**BÁO CÁO BÀI TẬP LỚN**

**Quản lý thư viện sách HPC**

**HÀ NỘI, THÁNG 07 NĂM 2025**

BỘ GIÁO DỤC VÀ ĐÀO TẠO

**TRƯỜNG CAO ĐẲNG CÔNG NGHỆ BÁCH KHOA HÀ NỘI**

A blue and white logo

AI-generated content may be incorrect.

**Môn học/Mô đun: LẬP TRÌNH TRÊN THIẾT BỊ DI ĐỘNG**

**SV thực hiện: Trần Thái Hòa**

**Lớp:** **2623LMT01**

**Ngành/Nghề:** **CNTT**

**GVHD: Hoàng Ngọc Hưng**

**PHIẾU CHẤM BÀI TẬP LỚN**

Họ và tên: Mã sinh viên:

Lớp: SBD:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **TT** | **Nội dung chấm điểm** | **Điểm**  **tối đa** | **Điểm**  **đạt được** | **Ghi chú** |
| **1** | **Xây dựng màn hình giới thiệu** | **1.0** |  |  |
| **2** | **Xây dựng chức năng Đăng Ký** | **1.0** |  |  |
| **3** | **Xây dựng chức năng Đăng Nhập** | **1.0** |  |  |
| **4** | **Xây dựng chức năng trang quản lý** | **4.0** |  |  |
|  | - Màn hình trang chủ | 2.0 |  |  |
|  | - Màn hình trang chức năng | 1.0 |  |  |
| - Màn hình trang quản lý người dùng | 1.0 |  |  |
| **5** | **Triển khai trên điện thoại Android** | **1.0** |  |  |
| **6** | **Trình bày báo cáo đầy đủ, đúng yêu cầu** | **2.0** |  |  |
|  | - Phân tích và mô tả đề tài | 1.0 |  |  |
| - Trình bày báo cáo đúng quy cách | 1.0 |  |  |
|  | **Tổng điểm** | **10** |  |  |

*Hà Nội, ngày ….. tháng ….. năm 2025*

**GV chấm thi**

[**LỜI NÓI ĐẦU**](#_5tms77qbsa0y)

Ngày nay, cùng với sự phát triển vượt bậc của công nghệ thông tin, việc ứng dụng các giải pháp phần mềm vào công tác quản lý đã trở thành một xu thế tất yếu, giúp tối ưu hóa hiệu quả công việc và nâng cao chất lượng dịch vụ. Đặc biệt, trong lĩnh vực giáo dục, việc hiện đại hóa công tác quản lý thư viện đóng một vai trò quan trọng trong việc phục vụ nhu cầu nghiên cứu và học tập của sinh viên.

Nhận thức được tầm quan trọng đó, trong khuôn khổ môn học "Lập trình trên thiết bị di động", em đã được giao thực hiện bài tập lớn với đề tài: **"Quản lý thư viện sách HPC"**. Đây là cơ hội quý báu để em có thể hệ thống lại những kiến thức đã học, đồng thời áp dụng các công nghệ tiên tiến như ngôn ngữ Dart, framework Flutter cho việc phát triển ứng dụng di động và nền tảng Node.js cho phần máy chủ để giải quyết một bài toán thực tế.

Mục tiêu chính của đề tài là xây dựng một ứng dụng di động tiện lợi, giúp đơn giản hóa các nghiệp vụ quản lý sách, quản lý độc giả và theo dõi quá trình mượn/trả sách tại thư viện, góp phần nâng cao trải nghiệm cho cả người quản lý và sinh viên.

Trong quá trình thực hiện đề tài, mặc dù đã rất cố gắng tìm tòi, học hỏi và vận dụng những kiến thức được truyền đạt, nhưng do kinh nghiệm thực tế còn hạn chế và kiến thức có giới hạn, nên bài báo cáo và sản phẩm không thể tránh khỏi những thiếu sót. Em rất mong nhận được sự chỉ bảo, góp ý của quý thầy cô để đề tài của em được hoàn thiện hơn.

Em xin chân thành cảm ơn thầy đã tận tình hướng dẫn, chỉ bảo và tạo mọi điều kiện thuận lợi để em có thể hoàn thành bài tập lớn này.

*Hà Nội, ngày ... tháng ... năm 2025*

**Sinh viên thực hiện**

*(ký, ghi rõ họ và tên)*

[**MỤC**](#_5tms77qbsa0y) **LỤC**

**CHƯƠNG 1. GIỚI THIỆU ĐỀ TÀI** 1

1.1. Đặt vấn đề 1

1.2. Mô tả hệ thống 2

1.2.1. Kiến trúc hệ thống 2

1.2.2. Các phân hệ chức năng chính 2  
1.3. Khảo sát hệ thống và đề xuất giải pháp 3  
 1.3.1. Khảo sát hiện trạng hệ thống 3  
 1.3.2. Đề xuất giải pháp 4

**CHƯƠNG 2. XÂY DỰNG ỨNG DỤNG** 6  
2.1. Lý thuyết liên quan 6  
 2.1.1. Giới thiệu về Dart và Flutter 6  
 2.1.2. Giới thiệu về Node.js 6  
 2.1.3. Mô hình kiến trúc hệ thống 7  
2.2. Xây dựng cơ sở dữ liệu 7  
 2.2.1. Các bảng dữ liệu 7  
 2.2.2. Mối quan hệ giữa các bảng 8  
2.3. Xây dựng chức năng đăng nhập 9  
2.4. Xây dựng chức năng trang quản trị 9  
 2.4.1. Trang Danh mục sách 9  
 2.4.2. Trang Danh sách sách 11  
 2.4.3. Trang Quản lý người dùng 14  
2.5. Xây dựng giao diện người dùng 15  
 2.5.1. Trang chủ phía người dùng 15  
 2.5.2. Tìm kiếm theo danh mục 14  
 2.5.3. Trang sách 16

2.6. Biểu đồ UML 17  
 2.6.1. Biểu đồ Usecase 17  
 2.6.2. Biểu đồ lớp 17  
 2.6.3. Biểu đồ hoạt động 20  
 2.6.4. Biểu đồ tuần tự 23

**KẾT LUẬN** 26

[**CHƯƠNG 1. GIỚI THIỆU ĐỀ TÀI**](#_7hc9a5wtpd3k)

* 1. [**Đặt vấn đề**](#_m2lxbzrs80pq)

Trong bối cảnh cuộc Cách mạng công nghiệp 4.0 đang diễn ra mạnh mẽ, việc chuyển đổi số và ứng dụng công nghệ thông tin vào mọi lĩnh vực của đời sống xã hội đã trở thành một yêu cầu tất yếu. Đối với các cơ sở giáo dục, thư viện đóng vai trò là trái tim tri thức, là nơi cung cấp nguồn tài liệu, giáo trình và sách tham khảo vô cùng quan trọng, phục vụ trực tiếp cho công tác giảng dạy, học tập và nghiên cứu.

Tuy nhiên, tại thư viện trường Cao đẳng Công nghệ Bách khoa Hà Nội (HPC), công tác quản lý và vận hành hiện nay vẫn còn tồn tại nhiều bất cập do một số quy trình vẫn được thực hiện theo phương pháp thủ công. Việc quản lý các đầu sách, theo dõi tình trạng mượn/trả, và quản lý thông tin độc giả bằng sổ sách hoặc các bảng tính rời rạc không chỉ tốn nhiều thời gian, công sức của cán bộ thư viện mà còn tiềm ẩn nguy cơ sai sót, thất lạc dữ liệu.

