

BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC DÂN LẬP HẢI PHÒNG

-----o0o-----



ĐỒ ÁN TỐT NGHIỆP

NGÀNH CÔNG NGHỆ THÔNG TIN

HẢI PHÒNG 2013

BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC DÂN LẬP HẢI PHÒNG

-----o0o-----

HỆ THỐNG QUẢN LÝ PHIÊN BẢN VĂN BẢN

ĐỒ ÁN TỐT NGHIỆP ĐẠI HỌC HỆ CHÍNH QUY

Ngành: Công nghệ Thông tin

BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC DÂN LẬP HẢI PHÒNG

-----o0o-----

HỆ THỐNG QUẢN LÝ PHIÊN BẢN VĂN BẢN

ĐỒ ÁN TỐT NGHIỆP ĐẠI HỌC HỆ CHÍNH QUY

Ngành: Công nghệ Thông tin

Sinh viên thực hiện: Phạm Trung Mạnh

Giáo viên hướng dẫn: ThS Nguyễn Trịnh Đông

Mã số sinh viên: 1351010013

BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC DÂN LẬP HẢI PHÒNG

CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM

Độc lập - Tự do - Hạnh phúc

-----o0o-----

NHIỆM VỤ THIẾT KẾ TỐT NGHIỆP

Sinh viên: Phạm Trung Mạnh

Mã SV: 1351010013

Lớp: CT1301

Ngành: Công nghệ Thông tin

Tên đề tài: Hệ thống quản lý phiên bản văn bản

NHIỆM VỤ ĐỀ TÀI

1. Nội dung và các yêu cầu cần giải quyết trong nhiệm vụ đề tài tốt nghiệp

a. Nội dung

- Nắm được các khái niệm cơ bản về quản lý phiên bản
- Nắm được các phương pháp trong quản lý phiên bản
- Cài đặt, thử nghiệm chương trình.
- Nghiêm túc thực hiện các nhiệm vụ và nội dung giáo viên hướng dẫn.

b. Các yêu cầu cần giải quyết

Các phần mềm cần thiết để xây dựng

- Web Server: Xampp
- Hệ quản trị cơ sở dữ liệu MySQL.

CÁN BỘ HƯỚNG DẪN ĐỀ TÀI TỐT NGHIỆP

Người hướng dẫn thứ nhất:

Họ và tên: Nguyễn Trịnh Đông

Học hàm, học vị: Thạc Sĩ

Cơ quan công tác: Trường Đại Học Dân Lập Hải Phòng

Nội dung hướng dẫn:

.....

.....

.....

Người hướng dẫn thứ hai:

Họ và tên:

Học hàm, học vị:

Cơ quan công tác:

Nội dung hướng dẫn:

.....

.....

.....

Đề tài tốt nghiệp được giao ngày tháng năm 2013

Yêu cầu phải hoàn thành trước ngày tháng năm 2013

Đã nhận nhiệm vụ: Đ.T.T.N

Đã nhận nhiệm vụ: Đ.T.T.N

Sinh viên

Cán bộ hướng dẫn Đ.T.T.N

ThS Nguyễn Trịnh Đông

Hải Phòng, ngàytháng.....năm 2013

HIỆU TRƯỞNG

GS.TS.NGŨT Trần Hữu Nghị

PHẦN NHẬN XÉT TÓM TẮT CỦA CÁN BỘ HƯỚNG DẪN

1. Tinh thần thái độ của sinh viên trong quá trình làm đề tài tốt nghiệp:

.....
.....
.....
.....
.....
.....

2. Đánh giá chất lượng của đề tài tốt nghiệp (so với nội dung yêu cầu đã đề ra trong nhiệm vụ đề tài tốt nghiệp)

.....
.....
.....
.....
.....
.....
.....

3. Cho điểm của cán bộ hướng dẫn:
(Điểm ghi bằng số và chữ)

.....
.....

Ngày.....tháng.....năm 2013

Cán bộ hướng dẫn chính

(Ký, ghi rõ họ tên)

PHẦN NHẬN XÉT ĐÁNH GIÁ CỦA CÁN BỘ CHẤM

PHẢN BIỆN ĐỀ TÀI TỐT NGHIỆP

1. Đánh giá chất lượng đề tài tốt nghiệp (về các mặt như cơ sở lý luận, thuyết minh chương trình, giá trị thực tế, ...)

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

2. Cho

điểm của cán bộ phản biện
(*Điểm ghi bằng số và chữ*)

.....

.....

Ngày.....tháng.....năm 2013

Cán bộ chấm phản biện
(Ký, ghi rõ họ tên)

MỤC LỤC

MỤC LỤC	3
DANH MỤC HÌNH ẢNH	5
DANH MỤC CÁC BẢNG	8
MỘT SỐ THUẬT NGỮ	9
LỜI CẢM ƠN	10
LỜI NÓI ĐẦU	11
CHƯƠNG 1: HỆ THỐNG QUẢN LÝ PHIÊN BẢN	12
1.1. Quản lý phiên bản văn bản trên thế giới và ở Việt Nam.....	12
1.2. Các công cụ hỗ trợ quản lý phiên bản hiện nay	12
1.3. Giới thiệu.....	13
1.4. Lịch sử phát triển	14
1.5. Các khái niệm và thuật ngữ quan trọng	14
1.5.1. Nơi lưu trữ	14
1.5.2. Thư mục làm việc	15
1.5.3. Khóa.....	15
1.5.4. Dự án, module và tập tin.....	15
1.5.5. Phân nhánh.....	16
1.5.6. Ghép tập tin.....	16
1.5.7. Phiên bản.....	16
1.6. Kiến trúc của hệ thống quản lý phiên bản.....	17
1.6.1. Hệ thống quản lý phiên bản cục bộ	17
1.6.2. Hệ thống quản lý phiên bản tập trung.....	18
1.6.3. Hệ thống quản lý phiên bản phân tán	19
1.6.4. Các thao tác của hệ thống quản lý phiên bản tập trung	20
1.6.5. Các thao tác của hệ thống quản lý phiên bản phân tán.....	28

1.7. Các phương thức xử lý	29
1.7.1. Vấn đề chia sẻ tập tin.....	29
1.7.2. Các giải pháp khóa – sửa đổi – mở khóa.....	30
1.7.3. Các giải pháp sao chép – chỉnh sửa – hợp nhất.....	31
CHƯƠNG 2: PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG.....	34
2.1. Tổng quan về hệ thống quản lý phiên bản văn bản.....	34
2.2. Bài toán quản lý phiên bản văn bản	35
2.2.1. Mô tả bài toán	35
2.2.2. Xác định mô hình nghiệp vụ.....	35
2.2.3. Phân tích hệ thống	49
2.2.4. Thiết kế hệ thống	75
CHƯƠNG 3. XÂY DỰNG CHƯƠNG TRÌNH.....	79
3.1. Giới thiệu ứng dụng	79
3.1.1. Thông tin kỹ thuật.....	79
3.1.2. Kết quả đạt được	79
3.2. Giao diện của chương trình.....	79
3.2.1. Giao diện đăng nhập	79
3.2.2. Giao diện quản lý	80
3.2.3. Giao diện thêm tài liệu.....	81
3.2.4. Giao diện thêm người dùng	82
3.2.5. Giao diện thêm danh mục và bộ phận	83
3.2.6. Giao diện tìm kiếm	83
3.2.7. Checkout tệp tin	84
3.2.8. Checkin tệp tin	84
3.2.9. Xem lịch sử tệp tin.....	85
KẾT LUẬN	86
TÀI LIỆU THAM KHẢO	87

DANH MỤC HÌNH ẢNH

Hình 1.1: Sơ đồ hệ thống quản lý phiên bản cục bộ	17
Hình 1.2 : Sơ đồ hệ thống quản lý phiên bản tập trung.....	18
Hình 1.3: Sơ đồ hệ thống quản lý phiên bản phân tán	19
Hình 1.4 : Tạo mới một kho lưu trữ rỗng.....	20
Hình 1.5: Tạo một bản sao làm việc.....	21
Hình 1.6 : Áp dụng các thay đổi trong bản sao với kho như thay đổi mới.	23
Hình 1.8 :Cập nhật các bản sao làm việc đối với các kho lưu trữ.....	23
Hình 1.8 : Hệ thống quản lý phiên bản tập trung	28
Hình 1.9 : Hệ thống quản lý phiên bản phân tán.....	28
Hình 1.10 : Thao tác đẩy	29
Hình 1.11 : Thao tác lấy	29
Hình 1.12 : Vấn đề cần tránh.....	30
Hình 1.13: Giải pháp khóa-sửa đổi-mở khóa	30
Hình 1.14 : Giải pháp sao chép-chỉnh sửa-hợp nhất	32
Hình 1.15 : Giải pháp bản sao-chỉnh sửa-hợp nhất (tiếp theo)	32
Hình 2.1 : Kiến trúc của một ứng dụng quản lý phiên bản	34
Hình 2.2 : Mô hình ca sử dụng mức tổng quát.....	38
Hình 2.3: Biểu đồ ca sử dụng gói “Quản lý bộ phận”	39
Hình 2.4 : Biểu đồ ca sử dụng gói “Quản lý người dùng”	41
Hình 2.5 : Biểu đồ ca sử dụng gói “Quản lý danh mục”	43
Hình 2.6 : Biểu đồ ca sử dụng gói “Quản lý tài liệu”.....	45
Hình 2.7 : Biểu đồ tuần tự thực thi ca sử dụng “Thêm bộ phận”	49
Hình 2.8 : Biểu đồ cộng tác thực thi ca sử dụng “Thêm bộ phận”.....	49
Hình 2.9 : Biểu đồ tuần tự thực thi ca sử dụng “Tìm kiếm bộ phận”	50
Hình 2.10 : Biểu đồ cộng tác thực thi ca sử dụng “Tìm kiếm bộ phận”	50
Hình 2.11 : Biểu đồ tuần tự thực thi ca sử dụng “Cập nhật bộ phận”.....	51
Hình 2.12 : Biểu đồ cộng tác thực thi ca sử dụng “Cập nhật bộ phận”	51
Hình 2.13 : Biểu đồ cộng tác thực thi ca sử dụng “Xóa bộ phận”	52

Hình 2.14 : Biểu đồ cộng tác thực thi ca sử dụng “Xóa bộ phận”	52
Hình 2.15 : Mô hình phân tích gói ca “Quản lý bộ phận”	53
Hình 2.16 : Biểu đồ tuần tự thực thi ca sử dụng “Thêm danh mục”	54
Hình 2.17 : Biểu đồ cộng tác thực thi ca sử dụng “Thêm danh mục”	54
Hình 2.18 : Biểu đồ tuần tự thực thi ca sử dụng “Tìm kiếm danh mục”	55
Hình 2.19 : Biểu đồ cộng tác thực thi ca sử dụng “Tìm kiếm danh mục”	55
Hình 2.20 : Biểu đồ tuần tự thực thi ca sử dụng “Cập nhật danh mục”	56
Hình 2.21 : Biểu đồ cộng tác thực thi ca sử dụng “Cập nhật danh mục”	56
Hình 2.22 : Biểu đồ tuần tự thực thi ca sử dụng “Xóa danh mục”	57
Hình 2.24 : Mô hình phân tích gói ca “Quản lý danh mục”	58
Hình 2.25 : Biểu đồ tuần tự thực thi ca sử dụng “Thêm người dùng”	59
Hình 2.26 : Biểu đồ cộng tác thực thi ca sử dụng “Thêm người dùng”	59
Hình 2.27 : Biểu đồ tuần tự thực thi ca sử dụng “Tìm kiếm người dùng”	60
Hình 2.28 : Biểu đồ cộng tác thực thi ca sử dụng “Tìm kiếm người dùng”	60
Hình 2.29 : Biểu đồ tuần tự thực thi ca sử dụng “Cập nhật người dùng”	61
Hình 2.30 : Biểu đồ cộng tác thực thi ca sử dụng “Cập nhật người dùng”	61
Hình 2.31 : Biểu đồ tuần tự thực thi ca sử dụng “Xóa người dùng”	62
Hình 2.32 : Biểu đồ cộng tác thực thi ca sử dụng “Xóa người dùng”	62
Hình 2.33 : Mô hình phân tích gói ca “Quản lý người dùng”	63
Hình 2.34 : Biểu đồ tuần tự thực thi ca sử dụng “Thêm tài liệu”	63
Hình 2.35 : Biểu đồ cộng tác thực thi ca sử dụng “Thêm tài liệu”	64
Hình 2.36 : Biểu đồ tuần tự thực thi ca sử dụng “Tìm kiếm tài liệu”	64
Hình 2.37 : Biểu đồ cộng tác thực thi ca sử dụng “Tìm kiếm tài liệu”	65
Hình 2.38 : Biểu đồ tuần tự thực thi ca sử dụng “Cập nhật tài liệu”	65
Hình 2.39 : Biểu đồ cộng tác thực thi ca sử dụng “Cập nhật tài liệu”	66
Hình 2.40 : Biểu đồ tuần tự thực thi ca sử dụng “Xóa tài liệu”	66
Hình 2.41 : Biểu đồ cộng tác thực thi ca sử dụng “Xóa tài liệu”	67
Hình 2.42 : Biểu đồ tuần tự thực thi ca sử dụng “Checkout tài liệu”	67
Hình 2.43 : Biểu đồ cộng tác thực thi ca sử dụng “Checkout tài liệu”	68

Hình 2.44 : Biểu đồ tuần tự thực thi ca sử dụng “Checkin tài liệu”.....	68
Hình 2.45 : Biểu đồ cộng tác thực thi ca sử dụng “Checkin tài liệu”	69
Hình 2.46 : Biểu đồ tuần tự thực thi ca sử dụng “Xem lịch sử tài liệu”	69
Hình 2.47 : Biểu đồ cộng tác thực thi ca sử dụng “Xem lịch sử tài liệu”	70
Hình 2.48 : Biểu đồ tuần tự thực thi ca sử dụng “Phục hồi tài liệu”	70
Hình 2.49 : Biểu đồ cộng tác thực thi ca sử dụng “Phục hồi tài liệu”	71
Hình 2.50 : Biểu đồ tuần tự thực thi ca sử dụng “Xét duyệt tài liệu”	71
Hình 2.51 : Biểu đồ cộng tác thực thi ca sử dụng “Xét duyệt tài liệu”	72
Hình 2.52 : Biểu đồ tuần tự thực thi ca sử dụng “Hủy trạng thái checkout” ...	72
Hình 2.53 : Biểu đồ cộng tác thực thi ca sử dụng “Hủy trạng thái checkout” .	73
Hình 2.54 : Mô hình phân tích gói ca “Quản lý tài liệu”	74
Hình 2.55 : Mô hình lớp thiết kế gói ca sử dụng “Quản lý bộ phận”	75
Hình 2.56 : Mô hình lớp thiết kế gói ca sử dụng “Quản lý danh mục”	75
Hình 2.57 : Mô hình lớp thiết kế gói ca sử dụng “Quản lý người dùng”	76
Hình 2.58 : Mô hình lớp thiết kế gói ca sử dụng “Quản lý tài liệu”	77
Hình 2.59 : Mô hình thiết kế cơ sở dữ liệu.....	78
Hình 3.1: Giao diện đăng nhập.....	79
Hình 3.2: Giao diện khi đăng nhập thành công.....	80
Hình 3.3: Giao diện quản lý	80
Hình 3.4: Giao diện thêm tài liệu	81
Hình 3.5: Giao diện khi thêm tài liệu thành công	81
Hình 3.6: Giao diện thêm người dùng.....	82
Hình 3.7: Giao diện thêm bộ phận	83
Hình 3.8: Giao diện thêm danh mục.....	83
Hình 3.9: Giao diện tìm kiếm.....	83
Hình 3.10: Giao diện checkout tập tin.....	84
Hình 3.11: Giao diện checkin tập tin.....	84
Hình 3.12: Giao diện xem lịch sử tập tin	85

DANH MỤC CÁC BẢNG

Bảng 1.1 : Ba thể hệ của quản lý phiên bản	14
Bảng 2.1 : Các chức năng nghiệp vụ.....	36
Bảng 2.2 : Các tác nhân hệ thống.....	36
Bảng 2.3 : Bảng biểu diễn mô hình hệ thống nghiệp vụ	37
Bảng 2.4 : Ca sử dụng “Thêm bộ phận”	39
Bảng 2.5 : Ca sử dụng “Tìm kiếm bộ phận”	40
Bảng 2.6 : Ca sử dụng “Sửa bộ phận”	40
Bảng 2.7 : Ca sử dụng “Xóa bộ phận”	40
Bảng 2.8 : Ca sử dụng “Thêm người dùng”	41
Bảng 2.9 : Ca sử dụng “Tìm kiếm người dùng”	42
Bảng 2.10 : Ca sử dụng “Cập nhật TT người dùng”	42
Bảng 2.11 : Ca sử dụng “Xóa hồ sơ”	42
Bảng 2.12 : Ca sử dụng “Thêm danh mục”.....	43
Bảng 2.13 : Ca sử dụng “Tìm kiếm danh mục”	44
Bảng 2.14 : Ca sử dụng “Sửa danh mục”	44
Bảng 2.15 : Ca sử dụng “Xóa danh mục”	44
Bảng 2.16 : Ca sử dụng “Thêm tài liệu”	45
Bảng 2.17 : Ca sử dụng “Tìm kiếm tài liệu”	46
Bảng 2.18 : Ca sử dụng “Cập nhật tài liệu”	46
Bảng 2.19 : Ca sử dụng “Xóa tài liệu”	46
Bảng 2.20 : Ca sử dụng “Checkout tài liệu”	47
Bảng 2.21 : Ca sử dụng “Checkin tài liệu”	47
Bảng 2.22 : Ca sử dụng “Xem lịch sử tài liệu”	47
Bảng 2.23 : Ca sử dụng “Phục hồi tài liệu”.....	48
Bảng 2.24 : Ca sử dụng “Xét duyệt tài liệu”	48
Bảng 2.25 : Ca sử dụng “Hủy trạng thái checkout”	48

MỘT SỐ THUẬT NGỮ

STT	Thuật ngữ	Diễn giải
1	Version Control System (VCS)	Hệ thống quản lý phiên bản.
2	Centralized Version Control System(CVCS)	Hệ thống quản lý phiên bản tập trung.
3	Distributed Version Control System (DVCS)	Hệ thống quản lý phiên bản phân tán.
4	Local Version Control Systems (LVCS)	Hệ thống quản lý phiên bản cục bộ.
5	Concurrent Versioning System(CVS)	Hệ thống phiên bản đồng thời.
6	Checkout	Lấy một tập tin từ kho chứa ra làm việc.
7	Checkin	Cập nhật một tập tin vào kho lưu trữ khi checkout.
8	Commit	Giống commit.

LỜI CẢM ƠN

Trước hết em xin bày tỏ tình cảm và lòng biết ơn thầy Nguyễn Trịnh Đông, người đã dành cho em rất nhiều thời gian quý báu, trực tiếp hướng dẫn tận tình giúp đỡ, chỉ bảo em trong suốt quá trình làm đồ án tốt nghiệp.

Em xin chân thành cảm ơn tất cả các thầy cô giáo trong Khoa Công nghệ Thông tin Trường Đại Học Dân Lập Hải Phòng, chân thành cảm ơn các thầy giáo, cô giáo tham gia giảng dạy và truyền đạt những kiến thức quý báu trong suốt thời gian em học tập tại trường, đã đọc và phản biện đồ án của em giúp em hiểu rõ hơn các vấn đề mình nghiên cứu, để em có thể hoàn thành đồ án này.

Tuy có nhiều cố gắng trong quá trình học tập, trong thời gian thực tập cũng như trong quá trình làm đồ án nhưng không thể tránh khỏi những thiếu sót, em rất mong được sự góp ý quý báu của tất cả các thầy giáo, cô giáo cũng như tất cả các bạn để kết quả của em được hoàn thiện hơn.

Em xin chân thành cảm ơn.

Hải Phòng, tháng 12 năm 2012

Sinh viên

Phạm Trung Mạnh

LỜI NÓI ĐẦU

Các hệ thống thông tin trong các tổ chức ngày càng phức tạp, hệ thống văn bản được chuẩn hóa và quản lý theo các quy trình tiên tiến. Các tổ chức nhà nước từ thấp đến cao, từ các công ty nhỏ đến lớn, từ các trung tâm nghiên cứu đến các doanh nghiệp sản xuất và cả các công ty phát triển phần mềm đều cần có hệ thống quản lý văn bản. Các văn bản được biên tập qua nhiều giai đoạn, mỗi lần sửa đổi đó là một phiên bản khác nhau. Việc quản lý các phiên bản rất phức tạp và đa dạng. Riêng đối với các công ty phần mềm, ngoài các phiên bản về văn bản hành chính còn cần quản lý các phiên bản về mã nguồn phần mềm. Mã nguồn phần mềm khi chưa biên dịch cũng là các văn bản. Việc phân bổ các công việc cho lập trình viên lập trình và thu về các mã nguồn rồi đồng bộ chúng là công việc khó nhất của những người đóng vai trò trưởng nhóm phần mềm hoặc của người quản trị dự án.

Việc tìm hiểu và nghiên cứu xây dựng một hệ thống quản lý phiên bản là công việc phức tạp nhưng rất lý thú và cần thiết. Xuất phát từ yêu cầu của từng tổ chức thì tính năng của các hệ thống này cũng khác nhau, nhưng tựu chung vẫn dựa trên những chức năng xương sống như: tạo, phân phối, cập nhật, đồng bộ phiên bản,...

Từ hướng tiếp cận trên, em đã tiến hành thực hiện đề tài “**Hệ thống quản lý phiên bản văn bản**”. Nhằm mục đích tìm hiểu kiến trúc, quy trình xây dựng hệ thống quản lý phiên bản và các chức năng của chúng. Trong khóa luận này em đã tìm hiểu các kiến thức liên quan và thử nghiệm xây dựng một hệ thống quản lý phiên bản văn bản dựa trên nền Web.

Khóa luận được tổ chức thành các phần như sau:

Chương 1: *Trình bày về kiến trúc của hệ thống quản lý phiên bản và các khái niệm liên quan.*

Chương 2: *Phân tích, thiết kế hệ thống quản lý phiên bản văn bản.*

Chương 3: *Trình bày chương trình ứng dụng quản lý phiên bản văn bản.*

Kết luận

Tài liệu tham khảo

CHƯƠNG 1: HỆ THỐNG QUẢN LÝ PHIÊN BẢN

1.1. Quản lý phiên bản văn bản trên thế giới và ở Việt Nam

Quản lý phiên bản (*control version*) trong phát triển phần mềm cũng như trong các ứng dụng quản lý phiên bản văn bản của các tổ chức đóng vai trò quan trọng nếu không nói là quan trọng nhất. Vào giai đoạn mới hình thành tiến trình quản lý các phiên bản văn bản chủ yếu được thực hiện đơn lẻ bởi các tổ chức. Tuy nhiên, hầu hết các tổ chức đều nhận thấy công việc quản lý sự thay đổi của văn bản là vô cùng phức tạp và tốn kém. Chính vì vậy, đã xuất hiện các công ty chuyên cung cấp các giải pháp chọn gói về quản lý phiên bản và các công cụ để hỗ trợ cho tiến trình này. Ví dụ: *Subversion, Tortoissvn, LogicalDOC, KnowledgeTree ...*

Trên thế giới, các công ty phần mềm đã tiến hành xây dựng tiến trình quản lý phiên bản riêng của mình nhờ vào kinh nghiệm phát triển của chính họ và sự tư vấn của các chuyên gia từ các viện nghiên cứu phần mềm và phát triển các công cụ riêng phục vụ cho tiến trình này. Với các công ty nhớt vốn và kinh nghiệm thì mua các công cụ sẵn và áp dụng linh hoạt các quy trình chuẩn được công bố vào quá trình phát triển phần mềm.

Ở Việt Nam, nền công nghiệp phần mềm vẫn chưa thực sự phát triển mạnh, phần lớn là các công ty nhỏ chuyên thực hiện các đề án về tin học hóa cho các cơ quan, xí nghiệp hoặc gia công phần mềm cho nước ngoài và quá trình phát triển phần mềm chỉ dựa trên kinh nghiệm với những phần mềm nhỏ và việc quản lý, kiểm soát tiến trình phát triển phần mềm ít được để ý. Thời gian gần đây, các công ty nhỏ đang dần dần tự cải tiến công nghệ để có thể thực hiện được những đề án phần mềm lớn trong và ngoài nước. Tuy nhiên, làm cách nào để có thể dùng tốt các công cụ và giải pháp hiện có vào hoàn cảnh thực tế ở Việt Nam là một điều không dễ dàng chút nào.

1.2. Các công cụ hỗ trợ quản lý phiên bản hiện nay

Hiện tại, các công cụ hỗ trợ cho việc quản lý phiên bản xuất hiện ngày càng nhiều. Một số công cụ đã xuất hiện từ lâu và được hoàn thiện dần nên có chất lượng rất tốt. Chẳng hạn như các công cụ quản lý phiên bản *Subversion, Tortoissvn, Git, Bazaar, Mercurial, KnowledgeTree ...* Các phần mềm này đều là mã nguồn mở, vì vậy việc cài đặt là khá dễ dàng.

1.3. Giới thiệu

Một hệ thống kiểm soát phiên bản là một phần mềm giúp các nhà phát triển trên một nhóm phần mềm làm việc cùng nhau và cũng có thể lưu trữ toàn bộ lịch sử của công việc. Hệ thống có khả năng tự cập nhật, so sánh và kết hợp tài nguyên mới vào tài nguyên cũ.

Hệ thống quản lý phiên bản giải quyết được các vấn đề

- Khi một nhóm làm việc trên cùng một dự án, việc nhiều người cùng chỉnh sửa nội dung của một tập tin là điều không thể tránh khỏi. hệ thống quản lý phiên bản cung cấp các chức năng để có thể thực hiện việc này một cách đơn giản và an toàn
- Hệ thống quản lý phiên bản được thiết kế với mục đích thay thế hệ thống quản lý phiên bản *Concurrent Versioning System (CVS)* đã cũ và có nhiều nhược điểm. Hệ thống quản lý phiên bản có thể được sử dụng để quản lý bất cứ hệ thống phiên bản nào
- Hệ thống quản lý phiên bản là hệ thống quản lý mạnh mẽ, hữu dụng, và linh hoạt.
- Hệ thống quản lý phiên bản quản lý tập tin và thư mục theo thời gian
- Hệ thống quản lý phiên bản giống như một hệ thống tập tin máy chủ mà các máy khách có thể tải xuống và tải lên các tập tin một cách bình thường
- Điểm đặc biệt của hệ thống quản lý phiên bản là nó lưu lại tất cả những gì thay đổi trên hệ thống tập tin: tập tin nào đã bị thay đổi lúc nào, thay đổi như thế nào, và ai đã thay đổi nó
- Hệ thống quản lý phiên bản cũng cho phép phục hồi lại những phiên bản cũ một cách chính xác. Các chức năng này giúp cho việc làm việc nhóm trở nên hiệu quả và an toàn hơn rất nhiều
- Thông thường, máy khách và máy chủ kết nối thông qua mạng *LAN* hoặc *Internet*. Máy khách và máy chủ có thể cùng chạy trên một máy nếu hệ thống quản lý phiên bản có nhiệm vụ theo vết lịch sử của dự án do các nhà phát triển phần mềm phát triển trong nội bộ
- Hệ thống quản lý phiên bản hỗ trợ khá nhiều giao thức để kết nối giữa máy khách và máy chủ

1.4.Lịch sử phát triển

	<i>Networking</i>	<i>Operations</i>	<i>Concurrency</i>	<i>Examples</i>
<i>First</i>	<i>None</i>	<i>One file at a time</i>	<i>Locks</i>	<i>RCS, SCCS</i>
<i>Second</i>	<i>Centralized</i>	<i>Multi-file</i>	<i>Merge before commit</i>	<i>CVS, SourceSafe, Subversion, Team Foundation Server</i>
<i>Third</i>	<i>Distributed</i>	<i>Changesets</i>	<i>Commit before merge</i>	<i>Bazaar, Git, Mercurial</i>

Bảng 1.1 :Ba thể hệ của quản lý phiên bản

Lịch sử bốn mươi năm của các công cụ quản lý phiên bản cho thấy một phong trào hướng tới ổn định hơn.

- Trong các công cụ thể hệ đầu tiên.Một người có thể làm việc trên một tập tin tại một thời điểm.
- Các công cụ thể hệ thứ hai là một chút công bằng hơn về sửa đổi cùng một lúc, với một hạn chế đáng chú ý. Người sử dụng phải hợp nhất các phiên bản hiện tại trước khi *commit*.
- Các công cụ thể hệ thứ ba cho phép *commit* trước khi hợp nhất.

1.5.Các khái niệm và thuật ngữ quan trọng

1.5.1. Nơi lưu trữ

Nơi lưu trữ (*repository*) là một nơi chung lưu trữ mọi bản chính (*master copy*) của các tập tin khi chúng được thêm vào trong cơ sở dữ liệu của một *VCS(Version Control System)*.

Có thể được lưu trữ riêng cho từng tập tin hoặc chung cho mọi tập tin.

Trong môi trường phát triển nhóm, nơi lưu trữ thường nằm trên một máy chủ riêng.

Nơi lưu trữ có thể lưu trữ: mã nguồn, sơ liệu, các tài liệu khác (hình ảnh, âm thanh, ...)

Nơi lưu trữ có thể lưu trữ mọi thứ. Vậy các loại tập tin nào nên được đưa vào? Nếu không có phiên bản cập nhật của tập tin `config.php`, chúng ta không thể xây dựng được ứng dụng của mình. Khi đó tập tin `config.php` nên được cho vào trong nơi lưu trữ.

1.5.2. Thư mục làm việc

Thư mục làm việc (*working folder*) là một nơi mà chúng ta lấy phiên bản của tập tin và hiệu chỉnh.

Thư mục làm việc thường được đặt trên máy client cho từng thành viên trong khi làm việc nhóm.

Khi các phiên bản được lấy ra, thư mục này là nơi mặc định mà một VCS sẽ dùng để lưu trữ các bản sao phiên bản đó.

1.5.3. Khóa

Khi làm việc nhóm, các thành viên có thể hiệu chỉnh cùng một tập tin tại một thời điểm. Xung đột là điều tất yếu xảy ra. Làm sao tránh xung đột? Tùy chọn khóa hay cơ chế khóa (*locking*) cho phép ngăn chặn xung đột. Có 2 loại khóa:

- Khóa nghiêm ngặt (*strict locking*): Trong khóa này, khi một tập tin được lấy ra để hiệu chỉnh, nó sẽ bị đặt thuộc tính là read-only. Do đó, thành viên có thể đọc nó nhưng không thể hiệu chỉnh nội dung của nó. Chỉ khi nào tập tin đó được cập nhật, thành viên khác mới có thể lấy ra lại.

Vấn đề: nếu một thành viên lấy ra hiệu chỉnh quá lâu và chưa cập nhật, những thành viên khác sẽ không làm gì được.

- Khóa tự do (*optimistic locking*): Khóa tự do cho phép khi một tập tin được lấy ra để sửa, các thành viên khác cũng có quyền cập nhật dữ liệu trong tập tin đó. Và khi hai thành viên cùng *check in* tập tin đã được hiệu chỉnh, nguyên tắc sau được thực hiện:
 - Nếu 2 bản cập nhật không cùng cập nhật 1 dòng, việc cập nhật sẽ ghép 2 bản đó lại.
 - Nếu 2 bản cùng cập nhật 1 dòng, VCS sẽ báo cho 2 thành viên đó và cho họ tự chọn việc cập nhật.

