

LIBRARY MANAGEMENT SYSTEM

SOFTWARE DESIGN DOCUMENTATION



September 20, 2025

DEEPANSHU MALAKAR, JELLA AMAN, KUMAR BHASKAR, JAYARUSHA K

B. tech CSE III YEAR

Group - 13

CONTENTS

|  |  |  |
| --- | --- | --- |
| S.NO | TITLE | PAGE NO |
| 1 | Introduction to Design | 2 |
| 2 | Dividing Design Requirements | 4 |
| 3 | Actors of the System | 5 |
| 4 | Classes and Attributes | 6 |
| 5 | User Interaction | 8 |
| 6 | Reports | 9 |
| 7 | Diagrams | 10 |

## 1.Introduction to Design

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

The Library Management System (LMS) is designed to automate library operations such as book issue, return, fine calculation, and report generation. This document provides a detailed blueprint for implementing the system.

🏆Goals of Design:

* **Modularity:** System divided into three parts.

1. **UI** – Different UI components like buttons, book icons, status bar, navbar, sidebar etc. should be present as a different module.
2. **Business Logic** – This section handles all the transactions like Issuing Books, Returning Books, Reserving Books, Imposing Fine etc. Every feature should be present in a separate module.
3. **Database layers --** This section will deal with the storage and manipulation of the data of all kinds, it includes the Insertion, Updation, Deletion of all data / records.

* **Efficiency:** Optimized book search and quick issue/return transactions.
* **Scalability:** Ability to handle thousands of users, books, and transactions.
* **Maintainability**: Easy to incorporate new features such as e-books.
* **Security:** Role-based access control for students, librarians, and admins.

🏆System-Specific Design Goals:

* **Intuitive user interface** for students, faculty, research scholars and librarians. An Explore page where students can view all the books present in the library.

1. A Navbar which groups similar features in different tabs.
2. Navbar tabs (Students / Faculty / Scholars) – Explore, Book center, Doubt section and Help & Support.
3. Navbar tabs (Librarian) – Book Management, Student Management, Reports and Help.
4. Every Tab of navbar has a sidebar that opens different page of software.
5. Sidebar Tabs:
6. Explore: Home, Favorites, History, Reserved Books, Study section
7. Book Center: Donate Book, Request Book.
8. Doubt Section: Raise Doubt, Solve Doubt, History, Ask Jarvis
9. Help and Support: No sidebar tab
10. Book Management: Issue Book, Return Book, Reserved Books, Due Books, Requested Books, Add Books.
11. Student Management: Student Details, Register New Student, Suspended Students, Blacklisted Students
12. Reports: No sidebar tabs.
13. Home page should contain all the books present in the library. User can scroll through the entire content of library.
14. Book Icons should contain book logo on left, book properties on right.
15. Search Bar to quickly get the required books.
16. Category bar: shows categories of books like {math’s, physics, data mining, all}. Clicking on any of these will show the books of only that particular category.
17. Status bar that shows user’s stats like total books issued, total fine due etc.

* **Fine calculation:**

1. System should automatically be able to calculate and keep track of fines for all the students.
2. Record Transaction ID for payment of fine and forward it to authorities for authentication.

* **Record Updation**:

1. Whenever a book is reserved or issued, change available copies by -1.
2. Whenever a book is returned then change the copies available by +1.
3. Track which users have paid the fine and who has not.
4. Whenever a new book is added then its details should be updated in the database.
5. Accounts of new users should be updated.

* **Role-based authentication:**

1. There are 4 roles (librarian, student, faculty, research scholar).
2. Whenever new user login or signup, ask their role and open a different login/signup page for each role.
3. This practice will increase security.

* **Report generation for Librarian:**

1. Total books issued per month graph.
2. Total books returned per month graph.
3. Total fine collected per month graph.

* **Notifications:**

1. When a student issues, returns or reserve a book, notification is sent to librarian.
2. When a new book is added, notification is sent to students.
3. When deadline is approaching then notification should be sent to students.

