**BỘ GIÁO DỤC VÀ ĐÀO TẠO**

**TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP. HỒ CHÍ MINH KHOA CÔNG NGHỆ THÔNG TIN**

**--------------------**

****

ĐỒ ÁN MÔN HỌC

**HỆ QUẢN TRỊ CSDL**

**QUẢN LÝ THƯ VIỆN (PHẦN QUẢN LÝ NHẬP SÁCH)**

**SINH VIÊN THỰC HIỆN:**

23110194 – Phan Ngọc Duy – Nhóm 03

**GIÁO VIÊN HƯỚNG DẪN:**

TS. Nguyễn Thành Sơn

**LỚP HP:**  DBMS330284

TP. HỒ CHÍ MINH – 2025

**MỤC LỤC**

[CHƯƠNG 1: TỔNG QUAN VỀ HỆ THỐNG 1](#_Toc209504212)

[**1.** **Đặc tả đề tài** 1](#_Toc209504213)

[**1.1.** **Mô tả bài toán** 1](#_Toc209504214)

[**1.2.** **Mô tả dữ liệu** 2](#_Toc209504215)

[**1.3.** **Mô tả chức năng** 6](#_Toc209504216)

[**1.4.** **Về phần giao diện** 7](#_Toc209504217)

[CHƯƠNG 2: PHÂN TÍCH THIẾT KẾ HỆ THỐNG 9](#_Toc209504218)

[**1.** **Thiết kế cơ sở dữ liệu mức quan niệm** 9](#_Toc209504219)

[**2.** **Thiết kế cơ sở dữ liệu mức logic** 11](#_Toc209504220)

[**3.** **Các ràng buộc cần có** 12](#_Toc209504221)

[**4.** **Cài đặt SQL và các ràng buộc** 15](#_Toc209504222)

[**4.1. Các bảng thuộc đề tài** 15](#_Toc209504223)

[**4.2. Các bảng cần tham chiếu của thành viên nhóm** 16](#_Toc209504224)

[**5.** **Lược đồ quan hệ sau khi cài đặt SQL** 18](#_Toc209504225)

[**6.** **Các view** 19](#_Toc209504226)

[**7.** **Các trigger** 23](#_Toc209504227)

[CHƯƠNG 3: THIẾT KẾ CÁC CHỨC NĂNG 26](#_Toc209504228)

[**1.** **Kết nối cơ sở dữ liệu** 26](#_Toc209504229)

[**2.** **Các chức năng chung của Admin và Nhân viên** 28](#_Toc209504230)

[**2.1.** **Nhập sách vào bảng nhập** 28](#_Toc209504231)

[**2.2 Cập nhật thông tin sách nhập** 31](#_Toc209504232)

[**2.3.** **Xóa sách khỏi bảng nhập** 34](#_Toc209504233)

[**2.4.** **Thêm ảnh bìa vào sách** 36](#_Toc209504234)

[**2.5.** **Cập nhật thêm bớt số lượng sách vào kho sau khi đã nhập sách** 38](#_Toc209504235)

[**2.6.** **Kiểm tra trạng thái sách trong kho sau khi đã nhập** 41](#_Toc209504236)

[**2.7.** **Thống kê sách đã nhập** 43](#_Toc209504237)

[**2.8.** **Thống kê ngày đã nhập** 44](#_Toc209504238)

[**3.** **Chức năng riêng của Admin** 45](#_Toc209504239)

[CHƯƠNG 4: PHÂN QUYỀN 47](#_Toc209504240)

[CHƯƠNG 5: TRANSACTION 49](#_Toc209504241)

[CHƯƠNG 6: THIẾT KẾ GIAO DIỆN 61](#_Toc209504242)

[**1.** **Môi trường cài đặt và công nghệ sử dụng** 61](#_Toc209504243)

[**2.** **Phân quyền vai trò đăng nhập** 61](#_Toc209504244)

[**3.** **Phân quyền chức năng** 62](#_Toc209504245)

[**3.1.** **Chức năng chung của Admin và Nhân viên** 62](#_Toc209504246)

[**3.2.** **Chức năng riêng của Admin** 71](#_Toc209504247)

**DANH MỤC HÌNH**

[Hình 1: Form đăng nhập vai trò Admin 61](#_Toc209505404)

[Hình 2: Form đăng nhập vai trò Nhân viên 62](#_Toc209505405)

[Hình 3: Form trang chủ của hệ thống quản lý nhập sách 62](#_Toc209505406)

[Hình 4: Form kho sách của hệ thống quản lý nhập sách 63](#_Toc209505407)

[Hình 5: Form kho sách thực hiện chức năng kiểm tra kho 63](#_Toc209505408)

[Hình 6: Form kho sách thực hiện chức năng cập nhật kho 64](#_Toc209505409)

[Hình 7: Form nhập sách của hệ thống quản lý nhập sách 64](#_Toc209505410)

[Hình 8: Form nhập sách vào bảng nhập sách 65](#_Toc209505411)

[Hình 9: Form nhập sách thành công vào bảng nhập sách 65](#_Toc209505412)

[Hình 10: Form nhập sách cập nhật thông tin 66](#_Toc209505413)

[Hình 11: Form nhập sách xóa sách nhập khỏi bảng nhập sách 66](#_Toc209505414)

[Hình 12: Form tra cứu của hệ thống quản lý nhập sách 67](#_Toc209505415)

[Hình 13: Form tra cứu sách của hệ thống quản lý nhập sách 67](#_Toc209505416)

[Hình 14: Form tra cứu tác giả của hệ thống quản lý nhập sách 68](#_Toc209505417)

[Hình 15: Form tra cứu thể loại của hệ thống quản lý nhập sách 68](#_Toc209505418)

[Hình 16: Form tra cứu nhà xuất bản của hệ thống quản lý nhập sách 69](#_Toc209505419)

[Hình 17: Form tra cứu lịch sử nhập kho của hệ thống quản lý nhập sách 69](#_Toc209505420)

[Hình 18: Form thống kê của hệ thống quản lý nhập sách 70](#_Toc209505421)

[Hình 19: Form thống kê theo sách đã nhập 70](#_Toc209505422)

[Hình 20: Form thống kê theo ngày đã nhập 71](#_Toc209505423)

[Hình 21: Form nhập sách phân quyền NhanVienRole không được xác nhận nhập vào kho 71](#_Toc209505424)

[Hình 22: Form nhập sách phân quyền AdminRole được xác nhận nhập vào kho 72](#_Toc209505425)

[Hình 23: Form nhập sách sau khi được Admin xác nhận nhập 72](#_Toc209505426)

# LỜI CẢM ƠN

Kính thưa thầy Nguyễn Thành Sơn,

Em xin gửi lời cảm ơn chân thành và sâu sắc đến thầy vì sự chỉ bảo tận tình và giúp đỡ quý báu trong suốt quá trình hoàn thiện đồ án môn Hệ quản trị cơ sở dữ liệu. Nhờ vào sự hướng dẫn nhiệt huyết của thầy, em đã có thể hoàn thành đồ án một cách hiệu quả và đạt được kết quả như mong đợi.

Em rất trân trọng và biết ơn thầy đã dành thời gian quý báu để chia sẻ những kiến thức chuyên sâu, cũng như kinh nghiệm quý giá. Những chỉ dẫn của thầy không chỉ giúp em hoàn thành đồ án mà còn trang bị cho em những kỹ năng cần thiết để làm việc với cơ sở dữ liệu trong tương lai.

Một lần nữa, em xin gửi lời cảm ơn sâu sắc đến thầy Nguyễn Thành Sơn vì sự dìu dắt và tất cả những gì thầy đã làm giúp em trong suốt quá trình học tập.

Trân trọng, Phan Ngọc Duy

# LỜI NÓI ĐẦU

Xin chào thầy và các bạn,

Đồ án “Quản lý thư viện UTE – Phần quản lý nhập sách” là thành quả của em sau quá trình học tập, nghiên cứu và thực hành môn Hệ quản trị cơ sở dữ liệu. Đây là một dự án đầy thách thức, nhưng cũng mở ra cơ hội để em được tiếp cận, áp dụng kiến thức vào thực tiễn và tích lũy những kinh nghiệm quý báu.

Trong quá trình thực hiện, em đã đối mặt với nhiều khó khăn, từ việc thiết kế hệ thống đến triển khai các chức năng cụ thể. Tuy nhiên, nhờ sự hướng dẫn tận tâm của thầy cùng với tinh thần làm việc nghiêm túc và đoàn kết, đồ án đã được hoàn thiện một cách trọn vẹn và chất lượng.

Trong bối cảnh giáo dục hiện đại, thư viện đóng vai trò quan trọng trong việc hỗ trợ học tập và nghiên cứu. Tại Trường Đại học Sư phạm Kỹ thuật TPHCM (HCMUTE), nhu cầu quản lý sách và tài liệu ngày càng trở nên cấp thiết, đặc biệt là trong khâu nhập sách – một công đoạn nền tảng để đảm bảo nguồn tài liệu phong phú và chất lượng cho người dùng. Đề tài “Quản lý thư viện UTE – Phần quản lý nhập sách” ra đời với mục tiêu xây dựng một hệ thống quản lý hiệu quả, giúp tự động hóa và tối ưu hóa quy trình nhập sách vào kho, từ đó nâng cao hiệu suất hoạt động của thư viện và hỗ trợ tốt hơn cho sinh viên, giảng viên.

Đồ án này không chỉ là một sản phẩm học thuật, mà còn mang tính ứng dụng cao, giúp thư viện UTE quản lý thông tin sách nhập, kho sách và các hoạt động nhập sách một cách khoa học và thuận tiện. Bằng cách áp dụng kiến thức về Hệ quản trị cơ sở dữ liệu, em đã xây dựng một hệ thống tích hợp các chức năng như nhập sách mới vào kho, theo dõi kiểm tra quá trình nhập kho, báo cáo thống kê, … góp phần đơn giản hóa công việc quản lý và nâng cao hiệu quả vận hành.

Em hy vọng rằng đồ án này không chỉ mang lại giá trị thực tiễn cho thư viện UTE, mà còn là nguồn cảm hứng, tham khảo hữu ích cho quý thầy cô và các bạn sinh viên trong việc ứng dụng công nghệ thông tin vào quản lý thực tế. Mọi ý kiến đóng góp từ quý thầy cô và các bạn sẽ là nguồn động lực quý giá để em tiếp tục hoàn thiện và phát triển dự án. Trân trọng cảm ơn sự quan tâm và hỗ trợ của quý thầy cô cùng các bạn sinh viên!

# CHƯƠNG 1: TỔNG QUAN VỀ HỆ THỐNG

1. **Đặc tả đề tài**
   1. **Mô tả bài toán**

Hệ thống Quản lý nhập sách giải quyết vấn đề bằng cách kết nối giữa quản lý thư viện

và nhân viên vận hành do đó nó sẽ có những nhóm đối tượng sau:

Thư viện trưởng (Admin): đây là đối tượng đại diện thư viện và thực hiện quản lý toàn diện quy trình nhập sách lên hệ thống. Họ sẽ tạo tài khoản và phân quyền cho nhân viên, duyệt và xác nhận nhập kho các phiếu nhập sách một cách cẩn trọng để đảm bảo tính chính xác và đầy đủ thông tin, quản lý toàn bộ danh mục sách bao gồm chi tiết về tác giả, thể loại, nhà xuất bản cùng với các thuộc tính như năm xuất bản, giá nhập và ảnh bìa, đồng thời theo dõi báo cáo thống kê toàn diện về tình hình nhập sách, tồn kho cũng như dự báo nhu cầu bổ sung tài liệu với mức độ bảo mật phù hợp, giúp duy trì sự cân bằng giữa nguồn lực và nhu cầu sử dụng của thư viện.

Nhân viên: đây là nhóm đối tượng thuộc sự quản lý của Admin, nhân viên sẽ thực hiện công việc nhập thông tin sách mới, bao gồm việc kiểm tra và cập nhật thông tin phiếu nhập trước khi xác nhận nhập để tránh sai sót về số lượng hay giá nhập. Quản lý kho sách bằng cách cập nhật số lượng tồn kho thời gian thực và kiểm tra trạng thái sách trong kho (còn sách hoặc hết sách) ngay sau khi quá trình nhập kho của sách đã kết thúc như đúng với quy trình, đồng thời hỗ trợ các nhiệm vụ bổ sung như tra cứu sách theo thể loại và tác giả, lịch sử nhập kho ngoài ra việc thống kê báo cáo sách nhập cùng với ngày nhập giúp quy trình vận hành hàng ngày diễn ra suôn sẻ và hiệu quả.

Mục tiêu của hệ thống là quản lý nhập sách vào kho một cách tự động và hiệu quả quy trình nhập sách từ khâu quản lý danh mục sách chi tiết, xác nhận nhập kho với kiểm soát chặt chẽ, cập nhật tồn kho thời gian thực hay báo cáo thống kê đa chiều, không chỉ giảm thiểu các lỗi thủ công như trùng lặp thông tin hoặc sai lệch số lượng mà còn tiện lợi cho việc theo dõi, phân tích dữ liệu chi tiết về xu hướng nhập sách theo thời gian, thể loại hoặc tác giả, qua đó giúp cho thư viện đạt được mục đích bổ sung nguồn tài liệu phong phú, đa dạng và phù hợp với nhu cầu học thuật, đồng thời cung cấp cho người đọc trải nghiệm dịch vụ tìm kiếm, mượn sách minh bạch, nhanh chóng và hiệu quả hơn bao giờ hết.

* 1. **Mô tả dữ liệu**

- **Mô tả các tập thực thể**

Hệ thống Quản lý nhập sách sẽ cung cấp dịch vụ quản lý nhập sách đối với nhiều loại tài liệu khác nhau. Ở đây sẽ chia thành 2 loại người dùng chính đó là: Admin và Nhân viên. Mỗi loại người dùng sẽ chỉ được thực hiện các chức năng phù hợp với vai trò của mình, điều này giúp hệ thống dễ dàng phân quyền và kiểm soát quy trình một cách hợp lý.

Quản lý nhập sách không thể thiếu dữ liệu về tài liệu. Trong hệ thống này, mỗi sách sẽ có các thông tin quan trọng như mã sách, tên sách, tác giả, thể loại, nhà xuất bản, năm xuất bản, giá nhập, ngày nhập, hình ảnh bìa. Mỗi đầu sách chỉ được nhập vào hệ thống theo thời điểm xác định, tùy thuộc vào phiếu nhập mà thư viện đã tạo, nếu sách đó đã được xác nhận nhập kho thì nó sẽ được cập nhật vào tồn kho và sẵn sàng cho việc mượn.

Ví dụ: nhân viên tạo phiếu nhập TN001 cho 50 cuốn sách "Cơ sở dữ liệu" trong ngày 20/09/2025, thì 50 cuốn này sẽ được ghi nhận trong phiếu nhập với trạng thái "Chưa nhập" do vậy chưa thể có sách vào kho được. Sau khi Admin xem xét kỹ lượng phiếu nhập xem có đúng thông tin sách cần nhập không và duyệt xác nhận, hệ thống sẽ cập nhật vào kho với số lượng 50 cuốn và trạng thái "Đã nhập" do vậy sinh viên có thể mượn được. Sau khi sinh viên mượn hết 50 cuốn, hệ thống sẽ tự động cập nhật số lượng còn lại là 0 và trạng thái thành "Hết sách" do vậy không thể mượn thêm cuốn nào nữa. Sau khi sinh viên trả sách, số lượng trong kho sẽ tăng trở lại và trạng thái kho lúc này sẽ hiển thị là “Còn sách” sinh viên có thể mượn tiếp.

Khi nhân viên nhập sách sẽ tiến hành tạo phiếu nhập, thông tin phiếu nhập bao gồm: mã nhân viên, mã phiếu nhập, mã sách, tên sách, tác giả, thể loại, nhà xuất bản, năm xuất bản, ngày nhập, số lượng, giá nhập, thành tiền, trạng thái. Và sẽ có một mã nhân viên để cho biết rằng phiếu nhập này do nhân viên nào thực hiện và chịu trách nhiệm kiểm tra thông tin.

Trong hệ thống quản lý nhập sách, các hoạt động như nhập thông tin, xác nhận nhập kho hoặc các nhiệm vụ khác đều được quản lý thông qua các phiếu nhập cụ thể. Mỗi phiếu nhập sẽ có trạng thái phiếu, ngày tạo phiếu và sẽ được gán cho một nhân viên thực hiện. Phiếu nhập này cũng sẽ liên kết với sách và nhân viên để phân biệt rõ nhiệm vụ, đảm bảo

tính trách nhiệm trong quy trình.

Mỗi phiếu nhập sẽ cần phải được xác nhận bởi Admin, điều này sẽ dễ dàng cho việc tạo số liệu tổng kết về nguồn tài liệu mới, xác nhận nhập bao gồm: mã phiếu, trạng thái xác nhận, ngày xác nhận. Đồng thời sẽ kèm thêm mã phiếu nhập để xác định chính xác cần xác nhận cho phiếu nào, giúp theo dõi lịch sử duyệt một cách rõ ràng và minh bạch.

Sau khi phiếu nhập được tạo và chờ xác nhận, nhân viên sẽ kiểm tra thông tin sách trước khi nhập kho. Thông tin của sách trước khi nhập sẽ được nhân viên lưu lại bao gồm: số lượng nhận được thực tế, giá trị tài liệu và trạng thái của sách. Mỗi lần nhập sách sẽ có một mã phiếu nhập để xác định nhiệm vụ đó được giao cho nhân viên nào và theo dõi tiến độ thực hiện một cách hiệu quả.

Khi Admin xác nhận nhập, nhân viên sẽ thực hiện việc cập nhật kho. Họ sẽ kiểm tra và ghi lại thông tin về kho như: số lượng hiện tại sau khi cộng vào, trạng thái kho (Còn sách/ Hết sách). Tương tự như nhập thông tin, quá trình cập nhật kho cũng có một mã phiếu nhập để xác định rõ phiếu nhập chuyên trách này và đảm bảo dữ liệu được cập nhật chính xác, tránh sai sót trong quản lý tồn kho.

Trong quá trình nhập sách, có thể xảy ra các sự cố phát sinh như thông tin sai sót, thiếu hụt số lượng. Nhân viên có thể linh động điều chỉnh lại thông tin sai sót trước khi chính thức nhập sách vào kho. Nếu nhân viên tạo thẻ nhập sách nhưng không muốn nhập nữa có thể xóa bỏ thẻ nhập đó nhưng việc này nên hạn chế để tránh phát sinh lỗi do tồn đọng thẻ nhập cũ. Từ đó có biện pháp xử lý kịp thời và ghi nhận kinh nghiệm để cải thiện quy trình.

Hệ thống cung cấp dịch vụ quản lý nhiều loại tài liệu khác nhau, từ sách chuyên ngành khoa học công nghệ đến giáo trình đại học. Mỗi loại tài liệu sẽ có một mã thể loại và những thông tin như mô tả, tên thể loại, và số lượng sách thuộc thể loại đó. Điều này giúp hệ thống dễ dàng phân loại và quản lý các tài liệu nhập mới, hỗ trợ việc sắp xếp và tìm kiếm hiệu quả trong thư viện.

Mỗi loại sách sẽ có một mức giá nhập riêng biệt. Hệ thống sẽ lưu trữ thông tin về giá nhập của từng sách, bao gồm năm xuất bản và giá nhập tương ứng. Điều này cho phép thư viện dễ dàng theo dõi chi phí nhập sách theo từng năm, lập kế hoạch ngân sách phù hợp và đánh giá hiệu quả đầu tư vào nguồn tài liệu một cách khoa học.

Về phần lưu trữ các thông tin của người dùng:

Tất cả người dùng trong hệ thống sẽ cần phải có tài khoản để đăng nhập và sử dụng dịch vụ. Tài khoản sẽ chứa các thông tin đăng nhập cơ bản như tên đăng nhập, mật khẩu đã mã hóa và vai trò của người dùng (Admin, Nhân viên). Vai trò này sẽ giúp hệ thống xác định quyền hạn của mỗi người dùng khi truy cập vào hệ thống. Đó là những thông tin cơ bản để giúp xác định người dùng nhằm biết danh tính của Admin và để quản lý nhân viên thực hiện các công việc nhập sách, đồng thời đảm bảo tính bảo mật và an toàn thông tin. Admin là người có quyền quản lý toàn bộ hệ thống và có quyền tạo, chỉnh sửa các thông tin quan trọng liên quan đến quy trình nhập sách và hoạt động của thư viện. Thông tin của Admin sẽ được lưu trữ trong hệ thống bao gồm họ tên, ngày sinh, email, số điện thoại. Mỗi Admin cũng sẽ có một tài khoản để đăng nhập vào hệ thống và quản lý hoạt động của mình, bao gồm việc duyệt xác nhận nhập kho, xem báo cáo thống kê và thiết lập chính sách quản lý tài liệu một cách chuyên nghiệp.

Nhân viên là những người đảm nhiệm các công việc như nhập thông tin sách, cập nhật phiếu nhập hoặc quản lý các tác vụ khác liên quan đến hoạt động nhập sách. Họ cũng có tài khoản đăng nhập vào hệ thống để thực hiện công việc. Thông tin của nhân viên được lưu trữ bao gồm họ tên, ngày sinh, email, số điện thoại và chức vụ. Mỗi nhân viên sẽ được phân công các công việc cụ thể thông qua hệ thống, đặc biệt là việc tạo phiếu nhập, kiểm tra thực tế sách nhận được và cập nhật kho sách sau khi xác nhận, đảm bảo quy trình vận hành trơn tru và hiệu quả.

- **Mô tả các mối quan hệ**

Mỗi người dùng có một tài khoản. Mối quan hệ này là 1 – 1 có nghĩa là một người dùng chỉ có thể có một tài khoản và ngược lại.

Người dùng chia làm 2 loại: Admin và Nhân viên.

Một tài khoản có thể liên kết với một Admin, nhưng một Admin chỉ có một tài khoản. Mối quan hệ này là 1 – 1 để đảm bảo mỗi thư viện trưởng có quyền truy cập riêng biệt.

Một tài khoản có thể liên kết với một Nhân viên, nhưng một Nhân viên chỉ có một tài khoản. Mối quan hệ này cũng là 1 – 1 đảm bảo mỗi thủ thư có quyền truy cập phù hợp với vai trò.

