việc của Trello giống hệt với các tờ giấy note được dán trên màn hình. Do vậy, bất kì ai nhìn vào cũng có thể dễ dàng nắm bắt và sử dụng. Các thao tác sử dụng cũng rất đơn giản.

- Theo dõi trực quan

Trello được thiết kế dựa trên phương pháp quản lý dự án Kanban, nên các giai đoạn công việc sẽ được phân chia thành các danh sách như các to-do list. Và chỉ cần nhìn vào giao diện, là nhà quản lý đã nắm bắt ngay được tiến độ dự án một cách trực quan nhất.

- Khả năng tích hợp lớn

Trello còn có ưu điểm đáng kể đó là khả năng tích hợp lớn khi sử dụng bản trả phí. Nếu bạn sử dụng gói Free, dung lượng file đính kèm chỉ có 10MB nhưng nếu bạn sử dụng gói Business, con số đó lên tới 250MB

Nhược điểm của Trello:

- Không lý tưởng cho các dự án có hàng trăm nhiệm vụ

- Khi số lượng thẻ bắt đầu gia tăng, Trello trở nên khó sử dụng, bởi người dùng bắt đầu mất tầm nhìn vào những gì đã làm, và cần phải tìm kiếm thẻ nhiều hơn và thường xuyên hơn.

- Không phân cấp thành viên quản trị

- Tất cả các thành viên tham gia dự án trên Trello đều có quyền chỉnh sửa, giao việc, xóa bỏ phần đã đánh dấu trước. Sự phân quyền lỏng lẻo hoàn toàn không phù hợp nếu có nhu cầu áp dụng Trello cho cấp độ doanh nghiệp.

- Môi trường giao tiếp kém

Mặc dù các thành viên có thể trao đổi trong các thẻ, tuy nhiên lại thiếu một giao diện cho các bình luận chung về toàn bộ dự án. Trello không có tính năng chat, vì vậy cần tích hợp với một công cụ giao tiếp khác như Slack.

- Không phù hợp cho quản lý thời gian

Với duy nhất một giao diện trải theo chiều ngang, Trello gây khó khăn cho người dùng trong việc quản lý thời gian chính xác của các công việc. Các thẻ được thiết kế độc lập, cản trở việc quản lý mối quan hệ giữa các đầu việc

- Thiếu báo cáo công việc

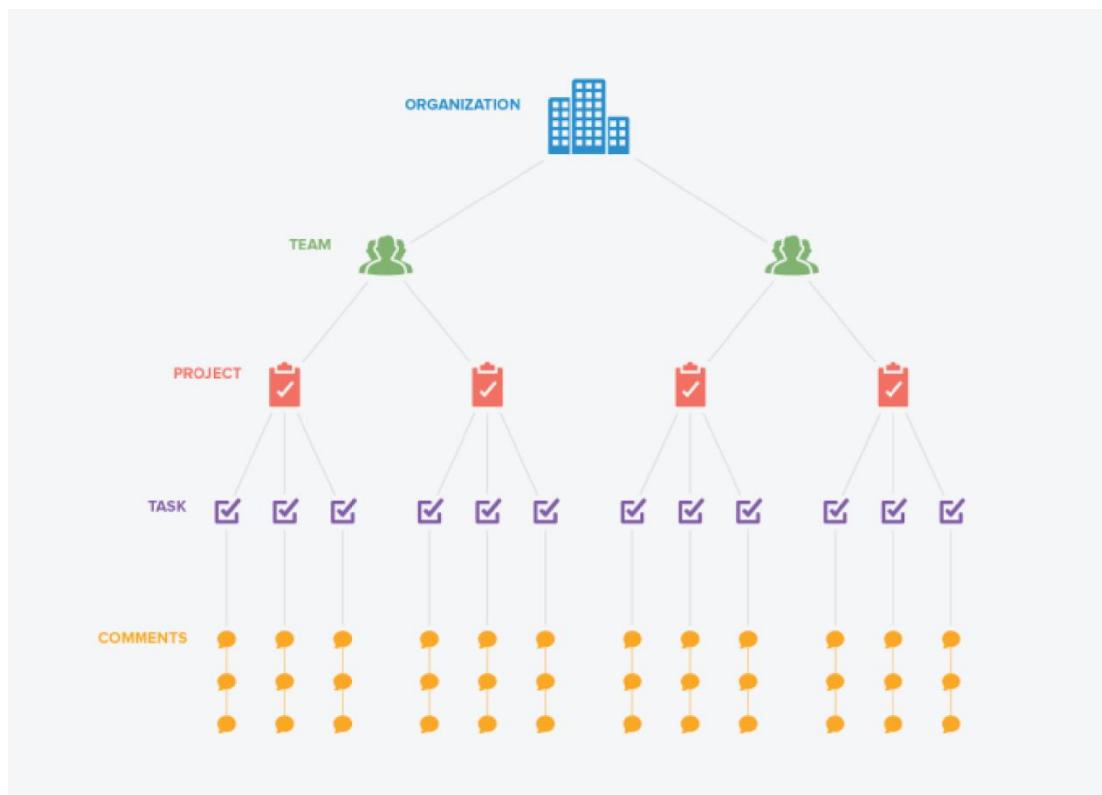
Trello có thể là một công cụ tuyệt vời cho làm việc nhóm, nhưng lại thiếu đi nhiều tính năng thiết yếu đối với vai trò của một người Quản lý (Project/Team manager); trong đó phải

kể đến việc báo cáo. Trello không có một giao diện cho phép người quản lý theo dõi ngay được công việc đã hoàn thành được bao nhiêu % so với dự kiến, những cá nhân nào đang đảm bảo được tiến độ công việc được giao,...).

b. Asana

Khác với Trello, Asana không trực quan dòng công việc (work flow), nó trông giống như một danh sách việc phải làm (to-do list) của cả công ty, nơi tất cả mọi người có thể tạo nhiệm vụ và giao việc cho người khác.

Cách quản lý của Asana được minh họa qua hình sau:



Hình 3.1.2. a. Hình ảnh minh họa mô hình quản lý của Asana.

Ở Asana, mỗi thành viên có một tài khoản riêng và có thể tham gia nhiều “Team” khác nhau. Ví dụ, bạn có thể đồng thời ở cả team Quản lý thi công, Quản lý dự án, Tài chính kế toán... Một người có thể là thành viên của nhiều hơn một team.

Trong mỗi team, “Dự án” được tạo ra và giúp xác định các khối công việc chính mà team cần làm. Bên trong các dự án này, bạn có danh sách “Nhiệm vụ- Task” và “Nhiệm vụ phụ- Subtask” được phân bổ cho các thành viên khác nhau. Những nhiệm vụ này chia nhỏ công việc trong một dự án và xác định rõ ràng ai đang làm gì và vào ngày nào (Trả lời được

2 W: Who + When). Bên trong mỗi công việc, bạn cũng có phần bình luận để giao tiếp với nhau về nhiệm vụ đang diễn ra.

Công cụ Asana quản lý dự án dựa trên các nhiệm vụ (task-based). Asana thích hợp cho các công ty có hệ thống công việc phức tạp, một người cần phải tham gia nhiều dự án/ phòng ban khác nhau. Khi đó, Asana không chỉ giúp team cộng tác hiệu quả, mà còn giúp người quản lý theo dõi tổng thể công việc ở tất cả các phòng ban và dự án.

Ưu điểm của Asana:

- Giao tiếp tốt

Asana có thể giúp loại bỏ việc dùng email hoặc ứng dụng thứ ba để giao tiếp, bởi người dùng được tự động nhận thông báo trong hộp thư đến của họ khi một nhiệm vụ được giao hoặc thay đổi. Tính năng bình luận ở từng nhiệm vụ có thể thay thế các email trao đổi công việc. Giao tiếp theo thời gian thực cho phép các thành viên phản hồi nhanh chóng, giảm thiểu các cuộc họp tốn thời gian.

- Theo dõi tổng quan dự án

Giao diện dạng danh sách, phân chia công việc theo thứ bậc từ lớn đến nhỏ, giúp người quản lý theo dõi được tổng thể dự án.

- Báo cáo công việc

Asana có báo cáo trực quan, cập nhật theo thời gian thực giúp nhà quản lý ngay lập tức đánh giá được hiệu quả của dự án, tuy nhiên tính năng này chỉ có ở phiên bản trả phí.

- Tuỳ chỉnh quyền riêng tư cho dự án

Ở phiên bản trả phí, Asana có thể tạo dự án với tùy chỉnh quyền riêng tư (dự án chỉ hiển thị với một số thành viên cho phép), tạo một không gian làm việc an toàn cho những công việc mang tính bảo mật cao.

Nhược điểm của Asana:

- Không phù hợp với các dự án có tính chất công việc theo tuần tự

Giao diện to-do list thiếu khả năng sắp xếp theo thứ tự chính là điểm yếu của Asana khi quản lý loại dự án này.

- Không có bản tiếng Việt

Hiện tại Asana chỉ có duy nhất một phiên bản tiếng Anh mà chưa mở rộng sang các ngôn ngữ khác.

- Không có biểu đồ Gantt cho quản lý dự án

Đối với các dự án lớn, có nhiều công việc phải hoàn thành cùng lúc, trình tự công việc chồng chéo nhau, trực quan hóa công việc bằng biểu đồ Gantt là điều cực kì cần thiết. Tuy nhiên hiện tại Asana vẫn chưa có tính năng này.

Từ những phân tích ở trên có thể bạn sẽ khó phân định được Trello hay Asana tối ưu hơn, bởi mỗi công cụ quản lý chỉ phục vụ tốt nhất cho những công việc mang tính đặc thù nhất định. Tùy theo tính chất công việc, quy mô dự án, quy mô đội ngũ, bạn có thể sử dụng Trello hoặc Asana kết hợp với một ứng dụng khác để đáp ứng những nhu cầu còn thiếu trong quản lý.

3.1.3. Kết luận

Với mỗi lĩnh vực khác nhau các dự án đều có những đặc thù riêng biệt. Để không mất nhiều thời gian, lãng phí nguồn lực cũng như đảm bảo ngân sách, các nhà quản lý cần nắm được các yếu tố tác động chủ yếu tới dự án của mình, nhanh chóng nhận biết và xử lý kịp thời. Dự án công nghệ thông tin cũng vậy, nhận biết được rủi ro công nghệ, rủi ro truyền thông, tập trung vào yếu tố con người sẽ giúp dự án khắc phục được các bất ổn tiêu cực mà đặc thù ngành mang lại.

Dựa trên những thông tin về đặc trưng của dự án CNTT, em nhận thấy đề tài “Thiết kế và xây dựng hệ thống quản lý dự án CNTT” sẽ đóng vai trò không nhỏ trong việc giải quyết được bài toán về rủi ro truyền thông trong việc quản lý dự án CNTT.

Để thấy, các hệ thống quản lý trên thế giới hiện nay đều không thể hỗ trợ sâu trong việc quản lý các dự án CNTT bởi vì họ muốn mở rộng phạm vi sang dự án của nhiều lĩnh vực khác nên chỉ quan tâm đến việc phát triển về mặt chiều rộng. Các công ty CNTT lớn trong nước thì sẽ tạo ra hệ thống kín cho riêng họ để quản lý nhưng các công ty khởi nghiệp, vừa và nhỏ không thể có đủ kinh phí và kinh nghiệm để xây dựng một hệ thống quản lý dự án thuộc riêng lĩnh vực CNTT.

3.2. Yêu cầu cho hệ thống

Phân tích làm rõ quy trình nghiệp vụ của hệ thống quản lý dự án CNTT bao gồm các chức năng :

- Quản lý tài khoản
 - ❖ Đăng nhập
 - ❖ Đăng xuất
 - ❖ Đổi mật khẩu
 - ❖ Đăng ký tài khoản, xác nhận email
 - ❖ Sửa tài khoản
 - ❖ Xem thông tin của người sở hữu tài khoản
 - ❖ Thông kê dự án đang tham gia
 - ❖ Xóa tài khoản
- Quản lý thông tin dự án
 - ❖ Tạo dự án
 - ❖ Chính sửa thông tin dự án
 - ❖ Đóng dự án
 - ❖ Xóa dự án
 - ❖ Thông kê dự án
- Quản chức vụ công việc trong ngành công nghệ thông tin
 - ❖ Thêm chức vụ
 - ❖ Sửa chức vụ
 - ❖ Phân quyền cho chức vụ trong mỗi dự án
 - ❖ Xóa chức vụ
- Quản lý thành viên trong dự án
 - ❖ Thêm thành viên vào dự án
 - ❖ Gán chức vụ cho thành viên trong dự án
 - ❖ Loại bỏ thành viên khỏi dự án
- Quản lý module của dự án
 - ❖ Thêm module
 - ❖ Sửa module
 - ❖ Xóa module
- Quản lý công việc trong dự án

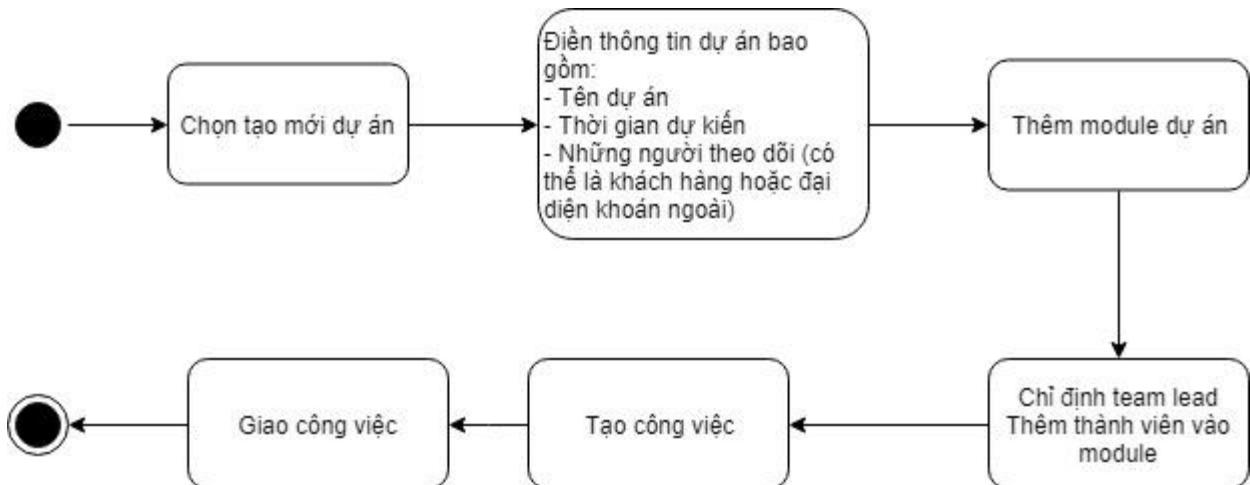
- ❖ Thêm công việc trong dự án
- ❖ Chỉnh sửa thông tin công việc
- ❖ Xóa công việc.
- ❖ Thông kê công việc trong dự án
- Thông kê tỷ lệ phần trăm vai trò của các thành viên trong dự án
- Thông kê nhật ký dự án
- Trích xuất nhật ký dự án ra file Excel
- Quản lý yêu cầu từ các thành viên trong dự án
 - ❖ Thêm yêu cầu
 - ❖ Chỉnh sửa yêu cầu
 - ❖ Duyệt yêu cầu
- Thông kê tiến độ dự án

3.3. Mô tả bài toán

Hệ thống quản lý dự án Công nghệ thông tin với mục tiêu hỗ trợ các doanh nghiệp vừa và nhỏ trong việc quản lý các dự án trong ngành CNTT một cách thuận lợi và hiệu quả. Hệ thống được mô tả như sau:

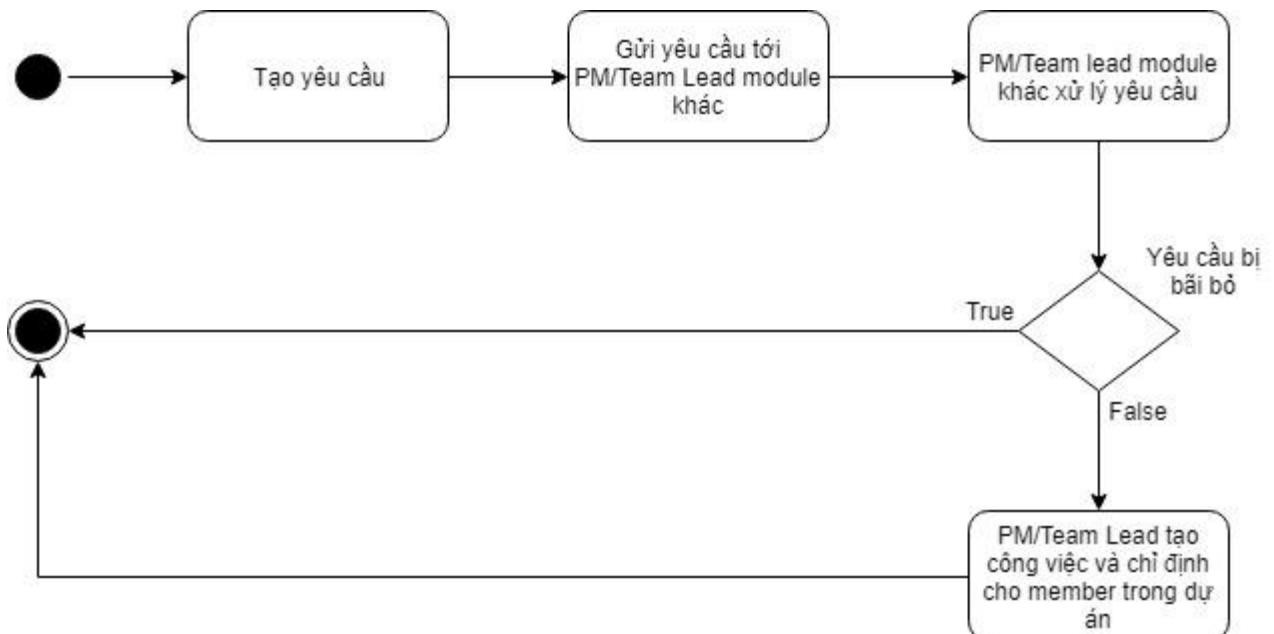
- Người dùng bao gồm các người quản lý dự án có nhu cầu cần một công cụ để quản lý dự án CNTT mà mình đang quản lý và các nhân viên có nhu cầu nhận, theo dõi các công việc mà mình được giao trong dự án, kèm theo đó nhân viên cũng có nhu cầu gửi các yêu cầu cá nhân của mình tới người quản lý dự án
- Quản lý và nhân viên truy cập vào website và đăng ký tài khoản tại website với phân quyền trong hệ thống là user, dùng tài khoản vừa tạo mới đó đăng nhập vào hệ thống để thực hiện chức năng của mình
- Trước tiên, người quản lý sẽ tạo dự án, sau khi thêm dự án, vai trò của người quản lý trong dự án này sẽ là Project Manager (PM). Sau đó, quản lý sẽ thêm các module con trong dự án. Mỗi module con sẽ được quản lý chỉ định cho một user quản lý gọi là Team Lead. Sau khi thêm các module con và các Team Lead tương ứng, người quản lý sẽ thêm các tài khoản khác vào dự án của mình tương ứng với các chức vụ của ngành CNTT, được người quản trị hệ thống quy định trong hệ thống, vào từng module. Chỉ duy nhất PM được quyền thay đổi thông tin của dự án này.

- Sau khi dự án đã được tạo và thiết lập nhân sự đầy đủ, PM và các team lead sẽ tạo các công việc tương ứng với mỗi module và chỉ định cho một thành viên trong module đó thực hiện. Sau khi thực hiện, người thực hiện công việc sẽ chuyển cho người tạo công việc kiểm tra, nếu hoàn thành đúng yêu cầu sẽ thực hiện đóng công việc. PM/Team Lead có quyền xóa các công việc không cần thiết.



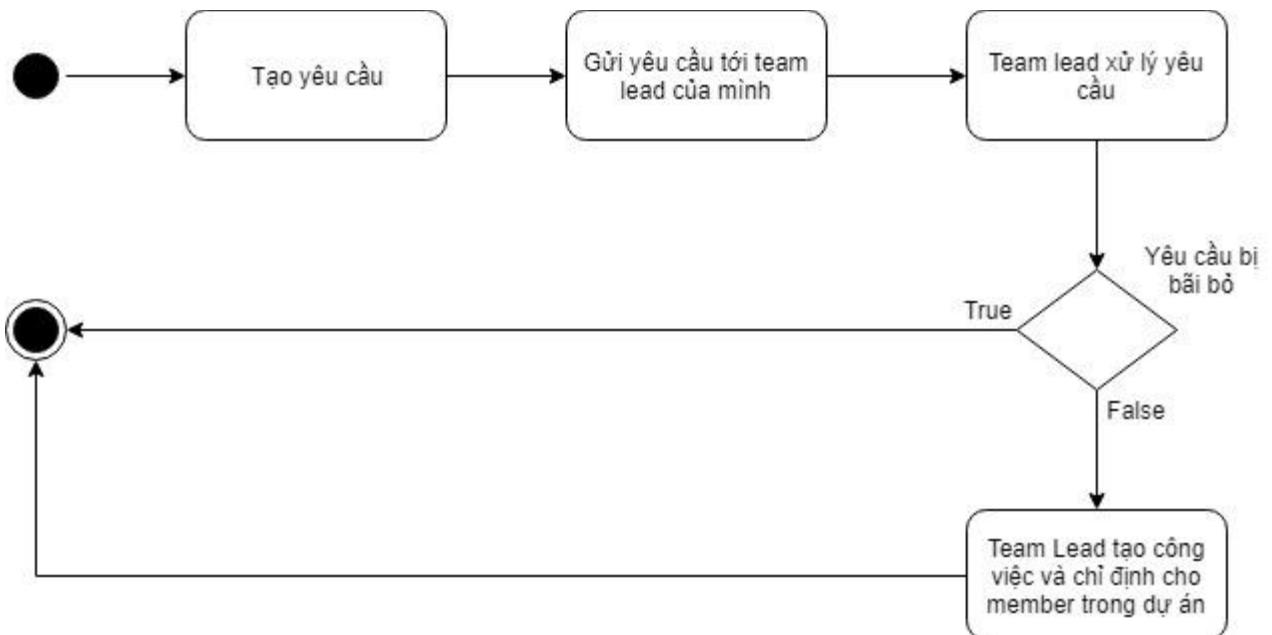
Hình 3.3. a. Sơ đồ mô tả quy trình làm việc của quản lý dự án.

- Khi team lead có một yêu cầu liên quan đến dự án mà cần module khác đáp ứng thì sẽ thực hiện tạo một yêu cầu tới Team Lead module khác hoặc PM, nếu yêu cầu được duyệt thì PM/Team lead module khác sẽ tạo một công việc và chỉ định cho một nhân viên nào đó trong module của mình thực hiện.



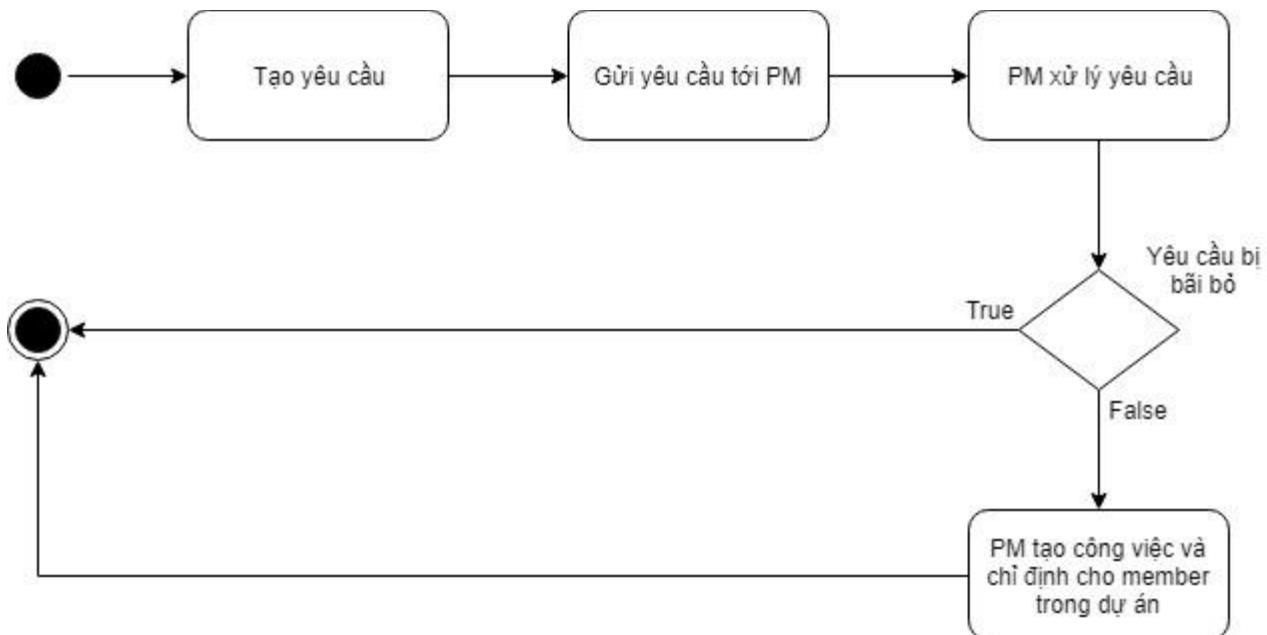
Hình 3.3. b. Sơ đồ mô tả quá trình team lead tạo và gửi yêu cầu.

- Khi team member có một yêu cầu liên quan đến dự án thì sẽ thực hiện tạo một yêu cầu tới Team Lead, nếu yêu cầu được duyệt thì team lead sẽ tạo một công việc và chỉ định cho một nhân viên nào đó trong module của mình thực hiện.



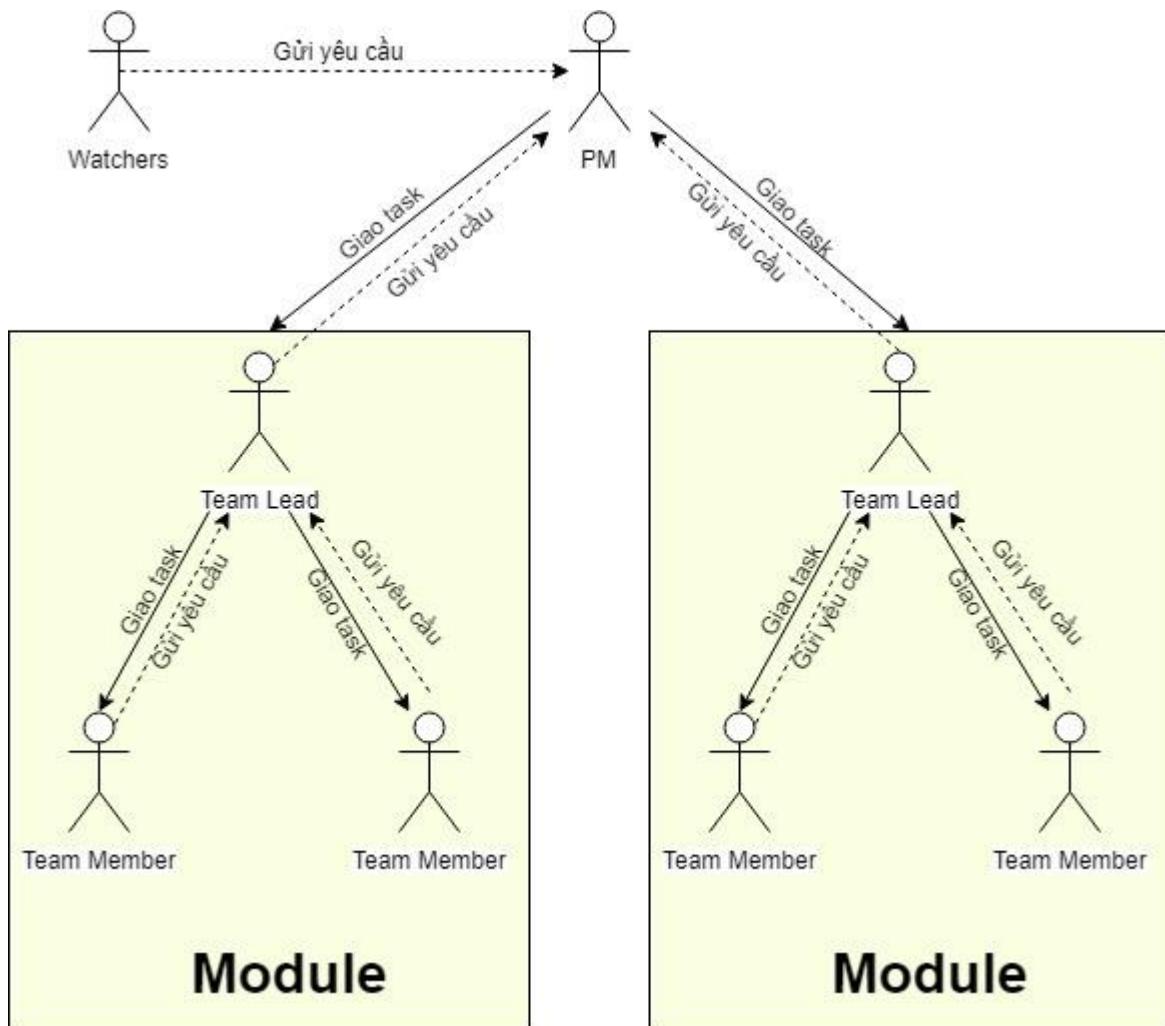
Hình 3.3. c. Sơ đồ mô tả quá trình team member tạo và gửi yêu cầu.

- Watcher không phải là thành viên tham gia trực tiếp vào dự án. Watcher có thể là khách hàng hoặc người giám sát của bên thứ 3. Với vai trò này, watcher không có quyền truy cập vào danh sách công việc mà chỉ có thể tạo yêu cầu tới PM của dự án.



Hình 3.3. d. Sơ đồ mô tả quá trình những người theo dõi dự án tạo và gửi yêu cầu.

Như vậy, mô hình quản lý công việc trong dự án sẽ được minh họa qua hình sau:



Hình 3.3. e. Sơ đồ mô hình quản lý công việc.

- Lịch sử nhật ký của dự án đó
- PM tiến hành theo dõi tiến độ dự án của mình qua việc thống kê tỷ lệ phần trăm vai trò của các thành viên trong dự án, nhật ký dự án, từ đó thống kê được tiến độ của dự án.

3.4. Phân tích thiết kế

3.4.1. Xác định các tác nhân (Actor) và chức năng (UseCase)

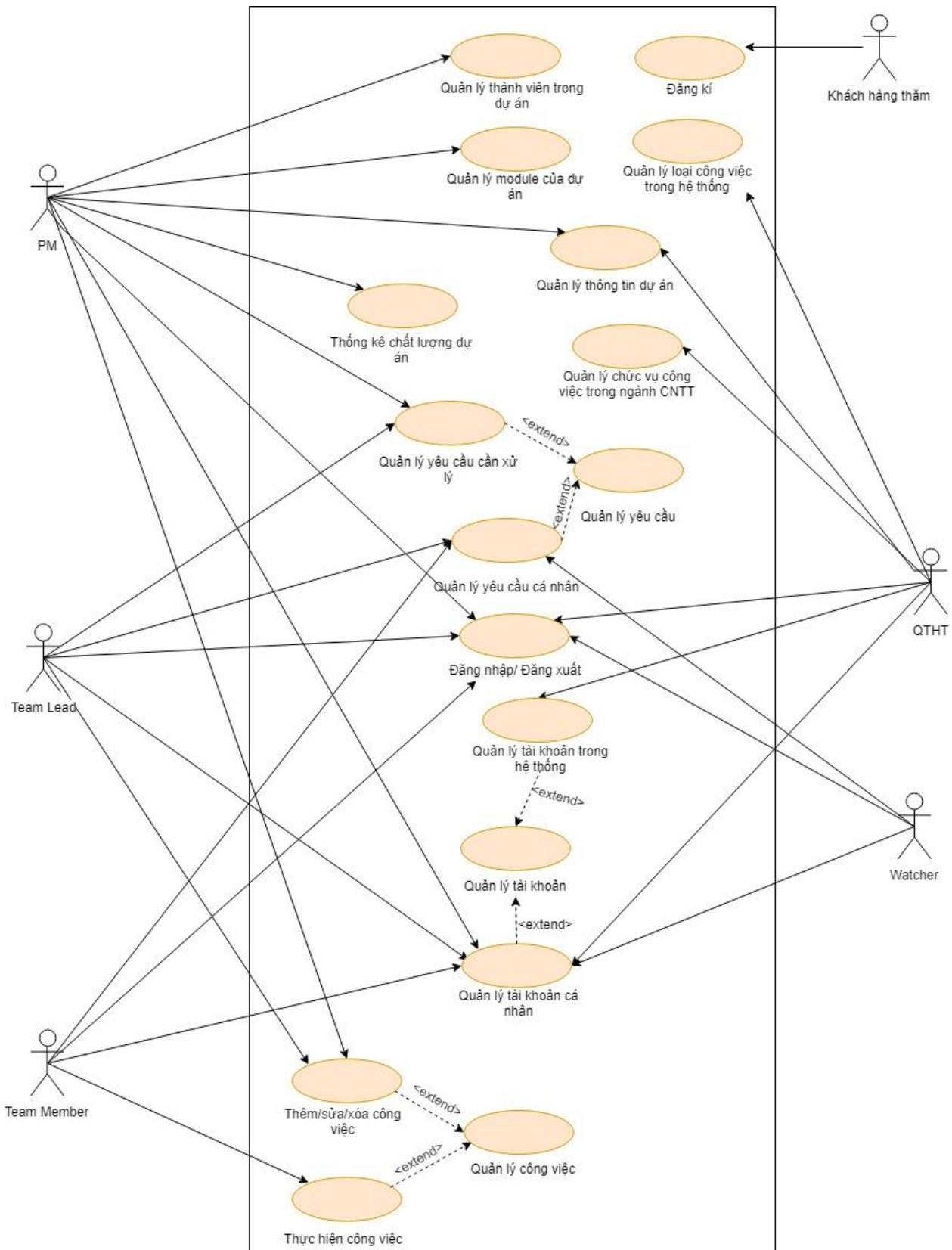
Bảng 3.4.1. a. Bảng tác nhân và chức năng

STT	Tác nhân	Mô tả tác nhân	Chức năng
1	PM	Quản lý dự án	<ul style="list-style-type: none"> ❖ Đăng nhập/ đăng xuất ❖ Quản lý tài khoản cá nhân ❖ Quản lý thông tin dự án, đóng dự án ❖ Quản lý module của dự án ❖ Quản lý thành viên trong dự án được tạo ❖ Quản lý công việc trong dự án được tạo ❖ Thống kê tỷ lệ phần trăm vai trò của các thành viên trong dự án được tạo ❖ Thông kê nhật ký dự án được tạo ❖ Trích xuất nhật ký của dự án ra file được tạo excel ❖ Quản lý yêu cầu của các thành viên trong dự án được tạo ❖ Thống kê tiến độ dự án được tạo ❖ Thống kê danh sách dự án được tạo
2	Team Lead	Quản lý một module trong dự án	<ul style="list-style-type: none"> ❖ Đăng nhập/ đăng xuất ❖ Quản lý tài khoản cá nhân ❖ Quản lý công việc trong module được tạo

			<ul style="list-style-type: none"> ❖ Quản lý yêu cầu của các thành viên trong team module của mình
3	Team member	Thành viên trong dự án (Developer, Tester, BA,...)	<ul style="list-style-type: none"> ❖ Đăng nhập/ đăng xuất ❖ Quản lý tài khoản cá nhân ❖ Thực hiện công việc được giao ❖ Quản lý yêu cầu cá nhân
4	Watcher	Khách hàng hoặc đại diện bên thứ 3	<ul style="list-style-type: none"> ❖ Đăng nhập/đăng xuất ❖ Quản lý tài khoản cá nhân ❖ Tạo yêu cầu
4	Khách hàng thăm	Những user không đăng nhập	<ul style="list-style-type: none"> ❖ Đăng ký tài khoản ❖ Đăng nhập ❖ Xem giới thiệu website
5	QTHT	Quản trị hệ thống	<ul style="list-style-type: none"> ❖ Đăng nhập/đăng xuất ❖ Quản lý dự án: Xóa dự án ❖ Quản lý chức vụ công việc trong ngành CNTT ❖ Quản lý tài khoản

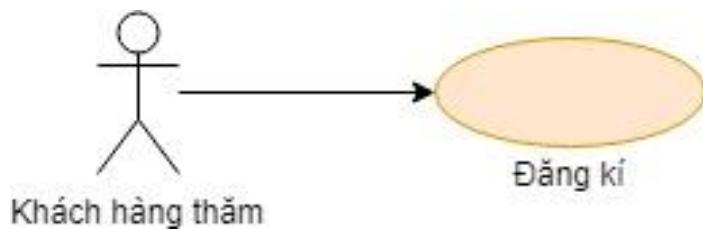
3.4.2. Biểu đồ ca sử dụng (Usecase Diagram)

3.4.2.1. Usecase tổng quát



Hình 3.4.2. a. Biểu đồ usecase tổng quát.

3.4.2.2. Biểu đồ chi tiết Usecase Đăng kí

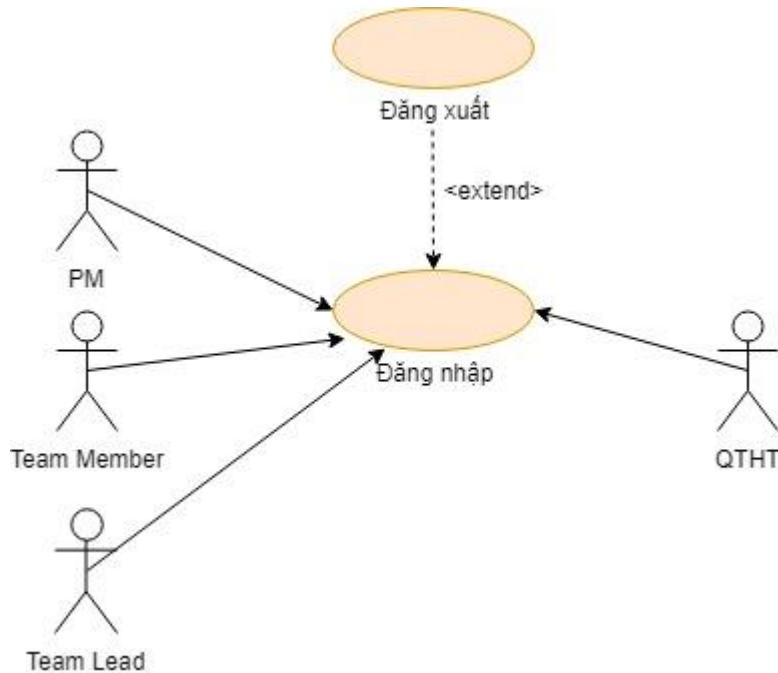


Hình 3.4.2. b. Biểu đồ usecase chi tiết đăng kí.

Bảng 3.4.2. a. Kịch bản usecase đăng kí

1	Tên UC	Đăng kí
2	Tác nhân	Khách hàng thăm
3	Mục đích	Đăng kí hệ thống
4	Mô tả	Người dùng đăng ký làm thành viên của hệ thống
5	Luồng sự kiện chính	1. Người sử dụng chọn chức năng “Đăng ký” 2. Hệ thống hiển thị form “Đăng ký” 3. Người sử dụng nhập thông tin 4. Hệ thống lưu thông tin người dùng vào CSDL và chuyển sang form “Đăng nhập”
6	Luồng sự kiện phụ	1. Nhập sai thông tin 2. Nhập thiếu thông tin

3.4.2.3. Biểu đồ chi tiết Usecase Đăng nhập



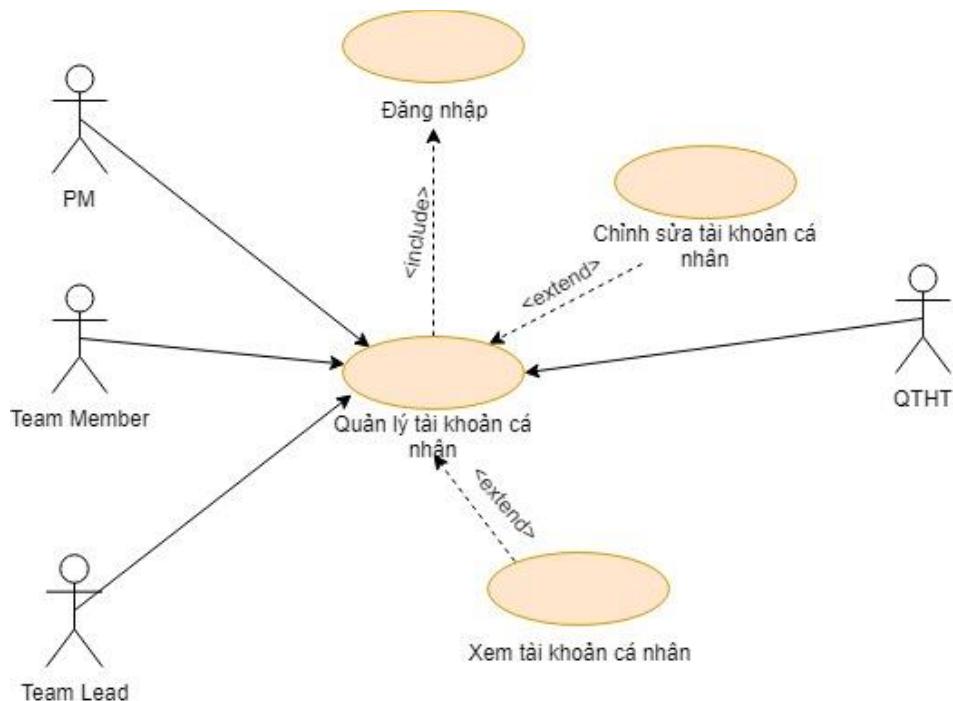
Hình 3.4.2. c. Biểu đồ chi tiết usecase đăng nhập.

