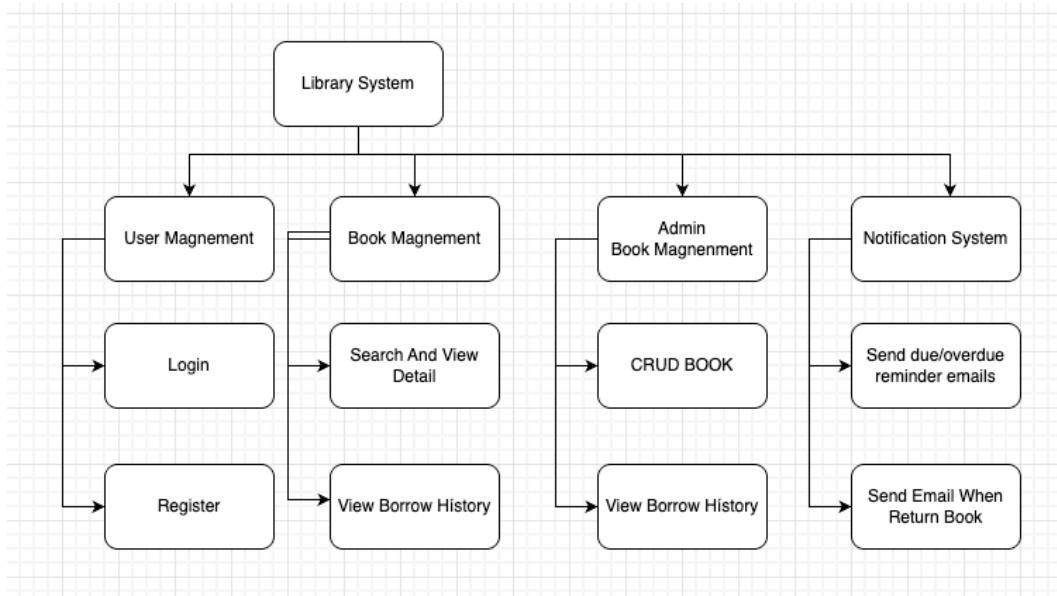


# Library Management System (Microservices Prototype)

## Sơ đồ phân rã chức năng



## USE CASE 1 – Đăng ký User (Register)

### 1. Thông tin chung

- **Tên Use Case:** Đăng ký User
- **Actor chính:** User
- **Mô tả ngắn:** User tạo tài khoản mới để sử dụng hệ thống

### 2. Tiền đề/Điều kiện (Preconditions)

- User chưa có tài khoản trong hệ thống.
- User có kết nối mạng và truy cập được API Gateway

### 3. Hậu đி/Điều kiện (Postconditions)

- **Thành công:** Tài khoản mới được lưu vào User Service DB với role mặc định là "user".
- **Thất bại:** Không có dữ liệu nào được lưu.

#### 4. Luồng sự kiện chính (Main Flow)

Bước	Mô tả
1	User chọn chức năng "Đăng ký".
2	User nhập thông tin: username, email, mật khẩu, v.v.
3	Hệ thống kiểm tra định dạng và validate dữ liệu.
4	Hệ thống kiểm tra username/email đã tồn tại hay chưa.
5	Nếu hợp lệ, User Service lưu tài khoản mới vào DB.
6	Hệ thống trả kết quả đăng ký thành công cho User.

#### 5. Luồng phụ / Ngoại lệ (Alternate / Exception Flow)

- **E1 – Dữ liệu không hợp lệ**  
→ Hệ thống trả lỗi.
- **E2 – Username hoặc email đã tồn tại**  
→ Hệ thống trả lỗi.
- **E3 – Lỗi kết nối DB**  
→ Hệ thống trả lỗi.

## USE CASE 2 – Đăng nhập User (Login)

### 1. Thông tin chung

- **Tên Use Case:** Đăng nhập User
- **Actor chính:** User / Admin
- **Mô tả ngắn:** Người dùng đăng nhập để lấy JWT token truy cập hệ thống.

### 2. Tiền đề kiện (Preconditions)

- User đã có tài khoản hợp lệ trong hệ thống.
- User nhập chính xác username/email và mật khẩu.

### 3. Hậu đài kiện (Postconditions)

- **Thành công:**
  - Hệ thống trả về JWT token chứa thông tin userId, role.
- **Thất bại:**
  - Không trả token, hệ thống chỉ gửi thông báo lỗi.

