**BỘ GIÁO DỤC VÀ ĐÀO TẠO**

**TRƯỜNG ĐẠI HỌC ĐẠI NAM**

Logo, company name

Description automatically generated

**BÀI TẬP LỚN**

**TÊN HỌC PHẦN: LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG**

**ĐỀ TÀI: BÀI TOÁN QUẢN LÝ HỒ SƠ, CÔNG VĂN**

**Giáo viên hướng dẫn: ThS. Phạm Văn Tiệp**

**Sinh viên thực hiện:**

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Mã sinh viên** | **Họ và tên** | **Lớp** |
| 1 | 1771020477 | Ngô Văn Minh | CNTT 17-05 |

**CÁN BỘ CHẤM THI**

**Hà Nội, năm 2024**

**BỘ GIÁO DỤC VÀ ĐÀO TẠO**

**TRƯỜNG ĐẠI HỌC ĐẠI NAM**

Logo, company name

Description automatically generated

**BÀI TẬP LỚN**

**TÊN HỌC PHẦN: LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG**

**ĐỀ TÀI: BÀI TOÁN QUẢN LÝ HỒ SƠ, CÔNG VĂN**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| STT | Mã Sinh Viên | Họ và Tên | Ngày Sinh | Điểm | |
| Bằng Số | Bằng Chữ |
| 1 | 1771020477 | Ngô Văn Minh | 17/06/2005 |  |  |

**CÁN BỘ CHẤM THI**

**Hà Nội, năm 2024**

**LỜI NÓI ĐẦU**

Trong thời đại công nghệ ngày nay, việc sử dụng phần mềm để tối ưu hóa các quy trình quản lý là không thể tránh khỏi, đặc biệt là trong lĩnh vực quản lý hồ sơ, công văn. Chúng tôi tự hào giới thiệu đến bạn dự án "Quản lý hồ sơ, công văn" một phương thức quản lý hiện đại và linh hoạt được xây dựng dựa trên nguyên tắc hướng đối tượng và sử dụng Java làm giao diện người dùng.

**Về Dự Án**

Dự án này không chỉ là một sản phẩm phần mềm thông thường, mà còn là một ví dụ xuất sắc về việc kết hợp giữa lập trình hướng đối tượng và giao diện đồ họa trong Java. Được xây dựng trên nguyên tắc của đối tượng và lớp, ứng dụng không chỉ giúp người quản lý dễ dàng theo dõi thông tin hồ sơ và công văn mà còn cung cấp trải nghiệm người dùng tốt nhất thông qua cách sử dụng dễ dàng.

**Về Hướng Đối Tượng trong Java**

Trong quá trình phát triển, chúng tôi đã tập trung vào việc sử dụng các nguyên tắc của lập trình hướng đối tượng để tạo ra các đối tượng, lớp, và kết nối chúng một cách linh hoạt và dễ dàng bảo trì. Điều này giúp chúng tôi xây dựng một mã nguồn sạch sẽ, dễ đọc và mở rộng.

**Mục Tiêu và Mong Muốn**

Mục tiêu của chúng tôi là cung cấp một giải pháp quản lý hồ sơ, công văn hoàn chỉnh và linh hoạt, giúp người quản lý dễ dàng theo dõi thông tin hồ sơ, công văn. Chúng tôi hy vọng bài toán này sẽ không chỉ là một công cụ hữu ích cho các công việc quản lý mà còn là một ví dụ tuyệt vời về việc áp dụng lập trình hướng đối tượng vào thực tiễn.

***Em xin chân thành cảm ơn***

**MỤC LỤC**

**CHƯƠNG 1. GIỚI THIỆU VÀ NỀN TẢNG CÔNG NGHỆ**

**1.1 Giới Thiệu Đề Tài**

Quản lý tài liệu hồ sơ, công văn:

Trong thời đại kỹ thuật số ngày nay, các tổ chức ở mọi quy mô đều phải vật lộn với khối lượng và tính phức tạp ngày càng tăng của hồ sơ và tài liệu. Quản lý hiệu quả các tài sản này là rất quan trọng để duy trì sự tuân thủ, tạo điều kiện thuận lợi cho việc ra quyết định và đảm bảo tính liên tục của hoạt động kinh doanh.

Các hệ thống giấy tờ truyền thống thường tỏ ra không đủ, dẫn đến tình trạng thiếu hiệu quả, lỗi và rủi ro bảo mật. Để giải quyết những thách thức này, các tổ chức đang chuyển sang các giải pháp dựa trên công nghệ.

Đề tài này khám phá sự phát triển của một ứng dụng dựa trên Java được thiết kế để hợp lý hóa việc quản lý hồ sơ và tài liệu, cung cấp một giải pháp mạnh mẽ và có thể mở rộng. Bằng cách tận dụng sức mạnh của Java, ứng dụng này nhằm mục đích tự động hóa các tác vụ thường xuyên, tăng cường bảo mật dữ liệu và cải thiện hiệu quả chung của tổ chức.

**1.2 Mục Tiêu và Phạm Vi**

***1.2.1 Mục Tiêu***

Mục tiêu chính của hệ thống quản lý hồ sơ và tài liệu dựa trên Java là tổ chức, lưu trữ, truy xuất và quản lý hiệu quả các tài liệu kỹ thuật số của tổ chức . Bằng cách tự động hóa các quy trình thủ công và triển khai các biện pháp bảo mật mạnh mẽ, hệ thống như vậy nhằm mục đích cải thiện năng suất và tính tuân thủ chung.

Các mục tiêu cụ thể bao gồm:

Lưu trữ tập trung: Cung cấp kho lưu trữ thống nhất cho tất cả tài liệu của tổ chức.

Truy xuất hiệu quả: Cho phép truy cập tài liệu nhanh chóng và dễ dàng thông qua khả năng tìm kiếm nâng cao.

Kiểm soát phiên bản: Theo dõi các phiên bản tài liệu và những thay đổi theo thời gian.

Kiểm soát truy cập: Triển khai các quyền truy cập chi tiết để bảo vệ thông tin nhạy cảm.

Tự động hóa quy trình làm việc: Tự động hóa quy trình làm việc của tài liệu, chẳng hạn như quy trình phê duyệt và định tuyến.

Tích hợp với các hệ thống khác: Tích hợp liền mạch với các hệ thống tổ chức hiện có như ERP và CRM.

Bảo mật và Tuân thủ: Đảm bảo tính toàn vẹn, tính bảo mật của dữ liệu và tuân thủ các quy định của ngành.

***1.2.2 Phạm Vi***

Phạm vi của hệ thống quản lý hồ sơ và tài liệu dựa trên Java có thể thay đổi tùy thuộc vào nhu cầu cụ thể của một tổ chức. Tuy nhiên, một hệ thống điển hình sẽ bao gồm các lĩnh vực chính sau:

Chức năng cốt lõi:

Tải lên và lưu trữ tài liệu: Cho phép người dùng tải lên và lưu trữ nhiều định dạng tài liệu khác nhau (PDF, Word, Excel, v.v.).

Quản lý siêu dữ liệu: Thu thập và lập chỉ mục siêu dữ liệu (tiêu đề, tác giả, từ khóa, ngày tạo) để tìm kiếm hiệu quả.

Tìm kiếm và Truy xuất: Triển khai các khả năng tìm kiếm mạnh mẽ để định vị tài liệu dựa trên từ khóa, siêu dữ liệu hoặc tìm kiếm toàn văn.

Kiểm soát phiên bản: Theo dõi các phiên bản tài liệu và cho phép so sánh và khôi phục dễ dàng.

Kiểm soát quyền truy cập: Xác định vai trò và quyền của người dùng để hạn chế quyền truy cập vào các tài liệu hoặc thư mục cụ thể.

Quản lý quy trình công việc: Tự động hóa quy trình công việc tài liệu, chẳng hạn như quy trình phê duyệt và định tuyến.

Tính năng bảo mật: Triển khai mã hóa, kiểm soát truy cập và theo dõi để bảo vệ thông tin nhạy cảm.

**1.3 Giới Thiệu về Ngôn Ngữ Lập Trình**

***Ngôn Ngữ Lập Trình Java***

Java là một trong những ngôn ngữ lập trình phổ biến và mạnh mẽ được sử dụng rộng rãi trong việc phát triển ứng dụng đa nhiệm, đa nền tảng. Được phát triển bởi Sun Microsystems (nay là Oracle Corporation), Java được thiết kế với triết lý "Write Once, Run Anywhere" (Viết Một Lần, Chạy Mọi Nơi), cho phép các chương trình Java chạy trên nhiều hệ thống máy tính khác nhau mà không cần biên dịch lại.

Java cung cấp một cộng đồng lớn, nhiều thư viện và framework hỗ trợ, giúp việc phát triển ứng dụng trở nên thuận tiện và hiệu quả. Đặc biệt, Java cũng hỗ trợ lập trình hướng đối tượng, giúp chúng ta xây dựng các ứng dụng có cấu trúc chặt chẽ, dễ duy trì và mở rộng

**1.4 Các Khái Niệm Cơ Bản Trong Lập Trình Hướng Đối Tượng**

Lập trình hướng đối tượng (OOP) là một phương pháp lập trình mà trong đó mọi thứ trong chương trình đều được coi như là một đối tượng. Các đối tượng này có thể chứa dữ liệu trong các biến được gọi là thuộc tính và các hàm được gọi là phương thức để thao tác dữ liệu. Dưới đây là một số nguyên tắc quan trọng của lập trình hướng đối tượng được chúng tôi áp dụng trong dự án "Quản Lý Hồ Sơ, Công Văn":

***1.4.1 Lớp và Đối Tượng (Class, Object)***

Lớp (Class): Là bản thiết kế của một đối tượng, xác định các thuộc tính (attribute) và hành vi (method) mà đối tượng đó sẽ có. Ví dụ: lớp HoSo và CongVan trong đoạn mã bạn cung cấp.