Một Nhân viên có thể tạo và quản lý nhiều phiếu nhập, nhưng một phiếu nhập chỉ do một nhân viên thực hiện. Mối quan hệ này là 1 – N (một nhân viên xử lý nhiều phiếu, một phiếu chỉ do một nhân viên chịu trách nhiệm) để theo dõi rõ ràng trách nhiệm trong quy trình nhập sách.

Một Nhân viên phải liên quan đến việc tạo phiếu nhập hoặc cập nhật kho. Đây là hai loại nhiệm vụ khác nhau trong quy trình quản lý nhập sách.

Mỗi sách có thể xuất hiện trong nhiều phiếu nhập, trong đó mỗi phiếu nhập sẽ liên quan đến một sách cụ thể, chứng tỏ cùng một đầu sách có thể được nhập nhiều lần theo các lô khác nhau.

Mỗi phiếu nhập có thể được một Admin xác nhận (quan hệ N – 1) nhưng mỗi xác nhận chỉ thuộc về một phiếu nhập. Điều này đảm bảo quy trình 2 bước: nhân viên tạo phiếu

→ Admin duyệt xác nhận.

Mỗi sách thuộc về một tác giả, nghĩa là một tác giả có thể có nhiều sách nhưng một sách chỉ có một tác giả chính.

Mỗi sách thuộc về một thể loại, nghĩa là một thể loại bao gồm nhiều sách nhưng một sách chỉ thuộc một thể loại chính.

Mỗi sách thuộc về một nhà xuất bản, nghĩa là một nhà xuất bản có thể xuất bản nhiều sách nhưng một sách chỉ do một nhà xuất bản phát hành.

Mỗi sách có thể có nhiều bản ghi trong kho sách theo từng lần nhập khác nhau,

nghĩa là một sách có thể có nhiều lô tồn kho với số lượng và trạng thái khác nhau.

Mỗi phiếu nhập khi được xác nhận sẽ tự động cập nhật vào kho sách, tạo mối quan hệ gián tiếp giữa phiếu nhập và tồn kho thực tế, đảm bảo dữ liệu đồng bộ thời gian thực.

* 1. **Mô tả chức năng**

Có hai nhóm người dùng chính là: Admin và Nhân viên

**Chức năng chung:**

Đăng nhập: Mỗi người dùng có tài khoản riêng để đăng nhập và truy cập các chức năng tương ứng với quyền hạn của mình.

Nhập sách mới vào kho: Tạo phiếu nhập sách với đầy đủ thông tin sách, tác giả, thể loại, nhà xuất bản, năm xuất bản, giá nhập và số lượng.

Cập nhật thông tin sách nhập: Chỉnh sửa thông tin phiếu nhập sách trước khi xác

nhận, bao gồm tên sách, tác giả, thể loại, nhà xuất bản, năm xuất bản và giá nhập.

Xóa sách nhập: Xóa phiếu nhập và thông tin sách khỏi hệ thống, bao gồm cả các danh mục liên quan nếu không còn sử dụng.

Thêm ảnh sách: Cập nhật hoặc thay đổi ảnh bìa cho sách trong hệ thống.

Kiểm tra trạng thái kho: Xem trạng thái kho sách, số lượng tồn kho hiện tại và tình

trạng còn sách hay hết sách cho từng cuốn.

Cập nhật kho: Điều chỉnh số lượng tồn kho, thêm hoặc giảm số lượng sách dựa trên

các giao dịch nhập.

Tra cứu: Tìm kiếm thông tin và xem danh sách sách chi tiết, thông tin tác giả, thể loại, nhà xuất bản; tra cứu lịch sử nhập kho và chi tiết phiếu nhập.

Thống kê: Tạo báo cáo về số lượng sách đã nhập theo khoảng thời gian hoặc theo ngày, đánh giá tổng quan kho sách.

**Chức năng admin phân quyền:**

Xác nhận nhập sách vào kho: Duyệt và xác nhận phiếu nhập từ nhân viên, cập nhật số lượng sách vào kho và thay đổi trạng thái phiếu nhập.

**Chú thích:**

Phân quyền quan trọng: Admin có quyền xác nhận nhập kho - chức năng cốt lõi

được bảo vệ, nhân viên không thể thực hiện mà chỉ có thể tạo và chỉnh sửa phiếu nhập.

Tự động hóa: Các mã định danh cho sách, phiếu nhập, tác giả, thể loại và nhà xuất bản được tạo tự động sau khi thêm mới.

Bảo mật dữ liệu: Hệ thống phân quyền chặt chẽ giữa admin và nhân viên, đảm bảo nhân viên không thể xác nhận nhập kho.

Hỗ trợ tra cứu: Các báo cáo và danh sách được thiết kế để hỗ trợ tra cứu nhanh và

nhất quán dữ liệu.

* 1. **Về phần giao diện**

Phần chung: Là những form mà tất cả người dùng đều thao tác được

Giao diện đăng nhập (LoginForm) cho phép người dùng truy cập vào hệ thống với các thành phần như: ô nhập tên đăng nhập, ô nhập mật khẩu (mật khẩu được ẩn đi), nút đăng nhập và nút hiển thị mật khẩu. Các ô nhập liệu được sắp xếp dọc ở trung tâm màn hình, với nút đăng nhập đặt ngay dưới ô nhập liệu.

Phần giao diện riêng: Bao gồm các phần giao diện con được quản lý trong một giao diện chính để tối ưu trải nghiệm.

Giao diện nhân viên chính (ThuThuForm) cho phép quản lý các chức năng như: nhập sách mới, quản lý kho sách, tra cứu sách, thống kê nhập sách và đăng xuất. Giao diện sử dụng các nút menu để tổ chức các chức năng trên một màn hình, bao gồm:

* Nút trang chủ: Hiển thị trang chính với thông tin chào mừng và hình ảnh thư viện.
* Nút nhập sách: Chuyển đến form nhập sách để tạo thẻ nhập mới.
* Nút kho sách: Chuyển đến form quản lý kho sách để kiểm tra và cập nhật kho.
* Nút tra cứu: Chuyển đến form trs cứu sách để tìm kiếm thông tin sách, tác giả, thể loại, nhà xuất bản, lịch sử nhập kho.
* Nút thống kê: Chuyển đến form thống kê nhập sách để xem báo cáo và biểu đồ.
* Nút đăng xuất: Đóng form hiện tại và quay về form đăng nhập.

Mỗi nút mở một form con riêng biệt trong panel chính, với bảng thông tin và các nút tương tác như thêm, sửa, xóa.

Giao diện nhập sách (NhapSachForm) cho phép người dùng tạo thẻ nhập sách mới, cập nhật thông tin và xác nhận nhập kho. Các thẻ nhập đã tạo sẽ được hiển thị trong bảng thông tin nhập sách, nơi người dùng có thể quản lý thẻ, xóa thẻ bằng nút xóa sách, hoặc xác nhận nhập kho qua nút xác nhận nhập chỉ dành cho admin vì đã phân quyền dựa theo vai trò, tạo nên giao diện trực quan và dễ sử dụng.

Giao diện kho sách (KhoSachForm) bao gồm các chức năng như kiểm tra kho, cập nhật kho. Form này hiển thị danh sách sách trong kho qua bảng kho sách, cùng với các nút thao tác như nút kiểm tra kho, nút cập nhật kho để điều chỉnh số lượng tồn kho.

Giao diện tra cứu sách (TraCuuSachForm) cho phép người dùng tra cứu sách theo các tiêu chí như sách, tác giả, thể loại, nhà xuất bản, lịch sử nhập kho. Người dùng có thể tìm kiếm qua ô tìm kiếm và nút tra cứu. Kết quả tra cứu sẽ được hiển thị trong bảng thông tin tra cứu, tạo nên giao diện trực quan và dễ sử dụng.

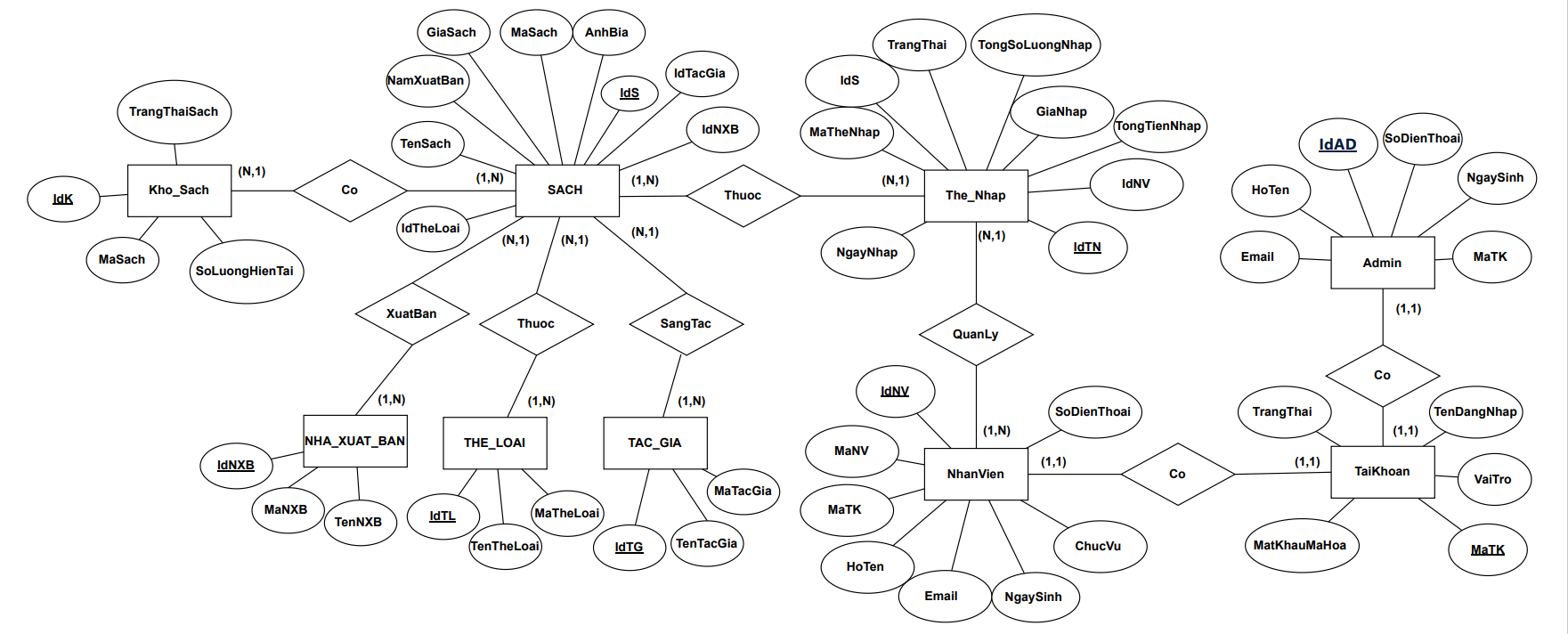
Giao diện thống kê nhập sách (ThongKeNhapSachForm) cho phép người dùng xem báo cáo thống kê sách đã nhập và thống kê theo theo ngày nhập. Người dùng có thể chọn khoảng thời gian qua ô ngày bắt đầu và ô ngày kết thúc và loại thống kê qua danh sách lựa chọn loại thống kê. Kết quả thống kê được hiển thị qua biểu đồ thống kê, kèm theo nút làm mới để cập nhật dữ liệu.

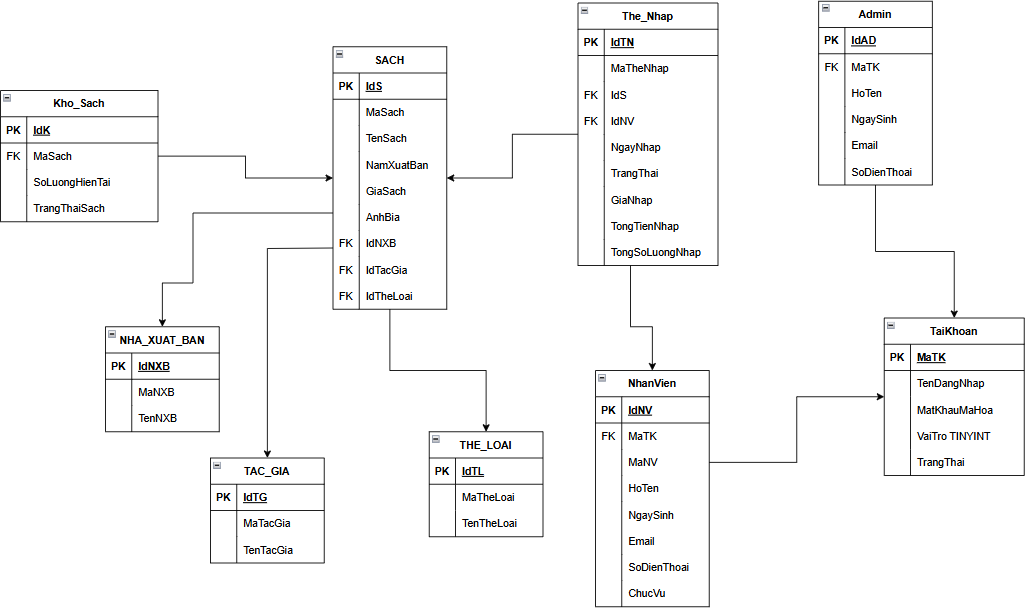
Việc xây dựng một hệ thống quản lý nhập sách thư viện là một bước tiến quan trọng trong việc tự động hóa và tối ưu hóa quy trình quản lý sách. Hệ thống không chỉ giúp tiết kiệm thời gian và nguồn lực mà còn nâng cao chất lượng phục vụ, mang lại sự hài lòng cho nhân viên. Bằng việc tích hợp các chức năng quản lý thông tin và phân tích dữ liệu, thư viện có thể dễ dàng điều hành và phát triển hơn trong môi trường hiện đại hiện nay.

# CHƯƠNG 2: PHÂN TÍCH THIẾT KẾ HỆ THỐNG

1. **Thiết kế cơ sở dữ liệu mức quan niệm**

Từ mô tả về dữ liệu cần có ở phần mô tả của bài toán ta hình thành được sơ đồ thực thể kết hợp (ERD).

****

1. **Thiết kế cơ sở dữ liệu mức logic**

TaiKhoan (MaTK, TenDangNhap, MatKhauMaHoa, VaiTro, TrangThai)

Admin (MaTK, IdAD, HoTen, NgaySinh, Email, SoDienThoai)

NhanVien (IdNV, MaNV, MaTK, HoTen, NgaySinh, Email, SoDienThoai, ChucVu)

TAC\_GIA (IdTG, MaTacGia, TenTacGia)

THE\_LOAI (IdTL, MaTheLoai, TenTheLoai)

NHA\_XUAT\_BAN (IdNXB, MaNXB, TenNXB)

SACH (IdS, MaSach, TenSach, NamXuatBan, GiaSach, AnhBia, IdNXB, IdTacGia, IdTheLoai)

The\_Nhap (IdTN, MaTheNhap, IdS, IdNV, NgayNhap, TrangThai, GiaNhap, TongTienNhap, TongSoLuongNhap)

Kho\_Sach (IdK, MaSach, SoLuongHienTai, TrangThaiSach)

1. **Các ràng buộc cần có**

|  |  |  |
| --- | --- | --- |
| **STT** | **BẢNG** | **RÀNG BUỘC** |
| 1 | **TaiKhoan**  MaTK INT  TenDangNhap VARCHAR(50) MatKhauMaHoa VARCHAR(255)  VaiTro TINYINT  TrangThai TINYINT | Khóa chính IDENTITY(1, 1) MaTK NOT NULL UNIQUE TenDangNhap NOT NULL MatKhauMaHoa  NOT NULL CHECK (VaiTro IN (0,1)) VaiTro NOT NULL DEFAULT 1 CHECK (TrangThai) IN (0,1,2)) TrangThai |
| 2 | **Admin**  IdAD INT MaTK INT  HoTen NVARCHAR(50)  NgaySinh DATE Email VARCHAR(50)  SoDienThoai VARCHAR(20) | Khóa chính IDENTITY(1, 1) IdAD  Khóa ngoại MaTK tham chiếu đến bảng  TaiKhoan (MATK)  NOT NULL HoTen  UNIQUE Email  UNIQUE NOT NULL SoDienThoai |
| 3 | **NhanVien**  IdNV INT  MaNV VARCHAR(50) MaTK INT  HoTen NVARCHAR(50)  NgaySinh DATE  Email VARCHAR(50)  SoDienThoai VARCHAR(20)  ChucVu VARCHAR(50) | Khóa chính IDENTITY(1, 1) IdNV NULL MaNV  Khóa ngoại MaTK tham chiếu đến bảng  TaiKhoan (MATK)  NOT NULL HoTen  UNIQUE Email  UNIQUE NOT NULL SoDienThoai  NOT NULL CHECK (ChucVu IN ('ThuThu',  'NhanVienPartTime', 'NhanVienFullTime')) ChucVu |
| 4 | **TAC\_GIA**  IdTG INT  MaTacGia VARCHAR(50) TenTacGia NVARCHAR(255) | Khóa chính IDENTITY(1, 1) IdTG NULL MaTacGia  NOT NULL TenTacGia |
| 5 | **THE\_LOAI**  IdTL INT  MaTheLoai VARCHAR(50) TenTheLoai NVARCHAR(255) | Khóa chính IDENTITY(1, 1) IdTL  NULL MaTheLoai  NOT NULL TenTheLoai |
| 6 | **NHA\_XUAT\_BAN**  IdNXB INT  MaNXB VARCHAR(50) TenNXB NVARCHAR(255) | Khóa chính IDENTITY(1, 1) IdNXB NULL MaNXB  NOT NULL TenNXB |
| 7 | **SACH**  IdS INT  MaSach VARCHAR(50) TenSach NVARCHAR(255)  NamXuatBan INT GiaSach DECIMAL(10, 2)  AnhBia VARBINARY(MAX) IdNXB INT  IdTacGia INT IdTheLoai INT | Khóa chính IDENTITY(1, 1) IdS NULL MaSach  NOT NULL TenSach  NOT NULL CHECK (NamXuatBan > 0)  NamXuatBan  NOT NULL DEFAULT 0.00 GiaSach  Khóa ngoại IdNXB tham chiếu đến bảng  NHA\_XUAT\_BAN (IdNXB)  Khóa ngoại IdTacGia tham chiếu đến bảng  TAC\_GIA (IdTG)  Khóa ngoại IdTheLoai tham chiếu đến bảng  THE\_LOAI (IdTL) |
| 8 | **The\_Nhap**  IdTN INT  MaTheNhap VARCHAR(50) IdS INT  IdNV INT  NgayNhap DATE  TrangThai VARCHAR(50)  GiaNhap DECIMAL(10, 2)  TongTienNhap DECIMAL(10, 2)  TongSoLuongNhap INT | Khóa chính IDENTITY(1, 1) IdTN NULL MaTheNhap  Khóa ngoại IdS tham chiếu đến bảng SACH(IdS)  Khóa ngoại IdNV tham chiếu đến bảng NhanVien(IdNV)  NOT NULL NgayNhap  NOT NULL CHECK (TrangThai IN ('DaNhap', 'ChuaNhap')) TrangThai  NOT NULL DEFAULT 0.00 CHECK (GiaNhap >= 0) GiaNhap  NOT NULL DEFAULT 0.00 TongTienNhap  NOT NULL CHECK (TongSoLuongNhap > 0) TongSoLuongNhap |
| 9 | **Kho\_Sach**  IdK INT  MaSach INT  SoLuongHienTai INT  TrangThaiSach VARCHAR(50) | Khóa chính IDENTITY(1, 1) IdK  Khóa ngoại MaSach tham chiếu đến bảng SACH(IdS)  NOT NULL DEFAULT 0 CHECK (SoLuongHienTai >= 0) SoLuongHienTai  NOT NULL DEFAULT 'HetSach' CHECK (TrangThaiSach IN ('ConSach', 'HetSach')) TrangThaiSach |

1. **Cài đặt SQL và các ràng buộc**

**4.1. Các bảng thuộc đề tài**

**Bảng TAC\_GIA**

CREATE TABLE TAC\_GIA (

IdTG INT PRIMARY KEY IDENTITY(1, 1), MaTacGia VARCHAR(50) NULL, TenTacGia NVARCHAR(255) NOT NULL

);

**Bảng THE\_LOAI**

CREATE TABLE THE\_LOAI (

IdTL INT PRIMARY KEY IDENTITY(1, 1), MaTheLoai VARCHAR(50) NULL, TenTheLoai NVARCHAR(255) NOT NULL

);

**Bảng NHA\_XUAT\_BAN**

CREATE TABLE NHA\_XUAT\_BAN (

IdNXB INT PRIMARY KEY IDENTITY(1, 1), MaNXB VARCHAR(50) NULL,

TenNXB NVARCHAR(255) NOT NULL

);

**Bảng SACH**

CREATE TABLE SACH (

IdS INT PRIMARY KEY IDENTITY(1, 1), MaSach VARCHAR(50) NULL,

TenSach NVARCHAR(255) NOT NULL,

NamXuatBan INT NOT NULL CHECK (NamXuatBan > 0), GiaSach DECIMAL(10, 2) NOT NULL DEFAULT 0.00, AnhBia VARBINARY(MAX),

IdNXB INT NOT NULL REFERENCES NHA\_XUAT\_BAN(IdNXB) ON DELETE NO ACTION, IdTacGia INT NOT NULL REFERENCES TAC\_GIA(IdTG) ON DELETE NO ACTION, IdTheLoai INT NOT NULL REFERENCES THE\_LOAI(IdTL) ON DELETE NO ACTION

);

**Bảng The\_Nhap**

CREATE TABLE The\_Nhap (

IdTN INT PRIMARY KEY IDENTITY(1, 1), MaTheNhap VARCHAR(50) NULL,

IdS INT NOT NULL REFERENCES SACH(IdS) ON DELETE NO ACTION,

IdNV INT NOT NULL REFERENCES NhanVien(IdNV) ON DELETE NO ACTION, NgayNhap DATE NOT NULL,

TrangThai VARCHAR(50) NOT NULL CHECK (TrangThai IN ('DaNhap', 'ChuaNhap')), GiaNhap DECIMAL(10, 2) NOT NULL DEFAULT 0.00 CHECK (GiaNhap >= 0), TongTienNhap DECIMAL(10, 2) NOT NULL DEFAULT 0.00,

TongSoLuongNhap INT NOT NULL CHECK (TongSoLuongNhap > 0)

);

