CSA-5003- Problem Solving and Programming Lab

Project title:

📘 Library Management System 📘

Submitted by: Prajyot Salelkar (2531)

Sampada Kelkar (2540)

Course: Masters of Computer Applications

(MCA)

Submitted To: Goa Business School - Goa

University, Taleigao, Goa

Academic Year: 2025–2026

Abstract

This project introduces a Library Management System developed in C, designed to efficiently manage library operations and records. The system enables the addition, modification, deletion, and listing of student and book records, while also maintaining borrow and return transactions. Leveraging C language constructs such as structures, arrays, and modular programming, the application ensures organized data handling and operational clarity.

Data persistence is achieved through binary file storage, allowing information to be securely retained between sessions. The project effectively demonstrates core concepts of C programming, including structured data representation and file handling, offering a practical solution for library record management.

Introduction

Library management is a crucial task in educational institutions. Traditional manual systems are time-consuming and prone to errors. This project automates the process using C programming concepts. The program provides functions for registering students, maintaining book records, issuing and returning books, and storing data permanently using files.

It focuses on demonstrating core programming logic, data structures, and file I/O operations while keeping the interface simple and text-based for console execution.

System Requirements

Hardware Requirements:-

- Processor: Intel Core i3 or above
- RAM: Minimum 2 GB
- Storage: 100 MB free space
- Display: Standard text console

Software Requirements:- •

Operating System: Windows/Linux

- Compiler: GCC or Code::Blocks , Turbo C, Dev C++
- Language: C Programming

System Design

The system design follows a modular structure, separating functionality into independent files:

- student.c and student.h handle student registration and management.
- book.c and book.h manage book details.
- borrow.c maintains borrowing and returning records.
- fileops.c provides functions for saving and loading all records to/from files.
- utils.c manages common input/output operations like reading integers, strings, and dates safely.

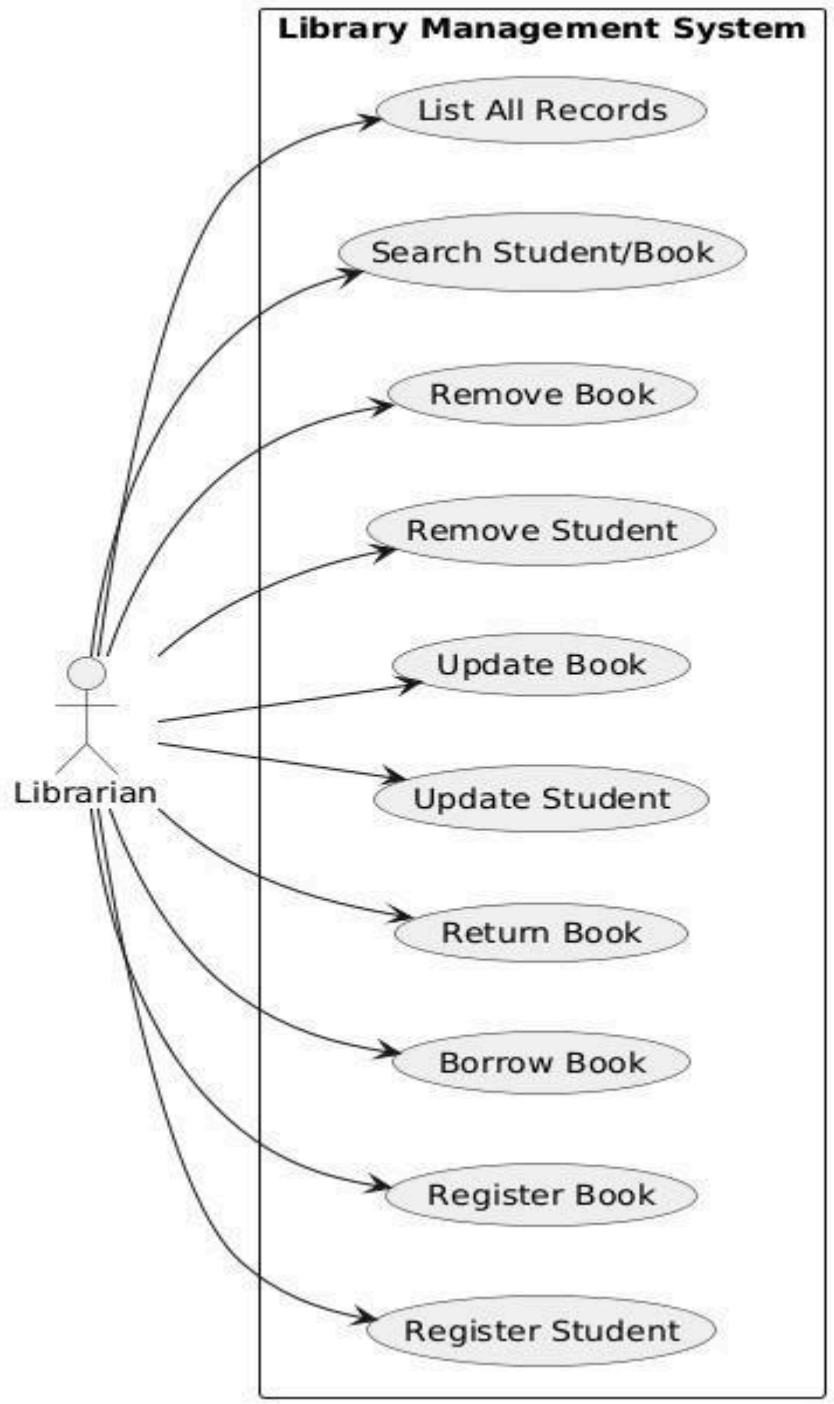
The system follows a modular design where each function (students, books, and borrow records) is handled by a separate module.

