**TRƯỜNG ĐẠI HỌC BÁCH KHOA**

**KHOA ĐIỆN TỬ VIỄN THÔNG**

**THUYẾT MINH PBL1**

**KỸ THUẬT LẬP TRÌNH**

**Sinh viên thực hiện:**

**1.Nguyễn Minh Tuấn Lớp: 23KTMT2 MSSV: 106230066**

**Người hướng dẫn:**

**TS.Nguyễn Duy Nhật Viễn**

**Đà Nẵng , 2024.**

**THUYẾT MINH PBL1.**

**kỹ thuật lập trình**

|  |  |  |  |
| --- | --- | --- | --- |
| STT | HỌ VÀ TÊN | NHIỆM VỤ | KHỐI LƯỢNG |
| 01 | NGUYỄN MINH TUẤN | tìm hiểu dự án code toàn bộ bài ,xây dựng nội dung, tham khảo các nguồn. | 100% |

Mục lục:

1.giới thiệu về tổng quan đề tài.

2.phân tích và thiết kế chương trình.

a. mô hình sử dụng.

b. mô hình hoạt động.

c. mô hình tuần tự.

d.mô hình class.

3. kết luận và đề xuất hướng phát triển trong tương lai.

a. kết luận.

b. đề xuất hướng phát triển trong tương lai.

Tài liệu tham khảo.

4.Phụ lục.

**1**.Giới thiệu tổng quan về đề tài.

* Quản lý thư viện sách là một phần không thể thiếu trong hoạt động vận hành của một thư viện. Điều này đòi hỏi sự tổ chức, quản lý và kiểm soát chặt chẽ về tài nguyên sách, thông tin về độc giả, và các hoạt động liên quan đến việc mượn và trả sách. Trong báo cáo này, chúng ta sẽ đi vào chi tiết về cấu trúc và chức năng của một hệ thống quản lý thư viện sách, cũng như những ưu điểm và thách thức khi triển khai một hệ thống như vậy.

**2**.Phân tích và thiết kế chương trình.

**A**.mô hình sử dụng.

* **Quản lý sách**: Đây là chức năng quản lý thông tin về sách trong thư viện, bao gồm thông tin về tựa đề, tác giả, số lượng, vị trí lưu trữ, và trạng thái của sách (có sẵn, đã mượn, đang được sửa chữa, vv.).
* **Quản lý độc giả**: Chức năng này giúp quản lý thông tin về độc giả, bao gồm thông tin cá nhân, lịch sử mượn sách, và thông tin liên lạc.
* **Quản lý mượn/trả sách**: Hệ thống cần hỗ trợ quản lý việc mượn và trả sách, bao gồm quy trình đăng ký, xác nhận danh tính, và ghi nhận thông tin về việc mượn/trả sách.
* **Tìm kiếm và tra cứu sách**: Đây là chức năng quan trọng giúp độc giả có thể tìm kiếm sách theo tiêu chí khác nhau như tựa đề, tác giả, chủ đề, vv.
* **Thống kê và báo cáo**: Hệ thống cần cung cấp các công cụ thống kê và báo cáo để quản lý có thể đánh giá hiệu suất hoạt động của thư viện, bao gồm số lượng sách được mượn, tình trạng của sách, và các thống kê về độc giả.
* **Quản lý người dùng và phân quyền**: Chức năng này giúp quản lý các tài khoản người dùng và phân quyền truy cập vào hệ thống, đảm bảo rằng chỉ những người được ủy quyền mới có thể thực hiện các thao tác quản lý như thêm sách mới vào hệ thống.
* **Quản lý phiếu phạt và thông báo**: Hệ thống cần hỗ trợ việc ghi nhận các khoản phạt đối với việc trả sách muộn, cũng như gửi thông báo nhắc nhở đến độc giả.

**B**. Mô hình hoạt động.

**- Thêm sách vào thư viện**: Quản trị viên nhập thông tin về sách mới vào hệ thống, bao gồm tựa đề, tác giả, mã vạch, và vị trí lưu trữ.

- **Đăng ký độc giả**: Người dùng mới cần đăng ký tài khoản với thư viện để có thể mượn sách. Quản trị viên xác nhận thông tin và cấp cho họ một tài khoản.

**- Tìm kiếm sách**: Độc giả sử dụng chức năng tìm kiếm để tìm sách theo tiêu chí khác nhau như tựa đề, tác giả, hoặc chủ đề.

