**Library System**

**Design Document**

**1. Introduction**

**1.1 Purpose of the System**

The Library Borrowing System is designed to streamline the borrowing and returning of books for library users while maintaining accurate records of all transactions. It provides library staff and users with an interface to manage book availability, user accounts, and borrowing history.

**1.2 Design Goals**

* **Efficiency:** Minimize response time for system operations.
* **Scalability:** Support a growing number of users and books.
* **Usability:** Offer a user-friendly interface for both library staff and users.
* **Security:** Ensure the integrity of user data and prevent unauthorized access.

**2. High-Level Software Architecture**

**2.1 Subsystem Decomposition**

The system is divided into the following subsystems:

* **User Management:** Handles user registration
* **Catalog Book Management:** Manages book inventory, including addition for books.
* **Borrowing System:** Tracks book checkouts, returns, and overdue records.
* **History tracking :** Generates reports for library books on system usage and inventory to know who borrowed what and when.

**2.2 Hardware/Software Mapping**

* **Client:** Web-based interface accessed via browser.
* **Server:** Application server hosting business logic and APIs.
* **Database:** Relational database for storing persistent data such as user accounts, book catalog, and transaction logs.

**2.3 Persistent Data Management**

Data is managed using a relational database schema:

* **Users Table:** Stores user information, including roles (e.g., staff, member).
* **Books Table:** Contains details about books, including availability.
* **Transactions Table:** Tracks borrowing and returning activities.

**2.4 Access Control and Security**

* Role-based access control (RBAC) ensures different access levels for users (e.g., staff can modify the catalog, members cannot).
* Data encryption for sensitive information like passwords and personal details.
* Input validation to prevent security threats.

**2.5 Boundary Conditions**

* **System Failure:** Automatic backup and recovery mechanisms ensure minimal data loss.
* **Peak Load:** The system can handle a high number of simultaneous transactions during peak hours.

**3. Low-Level Design**

**3.1 Object Design Trade-offs**

* **Relational Database vs. NoSQL:** Chose relational database for structured data and complex queries.
* **Monolithic vs. Microservices:** Opted for monolithic design for simplicity in initial implementation.

**3.2 Final Object Design**

Key objects include:

* **User:** Attributes include userID, name, role, password.
* **Book:** Attributes include bookID, title, author, Published Date ,Cover Photo, availability.
* **Transaction:** Attributes include BorrowingId, userID, bookID, borrowDate.

**3.3 Packages**

* **UserManagement:** Contains classes for user authentication and profile management.
* **CatalogManagement:** Handles book inventory operations.
* **BorrowingSystem:** Manages borrowing and returning processes.
* **History Reporting:** Generates history reports .

**3.4 Class Interfaces**

* **User:** Methods include login(), register().
* **Book:** Methods include addBook(), removeBook().
* **BorrowedBook:** Methods include borrowBook(), returnBook().
* **SystemLog:** Setters for each action occurs in the system.

**3.5 Design Patterns**

* **Factory Pattern:** Used for creating different types of users (e.g., Admin,User).
* **Singleton Pattern:** Ensures a single instance of database connection.

**4. Improvement Summary (Iteration 2 Only)**

(To be completed after the second iteration.)

**5. Glossary & References**

**Glossary**

* **RBAC:** Role-Based Access Control.
* **ORM:** Object-Relational Mapping.
* **API:** Application Programming Interface.
* **UML:** Unified Modeling Language.

**References**

* UML Diagrams:

Koç, H., Erdoğan, A. M., Barjakly, Y., & Peker, S. (2021, March). UML diagrams in software engineering research: a systematic literature review. In *Proceedings* (Vol. 74, No. 1, p. 13). MDPI.

* Design Patterns Cooper, J. W. (2000). Java design patterns: a tutorial.

Database Design: "Database System Concepts" by Silberschatz et al.