Bảng 3.4.2. b. Kích bản usecase đăng nhập

1	Tên UC	Đăng nhập
2	Tác nhân	QTHT, PM, Team Lead, Team Member
3	Mục đích	Đăng nhập hệ thống
4	Mô tả	Người sử dụng đăng nhập vào hệ thống thông qua “Email” và “Password”
5	Luồng sự kiện chính	1. Người sử dụng chọn chức năng “Đăng nhập” 2. Hệ thống hiển thị form “Đăng nhập” 3. Người sử dụng nhập “Email” và “Password”, gửi thông tin đến hệ thống 4. Hệ thống kiểm tra. Nếu đúng thì gửi mail xác nhận email 5. Nếu người dùng sử dụng đúng mail để đăng ký ấn xác nhận trong hộp thư thì hệ thống sẽ cho phép người dùng truy cập, và

		<p>sử dụng các chức năng theo đúng quyền hạn. Nếu không hợp lệ, hệ thống thông báo lỗi và yêu cầu người dùng nhập lại thông tin.</p> <p>6. Nếu người dùng ấn đăng xuất thì sẽ đăng xuất ra khỏi hệ thống và sẽ có quyền hạn của một khách hàng thăm.</p>
6	Luồng sự kiện phụ	<p>1. Nhập sai định dạng “Email” hoặc “Password”</p> <p>2. Nhập thiếu thông tin</p>

3.4.2.4. Quản lý tài khoản cá nhân



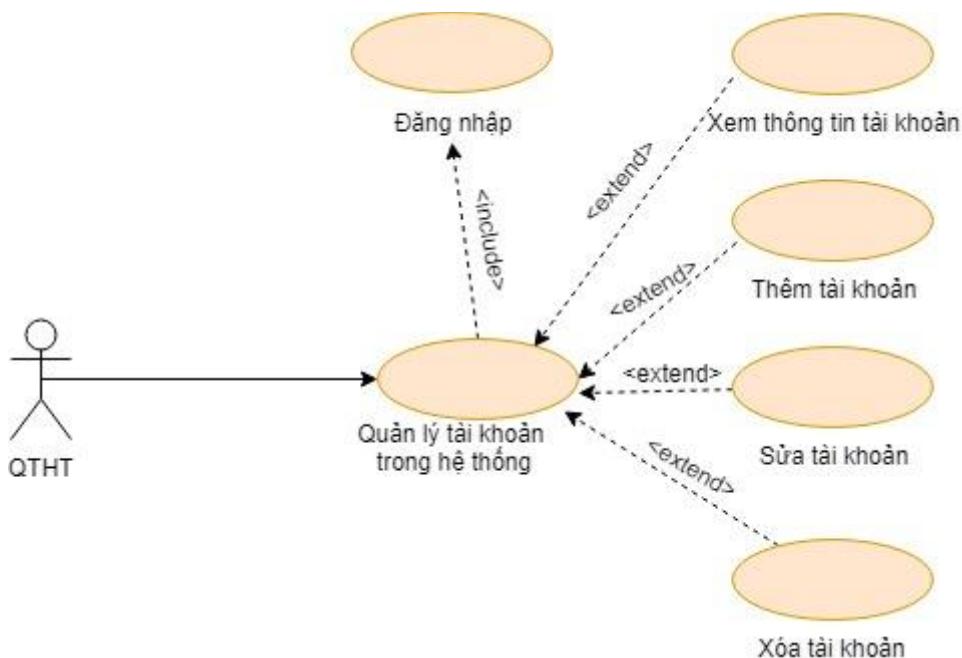
Hình 3.4.2. d. Biểu đồ chi tiết usecase quản lý tài khoản cá nhân.

Bảng 3.4.2. c. Kích bản usecase quản lý tài khoản cá nhân

1	Tên UC	Quản lý tài khoản cá nhân
2	Tác nhân	QTHT, PM, Team Lead, Team Member
3	Tiền điều kiện	Đăng nhập thành công vào hệ thống
4	Mục đích	Quản lý thông tin của tài khoản cá nhân

5	Mô tả	Người sử dụng xem và cập nhật thông tin tài khoản cá nhân của mình
6	Luồng sự kiện chính	1. Người sử dụng chọn chức năng “Xem tài khoản cá nhân” 2. Hệ thống hiển thị form “Tài khoản cá nhân” 3. Người sử dụng nhập thông tin tài khoản muốn sửa và ấn nút cập nhật 4. Hệ thống kiểm tra. Nếu hợp lệ thì cập nhật thông tin tài khoản của tác nhân
7	Luồng sự kiện phụ	1. Nhập thông tin tài khoản không hợp lệ 2. Nhập thiếu thông tin

3.4.2.5. Quản lý tài khoản trong hệ thống



Hình 3.4.2. e. Biểu đồ chi tiết usecase quản lý tài khoản trong hệ thống.

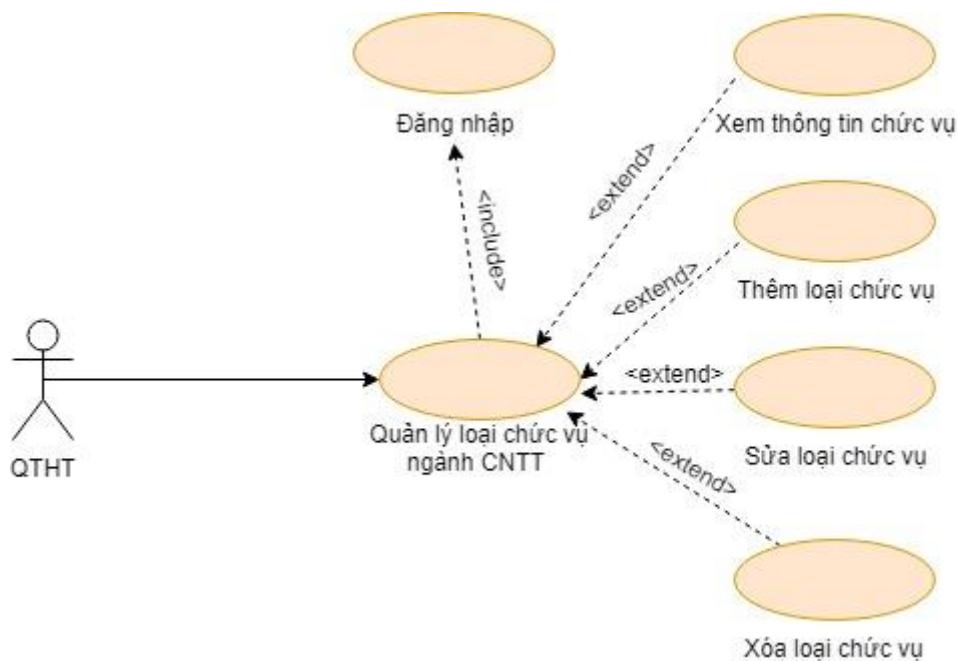
Bảng 3.4.2. d. Kích bản usecase quản lý tài khoản trong hệ thống

1	Tên UC	Quản lý tài khoản trong hệ thống
2	Tác nhân	QTHT
3	Tiền điều kiện	Đăng nhập thành công vào hệ thống

4	Mục đích	Quản lý thông tin tài khoản người dùng trong hệ thống
5	Mô tả	Cho phép người QTHT quản lý thông tin tài khoản của người dùng trong hệ thống
6	Luồng sự kiện chính	<p>1. Người QTHT chọn chức năng “Quản lý tài khoản”</p> <p>2. Hệ thống hiển thị form danh sách tài khoản có trong hệ thống</p> <p>3. Người QTHT chọn chức năng thêm mới tài khoản</p> <p>3.1. Người QTHT nhập thông tin tài khoản sau đó chọn chức năng lưu</p> <p>3.2. Hệ thống kiểm tra. Nếu hợp lệ thì thêm tài khoản mới (tài khoản do QTHT thêm sẽ không cần xác thực email).</p> <p>4. Người QTHT chọn chức năng sửa tài khoản người dùng trong hệ thống</p> <p>4.1. Hệ thống hiển thị form sửa tài khoản</p> <p>4.2. Người QTHT nhập thông tin cần sửa</p> <p>4.3. Hệ thống kiểm tra. Nếu hợp lệ thì cập nhật thông tin tài khoản.</p> <p>5. Người QTHT chọn chức năng xóa tài khoản</p> <p>5.1. Hệ thống hiển thị thông báo xác nhận xóa tài khoản</p> <p>5.2. Người QTHT xác nhận xóa tài khoản</p> <p>5.3. Hệ thống thực hiện xóa tài khoản khách hàng trong CSDL và thông báo nếu thành công.</p> <p>6. Người QTHT chọn chức năng tìm kiếm tài khoản</p> <p>6.1. Hệ thống hiển thị yêu cầu nhập từ khóa</p>

		tìm kiếm 6.2. Người QTHT nhập từ khóa tìm và gửi yêu cầu 6.3. Hệ thống hiển thị danh sách tài khoản cần tìm kiếm
7	Luồng sự kiện phụ	1. Nhập thông tin không hợp lệ 2. Nhập thiếu thông tin

3.4.2.6. Quản lý chức vụ trong ngành CNTT



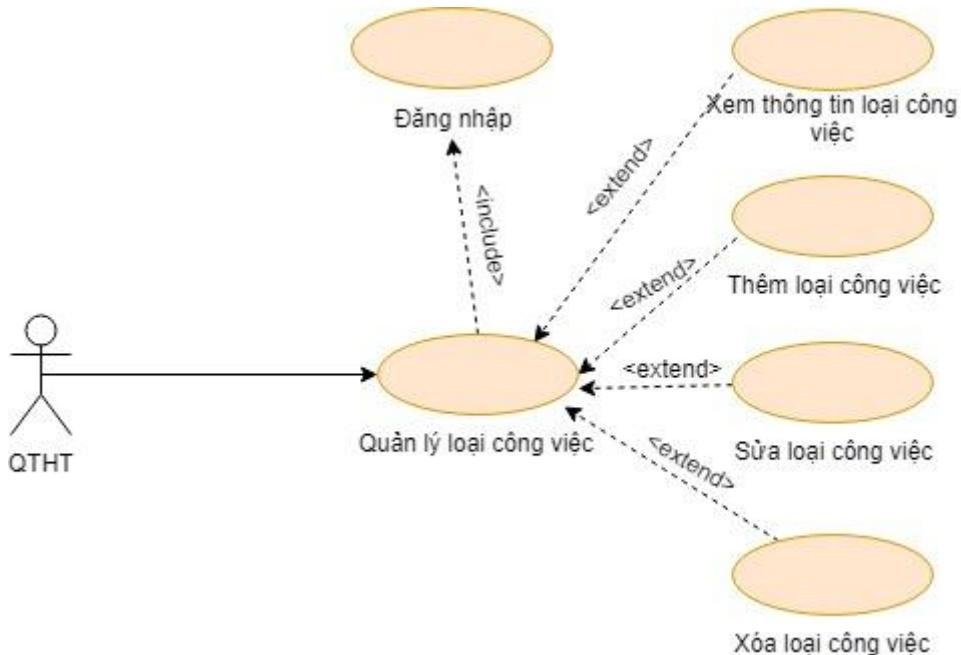
Hình 3.4.2. f. Biểu đồ chi tiết usecase quản lý chức vụ trong ngành CNTT.

Bảng 3.4.2. e. Kích bản usecase quản lý chức vụ trong ngành CNTT

1	Tên UC	Quản lý chức vụ trong ngành CNTT
2	Tác nhân	QTHT
3	Tiền điều kiện	Đăng nhập thành công vào hệ thống
4	Mục đích	Quản lý các chức vụ trong ngành CNTT được áp dụng trong hệ thống
5	Mô tả	Cho phép người QTHT quản lý các chức vụ trong ngành CNTT được áp dụng trong hệ thống

6	Luồng sự kiện chính	<p>1. Người QTHT chọn chức năng “Quản lý chức vụ”</p> <p>2. Hệ thống hiển thị form danh sách chức vụ có trong hệ thống</p> <p>3. Người QTHT chọn chức năng thêm mới chức vụ</p> <p>3.1. Người QTHT nhập thông tin chức vụ sau đó chọn chức năng lưu</p> <p>3.2. Hệ thống kiểm tra. Nếu hợp lệ thì thêm chức vụ mới.</p> <p>4. Người QTHT chọn chức năng sửa chức vụ trong hệ thống</p> <p>4.1. Hệ thống hiển thị form sửa chức vụ</p> <p>4.2. Người QTHT nhập thông tin cần sửa</p> <p>4.3. Hệ thống kiểm tra. Nếu hợp lệ thì cập nhật thông tin chức vụ.</p> <p>5. Người QTHT chọn chức năng xóa chức vụ</p> <p>5.1. Hệ thống hiển thị thông báo xác nhận xóa chức vụ</p> <p>5.2. Người QTHT xác nhận xóa chức vụ</p> <p>5.3. Hệ thống thực hiện xóa chức vụ trong CSDL và thông báo nếu thành công.</p>
7	Luồng sự kiện phụ	<p>1. Nhập thông tin không hợp lệ</p> <p>2. Nhập thiếu thông tin</p>

3.4.2.7. Quản lý loại công việc



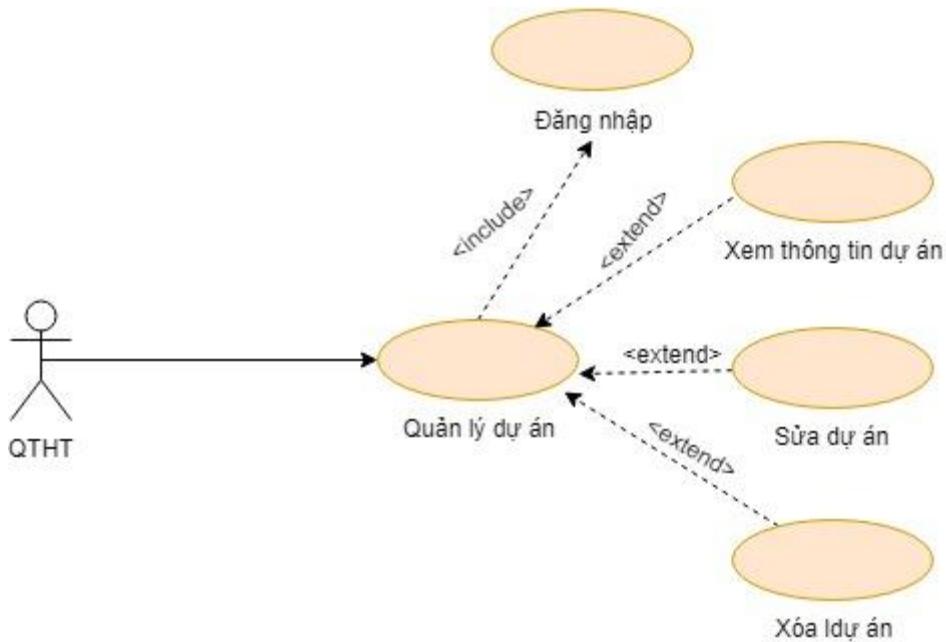
Hình 3.4.2. g. Biểu đồ chi tiết usecase quản lý loại công việc.

Bảng 3.4.2. f. Kích bản usecase quản lý loại công việc

1	Tên UC	Quản lý loại công việc trong ngành CNTT
2	Tác nhân	QTHT
3	Tiền điều kiện	Đăng nhập thành công vào hệ thống
4	Mục đích	Quản lý các loại công việc được áp dụng trong hệ thống
5	Mô tả	Cho phép người QTHT quản lý các loại công việc trong ngành CNTT được áp dụng trong hệ thống
6	Luồng sự kiện chính	<ol style="list-style-type: none"> Người QTHT chọn chức năng “Quản lý loại công việc” Hệ thống hiển thị form danh sách loại công việc có trong hệ thống Người QTHT chọn chức năng thêm mới loại công việc <ol style="list-style-type: none"> Người QTHT nhập thông tin loại công

		<p>việc sau đó chọn chức năng lưu</p> <p>3.2. Hệ thống kiểm tra. Nếu hợp lệ thì thêm loại công việc mới.</p> <p>4. Người QTHT chọn chức năng sửa loại công việc trong hệ thống</p> <p>4.1. Hệ thống hiển thị form sửa loại công việc</p> <p>4.2. Người QTHT nhập thông tin cần sửa</p> <p>4.3. Hệ thống kiểm tra. Nếu hợp lệ thì cập nhật thông tin loại công việc.</p> <p>5. Người QTHT chọn chức năng xóa loại công việc</p> <p>5.1. Hệ thống hiển thị thông báo xác nhận xóa loại công việc</p> <p>5.2. Người QTHT xác nhận xóa loại công việc</p> <p>5.3. Hệ thống thực hiện xóa loại công việc trong CSDL và thông báo nếu thành công.</p>
7	Luồng sự kiện phụ	<p>1. Nhập thông tin không hợp lệ</p> <p>2. Nhập thiếu thông tin</p>

3.4.2.8. Quản lý dự án trong hệ thống



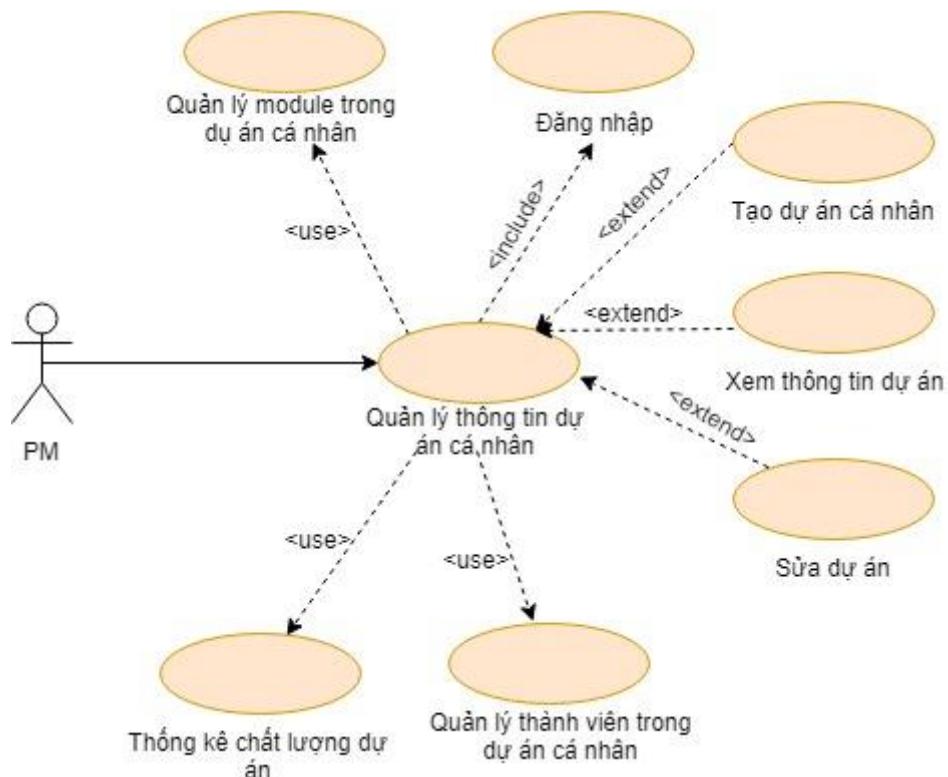
Hình 3.4.2. h. Biểu đồ chi tiết usecase quản lý dự án trong hệ thống.

Bảng 3.4.2. g. Kích bản usecase quản lý dự án trong hệ thống

1	Tên UC	Quản lý dự án
2	Tác nhân	QTHT
3	Tiền điều kiện	Đăng nhập thành công vào hệ thống
4	Mục đích	Quản lý thông tin các dự án được áp dụng trong hệ thống
5	Mô tả	Cho phép người QTHT quản lý thông tin các dự án được áp dụng trong hệ thống
6	Luồng sự kiện chính	1. Người QTHT chọn chức năng “Quản lý dự án” 2. Hệ thống hiển thị form danh sách dự án có trong hệ thống 3. Người QTHT chọn chức năng sửa dự án trong hệ thống 3.1. Hệ thống hiển thị form sửa dự án 3.2. Người QTHT nhập thông tin cần sửa

		<p>3.3. Hệ thống kiểm tra. Nếu hợp lệ thì cập nhật thông tin dự án.</p> <p>4. Người QTHT chọn chức năng xóa dự án</p> <p>4.1. Hệ thống hiển thị thông báo xác nhận xóa dự án</p> <p>4.2. Người QTHT xác nhận xóa dự án</p> <p>4.3. Hệ thống thực hiện xóa dự án trong CSDL và thông báo nếu thành công.</p>
7	Luồng sự kiện phụ	<p>1. Nhập thông tin không hợp lệ</p> <p>2. Nhập thiếu thông tin</p>

3.4.2.9. Quản lý dự án cá nhân



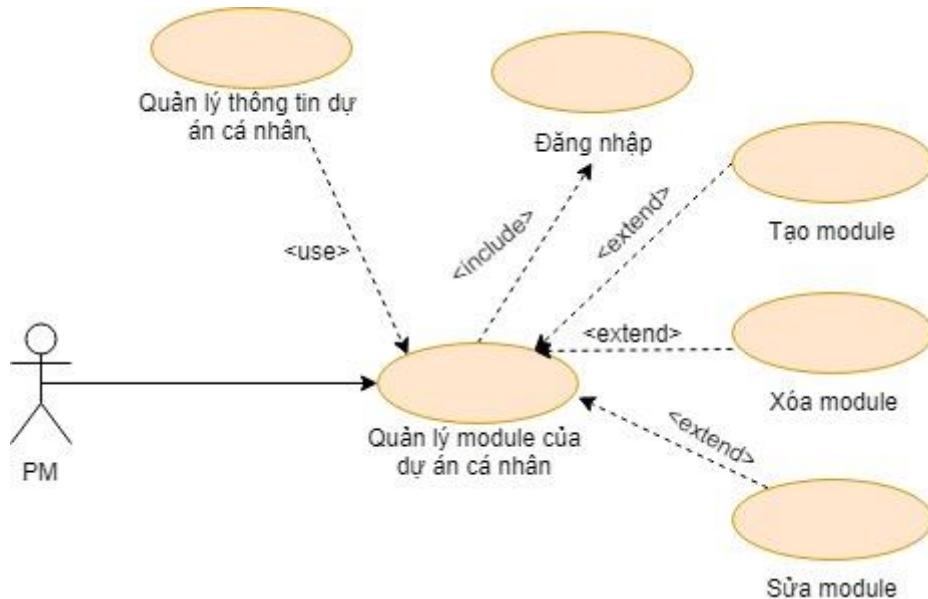
Hình 3.4.2. i. Biểu đồ chi tiết usecase quản lý dự án cá nhân.

Bảng 3.4.2. h. Kích bản usecase quản lý dự án cá nhân

1	Tên UC	Quản lý thông tin dự án cá nhân
2	Tác nhân	PM
3	Tiền điều kiện	Đăng nhập thành công vào hệ thống

4	Mục đích	Quản lý thông tin các dự án cá nhân đã được tạo
5	Mô tả	Cho phép PM quản lý các dự án cá nhân được tạo trong hệ thống
6	Luồng sự kiện chính	<p>1. PM chọn chức năng “Quản lý dự án cá nhân”</p> <p>2. Hệ thống hiển thị form danh sách dự án cá nhân đã được tạo</p> <p>3. PM chọn chức năng thêm mới dự án cá nhân</p> <p>3.1. PM nhập thông tin dự án sau đó chọn chức năng lưu</p> <p>3.2. Hệ thống kiểm tra. Nếu hợp lệ thì thêm dự án cá nhân mới, dự án cá nhân lúc này thuộc quyền sở hữu của người tạo.</p> <p>4. PM chọn chức năng sửa thông tin dự án cá nhân của mình</p> <p>4.1. Hệ thống hiển thị form sửa dự án cá nhân</p> <p>4.2. PM nhập thông tin cần sửa</p> <p>4.3. Hệ thống kiểm tra. Nếu hợp lệ thì cập nhật thông tin dự án cá nhân.</p>
7	Luồng sự kiện phụ	<p>1. Nhập thông tin không hợp lệ</p> <p>2. Nhập thiếu thông tin</p>

3.4.2.10. Quản lý module trong dự án cá nhân



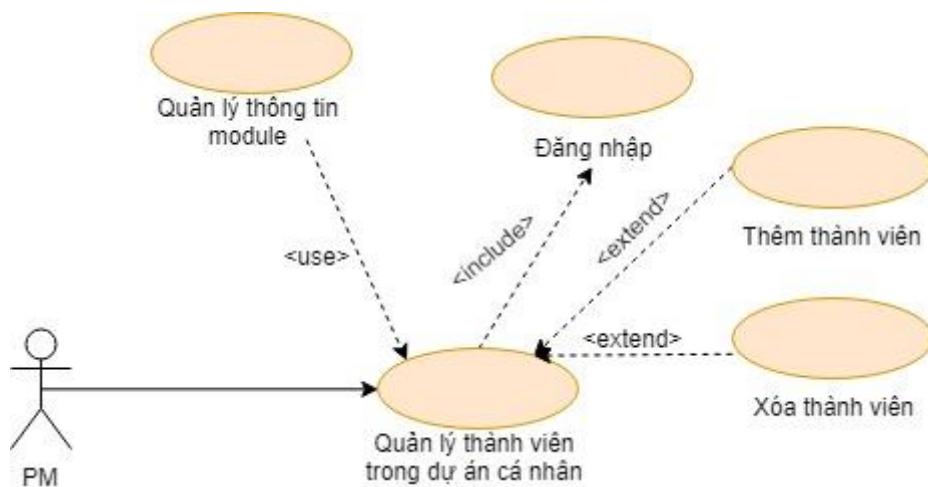
Hình 3.4.2. j. Biểu đồ chi tiết usecase quản lý module.

Bảng 3.4.2. i. Kích bản usecase quản lý module

1	Tên UC	Quản lý module trong dự án cá nhân
2	Tác nhân	PM
3	Tiền điều kiện	Đăng nhập thành công vào hệ thống Đang trong chức năng thêm / sửa thông tin dự án cá nhân
4	Mục đích	Quản lý các module trong các dự án cá nhân
5	Mô tả	Cho phép PM quản lý module trong dự án cá nhân được tạo trong dự án cá nhân
6	Luồng sự kiện chính	1. PM chọn chức năng “Quản lý dự án cá nhân” 2. Hệ thống hiển thị form danh sách dự án cá nhân đã được tạo 3. PM chọn chức năng thêm/sửa dự án cá nhân và chọn chức năng thêm mới module 3.1. PM nhập thông tin module sau đó chọn

		<p>chức năng lưu</p> <p>3.2. Hệ thống kiểm tra. Nếu hợp lệ thì thêm chức vụ mới.</p> <p>4. PM chọn chức năng sửa module trong hệ thống</p> <p>4.1. Hệ thống hiển thị form sửa module</p> <p>4.2. PM nhập thông tin cần sửa</p> <p>4.3. Hệ thống kiểm tra. Nếu hợp lệ thì cập nhật thông tin module</p> <p>5. PM chọn chức năng xóa module</p> <p>5.1. Hệ thống hiển thị thông báo xác nhận xóa module</p> <p>5.2. PM xác nhận xóa module</p> <p>5.3. Hệ thống thực hiện xóa module trong CSDL và thông báo nếu thành công.</p>
7	Luồng sự kiện phụ	<p>1. Nhập thông tin không hợp lệ</p> <p>2. Nhập thiếu thông tin</p>

3.4.2.11. Quản lý thành viên trong dự án cá nhân

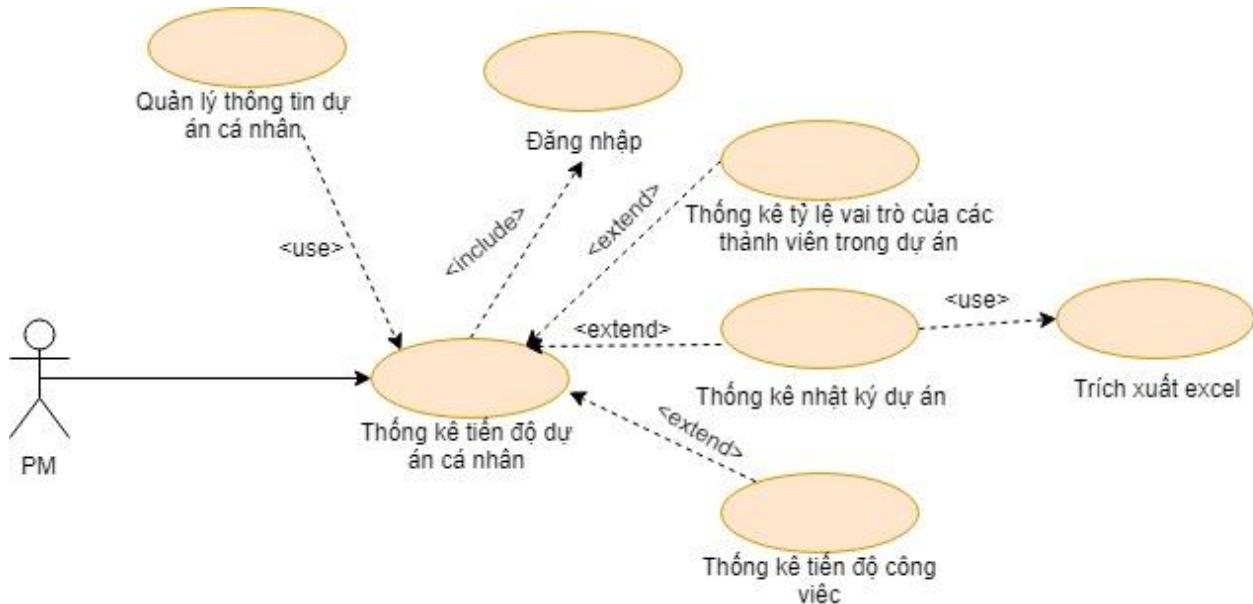


Hình 3.4.2. k. Biểu đồ chi tiết usecase quản lý thành viên trong dự án.

Bảng 3.4.2. j. Kích bản usecase quản lý thành viên trong dự án

1	Tên UC	Quản lý thành viên trong dự án cá nhân
2	Tác nhân	PM
3	Tiền điều kiện	Đăng nhập thành công vào hệ thống Đang trong chức năng thêm / sửa thông tin dự án cá nhân
4	Mục đích	Quản lý các thành viên trong dự án cá nhân
5	Mô tả	Cho phép PM quản lý thành viên trong dự án cá nhân
6	Luồng sự kiện chính	1. PM chọn chức năng “Quản lý module” 2. Hệ thống hiển thị form danh sách module được tạo trong dự án 3. PM chọn chức năng thêm/sửa module và chọn chức năng thêm mới thành viên 3.1. PM nhập account user muốn thêm làm thành viên sau đó chọn chức năng lưu 3.2. Hệ thống kiểm tra. Nếu hợp lệ thì thêm thành viên 4. PM chọn chức năng xóa thành viên 5.1. Hệ thống hiển thị thông báo xác nhận xóa thành viên 5.2. PM xác nhận xóa thành viên 5.3. Hệ thống thực hiện xóa thành viên khỏi dự án cá nhân và và thông báo nếu thành công.
7	Luồng sự kiện phụ	1. Nhập thông tin không hợp lệ 2. Nhập thiếu thông tin

3.4.2.12. Thống kê tiến độ dự án cá nhân



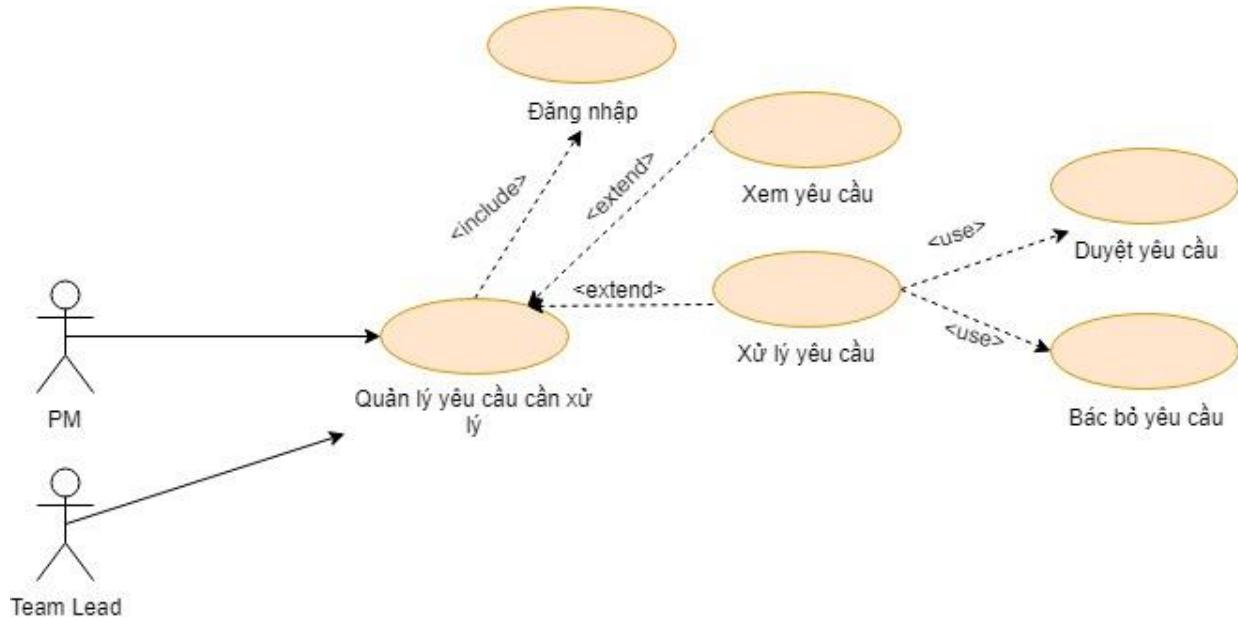
Hình 3.4.2. 1. Biểu đồ chi tiết usecase thống kê tiến độ dự án.

Bảng 3.4.2. k. Kích bản usecase thống kê tiến độ dự án

1	Tên UC	Thống kê tiến độ dự án cá nhân
2	Tác nhân	PM
3	Tiền điều kiện	Đăng nhập thành công vào hệ thống Đang trong chức năng thêm / sửa thông tin dự án cá nhân
4	Mục đích	Thống kê các yếu tố thể hiện tiến độ của dự án như số lượng công việc còn tồn đọng, số lượng yêu cầu cần giải quyết,...
5	Mô tả	Cho phép PM dựa trên các chỉ số thống kê để xác định được tiến độ dự án, nhằm đưa ra các kế hoạch phù hợp để phát triển dự án cá nhân
6	Luồng sự kiện chính	1. PM chọn chức năng “Quản lý dự án cá nhân” 2. Hệ thống hiển thị form danh sách dự án cá nhân đã được tạo

		<p>3. PM chọn chức năng thêm/sửa dự án cá nhân và chọn chức thống kê tỷ lệ vai trò của các thành viên trong dự án</p> <p>3.1. Hệ thống hiển thị form tỷ lệ vai trò của các thành viên trong dự án theo biểu đồ tròn.</p> <p>4. PM chọn chức năng thống kê nhật ký dự án</p> <p>4.1. Hệ thống hiển thị form danh sách lịch sử thay đổi trạng thái của công việc và nhân sự trong dự án</p> <p>4.2. PM chọn chức năng trích xuất nhật ký dự án ra file excel</p> <p>4.3. Hệ thống trả về cho PM một file excel để download</p> <p>5. PM chọn chức năng thống kê tiến độ dự án</p> <p>5.1. Hệ thống hiển thị form thống kê tiến độ dự án</p>
7	Luồng sự kiện phụ	<p>1. Nhập thông tin không hợp lệ</p> <p>2. Nhập thiếu thông tin</p>

3.4.2.13. Quản lý yêu cầu cần xử lý



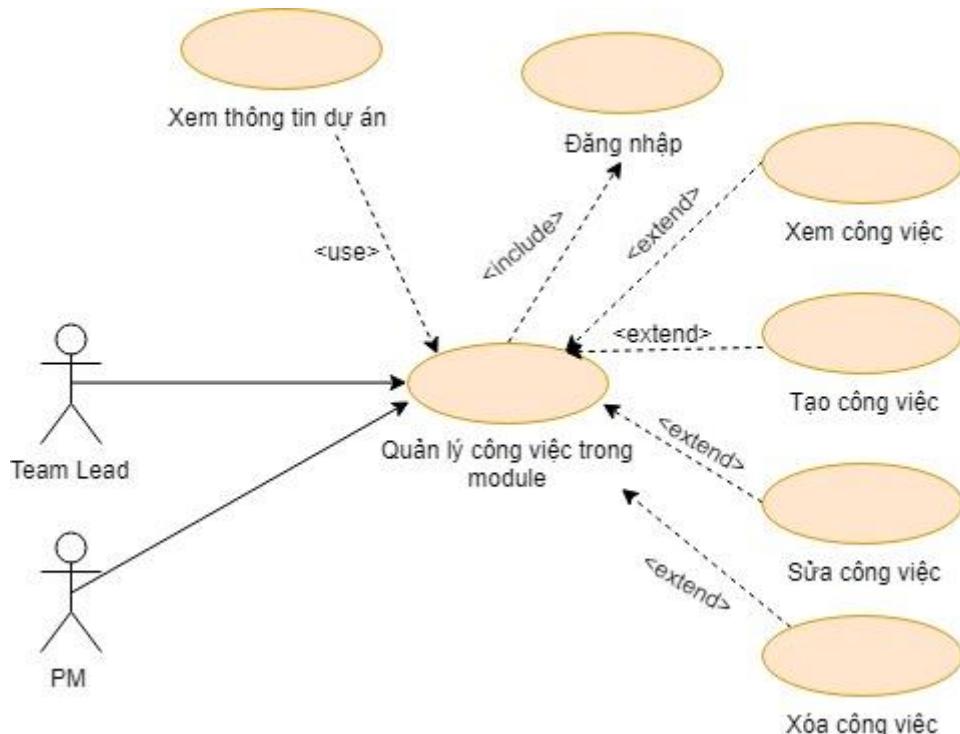
Hình 3.4.2. m. Biểu đồ chi tiết usecase quản lý yêu cầu cần xử lý.

Bảng 3.4.2. l. Kích bản usecase quản lý yêu cầu cần xử lý

1	Tên UC	Quản lý yêu cầu cần xử lý
2	Tác nhân	PM, Team Lead
3	Tiền điều kiện	Đăng nhập thành công vào hệ thống
4	Mục đích	Quản lý yêu cầu của các thành viên trong dự án cá nhân được yêu cầu xử lý
5	Mô tả	Cho phép PM/Team Lead nhận và xử lý các yêu cầu của các thành viên trong dự án mình đang quản lý
6	Luồng sự kiện chính	<ol style="list-style-type: none"> 1. PM/Team Lead chọn chức năng “Quản lý yêu cầu được giao cho tôi” 2. Hệ thống hiển thị form danh sách yêu cầu từ các thành viên trong dự án được giao cho tôi 3. PM/Team Lead chọn chức năng xem yêu cầu 4. Hệ thống hiển thị form chi tiết yêu cầu

		<p>5. PM/Team Lead chọn chức năng xử lý yêu cầu</p> <p>5.1. Hệ thống hiển thị thông báo xác nhận duyệt/bắc bỏ yêu cầu</p> <p>5.2. PM/Team Lead xác nhận xử lý yêu cầu</p> <p>5.3. Hệ thống thực hiện xử lý yêu cầu và thông báo nếu thành công.</p>
7	Luồng sự kiện phụ	

3.4.2.14. Quản lý công việc trong module



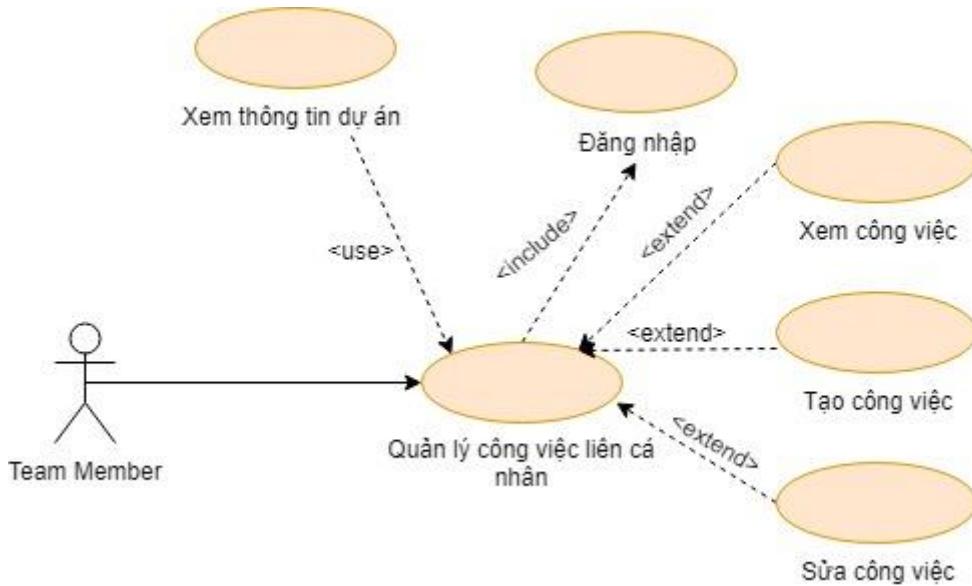
Hình 3.4.2. n. Biểu đồ chi tiết usecase quản lý công việc.

