EECS 447 Project

Database Requirements

Version 1.2

Revision History

Date	Version	Description	Author
2/8/2025	1.1	All team members completed the first draft for the instructions of this document. All subsections have been completed and a thorough brainstorm of the requirements were also implemented.	Sophia Jacob, Kusuma Murthy, Anna Lin, Nimra Syed, Ella Nguyen, Nikka Vuong
2/22/2025	1.2	All team members reviewed and refined the second and final draft for this document. Sophia and Anna checked the document with the GTA.	Sophia Jacob, Kusuma Murthy, Anna Lin, Nimra Syed, Ella Nguyen, Nikka Vuong

Table of Contents

1 Introduction

1.1 Project Overview

As a Software as a Service (SaaS) company, ASKNrEceive (ASKNE) hopes to help streamline database creation and management. The purpose of this project is to provide local libraries with a Library Management System (LMS) that ensures efficiency for both types of end-users: library staff and library customers. This abstracted database will facilitate better documentation of the books, magazines, and other available items for ease of management and organization.

Through this project, ASKNE will develop a fully functional and scalable database software for libraries to handle, maintain, and analyze the popularity of their collections through detailed reports. Using a streamlined approach, the goal is to design a robust relational database with a well-defined conceptual schema and physical implementation.

**** Note: Refer to *01 - Project Plan/Vision* for any additional information on the project overview and definition of the project.**

1.2 Scope

ASKNE will use MySQL and other database management tools to deliver an extensive database backed in software to provide better insights and organization to libraries regarding books, magazines, and other content they house. This project entails creating a full-scale LMS, powered by a relational database. ASKNE aims to create a reporting style that helps the library determine popular books, members' favorite genres, and more quantifiable statistics to help maintain a growing library. This project will be responsible for managing the library's database operations such as adding new books as parts of entries, cleaning the database by removing and editing the borrow and return status of loanable items, and finally creating comprehensive reports for the Library admins and staff. Other features include tracking books and members based on a variety of attributes like author names, item IDs, publication dates, and more. Users can view the database for available books, while library staff can edit and maintain the database. As for the technical aspect of this project, ASKNE will be utilizing Structured Query Language (SQL) as the main tool used for searching the database and will create various tables that will be structured in a Relational database format.

**** Note: Refer to *01 - Project Plan/Vision* for any additional information on the scope and definition of the project.**

1.3 Glossary

**** Note: Our team has a dedicated Document for Abbreviations and Glossary. Refer to *07 - Glossary*.**

For the purpose of this document, it will also be put in this subsection.

- ASKNE - ASKNrEceive
- LMS - Library Management System
- SaaS - Software as a Service
- SQL - Structured Query Language
- TAs - Teaching Assistants

2 Stakeholders

For this database, the primary stakeholders include library staff and library customers. Library staff will serve as the administrators of this database, managing library transactions such as checking in and out of books, as well as accessing book inventory and status. The end users of this database are library customers – community members – ranging from youth to elders, who may want to check out books for educational or recreational purposes. Given the diverse use base, the database should account for the needs and requirements of all individuals.

Additionally, for the context of this project, our stakeholders will also include the Professor and the Teaching Assistants (TAs).

3 Requirements

3.1 Functional Requirements

The Library Database will support various functions related to book management, membership tracking, and reporting. Below will specify some of the SQL-based queries that could be used on our database. Following the three-schema architecture, the user will be able to have a high-level external view of certain data depending on their role and membership type.

- *Book Management & Availability*
    - Search items through identifying information such as the specific author, publication year, or title.
    - Retrieve book availability for customers to check out and for staff to keep track of items.

    - Create due dates for all items each time it is loaned.
    - Update the availability status of items and maintain their history.
- *Membership & Borrowing Tracking:*
    - View membership status and track borrowing history.
    - Identify frequent borrows of specific genres.
    - Track members with overdue books and those who never returned items late.
    - List clients who have exceeded borrowing limits.
    - Update all transactions of members including timestamps and responsible clients based on their overdue tally.
    - Create waitlist requests to reserve items on loan.
- *Financial & Fine Management:*
    - Track and calculate fines for each member based on overdue items and member type.
    - Generate monthly fee reports based on each item broken down by membership.
    - Create a comprehensive revenue summary for the annual fiscal year.
- *Reporting & Analytics:*
    - Generate due dates for all borrowed books.
    - Calculate average borrowing time and loan duration.
    - Curate a monthly summary report for an individual member on items borrowed.
    - Manage a statistical breakdown of client type and item category.
- *Engagement & Recommendations:*
    - Generate a list of the most popular authors based on the amount borrowed and genre.
    - Create a report on the most frequently borrowed items by client type.
    - Maintain a list of staff book recommendations.
    - Generate a report of top readers, ranking them by books read at the end of the year, as well as, a report of best-rated books of the year.

3.2 Non-Functional Requirements

The Library Database must meet various non-functional requirements to ensure performance, security, reliability, and maintainability. These requirements define constraints and quality attributes that the system must adhere to.

- *Basic Queries:* The system should process basic queries, such as searching for a book, in a reasonable amount of time (a few seconds) under normal load conditions.