The **librarian** is the primary user who interacts directly with the system through a console interface to manage student records, add or remove books, and handle borrowing and returning transactions.
The **student** interacts **indirectly** with the system through the librarian when requesting or returning a book.

Use Case Diagram

Actors: Student, Librarian (or System User)

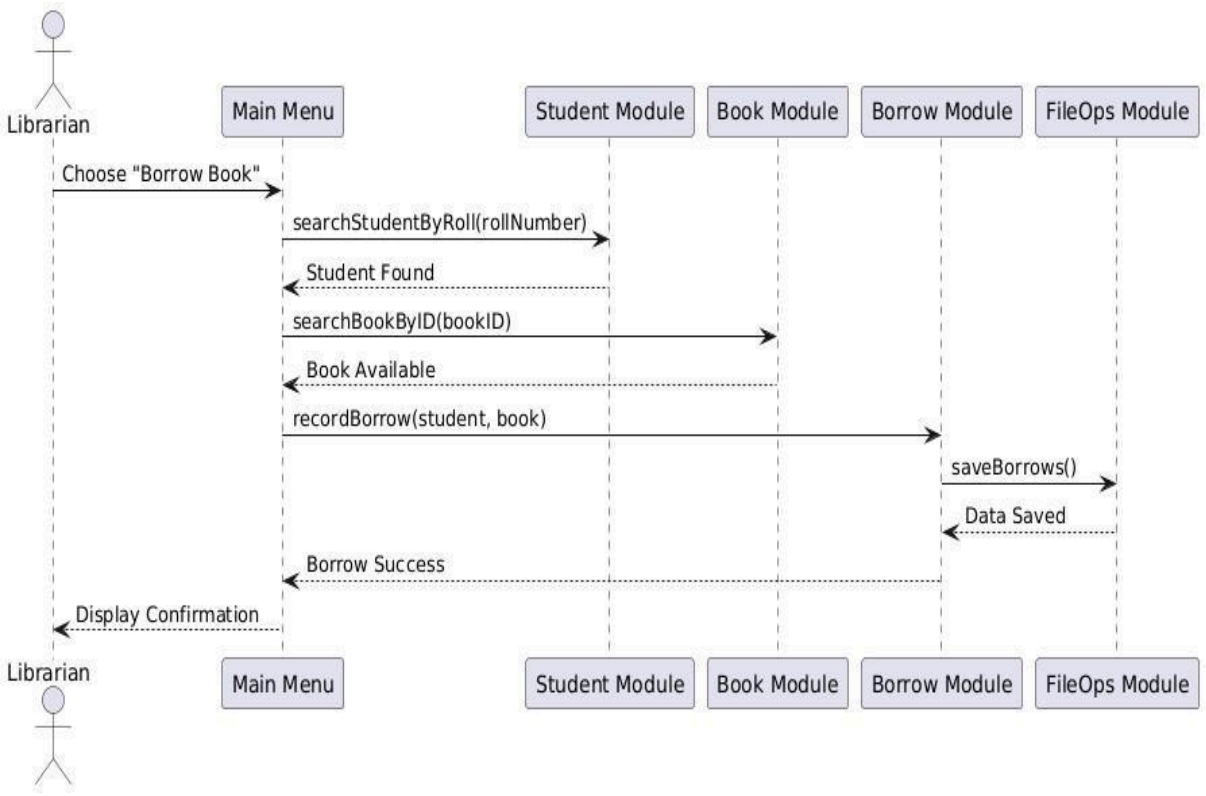
Use Cases: Register Student, Add / Update / Remove Book, Borrow Book, Return Book, View All Records



Sequence Diagram

Scenario: *Borrowing a Book*

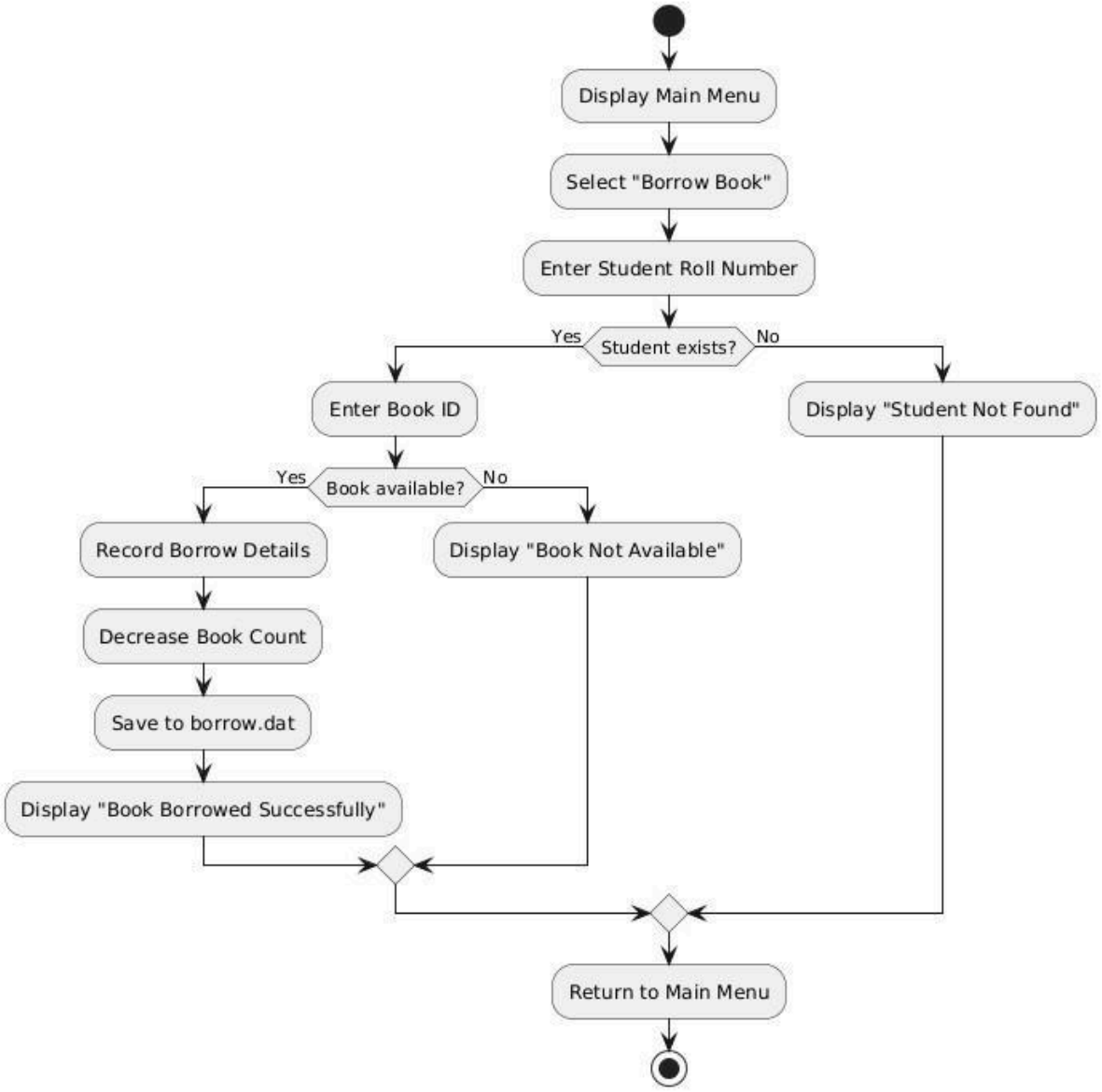
1. User selects “Borrow Book.”
2. System checks student record using roll number.
3. System searches for the requested book.
4. If available, a borrow record is created and stored in borrow.dat.
5. The book’s availability is updated in books.dat.
6. Confirmation message is displayed.