1.5.4. Dự án, module và tập tin

Dự án là một thư mục lưu trữ toàn bộ thông tin của các tập tin của nhóm. Một nhóm có thể thực hiện nhiều dự án khác nhau. Các dự án này có thể chia sẻ các tập tin (*shared file*). Mỗi VCS thường có một dự án gốc cho mọi dự án.

Mô đun là một dự án con (*sub-project*) nằm trong một dự án mẹ.

Tập tin trong dự án có thể thuộc về một hoặc nhiều dự án (*shared file*).

1.5.5. Phân nhánh

Khi làm việc nhóm, mọi thành viên đều có một phần mã chung, gọi là nhánh chính (*mainline*). Và các thành viên đều làm việc với một phần của nhánh chính. Khi có yêu cầu tách một phần mã chung đó, ví dụ như để tạo một phiên bản để thử nghiệm, các thành viên của nhóm phân phối thử nghiệm đó cần mã độc lập với nhánh chính. Giải pháp là dùng phân nhánh.

Chia nhánh (*branching*) là một cơ chế của hệ thống quản lý phiên bản cho phép tách một phần của dự án ra riêng. Phần này sẽ hoạt động độc lập với dự án chung. Khi đó, việc đánh phiên bản cũng thay đổi.

Ví dụ: nếu một tập tin `config.php` trong nhánh chính có số phiên bản là 1.14 và một nhánh mới được tạo ra, `config.php` sẽ có số phiên bản trong nhánh là 1.14.1.1. Việc hiệu chỉnh tập tin đó trong nhánh chính sẽ cho chỉ số phiên bản mới trong nhánh chính là 1.15 tuy nhiên nếu hiệu chỉnh tập tin đó trong nhánh thì nó có số phiên bản mới là 1.14.1.2.

1.5.6. Ghép tập tin

Có nhiều nhánh (*branch*) có thể được tách ra làm việc độc lập. Nếu họ trong khi làm thấy rằng có một số lỗi (bug) phát sinh và có ảnh hưởng tới nhánh chính. Họ sẽ làm sao để cập nhật nhanh nhất mã trong nhánh chính và dùng cách ghép tập tin (*merge*).

Ghép tập tin sẽ cho phép loại bỏ được việc phải cắt dán nhiều lần trên các phiên bản khác nhau của hệ thống.

1.5.7. Phiên bản

Mỗi tập tin có thể có nhiều phiên bản (*version*). Các phiên bản được đánh số khác nhau. Mỗi lần chúng ta hiệu chỉnh nội dung tập tin và cập nhật vào nơi lưu trữ của VCS, phiên bản sẽ được cập nhật.

Một số VCS sử dụng hệ thống phiên bản 1, 2, 3, ... trong khi có một số khác sử dụng hệ thống 1.0, 1.1, 1.2, ... Các VCS thường lưu phiên bản cuối cùng và các thay đổi của phiên bản cuối so với phiên bản trước đó.

Các số phiên bản (như 1.0, 1.1, hay 1.2, ...) còn được gọi là revision (bản sửa đổi).

Đi đôi với số phiên bản, VCS lưu thêm các thông tin về ngày giờ cập nhật, người cập nhật và một số ghi chú nếu có.

Một vấn đề nhỏ trong cách đánh số này là làm sao các thành viên trong nhóm biết được các phiên bản của dự án. Số phiên bản trong dự án sẽ thay đổi khi phiên

bản của tập tin thay đổi. Các VCS do đó sử dụng thêm một cách ghi nhận phiên bản là đánh nhãn (*labeling*). Khi đó, cho dù số phiên bản của dự án thay đổi, nhãn của dự án vẫn không đổi [9].

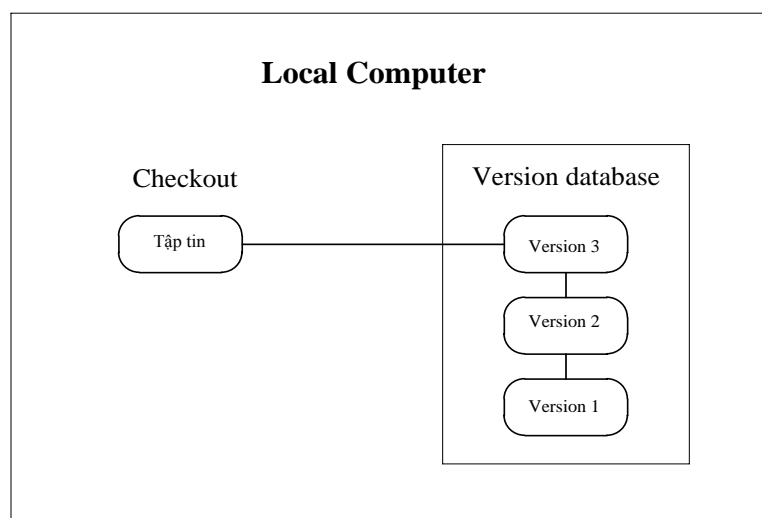
1.6. Kiến trúc của hệ thống quản lý phiên bản

Hầu hết hệ thống quản lý phiên bản phổ biến hiện nay được chia thành 3 loại chính: Quản lý phiên bản cục bộ (*Local Version Control Systems*), Quản lý phiên bản tập trung (*Centralized Version Control System*) và quản lý phiên bản phân tán (*Distributed Version Control System*)

1.6.1. Hệ thống quản lý phiên bản cục bộ

Phương pháp quản lý phiên bản này được nhiều người lựa chọn để sao chép các tập tin vào một thư mục khác (có thể là một thư mục đánh dấu theo thời gian). Phương pháp này là rất phổ biến vì nó rất đơn giản, nhưng lại rất dễ bị lỗi. Rất dễ nhầm khi vô tình ghi các tập tin sai hoặc chép các tập tin mà bạn không mong muốn vào thư mục đang làm.

Để giải quyết các vấn đề này các lập trình viên phát triển hệ thống quản lý phiên bản cục bộ đã có một cơ sở dữ liệu đơn giản mà giữ tất cả những thay đổi dưới sự quản lý các phiên bản sửa đổi (*Xem hình 1*)[8].

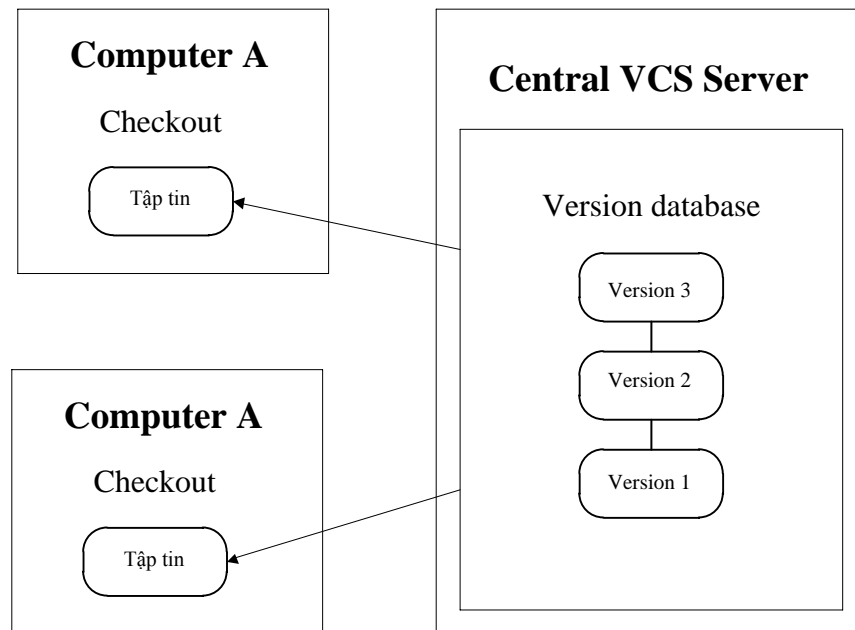


Hình 1.1: Sơ đồ hệ thống quản lý phiên bản cục bộ

Một trong những công cụ quản lý phiên bản phổ biến là một hệ thống gọi là *RCS* (*Revision Control System*). Hiện vẫn được sử dụng trong các máy tính ngày nay. Thậm chí cả hệ điều hành *Mac OS* nổi tiếng bao gồm các lệnh *RCS* khi cài đặt các công cụ phát triển. Công cụ này về cơ bản hoạt động bằng cách giữ tập hợp bản vá từ một phiên bản khác trong một dạng đặc biệt trên ổ đĩa, sau đó có thể tạo bất kỳ tập tin trông giống như thế tại bất kỳ thời điểm nào bằng cách cộng tất cả các bản vá lỗi lại.

1.6.2. Hệ thống quản lý phiên bản tập trung

Vấn đề quan trọng mà mọi người gặp phải là cần phải hợp tác với các nhà phát triển trên các hệ thống khác. Để giải quyết vấn đề này, hệ thống quản lý phiên bản tập trung được phát triển. Những hệ thống chẳng hạn như CVS (*Concurrent Versioning System*), Subversion,... có máy chủ duy nhất chứa tất cả các tập tin phiên bản và một số máy khách kiểm tra các tập tin đó từ nơi trung tâm. Trong nhiều năm nay, đã được chuẩn hóa trong quản lý phiên bản (*Xem hình 2*) [8].



Hình 1.2 : Sơ đồ hệ thống quản lý phiên bản tập trung

Các thiết lập này cung cấp nhiều lợi thế, đặc biệt là trên hệ thống quản lý phiên bản cục bộ. Ví dụ : ai cũng có quyền nhất định trên dự án mình đang làm. Các người quản lý có quyền kiểm soát các hạt mịn và nó dễ dàng hơn để quản lý một CVS hơn so với việc đối phó với cơ sở dữ liệu cục bộ trên tất cả các máy khách.

Tại thời điểm này hệ thống quản lý phiên bản phổ biến nhất được biết đến là Subversion , được coi là một hệ thống quản lý phiên bản tập trung. Khái niệm chính của hệ thống phiên bản tập trung là hoạt động trong quan hệ khách hàng và máy chủ. Kho nằm ở một nơi và cung cấp truy cập cho nhiều máy khách. Nó rất giống với FTP tại nơi bạn có một máy khách FTP có kết nối với một máy chủ FTP. Mọi sự thay đổi, người dùng, commit và thông tin phải được gửi và nhận từ kho lưu trữ trung tâm này

Ưu điểm của hệ thống quản lý phiên bản tập trung:

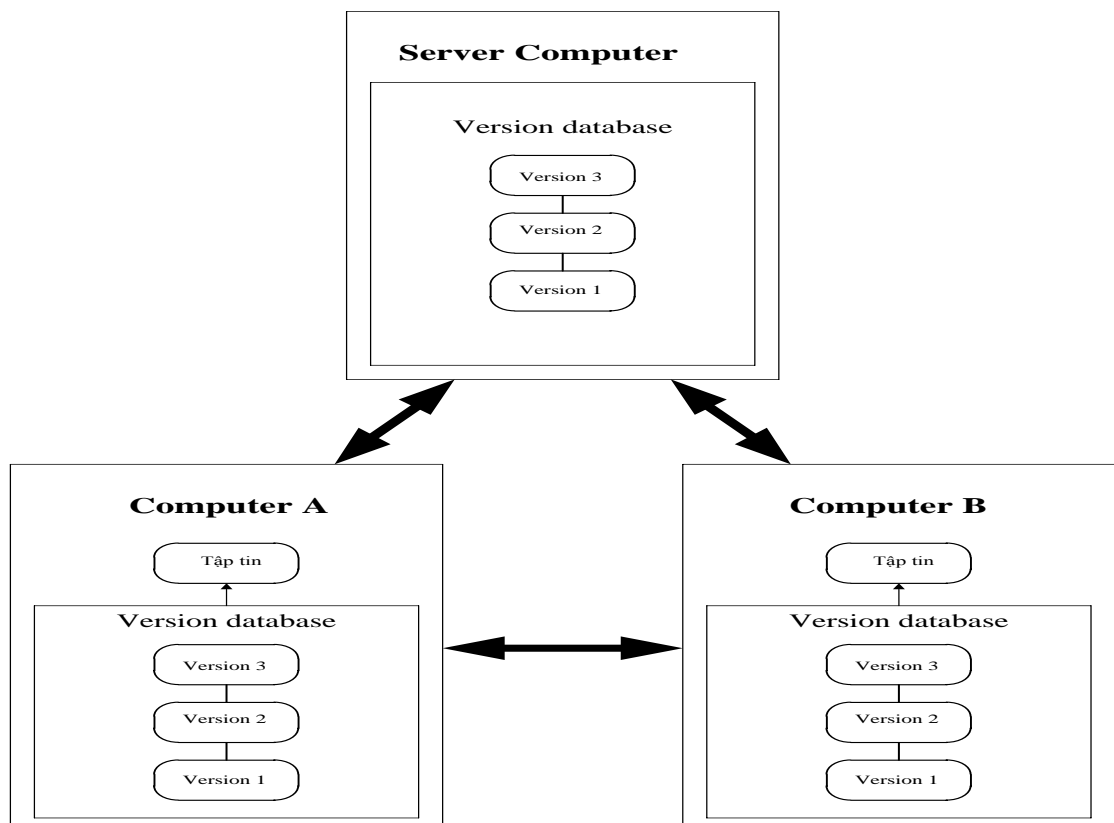
- Dễ hiểu.
- Có quyền quản lý người dùng và quyền truy cập.
- Rất đơn giản cho người mới bắt đầu.

Nhược điểm của hệ thống quản lý phiên bản tập trung:

- Phụ thuộc vào quyền truy cập máy chủ.
- Khó khăn để quản lý một máy chủ và các bản sao lưu.
- Rẽ nhánh và hợp nhất khó sử dụng[6].

1.6.3. Hệ thống quản lý phiên bản phân tán

Đây là hệ thống quản lý phiên bản phân tán (DVCS). Các DVCS (như Git, Mercurial, Bazaar hoặc Darcs) máy khách cần kiểm tra các ảnh chụp mới nhất của các tập tin. Vì vậy, nếu bất kỳ máy chủ bị hỏng, các hệ thống này sẽ cộng tác thông qua nó, bất kỳ những kho của máy khách có thể sao chép lại các thiết lập của máy chủ để khôi phục lại nó. Mọi thứ *checkout* là bản sao lưu đầy đủ của tất cả các dữ liệu (Xem hình 3) [8].



Hình 1.3: Sơ đồ hệ thống quản lý phiên bản phân tán

Hệ thống phân tán là một lựa chọn mới. Trong quản lý phiên bản phân tán, mỗi người sử dụng có 1 bản sao riêng của mình trong kho lưu trữ, không phải chỉ có tập tin mà có cả lịch sử lưu trữ. Hãy xem nó như là một mạng lưới kho lưu trữ cá nhân. Hai hệ thống quản lý phiên bản phân tán phổ biến nhất là *Git* và *Mercurial*.

Ưu điểm của hệ thống quản lý phiên bản phân tán:

- Theo dõi sự thay đổi mạnh mẽ và chi tiết hơn, có nghĩa là các cuộc xung đột ít hơn.
- Không gian làm việc riêng.
- Không cần máy chủ (*commit offline*).
- Rẽ nhánh và hợp nhất tin cậy hơn, do đó được sử dụng thường xuyên sử dụng.
- Làm việc nhanh hơn.
- Sao lưu ngầm.

Nhược điểm của hệ thống quản lý phiên bản phân tán:

- Các mô hình phân phối là khó hiểu.
- Kho lưu trữ lớn.
- Không có nhiều giao diện[6].

1.6.4. Các thao tác của hệ thống quản lý phiên bản tập trung

Có 18 thao tác cơ bản trong hệ thống quản lý phiên bản tập trung. Trong phần này giới thiệu một số thao của hệ thống quản lý phiên bản.

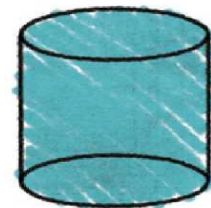
1.6.4.1. Tạo (Create)

Một kho lưu trữ là nơi người dùng lưu trữ tất cả các công việc. Nó theo dõi những tài liệu của người dùng (nghĩa là tất cả các tập tin của người dùng), cũng như cách bố trí của các tập tin mà chúng được lưu.

Nếu định nghĩa ở đoạn trên là toàn bộ vấn đề. Kho lưu trữ quản lý phiên bản không được phép nhiều hơn một hệ thống tập tin. Một kho lưu trữ tất cả các lịch sử.

Kho lưu trữ = Hệ thống tập tin * Thời gian

Một hệ thống tập tin có hai chiều: không gian của nó được xác định bởi các thư mục và tập tin. Ngược lại, một kho lưu trữ có ba chiều: Nó tồn tại trong một chuỗi liên tục được định nghĩa bởi các thư mục, tập tin, và thời gian. Một kho lưu trữ quản lý phiên bản lưu trữ tất cả các phiên bản của mã nguồn của người dùng đã từng tồn tại. Một khi người dùng thực hiện một số thay đổi đến kho lưu trữ, ngay cả



REPOSITORY

Hình 1.4 : Tạo mới một kho lưu trữ rỗng

khi sự thay đổi để xóa một cái gì đó, các kho lưu trữ sẽ tăng lên bởi vì lịch sử cũng tăng theo. Người dùng không bao giờ loại trừ bất cứ điều gì từ lịch sử.

Thao tác tạo được sử dụng để tạo ra một kho lưu trữ mới. Đây là một trong những hoạt động đầu tiên mà bạn sẽ sử dụng và sau đó, nó được sử dụng rất nhiều và thường xuyên hơn.

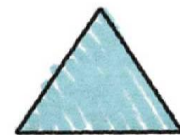
Hầu hết các hệ thống quản lý phiên bản có nhiều hơn 18 lệnh, bao gồm rất nhiều công cụ hữu ích. Chương này là khoảng 18 hoạt động phổ biến mà có thể được coi là khái niệm cốt lõi của quản lý phiên bản.

Khi người dùng tạo ra một kho lưu trữ mới, VCS của bạn sẽ mong bạn làm điều gì đó để nhận biết nó, chẳng hạn như nơi bạn muốn nó được tạo ra, hoặc những gì tên của nó nên được.

1.6.4.2. Lấy tập tin (Checkout)

Hoạt động *Checkout* được sử dụng khi bạn cần phải tạo một bản sao làm việc mới cho một kho lưu trữ đã tồn tại.

Một bản sao làm việc là một bản sao được sử dụng để làm việc. Một bản sao làm việc là một bản chụp của các kho lưu trữ được sử dụng bởi một nhà phát triển như là một nơi để thực hiện thay đổi. Kho lưu trữ được chia sẻ cho tất cả những người trong một nhóm, nhưng người ta không sửa đổi trực tiếp. Thay vào đó, mỗi người trong nhóm làm việc bằng cách sử dụng với một bản sao làm việc. Bản sao làm việc cho người dùng một không gian làm việc riêng, nơi mà người dùng có thể làm công việc một mình với phần còn lại của nhóm.



WORKING COPY

Hình 1.5: Tạo một bản sao làm việc.

Cuộc đời của một nhà phát triển là một vòng lặp vô hạn mà trông giống như sau:

1. Tạo một bản sao làm việc.
2. Sửa đổi các bản sao làm việc.
3. Sửa đổi các kho lưu trữ kết hợp những thay đổi.
4. Goto 2

Hãy tưởng tượng một chút những gì cuộc sống sẽ như thế nào mà không có sự phân biệt giữa bản sao làm việc và kho lưu trữ. Trong một nhóm người làm việc độc lập, tình huống này sẽ không có tranh chấp xảy ra. Tuy nhiên, đối các nhà phát triển làm việc cùng trên một tập tin gốc, mọi thứ có thể trở nên rất lộn xộn.

Người sử dụng lưu trữ mã của họ trên một máy chủ của tất cả tập tin. Tất cả mọi người sử dụng tập tin để chia sẻ và chỉnh sửa các tập tin gốc tại chỗ. Khi ai đó

muốn chỉnh sửa đó, người dùng sẽ không thể làm được gì nếu có ai đó đang sử dụng tập tin gốc đó.

Với một công cụ quản lý phiên bản, làm việc trên một nhóm đa người là đơn giản hơn nhiều. Mỗi nhà phát triển có một bản sao làm việc để sử dụng như một không gian làm việc riêng tư. Người sử dụng có thể thay đổi bản sao làm việc của mình mà không ảnh hưởng xấu đến phần còn lại của đội.

Bản sao làm việc thực sự không chỉ là một bản chụp của các nội dung của kho lưu trữ. Nó còn lưu trữ một số dữ liệu để có thể theo dõi cẩn thận tình trạng của sự vật.

Hãy giả sử người sử dụng có một bản sao làm việc mới. Nói cách khác, người dùng bắt đầu với một phiên bản đầu tiên từ kho lưu trữ. Tại thời điểm này, bản sao làm việc mới của người dùng hoàn toàn đồng bộ với các nội dung của kho lưu trữ. Nhưng với điều kiện là không giữ bản sao làm việc quá lâu. Người dùng sẽ làm thay đổi một số thứ của tập tin trong bản sao làm việc này, do đó nó sẽ trở nên mới hơn so với phiên bản đầu tiên kho lưu trữ. Người dùng khác cũng thay đổi tập tin trong kho lưu trữ, do đó làm cho bản sao của người dùng đầu tiên coi như là bản sao cũ đối với những người sử dụng tiếp theo. Bản sao này được coi như là vừa mới và vừa cũ. Những điều này sẽ gây nhầm lẫn. Các công cụ điều khiển phiên bản có trách nhiệm theo dõi tất cả mọi thứ. Trong thực tế, nó phải theo dõi tình trạng của từng tập tin.

Mục tiêu của công cụ là quản lý, các công cụ quản lý phiên bản thường giữ thông tin để theo dõi bản sao làm việc. Khi một tập tin được lấy ra, VCS lưu trữ nội dung của nó trong bản sao làm việc tương ứng của tập tin đó, mà nó còn ghi lại một số thông tin nhất định. Ví dụ:

- Công cụ quản lý phiên bản có thể ghi lại các dấu thời gian trên các tập tin làm việc để nó sau này có thể phát hiện ra người dùng nào đã thay đổi nó.
- Nó có thể ghi lại số phiên bản của tập tin đã được lấy ra từ kho lưu trữ để sau này có thể biết phiên bản bắt đầu thay đổi từ lúc nào.
- Nó thậm chí có thể gài đi một bản sao đầy đủ của tập tin đó đã được lấy ra để nó có thể hiển thị một người khác mà không cần truy cập vào máy chủ.

Công cụ này được lưu trữ trong khu vực quản lý, mà thường là một hoặc nhiều thư mục ẩn trong bản sao làm việc. Vị trí chính xác của nó phụ thuộc vào công cụ quản lý phiên bản bạn đang sử dụng.

1.6.4.3. Cập nhật tập tin khi checkout(Commit)

Thao tác này thực tế là sửa đổi các kho lưu trữ, một số thao tác khác sửa đổi bản sao làm việc và bổ sung một thao tác vào danh sách gọi là danh sách chờ thay đổi. Các thao tác *commit* lấy danh sách chờ và sử dụng nó để tạo ra một phiên bản mới của cây trong kho.



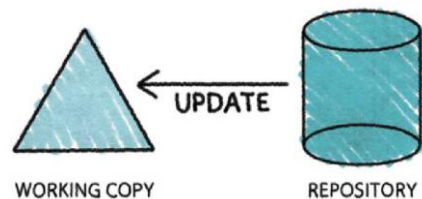
Hình 1.6 : Áp dụng các thay đổi trong bản sao với kho như thay đổi mới.

Tất cả các công cụ quản lý phiên bản hiện đại thực hiện thao tác một cách độc lập. Hay nói một cách khác, dù có bao nhiêu thay đổi trong danh sách chờ cũng không có vấn đề gì, kho lưu trữ sẽ kết thúc với tất cả các hoạt động (nếu hoạt động thành công), hoặc không một hoạt động nào được thực hiện (nếu hoạt động không thành công). Không thể xảy ra trường hợp kho chứa kết thúc trong trạng thái hoàn thành được một nửa. Tính toàn vẹn của kho lưu trữ được đảm bảo.

Điểm nổi bật là cung cấp một nhật ký khi *commit*, hay diễn giải các thay đổi đã tạo ra. Nhật ký là một phần của lịch sử kho lưu trữ.

1.6.4.4. Cập nhật (Update)

Cập nhật mang đến cho bản sao làm việc của được cập nhật bằng cách áp dụng những thay đổi từ kho lưu trữ, kết hợp chúng với bất kỳ thay đổi mà đã thực hiện cho bản sao làm việc nếu cần thiết. Khi các bản sao làm việc lần đầu tiên được tạo ra, nội dung của nó chính xác phản ánh một phiên bản cụ thể của kho. VCS ghi nhớ các sửa đổi đó để có thể theo dõi cẩn thận về nơi bắt đầu thực hiện thay đổi của bạn. Sửa đổi này thường được gọi là phần tử cấp trên của các bản sao làm việc, bởi vì nếu bạn chuyển thay đổi từ bản sao làm việc, sửa đổi đó sẽ là phần tử cấp trên của thay đổi mới



Hình 1.8 : Cập nhật các bản sao làm việc đối với các kho lưu trữ.

Cập nhật này giống như hình ảnh phản chiếu của *commit*. Cả hai hoạt động di chuyển thay đổi giữa các bản sao làm việc và kho lưu trữ. *Commit* đi từ bản sao làm việc vào kho. Cập nhật đi theo một hướng khác.

1.6.4.5. Thêm (Add)

Thêm một tập tin hoặc thư mục.

Sử dụng các thao tác thêm khi có một tập tin hoặc thư mục trong bản sao làm việc mà chưa kiểm soát phiên bản và muốn thêm nó vào kho lưu trữ. Các sự thay đổi thực tế không được bổ sung vào ngay. Thay vào đó, các sự thay đổi trở thành một phần của danh sách chờ, và được bổ sung vào kho lưu trữ khi bạn *commit*.

1.6.4.6. Sửa (Edit)

Sửa một tập tin

Đây là thao tác phổ biến nhất khi sử dụng một hệ thống quản lý phiên bản. Khi *checkout*, bản sao làm việc bao gồm một loạt các tập tin từ kho lưu trữ. Người dùng chỉnh sửa các tập tin sẽ thay đổi một phần trong kho lưu trữ.

Với hầu hết các công cụ quản lý phiên bản, các hoạt động chỉnh sửa không thực sự liên quan trực tiếp đến VCS. Người dùng sửa đổi các tập tin bằng bất kỳ trình soạn thảo văn bản hay môi trường phát triển nào và VCS sẽ ghi nhận sự thay đổi của tập tin và đưa vào danh sách chờ.

Mặt khác, một số công cụ quản lý phiên bản muốn bạn được rõ ràng hơn. Những công cụ này thường được thiết lập chỉ cho phép đọc tất cả các tập tin trong bản sao làm việc. Mà không cho phép sửa các tập tin gốc

1.6.4.7. Xóa (Delete)

Xóa một tập tin hoặc thư mục.

Sử dụng các thao tác xóa khi bạn muốn xóa một tập tin hoặc thư mục từ kho lưu trữ.

Nếu bạn cố gắng để xóa một tập tin đã được sửa đổi trong bản sao làm việc, VCS sẽ đưa ra cảnh báo.

Thông thường, các thao tác xóa sẽ lập tức xóa các bản sao làm việc của tập tin, nhưng thực tế việc xóa của các tập tin trong kho lưu trữ chỉ đơn giản là thêm vào danh sách chờ

Hãy nhớ rằng trong kho lưu trữ tập tin không thực sự bị xóa. Khi bạn *commit* một danh sách chờ bao hàm việc xóa, chỉ đơn giản là tạo ra một phiên bản mới của cây mà không lưu trữ các tập tin đã xóa. Các phiên bản trước của cây vẫn còn trong kho lưu trữ, và phiên bản đó vẫn còn lưu trữ các tập tin.

1.6.4.8. Đổi tên (Rename)

Đổi tên một tập tin hoặc thư mục.

Sử dụng thao tác đổi tên khi bạn muốn thay đổi tên của một tập tin hoặc thư mục. Các hoạt động được thêm vào danh sách chờ, nhưng các mục trong bản sao làm việc thường được đổi tên ngay lập tức.

Có rất nhiều sự đa dạng trong các công cụ quản lý phiên bản hỗ trợ đổi tên. Một số công cụ trước đó đã không hỗ trợ cho việc đổi tên tất cả.

Một số công cụ (*Bazaar* và *Veracity*) thực hiện đổi tên chính thức, đòi hỏi hiển thị thông báo một cách rõ ràng khi tập tin được đổi tên. Những công cụ xử lý tên của một tập tin hoặc thư mục chỉ đơn giản là thay đổi những thuộc tính theo thời gian.

Vẫn còn các công cụ khác (*Git*) thực hiện đổi tên chính thức, phát hiện đặt lại tên bằng cách quan sát những thay đổi chứ không phải bằng cách theo dõi danh tính của một tập tin. Việc phát hiện đổi tên thường hoạt động tốt trong thực tế, nhưng nếu một tập tin đã đổi tên và thay đổi, *VCS* sẽ làm việc sai lệch.

1.6.4.9. Di chuyển(Move)

Di chuyển một tập tin hoặc thư mục.

Sử dụng các thao tác di chuyển khi bạn muốn di chuyển một tập tin hoặc thư mục từ một nơi đến cây khác. Các hoạt động sẽ được thêm vào danh sách chờ, nhưng các mục trong bản sao làm việc thường được di chuyển ngay lập tức.

Một số công cụ xử lý đổi tên và di chuyển, trong khi những công cụ khác giữ các phiên bản riêng biệt.

1.6.4.10. Trạng thái (Status)

Liệt kê các thay đổi đã được thực hiện cho các bản sao làm việc.

Khi bạn thực hiện thay đổi trong bản sao làm việc của bạn, mỗi thay đổi được thêm vào danh sách chờ. Thao tác trạng thái được sử dụng để xem các danh sách chờ. Hay nói một cách khác, Trạng thái hiển thị những gì thay đổi sẽ được áp dụng cho các kho lưu trữ nếu người dùng *commit*.

1.6.4.11. Chỉ ra các chi tiết(Diff)

Chỉ ra các chi tiết của những sửa đổi đã được thực hiện với các bản sao làm việc.

Trạng thái cung cấp một danh sách các thay đổi nhưng không có chi tiết về các sự thay đổi. Để xem chính xác những gì thay đổi đã được thực hiện cho các tập tin,

bạn cần phải sử dụng thao tác *diff*. Hệ thống VCS có thể thực hiện *diff* theo nhiều cách khác nhau. Đối với một ứng dụng dòng lệnh, nó chỉ đơn giản có thể in *diff* ra cửa sổ dòng lệnh. Hoặc, hệ thống VCS có thể khởi động một ứng dụng trực quan khác.

1.6.4.12. Hoàn nguyên (Revert)

Hoàn nguyên sửa đổi được thực hiện cho các bản sao làm việc.

Đôi khi thay đổi bản sao làm việc mà đơn giản là không có ý định giữ. Có lẽ bạn đã cố gắng để sửa chữa một số lỗi và phát hiện ra rằng sửa chữa của bạn đưa ra một số lỗi mới là tồi tệ hơn so với cái mới bắt đầu. Hoặc, có lẽ bạn chỉ cần thay đổi ý. Trong mọi trường hợp, một tính năng rất tốt của một bản sao làm việc là khả năng hoàn nguyên các thay đổi mà bạn đã thực hiện.