Đối tượng (Object): Là một thực thể được tạo ra từ một lớp, nó có các thuộc tính và hành vi cụ thể của lớp đó. Ví dụ: mỗi hồ sơ hoặc công văn trong danh sách quanLyHoSo và quanLyCongVan là một đối tượng của lớp HoSo hoặc CongVan

***1.4.2 Kế Thừa (Inheritance)***

Kế thừa cho phép một lớp (lớp con) kế thừa các thuộc tính và phương thức từ một lớp khác (lớp cha).

Trong dự án, chúng tôi đã sử dụng kế thừa để tạo ra các lớp con như **HoSo** và **CongVan**, kế thừa từ lớp cha **TaiLieu** để tái sử dụng mã nguồn và giảm thiểu việc lặp lại mã.

***1.4.3 Đa Hình (Polymorphism)***

Là khả năng của một đối tượng có thể có nhiều hình dạng khác nhau. Điều này được thể hiện qua việc các đối tượng của các lớp con có thể được sử dụng thay cho các đối tượng của lớp cha ở những nơi mà lớp cha được yêu cầu.

***1.4.4 Trừu Tượng (Abstraction)***

Trừu tượng là việc ẩn đi các chi tiết phức tạp và chỉ hiển thị những thông tin cần thiết cho người sử dụng.

Trong dự án, chúng tôi đã sử dụng trừu tượng để xây dựng các lớp trừu tượng như **TaiLieu** để định nghĩa các thuộc tính và phương thức chung mà tất cả các sách đều có.

Sử dụng các nguyên tắc của lập trình hướng đối tượng giúp chúng tôi xây dựng ứng dụng một cách cấu trúc, linh hoạt và dễ bảo trì, đồng thời tăng tính tái sử dụng của mã nguồn và giúp chúng tôi phát triển dự án một cách hiệu quả.

***1.4.5 Xử lý ngoại lệ (Exception handling)***

Là cơ chế cho phép chương trình xử lý các lỗi xảy ra trong quá trình thực thi. Trong Java, sử dụng từ khóa try, catch và finally để xử lý ngoại lệ.

***1.4.6 Đầu vào và đầu ra (I/O):***

Là quá trình đưa dữ liệu vào chương trình (input) và đưa dữ liệu từ chương trình ra (output). Trong Java, có nhiều cách để thực hiện I/O, ví dụ như sử dụng các lớp trong gói java.io.

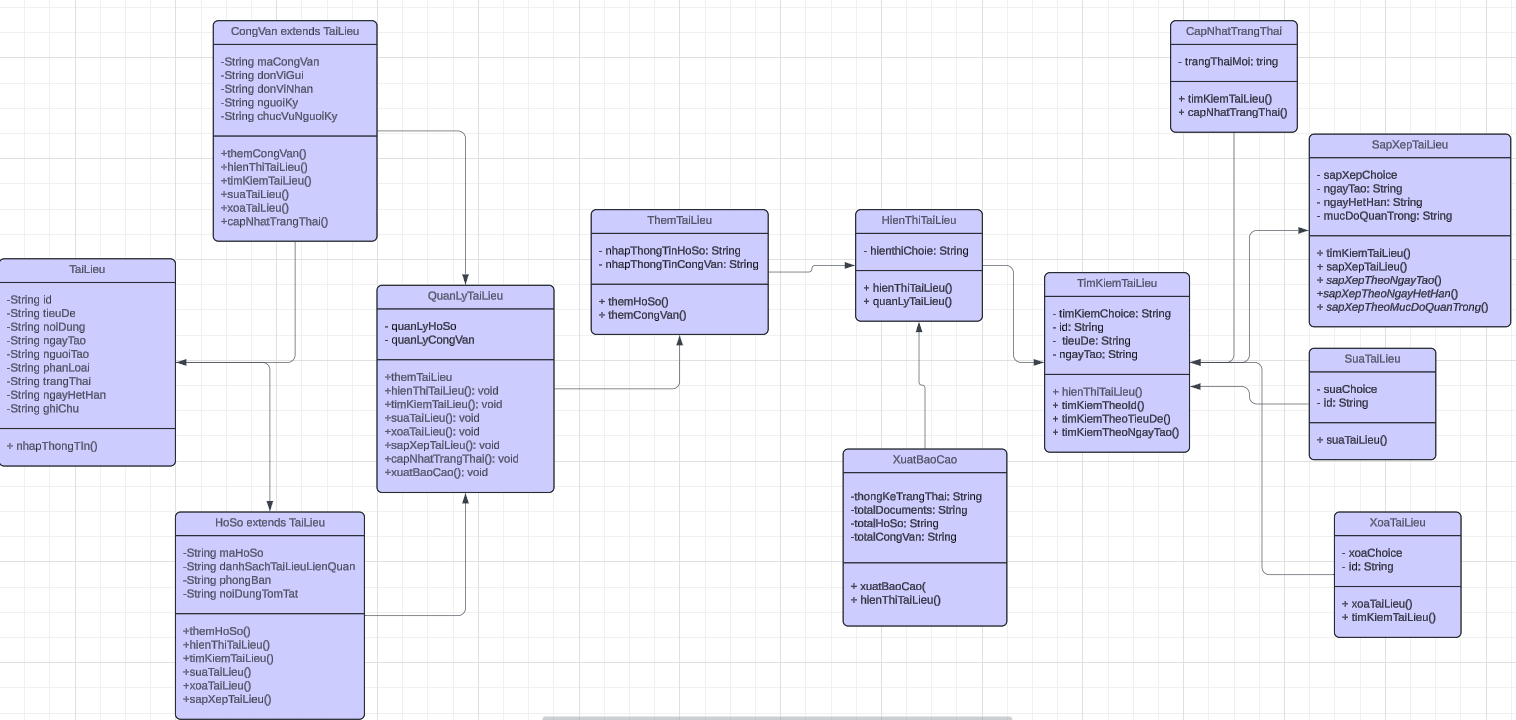
**CHƯƠNG 2. THIẾT KẾ VÀ TRIỂN KHAI BÀI TOÁN**

**2.1 Công Nghệ và Công Cụ Sử Dụng**

Ngôn ngữ Lập Trình Java: Ứng dụng được viết bằng ngôn ngữ lập trình Java, một ngôn ngữ linh hoạt và mạnh mẽ, hỗ trợ lập trình hướng đối tượng.

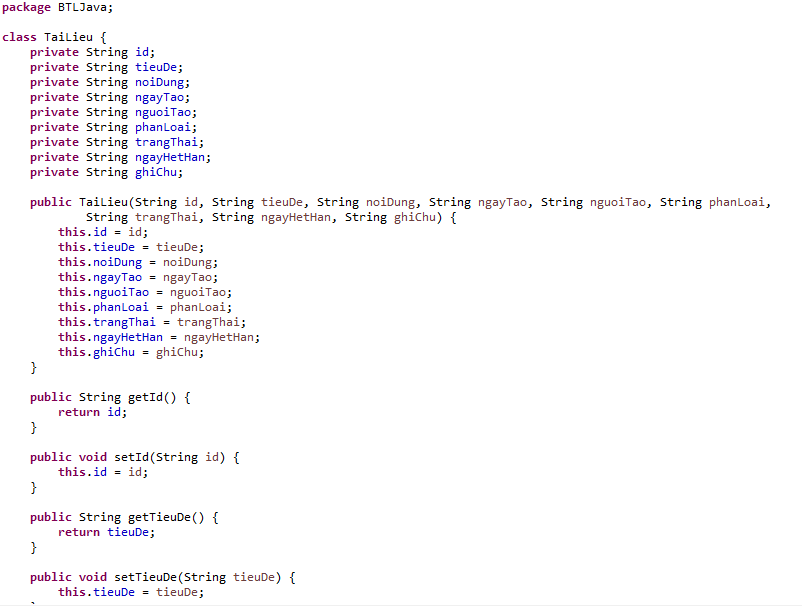
Eclipse IDE: Chúng tôi sử dụng Eclipse IDE để phát triển ứng dụng. Eclipse cung cấp các tính năng hỗ trợ debug, quản lý mã nguồn và tạo giao diện người dùng dễ dàng.