Activity Diagram

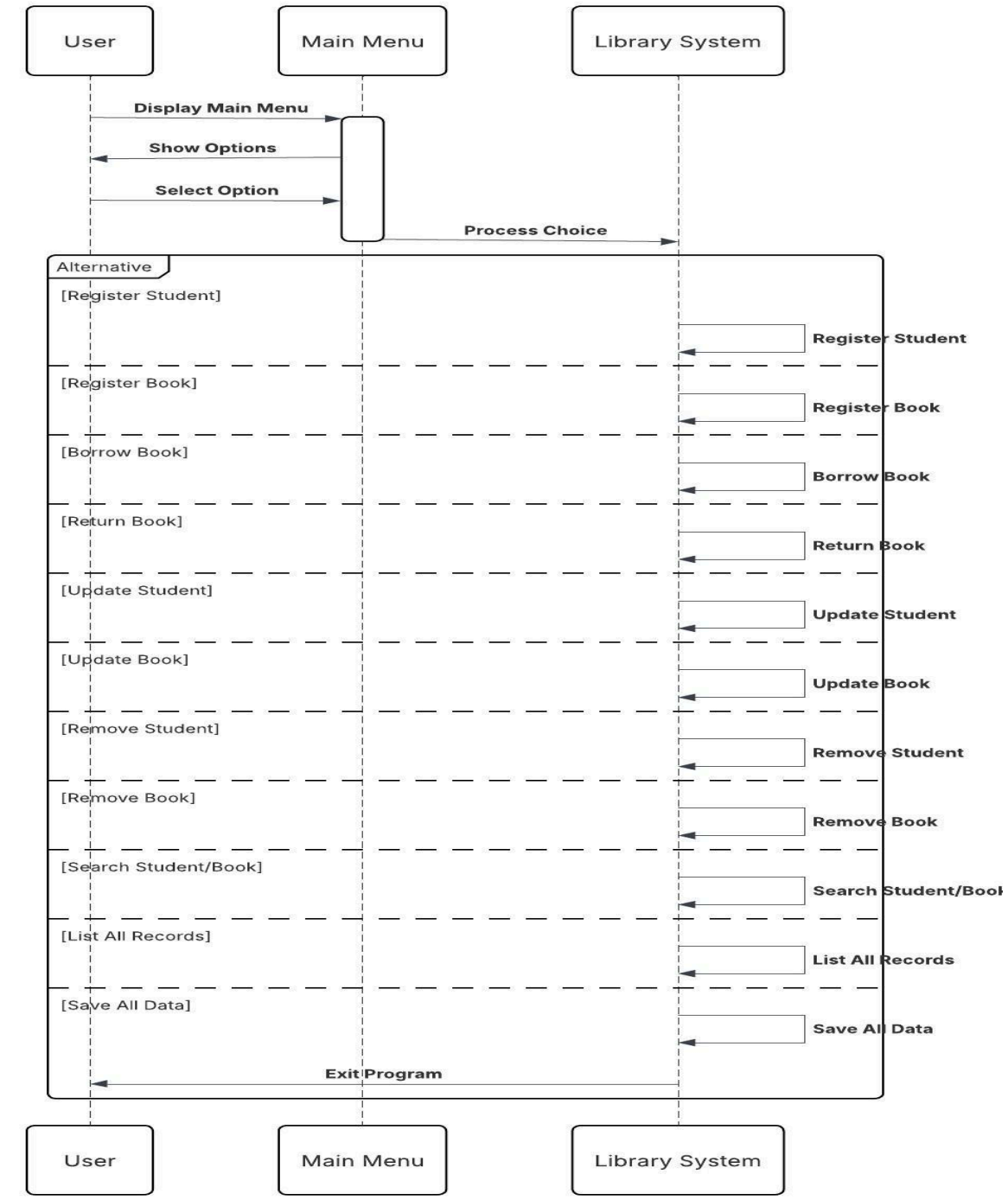
Main Process Flow:

Start → Display Menu → Choose Option
→ If Register Student → Input Details → Save to File
→ If Add Book → Input Book Details → Save
→ If Borrow → Validate Student & Book → Record Transaction
→ If Return → Update Borrow Record
→ Display Success → Back to Menu → Exit



Flow chart

It shows the step-by-step flow of the Library Management System, starting from loading data and displaying the menu to performing selected operations and saving data before exit.



