Library Management System

Submitted To: Amarinder Kaur, 21482

Submitted By:

Student 1 & Registration Number: Rahul Sutradhar, 12200111 ,(RK21ELA14)

Student 2 & Registration Number: Md Mazharul Haque,12211223, (RK21ELA22)

Student 3 & Registration Number: Anjali Kumari, 12212408, (RK21ELA23)

Subject: CSE210

Course: B. Tech C.S.E.

Section: K21EL

Abstract:

This project is a library management system that allows the user to add, delete, search, update, rent, and return books. The program is written in Java and uses a HashMap to store the book catalog and an ArrayList to store rented books. The user can add a book to the catalog by entering the ISBN, title, author, and year. The user can also delete a book by ISBN, search for a book by ISBN, update a book by ISBN, and display all available books. To rent a book, the user enters the ISBN, and if the book is available, the book is marked as rented, and the user is charged a fine if the book is returned late. The program provides a menu-based interface for the user to interact with the system. The book is represented by a Book class that has fields for ISBN, title, author, year, and rented status.

Introduction:

The Library Management System presented here is a program designed to manage a library's book inventory. It was created by Mazharul, Rahul, and Anjali and uses the Java programming language. The system utilizes several data structures, including a HashMap to store the book catalogue and an Array List to keep track of rented books. The program provides several features, including the ability to add, delete, search, and update books, as well as rent and return them. The system also has a display function that shows all available books.

The system's main function is implemented using a while loop that continuously prompts the user for options using a Scanner object. The program provides seven options: add book, delete book, search book, update book, rent book, return book, and display available books. Each option is associated with a specific method that performs the corresponding operation. The Book class is also implemented to create book objects with several attributes, including ISBN, title, author, year, and rented status.

The Library Management System presented here is a simple yet effective program that can be used by small libraries to manage their book inventory. It provides basic features that can be extended and customized according to the specific needs of a library.

What was used:

The Java programming language was used to write the code for the Library Management System. In addition to the core Java libraries, the code makes use of the following specific Java classes and data structures:

- ArrayList: A class in the Java Collections Framework that provides a resizable array implementation, which can be used to store and manipulate a collection of objects.
- HashMap: A class in the Java Collections Framework that provides a hash table implementation of the Map interface, which can be used to store and manipulate key-value pairs.
- Scanner: A class in the Java Standard Library that provides methods for reading user input from the console.
- System: A class in the Java Standard Library that provides access to the standard input, output, and error streams.
- Book: A custom class that represents a book, with properties such as ISBN, title, author, year, and rental status.

Module Wise Description:

The LibraryManagementSystem class is responsible for managing the library's book catalog and rented books. It contains several methods for adding, deleting, searching, updating, renting, and returning books, as well as displaying available books.

The class has two instance variables: a HashMap to store the book catalog, and an ArrayList to store the rented books. The book catalog is stored as key-value pairs, where the key is the ISBN of the book, and the value is the Book object. The rentedBooks ArrayList simply stores the Book objects that are currently rented out.

The constructor initializes the bookCatalog and rentedBooks variables as empty HashMap and ArrayList objects, respectively.

- The addBook() method takes a Book object as its parameter and adds it to the bookCatalog HashMap. The ISBN of the book is used as the key for the HashMap, and the Book object itself is the value. If the book is added successfully, a message is printed to the console.
- The deleteBook() method takes an ISBN as its parameter and attempts to remove the corresponding Book object from the bookCatalog HashMap. If the book is found and

removed successfully, a message is printed to the console. If the book is not found, a "Book not found" message is printed instead.