**Bảng Kho\_Sach**

CREATE TABLE Kho\_Sach (

IdK INT PRIMARY KEY IDENTITY(1, 1),

MaSach INT NOT NULL REFERENCES SACH(IdS) ON DELETE CASCADE,

SoLuongHienTai INT NOT NULL DEFAULT 0 CHECK (SoLuongHienTai >= 0),

TrangThaiSach VARCHAR(50)NOT NULL DEFAULT 'HetSach' CHECK (TrangThaiSach IN ('ConSach', 'HetSach'))

);

**4.2. Các bảng cần tham chiếu của thành viên nhóm**

**Bảng TaiKhoan**

CREATE TABLE TaiKhoan(

MaTK INT PRIMARY KEY IDENTITY(1, 1), TenDangNhap VARCHAR(50) NOT NULL UNIQUE, MatKhauMaHoa VARCHAR(255) NOT NULL,

VaiTro TINYINT NOT NULL CHECK (VaiTro IN (0,1)), -- 0 = Admin, 1 = NhanVien

TrangThai TINYINT NOT NULL DEFAULT 1 CHECK (TrangThai IN(0,1,2)) -- 0 = KhoaVinhVien, 1 = HoatDong, 2 = TamKhoa

);

**Bảng Admin**

CREATE TABLE Admin(

IdAD INT PRIMARY KEY IDENTITY(1, 1),

MaTK INT NULL REFERENCES TaiKhoan(MaTK) ON DELETE SET NULL, HoTen NVARCHAR(50) NOT NULL,

NgaySinh DATE,

Email VARCHAR(50) UNIQUE,

SoDienThoai VARCHAR(20) UNIQUE NOT NULL,

);

**Bảng NhanVien**

CREATE TABLE NhanVien(

IdNV INT PRIMARY KEY IDENTITY(1, 1), MaNV VARCHAR(50) NULL,

MaTK INT NULL REFERENCES TaiKhoan(MaTK) ON DELETE SET NULL, HoTen NVARCHAR(50) NOT NULL,

NgaySinh DATE,

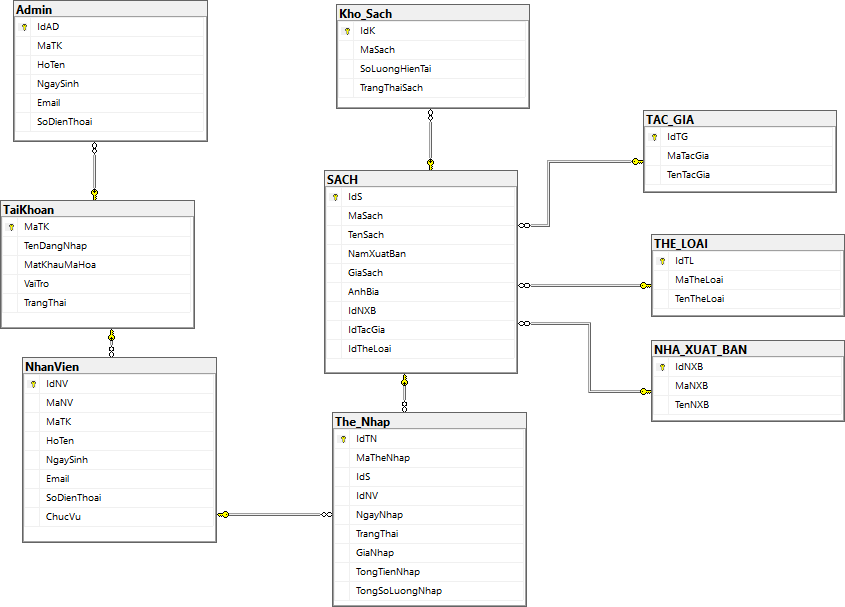
Email VARCHAR(50) UNIQUE,

SoDienThoai VARCHAR(20) UNIQUE NOT NULL,

ChucVu NVARCHAR(50) NOTNULL CHECK (ChucVu IN ('ThuThu','NhanVienPartTime', 'NhanVienFullTime'))

);

1. **Lược đồ quan hệ sau khi cài đặt SQL**



1. **Các view**
   1. **View Danh sách sách**

CREATE VIEW ViewDanhSachSach AS SELECT

s.MaSach, s.TenSach, t.TenTacGia, tl.TenTheLoai, nxb.TenNXB, s.NamXuatBan,

ISNULL(ks.SoLuongHienTai, 0) AS SoLuongHienTai, s.GiaSach,

CASE

WHEN ISNULL(ks.TrangThaiSach, 'HetSach') = 'ConSach' THEN N'Còn Sách' WHEN ISNULL(ks.TrangThaiSach, 'HetSach') = 'HetSach' THEN N'Hết Sách' ELSE N'Hết Sách'

END AS TrangThaiSach, s.AnhBia

FROM SACH s

JOIN TAC\_GIA t ON s.IdTacGia = t.IdTG JOIN THE\_LOAI tl ON s.IdTheLoai = tl.IdTL

JOIN NHA\_XUAT\_BAN nxb ON s.IdNXB = nxb.IdNXB

LEFT JOIN Kho\_Sach ks ON s.IdS = ks.MaSach;

* 1. **View Danh sách tác giả**

CREATE VIEW ViewDanhSachTacGia AS SELECT

t.MaTacGia, t.TenTacGia,

ISNULL(SUM(ks.SoLuongHienTai), 0) AS SoLuongSach FROM TAC\_GIA t

LEFT JOIN SACH s ON t.IdTG = s.IdTacGia

LEFT JOIN Kho\_Sach ks ON s.IdS = ks.MaSach GROUP BY t.MaTacGia, t.TenTacGia;

* 1. **View Danh sách thể loại**

CREATE VIEW ViewDanhSachTheLoai AS SELECT

tl.MaTheLoai, tl.TenTheLoai,

ISNULL(SUM(ks.SoLuongHienTai), 0) AS SoLuongSach FROM THE\_LOAI tl

LEFT JOIN SACH s ON tl.IdTL = s.IdTheLoai LEFT JOIN Kho\_Sach ks ON s.IdS = ks.MaSach

GROUP BY tl.MaTheLoai, tl.TenTheLoai;

* 1. **View danh sách nhà xuất bản**

CREATE VIEW ViewDanhSachNhaXuatBan AS SELECT

nxb.MaNXB, nxb.TenNXB,

ISNULL(SUM(ks.SoLuongHienTai), 0) AS SoLuongSach FROM NHA\_XUAT\_BAN nxb

LEFT JOIN SACH s ON nxb.IdNXB = s.IdNXB

LEFT JOIN Kho\_Sach ks ON s.IdS = ks.MaSach GROUP BY nxb.MaNXB, nxb.TenNXB;

* 1. **View Chi tiết nhập**

CREATE VIEW ViewChiTietTheNhap AS SELECT

nv.MaNV AS MaNhanVien, tn.MaTheNhap AS MaTheNhap, s.MaSach AS MaSach, s.TenSach AS TenSach, tg.TenTacGia AS TenTacGia, nxb.TenNXB AS NhaXuatBan, tl.TenTheLoai AS TheLoai,

s.NamXuatBan AS NamXuatBan,

tn.NgayNhap AS NgayNhap, tn.TongSoLuongNhap AS SoLuong, tn.GiaNhap AS GiaNhap, tn.TongTienNhap AS ThanhTien, CASE

WHEN tn.TrangThai = 'ChuaNhap' THEN N'Chưa nhập' WHEN tn.TrangThai = 'DaNhap' THEN N'Đã nhập' ELSE N'Chưa nhập'

END AS TrangThai

FROM

The\_Nhap tn

INNER JOIN NhanVien nv ON tn.IdNV = nv.IdNV INNER JOIN SACH s ON tn.IdS = s.IdS

INNER JOIN TAC\_GIA tg ON s.IdTacGia = tg.IdTG INNER JOIN THE\_LOAI tl ON s.IdTheLoai = tl.IdTL

INNER JOIN NHA\_XUAT\_BAN nxb ON s.IdNXB = nxb.IdNXB;

* 1. **View Lịch sử nhập kho**

CREATE VIEW ViewLichSuNhapKho AS SELECT

tn.MaTheNhap, nv.MaNV AS MaNV,

s.MaSach,

nv.HoTen AS TenNhanVien, s.TenSach,

tn.NgayNhap, tn.TongSoLuongNhap, tn.GiaNhap, tn.TongTienNhap, CASE

WHEN tn.TrangThai = 'DaNhap' THEN N'Đã Nhập' WHEN tn.TrangThai = 'ChuaNhap' THEN N'Chưa Nhập' ELSE N'Chưa Nhập'

END AS TrangThai FROM The\_Nhap tn

LEFT JOIN SACH s ON tn.IdS = s.IdS

LEFT JOIN NhanVien nv ON tn.IdNV = nv.IdNV;

* 1. **View Chi tiết nhập kho**

CREATE VIEW ViewChiTietNhapKho AS SELECT

nv.MaNV AS MaNhanVien, tn.MaTheNhap AS MaTheNhap, s.MaSach AS MaSach, s.TenSach AS TenSach, tg.MaTacGia AS MaTacGia, tg.TenTacGia AS TenTacGia, tl.MaTheLoai AS MaTheLoai, tl.TenTheLoai AS TheLoai,

nxb.MaNXB AS MaNhaXuatBan, nxb.TenNXB AS TenNhaXuatBan, s.NamXuatBan AS NamXuatBan, ISNULL(ks.SoLuongHienTai, 0) AS SoLuong, tn.NgayNhap AS NgayNhap,

tn.GiaNhap AS GiaNhap, tn.TongTienNhap AS ThanhTien, CASE

WHEN ISNULL(ks.TrangThaiSach, 'HetSach') = 'ConSach' THEN N'Còn Sách' WHEN ISNULL(ks.TrangThaiSach, 'HetSach') = 'HetSach' THEN N'Hết Sách' ELSE N'Hết Sách'

END AS TrangThaiSach

FROM The\_Nhap tn

INNER JOIN NhanVien nv ON tn.IdNV = nv.IdNV INNER JOIN SACH s ON tn.IdS = s.IdS

INNER JOIN TAC\_GIA tg ON s.IdTacGia = tg.IdTG INNER JOIN THE\_LOAI tl ON s.IdTheLoai = tl.IdTL

INNER JOIN NHA\_XUAT\_BAN nxb ON s.IdNXB = nxb.IdNXB

LEFT JOIN Kho\_Sach ks ON s.IdS = ks.MaSach;

1. **Các trigger**
   1. **Trigger cho NhanVien để tự động tăng Mã nhân viên theo đúng định dạng**

CREATE TRIGGER trg\_InsertNV ON NhanVien

AFTER INSERT AS

BEGIN

UPDATE NhanVien

SET MaNV = 'NV' + RIGHT('0000' + CAST(i.IdNV AS varchar(4)), 4)

FROM NhanVien NV

INNER JOIN inserted i ON NV.IdNV = i.IdNV

END;

* 1. **Trigger cho TAC\_GIA để tự động tăng Mã tác giả theo đúng định dạng**

CREATE TRIGGER trg\_InsertTacGia ON TAC\_GIA

AFTER INSERT AS

BEGIN

UPDATE TAC\_GIA

SET MaTacGia = 'TG' + RIGHT('000' + CAST(i.IdTG AS VARCHAR(3)), 3) FROM TAC\_GIA TG

INNER JOIN inserted i ON TG.IdTG = i.IdTG;

END;

* 1. **Trigger cho THE\_LOAI để tự động tăng Mã thể loại theo đúng định dạng**

CREATE TRIGGER trg\_InsertTheLoai ON THE\_LOAI

AFTER INSERT AS

BEGIN

UPDATE THE\_LOAI

SET MaTheLoai = 'TL' + RIGHT('000' + CAST(i.IdTL AS VARCHAR(3)), 3) FROM THE\_LOAI TL

INNER JOIN inserted i ON TL.IdTL = i.IdTL;

END;

* 1. **Trigger cho NHA\_XUAT\_BAN để tự động tăng Mã nhà xuất bản theo đúng định dạng**

CREATE TRIGGER trg\_InsertNXB ON NHA\_XUAT\_BAN

AFTER INSERT AS

BEGIN

UPDATE NHA\_XUAT\_BAN

SET MaNXB = 'NXB' + RIGHT('000' + CAST(i.IdNXB AS VARCHAR(3)), 3) FROM NHA\_XUAT\_BAN NXB

INNER JOIN inserted i ON NXB.IdNXB = i.IdNXB;

END;

* 1. **Trigger cho SACH để tự động tăng Mã sách theo đúng định dạng**

CREATE TRIGGER trg\_InsertSach ON SACH

AFTER INSERT AS

BEGIN

UPDATE SACH

SET MaSach = 'S' + RIGHT('000' + CAST(i.IdS AS VARCHAR(3)), 3) FROM SACH S

INNER JOIN inserted i ON S.IdS = i.IdS;

END;

* 1. **Trigger cho The\_Nhap để tự động tăng Mã thẻ nhập theo đúng định dạng**

CREATE TRIGGER trg\_InsertTheNhap ON The\_Nhap

AFTER INSERT AS

BEGIN

UPDATE The\_Nhap

SET MaTheNhap = 'TN' + RIGHT('000' + CAST(i.IdTN AS VARCHAR(3)), 3)

FROM The\_Nhap TN

INNER JOIN inserted i ON TN.IdTN = i.IdTN;

END;

# CHƯƠNG 3: THIẾT KẾ CÁC CHỨC NĂNG

1. **Kết nối cơ sở dữ liệu**

internal class DBConnect

{

private SqlConnection stringConnect; public DBConnect()

{

string strCon = @"Data Source=localhost;Initial Catalog=NhapSach;Persist Security Info=True;User ID=sa;Password=40938813dD#;Encrypt=True;TrustServerCertificate=True";

stringConnect = new SqlConnection(strCon);

}

public SqlConnection GetConnection

{

get { return stringConnect; }

}

public void OpenConnection()

{

try

{

if (stringConnect.State == ConnectionState.Closed)

{

stringConnect.Open();

}

}

catch (Exception ex)

{

MessageBox.Show("Lỗi khi mở kết nối: " + ex.Message); throw;

}

}

public void CloseConnection()

{

try

{

if (stringConnect.State == ConnectionState.Open)

{

stringConnect.Close();

}

}

catch (Exception ex)

{

MessageBox.Show("Lỗi khi đóng kết nối: " + ex.Message); throw;

}

}

1. **Các chức năng chung của Admin và Nhân viên**
   1. **Nhập sách vào bảng nhập**

CREATE PROCEDURE sp\_NhapSach @MaNV VARCHAR(50), @TenSach NVARCHAR(255), @TenTacGia NVARCHAR(255),

@TenTheLoai NVARCHAR(255), @TenNXB NVARCHAR(255),

@NamXuatBan INT, @GiaNhap DECIMAL(10, 2),

@SoLuong INT, @NgayNhap DATE

AS BEGIN

SET NOCOUNT ON; BEGIN TRY

IF @NamXuatBan > YEAR(GETDATE())

THROW 50005, 'Năm xuất bản không được lớn hơn năm hiện tại!', 1; IF @GiaNhap < 0 OR @SoLuong < 0

THROW 50006, 'Giá nhập hoặc số lượng không được âm!', 1;

IF @TenSach IS NULL OR @TenSach = '' OR @TenTacGia IS NULL OR @TenTacGia = '' OR @TenTheLoai

IS NULL OR @TenTheLoai = '' OR @TenNXB IS NULL OR @TenNXB = ''

THROW 50008, 'Thông tin sách, tác giả, thể loại hoặc nhà xuất bản không được để trống!', 1; BEGIN TRANSACTION;

DECLARE @IdNV INT;

SELECT @IdNV = IdNV FROM NhanVien WHERE MaNV = @MaNV; IF @IdNV IS NULL

BEGIN

THROW 50001, 'Mã nhân viên không tồn tại!', 1;

END;

DECLARE @IdTacGia INT;

IF NOT EXISTS (SELECT 1 FROM TAC\_GIA WHERE TenTacGia = @TenTacGia) INSERT INTO TAC\_GIA (TenTacGia) VALUES (@TenTacGia);

SET @IdTacGia = (SELECT IdTG FROM TAC\_GIA WHERE TenTacGia = @TenTacGia); DECLARE @IdTheLoai INT;

IF NOT EXISTS (SELECT 1 FROM THE\_LOAI WHERE TenTheLoai = @TenTheLoai)

INSERT INTO THE\_LOAI (TenTheLoai) VALUES (@TenTheLoai);

SET @IdTheLoai = (SELECT IdTL FROM THE\_LOAI WHERE TenTheLoai = @TenTheLoai);

DECLARE @IdNXB INT;

IF NOT EXISTS (SELECT 1 FROM NHA\_XUAT\_BAN WHERE TenNXB = @TenNXB) INSERT INTO NHA\_XUAT\_BAN (TenNXB) VALUES (@TenNXB);

SET @IdNXB = (SELECT IdNXB FROM NHA\_XUAT\_BAN WHERE TenNXB = @TenNXB);

DECLARE @IdS INT;

DECLARE @MaSach VARCHAR(50);

IF NOT EXISTS (SELECT 1 FROM SACH WHERE TenSach = @TenSach AND IdTacGia = @IdTacGia AND IdTheLoai = @IdTheLoai AND IdNXB = @IdNXB AND NamXuatBan = @NamXuatBan)

BEGIN

END ELSE BEGIN

INSERT INTO SACH (TenSach, NamXuatBan, GiaSach, IdTacGia, IdTheLoai, IdNXB, AnhBia) VALUES (@TenSach, @NamXuatBan, @GiaNhap, @IdTacGia, @IdTheLoai, @IdNXB, NULL); SET @IdS = SCOPE\_IDENTITY();

SET @MaSach = (SELECT MaSach FROM SACH WHERE IdS = @IdS);

SET @IdS = (SELECT TOP 1 IdS FROM SACH WHERE TenSach = @TenSach AND IdTacGia =

@IdTacGia AND IdTheLoai = @IdTheLoai AND IdNXB = @IdNXB AND NamXuatBan = @NamXuatBan); SET @MaSach = (SELECT MaSach FROM SACH WHERE IdS = @IdS);

END;

IF @IdS IS NULL BEGIN

THROW 50004, 'Không thể lấy hoặc tạo IdS cho sách!', 1;

END;

DECLARE @TongTienNhap DECIMAL(10, 2) = @GiaNhap \* @SoLuong; DECLARE @IdTN INT;

INSERT INTO The\_Nhap (IdNV, IdS, NgayNhap, TrangThai, GiaNhap, TongSoLuongNhap, TongTienNhap) VALUES (@IdNV, @IdS, @NgayNhap, 'ChuaNhap', @GiaNhap, @SoLuong, @TongTienNhap);

SET @IdTN = SCOPE\_IDENTITY(); DECLARE @MaTheNhap VARCHAR(50);

SET @MaTheNhap = (SELECT MaTheNhap FROM The\_Nhap WHERE IdTN = @IdTN); IF @MaTheNhap IS NULL

BEGIN

THROW 50009, 'Không thể tạo MaTheNhap!', 1;

END;

SELECT @MaSach AS MaSach, @MaTheNhap AS MaTheNhap;

END;

COMMIT TRANSACTION; END TRY

BEGIN CATCH

IF @@TRANCOUNT > 0

ROLLBACK TRANSACTION;

DECLARE @ErrorMessage NVARCHAR(4000) = ERROR\_MESSAGE();

DECLARE @ErrorSeverity INT = ERROR\_SEVERITY(); DECLARE @ErrorState INT = ERROR\_STATE();

RAISERROR (@ErrorMessage, @ErrorSeverity, @ErrorState); END CATCH;

**Xử lý bên C#:**

public bool InsertSach(string maNV, string tenSach, string tenTacGia, string tenTheLoai, string tenNXB, int namXuatBan, decimal giaNhap, int soLuong, DateTime ngayNhap, byte[] anhBia)

{

SqlParameter[] paramsArr = new SqlParameter[]

{

new SqlParameter("@MaNV", SqlDbType.VarChar) { Value = maNV },

new SqlParameter("@TenSach", SqlDbType.NVarChar) { Value = tenSach }, new SqlParameter("@TenTacGia", SqlDbType.NVarChar) { Value = tenTacGia }, new SqlParameter("@TenTheLoai", SqlDbType.NVarChar) { Value = tenTheLoai }, new SqlParameter("@TenNXB", SqlDbType.NVarChar) { Value = tenNXB },

new SqlParameter("@NamXuatBan", SqlDbType.Int) { Value = namXuatBan }, new SqlParameter("@GiaNhap", SqlDbType.Decimal) { Value = giaNhap }, new SqlParameter("@SoLuong", SqlDbType.Int) { Value = soLuong },

new SqlParameter("@NgayNhap", SqlDbType.Date) { Value = ngayNhap }

};

try

{

DataTable result = db.ExecuteQuery(

"EXEC sp\_NhapSach @MaNV, @TenSach, @TenTacGia, @TenTheLoai, @TenNXB, @NamXuatBan, @GiaNhap, @SoLuong, @NgayNhap",

paramsArr);

string maSach = result.Rows[0]["MaSach"].ToString();

string maTheNhap = result.Rows[0]["MaTheNhap"].ToString();

if (anhBia != null)

{

SqlParameter[] imageParams = new SqlParameter[]

{

new SqlParameter("@MaSach", SqlDbType.VarChar) { Value = maSach }, new SqlParameter("@AnhBia", SqlDbType.VarBinary) { Value = anhBia }

};

db.ExecuteNonQuery("EXEC sp\_CapNhatAnhBia @MaSach, @AnhBia", imageParams); tempImages[maTheNhap] = anhBia;

}

MessageBox.Show("Nhập sách thành công!", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information); return true;

}

catch (Exception ex)

{

MessageBox.Show("Lỗi: " + ex.Message, "Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Error); return false;

}

### **2.2 Cập nhật thông tin sách nhập**

CREATE PROCEDURE sp\_CapNhatSach @MaTheNhap VARCHAR(50), @TenSach NVARCHAR(255), @TenTacGia NVARCHAR(255),

@TenTheLoai NVARCHAR(255), @TenNXB NVARCHAR(255),

@NamXuatBan INT, @GiaNhap DECIMAL(10, 2),

@SoLuong INT, @NgayNhap DATE

AS BEGIN

SET NOCOUNT ON; BEGIN TRY

IF @NamXuatBan > YEAR(GETDATE())

THROW 50005, 'Năm xuất bản không được lớn hơn năm hiện tại!', 1; IF @GiaNhap < 0 OR @SoLuong < 0

