

**BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC CẦN THƠ
KHOA CÔNG NGHỆ THÔNG TIN & TRUYỀN THÔNG**

**LUẬN VĂN TỐT NGHIỆP ĐẠI HỌC
NGÀNH KỸ THUẬT PHẦN MỀM**

Đề tài

**PHÁT TRIỂN ỨNG DỤNG ƯỚC LƯỢNG GIÁ
TRỊ PHẦN MỀM VÀ THU THẬP YÊU CẦU
NGƯỜI DÙNG**

**Sinh viên: Trần Hữu Tính
Mã số: 1111350
Khóa: K37**

Cần Thơ, 05/2015

**BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC CẦN THƠ
KHOA CÔNG NGHỆ THÔNG TIN & TRUYỀN THÔNG
BỘ MÔN CÔNG NGHỆ PHẦN MỀM**

**LUẬN VĂN TỐT NGHIỆP ĐẠI HỌC
NGÀNH KỸ THUẬT PHẦN MỀM**

Đề tài

**PHÁT TRIỂN ỨNG DỤNG ƯỚC LƯỢNG GIÁ
TRỊ PHẦN MỀM VÀ THU THẬP YÊU CẦU
NGƯỜI DÙNG**

**Người hướng dẫn
TS. Ngô Bá Hùng**

**Sinh viên thực hiện
Trần Hữu Tính
Mã số: 1111350
Khóa: K37**

Cần Thơ, 05/2015

LỜI CẢM ƠN

Gần 4 năm học tập tại trường Đại Học Cần Thơ, được sự truyền đạt tận tình của thầy cô cùng với sự giúp đỡ nhiệt tình của các bạn, hôm nay em đã hoàn thành được đề tài luận văn tốt nghiệp.

Em xin chân thành cảm ơn quý thầy cô Trường Đại Học Cần Thơ nói chung cũng như thầy cô Khoa Công Nghệ Thông Tin và Truyền Thông nói riêng đã tận tình giảng dạy và truyền đạt những kiến thức quý báu cho em trong suốt gần 4 năm qua. Đặc biệt là thầy **Ngô Bá Hùng** đã tận tình hướng dẫn em trong quá trình làm đề tài luận văn này.

Do kiến thức còn hạn hẹp, thời gian tìm hiểu chưa sâu, chắc chắn bài báo cáo luận văn này không tránh khỏi những thiếu sót, hạn chế, rất mong nhận được sự đóng góp của quý thầy cô để đề tài này ngày càng hoàn thiện hơn.

Em xin gửi lời chúc sức khỏe và lòng biết ơn sâu sắc đến quý thầy cô trường Đại Học Cần Thơ, thầy cô khoa Công Nghệ Thông Tin & Truyền Thông đã giúp đỡ chúng em trong những năm qua. Xin dành những tình cảm chân thành nhất cho gia đình và cho người thân.

Cuối cùng, xin gửi lời cảm ơn đến tất cả bạn bè đã gắn bó cùng chúng tôi.

Cần Thơ, tháng 05-2015

TRẦN HỮU TÍNH

MỤC LỤC

LỜI CẢM ƠN	i
NHẬN XÉT CỦA CÁN BỘ HƯỚNG DẪN.....	ii
NHẬN XÉT CỦA CÁN BỘ PHẢN BIỆN	iii
MỤC LỤC	iv
DANH MỤC BẢNG	v
DANH MỤC HÌNH ẢNH.....	vii
TÓM TẮT	viii
ABSTRACT	ix
PHẦN I - GIỚI THIỆU	1
ĐẶT VẤN ĐỀ	2
LỊCH SỬ GIẢI QUYẾT VẤN ĐỀ	3
MỤC TIÊU CỦA ĐỀ TÀI	5
ĐỐI TƯỢNG VÀ PHẠM VI NGHIÊN CỨU	6
NỘI DUNG NGHIÊN CỨU	7
NHỮNG ĐÓNG GÓP CHÍNH CỦA ĐỀ TÀI.....	10
BỐ CỤC LUẬN VĂN.....	11
PHẦN II - NỘI DUNG	12
CHƯƠNG 1 – MÔ TẢ BÀI TOÁN.....	13
1.1. Mô tả chi tiết bài toán	13
1.2. Phân tích đánh giá lựa chọn giải pháp có liên quan đến bài toán	22
CHƯƠNG 2 – THIẾT KẾ VÀ CÀI ĐẶT GIẢI PHÁP	26
2.1. Thiết kế kiến trúc	26
2.2. Thiết kế dữ liệu	27
2.3. Thiết kế chức năng.....	35
2.4. Cài đặt giải pháp	53
CHƯƠNG 3 - KIỂM THỬ VÀ ĐÁNH GIÁ	62
3.1. Mục tiêu kiểm thử	62
3.2. Kịch bản kiểm thử.....	62
3.3. Kết quả kiểm thử.....	64
PHẦN III - KẾT LUẬN	65
KẾT QUẢ ĐẠT ĐƯỢC	66
HƯỚNG PHÁT TRIỂN	68
TÀI LIỆU THAM KHẢO	69
HƯỚNG DẪN CÀI ĐẶT VÀ SỬ DỤNG	70

DANH MỤC BẢNG

Bảng 1. Bảng phân chia công việc theo module chức năng.	7
Bảng 2. Bảng phân chia công việc chi tiết cho từng thành viên.	8
Bảng 3. Bảng sắp xếp thứ tự ưu tiên các yêu cầu chức năng của phần mềm.	14
Bảng 4. Bảng chuyển đổi yêu cầu chức năng sang trường hợp sử dụng.	14
Bảng 5. Bảng tính toán điểm các tác nhân (actors).	15
Bảng 6. Bảng tính toán điểm các trường hợp sử dụng.	15
Bảng 7. Bảng tính toán hệ số phức tạp kỹ thuật-công nghệ.	16
Bảng 8. Bảng tính toán hệ số tác động môi trường, nhóm làm việc, hệ số phức tạp về môi trường.	17
Bảng 9. Bảng tính toán giá trị phần mềm.	18
Bảng 10. Bảng tổng hợp chi phí phần mềm.	19
Bảng 11. Giải thích các thực thể trên mô hình.	27
Bảng 12. Bảng ACTOR.	31
Bảng 13. Bảng BMT.	31
Bảng 14. Bảng DIAGRAM.	31
Bảng 15. Bảng DIAGRAMACTOR.	32
Bảng 16. Bảng DIAGRAMUSECASE.	32
Bảng 17. Bảng FILE.	32
Bảng 18. Bảng LOIACTOR.	33
Bảng 19. Bảng NHOMUC.	33
Bảng 20. Bảng PHANLOAI.	34
Bảng 21. Bảng PROJECT.	34
Bảng 22. Bảng UI.	34
Bảng 23. Bảng UIUSECASE.	35

Bảng 24. Bảng USECASE.	35
Bảng 25. Bảng mô tả thành phần giao diện chức năng tạo mới sơ đồ usecase.....	38
Bảng 26. Bảng mô tả thành phần giao diện chức năng xuất ảnh.	39
Bảng 27. Bảng mô tả danh sách sơ đồ.	41
Bảng 28. Mô tả thành phần đổi tên sơ đồ.	42
Bảng 29. Bảng mô tả thành phần giao diện chức năng chỉnh sửa tác nhân.....	44
Bảng 30. Mô tả thành phần giao diện chức năng chỉnh sửa thông tin usecase.....	47
Bảng 31. Mô tả thành phần giao diện chức năng tạo mới giao diện.....	49
Bảng 32. Bảng mô tả thành phần giao diện chức năng gán giao diện vào usecase. .	50
Bảng 33. Bảng mô tả thành phần giao diện chức năng liên kết các giao diện.....	51
Bảng 33. Bảng mô tả thành phần giao diện Screen Flow.	52
Bảng 33. Bảng mô tả thành phần giao diện Screen Flow.	53
Bảng 34. Bảng các đối tượng cơ bản trong JOINTJS.....	53

DANH MỤC HÌNH ẢNH

Hình 1. Sơ đồ usecase mô tả chức năng của ứng dụng.....	20
Hình 2. Sơ đồ usecase chức năng cần giải quyết.	21
Hình 3. Sơ đồ use-case chi tiết chức năng vẽ sơ đồ usecase.....	22
Hình 4. Sơ đồ usecase chi tiết chức năng vẽ giao diện.	22
Hình 5. Kiến trúc tổng quan của hệ thống theo MVC.	26
Hình 6. Mô hình dữ liệu mức quan niệm.	27
Hình 7. Giao diện chức năng đăng nhập.....	36
Hình 8. Giao diện chức năng cập nhật thông tin cá nhân.	37
Hình 9. Giao diện chức năng tạo mới sơ đồ usecase.	38
Hình 10. Giao diện chức năng xuất ảnh.....	39
Hình 11. Giao diện danh sách sơ đồ usecase.	41
Hình 12. Giao diện đổi tên sơ đồ.	42
Hình 13. Giao diện chức năng chỉnh sửa thông tin tác nhân.	44
Hình 14. Giao diện chức năng chỉnh sửa thông tin use-case.	47
Hình 15. Giao diện chức năng tạo mới giao diện.	49
Hình 16. Giao diện gán giao diện vào usecase.	50
Hình 17. Giao diện chức năng liên kết các giao diện.	51
Hình 17. Giao diện Screen Flow.....	52
Hình 17. Giao diện chức năng gợi ý usecase theo chức năng đã có.....	53
Hình 18. Sơ đồ truyền nhận dữ liệu AJAX.....	57
Hình 19. Class Actor sử dụng hibernate.	59

TÓM TẮT

Ngày nay, công nghệ thông tin phát triển mạnh mẽ trên toàn thế giới. Nó cần thiết trong hầu hết các lĩnh vực cũng như trong cuộc sống của con người. Trên toàn thế giới nói chung và Việt Nam nói riêng, có rất nhiều công ty phần mềm được thành lập, kéo theo sau đó là hàng chục, hàng trăm, thậm chí là hàng ngàn, hàng triệu ứng dụng, phần mềm, website ra đời trên nhiều loại thiết bị khác nhau. Các ứng dụng, phần mềm, website là các sản phẩm tinh xảo do con người tạo ra. Giống như các sản phẩm khác nó cũng có giá trị và việc xác định giá trị của nó vô cùng quan trọng. Hiện tại ở Việt Nam, việc xác định giá trị phần mềm dựa trên tiêu chuẩn và quy định của công văn 2589/BTTTT-UDCNTT (ban hành ngày 24/08/2011).

Theo công văn, để xác định được giá trị của phần mềm cần rất nhiều thông tin. Để có những thông tin đó phải qua các quá trình từ thu thập yêu cầu người dùng, đến phân tích dữ liệu thu thập, sau đó đưa ra các biểu đồ theo ngôn ngữ mô hình hóa thống nhất (UML) và dựa trên các biểu đồ đó để lập công thức tính toán và kết hợp với các thông số khác để đưa ra chính xác giá trị của phần mềm.

Hiện tại chưa có công cụ, ứng dụng hay phần mềm nào xây dựng để ước lượng chi phí và thu thập phần mềm dựa theo công văn 2589/BTTTT-UDCNTT. Có nhiều công ty lớn, vừa và nhỏ cũng như các nhóm lập trình viên, các cá nhân riêng rẽ đã có khá nhiều cách thức để thực hiện toàn bộ quá trình trên. Nhưng hiện tại những cách thức đó còn rời rạc, thủ công và thiếu hiệu quả.

Để giải quyết vấn đề trên, đề tài này được xây dựng để phát triển một ứng dụng ước lượng chi phí và thu thập phần mềm dựa trên công văn 2589. Ứng dụng được xây dựng với các tính năng như: thu thập yêu cầu, mô tả yêu cầu chức năng - phi chức năng, thiết kế sơ đồ trường hợp sử dụng (usecase), ước lượng chi phí, thiết kế giao diện. Ứng dụng với các tính năng để sử dụng, giao diện thân thiện, xây dựng theo tiêu chuẩn, quy định của công văn 2589/BTTTT-UDCNTT, hỗ trợ người dùng xác định giá trị phần mềm một cách tiện lợi và hiệu quả nhất.

ABSTRACT

Nowadays, Information technology was strong developed in the world. It necessary in many fields and human life. In the world and Viet Nam, There are many companies was build, after that, there are dozens, hunreds, even thousands, milions application, software, website was born on many others device. They are delicate products was created by people. As well as other products, they comes with price and It is very important. Currently, Viet Nam determined value of software based on official dispatch no 2589/BTTTT-UDCNTT of August 24th 2011.

Follow official dispatch, to determined value of software must many information. To get them, we have to perform many process. From requirement collection to requirement analysic, after that give out the diagrams (UML) and base on them to implement expression and combine with other parameters to calculate exactly value of software.

At the present, Not have tool, application or software for determine value of software base on official dispatch no 2589. There are many big, medium and small companies and team, personal. They haved many methods to do all of above process. But they are disjointed, handiwork and inefficient.

To sovled these problems, this topic is built to determine value of software base on official dispatch no 2589. Appication covers features such as: Collecting requirement, descripting requirement function and requirement non-function, design usecase diagrams, estimating value of software, designing interface. With easy to use of features, friendly interface, building standards and regulations of official dispatch no 2589/BTTTT-UDCNTT. This application supports to determined value of software most convenient and most efficient.

PHẦN I - GIỚI THIỆU

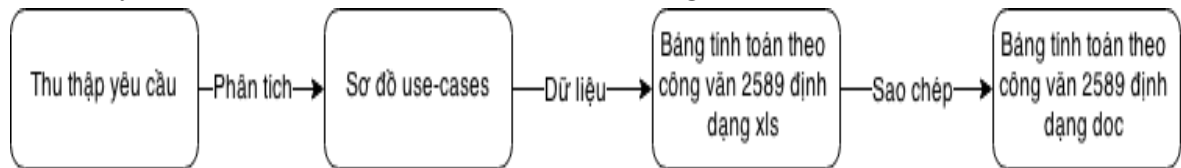
ĐẶT VẤN ĐỀ

Ở Việt Nam việc ước lượng chi phí được thực hiện dựa trên công văn 2589/BTTTT-UĐCNTT. Theo công văn để ước lượng chi phí cần chuẩn bị nhiều số liệu cần thiết về yêu cầu chức năng, sơ đồ usecase, kỹ thuật công nghệ, môi trường... Các số liệu này muốn có được phải thông qua từ nhiều quá trình khác nhau. Đầu tiên phải thông qua quá trình thu thập để có được yêu cầu chức năng, sau đó vẽ sơ đồ usecase. Tiếp theo là đưa các usecase vào file excel và cài đặt công thức tính toán. Các quá trình trên được thực hiện riêng rẽ, cần nhiều công cụ hỗ trợ và các công cụ này cần phải cài đặt để có thể sử dụng, một số công cụ còn tính phí. Việc ước lượng giá trị phần mềm và thu thập yêu cầu người dùng hết sức quan trọng nhưng hiện tại các công việc này chưa được thống nhất thành một hệ thống hoàn chỉnh, chưa có sự tương tác hay liên kết chặt chẽ, chính xác giữa các công việc. Ngoài ra, các công việc còn tổ chức, lưu trữ rời rạc, cần nhiều công sức và chi phí để thực hiện.

Tận dụng và nắm bắt xu hướng công nghệ và cách tiếp cận hiện đại: công nghệ và tương tác môi trường web. Đề tài này tập trung nghiên cứu và phát triển ứng dụng ước lượng giá trị phần mềm và thu thập yêu cầu người dùng trên nền web dựa trên công văn 2589/BTTTT-UĐCNTT, hỗ trợ cho mọi người có thể truy cập mọi lúc mọi nơi, lưu trữ tập trung, các công việc vẽ sơ đồ, thu thập yêu cầu, ước lượng chi phí ... được liên kết một cách chặt chẽ và không cần cài đặt nhiều công cụ để thực hiện. Người sử dụng có thể ước lượng chi phí một cách chính xác, dễ dàng hơn.