Bảng 3.4.2. m. Kích bản usecase quản lý công việc

1	Tên UC	Quản lý công việc trong module
2	Tác nhân	Team lead, PM
3	Tiền điều kiện	Đăng nhập thành công vào hệ thống Đang trong chức năng xem thông tin dự án

		cá nhân
4	Mục đích	Quản lý các công việc trong dự án cá nhân
5	Mô tả	Cho phép Team Lead quản lý các công việc trong module do cá nhân quản lý
6	Luồng sự kiện chính	<p>1. Team lead, PM chọn chức năng “Quản lý dự án cá nhân”</p> <p>2. Hệ thống hiển thị form danh sách dự án cá nhân đã được tạo</p> <p>3. Team lead, PM chọn chức năng xem dự án cá nhân và chọn chức năng thêm mới công việc</p> <p>3.1. Team lead, PM nhập thông tin công việc sau đó chọn chức năng lưu</p> <p>3.2. Hệ thống kiểm tra. Nếu hợp lệ thì thêm công việc mới.</p> <p>4. Team lead, PM chọn chức năng sửa công việc trong module mình quản lý</p> <p>4.1. Hệ thống hiển thị form sửa công việc</p> <p>4.2. Team lead, PM nhập thông tin cần sửa</p> <p>4.3. Hệ thống kiểm tra. Nếu hợp lệ thì cập nhật thông tin công việc</p> <p>5. Team lead, PM chọn chức năng xóa công việc</p> <p>5.1. Hệ thống hiển thị thông báo xác nhận xóa công việc</p> <p>5.2. Team lead, PM xác nhận xóa công việc</p> <p>5.3. Hệ thống thực hiện xóa công việc trong CSDL và thông báo nếu thành công.</p>
7	Luồng sự kiện phụ	<p>1. Nhập thông tin không hợp lệ</p> <p>2. Nhập thiếu thông tin</p>

3.4.2.15. Thực hiện công việc được giao



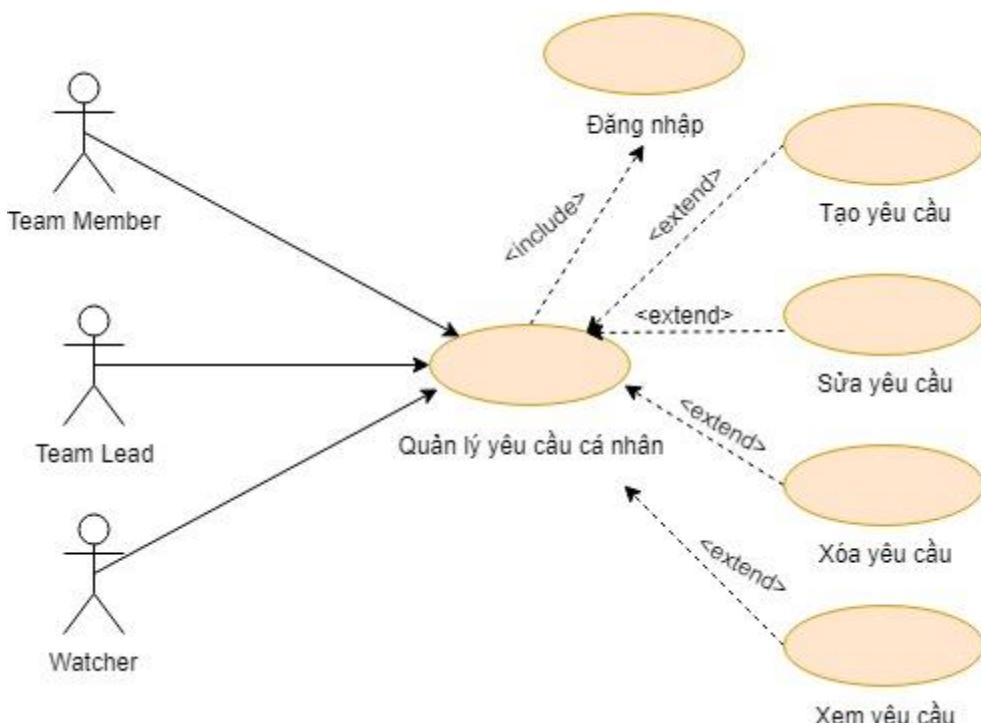
Hình 3.4.2. o. Biểu đồ chi tiết usecase thực hiện công việc được giao.

Bảng 3.4.2. n. Kích bản usecase thực hiện công việc được giao

1	Tên UC	Quản lý công việc trong module cá nhân
2	Tác nhân	Team member
3	Tiền điều kiện	Đăng nhập thành công vào hệ thống Đang trong chức năng xem thông tin dự án cá nhân
4	Mục đích	Quản lý các công việc cá nhân
5	Mô tả	Cho phép Team member quản lý các công việc liên quan đến bản thân trong dự án cá nhân
6	Luồng sự kiện chính	1. Team member chọn chức năng “Quản lý dự án cá nhân” 2. Hệ thống hiển thị form danh sách dự án cá nhân đã được tạo 3. Team member chọn chức năng xem dự án cá nhân và chọn chức năng xem danh sách công việc được giao

		<p>3.1. Team member chọn công việc được giao, chuyển trạng thái công việc và them phương án giải quyết công việc đó.</p> <p>3.2. Hệ thống kiểm tra. Nếu hợp lệ thì chuyển trạng thái công việc và them phương án giải quyết công việc.</p>
7	Luồng sự kiện phụ	<p>1. Nhập thông tin không hợp lệ</p> <p>2. Nhập thiếu thông tin</p>

3.4.2.16. Quản lý yêu cầu cá nhân



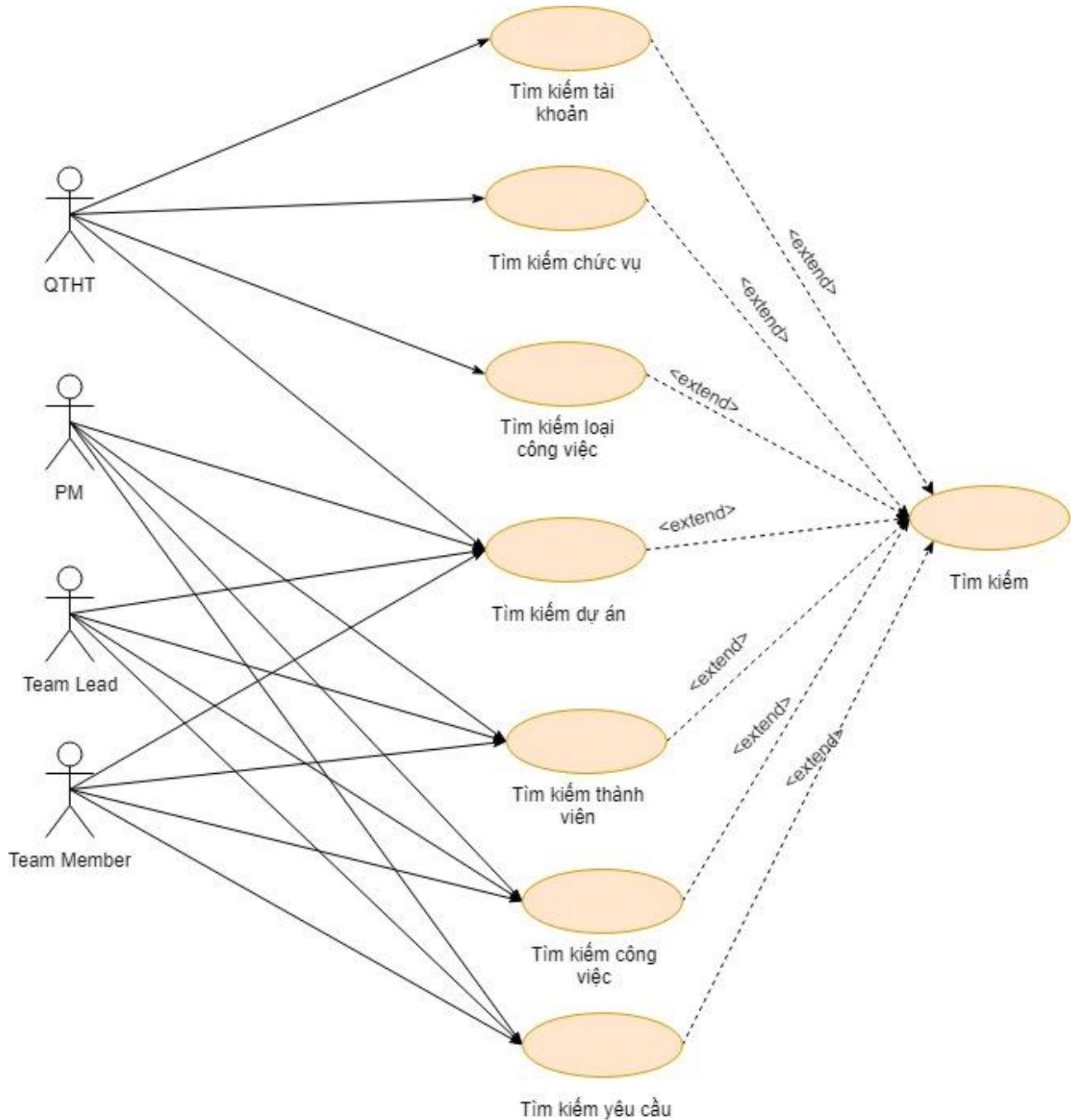
Hình 3.4.2. p. Biểu đồ chi tiết usecase quản lý yêu cầu cá nhân.

Bảng 3.4.2. o. Kích bản usecase quản lý yêu cầu cá nhân.

1	Tên UC	Quản lý công việc trong module cá nhân
2	Tác nhân	Team member, watcher, team leader
3	Tiền điều kiện	Đăng nhập thành công vào hệ thống
4	Mục đích	Quản lý các công việc cá nhân
5	Mô tả	Cho phép người dùng quản lý các yêu cầu

		được tạo bởi chính mình trong dự án.
6	Luồng sự kiện chính	<p>1. Người dùng chọn chức năng “Quản lý yêu cầu của tôi”</p> <p>2. Hệ thống hiển thị form danh sách yêu cầu cá nhân đã được tạo</p> <p>3. Người dùng chọn chức năng thêm mới yêu cầu</p> <p>3.1. Hệ thống hiển thị danh sách bộ phận có thể được chọn để chuyển yêu cầu tùy theo loại tác nhân</p> <p>3.2. Người dùng nhập thông tin yêu cầu sau đó chọn chức năng lưu</p> <p>3.3. Hệ thống kiểm tra. Nếu hợp lệ thì thêm yêu cầu mới.</p> <p>4. Người dùng chọn chức năng sửa yêu cầu mình đã tạo với yêu cầu chưa được xử lý</p> <p>4.1. Hệ thống hiển thị form sửa yêu cầu</p> <p>4.2. Người dùng nhập thông tin cần sửa</p> <p>4.3. Hệ thống kiểm tra. Nếu hợp lệ thì cập nhật thông tin yêu cầu</p>
7	Luồng sự kiện phụ	<p>1. Nhập thông tin không hợp lệ</p> <p>2. Nhập thiếu thông tin</p>

3.4.2.17. Tìm kiếm



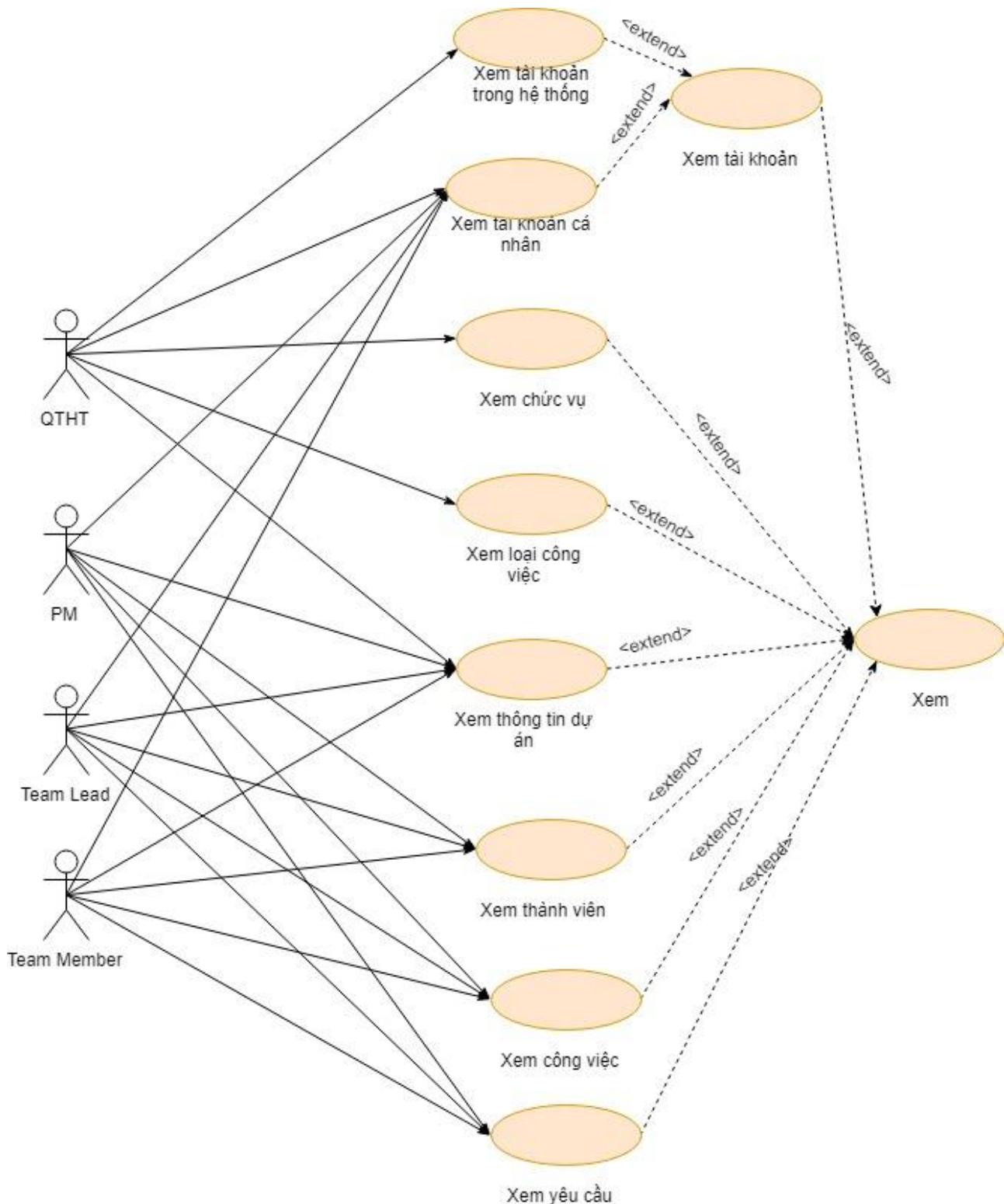
Hình 3.4.2. q. Biểu đồ chi tiết usecase tìm kiếm.

Bảng 3.4.2. p. Kích bản usecase tìm kiếm

1	Tên UC	Tìm kiếm
2	Tác nhân	QTHT, PM, Team Lead, Team member
3	Tiền điều kiện	Đăng nhập thành công vào hệ thống

4	Mục đích	Tìm kiếm các thông tin cần thiết
5	Mô tả	Người dùng nhập từ khóa tìm kiếm và hệ thống sẽ trả về thông tin cần tìm
6	Luồng sự kiện chính	<ol style="list-style-type: none"> 1. Người sử dụng nhập từ khóa cần tìm kiếm và gửi yêu cầu lên hệ thống 2. Hệ thống kiểm tra thông tin và hiển thị danh sách tìm kiếm
7	Luồng sự kiện phụ	1. Không tìm thấy thông tin

3.4.2.18. Xem



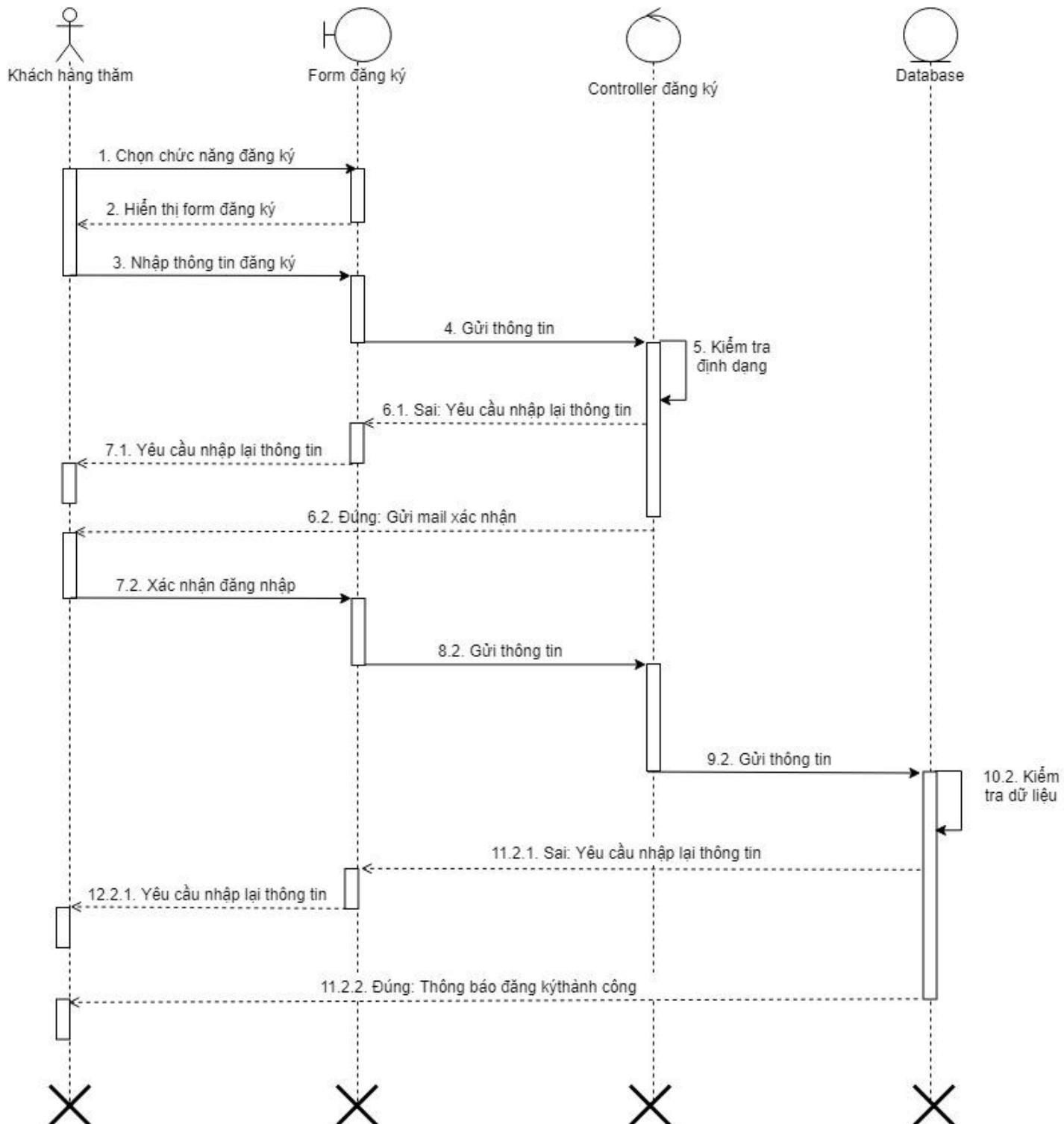
Hình 3.4.2. r. Biểu đồ chi tiết usecase xem.

Bảng 3.4.2. q.Kích bản usecase xem

1	Tên UC	Xem
2	Tác nhân	QTHT, PM, Team Lead. Team member
3	Tiền điều kiện	Đăng nhập thành công vào hệ thống
4	Mục đích	Xem chi tiết các thông tin cần thiết
5	Mô tả	Người dùng muốn xem thông tin các mục mà mình lựa chọn
6	Luồng sự kiện chính	<ol style="list-style-type: none">1. Người sử dụng vào trang cần xem2. Hệ thống hiển thị trang mà người dùng yêu cầu
7	Luồng sự kiện phụ	<ol style="list-style-type: none">1. Thông tin đã bị xóa

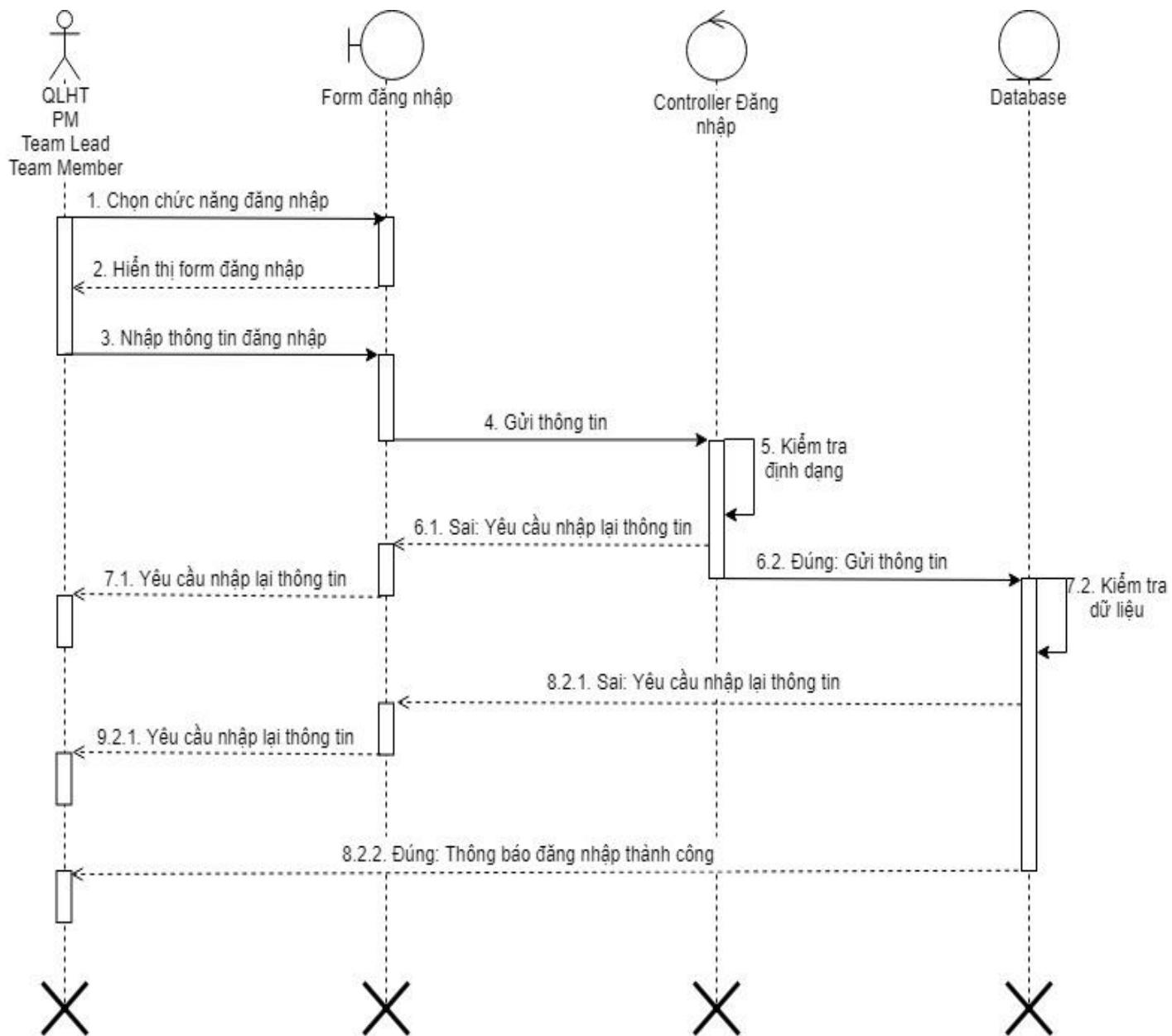
3.4.3. Biểu đồ trình tự

3.4.3.1. Biểu đồ trình tự đăng ký



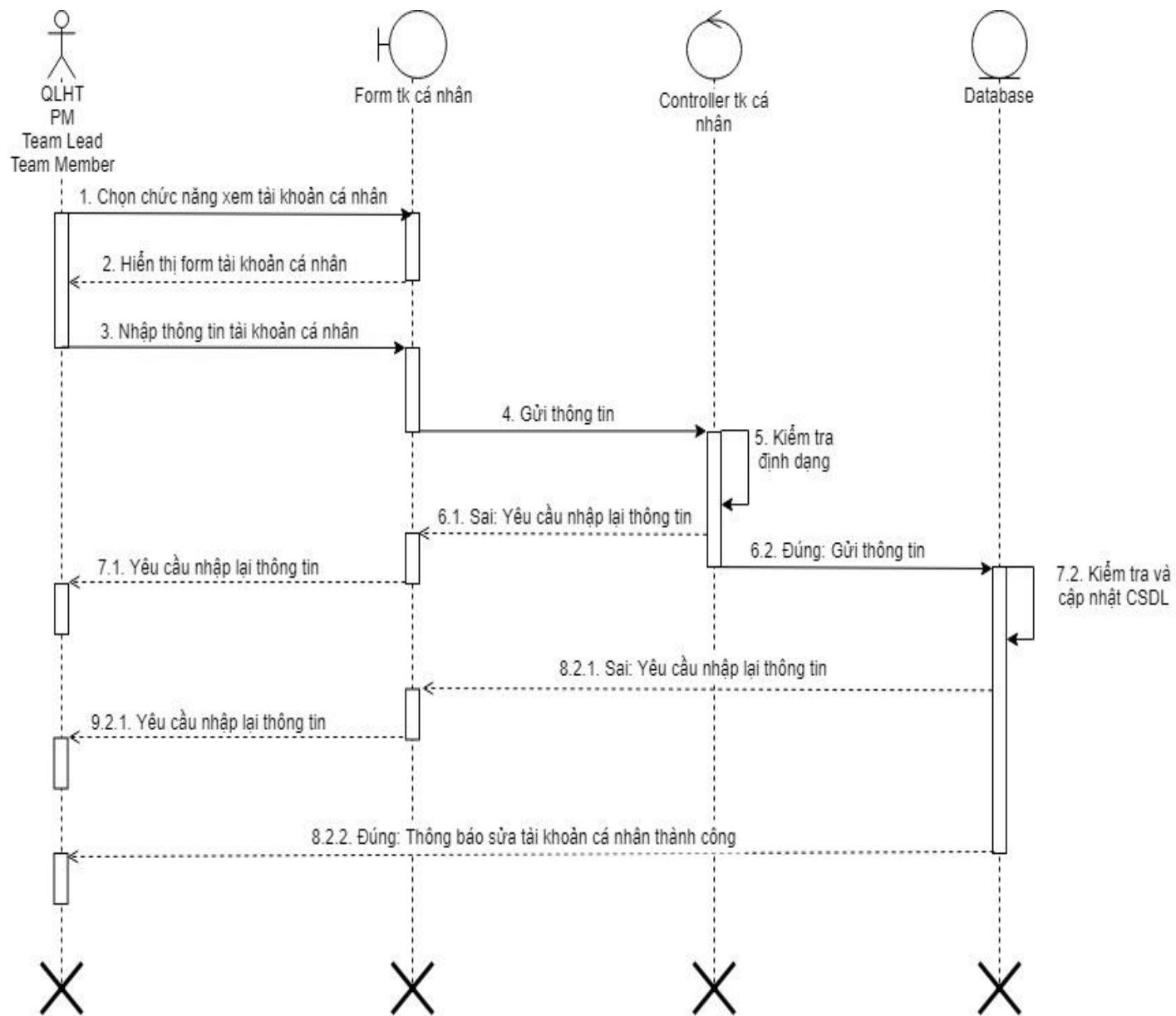
Hình 3.4.3. a. Biểu đồ trình tự đăng ký.

3.4.3.2. Biểu đồ trình tự đăng nhập



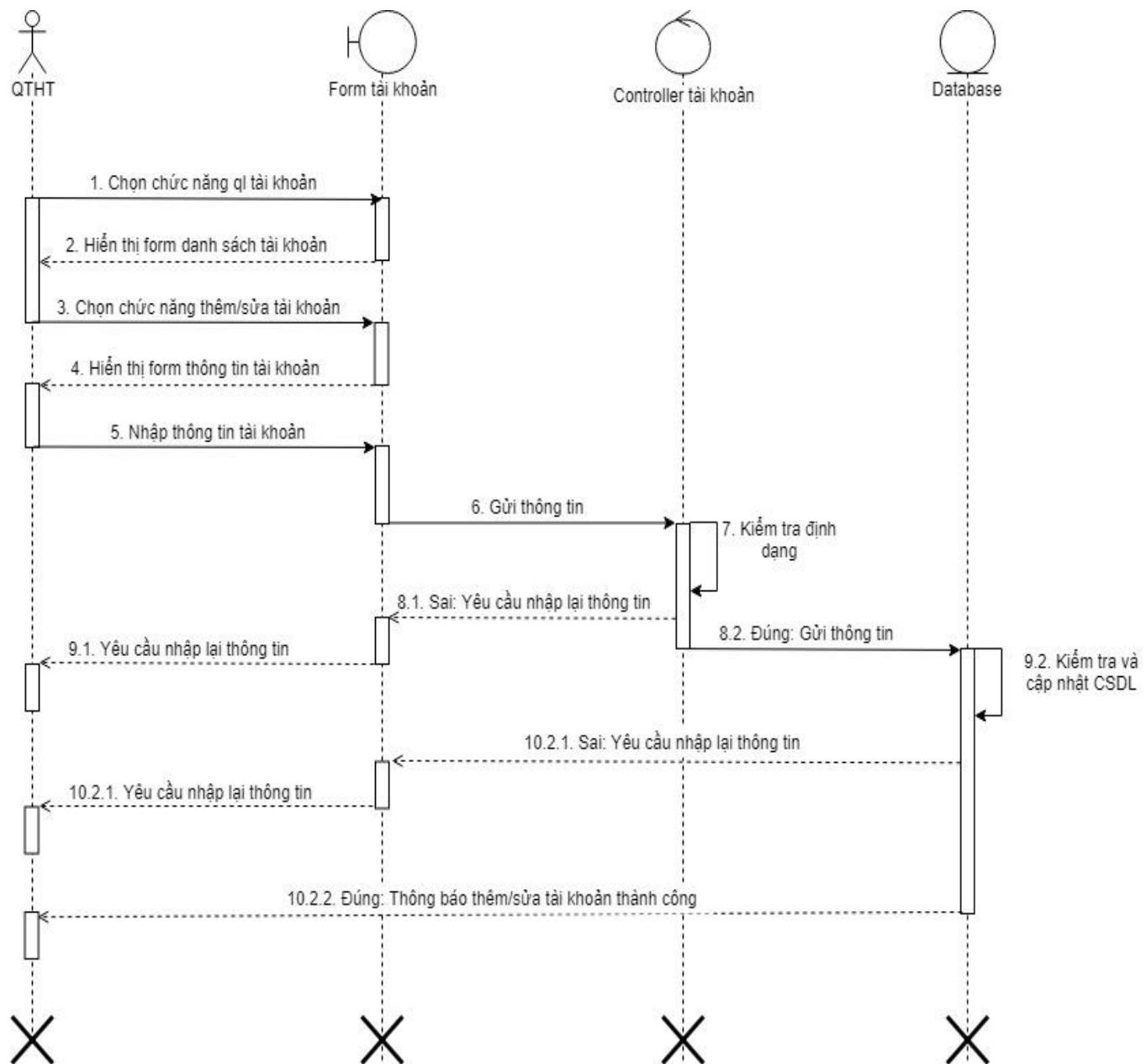
Hình 3.4.3. b. Biểu đồ trình tự đăng nhập

3.4.3.3. Biểu đồ trình tự quản lý tài khoản cá nhân

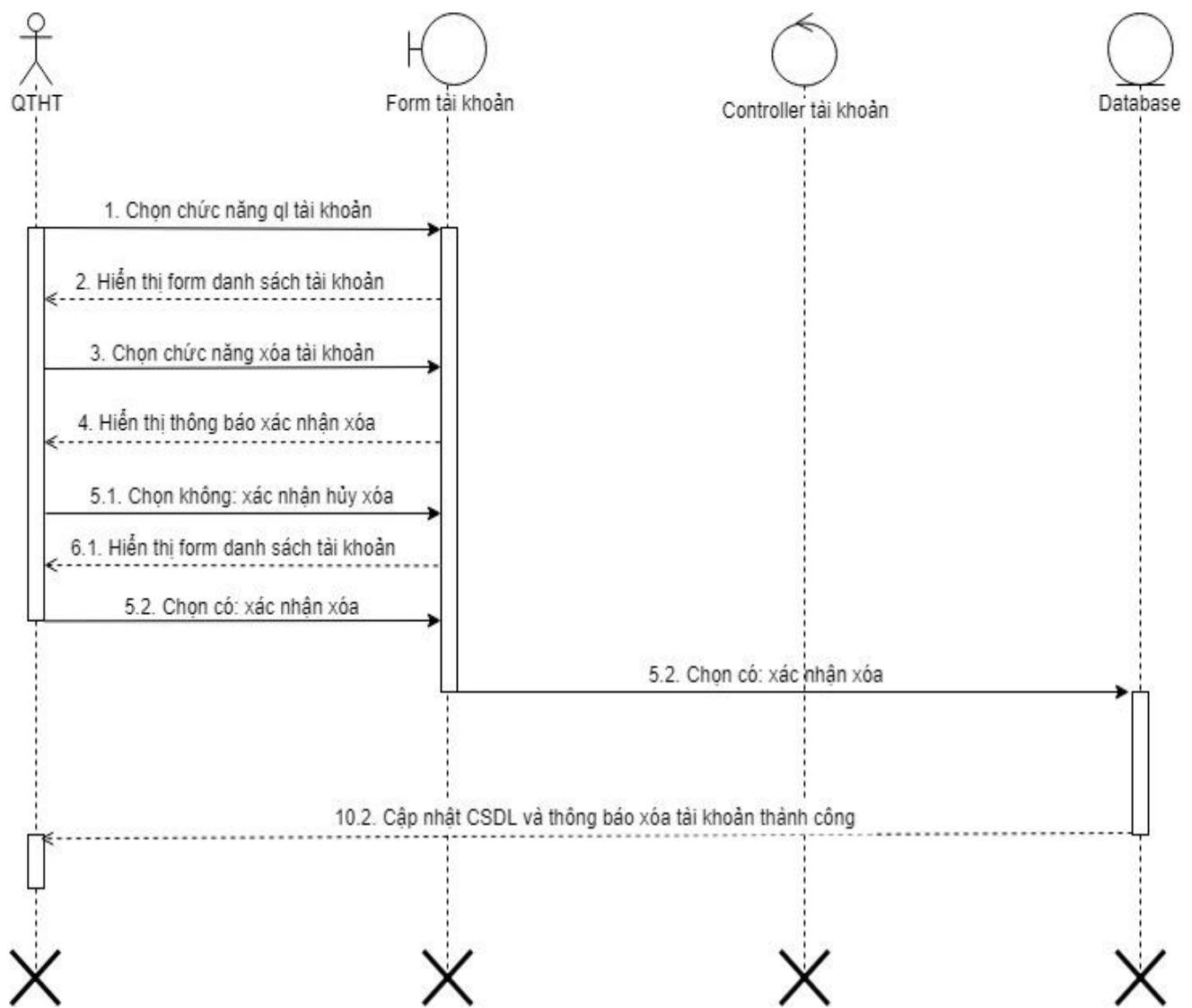


Hình 3.4.3. c. Biểu đồ trình tự quản lý tài khoản cá nhân.

3.4.3.4. Biểu đồ trình tự quản lý tài khoản trong hệ thống

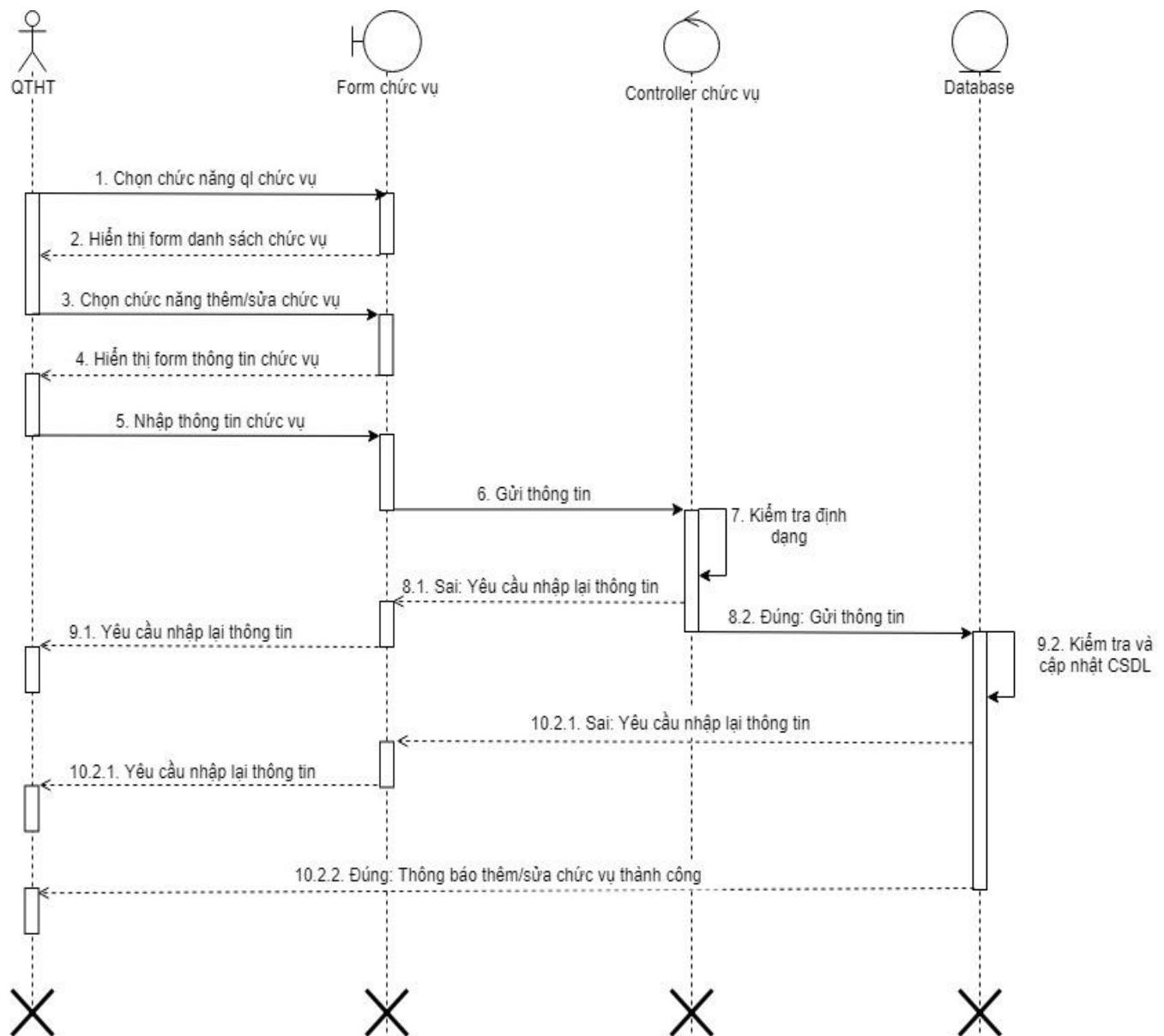


Hình 3.4.3. d. Biểu đồ trình tự thêm/sửa tài khoản trong hệ thống.

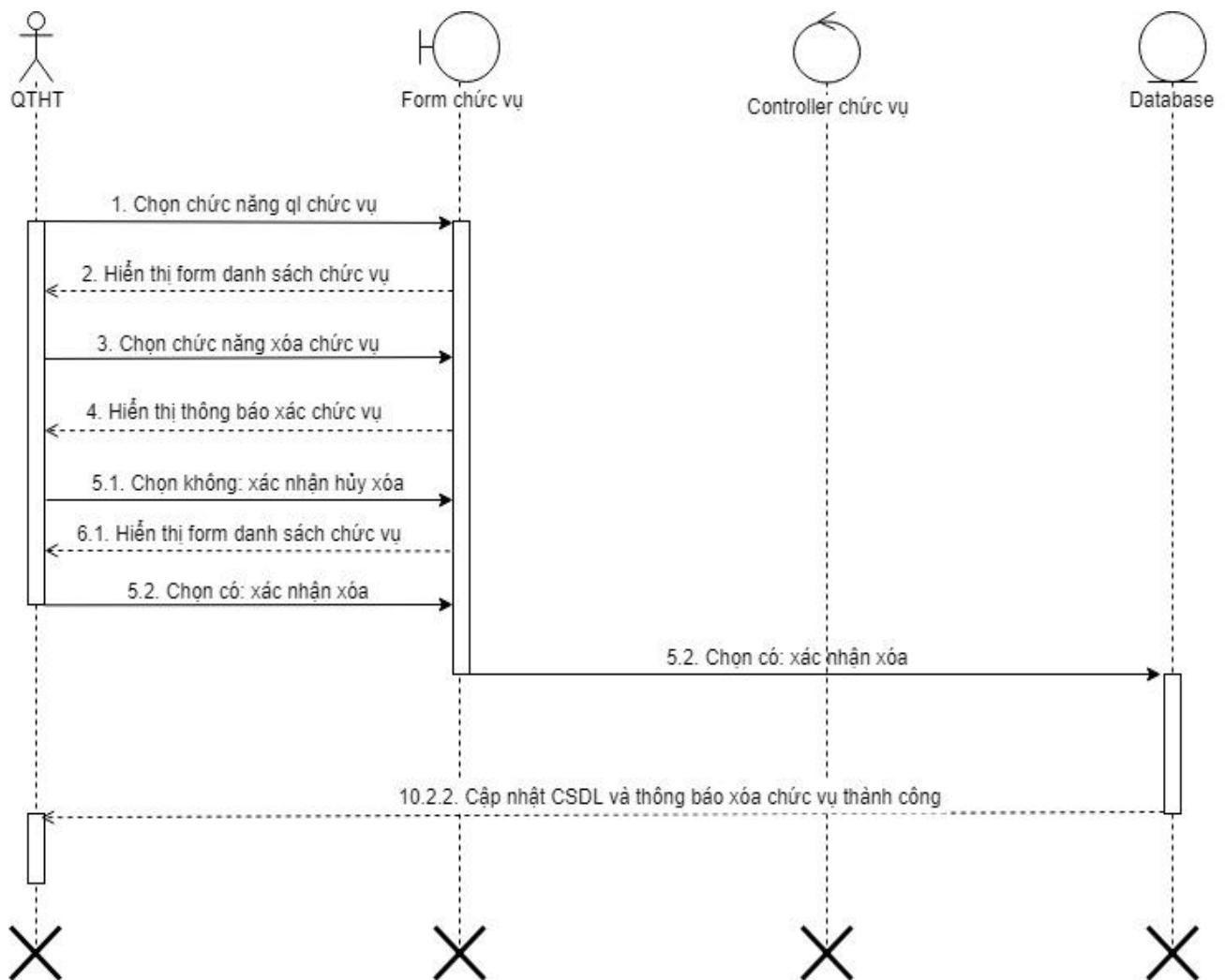


Hình 3.4.3. e. Biểu đồ trình tự xóa tài khoản trong hệ thống.

