

Coursework Assignment Specification
(70 % of the module assessment)
Submission Deadline: at 14:00 on 15th December 2022

Library Management System

1 Problem Description

1.1 Overview

Your task is to write a simple library management system for a librarian. The librarian should be able to check available books out and return any books they currently have. Details of all books in the library should be stored in a database (i.e., text files).

1.2 Book_Info.txt

For each book the following information should be stored: title, author, genre, purchase price, purchase date, a unique ID number which can be used to identify different copies of the same book.

See example data below:

ID	Genre	Title	Author	Purchase Price £	Purchase Date
1	Sci-Fi	Book_1	Author_1	10	1/8/2010
2	Fantasy	Book_2	Author_2	20	1/8/2014
3	Sci-Fi	Book_1	Author_1	10	1/8/2010
...
...
n	Classics	Book_n	Author_n	40	1/8/2014

Each line provides all the information for a particular book. Assume that the purchase price does not change over the years. Note that you are not allowed to add additional columns to the file.

1.3 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.

1.4 Searching for Books

Your program should include functionality to search for a book based on its title. Given a search term (e.g.: “Treasure Island”), your program should return a complete list of books with all their associated information (e.g., title, author, genre and loan availability, etc.).

1.5 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 the ID, not the book title, since there could be more than one copy of the same book). Your program should then

1. check that the input is valid (and return an error message if it is not),
2. return an appropriate message and **a book reservation option** if the book is not available due to being on loan,
3. if the book is available, allow the librarian to withdraw the book by updating the related records in the database accordingly.

1.6 Returning Books

The librarian should be able to return books simply by providing the book's IDs. If the ID is invalid, or the book is already available, the program should return an error message. Otherwise, the database should be updated accordingly. Additionally, an appropriate message should be displayed if the book is reserved by a member.

1.7 Selecting Books for Purchase Order

The library has a budget to purchase books every year. Your program must have functionality to find out popular book titles to purchase more copies if it is necessary, and recommend genres to purchase new book titles. The librarian should provide the budget so that the system is able to generate the recommendation list. The program should suggest how many book copies the library would purchase for each genre in the list based on the given budget.

The database should have a transaction log, which keeps the loan and reservation history of library books. Based on this log, the system can suggest the purchase list.

2 How to Structure Your Program

The following structure is needed as a final submission for your project:

Data Files: Stores all the required data.

Book_Info.txt Stores all the data (see previous section). You need to populate the file with the realistic data (minimum 20 records).

logfile.txt Stores loan and reservation history of library books (i.e., Book_id, Member id, Checkout Date and Return Date and reservation information etc.). You need to populate the file with the realistic data (minimum 60 records).

bookSearch.py: A Python module which contains functions used to allow librarian to input search terms as strings, and returns the output as described in the previous section.

bookCheckout.py: A Python module which contains functions used to ask the librarian for borrower's member-ID and the ID of the book(s) to be checked out. Then, after performing the validity checks and functionality described in the previous section, the program should return a message letting the librarian know whether they have checked out the book(s) successfully.

bookReturn.py: A Python module 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.

bookSelect.py: A Python module which contains functions used to list the recommended book genres for the librarian and appropriately visualise the list by using the Matplotlib module. You should come up with the details of the selection criteria.

database.py: A Python module which contains common functions that the book search, checkout, return and select modules use to interact with the data files.

menu.py: A python main program which provides the required menu options to the librarian for the program functionalities. The menu could be based on Python Graphical User Interface-GUI (namely the tkinter python module). In that case, the GUI must use only one window.

3 What to Submit

In addition to the files mentioned in Section 2 you may want to write a short text file called README (max 500 words). This is to provide any special instructions or warnings to the user (or assessor!), or to draw attention to any aspects of the program that you are particularly proud of (please don't waste time by writing an excessive amount).

All the files (including sample data files) should be compressed into a zip file and submitted electronically as directed.

4 Notes on Expectations:

You will be marked according to your overall achievement, marked according to the Assessment Matrix that will be provided to you separately from this document. However below follows a qualitative description of some general expectation that may help you understand the general level of expectation associated with this piece of coursework.

Technical mastery of Python Your programs should show mastery of what you have been taught.

Design Your programs should be well structured for the task in hand so that it is as easy as possible for:

- a user to use the program for any likely purpose,
- a programmer to understand the code structure and be able to develop it further,
- a programmer to be able to re-use as much as possible of the code in a related application.

Clarity and Self-Documentation Given the structure of your programs, they should be as easy to read and understand as possible. *Lay your code out* so that it can be listed sensibly on a variety of devices: avoid having any lines longer than 80 characters as these may wrap (to reduce the number of “problem lines” you should use 4 spaces for indentation rather than tabs). *Sensible names* should be chosen for all variables, methods etc. *Documentation strings* should be included for each:

Program Fully explain what the program does and how it should be used. Also state who (ID only) wrote it and when.

Function State what each function does and explain the roles of its parameters.

In addition, you should include occasional comments in your code; these may be (a) to introduce a new section in the code, or (b) to explain something that is not obvious. Bear in mind that pointless comments make your code harder to read, not easier.

Restriction

- 1) Your code must **NOT** include any Class type.
- 2) Your code must **NOT** have any SQL statements.
- 3) Your code must **NOT** have any nested function declaration.
- 4) You must use Python v3.8 or above.
- 5) Your code must use **ONLY** standard python libraries and Matplotlib.