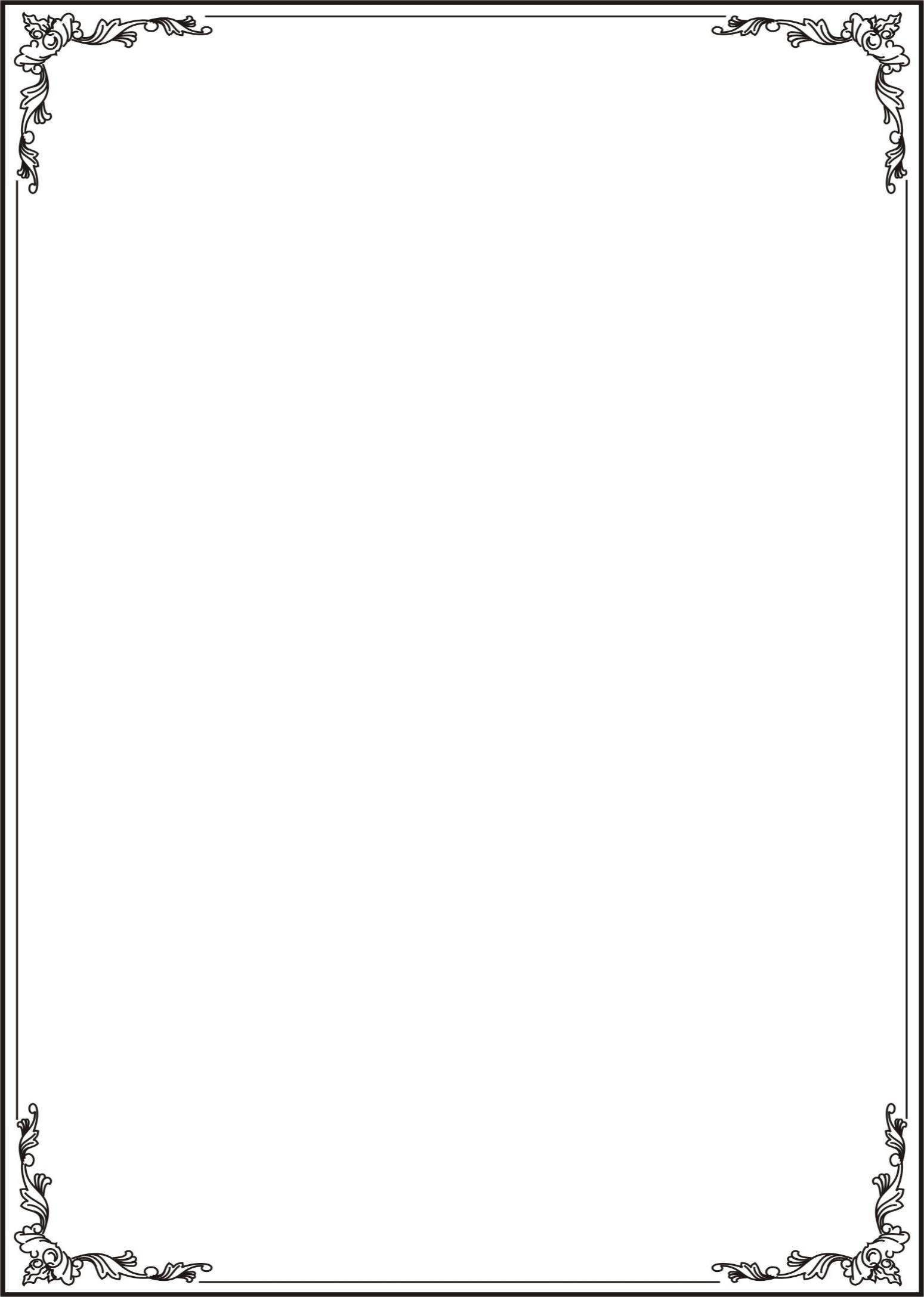
****TRƯỜNG ĐẠI HỌC SƯ PHẠM KĨ THUẬT TP.HỒ CHÍ MINH

KHOA ĐÀO TẠO QUỐC TẾ

--------------



**Môn học:** Database Management System

**TÊN ĐỀ TÀI:**

**QUẢN LÝ MƯỢN/TRẢ SÁCH TRONG THƯ VIỆN TRƯỜNG**

GVHD: Nguyễn Thành Sơn

Lớp HP: DBMSE02FIE

Nhóm thực hiện: Nhóm 01

Học kỳ: 2

Năm học: 2024 – 2025

DANH SÁCH SINH VIÊN NHÓM THỰC HIỆN

HỌC KÌ 2 NĂM HỌC 2024 - 2025

Nhóm 01

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Họ và tên | MSSV | Nhiệm vụ | Mức độ hoàn thành |
| 1 | Nguyễn Minh Phú Quang | 22110064 | Phân công, tổng hợp, chỉnh sửa | 100% |
| 2 | Trần Quốc Huy | 23110025 | Yêu cầu dữ liệu | 100% |
| 3 | Nguyễn Tịnh Nhân | 23110047 | Yêu cầu giao diện | 100% |
| 4 | Nguyễn Khôi Nguyên | 23110044 | Yêu cầu chức năng | 100% |

NHẬN XÉT CỦA GV

TP. Hồ Chí Minh, ngày......., tháng 2, năm 2025

Giảng viên chấm điểm

# **MỤC LỤC**

Contents

LỜI CẢM ƠN.................................................................................................................. 1

LỜI NÓI ĐẦU.................................................................................................................. 2

CHƯƠNG 1. TỔNG QUAN VỀ HỆ THỐNG............................................................................. 3

1. Đặc tả đề tài.................................................................................................................. 3

1.1. Mô tả bài toán........................................................................................................ 3

1.2. Nghiệp vụ của bài toán............................................................................................ 5

1.3. Mô tả các chức năng của bài toán............................................................................ 6

1.4. Giao diện................................................................................................................ 6

# **LỜI CẢM ƠN**

       Lời đầu tiên, chúng em xin chân thành cảm ơn sự chỉ bảo tận tình của thầy *Nguyễn Thành Sơn*, đã giúp chúng em hoàn thành bài báo cáo cuối kì bộ môn Hệ quản trị cơ sở dữ liệu suôn sẻ. Nhờ những kiến thức thầy hướng dẫn và truyền đạt đã góp phần giúp cho chúng em nắm chắc được những kiến thức cơ bản và chuyên sâu đối với các hệ quản trị cơ sở dữ liệu. Thông qua đồ án này, chúng em xin phép được trình bày một hệ thống quản trị cơ sở dữ liệu và lập trình hệ thống liên quan.

       Không thể tránh khỏi những sai sót trong quá trình làm bài, Vậy nên chúng em rất mong nhận được sự góp ý của thầy nhằm ngày càng hoàn thiện đồ án, phát triển và đầy đủ hơn. Kính chúc thầy sức khỏe, thành công trên con đường truyền dạy tri thức trong tương lai.

       Cuối cùng, chúng em xin chân thành cảm ơn tất cả thầy cô và các bạn đã cố gắng và giúp đỡ chúng em trong quá trình hoàn thành đồ án bộ môn Hệ quản trí cơ sở dữ liệu này.

# **LỜI NÓI ĐẦU**

       Trong thời kỳ cách mạng công nghiệp 4.0, việc ứng dụng công nghệ vào các lĩnh vực đời sống là điều tất yếu, góp phần nâng cao hiệu quả quản lý và tối ưu hóa quy trình làm việc. Từ giáo dục, y tế, kinh tế đến các lĩnh vực dịch vụ công cộng, công nghệ thông tin đóng vai trò quan trọng trong việc số hóa dữ liệu, giảm thiểu các thủ tục hành chính rườm rà và tăng cường tính minh bạch trong quản lý.

Hiện nay, tại các thư viện truyền thống, việc quản lý sách và quá trình mượn - trả vẫn gặp nhiều khó khăn do phương thức ghi chép thủ công. Việc tra cứu sách, kiểm soát số lượng sách còn hay đã mượn, cũng như theo dõi lịch sử mượn sách của bạn đọc đôi khi mất nhiều thời gian và dễ xảy ra sai sót. Điều này gây ảnh hưởng đến trải nghiệm của người đọc cũng như làm giảm hiệu quả hoạt động của thư viện.

