

BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ ĐÔNG Á
KHOA: CÔNG NGHỆ THÔNG TIN



BÀI TẬP LỚN
HỌC PHẦN: ĐỒ ÁN CHUYÊN NGÀNH 1

MÃ ĐỀ: 6.5

TÊN ĐỀ TÀI:

XÂY DỰNG ỨNG DỤNG QUẢN LÝ BÁN SÁCH

Sinh viên thực hiện	Lớp	Khóa
Hà Tiến Dũng	DCCNTT12.10.12	K12

Bắc Ninh, năm 2023

BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ ĐÔNG Á
KHOA: CÔNG NGHỆ THÔNG TIN

BÀI TẬP LỚN
HỌC PHẦN: ĐỒ ÁN CHUYÊN NGÀNH 1

Nhóm: 10

Mã đề: 6.5

Tên đề tài: Xây dựng ứng dụng quản lý bán sách

STT	Sinh viên thực hiện	Mã sinh viên	Điểm bằng số	Điểm bằng chữ
1	Hà Tiến Dũng	20213409		

CÁN BỘ CHẤM 1

(Ký và ghi rõ họ tên)

CÁN BỘ CHẤM 2

(Ký và ghi rõ họ tên)

THÔNG TIN CHUNG

Tên đề tài: Ứng dụng quản lý bán sách.

Sinh viên thực hiện: Hà Tiến Dũng **Lớp:** DCCCNTT12.10.12

Ngày bắt đầu: 21/08/2023 **Ngày kết thúc:** 08/10/2023

KẾ HOẠCH THỰC HIỆN

STT	Nội dung công việc	Ngày hoàn thành dự kiến	Ngày hoàn thành
1	Tìm hiểu các nhu cầu thực tế về quản lý một cửa hàng sách	21/08/2023	21/08/2023
2	Thiết kế các chức năng và giao diện của hệ thống	12/09/2023	12/09/2023
3	Xây dựng cơ sở dữ liệu và phát triển hệ thống	20/09/2023	20/09/2023
4	Kiểm tra hệ thống và đánh giá hệ thống	01/10/2023	02/10/2023
5	Hoàn thành đồ án và nộp báo cáo	05/10/2023	04/10/2023

NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN

Hà Nội, ngày ... tháng ... năm 2023

Giáo viên hướng dẫn

LỜI CẢM ƠN

Em xin gửi lời cảm ơn chân thành đến:

Các thầy cô trường Đại học Công nghệ Đông Á, đặc biệt là các thầy cô Khoa Công Nghệ Thông Tin đã tạo điều kiện thuận lợi từ lúc mới bước chân vào trường cho đến hiện nay và cả tương lai sau này. Chính nhờ các thầy cô khoa công nghệ thông tin đã chỉ dạy em những môn làm nền móng để em có thể thực hiện đề tài đồ án chuyên ngành 1 của mình.

Gửi lời cảm ơn đến giáo viên hướng dẫn trực tiếp môn đồ án chuyên ngành của em. Thầy đã có những ý kiến chỉ dẫn, định hướng giúp ích để em thực hiện hoàn chỉnh đề tài của mình.

Trong quá trình thực hiện đồ án em đã cố gắng tìm hiểu trao đổi các kiến thức mới để hoàn thành tốt đồ án của mình. Tuy nhiên do hạn chế về mặt thời gian, kiến thức và kinh nghiệm thực tế rất khó tránh khỏi những sai sót. Em kính mong đóng góp ý kiến để em hoàn thành tốt hơn các đồ án tiếp theo và cũng như để hoàn thiện bản thân hơn.

Hà Nội, ngày ... tháng ... năm 2023

Sinh viên thực hiện

LỜI CAM ĐOAN

Tôi xin cam đoan số liệu và kết quả nghiên cứu trong đồ án này là trung thực và chưa hề được sử dụng để bảo vệ một học vị nào. Mọi sự giúp đỡ này đã được cảm ơn và các thông tin trích dẫn trong đồ án này đã được chỉ rõ nguồn gốc rõ ràng và được phép công bố.

Hà Nội, ngày ... tháng ... năm 2023

Sinh viên thực hiện

MỤC LỤC

DANH MỤC CÁC TỪ VIẾT TẮT	1
DANH MỤC BẢNG	2
DANH MỤC HÌNH	3
CHƯƠNG I: MỞ ĐẦU	7
Lý do chọn đề tài.....	7
Giới hạn phạm vi nghiên cứu.....	7
Mục đích nghiên cứu.....	7
Khách thể và đối tượng nghiên cứu	7
Nhiệm vụ nghiên cứu	7
Phương pháp nghiên cứu.....	8
Ý nghĩa lý luận và thực tiễn của đề tài.....	8
Các công nghệ sử dụng	8
CHƯƠNG 2: QUY TRÌNH NGHIÊN CỨU THỰC TRẠNG	12
2.1 Quá trình khảo sát thực trạng	12
2.1.1 Các câu hỏi và câu trả lời thu thập được.....	12
2.2 Kết quả khảo sát thực trạng.....	13
CHƯƠNG 3: PHÂN TÍCH QUY TRÌNH HOẠT ĐỘNG CỦA HỆ THỐNG	14
3.1 Quy trình nhập sách	14
3.2 Quy trình bán sách	14
3.3 Quy trình quản lí sách	14
3.4 Quy trình quản lí các thông tin khác như tác giả, thể loại, nhà xuất bản	15
3.5 Quy trình quản lí nhân viên.....	15
CHƯƠNG 4: PHÂN TÍCH THIẾT KẾ HỆ THỐNG	16
4.1 Các chức năng của hệ thống.....	16
4.1.1 Quản lí hệ thống.....	16
4.1.2 Quản lí nhập sách.....	16
4.1.3 Quản lí bán sách.....	16
4.1.4 Quản lí sách	16
4.1.5 Quản lí nhân viên.....	16
4.1.6 Quản lí khác	16
4.1.7 Báo cáo thông kê	17
4.2 Biểu đồ Usecase	17

4.2.1 Danh sách các actor của hệ thống.....	17
4.2.2 Danh sách Usecase của hệ thống.....	18
4.2.3. Vẽ biểu đồ Usecase.....	19
4.2.4 Đặc tả Usecase	23
4.3 Mô hình hóa khái niệm	53
4.3.1 Sơ đồ hoạt động	53
4.3.2 Sơ đồ trạng thái.....	69
4.3.3 Sơ đồ lớp.....	71
4.3.4 Đặc tả các lớp	72
4.4 Mô hình cơ sở dữ liệu	76
4.4.1 Lược đồ cơ sở dữ liệu quản hệ.....	76
4.4.2 Mô hình cơ sở dữ liệu	77
4.4.3 Các bảng cơ sở dữ liệu vật lý.....	77
CHƯƠNG 5: PHÁT TRIỂN HỆ THỐNG	80
5.1 Tầng GUI	80
5.2 Tầng DTO	93
5.3 Tầng DAL	95
5.4 Tầng BLL.....	129
5.5 Kết chương	137
CHƯƠNG 6: KẾT QUẢ THU ĐƯỢC.....	138
Hạn chế và hướng phát triển của đề tài.....	156
Tài liệu tham khảo	157

DANH MỤC CÁC TỪ VIẾT TẮT

Số thứ tự	Chữ viết tắt	Giải thích
1	GUI	Graphical User Interface (Giao diện người dùng đồ họa)
2	DAL	Data Access Layer (Lớp truy cập dữ liệu)
3	BLL	Business Logic Layer (Lớp logic nghiệp vụ)
4	DTO	Data Transfer Object (Đối tượng chuyển dữ liệu)
5	CTHD	Chi tiết hóa đơn
6	CTPN	Chi tiết phiếu nhập
7	Tbl	Table (bảng)

DANH MỤC BẢNG

Số hiệu	Tên	Trang
Tbl4.1	Bảng đặc tả lớp tài khoản	72
Tbl4.2	Bảng đặc tả lớp NhanVien	72
Tbl4.3	Bảng đặc tả lớp PhieuNhap	73
Tbl4.4	Bảng đặc tả lớp CTPN	73
Tbl4.5	Bảng đặc tả lớp NhaXuatBan	74
Tbl4.6	Bảng đặc tả lớp TacGia	74
Tbl4.7	Bảng đặc tả lớp TheLoai	75
Tbl4.8	Bảng đặc tả lớp HoaDon	75
Tbl4.9	Bảng đặc tả lớp CTHD	75
Tbl4.10	Bảng đặc tả lớp Sach	76
Tbl4.11	Bảng TblTaiKhoan	77
Tbl4.12	Bảng TblNhanVien	77
Tbl4.13	Bảng TblCTHoaDon	78
Tbl4.14	Bảng TblCTPhieuNhap	78
Tbl4.15	Bảng TblHoaDon	78
Tbl4.16	Bảng TblNhaXuatBan	78
Tbl4.17	Bảng TblPhieuNhap	78
Tbl4.18	Bảng TblPhieuNhap	79
Tbl4.19	Bảng TblTacGia	79
Tbl4.20	Bảng TblTheLoai	79

DANH MỤC HÌNH

Số hiệu	Tên	Trang
4.0	Biểu đồ phân rã chức năng	17
4.1	Biểu đồ usecase tổng quát	19
4.2	Biểu đồ usecase quản lý hệ thống	20
4.3	Biểu đồ usecase quản lý nhập sách	20
4.4	Biểu đồ usecase quản lý bán sách	21
4.5	Biểu đồ usecase quản lý sách	21
4.6	Biểu đồ usecase quản lý nhân viên	22
4.7	Biểu đồ usecase quản lý khác	22
4.8	Biểu đồ usecase báo cáo thông kê	23
4.9	Sơ đồ hoạt động tác nhân đăng nhập	53
4.10	Sơ đồ hoạt động tác nhân thay đổi mật khẩu	53
4.11	Sơ đồ hoạt động tác nhân thay đăng xuất	54
4.12	Sơ đồ hoạt động tác nhân thêm phiếu nhập	54
4.13	Sơ đồ hoạt động tác nhân sửa thông tin phiếu nhập	55
4.14	Sơ đồ hoạt động tác nhân xóa thông tin phiếu nhập	55
4.15	Sơ đồ hoạt động tác nhân tìm phiếu nhập	56
4.16	Sơ đồ hoạt động tác nhân thêm hóa đơn	56
4.17	Sơ đồ hoạt động tác nhân sửa hóa đơn	57
4.18	Sơ đồ hoạt động tác nhân xóa hóa đơn	57
4.19	Sơ đồ hoạt động tác nhân in hóa đơn	58
4.20	Sơ đồ hoạt động tác nhân tìm kiếm hóa đơn	58
4.21	Sơ đồ hoạt động tác nhân tìm kiếm sách	59
4.22	Sơ đồ hoạt động tác nhân thêm nhân viên	59
4.23	Sơ đồ hoạt động tác nhân sửa nhân viên	60
4.24	Sơ đồ hoạt động tác nhân xóa nhân viên	60
4.25	Sơ đồ hoạt động tác nhân tìm kiếm nhân viên	61
4.26	Sơ đồ hoạt động tác nhân thêm tác giả	61
4.27	Sơ đồ hoạt động tác nhân sửa tác giả	62
4.28	Sơ đồ hoạt động tác nhân xóa tác giả	62
4.29	Sơ đồ hoạt động tác nhân tìm kiếm tác giả	63

4.30	Sơ đồ hoạt động tác nhân thêm thẻ loại	63
4.31	Sơ đồ hoạt động tác nhân sửa thẻ loại	64
4.32	Sơ đồ hoạt động tác nhân xóa thẻ loại	64
4.33	Sơ đồ hoạt động tác nhân tìm kiếm thẻ loại	65
4.34	Sơ đồ hoạt động tác nhân thêm nhà xuất bản	65
4.35	Sơ đồ hoạt động tác nhân sửa nhà xuất bản	66
4.36	Sơ đồ hoạt động tác nhân xóa nhà xuất bản	66
4.37	Sơ đồ hoạt động tác nhân tìm kiếm nhà xuất bản	67
4.38	Sơ đồ hoạt động tác nhân thống kê nhập sách	67
4.39	Sơ đồ hoạt động tác nhân thống kê bán sách	68
4.40	Sơ đồ hoạt động tác nhân thống kê số lượng tồn	68
4.41	Sơ đồ trạng thái chức năng nhập sách	69
4.42	Sơ đồ trạng thái chức năng bán sách sách	69
4.43	Sơ đồ trạng thái chức năng quản lý kho sách	79
4.44	Sơ đồ trạng thái chức năng quản lý nhân viên	70
4.45	Sơ đồ trạng thái chức năng quản lý tác giả	70
4.46	Sơ đồ trạng thái chức năng quản lý thẻ loại	70
4.47	Sơ đồ trạng thái chức năng quản lý nhà xuất bản	70
4.48	Sơ đồ trạng thái chức năng thống kê báo cáo	71
4.49	Sơ đồ lớp tổng quát	71
4.50	Lược đồ cơ sở dữ liệu quản hệ	76
4.51	Mô hình cơ sở dữ liệu	77
5.1	Các thành phần trong tầng GUI.	80
5.2	giao diện đăng nhập.	81
5.3	giao diện chính mục “Hệ thống”.	81
5.4	giao diện chính mục “Quản lí chính”.	82
5.5	giao diện chính mục “Quản lí danh mục”.	82
5.6	giao diện chính mục “Thống kê báo cáo”.	83
5.7	giao diện quản lí người dùng	83
5.8	giao diện thay mật khẩu	84
5.9	giao diện quản lí nhập sách	84
5.10	giao diện quản lí chi tiết phiếu nhập	84

5.11	giao diện quản lý bán sách	85
5.12	giao diện quản lý chi tiết hóa đơn	85
5.13	giao diện quản lý kho sách	86
5.14	giao diện quản lý nhân viên	86
5.15	giao diện quản lý nhà xuất bản	87
5.16	giao diện quản lý thể loại	87
5.17	giao diện quản lý tác giả	88
5.18	giao diện thống kê nhập	88
5.19	giao diện thống kê bán	88
5.20	giao diện thống kê số lượng tồn	89
5.21	giao diện hiển thị báo cáo	89
5.22	mẫu báo cáo nhập	90
5.23	mẫu báo cáo chi tiết nhập	90
5.24	mẫu báo cáo bán sách	91
5.25	mẫu báo cáo chi tiết bán sách	91
5.26	mẫu báo cáo số lượng tồn	92
5.27	mẫu in hóa đơn	92
5.28	Các thành phần trong tầng DTO	93
5.29	Các thành phần trong tầng DAL	95
5.30	Các thành phần trong tầng BLL	129
6.1	giao diện đăng nhập	138
6.2	giao diện chính	138
6.3	thực hiện thêm người dùng	139
6.4	kết quả sau khi thêm người dùng	139
6.5	sửa thông tin người dùng	140
6.6	kết quả sau khi sửa thông tin người dùng	140
6.7	Xóa người dùng	141
6.8	kết quả xóa người dùng	141
6.9	thay đổi mật khẩu	142
6.10	kết quả sau khi thay đổi mật khẩu	142
6.11	thêm phiếu nhập	143
6.11.1	kết quả khi thêm phiếu nhập	143

6.12	chọn chi tiết phiếu nhập của hóa đơn	144
6.13	kết quả khi vào chi tiết phiếu nhập của một phiếu nhập	144
6.14	Thêm sách nhập vào chi tiết phiếu nhập	145
6.15	Kết quả sau khi thêm sách nhập vào chi tiết phiếu nhập	145
6.16	Phiếu nhập được cập nhật tổng tiền khi thêm sách nhập vào chi tiết phiếu nhập đó	146
6.17	Thêm xong một hóa đơn	146
6.18	Thêm sách khách mua vào hóa đơn	147
6.19	Chọn hóa đơn và chọn in hóa đơn	147
6.20	hóa đơn được in	148
6.21	giao diện kho sách	148
6.22	chọn thông tin tìm kiếm sách	149
6.23	Kết quả khi ấn tìm kiếm sách	149
6.24	Giao diện quản lý nhân viên	150
6.25	Giao diện quản lý nhà xuất bản	150
6.26	Giao diện quản lý thể loại	150
6.27	Giao diện quản lý tác giả	151
6.28	Giao diện thống kê nhập	151
6.29	chọn các mục cần thiết và ấn in	151
6.30	chọn các mục cần thiết và ấn in	152
6.31	Kết quả sau khi ấn in	152
6.32	Kết quả sau khi ấn in	153
6.33	Giao diện thống kê bán sách	153
6.34	Kết quả in thống kê bán sách năm 2023	154
6.35	Kết quả in thống kê chi tiết bán sách năm 2023	154
6.36	Giao diện thống kê số lượng tồn	155
6.37	Kết quả in thống kê số lượng tồn	155

CHƯƠNG I: MỞ ĐẦU

Lý do chọn đề tài

Hiện nay, công nghệ thông tin đã không còn xa lạ với nhiều người và đang có vai trò quan trọng trong đời sống, kinh tế, xã hội cho nhiều quốc gia trên thế giới, là một phần không thể thiếu trong xã hội hiện nay. Công nghệ thông tin giúp con người thực hiện mọi thứ dễ dàng và thuận tiện hơn. Ngày nay, chúng ta chỉ cần soạn tin nhắn và ấn một nút ấn là có thể gửi tin nhắn cho một người ở một đất nước khác một cách nhanh chóng hoặc chỉ cần lướt các nền tảng bán hàng là có thể mua được các sản phẩm mình mong muốn mà không cần phải đến trực tiếp các cửa hàng. Công nghệ thông tin còn góp ích rất lớn cho cho các ngành giáo dục, y tế, kinh tế, giải trí,... Để có thể kể hết lợi ích của công nghệ thông tin đem lại thì là rất nhiều.

Nhờ có công nghệ thông tin mà việc quản lý một thức gì đó từ đơn giản cho đến phức tạp đều có thể quản lý một cách dễ dàng như quản lý các bệnh nhân của một bệnh viện, quản lý các học sinh, các giáo viên, hay quản lý lương cho các nhân viên của một công ty nào đó. Miễn là chúng ta cần quản lý một thứ gì đó công nghệ thông tin đều có thể giúp ích được cho chúng ta. Xuất phát từ lợi ích to lớn của công nghệ thông tin đem lại em muốn thực hiện tìm hiểu rõ xem trong công việc mua và bán sách, việc quản lý sách nhập, sách bán, quản lý lợi nhuận từ việc bán sách,... được thực hiện như thế nào. Vì vậy, em quyết định lựa chọn thực hiện đề tài “Xây dựng ứng dụng quản lý bán sách” là đề tài báo cáo cho môn đồ án chuyên ngành của nhóm em.

Do thời gian cho phép ngắn, trình độ chuyên môn, kinh nghiệm và kiến thức của bản thân còn hạn chế, nên em rất mong được sự góp ý của các thầy cô, để đề tài của chúng em ngày càng hoàn thiện hơn và được ứng dụng trong thực tế.

Xin chân thành cảm ơn!

Giới hạn phạm vi nghiên cứu

Quản lý một cửa hàng sách nhỏ.

Mục đích nghiên cứu

Xây dựng được phần mềm quản lý bán sách để quản lý cho một cửa hàng sách

Khách thê và đối tượng nghiên cứu

Khách thê: Hệ thống quản lý bán sách.

Đối tượng: Phần mềm quản lý bán sách cho một cửa hàng.

Nhiệm vụ nghiên cứu

Tìm hiểu nghiệp vụ quản lý bán sách (Quản lý sách, tác giả, nhập sách, doanh thu,...).

Tìm hiểu các công cụ xây dựng chương trình (Microsoft SQL server, Ngôn ngữ C#,...).

Phương pháp nghiên cứu

- Tìm kiếm trên mạng.
- Khảo sát các thông tin về cửa hàng sách xung quanh.

Ý nghĩa lý luận và thực tiễn của đề tài

- Giúp em hiểu hơn về nghiệp vụ của quản lý bán sách
- Ứng dụng nhằm vào việc tối ưu hóa quá trình quản lý sách như nhập hoặc bán và quản lý thông tin một cách chính xác.

Các công nghệ sử dụng

Ứng dụng lập trình: Microsoft Visual Studio 2019

- **Microsoft Visual Studio** là một môi trường phát triển tích hợp (IDE) từ Microsoft. Microsoft Visual Studio còn được gọi là "Trình soạn thảo mã nhiều người sử dụng nhất thế giới", được dùng để lập trình C++ và C# là chính. Nó được sử dụng để phát triển chương trình máy tính cho Microsoft Windows, cũng như các trang web, các ứng dụng web và các dịch vụ web. Visual Studio sử dụng nền tảng phát triển phần mềm của Microsoft như Windows API, Windows Forms, Windows Presentation Foundation, Windows Store và Microsoft Silverlight. Nó có thể sản xuất cả hai ngôn ngữ máy và mã số quản lý.
- Visual Studio bao gồm một trình soạn thảo mã hỗ trợ IntelliSense cũng như cài đặt mã nguồn. Trình gõ lỗi tích hợp hoạt động cả về trình gõ lỗi mức độ mã nguồn và gõ lỗi mức độ máy. Công cụ tích hợp khác bao gồm một mẫu thiết kế các hình thức xây dựng giao diện ứng dụng, thiết kế web, thiết kế lớp và thiết kế giản đồ cơ sở dữ liệu. Nó chấp nhận các plug-in nâng cao các chức năng ở hầu hết các cấp bao gồm thêm hỗ trợ cho các hệ thống quản lý phiên bản (như Subversion) và bổ sung thêm bộ công cụ mới như biên tập và thiết kế trực quan cho các miền ngôn ngữ cụ thể hoặc bộ công cụ dành cho các khía cạnh khác trong quy trình phát triển phần mềm.
- Visual Studio hỗ trợ nhiều ngôn ngữ lập trình khác nhau và cho phép trình biên tập mã và gõ lỗi để hỗ trợ (mức độ khác nhau) hầu như mọi ngôn ngữ lập trình. Các ngôn ngữ tích hợp gồm có C, C++ và C++/CLI (qua Visual C++), VB.NET (qua Visual Basic.NET), C# (qua Visual C#) và F# (như của Visual Studio 2010). Hỗ trợ cho các ngôn ngữ khác như J++/J#, Python và Ruby thông qua dịch vụ cài đặt riêng rẽ. Nó cũng hỗ trợ XML/XSLT, HTML/XHTML, JavaScript và CSS.

Ngôn ngữ lập trình: C#

- C# là **ngôn ngữ lập trình hiện đại, hướng đối tượng và nó được xây dựng trên nền tảng của hai ngôn ngữ mạnh nhất là C++ và Java**.

C# được thiết kế cho Common Language Infrastructure (CLI), mà gồm Executable Code và Runtime Environment, cho phép chúng ta sử dụng các ngôn ngữ high-level đa dạng trên các nền tảng và cấu trúc máy tính khác nhau.

- C# là một ngôn ngữ đơn giản: C# loại bỏ một vài sự phức tạp và rối rắm của những ngôn ngữ như Java và c++, bao gồm việc loại bỏ những macro, những template, đa kế thừa, và lớp cơ sở ảo (virtual base class). Ngôn ngữ C# đơn giản vì nó dựa trên nền tảng C và C++. Nếu chúng ta thân thiện với C và C++ hoặc thậm chí là Java, chúng ta sẽ thấy C# khá giống về diện mạo, cú pháp, biểu thức, toán tử và những chức năng khác được lấy trực tiếp từ ngôn ngữ C và C++, nhưng nó đã được cải tiến để làm cho ngôn ngữ đơn giản hơn.
- C# là một ngôn ngữ ít từ khóa: C# là ngôn ngữ sử dụng giới hạn những từ khóa. Phần lớn các từ khóa được sử dụng để mô tả thông tin. Chúng ta có thể nghĩ rằng một ngôn ngữ có nhiều từ khóa thì sẽ mạnh hơn. Điều này không phải sự thật, ít nhất là trong trường hợp ngôn ngữ C#, chúng ta có thể tìm thấy rằng ngôn ngữ này có thể được sử dụng để làm bất cứ nhiệm vụ nào.

Thư viện lớp đồ họa Winform

- **Winform** là một thư viện lớp đồ họa, mã nguồn mở và được cung cấp hoàn toàn miễn phí. Phần mềm này cung cấp nền tảng giúp bạn viết những lập trình đa dạng cho các thiết bị như máy tính bàn, laptop, máy tính bảng,... Winform cũng được coi như là một sự thay thế đối với thư viện lớp nền tảng Microsoft Foundation của C++.
- Mỗi màn hình Windows lại cung cấp một giao diện để người dùng có thể giao tiếp với ứng dụng được gọi là GUI (giao diện đồ họa của ứng dụng). Nó bao gồm các ứng dụng chạy trên máy tính Windows như Microsoft, Word, Excel, Mail, Access, Yahoo, Calculator,...
- Winform có các thành phần cơ bản như Forms Panel, Button Textbox, ComboBox, RadioButton,... Trong đó, Form là nơi chứa tất cả thành phần của chương trình, Panel chứa Button, Label, TextBox. Button là nút nhấn, TextBox dùng để nhập văn bản một dòng hay nhiều dòng. Label hiển thị văn bản hoặc thông tin trên Form và ComboBox là các lựa chọn có sẵn để bạn lựa chọn dễ dàng hơn.

- Ưu điểm:

- Winform dễ học, dễ dạy và thao tác linh hoạt với giao diện kéo thả, dễ sử dụng. Bằng cách double click, bạn có thể gắn các event cho các button một cách nhanh chóng. Hỗ trợ nhiều event như Hover, Click,... Viết code trực quan từ việc lấy text từ TextBox đến khi show dữ liệu bằng Messenger Box. Hoặc bạn cũng có thể dùng Grid để kết nối SQL.
- Phù hợp để làm các phần mềm quản lý, thống kê hay kế toán,... Đây là các loại ứng dụng mà công ty hoặc doanh nghiệp nhỏ và vừa rất cần có. Bên cạnh đó, bạn chỉ cần sử dụng các component như Telerik UI hoặc DevExpress để tạo ra một giao diện hiện đại và chất lượng. Hiện nay, các trình viên ở Việt Nam hầu như đều sử dụng crack nên các Component này là hoàn toàn miễn phí giúp bạn tạo ra nhiều giao diện hiện đại, đẹp mắt. Từ đó cải thiện tốt nhất trải nghiệm người dùng trên ứng dụng.
- Tốc độ xử lý dữ liệu nhanh chóng, đáp ứng yêu cầu người dùng.
- Đảm bảo thông tin được bảo mật an toàn.
- Winform có thể chạy trên nhiều phiên bản Windows khác nhau.
- Có thể thao tác trên nhiều giao diện khác nhau.

Bộ thư viện DevExpress

- Đối với những lập trình viên .NET thì DevEpress là một công cụ hết sức hữu dụng, cung cấp rất nhiều control trong Visual Studio. DevExpress không chỉ giúp thiết kế winform hay website đẹp hơn mà còn giúp cho việc lập trình được dễ dàng hơn, ta có thể thấy rõ nhất là trong việc tương tác với cơ sở dữ liệu.
- DevExpress được ra mắt lần đầu tiên vào năm 2011 và được đồng đảo lập trình viên .NET sử dụng. Từ đó đến nay đã trải qua rất nhiều phiên bản với nhiều nâng cấp đáng kể. Phiên bản mới nhất hiện nay là v18.2. Với DevExpress ta có thể tự tạo cho mình một bộ Office riêng chỉ trong vòng một vài tiếng.
- Ưu điểm :
 - Hạn chế xuất hiện nhiều form riêng lẻ
 - Có thể tự co giãn form bên trong form chính theo kích thước của form chính thay đổi
 - Cung cấp rất nhiều UI đẹp cho Winform, Web
 - Hỗ trợ rất nhiều Control hữu dụng
 - Giúp việc lập trình trở nên nhanh, dễ dàng hơn
 - Dễ quản lý
 - Có nhiều tài liệu hỗ trợ

Hệ quản trị CSDL: Microsoft SQL Sever

- **Microsoft SQL Server** là một hệ quản trị cơ sở dữ liệu quan hệ được phát triển bởi Microsoft. Là một máy chủ cơ sở dữ liệu, nó là một sản phẩm phần mềm có chức năng chính là lưu trữ và truy xuất dữ liệu theo yêu cầu của các ứng dụng phần mềm khác. Có thể chạy trên cùng một máy tính hoặc trên một máy tính khác trên mạng (bao gồm cả Internet).
- Microsoft tiếp thị ít nhất một chục phiên bản Microsoft SQL Server khác nhau, nhằm vào các đối tượng khác nhau và cho khối lượng công việc khác nhau, từ các ứng dụng máy đơn nhỏ đến các ứng dụng Internet lớn có nhiều người dùng đồng thời.

CHƯƠNG 2: QUY TRÌNH NGHIÊN CỨU THỰC TRẠNG

2.1 Quá trình khảo sát thực trạng

- Tìm kiếm và tham khảo các mẫu phiếu phỏng vấn, các cách thức hoạt động của một cửa hàng bán sách có trên mạng.

2.1.1 Các câu hỏi và câu trả lời thu thập được

- Quản lý việc nhập và bán sách như thế nào? : Thường có một quyền số riêng để kiểm tra số lượng sách nhập vào, sách bán ra.

- Để kiểm tra doanh số bán được mỗi ngày, mỗi tuần, mỗi tháng thì có mất nhiều thời gian không? : Cũng không gặp khó khăn nhưng mất nhiều thời gian. Phải dùng máy tính cá nhân để tổng kết.

- Anh có biết sử dụng phần mềm như : Word, Excel, Access không ? : Biết sử dụng nhưng không rành, chỉ biết sơ sơ.

- Nghĩ sao về việc nếu có một phần mềm giúp rút ngắn thời gian ghi chép cũng như tính toán? : Việc rút ngắn thời gian ghi chép và tính toán là rất cần thiết.

- Làm thế nào để biết lượng hàng hóa luân chuyển, hàng tồn kho? : Hàng ngày, mỗi khi nhập hoặc xuất hàng đều ghi chép lại vào sổ sách để kiểm tra số lượng sách nhập, sách bán được và còn tồn kho bao nhiêu.

- Có thương xuyênn thông kê số lượng sách nhập, xuất hay tồn kho bao nhiêu không?: Có, nhưng không thường xuyên khoảng 2 tuần 1 lần.

- Khi khách muốn mua một quyền sách bất kỳ, thì việc tìm quyền sách đó có dễ dàng hay không? : Được, nhưng mất khoảng vài phút.

- Có gặp phải các khó khăn gì khi phải thực hiện công việc bằng tay hay không? : Có, việc cập nhật và lưu trữ các thông tin sách mất nhiều thời gian và công sức. các việc lập hóa đơn, lập báo cáo, đều thực hiện bằng tay nên khó tránh khỏi sai sót.

- Việc kiểm soát nhập, xuất sách có đảm bảo không? Vì sao? : Có, vì chương trình chạy trên một cơ sở dữ liệu luôn được kiểm soát bởi người có trách nhiệm.

- Thu chi trong việc bán sách được xử lý như thế nào? : Việc thu chi được kiểm soát theo các phiếu hóa đơn sau khi khách hàng mua và được lưu lại.

- Báo cáo doanh số bán theo ngày, tháng, quý, năm có được kiểm soát chặt chẽ không? : Có, vì doanh số được bán sách được thống kê từng ngày, và tổng hợp hàng tuần, hàng tháng, hàng năm.

- Kế hoạch nhập xuất sách ? : Nhập hàng: phải lăm được số lượng hàng tồn kho, lên kế hoạch nhập hàng trong tháng, năm , tham khảo giá thị trường bằng cách đề nghị công ty báo giá các mặt hàng, chọn hàng chất lượng, giá rẻ và đem lại lợi nhuận. Xuất hàng: Khi bán phải theo dõi rõ ràng và ký nhận của người mua, thương xuyênn kiểm kê chống mối và phải nắm chắc số lượng hàng tồn kho trong kho.

2.2 Kết quả khảo sát thực trạng

- Nhân viên được sự chỉ đạo của người quản lý liên hệ với nhà xuất bản sách để lập hóa đơn mua sách. Sau khi mua sách, nhân viên nhận sách và căn cứ theo chứng từ mua sách để tiến hàng nhập sách vào kho.
- Do đó, ta cần có kho dữ liệu sách gồm các thông tin : mã sách, tên sách, thể loại, tác giả, nhà xuất bản, số lượng tồn.
- Mỗi lần nhập sách phải có phiếu nhập để lưu trữ thông tin nhập như: số phiếu nhập, mã nhà xuất bản, ngày nhập.
- Do nhập sách từ nhiều nhà xuất bản khác nhau nên cần có một kho dữ liệu để chứa các thông tin về các nhà xuất bản bao gồm: mã nhà xuất bản, tên nhà xuất bản, địa chỉ nhà xuất bản, địa chỉ, điện thoại.
- Tiệm sách cũng có nhiều thể loại sách nên cũng cần một kho dữ liệu thể loại sách gồm các thông tin: mã thể loại và tên thể loại.
- Trong trường hợp khách đến mua sách yêu cầu tìm kiếm sách của một tác giả thì cũng cần một kho dữ liệu về thông tin tác giả để phục vụ nhu cầu của khách hàng bao gồm : mã tác giả, tên tác giả, số điện thoại.
- Khi khách hàng mua hàng thì cần lập hóa đơn trong đó gồm có số hóa đơn và ngày bán
- Cuối tháng, năm, nhưng người bán hàng phải lập báo cáo tình hình nhập, xuất, tồn của từng loại sách. Hàng tháng, năm phải lập báo cáo doanh thu nhằm đánh giá xu hướng, tình hình tiêu thụ sách.

CHƯƠNG 3: PHÂN TÍCH QUY TRÌNH HOẠT ĐỘNG CỦA HỆ THỐNG

3.1 Quy trình nhập sách

- Thời gian: Thực hiện mỗi khi cửa hàng nhập tài liệu mới về. Tài liệu nhập bao gồm: Sách, truyện, giáo trình, đĩa cung,... Trong đó sách là tài liệu chính.
- Tác nhân tham gia vào quá trình nhập tài liệu: Quản lý thư viện.
- Vai trò: Tăng số lượng sách đáp ứng nhu cầu khách hàng, tạo nguồn sách phong phú.
- Các bước tiến hành:
 - + Mỗi tháng bổ sung sách. Sau khi nhập tài liệu về tiến hành phân loại.
 - + Phân loại tài liệu: nhân viên thư viện phân loại tài liệu thành các loại như: theo tác giả, theo thể loại,....
 - + Đánh mã sách: Sau khi gán mã cho từng loại sách, cho từng tài liệu bằng chữ và số. Mã được đánh theo quy định của từng thể loại.
 - + Sắp xếp tài liệu: Sau khi gắn mã số cho tài liệu xong, nhân viên thư viện sẽ sắp xếp tài liệu vào các giá sách tương ứng.

3.2 Quy trình bán sách

- Thời gian: thực hiện mỗi khi cửa hàng có khách hàng mua sách.
- Tác nhân tham gia vào quá trình bán sách: nhân viên cửa hàng, khác hàng.
- Vai trò: đáp ứng nhu cầu của khách hàng.
- Các bước tiến hành:
 - + Khi vào cửa hàng mua sách, khách hàng có thể bảo nhân viên tìm sách mình muốn tìm.
 - + Khi tìm xong khách hàng tiến hành yêu cầu thanh toán.
 - + Nhân viên tiến hành kiểm tra thông tin sách như mã sách,....
 - + Sau đó nhân viên tiến hành lập hóa đơn thanh toán.
 - + In hóa đơn thanh toán cho khách.

3.3 Quy trình quản lý sách

- Bất cứ khi nào muốn xem thông tin của sách.
- Tác nhân tham gia vào quá trình quản lý sách: Quản lý, nhân viên.
- Vai trò: giúp tìm kiếm thông tin và số lượng tồn kho của sách.
- Các bước tiến hành:
 - + Tìm tài liệu theo mã sách: hệ thống sẽ hiển thị danh sách sách theo mã sách.

- + Tìm kiếm sách theo thể loại: hệ thống sẽ hiển thị danh sách sách theo thể loại.
- + Tìm kiếm sách theo tác giả: hệ thống sẽ hiển thị danh sách sách theo tác giả.
- + Tìm kiếm sách theo nhà xuất bản: hệ thống sẽ hiển thị danh sách sách theo nhà xuất bản.
- + Tìm kiếm sách theo tên: hệ thống sẽ hiển thị danh sách sách theo tên sách.

3.4 Quy trình quản lý các thông tin khác như tác giả, thể loại, nhà xuất bản

- Bất cứ khi nào có sách mới có tác giả, thể loại hoặc nhà xuất bản khác.
- Tác nhân tham gia vào quá trình: quản lí.
- Vai trò: Giúp cho cửa hàng sách thêm phong phú về thể loại sách và tác giả.
- Các bước tiến hành:
 - + Khi nhập sách mà có một thể loại mới, một tác giả mới hoặc nhập sách từ một nhà xuất bản mới.
 - + Quản lí tiến hành nhập thông tin mới vào hệ thống.

3.5 Quy trình quản lý nhân viên

- Khi tuyển thêm nhân viên hoặc sa thải nhân viên.
- Tác nhân tham gia vào quản lí nhân viên: Sếp, quản lí.
- Vai trò: Nhằm tuyển được nhân viên có trình độ tốt nhằm nâng cao chất lượng của cửa hàng.
- Các bước tiến hành:
 - + Quản lí tiến hành gửi đề xuất cần tuyển thêm hay sa thải nhân viên thiếu năng lực cho sếp.
 - + Nếu đề xuất được thông qua quản lí tiến hành tuyển nhân viên hay sa thải nhân viên.
 - + Khi tìm được nhân viên đạt yêu cầu quản lí tiến hành nhập thông tin nhân viên vào hệ thống (khi sa thải nhân viên quản lí tiến hành xóa nhân viên đó khỏi hệ thống).

CHƯƠNG 4: PHÂN TÍCH THIẾT KẾ HỆ THỐNG

4.1 Các chức năng của hệ thống

4.1.1 Quản lý hệ thống

- Quản lý người dùng.
- Đăng nhập.
- Thay đổi mật khẩu.

4.1.2 Quản lý nhập sách

- Thêm phiếu nhập.
- Sửa phiếu nhập.
- Xóa phiếu nhập.
- Tìm kiếm.

4.1.3 Quản lý bán sách

- Tạo hóa đơn.
- Sửa hóa đơn.
- Xóa hóa đơn.
- Tìm kiếm
- In hóa đơn.

4.1.4 Quản lý sách

- Tìm kiếm kho sách theo thông tin sách.
- Kiểm tra số lượng.

4.1.5 Quản lý nhân viên

- Thêm nhân viên.
- Sửa thông tin nhân viên.
- Sa thải nhân viên.
- Tìm kiếm

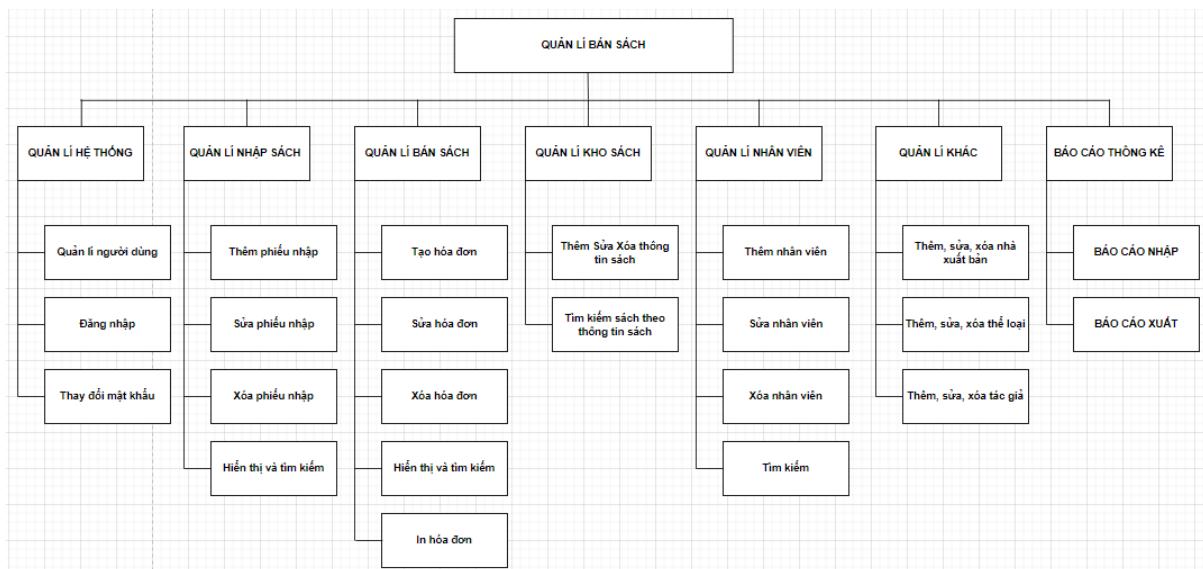
4.1.6 Quản lý khác

- Quản lý nhà xuất bản
 - + Thêm nhà xuất bản.
 - + Sửa thông tin nhà xuất bản.
 - + Xóa nhà xuất bản.
- Quản lý thể loại sách
 - + Thêm thể loại sách.

- + Sửa thông tin thẻ loại sách.
- + Xóa thẻ loại sách.
- Quản lý tác giả.
- + Thêm tác giả.
- + Sửa thông tin tác giả.
- + Xóa tác giả.

4.1.7 Báo cáo thông kê

- Báo cáo nhập sách.
- Báo cáo bán sách.



Hình 4.0: Biểu đồ phân rã chức năng

4.2 Biểu đồ Usecase

4.2.1 Danh sách các actor của hệ thống

Bảng 4.1: Danh sách actor của hệ thống:

STT	Tên Actor	Ý nghĩa
1	Sép	Có toàn quyền tương tác với hệ thống, có quyền điều khiển cũng như kiểm soát mọi hoạt động của hệ thống.
2	Quản lí	Thực hiện các chức năng: đăng nhập, thay đổi mật khẩu, quản lí nhập, bán, quản lí sách, quản lí nhân viên, quản lí các mục khác, báo cáo và thông kê.
3	Nhân viên	Thực hiện các chức năng : đăng nhập, thay đổi mật khẩu, quản lí bán sách, quản lí sách, báo cáo và thông kê.

4.2.2 Danh sách Usecase của hệ thống

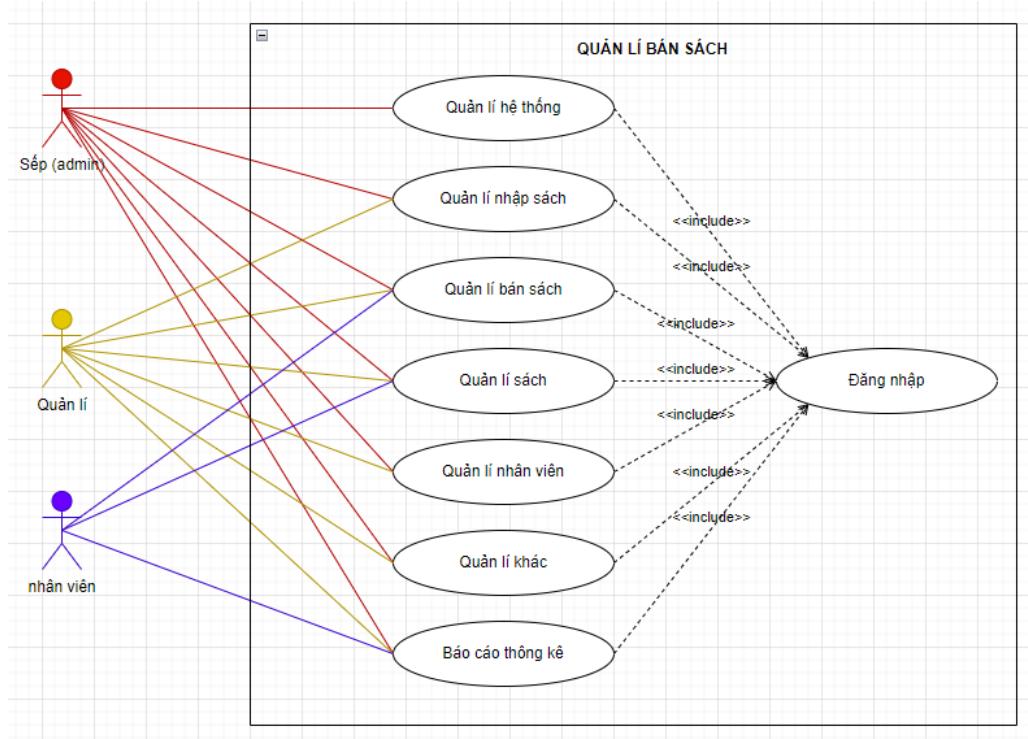
Bảng 4.2: Danh sách usecase của hệ thống

STT	Tên Actor	Ý nghĩa
1	Đăng nhập	Cho phép người dùng đăng nhập vào hệ thống
2	Quản lý người dùng	Cho phép Sếp (admin) thêm sửa xóa phân quyền cho người dùng.
3	Thay đổi mật khẩu	Cho phép người sử dụng thay đổi mật khẩu
4	Thêm phiếu nhập	Cho phép người sử dụng thêm thông tin phiếu nhập khi nhập sách về cửa hàng
5	Sửa thông tin phiếu nhập	Cho phép người dùng sửa thông tin phiếu nhập
6	Xóa phiếu nhập	Cho phép người dùng xóa phiếu nhập
7	Tìm kiếm phiếu nhập	Cho phép người dùng tìm kiếm phiếu nhập
8	Tạo hóa đơn	Cho phép người dùng tạo hóa đơn khi khách hàng mua sách
9	Sửa thông tin hóa đơn	Cho phép người dùng sửa thông tin hóa đơn
10	Xóa hóa đơn	Cho phép người dùng xóa hóa đơn
11	Tìm kiếm hóa đơn	Cho phép người dùng tìm kiếm hóa đơn
12	In hóa đơn	Cho phép người dùng in hóa đơn cho khách hàng
13	Tìm kiếm sách theo thông tin sách	Cho phép người dùng tìm các thông tin của sách và số lượng
14	Thêm nhân viên	Cho phép người dùng thêm thông tin nhân viên
15	Sửa nhân viên	Cho phép người dùng sửa thông tin nhân viên
16	Xóa nhân viên	Cho phép người dùng xóa nhân viên
17	Tìm kiếm nhân viên	Cho phép người dùng tìm kiếm nhân viên
18	Thêm nhà xuất bản	Cho phép người dùng thêm nhà xuất bản khi nhập sách từ một nhà xuất bản mới
19	Sửa thông tin nhà xuất bản	Cho phép người dùng sửa thông tin nhà xuất bản
20	Xóa nhà xuất bản	Cho phép người dùng xóa thông tin nhà xuất bản
21	Thêm thể loại	Cho phép người dùng thêm thể loại của sách

22	Sửa thẻ loại	Cho phép người dùng sửa thông tin thẻ loại sách
23	Thêm tác giả	Cho phép người dùng thêm một tác giả mới của một quyển sách có tác giả mới
24	Sửa thông tin tác giả	Cho phép người dùng sửa thông tin tác giả
25	Xóa tác giả	Cho phép người dùng xóa tác giả
26	Báo cáo nhập	Cho phép người dùng thông kê được sách nhập
27	Báo cáo xuất	Cho phép người dùng thông kê được sách bán ra

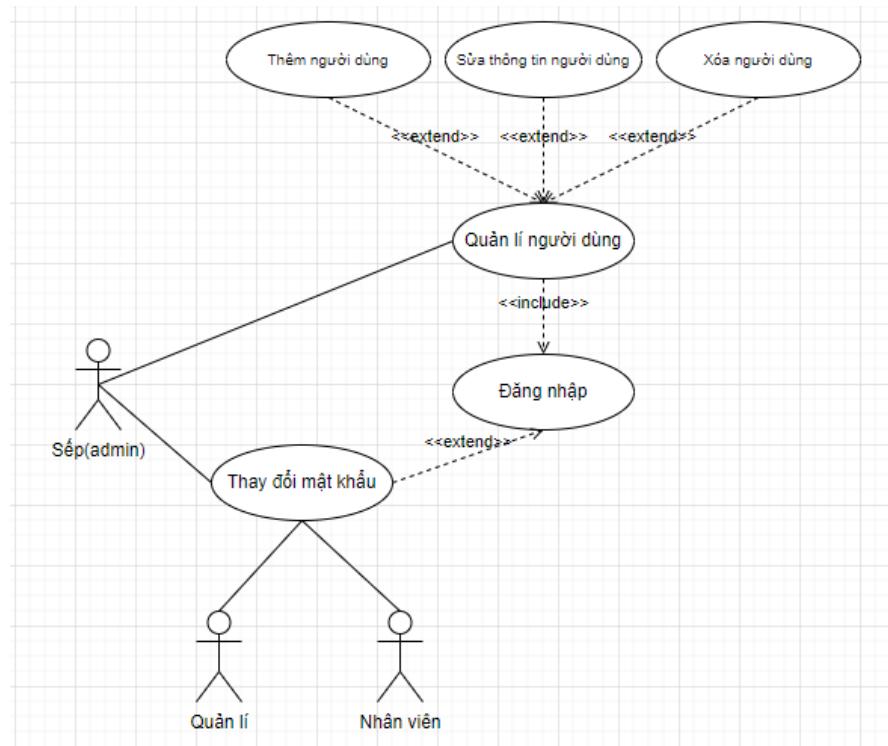
4.2.3. Vẽ biểu đồ Usecase

4.2.3.1 Biểu đồ Usecase tổng quát



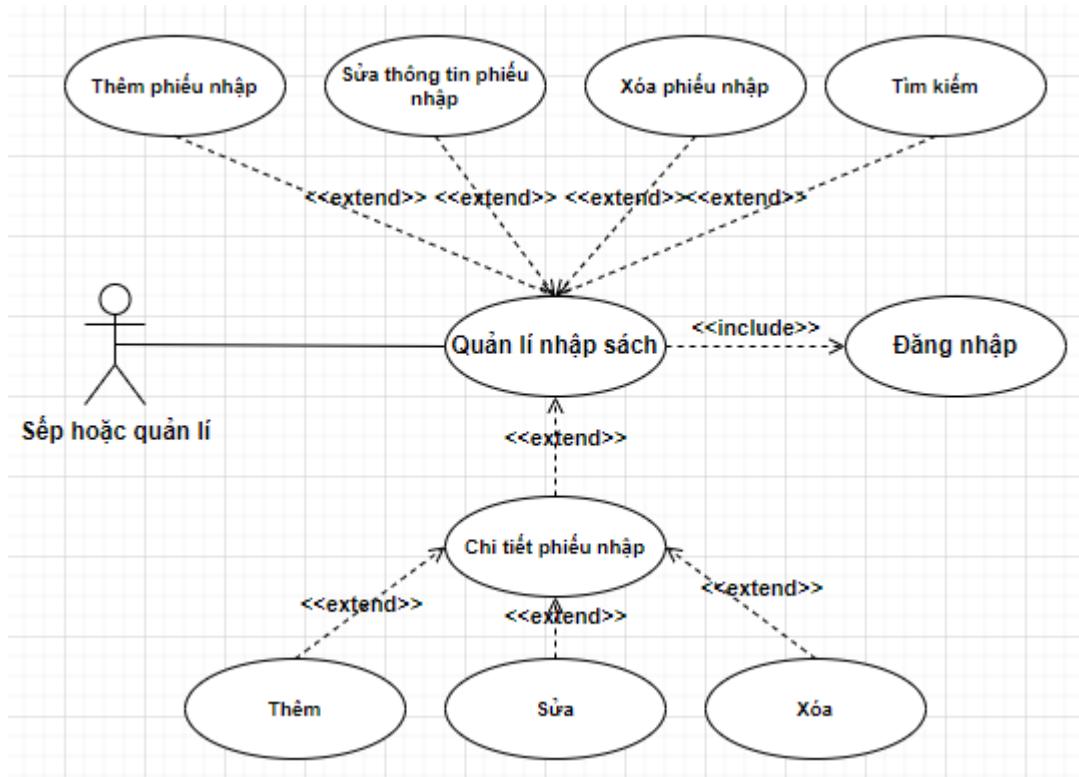
Hình 4.1: Biểu đồ usecase tổng quát

4.2.3.2 Biểu đồ Usecase Quản lý hệ thống



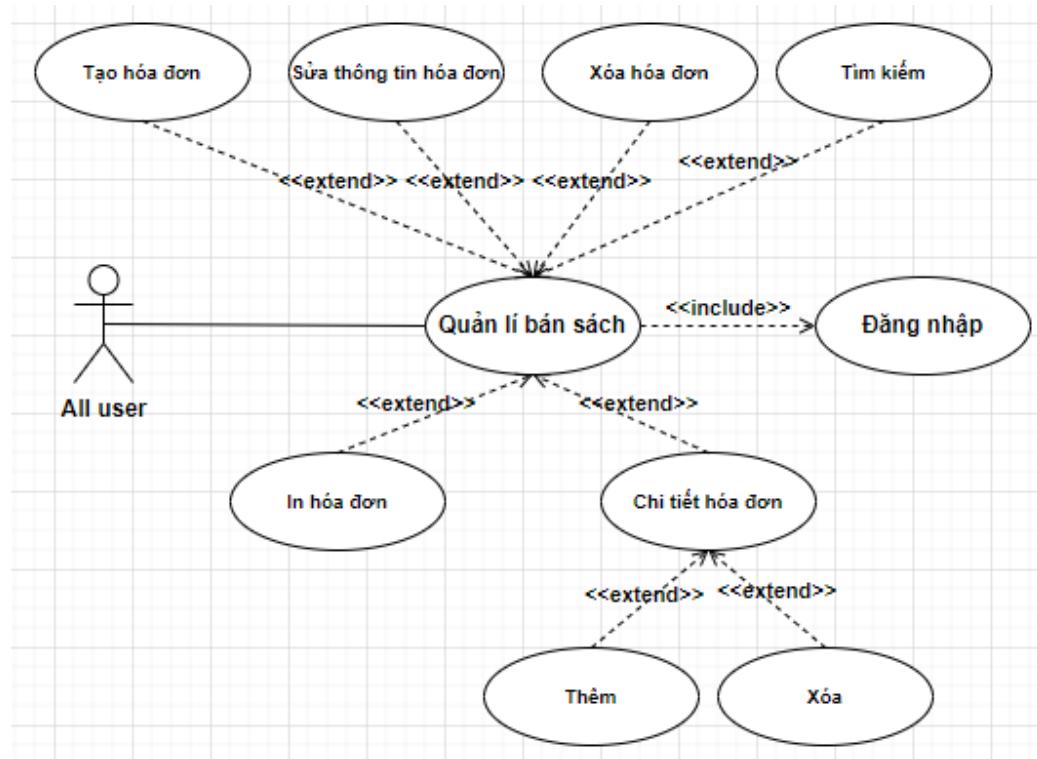
Hình 4.2: Biểu đồ usecase quản lý hệ thống

