Library Management System - Project Part 2: Library Database Requirements

EECS447: Database Systems

Team Name: ASYNCLIB
Team Members: Nick, Ashton, Cole, Sean, Yadhunath

Professor: Hossein Saiedian
Semester: Fall 2025

Revisions

Version: 1.4

Version	Date	Author	Description
1.4	09/23/25	Nick, Ashton, Cole, Sean, Yadhunath	Made the initial draft, assigned roles, created the document template.
1.5	09/26/25	Nick, Ashton, Cole, Sean, Yadhunath	Worked on finalizing the requirements and performed final clean-up for submission.

Project Overview

This Library Management System (LMS) project aims to efficiently manage extensive book information, memberships, clientele, profiles, and accounts. Its database consolidates a vast amount of data, presenting it through an intuitive and easily navigable interface for user convenience.

Scope

This project focuses on developing a comprehensive resource management system for books and magazines. It will store key details such as title, author, ISBN/issue number, publication date, genre, and availability.

Additionally, the system will feature a client and membership management component. This will include unique IDs, contact information, various membership categories (regular, student, senior), and defined borrowing restrictions.

A tracking system will be implemented to manage borrowed, returned, and reserved media. This system will record timestamps, client details, and enforce rules regarding borrowing limits and late return fees.

Finally, a user interface will be developed to enable staff to efficiently manage operations. This includes checking out and returning items, adding new resources, managing client accounts, calculating fines, and viewing both resource availability and client profiles.

Glossary

- **PostgreSQL**: Adds objects-oriented features to sql, such as inheritance and custom data types along with additional features
- **Flask:** Python based web framework
- **React:** JavaScript library for building user interfaces
- **Pyscopg2**: A python library used for a seamless integration of PostgreSQL database to a python project.

Stakeholders

Librarians and Library Associates (Library Staff): These individuals are

responsible for the day-to-day operations of the library. Their duties include managing physical and digital collections, assisting patrons with inquiries, processing new acquisitions, maintaining the library's catalog, and ensuring the smooth functioning of lending and return processes. They also play a crucial role in maintaining a welcoming and organized environment for all library users.

Clients (Members): This category encompasses all individuals who utilize the library's resources and services. Members are further categorized into:

- **Regular Members**: Standard patrons who have full access to library services based on general membership guidelines.
- **Student Members**: Patrons who are actively enrolled in an educational institution, potentially qualifying for specific discounts, extended loan periods, or access to academic resources.
- **Senior Members**: Patrons who meet a certain age criterion, often receiving benefits such as reduced fees or specialized programming.

IT Staff and Administrators: This group is vital for the technical infrastructure and overall management of the ASYNCLIB system. They are responsible for maintaining the library's computer systems, network, and database. Their tasks include troubleshooting technical issues, implementing security measures, managing user accounts, performing system upgrades, and ensuring data integrity and accessibility.

Developers (ASYNCLIB): This refers to the team responsible for designing, building, and maintaining the ASYNCLIB software itself. Their work involves writing code, implementing new features, fixing bugs, conducting testing, and ensuring the system meets the functional and performance requirements of all user groups. They are the creators and continuous improvers of the platform.

Professor and TA (Saiedian and Sophia): These individuals represent the academic oversight and instructional component related to the ASYNCLIB project.

- Professor Saiedian will provide us overall guidance, verify project requirements, and evaluate the work of the developers.
- Sophia, as the TA, would assist the professor in these tasks, provide more direct support to the development team, and facilitate learning and project progress.

Their role is primarily supervisory and educational, ensuring the project aligns with academic objectives and standards.

Functional Requirements

These are functions that will be included in the database at minimum

User Management

- Data user information: Unique profile IDs, membership level, contact information, name, age.
- Membership types(Age based): Junior, regular, student, senior.
- Borrowing limits correspond with membership types.

Storage information

- Data: book ISBN, issue number, publication date, genre, availability.
- Magazines: unique ID, number, publication date, genre, availability.
- Digital Media: Unique ID, publication date, director, genre, availability.

