Library Management System

Code

```
#include <iostream>
#include <list>
#include <queue>
#include <algorithm>
#include <string>

using namespace std;

struct Book {
    int id;
    string title;
    string author;
    bool isIssued;
    string issuedTo;
};

class Library {
private:
    list<Book> books;
    queue<int> issuedBooksQueue;

    list<Book>::iterator findBookById(int id) {
        return find_if(books.begin(), books.end(), [id](const Book& book) {
            return book.id == id;
        });
    }

    list<Book>::iterator findBookByTitle(const string& title) {
        return find_if(books.begin(), books.end(), [&title](const Book& book) {
            return book.title == title;
        });
    }

public:
    void addBook(int id, const string& title, const string& author) {
        if (findBookById(id) != books.end()) {
            cout << "Book with ID " << id << " already exists.\n";
            return;
        }
        books.push_back({id, title, author, false, ""});
        cout << "Book added successfully.\n";
    }

    void searchBookById(int id) {
        auto it = findBookById(id);
```

```
    if (it == books.end()) {
        cout << "Book with ID " << id << " not found.\n";
    } else {
        cout << "ID: " << it->id << ", Title: " << it->title
             << ", Author: " << it->author << ", Status: "
             << (it->isIssued ? "Issued" : "Available") << "\n";
    }
}

void searchBookByTitle(const string& title) {
    auto it = findBookByTitle(title);
    if (it == books.end()) {
        cout << "Book with title \"" << title << "\" not found.\n";
    } else {
        cout << "ID: " << it->id << ", Title: " << it->title
             << ", Author: " << it->author << ", Status: "
             << (it->isIssued ? "Issued" : "Available") << "\n";
    }
}

void issueBook(int id, const string& student) {
    auto it = findBookById(id);
    if (it == books.end()) {
        cout << "Book with ID " << id << " not found.\n";
        return;
    }
    if (it->isIssued) {
        cout << "Book with ID " << id << " is already issued.\n";
    } else {
        it->isIssued = true;
        it->issuedTo = student;
        issuedBooksQueue.push(id);
        cout << "Book issued to " << student << ".\n";
    }
}

void returnBook(int id) {
    auto it = findBookById(id);
    if (it == books.end()) {
        cout << "Book with ID " << id << " not found.\n";
        return;
    }
    if (!it->isIssued) {
        cout << "Book with ID " << id << " was not issued.\n";
    } else {
        it->isIssued = false;
        it->issuedTo = "";
        if (!issuedBooksQueue.empty() && issuedBooksQueue.front() == id) {
            issuedBooksQueue.pop();
        }
        cout << "Book returned successfully.\n";
```

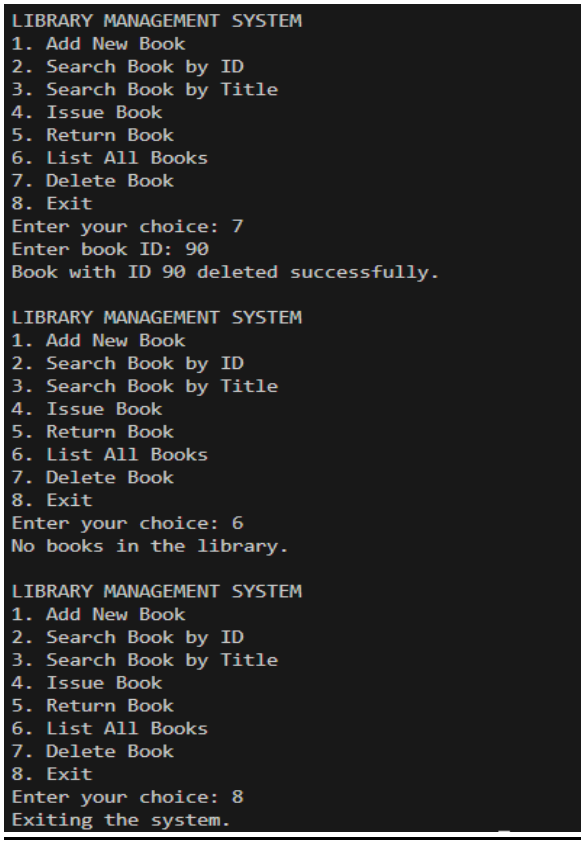
```
        }
    }

    void listAllBooks() {
        if (books.empty()) {
            cout << "No books in the library.\n";
            return;
        }
        books.sort([](const Book& a, const Book& b) {
            return a.title < b.title;
        });
        for (const auto& book : books) {
            cout << "ID: " << book.id << ", Title: " << book.title
                 << ", Author: " << book.author << ", Status: "
                 << (book.isIssued ? "Issued to " + book.issuedTo : "Available") << "\n";
        }
    }

    void deleteBook(int id) {
        auto it = findBookById(id);
        if (it == books.end()) {
            cout << "Book with ID " << id << " not found.\n";
        } else {
            books.erase(it);
            cout << "Book with ID " << id << " deleted successfully.\n";
        }
    }
};

int main() {
    Library library;
    int choice, id;
    string title, author, student;

    while (true) {
        cout << "\nLIBRARY MANAGEMENT SYSTEM\n";
        cout << "1. Add New Book\n";
        cout << "2. Search Book by ID\n";
        cout << "3. Search Book by Title\n";
        cout << "4. Issue Book\n";
        cout << "5. Return Book\n";
        cout << "6. List All Books\n";
        cout << "7. Delete Book\n";
        cout << "8. Exit\n";
        cout << "Enter your choice: ";
        cin >> choice;

        switch (choice) {
            case 1:
                cout << "Enter book ID: ";
                cin >> id;
```

```
            cin.ignore();
            cout << "Enter book title: ";
            getline(cin, title);
            cout << "Enter book author: ";
            getline(cin, author);
            library.addBook(id, title, author);
            break;
        case 2:
            cout << "Enter book ID: ";
            cin >> id;
            library.searchBookById(id);
            break;
        case 3:
            cin.ignore();
            cout << "Enter book title: ";
            getline(cin, title);
            library.searchBookByTitle(title);
            break;
        case 4:
            cout << "Enter book ID: ";
            cin >> id;
            cin.ignore();
            cout << "Enter student name: ";
            getline(cin, student);
            library.issueBook(id, student);
            break;
        case 5:
            cout << "Enter book ID: ";
            cin >> id;
            library.returnBook(id);
            break;
        case 6:
            library.listAllBooks();
            break;
        case 7:
            cout << "Enter book ID: ";
            cin >> id;
            library.deleteBook(id);
            break;
        case 8:
            cout << "Exiting the system.\n";
            return 0;
        default:
            cout << "Invalid choice. Please try again.\n";
      }
   }

   return 0;
}
```