Chính vì thế, việc ứng dụng công nghệ vào quản lý mượn trả sách là điều cần thiết. Hệ thống quản lý dữ liệu thư viện giúp tự động hóa quá trình mượn - trả, lưu trữ thông tin sách và người đọc một cách khoa học, đồng thời hỗ trợ tra cứu nhanh chóng, tiết kiệm thời gian và nâng cao hiệu suất hoạt động.

Với mong muốn tìm hiểu sâu hơn về ứng dụng công nghệ trong lĩnh vực này, nhóm chúng em đã chọn đề tài “Quản lý mượn trả sách thư viện”. Thông qua đề tài này, chúng em sẽ nghiên cứu cách xây dựng một hệ thống quản lý dữ liệu hiệu quả, góp phần hiện đại hóa hoạt động thư viện và mang lại nhiều lợi ích cho người sử dụng.

# **CHƯƠNG 1. TỔNG QUAN VỀ HỆ THỐNG**

## **1. Đặc tả đề tài**

### **1.1. Mô tả bài toán**

Thư viện trường học đóng vai trò quan trọng trong việc hỗ trợ học sinh, sinh viên và giáo viên tiếp cận tài liệu học tập, nghiên cứu. Tuy nhiên, việc quản lý thủ công thông tin sách, người mượn và quá trình mượn/trả có thể gặp nhiều vấn đề như thất lạc sách, quá hạn mượn không được kiểm soát, khó khăn trong việc tìm kiếm thông tin sách và người đọc. Những hạn chế này không chỉ gây bất tiện cho người sử dụng mà còn ảnh hưởng đến hiệu quả hoạt động của thư viện.

Hệ thống "Quản lý mượn/trả sách thư viện trường" được thiết kế nhằm tự động hóa quá trình theo dõi, quản lý việc mượn, trả sách, giúp thư viện hoạt động hiệu quả hơn. Hệ thống sẽ lưu trữ thông tin về sách (tựa sách, tác giả, thể loại, số lượng, vị trí), thông tin độc giả (học sinh, sinh viên, giáo viên) và lịch sử mượn/trả. Bên cạnh đó, hệ thống hỗ trợ kiểm tra tình trạng sách theo thời gian thực, nhắc nhở khi gần đến hạn trả hoặc khi sách bị quá hạn.

Ngoài ra, hệ thống sẽ tích hợp các tính năng như tìm kiếm sách nhanh chóng, phân quyền quản trị viên và độc giả, xuất báo cáo thống kê về sách và tình trạng mượn/trả. Việc áp dụng công nghệ vào quản lý thư viện không chỉ giảm thiểu sai sót, tiết kiệm thời gian mà còn nâng cao trải nghiệm của người dùng, giúp thư viện vận hành hiệu quả và hiện đại hơn. Một thư viện cần có một hệ thống quản lý mượn/trả sách, hệ thống đó cần lưu trữ những dữ liệu sau:

Khi vào trang web, học sinh, sinh viên, giáo viên hay thủ thư đều phải có tài khoản riêng. Thủ thư được cung cấp Username và Password có sẵn để đăng nhập. Người mượn phải đăng ký tài khoản mới, gồm Username, Password và cung cấp các thông tin cá nhân mới được phép đăng nhập.

Trong một thư viện có nhiều thủ thư. Mỗi thủ thư được cấp Tên đăng nhập và Mật khẩu để đăng nhập vào hệ thống. Mỗi thủ thư có các thông tin: Mã thủ thư, Họ tên, Ngày sinh, Địa chỉ, Điện thoại, Chức vụ. Một thủ thư có thể quản lý sách, hỗ trợ mượn/trả sách theo yêu cầu của người mượn, duyệt đăng ký tài khoản người mượn, kiểm tra lịch sử mượn.

Để phân biệt người mượn sách là sinh viên hay giáo viên thì khi lưu thông tin, trước tên của họ sẽ được đánh số phân biệt, giáo viên sẽ là 00...(họ và tên giáo viên) , sinh viên sẽ là 01..(họ và tên sinh viên).

Đối với sinh viên: mỗi học sinh khi mượn sách gồm có các thông tin cá nhân: Mã sinh viên, Họ tên, Ngày sinh, Địa chỉ, Điện thoại, Số căn cước công dân, Email. Số sách mượn tối đa: một sinh viên mượn có thể mượn 10 quyển sách. Thời gian mượn tối đa: 15 ngày.

Đối với cán bộ/giáo viên: Mỗi cán bộ/giáo viên khi mượn sách gồm có các thông tin cá nhân: Mã cán bộ/giáo viên, Họ tên, Ngày sinh, Địa chỉ, Điện thoại, Số căn cước công dân, Email. Số sách mượn tối đa: cán bộ và giáo viên có thể mượn 20 quyển sách. Thời gian mượn tối đa: 20 ngày.

Sẽ có một vài trường hợp nếu mong muốn mượn sách thêm khi đã đạt tối đã thời gian mượn thì người mượn có thể thực hiện trả sách và mượn lại hoặc có thể gặp trực tiếp thủ thư để gia hạn thời gian mượn sách.

Đối với sách, mỗi sách sẽ có thông tin như sau: Mã sách, tựa sách, tác giả, thể loại, số lượng sách đã mượn, số lượng sách còn trong thư viện, nhà xuất bản, vị trí, ngày mượn, ngày trả.

Vị trí sách: Mỗi thể loại sách được đặt trên 1 kệ ( cụ thể ), kệ sách được đánh dấu theo thứ tự bảng chữ cái A-Z.

Đối với nhà xuất bản: Mã nhà xuất bản, Tên nhà xuất bản, số điện thoại liên lạc, email, địa chỉ, số lượng sách đặt mua ( nếu có ).

Vấn đề mượn sách, trả sách, kiểm tra tình trạng sách: *Mượn sách*: Khi mượn sách offline, thủ thư chỉ cần nhập mssv và id sách khi đó hệ thống sẽ ghi nhận và xử lý thông tin. Khi mượn sách online, đối với sách vật lý, sinh viên sau khi đăng nhập sẽ chọn cuốn sách mình muốn, thông tin sinh viên và id cuốn sách sẽ được gửi về cho thủ thư. Khi đó thủ thư sẽ phản hồi lại cho sinh viên khung thời gian nhận sách. *Trả sách*: Khi trả sách, sinh viên cần đến gặp trực tiếp thủ thư. Khi đó thủ thư sẽ nhập mssv và id cuốn sách, khi đó hệ thống sẽ ghi nhận và xử lý thông tin. *Trễ hạn và phí phạt*: Nếu trễ 1 ngày thì phải trả 10 nghìn đồng, cứ thế nhân lên. Khi thủ thư *kiểm tra tình trạng sách*, sẽ có hai trường hợp xảy ra: nếu bị rách nhỏ bên ngoài thì sẽ nhắc nhở, còn nếu bị rách vẫn còn đọc được, không ảnh hưởng đến nội dung sách hoặc bị nhàu trang sách sẽ phạt 10 nghìn đồng, nếu bị mất trang, hỏng sách phải đền sách.

Cuối cùng là *Thống kê loại sách được mượn nhiều nhất* để có thể cho những bạn mới có thể nhận biết được đâu là loại sách được yêu thích hiện nay và các bạn có thể xem qua và mượn. Khi đó các bạn có thể vào phần Hỗ trợ người mượn sách để có thể nhận thêm sự trợ giúp của các thủ thư, tư vấn cho các bạn về sách và hình thức

### **1.2. Nghiệp vụ của bài toán**

Đối với hình thức offline: khi có giảng viên hoặc sinh viên mượn sách, họ chỉ cần đi đến kệ sách mình muốn hoặc tham quan thư viện để chọn ra những cuốn sách mình cần và sau đó đem sách đó lại thủ thư. Khi đó thủ thư sẽ chịu trách nhiệm lấy thông tin của người mượn vào thêm thông tin sách mượn cho người đó. Bằng cách này, đối với thủ thư và đối với người mượn sách khi đăng nhập vào tài khoản của mình, nếu muốn xem sách đã mượn thì chỉ cần gõ tên sách, thông tin của sách sẽ hiện ra.