**2.2 Sơ đồ bài toán**



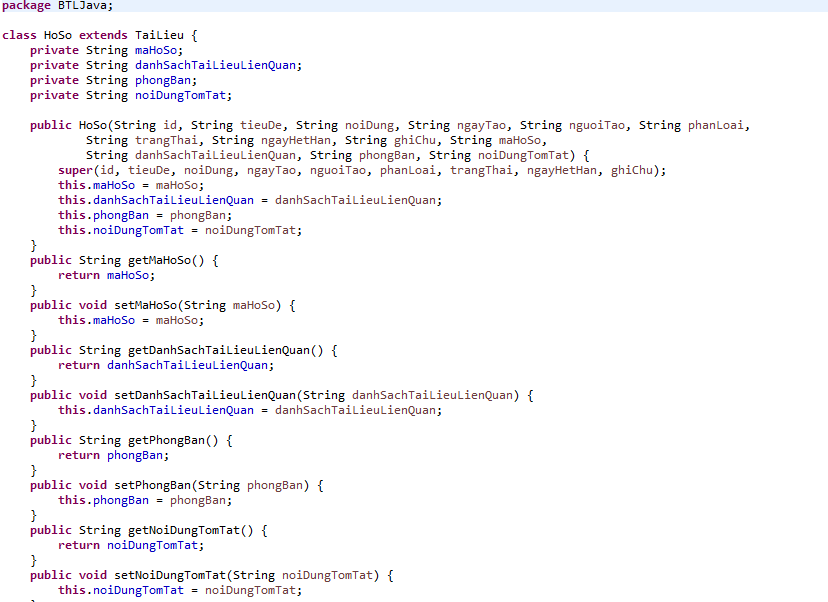
*Hình 2.0: Biểu đồ StarUML*

**2.3 Class TaiLieu**



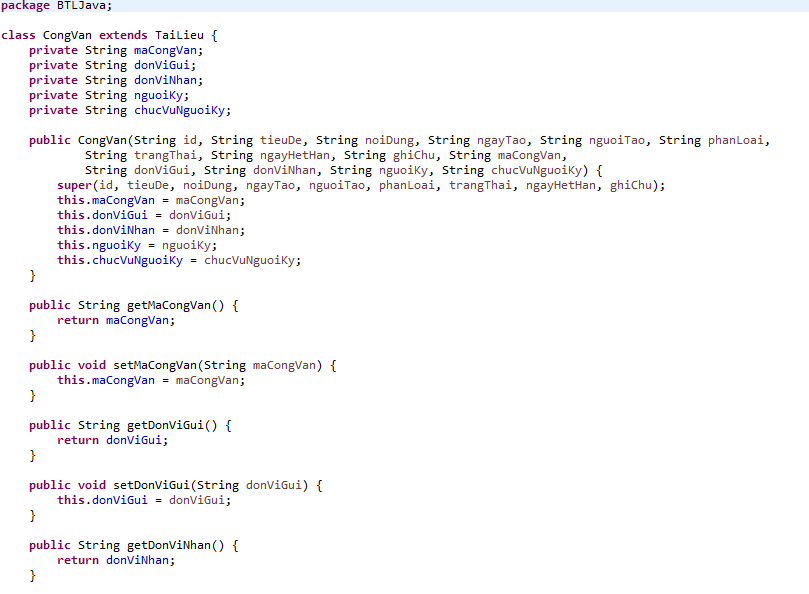
*Hình 2.1: Class TaiLieu*

**2.3.1 Class HoSo**



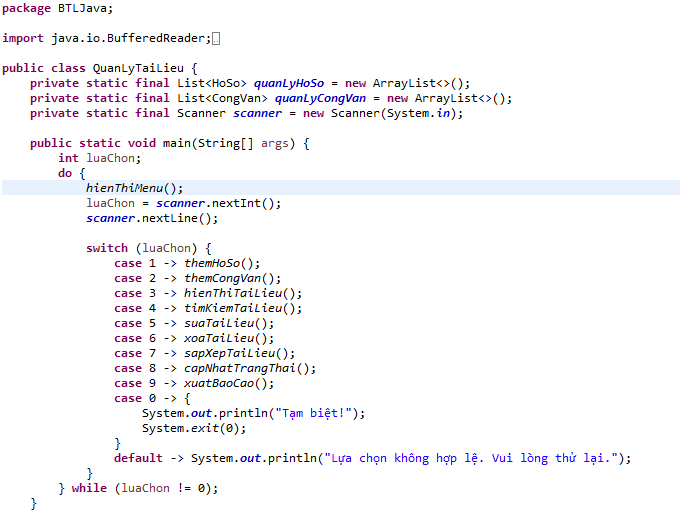
*Hình 2.2: ClassHoSo*

**2.3.2 ClassCongVan**

**

*Hình 2.3: ClassCongVan*

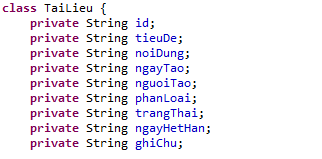
**2.3.3 ClassCongVan**

**

*Hình 2.3: ClassQuanLyTaiLieu*

**CHƯƠNG 3. MÔ TẢ THUẬT TOÁN – THAO TÁC**

**3.1. Lớp Tài liệu**

****

*Hình 3.0: Khai báo thông tin TaiLieu*

Lớp Java này, có tên là TaiLieu, có thể được thiết kế để biểu diễn một tài liệu hoặc tài nguyên kỹ thuật số. Nó bao gồm nhiều thuộc tính khác nhau đặc trưng cho một tài liệu như vậy, bao gồm:

3.1.1. Nhận dạng:

id: Mã định danh duy nhất cho tài liệu.

3.1.2. Nội dung:

tieuDe: Tiêu đề của tài liệu.

noiDung: Nội dung hoặc phần thân của tài liệu.

3.1.3. Siêu dữ liệu:

ngayTao: Ngày tạo tài liệu.

nguoiTao: Cá nhân hoặc tổ chức đã tạo ra tài liệu.

phanLoai: Thể loại hoặc phân loại của tài liệu.

trangThai: Trạng thái hiện tại của tài liệu (ví dụ: đang hoạt động, không hoạt động, đang chờ xử lý).

ngayHetHan: Ngày hết hạn của tài liệu, nếu có.

ghiChu: Ghi chú hoặc bình luận bổ sung về tài liệu.

3.1.4. Những điểm chính:

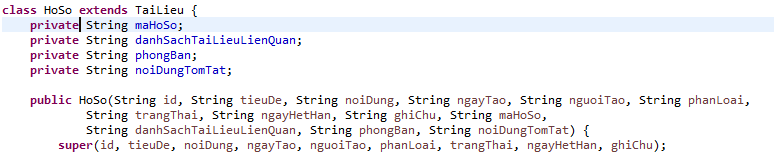
3.1.4.1. Đóng gói:

Lớp này đóng gói các thành viên dữ liệu (trường) trong thân lớp, làm cho chúng trở nên riêng tư. Điều này bảo vệ tính toàn vẹn của dữ liệu và ngăn chặn việc sửa đổi trực tiếp từ bên ngoài lớp.

3.1.4.2. Ẩn dữ liệu:

Bằng cách làm cho các trường trở nên riêng tư, lớp kiểm soát quyền truy cập vào dữ liệu thông qua các phương thức công khai (getter và setter). Điều này đảm bảo rằng dữ liệu được sử dụng và sửa đổi theo cách được kiểm soát.

**3.2. Lớp Hồ Sơ**

****

**Hình 3.1: Lớp hồ sơ**

Lớp Java này, có tên là HoSo, có thể được thiết kế để biểu diễn một tài liệu hoặc tài nguyên kỹ thuật số. Nó bao gồm nhiều thuộc tính khác nhau đặc trưng cho một hồ sơ như vậy, bao gồm:

3.2.1. Nhận dạng:

id: Mã định danh duy nhất cho tài liệu.

3.2.2. Nội dung:

tieuDe: Tiêu đề của tài liệu.

noiDung: Nội dung hoặc phần thân của tài liệu.

3.2.3. Siêu dữ liệu:

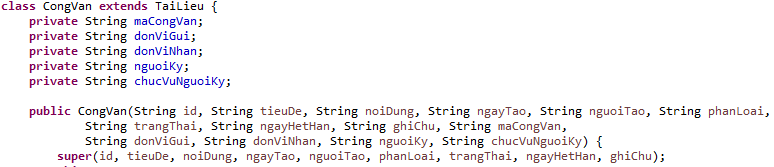
maHoSo: Mã của hồ sơ.

danhSachTaiLieuLienQuan: Dùng để them các danh sách tài liệu liên quan đế hồ sơ.

phongBan: Ban gửi đi hồ sơ.

noiDungTomTat: Dùng để tóm tắt lại nội dung của bạn.

**3.3. Lớp Công Văn**

****

*Hình 3.2: Lớp công văn*

Lớp này CongVan, có khả năng đại diện cho "Tài liệu chính thức ", là một phiên bản chuyên biệt khác của TaiLieulớp. Nó kế thừa tất cả các thuộc tính của TaiLieuvà thêm các thuộc tính cụ thể để mô tả một tài liệu chính thức.

3.3.1. Nhận dạng:

id: Mã định danh duy nhất cho tài liệu.

3.3.2. Nội dung:

tieuDe: Tiêu đề của tài liệu.

noiDung: Nội dung hoặc phần thân của tài liệu.

3.1.3. Siêu dữ liệu:

maCongVan: Mã định danh duy nhất cho tài liệu chính thức.

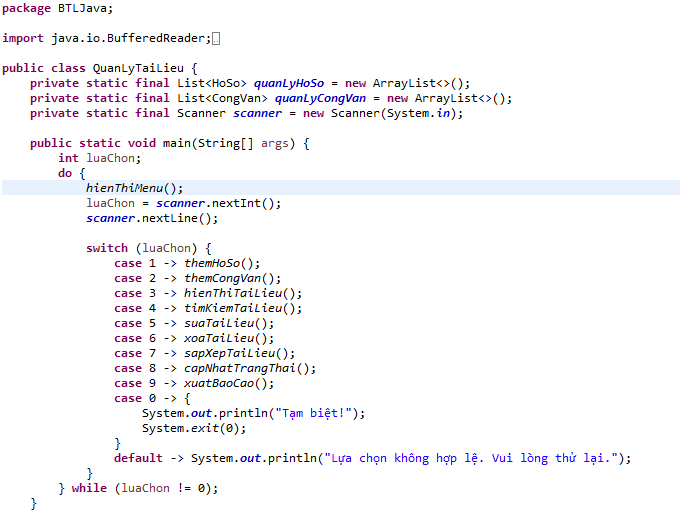
donViGui:Tổ chức hoặc phòng ban gửi.

donViNhan:Tổ chức hoặc phòng ban tiếp nhận.

nguoiKy:Người đã ký vào văn bản.

chucVuNguoiKy:Chức vụ hoặc chức danh của người ký văn bản.

