# LIBRARY MANAGEMENT SYSTEM

Project submitted to the

SRM University – AP, Andhra Pradesh

Submitted in partial fulfillment of the requirement for the award of the degree of

# Bachelor of Technology in

**Computer Science and Engineering**

**School of Engineering and Sciences**

Submitted By **ABHISHEK CHAITANYA**

**SRAVYA**

**ILIAZ KHAN**

**NIRANJAN**

Under the guidance of

**BHASKAR SANTHOSH KUMAR**



**Department of Computer Science and Engineering**

**SRM University,AP**

##### Neerukonda, Mangalagiri, Guntur Andhra Pradesh *–* 522 240

**[NOVEMBER, 2022]**

## Department of Computer Science and Engineering

SRM University,AP



# CERTIFICATE

This is to certify that the Project report entitled **“Library Management System ”** is being submitted by **Abhishek ,,Chaitanya, ,Sravya ,Iliazkhan, Niranjan** are students of Department of Computer Science and Engineering, SRM University,AP, in partial fulfillment of the requirement for the degree of **“B.Tech(CSE)”** carried out by her/his during the academic year 2021-2022.

Signature of the Supervisor Signature of Head of the Dept.

**BHASKAR SANTHOSH KUMAR JATINDRA KUMAR DASH**

**Acknowledgement**

The satisfaction that accompanies the successful completion of any task would be incomplete without introducing the people who made it possible and whose constant guidance and encouragement crowns all efforts with success.

I am extremely grateful and express my profound gratitude and indebtedness to my project guide, **Mr.Bhaskar Santhosh**, Lecturer, Department of Computer Science & Engineering, SRM University,Andhra pradesh, for her kind help and for giving me the necessary guidance and valuable suggestions in completing this project work.

## ABSTRACT

Online Library Management System is a system which maintains the information

about the books present in the library, the members of library to

whom books are issued, library staff and all. This is very difficult to organize

manually. Maintenance of all this information manually is a very complex task.

Owing to the advancement of technology, organization of an Online Library

becomes much simple. The Online Library Management has been designed to

computerize and automate the operations performed over the information about the members, book issues and returns and all other operations. This computerization of library helps in many instances of its maintenances. It reduces the workload of management as most of the manual work done will be reduced.

# CONTENTS

|  |  |  |  |
| --- | --- | --- | --- |
| Chapter No. | Chapter Name | | Page No. |
| 1 | Introduction | | 06 |
| 2 | Objective | | 07 |
| 3 | System Requirement specifications | | 08 |
| 4 | Source code | | 09 |
|  |  | |  |
| 5 | Conclusion | |  |
|  |  | |  |
|  | |

**CHAPTER 1**

# INTRODUCTION

Library Management System is an application which refers to library systems which are generally small or medium in size. It is used by librarian to manage the library using a computerized system where he/she can add new books, videos and Page sources. Books and student maintenance modules are also included in this system which would keep track of the students using the library and also a detailed description about the books a library contains. With this computerized system there will be no loss of book record or member record which generally happens when a non computerized system is used. All these modules are able to help librarian to manage the library with more convenience and in a more efficient way as compared to library systems which are not computerized.

# CHAPTER 2

#### OBJECTIVE

The project aims and objectives that will be achieved after completion of this project are

discussed in this subchapter. The aims and objectives are as follows:

 A perfect login and registration system for both student and staff without any obligations

 A column to search books according to category.

 Check the student details about the books borrowed .And checking the due dates and borrowed dates.

 Individual interfaces for students and library staff.

 Availability of books according including quantity of availability.

 Manage the book quantity after every book borrowed by student and after every return.

Librarian is able to search record by using few clicks of mouse and few search keywords thus saving his valuable time.

# SYSTEM REQUIREMENTS

**EFFICIENCY REQUIREMENT**

When a library management system will be implemented librarian and user will easily acess library as searching and book transaction will be very faster .

