### 1. Yêu cầu

**a. Yêu cầu chức năng :**

**+ Quản lý sách:**

- Thêm mới sách (tên sách, tác giả, nhà xuất bản, năm xuất bản, thể loại, số lượng, vị trí...).

- Chỉnh sửa thông tin sách.

- Xóa sách.

- Tìm kiếm sách (theo tên, tác giả, thể loại, ...).

- Xem chi tiết thông tin sách.

- Thống kê số lượng sách hiện có, sách đang mượn.

**+ Quản lý độc giả:**

- Đăng ký tài khoản độc giả mới (thông tin cá nhân, ngày đăng ký...).

- Cập nhật thông tin độc giả.

- Xóa tài khoản độc giả.

- Tìm kiếm độc giả (theo tên, mã độc giả...).

- Xem thông tin chi tiết độc giả.

**+ Quản lý mượn/trả sách:**

- Thực hiện mượn sách (chọn độc giả, chọn sách, ghi nhận ngày mượn, ngày hẹn trả).

- Thực hiện trả sách (ghi nhận ngày trả, kiểm tra tình trạng sách).

- Gia hạn mượn sách.

- Theo dõi lịch sử mượn/trả của độc giả.

- Thông báo sách quá hạn trả.

**+ Quản lý danh mục:**

- Thêm mới, chỉnh sửa, xóa các danh mục (thể loại sách, nhà xuất bản...).

**+ Quản lý người dùng hệ thống :**

- Đăng ký tài khoản quản trị viên mới.

- Quản lý quyền truy cập của người dùng (phân quyền cho các vai trò khác nhau).

- Theo dõi lịch sử hoạt động của hệ thống.

**+ Báo cáo và thống kê:**

- Thống kê số lượng sách theo thể loại, nhà xuất bản.

- Thống kê số lượng độc giả mới theo thời gian.

- Báo cáo sách đang được mượn, sách quá hạn.

- Báo cáo tình hình hoạt động của thư viện.

**b. Yêu cầu phi chức năng :** Mô tả các thuộc tính chất lượng của hệ thống.

**+ Hiệu suất:**

- Thời gian phản hồi nhanh chóng đối với các thao tác của người dùng.

- Khả năng xử lý đồng thời một lượng lớn người dùng.

**+ Bảo mật:**

- Bảo vệ dữ liệu người dùng và dữ liệu thư viện.

- Xác thực người dùng và phân quyền truy cập.

- Ngăn chặn các tấn công có thể xảy ra.

**+ Khả năng sử dụng:**

- Giao diện người dùng thân thiện, dễ hiểu và dễ sử dụng.

- Thiết kế trực quan, nhất quán.

- Hỗ trợ đa ngôn ngữ (nếu cần).

**+ Khả năng mở rộng:**

- Hệ thống có khả năng mở rộng để đáp ứng nhu cầu phát triển trong tương lai.

**+ Độ tin cậy:**

- Hệ thống hoạt động ổn định và ít xảy ra lỗi.

- Khả năng phục hồi dữ liệu khi có sự cố.

**+ Bảo trì:**

- Hệ thống dễ dàng bảo trì và nâng cấp.

### 2. Đặc tả

**+ Đặc tả giao diện người dùng :** Mô tả chi tiết các trang web, các thành phần giao diện, cách tương tác của người dùng. Có thể bao gồm wireframe, mockup.

**+ Đặc tả chức năng :** Mô tả chi tiết từng chức năng, bao gồm:

- Mô tả đầu vào, xử lý, đầu ra.

- Các quy tắc nghiệp vụ liên quan.

- Các trường hợp sử dụng (Use Cases).

**+ Đặc tả phi chức năng :** Mô tả chi tiết các tiêu chí về hiệu suất, bảo mật, khả năng sử dụng,... Ví dụ: "Thời gian tìm kiếm sách không quá 2 giây khi có 10.000 đầu sách."

**+ Đặc tả dữ liệu (Data Specification):** Mô tả cấu trúc cơ sở dữ liệu, các bảng, các trường, kiểu dữ liệu, mối quan hệ giữa các bảng.

### 3. Mô hình UML

**Sơ đồ Use Case (Use Case Diagram):** Mô tả các tương tác giữa người dùng (Actor) và hệ thống (Use Case). Các Actor có thể là Độc giả, Thủ thư (người quản lý mượn trả), Quản trị viên (Admin). Các Use Case có thể là: Đăng ký, Đăng nhập, Tìm kiếm sách, Mượn sách, Trả sách, Thêm sách, Xóa độc giả, Thống kê sách,...