3.4.3.5. Biểu đồ trình tự quản lý chức vụ trong ngành CNTT

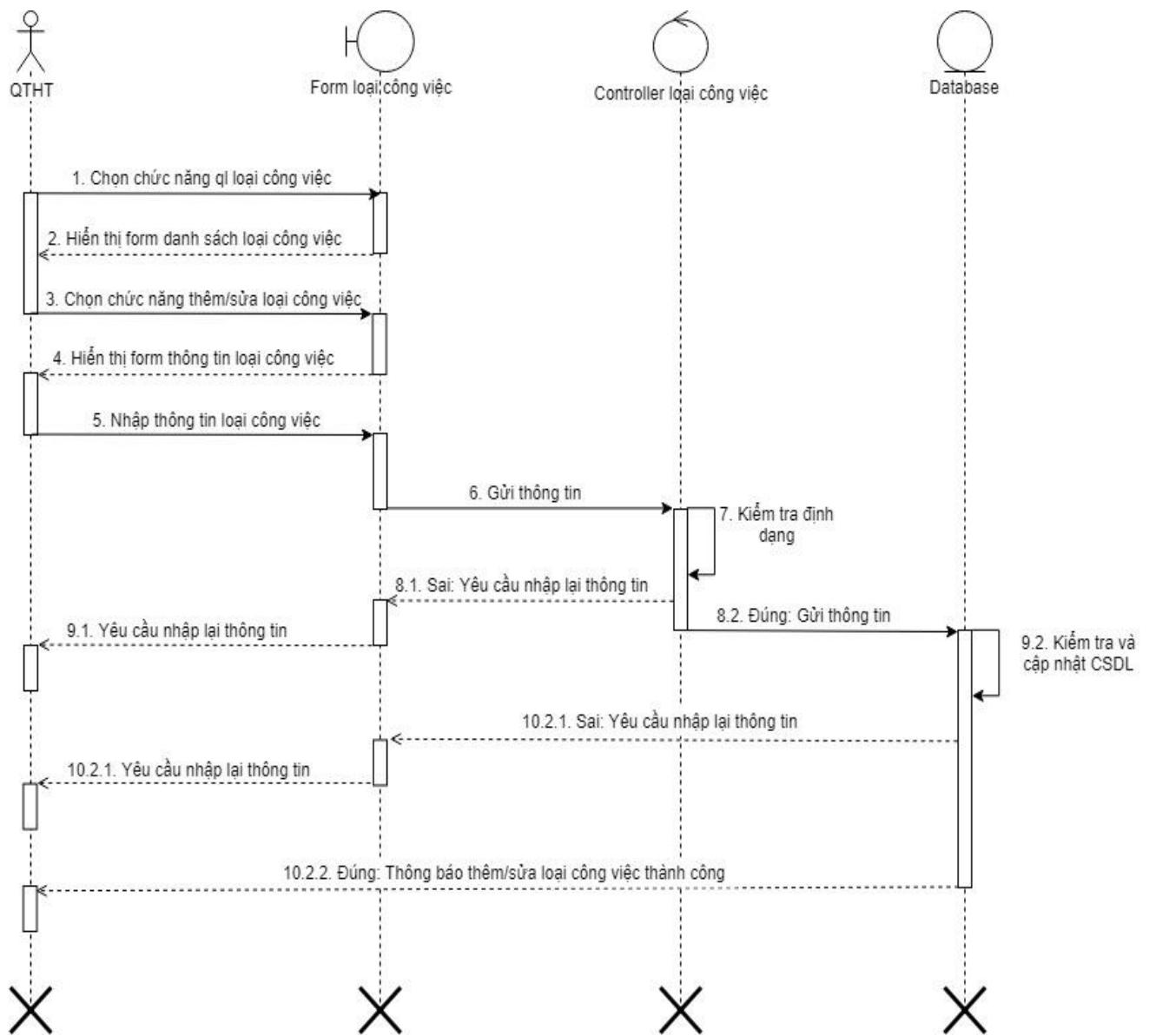


Hình 3.4.3. f. Biểu đồ trình tự thêm/sửa chức vụ.

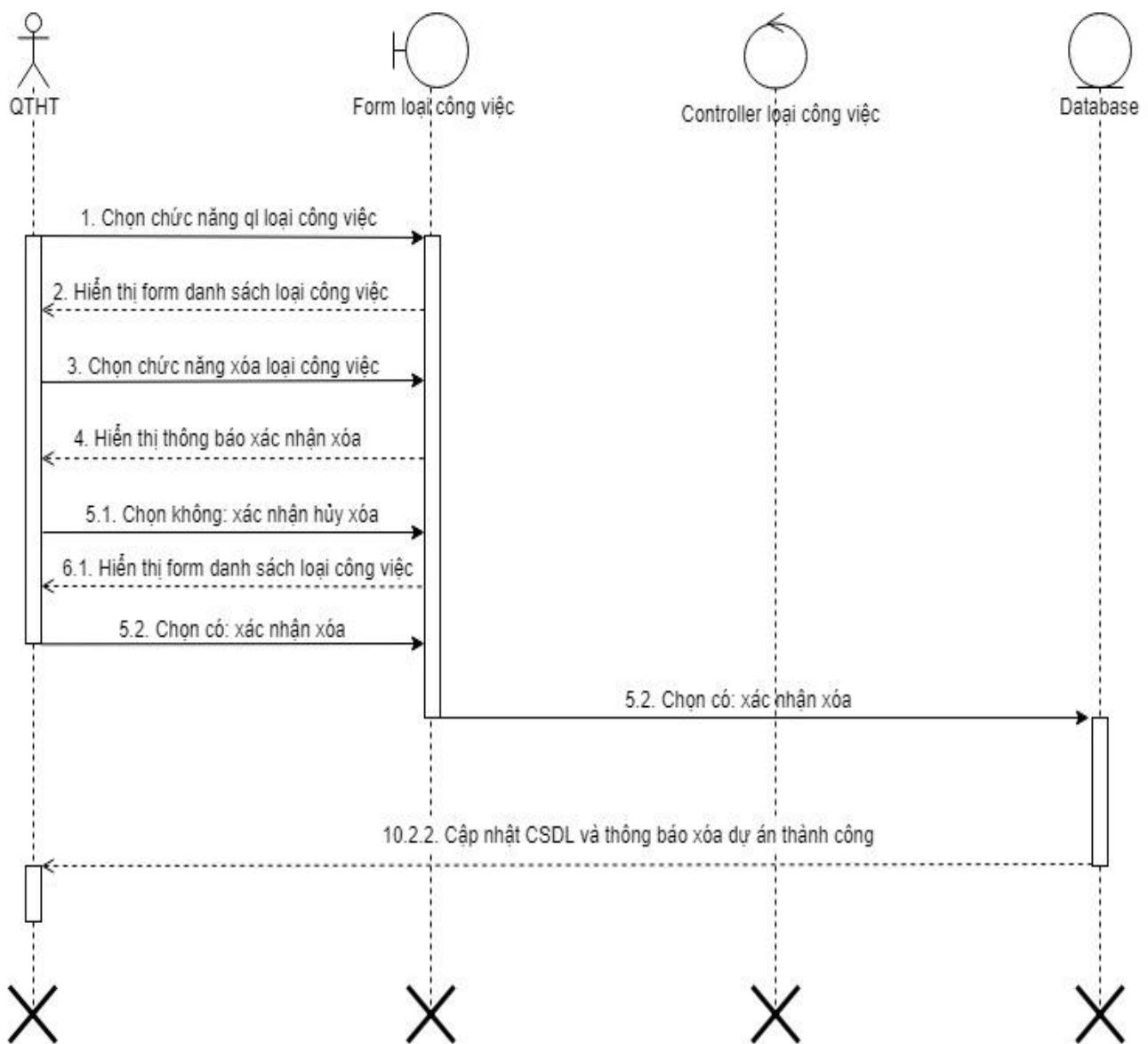


Hình 3.4.3. g. Biểu đồ trình tự xóa chức vụ.

3.4.3.6. Biểu đồ trình tự quản lý loại công việc

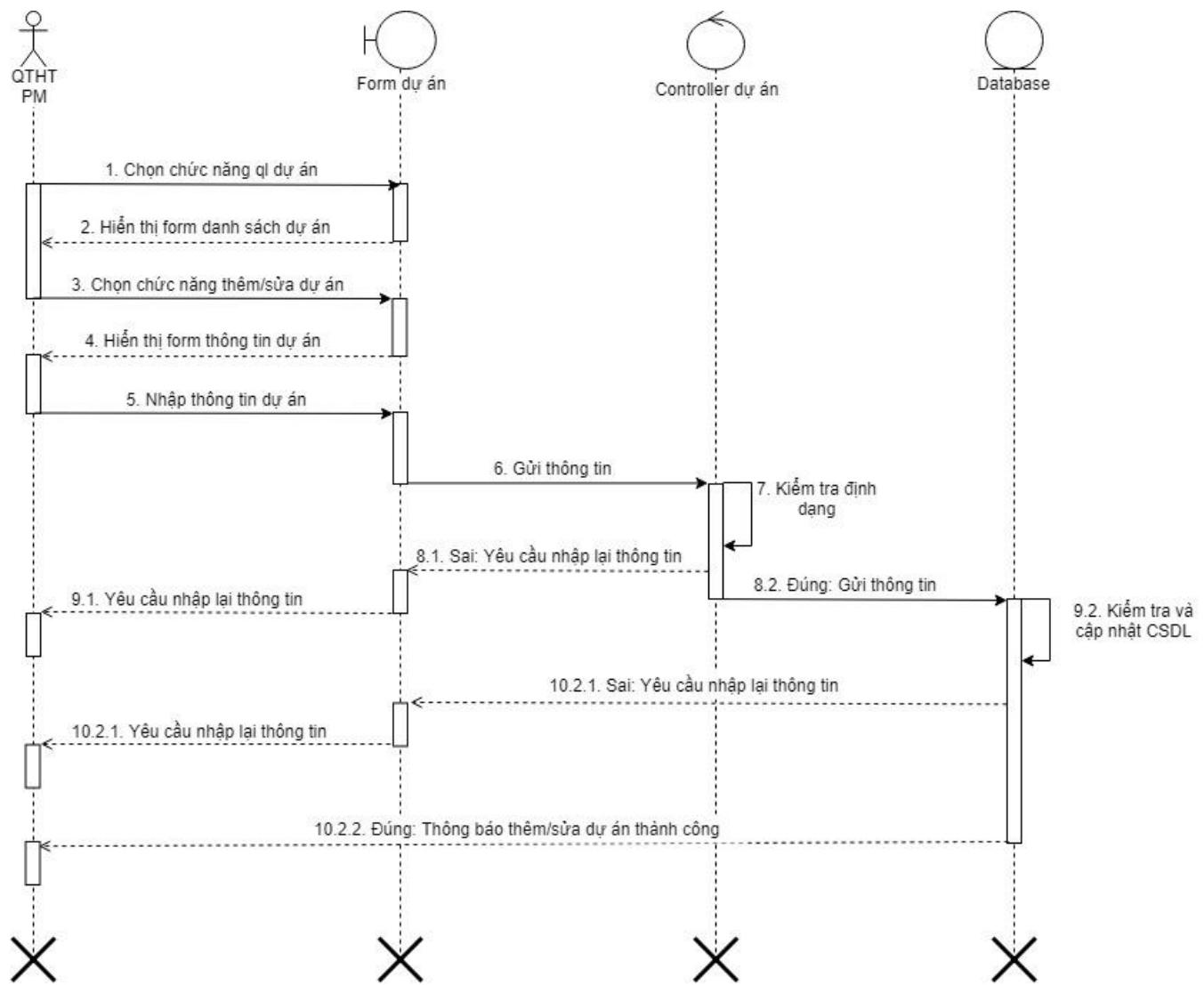


Hình 3.4.3. h. Biểu đồ trình tự thêm/sửa loại công việc.

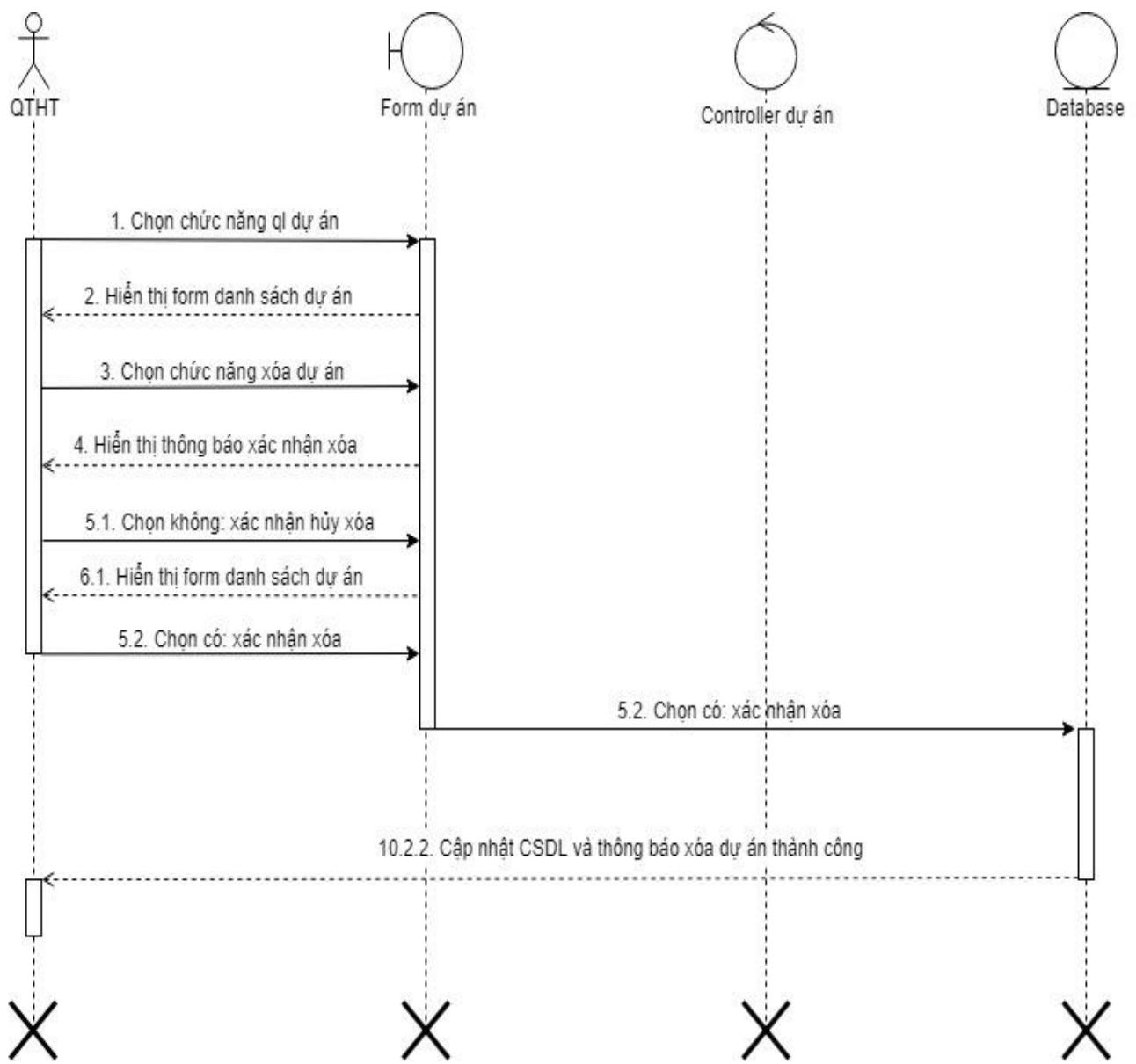


Hình 3.4.3. i. Biểu đồ trình tự xóa loại công việc.

3.4.3.7. Biểu đồ trình tự quản lý dự án trong hệ thống

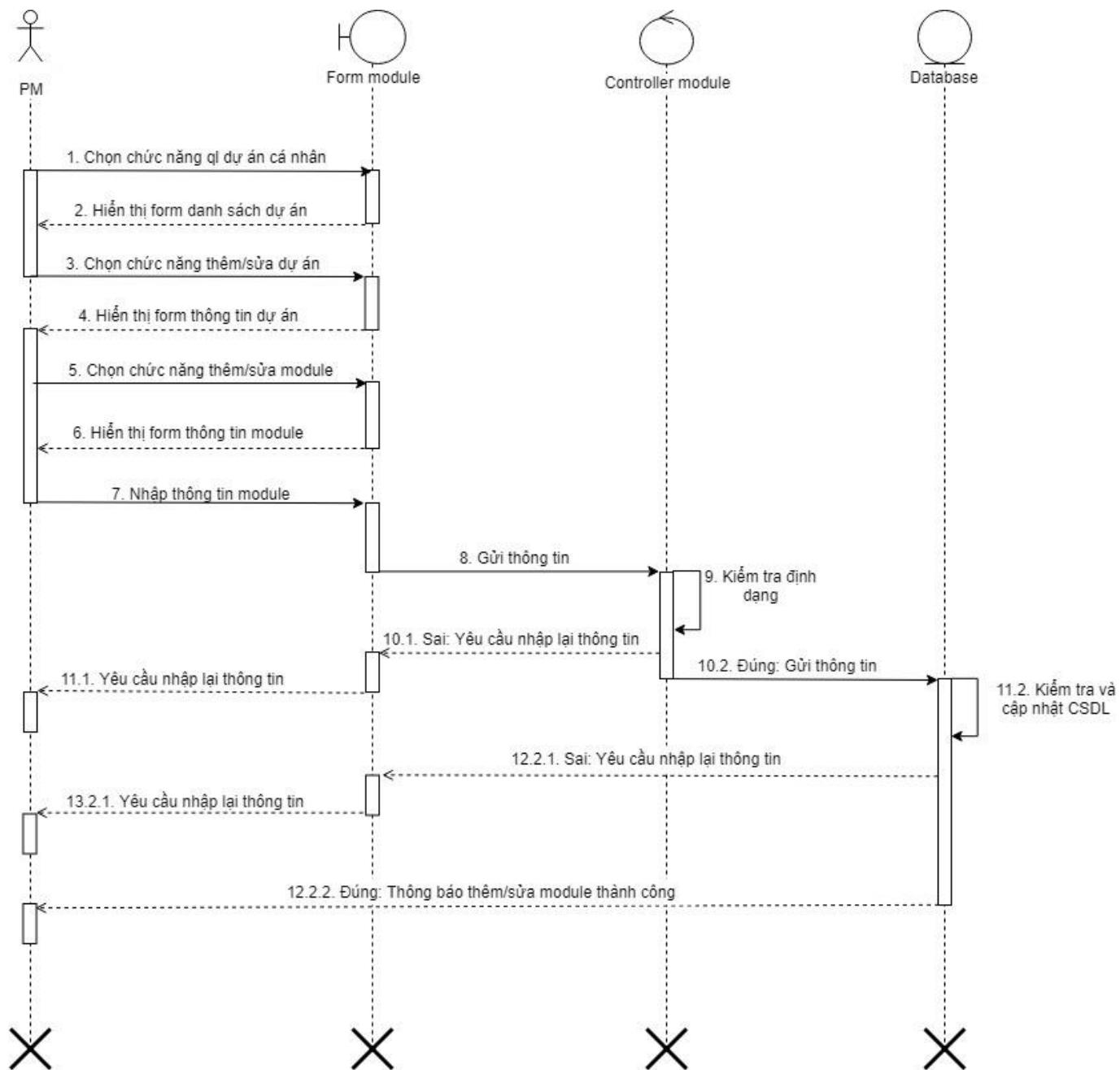


Hình 3.4.3. j. Biểu đồ trình tự thêm/sửa dự án trong hệ thống.

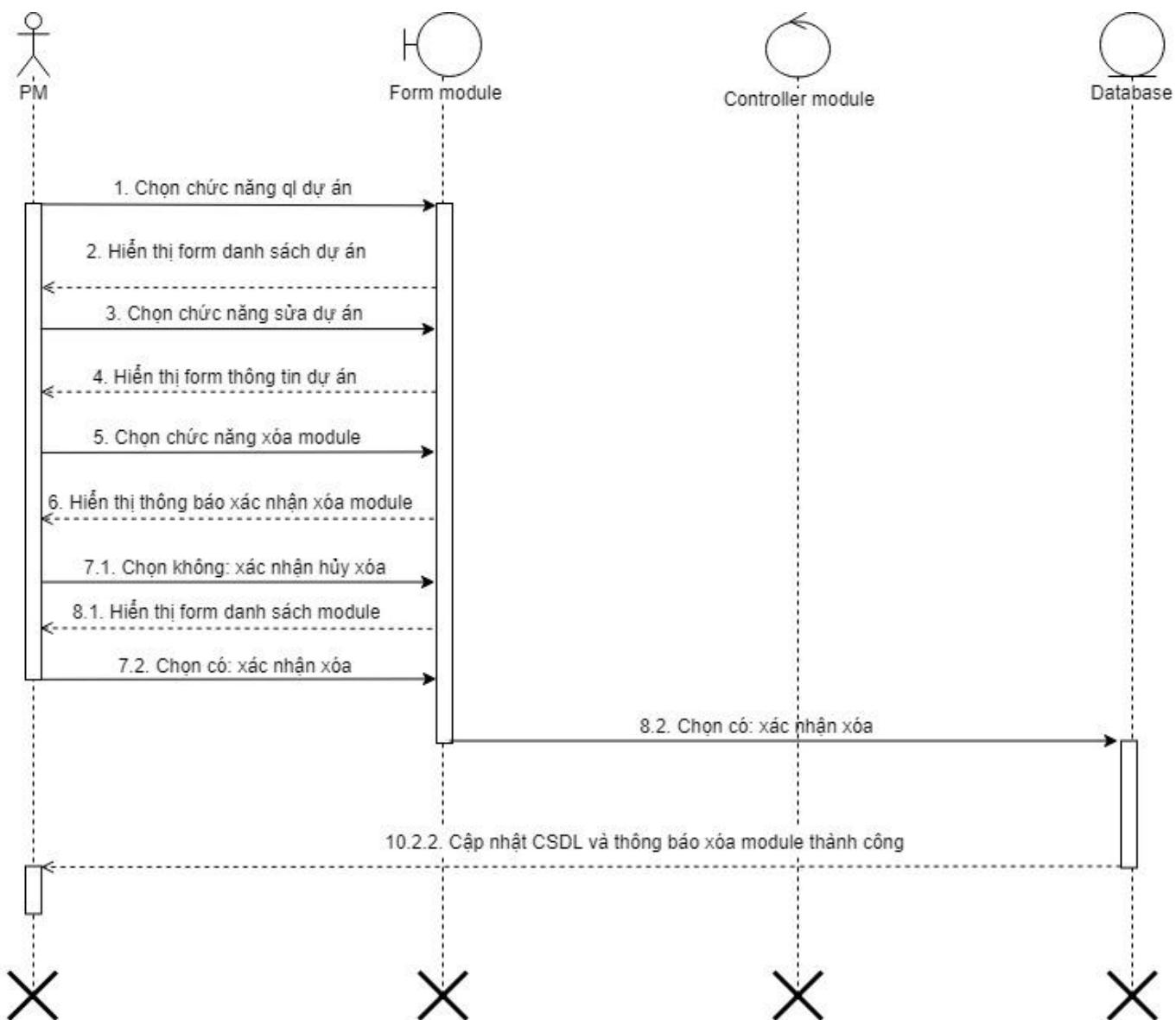


Hình 3.4.3. k. Biểu đồ trình tự xóa dự án trong hệ thống.

3.4.3.8. Biểu đồ trình tự quản lý module trong dự án cá nhân

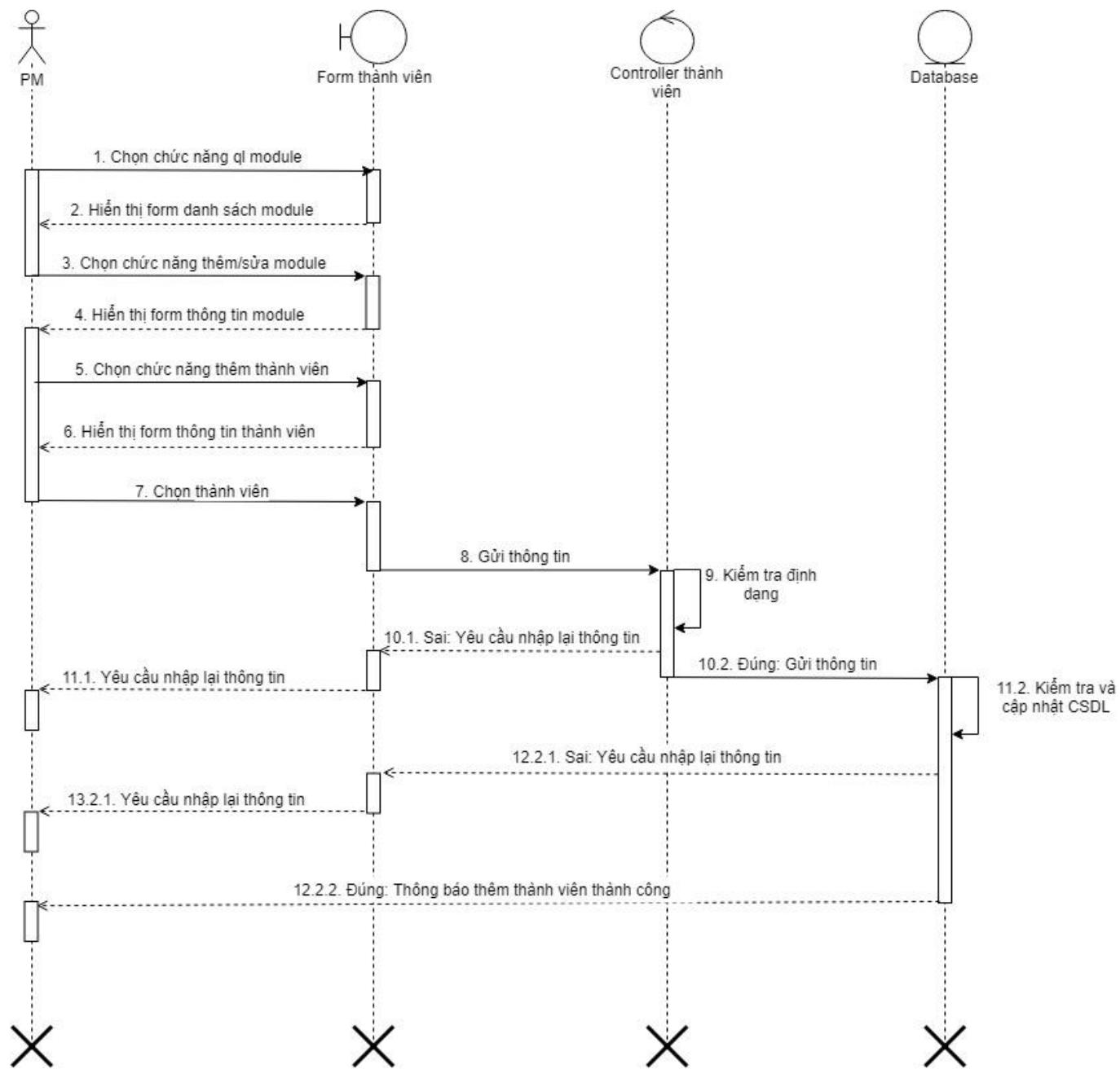


Hình 3.4.3. l. Biểu đồ trình tự thêm/sửa module.

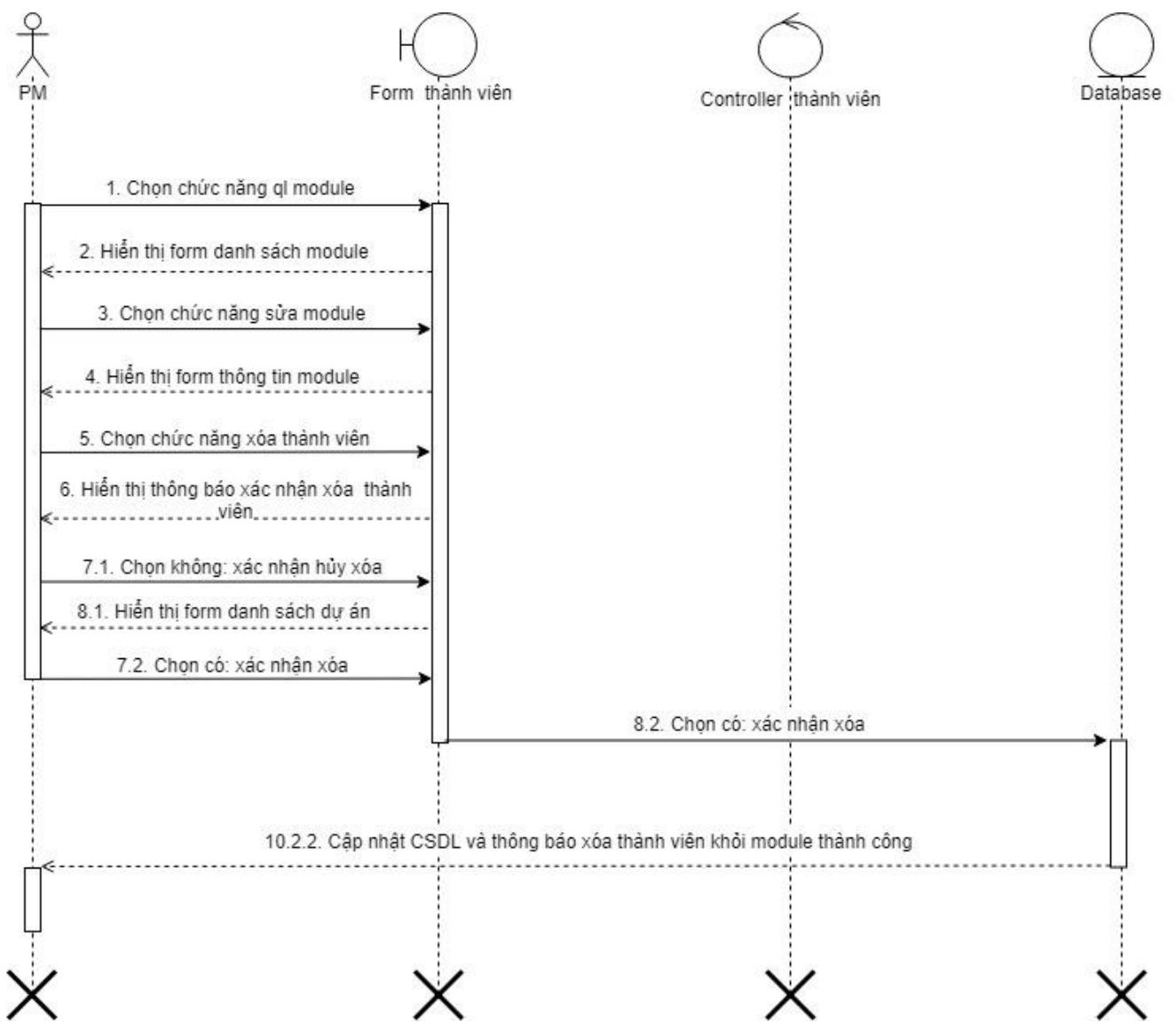


Hình 3.4.3. m. Biểu đồ trình tự xóa module.

3.4.3.9. Biểu đồ trình tự quản lý thành viên trong module

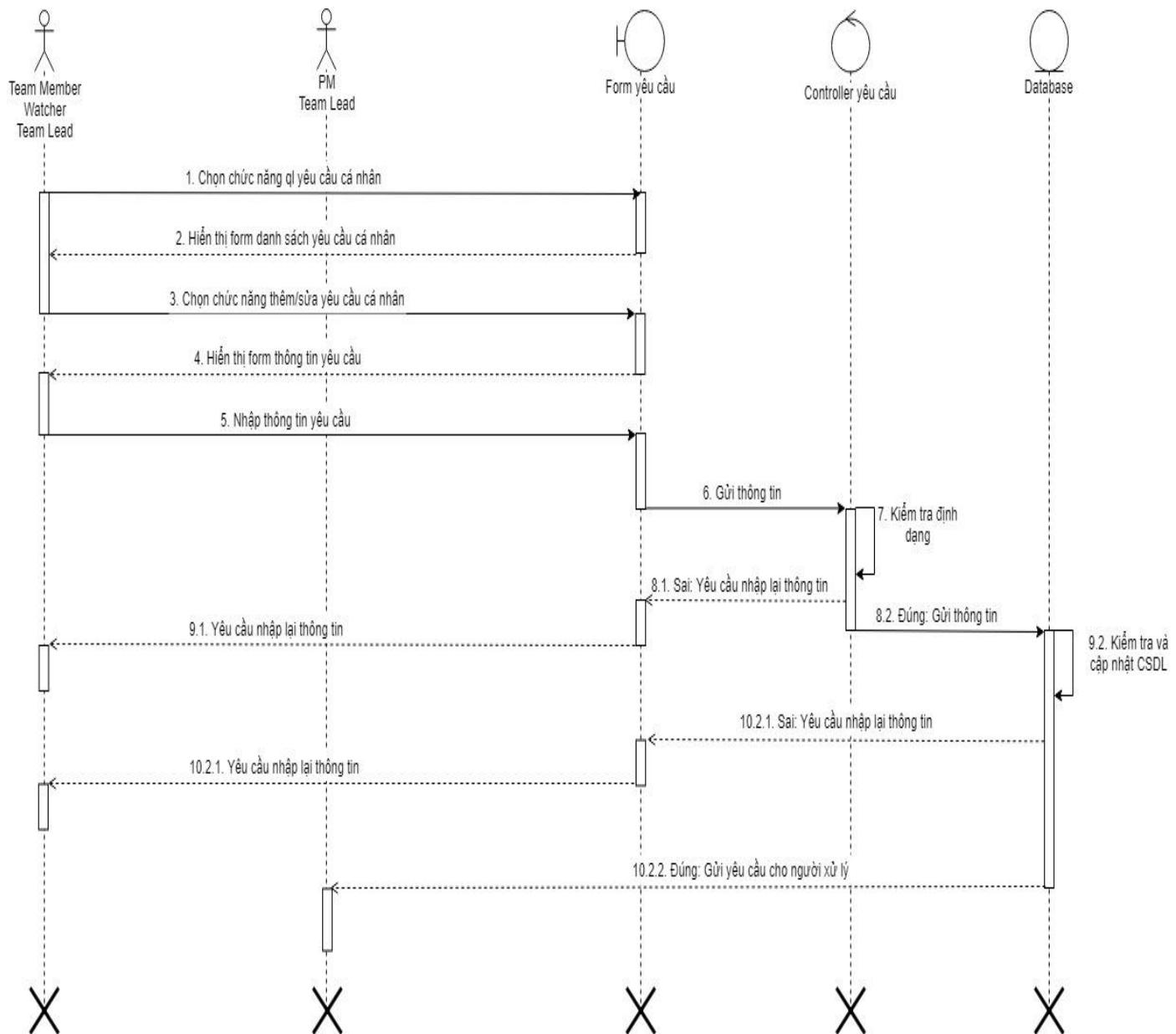


Hình 3.4.3. n. Biểu đồ trình tự thêm thành viên vào module.

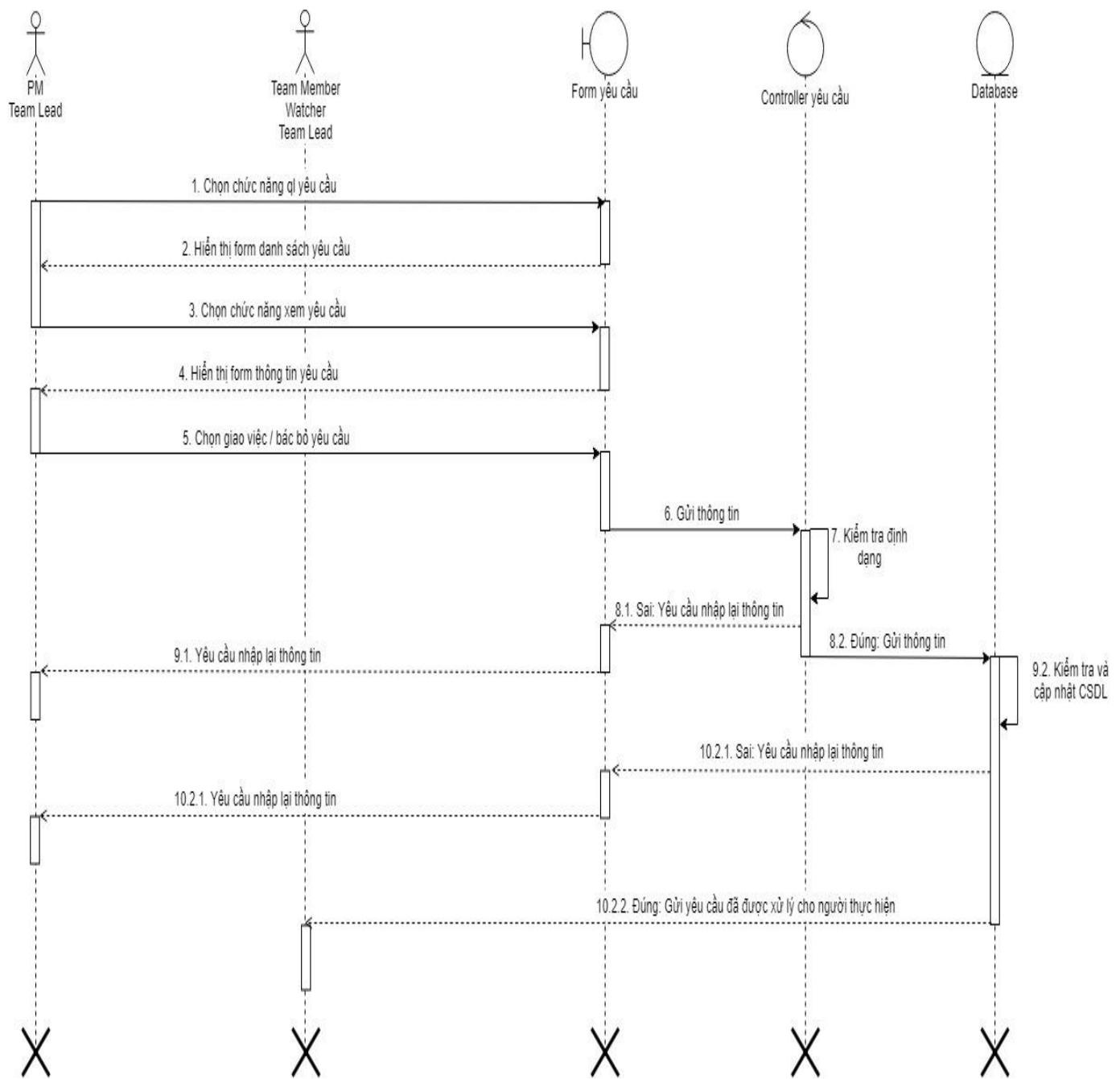


Hình 3.4.3. o. Biểu đồ trình tự xóa thành viên khỏi module.

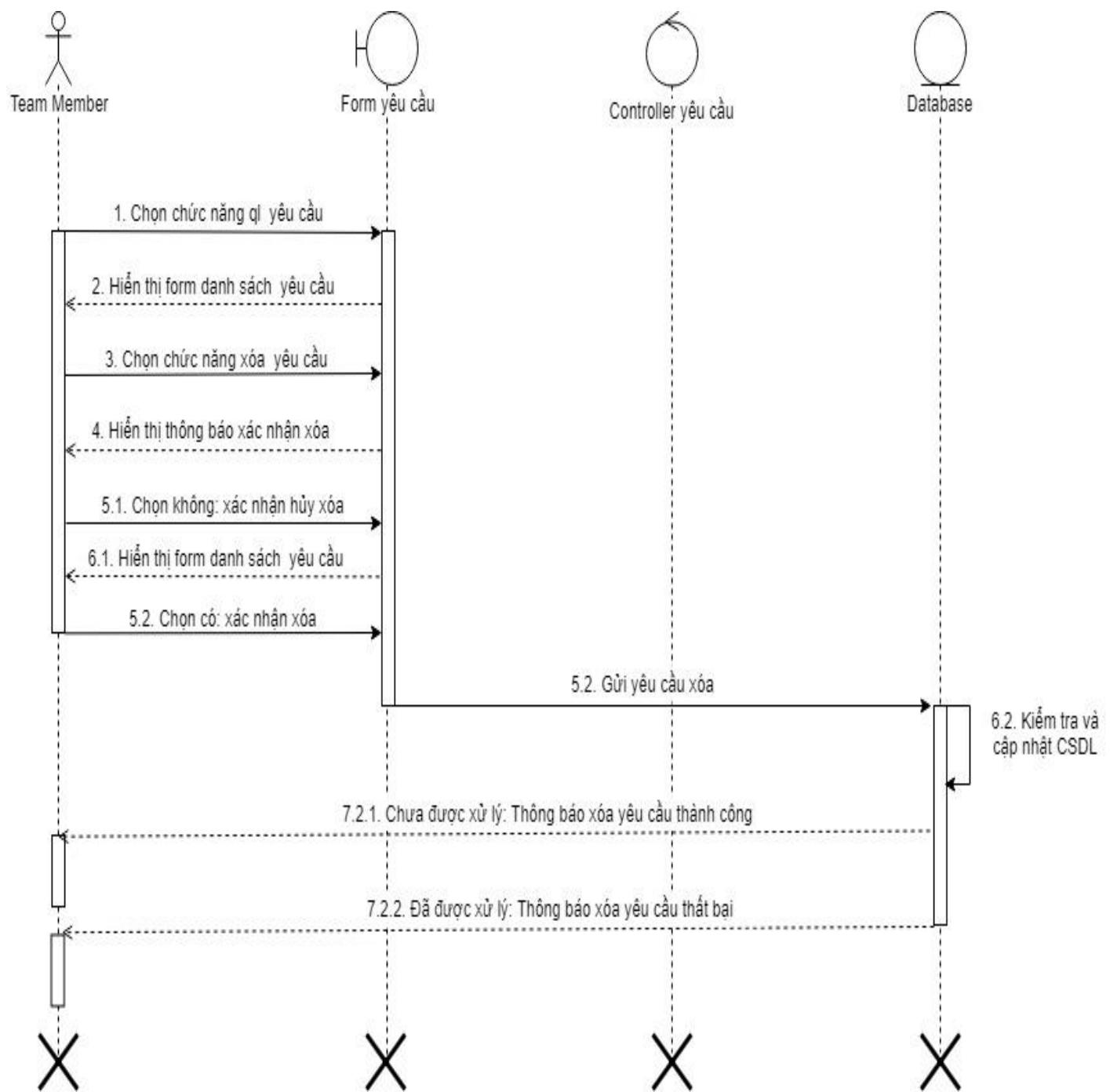
3.4.3.10. Biểu đồ trình tự quản lý yêu cầu



Hình 3.4.3. p. Biểu đồ trình tự thêm/sửa yêu cầu.