Đối với hình thức online: Khi GV hoặc Sv mượn sách, họ phải đăng nhập tài khoản của mình để vào trang thư viện online. Để mượn sách họ chỉ cần vào lựa sách mình muốn đọc hoặc để dễ dàng tìm kiếm, chỉ cần gõ tên tựa sách thì hệ thống sẽ hiện thông tin của sách và tình trạng còn hay hết sách. Còn nếu trường hợp không nhớ tựa nhưng nhớ tên tác giả thì chỉ cần tìm kiếm tên tác giả, sau đó hệ thống sẽ lọc ra những sách của tác giả đó có trong thư viện. Sau khi chọn sách xong thì chỉ cần chọn mượn sách, khi đó giống như ta đã đặt lịch hẹn mượn sách. Sau đó chúng ta đến gặp trực tiếp thủ thư và đọc thông tin của mình để thủ thư kiểm tra, nếu trùng khớp thông tin đặt thì họ sẽ lấy sách cho mình mượn.

Khi đọc xong và trả sách, thủ thư sẽ kiểm tra sách, và phân tích  tình hình cuốn sách xem có bị hư hại hay có bị ảnh hưởng gì không, nếu có thì sẽ dựa vào mức độ để phạt. Còn nếu không có gì thì sau khi trả sách, thủ thư sẽ vào hệ thống và delete sách đã trả lại ở thông tin người mượn rằng bạn này đã trả sách ( cụ thể tên tựa sách)

### **1.3. Mô tả các chức năng của bài toán**

*a) Chức Năng Dành Cho người mượn (Teacher/Student)*

*Quản lý thông tin cá nhân:* Người dùng có thể sử dụng thông tin cá nhân để đăng ký tài khoản sử dụng trang sách online ở tại thủ thư.Sau khi đăng ký tài khoản, người dùng có thể bắt đầu sử dụng trang sách online. Khi vào trang sách họ có thể *Đăng nhập/ đăng xuất ,* chophép người dùng đăng nhập để sử dụng hệ thống. Ngoài ra họ còn có thể quản lý thông tin cá nhân của chính mình như cập nhật thông tin email, số điện thoại.

*Quản Lý Mượn Trả Sách:* Người dùng có thể tìm kiếm, tra cứu sách theo tiêu đề, tác giả, thể loại, tên tác giả, nxb,... trong phần sách. Sau khi tìm thấy sách mình mong muốn, họ có thể gửi yêu cầu mượn sách nếu sách còn sẵn có. Ngoài ra họ còn có thể kiểm tra trạng thái mượn sách của bản thân để xem yêu cầu mượn sách đã được duyệt hay chưa và xem những sách mình đang mượn và còn có thể xem lại lịch sử mượn/trả sách. Nếu gần ngày đến hạn trả sách 3 ngày, người dùng sẽ nhận thông báo cho việc trả sách đúng hạn.

Khi cần sự hỗ trợ , người dùng có thể vào phần hỗ trợ và miêu tả khó khăn của mình trong việc mượn/trả sách hoặc xem sách, sau đó thủ thư sẽ xem và gửi phản hồi hỗ trợ. Ngoài ra còn có thể giúp tư vấn chọn sách dựa trên lịch sử mượn và sở thích cá nhân. Và có thể xem hệ thống đánh giá và những bình luận về sách, xem sách có lượt mượn cáo nhất trong tuần, tháng qua.

*b) Chức Năng Dành Cho Thủ thư-Quản lý sách(employee)*

Thủ thư trong hệ thống thư viện có nhiệm vụ quản lý toàn bộ danh mục sách nhằm đảm bảo việc tra cứu và mượn sách diễn ra thuận lợi. Họ có quyền thêm mới, chỉnh sửa hoặc xóa thông tin sách khỏi hệ thống khi có sự thay đổi, giúp cập nhật kịp thời dữ liệu thư viện. Bên cạnh đó, thủ thư còn theo dõi và cập nhật tình trạng sách, ghi nhận các trường hợp sách bị hư hỏng, mất mát hoặc cần bảo trì để có biện pháp xử lý phù hợp. Ngoài ra, họ cũng chịu trách nhiệm kiểm tra số lượng sách tồn kho, đảm bảo thư viện luôn có đủ tài liệu phục vụ người đọc, đồng thời đề xuất bổ sung sách mới khi cần thiết.

Quản Lý Mượn Trả: Thủ thư sẽ xem xét duyệt hoặc từ chối yêu cầu mượn sách của người dùng. Khi trả sách, thủ thư sẽ có trách nhiệm kiểm tra tình trạng sách khi trả: Xác nhận sách có bị hư hỏng hay không hư nhẹ( rách nhỏ, dính mực vô sách) hay hư nặng( rách trang, mất trang, sách bị ngập nước).

Tính phí phạt (nếu có): Áp dụng phí phạt khi sách bị trả trễ hoặc hư hỏng tùy theo mức độ  
 Quản Lý Hệ Thống: Thủ thư còn có thể quản lý một vài chức năng của hệ thống, như: Quản lý tài khoản người dùng: Tạo, sửa, thêm, xóa tài khoản khách hàng, Quản lý phí phạt: Xác định mức phạt đối với sách trả trễ hoặc hư hỏng. Và cuối cùng là xuất báo cáo thống kê: Tổng hợp dữ liệu về số sách mượn, trả, quá hạn theo từng thời gian.

*c) Chức năng dành cho Admin*

Admin có toàn quyền xử dụng hệ thống và có thể thực hiện tất cả các chức năng của khách hàng và thủ thư. Ngoài ra Admin còn có thể xem xét duyệt các loại sách từ nhà xuất bản, ví dụ khi nhập sách từ một nhà xuất bản nào đó hoặc khi một sách của nxb nào đó không phát hành hoặc cấm sử dụng thì admin có thể xem xét duyệt, cập nhật sách cho thư viện

·        Hỗ trợ tìm kiếm sách theo tác giả, nhà xuất bản, ngôn ngữ và thể loại.

### **1.4. Giao diện**

Trước khi đăng nhập, sẽ cho phép chọn 3 mục: Admin-Thủ thư, Giảng viên,Siinh viên

Người dùng cần chọn mục phù hợp với mục đích đăng nhập.

*Đối với cấp độ người dùng (SV/GV):*

       Khi đăng nhập được vào trang hệ thống. Người mượn có thể nhập thông tin sách  như tựa, nxb, năm sx,thể loại,.... ở thanh thông tin tìm kiếm.Sau khi nhập xong, nếu đúng thông tin sẽ hiện ra sách đó, còn nếu không sẽ hiện ra những sách có độ tương đồng, ví dụ như những sách có từ khóa tương đồng, sau đó sẽ chọn sách để mượn  
      Để xem sách mình đã mượn và thời hạn mượn, người mượn có thể vào phần xem thông tin mượn sách để xem thông tin sách và hạn mượn còn bao nhiêu ngày.

*Đối với cấp độ thủ thư*

Khi đăng nhập vào trang hệ thống. Sẽ có các mục hiện ra, Thống kê, Add book, issure book, view, User borrow book, delete book. User-borrow book  là để xem thông tin người mượn sách và sách đang được mượn, View là để xem thông tin sách thư viện, Add book là để thêm sách mỗi khi có sách mới về thư viện, Delete book để xóa những sách đã hết trong thư viện, cụ thể là không được cho phép sử dụng từ tác giả hoặc nxb.