Đoạn mã

@startuml

left to right direction

actor DocGia

actor ThuThu

actor QuanTriVien

rectangle "Hệ thống Quản lý Thư viện" {

DocGia -- (TimKiemSach)

DocGia -- (XemThongTinSach)

DocGia -- (MuonSach)

DocGia -- (TraSach)

DocGia -- (XemLichSuMuonTra)

ThuThu -- (MuonSach)

ThuThu -- (TraSach)

ThuThu -- (GiaHanMuon)

ThuThu -- (QuanLySach)

ThuThu -- (QuanLyDocGia)

QuanTriVien -- (QuanLySach)

QuanTriVien -- (QuanLyDocGia)

QuanTriVien -- (QuanLyDanhMuc)

QuanTriVien -- (QuanLyNguoiDung)

QuanTriVien -- (XemBaoCaoThongKe)

}

(QuanLySach) extends (ThemMoiSach)

(QuanLySach) extends (ChinhSuaSach)

(QuanLySach) extends (XoaSach)

(QuanLyDocGia) extends (ThemMoiDocGia)

(QuanLyDocGia) extends (ChinhSuaDocGia)

(QuanLyDocGia) extends (XoaDocGia)

@enduml

**Sơ đồ lớp (Class Diagram):** Mô tả cấu trúc tĩnh của hệ thống, các lớp đối tượng, thuộc tính và mối quan hệ giữa chúng. Một số lớp quan trọng có thể là: Sach, DocGia, PhieuMuon, PhieuTra, DanhMuc, NguoiDung,...

Đoạn mã

@startuml

class Sach {

+ tenSach: String

+ tacGia: String

+ nhaXuatBan: String

+ namXuatBan: int

+ theLoai: String

+ soLuong: int

+ ISBN: String

+ viTri: String

+ getThongTin(): String

}

class DocGia {

+ maDocGia: String

+ hoTen: String

+ ngaySinh: Date

+ diaChi: String

+ soDienThoai: String

+ email: String

}

class PhieuMuon {

+ maPhieuMuon: String

+ maDocGia: String

+ danhSachSachMuon: List<Sach>

+ ngayMuon: Date

+ ngayHenTra: Date

}

class PhieuTra {

+ maPhieuTra: String

+ maPhieuMuon: String

+ ngayTra: Date

+ tinhTrangSachTra: String

}

class DanhMuc {

+ maDanhMuc: String

+ tenDanhMuc: String

+ moTa: String

}

class NguoiDung {

+ tenDangNhap: String

+ matKhau: String

+ vaiTro: String

}

Sach "1..\*" -- "0..\*" PhieuMuon : duoc muon

DocGia "1" -- "\*" PhieuMuon : muon

PhieuMuon "1" -- "1" PhieuTra : tra

Sach "1" -- "\*" DanhMuc : thuoc

@enduml

**Sơ đồ tuần tự (Sequence Diagram):** Mô tả trình tự tương tác giữa các đối tượng theo thời gian để thực hiện một Use Case cụ thể. Ví dụ: sơ đồ tuần tự cho Use Case "Mượn sách".

**Sơ đồ trạng thái (State Machine Diagram):** Mô tả các trạng thái có thể của một đối tượng và các sự kiện gây ra sự chuyển đổi trạng thái. Ví dụ: trạng thái của đối tượng Sach có thể là "Còn", "Đang mượn", "Đã mất".

**Sơ đồ hoạt động (Activity Diagram):** Mô tả luồng công việc của một quy trình nghiệp vụ hoặc một Use Case. Ví dụ: quy trình "Mượn sách".

**Sơ đồ triển khai (Deployment Diagram):** Mô tả kiến trúc vật lý của hệ thống, các node (server, client) và các thành phần phần mềm được triển khai trên đó.

### 4. Công nghệ đề xuất (Suggested Technologies)

**+Frontend:** HTML, CSS

**+Backend:** NodeJS (Express),, Java (Spring Boot)

**+Database:** MySQL, PostgreSQL, MongoDB

**+Server:** Nginx

### 5. Các bước phát triển dự án (Project Development Steps)

Một quy trình phát triển phần mềm điển hình có thể bao gồm các bước sau:

1. **Thu thập yêu cầu và phân tích:** Xác định rõ ràng các yêu cầu của người dùng.
2. **Thiết kế hệ thống:** Xây dựng các mô hình UML, thiết kế cơ sở dữ liệu, thiết kế giao diện.
3. **Lập trình :** Triển khai các chức năng của hệ thống dựa trên thiết kế.
4. **Kiểm thử :** Kiểm tra các chức năng, hiệu suất, bảo mật của hệ thống.
5. **Triển khai :** Đưa hệ thống lên môi trường thực tế.
6. **Bảo trì và nâng cấp :** Sửa lỗi, cập nhật tính năng mới.