

61FIT3JSD - Assignment 2

Desktop Application Development

(version 2 – 2023-11-15)

IMPORTANT: This assignment must be done individually.

Read **Section 1 & 2** to understand the requirements and **Section 3** to know how to submit your work.

1. Description

As required from the stakeholder, you will change the LibMan program from [Assignment 1](#) to create LibMan2 program with a graphical user interface using Java's Swing API. Similar to LibMan, LibMan2 still uses the `LibraryManProg` class to coordinate other components and to start the program from the command line. In this assignment, you will implement the following features:

- the application main window
- the function to manage patrons
- the function to manage books
- the function to manage transactions
- the function to manage checkout book
- the function to display an transactions report & an checkout book report & an overdue book report

2. Requirements

2.1. Data requirements

The data structure requirements for this program remain consistent with those outlined in Assignment 1. The essential data classes include `Book`, `Patron`, and `LibraryTransaction`. In this assignment, the program must persistently store data objects in an SQLite database. An SQLite database named `library.db` with sample data has been provided and you must use this database to develop the LibMan2 program. You have to put this database file under the root of your project so that the connection string looks exactly like this:

`"jdbc:sqlite:library.db"`

2.2. The menu

The program's main GUI has a menu bar for the user to use. This menu bar has four menus:

- File – This menu has one menu item:
 - Exit – This menu item terminates the program.

- Patron – This menu has two menu items:
 - New patron – This menu item opens a window for user to add a new patron.
 - List patrons – This menu item opens a window which displays a table of patrons.
- Book – This menu has two menu items:
 - New book – This menu item opens a window for user to add a new book.
 - List books – This menu item opens a window which displays a table of books.
- Transaction
 - Checkout book – This menu item will open a window for the user to check out a book.
 - Transaction report – This menu item opens a window which displays transaction report. There should be one drop-down box for selecting “All transactions”, “All checked-out books” and “Overdue books” report.
 - Return book – This menu item opens a window for user to return a book. User can select the patron and the borrowed book from drop-down boxes.

2.3. *The windows*

- 1) The application’s main window should be an empty frame with just the menu. You can put your own copyright text, user guides or your picture on this window.
- 2) The ‘New patron’ window should allow user to enter patron details through text fields (except for the patron’s id, which should be auto-generated). The “New patron” window displays these components:
 - a. Name
 - b. Dob (format: DD/MM/YYYY)
 - c. Email
 - d. Phone
 - e. Patron Type: combo-box for selecting REGULAR and PREMIUM respectively.
- 3) The ‘List patrons’ window displays a table of patrons in system.
- 4) The ‘New book’ window should allow user to enter book details through text fields (except for the book’s id, which should be auto-generated, and the book’s genre, which can be selected). The “New book” window displays these components:
 - a. Title
 - b. Author
 - c. Genre: combo-box which lets user choose a genre
 - d. Publication year
 - e. Number of Copies Available
- 5) The ‘List books’ window displays a table of all books.

- 6) The ‘Checkout book’ window should allow user to checkout book. This window displays these components:
- a. Patron: a combo-box to select a patron
 - b. Book: a combo-box to select a book
 - c. Checkout date: a label to display the system’s current date (display format: DD/MM/YYYY)
 - d. Due date: text field (format: DD/MM/YYYY)
 - e. Checkout button

A dialog window should appear when user clicks the Checkout button. This dialog window should show brief details about the transaction and let the user confirm the transaction.

Note: checkout date must default to the system’s current date.

- 7) The ‘Transaction report’ window displays transactions with these options:
- a. All transactions
 - b. All checked-out books
 - c. Overdue books

The user selects an option from a combo-box and clicks a “Get report” button to see the report displayed in a table.

- 8) The “Return book” window should allow user to return book. This window displays these components:
- a. Select patron: a combo-box to select a patron
 - b. Select book: a combo-box which is empty when the user has not selected a patron. This combo-box will contain the books borrowed by the selected patron.
 - c. Return date: a text field to enter the return date (format: DD/MM/YYYY)
 - d. A “Return” button to confirm the user’s returning of the book

2.4. Other requirements

1. Update the `LibraryManProg` class, which is the main program class, to start the program’s main GUI. Don’t forget to remove unnecessary parts in your code.
2. You are no longer restricted to the provided classes in the `common` package of Assignment 1. Move the classes in this package to your main package if you’d like to keep using them.

3. Submissions

1. You must create a top-level package named `a2_sid` for the program where `sid` is your student id.
(*) *Your submission must contain only one package. Put all necessary classes in this package.*

2. You do not have to include the database file in your submission. We will use the provided database structure (with different data) to mark your work so please do not modify the structure of the `library.db` database.
3. Store any images/pictures that you use in your program inside your source code package folder.
() This is not the recommended way in practice, but it is for the convenience of marking your assignment submission.*
4. You must submit a single `zip` file named `a2_sid.zip` containing the `a2_sid` package to LMS by the due date. For example, if your student id is `1912345678` then your zip file must be named `a2_1912345678.zip`.

IMPORTANT: failure to name the file as shown will result in reduced marks!

NO PLAGIARISM, strict penalty of marks reduction will be applied!