**- Mượn sách**: Độc giả chọn sách mình muốn mượn và thực hiện quá trình mượn sách thông qua hệ thống. Hệ thống ghi nhận thông tin về việc mượn sách và cập nhật trạng thái của sách.

**- Trả sách**: Sau khi đọc xong, độc giả trả sách bằng cách nhập mã vạch hoặc thông tin sách vào hệ thống. Hệ thống ghi nhận thông tin về việc trả sách và cập nhật lại trạng thái của sách.

**- Quản lý độc giả và sách**: Quản trị viên có thể quản lý thông tin về độc giả và sách, bao gồm thêm mới, chỉnh sửa, hoặc xóa thông tin.

**- Thống kê và báo cáo**: Hệ thống tự động thu thập dữ liệu về các hoạt động mượn/trả sách và tạo ra các báo cáo thống kê cho quản trị viên, giúp họ đánh giá hiệu suất hoạt động của thư viện.

**- Quản lý người dùng và phân quyền**: Quản trị viên có thể quản lý các tài khoản người dùng và phân quyền truy cập vào hệ thống, đảm bảo rằng chỉ những người được ủy quyền mới có thể thực hiện các thao tác quản lý như thêm sách mới vào hệ thống.

**C.** Mô hình tuần tự.

**- Người dùng đăng ký độc giả hoặc đăng nhập vào hệ thống.**

**- Người dùng tìm kiếm sách trong cơ sở dữ liệu của thư viện.**

**- Người dùng chọn sách cần mượn và thực hiện yêu cầu mượn sách.**

**- Hệ thống kiểm tra số lượng sách có sẵn và trạng thái của sách.**

**- Nếu sách có sẵn, hệ thống ghi nhận việc mượn sách và cập nhật trạng thái.**

**- Người dùng nhận được xác nhận mượn sách và thực hiện việc đến thư viện để lấy sách.**

**- Sau khi đọc xong, người dùng trả sách vào thư viện hoặc qua hệ thống trực tuyến.**

**- Hệ thống ghi nhận việc trả sách và cập nhật trạng thái của sách.**

**- Người dùng có thể tiếp tục tìm kiếm, mượn hoặc trả sách khác.**

**D.** Mô hình class.

**Class** sách

**private:**

* Tựa đề
* Tác giả
* ISBN
* Thể loại
* Số lượng có sẵn
* Vị trí lưu trữ

**Class** người dùng

**private:**

* ID người dùng
* Tên
* Email
* Số điện thoại

**Class** thư viện

**private:**

* Danh sách sách
* Danh sách người dùng

**puclic:**

* Thêm sách
* Xóa sách
* Tìm sách
* Thêm người dùng
* Xóa người dùng
* Cho mượn sách
* Trả sách

**3.** Kết luận và đề xuất hướng phát triển trong tương lai.

**A.** kết luận.

Trong bối cảnh hiện nay, việc phát triển và duy trì một hệ thống quản lý thư viện sách hiệu quả là vô cùng quan trọng để đáp ứng nhu cầu đa dạng của người dùng và nâng cao trải nghiệm của họ. Quản lý thư viện sách không chỉ là việc điều chỉnh về việc tổ chức sách và quản lý người dùng mà còn là việc tối ưu hóa quy trình mượn/trả sách và cung cấp các dịch vụ hỗ trợ thông minh.

**B.**đề xuất hướng phát triển trong tương lai.

- **Ứng Dụng Di Động**: Phát triển ứng dụng di động cho cả hai nền tảng iOS và Android để người dùng có thể truy cập và sử dụng dịch vụ của thư viện từ bất kỳ đâu.

**- Tích Hợp Hệ Thống Đánh Giá và Nhận Xét**: Cho phép người dùng đánh giá và viết nhận xét về sách đã đọc trực tuyến thông qua ứng dụng, giúp người khác đưa ra quyết định mượn sách dựa trên đánh giá từ cộng đồng.

**- Cải Thiện Giao Diện Người Dùng**: Tối ưu hóa giao diện người dùng để tạo ra trải nghiệm người dùng trực quan và dễ sử dụng hơn, đặc biệt là trên các thiết bị di động.

**- Hệ Thống Thông Minh và Tích Hợp Trí Tuệ Nhân Tạo**: Phát triển tính năng gợi ý sách dựa trên lịch sử mượn sách của người dùng và sở thích cá nhân, cũng như dự đoán xu hướng sách nổi bật.

