

# Table of Contents

<b>1.Introduction.....</b>	<b>3</b>
1.1 purpose.....	3
1.2 Document Conventions.....	3
1.3 Intended Audience and Reading Suggestions.....	3
1.4 Project Scope.....	3
1.5 Definitions, Acronyms, and Abbreviations.....	4
1.6 References.....	4
<b>2. Overall Description.....</b>	<b>4</b>
2.1 Product Perspective.....	4
2.2 Product Functions.....	4
2.3 User Classes and Characteristics.....	4
2.4 Operating Environment.....	5
2.5 Design and Implementation Constraints.....	5
2.6 Assumptions and Dependencies.....	5
<b>3. External Interface Requirements.....</b>	<b>5</b>
3.1 User Interfaces.....	5
3.2 Hardware Interfaces.....	5
3.3 Software Interfaces.....	5
3.4 Communications Interfaces.....	6
<b>4. System Features (Functional Requirements).....</b>	<b>6</b>
4.1 User Authentication.....	6
4.2 Book Management.....	6
4.3 Book Issue and Return.....	6
4.4 Fine Calculation.....	7
4.5 Reviews and Ratings.....	7
<b>5. Non-Functional Requirements.....</b>	<b>7</b>
5.1 Performance Requirements.....	7

5.2 Security Requirements.....	7
5.3 Reliability and Availability.....	7
5.4 Maintainability and Scalability.....	7
5.5 Legal and Regulatory Requirements.....	7
<b>6. Other Requirements.....</b>	<b>8</b>
6.1 Database Requirements.....	8
6.2 Internationalization / Localization.....	8
<b>7. Appendices.....</b>	<b>9</b>
7.1 Analysis Models.....	9
7.1.1 Data Flow Diagram (DFD) .....	9
7.1.2 Entity Relationship Diagram (ERD) .....	9
7.1.3 State Transition Diagram.....	9
7.2 Detailed Use Cases.....	9
7.3 Requirements Traceability Matrix (RTM).....	10
<b>8. Detailed Functional Requirements Expansion.....</b>	<b>10</b>
8.1 User Management Module.....	10
8.2 Book Management Module.....	10
8.3 Transaction Management Module.....	10
8.4 Fine Management Module.....	11
<b>9. Expanded Non-Functional Requirements.....</b>	<b>11</b>
9.1 Usability.....	11
9.2 Scalability.....	11
9.3 Portability.....	12
9.4 Backup and Recovery.....	12
<b>10. Future Enhancements.....</b>	<b>12</b>
<b>11. Conclusion.....</b>	<b>12</b>

# 1. Introduction

The Software Requirements Specification (SRS) document provides a comprehensive and structured description of the Smart Online Library Management System. This document clearly defines the system's purpose, scope, functionality, constraints, and requirements. It serves as a formal agreement between stakeholders, project guides, developers, testers, and end users. The SRS acts as a foundation for system design, development, testing, deployment, and future maintenance.

In many educational institutions, library management is still carried out using manual registers or semi-automated tools. These methods are inefficient, time-consuming, and prone to errors. As the number of users and books increases, maintaining accuracy and efficiency becomes difficult. The Smart Online Library Management System is proposed as a complete end-to-end solution to address these challenges.

This document is prepared according to IEEE standards to ensure clarity, consistency, and completeness. It is written in a structured manner so that both technical and non-technical readers can understand the system requirements easily.

## 1.1 Purpose

The purpose of this SRS document is to describe in detail the functional and non-functional requirements of the Smart Online Library Management System. It provides a clear understanding of system behavior, interfaces, constraints, and performance expectations. This document will be used by developers during implementation, testers during validation, project guides for evaluation, and future teams for system enhancement and maintenance.

## 1.2 Document Conventions

This document follows IEEE SRS documentation conventions. All mandatory requirements are described using the keyword "shall". Optional or recommended requirements are described using "should". Headings and subheadings are numbered for easy reference. Diagrams, tables, and lists are used where necessary to improve readability and understanding.

