Design Document
IT 643 SOFTWARE DESIGN AND TESTING
LIBRARY MANAGEMENT SYSTEM
202412045 MANAN PANDYA
202412038 NIKHIL KHILWANI
202412015 SWARA CHOKSHI
202412050 NISHIT NAGAR

Library Management System – Design Document

1. Introduction

The Library Management System is designed to manage the cataloging, borrowing, restoration, and notification processes of a library. It handles multiple types of books, different categories of users, and maintains logs of borrow and restoration activities. The system will be built with **Java, Spring Boot, and JPA/Hibernate** using an object-relational mapping (ORM) approach.

The design follows an object-oriented structure with clear separation of entities, data transfer objects (DTOs), and enumerations.

2. System Objectives

- Manage different types of books (general, rare, ancient manuscripts).
- Support user roles (librarians, scholars, guests).
- Allow borrowing and returning of books.
- Maintain logs for restoration and borrowing activities.
- Provide notifications to users.
- Ensure extensibility for future features like digital access and archival processes.

3. Design Overview

The design is structured around two main hierarchies: **Books** and **Users**. Additional supporting entities manage logs and notifications. DTOs are introduced to decouple the presentation layer from persistence entities.

- **Entities:** Represent persistent domain objects.
- **DTOs:** Used for transferring data between layers.
- **Enums:** Represent fixed sets of values like roles and statuses.

---

4. Class Design

4.1 Book Hierarchy

- **`Book` (abstract)**
  - Attributes: `id`, `title`, `author`, `isbn`, `section`, `status`
  - Methods: Abstract `getLateFeeRate()`, `getType()`
  - Inheritance strategy: Single table with a discriminator column `book_type`
  - Purpose: Acts as the base class for all book types.
- **`GeneralBook`**
  - Field: `digitalAccess`
  - Late Fee Rate: `1.0`
  - Type: `"GENERAL"`
- **`RareBook`**
  - Fields: `preservationMethod`, `inLibraryUseOnly`

    - Late Fee Rate: `5.0`
    - Type: `"RARE"`
- **`AncientScript`**
    - Fields: `originalLanguage`, `translationNotes`, `archived`, `manuscriptPath`
    - Late Fee Rate: `10.0`
    - Type: `"ANCIENT"`
- **`BookStatus` (enum)**
    - Possible values: `AVAILABLE`, `BORROWED`, `RESERVED`, etc.

4.2 User Hierarchy

- **`User` (abstract)**
    - Attributes: `id`, `name`, `email`, `password`, `role`
    - Methods: Abstract `getType()`
    - Inheritance strategy: Single table with a discriminator column `user_type`
- **`Scholar`**
    - Type: `"SCHOLAR"`
- **`Librarian`**
    - Type: `"LIBRARIAN"`
- **`Guest`**

    - Type: `"GUEST"`
- **`Role`** **(enum)**
    - Values: `LIBRARIAN`, `SCHOLAR`, `GUEST`

4.3 Supporting Entities

- **`BorrowLog`**
    - Fields: `id`, `borrower (User)`, `book (Book)`, `borrowDate`, `returnDate`, `returned`
    - Purpose: Tracks borrowing and returning of books.
    - Relationships:
        - `ManyToOne` with `User`
        - `ManyToOne` with `Book`
- **`RestorationLog`**
    - Fields: `id`, `book (Book)`, `restorationDate`, `performedBy`, `notes`
    - Purpose: Logs restoration or preservation actions performed on books.
    - Relationships:
        - `ManyToOne` with `Book`
- **`Notification`**
    - Fields: `id`, `message`, `bookTitle`, `timestamp`, `isRead`
    - Purpose: Provides user-facing alerts (e.g., due dates, restoration updates).

4.4 Data Transfer Objects (DTOs)

- **`RegistrationDto`**
  - Fields: `name`, `email`, `password`, `role`
  - Purpose: Used for user registration without exposing internal entity structures.
- **`BookFormDto`**
  - Fields: `type`, `title`, `author`, `isbn`, `section`, `status`, `digitalAccess`, `preservationMethod`, `inLibraryUseOnly`, `originalLanguage`
  - Purpose: Simplifies book creation and form handling, encapsulating fields across different book types.

5. Relationships

- **Book Inheritance:** Single Table Inheritance with `AncientScript`, `GeneralBook`, and `RareBook`.
- **User Inheritance:** Single Table Inheritance with `Scholar`, `Librarian`, and `Guest`.
- **BorrowLog:** Connects a `User` and a `Book`.
- **RestorationLog:** Connects a `Book` with restoration metadata.
- **Notification:** Independent entity, but typically associated with a book or user context at the application level.

