**Objective:**

Building a Library Management System.

**Users of the System:**

1.students

2.teachers

3.faculty and management members

**Functional Requirements:**

1.User Management

- The system should allow librarians to create and manage user accounts for library members.

- The system should provide authentication and authorization mechanisms to ensure only authorized users can access the system.

- The system should allow librarians to update user information, such as contact details and membership status.

2. Catalog Management

- The system should allow librarians to add, update, and delete books and other library resources in the catalog.

- The system should store and maintain information about each resource, including title, author, publication date, ISBN, and availability status.

- The system should support searching and filtering the catalog based on different criteria, such as title, author, and category.

3.Borrowing Management

- The system should allow library members to borrow and return books and other resources.

- The system should enforce borrowing rules, such as maximum borrowing limit and loan duration.

- The system should track the borrowing history of each member, including borrowed items, due dates, and return dates.

- The system should send reminders to members for overdue items and manage fines for late returns.

4 Reservation Management

- The system should allow library members to reserve books and other resources that are currently unavailable.

- The system should manage the reservation queue and notify members when a reserved item becomes available.

- The system should automatically cancel reservations if not picked up within a specified time.

5.Reporting and Analytics

- The system should provide reporting and analytics capabilities to generate various reports, such as borrowing history, popular books, and overdue items.

- The system should allow librarians to export reports in different formats, such as PDF or Excel.

**Non-Functional Requirements:**

**Security**

1.Performance

- The system should be able to handle a large number of concurrent users and transactions without significant performance degradation.

- The system should respond to user actions within an acceptable time frame.

2.Security

- The system should ensure the privacy and security of user data by implementing appropriate security measures, such as encryption and access controls.

- The system should provide user authentication and authorization mechanisms to prevent unauthorized access to sensitive data and functionalities.

3.Usability

- The system should have a user-friendly interface that is easy to navigate and understand.

- The system should provide clear and intuitive error messages to help users troubleshoot any issues.

4.Reliability

- The system should be highly reliable, with minimal downtime and data loss.

- The system should have backup and recovery mechanisms to protect against data loss or corruption.

**Constraints**

- The system will be developed using Java for the backend and HTML/CSS/JavaScript for the frontend.

It will use a MySQL database for data storage.

The system should comply with relevant data protection and privacy laws.

**Standard Features**

- The system should support logging(app/web/DB) at all levels.

**Browser Compatible**

- All latest browsers

**Technology Stack**

- HTML&CSS

- JavaScript

**Key points to remember:**

-Understand the specific needs and requirements of the library staff, administrators, and patrons.

-Create a detailed system design that includes database schema, user interfaces, and workflows.

-Create comprehensive documentation for the system, including user manuals and developer guides.

-Define user roles (e.g., librarian, admin, student) and implement access control mechanisms to ensure data security.

**Application assumptions:**

- The system assumes that the library has an existing infrastructure, such as computers, servers, and network connectivity, to support the software application.

- The system depends on the availability and accuracy of data sources, such as book databases and user information.

**Project Tasks:**

1.Requirements Gathering:

-Identify and document the specific requirements of the library, including cataloging, circulation, patron management, and reporting needs.

2.System Design:

-Create a detailed system design, including database schema, user interfaces, and system architecture.

3.Database Design:

-Design and implement the database structure to store information about books, patrons, transactions, and other relevant data.

4.User Interface Design:

-Develop user-friendly interfaces for both library staff and patrons, including features like search functionality, user registration, and account management.

5.Cataloging and Classification:

-Implement cataloging and classification systems for books, such as Dewey Decimal Classification or Library of Congress Classification.

-Develop functionality for checking in and checking out books, managing due dates, and handling holds and reservations.

6.Patron Management:

-Implement features for patron registration, account management, and fine calculation.

7.Search and Discovery:

-Create robust search and discovery mechanisms, including keyword search, advanced search filters, and suggestions.

8.Reporting and Analytics:

-Include tools for generating reports on library activities, such as circulation statistics, overdue items, and inventory management.

9.Security and Access Control:

-Ensure data security by implementing authentication and authorization mechanisms to protect sensitive information.

10.Integration with External Systems:

-Integrate with external systems or services, such as online databases, digital libraries, and interlibrary loan services.

11.User Training and Support:

-Provide training materials and support for library staff and patrons to effectively use the system.

12.Testing and Quality Assurance:

-Thoroughly test the LMS to identify and resolve any bugs or issues, ensuring it functions as expected.

13.Deployment:

-Deploy the LMS to production servers and make it accessible to library staff and patrons.

14.Maintenance and Updates:

-Continuously monitor and maintain the system, applying updates and improvements as needed.