**RELIABILITY REQUIREMENT**

The system should accurately performs member registration ,member validation ,report generation, book transaction and search

**USABILITY REQUIREMENT**

The system is designed for a user friendly environment so that student and staff of

library can perform the various tasks easily and in an effective way

**SOURCE CODE :**

#include <iostream>

#include <fstream> // reading and writing files

#include <string> //to perform string operations

#include <ctime> //to get current date and time

#include <cstdio> // to rename and remove files

#include <cstring>//to work with strings

#include "duedate\_printter.cpp"//file to get de date

using namespace std;

string role, username;

char str[11];

class Login

{

public:

    string password, un, pw, ro;

    Login() // Default constructor to intialize username and password

    { cout<<"-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-..-.-.-.-.-\n";

        cout << "\tEnter Username: ";

        cin >> username;

        cout << "\tEnter password: ";

        cin >> password;

        cout<<"-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-..-.-.-.-.-\n";

    }

    bool checklogin() // checks if the login details are valid

    {

        // ifstream is used to read a file

        ifstream read(username + ".txt");

        // checking the first three lines of username's file

        getline(read, un); // username

        getline(read, pw); // password

        getline(read, ro); // role

        role = ro;

        if (un == username && pw == password)

        {

            return true;

        }

        else

        {

            return false;

        }

    }

};

class Register

{

public:

    string word, username, password, role;

    Register() // Default constructor to intialize username and password

    { cout<<"-\_-\_-\_-\_-\_-\_-\_-\_-\_-\_-\_-\_-\_-\_-\_-\_-\_-\_-\_-\_-\_-\_-\_-\_-\_-\_-\_-\_-\_\n";

        cout << "\tRegistration" << endl;

        cout << "\tEnter your username: " << endl;

        cin >> username;

        cout<<"-\_-\_-\_-\_-\_-\_-\_-\_-\_-\_-\_-\_-\_-\_-\_-\_-\_-\_-\_-\_-\_-\_-\_-\_-\_-\_-\_-\_-\_\n";

    }

    void registration()

    {

        fstream f; // fstream is used to read and write into files

        f.open("file\_names.txt");

        int flag = 0;

        while (f >> word)

        {

            if (word == username)

            {

                flag = 1;

            }

        }

        if (flag == 0)

        { cout<<"- - - - - - - - - - - - - - - - - - - - - - - - - -\n";

            cout << "\tEnter your password: ";

            cin >> password;

            cout << "\tEnter if you are student or staff: ";

            cin >> role;

            cout<<"- - - - - - - - - - - - - - - - - - - - - - - - - -\n";

            // writes into the file and creates if file doesn't exist

            ofstream fl;

            fl.open("file\_names.txt", ios\_base::app);

            fl << username << endl;

            ofstream outfile;

            outfile.open(username + ".txt", ios\_base::app); // to open file in append mode;

            outfile << username << endl;

            outfile << password << endl;

            outfile << role << endl;

            cout << "\tRegistered Successfully" << endl;

        }

        else

        {

            cout << "\tUsername already exist";

        }

        f.close();

    }

};

class Returnbook;

class Addbooks

{

public:

    string category, book\_name, code;

    int flag = 0;

    string count;

    void get\_add(int t) // Default constructor

    {

        flag = t;

        cout<<"...................................................\n";

        cout << "\tEnter the category the book belongs to: ";

        getline(cin >> ws, category);

        cout << "\tEnter the code of book: ";

        getline(cin >> ws, code);

        cout << "\tEnter the book name: ";

        getline(cin >> ws, book\_name);

        cout << "\tEnter the number of books added:";

        getline(cin >> ws, count);

        cout<<"...................................................\n";

    }

    friend void add(Addbooks &, Returnbook &);

};

class Returnbook

{

public:

    string category, code;

    int flag = 0;

    void get\_return(int t)

    {

        flag = t;

        cout << "\t------------------------Your books-------------------------" << endl;

        cout << "\tCategory Code Due-Date" << endl;

        string detail;

        ifstream fp;

        fp.open(username + ".txt", ios::app);

        int c = 0;

        while (getline(fp, detail))

        {

            if (c >= 3)

            {

                cout << detail << " ";

            }

            if (c >= 5)

            {

                c = 2;

                cout << endl;

            }

            c++;

        }

        cout << "\t\nenter the category of the book" << endl;

        cin >> category;

        cout << "\t\nenter the code of book you want to return " << endl;

        cin >> code;

        fp.close();

        string today;

        string detail1, detail2, dx;

        ifstream fx;

        fx.open(username + ".txt");

        while (getline(fx, dx))

        {

            if (dx == code)

            {

                getline(fx, detail1);

                getline(fx, detail2);

            }

        }

        cout << "\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\* \n";

        int choice;

        int ex;

        cout<<"^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^\n";

        cout << "\tDue date of the book : " << detail2 << endl;

        cout << "\tEnter '1' if not returning on time if not press any key. " << endl;

        cin >> choice;

        switch (choice)