Modules Description

1. Student Module

This module manages all student-related operations.
Functions include registering a new student, updating existing details, removing a student, listing all students, and searching by name or roll number.
Data such as roll number, name, program, and library card number are stored in a file (students.dat) for persistence.

2. Book Module

This module handles all operations related to books. The librarian can register new books, update book information, remove books, search for books by name, and view a list of all available books.
Each record contains book ID, name, author, edition, total copies, and borrowed copies, which are saved in books.dat.

3. Borrow and Return Module

This module manages the process of issuing and returning books. It tracks which student has borrowed which book, along with the borrow date and return status. It also supports viewing all borrowed books and identifying overdue books. All borrow records are stored in borrow.dat.

4. File Operations Module

This module performs reading and writing of data files for students, books, and borrow records. It ensures that all changes made during program execution are saved permanently. Functions include loadStudents(), saveStudents(), loadBooks(), saveBooks(), loadBorrows(), and saveBorrows().

5. Utility Module

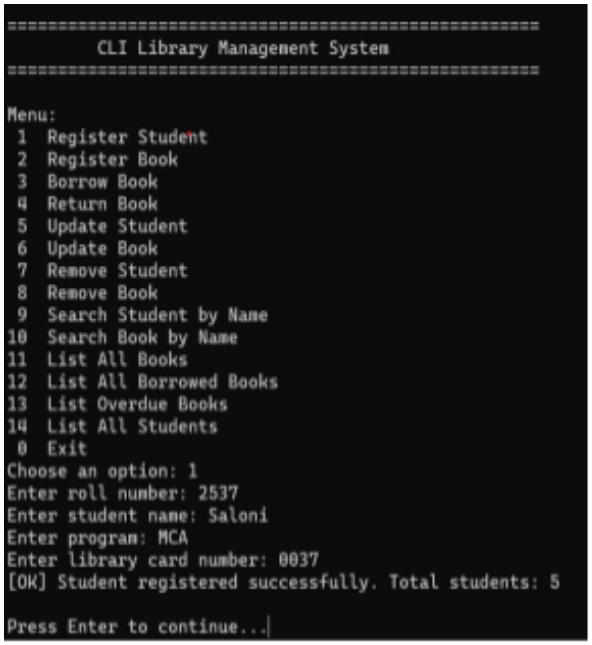
This module provides helper functions for safe input handling and date processing. It includes functions for reading integers and strings, validating input, and converting dates between string and time formats.
Examples include getInt(), getString(), parseDateString(), and formatDate().

6. Main Control Module

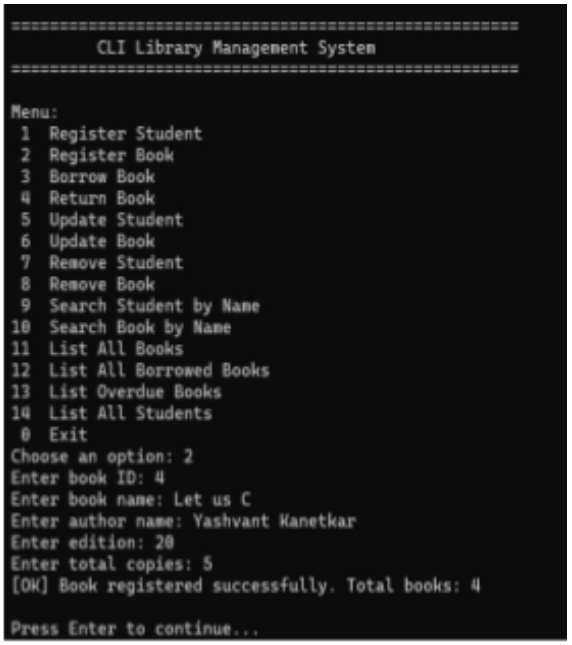
This is the central module that connects all others. It provides a command-line interface (CLI) menu through which the librarian can perform all operations like registering students, adding books, borrowing, returning, updating, or searching records. It calls appropriate functions from other modules and ensures that data is loaded at startup and saved before exit.

