

Project report

ON

**Library Management System**

As a part of

OOCP Practical

Subject code: CE0316

Submitted by

Student Name

Gautam .S. Makwana

IU2341230099

BACHELOR OF TECHNOLOGY

In

COMPUTER SCIENCE ENGINEERING

INSTITUTE OF TECHNOLOGY AND ENGINEERING

INDUS UNIVERSITY CAMPUS, RANCHARDA, VIA-THALTEJ AHMEDABAD-382115, GUJARAT, INDIA

WEB: [www.indusuni.ac.in](http://www.indusuni.ac.in)

* Project Title:- **Library Management System**

**Scenario**: Design a system to manage book borrowing in a library.  
**Classes**:

* + **Book**
    - Data Members: bookID, title, author, isIssued
    - Member Functions: issueBook(), returnBook(), displayDetails()
  + **Member**
    - Data Members: memberID, name, bookIssued
    - Member Functions: borrowBook(), returnBook(), displayMember() **Description**: Simulate the borrowing and returning process of books.
* **Overview:-**

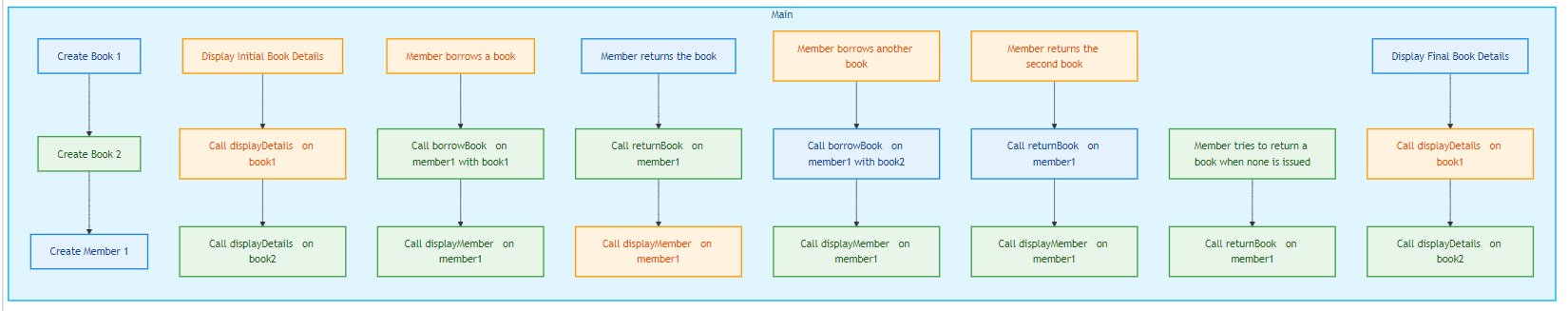
The Library Management System is designed to efficiently manage the borrowing and returning of books in a library setting. The system comprises two primary classes: **Book** and **Member**, each encapsulating relevant data and functionality.