Về phía sinh viên, việc phải đến trực tiếp thư viện để tra cứu xem một cuốn sách có sẵn hay không gây ra nhiều bất tiện và lãng phí thời gian. Quy trình mượn trả thủ công cũng thường dẫn đến việc phải chờ đợi, đặc biệt là vào các giờ cao điểm. Hơn nữa, sinh viên thiếu một công cụ hiệu quả để theo dõi lịch sử mượn sách và nhận thông báo về hạn trả sách, dễ dẫn đến tình trạng trả sách trễ hạn.

Những hạn chế này không chỉ làm giảm hiệu quả hoạt động của thư viện mà còn ảnh hưởng trực tiếp đến trải nghiệm học tập và nghiên cứu của sinh viên. Vì vậy, việc xây dựng một hệ thống phần mềm để tự động hóa các quy trình, tối ưu hóa công tác quản lý và mang lại sự tiện lợi cho người dùng là một nhu cầu cấp thiết.

Xuất phát từ những vấn đề thực tiễn đó, đề tài **“Quản lý thư viện sách HPC”** được thực hiện với mong muốn áp dụng công nghệ di động hiện đại để cung cấp một giải pháp toàn diện, giải quyết các khó khăn còn tồn tại, qua đó góp phần nâng cao chất lượng quản lý và phục vụ tại thư viện nhà trường.

**1.**[**2. Mô tả hệ thống**](#_eroq6s1na197)

Để giải quyết các vấn đề đã được đặt ra, hệ thống "Quản lý thư viện sách HPC" được đề xuất xây dựng theo mô hình kiến trúc Client-Server. Kiến trúc này bao gồm hai thành phần chính: một ứng dụng di động (Client) dành cho người dùng và một hệ thống máy chủ (Server) để xử lý logic và quản lý dữ liệu.

***1.2.1. Kiến trúc hệ thống***

* Backend:
  + Được xây dựng trên nền tảng Node.js.
  + Chức năng: Đóng vai trò là bộ não của toàn bộ hệ thống, chịu trách nhiệm xử lý tất cả các logic nghiệp vụ, xác thực người dùng, và quản lý tương tác với cơ sở dữ liệu.
  + Giao tiếp: Cung cấp một bộ các RESTful API để phía Client có thể giao tiếp, gửi yêu cầu và nhận dữ liệu. Dữ liệu được trao đổi chủ yếu dưới định dạng JSON.
  + Cơ sở dữ liệu: Hệ thống sử dụng cơ sở dữ liệu MySQL để lưu trữ toàn bộ thông tin về sách, độc giả, lịch sử mượn trả và các dữ liệu liên quan khác.
* Frontend :
  + Là một ứng dụng di động đa nền tảng, có thể chạy trên cả hệ điều hành Android và iOS.
  + Được phát triển bằng ngôn ngữ Dart và framework Flutter của Google.
  + Chức năng: Cung cấp giao diện đồ họa trực quan và thân thiện, cho phép người dùng (cả độc giả và thủ thư) tương tác với các chức năng của hệ thống. Ứng dụng sẽ gọi đến các API do backend cung cấp để hiển thị thông tin và gửi dữ liệu xử lý.

***1.2.2. Các phân hệ chức năng chính***

Hệ thống được thiết kế để phục vụ cho hai đối tượng người dùng chính là thủ thư và sinh viên, với các nhóm chức năng tương ứng:

1. Đối với thủ thư:

* Quản lý Sách: Thêm mới, cập nhật thông tin, xóa và quản lý số lượng của các đầu sách trong thư viện.
* Quản lý Độc giả: Xem danh sách độc giả, quản lý thông tin và trạng thái tài khoản của độc giả.
* Quản lý Thủ thư: Xem danh sách độc giả, quản lý thông tin và trạng thái tài khoản của độc giả.
* Quản lý Mượn/Trả sách: Tạo phiếu mượn sách cho độc giả, ghi nhận khi độc giả trả sách, xử lý các trường hợp trễ hạn.

2. Đối với sinh viên:

* Đăng ký/Đăng nhập: Cho phép người dùng tạo tài khoản mới và đăng nhập vào hệ thống bằng tài khoản cá nhân.
* Tra cứu Sách: Tìm kiếm sách nhanh chóng theo tên sách, tác giả, hoặc danh mục. Xem thông tin chi tiết của sách, bao gồm mô tả, số lượng còn lại.
* Quản lý tài khoản cá nhân: Xem và cập nhật thông tin cá nhân.
* Xem lịch sử mượn trả: Theo dõi những cuốn sách mình đã và đang mượn, cùng với ngày mượn và ngày trả.

Mô hình này đảm bảo sự tách biệt rõ ràng giữa giao diện và logic, giúp hệ thống dễ dàng bảo trì, nâng cấp và mở rộng trong tương lai.

**1.3. Khảo sát hệ thống và đề xuất giải pháp**

***1.3.1. Khảo sát hiện trạng hệ thống***

Qua quá trình tìm hiểu và khảo sát thực tế tại thư viện trường Cao đẳng Công nghệ Bách khoa Hà Nội (HPC), quy trình quản lý hiện tại vẫn còn phụ thuộc vào các phương pháp thủ công, cụ thể như sau:

* Đối với thủ thư:
  + Quản lý sách: Toàn bộ thông tin về sách chủ yếu được lưu trữ trong các file Excel hoặc sổ sách giấy. Việc cập nhật khi có sách mới về hoặc khi sách bị hỏng, mất mát tốn nhiều thời gian và dễ xảy ra sai sót.
  + Quản lý độc giả: Thông tin sinh viên được quản lý riêng, không có sự liên kết chặt chẽ với hệ thống quản lý sách.
  + Quy trình mượn/trả: Khi sinh viên mượn hoặc trả sách, thủ thư phải tìm kiếm thông tin sách và thông tin sinh viên một cách thủ công trên sổ hoặc file, sau đó ghi lại vào một sổ theo dõi khác. Quy trình này chậm chạp, dễ gây ùn tắc vào giờ cao điểm và khó khăn trong việc truy xuất lịch sử.
  + Thống kê, báo cáo: Việc tổng hợp số liệu để báo cáo (ví dụ: số lượng sách được mượn nhiều nhất, số sinh viên mượn quá hạn) là một công việc rất vất vả, tốn thời gian và thiếu tính chính xác.
* Đối với sinh viên:
  + Tra cứu thông tin: Sinh viên không có cách nào khác ngoài việc phải đến trực tiếp thư viện và hỏi thủ thư để biết một cuốn sách cụ thể có còn trong thư viện hay không.
  + Theo dõi cá nhân: Sinh viên không có công cụ để xem lại lịch sử mượn sách của mình hoặc kiểm tra chính xác ngày cần phải trả sách, dẫn đến tình trạng quên và trả sách trễ hạn.

Những hạn chế của hệ thống thủ công đã làm giảm hiệu suất làm việc của cán bộ thư viện và gây ra nhiều bất tiện cho sinh viên, cho thấy nhu cầu cấp thiết phải có một giải pháp công nghệ để cải thiện tình hình.