4.2.3.3 Biểu đồ Usecase Quản lý nhập sách



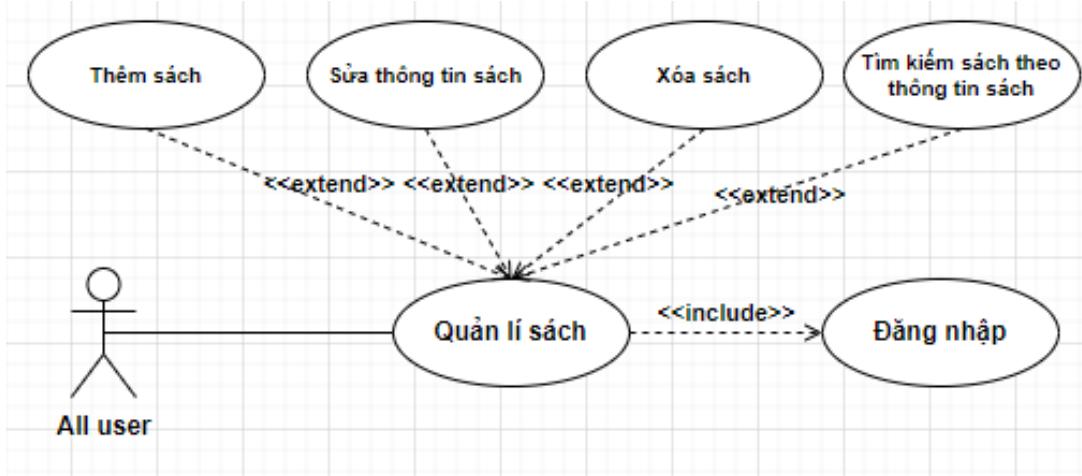
Hình 4.3: Biểu đồ usecase quản lý nhập sách

4.2.3.4 Biểu đồ Usecase Quản lý bán sách



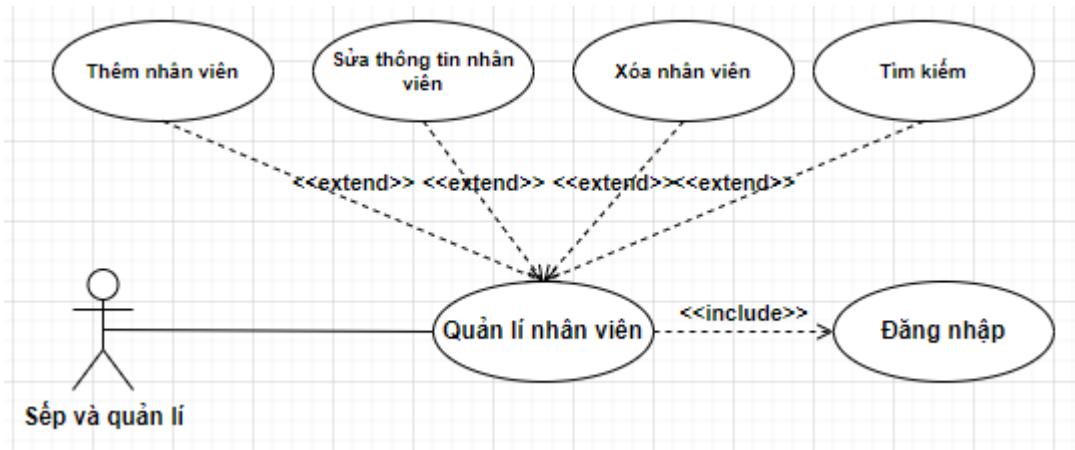
Hình 4.4: Biểu đồ usecase quản lý bán sách

4.2.3.5 Biểu đồ Usecase Quản lý sách



Hình 4.5: Biểu đồ usecase quản lý sách

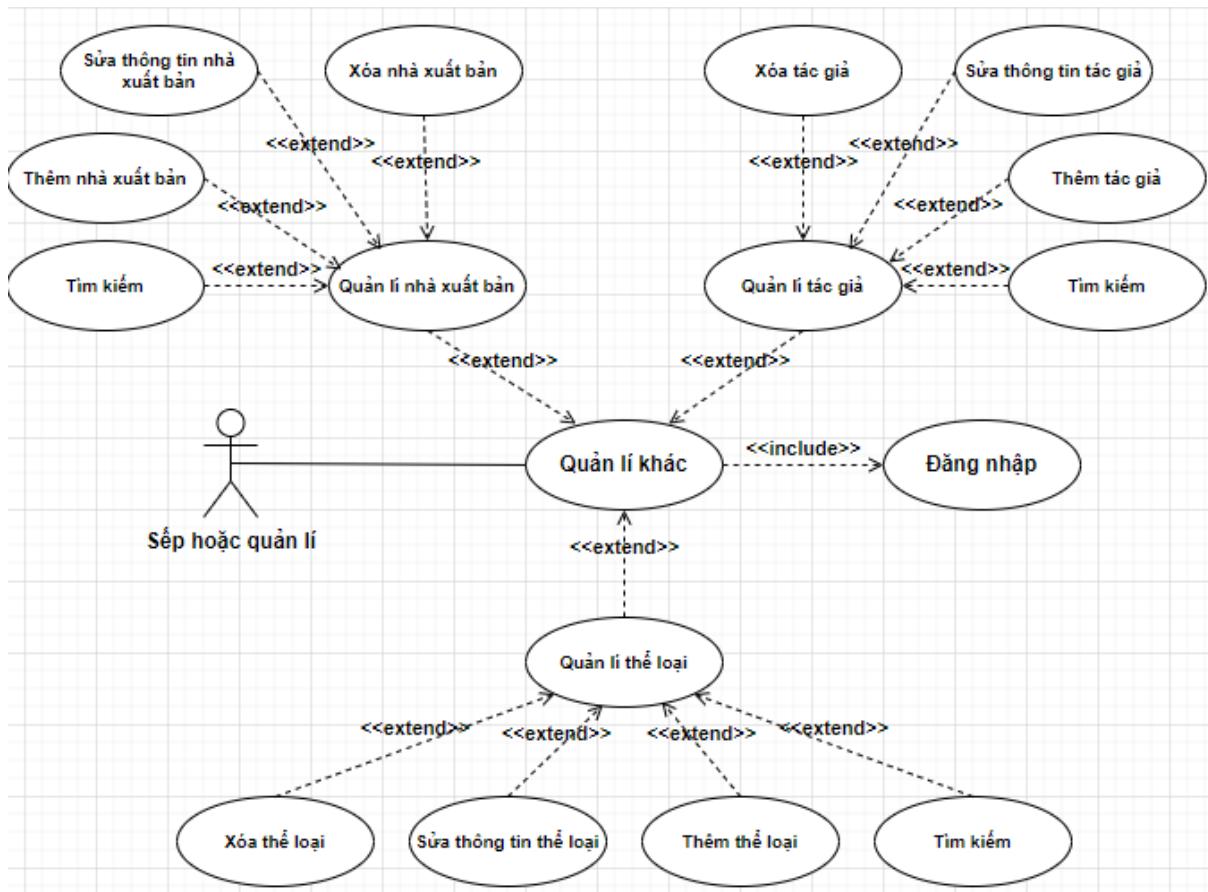
4.2.3.6 Biểu đồ Usecase Quản lý nhân viên



Hình 4.6: Biểu đồ usecase quản lý nhân viên

4.2.3.7 Biểu đồ Usecase Quản lý khác

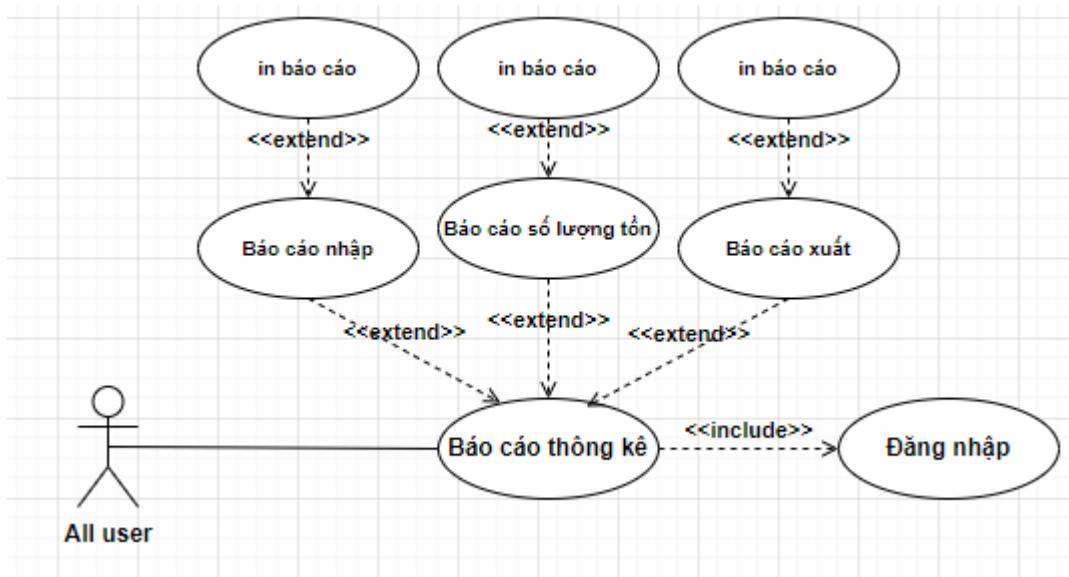
- Gồm có quản lý tác giả, thể loại, nhà xuất bản.



Hình 4.7: Biểu đồ usecase quản lý khác

4.2.3.8 Biểu đồ Usecase Quản lý khác

- Sếp và quản lý có thông kê nhập và thông kê xuất. nhân viên có thông kê xuất.



Hình 4.8: Biểu đồ usecase báo cáo có thông kê

4.2.4 Đặc tả Usecase

4.2.4.1 Đặc tả Usecase quản lý hệ thống

* Đặc tả usecase thêm người dùng

- Tóm tắt: Sếp(Admin) sử dụng usecase này để thực hiện cấp tài khoản cho một nhân viên khi có nhân viên mới, thêm các thông tin như tài khoản, mật khẩu, chức vụ, nhân viên được sử dụng tài khoản này.

- Dòng sự kiện:

+ Dòng sự kiện chính:

- Admin chọn chức quản lý người dùng.
- Hệ thống hiển thị ra giao diện quản lý người dùng.
- Admin nhập các thông tin id, password, chức vụ, nhân viên.
- Admin chọn thêm người dùng.
- Hệ thống kiểm tra thông tin có hợp lệ không và thêm người dùng vào hệ thống.
- Kết thúc usecase.

+ Dòng sự kiện phụ:

- Dòng sự kiện thứ nhất:

- Admin hủy thêm người dùng.
- Hệ thống trả về giao diện chính.
- Kết thúc usecase.

- Dòng sự kiện thứ hai:

- Thông tin admin thêm vào không hợp lệ.
- Hệ thống hiển thị thông báo lỗi.
- Kết thúc usecase.

- Trạng thái hệ thống trước khi Usecase thêm người dùng được thực hiện: admin phải đăng nhập vào hệ thống trước khi thực hiện chức năng này.

- Trạng thái hệ thống sau khi Usecase thêm người dùng được thực thi:

+ Trường hợp thành công: Thông tin người dùng được thêm vào hệ thống.

+ Trường hợp không thành công: Hệ thống thông báo lỗi và không thêm người dùng vào hệ thống.

- Điểm mở rộng: không có.

- Giao diện:

* **Đặc tả usecase sửa thông tin người dùng**

- Tóm tắt: Sếp(Admin) sử dụng usecase này để thực hiện sửa thông tin người dùng liên quan đến chức vụ, nhân viên, thậm chí là mật khẩu của họ.

- Dòng sự kiện:

+ Dòng sự kiện chính:

- Admin chọn chức quản lý người dùng.
- Hệ thống hiển thị ra giao diện quản lý người dùng.
- Admin chọn thông tin người dùng cần sửa và sửa thông tin của người dùng.
- Admin chọn sửa thông tin.
- Hệ thống kiểm tra sự hợp lệ và cập nhật thông tin mới cho người dùng.
- Kết thúc usecase.

+ Dòng sự kiện phụ:

- Dòng sự kiện thứ nhất:

- Admin hủy sửa thông tin người dùng.
- Hệ thống trả về giao diện chính.
- Kết thúc usecase.

- Dòng sự kiện thứ hai:

- Thông tin admin sửa vào không hợp lệ.
- Hệ thống hiển thị thông báo lỗi.
- Kết thúc usecase.

- Trạng thái hệ thống trước khi Usecase sửa thông tin người dùng được thực hiện: admin phải đăng nhập vào hệ thống trước khi thực hiện chức năng này.

- Trạng thái hệ thống sau khi Usecase sửa thông tin người dùng được thực thi:

- + Trường hợp thành công: Thông tin người dùng được cập nhật vào hệ thống.
 - + Trường hợp không thành công: Hệ thống thông báo lỗi và không cập nhật người dùng vào hệ thống.
 - Điểm mở rộng: không có.
 - Giao diện:
- * Đặc tả usecase xóa người dùng**
- Tóm tắt: Sếp(Admin) sử dụng usecase này để thực hiện việc xóa người dùng khỏi hệ thống.
 - Dòng sự kiện:
 - + Dòng sự kiện chính:
 - Admin chọn chức quản lý người dùng.
 - Hệ thống hiển thị ra giao diện quản lý người dùng.
 - Admin chọn người dùng cần xóa.
 - Admin chọn xóa người dùng.
 - Hệ thống kiểm tra sự hợp lệ và xóa người dùng khỏi hệ thống.
 - Kết thúc usecase.
 - + Dòng sự kiện phụ:
 - Dòng sự kiện thứ nhất:
 - Admin hủy xóa người dùng.
 - Hệ thống trả về giao diện chính.
 - Kết thúc usecase.
 - Dòng sự kiện thứ hai:
 - Thông tin admin xóa không hợp lệ.
 - Hệ thống hiển thị thông báo lỗi.
 - Kết thúc usecase.
 - Trạng thái hệ thống trước khi Usecase xóa người dùng được thực hiện: admin phải đăng nhập vào hệ thống trước khi thực hiện chức năng này.
 - Trạng thái hệ thống sau khi Usecase xóa người dùng được thực thi:
 - + Trường hợp thành công: Thông tin người dùng được xóa khỏi hệ thống.
 - + Trường hợp không thành công: Hệ thống thông báo lỗi và không xóa người dùng vào hệ thống.
 - Điểm mở rộng: không có.
 - Giao diện:

* **Đặc tả usecase thay đổi mật khẩu**

- Tóm tắt: Admin, quản lý, nhân viên sử dụng usecase đổi mật khẩu để thực hiện chức năng thay đổi mật khẩu.

- Dòng sự kiện:

+ Dòng sự kiện chính:

- Admin, quản lý, nhân viên chọn chức thay đổi mật khẩu.
- Hệ thống hiển thị ra giao diện thay đổi mật khẩu.
- Admin, quản lý, nhân viên nhập mật khẩu cũ và nhập mật khẩu mới muốn thay đổi.
- Admin, quản lý, nhân viên chọn đổi mật khẩu.
- Hệ thống kiểm tra sự hợp lệ và cập nhật lại mật khẩu cho tài khoản của người dùng.
- Kết thúc usecase.

+ Dòng sự kiện phụ:

- Dòng sự kiện thứ nhất:

- Admin hủy thay đổi mật khẩu.
- Hệ thống trở về giao diện chính.
- Kết thúc usecase.

- Dòng sự kiện thứ hai:

- Thông tin thay đổi mật khẩu không hợp lệ.
- Hệ thống hiển thị thông báo lỗi.
- Kết thúc usecase.

- Trạng thái hệ thống trước khi Usecase thay đổi mật khẩu được thực hiện: admin, quản lý, nhân viên phải đăng nhập vào hệ thống trước khi thực hiện chức năng này.

- Trạng thái hệ thống sau khi Usecase thay đổi mật khẩu được thực thi:

+ Trường hợp thành công: Thông tin mật khẩu mới được cập nhật vào hệ thống.

+ Trường hợp không thành công: Hệ thống thông báo lỗi và không cập nhật mật khẩu mới vào hệ thống.

- Điểm mở rộng: không có.

- Giao diện:

4.2.4.2 Đặc tả Usecase quản lý nhập sách

*** Đặc tả usecase thêm phiếu nhập.**

- Tóm tắt: Admin, quản lý, sử dụng usecase thêm phiếu nhập để thêm thông tin của phiếu nhập khi cửa hàng sách nhập sách về.

- Dòng sự kiện:

+ Dòng sự kiện chính:

- Admin, quản lý chọn chức năng quản lý nhập sách.
- Hệ thống hiển thị ra giao diện quản lý nhập sách.
- Admin, quản lý nhập thông tin như mã phiếu nhập, ngày nhập, nhà xuất bản.
- Admin, quản lý chọn thêm phiếu nhập.
- Hệ thống kiểm tra sự hợp lệ và thêm phiếu nhập mới vào hệ thống.
- Kết thúc usecase.

+ Dòng sự kiện phụ:

- Dòng sự kiện thứ nhất:

- Admin hủy thêm phiếu nhập.
- Hệ thống trả về giao diện chính.
- Kết thúc usecase.

- Dòng sự kiện thứ hai:

- Thông tin nhập không hợp lệ.
- Hệ thống hiển thị thông báo lỗi.
- Kết thúc usecase.

- Trạng thái hệ thống trước khi Usecase thêm phiếu nhập được thực hiện: admin, quản lý phải đăng nhập vào hệ thống trước khi thực hiện chức năng này.

- Trạng thái hệ thống sau khi Usecase thêm phiếu nhập được thực thi:

+ Trường hợp thành công: Thông tin phiếu nhập mới được thêm vào hệ thống.

+ Trường hợp không thành công: Hệ thống thông báo lỗi và không thêm thông tin phiếu nhập mới vào hệ thống.

- Điểm mở rộng: không có.

- Giao diện:

*** Đặc tả usecase sửa phiếu nhập.**

- Tóm tắt: Admin, quản lý, sử dụng usecase sửa phiếu nhập để sửa thông tin của phiếu nhập khi có thông tin thay đổi hoặc sai.

- Dòng sự kiện:

+ Dòng sự kiện chính:

- Admin, quản lý chọn chức năng quản lý nhập sách.
- Hệ thống hiển thị ra giao diện quản lý nhập sách.
- Admin, quản lý chọn phiếu nhập cần sửa và nhập thông tin cần sửa.
- Admin, quản lý chọn sửa phiếu nhập.
- Hệ thống kiểm tra sự hợp lệ và cập nhật phiếu nhập vào hệ thống.
- Kết thúc usecase.

+ Dòng sự kiện phụ:

- Dòng sự kiện thứ nhất:

- Admin hủy sửa phiếu nhập.
- Hệ thống trả về giao diện chính.
- Kết thúc usecase.

- Dòng sự kiện thứ hai:

- Thông tin sửa không hợp lệ.
- Hệ thống hiển thị thông báo lỗi.
- Kết thúc usecase.

- Trạng thái hệ thống trước khi Usecase sửa phiếu nhập được thực hiện: admin, quản lý phải đăng nhập vào hệ thống trước khi thực hiện chức năng này.

- Trạng thái hệ thống sau khi Usecase sửa phiếu nhập được thực thi:

- + Trường hợp thành công: Thông tin phiếu nhập mới được cập nhật vào hệ thống.
- + Trường hợp không thành công: Hệ thống thông báo lỗi và không cập nhật thông tin đã sửa vào hệ thống.

- Điểm mở rộng: không có.

- Giao diện:

* **Đặc tả usecase xóa phiếu nhập.**

- Tóm tắt: Admin, quản lý, sử dụng usecase xóa phiếu nhập để xóa phiếu nhập khỏi hệ thống trong trường hợp trả hàng.

- Dòng sự kiện:

+ Dòng sự kiện chính:

- Admin, quản lý chọn chức năng quản lý nhập sách.
- Hệ thống hiển thị ra giao diện quản lý nhập sách.
- Admin, quản lý chọn phiếu nhập cần xóa.
- Admin, quản lý chọn xóa phiếu nhập.
- Hệ thống kiểm tra sự hợp lệ và xóa phiếu nhập khỏi hệ thống.
- Kết thúc usecase.

+ Dòng sự kiện phụ:

- Dòng sự kiện thứ nhất:

- Admin hủy xóa phiếu nhập.
- Hệ thống trả về giao diện chính.
- Kết thúc usecase.

- Dòng sự kiện thứ hai:

- Xóa không hợp lệ.
- Hệ thống hiển thị thông báo lỗi.
- Kết thúc usecase.

- Trạng thái hệ thống trước khi Usecase xóa phiếu nhập được thực hiện: admin, quản lý phải đăng nhập vào hệ thống trước khi thực hiện chức năng này.

- Trạng thái hệ thống sau khi Usecase xóa phiếu nhập được thực thi:

+ Trường hợp thành công: Thông tin phiếu nhập được xóa khỏi hệ thống.

+ Trường hợp không thành công: Hệ thống thông báo lỗi và không xóa thông tin phiếu nhập khỏi hệ thống.

- Điểm mở rộng: không có.

- Giao diện:

* **Đặc tả usecase chi tiết phiếu nhập.**

- Tóm tắt: Admin, quản lý, sử dụng usecase chi tiết phiếu nhập để thêm các chi tiết của phiếu nhập như sách, số lượng, giá tiền hoặc xóa 1 số thông tin trong đó.

* **Đặc tả usecase thêm trong usecase chi tiết phiếu nhập.**

- Tóm tắt: Admin, quản lý, sử dụng usecase thêm để thêm các thông tin của quyền sách được nhập trong chi tiết của phiếu nhập đó.

- Dòng sự kiện:

+ Dòng sự kiện chính:

- Admin, quản lý chọn chức năng quản lý nhập sách.
- Hệ thống hiển thị ra giao diện quản lý nhập sách.
- Admin, quản lý chọn chức năng chi tiết phiếu nhập.
- Hệ thống hiển thị ra giao diện chi tiết phiếu nhập.
- Admin, quản lý chọn sách đã nhập và nhập số lượng.
- Admin, quản lý chọn thêm.
- Hệ thống kiểm tra sự hợp lệ và thêm sách vào thông tin của phiếu nhập đó vào hệ thống.
- Kết thúc usecase.

+ Dòng sự kiện phụ:

- Dòng sự kiện thứ nhất:

- Admin, quản lý hủy thêm ở chi tiết phiếu nhập.
- Hệ thống trả về giao diện quản lý phiếu nhập.
- Kết thúc usecase.

- Dòng sự kiện thứ hai:

- Thông tin thêm không hợp lệ.
- Hệ thống hiển thị thông báo lỗi.
- Kết thúc usecase.

- Trạng thái hệ thống trước khi Usecase thêm chi tiết phiếu nhập được thực hiện: admin, quản lý phải đăng nhập vào hệ thống trước khi thực hiện chức năng này.

- Trạng thái hệ thống sau khi Usecase thêm chi tiết phiếu nhập được thực thi:

+ Trường hợp thành công: Thông tin sách nhập được thêm vào chi tiết của phiếu nhập đó và thêm thông tin vào hệ thống.

+ Trường hợp không thành công: Hệ thống thông báo lỗi và không thêm vào hệ thống.

- Điểm mở rộng: không có.

- Giao diện quản lý phiếu nhập:

* **Đặc tả usecase sửa trong usecase chi tiết phiếu nhập.**

- Tóm tắt: Admin, quản lý, sử dụng usecase sửa để sửa thông tin các quyền sách trong chi tiết phiếu nhập

- Dòng sự kiện:

+ Dòng sự kiện chính:

- Admin, quản lý chọn chức năng quản lý nhập sách.
- Hệ thống hiển thị ra giao diện quản lý nhập sách.
- Admin, quản lý chọn chức năng chi tiết phiếu nhập.
- Hệ thống hiển thị ra giao diện chi tiết phiếu nhập.
- Admin, quản lý chọn chi tiết phiếu nhập cần sửa và nhập thông tin cần sửa.
- Admin, quản lý chọn sửa.
- Hệ thống kiểm tra sự hợp lệ và sửa thông tin trong hệ thống
- Kết thúc usecase.

+ Dòng sự kiện phụ:

- Dòng sự kiện thứ nhất:

- Admin, quản lý hủy sửa ở chi tiết phiếu nhập.
- Hệ thống trả về giao diện quản lý phiếu nhập.
- Kết thúc usecase.

- Dòng sự kiện thứ hai:

- Thông tin sửa không hợp lệ.
- Hệ thống hiển thị thông báo lỗi.
- Kết thúc usecase.

- Trạng thái hệ thống trước khi Usecase sửa chi tiết phiếu nhập được thực hiện: admin, quản lý phải đăng nhập vào hệ thống trước khi thực hiện chức năng này.

- Trạng thái hệ thống sau khi Usecase sửa chi tiết phiếu nhập được thực hiện:

- + Trường hợp thành công: Thông tin được sửa cập nhật vào hệ thống.
- + Trường hợp không thành công: Hệ thống thông báo lỗi và không sửa ở hệ thống

- Điểm mở rộng: không có.

- Giao diện quản lý phiếu nhập:

*** Đặc tả usecase xóa trong usecase chi tiết phiếu nhập.**

- Tóm tắt: Admin, quản lý, sử dụng usecase xóa để xóa các một số thông tin nhập trong chi tiết phiếu nhập của phiếu nhập

- Dòng sự kiện:

+ Dòng sự kiện chính:

- Admin, quản lý chọn chức năng quản lý nhập sách.
- Hệ thống hiển thị ra giao diện quản lý nhập sách.
- Admin, quản lý chọn chức năng chi tiết phiếu nhập.
- Hệ thống hiển thị ra giao diện chi tiết phiếu nhập.
- Admin, quản lý chọn chi tiết phiếu nhập cần xóa.
- Admin, quản lý chọn xóa.
- Hệ thống kiểm tra sự hợp lệ và xóa thông tin khỏi hệ thống
- Kết thúc usecase.

+ Dòng sự kiện phụ:

- Dòng sự kiện thứ nhất:

- Admin, quản lý hủy xóa ở chi tiết phiếu nhập.
- Hệ thống trả về giao diện quản lý phiếu nhập.
- Kết thúc usecase.

- Dòng sự kiện thứ hai:

- Thông tin xóa không hợp lệ.
- Hệ thống hiển thị thông báo lỗi.
- Kết thúc usecase.

- Trạng thái hệ thống trước khi Usecase xóa chi tiết phiếu nhập được thực hiện: admin, quản lý phải đăng nhập vào hệ thống trước khi thực hiện chức năng này.
- Trạng thái hệ thống sau khi Usecase xóa chi tiết phiếu nhập được thực thi:
 - + Trường hợp thành công: Thông tin được xóa khỏi hệ thống.
 - + Trường hợp không thành công: Hệ thống thông báo lỗi và không xóa thông tin khỏi hệ thống.
- Giao diện quản lý phiếu nhập:
- Giao diện:

4.2.4.3 Đặc tả Usecase quản lý bán sách

*** Đặc tả usecase thêm hóa đơn.**

- Tóm tắt: Admin, quản lý, nhân viên sử dụng usecase thêm hóa đơn để thêm thông tin của hóa đơn khi cửa hàng có khách hàng mua sách.
- Dòng sự kiện:
 - + Dòng sự kiện chính:
 - Admin, quản lý, nhân viên chọn chức năng quản lý bán sách.
 - Hệ thống hiển thị ra giao diện quản lý bán sách.
 - Admin, quản lý, nhân viên nhập thông tin như mã hóa đơn, ngày bán.
 - Admin, quản lý, nhân viên chọn thêm hóa đơn.
 - Hệ thống kiểm tra sự hợp lệ và thêm hóa đơn mới vào hệ thống.
 - Kết thúc usecase.
 - + Dòng sự kiện phụ:
 - Dòng sự kiện thứ nhất:
 - Admin, quản lý, nhân viên hủy thêm hóa đơn.
 - Hệ thống trả về giao diện chính.
 - Kết thúc usecase.
 - Dòng sự kiện thứ hai:
 - Thông tin hóa đơn không hợp lệ.
 - Hệ thống hiển thị thông báo lỗi.
 - Kết thúc usecase.
- Trạng thái hệ thống trước khi Usecase thêm hóa đơn được thực hiện: admin, quản lý, nhân viên phải đăng nhập vào hệ thống trước khi thực hiện chức năng này.
- Trạng thái hệ thống sau khi Usecase thêm hóa đơn được thực thi:
 - + Trường hợp thành công: Thông tin hóa đơn mới được thêm vào hệ thống.

+ Trường hợp không thành công: Hệ thống thông báo lỗi và không thêm thông tin hóa đơn mới vào hệ thống.

- Điểm mở rộng: không có.

- Giao diện:

*** Đặc tả usecase sửa hóa đơn.**

- Tóm tắt: Admin, quản lý, nhân viên sử dụng usecase sửa hóa đơn để sửa thông tin của hóa đơn khi có thông tin thay đổi hoặc sai.

- Dòng sự kiện:

+ Dòng sự kiện chính:

- Admin, quản lý, nhân viên chọn chức năng quản lý bán sách.
- Hệ thống hiển thị ra giao diện quản lý bán sách.
- Admin, quản lý chọn hóa đơn cần sửa và nhập thông tin cần sửa.
- Admin, quản lý chọn sửa hóa đơn.
- Hệ thống kiểm tra sự hợp lệ và cập nhật hóa đơn vào hệ thống.
- Kết thúc usecase.

+ Dòng sự kiện phụ:

- Dòng sự kiện thứ nhất:

- Admin, quản lý, nhân viên hủy sửa hóa đơn.
- Hệ thống trả về giao diện chính.
- Kết thúc usecase.

- Dòng sự kiện thứ hai:

- Thông tin sửa không hợp lệ.
- Hệ thống hiển thị thông báo lỗi.
- Kết thúc usecase.

- Trạng thái hệ thống trước khi UseCase sửa hóa đơn được thực hiện: admin, quản lý, nhân viên phải đăng nhập vào hệ thống trước khi thực hiện chức năng này.

- Trạng thái hệ thống sau khi UseCase sửa hóa đơn được thực hiện:

+ Trường hợp thành công: Thông tin hóa đơn mới được cập nhật vào hệ thống.

+ Trường hợp không thành công: Hệ thống thông báo lỗi và không cập nhật thông tin đã sửa vào hệ thống.

- Điểm mở rộng: không có.

- Giao diện:

* **Đặc tả usecase xóa phiếu nhập.**

- Tóm tắt: Admin, quản lý, nhân viên sử dụng usecase xóa hóa đơn để xóa hóa đơn khỏi hệ thống trong trường hợp trả hàng.

- Dòng sự kiện:

+ Dòng sự kiện chính:

- Admin, quản lý, nhân viên chọn chức năng quản lý bán sách.
- Hệ thống hiển thị ra giao diện quản lý bán sách.
- Admin, quản lý, nhân viên chọn hóa đơn cần xóa.
- Admin, quản lý, nhân viên chọn xóa hóa đơn.
- Hệ thống kiểm tra sự hợp lệ và xóa hóa đơn khỏi hệ thống.
- Kết thúc usecase.

+ Dòng sự kiện phụ:

- Dòng sự kiện thứ nhất:

- Admin hủy xóa hóa đơn.
- Hệ thống trở về giao diện chính.
- Kết thúc usecase.

- Dòng sự kiện thứ hai:

- Xóa không hợp lệ.
- Hệ thống hiển thị thông báo lỗi.
- Kết thúc usecase.

- Trạng thái hệ thống trước khi Usecase xóa hóa đơn được thực hiện: admin, quản lý, nhân viên phải đăng nhập vào hệ thống trước khi thực hiện chức năng này.

- Trạng thái hệ thống sau khi Usecase xóa hóa đơn được thực thi:

+ Trường hợp thành công: Thông tin hóa đơn được xóa khỏi hệ thống.

+ Trường hợp không thành công: Hệ thống thông báo lỗi và không xóa thông tin hóa đơn khỏi hệ thống.

- Điểm mở rộng: không có.

- Giao diện:

* **Đặc tả usecase chi tiết hóa đơn.**

- Tóm tắt: Admin, quản lý, nhân viên sử dụng usecase chi tiết hóa đơn để thêm các chi tiết của hóa đơn như sách, số lượng, giá tiền hoặc xóa 1 số thông tin trong đó.

* **Đặc tả usecase thêm trong usecase chi tiết hóa đơn.**

- Tóm tắt: Admin, quản lý, nhân viên sử dụng usecase thêm để thêm các thông tin của quyền sách được bán trong chi tiết của hóa đơn đó.

- Dòng sự kiện:

+ Dòng sự kiện chính:

- Admin, quản lý, nhân viên chọn chức năng quản lý bán sách.
- Hệ thống hiển thị ra giao diện quản lý bán sách.
- Admin, quản lý, nhân viên chọn chức năng chi tiết hóa đơn.
- Hệ thống hiển thị ra giao diện chi tiết hóa đơn.
- Admin, quản lý, nhân viên chọn sách và nhập số lượng.
- Admin, quản lý, nhân viên chọn thêm.
- Hệ thống kiểm tra sự hợp lệ và thêm sách vào thông tin của hóa đơn đó vào hệ thống.
- Kết thúc usecase.

+ Dòng sự kiện phụ:

- Dòng sự kiện thứ nhất:

- Admin, quản lý hủy thêm ở chi tiết hóa đơn.
- Hệ thống trả về giao diện quản lý hóa đơn.
- Kết thúc usecase.

- Dòng sự kiện thứ hai:

- Thông tin thêm không hợp lệ.
- Hệ thống hiển thị thông báo lỗi.
- Kết thúc usecase.

- Trạng thái hệ thống trước khi Usecase thêm chi tiết hóa đơn được thực hiện: admin, quản lý, nhân viên phải đăng nhập vào hệ thống trước khi thực hiện chức năng này.

- Trạng thái hệ thống sau khi Usecase thêm chi tiết hóa đơn được thực thi:

+ Trường hợp thành công: Thông tin sách bán được thêm vào chi tiết của hóa đơn đó và thêm thông tin vào hệ thống.

+ Trường hợp không thành công: Hệ thống thông báo lỗi và không thêm vào hệ thống.

- Điểm mở rộng: không có.

- Giao diện quản lý bán sách:

* **Đặc tả usecase xóa trong usecase chi tiết hóa đơn.**

- Tóm tắt: Admin, quản lý, nhân viên sử dụng usecase xóa để xóa các một số thông tin bán trong chi tiết hóa đơn của hóa đơn

- Dòng sự kiện:

+ Dòng sự kiện chính:

- Admin, quản lý, nhân viên chọn chức năng quản lý bán sách.
- Hệ thống hiển thị ra giao diện quản lý bán sách.
- Admin, quản lý, nhân viên chọn chức năng chi tiết hóa đơn.
- Hệ thống hiển thị ra giao diện chi tiết hóa đơn.
- Admin, quản lý, nhân viên chọn chi tiết hóa đơn cần xóa.
- Admin, quản lý, nhân viên chọn xóa.
- Hệ thống kiểm tra sự hợp lệ và xóa thông tin khỏi hệ thống
- Kết thúc usecase.

+ Dòng sự kiện phụ:

- Dòng sự kiện thứ nhất:

- Admin, quản lý hủy xóa ở chi tiết hóa đơn.
- Hệ thống trở về giao diện quản lý hóa đơn.
- Kết thúc usecase.

- Dòng sự kiện thứ hai:

- Thông tin xóa không hợp lệ.
- Hệ thống hiển thị thông báo lỗi.
- Kết thúc usecase.

- Trạng thái hệ thống trước khi Usecase xóa chi tiết hóa đơn được thực hiện: admin, quản lý, nhân viên phải đăng nhập vào hệ thống trước khi thực hiện chức năng này.

- Trạng thái hệ thống sau khi Usecase xóa chi tiết hóa đơn được thực thi:

+ Trường hợp thành công: Thông tin được xóa khỏi hệ thống.

+ Trường hợp không thành công: Hệ thống thông báo lỗi và không xóa thông tin khỏi hệ thống.

- Giao diện quản lý hóa đơn:

- Giao diện:

*** Đặc tả usecase in hóa đơn.**

- Tóm tắt: Admin, quản lý, nhân viên sử dụng usecase in hóa đơn để in hóa đơn cho khách hàng.

- Dòng sự kiện:

+ Dòng sự kiện chính:

- Admin, quản lý, nhân viên chọn chức năng quản lý bán sách.
- Hệ thống hiển thị ra giao diện quản lý bán sách.
- Admin, quản lý, nhân viên chọn hóa đơn cần in.
- Admin, quản lý, nhân viên chọn in hóa đơn.
- Hệ thống kiểm tra sự hợp lệ và xuất một bản in hóa đơn đó.

- Kết thúc usecase.

+ Dòng sự kiện phụ:

- Dòng sự kiện thứ nhất:

- Admin hủy in hóa đơn.
- Hệ thống trả về giao diện chính.
- Kết thúc usecase.

- Dòng sự kiện thứ hai:

- in không hợp lệ.
- Hệ thống hiển thị thông báo lỗi.
- Kết thúc usecase.

- Trạng thái hệ thống trước khi Usecase in hóa đơn được thực hiện: admin, quản lý, nhân viên phải đăng nhập vào hệ thống trước khi thực hiện chức năng này.

- Trạng thái hệ thống sau khi Usecase in hóa đơn được thực thi:

- + Trường hợp thành công: Hệ thống xuất một bản in thông tin hóa đơn đó.
- + Trường hợp không thành công: Hệ thống thông báo lỗi và không xuất bản in.
- Điểm mở rộng: không có.
- Giao diện:

4.2.4.4 Đặc tả Usecase quản lý sách.

- Mô tả: Admin sẽ được thêm, sửa, xóa. Quản lý và Nhân viên chỉ được xem thông tin của sách.

*** Đặc tả usecase thêm sách.**

- Tóm tắt: Admin sử dụng usecase thêm sách để thêm thông tin của sách khi cửa hàng có loại sách mới sắp nhập.

- Dòng sự kiện:

+ Dòng sự kiện chính:

- Admin chọn chức năng quản lý sách.
- Hệ thống hiển thị ra giao diện quản lý sách.
- Admin nhập thông tin như mã sách, tên sách, giá nhập, giá bán, số lượng tồn, thể loại, tác giả, nhà xuất bản.
- Admin chọn thêm sách.
- Hệ thống kiểm tra sự hợp lệ và thêm sách mới vào hệ thống.
- Kết thúc usecase.

+ Dòng sự kiện phụ:

- Dòng sự kiện thứ nhất:

- Admin hủy thêm sách.
- Hệ thống trả về giao diện chính.
- Kết thúc usecase.

- Dòng sự kiện thứ hai:

- Thông tin sách không hợp lệ.
- Hệ thống hiển thị thông báo lỗi.
- Kết thúc usecase.

- Trạng thái hệ thống trước khi Usecase thêm sách được thực hiện: admin phải đăng nhập vào hệ thống trước khi thực hiện chức năng này.

- Trạng thái hệ thống sau khi Usecase thêm sách được thực thi:

+ Trường hợp thành công: Thông tin sách mới được thêm vào hệ thống.

+ Trường hợp không thành công: Hệ thống thông báo lỗi và không thêm thông tin sách mới vào hệ thống.

- Điểm mở rộng: không có.

- Giao diện:

* **Đặc tả usecase sửa thông tin sách.**

- Tóm tắt: Admin sử dụng usecase sửa thông tin sách để thêm thông tin của sách khi thông tin bị sai hoặc thay đổi.

- Dòng sự kiện:

+ Dòng sự kiện chính:

- Admin chọn chức năng quản lý sách.
- Hệ thống hiển thị ra giao diện quản lý sách.
- Admin chọn sách cần sửa và nhập thông tin cần sửa.
- Admin chọn sửa sách.
- Hệ thống kiểm tra sự hợp lệ và cập nhật thông tin mới vào hệ thống.
- Kết thúc usecase.

+ Dòng sự kiện phụ:

- Dòng sự kiện thứ nhất:

- Admin hủy sửa sách.
- Hệ thống trả về giao diện chính.
- Kết thúc usecase.

- Dòng sự kiện thứ hai:

- Thông tin sửa không hợp lệ.
- Hệ thống hiển thị thông báo lỗi.

- Kết thúc usecase.

- Trạng thái hệ thống trước khi Usecase sửa sách được thực hiện: admin phải đăng nhập vào hệ thống trước khi thực hiện chức năng này.

- Trạng thái hệ thống sau khi Usecase sửa sách được thực thi:

+ Trường hợp thành công: Thông tin sách mới được cập nhật vào hệ thống.

+ Trường hợp không thành công: Hệ thống thông báo lỗi và không cập nhật thông tin sách mới vào hệ thống.

- Điểm mở rộng: không có.

- Giao diện:

* **Đặc tả usecase xóa sách.**

- Tóm tắt: Admin sử dụng usecase xóa sách để xóa thông tin của sách khi cửa hàng sẽ không bán quyền sách này nữa.

- Dòng sự kiện:

+ Dòng sự kiện chính:

- Admin chọn chức năng quản lý sách.
- Hệ thống hiển thị ra giao diện quản lý sách.
- Admin chọn sách muốn xóa.
- Admin chọn xóa sách.
- Hệ thống kiểm tra sự hợp lệ và xóa sách hệ thống.
- Kết thúc usecase.

+ Dòng sự kiện phụ:

- Dòng sự kiện thứ nhất:

- Admin hủy xóa sách.
- Hệ thống trở về giao diện chính.
- Kết thúc usecase.

- Dòng sự kiện thứ hai:

- Thông tin xóa không hợp lệ.
- Hệ thống hiển thị thông báo lỗi.
- Kết thúc usecase.

- Trạng thái hệ thống trước khi Usecase xóa sách được thực hiện: admin phải đăng nhập vào hệ thống trước khi thực hiện chức năng này.

- Trạng thái hệ thống sau khi Usecase xóa sách được thực thi:

+ Trường hợp thành công: Thông tin sách bị xóa khỏi hệ thống.

+ Trường hợp không thành công: Hệ thống thông báo lỗi và không xóa sách khỏi hệ thống.

- Điểm mở rộng: không có.

- Giao diện:

4.2.4.5 Đặc tả Usecase quản lý nhân viên.

* Đặc tả usecase thêm nhân viên.

- Tóm tắt: Admin, quản lý sử dụng usecase thêm nhân viên để thêm thông tin của nhân viên khi cửa hàng tuyển được nhân viên mới.

- Dòng sự kiện:

+ Dòng sự kiện chính:

- Admin, quản lý chọn chức năng quản lý nhân viên.
- Hệ thống hiển thị ra giao diện quản lý nhân viên.
- Admin, quản lý nhập thông tin như mã nhân viên, tên nhân viên, ngày sinh, cccd, sđt, đc, lịch làm.
- Admin, quản lý chọn thêm nhân viên.
- Hệ thống kiểm tra sự hợp lệ và thêm nhân viên mới vào hệ thống.
- Kết thúc usecase.

+ Dòng sự kiện phụ:

- Dòng sự kiện thứ nhất:

- Admin hủy thêm nhân viên.
- Hệ thống trả về giao diện chính.
- Kết thúc usecase.

- Dòng sự kiện thứ hai:

- Thông tin nhân viên không hợp lệ.
- Hệ thống hiển thị thông báo lỗi.
- Kết thúc usecase.

- Trạng thái hệ thống trước khi Usecase thêm nhân viên được thực hiện: admin, quản lý phải đăng nhập vào hệ thống trước khi thực hiện chức năng này.

- Trạng thái hệ thống sau khi Usecase thêm nhân viên được thực hiện:

+ Trường hợp thành công: Thông tin nhân viên mới được thêm vào hệ thống.

+ Trường hợp không thành công: Hệ thống thông báo lỗi và không thêm thông tin nhân viên mới vào hệ thống.

- Điểm mở rộng: không có.

- Giao diện:

* **Đặc tả usecase sửa nhân viên.**

- Tóm tắt: Admin, quản lý sử dụng usecase sửa nhân viên để sửa thông tin của nhân viên khi thông tin bị sai hoặc thay đổi.

- Dòng sự kiện:

+ Dòng sự kiện chính:

- Admin, quản lý chọn chức năng quản lý nhân viên.
- Hệ thống hiển thị ra giao diện quản lý nhân viên.
- Admin, quản lý chọn nhân viên cần sửa và nhập thông tin cần sửa.
- Admin, quản lý chọn sửa nhân viên.
- Hệ thống kiểm tra sự hợp lệ và cập nhật thông tin nhân viên vào hệ thống.
- Kết thúc usecase.

+ Dòng sự kiện phụ:

- Dòng sự kiện thứ nhất:

- Admin hủy sửa nhân viên.
- Hệ thống trở về giao diện chính.
- Kết thúc usecase.

- Dòng sự kiện thứ hai:

- Thông tin sửa nhân viên không hợp lệ.
- Hệ thống hiển thị thông báo lỗi.
- Kết thúc usecase.

- Trạng thái hệ thống trước khi Usecase sửa nhân viên được thực hiện: admin, quản lý phải đăng nhập vào hệ thống trước khi thực hiện chức năng này.

- Trạng thái hệ thống sau khi Usecase sửa nhân viên được thực thi:

+ Trường hợp thành công: Thông tin nhân viên mới sửa được cập nhật vào hệ thống.

+ Trường hợp không thành công: Hệ thống thông báo lỗi và không cập nhật thông tin nhân viên mới sửa vào hệ thống.

- Điểm mở rộng: không có.

- Giao diện:

* **Đặc tả usecase xóa nhân viên.**

- Tóm tắt: Admin, quản lý sử dụng usecase xóa nhân viên để xóa thông tin của nhân viên khi nhân viên nghỉ việc.

- Dòng sự kiện:

+ Dòng sự kiện chính:

- Admin, quản lý chọn chức năng quản lý nhân viên.
- Hệ thống hiển thị ra giao diện quản lý nhân viên.
- Admin, quản lý chọn nhân viên cần xóa.
- Admin, quản lý chọn xóa nhân viên.
- Hệ thống kiểm tra sự hợp lệ và xóa nhân viên khỏi hệ thống.
- Kết thúc usecase.

+ Dòng sự kiện phụ:

- Dòng sự kiện thứ nhất:

- Admin hủy xóa nhân viên.
- Hệ thống trở về giao diện chính.
- Kết thúc usecase.

- Dòng sự kiện thứ hai:

- Xóa nhân viên không hợp lệ.
- Hệ thống hiển thị thông báo lỗi.
- Kết thúc usecase.

- Trạng thái hệ thống trước khi Usecase xóa nhân viên được thực hiện: admin, quản lý phải đăng nhập vào hệ thống trước khi thực hiện chức năng này.

- Trạng thái hệ thống sau khi Usecase xóa nhân viên được thực thi:

+ Trường hợp thành công: nhân viên bị xóa khỏi hệ thống.

+ Trường hợp không thành công: Hệ thống thông báo lỗi và không xóa nhân viên khỏi hệ thống.

- Điểm mở rộng: không có.

- Giao diện:

4.2.4.6 Đặc tả Usecase quản lý nhà xuất bản.

* **Đặc tả usecase thêm nhà xuất bản.**

- Tóm tắt: Admin, quản lý sử dụng usecase thêm nhà xuất bản để thêm thông tin của nhà xuất bản khi cửa hàng hợp tác với một nhà xuất bản mới.

- Dòng sự kiện:

+ Dòng sự kiện chính:

- Admin, quản lý chọn chức năng quản lý nhà xuất bản.
- Hệ thống hiển thị ra giao diện quản lý nhà xuất bản.
- Admin, quản lý nhập thông tin như mã nhà xuất bản, tên nhà xuất bản, sđt, đc.

- Admin, quản lý chọn thêm nhà xuất bản.
- Hệ thống kiểm tra sự hợp lệ và thêm nhà xuất bản mới vào hệ thống.
- Kết thúc usecase.

+ Dòng sự kiện phụ:

- Dòng sự kiện thứ nhất:

- Admin, quản lý hủy thêm nhà xuất bản.
- Hệ thống trả về giao diện chính.
- Kết thúc usecase.

- Dòng sự kiện thứ hai:

- Thông tin nhà xuất bản không hợp lệ.
- Hệ thống hiển thị thông báo lỗi.
- Kết thúc usecase.

- Trạng thái hệ thống trước khi Usecase thêm nhà xuất bản được thực hiện: admin, quản lý phải đăng nhập vào hệ thống trước khi thực hiện chức năng này.

- Trạng thái hệ thống sau khi Usecase thêm nhà xuất bản được thực hiện:

- + Trường hợp thành công: Thông tin nhà xuất bản mới được thêm vào hệ thống.
- + Trường hợp không thành công: Hệ thống thông báo lỗi và không thêm thông tin nhà xuất bản mới vào hệ thống.

- Điểm mở rộng: không có.

- Giao diện:

*** Đặc tả usecase sửa nhà xuất bản.**

- Tóm tắt: Admin, quản lý sử dụng usecase sửa nhà xuất bản để sửa thông tin của nhà xuất bản khi có thông tin sai hoặc thay đổi.

- Dòng sự kiện:

+ Dòng sự kiện chính:

- Admin, quản lý chọn chức năng quản lý nhà xuất bản.
- Hệ thống hiển thị ra giao diện quản lý nhà xuất bản.
- Admin, quản lý chọn nhà xuất bản cần sửa và nhập thông tin cần sửa.
- Admin, quản lý chọn sửa nhà xuất bản.
- Hệ thống kiểm tra sự hợp lệ và cập nhật thông tin sửa nhà xuất bản vào hệ thống.
- Kết thúc usecase.

+ Dòng sự kiện phụ:

- Dòng sự kiện thứ nhất:

- Admin, quản lý hủy sửa nhà xuất bản.
- Hệ thống trả về giao diện chính.
- Kết thúc usecase.

- Dòng sự kiện thứ hai:

- Thông tin sửa nhà xuất bản không hợp lệ.
- Hệ thống hiển thị thông báo lỗi.
- Kết thúc usecase.

- Trạng thái hệ thống trước khi Usecase sửa nhà xuất bản được thực hiện: admin, quản lý phải đăng nhập vào hệ thống trước khi thực hiện chức năng này.

- Trạng thái hệ thống sau khi Usecase sửa nhà xuất bản được thực thi:

+ Trường hợp thành công: Thông tin nhà xuất bản mới sửa được cập nhật vào hệ thống.

+ Trường hợp không thành công: Hệ thống thông báo lỗi và không cập nhật thông tin nhà xuất bản mới sửa vào hệ thống.

- Điểm mở rộng: không có.

- Giao diện:

* **Đặc tả usecase xóa nhà xuất bản.**

- Tóm tắt: Admin, quản lý sử dụng usecase xóa nhà xuất bản để xóa thông tin của nhà xuất bản khi cửa hàng không hợp tác với nhà xuất bản.

