Supplementary Specification for Library Management System (LMS)

Revision History

Version	Date	Description	Author
1.0	DEC 31,2024	First Draft	User

Introduction

This document captures the supplementary specifications for the Library Management System (LMS), detailing requirements not covered in the use cases.

Functionality

- **Catalog Management:** The system should support adding, updating, searching, and removing books from the catalog.
- **User Account Management:** Users can register, log in, and manage their profiles.
- **Borrowing and Returning Books:** The system will handle borrowing, renewal, and return processes, tracking due dates and fines.
- **Logging and Error Handling:** Log all errors to a persistent storage system for debugging and system monitoring.
- **Security:** Role-based access control to ensure data security and prevent unauthorized access.

Usability

- **Human Factors:**

    - The system interface should be intuitive and support both novice and expert users.
    - Fonts and colors should meet accessibility standards (e.g., contrast ratio for color blindness).
    - Responsive design to support desktops, tablets, and mobile devices.
- **Search Functionality:**
    - Users should be able to filter search results by author, title, genre, or availability status.

Reliability

- **Recoverability:**
    - The system must include backup mechanisms to prevent data loss in case of failures.
    - If the database server is unreachable, the system should provide offline access to limited functionalities.
- **Error Notifications:**
    - Users should receive clear error messages when operations fail.

Performance

- The system should handle up to 1,000 concurrent users without significant performance degradation.
- Search and retrieval operations should execute within 2 seconds for up to 10,000 records.

Supportability

- **Adaptability:**
    - The system should support additional features like e-books or audiobook lending in future updates.
- **Configurability:**
    - Administrators should be able to configure borrowing limits, fine rates, and notification preferences.

Implementation Constraints

- The system will use ReactJS for the front end, Node.js for the backend, and MySQL as the database.
- The system must adhere to privacy laws, such as GDPR or CCPA, for handling user data.

Interfaces

- **Noteworthy Hardware and Interfaces:**
  - Barcode scanner for book check-in/check-out.
  - Receipt printer for transaction receipts.
  - Signature pad for physical borrowing agreements (optional).
- **Software Interfaces:**
  - Integration with payment gateways for fine collection.
  - APIs for accessing external book metadata (e.g., ISBN lookup services).

Purchased Components

- None required initially. Open-source libraries will be leveraged.

Free Open Source Components

- Use open-source frameworks such as ReactJS and Express.js.
- Logging libraries like Winston or Bunyan for server-side logging.