LỊCH SỬ GIẢI QUYẾT VẤN ĐỀ

Hiện tại ở Việt Nam nói chung và trong khoa nói riêng, chưa có phần mềm nào hỗ trợ việc thu thập yêu cầu người dùng và xác định chi phí phần mềm dựa theo công văn 2589/BTTTT-UĐCNTT. Hiện tại, việc thu thập thông tin hầu hết sử dụng công cụ soạn thảo văn bản để ghi lại. Sau đó sử dụng các công cụ hỗ trợ để vẽ biểu đồ use-cases. Từ biểu đồ usecase họ đưa thông tin vào tập tin “.xls” đã cài đặt các công thức tính toán theo công văn để có được các bảng tính toán giá trị phần mềm. Cuối cùng là phải gom tất cả các bảng tính toán qua tập tin “.doc” để lưu trữ, báo cáo. Quy trình thực hiện có thể được tóm tắt bằng sơ đồ sau:



Hình 1. Sơ đồ thực hiện xác định giá trị của phần mềm

- **Thu thập yêu cầu:** Có nhiều loại phương pháp như phỏng vấn, bảng câu hỏi và điều tra, quan sát thực tế. Công cụ sử dụng: giấy viết, các công cụ soạn thảo văn bản như Microsoft Word. Lưu trữ: trên giấy, trong máy tính, máy tính bảng, điện thoại thông minh, usb, trên mây ...
- **Sơ đồ use-cases:** theo ngôn ngữ mô hình hóa thống nhất (UML). Lưu trữ: trên giấy, trong máy tính cá nhân, máy tính bảng, điện thoại thông minh, usb, trên mây ... Công cụ sử dụng:
 - **Rational Rose:** là phần mềm hỗ trợ phân tích, thiết kế hệ thống phần mềm. Nó cho phép ta vẽ nhiều loại mô hình, sơ đồ khác nhau theo phương pháp hướng đối tượng. Là phần mềm có tính phí của hãng IBM.
 - **Astah:** là phần mềm cho phép vẽ các sơ đồ và mô hình khác nhau. Có hai phiên bản để sử dụng: phiên bản “community” không tính phí nhưng các chức năng sử dụng bị hạn chế rất nhiều, phiên bản “Professional” đầy đủ các chức năng nhưng tính phí.
 - **Draw.io:** là website hỗ trợ vẽ nhiều loại sơ đồ, mô hình khác nhau. Không tính phí.
- Bảng tính toán theo công văn 2589 dưới dạng tập tin xls: cài đặt sẵn các công thức tính toán về tác nhân, yêu cầu chức năng, môi trường, ... Lưu trữ: trong máy tính cá nhân, máy tính bảng, điện thoại thông minh, usb, trên mây ... Công cụ sử dụng: phổ biến là Microsoft Excel.
- Bảng tính toán theo công văn 2589 dưới dạng tập tin doc: thông tin được lấy từ bảng tính toán dưới dạng xls. Lưu trữ: trong máy tính cá nhân, máy tính

bảng, điện thoại thông minh, usb, trên mây ... Công cụ sử dụng: phổ biến là Microsoft Word.

➔ Sử dụng nhiều công cụ hỗ trợ riêng rẽ, lưu trữ rời rạc tốn nhiều thời gian công sức.

MỤC TIÊU CỦA ĐỀ TÀI

Dựa trên công văn 2589/BTTTT-UĐCNTT và các kiến thức đã học được để xây dựng phần mềm ước lượng giá trị phần mềm và thu thập yêu cầu người dùng với đầy đủ các tính năng và có thể áp dụng tại trung tâm điện tử tin học Trường Đại Học Cần Thơ. Các tính năng cần đạt được:

- Cho phép người dùng tạo lập, chỉnh sửa, xóa dự án.
- Cho phép người dùng mô tả yêu cầu chức năng và phi chức năng của dự án.
- Cho phép người dùng vẽ sơ đồ use-cases, thu thập yêu cầu trên từng use-cases, dựa trên dựa trên sơ đồ use-case để ước lượng giá trị phần mềm.
- Cho phép người dùng thiết kế giao diện, liên kết giữa các giao diện, liên kết giao diện đến use-cases, xuất ảnh giao diện.
- Cho phép người dùng xuất các bảng tính toán theo công văn.

ĐỐI TƯỢNG VÀ PHẠM VI NGHIÊN CỨU

Đề tài sẽ tập trung nghiên cứu các lý thuyết, kỹ thuật để có thể xây dựng được ứng dụng ước lượng chi phí và thu thập yêu cầu người dùng dựa trên công văn 2589/BTTTT-UĐCNTT:

- Về lý thuyết:

- Tìm hiểu nội dung, công thức, các bảng tính toán, quy trình thực hiện của việc xác định giá trị phần mềm theo công văn 2589/BTTTT-UĐCNTT.
- Tìm hiểu cách thức xây dựng một ứng dụng trên nền web với ngôn ngữ Java.
- Tìm hiểu thu thập yêu cầu dựa trên phương pháp phỏng vấn.
- Tìm hiểu biểu đồ use-cases theo ngôn ngữ mô hình hóa thống nhất (UML).

- Về kỹ thuật:

- Nghiên cứu về mô hình lập trình: Spring Web MVC Framework.
- Nghiên cứu framework Bootstrap 3 hỗ trợ thiết kế layout, các thành phần web nhanh chóng, tiện lợi.
- Nghiên cứu công nghệ AJAX hỗ trợ cho việc vẽ, các xử lý gửi và nhận dữ liệu từ máy khách đến máy chủ không bị nạp lại trang.
- Nghiên cứu thư viện Jointjs viết bằng javascript phục vụ cho việc vẽ sơ đồ và giao diện.
- Tìm hiểu các thư viện hỗ trợ việc xuất tập tin.

- Về ứng dụng:

- Nghiên cứu giải quyết bài toán ước lượng giá trị phần mềm và thu thập yêu cầu người dùng dựa trên công văn 2589/BTTTT-UĐCNTT có thể áp dụng vào các đề tài của trung tâm tin học Trường Đại Học Cần Thơ.

NỘI DUNG NGHIÊN CỨU

- Phát triển ứng dụng ước lượng chi phí phần mềm và thu thập yêu cầu người dùng là đề tài nhóm, công việc được phân chia theo module chức năng:

Bảng 1. Bảng phân chia công việc theo module chức năng.

Thành viên	Module
Trần Hữu Tính	<ul style="list-style-type: none"> - Đăng nhập. - Cập nhật thông tin cá nhân. - Vẽ sơ đồ usecase. - Vẽ giao diện.
Ngô Minh Phương	<ul style="list-style-type: none"> - Đăng ký. - Đổi mật khẩu. - Thu thập yêu cầu. - Cập nhật dự án. - Mô tả yêu cầu chức năng. - Mô tả yêu cầu phi chức năng. - Ước lượng chi phí. - Xuất tập tin.

- Địa điểm nghiên cứu: Phòng nghiên cứu, khoa Công Nghệ Thông Tin và Truyền Thông, trường Đại Học Cần Thơ.
- Cách thức liên lạc trong khi nghiên cứu: Họp tại địa điểm nghiên cứu, qua Facebook, Gmail, điện thoại.
- Lịch biểu nghiên cứu của nhóm:
 - Tuần 0 và tuần 1 (22/12/2014-02/01/2015): liên hệ giảng viên để nhận đề tài. Công việc nhóm bắt đầu thực hiện từ tuần 2 (từ 05/01/2015).
 - Tuần 2: Tìm hiểu, phân tích, tìm kiếm các vấn đề, công cụ, tài liệu, giải pháp liên quan đến đề tài.
 - Tuần 3: Lựa chọn công cụ, giải pháp giải quyết đề tài. Tiến hành cài đặt các công cụ, thư viện cần thiết.
 - Tuần 4: Phân tích và thiết kế các chức năng của đề tài.
 - Tuần 5 – 16: Tiến hành lập trình.

- Tuần 17: Tiến hành kiểm thử và chỉnh sửa lỗi phần mềm.
- Tuần 18: Viết báo cáo.

Bảng 2. Bảng phân chia công việc chi tiết cho từng thành viên.

Thành viên thực hiện	Công việc
Trần Hữu Tính	Tìm hiểu: <ul style="list-style-type: none"> - Công văn 2589/BTTTT-U'ĐCNTT. - Framework Spring Web MVC + Hibernate. - Framework Bootstrap 3. - Công nghệ Ajax. - Thư viện vẽ Jointjs.
	Cài đặt các phần mềm cần thiết như: <ul style="list-style-type: none"> - Github. - Rational Rose. - Astah. - Spring Tool Suite. - XAMPP. - Browser: Firefox, Google Chrome ...
	Phân tích, thiết kế chức năng: <ul style="list-style-type: none"> - Đăng nhập. - Cập nhật thông tin cá nhân. - Vẽ giao diện. - Vẽ sơ đồ.
	Lập trình cài đặt chức năng đã phân chia.
	Kiểm thử chức năng đã phân chia.
	Viết báo cáo về chức năng đã phân chia.
Ngô Minh Phương	Tìm hiểu: <ul style="list-style-type: none"> - Công văn 2589/BTTTT-U'ĐCNTT. - Framework Spring Web MVC + Hibernate. - Framework Bootstrap 3. - Spring Sercurity. - Thư viện hỗ trợ xuất tập tin.
	Cài đặt các phần mềm cần thiết như: <ul style="list-style-type: none"> - Github. - Rational Rose.

	<ul style="list-style-type: none"> - Astah. - Spring Tool Suite. - XAMPP. - Browser: Firefox, Google Chrome ...
	Phân tích, thiết kế chức năng: <ul style="list-style-type: none"> - Đăng ký. - Đổi mật khẩu. - Thu thập yêu cầu. - Cập nhật dự án. - Mô tả yêu cầu chức năng. - Mô tả yêu cầu phi chức năng. - Ước lượng chi phí. - Xuất tập tin.
	Lập trình cài đặt chức năng đã phân chia.
	Kiểm thử chức năng đã phân chia.
	Viết báo cáo về chức năng đã phân chia.

NHỮNG ĐÓNG GÓP CHÍNH CỦA ĐỀ TÀI

- Kết hợp các quá trình từ thu thập yêu cầu đến phân tích vẽ sơ đồ hay xác định giá trị phần mềm. Giúp người dùng thực hiện các công việc tiện lợi, dễ dàng và hiệu quả hơn mà không cần cài đặt nhiều công cụ.
- Hỗ trợ người dùng lưu trữ trực tuyến các thông tin, dữ liệu, tài liệu thu thập yêu cầu cũng như các bảng tính toán cần thiết để xác định giá trị phần mềm theo công văn.
- Người dùng có thể dễ dàng tính toán giá trị phần mềm dựa trên các use-cases đã vẽ mà không cần cài đặt các công thức phức tạp trong công văn.
- Người dùng có thể dễ dàng thiết kế các giao diện tương ứng với những yêu cầu chức năng để so sánh với yêu cầu của khách hàng, so sánh xem có đúng với mong muốn của họ không.
- Hỗ trợ người dùng xuất các tài liệu, bảng tính toán chi phí cần thiết liên quan đến xác định giá trị phần mềm.
- Các chức năng, giao diện được xây dựng trong phần mềm thân thiện và dễ dàng sử dụng.

BỐ CỤC LUẬN VĂN

Bố cục quyền luận văn gồm 5 phần:

- Phần giới thiệu gồm có:
 - Đặt vấn đề: đưa ra bài toán, vấn đề đặt ra đối với đề tài. Lý do làm đề tài.
 - Lịch sử giải quyết vấn đề: đưa ra các nghiên cứu trong nước và ngoài nước về vấn đề đặt ra và những vấn đề còn chưa giải quyết được.
 - Mục tiêu của đề tài: đưa ra mục tiêu chính cần đạt được của đề tài.
 - Đối tượng và phạm vi nghiên cứu: trình bày các lý thuyết, kỹ thuật và ứng dụng của đề tài.
 - Nội dung nghiên cứu: trình bày các công việc, lịch biểu của nhóm.
 - Những đóng góp chính của đề tài: kết quả chính của đề tài đạt được là gì.
 - Bố cục của quyền luận văn: đưa ra bố cục nội dung của đề tài và giới thiệu sơ lược về chúng.
- Phần nội dung gồm có:
 - Chương 1 - Mô tả bài toán: trình bày chi tiết về bài toán mà đề tài giải quyết, đưa ra các phân tích đánh giá và giải pháp và chọn lựa giải pháp phù hợp.
 - Chương 2 - Thiết kế và cài đặt giải pháp giải pháp: đưa ra các kiến trúc hệ thống, thiết kế chức năng, thiết kế dữ liệu và trình bày cách thức cài đặt các thiết kế giải pháp đó.
 - Chương 3 - Kiểm thử và đánh giá: đưa ra mục tiêu, kịch bản và kết quả của kiểm thử.
- Phần kết luận gồm có:
 - Kết quả đạt được: trình bày những thành quả thu được về kiến thức, kỹ năng, ứng dụng của phần mềm và những hạn chế phần mềm cần khắc phục.
 - Hướng phát triển: trình bày hướng phát triển, cải thiện lại đề tài.
- Tài liệu tham khảo.
- Hướng dẫn cài đặt và sử dụng.

PHẦN II - NỘI DUNG

CHƯƠNG 1 – MÔ TẢ BÀI TOÁN

Đưa ra mô tả chi tiết bài toán, phân tích đánh giá các giải pháp và tiếp cận vấn đề và lựa chọn giải pháp giải quyết.

1.1. Mô tả chi tiết bài toán

Bài toán đặt ra là phát triển ứng dụng ước lượng chi phí phần mềm và thu thập yêu cầu người dùng dựa trên công văn 2589/BTTTT-UĐCNTT.

1.1.1. Bài toán công văn 2589/BTTTT-UĐCNTT

- Một số khái niệm về công văn:
 - Công văn 2589/BTTTT-UĐCNTT là công văn được ban hành ngày 24/08/2011 của Bộ Thông Tin và Truyền Thông dùng để hướng dẫn xác định chi phí phát triển và nâng cấp phần mềm nội bộ.
 - Trường hợp sử dụng (use case) là một tập hợp các giao dịch giữa hệ thống phần mềm với các tác nhân bên ngoài hệ thống nhằm đạt được một mục tiêu sử dụng nào đó của tác nhân. Một trường hợp sử dụng mô tả một hoặc nhiều tình huống sử dụng xảy ra khi tác nhân tương tác với hệ thống phần mềm.
 - Giao dịch (transaction) là một chuỗi các hành động có tính chất tương tác giữa tác nhân và hệ thống phần mềm. Khởi đầu của chuỗi hành động này là một hành động từ tác nhân tới hệ thống. Kết thúc của chuỗi hành động này là một hành động ngược trở lại của hệ thống lên tác nhân.
 - Biểu đồ trường hợp sử dụng (use case diagram) dùng mô tả các tác nhân và kết nối giữa tác nhân với các trường hợp sử dụng nhằm miêu tả chức năng mà phần mềm cung cấp.
 - Tác nhân (actor) là người hay hệ thống bên ngoài tương tác, trao đổi thông tin với phần mềm.
- Hồ sơ phục vụ xác định chi phí phần mềm sẽ gồm có:
 - Tên phần mềm.
 - Các qui trình nghiệp vụ cần được tin học hóa.
 - Các tác nhân tham gia vào qui trình nghiệp vụ và mối liên hệ giữa các tác nhân.

- Danh sách các yêu cầu chức năng và yêu cầu phi chức năng của phần mềm.
 - Bảng sắp xếp thứ tự ưu tiên các yêu cầu chức năng của phần mềm.
 - Bảng chuyển đổi yêu cầu chức năng sang trường hợp sử dụng (phân theo 3 loại: B, M, T).
 - Biểu đồ về các trường hợp sử dụng lập theo ngôn ngữ Mô hình hóa thống nhất (UML) trên cơ sở nhóm các chức năng từ Bảng sắp xếp thứ tự ưu tiên các yêu cầu chức năng của phần mềm.
 - Các yêu cầu về năng lực của cán bộ tham gia phát triển, nâng cấp, chỉnh sửa phần mềm.
 - Các yêu cầu về môi trường cho phát triển, nâng cấp phần mềm.
 - Yêu cầu về độ phức tạp kỹ thuật-công nghệ của phần mềm.
- Trình tự xác định chi phí phần mềm theo công văn:
- Nghiên cứu, kiểm tra nắm vững các thông tin cần có trong Hồ sơ phục vụ xác định giá trị phần mềm. Trường hợp cần thiết yêu cầu giải thích rõ các vấn đề có liên quan đến nội dung mô tả yêu cầu kỹ thuật của phần mềm đã nêu trong Hồ sơ.
 - Lập Bảng sắp xếp thứ tự ưu tiên các yêu cầu chức năng của phần mềm.

Bảng 3. Bảng sắp xếp thứ tự ưu tiên các yêu cầu chức năng của phần mềm.

TT	Mô tả yêu cầu	Phân loại	Ghi chú
1			
2			
...			

- Lập Bảng chuyển đổi yêu cầu chức năng sang trường hợp sử dụng.

Bảng 4. Bảng chuyển đổi yêu cầu chức năng sang trường hợp sử dụng.

TT	Tên Use-case	Tên tác	Tên tác	Mô tả	Mức độ
1					
2					
...					

- Lập Bảng tính toán điểm các tác nhân (actors) tương tác, trao đổi thông tin với phần mềm. Bảng tính toán này phải phù hợp với Biểu

đồ trường hợp sử dụng, thể hiện được đầy đủ loại, số lượng tác nhân và bản chất tương tác của các tác nhân.

Bảng 5. Bảng tính toán điểm các tác nhân (actors).