- The searchBook() method takes an ISBN as its parameter and attempts to retrieve the corresponding Book object from the bookCatalog HashMap. If the book is found, its details are printed to the console. If the book is not found, a "Book not found" message is printed instead.
- The updateBook() method takes an ISBN, title, author, and year as its parameters and attempts to update the corresponding Book object in the bookCatalog HashMap. If the book is found, its title, author, and year are updated with the new values, and a "Book updated successfully" message is printed to the console. If the book is not found, a "Book not found" message is printed instead.
- The rentBook() method takes an ISBN as its parameter and attempts to rent the corresponding Book object from the bookCatalog HashMap. If the book is found and is not currently rented out, its rented variable is set to true, and the Book object is added to the rentedBooks ArrayList. A "Book rented successfully" message is printed to the console. If the book is already rented out, a "Book is already rented" message is printed instead. If the book is not found, a "Book not found" message is printed instead.
- The returnBook() method takes an ISBN and the number of days late as its parameters and attempts to return the corresponding Book object to the bookCatalog HashMap. If the book is found and is currently rented out, its rented variable is set to false, and the Book object is removed from the rentedBooks ArrayList. A "Book returned successfully" message is printed to the console, along with the fine charged for returning the book late. The fine is calculated by multiplying the number of days late by 0.5. If the book is not found or is not currently rented out, a "Book not found or not rented" message is printed instead.
- The displayAvailableBooks() method iterates through the bookCatalog HashMap and prints the details of each Book object that is not currently rented out. If no available books are found, a "No books available for rent" message is printed instead.
- The main() method is the entry point of the program. It creates an instance of the LibraryManagementSystem class, and then prompts the user to select an option from a menu using a while loop. Depending on the option selected, the corresponding method of the LibraryManagementSystem class is called with the appropriate parameters. The loop continues until the user selects the "Exit" option, at which point the program terminates.

Development Environment:

The project was developed using Java programming language and Visual Studio Code (VS Code) as the primary development environment. VS Code is a lightweight and powerful code editor that provides built-in support for Java development. It also provides useful features such as debugging, IntelliSense, and Git integration, which were used by the team to enhance the development experience. The project was developed using the Java Development Kit (JDK) version 16.0.2.

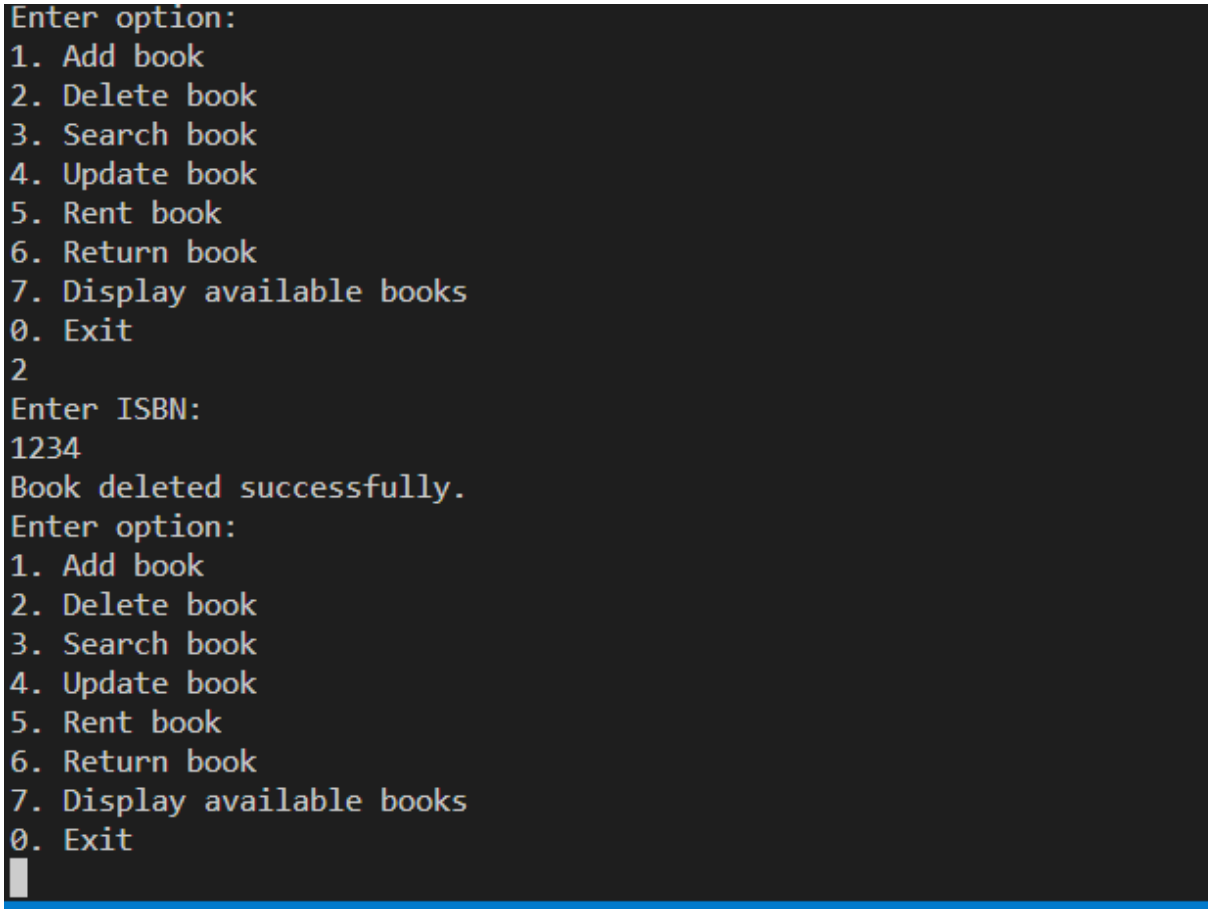
Output:

