Software Requirement Specification

of

Library Book Borrowing System (LBBS)

Section 1: Introduction

This document specifies the Software Requirements Specification (SRS) for the **Library Book Borrowing System (LBBS).** It defines the functional and non-functional requirements of the system and provides a reference for all stakeholders involved in the development process.

This SRS covers the requirements for the entire system. The Library Book Borrowing System is not a subsystem of a larger project but a complete solution in itself, completely focused on managing library resources. It automates key library tasks and helps both readers and librarians to interact with the system more efficiently.

The intended users of this document include students or readers, who will use the system to check availability and borrow or return books, and librarians or administrators, who will manage the collection by adding and removing books. It is also meant for developers, who will implement the system, and testers, who will verify that the requirements are correctly fulfilled.

This SRS is organized into four sections: Section 1 (Introduction) explains the purpose and scope, Section 2 (Overall Description) provides an overview of the product and its environment, Section 3 (Specific Requirements) lists the functional and non-functional requirements, and Section 4 (External Interfaces) describes user, software, and hardware interfaces. For reading sequence, all readers should start with Sections 1 and 2 for a general overview, while project managers and librarians may focus on Section 2, and developers and testers should concentrate on Section 3 for technical details.

Project Scope:

- **Product Description:** The Library Book Borrowing System will manage a collection of books, allowing users to borrow and return them designed to managing library books efficiently by allowing borrowing, returning, and tracking availability in a simple way.

- **Relation to business goals:** Companies may also use such business purpose as engagement tools on websites or apps, and there is potential for ad-based revenue in mobile markets. Educational institutions can adopt such projects as examples for teaching programming.

Section 2: Overall Description

1. Product Perspective

- The Library Book Borrowing System (LBBS) is a new, self-contained system designed to manage library books efficiently by allowing users to check availability, borrow, and return books**.**
- It is a replacement system for the traditional manual record-keeping method, which often suffers from inefficiency, lack of scalability, and inaccurate data.
- The system improves performance and usability by automating book tracking, reducing human error, and maintaining records through file persistence.
- The LBBS is a standalone application and does not interact with external systems. However, it can be extended in the future to integrate with larger library management systems or online catalogues.

2. Product Features

The **Library Book Borrowing System** provides several key features designed to simplify and automate library operations. Major features include:

- **Book Management:** Add new books, remove old books, and maintain a list of available books with details such as title and author.
- **Availability Check:** View the complete list of books and check whether a book is currently available or borrowed.
- **Borrowing Function:** Allow users to borrow books if available and automatically update the status.
- **Returning Function:** Enable users to return borrowed books and restore availability status.

- **File Persistence:** Save and load all book records from a local file to ensure data is not lost when the system is closed.

3. User Classes and Characteristics

(a) Students / Readers (End Users)

- **Frequency of use:** Occasional, mostly when borrowing or returning books.
- **Functions used:** View book list, check availability, borrow books, return books.
- **Expertise:** Low technical knowledge (basic computer usage).
- **Privileges:** Limited access (can only borrow/return books, cannot add/remove).
- **Importance**: Most important user class, as the system is primarily designed for them.

(b) Librarians / Administrators

- **Frequency of use:** Daily to frequent usage.
- **Functions used:** Add/remove books, view all records, manage availability, monitor borrow/return transactions.
- **Expertise:** Moderate technical knowledge (familiar with library operations and basic software usage).
- **Privileges:** Full access rights to manage the system.
- **Importance:** Critical user class to maintain the system and support readers.

(c) Developers / Testers

- **Frequency of use:** Occasional, during system development, testing, or updates.
- **Functions used:** Implement system features, verify requirements, debug, and test functionalities.
- **Expertise:** High technical knowledge (Java programming, testing tools).
- **Privileges:** Backend/system-level access during development and testing only.
- **Importance:** Secondary but essential for ensuring correct functionality and long-term reliability.

4. Operating Environment <u>Hardware Platform:</u>

- Personal Computer or Laptop with at least 2 GB RAM and 1 GHz processor.
- Standard keyboard and monitor for user interaction.
- Local storage (Hard disk/SSD) to store data files (book records).

