**ĐỒ ÁN CÁ NHÂN**

**Môn: Lập trình Hướng Đối Tượng – CS311**

*Nguyen Minh Nhat*

**Lưu ý:**

* *Mỗi SV sẽ bốc xăm ngẫu nhiên các đề từ 1 – 10.*
* *Thời gian thực hiện:*

*+Bắt đầu:13/09/2023 đến 22g00, ngày 24/12/2023*

* *Project được đặt tên:0123\_LEVANTRUONG\_Project\_CS311A và nộp vào Assignments.*

*Trong đó:*

*0123: Mã sinh viên  
LEVANTRUONG:Họ và tên SV*

*Project: Tên Project (Ví dụ: QLTHUVIEN)*

*CS311A: Lớp, nếu là lớp C thì viết: CS311C*

**CHƯƠNG I**

**GIỚI THIỆU BÀI TOÁN**

**`**

**1.1. Giới thiệu bài toán**

Bài 8.

Thư viện trung tâm đại học quốc gia có nhu cầu quản lý việc mượn, trả sách. Sinh viên đăng ký tham gia mượn sách thông qua thẻ mà thư viện cung cấp.

Với mỗi thẻ sẽ lưu các thông tin sau: Mã phiếu mượn, ngày mượn, hạn trả, số hiệu sách, và các thông tin cá nhân của sinh viên mượn sách. Các thông tin của sinh viên mượn sách bao gồm: Họ tên, tuổi, lớp.

Để đơn giản cho ứng dụng console. Chúng ta mặc định ngày mượn, ngày trả là số nguyên dương.

Yêu cầu 1: Xây dựng lớp SinhVien để quản lý thông tin của mỗi sinh viên.

Yêu cầu 2: Xây dựng lớp TheMuon để quản lý việc mượn trả sách của các sinh viên.

Yêu cầu 3: Xây dựng các phương thức: Thêm, xoá theo mã phiếu mượn và hiển thị thông tin các thẻ mượn.

Yêu cầu 4: Thoát chương trình.

Hãy thực hiện:

Thiết kế giao diện GUI để thực hiện được tất cả các chức năng nêu trên

Thiết kế cơ sở dữ liệu phù hợp cho project. Dữ liệu đọc và lưu từ file hoặc từ Hệ Quản Trị CSDL như: MS. Access, SQL Server, MySQL v.v…

**1.2.1. Đầu vào**

**+Cơ sở dữ liệu:**

Lựa chọn Hệ thống file văn bản, gồm các file:

Mã Phiếu Ngày Muợn Ngày Trả Số Hiệu Sách Họ Tên Tuổi Lớp

PM001 21102023 25122023 S001 Nguyen Van A 17 12A2

PM002 19102023 26102023 S002 Tran Thi C 17 12B3

PM003 15112022 27092023 S004 Le Van Teo 17 12A1

PM004 11112023 25012023 S005 Ho Van Cam 16 11C5

PM005 01112023 25012023 S005 Nguyen Van Duc 16 11B3

PM006 12112023 25012023 S005 Tran Van Du 16 11B3

PM007 29112023 25012023 S005 Du Tran An 16 11C5

**1.2.2. Đầu ra**

* Xây dựng các lớp:

*+SinhVien* class.

*+TheMuon* thừa kế *SinhVien* class

* Xây dựng lớp *QuanLyMuonSach* với các phương thức:

+ Thêm mới thông tin các thẻ mượn

+ Xoá theo mã các thẻ mượn

+ Hiển thị thông tin về danh sách các thẻ mượn.

+ Thoát khỏi chương trình

**CHƯƠNG II**

**PHÂN TÍCH VÀ THIẾT KẾ**

* 1. **Phân tích**
     1. **Lựa chọn ngôn ngữ lập trình**
* Ngôn ngữ lập trình: Java
* Công cụ phát triển: NetBean
* Môi trường vận hành: Thực hiện PC, Desktop với hệ điều hành Window X.
  1. **Thiết kế**
     1. **Thiết kế cơ sở dữ liệu**
* Hệ thống lưu trữ: Hệ thống File văn bản (dạng \*.txt)
* Gồm các file có cấu trúc như sau:

Mã Phiếu Ngày Muợn Ngày Trả Số Hiệu Sách Họ Tên Tuổi Lớp

PM001 21102023 25122023 S001 Nguyen Van A 17 12A2

PM002 19102023 26102023 S002 Tran Thi C 17 12B3

PM003 15112022 27092023 S004 Le Van Teo 17 12A1

PM004 11112023 25012023 S005 Ho Van Cam 16 11C5

PM005 01112023 25012023 S005 Nguyen Van Duc 16 11B3

PM006 12112023 25012023 S005 Tran Van Du 16 11B3

PM007 29112023 25012023 S005 Du Tran An 16 11C5

…..