TT	Loại Actor	Mô tả	Số tác nhân	Điểm của từng loại tác nhân	Ghi chú
1	Đơn giản	Thuộc loại giao diện của chương trình			
2	Trung bình	Giao diện tương tác hoặc phục vụ một giao thức			
3	Phức tạp	Giao diện đồ họa			
	Cộng (1+2+3)	TAW			

- Thực hiện đếm các trường hợp sử dụng (use-case) và lập Bảng tính toán điểm các trường hợp sử dụng. Bảng tính toán này cần lập theo thứ tự triển khai phát triển hoặc nâng cấp phần mềm.

Bảng 6. Bảng tính toán điểm các trường hợp sử dụng.

STT	Loại	Số trường hợp sử dụng	Điểm của từng loại trường hợp sử dụng
1	B		
	Đơn giản		
	Trung bình		
	Phức tạp		
2	M		
	Đơn giản		
	Trung bình		
	Phức tạp		

3	T		
	Đơn giản		
	Trung bình		
	Phức tạp		
	Cộng 1+2+3	TBF	

- Lập Bảng tính toán hệ số phức tạp kỹ thuật-công nghệ. Bảng tính toán này phải phù hợp với nội dung yêu cầu chức năng và phi chức năng của phần mềm đã được nêu trong Hồ sơ.

Bảng 7. Bảng tính toán hệ số phức tạp kỹ thuật-công nghệ.

TT	Các hệ số	Trọng số	Giá trị xếp hạng	Kết quả	Ghi chú
I	Hệ số KT-CN (TFW)				
1	Hệ thống phân tán	2			
2	Tính chất đáp ứng tức thời hoặc yêu cầu đảm bảo thông lượng	1			
3	Hiệu quả sử dụng trực tuyến	1			
4	Độ phức tạp của xử lý bên trong	1			
5	Mã nguồn phải tái sử dụng được	1			
6	Dễ cài đặt	0,5			
7	Dễ sử dụng	0,5			
8	Khả năng chuyển đổi	2			
9	Khả năng dễ thay đổi	1			
10	Sử dụng đồng thời	1			
11	Có các tính năng bảo mật đặc biệt	1			

12	Cung cấp truy nhập trực tiếp tới các phần mềm của các hãng thứ ba	1			
13	Yêu cầu phương tiện đào tạo đặc biệt cho người sử dụng	1			
II	Hệ số phức tạp về KT-CN (TCF)				

- Lập Bảng tính toán hệ số tác động môi trường, nhóm làm việc, hệ số phức tạp về môi trường. Bảng tính toán này phải phù hợp với các yêu cầu về năng lực của cán bộ tham gia phát triển, nâng cấp phần mềm.
- Nội suy đánh giá kinh nghiệm, nội suy thời gian lao động trên cơ sở Bảng tính hệ số tác động môi trường.

Bảng 8. Bảng tính toán hệ số tác động môi trường, nhóm làm việc, hệ số phức tạp về môi trường.

TT	Các hệ số tác động môi trường	Trọng số	Giá trị xếp hạng	Kết quả	Độ ổn định kinh nghiệm
I	Hệ số tác động môi trường và nhóm làm việc (EFW)				
	Đánh giá cho từng thành viên				
1	Có áp dụng qui trình phát triển phần mềm theo mẫu RUP và có hiểu biết về RUP hoặc quy trình phát triển phần mềm tương đương	1,5			
2	Có kinh nghiệm về ứng dụng tương tự	0,5			
3	Có kinh nghiệm về hướng đối tượng	1			
4	Có khả năng lãnh đạo Nhóm	0,5			
5	Tính chất năng động	1			
	Đánh giá chung cho Dự án				
6	Độ ổn định của các yêu cầu	2			

TT	Các hệ số tác động môi trường	Trọng số	Giá trị xếp hạng	Kết quả	Độ ổn định kinh nghiệm
7	Sử dụng các nhân viên làm bán thời gian	-1			
8	Dùng ngôn ngữ lập trình loại khó	-1			
II	Hệ số phức tạp về môi trường (EF)				
III	Độ ổn định kinh nghiệm (ES)				
IV	Nội suy thời gian lao động (P)				

- Xác định mức lương lao động bình quân đối với việc phát triển hoặc nâng cấp phần mềm.
- Tập hợp giá trị đã tính toán ở các bước trên vào Bảng tính toán giá trị phần mềm theo các trường hợp: phát triển hoặc nâng cấp phần mềm. Giá trị sau khi tính toán được xử lý theo nguyên tắc làm tròn các trị số.

Bảng 9. Bảng tính toán giá trị phần mềm.

TT	Hạng mục	Diễn giải	Giá trị	Ghi chú
I	Tính điểm trường hợp sử dụng (Use-case)			
1	Điểm Actor (TAW)	Phụ lục III		
2	Điểm Use-case (TBF)	Phụ lục IV		
3	Tính điểm UUCP	$UUCP = TAW + TBF$		
4	Hệ số phức tạp về KT-CN (TCF)	$TCF = 0,6 + (0,01 \times TFW)$		
5	Hệ số phức tạp về môi trường (EF)	$EF = 1,4 + (-0,03 \times EFW)$		
6	Tính điểm AUCP	$AUCP = UUCP \times TCF \times EF$		
II	Nội suy thời gian lao động (P)	$P : \text{người/giờ/AUCP}$		
III	Giá trị nỗ lực thực tế (E)	$E = 10/6 \times AUCP$		

IV	Mức lương lao động bình quân (H)	H: người/giờ		
V	Giá trị phần mềm nội bộ (G)	$G = 1,4 \times E \times P \times H$		

- Lập Bảng tổng hợp chi phí phần mềm.

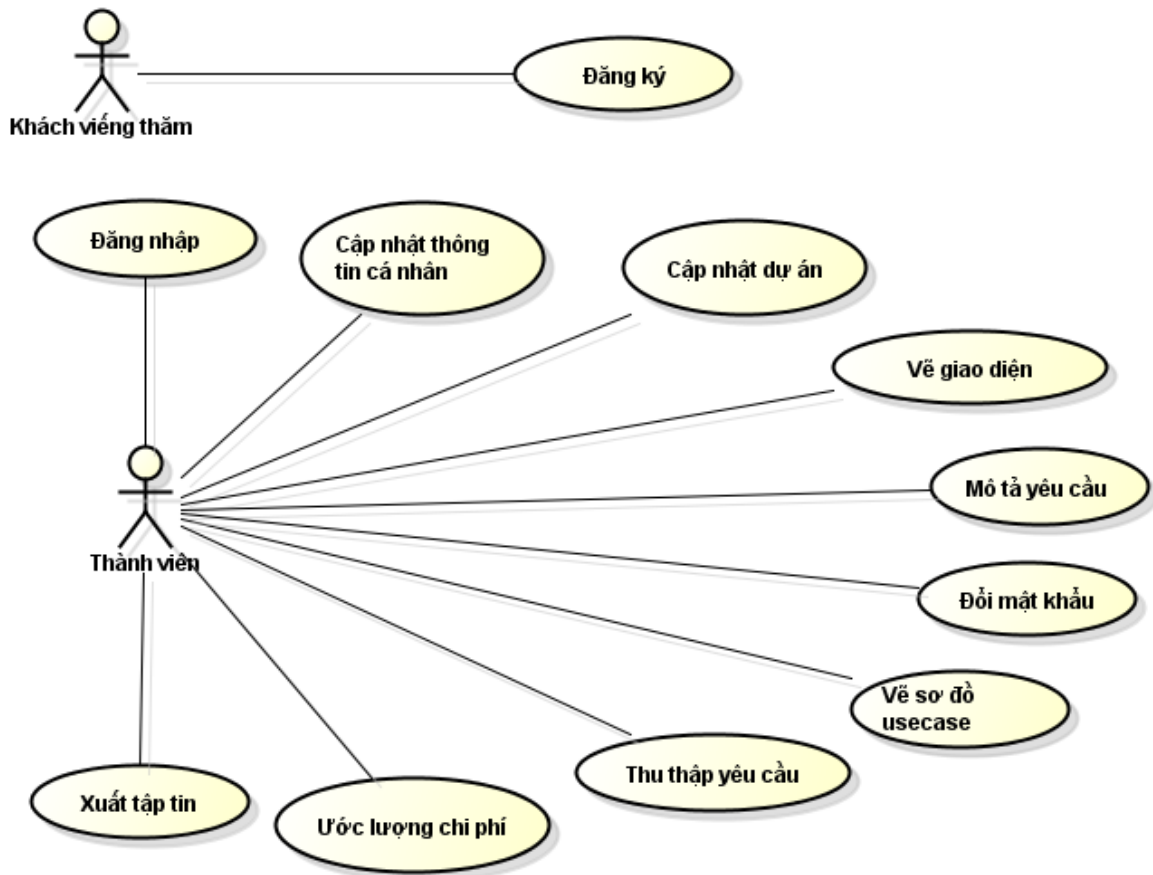
Bảng 10. Bảng tổng hợp chi phí phần mềm.

TT	Khoản mục chi phí	Cách tính	Giá trị	Ký hiệu
1	Giá trị phần mềm	$1,4 \times E \times P \times H$		G
2	Chi phí chung	$G \times \text{tỷ lệ}$		C
3	Thu nhập chịu thuế tính trước	$(G+C) \times \text{tỷ lệ}$		TL
4	Chi phí phần mềm	$G + C + TL$		G_{PM}
	TỔNG CỘNG	G_{PM}		

1.1.2. Mô tả chức năng tổng quát

- Các chức năng tổng quát được mô tả bằng sơ đồ usecase như sau:

Hình 1. Sơ đồ usecase mô tả chức năng của ứng dụng.



- Mô tả về chức năng trong sơ đồ:

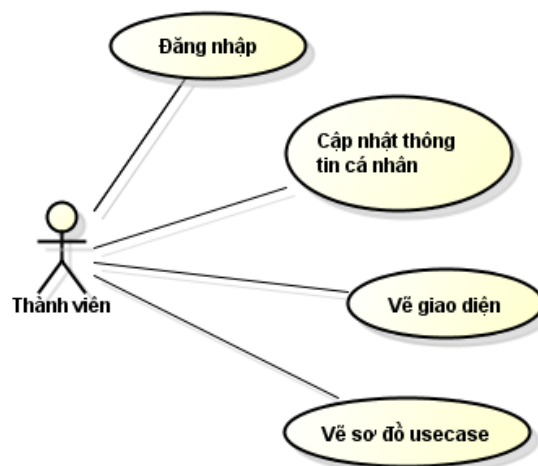
Chức năng	Mô tả
Đăng ký	Cho phép người dùng đăng ký tài khoản để đăng nhập vào hệ thống, sau khi đăng ký cần kích hoạt tài khoản để sử dụng.
Đăng nhập	Cho phép người dùng sử dụng tài khoản đăng ký đã kích hoạt thành công để đăng nhập vào hệ thống.
Cập nhật thông tin cá nhân	Cho phép người dùng cập nhật lại thông tin cá nhân: tên, số điện thoại, địa chỉ, email ...
Đổi mật khẩu	Cho phép người dùng đổi mật khẩu.
Cập nhật dự án	Cho phép người dùng tạo, sửa, xóa dự án.
Mô tả yêu cầu	Cho phép người dùng mô tả các yêu cầu chức năng và yêu cầu phi chức năng của một dự án.
Vẽ sơ đồ usecase	Cho phép người dùng vẽ sơ đồ usecase.
Thu thập yêu cầu	Cho phép người dùng thu thập yêu cầu trên các usecase đã vẽ.

Ước lượng chi phí	Cho phép người dùng ước lượng chi phí dựa trên sơ đồ usecase đã vẽ.
Vẽ giao diện	Cho phép người dùng vẽ giao diện.
Xuất tập tin	Cho phép người dùng xuất các loại tập tin khác nhau như excel, pdf...

1.1.3. Chức năng cần giải quyết

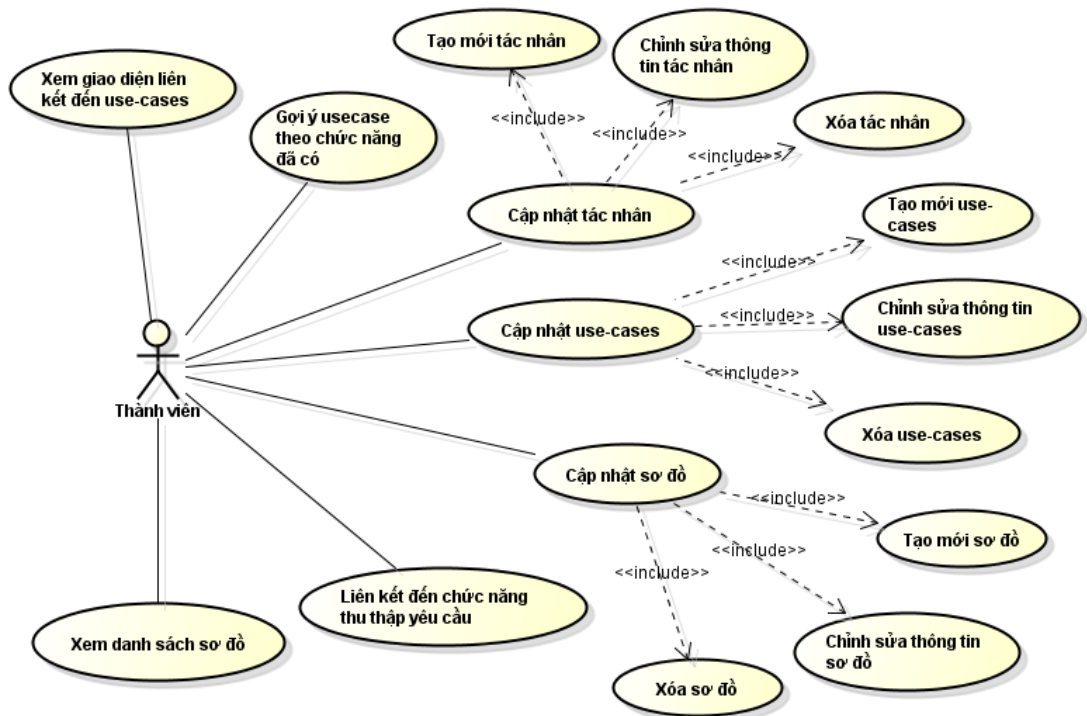
- Vì đây là đề tài nhóm nên không giải quyết toàn bộ chức năng mà chỉ thiết kế và cài đặt các chức năng được phân chia:

Hình 2. Sơ đồ usecase chức năng cần giải quyết.



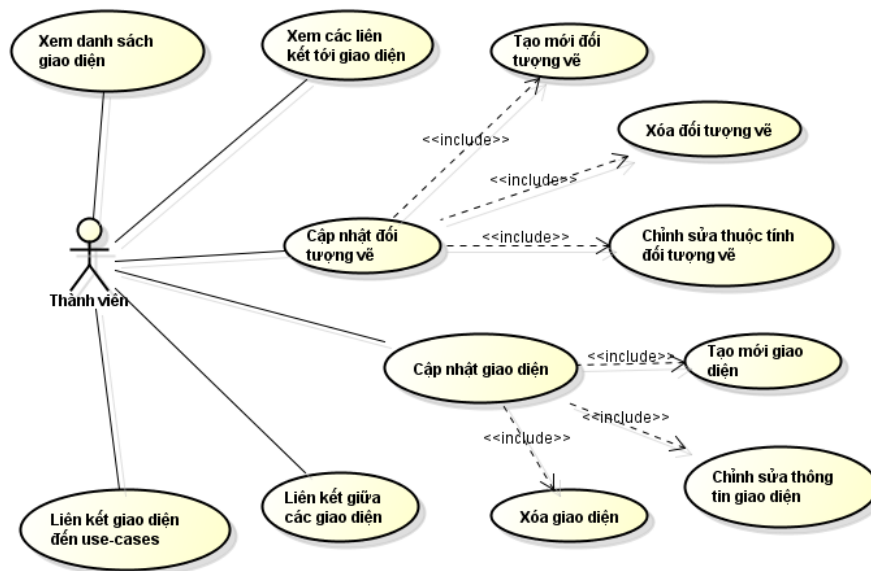
- Sơ đồ usecase chi tiết cho chức năng vẽ sơ đồ usecase:

Hình 3. Sơ đồ use-case chi tiết chức năng vẽ sơ đồ usecase.



- Sơ đồ usecase chi tiết cho chức năng vẽ giao diện:

Hình 4. Sơ đồ usecase chi tiết chức năng vẽ giao diện.



1.2. Phân tích đánh giá lựa chọn giải pháp có liên quan đến bài toán

1.2.1. Về nền của ứng dụng

- Ứng dụng được xây dựng trên nền web không phải trên nền desktop vì lý do:

- Có thể truy cập mọi lúc mọi nơi.
- Cùng một thời điểm có thể phục vụ được nhiều người sử dụng.
- Có thể tích hợp với các dịch vụ trực tuyến như email ...
- Không cần cài đặt tiết kiệm bộ nhớ của máy tính.
- Lưu trữ trực tuyến tránh mất dữ liệu.
- Thời gian phát triển ứng dụng nhanh, dễ dàng nâng cấp và bảo trì.