<u>Operating System:</u>

- Compatible with Windows (7/8/10/11), Linux distributions (Ubuntu, Fedora, etc.), or macOS.

- Requires Java Runtime Environment (JRE) version 8 or higher installed.

Software Dependencies:

- Java Development Kit (JDK 8+) for development.
- Java I/O and Serialization for file storage and data persistence.
- Text-based console interface (no external GUI libraries required at this stage).

Other Requirements:

- Must run offline without requiring internet connectivity.
- Should operate smoothly on low-resource systems (basic computers in educational institutes).
- Data files should be portable and usable across multiple OS platforms.

5. Design and Implementation Constraints

- **Programming Language Constraint:** The system must be developed in Java and follow Object-Oriented Programming (OOP) principles. Other programming languages are not allowed for this version.
- **Storage Constraint:** Book data must be stored in local files using Java File I/O or Serialization. No external database (e.g., MySQL, Oracle) integration is permitted in this version.
- **Interface Constraint:** The system will use a console-based interface (text menus). Graphical User Interface (GUI) is not required at this stage.
- **Hardware Constraint:** The software must run on basic PCs/laptops with a minimum of 2 GB RAM and standard CPU. Designed for single-user access; multi-user concurrency is not supported.
- **Security Constraint:** Only basic access control (e.g., preventing borrowing of already borrowed books) is required. No advanced encryption or authentication is implemented in this version.
- **Design Conventions:** The system must follow modular class design (e.g., Book class, Library class). Coding standards such as proper naming conventions and comments must be followed for maintainability.

Section 3: System Feature

Feature 1: User Authentication

- **Description and Priority**
  - **Description**: Provides secure access to the library system by verifying the user's identity before allowing access.
  - **Priority**: High
  - **Priority Components:**
    - ✓ Benefit : 9
    - ✓ Penalty : 9
    - ✓ Cost : 4
    - ✓ Risk : 8

- **Stimulus/Response Sequences**
  - User opens system then Login page appears.
  - System asks for credentials.
  - User enters input.
  - System validates from database.
  - If valid then Dashboard opens.
  - If invalid then Show error + retry (max 3 attempts).
  - After 3 fails then Account locked.

- **Functional Requirements**
  - **FR1**. System shall allow login with unique credentials.
  - **FR2**. System shall lock account after 3 failed attempts.
  - **FR3**. System shall show error messages for invalid input.
  - **FR4**. System shall separate access roles (Admin/Librarian/Student).
  - **FR5**. System shall log all authentication attempts.
  - **FR6**. System shall auto-logout after X minutes inactivity.

Feature 2: Search & Filter System

- **Description and Priority**
    - **Description:** Allows users to search books by title, author, ISBN, category.
    - **Priority:** High
    - **Components:**
        - ✓ **Benefit: 9**
        - ✓ **Penalty: 8**
        - ✓ **Cost: 3**
        - ✓ **Risk: 4**
- **Stimulus/Response Sequences**
    - User enters keyword in search bar.
    - System fetches data from DB.
    - Shows list with filters (author, year, subject).
- **Functional Requirements**
    - **FR1.** System shall support keyword-based search.
    - **FR2.** System shall provide filters (author, category, availability).
    - **FR3.** System shall show "No results found" message when applicable.

Feature 3: Book Availability Status

- **Description and Priority**
    - **Description:** Shows number of available and issued copies.
    - **Priority:** Medium
    - **Components:**
        - ✓ **Benefit:** 8
        - ✓ **Penalty:** 6
        - ✓ **Cost:** 3
        - ✓ **Risk:** 3

- **Stimulus/Response Sequences**
  - User searches book.
  - System shows availability (Available/Issued/Reserved).
- **Functional Requirements**
  - **FR1.** System shall display availability in real-time.
  - **FR2.** System shall update status upon issue/return.

