LIBRARY MANAGEMENT

A MINI PROJECT REPORT

18CSC204J -Design and Analysis of Algorithms Laboratory

Submitted by

Kolli loka Prashanth reddy [RA2011028010111]

Under the guidance of

Dr. B Yamini

Assistant Professor, Department of Networking and Communication

In Partial Fulfillment of the Requirements for the Degree of

BACHELOR OF TECHNOLOGY
in
COMPUTER SCIENCE ENGINEERING
with specialization in CLOUD COMPUTING

DEPARTMENT OF NETWORKING AND COMMUNICATIONS COLLEGE OF ENGINEERING AND TECHNOLOGY SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

KATTANKULATHUR- 603 203

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY
KATTANKULATHUR – 603 203

BONAFIDE CERTIFICATE

Certified that this mini project report titled “LIBRARY MANNAGEMENT” is the bonafide work done by <KOLLI LOKA PRASHANTH REDDY> (RA2011028010111) who carried out the mini project work and Laboratory exercises under my supervision for **18CSC204J -Design and Analysis of Algorithms Laboratory**. Certified further, that to the best of my knowledge the work reported herein does not form part of any other work.

Dr. B Yamini
ASSISTANT PROFESSOR
18CSC204J -Design and Analysis of Algorithms
Course Faculty
Department of Networking and Communications

Signature of the Internal Examiner-I **Signature of the Internal Examiner-II**

Abstract:

1.Library Transaction Process Library Transaction System is a library Management Software for Monitoring and controlling the transactions in a Library. The Project is developed in JAVA, which mainly focus on basic Operations in a library like Adding New Member, New Books, and Updating New Information, Searching Books, Members and facility to Borrow and Return Books.
Modules of The software Library Registration Books search Borrow books Login registration
Library Registration: The first procedure is the registration of the people who arrive to the library. The receptionist has the authority to enter the name, address and contact number of the people who visit the library along with the name of the book they want to read.
Books : Admin has the authority to add, delete or modify the details of the book available to/from the system. The software keeps track of all the information about the books in the library, their cost, their complete details and total number of books available in the Library.
Search : Available Books in the Library can be searched through this Module. Data can be extracted from the Database in this Module.
Borrow books : A person can also borrow the book for particular days. All the information will be entered in the system. If the person doesn't return the book before the due date, a fine will be added and the information will be include.

Table of Contents:

Chapter 1

Introduction

Fig 1. library management:

1.1 Problem Statement

Library Management System is a software which handles the entire data of library. It makes the work of librarian very easy instead of writing data in a notebook. In past the librarians were using notebooks to write the data of books along with students name who borrowed that book. So it was very difficult to keep track on each and every book. If librarian want's to search for a particular book then that task was very time consuming . So to make this task easy the programming languages were developed and C++language is one of them.

Chapter 2

A library is a pool of sources of information. This similar resources had made a well-defined community including readers, students etc. to refer or to borrow the book more conveniently.
The Library Management System Software for Library Management is used to find books and access journals easily. The library automation system automates the typical procedures of libraries and reduces the workload for library staffs. It makes the consistency of the record and the standard quality. When people value information more and more the information industry got developed and the technology changed the expectations of library patrons. It gives both an opportunity and challenge to the libraries. The integrated library system is used to manage the more-complex activities and it enables the librarian to manage library resources in a more effective way to save time and effort.

DESIGN TECHNIQUE

DIVIDE& CONQUER

In divide and conquer approach, a problem is divided into smaller problems, then the smaller problems are solved independently and finally the solutions of smaller problems are combined into a solution for the large problem.

Generally, divide-and-conquer algorithms have three parts −

- Divide the problem into a number of sub-problems that are smaller instances of the same problem.
- Conquer the sub-problems by solving them recursively. If they are small enough, solve the sub-problems as base cases.
- Combine the solutions to the sub-problems into the solution for the original problem.

2.3 ALGORITHM

1. Start with welcome screen
2. Get password from user
3. Check password. Is password correct
   Yes:-goto step 4
   No:-print wrong message and goto step 2
4. Display mainmenu as below
   - 1.add books
   - 2.delete books
   - 3.search books
   - 4.issue books
   - 5.view book list
   - 6.edit book's record
   - 7.close application
5. Get choice from user
   Choice:-1 call function addbook
   Choice:-2 call function deletebook
   Choice:-3 call function searchbook
   Choice:-4 call function Issuebooks
   Choice:-5 call function viewbook
   Choice:-6 call function editbooks
   Choice:-7 goto step 6

6.stop

EXPLANATION OF ALGORITH

Addbook function

Steps

1.declare file pointer 'fp'

2.display categories of book to be added

3.get option from user.Is user want to add books

Yes:-goto step 3

No:-go back to main menu

4. Open file 'fp' to write

5. Assign the pointer to the end of the file to write

6. get data from user

7. write input data on a file

8. close file

9.print option to add another books

Yes:-goto step 1

No:-go back to main menu

Deletebook function

Steps

1.get book ID from user to delete

2. Declare file pointer 'ft'

3.open file 'fp' to read

4. Assign the pointer to the beginning of the file to be read

5.loop until 'End of file' is not encountered read data from file.