#### 4. Luồng sự kiện chính (Main Flow)

Bước	Mô tả
1	User chọn chức năng "Đăng nhập".
2	User nhập username/email + mật khẩu.
3	Hệ thống nhận request và kiểm tra thông tin đăng nhập.
4	Nếu hợp lệ, hệ thống sinh JWT token.
5	Hệ thống trả token và echo User để sử dụng cho các API tiếp theo.

#### 5. Luồng phụ / Ngoại lệ

- **E1 – Sai username/email hoặc mật khẩu**  
→ Hệ thống trả lỗi.
- **E2 – Tài khoản bị khóa**  
→ Hệ thống trả lỗi.
- **E3 – Lỗi hệ thống khi tạo token**  
→ Hệ thống trả lỗi.

## USE CASE 3 – Thêm sách (Create Book)

### 1. Thông tin chung

- **Tên Use Case:** Thêm sách
- **Actor chính:** Admin
- **Mô tả ngắn:** Admin gửi yêu cầu tạo mới sách vào hệ thống Book Catalog Service.

### 2. Tiền đề kiện (Preconditions)

- Request phải đi qua API Gateway với **JWT hợp lệ** và role **admin**.
- Các trường dữ liệu bắt buộc của sách phải có (title, author, quantity...).
- Book Catalog Service đang kết nối được DB.

### 3. Hậu đிều kiện (Postconditions)

- **Thành công:** Một record trong bảng Books được tạo mới.
- **Thất bại:** Không có thay đổi nào trong DB.

### 4. Luồng sự kiện chính (Main Flow)

Bước	Mô tả
1	Backend nhận request POST /books từ API Gateway (kèm JWT).
2	Book Catalog Service xác thực role = admin.
3	Service validate dữ liệu sách (title, author, quantity > 0).
4	Service lưu dữ liệu sách mới vào PostgreSQL.
5	Service trả về kết quả thành công (ID của sách vừa được tạo).

### 5. Luồng phụ / Ngoại lệ (Exception Flow)

- **E1 – JWT không hợp lệ hoặc không phải admin**
- **E2 – Dữ liệu sách không hợp lệ**
- **E3 – Lỗi DB / không tạo được sách**

## USE CASE 4 – Cập nhật sách (Update Book)

### 1. Thông tin chung

- **Tên Use Case:** Cập nhật sách
- **Actor chính:** Admin
- **Mô tả ngắn:** Admin cập nhật thông tin sách trong Book Catalog Service.

## 2. Tiêu đề yêu cầu

- Request đi qua API Gateway và JWT hợp lệ, role = admin.
- Sách cần cập nhật phải tồn tại trong DB.

## 3. Hậu đính yêu

- Thành công:** Dữ liệu sách trong DB được cập nhật theo request.
- Thất bại:** Không có thay đổi nào trong DB.

## 4. Luồng sự kiện chính

Bước	Mô tả
1	Backend nhận request PUT /books/{id} từ Gateway.
2	Book Catalog Service kiểm tra JWT + role.
3	Service tìm sách theo ID trong DB.
4	Service validate các trường dữ liệu cập nhật.
5	Service cập nhật thông tin sách vào PostgreSQL.
6	Service trả về thông báo cập nhật thành công.

## 5. Lưu ý / Ngoại lệ

- E1 – JWT không hợp lệ / không phải admin**
- E2 – Không tìm thấy sách với ID yêu cầu**
- E3 – Dữ liệu cập nhật không hợp lệ**
- E4 – Lỗi DB khi cập nhật**

# USE CASE 5 – Xóa sách (Delete Book)

## 1. Thông tin chung

- Tên Use Case:** Xóa sách

- **Actor chính:** Admin
- **Mô tả ngắn:** Admin gửi yêu cầu xóa một sách khỏi Book Catalog Service

## 2. Tiền đề kiện

- JWT hợp lệ, role = **admin**.
- Sách cần xóa **tồn tại** trong DB.

## 3. Hậu đி kiện (Postconditions)

- **Thành công:** Record sách được xóa khỏi DB (hoặc đánh dấu isDeleted tùy thiết kế).
- **Thất bại:** Không có thay đổi nào trong DB.

## 4. Luồng sự kiện chính (Main Flow)

Bước	Mô tả
1	Backend nhận request DELETE /books/{id} từ API Gateway.
2	Book Catalog Service xác thực JWT và role admin.
3	Service tìm sách theo ID trong DB.
4	Nếu tồn tại, service thực hiện xóa sách (delete hoặc soft delete).
5	Service trả về thông báo xóa thành công.