Feature 4: Book Request & Reservation

- **Description and Priority**
  - **Description:** Allows students to reserve/request books online.
  - **Priority:** Medium
  - **Components:**
    - ✓ **Benefit:** 7
    - ✓ **Penalty:** 6
    - ✓ **Cost:** 5
    - ✓ **Risk**: 4
- **Stimulus/Response Sequences**
  - User selects book then Clicks "Reserve".
  - System marks as reserved.
  - When available then Notify user.
- **Functional Requirements**
  - **FR1.** System shall allow reservation of issued books.
  - **FR2.** System shall send notification when available.

Feature 5: Notifications & Alerts

- **Description and Priority**
    - **Description:** Sends alerts for due dates, issued books, reserved book availability.
    - **Priority:** High
    - **Components:**
        - ✓ **Benefit:** 9
        - ✓ **Penalty:** 8
        - ✓ **Cost:** 6
        - ✓ **Risk:** 5
- **Stimulus/Response Sequences**
    - **System checks due dates daily.**
    - **Sends SMS/Email reminders.**
- **Functional Requirements**
    - **FR1.** System shall send automated reminders before due date.
    - **FR2.** System shall notify on reservation availability.
    - **FR3.** System shall notify users of fines.

Feature 6: Reports & Analytics

- **Description and Priority**
    - **Description:** Generates reports of issued/returned books, fines, and usage stats.
    - **Priority: Medium**
    - **Components**
        - ✓ **Benefit: 8**
        - ✓ **Penalty: 7**
        - ✓ **Cost: 6**
        - ✓ **Risk: 5**

- **Stimulus/Response Sequences**
    - Admin selects "Generate Report".
    - System compiles data.
    - Report displayed/exported.
- **Functional Requirements**
    - **FR1.** System shall generate daily/weekly/monthly reports.
    - **FR2.** Reports shall include books issued, returned, fines collected.
    - **FR3**. Reports shall be exportable (PDF/Excel).

Section 4: External Interference Requirements

(a) User Interfaces

- Logical Design:
    - The system will provide a simple and intuitive interface.
    - Main modules include: Login Page, Dashboard, Book Catalog, Search/Filter, Issue/Return, Reports.
    - Navigation flow: Login → Dashboard → Modules → Logout.
- GUI Standards:
    - Responsive design (works on desktop, tablet, and mobile).
    - Consistent navigation bar on every page (Home, Search, Catalog, Reports, Help).
    - Corporate style guide: clean layout, minimalistic theme, institutional branding.
- Error Message Conventions:
    - Clear, user-friendly alerts.
        - Example: *"Invalid username or password. Please try again."*
        - Example: *"Book not available. You may reserve it."*
- Components Requiring UIs:
    - Login/Registration Page
    - Dashboard (different for Admin, Librarian, Student) ○ Book Catalog (add/update/delete/search books) ○ Issue & Return interface (barcode/QR option)
    - Report Generation page
    - Notifications & Alerts screen

- Accessibility Considerations:
    - Multi-language support (English, Bengali). ○ Large-text mode for visually impaired users.
    - Keyboard shortcuts for quick access.

(b) Hardware Interfaces

- The LMS primarily interacts with standard computing devices.
- Devices:
    - Computer (Desktop/Laptop) with Internet access.
    - Barcode/QR Code Scanner (optional, for quick issue/return).
    - Printer (for receipts, reports, and records).
    - Server hardware for database and web hosting.
- Data/Control Interactions:
    - Barcode scanner sends book ID to system.
    - System retrieves book details automatically.
- Protocols/Connections:
    - USB/Serial connections for barcode scanner.
    - LAN/Wi-Fi for client–server communication.

(c) Software Interfaces

- Database:
    - MySQL/PostgreSQL database for storing user, book, and transaction data.
- Operating System:
    - Windows/Linux for server hosting.
    - Cross-platform support for clients (Windows, macOS, Linux).
- Tools/APIs/Libraries:
    - REST APIs for communication between client and server. ○ Email/SMS Gateway APIs for notifications.
    - Barcode/QR Code libraries for scanning.
- Data Exchange Format:
    - JSON/XML for client-server communication.
    - Example: {"book_id": "BK2025", "status": "Issued", "due_date": "2025-09-01"}

(d) Communications Interfaces

- Methods:
  - Web-based communication over HTTPS.
  - Email and SMS services for notifications.
- Standards/Protocols:
  - HTTPS for secure communication. ○ SMTP for sending emails.
  - REST API calls for client–server interactions.
- Security Considerations:
  - Token-based authentication.
  - Encrypted communication (SSL/TLS).
  - Role-based access control.
- Performance Expectations:
  - Response time: ≤ 2 seconds for search queries.
  - Notifications delivered within 5 seconds of due-date trigger.
  - System should support 100+ concurrent users.