***1.3.2. Đề xuất giải pháp***

Dựa trên những phân tích từ hiện trạng, giải pháp được đề xuất là xây dựng một Hệ thống quản lý thư viện sách toàn diện theo mô hình Client-Server, bao gồm:

1. Ứng dụng di động (Client):
   * Công nghệ: Sử dụng ngôn ngữ Dart và framework Flutter.
   * Mục tiêu: Cung cấp một giao diện người dùng hiện đại, thân thiện và tiện lợi trên các thiết bị di động (Android và iOS) cho cả sinh viên và thủ thư.
   * Giải quyết vấn đề cho sinh viên:
     + Cho phép sinh viên đăng nhập, tra cứu sách mọi lúc mọi nơi.
     + Hiển thị thông tin chi tiết và trạng thái (còn/hết) của sách theo thời gian thực.
     + Cung cấp tính năng xem lại lịch sử mượn và gửi thông báo tự động nhắc nhở khi sắp đến hạn trả sách.
   * Hỗ trợ thủ thư: Cung cấp các chức năng quản lý ngay trên thiết bị di động, giúp thủ thư có thể thực hiện một số tác vụ nhanh chóng mà không cần ngồi tại máy tính.
2. Hệ thống máy chủ (Backend):
   * Công nghệ: Sử dụng nền tảng Node.js và cơ sở dữ liệu MySQL.
   * Mục tiêu: Là trung tâm xử lý toàn bộ dữ liệu và logic nghiệp vụ của hệ thống, đảm bảo tính nhất quán, bảo mật và chính xác.
   * Giải quyết vấn đề cho thủ thư:
     + Cung cấp một hệ quản trị tập trung để quản lý toàn bộ đầu sách, danh mục, tác giả.
     + Số hóa quy trình mượn/trả sách, giúp thủ thư thực hiện thao tác chỉ qua vài cú nhấp chuột, giảm thiểu sai sót và tiết kiệm thời gian.
     + Tự động hóa việc tạo các báo cáo, thống kê, giúp ban quản lý có cái nhìn tổng quan về hoạt động của thư viện một cách nhanh chóng.

Kết luận: Việc triển khai hệ thống ứng dụng này được kỳ vọng sẽ khắc phục triệt để các nhược điểm của quy trình quản lý thủ công, mang lại hiệu quả làm việc cao hơn cho cán bộ thư viện và cung cấp một dịch vụ tiện ích, hiện đại cho sinh viên, qua đó nâng cao chất lượng tổng thể của thư viện trường HPC.

[**CHƯƠNG 2.**](#_x5r909lpfboz) **XÂY DỰNG** **ỨNG DỤNG**

**2.1. Lý thuyết liên quan**

***2.1.1. Giới thiệu về Dart và Flutter***

Dart là một ngôn ngữ lập trình được phát triển bởi Google, được thiết kế cho các ứng dụng phía client như web, desktop và mobile. Dart nổi bật bởi cú pháp đơn giản, dễ học, hỗ trợ lập trình hướng đối tượng, và đặc biệt phù hợp với phát triển giao diện người dùng nhờ khả năng hot reload, hiệu năng cao và tích hợp tốt với Flutter.

Flutter là bộ công cụ phát triển giao diện người dùng đa nền tảng (UI toolkit) của Google, cho phép xây dựng ứng dụng cho Android, iOS, web và desktop từ cùng một cơ sở mã nguồn. Flutter sử dụng Dart làm ngôn ngữ lập trình chính, và cung cấp tập hợp các widget tùy biến cao để tạo ra giao diện người dùng linh hoạt và đẹp mắt.

Một số đặc điểm nổi bật của Flutter:

* Hot Reload giúp phát triển nhanh chóng và dễ dàng kiểm thử.
* UI được xây dựng hoàn toàn bằng code dựa trên các widget.
* Hiệu năng cao nhờ sử dụng Skia để render giao diện trực tiếp xuống canvas.

***2.1.2. Giới thiệu về Node.js***

Node.js là môi trường thực thi JavaScript phía server dựa trên công cụ V8 Engine của Google Chrome. Node.js được thiết kế để xây dựng các ứng dụng mạng hiệu suất cao, có khả năng mở rộng, nhờ vào kiến trúc bất đồng bộ (non-blocking I/O) và hướng sự kiện.

Ưu điểm của Node.js:

* Thực thi nhanh, nhẹ, hiệu quả cho ứng dụng thời gian thực.
* Có hệ sinh thái phong phú với trình quản lý gói npm (Node Package Manager).
* Phù hợp để xây dựng RESTful API phục vụ cho các ứng dụng client như Flutter.

Trong đồ án này, Node.js được sử dụng để xây dựng phần backend RESTful, quản lý dữ liệu thư viện, xử lý nghiệp vụ mượn trả sách, xác thực người dùng và cung cấp API cho frontend Flutter gọi đến.

***2.1.3. Giới thiệu mô hình kiến trúc hệ thống***

Ứng dụng được xây dựng theo mô hình Client - Server, cụ thể:

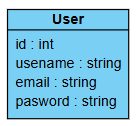
* Client (Flutter App): Giao diện người dùng được xây dựng bằng Flutter, chạy trên thiết bị di động (Android). Người dùng có thể thực hiện các thao tác như đăng nhập, xem sách, mượn/trả sách, tra cứu phiếu mượn.
* Server (Node.js API): Xử lý yêu cầu từ client, kết nối cơ sở dữ liệu (MySQL), thực hiện các nghiệp vụ như quản lý sách, người dùng, phiếu mượn và danh mục.
* Cơ sở dữ liệu (MySQL): Lưu trữ toàn bộ dữ liệu của hệ thống như thông tin sách, người dùng, danh mục, phiếu mượn và chi tiết mượn.

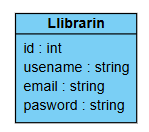
Mô hình backend tuân theo kiến trúc MVC (Model - View - Controller):

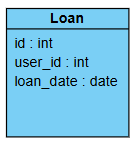
* Model: Định nghĩa cấu trúc dữ liệu và xử lý truy vấn đến CSDL.
* Controller: Xử lý logic nghiệp vụ và định tuyến yêu cầu.
* Routes: Định nghĩa các endpoint API mà frontend sẽ gọi.