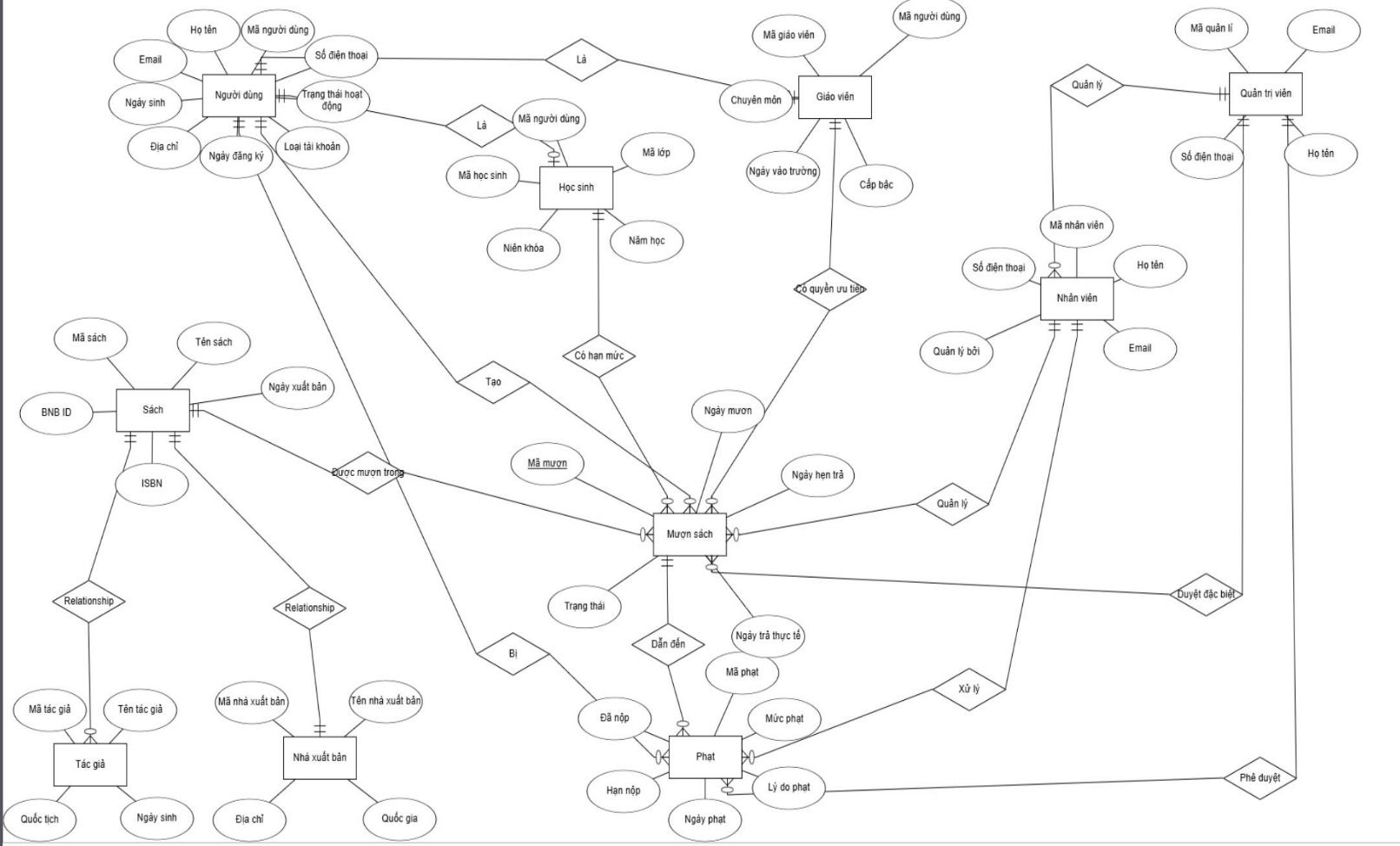
### **1.5. Phương pháp phạt khi trễ hạn trả sách**

Đối với các lỗi quá hạn: thì cứ qua một ngày mới sẽ phạt thêm 10.000vnd

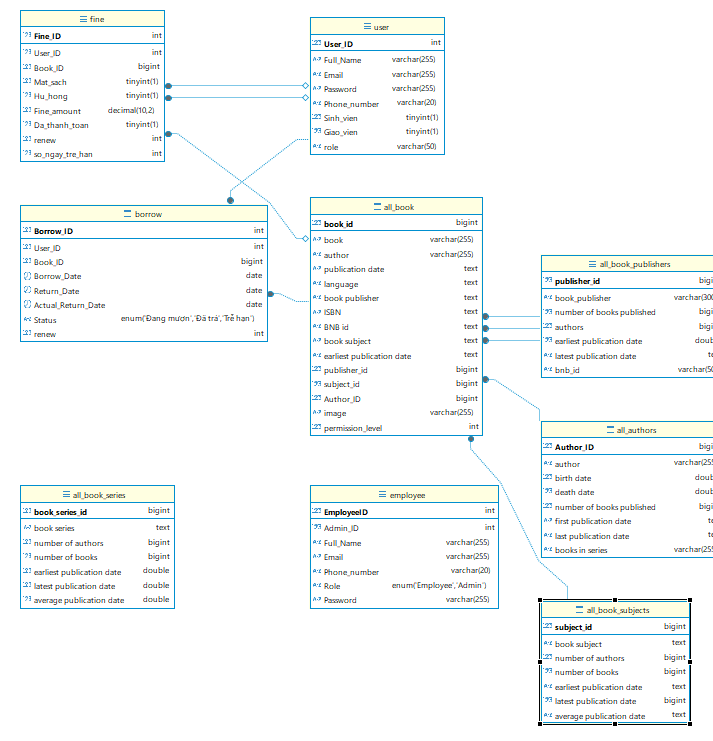
Đối với các sách bị hư họng sẽ xem xét theo mức độ thiệt hại.Nếu sách bị rách nhẹ nhưng không ảnh hưởng đến ngoại hình và nội dung bên trong thì sẽ nhắc nhở.Nếu sách bị rách hoặc nhàu trang, nếu số lượng không đáng kể nhưng vẫn ảnh hưởng đến chữ đọc bên trong thì sẽ bị phạt 10.000-20.000 vnd.Nếu bị mất trang sách hoặc mất sách thì phải đền lại sách mới.Đối với các lỗi quá hạn: Cứ qua một ngày mới sẽ phạt thêm 10.000vnd

# **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 bài toán, hình thành được sơ đồ thực thể kết hợp (ERD).

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



1. **. Các ràng buộc cần có**

|  |  |  |
| --- | --- | --- |
| **STT** | **TABLE** | **RÀNG BUỘC** |
| **1** | user | Ràng buộc khóa chính user\_ID |
| **2** | employee | Ràng buộc khóa chính employeeID |
| **3** | All\_book | Ràng buộc khóa chính book\_id,  -Khóa ngoại fk\_author: Ràng buộc này đảm bảo rằng Author\_ID trong bảng all\_book sẽ phải tham chiếu tới cột Author\_ID trong bảng all\_authors.  -Khóa ngoại fk\_book\_publisher: Ràng buộc này đảm bảo rằng publisher\_id trong bảng all\_book sẽ phải tham chiếu tới cột publisher\_id trong bảng all\_book\_publishers.  -Khóa ngoại fk\_subject: Ràng buộc này đảm bảo rằng subject\_id trong bảng all\_book sẽ phải tham chiếu tới cột subject\_id trong bảng all\_book\_subjects |
| **4** | All\_book\_publisher | Ràng buộc khóa chính publisher\_id |
| **5** | All\_author | Ràng buộc khóa chính Author\_ID |
| **6** | fine | -Ràng buộc khóa chính FineID  -Ràng buộc khóa ngoại User\_ID tham chiếu đến USER  -Ràng buộc khóa ngoại Book\_ID tham chiếu đến all\_book  -Ràng buộc kiểm tra giá trị đảm bảo các giá trị trong cột Da\_thanh\_toan và Fine\_amount là hợp lệ và không vi phạm các quy tắc dữ liệu. |
| **7** | All\_Book\_Series | -Ràng buộc khóa chính (book\_series\_id) |
| **8** | All\_book\_subjects | -Ràng buộc khóa chính (subject\_id), ràng buộc khóa ngoại subject\_id tham chiếu đến all\_book |
| **9** | Borrow | - Ràng buộc khóa chính (borrow\_id) |

## **4.Cài đặt CSDL**

### **4.1. Tạo bảng và các ràng buộc**

**Bảng user**

CREATE TABLE `user` (

`User\_ID` int NOT NULL AUTO\_INCREMENT,

`Full\_Name` varchar(255) NOT NULL,

`Email` varchar(255) NOT NULL,

`Password` varchar(255) NOT NULL,

`Phone\_number` varchar(20) DEFAULT NULL,

`Sinh\_vien` tinyint(1) DEFAULT '0',

`Giao\_vien` tinyint(1) DEFAULT '0',

`role` varchar(50) DEFAULT 'user',

PRIMARY KEY (`User\_ID`),

UNIQUE KEY `Email` (`Email`),

UNIQUE KEY `unique\_user\_email` (`Email`),

UNIQUE KEY `unique\_user\_phone` (`Phone\_number`),

CONSTRAINT `chk\_user\_type` CHECK ((((`Sinh\_vien` = 1) and (`Giao\_vien` = 0)) or ((`Sinh\_vien` = 0) and (`Giao\_vien` = 1)) or ((`Sinh\_vien` = 0) and (`Giao\_vien` = 0))))