Output

```
PS C:\Users\ABHAY PRATAP CHAUHAN> cd "c:\Users\ABHAY PRATAP CHAUHAN\Desktop\" ; if ($?) { g++ tempCodeRunnerFile.cpp -o tempCodeRunnerFile } ; if ($?) { .\
tempCodeRunnerFile }

LIBRARY MANAGEMENT SYSTEM
1. Add New Book
2. Search Book by ID
3. Search Book by Title
4. Issue Book
5. Return Book
6. List All Books
7. Delete Book
8. Exit
Enter your choice: 1
Enter book ID: 90
Enter book title: A1
Enter book author: Dr Dayal
Book added successfully.

LIBRARY MANAGEMENT SYSTEM
1. Add New Book
2. Search Book by ID
3. Search Book by Title
4. Issue Book
5. Return Book
6. List All Books
7. Delete Book
8. Exit
Enter your choice: 2
Enter book ID: 90
ID: 90, Title: A1, Author: Dr Dayal, Status: Available

LIBRARY MANAGEMENT SYSTEM
1. Add New Book
2. Search Book by ID
3. Search Book by Title
4. Issue Book
5. Return Book
6. List All Books
7. Delete Book
8. Exit
Enter your choice: 3
Enter book title: A1
ID: 90, Title: A1, Author: Dr Dayal, Status: Available
```

```
Enter book title: A1
ID: 90, Title: A1, Author: Dr Dayal, Status: Available

LIBRARY MANAGEMENT SYSTEM
1. Add New Book
2. Search Book by ID
3. Search Book by Title
4. Issue Book
5. Return Book
6. List All Books
7. Delete Book
8. Exit
Enter your choice: 4
Enter book ID: 90
Enter student name: Abhay
Book issued to Abhay.

LIBRARY MANAGEMENT SYSTEM
1. Add New Book
2. Search Book by ID
3. Search Book by Title
4. Issue Book
5. Return Book
6. List All Books
7. Delete Book
8. Exit
Enter your choice: 6
ID: 90, Title: A1, Author: Dr Dayal, Status: Issued to Abhay

LIBRARY MANAGEMENT SYSTEM
1. Add New Book
2. Search Book by ID
3. Search Book by Title
4. Issue Book
5. Return Book
6. List All Books
7. Delete Book
8. Exit
Enter your choice: 5
Enter book ID: 90
Book returned successfully.
```


LIBRARY MANAGEMENT SYSTEM
1. Add New Book
2. Search Book by ID
3. Search Book by Title
4. Issue Book
5. Return Book
6. List All Books
7. Delete Book
8. Exit
Enter your choice: 7
Enter book ID: 90
Book with ID 90 deleted successfully.
LIBRARY MANAGEMENT SYSTEM
1. Add New Book
2. Search Book by ID
3. Search Book by Title
4. Issue Book
5. Return Book
6. List All Books
7. Delete Book
8. Exit
Enter your choice: 6
No books in the library.
LIBRARY MANAGEMENT SYSTEM
1. Add New Book
2. Search Book by ID
3. Search Book by Title
4. Issue Book
5. Return Book
6. List All Books
7. Delete Book
8. Exit
Enter your choice: 8
Exiting the system.