**3.4. Lớp Quản Lý Hồ Sơ, Công Văn**

****

*Hình 3.3: Lớp quản lý*

Lớp này biểu thị một chương trình quản lý tài liệu, có thể bao gồm các tệp ("HoSo") và tài liệu chính thức ("CongVan"). Sau đây là phân tích về mã:

Biến tĩnh:

* quanLyHoSo: ArrayListĐể lưu trữ HoSocác đối tượng (tệp).
* quanLyCongVan: Nơi ArrayListlưu trữ CongVanđồ vật (văn bản chính thức).
* scanner: Một Scannerđối tượng để đọc dữ liệu đầu vào của người dùng từ bảng điều khiển.

mainPhương pháp:

* Đây là điểm bắt đầu của chương trình.
* Nó tạo ra một vòng lặp tiếp tục cho đến khi người dùng chọn thoát (tùy chọn 0).
* Bên trong vòng lặp:
  + hienThiMenu(): Hiển thị menu có các tùy chọn để quản lý tài liệu.
  + Người dùng nhập một lựa chọn ( luaChon).
  + Một switchcâu lệnh xử lý các lựa chọn khác nhau:
    - 1: Gọi themHoSo()để thêm tệp mới.
    - 2: Kêu gọi themCongVan()thêm văn bản chính thức mới.
    - 3: Gọi hienThiTaiLieu()để hiển thị các tài liệu hiện có.
    - 4: Gọi timKiemTaiLieu()để tìm kiếm tài liệu.
    - 5: Gọi suaTaiLieu()để chỉnh sửa tài liệu hiện có.
    - 6: Gọi xoaTaiLieu()để xóa một tài liệu hiện có.
    - 7: Gọi sapXepTaiLieu()để sắp xếp tài liệu (không cung cấp cách triển khai).
    - 8: Gọi capNhatTrangThai()để cập nhật trạng thái của tài liệu (không cung cấp cách triển khai).
    - 9: Gọi xuatBaoCao()để tạo báo cáo (không cung cấp cách triển khai).
    - 0: Thoát khỏi chương trình.
  + Nếu lựa chọn không hợp lệ, thông báo lỗi sẽ hiển thị.

*3.4.1. hienThiMenu():*

Có thể hiển thị menu dạng văn bản với các tùy chọn để thêm, hiển thị, tìm kiếm, chỉnh sửa, xóa, sắp xếp tài liệu và tạo báo cáo.

*3.4.2.* themHoSo():

Yêu cầu người dùng nhập thông tin để tạo HoSođối tượng mới và thêm đối tượng đó vào quanLyHoSodanh sách.

*3.4.3.* themCongVan():

Tương tự như themHoSo()nhưng tạo ra một CongVanđối tượng.

*3.4.4.* hienThiTaiLieu():

Lặp qua danh sách tài liệu (hoặc quanLyHoSo) quanLyCongVanvà hiển thị thông tin về từng tài liệu.

*3.4.5.* timKiemTaiLieu():

Cho phép người dùng tìm kiếm tài liệu dựa trên các tiêu chí như tiêu đề, ID, v.v.

*3.4.6.* suaTaiLieu():

Yêu cầu người dùng chọn một tài liệu và cho phép chỉnh sửa thông tin của tài liệu đó.

*3.4.7.* xoaTaiLieu():

Cho phép người dùng chọn một tài liệu và xóa nó khỏi danh sách thích hợp.

*3.4.8.* sapXepTaiLieu():

Sắp xếp tài liệu dựa trên các tiêu chí cụ thể (ví dụ: ngày, tiêu đề).

*3.4.9.* capNhatTrangThai():

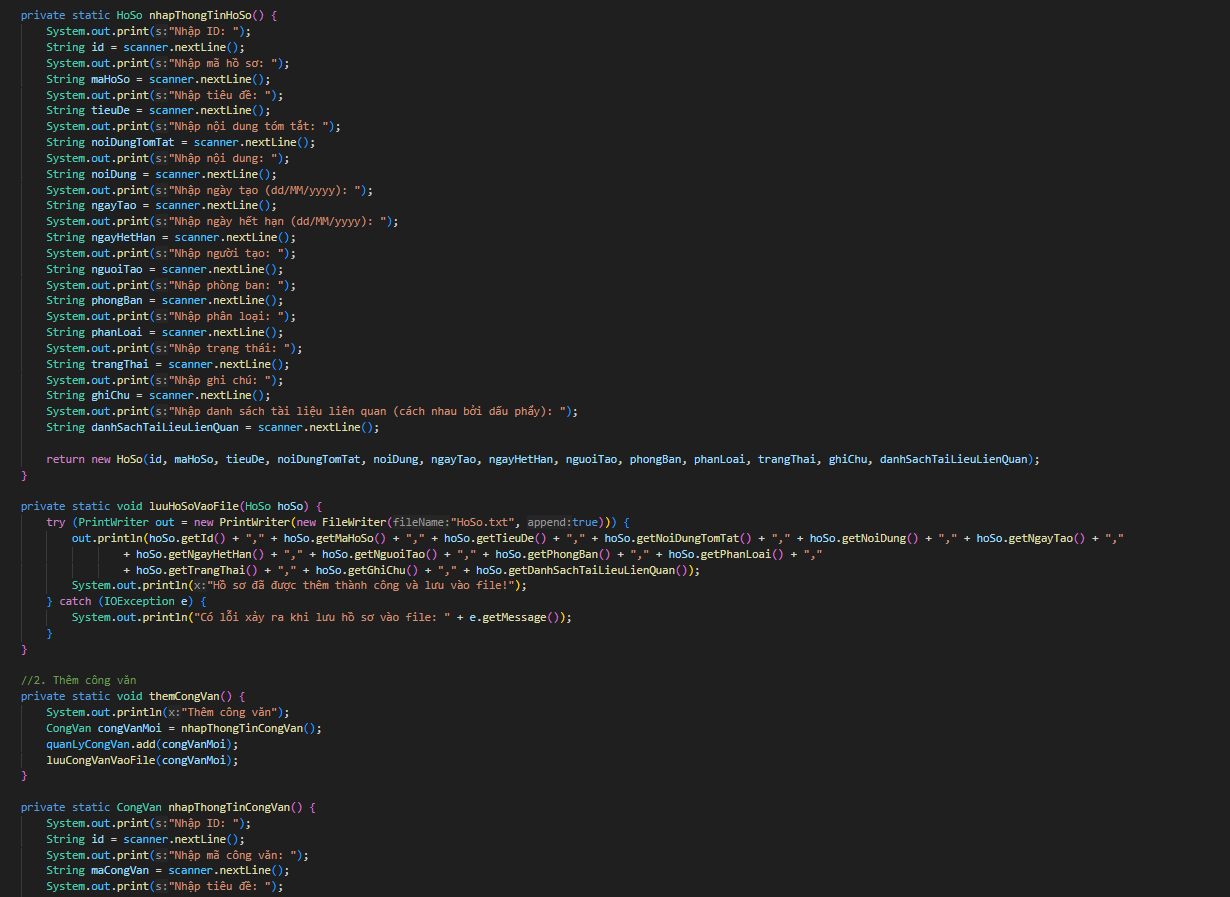
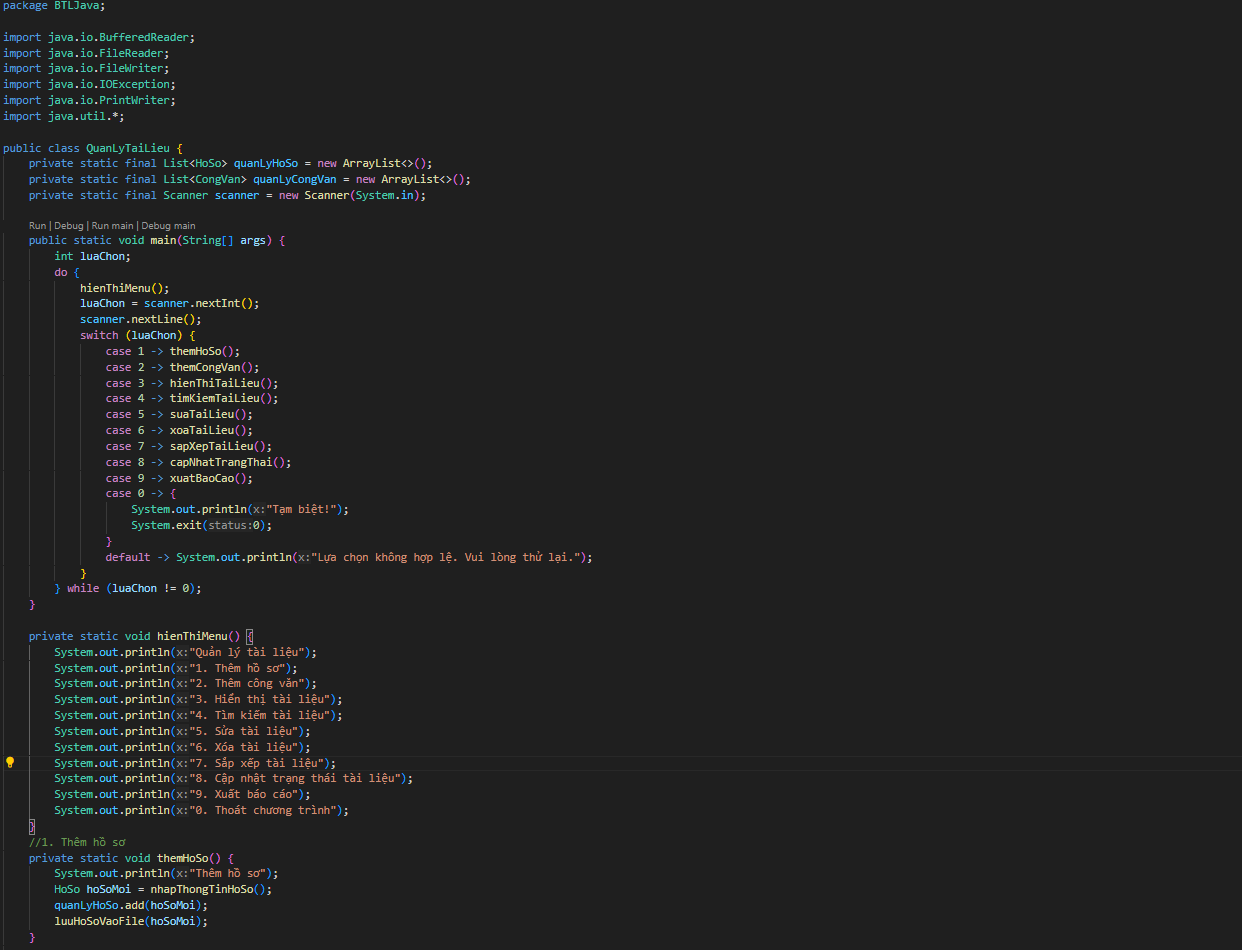
Cho phép người dùng cập nhật trạng thái của tài liệu (ví dụ: đang hoạt động, không hoạt động).