15.Documentation:

-Create user and administrative documentation to guide users and library staff on how to use the system.

16.User Feedback and Enhancement:

-Collect feedback from users and stakeholders to identify areas for improvement and implement enhancements accordingly.

17.Data Backup and Disaster Recovery:

-Establish a data backup and disaster recovery plan to protect against data loss and system downtime.

18.Compliance and Regulations:

-Ensure that the LMS(library management system) complies with relevant regulations, such as privacy laws and copyright requirements.

**Validations:**

1.User Authentication: Ensure that only authorized users can access the system by validating their credentials (username and password).

2.Book Availability: Check if a book is available before allowing users to borrow it. Prevent borrowing if the book is already checked out.

3.User Role Validation: Verify that users have the appropriate roles (e.g., student, librarian, admin) to perform certain actions within the system.

4.Book ISBN Validation: Validate the ISBN (International Standard Book Number) to ensure it follows the correct format and is unique.

5.Membership Validity: Confirm that a user's library membership is valid before allowing them to borrow books.

6.Fine Calculation: Accurately calculate fines for overdue books based on the due date and return date.

7.Reservation Validation: Allow users to reserve books, but ensure that they do not exceed the maximum allowed reservations.

8.Book Information Completeness: Ensure that essential book information (title, author, publication date, etc.) is complete and accurate.

9.Book Quantity: Prevent users from borrowing more copies of a book than are available in the library.

10.User Contact Information: Validate and maintain user contact details to facilitate communication and notifications.

11.Search Validation: Validate search queries to avoid potential SQL injection attacks or search errors.

12.Date and Time Formats: Enforce a consistent date and time format for all system inputs and outputs.

13.Transaction Records: Maintain accurate transaction records for borrowing, returning, and renewing books.

14.Data Backup: Implement regular data backup procedures to prevent data loss in case of system failures.

15.Security: Protect sensitive user and book data by implementing proper security measures, including encryption and access controls.

These validations help ensure the reliability, accuracy, and security of a Library Management System.

**Backend:**

**Class and Method description:**

**Controller Layer:**

1.User Controller: Manages user-related interactions, such as user registration, login, profile management, and authentication.

2.Book Controller: Handles book-related operations, including adding, updating, deleting, and searching for books in the library.

3.Transaction Controller: Manages library transactions like checking out and returning books, viewing transaction history, and calculating late fees if applicable.

4.Reservation Controller: Deals with book reservation functionalities, allowing users to reserve books that are currently checked out.

5.Dashboard Controller: Provides statistics and reports related to library activities, such as the number of available books, borrowed books, overdue books, etc.

6.Notification Controller: Handles sending notifications to users, such as due date reminders, reservation notifications, and account-related alerts.

**Model Layer:**

1.User Model: Represents user data, including user information, authentication details, and user roles.

2.Book Model: Represents book information, such as title, author, ISBN, availability status, and copies.

3.Transaction Model: Represents the history of book transactions, including checkouts, returns, and due dates.

4.Reservation Model: Stores information about book reservations, including the user who reserved the book and the expected pickup date.

5.Notification Model: Contains data related to notifications, including content, recipient, and delivery status.

**Frontend:**

**Customer:** 

1.User Authentication:

Implement a login and registration system for library staff and patrons.

Use secure authentication methods to protect user data.

2.Dashboard:

Create a dashboard for library staff and patrons after logging in.

Staff dashboard should include options for managing library resources, users, and generating reports.

Patron dashboard should allow users to search for books, check their account, and place holds or borrow books.

3.Search and Catalog:

Develop a robust search functionality for users to find books and other resources.

Include filters for refining search results by author, title, genre, and more.

Display book details, including availability status, location, and call number.

4.User Profiles:

Enable users to view and update their profiles.

Patrons should be able to see their borrowing history, holds, and fines.

Staff profiles should include access to administrative tools.

5.Book Management:

Create forms for library staff to add, edit, and remove books from the catalog.

Implement barcode scanning or ISBN lookup for faster data entry.

Track book availability and due dates.

6.Check-Out/Check-In:

Design a user-friendly interface for patrons and staff to check out and return books.

Include features like renewing loans and viewing due dates.

7.Hold Requests:

Allow patrons to place holds on books that are currently checked out.

Notify patrons when their requested items become available.

8.Fines and Payments:

Implement a fines system for late returns.

Allow patrons to view and pay fines online.

9.Notifications:

Send email or push notifications for due dates, hold availability, and overdue items.

10.Accessibility:

Ensure that the front-end is accessible to all users, including those with disabilities, by following web accessibility standards (WCAG).

**By-**

**Sham**

**Vijay**

**Titus**

**Kishore**

**kashvith**