Interface functionality

- User borrowing and returning processes.
- Paying late fees.
- Track date of return, late fees, check out date, and renewal dates.
- Borrowing history tracking.
- Automate calculating fees and fee reports.

Access

- Library admins have admin access.
- Developers have admin and full access.
- Clients have restricted access.

Non-Functional Requirements:

- ***Scalability:*** The database system must be highly scalable to accommodate a growing volume of book information, client data, and transaction records, ensuring efficient performance even during peak usage periods.

- ***Data Abstraction and Views*:** The system must support data abstraction, allowing different stakeholders (e.g., librarians, clients, IT staff) to access and interact with relevant data through customized, simplified views without needing to understand the underlying data structure.

- ***Security and Data Integrity:*** The system must implement robust security measures to prevent unauthorized access and data leakage. It must also ensure data reliability through validation checks during data entry and consistent data integrity across all operations.

- ***Availability***: High availability is crucial, ensuring minimal downtime and continuous access to library resources.

- ***Performance***: The system needs quick response times for all operations, providing a smooth experience for staff and clients.

- ***Maintainability***: Well-documented, modular design and code will simplify future maintenance, troubleshooting, and updates.

Database Requirements

Hardware Requirements

EECS Server Environment (Primary Deployment)

- **Processor**: Multi-core x86_64 architecture
- **Memory**: Minimum 2GB RAM allocated for database operations
- **Storage**: Minimum 10GB available disk space for:
  - Database files (~500MB estimated for project data)
  - Transaction logs and backups
  - System overhead
- **Network**: Stable network connection for remote access

Local Development Environment (Alternative)

- **Processor**:
  - Intel Core i3 or AMD equivalent (minimum)

- ➢ Intel Core i5 or better (recommended)

- ❖ **Memory**:
  - ➢ 4GB RAM (minimum)
  - ➢ 8GB RAM (recommended for optimal performance)
- ❖ **Storage**:
  - ➢ 50GB available disk space (minimum)
  - ➢ SSD recommended for better I/O performance
- ❖ **Network**: Internet connection for software downloads and GitHub access

Software Requirements

Backend

- ❖ **PostgreSQL:** A powerful open-source relational database system.
- ❖ **Flask (Python):** A micro web framework for Python.
- ❖ **Pyscopg2**: Facilitates interaction with the database to execute commands.

Frontend

- ❖ **React (Javascript)**: For a simple UI for the users to interact with the backend and the database.

Data Entities:

* - indicates that an attribute can be NULL

ITEMS (Every book/movie/magazine in the database)

item_id	type	title	publication_date	publisher	availability	held_by*
INT	VARCHAR / INT	VARCHAR(128)	DATETIME	VARCHAR(32)	BOOL	INT
Primary key	Whether it is a book or magazine	Title of work	Can be a full date and time or just a year	-	Is it in stock?	*Foreign key referencing clid - ON DELETE SET NULL*

BOOKS

item_id	authorfirst*	authorlast	ISBN	genre
INT	VARCHAR(32)	VARCHAR(32)	INT	String array
Foreign key ref. item_id	-	-	-	-

MEDIA

item_id	directorfirst	directorlast	genre
INT	VARCHAR(32)	VARCHAR(32)	String array
Foreign key ref. item_id	-	-	-

MAGAZINES

item_id	issue_number
INT	INT
Foreign key ref. item_id	

CLIENTS

clid	firstname	lastname	phone	email	member_type	account_status
INT	VARCHAR(32)	VARCHAR(32)	INT?	VARCHAR(64)	INT	VARCHAR(8)
Primary key	-	-	-	-	*Foreign key ref. member_type_id - ON DELETE SET NULL*	-

MEBERSHIPS

member_type_id	member_type_name	borrowing_limit	late_fee_rate	checkout_length
INT	VARCHAR(16)	INT	FLOAT	INT
Primary key	-	-	Daily rate of late fee	Days before late

RESERVATIONS

item_id	reserved_by	date_of_reservation
INT	INT	DATETIME
Foreign key ref. item_id	*Foreign key ref. clid*	

[Database spreadsheet]
