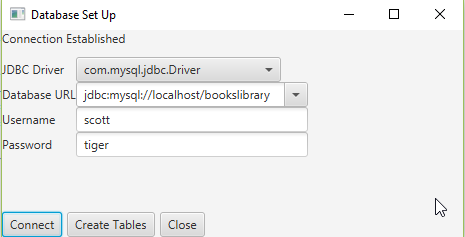
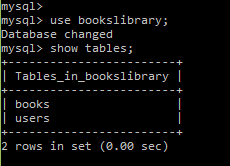
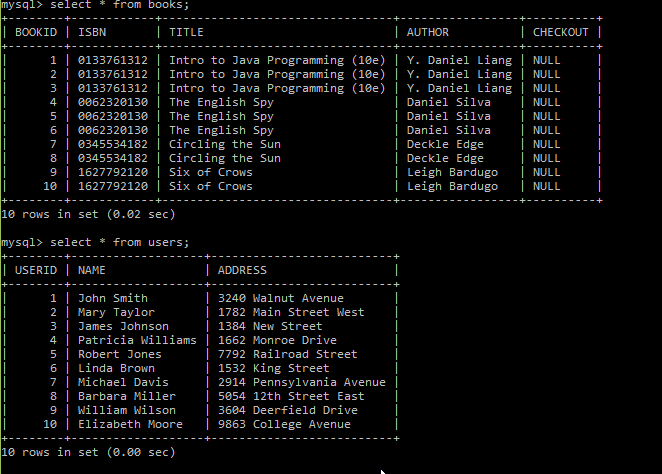
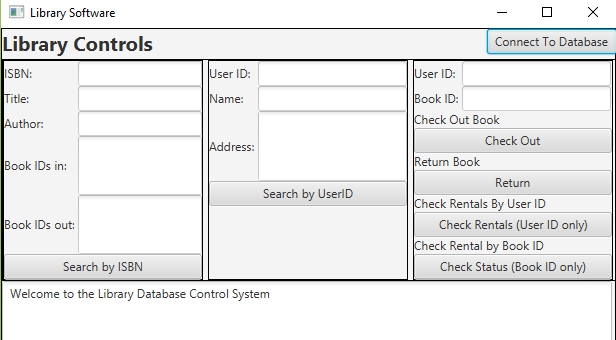
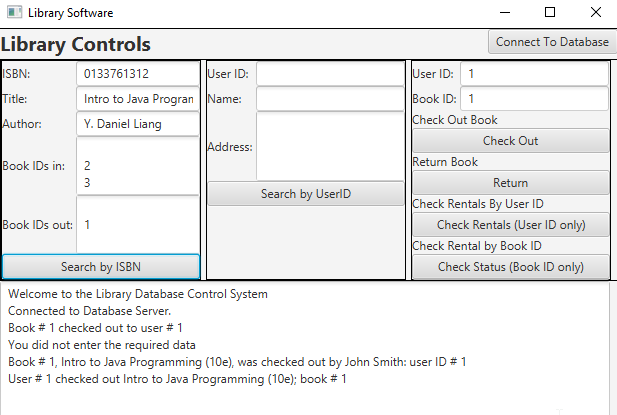
**CMSC 214: Final Exam**

Problem Description:

Design a set of database tables to store library books and patrons. A book has an ISBN (International Standard Book Number), an author, and a title. The library may have multiple copies of each book, each with a different book ID. A patron has a name, a unique ID, and an address. A book may be checked out by at most one patron, but one patron can check out multiple books.

Write a Java program that creates a library database of books and patron data as described above. Patrons should be able to check out and return books. Supply commands to print the books that a patron has checked out and to find who has checked out a particular book. You may create and populate Patron and Book tables before running the program.

1. DB setup program ….  
     
   
2. Click on connect to establish connection
3. Click on “create tables” to create tables and sample data  
     
   
4. This is how the tables structure looks like after it the setup (your java setup program needs to build this structure):  
     
   
5. Library System Software look and feel  
     
     
     
   

Submit the following:

* Your Java source code file(s)
  + Java file that does the DB tables setup with “bookslibrary” database already pre-created manually  
    
* Algorithm file(s)
* Your SQL scripts that create necessary tables in a DB
* Users Guide that outlines
  + How to use the software application that you created
  + What platform are you running your program on (operating system, Java IDE product and version, DB product and version, java JDBC version)?
* Steps to run your program
* Screenshots (lot of them)
* UML

**Deliverables:**

In this programming project the student will design, develop, test and document a Java application.

**Projects submitted with evidence of plagiarism will be given a score of 0.**

**Code Documentation and Style Requirements**The documentation requirement for all programming projects is one block comment at the top of the program containing the course name, the project number, your name, the date and platform/compiler that you used to develop the project. In addition, there should be at least one comment for each class in the program describing what that class does. Additional comments should be provided as necessary to clarify the program.

Indentation must be consistent throughout the program. Variable and method names should be descriptive of the role of the variable or method. Single letter names should be avoided. All constants, except 0 and 1, should be named. Constant names should be all upper-case. Variable names should begin in lower-case, but subsequent words should be in title case (e.g., finalSpeed).

Separate compilation must be used in accordance with standard Java practice. Every class must be saved in a separate .java file

**Rubric for Final Project**

**Grading:**

This project has three parts, with **100 total points** possible.

  (40 points) Carefully declare all the variables with appropriate data types. Use comments effectively to make the program more readable. Implement the calculations correctly.

  (40 points) Compile and build the Java program. Test the program using the Test Plan as described above and make sure the program produces the desired answer.

  (20 points) Submit your source code, screenshots, UML diagram, and algorithm as attached files (**in a single zip file**) to Final Exam Folder