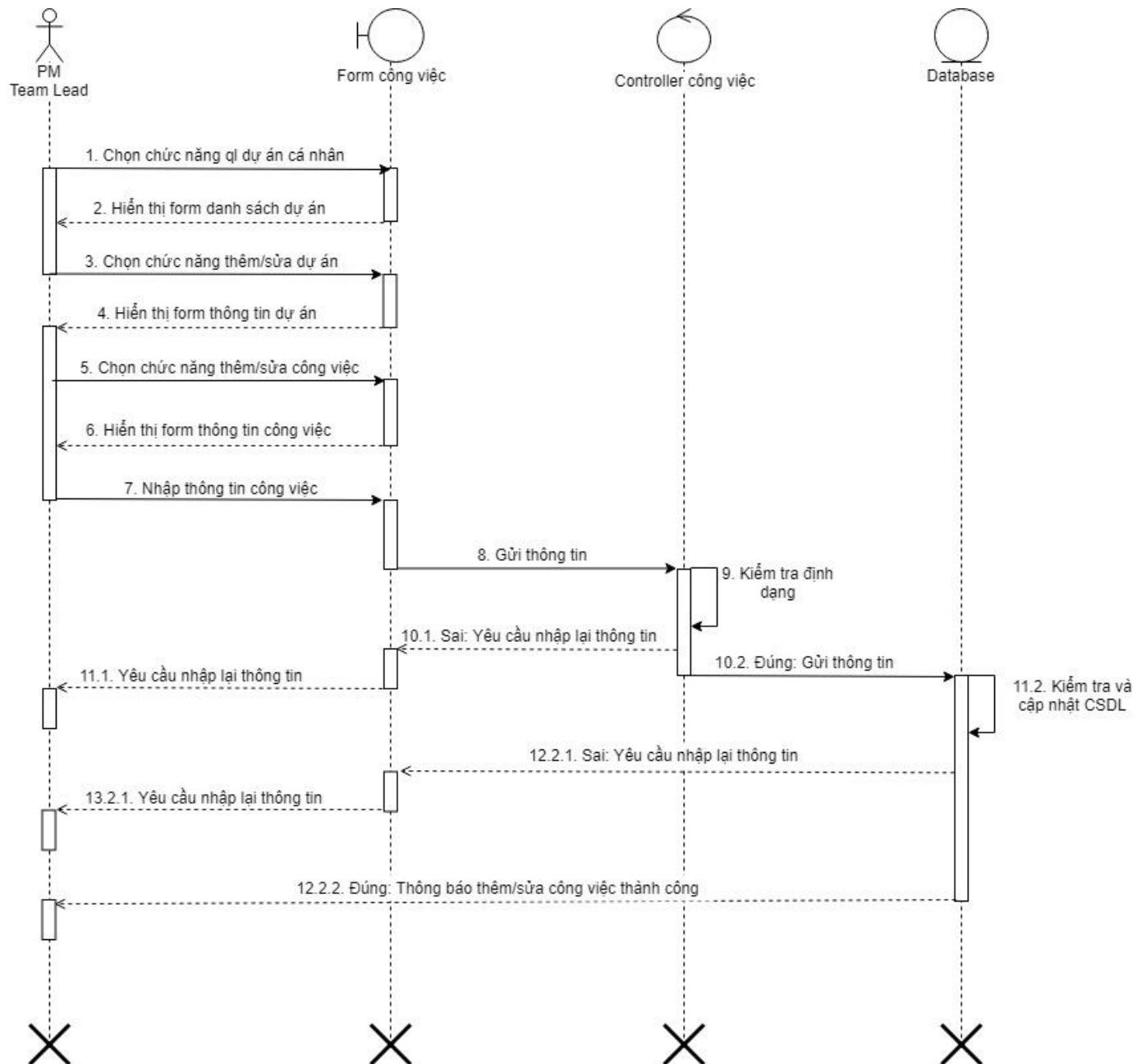


Hình 3.4.3. q. Biểu đồ trình tự xử lý yêu cầu.

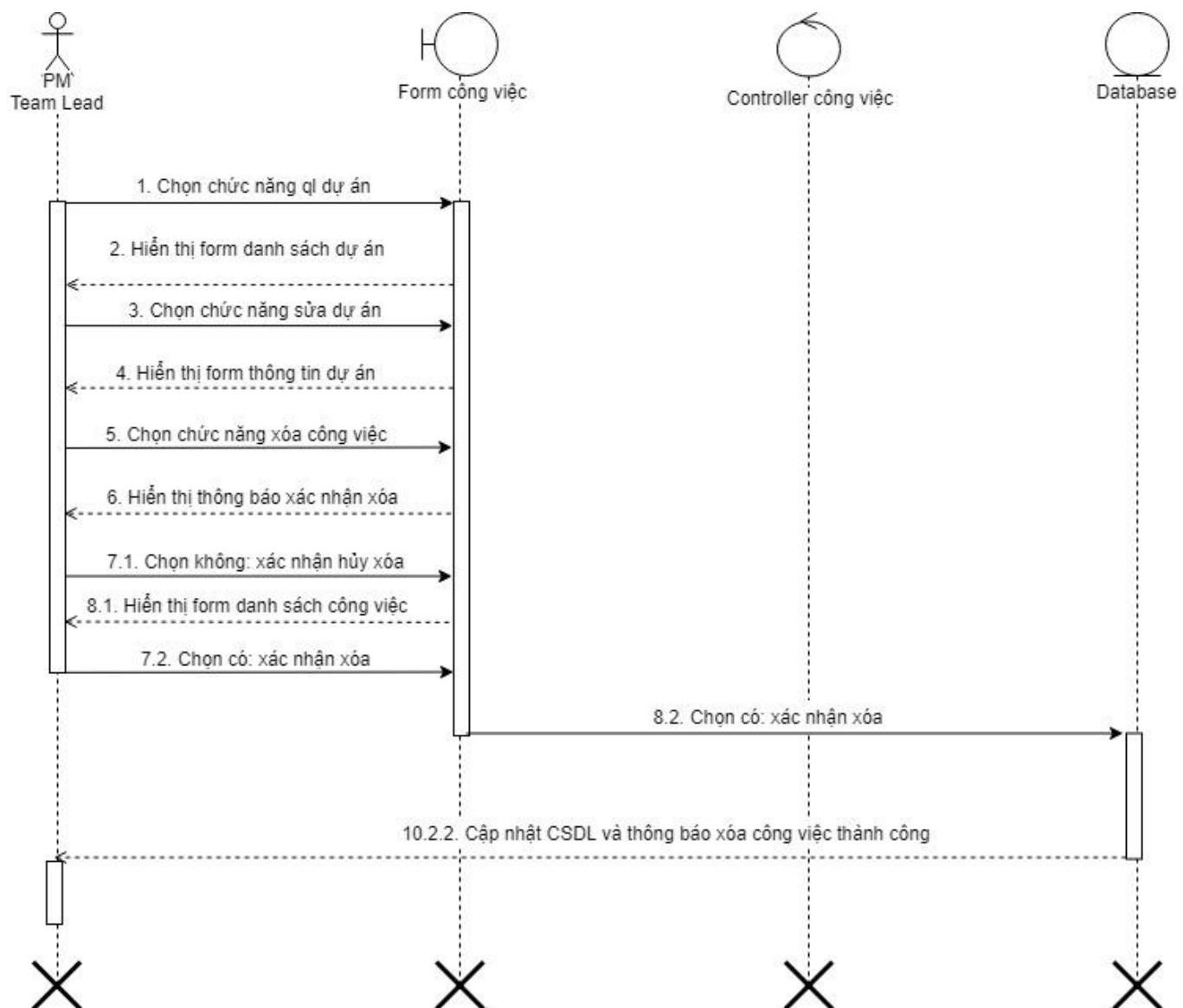


Hình 3.4.3. r. Biểu đồ trình tự xóa yêu cầu.

3.4.3.11. Biểu đồ trình tự quản lý công việc

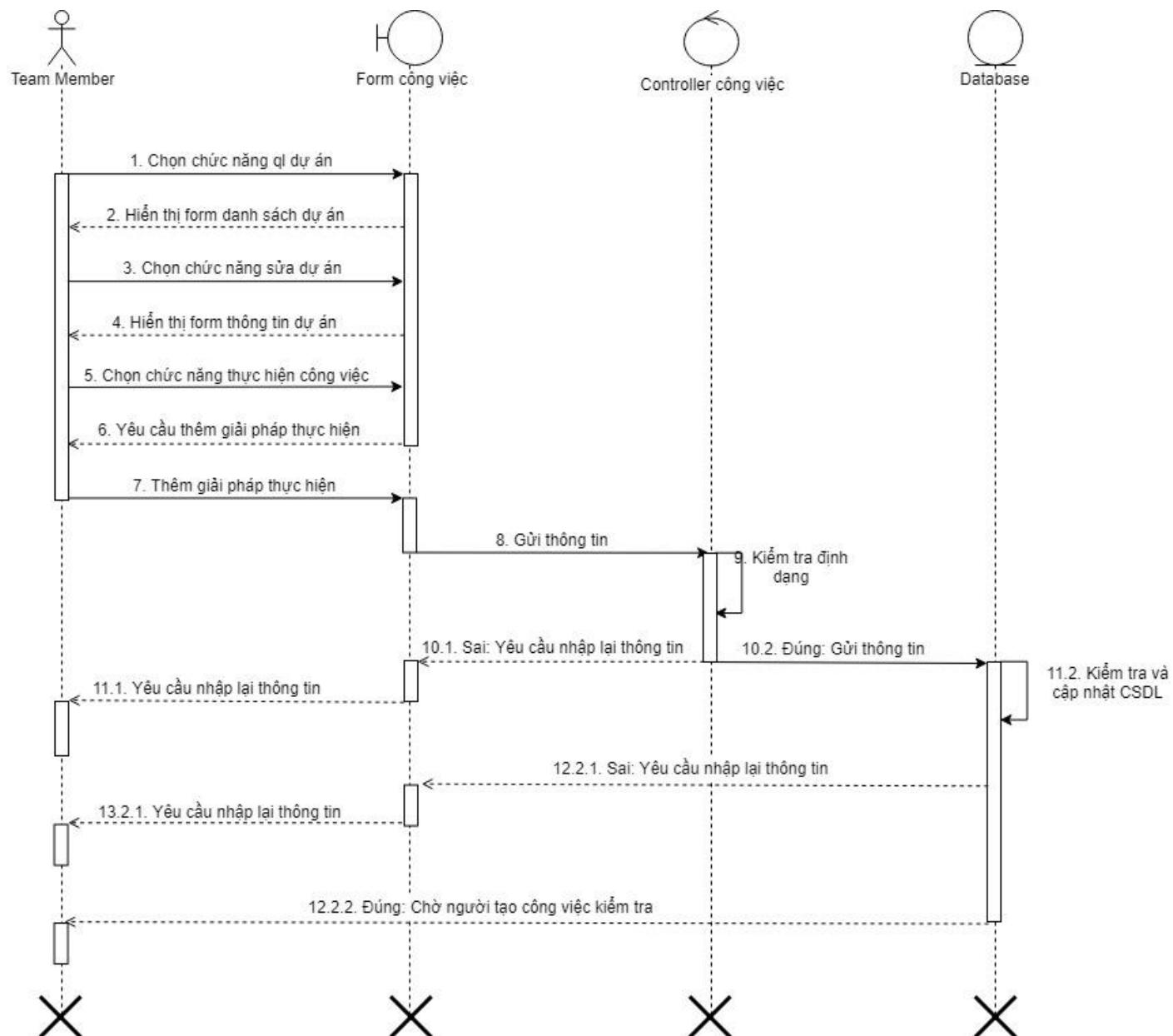


Hình 3.4.3. s. Biểu đồ trình tự thêm/sửa công việc.



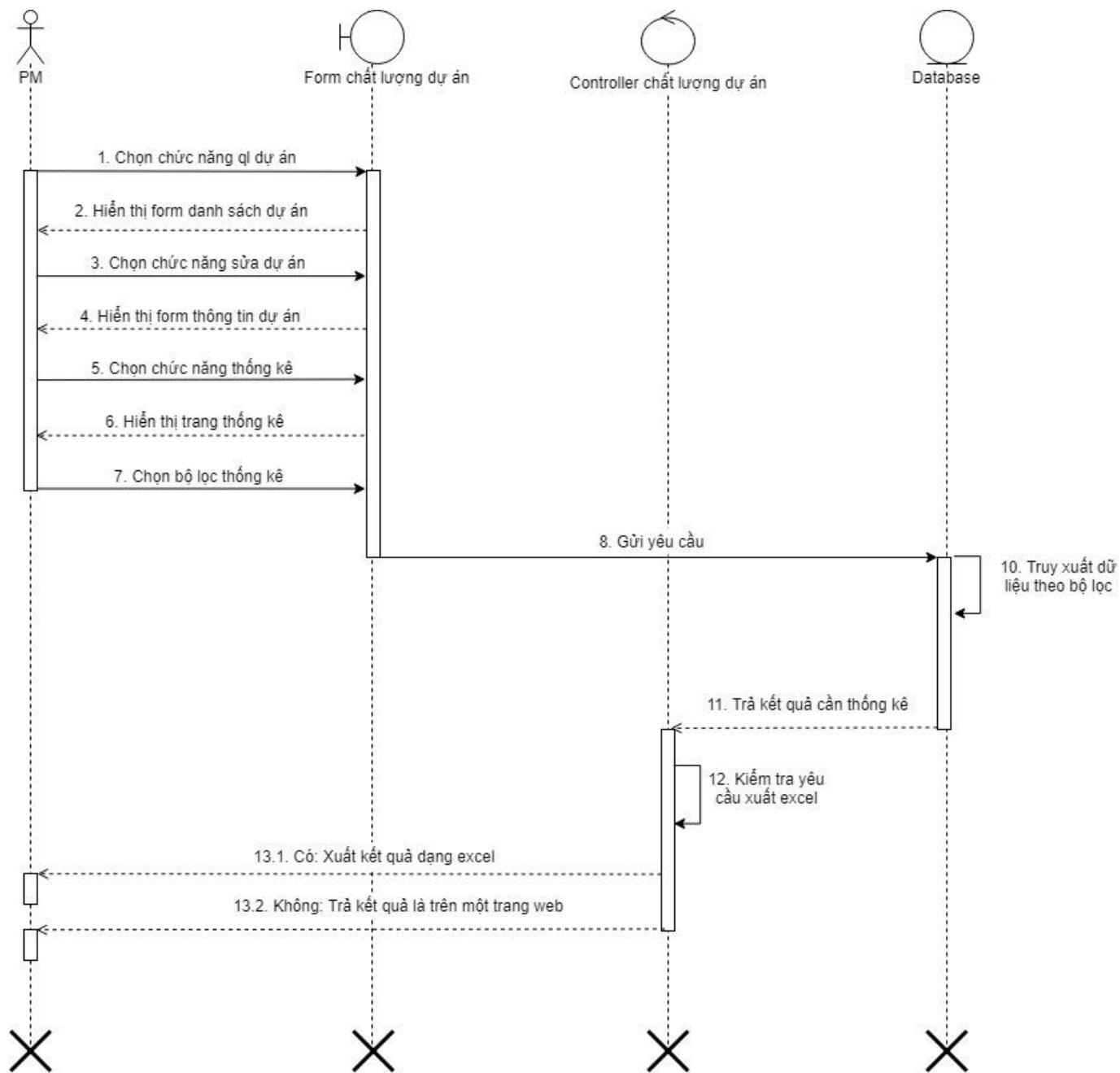
Hình 3.4.3. t. Biểu đồ trình tự xóa công việc.

3.4.3.12. Biểu đồ trình tự thực hiện công việc được giao



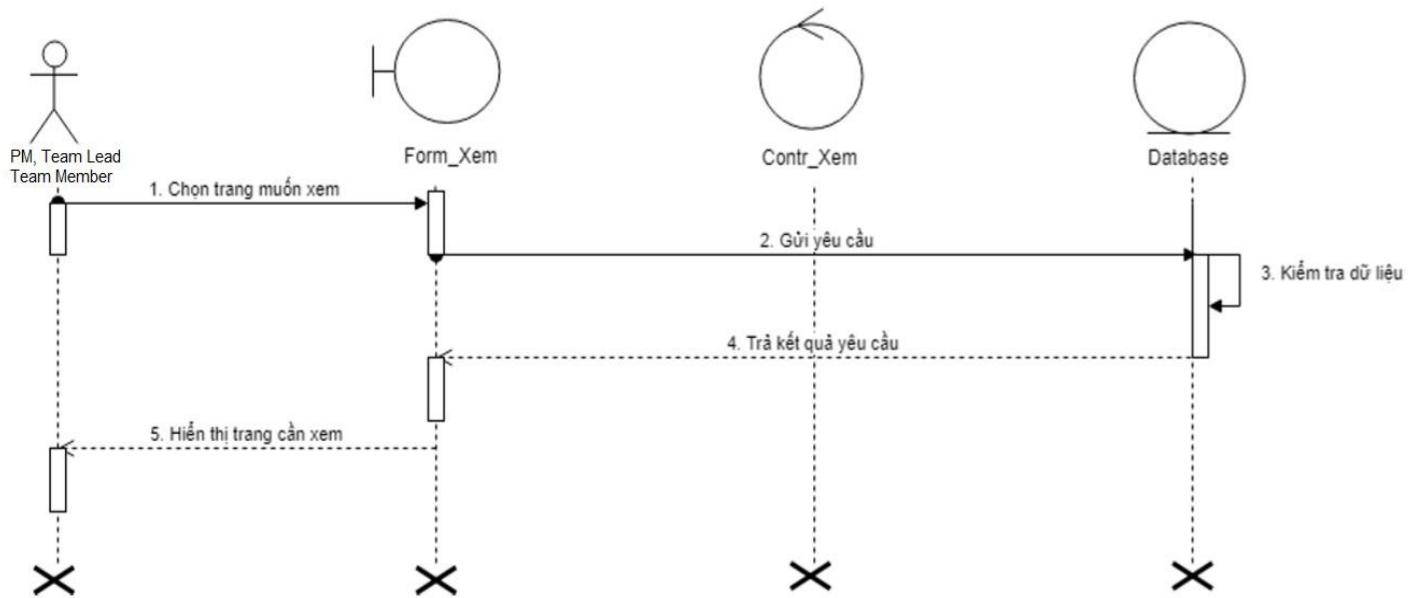
Hình 3.4.3. u. Biểu đồ trình tự thực hiện công việc.

3.4.3.13. Thống kê tiến độ dự án cá nhân



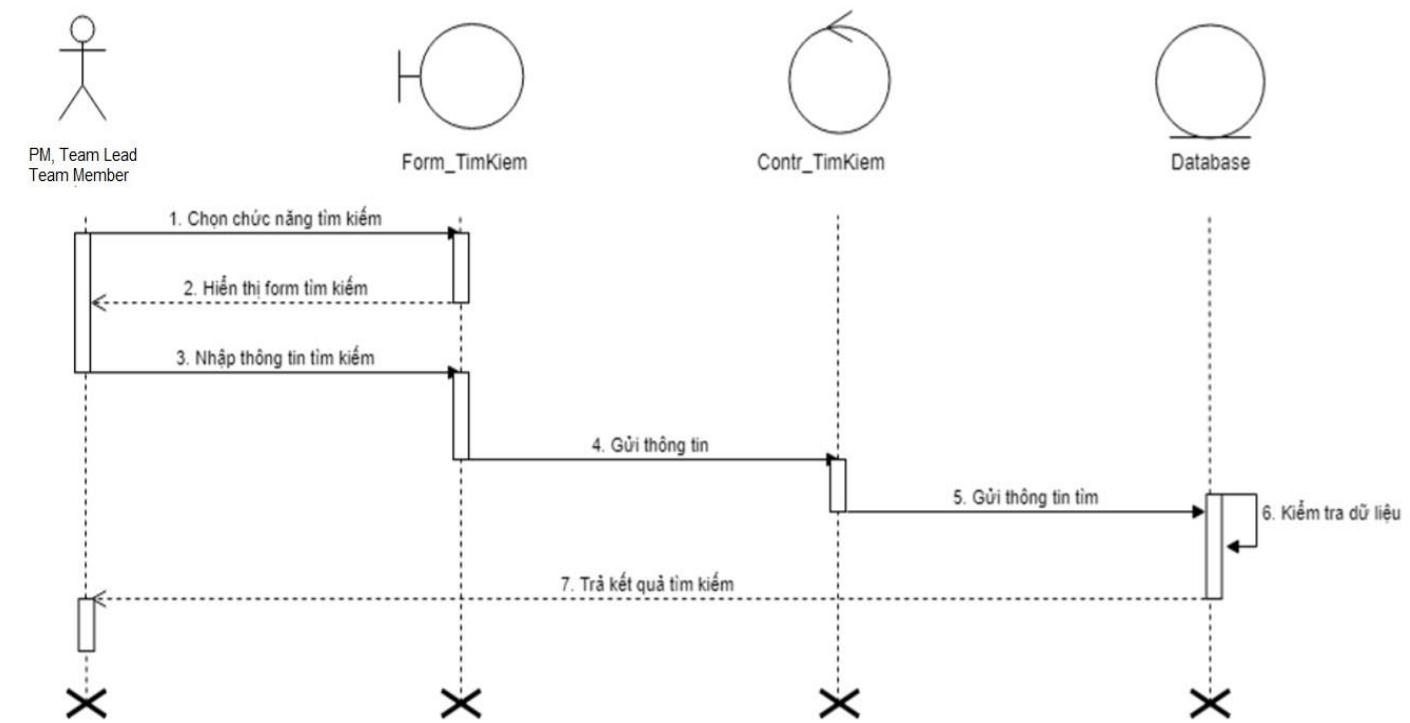
Hình 3.4.3. v. Biểu đồ trình tự thống kê tiến độ dự án cá nhân.

3.4.3.14. Xem chi tiết



Hình 3.4.3. w. Biểu đồ trình tự chức năng xem chi tiết.

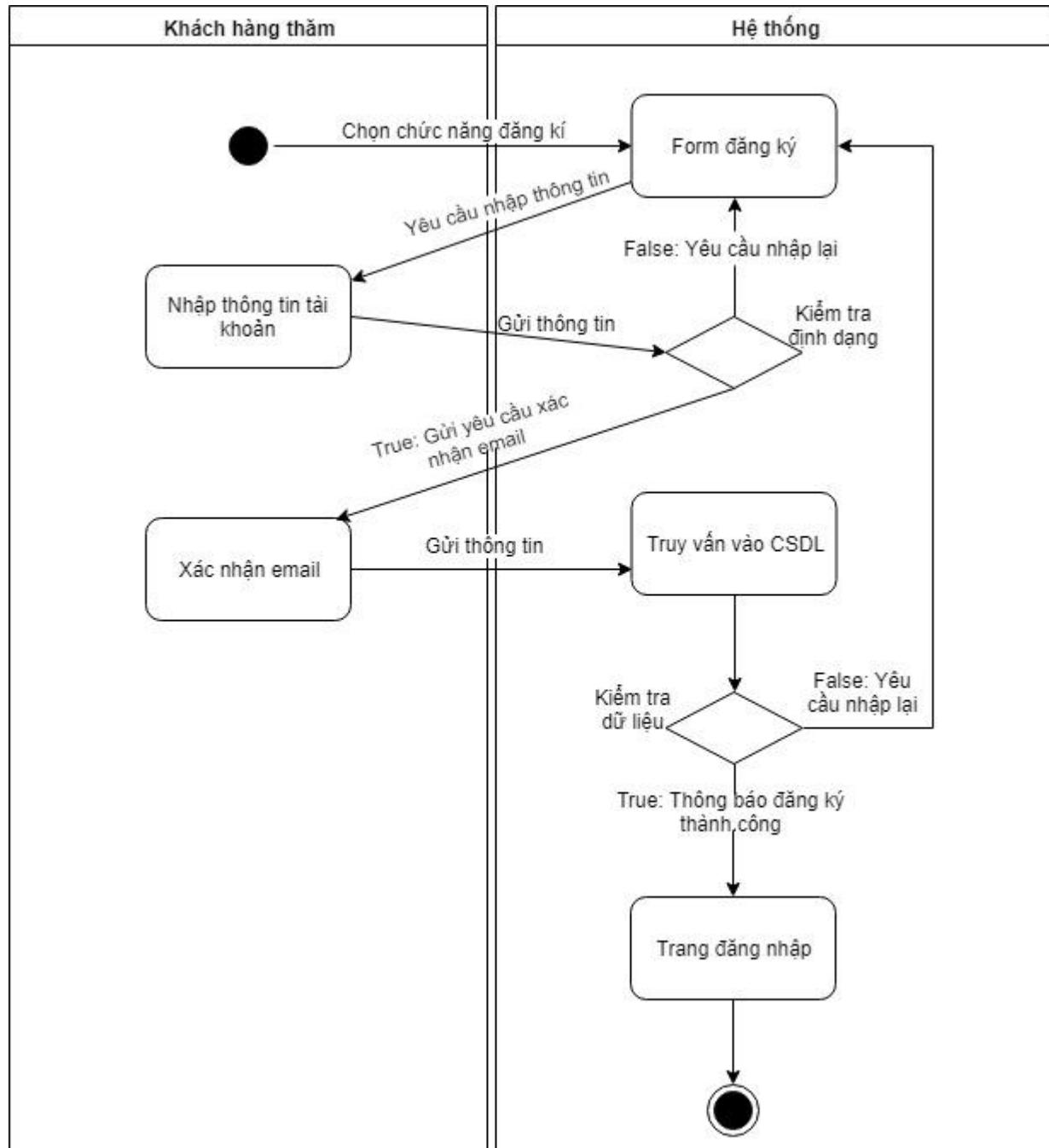
3.4.3.15. Tìm kiếm



Hình 3.4.3. x. Biểu đồ trình tự chức năng tìm kiếm.

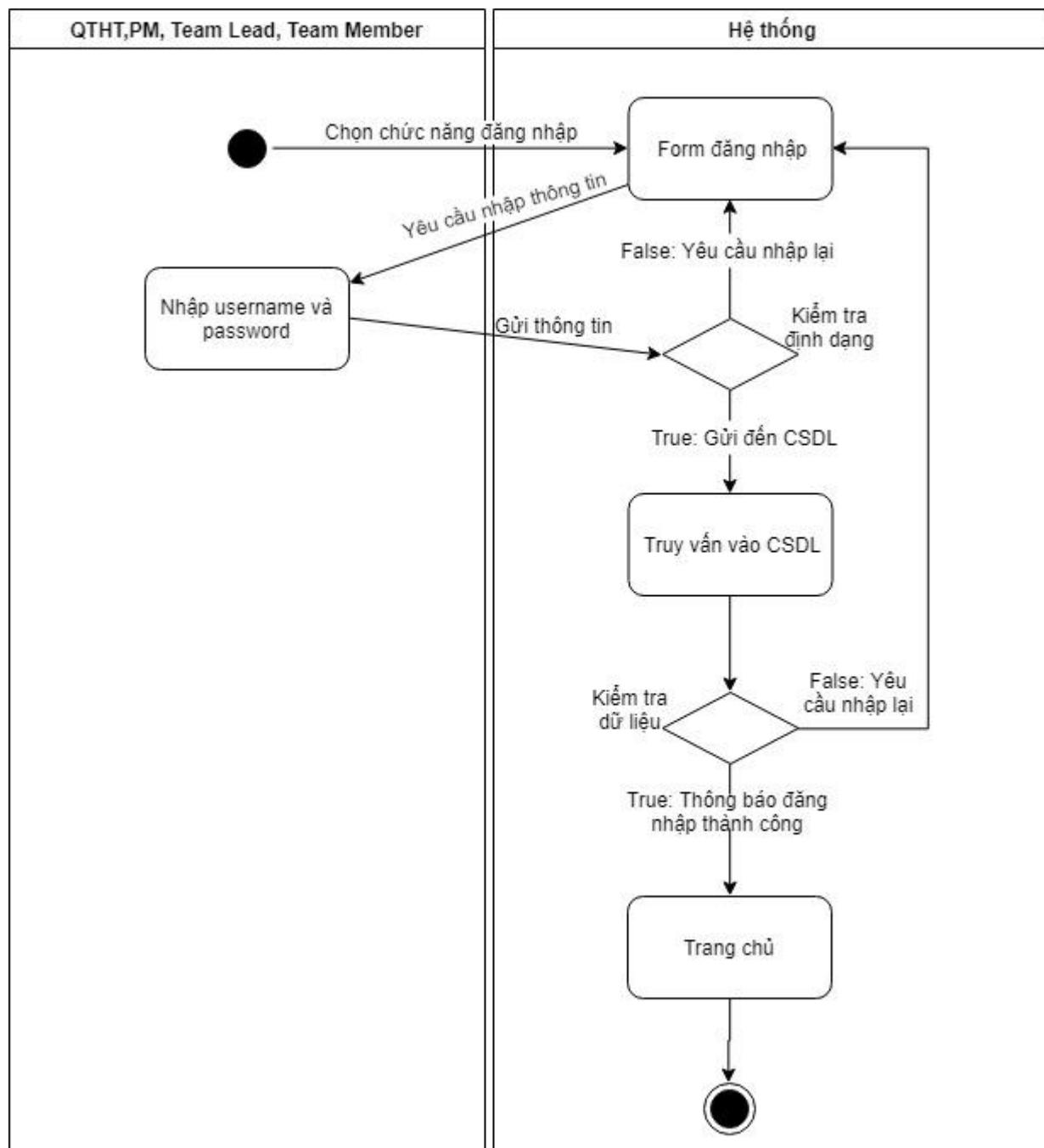
3.4.4. Biểu đồ hoạt động

3.4.4.1. Biểu đồ hoạt động đăng ký



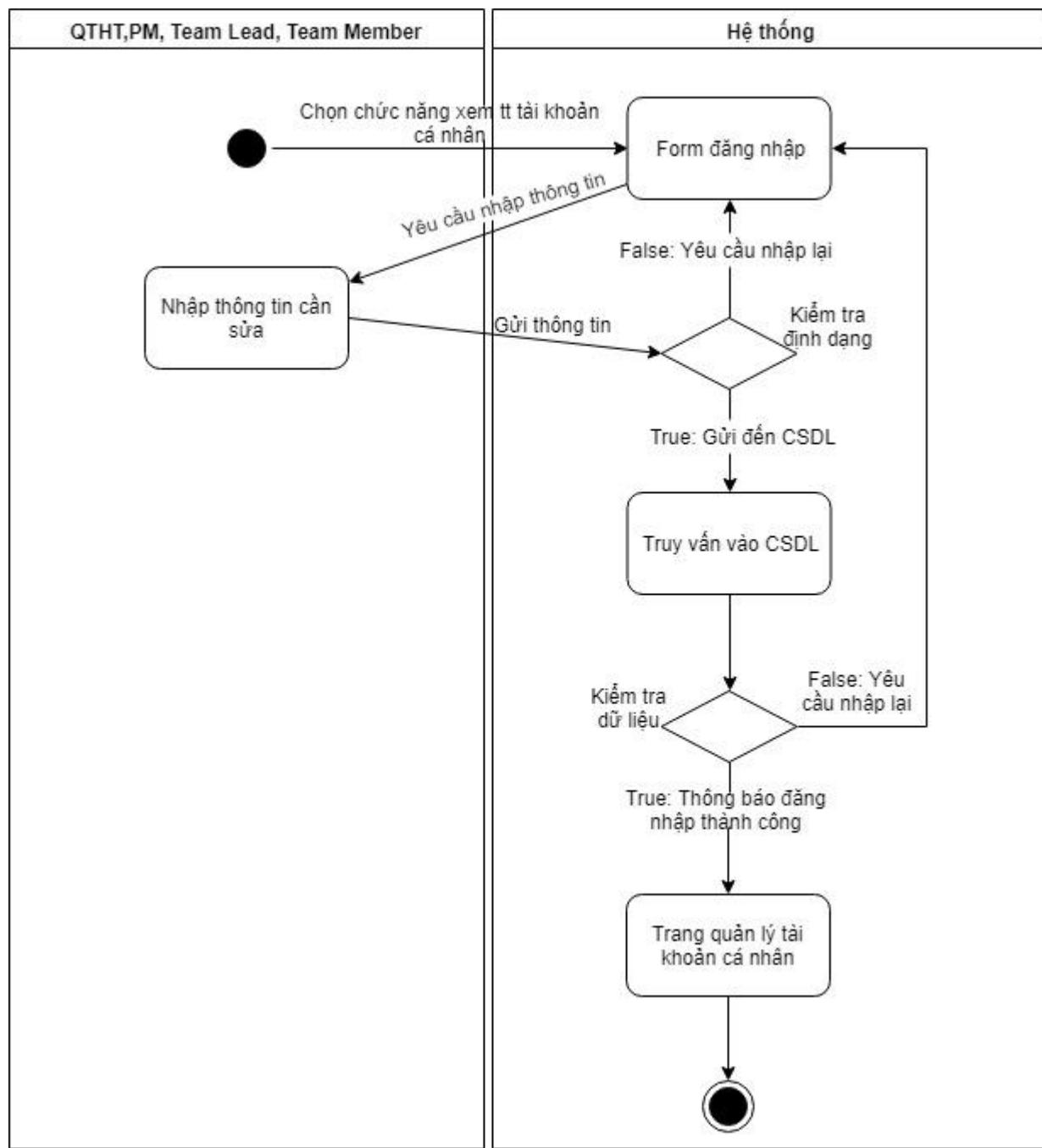
Hình 3.4.4. a. Biểu đồ hoạt động đăng ký.

3.4.4.2. Biểu đồ hoạt động đăng nhập



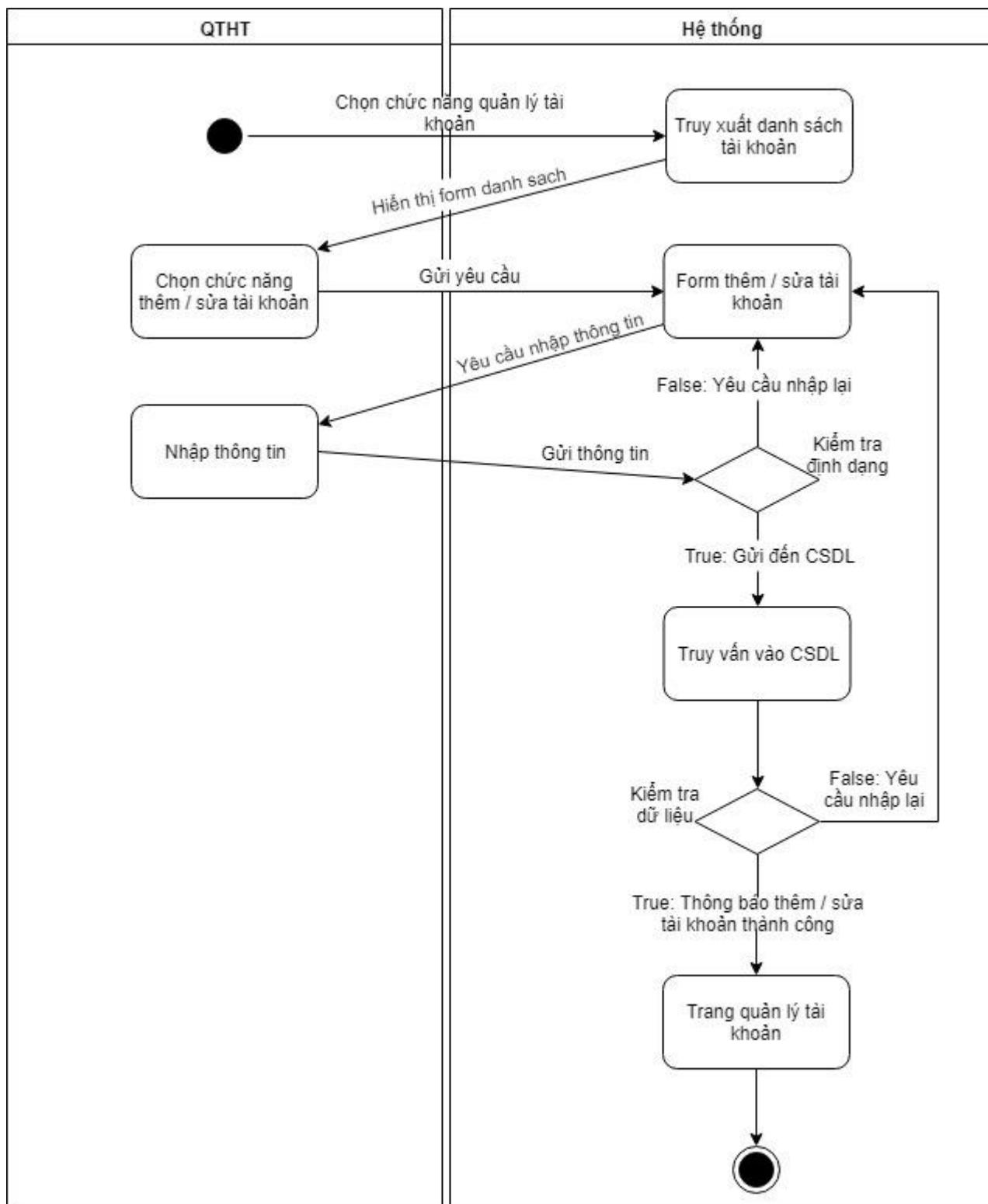
Hình 3.4.4. b. Biểu đồ hoạt động đăng nhập.

3.4.4.3. Biểu đồ hoạt động quản lý tài khoản cá nhân

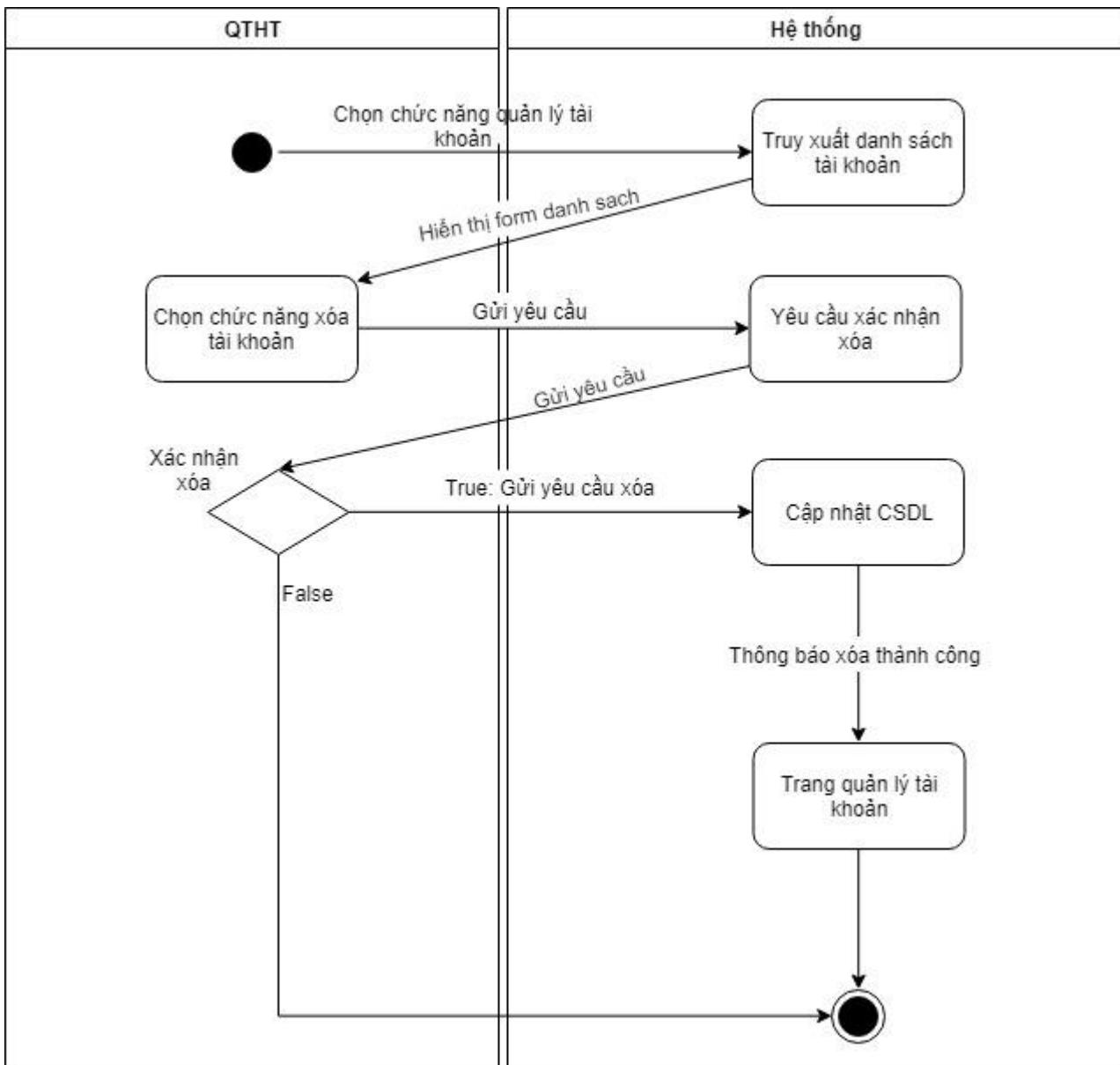


Hình 3.4.4. c. Biểu đồ hoạt động quản lý tài khoản cá nhân.

3.4.4.4. Biểu đồ hoạt động quản lý tài khoản trong hệ thống

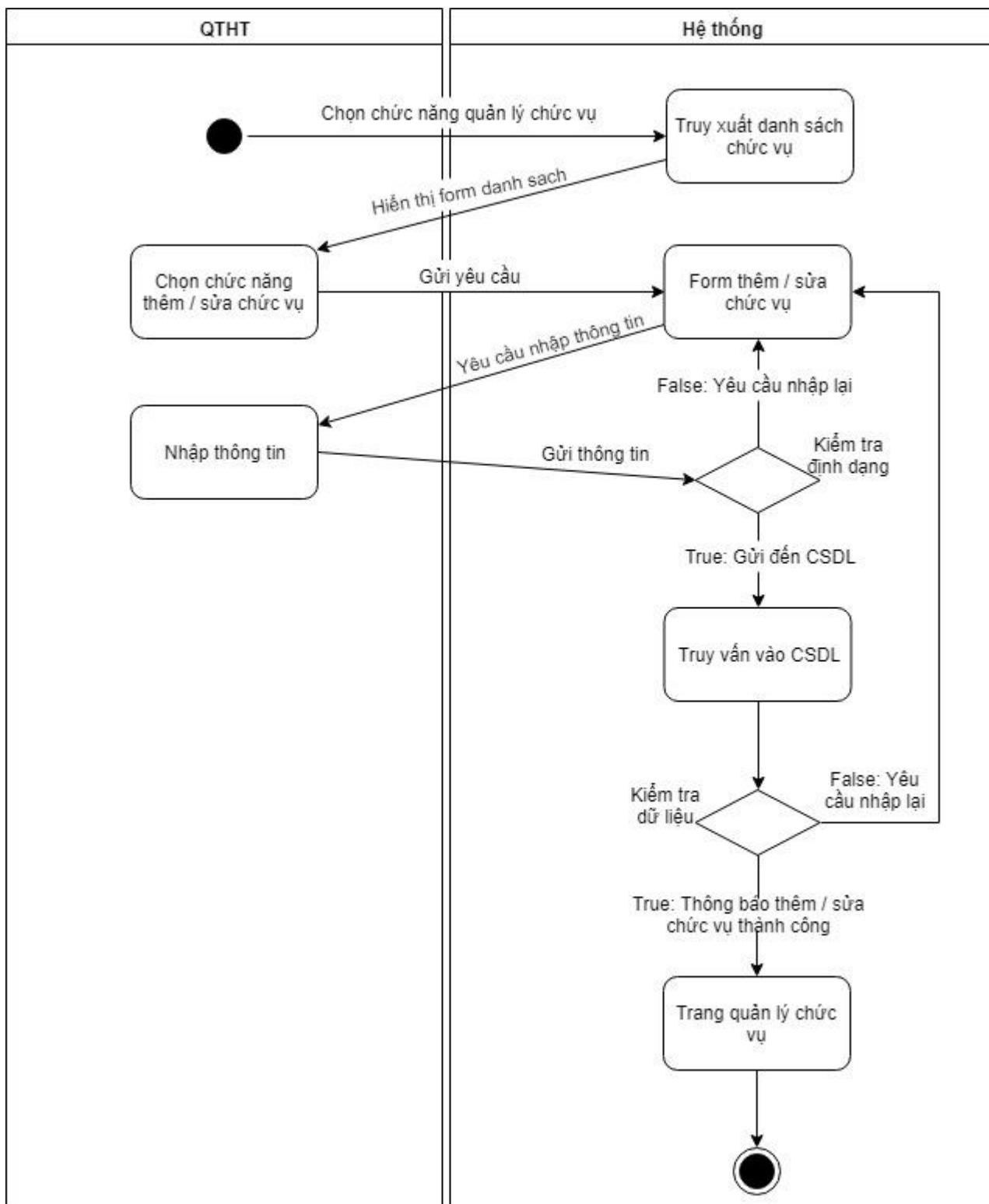


Hình 3.4.4. d. Biểu đồ hoạt động thêm/sửa tài khoản trong hệ thống.