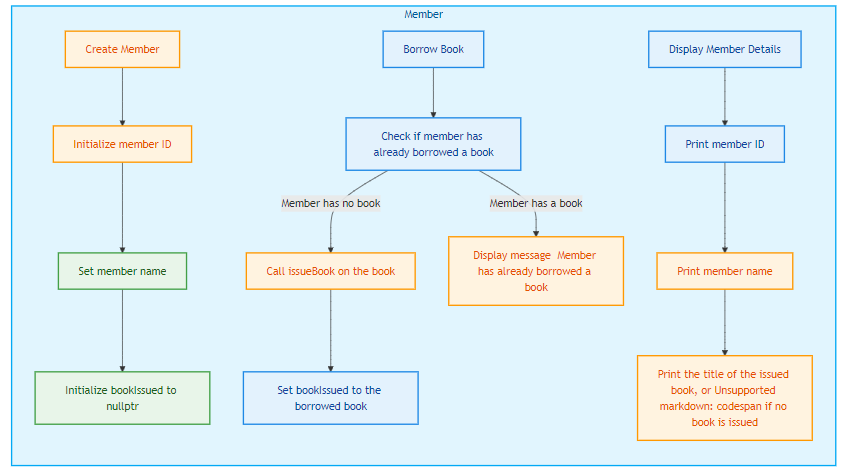
**Class: Book**

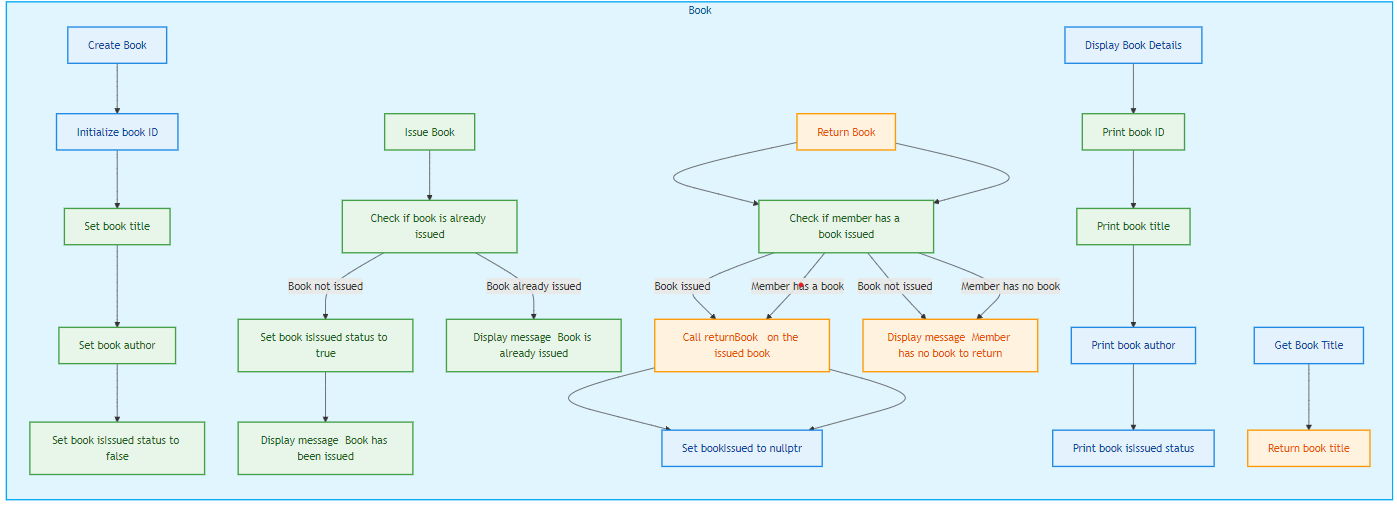
* **Data Members**:
  + **bookID**: A unique identifier for each book.
  + **title**: The title of the book.
  + **author**: The author of the book.
  + **isIssued**: A boolean flag indicating whether the book is currently issued to a member.
* **Member Functions**:
  + **issueBook()**: This function allows a book to be issued. It checks the isIssued status and updates it accordingly, providing feedback on whether the operation was successful.
  + **returnBook()**: This function processes the return of a book, updating the isIssued status to reflect that the book is now available.
  + **displayDetails()**: This function displays the details of the book, including its ID, title, author, and issued status.

**Class: Member**

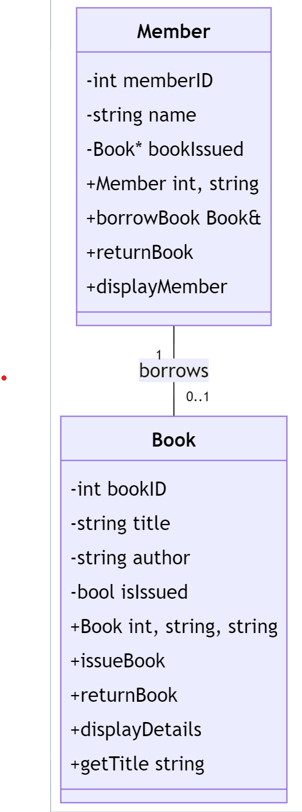
* **Data Members**:
  + **memberID**: A unique identifier for each library member.
  + **name**: The name of the member.
  + **bookIssued**: A pointer/reference to the book currently issued to the member (if any).
* **Member Functions**:
  + **borrowBook(Book &book)**: This function allows a member to borrow a book, checking if the member already has a book issued and whether the selected book is available.
  + **returnBook()**: This function allows the member to return the borrowed book, updating both the member's record and the book's status.
  + **displayMember()**: This function displays the member's details, including their ID, name, and the title of the book currently issued to them.
* **Flow Chart:-**