THROW 50006, 'Giá nhập hoặc số lượng không được âm!', 1; BEGIN TRANSACTION;

IF NOT EXISTS (SELECT 1 FROM The\_Nhap WHERE MaTheNhap = @MaTheNhap) BEGIN

THROW 50002, 'Mã thẻ nhập không tồn tại!', 1;

END;

DECLARE @TrangThai VARCHAR(50);

SELECT @TrangThai = TrangThai FROM The\_Nhap

WHERE MaTheNhap = @MaTheNhap; IF @TrangThai = 'DaNhap'

BEGIN

THROW 50007, 'Thẻ nhập đã được xác nhận, không thể chỉnh sửa thông tin!', 1;

END;

DECLARE @IdS INT; SELECT @IdS = IdS

FROM The\_Nhap

WHERE MaTheNhap = @MaTheNhap; DECLARE @IdTacGia INT;

IF NOT EXISTS (SELECT 1 FROM TAC\_GIA WHERE TenTacGia = @TenTacGia) INSERT INTO TAC\_GIA (TenTacGia) VALUES (@TenTacGia);

SET @IdTacGia = (SELECT IdTG FROM TAC\_GIA WHERE TenTacGia = @TenTacGia); DECLARE @IdTheLoai INT;

IF NOT EXISTS (SELECT 1 FROM THE\_LOAI WHERE TenTheLoai = @TenTheLoai)

INSERT INTO THE\_LOAI (TenTheLoai) VALUES (@TenTheLoai);

SET @IdTheLoai = (SELECT IdTL FROM THE\_LOAI WHERE TenTheLoai = @TenTheLoai); DECLARE @IdNXB INT;

IF NOT EXISTS (SELECT 1 FROM NHA\_XUAT\_BAN WHERE TenNXB = @TenNXB) INSERT INTO NHA\_XUAT\_BAN (TenNXB) VALUES (@TenNXB);

SET @IdNXB = (SELECT IdNXB FROM NHA\_XUAT\_BAN WHERE TenNXB = @TenNXB); UPDATE SACH

SET TenSach = @TenSach, NamXuatBan = @NamXuatBan, GiaSach = @GiaNhap, IdTacGia = @IdTacGia, IdTheLoai = @IdTheLoai, IdNXB = @IdNXB

WHERE IdS = @IdS;

DECLARE @TongTienNhap DECIMAL(10, 2) = @GiaNhap \* @SoLuong; UPDATE The\_Nhap

SET NgayNhap = @NgayNhap, GiaNhap = @GiaNhap, TongSoLuongNhap = @SoLuong, TongTienNhap = @TongTienNhap

WHERE MaTheNhap = @MaTheNhap; COMMIT TRANSACTION;

END TRY BEGIN CATCH

IF @@TRANCOUNT > 0

ROLLBACK TRANSACTION;

DECLARE @ErrorMessage NVARCHAR(4000) = ERROR\_MESSAGE();

DECLARE @ErrorSeverity INT = ERROR\_SEVERITY(); DECLARE @ErrorState INT = ERROR\_STATE();

RAISERROR (@ErrorMessage, @ErrorSeverity, @ErrorState); END CATCH;

END;

**Xử lý bên C#:**

public bool UpdateSach(string maTheNhap, string tenSach, string tenTacGia, string tenTheLoai, string tenNXB, int namXuatBan, decimal giaNhap, int soLuong, DateTime ngayNhap)

{

SqlParameter[] paramsArr = new SqlParameter[]

{

new SqlParameter("@MaTheNhap", SqlDbType.VarChar) { Value = maTheNhap }, new SqlParameter("@TenSach", SqlDbType.NVarChar) { Value = tenSach },

new SqlParameter("@TenTacGia", SqlDbType.NVarChar) { Value = tenTacGia }, new SqlParameter("@TenTheLoai", SqlDbType.NVarChar) { Value = tenTheLoai }, new SqlParameter("@TenNXB", SqlDbType.NVarChar) { Value = tenNXB },

new SqlParameter("@NamXuatBan", SqlDbType.Int) { Value = namXuatBan }, new SqlParameter("@GiaNhap", SqlDbType.Decimal) { Value = giaNhap }, new SqlParameter("@SoLuong", SqlDbType.Int) { Value = soLuong },

new SqlParameter("@NgayNhap", SqlDbType.Date) { Value = ngayNhap }

};

try

{

db.ExecuteNonQuery("sp\_CapNhatSach", paramsArr, CommandType.StoredProcedure);

MessageBox.Show("Cập nhật thông tin thành công!", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);

return true;

}

catch (Exception ex)

{

MessageBox.Show("Lỗi: " + ex.Message, "Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Error); return false;

}

}

* 1. **Xóa sách khỏi bảng nhập**

CREATE PROCEDURE sp\_XoaSach @MaTheNhap VARCHAR(50), @MaSach VARCHAR(50)

AS BEGIN

SET NOCOUNT ON; BEGIN TRY

IF NOT EXISTS (SELECT 1 FROM The\_Nhap WHERE MaTheNhap = @MaTheNhap) BEGIN

THROW 50002, 'Mã thẻ nhập không tồn tại!', 1;

END;

DECLARE @TrangThai VARCHAR(50);

SELECT @TrangThai = TrangThai FROM The\_Nhap

WHERE MaTheNhap = @MaTheNhap; IF @TrangThai = 'DaNhap'

BEGIN

THROW 50007, 'Thẻ nhập đã được xác nhận, không thể xóa!', 1;

END;

IF NOT EXISTS (SELECT 1 FROM SACH WHERE MaSach = @MaSach) BEGIN

THROW 50010, 'Mã sách không tồn tại!', 1;

END;

DECLARE @IdS INT; SELECT @IdS = IdS

FROM The\_Nhap

WHERE MaTheNhap = @MaTheNhap;

DECLARE @IdTacGia INT, @IdTheLoai INT, @IdNXB INT;

SELECT @IdTacGia = IdTacGia, @IdTheLoai = IdTheLoai, @IdNXB = IdNXB FROM SACH

WHERE MaSach = @MaSach; BEGIN TRANSACTION;

DELETE FROM The\_Nhap WHERE MaTheNhap = @MaTheNhap; DELETE FROM SACH WHERE MaSach = @MaSach;

IF NOT EXISTS (SELECT 1 FROM SACH WHERE IdTacGia = @IdTacGia) BEGIN

DELETE FROM TAC\_GIA WHERE IdTG = @IdTacGia;

END;

IF NOT EXISTS (SELECT 1 FROM SACH WHERE IdTheLoai = @IdTheLoai) BEGIN

DELETE FROM THE\_LOAI WHERE IdTL = @IdTheLoai;

END;

IF NOT EXISTS (SELECT 1 FROM SACH WHERE IdNXB = @IdNXB) BEGIN

DELETE FROM NHA\_XUAT\_BAN WHERE IdNXB = @IdNXB;

END;

COMMIT TRANSACTION; END TRY

BEGIN CATCH

IF @@TRANCOUNT > 0

ROLLBACK TRANSACTION;

DECLARE @ErrorMessage NVARCHAR(4000) = ERROR\_MESSAGE();

DECLARE @ErrorSeverity INT = ERROR\_SEVERITY(); DECLARE @ErrorState INT = ERROR\_STATE();

RAISERROR (@ErrorMessage, @ErrorSeverity, @ErrorState); END CATCH;

END;

**Xử lý bên C#:**

,

public bool DeleteSach(string maTheNhap, string maSach)

{

SqlParameter[] paramsArr = new SqlParameter[]

{

new SqlParameter("@MaTheNhap", SqlDbType.VarChar) { Value = maTheNhap }, new SqlParameter("@MaSach", SqlDbType.VarChar) { Value = maSach }

};

try

{

db.ExecuteNonQuery("sp\_XoaSach", paramsArr, CommandType.StoredProcedure);

tempImages.Remove(maTheNhap);

MessageBox.Show("Xóa sách

MessageBoxIcon.Information);

return true;

}

catch (Exception ex)

{

thành

công!", "Thông báo", MessageBoxButtons.OK

MessageBox.Show("Lỗi: " + ex.Message, "Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Error);

return false;

}

}

* 1. **Thêm ảnh bìa vào sách**

CREATE PROCEDURE sp\_CapNhatAnhBia @MaSach VARCHAR(50), @AnhBia VARBINARY(MAX)

AS BEGIN

SET NOCOUNT ON; BEGIN TRY

BEGIN TRANSACTION;

IF NOT EXISTS (SELECT 1 FROM SACH WHERE MaSach = @MaSach)

THROW 50010, 'Mã sách không tồn tại!', 1; UPDATE SACH

SET AnhBia = @AnhBia WHERE MaSach = @MaSach; COMMIT TRANSACTION;

END TRY

BEGIN CATCH

IF @@TRANCOUNT > 0

ROLLBACK TRANSACTION;

DECLARE @ErrorMessage NVARCHAR(4000) = ERROR\_MESSAGE();

DECLARE @ErrorSeverity INT = ERROR\_SEVERITY(); DECLARE @ErrorState INT = ERROR\_STATE();

RAISERROR (@ErrorMessage, @ErrorSeverity, @ErrorState); END CATCH;

END;

**Xử lý bên C#:**

public bool UpdateAnhBia(string maSach, byte[] anhBia)

{

SqlParameter[] imageParams = new SqlParameter[]

{

new SqlParameter("@MaSach", SqlDbType.VarChar) { Value = maSach },

new SqlParameter("@AnhBia", SqlDbType.VarBinary) { Value = anhBia }

};

try

{

db.ExecuteNonQuery("sp\_CapNhatAnhBia", imageParams, CommandType.StoredProcedure);

MessageBox.Show("Tải MessageBoxIcon.Information);

return true;

}

catch (Exception ex)

{

ảnh

lên thành công!", "Thông báo", MessageBoxButtons.OK

MessageBox.Show("Lỗi khi cập nhật ảnh:

MessageBoxIcon.Error); return false;

}

}

" + ex.Message, "Lỗi",

MessageBoxButtons.OK,

,

* 1. **Cập nhật thêm bớt số lượng sách vào kho sau khi đã nhập sách**

CREATE PROCEDURE sp\_CapNhatKhoSach @IdS INT,

@SoLuongThem INT,

@ResultMessage NVARCHAR(4000) OUTPUT

AS BEGIN

SET NOCOUNT ON; BEGIN TRY

DECLARE @SoLuongHienTai INT; DECLARE @TongSoLuongNhap INT; DECLARE @HasNhapKho BIT; DECLARE @SoLuongMoi INT; BEGIN TRANSACTION;

IF NOT EXISTS (SELECT 1 FROM SACH WHERE IdS = @IdS) BEGIN

END;

SET @ResultMessage = N'Không có thông tin trong kho'; ROLLBACK TRANSACTION;

RETURN 1;

'DaNhap');

SET @HasNhapKho = (SELECT COUNT(\*) FROM The\_Nhap WHERE IdS = @IdS AND TrangThai =

IF @HasNhapKho = 0 BEGIN

END;

SET @ResultMessage = N'Không có thông tin trong kho'; ROLLBACK TRANSACTION;

RETURN 1;

SELECT @SoLuongHienTai = ISNULL(SoLuongHienTai, 0) FROM Kho\_Sach

WHERE MaSach = @IdS;

SET @SoLuongMoi = @SoLuongHienTai + @SoLuongThem; IF @SoLuongMoi < 0

BEGIN

SET @ResultMessage = N'Số lượng trong kho hiện tại là ' + CAST(@SoLuongHienTai AS NVARCHAR(10))

+ N' không đủ sách để giảm ' + CAST(ABS(@SoLuongThem) AS NVARCHAR(10)) + N'\nCập nhật thất bại'; ROLLBACK TRANSACTION;

RETURN 1;

END;

SELECT @TongSoLuongNhap = ISNULL(SUM(TongSoLuongNhap), 0) FROM The\_Nhap

WHERE IdS = @IdS AND TrangThai = 'DaNhap'; IF @SoLuongMoi > @TongSoLuongNhap

BEGIN

SET @ResultMessage = N'Số lượng thêm vượt quá số lượng đã nhập là ' + CAST(@TongSoLuongNhap AS

NVARCHAR(10)) + N'\nCập nhật thất bại';

ROLLBACK TRANSACTION; RETURN 1;

END;

IF EXISTS (SELECT 1 FROM Kho\_Sach WHERE MaSach = @IdS) BEGIN

END

UPDATE Kho\_Sach

SET SoLuongHienTai = @SoLuongMoi,

TrangThaiSach = CASE WHEN @SoLuongMoi > 0 THEN N'ConSach' ELSE N'HetSach' END WHERE MaSach = @IdS;

ELSE IF @SoLuongThem > 0 BEGIN

END;

INSERT INTO Kho\_Sach (MaSach, SoLuongHienTai, TrangThaiSach) VALUES (@IdS, @SoLuongThem, N'ConSach');

SET @ResultMessage = N'Cập nhật thành công'; COMMIT TRANSACTION;

RETURN 0; END TRY

BEGIN CATCH

IF @@TRANCOUNT > 0

ROLLBACK TRANSACTION;

DECLARE @ErrorMessage NVARCHAR(4000) = ERROR\_MESSAGE();

DECLARE @ErrorSeverity INT = ERROR\_SEVERITY(); DECLARE @ErrorState INT = ERROR\_STATE();

RAISERROR (@ErrorMessage, @ErrorSeverity, @ErrorState); SET @ResultMessage = @ErrorMessage;

RETURN 1;

END CATCH; END;

**Xử lý bên C#:**

public bool UpdateKhoSach(int idS, int soLuongThem, out string resultMessage)

{

resultMessage = "";

SqlParameter[] parameters = new SqlParameter[]

{

new SqlParameter("@IdS", SqlDbType.Int) { Value = idS },

new SqlParameter("@SoLuongThem", SqlDbType.Int) { Value = soLuongThem },

new SqlParameter("@ResultMessage", SqlDbType.NVarChar, 4000) { Direction = ParameterDirection.Output }

};

try

{

dbConnect.ExecuteNonQuery("sp\_CapNhatKhoSach", parameters, CommandType.StoredProcedure); resultMessage = parameters[2].Value?.ToString() ?? "Cập nhật thất bại";

return true;

}

catch (Exception)

{

resultMessage = "Cập nhật thất bại"; return false;

}

}

* 1. **Kiểm tra trạng thái sách trong kho sau khi đã nhập**

CREATE FUNCTION fn\_KiemTraTrangThaiKhoSach (@p\_IdS INT) RETURNS NVARCHAR(300)

AS BEGIN

DECLARE @KetQua NVARCHAR(300); DECLARE @MaSach NVARCHAR(50); DECLARE @TenSach NVARCHAR(100);

DECLARE @SoLuong INT;

DECLARE @TrangThai NVARCHAR(50); DECLARE @MaNV NVARCHAR(50); DECLARE @HoTen NVARCHAR(50); DECLARE @MaTheNhap NVARCHAR(50);

SELECT @MaSach = MaSach, @TenSach = TenSach FROM SACH

WHERE IdS = @p\_IdS; IF @MaSach IS NULL BEGIN

SET @KetQua = N'Mã sách ' + CAST(@p\_IdS AS NVARCHAR(10)) + N' không tồn tại trong danh mục sách'; RETURN @KetQua;

END;

SELECT TOP 1

@MaNV = nv.MaNV, @HoTen = nv.HoTen,

@MaTheNhap = tn.MaTheNhap,

@SoLuong = ISNULL(ks.SoLuongHienTai, 0), @TrangThai = CASE

WHEN ISNULL(ks.SoLuongHienTai, 0) > 0 THEN N'Còn Sách' ELSE N'Hết Sách'

END

FROM The\_Nhap tn

LEFT JOIN NhanVien nv ON tn.IdNV = nv.IdNV LEFT JOIN Kho\_Sach ks ON tn.IdS = ks.MaSach

WHERE tn.IdS = @p\_IdS AND tn.TrangThai = 'DaNhap' ORDER BY tn.NgayNhap DESC;

IF @MaNV IS NULL BEGIN

SET @KetQua = N'Mã sách ' + @MaSach + N' chưa có dữ liệu nhập kho'; RETURN @KetQua;

END;

END;

SET @KetQua = N'Mã nhân viên: ' + @MaNV + NCHAR(13) + NCHAR(10) + N'Tên nhân viên: ' + @HoTen + NCHAR(13) + NCHAR(10) + N'Mã thẻ nhập: ' + @MaTheNhap + NCHAR(13) + NCHAR(10) + N'Mã sách: ' + @MaSach + NCHAR(13) + NCHAR(10) +

N'Tên sách: ' + @TenSach + NCHAR(13) + NCHAR(10) +

N'Số lượng: ' + CAST(@SoLuong AS NVARCHAR(10)) + NCHAR(13) + NCHAR(10) +

N'Trạng thái: ' + @TrangThai; RETURN @KetQua;

**Xử lý bên C#:**

public bool CheckKhoSach(int idS, out string kiemTraKho)

{

kiemTraKho = "";

SqlParameter[] parameters = new SqlParameter[]

{

new SqlParameter("@p\_IdS", SqlDbType.Int) { Value = idS }

};

try

{

DataTable dt = dbConnect.ExecuteQuery("SELECT dbo.fn\_KiemTraTrangThaiKhoSach(@p\_IdS) AS KiemTraKho", parameters);

if (dt.Rows.Count > 0)

{

kiemTraKho = dt.Rows[0]["KiemTraKho"].ToString(); return true;

}

else

{

kiemTraKho = $"Mã sách S{idS:D3} không có dữ liệu trạng thái kho!"; return false;

}

}

catch (Exception ex)

{

kiemTraKho = $"Đã xảy ra lỗi: {ex.Message}"; return false;

}

}

* 1. **Thống kê sách đã nhập**

CREATE PROCEDURE sp\_ThongKeSachDaNhap @TuNgay DATE = NULL,

@DenNgay DATE = NULL

AS BEGIN

SET NOCOUNT ON; SELECT

s.MaSach, s.TenSach,

SUM(tn.TongSoLuongNhap) AS TongSoLuongNhap FROM The\_Nhap tn

INNER JOIN SACH s ON tn.IdS = s.IdS

WHERE tn.TrangThai = 'DaNhap'

AND (@TuNgay IS NULL OR tn.NgayNhap >= @TuNgay) AND (@DenNgay IS NULL OR tn.NgayNhap <= @DenNgay)

GROUP BY s.MaSach, s.TenSach ORDER BY s.MaSach;

END;

**Xử lý bên C#:**

public bool GetThongKeSachDaNhap(DateTime tuNgay, DateTime denNgay, out DataTable dt)

{

dt = new DataTable(); SqlParameter[] parameters = new[]

{

new SqlParameter("@TuNgay", SqlDbType.Date) { Value = tuNgay }, new SqlParameter("@DenNgay", SqlDbType.Date) { Value = denNgay }

};

try

{

dt = dbConnect.ExecuteQuery("EXEC sp\_ThongKeSachDaNhap @TuNgay, @DenNgay", parameters); return true;

}

catch (Exception ex)

{

MessageBox.Show($"Lỗi khi tải dữ liệu thống kê: {ex.Message}", "Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Error);

return false;

}

}

* 1. **Thống kê ngày đã nhập**

CREATE PROCEDURE sp\_ThongKeNhapSachTheoNgay @TuNgay DATE = NULL,

@DenNgay DATE = NULL

AS BEGIN

SET NOCOUNT ON; SELECT

tn.NgayNhap AS NgayNhap, SUM(tn.TongSoLuongNhap) AS TongSoLuongNhap

FROM The\_Nhap tn

WHERE tn.TrangThai = 'DaNhap'

AND (@TuNgay IS NULL OR tn.NgayNhap >= @TuNgay) AND (@DenNgay IS NULL OR tn.NgayNhap <= @DenNgay)

GROUP BY tn.NgayNhap ORDER BY tn.NgayNhap;

END;

**Xử lý bên C#:**

public bool GetThongKeNhapSachTheoNgay(DateTime tuNgay, DateTime denNgay, out DataTable dt)

{

dt = new DataTable(); SqlParameter[] parameters = new[]

{

new SqlParameter("@TuNgay", SqlDbType.Date) { Value = tuNgay }, new SqlParameter("@DenNgay", SqlDbType.Date) { Value = denNgay }

};

try

{

dt = dbConnect.ExecuteQuery("EXEC sp\_ThongKeNhapSachTheoNgay @TuNgay, @DenNgay", parameters);

return true;

}

catch (Exception ex)

{

MessageBox.Show($"Lỗi khi tải dữ liệu thống kê: {ex.Message}", "Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Error);

return false;

}

}

1. **Chức năng riêng của Admin**

**Duyệt xác nhận nhập vào kho sau khi đã nhập sách vào bảng nhập**

CREATE PROCEDURE sp\_XacNhanNhap @MaTheNhap VARCHAR(50)

AS BEGIN

SET NOCOUNT ON; BEGIN TRY

BEGIN TRANSACTION;

IF NOT EXISTS (SELECT 1 FROM The\_Nhap WHERE MaTheNhap = @MaTheNhap AND TrangThai = 'ChuaNhap')

BEGIN

THROW 50003, 'Mã thẻ nhập không tồn tại hoặc đã được xác nhận!', 1;

END;

DECLARE @IdS INT, @SoLuong INT;

SELECT @IdS = IdS, @SoLuong = TongSoLuongNhap FROM The\_Nhap

WHERE MaTheNhap = @MaTheNhap; UPDATE The\_Nhap

SET TrangThai = 'DaNhap'

WHERE MaTheNhap = @MaTheNhap; DECLARE @ResultMessage NVARCHAR(4000);

EXEC sp\_CapNhatKhoSach @IdS, @SoLuong, @ResultMessage OUTPUT; IF @ResultMessage != N'Cập nhật thành công'

BEGIN

THROW 50004, @ResultMessage, 1;

END;

COMMIT TRANSACTION; END TRY

BEGIN CATCH

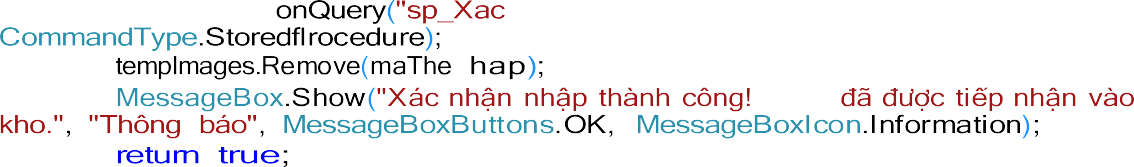
IF @@TRANCOUNT > 0

ROLLBACK TRANSACTION; THROW;

END CATCH;

END;

**Xử lý bên C#:**



q a a e e a he

# CHƯƠNG 4: PHÂN QUYỀN

Hệ thống sẽ bao gồm 2 Role được định nghĩa trong bảng TaiKhoan, NhanVien bao

gồm: Admin và Nhân viên.

