CA 3: Experiential Learning

Group Members:

Sr. No.	PRN	Name of Student	Mail id
1.	22070122208	Shubham singh	Shubham.singh.btech2022@sitpune.edu.in
2.	22070122182	Sanjana balaji	Sanjana.balaji.btech2022@sitpune.edu.in
3.	22070122183	Sanvi verma	Sanvi.verma.btech2022@sitpune.edu.in

Problem Statement:

To develop a Library Management System in C++ to manage library items and users. The system should allow users to add and delete library items (such as books, DVDs, and magazines) and users. Users should also be able to check out and return library items.

Explanation:

Library Management System solves the problem of efficiently managing and maintaining a library's collection of items and users. It offers a range of functionality to facilitate the smooth operation of a library. Here are the main problems it addresses:

Item Management**:** The code allows the library to keep track of various types of items, including books, DVDs, and magazines. It helps in storing essential information about each item, such as its unique ID, title, and availability status. This aids librarians in managing and organizing the library's collection more effectively.

User Management: The system enables the creation and management of library users, each with a unique ID and name. This is essential for tracking who has checked out which items and ensuring accountability.

Checkout and return: Library patrons can check out items for reading or viewing, and subsequently return them. The code handles the process of marking items as checked out and available for checkout again upon return. This helps in preventing multiple users from checking out the same item simultaneously.

Displaying Available Items: The system provides the functionality to display available items. This feature is valuable for library patrons who want to know what is currently accessible in the library's collection.

Deletion of Items: The code permits the deletion of items from the library's collection. This feature is useful when items need to be removed permanently from the library's records.

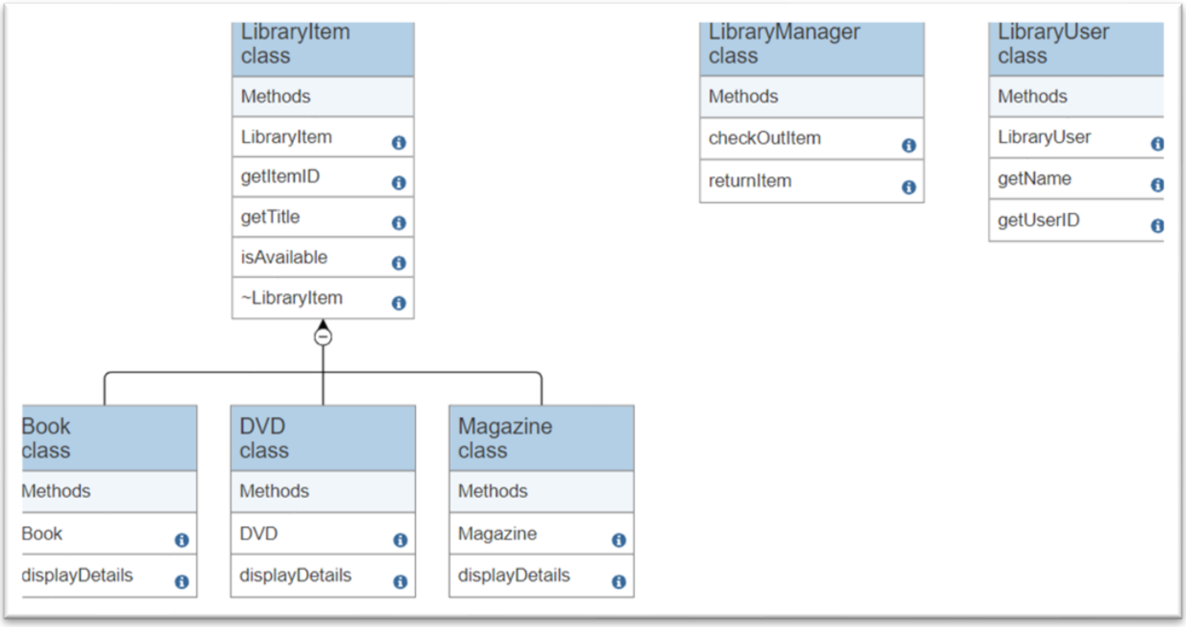
Dynamic Memory Management: The code properly manages dynamic memory for items, ensuring that memory is allocated and deallocated appropriately. This is essential for preventing memory leaks and optimizing resource usage.

