**LIBRARY MANAGEMENT SYSTEM**

Group 9

Fatima (TCT-001)

M.Tayyab (TCT-009)

Nimrata (TCT-016)

Deeya kumari (TCT-019)

Charan Kumar (TCT-026)

Programming fundamentals (CT-175)

Muhammad Abbas

January,15,2024

**INTRODUCTION**

This Library Management System implemented in C programming language. It includes functionalities for both users and administrators, allowing users to read, borrow, or buy books, while admin can add books, change passwords, and view details of borrowed books and sale books. The system operates through a console-based interface.

**Approach**

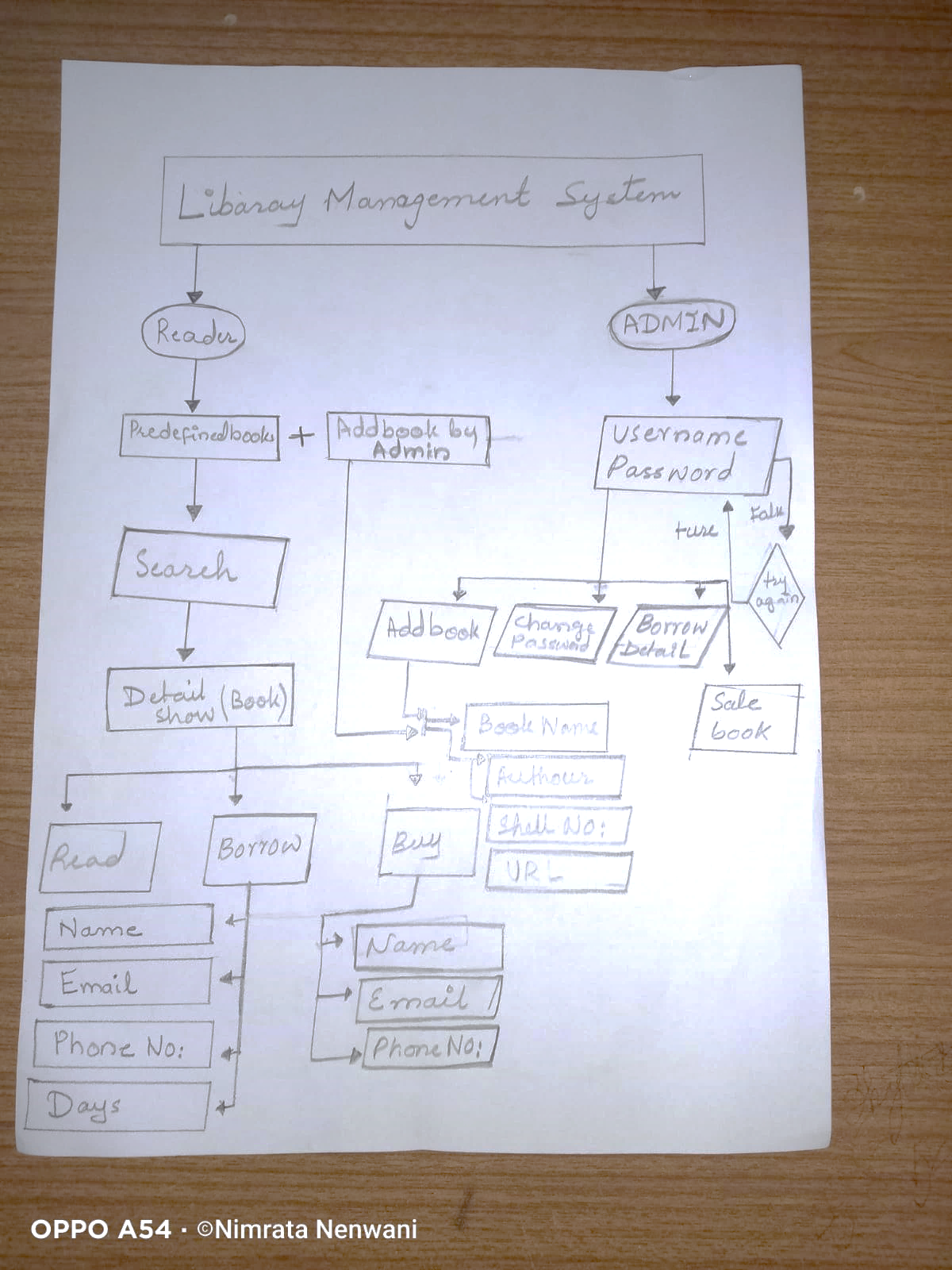
The system uses a modular approach with various functions like main, readBook,addBook,displayBookDetails,bookAction,changePassword, and valid\_days. The program employs switch-case statements for user interaction and book-related actions.