## 1.3 Intended Audience and Reading Suggestions

This document is intended for multiple stakeholders:

- Project Guides and Evaluators should focus on the overall description, system features, and non-functional requirements.
- Developers should refer to functional requirements, interface requirements, and database specifications.
- Testers should focus on functional and non-functional requirements.
- End users can refer to product functions and user interface descriptions

## 1.4 Project Scope

The Smart Online Library Management System is a web-based application designed to automate library operations such as book management, user management, book issue and return, and fine calculation. The system provides real-time availability of books and ensures accurate record keeping. The scope of the system includes administrative control, member interaction, transaction management, and reporting.

The system does not include physical book maintenance, procurement processes, or offline operations. The main objective is to reduce manual effort, minimize errors, and improve efficiency in library operations.

## **1.5 Definitions, Acronyms, and Abbreviations**

- SRS: Software Requirements Specification
- E2E: End-to-End
- Admin: Library Administrator
- Member: Registered user of the library
- QR Code: Quick Response Code

## **1.6 References**

- Smart Online Library Management System Project PPT
- Smart Online Library Management System Project Report
- IEEE 830 / ISO/IEC/IEEE 29148 Standards

# **2. Overall Description**

## **2.1 Product Perspective**

The Smart Online Library Management System is a standalone web-based application that can be integrated with institutional portals in the future. It uses a centralized database to store all library-related data in a structured and secure manner. The system interacts with users through a web browser, eliminating the need for additional software installation. It supports optional QR code-based book transactions to simplify issue and return operations. This approach improves efficiency, reduces manual effort, and enhances system scalability..

## **2.2 Product Functions**

The major functions of the system include user registration and authentication, book catalog management, book search, book issue and return, automatic fine calculation, review and rating system, and administrative reporting. These functions work together to automate daily library activities. Each function is designed to reduce manual intervention and improve accuracy. The system ensures real-time data updates for all operations. These functions ensure smooth and transparent library management.

## **2.3 User Classes and Characteristics**

The system supports two main user classes: Administrator and Member. Administrators manage books, users, and transactions to ensure proper library operations. They are responsible for maintaining accurate records and monitoring system activity. Members interact with the system to search for books and perform issue or return operations. Members can also view their borrowing history and fine details.

## **2.4 Operating Environment**

The system operates in a web-based environment and can be accessed using modern web browsers such as Chrome, Firefox, and Edge. It does not require any platform-specific installation, making it easy to use. The application runs on servers that support database management systems. It is compatible with standard operating systems such as Windows or Linux. This ensures flexibility and wide accessibility.

## **2.5 Design and Implementation Constraints**

The system must be developed using secure authentication mechanisms to protect user data. It must follow standard web application development practices. The system should support database-driven operations for efficient data handling. Security constraints must be strictly followed to prevent unauthorized access. Performance constraints ensure fast response times and reliable system behavior.

## **2.6 Assumptions and Dependencies**

The system assumes stable internet connectivity for proper operation. Availability of server infrastructure is necessary to host the application. The system depends on database services for data storage and retrieval. Optional QR code scanning devices may be used for enhanced functionality. Changes in these assumptions may affect system performance and availability.

# **3. External Interface Requirements**

The External Interface Requirements describe how the Smart Online Library Management System interacts with users, hardware components, software components, and communication mechanisms. These interfaces define the boundaries between the system and external entities and ensure smooth interaction, usability, and interoperability. Properly defined interfaces help in reducing system complexity, improving user experience, and ensuring reliable data exchange between different components of the system.

## **3.1 User Interfaces**

The system provides a user-friendly web interface designed to support both administrators and members. The user interface is browser-based and can be accessed using standard web browsers without the need for additional software installation. The interface is designed with simplicity, clarity, and ease of navigation in mind, ensuring that users with minimal technical knowledge can operate the system efficiently.

The system includes separate dashboards for Admin and Members, each tailored to their specific roles and responsibilities. The Admin dashboard allows administrators to manage books, users, issue and return transactions, and view reports. The Member dashboard enables users to search for books, check availability, issue or return books, and view transaction history and fine details.