- Dòng sự kiện:

+ Dòng sự kiện chính:

- Admin, quản lý chọn chức năng quản lý nhà xuất bản.
- Hệ thống hiển thị ra giao diện quản lý nhà xuất bản.
- Admin, quản lý chọn nhà xuất bản cần xóa.
- Admin, quản lý chọn xóa nhà xuất bản.
- Hệ thống kiểm tra sự hợp lệ và xóa nhà xuất bản khỏi hệ thống.
- Kết thúc usecase.

+ Dòng sự kiện phụ:

- Dòng sự kiện thứ nhất:

- Admin, quản lý hủy xóa nhà xuất bản.
- Hệ thống trả về giao diện chính.
- Kết thúc usecase.

- Dòng sự kiện thứ hai:

- Xóa nhà xuất bản không hợp lệ.

- Hệ thống hiển thị thông báo lỗi.
- Kết thúc usecase.

- Trạng thái hệ thống trước khi Usecase xóa nhà xuất bản được thực hiện: admin, quản lý phải đăng nhập vào hệ thống trước khi thực hiện chức năng này.

- Trạng thái hệ thống sau khi Usecase xóa nhà xuất bản được thực thi:

+ Trường hợp thành công: Thông tin nhà xuất bản bị xóa khỏi hệ thống.

+ Trường hợp không thành công: Hệ thống thông báo lỗi và không xóa thông tin nhà xuất bản khỏi hệ thống.

- Điểm mở rộng: không có.

- Giao diện:

4.2.4.7 Đặc tả Usecase quản lý thẻ loại.

*** Đặc tả usecase thêm thẻ loại.**

- Tóm tắt: Admin, quản lý sử dụng usecase thêm thẻ loại để thêm thông tin của thẻ loại khi cửa hàng có sách có thẻ loại mới.

- Dòng sự kiện:

+ Dòng sự kiện chính:

- Admin, quản lý chọn chức năng quản lý thẻ loại.
- Hệ thống hiển thị ra giao diện quản lý thẻ loại.
- Admin, quản lý nhập thông tin như mã thẻ loại và tên thẻ loại.
- Admin, quản lý chọn thêm thẻ loại.
- Hệ thống kiểm tra sự hợp lệ và thêm thẻ loại mới vào hệ thống.
- Kết thúc usecase.

+ Dòng sự kiện phụ:

- Dòng sự kiện thứ nhất:

- Admin, quản lý hủy thêm thẻ loại.
- Hệ thống trả về giao diện chính.
- Kết thúc usecase.

- Dòng sự kiện thứ hai:

- Thông tin thẻ loại không hợp lệ.
- Hệ thống hiển thị thông báo lỗi.
- Kết thúc usecase.

- Trạng thái hệ thống trước khi Usecase thêm thẻ loại được thực hiện: admin, quản lý phải đăng nhập vào hệ thống trước khi thực hiện chức năng này.

- Trạng thái hệ thống sau khi Usecase thêm thẻ loại được thực thi:

- + Trường hợp thành công: Thông tin thẻ loại mới được thêm vào hệ thống.
 - + Trường hợp không thành công: Hệ thống thông báo lỗi và không thêm thông tin thẻ loại mới vào hệ thống.
 - Điểm mở rộng: không có.
 - Giao diện:
- * Đặc tả usecase sửa thẻ loại.**
- Tóm tắt: Admin, quản lý sử dụng usecase sửa thẻ loại để sửa thông tin của thẻ loại khi có thông tin sai hoặc thay đổi.
 - Dòng sự kiện:
 - + Dòng sự kiện chính:
 - Admin, quản lý chọn chức năng quản lý thẻ loại.
 - Hệ thống hiển thị ra giao diện quản lý thẻ loại.
 - Admin, quản lý chọn thẻ loại cần sửa và sửa thông tin cần sửa
 - Admin, quản lý chọn sửa thẻ loại.
 - Hệ thống kiểm tra sự hợp lệ và cập nhật thông tin thẻ loại mới sửa vào hệ thống.
 - Kết thúc usecase.
 - + Dòng sự kiện phụ:
 - Dòng sự kiện thứ nhất:
 - Admin, quản lý hủy sửa thẻ loại.
 - Hệ thống trả về giao diện chính.
 - Kết thúc usecase.
 - Dòng sự kiện thứ hai:
 - Thông tin thẻ loại sửa không hợp lệ.
 - Hệ thống hiển thị thông báo lỗi.
 - Kết thúc usecase.
 - Trạng thái hệ thống trước khi Usecase sửa thẻ loại được thực hiện: admin, quản lý phải đăng nhập vào hệ thống trước khi thực hiện chức năng này.
 - Trạng thái hệ thống sau khi Usecase sửa thẻ loại được thực thi:
 - + Trường hợp thành công: Thông tin thẻ loại mới sửa được cập nhật vào hệ thống.
 - + Trường hợp không thành công: Hệ thống thông báo lỗi và không cập nhật thông tin thẻ loại mới sửa vào hệ thống.
 - Điểm mở rộng: không có.
 - Giao diện:

* **Đặc tả usecase xóa thẻ loại.**

- Tóm tắt: Admin, quản lý sử dụng usecase xóa thẻ loại để xóa thông tin của thẻ loại trong một số trường hợp cần xóa.

- Dòng sự kiện:

+ Dòng sự kiện chính:

- Admin, quản lý chọn chức năng quản lý thẻ loại.
- Hệ thống hiển thị ra giao diện quản lý thẻ loại.
- Admin, quản lý chọn thẻ loại cần xóa.
- Admin, quản lý chọn xóa thẻ loại.
- Hệ thống kiểm tra sự hợp lệ và xóa thẻ loại khỏi hệ thống.
- Kết thúc usecase.

+ Dòng sự kiện phụ:

- Dòng sự kiện thứ nhất:

- Admin, quản lý hủy xóa thẻ loại.
- Hệ thống trả về giao diện chính.
- Kết thúc usecase.

- Dòng sự kiện thứ hai:

- xóa thẻ loại không hợp lệ.
- Hệ thống hiển thị thông báo lỗi.
- Kết thúc usecase.

- Trạng thái hệ thống trước khi Usecase xóa thẻ loại được thực hiện: admin, quản lý phải đăng nhập vào hệ thống trước khi thực hiện chức năng này.

- Trạng thái hệ thống sau khi Usecase xóa thẻ loại được thực thi:

+ Trường hợp thành công: Thông tin thẻ loại bị xóa khỏi hệ thống.

+ Trường hợp không thành công: Hệ thống thông báo lỗi và không xóa thẻ loại hệ thống.

- Điểm mở rộng: không có.

- Giao diện:

4.2.4.8 Đặc tả Usecase quản lý tác giả.

* **Đặc tả usecase thêm tác giả.**

- Tóm tắt: Admin, quản lý sử dụng usecase thêm tác giả để thêm thông tin của tác giả khi cửa hàng có sách có tác giả mới.

- Dòng sự kiện:

+ Dòng sự kiện chính:

- Admin, quản lý chọn chức năng quản lý tác giả.
- Hệ thống hiển thị ra giao diện quản lý tác giả.
- Admin, quản lý nhập thông tin như mã tác giả, tên tác giả và đc.
- Admin, quản lý chọn thêm tác giả.
- Hệ thống kiểm tra sự hợp lệ và thêm tác giả mới vào hệ thống.
- Kết thúc usecase.

+ Dòng sự kiện phụ:

- Dòng sự kiện thứ nhất:

- Admin, quản lý hủy thêm tác giả.
- Hệ thống trả về giao diện chính.
- Kết thúc usecase.

- Dòng sự kiện thứ hai:

- Thông tin tác giả không hợp lệ.
- Hệ thống hiển thị thông báo lỗi.
- Kết thúc usecase.

- Trạng thái hệ thống trước khi Usecase thêm tác giả được thực hiện: admin, quản lý phải đăng nhập vào hệ thống trước khi thực hiện chức năng này.

- Trạng thái hệ thống sau khi Usecase thêm tác giả được thực thi:

+ Trường hợp thành công: Thông tin tác giả mới được thêm vào hệ thống.

+ Trường hợp không thành công: Hệ thống thông báo lỗi và không thêm thông tin tác giả mới vào hệ thống.

- Điểm mở rộng: không có.

- Giao diện:

* **Đặc tả usecase sửa tác giả.**

- Tóm tắt: Admin, quản lý sử dụng usecase sửa tác giả để sửa thông tin của tác giả khi thông tin sai hoặc thay đổi.

- Dòng sự kiện:

+ Dòng sự kiện chính:

- Admin, quản lý chọn chức năng quản lý tác giả.
- Hệ thống hiển thị ra giao diện quản lý tác giả.
- Admin, quản lý chọn tác giả cần sửa và nhập thông tin cần sửa.
- Admin, quản lý chọn sửa tác giả.
- Hệ thống kiểm tra sự hợp lệ và cập nhật tác giả mới sửa vào hệ thống.
- Kết thúc usecase.

+ Dòng sự kiện phụ:

- Dòng sự kiện thứ nhất:

- Admin, quản lý hủy sửa tác giả.
- Hệ thống trả về giao diện chính.
- Kết thúc usecase.

- Dòng sự kiện thứ hai:

- Thông tin sửa tác giả không hợp lệ.
- Hệ thống hiển thị thông báo lỗi.
- Kết thúc usecase.

- Trạng thái hệ thống trước khi Usecase sửa tác giả được thực hiện: admin, quản lý phải đăng nhập vào hệ thống trước khi thực hiện chức năng này.

- Trạng thái hệ thống sau khi Usecase sửa tác giả được thực thi:

- + Trường hợp thành công: Thông tin tác giả mới sửa được cập nhật vào hệ thống.
- + Trường hợp không thành công: Hệ thống thông báo lỗi và không cập nhật thông tin tác giả mới sửa vào hệ thống.

- Điểm mở rộng: không có.

- Giao diện:

* **Đặc tả usecase xóa tác giả.**

- Tóm tắt: Admin, quản lý sử dụng usecase xóa tác giả để xóa thông tin của tác giả khi cần xóa trong trường hợp nào đó.

- Dòng sự kiện:

+ Dòng sự kiện chính:

- Admin, quản lý chọn chức năng quản lý tác giả.
- Hệ thống hiển thị ra giao diện quản lý tác giả.
- Admin, quản lý chọn tác giả cần xóa.
- Admin, quản lý chọn xóa tác giả.
- Hệ thống kiểm tra sự hợp lệ và xóa tác giả khỏi hệ thống.
- Kết thúc usecase.

+ Dòng sự kiện phụ:

- Dòng sự kiện thứ nhất:

- Admin, quản lý hủy xóa tác giả.
- Hệ thống trả về giao diện chính.
- Kết thúc usecase.

- Dòng sự kiện thứ hai:

- Xóa tác giả không hợp lệ.
- Hệ thống hiển thị thông báo lỗi.
- Kết thúc usecase.

- Trạng thái hệ thống trước khi Usecase xóa tác giả được thực hiện: admin, quản lý phải đăng nhập vào hệ thống trước khi thực hiện chức năng này.

- Trạng thái hệ thống sau khi Usecase xóa tác giả được thực thi:

- + Trường hợp thành công: Thông tin tác giả bị xóa khỏi hệ thống.
- + Trường hợp không thành công: Hệ thống thông báo lỗi và không xóa thông tin tác giả khỏi hệ thống.

- Điểm mở rộng: không có.

- Giao diện:

4.2.4.9 Đặc tả Usecase thống kê nhập.

*** Đặc tả usecase in báo cáo.**

- Tóm tắt: Admin, quản lý sử dụng usecase in báo cáo để in bản báo cáo định kỳ theo quy định của cửa hàng.

- Dòng sự kiện:

+ Dòng sự kiện chính:

- Admin, quản lý chọn chức năng thống kê nhập.
- Hệ thống hiển thị ra giao diện thống kê nhập.
- Admin, quản lý nhập chọn các lựa chọn cần thống kê như theo nxb, tháng, năm, thống kê phiếu nhập, thống kê chi tiết.
- Admin, quản lý chọn in báo cáo.
- Hệ thống kiểm tra sự hợp lệ và xuất bản báo cáo thống kê nhập sách.
- Kết thúc usecase.

+ Dòng sự kiện phụ:

- Dòng sự kiện thứ nhất:

- Admin, quản lý hủy in báo cáo.
- Hệ thống trả về giao diện chính.
- Kết thúc usecase.

- Dòng sự kiện thứ hai:

- Chọn các lựa chọn không hợp lệ.
- Hệ thống hiển thị thông báo lỗi.
- Kết thúc usecase.

- Trạng thái hệ thống trước khi Usecase in báo cáo được thực hiện: admin, quản lý phải đăng nhập vào hệ thống trước khi thực hiện chức năng này.
- Trạng thái hệ thống sau khi Usecase in báo cáo được thực thi:
 - + Trường hợp thành công: hệ thống xuất một bản báo cáo thông kê nhập.
 - + Trường hợp không thành công: Hệ thống thông báo lỗi và không xuất bản báo cáo phiếu nhập.
 - Điểm mở rộng: không có.
 - Giao diện:

4.2.4.10 Đặc tả Usecase thống kê bán.

*** Đặc tả usecase in báo cáo.**

- Tóm tắt: Admin, quản lý, nhân viên sử dụng usecase in báo cáo để in bản báo cáo định kỳ theo quy định của cửa hàng.

- Dòng sự kiện:

- + Dòng sự kiện chính:

- Admin, quản lý, nhân viên chọn chức năng thống kê bán sách.
- Hệ thống hiển thị ra giao diện thống kê bán sách.
- Admin, quản lý, nhân viên nhập chọn các lựa chọn cần thống kê như theo tháng, năm, thống kê hóa đơn, thống kê chi tiết.
- Admin, quản lý, nhân viên chọn in báo cáo.
- Hệ thống kiểm tra sự hợp lệ và xuất bản báo cáo thống kê bán sách.
- Kết thúc usecase.

- + Dòng sự kiện phụ:

- Dòng sự kiện thứ nhất:

- Admin, quản lý hủy in báo cáo.
- Hệ thống trở về giao diện chính.
- Kết thúc usecase.

- Dòng sự kiện thứ hai:

- Chọn các lựa chọn không hợp lệ.
- Hệ thống hiển thị thông báo lỗi.
- Kết thúc usecase.

- Trạng thái hệ thống trước khi Usecase in báo cáo được thực hiện: admin, quản lý phải đăng nhập vào hệ thống trước khi thực hiện chức năng này.

- Trạng thái hệ thống sau khi Usecase in báo cáo được thực thi:

- + Trường hợp thành công: hệ thống xuất một bản báo cáo thống kê bán sách.

+ Trường hợp không thành công: Hệ thống thông báo lỗi và không xuất bản báo cáo bán sách.

- Điểm mở rộng: không có.

- Giao diện:

4.2.4.10 Đặc tả Usecase thống kê số lượng tồn.

* Đặc tả usecase in báo cáo.

- Tóm tắt: Admin, quản lý, nhân viên sử dụng usecase in báo cáo để in bản báo cáo định kỳ theo quy định của cửa hàng.

- Dòng sự kiện:

+ Dòng sự kiện chính:

- Admin, quản lý, nhân viên chọn chức năng thống kê số lượng tồn.
- Hệ thống hiển thị ra giao diện thống kê số lượng tồn.
- Admin, quản lý, nhân viên nhập chọn các lựa chọn cần thống kê như theo tên tác giả, theo tên nhà xuất bản.
- Admin, quản lý, nhân viên chọn in báo cáo.
- Hệ thống kiểm tra sự hợp lệ và xuất bản báo cáo thống kê số lượng tồn.
- Kết thúc usecase.

+ Dòng sự kiện phụ:

- Dòng sự kiện thứ nhất:

- Admin, quản lý, nhân viên hủy in báo cáo.
- Hệ thống trả về giao diện chính.
- Kết thúc usecase.

- Dòng sự kiện thứ hai:

- Chọn các lựa chọn không hợp lệ.
- Hệ thống hiển thị thông báo lỗi.
- Kết thúc usecase.

- Trạng thái hệ thống trước khi UseCase in báo cáo được thực hiện: admin, quản lý, nhân viên phải đăng nhập vào hệ thống trước khi thực hiện chức năng này.

- Trạng thái hệ thống sau khi UseCase in báo cáo được thực thi:

+ Trường hợp thành công: hệ thống xuất một bản báo cáo thống kê số lượng tồn.

+ Trường hợp không thành công: Hệ thống thông báo lỗi và không xuất bản báo cáo số lượng tồn.

- Điểm mở rộng: không có.

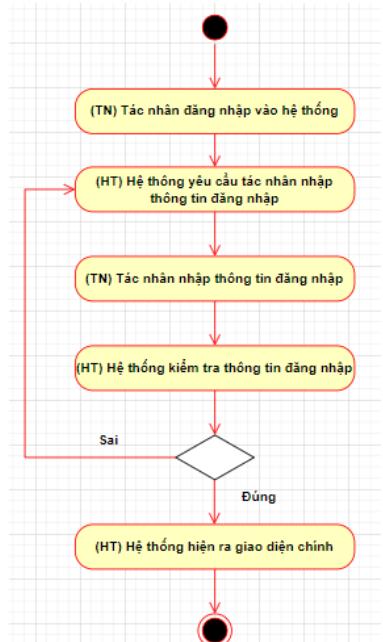
- Giao diện:

4.3 Mô hình hóa khái niệm

4.3.1 Sơ đồ hoạt động

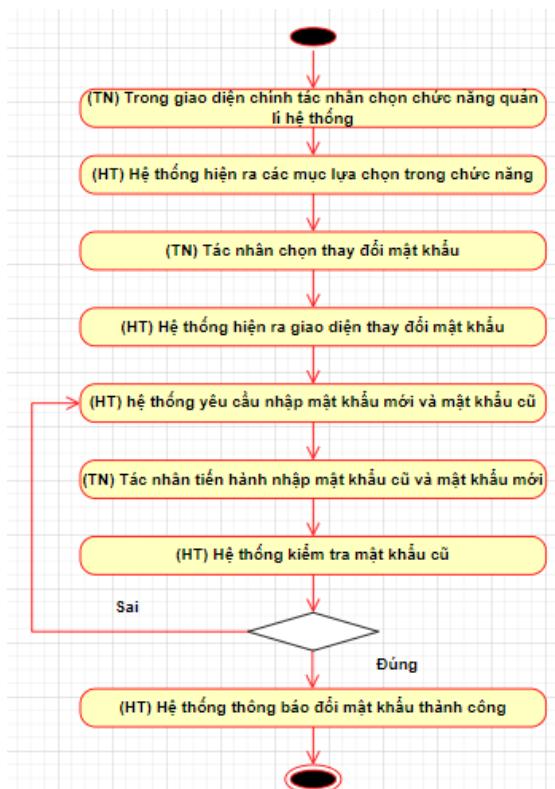
4.3.1.1 Chức năng quản lý hệ thống

- Tác nhân đăng nhập



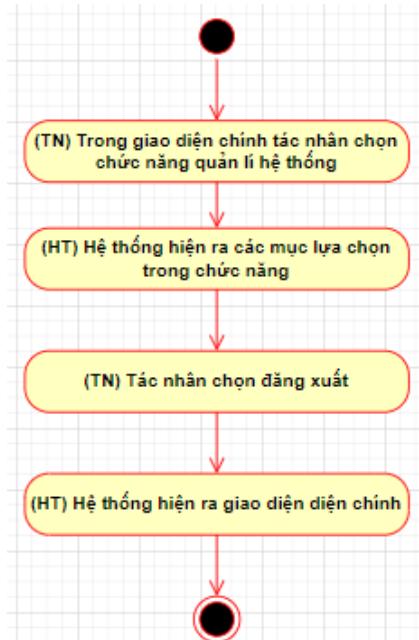
Hình 4.9: Sơ đồ hoạt động tác nhân đăng nhập

- Tác nhân thay đổi mật khẩu



Hình 4.10: Sơ đồ hoạt động tác nhân thay đổi mật khẩu

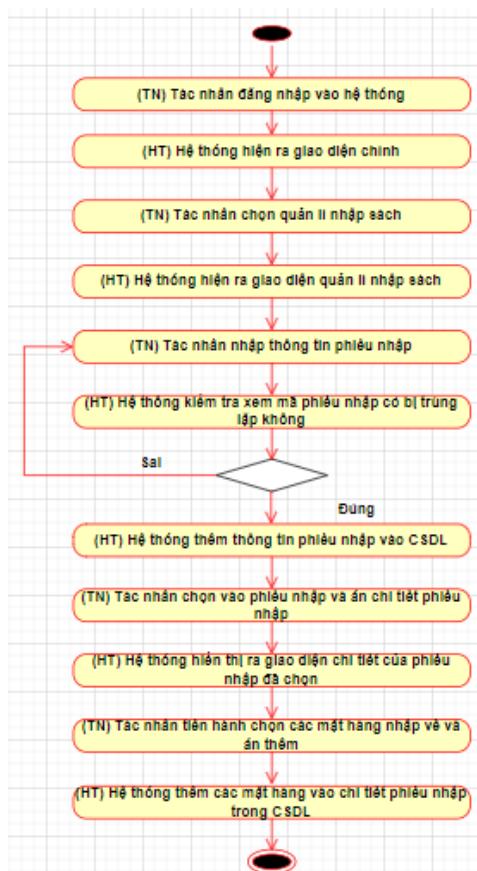
- Tác nhân đăng nhập



Hình 4.11: Sơ đồ hoạt động tác nhân thay đăng xuất

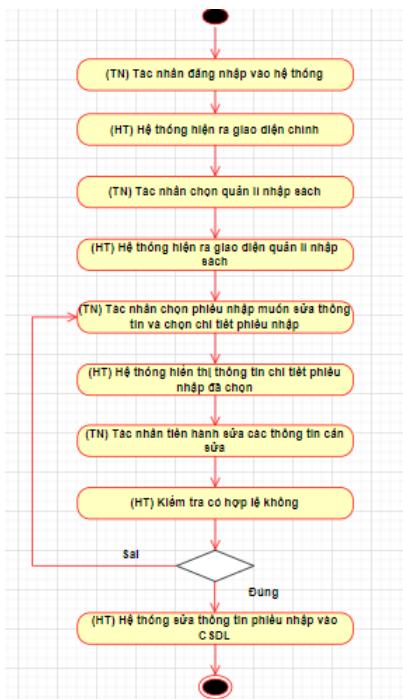
4.3.1.2 Chức năng quản lý nhập sách

- Tác nhân thêm phiếu nhập



Hình 4.12: Sơ đồ hoạt động tác nhân thêm phiếu nhập

- Tác nhân sửa thông tin phiếu nhập



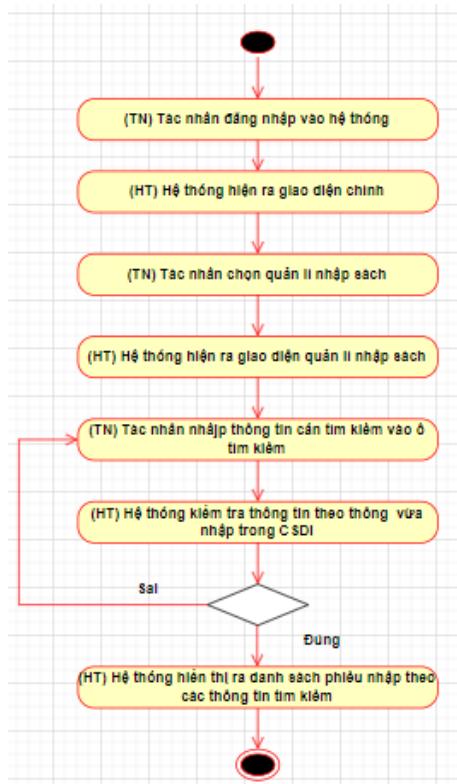
Hình 4.13: Sơ đồ hoạt động tác nhân sửa thông tin phiếu nhập

- Tác nhân xóa phiếu nhập



Hình 4.14: Sơ đồ hoạt động tác nhân xóa thông tin phiếu nhập

- Tác nhân tìm kiếm phiếu nhập



Hình 4.15: Sơ đồ hoạt động tác nhân tìm phiếu nhập

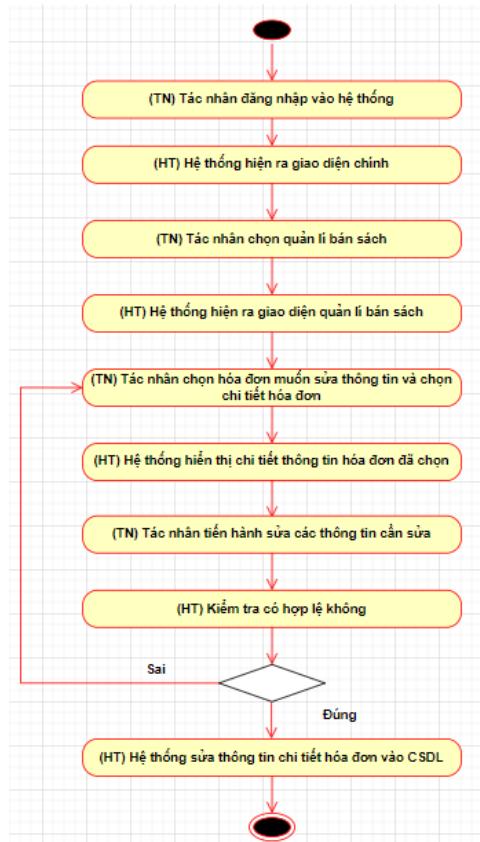
4.3.1.3 Chức năng quản lý bán sách

- Tác nhân thêm hóa đơn



Hình 4.16: Sơ đồ hoạt động tác nhân thêm hóa đơn

- Tác nhân sửa hóa đơn



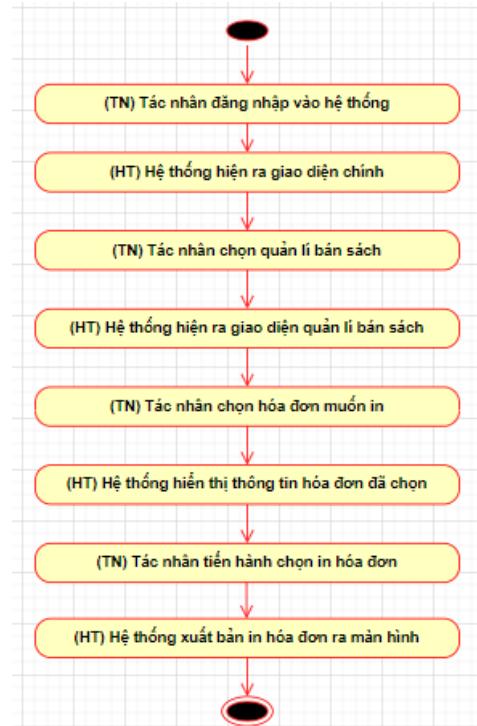
Hình 4.17: Sơ đồ hoạt động tác nhân sửa hóa đơn

- Tác nhân xóa hóa đơn



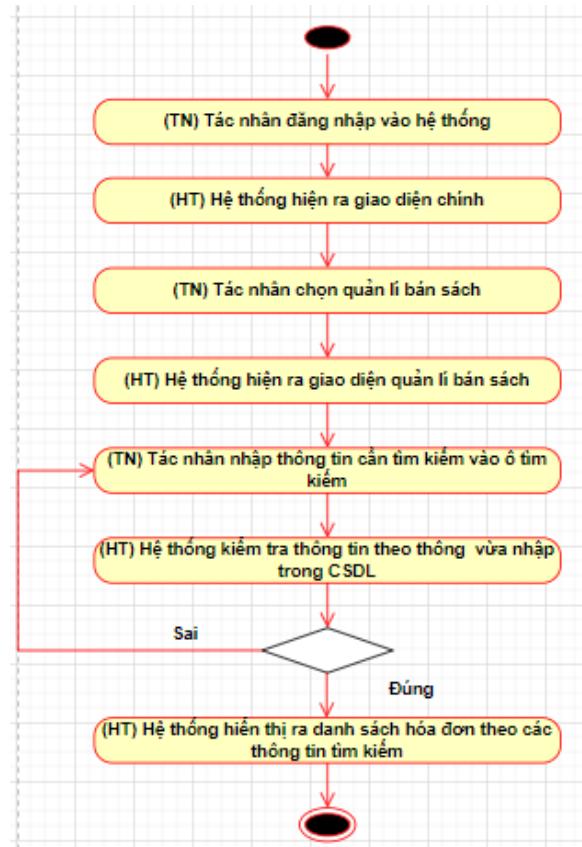
Hình 4.18: Sơ đồ hoạt động tác nhân xóa hóa đơn

- Tác nhân in hóa đơn



Hình 4.19: Sơ đồ hoạt động tác nhân in hóa đơn

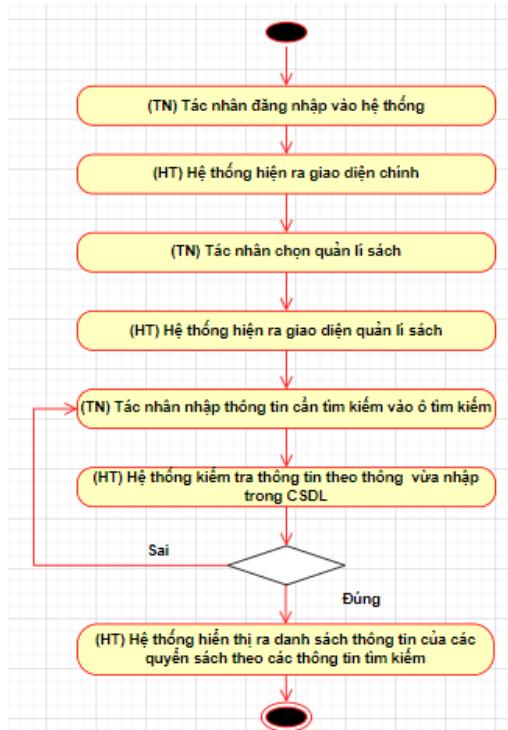
- Tác nhân tìm kiếm hóa đơn



Hình 4.20: Sơ đồ hoạt động tác nhân tìm kiếm hóa đơn

4.3.1.4 Chức năng quản lý kho sách

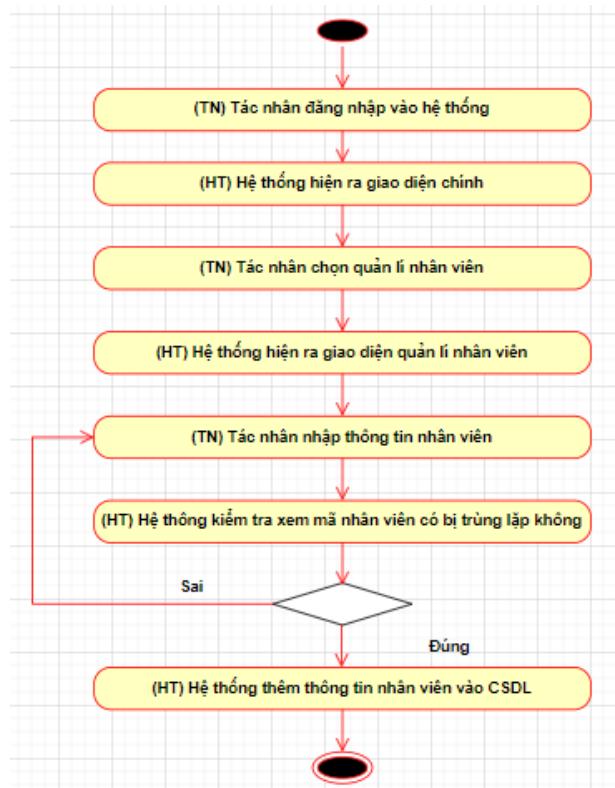
- Tác nhân tìm kiếm sách



Hình 4.21: Sơ đồ hoạt động tác nhân tìm kiếm sách

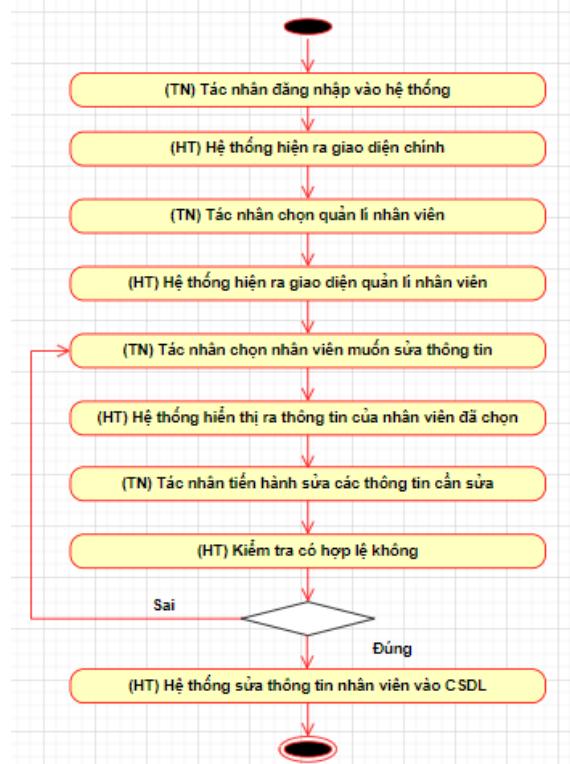
4.3.1.5 Chức năng quản lý nhân viên

- Tác nhân thêm nhân viên



Hình 4.22: Sơ đồ hoạt động tác nhân thêm nhân viên

- Tác nhân sửa nhân viên



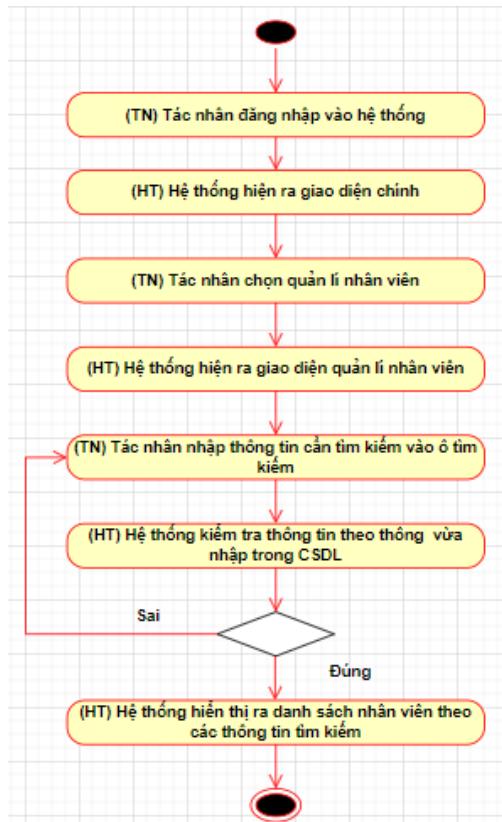
Hình 4.23: Sơ đồ hoạt động tác nhân sửa nhân viên

- Tác nhân xóa nhân viên



Hình 4.24: Sơ đồ hoạt động tác nhân xóa nhân viên

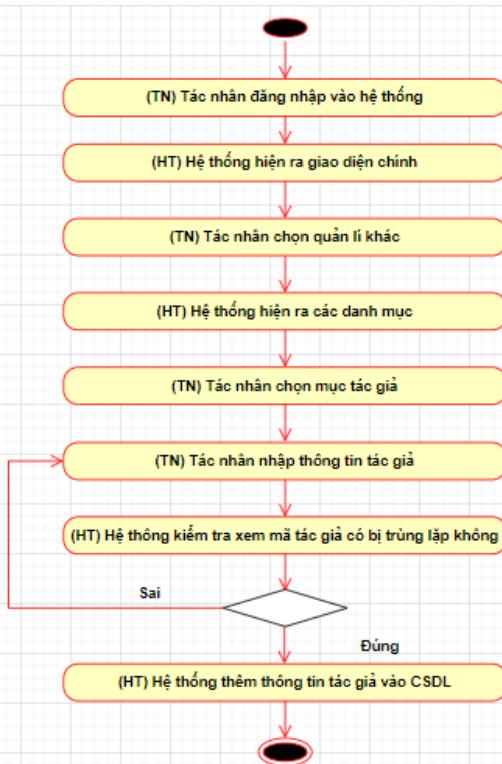
- Tác nhân tìm kiếm nhân viên



Hình 4.25: Sơ đồ hoạt động tác nhân tìm kiếm nhân viên

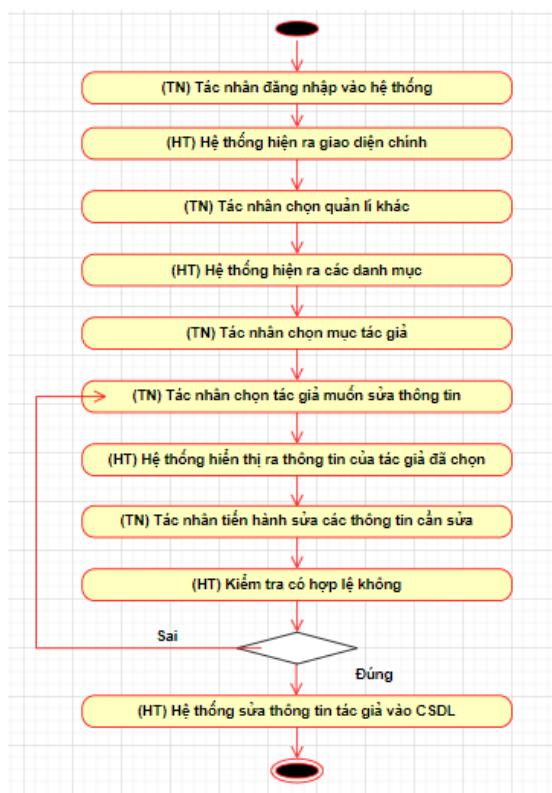
4.3.1.6 Chức năng quản lý tác giả

- Tác nhân thêm tác giả



Hình 4.26: Sơ đồ hoạt động tác nhân thêm tác giả

- Tác nhân sửa tác giả



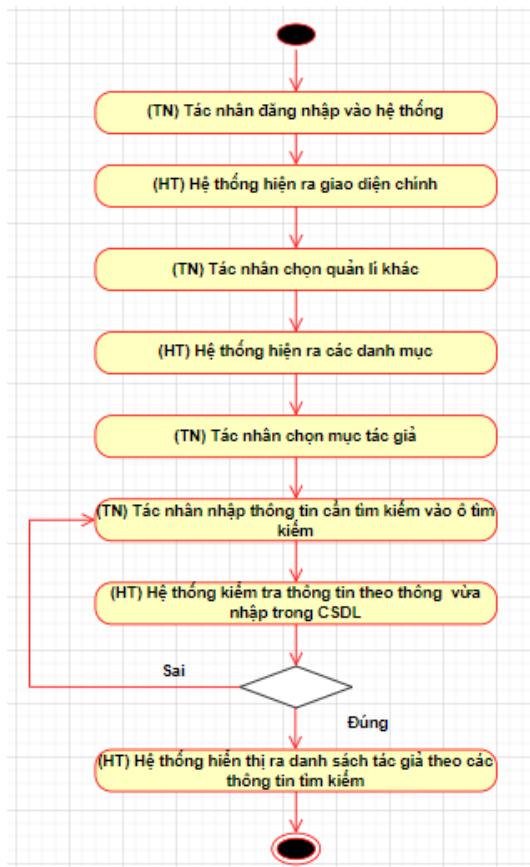
Hình 4.27: Sơ đồ hoạt động tác nhân sửa tác giả

- Tác nhân xóa tác giả



Hình 4.28: Sơ đồ hoạt động tác nhân xóa tác giả

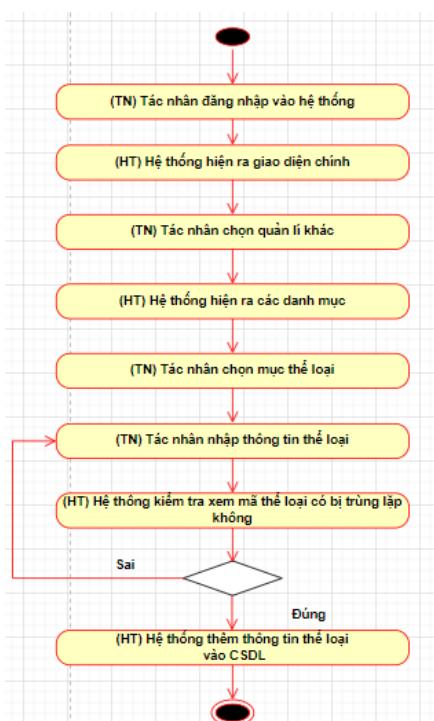
- Tác nhân tìm kiếm tác giả



Hình 4.29: Sơ đồ hoạt động tác nhân tìm kiếm tác giả

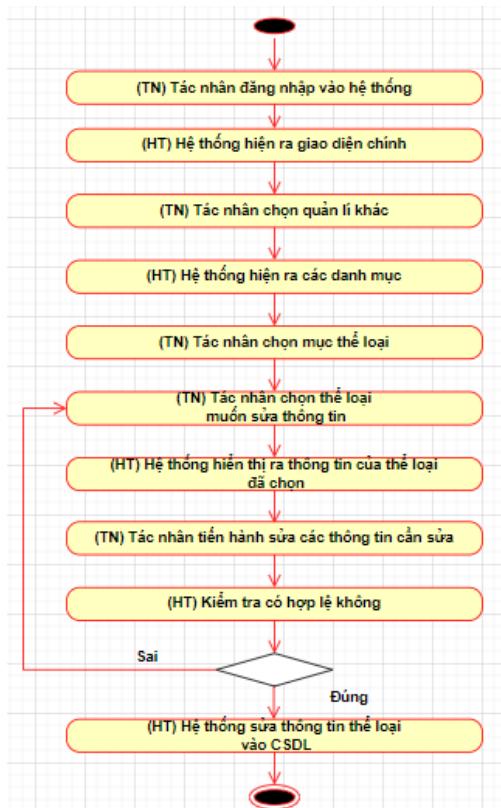
4.3.1.7 Chức năng quản lý thể loại

- Tác nhân thêm thể loại



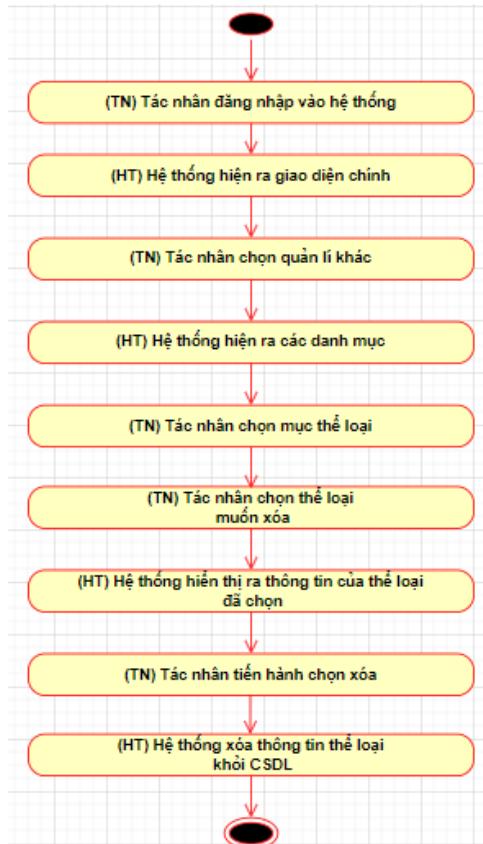
Hình 4.30: Sơ đồ hoạt động tác nhân thêm thể loại

- Tác nhân sửa thẻ loại



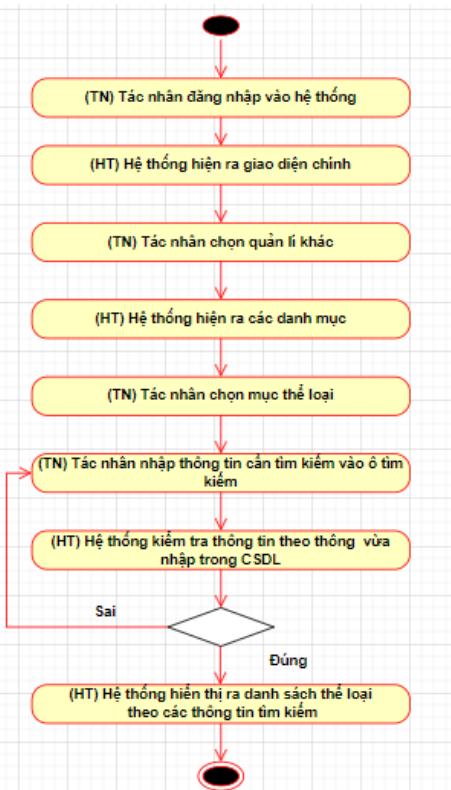
Hình 4.31: Sơ đồ hoạt động tác nhân sửa thẻ loại

- Tác nhân xóa thẻ loại



Hình 4.32: Sơ đồ hoạt động tác nhân xóa thẻ loại

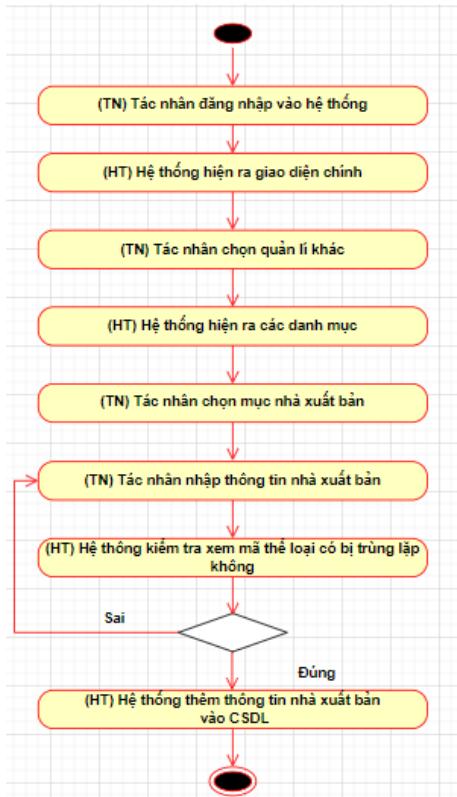
- Tác nhân tìm kiếm thẻ loại



Hình 4.33: Sơ đồ hoạt động tác nhân tìm kiếm thẻ loại

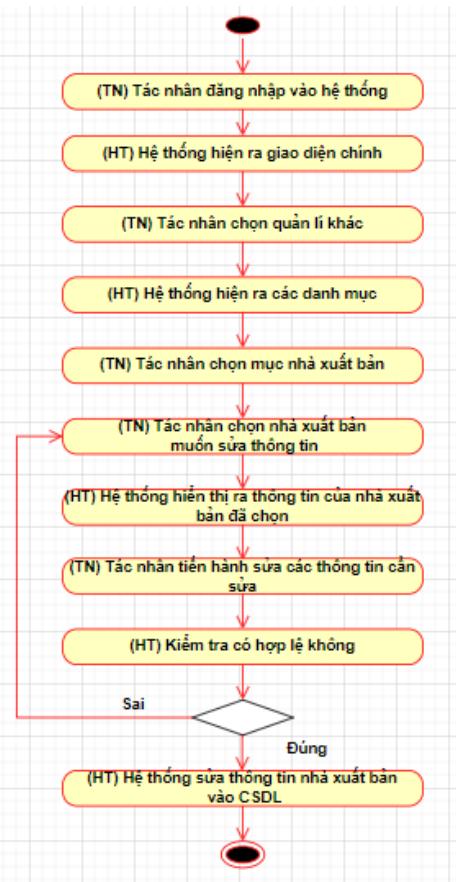
4.3.1.8 Chức năng quản lý nhà xuất bản

- Tác nhân thêm nhà xuất bản



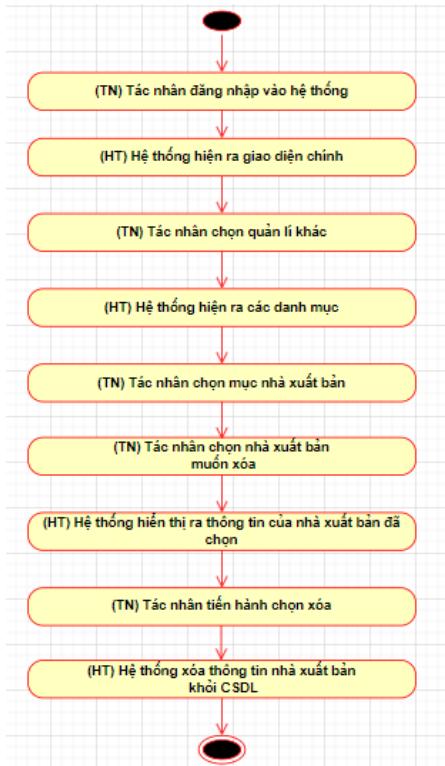
Hình 4.34: Sơ đồ hoạt động tác nhân thêm nhà xuất bản

- Tác nhân sửa nhà xuất bản



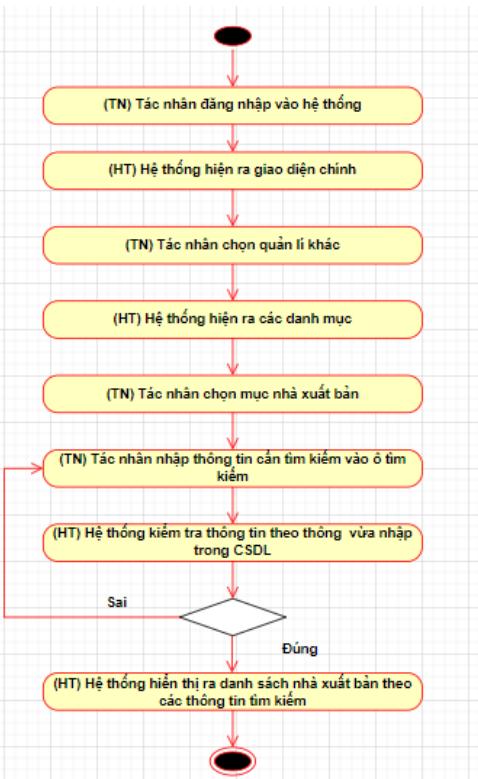
Hình 4.35: Sơ đồ hoạt động tác nhân sửa nhà xuất bản

- Tác nhân xóa nhà xuất bản



Hình 4.36: Sơ đồ hoạt động tác nhân xóa nhà xuất bản

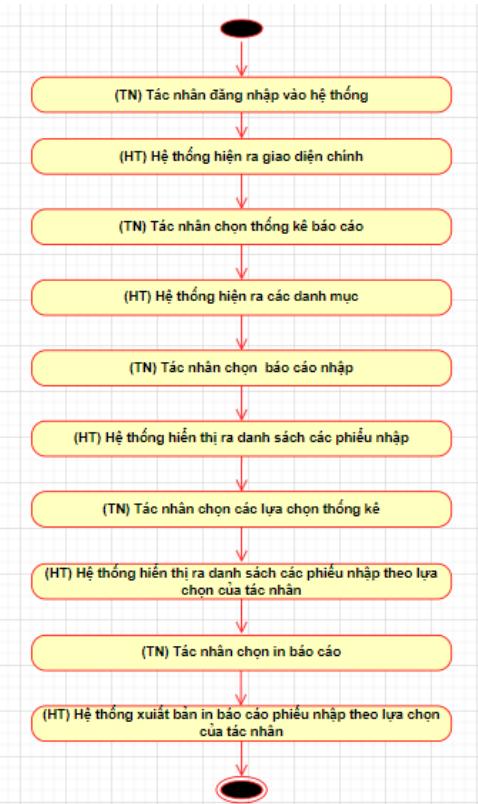
- Tác nhân tìm kiếm nhà xuất bản



Hình 4.37: Sơ đồ hoạt động tác nhân tìm kiếm nhà xuất bản

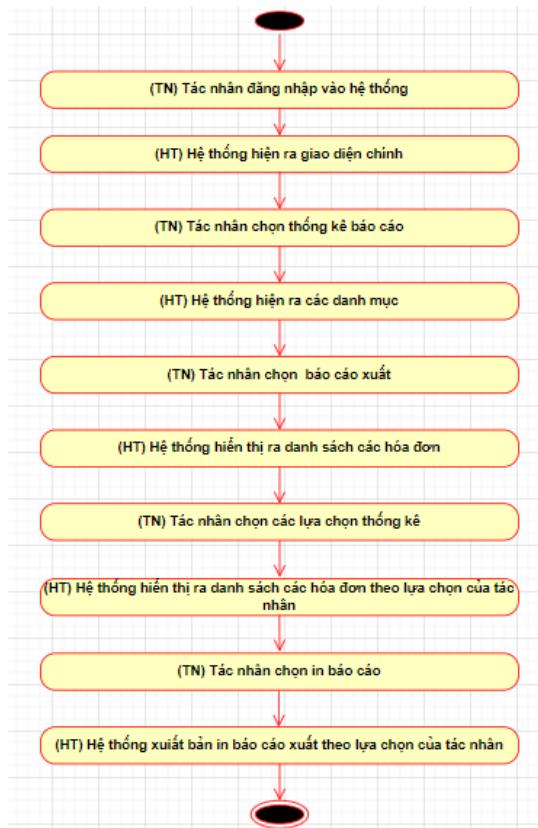
4.3.1.9 Chức năng thống kê báo cáo

- Tác nhân thống kê nhập



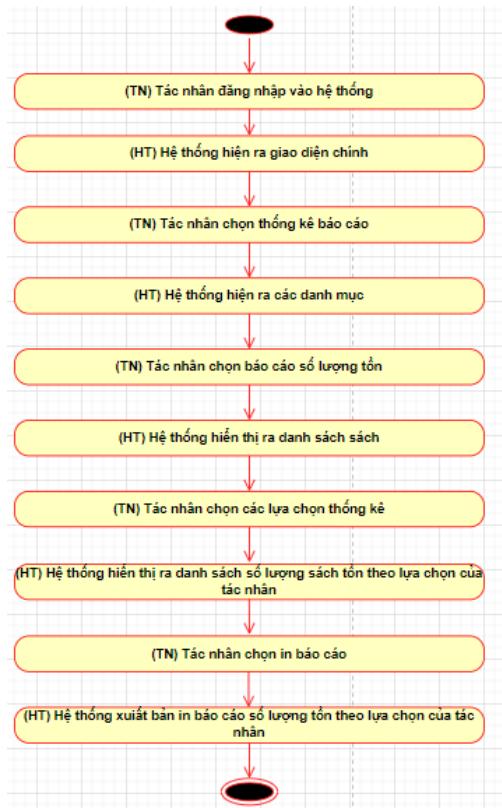
Hình 4.38: Sơ đồ hoạt động tác nhân thống kê nhập

