

Project File- Library Management System In C

Course- Btech Cse

Group Members:

- Ishanvi kalra - 590025196
- Navya Agarwal- 590024522

Faculty- Vinod Kumar

Problem Definition

In our college library, most of the work like keeping record of books, issuing and returning books, and checking availability is often done manually.

Manual system has:

- Chances of human error
- Difficulty in searching books
- Time taken in writing slips and registers

Problem Statement:

To design and implement a Library Management System in C language which can:

- Store book details (ID, title, author, quantity)
- Add new books
- Display all books
- Search a book by ID
- Issue a book (decrease available quantity)
- Return a book (increase available quantity)

The data should be stored in a file so that it is not lost when the program is closed.

Algorithm

Overall Algorithm (Main Menu)

1. Start
2. Create a file (if not already present) to store book records (e.g. library.dat).
3. Display main menu:
 1. Add Book
 2. Display All Books
 3. Search Book by ID
 4. Issue Book
 5. Return Book
 6. Exit
4. Read user choice.
5. If choice = 1, call Add Book module.
6. If choice = 2, call Display Books module.
7. If choice = 3, call Search Book module.
8. If choice = 4, call Issue Book module.
9. If choice = 5, call Return Book module.
10. If choice = 6, Exit the program.
11. After executing any option (1–5), go back to Step 3.
12. Stop.

Algorithm to Add Book

1. Start
2. Open library.dat file in append or append-binary mode.
3. Input book details: Book ID, Title, Author, Quantity.
4. Write the book record into the file.
5. Close the file.
6. Display message: “Book added successfully.”
7. Stop.

Algorithm to Display All Books

1. Start
2. Open library.dat file in read mode.
3. If file does not exist, show message “No records found” and Stop.
4. Read each book record in a loop until end of file.
5. For each book, print Book ID, Title, Author, Quantity.
6. Close the file.

Algorithm to Search Book by ID

1. Start
2. Input the Book ID to search.
3. Open library.dat file in read mode.
4. Set a flag variable found = 0.
5. Read each record in a loop:
 - If Book ID matches:
 - Set found = 1.
 - Display book details.
6. After loop, if found == 0, display “Book not found.”
7. Close the file.
8. Stop.

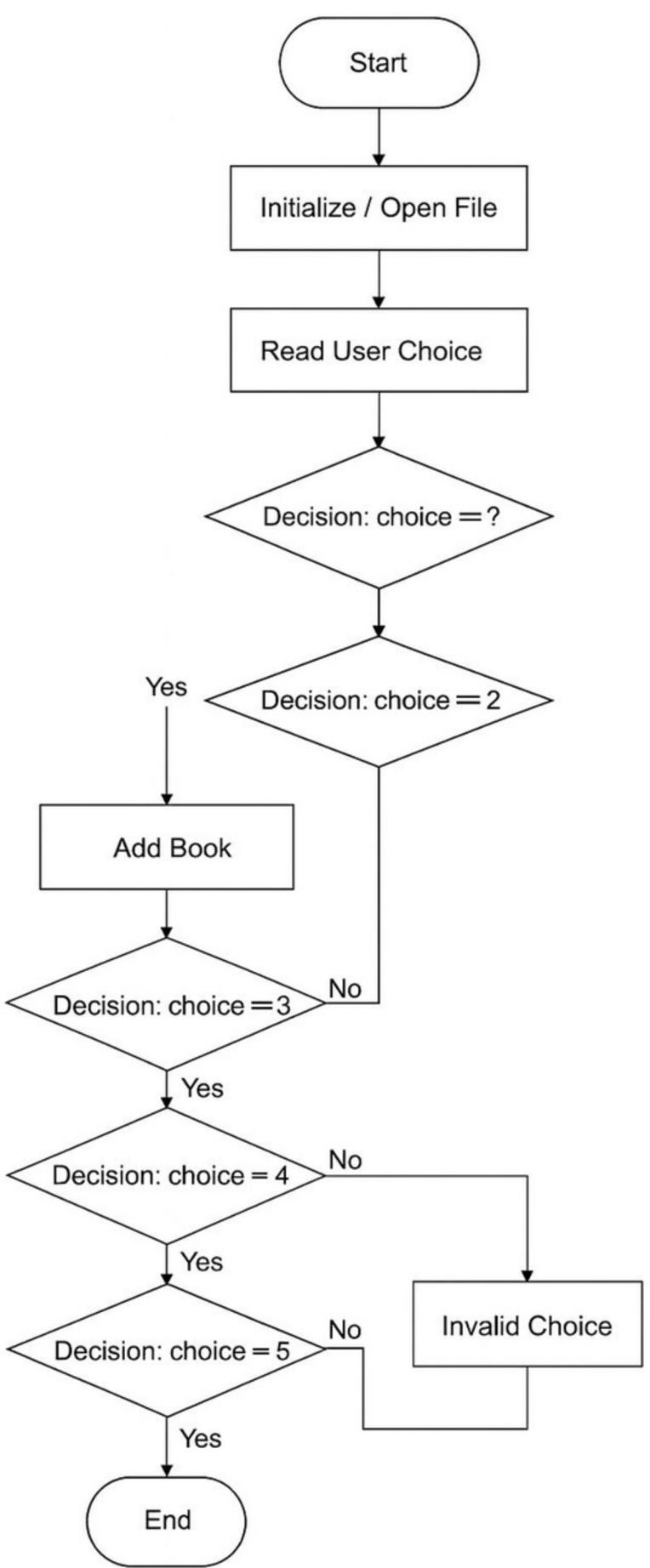
Algorithm to Issue Book

1. Start
2. Input Book ID to issue.
3. Open library.dat file in read mode and a temporary file (e.g. temp.dat) in write mode.
 4. Set found = 0.
 5. For each record in library.dat:
 - If Book ID matches:
 - Set found = 1.
 - If Quantity > 0:
 - Decrease Quantity by 1.
 - Display “Book issued successfully.”
 - Else display “Book not available.”
 - Write the (possibly updated) record into temp.dat.
 6. Close both files.
 7. Replace library.dat with temp.dat.
 8. If found == 0, display “Book not found.”
 9. Stop.

