**Team: SANKALPA**

**Project Title: LIBRARY MANAGEMENT SYSTEM**

**Project Start Date: 18TH SEPTEMBER,2023**

**By:**

* Aastha
* Adity
* Devansh
* Divyanshi Bhatia
* Ravi
* Sakshi

**Requirement Analysis**

## **1.Introduction**

* LMSs are software programs that assist libraries in managing their operations and collection management. We have modules for circulation, purchases, serials administration, reporting, and cataloguing.
* Automating library activities including cataloguing, circulation, acquisitions, membership management, and reporting is the major goal of a library management system. It strives to provide superior experiences for both library personnel and users while enhancing the accessibility, tracking, and consumption of library resources.

### **1.1. Purpose**

This document's goal is to provide an overview of the standards and requirements for creating a library management system (LMS).

### **1.2. Scope**

This system will handle all aspects of running a library, including user management, circulation, cataloguing, and reporting.

### **1.3. Users**

* Librarian
* Library Staff
* Library Member

### **1.4. Success metrics**

User Experience and Accessibility/Ease of use:

* **Objective:** Provide an intuitive and accessible user interface.
* **Success Criteria:** LMS is user-friendly, follows UX best practices, and complies with accessibility standards (e.g., WCAG).

Data Security and Privacy:

* **Objective:** Safeguard user and library data.
* **Success Criteria:** LMS incorporates robust security measures, data encryption, role-based access control, and complies with data protection regulations.

Search and Retrieval Efficiency/ Easy to Admin:

* **Objective:** Enable efficient resource search and retrieval.
* **Success Criteria:** LMS provides quick and accurate search results for titles, authors, subjects, ISBN, etc.

Scalability and Performance:

* **Objective:** Ensure scalability while maintaining performance.
* **Success Criteria:** LMS performs consistently and responsively as the collection and user load increase.

Compliance with Legal Requirements:

* **Objective:** Comply with legal and regulatory requirements.
* **Success Criteria:** LMS adheres to relevant laws governing libraries, copyright, intellectual property, and digital resources.

Feedback and Improvement:

* **Objective:** Continually improve the LMS based on user feedback.
* **Success Criteria:** Structured feedback mechanisms are in place, leading to regular updates and enhancements aligned with user needs

### **1.5. Precedent of Application:**

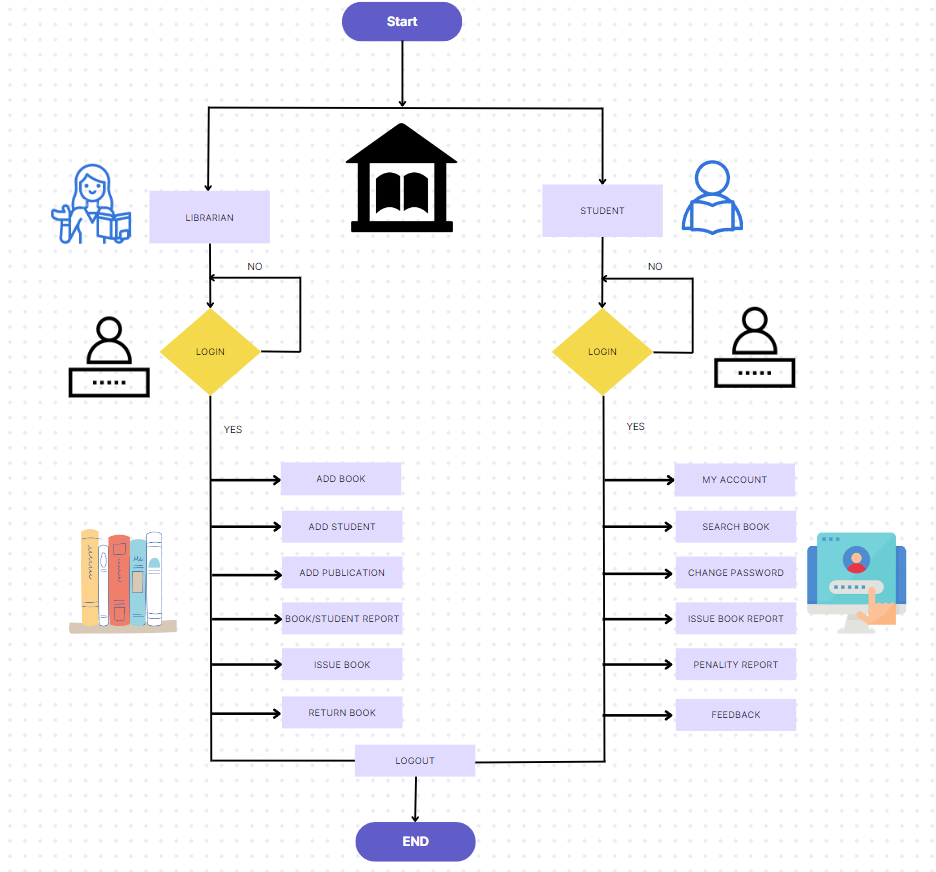
LMS applications encompass a wide range of functionalities, and there are several software applications available in this domain. Some of those are:

* **Atriuum ILS:** Atriuum is a cloud-based Integrated Library System (ILS) designed to work with any library and is built on over 30 years of library technology and library automation experience. Atriuum provides advanced features such as configurable user interfaces, customizable reports, mobile searching, and eBook resource sharing.
* **Follett Destiny:** Follett Destiny is a library management solution designed for K-12 schools. It includes features for cataloging, circulation, textbook management, and resource discovery.
* **Virtua:** Virtua is an ILS by VTLS, primarily used by academic and research libraries. It offers modules for cataloging, circulation, and resource management.
* **Alma:** Ex Libris Alma is a cloud based LMS designed for academic and research libraries. It offers comprehensive functionality for resource management, fulfilment, analytics, and more. Alma is known for its integration capabilities.
* **LibSys:** LibSys is a popular library management software used in various academic and research libraries. It provides modules for cataloging, acquisitions, circulation, serials management, and more.

**Functional Specifications**

## **2. System overview**

The Library Management System will be developed utilizing a client-server architecture, where the client (web application and mobile app) communicates with the server and the server hosts the database and business logic.



### **2.1. Scenarios**

**Modernizing a University Library with a Library Management System (LMS)**

* Imagine a university library with thousands of books and a diverse range of academic disciplines. Students and faculty often struggle to efficiently locate specific books or related materials due to the large volume of resources. Borrowing and returning books can be a time-consuming process, especially during peak academic periods.
* The university administration recognizes the need to enhance the library's efficiency and user experience. They decide to implement a Library Management System (LMS) to modernize their library operations.

**2.2. System Architecture**



### **2.3. Details of scenarios**

**Challenges:**

**Inefficient Book Management:** The current manual cataloging and inventory management make it difficult to track books effectively, resulting in frequent misplacements and inconsistencies in the catalogue.

**Time-Consuming Transactions:** The manual borrowing and returning process is time-consuming and inconvenient for both students and library staff, especially during busy academic periods.

**Limited Accessibility:** The physical cataloging system restricts access to the library's resources, especially for remote or off-campus users.

**Solution with LMS:**

The university implements a robust LMS, allowing students and faculty to access the library's resources seamlessly. The LMS addresses the challenges by offering the following benefits:

**1. Efficient Book Management:**

Books are digitally catalogued and categorized based on various criteria (title, author, subject, etc.), enabling quick and accurate searches.

**2. Streamlined Transactions:**

Borrowing and returning books are simplified through an online portal, reducing the time and effort required for these transactions.

**3.Enhanced Accessibility:**

The LMS is accessible remotely, enabling users to search for and reserve books from their homes or other locations. They can also receive notifications and reminders for due dates.

**Outcomes:**

**Improved User Experience:** Students and faculty can easily find and borrow books, improving their academic experience and productivity.

**Time and Resource Efficiency:** Library staff can focus more on assisting users and improving library services rather than managing manual transactions and book placements.

**Increased Resource Utilization:** With an efficient system, the library's resources are utilized optimally, benefiting a larger user base.

The LMS successfully modernizes the university library, meeting the needs of a technologically advanced academic community and enhancing overall operational efficiency and user satisfaction.

### **2.4. Open issues**

* **Mobile Application Development:** Determine whether the mobile application should be developed for iOS (Apple) and/or Android platforms.
* **Barcode Scanner Compatibility:** Support various barcode types commonly used in library materials, such as ISBN barcodes.
* **Integration Issues:** Interlibrary loan requests are not consistently syncing with partner libraries.
* **Multi-Lingual Support:** Implement language localization for the LMS user interface, allowing users to select their preferred language.
* **Programming Language Selection:** Determine if the programming language can easily integrate with databases and libraries that the LMS relies on.

### **2.5. Side notes**

After researching the existing library management system, we found that the public had offered suggestions for improving the system which includes the following: -

* **User Training and Support**: Develop comprehensive training materials, including user manuals, guides, video tutorials, and FAQs, to assist users in learning how to use the LMS.
* **Security and Privacy:** Ensure that data transmission between users and the LMS, as well as data at rest, is encrypted to protect against unauthorized access.
* **Disaster Recovery and Backup:** Implement a robust data backup strategy that includes regular, automated backups of the LMS database.

## **3. Functional Requirement**

## **3.1. Features**

* **Digital Membership**: while getting physical membership not only seems a tedious task but is also time consuming. This feature of digital membership will help the users by providing membership within minimal time, hence saving their precious time and energy.
* **Recommendation System**: When a reader reads a book of a particular author or a particular genre, our Library management system will recommend books to that reader based on his personal choice Instead of going with the choice of the mass.
* **Reviews:** After reading a book, the reader will be allowed to provide review for the same.
* **Ratings:** Besides giving an overview of the book, our Library management system will help users by providing the rating for each book which will help the new readers
* **Reminders/Notifications**: This system will also provide a feature of frequent reminders for returning the book on time.
* **Penalty System**: If a reader fails to return the book on time, the system will start adding fine in that user's account which will keep updating automatically.
* **ISBN support:** ISBN support will help get the details of the book such as author, publisher, publication date, number of pages, etc.
* **Addition/Deletion of Users**: After a user has been provided with the membership to the library, he will be automatically added to this system and provided with necessary credentials. On the other hand, if a user does not want to renew his/her membership, they will be removed from this.
* **Addition/Deletion of Books**: Our system will help the users by providing the information of newly added of newly arrived books. In case of the books which are not being read for a long time or which are no longer in demand, the system will take care that it is removed from the list.
* **Pre-Booking:** If a reader wants to read some book which is currently unavailable, the user can pre- book that book. As soon as the book will be returned by the previous reader, this reader will get the opportunity to read it.