**- Tích Hợp Thẻ Thư Viện Vào Ứng Dụng Di Động**: Cho phép người dùng sử dụng thẻ thư viện điện tử trên ứng dụng di động để tiện lợi hơn trong việc mượn và trả sách.

**- Mở Rộng Kết Nối và Tài Nguyên**: Xây dựng cơ sở dữ liệu lớn hơn bằng cách kết nối với các thư viện khác và các nhà xuất bản để mở rộng tài nguyên sách và dịch vụ.

**- Bảo Mật và Bảo Vệ Dữ Liệu**: Tăng cường bảo mật thông tin cá nhân của người dùng và dữ liệu quản lý thư viện, đảm bảo an toàn và tin cậy trong quá trình sử dụng dịch vụ.

**Tài liệu tham khảo.**

1. <https://cdn.fbsbx.com/v/t59.2708-21/443778612_350235017676835_57962299823867591_n.docx/MauBaoCao-PBL1-MaSV-HoVaTen.docx?_nc_cat=102&ccb=1-7&_nc_sid=2b0e22&_nc_ohc=WHDxvuRrY0UQ7kNvgGjfHnp&_nc_ht=cdn.fbsbx.com&oh=03_Q7cD1QH9UJbFFaomeLzaFBwwhw15m99NQvy8qnpi7gocBBzQeg&oe=665749BB&dl=1>
2. <https://faceworks.vn/chi-tiet/giai-phap-quan-ly-thu-vien-sach-hieu-qua/>

**4.** Phụ lục.

#include <iostream>

#include<string>

using namespace std;

class sach {

private:

string tuade;

string tacgia, ISBN, theloai, vitriluutru;

int soluongcosan;

public:

sach(string td = "", string tg = "", string I = "", string tl = "", int slcs = 0, string vitri = "") :tuade(td), tacgia(tg), ISBN(I), theloai(tl), vitriluutru(vitri), soluongcosan(slcs) {}

string gettuade() const { return tuade; }

string gettacgia() const { return tacgia; }

string getISBN() const { return ISBN; }

string gettheloai() const { return theloai; }

string getvitriluutru() const { return vitriluutru; }

int getsoluongcosan() const { return soluongcosan; }

void setsoluongcosan(int soluong) { soluongcosan = soluong; }

void hienthithongtin() const {

cout << " tua de : " <<tuade << "tac gia" << tacgia << "ISBN" << ISBN << "the loai" <<

theloai << "so luong san co" << soluongcosan << "vi tri luu tru" << vitriluutru << endl;

}

void nhapthongtin() {

cout << "nhap tua de " << endl; getline(cin, tuade);

cout << "nhap ten tac gia" << endl; getline(cin, tacgia);

cout << "nhap ISBN " << endl; getline(cin, ISBN);

cout << "nhap the loai" << endl; getline(cin, theloai);

cout << "nhap so luong co san " << endl; cin >> soluongcosan;

cin.ignore();

cout << "nhap vi tri luu tru" << endl; getline(cin, vitriluutru);

}

};

class nguoidung {

private:

int idnguoidung;

string ten, email, sodienthoai;

public:

nguoidung(int i = 0, string Ten = "", string Email = "", string Sodienthoai = "")

:idnguoidung(i), ten(Ten), email(Email), sodienthoai(Sodienthoai) {}

int getidnguoidung() const { return idnguoidung; }

string getten() const { return ten; }

string getemail() const { return email; }

string getsodienthoai() const { return sodienthoai; }

void hienthithongtin() const {

cout << "id nguoi dung" << idnguoidung << "ten nguoi dung" << ten << "email nguoi dung" << email << "so dien thoai nguoi dung"

<< sodienthoai << endl;

}

void nhapthongtin() {

cout << "nhap id nguoi dung" << endl; cin >> idnguoidung;

cin.ignore();

cout << "nhap ten nguoi dung" << endl; getline(cin, ten);

cout << "nhap email nguoi dung" << endl; getline(cin, email);

cout << "nhap so dien thoai nguoi dung" << endl; getline(cin, sodienthoai);

}

};

class thuvien {

private:

sach\* danhsachsach;

int soluongsach;

nguoidung\* danhsachnguoidung;

int soluongnguoidung;

public:

thuvien() :danhsachsach(0), soluongsach(0), danhsachnguoidung(0), soluongnguoidung(0) {}