Algorithm to Return Book

1. Start
2. Input Book ID to return.
3. Open library.dat in read mode and temp.dat in write mode.
 4. Set found = 0.
 5. For each record read:
 - If Book ID matches:
Increase Quantity by 1.
 - Set found = 1.
 - Write record to temp.dat.
 6. Close files.
 7. Replace library.dat with temp.dat.
 8. If found == 0, display “Book not found.”
 9. Stop.

Flowchart



Problems Faced by Our Group

While doing this project, our group faced the following problems:

1. Understanding File Handling in C

We were confused between text file and binary file.

We also faced errors while using fopen, fread, fwrite and fclose.

2. Updating Records in File

Directly modifying a record inside the file was difficult.

We solved it by using a temporary file (temp.dat), copying all records, modifying the required record, and then renaming the file.

3. Handling String Inputs

Using scanf for strings with spaces (for book title/author) was giving issues.

We learned to use fgets and clear the input buffer.

4. Logical Errors

Sometimes the quantity did not change correctly after issuing/returning.

We debugged by printing values in between and rechecking conditions.

5. Team Coordination

Initially, work division was not proper.

Later, we divided task as:

Member 1: Add and Display module

Member 2: Search module

Slip of Hold (Issue Slip)

In library, when we issue a book to a student, we usually generate a small Issue Slip / Hold Slip.

6.1 Format of Slip

You can show this in the report as a sample:

----- LIBRARY ISSUE SLIP -----

Library Name : [Your College Name]

Student Name : _____

Enrollment No: _____

Book ID : _____

Book Title : _____

Issue Date : _ / _ / _

Due Date : _ / _ / _

Librarian Sign: _____ Student Sign: _____

6.2 Sample Slip Output from Program

If your program prints a slip after issuing a book, sample console output can be:

----- LIBRARY ISSUE SLIP -----

Student Name : Aditi Sharma

Enrollment No: 123456

Book ID : 101

Book Title : Let Us C

Issue Date : 27-11-2025

Due Date : 04-12-2025

You can write that slip can be printed or written on paper and attached with book.

Sample Output

BOOK INFO

- 1. Show All Books
- 2. Show Available Books
- 3. Search Book
- 4. Issue Book
- 5. Return Book
- 0. Back

Enter: 2

AVAILABLE BOOKS:

- 1 | Let Us C | Left: 10
- 2 | Programming in C | Left: 8
- 3 | Data Structures | Left: 5
- 4 | Operating Systems | Left: 6
- 5 | Computer Networks | Left: 4

LIBRARY MANAGEMENT

- 1. Student Info
- 2. Book Info
- 3. Exit

Enter choice: 2

BOOK INFO

- 1. Show All Books
- 2. Show Available Books
- 3. Search Book
- 4. Issue Book
- 5. Return Book
- 0. Back

BOOK INFO

- 1. Show All Books
- 2. Show Available Books
- 3. Search Book
- 4. Issue Book
- 5. Return Book
- 0. Back

Enter: 5

Enter student roll to return book: 101
Book 'Let Us C' returned by Rohan Sharma successfully!

BOOK INFO

- 1. Show All Books
- 2. Show Available Books
- 3. Search Book
- 4. Issue Book
- 5. Return Book
- 0. Back

Enter: 3

Enter book name / sr number to search: Let Us C

Search results:

1 | Let Us C | Total: 10 | Left: 10

BOOK INFO

- 1. Show All Books
- 2. Show Available Books
- 3. Search Book
- 4. Issue Book
- 5. Return Book
- 0. Back

Enter: 4

Enter student roll number: 101

Enter student name: Rohan Sharma

Enter student course: B.Design

Enter book name to issue: Let Us C

Enter days since book was issued: 7

Book issued to Rohan Sharma successfully!

.c File / Header File and Folder Structure

You can mention how you organized your project files.

8.1 Folder Structure

```
LibraryProject/
    |
    ├── main.c      // main program file
    ├── library.h   // function declarations and struct definition
    └── library.c   // function definitions (add, display, search, issue, return)
        └── library.dat // data file (created at runtime)
```

Appendix – Sample C Code

```
#include <stdio.h>

struct Book {
    int id;
    char title[50];
    char author[50];
};

int main() {
    struct Book b;

    // Input book details
    printf("Enter Book ID: ");
    scanf("%d", &b.id);

    printf("Enter Book Title: ");
    scanf(" %[^\n]", b.title); // Accepts string with spaces

    printf("Enter Author Name: ");
    scanf(" %[^\n]", b.author);

    // Display book details
    printf("\n--- Book Details ---\n");
    printf("Book ID : %d\n", b.id);
    printf("Title   : %s\n", b.title);
    printf("Author  : %s\n", b.author);

    return 0;
}
```

Thank You