* + 1. **Thiết kế giao diện**
       1. **Giao diện Command Line**

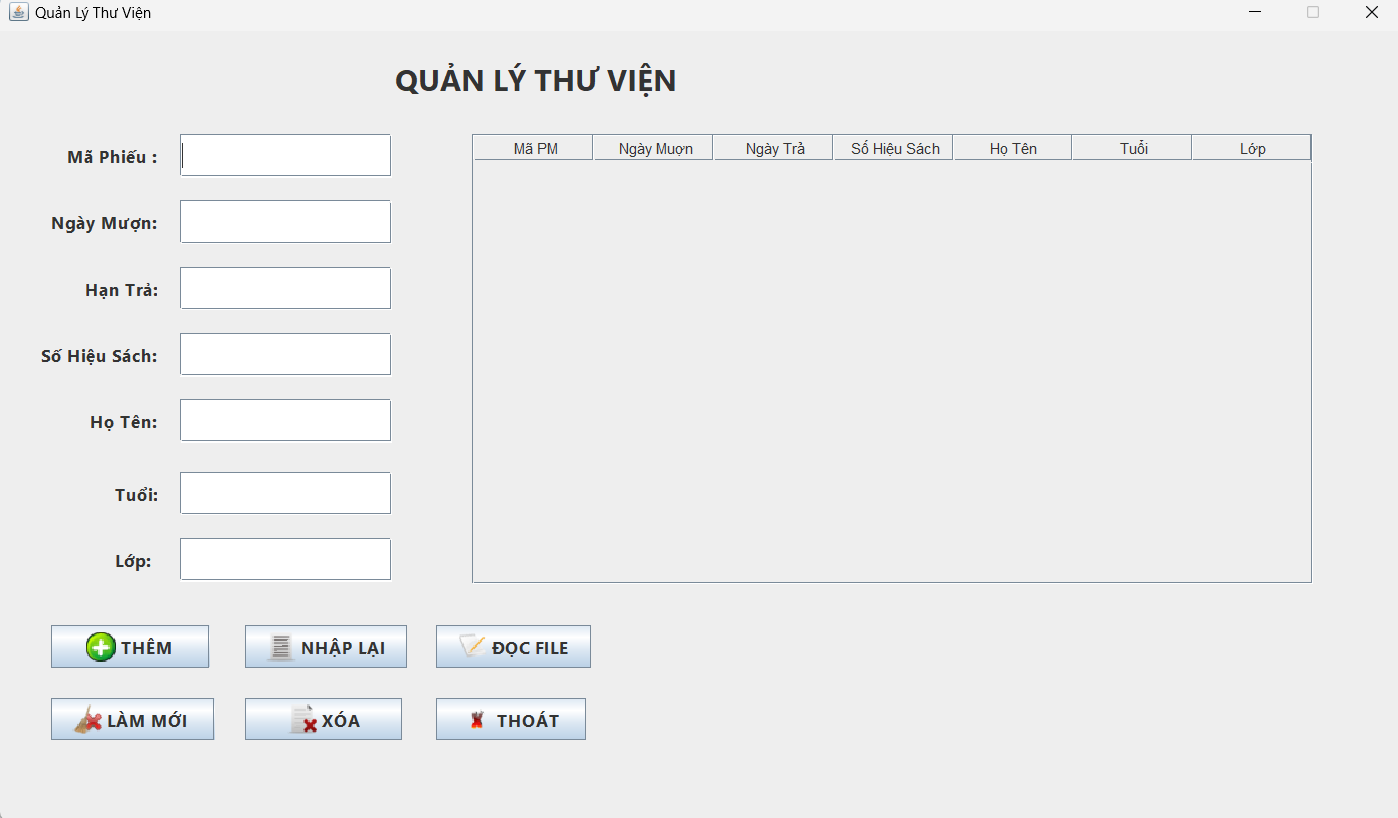
**DANH SÁCH QUẢ N LÝ PHIẾU MƯỢN**

1. Thêm mới phiếu mượn
2. Xoá phiếu mượn có mã phiếu nhập từ bàn phím ra khỏi danh sách
3. Hiển thị thông tin về danh sách các phiếu mượn.
4. Thoát khỏi chương trình

Hãy chọn mục: \_

*Hình 1. Giao diện chinh dạng Command Line*

* + - 1. **Giao diện GUI**

****

*Hình 2. Giao diện chinh dạng đồ hoạ*

* + 1. **Thiết kế lớp**
       1. **Thiết kế lớp**

1. Lớp SinhVien

* Tên lớp: SinhVien.java
* Thuộc tính: *<Copy mã nguồn của chương trình>*

protected String hoTen;

protected int tuoi;

protected String lop;

* Phương thức: *<Copy mã nguồn của chương trình>*

public SinhVien() {

}

public SinhVien(String hoTen, int tuoi, String lop) {

this.hoTen = hoTen;

this.tuoi = tuoi;

this.lop = lop;

}

public String getHoTen() {

return hoTen;

}

public void setHoTen(String hoTen) {

this.hoTen = hoTen;

}

public int getTuoi() {

return tuoi;

}

public void setTuoi(int tuoi) {

this.tuoi = tuoi;

}

public String getLop() {

return lop;

}

public void setLop(String lop) {

this.lop = lop;

}

@Override

public String toString() {

return hoTen + ", " + tuoi + ", " + lop;

}

1. Lớp TheMuon

* Tên lớp: TheMuon.java
* Thuộc tính: *<Copy mã nguồn của chương trình>*

protected String maPhieuMuon ;

protected int ngayMuon , hanTra ;

protected String soHieuSach;

* Phương thức: *<Copy mã nguồn của chương trình>*

public TheMuon() {

}

public TheMuon(String maPhieuMuon, int ngayMuon, int hanTra, String soHieuSach, String hoTen, int tuoi, String lop) {

super(hoTen, tuoi, lop);

this.maPhieuMuon = maPhieuMuon;

this.ngayMuon = ngayMuon;

this.hanTra = hanTra;

this.soHieuSach = soHieuSach;

}

public String getMaPhieuMuon() {

return maPhieuMuon;

}

public void setMaPhieuMuon(String maPhieuMuon) {

this.maPhieuMuon = maPhieuMuon;

}

public int getNgayMuon() {

return ngayMuon;

}

public void setNgayMuon(int ngayMuon) {

this.ngayMuon = ngayMuon;

}

public int getHanTra() {

return hanTra;

}

public void setHanTra(int hanTra) {

this.hanTra = hanTra;

}

public String getSoHieuSach() {

return soHieuSach;

}

public void setSoHieuSach(String soHieuSach) {

this.soHieuSach = soHieuSach;

}

public String toString(){

return super.toString() + ", "+ ", " + maPhieuMuon + ", "+ngayMuon+", "+" "+hanTra+", " +soHieuSach;

}

1. Lớp QuanLyMuonSach

* Tên lớp: QuanLyMuonSach.java
* Thuộc tính: *<Copy mã nguồn của chương trình>*
* Phương thức: *<Copy mã nguồn của chương trình>*

public ArrayList<TheMuon> getA() {

return a;

}

public void setA(ArrayList<TheMuon> a) {

this.a = a;

}

public void docFile(String tenFile) {

try {

File f = new File(tenFile);

if (f.exists()) {

System.out.println("Da mo dc file");

Scanner read = new Scanner(f);

while (read.hasNextLine()) {

Scanner line = new Scanner(read.nextLine());

// Sinh vien

String hoTen = "";

int tuoi;

String lop = "";

//Muon sach

String maPhieuMuon = "";

int ngayMuon, hanTra;

String soHieuSach = "";

maPhieuMuon = line.next();

ngayMuon = line.nextInt();

hanTra = line.nextInt();

soHieuSach = line.next();

while (!line.hasNextInt()) {

hoTen = hoTen + line.next() + " ";

}

tuoi = line.nextInt();

lop = line.next();

a.add(new TheMuon(maPhieuMuon, ngayMuon, hanTra, soHieuSach, hoTen, tuoi, lop));

}

}

} catch (Exception e) {

System.out.println("file ko ton tai");

e.printStackTrace(); // Handle exceptions appropriately in your application

}

}

public void outPut(String cauDan) {

for (TheMuon x : a) {

System.out.println(x);

}

}

public boolean them(TheMuon y) {

for (TheMuon x : a) {

if (x.getMaPhieuMuon().equalsIgnoreCase(y.getMaPhieuMuon())) {

return false;

}

}

a.add(y);

return true;

}

public boolean xoaTheoMa(String ma) {

for (TheMuon x : a) {

if (x.getMaPhieuMuon().equalsIgnoreCase(ma.trim())) {

a.remove(x);

