Part 2: Database Requirements

Introduction

Project Overview

- Purpose
  - This database is designed to replace outdated, handwritten record-keeping systems for tracking library assets. By automating the process, librarians and staff can reduce the time spent on administrative tasks and focus more on assisting students. It also minimizes human errors, ensures accurate record-keeping, and improves overall efficiency in managing library materials.

- Intended Use
  - The database serves as an automated system for tracking library assets, making it easier to manage inventory, lending, and returns. It will be used by students, teachers, librarians, school administrators, and the IT department. Students will primarily use it to search the catalog, check out books, and track their borrowing history. Librarians and administrators will oversee inventory, manage book conditions, and handle late fees or lost book charges. By streamlining these processes, the database ensures smoother library operations and better access to educational resources.

Project Scope

The School Library Database project will create a centralized system for managing library assets, user accounts, and borrowing activities. It will track various library items, their availability, and condition while monitoring user accounts. The system will automate fee calculations, track borrowing history, and loan durations. Administrators can impose fees, manage reservations, and generate reports on library usage. Additionally, the database will meet essential non-functional requirements, such as security, data integrity, scalability, and reliability, ensuring smooth operations with real-time updates, minimal downtime, and multi-device compatibility.

Glossary

- **ISBN:** (International Standard Book Number) is a unique identifier assigned to books for tracking and cataloging purposes.
- **IT:** (Information Technology) refers to the use, development, and management of computer systems, networks, and software to store, process, and transmit data.
- **Linux:** Open-source, Unix-like operating system kernel that powers a wide range of devices, from servers and desktops to embedded systems.
- **SQL:** (Structured Query Language) is a domain-specific language used to manage, query, and manipulate relational databases.
- **SQLite 3:** Lightweight, self-contained, serverless database engine that uses a single file to store data and supports SQL for efficient data management.

Stakeholders

- Students - Access to only checking out items, view check out status of books and other media
- Teachers - Access to checking out items, checking in items, and editing status of books and other media
- School Admin / Librarians - Full access to frontend of database
- IT department - Full access to frontend and backend of database

Requirements

Functional Requirements

There needs to be an interface to search the catalog, reserve items, check loan status, and such for students. For the library staff, there needs to be an interface to check out items, process returns, add new items, and manage clients.

All users will be able to:

- Search the catalog
  - Keyword, ISBN, Title, Author, and Category search
- Reserve Items
  - User ID, book ID, reservation date, and due date
- Check Loan Status
  - See if an item is loaned out to another student. Let librarians see if an item is overdue. Let students see if they have an overdue item.
- Process Returns
  - Users place items in a physical return slot. Librarians will process the returned items.
- Manage clients
  - Supervise the interactions and needs to the students

Librarians will be able to:

- Check in returned items
- Add new items
  - Bring items to the library for students to borrow or access

School Admin / Librarians will be able to:

- Override late fees
  - If a user has a high enough status such as a school admin/teacher they will have the the ability to override late fees
- Apply Fees

School Admin / Librarians and IT will be able to:

- Manage membership records
- Manage membership permissions

Data Entries

- **Books**
  - The attributes are ISBN, title, author, category, publication date, publisher name, edition, price, condition, faculty_only, book_id, status
- **Author**

    - The attributes are Full Name (First Name, Middle Name, Last Name), author ID
- **Magazines**
    - The attributes are title, issue number, publication date, publisher, price, condition, magazine_id, status
- **Digital Media**
    - The attributes are title, medium (DVD, VHS, etc.), publication date, publisher, price, condition, digital_id, status
- **Fees**
    - The attributes are FeeID, Amount, Reason, Student ID, Faculty_ID
- **Students**
    - The attributes are Student ID, Date of Birth, Full Name (First Name, Middle Name, Last Name), email, phone #, address
- **Faculty**
    - The attributes are Faculty ID, Full Name (First Name, Middle Name, Last Name), email, phone #, address, role

Non-Functional Requirements

- Security: To keep user data private. Additionally, user data can only be updated by the user or an admin
- Data integrity: Library records are kept up to date with the latest user changes
- Scalability: The system should efficiently handle increasing numbers of users and transactions
- Reliability: The system should function correctly and be available with minimal downtime, ensuring users can access library records whenever needed.

Hardware and Software Requirements

Hardware

The database system would be required to run on our personal laptops and the school computers.

Software

The database system would be required to run on a Linux operating system. The SQL database engine that would be required is SQLite 3. Then Python would be required for additional UI needs for the database system