Software Requirements Specification (SRS) for Library Management System
Version 1.0

Prepared by
- Megha Shyam – SE22UARI071
- Aryan Karthik – SE22UARI022
- Aryan Reddy – SE22UARI116
- Krishna Bhardwaj – SE22UARI029
- Uthejini – SE22UARI129
- Nishitha – SE22UARI113

Instructor: Mr. Avinash Arun Chauhan
Course: Software Engineering (SE)
Date: 10/03/25

Revisions

Version	Primary Author(s)	Description of Version	Date
1.0	All Team Members	Initial draft of SRS for Library Management System	2025-03-09

Contents

---

1. Introduction

1.1 Document Purpose

This Software Requirements Specification (SRS) document defines the requirements for a Library Management System intended for a college library environment. It outlines both functional and non-functional requirements, serving as a guide for developers, testers, and stakeholders to ensure a clear understanding of the system's objectives and constraints.

1.2 Product Scope

The Library Management System is designed to automate and enhance core library processes, including:

- Book catalog management (adding, updating, removing records)
- Borrowing and returning books
- Overdue tracking and fine calculation
- Reporting on library usage

The system aims to streamline library operations, reduce manual errors, and provide convenient access to users and administrators.

1.3 Intended Audience and Document Overview

- Developers: Will use this SRS to understand and implement system functionalities.
- Testers: Will create test cases based on these requirements.
- Project Managers: Will verify that the requirements align with project goals and scope.
- College Librarians or Administrators: Will ensure the system meets the library's operational needs.

Document Overview:

1. Introduction – Purpose, scope, audience, definitions

2. Overall Description – High-level overview of system context, constraints, and assumptions
3. Specific Requirements – Detailed functional requirements, external interfaces, and use cases
4. Other Non-functional Requirements – Performance, security, quality attributes
5. Other Requirements – Additional details or domain-specific considerations
Appendices – Data dictionary and group log

1.4 Definitions, Acronyms and Abbreviations
- LMS: Library Management System
- User: Any individual using the system (student, librarian, administrator)
- Borrow: The process of checking out a book
- Return: The process of returning a previously borrowed book
- Fine: A penalty for overdue books

1.5 Document Conventions
- This document follows IEEE formatting standards.
- Requirements labeled FR refer to Functional Requirements, while NFR refer to Non-functional Requirements.
- All headings are in plain text, and standard font sizes (11 or 12 pt) are used throughout.

1.6 References and Acknowledgments
- IEEE 830-1998: Recommended Practice for Software Requirements Specifications
- College Library Policy Documents (if applicable)
- Acknowledgments to Instructor, Teaching Assistant, and any external sources consulted during requirements gathering

---

2. Overall Description

2.1 Product Overview
The Library Management System is intended to be deployed within a college's network infrastructure or via a cloud-based service. It will provide:
- A user interface for students and staff to search and manage books
- Administrative functions for librarians to oversee library operations
- Role-based access to ensure different privileges for students, librarians, and administrators

2.2 Product Functionality
1. User Management: Creation, authentication, and management of user accounts
2. Book Management: Adding, updating, and removing book records
3. Borrowing and Returning: Processing check-outs and check-ins of books
4. Overdue and Fine Tracking: Calculating and imposing fines for overdue items
5. Reporting: Generating various reports (overdue books, inventory status, usage statistics)

2.3 Design and Implementation Constraints

- Programming Language: Could be Java, Python, or a similar high-level language
- Database: MySQL or PostgreSQL
- User Interface: Web-based with HTML, CSS, and JavaScript
- Security: Must comply with basic authentication and access control policies
- Time and Resource Limits: This is a semester project with limited scope

2.4 Assumptions and Dependencies
- The college provides stable network connectivity
- Users have basic computer skills
- Email notifications (if implemented) rely on a functioning email service
- Adequate hardware resources (server, client machines) are available

---

3. Specific Requirements

3.1 External Interface Requirements

3.1.1 User Interfaces
- Web Interface: A user-friendly portal accessible via standard browsers
- Navigation: Clear menus for searching, borrowing, returning, and administrative tasks
- Administrator Dashboard: For librarians and administrators to manage books, users, and generate reports

3.1.2 Hardware Interfaces
- Desktop or Laptop: Access via browser
- Barcode Scanner (optional): For quick check-in or check-out
- Server Infrastructure: Hosting the database and the application

3.1.3 Software Interfaces
- Database: MySQL or PostgreSQL for storing and retrieving data
- Web Server: For serving application pages (for example, Apache or Tomcat)
- Email Service (optional): For sending notifications regarding due or overdue books

3.2 Functional Requirements

FR1: User Authentication and Authorization
- FR1.1: The system shall require valid credentials for each user.
- FR1.2: The system shall enforce role-based access (student, librarian, administrator).

