CS6360.MS1 Spring 2021

Assignment 4 – Database Program

Now that you have designed the database for a library system, your next task is to write a scaled-back library program. Ideally, your program would have table-maintenance dialogs for every actual data table other than those linked to the book itself. Since that isn’t practical, create a screen that lets you add, change, and delete books. The purpose of this assignment is for you to learn how to interface to a database from within a program.

Add, change, and delete can work with a limited set of tables, which you have designed in the previous assignment. Use your design. For example, you will need to handle multiple copies of a book, possibly in multiple formats. Stored procedures will help greatly.

Always maintain referential integrity both at the database level and at the level of your program; that is, only allow delete of a secondary table item if there are no references to it. Here is a sample of how the screen could look, although you should have considerably more information; this is intended as a suggestion, not definitive. Obviously, the title can be quite long. The list of authors should be separated by semicolons so that your program can use the *split()* function to separate them. The publisher should be a drop-down. In a real system, there would be an “Add” button next to this so you can add a new publisher.

The Save button saves the new book to your database. You cannot have two books with the same ISBN, so catch that as an error. When you save a book, put the information in the list shown.

The Search button takes any of the shown fields except Publisher and searches for books that match any non-null field. You can use the “like” operator in SQL to find partial matches on authors and titles, but ISBN and Dewey Decimal numbers must be exact. Show all matches in the list. If a user clicks an item in the list, fill in the entry fields and go to change mode to update book information. Now the Save button updates a modified record

The Clear button clears all fields, including the list, and returns to “Add” mode.

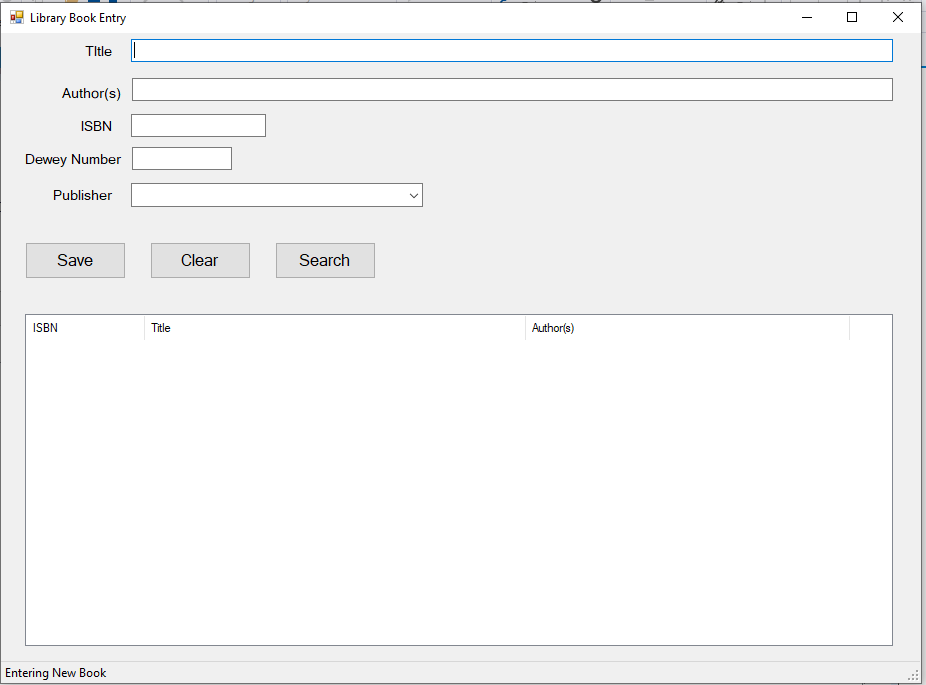
The status bar at the bottom can be used to show the mode as well as completed actions.

You can use SSMS or MySQL Workbench to add authors and publishers, and data in any other secondary tables.

Allowable languages are Java and C#. If you enjoy a challenge and have lots of time, C++ will work too, with MFC, the Microsoft Foundation Classes. If you use C#, you may not use LINQ; you must actually create the queries or calls to a stored procedure. If you are writing in Java, use of JavaFX or Swing is acceptable. If writing in C#, use WinForms or, if you’re adventurous or have used it before, Windows Presentation Foundation (WPF.) Use only libraries that are part of the language; using libraries you download from the Internet for is not acceptable and will result in a failing grade, since that makes it much more difficult for the TA and me to compile and run.

If you write in C#, hand in your entire Visual Studio project. If you write in Java, hand in only your .java source files (and XML if you use a screen designer.) You will also be required to show your program to the TA to verify functionality, so test carefuily. This can be done through Teams.

This is an individual assignment.



|  |  |
| --- | --- |
| Grading Criteria | |
| Program allows add, change, delete of books | 50% |
| Program rejects invalid input | 30% |
| Program is not vulnerable to injected SQL attacks | 10% |
| Program comments meet standards | 10% |
| Total | 100% |

Additional grading notes:

1. Any crash that produces an error that the user sees: -5
2. Any of the major add, change, or delete functions not working: -10 to -15 depending upon severity
3. Program accepts invalid input and allows invalid database states: -10 to -30 depending upon severity
4. Program submits SQL strings to be executed without checking (stored procedures will eliminate this issue): -10
5. Not using your original (or modified) database: -20