
Software Requirements Specification

for

Library Management System

Prepared by

Nourhan Ibrahim

Shahd Mohamed

Rehab Hamed

Rana Salem

Table of Contents

1.	Introduction	2
1.1	Purpose	2
1.2	Product Scope	2
2.	System Functions	3
2.1	Create main menu	3
2.2	Make a registration	3
2.3	Access librarian book management page	3
2.4	Access member book management page	4
3.	Nonfunctional Requirements	5
3.1	Security	5
3.2	Availability	5
3.3	Performance	5
3.4	Usability	5
3.5	Maintainability	5
3.6	Data Integrity	5
4.	Diagrams	6
4.1	Class diagram	6
4.2	Use case diagram	6
4.3	Activity diagram	6
4.4	Sequence diagram	6

1. Introduction

1.1 Purpose

*The purpose of this project is to design and implement a **Library Management System (LMS)** that automates the process of managing books, members, and borrowing operations in a library. The system is intended to provide **both librarians and members** with an easy-to-use console interface to perform their respective tasks efficiently.*

This system aims to:

- *Simplify book inventory management for librarians.*
- *Allow members to **search, borrow, and return books** efficiently.*
- *Track borrowed books and ensure data consistency.*
- *Provide a structured, OOP-based implementation suitable for learning and demonstrating software engineering principles.*

*The system is designed as a **console-based application using C++**, focusing on **object-oriented programming concepts** such as inheritance, polymorphism, and encapsulation.*

1.2 Product Scope

The Library Management System is intended for **small-scale libraries** or educational institutions that need a simple, reliable system to manage their book inventory and member transactions.

Scope includes:

- **Member Management:** Registration, login, and account management for library members.
- **Librarian Management:** Librarian login and book inventory management.
- **Book Management:** Adding, updating, removing, and searching books.
- **Borrowing Management:** Borrowing and returning books, tracking loan records, and handling book availability.
- **Console Interface:** Menu-driven interaction for both librarians and members.

Benefits:

- Reduces manual work for librarians.
- Improves accuracy in tracking borrowed books.
- Provides a clear, user-friendly interface for small libraries.
- Serves as a practical demonstration of **OOP principles and software engineering methodologies**.

2. System Functions

2.1 Create main menu

Code	Requirements statement	Must/should	Comments
FR001	Add a main menu that shows options for member login, librarian login, and exit	Must	
FR002	Add a welcome message displaying "Welcome to Library Management System"	Must	
FR003	Allow the user to navigate back to the main menu from any sub-menu	Must	

2.2 Make a registration

Code	Requirements statement	Must/Should	Comments
FR004	Add a function to allow new members to register with data: username and password.	Must	Check on username if it exists or not
FR005	Add a function that allows existing members to log in using username and password	Must	
FR006	Add a function to allow librarians to log in using librarian credentials	Must	

2.3 Access librarian book management page

Code	Requirements statement	Must/should	Comments
FR007	Add a function that allows the librarian to add a new book with details: book ID, title, author, category, and number of copies	Must	
FR008	Add a function to allow the librarian to update an existing book's details	Must	
FR009	Add a function to allow the librarian to remove a book by ID	Must	
FR010	Add a function to allow the librarian to view all books with their details	Must	
FR011	Add a function to search for books by title or ID	Must	

2.4 Access member book management page

Code	Requirements statement	Must/should	Comments
FR012	Add a function that allows a member to borrow a book if at least one copy is available	Must	<ul style="list-style-type: none">- Prevents borrowing unavailable books- Check if the user didn't exceed borrowing limit- Decrease available copies
FR013	Add a function that allows a member to return a borrowed book	Must	<ul style="list-style-type: none">- Increase available copies
FR014	Add a function to display all books currently borrowed by the member	Must	
FR015	Add a function to allow the librarian to view all books with their details		
FR016	Add a function to search for books by title	Must	

3. Nonfunctional Requirements

3.1 Security

Code	Requirements statement	Must/should	Comments
NR001	The system must authenticate members and librarians using username and password	Must	

3.2 Availability

Code	Requirements statement	Must/should	Comments
NR002	The system must be available 24/7	Must	

3.3 Performance

Code	Requirements statement	Must/should	Comments
NR003	The system must respond to user actions within 2 seconds	Must	

3.4 Usability

Code	Requirements statement	Must/should	Comments
NR004	The system must have a simple console menu-driven interface for ease of use	Must	
NR005	The system must display informative error messages for invalid operations	Must	

3.5 Maintainability

Code	Requirements statement	Must/should	Comments
NR006	The system must be designed modularly to allow future extension	Must	

3.6 Data Integrity

Code	Requirements statement	Must/should	Comments
NR007	The system must ensure that member data and book data are stored securely and not lost when the application is closed	Must	

4. Diagrams

4.1 Class diagram

4.2 Use case diagram

4.3 Activity diagram

4.4 Sequence diagram