- Tác nhân thông kê bán



Hình 4.39: Sơ đồ hoạt động tác nhân thông kê bán sách

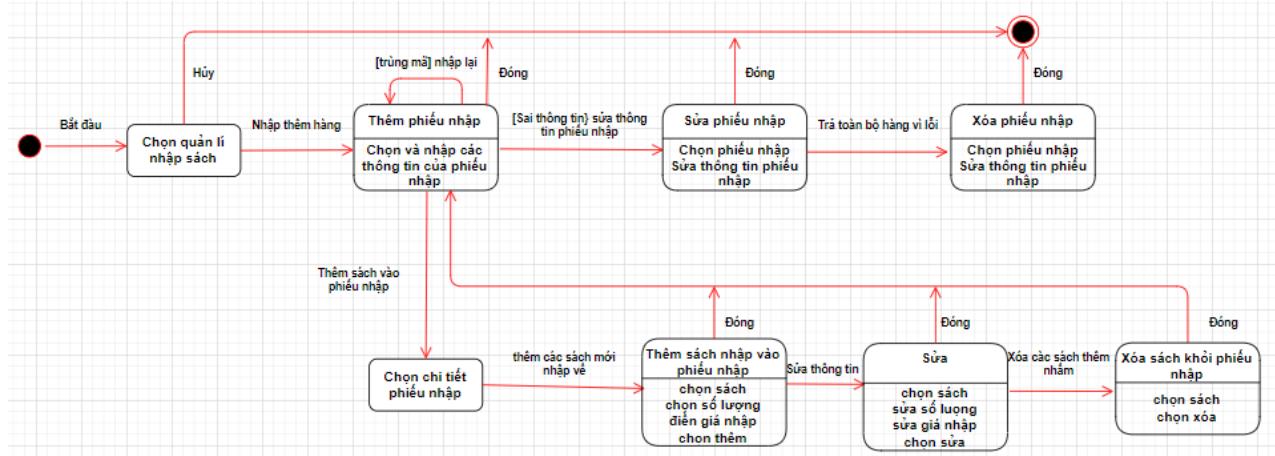
- Tác nhân thông kê số lượng tồn



Hình 4.40: Sơ đồ hoạt động tác nhân thông kê số lượng tồn

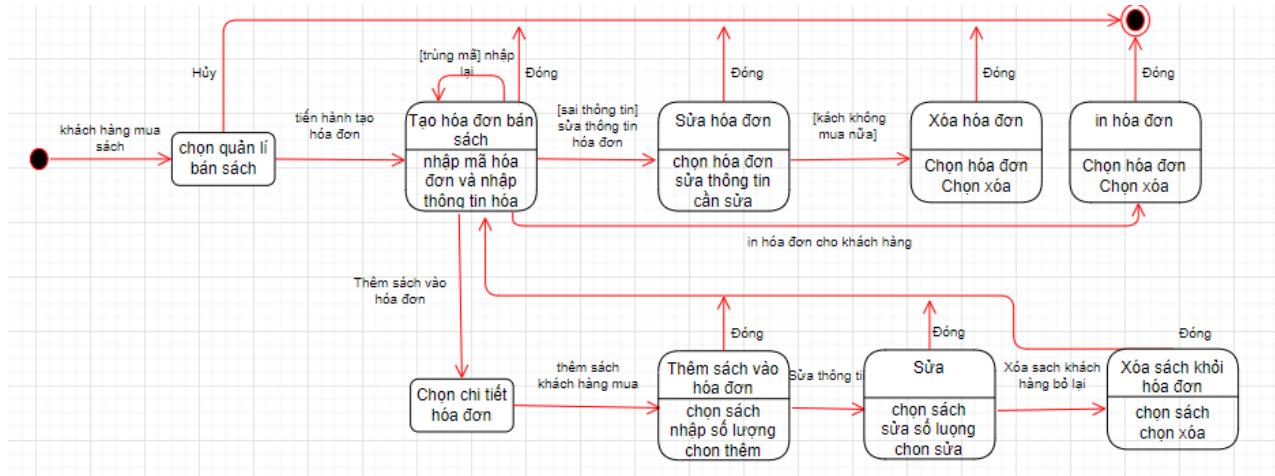
4.3.2 Sơ đồ trạng thái

4.3.2.1 Sơ đồ trạng thái nhập sách



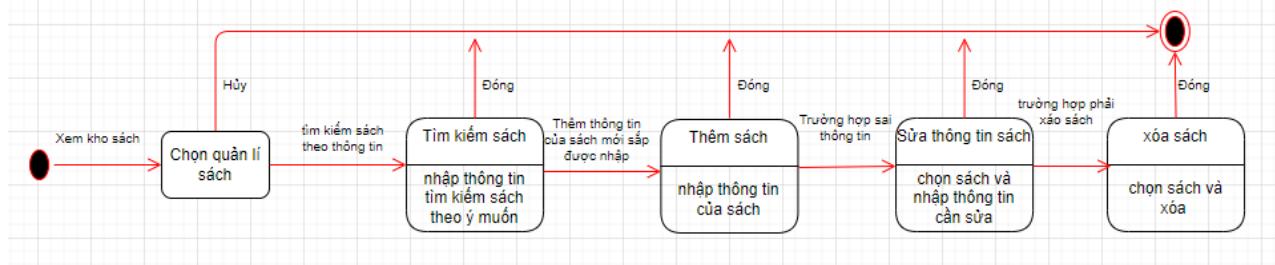
Hình 4.41: Sơ đồ trạng thái chức năng nhập sách

4.3.2.2 Sơ đồ trạng thái bán sách



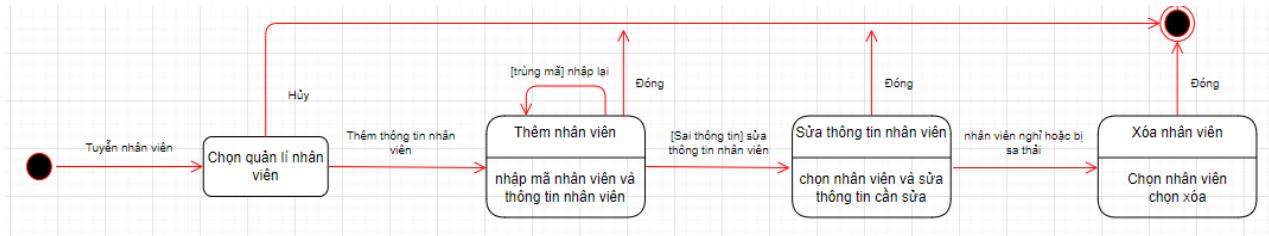
Hình 4.42: Sơ đồ trạng thái chức năng bán sách

4.3.2.3 Sơ đồ trạng thái quản lý kho sách



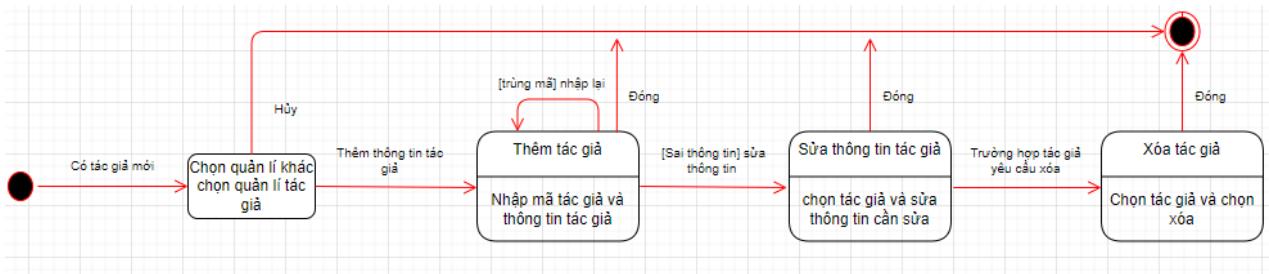
Hình 4.43: Sơ đồ trạng thái chức năng quản lý kho sách

4.3.2.4 Sơ đồ trạng thái quản lý nhân viên



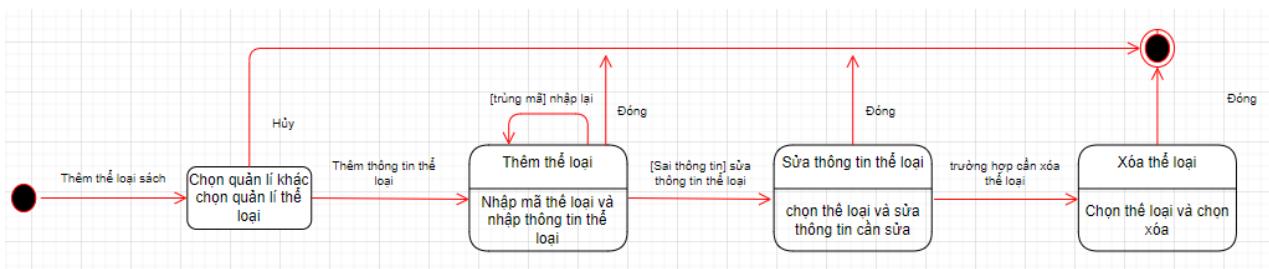
Hình 4.44: Sơ đồ trạng thái chức năng quản lý nhân viên

4.3.2.5 Sơ đồ trạng thái quản lý tác giả



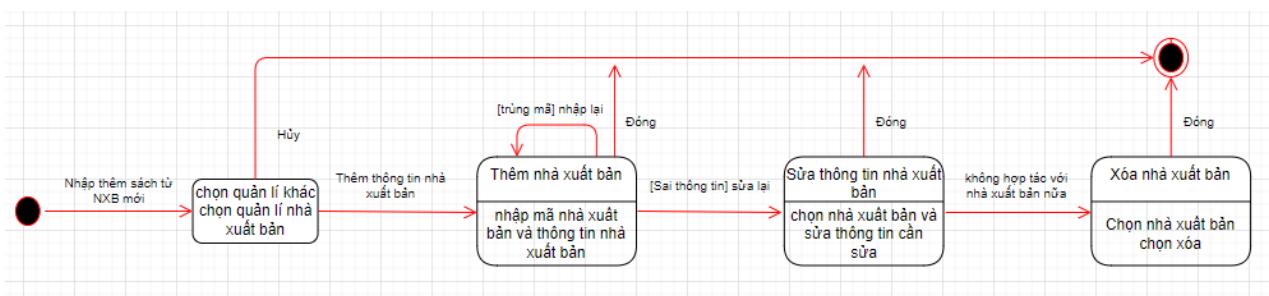
Hình 4.45: Sơ đồ trạng thái chức năng quản lý tác giả

4.3.2.6 Sơ đồ trạng thái quản lý thể loại



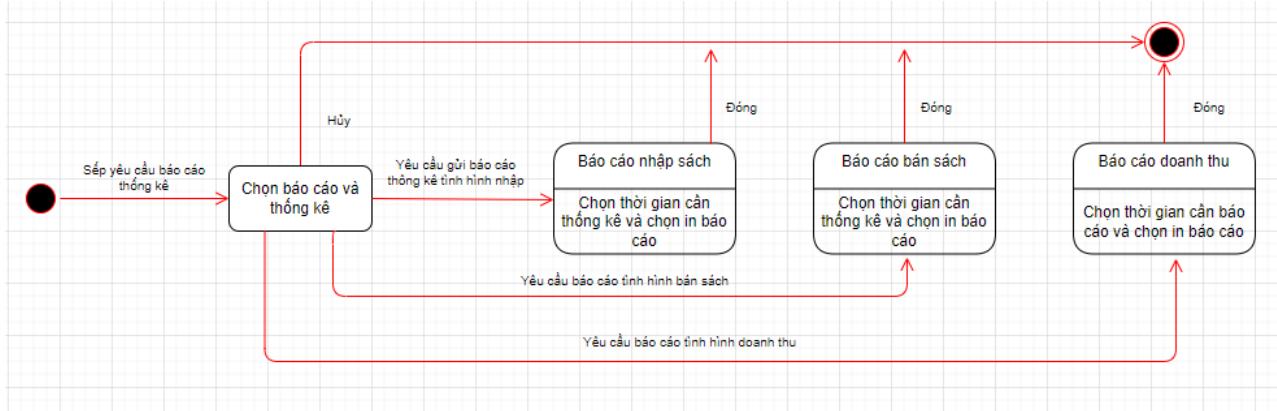
Hình 4.46: Sơ đồ trạng thái chức năng quản lý thể loại

4.3.2.7 Sơ đồ trạng thái quản lý nhà xuất bản



Hình 4.47: Sơ đồ trạng thái chức năng quản lý nhà xuất bản

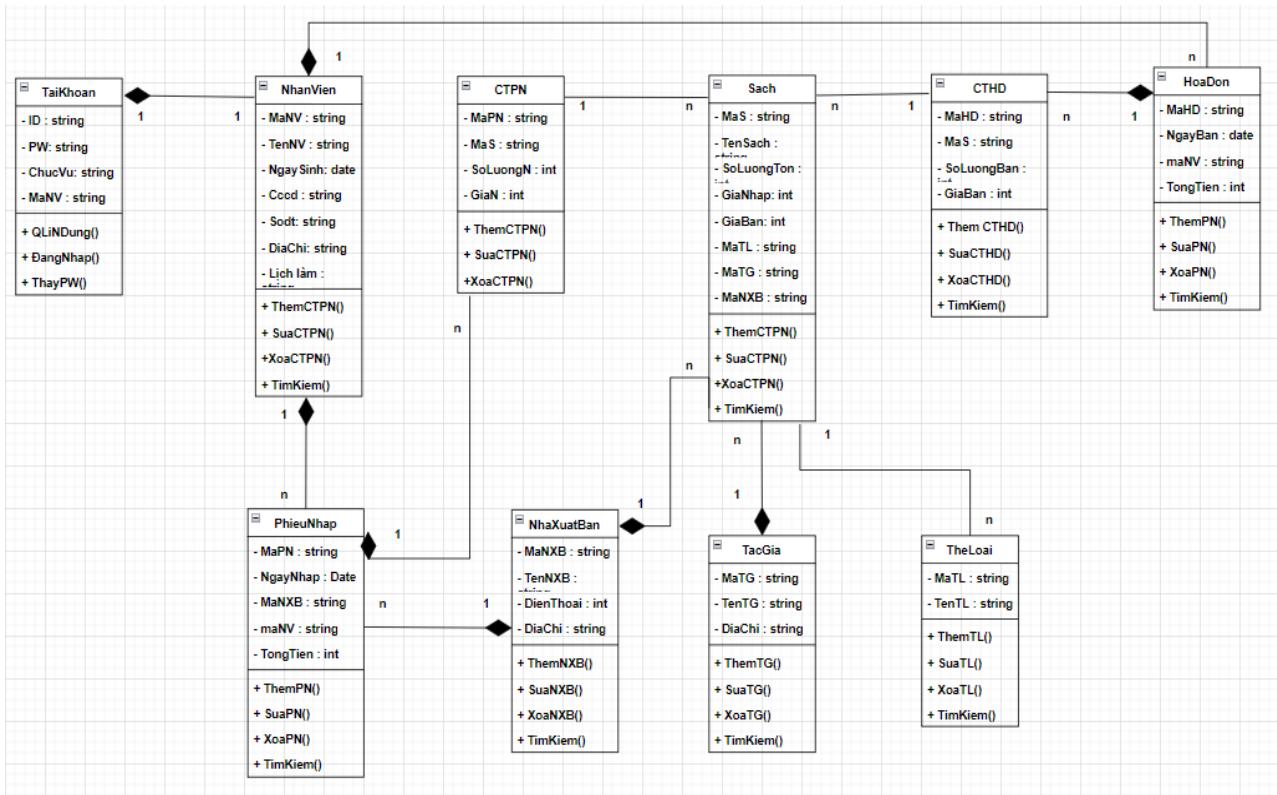
4.3.2.8 Sơ đồ trạng thái thông kê báo cáo



Hình 4.48: Sơ đồ trạng thái chức năng thống kê báo cáo

4.3.3 Sơ đồ lớp

4.3.3.1 Sơ đồ lớp tổng quát



Hình 4.49: Sơ đồ lớp tổng quát

4.3.4 Đặc tả các lớp

Tbl4.1 :Bảng đặc tả lớp tài khoản

	Định nghĩa	Chức các thông tin và phương thức của tài khoản
	Thuộc tính	ID: string : là tài khoản đăng nhập. PW: string : là mật khẩu ChucVu: string : là chức vụ của các tác nhân trong hệ thống. MaNV : string : là nhân viên được cấp tài khoản có id.
	Phương thức	QLiNDung(): Thực hiện việc phân quyền và cấp tài khoản cho nhân viên. DangNhap(): Dùng để đăng nhập vào hệ thống. ThayPW(): Dùng để thay đổi mật khẩu của người dùng.

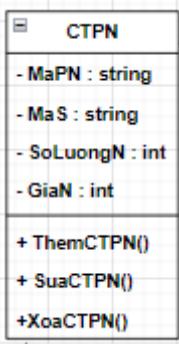
Tbl4.2 :Bảng đặc tả lớp NhanVien

	Định nghĩa	Chứa các thuộc tính và phương thức của nhân viên
	Thuộc tính	MaNV: string: mã nhân viên. TenNV: string: tên nhân viên. NgaySinh: string: ngày sinh. Cccd: string: căn cước công dân. Sodt: string: số điện thoại. DiaChi: string: địa chỉ LichLam: string: lịch làm
	Phương thức	ThemNV(): Thêm một nhân viên. SuaNV(): Sửa thông tin nhân viên. XoaNV(): Xóa một nhân viên. TimKiem(): Tìm kiếm nhân viên

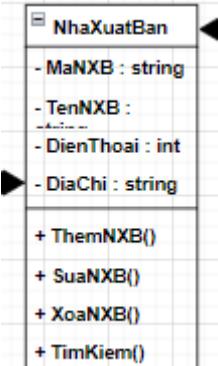
Tbl4.3 :Bảng đặc tả lớp PhieuNhap

	Định nghĩa	Chức các thuộc tính và phương thức của phiếu nhập sách
	Thuộc tính	<p>MaPN: string: mã phiếu nhập.</p> <p>NgayNhap: Date: ngày nhập.</p> <p>MaNXB: string: mã nhà xuất bản.</p> <p>maNV: string: mã nhân viên lập phiếu.</p>
	Phương thức	<p>ThemPN(): Thêm một phiếu nhập.</p> <p>SuaPN(): Sửa một phiếu nhập.</p> <p>XoaPN(): Xóa một phiếu nhập.</p> <p>TimKiem(): Tìm kiếm phiếu nhập.</p>

Tbl4.4 :Bảng đặc tả lớp CTPN

	Định nghĩa	Chứa các thuộc tính và phương thức của chi tiết phiếu nhập
	Thuộc tính	<p>MaPN: string: mã phiếu nhập.</p> <p>MaS: string: mã sách.</p> <p>SoLuongN: int: số lượng nhập.</p> <p>GiaN: int: thành tiền</p>
	Phương thức	<p>ThemCTPN(): thêm một mặt hàng vào CTPN.</p> <p>SuaCTPN(): sửa mặt hàng trong CTPN.</p> <p>XoaCTPN(): xóa hàng trong CTPN.</p>

Tbl4.5 :Bảng đặc tả lớp NhaXuatBan

	Định nghĩa	Chức các thuộc tính và phương thức của nhà xuất bản
	Thuộc tính	<p>MaNXB: string: mã nhà xuất bản.</p> <p>TenNXB: string: tên nhà xuất bản.</p> <p>DienThoai: string: số điện thoại.</p> <p>DiaChi: string: địa chỉ của nhà xuất bản.</p>
	Phương thức	<p>ThemNXB(): thêm một nhà xuất bản.</p> <p>SuaNXB(): sửa thông tin của một nhà xuất bản.</p> <p>XoaNXB(): xóa một nhà xuất bản.</p> <p>TimKiem(): Tìm kiếm nhà xuất bản.</p>

Tbl4.6 :Bảng đặc tả lớp TacGia

	Định nghĩa	Chức các thuộc tính và phương thức của tác giả
	Thuộc tính	<p>MaTG: string : mã tác giả.</p> <p>TenTG: string: tên tác giả.</p> <p>DiaChi: string: địa chỉ của tác giả.</p>
	Phương thức	<p>ThemTG(): thêm một tác giả.</p> <p>SuaTG(): sửa thông tin của một tác giả.</p> <p>XoaTG(): xóa một tác giả.</p> <p>TimKiem(): tìm kiếm tác giả</p>

Tbl4.7 :Bảng đặc tả lớp TheLoai

<pre> classDiagram class TheLoai { -MaTL : string -TenTL : string +ThemTL() +SuaTL() +XoaTL() +TimKiem() } </pre>	Định nghĩa	Chứa các thuộc tính và phương thức của thể loại
	Thuộc tính	MaTL: string: mã thể loại TenTL: string: tên thể loại
	Phương thức	ThemTL(): thêm một thể loại. SuaTL(): sửa thông tin của một thể loại. XoaTL(): Xóa một thể loại. Timkiem(): tìm kiếm thể loại

Tbl4.8 :Bảng đặc tả lớp HoaDon

<pre> classDiagram class HoaDon { -MaHD : string -NgayBan : date -maNV : string -TongTien : int +ThemPN() +SuaPN() +XoaPN() +TimKiem() } </pre>	Định nghĩa	Chức các thuộc tính và phương thức của hóa đơn
	Thuộc tính	MaHD: string: mã hóa đơn NgayBan: date: ngày bán. maNV: string: mã nhân viên. TongTien: int: tổng tiền của hóa đơn.
	Phương thức	ThemHD(): Thêm một hóa đơn. SuaHD(): Sửa thông tin của một hóa đơn. XoaHD(): Xóa một hóa đơn. TimKiem(): tìm kiếm hóa đơn.

Tbl4.9 :Bảng đặc tả lớp CTHD

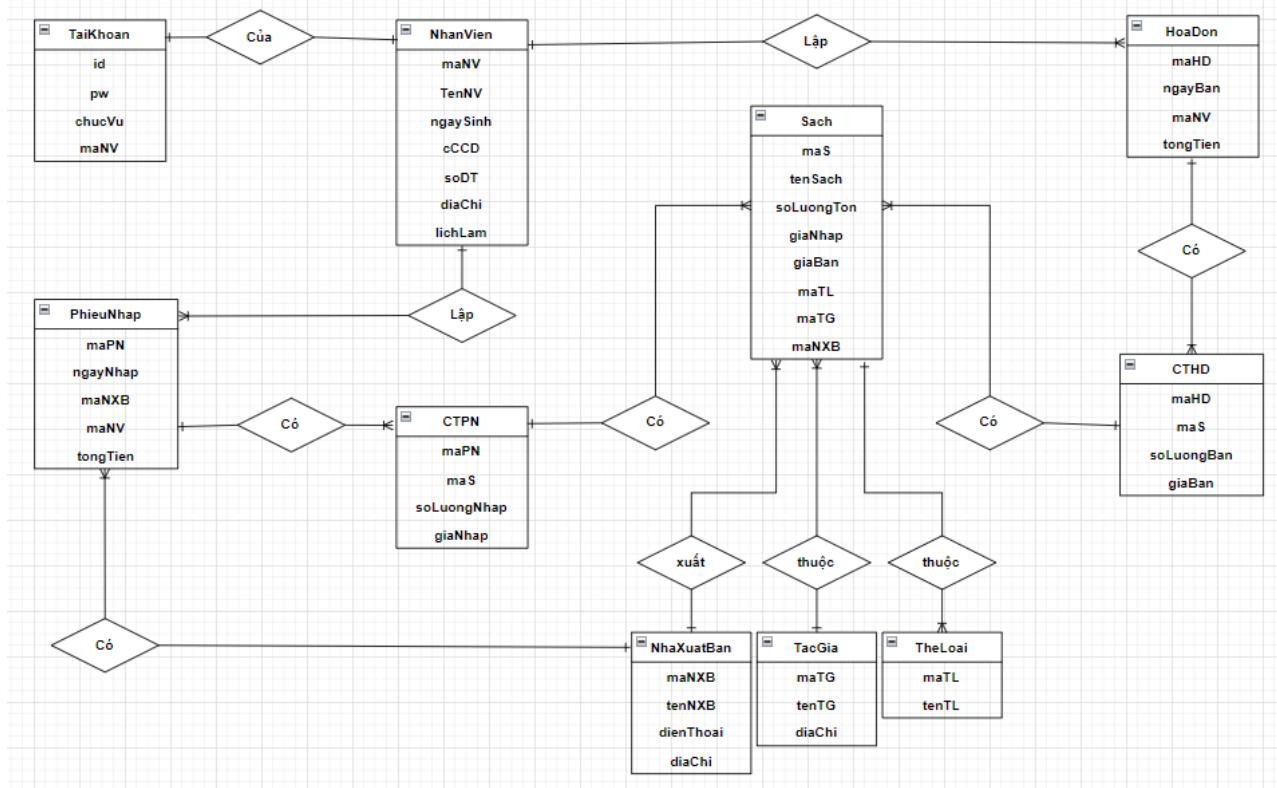
<pre> classDiagram class CTHD { -MaHD : string -MaS : string -SoLuongBan : int -GiaBan : int +ThemCTHD() +SuaCTHD() +XoaCTHD() +TimKiem() } </pre>	Định nghĩa	Chứa thuộc tính và phương thức của chi tiết hóa đơn
	Thuộc tính	MaHD: string: mã hóa đơn MaS: string: mã sách. SoLuongBan: int : số lượng bán. GiaBan: int: thành tiền
	Phương thức	ThemCTHD(): Thêm hàng vào chi tiết hóa đơn. XoaCTHD(): Xóa một mặt hàng trong chi tiết hóa đơn

Tbl4.10 :Bảng đặc tả lớp Sach

	Định nghĩa	Chức thuộc tính và phương thức của sách
<pre> class Sach { - MaS : string - TenSach : - SoLuongTon : - GiaNhap: int - GiaBan: int - MaTL : string - MaTG : string - MaNXB : string + ThemCTPN() + SuaCTPN() + XoaCTPN() + TimKiem() } </pre>	Thuộc tính	<p>MaS: string: mã sách. TenSach: string: tên sách. SoLuongTon: int: số lượng tồn của sách. GiaNhap: int: giá nhập. GiaBan: int: giá bán. MaTL: string: mã thể loại. MaTG: string: mã tác giả. MaNXB: string: mã nhà xuất bản</p>
	Phương thức	<p>ThemS(): Thêm thông tin của một quyển sách. SuaS(): Sửa thông tin của một quyển sách. XoaS(): xóa một quyển sách. TimKiem(): Tìm kiếm sách</p>

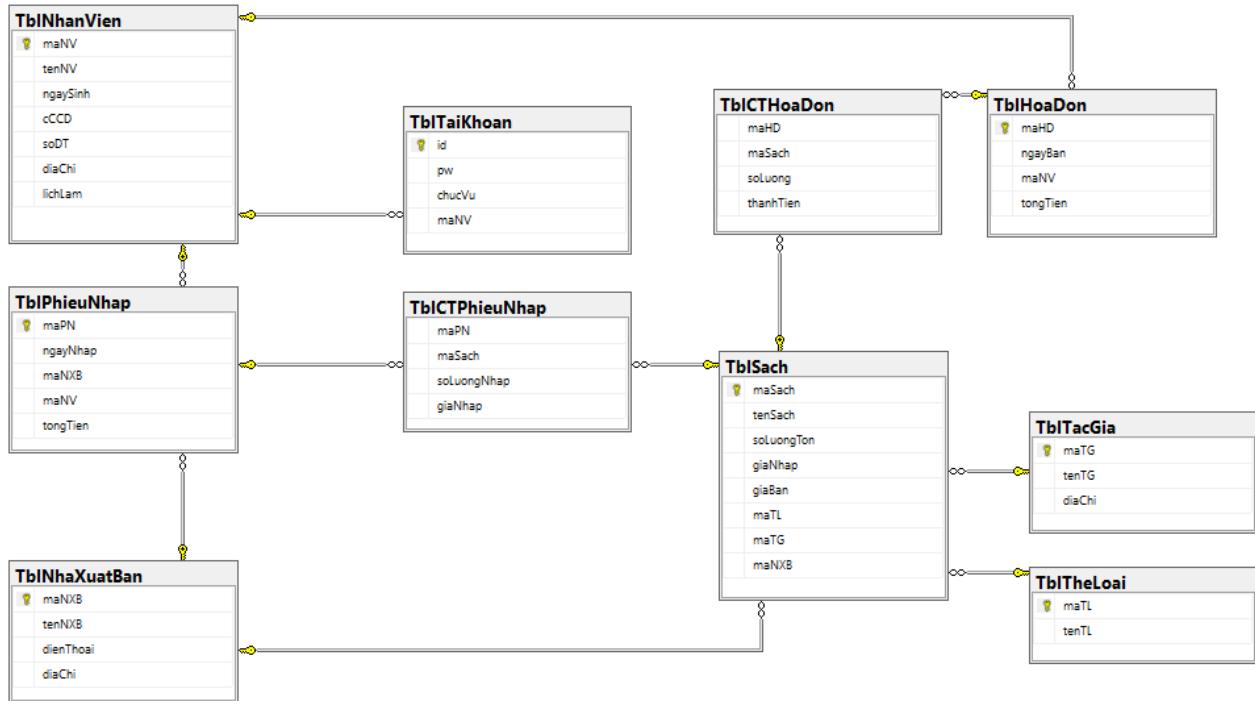
4.4 Mô hình cơ sở dữ liệu

4.4.1 Lược đồ cơ sở dữ liệu quản hệ



Hình 4.50: Lược đồ cơ sở dữ liệu quản hệ

4.4.2 Mô hình cơ sở dữ liệu



Hình 4.51: Mô hình cơ sở dữ liệu

4.4.3 Các bảng cơ sở dữ liệu vật lý

Tbl4.11: Bảng TblTaiKhoan

STT	Tên thuộc tính	Kiểu độ rộng	Ràng buộc	Khóa chính/Khóa phụ (PK/FK)	Mô tả
1	id	char(15)	NotNull	PK	Tên tài khoản
2	pw	nvarchar(15)	NotNull		Mật khẩu
3	chucVu	nvarchar(20)			Chức vụ
4	maNV	char(15)	NotNull	FK	Mã nhân viên

Tbl4.12: Bảng TblNhanVien

STT	Tên thuộc tính	Kiểu độ rộng	Ràng buộc	Khóa chính/Khóa phụ (PK/FK)	Mô tả
1	maNV	char(15)	NotNull	PK	Mã nhân viên
2	tenNV	nvarchar(30)			Tên nhân viên
3	ngaySinh	date			Ngày sinh
4	cCCD	char(15)			Căn cước công dân
5	soDT	char(15)			Số điện thoại
6	diaChi	nvarchar(100)			Địa chỉ
7	lichLam	nvarchar(50)			Lịch làm

Tbl4.13: Bảng TblCTHoaDon

STT	Tên thuộc tính	Kiểu độ rộng	Ràng buộc	Khóa chính/Khóa phụ (PK/FK)	Mô tả
1	maHD	char(15)	NotNull	FK	Mã hóa đơn
2	maSach	char(15)	NotNull	FK	Mã sách
3	soLuong	int			Số lượng
4	thanhTien	bigint			Thành tiền

Tbl4.14: Bảng TblCTPhieuNhap

STT	Tên thuộc tính	Kiểu độ rộng	Ràng buộc	Khóa chính/Khóa phụ (PK/FK)	Mô tả
1	maPN	char(15)	NotNull	FK	Mã phiếu nhập
2	maSach	char(15)	NotNull	FK	Mã sách
3	soLuong	int			Số lượng
4	thanhTien	bigint			Thành tiền

Tbl4.15: Bảng TblHoaDon

STT	Tên thuộc tính	Kiểu độ rộng	Ràng buộc	Khóa chính/Khóa phụ (PK/FK)	Mô tả
1	maHD	char(15)	NotNull	PK	Mã hóa đơn
2	ngayBan	date			Ngày bán
3	maNV	char(15)	NotNull	FK	Mã nhân viên
4	tongTien	bigint			Tổng tiền

Tbl4.16: Bảng TblNhaXuatBan

STT	Tên thuộc tính	Kiểu độ rộng	Ràng buộc	Khóa chính/Khóa phụ (PK/FK)	Mô tả
1	maNXB	char(15)	NotNull	PK	Mã nhà xuất bản
2	tenNXB	char(30)			Tên nhà xuất bản
3	dienThoai	char(15)			Số điện thoại
4	diaChi	nvarchar(100)			Địa chỉ

Tbl4.17: Bảng TblPhieuNhap

STT	Tên thuộc tính	Kiểu độ rộng	Ràng buộc	Khóa chính/Khóa phụ (PK/FK)	Mô tả
1	maPN	char(15)	NotNull	PK	Mã phiếu nhập
2	ngayNhap	date			Ngày nhập
3	maNXB	char(15)	NotNull	FK	Mã nhà xuất bản
4	maNV	char(15)	NotNull	FK	Mã nhân viên
5	tongTien	bigint			Tổng tiền

Tbl4.18: Bảng TblPhieuNhap

STT	Tên thuộc tính	Kiểu độ rộng	Ràng buộc	Khóa chính/Khóa phụ (PK/FK)	Mô tả
1	maSach	char(15)	NotNull	PK	Mã sách
2	tenSach	nvarchar(50)			Tên sách
3	soLuongTon	int			Số lượng tồn
4	giaNhap	int			Giá nhập
5	giaBan	int			Giá bán
6	maTL	char(15)	NotNull	FK	Mã thể loại
7	maTG	char(15)	NotNull	FK	Mã tác giả
8	maNXB	char(15)	NotNull	FK	Mã nhà xuất bản

Tbl4.19: Bảng TblTacGia

STT	Tên thuộc tính	Kiểu độ rộng	Ràng buộc	Khóa chính/Khóa phụ (PK/FK)	Mô tả
1	maTG	char(15)	NotNull	PK	Mã tác giả
2	tenTG	nvarchar(30)			Tên tác giả
3	diaChi	nvarchar(100)			Địa chỉ

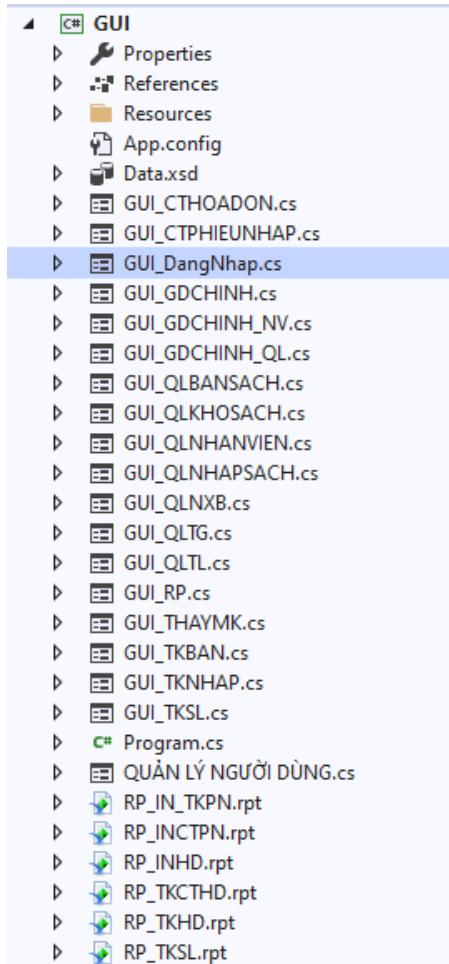
Tbl4.20: Bảng TblTheLoai

STT	Tên thuộc tính	Kiểu độ rộng	Ràng buộc	Khóa chính/Khóa phụ (PK/FK)	Mô tả
1	maTL	char(15)	NotNull	PK	Mã thể loại
2	tenTL	nvarchar(30)			Tên thể loại

CHƯƠNG 5: PHÁT TRIỂN HỆ THỐNG

5.1 Tầng GUI

- Các thành phần trong tầng GUI



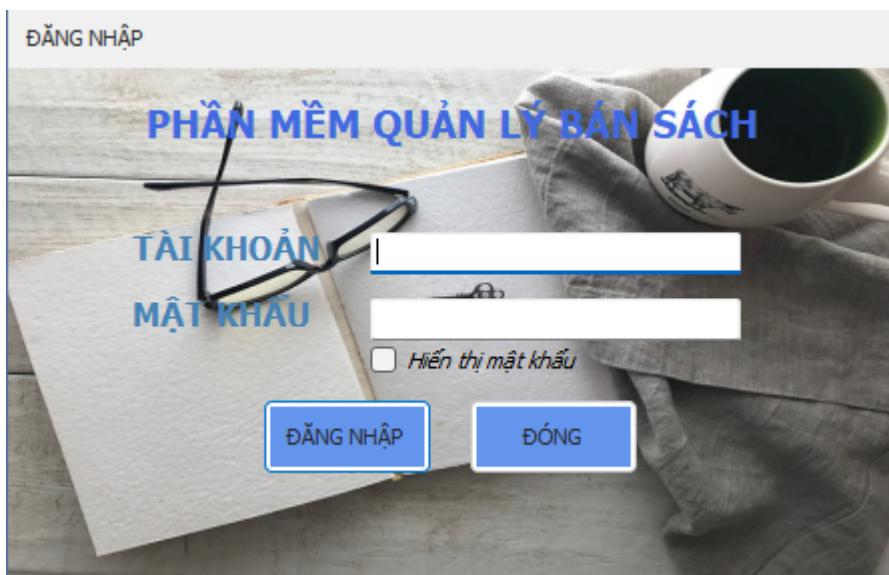
Hình 5.1: Các thành phần trong tầng GUI.

- Các thư viện sử dụng trong tầng GUI

```
using DevExpress.XtraEditors;
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using DTO;
using BLL;
```

- **Giao diện form đăng nhập**

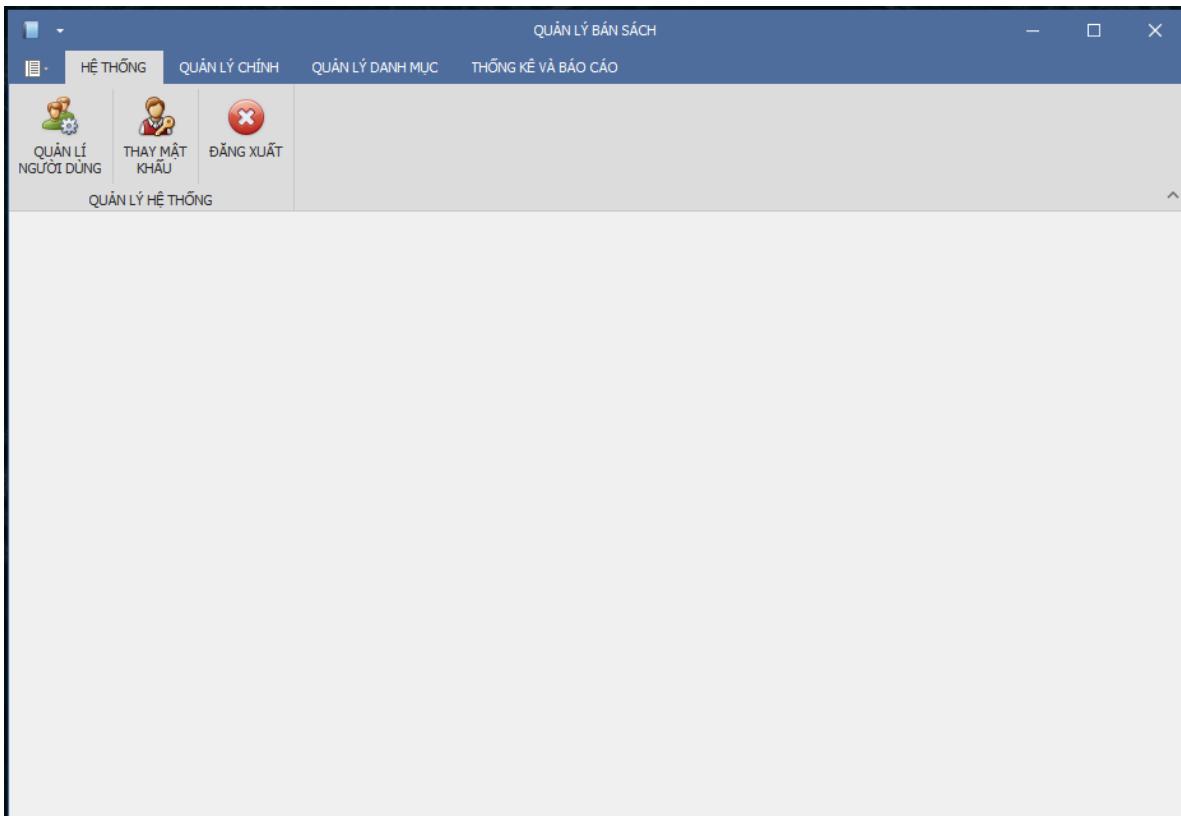
- Là form cho người dùng đăng nhập vào phần mềm.



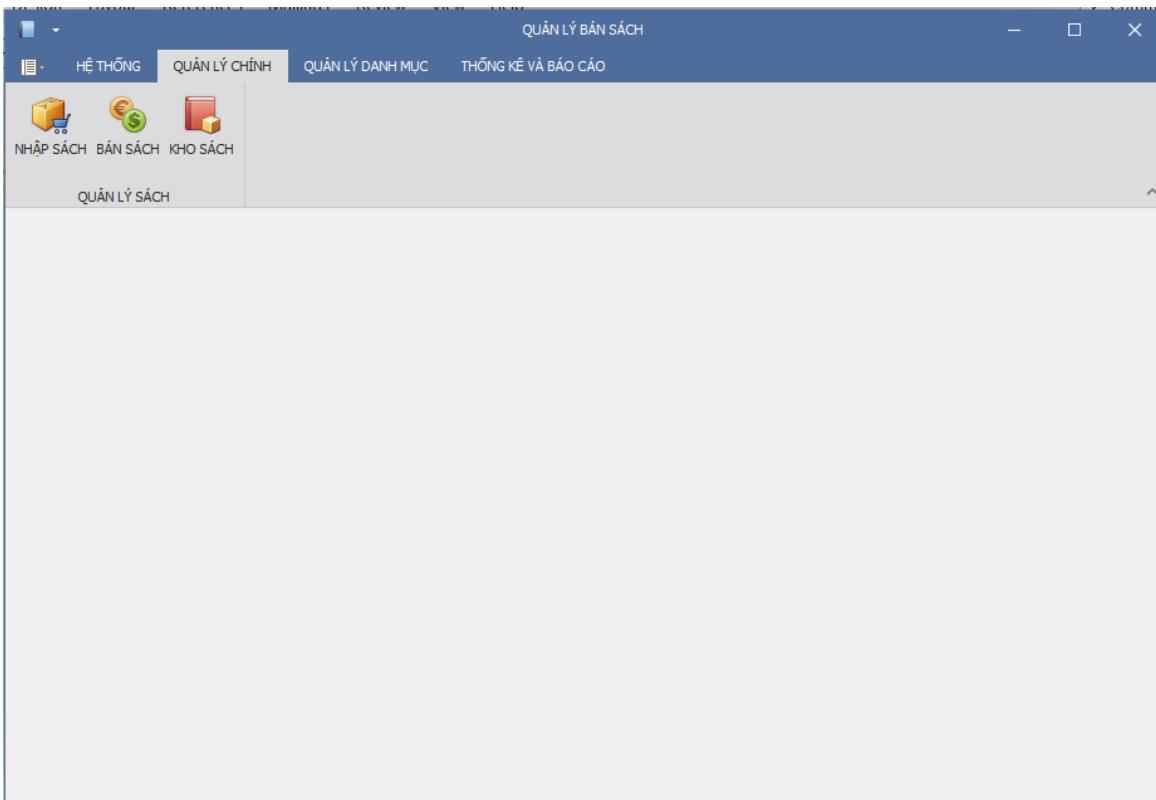
Hình 5.2: giao diện đăng nhập.

- **Giao diện form chính**

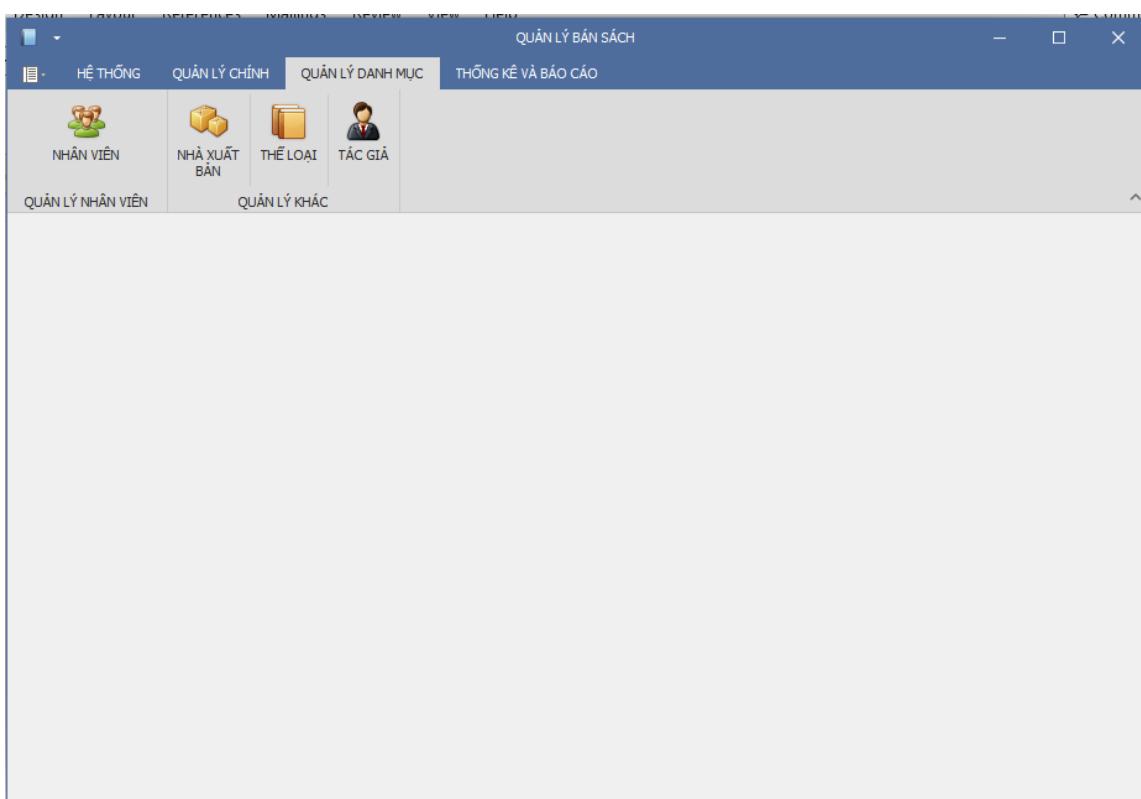
- Đóng vai trò làm form cha khi chọn vào các chức năng nào đó thì form(form con) của chức năng đó sẽ hiện bên trong form cha.



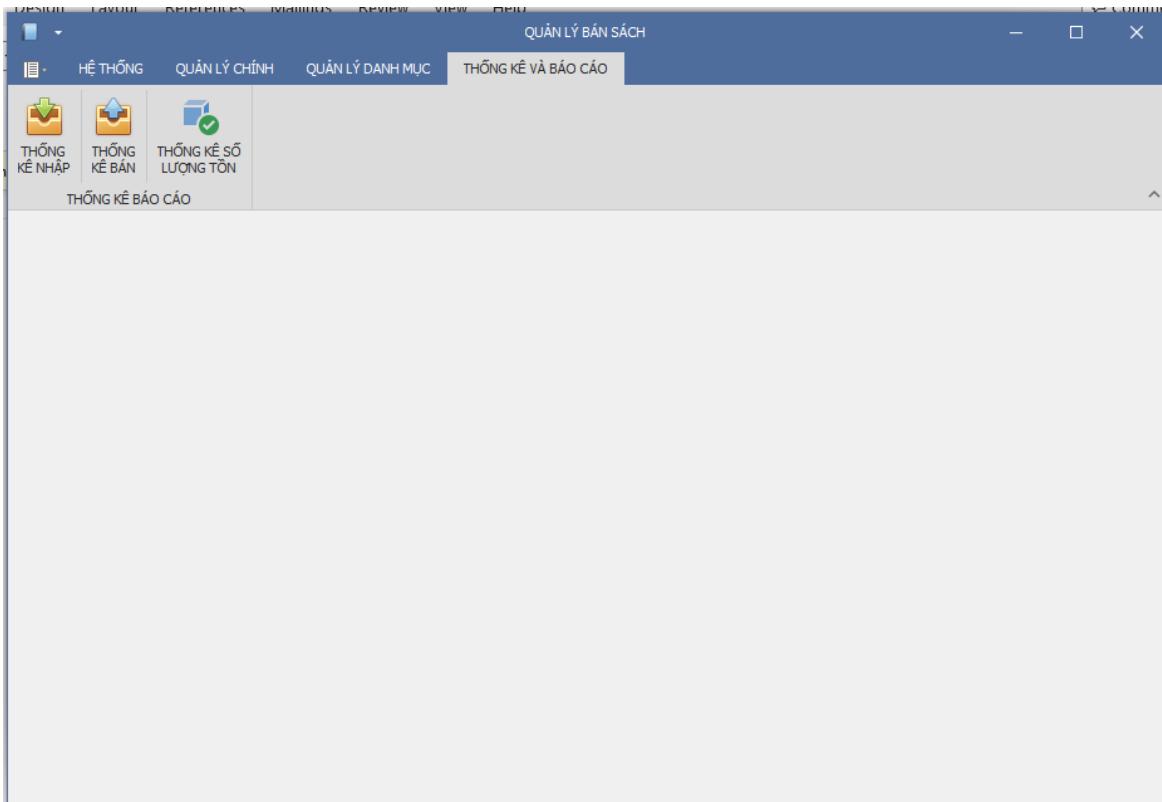
Hình 5.3: giao diện chính mục “Hệ thống”.



Hình 5.4: giao diện chính mục “Quản lí chính”.



Hình 5.5: giao diện chính mục “Quản lí danh mục”.



Hình 5.6: giao diện chính mục “Thông kê báo cáo”.