6. Business Logic Notes

- Late fee rates are defined per book type (`General = 1.0`, `Rare = 5.0`, `Ancient = 10.0`).
- Librarians may have elevated privileges (e.g., restoring, approving borrow requests).
- Scholars and guests differ in access privileges (guests might be restricted from borrowing certain books).
- Restoration logs provide accountability for preservation actions.
- Notifications may support reminders for return deadlines or book availability.

---

7. Extensibility

- The design supports adding new **book types** by subclassing `Book`.
- New **user roles** can be introduced by subclassing `User` and extending the `Role` enum.
- Additional features like reservations, fines, or digital lending can be added by introducing new entities and logs.

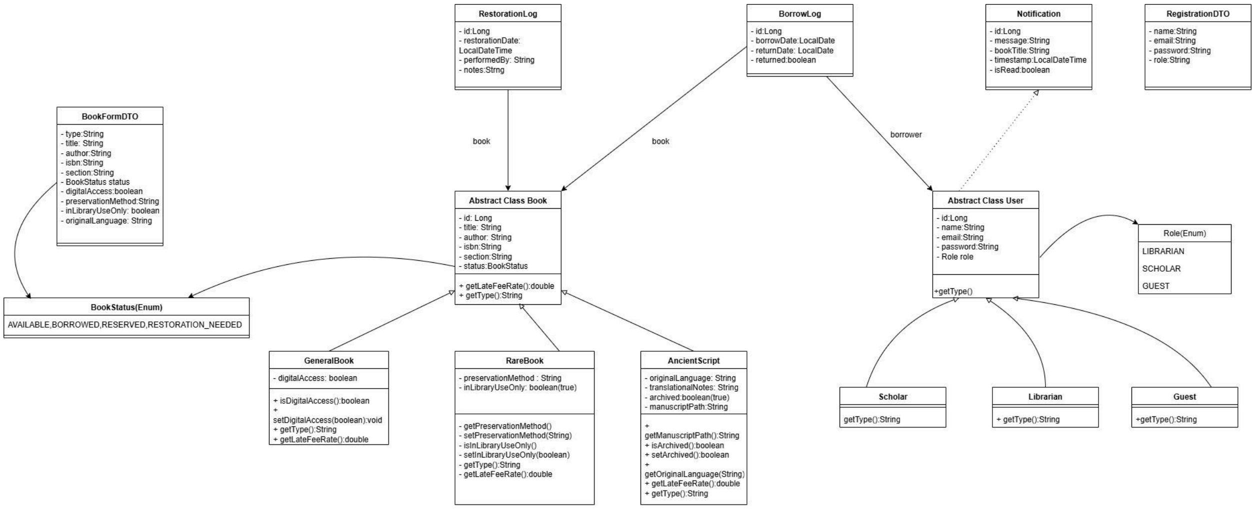
---

8. UML Diagram Reference

The class diagram provided (PlantUML) illustrates:

- Inheritance hierarchies
- Entity relationships
- DTOs and enums

This document corresponds directly to the UML diagram to maintain consistency between design and implementation.


RestorationLog
- id:Long
- restorationDate: LocalDateTime
- performedBy: String
- notes:Strng
BorrowLog
- id:Long
- borrowDate:LocalDate
- returnDate: LocalDate
- returned:boolean
Notification
- id:Long
- message:String
- bookTitle:String
- timestamp:LocalDateTime
- isRead:boolean
RegistrationDTO
- name:String
- email:String
- password:String
- role:String
BookFormDTO
- type:String
- title: String
- author:String
- isbn:String
- section:String
- BookStatus status
- digitalAccess:boolean
- preservationMethod:String
- inLibraryUseOnly: boolean
- originalLanguage: String
book
book
borrower
Abstract Class Book
- id: Long
- title: String
- author: String
- isbn:String
- section:String
- status:BookStatus
+ getLateFeeRate():double
+ getType():String
Abstract Class User
- id:Long
- name:String
- email:String
- password:String
- Role role
+getType()
Role(Enum)
LIBRARIAN
SCHOLAR
GUEST
BookStatus(Enum)
AVAILABLE,BORROWED,RESERVED,RESTORATION_NEEDED
GeneralBook
- digitalAccess: boolean
+ isDigitalAccess():boolean
+ setDigitalAccess(boolean):void
+ getType():String
+ getLateFeeRate():double
RareBook
- preservationMethod : String
- inLibraryUseOnly: boolean(true)
- getPreservationMethod()
- setPreservationMethod(String)
- isInLibraryUseOnly()
- setInLibraryUseOnly(boolean)
- getType():String
- getLateFeeRate():double
AncientScript
- originalLanguage: String
- translationalNotes: String
- archived:boolean(true)
- manuscriptPath:String
+ getManuscriptPath():String
+ isArchived():boolean
+ setArchived():boolean
+ getOriginalLanguage(String)
+ getLateFeeRate():double
+ getType():String
Scholar
getType():String
Librarian
+ getType():String
Guest
+getType():String