(e) Non-Functional Requirements

- Performance Requirements:
  - The system shall support up to 10,000 books and 5,000 registered users.
  - Average response time: ≤ 2 seconds.
  - 99% uptime reliability.
- Safety Requirements:
  - Data backup performed daily.
  - Power outage protection on server (UPS).
  - Prevention of data loss through transaction logging.
- Security Requirements:
  - Confidentiality: Encrypted passwords (hashing).
  - Integrity: Database transaction validation.
  - Availability: Role-based access, backup & restore features.
  - Compliance: Follows institutional IT policy and GDPR-like data protection.
- Software Quality Attributes:
  - Usability: Simple and intuitive UI for students and librarians.
  - Reliability: 24/7 availability with minimal downtime.
  - Maintainability: Modular design for easy updates.

- Portability: Runs on any standard web browser.
- Scalability: Supports future expansion to multiple libraries.

• Other Requirements ○ Database Requirements: Must support relational schema for books, users, and transactions.

- Internationalization: Must support English and Bengali.
- Legal Requirements: Must comply with copyright policies for digital resources (e-books, journals).
- Reuse Objective: The system can be reused for multiple institutions with minimal changes.

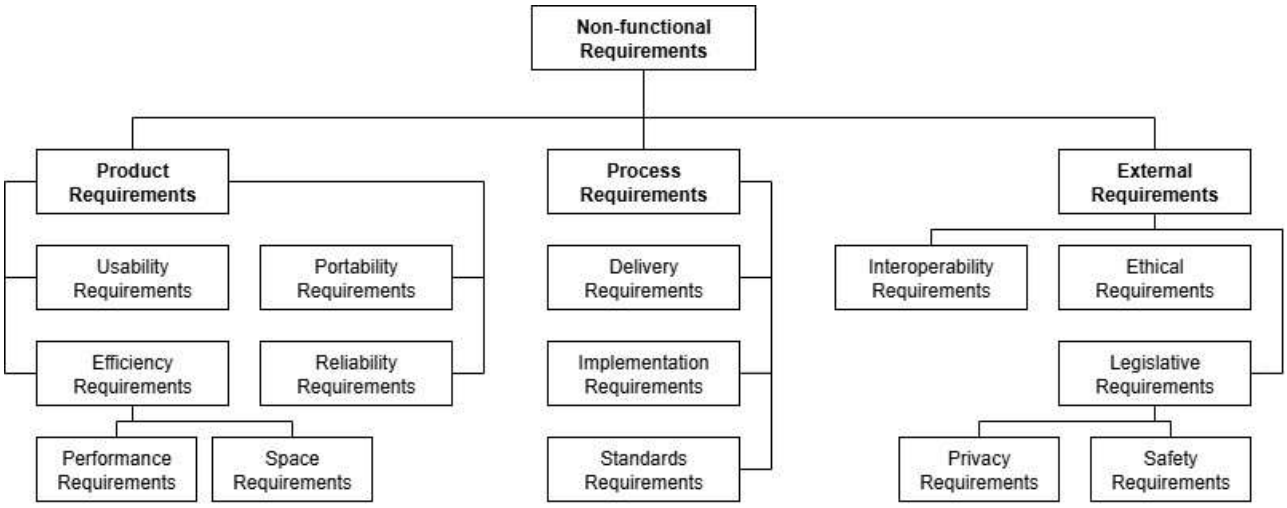

Figure 1: Non-functional Requirements of Library Book Borrowing System

Contributors:

Name	Student ID
Masfiq Rahman Misha	2303020
Faria Naz Tama	2303025
Md. Akash Mia	2303039