NAMAL UNIVERSITY MIANWALI

DEPARTMENT OF ELECTRICAL ENGINEERING

QUALITATIVE AND COMPUTATIONAL REASONING 1 (LAB)

(PROJECT REPORT)

NAME	KANZA IMRAN	MUHAMMAD USAMA	MOHSIN GHAFFAR
REG. No.	NUM-BSEE-2024-16	NUM-BSEE-2024-27	NUM-BSEE-2024-18

Lab Engineer: Engr.Junaid Ashraf

Instructor: Dr.Naureen Shaukat

Submitted Date: January 14, 2025

Table of Contents:

LIBRARY BOOKS MANAGEMENT SYSTEM

➢ Abstract

This report provides an overview of the development and implementation of a Library Management System using C++. The program allows users to manage a library's collection of books through functionalities such as adding new books, viewing all books, searching for books by ID, issuing books, and returning books. The system is designed to be user-friendly and efficient, ensuring smooth library operations.

➢ Introduction

The Library Management System is such a code that facilitates the management of books within a library. The system aims to streamline the process of adding, searching, issuing, and returning books, thereby improving the overall efficiency and user experience. Libraries play a crucial role in providing access to knowledge and resources, and an efficient Library Management System is essential for maintaining their operations. This report details the objectives, methodologies, results, and conclusions of developing a Library Management System using C++.

➢ Objectives

The primary objectives of the Library Management System are:

1. To provide a simple and user-friendly interface for managing library operations.
2. To enable the addition of new books to the library's collection.
3. To allow users to view all books available in the library.
4. To facilitate searching for books by their unique ID.
5. To manage the issuing and returning of books efficiently.
6. To ensure accurate tracking of book status (issued or available).

➢ Methodologies

The development of the Library Management System involved the following steps:

1. **Requirements Gathering:** Identifying the key features and functionalities required for the system, such as adding books, viewing books, searching for books, issuing books, and returning books.
2. **Design:** Designing the user interface. The system uses arrays to store book information (ID, title, author, status) and provides a menu-driven interface for user interaction.
3. **Implementation:** Writing the C++ code to implement the system's functionalities. The code includes a main loop to display the menu and handle user input, as well as functions to perform specific tasks like adding, viewing, searching, issuing, and returning books.

4. **Testing:** Testing the system to ensure it functions correctly and handles edge cases, such as attempting to issue a book that is already issued or returning a book that was not issued.

➢ Codes

```
#include <iostream>
#include <string>
using namespace std;
int main()
{
        int size =100;
        int id[size];
        string title[size];
        string author[size];
        bool status[size];
        int element=0;
        int choice =0;

        do
        {
            cout << "\nLibrary Management System\n";
    cout << "1. Add Book"<<endl;
    cout << "2. View All Books"<<endl;
    cout << "3. Search Book by ID"<<endl;
    cout << "4. Issue Book"<<endl;
    cout << "5. Return Book"<<endl;
    cout << "6. Exit"<<endl;
    cout << "Enter your choice: ";
    cin >> choice;

    if(choice==1)
    {
        if(element>=size){
            cout<<"no space to add"<<endl;
          continue;
                }
                cout<<"enter tittle "<<endl;
                cin>>title[element];
                cout<<"enter author "<<endl;
                cin>>author[element];
                id[element]=element+1;
                cout<<"id of book is "<<id[element]<<endl;
                status[element] = false;
                ++element;


            }
```

```
                else if(choice==2)
                {
                        if(element==0){
                                cout<<"no book avalaible"<<endl;
                                continue;
                        }
                        for(int i=0;i<element;i++){
                                cout<<"book author "<<author[i]<<"|book title "<<title[i]<<"|book id "<<id[i]<<"|status "<<(status[i] ? "issued ": "available")<<endl;
                        }
                }
                else if(choice==3)
                {
                        int search;
                        cout<<"enter book id"<<endl;
                        cin>>search;
                        bool found = false;
                        for(int i=0;i<element;i++)
                        {
                                if(search==element){
                                        cout<<"book available "<<endl<<"|book id"<<id[i]<<"|book tittle"<<title[i]<<"|status"<<(status[i] ? "issued ": "available")<<endl;
                                        found = true;
                                   break;
                           }
                           if(!found)
                           cout<<"book with id "<<search<<" book is not found"<<endl;
                        }

                }
                else if(choice ==4)
                {
                        int issue;
                        cout<<"enter id of the book to issue"<<endl;
                        cin>>issue;
                        bool found = false;
                        for(int i =0;i<element;i++)
                        {
                                if(issue==id[i])
                                {
                                        found = true;
              if (status[i]==false) {
                 status[i] = true;
                 cout << "Book issued successfully." << endl;
              } else {
                 cout << "Book is already issued." << endl;
```

```
            }
            break;
                                }
                                if (found==false) {
        cout << "Book with ID " << issue << " not found" << endl;
        }
                        }
            }
            else if(choice==5)
            {
               int retun;
               cout<<"enter id to return"<<endl;
               cin>>retun;
               bool found = false;
               for(int i=0;i<element;i++)
               {
                        if(id[i] == retun)
                        {
                                found=true;
                                if (status[i]) {
               status[i] = false;
               cout << "Book returned successfully." << endl;
            }
            else
                                        {
               cout << "Book was not issued." << endl;
               break;
            }
                                }
                             if (found==false) {
               cout << "Book with ID " << retun << " not found." << endl;
               }
      }
               }
               else if (choice == 6) {
      cout << "Exiting..." << endl;

   } else {
      cout << "Invalid choice. Please try again." << endl;
   }

        }while(choice!=6);
}
```

➢ Outputs

```
Library Management System
1. Add Book
2. View All Books
3. Search Book by ID
4. Issue Book
5. Return Book
6. Exit
Enter your choice: 1
enter tittle
aaa
enter author
AAA
id of book is 1

Library Management System
1. Add Book
2. View All Books
3. Search Book by ID
4. Issue Book
5. Return Book
6. Exit
Enter your choice: 2
book author AAA|book title aaa|book id 1|status available
```

```
1. Add Book
2. View All Books
3. Search Book by ID
4. Issue Book
5. Return Book
6. Exit
Enter your choice: 3
enter book id
1
book available
|book id1|book tittleaaa|statusavailable

Library Management System
1. Add Book
2. View All Books
3. Search Book by ID
4. Issue Book
5. Return Book
6. Exit
Enter your choice: 4
enter id of the book to issue
1
Book issued successfully.
```

```
Enter your choice: 5
Enter ID to return: 1
Book returned successfully.
```

➢ Results

The Library Management System was tested rigorously to ensure it met the objectives. The key results observed during testing include:

1. **Functionality:** The system successfully added, viewed, searched, issued, and returned books. Each function operated correctly, and the menu-driven interface was intuitive for users.
2. **User Interface:** The interface was clear and user-friendly, with prompts guiding users through each action.
3. **Book Status:** The system accurately tracked the status of each book (issued or available), preventing multiple issues of the same book.
4. **Error Handling:** The system appropriately handled errors, such as attempting to issue a book that was already issued or returning a book that was not issued, and provided clear feedback to the user.

➢ Conclusions

The Library Management System developed in this project successfully meets the outlined objectives. The system provides a simple and efficient way to manage a library's collection of books, with functionalities for adding, viewing, searching, issuing, and returning books. The user-friendly interface and clear output messages ensure a smooth user experience. Future improvements could include adding features such as deleting books, updating book information, and implementing a more advanced data storage mechanism (e.g., using files or a database) to handle larger collections and maintain persistence across sessions.