Project Report

Project: Simple Library Management System

Used Programming Language: Java

Name: Venkata Jayanth

Registration No: 12205857

Gamil: venkatajayanth.tummalapalli@gmail.com

Index:

Introduction

A Library Management System (LMS) is a software application designed to automate and manage the operations of a library. It facilitates efficient management of library resources, including books, magazines, journals, and other materials, along with patron information and transactions. Here's an introduction to key aspects and benefits of an LMS:

□ **Book Class:** Represents a book with attributes like ID, title, author, and availability status.

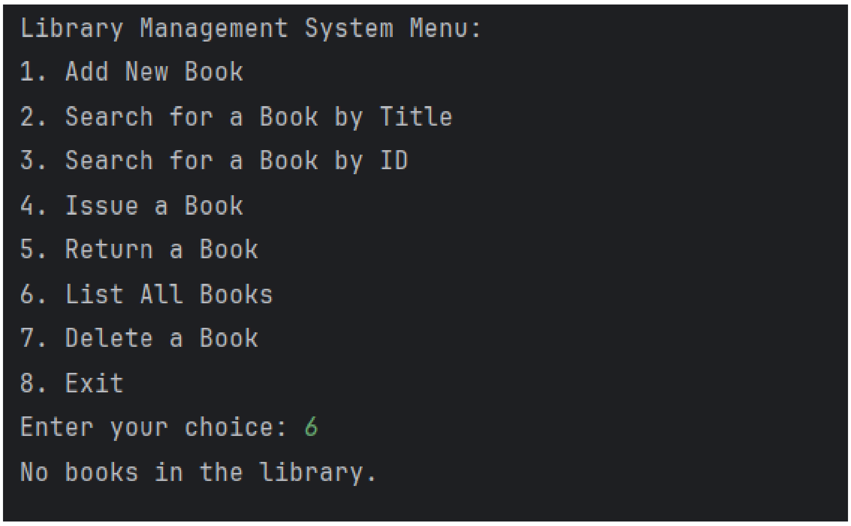
□ **Library Class:** Manages a collection of books and provides methods to add, search, issue, return, list, and delete books.

□ **Main Class:** Implements a simple console-based menu-driven interface to interact with the library system. It utilizes a Scanner for user input and delegates operations to the Library class based on user choices.

□ **Functionality:** This system allows adding new books, searching by title or ID, issuing and returning books (updating availability), listing all books, and deleting books from the library.

□ **Error Handling:** Basic error handling is included (e.g., book not found). You may enhance this based on specific requirements.

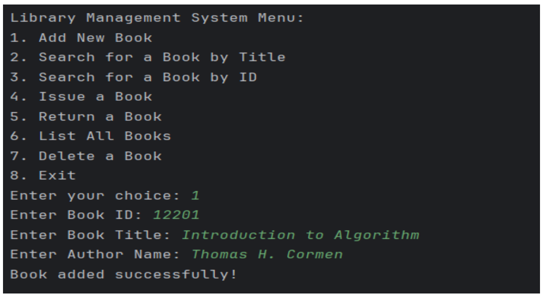
□ **Data Storage:** Books are stored in-memory (ArrayList). For a real-world application, consider using a database for persistent storage.



```
Library Management System Menu:
1. Add New Book
2. Search for a Book by Title
3. Search for a Book by ID
4. Issue a Book
5. Return a Book
6. List All Books
7. Delete a Book
8. Exit
Enter your choice: 6
No books in the library.
```

Explanation of Adding a New Book Functionality

1. **User Input:**
    - When the user chooses the option to add a new book from the menu, the program prompts them to enter the following details:
        - **Book ID:** A unique identifier for each book, which typically distinguishes it from other books in the library.
        - **Title:** The title of the book, which identifies the book's name or subject matter.
        - **Author:** The author's name, indicating who wrote or compiled the book.
2. **Validation:**
    - Before proceeding to add the book to the library collection, the system should validate the input:
        - **Unique Book ID:** Ensure that the book ID entered by the user is unique and not already assigned to another book in the library. This prevents duplication and maintains data integrity.
        - **Non-Empty Fields:** Validate that all fields (book ID, title, author) are filled out by the user to avoid adding incomplete or erroneous data.
3. **Book Creation and Storage:**
    - Once validated, the program creates a new Book object using the provided details (bookId, title, author).
    - The Book object is then added to a data structure that manages the library's collection of books. In the provided example, an ArrayList<Book> is used for its flexibility in managing a dynamic list of books.
4. **Feedback to User:**
    - After successfully adding the book, the system should provide feedback to the user confirming the addition. This ensures transparency and informs the user that their action was successful.
    - If any validation fails (e.g., duplicate book ID), appropriate error messages should be displayed to guide the user on how to correct their input.