) ENGINE=InnoDB AUTO\_INCREMENT=24126137 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_0900\_ai\_ci;

**Bảng Employee**

CREATE TABLE `employee` (

`EmployeeID` int NOT NULL AUTO\_INCREMENT,

`Admin\_ID` int DEFAULT NULL,

`Full\_Name` varchar(255) NOT NULL,

`Email` varchar(255) NOT NULL,

`Phone\_number` varchar(20) DEFAULT NULL,

`Role` enum('Employee','Admin') NOT NULL DEFAULT 'Employee',

`Password` varchar(255) NOT NULL,

PRIMARY KEY (`EmployeeID`),

UNIQUE KEY `Email` (`Email`),

KEY `fk\_employee\_admin` (`Admin\_ID`)

) ENGINE=InnoDB AUTO\_INCREMENT=4 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_0900\_ai\_ci;

**Bảng all\_book**

CREATE TABLE `all\_book` (

`book` varchar(255) NOT NULL,

`author` varchar(255) DEFAULT NULL,

`publication date` text,

`language` text,

`book publisher` text,

`ISBN` text,

`BNB id` text,

`book subject` text,

`earliest publication date` text,

`book\_id` bigint NOT NULL AUTO\_INCREMENT,

`publisher\_id` bigint NOT NULL,

`subject\_id` bigint NOT NULL,

`Author\_ID` bigint NOT NULL,

`image` varchar(255) DEFAULT '',

PRIMARY KEY (`book\_id`),

KEY `idx\_author` (`author`),

KEY `fk\_subject` (`subject\_id`),

KEY `fk\_author` (`Author\_ID`),

KEY `fk\_publisher` (`publisher\_id`),

CONSTRAINT `fk\_author` FOREIGN KEY (`Author\_ID`) REFERENCES `all\_authors` (`Author\_ID`),

CONSTRAINT `fk\_book\_publisher` FOREIGN KEY (`publisher\_id`) REFERENCES `all\_book\_publishers` (`publisher\_id`),

CONSTRAINT `fk\_publisher` FOREIGN KEY (`publisher\_id`) REFERENCES `all\_book\_publishers` (`publisher\_id`),

CONSTRAINT `fk\_publisher\_id` FOREIGN KEY (`publisher\_id`) REFERENCES `all\_book\_publishers` (`publisher\_id`),

CONSTRAINT `fk\_subject` FOREIGN KEY (`subject\_id`) REFERENCES `all\_book\_subjects` (`subject\_id`)

) ENGINE=InnoDB AUTO\_INCREMENT=104 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_0900\_ai\_ci;

**Bảng all\_book\_series**

CREATE TABLE `all\_book\_series` (

`book series` text,

`number of authors` bigint DEFAULT NULL,

`number of books` bigint DEFAULT NULL,

`earliest publication date` double DEFAULT NULL,

`latest publication date` double DEFAULT NULL,

`average publication date` double DEFAULT NULL,

`book\_series\_id` bigint NOT NULL AUTO\_INCREMENT,

PRIMARY KEY (`book\_series\_id`)

) ENGINE=InnoDB AUTO\_INCREMENT=176915 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_0900\_ai\_ci**;**

**Bảng all\_book\_subjects**

CREATE TABLE `all\_book\_subjects` (

`book subject` text,

`number of authors` bigint DEFAULT NULL,

`number of books` bigint DEFAULT NULL,

`earliest publication date` text,

`latest publication date` bigint DEFAULT NULL,

`average publication date` text,

`subject\_id` bigint NOT NULL AUTO\_INCREMENT,

PRIMARY KEY (`subject\_id`)

) ENGINE=InnoDB AUTO\_INCREMENT=586326 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_0900\_ai\_ci;

**Bảng all\_book\_publishers**

CREATE TABLE `all\_book\_publishers` (

`book\_publisher` varchar(300) DEFAULT NULL,

`number of books published` bigint DEFAULT NULL,

`authors` bigint DEFAULT NULL,

`earliest publication date` double DEFAULT NULL,

`latest publication date` text,

`publisher\_id` bigint NOT NULL AUTO\_INCREMENT,

`bnb\_id` varchar(50) DEFAULT NULL,

PRIMARY KEY (`publisher\_id`),

UNIQUE KEY `bnb\_id` (`bnb\_id`)

) ENGINE=InnoDB AUTO\_INCREMENT=3516031 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_0900\_ai\_ci;

**Bảng all\_authors**

CREATE TABLE `all\_authors` (

`author` varchar(255) DEFAULT NULL,

`birth date` double DEFAULT NULL,

`death date` double DEFAULT NULL,

`number of books published` bigint DEFAULT NULL,

`first publication date` text,

`last publication date` text,

`books in series` varchar(255) DEFAULT NULL,

`Author\_ID` bigint NOT NULL,

PRIMARY KEY (`Author\_ID`),

KEY `idx\_author\_works` (`author`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_0900\_ai\_ci;

**Bảng borrow**

CREATE TABLE `borrow` (

`Borrow\_ID` int NOT NULL AUTO\_INCREMENT,

`User\_ID` int NOT NULL,

`Book\_ID` bigint NOT NULL,

`Borrow\_Date` date DEFAULT NULL,

`Return\_Date` date DEFAULT NULL,

`Actual\_Return\_Date` date DEFAULT NULL,

`Status` enum('Đang mượn','Đã trả','Trễ hạn') DEFAULT NULL,

`Fine\_ID` int DEFAULT NULL,

`renew` int DEFAULT '0',

PRIMARY KEY (`Borrow\_ID`),

KEY `Book\_ID` (`Book\_ID`),

KEY `borrow\_ibfk\_1` (`User\_ID`),

CONSTRAINT `borrow\_ibfk\_1` FOREIGN KEY (`User\_ID`) REFERENCES `user` (`User\_ID`) ON DELETE CASCADE,

CONSTRAINT `borrow\_ibfk\_2` FOREIGN KEY (`Book\_ID`) REFERENCES `all\_book` (`book\_id`),

CONSTRAINT `check\_borrow\_date` CHECK ((`Borrow\_Date` <= `Return\_Date`))

) ENGINE=InnoDB AUTO\_INCREMENT=12 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_0900\_ai\_ci;

**Bảng fine**

CREATE TABLE `fine` (

`Fine\_ID` int NOT NULL AUTO\_INCREMENT,

`User\_ID` int DEFAULT NULL,

`Book\_ID` bigint DEFAULT NULL,

`Mat\_sach` tinyint(1) DEFAULT '0',

`Hu\_hong` tinyint(1) DEFAULT '0',

`Fine\_amount` decimal(10,2) DEFAULT NULL,

`Da\_thanh\_toan` tinyint(1) DEFAULT '0',

`so\_ngay\_tre\_han` int DEFAULT NULL,

PRIMARY KEY (`Fine\_ID`),

KEY `fk\_fine\_user` (`User\_ID`),

KEY `fk\_fine\_book` (`Book\_ID`),

CONSTRAINT `fine\_ibfk\_1` FOREIGN KEY (`User\_ID`) REFERENCES `user` (`User\_ID`) ON DELETE CASCADE,

CONSTRAINT `fine\_ibfk\_2` FOREIGN KEY (`Book\_ID`) REFERENCES `all\_book` (`book\_id`) ON DELETE CASCADE,

CONSTRAINT `fk\_fine\_book` FOREIGN KEY (`Book\_ID`) REFERENCES `all\_book` (`book\_id`) ON DELETE CASCADE ON UPDATE CASCADE,

CONSTRAINT `fk\_fine\_user` FOREIGN KEY (`User\_ID`) REFERENCES `user` (`User\_ID`) ON DELETE CASCADE ON UPDATE CASCADE,

CONSTRAINT `check\_da\_thanh\_toan` CHECK ((`Da\_thanh\_toan` in (0,1))),

CONSTRAINT `check\_fine\_amount` CHECK ((`Fine\_amount` >= 0))

) ENGINE=InnoDB AUTO\_INCREMENT=21 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_0900\_ai\_ci;

