LIBRARY MANAGEMENT SYSTEM

INTRODUCTION:

In a student life, libraries always have great impact on academic growth. However, the manual management of that much books is really hard and time consuming for librarians. To address these challenges, I have created a user-friendly library Management System, using C programming language. The system allows us to preform various tasks like adding book details, issuing books to students and recording the data of issued books as well as students permanently into text file database. In addition to that it will be able to handles book cataloging, user interactions for borrowing and returning books, book availability checks, and basic late return tracking.

OBJECTIVE:

The objective of this project to develop a Library Management System that efficiently handles book cataloging, user interactions for borrowing and returning books, book availability checks, and basic late return tracking, all within a user-friendly text-based interface

METHODOLOGY:

I have chosen the c-programming language to develop this project. I have used Dev C IDE to write the source code. File handling is used for database to store the record. Similarly, other various component of c such as structure, function, array, pointers, etc. are also used here. For now, no GUI is used so the program is used to run in terminal.

CODE OVERVIEW

Libraries Used

The code for the Library Management System incorporates standard libraries:

stdio.h: Standard I/O functions for input and output.

stdlib.h: Standard library functions for memory allocation and other utilities.

time.h: Functions for time and date handling.

string.h: String manipulation functions.

Data Structures

The system employs two key data structures:

Book Structure (struct books): Stores book information, including a unique book ID, title, author, and date of addition to the library.

Student Structure (struct student): Manages student details, including a unique student ID, name, class, associated book, and date of transaction.

Functions

The system includes several functions that handle various aspects of library management:

addBook() : Allows library staff to add new books to the catalog.

booksList() : Displays a list of available books with their details.

del() : Enables staff to remove books from the catalog.

issueBook() : Facilitates book issuance to patrons, recording details of the transaction.

issueList(): Displays a list of issued books and their associated user details.

returnBook(): Manages the return of books, calculates fines for late returns, and updates the issuance records.

ALGORITHM, FLOWCHART & OUTCOMES

Step 1: Initialize Data Structures and Open Files

1.Initialize data structures, including structures for books and students.

2.Open necessary files, such as "books.txt" and "issue.txt," for reading and writing data.

Step 2: Display Main Menu

1.Clear the screen.

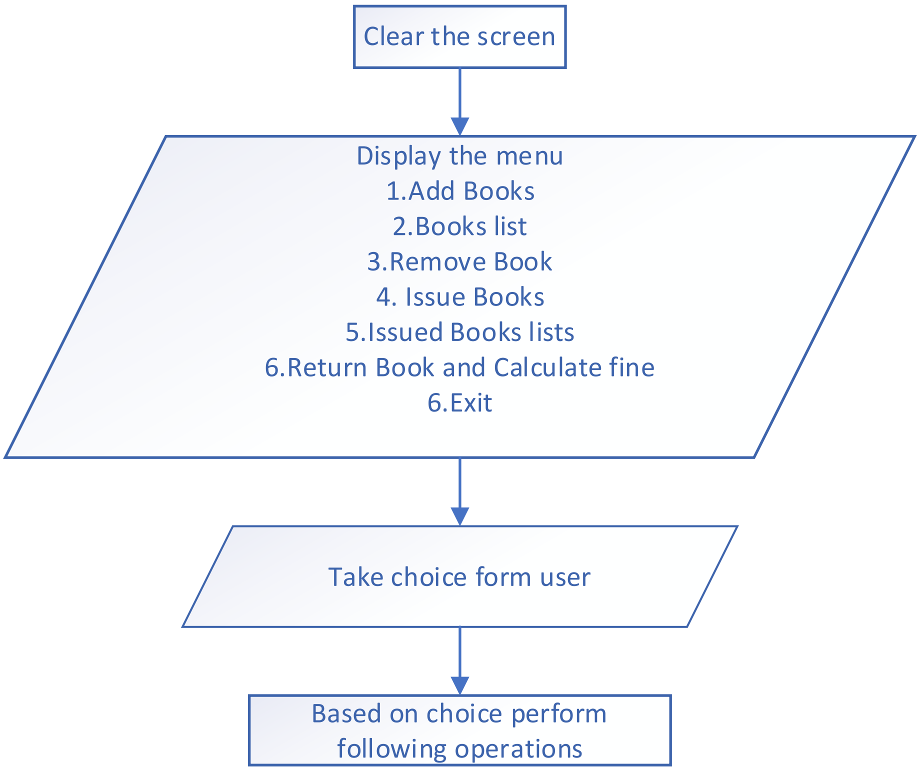
2.Display the main menu with options for various library operations:

- Add Book
- Books List
- Remove Book
- Issue Book
- Issued Book List
- Return Book and Calculate Fine
- Exit

Step 3: User Input and Selection

1.Prompt the user to enter their choice (integer).

2.Based on the choice, execute the corresponding operation.



Step 4: Add Book (addBook Function)

1.Get the current date and store it in myDate.

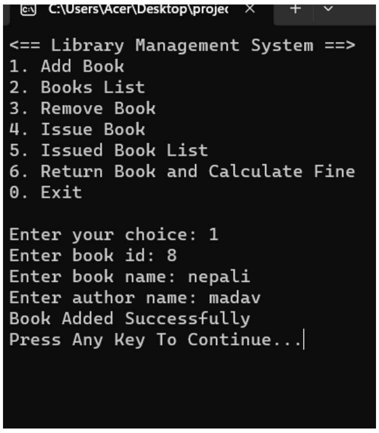
2.Prompt the user for book information:

- Book ID
- Book Name
- Author Name

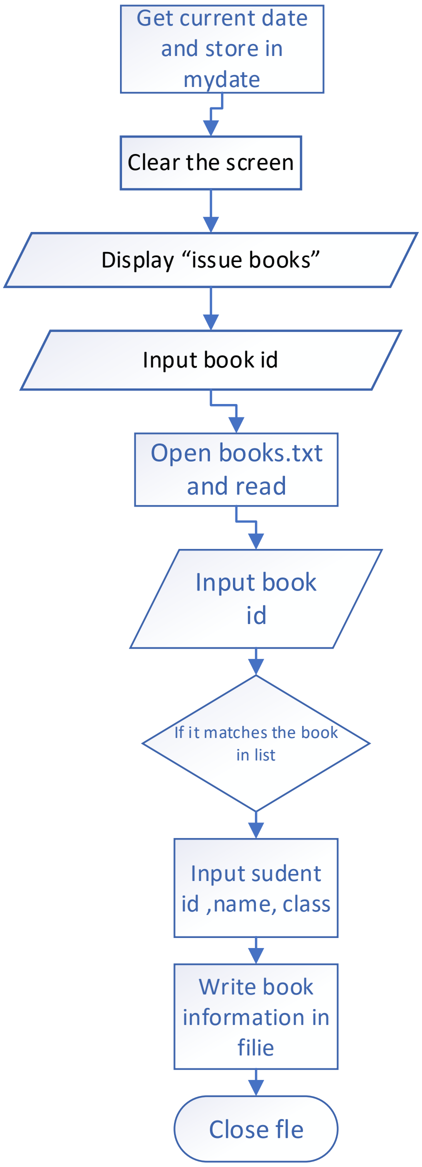
3.Display a success message.

4.Write the book information to the "books.txt" file.