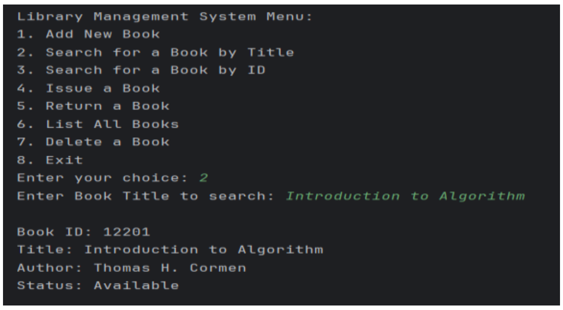
```
Library Management System Menu:
1. Add New Book
2. Search for a Book by Title
3. Search for a Book by ID
4. Issue a Book
5. Return a Book
6. List All Books
7. Delete a Book
8. Exit
Enter your choice: 1
Enter Book ID: 12201
Enter Book Title: Introduction to Algorithm
Enter Author Name: Thomas H. Cormen
Book added successfully!
```

Explanation of Searching for a Book Functionality

1. **User Input:**

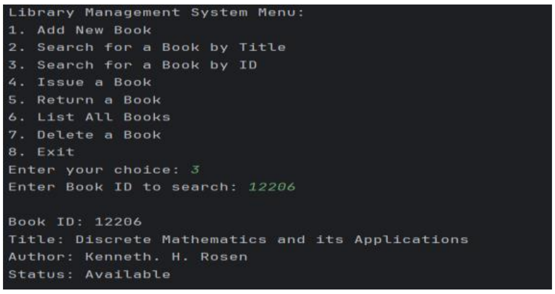
- When the user chooses the option to search for a book by title or ID from the menu, the program prompts them to enter either:
  - **Book Title:** The title of the book they want to search for.
  - **Book ID:** The unique identifier of the book they want to search for.

2. **Search Implementation:**
   - The program then iterates through the list of books in the library.
   - For searching by **Title**:
     - It compares the entered title (case-insensitive) with the titles of all books in the library.
     - If a match is found, it displays detailed information about the book (e.g., ID, title, author, availability).
   - For searching by **ID**:
     - It compares the entered ID (case-insensitive) with the IDs of all books in the library.
     - If a match is found, it displays detailed information about the book.
3. **Feedback to User:**
   - If the book is found based on the search criteria (title or ID), the system displays its detailed information.
   - If no matching book is found, the system notifies the user that the book is not found.
4. **Edge Cases:**
   - Handle scenarios where the entered title or ID does not match any existing books in the library.
   - Provide clear and informative messages to guide the user in case of input errors or unexpected behavior.



```
Library Management System Menu:
1. Add New Book
2. Search for a Book by Title
3. Search for a Book by ID
4. Issue a Book
5. Return a Book
6. List All Books
7. Delete a Book
8. Exit
Enter your choice: 2
Enter Book Title to search: Introduction to Algorithm

Book ID: 12201
Title: Introduction to Algorithm
Author: Thomas H. Cormen
Status: Available
```



```
Library Management System Menu:
1. Add New Book
2. Search for a Book by Title
3. Search for a Book by ID
4. Issue a Book
5. Return a Book
6. List All Books
7. Delete a Book
8. Exit
Enter your choice: 3
Enter Book ID to search: 12206

Book ID: 12206
Title: Discrete Mathematics and its Applications
Author: Kenneth. H. Rosen
Status: Available
```

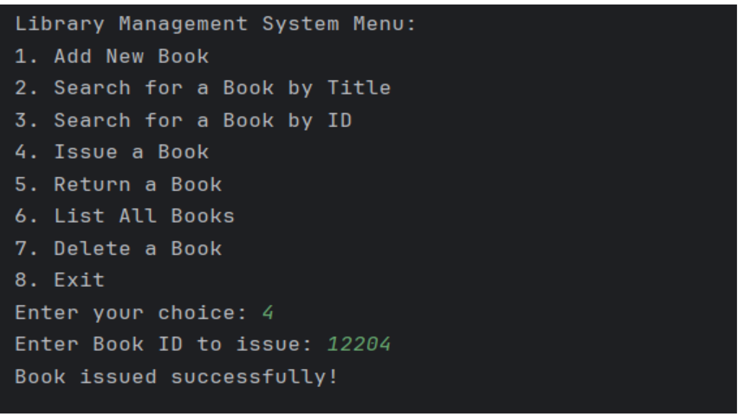
Explanation of Issuing and Returning a Book Functionality