- **Giao diện form quản lý người dùng**

	id	pw	chucVu	maNV
▶	TKBS_HTD	12345	Sếp	BOSSHLD
	TKNV_NPT	12345	Nhân viên	NV02
	TKNV_VDP	12345	Nhân viên	NV01
*	TKQL_LVH	12345	Quản lý	QL01

Hình 5.7: giao diện quản lý người dùng

- Giao diện form thay mật khẩu

THAY MẬT KHẨU

: Đóng

ĐỔI MẬT KHẨU TÀI KHOẢN: ID: TKBS_HTD

Mật khẩu cũ: _____

Mật khẩu mới: _____

Hiển mật khẩu

Đổi

Hình 5.8: giao diện thay mật khẩu

- Giao diện form quản lý nhập sách

QUẢN LÝ NHẬP SÁCH

: Thêm phiếu nhập | Sửa thông tin | Xóa phiếu nhập | Chi tiết phiếu nhập | Làm mới | Đóng

Thông tin phiếu nhập:

Mã phiếu nhập: _____ Mã nhà xuất bản: KIM MAI SU

Ngày nhập: 10/ 7/2023 _____ Mã nhân viên: BOSSHLD

Tổng tiền: _____

Danh sách phiếu nhập:

Tim theo mã: _____

MA_PHIEU_NHAP	NGAY_NHAP	TEN_NXB	MA_NV	TONG_TIEN
PN00001	1/1/2022	KIM MAI SU	BOSSHLD	2520000
PN00002	1/1/2022	SACH VIET	BOSSHLD	3530000
PN00003	1/1/2022	DUNG HA	BOSSHLD	2430000
PN00004	1/1/2022	SHURIMA	BOSSHLD	2740000
PN00005	1/1/2022	SATA BERY	BOSSHLD	6780000
PN00006	2/1/2022	KIM MAI SU	BOSSHLD	1560000
PN00007	2/1/2022	DUNG HA	BOSSHLD	2835000
PN00008	2/1/2022	SHURIMA	BOSSHLD	2665000
PN00009	2/2/2022	SACH VIET	BOSSHLD	2955000
PN00010	2/1/2022	SATA BERY	BOSSHLD	4600000
PN00011	1/5/2022	KIM MAI SU	BOSSHLD	1630000

Hình 5.9: giao diện quản lí nhập sách

- Giao diện form chi tiết phiếu nhập

CHI TIẾT PHIẾU NHẬP

: Thêm | Sửa | Xóa | Đóng

Danh sách sách:

Tim tên sách: _____

MA_SACH	TEN_SACH	SO_LUONG_TON	GIA_NHAP	GIA_B
893123456789	Kiến thức sống	36	20000	26000
893332147890	Bạn là ai?	34	32000	41000
*				

Thông tin chi tiết hóa đơn:

Mã phiếu nhập: PN00001

Mã sách:

Số lượng: 0

Giá tiền: _____

Danh sách sách đã thêm trong phiếu nhập:

MA_PHIEU_NHAP	MA_SACH	SO_LUONG_NHAP	GIA_NHAP
PN00001	893332147890	60	1920000
PN00001	893123456789	30	600000
*			

Hình 5.10: giao diện quản lí chi tiết phiếu nhập

- Giao diện form quản lý bán sách

MA_HOA_DON	NGAY_BAN	MA_NHAN_VIEN	TONG_TIEN
HD00001	10/1/2023	BOSSHDT	260000
HD000010	1/2/2022	BOSSHDT	185000
HD00002	10/1/2023	BOSSHDT	149000
HD00003	10/1/2023	BOSSHDT	215000
HD00004	1/2/2022	BOSSHDT	275000
HD00005	1/2/2022	BOSSHDT	199000
HD00006	1/2/2022	BOSSHDT	161000
HD00007	1/2/2022	BOSSHDT	154000
HD00008	1/2/2022	BOSSHDT	100000
HD00009	1/2/2022	BOSSHDT	351000
HD00010	1/2/2022	BOSSHDT	169000
HD000100	1/12/2022	NV02	230000

Hình 5.11: giao diện quản lý bán sách

- Giao diện form chi tiết hóa đơn

MA_SACH	TEN_SACH	SO_LUONG_TON	GIA_NHAP
893123129991	Cái bóng của quá khứ	33	90000
893123456789	Kiến thức sống	36	20000
893271234908	Lạc vào thế giới cổ tích		29000
893312875213	Tình yêu không có lỗi	36	56000
893332147890	Bạn là ai?	34	32000
893497123765	Cuộc sống là một chuyến đi	35	43000
907446401100	Linh hồn hàn hùng	25	50000

MA_HOA_DON	MA_SACH	SO_LUONG	GIA_BAN	THANH_TIEN
HD00001	893123129991	2	117000	234000
HD00001	893123456789	1	26000	26000
*				

Hình 5.12: giao diện quản lý chi tiết hóa đơn

• Giao diện form kho sách

QUẢN LÝ KHO SÁCH

: Thêm thông tin sách | Sửa thông tin sách | Xóa sách | Làm mới | Đóng

Thông tin sách:

Mã sách:	Giá nhập:	Thể loại:
Tên sách:	Giá bán:	Tác giả:
Số lượng tồn:	Nhà xuất bản:	

Tìm kiếm sách:

Mã sách:	Tên sách:	Thể loại:	Tác giả:	Tìm
----------	-----------	-----------	----------	------------

Danh sách sách:

MA_SACH	TEN_SACH	SO_LUONG_TON	GIA_NHAP	GIA_BAN	TEN_TL	TEN_TG	TEN_NXB
893123129991	Cái bóng của quá khứ	33	90000	117000	Bí kíp	Lê Đại Nam	SATA BERY
893123456789	Kiến thức sống	36	20000	26000	Cẩm nang	Vũ Mai An	KIM MAI SU
893271234908	Lạc vào thế giới cổ tích	35	29000	37000	Truyện ngắn	Nguyễn Đình Dương	SHURIMA
893312875213	Tình yêu không có lỗi	36	56000	72000	Tình cảm	Phan Đại Vũ	DUNG HA
893332147890	Bạn là ai?	34	32000	41000	Cẩm nang	Vũ Mai An	KIM MAI SU
893497123765	Cuộc sống là một chuyến đi	35	43000	54000	Du lịch	Nguyễn Đình Dương	SHURIMA
893554690120	Lịch sử hào hùng	35	52000	67000	Lịch sử	Đinh Nam Học	SACH VIET
893758395012	Tôi khi 30 tuổi	35	25000	32000	Hồi ký	Phan Đại Vũ	DUNG HA
893900278921	Thế giới là một bí ẩn	35	31000	40000	Tiểu thuyết	Đinh Nam Học	SACH VIET

Hình 5.13: giao diện quản lí kho sách

• Giao diện form nhân viên

QUẢN LÝ NHÂN VIÊN

: Thêm nhân viên | Sửa thông tin nhân viên | Xóa nhân viên | Đóng

Thông tin nhân viên:

Mã nhân viên:	Căn cước:	Địa chỉ:
Tên nhân viên:	Số điện thoại:	Lịch làm:
Ngày sinh:	10/ 7/2023	

Danh sách nhân viên:

maNV	tenNV	ngaySinh	cCCD	soDT	diaChi	lichLam
NV01	Vũ Đình Phong	12/3/2000	013203459988	0775104623	24 Tran Quoc Toan, Hang Bai, Hoan Kiem, Ha Noi	Ca sáng _T2 - T7
NV02	Nguyễn Phương Trinh	5/21/2001	018204007382	0986237799	52 Nguyen Thai Hoc, Ba Dinh, Ha Noi	Ca Chiều _T2 - T7
QL01	Lê Văn Hải	10/12/2000	014204335763	0376860989	3b Dang Thai Than, Phan Chu Trinh, Hoan Kiem, Ha Noi	Full time _T2 - T7
*						

Hình 5.14: giao diện quản lí nhân viên

• Giao diện form nhà xuất bản

QUẢN LÝ NHÀ XUẤT BẢN

: Thêm nhà xuất bản | Sửa thông tin | Xóa nhà xuất bản | Đóng

Thông tin nhà xuất bản:

Mã nhà xuất bản:	<input type="text"/>	Số điện thoại:	<input type="text"/>
Tên nhà xuất bản:	<input type="text"/>	Địa chỉ:	<input type="text"/>

Danh sách nhà xuất bản:

Tìm theo mã: Tìm theo tên:

MA_NXB	TEN_NXB	DIEN_THOAI	DIA_CHI
NXB22113	KIM MAI SƯ	0997777333	Sai Dong, Long Bien, Ha Noi
NXB33560	DUNG HA	0666699999	Dong Dau, Duc Tu, Dong Anh, Ha Noi
NXB66821	SHURIMA	0987222333	Dong Ky, Dong Quang, Tu Son, Bac Ninh
NXB76430	SACH VIET	0333666999	5 Nguyen Thai Hoc, Da Lat, Lam Dong
NXB99832	SATA BERY	0386644222	178A Truong Dinh, Hai Ba Trung, Ha Noi
*			

Hình 5.15: giao diện quản lí nhà xuất bản

• Giao diện form thể loại

QUẢN LÝ THỂ LOẠI

: Thêm thể loại | Sửa thông tin | Xóa thể loại | Đóng

Thông tin thể loại:

Mã thể loại:	<input type="text"/>	Tên thể loại:	<input type="text"/>
--------------	----------------------	---------------	----------------------

Danh sách thể loại:

Tìm theo mã: Tìm theo tên:

MA_THE_LOAI	TEN_THE_LOAI
tbk	Bí kíp
tln	Cẩm nang
tdl	Du lịch
thd	Hành Động
thk	Hài kịch
thky	Hội ký
tlkd	Kinh dị
tls	Lịch sử
tlc	Tình cảm
tltn	Truyện ngắn
tltt	Tiểu thuyết
tvvt	Viễn Tưởng
*	

Hình 5.16: giao diện quản lí thể loại

- Giao diện form tác giả

Hình 5.17: giao diện quản lí tác giả

- Giao diện form thống kê nhập

Hình 5.18: giao diện thống kê nhập

- Giao diện form thống kê bán

Hình 5.19: giao diện thống kê bán

- Giao diện form thống kê số lượng tồn

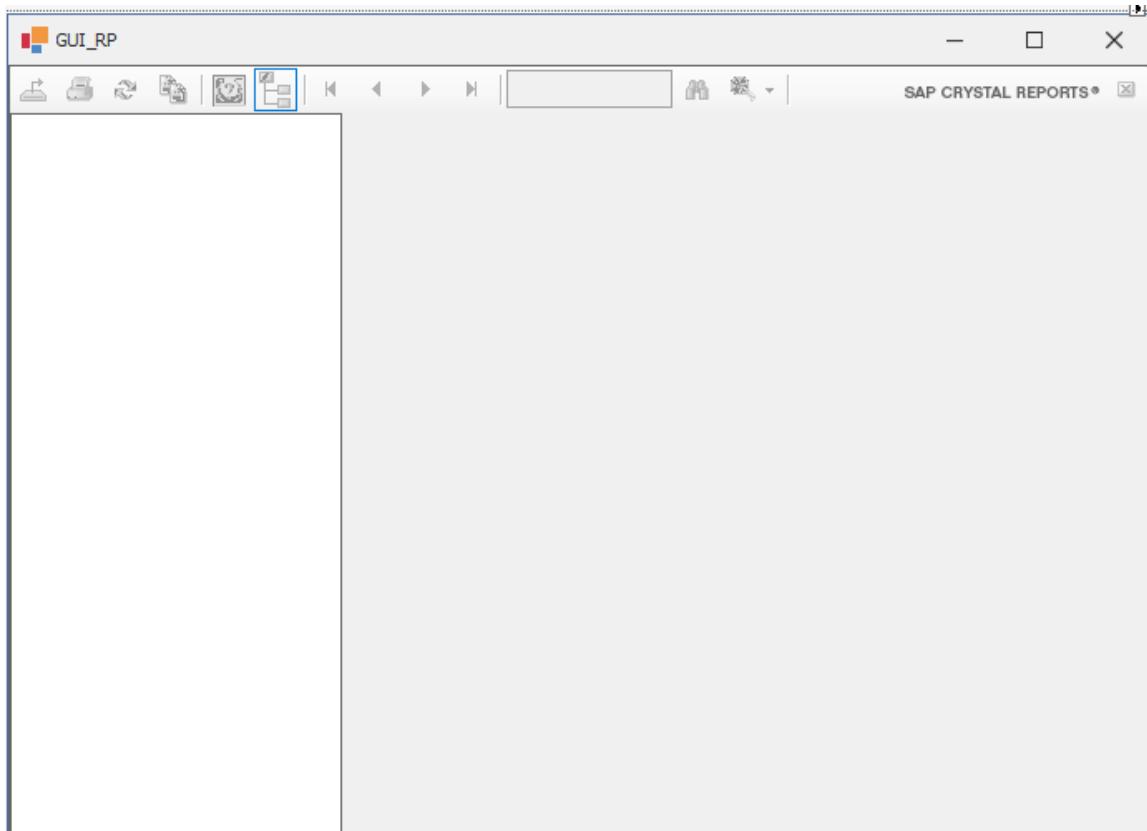
DANH SÁCH SÁCH					
TÊN_SÁCH	SỐ_LƯỢNG_TỒN	GIÁ_NHẬP	GIÁ_BÁN	TÊN_NXB	TÊN_TG
Cái bóng của quá khứ	33	90000	117000	SATA BERY	Lê Đại Nam
Kiến thức sống	36	20000	26000	KIM MAI SU	Vũ Mai An
Lạc vào thế giới cổ tích	35	29000	37000	SHURIMA	Nguyễn Đình Dương
Tình yêu không có lỗi	36	56000	72000	DUNG HA	Phan Đại Vũ
Bạn là ai?	34	32000	41000	KIM MAI SU	Vũ Mai An
Cuộc sống là một chuyến đi	35	43000	54000	SHURIMA	Nguyễn Đình Dương
Lịch sử hào hùng	35	52000	67000	SACH VIET	Đinh Nam Học
Tôi khi 30 tuổi	35	25000	32000	DUNG HA	Phan Đại Vũ
Thế giới là một bí ẩn	35	31000	40000	SACH VIET	Đinh Nam Học
Bí mật của ngôi làng	35	76000	98000	SATA BERY	Lê Đại Nam

CHỌN NHÀ XUẤT BẢN CHỌN TÁC GIẢ:

[Printer icon] [Refresh icon] [Close icon]

Hình 5.20: giao diện thống kê số lượng tồn

- Giao diện form report



Hình 5.21: giao diện hiển thị báo cáo

- **Mẫu báo cáo phiếu nhập**

CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM Độc lập - Tự do - Hạnh phúc				
 CỬA HÀNG SÁCH ~~DUNGHA~~				
BÁO CÁO TÌNH HÌNH CÔNG VIỆC NHẬP SÁCH				
Thống kê thời điểm nhập:/...../..... .				
Họ và tên: Chức vụ:.....				
MÃ PHIẾU NHẬP	NGÀY NHẬP	TÊN NHÀ XUẤT BẢN	NHÂN VIÊN TẠO	TỔNG TIỀN
PN00001	01/01/2022	KIM MAI SU	Hà Tiên Dũng	2,520,000
PN00006	02/01/2022	KIM MAI SU	Hà Tiên Dũng	1,560,000
PN00011	01/05/2023	KIM MAI SU	Hà Tiên Dũng	1,620,000
Tổng tiền chi cho việc nhập sách				5,700,000
Ghi chú :				
Hà Nội, ngày..... tháng..... năm Người báo cáo				

Hình 5.22: mẫu báo cáo nhập

- **Mẫu báo cáo chi tiết phiếu nhập**

CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM Độc lập - Tự do - Hạnh phúc						
 CỬA HÀNG SÁCH ~~DUNGHA~~						
BÁO CÁO THỐNG KÊ CÁC QUYỀN SÁCH ĐÃ ĐƯỢC NHẬP						
Thống kê thời điểm nhập:/...../..... .						
Họ và tên: Chức vụ:.....						
MÃ PHIẾU NHẬP	TÊN SÁCH	NGÀY NHẬP	SỐ LƯỢNG	GIÁ NHẬP	THÀNH TIỀN	NHÀ XUẤT BẢN
PN00006	Kiến thức sống	02/01/2022	30	20,000	600,000	KIM MAI SU
PN00006	Bạn là ai?	02/01/2022	30	32,000	960,000	KIM MAI SU
PN00001	Bạn là ai?	01/01/2022	60	32,000	1,920,000	KIM MAI SU
PN00001	Kiến thức sống	01/01/2022	30	20,000	600,000	KIM MAI SU
PN00011	Bạn là ai?	01/05/2023	35	32,000	1,120,000	KIM MAI SU
PN00011	Kiến thức sống	01/05/2023	25	20,000	500,000	KIM MAI SU
TỔNG SỐ LƯỢNG SÁCH NHẬP				210	TỔNG TIỀN CHI	5,700,000
Ghi chú :						
Hà Nội, ngày..... tháng..... năm Người báo cáo						

Hình 5.23: mẫu báo cáo chi tiết nhập

- **Mẫu báo cáo bán sách**

 <p>CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM Độc lập - Tự do - Hạnh phúc</p>	<p>BÁO CÁO TÌNH HÌNH CÔNG VIỆC BÁN SÁCH</p> <p>Thống kê thời điểm nhập:/...../..... .</p> <p>Họ và tên: Chức vụ:</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>MÃ_HÓA ĐƠN</th> <th>NGÀY BÁN</th> <th>NHÂN VIÊN LẤP</th> <th>TỔNG TIỀN</th> </tr> </thead> <tbody> <tr> <td>HD00001</td> <td>10/01/2023</td> <td>Hà Tiên Dũng</td> <td>260,000</td> </tr> <tr> <td>HD00002</td> <td>10/01/2023</td> <td>Hà Tiên Dũng</td> <td>149,000</td> </tr> <tr> <td>HD00003</td> <td>10/01/2023</td> <td>Hà Tiên Dũng</td> <td>215,000</td> </tr> <tr> <td></td> <td></td> <td></td> <td>624,000</td> </tr> </tbody> </table> <p>Ghi chú :</p>	MÃ_HÓA ĐƠN	NGÀY BÁN	NHÂN VIÊN LẤP	TỔNG TIỀN	HD00001	10/01/2023	Hà Tiên Dũng	260,000	HD00002	10/01/2023	Hà Tiên Dũng	149,000	HD00003	10/01/2023	Hà Tiên Dũng	215,000				624,000
MÃ_HÓA ĐƠN	NGÀY BÁN	NHÂN VIÊN LẤP	TỔNG TIỀN																		
HD00001	10/01/2023	Hà Tiên Dũng	260,000																		
HD00002	10/01/2023	Hà Tiên Dũng	149,000																		
HD00003	10/01/2023	Hà Tiên Dũng	215,000																		
			624,000																		
<p>Hà Nội, ngày..... tháng..... năm</p> <p>Người báo cáo</p>																					

Hình 5.24: mẫu báo cáo bán sách

- **Mẫu báo cáo chi tiết bán sách**

 <p>CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM Độc lập - Tự do - Hạnh phúc</p>	<p>BÁO CÁO CHI TIẾT NHỮNG QUYỀN SÁCH ĐÃ BÁN</p> <p>Thống kê thời điểm nhập:/...../..... .</p> <p>Họ và tên: Chức vụ:</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>MÃ_HÓA ĐƠN</th> <th>TÊN SÁCH</th> <th>SỐ LƯỢNG</th> <th>GIÁ BÁN</th> <th>THÀNH TIỀN</th> </tr> </thead> <tbody> <tr> <td>HD00001</td> <td>Cái bóng của quá khứ</td> <td>2</td> <td>117,000</td> <td>234,000</td> </tr> <tr> <td>HD00001</td> <td>Kiến thức sống</td> <td>1</td> <td>26,000</td> <td>26,000</td> </tr> <tr> <td>HD00002</td> <td>Ban là ai?</td> <td>2</td> <td>41,000</td> <td>82,000</td> </tr> <tr> <td>HD00002</td> <td>Lịch sử hào hùng</td> <td>1</td> <td>67,000</td> <td>67,000</td> </tr> <tr> <td>HD00003</td> <td>Cái bóng của quá khứ</td> <td>1</td> <td>117,000</td> <td>117,000</td> </tr> <tr> <td>HD00003</td> <td>Kiến thức sống</td> <td>1</td> <td>26,000</td> <td>26,000</td> </tr> <tr> <td>HD00003</td> <td>Tình yêu không có lỗi</td> <td>1</td> <td>72,000</td> <td>72,000</td> </tr> <tr> <td>TỔNG SỐ LƯỢNG</td> <td>9</td> <td></td> <td>TỔNG TIỀN THU</td> <td>624,000</td> </tr> </tbody> </table> <p>Ghi chú :</p>	MÃ_HÓA ĐƠN	TÊN SÁCH	SỐ LƯỢNG	GIÁ BÁN	THÀNH TIỀN	HD00001	Cái bóng của quá khứ	2	117,000	234,000	HD00001	Kiến thức sống	1	26,000	26,000	HD00002	Ban là ai?	2	41,000	82,000	HD00002	Lịch sử hào hùng	1	67,000	67,000	HD00003	Cái bóng của quá khứ	1	117,000	117,000	HD00003	Kiến thức sống	1	26,000	26,000	HD00003	Tình yêu không có lỗi	1	72,000	72,000	TỔNG SỐ LƯỢNG	9		TỔNG TIỀN THU	624,000
MÃ_HÓA ĐƠN	TÊN SÁCH	SỐ LƯỢNG	GIÁ BÁN	THÀNH TIỀN																																										
HD00001	Cái bóng của quá khứ	2	117,000	234,000																																										
HD00001	Kiến thức sống	1	26,000	26,000																																										
HD00002	Ban là ai?	2	41,000	82,000																																										
HD00002	Lịch sử hào hùng	1	67,000	67,000																																										
HD00003	Cái bóng của quá khứ	1	117,000	117,000																																										
HD00003	Kiến thức sống	1	26,000	26,000																																										
HD00003	Tình yêu không có lỗi	1	72,000	72,000																																										
TỔNG SỐ LƯỢNG	9		TỔNG TIỀN THU	624,000																																										
<p>Hà Nội, ngày..... tháng..... năm</p> <p>Người báo cáo</p>																																														

Hình 5.25: mẫu báo cáo chi tiết bán sách

- **Mẫu báo cáo số lượng sách tồn**

CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM Độc lập - Tự do - Hạnh phúc					
BÁO CÁO SỐ LƯỢNG SÁCH TỒN					
Thời điểm thống kê:	/	/
Họ và tên:				
Chức vụ:				
TÊN SÁCH	GIÁ NHẬP	GIÁ BÁN	tenNXB	tenTG	SỐ LƯỢNG TỒN
Cái bóng của quá khứ	90,000	117,000	SATA BERY	Lê Đại Nam	33
Kiến thức sống	20,000	26,000	KIM MAI SU	Vũ Mai An	36
Lạc vào thế giới cổ tích	29,000	37,000	SHURIMA	Nguyễn Đình Dương	35
Tình yêu không có lỗi	56,000	72,000	DUNG HA	Phan Đại Vũ	36
Bạn là ai?	32,000	41,000	KIM MAI SU	Vũ Mai An	34
Cuộc sống là một chuyến đi	43,000	54,000	SHURIMA	Nguyễn Đình Dương	35
Lịch sử hào hùng	52,000	67,000	SACH VIET	Đinh Nam Học	35
Tôi khi 30 tuổi	25,000	32,000	DUNG HA	Phan Đại Vũ	35
Thế giới là một bí ẩn	31,000	40,000	SACH VIET	Đinh Nam Học	35
Bí mật của ngôi làng	76,000	98,000	SATA BERY	Lê Đại Nam	35
TỔNG SỐ LƯỢNG SÁCH TỒN					349
Ghi chú :				
Hà Nội, ngày..... tháng..... năm..... Người báo cáo					

Hình 5.26: mẫu báo cáo số lượng tồn

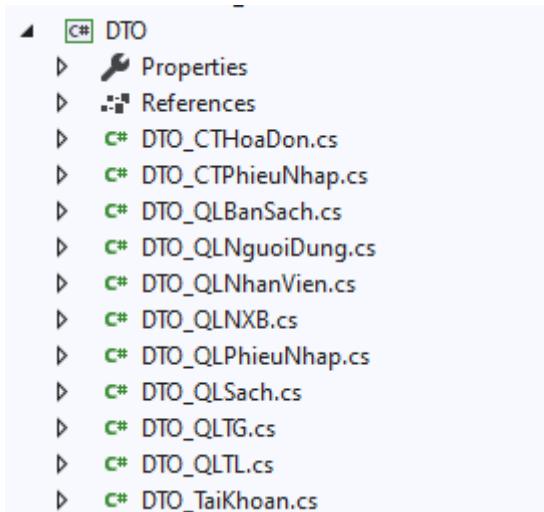
- **Mẫu in hóa đơn**

CỬA HÀNG SÁCH ~~DUNGHA~~ 108-E2 Phuong Mai, Dong Da, Ha Noi ĐT: 0383366423 EMAIL: dunghasonline@outlook.com			
HÓA ĐƠN BÁN SÁCH			
TÊN SÁCH	SỐ LƯỢNG	GIÁ BÁN	THÀNH TIỀN
Kiến thức sống	1	26,000	26000
Cái bóng của quá khứ	2	117,000	234000
TỔNG TIỀN			260000
RẤT HÂN HẠNH ĐƯỢC ĐÓNG TIẾP QUÝ KHÁCH!			

Hình 5.27: mẫu in hóa đơn

5.2 Tầng DTO

- Lớp này đóng vai trò là lớp định nghĩa các bảng trong CSDL, định nghĩa các cột cũng như để gán dữ liệu khi lấy dữ liệu.
- Các thành phần trong tầng DTO



Hình 5.28: Các thành phần trong tầng DTO

- Các Class

- Class DTO_TaiKhoan

```
public class DTO_TaiKhoan
{
    public string tk { get; set; }
    public string mk { get; set; }
}
```

- Class DTO_QLTL

```
public class DTO_QLTL
{
    public string maTL { get; set; }
    public string tenTL { get; set; }
}
```

- Class DTO_QLTG

```
public class DTO_QLTG
{
    public string maTG { get; set; }
    public string tenTG { get; set; }
    public string diaChi { get; set; }
}
```

- Class DTO_QLSach

```

public class DTO_QLSach
{
    public string maS { get; set; }
    public string tenS { get; set; }
    public int soLuongTon { get; set; }
    public int giaNhap { get; set; }
    public int giaBan { get; set; }
    public string maTL { get; set; }
    public string maTG { get; set; }
    public string maNxb { get; set; }

}

```

- Class DTO_QLPhieuNhap

```

public class DTO_QLPhieuNhap
{
    public string maPN { get; set; }
    public string ngayNhap { get; set; }
    public string maNxb { get; set; }
    public string maNV { get; set; }
    public long tongTien { get; set; }
}

```

- Class DTO_QLNXB

```

public class DTO_QLNXB
{
    public string maNxb { get; set; }
    public string tenNxb { get; set; }
    public string dienThoai { get; set; }
    public string diaChi { get; set; }
}

```

- Class DTO_QLNhanVien

```

public class DTO_QLNhanVien
{
    public string maNV { get; set; }
    public string tenNV { get; set; }
    public string ngaySinh { get; set; }
    public string cccd { get; set; }
    public string soDT { get; set; }
    public string diaChi { get; set; }
    public string lichLam { get; set; }
}

```

- Class DTO_QLNguoiDung

```

public class DTO_QLNguoiDung
{
    public string iD { get; set; }
    public string pW { get; set; }
    public string chucVu { get; set; }
    public string maNV { get; set; }
}

```

- Class DTO_QLBanSach

```

public class DTO_QLBanSach
{
    public string maHD { get; set; }
    public string ngayBan { get; set; }
    public string maNV { get; set; }
    public long tongTien { get; set; }
}

```

- Class DTO_CTPhieuNhap

```

public class DTO_CTPhieuNhap
{
    public string maPN { get; set; }
    public string maS { get; set; }
    public int soLuong { get; set; }
    public long giaNhap { get; set; }
}

```

- Class DTO_CTHoaDon

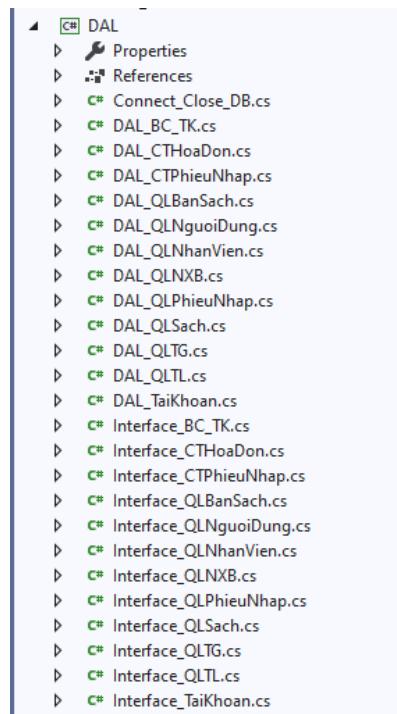
```

public class DTO_CTHoaDon
{
    public string maHD { get; set; }
    public string maS { get; set; }
    public int sL { get; set; }
    public long giaTien { get; set; }
}

```

5.3 Tầng DAL

- Tầng này đóng vai trò truy xuất và cập nhật với cơ sở dữ liệu.
- Các thành phần trong tầng DAL



Hình 5.29: Các thành phần trong tầng DAL

- Các thư viện sử dụng trong tầng DAL

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Data.SqlClient;
using System.Data;
using DTO;

```

- Các class thực thi trong DAL
 - Các class hỗ trợ sử dụng trong DAL

```

SqlCommand cmd;
SqlDataAdapter adt = new SqlDataAdapter();
DataTable dt = new DataTable();

```

- Các class interface
- + Class [Interface_TaiKhoan](#)

```

public interface Interface_TaiKhoan
{
    string dangNhap(DTO_TaiKhoan tk);
    bool doiMaKhau(DTO_TaiKhoan tk);
    string getCV(string id);
}

```

- + Class [Interface_QLTL](#)

```

interface Interface_QLTL
{
    DataTable showTL();
    DataTable searchMaTL(string maTL);
    DataTable searchTenTL(string tenTL);
    bool addTL(DTO_QLTL tl);
    bool editTL(DTO_QLTL tl);
    bool deleteTL(DTO_QLTL tl);
}

```

- + Class [Interface_QLTG](#)

```

interface Interface_QLTG
{
    DataTable showTG();
    DataTable searchMaTG(string maTG);
    DataTable searchTenTG(string tenTG);
    bool addTG(DTO_QLTG tg);
    bool editTG(DTO_QLTG tg);
    bool deleteTG(DTO_QLTG tg);
}

```

- + Class [Interface_QLSach](#)

```

interface Interface_QLSach
{
    DataTable showSach();
    bool addSach(DTO_QLSach s);
    bool editSach(DTO_QLSach s);
    bool deleteSach(DTO_QLSach s);
    DataTable loadTenTL();
    DataTable loadTenTG();
}

```

```

        DataTable loadTenNxb();
        string loadMaTL(string tenTL);
        string loadMaTG(string tenTG);
        string loadMaNxb(string tenNxb);
        DataTable searchSach(string maS, string tenS, string tl, string tg);
    }

```

+ Class [Interface_QLPhieuNhap](#)

```

interface Interface_QLPhieuNhap
{
    DataTable showPN();
    bool addPN(DTO_QLPhieuNhap pn);
    bool editPN(DTO_QLPhieuNhap pn);
    bool deletePN(DTO_QLPhieuNhap pn);
    DataTable searchPN(string maPN);
    DataTable loadTenNxb();
    String getMaNxb(string tenNxb);
    String getMaNV(string tk);
}

```

+ Class [Interface_QLNxb](#)

```

interface Interface_QLNxb
{
    DataTable showNxb();
    DataTable searchMaNxb(string maNxb);
    DataTable searchTenNxb(string tenNxb);
    bool addNxb(DTO_QLNxb nxb);
    bool editNxb(DTO_QLNxb nxb);
    bool deleteNxb(DTO_QLNxb nxb);
}

```

+ Class [Interface_QLNhanVien](#)

```

interface Interface_QLNhanVien
{
    DataTable ShowNV();
    bool addNV(DTO_QLNhanVien nv);
    bool editNV(DTO_QLNhanVien nv);
    bool deleteNV(DTO_QLNhanVien nv);
    DataTable searchMaNV(string maNV);
    DataTable searchTenNV(string tenNV);
}

```

+ Class [Interface_QLNguoiDung](#)

```

interface Interface_QLNguoiDung
{
    DataTable showUser();
    bool addUser(DTO_QLNguoiDung nd);
    bool editUser(DTO_QLNguoiDung nd);
    bool deleteUser(DTO_QLNguoiDung nd);
    DataTable loadTenNV();
    string loadMaNV(string tenNV);
}

```

+ Class [Interface_QLBanSach](#)

```

interface Interface_QLBanSach
{
    DataTable showHD();
    bool addHD(DTO_QLBanSach hd);
    bool editHD(DTO_QLBanSach hd);
    bool deleteHD(DTO_QLBanSach hd);
    DataTable searchHD(string maHD);
    string getMaNV(string tk);
    DataTable inHD(DTO_QLBanSach hd);
}

```

+ Class [Interface_CTPhieuNhap](#)

```

interface Interface_CTPhieuNhap
{
    DataTable showSach(string maPN);
    DataTable showCTPN(string maPN);
    bool addS(DTO_CTPhieuNhap ctpn);
    bool edits(DTO_CTPhieuNhap ctpn, int slf);
    string getGiaNhap(int sl, string maS);
    bool deleteS(DTO_CTPhieuNhap ctpn);
    DataTable searchTenS(string maPN, string tenS);
    bool updateTongPN(DTO_CTPhieuNhap ctpn);
    bool updateTongPN(string maPN, int tmp);
}

```

+ Class [Interface_CTHoaDon](#)

```

interface Interface_CTHoaDon
{
    DataTable showSach();
    DataTable showCTHD(string maHD);
    bool addS(DTO_CTHoaDon cthd);
    bool deleteS(DTO_CTHoaDon cthd);
    string getGiaBan(int sl, string maS);
    DataTable searchTenS(string tenS);
    bool updateTongHD(DTO_CTHoaDon cthd);
    bool updateTongHD(string maHD, int tmp);
}

```

+ Class [Interface_BC_TK](#)

```

interface Interface_BC_TK
{
    DataTable showDSPN();
    DataTable showCTDSPN(string maPN);
    DataTable inPN(string thang, string nam, string nxb);
    DataTable inCTPN(string thang, string nam, string nxb);
    DataTable LoadTenNXB();
    DataTable LoadTenTG();
    DataTable showDSHD();
    DataTable showCTDSHD(string maPN);
    DataTable inHD(string thang, string nam);
    DataTable inCTHD(string thang, string nam);
    DataTable showDSS();
    DataTable inTKSL(string tenNXB, string tenTG);
}

```

- Các class làm việc với CSDL

+ Class `Connect_Close_DB`

o Các Method của class

▪ `public void connect_DB()`

```
public void connect_DB()
{
    sqlcon = new SqlConnection(strCon);
    if (sqlcon == null)
    {
        sqlcon.Open();
    }
    if (sqlcon.State == ConnectionState.Closed)
    {
        sqlcon.Open();
    }
}
```

- Mở kết nối tới SQL.

▪ `public void Close_DB()`

```
public void Close_DB()
{
    if (sqlcon != null || sqlcon.State == ConnectionState.Open)
    {
        sqlcon.Close();
    }
}
```

- Đóng kết nối SQL.

+ Class `DAL_TaiKhoan`

```
public class DAL_TaiKhoan : Connect_Close_DB, Interface_TaiKhoan
```

o Các Method của class

▪ `public string dangNhap(DTO_TaiKhoan tk)`

```
public string dangNhap(DTO_TaiKhoan tk)
{
    connect_DB();
    string mk = null;
    cmd = sqlcon.CreateCommand();
    cmd.CommandText = $"SELECT pw FROM TblTaiKhoan AS tk WHERE
id ='{tk.tk}'";
    var rd = cmd.ExecuteReader();
    while (rd.Read())
    {
        mk = rd.GetString(0);
    }
    rd.Close();
    Close_DB();
    return mk;
}
```

- Trả về mật khẩu của tài khoản nhập vào.

```

    ■ public bool doiMaKhau(DTO_TaiKhoan tk)

public bool doiMaKhau(DTO_TaiKhoan tk)
{
    connect_DB();
    cmd = sqlcon.CreateCommand();
    cmd.CommandText = $"UPDATE TblTaiKhoan SET pw = '{tk.mk}'"
    WHERE id = '{tk.tk}'";
    int kt = cmd.ExecuteNonQuery();
    if (kt > 0)
    {
        Close_DB();
        return true;
    }
    return false;
}

```

- Cập nhật lại mật khẩu của tài khoản trong sql.

■ public string getCV(string id)

```

public string getCV(string id)
{
    connect_DB();
    string tmp = null;
    cmd = sqlcon.CreateCommand();
    cmd.CommandText = $"SELECT chucVu FROM TblTaiKhoan WHERE id ='{id}'";
    var rd = cmd.ExecuteReader();
    while (rd.Read())
    {
        tmp = rd.GetString(0);
    }
    rd.Close();
    Close_DB();
    return tmp;
}

```

- Trả về chức vụ của nhân viên.

+ Class DAL_QLTL

```
public class DAL_QLTL : Connect_Close_DB, Interface_QLTL
```

- Các Method của class

■ public bool addTL(DTO_QLTL tl)

```

public bool addTL(DTO_QLTL tl)
{
    connect_DB();
    cmd = sqlcon.CreateCommand();
    cmd.CommandText = $@"INSERT INTO TblTheLoai VALUES
    ('{tl.maTL}', N'{tl.tenTL}')";
    int kt = cmd.ExecuteNonQuery();
    if(kt > 0)
    {
        Close_DB();
        return true;
    }
    Close_DB();
    return false;
}

```

- Trả về kết quả sau khi thực hiện thêm thể loại có thành công hay không.

- `public bool deleteTL(DTO_QLTL tl)`

```
public bool deleteTL(DTO_QLTL tl)
{
    connect_DB();
    cmd = sqlcon.CreateCommand();
    cmd.CommandText = $"DELETE TblTheLoai
WHERE maTL = '{tl.maTL}'";
    int kt = cmd.ExecuteNonQuery();
    if (kt > 0)
    {
        Close_DB();
        return true;
    }
    Close_DB();
    return false;
}
```

- Trả về kết quả sau khi thực hiện xóa thẻ loại có thành công hay không.

- `public bool editTL(DTO_QLTL tl)`

```
public bool editTL(DTO_QLTL tl)
{
    connect_DB();
    cmd = sqlcon.CreateCommand();
    cmd.CommandText = $"UPDATE TblTheLoai SET tenTL = N'{tl.tenTL}'
WHERE maTL = '{tl.maTL}'";
    int kt = cmd.ExecuteNonQuery();
    if (kt > 0)
    {
        Close_DB();
        return true;
    }
    Close_DB();
    return false;
}
```

- Trả về kết quả sau khi thực hiện sửa thẻ loại có thành công hay không.

- `public DataTable searchMaTL(string maTL)`

```
public DataTable searchMaTL(string maTL)
{
    connect_DB();
    cmd = sqlcon.CreateCommand();
    cmd.CommandText = $"SELECT maTL AS MA_THE_LOAI, tenTL AS
TEN_THE_LOAI FROM TblTheLoai WHERE matL LIKE '%{maTL}%'";
    var dtMaTL = new DataTable();
    adt.SelectCommand = cmd;
    dtMaTL.Clear();
    adt.Fill(dtMaTL);
    Close_DB();
    return dtMaTL;
}
```

- Trả về bảng dữ liệu tìm kiếm thẻ loại theo mã.

▪ `public DataTable searchTenTL(string tenTL)`

```
public DataTable searchTenTL(string tenTL)
{
    connect_DB();
    cmd = sqlcon.CreateCommand();
    cmd.CommandText = $"SELECT maTL AS MA_THE_LOAI, tenTL AS
TEN_THE_LOAI FROM TblTheLoai WHERE tenTL LIKE N'%{tenTL}%'";
    var dtTenTL = new DataTable();
    adt.SelectCommand = cmd;
    dtTenTL.Clear();
    adt.Fill(dtTenTL);
    Close_DB();
    return dtTenTL;
}
```

- Trả về bảng dữ liệu tìm kiếm thẻ loại theo tên.

▪ `public DataTable showTL()`

```
public DataTable showTL()
{
    connect_DB();
    cmd = sqlcon.CreateCommand();
    cmd.CommandText = $"SELECT maTL AS MA_THE_LOAI, tenTL AS
TEN_THE_LOAI FROM TblTheLoai";
    adt.SelectCommand = cmd;
    dt.Clear();
    adt.Fill(dt);
    Close_DB();
    return dt;
}
```

- Trả về bảng dữ liệu danh sách thẻ loại.

+ Class `DAL_QLTG`

```
public class DAL_QLTG : Connect_Close_DB, Interface_QLTG
```

○ Các Method của class

▪ `public bool addTG(DTO_QLTG tg)`

```
public bool addTG(DTO_QLTG tg)
{
    connect_DB();
    cmd = sqlcon.CreateCommand();
    cmd.CommandText = $"INSERT INTO TblTacGia VALUES
({tg.maTG}', N'{tg.tenTG}', N'{tg.diaChi}')";
    int kt = cmd.ExecuteNonQuery();
    if (kt > 0)
    {
        Close_DB();
        return true;
    }
    Close_DB();
    return false;
}
```

- Trả về kết quả thêm tác giả thành công hay không.

- `public bool deleteTG(DTO_QLTG tg)`

```

public bool deleteTG(DTO_QLTG tg)
{
    connect_DB();
    cmd = sqlcon.CreateCommand();
    cmd.CommandText = $"DELETE TblTacGia
WHERE maTG = '{tg.maTG}'";
    int kt = cmd.ExecuteNonQuery();
    if (kt > 0)
    {
        Close_DB();
        return true;
    }
    Close_DB();
    return false;
}

```

- Trả về kết quả xóa tác giả có thành công hay không.

- `public bool editTG(DTO_QLTG tg)`

```

public bool editTG(DTO_QLTG tg)
{
    connect_DB();
    cmd = sqlcon.CreateCommand();
    cmd.CommandText = $"UPDATE TblTacGia SET tenTG = N'{tg.tenTG}', 
diaChi = N'{tg.diaChi}' WHERE maTG = '{tg.maTG}'";
    int kt = cmd.ExecuteNonQuery();
    if (kt > 0)
    {
        Close_DB();
        return true;
    }
    Close_DB();
    return false;
}

```

- Trả về kết quả sửa tác giả có thành công hay không.

- `public DataTable searchMaTG(string maTG)`

```

public DataTable searchMaTG(string maTG)
{
    connect_DB();
    var dtMaTG = new DataTable();
    cmd = sqlcon.CreateCommand();
    cmd.CommandText = $"SELECT maTG AS MA_TAC_GIA, tenTG AS TEN_TAC_GIA,
diaChi AS DIA_CHI FROM TblTacGia WHERE maTG LIKE '%{maTG}%'";
    adt.SelectCommand = cmd;
    dtMaTG.Clear();
    adt.Fill(dtMaTG );
    Close_DB();
    return dtMaTG;
}

```

- Trả về bảng dữ liệu tác giả tìm theo mã.

```

    ■ public DataTable searchTenTG(string tenTG)

public DataTable searchTenTG(string tenTG)
{
    connect_DB();
    var dtTenTG = new DataTable();
    cmd = sqlcon.CreateCommand();
    cmd.CommandText = $"SELECT maTG AS MA_TAC_GIA, tenTG AS TEN_TAC_GIA, diaChi
AS DIA_CHI FROM TblTacGia WHERE tentG LIKE N'{tenTG}%'";
    adt.SelectCommand = cmd;
    dtTenTG.Clear();
    adt.Fill(dtTenTG);
    Close_DB();
    return dtTenTG;
}

```

- Trả về bảng dữ liệu tác giả tìm theo tên.

■ public DataTable showTG()

```

public DataTable showTG()
{
    connect_DB();
    cmd = sqlcon.CreateCommand();
    cmd.CommandText = $"SELECT maTG AS MA_TAC_GIA, tenTG AS TEN_TAC_GIA,
diaChi AS DIA_CHI FROM TblTacGia";
    adt.SelectCommand = cmd;
    dt.Clear();
    adt.Fill(dt);
    Close_DB();
    return dt;
}

```

- Trả về bảng dữ liệu tác giả.

+ Class DAL_QLSach

```
public class DAL_QLSach : Connect_Close_DB, Interface_QLSach
```

○ Các Method của class

■ public bool addSach(DTO_QLSach s)

```

public bool addSach(DTO_QLSach s)
{
    connect_DB();
    cmd = sqlcon.CreateCommand();
    cmd.CommandText = $"INSERT INTO TblSach VALUES ('{s.maS}', 
N'{s.tenS}', {s.soLuongTon}, {s.giaNhap}, {s.giaBan}, '{s.maTL}', '{s.maTG}', 
'{s.maNxb}')";
    int kt = cmd.ExecuteNonQuery();
    if(kt > 0 )
    {
        Close_DB();
        return true;
    }
    Close_DB();
    return false;
}

```

- Trả về kết quả thêm sách có thành công hay không.

■ public bool deleteSach(DTO_QLSach s)

```
public bool deleteSach(DTO_QLSach s)
{
```

```

        connect_DB();
        cmd = sqlcon.CreateCommand();
        cmd.CommandText = $"DELETE TblSach WHERE maSach = '{s.maS}'";
        int kt = cmd.ExecuteNonQuery();
        if (kt > 0)
        {
            Close_DB();
            return true;
        }
        Close_DB();
        return false;
    }
}

```

- Trả về kết quả xóa sách có thành công hay không.

- `public bool editSach(DTO_QLSach s)`

```

public bool editSach(DTO_QLSach s)
{
    connect_DB();
    cmd = sqlcon.CreateCommand();
    cmd.CommandText = $"UPDATE TblSach SET tenSach = N'{s.tenS}', 
soLuongTon = {s.soLuongTon}, giaNhap = {s.giaNhap}, giaBan = {s.giaBan}, maTL =
'{s.maTL}', maTG = '{s.maTG}', maNXB = '{s.maNXB}' WHERE maSach = '{s.maS}'";
    int kt = cmd.ExecuteNonQuery();
    if (kt > 0)
    {
        Close_DB();
        return true;
    }
    Close_DB();
    return false;
}

```

- Trả về kết quả sửa sách có thành công hay không.

- `public string loadMaNXB(string tenNXB)`

```

public string loadMaNXB(string tenNXB)
{
    string tmp = "";
    connect_DB();
    cmd = sqlcon.CreateCommand();
    cmd.CommandText = $"SELECT maNXB FROM TblNhaXuatBan
WHERE tenNXB = N'{tenNXB}'";
    var rd = cmd.ExecuteReader();
    while(rd.Read())
    {
        tmp = rd.GetString(0);
    }
    Close_DB();
    return tmp;
}

```

- Trả về mã nhà xuất bản của tên nhà xuất bản.

- `public string loadMaTG(string tenTG)`

```

public string loadMaTG(string tenTG)
{
    string tmp = "";
    connect_DB();
    cmd = sqlcon.CreateCommand();
    cmd.CommandText = $"SELECT maTG FROM TblTacGia
WHERE tenTG = N'{tenTG}'";
    var rd = cmd.ExecuteReader();
}

```

```

        while (rd.Read())
    {
        tmp = rd.GetString(0);
    }
    Close_DB();
    return tmp;
}

```

- Trả về mã tác giả của tên nhà tác giả.

- `public string loadMaTL(string tenTL)`

```

public string loadMaTL(string tenTL)
{
    string tmp = "";
    connect_DB();
    cmd = sqlcon.CreateCommand();
    cmd.CommandText = $@"SELECT maTL FROM TblTheLoai
WHERE tenTL = N'{tenTL}'";
    var rd = cmd.ExecuteReader();
    while (rd.Read())
    {
        tmp = rd.GetString(0);
    }
    Close_DB();
    return tmp;
}

```

- Trả về mã thể loại của tên thể loại.

- `public DataTable loadTenNXB()`

```

public DataTable loadTenNXB()
{
    connect_DB();
    cmd = sqlcon.CreateCommand();
    cmd.CommandText = $@"SELECT tenNXB FROM TblNhaXuatBan";
    adt.SelectCommand = cmd;
    var dtNXB = new DataTable();
    dtNXB.Clear();
    adt.Fill(dtNXB);
    Close_DB();
    return dtNXB;
}

```

- Trả về bảng dữ liệu tên nhà xuất bản để hiện vào combobox.

- `public DataTable loadTenTG()`

```

public DataTable loadTenTG()
{
    connect_DB();
    cmd = sqlcon.CreateCommand();
    cmd.CommandText = $@"SELECT tenTG FROM TblTacGia";
    adt.SelectCommand = cmd;
    var dtTG = new DataTable();
    dtTG.Clear();
    adt.Fill(dtTG);
    Close_DB();
    return dtTG;
}

```

- Trả về bảng dữ liệu tên tác giả để hiện trong combobox

```

■ public DataTable loadTenTL()

public DataTable loadTenTL()
{
    connect_DB();
    cmd = sqlcon.CreateCommand();
    cmd.CommandText = $"SELECT tenTL FROM TblTheLoai";
    adt.SelectCommand = cmd;
    var dtTL = new DataTable();
    dtTL.Clear();
    adt.Fill(dtTL);
    Close_DB();
    return dtTL;
}

```

- Trả về bảng dữ liệu tên thể loại để hiện trong combobox

```

■ public DataTable searchSach(string maS, string tenS, string tl, string tg)

```

```

public DataTable searchSach(string maS, string tenS, string tl, string tg)
{
    connect_DB();
    cmd = sqlcon.CreateCommand();
    cmd.CommandText = $"SELECT maSach AS MA_SACH, tenSach AS TEN_SACH,
soLuongTon AS SO_LUONG_TON, giaNhap AS GIA_NHAP, giaBan AS GIA_BAN, TL.tenTL AS
TEN_TL, TG.tenTG AS TEN_TG, NXB.tenNXB AS TEN_NXB
    FROM TblSach AS S
    INNER JOIN TblTheLoai AS TL
    ON TL.maTL = S.maTL
    INNER JOIN TblTacGia AS TG
    ON TG.maTG = S.maTG
    INNER JOIN TblNhaXuatBan AS NXB
    ON NXB.maNXB = S.maNXB
    WHERE S.maSach LIKE '{maS}' AND S.tenSach LIKE
N'{tenS}' AND tenTL LIKE N'{tl}' AND tenTG LIKE N'{tg}'";
    adt.SelectCommand = cmd;
    var dtsS = new DataTable();
    dtsS.Clear();
    adt.Fill(dtsS);
    Close_DB();
    return dtsS;
}

```

- Trả về bảng dữ liệu sách theo thông tin được tìm kiếm

```

■ public DataTable showSach()

```

```

public DataTable showSach()
{
    connect_DB();
    cmd = sqlcon.CreateCommand();
    cmd.CommandText = $"SELECT maSach AS MA_SACH, tenSach AS TEN_SACH,
soLuongTon AS SO_LUONG_TON, giaNhap AS GIA_NHAP, giaBan AS GIA_BAN, TL.tenTL AS
TEN_TL, TG.tenTG AS TEN_TG, NXB.tenNXB AS TEN_NXB
    FROM TblSach AS S
    INNER JOIN TblTheLoai AS TL
    ON TL.maTL = S.maTL
    INNER JOIN TblTacGia AS TG
    ON TG.maTG = S.maTG
    INNER JOIN TblNhaXuatBan AS NXB
    ON NXB.maNXB = S.maNXB";
    adt.SelectCommand = cmd;
    dt.Clear();
    adt.Fill(dt);
    Close_DB();
}

```

```
        return dt;
    }
```

- Trả về bảng dữ liệu sách

+ Class DAL_QLPhieuNhap

```
public class DAL_QLPhieuNhap : Connect_Close_DB, Interface_QLPhieuNhap
```

o Các Method của class

▪ `public bool addPN(DTO_QLPhieuNhap pn)`

```
public bool addPN(DTO_QLPhieuNhap pn)
{
    connect_DB();
    cmd = sqlcon.CreateCommand();
    cmd.CommandText = $"@INSERT INTO TblPhieuNhap VALUES
({pn.maPN}', '{pn.ngayNhap}', '{pn.maNxb}', '{pn.maNV}', {pn.tongTien})";
    int kt = cmd.ExecuteNonQuery();
    if(kt > 0)
    {
        Close_DB();
        return true;
    }
    Close_DB();
    return false;
}
```

- Trả về kết quả sau khi thực hiện thêm phiếu nhập.

▪ `public bool deletePN(DTO_QLPhieuNhap pn)`

```
public bool deletePN(DTO_QLPhieuNhap pn)
{
    connect_DB();
    cmd = sqlcon.CreateCommand();
    cmd.CommandText = $"@DELETE TblCTPhieuNhap
WHERE maPN = '{pn.maPN}'";
    int ktr = cmd.ExecuteNonQuery();
    cmd = sqlcon.CreateCommand();
    cmd.CommandText = $"@DELETE TblPhieuNhap
WHERE maPN = '{pn.maPN}'";
    int kt = cmd.ExecuteNonQuery();
    if (kt > 0 && ktr > 0)
    {
        Close_DB();
        return true;
    }
    Close_DB();
    return false;
}
```

- Trả về kết quả sau khi thực hiện xóa phiếu nhập.

▪ `public bool editPN(DTO_QLPhieuNhap pn)`

```
public bool editPN(DTO_QLPhieuNhap pn)
{
    connect_DB();
    cmd = sqlcon.CreateCommand();
    cmd.CommandText = $"@UPDATE TblPhieuNhap SET ngayNhap =
'{pn.ngayNhap}', maNxb = '{pn.maNxb}', maNV = '{pn.maNV}' WHERE maPN =
'{pn.maPN}'";
    int kt = cmd.ExecuteNonQuery();
    if (kt > 0)
```

```

    {
        Close_DB();
        return true;
    }
    Close_DB();
    return false;
}

```

- Trả về kết quả sau khi thực hiện sửa phiếu nhập.

- `public string getMaNV(string tk)`

```

public string getMaNV(string tk)
{
    string tmp = "";
    connect_DB();
    cmd = sqlcon.CreateCommand();
    cmd.CommandText = $"SELECT maNV FROM TblTaiKhoan
WHERE id = '{tk}'";
    var rd = cmd.ExecuteReader();
    while(rd.Read())
    {
        tmp = rd.GetString(0);
    }
    rd.Close();
    Close_DB();
    return tmp;
}

```

- Trả về mã nhân viên của id tài khoản

- `public string getMaNXB(string tenNXB)`

```

public string getMaNXB(string tenNXB)
{
    string tmp = "";
    connect_DB();
    cmd = sqlcon.CreateCommand();
    cmd.CommandText = $"SELECT maNXB FROM TblNhaXuatBan
WHERE tenNXB = '{tenNXB}'";
    var rd = cmd.ExecuteReader();
    while(rd.Read())
    {
        tmp = rd.GetString(0);
    }
    rd.Close();
    Close_DB();
    return tmp;
}

```

- Trả về mã nhà xuất bản của tên nhà xuất bản

- `public DataTable loadTenNXB()`

```

public DataTable loadTenNXB()
{
    var dtTenNXB = new DataTable();
    connect_DB();
    cmd = sqlcon.CreateCommand();
    cmd.CommandText = $"SELECT tenNXB FROM TblNhaXuatBan";
    adt.SelectCommand = cmd;
    dtTenNXB.Clear();
    adt.Fill(dtTenNXB);
    Close_DB();
    return dtTenNXB;
}

```

- Trả về bảng dữ liệu tên nhà xuất bản hiện lên combobox

- `public DataTable searchPN(string maPN)`

```
public DataTable searchPN(string maPN)
{
    var dts = new DataTable();
    connect_DB();
    cmd = sqlcon.CreateCommand();
    cmd.CommandText = $@"SELECT maPN AS MA_PHIEU_NHAP, ngayNhap AS
NGAY_NHAP, NXB.tenNXB AS TEN_NXB, maNV AS MA_NV, tongTien AS TONG_TIEN
FROM TblPhieuNhap AS PN
INNER JOIN TblNhaXuatBan AS NXB
ON NXB.maNXB = PN.maNXB
WHERE PN.maPN LIKE '%{maPN}%'";
    adt.SelectCommand = cmd;
    dts.Clear();
    adt.Fill(dts);
    Close_DB();
    return dts;
}
```

- Trả về bảng dữ liệu phiếu nhập khi tìm bằng mã phiếu nhập.