5.Close the file.



```
<== Library Management System ==>
1. Add Book
2. Books List
3. Remove Book
4. Issue Book
5. Issued Book List
6. Return Book and Calculate Fine
0. Exit

Enter your choice: 1
Enter book id: 8
Enter book name: nepali
Enter author name: madav
Book Added Successfully
Press Any Key To Continue...
```


Get current date and store in mydate
Clear the screen
Display “issue books”
Input book id
Open books.txt and read
Input book id
If it matches the book in list
Input sudent id ,name, class
Write book information in filie
Close fle

Step 5: Display Books List (booksList Function)

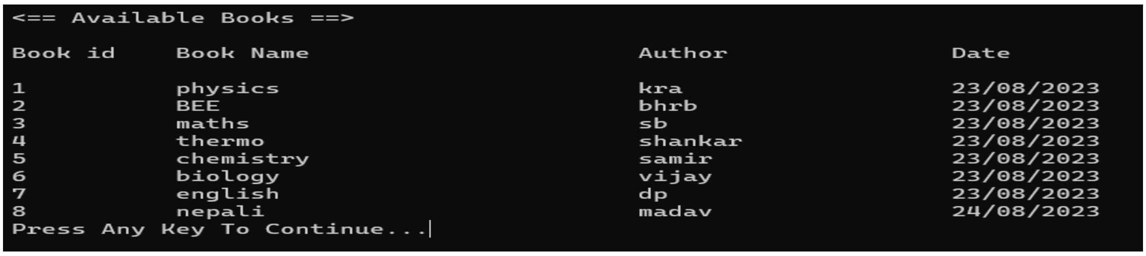
1.Clear the screen.

2.Display the "Available Books" header.

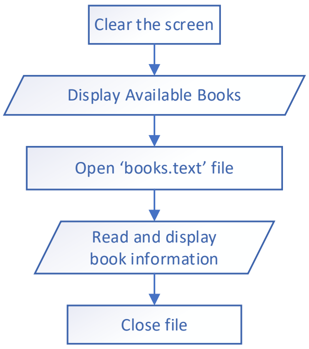
3.Open the "books.txt" file for reading.

4.Read and display book information (ID, Title, Author, Date) from the file.

5.Close the file.



```
<== Available Books ==>

Book id     Book Name                              Author                  Date

1           physics                                kra                     23/08/2023
2           BEE                                    bhrb                    23/08/2023
3           maths                                  sb                      23/08/2023
4           thermo                                 shankar                 23/08/2023
5           chemistry                              samir                   23/08/2023
6           biology                                vijay                   23/08/2023
7           english                                dp                      23/08/2023
8           nepali                                 madav                   24/08/2023
Press Any Key To Continue...
```



Step 6: Remove Book (del Function)

1.Clear the screen.

2.Display the "Remove Books" header.

3.Prompt the user to enter the Book ID to remove.

4.Open the "books.txt" file for reading and a temporary file for writing.

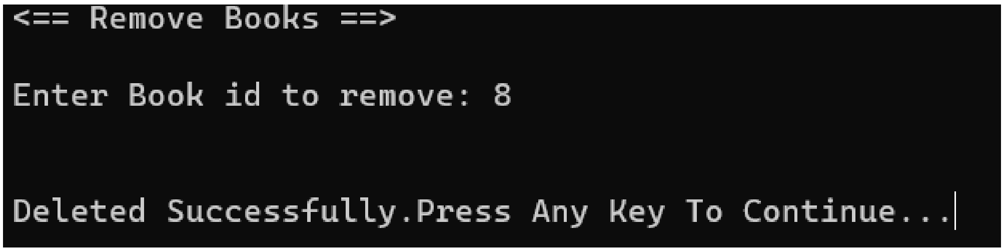
5.While reading book information from the file:

- If the Book ID matches the user's input, mark it as found.
- If not, write the book information to the temporary file.