## 5. Luồng phụ / Ngoại lệ (Exception Flow)

- **E1 – JWT không hợp lệ hoặc không phải admin**
- **E2 – Không tìm thấy sách theo ID**
- **E3 – Lỗi DB khi xóa**

# USE CASE 6 – Xem danh sách / chi tiết sách

## 1. Thông tin chung

- **Tên Use Case:** Xem danh sách / chi tiết sách
- **Actor chính:** User / Admin

- **Mô tả ngắn:** Backend trả về danh sách sách hoặc chi tiết một sách cụ thể.

## 2. Tiềm đe dọa

- Request đi qua API Gateway.
- Book Catalog Service đang kết nối DB bình thường.

## 3. Hậu đe dọa

- **Thành công:**
  - Trả về danh sách tất cả sách (GET /books) hoặc thông tin chi tiết 1 sách (GET /books/{id}).
- **Thất bại:**
  - Không trả dữ liệu, chỉ trả lỗi tương ứng.

## 4. Lưu ý sự kiện chính

### Trường hợp A – Lấy danh sách sách

Bước	Mô tả
1	Backend nhận request GET /books từ Gateway.
2	Book Catalog Service truy vấn tất cả sách trong DB.
3	Service trả về danh sách sách, bao gồm trạng thái availability.

### Trường hợp B – Xem chi tiết sách

Bước	Mô tả
1	Backend nhận request GET /books/{id}.
2	Book Catalog Service tìm sách trong DB theo ID.
3	Trả về thông tin chi tiết (title, author, quantity, availability, ...).

## 5. Lưu ý phụ / Ngoại lệ (Exception Flow)

- **E1 – Không tìm thấy sách**

- **E2 – Lỗi DB khi truy vấn**

# USE CASE 7 – Tìm kiếm sách (Search Books)

## 1. Thông tin chung

- **Tên Use Case:** Tìm kiếm sách
- **Actor chính:** User / Admin
- **Mục tiêu:** tìm sách theo các tiêu chí cơ bản.

## 2. Tiêu đề yêu cầu

- Request đi qua API Gateway.
- Book Catalog Service hoạt động bình thường.
- Dữ liệu tìm kiếm (title/author/category/availability) hợp lệ.

## 3. Hậu đài yêu cầu

- **Thành công:** Trả về danh sách sách thỏa đài yêu cầu lọc.
- **Thất bại:** Không có thay đổi dữ liệu, chỉ trả lỗi.

## 4. Luồng sự kiện chính

Bước	Mô tả
1	Backend nhận request GET /books?filters... từ Gateway.
2	Book Catalog Service parse các tham số lọc title.
3	Service truy vấn DB theo bộ lọc.
4	Service trả danh sách kết quả cho Gateway.

## 5. Ngoại lệ

- **E1 – Tham số lọc không hợp lệ**

- E2 – Lỗi DB

# USE CASE 8 – gRPC cập nhật số lượng sách

## 1. Thông tin chung

- **Tên Use Case:** Cập nhật số lượng sách qua gRPC
- **Actor chính:** Borrowing Service
- **Mục tiêu:** Cho phép Borrowing Service kiểm tra & cập nhật availability sách khi mượn/trả.

## 2. Tiêu đề yêu cầu

- Book Catalog Service expose các gRPC method:
  - CheckBookAvailability(bookId)
  - ReserveBook(bookId)
  - ReturnBook(bookId)
- Borrowing Service kết nối gRPC được.
- Book tồn tại trong DB.

## 3. Hậu đài yêu cầu

- **Thành công:**
  - Khi mượn: số lượng available giảm.
  - Khi trả: số lượng available tăng.
- **Thất bại:**
  - Không thay đổi dữ liệu sách.

## 4. Lưu ý sự kiện chính

### A. Kiểm tra availability

Bước	Mô tả
1	Borrowing Service gọi <code>CheckBookAvailability(bookId)</code> .
2	Book Catalog Service truy vấn DB lấy số lượng available.
3	Service trả kết quả (Available = true/false).

### B. Giảm tồn kho khi mượn (ReserveBook)

Bước	Mô tả
1	Borrowing Service gọi <code>ReserveBook(bookId)</code> .
2	Book Catalog Service giảm giá trị available trong DB.
3	Trả về kết quả thành công.

### C. Tăng tồn kho khi trả (ReturnBook)

Bước	Mô tả
1	Borrowing Service gọi <code>ReturnBook(bookId)</code> .
2	Book Catalog Service tăng số lượng available.
3	Trả kết quả thành công.

## 5. Ngoại lệ (Exception Flow)

- E1 – Sách không tồn tại
- E2 – Không đủ số lượng để Reserve
- E3 – Lỗi DB
- E4 – gRPC connection failed

# USE CASE 9 – Mượn sách

## 1. Thông tin chung

- **Tên Use Case:** Mượn sách

- **Actor chính:** User
- **Mục tiêu:** User gửi yêu cầu mượn sách, hệ thống cập nhật DB và availability.