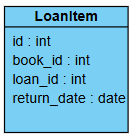
**2.2. Xây dựng cơ sở dữ liệu**

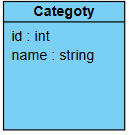
***2.2.1. Các bảng dữ liệu***

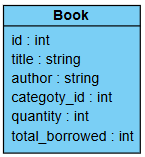
******

******

******

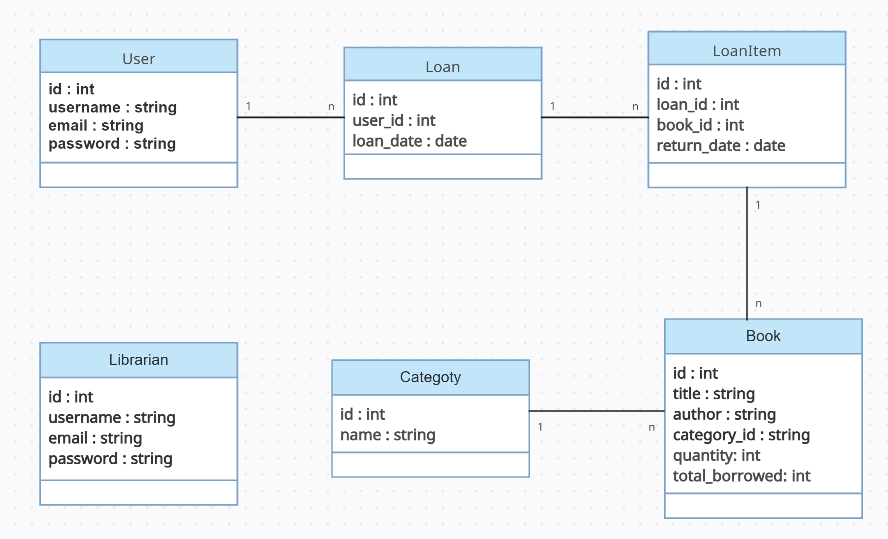
******

******

******

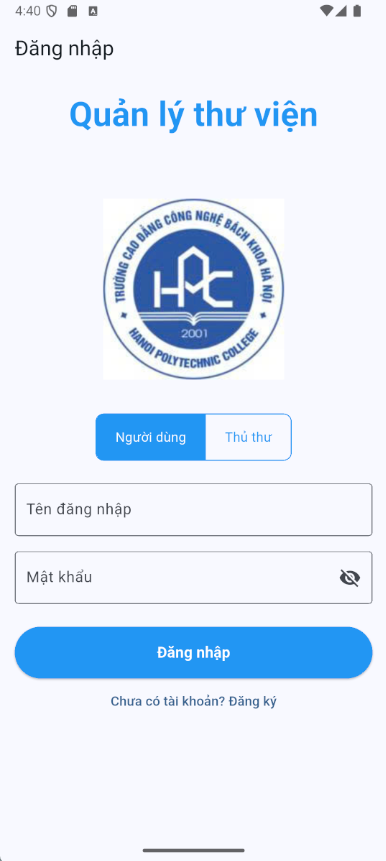
**Hình 2.1 – Các bảng dữ liệu**

***2.2.2. Mối quan hệ giữa các bảng***

******

**Hình 2.2 - Sơ đồ quan hệ thực thể (ERD) của cơ sở dữ liệu**

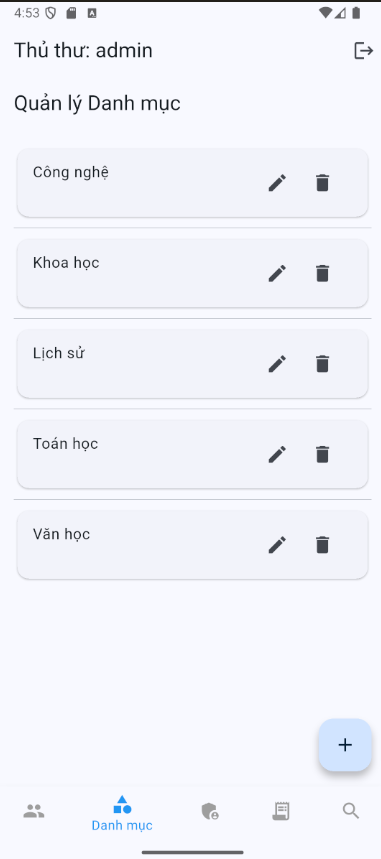
**2.3. Xây dựng chức năng đăng nhập**

******

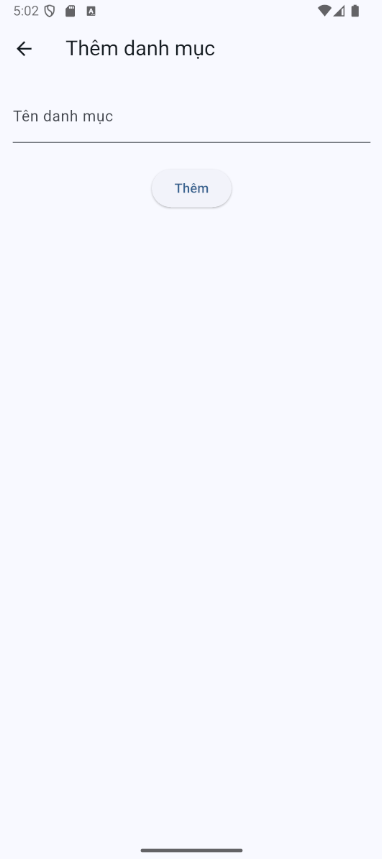
**Hình 2.3 - Giao diện màn hình Đăng nhập**

**2.4. Xây dựng chức năng trang quản trị**

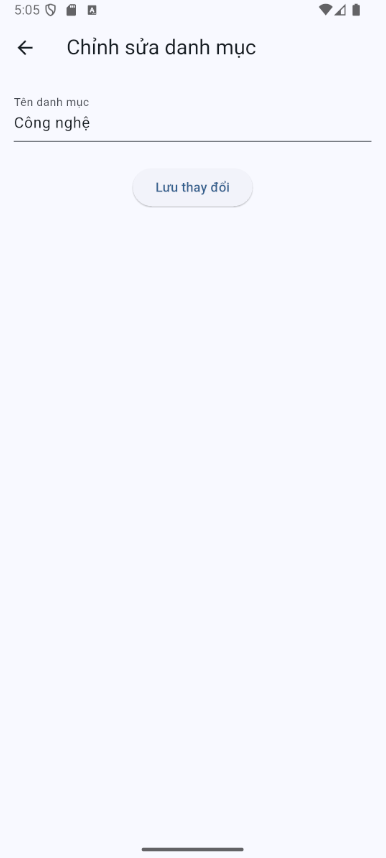
***2.4.1. Trang Danh mục sách***

******

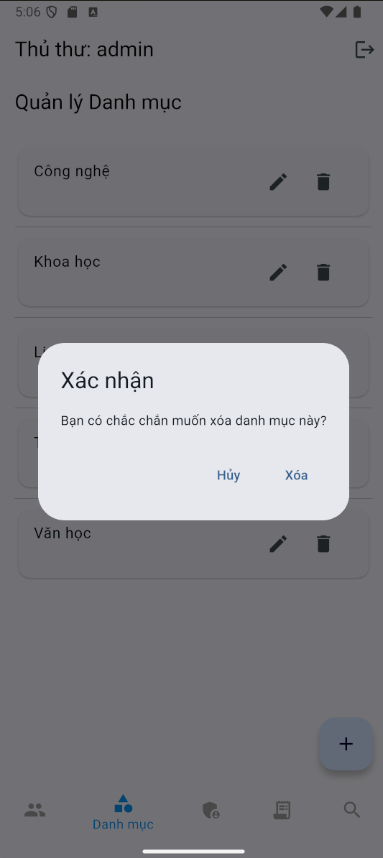
**Hình 2.4 - Giao diện màn hình Quản lý danh mục**