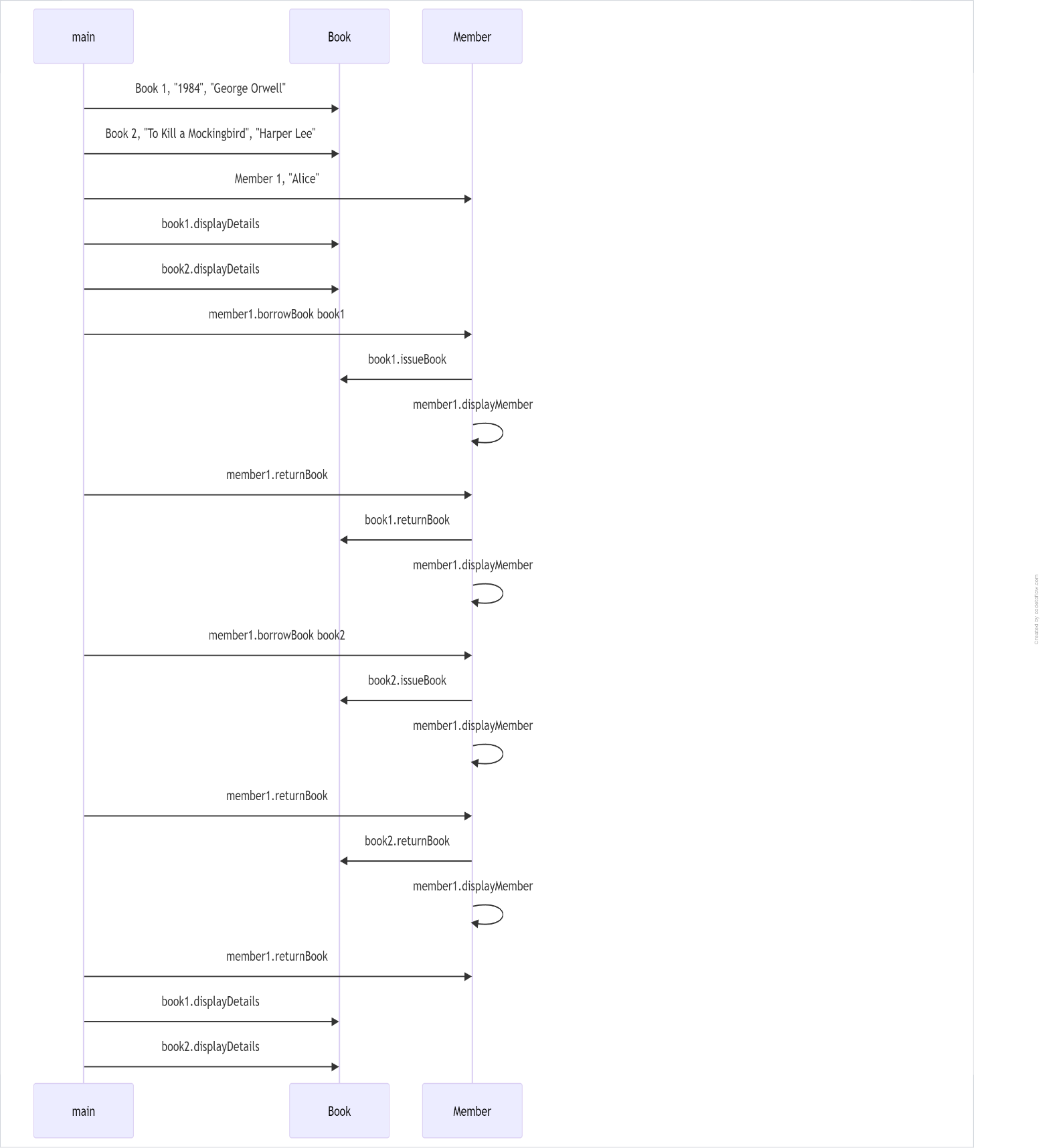




* **Class Diagram :-**

****

* **Sequence Diagram:-**

****

* **Source Code:-**

**#include <iostream>**

**#include <string>**

**#include<conio.h>**

**class Book {**

**private:**

**int bookID;**

**std::string title;**

**std::string author;**

**bool isIssued;**

**public:**

**// Constructor**

**Book(int id, const std::string& t, const std::string& a)**

**: bookID(id), title(t), author(a), isIssued(false) {}**

**// Issue the book**

**void issueBook() {**

**if (!isIssued) {**

**isIssued = true;**

**std::cout << "Book \"" << title << "\" has been issued." << std::endl;**

**} else {**

**std::cout << "Book \"" << title << "\" is already issued." << std::endl;**

**}**

**}**

**// Return the book**

**void returnBook() {**

**if (isIssued) {**

**isIssued = false;**

**std::cout << "Book \"" << title << "\" has been returned." << std::endl;**

**} else {**

**std::cout << "Book \"" << title << "\" is not issued." << std::endl;**

**}**

**}**

**// Display book details**

**void displayDetails() const {**

**std::cout << "Book ID: " << bookID << ", Title: \"" << title << "\", Author: \"" << author**

**<< "\", Issued: " << (isIssued ? "Yes" : "No") << std::endl;**

**}**

**// Getter for title (for member display)**

**const std::string& getTitle() const {**

**return title;**

**}**

**};**

**class Member {**

**private:**

**int memberID;**

**std::string name;**

**Book\* bookIssued;**

**public:**

**// Constructor**

**Member(int id, const std::string& n) : memberID(id), name(n), bookIssued(nullptr) {}**

**// Borrow a book**

**void borrowBook(Book& book) {**

**if (bookIssued == nullptr) {**

**book.issueBook();**

**bookIssued = &book;**

**} else {**

**std::cout << name << " has already borrowed a book." << std::endl;**

**}**

**}**

**// Return the borrowed book**

**void returnBook() {**

**if (bookIssued != nullptr) {**

**bookIssued->returnBook();**

**bookIssued = nullptr;**

**} else {**

**std::cout << name << " has no book to return." << std::endl;**

**}**

**}**

**// Display member details**

**void displayMember() const {**

**std::cout << "Member ID: " << memberID << ", Name: \"" << name << "\", Book Issued: "**

**<< (bookIssued ? bookIssued->getTitle() : "None") << std::endl;**

**}**

**};**

**int main() {**

**// Create books**

**Book book1(1, "1984", "George Orwell");**

**Book book2(2, "To Kill a Mockingbird", "Harper Lee");**

**// Create a member**

**Member member1(1, "Alice");**

**// Display book details**

**std::cout << "Initial Book Details:" << std::endl;**

**book1.displayDetails();**

**book2.displayDetails();**

**std::cout << std::endl;**

**// Member borrows a book**

**std::cout << "Alice borrows a book:" << std::endl;**

**member1.borrowBook(book1);**

**member1.displayMember();**

**std::cout << std::endl;**

**// Return the book**

**std::cout << "Alice returns the book:" << std::endl;**

**member1.returnBook();**

**member1.displayMember();**

**std::cout << std::endl;**

**// Attempt to borrow another book**

**std::cout << "Alice borrows another book:" << std::endl;**

**member1.borrowBook(book2);**

**member1.displayMember();**

**std::cout << std::endl;**

**// Return the second book**

**std::cout << "Alice returns the second book:" << std::endl;**

**member1.returnBook();**

**member1.displayMember();**

**std::cout << std::endl;**

**// Attempt to return a book when none is issued**

**std::cout << "Alice tries to return a book again:" << std::endl;**

**member1.returnBook();**

**std::cout << std::endl;**

**// Display final book details**

**std::cout << "Final Book Details:" << std::endl;**

**book1.displayDetails();**

**book2.displayDetails();**

**return 0;**

**}**

* + **Output:-**

