Digital Library Management System

Melbin K Thomas

Roll Number:55

Course Name: C Programming

Date:July 17,2024

Introduction

- Brief Overview of the Project

This project involves the creation of a digital library management system in C. The system allows the user to manage a catalog of books, track the availability status of each book, and maintain a history of user borrowing activities.

- Problem Statement

Managing a library manually can be cumbersome and prone to errors. This project aims to automate the library management process, ensuring efficient tracking of books and user transactions.

- Objective

The objective of this project is to develop a C program that:

Stores and manages book details.

Tracks the availability status of each book.

Records and manages user borrowing history.

System Requirements

- Hardware Requirements

Processor: Intel Core i3 or higher

RAM: 4 GB or higher

Storage: 100 MB of free space

- Software Requirements

Operating System: Windows 7 or higher / Linux / macOS

Compiler: GCC or any compatible C compiler

Text Editor: Code::Blocks, Visual Studio Code, or any text editor

Design and Development

- Description of the Program Logic

The program consists of several functions to handle different operations:

Adding books to the library.

Displaying the list of books.

Adding users to the system.

Borrowing books.

Returning books.

Displaying user borrowing history.

The program uses arrays to store book and user details and provides a simple menu-driven interface for interaction.

- Flowchart

Start

|

V

Display Menu

|

V

Get User Choice

|

|--> Add Book

|--> Display Books

|--> Add User

|--> Borrow Book

|--> Return Book

|--> Display User History

|--> Exit

|

V

Perform Selected Operation

|

V

Repeat Until Exit

|

V

End

- **Pseudocode**

```
BEGIN
  WHILE true DO
    DISPLAY Menu
    READ userChoice
    IF userChoice = 1 THEN
      CALL addBook()
    ELSE IF userChoice = 2 THEN
      CALL displayBooks()
    ELSE IF userChoice = 3 THEN
      CALL addUser()
    ELSE IF userChoice = 4 THEN
```

```
      CALL borrowBook()
    ELSE IF userChoice = 5 THEN
      CALL returnBook()
    ELSE IF userChoice = 6 THEN
      CALL displayUserHistory()
    ELSE IF userChoice = 7 THEN
      EXIT
    ELSE
      DISPLAY "Invalid choice!"
    END IF
  END WHILE
END
```

Testing and Results

- Test Cases

Test Case 1: Add Book

Input: Book Title: "C Programming", Author: "Dennis Ritchie"

Expected Result: Book added successfully with ID 1.

Actual Result: Book added successfully with ID 1.

Test Case 2: Add User

Input: User Name: "Alice"

Expected Result: User added successfully with ID 1.

Actual Result: User added successfully with ID 1.

Test Case 3: Borrow Book

Input: User ID: 1, Book ID: 1

Expected Result: Book borrowed successfully.

Actual Result: Book borrowed successfully.

Test Case 4: Return Book

Input: User ID: 1

Expected Result: Book returned successfully.

Actual Result: Book returned successfully.

Output Screenshots or Results

- Discussion of Results

The system successfully handles adding books and users, borrowing and returning books, and displaying user borrowing history. All test cases passed, indicating that the system meets its requirements.

Conclusion

- Summary of the Project

This project demonstrates the creation of a digital library management system using C programming. It successfully automates the tasks of managing book details, tracking availability status, and recording user borrowing history.

- Future Enhancements

Implementing file storage for persistent data.

Enhancing the user interface.

Adding more advanced search and sorting capabilities.

References

"The C Programming Language" by Brian W. Kernighan and Dennis M. Ritchie

Online resources and tutorials on C programming

Appendices

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

#define MAX_BOOKS 100
#define MAX_USERS 50

typedef struct {
    int id;
    char title[50];
    char author[50];
    int available; // 1 if available, 0 if borrowed
} Book;
```

```
typedef struct {
    int userId;
    char userName[50];
    int borrowedBookId;
} User;

Book library[MAX_BOOKS];
User users[MAX_USERS];
int bookCount = 0;
int userCount = 0;

void addBook() {
    if (bookCount < MAX_BOOKS) {
        library[bookCount].id = bookCount + 1;
        printf("Enter book title: ");
        scanf(" %[^\n]", library[bookCount].title);
        printf("Enter book author: ");
        scanf(" %[^\n]", library[bookCount].author);
        library[bookCount].available = 1;
        bookCount++;
        printf("Book added successfully!\n");
    } else {
        printf("Library is full!\n");
    }
}
```

```
void displayBooks() {
    for (int i = 0; i < bookCount; i++) {
        printf("Book ID: %d\n", library[i].id);
        printf("Title: %s\n", library[i].title);
        printf("Author: %s\n", library[i].author);
        printf("Availability: %s\n\n", library[i].available ? "Available" : "Borrowed");
    }
}

void addUser() {
    if (userCount < MAX_USERS) {
        users[userCount].userId = userCount + 1;
        printf("Enter user name: ");
        scanf(" %[^\n]", users[userCount].userName);
        users[userCount].borrowedBookId = -1;
        userCount++;
        printf("User added successfully!\n");
    } else {
        printf("User limit reached!\n");
    }
}

void borrowBook() {
    int userId, bookId;
    printf("Enter user ID: ");
    scanf("%d", &userId);
```

```
    printf("Enter book ID: ");
    scanf("%d", &bookId);

    if (userId > 0 && userId <= userCount && bookId > 0 && bookId <= bookCount) {
        if (library[bookId - 1].available && users[userId - 1].borrowedBookId == -1) {
            library[bookId - 1].available = 0;
            users[userId - 1].borrowedBookId = bookId;
            printf("Book borrowed successfully!\n");
        } else {
            printf("Book is not available or user already borrowed a book.\n");
        }
    } else {
        printf("Invalid user ID or book ID.\n");
    }
}

void returnBook() {
    int userId;
    printf("Enter user ID: ");
    scanf("%d", &userId);

    if (userId > 0 && userId <= userCount) {
        if (users[userId - 1].borrowedBookId != -1) {
            int bookId = users[userId - 1].borrowedBookId;
            library[bookId - 1].available = 1;
            users[userId - 1].borrowedBookId = -1;
```

```
            printf("Book returned successfully!\n");
        } else {
            printf("User has not borrowed any book.\n");
        }
    } else {
        printf("Invalid user ID.\n");
    }
}

void displayUserHistory() {
    for (int i = 0; i < userCount; i++) {
        printf("User ID: %d\n", users[i].userId);
        printf("Name: %s\n", users[i].userName);
        if (users[i].borrowedBookId != -1) {
            printf("Borrowed Book ID: %d\n\n", users[i].borrowedBookId);
        } else {
            printf("No borrowed books\n\n");
        }
    }
}

int main() {
    int choice;
    while (1) {
        printf("Library Management System\n");
        printf("1. Add Book\n");
```

```
printf("2. Display Books\n");
printf("3. Add User\n");
printf("4. Borrow Book\n");
printf("5. Return Book\n");
printf("6. Display User History\n");
printf("7. Exit\n");
printf("Enter your choice: ");
scanf("%d", &choice);

switch (choice) {
  case 1:
    addBook();
    break;
  case 2:
    displayBooks();
    break;
  case 3:
    addUser();
    break;
  case 4:
    borrowBook();
    break;
  case 5:
    returnBook();
    break;
  case 6:
```

```
    displayUserHistory();
    break;
case 7:
    exit(0);
Default:   printf("Invalid choice!\n");
```