~thuvien() {

delete[] danhsachsach;

delete[] danhsachnguoidung;

}

void themsach() {

sach Sach;

Sach.nhapthongtin();

sach\* temp = new sach[soluongsach + 1];

for (int i = 0; i < soluongsach; i++) {

temp[i] = danhsachsach[i];

}

temp[soluongsach] = Sach;

delete[]danhsachsach;

danhsachsach = temp;

soluongsach++;

}

void xoasach(const string& ISBN) {

int index = -1;

for (int i = 0; i < soluongsach; i++) {

if (danhsachsach[i].getISBN() == ISBN) {

index = i;

break;

}

}

if (index == -1) {

cout << "khong tim thay sach voi ISBN" << endl;

return;

}

sach\* temp = new sach[soluongsach - 1];

for (int i = 0, j = 0; i < soluongsach; ++i) {

if (i != index) {

temp[j++] = danhsachsach[i];

}

}

delete[] danhsachsach;

danhsachsach = temp;

soluongsach--;

cout << "sach da xoa " << ISBN << endl;

}

sach\* timsach(const string& ISBN) {

for (int i = 0; i < soluongsach; i++) {

if (danhsachsach[i].getISBN() == ISBN) {

return &danhsachsach[i];

}

}

return 0;

}

void themnguoidung() {

nguoidung Nguoidung;

Nguoidung.nhapthongtin();

nguoidung\* temp = new nguoidung[soluongnguoidung + 1];

for(int i = 0; i < soluongnguoidung; i++){

temp[i] = danhsachnguoidung[i];

}

temp[soluongnguoidung] = Nguoidung;

delete[] danhsachnguoidung;

danhsachnguoidung = temp;

soluongnguoidung++;

}

void xoanguoidung(int id) {

int index = -1;

for (int i = 0; i < soluongnguoidung; i++) {

if (danhsachnguoidung[i].getidnguoidung() == id) {

index = i;

break;

}

}

if (index == -1) {

cout << "khong tim thay thong tin nguoi dung voi ID" << endl;

return;

}

nguoidung\* temp = new nguoidung[soluongnguoidung - 1];

for (int i = 0,j=0; i < soluongnguoidung; i++) {

if (i != index) {

temp[j++] = danhsachnguoidung[i];

}

}

delete danhsachnguoidung;

danhsachnguoidung = temp; soluongnguoidung--;

cout << "da xoa nguoi dung voi id" << id << endl;

}

void chomuonsach(const string& ISBN) {

sach\* Sach = timsach(ISBN);

if (Sach != 0) {

Sach->setsoluongcosan(Sach->getsoluongcosan() + 1);

cout << "da tra sach voi ISBN" << ISBN << endl;

}

else {

cout << "khong tim thay sach voi ISBN" << ISBN << endl;

}

}

void trasach(const string& ISBN) {

sach\* Sach = timsach(ISBN);

if (Sach != 0) {

Sach->setsoluongcosan(Sach->getsoluongcosan() + 1);

cout << "da tra sach voi ISBN" << ISBN << endl;

}

else {

cout << "khong tim thay sach voi ISBN" << ISBN << endl;

}

}

void hienthisach()const {

for (int i = 0; i < soluongsach; i++) {

danhsachsach[i].hienthithongtin();

cout << endl;

}

}

void hienthinguoidung()const {

for (int i = 0; i < soluongnguoidung; i++) {

danhsachnguoidung[i].hienthithongtin();

}

}

};

int main()

{

thuvien ThuVien;

cout << "nhap thong tin sach" << endl;

ThuVien.themsach();

ThuVien.themsach();

cout << "nhap thong tin nguoi dung" << endl;

ThuVien.themsach();

ThuVien.themsach();

cout << "danh sach sach trong thu vien" << endl;

ThuVien.hienthinguoidung();

string ISBN;

cout << "nhap ISBN sach muon muon" << endl;

getline(cin, ISBN);

ThuVien.chomuonsach(ISBN);

cout << "danh sach sau khi muon 1 cuon" << endl;

cout << "nhap ISBN muon tra" << endl;

getline(cin, ISBN);

ThuVien.trasach(ISBN);

cout << "danh sach sau khi tra 1 cuon" << endl;

ThuVien.hienthisach();

return 0;

}