Một hoàn nguyên hoàn toàn của các bản sao làm việc sẽ vứt bỏ tất cả các danh sách chờ và trả lại bản sao làm việc đã được chỉ sau khi kiểm tra.

1.6.4.13. Log (Nhật ký)

Hiện lịch sử của những thay đổi trong kho lưu trữ.

Kho lưu trữ theo dõi tất cả các phiên bản đã từng tồn tại. Thao tác *log* là cách để xem các hồ sơ này. Nó sẽ hiển thị từng danh sách cùng với dữ liệu bổ sung như:

Ai thực hiện thay đổi.

Những thay đổi thực hiện khi nào.

Các thông báo đăng nhập là gì.

Hầu hết các công cụ quản lý phiên bản hiện nay vận hành theo 2 cách: *slicing* and *dicing*.

1.6.4.14. Tag (Đánh dấu)

Kết hợp một tên có ý nghĩa với một phiên bản cụ thể trong các kho lưu trữ.

Công cụ quản lý phiên bản cung cấp cách để đánh dấu một phiên bản đặc biệt trong lịch sử của kho lưu trữ với một tên có ý nghĩa.

Điều này không hoàn toàn khác với tên mô tả khi sử dụng tên cho các biến và các hằng số trong mã. Trong hai dòng mã sau đây là dễ hiểu?

```
if (-43 == e)
```

```
if (ERR_FILE_NOT_FOUND == errorcode)
```


1.6.4.15. Rẽ nhánh (Branch)

Tạo ra một nhánh khác khi phát triển.

Thao tác rẽ nhánh là những gì người sử dụng muốn trong quá trình phát triển rẽ thành các nhánh khác nhau. Ví dụ, khi người sử dụng phát hành phiên bản 3.0, có thể muốn tạo ra một nhánh để phát triển 4.0 tính năng có thể được lưu giữ riêng biệt từ 3.0.x khi sửa lỗi.

1.6.4.16. Hợp nhất (Merge)

Áp dụng thay đổi từ một nhánh này sang nhánh khác.

Thông thường khi bạn đã sử dụng một nhánh để cho phép khi phát triển có thể phân ra, sau đó muốn tập trung một lần nữa, ít nhất là một phần. Ví dụ: nếu bạn tạo ra một nhánh cho 3.0.x bản sửa lỗi, bạn có thể muốn sửa những lỗi xảy ra trong dòng chính của bản phát triển là tốt nhất. Nếu không có các thao tác hợp nhất, bạn vẫn có thể thực hiện điều này bằng cách thủ công làm sửa lỗi trong cả hai nhánh. Hợp nhất làm cho hoạt động này đơn giản hơn bằng cách tự động hoá công việc nhiều nhất có thể.

1.6.4.17. Giải quyết xung đột (Resolve)

Xử lý xung đột trong quá trình hợp nhất

Trong một số trường hợp, các hoạt động hợp nhất đòi hỏi sự can thiệp của người dùng. Hợp nhất tự động thỏa thuận với tất cả mọi thứ có thể để được thực hiện một cách an toàn nhất. Mọi thứ khác sẽ được coi là xung đột. Ví dụ, những gì nếu tập tin `config.php` đã được sửa đổi trong một nhánh này và xóa trong nhánh khác. Tình huống này đòi hỏi phải có một người đưa ra quyết định. Thao tác `resolve` được dùng để giúp người dùng hình dung được cách VCS giải quyết được xung đột.

1.6.4.18. Khóa (Lock)

Ngăn người khác sửa đổi một tập tin

Thao tác khóa được sử dụng để loại trừ quyền sửa đổi một tập tin. Không phải tất cả các công cụ điều khiển phiên bản có tính năng này. Trong một số trường hợp, nó được cung cấp nhưng ít được sử dụng nhất. Đối với bất kỳ tập tin có định dạng văn bản đơn giản (mã nguồn, XML...), cách tốt nhất là để các VCS xử lý các vấn đề tương tranh. Nhưng đối với tập tin nhị phân có thể không được tự động hợp nhất, có thể thao tác thủ công để lấy một khóa trên một tập tin [4].

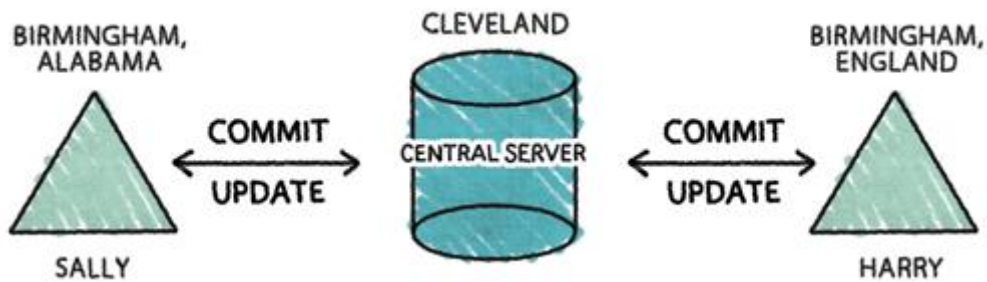
1.6.5. Các thao tác của hệ thống quản lý phiên bản phân tán

Ngoài các thao tác kế thừa từ hệ thống quản lý phiên bản tập trung, thì hệ thống quản lý phiên bản phân tán còn thêm một số thao tác khác.

1.6.5.1. Nhân bản (Clone)

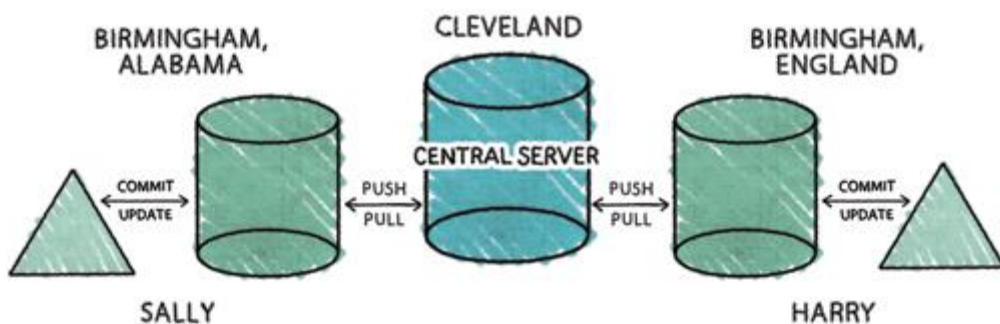
Tạo ra một phiên bản kho lưu trữ mới là bản sao của người khác.

Sự khác biệt chủ yếu giữa một hệ thống CVCS và một DVCS là khái niệm về một phiên bản kho. Cũng giống như với một CVCS, khi một kho lưu trữ được tạo ra lần đầu tiên, người dùng tạo ra. Sau đó, điều làm cho một DVCS khác là chúng ta có thể có nhiều phiên bản kho lưu trữ đó. Các thao tác tạo bản sao là cách phiên bản mới được tạo ra. Trong một CVCS, kho tồn tại ở một nơi trên một máy chủ trung tâm. Tất cả phần của phần mềm được sử dụng để truy cập vào khobao gồm mạng các máy khách



Hình 1.8 : Hệ thống quản lý phiên bản tập trung

Ngược lại, một DVCS cho phép các kho để tồn tại ở nhiều nơi. Người dùng có thể có nhiều phiên bản kho. Tất nhiên, có nhiều hơn một phiên bản kho có nghĩa là ta cần giữ chúng đồng bộ. Trong thực tế, không chỉ một DVCS cho phép nhiều phiên bản kho lưu trữ. Hầu hết các thao tác đều tương tác với một phiên bản kho cục bộ, không phải là một mạng máy chủ. Lần duy nhất mã mạng được tham gia là khi các phiên bản kho được đồng bộ. Mỗi nhà phát triển đều có riêng một phiên bản kho làm việc.



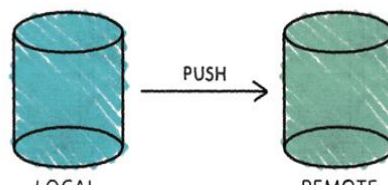
Hình 1.9 : Hệ thống quản lý phiên bản phân tán

1.6.5.2. Đẩy (Push)

Sao chép danh sách từ một phiên bản kho lưu trữ cục bộ đến sang một nơi khác

Các thao tác đẩy được sử dụng để đồng bộ hóa giữa hai phiên bản kho. Cụ thể, thao tác này xảy ra khi một phiên bản kho lưu trữ

cục bộ mà muốn sao chép một số danh sách chờ vào phiên bản kho từ xa. Thông thường, các phiên bản kho từ xa là một trong các phiên bản cục bộ được nhân bản. Lưu ý hai phiên bản kho không nhất thiết phải giống hệt nhau sau khi đẩy. Chúng ta có thể hạn chế việc đẩy bằng cách hướng dẫn các VCS chỉ gửi một số thay đổi cục bộ. Hoặc các trường hợp điều khiển từ xa có thể chứa những điều không có trong phiên bản cục bộ

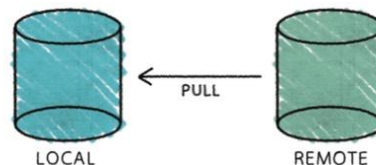


Hình 1.10 : Thao tác đẩy

1.6.5.3. Lấy (Pull)

Sao chép danh sách chờ từ một phiên bản kho lưu trữ từ xa về kho cục bộ

.Thao tác lấy cũng được sử dụng để đồng bộ hóa giữa hai phiên bản kho. Cụ thể, thao tác này xảy ra khi một phiên bản kho lưu trữ cục bộ muốn sao chép một số danh sách từ một phiên bản kho lưu trữ từ xa. Thông thường, các phiên bản kho điều khiển từ xa là một phiên bản cục bộ đã được nhân bản.



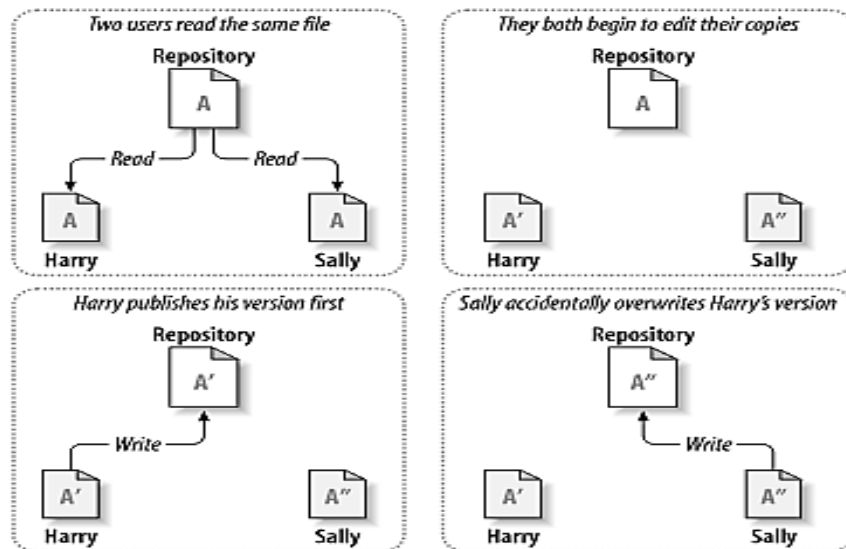
Hình 1.11 : Thao tác lấy

1.7. Các phương thức xử lý

1.7.1. Vấn đề chia sẻ tập tin

Tất cả các hệ thống quản lý phiên bản phải giải quyết các vấn đề cơ bản như nhau: Hệ thống sẽ cho phép người dùng chia sẻ thông tin, nhưng ngăn người sử dụng vô tình ghi đè lên những thay đổi của nhau trong kho.

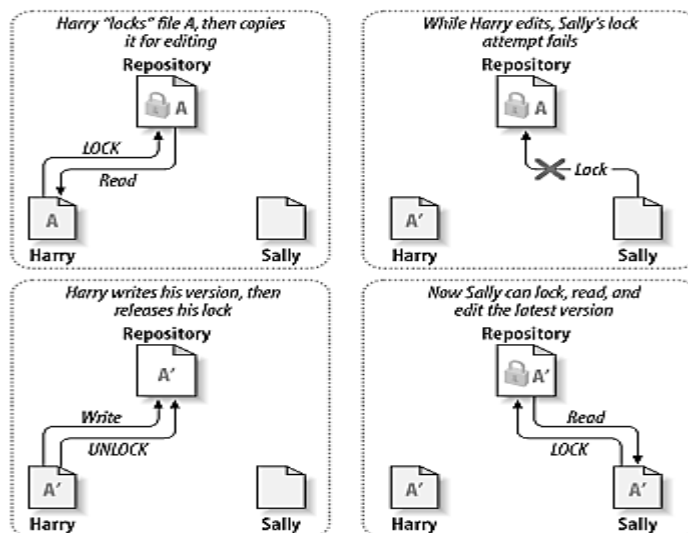
Xét tình huống thể hiện trong Hình 8 : “Vấn đề cần tránh”.Giả sử có hai người dùng,Harry và Sally. Mỗi người cùng chỉnh sửa các tập tin cùng một kho lưu trữ cùng một lúc. Nếu Harry lưu thay đổi của mình vào kho đầu tiên, có thể là một vài phút sau đó Sally vô tình ghi đè lên chúng với phiên bản tập tin mới của mình. Mặc dù, phiên bản tập tin của Harry không bị mất (vì hệ thống lưu lại mỗi lần thay đổi), bất kỳ những thay đổi mà Harry làm sẽ không có trong phiên bản tập tin mà Sally làm, Sally không bao giờ nhìn thấy những thay đổi khi bắt đầu làm. Việc Harry làm sẽ bị mất hoặc có hiệu quả thấp từ các phiên bản tập tin mới. Đây chắc chắn là tình huống mà người sử dụng nên tránh.



Hình 1.12 : Vấn đề cần tránh

1.7.2. Các giải pháp khóa – sửa đổi – mở khóa

Nhiều hệ thống quản lý phiên bản sử dụng mô hình khóa – sửa đổi – mở khóa để giải quyết vấn đề của nhiều người dùng xung đột công việc của nhau. Trong mô hình này, các kho lưu trữ cho phép chỉ có một người thay đổi một tập tin tại một thời điểm. Chính sách độc quyền này được quản lý việc sử dụng ổ khóa. Nếu Harry đã khóa một tập tin, Sally cũng không thể khóa nó, và do đó không thể thực hiện bất kỳ thay đổi nào trên tập tin đó. Những gì Sally có thể làm là đọc các tập tin và chờ cho Harry kết thúc thay đổi và mở khóa. Sau khi Harry mở các tập tin, Sally có thể khóa và chỉnh sửa tập tin. Hình 9: “Giải pháp khóa-sửa đổi-mở khóa” thể hiện giải pháp đơn giản này.



Hình 1.13: Giải pháp khóa-sửa đổi-mở khóa

Vấn đề với mô hình khóa – sửa đổi – mở khóa có một chút hạn chế và thường trở thành một rào cản cho người sử dụng:

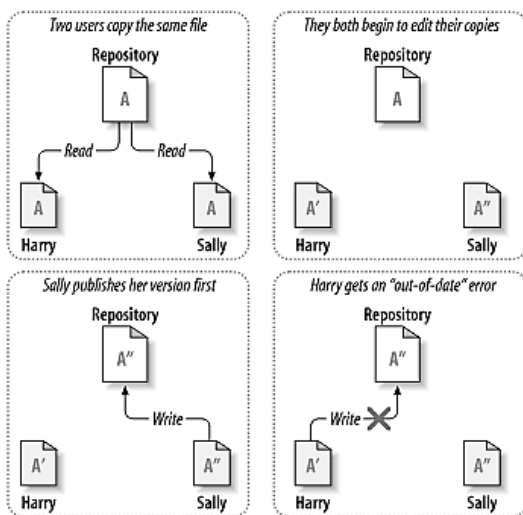
- *Khóa có thể gây ra các vấn đề quản lý.* Đôi khi Harry khóa một tập tin và sau đó quên nó. Trong lúc đó, Sally vẫn đang chờ đợi để được chỉnh sửa các tập tin. Và sau đó Harry nghỉ. Bây giờ Sally cần có quyền quản lý để mở khóa. Tình hình này gây ra rất nhiều trì hoãn không cần thiết và lãng phí thời gian.
- *Khóa có thể gây ra tuần tự không cần thiết:* Nếu Harry là người được chỉnh sửa đầu tiên của một tập tin văn bản, và Sally chỉ đơn giản là người chỉnh sửa cuối cùng một tập tin? Những thay đổi này không chồng chéo lên nhau ở tất cả thời điểm. Họ có thể dễ dàng chỉnh sửa các tập tin cùng một lúc, và không gây ra thiệt hại lớn, giả sử những thay đổi đã được sáp nhập lại với nhau đúng cách. Không cần phải cho họ thay phiên nhau trong tình huống này, giả sử những thay đổi được hợp nhất lại với nhau đúng cách. Không cần phải cho họ thay phiên nhau trong tình huống này.
- *Khóa có thể tạo ra ý nghĩa sai lệch về bảo mật:* Giả sử Harry khóa và chỉnh sửa tập tin config.php, trong khi Sally đồng thời ổ khóa và chỉnh sửa tập tin help.php. Nhưng nếu config.php và help.php phụ thuộc vào nhau, và những thay đổi được thực hiện cho từng ngữ nghĩa không phù hợp. Đột nhiên config.php và help.php không làm việc cùng nhau nữa. Hệ thống khóa đã không có cách nào để ngăn chặn các vấn đề nhưng nó có thể giải quyết vấn đề bằng cách cung cấp một ý nghĩa sai lệch về bảo mật.

1.7.3. Các giải pháp sao chép – chỉnh sửa – hợp nhất

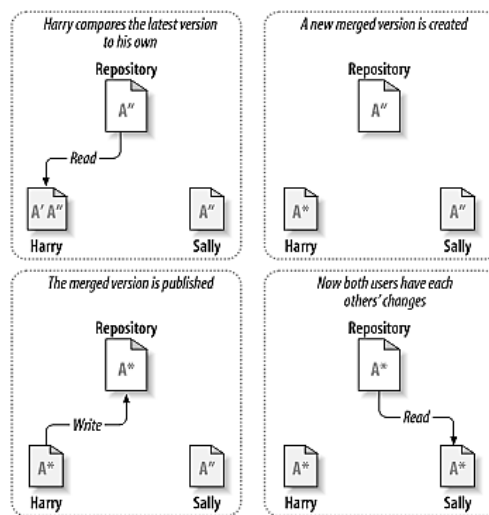
Các hệ thống quản lý phiên bản sử dụng một mô hình sao chép – chỉnh sửa – hợp nhất để thay thế cho khóa. Trong mô hình này, Mỗi người dùng sẽ làm việc với một kho lưu trữ và tạo ra các bản sao làm việc cá nhân. Sau đó người dùng làm việc một cách đồng thời và độc lập sửa đổi các bản sao làm việc của họ. Cuối cùng các bản sao được hợp nhất với nhau thành một phiên bản cuối cùng. Hệ thống quản lý phiên bản thường được hỗ trợ hợp nhất. Nhưng cuối cùng, người dùng có trách nhiệm làm cho nó hoạt động một cách chính xác nhất.

Ví dụ. Harry và Sally từng tạo ra bản sao làm việc trong cùng một dự án, sao chép từ các kho lưu trữ. Họ làm việc đồng thời và thay đổi các tập tin cùng một lúc trong các bản sao của họ. Sally lưu thay đổi của mình vào kho đầu tiên. Khi Harry cố gắng để lưu các thay đổi của mình sau đó. các kho lưu trữ thông báo tập tin config.php của Harry đã cũ. Nói cách khác, tập tin config.php trong kho lưu trữ bằng cách nào đó đã được thay đổi kể từ lần cuối cùng ông sao chép nó. Vì vậy, Harry yêu cầu máy tính hợp nhất bất kỳ thay đổi mới từ kho lưu trữ vào bản sao làm việc

(tập tin config.php). Có thể những thay đổi của Sally không trùng lặp với bản sao của Harry, một khi Harry có cả hai bộ thay đổi hợp nhất và lưu bản sao làm việc của mình trở lại kho. Hình 10 : “Giải pháp sao chép-chỉnh sửa-hợp nhất” và Hình 11: “Giải pháp bản sao-chỉnh sửa-hợp nhất (tiếp theo)” cho thấy quá trình này.



Hình 1.14 : Giải pháp sao chép-chỉnh sửa-hợp nhất



Hình 1.15 : Giải pháp bản sao-chỉnh sửa-hợp nhất (tiếp theo)

Nhưng nếu thay đổi của Sally làm chồng lên nhau với những thay đổi của Harry. Vậy thì, tình trạng này được gọi là xung đột, Harry sẽ hợp nhất những thay đổi mới của bản sao làm việc vào kho lưu trữ. Bản sao config.php được đánh dấu trong trạng thái xung đột, và người làm việc có thể nhìn thấy tập các mâu thuẫn và tự lựa chọn giữa chúng. Lưu ý phần mềm sẽ không tự động giải quyết các xung đột.

Mô hình sao chép – chỉnh sửa – hợp nhất có thể khiến người xem khó hiểu. Nhưng trong thực tế, nó làm việc rất trơn tru. Người dùng có thể làm việc song song, không bao giờ phải chờ đợi nhau. Khi họ làm việc trên cùng một tập tin, nó chỉ ra rằng hầu hết các thay đổi cùng lúc của họ không chồng chéo lên nhau, xung đột hiếm khi xảy ra. Và khoảng thời gian cần thiết để giải quyết xung đột thường là ít hơn so với thời gian đã mất bằng hệ thống khóa.

Tóm lại, Trong khi mô hình khóa – sửa đổi – mở khóa thường được xem có hại cho sự hợp tác, đôi khi khóa lại thích hợp.

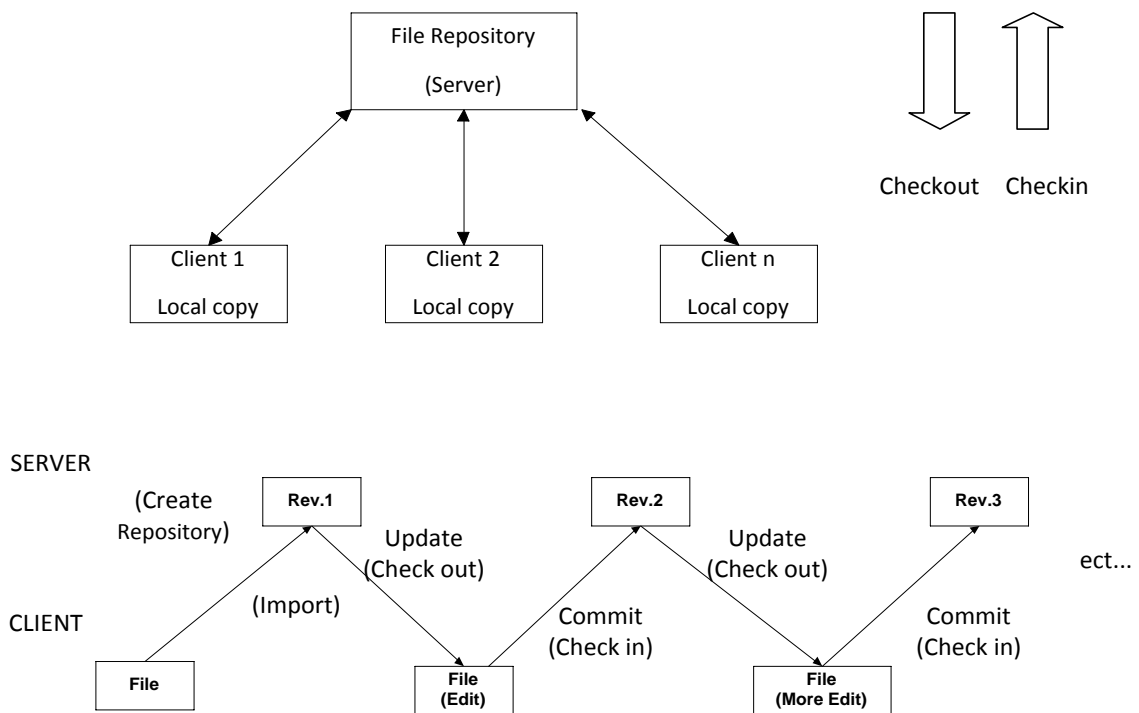
Mô hình sao chép – chỉnh sửa – hợp nhất dựa trên giả định trong ngữ cảnh tập tin có thể trộn được, phần lớn các tập tin trong kho lưu trữ là tập tin văn bản dựa trên dòng (ví dụ như chương trình mã nguồn). Nhưng các tập tin với các định dạng nhị phân, chẳng hạn như tác hình vẽ hay âm thanh, nó thường không thể hợp nhất với nhau. Trong những tình huống này, nó thực sự là cần thiết cho người dùng để

thay phiên nhau quản lý chặt chẽ khi thay đổi các tập tin. Mà không cần truy cập tuần tự [7].

CHƯƠNG 2: PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG

2.1. Tổng quan về hệ thống quản lý phiên bản văn bản

Một hệ thống kiểm soát phiên bản là một ứng dụng giúp các nhà phát triển trên một nhóm người làm việc cùng nhau và cũng có thể lưu trữ toàn bộ lịch sử của công việc. Hệ thống có khả năng tự cập nhật, so sánh và kết hợp tài nguyên mới vào tài nguyên cũ.



Hình 2.1 : Kiến trúc của một ứng dụng quản lý phiên bản

Cùng với sự phát triển của công nghệ thông tin, có thể xây dựng được một chương trình quản lý phiên bản văn bản nhằm tăng hiệu quả công việc quản lý, giảm tải sức lao động của người dùng với việc quản lý tài liệu văn bản.

2.2. Bài toán quản lý phiên bản văn bản

2.2.1. Mô tả bài toán

Bài toán quản lý phiên bản văn bản gồm các vấn đề chính:

- + Quản lý người dùng
- + Quản lý tài liệu
- + Quản lý bộ phận
- + Quản lý danh mục

2.2.2. Xác định mô hình nghiệp vụ

2.2.2.1. Các chức năng nghiệp vụ

Ta có thể xác định các chức năng nghiệp vụ của hệ thống như sau:

Tham chiếu	Chức năng
R₁	Quản lý người dùng
R ₁₁	Thêm người dùng
R ₁₂	Cập nhật người dùng
R ₁₃	Xóa người dùng
R ₁₄	Tìm kiếm người dùng
R₂	Quản lý tài liệu
R ₂₁	Thêm tài liệu
R ₂₂	Cập nhật tài liệu
R ₂₃	Xóa tài liệu
R ₂₄	Tìm kiếm tài liệu
R ₂₅	Checkout tài liệu
R ₂₆	Checkin tài liệu
R ₂₇	Xem lịch sử tài liệu

R ₂₈	Phục hồi tài liệu
R ₂₉	Xét duyệt tài liệu
R ₂₁₀	Hủy trạng thái checkout
R₃	Quản lý bộ phận
R ₃₁	Thêm bộ phận
R ₃₂	Tìm kiếm bộ phận
R ₃₃	Cập nhật bộ phận
R ₃₄	Xóa bộ phận
R₄	Quản lý danh mục
R ₄₁	Thêm danh mục
R ₄₂	Cập nhật danh mục
R ₄₃	Tìm kiếm danh mục
R ₄₄	Xóa danh mục

Bảng 2.1 : Các chức năng nghiệp vụ

2.2.2.2. Các tác nhân hệ thống

Tác nhân là một bộ phận bên ngoài hệ thống nhưng có tương tác với hệ thống. Nó chính là đối tượng mà hệ thống phục vụ hoặc cần cung cấp dữ liệu. Hệ thống quản lý phiên bản văn bản bao gồm các tác nhân sau:

Tác nhân	Vai trò
Người quản lý	Là đối tượng quản lý.
Người dùng	Là đối tượng phối hợp và thực hiện các công việc

Bảng 2.2 : Các tác nhân hệ thống

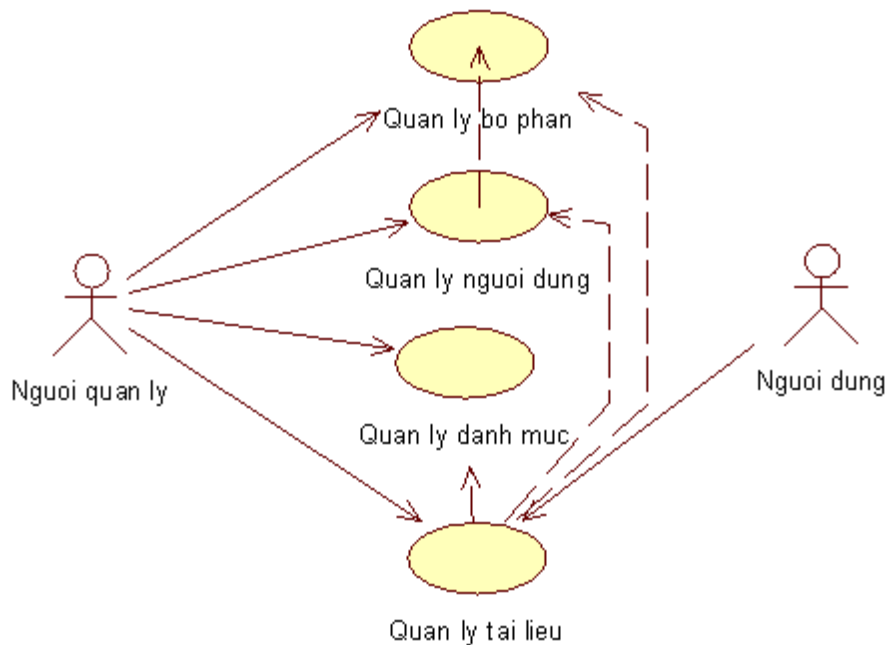
2.2.2.3. Biểu diễn mô hình hệ thống nghiệp vụ

Gói ca sử dụng	Các ca sử dụng chi tiết	Tác nhân
<i>1. Quản lý người dùng</i>	UC1. Thêm người dùng UC2. Sửa người dùng UC3. Xóa người dùng UC4. Tìm kiếm người dùng	Người quản lý
<i>2. Quản lý bộ phận</i>	UC5. Thêm bộ phận UC6. Sửa bộ phận UC7. Xóa bộ phận UC8. Tìm kiếm bộ phận	Người quản lý
<i>3. Quản lý danh mục</i>	UC9. Thêm danh mục UC10. Sửa danh mục UC11. Xóa danh mục UC12. Tìm kiếm danh mục	Người quản lý
<i>4. Quản lý tài liệu</i>	UC13. Thêm tài liệu UC14. Xóa tài liệu UC15. Sửa tài liệu UC16. Checkout tài liệu UC17. Checkin tài liệu UC18. Xem lịch sử tài liệu UC19. Tìm kiếm tài liệu UC20. Phục hồi tài liệu UC21. Xét duyệt tài liệu UC22. Hủy trạng thái checkout	Người dùng Người quản lý

Bảng 2.3 : Bảng biểu diễn mô hình hệ thống nghiệp vụ

2.2.2.3.1. Mô hình ca sử dụng mức cao

a. Biểu đồ ca sử dụng



Hình 2.2 : Mô hình ca sử dụng mức tổng quát

b. Mô tả khái quát các hệ con

Hệ thống gồm bốn hệ con:

Quản lý người dùng có tác nhân duy nhất là người quản lý. Có tác dụng thêm người dùng, tìm kiếm người dùng, sửa người dùng và xóa người dùng.

