Faculty of Computers and Data Science
Alexandria University
2024 / 2025 Programming |

Programming 1 Final Project

Library Management System (LMS)

The Library Management System (LMS) allows users to manage a library's books by performing various operations such as adding, searching, issuing, returning, deleting, and editing books.

It supports the management of book titles and descriptions, and tracking their availability.

The program uses arrays to store the book details and their statuses.

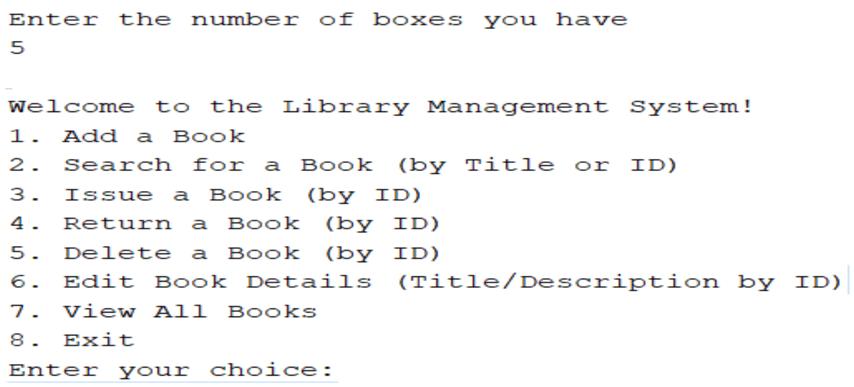
Project Outlines:

1. **Add a Book:** Add new books by providing a title and description.
2. **Search for a Book:** Search for books by title or ID, and view details such as title, description, and availability.
3. **Issue a Book:** Mark a book as issued by providing its ID, making it unavailable for borrowing until returned.
4. **Return a Book:** Mark a book as available again by providing its ID, indicating it has been returned.
5. **Delete a Book:** Delete a book from the system by providing its ID. This removes the book from the list and shifts the remaining books.
6. **Edit Book Details:** Edit the title or description of a book by its ID.
7. **View All Books:** List all books with their details, including title, description, and availability status.
8. **Exit:** Exit the system.

How It Works:

The program runs in a continuous loop where the user can select various options to manage books in the library.

It asks for the number of books that can be stored and provides a menu with choices, based on the user's choice the system performs the following actions :



```
Enter the number of boxes you have
5

Welcome to the Library Management System!
1. Add a Book
2. Search for a Book (by Title or ID)
3. Issue a Book (by ID)
4. Return a Book (by ID)
5. Delete a Book (by ID)
6. Edit Book Details (Title/Description by ID)
7. View All Books
8. Exit
Enter your choice:
```

Main Menu and Choices:

Once the program is running, it shows a menu with the following options:

1. **Add a Book:** Prompts the user for the book's title and description, and adds the book to the library if space is available.

2. **Search for a Book:** Allows the user to search by either title or ID. It displays the book's details including status (Issued or Available).

3. **Issue a Book:** Prompts the user for the book ID. If the book is available, it marks it as issued (unavailable or already borrowed).

   the term **"issue"** refers to the process of **loaning or borrowing** a book or resource to a user for a specific period. when a book is issued, it is **temporarily removed** from the library's available inventory and provided to the user until they return it.

4. **Return a Book:** Prompts for a book ID to return it (make it available to be borrowed).

5. **Delete a Book:** Prompts for a book ID and deletes the book from the library, shifting the remaining books to fill the gap.

6. **Edit Book Details:** Allows the user to update the book's title or description by its ID.

7. **View All Books:** Displays a list of all books in the library with their title, description, and availability.
8. **Exit:** Exits the system.

Set Up the Library System:

The program initializes arrays:

- bookTitles[]: Stores the titles of the books.
- bookDescriptions[]: Stores the descriptions of the books.
- bookIssued []: Tracks the availability of each book (false = available, true = issued).
- bookCount: Tracks the number of books currently added.

Detailed Breakdown of How to Implement Each Feature:

1. Add a Book:

- When the user selects "Add a Book" the program asks for the book's title and description.
- If the library is not full, the book is added to the bookTitles[], bookDescriptions[], and bookStatuses[] arrays, and bookCount is incremented.
- If the title is empty or null, an error message is shown.

2. Search for a Book:

- The user can search by either book title or ID:
  - **By Title:** The program loops through bookTitles[] and checks for matches.
  - **By ID:** The program checks if the provided ID is valid and displays the corresponding book details.

3. Issue a Book:

- The user provides the book's ID.
- If the book is available (bookStatuses[issueBook_Id - 1] == false), the book is marked as issued (bookStatuses[issueBook_Id - 1] ==true).

4. Return a Book:

- The user provides the book's ID to return it.
- If the book is issued, its status is changed back to available (bookStatuses[returnBook_Id - 1] = false).

5. Delete a Book:

- The user provides a book ID to delete.
- The program shifts the remaining books in the arrays to fill the gap left by the deleted book.
- The book is removed by setting its entry to null in bookTitles[] and bookDescriptions[], and resetting its status to false in bookStatuses[].

6. Edit Book Details:

- The user selects whether to edit the book's title or description.
- The program checks for valid input and updates the selected book's information.

7. View All Books:

- The program loops through the bookTitles[] array and displays each book's title, description, and availability status.

8. Exit:

- Exits the program and closes the Scanner.

Error Handling:

- **Invalid Inputs:** The program checks for invalid or empty inputs in book titles and descriptions.
- **Book ID Validation:** The program ensures that book IDs entered for actions like issuing, returning, or deleting are valid.