# Group SRS - LibCore

Software Requirements Specification

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Revision** | **Description** | **Author** |
| 2/24/2025 | 1.0 | Initial Version | Thomas Tran, Alejandro Avina, Steven Le, Kartik Tripathi |
| 3/2/2025 | 1.1 | Additions to many segments of the document | Steven Le |
| 3/2/2025 | 1.2 | Additions and Modifications | Kartik Tripathi |
| 3/3/2025 | 1.3 | Additions & Modifications to requirements | Alejandro Avina |
| 3/3/2025 | 1.4 | Edited Table of Contents and added to Environmental Requirements | Thomas Tran |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Table of Contents

[1. Purpose 3](#_Toc1694314340)

[1.1. Scope 4](#_Toc602236455)

[1.2. Definitions, Acronyms, Abbreviations 4](#_Toc814300181)

[1.3. References 4](#_Toc19335591)

[1.4. Overview 4](#_Toc1495469905)

[2. Overall Description 4](#_Toc308466827)

[2.1. Product Perspective 5](#_Toc538158101)

[2.2. Product Architecture 5](#_Toc470395087)

[2.3. Product Functionality/Features 5](#_Toc1357825759)

[2.4. Constraints 5](#_Toc1507469583)

[2.5. Assumptions and Dependencies 5](#_Toc810167258)

[3. Specific Requirements 5](#_Toc1586058096)

[3.1. Functional Requirements 6](#_Toc1279835385)

[3.1.1. Common Requirements: 6](#_Toc779594918)

[3.1.2. User Management Module Requirements: 6](#_Toc867007469)

[3.1.3. Resource Management Module Requirements: 6](#_Toc41039956)

[3.1.4. Transaction Management Module Requirements: 6](#_Toc1597891318)

[3.2. External Interface Requirements 6](#_Toc2078520583)

[3.3. Internal Interface Requirements 7](#_Toc1182796163)

[4. Non-Functional Requirements 7](#_Toc709598879)

[4.1. Security and Privacy Requirements 8](#_Toc1343037124)

[4.2. Environmental Requirements 8](#_Toc894889411)

[4.3. Performance Requirements 8](#_Toc1569069778)

# Purpose

This document outlines the requirements for the LibCore Library Management System (LMS), a web-based system for managing library operations.

## Scope

The LMS will enable library members to register, search for available resources, and borrow or return them as needed. Library staff will have administrative privileges to manage inventory, approve transactions, and oversee user activities. The system will implement secure authentication and session management to prevent unauthorized access. Additionally, it will maintain a detailed loan history to track borrowed resources and overdue items, ensuring efficient management and accountability within the library.

## Definitions, Acronyms, Abbreviations

LMS: Library Management System

Admin: Library staff with special privileges

Member: Registered library user

Resource: Any book, magazine, research paper, or digital media

Loan History: A record of all borrowed resources

## References

Use Case Specification Document

UML Use Case Diagrams Document

Class Diagrams

Sequence Diagrams

## Overview

This document defines the system’s key functionalities, including user authentication, resource lending, inventory management, and loan history tracking. It also specifies performance, security, and usability requirements to ensure an efficient and secure library system.

# Overall Description

## Product Perspective

The LibCore LMS is a web-based application designed to streamline library operations and improve resource accessibility. It integrates multiple functionalities, including user management, inventory tracking, and transaction handling. The system replaces manual record-keeping with an automated, efficient, and scalable solution.

## Product Architecture

The system will be organized into three major modules:

User Management Module – Handles user authentication, registration, and session management.

Resource Management Module – Manages library inventory, including book details, availability status, and categorization.

Transaction Management Module – Tracks borrowing and returning processes, records loan history, and manages overdue notifications.

## Product Functionality/Features

The high-level features of the system include:

* User registration and authentication
* Secure session management
* Resource catalog search and filtering
* Borrowing and returning functionality
* Loan history tracking
* Inventory management for library staff
* Overdue notification system

## Constraints

SR1: The system must be accessible via web browsers without requiring any additional software installation.

SR2: User sessions must expire after a set period of inactivity.

SR3: All passwords must be stored securely using encryption techniques.

SR4: The system should be scalable to accommodate growing library needs.

SR5: No browser-specific code should be used to ensure cross-platform compatibility.

## Assumptions and Dependencies

It is assumed that the maximum number of concurrent users will not exceed 15,000.

The system depends on a stable internet connection for real-time updates and transactions.

It is assumed that users will have valid credentials to access their respective functionalities.

The system requires a relational database to store user, inventory, and transaction data.

# Specific Requirements

## Functional Requirements

### Common Requirements:

3.1.1.1 SR1 Users should be allowed to log in using their issued ID and password, both of which must be alphanumeric strings between 6 and 20 characters in length.

3.1.1.2 SR2 The system should provide HTML-based help pages on each screen that describe the purpose of each function within the system.

### User Management Module Requirements:

3.2.1 SR3 Library members should be able to register by providing their name, address, contact number, and unique membership ID.

3.2.2 SR4 The system must enforce secure authentication with a username and password for both members and staff.

3.2.3 SR5 User sessions should expire after a predefined period of inactivity to enhance security.

### Resource Management Module Requirements:

3.3.1 SR6 Members should be able to search for available resources using title, author, category, and availability.

3.3.2 SR7 Members should be able to request to borrow available resources.

3.3.3 SR8 Each loan transaction should record details such as resource ID, member ID, issue date, and due date.

3.3.4 SR9 Overdue resources should trigger automatic notifications to members.

3.3.5 SR10 Staff members should have the ability to override loan policies in special circumstances.

### Transaction Management Module Requirements:

3.4.1 SR11 The system should maintain a list of all library resources, including books, magazines, research papers, and digital media.

3.4.2 SR12 Each resource should have attributes such as title, author, publication date, category, and availability status.

3.4.3 SR13 When a resource is borrowed, its status should update to "Checked Out," and when returned, it should be marked as "Available."

3.4.4 SR14 Staff should have the ability to add, edit, and remove resources from the system.

## External Interface Requirements

3.5.1 SR15 The system should maintain a transaction log that records all loan activities.

3.5.2 SR16 Library members should be able to view their personal borrowing history, including details of currently borrowed and previously returned resources.

3.5.3 SR17 Staff should be able to view the loan history of all members, including overdue resources and frequently borrowed items.

3.5.4 SR18 The system should support filtering and sorting of loan records based on parameters like member ID, resource title, checkout date, and return status.

## Internal Interface Requirements

3.6.1 SR19 The system should provide an interface for exporting loan data in a standardized format (TXT File) for reporting and analysis.

# Non-Functional Requirements

## Security and Privacy Requirements

4.1.1 SR20 Logging of all important actions (Staff Failed Logins, etc.) (using TXT)

4.1.2 SR21 All passwords must be stored securely (e.g., hashed and salted).

4.1.3 SR22 Access levels should be enforced based on user roles (Member vs. Staff).

## Environmental Requirements

4.2.1 SR23 The system must be accessible via a web browser without requiring additional software installations.

4.2.2 SR24 The system must be hosted on a Linux-based server infrastructure.

4.2.3 SR25 The system must use an SQL-based relational database for data storage.

4.2.4 SR26 The system must be able to work on multiple operating systems such as Linux, Windows, and Mac.

## Performance Requirements

4.3.1 SR27 The system should handle multiple concurrent users without significant lag.

4.3.2 SR28 Search operations should return results within 2 seconds.

4.3.3 SR29 System must render all UI pages in no more than 9 seconds for dynamic pages, and static pages should be rendered in less than 3 seconds.