Project document 2 - Database Requirements

Tej Gumaste, Jay Patel, Shayaan Mohammed, Kaleb Howard,

Andrew Reyes, James Hanselmann

Introduction

Project Overview

Our Library Database System will be designed to make inventory management more efficient, improve overall resource tracking, and generally provide a better experience for both end users and administrators. By keeping track of books, research articles, and technical equipment, our Database System will ensure that all of the library's resources are easy to track and locate at all times. Our system will be capable of generating varying library reports that will add to the efficient and organized experience.

Scope

The scope of our Library Database System can be summarized by including inventory management, item tracking system, technology checkouts, and overdue fees management. It will keep track of the books, articles, and technology that are in the system with inventory management. The item tracking system and the technology checkouts will focus on recording and keeping track of whether an item is checked in or out and its history. Overdue fees management will keep track of an item's due date and implement fines in the case of overdue items.

Stakeholders

Our Library Database System will include a wide range of stakeholders, each with their own roles and responsibilities, that will benefit from the new database system. Carol Smith, the Dean of the university library, will be one of our primary stakeholders, and she will have the responsibility to ensure the database supports the overall library goals as well as help with organizational planning. KU students, as the primary database users, expect the system to be able to quickly look up and borrow books, articles, and technical equipment for their academic needs.

Lawrence residents will also be included in the user base, and they will be provided access to a variety of library materials, making the library resources more accessible for all. Faculty members including professors, staff, and researchers will depend on up-to-date resource availability for teaching, research, and preparing lectures, and the database will help them quickly find and manage the materials they need. Also, library administrators and staff will use the system daily to manage the library catalog, add new items to the database, monitor checkouts and returns, track overdue items, and generate reports about usage. Lastly, the IT support team will be responsible for upkeep and maintaining the system's reliability and security, making sure that it runs without interruption for all users.

Requirements

Functional Requirements

The database system will perform the following essential functions:

User Administration

- Allow the library to add, update, and remove user accounts.
- Assign user roles such as student, faculty, or administrator.
- Authentication and Authorization mechanisms to maintain the access control.

Item Management

One of the database's primary components - the item management system, is in charge of keeping track, organizing, and preserving the library's many resources. This system lets users engage with the collection in a seamless manner while guaranteeing effective item tracking, storage, and accessibility.

- **Books**: Author, status, **ISBN**, year of release, type (fictional, academic, non-fictional, thriller, adult fiction), return date, storage (digital, physical: shelf, back office, storage).
- **Published Articles**: **Title, author,** status, return date, field of study, journal, year, DOI, storage (digital, physical: on shelf).
- **Technology**: Status, return date, physical storage, **serial number**, type (chargers, laptops), software (type, license duration: start, end date).

Tracking System for Items

- Keeps track of checked-out, returned, and overdue items.
- Identifies most borrowed and least borrowed items.
- Logs user engagement with library materials.
- Additional idea: Creates a waitlist of users requesting access to items?

Transaction Management

- Users can check out and return items through the system.
- A transaction log will record the date and time of each checkout and return.
- Handle requests for extensions of borrowed items

Overdue Fees Management

- The system will calculate overdue fees based on the timestamps.
- Overdue fees can be either billed directly to student accounts or paid on the spot.
- Admins will be able to configure fee policies based on item type and borrowing duration.

Reports & Analytics

- Reports on the number and types of items in the library, including usage trends.
- A list of items that have exceeded their return dates and corresponding user details.
- Tracks the usage of technology items, such as laptops and chargers
- Identifies the most popular books, articles, and items that have been overdue for statistical analysis.

Non-Functional Requirements

- Database queries must execute efficiently, with response times under a defined threshold
- The system should support concurrent users without significant performance degradation.
- The system should maintain the privacy of all client information and not reveal it to anyone.
- The transaction section should be timed to logout after certain minutes to maintain security.

Data Entities

1. User

- Attributes:
  - **Online ID (Primary Key)**: Unique identifier assigned to each user.
  - **Student/Faculty ID**: Official university identification number.
  - **Name**: Full name of the user.
  - **Email**: University email address.
  - **User Type**: student, faculty member, or admin.
  - **Payment**: Users may have outstanding fines or bills linked to their student account.
  - **Borrowing History**: Tracks all past transactions associated with the user.

2. Books

- Attributes:
    - **ISBN**[2]**(Primary key)**: International Standard Book Identifier, which is a 13-digit unique ID assigned to all commercially published books
    - **Return Date**: Date when this item is returned
    - **Author**: First name and last name of author(s)
    - **Year released**: The year released in YYYY format
    - **Type**: Genre/Type of book
        - Fiction
        - Non-Fiction
        - Academic
        - Thriller
        - Adult Fiction
    - **Storage**: Means of storing the book
        - Digital
        - Physical
            - Shelf
            - Back office
            - Internal Storeroom
    - **Status**: Denotes whether the book is checkout out, out of stock or available

3. Articles

- **Title(Primary Key)**: Title of Paper
- **Author(Primary Key)**: First name and last name of the author(s)
- **Status**: Denotes whether the book is checkout out, out of stock or available
- **Field of Study**: Indicates general discipline of the article (ex. Physics, chemistry etc.)
- **Journal**: Title of the journal the article is published in
- **Year**: year published
- **DOI**[3]: Stands for digital object identifier and is a unique code assigned to digital articles
- **Return Date**: Date when this item is returned
- **Storage**: Means of storing article
    - Digital
    - Physical:
        - Shelf
        - Back Office
        - Internal Storeroom

4. Technology

- **Status**: Denotes whether the book is checkout out, out of stock or available
- **Serial number (Primary Key)**
- **Return Date**: Date when this item is returned
- **Type**: Type of technology provided
    - **Hardware**:
        - **Type**:
            - Laptops
            - Chargers
    - **Software**:
        - License duration:
            - Start date
            - End date

5. Transactions(tentative requirements)

- **Transaction ID:** Unique number assigned to each transaction
- **Type**: Type of transaction conducted with library items
    - Checkout
        - **Due Date**: Date by which the item must be returned to be considered on time
    - Extension Request
    - One-time purchase

Hardware and Software Requirements

Our MySQL database will be running within a Docker container, using the MySQL image[1] , on one of our laptops. This will ensure that the database can be run from any group member's laptop, regardless of that laptop's operating system. Users can connect to the DB via Docker's CLI and execute queries from there.

Appendices

1. https://hub.docker.com/_/mysql
2. https://www.isbn.org/ISBN_history
3. https://www.doi.org/