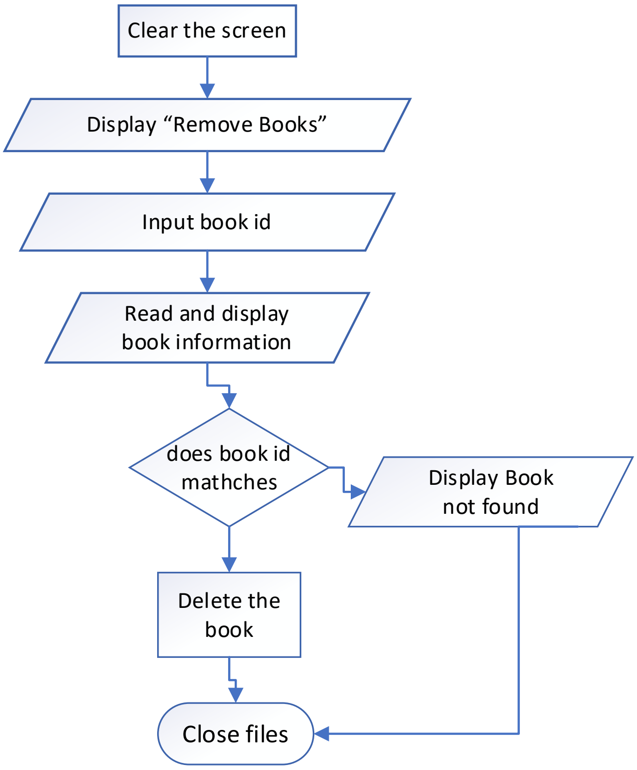
6.If the book is found, display a deletion message; otherwise, display a not found message.

7.Close both files.

8.Delete the original "books.txt" file and rename the temporary file to "books.txt."



```
<== Remove Books ==>

Enter Book id to remove: 8


Deleted Successfully.Press Any Key To Continue...
```



Step 7: Display Issued Books List (issueList Function)

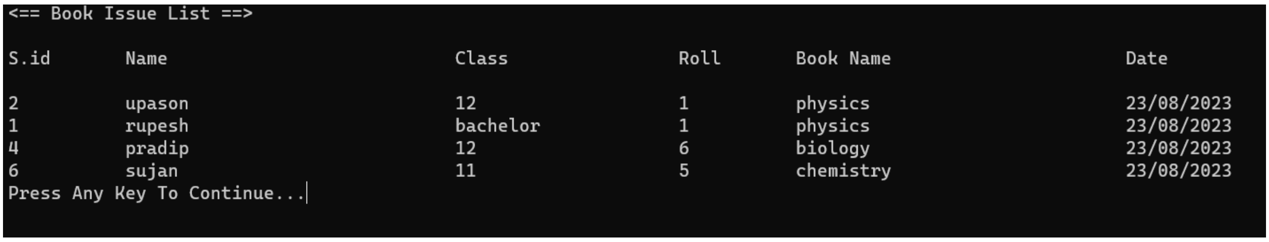
1.Clear the screen.

2.Display the "Book Issue List" header.

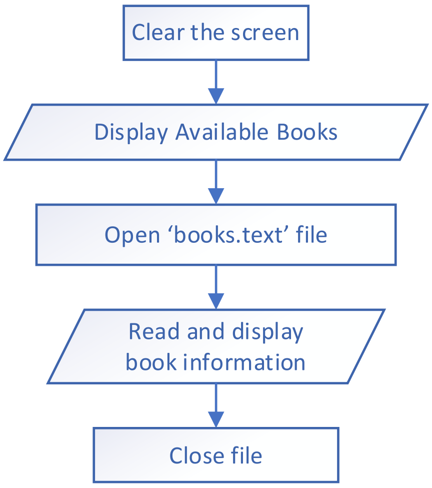
3.Open the "issue.txt" file for reading.

4.Read and display issued book information (Student ID, Name, Class, Roll, Book Name, Date) from the file.

5.Close the file.



```
<== Book Issue List ==>

S.id       Name                          Class             Roll      Book Name                  Date

2          upason                        12                1         physics                    23/08/2023
1          rupesh                        bachelor          1         physics                    23/08/2023
4          pradip                        12                6         biology                    23/08/2023
6          sujan                         11                5         chemistry                  23/08/2023
Press Any Key To Continue...
```



Step 8: Issue Book (issueBook Function)

1.Get the current date and store it in myDate.

