Functional and Non-Functional Requirements

Functional Requirements

The following are the **functional requirements** for the Library Management System:

1. **Admin Management**

   Admin (librarian) should be able to:

   - Add, update, and delete books.
   - View all the books in the library.
   - Search for books by title, author, or genre.
   - Manage member accounts (add, update, deactivate).

2. **Member Management**

   Members should be able to:

   - Register an account with required details (name, email, contact).
   - Login to their account.
   - View available books in the library.
   - Borrow and return books.
   - Check the status of their borrowed books.
   - View due dates and overdue books.

3. **Book Borrowing and Return**

   Members can borrow a book if:

   - The book is available.
   - The member has not exceeded the borrowing limit.
   - A member can return a borrowed book, and the system should:

     [Update the availability status of the book.]

     [Track the return date.]

4. **Notifications and Alerts**

   The system should send notifications to members:

   - When a book is about to be due for return.
   - When a borrowed book has exceeded the due date.
   - Admins should receive alerts about overdue books and member activities.

5. **Search and Filter Books**

   User should be able to search for books using different filters:

   - Search by book title, author, or genre.
   - Filter by available books or overdue books.

6. **Book Reservations**

   - Members should be able to place reservations on unavailable books.
   - Once a reserved book is returned, the member should be notified.

7. **Reports and Analytics**

   Admins should be able to generate reports on:

   - Books borrowed by members.
   - Overdue books.
   - Reservation history.

Non-Functional Requirements

The following are the **non-functional requirements** for the Library Management System:

1. **Performance**
    - The system should handle up to **1000 concurrent users** without performance degradation.
    - Page load time for the library interface should be **under 2 seconds.**

2. **Scalability**
    - The system should be scalable to handle more users, books, and transactions as the library grows.
    - The database should be able to scale horizontally, especially for larger libraries with more books and members.

3. **Security**
    - All user passwords should be **hashed** before being stored in the database.
    - The system should support **Role-Based Access Control (RBAC)** for different types of users (Admin, Member).
    - Sensitive information such as user emails and borrowing history should be encrypted.
    - The system should ensure secure login i.e. authentication and authorization.

4. **Usability**
    - The system should have an intuitive user interface that is easy to use for both admins and members.
    - The design should be **responsive** and work seamlessly across desktop and mobile devices.

5. **Availability**
    - The system should be available **24/7** with a maximum downtown of 1 hour per month for maintenance.
    - Regular backups of the database should be taken to ensure data availability in case of a disaster.

6. **Backup and Recovery**
    - The system should support regular **backups** of critical data such as member details and book records.
    - In case of failure, the system should provide automatic **recovery** within a reasonable time frame (preferably within 1 hour).

7. **Compliance**
    - The system should comply with **data protection regulations (such as GDPR or local laws)** to protect users' personal information.
    - Access to sensitive data should be logged for auditing purposes.

8. **Localization and Language Support**
    - The system should support multiple languages and be easy to localize for use in different regions.

Conclusion

This document outlines both the functional and non-functional requirements for the **Library Management System**. These requirements will guide the development process, ensuring that the system meets the needs of its users while being efficient, secure, and scalable.