# Library Management System Documentation

# Group members:

1) Hadi Kashif F2023266412

2) Halima Amjad F2023266494

3) Hafiz Abdul Moiz F2023266477

4) Daniyal Saqib F2023266481.

## 1. Introduction

The Library Management System is a console-based C++ program designed to manage books and members in a library efficiently. It provides functionalities such as member registration, book management, login authentication, book issuance, and logout.

## 2. Problem Statement

Managing a library manually is inefficient and time-consuming. This system aims to digitalize the process, making book tracking and member management more systematic and user-friendly.

## 3. Objective

The primary objective is to develop a structured and efficient library management system that ensures ease of access for both librarians and members while maintaining data security and accuracy.

## 4. Feasibility

- \*\*Technical Feasibility\*\*: Implemented in C++, which is widely used and supports object-oriented programming.  
- \*\*Operational Feasibility\*\*: Reduces manual workload and enhances operational efficiency.  
- \*\*Economic Feasibility\*\*: Developed as a standalone application with minimal resource requirements.

## 5. Methodology

The system follows a structured development methodology:  
- Requirement gathering and analysis  
- System design (Flowcharts and Class Diagrams)  
- Implementation using C++  
- Testing and validation  
- Deployment and maintenance

## 6. Key Features

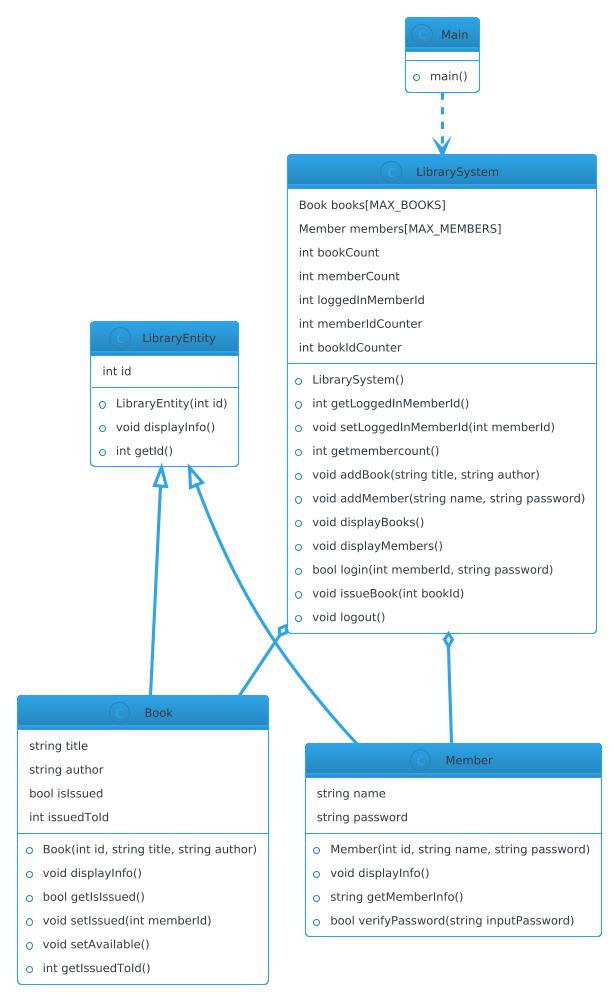
- Member registration with password authentication.  
- Book addition and display.  
- Secure login system with password verification.  
- Issuing books to members.  
- Logout and exit options.

## 7. System Components and Concepts

- \*\*Encapsulation\*\*: Data members of classes are private to ensure secure data handling.  
- \*\*Inheritance\*\*: `Book` and `Member` classes inherit from `LibraryEntity`.  
- \*\*Polymorphism\*\*: Virtual functions allow different implementations in derived classes.  
- \*\*Abstraction\*\*: `LibraryEntity` provides an abstract base for `Book` and `Member`.

## 8. Working

1. \*\*User Registration\*\*: New members can register with a unique ID and password.  
2. \*\*Login System\*\*: Members must log in using their credentials to access system functionalities.  
3. \*\*Book Management\*\*: The librarian can add and display books.  
4. \*\*Issuing Books\*\*: Members can issue books, and the system tracks issued books.  
5. \*\*Error Handling\*\*: Input validation and error messages enhance usability.  
6. \*\*Logout\*\*: Users can securely log out after their session.

UML: 

## 9. Results

- Successfully implemented core functionalities: user authentication, book management, and book issuance.  
- Improved efficiency in managing library operations.  
- Enhanced user experience with a structured and error-free approach.

## 10. Future Scope

- Implement database connectivity.  
- Add a graphical user interface (GUI).  
- Improve error handling and user feedback.  
- Introduce book return functionality and overdue penalties.

## 11. Conclusion

This system provides basic library functionalities using OOP principles. It ensures secure login, efficient book management, and structured data handling, making it a strong foundation for further enhancements.