```
public DataTable showPN()
```

```
public DataTable showPN()
{
    connect_DB();
    cmd = sqlcon.CreateCommand();
    cmd.CommandText = $@"SELECT maPN AS MA_PHIEU_NHAP, ngayNhap AS
NGAY_NHAP, NXB.tenNXB AS TEN_NXB, maNV AS MA_NV, tongTien AS TONG_TIEN
FROM TblPhieuNhap AS PN
INNER JOIN TblNhaXuatBan AS NXB
ON NXB.maNXB = PN.maNXB";
    adt.SelectCommand = cmd;
    dt.Clear();
    adt.Fill(dt);
    Close_DB();
    return dt;
}
```

- Trả về bảng dữ liệu phiếu nhập.

+ Class `DAL_QLNXB`

```
public class DAL_QLNXB : Connect_Close_DB, Interface_QLNXB
```

- o Các Method của class

- `public bool addNXB(DTO_QLNXB nxb)`

```
public bool addNXB(DTO_QLNXB nxb)
{
    connect_DB();
    cmd = sqlcon.CreateCommand();
    cmd.CommandText = $@"INSERT INTO TblNhaXuatBan VALUES
({nxb.maNXB}', '{nxb.tenNXB}', '{nxb.dienThoai}', '{nxb.diaChi}')";
    int kt = cmd.ExecuteNonQuery();
    if(kt > 0)
    {
        Close_DB();
        return true;
    }
    Close_DB();
}
```

```
        return false;
    }
```

- Trả về kết quả sau khi thực hiện thêm nhà xuất bản.

- `public bool deleteNXB(DTO_QLNXB nxb)`

```
public bool deleteNXB(DTO_QLNXB nxb)
{
    connect_DB();
    cmd = sqlcon.CreateCommand();
    cmd.CommandText = $"DELETE TblNhaXuatBan WHERE maNXB = '{nxb.maNXB}'";
    int kt = cmd.ExecuteNonQuery();
    if (kt > 0)
    {
        Close_DB();
        return true;
    }
    Close_DB();
    return false;
}
```

- Trả về kết quả sau khi thực hiện xóa nhà xuất bản.

- `public bool editNXB(DTO_QLNXB nxb)`

```
public bool editNXB(DTO_QLNXB nxb)
{
    connect_DB();
    cmd = sqlcon.CreateCommand();
    cmd.CommandText = $"UPDATE TblNhaXuatBan SET tenNXB =
N'{nxb.tenNXB}', dienThoai = '{nxb.dienThoai}', diaChi = N'{nxb.diaChi}''
WHERE maNXB = '{nxb.maNXB}'";
    int kt = cmd.ExecuteNonQuery();
    if (kt > 0)
    {
        Close_DB();
        return true;
    }
    Close_DB();
    return false;
}
```

- Trả về kết quả sau khi thực hiện sửa nhà xuất bản.

- `public DataTable searchMaNXB(string maNXB)`

```
public DataTable searchMaNXB(string maNXB)
{
    var dtMa = new DataTable();
    connect_DB();
    cmd = sqlcon.CreateCommand();
    cmd.CommandText = $"SELECT maNXB AS MA_NXB, tenNXB AS TEN_NXB, dienThoai
AS DIEN_THOAI, diaChi AS DIA_CHI FROM TblNhaXuatBan WHERE maNXB LIKE '%{maNXB}%'";
    adt.SelectCommand = cmd;
    dtMa.Clear();
    adt.Fill(dtMa);
    Close_DB();
    return dtMa;
}
```

- Trả về bảng dữ liệu nhà xuất bản khi tìm bằng mã nhà xuất bản.

```

    ■ public DataTable searchTenNXB(string tenNXB)

public DataTable searchTenNXB(string tenNXB)
{
    var dtTen = new DataTable();
    connect_DB();
    cmd = sqlcon.CreateCommand();
    cmd.CommandText = $@"SELECT maNXB AS MA_NXB, tenNXB AS TEN_NXB, dienThoai
AS DIEN_THOAI, diaChi AS DIA_CHI FROM TblNhaXuatBan WHERE tenNXB LIKE '%{tenNXB}%'";
    adt.SelectCommand = cmd;
    dtTen.Clear();
    adt.Fill(dtTen);
    Close_DB();
    return dtTen;
}

```

- Trả về bản dữ liệu nhà xuất bản khi tìm bằng tên nhà xuất bản.

```

    ■ public DataTable showNXB()

public DataTable showNXB()
{
    connect_DB();
    cmd = sqlcon.CreateCommand();
    cmd.CommandText = $@"SELECT maNXB AS MA_NXB, tenNXB AS TEN_NXB,
dienThoai AS DIEN_THOAI, diaChi AS DIA_CHI FROM TblNhaXuatBan";
    adt.SelectCommand = cmd;
    dt.Clear();
    adt.Fill(dt);
    Close_DB();
    return dt;
}

```

- Trả về bản dữ liệu nhà xuất bản

+ Class DAL_QLNhanVien

```
public class DAL_QLNhanVien : Connect_Close_DB, Interface_QLNhanVien
```

- Các Method của class

```

    ■ public bool addNV(DTO_QLNhanVien nv)

public bool addNV(DTO_QLNhanVien nv)
{
    connect_DB();
    cmd = sqlcon.CreateCommand();
    cmd.CommandText = $@"INSERT INTO TblNhanVien VALUES ('{nv.maNV}', 
N'{nv.tenNV}', '{nv.ngaySinh}', '{nv.cccd}', '{nv.soDT}', N'{nv.diaChi}', 
N'{nv.lichLam}')";
    int kt = cmd.ExecuteNonQuery();
    if(kt > 0)
    {
        Close_DB();
        return true;
    }
    Close_DB();
    return false;
}

```

- Trả về kết quả khi thêm nhân viên

```

    ■ public bool deleteNV(DTO_QLNhanVien nv)

public bool deleteNV(DTO_QLNhanVien nv)
{
    connect_DB();
    cmd = sqlcon.CreateCommand();
    cmd.CommandText = $"DELETE TblNhanVien WHERE maNV = '{nv.maNV}'";
    int kt = cmd.ExecuteNonQuery();
    if (kt > 0)
    {
        Close_DB();
        return true;
    }
    Close_DB();
    return false;
}

```

- Trả về kết quả khi xóa nhân viên

```

    ■ public bool editNV(DTO_QLNhanVien nv)

public bool editNV(DTO_QLNhanVien nv)
{
    connect_DB();
    cmd = sqlcon.CreateCommand();
    cmd.CommandText = $"UPDATE TblNhanVien SET tenNV = N'{nv.tenNV}', ngaySinh =
'{nv.ngaySinh}', cCCD = '{nv.cccd}', soDT = '{nv.soDT}', diaChi = N'{nv.diaChi}', lichLam
= N'{nv.lichLam}' WHERE maNV = '{nv.maNV}'";
    int kt = cmd.ExecuteNonQuery();
    if (kt > 0)
    {
        Close_DB();
        return true;
    }
    Close_DB();
    return false;
}

```

- Trả về kết quả khi sửa nhân viên

■ public DataTable searchMaNV(string maNV)

```

public DataTable searchMaNV(string maNV)
{
    connect_DB();
    cmd = sqlcon.CreateCommand();
    cmd.CommandText = $"SELECT * FROM TblNhanVien WHERE
maNV LIKE '%{maNV}%' AND (maNV LIKE '%QL%' OR maNV LIKE '%NV%')");
    adt.SelectCommand = cmd;
    var dtMaNV = new DataTable();
    dtMaNV.Clear();
    adt.Fill(dtMaNV);
    Close_DB();
    return dtMaNV;
}

```

- Trả về bảng dữ liệu nhân viên khi tìm bằng mã nhân viên.

■ public DataTable searchTenNV(string tenNV)

```

public DataTable searchTenNV(string tenNV)
{
    connect_DB();
    cmd = sqlcon.CreateCommand();
    cmd.CommandText = $"SELECT * FROM TblNhanVien WHERE tenNV LIKE
N'%{tenNV}%' AND (maNV LIKE '%QL%' OR maNV LIKE '%NV%')");
}

```

```

        adt.SelectCommand = cmd;
        var dtTenNV = new DataTable();
        dtTenNV.Clear();
        adt.Fill(dtTenNV);
        Close_DB();
        return dtTenNV;
    }

```

- Trả về bảng dữ liệu nhân viên khi tìm bằng tên nhân viên.

- `public DataTable ShowNV()`

```

public DataTable ShowNV()
{
    connect_DB();
    cmd = sqlcon.CreateCommand();
    cmd.CommandText = $"SELECT * FROM TblNhanVien WHERE
maNV LIKE '{QL}' OR maNV LIKE '{NV}%'";
    adt.SelectCommand = cmd;
    dt.Clear();
    adt.Fill(dt);
    Close_DB();
    return dt;
}

```

- Trả về bảng dữ liệu nhân viên

+ Class `DAL_QLNguoiDung`

```
public class DAL_QLNguoiDung : Connect_Close_DB, Interface_QLNguoiDung
```

- Các Method của class

- `public DataTable showUser()`

```

public DataTable showUser()
{
    connect_DB();
    cmd = sqlcon.CreateCommand();
    cmd.CommandText = @"SELECT * FROM TblTaiKhoan";
    adt.SelectCommand = cmd;
    dt.Clear();
    adt.Fill(dt);
    Close_DB();
    return dt;
}

```

- Trả về bảng dữ liệu người dùng.

- `public bool addUser(DTO_QLNguoiDung nd)`

```

public bool addUser(DTO_QLNguoiDung nd)
{
    connect_DB();
    cmd = sqlcon.CreateCommand();
    cmd.CommandText = $"INSERT INTO TblTaiKhoan VALUES
({nd.id}, {nd.pw}, {nd.chucVu}, '{nd.maNV}');
    int kt = cmd.ExecuteNonQuery();
    if(kt > 0)
    {
        Close_DB();
        return true;
    }
    Close_DB();
    return false;
}

```

```
}
```

- Trả về kết quả khi thêm người dùng.

- `public bool editUser(DTO_QLNguoiDung nd)`

```
public bool editUser(DTO_QLNguoiDung nd)
{
    connect_DB();
    cmd = sqlcon.CreateCommand();
    cmd.CommandText = $@"UPDATE TblTaiKhoan SET pw = N'{nd.pW}', 
chucVu = N'{nd.chucVu}', maNV = '{nd.maNV}' WHERE id = '{nd.iD}'";
    int kt = cmd.ExecuteNonQuery();
    if (kt > 0)
    {
        Close_DB();
        return true;
    }
    Close_DB();
    return false;
}
```

- Trả về kết quả khi sửa người dùng.

- `public bool deleteUser(DTO_QLNguoiDung nd)`

```
public bool deleteUser(DTO_QLNguoiDung nd)
{
    connect_DB();
    cmd = sqlcon.CreateCommand();
    cmd.CommandText = $@"DELETE TblTaiKhoan WHERE id = '{nd.iD}'";
    int kt = cmd.ExecuteNonQuery();
    if (kt > 0)
    {
        Close_DB();
        return true;
    }
    Close_DB();
    return false;
}
```

- Trả về kết quả khi xóa người dùng.

- `public DataTable loadTenNV()`

```
public DataTable loadTenNV()
{
    connect_DB();
    cmd = sqlcon.CreateCommand();
    cmd.CommandText = @"SELECT tenNV FROM TblNhanVien";
    adt.SelectCommand = cmd;
    var dtTenNV = new DataTable();
    dtTenNV.Clear();
    adt.Fill(dtTenNV);
    Close_DB();
    return dtTenNV;
}
```

- Trả về bảng dữ liệu tên nhân viên

- `public string loadMaNV(string tenNV)`

```
public string loadMaNV(string tenNV)
{
    connect_DB();
    string tmp = "";
    cmd = sqlcon.CreateCommand();
```

```

cmd.CommandText = $"SELECT maNV FROM TblNhanVien WHERE
tenNV = N'{tenNV}'";
var rd = cmd.ExecuteReader();
while(rd.Read())
{
    tmp = rd.GetString(0);
}
rd.Close();
Close_DB();
return tmp;
}

```

- Trả về mã nhân viên của tên nhân viên.

+ Class DAL_QLBanSach

```
public class DAL_QLBanSach : Connect_Close_DB, Interface_QLBanSach
```

o Các Method của class

- `public bool addHD(DTO_QLBanSach hd)`

```

public bool addHD(DTO_QLBanSach hd)
{
    connect_DB();
    cmd = sqlcon.CreateCommand();
    cmd.CommandText = $"INSERT INTO TblHoaDon VALUES ('{hd.maHD}', '{hd.ngayBan}', '{hd.maNV}', {hd.tongTien})";
    int kt = cmd.ExecuteNonQuery();
    if(kt > 0)
    {
        Close_DB();
        return true;
    }
    Close_DB();
    return false;
}

```

- Trả về kết quả khi thêm hóa đơn

- `public bool deleteHD(DTO_QLBanSach hd)`

```

public bool deleteHD(DTO_QLBanSach hd)
{
    connect_DB();
    cmd = sqlcon.CreateCommand();
    cmd.CommandText = $"DELETE TblCTHoaDon WHERE maHD = '{hd.maHD}'";
    int ktr = cmd.ExecuteNonQuery();
    cmd = sqlcon.CreateCommand();
    cmd.CommandText = $"DELETE TblHoaDon WHERE maHD = '{hd.maHD}'";
    int kt = cmd.ExecuteNonQuery();
    if (kt > 0 && ktr > 0)
    {
        Close_DB();
        return true;
    }
    Close_DB();
    return false;
}

```

- Trả về kết quả khi xóa hóa đơn

```

    ■ public bool editHD(DTO_QLBanSach hd)

public bool editHD(DTO_QLBanSach hd)
{
    connect_DB();
    cmd = sqlcon.CreateCommand();
    cmd.CommandText = $@"UPDATE TblHoaDon SET ngayBan = '{hd.ngayBan}',
maNV = '{hd.maNV}', tongTien = {hd.tongTien} WHERE maHD = '{hd.maHD}'";
    int kt = cmd.ExecuteNonQuery();
    if (kt > 0)
    {
        Close_DB();
        return true;
    }
    Close_DB();
    return false;
}

```

- Trả về kết quả khi sửa hóa đơn.

```

    ■ public string getMaNV(string tk)

public string getMaNV(string tk)
{
    string tmp = "";
    connect_DB();
    cmd = sqlcon.CreateCommand();
    cmd.CommandText = $@"SELECT maNV FROM TblTaiKhoan WHERE id = '{tk}'";
    var rd = cmd.ExecuteReader();
    while (rd.Read())
    {
        tmp = rd.GetString(0);
    }
    rd.Close();
    Close_DB();
    return tmp;
}

```

- Trả về mã nhân viên của id tài khoản

```

    ■ public DataTable searchHD(string maHD)

public DataTable searchHD(string maHD)
{
    connect_DB();
    cmd = sqlcon.CreateCommand();
    cmd.CommandText = $@"SELECT maHD AS MA_HOA_DON, ngayBan AS NGAY_BAN,
maNV AS MA_NHAN_VIEN, tongTien AS TONG_TIEN FROM TblHoaDon
WHERE maHD LIKE '%{maHD}%'";
    adt.SelectCommand = cmd;
    var dts = new DataTable();
    dts.Clear();
    adt.Fill(dts);
    Close_DB();
    return dts;
}

```

- Trả về bảng dữ liệu khi tìm bằng mã hóa đơn.

```

    ■ public DataTable showHD()

public DataTable showHD()
{
    connect_DB();
    cmd = sqlcon.CreateCommand();
}

```

```

        cmd.CommandText = $"SELECT maHD AS MA_HOA_DON, ngayBan AS NGAY_BAN,
maNV AS MA_NHAN_VIEN, tongTien AS TONG_TIEN FROM TblHoaDon";
        adt.SelectCommand = cmd;
        dt.Clear();
        adt.Fill(dt);
        Close_DB();
        return dt;
    }
}

```

- Trả về bảng dữ liệu hóa đơn.

- `public DataTable inHD(DTO_QLBanSach hd)`

```

public DataTable inHD(DTO_QLBanSach hd)
{
    connect_DB();
    cmd = sqlcon.CreateCommand();
    cmd.CommandText = $"SELECT TblSach.tenSach,
SUM(TblCTHoaDon.soLuong), TblSach.giaBan, SUM(TblCTHoaDon.thanhTien)
FROM TblCTHoaDon
INNER JOIN TblHoaDon ON TblCTHoaDon.maHD = TblHoaDon.maHD
INNER JOIN TblSach ON TblCTHoaDon.maSach = TblSach.maSach
WHERE TblHoaDon.maHD = '{hd.maHD}'
GROUP BY tenSach, giaBan";
    adt.SelectCommand = cmd;
    var inHD = new DataTable();
    inHD.Clear();
    adt.Fill(inHD);
    Close_DB();
    return inHD;
}

```

- Trả về bảng dữ liệu các thông tin cần để in hóa đơn.

+ Class `DAL_CTPhieuNhap`

```
public class DAL_CTPhieuNhap : Connect_Close_DB, Interface_CTPhieuNhap
```

○ Các Method của class

- `public bool addS(DTO_CTPhieuNhap ctpn)`

```

public bool addS(DTO_CTPhieuNhap ctpn)
{
    int tmp = 0;
    connect_DB();
    cmd = sqlcon.CreateCommand();
    cmd.CommandText = $"SELECT soLuongTon FROM TblSach WHERE maSach =
'{ctpn.maS}'";
    var rd = cmd.ExecuteReader();
    while (rd.Read())
    {
        tmp = rd.GetInt32(0);
    }
    rd.Close();

    cmd = sqlcon.CreateCommand();
    cmd.CommandText = $"UPDATE TblSach SET soLuongTon = {tmp + ctpn.soLuong}
WHERE maSach = '{ctpn.maS}'";
    int ktr = cmd.ExecuteNonQuery();

    cmd = sqlcon.CreateCommand();
    cmd.CommandText = $"INSERT INTO TblCTPhieuNhap VALUES ('{ctpn.maPN}', 
'{ctpn.maS}', {ctpn.soLuong}, {ctpn.giaNhap})";
    int kt = cmd.ExecuteNonQuery();
}

```

```

        if(kt > 0 && ktr > 0)
        {
            Close_DB();
            return true;
        }
        Close_DB();
        return false;
    }
}

```

- Trả về kết quả khi thêm sách vào chi tiết phiếu nhập

- `public bool editS(DTO_CTPhieuNhap ctpn, int slf)`

```

public bool editS(DTO_CTPhieuNhap ctpn, int slf)
{
    connect_DB();
    int tmp = 0;
    cmd = sqlcon.CreateCommand();
    cmd.CommandText = $"SELECT soLuongTon FROM TblSach WHERE maSach =
'{ctpn.maS}'";
    var rd = cmd.ExecuteReader();
    while (rd.Read())
    {
        tmp = rd.GetInt32(0);
    }
    rd.Close();

    cmd = sqlcon.CreateCommand();
    cmd.CommandText = $"UPDATE TblSach SET soLuongTon = {tmp - slf +
ctpn.soLuong} WHERE maSach = '{ctpn.maS}'";
    int ktr = cmd.ExecuteNonQuery();

    cmd = sqlcon.CreateCommand();
    cmd.CommandText = $"UPDATE TblCTPhieuNhap SET soLuongNhap =
{ctpn.soLuong}, giaNhap = {ctpn.giaNhap} WHERE maPN = '{ctpn.maPN}' AND maSach
= '{ctpn.maS}'";
    int kt = cmd.ExecuteNonQuery();
    if (kt > 0 && ktr > 0)
    {
        return true;
    }
    Close_DB();
    return false;
}

```

- Trả về kết quả khi sửa sách trong chi tiết phiếu nhập

- `public string getGiaNhap(int sl, string maS)`

```

public string getGiaNhap(int sl, string maS)
{
    connect_DB();
    int giaNhap = 0;
    cmd = sqlcon.CreateCommand();
    cmd.CommandText = $"SELECT giaNhap FROM TblSach WHERE maSach = '{maS}'";
    var rdgn = cmd.ExecuteReader();
    while(rdgn.Read())
    {
        giaNhap = rdgn.GetInt32(0);
    }
    rdgn.Close();
    Close_DB();
    return Convert.ToString(sl*giaNhap);
}

```

- Trả về giá tiền của số lượng * giá sản phẩm.

```

    ■ public bool deleteS(DTO_CTPHieuNhap ctpn)

public bool deleteS(DTO_CTPHieuNhap ctpn)
{
    connect_DB();
    int tmp = 0;
    cmd = sqlcon.CreateCommand();
    cmd.CommandText = $"SELECT soLuongTon FROM TblSach WHERE maSach
= '{ctpn.maS}'";
    var rd = cmd.ExecuteReader();
    while (rd.Read())
    {
        tmp = rd.GetInt32(0);
    }
    rd.Close();

    cmd = sqlcon.CreateCommand();
    cmd.CommandText = $"SELECT SUM(soLuongNhap) FROM TblCTPHieuNhap
WHERE maPN = '{ctpn.maPN}' AND maSach = '{ctpn.maS}'";
    var rds = cmd.ExecuteReader();
    while(rds.Read())
    {
        ctpn.soLuong = rds.GetInt32(0);
    }
    rds.Close();

    cmd = sqlcon.CreateCommand();
    cmd.CommandText = $"UPDATE TblSach SET soLuongTon = {tmp -
ctpn.soLuong} WHERE maSach = '{ctpn.maS}'";
    int ktr = cmd.ExecuteNonQuery();

    cmd = sqlcon.CreateCommand();
    cmd.CommandText = $"DELETE TblCTPHieuNhap WHERE maPN =
'{ctpn.maPN}' AND maSach = '{ctpn.maS}'";
    int kt = cmd.ExecuteNonQuery();
    if (kt > 0 && ktr > 0)
    {
        Close_DB();
        return true;
    }
    Close_DB();
    return false;
}

```

- Trả về kết quả khi xóa sách trong chi tiết phiếu nhập

```

    ■ public DataTable searchTenS(string maPN, string tenS)

public DataTable searchTenS(string maPN, string tenS)
{
    connect_DB();
    cmd = sqlcon.CreateCommand();
    cmd.CommandText = $"SELECT S.maSach AS MA_SACH, S.tenSach AS TEN_SACH,
S.soLuongTon AS SO_LUONG_TON, giaNhap AS GIA_NHAP, giaBan AS GIA_BAN FROM TblSach AS S
INNER JOIN TblNhaXuatBan AS NXB
ON NXB.maNXB = S.maNXB
INNER JOIN TblPhieuNhap AS PN
ON PN.maNXB = NXB.maNXB
WHERE PN.maPN = '{maPN}' AND S.tenSach LIKE N'%{tenS}%'";
    adt.SelectCommand = cmd;
    dtS.Clear();
    adt.Fill(dtS);
    Close_DB();
    return dtS;
}

```

- Trả về bảng dữ liệu sách khi tìm kiếm bằng tên sách.

- `public DataTable showCTPN(string maPN)`

```
public DataTable showCTPN(string maPN)
{
    connect_DB();
    cmd = sqlcon.CreateCommand();
    cmd.CommandText = $@"SELECT maPN AS MA_PHIEU_NHAP, maSach AS MA_SACH,
soLuongNhap AS SO_LUONG_NHAP, giaNhap AS GIA_NHAP
                    FROM TblCTPhieuNhap
                    WHERE maPN = '{maPN}'";
    adt.SelectCommand = cmd;
    dtCTPN.Clear();
    adt.Fill(dtCTPN);
    Close_DB();
    return dtCTPN;
}
```

- Trả về bảng dữ liệu chi tiết phiếu nhập

- `public DataTable showSach(string maPN)`

```
public DataTable showSach(string maPN)
{
    connect_DB();
    cmd = sqlcon.CreateCommand();
    cmd.CommandText = $@"SELECT S.maSach AS MA_SACH, S.tenSach AS TEN_SACH,
S.soLuongTon AS SO_LUONG_TON, giaNhap AS GIA_NHAP, giaBan AS GIA_BAN FROM TblSach AS S
INNER JOIN TblNhaXuatBan AS NXB
ON NXB.maNXB = S.maNXB
INNER JOIN TblPhieuNhap AS PN
ON PN.maNXB = NXB.maNXB
WHERE PN.maPN = '{maPN}'";
    adt.SelectCommand = cmd;
    dtS.Clear();
    adt.Fill(dtS);
    Close_DB();
    return dtS;
}
```

- Trả về bảng dữ liệu sách

- `public bool updateTongPN(DTO_CTPhieuNhap ctpn)`

```
public bool updateTongPN(DTO_CTPhieuNhap ctpn)
{
    long tmp = 0;
    connect_DB();
    cmd = sqlcon.CreateCommand();
    cmd.CommandText = $@"SELECT SUM(giaNhap) FROM TblCTPhieuNhap
WHERE maPN = '{ctpn.maPN}'";
    var rd = cmd.ExecuteReader();
    while (rd.Read())
    {
        tmp = rd.GetInt64(0);
    }
    rd.Close();
    cmd = sqlcon.CreateCommand();
    cmd.CommandText = $@"UPDATE TblPhieuNhap SET tongTien = {tmp}
WHERE maPN = '{ctpn.maPN}'";
    int kt = cmd.ExecuteNonQuery();
    if(kt > 0)
    {
        Close_DB();
    }
}
```

```

        return true;
    }
    Close_DB();
    return false;
}

```

- Trả về kết quả khi sửa tổng tiền phiếu nhập

- `public bool updateTongPN(string maPN, int tmp)`

```

public bool updateTongPN(string maPN, int tmp)
{
    connect_DB();
    cmd = sqlcon.CreateCommand();
    cmd.CommandText = $"UPDATE TblPhieuNhap SET tongTien = {tmp}"
WHERE maPN = '{maPN}'";
    int kt = cmd.ExecuteNonQuery();
    if (kt > 0)
    {
        Close_DB();
        return true;
    }
    Close_DB();
    return false;
}

```

- Trả về kết quả khi sửa tổng tiền phiếu nhập

+ Class `DAL_CTHoaDon`

```
public class DAL_CTHoaDon : Connect_Close_DB, Interface_CTHoaDon
```

- Các Method của class

- `public bool addS(DTO_CTHoaDon cthd)`

```

public bool addS(DTO_CTHoaDon cthd)
{
    int tmp = 0;
    connect_DB();
    cmd = sqlcon.CreateCommand();
    cmd.CommandText = $"SELECT soLuongTon FROM TblSach WHERE maSach =
'{cthd.maS}'";
    var rd = cmd.ExecuteReader();
    while (rd.Read())
    {
        tmp = rd.GetInt32(0);
    }
    rd.Close();

    cmd = sqlcon.CreateCommand();
    cmd.CommandText = $"UPDATE TblSach SET soLuongTon = {tmp - cthd.sL}
WHERE maSach = '{cthd.maS}'";
    int ktr = cmd.ExecuteNonQuery();

    cmd = sqlcon.CreateCommand();
    cmd.CommandText = $"INSERT INTO TblCTHoaDon VALUES ('{cthd.maHD}', '{cthd.maS}', {cthd.sL}, {cthd.giaTien})";
    int kt = cmd.ExecuteNonQuery();
    if (kt > 0 && ktr > 0)
    {
        Close_DB();
        return true;
    }
    return false;
}

```

```
}
```

- Trả về kết quả khi thêm sách vào chi tiết hóa đơn.

- `public bool deleteS(DTO_CTHoaDon cthd)`

```
public bool deleteS(DTO_CTHoaDon cthd)
{
    int tmp = 0;
    connect_DB();
    cmd = sqlcon.CreateCommand();
    cmd.CommandText = $"SELECT soLuongTon FROM TblSach WHERE maSach =
'{cthd.maS}'";
    var rd = cmd.ExecuteReader();
    while (rd.Read())
    {
        tmp = rd.GetInt32(0);
    }
    rd.Close();

    cmd = sqlcon.CreateCommand();
    cmd.CommandText = $"UPDATE TblSach SET soLuongTon = {tmp + cthd.sL}
WHERE maSach = '{cthd.maS}'";
    int ktr = cmd.ExecuteNonQuery();

    cmd = sqlcon.CreateCommand();
    cmd.CommandText = $"DELETE TblCTHoaDon WHERE maHD = '{cthd.maHD}' and
maSach = '{cthd.maS}'";
    int kt = cmd.ExecuteNonQuery();
    if (kt > 0 && ktr > 0)
    {
        Close_DB();
        return true;
    }
    return false;
}
```

- Trả về kết quả khi xóa sách trong chi tiết hóa đơn

- `public string getGiaBan(int sl, string maS)`

```
public string getGiaBan(int sl, string maS)
{
    connect_DB();
    int giaBan = 0;
    cmd = sqlcon.CreateCommand();
    cmd.CommandText = $"SELECT giaBan FROM TblSach WHERE maSach = '{maS}'";
    var rdgn = cmd.ExecuteReader();
    while (rdgn.Read())
    {
        giaBan = rdgn.GetInt32(0);
    }
    rdgn.Close();
    Close_DB();
    return Convert.ToString(sl * giaBan);
}
```

- Trả về giá bán * số lượng của sách

```

    ■ public DataTable searchTenS(string tenS)

public DataTable searchTenS(string tenS)
{
    connect_DB();
    cmd = sqlcon.CreateCommand();
    cmd.CommandText = $"SELECT S.maSach AS MA_SACH, S.tenSach AS TEN_SACH,
S.soLuongTon AS SO_LUONG_TON, gianhap AS GIA_NHAP, giaBan AS GIA_BAN FROM TblSach AS S
WHERE S.tenSach LIKE N'%{tenS}%'";
    adt.SelectCommand = cmd;
    dtS.Clear();
    adt.Fill(dtS);
    Close_DB();
    return dtS;
}

```

- Trả về bảng dữ liệu sách khi tìm bằng tên sách

```

    ■ public DataTable showCTHD(string maHD)

public DataTable showCTHD(string maHD)
{
    connect_DB();
    cmd = sqlcon.CreateCommand();
    cmd.CommandText = $"SELECT maHD AS MA_HOA_DON, CTHD.maSach AS MA_SACH,
SUM(soLuong) AS SO_LUONG, S.giaBan AS GIA_BAN, SUM(CTHD.thanhTien) AS THANH_TIEN FROM
TblCTHoaDon AS CTHD
INNER JOIN TblSach AS S
ON S.maSach = CTHD.maSach
WHERE CTHD.maHD = '{maHD}'
GROUP BY maHD, CTHD.maSach, S.giaBan";
    adt.SelectCommand = cmd;
    dtcthd.Clear();
    adt.Fill(dtcthd);
    Close_DB();
    return dtcthd;
}

```

- Trả về bảng dữ liệu chi tiết hóa đơn

```

    ■ public DataTable showSach()

public DataTable showSach()
{
    connect_DB();
    cmd = sqlcon.CreateCommand();
    cmd.CommandText = $"SELECT S.maSach AS MA_SACH, S.tenSach AS TEN_SACH,
S.soLuongTon AS SO_LUONG_TON, gianhap AS GIA_NHAP, giaBan AS GIA_BAN FROM TblSach
AS S";
    adt.SelectCommand = cmd;
    dtS.Clear();
    adt.Fill(dtS);
    Close_DB();
    return dtS;
}

```

- Trả về bảng dữ liệu sách.

```

    ■ public bool updateTongHD(DTO_CTHoaDon cthd)

public bool updateTongHD(DTO_CTHoaDon cthd)
{
    long tmp = 0;
    connect_DB();
    cmd = sqlcon.CreateCommand();
    cmd.CommandText = $"SELECT SUM(thanhTien) FROM TblCTHoaDon
WHERE maHD = '{cthdd.maHD}'";
    var rd = cmd.ExecuteReader();
    while (rd.Read())
    {
        tmp = rd.GetInt64(0);
    }
    rd.Close();
    cmd = sqlcon.CreateCommand();
    cmd.CommandText = $"UPDATE TblHoaDon SET tongTien = {tmp}
WHERE maHD = '{cthdd.maHD}'";
    int kt = cmd.ExecuteNonQuery();
    if (kt > 0)
    {
        Close_DB();
        return true;
    }
    return false;
}

```

- Trả về kết quả khi sửa tổng tiền hóa đơn

```

    ■ public bool updateTongHD(string maHD, int tmp)

public bool updateTongHD(string maHD, int tmp)
{
    connect_DB();
    cmd = sqlcon.CreateCommand();
    cmd.CommandText = $"UPDATE TblHoaDon SET tongTien = {tmp}
WHERE maHD = '{maHD}'";
    int kt = cmd.ExecuteNonQuery();
    if (kt > 0)
    {
        Close_DB();
        return true;
    }
    return false;
}

```

- Trả về kết quả khi sửa tổng tiền hóa đơn

+ Class [DAL_BC_TK](#)

```

public class DAL_BC_TK : Connect_Close_DB, Interface_BC_TK

```

- Các method của class

```

    ■ public DataTable inCTPN(string thang, string nam, string nxb)

public DataTable inCTPN(string thang, string nam, string nxb)
{
    connect_DB();
    cmd = sqlcon.CreateCommand();
    cmd.CommandText = $"SELECT TblPhieuNhap.maPN, TblSach.tenSach,
TblPhieuNhap.ngayNhap, TblCTPhieuNhap.soLuongNhap, TblSach.giaNhap,
TblCTPhieuNhap.giaNhap AS Expr1, TblNhaXuatBan.tenNXB
FROM TblCTPhieuNhap INNER JOIN TblSach ON TblSach.maSach = TblCTPhieuNhap.maSach
INNER JOIN TblPhieuNhap ON TblPhieuNhap.maPN = TblCTPhieuNhap.maPN

```

```

    INNER JOIN TblNhaXuatBan ON TblNhaXuatBan.maNXB = TblSach.maNXB AND TblNhaXuatBan.maNXB =
    TblPhieuNhap.maNXB
    WHERE MONTH(ngayNhap) LIKE '%{thang}%' AND YEAR(ngayNhap) LIKE '%{nam}%' AND tenNXB LIKE
    '%{nxb}%' ;
        adt.SelectCommand = cmd;
        dtInCTPN.Clear();
        adt.Fill(dtInCTPN);
        Close_DB();
        return dtInCTPN;
    }
}

```

- Trả về bảng dữ liệu để thống kê chi tiết phiếu nhập

- `public DataTable inPN(string thang, string nam, string nxb)`

```

public DataTable inPN(string thang, string nam, string nxb)
{
    connect_DB();
    cmd = sqlcon.CreateCommand();
    cmd.CommandText = $@"SELECT maPN, ngayNhap, tenNXB, tenNV, tongTien
FROM TblPhieuNhap
INNER JOIN TblNhaXuatBan
ON TblNhaXuatBan.maNXB = TblPhieuNhap.maNXB
INNER JOIN TblNhanVien
ON TblNhanVien.maNV = TblPhieuNhap.maNV
WHERE MONTH(ngayNhap) LIKE '{thang}' AND YEAR(ngayNhap) LIKE '{nam}' AND tenNXB
LIKE '{nxb}' ;
        adt.SelectCommand = cmd;
        dtInPN.Clear();
        adt.Fill(dtInPN);
        Close_DB();
        return dtInPN;
}

```

- Trả về bảng dữ liệu để thống kê phiếu nhập

- `public DataTable showDSPN()`

```

public DataTable showDSPN()
{
    connect_DB();
    cmd = sqlcon.CreateCommand();
    cmd.CommandText = $@"SELECT maPN AS MÃ_PHIẾU_NHẬP, ngayNhap AS NGÀY_NHẬP,
tenNXB AS TÊN_NHÀ_XUẤT_BẢN, tenNV AS NGƯỜI_LẬP_PHIẾU, tongTien AS TỔNG_TIỀN
FROM TblPhieuNhap
INNER JOIN TblNhaXuatBan
ON TblNhaXuatBan.maNXB = TblPhieuNhap.maNXB
INNER JOIN TblNhanVien
ON TblNhanVien.maNV = TblPhieuNhap.maNV";
        adt.SelectCommand = cmd;
        dtshowPN.Clear();
        adt.Fill(dtshowPN);
        Close_DB();
        return dtshowPN;
}

```

- Trả về bảng dữ liệu phiếu nhập

- `public DataTable showCTDSPN(string maPN)`

```

public DataTable showCTDSPN(string maPN)
{
    connect_DB();
    cmd = sqlcon.CreateCommand();
    cmd.CommandText = $@"SELECT tenSach AS TÊN_SÁCH, soLuongNhap AS
SỐ_LƯỢNG, TblSach.giaNhap AS GIÁ_NHẬP, TblCTPhieuNhap.giaNhap AS THANH_TIỀN

```

```

FROM TblCTPhieuNhap
INNER JOIN TblSach ON TblSach.maSach = TblCTPhieuNhap.maSach
WHERE maPN = '{maPN}';
adt.SelectCommand = cmd;
dtshowCTPN.Clear();
adt.Fill(dtshowCTPN);
Close_DB();
return dtshowCTPN;
}

```

- Trả về bảng dữ liệu chi tiết phiếu nhập.

- `public DataTable LoadTenNXB()`

```

public DataTable LoadTenNXB()
{
    var nxb = new DataTable();
    connect_DB();
    cmd = sqlcon.CreateCommand();
    cmd.CommandText = $@"SELECT tenNXB FROM TblNhaXuatBan";
    adt.SelectCommand = cmd;
    nxb.Clear();
    adt.Fill(nxb);
    Close_DB();
    return nxb;
}

```

- Trả về bảng dữ liệu tên nhà xuất bản.

- `public DataTable showDSHD()`

```

public DataTable showDSHD()
{
    connect_DB();
    cmd = sqlcon.CreateCommand();
    cmd.CommandText = $@"SELECT maHD AS MÃ_HÓA ĐƠN, ngayBan as NGÀY_BÁN,
tenNV AS NHÂN VIÊN_LẬP, tongTien AS TỔNG TIỀN
FROM TblHoaDon
INNER JOIN TblNhanVien ON TblNhanVien.maNV = TblHoaDon.maNV";
    adt.SelectCommand = cmd;
    dtshowHD.Clear();
    adt.Fill(dtshowHD);
    Close_DB();
    return dtshowHD;
}

```

- Trả về bảng dữ liệu hóa đơn.

- `public DataTable showCTDSHD(string maHD)`

```

public DataTable showCTDSHD(string maHD)
{
    connect_DB();
    cmd = sqlcon.CreateCommand();
    cmd.CommandText = $@"SELECT TblHoaDon.maHD AS MÃ_HÓA ĐƠN, tenSach AS
TÊN_SÁCH, SUM(soLuong) AS SỐ_LƯỢNG, giaBan AS GIÁ_BÁN, SUM(thanhTien) AS THÀNH TIỀN
FROM TblCTHoaDon
INNER JOIN TblSach ON TblSach.maSach = TblCTHoaDon.maSach
INNER JOIN TblHoaDon ON TblHoaDon.maHD = TblCTHoaDon.maHD
WHERE TblHoaDon.maHD = '{maHD}'
GROUP BY TblHoaDon.maHD, tenSach, giaBan";
    adt.SelectCommand = cmd;
    dtshowCTHD.Clear();
    adt.Fill(dtshowCTHD);
    Close_DB();
    return dtshowCTHD;
}

```

```
}
```

- Trả về bảng dữ liệu chi tiết hóa đơn

- `public DataTable inHD(string thang, string nam)`

```
public DataTable inHD(string thang, string nam)
{
    connect_DB();
    cmd = sqlcon.CreateCommand();
    cmd.CommandText = $@"SELECT maHD AS MÃ_HÓA ĐƠN, ngayBan as NGÀY_BÁN,
tenNV AS NHÂN VIÊN_LẬP, tongTien AS TỔNG TIỀN
FROM TblHoaDon
INNER JOIN TblNhanVien ON TblNhanVien.maNV = TblHoaDon.maNV
WHERE MONTH(ngayBan) LIKE '{thang}' AND YEAR(ngayBan) LIKE '{nam}';
    adt.SelectCommand = cmd;
    dtInHD.Clear();
    adt.Fill(dtInHD);
    Close_DB();
    return dtInHD;
}
```

- Trả về bảng dữ liệu để thống kê hóa đơn

- `public DataTable inCTHD(string thang, string nam)`

```
public DataTable inCTHD(string thang, string nam)
{
    connect_DB();
    cmd = sqlcon.CreateCommand();
    cmd.CommandText = $@"SELECT TblHoaDon.maHD, tenSach AS TÊN_SÁCH,
SUM(soLuong) AS SỐ_LƯỢNG, giaBan AS GIÁ_BÁN, SUM(thanhTien) AS THÀNH TIỀN
FROM TblCTHoaDon
INNER JOIN TblSach ON TblSach.maSach = TblCTHoaDon.maSach
INNER JOIN TblHoaDon ON TblHoaDon.maHD = TblCTHoaDon.maHD
WHERE MONTH(ngayBan) LIKE '{thang}' AND YEAR(ngayBan) LIKE '{nam}'
GROUP BY TblHoaDon.maHD, tenSach, giaBan";
    adt.SelectCommand = cmd;
    dtInCTHD.Clear();
    adt.Fill(dtInCTHD);
    Close_DB();
    return dtInCTHD;
}
```

- Trả về bảng dữ liệu để thống kê chi tiết hóa đơn.

- `public DataTable LoadTenTG()`

```
public DataTable LoadTenTG()
{
    var tg = new DataTable();
    connect_DB();
    cmd = sqlcon.CreateCommand();
    cmd.CommandText = $@"SELECT tenTG FROM TblTacGia";
    adt.SelectCommand = cmd;
    tg.Clear();
    adt.Fill(tg);
    Close_DB();
    return tg;
}
```

- Trả về bảng dữ liệu tên tác giả.

```

    ■ public DataTable inTKSL(string tenNXB, string tenTG)

public DataTable inTKSL(string tenNXB, string tenTG)
{
    connect_DB();
    cmd = sqlcon.CreateCommand();
    cmd.CommandText = $@"SELECT TblSach.tenSach, TblSach.soLuongTon,
TblSach.giaNhap, TblSach.giaBan, TblNhaXuatBan.tenNXB, TblTacGia.tenTG
FROM TblSach
INNER JOIN TblNhaXuatBan ON TblSach.maNXB = TblNhaXuatBan.maNXB
INNER JOIN TblTacGia ON TblSach.maTG = TblTacGia.maTG
WHERE tenNXB LIKE N'%{tenNXB}%' AND tenTG LIKE N'%{tenTG}%'";
    adt.SelectCommand = cmd;
    dtInTKSL.Clear();
    adt.Fill(dtInTKSL);
    Close_DB();
    return dtInTKSL;
}

```

- Trả về bảng dữ liệu để thống kê số lượng tồn.

■ public DataTable showDSS()

```

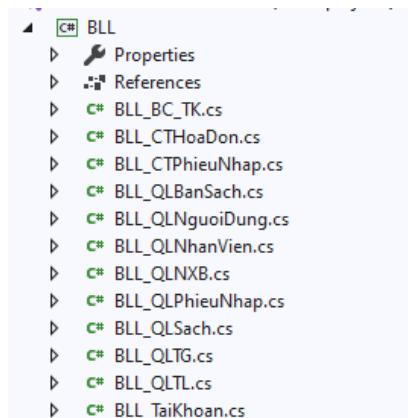
public DataTable showDSS()
{
    connect_DB();
    cmd = sqlcon.CreateCommand();
    cmd.CommandText = $@"SELECT TblSach.tenSach AS TÊN_SÁCH,
TblSach.soLuongTon AS SỐ_LƯỢNG_TỒN, TblSach.giaNhap AS GIÁ_NHẬP, TblSach.giaBan
AS GIÁ_BÁN, TblNhaXuatBan.tenNXB AS TÊN_NXB, TblTacGia.tenTG AS TÊN_TG
FROM TblSach
INNER JOIN TblNhaXuatBan ON TblSach.maNXB = TblNhaXuatBan.maNXB
INNER JOIN TblTacGia ON TblSach.maTG = TblTacGia.maTG";
    adt.SelectCommand = cmd;
    dtShowTKSL.Clear();
    adt.Fill(dtShowTKSL);
    Close_DB();
    return dtShowTKSL;
}

```

- Trả về bảng dữ liệu sách.

5.4 Tầng BLL

- Lớp này em sử dụng chủ yếu đóng vai trò nhận các yêu cầu từ tầng GUI và truy xuất đến tầng DAL để lấy thông tin trả về GUI.
- Các thành phần trong tầng BLL



Hình 5.30: Các thành phần trong tầng BLL

- Các class trong tầng BLL

- Class `BLL_TaiKhoan`

```
public class BLL_TaiKhoan
{
    DAL_TaiKhoan _TK = new DAL_TaiKhoan();

    public string DangNhap(DTO_TaiKhoan dn)
    {
        return _TK.dangNhap(dn);
    }
    public bool DoiMatKhau(DTO_TaiKhoan dn)
    {
        return _TK.doiMaKhau(dn);
    }
    public string getCV(string id)
    {
        return _TK.getCV(id);
    }
}
```

- Class `BLL_QLTL`

```
public class BLL_QLTL
{
    DAL_QLTL qltl = new DAL_QLTL();
    public DataTable showTL()
    {
        return qltl.showTL();
    }
    public DataTable searchMaTL(string maTL)
    {
        return qltl.searchMaTL(maTL);
    }
    public DataTable searchTenTL(string tenTL)
    {
        return qltl.searchTenTL(tenTL);
    }
    public bool addTL(DTO_QLTL tl)
    {
        return qltl.addTL(tl);
    }
    public bool editTL(DTO_QLTL tl)
    {
        return qltl.editTL(tl);
    }
    public bool deleteTL(DTO_QLTL tl)
    {
        return qltl.deleteTL(tl);
    }
}
```

- Class `BLL_QLTG`

```
public class BLL_QLTG
{
    DAL_QLTG qltg = new DAL_QLTG();
    public DataTable showTG()
    {
        return qltg.showTG();
    }
    public DataTable searchMaTG(string maTG)
```

```

    {
        return qltg.searchMaTG(maTG);
    }
    public DataTable searchTenTG(string tenTG)
    {
        return qltg.searchTenTG(tenTG);
    }
    public bool addTG(DTO_QLTG tg)
    {
        return qltg.addTG(tg);
    }
    public bool editTG(DTO_QLTG tg)
    {
        return qltg.editTG(tg);
    }
    public bool deleteTG(DTO_QLTG tg)
    {
        return qltg.deleteTG(tg);
    }
}

```

- Class [BLL_QLSach](#)

```

public class BLL_QLSach
{
    DAL_QLSach qls = new DAL_QLSach();
    public DataTable showSach()
    {
        return qls.showSach();
    }
    public bool addSach(DTO_QLSach s)
    {
        return qls.addSach(s);
    }
    public bool editSach(DTO_QLSach s)
    {
        return qls.editSach(s);
    }
    public bool deleteSach(DTO_QLSach s)
    {
        return qls.deleteSach(s);
    }
    public DataTable loadTenTL()
    {
        return qls.loadTenTL();
    }
    public DataTable loadTenTG()
    {
        return qls.loadTenTG();
    }
    public DataTable loadTenNXB()
    {
        return qls.loadTenNXB();
    }
    public string loadMaTL(string tenTL)
    {
        return qls.loadMaTL(tenTL);
    }
    public string loadMaTG(string tenTG)
    {
        return qls.loadMaTG(tenTG);
    }
    public string loadMaNXB(string tenNXB)
}

```

```

    {
        return qls.loadMaNXB(tenNXB);
    }
    public DataTable searchSach(string maS, string tenS, string tl, string tg)
    {
        return qls.searchSach(maS, tenS, tl, tg);
    }
}

```

- Class **BLL_QLPhieuNhap**

```

public class BLL_QLPhieuNhap
{
    DAL_QLPhieuNhap qlpn = new DAL_QLPhieuNhap();
    public DataTable showPN()
    {
        return qlpn.showPN();
    }
    public bool addPN(DTO_QLPhieuNhap pn)
    {
        return qlpn.addPN(pn);
    }
    public bool editPN(DTO_QLPhieuNhap pn)
    {
        return qlpn.editPN(pn);
    }
    public bool deletePN(DTO_QLPhieuNhap pn)
    {
        return qlpn.deletePN(pn);
    }
    public DataTable searchPN(string maPN)
    {
        return qlpn.searchPN(maPN);
    }
    public DataTable loadTenNXB()
    {
        return qlpn.loadTenNXB();
    }
    public String getMaNXB(string tenNXB)
    {
        return qlpn.getMaNXB(tenNXB);
    }
    public String getMaNV(string tk)
    {
        return qlpn.getMaNV(tk);
    }
}

```

- Class **BLL_QLNXB**

```

public class BLL_QLNXB
{
    DAL_QLNXB qlnxb = new DAL_QLNXB();
    public bool addNXB(DTO_QLNXB nxb)
    {
        return qlnxb.addNXB(nxb);
    }
    public bool editNXB(DTO_QLNXB nxb)
    {
        return qlnxb.editNXB(nxb);
    }
    public bool deleteNXB(DTO_QLNXB nxb)
    {
    }
}

```

```

        return qlnxb.deleteNXB(nx);
    }
    public DataTable searchMaNXB(string maNXB)
    {
        return qlnxb.searchMaNXB(maNXB);
    }
    public DataTable searchTenNXB(string tenNXB)
    {
        return qlnxb.searchTenNXB(tenNXB);
    }
    public DataTable showNXB()
    {
        return qlnxb.showNXB();
    }
}

```

- Class [BLL_QLNhanVien](#)

```

public class BLL_QLNhanVien
{
    DAL_QLNhanVien qlnv = new DAL_QLNhanVien();

    public DataTable showNV()
    {
        return qlnv.ShowNV();
    }
    public DataTable searchMaNV(string maNV)
    {
        return qlnv.searchMaNV(maNV);
    }
    public DataTable searchTenNV(string tenNV)
    {
        return qlnv.searchTenNV(tenNV);
    }
    public bool addNV(DTO_QLNhanVien nv)
    {
        return qlnv.addNV(nv);
    }
    public bool editNV(DTO_QLNhanVien nv)
    {
        return qlnv.editNV(nv);
    }
    public bool deleteNV(DTO_QLNhanVien nv)
    {
        return qlnv.deleteNV(nv);
    }
}

```

- Class [BLL_QLNguoiDung](#)