1.2.2. Về ngôn ngữ lập trình

- Ứng dụng được xây dựng bằng ngôn ngữ java.
- Đặc điểm của ngôn ngữ java:
 - Hướng đối tượng.
 - Độc lập với hệ nền.
 - Tính khả chuyển cao.
 - An toàn và bảo mật.
 - Cộng đồng sử dụng lớn, tài liệu tham khảo phong phú.

1.2.3. Về hệ quản trị cơ sở dữ liệu

- Ứng dụng sử dụng hệ quản trị MySQL.
- Một số đặc điểm của MySQL:
 - Là hệ cơ sở dữ liệu mã nguồn mở phổ biến nhất hiện nay.
 - Có đầy đủ tính năng của một hệ quản trị cơ sở dữ liệu thương mại như transaction, view, trigger, function ...
 - Nhanh, tin cậy, dễ dàng sử dụng.
 - Đa ngôn ngữ, đa nền.

1.2.5. Về kiến trúc ứng dụng

- Ứng dụng xây dựng dựa trên kiến trúc Spring Web MVC.
- Đặc điểm của Spring Web MVC:
 - Giúp cho việc xây dựng ứng dụng chặt chẽ và linh động.
 - Tách biệt các thành phần dữ liệu, xử lý và hiển thị.
 - Dễ dàng bảo trì và nâng cấp.
 - Cộng đồng sử dụng lớn, tài liệu tham khảo phong phú.

1.2.4. Về server

- Ứng dụng sử dụng server Tomcat.
- Một số đặc điểm của Tomcat:
 - Là một web server mã nguồn mở.
 - Độc lập hệ nền.

- Nhỏ gọn có tính ổn định cao.
- Dễ dàng triển khai ứng dụng.

1.2.6. Về công nghệ và framework

- Ứng dụng sử dụng các công nghệ như: AJAX, Hibernate và framework bootstrap.
- AJAX (tiếng anh: “Asynchronous JavaScript and XML” – nghĩa là “JavaScript và XML không đồng bộ”) là công nghệ phát triển web được sử dụng để tạo ra các ứng dụng web động. Với AJAX việc xử lý thông tin được thực hiện trên máy yêu cầu dịch vụ thay vì trên máy xử lý yêu cầu dịch vụ như cách truyền thống. Máy xử lý yêu cầu dịch vụ chỉ làm một việc đơn giản là nhận thông tin từ máy khách và trả các dữ liệu về cho máy khách. Máy yêu cầu dịch vụ xử lý sơ bộ thông tin của người dùng nhập vào, sau đó chuyển về máy xử lý yêu cầu dịch vụ rồi nhận dữ liệu từ máy xử lý yêu cầu dịch vụ và xử lý để hiển thị cho người dùng.
- Ưu điểm AJAX:
 - Trong nhiều trường hợp, các trang web chứa rất nhiều nội dung thông thường trong trang. Nếu sử dụng các phương pháp truyền thống, những nội dung đó sẽ phải nạp lại toàn bộ với từng yêu cầu. Tuy nhiên, nếu sử dụng AJAX một ứng dụng web có thể chỉ yêu cầu cho các nội dung cần thiết phải cập nhật, do đó giảm lượng lớn băng thông và thời gian nạp trang.
 - Việc dùng các yêu cầu không đồng bộ cho phép giao diện người dùng của ứng dụng hiển thị trên trình duyệt giúp người dùng trải nghiệm sự tương tác cao, với nhiều phần riêng lẻ.
 - Việc sử dụng Ajax có thể làm giảm các kết nối đến server, do các mã kịch bản (script) và các style sheet chỉ phải yêu cầu một lần.
- Hibernate: là một giải pháp ORM (Object/Relational mapping) mã nguồn mở cho các ứng dụng java. Đặc điểm của hibernate:
 - Mapping giữa các lớp đối tượng Java với các bảng trong cơ sở dữ liệu.
 - Truy xuất dễ dàng với các lớp và đối tượng.

- Hạn chế việc lặp lại các câu lệnh SQL.
- Bootstrap: là một front-end framework hỗ trợ thiết kế layout, các thành phần của website như: button, input, table, popup, màu sắc, icon ... Ưu điểm của bootstrap:
 - Thiết kế nhanh với các lớp css được viết sẵn.
 - Dễ dàng thay đổi layout, màu sắc.
 - Hỗ trợ responsive cho nhiều loại thiết bị.
 - Cộng đồng sử dụng lớn, tài liệu tham khảo phong phú.

1.2.7. Về phương pháp vẽ

Để xây dựng một website Java “động” cho phép người dùng vẽ sơ đồ, thiết kế giao diện có rất nhiều giải pháp để thực hiện. Dưới đây là một số giải pháp:

- Giải pháp 1: Mua bộ thư viện vẽ có đầy đủ các tính năng cần thiết cho việc xây dựng một phần mềm vẽ như: kéo thả, tạo các đối tượng phức tạp, chỉnh sửa thuộc tính các đối tượng, xuất tập tin, các thuật toán phục vụ việc vẽ v.v...
- Ưu điểm: cài đặt nhanh chóng, có nhiều thời gian để xây dựng một chương trình vẽ với nhiều chức năng, thành phần hơn.
- Nhược điểm: tính phí, không tích lũy được nhiều kiến thức về việc vẽ trên nền web.
- Giải pháp 2: Sử dụng api vẽ cơ bản của java.
 - Ưu điểm: không tính phí, kỹ năng lập được nâng cao, có nhiều tài liệu chi tiết về các hàm api.
 - Nhược điểm: thời gian cài đặt lâu, phải cài đặt toàn bộ các đối tượng, các xử lý cần để vẽ hình phức tạp. Cần nhiều thời gian để xây dựng một bộ thư viện riêng “hoàn hảo” dựa trên các api đó.
- Giải pháp 3: Sử dụng thư viện hỗ trợ vẽ miễn phí. Có hỗ trợ một vài hàm api di chuyển, liên kết đối tượng vẽ ...
 - Ưu điểm: không tính phí, giảm thời gian cài đặt các đối tượng và các xử lý trên đối tượng, có thêm thời gian để tập trung phát triển đề tài, nâng cao kỹ năng lập trình.
 - Nhược điểm: các hàm api chỉ hỗ trợ mức tối thiểu có thể xây dựng được một phần mềm vẽ.

Theo các phân tích và đánh giá trên thì giải pháp 3 là giải pháp phù hợp với đề tài luận văn này. Thư viện được lựa chọn là JointJS.

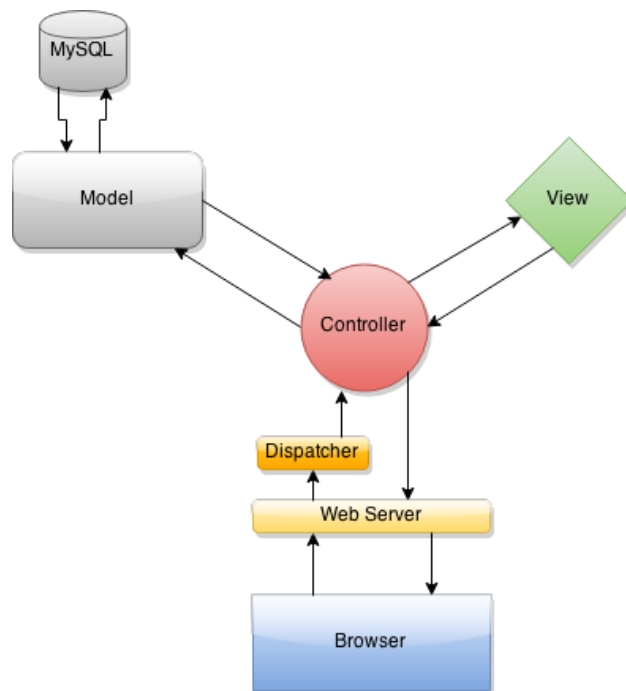
CHƯƠNG 2 – THIẾT KẾ VÀ CÀI ĐẶT GIẢI PHÁP

Đưa ra các kiến trúc hệ thống, thiết kế chức năng, thiết kế dữ liệu và trình bày cách thức cài đặt các thiết kế đó.

2.1. Thiết kế kiến trúc

- Hệ thống được xây dựng dựa theo kiến trúc của Spring Web MVC.

Hình 5. Kiến trúc tổng quan của hệ thống theo MVC.



- Giải thích các thành phần của hệ thống:
 - Browser: Các loại trình duyệt để chạy web như: IE, FireFox, Chrome ... có nhiệm vụ nhận các tương tác từ người dùng sau đó gửi các yêu cầu lên web server.
 - Webserver: Sử dụng tomcat 7 có nhiệm vụ lắng nghe các yêu cầu từ browser sau đó gửi các yêu cầu đó cho dispatcher.
 - Dispatcher: Bộ điều khiển phía trước có nhiệm vụ là nhận các yêu cầu và đưa ra các phản hồi.
 - Controller: đồng bộ hoá giữa Khung nhìn và Mô hình. Tức là với một trang JSP này thì sẽ tương ứng với lớp java nào để xử lý nó và ngược lại, kết quả sẽ trả về trang jsp nào. Nó đóng vai trò điều tiết giữa View và Model.
 - Model: gồm các lớp java có nhiệm vụ biểu diễn dữ liệu và cho phép truy tới để get và set dữ liệu, nó mô phỏng 1 cách đầy đủ đối tượng từ

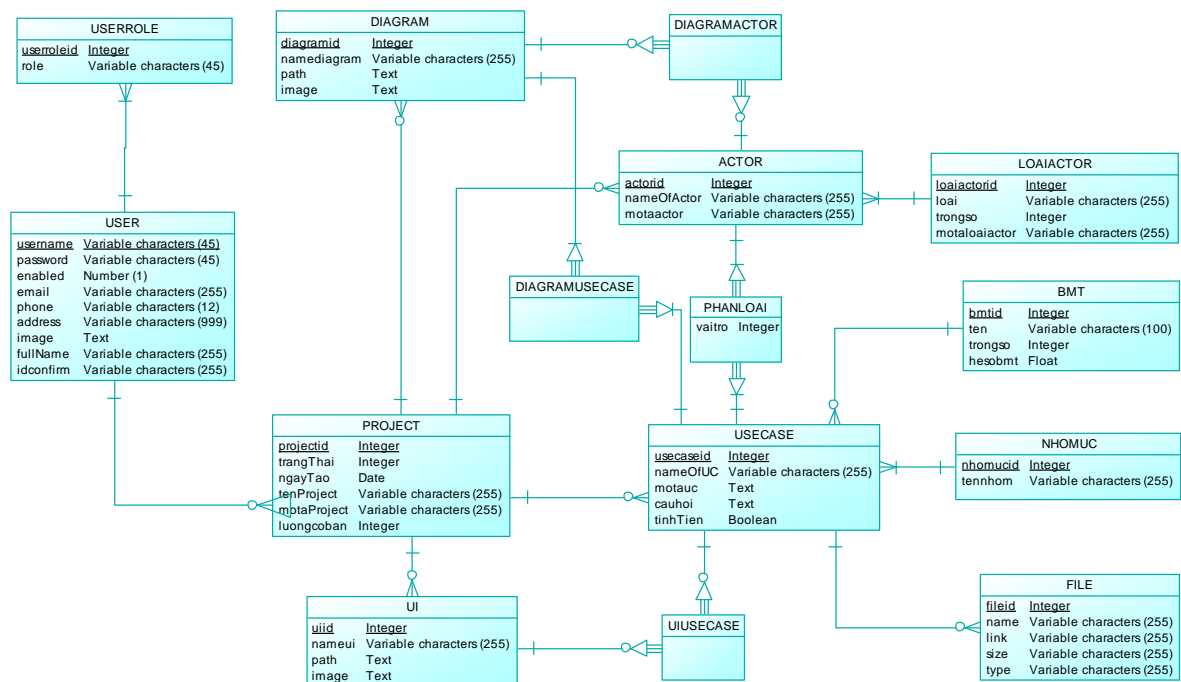
thể giới thực. Nhận yêu cầu từ Controller và thi hành các yêu cầu đó sau đó trả về giá trị tính toán theo yêu cầu.

- View: bao gồm các mã tương tự như JSP, HTML, CSS, XML, Javascript, JSON... để hiển thị giao diện người dùng, các dữ liệu trả về từ Model thông qua Controller.
- MySQL: Hệ quản trị cơ sở dữ liệu sử dụng, ở đây một bảng dữ liệu trong MySQL sẽ được ánh xạ thành một đối tượng java thông qua Hibernate.

2.2. Thiết kế dữ liệu

- Vì đề tài nhóm nên phần thiết kế dữ liệu chỉ thiết kế phần module đã được phân chia. Dưới đây trình bày mô hình dữ liệu mức quan niệm của module: đăng nhập, cập nhật thông tin cá nhân, vẽ sơ đồ usecase, vẽ giao diện.

Hình 6. Mô hình dữ liệu mức quan niệm.



- Giải thích các thực thể trên mô hình:

Bảng 11. Giải thích các thực thể trên mô hình

Thực thể	Giải thích
USER	<ul style="list-style-type: none"> - Mô tả đối tượng người dùng trong hệ thống. - Một (USER) có thể có nhiều (USERROLE). - Một (USER) có thể có nhiều (PROJECT) hoặc không có

	<p>PROJECT nào.</p> <ul style="list-style-type: none"> - Một USER trong hệ thống sau khi đăng ký thành công sẽ có các thông tin là: <ul style="list-style-type: none"> ○ username: tài khoản đăng nhập. - password: mật khẩu đăng nhập. - enabled: đã kích hoạt tài khoản hay chưa. (1 – đã kích hoạt, 0 – chưa kích hoạt). - fullname: tên đầy đủ của người dùng. - email: email của người dùng. - phone: số điện thoại của người dùng. - address: địa chỉ của người dùng. - image: ảnh đại diện của người dùng. - idconfirm: để so sánh khi người dùng nhấp vào link kích hoạt tài khoản.
USERROLE	<ul style="list-style-type: none"> - Mô tả quyền của người dùng ở đây có hai quyền khác nhau: thành viên và khách viếng thăm. - Một (USERROLE) chỉ thuộc một (USER). - Thuộc tính trong USERROLE: <ul style="list-style-type: none"> ○ userroleid: là id của quyền. ○ role: mô tả quyền của người dùng.
PROJECT	<ul style="list-style-type: none"> - Mô tả dự án của người dùng. - Một (PROJECT) phải thuộc một (USER). - Một (PROJECT) có thể có nhiều hoặc không có (DIAGRAM). - Một (PROJECT) có thể có nhiều hoặc không có (UI). - Một (PROJECT) có thể có nhiều hoặc không có (USECASE). - Một (PROJECT) có thể có nhiều hoặc không có (ACTOR). - Một dự án có các thuộc tính sau: <ul style="list-style-type: none"> ○ projectid: là id của dự án. ○ tenproject: tên của dự án. ○ motaproject: mô tả dự án. ○ trangthai: là trạng thái hiện tại của dự án (1 - Đang tiến hành, 2 - đã hoàn thành, 3 – tạm dừng). ○ ngaytao: ngày tạo ra dự án. ○ luongcoban: là lương được chọn để ước tính trong dự án.
DIAGRAM	<ul style="list-style-type: none"> - Mô tả sơ đồ usecase. - Một (DIAGRAM) phải thuộc một (PROJECT).

	<ul style="list-style-type: none"> - Một (DIAGRAM) có thể có nhiều (ACTOR). - Một (DIAGRAM) có thể có nhiều (USECASE). - Các thuộc tính của sơ đồ: <ul style="list-style-type: none"> o diagramid: là id của sơ đồ. o namediagram: tên của sơ đồ. o path: hình vẽ của sơ đồ dạng JSON. o image: hình vẽ của sơ đồ dạng SVG.
UI	<ul style="list-style-type: none"> - Mô tả giao diện. - Một (UI) có thể gắn trên nhiều hoặc không gắn (USECASE). - Các thuộc tính của giao diện: <ul style="list-style-type: none"> o uiid: là id của giao diện. o namui: tên của giao diện. o path: hình vẽ giao diện dạng JSON. o image: hình vẽ giao diện dạng SVG.
ACTOR	<ul style="list-style-type: none"> - Mô tả tác nhân trên sơ đồ usecase. - Một (ACTOR) phải thuộc một (PROJECT). - Một (ACTOR) phải thuộc một (LOIACTOR). - Một (ACTOR) có thể thuộc nhiều hoặc không thuộc (DIAGRAM). - Ứng với quan hệ của một (ACTOR) và một (USECASE) phải có một (PHANLOAI) với vai trò là tác nhân phụ hay tác nhân chính. - Các thuộc tính của tác nhân: <ul style="list-style-type: none"> o actorid: là id của tác nhân. o nameofactor: tên của tác nhân. o motaactor: mô tả về tác nhân.
LOIACTOR	<ul style="list-style-type: none"> - Mô tả loại của tác nhân. - Một (LOIACTOR) có thể có nhiều (ACTOR). - Các thuộc tính của loại tác nhân: <ul style="list-style-type: none"> o loaiactorid: là id của tác nhân. o loai: thuộc loại nào đơn giản, trung bình hay phức tạp. o trongso: đơn giản trọng số là 1, trung bình là 2, phức tạp là 3. o motaloaiactor: đưa ra mô tả loại tác nhân. (Đơn giản – thuộc loại giao diện của chương trình, trung bình – giao diện tương tác hoặc phục vụ một giao thức đồ họa, phức tạp – giao diện đồ họa.)

