**Pratical-3**

**AIM:** prepare a software Requirement specification (SRS) for the PULMS

**1. Functional Requirements**

* **User Management**

1. **User Registration:**
   * The system must allow users (students, faculty, library staff) to register using their university credentials (e.g., student ID, faculty ID, or email).
   * Users should be able to register with basic information: full name, email address, ID number, and password.
2. **User Authentication:**
   * The system must provide secure login functionality using username and password.
   * Password recovery must be available through email verification.
   * Role-based authentication should be implemented, with access privileges determined by user roles (student, faculty, librarian, admin).
3. **Profile Management:**
   * Users should be able to update their personal details such as address, phone number, and email.
4. **Role-based Access Control:**
   * **Administrator**: Full access to manage system settings, user accounts, and library resources.
   * **Librarian**: Can manage book catalog, issue/return books, and generate reports.
   * **Faculty/Student**: Can search for resources, borrow and return items, reserve books, and view their borrowing history.
   * **Guest**: Can browse the catalog but cannot borrow or reserve items.

* **Catalog Management**

1. **Add/Update/Delete Books:**
   * Librarians should be able to add new books, update existing book details, or remove books from the catalog.
   * Each book should have details like title, author(s), publisher, publication year, ISBN, category, and number of copies.
2. **Book Availability:**
   * Each book must have an availability status: "Available," "Checked out," or "Reserved."
   * Books should be categorized (e.g., Fiction, Science, History) for easier search.
3. **Barcode Scanning:**
   * The system must support barcode scanning for quick addition and check-out of books.
4. **Multimedia Resources:**
   * The system should also support the cataloging of non-book resources like journals, e-books, or audiobooks.

* **Book Borrowing and Returning**

1. **Book Checkout:**
   * Users can borrow available books for a predefined time period (e.g., 14 days for students, 30 days for faculty).
   * A user can borrow multiple books simultaneously, but there should be a maximum limit (e.g., 10 books).
2. **Book Return:**
   * The system must allow users to return books on or before the due date.
   * Users should receive a confirmation of the return.
3. **Overdue Management:**
   * If a book is not returned on time, an overdue fine should be applied based on university policies (e.g., $1/day).
   * Notifications for overdue items should be sent to users via email or in-app notifications.
4. **Reservations:**
   * Users should be able to reserve a book that is currently checked out.
   * When the book becomes available, users should be notified.

* **Search Functionality**

1. **Search Books:**
   * The system should allow users to search the catalog by title, author, ISBN, category, or publication year.
2. **Filters:**
   * The search results should support filtering by availability (e.g., show only available books), publication year, or author.
3. **Sorting:**
   * Search results should be sortable by title, author, or publication date.

**2. Non-Functional Requirements**

* **Performance Requirements**

1. **System Responsiveness:**
   * The system should respond to user actions (such as search queries and borrowing requests) within 2 seconds.
2. **Concurrent Users:**
   * The system must support at least 1000 concurrent users without significant performance degradation.
3. **Data Consistency:**
   * The system should ensure that transactions (like borrowing and returning books) are processed consistently across multiple users.

* **Security Requirements**

1. **User Data Protection:**
   * The system must encrypt sensitive user data such as passwords, emails, and payment details.
2. **Authentication:**
   * Passwords must be stored using secure hash algorithms (e.g., bcrypt).
   * The system must support multi-factor authentication (MFA) for admins and staff accounts.
3. **Role-Based Access Control:**
   * Users should only have access to data and features that are appropriate for their role in the system.
4. **Audit Logs:**
   * All user actions (such as borrowing/returning books, updating records) should be logged for auditing purposes.

* **Maintainability and Extensibility**

1. **Modular Design:**
   * The system should be modular, allowing easy updates and future enhancements.
2. **Code Documentation:**
   * The codebase should be well-documented to allow future developers to maintain and enhance the system with ease.