return true; //xoa duoc

}

}

return false; //ko co ma de xoa

}

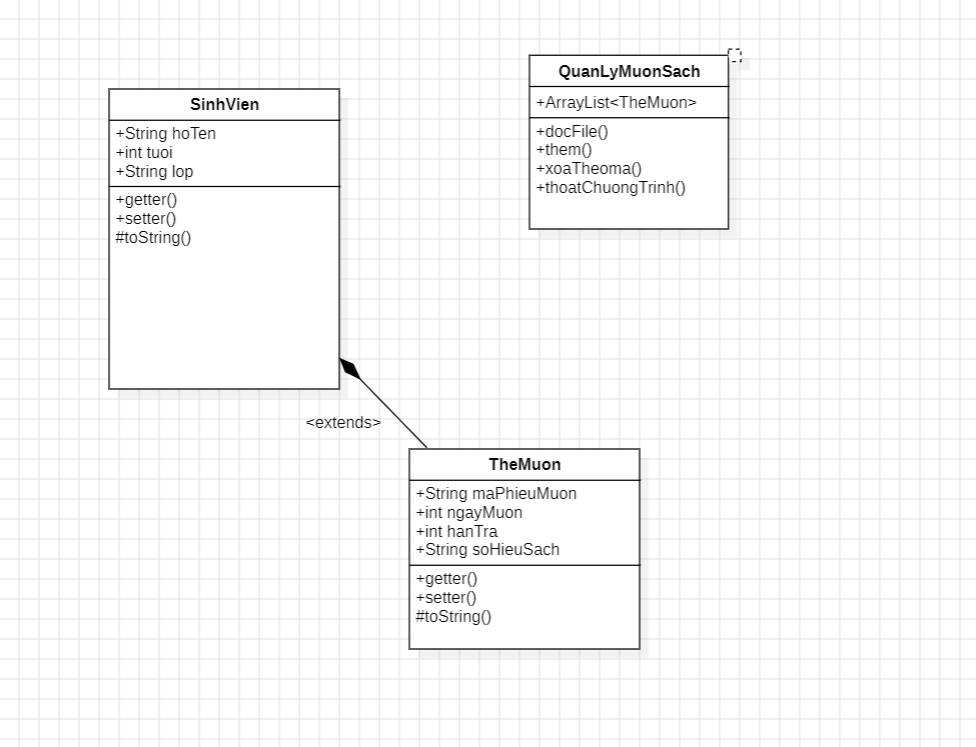
public void thoatChuongTrinh() {

System.exit(0);

}

* + - 1. **Sơ đồ lớp**

Lưu ý: Sử dụng công cụ vẽ sơ đồ lớp (Ví dụ: UML Start, Balsamiq, Figma v.v…)