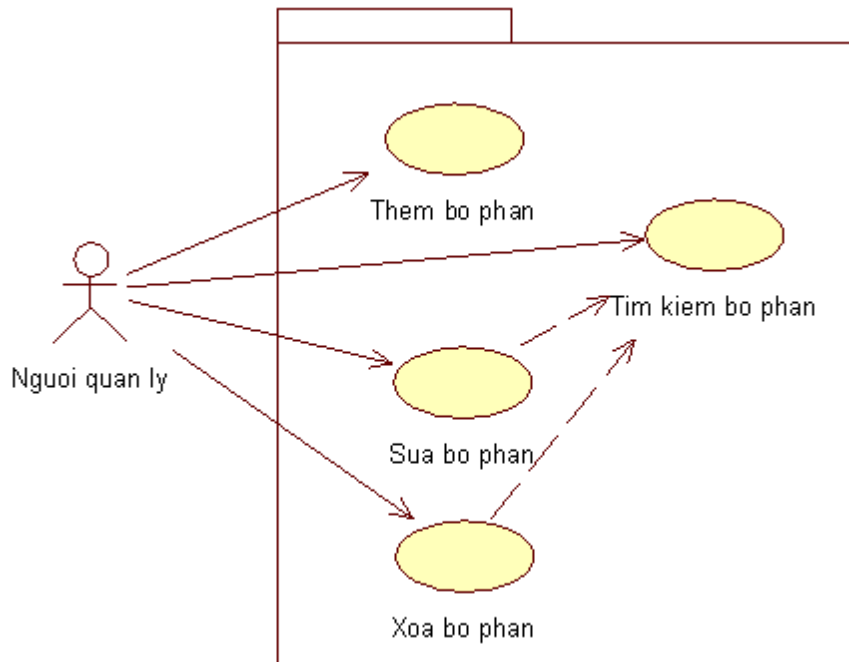
Quản lý bộ phận có tác nhân duy nhất là người quản lý. Có tác dụng thêm bộ phận, tìm kiếm bộ phận, sửa bộ phận và xóa bộ phận.

Quản lý danh mục có tác nhân duy nhất là người quản lý. Có tác dụng thêm danh mục, tìm kiếm danh mục, sửa danh mục và xóa danh mục.

Quản lý tài liệu có tác dụng thêm tài liệu, tìm kiếm tài liệu, sửa tài liệu, xóa tài liệu, checkout tài liệu, checkin tài liệu, xem lịch sử tài liệu, phục hồi tài liệu, Xét duyệt tài liệu, Xét duyệt tài liệu bị hủy khi checkin, Hủy trạng thái checkout.

2.2.2.3.2. Các mô hình ca sử dụng chi tiết

a. Gói ca sử dụng “Quản lý bộ phận”



Hình 2.3: Biểu đồ ca sử dụng gói “Quản lý bộ phận”

▪ Mô tả chi tiết ca sử dụng

- Ca sử dụng “Thêm bộ phận”

Tên ca sử dụng	Thêm bộ phận
Tác nhân	Người quản lý
Mục đích	Thêm bộ phận vào hệ thống
Mô tả khái quát	Khi muốn bổ sung thêm một bộ phận.
Các tham chiếu	R ₃₁

Bảng 2.4 : Ca sử dụng “Thêm bộ phận”

- Ca sử dụng “Tìm kiếm bộ phận”

<i>Tên ca sử dụng</i>	Tìm kiếm bộ phận
<i>Tác nhân</i>	Người quản lý
<i>Mục đích</i>	Tìm kiếm thông tin bộ phận
<i>Mô tả khái quát</i>	Khi cần sửa thông tin bộ phận hoặc cần xóa bộ phận cần tìm ra người dùng đó.
<i>Các tham chiếu</i>	R ₃₂

Bảng 2.5 : Ca sử dụng “Tìm kiếm bộ phận”

- Ca sử dụng “Sửa bộ phận”

<i>Tên ca sử dụng</i>	Sửa bộ phận
<i>Tác nhân</i>	Người quản lý
<i>Mục đích</i>	Sửa bộ phận
<i>Mô tả khái quát</i>	Sửa thông tin bộ phận khi có thay đổi
<i>Các tham chiếu</i>	R ₃₃

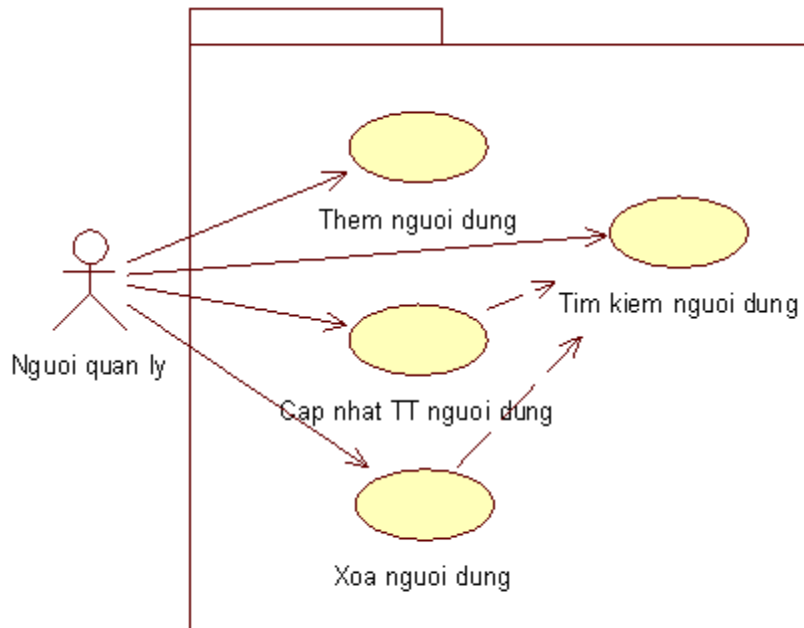
Bảng 2.6 : Ca sử dụng “Sửa bộ phận”

- Ca sử dụng “Xóa bộ phận”

<i>Tên ca sử dụng</i>	Xóa bộ phận
<i>Tác nhân</i>	Người quản lý
<i>Mục đích</i>	Xóa bộ phận ra khỏi hệ thống
<i>Mô tả khái quát</i>	Khi muốn loại bỏ bộ phận ra khỏi hệ thống.
<i>Các tham chiếu</i>	R ₃₄

Bảng 2.7 : Ca sử dụng “Xóa bộ phận”

b. Gói ca sử dụng “Quản lý người dùng”



Hình 2.4 : Biểu đồ ca sử dụng gói “Quản lý người dùng”

▪ **Mô tả chi tiết ca sử dụng**

- **Ca sử dụng “Thêm người dùng”**

<i>Tên ca sử dụng</i>	Thêm người dùng
<i>Tác nhân</i>	Người quản lý
<i>Mục đích</i>	Thêm thông tin người sử dụng vào hệ thống
<i>Mô tả khái quát</i>	Khi muốn bổ sung thêm một người quản lý hoặc người dùng .
<i>Các tham chiếu</i>	R ₁₁

Bảng 2.8 : Ca sử dụng “Thêm người dùng”

- Ca sử dụng “Tìm kiếm người dùng”

<i>Tên ca sử dụng</i>	Tìm kiếm người dùng
<i>Tác nhân</i>	Người quản lý
<i>Mục đích</i>	Tìm kiếm thông tin người dùng
<i>Mô tả khái quát</i>	Khi cần sửa thông tin người dùng hoặc cần xóa người dùng, cần tìm ra người dùng đó.
<i>Các tham chiếu</i>	R ₁₄

Bảng 2.9 : Ca sử dụng “Tìm kiếm người dùng”

- Ca sử dụng “Cập nhật TT người dùng”

<i>Tên ca sử dụng</i>	Cập nhật TT người dùng
<i>Tác nhân</i>	Người quản lý
<i>Mục đích</i>	Cập nhật thông tin của người dùng
<i>Mô tả khái quát</i>	Cập nhật thông tin của người dùng khi muốn thay đổi. Khi phân quyền một người dùng bình thường nên người quản lý
<i>Các tham chiếu</i>	R ₁₂

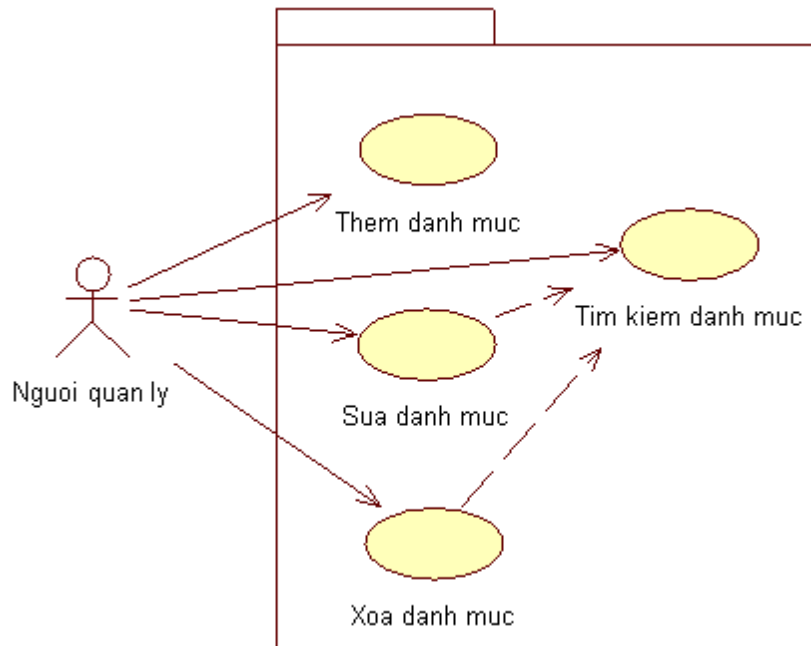
Bảng 2.10 : Ca sử dụng “Cập nhật TT người dùng”

- Ca sử dụng “Xóa hồ sơ”

<i>Tên ca sử dụng</i>	Xóa người dùng
<i>Tác nhân</i>	Người quản lý
<i>Mục đích</i>	Xóa người dùng ra khỏi hệ thống
<i>Mô tả khái quát</i>	Khi muốn loại bỏ người dùng ra khỏi hệ thống.
<i>Các tham chiếu</i>	R ₁₃

Bảng 2.11 : Ca sử dụng “Xóa hồ sơ”

c. Gói ca sử dụng “Quản lý danh mục”



Hình 2.5 : Biểu đồ ca sử dụng gói “Quản lý danh mục”

▪ Mô tả chi tiết ca sử dụng

- Ca sử dụng “Thêm danh mục”

<i>Tên ca sử dụng</i>	Thêm danh mục
<i>Tác nhân</i>	Người quản lý
<i>Mục đích</i>	Thêm danh mục vào hệ thống
<i>Mô tả khái quát</i>	Khi muốn bổ sung thêm một danh mục.
<i>Các tham chiếu</i>	R ₄₁

Bảng 2.12 : Ca sử dụng “Thêm danh mục”

- Ca sử dụng “Tìm kiếm danh mục”

<i>Tên ca sử dụng</i>	Tìm kiếm danh mục
<i>Tác nhân</i>	Người quản lý
<i>Mục đích</i>	Tìm kiếm thông tin danh mục
<i>Mô tả khái quát</i>	Khi cần sửa thông tin danh mục hoặc cần xóa danh mục cần tìm ra người dùng đó.
<i>Các tham chiếu</i>	R ₄₃

Bảng 2.13 : Ca sử dụng “Tìm kiếm danh mục”

- Ca sử dụng “Sửa danh mục”

<i>Tên ca sử dụng</i>	Sửa danh mục
<i>Tác nhân</i>	Người quản lý
<i>Mục đích</i>	Sửa danh mục
<i>Mô tả khái quát</i>	Sửa thông tin danh mục khi có thay đổi
<i>Các tham chiếu</i>	R ₄₂

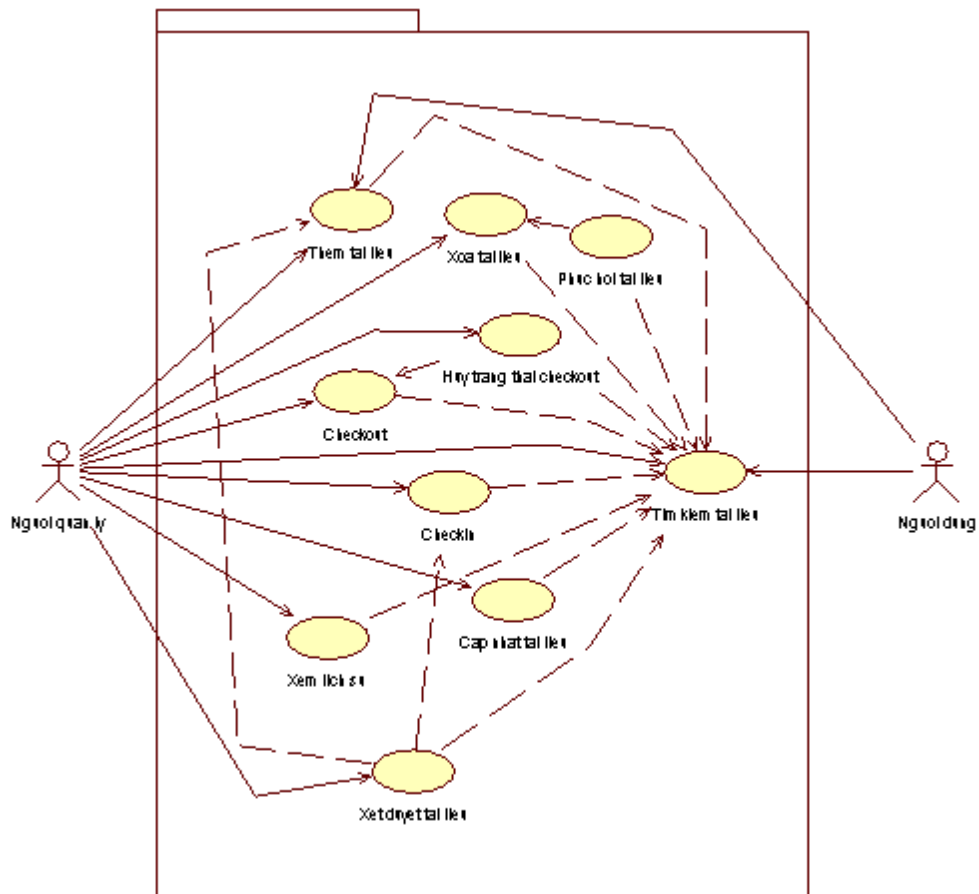
Bảng 2.14 : Ca sử dụng “Sửa danh mục”

- Ca sử dụng “Xóa danh mục”

<i>Tên ca sử dụng</i>	Xóa danh mục
<i>Tác nhân</i>	Người quản lý
<i>Mục đích</i>	Xóa danh mục ra khỏi hệ thống
<i>Mô tả khái quát</i>	Khi muốn loại bỏ danh mục ra khỏi hệ thống.
<i>Các tham chiếu</i>	R ₄₄

Bảng 2.15 : Ca sử dụng “Xóa danh mục”

d. Gói ca sử dụng “Quản lý tài liệu”



Hình 2.6 : Biểu đồ ca sử dụng gói “Quản lý tài liệu”

▪ Mô tả chi tiết ca sử dụng

- Ca sử dụng “Thêm tài liệu”

Tên ca sử dụng	Thêm tài liệu
Tác nhân	Người quản lý, người dùng
Mục đích	Thêm tài liệu vào hệ thống
Mô tả khái quát	Khi muốn bổ sung thêm một tài liệu mới vào hệ thống khi cần
Các tham chiếu	R ₂₁

Bảng 2.16 : Ca sử dụng “Thêm tài liệu”

- Ca sử dụng “Tìm kiếm tài liệu”

<i>Tên ca sử dụng</i>	Tìm kiếm tài liệu
<i>Tác nhân</i>	Người quản lý, người dùng
<i>Mục đích</i>	Tìm kiếm thông tin tài liệu
<i>Mô tả khái quát</i>	Khi cần sửa thông tin danh mục hoặc cần xóa danh mục cần tìm ra người dùng đó.
<i>Các tham chiếu</i>	R ₂₄

Bảng 2.17 : Ca sử dụng “Tìm kiếm tài liệu”

- Ca sử dụng “Cập nhật tài liệu”

<i>Tên ca sử dụng</i>	Cập nhật tài liệu
<i>Tác nhân</i>	Người quản lý, người dùng
<i>Mục đích</i>	Cập nhật tài liệu
<i>Mô tả khái quát</i>	Cập nhật thông tin tài liệu khi có thay đổi
<i>Các tham chiếu</i>	R ₂₂

Bảng 2.18 : Ca sử dụng “Cập nhật tài liệu”

- Ca sử dụng “Xóa tài liệu”

<i>Tên ca sử dụng</i>	Xóa tài liệu
<i>Tác nhân</i>	Người quản lý, người dùng
<i>Mục đích</i>	Xóa tài liệu ra khỏi hệ thống
<i>Mô tả khái quát</i>	Khi muốn loại bỏ tài liệu ra khỏi hệ thống khi không dùng đến.
<i>Các tham chiếu</i>	R ₂₃

Bảng 2.19 : Ca sử dụng “Xóa tài liệu”

- Ca sử dụng “Checkout tài liệu”

<i>Tên ca sử dụng</i>	Checkout tài liệu
<i>Tác nhân</i>	Người quản lý, người dùng
<i>Mục đích</i>	Checkout tài liệu từ hệ thống xuống máy cá nhân
<i>Mô tả khái quát</i>	Khi người dùng muốn lấy dữ liệu từ máy chủ xuống để làm việc
<i>Các tham chiếu</i>	R ₂₅

Bảng 2.20 : Ca sử dụng “Checkout tài liệu”

- Ca sử dụng “Checkin tài liệu”

<i>Tên ca sử dụng</i>	Checkin tài liệu
<i>Tác nhân</i>	Người quản lý, người dùng
<i>Mục đích</i>	Checkin tài liệu lên hệ thống
<i>Mô tả khái quát</i>	Cập nhật dữ liệu lên hệ thống sau khi checkout
<i>Các tham chiếu</i>	R ₂₆

Bảng 2.21 : Ca sử dụng “Checkin tài liệu”

- Ca sử dụng “Xem lịch sử tài liệu”

<i>Tên ca sử dụng</i>	Xem lịch sử tài liệu
<i>Tác nhân</i>	Người quản lý, người dùng
<i>Mục đích</i>	Xem lịch sử của một tài liệu
<i>Mô tả khái quát</i>	Khi người dùng muốn biết các phiên bản của tập tin, ngày sửa đổi, được ai checkin, và chú thích.
<i>Các tham chiếu</i>	R ₂₇

Bảng 2.22 : Ca sử dụng “Xem lịch sử tài liệu”

- Ca sử dụng “Phục hồi tài liệu”

<i>Tên ca sử dụng</i>	Phục hồi tài liệu
<i>Tác nhân</i>	Người quản lý
<i>Mục đích</i>	Phục hồi tài liệu từ nơi lưu trữ
<i>Mô tả khái quát</i>	Khi muốn phục hồi các tài liệu đã bị xóa hoặc các tài liệu đã bị hủy khi xét duyệt checkin
<i>Các tham chiếu</i>	R ₂₈

*Bảng 2.23 : Ca sử dụng “Phục hồi tài liệu”***- Ca sử dụng “Xét duyệt tài liệu”**

<i>Tên ca sử dụng</i>	Xét duyệt tài liệu
<i>Tác nhân</i>	Người quản lý
<i>Mục đích</i>	Xét duyệt một tài liệu
<i>Mô tả khái quát</i>	Khi người dùng hay người quản lý hoặc người dùng thêm tài liệu, checkin tài liệu lên hệ thống
<i>Các tham chiếu</i>	R ₂₉

*Bảng 2.24 : Ca sử dụng “Xét duyệt tài liệu”***- Ca sử dụng “Hủy trạng thái checkout”**

<i>Tên ca sử dụng</i>	Hủy trạng thái checkout
<i>Tác nhân</i>	Người quản lý
<i>Mục đích</i>	Hủy trạng thái checkout
<i>Mô tả khái quát</i>	Người dùng hoặc người quản lý đã check out tập tin từ hệ thống để làm việc, nhưng hủy checkout để tập tin về trạng thái ban đầu
<i>Các tham chiếu</i>	R ₂₁₀

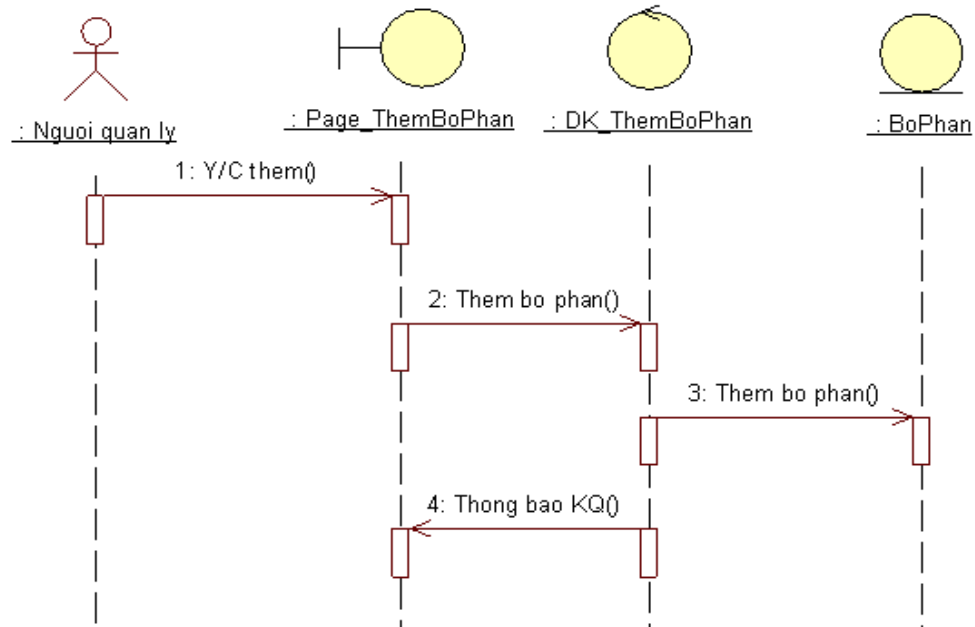
Bảng 2.25 : Ca sử dụng “Hủy trạng thái checkout”

2.2.3. Phân tích hệ thống

2.2.3.1. Phân tích gói ca sử dụng “Quản lý bộ phận”

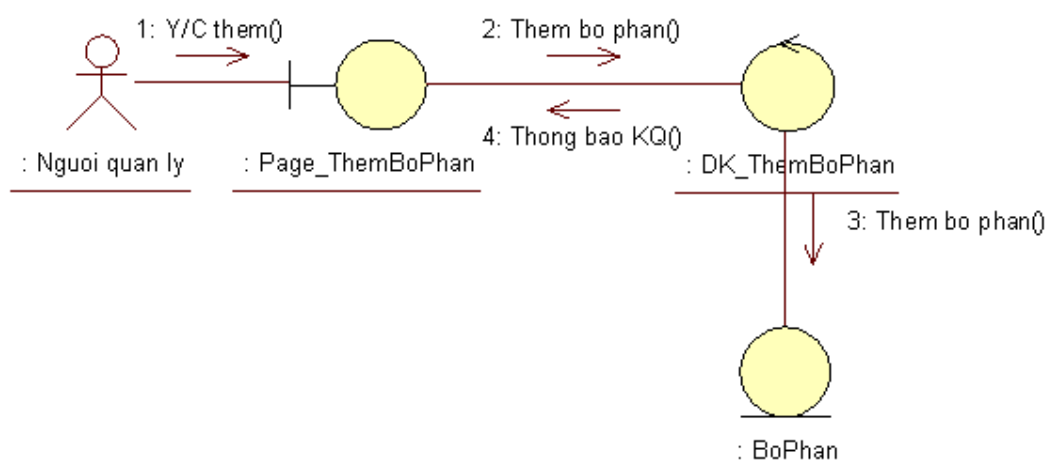
2.2.3.1.1. Ca sử dụng “Thêm bộ phận”

- **Biểu đồ tuần tự thực thi ca sử dụng**



Hình 2.7 : Biểu đồ tuần tự thực thi ca sử dụng “Thêm bộ phận”

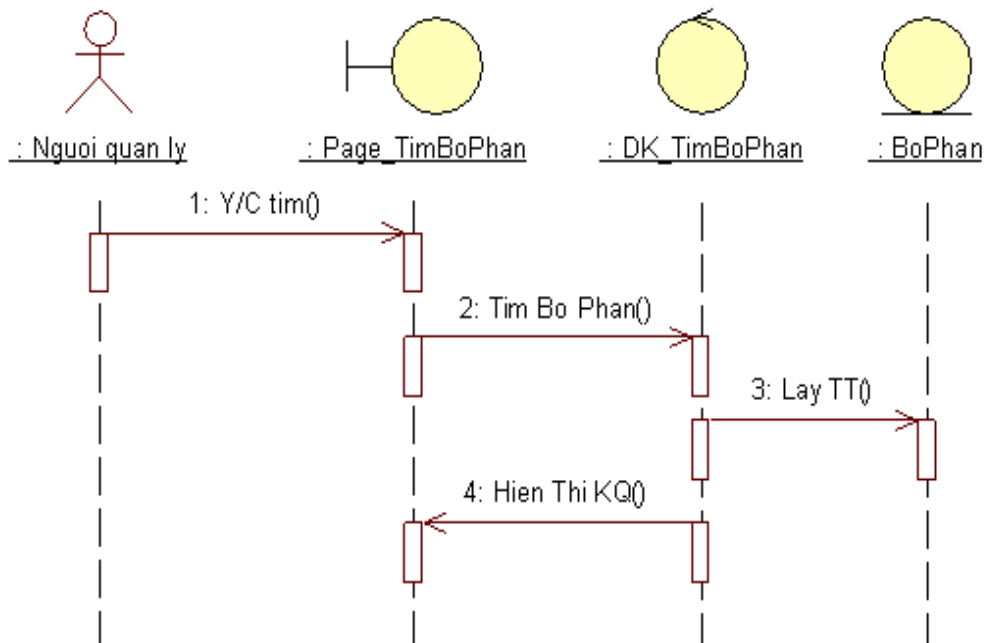
- **Biểu đồ cộng tác thực thi ca sử dụng**



Hình 2.8 : Biểu đồ cộng tác thực thi ca sử dụng “Thêm bộ phận”

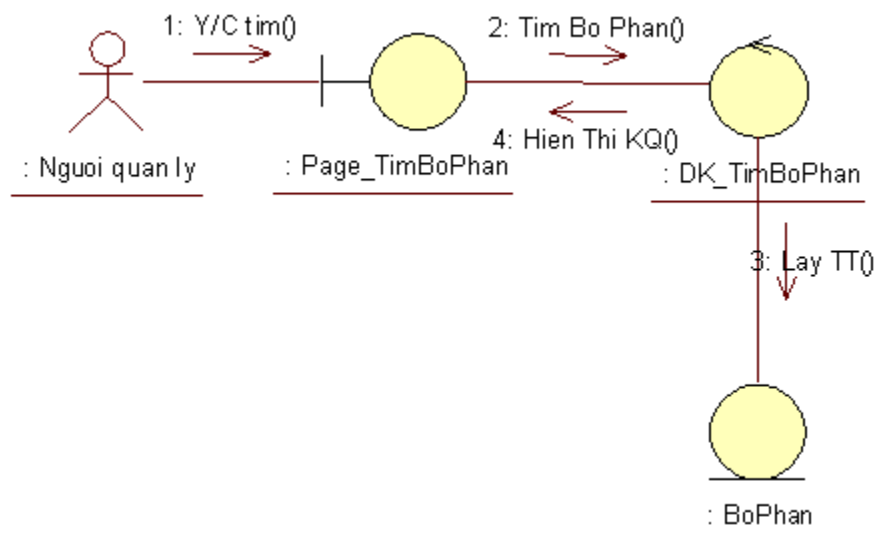
2.2.3.1.2. Ca sử dụng “Tìm kiếm bộ phận”

▪ Biểu đồ tuần tự thực thi ca sử dụng



Hình 2.9 : Biểu đồ tuần tự thực thi ca sử dụng “Tìm kiếm bộ phận”

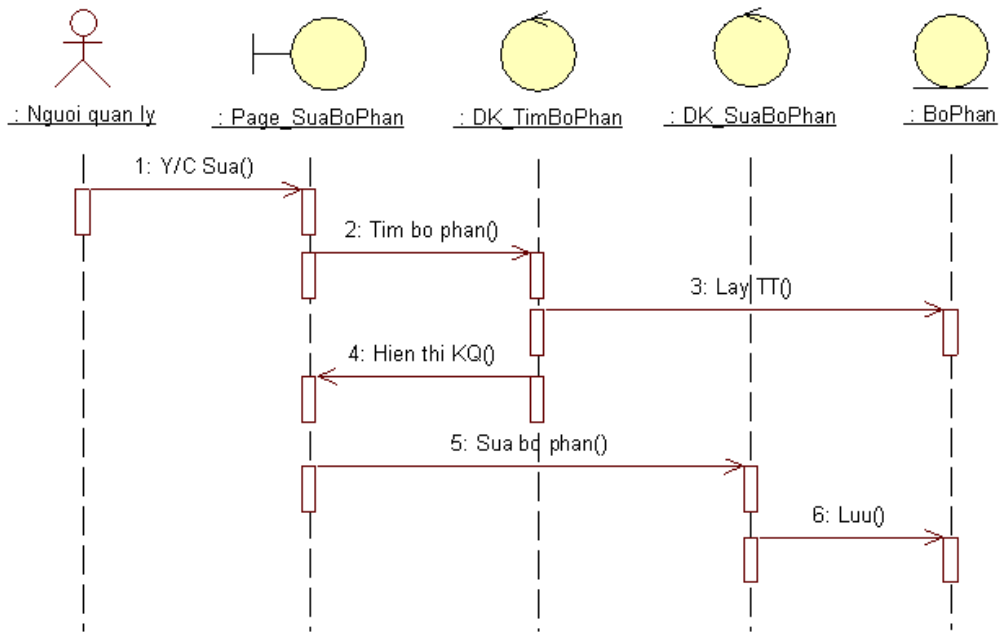
▪ Biểu đồ cộng tác thực thi ca sử dụng



Hình 2.10 : Biểu đồ cộng tác thực thi ca sử dụng “Tìm kiếm bộ phận”

