# Requirements Document – Library Management System

## Part 1: Introduction

Project Overview: The purpose of this database is to support a small library in managing its collection of books, magazines, and digital media. It will streamline operations such as cataloging items, tracking borrowing and returning transactions, and calculating late fees. The system will also help staff generate reports on library usage, member activity, and item availability. Its intended use is to replace manual processes with an efficient and reliable database system that improves accuracy, reduces administrative workload, and improves the borrowing experience for library members.  
  
Scope:  
The boundaries of the project include storing and managing data about library items, members, staff, and borrowing transactions. The system will enforce borrowing limits based on membership type, handle overdue fines, allow reservations, and provide notifications. It will also support the generation of analytical reports for both operations and decision-making.

The system will not cover multiple libraries/ mobile app development.

Glossary:  
- ISBN: International Standard Book Number

- Transaction: A record of borrowing or returning an item.

- Entity: An object or concept in the database.

- Attribute: A property or characteristic of an entity.

- Constraint: A rule enforced in the database to ensure data integrity.

-Primary Key: A unique identifier for each record in a table.

-Foreign Key: An attribute in one table that links to a primary key in another table.

## Part 2: Stakeholders

The main stakeholders of this system are:  
- End Users (Library Members)  
- Library Staff (Administrators)  
- System Maintainers (DB Admins)

Their expected interactions with the system include:  
- Responsible for the day-to-day operation, performance, and security of the database.  
- Backup, recovery, updates, and maintaining integrity constraints.

## Part 3: Requirements Functional Requirements:

The system must provide the following functions:  
1. User administration

Includes adding new members, managing the permissions of existing members, and removing members.   
2. Data management (entry, retrieval, modification)

Includes querying attribute data of items and members, adding new data, and modifying/removing existing data.  
3. Reporting user data (overdue items, checked out items, and users with either)

Predefined data reports must be defined to provide easy interpretation to administrators, including lists of overdue items, items that are checked out, users with checked out items, and users with overdue items.

4. Item Management

Users should be able to reserve specific items in advance, as well as pay off fines.

### Data Entities:

The main data entities and attributes are:

- Entity 1: Item Categories  
 Attributes: item id (specific to library), item type  
 Data Types: int, varchar (respectively)  
 Constraints: item id is a primary key, all key constraints hold

- Entity 2: Book  
 Attributes: IID, title, author, ISBN, publication year, genre, availability status, borrow limit  
 Data Types: int, varchar, varchar, int, varchar, varchar, bool, int (respectively)  
 Constraints: IID (item id is a primary key, all key constraints hold)  
  
- Entity 3: Magazine  
 Attributes: IID, title, issue number, publication date, availability status, borrow limit  
 Data Types: int, varchar, int, varchar, bool, int (respectively)  
 Constraints: IID (item id is a primary key, all key constraints hold)  
  
- Entity 4: Client  
 Attributes: ID, name, email, membership type, and account status.  
 Data Types: int, varchar, varchar, varchar, and bool (respectively)  
 Constraints: ID is a primary key, all key constraints hold; membership type must exist in the MembershipType relation  
  
- Entity 5: MembershipType  
 Attributes: type, borrowing limit, late fee rate  
 Data Types: varchar, int, int (respectively)  
 Constraints: type is a primary key, all key constraints hold

- Entity 6: Transaction  
 Attributes: transaction id, client id, IID, borrow date, due date, days overdue, returned status  
 Data Types: int, int, int, varchar, varchar, int, bool (respectively)  
 Constraints: transaction id is the primary key, all key constraints hold; client id must exist in the Client relation; IID must exist in the Item Categories relation

- Entity 7: Reservation  
 Attributes: client id, IID, reservation date, transaction id  
 Data Types: int, int, varchar, int (respectively)  
 Constraints: the primary key for this relation is transaction id, and this relation is inherently dependent on its existence as a tuple in the Transaction relation where Reservation.reservation date=Transaction.borrow date; client id must exist in the Client relation; IID must exist in the Item Categories relation

## Part 4: Hardware and Software Requirements

The database system will run on:  
- **Database**: Maria DB 10.6 LTS (\*Consult with Professor Hossein for accessibility\*), deploy personal logins for each member to access the database via on/off campus. Furthermore, the defined chatset is utf8mb4, avoids truncation issues and establishes standard entry of characters.