Tương ứng sẽ tạo 2 Role trong SQL Server và thực hiện phân quyền cho các Role

này: AdminRole, NhanVienRole.

**Tạo các role**

CREATE ROLE AdminRole; CREATE ROLE NhanVienRole;

ALTER ROLE NhanVienRole ADD MEMBER AdminRole; GO

**Gán quyền cho các role**

-- Gán quyền cho NhanVienRole (quyền cơ bản trên từng bảng)

GRANT SELECT, INSERT, UPDATE, DELETE ON TaiKhoan TO NhanVienRole; GRANT SELECT, INSERT, UPDATE, DELETE ON NhanVien TO NhanVienRole; GRANT SELECT, INSERT, UPDATE, DELETE ON TAC\_GIA TO NhanVienRole; GRANT SELECT, INSERT, UPDATE, DELETE ON THE\_LOAI TO NhanVienRole; GRANT SELECT, INSERT, UPDATE, DELETE ON NHA\_XUAT\_BAN TO NhanVienRole; GRANT SELECT, INSERT, UPDATE, DELETE ON SACH TO NhanVienRole;

GRANT SELECT, INSERT, UPDATE, DELETE ON The\_Nhap TO NhanVienRole; GRANT SELECT, INSERT, UPDATE, DELETE ON Kho\_Sach TO NhanVienRole;

GRANT SELECT ON [Admin] TO NhanVienRole;

-- Gán quyền thực thi stored procedure cho NhanVienRole GRANT EXECUTE ON sp\_NhapSach TO NhanVienRole; GRANT EXECUTE ON sp\_CapNhatSach TO NhanVienRole; GRANT EXECUTE ON sp\_CapNhatKhoSach TO NhanVienRole; GRANT EXECUTE ON sp\_CapNhatAnhBia TO NhanVienRole;

GRANT EXECUTE ON sp\_ThongKeSachDaNhap TO NhanVienRole; GRANT EXECUTE ON sp\_ThongKeNhapSachTheoNgay TO NhanVienRole;

-- Từ chối quyền thực thi sp\_XacNhanNhap cho NhanVienRole

DENY EXECUTE ON sp\_XacNhanNhap TO NhanVienRole;

-- Gán quyền thực thi hàm cho NhanVienRole

GRANT EXECUTE ON fn\_KiemTraTrangThaiKhoSach TO NhanVienRole;

-- Gán quyền xem view cho NhanVienRole

GRANT SELECT ON ViewDanhSachSach TO NhanVienRole; GRANT SELECT ON ViewDanhSachTheLoai TO NhanVienRole; GRANT SELECT ON ViewDanhSachTacGia TO NhanVienRole; GRANT SELECT ON ViewDanhSachNhaXuatBan TO NhanVienRole; GRANT SELECT ON ViewLichSuNhapKho TO NhanVienRole; GRANT SELECT ON ViewChiTietNhapKho TO NhanVienRole; GRANT SELECT ON ViewChiTietTheNhap TO NhanVienRole;

GO

-- Gán quyền bổ sung cho AdminRole (chỉ quyền không kế thừa từ NhanVienRole)

GRANT SELECT, INSERT, UPDATE, DELETE ON [Admin] TO AdminRole;

GRANT EXECUTE ON sp\_XacNhanNhap TO AdminRole; GO

# CHƯƠNG 5: TRANSACTION

Các thủ tục (stored procedure) thực hiện thêm, sửa, xóa, cập nhật ảnh bìa, và các thao tác liên quan đến thẻ nhập đều sử dụng BEGIN TRANSACTION và COMMIT TRANSACTION để đảm bảo tính toàn vẹn dữ liệu. Trong trường hợp xảy ra lỗi, khối CATCH sẽ thực hiện ROLLBACK TRANSACTION để hủy bỏ các thay đổi, ngăn chặn dữ liệu bị sai lệch.

**Transaction nhập sách vào bảng nhập**

CREATE PROCEDURE sp\_NhapSach @MaNV VARCHAR(50), @TenSach NVARCHAR(255), @TenTacGia NVARCHAR(255),

@TenTheLoai NVARCHAR(255), @TenNXB NVARCHAR(255),

@NamXuatBan INT, @GiaNhap DECIMAL(10, 2),

@SoLuong INT, @NgayNhap DATE

AS BEGIN

SET NOCOUNT ON; BEGIN TRY

IF @NamXuatBan > YEAR(GETDATE())

THROW 50005, 'Năm xuất bản không được lớn hơn năm hiện tại!', 1; IF @GiaNhap < 0 OR @SoLuong < 0

THROW 50006, 'Giá nhập hoặc số lượng không được âm!', 1;

IF @TenSach IS NULL OR @TenSach = '' OR @TenTacGia IS NULL OR @TenTacGia = '' OR @TenTheLoai

IS NULL OR @TenTheLoai = '' OR @TenNXB IS NULL OR @TenNXB = ''

THROW 50008, 'Thông tin sách, tác giả, thể loại hoặc nhà xuất bản không được để trống!', 1; BEGIN TRANSACTION;

DECLARE @IdNV INT;

SELECT @IdNV = IdNV FROM NhanVien WHERE MaNV = @MaNV; IF @IdNV IS NULL

BEGIN

THROW 50001, 'Mã nhân viên không tồn tại!', 1;

END;

DECLARE @IdTacGia INT;

IF NOT EXISTS (SELECT 1 FROM TAC\_GIA WHERE TenTacGia = @TenTacGia) INSERT INTO TAC\_GIA (TenTacGia) VALUES (@TenTacGia);

SET @IdTacGia = (SELECT IdTG FROM TAC\_GIA WHERE TenTacGia = @TenTacGia); DECLARE @IdTheLoai INT;

IF NOT EXISTS (SELECT 1 FROM THE\_LOAI WHERE TenTheLoai = @TenTheLoai)

INSERT INTO THE\_LOAI (TenTheLoai) VALUES (@TenTheLoai);

SET @IdTheLoai = (SELECT IdTL FROM THE\_LOAI WHERE TenTheLoai = @TenTheLoai);

DECLARE @IdNXB INT;

IF NOT EXISTS (SELECT 1 FROM NHA\_XUAT\_BAN WHERE TenNXB = @TenNXB) INSERT INTO NHA\_XUAT\_BAN (TenNXB) VALUES (@TenNXB);

SET @IdNXB = (SELECT IdNXB FROM NHA\_XUAT\_BAN WHERE TenNXB = @TenNXB);

DECLARE @IdS INT;

DECLARE @MaSach VARCHAR(50);

IF NOT EXISTS (SELECT 1 FROM SACH WHERE TenSach = @TenSach AND IdTacGia = @IdTacGia AND IdTheLoai = @IdTheLoai AND IdNXB = @IdNXB AND NamXuatBan = @NamXuatBan)

BEGIN

END ELSE BEGIN

INSERT INTO SACH (TenSach, NamXuatBan, GiaSach, IdTacGia, IdTheLoai, IdNXB, AnhBia) VALUES (@TenSach, @NamXuatBan, @GiaNhap, @IdTacGia, @IdTheLoai, @IdNXB, NULL); SET @IdS = SCOPE\_IDENTITY();

SET @MaSach = (SELECT MaSach FROM SACH WHERE IdS = @IdS);

SET @IdS = (SELECT TOP 1 IdS FROM SACH WHERE TenSach = @TenSach AND IdTacGia =

@IdTacGia AND IdTheLoai = @IdTheLoai AND IdNXB = @IdNXB AND NamXuatBan = @NamXuatBan); SET @MaSach = (SELECT MaSach FROM SACH WHERE IdS = @IdS);

END;

IF @IdS IS NULL BEGIN

THROW 50004, 'Không thể lấy hoặc tạo IdS cho sách!', 1;

END;

DECLARE @TongTienNhap DECIMAL(10, 2) = @GiaNhap \* @SoLuong; DECLARE @IdTN INT;

INSERT INTO The\_Nhap (IdNV, IdS, NgayNhap, TrangThai, GiaNhap, TongSoLuongNhap, TongTienNhap) VALUES (@IdNV, @IdS, @NgayNhap, 'ChuaNhap', @GiaNhap, @SoLuong, @TongTienNhap);

SET @IdTN = SCOPE\_IDENTITY(); DECLARE @MaTheNhap VARCHAR(50);

SET @MaTheNhap = (SELECT MaTheNhap FROM The\_Nhap WHERE IdTN = @IdTN); IF @MaTheNhap IS NULL

BEGIN

THROW 50009, 'Không thể tạo MaTheNhap!', 1;

END;

SELECT @MaSach AS MaSach, @MaTheNhap AS MaTheNhap;

END;

COMMIT TRANSACTION; END TRY

BEGIN CATCH

IF @@TRANCOUNT > 0

ROLLBACK TRANSACTION;

DECLARE @ErrorMessage NVARCHAR(4000) = ERROR\_MESSAGE();

DECLARE @ErrorSeverity INT = ERROR\_SEVERITY(); DECLARE @ErrorState INT = ERROR\_STATE();

RAISERROR (@ErrorMessage, @ErrorSeverity, @ErrorState); END CATCH;

**Transaction cập nhật thông tin sách nhập**

CREATE PROCEDURE sp\_CapNhatSach @MaTheNhap VARCHAR(50), @TenSach NVARCHAR(255), @TenTacGia NVARCHAR(255),

@TenTheLoai NVARCHAR(255), @TenNXB NVARCHAR(255),

@NamXuatBan INT, @GiaNhap DECIMAL(10, 2),

@SoLuong INT, @NgayNhap DATE

AS

BEGIN

SET NOCOUNT ON; BEGIN TRY

IF @NamXuatBan > YEAR(GETDATE())

THROW 50005, 'Năm xuất bản không được lớn hơn năm hiện tại!', 1; IF @GiaNhap < 0 OR @SoLuong < 0

THROW 50006, 'Giá nhập hoặc số lượng không được âm!', 1; BEGIN TRANSACTION;

IF NOT EXISTS (SELECT 1 FROM The\_Nhap WHERE MaTheNhap = @MaTheNhap) BEGIN

THROW 50002, 'Mã thẻ nhập không tồn tại!', 1;

END;

DECLARE @TrangThai VARCHAR(50);

SELECT @TrangThai = TrangThai FROM The\_Nhap

WHERE MaTheNhap = @MaTheNhap; IF @TrangThai = 'DaNhap'

BEGIN

THROW 50007, 'Thẻ nhập đã được xác nhận, không thể chỉnh sửa thông tin!', 1;

END;

DECLARE @IdS INT; SELECT @IdS = IdS

FROM The\_Nhap

WHERE MaTheNhap = @MaTheNhap; DECLARE @IdTacGia INT;

IF NOT EXISTS (SELECT 1 FROM TAC\_GIA WHERE TenTacGia = @TenTacGia) INSERT INTO TAC\_GIA (TenTacGia) VALUES (@TenTacGia);

SET @IdTacGia = (SELECT IdTG FROM TAC\_GIA WHERE TenTacGia = @TenTacGia); DECLARE @IdTheLoai INT;

IF NOT EXISTS (SELECT 1 FROM THE\_LOAI WHERE TenTheLoai = @TenTheLoai)

INSERT INTO THE\_LOAI (TenTheLoai) VALUES (@TenTheLoai);

SET @IdTheLoai = (SELECT IdTL FROM THE\_LOAI WHERE TenTheLoai = @TenTheLoai); DECLARE @IdNXB INT;

IF NOT EXISTS (SELECT 1 FROM NHA\_XUAT\_BAN WHERE TenNXB = @TenNXB) INSERT INTO NHA\_XUAT\_BAN (TenNXB) VALUES (@TenNXB);

SET @IdNXB = (SELECT IdNXB FROM NHA\_XUAT\_BAN WHERE TenNXB = @TenNXB); UPDATE SACH

SET TenSach = @TenSach, NamXuatBan = @NamXuatBan, GiaSach = @GiaNhap, IdTacGia = @IdTacGia, IdTheLoai = @IdTheLoai, IdNXB = @IdNXB

WHERE IdS = @IdS;

DECLARE @TongTienNhap DECIMAL(10, 2) = @GiaNhap \* @SoLuong; UPDATE The\_Nhap

SET NgayNhap = @NgayNhap, GiaNhap = @GiaNhap, TongSoLuongNhap = @SoLuong, TongTienNhap = @TongTienNhap

WHERE MaTheNhap = @MaTheNhap; COMMIT TRANSACTION;

END TRY BEGIN CATCH

IF @@TRANCOUNT > 0

ROLLBACK TRANSACTION;

DECLARE @ErrorMessage NVARCHAR(4000) = ERROR\_MESSAGE();

DECLARE @ErrorSeverity INT = ERROR\_SEVERITY(); DECLARE @ErrorState INT = ERROR\_STATE();

RAISERROR (@ErrorMessage, @ErrorSeverity, @ErrorState); END CATCH;

END;

**Transaction xóa sách khỏi bảng nhập**

CREATE PROCEDURE sp\_XoaSach @MaTheNhap VARCHAR(50), @MaSach VARCHAR(50)

AS BEGIN

SET NOCOUNT ON; BEGIN TRY

IF NOT EXISTS (SELECT 1 FROM The\_Nhap WHERE MaTheNhap = @MaTheNhap) BEGIN

THROW 50002, 'Mã thẻ nhập không tồn tại!', 1;

END;

DECLARE @TrangThai VARCHAR(50);

SELECT @TrangThai = TrangThai FROM The\_Nhap

WHERE MaTheNhap = @MaTheNhap; IF @TrangThai = 'DaNhap'

BEGIN

THROW 50007, 'Thẻ nhập đã được xác nhận, không thể xóa!', 1;

END;

IF NOT EXISTS (SELECT 1 FROM SACH WHERE MaSach = @MaSach) BEGIN

THROW 50010, 'Mã sách không tồn tại!', 1;

END;

DECLARE @IdS INT; SELECT @IdS = IdS

FROM The\_Nhap

WHERE MaTheNhap = @MaTheNhap;

DECLARE @IdTacGia INT, @IdTheLoai INT, @IdNXB INT;

SELECT @IdTacGia = IdTacGia, @IdTheLoai = IdTheLoai, @IdNXB = IdNXB FROM SACH

WHERE MaSach = @MaSach; BEGIN TRANSACTION;

DELETE FROM The\_Nhap WHERE MaTheNhap = @MaTheNhap; DELETE FROM SACH WHERE MaSach = @MaSach;

IF NOT EXISTS (SELECT 1 FROM SACH WHERE IdTacGia = @IdTacGia) BEGIN

DELETE FROM TAC\_GIA WHERE IdTG = @IdTacGia;

END;

IF NOT EXISTS (SELECT 1 FROM SACH WHERE IdTheLoai = @IdTheLoai) BEGIN

DELETE FROM THE\_LOAI WHERE IdTL = @IdTheLoai;

END;

IF NOT EXISTS (SELECT 1 FROM SACH WHERE IdNXB = @IdNXB) BEGIN

DELETE FROM NHA\_XUAT\_BAN WHERE IdNXB = @IdNXB;

END;

COMMIT TRANSACTION; END TRY

BEGIN CATCH

IF @@TRANCOUNT > 0

ROLLBACK TRANSACTION;

DECLARE @ErrorMessage NVARCHAR(4000) = ERROR\_MESSAGE();

DECLARE @ErrorSeverity INT = ERROR\_SEVERITY(); DECLARE @ErrorState INT = ERROR\_STATE();

RAISERROR (@ErrorMessage, @ErrorSeverity, @ErrorState); END CATCH;

END;

**Transaction thêm ảnh bìa vào sách**

CREATE PROCEDURE sp\_CapNhatAnhBia @MaSach VARCHAR(50), @AnhBia VARBINARY(MAX)

AS BEGIN

SET NOCOUNT ON; BEGIN TRY

BEGIN TRANSACTION;

IF NOT EXISTS (SELECT 1 FROM SACH WHERE MaSach = @MaSach)

THROW 50010, 'Mã sách không tồn tại!', 1; UPDATE SACH

SET AnhBia = @AnhBia WHERE MaSach = @MaSach; COMMIT TRANSACTION;

END TRY

BEGIN CATCH

IF @@TRANCOUNT > 0

ROLLBACK TRANSACTION;

DECLARE @ErrorMessage NVARCHAR(4000) = ERROR\_MESSAGE();

DECLARE @ErrorSeverity INT = ERROR\_SEVERITY(); DECLARE @ErrorState INT = ERROR\_STATE();

RAISERROR (@ErrorMessage, @ErrorSeverity, @ErrorState); END CATCH;

END;

**Transaction cập nhật thêm bớt số lượng sách vào kho sau khi đã nhập sách**

CREATE PROCEDURE sp\_CapNhatKhoSach @IdS INT,

@SoLuongThem INT,

@ResultMessage NVARCHAR(4000) OUTPUT

AS BEGIN

SET NOCOUNT ON; BEGIN TRY

DECLARE @SoLuongHienTai INT; DECLARE @TongSoLuongNhap INT; DECLARE @HasNhapKho BIT; DECLARE @SoLuongMoi INT; BEGIN TRANSACTION;

IF NOT EXISTS (SELECT 1 FROM SACH WHERE IdS = @IdS) BEGIN

END;

SET @ResultMessage = N'Không có thông tin trong kho'; ROLLBACK TRANSACTION;

RETURN 1;

'DaNhap');

SET @HasNhapKho = (SELECT COUNT(\*) FROM The\_Nhap WHERE IdS = @IdS AND TrangThai =

IF @HasNhapKho = 0 BEGIN

END;

SET @ResultMessage = N'Không có thông tin trong kho'; ROLLBACK TRANSACTION;

RETURN 1;

SELECT @SoLuongHienTai = ISNULL(SoLuongHienTai, 0) FROM Kho\_Sach

WHERE MaSach = @IdS;

SET @SoLuongMoi = @SoLuongHienTai + @SoLuongThem; IF @SoLuongMoi < 0

BEGIN

SET @ResultMessage = N'Số lượng trong kho hiện tại là ' + CAST(@SoLuongHienTai AS NVARCHAR(10)) + N' không đủ sách để giảm ' + CAST(ABS(@SoLuongThem) AS NVARCHAR(10)) + N'\nCập nhật thất bại';

ROLLBACK TRANSACTION; RETURN 1;

END;

SELECT @TongSoLuongNhap = ISNULL(SUM(TongSoLuongNhap), 0) FROM The\_Nhap

WHERE IdS = @IdS AND TrangThai = 'DaNhap'; IF @SoLuongMoi > @TongSoLuongNhap

BEGIN

SET @ResultMessage = N'Số lượng thêm vượt quá số lượng đã nhập là ' + CAST(@TongSoLuongNhap AS

NVARCHAR(10)) + N'\nCập nhật thất bại';

ROLLBACK TRANSACTION; RETURN 1;

END;

IF EXISTS (SELECT 1 FROM Kho\_Sach WHERE MaSach = @IdS) BEGIN

END

UPDATE Kho\_Sach

SET SoLuongHienTai = @SoLuongMoi,

TrangThaiSach = CASE WHEN @SoLuongMoi > 0 THEN N'ConSach' ELSE N'HetSach' END WHERE MaSach = @IdS;

ELSE IF @SoLuongThem > 0 BEGIN

END;

INSERT INTO Kho\_Sach (MaSach, SoLuongHienTai, TrangThaiSach) VALUES (@IdS, @SoLuongThem, N'ConSach');

SET @ResultMessage = N'Cập nhật thành công'; COMMIT TRANSACTION;

RETURN 0; END TRY

BEGIN CATCH

IF @@TRANCOUNT > 0

ROLLBACK TRANSACTION;

DECLARE @ErrorMessage NVARCHAR(4000) = ERROR\_MESSAGE();

DECLARE @ErrorSeverity INT = ERROR\_SEVERITY(); DECLARE @ErrorState INT = ERROR\_STATE();

RAISERROR (@ErrorMessage, @ErrorSeverity, @ErrorState); SET @ResultMessage = @ErrorMessage;

RETURN 1;

END CATCH; END;

**Transaction kiểm tra trạng thái sách trong kho sau khi đã nhập**

CREATE FUNCTION fn\_KiemTraTrangThaiKhoSach (@p\_IdS INT) RETURNS NVARCHAR(300)

AS BEGIN

DECLARE @KetQua NVARCHAR(300); DECLARE @MaSach NVARCHAR(50); DECLARE @TenSach NVARCHAR(100);

DECLARE @SoLuong INT;

DECLARE @TrangThai NVARCHAR(50); DECLARE @MaNV NVARCHAR(50); DECLARE @HoTen NVARCHAR(50); DECLARE @MaTheNhap NVARCHAR(50);

SELECT @MaSach = MaSach, @TenSach = TenSach FROM SACH

WHERE IdS = @p\_IdS; IF @MaSach IS NULL BEGIN

SET @KetQua = N'Mã sách ' + CAST(@p\_IdS AS NVARCHAR(10)) + N' không tồn tại trong danh mục

sách';

RETURN @KetQua;

END;

SELECT TOP 1

@MaNV = nv.MaNV, @HoTen = nv.HoTen,

@MaTheNhap = tn.MaTheNhap,

@SoLuong = ISNULL(ks.SoLuongHienTai, 0), @TrangThai = CASE

WHEN ISNULL(ks.SoLuongHienTai, 0) > 0 THEN N'Còn Sách'

ELSE N'Hết Sách'

END

FROM The\_Nhap tn

LEFT JOIN NhanVien nv ON tn.IdNV = nv.IdNV LEFT JOIN Kho\_Sach ks ON tn.IdS = ks.MaSach

WHERE tn.IdS = @p\_IdS AND tn.TrangThai = 'DaNhap'

ORDER BY tn.NgayNhap DESC; IF @MaNV IS NULL

BEGIN

SET @KetQua = N'Mã sách ' + @MaSach + N' chưa có dữ liệu nhập kho'; RETURN @KetQua;

END;

END;

SET @KetQua = N'Mã nhân viên: ' + @MaNV + NCHAR(13) + NCHAR(10) + N'Tên nhân viên: ' + @HoTen + NCHAR(13) + NCHAR(10) + N'Mã thẻ nhập: ' + @MaTheNhap + NCHAR(13) + NCHAR(10) + N'Mã sách: ' + @MaSach + NCHAR(13) + NCHAR(10) +