Data Validation: The program validates user inputs to ensure that they are within acceptable bounds. For instance, it checks whether a user is trying to check out an available item, ensuring data integrity.

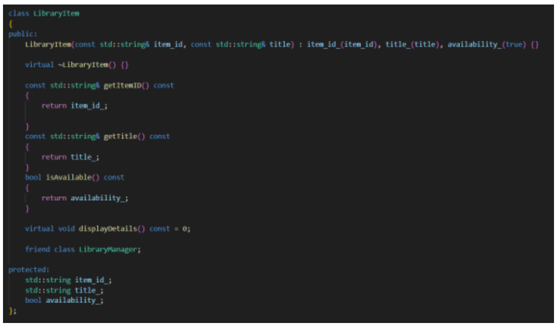
Exit Options: It offers options for exiting the program safely, including the ability to confirm the exit, delete all items and users, or cancel the exit. This is critical for ensuring that the system remains in a consistent state and all resources are cleaned up when exiting.

In summary, this code solves the problems of efficient library item and user management, tracking item availability, and providing a user-friendly interface for librarians and patrons. It helps maintain order in a library and provides a structured way to interact with the library's collection.

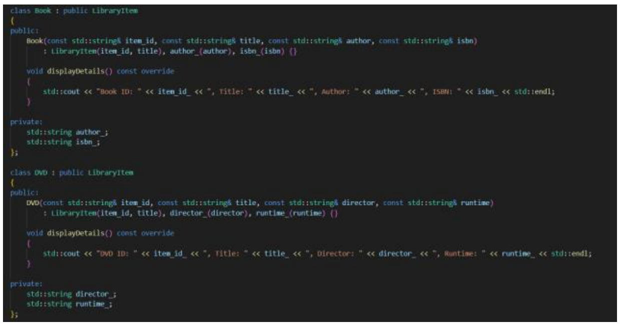
Class Diagram:



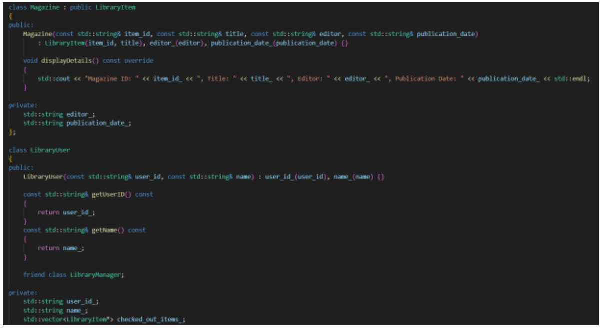
Code snippets:



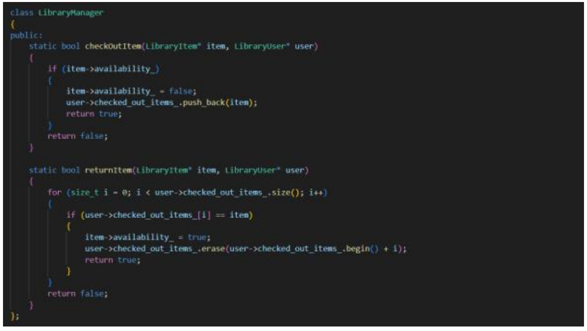
```
class LibraryItem
{
public:
    LibraryItem(const std::string& item_id, const std::string& title) : item_id_(item_id), title_(title), availability_(true) {}

    virtual ~LibraryItem() {}

    const std::string& getItemID() const
    {
        return item_id_;

    }
    const std::string& getTitle() const
    {
        return title_;
    }
    bool isAvailable() const
    {
        return availability_;
    }

    virtual void displayDetails() const = 0;

    friend class LibraryManager;

protected:
    std::string item_id_;
    std::string title_;
    bool availability_;
};
```



```
class Book : public LibraryItem
{
public:
    Book(const std::string& item_id, const std::string& title, const std::string& author, const std::string& isbn)
        : LibraryItem(item_id, title), author_(author), isbn_(isbn) {}

    void displayDetails() const override
    {
        std::cout << "Book ID: " << item_id_ << ", Title: " << title_ << ", Author: " << author_ << ", ISBN: " << isbn_ << std::endl;
    }

private:
    std::string author_;
    std::string isbn_;
};

class DVD : public LibraryItem
{
public:
    DVD(const std::string& item_id, const std::string& title, const std::string& director, const std::string& runtime)
        : LibraryItem(item_id, title), director_(director), runtime_(runtime) {}

    void displayDetails() const override
    {
        std::cout << "DVD ID: " << item_id_ << ", Title: " << title_ << ", Director: " << director_ << ", Runtime: " << runtime_ << std::endl;
    }

private:
    std::string director_;
    std::string runtime_;
};
```