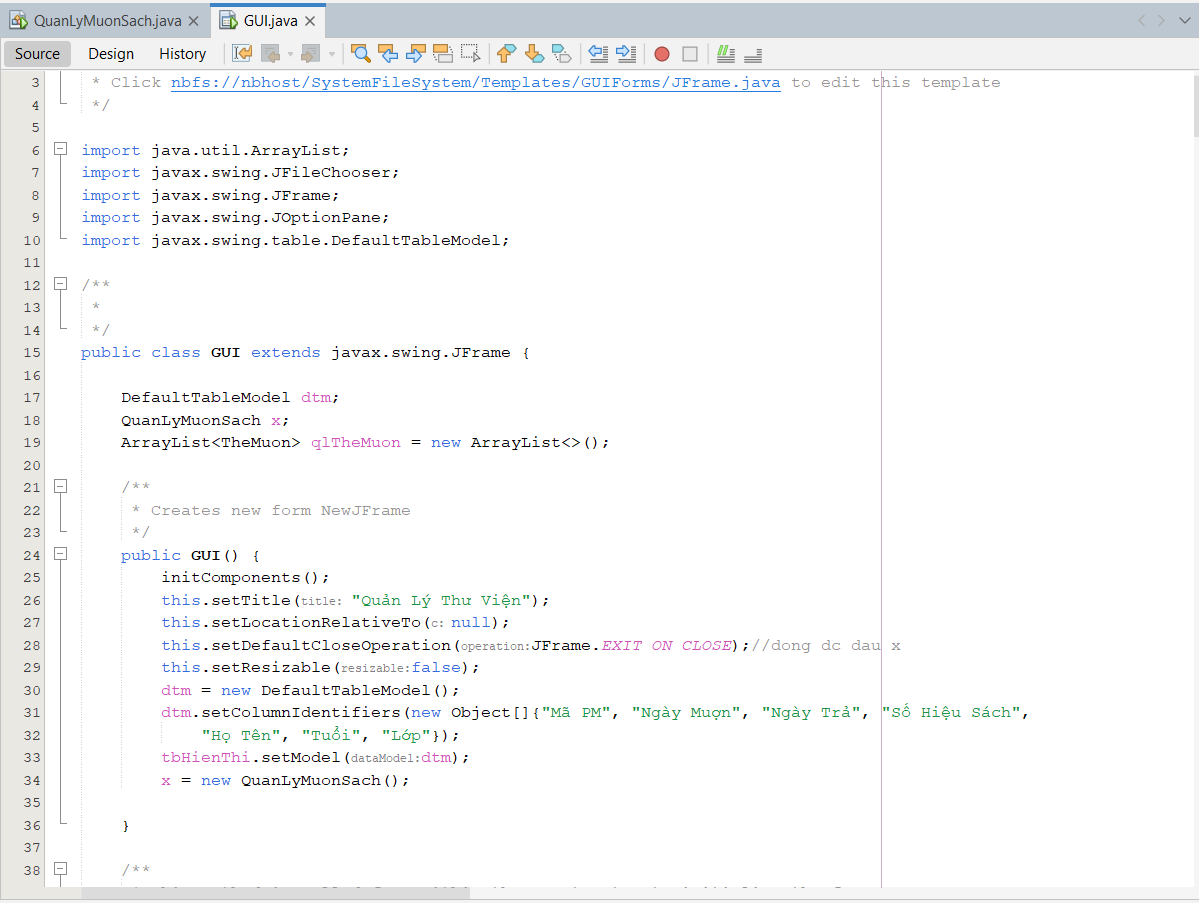


**CHƯƠNG III**

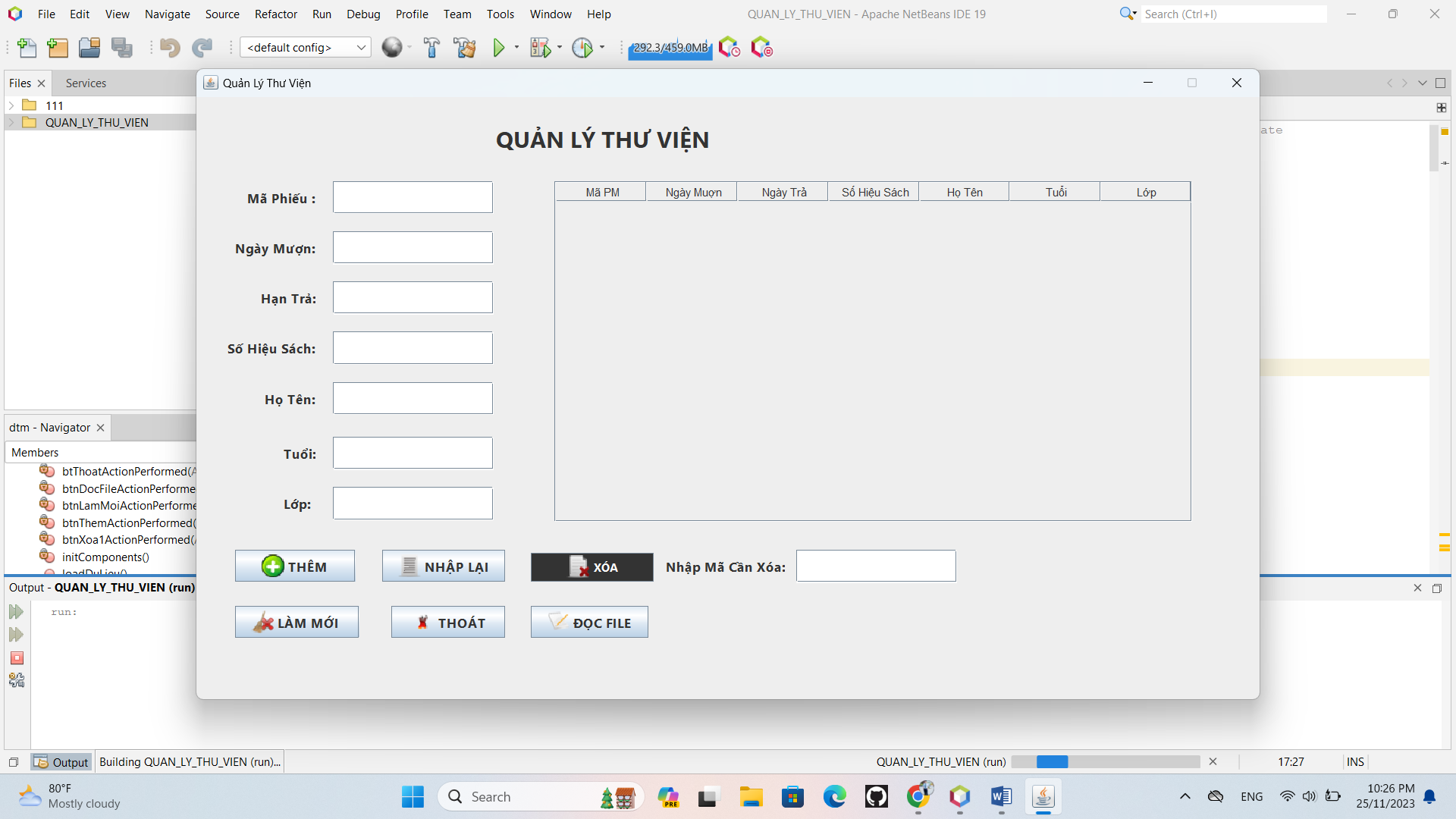
**CÀI ĐẶT VÀ KIỂM THỬ**

**3.1. GIAO DIỆN CHÍNH**

**3.1.1. Giao diện Comman Line**

****

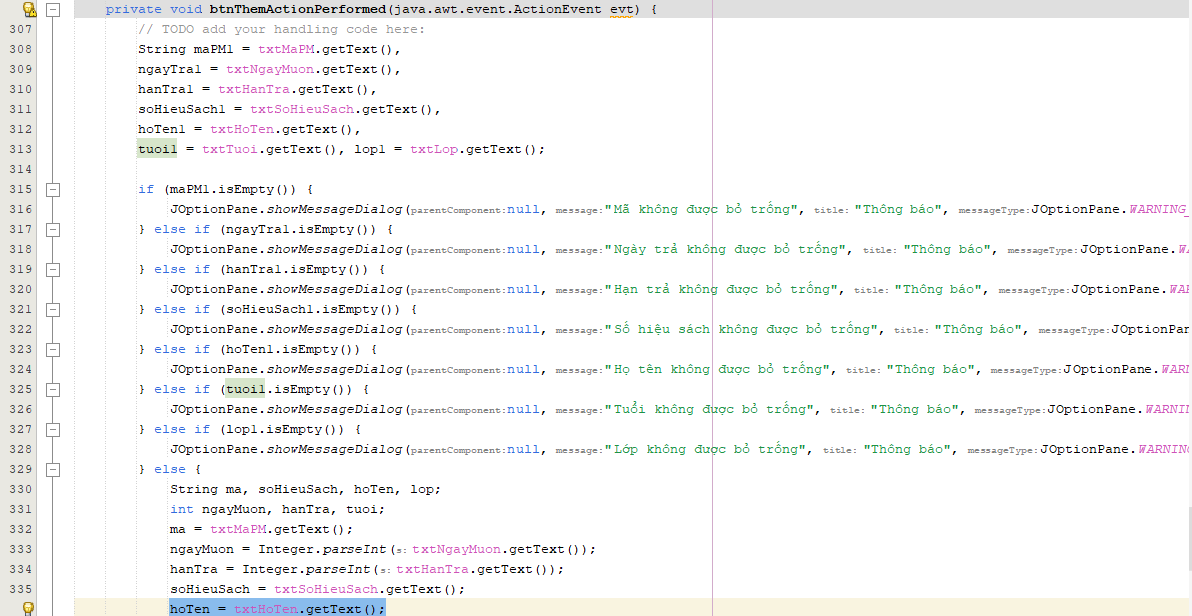