******

**Hình 2.5 - Giao diện chức năng Thêm danh mục**

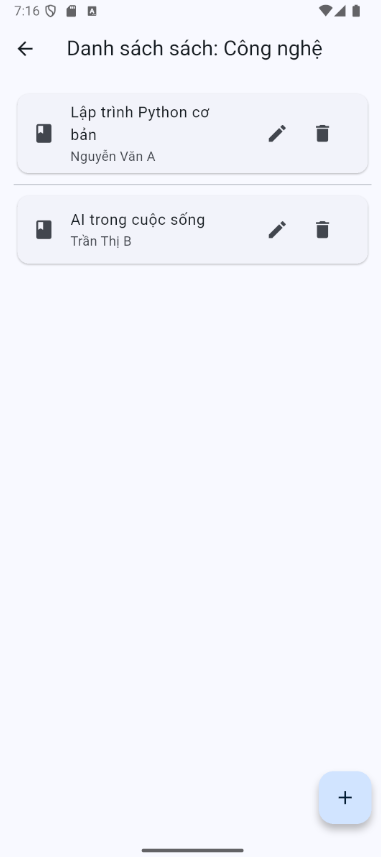


**Hình 2.6 - Giao diện chức năng Sửa danh mục**

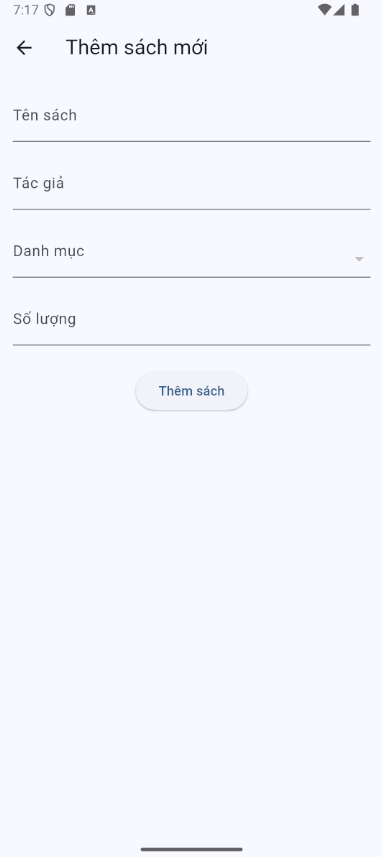


**Hình 2.7 - Hộp thoại xác nhận Xóa danh mục**

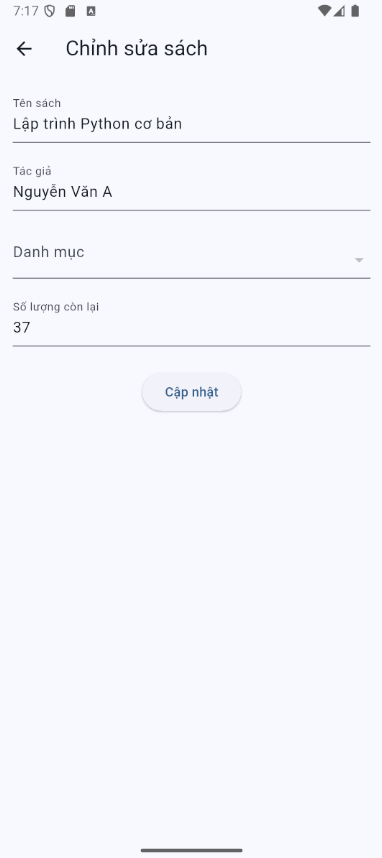
***2.4.2. Trang Danh sách sách***



**Hình 2.8 - Giao diện màn hình Danh sách sách**



**Hình 2.9 - Giao diện chức năng Thêm sách**

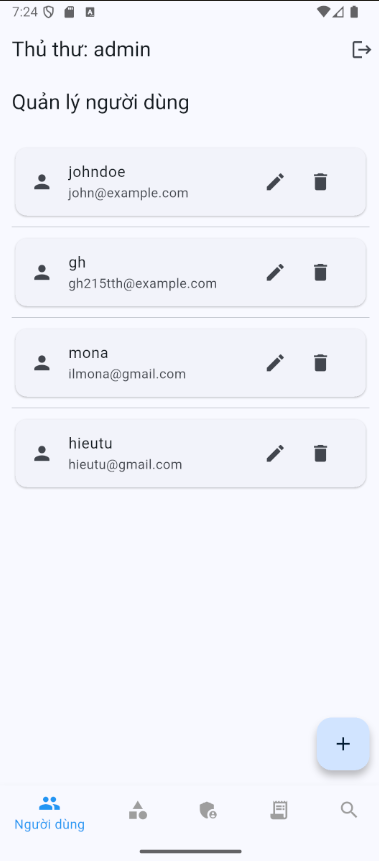


**Hình 2.10 - Giao diện chức năng Sửa sách**

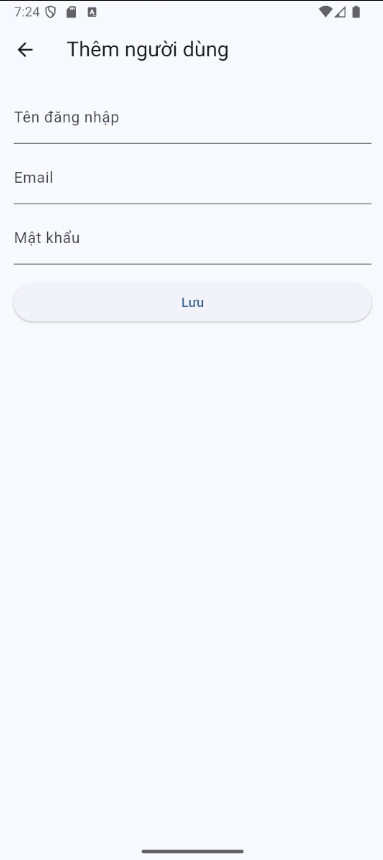


**Hình 2.11 - Hộp thoại xác nhận Xóa sách**

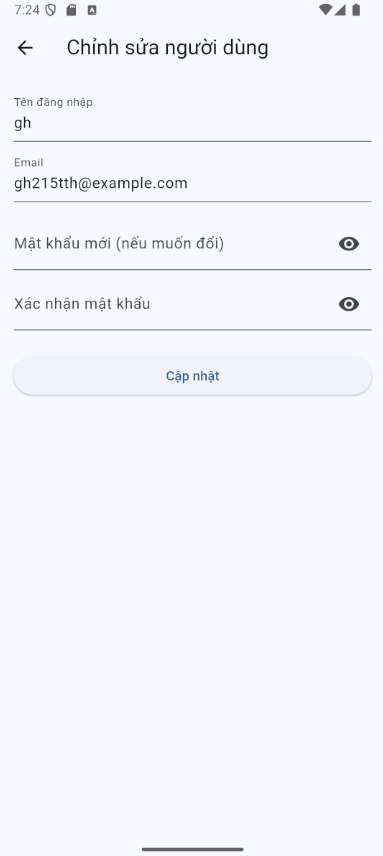
***2.4.3. Trang Quản lý người dùng***



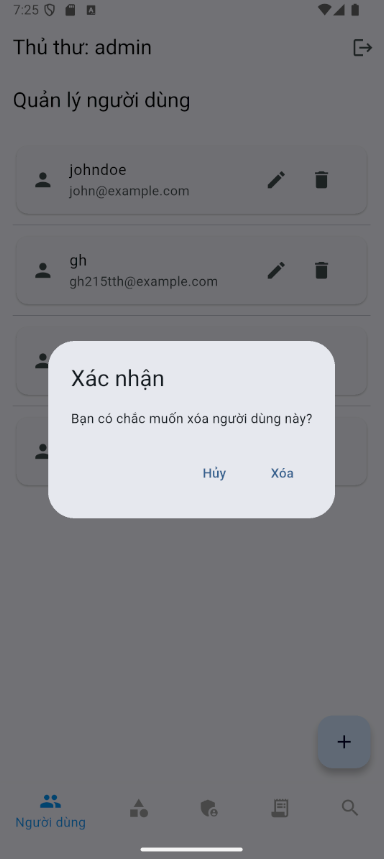
**Hình 2.12 - Giao diện màn hình Quản lý người dùng**



**Hình 2.13 - Giao diện chức năng Thêm người dùng**



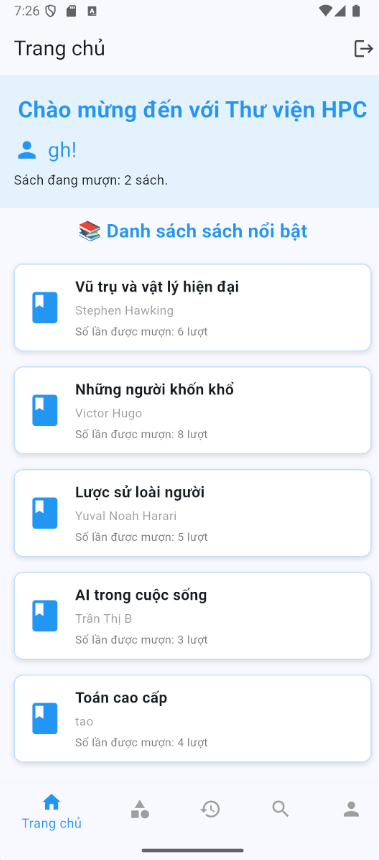
**Hình 2.14 - Giao diện chức năng Sửa người dùng**