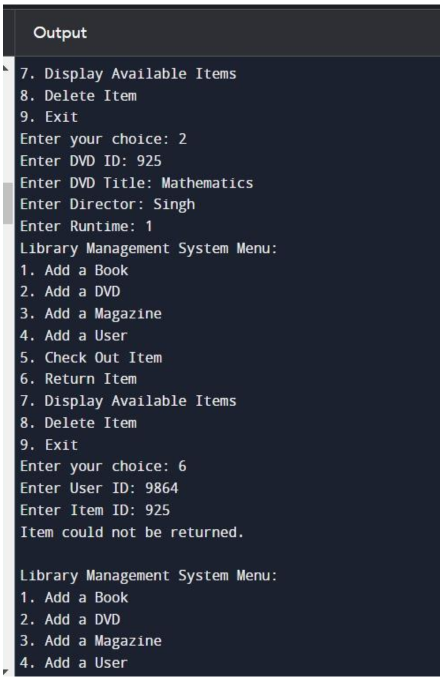


```
class Magazine : public LibraryItem
{
public:
    Magazine(const std::string& item_id, const std::string& title, const std::string& editor, const std::string& publication_date)
        : LibraryItem(item_id, title), editor_(editor), publication_date_(publication_date) {}

    void displayDetails() const override
    {
        std::cout << "Magazine ID: " << item_id_ << ", Title: " << title_ << ", Editor: " << editor_ << ", Publication Date: " << publication_date_ << std::endl;
    }

private:
    std::string editor_;
    std::string publication_date_;
};

class LibraryUser
{
public:
    LibraryUser(const std::string& user_id, const std::string& name) : user_id_(user_id), name_(name) {}

    const std::string& getUserID() const
    {
        return user_id_;
    }
    const std::string& getName() const
    {
        return name_;
    }

    friend class LibraryManager;

private:
    std::string user_id_;
    std::string name_;
    std::vector<LibraryItem*> checked_out_items_;
```



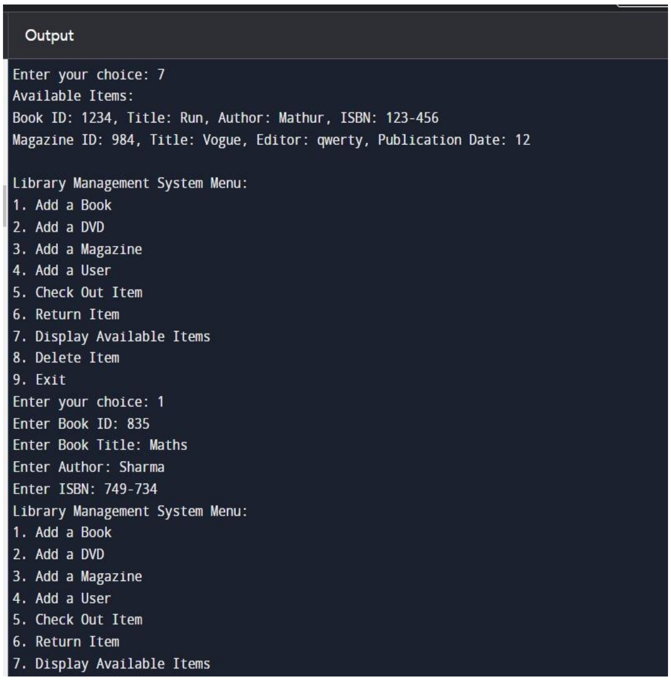
```
class LibraryManager
{
public:
    static bool checkOutItem(LibraryItem* item, LibraryUser* user)
    {
        if (item->availability_)
        {
            item->availability_ = false;
            user->checked_out_items_.push_back(item);
            return true;
        }
        return false;
    }

    static bool returnItem(LibraryItem* item, LibraryUser* user)
    {
        for (size_t i = 0; i < user->checked_out_items_.size(); i++)
        {
            if (user->checked_out_items_[i] == item)
            {
                item->availability_ = true;
                user->checked_out_items_.erase(user->checked_out_items_.begin() + i);
                return true;
            }
        }
        return false;
    }
};
```

