

# Database Requirements

## Introduction

### **Purpose**

We are developing a Library Management System to maintain a database for a local library, allowing members of the community to effectively and efficiently use library resources. By implementing this system, library workers can track important information on user activity, while also being able to adjust the rules to their needs (like changing membership status), generate reports, etc. In general, this system will provide ease to the community, the staff, and help make people's experience at the library more enjoyable.

### **Scope**

Our Library Management System efficiently tracks books, digital media, and magazines, maintaining key details like title, author, ISBN, publication year, genre, and availability. It also monitors guest desktop computers, recording user sessions and login/logout times. The system manages item availability, loan statuses, due dates, and check-out history while allowing users to search the catalog, reserve and renew items, and check loan statuses. Library staff can process checkouts, returns, add new items, and manage user accounts, which store information on checked-out items and holds. Additionally, the system will only generates reports on borrowing trends by genre.

# Stakeholders

The stakeholders of the Library Management System include individuals and groups who will either use or be affected by the system. These stakeholders play an important role in defining the requirements and making sure of its success.

1. **Library Staff:** Librarians and administrative personnel will use the system to manage books, digital media, magazines, and client records. Their responsibilities include checking out items, processing returns, enforcing borrowing policies, handling late fees, and maintaining the collection database.
2. **Library Members:** Clients who borrow items from the library, including students, faculty, researchers, and the general public. They will interact with the system to search for materials, reserve books, check due dates, and receive notifications about loans and overdue items.
3. **Library Management:** Decision-makers who oversee library operations, including directors and department heads
4. **IT:** Technical staff responsible for maintaining the LMS, ensuring data security, managing software updates, and troubleshooting system issues.
5. **Software Development Team:** The project team designing, developing, and maintaining the system. They are responsible for implementing the database schema, developing user interfaces, and ensuring the system meets the needs of all stakeholders.
6. **External Vendors & Publishers:** Entities that provide digital media subscriptions, book acquisitions, and other library resources.

# Requirements

## Functional Requirements:

The database will manage and organize different aspects of the library's operations. First, a user administration will be established to differentiate between roles and access requirements. For example, a library member should only have read permissions, while the library staff should have read and write permissions to update inventory. The database should maintain a comprehensive list of all books, digital media, and magazines, storing key details such as title, author or creator, ISBN, publication year, genre, and availability status. The system must also keep track of guest desktop computers, recording important information such as computer numbers, users of each computer, and login/logout times. The database must also monitor item availability, loan status, and due dates, providing timestamps for checkouts and details of the users who borrowed them.

Additionally, the system should enable users to reserve and renew items, check their loan statuses, and search the catalog for books and digital media. Library workers must have the ability to check out items, process returns, add new items to the database, and manage client accounts easily. User profiles should be maintained within the system, including records of checked-out items and any holds. Lastly, the database should generate reports on borrowing trends, categorized by genre, to help in library resource planning.

## Data Entities:

- Roles
  - Library Member (View permissions)
    - UserID
    - First Name
    - Last Name
    - Contact Information(Composite, depending what information we will receive we will break this down accordingly,
    - MembershipStatus
    - ItemsCheckedOut
  - Library Staff (View/Edit permissions)

- StaffID
  - FirstName
  - LastName
  - Contact Information(Composite, depending what information we will receive we will break this down accordingly,
- SysAd (View/Edit permissions)
  - StaffID
  - First Name
  - Last Name
  - Contact Information(Composite, depending what information we will receive we will break this down accordingly,
- Books
  - ItemID
  - ISBN
  - Title
  - Author
  - Publication Year
  - Genre
  - AvailabilityStatus
- Digital Media
  - ItemID
  - ISBN
  - Title
  - Author
  - PublicationYear
  - Genre
  - AvailabilityStatus
- Transactions
  - CheckedOutID
  - UserID
  - ItemID
  - Status
  - DateCheckedOut
- Report
  - ReportID
  - BorrowingTrends
  - NumberofCheckouts

## Non-Functional Requirements:

Below are key non-functional requirements that will ensure the Library Management System is secure, scalable and reliable.

### 1. Security

- The system will implement role-based access control (RBAC) to restrict or grant access to different functionalities:
  - Library Administrators can manage user accounts, add/remove books, update rules, and generate reports.
  - Library Staff can check out and return books, modify loan statuses, and manage reservations.
  - Library Members can search the catalog, reserve, renew, and check out items.
- The database will enforce encryption for sensitive data, including user personal details and login credentials.
- Authentication should be managed through multi-factor authentication for administrative users
- Logs should track all data access and modifications, providing an audit trail for accountability.

### 2. Performance and Scalability

- The system will handle simultaneous user requests efficiently without delays, and support at most 50 concurrent users.
- The database will support indexing and query optimization to ensure fast searches and retrieval of book records

### 3. Reliability

- The database will ensure 99.9% uptime, with scheduled maintenance occurring during off-peak hours.
- The system will have an automatic failover mechanism and regular backups to prevent data loss in case of hardware failures or cyberattacks.
- All transactions (books, checkouts, returns, reservations) must follow ACID (Atomicity, Consistency, Isolation, Durability) principles to maintain data integrity.

# Hardware and Software Requirements

## Hardware and Additional Software:

Regardless of the chosen database system, the hardware requirements include a system with at least 8GB of RAM, a multi-core processor (Intel i5 or equivalent), and at least 10GB of free disk space to accommodate the database and necessary dependencies. Additional software includes a terminal or command-line interface (CLI) for executing SQL operations and an SSH client (like PuTTY or a Unix-based terminal) for securely accessing remote servers. Ensuring these requirements are met will provide a stable and efficient environment for database management and query execution.

## POSTGRES

### Pros:

- **Advanced Features:** PostgreSQL supports complex queries, large datasets, and follows strict SQL standards, which could be useful for handling more sophisticated database operations
- **Highly Extensible:** It allows custom functions, data types, and operators, which could be helpful if the project needs more flexibility in how data is stored or processed
- **Strong Community Support:** Since it's open source and widely used there's plenty of documentation and support available

### Cons:

- **Resource Heavy :**PostgreSQL tends to require more memory and processing power, which could be a problem given the hardware limitations for this project.
- **More Complex :**It has a lot of powerful features, but that also makes it more complicated to set up and use compared to MySQL or MariaDB.

For this project, PostgreSQL is the best choice because of its advanced features strict SQL compliance and extensibility, which will allow for more flexibility in handling data. While it is more resource intensive its ability to manage complex queries and large datasets outweighs the hardware constraints. If any performance issues arise, MariaDB will serve as a backup option due to its lightweight nature and ease of setup.

## **MariaDB (EECS Servers): Backup**

MariaDB is a powerful, open-source relational database that will be hosted on the EECS servers. This option provides:

- Remote accessibility: The database can be accessed from anywhere with an internet connection.
- Automatic backups: EECS servers often include backup mechanisms, reducing the risk of data lost
- Scalability: Suitable for handling larger datasets without overloading a personal machine.
- Consistency: Ensures all team members are working in the same environment, reducing compatibility issues.