USECASE	<ul style="list-style-type: none"> - Mô tả usecase trên sơ đồ usecase. - Một (USECASE) phải thuộc một (PROJECT). - Một (USECASE) có thể liên kết đến nhiều hoặc không liên kết (UI). - Ứng với quan hệ của một (USECASE) và một (ACTOR) phải có một (PHANLOAI) với vai trò là tác nhân phụ hay tác nhân chính. - Một (USECASE) phải thuộc một (NHOMUC). - Một (USECASE) phải được gán một (BMT). - Một (USECASE) có thể không có hoặc có nhiều (FILE). - Các thuộc tính của usecase: <ul style="list-style-type: none"> o usecaseid: là id của usecase. o nameofuc: tên của usecase. o motauc: mô tả của usecase. o cauhoi: các câu hỏi liên quan đến usecase. o tinhvien: là usecase tính tiền hay không tính tiền. (1 – tính tiền, 0 – không tính tiền).
NHOMUC	<ul style="list-style-type: none"> - Mô tả nhóm của usecase. - Một (NHOMUC) có thể có nhiều (USECASE). - Các thuộc tính của nhóm usecase: <ul style="list-style-type: none"> o nhomucid: là id của nhóm. o tennhom: tên nhóm usecase.
BMT	<ul style="list-style-type: none"> - Hệ số môi trường. - Một (BMT) có thể không có hoặc có nhiều (USECASE). - Các thuộc tính của hệ số môi trường: <ul style="list-style-type: none"> o bmtid: là id của hệ số môi trường. o ten: tên của hệ số môi trường. o trongso: trọng số của hệ số môi trường. o hesobmt: hệ số của môi trường.
FILE	<ul style="list-style-type: none"> - Các tập tin được thu thập theo usecase. - Một (FILE) phải thuộc một (USECASE). - Các thuộc tính của tập tin: <ul style="list-style-type: none"> o fileid: là id của tập tin. o name: tên của tập tin. o link: đường dẫn của tập tin. o size: kích thước của tập tin. o type: loại tập tin.

- Mô hình dữ liệu mức vật lý:

Bảng 12. Bảng ACTOR

Tên thuộc tính	Kiểu dữ liệu	Khóa chính	Khóa ngoại	Giá trị mặc định	NN	Diễn giải
actorid	INTEGER	✓			✓	Mã tác nhân
motaactor	VARCHAR(255)					Mô tả tác nhân
nameofactor	VARCHAR(255)				✓	Tên tác nhân
loaiactorid	INTEGER				✓	Mã loại tác nhân
projectid	INTEGER		✓		✓	Mã dự án

Bảng 13. Bảng BMT.

Tên thuộc tính	Kiểu dữ liệu	Khóa chính	Khóa ngoại	Giá trị mặc định	NN	Diễn giải
bmtid	INTEGER	✓			✓	Mã BMT
ten	VARCHAR(255)				✓	Tên BMT
trongso	VARCHAR(255)				✓	Trọng số
hesobmt	INTEGER				✓	Hệ số BMT

Bảng 14. Bảng DIAGRAM.

Tên thuộc tính	Kiểu dữ liệu	Khóa chính	Khóa ngoại	Giá trị mặc định	NN	Diễn giải
diagramid	INTEGER	✓			✓	Mã sơ đồ
namediagram	VARCHAR(255)				✓	Tên sơ đồ

path	LONG TEXT					Dữ liệu sơ đồ
image	LONG TEXT					Ảnh hiển thị sơ đồ
projectid	INTEGER		✓		✓	Mã dự án

Bảng 15. Bảng DIAGRAMACTOR.

Tên thuộc tính	Kiểu dữ liệu	Khóa chính	Khóa ngoại	Giá trị mặc định	NN	Diễn giải
actorid	INTEGER	✓			✓	Mã tác nhân
diagramid	INTEGER	✓			✓	Mã sơ đồ

Bảng 16. Bảng DIAGRAMUSECASE.

Tên thuộc tính	Kiểu dữ liệu	Khóa chính	Khóa ngoại	Giá trị mặc định	NN	Diễn giải
Usecaseid	INTEGER	✓			✓	Mã use-case
diagramid	INTEGER	✓			✓	Mã sơ đồ

Bảng 17. Bảng FILE.

Tên thuộc tính	Kiểu dữ liệu	Khóa chính	Khóa ngoại	Giá trị mặc định	NN	Diễn giải
fileid	INTEGER	✓			✓	Mã tập tin
name	VARCHAR(255)				✓	Tên tập tin
size	VARCHAR(255)					Kích thước tập tin

type	VARCHAR(255)					Loại tập tin
link	INTEGER				✓	Đường dẫn tập tin
usecaseid	INTEGER		✓		✓	Mã use-case

Bảng 18. Bảng LOAIActor.

Tên thuộc tính	Kiểu dữ liệu	Khóa chính	Khóa ngoại	Giá trị mặc định	NN	Diễn giải
loaiactorid	INTEGER	✓			✓	Mã loại tác nhân
loai	VARCHAR(255)				✓	Tên loại tác nhân
trongso	INTEGER				✓	Trọng số
motaloaiactor	VARCHAR(255)				✓	Mô tả loại tác nhân

Bảng 19. Bảng NHOMUC.

Tên thuộc tính	Kiểu dữ liệu	Khóa chính	Khóa ngoại	Giá trị mặc định	NN	Diễn giải
nhomucid	INTEGER	✓			✓	Mã nhóm use-cases
tennhom	VARCHAR(255)				✓	Tên nhóm use-cases
projectid	INTEGER				✓	Mã dự án

Bảng 20. Bảng PHANLOAI.

Tên thuộc tính	Kiểu dữ liệu	Khóa chính	Khóa ngoại	Giá trị mặc định	NN	Diễn giải
actorid	INTEGER	✓			✓	Mã tác nhân
usecaseid	INTEGER				✓	Mã use-case
vaitro	INTEGER				✓	Vai trò

Bảng 21. Bảng PROJECT.

Tên thuộc tính	Kiểu dữ liệu	Khóa chính	Khóa ngoại	Giá trị mặc định	NN	Diễn giải
projectid	INTEGER	✓			✓	Mã dự án
trangthai	INTEGER				✓	Trạng thái
ngaytao	DATE				✓	Ngày tạo
tenproject	VARCHAR(255)				✓	Tên dự án
motaproject	VARCHAR(255)					Mô tả dự án

Bảng 22. Bảng UI.

Tên thuộc tính	Kiểu dữ liệu	Khóa chính	Khóa ngoại	Giá trị mặc định	NN	Diễn giải
uiid	INTEGER	✓			✓	Mã giao diện
nameui	VARCHAR(255)				✓	Tên giao diện
path	LONG TEXT				✓	Dữ liệu giao diện
image	LONG TEXT					Ảnh giao diện
projectid	INTEGER		✓		✓	Mã dự án

Bảng 23. Bảng UIUSECASE.

Tên thuộc tính	Kiểu dữ liệu	Khóa chính	Khóa ngoại	Giá trị mặc định	NN	Diễn giải
uiid	INTEGER	✓			✓	Mã giao diện
usecaseid	INTEGER	✓			✓	Mã use-cases

Bảng 24. Bảng USECASE.

Tên thuộc tính	Kiểu dữ liệu	Khóa chính	Khóa ngoại	Giá trị mặc định	NN	Diễn giải
usecaseid	INTEGER	✓			✓	Mã use-case
namofuc	VARCHAR(255)				✓	Tên use-case
motauc	TEXT					Mô tả use-cases
cauhoi	TEXT					Câu hỏi phỏng vấn
tinhtien	INTEGER				✓	Có tính tiền hay không
projectid	INTEGER		✓		✓	Mã dự án
nhomucid	INTEGER		✓		✓	Mã nhóm use-cases
bmtid	INTEGER		✓		✓	Mã BMT

2.3. Thiết kế chức năng

2.3.1. Chức năng đăng nhập

Mô tả	Cho phép người dùng đăng nhập vào hệ thống với tài khoản và mật khẩu đăng ký đã kích hoạt thành công.
-------	---

Tác nhân	Thành viên
Dòng sự kiện chính	1. Người dùng click nút đăng nhập tại trang chủ. 2. Hệ thống hiển thị form đăng nhập. 3. Người dùng nhập tài khoản và mật khẩu. 4. Người dùng click đăng nhập. 5. Hệ thống kiểm tra tài khoản và mật khẩu. 6. Hệ thống chuyển đến trang các dự án.
Dòng sự kiện thay thế	Luồng A – người dùng nhập sai tài khoản hoặc mật khẩu. Chuỗi A bắt đầu từ bước 4 của dòng sự kiện chính 5. Hệ thống thông báo đăng nhập không thành công. 6. Hệ thống chuyển đến trang đăng nhập.
Tiền điều kiện	

- Giao diện chức năng đăng nhập:

Hình 7. Giao diện chức năng đăng nhập

- Mô tả các thành phần của giao diện chức năng đăng nhập:

Thành phần	Loại	Ghi chú
	Input – text	Độ dài tối thiểu 6 ký tự và tối đa 30 ký tự.
	Input - password	Độ dài tối thiểu 6 ký tự và tối đa 30 ký tự.
	Button	

2.3.2. Chức năng cập nhật thông tin cá nhân

Mô tả	Cho phép người dùng chỉnh sửa thông tin cá nhân: tên, email, địa chỉ, số điện thoại.
--------------	--

Tác nhân	Thành viên
Dòng sự kiện chính	<ol style="list-style-type: none"> 1. Người dùng click chọn cập nhật thông tin cá nhân 2. Hệ thống hiển thị form cập nhật thông tin cá nhân: tên, email, địa chỉ, số điện thoại. 3. Người dùng chỉnh sửa thông tin cần cập nhật. 4. Người dùng click nút cập nhật. 5. Hệ thống lưu thay đổi vào CSDL. 6. Hệ thống thông báo cập nhật thành công.
Tiền điều kiện	Đăng nhập thành công vào hệ thống.

- Giao diện chức năng cập nhật thông tin cá nhân:

Hình 8. Giao diện chức năng cập nhật thông tin cá nhân.

Họ và tên:

Nhập họ và tên vào đây

Email:

boymattrom3@gmail.com

Số điện thoại:

Nhập số điện thoại vào đây

Ví dụ: 0123333444 hoặc 01233334445 (Tổng cộng 10 số hoặc 11 số)

Địa chỉ:

CẬP NHẬT

- Mô tả các thành phần của giao diện chức năng cập nhật thông tin cá nhân:

Thành phần	Loại	Ghi chú
<p>Họ và tên:</p> <p>Nhập họ và tên vào đây</p>	Label, input – text	Họ tên
<p>Email:</p> <p>boymattrom3@gmail.com</p>	Label, input – email	Email

Số điện thoại: Nhập số điện thoại vào đây	Label, input – text	Số điện thoại
Địa chỉ: _____	Lable, textarea	Địa chỉ
CẬP NHẬT	Button	

2.3.3. Chức năng tạo mới sơ đồ usecase.

Mô tả	Cho phép người tạo mới sơ đồ usecase.
Tác nhân	Thành viên
Dòng sự kiện chính	<ol style="list-style-type: none"> 1. Người dùng click chọn tạo mới sơ đồ. 2. Hệ thống hiển thị form tạo mới sơ đồ. 3. Người dùng nhập tên sơ đồ cần tạo. 4. Người dùng click nút tạo. 5. Hệ thống lưu sơ đồ vào CSDL. 6. Hệ thống hiển thị sơ đồ vừa mới tạo.
Tiền điều kiện	Đăng nhập thành công vào hệ thống.

- Giao diện chức năng tạo mới sơ đồ usecase:

Hình 9. Giao diện chức năng tạo mới sơ đồ usecase.

- Mô tả các thành phần giao diện chức năng tạo mới sơ đồ usecase:

Bảng 25. Bảng mô tả thành phần giao diện chức năng tạo mới sơ đồ usecase.

Thành phần	Loại	Mô tả
+ Tạo sơ đồ mới	Label	Tên của giao diện.

<input type="text" value="Nhập tên nhóm"/>	Input	Cho phép người dùng nhập tên nhóm.
<input type="button" value="Tạo"/>	Button	Nhấn để tạo sơ đồ.
<input type="button" value="X"/>	Button	Đóng giao diện.

2.3.4. Chức năng xuất ảnh

Mô tả	Cho phép người dùng xuất ảnh với định dạng svg hoặc png.
Tác nhân	Thành viên
Dòng sự kiện chính	<ol style="list-style-type: none"> 1. Người dùng click chọn xuất ảnh. 2. Hệ thống hiển thị form xuất ảnh. 3. Người dùng chọn loại ảnh cần xuất. 4. Người dùng nhập tên ảnh cần xuất. 5. Người dùng click nút xuất ảnh. 6. Hệ thống chuyển sơ đồ đang vẽ thành ảnh. 5. Hệ thống tải ảnh về máy người dùng.
Tiền điều kiện	Đăng nhập thành công vào hệ thống.

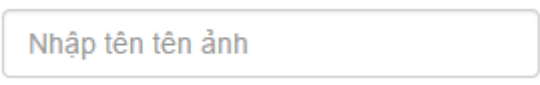

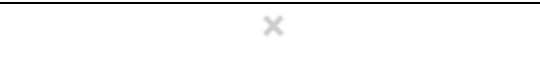
- Giao diện chức năng xuất ảnh:

Hình 10. Giao diện chức năng xuất ảnh.

- Mô tả các thành phần của giao diện chức năng xuất ảnh:

Bảng 26. Bảng mô tả thành phần giao diện chức năng xuất ảnh.

Thành phần	Loại	Mô tả
Xuất ảnh	Label	Tên của giao diện.

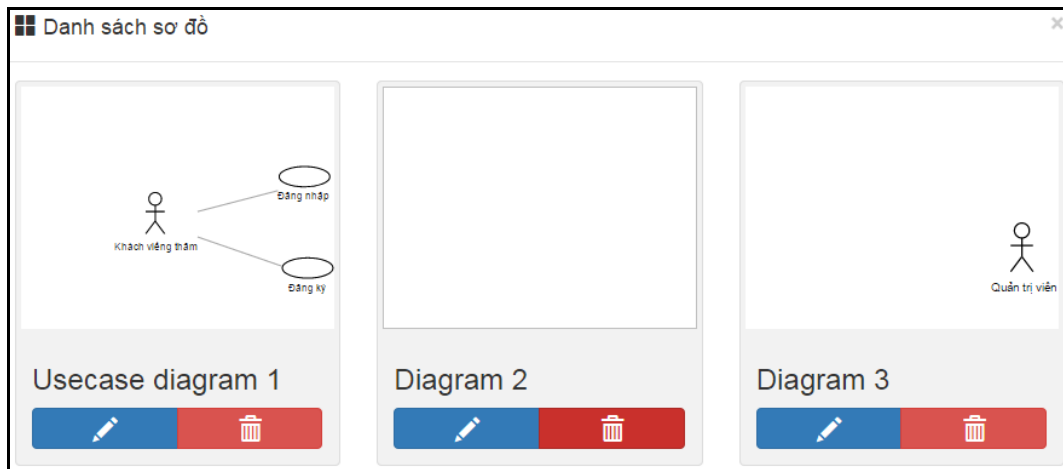
	Input	Cho phép người dùng nhập tên ảnh.
	Button	Nhấn để xuất ảnh.
	Button	Đóng giao diện.

2.3.5. Chức năng xem danh sách sơ đồ usecase

Mô tả	Cho phép người dùng xem danh sách các sơ đồ usecase đã vẽ.
Tác nhân	Thành viên
Dòng sự kiện chính	<ol style="list-style-type: none"> 1. Người dùng click chọn xem danh sách sơ đồ. 2. Hệ thống hiển thị danh sách các sơ đồ.
Dòng sự kiện phụ	<p>Luồng A – người dùng chỉnh sửa sơ đồ. Chuỗi a bắt đầu từ bước 2 của dòng sự kiện chính.</p> <ol style="list-style-type: none"> 3. Người dùng chọn chỉnh sửa sơ đồ. 4. Hệ thống hiển thị sơ đồ cần chỉnh sửa. <p>Luồng B – người dùng xóa sơ đồ. Chuỗi b bắt đầu từ bước 2 của dòng sự kiện chính.</p> <ol style="list-style-type: none"> 3. Người dùng chọn xóa sơ đồ. 4. Hệ thống hiện thông báo “Bạn có chắc chắn xóa không?”. 5. Người dùng chọn có. 6. Hệ thống xóa sơ đồ. <p>Luồng B1 – người dùng hủy xóa sơ đồ. Chuỗi b1 bắt đầu từ bước 4 của dòng sự kiện chính.</p> <ol style="list-style-type: none"> 5. Người dùng chọn không.
Tiền điều kiện	Đăng nhập thành công vào hệ thống.


