Library Management System

Software Requirement Specification

Prepared By :
Ayushi Thummar
B.Tech – Computer Engineering
RK University

Date : July 2025

Table of Contents

1. Introduction

1.1 Purpose of this document

1.2 Scope of this document

1.3 Overview

1.4 Definitions, Acronyms, and Abbreviations

1.5 Business Context

1.6 References

2. General Description

2.1 Product Functions

2.2 Similar System Information

2.3 User Characteristics

2.4 User Problem Statement

2.5 User Objectives

2.6 General Constraints

3. Functional Requirements

3.1 Librarian

3.2 Members

4. Interface Requirements

4.1 User Interfaces

4.2 Hardware Interfaces

4.3 Communications Interfaces

4.4 Software Interfaces

5. Performance Requirements

6. Other non-functional attributes

6.1 Security

6.2 Reliability

6.3 Maintainability

6.4 Portability

6.5 Extensibility

6.6 Reusability

1. INTRODUCTION

1.1 Purpose of this document

The purpose of this document is to clearly define the software requirements for the Library Management System. This system is designed to manage all participants and activities within a library, including book inventory, users, and transactions. The system allows users to view available books, reserve them online, and access services without physically visiting the library. All core library operations—such as issuing, returning, and tracking books—can be efficiently maintained through this digital system.

1.2 Scope of this document

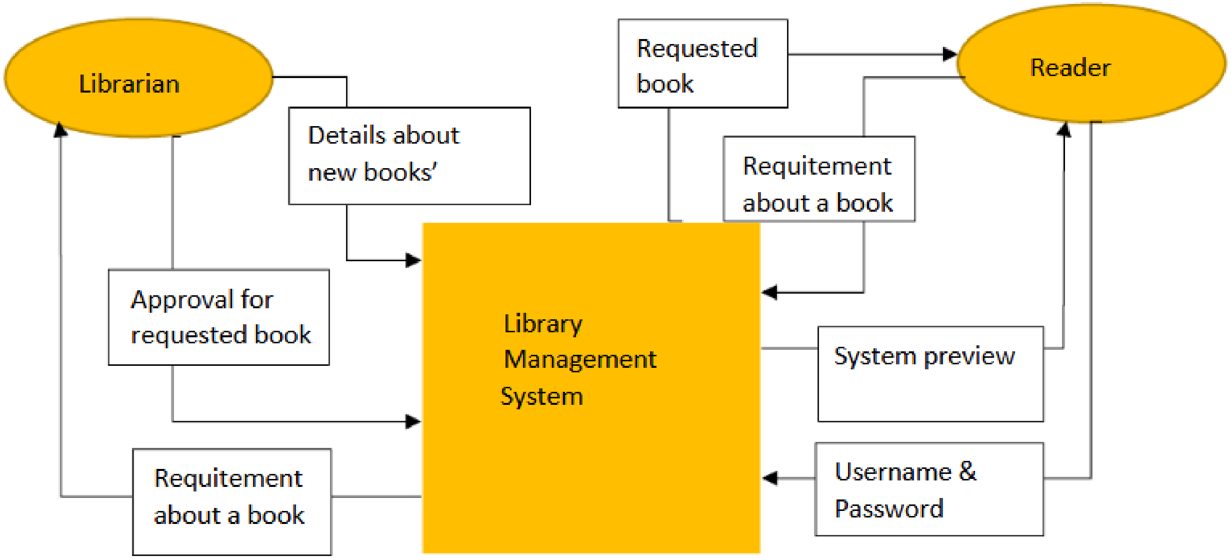
Library consists of thousands of books, and with the increase in the number of readers it needs to be well organized. So, this software will provide the facility of organization of books, monitoring them and controlling the transactions. It maintains the details of books and members. It provides easy searching for the available books in the library. It provides the functionality of reserving the books and adding new books. The main objective of this LMS software is to **simplify the day-to-day process of the library**, ensure **secure transactions**, enable **quick retrieval of information**, and provide **ease of use**.

1.3 Overview

It outlines both functional and nonfunctional requirements to provide the development team with a detailed understanding of the system's design and workflow. Our Library Management System will have two end users : **Librarian** and **Members**.