        {

        case 1:

            cout << "\tYou have to the fine as you had not returned time:" << endl;

            cout << "\tEnter how many days you exceeded : " << endl;

            cin >> ex;

            cout << "\tPlease pay a fine of $0.5 perday " << endl

                 << endl;

            ;

            cout << "\tYou need to pay $" << ((0.5) \* ex) << " rupees." << endl;

            cout << "\tThank you" << endl;

            break;

        default:

            cout << "\tThank You" << endl;

            break;

        }

         cout << "\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\* \n";

    }

    void del\_inuser()

    {

        fstream fi;

        string detail;

        fi.open(username + ".txt");

        ofstream out;

        out.open("temp.txt", ios::app);

        while (getline(fi, detail))

        {

            if (detail == category)

            {

                getline(fi, detail);

                getline(fi, detail);

                getline(fi, detail);

                getline(fi, detail);

            }

            out << detail << endl;

        }

        out.close();

        fi.close();

        char char\_array[(username + ".txt").length()];

        strcpy(char\_array, (username + ".txt").c\_str());

        remove(char\_array);

        rename("temp.txt", char\_array);

    }

    friend void add(Addbooks &, Returnbook &);

};

void add(Addbooks &a, Returnbook &r)

{

    string category, code, count;

    if (a.flag > r.flag)

    {

        category = a.category;

        code = a.code;

        count = a.count;

    }

    else

    {

        category = r.category;

        code = r.code;

        count = "1";

    }

    fstream f;

    f.open("all\_books.txt");

    string word;

    int c = 0, flag = 0;

    while (f >> word)

    {

        if (word == category)

        {

            c = 1;

        }

    }

    if (c == 0)

    {

        // entering category name into all\_books.txt

        ofstream outfile;

        outfile.open("all\_books.txt", ios\_base::app);

        outfile << category << endl;

        outfile.close();

    }

    fstream fi;

    int se;

    fi.open(category + ".txt");

    string b;

    string temper;

    int temp, position, size;

    int ci = 0;

    ofstream out;

    out.open("temp.txt", ios\_base::app);

    while (getline(fi, b))

    {

        // if entered code is already present in the category count will be increased

        if (flag == 1)

        {

            temp = stoi(b);

            temp += stoi(count);

            temper = to\_string(temp);

            out << temper << endl;

            getline(fi, b);

            ci = 1;

            flag = 0;

        }

        // entering every line into a temporary file

        out << b << endl;

        if (b == code)

        {

            flag = 1;

        }

    }

    fi.close();

    out.close();

    if (ci == 0)

    {

        // entering code and book name into the respective category

        ofstream file;

        file.open(category + ".txt", ios\_base::app);

        file << code << endl;

        file << count << endl;

        file << a.book\_name << endl;

        file.close();

        remove("temp.txt");

    }

    // converting category name into character array to use remove and rename functions

    char char\_array[(category + ".txt").length()];

    strcpy(char\_array, (category + ".txt").c\_str());

    if (ci == 1)

    {

        // renaming temp file with category

        remove(char\_array);

        rename("temp.txt", char\_array);

    }

}

class Takebook; // forward declaration

class Deletebooks

{

public:

    string category, code, count;

    int flag = 0;

    void get\_delete(int t)

    {

        flag = t;

        cout<<"-------------------------------------------------------------------\n";

        cout << "Delete a book" << endl;

        cout << "enter the category from which the book should be deleted " << endl;

        cin >> category;

        cout << "enter the code of book to be deleted" << endl;

        cin >> code;

        cout << "enter the count of books to be deleted" << endl;

        cin >> count;

        cout<<"-------------------------------------------------------------------\n";

    }

    friend void deleted(Deletebooks &, Takebook &);

};

class Display

{

public:

    void dis()

    {

        fstream f;

        string word;

        string detail;

        f.open("all\_books.txt", ios::in);

        if (f.is\_open())

        {

            while (getline(f, word))

            {

                fstream file;

                file.open(word + ".txt", ios::in);

                int temp = 1;

                cout << "---------------------------------------" << endl;

                cout << "Category: " << word << endl;

                cout << "Codes Count Books" << endl;

                while (getline(file, detail))

                {

                    if (temp > 3)

                        temp = 1;

                    cout << detail << " ";

                    temp += 1;

                    if (temp > 3)

                        cout << endl;

                }

            }

        }

    }

};

