Group 15 Project: Deliverable 1

SER 322 – Database Management

Group Members:

Zachary Davis
Kelvin Hoang
John Sullins

1. Application Domain

Library Management

2. Application Name

Library Database System

3. Application Overview and Description

The Library Database System is designed to manage a library's catalog, patrons, staff, and borrowing activity in a structured and efficient manner. The system stores information about books, including identifying details and classification by topic. Books may be written by one or more authors, and each author may contribute to multiple books.

The system also maintains records for library members who borrow books and employees who manage and process borrowing transactions. Each checkout and return transaction is recorded so the library can track which member currently has a book, when it is due, and whether it has been returned. The database supports common library operations such as catalog browsing, tracking member borrowing history, and generating reports including overdue items and frequently borrowed books or topics.

4. Requirements Statement

The Library Database System is intended to support the day-to-day operations of a library by providing a centralized database for managing books, members, employees, and borrowing transactions. The system allows library staff to add, update, remove, and view information related to books, topics, members, and employees. Employees are able to check books out to members and record book returns, allowing the system to track availability, due dates, and overdue items.

The system also supports queries and reports such as searching for books by title, author, or topic, viewing a member's current checkouts and borrowing history, and identifying the most frequently borrowed books or topics. The database is embedded in a Java application that provides a user interface for interacting with these features. The primary users of the system are library employees, while members are supported through search and borrowing-history lookups.

5. Library Database Requirements

Functional Requirements

- The system shall allow users to **add, edit, delete, and list books** in the library catalog.
- The system shall allow users to **add, edit, delete, and list topics** and associate topics with books.
- The system shall allow users to **add, edit, delete, and list authors** and associate authors with books.
- The system shall allow users to **add, edit, delete, and list members**.
- The system shall allow users to **add, edit, delete, and list employees**.
- The system shall allow an employee to **check out a book** to a member by creating a checkout record that includes a checkout date and due date.
- The system shall allow an employee to **return a book** by updating the corresponding checkout record with a return date or status.

Query Requirements

- Search for books by **title**, **author**, or **topic**.
- List all books currently checked out by a specific member.
- List all overdue checkouts.
- View the borrowing history of a member.
- Generate reports identifying the most borrowed books and topics.

6. Core Relationships

- A **Book** has one or more **Authors**, and an **Author** can write one or more **Books**.
- A **Book** is classified under one or more **Topics**, and a **Topic** can include many **Books**.
- A **Member** can have zero or more **Checkouts**, and each checkout belongs to exactly one **Member**.
- Each **Checkout** is processed by one **Employee**, and an **Employee** may process many **Checkouts**.

- Each **Checkout** references the borrowed **Book**.

7. Entities

- **Book** (catalog item)
- **Author** (writes books)
- **Member** (borrows books)
- **Employee** (manages and processes borrowing activity)
- **Topic** (category or classification of books)
- **Checkout** (tracks book loans and returns)

8. Entity Attributes (Initial List)

- **Book**
  Book(BookID, ISBN, Title, PublicationYear)
- **Author**
  Author(AuthorID, Name)
- **Member**
  Member(MemberID, Name)
- **Employee**
  Employee(EmployeeID, Name, Role)
- **Topic**
  Topic(TopicID, Name)
- **Checkout**
  Checkout(CheckoutID, CheckoutDate, DueDate, ReturnDate, BookID, MemberID, EmployeeID)

Group 15 Project: Deliverable 2

Zachary Davis
Kelvin Hoang
John Sullins

Entities with Attributes:

1. Book(int **BookID**, string ISBN, string Title, int PublicationYear, int TotalCopies)
2. Author(int **AuthorID**, string Name)
3. Member(int **MemberID**, string Name, string Email)
4. Employee(int **EmployeeID**, string Name, string Role, string Email)
5. Topic(int **TopicID**, string Name)
6. Checkout(int **CheckoutID**, date CheckoutDate, date DueDate, date ReturnDate, int BookID, int MemberID, int EmployeeID)

Associative Entities:

1.) Book_Author

- BookID(PK, FK to Book,BookID)
- AuthorID (PK,FK to Author, AuthorID)

2.) Book_Topic

- BookID(PK, FK to Book,BookID)
- TopicID(PK, FK to Topic, TopicID)

Relationships:

1. A Book has one or more Authors.
2. An Author can write one or more Books.
3. A Book is classified under one or more Topics.
4. A Topic can include many Books.
5. A Member can have zero or more Checkouts.
6. Each Checkout belongs to exactly one Member.
7. Each Checkout is processed by one Employee.
8. An Employee may process many Checkouts.
9. Each Checkout references the borrowed Book.

References:

- Checkout.MemberID references Member.MemberID
- Checkout.EmployeeID references Employee.EmployeeID
- Checkout.BookID references Book.BookID

- Book_Topic.BookID references Book.BookID
- Book_Topic.TopicID references Topic.TopicID
- Book_Author.BookID references Book.BookID
- Book_Author.AuthorID references Author.AuthorID