- **Librarians** will be able to check member's details, check in or check out books from the library, and manage the books.

- **Members** would be able to view their account, check for their details such as list of books borrowed, view the due date, and view the payment history.



1.4 Definitions, Acronyms, and Abbreviations

Abbreviation	**Full Form**
LMS	Library Management System
ISBN	International Standard Book Number
FR	Functional Requirement
GUI	Graphical User Interface
QR Code	Quick Response Code

1.5 Business Context

With the growing number of books and users, manual library management becomes slow and error-prone. The LMS offers a digital solution to automate book handling, user tracking, and transactions. It helps librarians work efficiently and allows members to access services like book search and reservations without visiting the library.

1.6 References

Websites :

- chatgpt.com
- library.rku.ac.in
- https://ndl.iitkgp.ac.in/
- https://www.gutenberg.org/
- en.wikipedia.org
- https://www.researchgate.net/publication/378267936_SRS_document_for_online_library_management_system
- https://spada.uns.ac.id/pluginfile.php/819931/mod_resource/content/1/W04%20Software%20Requirements%20Specification%20-%20Library%20Management%20System.pdf
- https://www.slideshare.net/slideshow/library-management-system-srs/83710046
- www.geeksforgeeks.org
- https://github.com/Majhar-98/SRS-on-Library-management-system/blob/main/SRS%20On%20Libraray_Management_System.docx%20(1).pdf

2. General Description

2.1 Product Functions

The Library Management System will offer the following core functions :

- Add, update, delete, and view book records.
- Issue and return books with due date tracking.
- Search for books by title, author, or ISBN.
- Register and manage user accounts (students and staff).
- Allow members to view borrowing history, due dates, and fines.
- Admin dashboard for managing books, users, and transactions.

2.2 Similar System Information

Traditional library systems rely on manual registers or spreadsheet files to track books and members. These systems are prone to human error, difficult to update, and time-consuming. Compared to that, this LMS offers a digital, automated solution with better organization, faster access to records, and improved accuracy.

2.3 User Characteristics

As discussed in Section 1.3, the system will have two main types of users : **Librarians** and **Members**.

- **Librarians (Admin) :**
  - Issue a book to a reader.
  - Can view books according to each category.
  - Add books and information about the books to the database and edit the information in the existing books.
  - Can access all the accounts of the library participants.

- **Members (Readers) :**
    - Can view each category of books available in the library.
    - Can view the available books category wise.
    - Can obtain a library account.
    - Can put in a request for a new book.

2.4 User Problem Statement

Users face difficulties with traditional library systems due to long queues, lack of real-time information, misplaced records, and limited access to services. Librarians struggle with manual updates, inefficient search processes, and data loss.

2.5 User Objectives

- To search, reserve, and borrow books quickly.
- To view issue history, return dates, and fines.
- To access the library system from anywhere (if web-based).
- To reduce manual work and errors in library operations.
- To ensure a smooth and user-friendly experience for both librarians and members.

2.6 General Constraints

- The system will require secure login authentication for both librarians and members.
- The software will be developed as a mobile application and used on Android/iOS devices, both inside and outside the library premises.
- Only authorized users will be allowed to perform specific actions (e.g., only librarians can add or remove books).
- Data storage and regular backups are essential to prevent data loss and ensure system reliability.
- The user interface must be simple and intuitive for users with basic computer skills.

3. Functional Requirements

3.1 Librarian

3.1.1 Admin Access

- **FR1 :** The system shall verify admin login credentials to grant access to administrative functionalities.

3.1.2 Register Members

- **FR2 :** The system shall store the registered member’s ID and personal information in the membership database.

3.1.3 Update Resource

- **FR3 :** The system shall allow the librarian to add, edit, or delete resource details including Resource ID, title, description, location, category, and author.

3.1.4 Check In / Check Out

- **FR4 :** The system shall record transaction details such as Resource ID, Member ID, Due Date, Checkout Date, Check-in Date, and Status.

3.2 Members

3.2.1 Logging In

- **FR5 :** The system shall verify valid member ID and password.
- **FR6 :** The system shall deny access to users with invalid credentials.
- **FR7 :** The system shall allow members with valid credentials to log in.