1. **Issuing a Book:**

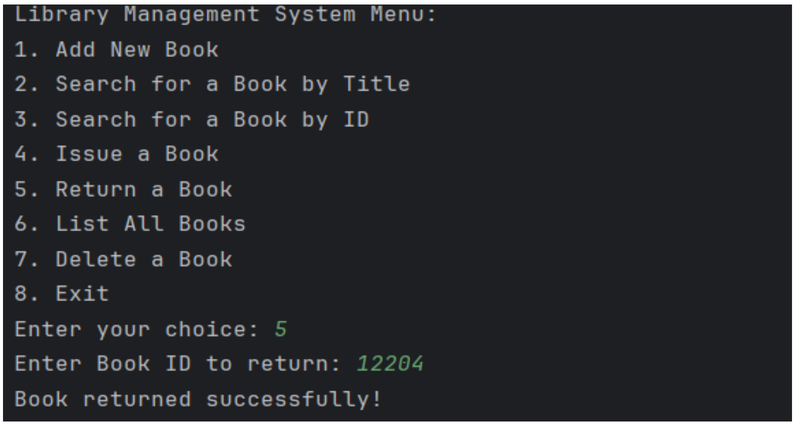
   - **User Input:** When the user selects the option to issue a book from the menu, they are prompted to enter the unique bookId of the book they want to issue.
   - **Search for Book:** The system searches for the book in the library's collection using the bookId.
   - **Availability Check:** If the book is found and is currently marked as available (isAvailable is true), the system marks the book as issued (isAvailable set to false).
   - **Feedback:** Provide feedback to the user confirming whether the book was successfully issued or if it was already issued.

2. **Returning a Book:**
   - **User Input:** When the user selects the option to return a book from the menu, they are prompted to enter the unique bookId of the book they want to return.
   - **Search for Book:** The system searches for the book in the library's collection using the bookId.
   - **Availability Check:** If the book is found and is currently marked as issued (isAvailable is false), the system marks the book as returned (isAvailable set to true).
   - **Feedback:** Provide feedback to the user confirming whether the book was successfully returned or if it was already available.

3. **Edge Cases:**
   - Handle scenarios where the entered bookId does not match any existing books in the library.
   - Provide clear and informative messages to guide the user in case of input errors or unexpected behavior.



```
Library Management System Menu:
1. Add New Book
2. Search for a Book by Title
3. Search for a Book by ID
4. Issue a Book
5. Return a Book
6. List All Books
7. Delete a Book
8. Exit
Enter your choice: 4
Enter Book ID to issue: 12204
Book issued successfully!
```



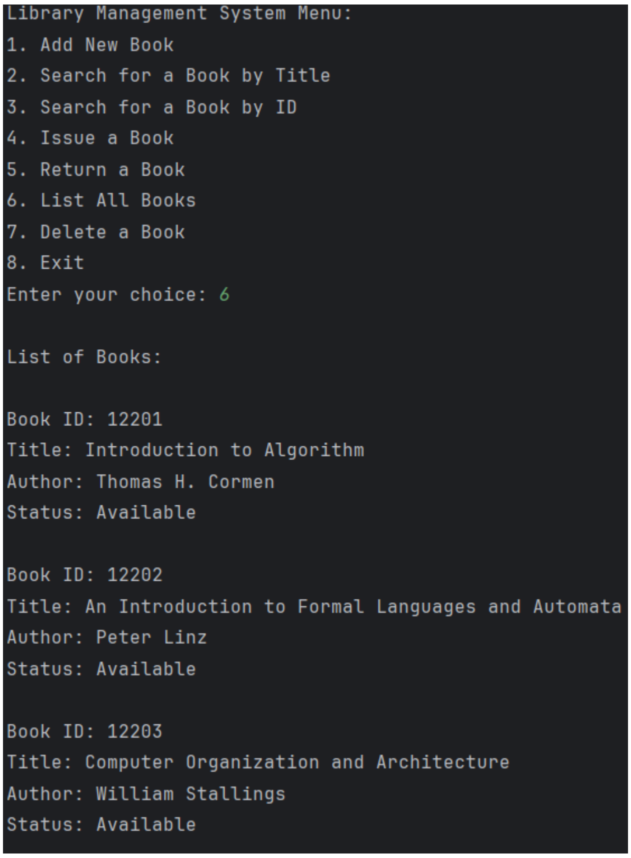
```
Library Management System Menu:
1. Add New Book
2. Search for a Book by Title
3. Search for a Book by ID
4. Issue a Book
5. Return a Book
6. List All Books
7. Delete a Book
8. Exit
Enter your choice: 5
Enter Book ID to return: 12204
Book returned successfully!
```

Explanation of Listing All Books Functionality

1. **Display All Books:**

   - **User Request:** When the user selects the option to list all books from the menu, the system iterates through the library's collection of books.
   - **Detailed Information:** For each book, the system displays:
      - **Book ID:** The unique identifier of the book.
      - **Title:** The title of the book.
      - **Author:** The author of the book.
      - **Availability Status:** Indicates whether the book is available (not issued) or issued.
   - **Formatting:** Display the information in a structured format to ensure readability and clarity.