```

public class BLL_QLNguoiDung
{
    DAL_QLNguoiDung qlnd = new DAL_QLNguoiDung();

    public DataTable showUser()
    {
        return qlnd.showUser();
    }

    public bool addUser(DTO_QLNguoiDung nd)
    {
        return qlnd.addUser(nd);
    }
}

```

```

public bool editUser(DTO_QLNguoiDung nd)
{
    return qlnd.editUser(nd);
}

public bool deleteUser(DTO_QLNguoiDung nd)
{
    return qlnd.deleteUser(nd);
}

public DataTable showTenNV()
{
    return qlnd.loadTenNV();
}

public string loadMaNV(string tenNV)
{
    return qlnd.loadMaNV(tenNV);
}
}

```

- Class [BLL_QLBanSach](#)

```

public class BLL_QLBanSach
{
    DAL_QLBanSach qlhd = new DAL_QLBanSach();
    public bool addHD(DTO_QLBanSach hd)
    {
        return qlhd.addHD(hd);
    }
    public bool editHD(DTO_QLBanSach hd)
    {
        return qlhd.editHD(hd);
    }
    public bool deleteHD(DTO_QLBanSach hd)
    {
        return qlhd.deleteHD(hd);
    }

    public string getMaNV(string tk)
    {
        return qlhd.getMaNV(tk);
    }

    public DataTable searchHD(string maHD)
    {
        return qlhd.searchHD(maHD);
    }

    public DataTable showHD()
    {
        return qlhd.showHD();
    }

    public DataTable inHD(DTO_QLBanSach hd)
    {
        return qlhd.inHD(hd);
    }
}

```

- Class [BLL_CTPhieuNhap](#)

```
public class BLL_CTPhieuNhap
{
    DAL_CTPhieuNhap qlctpn = new DAL_CTPhieuNhap();
    public DataTable showCTPN(string maPN)
    {
        return qlctpn.showCTPN(maPN);
    }

    public DataTable showSach(string maNXB)
    {
        return qlctpn.showSach(maNXB);
    }
    public string getGiaNhap(int sl, string maS)
    {
        return qlctpn.getGiaNhap(sl, maS);
    }
    public bool addS(DTO_CTPhieuNhap ctpn)
    {
        return qlctpn.addS(ctpn);
    }
    public bool editS(DTO_CTPhieuNhap ctpn, int slf)
    {
        return qlctpn.editS(ctpn, slf);
    }
    public bool deleteS(DTO_CTPhieuNhap ctpn)
    {
        return qlctpn.deleteS(ctpn);
    }

    public DataTable searchTenS(string maPN, string tenS)
    {
        return qlctpn.searchTenS(maPN, tenS);
    }
    public bool updateTongPN(DTO_CTPhieuNhap ctpn)
    {
        return qlctpn.updateTongPN(ctpn);
    }
    public bool updateTongPN(string maPN, int tmp)
    {
        return qlctpn.updateTongPN(maPN, tmp);
    }
}
```

- Class [BLL_CTHoaDon](#)

```
public class BLL_BC_TK
{
    DAL_BC_TK bctk = new DAL_BC_TK();
    public DataTable showDSPN()
    {
        return bctk.showDSPN();
    }
    public DataTable showCTDSPN(string maPN)
    {
        return bctk.showCTDSPN(maPN);
    }
    public DataTable inPN(string thang, string nam, string nxb)
    {
        return bctk.inPN(thang, nam, nxb);
    }
    public DataTable inCTPN(string thang, string nam, string nxb)
```

```

    {
        return bctk.inCTPN(thang, nam, nxb);
    }
    public DataTable LoadTenNXB()
    {
        return bctk.LoadTenNXB();
    }

    public DataTable showDSHD()
    {
        return bctk.showDSHD();
    }
    public DataTable showCTDSHD(string maHD)
    {
        return bctk.showCTDSHD(maHD);
    }
    public DataTable inHD(string thang, string nam)
    {
        return bctk.inHD(thang, nam);
    }
    public DataTable inCTHD(string thang, string nam)
    {
        return bctk.inCTHD(thang, nam);
    }
    public DataTable LoadTenTG()
    {
        return bctk.LoadTenTG();
    }

    public DataTable inTKSL(string tenNXB, string tenTG)
    {
        return bctk.inTKSL(tenNXB, tenTG);
    }
    public DataTable showDSS()
    {
        return bctk.showDSS();
    }
}

```

- Class [BLL_BC_TK](#)

```

public class BLL_BC_TK
{
    DAL_BC_TK bctk = new DAL_BC_TK();
    public DataTable showDSPN()
    {
        return bctk.showDSPN();
    }
    public DataTable showCTDSPN(string maPN)
    {
        return bctk.showCTDSPN(maPN);
    }
    public DataTable inPN(string thang, string nam, string nxb)
    {
        return bctk.inPN(thang, nam, nxb);
    }
    public DataTable inCTPN(string thang, string nam, string nxb)
    {
        return bctk.inCTPN(thang, nam, nxb);
    }
    public DataTable LoadTenNXB()
    {
        return bctk.LoadTenNXB();
    }
}

```

```

public DataTable showDSHD()
{
    return bctk.showDSHD();
}
public DataTable showCTDSHD(string maHD)
{
    return bctk.showCTDSHD(maHD);
}
public DataTable inHD(string thang, string nam)
{
    return bctk.inHD(thang, nam);
}
public DataTable inCTHD(string thang, string nam)
{
    return bctk.inCTHD(thang, nam);
}
public DataTable LoadTenTG()
{
    return bctk.LoadTenTG();
}

public DataTable inTKSL(string tenNXB, string tenTG)
{
    return bctk.inTKSL(tenNXB, tenTG);
}
public DataTable showDSS()
{
    return bctk.showDSS();
}
}