## 2. Tiền đề kiện

- User đã đăng nhập → có JWT hợp lệ.
- Sách tồn tại trong Book Catalog.
- Borrowing Service kết nối được Book Catalog Service (gRPC).
- RabbitMQ hoạt động bình thường.

## 3. Hậu đài kiện

- **Thành công:**
  - 1 phiếu mượn được tạo trong DB Borrowing Service.
  - Book Catalog Service giảm số lượng available (ReserveBook).
  - Event “borrowed” được gửi lên MQ (history + email).
- **Thất bại:**
  - Không tạo phiếu mượn.
  - Không cập nhật availability.

## 4. Luồng sự kiện chính (Main Flow)

Bước	Mô tả
1	Backend nhận request POST /borrowings/borrow (kèm JWT) qua Gateway.
2	Borrowing Service xác thực JWT.
3	Service validate yêu cầu mượn (bookId, dueDate, điều kiện nghiệp vụ).
4	Service gọi gRPC CheckBookAvailability(bookId).
5	Nếu đủ sách → gọi ReserveBook(bookId) để giảm available.
6	Borrowing Service tạo record Borrowing trong PostgreSQL.
7	Service gửi 2 event lên MQ:

	<ul style="list-style-type: none"> <li>– Event 1: cập nhật lịch sử mượn.</li> </ul>
	<ul style="list-style-type: none"> <li>– Event 2: gửi email thông báo mượn.</li> </ul>
8	Borrowing Service trả kết quả thành công về Gateway.

## 5. Ngoại lệ

- **E1 – JWT không hợp lệ**
- **E2 – Không đủ sách để mượn**
- **E3 – gRPC failure khi check/update availability**
- **E4 – Lỗi khi ghi DB**
- **E5 – Lỗi publish MQ**

# USE CASE 10 – Trả sách (Return Book)

## 1. Thông tin chung

- **Tên Use Case:** Trả sách
- **Actor chính:** User
- **Mục tiêu:** User trả lại sách đã mượn, hệ thống cập nhật trạng thái mượn và số lượng sách.

## 2. Tiền đề/Điều kiện

- User đã đăng nhập, có **JWT hợp lệ**.
- Phiếu mượn tồn tại và đang ở trạng thái **Borrowed**.
- Borrowing Service kết nối được với Book Catalog Service (gRPC).
- RabbitMQ hoạt động bình thường.

## 3. Hậu đĩa/Điều kiện

- **Thành công:**
  - Phiếu mượn được cập nhật trạng thái **Returned**, có ngày trả (ReturnDate).

- Book Catalog Service tăng **availability** .
- Event “returned” được gửi lên MQ (history + email).
- **Thất bại:**
  - Không thay đổi trạng thái phiếu mượn.
  - Không thay đổi availability sách.

#### 4. Luồng sự kiện chính (Main Flow)

Bước	Mô tả
1	Backend nhận request POST /borrowings/return (kèm JWT, borrowingId hoặc bookId).
2	Borrowing Service xác thực JWT.
3	Service tìm phiếu mượn trong DB theo borrowingId + userId.
4	Kiểm tra phiếu mượn có trạng thái <b>Borrowed</b> .
5	Gọi gRPC <b>ReturnBook (bookId)</b> sang Book Catalog Service để tăng available.
6	Cập nhật phiếu mượn sang trạng thái <b>Returned</b> , set ReturnDate, tính tiền phạt (nếu có).
7	Gửi 2 event lên MQ: history (action=return) và email (type=BookReturned).
8	Trả kết quả thành công về Gateway.

#### 5. Ngoại lệ (Exception Flow)

- **E1 – JWT không hợp lệ**
- **E2 – Không tìm thấy phiếu mượn**
- **E3 – Phiếu mượn đã ở trạng thái Returned**
- **E4 – gRPC lỗi khi gọi ReturnBook**
- **E5 – Lỗi DB khi cập nhật phiếu mượn**

## USE CASE 11 – Cập nhật lịch sử mượn trả

### 1. Thông tin chung

- **Tên Use Case:** Cập nhật lịch sử mượn trả

- **Actor chính:** System
- **Mục tiêu:** Nhận event từ MQ và ghi/ cập nhật lịch sử mượn trả trong DB.

## 2. Tiềm đi kèm kiện

- RabbitMQ đang chạy và có queue lịch sử (ví dụ: lịch sử mượn).
- Borrowing Service đăng ký lắng nghe queue này.
- DB Borrowing Service hoạt động bình thường.