2.Clear the screen.

3.Display the "Issue Books" header.

4.Prompt the user to enter the Book ID to issue.

5.Open the "books.txt" file for reading.

6.While reading book information from the file:

- If the Book ID matches the user's input, mark it as found and store book details.

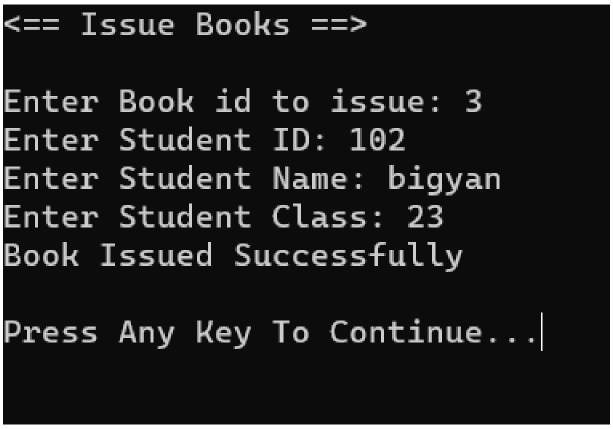
7.If the book is found, prompt the user for:

- Student ID
- Student Name
- Student Class

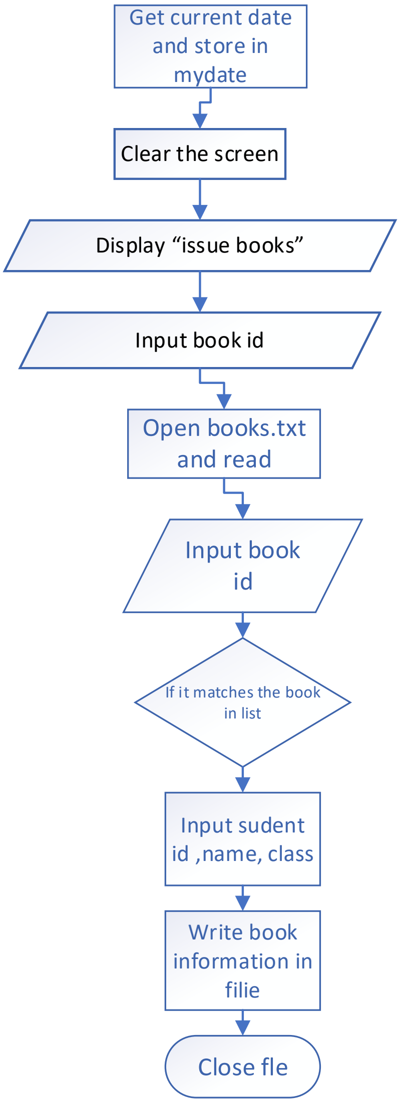
8.Display a success message.

9.Write the student's information, including the book's Roll (ID), to the "issue.txt" file.

10.Close the file.

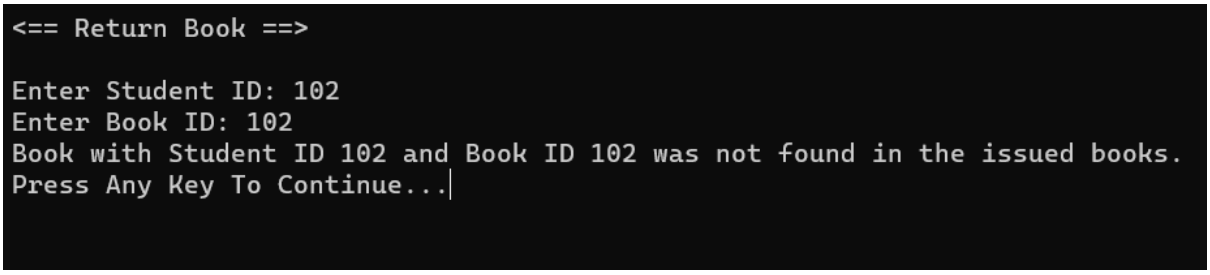


```
<== Issue Books ==>

Enter Book id to issue: 3
Enter Student ID: 102
Enter Student Name: bigyan
Enter Student Class: 23
Book Issued Successfully

Press Any Key To Continue...
```