## **3.2 Hardware Interfaces**

The Smart Online Library Management System primarily operates as a software-based application and does not require specialized hardware for its basic functionality. However, the system supports optional hardware interfaces to enhance operational efficiency.

A QR Code scanner can be used optionally to simplify the book issue and return process. Each book can be associated with a unique QR code that contains its identification details. When the QR code is scanned, the system automatically retrieves the corresponding book information and processes the transaction, reducing manual data entry and minimizing errors.

## **3.3 Software Interfaces**

The system interacts with a Database Management System such as MySQL or a similar relational database. The database stores all essential information related to users, books, issue and return transactions, and fine details. The system performs create, read, update, and delete operations through backend services that communicate with the database.

A web server is used to host the application and manage incoming requests from client browsers. The web server processes requests, routes them to the backend application, and returns appropriate responses to the frontend. Proper integration between the web server, backend application, and database ensures smooth system operation and data consistency.

These software interfaces play a critical role in maintaining system reliability, security, and performance.

### **3.4 Communications Interfaces**

The system uses standard HTTP and HTTPS protocols for communication between client devices and the server. HTTPS is preferred to ensure secure transmission of sensitive information such as login credentials and user data.

The Smart Online Library Management System follows a REST-based communication approach between the frontend and backend. Requests and responses are exchanged in a structured format, enabling efficient data transfer and system scalability. This communication mechanism allows the system to maintain clear separation between presentation logic and business logic, ensuring flexibility and ease of maintenance.

## **4. System Features (Functional Requirements)**

This section describes the detailed functional requirements of the system. Each function is explained clearly to define how the system behaves during user interaction. The functional requirements focus on the core operations of the Smart Online Library Management System. These requirements ensure proper system functionality and smooth library operations. All requirements are written in a testable format to support system verification and validation..

### **4.1 User Authentication**

**Priority:** High

- The system shall allow users to register using valid credentials.
- The system shall provide secure login functionality.
- User credentials shall be verified before granting access.
- The system shall identify users as Admin or Member.
- Access shall be provided based on user role.

### **4.2 Book Management**

**Priority:** High

- Admin shall be able to add new books to the system.
- Admin shall update existing book details when required.
- Admin shall delete outdated or unavailable books.
- The system shall store all book information securely.
- The system shall display real-time book availability..

### **4.3 Book Issue and Return**

**Priority:** High

- Members shall be able to issue available books.
- The system shall record issue dates and due dates.

- Members shall return books using the system.
- The system shall update book status automatically.
- All transactions shall be stored in the database.

#### **4.4 Fine Calculation**

**Priority:** Medium

- The system shall calculate fines automatically.
- Fines shall be based on due dates.
- Late returns shall trigger fine calculation.
- Fine details shall be stored securely.
- Members shall be able to view fine information.

#### **4.5 Reviews and Ratings**

**Priority:** Low

- Members shall be able to rate books.
- Members shall submit reviews for books.
- Reviews shall be visible to other users.

### **5. Non-Functional Requirements**

Non-functional requirements define the quality attributes of the system. These requirements focus on how the system performs rather than what the system does. They ensure that the system operates efficiently, securely, and reliably under expected working conditions. Non-functional requirements help improve user satisfaction and system dependability. These requirements are critical for the long-term success of the system.

#### **5.1 Performance Requirements**

- Pages shall load within 2 seconds under normal load.
- Fast response time improves user experience.
- The system shall handle multiple user requests efficiently.
- Performance consistency shall be maintained during regular usage.
- This requirement ensures smooth system operation.

#### **5.2 Security Requirements**

- Passwords shall be encrypted to protect user credentials.
- Encryption prevents unauthorized access to sensitive data.
- Secure authentication improves system reliability.
- User data confidentiality is maintained at all times.
- Security mechanisms protect the system from misuse.
- Role-based access control shall be implemented.