**Hình 2.15 - Hộp thoại xác nhận Xóa người dùng**

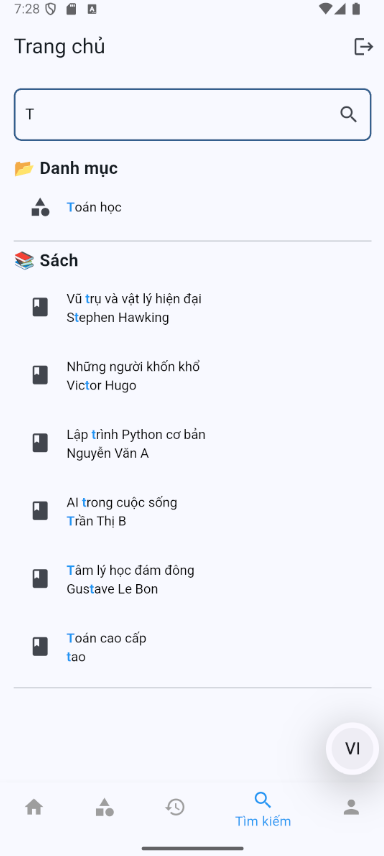
**2.5. Xây dựng chức năng trang giao diện người dùng**

***2.5.1. Hiển thị dữ liệu trên trang chủ.***

****

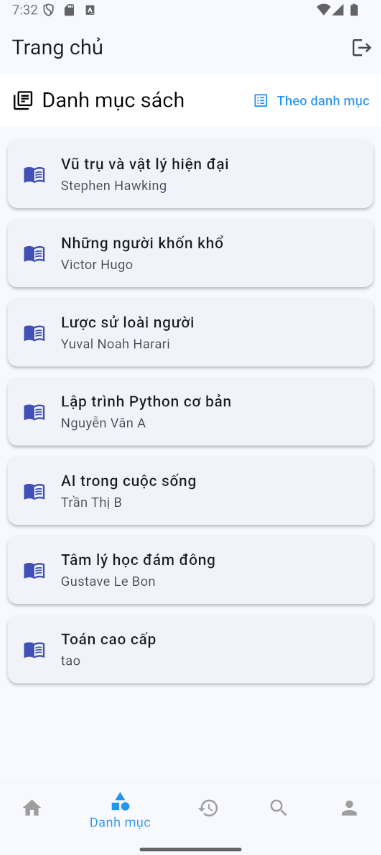
**Hình 2.16 - Giao diện Trang chủ phía người dùng**

***2.5.2. Tìm kiếm theo danh mục***

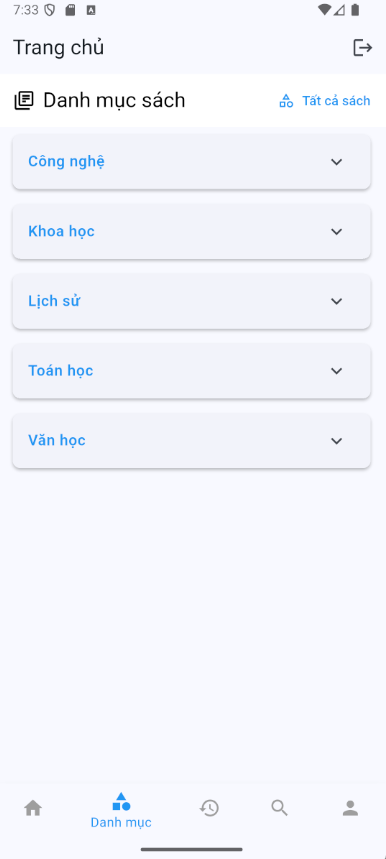


**Hình 2.17 - Giao diện chức năng Tìm kiếm**

***2.5.3. Trang sách***



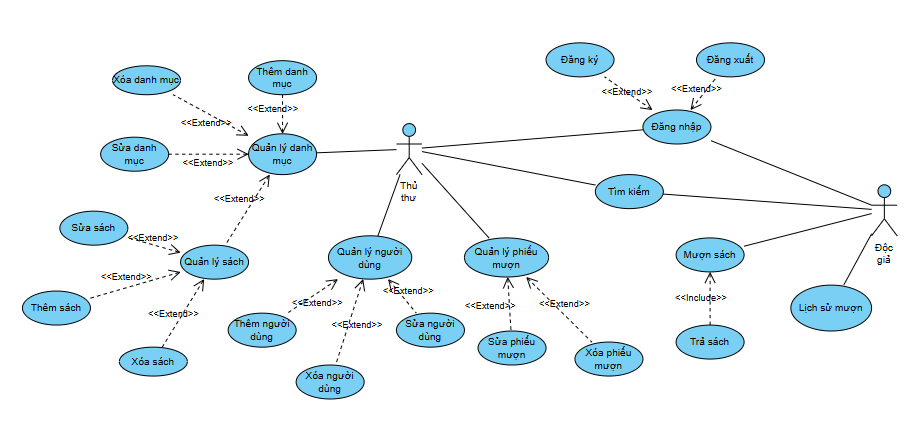
**Hình 2.18 - Giao diện trang Danh mục sách (chế độ xem tất cả)**



**Hình 2.19 - Giao diện trang Danh mục sách (chế độ xem theo danh mục)**

**2.3. Xây dựng biểu đồ UML**

***2.3.1. Biểu đồ usecase***



**Hình 2.20 - Biểu đồ Usecase của hệ thống**

Mô tả  
Các Actor (Tác nhân)

1. Thủ thư (Librarian):
   * Có quyền quản lý toàn bộ hệ thống (người dùng, sách, danh mục, phiếu mượn).
   * Tương tác với hầu hết các Use Case.
2. Độc giả (User/Reader):
   * Chỉ có quyền sử dụng chức năng liên quan đến mượn/trả sách và xem lịch sử cá nhân.

Các Use Case chính

1. Chung cho cả 2 actor:

* Đăng nhập (Login)
* Tìm kiếm (Search)
* Đăng ký (<<extend>> từ Đăng nhập)
* Đăng xuất (<<extend>> từ Đăng nhập)

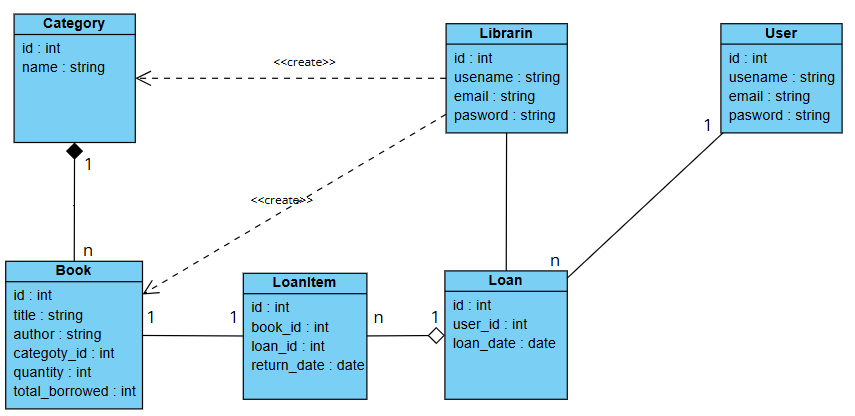
2. Chức năng dành cho Thủ thư

* Quản lý danh mục
  + <<extend>> Thêm danh mục
  + <<extend>> Sửa danh mục
  + <<extend>> Xóa danh mục
* Quản lý sách
  + <<extend>> Thêm sách
  + <<extend>> Sửa sách
  + <<extend>> Xóa sách
* Quản lý người dùng
  + <<extend>> Thêm người dùng
  + <<extend>> Sửa người dùng
  + <<extend>> Xóa người dùng
* Quản lý phiếu mượn
  + <<extend>> Sửa phiếu mượn
  + <<extend>> Xóa phiếu mượn

