# ****Library Management System****

# ****Interim Report****

**Bachelor of Information Technology**

**University Of Moratuwa**



### E2245624

### Wijesooriya GSK

## ****Chapter 1: Introduction****

### ****1.1 Project Overview****

The Library Management System (LMS) is designed to manage the functionalities related to a library's operations, including borrowing and returning books, managing user memberships, tracking inventory, and generating reports. This system helps streamline administrative tasks and enhance the user experience by automating manual processes, allowing librarians and users to access information quickly and efficiently.

### ****1.2 Objectives****

1. To automate the library's day-to-day operations.
2. To enable easy book borrowing and returning processes.
3. To keep an up-to-date record of available and borrowed books.
4. To manage users’ memberships, including student and staff registrations.
5. To generate various reports related to book usage, inventory, and user activity.

## ****Chapter 2: Similar Systems****

### ****2.1 Comparative Analysis****

Before designing the system, several existing library management systems were analyzed to understand their strengths and weaknesses:

**1.Koha**  
Koha is an open-source integrated library system. It offers features such as book borrowing, cataloging, and reporting, but it requires heavy technical knowledge for setup and maintenance.

**2.Evergreen**  
Evergreen is a highly scalable library system. It handles multi-branch libraries but might be too complex for small-to-medium institutions.

**3.SLiMS (Senayan Library Management System)**  
SLiMS is a simple-to-use system, but it lacks customizable reports and advanced user management functionalities.

### ****2.2 Lessons Learned****

From the analysis, the following were identified as key features and limitations:

1. **Key Features to Implement**: Book tracking, membership management, reporting features, and an intuitive user interface.
2. **Key Challenges to Avoid**: Excessive complexity in user management, difficult setup processes, and lack of mobile support.

## ****Chapter 3: Solution****

### ****3.1 Functional Requirements****

**3.1.1 User Management**

1. Register and manage different user types (e.g., students, staff).
2. Allow users to view available books and check borrowing history.

**3.1.2 Book Inventory Management**

1. Add, update, and remove books from the library system.
2. Track book availability, categories, and location in the library.

**3.1.3 Borrow and Return Books**

1. Enable users to borrow books and automatically update the inventory.
2. Provide due date reminders and penalty management for late returns.

**3.1.4 Search Functionality**

1. Allow users to search for books by title, author, ISBN, or category.

**3.1.5 Report Generation**

1. Generate reports on borrowed books, overdue books, inventory status, and user activity.

### ****3.2 Non-functional Requirements****

**3.2.1 Performance**  
The system should handle up to 100 concurrent users without noticeable slowdowns.

**3.2.2 Security**  
User data, especially passwords and borrowing histories, should be encrypted and stored securely.

**3.2.3 Usability**  
The interface should be user-friendly, even for non-technical users, with clear navigation and instructions.

**3.2.4 Scalability**  
The system should be able to handle growth in the number of books and users over time.

### ****3.3 Flowcharts****

Below are sample flowcharts representing key functionalities. Replace placeholders with your system’s actual flowcharts.

#### 3.3.1 User Registration Flowchart

**Start**

**Registered?**

**No**

**Desplay Registration Form**

**Yes**

**Desplay Login Form**

**END**

**3.3.1 Book Borrowing Flowchart**

**Start**

**Check The availability of book from Library database**

**Book is available?**

**No**

**Print Book is Not available**

**No**

**YES**

**YES**

**Need another book?**

**Display the return date**

**No**

**End**

**3.3.2 Book Returning Flowchart**

**Start**

**End**

**“Book is Recived “ massage print**

**Update Database**

**Enter the Book ID**