// borrowing and renewal

class Takebook

{

public:

    string book, category, detail, code;

    int flag = 1;

    char date[80];

    void borrow(int t)

    {

        flag = t;

        cout << "Subjects available : " << endl;

        cout << "-- -- -- -- -- -- -- -- -- -- -- -- -- --" << endl;

        ifstream ifs("all\_books.txt");

        while (!ifs.eof())

        {

            getline(ifs, book);

            cout << "" << book << "\n";

        }

        cout << "\nEnter category of book you need : " << endl;

        cin >> category;

        ifstream f(category + ".txt");

        int temp = 1;

        while (getline(f, detail))

        {

            if (temp > 3)

                temp = 1;

            cout << detail << " ";

            temp += 1;

            if (temp > 3)

                cout << endl;

        }

        cout << "-- -- -- -- -- -- -- -- -- -- -- -- -- --" << endl;

        cout << "\nEnter the book code carefully : " << endl;

        cin >> code;

    }

    friend void deleted(Deletebooks &, Takebook &);

    void datechange()

    {

        cout << "^^^^^^^^^^^^^^^^^^^^^^^^^ :) ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^" << endl;

        ofstream file;

        file.open(username + ".txt", ios\_base::app);

        file << category << endl;

        file << code << endl;

        dates();

        strcpy(str, buffer);

        strcpy(temp, buffer);

        int d1[3];

        int i = 0;

        char \*token = strtok(str, "-");

        while (token != NULL)

        {

            d1[i] = stoi(token);

            token = strtok(NULL, "-");

            i++;

        }

        file << d1[0] << "-" << d1[1] << "-" << d1[2] << endl;

        cout << "\n---------------------------------------------------------\n";

        int x = 20;

        cout << addDays(d1[0], d1[1], d1[2], x)<<endl;

        file << addDays(d1[0], d1[1], d1[2], x) << endl;

             cout<< endl;

    }

};

void deleted(Deletebooks &d, Takebook &t)

{

    string count, category, code;

    if (d.flag > t.flag)

    {

        count = d.count;

        category = d.category;

        code = d.code;

    }

    else

    {

        count = "1";

        category = t.category;

        code = t.code;

    }

    fstream file;

    file.open(category + ".txt");

    string word;

    int c = 3;

    char char\_array[(category + ".txt").length()];

    strcpy(char\_array, (category + ".txt").c\_str());

    int temper;

    string temp;

    if (file.is\_open())

    {

        ofstream f;

        f.open("temp.txt", ios::app);

        while (getline(file, word))

        {

            if (word == code)

            {

                temp = word;

                c = 0;

            }

            if (c == 1)

            {

                if (stoi(word) < stoi(count))

                {

                    f.close();

                    remove("temp.txt");

                    break;

                }

                else if (stoi(word) > stoi(count))

                {

                    // temp = stoi(word);

                    f << temp << endl;

                    temper = stoi(word) - stoi(count);

                    f << to\_string(temper) << endl;

                    c = 2;

                }

            }

            if (c >= 3)

            {

                f << word << endl;

            }

            c = c + 1;

        }

    }

    file.close();

    remove(char\_array);

    rename("temp.txt", char\_array);

}

class Display\_stu

{

public:

    void get()

    {

        ifstream fp;

        fp.open("file\_names.txt");

        string detail;

        cout << "List of students" << endl;

        while (getline(fp, detail))

        {

            cout << detail << endl;

        }

        fp.close();

        int choice;

        cout << "1. To display all records" << endl;

        cout << "2. To display a student's record" << endl;

        cin >> choice;

        ifstream out;

        out.open("file\_names.txt");

        string word;

        if (choice == 1)

        {

            cout << "Details of all the students" << endl;

            while (getline(out, detail))

            {

                cout << "------------------------------" << endl;

                cout << "Username: " << detail << endl;

                cout << "Category Code Due-Date" << endl;

                ifstream fi;

                fi.open(detail + ".txt");

                int counte = 1, c = 0;

                while (getline(fi, word))

                {

                    if (counte > 3)

                    {

                        cout << word << " ";

                        c++;

                    }

                    if (c == 4)

                    {

                        cout << endl;

                        c = 0;

                    }

                    counte++;

                }

            }

        }

        if (choice == 2)

        {

            int c = 0;

            cout << "enter the student's username to visit their profile" << endl;

            cin >> username;

            ifstream fi;

            fi.open(username + ".txt");

            cout << "--------------------------------" << endl;

            cout << "Username: " << username << endl;

            cout << "Details of the books taken by the student" << endl;

            cout << "Category Code Due-Date Borrow Date" << endl<<endl;

            while (getline(fi, detail))