Screenshots and App flow

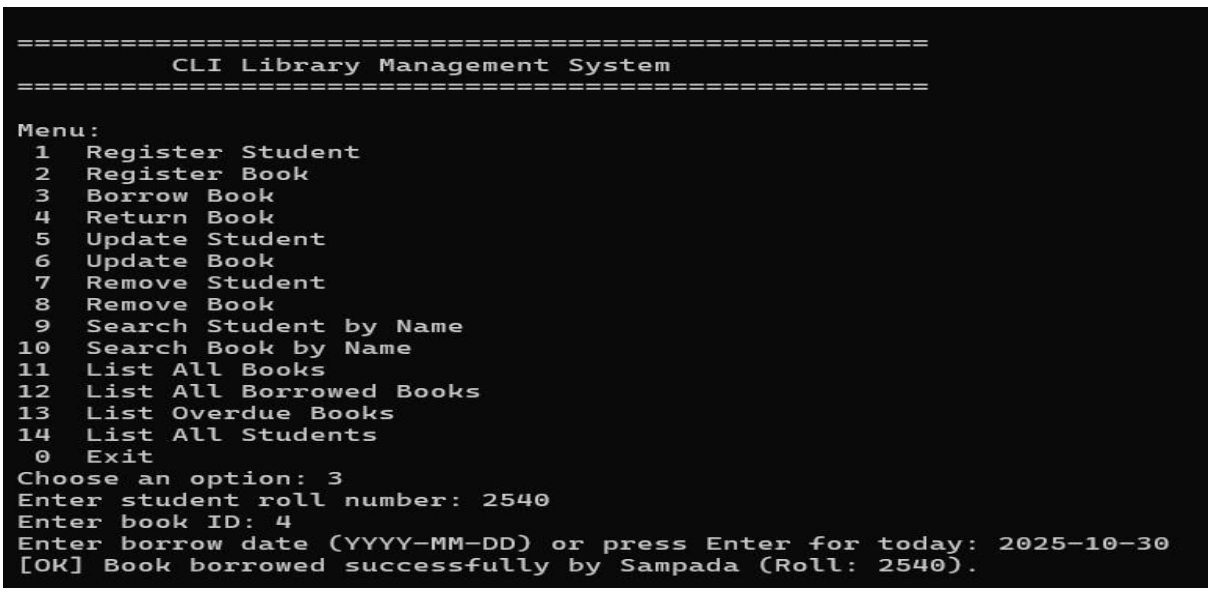
Register a student (option 1)



Register a book (option 2)



Borrow a book (option 3)



Return book (option 4)

```
[INFO] Loaded 5 students.
[INFO] Loaded 4 books.
[INFO] Loaded 1 borrow records.

======================================================
         CLI Library Management System
======================================================

Menu:
 1  Register Student
 2  Register Book
 3  Borrow Book
 4  Return Book
 5  Update Student
 6  Update Book
 7  Remove Student
 8  Remove Book
 9  Search Student by Name
10  Search Book by Name
11  List All Books
12  List All Borrowed Books
13  List Overdue Books
14  List All Students
 0  Exit
Choose an option: 4
Enter student roll number: 2540
Enter book ID: 4
[OK] Book returned on time. No fine.
[OK] Book returned successfully.
```

Update student (option 5)

```
==============================================
        CLI Library Management System
==============================================

Menu:
 1  Register Student
 2  Register Book
 3  Borrow Book
 4  Return Book
 5  Update Student
 6  Update Book
 7  Remove Student
 8  Remove Book
 9  Search Student by Name
10  Search Book by Name
11  List All Books
12  List All Borrowed Books
13  List Overdue Books
14  List All Students
 0  Exit
Choose an option: 5
Enter roll number to update: 2540
Enter new name: Sampada_kelkar
Enter new program: MCA
Enter new library card number: 0069
[OK] Student updated successfully.

Press Enter to continue...
```

List of students after update (option 5-14)

```
==============================================
       CLI Library Management System
==============================================

Menu:
 1  Register Student
 2  Register Book
 3  Borrow Book
 4  Return Book
 5  Update Student
 6  Update Book
 7  Remove Student
 8  Remove Book
 9  Search Student by Name
10  Search Book by Name
11  List All Books
12  List All Borrowed Books
13  List Overdue Books
14  List All Students
 0  Exit
Choose an option: 14

+------------------ Registered Students ------------------+
| Roll:   2531 | Name: Prajyot          | Program: MCA | Card: 001  |
| Roll:   2540 | Name: Sampada_kelkar   | Program: MCA | Card: 0069 |
| Roll:   2521 | Name: Eesha            | Program: MCA | Card: 3    |
| Roll:   2538 | Name: Sam              | Program: MCA | Card: 004  |
| Roll:   2537 | Name: Saloni           | Program: MCA | Card: 0037 |
+---------------------------------------------------------+
```

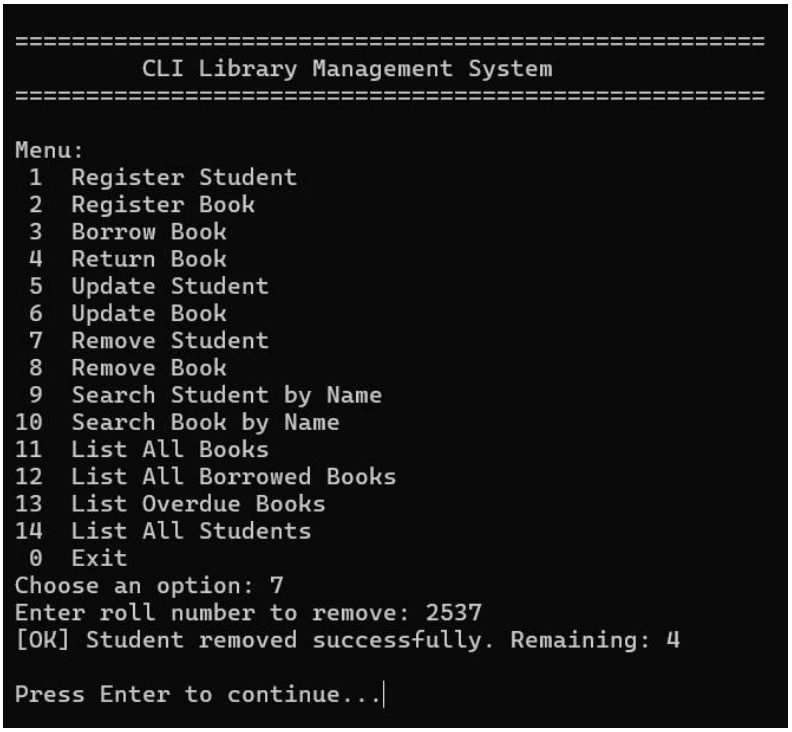
Update books (option 6)

```
======================================================
         CLI Library Management System
======================================================

Menu:
 1  Register Student
 2  Register Book
 3  Borrow Book
 4  Return Book
 5  Update Student
 6  Update Book
 7  Remove Student
 8  Remove Book
 9  Search Student by Name
10  Search Book by Name
11  List All Books
12  List All Borrowed Books
13  List Overdue Books
14  List All Students
 0  Exit
Choose an option: 6
Enter book ID to update: 2
Enter new name: Network_2
Enter new author: Andrew_T
Enter new edition: 8
Enter total copies: 10
[OK] Book updated successfully.
```