N'Tên sách: ' + @TenSach + NCHAR(13) + NCHAR(10) +

N'Số lượng: ' + CAST(@SoLuong AS NVARCHAR(10)) + NCHAR(13) + NCHAR(10) +

N'Trạng thái: ' + @TrangThai; RETURN @KetQua;

**Transaction duyệt xác nhận nhập vào kho sau khi đã nhập sách vào bảng nhập**

CREATE PROCEDURE sp\_XacNhanNhap @MaTheNhap VARCHAR(50)

AS BEGIN

SET NOCOUNT ON; BEGIN TRY

BEGIN TRANSACTION;

IF NOT EXISTS (SELECT 1 FROM The\_Nhap WHERE MaTheNhap = @MaTheNhap AND TrangThai =

'ChuaNhap')

BEGIN

THROW 50003, 'Mã thẻ nhập không tồn tại hoặc đã được xác nhận!', 1;

END;

DECLARE @IdS INT, @SoLuong INT;

SELECT @IdS = IdS, @SoLuong = TongSoLuongNhap FROM The\_Nhap

WHERE MaTheNhap = @MaTheNhap; UPDATE The\_Nhap

SET TrangThai = 'DaNhap'

WHERE MaTheNhap = @MaTheNhap; DECLARE @ResultMessage NVARCHAR(4000);

EXEC sp\_CapNhatKhoSach @IdS, @SoLuong, @ResultMessage OUTPUT; IF @ResultMessage != N'Cập nhật thành công'

BEGIN

THROW 50004, @ResultMessage, 1;

END;

COMMIT TRANSACTION; END TRY

BEGIN CATCH

IF @@TRANCOUNT > 0

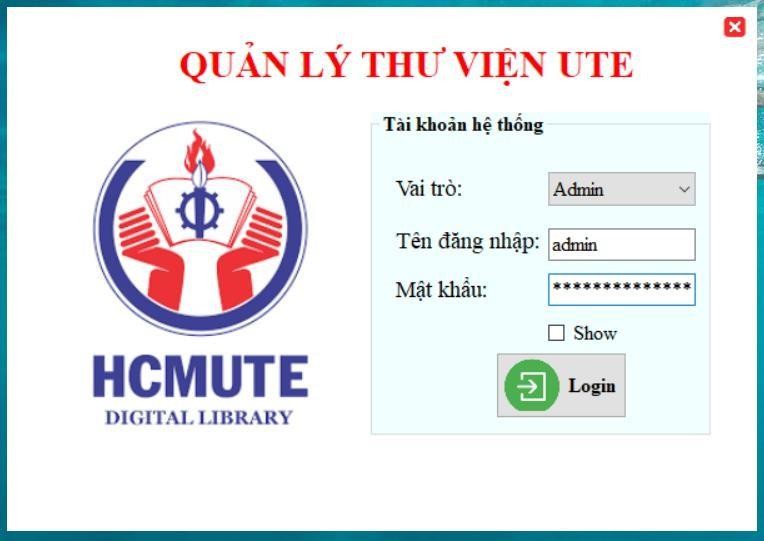
ROLLBACK TRANSACTION; THROW;

END CATCH;

END;

# CHƯƠNG 6: THIẾT KẾ GIAO DIỆN

1. **Môi trường cài đặt và công nghệ sử dụng**
   * Hệ cơ sở dữ liệu Microsoft SQL Server Management Studio 20.
   * Chương trình Winform được viết bằng Microsoft Visual Studio 2022 - trên nền tảng ASP.NET Web Application (.NET Framework) C#.
2. **Phân quyền vai trò đăng nhập**

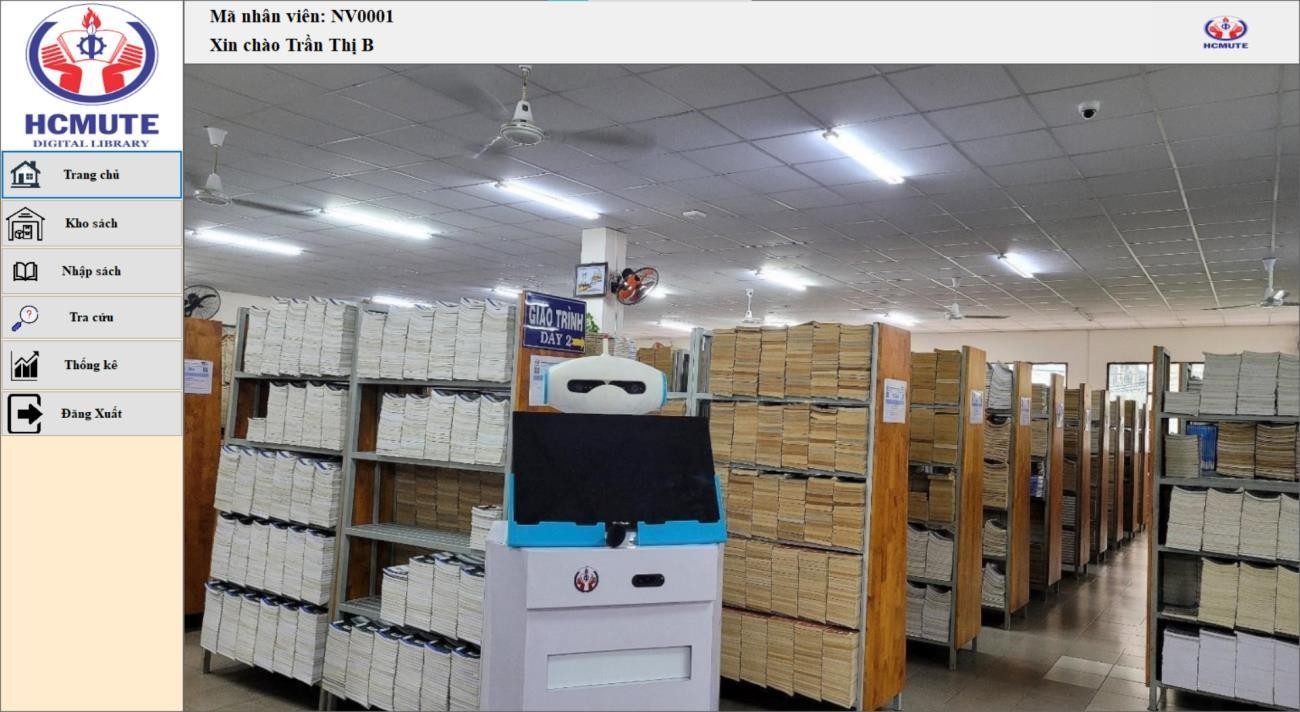
****

Hình 1: Form đăng nhập vai trò Admin

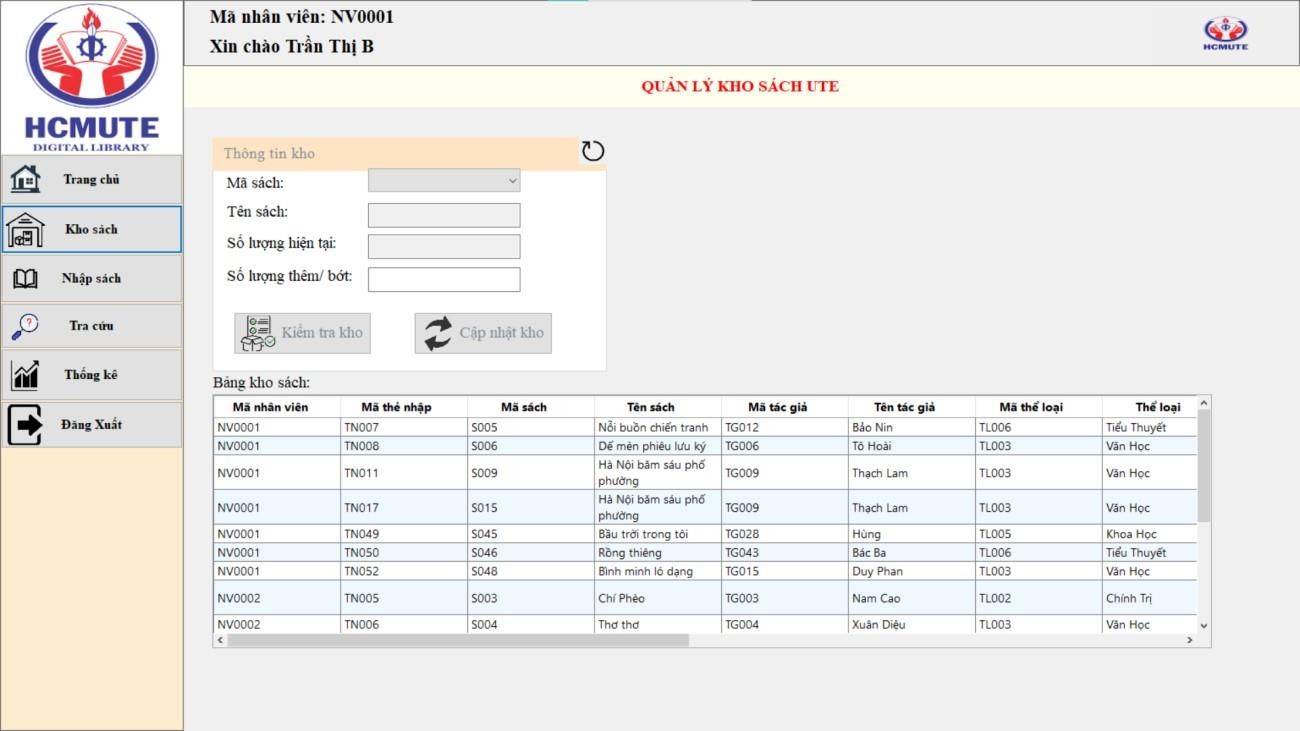


Hình 2: Form đăng nhập vai trò Nhân viên

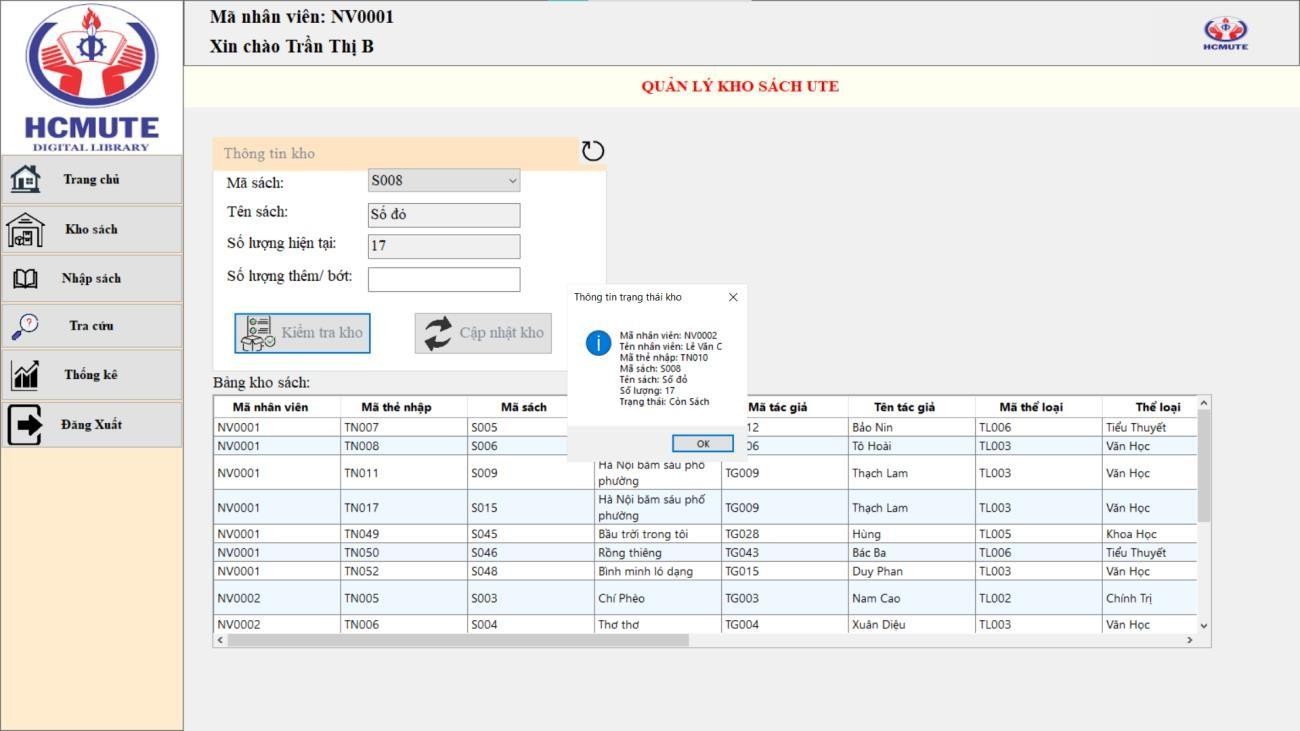
1. **Phân quyền chức năng**
   1. **Chức năng chung của Admin và Nhân viên**

****

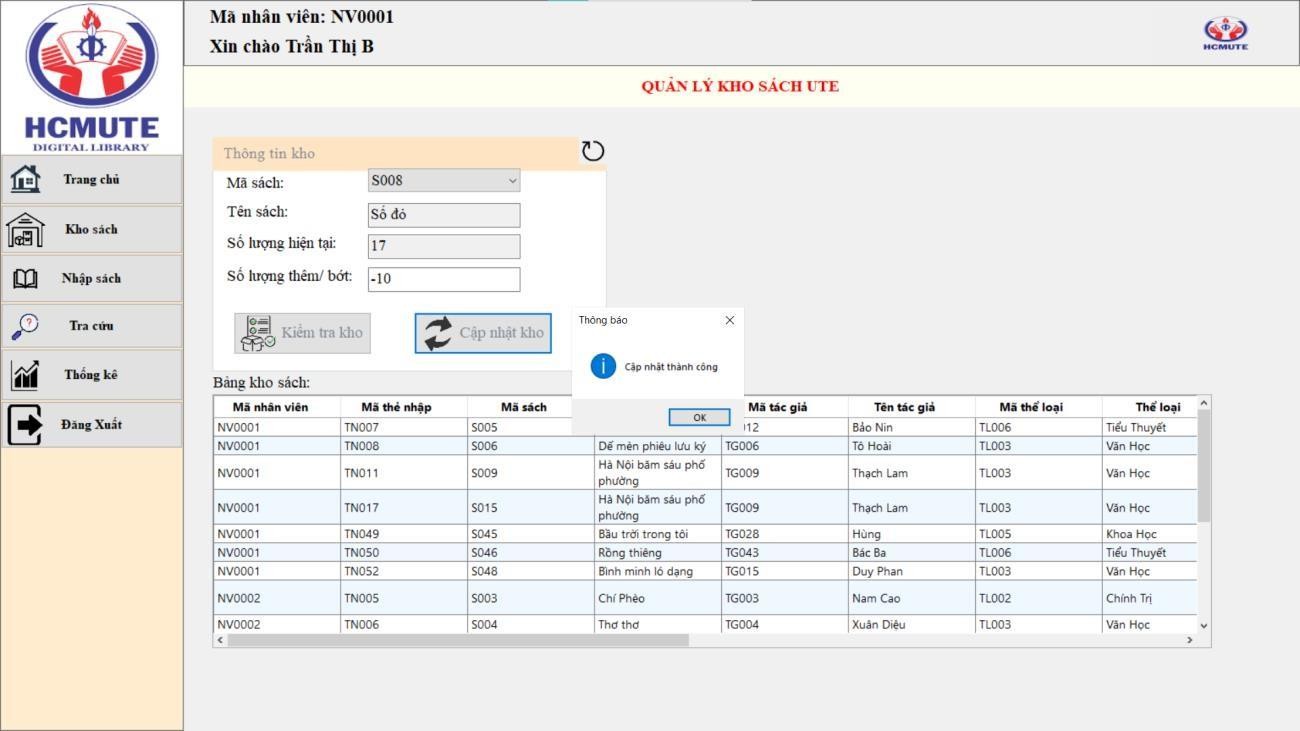
Hình 3: Form trang chủ của hệ thống quản lý nhập sách



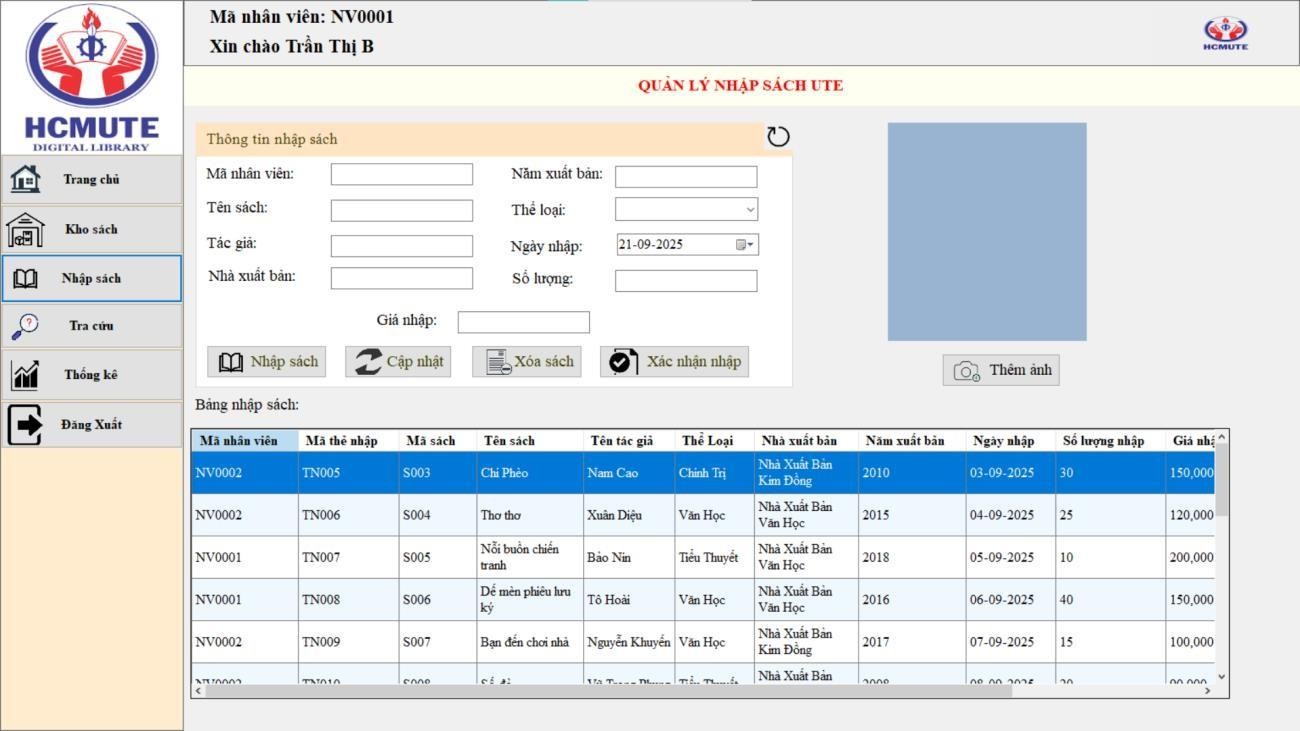
Hình 4: Form kho sách của hệ thống quản lý nhập sách

****

Hình 5: Form kho sách thực hiện chức năng kiểm tra kho



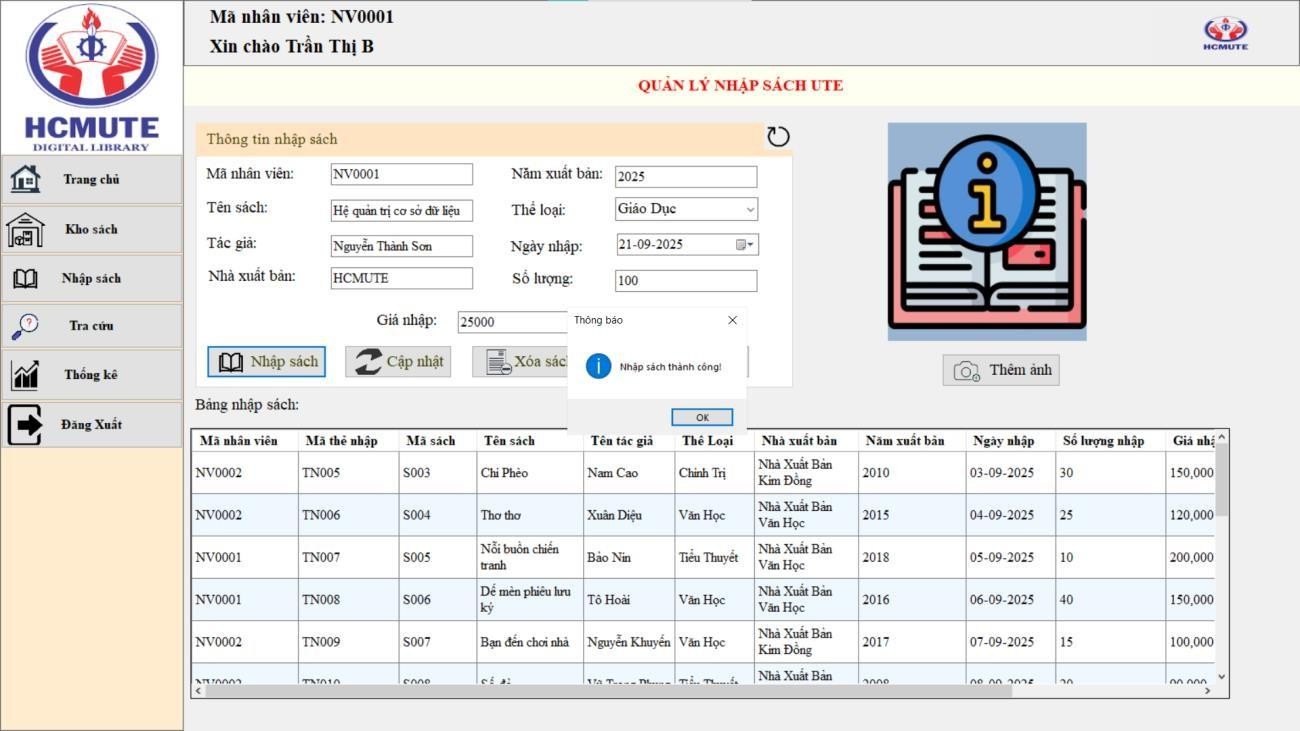
Hình 6: Form kho sách thực hiện chức năng cập nhật kho