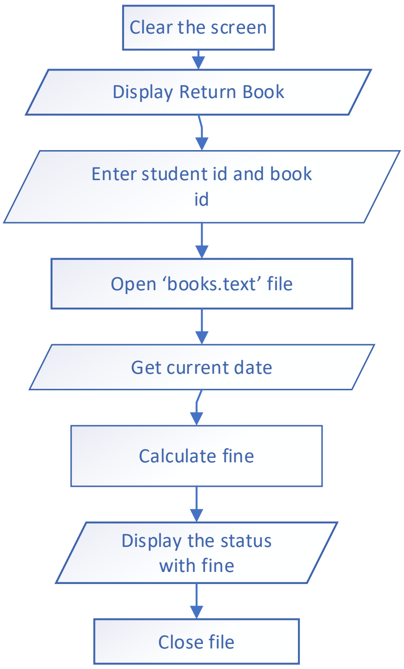

Get current date and store in mydate
Clear the screen
Display “issue books”
Input book id
Open books.txt and read
Input book id
If it matches the book in list
Input sudent id ,name, class
Write book information in filie
Close fle

Step 9: Return Book and Calculate Fine (returnBook Function)

1. Clear the screen.
2. Display the "Return Book" header.
3. Prompt the user to enter:
   - Student ID
   - Book ID
4. Open the "issue.txt" file for reading and a temporary file for writing.
5. Get the current date and calculate the fine for late returns if applicable.
6. Display the return status (on time or overdue).
7. While reading issued book information from the file:
   - If the Student ID and Book ID match the user's input, mark it as found.
   - If not, write the issued book information to the temporary file.
8. If the issued book is found, calculate and display the fine (if any) and mark the book as returned.
9. Close both files.
10. Delete the original "issue.txt" file and rename the temporary file to "issue.txt."

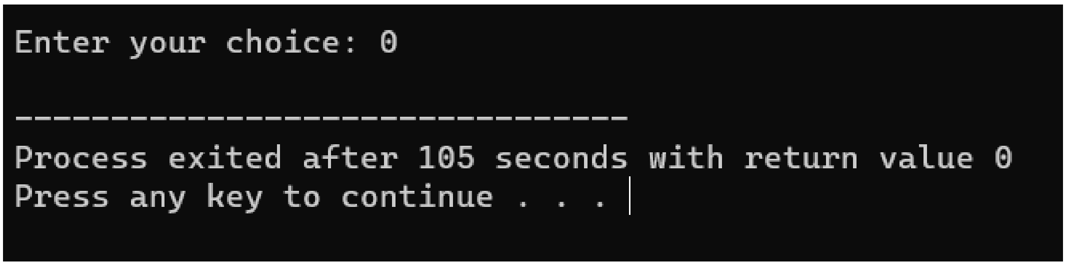


```
<== Return Book ==>

Enter Student ID: 102
Enter Book ID: 102
Book with Student ID 102 and Book ID 102 was not found in the issued books.
Press Any Key To Continue...
```



Step 10: Exit

1.If the user selects the Exit option, the program terminates.



```
Enter your choice: 0

--------------------------------
Process exited after 105 seconds with return value 0
Press any key to continue . . .
```

CONCLUSION

In summary, the Library Management System project successfully achieves its primary goals of efficient book cataloging, user management, book lending, and late return management, while providing user friendly interface. The system, built in C programming, prioritizes simplicity and ease of use, making it a valuable tool for small to medium-sized libraries.