### **4.2. Trigger kiểm tra các ràng buộc**

**Trigger tự động thêm dòng vào fine mỗi khi người dung mượn sách**

**CREATE** **DEFINER**=**`root`**@**`%`** **TRIGGER** **`after\_borrow\_insert`** **AFTER** **INSERT** **ON** **`borrow`** **FOR** **EACH** **ROW** **BEGIN**

**INSERT** **INTO** fine (Fine\_ID, User\_ID, Book\_ID, Fine\_amount, Da\_thanh\_toan, so\_ngay\_tre\_han)

**VALUES** (**NEW**.Borrow\_ID, **NEW**.User\_ID, **NEW**.Book\_ID, 0, 0, 0);

**END**

**Trigger tự động cập nhật trạng thái "Đã trả" cho phiếu mượn khi người dùng nhập ngày trả sách.**

**CREATE** **DEFINER**=**`root`**@**`%`** **TRIGGER** **`trg\_set\_status\_returned`** **BEFORE** **UPDATE** **ON** **`borrow`** **FOR** **EACH** **ROW** **BEGIN**

**IF** **NEW**.Actual\_Return\_Date **IS** **NOT** **NULL** **AND** **OLD**.Actual\_Return\_Date **IS** **NULL** **THEN**

**SET** **NEW**.Status = 'Đã trả';

**END** **IF**;

**END**

# **CHƯƠNG 3: THIẾT KẾ CÁC CHỨC NĂNG**

* 1. **Kết nối cơ sở dữ liệu**

**const mysql = require("mysql2");**

**require('dotenv').config();**

**// Cấu hình kết nối MySQL (sử dụng pool để quản lý kết nối)**

**const pool = mysql.createPool({**

**host: process.env.MYSQLHOST,**

**user: process.env.MYSQLUSER,**

**password: process.env.MYSQL\_ROOT\_PASSWORD,**

**database: process.env.MYSQL\_DATABASE,**

**port: process.env.MYSQLPORT**

**});**

**// Sử dụng Promise API**

**const db = pool.promise(); // Đảm bảo db trả về một promise**

**// Kiểm tra kết nối**

**async function testConnection() {**

**try {**

**const [rows, fields] = await db.query("SELECT 1"); // Chỉ cần thực hiện một query đơn giản để kiểm tra kết nối**

**console.log("Đã kết nối MySQL");**

**} catch (err) {**

**console.error("Lỗi kết nối MySQL:", err);**

**}**

**}**

**testConnection();**

**module.exports = db;**

* 1. **Chức năng tạo, đăng ký tài khoản**

### **1.Procedure setup ban đầu để tạo bảng lưu thông tin tài khoản.**

**CREATE** **DEFINER**=**`root`**@**`%`** **PROCEDURE** **`library\_management`**.**`account\_registration`**()

**BEGIN**

**CREATE** **TABLE** **IF** **NOT** **EXISTS** account\_registration (

registration\_id **INT** **AUTO\_INCREMENT** **PRIMARY** **KEY**,

username **VARCHAR**(50) **UNIQUE** **NOT** **NULL**,

password\_hash **VARCHAR**(255) **NOT** **NULL**,

full\_name **VARCHAR**(100) **NOT** **NULL**,

email **VARCHAR**(100) **UNIQUE** **NOT** **NULL**,

account\_type **VARCHAR**(20) **CHECK** (account\_type **IN** ('admin', 'employee', 'user')),

registration\_date **TIMESTAMP** **DEFAULT** **CURRENT\_TIMESTAMP**,

approval\_status **VARCHAR**(20) **DEFAULT** 'pending' **CHECK** (approval\_status **IN** ('pending', 'approved', 'rejected')),

approved\_by **VARCHAR**(50) **NULL**,

approval\_date **TIMESTAMP** **NULL**

);

**END**

**2.Đăng ký tài khoản người dùng mới** vào hệ thống thư viện

**CREATE** **DEFINER**=**`root`**@**`%`** **PROCEDURE** **`library\_management`**.**`register\_new\_account`**(

**IN** p\_username **VARCHAR**(50),

**IN** p\_password **VARCHAR**(255),

**IN** p\_full\_name **VARCHAR**(100),

**IN** p\_email **VARCHAR**(100),

**IN** p\_account\_type **VARCHAR**(20)

)

**BEGIN**

**DECLARE** user\_exists **INT**;

**DECLARE** email\_exists **INT**;

-- Kiểm tra tên người dùng đã tồn tại chưa

**SELECT** **COUNT**(\*) **INTO** user\_exists **FROM** account\_registration **WHERE** username = p\_username;

**IF** user\_exists > 0 **THEN**

**SIGNAL** **SQLSTATE** '45000' **SET** MESSAGE\_TEXT = 'Tên đăng nhập đã tồn tại';

**END** **IF**;

-- Kiểm tra email đã tồn tại chưa

**SELECT** **COUNT**(\*) **INTO** email\_exists **FROM** account\_registration **WHERE** email = p\_email;

**IF** email\_exists > 0 **THEN**

**SIGNAL** **SQLSTATE** '45000' **SET** MESSAGE\_TEXT = 'Email đã được sử dụng';

**END** **IF**;

-- Hash mật khẩu trước khi lưu vào database

**INSERT** **INTO** account\_registration (username, password\_hash, full\_name, email, account\_type)

**VALUES** (p\_username, SHA2(p\_password, 256), p\_full\_name, p\_email, p\_account\_type);

-- Thông báo đăng ký thành công, chờ phê duyệt

**SELECT** 'Đăng ký tài khoản thành công. Vui lòng chờ quản trị viên phê duyệt.' **AS** *message*;

**END**

**3.Tạo tài khoản người dùng**

**CREATE** **DEFINER**=**`root`**@**`%`** **PROCEDURE** **`library\_management`**.**`CreateUserAccount`**(

**IN** p\_UserID **VARCHAR**(50),

**IN** p\_FullName **VARCHAR**(255),

**IN** p\_Email **VARCHAR**(255),

**IN** p\_HashedPassword **VARCHAR**(255),

**IN** p\_Phone **VARCHAR**(20),

**IN** p\_SinhVien **TINYINT**,

**IN** p\_GiaoVien **TINYINT**

)

**BEGIN**

**INSERT** **INTO** **user** (User\_ID, Full\_Name, Email, **Password**, Phone\_number, Sinh\_vien, Giao\_vien)

**VALUES** (p\_UserID, p\_FullName, p\_Email, p\_HashedPassword, p\_Phone, p\_SinhVien, p\_GiaoVien);

**END**

* 1. **Phê duyệt / từ chối tài khoản**
  2. **Đăng nhập, đổi mật khẩu**

**1.Cập nhật mật khẩu của người dùng**

**CREATE** **DEFINER**=**`root`**@**`%`** **PROCEDURE** **`library\_management`**.**`change\_user\_password`**(

**IN** p\_UserID **VARCHAR**(50),

**IN** p\_NewHashedPassword **VARCHAR**(255)

)

**BEGIN**

**UPDATE** **user**

**SET** **password** = p\_NewHashedPassword

**WHERE** User\_ID = p\_UserID;

**END**

* 1. **Quản lý nhân viên, người dùng**

**1.Thêm một nhân viên mới**

**CREATE** **DEFINER**=**`root`**@**`%`** **PROCEDURE** **`library\_management`**.**`add\_employee`**(

**IN** p\_admin\_id **INT**,

**IN** p\_full\_name **VARCHAR**(255),

**IN** p\_email **VARCHAR**(255),

**IN** p\_phone\_number **VARCHAR**(28),

**IN** p\_role **ENUM**('Employee', 'Admin'),

**IN** p\_password **VARCHAR**(255)

)

**BEGIN**

**DECLARE** existing\_email **INT**;

-- Kiểm tra email trùng lặp

**SELECT** **COUNT**(\*) **INTO** existing\_email

**FROM** employee

**WHERE** Email = p\_email;

**IF** existing\_email > 0 **THEN**

**SIGNAL** **SQLSTATE** '45000'

**SET** MESSAGE\_TEXT = 'Email đã tồn tại!';

**END** **IF**;

-- Chèn dữ liệu vào bảng employee

**INSERT** **INTO** employee (

Admin\_ID,

Full\_Name,

Email,

Phone\_number,

Role,

Password

)

**VALUES** (

p\_admin\_id,

p\_full\_name,

p\_email,

p\_phone\_number,

p\_role,

p\_password

);