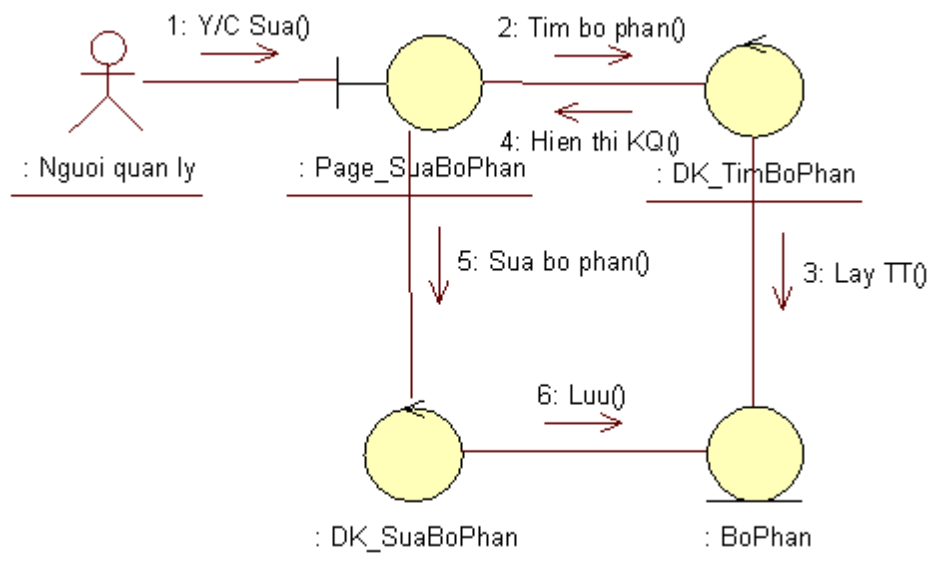
2.2.3.1.3. Ca sử dụng “Cập nhật bộ phận”

▪ Biểu đồ tuần tự thực thi ca sử dụng



Hình 2.11 : Biểu đồ tuần tự thực thi ca sử dụng “Cập nhật bộ phận”

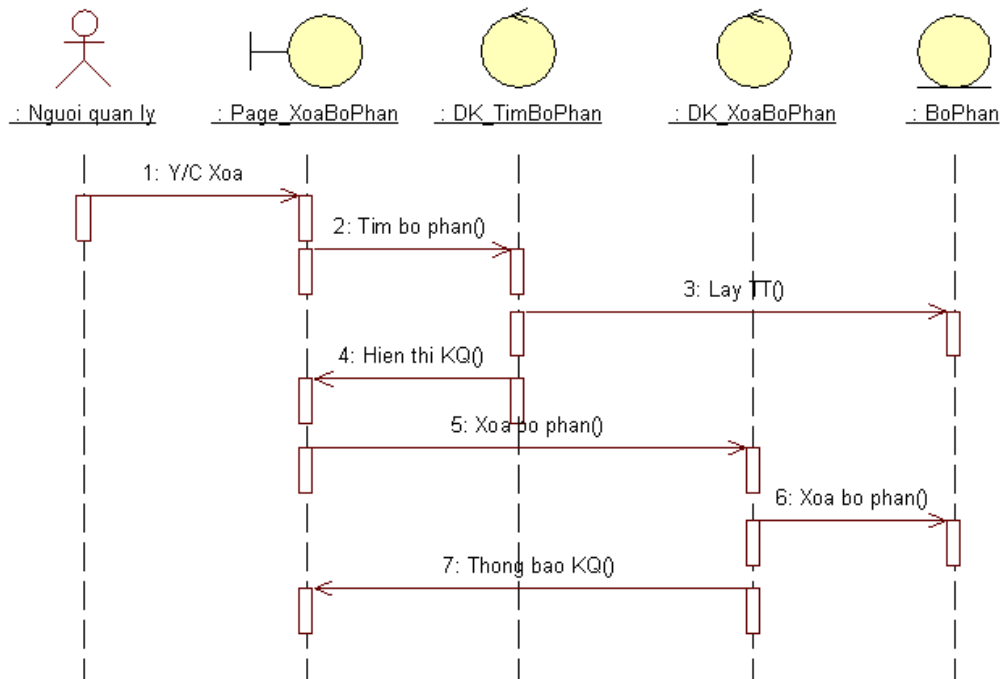
▪ **Biểu đồ cộng tác thực thi ca sử dụng**



Hình 2.12 : Biểu đồ cộng tác thực thi ca sử dụng “Cập nhật bộ phận”

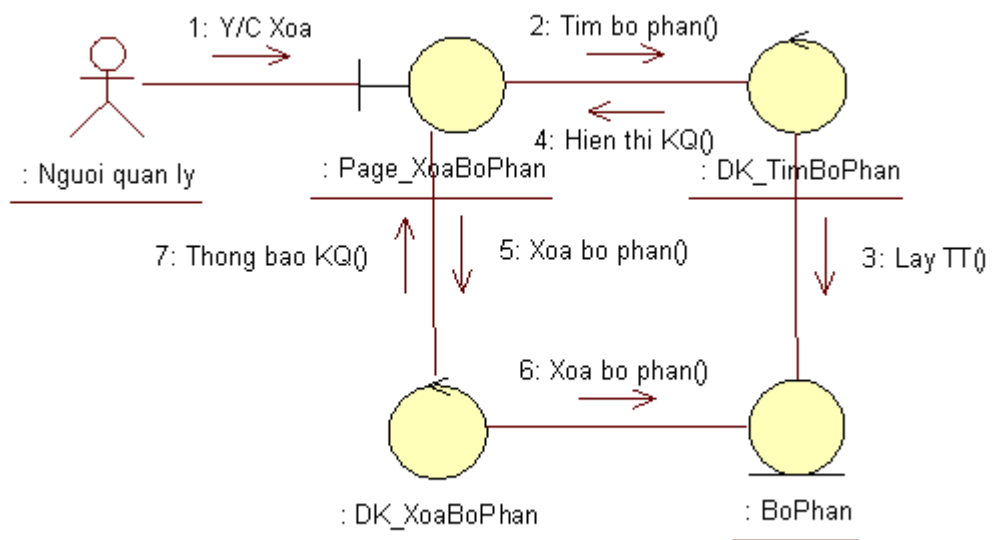
2.2.3.1.4. Ca sử dụng “Xóa bộ phận”

▪ Biểu đồ tuần tự thực thi ca sử dụng



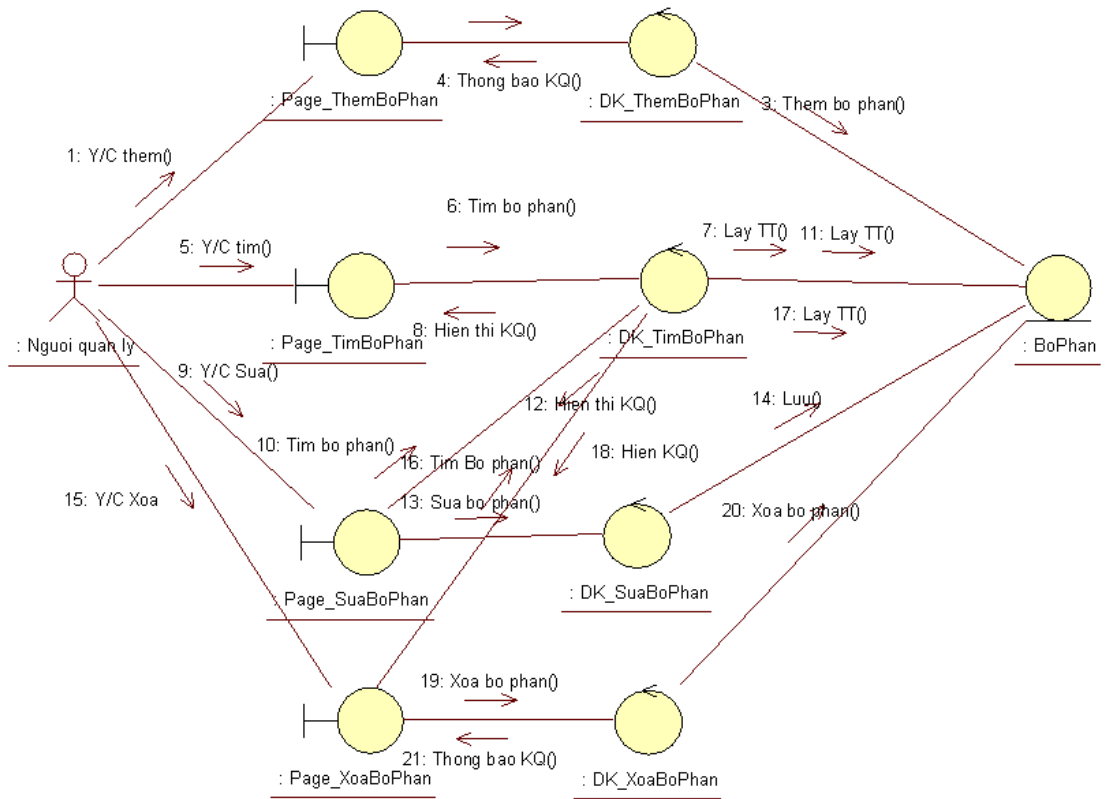
Hình 2.13 : Biểu đồ cộng tác thực thi ca sử dụng “Xóa bộ phận”

▪ Biểu đồ cộng tác thực thi ca sử dụng



Hình 2.14 : Biểu đồ cộng tác thực thi ca sử dụng “Xóa bộ phận”

Mô hình phân tích gói ca “Quản lý bộ phận”

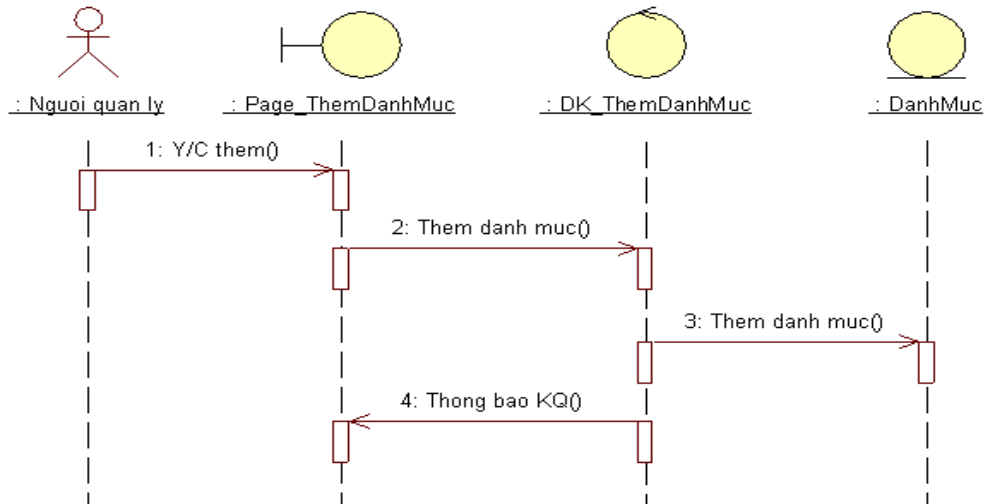


Hình 2.15 : Mô hình phân tích gói ca “Quản lý bộ phận”

2.2.3.2. Phân tích gói ca sử dụng “Quản lý danh mục”

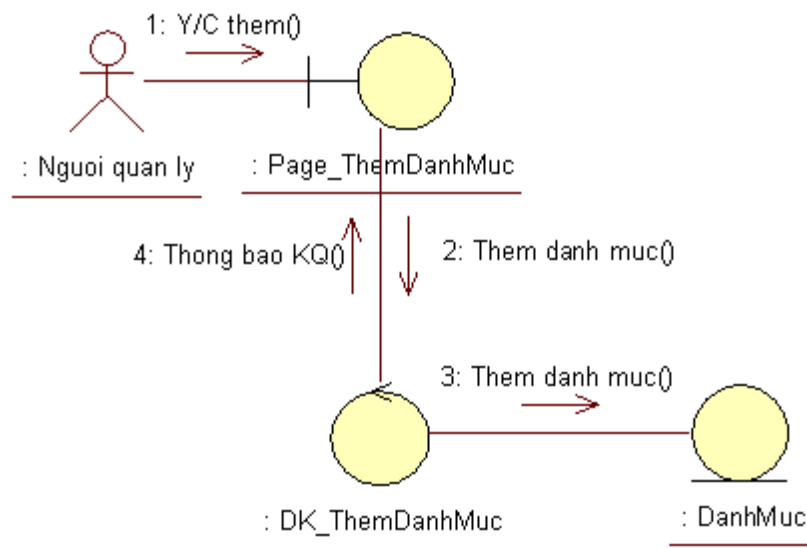
2.2.3.2.1. Ca sử dụng “Thêm danh mục”

▪ **Biểu đồ tuần tự thực thi ca sử dụng**



Hình 2.16 : Biểu đồ tuần tự thực thi ca sử dụng “Thêm danh mục”

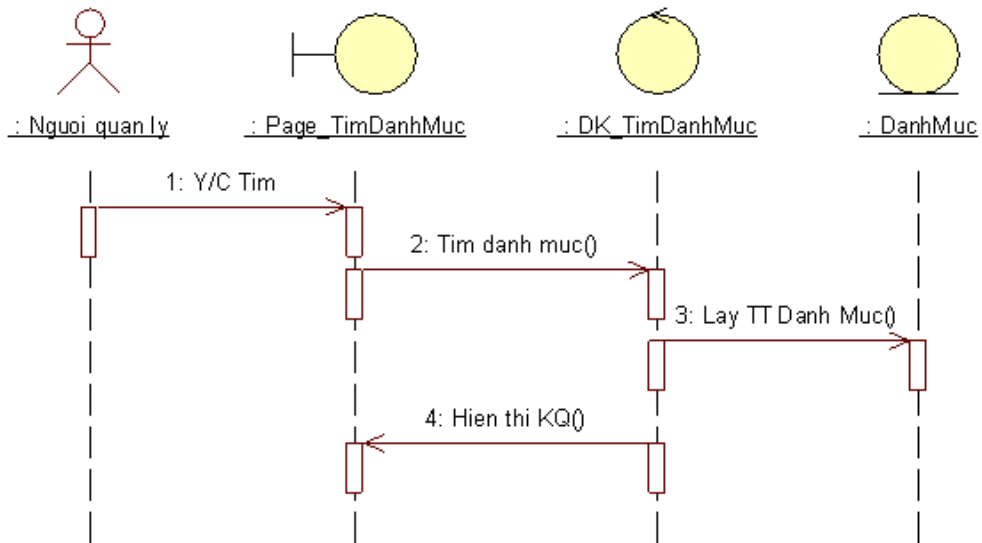
▪ **Biểu đồ cộng tác thực thi ca sử dụng**



Hình 2.17 : Biểu đồ cộng tác thực thi ca sử dụng “Thêm danh mục”

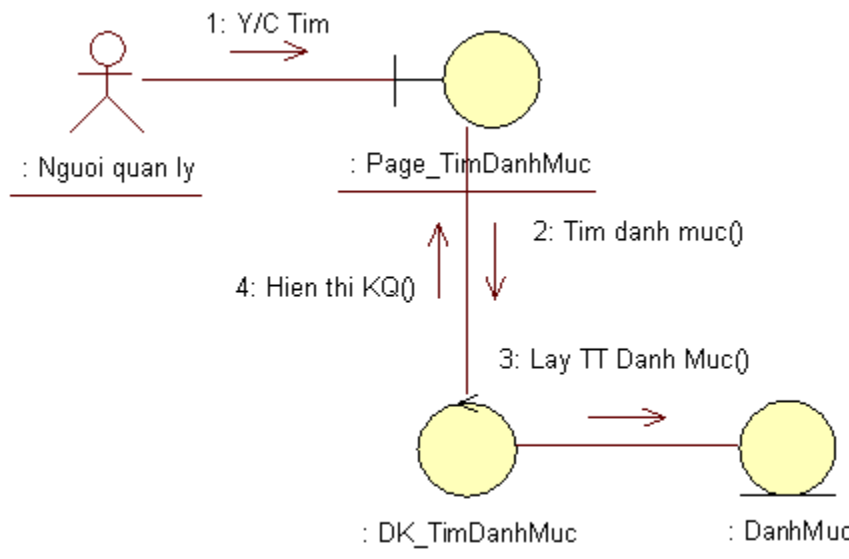
3.2.2.2.2. *Ca sử dụng “Tìm kiếm danh mục”*

▪ **Biểu đồ tuần tự thực thi ca sử dụng**



Hình 2.18 : *Biểu đồ tuần tự thực thi ca sử dụng “Tìm kiếm danh mục”*

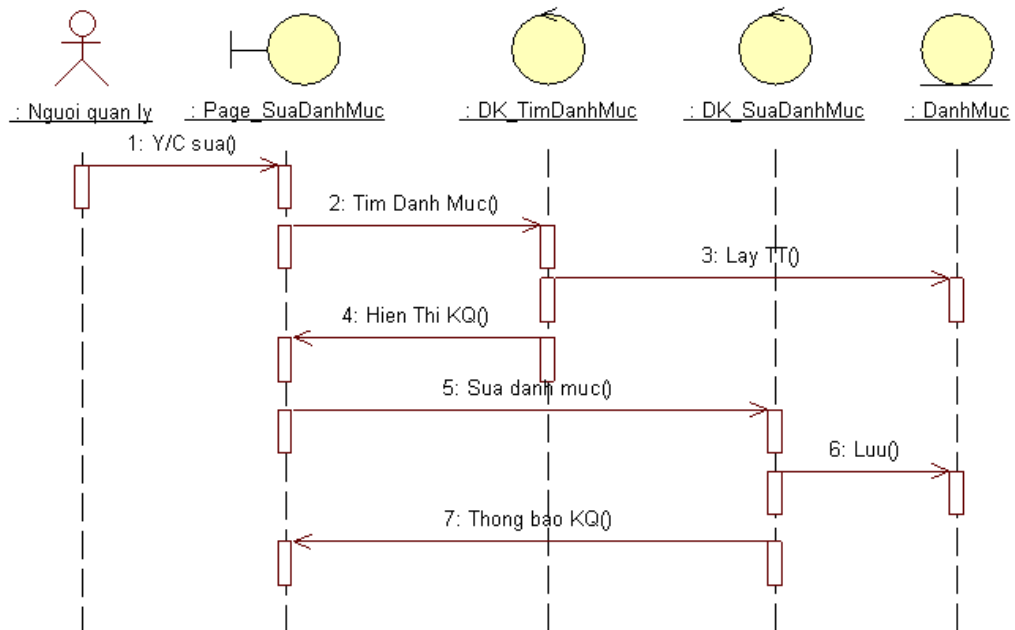
▪ **Biểu đồ cộng tác thực thi ca sử dụng**



Hình 2.19 : *Biểu đồ cộng tác thực thi ca sử dụng “Tìm kiếm danh mục”*

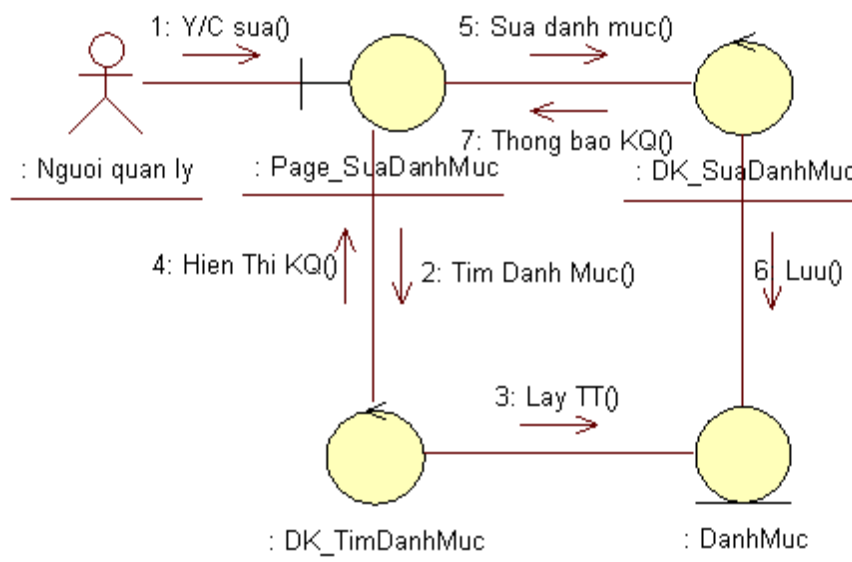
2.2.3.2.3. Ca sử dụng “Cập nhật danh mục”

▪ Biểu đồ tuần tự thực thi ca sử dụng



Hình 2.20 : Biểu đồ tuần tự thực thi ca sử dụng “Cập nhật danh mục”

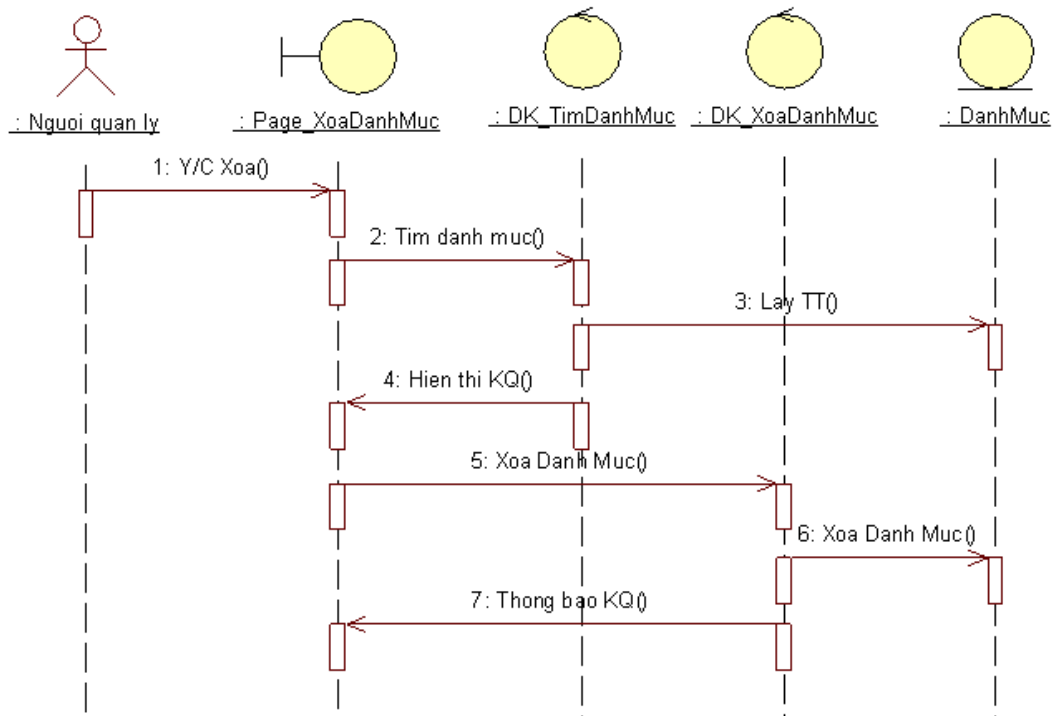
▪ Biểu đồ cộng tác thực thi ca sử dụng



Hình 2.21 : Biểu đồ cộng tác thực thi ca sử dụng “Cập nhật danh mục”

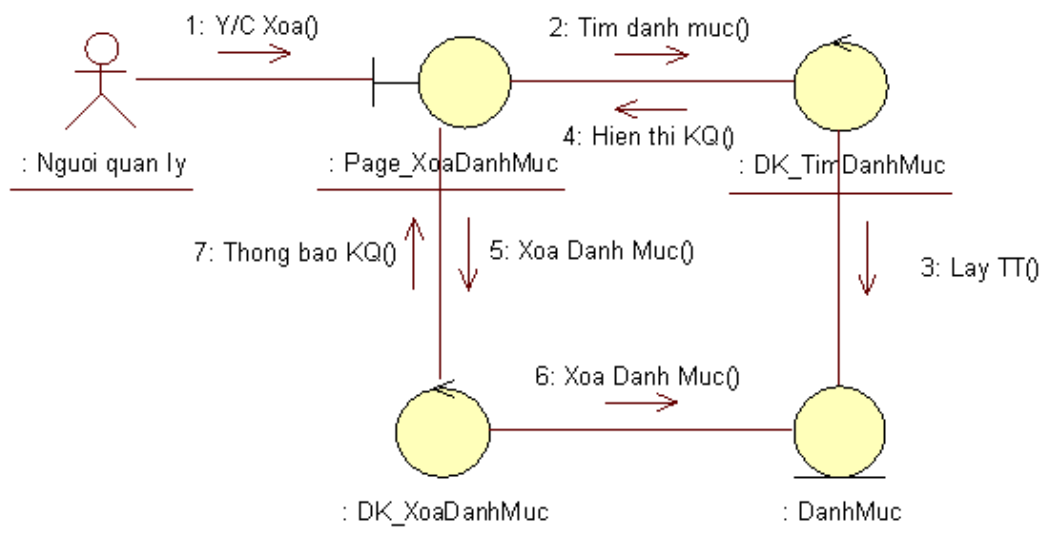
2.2.3.1.4. Ca sử dụng “Xóa danh mục”

▪ Biểu đồ tuần tự thực thi ca sử dụng



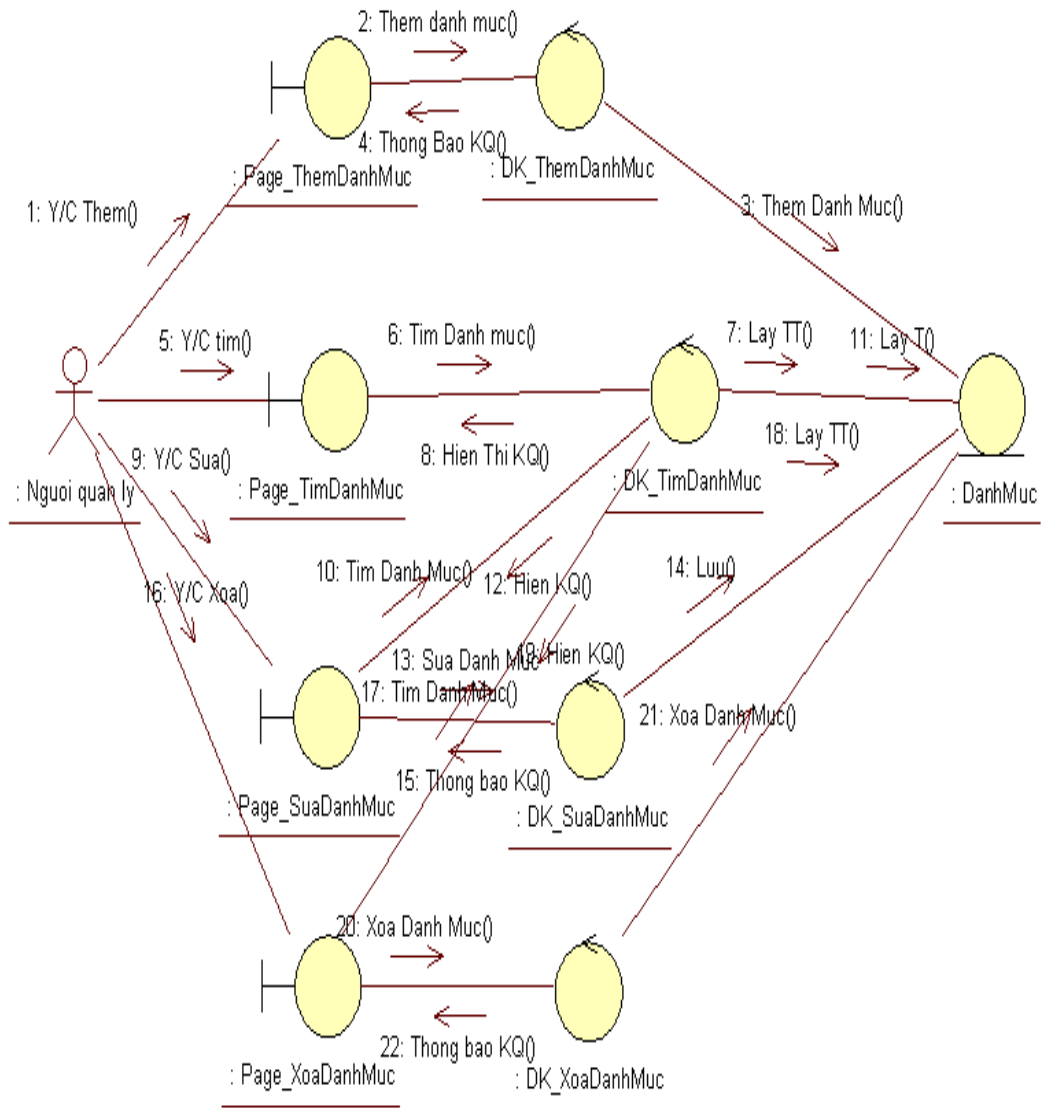
Hình 2.22 : Biểu đồ tuần tự thực thi ca sử dụng “Xóa danh mục”

▪ Biểu đồ cộng tác thực thi ca sử dụng



Hình 2.23 : Biểu đồ cộng tác thực thi ca sử dụng “Xóa danh mục”

Mô hình phân tích gói ca “Quản lý danh mục”

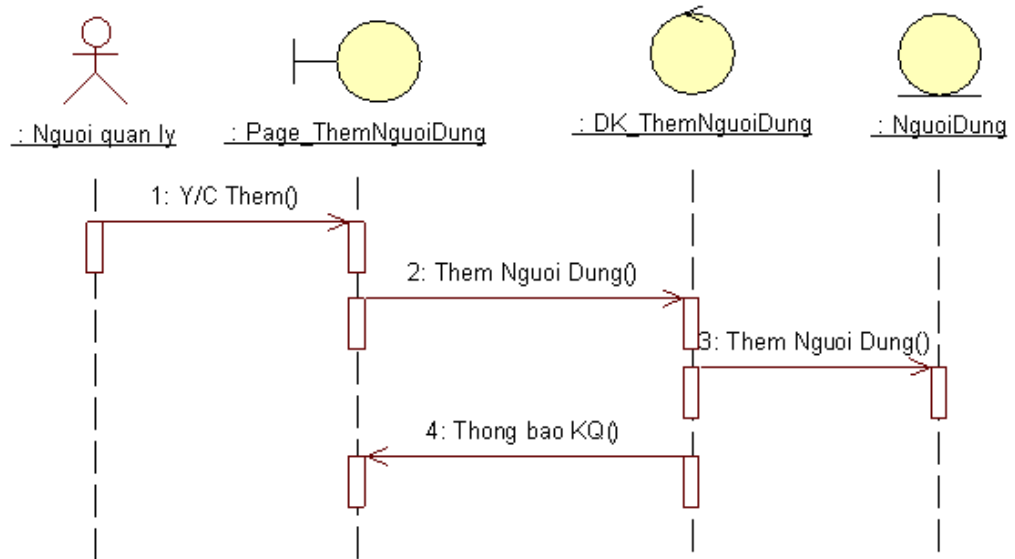


Hình 2.24 : Mô hình phân tích gói ca “Quản lý danh mục”

2.2.3.2. Phân tích gói ca sử dụng “Quản lý người dùng”

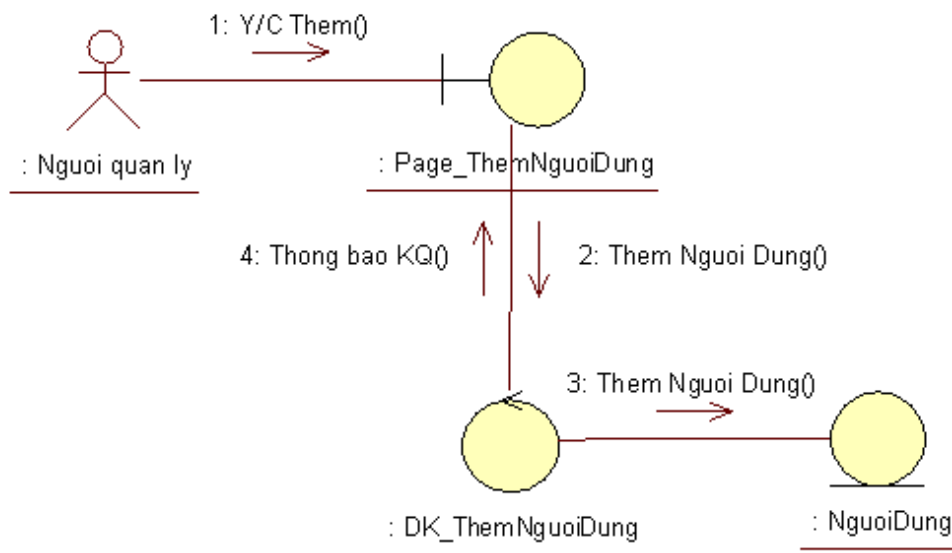
2.2.3.2.1. Ca sử dụng “Thêm người dùng”