**3.1.1. Giao diện GUI**

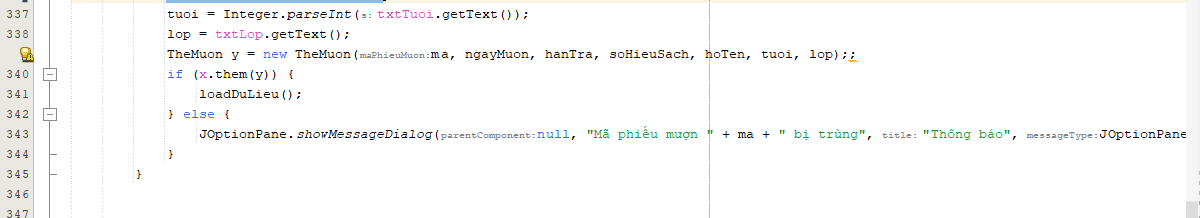


**3.2. Một số kết quả**

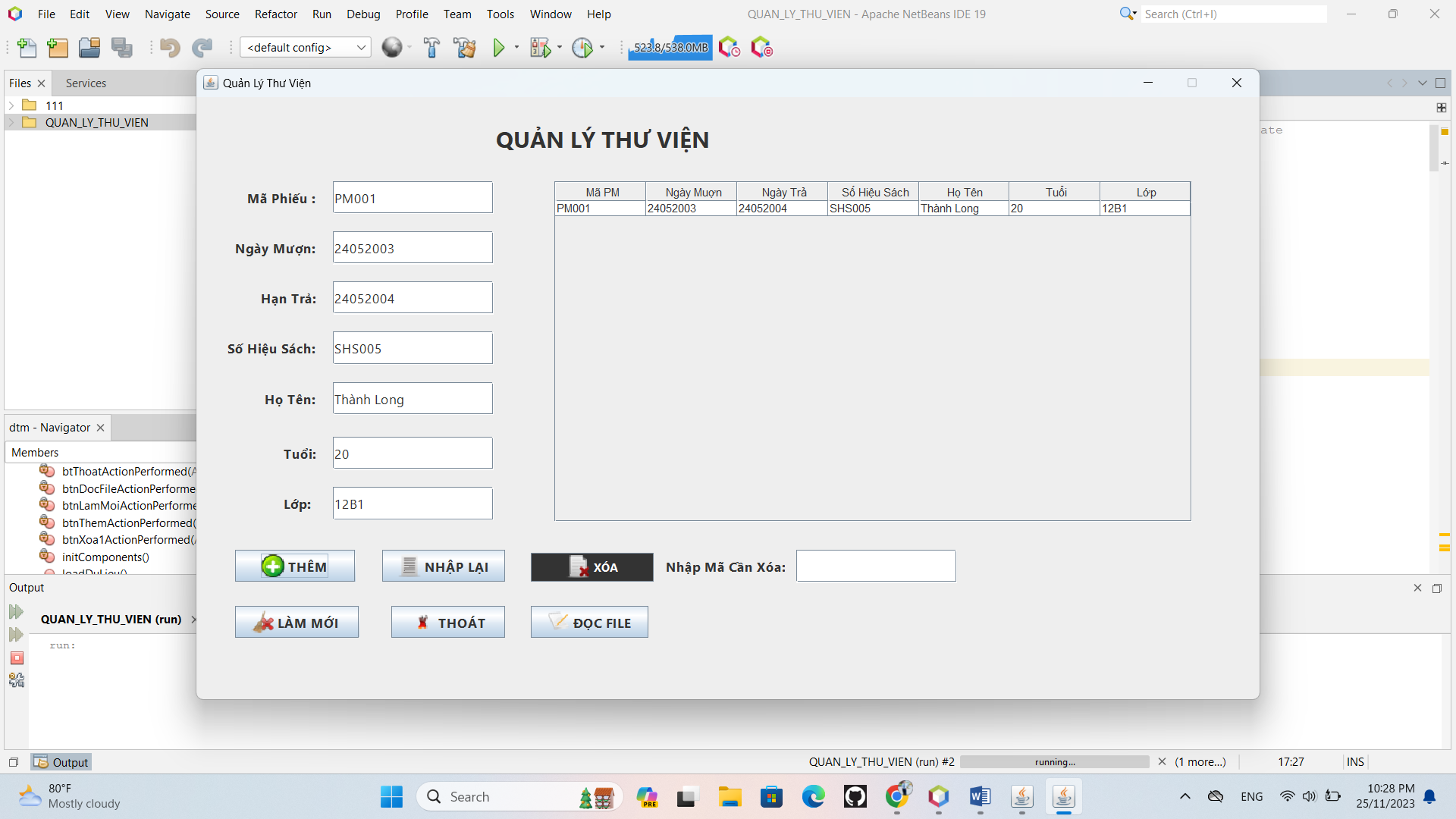
**3.2.1. Nhập danh sách phiếu mượn**

**3.2.1.1. Giao diện Comman Line**

****

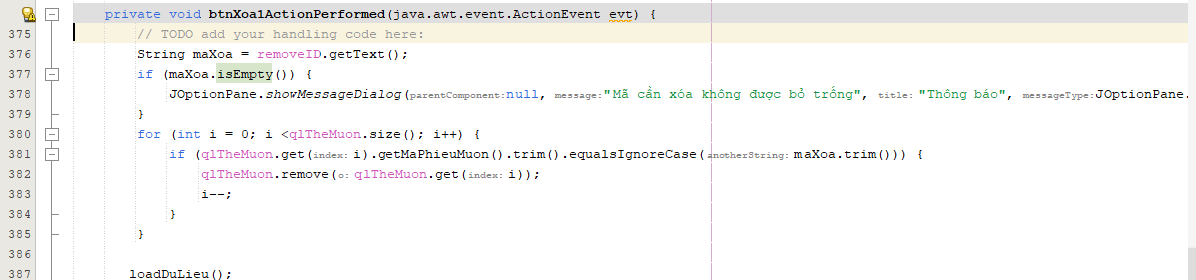
