Assignment 2

Entity-Relationship Diagram (ERD) and Relational Data Model

Submitted by:

Nada Adel - ID: 202102397

CCAS.4.3: Software Engineering Design

Submission Date: October 18, 2024

1 Entities and Attributes

1. **User:** The User entity stores information about the users of the library system.
   (a) UserID (Primary Key) - A unique identifier for each user.
   (b) FirstName - The first name of the user.
   (c) LastName - The last name of the user.
   (d) Email - The email address of the user.
   (e) PhoneNumber - The contact number of the user.
2. **Book:** The Book entity stores information about the books in the library.
   (a) BookID (Primary Key) - A unique identifier for each book.
   (b) Title - The title of the book.
   (c) Author - The author of the book.
   (d) Condition - The condition of the book (e.g., New, Good, Poor).
   (e) Availability - Indicates if the book is currently available for loan.
   (f) Location - The location of the book in the library (e.g., Floor, Section, Shelf).
   (g) CategoryID (Foreign Key) - A reference to the Category entity to classify the book's genre.
   (h) UpdatedBy (Foreign Key) - A reference to the Librarian entity to track who last updated the book's information.
   (i) UpdateDate - The date the book's information was last updated.
3. **Loan:** The Loan entity tracks the books borrowed by users.
   (a) LoanID (Primary Key) - A unique identifier for each loan.
   (b) UserID (Foreign Key) - A reference to the User entity to track who borrowed the book.
   (c) BookID (Foreign Key) - A reference to the Book entity to track the borrowed book.
   (d) LoanDate - The date the book was borrowed.
   (e) DueDate - The date the book is due for return.
   (f) ReturnDate - The date the book was returned.
   (g) ReminderSent - Indicates whether an overdue reminder has been sent (Boolean).
   (h) RenewalCount - Tracks the number of times a book loan has been renewed.
4. **Reservation:** The Reservation entity tracks books that users have reserved.
   (a) ReservationID (Primary Key) - A unique identifier for each reservation.
   (b) UserID (Foreign Key) - A reference to the User entity to track who made the reservation.
   (c) BookID (Foreign Key) - A reference to the Book entity to track the reserved book.
   (d) ReservationDate - The date the book was reserved.
   (e) IsAvailable - Indicates whether the reserved book is available for pickup.
5. **Fine:** The Fine entity tracks fines imposed on users for late book returns.
   (a) FineID (Primary Key) - A unique identifier for each fine.
   (b) UserID (Foreign Key) - A reference to the User entity to track who incurred the fine.
   (c) BookID (Foreign Key) - A reference to the Book entity for which the fine was imposed.
   (d) Amount - The amount of the fine.
   (e) DatePaid - The date the fine was paid.
6. **Category:** The Category entity classifies books into genres.
   (a) CategoryID (Primary Key) - A unique identifier for each category.

(b) CategoryName - The name of the category (e.g., Fiction, Non-fiction, Science).

7. **Notification:** The Notification entity stores notifications sent to users.
   (a) NotificationID (Primary Key) - A unique identifier for each notification.
   (b) UserID (Foreign Key) - A reference to the User entity to track who receives the notification.
   (c) BookID (Foreign Key) - A reference to the Book entity related to the notification.
   (d) NotificationType - Describes the type of notification (e.g., OverdueReminder, ReservedBookAvailable).
   (e) DateSent - The date the notification was sent.
8. **BorrowedHistory:** The BorrowedHistory entity maintains a record of all past loans.
   (a) HistoryID (Primary Key) - A unique identifier for each history record.
   (b) UserID (Foreign Key) - A reference to the User entity to track who borrowed the book.
   (c) BookID (Foreign Key) - A reference to the Book entity that was borrowed.
   (d) LoanDate - The date the book was borrowed.
   (e) DueDate - The date the book was due for return.
   (f) ReturnDate - The date the book was returned.
9. **Review:** The Review entity allows users to rate and review books.
   (a) ReviewID (Primary Key) - A unique identifier for each review.
   (b) UserID (Foreign Key) - A reference to the User entity who submitted the review.
   (c) BookID (Foreign Key) - A reference to the Book entity being reviewed.
   (d) ReviewText - The text of the user's review.
   (e) Rating - The user's rating for the book.
   (f) ReviewDate - The date the review was submitted.
10. **InventoryReport:** The InventoryReport entity generates reports on the library's inventory.
    (a) ReportID (Primary Key) - A unique identifier for each report.
    (b) LibrarianID (Foreign Key) - A reference to the Librarian entity that generated the report.
    (c) TotalBooks - The total number of books in the library.
    (d) AvailableBooks - The number of books currently available.
    (e) BorrowedBooks - The number of books currently borrowed.
    (f) ReservedBooks - The number of books currently reserved.
11. **Librarian:** The Librarian entity tracks information about the librarians who manage the system.
    (a) LibrarianID (Primary Key) - A unique identifier for each librarian.
    (b) FirstName - The first name of the librarian.
    (c) LastName - The last name of the librarian.
    (d) Email - The email address of the librarian.
    (e) PhoneNumber - The contact number of the librarian.

