Library Management System

In this assignment you have to create a simple library management system for **a** **librarian**. The librarian should be able to search for books by title to check their availability, check out available books, and return any books the members currently have. Details of all books in the library should be stored in an external text file. For each book the following information should be stored: ISBN, title, author, current loan status (is it available, and if not, who has it?), purchase date, and a unique ID number which can be used to identify different copies of the same book.

**Member**

Members should be identified using their unique ID-numbers. For simplicity it is suggested that you use 4-digit integers (e.g.: 1000-9999) for these IDs, and you may assume that all 4-digit numbers are valid members. (Note however that your program should be able to distinguish between valid and invalid IDs, so that e.g. hh9# would not be accepted. You do not store any further member information.

**Book**

Your text file storing the information on each book should contain all the data mentioned in the overview section. You must organise the file as follows:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ID | ISBN | Title | Author | Purchase Date | Member Id |
| 1 | 9783161484100 | Book\_1 | Author\_1 | 1/8/2010 | 1023 |
| 2 | 9783161484105 | Book\_2 | Author\_2 | 1/8/2014 | 0 |
| 3 | 9783161484100 | Book\_1 | Author\_1 | 1/8/2010 | 0 |
| **…** | **…** | **…** | **…** | **…** | **…** |
| **…** | **…** | **…** | **…** | **…** | **…** |
| **…** | **…** | **…** | **…** | **…** | **…** |
| n | 9783161484199 | Book\_n | Author\_n | 1/8/2014 | 0 |

where each line provides all the information for a particular book, with the book ID on the left, and the status on the right given either by a 4-digit member ID (meaning that member currently has the book), or a value indicating that the book is currently available (in this case 0).

**Searching for Books**

Your program should include functionality to search for a book based on its title. Given a search term (e.g. Book\_1), your program should return a complete list of books with that title and all their associated information.

**Checking Out Books**

In order to withdraw (or check-out) a book from the library, the librarian should provide a member-ID and the book's ID number (note it must be ID, not book title, since there could be more than one copy of the same book). Your program should then

* Check that the input is valid (and return an error message if it is not),
* return an error message if the book is not available due to being on loan to

someone else,

* if the book is available, allow the librarian to withdraw the book by updating the book's status in text file accordingly.

**Returning Books**

The librarian should be able to return books simply by providing the book's ID number. If the ID is invalid, or the book is already available, the program should return an error message. Otherwise, the text file should be updated accordingly.

Structure of program

The following structure is needed

**database.txt** Stores all the data (see previous section).

**logfile.txt** Stores loan history of library books (i.e., book\_id, Checkout Date, Return Date). You need to populate the files with the realistic data (minimum 10 records).

**booksearch.****java**: **A Java Class which contains functions used to allows librarian to input search terms as strings, and returns the output as described in the previous section. You can either: load the contents of database.txt into a list, and store this as a global variable for repeated use, or load the contents of database.txt each time a search is performed**.

**bookcheckout.java**: **A Java Class which contains functions used to ask librarian for borrower’s member-ID and the ID of the book(s) they wish to withdraw. Then, after performing the validity checks and functionality described in the previous section, should return a message letting the librarian know whether they have withdrawn the book successfully.**

**bookreturn.java**: **A Java Class which contains functions used to ask the librarian for the ID of the book(s) they wish to return and provide either an appropriate error message, or a message letting them know they have returned the book(s) successfully.**

**database.java**: **A Java Class which contains common functions that the book search, checkout and return modules use to interact with the database and log files.**

**menu.java: A Java main program which provides the required menu options to the librarian for the program functionalities. The menu could be based on simple console or you may use JOptionPane if you know about this. This is optional to you**.

Submission:

1. You must submit an Eclipse project and project name must be your ID.
2. Clearly follow the instructions, because my system will auto check your assignments, so follow the instructions exactly the way they are.