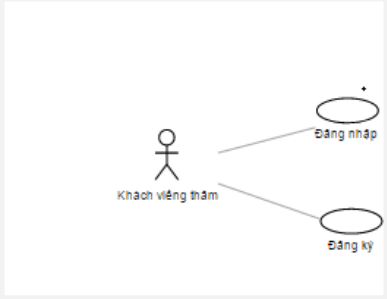




- Giao diện chức năng xem danh sách sơ đồ usecase:

Hình 11. Giao diện danh sách sơ đồ usecase.



- Mô tả các thành phần của chức năng xem danh sách sơ đồ usecase:

Bảng 27. Bảng mô tả danh sách sơ đồ.

Thành phần	Loại	Mô tả
 Danh sách sơ đồ	Label	Tên của giao diện.
	Div, Image	Hiển thị ảnh của sơ đồ.
Usecase diagram 1	Label	Tên sơ đồ.
	Button	Cho phép người dùng nhấn  để chỉnh sửa sơ đồ hoặc nhấn  để xóa.
	Button	Đóng giao diện.

2.3.6. Chức năng đổi tên sơ đồ

Mô tả	Cho phép người dùng đổi tên sơ đồ usecase.
Tác nhân	Thành viên
Dòng sự kiện	1. Người dùng click chọn đổi tên sơ đồ.

chính	2. Hệ thống hiển thị form đổi tên sơ đồ. 3. Người dùng nhập tên sơ đồ cần đổi. 4. Người dùng click nút lưu thay đổi. 5. Hệ thống kiểm tra tên sơ đồ. 6. Hệ thống lưu tên sơ đồ đã thay đổi.
Dòng sự kiện phụ	Luồng A – tên sơ đồ đã tồn tại Chuỗi a bắt đầu từ bước 5 của dòng sự kiện chính. 5. Hệ thống thông báo tên sơ đồ đã tồn tại.
Tiền điều kiện	Đăng nhập thành công vào hệ thống.

- Giao diện chức năng đổi tên sơ đồ usecase:

Hình 12. Giao diện đổi tên sơ đồ.

Bảng 28. Mô tả thành phần đổi tên sơ đồ.

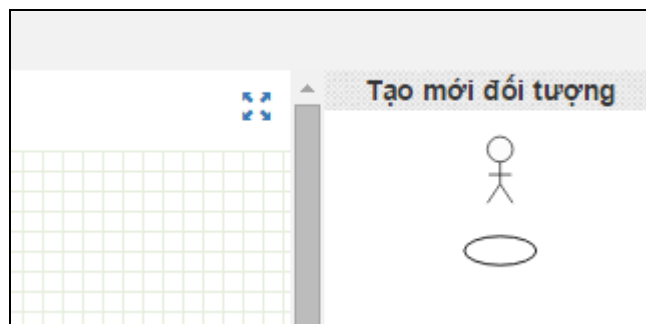
Thành phần	Loại	Mô tả
Đổi tên sơ đồ	Label	Tên của giao diện.
<input type="text" value="Nhập tên sơ đồ"/>	Input	Cho phép người dùng nhập tên sơ đồ.
	Button	Nhấn để đổi tên.
	Button	Đóng giao diện.

2.3.6. Chức năng tạo mới tác nhân





Mô tả	Cho phép người dùng tạo mới tác nhân.
--------------	---------------------------------------

Tác nhân	Thành viên
Dòng sự kiện chính	<ol style="list-style-type: none"> 1. Người dùng kéo thả tác nhân từ “tạo mới đối tượng”. 2. Hệ thống lưu tác nhân vào cơ sở dữ liệu. 3. Hệ thống hiển thị tác nhân lên trang vẽ. 4. Người dùng click đôi vào tên tác nhân. 5. Người dùng thay đổi tên tác nhân. 6. Hệ thống lưu thay đổi vào CSDL. 7. Hệ thống hiển thị lại tên tác nhân lên trang vẽ.
Tiền điều kiện	Đăng nhập thành công vào hệ thống.

- Giao diện chức năng tạo mới tác nhân:



- Mô tả các thành phần giao diện chức năng tạo mới giao diện:

Thành phần	Loại	Mô tả
	Image	Thể hiện tác nhân mới.
	Image	Thể hiện usecase mới.
	Div	Trang vẽ
	Button	Cho phép phóng to thu nhỏ trang vẽ.

2.3.7. Chức năng chỉnh sửa thông tin tác nhân.

Mô tả	Cho phép người dùng chỉnh sửa thông tin tác nhân.
Tác nhân	Thành viên
Dòng sự kiện chính	<ol style="list-style-type: none"> 1. Người dùng click đôi vào một tác nhân. 2. Hệ thống hiển thị form thông tin tác nhân gồm có: tên tác nhân, loại tác nhân, mô tả, vai trò với chức năng. 3. Người dùng chỉnh sửa thông tin tác nhân. 4. Người dùng click nút lưu thay đổi. 5. Hệ thống lưu thông tin tác nhân.
Tiền điều kiện	Đăng nhập thành công vào hệ thống.

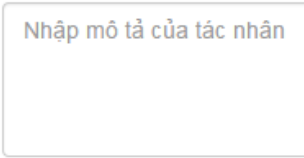
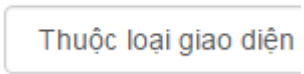
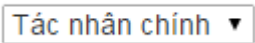
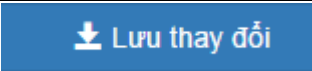

- Giao diện chức năng chỉnh sửa thông tin tác nhân:

Hình 13. Giao diện chức năng chỉnh sửa thông tin tác nhân.

- Mô tả các thành phần của giao diện chức năng chỉnh sửa thông tin tác nhân:

Bảng 29. Bảng mô tả thành phần giao diện chức năng chỉnh sửa tác nhân.

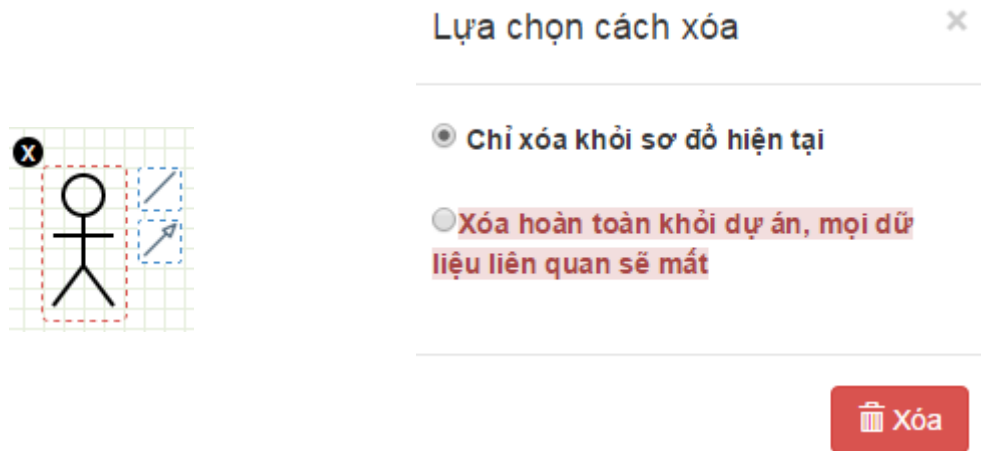
Thành phần	Loại	Mô tả
Tác nhân: Quản trị viên	Label	Tên của tác nhân.

Mô tả 	Input	Cho phép người dùng nhập mô tả tác nhân.
Loại tác nhân 	Selectbox	Cho phép người dùng chọn các loại cho tác nhân.
Vai trò với chức năng Đăng nhập	Label	Hiển thị danh sách tên chức năng.
	Selectbox	Cho phép người dùng lựa chọn vai trò của tác nhân với use-case là tác nhân chính hay tác nhân phụ.
	Button	Nhấn để lưu thông tin thay đổi.
	Button	Đóng giao diện.




2.3.8. Chức năng xóa tác nhân

Mô tả	Cho phép người dùng xóa tác nhân trên trang vẽ hoặc xóa hoàn toàn khỏi CSDL.
Tác nhân	Thành viên
Dòng sự kiện chính	<ol style="list-style-type: none"> 1. Người dùng nhấp chọn tác nhân. 2. Hệ thống hiển thị các thao tác có thể thực hiện xung quanh tác nhân. 3. Người dùng click vào button đóng “X” trên tác nhân. 4. Hệ thống xóa tác nhân khỏi trang vẽ nhưng không xóa khỏi CSDL.
Dòng sự kiện thay thế	Luồng A – Người dùng muốn xóa tác nhân khỏi CSDL Chuỗi a bắt đầu từ bước 3 của dòng sự kiện chính <ol style="list-style-type: none"> 1. Người dùng nhấn phím “delete”. 2. Hệ thống hiển thị form cho phép người dùng lựa chọn cách xóa. 3. Người dùng lựa chọn xóa khỏi CSDL. 4. Hệ thống xóa tác nhân khỏi trang vẽ và CSDL.
Tiền điều kiện	Đăng nhập thành công vào hệ thống.

- Giao diện chức năng xóa tác nhân:



- Mô tả các thành phần giao diện chức năng xóa tác nhân:

Thành phần	Loại	Ghi chú
	Button	Click để xóa tác nhân khỏi trang vẽ.
	Imge	Cho phép người dùng kéo thả các mối quan hệ.
<input checked="" type="radio"/> Chỉ xóa khỏi sơ đồ hiện tại <input type="radio"/> Xóa hoàn toàn khỏi dự án, mọi dữ liệu liên quan sẽ mất	Radio	Cho phép người dùng chọn cách thức xóa.
	Button	Click để xóa.

2.3.9. Chức năng chỉnh sửa thông tin usecase.

Mô tả	Cho phép người dùng chỉnh sửa thông tin usecase.
Tác nhân	Thành viên
Dòng sự kiện chính	1. Người dùng click đôi vào một usecase. 2. Hệ thống hiển thị form thông tin usecase gồm có: tên usecase, mô tả, mức độ, nhóm, tính tiền, gắn trên các giao

	diện, đặt câu hỏi và upload các tập tin, tác nhân sử dụng. 3. Người dùng chỉnh sửa thông tin usecase. 4. Người dùng click nút lưu thay đổi. 5. Hệ thống lưu thông tin usecase.
Dòng sự kiện thay thế	Luồng A – người dùng click xem liên kết giao diện Chuỗi a bắt đầu từ bước 2 của dòng sự kiện chính. 3. Người dùng click vào link liên kết giao diện. 4. Hệ thống hiển thị giao diện liên kết. Luồng B – liên kết đến chức năng thu thập yêu cầu Chuỗi b bắt đầu từ bước 2 của dòng sự kiện chính 3. Người dùng click vào link thu thập yêu cầu. 4. Hệ thống hiển thị trang thu thập yêu cầu.
Tiền điều kiện	Đăng nhập thành công vào hệ thống.

- Giao diện chức năng chỉnh sửa thông tin usecase:

Hình 14. Giao diện chức năng chỉnh sửa thông tin use-case.

Usecase: Đăng nhập

Mô tả

Nhập mô tả của trường hợp sử dụng

Mức độ

B - Chức năng bắt buộc

M - Chức năng mong muốn

T - Chức năng tùy chọn

B - Đơn giản

☒ Tính tiền

Đặt câu hỏi và upload các tập tin liên quan

Thuộc nhóm

sdf

sdf

Gắn trên các giao diện

Giao diện đăng nhập

Tác nhân sử dụng

Khách viếng thăm

Lưu thay đổi

- Mô tả các thành phần của giao diện chức năng chỉnh sửa thông tin usecase:

Bảng 30. Mô tả thành phần giao diện chức năng chỉnh sửa thông tin usecase.

Thành phần	Loại	Mô tả
------------	------	-------

Usecase: Quản lý nhân sự	Label	Tên của use-case.
Mô tả <input type="text" value="Nhập mô tả của trường hợp sử dụng"/>	Input	Cho phép người dùng nhập mô tả use-case.
<input type="text" value="B - Đơn giản"/>	Selectbox	Cho phép người dùng chọn các mức độ cho use-case.
<input checked="" type="checkbox"/> Tính tiền	Checkbox	Nếu có chọn thì là chức năng có tính tiền, nếu không chọn là chức năng không tính tiền.
Đặt câu hỏi và upload các tập tin liên quan	Link	Nhấn để chuyển sang chức năng thu thập yêu cầu.
Thuộc nhóm <input type="text" value="so do abc"/>	Input, Selectbox	Hiện thị nhóm đã chọn và danh sách các nhóm có trong dự án.
Gắn trên các giao diện Giao diện đăng nhập	Link	Liên kết đến giao diện đang gắn.
Tác nhân sử dụng <input type="text" value="Khách viếng thăm"/>	Lable	Hiện thị tác nhân đang sử dụng use-case.
<input type="button" value="Lưu thay đổi"/>	Button	Nhấn để lưu thay đổi.
<input type="button" value="X"/>	Button	Đóng giao diện.

2.3.10. Chức năng tạo mới giao diện

Mô tả	Cho phép người dùng tạo mới giao diện.
Tác nhân	Thành viên
Dòng sự kiện chính	<ol style="list-style-type: none"> 1. Người dùng chọn tạo mới giao diện. 2. Hệ thống hiển thị form tạo mới giao diện. 3. Người dùng nhập tên giao diện. 4. Người dùng click nút tạo. 5. Hệ thống lưu giao diện vào CSDL.

	5. Hệ thống hiển thị giao diện vừa mới tạo.
Tiền điều kiện	Đăng nhập thành công vào hệ thống.

- Giao diện chức năng tạo mới giao diện:

Hình 15. Giao diện chức năng tạo mới giao diện.

- Mô tả các thành phần giao diện chức năng tạo mới giao diện:

Bảng 31. Mô tả thành phần giao diện chức năng tạo mới giao diện.

Thành phần	Loại	Mô tả
Tạo giao diện mới	Label	Tên của giao diện.
<input type="text" value="Nhập tên giao diện"/>	Input	Cho phép người dùng nhập tên giao diện.
	Button	Nhấn để tạo giao diện.
	Button	Đóng giao diện.

2.3.11. Chức năng gán giao diện vào usecase.

Mô tả	Cho phép người dùng gán giao diện vào các usecase đã vẽ.
Tác nhân	Thành viên
Dòng sự kiện chính	<ol style="list-style-type: none"> 1. Người dùng chọn gán giao diện vào usecase. 2. Hệ thống hiển thị form gán giao diện vào usecase. 3. Người dùng click chọn usecase cần gán. 4. Người dùng click nút gán. 5. Hệ thống lưu thông tin vào CSDL.
Tiền điều kiện	Đăng nhập thành công vào hệ thống.

- Giao diện chức năng gán giao diện vào usecase:

Hình 16. Giao diện gán giao diện vào usecase.

- Mô tả các thành phần giao diện chức năng gán giao diện vào usecase.

Bảng 32. Bảng mô tả thành phần giao diện chức năng gán giao diện vào usecase.

Thành phần	Loại	Mô tả
Lựa chọn Use-case gán vào giao diện	Label	Tên của giao diện.
Nhóm sdf	Label	Tên nhóm use-case.
<input checked="" type="checkbox"/> Đăng nhập <input type="checkbox"/> Usecase	Checkbox	Danh sách các use-case có thể gán. Có check là đã gán, ngược lại chưa gán.
Gán	Button	Nhấn để lưu thay đổi.
X	Button	Đóng giao diện.

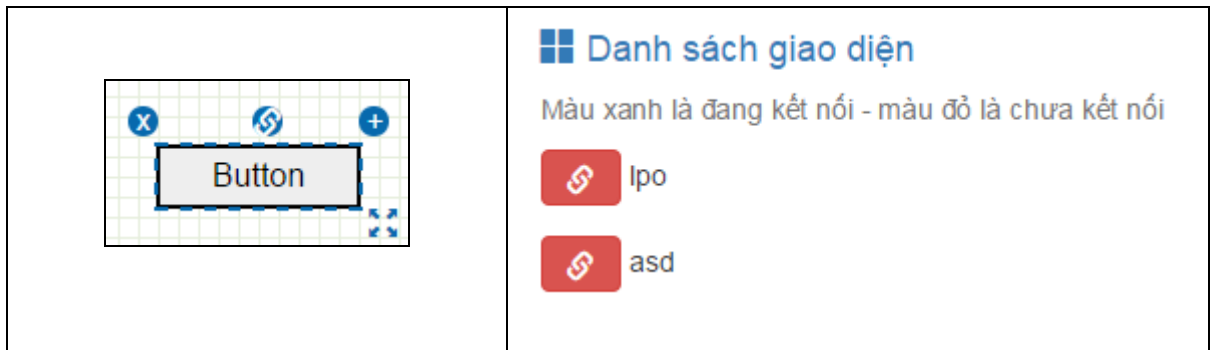
2.3.12. Chức năng liên kết các giao diện

Mô tả	Cho phép người dùng liên kết giữa các giao diện với nhau
Tác nhân	Thành viên
Dòng sự kiện chính	1. Người dùng tạo một Button. 2. Người dùng nhấp chọn Button. 3. Hệ thống hiển thị các thao tác xung quanh button. 4. Người dùng click chọn thao tác link 5. Hệ thống hiển thị form liên kết giao diện.

