**1.Problem Statement**

To elucidate the intricate challenges associated with library management, particularly for smaller institutions, I present a detailed problem statement divided into several sections:

Manually inputting bibliographic details is time-consuming and prone to errors. A lack of standardization leads to inconsistent records across various platforms, making it difficult to locate materials effectively.

Keeping track of checked-out books, calculating fines, and monitoring due dates are laborious tasks when done manually. An automated system would streamline these processes, reducing human error and improving efficiency

Smaller libraries often struggle to maintain up-to-date inventories and offer digital content. A robust LMS could bridge this gap, enabling remote access to electronic resources and fostering a more inclusive community

Tracking reader habits, recommending relevant literature, and identifying areas for improvement require advanced analytics. Currently, many libraries lack the necessary tools to gather meaningful insights from their patrons

Without accurate metrics, librarians cannot assess the effectiveness of their programs or identify trends in usage patterns. A reliable LMS should enable administrators to generate customizable reports and visualizations.

Limited budgets force librarians to prioritize certain aspects of their work at the expense of others. A flexible LMS allows institutions to allocate funds strategically, optimizing their investments and maximizing returns.

**ANALYSIS**

2.REQUIREMENTS SPECIFICATION:

The following are the functional and non-functional requirements for the system:

| **Functional Requirements** | **Description** |
| --- | --- |
| Book Cataloging | Allow librarians to add, edit, and delete books with details such as title, author, genre, ISBN, and availability status. |
| User Management | Enable librarians to create and manage user accounts, track borrowing history, and set user privileges. |
| Borrowing and Returning | Facilitate the process of checking out and returning books, including due date reminders and fine calculation. |
| Search Functionality | Provide users with a search feature to find books by title, author, genre, or keyword. |
| Reporting System | Generate reports on book circulation, popular titles, overdue items, and user activity for analysis. |
| Notifications | Send automated notifications for overdue books, reservation pickups, and system updates to users and librarians. |
| **Non-Functional Requirements** | **Description** |
| Performance | The system should respond quickly to user actions, handle concurrent requests efficiently, and support a large number of users simultaneously. |
| Security | Implement robust authentication mechanisms, data encryption, access control policies, and regular security audits to protect sensitive information. |
| Usability | Design an intuitive user interface with clear navigation, consistent layout, accessibility features, and responsive design for various devices. |
| Reliability | Ensure high system availability with minimal downtime, data backup mechanisms, error handling procedures, and disaster recovery plans in place. |
| Scalability | Design the system to scale easily as the library collection grows or user base expands without compromising performance or functionality. |
| Compatibility | Ensure compatibility with different operating systems, web browsers, devices, and network configurations to maximize accessibility for users. |

These requirements serve as a foundation for developing a robust Library Management System that meets the needs of both librarians and library patrons effectively.

**USECASE VIEW**

**3. IDENTIFICATION OF ACTORS**

* Actors are NOT part of the system.
* Actors represent anyone or anything that interacts with (input to or receive output from) the system.
* An actor is someone or something that:
  + Interacts with or uses the system.
  + Provides input to and receives information from the system
  + Is external to the system and has no control over the use cases.
* Actors are discovered by examining:
  + Who directly uses the system?
  + Who is responsible for maintaining the system?
  + External hardware used by the system.
  + Other systems that need to interact with the system.
* The needs of the actor are used to develop use cases. This insures that the system will be what the user expected.

Graphical Depiction

* An actor is a stereotype of a class and is depicted as a "stickman" on a use-case diagram.

reservation

clerk

Naming: The name of the actor is displayed below the icon.

Questions that help to identify actors

1. Who is interested in a certain requirement?
2. Where is the system used within the organization?
3. Who will benefit from the use of the system?
4. Who will supply the system with information, use this information, and remove this information?
5. Who will support and maintain the system?
6. Does the system use an external resource?
7. Does one person play several different roles?
8. Do several people play the same role?
9. Does the system interact with a legacy system?

Using the above questions we have identified two actors in Library Management System. They are

1 .LIBRARIAN

2. USER

**LIBRARIAN:** Librarian is a person who is responsible le for issuing books , managing users , updating the number of books and **Circulation Management**.

Librarian

**USER**: The person who takes/returns book from library.

User

**4. IDENTIFICATION OF USE-CASES AND SUB USE-CASES**

Use case is a sequence of transactions performed by a system that yields a measurable result of values for a particular actor. The use cases are all the ways the system may be used.

Graphical Depiction:

* The basic shape of a use case is an ellipse:



Naming

* A use case may have a name, although it is typically not a simple name. It is often written as an informal text description of the actors and the sequences of events between objects. Use case names often start with a verb.
* The name of the use case is displayed below the icon.

booking

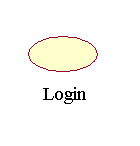
Questions that help to find use cases

1. What are the tasks of each actor?
2. Will any actor create, store, change, remove or read information in the system?
3. What use cases will create, store, change, remove, or read this information?
4. Will any actor need to inform the system about sudden, external changes?
5. Does any actor need to be informed about certain occurrences in the system?
6. What use cases will support or maintain the system?
7. Can all functional requirements be performed by the use cases?

By applying above questions to bus reservation system application the following use cases are identified .They are

**1)Login**: This usecase provides the user to login into the library.

UML notation:



2)**Borrow Book:** This use case provides what book the user borrowed.

UML notation:

Borrow Book

3**) Return the Book**: This use case provides returning a book by user.

UML notation:

Return Book

4) **Manage Books:** This use case is used by Librarian .It contains adding, updating, removing, searching

Of the books

UML notation:

Manage Books

5) **Search Books:** This use case is used by both librarian and users to search for desired books

UML notation:

Search Book

**5.Flow of Events for Library Management System:**

| **Use Case** | **Flow of Events** |
| --- | --- |
| **Manage Student Records** | 1. Admin logs into the system. 2. Admin selects the "Manage Student Records" option. 3. Admin adds a new student record with details. 4. System saves the student record in the database. |
| **Manage Books** | 1. Librarian logs into the system. 2. Librarian navigates to the "Manage Books" section. 3. Librarian adds a new book to the catalog with details. 4. System updates the book database with the new entry. |
| **Manage Issues** | 1. User requests to borrow a book. 2. Librarian checks book availability and issues it to the user. 3. System updates the book status as borrowed and sets a due date. |
| **Manage Librarian** | 1. Admin accesses the "Manage Librarian" module. 2. Admin adds a new librarian profile with necessary information. 3. System stores the librarian details in the database for future reference. |
| **Manage Member** | 1. User logs into their account. 2. User updates their contact information in the profile section. 3. System saves the updated user details for future interactions. |
| **Full Library Management System Operations** | 1. User/Admin accesses the main menu of the system after login. 2. User/Admin selects various functions like searching books, issuing/returning books, etc., from the menu options provided by the system. |
| **System Administration** | 1. Admin logs into the system with administrative credentials. 2.Admin navigates to system settings. 3.Admin makes necessary changes to system settings. 4.System saves changes made by admin. |

These flow of events outline how different users interact with various modules of the Library Management System, performing tasks such as managing student records, books, issues, librarians, and system administration