### Quality Attributes for PULMS (Public University Library Management System)

Quality attributes are non-functional properties that describe how well the system should perform in various aspects such as performance, security, usability, and maintainability. Below are the key quality attributes for the ***Public University Library Management System (PULMS)****.*

**1. Performance**

* **Response Time**
* The system must provide fast responses for all interactions. For example:
  + Search queries should return results within ***2 seconds*** for users.
  + Actions such as borrowing, returning, and reserving books should be processed within ***1-2 seconds***.
* **Scalability**
* The system should be able to handle a growing number of users and resources without significant degradation in performance.
  + The system should support ***up to 1000 concurrent users*** without affecting the user experience.
  + As the number of library resources increases (e.g., books, journals), the system should maintain performance by scaling horizontally or vertically.
* **Load Handling**
* The system should be capable of handling high loads, especially during peak usage times (e.g., during exams or semester start).
  + The system must support ***high traffic spikes*** without crashing, slowing down, or becoming unresponsive.

**2. Security**

* **Data Privacy and Protection**
* All sensitive user information, including personal details, passwords, and payment information, must be encrypted both at rest and in transit.
  + Use ***strong encryption algorithms*** such as AES-256 for data storage and ***SSL/TLS*** for secure data transmission.
* **Authentication and Authorization**
* Implement ***strong user authentication***, using secure password hashing (e.g., bcrypt) and optional multi-factor authentication (MFA) for administrator and librarian accounts.
  + Role-based access control (RBAC) should restrict access to resources based on the user's role.
* **Audit Trails**
* The system must log all user actions (e.g., borrowing, returning, book catalog updates) for security and auditing purposes.
  + Logs should be accessible by administrators to track malicious activities or system misuse.
* **Vulnerability Protection**
* The system should undergo regular ***security testing*** (e.g., penetration testing) to identify and fix vulnerabilities.
  + Implement protections against common threats such as ***SQL injection****,* ***Cross-Site Scripting (XSS)****,* ***Cross-Site Request Forgery (CSRF)***, and ***brute force attacks****.*

**3. Usability**

* **Ease of Use**
  + The user interface (UI) should be intuitive and easy to navigate for all user types (students, faculty, librarians, administrators).
  + The ***navigation*** should be clear, with easy access to key functions such as searching, borrowing, returning, and managing the catalog.
* **User Interface Design**
* The system must have an attractive and consistent design across all devices (web and mobile).
  + The interface should follow modern ***UI/UX best practices*** for accessibility, readability, and interaction.
* **Accessibility**
* The system should be accessible to users with disabilities.
  + It should comply with accessibility standards such as ***WCAG 2.1*** to ensure compatibility with screen readers and other assistive technologies.
  + Use clear fonts, proper color contrasts, and keyboard navigability.
* **Mobile Usability**
* The system should offer a ***mobile app*** (iOS and Android) with similar functionality to the web version.
  + The mobile app must be optimized for various screen sizes and responsive interactions.

**4. Availability**

* **System Uptime**
  + The system should be***available 99.9% of the time****,* ensuring minimal downtime.
  + The system must support ***automatic failover*** in case of hardware or network failure to minimize downtime.
* **Disaster Recovery**
* The system should have a disaster recovery plan in place to recover from catastrophic events like server crashes or data corruption.
  + ***Backup***: Data should be backed up regularly (e.g., daily) and stored securely. Backups should be tested regularly for integrity.
  + ***Failover***: In case of server failure, the system should automatically switch to a backup server with minimal user disruption.

**5. Maintainability**

* **Modular Design**

The system should be built with a ***modular architecture*** to ensure ease of updates and scalability.

* + The application should be split into independent modules or microservices that can be updated or scaled without affecting other parts of the system.
* **Code Quality and Documentation**

The codebase should be ***well-structured*, *readable***, and adhere to industry best practices.

* + **Documentation**: Code should be thoroughly documented to ensure future developers can easily understand and maintain it.
  + There should be clear ***API documentation***for any integrations with external systems or services (e.g., payment gateways, external authentication systems).
* **Error Handling**

The system should have robust error handling and clear, actionable error messages.

* + Users should not encounter raw system errors; instead, they should see ***user-friendly messages*** like “We are experiencing a temporary issue. Please try again later.”

**Conclusion:**

The ***Public University Library Management System (PULMS)*** is a comprehensive solution designed to streamline library operations and enhance the user experience for students, faculty, and staff.

The system's robust ***performance****,* ***security***, and ***usability*** features, including role-based access, data encryption, and mobile support, guarantee a secure, responsive, and user-friendly environment. With ***scalability****,* ***reliability***, and ***maintainability***in mind, PULMS is built to grow with the institution and adapt to future needs.