#### **5.3 Reliability and Availability**

- System shall be available 99% of the time.

- High availability ensures continuous access for users.
- System downtime is minimized.
- Reliable operation builds user confidence.
- This requirement supports uninterrupted library services..

## **5.4 Maintainability and Scalability**

- System shall be available 99% of the time.
- High availability ensures continuous access for users.
- System downtime is minimized.
- Reliable operation builds user confidence.
- This requirement supports uninterrupted library services.

## **5.5 Legal and Regulatory Requirements**

- User data shall be handled securely and confidentially.
- Data privacy policies are followed strictly.
- Sensitive information is protected from unauthorized access.
- Data handling complies with institutional rules.
- This ensures ethical and legal system usage.

# **6. Other Requirements**

This section covers additional requirements that do not fit directly into functional or non-functional categories but are essential for system completeness. These requirements support proper system operation and data management. They ensure that the system remains reliable, usable, and adaptable. Other requirements help in maintaining data accuracy and future readiness. These requirements complement both functional and non-functional aspects of the system.

## **6.1 Database Requirements**

- Store book details, user details, transaction history, and fines. The database serves as the central repository for all system data. It ensures that information is stored securely and can be retrieved efficiently. Proper data organization improves system performance. Accurate data storage supports smooth library operations
- Ensure data integrity and consistency. The database enforces rules to prevent duplicate or invalid records. Relationships between data entities are maintained correctly. Data consistency is preserved during updates and transactions. This ensures reliable and error-free data management.

## **6.2 Internationalization / Localization**

The system supports English language initially. All system interfaces and messages are displayed in English. This ensures ease of use for the current user base. Language support can be expanded in the future. The design allows for easy addition of multiple languages.



## 7. Appendices

### 7.1 Analysis Models

#### 7.1.1 Data Flow Diagram (DFD)

The Data Flow Diagram represents the flow of information within the Smart Online Library Management System. At the highest level, the system interacts with two external entities: Administrator and Member. Inputs such as book details, user details, and transaction requests are processed by the system and stored in the database. Outputs include availability status, transaction confirmations, fine details, and reports.

Level 0 DFD shows the system as a single process handling all operations. Level 1 DFD further decomposes the system into processes such as User Management, Book Management, Transaction Management, and Fine Calculation.

#### 7.1.2 Entity Relationship Diagram (ERD)

The ERD describes the database structure of the system. Major entities include User, Book, Transaction, Fine, Review, and Admin. Relationships define how users issue books, how fines are generated, and how reviews are linked to books. Primary keys and foreign keys ensure data integrity.

#### 7.1.3 State Transition Diagram

State transition diagrams describe how a book moves through different states such as Available, Issued, Overdue, and Returned. These states help in tracking book lifecycle and fine calculation.

### 7.2 Detailed Use Cases

#### Use Case 1: User Login

**Actor:** Member/Admin

**Description:** User logs into the system using valid credentials.

**Precondition:** User must be registered.

**Postcondition:** User is redirected to the respective dashboard.

#### Use Case 2: Search Book

**Actor:** Member

**Description:** Member searches for books using title, author, or category.

**Outcome:** System displays matching results with availability status.

#### Use Case 3: Issue Book

**Actor:** Member

**Description:** Member issues a book using system interface or QR code.

**Outcome:** Transaction is recorded and book status is updated.

#### Use Case 4: Return Book

**Actor:** Member

**Description:** Member returns an issued book.

**Outcome:** System updates availability and calculates fine if overdue.

## **Use Case 5: Manage Books**

**Actor:** Admin

**Description:** Admin adds, edits, or deletes book records.

### **7.3 Requirements Traceability Matrix (RTM)**

The RTM ensures that every requirement is traced through design, development, and testing phases. It maps functional requirements to use cases and test cases, ensuring full coverage and quality assurance.