****

**3.2.1.2. Giao diện đồ hoạ**

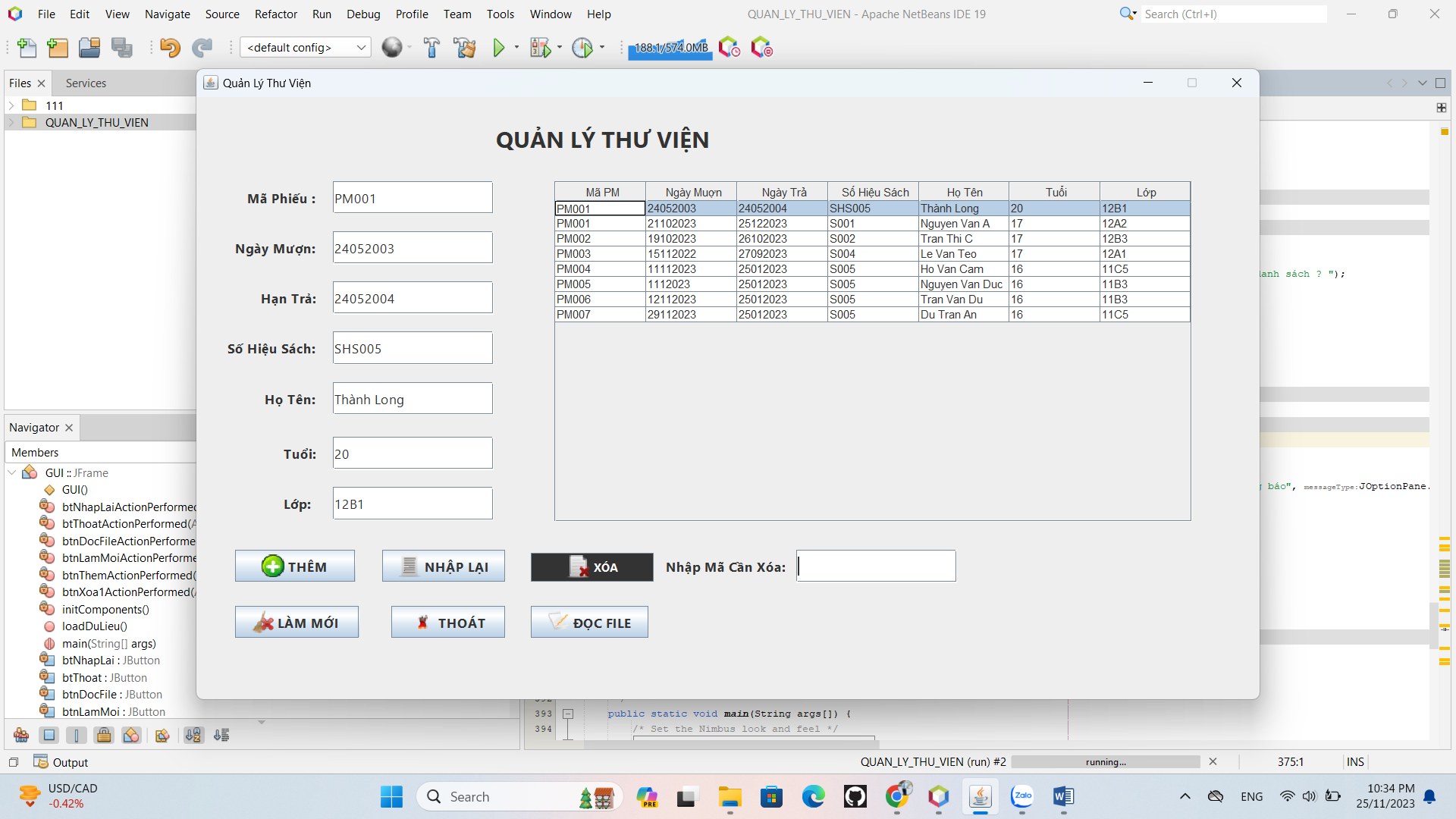


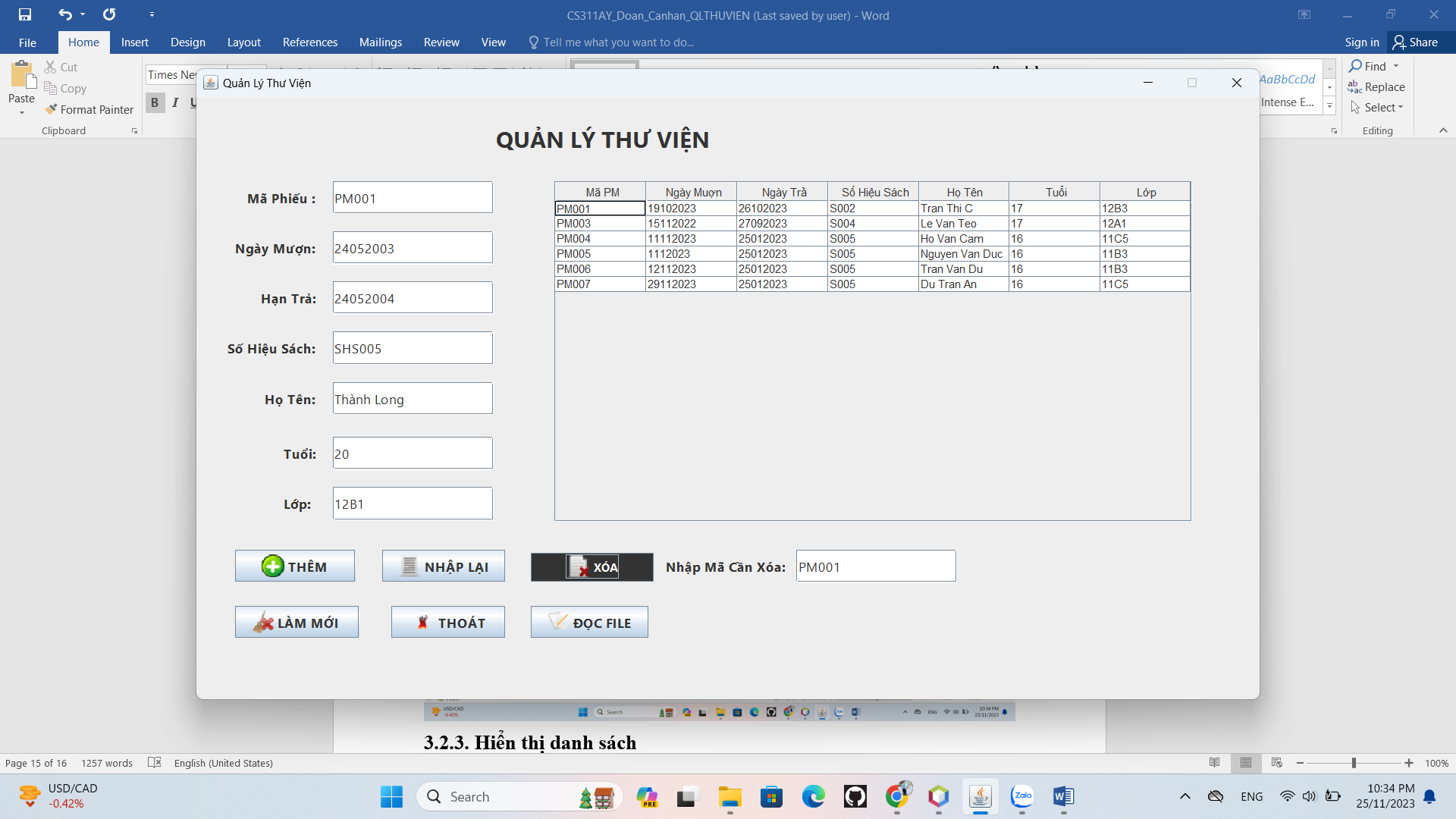
**3.2.2. Xoá**

**3.2.2.1. Giao diện Comman Line**

****

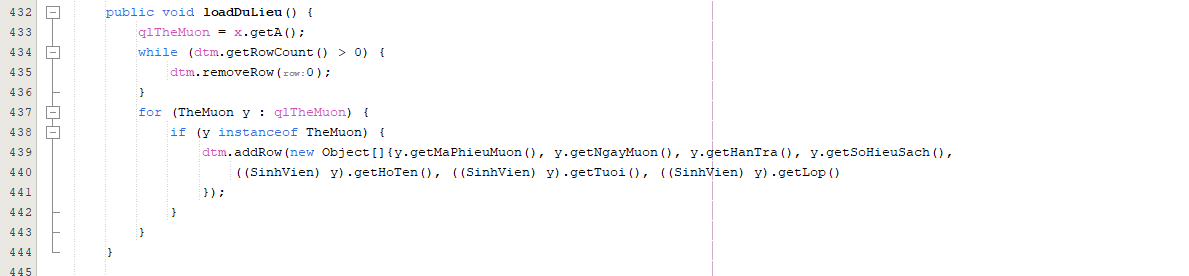
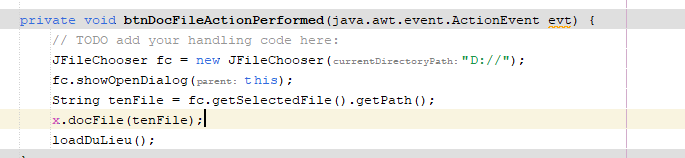
**3.2.2.2. Giao diện đồ hoạ**