The program provides the following functionalities:

1. Add book: Allows users to add a book to the inventory by entering its ISBN, title, author, and year.

```
Enter option:
1. Add book
2. Delete book
3. Search book
4. Update book
5. Rent book
6. Return book
7. Display available books
0. Exit
1
Enter ISBN:
1234
Enter title:
Java
Enter author:
xyz
Enter year:
2023
Book added successfully.
Enter option:
1. Add book
2. Delete book
3. Search book
4. Update book
5. Rent book
6. Return book
7. Display available books
0. Exit
```

2. Delete book: Allows users to delete a book from the inventory by entering its ISBN.



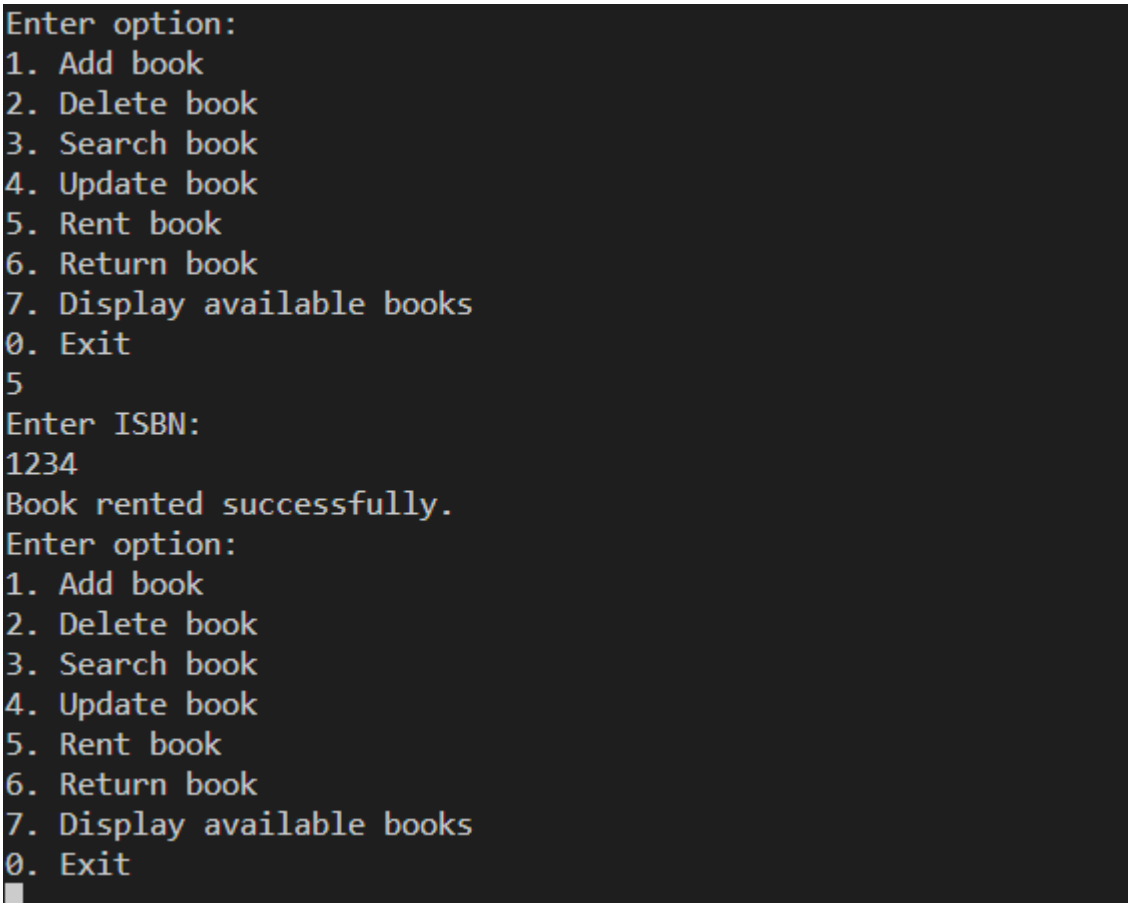
3. Search book: Allows users to search for a book in the inventory by entering its ISBN.

```
Enter option:
1. Add book
2. Delete book
3. Search book
4. Update book
5. Rent book
6. Return book
7. Display available books
0. Exit
3
Enter ISBN:
1234
Book{ISBN=1234, title=xyz, author=xyz, year=2022}
Enter option:
1. Add book
2. Delete book
3. Search book
4. Update book
5. Rent book
6. Return book
7. Display available books
0. Exit
```