## **8. Detailed Functional Requirements Expansion**

### **8.1 User Management Module**

The User Management Module handles registration, authentication, authorization, and profile management. It ensures that only authorized users can access specific functionalities based on their roles. This module controls user access from account creation to deactivation. It maintains secure login mechanisms to protect user information. Proper user management improves system security and accountability. Handles user registration, authentication, and authorization.

- Ensures only authorized users can access the system.
- Admins can create and deactivate user accounts.
- Members can update their personal details.
- Enforces password complexity for security.

### **8.2 Book Management Module**

This module maintains complete book inventory. It manages all book-related information in a centralized manner. Accurate inventory helps in efficient library operations. It reduces errors related to book tracking. This module is essential for effective catalog management. Maintains complete inventory of books.

- Stores book details such as ISBN, title, author, and category.
- Supports bulk upload of book records.
- Prevents duplicate book entries.
- Ensures accurate and updated book data.

### **8.3 Transaction Management Module**

Manages issuing and returning of books. This module ensures accurate tracking of all borrowing activities. Automated transactions reduce manual errors. Complete transaction history is maintained securely. Transparency in library operations is ensured. Manages book issue and return operations.

- Restricts the number of books a member can issue.
- Records issue date, due date, and return date.
- Sends reminders for upcoming due dates.
- Maintains transaction history accurately.

## **8.4 Fine Management Module**

Automatically calculates fines. This module ensures accurate and consistent fine calculation. Manual errors are eliminated. Transparency in fine handling is maintained. It supports fair library policies. Automatically calculates fines for overdue books.

- Fine calculation is based on overdue days.
- Displays fine details to members.
- Ensures transparency in fine handling.
- Reduces manual calculation errors.

# **9. Expanded Non-Functional Requirements**

## **9.1 Usability**

The system shall be easy to navigate for first-time users. The interface is designed to be simple and intuitive so that new users can understand system functionality without training. Clear menus and labels guide users through tasks. Minimal steps are required to complete common operations. This improves user satisfaction and adoption.

- The system shall be easy to navigate for first-time users.
- Simple and clear menus improve understanding.
- Users can complete tasks with minimal steps.
- Instructions and labels guide user actions.
- This improves user satisfaction.
- Consistent UI design shall be maintained.

## **9.2 Scalability**

The system shall support growth in users and books without performance degradation. The architecture is designed to handle increasing data volumes. Additional users can access the system simultaneously without slowdown. The database can store a growing number of records efficiently. This ensures long-term system usability.

- The system shall support growth in users and books without performance degradation.
- It can handle increased data volume efficiently.
- Multiple users can access the system simultaneously.
- Performance remains stable as usage grows.
- This ensures long-term scalability.

### **9.3 Portability**

The application shall be accessible across different browsers and devices. Users can access the system using popular browsers without compatibility issues. The system does not depend on a specific platform. This allows flexibility in usage across devices. Portability improves accessibility and convenience.

- The application shall be accessible across different browsers and devices.
- It supports popular web browsers.
- No platform-specific installation is required.
- Users can access the system on different devices.
- This improves accessibility.

### **9.4 Backup and Recovery**

The system shall support periodic database backups. Regular backups protect data from accidental loss. Backup operations ensure data availability. Stored data can be restored when required. This improves system reliability.

- The system shall support periodic database backups.
- Regular backups prevent data loss.
- Backup data is stored securely.
- Data can be restored when required.
- This improves reliability.
- Recovery mechanisms shall be available in case of system failure

## **10. Future Enhancements**

The system can be enhanced by providing mobile application support for easier access. Integration with RFID technology can automate book issue and return processes. An AI-based book recommendation system can help users find relevant books. Online payment gateway integration can simplify fine payment. These enhancements will improve system efficiency and user experience.

- Mobile application support
- Integration with RFID technology
- AI-based book recommendation system
- Integration with payment gateways for fine payment

## **11. Conclusion**

The Smart Online Library Management System provides a comprehensive, secure, and efficient solution for automating library operations. By replacing manual processes with a digital platform, the system enhances accuracy, transparency, and user satisfaction. This SRS document serves as a complete reference for development, testing, and future enhancement of the system.