**Initial Book Details:**

**Book ID: 1, Title: "1984", Author: "George Orwell", Issued: No**

**Book ID: 2, Title: "To Kill a Mockingbird", Author: "Harper Lee", Issued: No**

**Alice borrows a book:**

**Book "1984" has been issued.**

**Member ID: 1, Name: "Alice", Book Issued: 1984**

**Alice returns the book:**

**Book "1984" has been returned.**

**Member ID: 1, Name: "Alice", Book Issued: None**

**Alice borrows another book:**

**Book "To Kill a Mockingbird" has been issued.**

**Member ID: 1, Name: "Alice", Book Issued: To Kill a Mockingbird**

**Alice returns the second book:**

**Book "To Kill a Mockingbird" has been returned.**

**Member ID: 1, Name: "Alice", Book Issued: None**

**Alice tries to return a book again:**

**Alice has no book to return.**

**Final Book Details:**

**Book ID: 1, Title: "1984", Author: "George Orwell", Issued: No**

**Book ID: 2, Title: "To Kill a Mockingbird", Author: "Harper Lee", Issued: No**

* + **Code Explanation:-**

1. The class diagram represents two main classes: Book and Member.

2. The Book class has the following attributes:

* bookID: an integer representing the unique identifier of the book.
* title: a string representing the title of the book.
* author: a string representing the author of the book.
* isIssued: a boolean flag indicating whether the book is currently issued or not.

The Book class also has the following methods:

* Book(int, string, string): a constructor that initializes the book's attributes.
* issueBook(): a method to issue the book.
* returnBook(): a method to return the book.
* displayDetails(): a method to display the book's details.
* getTitle(): a getter method to retrieve the book's title.

3. The Member class has the following attributes:

* memberID: an integer representing the unique identifier of the member.
* name: a string representing the name of the member.
* bookIssued: a pointer to a Book object, representing the book currently issued to the member.

The Member class also has the following methods:

* Member(int, string): a constructor that initializes the member's attributes.
* borrowBook(Book&): a method to borrow a book.
* returnBook(): a method to return the borrowed book.
* displayMember(): a method to display the member's details.

4. The relationship between the Member and Book classes is represented by an association, where a Member can borrow a Book. The cardinality of this association is "1" (one member) to "0..1" (zero or one book).

This class diagram accurately represents the static structure of the system described in the provided code, including the classes, their attributes, methods, and the relationships between them.

* + **Conclusion:-**

In conclusion, the Library Management System effectively simulates the borrowing and returning of books, combining functionality and ease of use. Its structured approach serves as a strong foundation for further development, paving the way for more advanced features and capabilities in the future.