**END**

**3.** **Lấy thông tin tất cả nhân viên**

**CREATE** **DEFINER**=**`root`**@**`%`** **PROCEDURE** **`library\_management`**.**`get\_employees`**()

**BEGIN**

**SELECT** \* **FROM** employee;

**END**

**4.** **Thêm một người dùng mới**

**CREATE** **DEFINER**=**`root`**@**`%`** **PROCEDURE** **`library\_management`**.**`add\_user`**(

**IN** p\_user\_id **VARCHAR**(50),

**IN** p\_full\_name **VARCHAR**(255),

**IN** p\_email **VARCHAR**(255),

**IN** p\_password **VARCHAR**(255),

**IN** p\_phone\_number **VARCHAR**(20),

**IN** p\_sinh\_vien **TINYINT**,

**IN** p\_giao\_vien **TINYINT**

)

**BEGIN**

**DECLARE** existing\_user\_id **INT**;

**DECLARE** existing\_email **INT**;

**DECLARE** existing\_phone **INT**;

-- Kiểm tra User\_ID trùng lặp

**SELECT** **COUNT**(\*) **INTO** existing\_user\_id

**FROM** user

**WHERE** User\_ID = p\_user\_id;

**IF** existing\_user\_id > 0 **THEN**

**SIGNAL** **SQLSTATE** '45000'

**SET** MESSAGE\_TEXT = 'Mã người dùng đã tồn tại!';

**END** **IF**;

-- Kiểm tra email trùng lặp

**SELECT** **COUNT**(\*) **INTO** existing\_email

**FROM** user

**WHERE** Email = p\_email;

**IF** existing\_email > 0 **THEN**

**SIGNAL** **SQLSTATE** '45000'

**SET** MESSAGE\_TEXT = 'Email đã tồn tại!';

**END** **IF**;

-- Kiểm tra số điện thoại trùng lặp

**SELECT** **COUNT**(\*) **INTO** existing\_phone

**FROM** **user**

**WHERE** Phone\_number = p\_phone\_number;

**IF** existing\_phone > 0 **THEN**

**SIGNAL** **SQLSTATE** '45000'

**SET** MESSAGE\_TEXT = 'Số điện thoại đã tồn tại!';

**END** **IF**;

-- Kiểm tra vai trò: chỉ được chọn một vai trò (Sinh\_vien hoặc Giao\_vien)

**IF** (p\_sinh\_vien = 1 **AND** p\_giao\_vien = 1) **THEN**

**SIGNAL** **SQLSTATE** '45000'

**SET** MESSAGE\_TEXT = 'Chỉ được chọn một vai trò: Sinh viên hoặc Giáo viên!';

**END** **IF**;

**IF** (p\_sinh\_vien = 0 **AND** p\_giao\_vien = 0) **THEN**

**SIGNAL** **SQLSTATE** '45000'

**SET** MESSAGE\_TEXT = 'Phải chọn một vai trò: Sinh viên hoặc Giáo viên!';

**END** **IF**;

-- Chèn dữ liệu vào bảng user

**INSERT** **INTO** **user** (

User\_ID,

Full\_Name,

Email,

**Password**,

Phone\_number,

Sinh\_vien,

Giao\_vien

)

**VALUES** (

p\_user\_id,

p\_full\_name,

p\_email,

p\_password,

p\_phone\_number,

p\_sinh\_vien,

p\_giao\_vien

);

**END**

**6.Lấy thông tin người dùng**

**CREATE** **DEFINER**=**`root`**@**`%`** **PROCEDURE** **`library\_management`**.**`get\_users`**()

**BEGIN**

**SELECT** \* **FROM** **user**;

**END**

**7.Tìm kiếm và trả về thông tin chi tiết của User**

**CREATE** **DEFINER**=**`root`**@**`%`** **PROCEDURE** **`library\_management`**.**`GetUserInfo`**(

**IN** p\_UserID **VARCHAR**(50)

)

**BEGIN**

**SELECT**

User\_ID,

Full\_Name,

Email,

Phone\_number,

Sinh\_vien,

Giao\_vien

**FROM** **user**

**WHERE** User\_ID = p\_UserID;

**END**

**8.** **Cập nhật thông tin người dùng**

**CREATE** **DEFINER**=**`root`**@**`%`** **PROCEDURE** **`library\_management`**.**`update\_user\_info`**(

**IN** p\_UserID **VARCHAR**(50),

**IN** p\_FullName **VARCHAR**(255),

**IN** p\_Email **VARCHAR**(255),

**IN** p\_PhoneNumber **VARCHAR**(20)

)

**BEGIN**

**UPDATE** **user**

**SET**

Full\_Name = p\_FullName,

Email = p\_Email,

Phone\_number = p\_PhoneNumber

**WHERE** User\_ID = p\_UserID;

**END**

* 1. **Quản lý tác giả, nhà xuất bản, chủ đề**

**1.Thêm tác giả**

**CREATE** **DEFINER**=**`root`**@**`%`** **PROCEDURE** **`library\_management`**.**`add\_author`**( **IN** p\_author\_id **BIGINT**,

**IN** p\_author **VARCHAR**(255),

**IN** p\_birth\_date **DOUBLE**,

**IN** p\_death\_date **DOUBLE**,

**IN** p\_number\_of\_books **BIGINT**,

**IN** p\_first\_publication\_date **TEXT**,

**IN** p\_last\_publication\_date **TEXT**,

**IN** p\_books\_in\_series **VARCHAR**(255))

**BEGIN**

-- Chèn dữ liệu vào bảng all\_book\_subjects

**DECLARE** existing\_author\_id **INT**;

-- Kiểm tra Author\_ID trùng lặp

**SELECT** **COUNT**(\*) **INTO** existing\_author\_id

**FROM** all\_authors

**WHERE** Author\_ID = p\_author\_id;

**IF** existing\_author\_id > 0 **THEN**

**SIGNAL** **SQLSTATE** '45000'

**SET** MESSAGE\_TEXT = 'Mã tác giả đã tồn tại!';

**END** **IF**;

-- Chèn dữ liệu vào bảng all\_authors

**INSERT** **INTO** all\_authors (

Author\_ID,

author,

`birth date`,

`death date`,

`number of books published`,

`first publication date`,

`last publication date`,

`books in series`

)

**VALUES** (

p\_author\_id,

p\_author,

p\_birth\_date,

p\_death\_date,

p\_number\_of\_books,

p\_first\_publication\_date,

p\_last\_publication\_date,

p\_books\_in\_series

);

**END**

**2.Tìm kiếm và trả về thông tin của một tác giả**

**CREATE** **DEFINER**=**`root`**@**`%`** **PROCEDURE** **`library\_management`**.**`GetAuthor`**(**IN** inputAuthorId **INT**)

**BEGIN**

**SELECT** \* **FROM** all\_authors **WHERE** author\_id = inputAuthorId;

**END**

**3.Tìm kiếm tác giả dựa theo tên**

**CREATE** **DEFINER**=**`root`**@**`%`** **PROCEDURE** **`library\_management`**.**`SearchAuthorBeLike`**(

**IN** p\_name **VARCHAR**(255)

)

**BEGIN**

-- Tìm các tác giả có tên giống hoặc chứa tên tìm kiếm

**SELECT** \*

**FROM** all\_authors

**WHERE** **LOWER**(author) **LIKE** **LOWER**(**CONCAT**('%', p\_name, '%'));

**END**

**4.** **Thêm thông tin về một nhà xuất bản**

**CREATE** **DEFINER**=**`root`**@**`%`** **PROCEDURE** **`library\_management`**.**`add\_publisher`**(

**IN** p\_book\_publisher **VARCHAR**(300),

**IN** p\_number\_of\_books\_published **BIGINT**,

**IN** p\_earliest\_publication\_date **DOUBLE**,

**IN** p\_latest\_publication\_date **TEXT**)

**BEGIN**

-- Chèn dữ liệu vào bảng all\_book\_publishers

**INSERT** **INTO** all\_book\_publishers (

book\_publisher,

**`number of books published`**,

**`earliest publication date`**,

**`latest publication date`**

)

**VALUES** (

p\_book\_publisher,

p\_number\_of\_books\_published,

p\_earliest\_publication\_date,

p\_latest\_publication\_date

);