**Project Implementation**

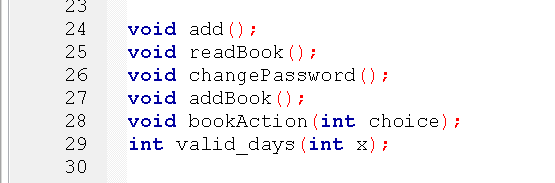
* The program maintains an array of pre-defined books and supports the addition of new books by the admin.
* User authentication is required for admin access, and the system prompts users to choose between user and admin modes.
* Users can interact with the system to read, borrow, or buy books, with associated actions such as displaying book details.

**Key Algorithm**

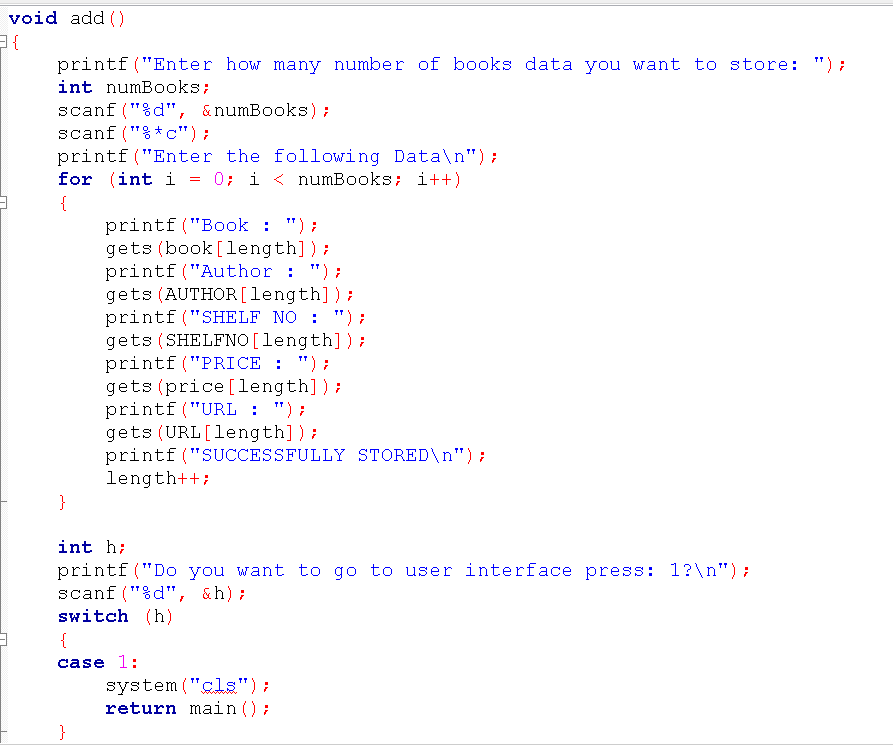
The main algorithm involves switch-case statements to handle different user choices and actions. Our main function is very small and it is mostly depending on the user defined functions like readBook, displayBookDetails, and bookAction are crucial for user interactions and managing book-related operations