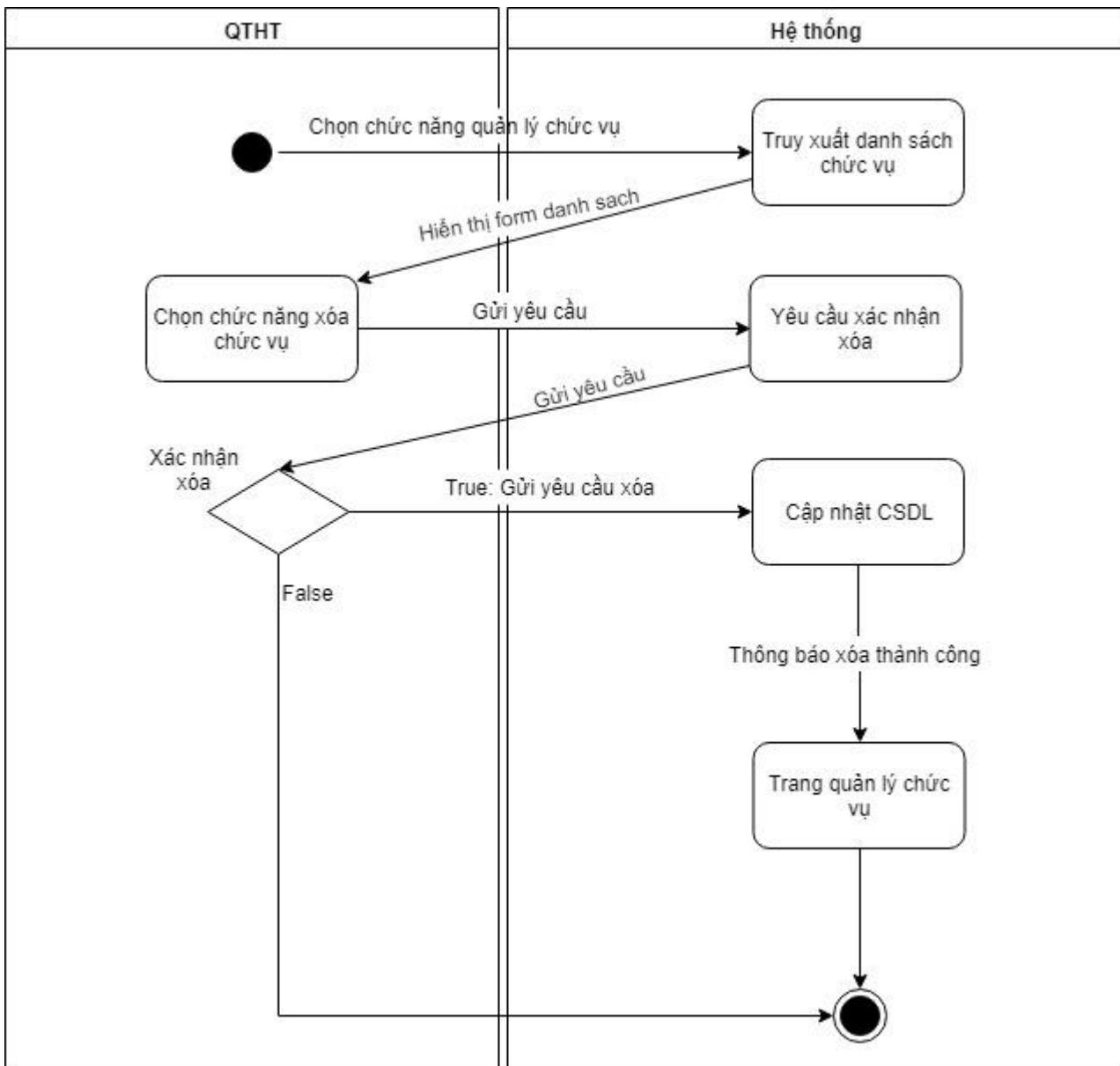


Hình 3.4.4. e. Biểu đồ hoạt động xóa tài khoản trong hệ thống.

3.4.4.5. Biểu đồ hoạt động quản lý chức vụ trong ngành CNTT

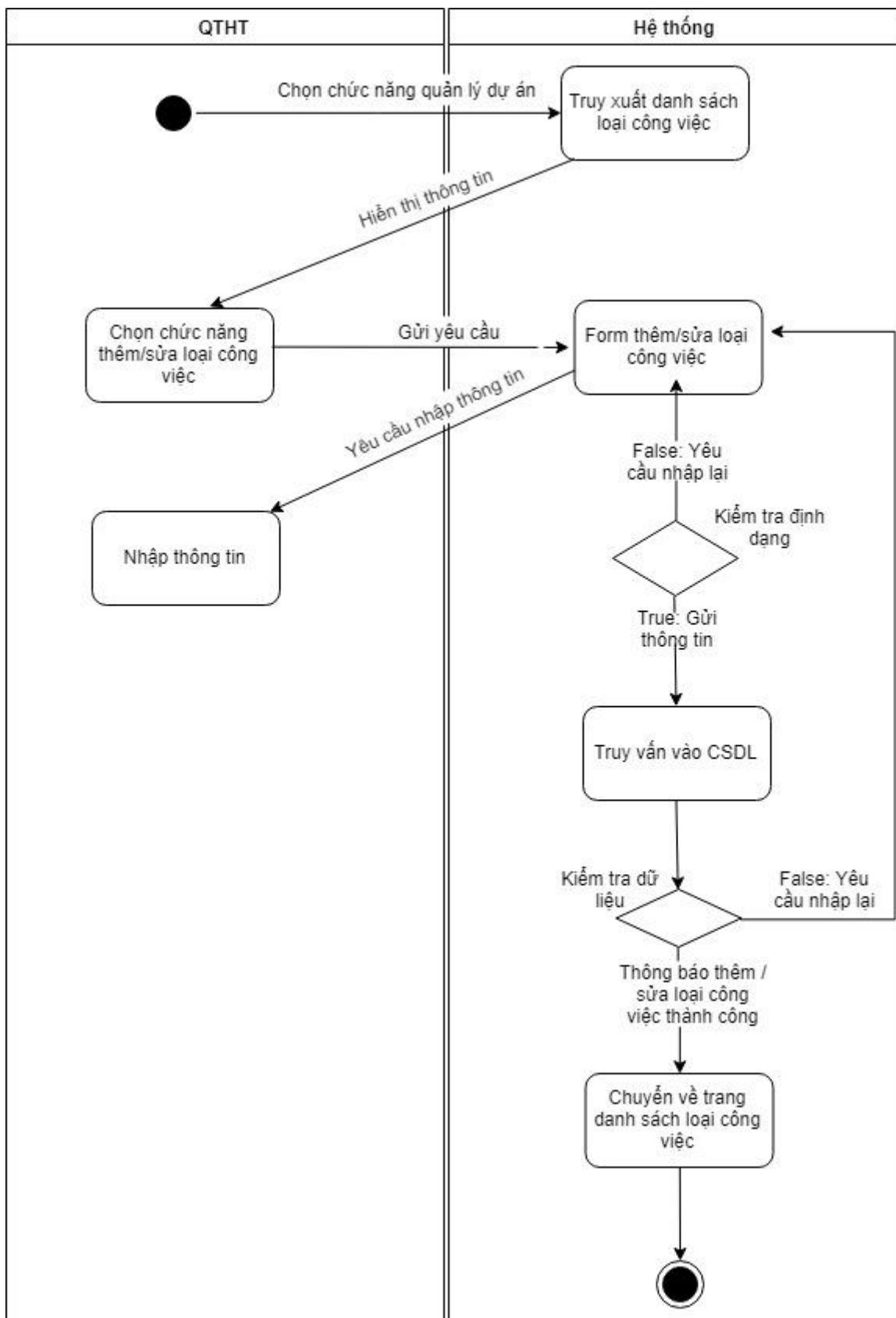


Hình 3.4.4. f. Biểu đồ hoạt động thêm/sửa chức vụ.

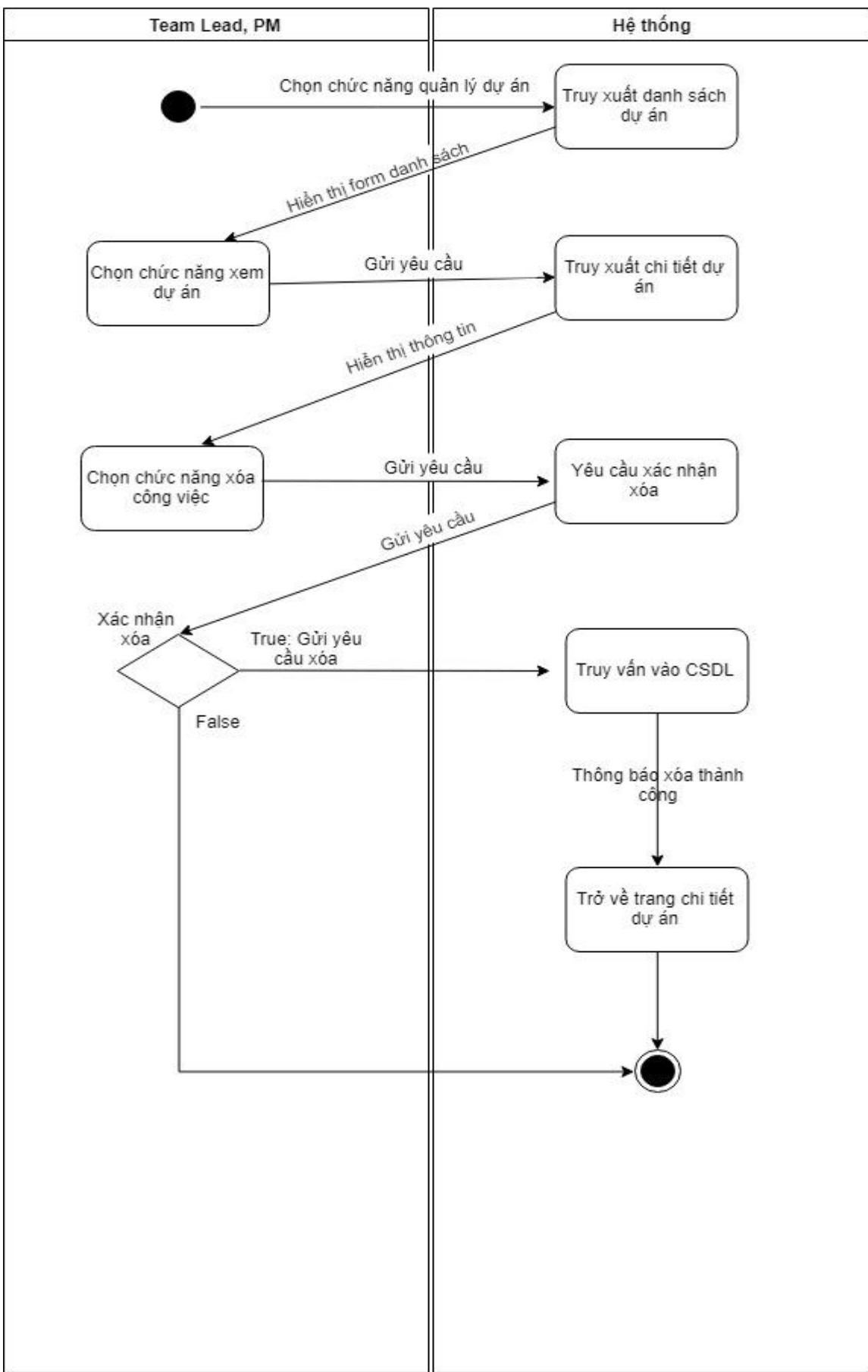


Hình 3.4.4. g. Biểu đồ hoạt động xóa chức vụ.

3.4.4.6. Biểu đồ hoạt động quản lý loại công việc

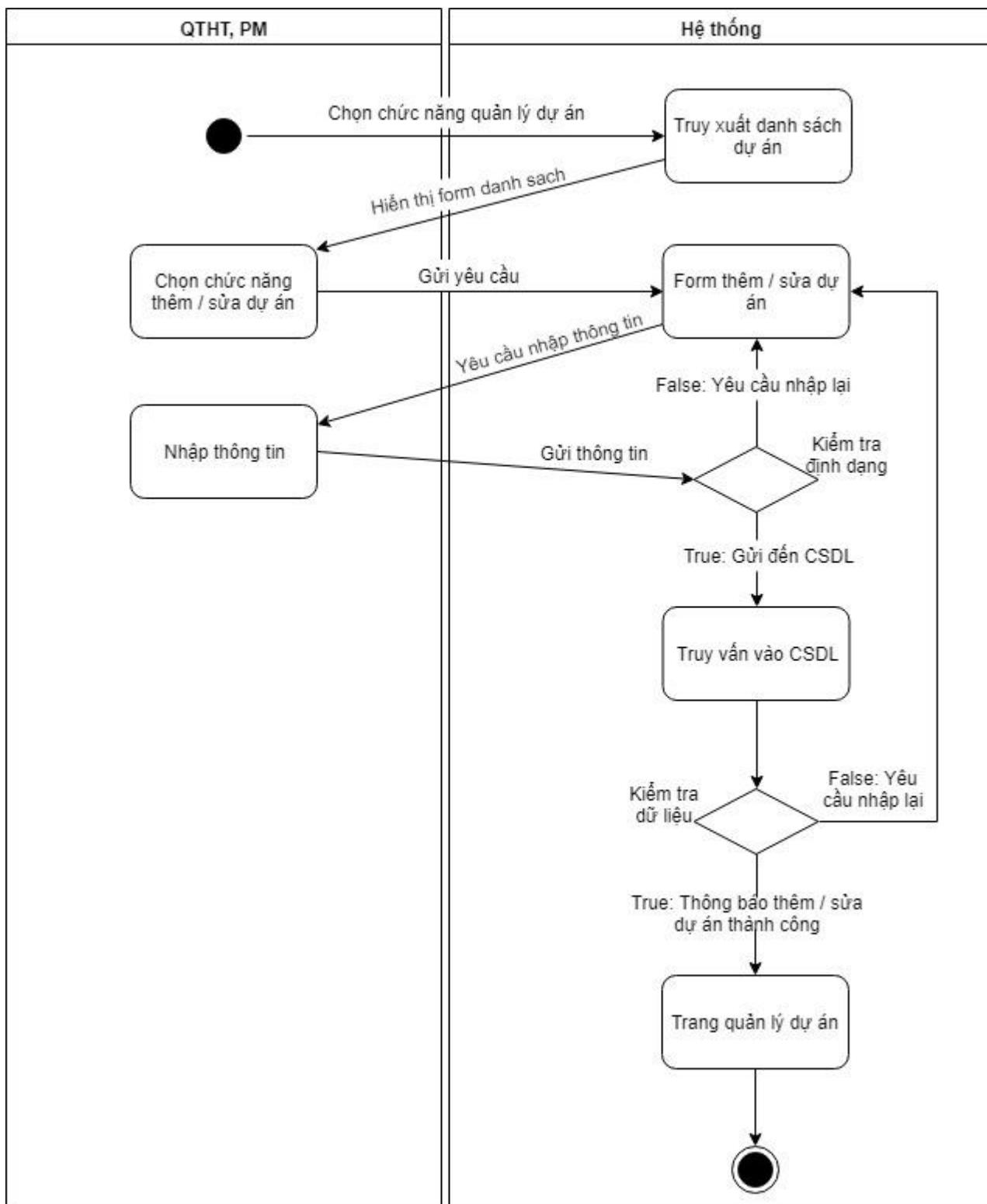


Hình 3.4.4. h. Biểu đồ hoạt động thêm/sửa loại công việc.

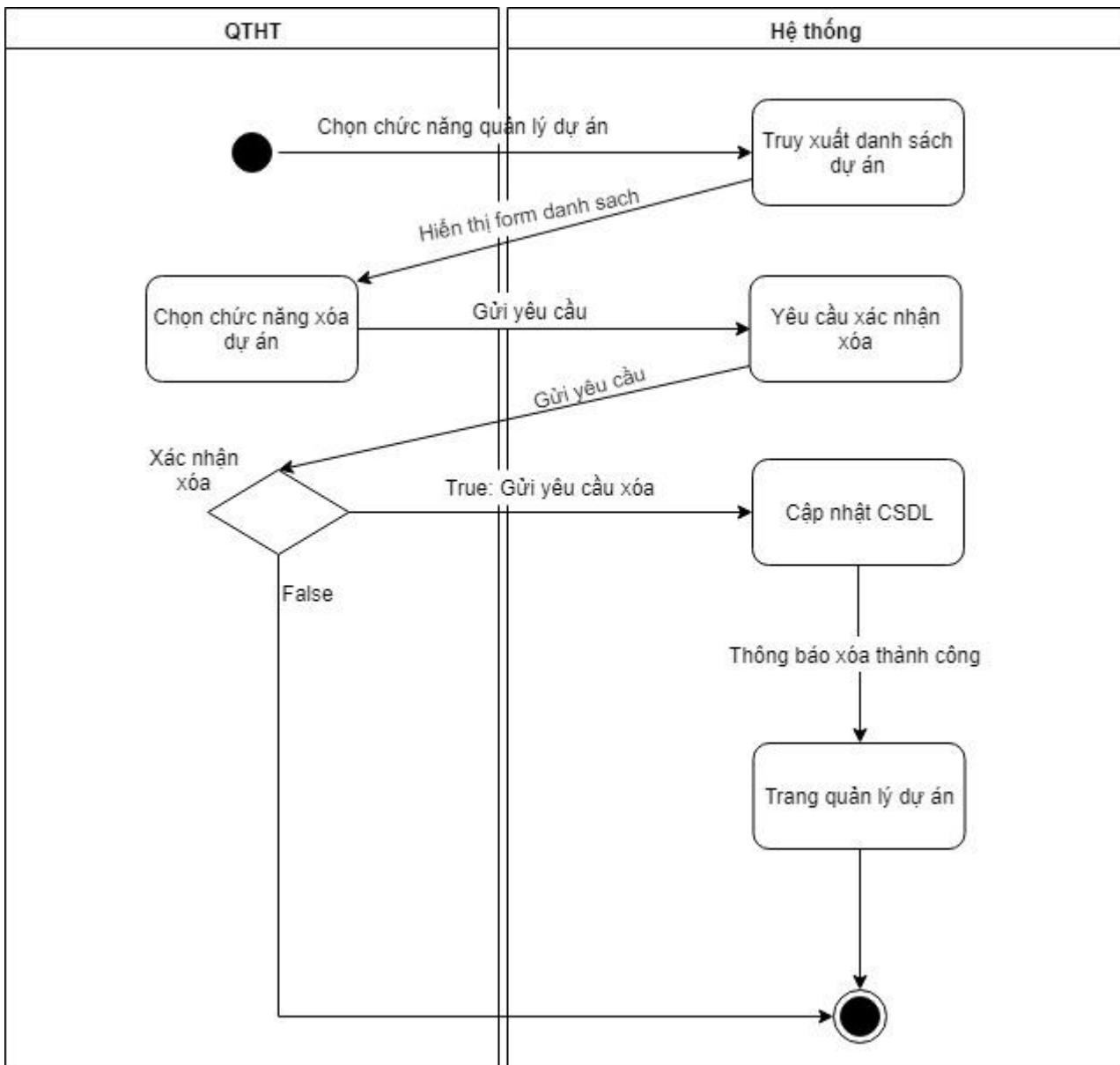


Hình 3.4.4. i. Biểu đồ hoạt động xóa loại công việc.

3.4.4.7. Biểu đồ hoạt động quản lý dự án

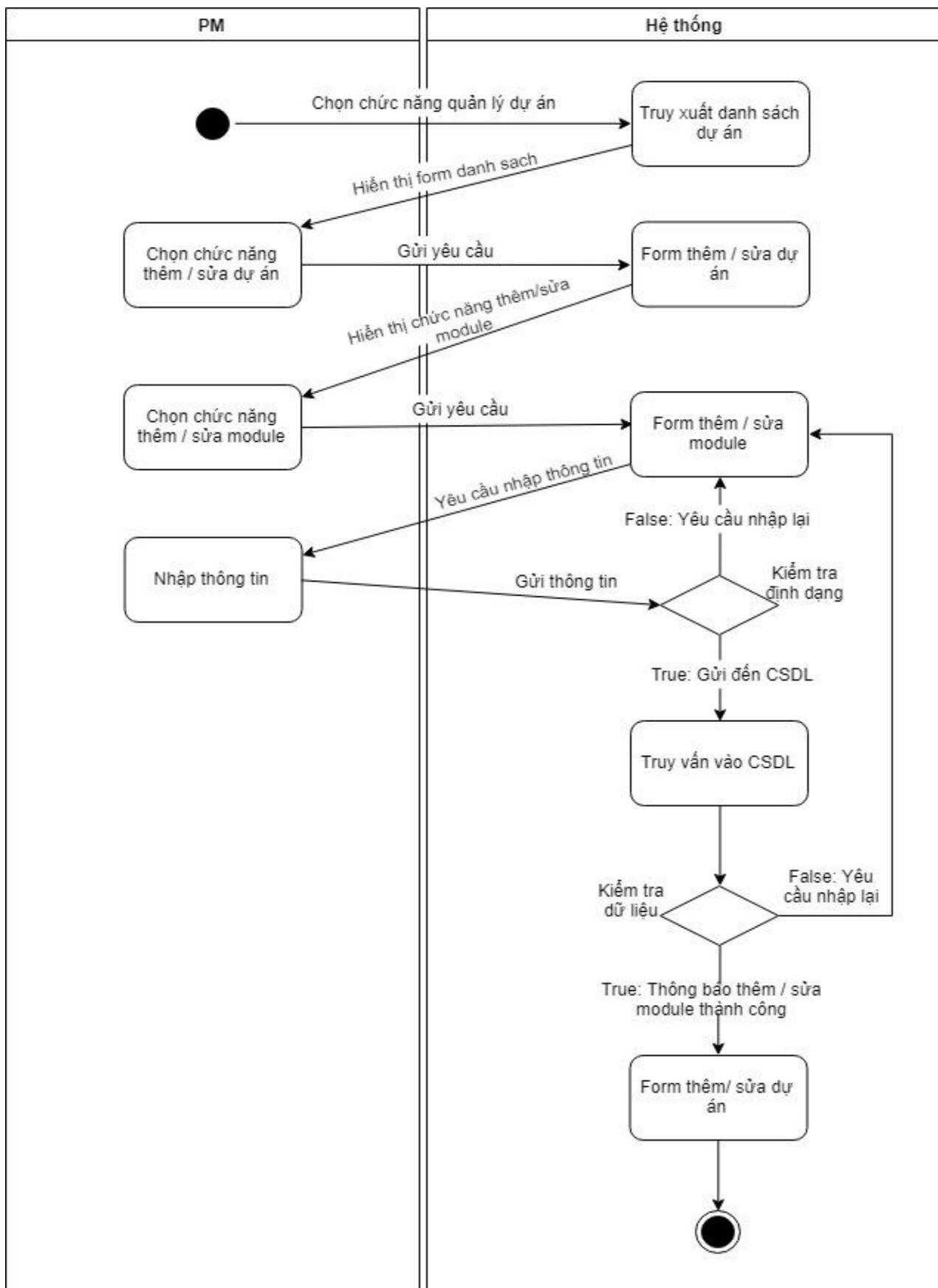


Hình 3.4.4. j. Biểu đồ hoạt động thêm/sửa dự án trong hệ thống.

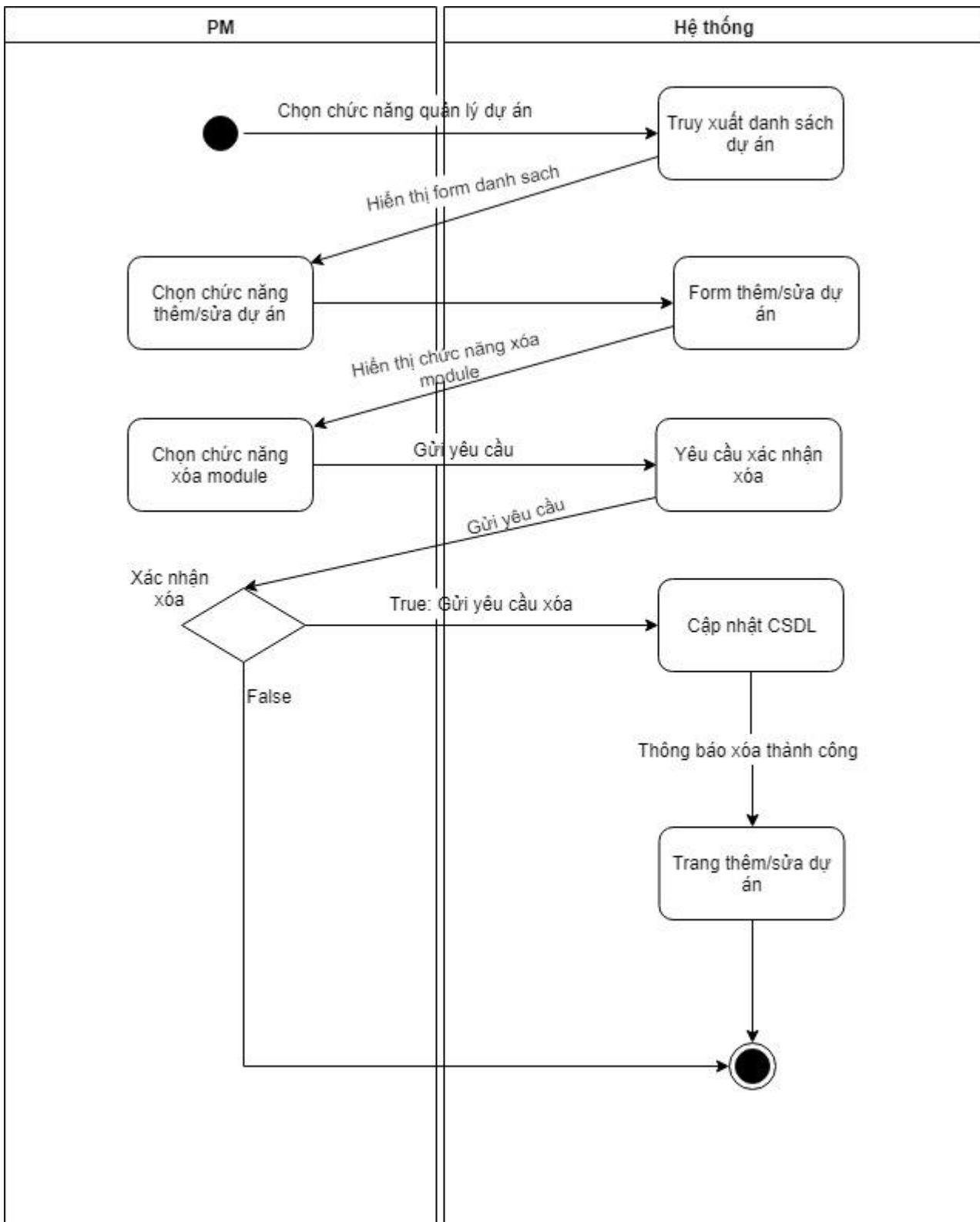


Hình 3.4.4. k. Biểu đồ hoạt động xóa dự án trong hệ thống.

3.4.4.8. Biểu đồ hoạt động quản lý module trong dự án cá nhân

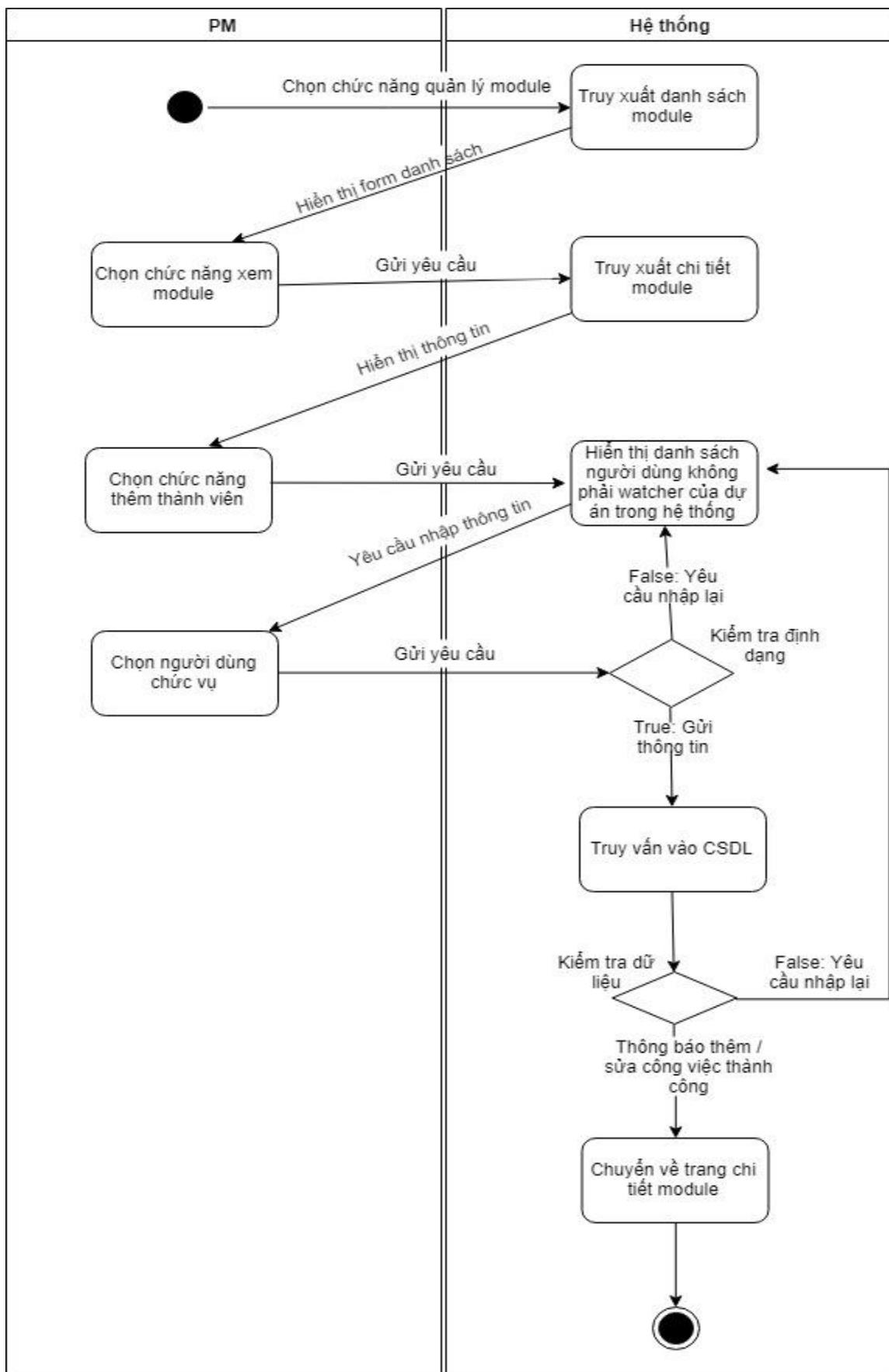


Hình 3.4.4. l. Biểu đồ hoạt động thêm/sửa module.

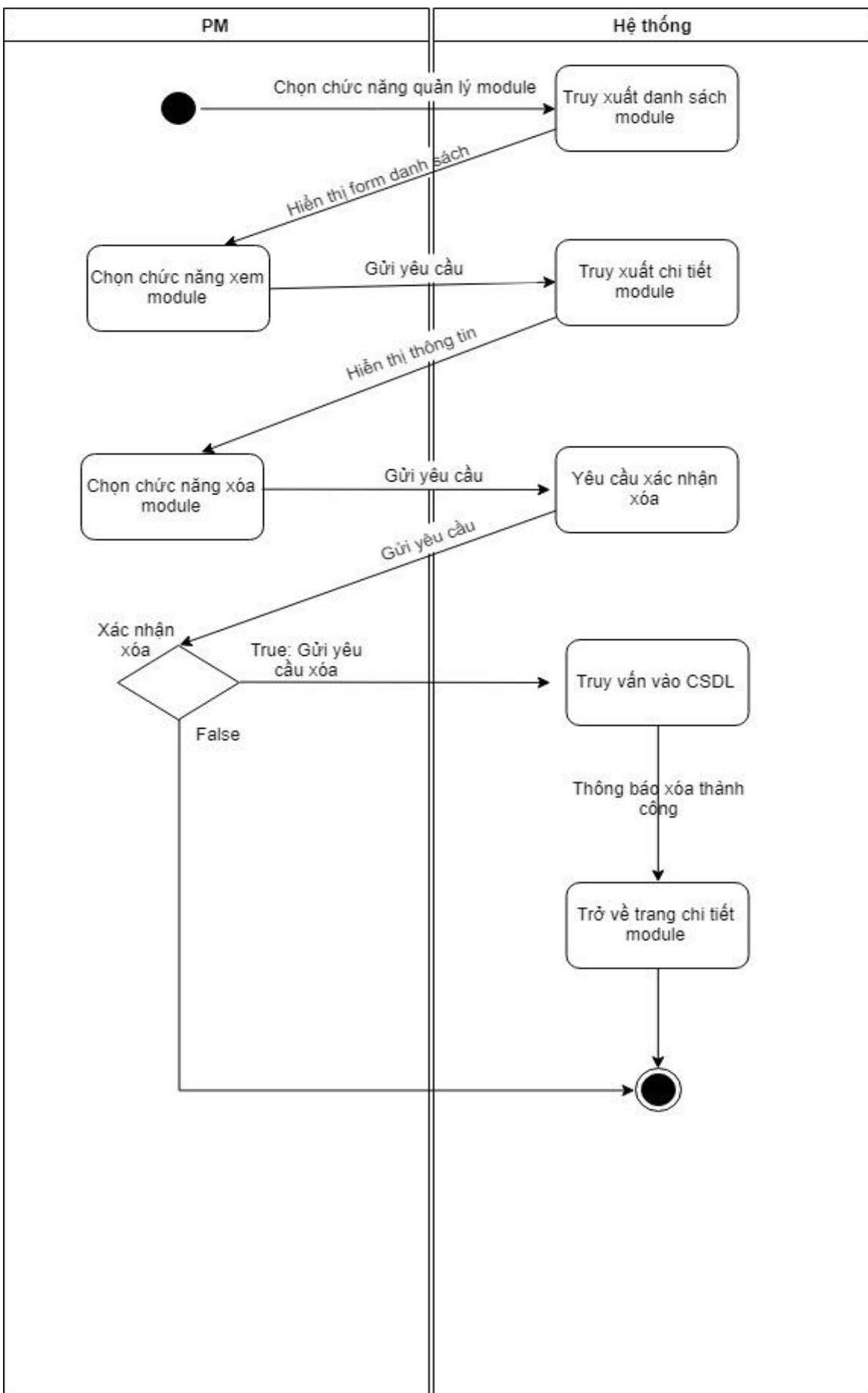


Hình 3.4.4. m. Biểu đồ hoạt động xóa module.

3.4.4.9. Biểu đồ hoạt động quản lý thành viên trong module

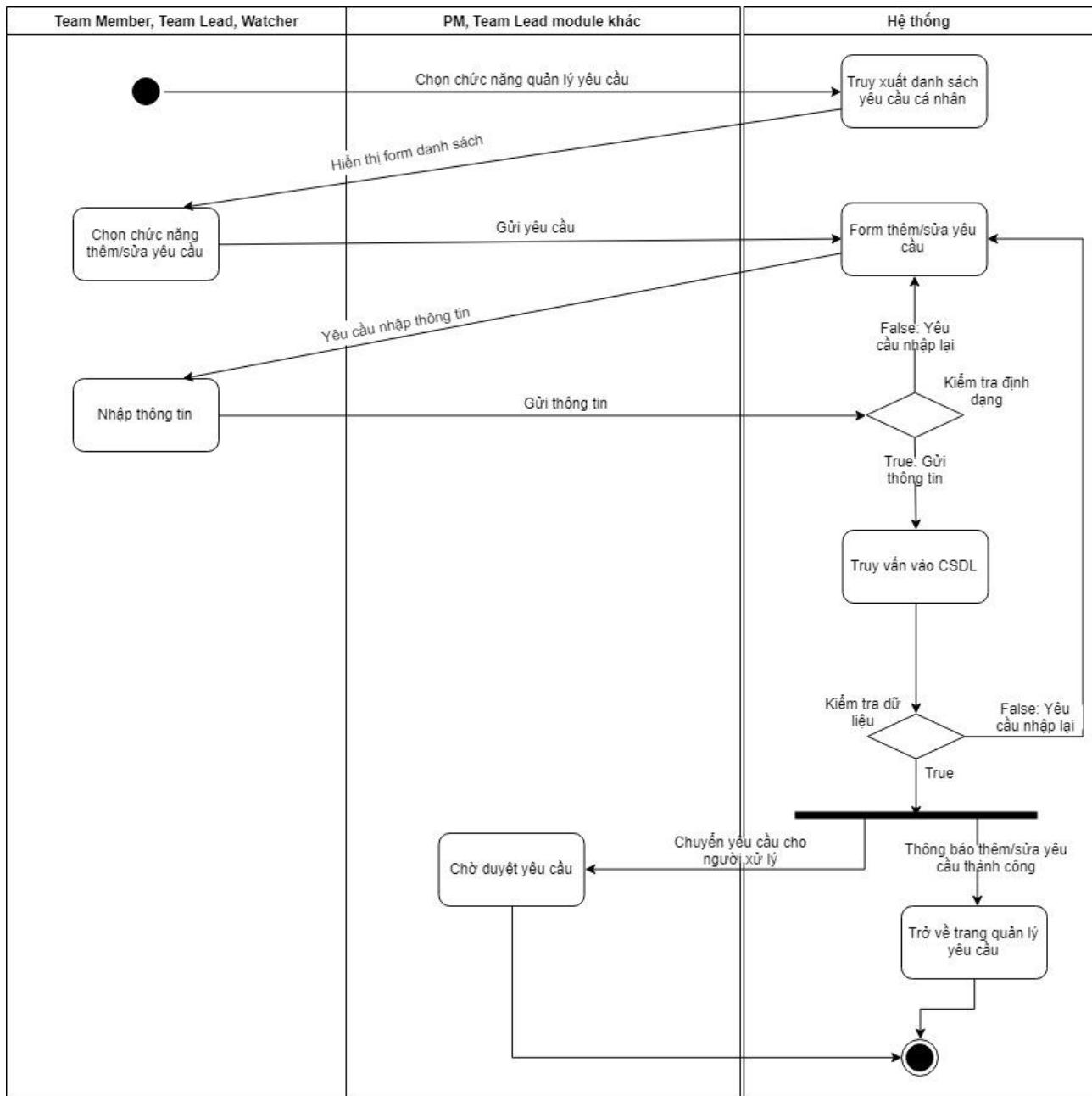


Hình 3.4.4. n. Biểu đồ hoạt động thêm thành viên vào dự án.

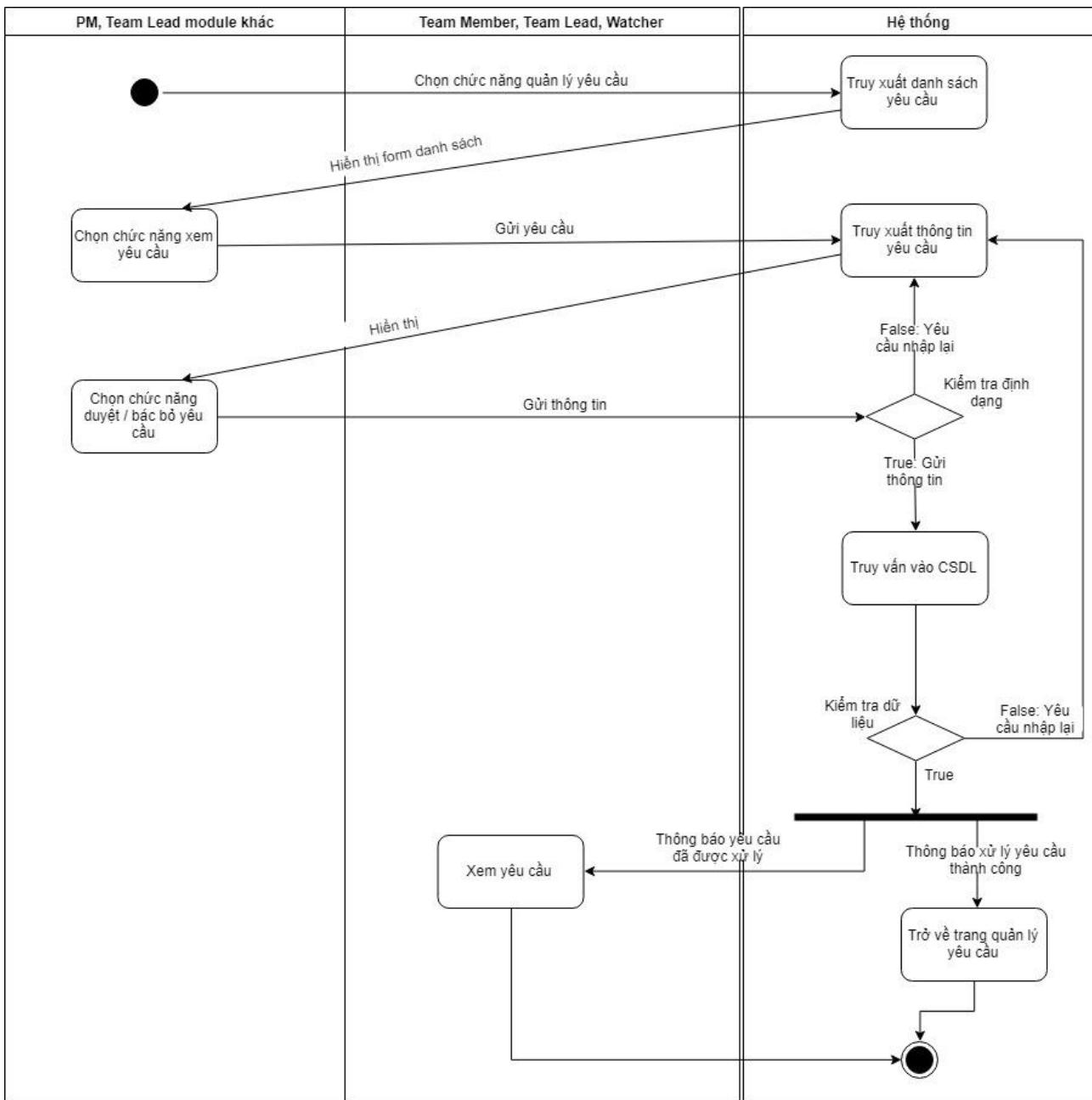


Hình 3.4.4. o. Biểu đồ hoạt động xóa thành viên khỏi module.