- *User Traffic:* The system should be able to handle a decent amount of users without significant performance degradation.
- *Transactions:* Transactions, such as returning or borrowing a book, should be completed within a few seconds to maintain system responsiveness and should be atomic.
- *Indexed Searches:* Indexed searches should be implemented to optimize general lookup times for books, members, and borrowed history.
- *Data Validation:* Data validation rules should be enforced to prevent invalid entries (such as missing required fields).
- *Authorized Usage:* Only authorized users like library staff and administrators should have access to administrative functions like fine adjustments and membership modifications.
- *Privacy:* The system should not reveal client information to other members.
- *Secure Borrowing System:* The system should be able to continue if a transaction fails.
- *Back-Ups:* The database should support regularly automated backups to prevent data loss in the case of accidental deletions or system failures.

3.3 Data Entities

The Library Database will consist of several key entities and each of these entities will be defined by attributes and relationships.

- *Key Library Entities and Attributes*
  - **Library Item:** Item ID, Inventory
    - Specialization:
      - **Book**: Title, Author/Creator, ISBN (International Standard Book Number), Publication Year, Genre, Availability Status, Book Rating
      - **Digital Media Item**: Title, Author/Creator, ISBN (International Standard Book Number), Publication Year, Genre, Availability Status
      - **Magazines**: Title, Issue Number, ISSN (International Standard Serial Number), Publication Date, Genre, Availability Status
- *Key Membership Type Entities and Attributes*
  - **Client**: Name, Contact Information, Membership Type, Account Status, Fine Rate.
    - Specialization
      - **Staff**
      - **Library Members**: Member ID, Membership Type, Fee Type, Limit
        - Specialization

        - **Regular**
        - **Students**
        - **Senior Citizen**
- *Key Account Entities and Attributes*
    - **Library Account:** Date, Total Amount Paid Per Day, Total Amount Owed Per Day
    - **Member Account:** History (Multi-Valued), Incurred Fees, Total Amount Paid
- *Other Key Relations for Entities*
    - **Help**
        - *Related Entities:* Library Member → Staff
    - **Borrow:** Client ID, Item Name, Due Date, Status, Limit, Checked-Out, Amount Paid, Amount Owed
        - *Related Entities:* Library Items → Regular Members, Students, Senior Citizens
    - **Recommend Books**
        - *Related Entities:* Member Account → Client
    - **Charge**
        - *Related Entities:* Library Member → Library Account
    - **Can Be**
        - *Related Entities:* Library Member → Staff
    - **View**
        - *Related Entities:* Library Member → Member Account
    - **Has**
        - *Related Entities:* Client → Member Account
    - **Paid:** Amount Owed, Amount Paid, Member ID
        - *Related Entities:* Staff → Library Account
    - **Rating:** Stars given
        - *Related Entities:* Library Member → Book
- *Entity Constraints*
    - Clients are allowed to borrow a maximum number of items depending on their membership type.
    - There will be fees incurred for late returns.

  - Each membership type entails different rates for fees (eg. $0.30 for regulars, $0.15 for senior citizens, $0.20 for students).
  - Rare and limited edition books have borrowing restrictions.

4 Hardware and Software Requirements

This section outlines the hardware and software requirements necessary for the implementation and deployment of the Library Management System database. This system will be hosted either on EECS servers (MariaDB) or on local machines using MySQL, ensuring flexibility and scalability.

4.1 Hardware Requirements

The following hardware specifications are highly recommended for hosting the database on EECS servers:

- *Processor:* Multi-core CPU
- *Memory (RAM):* Minimum 8GB (16GB for optimized performance)
- *Storage:* Minimum 500GB SSD (for fast query execution and data storage)
- *Network Interface:* High-speed Ethernet or Wi-Fi for remote access

For company members ASKNE developing and testing the database locally, the following hardware specifications are recommended:

- *Processor:* Multi-core CPU
- *Memory (RAM):* Minimum 4GB (8GB for optimized performance)
- *Storage:* 10GB of free disk space for database storage
- *Interface:* Display screen, keyboard, or touchscreen for user input

4.2 Software Requirements

Here are the software tools ASKNE has chosen to ensure the efficient operation and management of the ASKNrEceive Library Management System:

- *Database Management System*: Our primary choice is MariaDB for EECS servers and MySQL for local machines like laptops and even on the cloud.
  - ASKNE chose MariaDB because of its open-source nature, great performance, and scalability which make it very ideal/reliable for handling large amounts of data for

libraries. For the local development, ASKNE chose MySQL for its user-friendly interfaces, and efficient querying capabilities for reliable usage and management.

- *Operating System*: ASKNE chose Linux (specifically Ubuntu for server environments), Windows 10/11, and macOS, as they provide good performance and security for hosting our reliable DBMS.
- *User Management and Security Tool*: ASKNE chose PHPMyAdmin.
    - It provides a web-based interface for managing MySQL and MariaDB databases, which offers an easy way for library staff to manage users' records, book data, and membership details. It also provides a graphical user interface (GUI) for tasks like querying, creating tables, and managing privileges.
- *Other Development Tools*:
    - GitHub for version control and team collaboration
    - MySQL Workbench for database modeling and query execution

5 Appendices

Refer to *01 - Project Plan/Vision* for any additional information on the scope and definition of the project.

Refer to *07 - Glossary* for any additional information regarding abbreviations and terms.

6 GitHub Repository Management

All members of the ASKNrEceive team will regularly manage, update, and commit to the GitHub Repository. The repository will be publicly available for view and accessed here: Library Database Management Project.

** **Note: All our Project Meeting Logs will be housed in the GitHub Repository on the Wiki Page. Please reference it as needed. Our Team Profile is also on the Wiki Page.**