Project Part 2: Database Requirements

Introduction:

The Library Management System is designed to streamline the organization and tracking of a fictional library's inventory over the course of a working day. The database will store essential details about different media items such as books and magazines, track users' accounts, enforce borrowing limits, and calculate overdue fees. Additionally, the tool will provide library staff with check-ins and check-outs to ensure that every piece of media is accounted for. The goal is to enhance operational efficiency for library workers while improving the borrowing experience for library visitors.

Scope

This project focuses on developing a command-line application that efficiently manages library materials and user accounts. The system will:

- Store book details, user records, and borrowing history.
- Enforce borrowing rules based on membership type.
- Calculate and track overdue fees.
- Allow library staff to check materials in and out and generate reports on borrowing trends.

While the initial version will be a command-line application, the system will be designed with potential future expansion into a graphical user interface (GUI) for improved usability. Advanced features such as book reservations are beyond the current scope but may be considered for future iterations.

Glossary

- GUI – Graphical user interface
- ISBN – International Standard Book Number
- PostgreSQL – A relational database management system designed to efficiently manage structured data with complex relationships
- ACID – A set of principles that ensure database transactions are processed reliably (Atomicity, Concurrency, Isolation, Durability)
- ID (Identity) – A unique identifier assigned to entities such as books, users, or transactions in the database.

Stakeholder Information:

- The stakeholders of the Library Management System include the individuals and groups who will use or manage the system.
- **Library Staff:** They are responsible for maintaining content in the system, managing book and media item check-ins and check-outs, enforcing borrowing policies, and generating reports. They will need an efficient system to keep track of inventory and user activity with minimal effort.
- **Library Members:** These are the clients who will use the system to search for books and media items, check their borrowing status, reserve items, and return borrowed materials. A well-organized system will make it easier for them to access and manage their loans.
- **Database Administrators:** This group ensures the system runs smoothly, maintains data integrity, and troubleshoots technical issues. They are responsible for implementing updates and managing database security.
- **Board of Directors:** They oversee the project development, evaluate deliverables, and provide feedback. Their role ensures the project meets academic and technical standards.

Requirements:

Essential functions the database must perform (ie. user administration, data entry, retrieval, updates, deletions, and report generation)

- Books have titles, authors, genres, ISBNs, and publication years.
- Magazines have titles, issue numbers, and publication dates.
- Digital media has titles, authors, genres, ISBNs, and publication years.
- Each physical copy of a book, magazine, or digital media license has a unique item ID.
- Clients can borrow books, magazines, and digital media.
- Clients have IDs, names, membership types, and account status.
- Clients can have various membership types that affect aspects like late fees and return times
- Each instance of content being borrowed has a date borrowed, expected return date, date returned, user ID, and item ID.

- Late fees can be automatically calculated and applied.
- Reports can be made based on book availability, overdue items, borrowing trends, and user activity.
- Users and library staff can interact with the system to do various actions
- Some actions require authentication using a library-specific passphrase

Other Queries and Reports

- Hot Items Report
  - A detailed report breaking down the most borrowed items in the past 30 days that are currently available to borrow to help clients find new items.
- Percentage Checked Out
  - Query that returns percentage of the time that a specific book, magazine, or was/is checked out in the last 90 days. Is based on the specific printing, meaning each copy of the same media may have a different percentage checked out. This can be used to see if a book would be a good candidate to purchase additional copies for.

Main Entities:

- Book
  - Title
    - Data Type: VARCHAR (100)
    - Constraints: NOT NULL
  - Author
    - Data Type: VARCHAR (100)
    - Constraints: NOT NULL
  - Genre
    - Data Type: VARCHAR (20)
  - ISBN
    - Data Type: INT
    - Constraint: Value >= 0
  - Publication year
    - Data Type: YEAR
  - Availability status
    - Data Type: ENUM (‘Available’, ‘Unavailable’)
    - Constraints: NOT NULL

  - Item ID
    - Data Type: INT
    - Constraints: Primary key, Value >= 0, NOT NULL, Auto increment
- Magazine
  - Title
    - Data Type: VARCHAR (100)
    - Constraints: NOT NULL
  - Issue #
    - Data Type: INT
    - Constraints: NOT NULL
  - Publication date
    - Data Type: DATE
  - Availability status
    - Data Type: ENUM (‘Available’, ‘Unavailable’)
    - Constraints: NOT NULL
  - Item ID
    - Data Type: Int
    - Constraints: Primary key, Value >= 0, NOT NULL, Auto increment
- Digital Media
  - Title
    - Data Type: VARCHAR (100)
    - Constraints: NOT NULL
  - Author
    - Data Type: VARCHAR (100)
    - Constraints: NOT NULL
  - Genre
    - Data Type: VARCHAR (20)
  - ISBN
    - Data Type: INT
    - Constraint: Value >= 0
  - Publication year
    - Data Type: YEAR
  - Availability status
    - Data Type: ENUM (‘Available’, ‘Unavailable’)
    - Constraints: NOT NULL
  - Item ID
    - Data Type: INT

- Constraints: Primary key, Value >= 0, NOT NULL, Auto increment

- Client
    - Client ID
        - Data Type: INT
        - Constraints: Primary Key, Auto increment, INCREMENT, NOT NULL
    - Name
        - Data Type: VARCHAR (100)
        - Constraint: NOT NULL
    - Membership Type
        - Data Type: ENUM (‘Regular’, ‘Student’, ‘Senior Citizen’, ‘Other’)
        - Constraints: NOT NULL
    - Account Status
        - Data Type: ENUM (‘Active’, ‘Suspended’, ‘Inactive’)
        - Constraints: NOT NULL
    - Email Address:
        - Data Type: VARCHAR (255)
        - Constraints: UNIQUE, NOT NULL
    - Phone Number
        - Data Type: VARCHAR (15)
        - Constraint: UNIQUE, NOT NULL

Hardware & Software Requirements

This application will be hosted using a cloud service. Using a cloud service ensures that the database will remain easily accessible to all the users and developers of the system. Due to the ubiquity of the platform, we intend to use the database hosting service provided by Amazon Web Services (AWS). AWS is also convenient due to the sheer number of database frameworks they support.

The framework we intend to use for our database is PostgreSQL. PostgreSQL is nice because of how well it can reflect our ER model. Additionally, it fully supports ACID, more advanced queries, and still performs better than a typical MySQL implementation. We will be using Python for scripting and GitHub for version control.

Clients of the system will need a stable internet connection and the ability to run Python 3.x, as the backend of the system will rely on Python for processing commands and interacting with the database.

Appendices

None as of 3/1/2025