Introduction

Project Overview -

This project involves the design, implementation, and management of a relational database system for a small library. The system is intended to efficiently handle the library's collection of loanable items, manage different types of memberships, enforce borrowing policies, and generate insightful reports. By creating a well-structured database, the library can streamline operations, improve accessibility to resources, and ensure proper tracking of borrowed materials.

Scope -

This project will include a user interface for both the library staff and their clients. The library staff interface will include the ability to check out items, process returns, add new items, and manage client accounts. The client side will allow for searching the catalog, reserving items, checking loan status, etc. This project will be structured using SQL RDBMS databases. We will have client IDs, staff IDs, and inventory IDs to keep track of all interactions between users, the material, and the library staff. The database will implement rules for borrowing, due dates, and fines, ensuring compliance with library policies. Additionally, the system will provide reporting features for analyzing borrowing trends and member activity.

Stakeholders

The stakeholders of this project are the following:

- Library/Database Owner
    - Owns the database. This individual holds administrative powers. They can create, edit, and remove elements within the database. They can also add or remove librarians.
- Government
    - Ensures the library follows legal and ethical operations. They can audit the library and its database at any time.
- Librarians
    - Are able to act as users, as well as be able to search other users and manage their accounts by checking if they've checked out an item, if they have any fines, if they have any overdue items, etc. They are also able to add items to the catalog and process check-outs.
- Users
    - Can look up books, magazines, and other items that can be looked up in the database. They can also view their accounts to see any fines and see what books they have checked out. They can also reserve items.

Requirements

Functional Requirements:

User & Staff Administration -

- User Registration & Login:
  Allow users to sign up, log in, update profiles, and handle password recovery.
- Staff Management:
  Enable staff to create/update accounts and manage library items.

Media Checkout/Return -

- **Checkout Process:**
  - Verify user membership limits (5 items for adults, 8 for youth).
  - Check item availability and age restrictions.
  - Update the media record with User_Id, Checked_Date, Due_Date, and change Availability to "Checked Out".
- **Return Process:**
  - Mark items as "Available" and clear checkout information.
  - Check and update the Reservation Queue if needed.

Reservation/Waitlist System -

- Place a Reservation:
  Allow users to add themselves to a queue if the item is already checked out.
- Queue Management:
  Automatically notify or assign the next user when an item becomes available.

Overdue and Fee Management -

- Overdue Tracking:
  Automatically flag items as overdue when Due_Date is exceeded.
- Fine Calculation:
  Calculate fees based on overdue duration and membership type (cheaper for oldies).
- Fee Payment Handling:
  Track and update users' Current_Fee_Amount.

Report Generation -

- **Standard Reports:**
  - List overdue items.
  - Display current checkouts per user.
  - Summarize inventory status (available, checked out, reserved).
- **Additional Queries:**

    - Popular media items (e.g., items most frequently checked out).
    - Reservation queue details per item.
    - Historical trends in media circulation.

Data Entities:

User:

- *Attributes:* User_Id(Str), Media_Checked(Int), Name(Str), DOB(datetime), Email(Str), Password(Str), Phone(Str), Membership(Str), Date_Registered(datetime), Current_Fee_Amount(Int)
- *Constraints:* Limited # of media a user can checkout depending on membership (5 for adult and 8 for youth), for youth account age must be less than 13, youth members will not be allowed to check out certain media, fees for overdue media, membership cheaper for seniors

Staff:

- *Attributes:* Staff_Id(Int), Name(Str), Email(Str), Password(Str)
- *Constraints*: works for the library, can be both a user and a staff member, must be an adult

Magazine:

- *Attributes:* Magazine_Id(Int), Title(Str), Issue_Number(Int), Publication_Date(datetime), Publisher(Str), Genre(Str), ISSN(Str), Date_Lib_Acquired(datetime), Availability(Str), User_Id(Str), Checked_Date(datetime), Due_Date(datetime)
- *Constraints:* Can only be checked out by one user at a time

Games:

- *Attributes:* Game_Id(Int), Publication_Date(datetime), Publisher(Str), Console(Str), UPC(int), Date_Lib_Acquired(datetime), Availability(Str), User_Id(Str), Checked_Date(datetime), Due_Date(datetime)
- *Constraints:* Can only be checked out by one user at a time

Books:

- *Attributes:* ISBN(Str), Title(Str), Author(Str), Release_Date(datetime), Genre(Str), Publication_Date(datetime), Date_Lib_Acquired(datetime), Availability(Str), User_Id(Str), Checked_Date(datetime), Due_Date(datetime)
- *Constraints:* Can only be checked out by one user at a time

Movies:

- *Attributes:* IMDb_Id(Int), Title(Str), Director(Str), Release_Date(datetime), Availability(Str), Genre(Str), Duration(Str), Rating(Str), IMDb_SCore(Int), Metascore(Int), Date_Lib_Acquired(datetime), User_Id(Str), Checked_Date(datetime), Due_Date(datetime)
- *Constraints:* Can only be checked out by one user at a time

Reservation Queue:

- *Attributes:* Media_Id(Int), User_id(Int), Queue_Number(Int)
- *Constraints:* A user can only be in the queue to check out some media once per item

Non-Functional Requirements:

Performance:

- The system should process user logins within 5 seconds.
- Media checkout and return transactions must be completed within 5 seconds.
- Report generation should take no more than 5 seconds.

Scalability:

- The system must support up to 1,000 concurrent users without performance issues.
- It should handle up to 1 million media items in the database.

Availability:

- The system should be available 99.9% of the time, excluding scheduled maintenance.
- Maintenance downtime should not exceed 4 hours per month.

Security:

- Sensitive user data must be securely encrypted.
- Only authorized staff can access or modify sensitive user information.
- The system should track all login attempts, successful and unsuccessful, for security purposes.

Usability:

- The user interface should be intuitive and easy to navigate, with clear prompts and feedback.

Data Integrity:

- The system must ensure that media availability status is always accurate and up-to-date.
- User fee calculations should be precise, with no rounding errors.

Backup and Recovery:

- System data must be backed up daily, with backups lasting 30 days.
- In case of system failure, data recovery should be completed within 1 hour.

Compliance:

- The system must comply with applicable data privacy regulations.
- User consent must be obtained before storing personal information, EULA.

Hardware and Software Requirements

We are going to put the database onto either the KU cycle servers or Chris's computer if we can't use the cycle servers while using MySQL. We are planning to use MySQL since it's the language that we most know for putting data into a database. The database will be managed using the MySQL server and SQL queries.

Meeting Notes:

Date: February 17, 2022
Time: 9:00 AM - 10:00 AM
Location: Discord Meeting
Objective: To discuss our objects in our database.
Team Members Present: Chris, Carter, Christina, Elizabeth, Jaret, Ryan

Task Completion Confirmation:

Chris: Yes
Carter: Yes
Christina: Yes
Elizabeth: Yes
Jaret: Yes
Ryan: Yes

Brainstorming Session:

- We started by talking about all that we got done.
- Next we started filling out all of our books, magazines, etc. onto a google sheet.
- We then decided who would be working on what part of Part 2 of the project.
- Next, we talked about constraints, and extra things we may need

Tasks Allocated:

Chris: Work on non-functional requirements.
Carter: Work on hardware and software requirements.
Christina: Work on Stakeholders.
Elizabeth: Work on Introduction.
Jaret: Work on functional requirements.
Ryan: Work on data entities.

Follow-Up Actions:

- Work on Part 2 of the project!

- Double check features for the next meeting.

Schedule the next meeting: 3/3/2025 @ 9:00AM