**External Interface Specification**

## **4. Interaction of software with other components**

**4.1How will the software interact with the users?**

* **Email and Notifications**: The library system will be responsible to send regular emails regarding the deadline to return a book, about the fine that the user is supposed to pay, about the newly arrived collection of books, etc.
* **Authentication Services**: The system will work with the database to tally the credentials and hence provide security to users while preventing invalid logins.
* **Dashboard**: The users will have a personalized dashboard according to their roles which will assist them in doing their respective task such as issue a book, search for a book etc.
* **Catalogue Search**: The user can easily search through the catalogue to find a particular book as per their requirement. Moreover, the user can use the feature of search filter, sorting etc, to do the same.
* **Pre-booking or reservation system**: The system will also provide the user with the feature of placing reservation for the book that he/she wants but is currently unavailable.
* **Customer Suppor**t: The system will assist the users by providing them support in case of any confusion or help required.

**4.2 How will the software interact with other software?**

* **Email**: Our software i.e., library management system will be working along with email to send notifications and remainders.
* **WhatsApp**: Since there may be users who do not check their email regularly, our system will also deal with sending notifications and reminders through WhatsApp.
* **API**: APIs will be used to integrate the dashboard of the library management system with the database for performing actions like searching the catalogue, checking out books, or managing user accounts (e.g., RESTful APIs)
* **Database Integration**: The system will be integrating with the database management system to store the information of the users, book details, stock details etc. Software will interact with the database using SQL or other relational frameworks like Hibernate or Django ORM.
* **Web Services**: The system will be a web-based interface which can be accessed through web browsers. Therefore, the management system will be interacting with the backend services to provide the required features like that of signing-in or searching the catalogue.

**Technical Specification**

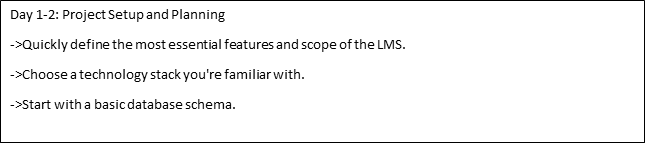
## **5. Deliverables**

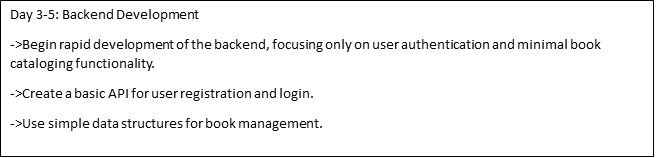
These deliverables help in documenting and communicating the progress, outcomes, and results of the project.

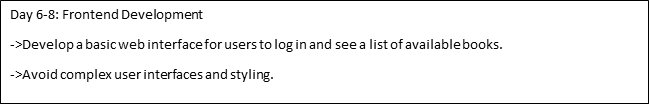
**5.1. System constraints**

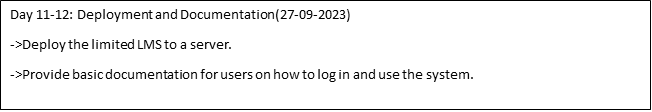
* The solution ought to work with popular web browsers including Chrome, Firefox, and Safari.
* Real-time updates and notifications require an internet connection.

**5.2. Timeline**









**6. Quality Attribute**

### **6.1. Performance Expectation**

The main goal of creating a library management system is to make libraries work better and easier. It helps people find and borrow books more easily. It also makes it simpler for librarians to organize books and take care of library tasks. Ultimately, it's about making the library experience great for everyone: -

* **Response Time**:

The system should respond to user requests within a reasonable timeframe, typically within a few hundred milliseconds, to ensure a smooth and responsive user experience.

* **Database Performance**:

Database queries should be optimized for efficiency, ensuring fast retrieval and updating of data to support the application's functionality.

* **Search Performance**:

Implement efficient and fast search capabilities within the library's collection, ensuring users can quickly find books based on various search criteria.

### **6.2. Security**

* User information must be encrypted and maintained securely.
* Strong user authentication and access control are required.
* regular updates and examinations of security.

### **6.3. Usability**

* The user interface should be intuitive and user-friendly, minimizing the learning curve for users.
* The system should be accessible to people with disabilities following appropriate accessibility guidelines.

### **6.4. Reliability**

* With little downtime, the system should be accessible around-the-clock.
* recurring backups and strategies for catastrophe recovery.

### **6.5. User Interface**

### **Web Application**

* Landing page
* Login and registration screens
* Cataloguing interface
* Member management interface
* Borrowing and returning interface
* Report generation interface

### 