3. Chức năng dành cho Độc giả

* Mượn sách
  + <<include>> Trả sách
* Lịch sử mượn
* Đăng ký / Đăng nhập / Đăng xuất

***2.3.2. Biểu đồ lớp***



**Hình 2.21 - Biểu đồ lớp của hệ thống**

Mô tả

Các lớp chính trong hệ thống

User:

* Thuộc tính: id, username, email, password
* Mô tả: Người dùng là độc giả có thể đăng ký, đăng nhập, mượn sách

Librarian:

* Thuộc tính: id, username, email, password
* Mô tả: Thủ thư có quyền quản lý người dùng, sách, danh mục, phiếu mượn

Category:

* Thuộc tính: id, name
* Mô tả: Danh mục chứa nhiều sách

Book:

* Thuộc tính: id, title, author, category\_id, quantity, total\_borrowed
* Mô tả: Sách trong thư viện, thuộc về một danh mục

Loan:

* Thuộc tính: id, user\_id, loan\_date
* Mô tả: Phiếu mượn của người dùng

LoanItem:

* Thuộc tính: id, book\_id, loan\_id, return\_date
* Mô tả: Chi tiết sách trong một phiếu mượn

Mối quan hệ giữa các lớp

User 1 --- n Loan:

* Một người dùng có thể có nhiều phiếu mượn

Loan 1 --- n LoanItem:

* Một phiếu mượn có thể chứa nhiều sách

Book 1 --- n LoanItem:

* Một sách có thể được mượn trong nhiều phiếu

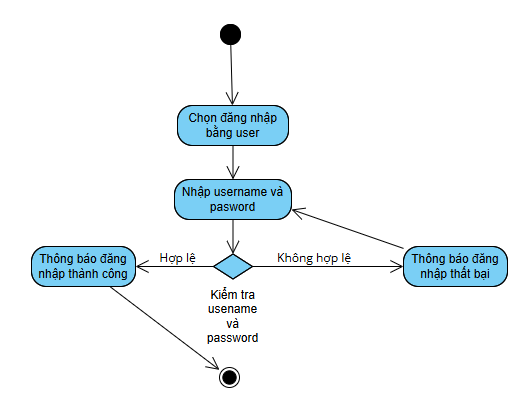
Category 1 --- n Book:

* Một danh mục có thể chứa nhiều sách

Librarian - --> create Category, Book:

* Thủ thư có thể tạo danh mục và sách

***2.3.3. Biểu đồ hoạt động***

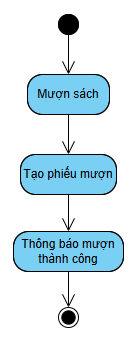
****

**Hình 2.22 - Biểu đồ hoạt động đăng nhập của hệ thống**

Mô tả

1. Bắt đầu  
   → Hệ thống khởi động hoặc người dùng vào giao diện đăng nhập.
2. Chọn đăng nhập bằng user  
   → Người dùng chọn đăng nhập với vai trò user (không phải librarian).
3. Nhập username và password  
   → Nhập thông tin xác thực vào form.
4. Kiểm tra username và password  
   → Hệ thống thực hiện truy vấn và kiểm tra trong CSDL.
5. Rẽ nhánh:

* Nếu hợp lệ:  
  → Thông báo đăng nhập thành công  
  → Chuyển tới trang chính, kết thúc quy trình.
* Nếu không hợp lệ:  
  → Hiển thị thông báo đăng nhập thất bại  
  → Quay lại bước nhập thông tin (cho phép thử lại).



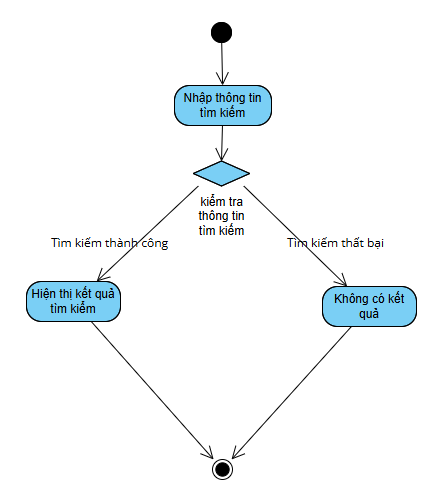
**Hình 2.23 - Biểu đồ hoạt động mượn sách của hệ thống**

Mô tả

1. Bắt đầu  
   → Người dùng khởi động hành động mượn sách.
2. Mượn sách  
   → Gửi yêu cầu mượn sách, bao gồm thông tin sách và người dùng.
3. Tạo phiếu mượn  
   → Hệ thống thực hiện:

* Tạo bản ghi mới trong bảng loan
* Tạo chi tiết phiếu mượn trong loan\_item
* Giảm số lượng tồn kho (quantity - 1)
* Cập nhật total\_borrowed + 1 cho sách tương ứng

1. Thông báo mượn thành công  
   → Hệ thống hiển thị thông báo cho người dùng.
2. Kết thúc  
   → Kết thúc quy trình.



**Hình 2.24 - Biểu đồ hoạt động tìm kiếm của hệ thống**

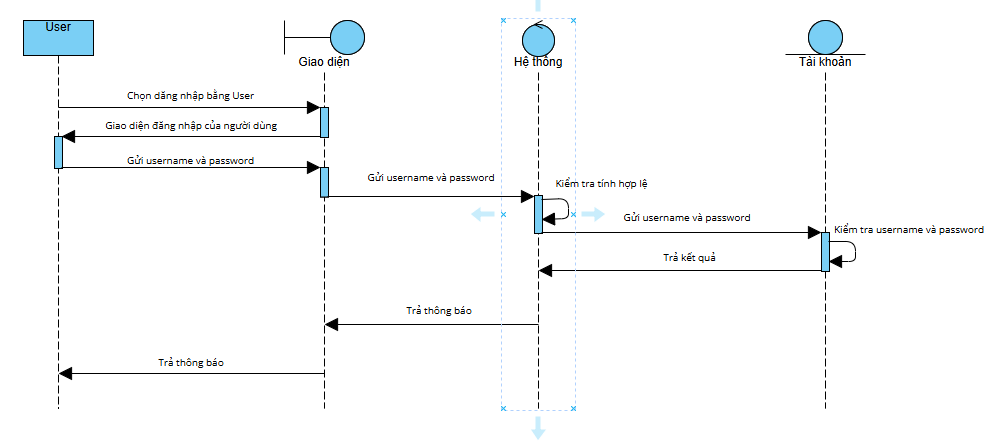
Mô tả

1. Bắt đầu  
   → Người dùng khởi tạo quy trình tìm kiếm.
2. Nhập thông tin tìm kiếm  
   → Nhập từ khóa (tên sách, tác giả, danh mục...).
3. Rẽ nhánh

* Nếu tìm kiếm thành công:  
  → Hiện thị kết quả
* Nếu tìm kiếm thất bại:  
  → Hiện thị không có kết quả

1. Kết thúc  
   → Kết thúc quy trình.

***2.3.3. Biểu đồ tuần tự***



**Hình 2.25 - Biểu đồ tuần tự đăng nhập của hệ thống**

Mô tả

1. Thành phần trong biểu đồ

User: Người dùng thao tác đăng nhập  
Giao diện: Giao diện nhập username/password