3.2.2 Search

- **FR8 :** The system shall allow members to search for available resources by Title, Resource ID, or Author Name.
- **FR9 :** The system shall display search results with detailed information about the resource.

3.2.3 Issue Book

- **FR10 :** The system shall allow members to request the issue of an available resource.
- **FR11 :** The system shall verify the member's borrowing eligibility (e.g., limit, account status) before processing the issue request.
- **FR12 :** The system shall update the resource status and associate it with the member upon successful issue.
- **FR13 :** The system shall display issue confirmation including resource details, issue date, and due date.

3.2.4 Renew Resource

- **FR14 :** The system shall allow members to renew a borrowed resource within 72 hours before the due date.

3.2.5 Manage / View Account

- **FR15 :** The system shall allow members to view their history, including a list of rented resources, due dates, renewal dates, and status.
- **FR16 :** The system shall allow members to update personal information such as phone number, address, email, and password.

3.2.6 Payment Management

- **FR17 :** The system shall display due payments, along with description and deadline.
- **FR18 :** The system shall provide a payment interface to collect and store member payment information.
- **FR19 :** The system shall verify payment details submitted by the member.
- **FR20 :** The system shall allow members to view their payment history including purpose, payment ID, and date of payment.

4. Interface Requirements

4.1 User Interfaces

The system will provide a mobile-friendly Graphical User Interface (GUI) with intuitive touch-based navigation, optimized for Android and iOS devices. Login screens will be available within the mobile app for both librarians (admin) and members.

Librarian Dashboard will include options to :

- Add, update, delete books.
- Issue/return resources.
- Register new members.
- View reports.

Member Dashboard will allow :

- Book search and reservation.
- Viewing borrowed resources and due dates.
- Viewing and editing personal profiles.
- Viewing payment history.

4.2 Hardware Interfaces

The Library Management System (LMS) will run on Android and/or iOS smartphones and tablets. The application will be designed to support devices with minimum hardware requirements, such as :

- A modern multi-touch screen.
- Minimum 4 GB RAM.
- Basic camera access (for barcode scanning).
- Stable Wi-Fi or mobile data connectivity.

The app may utilize the device camera to scan book barcodes for easier check-in/check-out and resource tracking.

4.3 Communications Interfaces

The Library Management System requires a stable and reliable internet connection to function properly. The system will connect to a centralized online database to store and retrieve user, resource, and transaction data.

Internet connectivity is essential for :

- Real-time book availability and updates
- Member login and data synchronization

4.4 Software Interfaces

- **Operating System** : Android 10+ / iOS 13+
- **Database System** : Firebase / SQLite (for local storage) / Cloud-based MySQL
- **Programming Language** : Flutter or React Native (cross-platform)

5. Performance Requirements

- The LMS mobile application shall respond to user interactions (e.g., log in, search, issue books) within 3 seconds under normal network conditions.
- The system shall support at least 100 concurrent users accessing the application.
- The app shall synchronize data (e.g., book availability, member activity) with the cloud/database within 5 seconds of performing a transaction.
- The application shall be optimized for performance on both mid-range and high-end Android/iOS devices.
- The system shall ensure minimal downtime and be available for 99% of the time, excluding planned maintenance.

6. Other non-functional attributes

6.1 Security

- Payment information should be protected and encrypted.
- Communication between a server and a user should be secure.

6.2 Reliability

The system has to be fully reliable due to the importance of data and the damage that can be done by incorrect or incomplete data.

6.3 Maintainability

The system should provide automatic notification to patrons by e-mail about items that are overdue, reservation results, availability of reserved items, etc.

6.4 Portability

The application shall run on a wide range of mobile devices with :

- Android version 10 and above.
- iOS version 13 and above.

6.5 Extensibility

- The system architecture will allow new features (e.g., QR-code login, chat support, e-book integration) to be added without rewriting core functionality.
- New user roles or modules (like event management or book reviews) can be integrated seamlessly.

6.6 Reusability

- Core components like authentication, search, and payment modules shall be developed as independent modules, reusable across future versions or similar systems.
- Code components shall follow standard design patterns to enhance reusability in other academic or inventory-based apps.