Library Database Manager

Author: Pavel Halík

E-mail: pavel.halik06@gmail.com

School: Střední průmyslová škola elektrotechnická, Praha 2, Ječná 30

Date: 9.1. 2026

Project Type: School Project – Database Application

GitHub repository: https://github.com/Forkxel/Library-Database-Manager

1. Project Overview

The Library Database Manager is a Windows Forms application built to manage a small library. The system allows users to:

- Add, update, delete, and view books, authors, categories, members, and loans.
- Handle borrowing and returning books using transactions to ensure data consistency.
- Generate reports on book loans.
- Import data from CSV files for authors and categories.

The project uses SQL Server as a relational database and implements the Repository Pattern for database access.

2. Requirements & Use Cases

Functional Requirements

1. Manage authors, categories, books, members, and loans.
2. Borrow and return books using database transactions.
3. Display data in a DataGridView.
4. Import authors and categories from CSV.
5. Generate a summary report of author and book statistics.

Use Case Example

Borrow a Book:

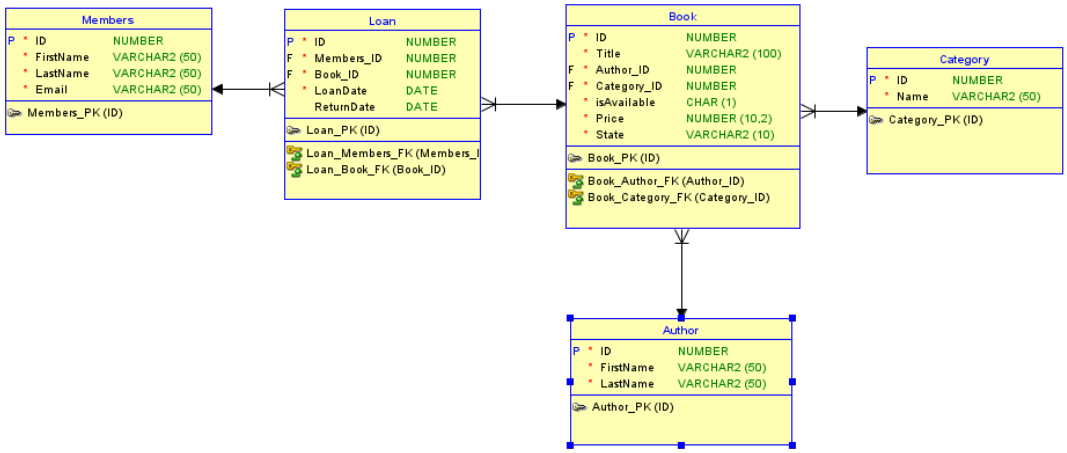
1. User selects a member and a book.
2. System checks if the book is available.
3. Loan record is created, and the book's availability is updated.
4. All operations are executed within a transaction to prevent inconsistencies.

Return a Book:

1. User selects a loan to return.
2. The loan's returnDate is updated, and the book is marked as available.
3. Operations are executed within a transaction.

3. Database Model

The system uses SQL Server and user can import the database using Export.sql with the following tables:



1. Author: id, firstName, lastName
2. Category: id, name
3. Members: id, firstName, lastName, email
4. Book: id, title, categoryId (FK), authorId (FK), isAvailable, price, state (enum: New, Damaged, Used)
5. Loan: id, memberId (FK), bookId (FK), loanDate, returnDate

4. Transactions behavior

BorrowBook Transaction:

1. Check if book is available.
1. If yes, insert new loan record.
2. Update book availability.
3. Commit transaction.

ReturnBook Transaction:

1. Retrieve loan with returnDate IS NULL.
2. Update returnDate.
3. Mark book as available.
4. Commit transaction.

5. Import, Config

Import:

- Authors and categories can be imported from CSV files using OpenFileDialog.

Authors:

```
firstName,lastName
Karel,Čapek
Božena,Němcová

```

Categories:

```
name
Sci-Fi
Drama
History
```

Config:

- Configuration is loaded from the appsettings.json that is placed in the same folder as the exe file.

```
{
  "ConnectionString": "Server=SERVER_EXAMPLE;Database=DATABASE_EXAMPLE;User Id=USER_EXAMPLE;Password=PASSWORD_EXAMPLE;"
}
```

6. Installation

All is written in the README.

7. Libraries

- System.Data.SqlClient

8. Summary

The Library Database Manager meets the following:

- Repository Pattern implemented.
- Five tables with relationships and constraints.
- Transactions for loan operations.
- CSV import supported.
- Test scenarios documented.
- Reports and views implemented for aggregated data.