****

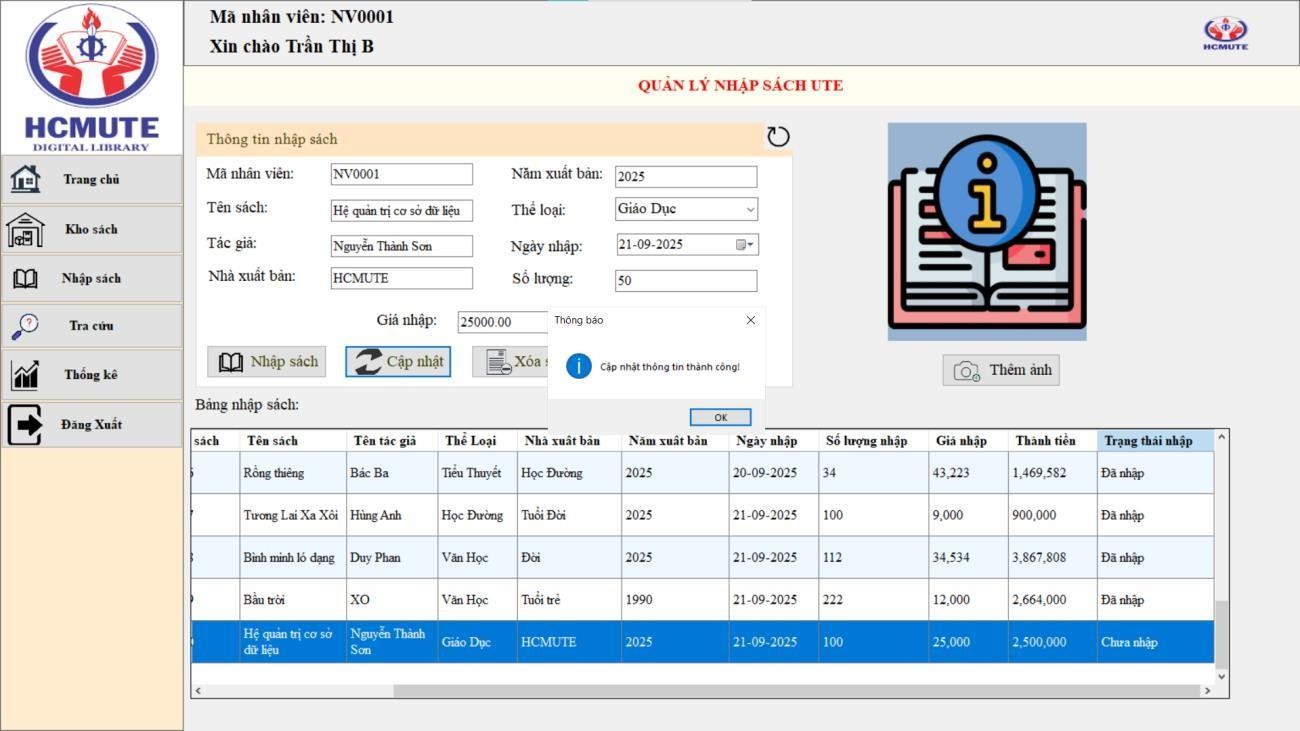
Hình 7: Form nhập sách của hệ thống quản lý nhập sách



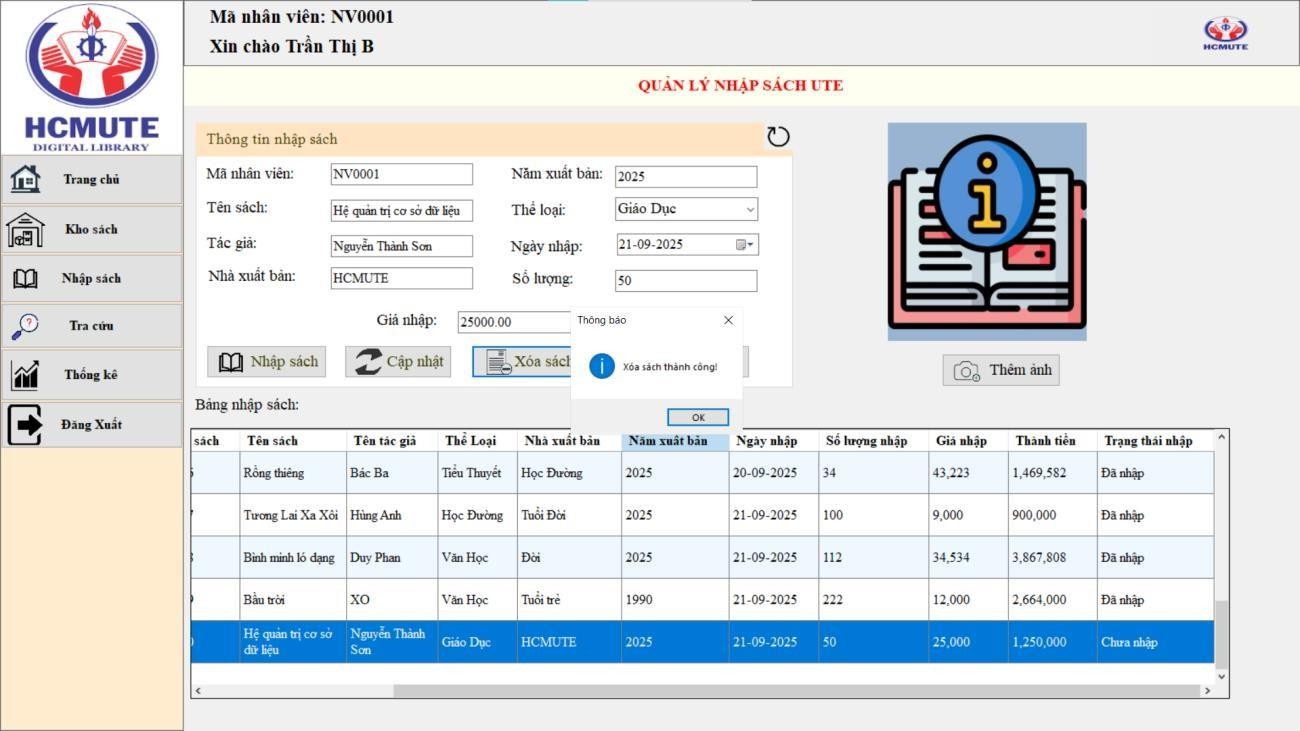
Hình 8: Form nhập sách vào bảng nhập sách

****

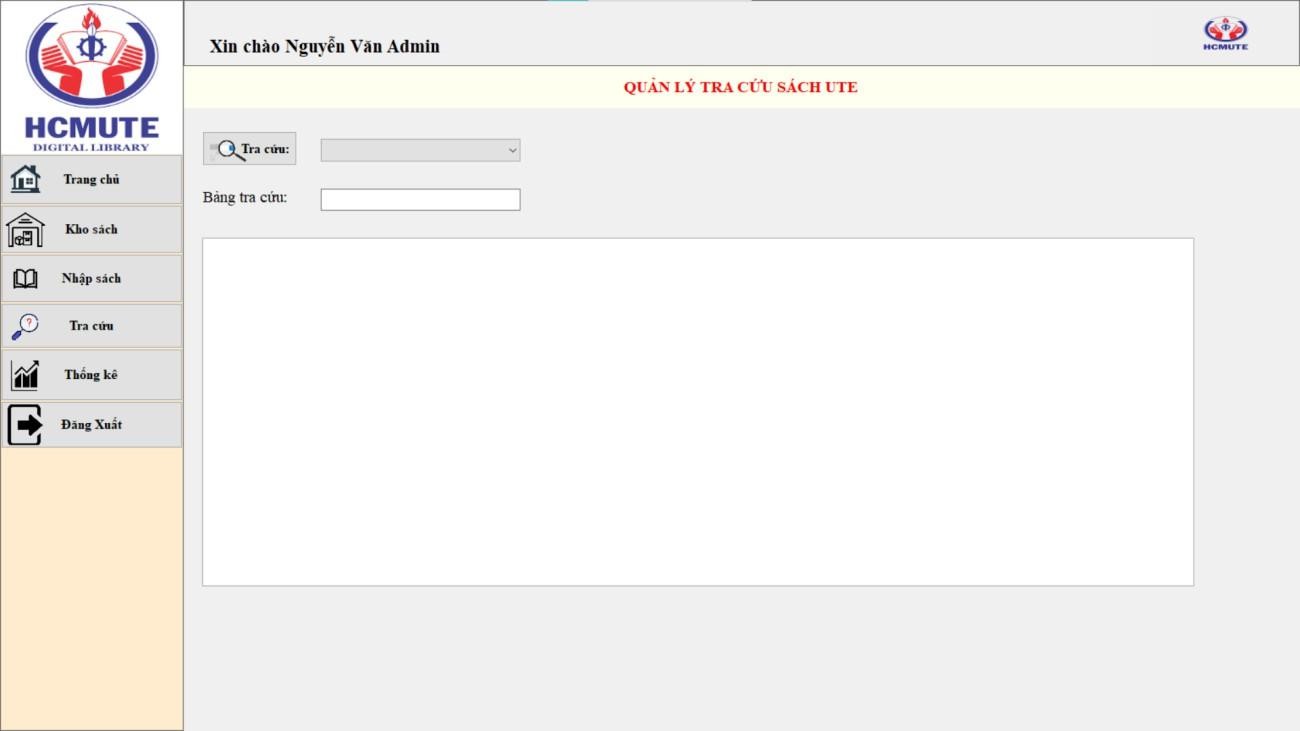
Hình 9: Form nhập sách thành công vào bảng nhập sách



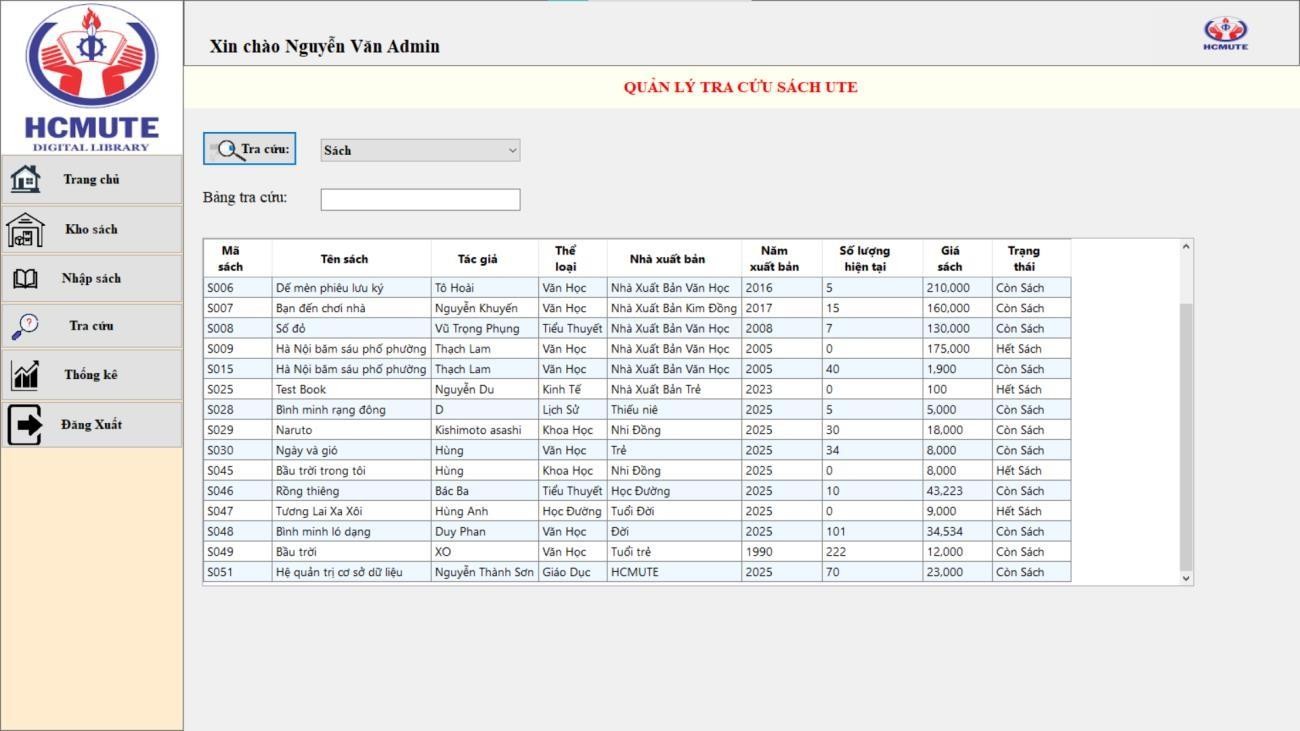
Hình 10: Form nhập sách cập nhật thông tin

****

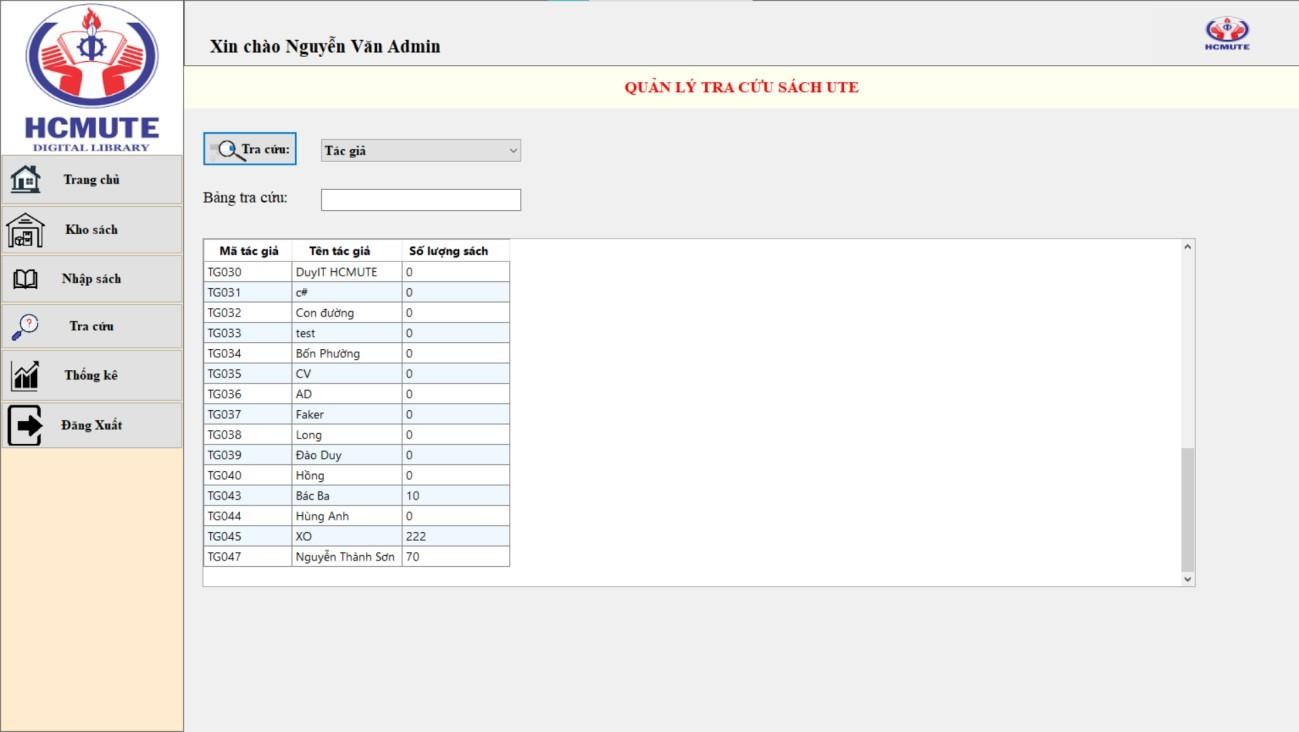
Hình 11: Form nhập sách xóa sách nhập khỏi bảng nhập sách

****

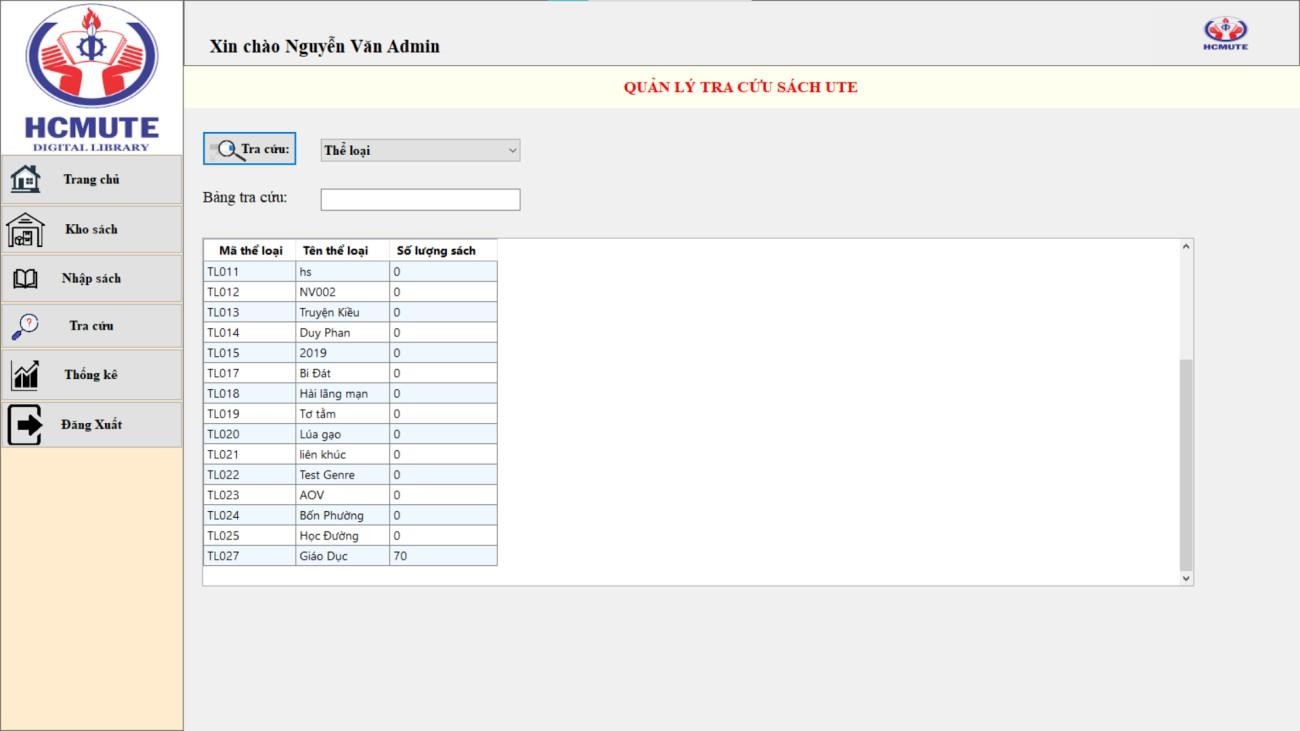
Hình 12: Form tra cứu của hệ thống quản lý nhập sách



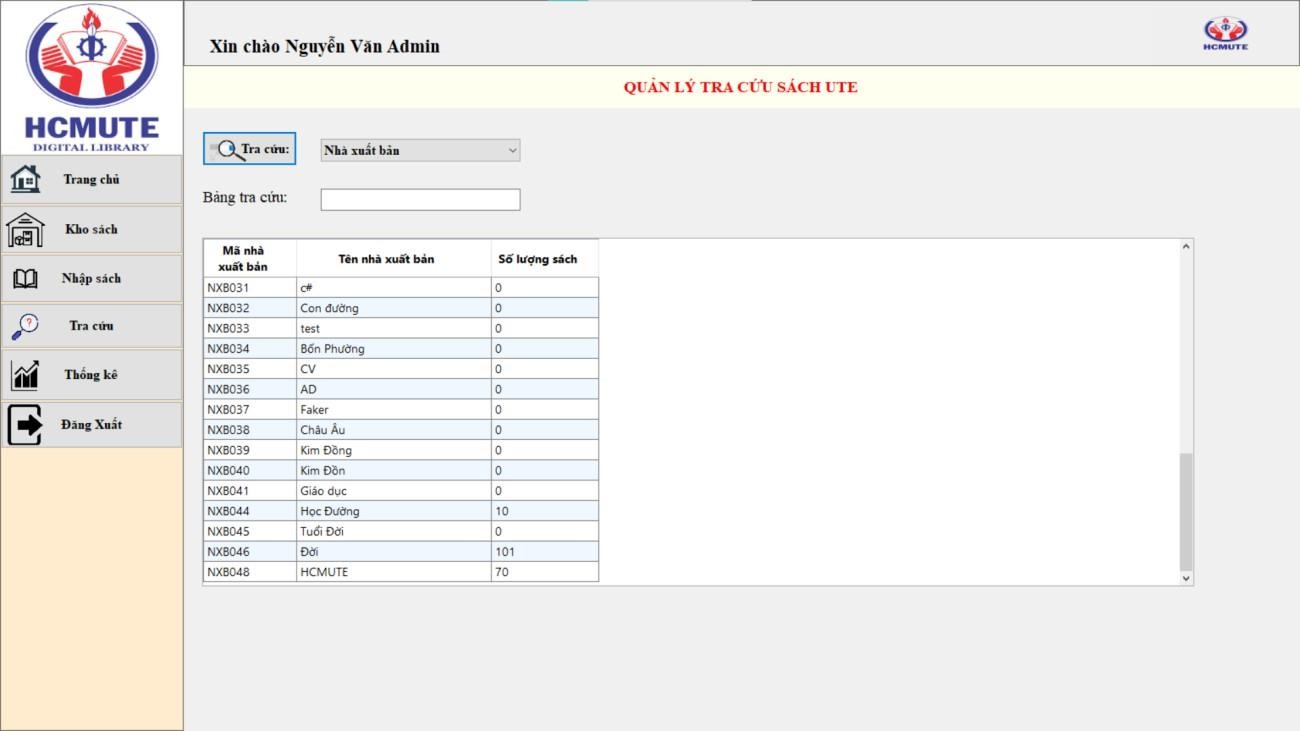
Hình 13: Form tra cứu sách của hệ thống quản lý nhập sách