	6. Người dùng click chọn giao diện cần liên kết. 7. Hệ thống lưu thay đổi vào CSDL.
Tiền điều kiện	Đăng nhập thành công vào hệ thống.

- Giao diện chức năng liên kết các giao diện:

Hình 17. Giao diện chức năng liên kết các giao diện.



- Mô tả các thành phần giao diện chức năng liên kết các giao diện:

Bảng 33. Bảng mô tả thành phần giao diện chức năng liên kết các giao diện.

Thành phần	Loại	Mô tả
	Button	Click để xóa.
	Button	Click để mở form liên kết giao diện
	Button	Click để copy thành phần.
	Button	Kéo để thay đổi kích thước.
lpo	Button	Click để liên kết đến giao diện.

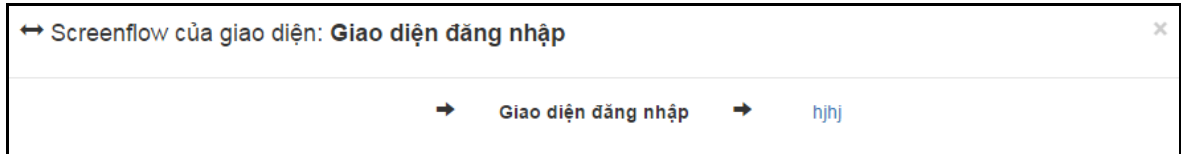
2.3.13. Chức năng xem liên kết giữa các giao diện

Mô tả	Cho phép người dùng xem liên kết giữa các giao diện đã vẽ.
Tác nhân	Thành viên
Dòng sự kiện chính	1. Người dùng chọn screenflow. 2. Hệ thống hiển thị danh sách liên kết đến giao diện.

	3. Người dùng click vào tên giao diện để xem giao diện. 4. Hệ thống hiển thị giao diện người dùng cần xem.
Tiền điều kiện	Đăng nhập thành công vào hệ thống.

- Giao diện chức năng xem liên kết giữa các giao diện:

Hình 18. Giao diện Screen Flow.



- Mô tả các thành phần giao diện chức năng xem liên kết giữa các giao diện:

Bảng 34. Bảng mô tả thành phần giao diện Screen Flow.

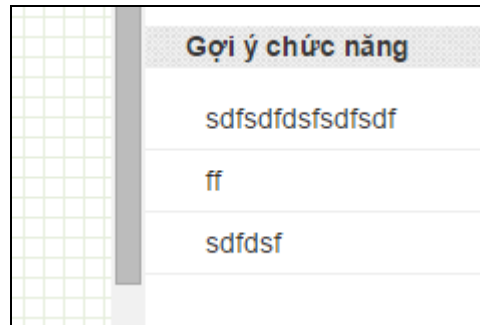
Thành phần	Loại	Mô tả
↔ Screenflow của giao diện	Label	Tên của giao diện.
hjhj	Link	Tên giao diện đang liên kết tới.
X	Button	Đóng giao diện.

2.3.14. Chức năng gợi ý usecase theo chức năng đã có

Mô tả	Gợi ý cho người dùng các usecase theo chức năng đã có để vẽ.
Tác nhân	Thành viên
Dòng sự kiện chính	1. Hiện thị danh sách usecase gợi ý. 2. Kéo thả usecase gợi ý vào trang vẽ. 3. Hệ thống kiểm tra usecase. 4. Hệ thống lưu usecase vào CSDL. 5. Hệ thống hiển thị usecase lên trang vẽ.
Dòng sự kiện thay thế	Luồng A – usecase đã tồn tại. Chuỗi a bắt đầu từ bước 3 của dòng sự kiện chính. 4. Hệ thống thông báo usecase đã tồn tại.
Tiền điều kiện	Đăng nhập thành công vào hệ thống.

- Giao diện chức năng gợi ý usecase theo chức năng đã có:

Hình 19. Giao diện chức năng gợi ý usecase theo chức năng đã có.



- Mô tả các thành phần giao diện chức năng xem liên kết giữa các giao diện:

Bảng 35. Bảng mô tả thành phần giao diện Screen Flow.

Thành phần	Loại	Ghi chú
<u>sdfsdfsdfsdfsdf</u>	div	Chức năng gợi ý

2.4. Cài đặt giải pháp

2.4.1. Cài đặt thư viện JointJS

- JointJS là một thư viện hỗ trợ vẽ các biểu đồ tương tác trên nền web. Nó được xây dựng dựa trên JQuery, Backbone, Undercore và SVG. Một số biểu đồ có thể sử dụng JointJS để cài đặt: sơ đồ lớp, sơ đồ thực thể quan hệ, sơ đồ trạng thái, biểu đồ báo cáo ... Sử dụng JointJS để cài đặt các đối tượng vẽ như: Actor, Usecase, Div, Form, input text, textarea, label, button, table ...
- JointJS có 2 bộ API chúng ta có thể sử dụng:
 - JointJS API: sử dụng miễn phí, các tính năng hạn chế.
 - Rappid API: tính phí. Đầy đủ các tính năng cần thiết.
- Đề tài này sử dụng bộ JointJS API miễn phí.
- Các đối tượng cơ bản trong JointJS:

Bảng 36. Bảng các đối tượng cơ bản trong JOINTJS.

Đối tượng	Mô tả	Cách thức khai báo
Graph	Là một model nó chứa tất	<pre>var graph = new joint.dia.Graph;</pre>

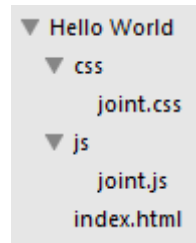
	cả các phần tử vẽ.	
Paper	<ul style="list-style-type: none"> - Là view hiển thị của Graph. - Ánh xạ tới thành phần HTML thông qua thuộc tính “el”. - Phải gắn model cần hiển thị qua thuộc tính “model”. 	<pre>var paper = new joint.dia.Paper({ el: \$('#paper'), width: 600, height: 200, model: graph, gridSize: 1 });</pre>
Rect	Biểu diễn hình chữ nhật.	<pre>var rect1 = new joint.shapes.basic.Rect({ position: { x: 100, y: 30 }, size: { width: 100, height: 30 }, attrs: { rect: { fill: '#009587' }, text: { text: 'Hello', fill: 'white' } } });</pre>
Circle	Biểu diễn hình tròn	<pre>var circle = new joint.shapes.basic.Circle({ position: { x: 50, y: 70 }, size: { width: 100, height: 40 } });</pre>
Link	Kết nối các đối tượng với nhau thông qua id của chúng.	<pre>var link = new joint.dia.Link({ source: { id: rect1.id }, target: { id: rect2.id } });</pre>
Text	Biểu diễn nhãn chữ	<pre>var text = new joint.shapes.basic.Text({ position: { x: 50, y: 70 }, size: { width: 100, height: 40 } });</pre>

- Cài đặt JointJS:

- Bước 1: Tải thư viện JointJS: <http://jointjs.com/download> .
- Bước 2: Sao chép tập tin .css vào thư mục chứa các tập tin css.
- Bước 3: Sao chép tập tin .js vào thư mục chứa các tập tin js.
- Bước 4: Chỉ 2 đường dẫn bên dưới đến đến trang html, jsp nào muốn sử dụng.

```
<link rel="stylesheet" href="joint.css" />
<script src="joint.js"></script>
```

- Ví dụ “Hello World!” sử dụng JointJS.
 - Chuẩn bị 1 thư mục như hình bên dưới



- Nội dung tập tin index.html

```

<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <title>Hello World</title>

  <!-- Chèn JointJS CSS -->
  <link rel="stylesheet" href="css/joint.css">
</head>
<body>
  <!-- Tạo nơi chứa trang vẽ trên HTML -->
  <div id="paper" style="width: 600px; height: 200px;"></div>

  <!-- Chèn JointJS Javascript -->
  <script type="text/javascript" src="js/joint.js"></script>

```

```

<script type="text/javascript">
  // Tạo một model chứa tất cả phần tử.
  var graph = new joint.dia.Graph;

  // Tạo một view hiển thị các phần tử của model graph và ánh xạ lên paper HTML.
  var paper = new joint.dia.Paper({
    el: $('#paper'),
    width: 600,
    height: 200,
    model: graph,
    gridSize: 1
  });

  /*
   Tạo một hình chữ nhật vị trí với thông tin:
   x = 100, y = 30, rộng = 100, cao = 30, màu = #009587, tên là Hello, chữ màu trắng.
  */
  var rect1 = new joint.shapes.basic.Rect({
    position: { x: 100, y: 30 },
    size: { width: 100, height: 30 },
    attrs: { rect: { fill: '#009587' }, text: { text: 'Hello', fill: 'white' } }
  });

  /*
   Tạo một hình chữ nhật vị trí với thông tin:
   x = 400, y = 30, rộng = 100, cao = 30, màu = #0769AD, tên là World, chữ màu trắng.
  */
  var rect2 = new joint.shapes.basic.Rect({
    position: { x: 400, y: 30 },
    size: { width: 100, height: 30 },
    attrs: { rect: { fill: '#0769AD' }, text: { text: 'World', fill: 'white' } }
  });

  // Tạo một kết nối từ rect1 đến rect2.
  var link = new joint.dia.Link({
    source: { id: rect1.id },
    target: { id: rect2.id }
  });

  // Chèn các phần tử vào model.
  graph.addCell([rect1, rect2, link]);
</script>
</body>
</html>

```

- Kết quả chạy ví dụ:



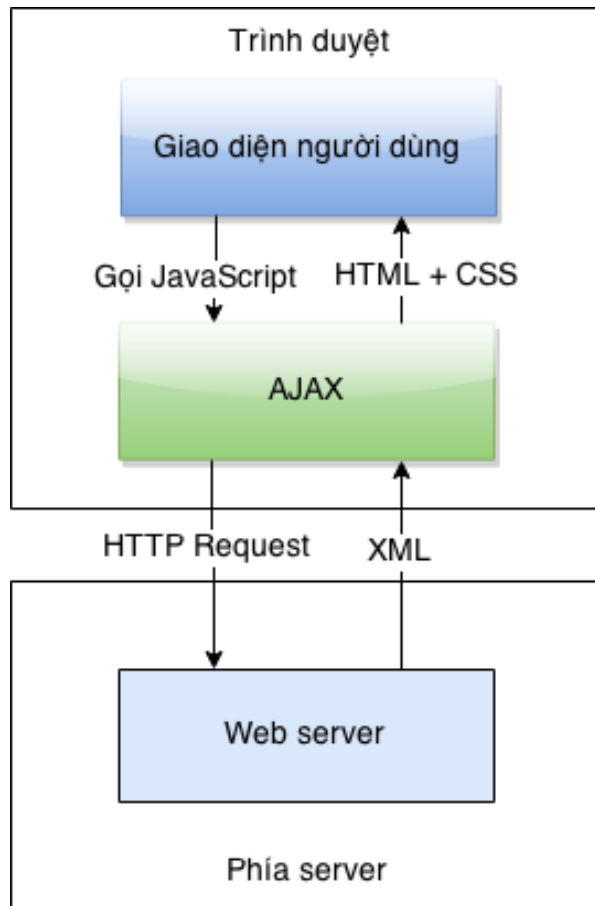
2.4.2. Cài đặt AJAX

- Ajax sử dụng để truyền và nhận dữ liệu giữa server với client. Khi người dùng thao tác trên client AJAX sẽ gửi dữ liệu đến một controller bên phía server.

Controller sẽ nhận dữ liệu và xử lý nó sau đó sẽ gửi dữ liệu phản hồi cho client.

- Sơ đồ truyền nhận dữ liệu AJAX:

Hình 20. Sơ đồ truyền nhận dữ liệu AJAX.



- Hàm AJAX sử dụng JQuery:

```
$.ajax({
  url: '/luanvan/dangnhap/',
  type: 'POST',
  data: 'taiKhoan='+taiKhoan+'&matKhau='+matKhau,
  success:function(result){
    $('#message').html("Đăng nhập thành công!");
  },
  error:function() {
    alert("Không thể kết nối tới server");
  }
});
```

- url: đường dẫn, hành động, thao tác cần thực hiện, gọi tới một controller.
- type: sử dụng loại GET hay POST dữ liệu.
- data: dữ liệu truyền đi.
- success: xử lý dữ liệu trả hay thông báo khi thực hiện thành công.
- error: xử lý khi thao tác thất bại.

2.4.3. Cài đặt Bootstrap

- Bootstrap sử dụng để cài đặt giao diện hiển thị. Với các thành phần được xây dựng sẵn ta có thể cài đặt các thành phần của trang web một cách nhanh chóng.
- Ví dụ về bootstrap, đoạn code để cài đặt một form:

```
<form>
  <div class="form-group">
    <label for="exampleInputEmail1">Email address</label>
    <input type="email" class="form-control" id="exampleInputEmail1" placeholder="Enter email">
  </div>
  <div class="form-group">
    <label for="exampleInputPassword1">Password</label>
    <input type="password" class="form-control" id="exampleInputPassword1" placeholder="Password">
  </div>
  <div class="form-group">
    <label for="exampleInputFile">File input</label>
    <input type="file" id="exampleInputFile">
    <p class="help-block">Example block-level help text here.</p>
  </div>
  <div class="checkbox">
    <label>
      <input type="checkbox"> Check me out
    </label>
  </div>
  <button type="submit" class="btn btn-default">Submit</button>
</form>
```

- Kết quả của đoạn code:

EXAMPLE

Email address

Password

File input

Chọn tệp

Không có tệp nào được chọn

Example block-level help text here.

☐ Check me out

2.4.4. Cài đặt Hibernate

- Hibernate sẽ mapping thực thể dữ liệu bằng một class trong java. Ví dụ class Actor mô tả thực thể ACTOR có quan hệ một nhiều với dự án và phân loại tương ứng với USECASE mà nó quan hệ.

Hình 21. Class Actor sử dụng hibernate.

```
@SuppressWarnings("serial")
@Entity
@Table(name = "actor", catalog = "luanvan")
public class Actor implements java.io.Serializable {

    private Integer actorid;
    private Loaiactor loaiactor;
    private String motaactor;
    private String nameofactor;
    private Project project;
    private Set<PhanLoai> phanloais = new HashSet<PhanLoai>();

    @OneToMany(fetch = FetchType.EAGER, mappedBy = "actor", cascade = CascadeType.REMOVE)
    public Set<PhanLoai> getPhanloais() {
        return this.phanloais;
    }

    public void setPhanloais(Set<PhanLoai> phanloais) {
        this.phanloais = phanloais;
    }

    @ManyToOne(fetch = FetchType.LAZY)
    @JoinColumn(name = "PROJECTID", nullable = false)
    public Project getProject() {
        return this.project;
    }

    public void setProject(Project project) {
        this.project = project;
    }
}
```

- Cài đặt hibernate:
 - o Thêm thư viện hibernate vào file pom.xml:

```
<!-- Hibernate ORM -->
<dependency>
    <groupId>org.hibernate</groupId>
    <artifactId>hibernate-core</artifactId>
    <version>4.3.7.Final</version>
</dependency>
```

- o Cấu hình tham chiếu đến CSDL:

```

<!-- MySQL data source -->
<beans:bean id="dataSource" class="org.apache.commons.dbcp.BasicDataSource">
  <beans:property name="driverClassName" value="com.mysql.jdbc.Driver" />
  <beans:property name="url" value="jdbc:mysql://localhost:3306/luanvan?autoReconnect=true&useUnicode=true&characterSetResults=UTF-8" />
  <beans:property name="username" value="root" />
  <!-- <beans:property name="password" value="123456aA@" /> -->
  <beans:property name="password" value="" />
</beans:bean>

```



○ Cấu hình hibernate trong file servlet-context.xml:

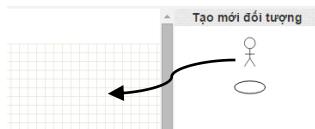
```

<!-- Hibernate session factory -->
<beans:bean id="sessionFactory" class="org.springframework.orm.hibernate4.LocalSessionFactoryBean">
  <beans:property name="dataSource" ref="dataSource" />
  <beans:property name="packagesToScan">
    <beans:array>
      <beans:value>vn.com.luanvan</beans:value>
    </beans:array>
  </beans:property>
  <beans:property name="hibernateProperties">
    <beans:props>
      <beans:prop key="hibernate.connection.useUnicode">true</beans:prop>
      <beans:prop key="hibernate.connection.characterEncoding">UTF-8</beans:prop>
      <beans:prop key="hibernate.connection.charset">UTF-8</beans:prop>
      <beans:prop key="hibernate.dialect">org.hibernate.dialect.MySQL5Dialect</beans:prop>
    </beans:props>
  </beans:property>
</beans:bean>

```

2.4.5. Ví dụ tạo một usecase trong ứng dụng

- Sau khi người dùng đăng nhập thành công vào hệ thống, truy cập đến chức năng vẽ sơ đồ usecase và đang thao tác trên một sơ đồ usecase.
- Để tạo usecase người dùng kéo thả đối tượng usecase  vào trang vẽ .