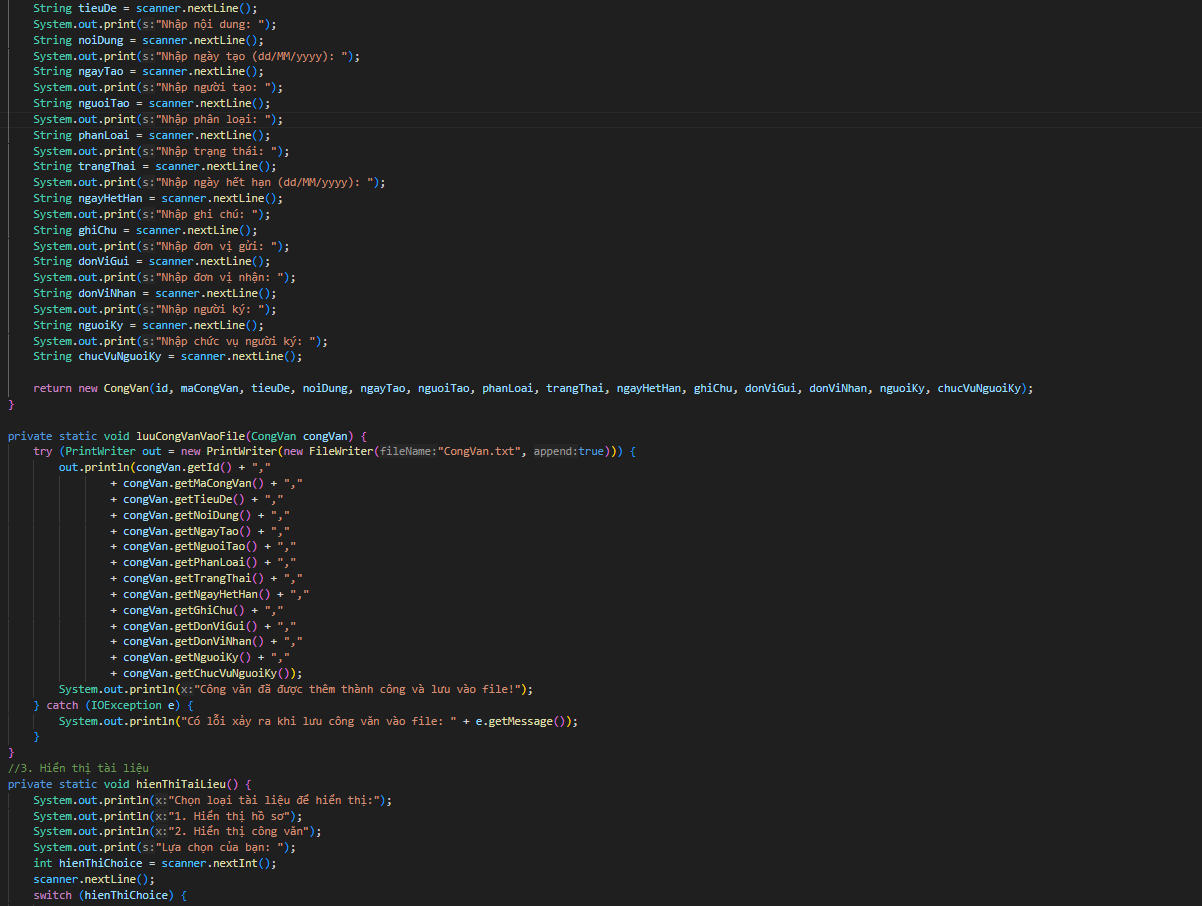
*3.4.10.* xuatBaoCao():

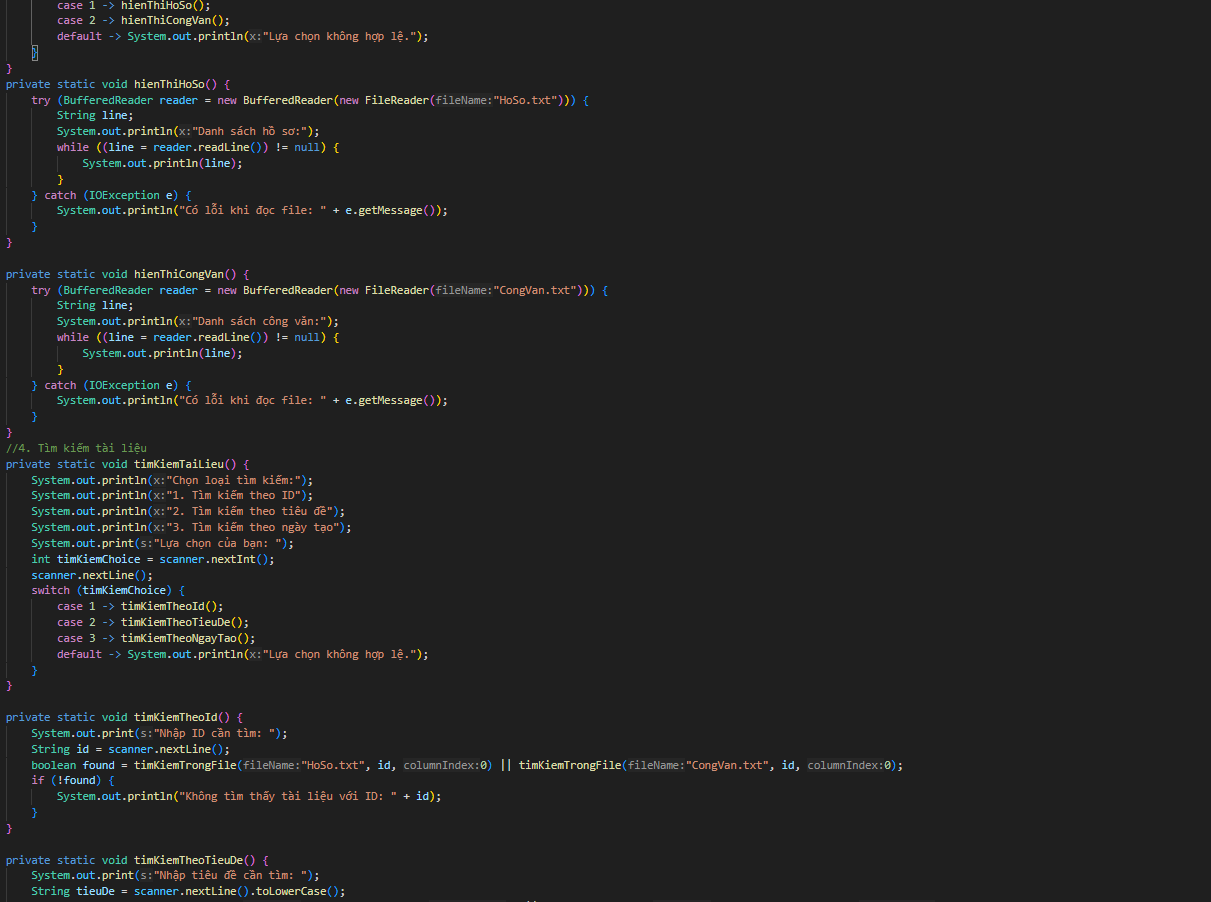
Tạo báo cáo chứa thông tin về tài liệu dựa trên tiêu chí do người dùng xác định.