Hệ thống: Logic xử lý kiểm tra tài khoản

Tài khoản: Thành phần quản lý dữ liệu tài khoản trong CSDL

2. Luồng xử lý

Bước 1: User → Giao diện

* Chọn chức năng đăng nhập

Bước 2: Giao diện → Hệ thống

* Gửi username và password mà người dùng nhập

Bước 3: Hệ thống → Tài khoản

* Gửi yêu cầu kiểm tra thông tin đăng nhập

Bước 4: Tài khoản → Hệ thống

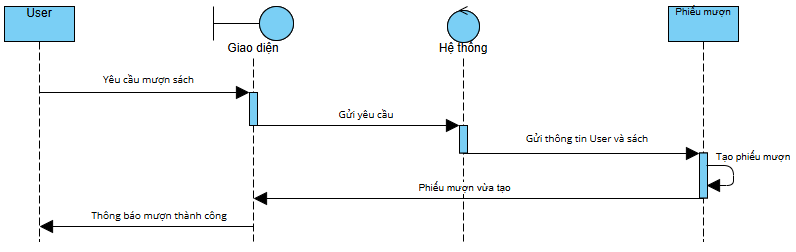
* Trả về kết quả

Bước 5: Hệ thống → Giao diện

* Trả thông báo đăng nhập thành công/thất bại

Bước 6: Giao diện → User

* Hiển thị thông báo cho người dùng

****

**Hình 2.26 - Biểu đồ tuần tự mượn sách của hệ thống**

Mô tả

1. Thành phần trong biểu đồ

User: Người dùng gửi yêu cầu mượn sách  
Giao diện: Giao diện nơi người dùng thao tác

Hệ thống: Logic xử lý nghiệp vụ backend

Phiếu mượn: Thành phần quản lý dữ liệu phiếu mượn trong CSDL

2. Luồng xử lý

Bước 1: User → Giao diện

* Gửi yêu cầu mượn sách

Bước 2: Giao diện → Hệ thống

* Gửi thông tin yêu cầu

Bước 3: Hệ thống → Tài khoản

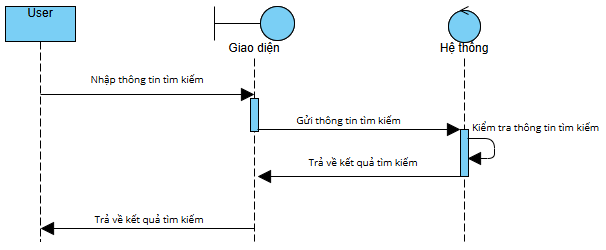
* Gửi thông tin user và sách, tạo mới phiếu mượn

Bước 4: Tài khoản → Hệ thống

* Trả về thông tin phiếu mượn mới tạo

Bước 5: Hệ thống → Giao diện

* Trả thông báo mượn sách thành công

** Hình 2.26 - Biểu đồ tuần tự tìm kiếm của hệ thống**

Mô tả

1. Thành phần trong biểu đồ

User: Người dùng thực hiện hành vi tìm kiếm  
Giao diện: Giao diện nhập thông tin và hiển thị kết quả tìm kiếm

Hệ thống: Phần backend xử lý nghiệp vụ và truy vấn dữ liệu sách

2. Luồng xử lý

Bước 1: User → Giao diện

* Nhập thông tin tìm kiếm

Bước 2: Giao diện → Hệ thống

* Gửi thông tin tìm kiếm đến backend

Bước 3: Hệ thống

* Kiểm tra thông tin tìm kiếm và truy vấn dữ liệu

Bước 4: Hệ thống → User

* Trả về kết quả tìm kiếm

**KẾT LUẬN**

Qua thời gian thực hiện bài tập lớn môn học "Lập trình trên thiết bị di động", với sự hướng dẫn tận tình của thầy giáo và nỗ lực của bản thân, em đã hoàn thành đề tài “Xây dựng ứng dụng Quản lý thư viện sách HPC”. Đề tài đã là một cơ hội quý báu để em vận dụng những kiến thức đã học về phân tích hệ thống, thiết kế cơ sở dữ liệu, và đặc biệt là áp dụng các công nghệ hiện đại như Flutter/Dart cho Frontend và Node.js cho Backend vào việc giải quyết một bài toán thực tế.

1. Kết quả đạt được

Dựa trên mục tiêu đã đề ra, đề tài đã hoàn thành các nội dung chính sau:

* Về mặt phân tích, thiết kế: Đã tiến hành khảo sát và phân tích hiện trạng của công tác quản lý thư viện, từ đó xây dựng các biểu đồ UML cần thiết như Usecase, biểu đồ lớp, biểu đồ hoạt động và tuần tự để mô tả rõ ràng yêu cầu và kiến trúc của hệ thống.
* Về mặt kỹ thuật:
  + Xây dựng thành công ứng dụng di động phía người dùng (Frontend) bằng ngôn ngữ Dart và framework Flutter, đảm bảo giao diện thân thiện và dễ sử dụng.
  + Xây dựng thành công hệ thống máy chủ (Backend) bằng nền tảng Node.js và Express.js, cung cấp các RESTful API để xử lý logic nghiệp vụ và giao tiếp với cơ sở dữ liệu MySQL.
* Về mặt chức năng: Ứng dụng đã đáp ứng được các chức năng cốt lõi cho hai đối tượng người dùng:
  + Thủ thư: Quản lý danh mục, quản lý sách, quản lý người dùng.
  + Sinh viên: Đăng ký, đăng nhập, tra cứu sách, xem danh sách sách theo danh mục.

Sản phẩm bước đầu đã giải quyết được các vấn đề tồn tại của quy trình quản lý thủ công, góp phần số hóa và nâng cao hiệu quả hoạt động của thư viện.

2. Hạn chế của đề tài

Bên cạnh những kết quả đã đạt được, do thời gian và kiến thức còn hạn chế, đề tài vẫn còn một số nhược điểm:

* Các chức năng quản lý mượn/trả sách mới chỉ được phân tích trong biểu đồ, chưa được triển khai đầy đủ trong ứng dụng.
* Giao diện người dùng vẫn còn ở mức cơ bản, chưa có nhiều hiệu ứng và tối ưu sâu về trải nghiệm người dùng (UI/UX).
* Hệ thống chưa có các tính năng nâng cao như thông báo đẩy (push notification) để nhắc hạn trả sách, quét mã vạch/QR code để mượn sách nhanh.
* Ứng dụng chưa được kiểm thử với lượng dữ liệu lớn và nhiều người dùng đồng thời.

3. Hướng phát triển trong tương lai

Để hệ thống được hoàn thiện hơn và có khả năng áp dụng vào thực tế, em xin đề xuất một số hướng phát triển như sau:

* Hoàn thiện đầy đủ các chức năng đã phân tích, đặc biệt là quy trình quản lý mượn/trả sách.
* Nâng cấp và cải tiến giao diện người dùng để trở nên chuyên nghiệp và hấp dẫn hơn.
* Phát triển thêm các tính năng nâng cao như quét mã QR, thông báo, thống kê, báo cáo trực quan.
* Tối ưu hóa hiệu năng của cả Frontend và Backend để hệ thống hoạt động ổn định và mượt mà hơn.
* Triển khai ứng dụng lên các kho ứng dụng như Google Play Store và Apple App Store để người dùng có thể dễ dàng tiếp cận.

Một lần nữa, em xin chân thành cảm ơn sự hướng dẫn của thầy giáo đã giúp em hoàn thành tốt bài tập lớn này.