## 3. Hậu đi kèm kiện

- **Thành công:**
  - Lịch sử mượn/trả (BorrowHistory) được tạo mới hoặc cập nhật tương ứng với event.
- **Thất bại:**
  - Event chưa được xử lý, có thể được retry hoặc nằm lại trong queue (tùy config).

## 4. Luồng sự kiện chính

Bước	Mô tả
1	Worker/Service subscribe queue <b>borrow-history</b> trên MQ.
2	Khi có message mới (borrow/return), service nhận message.
3	Parse nội dung message: userId, bookId, action, timestamp,...
4	Ghi mới hoặc cập nhật record trong bảng BorrowHistory (hoặc bảng thống kê tương ứng).
5	Ghi log kết quả xử lý thành công.

## 5. Ngoại lệ (Exception Flow)

- **E1 – Message không đúng format / thiếu field**
- **E2 – Lỗi DB khi ghi lịch sử**
- **E3 – Mất kết nối MQ**

# USE CASE 12 – Gửi thông báo Email

## 1. Thông tin chung

- **Tên Use Case:** Gửi thông báo Email
- **Actor chính:** Notification Service (System)
- **Mục tiêu:** Nhận event từ MQ và gửi email tương ứng (mượn, trả, nhắc hạn...).
- **Thuộc Service:** Notification Service

## 2. Tiền đề kiện (Preconditions)

- Notification Service subscribe queue email (ví dụ: `email-notification`).
- RabbitMQ hoạt động bình thường.
- Cấu hình SMTP / Email provider hợp lệ, gửi mail được.

## 3. Hậu đài kiện (Postconditions)

- **Thành công:**
  - Email được gửi đến địa chỉ mail user tương ứng với loại event (Borrowed / Returned / DueSoon...).
- **Thất bại:**
  - Email không gửi được

## 4. Luồng sự kiện chính (Main Flow)

Bước	Mô tả
1	Notification Service subscribe queue <code>email-notification</code> .
2	Nhận message từ MQ (type: Borrowed/Returned/DueSoon, userEmail, data...).
3	Parse message để lấy thông tin cần thiết cho email.
4	Build nội dung email (subject, body) theo loại sự kiện.
5	Gửi email thông qua SMTP / email provider.

## 5. Ngoại lệ (Exception Flow)

- **E1 – Message không đúng format**
- **E2 – Lỗi khi gửi email (mạng, SMTP lỗi)**
- **E3 – Mất kết nối MQ**

CSDL

### Bảng Users

Cột	Kiểu	Ghi chú
Id	uuid / int	PK
Username	varchar	unique, dùng để login
Password Hash	varchar	lưu hash mật khẩu
Password HashSalt	Varchar	lưu hashsalt mật khẩu

### Bảng ROLE

Cột	Kiểu	Ghi chú
Id	int	PK
Role	varchar	Tên Role

### Bảng UsersRole

Cột	Kiểu	Ghi chú
UserId	int	ID lấy từ User
RoleID	Int	ID lấy từ Role

### Bảng Books

Cột	Kiểu	Ghi chú
Id	int	PK

Title	varchar	tên sách
Author	varchar	tên tác giả
categoryId	Int	mã category
AvailableCopies	int	số lượng hiện còn để mượn

## Bảng Borrowings

Cột	Kiểu	Ghi chú
Id	Int	PK
UserId	int	Id user (từ User Service)
borrowDate	datetime	Id sách (từ Book Service)
dueDate	datetime	Ngày Hạn
returnDate	datetime	Ngày trả
Status	Varchar	Trạng thái

## Bảng borrowing\_item

Cột	Kiểu	Ghi chú
Id	Int	PK
bookId	int	Id user (từ User Service)
borrowingId	datetime	Id sách (từ Book Service)
Quantity	datetime	Ngày Hạn
returnDate	datetime	Ngày trả
Status	Varchar	Trạng thái

## Bảng Categories

Cột	Kiểu	Ghi chú
Id	Int	PK
Name	Varchar	Tên loại

Bảng users lưu trữ thông tin người dùng của hệ thống

Bảng userrole dùng để lưu trữ

Bảng role dùng để lưu các role

Bảng Borrowing\_items: lưu danh sách các sách nằm trong một phiếu mượn

Bảng Borrowings: Đại diện cho một “phiếu mượn”, liên kết tới người mượn qua userId

Bảng books lưu trữ thông tin chi tiết về sách

Bảng categories sẽ lưu các categories của sách

