Library Management System with Privacy (C++ OOP Project)

1. Introduction

This project implements a Library Management System in C++ using Object-Oriented Programming (OOP) principles. It supports user registration, login with encrypted credentials, book issue/return functionality, and persistence of data across sessions. The system is designed for security, scalability, and modularity.

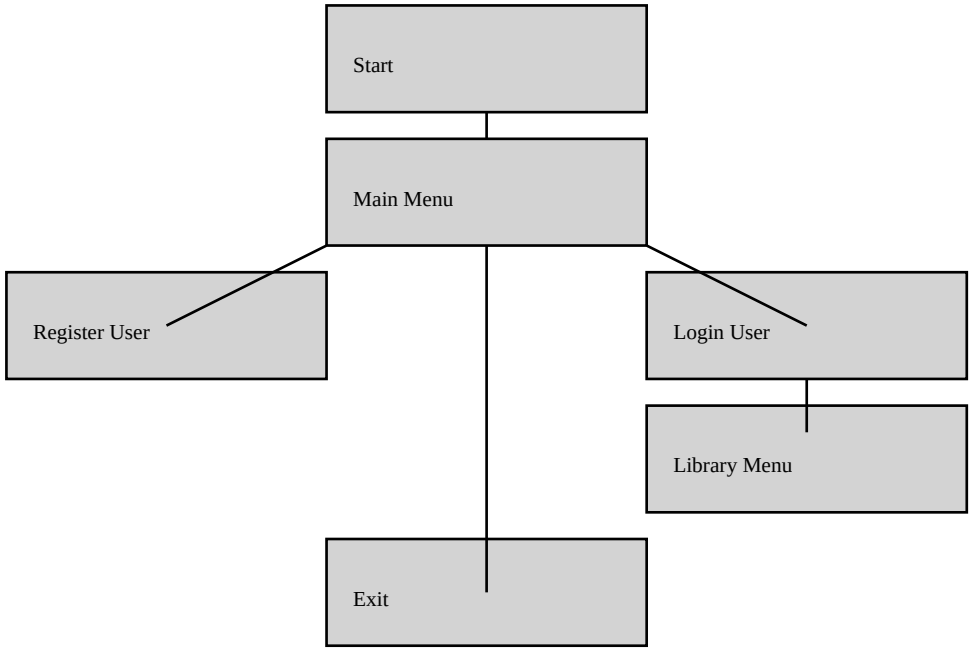
2. Objectives

- Provide a secure library system with user authentication.
- Manage books with issue/return and status tracking.
- Demonstrate OOP principles: encapsulation, inheritance, abstraction.
- Ensure persistence of user data using file handling.

3. System Architecture

The system is modular and divided into four main components:
1. User & Privacy: Handles registration, login, and credential encryption.
2. Library: Manages books, their availability, and issue/return operations.
3. Manager: Integrates user and library modules, acts as a controller.
4. File Handling: Stores and retrieves user credentials persistently.

4. Flow Diagram



5. Methodology

Step 1: User registers with username and password. Password is encrypted using Caesar cipher.
Step 2: Credentials are stored in 'users.txt' for persistence.
Step 3: On login, entered credentials are matched with encrypted records.
Step 4: User accesses the Library Menu (Show Books, Issue, Return).
Step 5: Book status is updated accordingly.
Step 6: User can logout or exit the system.

6. OOP Concepts Used

- Encapsulation: Private members in classes like UserData, Book.
- Abstraction: Manager hides internal complexity of user and library handling.
- Modularity: Code divided into classes (UserData, Encryptor, Book, Library, Manager).
- File Handling: Used to persist user data in text files.

7. Working Flow

1. Program starts with Main Menu.
2. User chooses to Register, Login, or Exit.
3. On Login, Library Menu is displayed.
4. Books can be shown, issued, or returned.
5. User can logout to return to Main Menu.
6. Program ends when Exit option is chosen.

8. Advantages

- Simple, modular design.
- Secure password handling with encryption.
- Scalable: more features can be added (fine system, due dates, etc.).
- Persistent storage with text file handling.

9. Future Scope

- Replace Caesar cipher with stronger encryption (e.g., AES).
- Add GUI for better usability.
- Integrate database instead of flat text file.
- Support for multiple copies of books and due-date management.

10. Conclusion

This project demonstrates the implementation of a secure Library Management System using Object-Oriented Programming in C++. It highlights modular design, secure user handling, and persistence through file handling. The project can be extended into a full-fledged application with advanced features in the future.