▪ **Biểu đồ tuần tự thực thi ca sử dụng**



Hình 2.25 : Biểu đồ tuần tự thực thi ca sử dụng “Thêm người dùng”

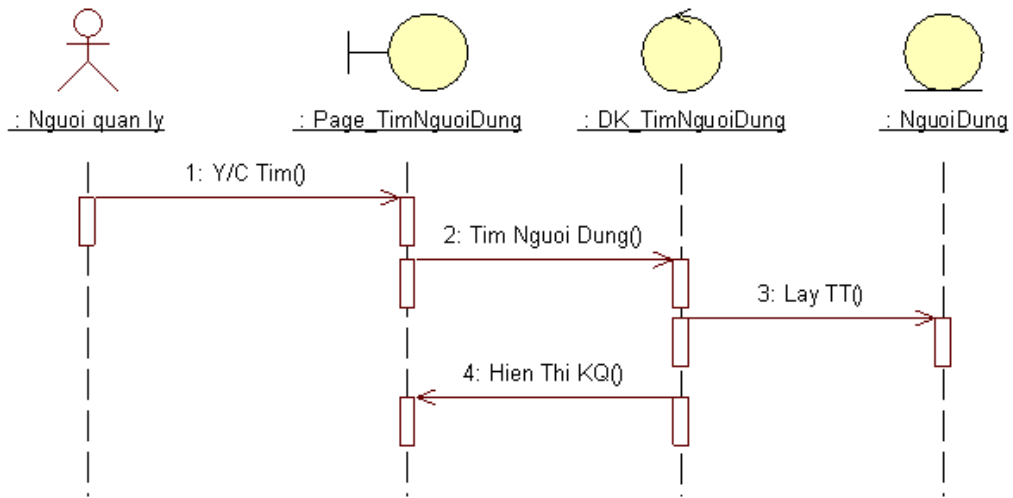
▪ **Biểu đồ cộng tác thực thi ca sử dụng**



Hình 2.26 : Biểu đồ cộng tác thực thi ca sử dụng “Thêm người dùng”

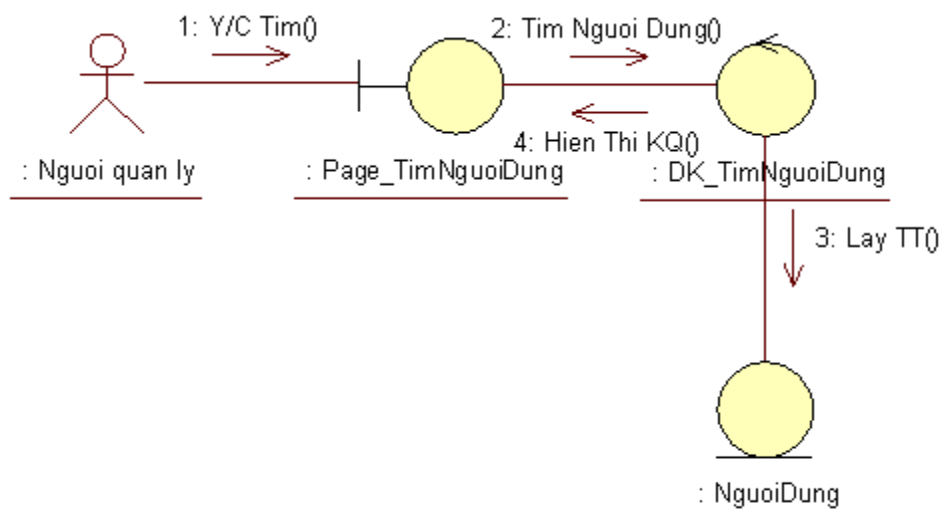
3.2.2.2.2. *Ca sử dụng “Tìm kiếm người dùng”*

▪ **Biểu đồ tuần tự thực thi ca sử dụng**



Hình 2.27 : *Biểu đồ tuần tự thực thi ca sử dụng “Tìm kiếm người dùng”*

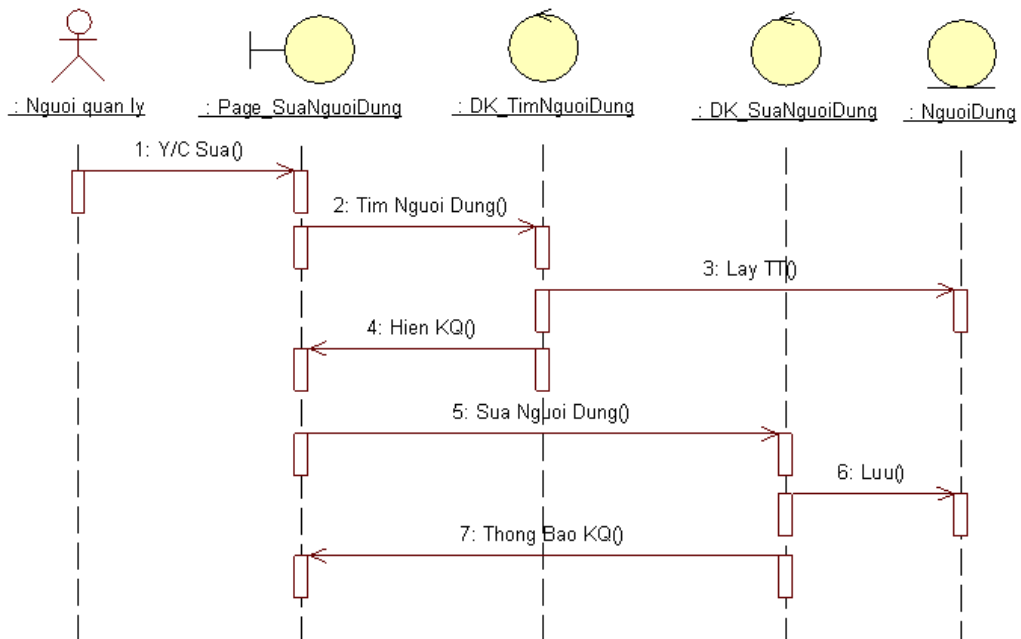
▪ **Biểu đồ cộng tác thực thi ca sử dụng**



Hình 2.28 : *Biểu đồ cộng tác thực thi ca sử dụng “Tìm kiếm người dùng”*

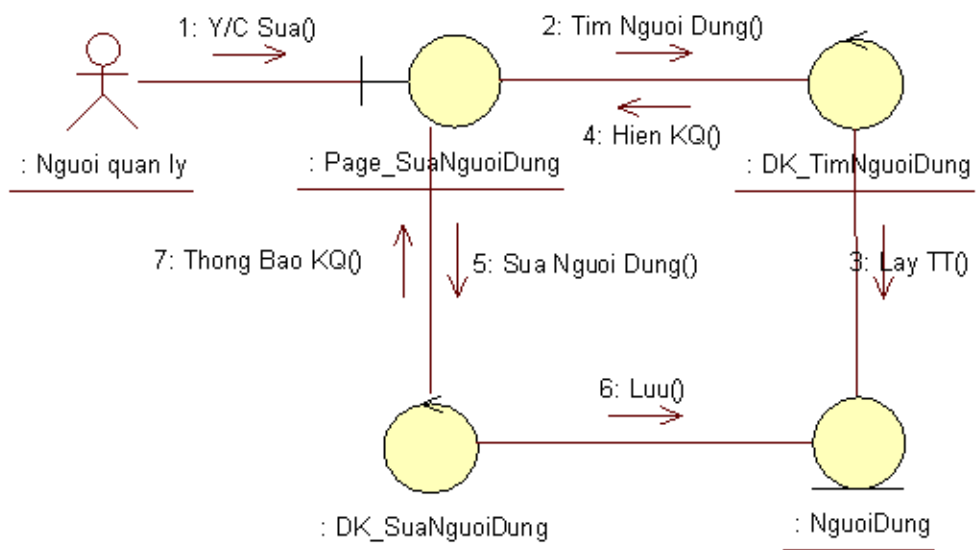
2.2.3.2.3. Ca sử dụng “Cập nhật người dùng”

▪ Biểu đồ tuần tự thực thi ca sử dụng



Hình 2.29 : Biểu đồ tuần tự thực thi ca sử dụng “Cập nhật người dùng”

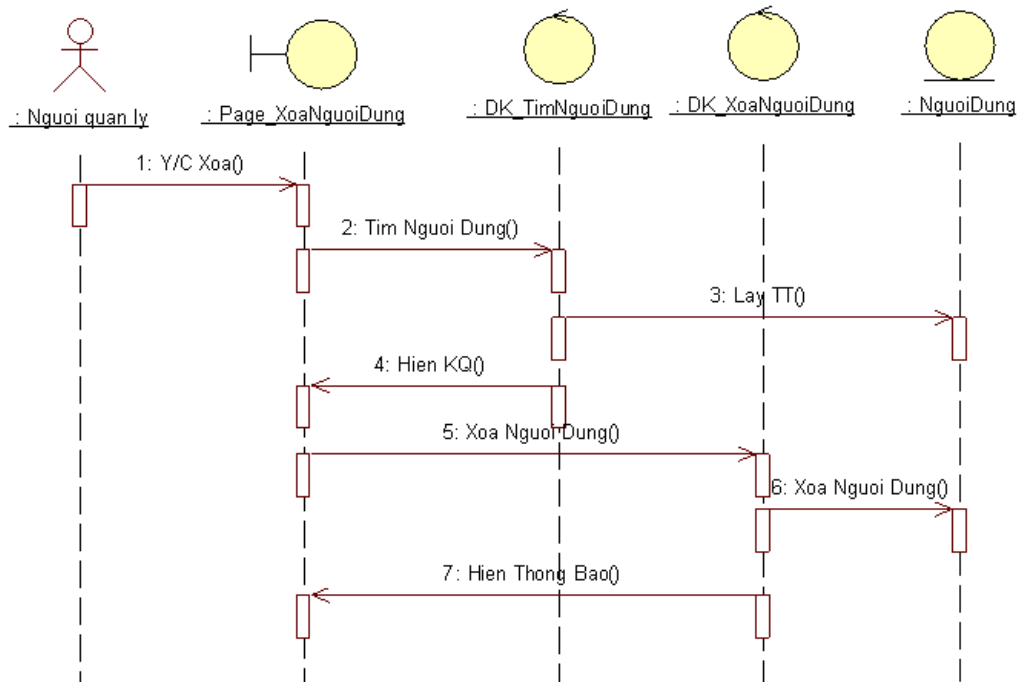
▪ Biểu đồ cộng tác thực thi ca sử dụng



Hình 2.30: Biểu đồ cộng tác thực thi ca sử dụng “Cập nhật người dùng”

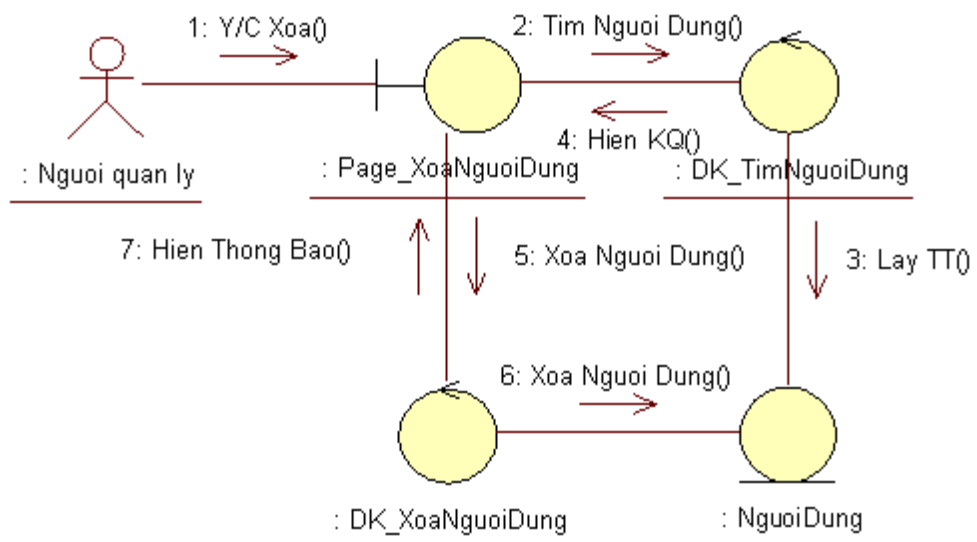
2.2.3.1.4. Ca sử dụng “Xóa người dùng”

▪ Biểu đồ tuần tự thực thi ca sử dụng



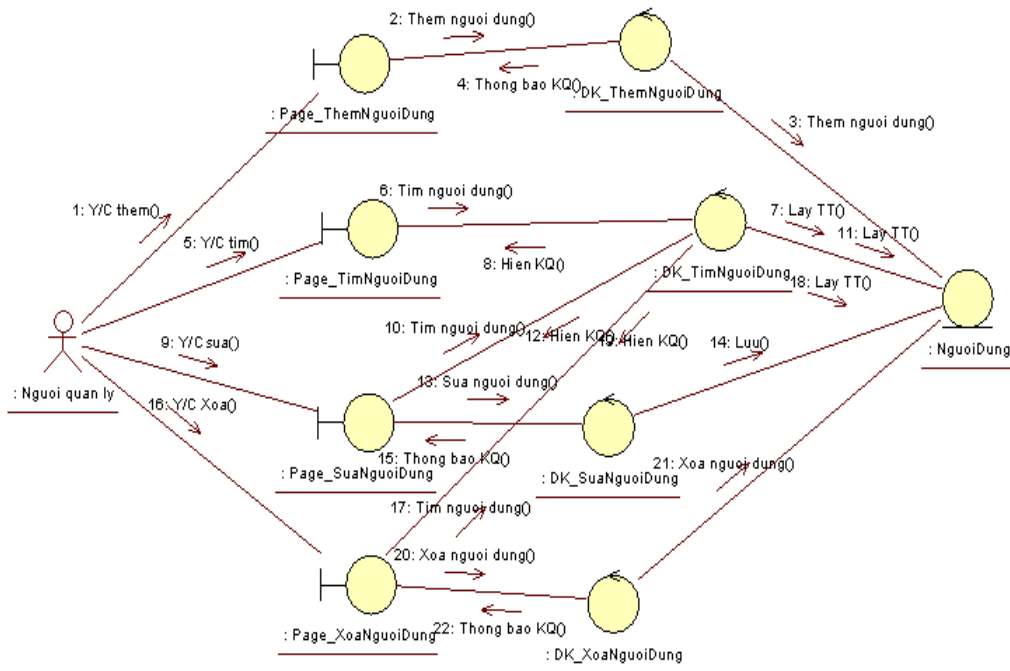
Hình 2.31 : Biểu đồ tuần tự thực thi ca sử dụng “Xóa người dùng”

▪ Biểu đồ cộng tác thực thi ca sử dụng



Hình 2.32 : Biểu đồ cộng tác thực thi ca sử dụng “Xóa người dùng”

Mô hình phân tích gói ca “Quản lý người dùng”

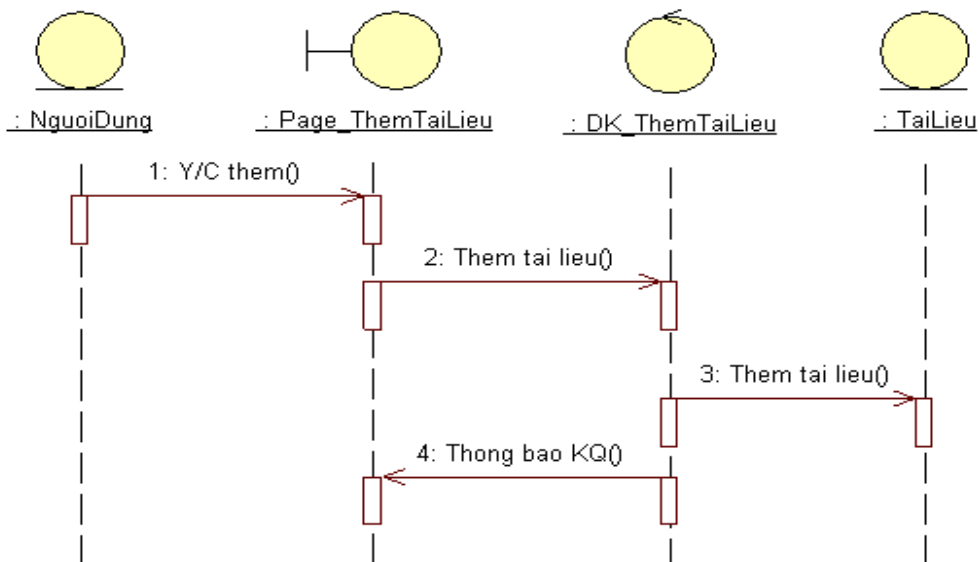


Hình 2.33 : Mô hình phân tích gói ca “Quản lý người dùng”

2.2.3.3. Phân tích gói ca sử dụng “Quản lý tài liệu”

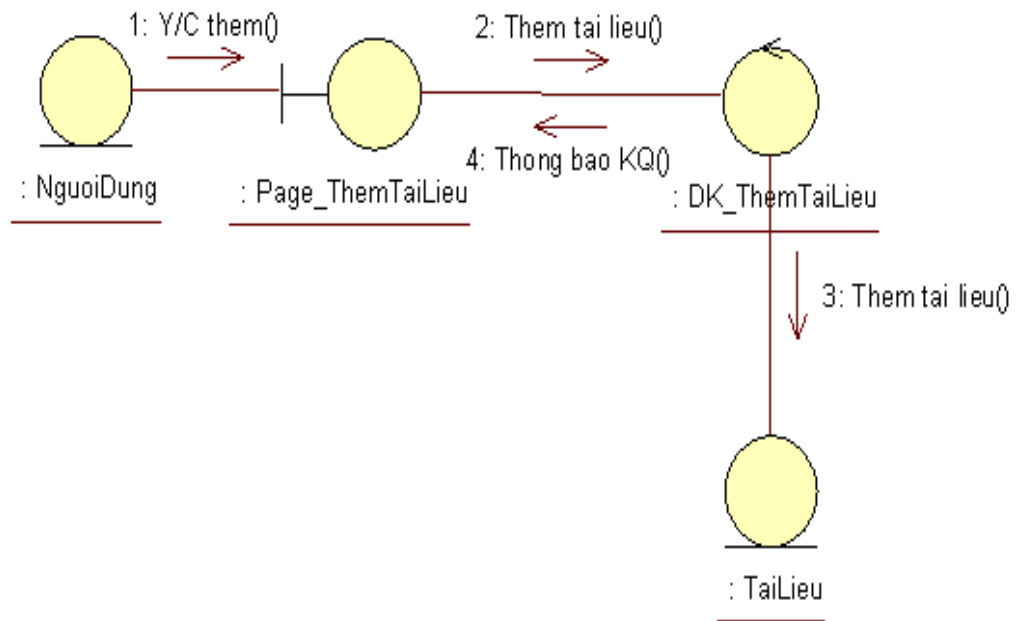
2.2.3.2.1. Ca sử dụng “Thêm tài liệu”

- **Biểu đồ tuần tự thực thi ca sử dụng**



Hình 2.34 : Biểu đồ tuần tự thực thi ca sử dụng “Thêm tài liệu”

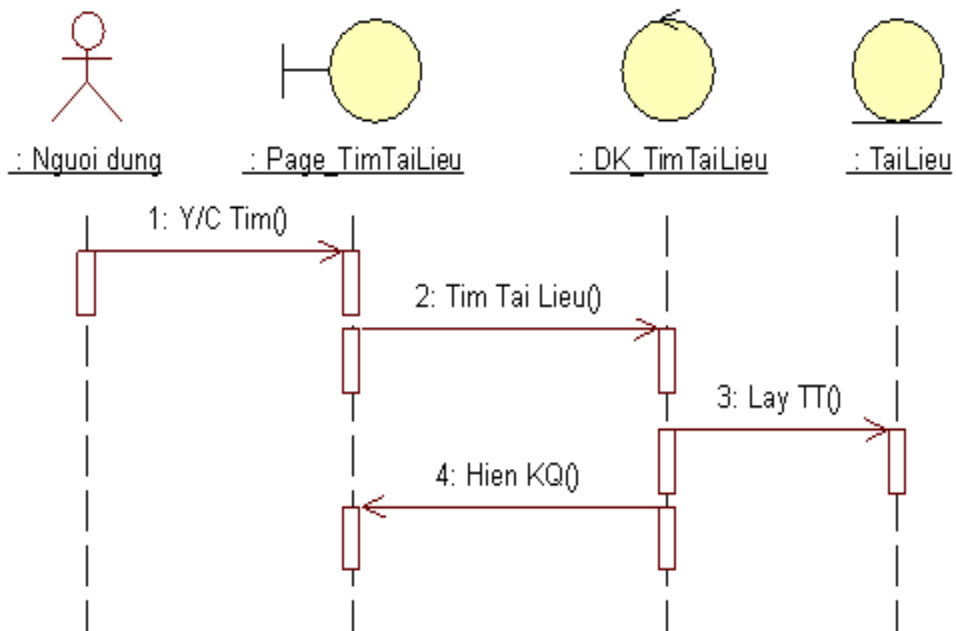
▪ **Biểu đồ cộng tác thực thi ca sử dụng**



Hình 2.35 : Biểu đồ cộng tác thực thi ca sử dụng “Thêm tài liệu”

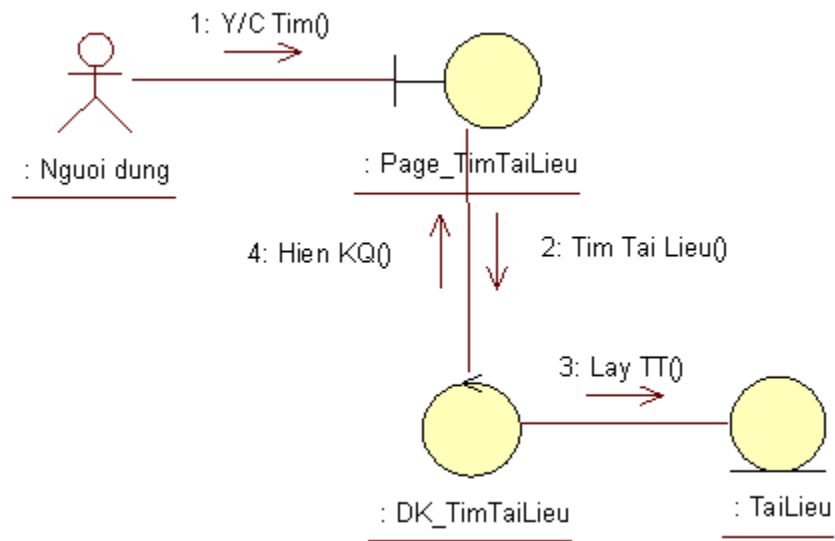
3.2.2.2.2. Ca sử dụng “Tìm kiếm tài liệu”

▪ **Biểu đồ tuần tự thực thi ca sử dụng**



Hình 2.36 : Biểu đồ tuần tự thực thi ca sử dụng “Tìm kiếm tài liệu”

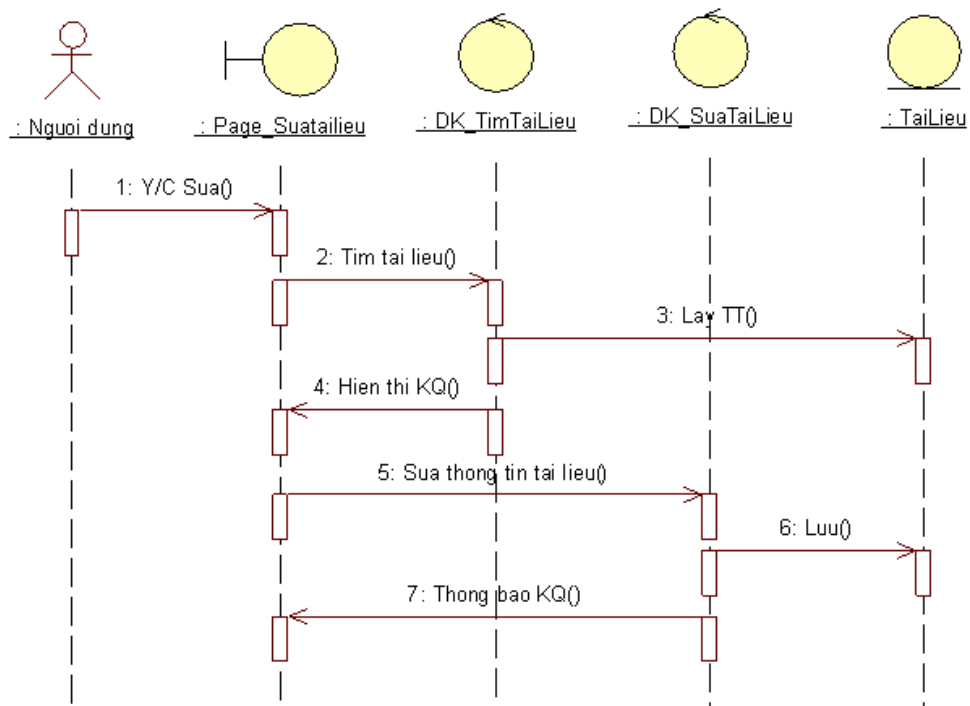
▪ **Biểu đồ cộng tác thực thi ca sử dụng**



Hình 2.37 : Biểu đồ cộng tác thực thi ca sử dụng “Tìm kiếm tài liệu”

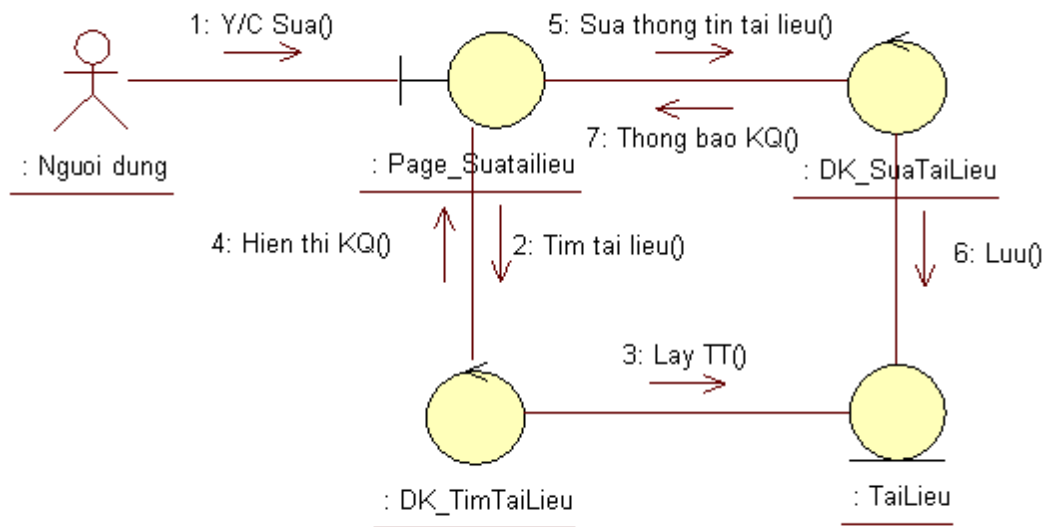
2.2.3.2.3. Ca sử dụng “Cập nhật tài liệu”

▪ **Biểu đồ tuần tự thực thi ca sử dụng**



Hình 2.38 : Biểu đồ tuần tự thực thi ca sử dụng “Cập nhật tài liệu”

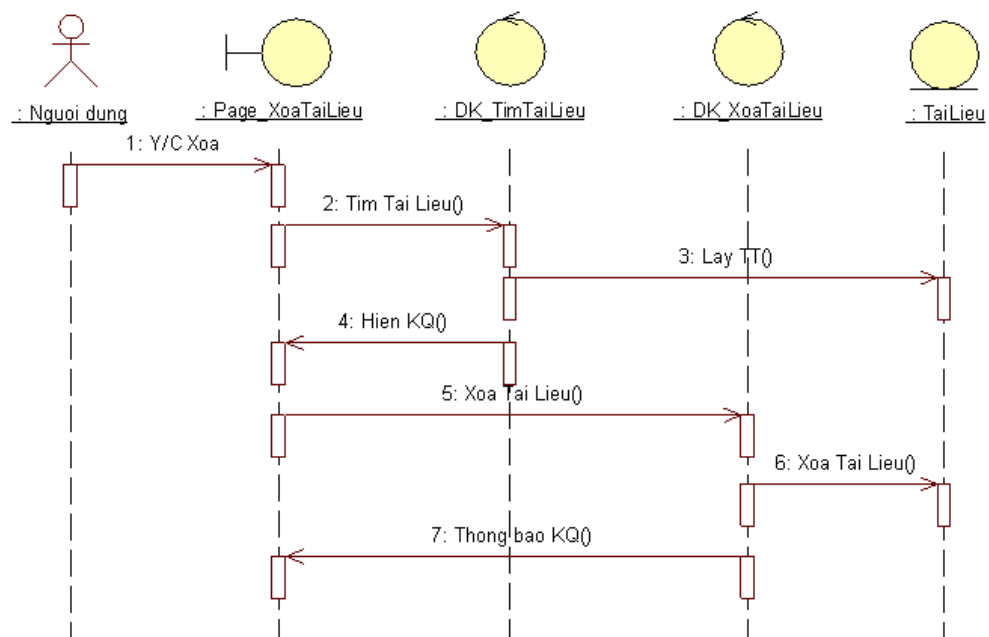
▪ **Biểu đồ cộng tác thực thi ca sử dụng**



Hình 2.39 : Biểu đồ cộng tác thực thi ca sử dụng “Cập nhật tài liệu”

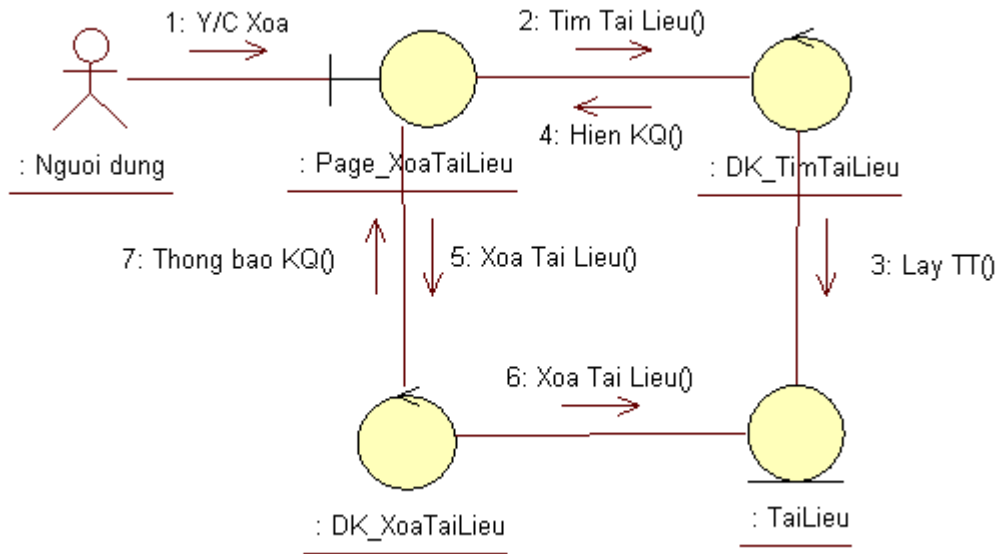
2.2.3.1.4. Ca sử dụng “Xóa tài liệu”