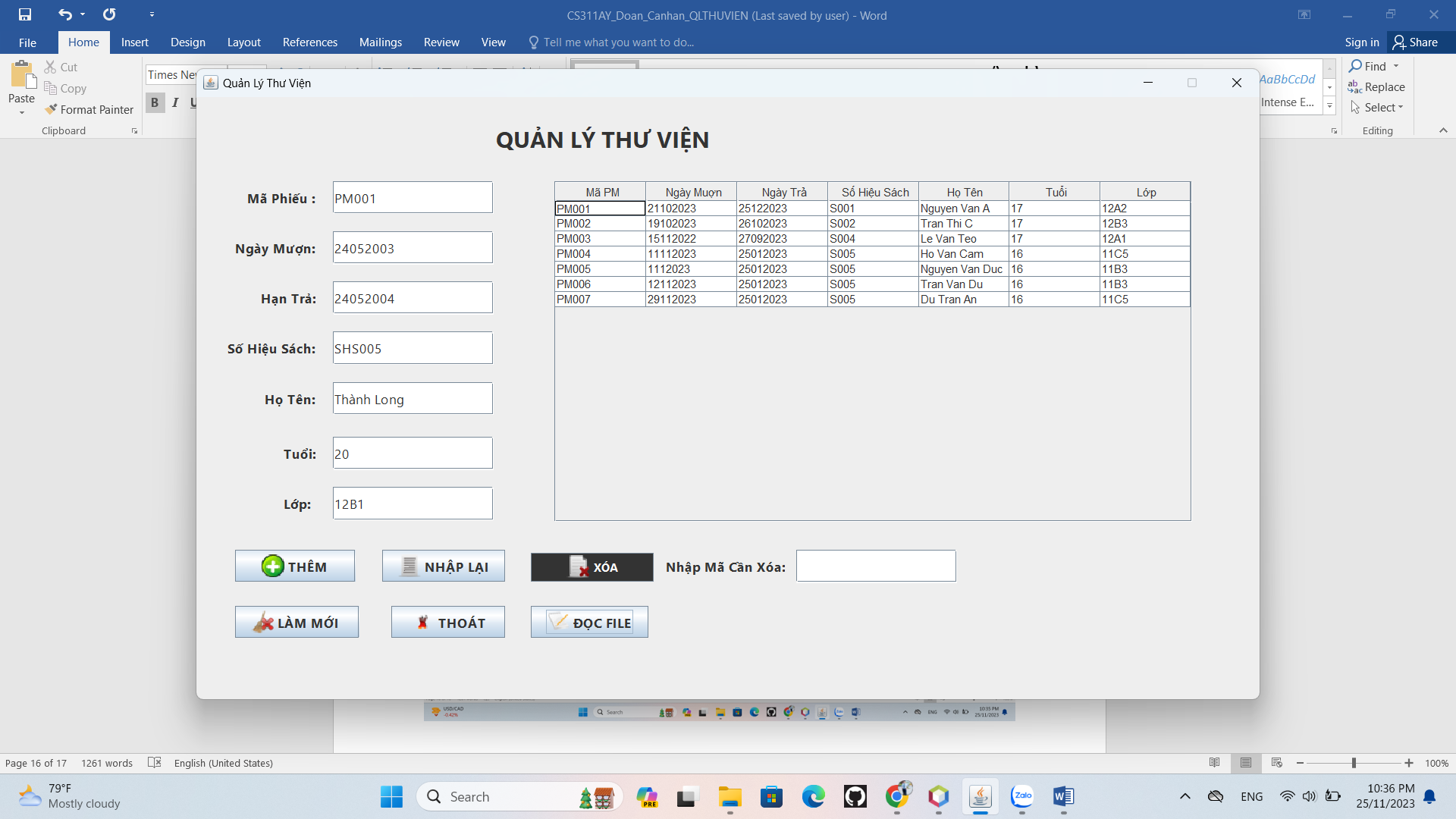
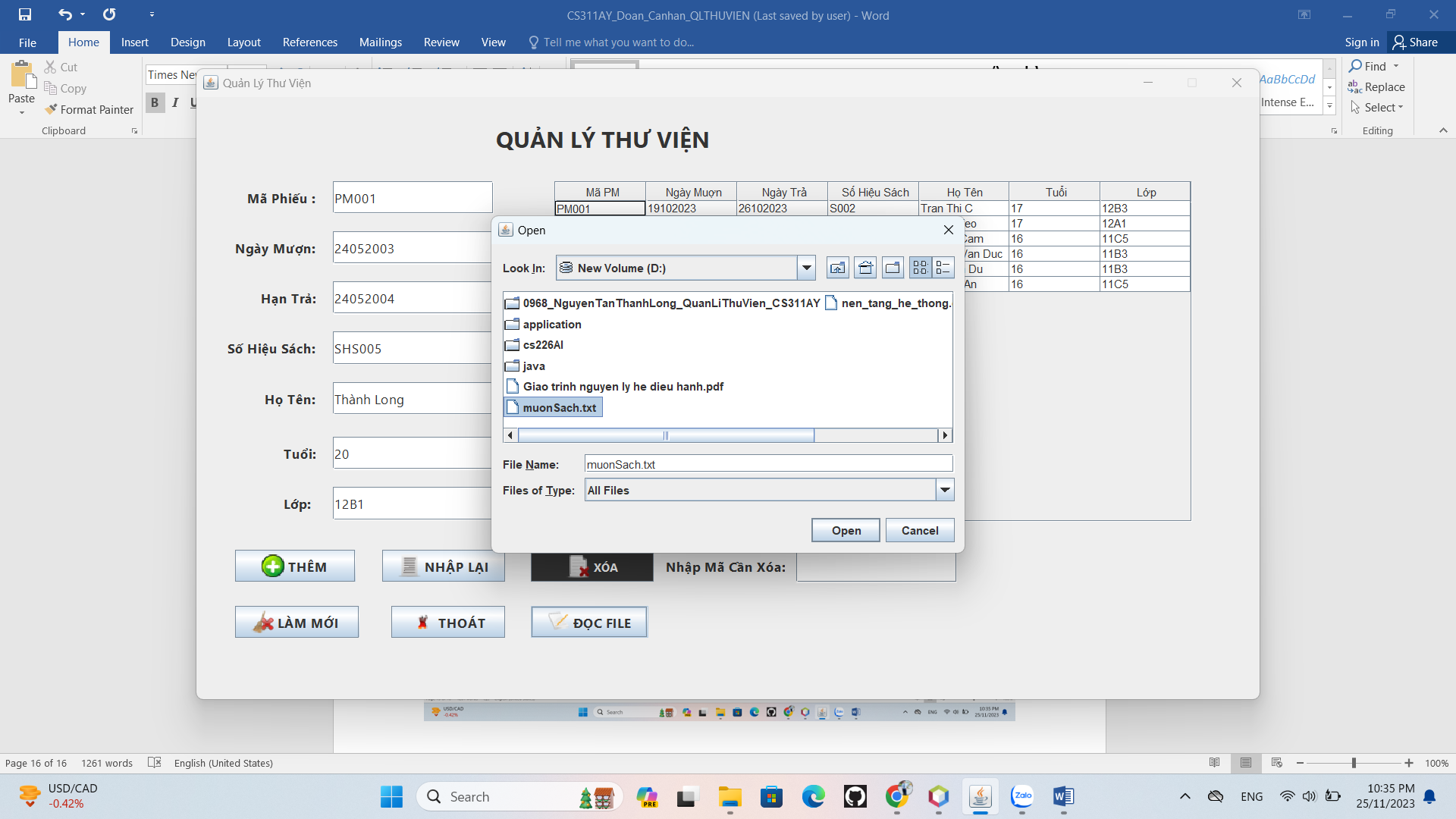
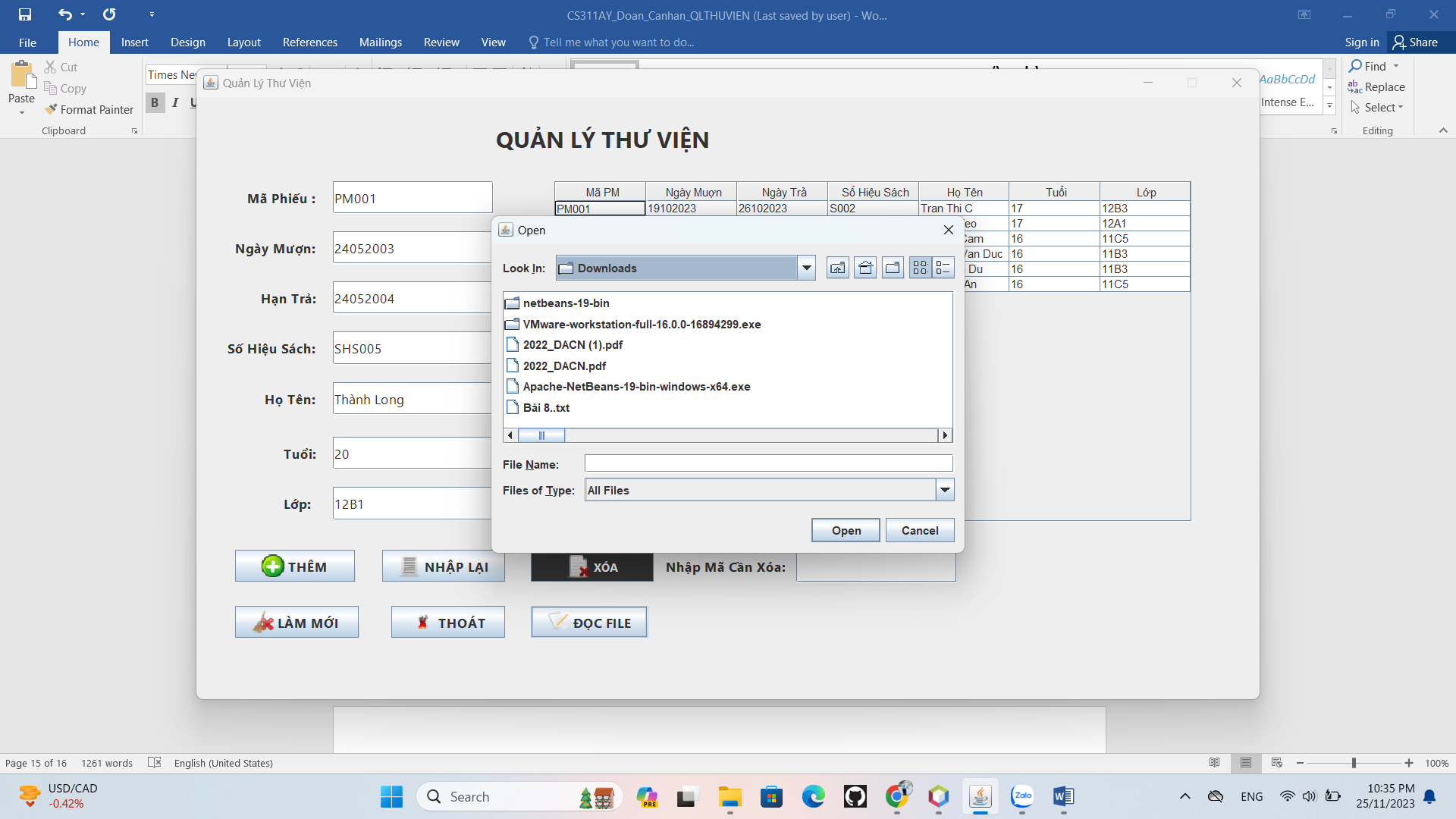