## 2. Dividing Design Requirements

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

📜 Non-Functional Requirements:

* **Book Issue/Return:** Students issue and return books; system updates availability.
* **Member Registration:** Register students/faculty with unique IDs.
* **Fine Calculation:** Automatic fine calculation for overdue returns.
* **Search Functionality:** Search by title, author, ISBN, or category.
* **Report Generation:** Generate transaction and usage reports.

📜 Functional Requirements:

* **Security:** Role-based login (Admin, Librarian, Student).
* **Performance:** Average search response < 2 seconds.
* **Scalability:** Handle 10,000+ users and 50,000+ books without slowdown.
* **Reliability:** Backup and recovery in case of crash.
* **Usability:** Simple GUI for non-technical users.

## 3. Actors of the System

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

* **🧑‍🎓Student:**

1. Search books
2. Request issue
3. Return books
4. Check fines.

* **📕Librarian:**

1. Manage inventory
2. Approve/reject request
3. Maintain fines
4. Generate reports
5. Manage accounts.

* **👨‍🏫Faculty:**

1. Search books
2. request issue
3. return books
4. check fines.

* **🎓Scholar:**

1. Search books
2. request issue
3. return books
4. check fines.

* **💻 System (Automated):**

1. Tracks availability
2. calculates fines
3. stores transaction history
4. sends notifications.

## 4.Classes and Attributes

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. 📦Entity Classes (Data Storage)

* **Book:**

1. Book ID
2. Title
3. Author
4. ISBN
5. Publisher
6. Edition
7. Availability Status.

* **Student:**

1. Student ID
2. Name
3. Email
4. ContactNo
5. Department
6. password

* **Transaction:**

1. Transaction ID
2. Book ID
3. Member ID
4. Issue Date
5. Due Date
6. Return Date
7. Fine.

* **Faculty:**

1. Student ID
2. Name
3. Email
4. ContactNo
5. Department
6. password

* **Scholar:**

1. Student ID
2. Name
3. Email
4. Contact No
5. Department
6. password
7. 🕹️Control Classes (Business Logic)

* **Login Manager:** Handles authentication and role assignment.
* **Search Manager:** Manages search queries.
* **Issue Manager:** Handles issue/return logic, fine calculation.
* **Report Manager:** Generates reports.

1. 🚧Boundary Classes (UI/Interfaces)

* **Student UI:** Search, issue, return, check fines.
* **Librarian UI**: Book management, approve/decline requests.
* **Admin UI:** Manage members/librarians, generate reports.

Entity Class Diagram

## User Interactions

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

* **Student Workflow:**

1. Login with credentials.
2. Search for books by keyword.
3. Request book if available.
4. Return book within due date.
5. View/update fines.

* **Librarian Workflow:**

1. Login with librarian credentials.
2. Manage book inventory (add/remove/update).
3. Approve/decline issue/return requests.
4. Update fines.

* **Admin Workflow:**

1. Login with admin account.
2. Create/manage librarian and student accounts.
3. Generate reports (transactions, usage, fines).
4. Monitor backups and system health.

## Reports

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

* **Transaction-Based Reports:**

1. **Daily Book Issue Report:** Books issued each day.
2. **Book Return Report:** Returned books per day.
3. **Fine Collection Report**: Fines collected daily/weekly/monthly.

* **Member-Based Reports:**

1. **Member Transaction Report:** All transactions by a student.
2. **Defaulter Report:** Students with overdue books/fines.

* **System-Wide Reports:**

1. **Popular Books Report:** Most issued books.
2. **System Usage Report:** Total logins, active users, search statistics.

## Diagrams

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

activity diagram for student borrowing a book

activity diagram for student return a book

Librarian Adding New Book

Librarian Issuing

Faculty Requesting New Book

Fine Calculation

Combined High Level Activity Diagram

Book Issue Object Diagram

Book IReturn Object Diagram

Reserve Book Object Diagram

Librarian Managing Catlog

Use Case Diagram