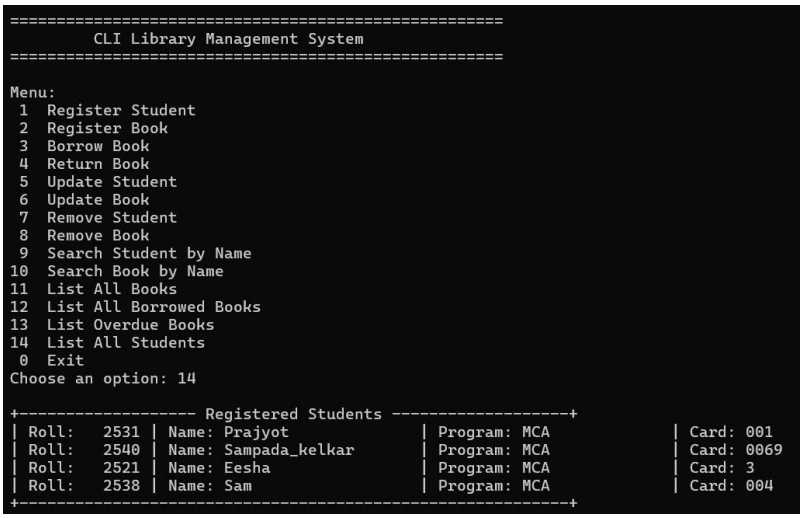
Check if book is updated (option 11)

```
================================================
        CLI Library Management System
================================================

Menu:
 1  Register Student
 2  Register Book
 3  Borrow Book
 4  Return Book
 5  Update Student
 6  Update Book
 7  Remove Student
 8  Remove Book
 9  Search Student by Name
10  Search Book by Name
11  List All Books
12  List All Borrowed Books
13  List Overdue Books
14  List All Students
 0  Exit
Choose an option: 11

+------------------- Library Books -------------------+
| ID: 1 | OS                          | Author: Tanenbaum       | Ed: 5   | Total:   3 | Borrowed:   0 | Avail:   3
| ID: 2 | Network_2                   | Author: Andrew_T        | Ed: 8   | Total:  10 | Borrowed:   0 | Avail:  10
| ID: 3 | SE                          | Author: SK              | Ed: 9   | Total:   3 | Borrowed:   0 | Avail:   3
| ID: 4 | Let us C                    | Author: Yashvant Kanetkar | Ed: 20  | Total:   5 | Borrowed:   0 | Avail:   5
+-----------------------------------------------------+
```

Remove student(option 7)



List of students after removing student(option 7-14)



Remove a book(option 8)

```
=====================================================
         CLI Library Management System
=====================================================

Menu:
 1  Register Student
 2  Register Book
 3  Borrow Book
 4  Return Book
 5  Update Student
 6  Update Book
 7  Remove Student
 8  Remove Book
 9  Search Student by Name
10  Search Book by Name
11  List All Books
12  List All Borrowed Books
13  List Overdue Books
14  List All Students
 0  Exit
Choose an option: 8
Enter book ID to remove: 4
[OK] Book removed successfully. Remaining: 3

Press Enter to continue...
```

After removing book(option 11)

```
=====================================================
         CLI Library Management System
=====================================================

Menu:
 1  Register Student
 2  Register Book
 3  Borrow Book
 4  Return Book
 5  Update Student
 6  Update Book
 7  Remove Student
 8  Remove Book
 9  Search Student by Name
10  Search Book by Name
11  List All Books
12  List All Borrowed Books
13  List Overdue Books
14  List All Students
 0  Exit
Choose an option: 11

+------------------- Library Books -------------------+
| ID: 1 | OS                          | Author: Tanenbaum       | Ed: 5   | Total:   3 | Borrowed:   0 | Avail:   3
| ID: 2 | Network_2                   | Author: Andrew_T        | Ed: 8   | Total:  10 | Borrowed:   0 | Avail:  10
| ID: 3 | SE                          | Author: SK              | Ed: 9   | Total:   3 | Borrowed:   0 | Avail:   3
+-----------------------------------------------------+
```

Search student(option 9)

```
=====================================================
          CLI Library Management System
=====================================================

Menu:
 1  Register Student
 2  Register Book
 3  Borrow Book
 4  Return Book
 5  Update Student
 6  Update Book
 7  Remove Student
 8  Remove Book
 9  Search Student by Name
10  Search Book by Name
11  List All Books
12  List All Borrowed Books
13  List Overdue Books
14  List All Students
 0  Exit
Choose an option: 9
Enter student name to search: Prajyot
[FOUND] Prajyot | Roll: 2531 | Program: MCA | Card: 001

Press Enter to continue...
```

Search student by name (if not found)

```
=====================================================
          CLI Library Management System
=====================================================

Menu:
 1  Register Student
 2  Register Book
 3  Borrow Book
 4  Return Book
 5  Update Student
 6  Update Book
 7  Remove Student
 8  Remove Book
 9  Search Student by Name
10  Search Book by Name
11  List All Books
12  List All Borrowed Books
13  List Overdue Books
14  List All Students
 0  Exit
Choose an option: 9
Enter student name to search: Prachi
[INFO] Student not found.
```