FR2: Book Management
- FR2.1: Librarians shall be able to add, update, or remove book records.
- FR2.2: The system shall display real-time availability status for each book.

FR3: Borrowing and Returning
- FR3.1: Users shall be able to borrow available books, which updates the status to Checked Out.
- FR3.2: Users shall be able to return books, updating the status to Available.

FR4: Overdue and Fine Management
- FR4.1: The system shall calculate overdue fines based on a daily rate.
- FR4.2: The system shall maintain a record of outstanding fines for each user.

FR5: Reporting
- FR5.1: The system shall generate overdue reports listing items and fines.
- FR5.2: The system shall generate inventory and usage reports for administrators.

3.3 Use Case Model

3.3.1 Use Case 1: Borrow Book
Author: Megha Shyam
Purpose: Allow a user (student) to borrow an available book
Requirements Traceability: FR3.1
Priority: High
Preconditions: User is logged in; book is available
Postconditions: Book status becomes Checked Out
Actors: Member (primary), Librarian (secondary if confirmation is needed)
Flow of Events:
1. User searches for a book.
2. User clicks Borrow for an available book.
3. System confirms the transaction and updates the book status.
4. System sets a due date in the user's record.
Alternative Flow: If the book is unavailable or the user has exceeded their borrowing limit, the system displays an error.

3.3.2 Use Case 2: Return Book
Author: Aryan Karthik
Purpose: Allow a user (student) to return a borrowed book
Requirements Traceability: FR3.2
Priority: High
Preconditions: User has borrowed at least one book
Postconditions: Book status becomes Available. If overdue, a fine is recorded
Actors: Member (primary), Librarian (secondary if needed)
Flow of Events:
1. User navigates to My Borrowed Books.
2. User selects Return for the chosen book.
3. System updates the status to Available.
4. If overdue, the system calculates the fine and adds it to the user's record.

Alternative Flow: If no valid borrowing record is found, the system displays an error.

---

4. Other Non-functional Requirements

4.1 Performance Requirements
- NFR1: The system shall support up to 200 concurrent users without significant performance degradation.
- NFR2: Searches and book-related queries shall complete within 2 seconds under normal load.

4.2 Safety and Security Requirements
- NFR3: Passwords must be stored securely (encrypted).
- NFR4: Only librarians or administrators can modify book records or user details.
- NFR5: The system shall automatically log out users after 15 minutes of inactivity (if configured).

4.3 Software Quality Attributes

4.3.1 Reliability
- NFR6: Regular database backups shall be performed to prevent data loss in case of system failure.
- NFR7: The system should handle unexpected errors gracefully and log them for debugging.

4.3.2 Maintainability
- NFR8: The codebase shall follow a modular architecture to simplify updates and bug fixes.
- NFR9: Comprehensive documentation (in-code comments, developer guides) must be provided.

4.3.3 Usability
- NFR10: The user interface shall be intuitive, requiring minimal training for librarians.
- NFR11: Clear error messages and tooltips shall guide users in case of incorrect inputs.

---

5. Other Requirements
- Data Volume: The system must handle up to 50,000 book records and 10,000 user records.
- Internationalization (Optional): Future versions may support multiple languages.
- Legal: Must comply with basic data privacy guidelines and college IT policies.

Appendix A – Data Dictionary

Name	Type	Description	Related Operations
userID	Integer	Unique identifier for each user	CreateUser, UpdateUser, DeleteUser
bookID	Integer	Unique identifier for each book	AddBook, RemoveBook, UpdateBook
title	String	Title of the book	SearchBook, ViewBookDetails
dueDate	Date	The date by which a borrowed book is due	BorrowBook, ReturnBook
fineAmount	Float	Amount owed for overdue returns	CalculateFine, PayFine
status	String	Book status (Available, Checked Out)	BorrowBook, ReturnBook

---

Appendix B – Group Log

Date	Attendees	Topics Discussed	Action Items
2025-02-20	Aryan R., Krishna	Project outline, initial requirements	Each member to gather library data
2025-03-04	Uthejini, Nishitha	Draft functional specs, use cases	Draft initial SRS sections
2025-03-09	Megha, Aryan K	SRS review, finalizing details	Finalize SRS, prepare for submission

End of Document

This completes the Software Requirements Specification for the Library Management System. All requirements, constraints, and specifications have been documented without any bold text or asterisks.