**Library Management System**

**Software Requirements Specification (SRS)**

Team Members: Shekar, Naga Sai, Ajay, Vyshnavi, Bharadwaj

Date: [17-07-2025]

**Table of Contents**

1. Introduction

1.1 Purpose

1.2 Project Objectives

1.3 Scope

2. Functional Requirements

3. Non-Functional Requirements

4. System Models

5. Database Design

6. User Interface Design

7. Use Case Diagram

8. Entity Relationship Diagram

# Introduction

**Problem Statement**

1. There is no centralized system for tracking issued books, return dates, and availability status.
2. Administrators face difficulty in maintaining accurate and up-to-date records of members and books.
3. Users often struggle to find out whether a book is available or due without visiting the library physically.

**Project Objectives**

* To create a system that manages library books and members easily.
* To help the admin add, edit, issue, and return books quickly.
* To reduce manual work and avoid mistakes in book tracking.
* To allow users to find books and check availability easily.

**Scope**

* The system will handle book records and member registrations.
* It allows issuing and returning books with automatic updates.
* Search and filter features help users find books faster.
* Only admin will have full control; members can only view details.

# Functional Requirements

1. The system should allow adding, editing, deleting, and viewing books.
2. Members must be registered with their name, email, and a unique membership ID.
3. The system should support issuing books to members and updating return details.
4. Admin must be able to search for books using title/author.
5. Members can view available books and their status.
6. The system should validate inputs like email, ISBN, and availability before saving.

**3. Non-Functional Requirements**

**Usability**

The user interface must be simple, clean, and easy to use. Even non-technical users should understand how to operate the system.

**Scalability**

It should support a growing number of books and members over time. The system must perform well even if the data increases to thousands of records.

**Security**

Only authorized users should access the system with login. Sensitive data like member emails should be stored securely.

**Data Integrity**

Every action (like book issue/return) should update the database correctly. No duplicate or inconsistent data should exist in the system.

**4. System Models**

**User Roles & Permissions**

**Admin:**

* Can log in to the system.
* Can add, update, or delete book records.
* Can register and manage members.
* Can issue or return books.
* Can search and filter book listings.
* Can view reports or system logs (optional).

**Member ( User ) :**

* Can view list of available books.
* Can check issued books and return due dates.

**5. Database Design**A screenshot of a computer

AI-generated content may be incorrect.

**6. User Interface Design**

**6.1 Admin Screens**

* Login
* Dashboard
* Add/Edit/Delete Book
* Add/Edit/Delete Member
* Issue/Return Book

**6.2 Member Screens**

* View Books
* Search/Filter Books
* View Issued Books

**7. Use Case Diagram**

A diagram of a user

AI-generated content may be incorrect.

**8. ER Diagram**

A diagram of a member

AI-generated content may be incorrect.