**CHƯƠNG 4: CÀI ĐẶT BÀI TOÁN**

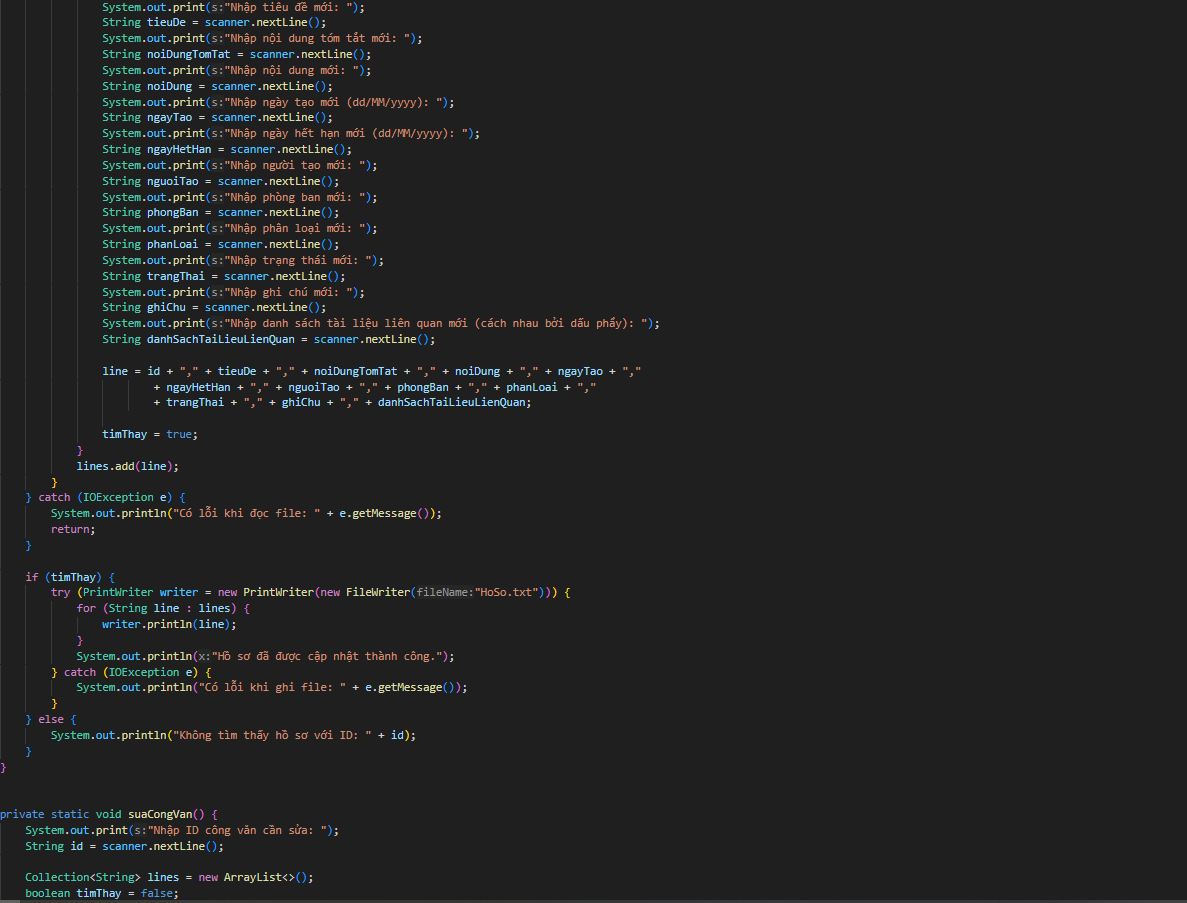
**4.1. Mã nguồn**

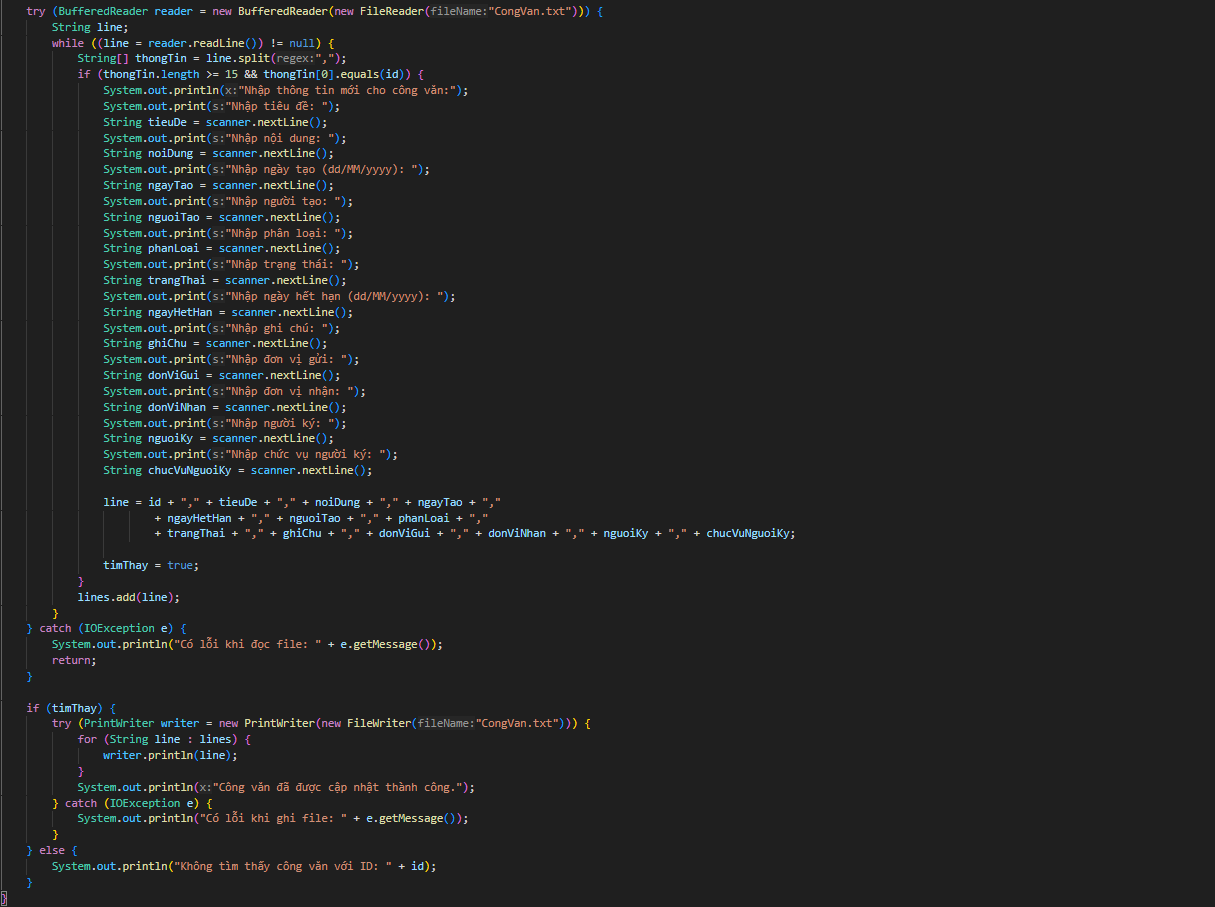


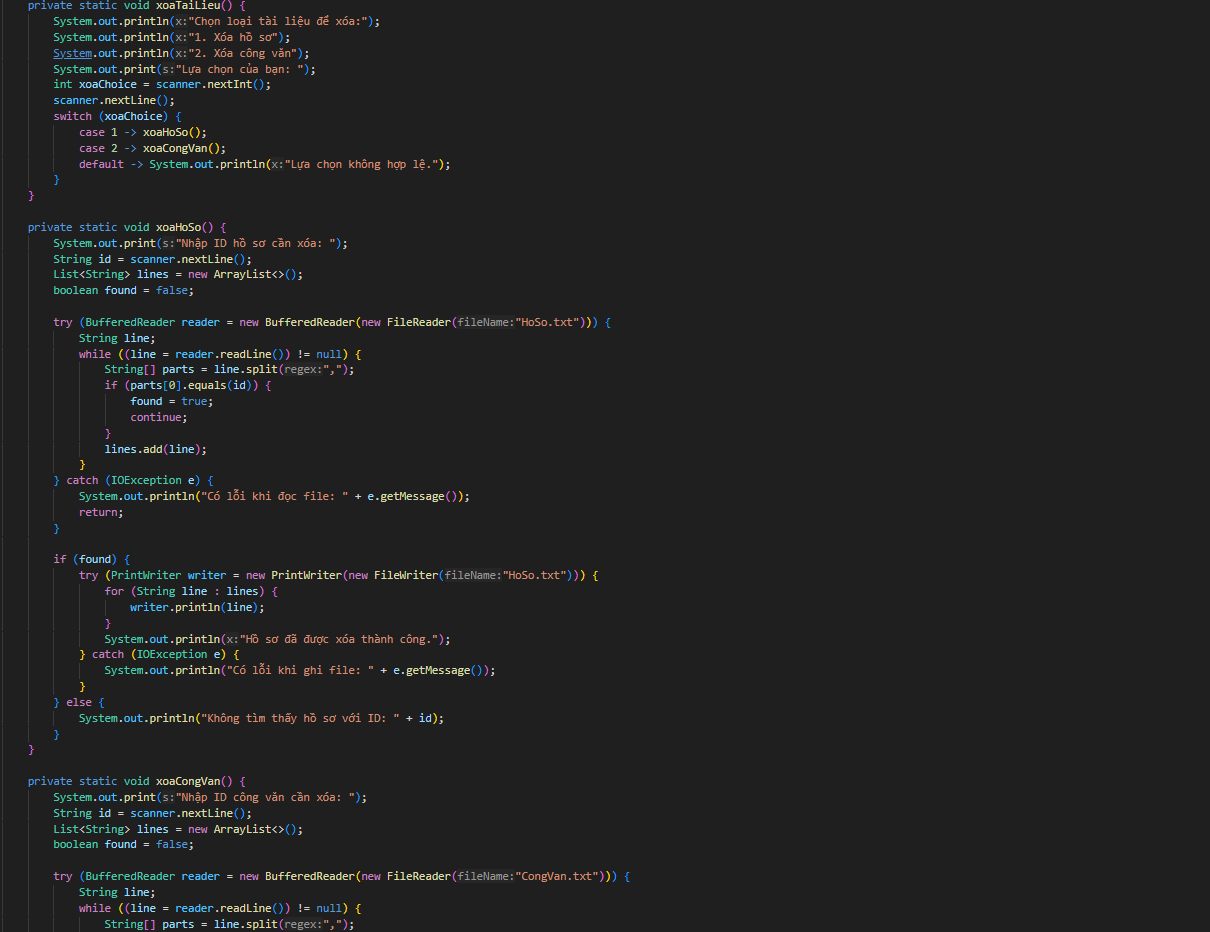


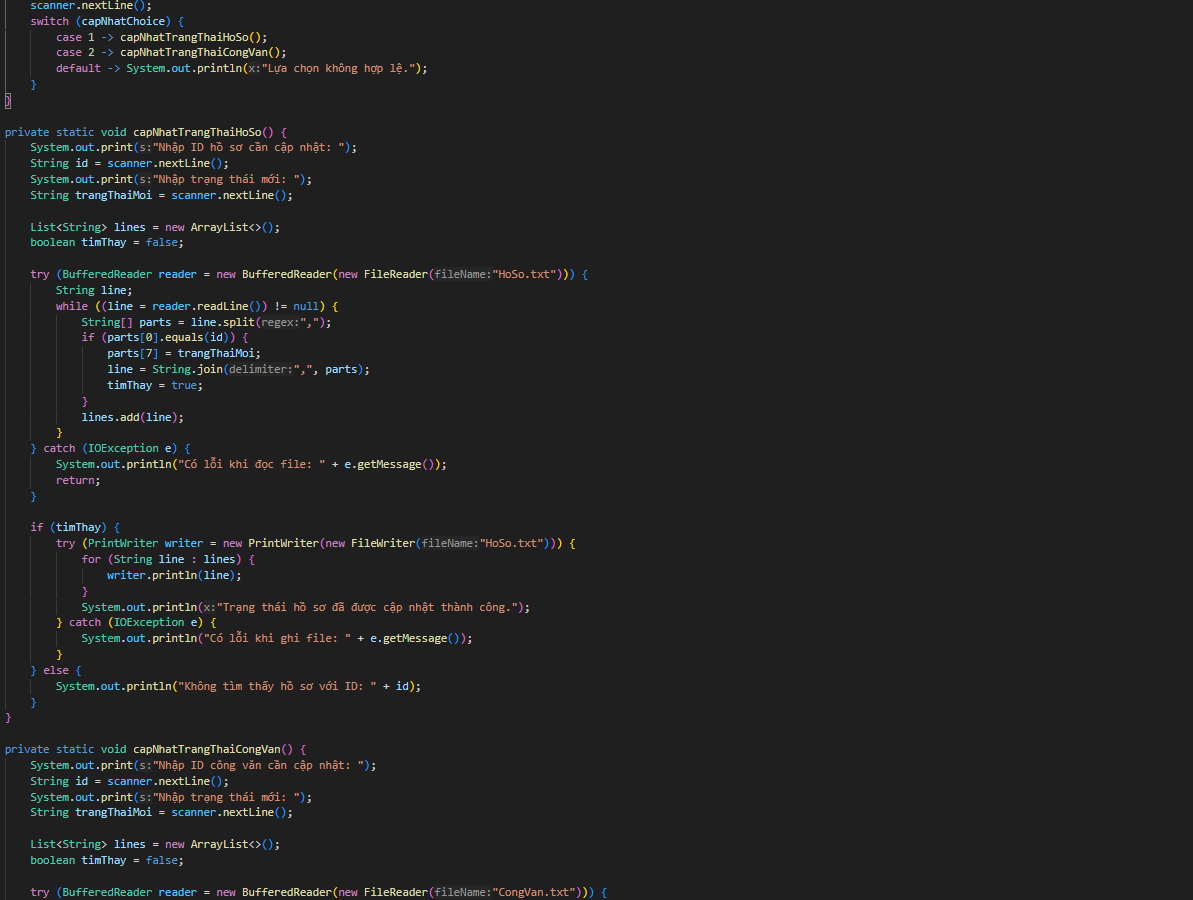
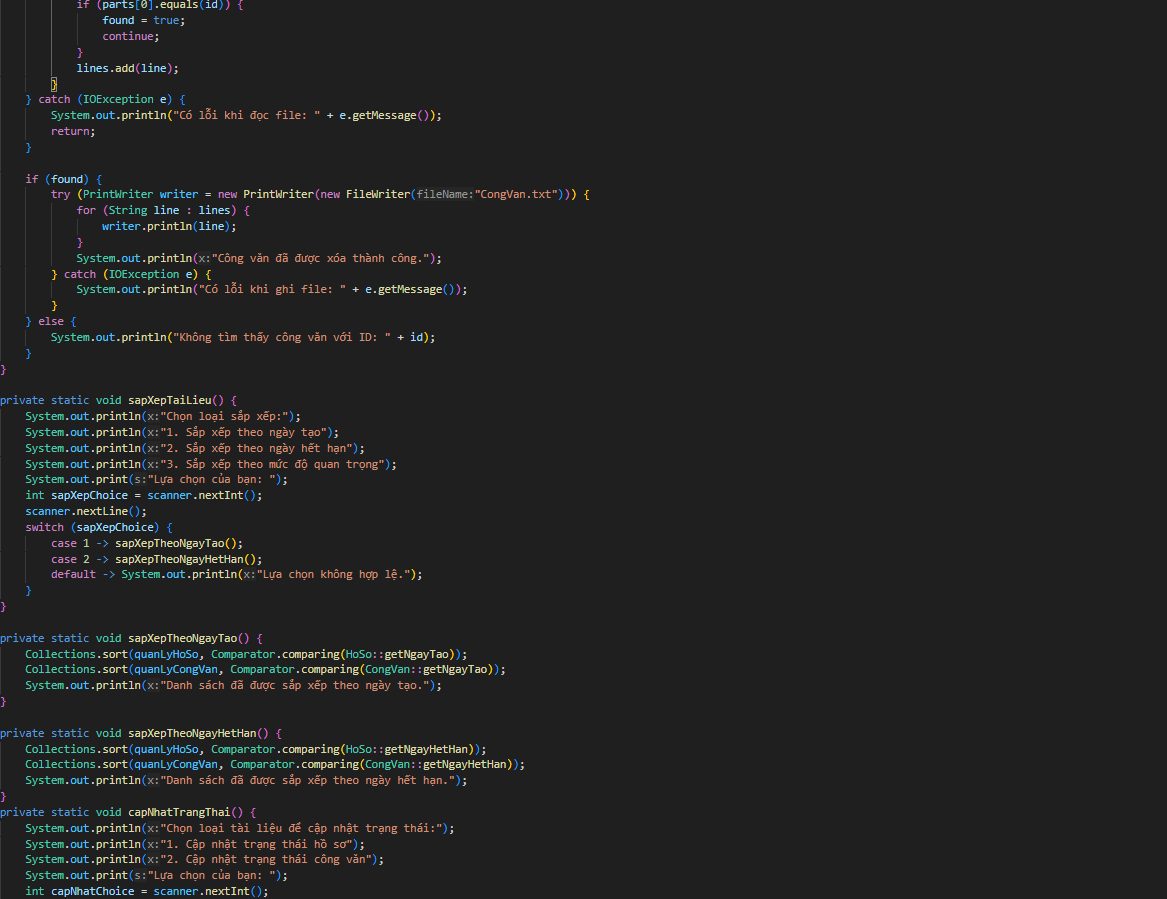
****

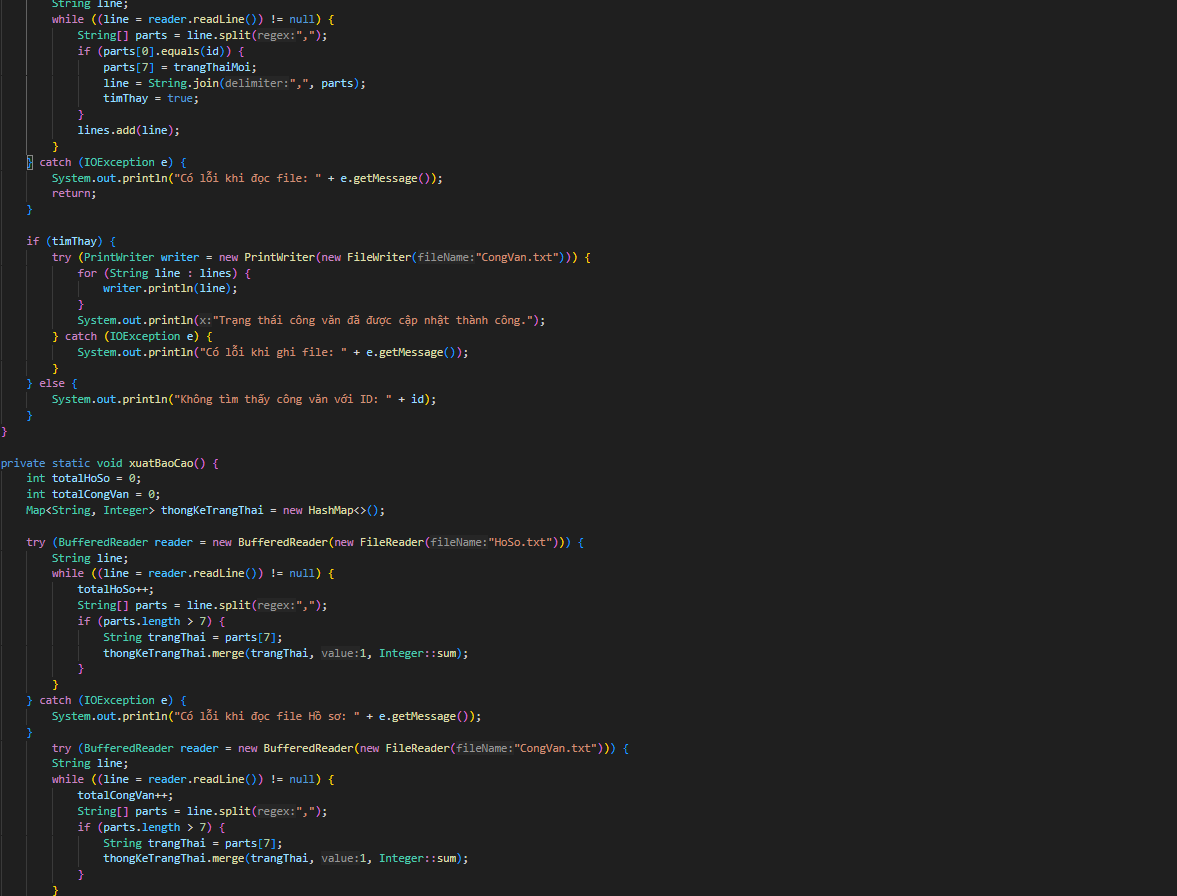
****

****

****

****

****

****

****

**4.2. Tổng quan về mã nguồn**

**4.2.1. Các thư viện được sử dụng**

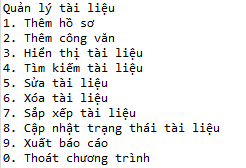
* **java.util.Scanner:** Thư viện này được sử dụng để nhập dữ liệu từ bàn phím.
* **java.io.BufferedReader:** Dùng để đọc dữ liệu từ file.
* **java.io.FileReader:** Tạo một đối tượng để đọc dữ liệu từ một file.
* **java.io.FileWriter:** Tạo một đối tượng để ghi dữ liệu vào một file.
* **java.io.PrintWriter:** Dùng để ghi dữ liệu vào file một cách hiệu quả.
* **java.util.List:** Dùng để lưu trữ danh sách các đối tượng HoSo và CongVan.
* **java.util.ArrayList:** Một triển khai cụ thể của List.
* **java.util.Scanner:** Dùng để nhập dữ liệu từ bàn phím.
* **java.util.Comparator:** Dùng để so sánh các đối tượng và sắp xếp.
* **java.util.Map:** Dùng để lưu trữ các cặp key-value.
* **java.util.HashMap:** Một triển khai cụ thể của Map.