- **Operating System**: EECS-managed Linux; Our database will run on the EECS department’s Linux servers. These servers are centrally managed, which means updates, security patches, and backups are handled by EECS IT staff. We will not have administrator access to the OS, but we can fully manage our own schema and data inside MariaDB.

- **Development Tools**: We will use MySQL Workbench as a SQL client to connect to the MariaDB server, run queries, and design schemas. Command-line clients will be run through the terminal to make further iterations on the database.

- **Minimum Hardware**:

Processor (CPU): 2 virtual cores (vCPU)  
 Enough to handle a multitude of students running queries at once.

Memory (RAM): 4 GB  
 Prevents slowdowns when users join.

Storage (Disk): 50 GB SSD  
More than optimal amount of storage for a medium size database.

Network: 1 Gbps campus LAN  
 Ensures quick response when you query from the lab, library, or over VPN.

## Part 5: GitHub Repository Management

The project repository will be managed as follows:  
- Repository Name: EECS 447 Semester Project  
- Branching Strategy: The main branch will contain stable ready code. A dev branch will serve as the integration branch for features. New features or fixes will be developed on separate branches named respectively feature and bugfix  
- File Organization:  
 - /docs → Project documentation (requirements, design, user specifications, user manuals)  
 - /sql → Database schema definitions, sample data  
 - /reports → Progress reports, final reports, and any presentation materials  
 - /tests → Unit tests, integration tests, and test data files  
- Pull Request / Merge Process: Contributions will be made through pull requests into the dev branch. Each pull request must undergo peer review and testing prior to approval. Stable, reviewed code will be merged from the dev branch into the main branch at designated milestones

## Part 6: Appendices (Optional)

## 

## Second Project Meeting Log

Date: September 27, 2025

Time: 8:00 PM – 9:00 PM

Location: Zoom Meeting

Objective: Review and assign tasks for the Requirements Document (Parts 1–5)

Team Members Present: Jack, Jamie, K, Aidan, Maxwell

Task Completion Confirmation:

K Li: Completed Part 1 (Introduction) of the Requirements Document.

Aidan Lowry: Completed Part 4 (Hardware and Software Requirement)

Jack Gerety: Completed Entities, Attributes, and Constraints on Part 3

Jamie King: Completed Functional Requirements on Part 3, reviewed completion of other parts

Brainstorming Session:

Discussed how to divide the Requirements Document into five parts.

Shared ideas for glossary terms, functional requirements, and data entities.

Talked about the simple system scope and what to leave out.

Agreed on individual responsibilities for each part.

Team responsibilities:

Each member is responsible for one part of the Requirements Document.

Ensure drafts are uploaded to GitHub before the next meeting.

Review each other’s sections for consistency and clarity.

Tasks Allocated:

- [Aidan Lowry]: Responsible for Part 4 of the Library requirements (Hardware and Software: Database, Operating system, Development Tools, and Minimum Hardware)

- [Jamie King]: Responsible for Part 3, outlining the Functional Requirements, as well as reviewing other parts of assignment

- [K Li ]: Responsible for Part 1 of the Library Requirements (Introduction: Project Overview, Scope, and Glossary).

- [Jack Gerety]: Entities and their attributes and constraints on Part 3

- [Maxwell Phachanla]: Responsible for part 5 of Library Requirements, created dev, feature, and bugfix branches for future work, and uploaded the completed file to GitHub.

Follow-Up Actions:  
 Next meeting scheduled for **10/4/25** at **8:00PM**