LAB 211 Assignment

Type:	**Long Assignment**
Code:	**J1.L.P0022**
LOC:	**500**
Slot(s):	**N/A**

Title

Library Management System.

Background

FPT University's library seeks to develop an application to manage books, documents, and borrowing/return activities. This application will automate processes and enhance user experience.

Program Specifications

Build a library management program. With the following basic functions:

1. Build Your Data Structure: (Books.dat, Users.dat, Loans.dat - binary files)
    - Book Information in the file, named as Books.dat: Book ID, book title, author, publication year, publisher, ISBN, and active_book (boolean field).
    - User Information in the file, named as Users.dat: User ID, name, date of birth, phone number, email, and active_user (boolean field).
    - Borrow/Return Information in the file named as loans.dat: Transaction ID, Book ID, User ID, borrow date, return date.
    - **The Map data structure must be used to represent the list.**
2. **Manage Books**

    2.1. ***Add a Book***

    - Create a submenu to add a new book.
    - Check for duplicate Book ID.
    - Add the new book to the collection. When a new book is added, the 'active_book' field is set to 'true' by default.
    - Ask the librarian if they want to continue adding books or return to the main menu.

    2.2. ***Update Book Information***

    - Require entry of the Book ID.
    - Display an error if the Book ID does not exist.
    - If no new information is entered, the old information will remain unchanged.
    - Validate new information.

    2.3. ***Delete a Book***

    - Display a confirmation message before deletion. When the librarian agrees to delete, only update the field active_book = false
    - Show the deletion result (Only show books with active_book = true)

2.4. ***Show All Books***

- Display all data in the book collection.

3. **Manage Users**

3.1. ***Add a User***

- Create a submenu to add a new user.
- Check for duplicate User ID.
- Add the new user to the collection. When a new user is added, the 'active_user' field is set to 'true' by default.
- Ask if the librarian wants to continue adding more users or return to the main menu.

3.2. ***Update User Information***

- Require entry of the User ID.
- Display an error if the User ID does not exist.
- If no new information is entered, the old information will remain unchanged.
- Validate new information.

3.3. ***Delete a User***

- Display a confirmation message before deletion. When the librarian agrees to delete, only update the field active_user = false
- Show the deletion result (Only show users with active_user == true)

4. **Managing Borrowing and Returning Books (Managing Loans)**

- Allow users to borrow books.
- Update borrowing information.
- Display all currently borrowed books.

5. **Reporting**

- Report on books currently borrowed.
- Report on books overdue.
- Report all borrowing activities.

6. **Store Data to Files**

- Save the book list to a file named Books.dat.
- Save user information to a file named Users.dat.
- Save borrowing and returning information to a file named Loans.dat.
- Reload data from these files when the program starts.

7. **Exit**

- Allow the librarian to exit the application, ensuring that all data is stored safely before closing the application.

Each menu choice should invoke an appropriate function to perform the selected menu item. The program must display the menu after each task and wait for the librarian to select another option until the librarian chooses to quit the program.

Features:

Function 1: Build Data Structure (50 LOC)

- ***Class and Interface Design***: Develop classes (Book, User, Loan) and an interface (Manageable) to define basic methods (add, update, delete). Apply abstraction and polymorphism for easy reuse and extension.
- ***Data Initialization and Linking***: Create objects and link them with appropriate properties (for example, each Loan is linked to a User and a Book).

Function 2: Manage Books (100 LOC)

- ***Add a New Book (25 LOC):*** Create a submenu, check for duplicate Book IDs, add the book to the collection, and inquire if the librarian wishes to continue adding books or return to the main menu. When a new book is added, the 'active_book' field is set to 'true' by default.
- ***Update Book Information (25 LOC):*** Search for and update book information based on the Book ID, validate new information.
- ***Delete a Book (25 LOC):*** Display a confirmation message before deletion. When the librarian agrees to delete, only update the field active_book = false. Show the deletion result, only show books with active_book == true).
- ***Show All Books (25 LOC):*** List and display information for all books in the library, sorted by Book ID in ascending order.

Function 3: Manage Users (100 LOC)

- ***Add a New User (25 LOC):*** Create a submenu, check for duplicate User IDs, add the user to the database, and ask if the librarian wishes to continue adding more users or return to the main menu. When a new user is added, the 'active_user' field is set to 'true' by default.
- ***Update User Information (25 LOC):*** Search for and update user information based on the User ID, validate new information.
- ***Delete a User (25 LOC):*** Display a confirmation message before deletion. When the librarian agrees to delete, only update the field active_user = false. Show the deletion result, only show users with active_user == true).
- ***Display User List (25 LOC):*** List and display detailed information for all users, sorted by User ID in ascending order.

Function 4: Managing Borrowing and Returning Books (Manage Loans) (120 LOC)

- ***Borrow Books (40 LOC):*** Handle the book borrowing process including user authentication and book availability.
- ***Update Loan Information (40 LOC):*** Allow updates to the return date and other relevant loan transaction details.
- ***Display All Borrowed Books (40 LOC):*** Report and display a list of all currently borrowed books, including borrow and expected return dates.

Function 5: Reporting (70 LOC)

- ***Report on Borrowed Books (25 LOC):*** Generate a report listing all currently borrowed books.
- ***Report on Overdue Books (25 LOC):*** List books that are overdue.
- ***Report on Total Borrowing Activities (20 LOC):*** Compile and report all book borrowing activities over a specific period.

Function 6: Store Data to Files (50 LOC)

- ***Data Storage:*** Save lists of books, users, and borrowing transactions to corresponding files (Books.dat, Users.dat, Loans.dat).
- ***Data Reloading:*** Reload data from these files when the program starts to ensure data is up-to-date.

Function 7: Exit Program (10 LOC)

- ***Close Application:*** Allow users to exit the application, ensuring all data is safely stored before closing.

- The above specifications are only basic information; you must perform a requirements analysis step and build the application according to real requirements.
- The lecturer will explain the requirement only once in the first slot of the assignment.