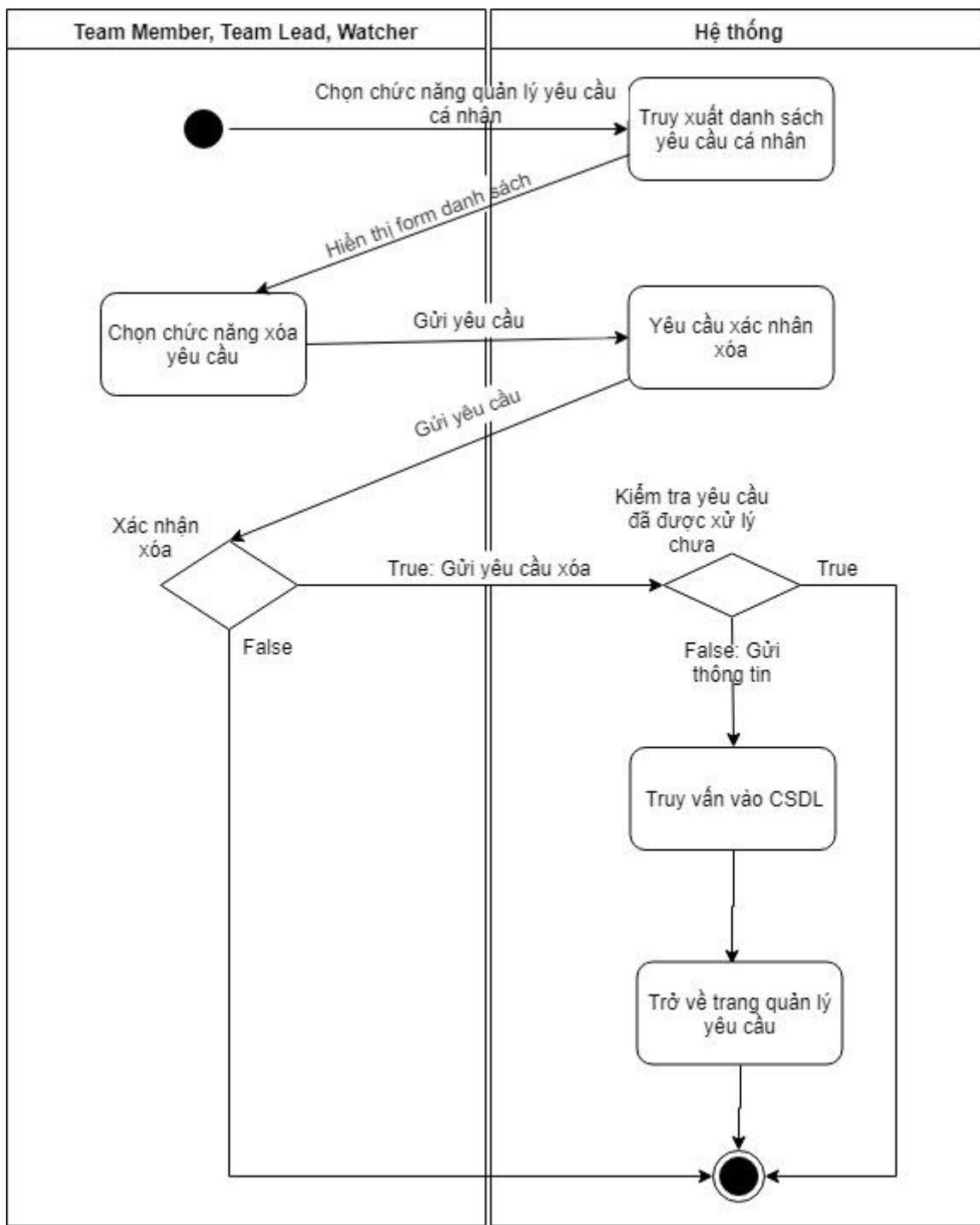
3.4.4.10. Biểu đồ hoạt động quản lý yêu cầu



Hình 3.4.4. p. Biểu đồ hoạt động thêm/sửa yêu cầu.

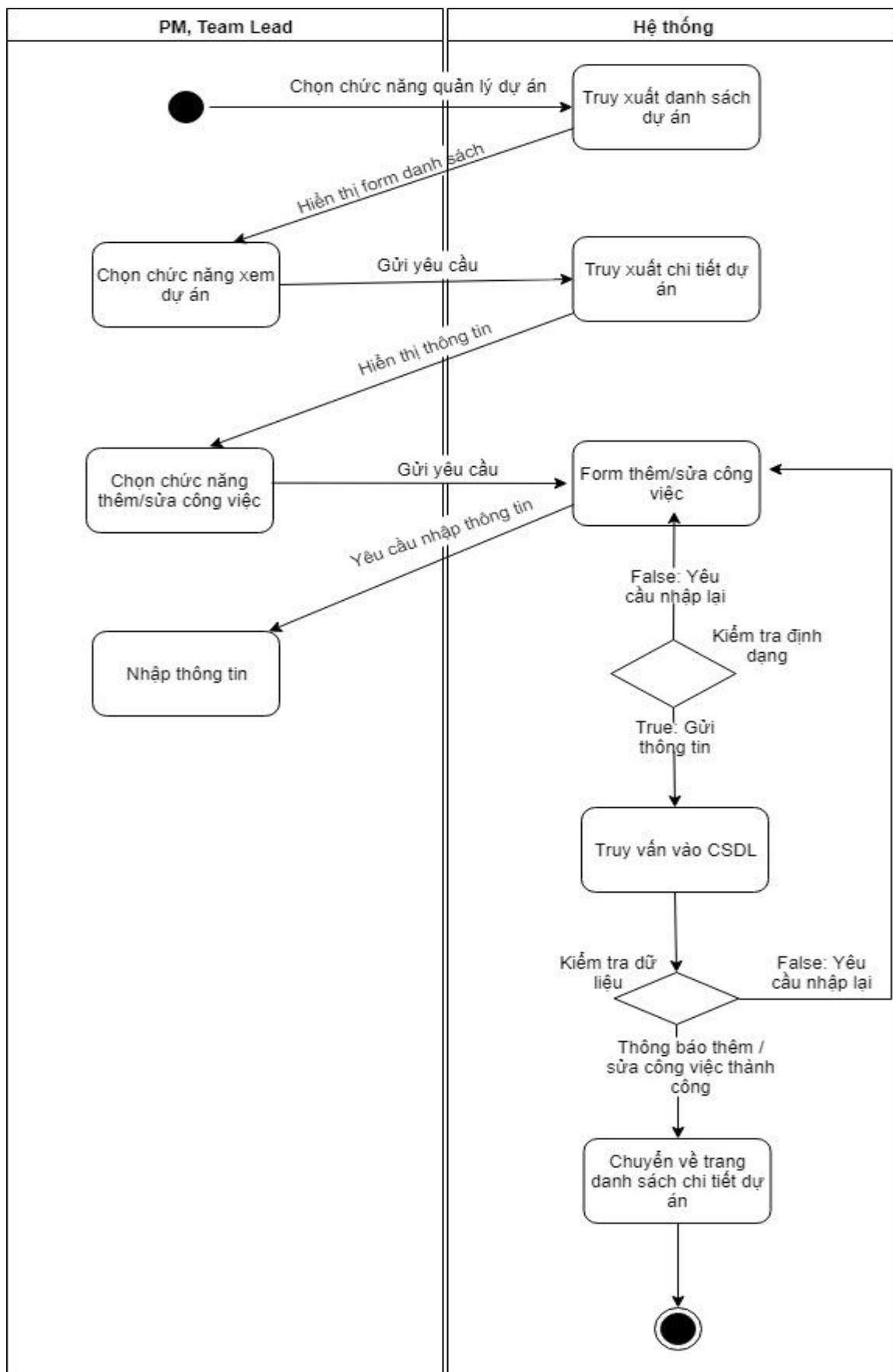


Hình 3.4.4. q. Biểu đồ hoạt động xử lý yêu cầu.

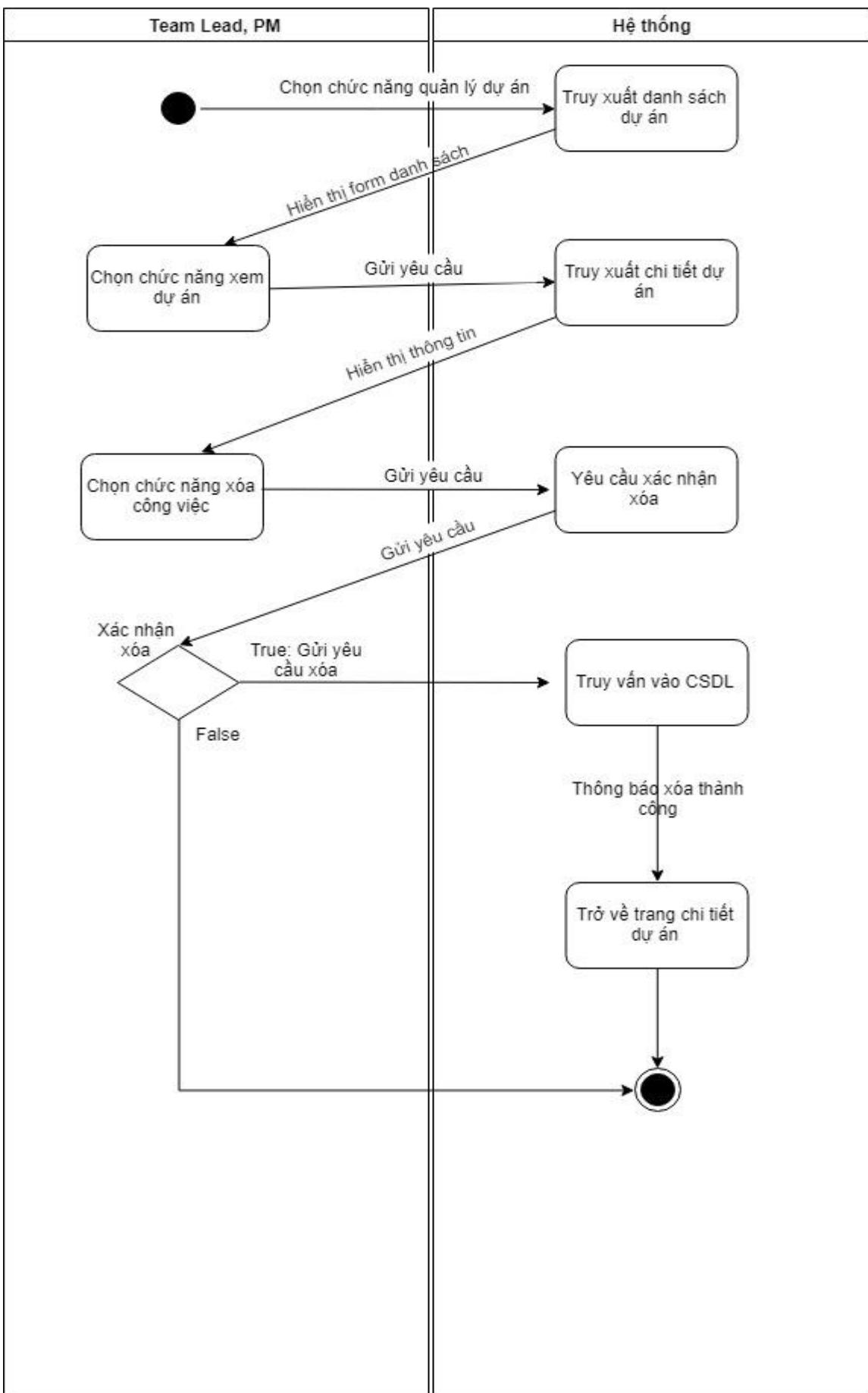


Hình 3.4.4. r. Biểu đồ hoạt động xóa yêu cầu.

3.4.4.11. Biểu đồ hoạt động quản lý công việc

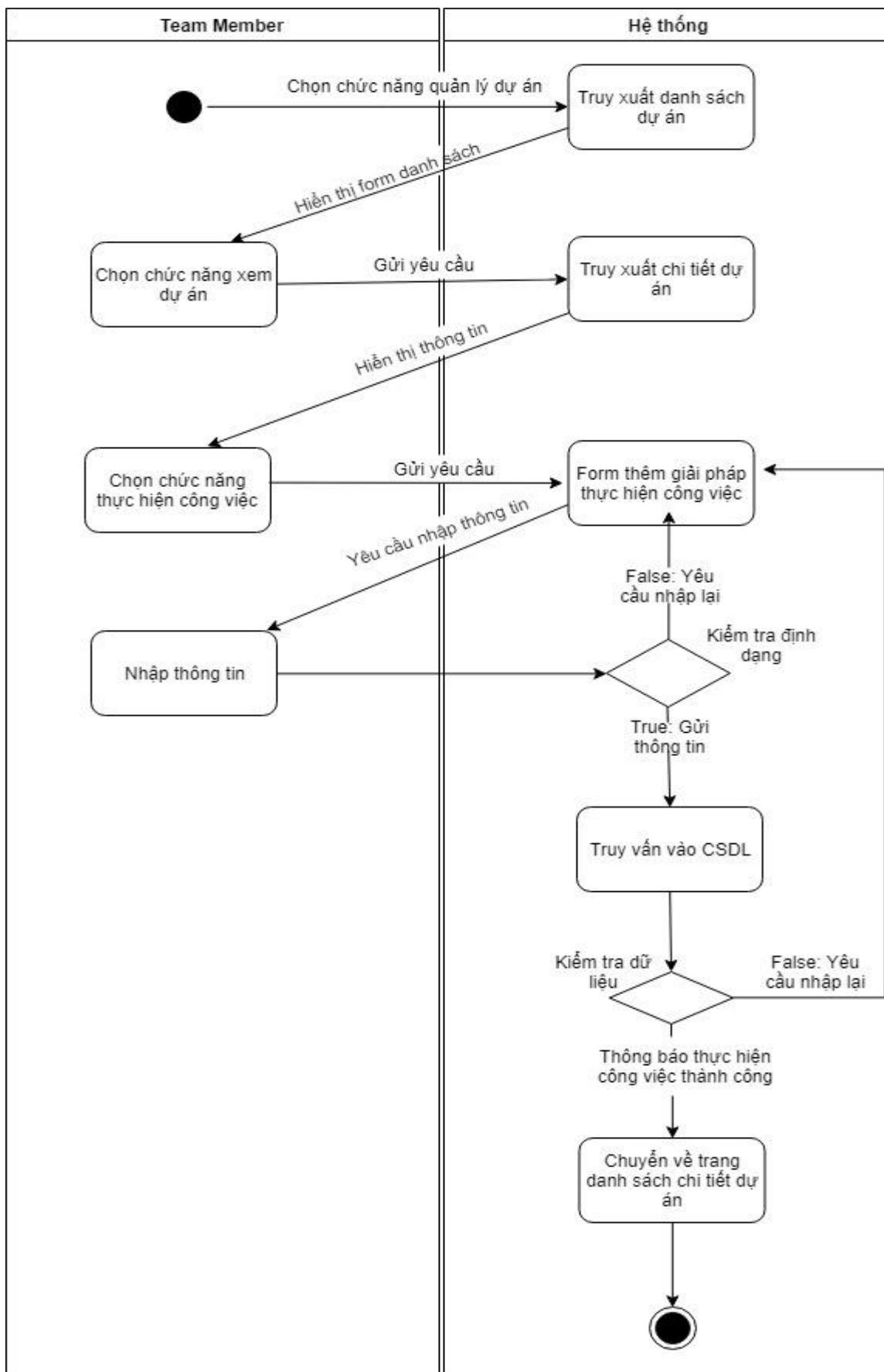


Hình 3.4.4. s. Biểu đồ hoạt động thêm/sửa công việc.



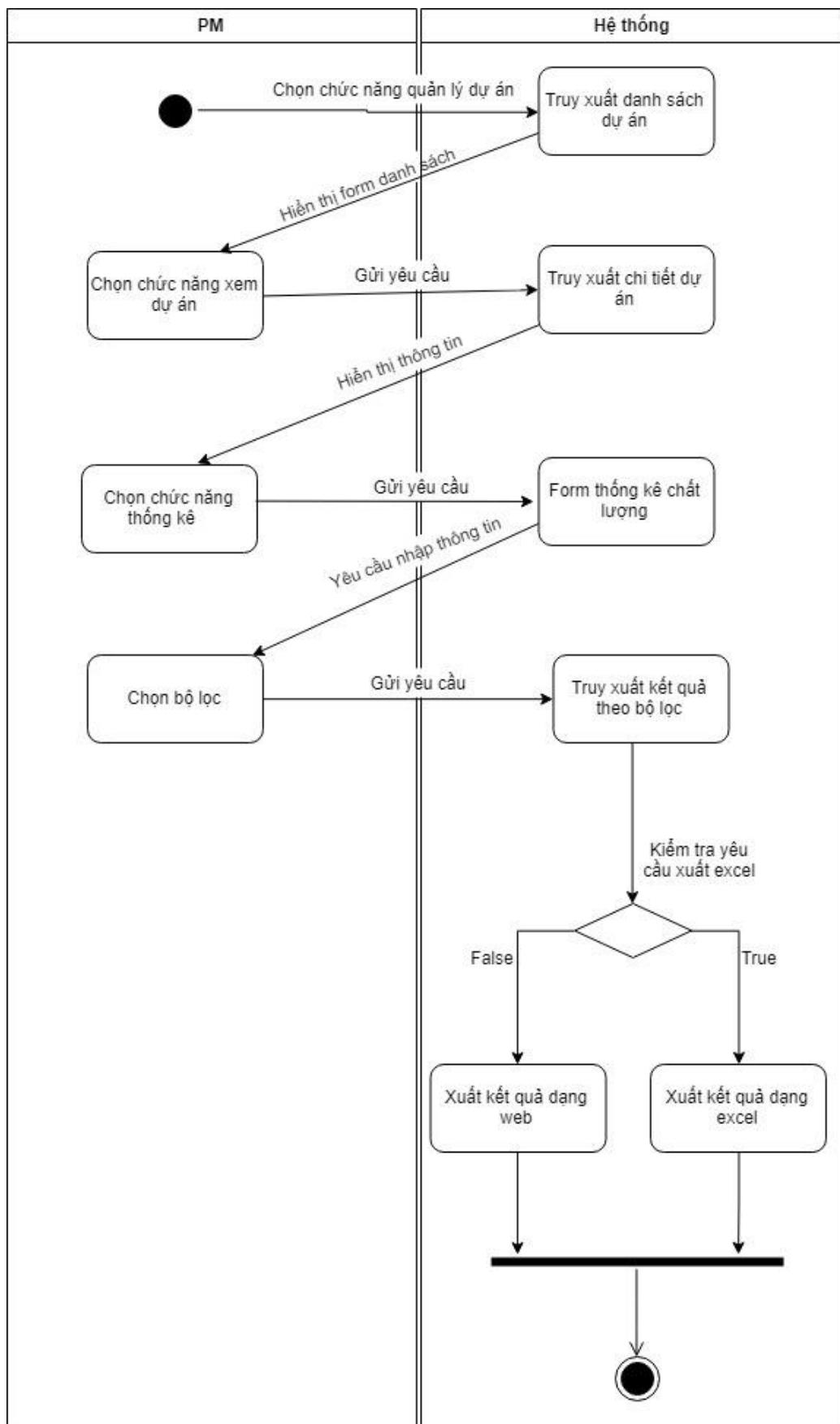
Hình 3.4.4. t. Biểu đồ hoạt động xóa công việc.

3.4.4.12. Biểu đồ hoạt động thực hiện công việc được giao



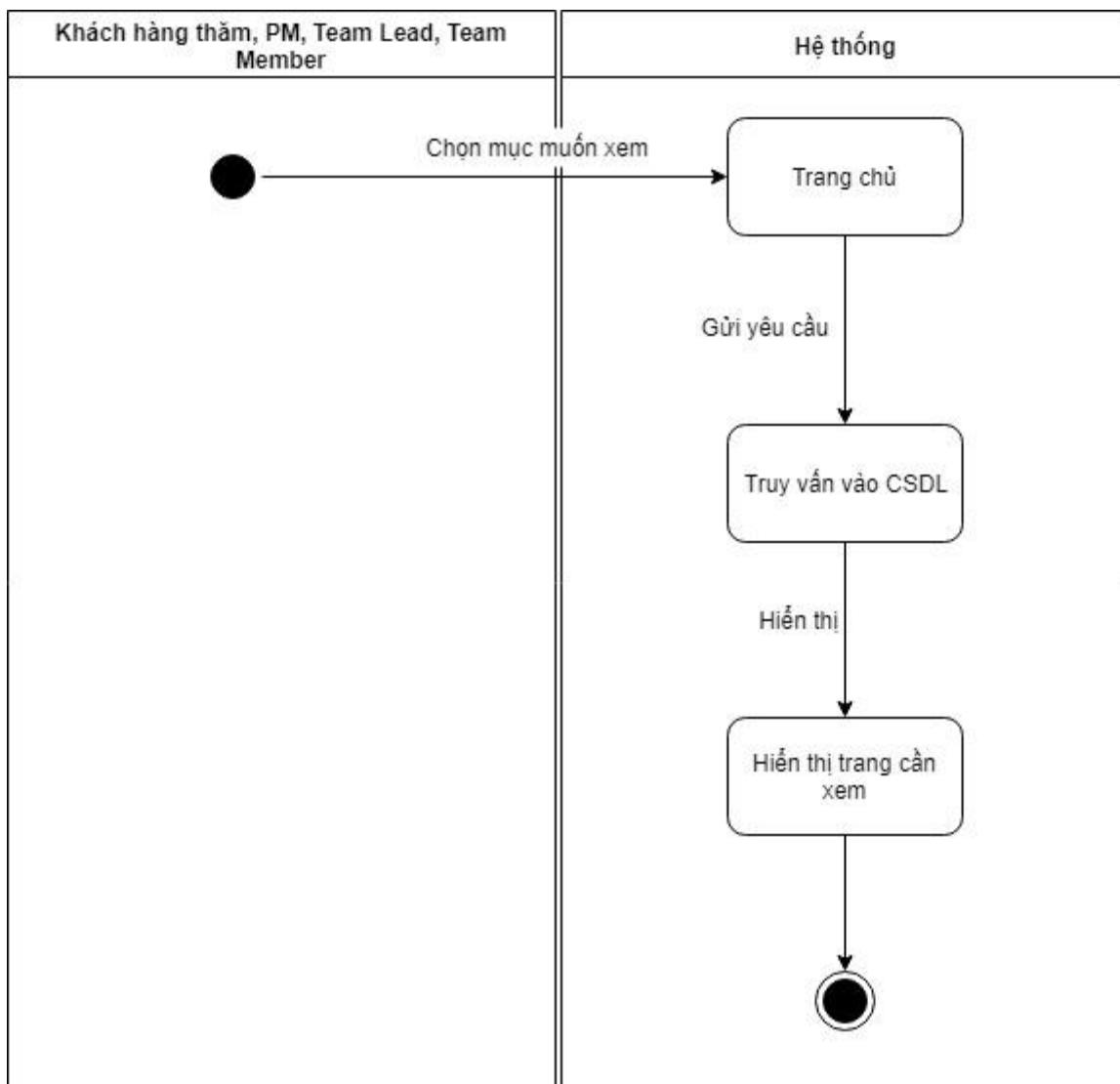
Hình 3.4.4. u. Biểu đồ hoạt động thực hiện công việc được giao.

3.4.4.13. Biểu đồ hoạt động thống kê tiến độ dự án cá nhân



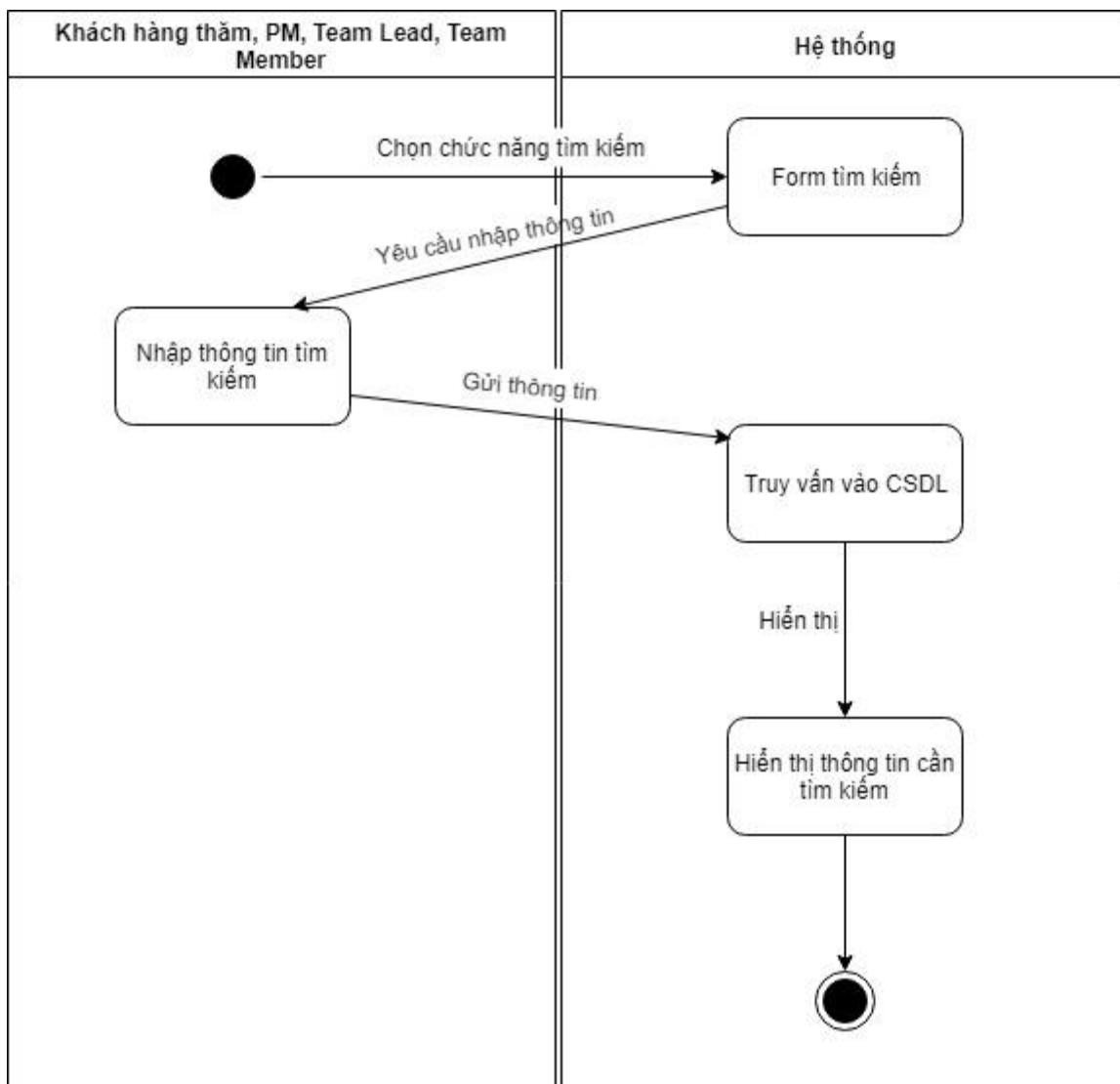
Hình 3.4.4. v. Biểu đồ hoạt động chức năng thống kê tiến độ dự án.

3.4.4.14. Xem chi tiết



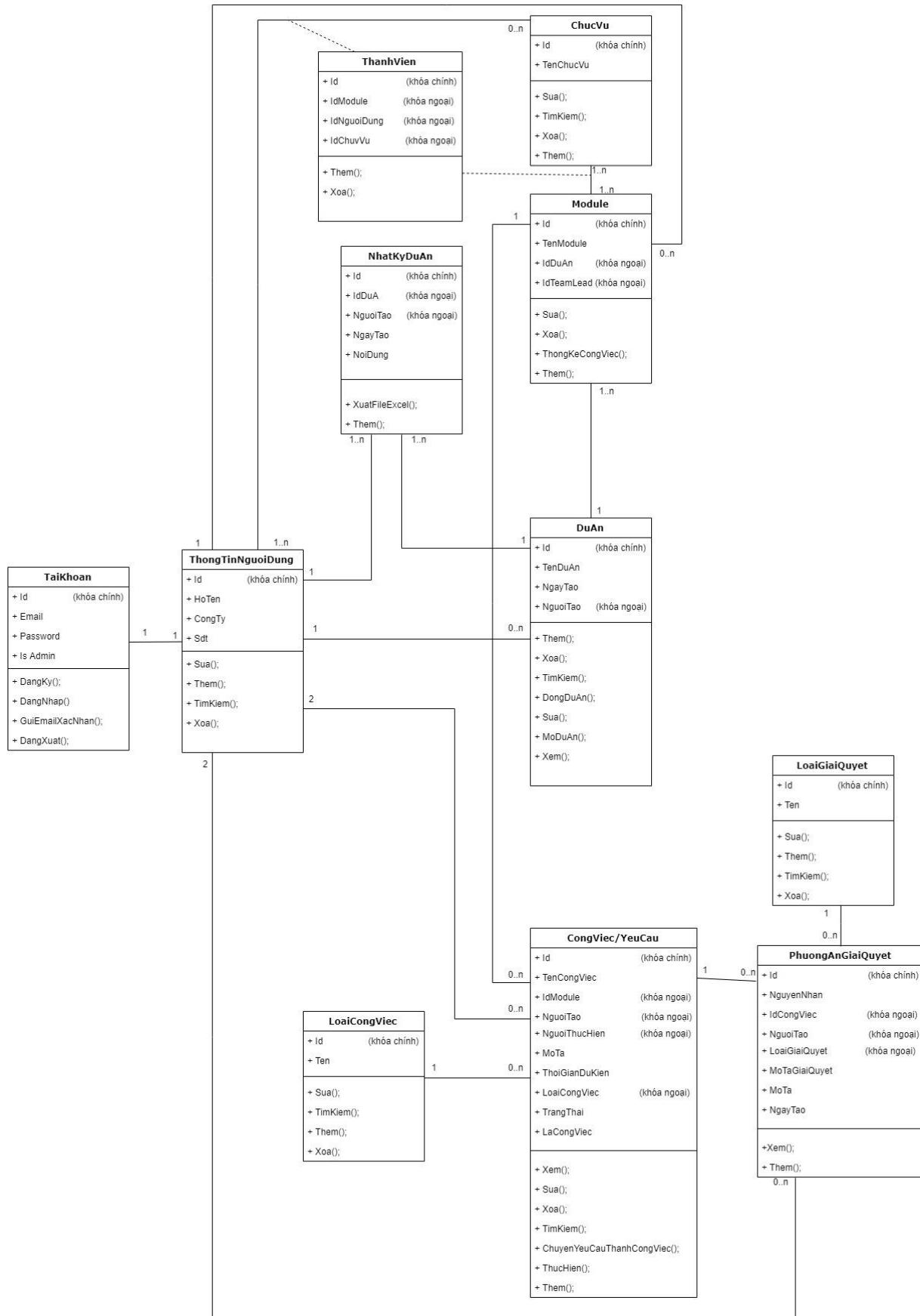
Hình 3.4.4. w. Biểu đồ hoạt động chức năng xem chi tiết.

3.4.4.15. Tìm kiếm



Hình 3.4.4. x. Biểu đồ hoạt động chức năng tìm kiếm.

3.4.5. Biểu đồ lớp



Hình 3.4.5. a. Biểu đồ lớp.

Dựa vào kịch bản mô tả bài toán, quy trình khảo sát và phân tích, ta xác định được các lớp đối tượng:

- TaiKhoan: là lớp chứa thông tin tài khoản của người dùng trong hệ thống
- ThongTinNguoiDung: Là lớp chứa thông tin người dùng trong hệ thống
- ChucVu: Là lớp chứa thông tin chức vụ của các dự án trong hệ thống
- LoaiCongViec: Là lớp chứa thông tin loại công việc của các công việc trong các dự án
- Module: Là lớp chứa thông tin của module trong các dự án
- CongViec/YeuCau: Là lớp chứa thông tin công việc trong các module, trường LaCongViec quy định đối tượng đó là công việc hay là yêu cầu.
- NhatKyDuAn: Lớp chứa thông tin của nhật ký dự án
- ThanhVien: Là lớp chứa thông tin các thành viên trong dự án
- PhuongAnGiaiQuyet: Chứa thông tin của phương án giải quyết một công việc nào đó
- LoaiGiaiQuyet: Chứa thông tin các loại phương án giải quyết.

3.4.6. Thiết kế cơ sở dữ liệu vật lý

a. Bảng lưu trữ thông tin người dùng

Bảng 3.4.6. a. Bảng thông tin người dùng (UserInfo)

Tên cột	Kiểu dữ liệu	Khóa	Mô tả
UserName	varchar(50)	Chính	Tên đăng nhập
Password	varchar(200)		Mật khẩu
Email	varchar(200)		Email
Sdt	varchar(200)		Số điện thoại
FullName	varchar(200)		Họ và tên
CurrentJob	varchar(200)		Công việc hiện tại
Company	varchar(200)		Công ty
IsAdmin	bit		Quy định tài khoản là QTHT

b. Bảng lưu trữ thông tin chức vụ ngành CNTT trong hệ thống

Bảng 3.4.6. b. Bảng chức vụ (JobRole)

Tên cột	Kiểu dữ liệu	Khóa	Mô tả
Id	int	Chính	Id
Title	nvarchar(50)		Tiêu đề chức vụ
Description	ntext		Mô tả về chức vụ
IsActive	bit		Quy định chức vụ có được sử dụng trong hệ thống
CanDelete	bit		Quy định chức vụ có được xóa trong hệ thống

c. Bảng lưu trữ thông tin dự án

Bảng 3.4.6. c. Bảng dự án (Project)

Tên cột	Kiểu dữ liệu	Khóa	Mô tả
Id	int	Chính	Id
Name	nvarchar(100)		Tên dự án
[From]	date		Ngày bắt đầu
[To]	date		Ngày kết thúc
CreatedBy	varchar(50)	Ngoại	Username người tạo dự án
CreatedDate	date		Ngày tạo
StatusId	int	Ngoại	Trạng thái của dự án
IsActive	bit		Quy định dự án còn khả dụng trong hệ thống
Code	nvarchar(100)		Tên viết tắt của dự án
Description	nvarchar(300)		Mô tả dự án

d. Bảng lưu trữ thành viên trong module

Bảng 3.4.6. d. Bảng thành viên (RoleInProject)

Tên cột	Kiểu dữ liệu	Khóa	Mô tả
Id	int	Chính	Id
ModuleId	int	Ngoại	Id của module
RoleId	int	Ngoại	Id chức vụ
UserName	varchar(50)	Ngoại	Username thành viên trong module
IsActive	bit		Quy định tính khả dụng của thành viên trong module

e. Bảng lưu trữ thông tin các module trong dự án

Bảng 3.4.6. e. Bảng module (Module)

Tên cột	Kiểu dữ liệu	Khóa	Mô tả
Id	int	Chính	Id
Title	nvarchar(100)		Tên của module
ProjectId	int	Ngoại	Id của dự án
TeamLead	varchar(50)	Ngoại	Username của team lead
IsActive	bit		Quy định tính khả dụng của module trong dự án

f. Bảng lưu trữ loại công việc

Bảng 3.4.6. f. Bảng loại công việc (TaskType)

Tên cột	Kiểu dữ liệu	Khóa	Mô tả
Id	int	Chính	Id
Title	nvarchar(50)		Tên loại công việc
IsActive	bit		Quy định tính khả dụng của loại công việc trong hệ thống
Description	ntext		Mô tả loại công việc

g. Bảng lưu trữ thông tin công việc / yêu cầu

Bảng 3.4.6. g. Bảng công việc / yêu cầu (Task)

Tên cột	Kiểu dữ liệu	Khóa	Mô tả
Id	int	Chính	Id
Title	nvarchar(100)		Tiêu đề công việc
AssignedTo	varchar(50)	Ngoại	Username người thực hiện công việc
CreatedBy	varchar(50)	Ngoại	Username người tạo công việc
TaskType	nvarchar(100)		Loại công việc

ModuleId	int	Ngoại	Id module
[From]	datetime		Thời gian bắt đầu công việc
[To]	datetime		Thời gian công việc kết thúc
Severity	nvarchar(100)		Quy định độ nghiêm trọng của bug
Priority	nvarchar(100)		Quy định độ ưu tiên của công việc
StatusId	int	Ngoại	Trạng thái công việc
IsTask	bit		Quy định công việc có phải là yêu cầu từ member trong dự án hay không. Mang giá trị true nếu là công việc
Description	ntext		Mô tả công việc
IsActive	bit		Quy định tính khả dụng của công việc trong module

i. Bảng lưu trữ thông tin nhật ký dự án

Bảng 3.4.6. h. Bảng nhật ký dự án (ProjectLog)

Tên cột	Kiểu dữ liệu	Khóa	Mô tả
Id	int	Chính	Id
[Content]	int		Nội dung nhật ký
ProjectId	int	Ngoại	Id của dự án
CreatedBy	varchar(50)	Ngoại	Username người tạo
CreatedDate	int		Thời gian tạo

j. Bảng lưu trữ thông tin phương án giải quyết công việc

Bảng 3.4.6. i. Bảng phương án giải quyết công việc (Solution)

Tên cột	Kiểu dữ liệu	Khóa	Mô tả
Id	int	Chính	Id

TaskId	int	Ngoại	Id công việc
Reason	ntext		Nguyên nhân
Solution	ntext		Cách giải quyết
Description	ntext		Mô tả
CreatedBy	varchar(50)	Ngoại	Username người tạo giải pháp
CreatedDateTime	datetime		Thời gian tạo
ResolveType	int	Ngoại	Loại giải quyết

k. Bảng lưu trữ thông tin loại giải quyết

Bảng 3.4.6. j. Bảng loại giải quyết (ResolveType)

Tên cột	Kiểu dữ liệu	Khóa	Mô tả
Id	int	Chính	Id
Title	nvarchar(50)		Tên loại giải quyết
IsActive	bit		Quy định loại giải quyết có khả dụng trong hệ thống hay không
Description	ntext		Mô tả loại giải quyết

l. Bảng lưu trữ các loại trạng thái

Bảng 3.4.6. k. Bảng loại trạng thái (LookupStatus)

Tên cột	Kiểu dữ liệu	Khóa	Mô tả
Id	int	Chính	Id
IsProject	bit		Quy định trạng thái này là của đối tượng dự án
IsTask	bit		Quy định trạng thái này là của đối tượng công việc
IsActive	bit		Quy định tính khả dụng của trạng thái trong hệ thống
Name	nvarchar(50)		Tên loại trạng thái

CHƯƠNG 4. XÂY DỰNG CHƯƠNG TRÌNH

Đây là chương trình bày về cách thiết kế giao diện và một số giao diện tiêu biểu trong phần mềm và một số mã code đại diện cho chương trình. Ngoài ra, em còn liệt kê ra những điều cần thiết để cài đặt hệ thống.

4.1. Xây dựng hệ thống

4.1.1. Giao diện chương trình

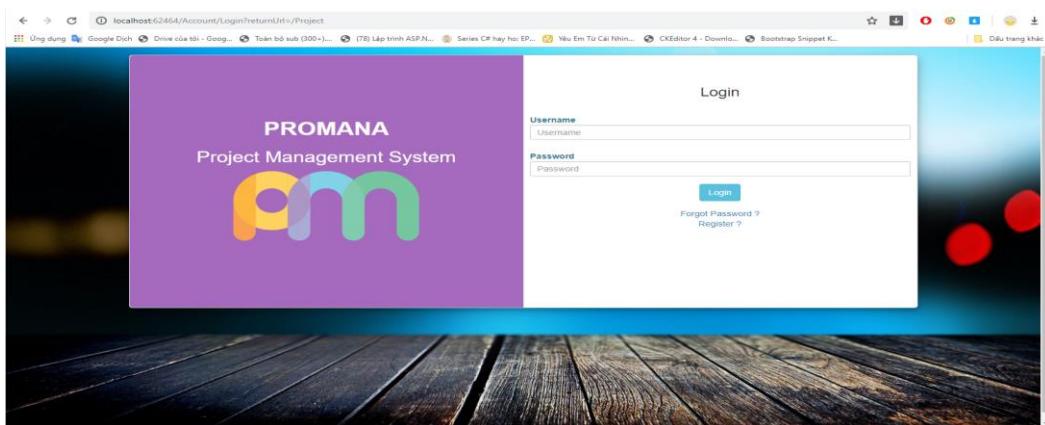
Cách thức thiết kế giao diện website dựa vào các bước sau:

- Bước 1: Xác định yêu cầu của người dùng
- Bước 2: Phác thảo ý tưởng trên giấy
- Bước 3: Phối màu cho giao diện web
- Bước 4: Xây dựng tài liệu về chuẩn CSS, clientsite script, ảnh, folder cho trang web
- Bước 5: Sử dụng các ngôn ngữ đánh dấu, lập trình để thiết kế giao diện
- Bước 6: Test giao diện trên các trình duyệt
- Bước 7: Chuyển mã nguồn tới bộ phận phát triển web

4.1.1.1. Giao diện đăng nhập

Giao diện đăng nhập bao gồm:

- Một nút “login” để tiến hành đăng nhập sau khi nhập đúng tài khoản và mật khẩu
- Một nút “Forgot password” để cài đặt lại mật khẩu
- Một nút Register để tiến hành đăng ký.



