LIBRARY MANAGEMENT SYSTEM

1) Project Overview

The Library Management System (LMS) is a comprehensive web-based application designed to streamline book cataloguing, lending operations, and user membership management. The goal is to provide a seamless experience for members to discover, reserve, borrow, and return books, and for librarians/admins to manage inventory, loans, fines, and analytics—backed by secure APIs using JWT.

Objectives:

- Centralize book and member data.
- Digitize lending/return workflows with due dates and fines.
- Enable search, reservation, and notifications.
- Provide role-based access with Spring Security (JWT).
- Support standardized API responses, validations, and robust exception handling.

---

2) User Roles

1. Reader
    - Browse/search catalog; view availability.
    - Place reservations; borrow and return books.
    - View loan history, due dates, and fines.
    - Rate/review books.
2. Librarian
    - Manage inventory: add/edit/remove books & copies.
    - Approve/deny reservations; issue/return books; adjust due dates.
    - Manage members (create/update/disable).
    - Monitor overdue items and fines; process payments.
3. Admin
    - Full control over roles & permissions.
    - Manage users (librarians & readers), categories.

---

3) Project Flow

1. Reader
   - Sign up/sign in → Search catalog → Reserve/borrow a book → Track due date → Return → Pay fine if applicable → Rate/review.
2. Librarian
   - Sign in → Manage catalog (books, copies) → Validate reservations → Issue loans → Handle returns → Assess & update fines → Assist readers.
3. Admin
   - Sign in → Configure loan/fine policies → Manage users/roles → View dashboards & reports → Audit system activities.

---

4) Core Features & Functionalities

1. User Registration & Authentication (JWT)
   - Sign up/sign in endpoints .
   - Role-based authorization (Reader/Librarian/Admin).
2. Catalog Management
   - Manage books (title, author(s), ISBN, genre, language, publisher, published year).
   - Copies management with unique copy IDs, status (Available, On Loan, Reserved).
   - Metadata: categories, tags, shelf/location.
3. Search & Discovery
   - Full-text search (title/author/ISBN/keywords).
   - Filters: availability, category, language, rating.
   - Sorting: relevance, published year, popularity.
4. Reservation & Borrowing
   - Members can reserve available copies.
   - Librarians approve/auto-approve based on policy.
   - Issue loans with due date calculation.
5. Returns & Fines
   - Return processing updates copy status.
   - Fine calculation for overdue returns (configurable per day rate).

6. User Profile & History
    - Loan/reservation history, current loans, fines summary.
    - Saved lists (Wishlist).
7. Admin Dashboard & Reports
    - Active loans, overdue count, fines collected.
    - Most borrowed titles/authors/categories.
    - Member engagement metrics.
8. Security & Compliance
    - JWT-based authentication; role-based endpoints.

---

5) Technology Stack

- **Backend**: Spring Boot, Spring MVC, Spring Security (JWT), Spring Data JPA.
- **Database**: MySQL using JDBC.
- **Validation**: Bean Validation.
- **Logging**: log4j.
- **Testing**: JUnit.

---

6) Common Entities

1. **User**
    - **Purpose:** Represents individuals interacting with the system (Readers, Librarians, Admins).
    - **Attributes:** id, name, email, password, role, status, createdAt.
    - **Why:** To manage authentication, authorization, and user-specific operations like borrowing or managing books.
2. **Book**
    - **Purpose:** Represents a book in the library catalog.
    - **Attributes:** id, title, authors, isbn, category, language, publishedYear, totalCopies, availableCopies.
    - **Why:** To store and manage book details for searching, borrowing, and reporting.
3. **Copy**
    - **Purpose:** Represents a physical copy of a book.

- **Attributes:** id, bookId, barcode, shelfLocation, status (AVAILABLE, ON_LOAN, RESERVED, LOST).
- **Why:** To track individual copies for lending and inventory management.

4. **Loan**
    - **Purpose:** Represents the borrowing transaction of a book copy by a user.
    - **Attributes:** id, userId, copyId, issuedAt, dueAt, returnedAt, status.
    - **Why:** To manage lending operations, due dates, and overdue checks.
5. **Reservation**
    - **Purpose:** Represents a user's request to hold a book when no copies are available.
    - **Attributes:** id, userId, bookId, status (PENDING, APPROVED, CANCELLED, FULFILLED), createdAt, expiresAt.
    - **Why:** To ensure fair allocation of popular books and manage waitlists.
6. **Fine**
    - **Purpose:** Represents penalties for overdue returns.
    - **Attributes:** id, loanId, userId, amount, calculatedAt, paid, paidAt.
    - **Why:** To enforce timely returns and maintain accountability.