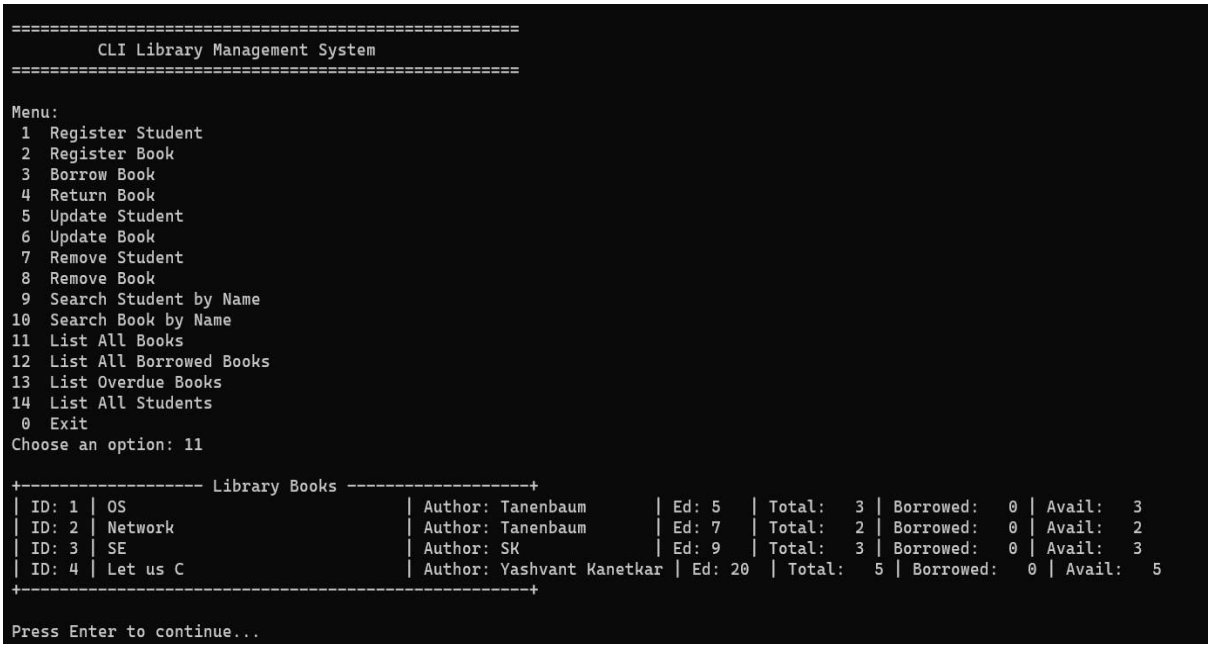
Search book by Name (option 10)

```
======================================================
          CLI Library Management System
======================================================

Menu:
 1  Register Student
 2  Register Book
 3  Borrow Book
 4  Return Book
 5  Update Student
 6  Update Book
 7  Remove Student
 8  Remove Book
 9  Search Student by Name
10  Search Book by Name
11  List All Books
12  List All Borrowed Books
13  List Overdue Books
14  List All Students
 0  Exit
Choose an option: 10
Enter book name to search: SE
[FOUND] ID: 3 | SE | Author: SK | Ed: 9 | Total: 3 | Borrowed: 0

Press Enter to continue...
```

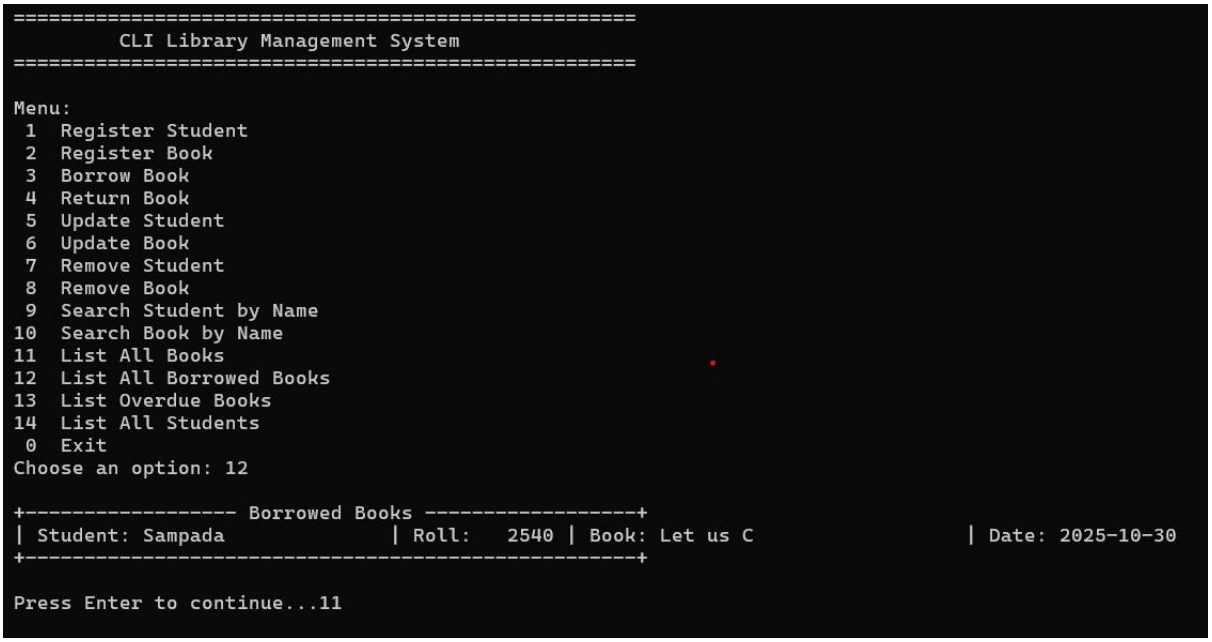
Search book by Name (if not found)

```
======================================================
          CLI Library Management System
======================================================

Menu:
 1  Register Student
 2  Register Book
 3  Borrow Book
 4  Return Book
 5  Update Student
 6  Update Book
 7  Remove Student
 8  Remove Book
 9  Search Student by Name
10  Search Book by Name
11  List All Books
12  List All Borrowed Books
13  List Overdue Books
14  List All Students
 0  Exit
Choose an option: 10
Enter book name to search: Object Oriented
[INFO] Book not found.
```