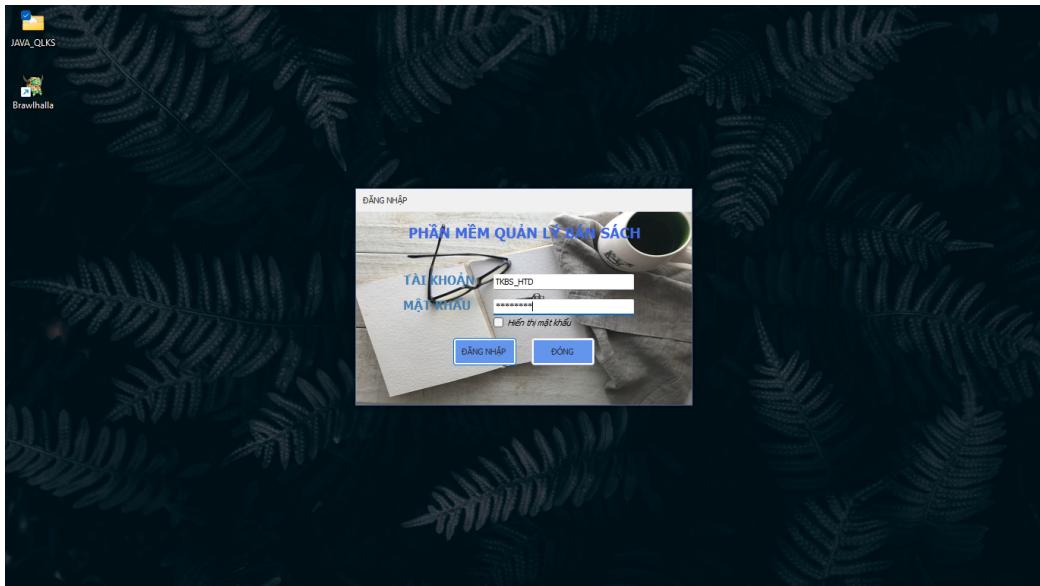
```

5.5 Kết chương

- Từ các giai đoạn trên em đã hoàn thành đồ án chuyên ngành 1 của mình tạo ra được một ứng dụng quản lý bán sách. Do kiến thức khi xây dựng ứng dụng còn hạn chế nên không tránh được những thiếu sót. Em rất vui khi thầy, cô đã đọc và góp ý cho em nhận ra và rút kinh nghiệm cho những lần sau.
- Cuối cùng em xin cảm ơn thầy Hà Trọng Thắng giảng viên môn đồ án chuyên ngành 1 của lớp em. Tuy thời gian không nhiều nhưng những góp ý của thầy khi chúng em báo cáo sơ bộ là vô cùng bổ ích và dễ hiểu.

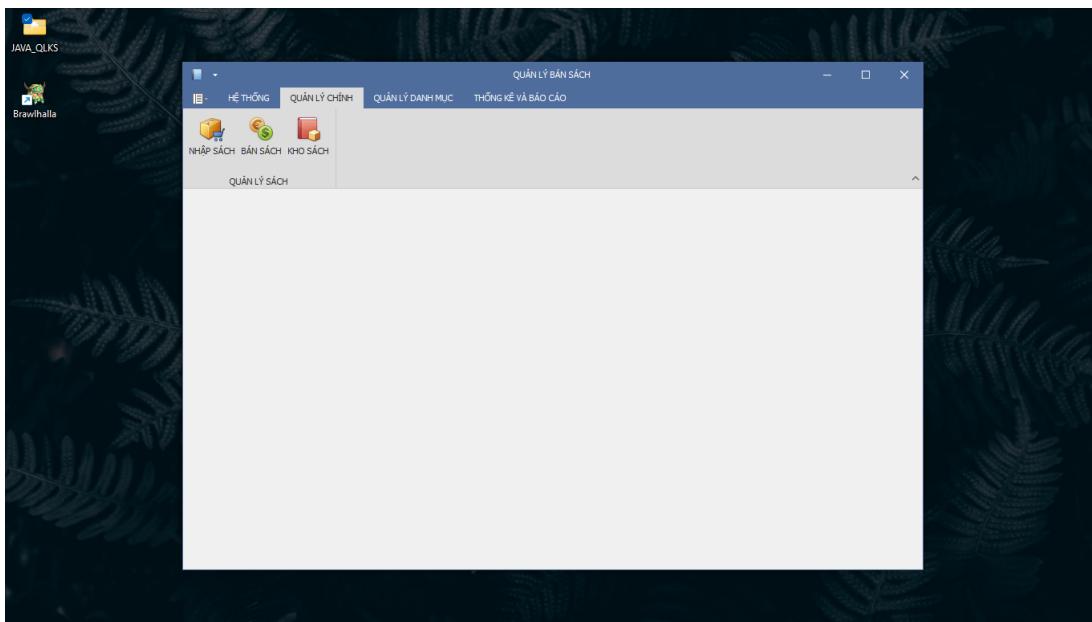
CHƯƠNG 6: KẾT QUẢ THU ĐƯỢC

- Chức năng đăng nhập



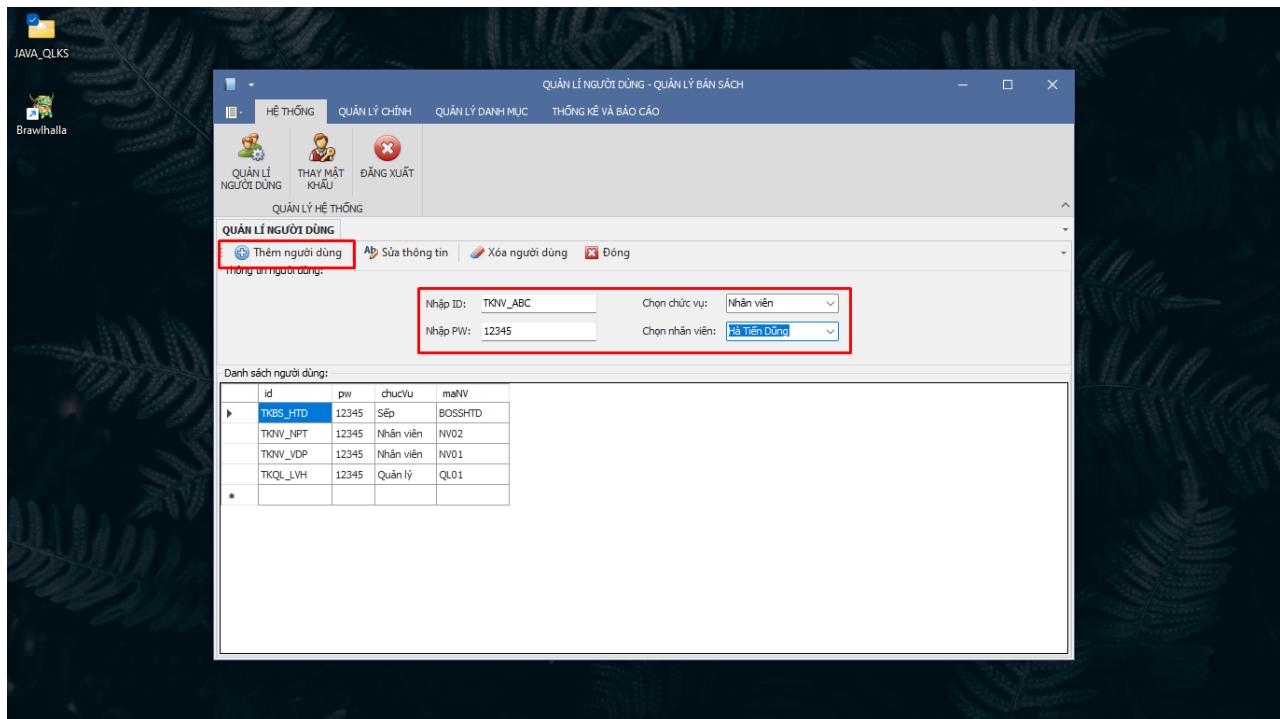
Hình 6.1: giao diện đăng nhập

- Sau khi đăng nhập thành công sẽ hiện ra giao diện chính (ảnh dưới là giao diện của admin. Khi đăng nhập tài khoản quản lý hay nhân viên giao diện sẽ chỉ có chức năng như ở use case đã thể hiện.)



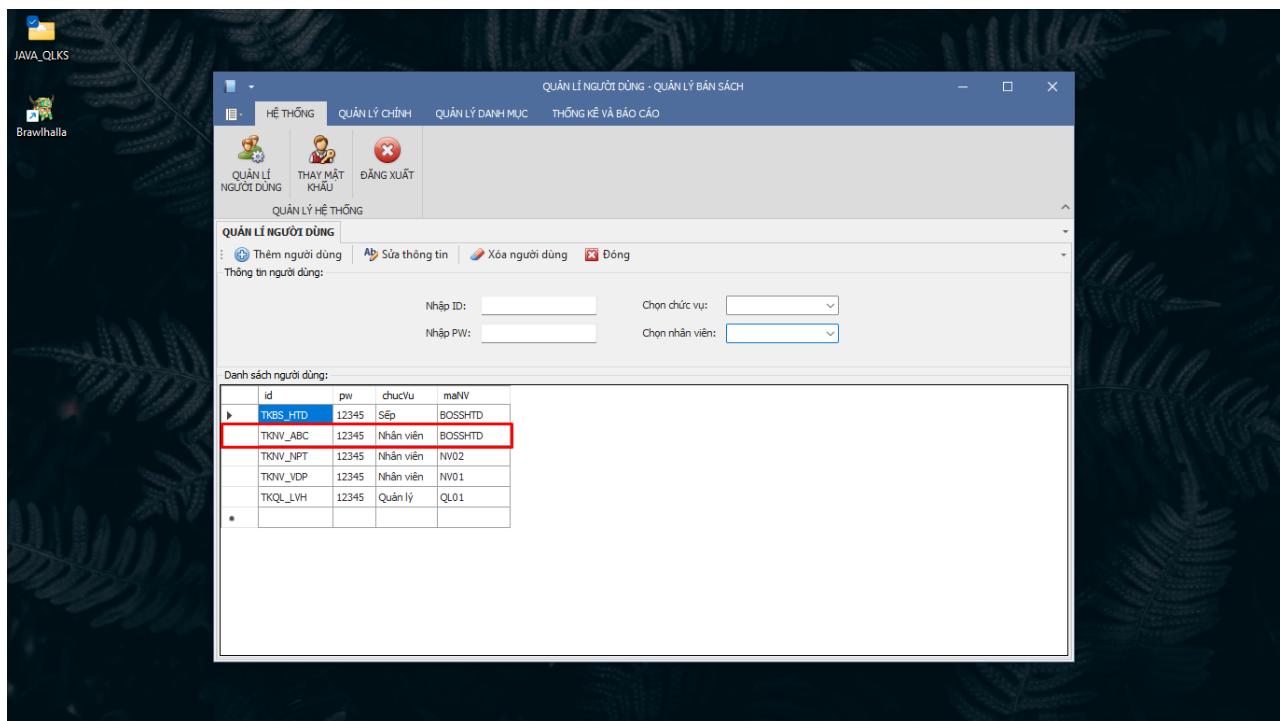
Hình 6.2: giao diện chính

- Chức năng quản lí người dùng
 - Thêm người dùng



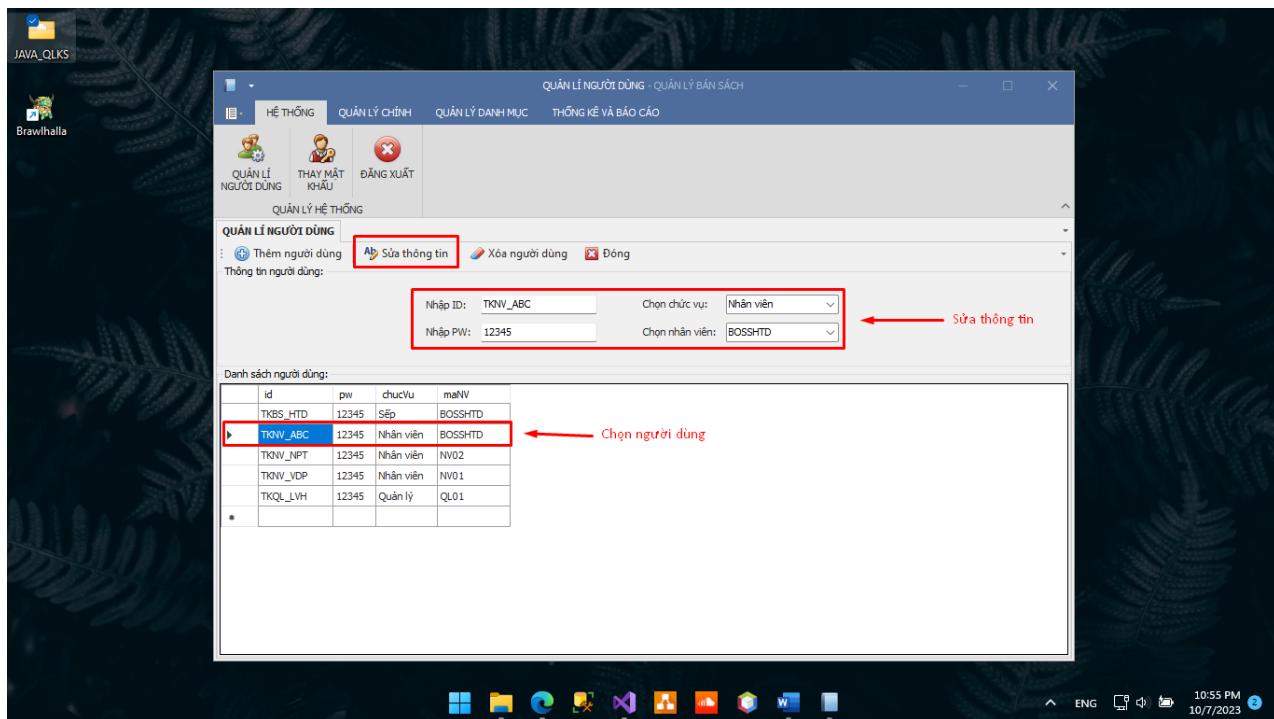
Hình 6.3: thực hiện thêm người dùng

- Kết quả



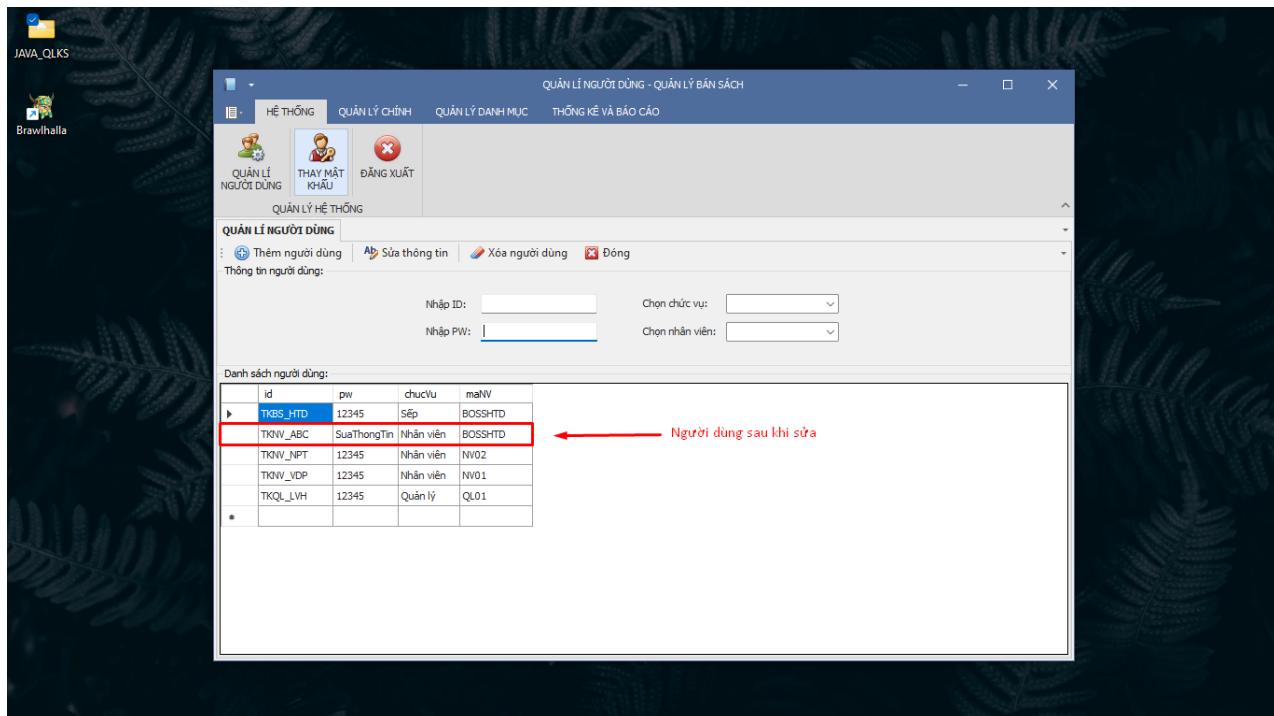
Hình 6.4: kết quả sau khi thêm người dùng

- Sửa thông tin người dùng



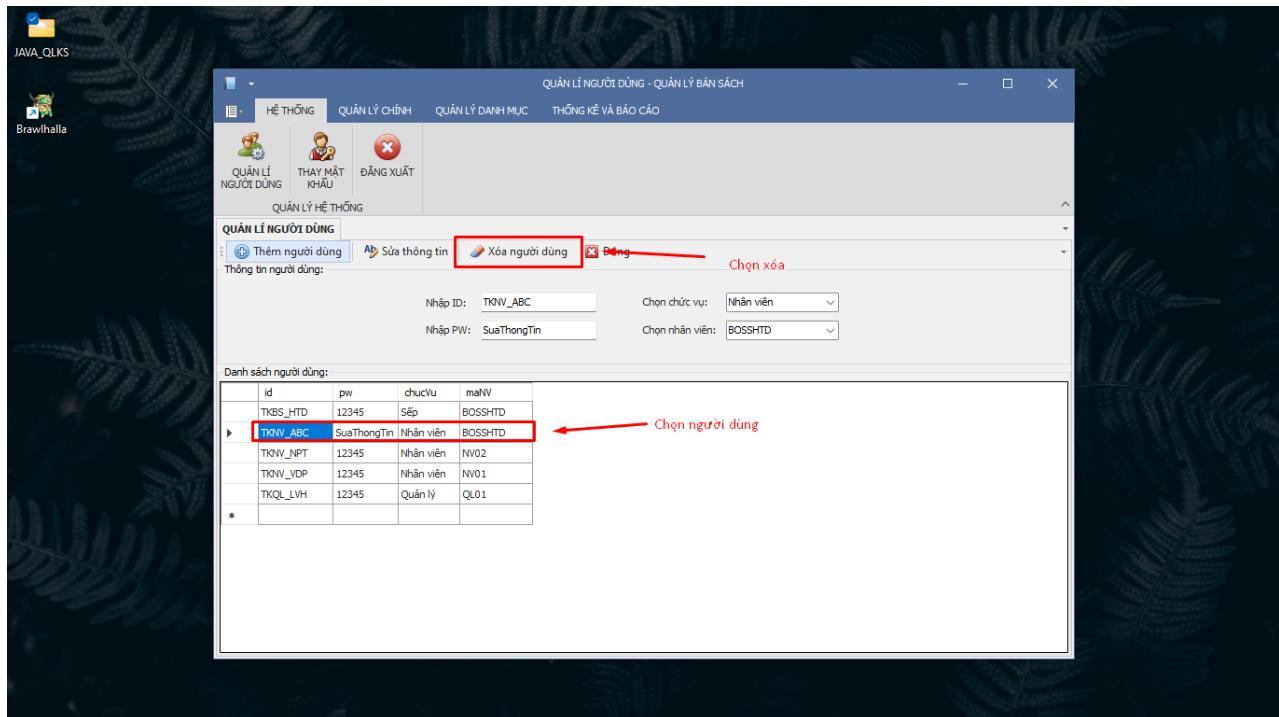
Hình 6.5: sửa thông tin người dùng

- Kết quả



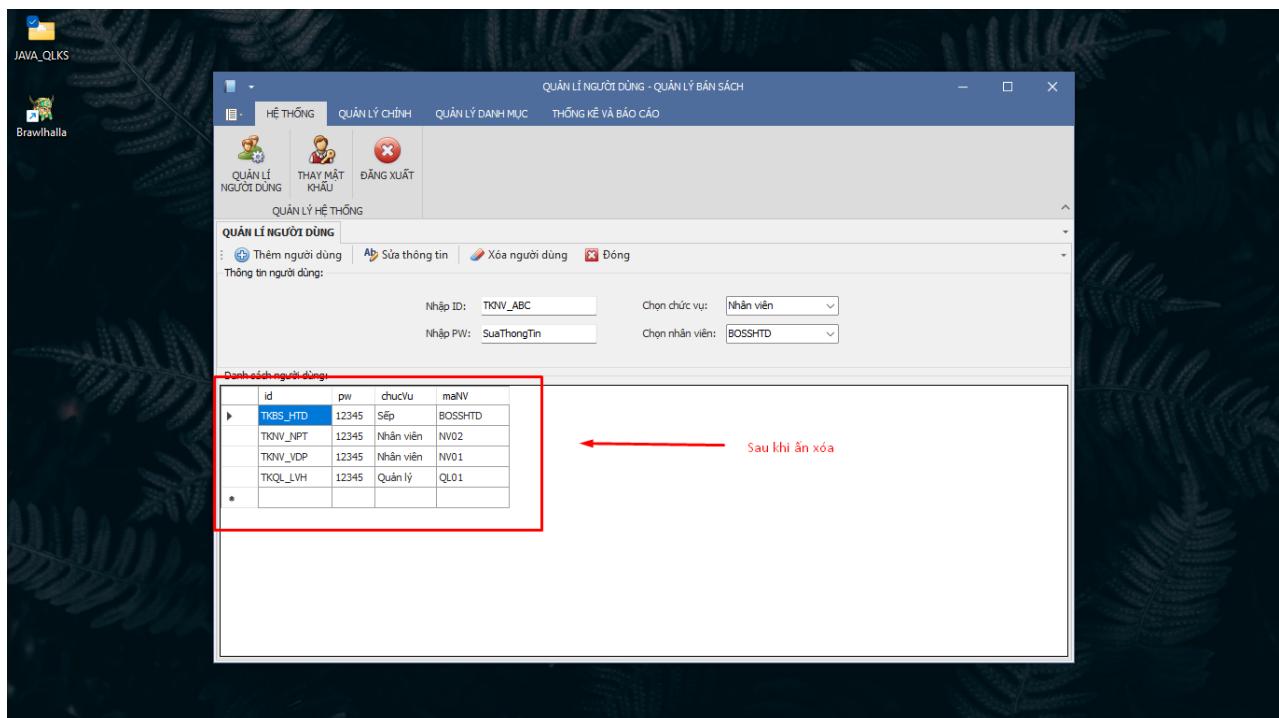
Hình 6.6: kết quả sau khi sửa thông tin người dùng

- Xóa người dùng



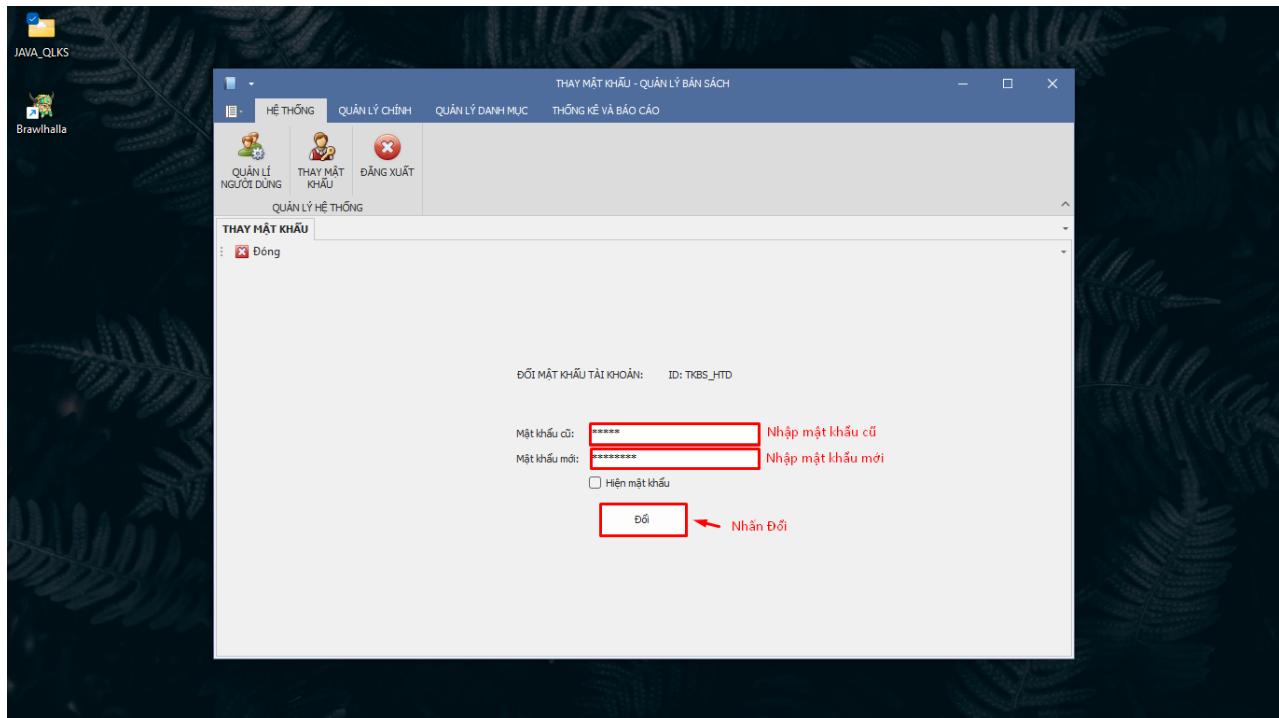
Hình 6.7: Xóa người dùng

- Kết quả



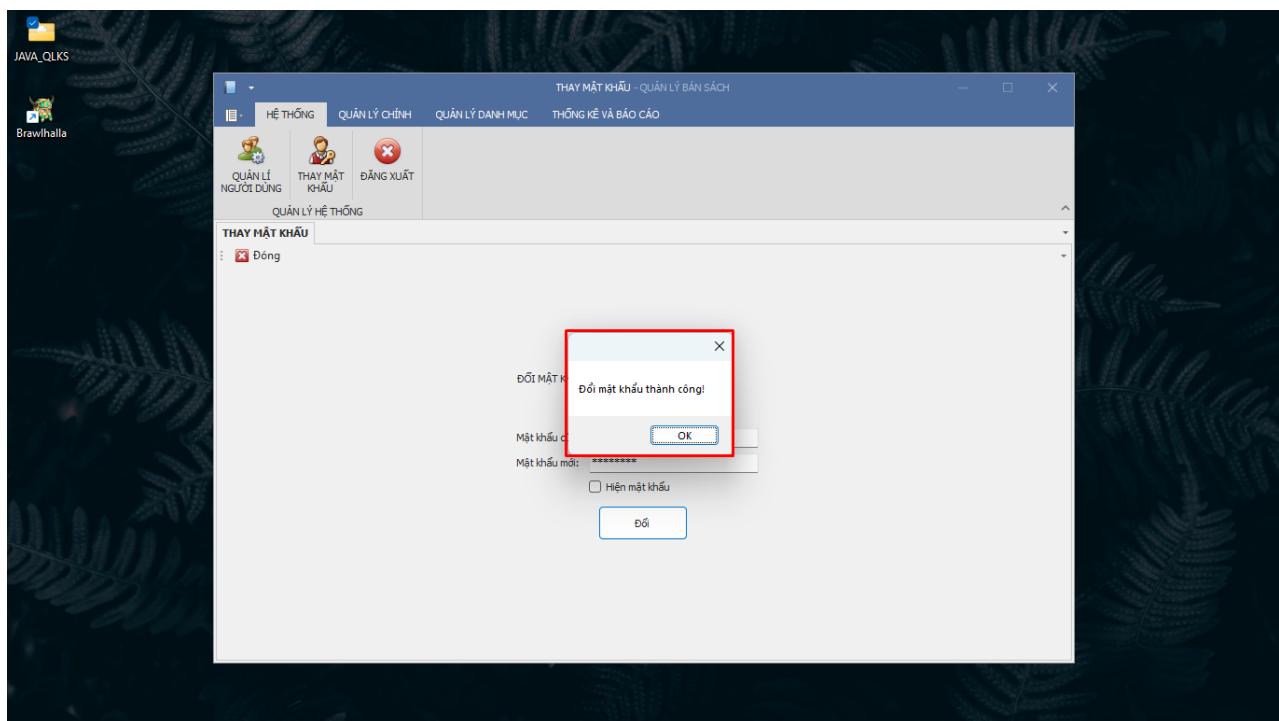
Hình 6.8:kết quả xóa người dùng

- Chức năng đổi mật khẩu



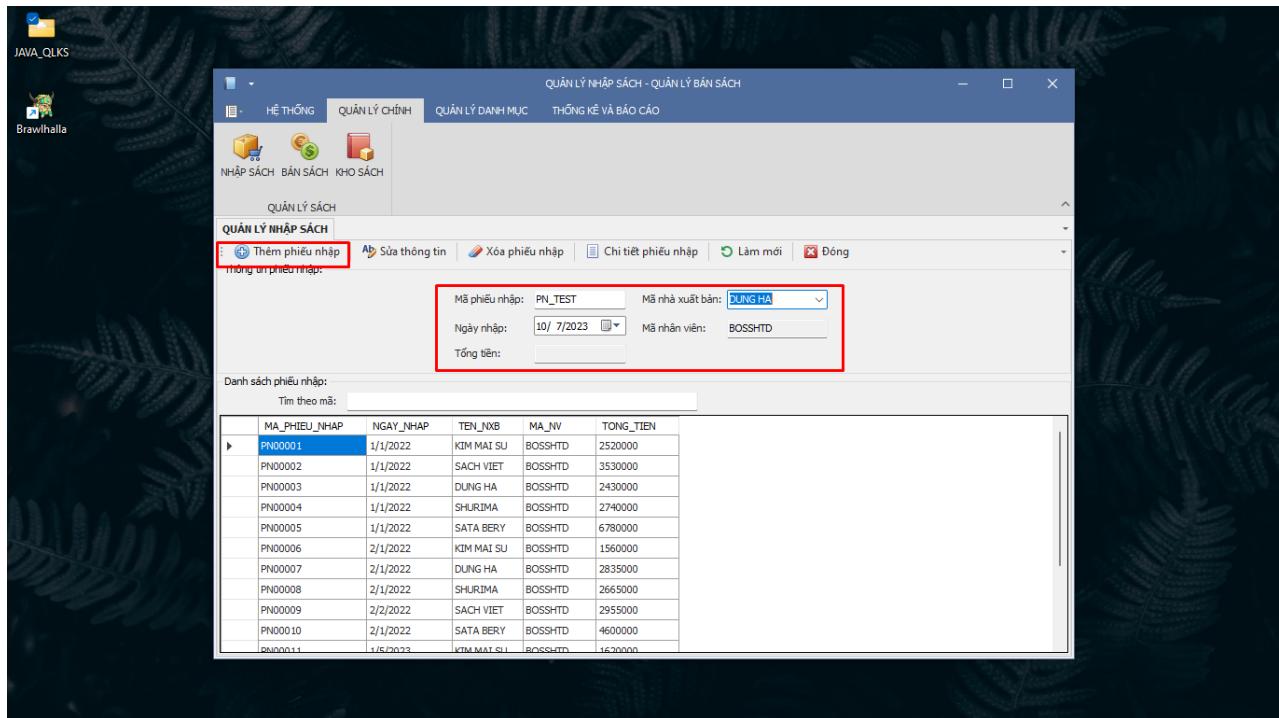
Hình 6.9: thay đổi mật khẩu

- Kết quả



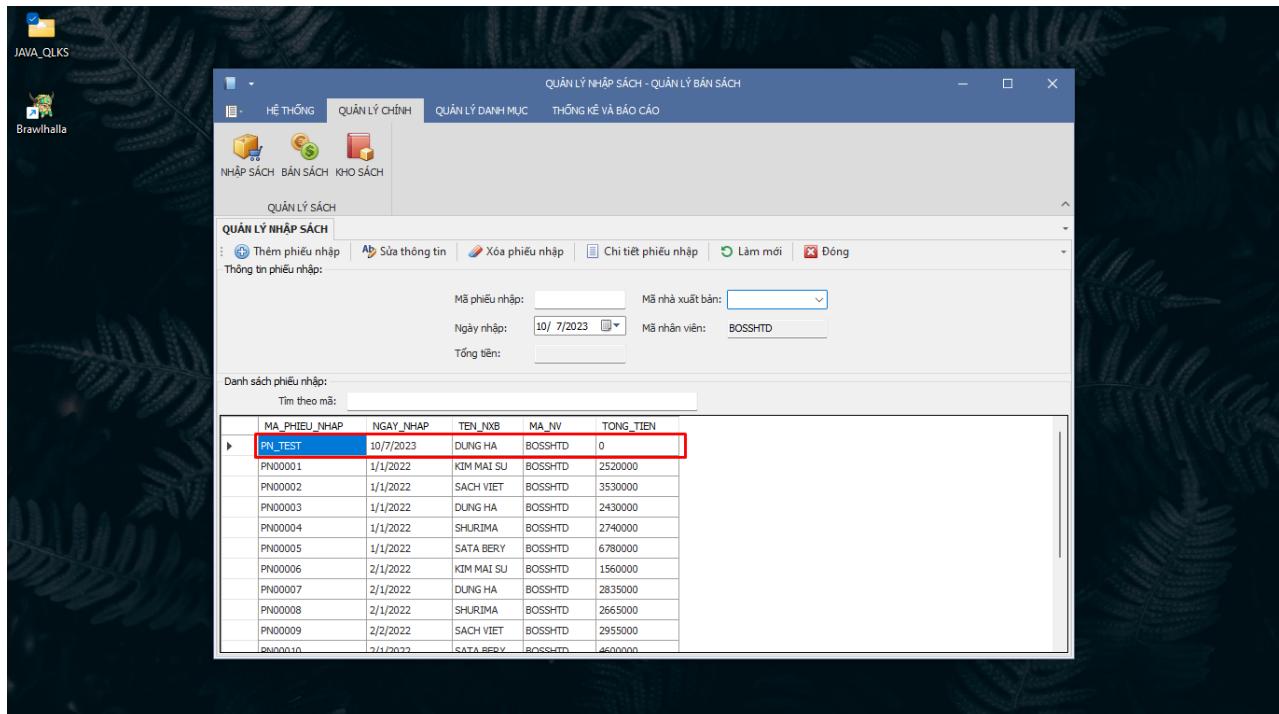
Hình 6.10: kết quả sau khi thay đổi mật khẩu

- Chức năng thêm phiếu nhập



Hình 6.11: thêm phiếu nhập

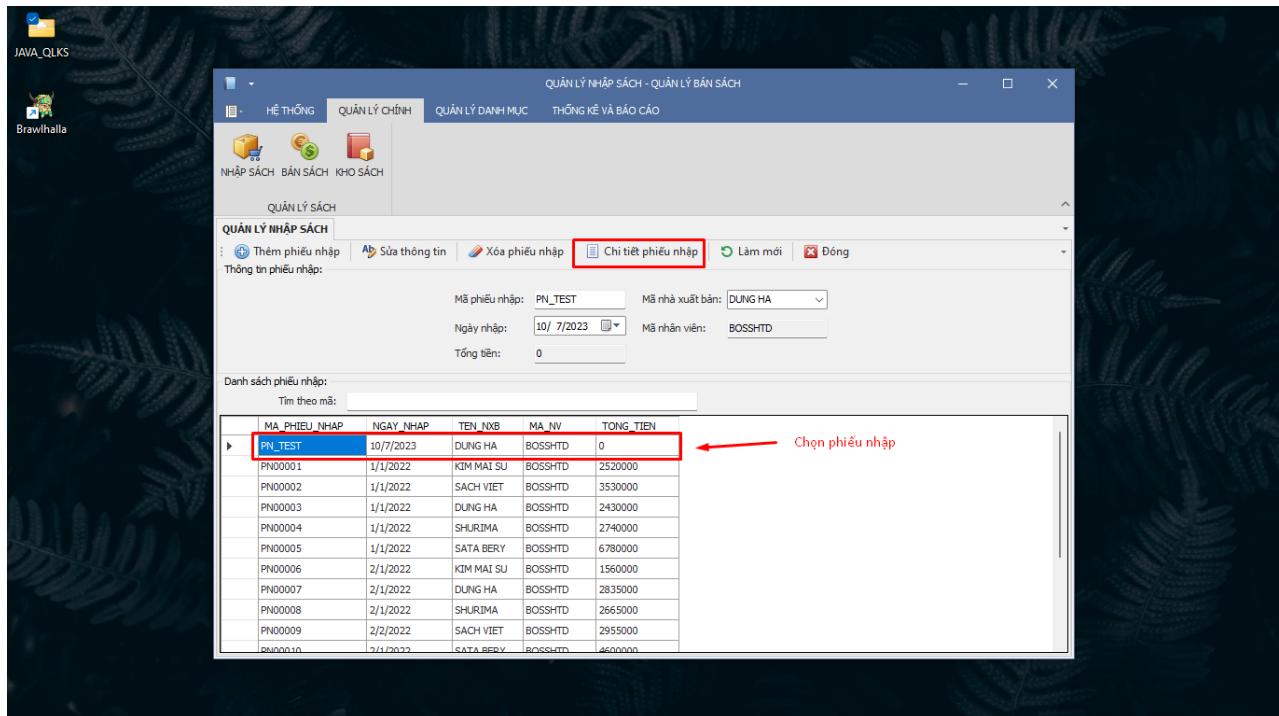
- Kết quả



Hình 6.11.1: kết quả khi thêm phiếu nhập

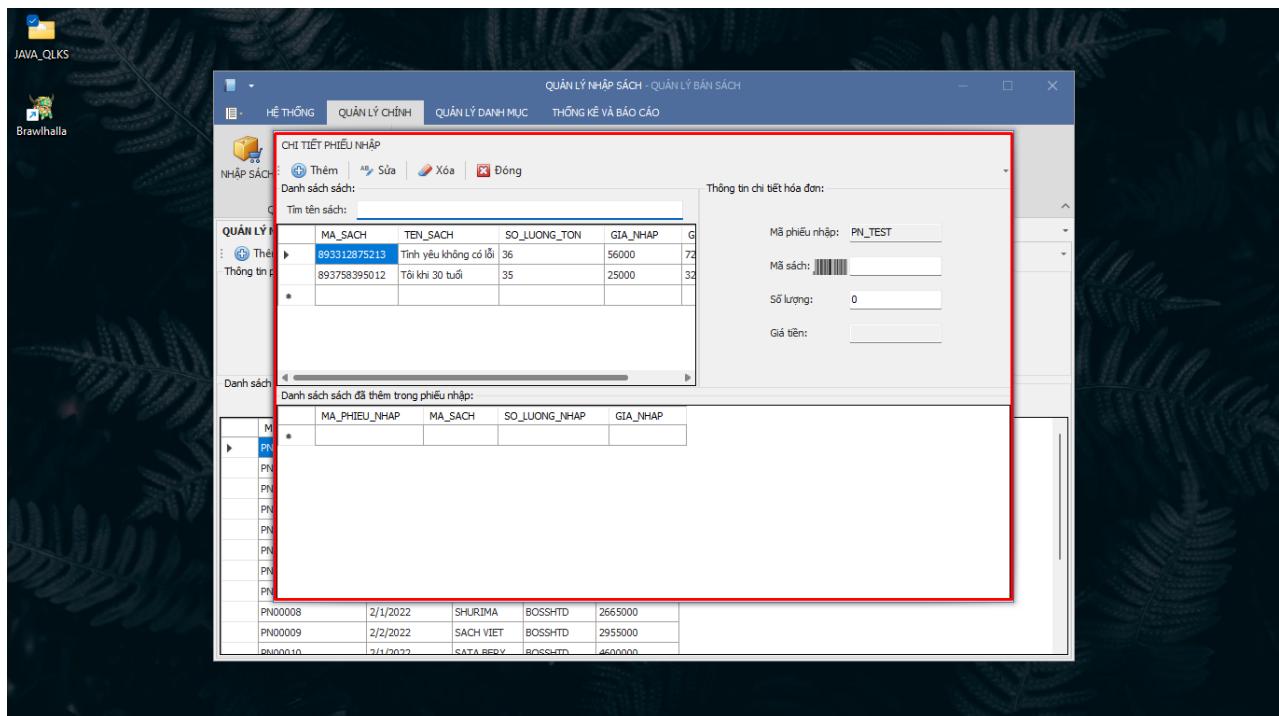
- Chức năng sửa xóa cũng tương tự như quản lý người dùng.

- Chức năng chi tiết phiếu nhập



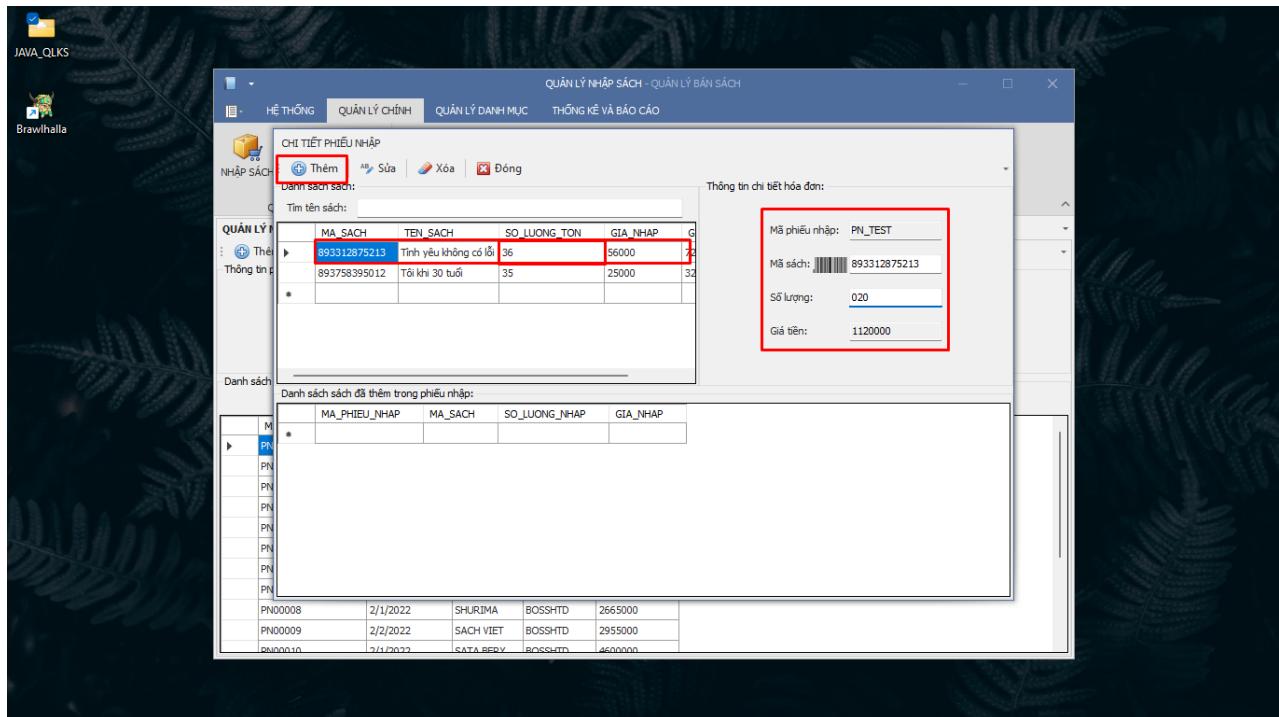
Hình 6.12: chọn chi tiết phiếu nhập của hóa đơn

- Kết quả



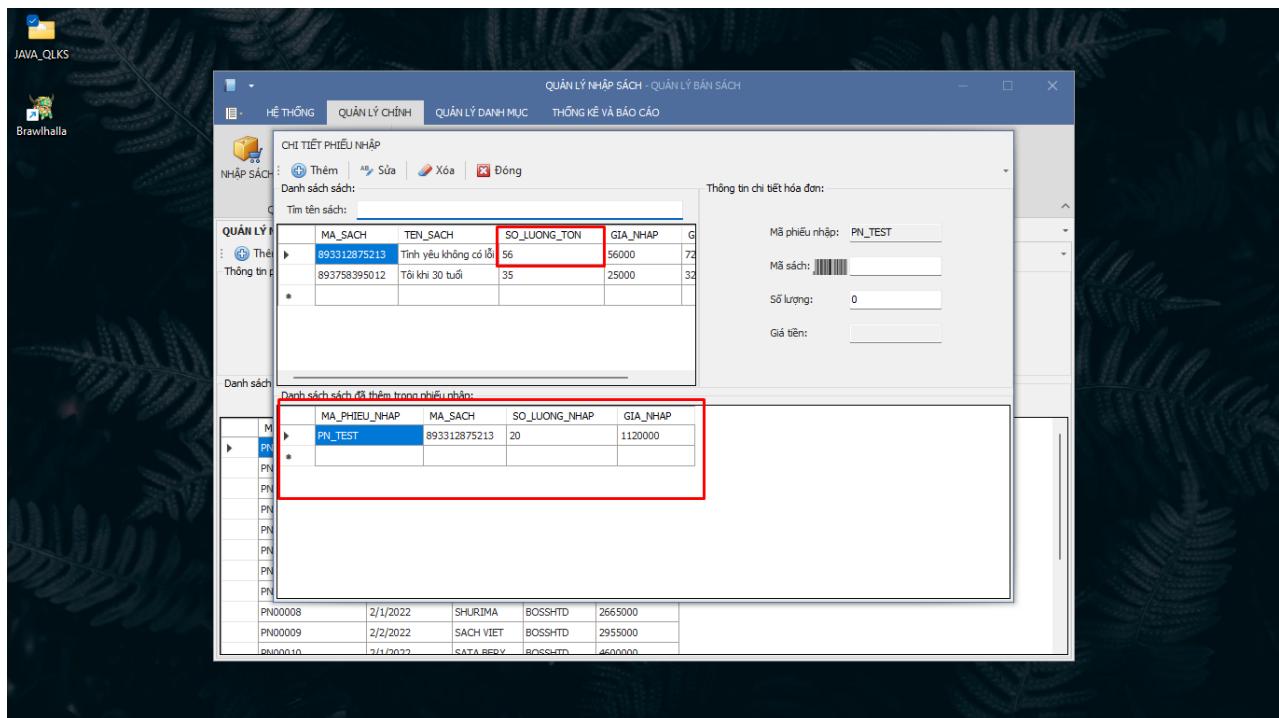
Hình 6.13: kết quả khi vào chi tiết phiếu nhập của một phiếu nhập

- Thêm sách vào phiếu nhập



Hình 6.14: Thêm sách nhập vào chi tiết phiếu nhập

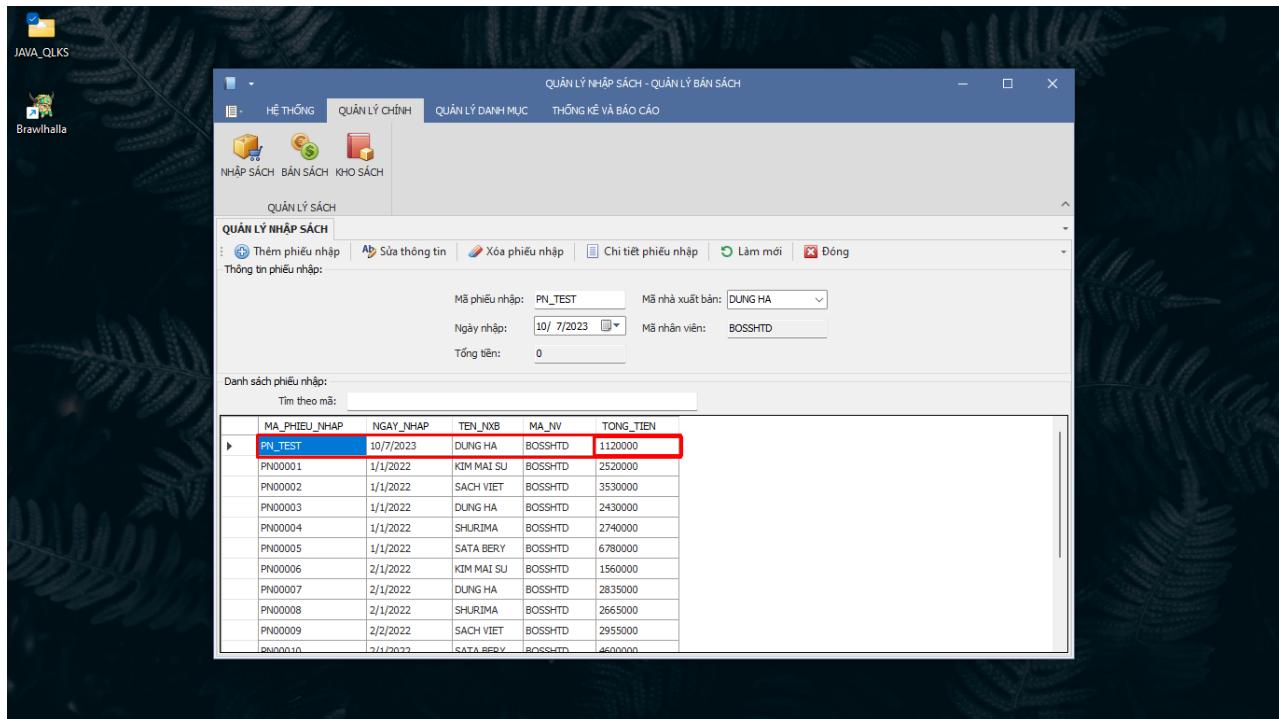
- Kết quả: Số lượng tồn của sách sẽ tăng lên và khi xóa đi thì số lượng tồn của sách cũng sẽ bị trừ đi tương ứng với số lượng xóa ở phiếu nhập.



Hình 6.15: Kết quả sau khi thêm sách nhập vào chi tiết phiếu nhập

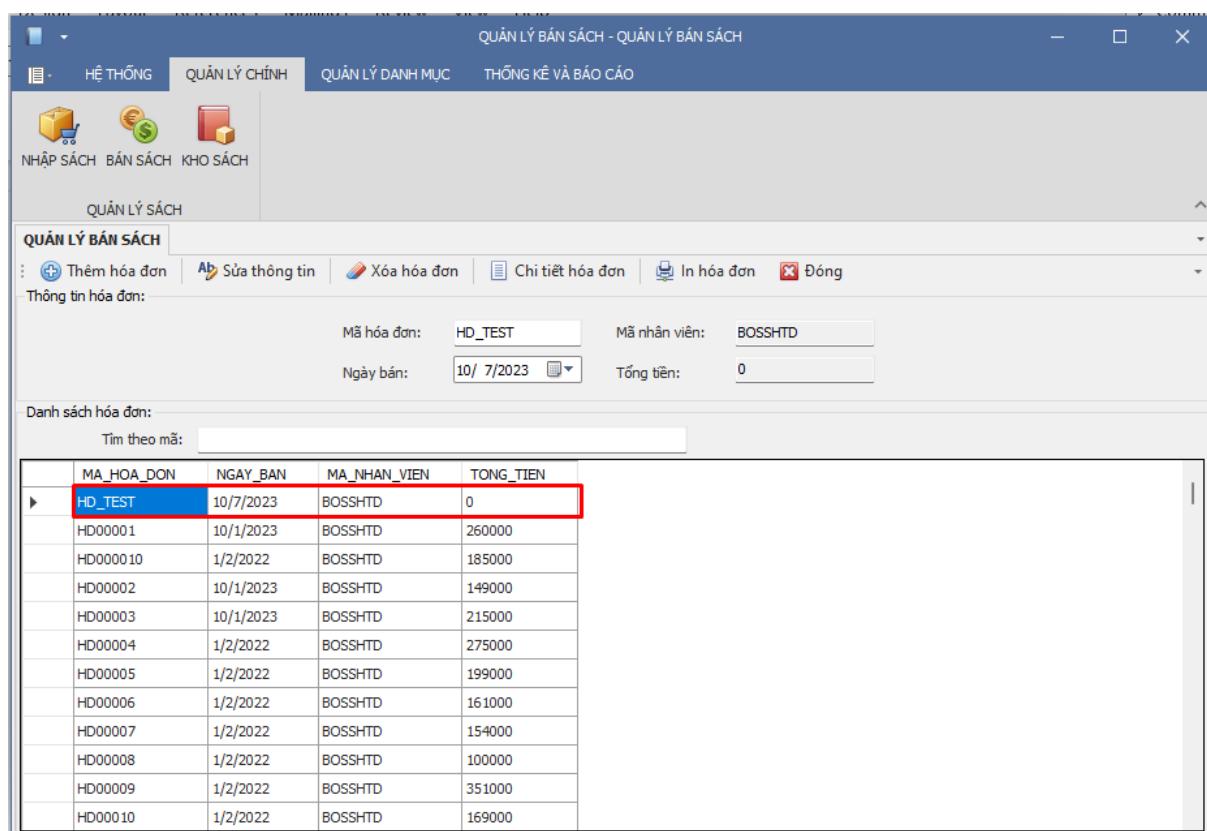
- Chức năng sửa xóa cũng tương tự như các chức năng thêm sửa xóa khác của chương trình.

- Sau khi đóng chi tiết phiếu nhập thì phiếu nhập cũng được cập nhật tổng tiền.

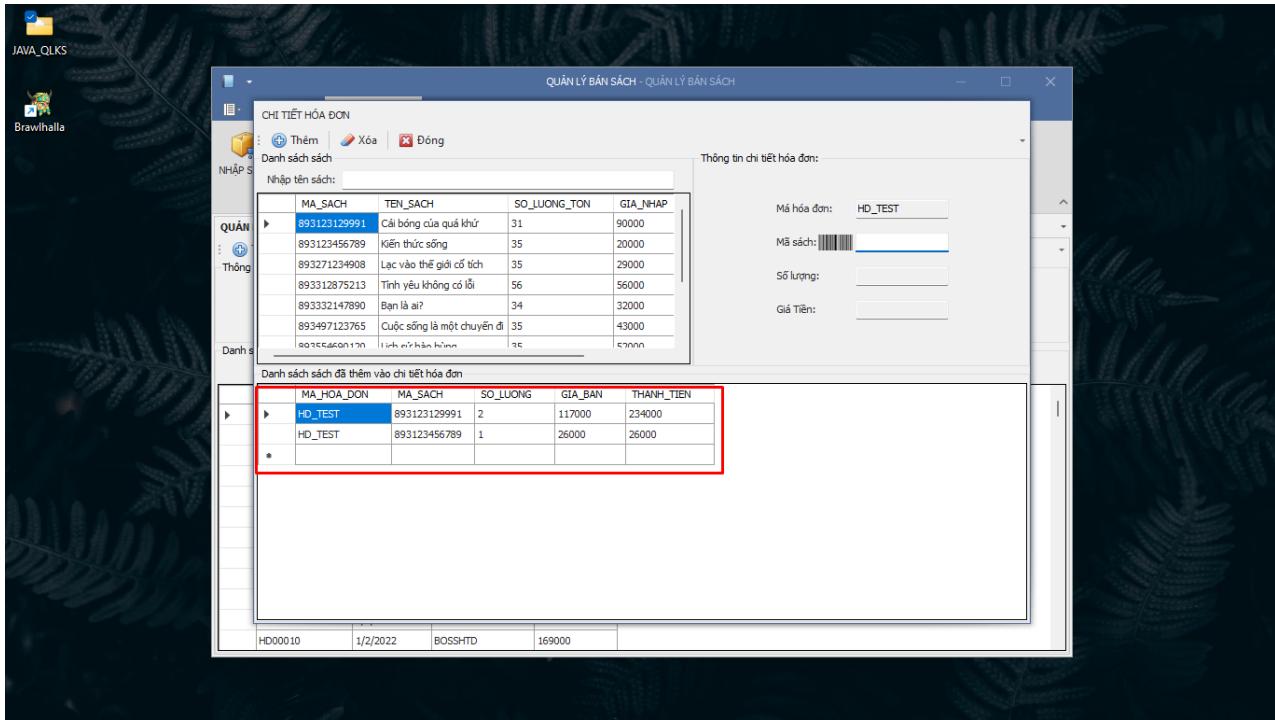


Hình 6.16: Phiếu nhập được cập nhật tổng tiền khi thêm sách nhập vào chi tiết phiếu nhập đó

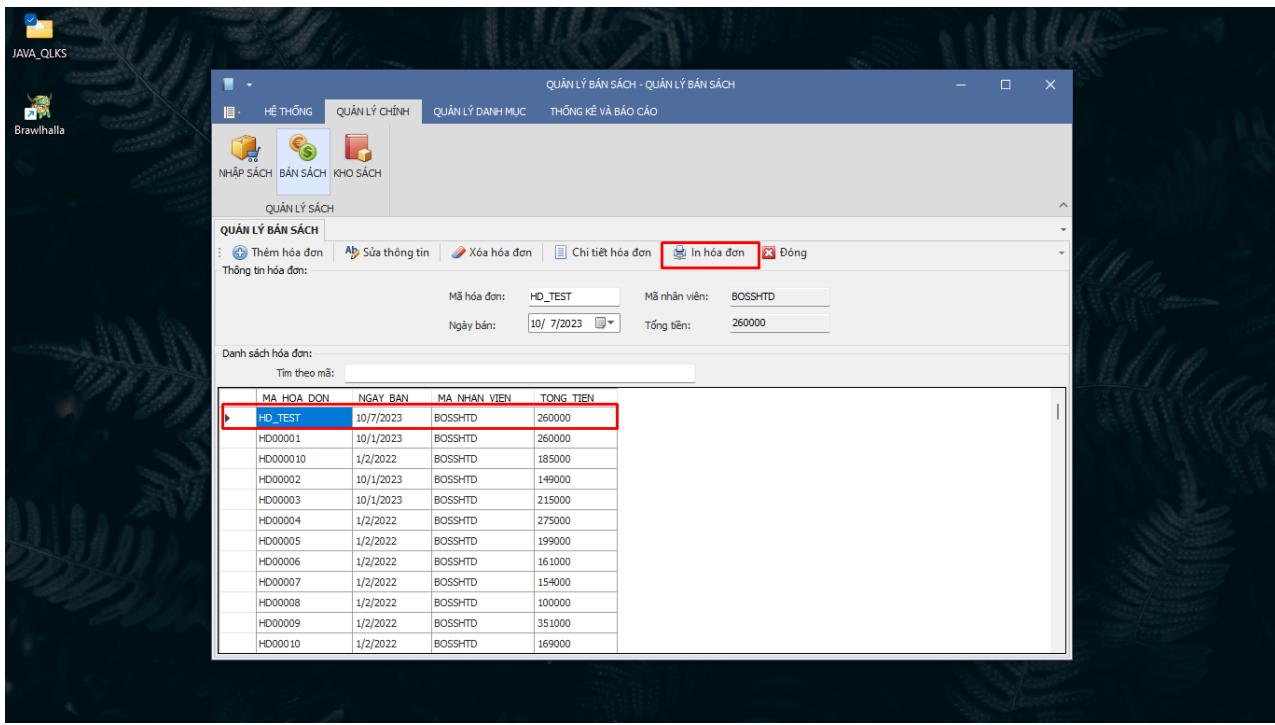
- **Chức năng bán sách**
 - Vì các chức năng thêm, sửa, xóa cũng tương tự như chức năng nhập sách chỉ khác hóa đơn có thêm chức năng in hóa đơn nên em sẽ demo in hóa đơn.



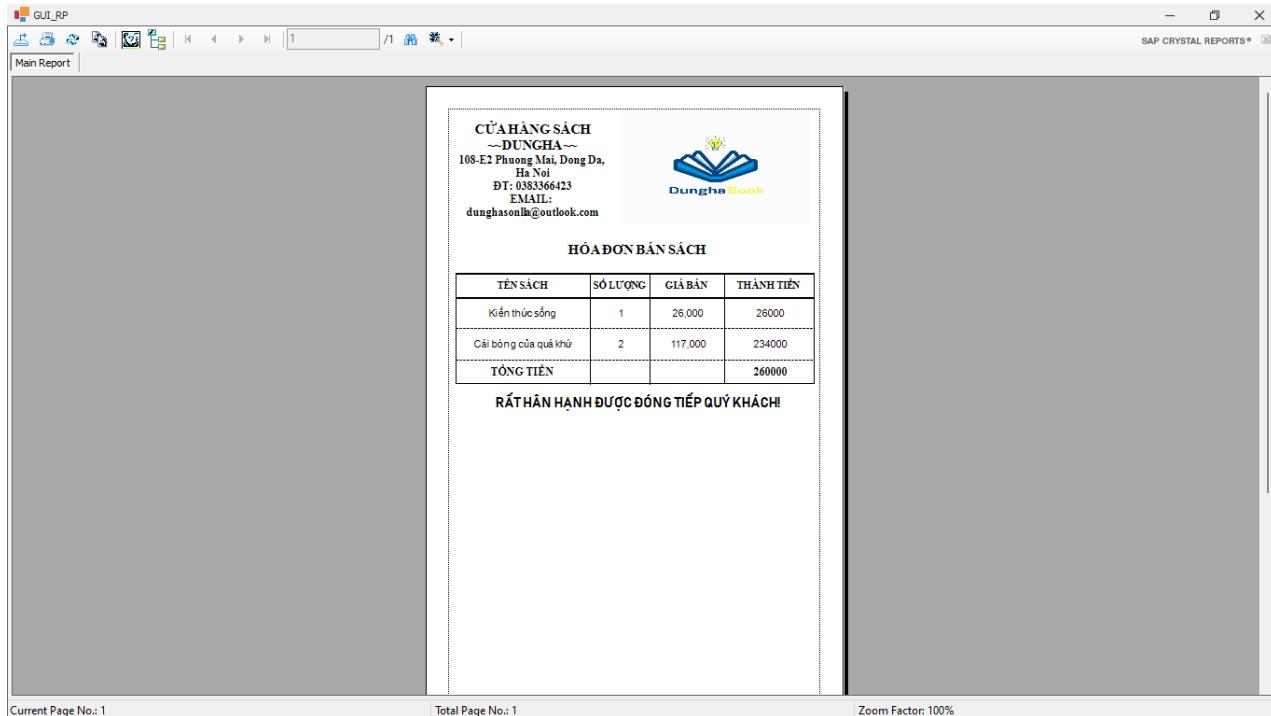
Hình 6.17: Thêm xong một hóa đơn



Hình 6.18: Thêm sách khách mua vào hóa đơn.

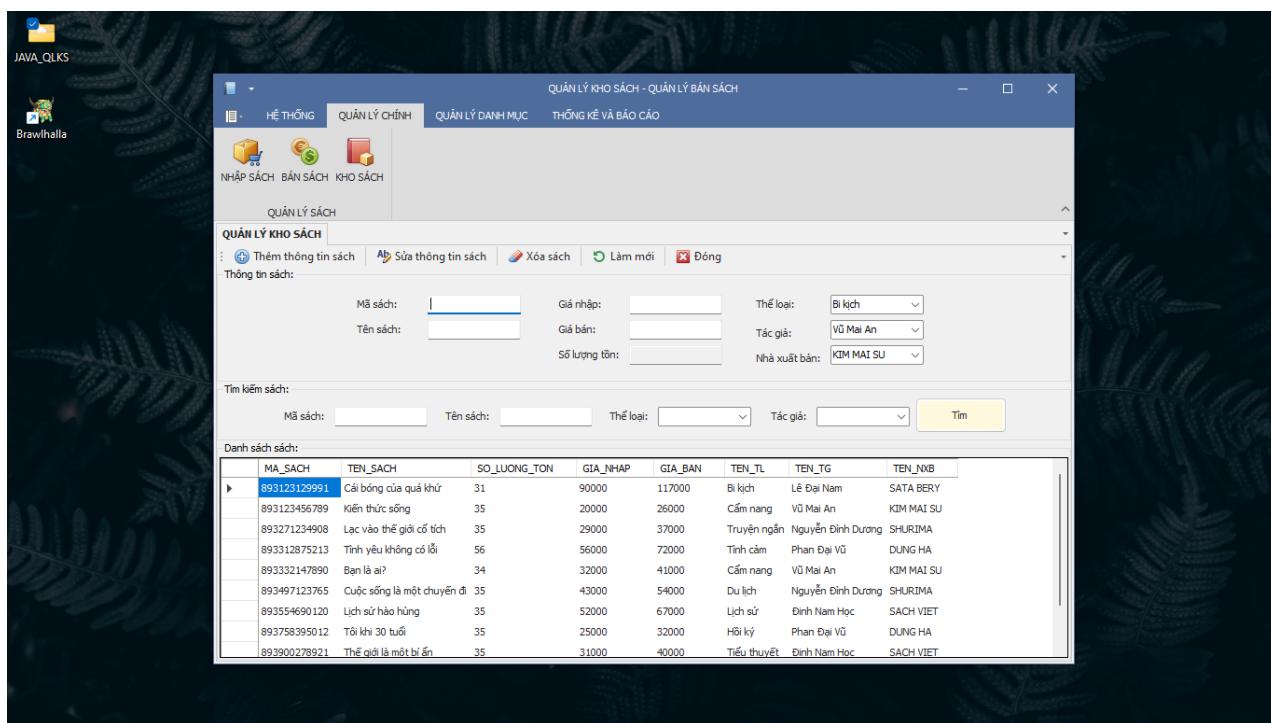


Hình 6.19: Chọn hóa đơn và chọn in hóa đơn



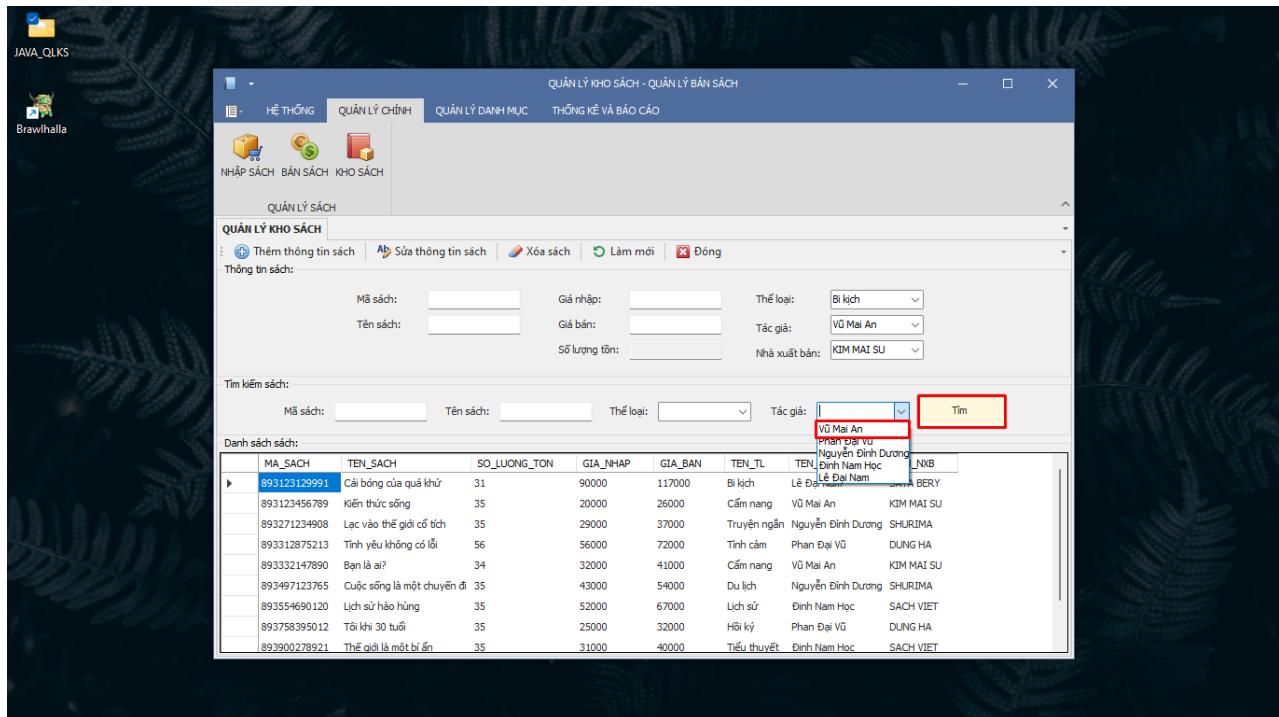
Hình 6.20: hóa đơn được in

- Chức năng quản lý kho sách
 - Để thêm sách thì cần nhập thông tin và ấn thêm. Sửa và xóa cũng tương tự như các chức năng khác.



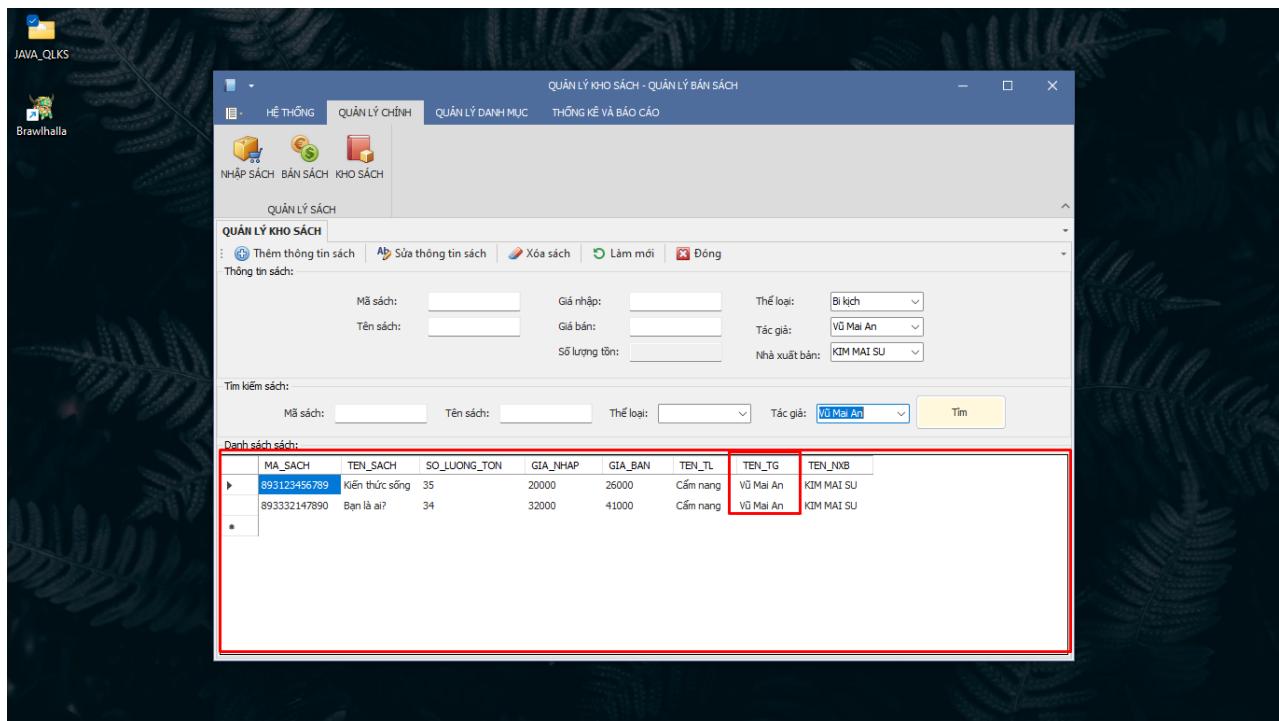
Hình 6.21: giao diện kho sách

- Tìm kiếm sách: chọn các thông tin của quyển sách muốn tìm



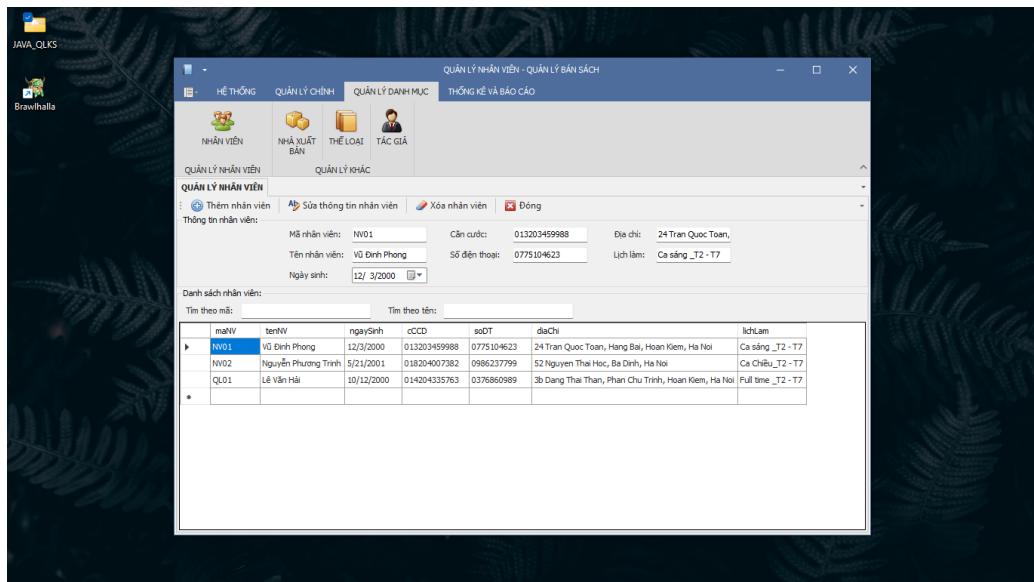
Hình 6.22: chọn thông tin tìm kiếm sách

- Kết quả

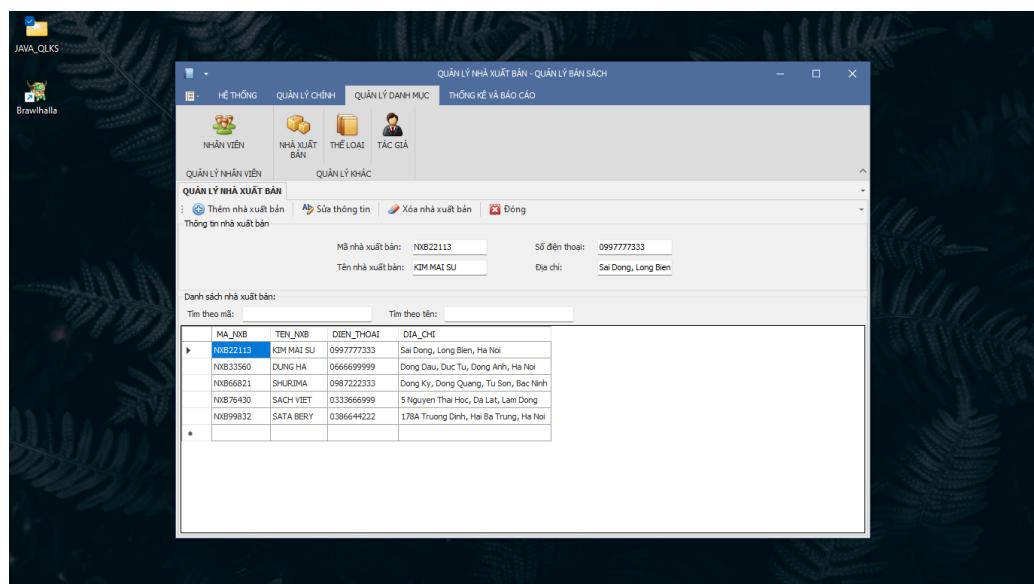


Hình 6.23: Kết quả khi áp dụng tìm kiếm

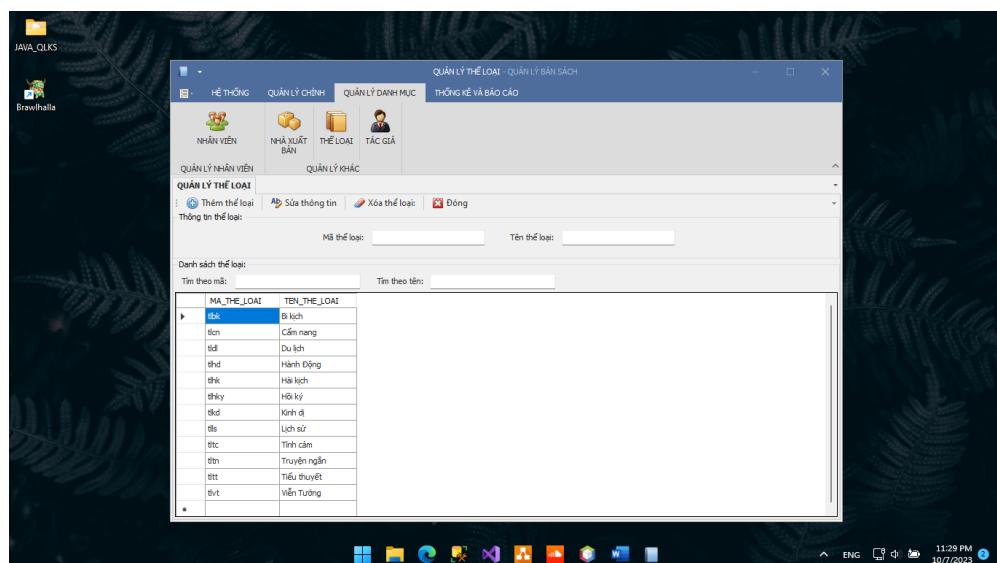
- Chức năng quản lý nhân viên, tác giả, thể loại và nhà xuất bản cũng có các chức năng thêm, sửa, xóa, tìm kiếm tương tự như các chức năng khác nên em sẽ không trình bày thêm.



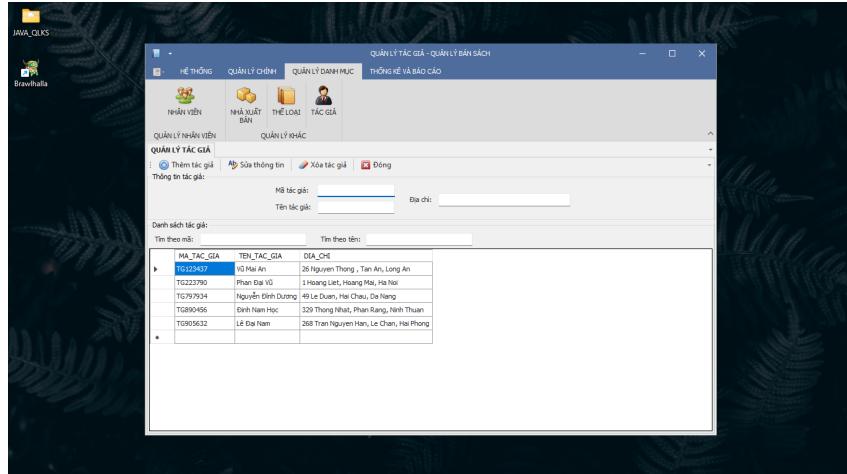
Hình 6.24: Giao diện quản lí nhân viên



Hình 6.25: Giao diện quản lí nhà xuất bản

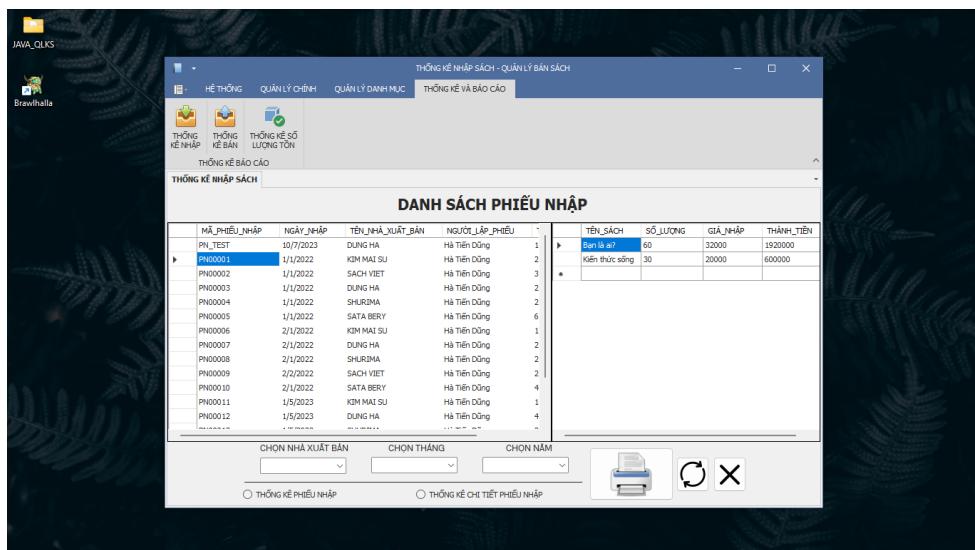


Hình 6.26: Giao diện quản lí thể loại



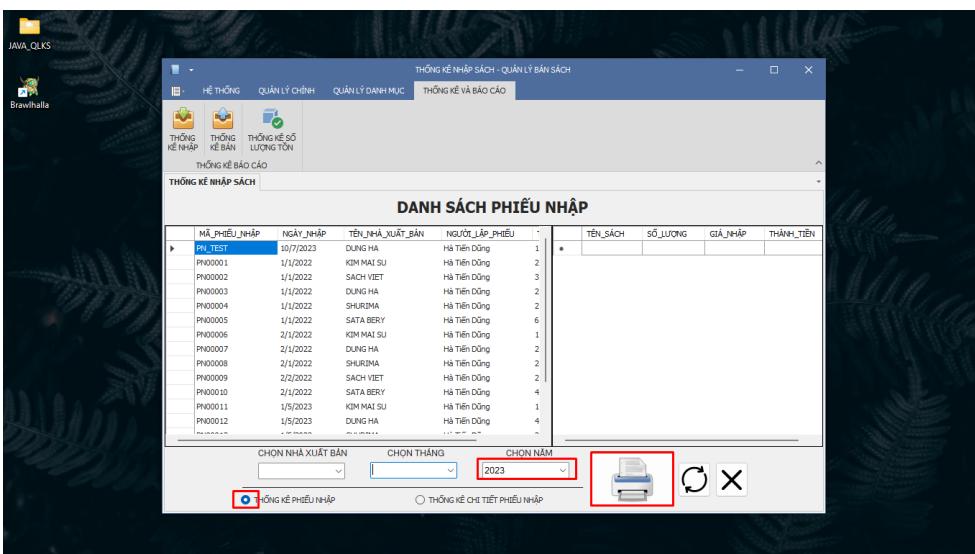
Hình 6.27: Giao diện quản lí tác giả

- Chức năng thống kê nhập

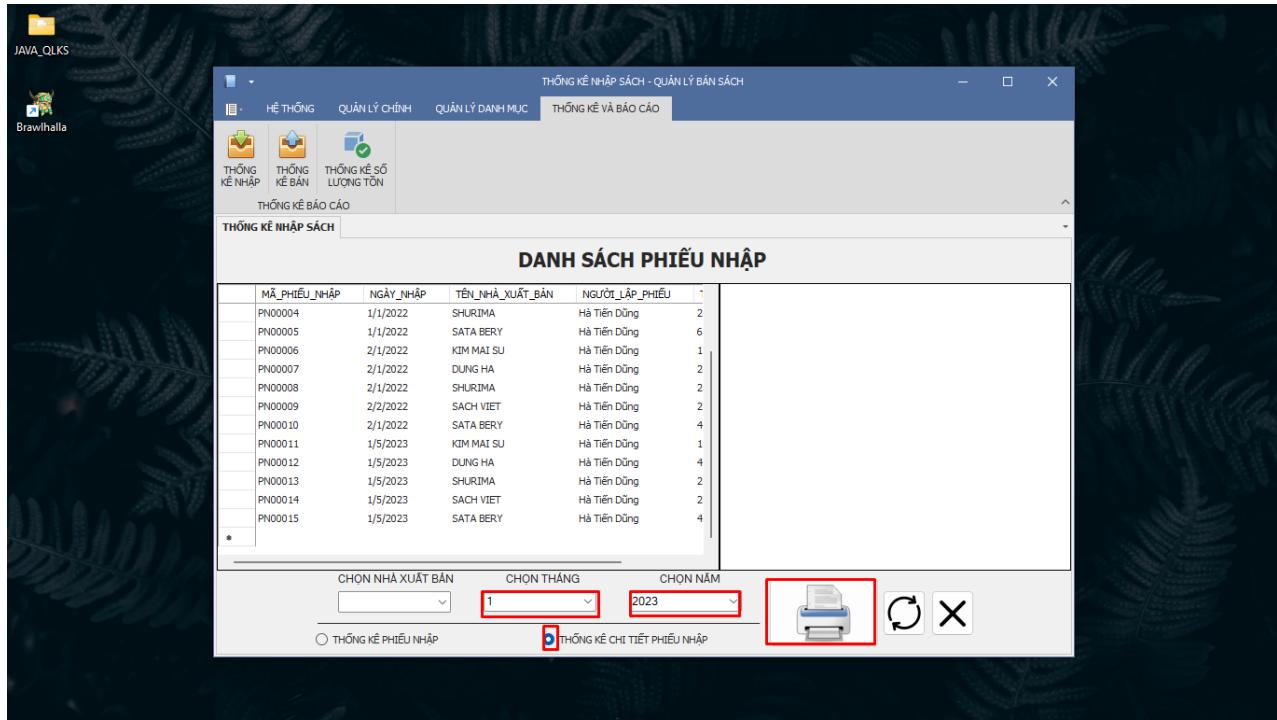


Hình 6.28: Giao diện thống kê nhập

- Chọn các thông tin muốn thống kê sau đó nhấn biểu tượng máy in



Hình 6.29: chọn các mục cần thiết và ấn in



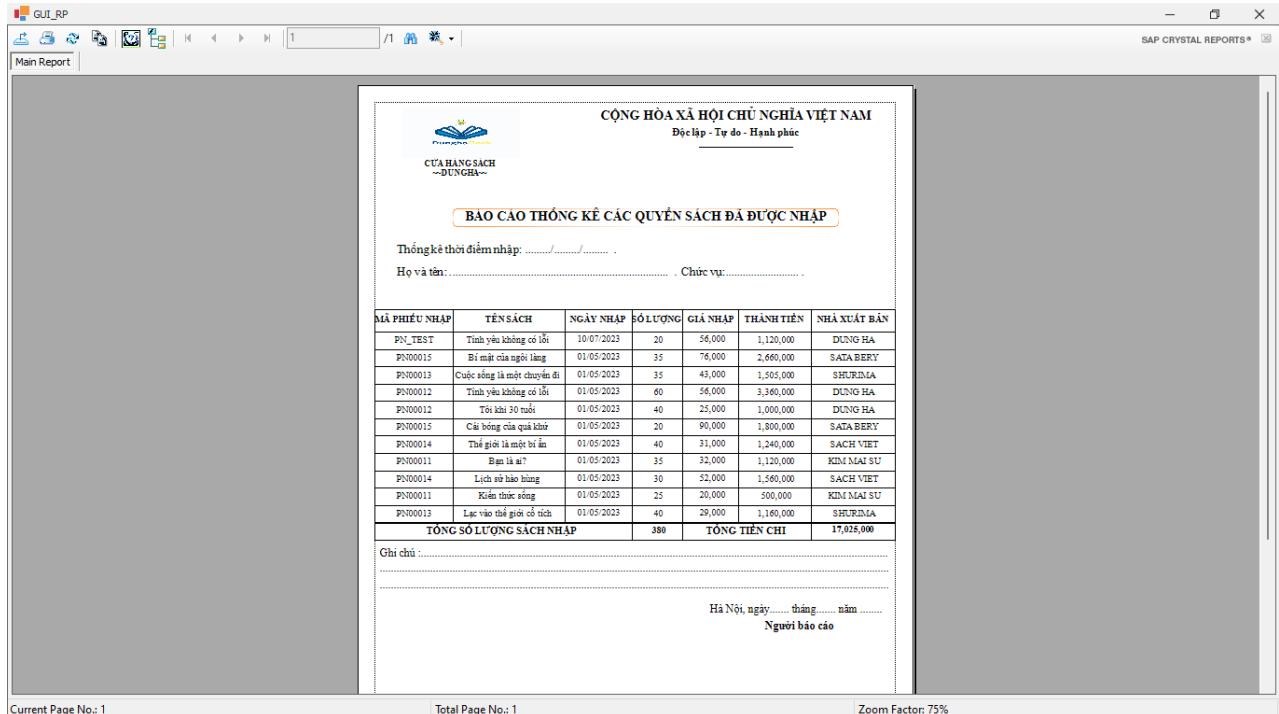
Hình 6.30: chọn các mục cần thiết và ấn in

- Kết quả thống kê phiếu nhập 2023:

MÃ PHIẾU NHẬP	NGÀY NHẬP	TÊN NHÀ XUẤT BẢN	NHÂN VIÊN TẠO	TỔNG TIỀN
PN TEST	10/07/2023	DUNG HA	Hà Tiên Dũng	1,120,000
PN00011	01/05/2023	KIM MAI SU	Hà Tiên Dũng	1,620,000
PN00012	01/05/2023	DUNG HA	Hà Tiên Dũng	4,360,000
PN00013	01/05/2023	SHURIMA	Hà Tiên Dũng	2,665,000
PN00014	01/05/2023	SACH VIET	Hà Tiên Dũng	2,800,000
PN00015	01/05/2023	SATA BERY	Hà Tiên Dũng	4,460,000
Tổng tiền chi cho việc nhập sách				17,025,000

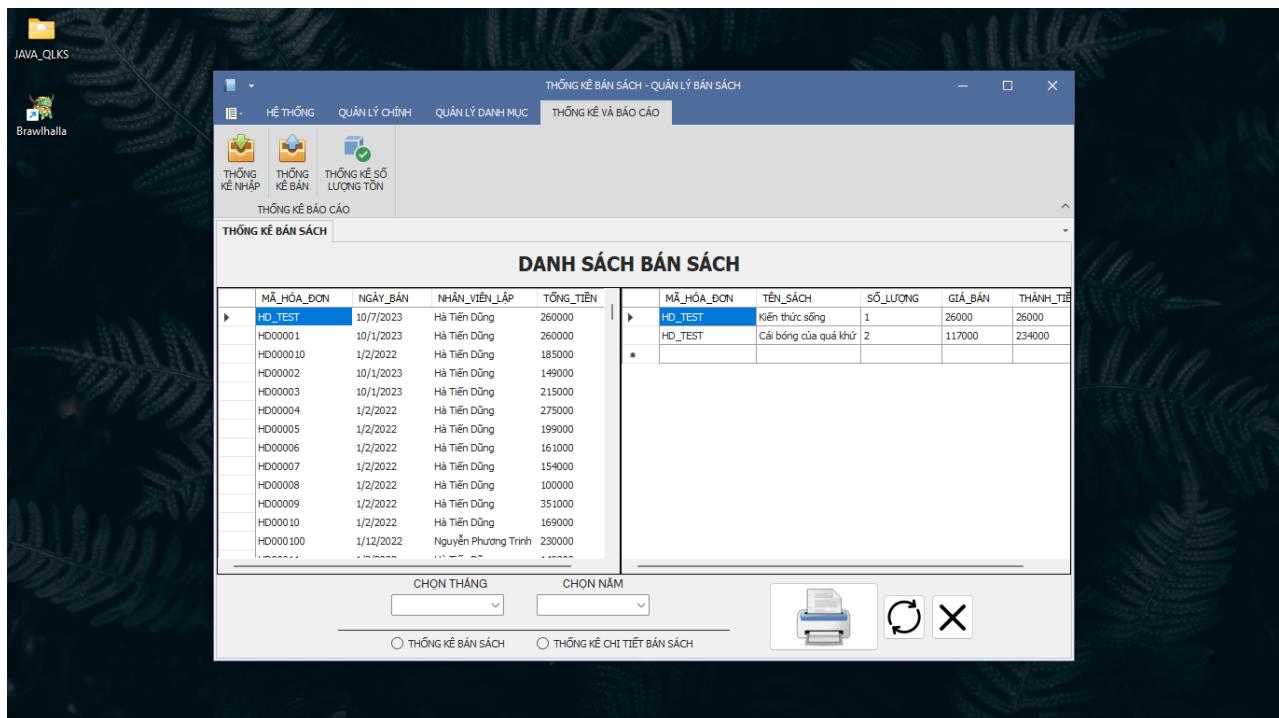
Hình 6.31: Kết quả sau khi ấn in

- Kết quả thông kê chi tiết phiếu nhập tháng 1 năm 2023



Hình 6.32: Kết quả sau khi ấn in

- Chức năng thống kê bán
 - Cũng tương tự như chức năng thống kê nhập chọn các mục muốn thống kê và ấn nút in thống kê.



Hình 6.33: Giao diện thống kê bán sách

- Thống kê bán sách năm 2023

The screenshot shows a SAP Crystal Reports interface with the following details:

- Report Title:** CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM
Độc lập - Tự do - Hạnh phúc
CỦA HÀNG SÁCH
--DUNGHA--
- Section:** BÁO CÁO TÌNH HÌNH CÔNG VIỆC BÁN SÁCH
- Text Fields:** Thông kê thời điểm nhập: / /
Họ và tên: Chức vụ:
- Table:** A table showing book sales data:

MÃ_HÓA_DON	NGÀY_BÁN	NHÂN_VIÊN_LAP	TỔNG TIỀN
HD_TEST	10/07/2023	Hà Tiên Dũng	260,000
HD00001	10/01/2023	Hà Tiên Dũng	260,000
HD00002	10/01/2023	Hà Tiên Dũng	149,000
HD00003	10/01/2023	Hà Tiên Dũng	215,000
			884,000
- Text Fields:** Ghi chú:
- Text at Bottom:** Hà Nội, ngày tháng năm
Người báo cáo

Hình 6.34: Kết quả in thống kê bán sách năm 2023

- Thống kê chi tiết các quyền sách đã bán năm 2023

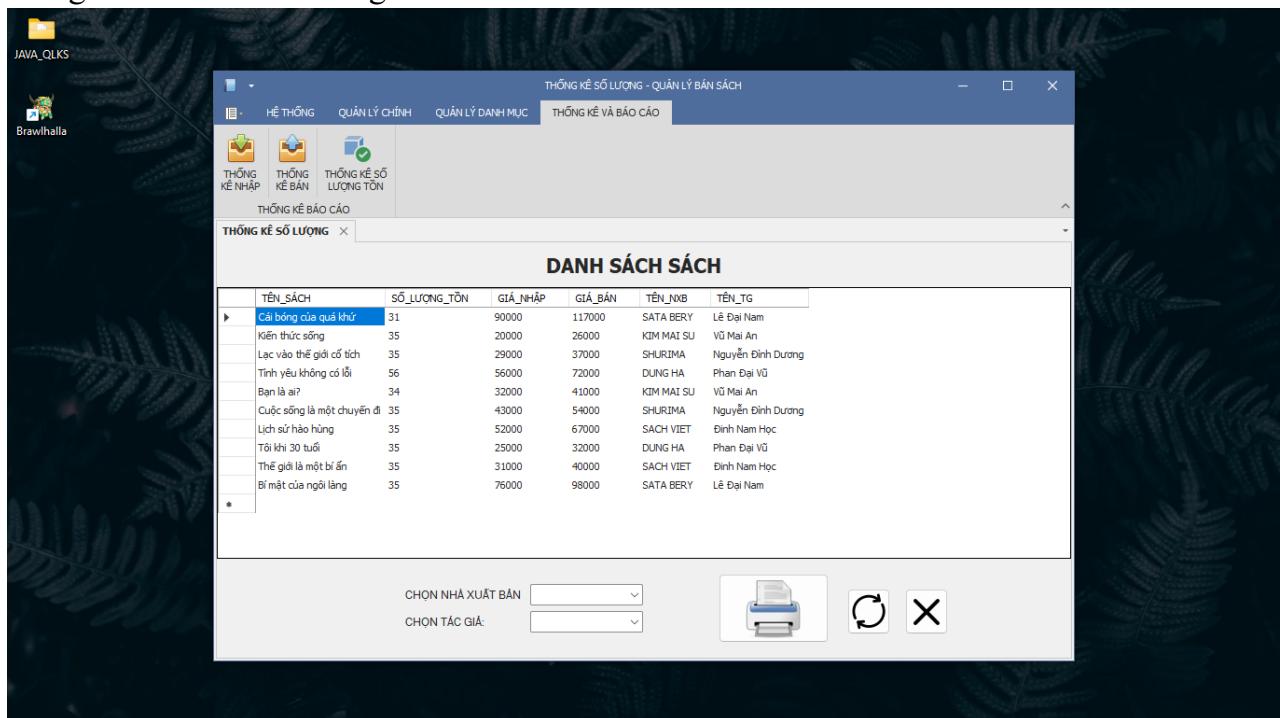
The screenshot shows a SAP Crystal Reports interface with the following details:

- Report Title:** CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM
Độc lập - Tự do - Hạnh phúc
CỦA HÀNG SÁCH
--DUNGHA--
- Section:** BÁO CÁO CHI TIẾT NHỮNG QUYỀN SÁCH ĐÃ BÁN
- Text Fields:** Thông kê thời điểm nhập: / /
Họ và tên: Chức vụ:
- Table:** A table showing detailed book sales data:

MÃ_HÓA_DON	TÊN SÁCH	SỐ LƯỢNG	GIÁ BÁN	THÀNH TIỀN
HD_TEST	Cái bóng của quá khứ	2	117,000	234,000
HD_TEST	Kiến thức sóng	1	26,000	26,000
HD00001	Cái bóng của quá khứ	2	117,000	234,000
HD00001	Kiến thức sóng	1	26,000	26,000
HD00002	Ban là ai?	2	41,000	82,000
HD00002	Lịch sử hào hùng	1	67,000	67,000
HD00003	Cái bóng của quá khứ	1	117,000	117,000
HD00003	Kiến thức sóng	1	26,000	26,000
HD00003	Tinh yêu không có lối	1	72,000	72,000
	TỔNG SỐ LƯỢNG	12	TỔNG TIỀN THU	884,000
- Text Fields:** Ghi chú:
- Text at Bottom:** Hà Nội, ngày tháng năm
Người báo cáo

Hình 6.35: Kết quả in thống kê chi tiết bán sách năm 2023

- Chức năng thống kê số lượng tồn
 - Cũng tương tự như chức năng thống kê nhập và thống kê bán chọn các mục muốn thống kê và ấn nút in thống kê.



Hình 6.36: Giao diện thống kê số lượng tồn

- Kết quả báo cáo số lượng tồn hiện tại

The screenshot shows a Crystal Reports print preview window titled 'Main Report'. The report has a header section with the logo of 'CÔNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM' and the motto 'Độc lập - Tự do - Hạnh phúc'. The main title of the report is 'BÁO CÁO SỐ LƯỢNG TỒN'. Below the title, there are fields for 'Thời điểm thống kê:/...../.....', 'Họ và tên: Chức vụ:,', and a note 'Ghi chú:'. The central part of the report is a table with columns: TÊN SÁCH, GIÁ NHẬP, GIÁ BÁN, tenNXB, tenTG, and SỐ LƯỢNG TỒN. The table lists the same books as in the previous screenshot. At the bottom of the report, there is a signature line with the text 'Hà Nội, ngày tháng năm' and 'Người báo cáo'.

Hình 6.37: Kết quả in thống kê số lượng tồn

Hạn chế và hướng phát triển của đề tài.

Chương trình ứng dụng trong quá trình lập trình có thể sẽ không tối ưu do chưa được tiếp cận với các dự án thực tế.

Nhóm em sẽ cập nhật và nâng cấp lại hệ thống khi đã có nhiều kiến thức mới và được tiếp cận với nhiều thứ mới hơn sau này.

Tài liệu tham khảo

https://nvdanh.files.wordpress.com/2011/09/nhom5_08cdtp2_lancuoi.pdf

Tham khao - Quan ly thu vien - Đò án mẫu.pdf - Google Drive

Một số kiến thức lý thuyết tham khảo qua công cụ BingAi, Bard