▪ **Biểu đồ tuần tự thực thi ca sử dụng**



Hình 2.40 : Biểu đồ tuần tự thực thi ca sử dụng “Xóa tài liệu”

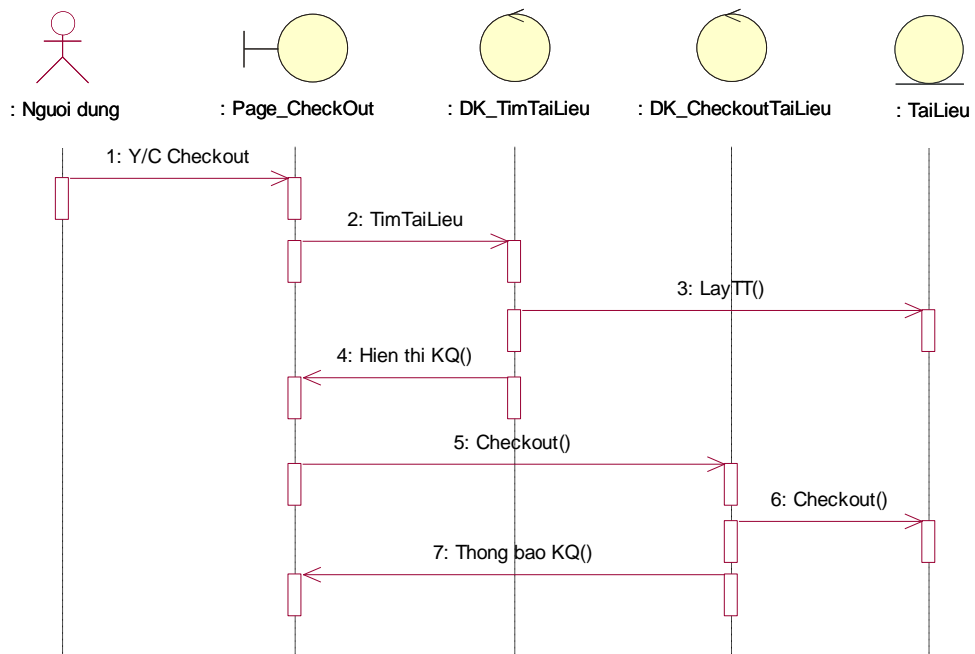
▪ **Biểu đồ cộng tác thực thi ca sử dụng**



Hình 2.41 : Biểu đồ cộng tác thực thi ca sử dụng “Xóa tài liệu”

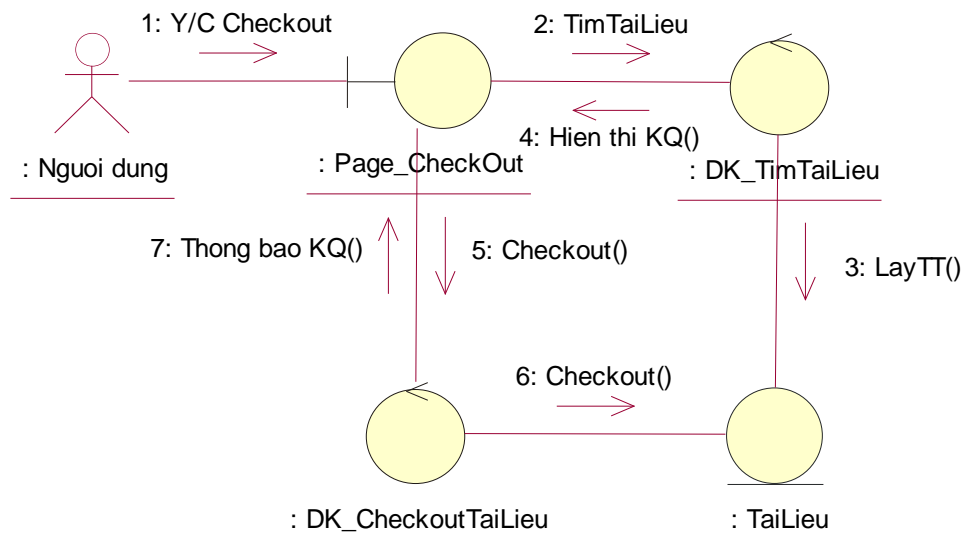
2.2.3.1.5. Ca sử dụng “Checkout tài liệu”

▪ **Biểu đồ tuần tự thực thi ca sử dụng**



Hình 2.42 : Biểu đồ tuần tự thực thi ca sử dụng “Checkout tài liệu”

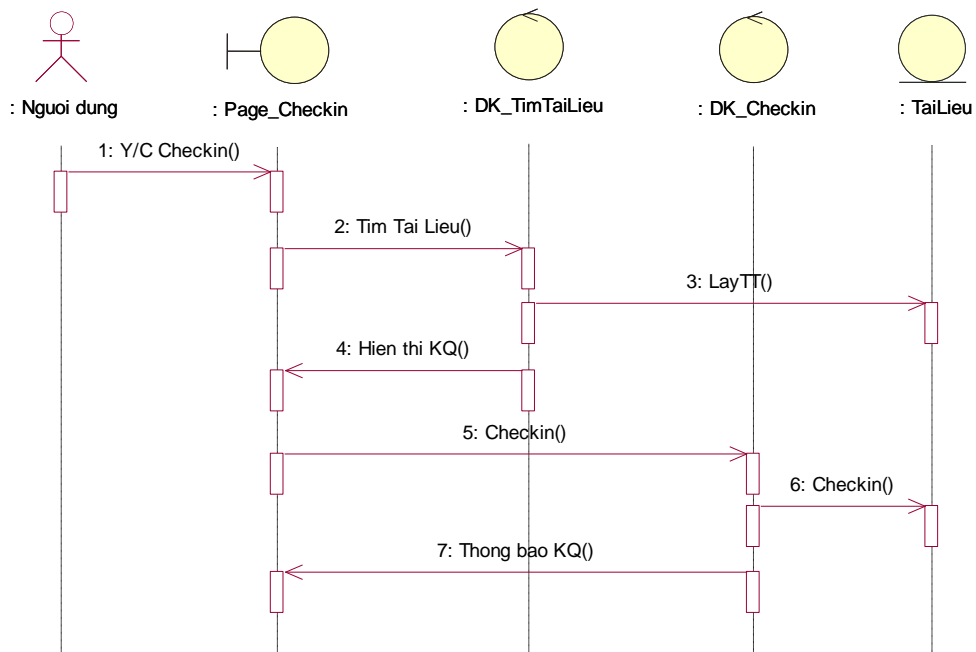
▪ **Biểu đồ cộng tác thực thi ca sử dụng**



Hình 2.43 : Biểu đồ cộng tác thực thi ca sử dụng “Checkout tài liệu”

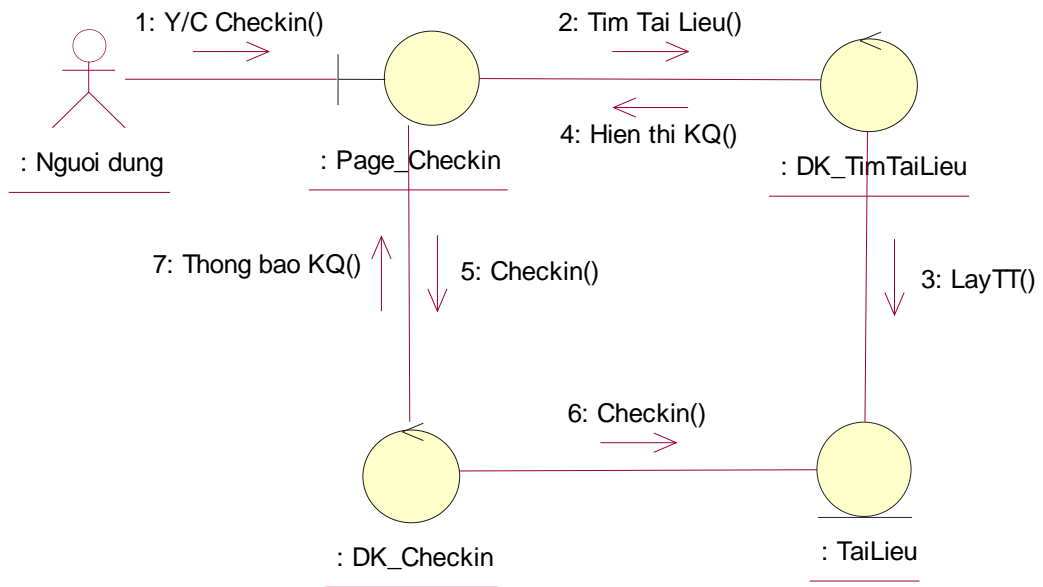
2.2.3.1.6. Ca sử dụng “Checkin tài liệu”

▪ **Biểu đồ tuần tự thực thi ca sử dụng**



Hình 2.44 : Biểu đồ tuần tự thực thi ca sử dụng “Checkin tài liệu”

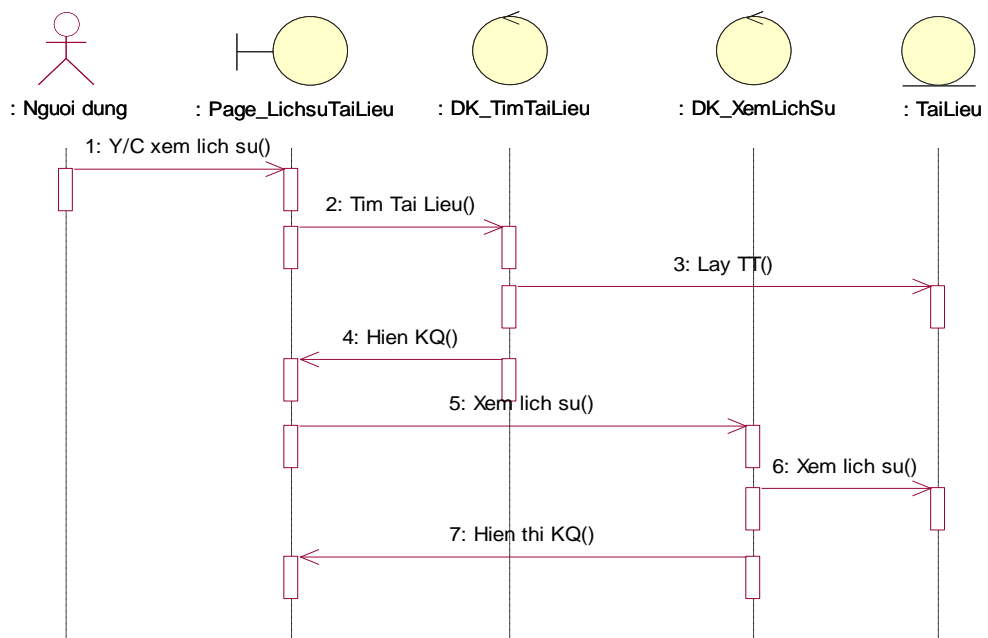
▪ **Biểu đồ cộng tác thực thi ca sử dụng**



Hình 2.45 : Biểu đồ cộng tác thực thi ca sử dụng “Checkin tài liệu”

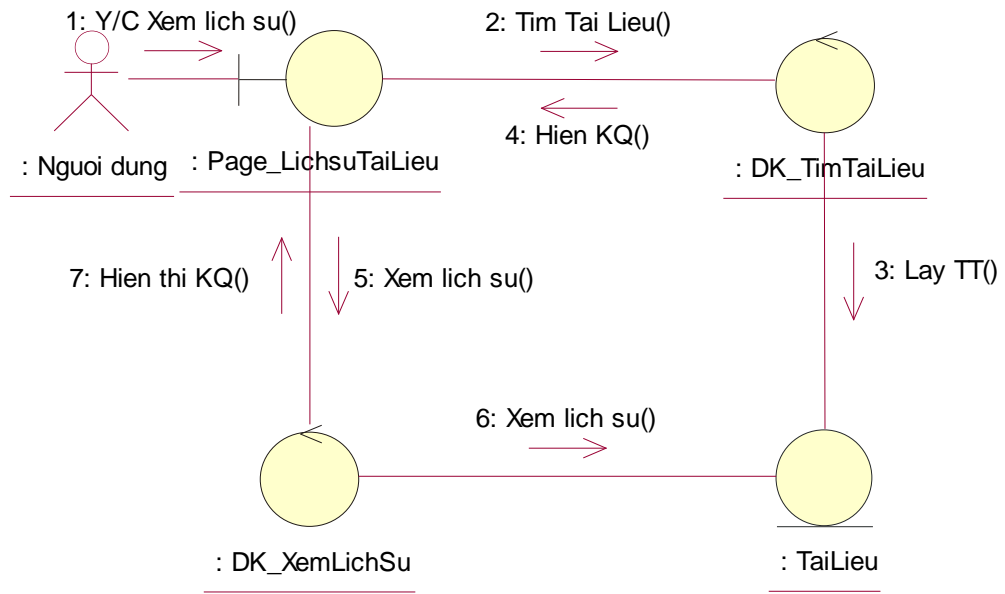
2.2.3.1.8. Ca sử dụng “Xem lịch sử tài liệu”

▪ **Biểu đồ tuần tự thực thi ca sử dụng**



Hình 2.46 : Biểu đồ tuần tự thực thi ca sử dụng “Xem lịch sử tài liệu”

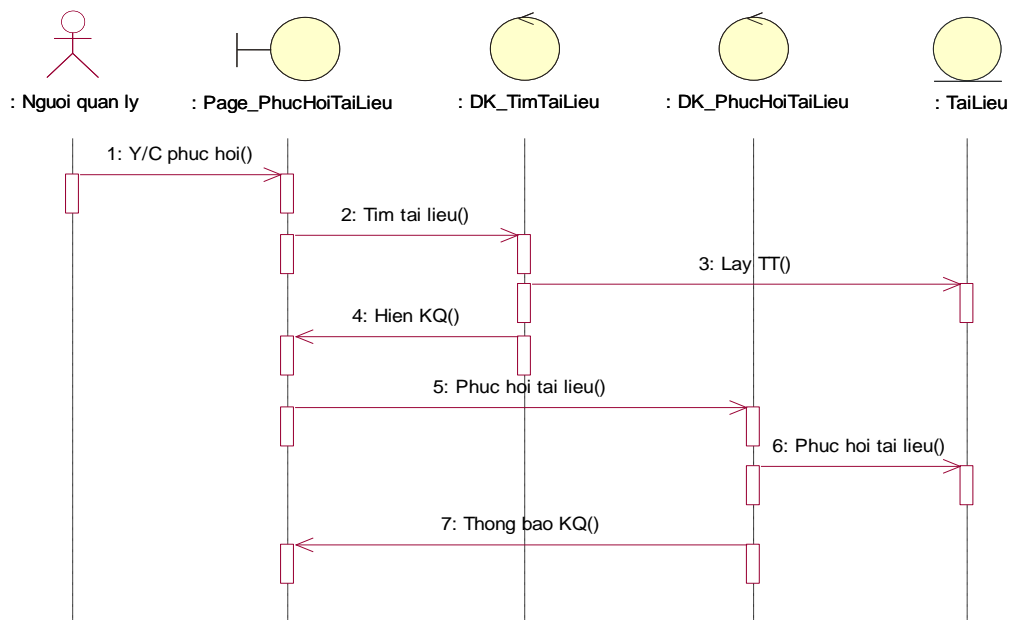
▪ **Biểu đồ cộng tác thực thi ca sử dụng**



Hình 2.47 : Biểu đồ cộng tác thực thi ca sử dụng “Xem lịch sử tài liệu”

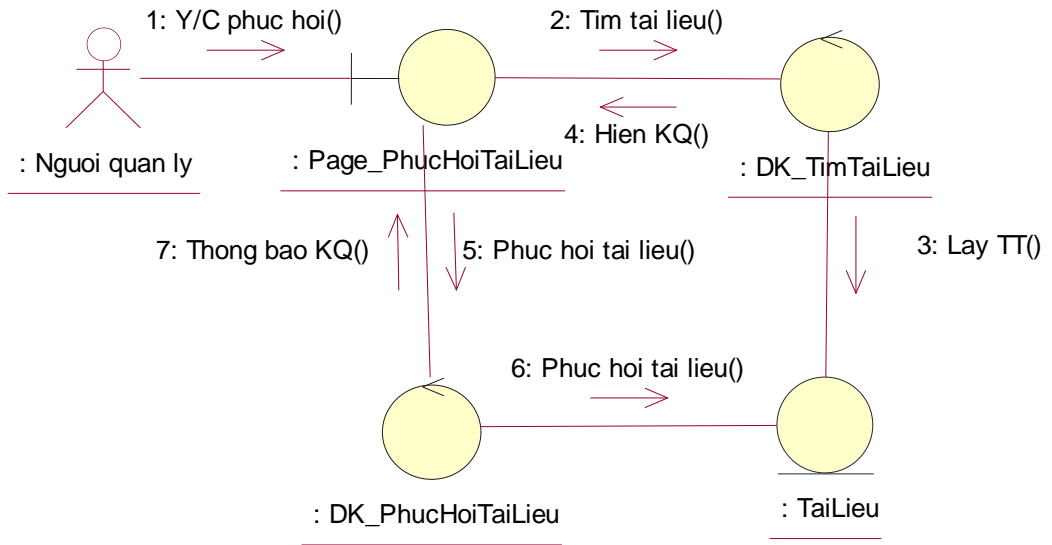
2.2.3.1.8. Ca sử dụng “Phục hồi tài liệu”

▪ **Biểu đồ tuần tự thực thi ca sử dụng**



Hình 2.48 : Biểu đồ tuần tự thực thi ca sử dụng “Phục hồi tài liệu”

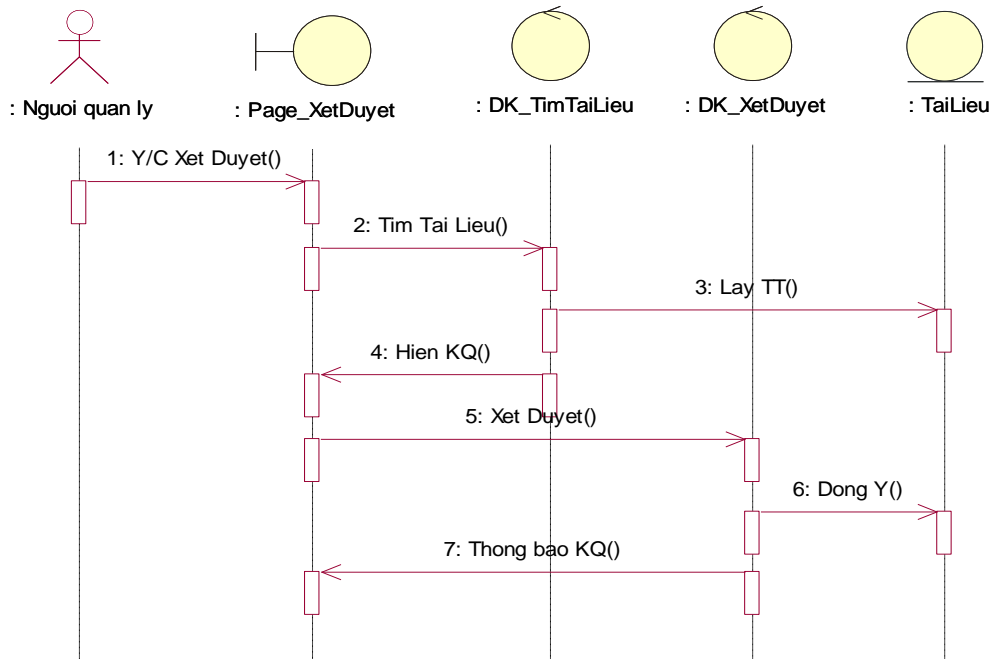
▪ **Biểu đồ cộng tác thực thi ca sử dụng**



Hình 2.49 : Biểu đồ cộng tác thực thi ca sử dụng “Phục hồi tài liệu”

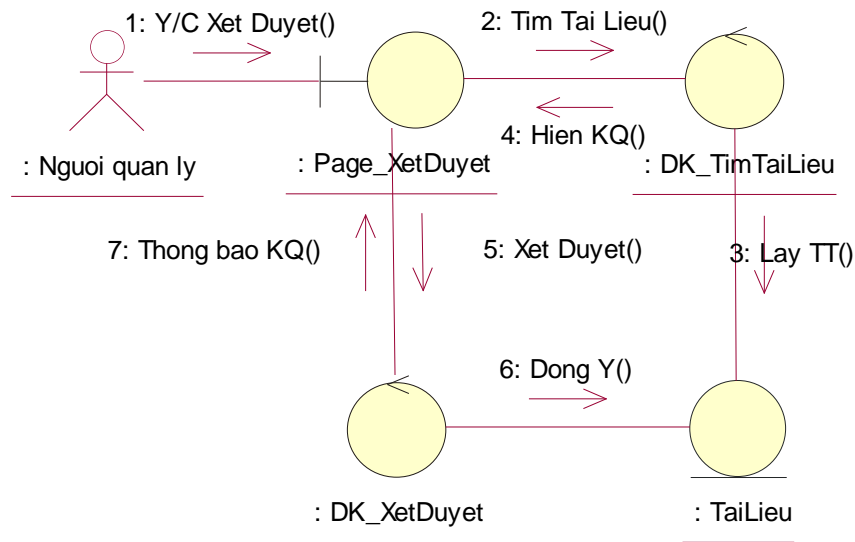
2.2.3.1.9. Ca sử dụng “Xét duyệt tài liệu khi thêm”

▪ **Biểu đồ tuần tự thực thi ca sử dụng**



Hình 2.50 : Biểu đồ tuần tự thực thi ca sử dụng “Xét duyệt tài liệu”

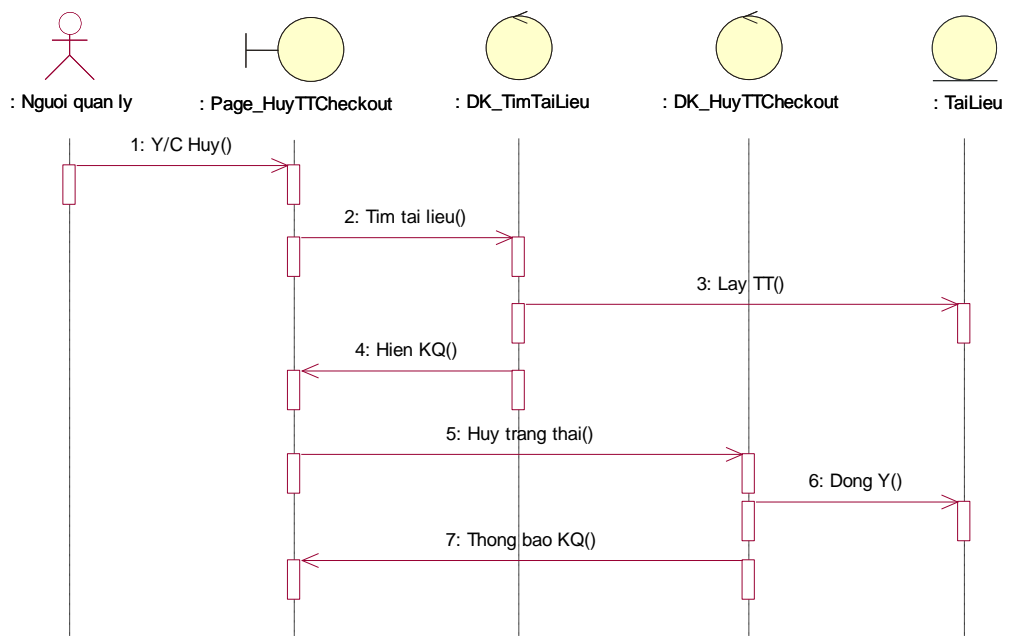
▪ **Biểu đồ cộng tác thực thi ca sử dụng**



Hình 2.51 : Biểu đồ cộng tác thực thi ca sử dụng “Xét duyệt tài liệu”

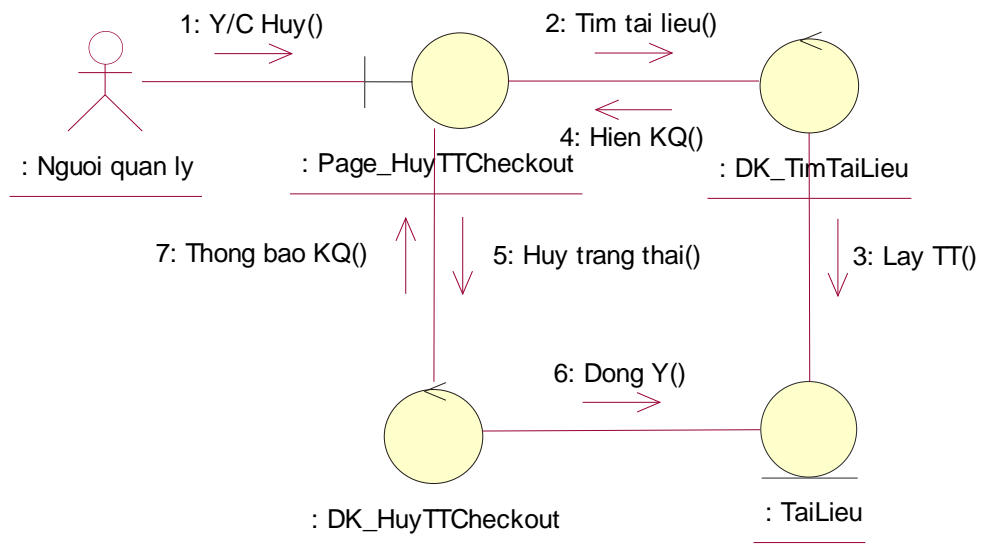
2.2.3.1.10. Ca sử dụng “Hủy trạng thái checkout”

▪ **Biểu đồ tuần tự thực thi ca sử dụng**



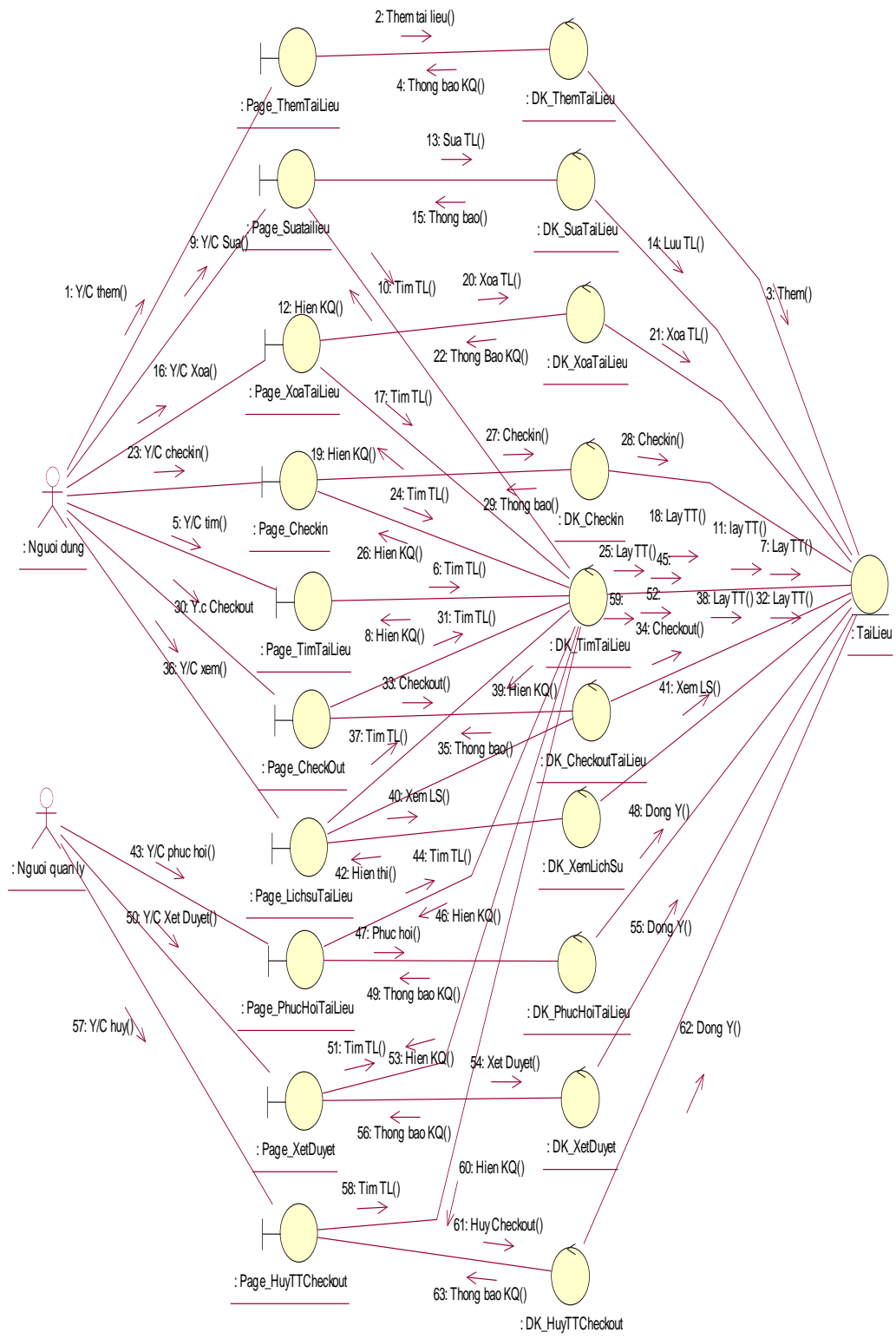
Hình 2.52 : Biểu đồ tuần tự thực thi ca sử dụng “Hủy trạng thái checkout”

▪ **Biểu đồ cộng tác thực thi ca sử dụng**



Hình 2.53 : Biểu đồ cộng tác thực thi ca sử dụng “Hủy trạng thái checkout”

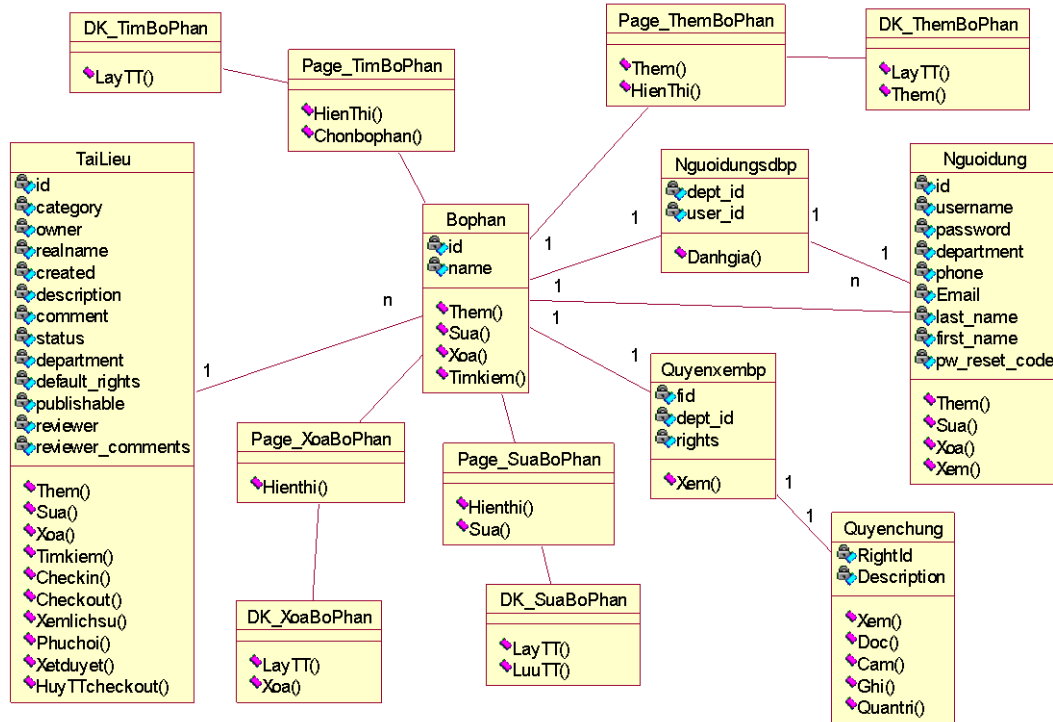
Mô hình phân tích gói ca “Quản lý tài liệu”



