Software Development Plan

Library Management System (LMS)

09/03/2023

Carlos Lopez Padilla

Software Development I CEN 3024C

CRN: 14835

Developer Requirements

The Library Management System (LMS) will be a tool that will allow faculty, specifically library staff, to manage the collection of books in the library. The faculty will have the ability to add and remove books, as well as list/display all books currently in the collection.

User Requirements

The library staff have expressed their needs for the LMS tool as follows:

- The program must run on the console.
- To add a book(s) to the collection, the program must read a text file, that will list each book entry in a new line. Each line will be formatted with a unique book ID, the tile, and the author's name, separated by a comma ("461098, It Ends With US, Colleen Hoover").
- To remove a book from the collection, the library staff must enter the book's ID in the console. The program will proceed to find the book and delete it.
- The list functionality must display all items currently in the collection.

Implementation Plan

The program will be implemented as a console-based Java program. When first run, the program will greet the user and present it with the main menu – "Add books", "Remove a book", "List collection", and "Exit". Each of these functionalities will have its method, *addBooks*, *removeBook*, and *listCollection* respectively.

addBooks – This method will scan a .txt file containing the details of the book(s) to be added to the collection and insert each entry into the collection utilizing the merge sort algorithm. The contents of said file will be formatted as described previously in the user requirements.

removeBook – This method will prompt the user to type the book ID of the item to be removed and will search for the item using the merge sort algorithm. Once found, the book will be deleted from the collection.

listCollection – This method will simply display to the console all books in the collection.

Once any of these functionalities are performed the program will return the user to the main menu for further actions.

Because this project will be completed following the spiral model, this plan is subject to change as further progress on development and testing is made.

Testing Plan

Each functionality will be tested thoroughly and incrementally as development progresses to guarantee an error-free experience. Test cases varying in size and format will be used to account for invalid inputs.