**Project Title:Online Book Lending System**

**Submitted to: Sir Ghulam Mujtaba**

**Submitted by: Janees Shahid**

**Reg No:FA22-BCE-003**

**Course: Software Engineering**

**Date:29th April,2025**

***1. Introduction***

**1.1 Purpose**

This document describes the requirements for developing an "Online Book Lending System." The aim is to create an accessible web platform that connects book enthusiasts by allowing them to lend books they own and borrow books they need. The system is designed for individuals, book clubs, or small libraries. It provides a virtual meeting place where users can register, list books for loan, request to borrow books, and manage lending activities. The goal of this project is not only to facilitate borrowing but also to encourage reading and sharing within a community.

**1.2 Scope**

The Online Book Lending System is envisioned as a user-friendly website where registered members can interact with one another through:

**Book Listings:**Members can list the books they are willing to lend.

**Search & Browse**:Users can browse through categorized book lists and use search filters (like title, author, or genre).

**Borrowing Requests**:Users can send and track borrowing requests, with an approval process in place to ensure fairness.

**Return & Feedback**:The system supports return confirmation and collects feedback to keep the exchange trustworthy.

This SRS covers the system’s functional behavior, the expected interactions between users and the system, as well as technical and performance requirements.

**1.3 Definitions, Acronyms, and Abbreviations**

**User**: A person who registers to lend or borrow books.

**Admin**:A system administrator who manages user accounts, book inventories, and overall system operations.

**Lending**: The act of a user offering his/her book for borrowing.

**Borrowing**: The process by which a registered user receives a book for temporary use.

**SRS:**Software Requirements Specification, this document.

**UI**: User Interface.

**1.4 References**

**IEEE Standard for Software Requirements Specifications:**

**->** Provides guidelines to structure and document system requirements.

**University Library System Documentation:** May serve as a practical reference if a similar system exists.

**Online Resources**:Articles and guides on modern web application development.

***2. Overall Description***

**2.1 Product Perspective**

The Online Book Lending System is intended as a new, standalone web application that can integrate with existing library databases if necessary. It is envisioned as a central hub for book exchanges among a community, allowing members to list their books, browse what others offer, and manage requests. The solution is not simply a catalog but a full system that handles user interactions, notifications, and the overall lifecycle of a lending transaction.

**2.2 User Needs and Product Functions**

Over the course of a typical user journey, several functions of the system address the needs of different user groups:

**For Lenders:**

- Easy registration and secure login.

- Ability to add detailed book information, including cover images, title, author, publication date, and a brief description.

- Management tools to track which books are currently on loan.

**For Borrowers:**

- A searchable and filterable book database.

- Simple process to request to borrow a book.

- Notification alerts when a request is approved or if the book becomes available.

**For Administrators:**

- Tools to manage the overall system, including adding or removing user privileges.

- Monitoring of book listings to ensure that all information is accurate and that community guidelines are followed.

- Administration over lending/borrowing disputes if they arise.

**2.3 Assumptions and Dependencies**

**User Assumptions**:

- All users have basic computer literacy and access to the internet.

- Users are honest and will return books within an agreed time frame**.**

**System Dependencies:**

- The system will run on popular web browsers and requires a stable internet connection.

- It relies on a back-end database (e.g., MySQL) for storing user information and book records.

- Integration may be required with email or SMS services for notifications.

***3. Specific Requirements***

**3.1 Functional Requirements**

**3.1.1 User Authentication and Account Management**

**Account Creation:**

- Users must be able to register by providing basic information (name, email, password). Verification steps (such as email confirmation) are mandatory.

**Login and Logout:**

- Secure authentication is required for every user session. The system must support password recovery.

**Profile Management**:

- Users can update their profile information, including contact details and preferences.

**3.1.2 Book Management**

**Add Book:**

- Lenders can add books to their listing. Each entry should include the book title, author, ISBN, cover image, category, condition of the book, and any additional remarks.

**Edit/Delete Book:**

- Users have the ability to edit or remove a book listing if the book is no longer available for lending.

**Book Search and Browse:**

- A robust search feature allowing filtering by title, author, category, or publication date.

**Book Details:**

- Detailed pages for each book listing, including user ratings and reviews.

**3.1.3 Borrowing and Lending Process**

**Request Submission:**

Borrowers can submit a request to borrow a listed book. The request includes the desired lending period and contact information.

**Approval and Notification:**

- The lender receives a notification about the borrowing request and can either approve or reject it. Upon approval or rejection, the borrower is notified.

**Loan Management:**

- A tracking system must record the status of each book loan (active, overdue, returned).

**Return Process:**

- Borrowers have a simple mechanism to mark books as returned, and the lenders can confirm the return.

**3.1.4 Communication and Notifications**

**Message Center:**

- A messaging feature for direct communication between lenders and borrowers regarding loan details.

**System Notifications**:

- Automatic notifications (via email or SMS) for updates such as request approvals, overdue reminders, and confirmation of returns.

**3.2 Non-functional Requirements**

**3.2.1 Usability and Accessibility**

- The user interface must be clear, consistent, and intuitive.

- The system should follow web accessibility standards (such as the WCAG guidelines) to ensure it is usable by people with disabilities.

- Mobile responsiveness is a plus even if the primary target is desktop users.

**3.2.2 Performance**

- Typical operations like searching for a book should complete within 3 seconds.

- The system should be scalable to support an increasing number of users and a growing database of books.

**3.2.3 Security**

- All sensitive user data must be stored securely using encryption.

- Secure communication protocols (like HTTPS) must be used to protect data transmitted over the network.

- Users should have multi-factor authentication (MFA) options to enhance account security.