**3.2.3. Hiển thị danh sách ( Đọc File )**

**3.2.3.1. Giao diện Comman Line**

****

**3.2.3.2. Giao diện đồ hoạ**



**KẾT LUẬN**

1. **Các vấn đề đạt được**

* Thêm mới: Bạn đã phát triển chức năng cho phép người dùng thêm mới thông tin các phiếu mượn vào hệ thống. Điều này cho phép người dùng nhập thông tin mới và lưu trữ nó để sử dụng sau này.
* Xóa: Bạn đã xây dựng chức năng xóa, cho phép người dùng xóa thông tin các phiếu mượn khỏi hệ thống. Điều này giúp người dùng có khả năng quản lý danh sách các phiếu mượn và loại bỏ các mục không cần thiết.
* Nhập lại: Bạn đã xây dựng chức năng nhập lại, cho phép người dùng xóa các trường dữ liệu và bắt đầu lại từ đầu. Điều này giúp người dùng có khả năng làm mới thông tin và nhập lại dễ dàng khi cần thiết.
* Thoát: Bạn đã tạo chức năng thoát, cho phép người dùng đóng ứng dụng quản lý thư viện. Điều này cung cấp một cách thoát an toàn và tiện lợi cho người dùng.
* Đưa dữ liệu lên bảng: Bạn đã phát triển chức năng hiển thị dữ liệu thông tin các phiếu mượn trên một bảng. Điều này giúp người dùng có thể dễ dàng xem và quản lý thông tin thư viện trong một giao diện đồ hoạ.
* Làm mới danh sách trong bảng: Bạn đã cho phép người dung reset lại toàn bộ dữ liệu trong bảng

1. **Các vấn đề chưa đạt được**

* Tương tác người dùng tương đồng với giao diện hiện đại: Giao diện của Java Swing có thể trông cũ kỹ và không có ngoại hình chuyên nghiệp như các ứng dụng hiện đại khác. Chưa đạt được việc tạo ra một giao diện người dùng hấp dẫn và tương tác tương đồng với các tiêu chuẩn ngày nay.
* Hiệu suất và tối ưu hóa: Java Swing có thể gặp vấn đề về hiệu suất khi xử lý dữ liệu lớn hoặc thực hiện các tác vụ phức tạp. Chưa đạt được việc cải thiện hiệu suất và tối ưu hóa ứng dụng để đáp ứng yêu cầu xử lý nhanh và mượt mà.
* Đa nền tảng: Java Swing chỉ hỗ trợ chạy trên JVM, giới hạn khả năng ứng dụng chỉ chạy trên các hệ điều hành hỗ trợ JVM như Windows, Linux và macOS. Chưa đạt được việc tạo ra ứng dụng đa nền tảng, có khả năng chạy trên các thiết bị di động và nền tảng khác như Android và iOS.
* Tính năng và giao diện hiện đại: Java Swing thiếu một số tính năng và giao diện hiện đại như hiệu ứng chuyển động mượt mà, giao diện đáp ứng, hỗ trợ cảm ứng và tích hợp dễ dàng với các công nghệ web. Chưa đạt được việc cung cấp trải nghiệm người dùng tương tự như các ứng dụng tiên tiến khác.
* Quản lý mã nguồn và mở rộng: Java Swing không cung cấp cấu trúc dựa trên thành phần rõ ràng và dễ sử dụng như các công nghệ giao diện người dùng khác. Chưa đạt được việc xây dựng mã nguồn dễ bảo trì và mở rộng, điều này có thể làm cho việc bảo trì và phát triển ứng dụng trở nên khó khăn hơn.

1. **Hướng phát triển**

* Thiết kế giao diện người dùng hấp dẫn: Tìm hiểu về các nguyên tắc thiết kế giao diện người dùng hấp dẫn và tương tác tốt. Tối ưu hóa giao diện người dùng, màu sắc, kiểu chữ và cách bố trí để tạo ra trải nghiệm người dùng tốt hơn.
* Tối ưu hóa hiệu suất: Nếu dự án của bạn đang xử lý dữ liệu lớn hoặc có các tác vụ phức tạp, hãy xem xét các cách để tối ưu hóa hiệu suất. Sử dụng các thuật toán tối ưu và phân tích hiệu suất để đảm bảo ứng dụng của bạn hoạt động mượt mà và đáp ứng nhanh chóng.
* Bảo mật dữ liệu: Xác định các vấn đề liên quan đến bảo mật và quyền riêng tư trong ứng dụng quản lý thư viện của bạn. Áp dụng các biện pháp bảo mật như mã hóa dữ liệu và xác thực người dùng để đảm bảo an toàn cho thông tin quan trọng.
* Định dạng và kiểm tra dữ liệu đầu vào: Xử lý đầu vào từ người dùng một cách an toàn và đảm bảo tính nhất quán của dữ liệu. Kiểm tra và xử lý lỗi nhập liệu để tránh các lỗi hoặc xâm nhập dữ liệu không mong muốn.
* Tích hợp cơ sở dữ liệu: Nếu chưa có, xem xét việc tích hợp một cơ sở dữ liệu như MySQL hoặc SQLite để lưu trữ thông tin về thư viện. Sử dụng truy vấn SQL để thực hiện các tác vụ thêm, xóa và tìm kiếm dữ liệu.

**TÀI LIỆU THAM KHẢO**

[1]. https://teamvietdev.com/chia-se-bo-tai-lieu-lap-trinh-java-swing/

[2]. https://codegym.vn/blog/2021/04/09/tai-lieu-lap-trinh-giao-dien-swing-trong-java/

[3]. https://hocjava.com/java-swing-swing-trong-java/