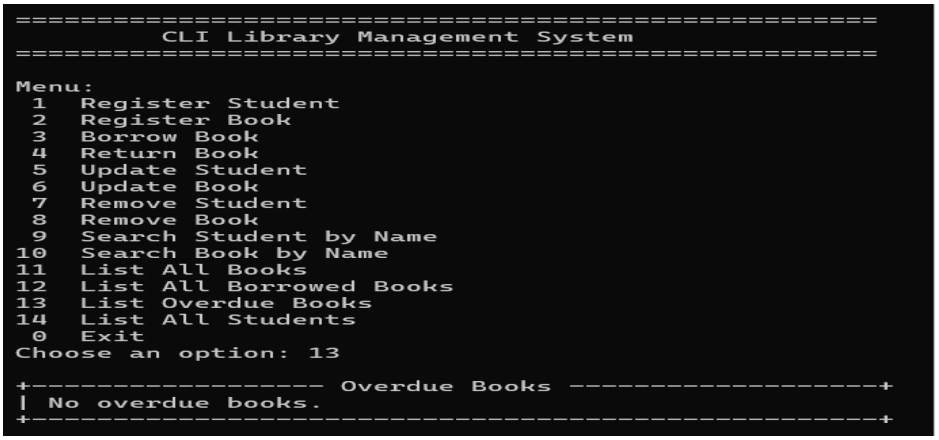
List of books (option 11)



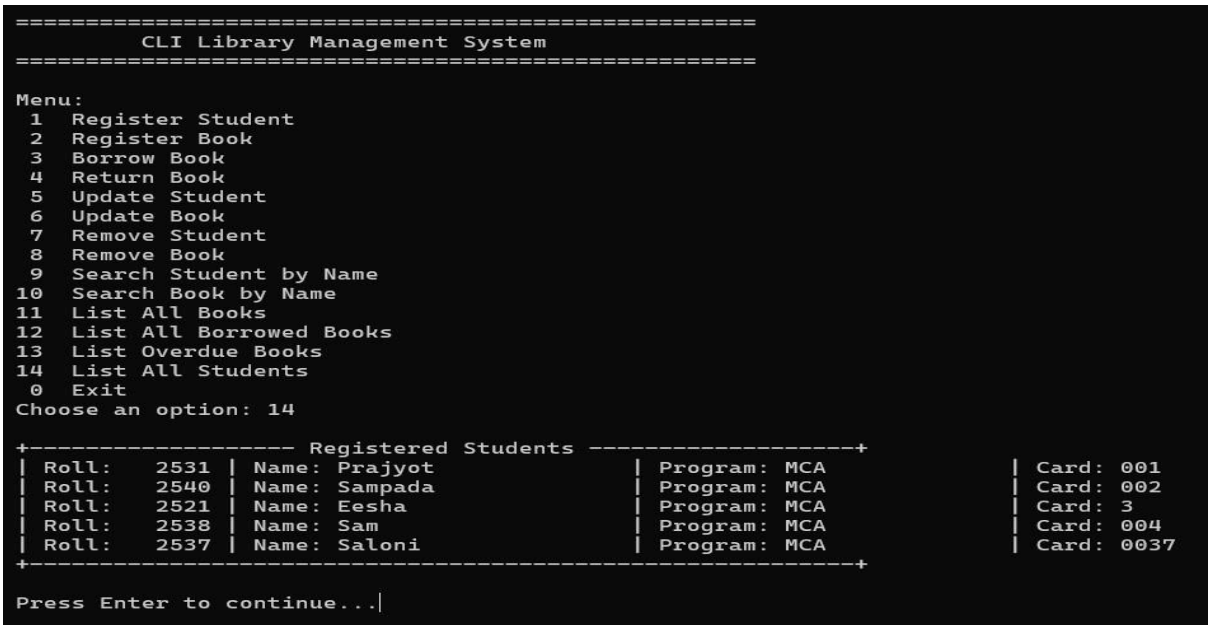
List of Borrowed books (option 12)



Overdue book list (option 13)



List of all students(option 14)



Advantages

1. Simple and user-friendly console interface.
2. Secure and persistent data storage using binary files.
3. Modular and structured programming makes maintenance easy.
4. Fast operations due to in-memory arrays and direct file access.
5. Demonstrates key C programming concepts like structures, file I/O, and modular design.

Limitations

1. No GUI interface (console-based only).
2. Single-user access at a time.
3. No online or networked database integration.
4. Limited error handling for corrupted files.

Future Scope

1. Develop a graphical version using GTK or C++ GUI frameworks.
2. Integrate with an online database (like Firebase or MySQL).
3. Add user authentication and different access levels.
4. Implement fine calculation and due-date reminders.
5. Create reports for issued and returned books automatically.

Conclusion

The Library Management System successfully automates the core functions of a small library. It provides a reliable and efficient way to handle book and student records. The project demonstrates the practical use of structures, arrays, and file operations in C programming and forms a solid base for future improvements like GUI and online integration.

References

- TutorialsPoint: File Handling in C
- GeeksforGeeks: Structure and Pointers in C
- W3 schools : https://www.w3schools.com/c/