- Một AJAX sẽ được gọi và nó sẽ gửi dữ liệu data gồm có tên dự án, tên sơ đồ và tên usecase đến controller “Createusecase” với loại method là “post”. Sau khi thành công sẽ nhận chuỗi “id” trả về từ controller và gán nó vào đối tượng usecase sau đó vẽ lên sơ đồ thông qua graph.addCell(usecase).

```

$.ajax({
  url: "/luanvan/diagram/createusecase",
  type: "post",
  data: "nameProject=" + $("#nameProject").val() + "&nameDiagram=" + $("#name-diagram-show").html()
    + "&nameUsecase=" + nameUsecase,
  success: function(result) {
    usecase.attributes.created = result;
    graph.addCell(usecase);
    setIconSave("saved");
    listUsecase();
    setPathDiagram();
    setContentMessage("Sơ đồ đã thay đổi");
  },
  error: function() {
    alert("Không thể tạo use case. Xin vui lòng liên hệ ADMIN để giải quyết.");
  }
});

```

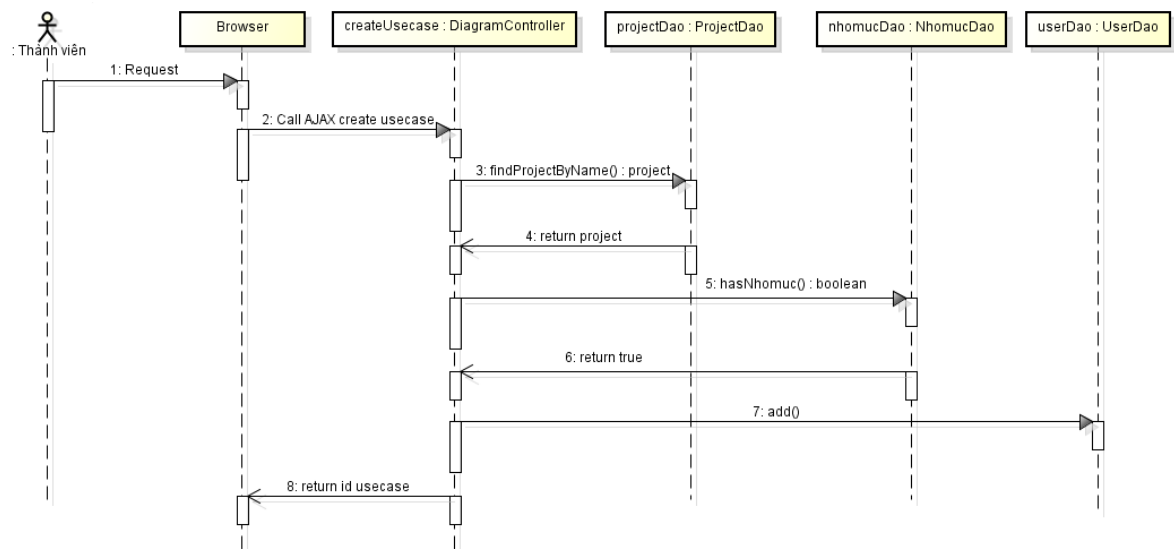
- Controller nhận dữ liệu và xử lý dữ liệu, đầu tiên controller sẽ lấy dự án hiện tại đang vẽ sơ đồ từ CSDL thông qua tên của dự án bằng cách sử dụng đối tượng `projectDao` và phương thức `findProjectByName(Người dùng, tên dự án)`. Sau đó sẽ tạo một đối tượng `Usecase` và cài đặt các thông tin lên đối tượng. Cuối cùng là sử dụng đối tượng `usecaseDao` và phương thức `add(Đối tượng usecase)` để lưu đối tượng `usecase` vào cơ sở dữ liệu và trả về id của đối tượng `usecase` đã tạo.

```
@RequestMapping(value = "/diagram/createusecase", method = RequestMethod.POST)
public @ResponseBody String createUsecase (@RequestParam(value="nameProject") String nameProject,
    @RequestParam(value="nameDiagram") String nameDiagram,
    @RequestParam(value="nameUsecase") String nameUsecase,
    Principal principal) {

    // Lay du an hien tai
    Project project = projectDao.findProjectByName(principal.getName(), nameProject);

    Usecase usecase = new Usecase();
    usecase.setBmt(bmtDao.getBmtById(1));
    usecase.setTinhtien(true);
    usecase.setNameofuc(nameUsecase);
    if (!nhomucDao.hasNhomuc(nameDiagram, project.getProjectid())) {
        Nhomuc nhomuc = new Nhomuc();
        nhomuc.setProject(project);
        nhomuc.setTennhom(nameDiagram);
        nhomucDao.add(nhomuc);
        usecase.setNhomuc(nhomuc);
    } else {
        usecase.setNhomuc(nhomucDao.getNhomucByName(nameDiagram, project.getProjectid()));
    }
    usecaseDao.add(usecase);
    return "" + usecase.getUsecaseid();
}
```

- Sơ đồ trình tự của chức năng tạo usecase:



CHƯƠNG 3 - KIỂM THỬ VÀ ĐÁNH GIÁ

Trình bày mục tiêu, kịch bản và đưa ra các đánh giá về kết quả của kiểm thử.

3.1. Mục tiêu kiểm thử

- Nhằm phát hiện lỗi của phần mềm.
- Chứng tỏ phần mềm thực hiện đúng các chức năng mong đợi, đúng với đặc tả yêu cầu đã đề ra.
- Nhằm xác lập độ tin cậy cái mà phần mềm muốn thực hiện có đúng với yêu cầu khách hàng đề ra hay không?
- Kiểm thử các chức năng: thiết kế sơ đồ và thiết kế giao diện có chạy được, chạy đúng và đạt các mục tiêu đề ra không?

3.2. Kịch bản kiểm thử

- Chuẩn bị môi trường kiểm thử:
 - o Hệ điều hành: Window 7 Professional 64-bit.
 - o Web server: Tomcat.
 - o Hệ quản trị cơ sở dữ liệu: MySQL.
 - o Trình duyệt: Mozilla FireFox 35.0.1, Google Chrome 41.0.2272.101 dev-m, Cốc Cốc 39.0.2171.103.
 - o Phần cứng: RAM 8 GB, bộ xử lý: Intel Pentium P6200 2.13GHz, dung lượng ổ đĩa còn tối thiểu 1 GB, có kết nối mạng.
- Kịch bản cho thiết kế sơ đồ:
 - o Truy cập vào địa chỉ: <http://localhost/luanvan>.
 - o Đăng nhập vào với tài khoản: “tinh123” và mật khẩu: “123456”.
 - o Sau khi đăng nhập thành công, trang danh sách các dự án sẽ được hiển thị.
 - o Click nút tạo dự án. Form tạo dự án hiển thị. Nhập tên dự án “Kiểm thử dự án”. Click nút tạo dự án. Một dự án “Kiểm thử dự án” sẽ được thêm vào danh sách dự án đang tiến hành.
 - o Chọn dự án “Kiểm thử dự án”. Trang chi tiết dự án sẽ được hiển thị với các chức năng có thể lựa chọn: mô tả yêu cầu, thiết kế sơ đồ, thiết kế giao diện ...
 - o Chọn chức năng vẽ sơ đồ usecase. Danh sách các sơ đồ sẽ được hiển thị.
 - o Chọn tạo mới sơ đồ. Hệ thống chuyển sang trang vẽ sơ đồ. Click nút tạo mới sơ đồ form tạo mới được hiển thị, nhập tên sơ đồ: “Sơ đồ

kiểm thử”. Click nút tạo, một sơ đồ được hiện ra với tên là “Sơ đồ kiểm thử”.

- Kéo thả đối tượng tác nhân từ thanh công cụ bên phải để tạo một tác nhân mới. Một tác nhân tên “Actor” được tạo ra trên trang vẽ và thanh công cụ bên trái danh sách “Actor đã có” sẽ có thêm một phần tử là “Actor”.
 - Kéo thả thêm một đối tượng tác nhân từ thanh công cụ bên phải để tạo thêm một tác nhân. Một tác nhân tên “Actor:1” được tạo ra trên trang vẽ và thanh công cụ bên trái danh sách “Actor đã có” sẽ có thêm một phần tử là “Actor:1”.
 - Nhấn đôi vào tên “Actor:1” và thay đổi tên thành “Kiểm thử viên”. Thanh công cụ bên trái danh sách “Actor đã có” phần tử “Actor:1” sẽ đổi thành “Kiểm thử viên”.
 - Nhấn đôi vào hình vẽ “Kiểm thử viên”. Một form cập nhật thông tin có tên “Tác nhân: Kiểm thử viên” sẽ được hiện ra. Ta thay đổi thông tin của “Kiểm thử viên” sau đó nhấn nút lưu.
 - Ta tiến hành làm tương tự đối với đối tượng use-case để tạo đối tượng use-case tên là “Chức năng kiểm thử”.
 - Nhấp chọn “Kiểm thử viên” các lựa chọn quan hệ được hiển thị, lựa chọn quan hệ “Association” kéo qua “Chức năng kiểm thử”. Một đường thẳng nối từ “Kiểm thử viên” đến “Chức năng kiểm thử”.
 - Nhấn đôi vào “Kiểm thử viên”, thông tin “Vai trò với chức năng” đã có “Chức năng kiểm thử”.
 - Nhấn đôi vào “Chức năng kiểm thử”, thông tin “Tác nhân sử dụng” đã có tác nhân “Kiểm thử viên”.
 - Nhấn đôi vào “Kiểm thử viên” sau đó nhấn vào “x”. Đối tượng “Kiểm thử viên” sẽ biến mất khỏi trang vẽ nhưng trên thanh công cụ bên trái vẫn còn phần tử “Kiểm thử viên”.
 - Nhấn đôi vào “Chức năng kiểm thử” sau đó nhấn nút “Delete” trên bàn phím. Giao diện lựa chọn thao tác sẽ được hiển thị. Chọn “Xóa hoàn toàn khỏi dự án” sau đó nhấn nút “Xóa”. Đối tượng “Chức năng kiểm thử” sẽ biến mất khỏi trang vẽ và biến mất trên thanh công cụ bên trái.
- Kịch bản cho thiết kế giao diện:
- Truy cập vào địa chỉ: <http://localhost/luanvan>.
 - Đăng nhập vào với tài khoản: “tinh123” và mật khẩu: “123456”.

- Sau khi đăng nhập thành công, trang danh sách các dự án sẽ được hiển thị.
- Chọn dự án “Kiểm thử dự án”. Trang chi tiết dự án sẽ được hiển thị với các chức năng có thể lựa chọn: mô tả yêu cầu, thiết kế sơ đồ, thiết kế giao diện ...
- Chọn chức năng thiết kế giao diện. Danh sách các giao diện sẽ được hiển thị.
- Chọn giao diện “Giao diện kiểm thử”. Trang thiết kế giao diện hiện ra.
- Kéo thả đối tượng nút nhấn từ thanh công cụ bên phải để tạo một nút nhấn mới mới. Một nút nhấn tên “Button” được tạo ra trên trang vẽ.
- Nhấn đôi vào tên “Button” và thay đổi tên thành “Đăng nhập”.
- Chỉnh sửa thuộc tính trên thanh thuộc tính bên trái.
- Ta tiến hành làm tương tự đối với các đối tượng còn lại bên thanh công cụ bên phải.
- Ta di chuyển các đối tượng trên trang vẽ.
- Nhấn nút lưu để lưu lại giao diện.

3.3. Kết quả kiểm thử

- Các chức năng thực hiện mượt mà, chính xác.
- Các chức năng thực hiện đúng với yêu cầu của bài toán.

PHẦN III - KẾT LUẬN

KẾT QUẢ ĐẠT ĐƯỢC

Trình bày những thành quả thu được về kiến thức, kỹ năng, ứng dụng của phần mềm và những hạn chế phần mềm cần khắc phục.

- Về kiến thức và kỹ năng:

- Nâng cao kiến thức và kỹ năng về các ngôn ngữ: JAVA, HTML5, CSS3, JAVASCRIPT, JQUERY, JSON, SVG, JSP.
- Hiểu rõ hơn về quy trình xây dựng một ứng dụng trên nền web.
- Hiểu rõ hơn về mô hình MVC và Framework Spring Web MVC.
- Sử dụng tốt thư viện vẽ trên nền web và ứng dụng nó để xây dựng thành một chức năng hoàn chỉnh.
- Áp dụng được AJAX vào ứng dụng web.
- Hiểu rõ quy trình cách thức ước lượng chi phí của một phần mềm theo công văn 2589/BTTTT-UDCNTT.
- Nâng cao được kỹ năng làm việc nhóm.
- Nâng cao kỹ năng quản lý thời gian, dự án phần mềm.

- Về ứng dụng:

- Giao diện thiết kế dễ sử dụng, màu sắc hài hòa, bố cục hợp lí.
- Giao diện có thể thay đổi phù hợp với nhiều loại màn hình khác nhau: điện thoại, máy tính bảng, laptop.
- Hoàn thành 100% chức năng được đề ra, các chức năng sử dụng dễ dàng và có sự liên kết với nhau.
- Có thể sử dụng phần mềm để xác định đúng giá trị của phần mềm theo công văn 2589/BTTTT-UDCNTT.
- Có thể sử dụng phần mềm để thay thế công cụ vẽ sơ đồ use-cases.
- Có thể sử dụng phần mềm để thay thế công cụ thiết kế giao diện.

- Hạn chế:

- Chưa cho phép người dùng chỉnh sửa thuộc tính của các đối tượng vẽ use-cases.
- Các thuộc tính có thể thay đổi ở các đối tượng vẽ phân thiết kế giao diện còn quá ít.

HƯỚNG PHÁT TRIỂN

- Cải thiện các xử lý ở chức năng thiết kế giao diện và vẽ sơ đồ use-cases.
- Cho phép người dùng chỉnh sửa thuộc tính của các đối tượng vẽ use-cases.
- Cung cấp thêm các thuộc tính có thể thay đổi trên các đối tượng vẽ phần thiết kế giao diện.
- Cho phép nhiều người cùng thao tác trên một sơ đồ vẽ hoặc trên một bảng thiết kế giao diện.
- Từ giao diện thiết kế sinh code HTML, JAVA.

TÀI LIỆU THAM KHẢO

[1] AJAX:

<http://api.jquery.com/jquery.ajax/>

http://vi.wikipedia.org/wiki/Ajax_%28l%E1%BA%ADp_tr%C3%ACnh%29

http://www.w3schools.com/jquery/ajax_ajax.asp

[2] BOOTSTRAP:

<http://getbootstrap.com/>

<http://www.w3schools.com/bootstrap/default.asp>

[3] CSS:

<http://www.w3schools.com/css/default.asp>

[4] HTML:

<http://www.w3schools.com/html/default.asp>

[5] JAVASCRIPT:

<http://www.w3schools.com/js/default.asp>

[6] JOINTJS:

<http://www.jointjs.com/>

<http://stackoverflow.com/questions/tagged/jointjs>

<https://groups.google.com/forum/#!forum/jointjs>

[7] JQUERY:

<http://www.w3schools.com/jquery/default.asp>

[8] SPRING Web MVC:

http://www.tutorialspoint.com/spring/spring_web_mvc_framework.htm

HƯỚNG DẪN CÀI ĐẶT VÀ SỬ DỤNG

- Chuẩn bị:

- Cài đặt Tomcat: sử dụng Tomcat 7.
- Tập tin “.sql” cơ sở dữ liệu.
- Tập tin “.war” của dự án.

- Sau khi cài đặt và chuẩn bị ta deploy file war lên tomcat:

- Truy cập server: <http://diachiip:8080/manager/html>
- Chọn file war sau đó nhấn chọn deployer.

Choisir le fichier WAR à téléverser LuanVanVersion5.war

- Khi deploy thành công sẽ hiển thị

Applications	
Chemin	Version
/	None specified
<u>/LuanVanVersion5</u>	None specified

- Truy cập ứng dụng: <http://diachiip:8080/LuanVanVersion5>