**Library Management System (LMS) – Software Design Document(SDD)**

**1. Introduction**

This document outlines the software design for a Library Management System (LMS) that allows users to search for books, view book information, and potentially checkout books (though the checkout functionality isn't explicitly shown in the class diagram). There are two user roles: admin and user.

**2. System Overview**

The LMS is a web-based application that caters to two user roles:

* **User:** General library member who can search for books, view details, and potentially borrow books (checkout functionality not explicitly shown in the class diagram).
* **Admin:** Library administrator with privileges to manage the book collection, user accounts, and book ratings.

**3. System Requirements**

**3.1 Functional Requirements**

* **User:**
  + Search for books by title, author, genre, or ISBN.
  + View detailed information about a book (title, author, genre, ISBN, publication year, description, average rating).
  + View their account information and checkout history (to be confirmed based on project scope).
* **Admin:**
  + Login to the system.
  + Logout of the system.
  + Add new books to the system.
  + Edit information for existing books.
  + Delete books from the system.
  + Manage user accounts:
    - Approve new user registrations (potentially).
    - View a list of all registered users.
    - Edit existing user information.
    - Delete user accounts.
  + View all book ratings.
  + Manage book ratings (functionality to be defined - edit, delete ratings).
  + Add book ratings (potentially for sample data or managing user ratings).

**3.2 Non-Functional Requirements**

* The system should be user-friendly and easy to navigate.
* The system should be responsive and perform searches efficiently.
* The system should be secure and protect user data.

**4. System Architecture**

The LMS will be implemented as a three-tier architecture:

* **Presentation Layer:** The user interface (UI) for both users and admins will be web-based. This layer will handle user interactions and display relevant information.
* **Business Logic Layer:** This layer will manage the core functionalities of the system, including searching for books, managing user accounts, and handling book operations for administrators.
* **Data Access Layer:** This layer will interact with the database to store and retrieve book data, user information, and book ratings.

**5. Detailed Design**

**5.1 User Interface (UI) Design**

* The UI will cater to both users and admins with separate dashboards.
* The user dashboard will allow users to:
  + Search for books by title, author, genre, or ISBN using a search bar.
  + View a list of search results with basic book information (title, author, genre).
  + Click on a book title to view detailed book information (title, author, genre, ISBN, publication year, description, average rating).
  + Potentially view their account information and checkout history (to be confirmed based on project scope).
* The admin dashboard will allow users to:
  + Login: Provides a login form for admins to enter their ID and password.
  + Logout: A button or menu option for admins to log out of the system.
  + Add Book: A dedicated form to add new books with fields for title, author, genre, ISBN, publication year, and description.
  + Edit Book: Clicking on a book title in the book list allows admins to edit existing book information.
  + Delete Book: An option to delete a book from the system (confirmation prompt recommended).
  + Manage User Accounts:
    - A section to view a list of all users.
    - Functionality to approve new user registrations (if applicable).
    - Option to edit existing user information.
    - Option to delete user accounts.
  + View Book Ratings: A section to view all book ratings submitted by users.
  + Manage Book Ratings (functionality to be defined): Based on requirements, this might involve editing or deleting ratings.
  + Add Book Ratings (potential): A form to add sample ratings or manage user ratings (functionality to be defined).

**5.2 Database Design**

The LMS will utilize a relational database management system (RDBMS) to store book data, user information, and book ratings. Here's a sample schema:

* **Users Table:**
  + id (INT, Primary Key)
  + username (VARCHAR, Unique)
  + password (VARCHAR)
  + email (VARCHAR)
  + is\_acceptable (bool)
  + is\_admin (bool)
* **Books Table:**
  + id (INT, Primary Key)
  + title (VARCHAR)
  + author (VARCHAR)
  + genre (VARCHAR)
  + poster (VARCHAR)
  + rate (Float)
  + status (bool)

**5.3 System Security**

* User passwords will be hashed and stored securely in the database

**5.4 Error Handling**

The LMS will implement error handling specific to the use cases:

* **Search Book:** Handle invalid search criteria entered by users (e.g., empty search bar).
* **View Book:** Provide informative messages if a searched book isn't found in the system.
* **View Book Rates/Checkout History:** Address scenarios where a user has no ratings or checkout history (display appropriate messages).
* **Admin Login:** Handle incorrect login credentials and potential account lockout after multiple failed attempts.
* **Admin Book Management:** Provide error messages for missing information when adding/editing books and confirmation prompts before deleting books.
* **Admin User Management:** Address errors related to user account management actions (e.g., trying to delete the admin account).
* **Admin Ratings Management:** Provide informative messages based on the chosen functionality (editing/deleting ratings).

**6. Testing Strategy**

The testing strategy will incorporate use case scenarios:

* **User Testing:** Focuses on functionalities exposed to users, like searching for books, viewing book details, and potentially managing user accounts and checkout history (if applicable).
* **Admin Testing:** Verifies functionalities specific to admins, including book management, user account management, and book rating management (based on chosen functionality).