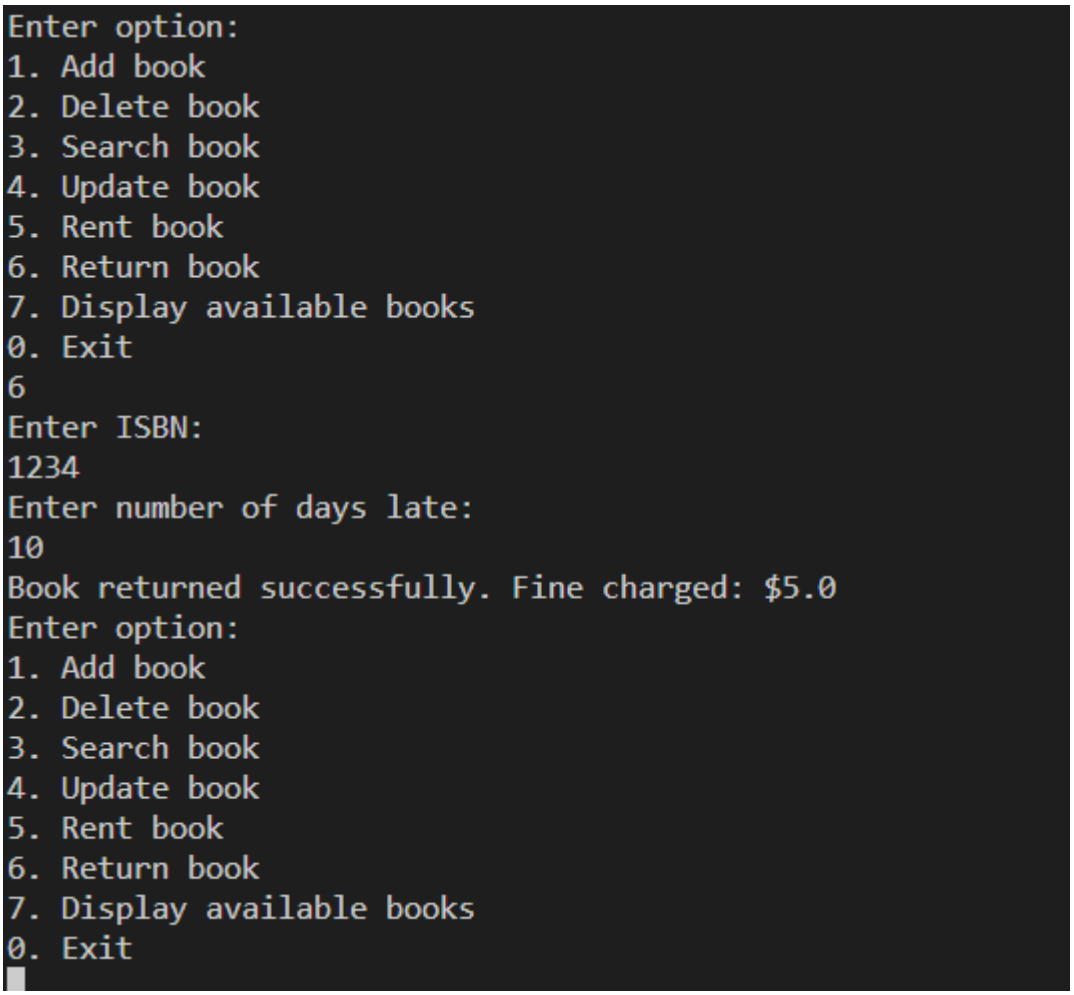
4. Update book: Allows users to update the details of a book in the inventory by entering its ISBN, and new title, author, and year.

```
Enter option:
1. Add book
2. Delete book
3. Search book
4. Update book
5. Rent book
6. Return book
7. Display available books
0. Exit
4
Enter ISBN:
1234
Enter new title (press enter to keep current title):
JAVA
Enter new author (press enter to keep current author):
XYZ
Enter new year (press 0 to keep current year):
2023
Book updated successfully.
Enter option:
1. Add book
2. Delete book
3. Search book
4. Update book
5. Rent book
6. Return book
7. Display available books
0. Exit
```

5. Rent book: Allows users to rent a book by entering its ISBN.



6. Return book: Allows users to return a rented book by entering its ISBN and the number of days it was late.



7. Display available books: Displays the list of books that are available for rent.

```
1. Add book
2. Delete book
3. Search book
4. Update book
5. Rent book
6. Return book
7. Display available books
0. Exit
7
Book{ISBN=123, title=JAVA script, author=xyz, year=2022}
Book{ISBN=1234, title=Java, author=xyz, year=2023}
Book{ISBN=11111, title=js, author=xyz, year=2000}
Enter option:
1. Add book
2. Delete book
3. Search book
4. Update book
5. Rent book
6. Return book
7. Display available books
0. Exit
```

Conclusion:

As LPU students, we have created a library management system using Java programming language. The system is designed to manage books in a library, keeping track of information such as ISBN, title, author, year, and whether or not a book has been rented.

The system is implemented using a HashMap to store the book catalog and an ArrayList to keep track of rented books. The system allows users to add, delete, search, and update books in the catalog, as well as rent and return books. A user can also view available books in the library.

The main method of the program presents a menu of options for the user to choose from. For example, the user can select option 1 to add a book to the catalog. The program then prompts the user to enter the book's ISBN, title, author, and year, which are stored in a new Book object and added to the catalog. Similarly, the user can select option 2 to delete a book from the catalog, and option 3 to search for a book using its ISBN.

The system also allows users to update book information, rent a book, and return a rented book. If a book is returned late, the system calculates a fine based on the number of days late and displays the amount charged to the user.

Overall, the library management system provides a simple and efficient way to manage books in a library. With its easy-to-use menu system and efficient data storage, the program is ideal for libraries of all sizes. By using this program, librarians can keep track of books in their collection, allowing for an organized and hassle-free library experience for all.

GITHUB LINK : https://github.com/cpon143/Library-management-system