Hình 2.54 : Mô hình phân tích gói ca “Quản lý tài liệu”

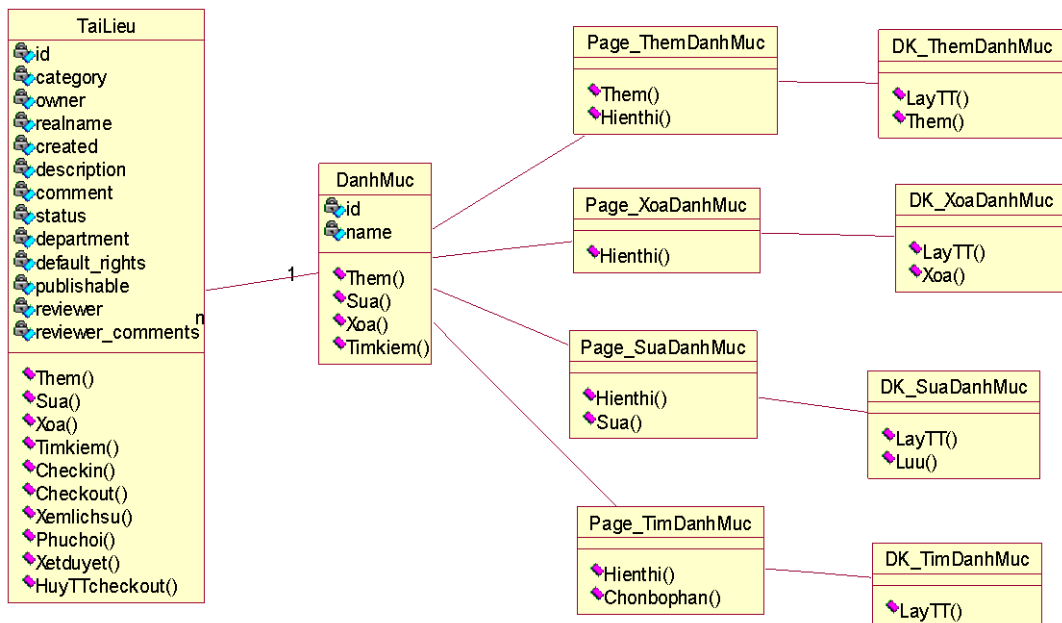
2.2.4. Thiết kế hệ thống

2.2.4.1. Thiết kế hệ thống “Quản lý bộ phận”



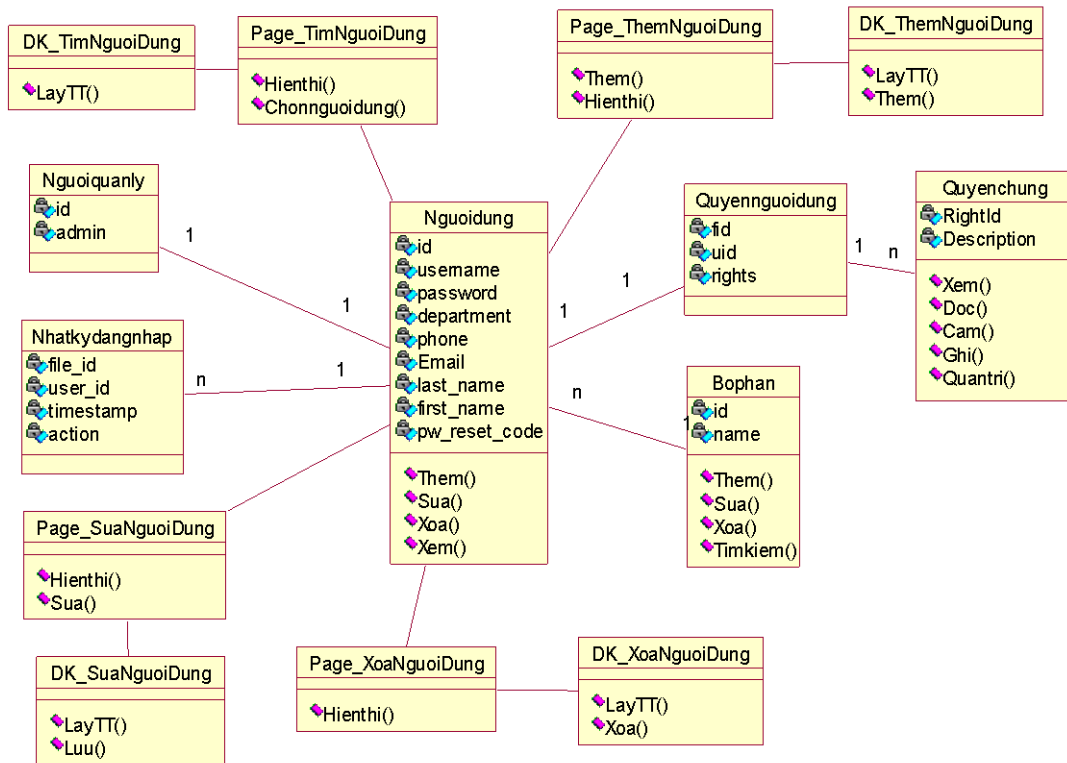
Hình 2.55 : Mô hình lớp thiết kế gói ca sử dụng “Quản lý bộ phận”

2.2.3.2. Thiết kế hệ thống “Quản lý danh mục”



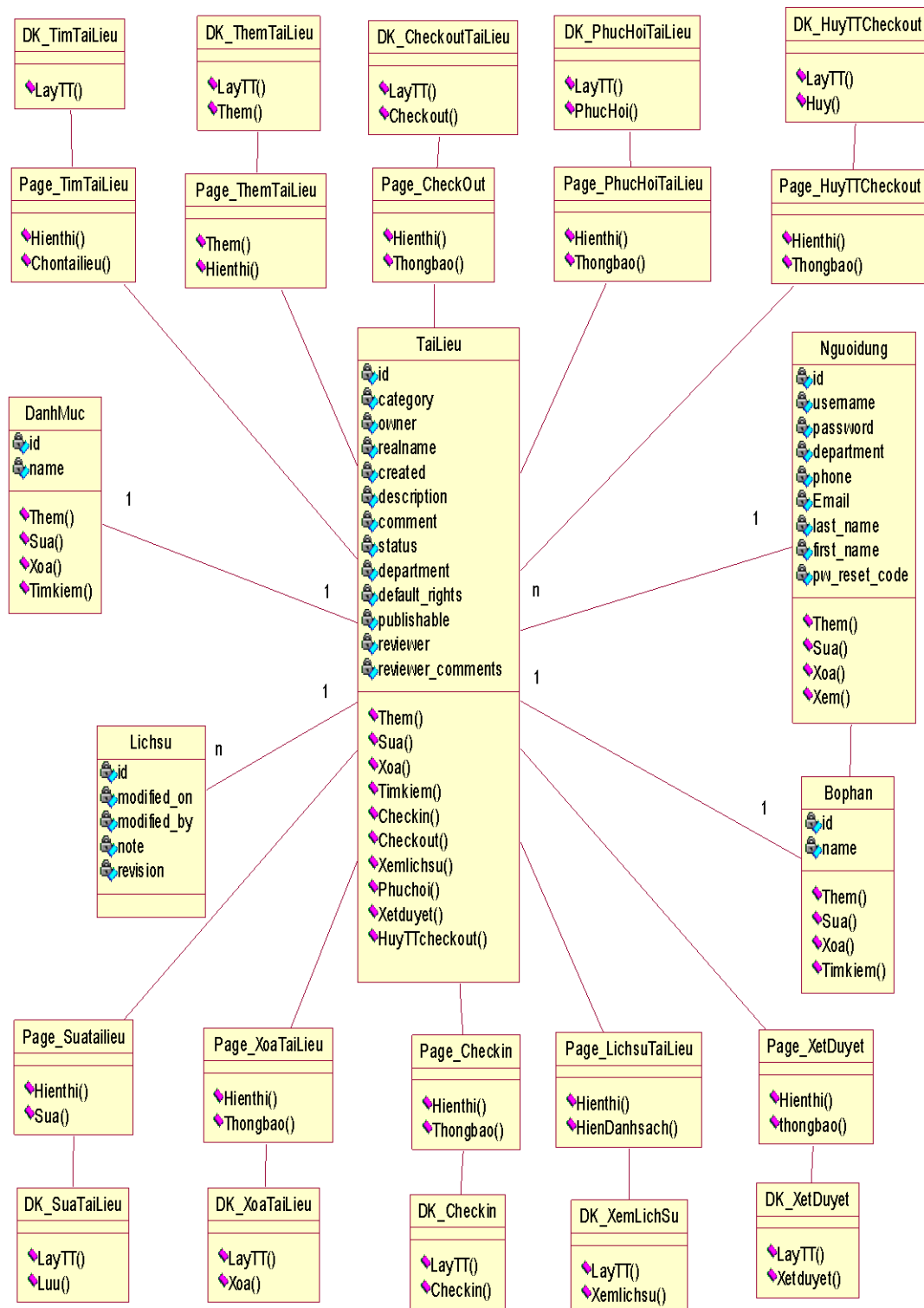
Hình 2.56 : Mô hình lớp thiết kế gói ca sử dụng “Quản lý danh mục”

2.2.4.3. Thiết kế hệ thống “Quản lý người dùng”



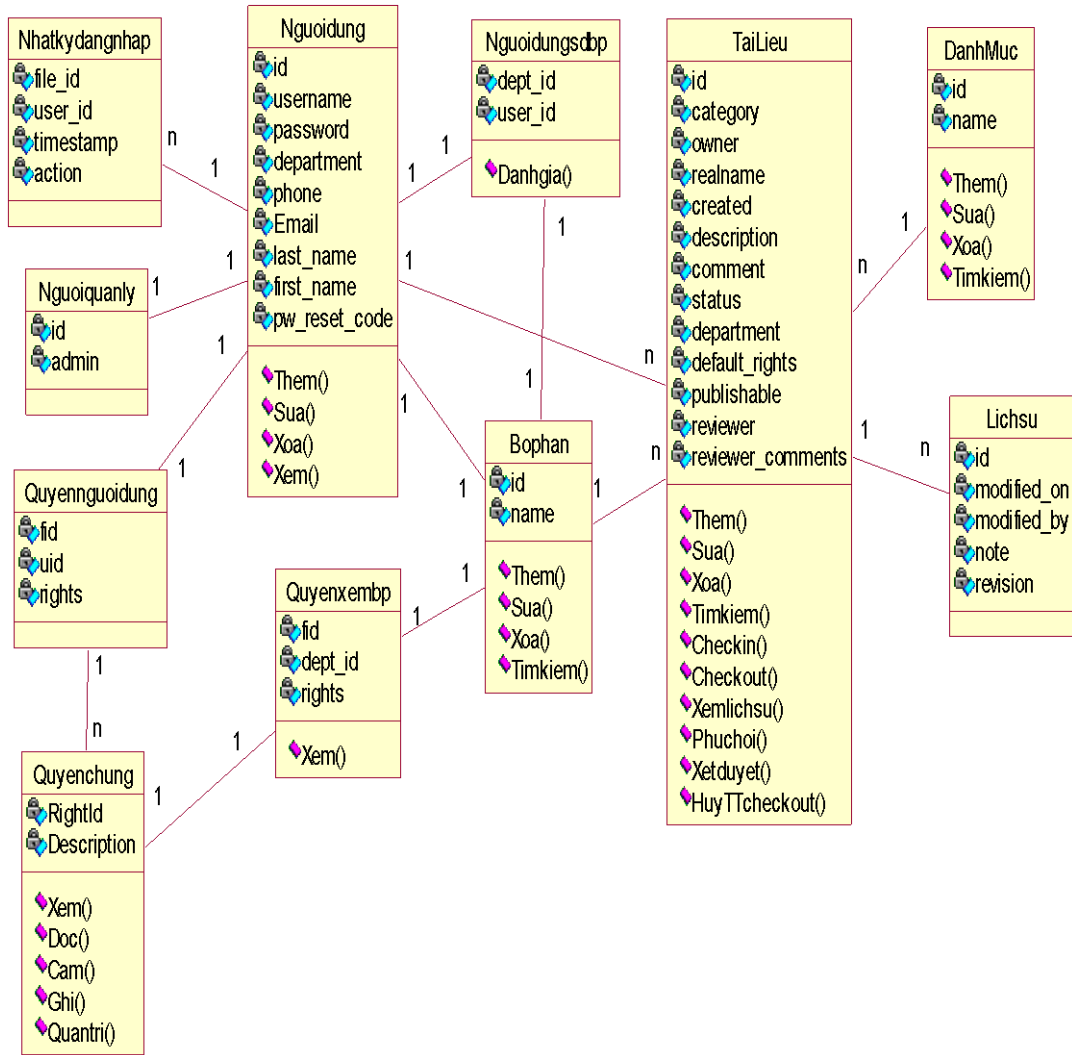
Hình 2.57 : Mô hình lớp thiết kế gói ca sử dụng “Quản lý người dùng”

2.2.4.3. Thiết kế hệ thống “Quản lý tài liệu”



Hình 2.58 : Mô hình lớp thiết kế gói ca sử dụng “Quản lý tài liệu”

2.2.4.4. Thiết kế hệ thống



Hình 2.59 : Mô hình thiết kế cơ sở dữ liệu

CHƯƠNG 3. XÂY DỰNG CHƯƠNG TRÌNH

3.1. Giới thiệu ứng dụng

3.1.1. Thông tin kỹ thuật

- Ứng dụng được viết bằng ngôn ngữ lập trình PHP, CSS3, JavaScript, HTML.
- Hệ quản trị cơ sở dữ liệu: MySQL.
- Máy chủ web: Apache.

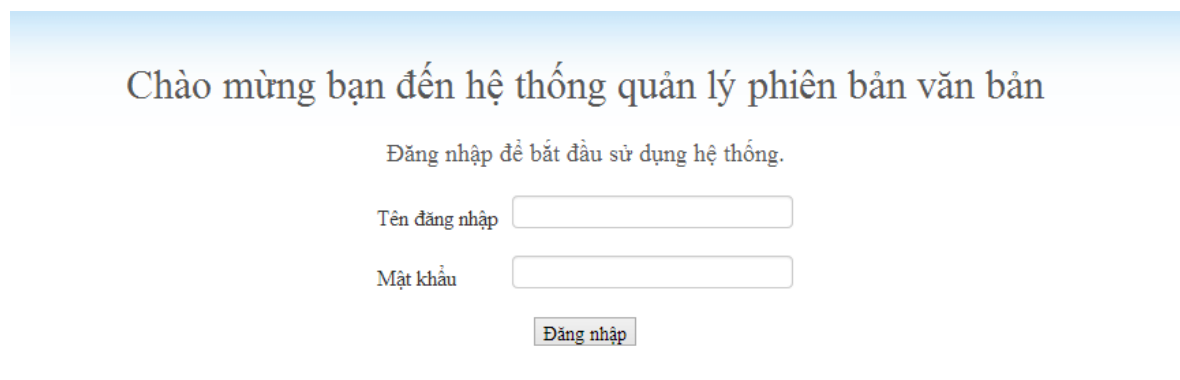
3.1.2. Kết quả đạt được

- Chương trình đã giải quyết được vấn đề cơ bản mà bài toán đưa ra, chương trình chạy ổn định, ít phát sinh lỗi.
- Chương trình đã xây dựng được các chức năng cơ bản mà bài toán đưa ra:
 - Cho phép *check out /check in* tài liệu
 - Cập nhật, thêm mới, xóa, xem các thông tin về người dùng, bộ phận, danh mục, tài liệu...
 - Cho phép người dùng tìm kiếm tài liệu
 - Xem lịch sử của tài liệu

3.2. Giao diện của chương trình

3.2.1. Giao diện đăng nhập

Giao diện đăng nhập để kết nối với CSDL: Người quản lý nhập *username* và *password* để kết nối với cơ sở dữ liệu và chọn đăng nhập để sử dụng chương trình.



Chào mừng bạn đến hệ thống quản lý phiên bản văn bản

Đăng nhập để bắt đầu sử dụng hệ thống.

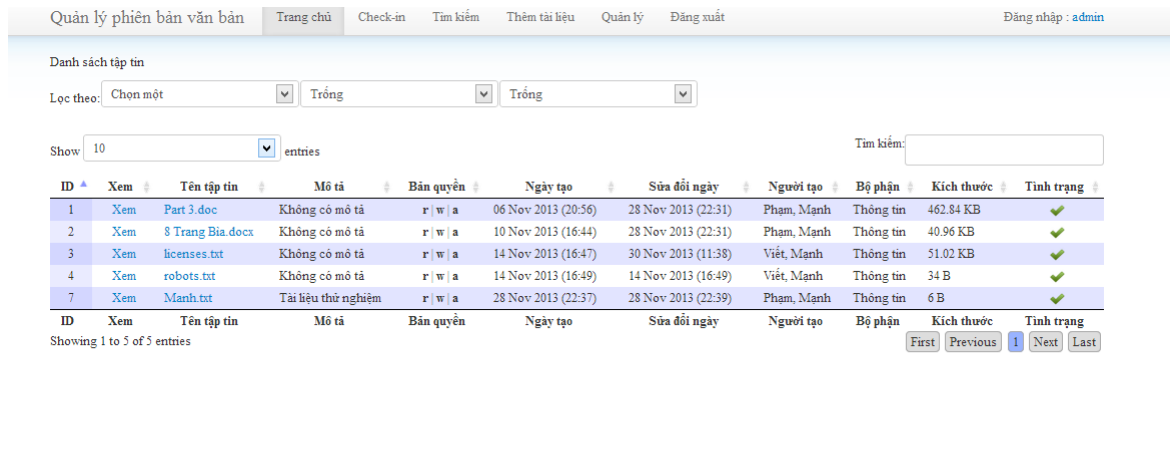
Tên đăng nhập

Mật khẩu

Trường Đại Học Dân Lập Hải Phòng
Khoa Công Nghệ Thông Tin
Phần Mềm Quản Lý Phiên Bản Văn Bản - Phát Triển Bởi Phạm Trung Mạnh

Hình 3.1: Giao diện đăng nhập

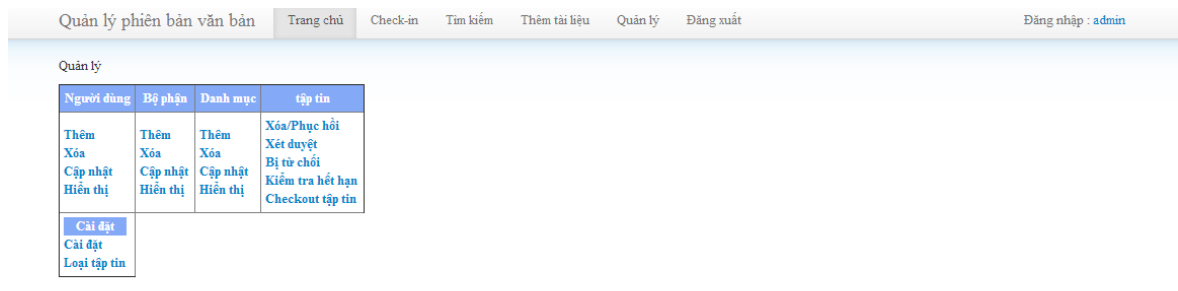
Giao diện khi đăng nhập thành công của chương trình



Trường Đại Học Dân Lập Hải Phòng
 Khoa Công Nghệ Thông Tin
 Phần Mềm Quản Lý Phiên Bản Văn Bản - Phát Triển Bởi Phạm Trung Mạnh

Hình 3.2: Giao diện khi đăng nhập thành công

3.2.2. Giao diện quản lý



Trường Đại Học Dân Lập Hải Phòng
 Khoa Công Nghệ Thông Tin
 Phần Mềm Quản Lý Phiên Bản Văn Bản - Phát Triển Bởi Phạm Trung Mạnh

Hình 3.3: Giao diện quản lý

3.2.3. Giao diện thêm tài liệu

Quản lý phiên bản văn bản | Trang chủ | Check-in | Tìm kiếm | Thêm tài liệu | Quản lý | Đăng xuất | Đăng nhập : admin

Thêm mới tập tin

Vị trí tập tin: Không có tệp nào được chọn

Gán cho chủ sở hữu:

Gán cho bộ phận:

Danh mục:

Bộ phận:

Quyền sử dụng bộ phận (Đặc biệt): Cấm | None | Xem | Đọc | Ghi | Quản trị viên

Mô tả:

Bình luận:

Cho phép người dùng (Đặc biệt):

Cấm | Xem | Đọc | Sửa đổi | Quản lý

* * * * *

Trường Đại Học Dân Lập Hải Phòng
 Khoa Công Nghệ Thông Tin
 Phần Mềm Quản Lý Phiên Bản Văn Bản - Phát Triển Bởi Phạm Trung Mạnh

Hình 3.4: Giao diện thêm tài liệu

- Chức năng **chọn tệp**: Cho phép người sử dụng upload tập tin từ client lên server
- Chức năng **gửi**: Khi nhập đầy đủ tất cả thông tin. Người sử dụng nhấn nút gửi. Chương trình sẽ thêm vào cơ sở dữ liệu với điều kiện dữ liệu nhập vào không trùng với dữ liệu đã có

Giao diện khi thêm tài liệu thành công

Quản lý phiên bản văn bản | Trang chủ | Check-in | Tìm kiếm | Thêm tài liệu | Quản lý | Đăng xuất | Đăng nhập : admin

Chi tiết tập tin

Tài liệu thêm thành công

Đồ án.docx

Danh mục: Hướng dẫn

Kích thước: 284.52 KB

Ngày tạo: 02 Dec 2013 (16:28)

Người tạo: Phạm, Mạnh

Mô tả: Đồ án tốt nghiệp

Bình luận: Không được xóa

Revision: 1

Trường Đại Học Dân Lập Hải Phòng
 Khoa Công Nghệ Thông Tin
 Phần Mềm Quản Lý Phiên Bản Văn Bản - Phát Triển Bởi Phạm Trung Mạnh

Hình 3.5: Giao diện khi thêm tài liệu thành công

3.2.4. Giao diện thêm người dùng

The screenshot shows a web interface for adding a new user. At the top, there is a navigation bar with links: "Quản lý phiên bản văn bản", "Trang chủ", "Check-in", "Tìm kiếm", "Thêm tài liệu", "Quản lý", "Đăng xuất", and "Đăng nhập : admin". Below the navigation bar, the breadcrumb "Quản lý > Thêm mới người dùng" is displayed. The form contains the following fields and controls:

- Họ: Text input field.
- Tên: Text input field.
- Tên đăng nhập: Text input field.
- Số điện thoại: Text input field.
- Vi dụ: 999 9999999 (example number).
- Mật khẩu: Text input field with a strength indicator "7h4kts".
- Địa chỉ E-mail: Text input field.
- Bộ phận: Dropdown menu with "Thông tin" selected.
- Admin?: Radio button.
- Bộ phận đánh giá: Dropdown menu with "Lựa chọn" selected.
- Buttons: "Thêm người dùng" and "Hủy bỏ".

Trường Đại Học Dân Lập Hải Phòng
Khoa Công Nghệ Thông Tin
Phần Mềm Quản Lý Phiên Bản Văn Bản - Phát Triển Bởi Phạm Trung Mạnh

Hình 3.6: Giao diện thêm người dùng

- Chức năng **Thêm người dùng**: Khi người quản lý muốn thêm người dùng vào cơ sở dữ liệu. Người quản lý sẽ nhập vào các trường tương ứng. Click chuột vào chức năng thêm, chương trình sẽ thêm vào cơ sở dữ liệu và hiển thị kết quả nếu dữ liệu nhập vào không trùng với dữ liệu đã có.
- Chức năng **Hủy**: Khi người sử dụng muốn hủy bỏ thao tác

3.2.5. Giao diện thêm danh mục và bộ phận

Hình 3.7: Giao diện thêm bộ phận

Hình 3.8: Giao diện thêm danh mục

- Chức năng **Thêm bộ phận** hoặc **thêm mục**: Khi người quản lý muốn thêm bộ phận hay danh mục vào cơ sở dữ liệu. Người quản lý sẽ nhập vào tên tương ứng. Click chuột vào chức năng thêm bộ phận hoặc thêm mục, chương trình sẽ thêm vào cơ sở dữ liệu và hiển thị kết quả nếu dữ liệu nhập vào không trùng với dữ liệu đã có.
- Chức năng **Hủy**: Khi người sử dụng muốn hủy bỏ thao tác

3.2.6. Giao diện tìm kiếm

Hình 3.9: Giao diện tìm kiếm

- Chức năng **Tìm kiếm**: Khi người dung muốn tìm kiếm tài liệu. Người dùng sẽ nhập vào tên tương ứng. Click chuột vào chức năng tìm kiếm, chương trình sẽ tìm trong hệ thống tài liệu mà người dung muốn tìm.

3.2.7. Checkout tập tin



Hình 3.10: Giao diện checkout tập tin

- Chức năng **Check out tập tin**: Khi người dùng muốn **check out** tập tin từ hệ thống. Người dùng sẽ chọn tập tin cần **check out**. Click chuột vào chức năng **check out** để hoàn thành

3.2.8. Checkin tập tin



Hình 3.11: Giao diện checkin tập tin

- Chức năng **Check in tập tin**: Khi người dùng muốn **check in** tập tin lên hệ thống. Người dùng sẽ chọn tập tin cần **check in**. Click chuột vào chức năng **check in** để hoàn thành

3.2.9. Xem lịch sử tập tin

Quản lý phiên bản văn bản Trang chủ Check-in Tìm kiếm Thêm tài liệu Quản lý Đăng xuất Đăng nhập : admin

Danh sách tập tin > Chi tiết tập tin > Lịch sử

✔ Part 3.doc

Thể loại: Thư
 Kích thước tập tin: 462,84 KB
 Ngày tạo: 06 Nov 2013 (20:56)
 Chủ sở hữu: Phạm, Mạnh
 Mô tả: Không có mô tả
 Bình luận: Không được người tạo bình luận
 Sửa đổi: Mới nhất

 Lịch sử

Phiên bản	Ngày sửa đổi	Bởi	Chú thích
current	28 Nov 2013 (22:31)	Phạm, Mạnh	Đã thay đổi tập tin
4	10 Nov 2013 (20:53)	Phạm, Mạnh	Không có
3	10 Nov 2013 (16:41)	Phạm, Mạnh	Đã sửa đổi
2	06 Nov 2013 (20:58)	Phạm, Mạnh	
1	06 Nov 2013 (20:56)	Phạm, Mạnh	Initial import

Trường Đại Học Dân Lập Hải Phòng
 Khoa Công Nghệ Thông Tin
 Phần Mềm Quản Lý Phiên Bản Văn Bản - Phát Triển Bởi Phạm Trung Mạnh

Hình 3.12: Giao diện xem lịch sử tập tin

KẾT LUẬN

Quản lý phiên bản văn bản là một lĩnh vực đang được các tổ chức và các nhà nghiên cứu quan tâm, phát triển nhằm đem lại các ứng dụng trong thực tế. Trong quá trình thực hiện khóa luận em đã tìm hiểu, nghiên cứu và cài đặt thử nghiệm ứng dụng quản lý phiên bản văn bản, đáp ứng được yêu cầu đề ra. Khóa luận đạt được một số kết quả sau:

- *Tìm hiểu khái quát hệ thống quản lý phiên bản văn bản.*
- *Nắm được kiến trúc của hệ thống quản lý phiên bản văn bản cũng như các ứng dụng của nó.*
- *Phân tích, thiết kế và triển khai ứng dụng quản lý phiên bản văn bản dựa trên nền Web.*
- *Thời gian thực hiện khóa luận đã giúp em mở mang kiến thức về lĩnh vực quản lý văn bản và kiến thức về công nghệ phần mềm, cũng như củng cố các kiến thức đã học trong trường.*

Khóa luận này cho thấy được các công việc cần phải làm để xây dựng ứng dụng trong thực tế. Bên cạnh các kiến thức lý thuyết đã nắm được khóa luận còn thể hiện được kiến thức thực tế và chương trình thử nghiệm có những chức năng sau:

- *Cho phép **check out /check in** tài liệu*
- *Cập nhật, thêm mới, xóa, xem các thông tin về người dùng, bộ phận, danh mục, tài liệu...*
- *Cho phép người dùng tìm kiếm tài liệu.*
- *Xem lịch sử của tài liệu.*
- *Có chức năng báo cáo các tài liệu.*

Chương trình thử nghiệm đạt được một số chức năng chính để phục vụ cho quản lý phiên bản. Nhưng đó cũng chỉ đáp ứng được một phần nhỏ các yêu cầu hiện nay. Em mong nhận được ý kiến đóng góp để chương trình được phát triển hoàn thiện hơn trong thời gian tới và đáp ứng đầy đủ các yêu cầu.

Mặc dù đã cố gắng rất nhiều trong quá trình tìm hiểu và phân tích đề tài nhưng không tránh khỏi những thiếu sót. Vì vậy em mong quý thầy cô cũng như những ai quan tâm đến đề tài này chỉ dẫn và góp ý kiến về khóa luận hoàn thiện website một cách đầy đủ.

Mong muốn của em là phát triển website với nhiều tính năng hơn.

TÀI LIỆU THAM KHẢO

- [1] Nguyễn Văn Vy (2002), Phân tích thiết kế các hệ thống thông tin hiện đại, hướng cấu trúc và hướng đối tượng, NXB Thống kê, Hà Nội.
- [2] Đoàn Văn Ban (2003), Phân tích thiết kế hướng đối tượng bằng UML, NXB Thống kê.
- [3] Đặng Văn Đức (2000), Phân tích hướng đối tượng bằng UML, NXB Giáo dục, Hà Nội.
- [4]. Version Control by Example [Online]
<http://www.ericSink.com/vcbe/html/index.html>
- [5]. Version Control System – VCS [Online]
<http://abdullin.com/wiki/version-control-system-vcs.html>
- [6]. An introduction to version control – Beanstalk Guides [Online]
<http://guides.beanstalkapp.com/version-control/intro-to-version-control.html>
- [7] Version Control with Subversion [Online] <http://svnbook.red-bean.com/>
- [8] Getting Started - About Version Control
<http://git-scm.com/book/en/getting-started-about-version-control>
- [9] Hệ thống quản lý phiên bản – Wikipedia
<http://vi.wikipedia.org/wiki/>