**3.2.4 Reliability and Availability**

- The system should be available at least 99% of the time.

- Routine backups of the database are necessary to ensure data recovery in the event of a system failure.

**3.3 External Interface Requirements**

**3.3.1 User Interface (UI)**

- The UI should be designed with a clear layout: a homepage with featured books, user dashboards, and intuitive navigation menus.

- Use of friendly language in messages and help sections is important to guide users through the process.

**3.3.2 Hardware Interfaces**

-**Server**: The back-end server must be capable of handling concurrent requests from multiple users.

- **Client Devices**: The system should be usable from standard desktop computers, laptops, and tablets.

**3.3.3 Software Interfaces**

- **Database**: Connection to a SQL-based database (e.g., MySQL) to store and retrieve information.

- **Third-party APIs**: Integration with mailing services for notifications and a possible integration with social media accounts for login credentials.

**3.3.4 Communication Interfaces**

- **Networking**: The system should support standard TCP/IP protocols.

- **APIs**:RESTful web services may be implemented for inter-system communication if needed.

**3.4 Design Constraints**

- The system will be web-based and should work seamlessly on major web browsers (Chrome, Firefox, Edge, Safari).

- It must follow standard coding practices and be documented well to aid future enhancements.

- Data privacy regulations (like GDPR or local equivalents) must be adhered to when handling user data.

**3.5 Assumptions and Dependencies**

- **User Participation**:It is assumed that users are committed to respecting the lending periods.

- **Technical Environment**:The server environment is stable and has the necessary resources for hosting the website.

- **Third-party Services**: Availability of reliable third-party communication APIs for email and SMS notifications.

**4. System Architecture and Design Considerations**

**4.1 Overall Architecture**

The system employs a three-tier architecture:

- **Presentation Layer**: The web user interface built with HTML, CSS, and JavaScript frameworks (such as React or Angular).

- **Application Layer**: Backend logic implemented using a server-side language (such as PHP, Python, or Node.js) that handles business processes and user interaction.

- **Data Layer**: A relational database (like MySQL) that stores user information, book records, and transaction history.

**4.2 Data Flow and Process Diagrams**

- **User Registration and Login**:The process flow includes data entry, verification, account creation, and secure session management.

- **Book Listing and Search**:Users enter book details; the information is stored in the database. When a user searches, the application layer processes the query and returns results dynamically.

- **Loan Request Handling**:A flow diagram illustrates sending a request from a borrower, notification to the lender, and updating the status after approval.

**4.3 User Interface Mockups**

While the system is still in the design phase, rough sketches or mockups of the user dashboard, book detail page, and admin panel should be prepared. These mockups help in visualizing the navigation and overall user experience.

**5. Implementation Plan**

**5.1 Development Environment**

- **Tools & Languages**:

- Front-End: HTML, CSS, JavaScript (with possible framework like React).

- **Back-End**: A server-side language such as Python (using Django/Flask) or Node.js.

- **Database**: MySQL or PostgreSQL.

- **Development** **Tools**:

- Code repository (e.g., GitHub or GitLab) for version control.

- Integrated Development Environment (IDE) for coding and debugging.

- Testing tools for both unit and integration testing.

5.2 **Project Timeline**

- **Phase 1**: Requirements Gathering & Planning

– Detailing project requirements, mockups, and design reviews.

- **Phase 2**: Development – Building the front-end and back-end with iterative testing.

- **Phase 3**: Testing & Quality Assurance

– Rigorous testing of all modules; security and performance testing.

- **Phase 4:** Deployment & Maintenance

- Launching the system on a live server and planning for regular updates and bug fixes.

**5.3 Testing Strategy**

- **Unit Testing**: Each module will be individually tested to ensure it works as expected.

- **Integration Testing**:All modules (UI, back-end, database) will be tested collectively to check for interoperability.

- **User Acceptance Testing (UAT):**Real users will be invited to test the system in a controlled environment to collect feedback and ensure it meets user expectations.

***6. Maintenance and Future Enhancements***

**6.1 Maintenance Plan**

After deployment, the system will undergo regular maintenance:

**Software Updates**: Periodic updates to libraries and frameworks for improved security and performance.

**Bug Fixes**: A maintenance schedule will ensure that bugs reported by users are fixed promptly.

**System Monitoring**: Continuous monitoring will be set up to track system uptime and performance metrics.

**6.2 Future Enhancements**

Potential future features to expand the system include:

**Mobile Application Version:**A native app for Android and iOS to complement the web interface.

**Enhanced Social Features:** Integration of a rating and review system and social media sharing functionalities.

**Recommendation Engine:**Using user preferences and past borrowing history to suggest books.

**Integration with External Libraries:** Allowing the system to query public library databases for additional book listings.

***7. Appendices***

**7.1 Glossary**

**Lending**:The process by which a registered user offers their book for temporary use by another user.

**Borrowing**:The action of receiving a book from the system for a limited time period.

**Overdue**: A term used when a borrowed book is not returned by the agreed deadline.

**7.2** **Supporting Documents and Diagrams**

**Database Schema Diagrams**:Visual representations of the database structure.

**UI Mockup Diagrams**:Early sketches or design prototypes for the user interface.

**Flow Diagrams**: Detailed process flows for key functions like registration, request processing, and notifications.

**7.3** **Change Management Log**

This section will track changes to the SRS during development. Any modifications to requirements or design elements will be documented with dates and reasons for change.

**8.** **Summary and Approval**

This document outlines the detailed requirements for the "Online Book Lending System." It is intended to serve as a comprehensive guide for developers, testers, and stakeholders throughout the life of the project. Before beginning development, this document should be reviewed and approved by all key stakeholders to ensure that it captures the intended functionality and business logic.