2 Connections Between Entities

1. **User - Loan:** One-to-Many relationship. A user can borrow multiple books, but a loan is associated with only one user.
2. **User - Reservation:** One-to-Many relationship. A user can reserve multiple books, but each reservation is associated with one user.
3. **User - Fine:** One-to-Many relationship. A user can incur multiple fines, but each fine is related to only one user.
4. **User - Review:** One-to-Many relationship. A user can write multiple reviews, but each review is associated with one user.
5. **Book - Loan:** One-to-Many relationship. A book can be loaned multiple times, but each loan is associated with one book.
6. **Book - Reservation:** One-to-Many relationship. A book can be reserved by multiple users, but each reservation is associated with one book.
7. **Book - Fine:** One-to-Many relationship. A book can have multiple fines associated with it, but each fine is for one book.
8. **Book - Review:** One-to-Many relationship. A book can have multiple reviews, but each review is for one book.
9. **Book - Category:** Many-to-One relationship. A book belongs to one category, but a category can have many books.
10. **Librarian - Book:** Many-to-One relationship. A librarian can update multiple books, but each update is attributed to one librarian.
11. **Librarian - InventoryReport:** One-to-Many relationship. A librarian can generate multiple inventory reports, but each report is generated by one librarian.

3 Functional Requirements and How They're Met

3.1 Must Have (M) Requirements

1. **FR1: The library system must record new books in the library.**
   The Book entity includes attributes like BookID, Title, Author, CategoryID, Condition, and Availability. Librarians can update these details, ensuring that new books are recorded.
2. **FR2: The system must allow users to search for books by title, author, or category.**
   The Book entity includes Title, Author, and a foreign key CategoryID pointing to the Category entity, allowing books to be searched by these fields.
3. **FR3: The system must maintain user records, including contact details and borrowed books.**
   The User entity includes UserID, FirstName, LastName, Email, and PhoneNumber to store user information. Additionally, the Loan and BorrowedHistory entities record which books have been borrowed by which users.
4. **FR4: The system must support the borrowing and returning of books by users.**
   The Loan entity tracks LoanID, UserID, BookID, LoanDate, DueDate, and ReturnDate, allowing for borrowing and returning of books.
5. **FR5: The system must send overdue reminders to users.**
   The Notification entity includes NotificationID, UserID, BookID, and NotificationType, which can be used to send overdue reminders when the DueDate in the Loan entity is passed.
6. **FR6: The system must allow reservation of books by users.**
   The Reservation entity includes ReservationID, UserID, BookID, ReservationDate, and IsAvailable, allowing users to reserve books.

7. **FR7: The system must maintain a record of each book's borrowing history.**
   The BorrowedHistory entity records HistoryID, UserID, BookID, LoanDate, DueDate, and ReturnDate, tracking the borrowing history of each book.

8. **FR8: The system must allow librarians to update book information, such as condition, availability, and location within the library.**
   The Book entity contains Condition, Availability, and Location attributes. The UpdatedBy and UpdateDate attributes allow librarians to update book information.

9. **FR9: The system must provide an option to generate fine receipts for users who return books late.**
   The Fine entity includes FineID, UserID, BookID, Amount, and DatePaid, allowing fines to be generated and managed.

3.2 Should Have (S) Requirements

1. **FR10: The system should generate reports on the library's inventory, including available, borrowed, and reserved books.**
   The InventoryReport entity includes ReportID, TotalBooks, AvailableBooks, BorrowedBooks, and ReservedBooks, allowing the system to generate inventory reports.

2. **FR11: The system should allow users to renew book loans online.**
   The Loan entity includes RenewalCount, which tracks how many times a book has been renewed, allowing users to renew loans.

3. **FR12: The library system should provide notifications to users when a reserved book becomes available.**
   The Notification entity includes NotificationType, allowing the system to notify users when a reserved book becomes available.

4. **FR13: The system should support the registration of new users.**
   The User entity captures user details such as UserID, FirstName, LastName, Email, and PhoneNumber, enabling new user registrations.

5. **FR14: The system should provide a book categorization feature based on genre.**
   The Category entity is linked to the Book entity via CategoryID, allowing books to be categorized by genre.

3.3 Could Have (C) Requirements

1. **FR15: The system could allow users to rate books.**
   The Review entity includes ReviewID, UserID, BookID, ReviewText, and Rating, allowing users to rate and review books.

2. **FR16: The library system could allow users to write reviews for books.**
   Same as FR15. The Review entity stores user-written reviews.

3. **FR17: The system could suggest books based on user history.**
   This functionality could be developed later by using the BorrowedHistory and Review entities to recommend books based on user activity.