**4.2.2. Mục đích sử dụng của từng thư viện**

* **java.util.Scanner:** Cho phép người dùng tương tác với chương trình bằng cách nhập dữ liệu từ bàn phím.
* **java.io.BufferedReader, FileReader, FileWriter, PrintWriter:** Được sử dụng để đọc và ghi dữ liệu vào các file văn bản. Việc lưu trữ thông tin vào file giúp cho dữ liệu được lưu trữ lâu dài và có thể truy xuất lại sau này.
* **java.util.List, ArrayList:** Dùng để quản lý danh sách các hồ sơ và công văn. Cấu trúc dữ liệu này cho phép thực hiện các thao tác như thêm, xóa, tìm kiếm một cách hiệu quả.
* **java.util.Comparator:** Được sử dụng để sắp xếp các đối tượng HoSo và CongVan theo các tiêu chí khác nhau như ngày tạo, ngày hết hạn.
* **java.util.Map, HashMap:** Dùng để lưu trữ thông tin thống kê về trạng thái của các tài liệu.

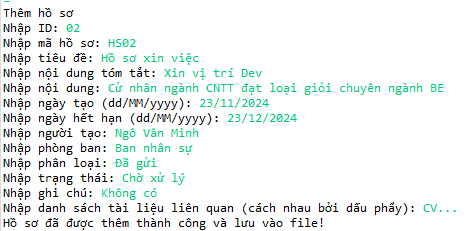
**CHƯƠNG 5: KIỂM THỬ VÀ ĐÁNH GIÁ CHƯƠNG TRÌNH**

**5.1. Menu chương trình quản lý**



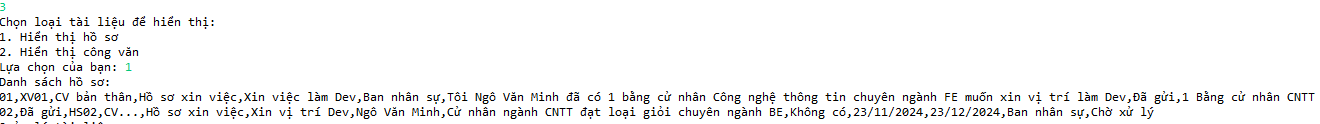
*Hình 5.0: Menu*

**5.2. Thêm tài liệu**

****

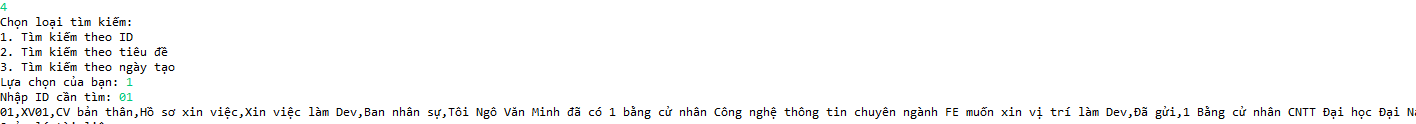
*Hình 5.1: Thêm hồ sơ*

**5.3. Hiển thị tài liệu**



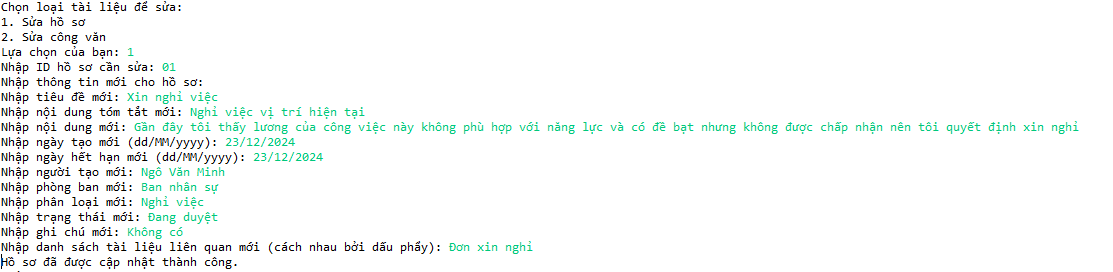
*Hình 5.2: Hiển thị hồ sơ*

**5.4. Tìm kiếm tài liệu**



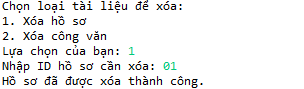
*Hình 5.3: Tìm kiếm hồ sơ*

**5.5. Sửa tài liệu**

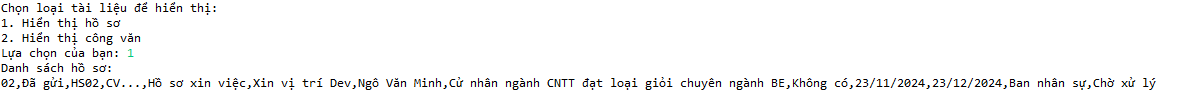
****

*Hình 5.4: Sửa hồ sơ*

**5.6. Xoá tài liệu**

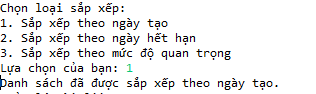
****

*Hình 5.5: Xoá tài liệu*

****

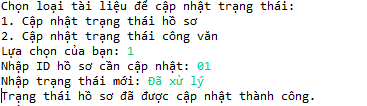
*Hình 5.6: Tài liệu còn lại sau khi đã xoá*

**5.7. Sắp xếp tài liệu**

****

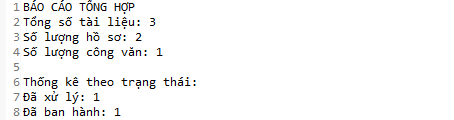
**Hình 5.6: Sắp xếp tài liệu theo ngày tạo**

**5.8. Cập nhật tài liệu**

****

*Hình 5.7: Cập nhật trạng thái hồ sơ*

**5.9. Xuất báo cáo tài liệu**



*Hình 5.8: Xuất báo cáo*

**KẾT LUẬN**

Qua quá trình nghiên cứu và phát triển đề tài "Quản lý hồ sơ, công văn sử dụng hướng đối tượng (OOP) trong Java, chúng tôi rút ra những kết luận quan trọng về ưu điểm, nhược điểm và hướng phát triển của dự án.

1. Ưu Điểm:

Dự án của chúng tôi có một số ưu điểm đáng chú ý. Đầu tiên, việc sử dụng hướng đối tượng (OOP) giúp chúng tôi thiết kế hệ thống một cách cấu trúc, linh hoạt và dễ dàng mở rộng. Điều này tạo điều kiện thuận lợi để thêm các chức năng mới mà không làm thay đổi cấu trúc tổng thể của ứng dụng.

2. Nhược Điểm:

Mặc dù có những ưu điểm, dự án của chúng tôi cũng có nhược điểm. Một trong những nhược điểm chính là hiệu suất. Ứng dụng hiện tại được xây dựng trên nền tảng Java không thích hợp cho các dự án lớn và phức tạp. Việc xử lý dữ liệu lớn có thể gây ra trễ chậm và làm giảm trải nghiệm người dùng. Thứ hai, hệ thống chưa tích hợp cơ sở dữ liệu thực sự, điều này có thể gây ra hạn chế trong việc lưu trữ và truy xuất dữ liệu khi ứng dụng phát triển lớn.

3. Hướng Phát Triển:

Để cải thiện dự án, chúng tôi đề xuất một số hướng phát triển tiềm năng. Đầu tiên, việc chuyển đổi sang một hệ quản trị cơ sở dữ liệu thực sự sẽ giúp chúng tôi quản lý dữ liệu hiệu quả hơn, tăng hiệu suất và giảm trễ chậm. Thứ hai, việc phát triển ứng dụng trên nền tảng web sẽ tăng tính tiện lợi, cho phép người dùng truy cập từ xa và sử dụng trên nhiều thiết bị. Cuối cùng, chúng tôi cũng kế hoạch tối ưu mã nguồn và cải thiện giao diện người dùng để tạo ra một trải nghiệm tốt hơn cho người dùng cuối.

Nhìn chung, dự án của chúng tôi không chỉ giúp chúng tôi áp dụng và cải thiện kỹ năng lập trình hướng đối tượng và phát triển giao diện đồ họa, mà còn đưa ra những hướng phát triển tiềm năng giúp ứng dụng trở nên mạnh mẽ và linh hoạt hơn trong tương lai.

**DANH MỤC TÀI LIỆU THAM KHẢO**

1. Nguyễn Hồng Sơn (2007), *Giáo trình hệ thống Mạng máy tính CCNA* (Semester 1), NXB Lao động xã hội.
2. James F. Kurose and Keith W. Ross (2013), *Computer Networking: A top-down approach sixth Edition*, Pearson Education.
3. NEOS.THÀNH, Lập trình hướng đối tượng Java Core, Đại học FPT, 2017.
4. Y. Daniel Liang, Introduction to Java Programming, Pearson, 2015.
5. Phạm Quốc Hùng (2017), *Đề cương bài giảng Mạng máy tính*, Đại học SPKT Hưng Yên.