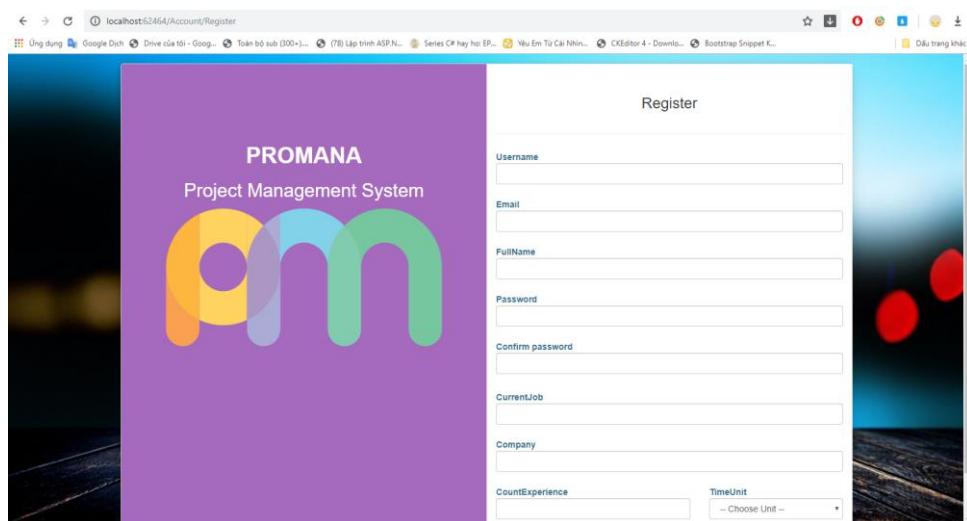
Hình 4.1.1. a. Giao diện đăng nhập.

4.1.1.2. Giao diện đăng ký

Giao diện đăng ký bao gồm:

- Textbox username để điền mật khẩu.
- Textbox Email để điền email
- Textbox Password để nhập mật khẩu
- Textbox Confirm password để xác nhận lại mật khẩu
- Nút Register để gửi thông tin đăng ký lên hệ thống

Khi đăng nhập xảy ra lỗi hệ thống sẽ trả về một dòng chữ báo lỗi màu đỏ ở phía trên nút Register.



Hình 4.1.1. b. Giao diện đăng ký.

4.1.1.3. Giao diện trang quản lý chức vụ trong ngành CNTT

Trang này bao gồm các chức năng thêm, sửa xóa chức vụ.

A screenshot of the PROMANA Admin interface. The left sidebar shows navigation links for "Job Role", "Task Type", "Resolve Type", "Project", and "Account". The main content area is titled "Job Role List" and displays a table of job roles. The table has columns for "Title" and "Description". The data in the table is as follows:

Title	Description	Action
Developer		Edit Delete
PM	Project Manager	Edit Delete
QA		Edit Delete
Team Lead		Edit Delete
Tester		Edit Delete
Watcher		Edit Delete

At the bottom of the page, there is a copyright notice: "Copyright © 2019 Promana. All rights reserved." and a placeholder text: "Anything you want".

Hình 4.1.1. c. Giao diện trang quản lý chức vụ.

4.1.1.4. Giao diện trang quản lý loại công việc

Trang này bao gồm các chức năng thêm, sửa xóa loại công việc.

The screenshot shows a web application interface titled "Task Type List". On the left, there is a sidebar with navigation links: "Job Role", "Task Type" (selected), "Resolve Type", "Project", and "Account". The main content area displays a table with three rows. The columns are "Title" and "Description". The rows contain "Bug" (description: "aasd"), "Study" (description: "asdasd"), and "Task". Each row has "Edit | Delete" links. Below the table, it says "Showing 1 to 3 of 3 entries". At the bottom right, there are "Previous", "1", and "Next" buttons. The footer includes copyright information and a search bar.

Title	Description
Bug	aasd
Study	asdasd
Task	

Hình 4.1.1. d. Giao diện trang quản lý loại công việc.

4.1.1.5. Giao diện trang quản lý loại phương án giải quyết

Trang này bao gồm các chức năng thêm, sửa xóa loại phương án giải quyết.

The screenshot shows a web application interface titled "List Resolve Type". On the left, there is a sidebar with navigation links: "Job Role", "Task Type", "Resolve Type" (selected), "Project", and "Account". The main content area displays a table with two rows. The columns are "Title" and "Description". The rows contain "Create" (description: "createeeee") and "Fixed" (description: "123"). Each row has "Edit | Delete" links. Below the table, it says "Showing 1 to 2 of 2 entries". At the bottom right, there are "Previous", "1", and "Next" buttons. The footer includes copyright information and a search bar.

Title	Description
Create	createeeee
Fixed	123

Hình 4.1.1. e. Giao diện trang quản lý loại phương án giải quyết công việc.

4.1.1.6. Giao diện trang quản lý dự án trong hệ thống

Trang này bao gồm các chức năng thêm, sửa dự án trong hệ thống.

The screenshot shows a web application interface titled "Promana". On the left, there is a sidebar with navigation links: "Job Role", "Task Type", "Resolve Type", "Project" (selected), "Project List", "Create Project", and "Account". The main content area is titled "List Of Project" and contains a table with 10 entries. The columns are Name, Code, Time, Owner, and Status. Each row has an "Edit | Delete" link. At the bottom of the table, there are buttons for "Previous", "1", and "Next". The footer includes copyright information and a placeholder for user input.

Name	Code	Time	Owner	Status
ASP.NET MVC 6	AM6	Sunday, July 28, 2019 - Saturday, August 31, 2019	pmttest	Opened
Automation Testing Tool 2.0	TT2	Sunday, August 4, 2019	devtest	Opened
Font awesome css	fc2	Sunday, July 28, 2019 - Friday, August 30, 2019	devtest	Opened
Home Page	HP	Sunday, July 28, 2019	pmttest	Opened
Iden titty sample 3.0	IS3	Sunday, July 28, 2019	devtest	Opened
Logo PNG	FI22	Sunday, July 28, 2019 - Saturday, August 24, 2019	pmttest	Opened
Petronas Management	PET	Sunday, July 28, 2019 - Saturday, August 31, 2019	pmttest	Opened
Responsiblity UI	RUI	Monday, July 29, 2019	watchertest	Opened
SQL Serrver 2020	SS220	Sunday, July 28, 2019	devtest	Opened
Viewnamese FB	VFB	Sunday, July 28, 2019	devtest	Opened

Hình 4.1.1. f. Giao diện trang quản lý dự án trong hệ thống của người QTHT.

4.1.1.7. Giao diện trang quản lý tài khoản trong hệ thống

Trang này bao gồm các chức năng thêm, sửa xóa tài khoản trong hệ thống.

The screenshot shows a web application interface titled "Promana". On the left, there is a sidebar with navigation links: "Job Role", "Task Type", "Resolve Type", "Project", "Account" (selected), "Account List", and "Create Account". The main content area is titled "Index" and contains a table with 5 entries. The columns are Username, FullName, Current Job, and Company. Each row has an "Edit | Delete" link. At the bottom of the table, there are buttons for "Previous", "1", and "Next". The footer includes copyright information and a placeholder for user input.

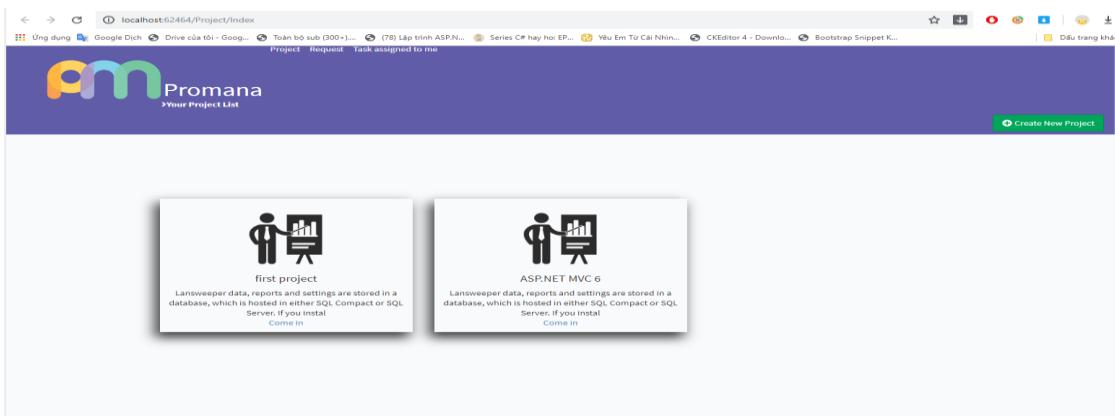
Username	FullName	Current Job	Company
devtest	Dev Test	Dev	123
pmttest	PM Test	Project Manager	1123
teamleadtest	Team Lead Test	Team Lead	123
testertest	Tester Test	Tester	123
watchertest	Watcher Test	BrsE	123

Hình 4.1.1. g. Giao diện trang quản lý tài khoản.

4.1.1.8. Giao diện trang danh sách dự án người dùng tham gia

Trang này hiển thị các dự án mà người sử dụng hệ thống đang tham gia theo từng ô card. Tại mỗi ô, hệ thống sẽ hiển thị cho người dùng bao gồm:

- Tên dự án
- Ngày tạo
- Mô tả của dự án
- Nút “Come in” để điều hướng người dùng tới trang thông tin chi tiết của dự án.



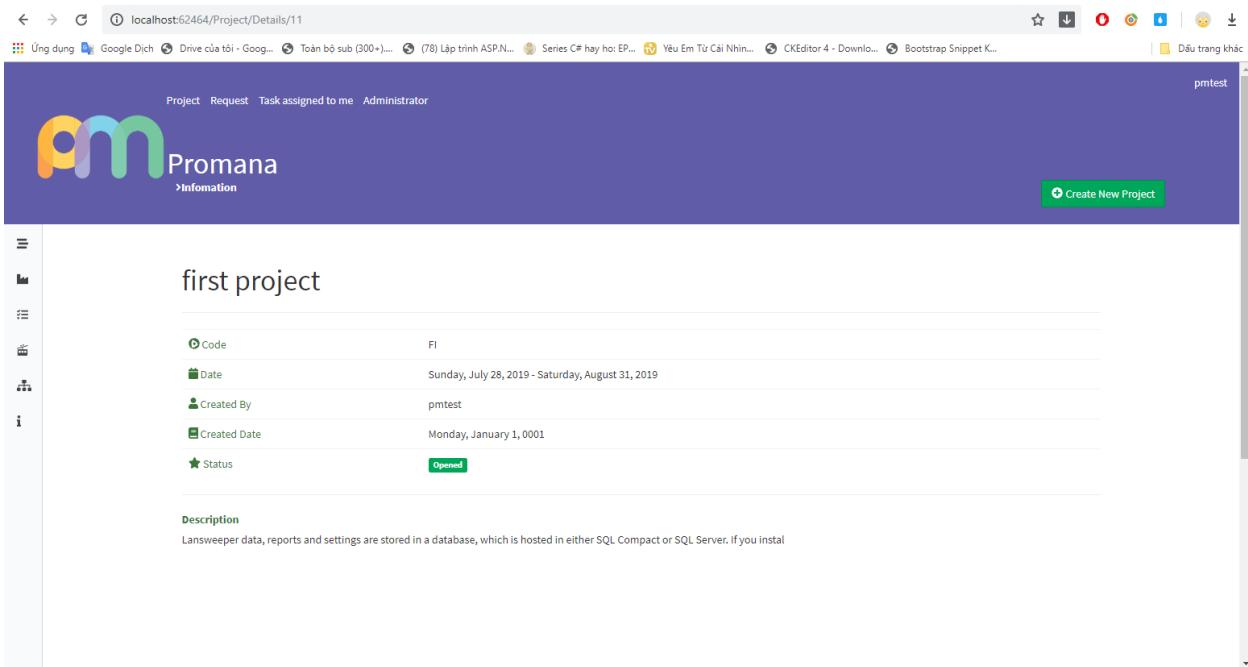
Hình 4.1.1. h. Giao diện trang danh sách dự án người dùng tham gia.

4.1.1.9. Giao diện trang thông tin dự án của người sử dụng

Trang này hiển thị thông tin của dự án cụ thể của người dùng.

Thanh menu bên trái bao gồm:

- Nút “Gantt Chart” điều hướng tới trang biểu đồ Gantt của các công việc trong dự án
- Nút “Kanban Board” điều hướng tới trang thông kê danh sách công việc dưới dạng bảng Kanban
- Nút “Task List” điều hướng tới trang hiển thị danh sách công việc trong dự án
- Nút “Request List” điều hướng tới trang hiển thị danh sách yêu cầu.



Hình 4.1.1. i. Giao diện thông tin dự án.

4.1.1.10. Giao diện trang danh sách công việc/yêu cầu của người sử dụng

Trang này hiển thị danh sách các dự án trong hệ thống bao gồm:

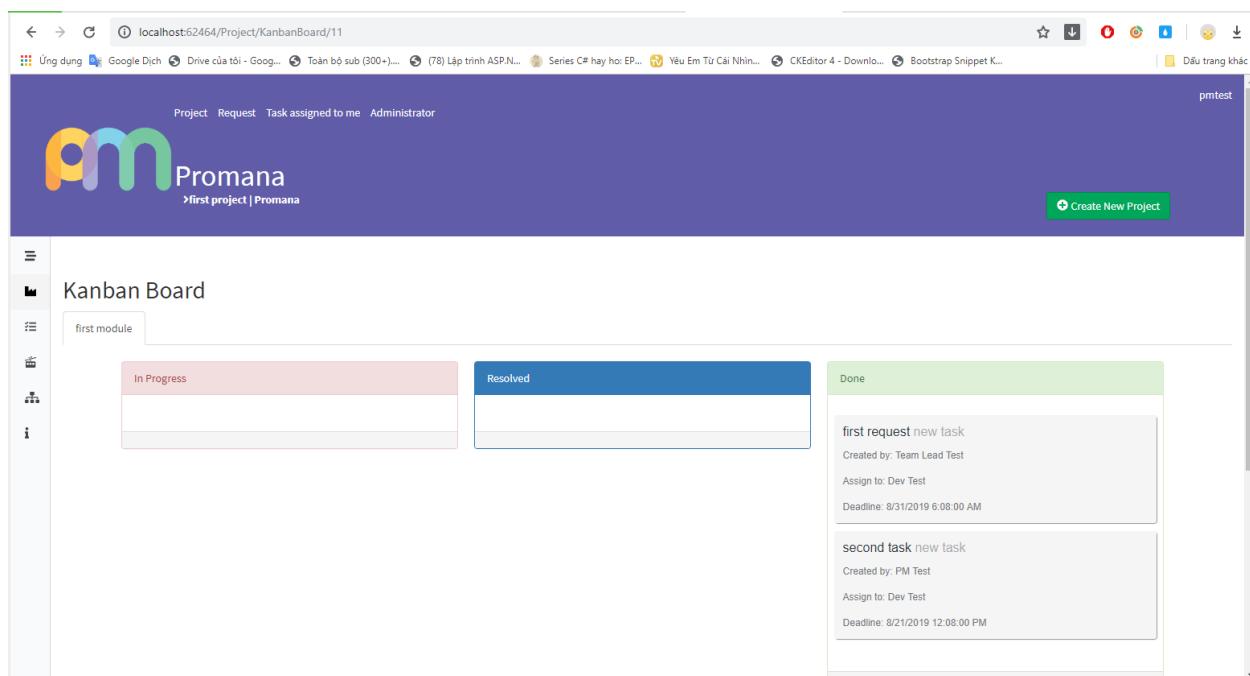
- Thanh bên trái liệt kê tất cả các danh sách dự án theo từng ô chữ nhặt
- Khi người dùng click chọn một công việc trong danh sách, thông tin chi tiết của công việc đó sẽ được hiển thị ở phần giao diện bên phải.
- Chi tiết công việc có bảng “Solution” là bảng giải pháp đã được áp dụng cho công việc đó. Khi click vào từng dòng giải pháp, hệ thống sẽ hiển thị một popup thông tin chi tiết về giải pháp đó
- Đối với trang danh sách yêu cầu cũng có giao diện tương tự.

Hình 4.1.1. j. Giao diện trang danh sách công việc.yêu cầu của người sử dụng.

4.1.1.11. Giao diện trang thống kê công việc theo bảng Kanban

Trang này thống kê các công việc trong dự án theo dạng bảng Kanban bao gồm:

- Cột “In Progress”: Hiển thị các công việc đang được tiến hành
- Cột “Resolve”: Hiển thị các công việc đã được giải quyết và chờ xác nhận từ người tạo ra công việc đó (PM / Team lead)
- Cột “Done”: Hiển thị các công việc sau khi đã được người tạo xác nhận hoàn thành đúng yêu cầu
- Mỗi lần người dùng kéo công việc từ cột “In Progress” sang cột “Resolve”, một popup sẽ được hiển thị để yêu cầu người dùng nhập giải pháp thực hiện công việc đó.



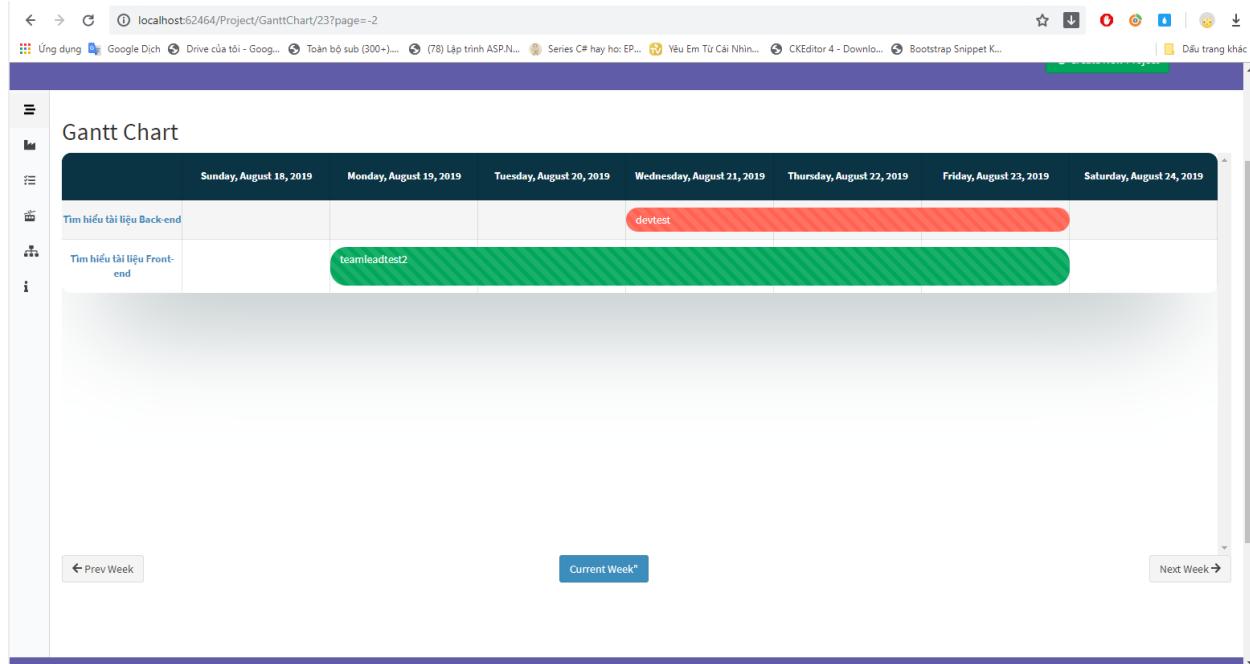
Hình 4.1.1. k. Giao diện trang thống kê công việc theo bảng Kanban.

4.1.1.12. Giao diện trang thống kê công việc theo bảng Gantt

Trang này thống kê các công việc trong dự án theo dạng bảng Gantt bao gồm:

- Hàng trên cùng là danh sách các ngày trong tuần, mặc định sẽ là tuần hiện tại
- Cột ngoài cùng bên trái là tiêu đề các công việc có trong tuần đó
- Các thanh tiến trình của từng công việc sẽ thể hiện công việc đó được thực hiện từ ngày nào đến ngày nào. Trên thanh tiến trình hiển thị username của người thực hiện công việc đó
- Ngày hiện tại sẽ được phủ bởi lớp màu đen mờ

- Nút “Prev Week” sẽ điều hướng bảng Gantt đến tuần trước đó
- Nút “Next Week” sẽ điều hướng bảng Gantt đến tuần tiếp theo
- Nút “Current Week” sẽ điều hướng bảng Gantt trở lại tuần hiện tại.



Hình 4.1.1. l. Giao diện trang thống kê công việc theo bảng Gantt.

4.1.1.13. Giao diện trang quản lý thông tin dự án

Trang này cho phép người quản lý dự án thay đổi thông tin của dự án, bao gồm các chức năng Open và Close dự án. Tại thông tin của mỗi module sẽ có nút “Edit” để sửa module đó.

Information				
Name	Hệ thống quản lý thư viện UTT			
Code	UTTLibs			
PM	pmtest			
From	04/01/2019		To	04/04/2020
Description	Hệ thống sử dụng ASP.NET Core và Angular			
Watcher	Username	Fullname	Current Job	Company
	watchertest	Watchertest Test	BraE	ASB
	watchertest2	Đất văn chúa 2	BraE	ASB
				Remove
				Remove

Modules					
Team Back-end					
Edit					
Show	10	entries	Search:		
Username	Full Name	Module	Job Role	Join Date	
devtest	Dev Test	Team Back-end	Developer	Monday, January 1, 1900	
teamleadtest	Team Lead Test	Team Back-end	Team Lead	Monday, January 1, 1900	
testertest	Tester Test	Team Back-end	Tester	Monday, January 1, 1900	
Showing 1 to 5 of 5 entries					

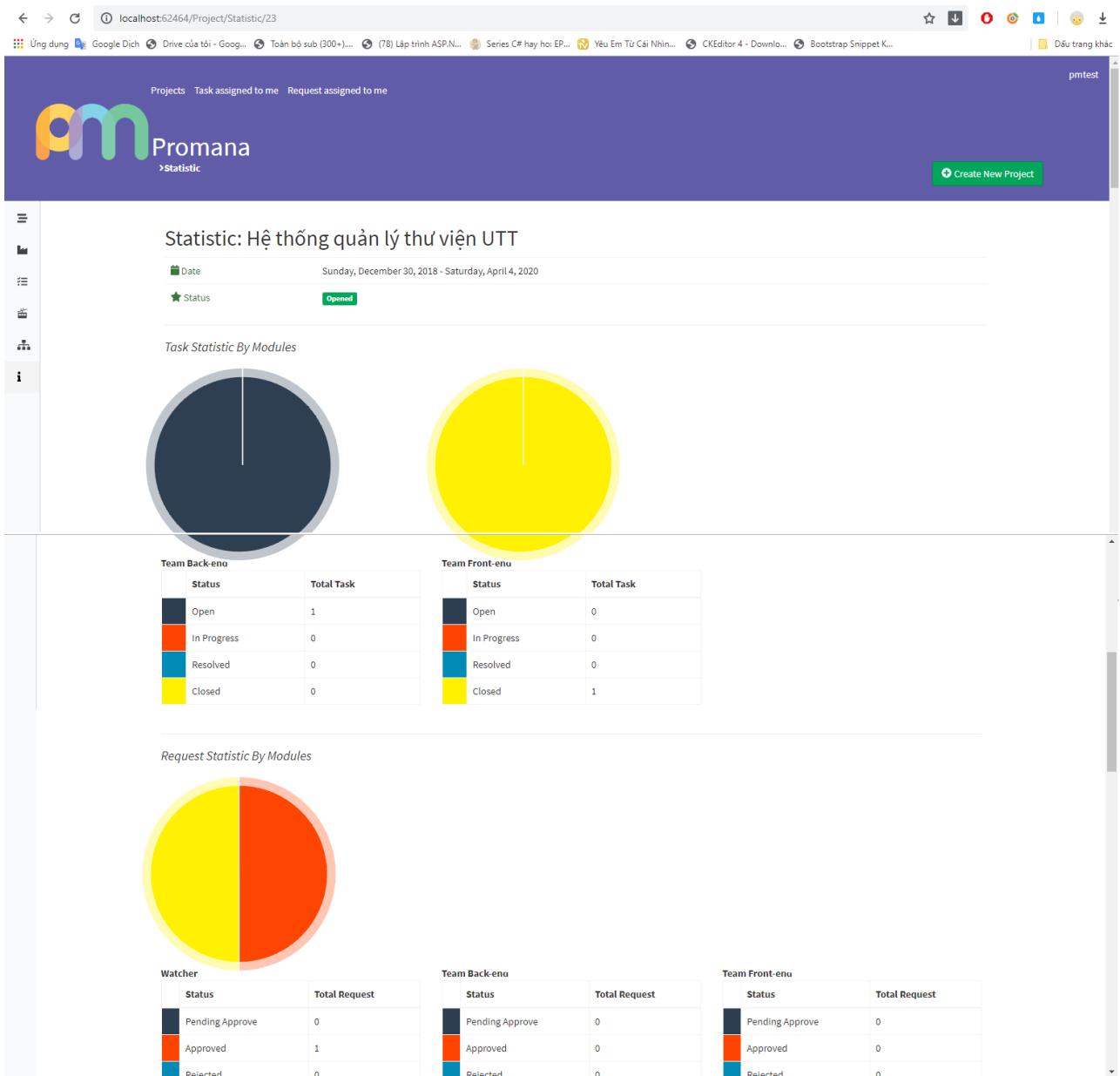
Team Front-end					
Edit					
Show	10	entries	Search:		
Username	Full Name	Module	Job Role	Join Date	
devtest2	Dev 2	Team Front-end	Developer	Monday, January 1, 1900	
teamleadtest2	team lead 2	Team Front-end	Team Lead	Monday, January 1, 1900	
testertest2	tester 2	Team Front-end	Tester	Monday, January 1, 1900	
Showing 1 to 3 of 3 entries					

Hình 4.1.1. m. Giao diện trang quản lý thông tin dự án.

4.1.1.14. Giao diện trang thống kê tiến độ dự án

Trang này cho phép người quản lý dự án thống kê chất lượng của dự án, bao gồm:

- Thống kê công việc theo từng module theo dạng biểu đồ tròn và bảng
- Thống kê yêu cầu đến từng module theo dạng biểu đồ tròn và bảng
- Thống kê công việc theo từng loại công việc theo dạng biểu đồ cột
- Thống kê số lượng công việc và số lượng yêu cầu theo từng thành viên trong dự án
- Hiển thị nhật ký dự án và nút “Download Excel” cho phép tải file excel nhật ký dự án.



Hình 4.1.1. n. Giao diện trang thống kê tiến độ dự án.

4.1.1.15. Giao diện trang quản lý module trong dự án

Trang này bao gồm các chức năng thêm, sửa, xóa module trong dự án.

The screenshot shows a web application interface for managing modules. At the top, there is a header bar with the Promana logo and the text 'Hệ thống quản lý thư viện UTT'. Below the header, there is a navigation bar with links for 'Projects', 'Task assigned to me', and 'Request assigned to me'. On the right side of the header, there is a dropdown menu labeled 'pmtest'. The main content area is titled 'Modules' and contains a table with two rows of data. The columns are 'Title' and 'TeamLead'. The first row has 'Team Back-end' under 'Title' and 'Team Lead Test (teamleadtest)' under 'TeamLead'. The second row has 'Team Front-end' under 'Title' and 'team lead 2 (teamleadtest2)' under 'TeamLead'. Each row has 'Edit | Delete' links at the end. At the bottom of the table, it says 'Showing 1 to 2 of 2 entries'. There are also 'Previous' and 'Next' buttons. On the left side of the main content area, there is a sidebar with icons for 'Add new', 'Edit', 'Delete', and 'Info'.

Hình 4.1.1. o. Giao diện trang quản lý module trong dự án.

4.1.2. Một số mã code

Dưới đây là một số mã code đại diện cho chương trình website được thiết kế theo mô hình Repository với MVC làm tầng giao diện:

a. Đại diện cho tầng DTO

```
public partial class Task
{
    [System.Diagnostics.CodeAnalysis.SuppressMessage("Microsoft.Usage",
    "CA2214:DoNotCallOverridableMethodsInConstructors")]
    public Task()
    {
        Solutions = new HashSet<Solution>();
    }
    public int Id { get; set; }

    [Required]
    [StringLength(100)]
    public string Title { get; set; }

    [Required]
    [StringLength(50)]
    public string CreatedBy { get; set; }
```

```

[Required]
[StringLength(50)]
public string AssignedTo { get; set; }

public string TaskType { get; set; }

public int? ModuleId { get; set; }

public DateTime From { get; set; }

public DateTime To { get; set; }

public string Severity { get; set; }
public string Priority { get; set; }
public bool IsTask { get; set; }

public int StatusId { get; set; }

[Column(TypeName = "ntext")]
public string Description { get; set; }

public bool IsActive { get; set; }

public virtual LookupStatus LookupStatus { get; set; }

public virtual Module Module { get; set; }

public virtual ICollection<Solution> Solutions { get; set; }
public virtual UserInfo UserInfo { get; set; }
public virtual UserInfo UserInfo1 { get; set; }
public virtual LookupSeverity LookupSeverity { get; set; }
public virtual LookupPriority LookupPriority { get; set; }
}

}

```

b. Đại diện cho tầng DAL

```

public partial class DbProManaContext : DbContext
{
    public DbProManaContext()
        : base("name=DbProManaContext")
    {
        Database.SetInitializer(new Initialaze());
    }

    public virtual DbSet<JobRole> JobRoles { get; set; }
    public virtual DbSet<LookupStatus> LookupStatus { get; set; }
    public virtual DbSet<Module> Modules { get; set; }
    public virtual DbSet<Project> Projects { get; set; }
    public virtual DbSet<ProjectLog> ProjectLogs { get; set; }
    public virtual DbSet<Request> Requests { get; set; }
    public virtual DbSet<ResolveType> ResolveTypes { get; set; }
    public virtual DbSet<RoleInProject> RoleInProjects { get; set; }
}

```

```

public virtual DbSet<Solution> Solutions { get; set; }
public virtual DbSet<sysdiagram> sysdiagrams { get; set; }
public virtual DbSet<Task> Tasks { get; set; }
public virtual DbSet<TaskType> TaskTypes { get; set; }
public virtual DbSet<UserInfo> UserInfoes { get; set; }
public virtual DbSet<LookupSeverity> LookupSeverities { get; set; }
public virtual DbSet<LookupPriority> LookupPriorities { get; set; }

```

c. Đại diện cho tầng Repository

```

public virtual async Task<IEnumerable<T>> Get(Expression<Func<T, bool>> predicate, string
fieldOrderBy, bool @ascending, int skip, int take)
{
    var p = typeof(T).GetProperty(fieldOrderBy);
    var t = p.PropertyType;
    if (t == typeof(int))
    {
        var pe = Expression.Parameter(typeof(T), "p");
        var expr1 = Expression.Lambda<Func<T, int>>(Expression.Property(pe,
fieldOrderBy), pe);
        return await (ascending ?
_dbContext.Set<T>().Where(predicate).OrderBy(expr1).Skip(skip).Take(take).ToListAsync() :
_dbContext.Set<T>().Where(predicate).OrderByDescending(expr1).Skip(skip).Take(take).ToListAsync());
    }
    else if (t == typeof(DateTime))
    {
        var pe = Expression.Parameter(typeof(T), "p");
        var expr1 = Expression.Lambda<Func<T, DateTime>>(Expression.Property(pe,
fieldOrderBy), pe);
        return await (ascending ?
_dbContext.Set<T>().Where(predicate).OrderBy(expr1).Skip(skip).Take(take).ToListAsync() :
_dbContext.Set<T>().Where(predicate).OrderByDescending(expr1).Skip(skip).Take(take).ToListAsync());
    }
    else
    {
        var pe = Expression.Parameter(typeof(T), "p");
        var expr1 = Expression.Lambda<Func<T, string>>(Expression.Property(pe,
fieldOrderBy), pe);
        return await (ascending ?
_dbContext.Set<T>().Where(predicate).OrderBy(expr1).Skip(skip).Take(take).ToListAsync() :
_dbContext.Set<T>().Where(predicate).OrderByDescending(expr1).Skip(skip).Take(take).ToListAsync());
    }
}

```

d. Đại diện cho tầng Bussiness

Dưới đây là hàm có chức năng thêm dự án.

```
public async Task<bool> Create(Project  
project,IEnumerable<MemberParamsViewModel>members,string userCreate,  
List<string>errors)  
{  
    try  
    {  
        if (await Validate(project, errors) == false)  
        {  
            return false;  
        }  
        //insert project  
        project.CreatedBy = userCreate;  
        project.StatusId = ProjectStatusKey.Opened;  
        project.CreatedDate = DateTime.Now;  
  
        //insert watcher  
        var listMember = members.Select(m => new RoleInProject { IsActive = true, RoleId  
= HardFixJobRole.Watcher, UserName = m.Username }).ToList();  
        listMember.Add(new RoleInProject {  
            IsActive = true,  
            RoleId = HardFixJobRole.PM,  
            UserName = userCreate  
        });  
        project.Modules = new List<Module>()  
        {  
            new Module  
            {  
                IsActive=true,  
                Title=HardFixJobRoleTitle.Watcher,  
                RoleInProjects= listMember  
            }  
        };  
  
        //commit  
        _unitOfWork.Projects.Insert(project);  
        var result = await _unitOfWork.CommitAsync() > 0;  
  
        return result;  
    }  
    catch (Exception ex)  
    {  
        errors.Add(ex.Message);  
        return false;  
    }  
}
```

e. Đại diện cho tầng Controller trong MVC

Dưới đây là action có chức năng thêm dự án.

```
[HttpPost]
public async Task<ActionResult> Create(Project project, string members)
{
    try
    {
        bool result = false;
        if (ModelState.IsValid)
        {
            var listMembers =
JsonConvert.DeserializeObject<List<MemberParamsViewModel>>(members);
            result = await _projectBus.Create(project, listMembers,
User.Identity.GetUserName(), errors);
        }

        if (result)
        {
            return RedirectToAction("Index", "Module", new { id = project.Id });
        }
        else
        {
            ViewBag.GetUserDoNotInProject = await _projectBus.GetUserDoNotInProject(0);
            ViewBag.GetSoftRole = await _jobRoleBUS.GetSoftRole();
            ViewBag.Errors = errors;
            return View(project);
        }
    }
    catch
    {
        ViewBag.GetUserDoNotInProject = await _projectBus.GetUserDoNotInProject(0);
        ViewBag.GetSoftRole = await _jobRoleBUS.GetSoftRole();
        return View(project);
    }
}
```

f. Đại diện cho tầng View trong MVC

View dưới đây là code giao diện của form thêm dự án.

```
@using (Html.BeginForm("Create", "Project", FormMethod.Post, new { @id = "form" }))
{
    @Html.AntiForgeryToken()

<div class="form-horizontal">
    <hr />
    @Html.ValidationSummary(true, "", new { @class = "text-danger" })
    @{
        var errors = (List<string>
    )ViewBag.Errors;
```

```

if (errors != null && errors.Any())
{
    <ul class="list-group">
        @foreach (var item in errors)
        {
            <li class="list-group-item text-danger">@item</li>
        }
    </ul>
}
<input name="CreatedBy" type="hidden" value="abc"/>
<div class="form-group">
    @Html.LabelFor(model => model.Name, htmlAttributes: new { @class = "control-label col-md-2" })
    <div class="col-md-10">
        <input name="Name" id="Name" type="text" class="form-control" />
    </div>
</div>

<div class="form-group">
    @Html.LabelFor(model => model.Code, htmlAttributes: new { @class = "control-label col-md-2" })
    <div class="col-md-10">
        @Html.EditorFor(model => model.Code, new { htmlAttributes = new { @class = "form-control" } })
        @Html.ValidationMessageFor(model => model.Code, "", new { @class = "text-danger" })
    </div>
</div>

<div class="form-group">
    @Html.LabelFor(model => model.From, htmlAttributes: new { @class = "control-label col-md-2" })
    <input type="text" onchange="ValidateDateTo()" autocomplete="off" value="" class="control-label col-md-3" id="from" style="margin-left:1.3% !important" />
    <input type="hidden" name="From" id="fromValue" />
    @Html.LabelFor(model => model.To, htmlAttributes: new { @class = "control-label col-md-2" })
    <input type="text" autocomplete="off" value="" class="control-label col-md-3" id="to" />
    <input type="hidden" name="To" id="toValue" />
</div>

<div class="form-group">
    @Html.LabelFor(model => model.Description, htmlAttributes: new { @class = "control-label col-md-2" })
    <div class="col-md-10">
        <textarea name="Description" id="Description" class="form-control"></textarea>
        @Html.ValidationMessageFor(model => model.Description, "", new { @class = "text-danger" })
    </div>

```

```

</div>
<div class="form-group">
    <label class="control-label col-md-2">Watcher</label>
    <div class="col-md-10">
        <div>
            <table class="table table-bordered table-striped">
                <thead>
                    <tr>
                        <th>Username</th>
                        <th>Fullname</th>
                        <th>Current Job</th>
                        <th>Company</th>
                        <th></th>
                    </tr>
                </thead>
                <tbody id="listMember"></tbody>
            </table>
        </div>
        <label style="width:100% !important" onclick="#memberModal).modal('show')"">
            <button type="button" class="btn btn-default"><i class="fas fa-plus-circle"></i></button>
        </label>
    </div>
    <input type="hidden" name="members" id="members" />
    <div class="form-group">
        <div class="col-md-offset-2 col-md-10">
            <input id="submit" type="submit" value="Create" class="btn btn-success" />
        </div>
    </div>

```

4.2. Cài đặt hệ thống

Để tiến hành cài đặt hệ thống, yêu cầu gồm có:

- Tên miền và máy tính cài đặt hệ điều hành Window Server chạy IIS 7.0
- SQL Server 2014
- Visual Studio 2017.

CHƯƠNG 5. KẾT LUẬN

Trong thời gian làm đồ án này đã giúp em tìm hiểu và nắm bắt được nhiều kiến thức bổ ích, đồng thời rút ra nhiều kinh nghiệm thực tế. Em xin chân thành cảm ơn và mong nhận được sự đóng góp ý kiến từ các thầy cô và các bạn để đồ án được hoàn thiện hơn.

Qua đợt tài xâ dựng và thiết kế hệ thống quản lý dự án công nghệ thông tin, em đã trau dồi được thêm kiến thức về HTML, CSS, Javascript trong thiết kế giao diện và kiến thức về C#, ASP.NET MVC, Entity Framework, SQL Server trong việc xây dựng backend. Từ những kiến thức đã được trau dồi, đồ án của em đã hoàn thiện và đạt được các kết quả sau: Đã phân tích, khảo sát và làm rõ được các quy nghiệp cũng như nghiệp vụ của hệ thống quản lý dự án công nghệ thông tin từ đó vẽ được biểu đồ use case tổng quát, phân rã các use case để xây dựng được các đặc tả yêu cầu của từng usecase, đưa ra được các biểu đồ hoạt động, trình tự và biểu đồ lớp, mô hình hoạt động của hệ thống để từ đó xây dựng được cơ sở dữ liệu hợp lý và tối ưu. Đáp ứng được những tiêu chí: hoạt động được, dễ sử dụng, đầy đủ chức năng cần thiết như thêm sửa xóa dự án, thêm sửa xóa công việc, yêu cầu, thống kê tiến độ dự án bằng các biểu đồ tròn, biểu đồ cột hoặc file excel, ngoài ra người sử dụng có thể sửa đổi được hồ sơ, người quản trị có thể quản lý chức vụ, quản lý tài khoản trong hệ thống, quản lý loại công việc và loại giải pháp. Ngoài ra, giao diện cũng đáp ứng được các tiêu chí bắt mắt, dễ sử dụng, trực quan cho người dùng.

Do hạn chế về mặt và kiến thức nên hệ thống của em chưa hoàn thiện được các chức năng như quên mật khẩu, thông báo thời gian thực. Hệ thống mới chỉ chạy đúng, đầy đủ nghiệp vụ và thông luồng các chức năng chính ngoài ra những trường hợp ngoại lệ vẫn chưa được kiểm tra và kiểm thử sâu. Cấu trúc chương trình vẫn chưa tối ưu, chưa thể kiểm thử được hiệu năng về tốc độ xử lý, tốc độ truy cập và mức độ tiêu tốn RAM, CPU của hệ thống, giao diện vẫn chưa thể hiển thị đúng trên nền tảng di động.

Trong thời gian sắp tới, em mong nhận được sự giúp đỡ quý báu của thầy cô và bạn bè để em có thể hoàn thiện và bổ sung đồ án tốt hơn để xây dựng hoàn thiện các chức năng cụ thể như: Hoàn thiện và bổ sung thêm chức năng tìm kiếm nâng cao, thống kê theo lọc nâng cao, chức năng quên mật khẩu cho người dùng. Xây dựng cơ chế bảo mật và an toàn dữ liệu tốt hơn. Xây dựng chức năng thông báo thời gian thực để người sử dụng có thể nắm bắt được công việc trong dự án một cách dễ dàng hơn, mở rộng cả trên nền tảng

responsive. Bên cạnh đó triển khai mở rộng thêm đi thu thập ý kiến khảo sát thêm từ các doanh nghiệp lớn, vừa và nhỏ để hiểu được doanh nghiệp mong muốn gì. Phát triển và kiểm thử hệ thống trên tất cả các trình duyệt và phiên bản để đánh giá được tốc độ xử lý cũng như khả năng truy cập.

TÀI LIỆU THAM KHẢO

- [1]. ThS. Nguyễn Minh Hạo, ‘Giáo trình Lập trình Web với ASP.NET’, Nhà xuất bản: Đại học Quốc gia TP. HCM (01/2014)
- [2]. Thạc Bình Cường, ‘Phân tích và thiết kế hệ thống thông tin’, Nhà xuất bản khoa học và kỹ thuật, 2002
- [3]. Eric Freeman & Elisabeth Robson, “Head First Design Patterns: Một hướng dẫn “thân thiện”. 2004.
- [4]. Trịnh Thế Tiến, “Giáo trình hệ quản trị cơ sở dữ liệu SQL”, 2009.
- [5]. Nguyễn Tân Bình, “Quản lý dự án”, 2004.