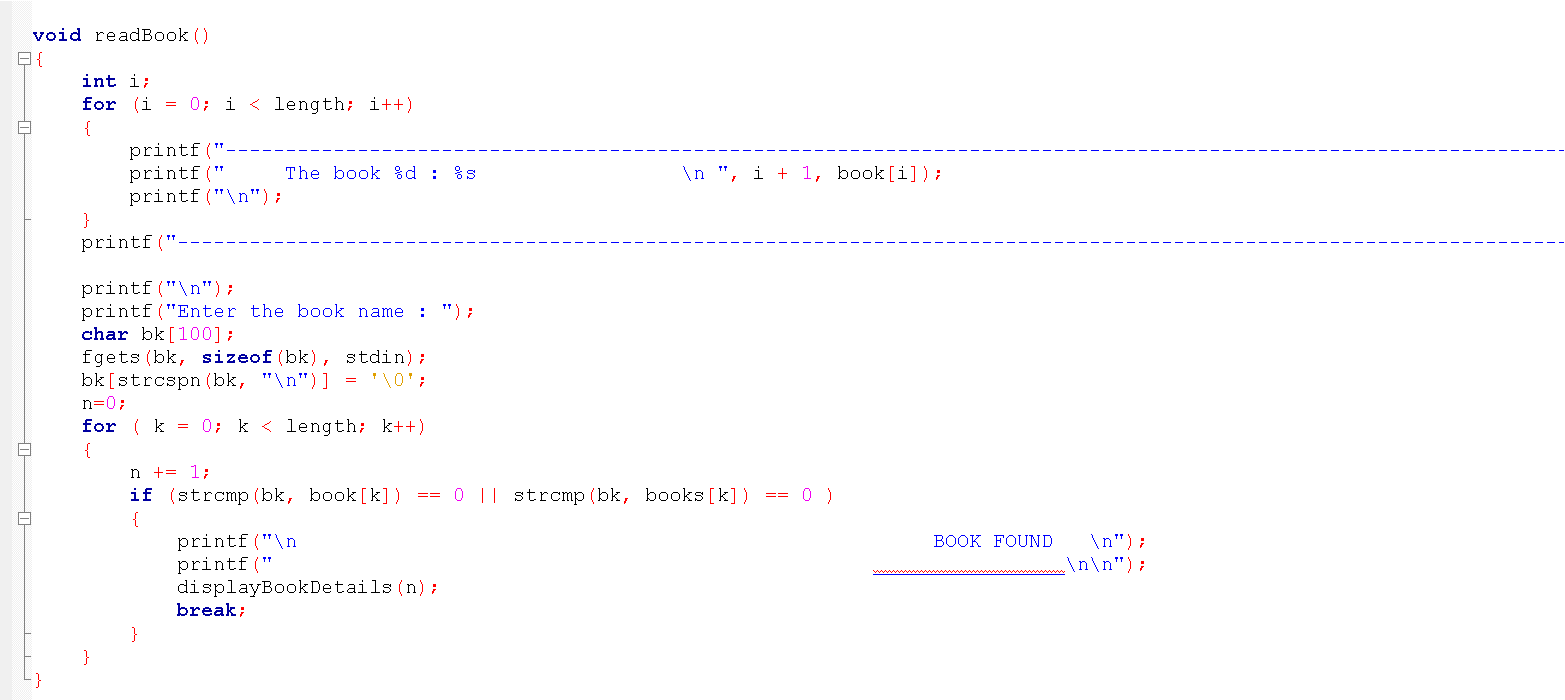
**FLOWCHART**

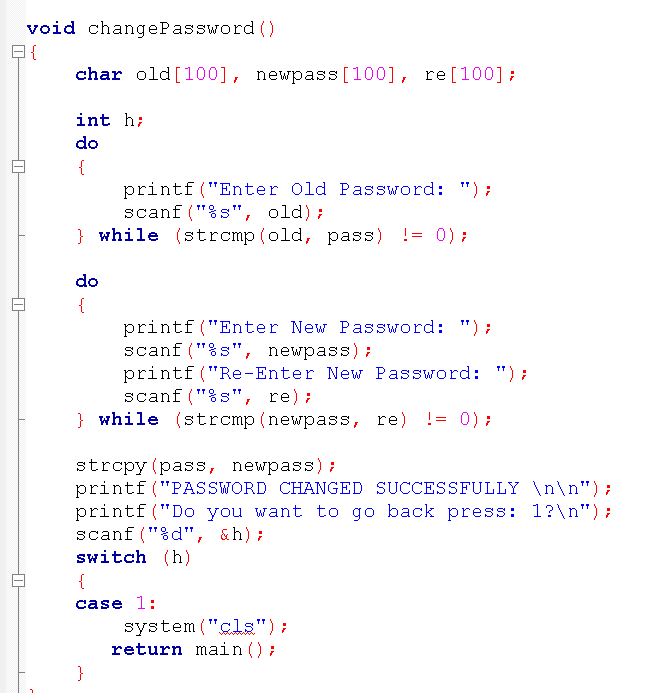
**SNIPPETS**

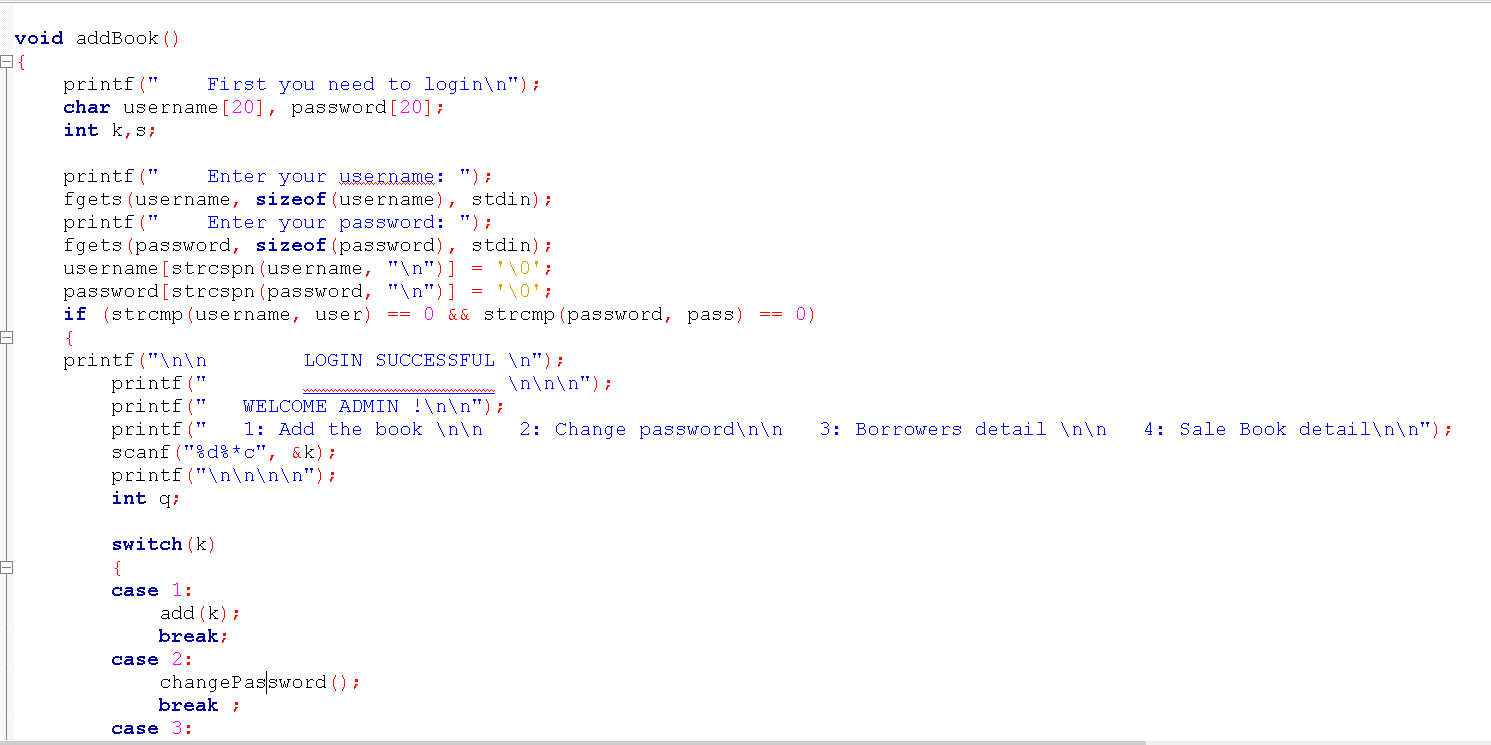


1. Function prototypes:

****

****

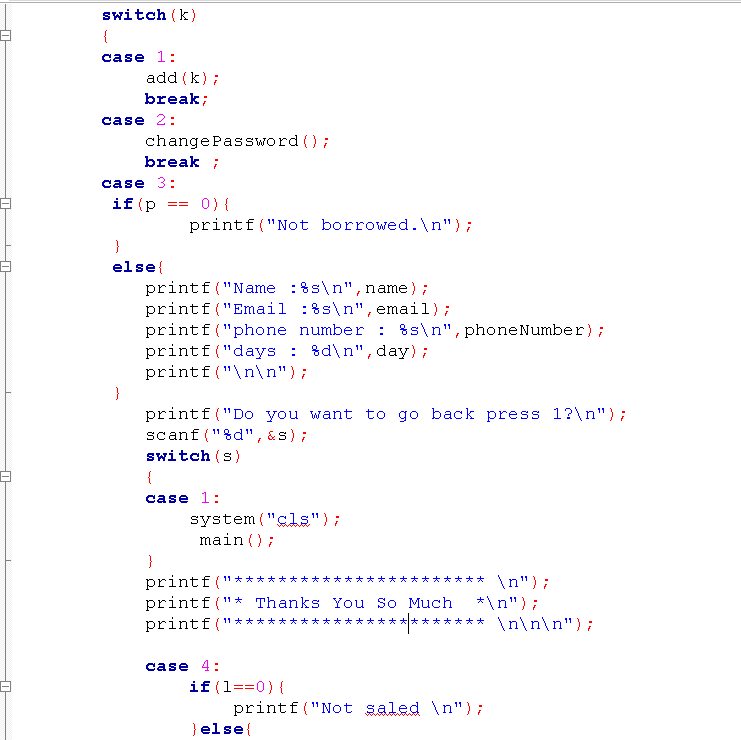
****

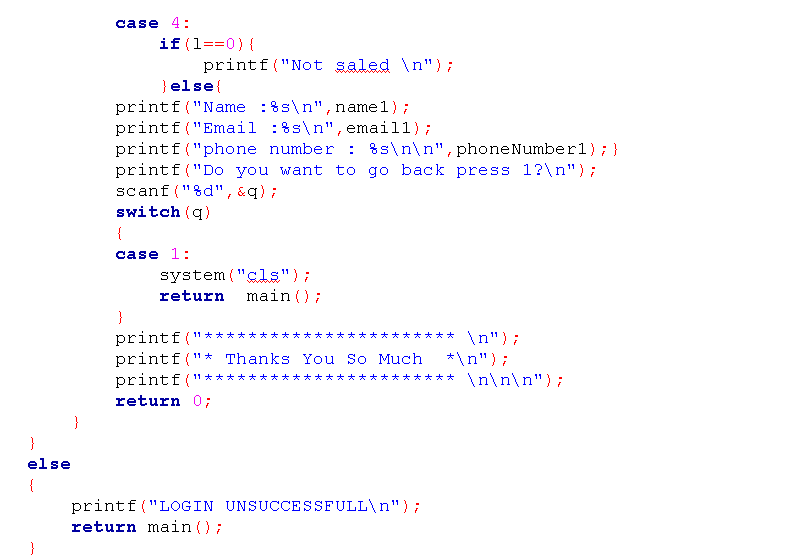
****

1. Switch Case for ‘displayBookDetails’ Function

****

3. Switch Case in ‘addBook’ Function (admin)

****

****

**Result**

The system successfully performs basic library operations, such as displaying book details, allowing users to read or borrow books, and enabling admin to add new books or change passwords.

**Discussion**

While the system demonstrates functionality, dynamic memory allocation for book storage, and comprehensive user interfaces. We only see the data of only one borrower and one sales book data. Additionally, the code could benefit from Improvements in terms of code structure and organization**.**

**Limitations and Challenges**

* It lacks certain features like proper error handling.
* The absence of dynamic memory allocation for book storage limits scalability.
* We Can’t delete any book we have already stored.
* We can see the data of only one borrower and only one Sales book data at a time.
* Code challenges could include implementing proper error handling.
* Incorporating dynamic memory allocation for scalability, and refining user interfaces for a more intuitive experience.
* Code improvement so that we can see the data of more then one borrowers data and sales book data.
* Consideration of data structures like linked lists or arrays for book management could enhance efficiency.

**Conclusion**

The provided code offers a foundation for a simple library management system but requires refinements for improved functionality, security, and user experience. Further development could involve incorporating data structures for book storage, implementing dynamic memory allocation, and enhancing the user interface**.**

**Appendix**

The system exhibits sophisticated user interfaces, dynamic memory allocation for book storage, and functionality. Only one borrower's data and one sales book's worth of data are visible to us. Further improvements to the code's organization and structure would be beneficial**.**