****

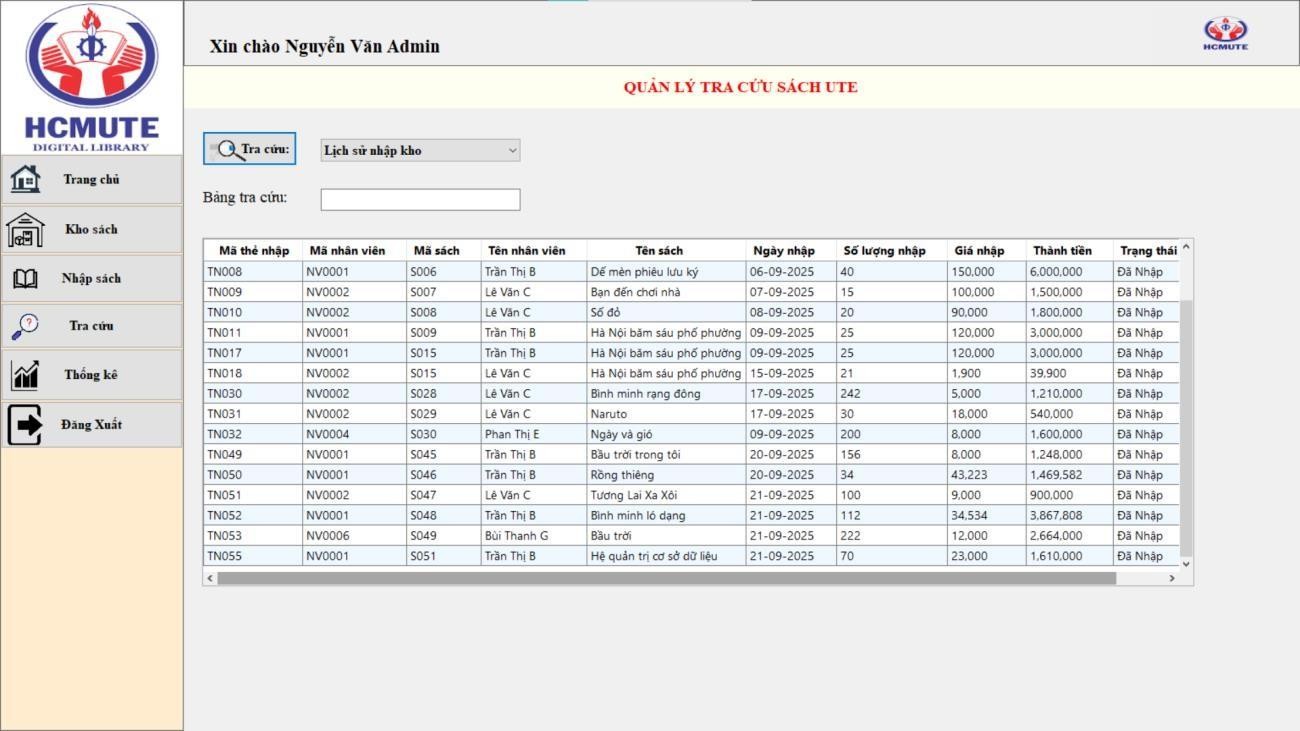
Hình 14: Form tra cứu tác giả của hệ thống quản lý nhập sách



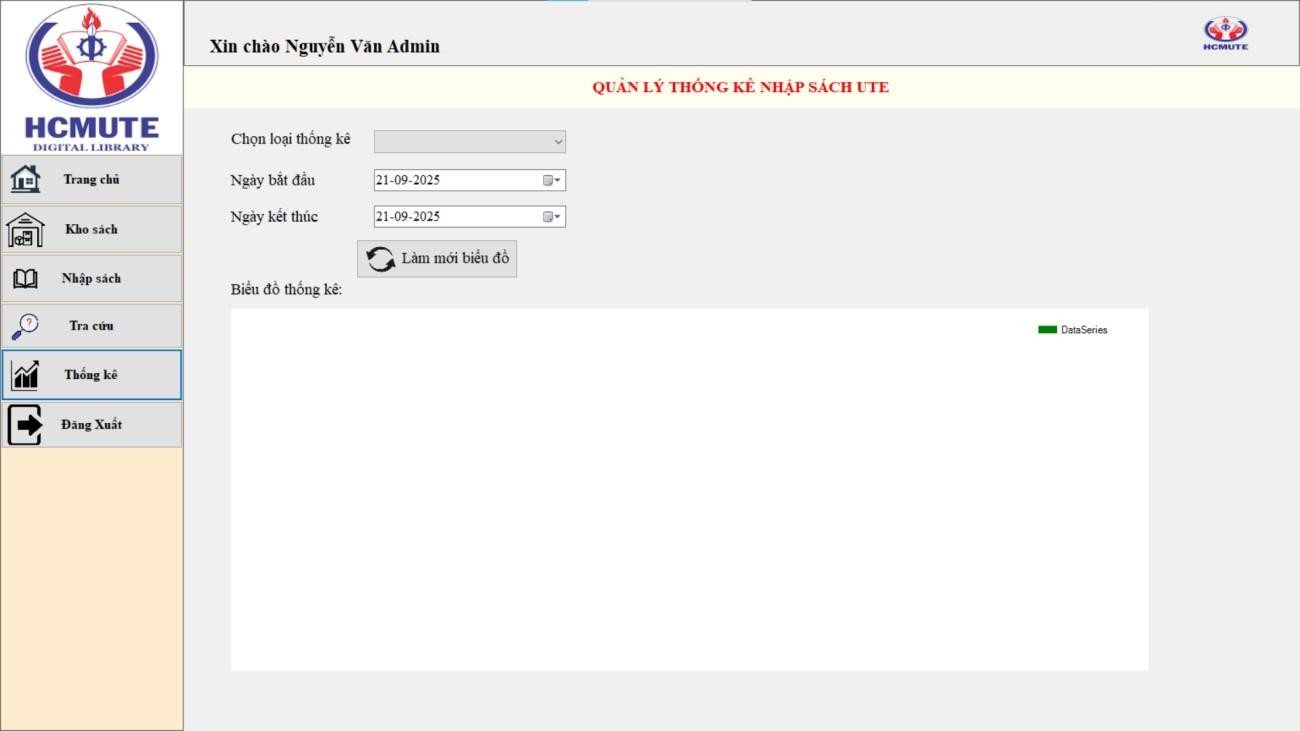
Hình 15: Form tra cứu thể loại của hệ thống quản lý nhập sách

****

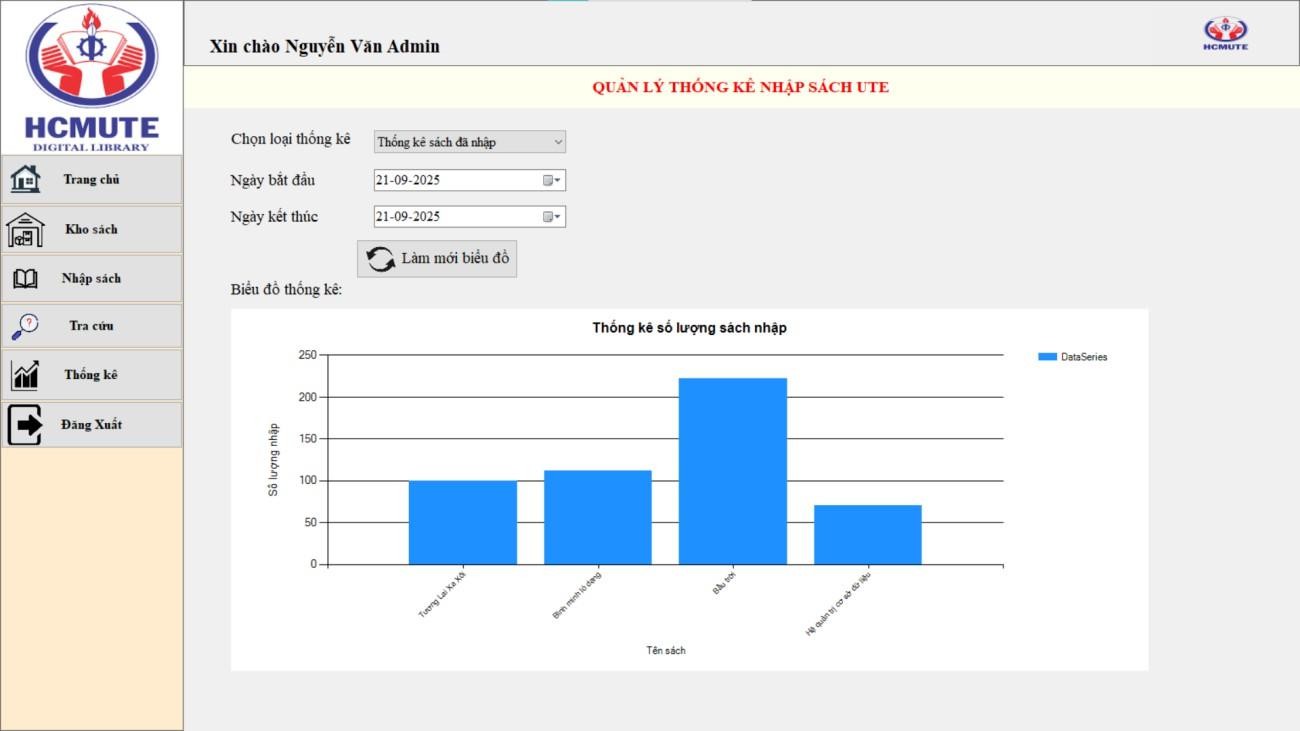
Hình 16: Form tra cứu nhà xuất bản của hệ thống quản lý nhập sách



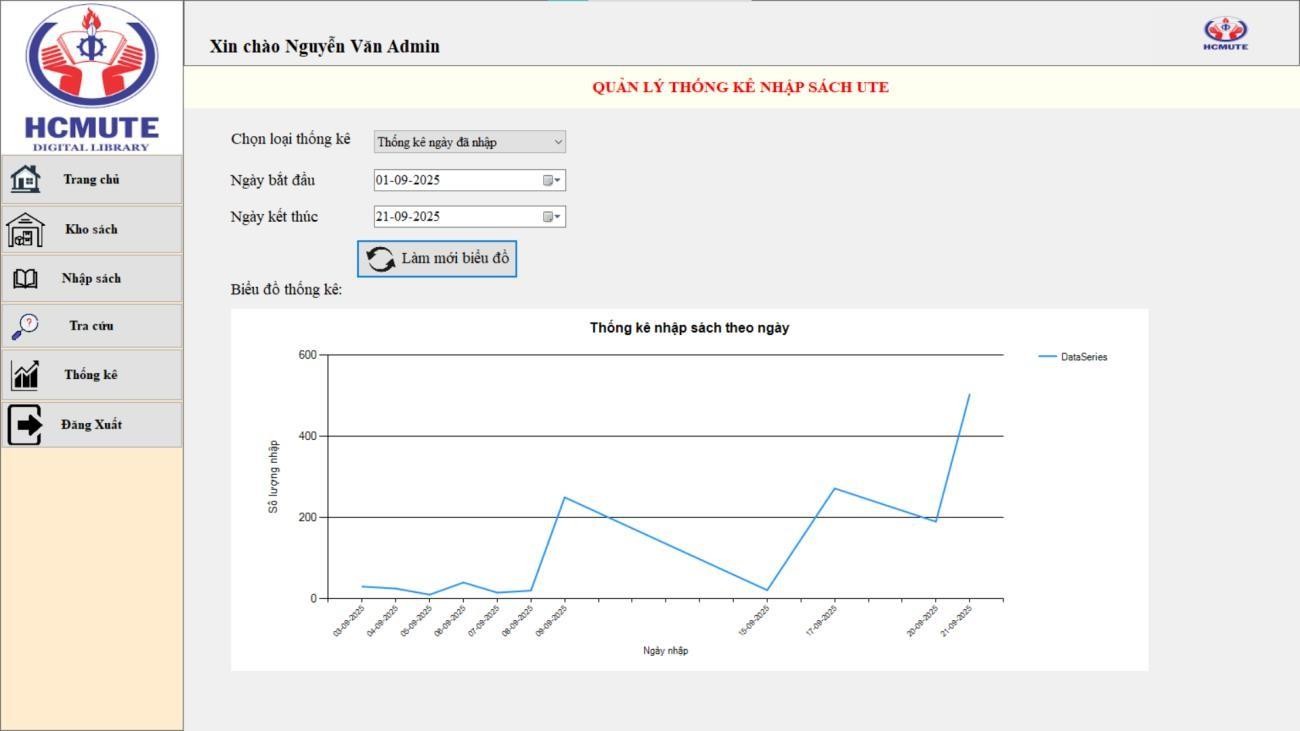
Hình 17: Form tra cứu lịch sử nhập kho của hệ thống quản lý nhập sách

****

Hình 18: Form thống kê của hệ thống quản lý nhập sách

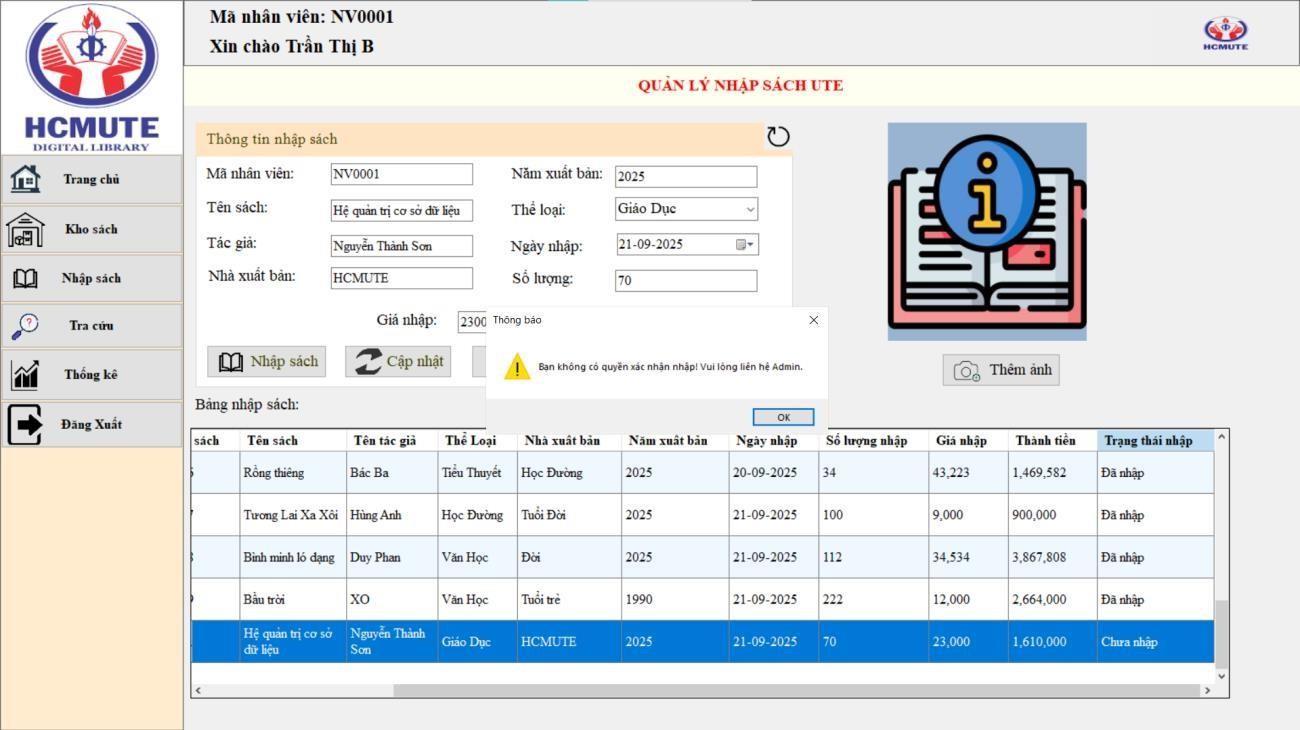


Hình 19: Form thống kê theo sách đã nhập

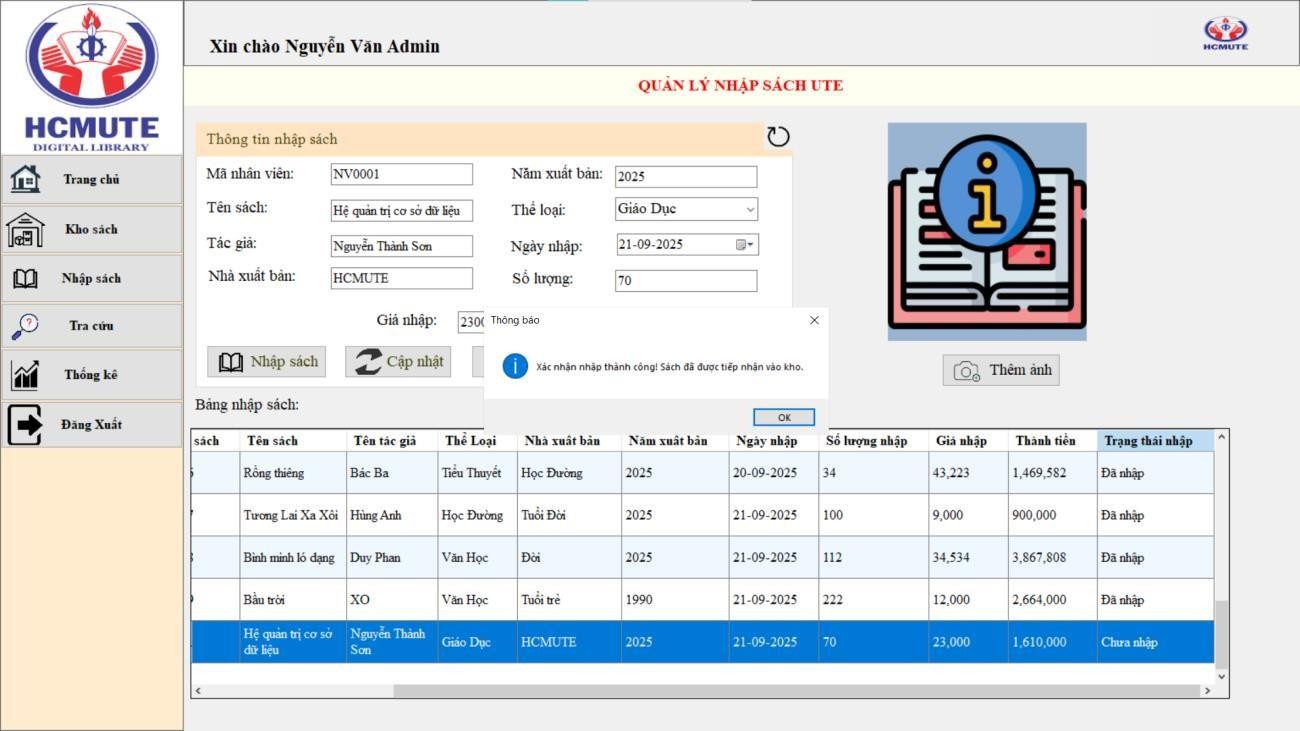
****

Hình 20: Form thống kê theo ngày đã nhập

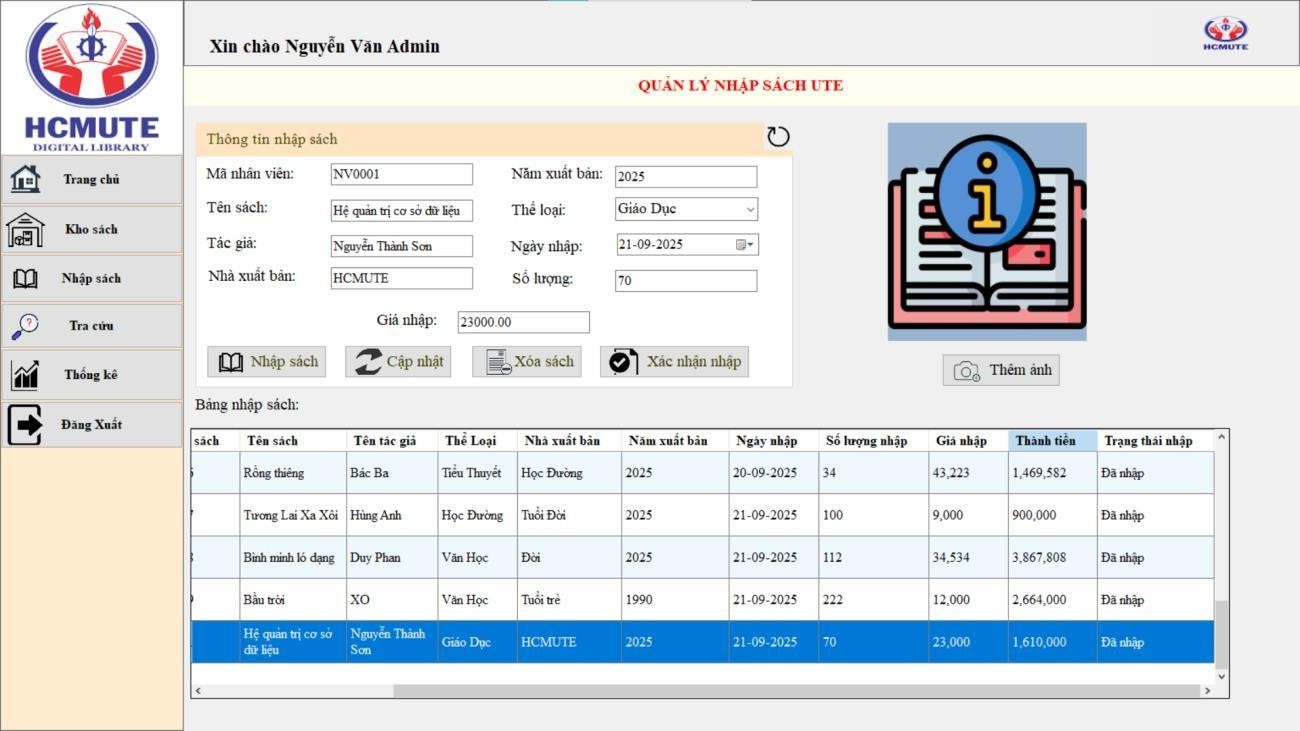
* 1. **Chức năng riêng của Admin**



Hình 21: Form nhập sách phân quyền NhanVienRole không được xác nhận nhập vào kho

****

Hình 22: Form nhập sách phân quyền AdminRole được xác nhận nhập vào kho



Hình 23: Form nhập sách sau khi được Admin xác nhận nhập