**END**

**5.Lấy thông tin chi tiết của một nhà xuất bản**

**CREATE** **DEFINER**=**`root`**@**`localhost`** **PROCEDURE** **`library\_management`**.**`GetPublisherById`**(**IN** input\_publisher\_id **INT**)

**BEGIN**

**SELECT** \* **FROM** all\_book\_publishers

**WHERE** publisher\_id = input\_publisher\_id;

**END**

**6.Tìm kiếm nhà xuất bản trong bảng all\_book\_publishers**

**CREATE** **DEFINER**=**`root`**@**`%`** **PROCEDURE** **`library\_management`**.**`SearchBookPublisherBeLike`**(**IN** inputName **VARCHAR**(255))

**BEGIN**

**SELECT** publisher\_id, book\_publihser

**FROM** all\_book\_publishers

**WHERE** book\_publihser **LIKE** **CONCAT**('%', inputName, '%');

**END**

**7.Thêm thông tin về một chủ đề sách**

**CREATE** **DEFINER**=**`root`**@**`%`** **PROCEDURE** **`library\_management`**.**`add\_subject`**(

**IN** p\_book\_subject **TEXT**,

**IN** p\_number\_of\_authors **BIGINT**,

**IN** p\_number\_of\_books **BIGINT**

)

**BEGIN**

**INSERT** **INTO** all\_book\_subjects (

**`book subject`**,

**`number of authors`**,

**`number of books`**

)

**VALUES** (

p\_book\_subject,

p\_number\_of\_authors,

p\_number\_of\_books

);

**END**

**8.** **Lấy thông tin chi tiết của một thể loại sách**

**CREATE** **DEFINER**=**`root`**@**`%`** **PROCEDURE** **`library\_management`**.**`GetSubject`**(**IN** subjectId **INT**)

**BEGIN**

**SELECT** subject\_id, **`book subject`**

**FROM** all\_book\_subjects

**WHERE** subject\_id = subjectId;

**END**

**9.Tìm kiếm thể loại/sách theo tên gần đúng**

**CREATE** **DEFINER**=**`root`**@**`%`** **PROCEDURE** **`library\_management`**.**`SearchSubjectBeLike`**(**IN** inputName **VARCHAR**(255))

**BEGIN**

**SELECT** subject\_id, **`book subject`**

**FROM** all\_book\_subjects

**WHERE** **`book subject`** **LIKE** **CONCAT**('%', inputName, '%');

**END**

* 1. **Quản lý sách**

**1.** **Thêm thông tin về một cuốn sách mới**

**CREATE** **DEFINER**=**`root`**@**`%`** **PROCEDURE** **`library\_management`**.**`add\_book`**(

**IN** p\_book **VARCHAR**(255),

**IN** p\_author **VARCHAR**(255),

**IN** p\_subject **VARCHAR**(255),

**IN** p\_publisher **VARCHAR**(255),

**IN** p\_image **TEXT**,

**IN** p\_pub\_year **YEAR**,

**IN** p\_earliest\_pub\_year **YEAR**,

**IN** p\_language **VARCHAR**(100),

**IN** p\_isbn **VARCHAR**(50),

**IN** p\_permission\_level **INT**

)

**BEGIN**

-- Thêm sách vào bảng all\_book

**INSERT** **INTO** all\_book (

book, author, **`book subject`**, image,

**`publication date`**, **`earliest publication date`**, **language**,

isbn, permission\_level, **`book publisher`**

) **VALUES** (

p\_book, p\_author, p\_subject, p\_image,

p\_pub\_year, p\_earliest\_pub\_year, p\_language,

p\_isbn, p\_permission\_level, p\_publisher

);

**END**

**2.** **Lấy thông tin chi tiết của một cuốn sách**

**CREATE** **DEFINER**=**`root`**@**`%`** **PROCEDURE** **`library\_management`**.**`GetBookById`**(**IN** inputBookId **INT**)

**BEGIN**

**SELECT** \* **FROM** all\_book **WHERE** book\_id = inputBookId;

**END**

**3.Lấy thông tin chi tiết về các bộ sách của một tác giả**

**CREATE** **DEFINER**=**`root`**@**`%`** **PROCEDURE** **`library\_management`**.**`GetBookSeriesDetail`**(

**IN** p\_authorId **INT**

)

**BEGIN**

**SELECT**

a.id **AS** author\_id,

a.name **AS** author\_name,

bs.id **AS** series\_id,

bs.title **AS** series\_title,

bs.description **AS** series\_description

**FROM**

authors a

**JOIN**

book\_series bs **ON** a.id = bs.author\_id

**WHERE**

a.id = p\_authorId;

**END**

**4. Tìm kiếm nhà xuất bản trong bảng all\_book\_publishers**

**CREATE** **DEFINER**=**`root`**@**`%`** **PROCEDURE** **`library\_management`**.**`SearchBookPublisherBeLike`**(**IN** inputName **VARCHAR**(255))

**BEGIN**

**SELECT** publisher\_id, book\_publihser

**FROM** all\_book\_publishers

**WHERE** book\_publihser **LIKE** **CONCAT**('%', inputName, '%');

**END**

**5.Xóa sách**

**CREATE** **DEFINER**=**`root`**@**`%`** **PROCEDURE** **`library\_management`**.**`delete\_book`**(

**IN** p\_book\_id **INT**

)

**BEGIN**

**DELETE** **FROM** all\_book

**WHERE** book\_id = p\_book\_id;

**END**

**6.** **Cập nhật thông tin sách**

**CREATE** **DEFINER**=**`root`**@**`%`** **PROCEDURE** **`library\_management`**.**`update\_book\_info`**(

**IN** p\_book **VARCHAR**(255),

**IN** p\_author **VARCHAR**(255),

**IN** p\_book\_subject **VARCHAR**(255),

**IN** p\_book\_publisher **VARCHAR**(255),

**IN** p\_image **VARCHAR**(255),

**IN** p\_pub\_date **INT**,

**IN** p\_language **VARCHAR**(50),

**IN** p\_earliest\_pub\_date **INT**,

**IN** p\_ISBN **VARCHAR**(20),

**IN** p\_book\_id **INT**

)

**BEGIN**

**UPDATE** all\_book

**SET**

book = p\_book,

author = p\_author,

**`book subject`** = p\_book\_subject,

**`book publisher`** = p\_book\_publisher,

image = p\_image,

**`publication date`** = p\_pub\_date,

**language** = p\_language,

**`earliest publication date`** = p\_earliest\_pub\_date,

ISBN = p\_ISBN

**WHERE**

book\_id = p\_book\_id;

**END**

* 1. **Mượn và trả sách**

**3.** **Lấy thông tin các cuốn sách mà người dùng đã mượn**

**CREATE** **DEFINER**=**`root`**@**`%`** **PROCEDURE** **`library\_management`**.**`GetBorrowedBooksByUser`**(

**IN** p\_userId **VARCHAR**(255)

)

**BEGIN**

**SELECT**

b.Borrow\_ID,

ab.book,

ab.author,

b.Borrow\_Date,

b.Return\_Date,

b.Actual\_Return\_Date,

b.Status

**FROM**

borrow b

**JOIN**

all\_book ab **ON** b.Book\_ID = ab.book\_id

**WHERE**

b.User\_ID = p\_userId;

**END**

* 1. **Quản lý tiền phạt**

**1. Lấy thông tin khoản phạt**

**CREATE** **DEFINER**=**`root`**@**`%`** **PROCEDURE** **`library\_management`**.**`GetFineByFineId`**(

**IN** p\_borrow\_id **INT**

)

**BEGIN**

**SELECT**

f.fine\_id,

f.borrow\_id,

f.amount,

f.reason,

f.fine\_date

**FROM**

fines f

**WHERE**

f.borrow\_id = p\_borrow\_id;

**END**

**2. Lấy thông tin chi tiết khoản phạt cho một giao dịch mượn sách**

**CREATE** **DEFINER**=**`root`**@**`%`** **PROCEDURE** **`library\_management`**.**`GetFineDetail`**(

**IN** p\_borrow\_id **INT**

)

**BEGIN**

**SELECT**

f.fine\_id,

f.borrow\_id,

f.amount,

f.reason,

f.fine\_date

**FROM**

fines f

**WHERE**

f.borrow\_id = p\_borrow\_id

**LIMIT** 1;

**END**

* 1. **Phân quyền hệ thống**