Input/Output:



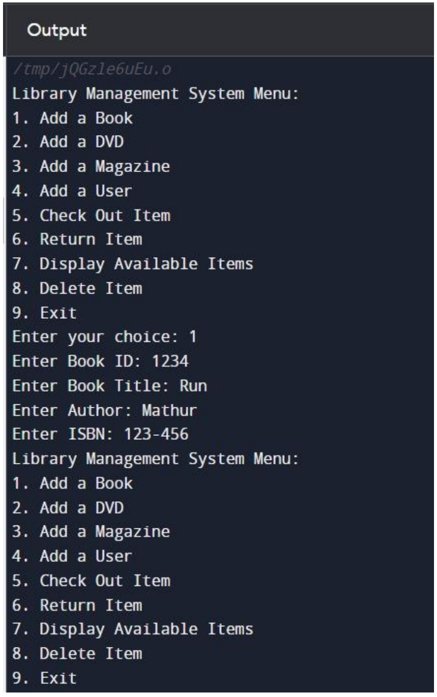
Output

```
7. Display Available Items
8. Delete Item
9. Exit
Enter your choice: 2
Enter DVD ID: 925
Enter DVD Title: Mathematics
Enter Director: Singh
Enter Runtime: 1
Library Management System Menu:
1. Add a Book
2. Add a DVD
3. Add a Magazine
4. Add a User
5. Check Out Item
6. Return Item
7. Display Available Items
8. Delete Item
9. Exit
Enter your choice: 6
Enter User ID: 9864
Enter Item ID: 925
Item could not be returned.

Library Management System Menu:
1. Add a Book
2. Add a DVD
3. Add a Magazine
4. Add a User
```



Output

```
Enter your choice: 7
Available Items:
Book ID: 1234, Title: Run, Author: Mathur, ISBN: 123-456
Magazine ID: 984, Title: Vogue, Editor: qwerty, Publication Date: 12

Library Management System Menu:
1. Add a Book
2. Add a DVD
3. Add a Magazine
4. Add a User
5. Check Out Item
6. Return Item
7. Display Available Items
8. Delete Item
9. Exit
Enter your choice: 1
Enter Book ID: 835
Enter Book Title: Maths
Enter Author: Sharma
Enter ISBN: 749-734
Library Management System Menu:
1. Add a Book
2. Add a DVD
3. Add a Magazine
4. Add a User
5. Check Out Item
6. Return Item
7. Display Available Items
```



Output

```
/tmp/jQGzle6uEu.o
Library Management System Menu:
1. Add a Book
2. Add a DVD
3. Add a Magazine
4. Add a User
5. Check Out Item
6. Return Item
7. Display Available Items
8. Delete Item
9. Exit
Enter your choice: 1
Enter Book ID: 1234
Enter Book Title: Run
Enter Author: Mathur
Enter ISBN: 123-456
Library Management System Menu:
1. Add a Book
2. Add a DVD
3. Add a Magazine
4. Add a User
5. Check Out Item
6. Return Item
7. Display Available Items
8. Delete Item
9. Exit
```

GitHub repository link:

https://github.com/ssingh-shubham/Librabry-Management