            {

                c++;

                if (detail == username)

                {

                    getline(fi, detail);

                    getline(fi, detail);

                    getline(fi, detail);

                }

                cout << detail << " ";

                if (c == 3)

                {

                    cout << endl;

                    c = 0;

                }

            }

        }

    }

};

class MyProfile

{

public:

    void display()

    {

        ifstream fp;

        fp.open(username + ".txt");

        string detail;

        cout << "------------------------------------" << endl;

        cout << "Username: " << username << endl;

        cout << "\tCategory Code Due-Date Borrow Date" << endl<<endl;

        int counte = 1, c = 0;

        while (getline(fp, detail))

        {

            if (counte > 3)

            {

                cout <<"\t"<<detail << " ";

                c++;

            }

            if (c == 4)

            {

                cout << endl;

                c = 0;

            }

            counte++;

        }

    }

};

int main()

{

    int temp = 1;

    cout << "DASHBOARD" << endl;

    int choice;

    cout << "-- -- -- -- -- -- -- -- -- -- -- -- - " << endl;

    cout << "1. Register " << endl;

    cout << "2. Login " << endl;

    cout << "Enter your choice" << endl;

    cin >> choice;

    if (choice == 1)

    {

        Register r;

        r.registration();

    }

    else if (choice == 2)

    {

        Login l;

        if (l.checklogin())

        {

            while (temp)

            {

                int stac, stuc;

                if (l.ro == "staff")

                {

                    cout << "1. Add Books" << endl;

                    cout << "2. Delete books" << endl;

                    cout << "3. Display books" << endl;

                    cout << "4. Display student records" << endl;

                    cout << "5. Back to Menu" << endl;

                    cout << "6. To exit " << endl;

                }

                else

                {

                    cout << "1. Take a book" << endl;

                    cout << "2. Return a book" << endl;

                    cout << "3. Renewal" << endl;

                    cout << "4. Display books" << endl;

                    cout << "5. My Profile" << endl;

                    cout << "6. Back to Menu" << endl;

                    cout << "7. To exit " << endl;

                }

                if (l.ro == "staff")

                    cin >> stac;

                else

                    cin >> stuc;

                if (stac == 1)

                {

                    char resp = 'y';

                    while (resp == 'y')

                    {

                        Addbooks a;

                        Returnbook r;

                        a.get\_add(1);

                        add(a, r);

                        cout << "Enter 'y' if you want to add more books and 'n' if you want to stop" << endl;

                        cin >> resp;

                    }

                }

                if (stuc == 1)

                {

                    Takebook t;

                    Deletebooks d;

                    t.borrow(1);

                    deleted(d, t);

                    t.datechange();

                }

                if (stuc == 2)

                {

                    Returnbook r;

                    Addbooks a;

                    r.get\_return(1);

                    add(a, r);

                    r.del\_inuser();

                }

                if (stuc == 3)

                {

                    string count = "1";

                }

                if (stac == 2)

                {

                    Deletebooks d;

                    Takebook t;

                    d.get\_delete(2);

                    deleted(d, t);

                }

                if (stac == 3 || stuc == 4)

                {

                    cout << "Here is the list of available books with their codes" << endl;

                    Display d;

                    d.dis();

                }

                if (stac == 4)

                {

                    Display\_stu d;

                    d.get();

                }

                if (stuc == 5)

                {

                    MyProfile m;

                    m.display();

                }

                if (stac == 5 || stuc == 6)

                {

                    main();

                }

                if (stac == 6 || stuc == 7)

                {

                    temp = 0;

                }

            }

        }

        else

        {

            cout << "Invalid Login" << endl;

            main();

        }

    }

    else if (choice == 3)

    {

        temp = 0;

    }

}

OUTPUT:

Text

Description automatically generated

Text

Description automatically generatedText

Description automatically generatedText

Description automatically generatedText

Description automatically generatedText

Description automatically generatedText

Description automatically generated

CONCLUSION :

To help with day-to-day operations, the **Library Management System Project In C++** automates the fundamental library tasks. The books stored information about many volumes and allows staff and students to add, borrow, and return them. Different user types receive various rights.

Additionally, it keeps track of information on books,and students that is necessary for a variety of library operations.

The system is intended to be user-friendly and effective thanks to the software. If the book is returned late, there should be a fine paid.