2. **Edge Cases:**
   - Handle scenarios where the library's collection is empty, and no books need to be displayed.
   - Provide clear and informative messages to guide the user in case of unexpected behavior or errors.



```
Library Management System Menu:
1. Add New Book
2. Search for a Book by Title
3. Search for a Book by ID
4. Issue a Book
5. Return a Book
6. List All Books
7. Delete a Book
8. Exit
Enter your choice: 6

List of Books:

Book ID: 12201
Title: Introduction to Algorithm
Author: Thomas H. Cormen
Status: Available

Book ID: 12202
Title: An Introduction to Formal Languages and Automata
Author: Peter Linz
Status: Available

Book ID: 12203
Title: Computer Organization and Architecture
Author: William Stallings
Status: Available
```

Detailed Description of the Library Management System (LMS) Code

Overview:

This Java program simulates a basic Library Management System (LMS) with functionalities to add, search, issue, return, list, and delete books. It utilizes object-oriented principles and basic data structures to manage library operations efficiently.

Classes:

1. **Book Class:**
   - **Attributes:**
     - bookId: Unique identifier for each book.
     - title: Title of the book.
     - author: Author of the book.
     - isAvailable: Status indicating whether the book is available for borrowing (true) or already issued (false).
   - **Constructor and Methods:**
     - Book(String bookId, String title, String author): Initializes a new Book object with the provided details.
     - displayBook(): Displays detailed information about the book including ID, title, author, and availability status.
     - Getters and setters for accessing and modifying the book attributes.
2. **Library Class:**
   - **Attributes:**
     - books: A list (ArrayList) to store instances of Book representing the library's collection.
   - **Constructor and Methods:**
     - Library(): Initializes an empty list of books (ArrayList).
     - addBook(String bookId, String title, String author): Adds a new book to the library collection.
     - searchByTitle(String title): Searches for a book by its title and displays its details if found.
     - searchById(String bookId): Searches for a book by its unique ID and displays its details if found.
     - issueBook(String bookId): Marks a book as issued (changes isAvailable to false) if it is currently available.

        - returnBook(String bookId): Marks a book as returned (changes isAvailable to true) if it is currently issued.
        - displayAllBooks(): Displays details of all books in the library.
        - deleteBook(String bookId): Deletes a book from the library collection based on its ID.

3. **Main Class:**
    - **Attributes:**
        - library: An instance of the Library class to manage library operations.
        - scanner: A Scanner object to read user input from the console.
    - **Methods:**
        - main(String[] args): The main method that serves as the entry point of the program.
        - Displays a menu-driven interface allowing users to perform various operations like adding, searching, issuing, returning, listing, and deleting books.

Features Implemented:

1. **Adding a New Book:**
    - Users can input a unique book ID, title, and author to add a new book to the library collection.
2. **Searching for a Book:**
    - Users can search for a book either by its title or unique ID. The system displays detailed information if the book is found; otherwise, it notifies the user that the book is not found.
3. **Issuing and Returning a Book:**
    - Books can be marked as issued or returned. The system manages the availability status (isAvailable) accordingly.
4. **Listing All Books:**
    - Displays a comprehensive list of all books currently stored in the library, including their ID, title, author, and availability status.
5. **Deleting a Book:**
    - Allows the removal of a book from the library collection based on its unique ID.

Data Structures and Algorithms Concepts Used:

- **ArrayList:** Used to store and manage the collection of Book objects efficiently. It provides dynamic resizing and easy access to elements.
- **Iteration:** Iterates through the list of books using enhanced for-loop and Iterator to perform operations like searching and deleting books.
- **Basic Search Operations:** Utilizes linear search to find books based on title or ID. Although not explicitly mentioned in the code, binary search could be implemented for more efficient searching if the list of books were sorted.
- **Object-Oriented Programming (OOP) Concepts:** Encapsulation (private attributes with public getters/setters), Constructor for initialization, and methods for operations on Book and Library objects.

Reference

1.Geeks for Geeks

2.Introduction to Algorithm   Thomas H. Cormen