6.Is user input book id= book Id on a file

Yes:- a. open 'ft' file and copy all data of 'fp' file in 'ft'file except that data which we want to delete

b. delete 'fp' file and rename ft file by ' fp' file name and goto step 7

No:-print error message and close file and go back to main menu

7.close file

8.print option to delete another book

Yes:-goto step1

No:-go back to main menu

Searchbook function

Steps

1. Display option for search
    - 1. Search by id
    - 2.Search by book name
2. Open the fie 'fp' to read
3. Assign the pointer to the beginning of the file to be read
4. If option is search by id
    - Get book id from user
    - Loop until 'End of file' is not encountered read data from file
    - Is user input book id =book id on a file
      Yes:- Display all information about that book id and goto step 6
      No:- Print sorry message and goto step 6

5. If option is search by book name
    - Get book name from user
    - Loop until 'End of file' is not encountered read data from file
    - Is user input book name=book name on a file
      Yes:-display all information about that book and goto step 6
      No:-print sorry message and goto step 7

6. Close file
7. Display option to search another book

Issuebook function

Steps

1. Display option as below
   - 1.Issue book
   - 2.View issued book
   - 3.Search issued book
   - 4.Remove issued book
2. Open 'fs' file
3. Assign the pointer to the beginning of the file to be read
4. Get choice from user

   Choice -1

   a. Get book id from user
   b. Loop until 'End of file' is not encountered read data from file
   c. Is input book id=book id on a file

      Yes:-get student name whom book has to issue

      No:-print sorry message and goto step e
   d. Write information of student and issued data on a file
   e. Display option to issue another book

      Yes:-goto step a

      No:-goto step 5

Editbook function

Steps

1. Get book id to be edited from user
2. Open file 'fs'
3. Assign the pointer to the beginning of the file to be read
4. Loop until 'End of file ' is not encountered read data from file
5. Call function checkid. Is checkid=1

   Yes:-goto step 6

   No:-display sorry message and goto step 7
6. a.get new data from user of that book which to be edited

   b.Assign the pointer to the current position

   c.Overwrite the new data on old data of that book and goto step 7

   No:- Display sorry message and goto step 7
7. Close file
8. Display option to edit another book

   Yes:-goto step 1

   No:-goto step 8
9. Go back to main menu

Viewbook function

Steps

1. Open ‘fs’ file
2. Assign the pointer to the beginning of the file to be read
3. Loop until ‘End of file’ is not encountered read data from file
4. Display list of all book with complete information
5. Show total number of books in library
6. Close file
7. Go back to main menu

Password function

Steps

1. Declare two string ‘pass’ and ‘ password=sharda’
2. Print “Enter the password”.
3. Until user entered the enter key get character for user
4. Store input character in string ‘pass’
5. Compare two string ‘pass’ and ‘password’ .Is both equal

   Yes:-goto step 6

   No:-print error message and goto step 2
6. Print password match message
7. Goto main menu

Chapter 3

This chapter covers the time complexity analysis for the approach used above.

3.1 Time Complexity Analysis

Time complexity of the above solution is O(nlog n).

CODE:

```
//***************************************************************
//                HEADER FILES USED IN PROJECT
//***************************************************************

#include<fstream>
#include<conio.h>
#include<stdio.h>
#include<process.h>
#include<string.h>
#include<iomanip>
#include<iostream>

using namespace std;

//***************************************************************
//                CLASSES USED IN PROJECT
//***************************************************************

//***************************************************************
//                CLASSES FOR BOOK DATA
//***************************************************************

class book
{
        char bno[6];
           char bname[50];
           char aname[20];
```

```
public:
            void create_book()
            {
        cout<<"\nNEW BOOK ENTRY...\n";
        cout<<"\nEnter The book no.";
        cin>>bno;
        cout<<"\n\nEnter The Name of The Book ";
        cin.get(bname,50);
        cout<<"\n\nEnter The Author's Name ";
        cin.get(aname,20);
        cout<<"\n\n\nBook Created..";
    }

    void show_book()
    {
        cout<<"\nBook no. : "<<bno;
        cout<<"\nBook Name : "<<bname;
//puts(bname);
        cout<<"Author Name : "<<aname;
        //puts(aname);
    }

    void modify_book()
    {
        cout<<"\nBook no. : "<<bno;
        cout<<"\nModify Book Name : ";
        gets(bname);
        cout<<"\nModify Author's Name of Book : ";
        gets(aname);
    }
```

```
void report()

{cout<<bno<<setw(30)<<bname<<setw(30)<<anam
e<<endl;}


};        //class ends here

//************************************************************
******
//                  CLASSES FOR STUDENT DATA
//************************************************************
******

class student
{
    char admno[6];
    string name;
    char stbno[6];
    int token;
public:
    void create_student()
    {
        cout<<"\n\n\n\n";
        cout<<"\nNEW STUDENT ENTRY...\n";
        cout<<"\nEnter The admission no. ";
         cin>>admno;
        cout<<"\n\nEnter The Name of The Student ";
        cin>>name;
         token=0;
        stbno[0]='/0';
        cout<<"\n\nStudent Record Created..";
    }
```

```
void show_student()
  {
     cout<<"\nAdmission no. : "<<admno;
     cout<<"\nStudent Name : "<<name;
     //puts(name);
     cout<<"\nNo of Book issued : "<<token;
     if(token==1)
        cout<<"\nBook No "<<stbno;
  }

  void modify_student()
  {
     cout<<"\nAdmission no. : "<<admno;
     cout<<"\nModify Student Name : ";
     cin>>name;
  }

  char* retadmno()
  {
     return admno;
  }

  char* retstbno()
  {
     return stbno;
  }

  int rettoken()
  {
     return token;
  }
```

```
void addtoken()
   {token=1;}

   void resettoken()
   {token=0;}

   void getstbno(char t[])
   {
      strcpy(stbno,t);
   }

   void report()

{cout<<"\t"<<admno<<setw(20)<<name<<setw(10)<<token<<endl
;}

};        //class ends here

//***************************************************************
//        global declaration for stream object, object
//***************************************************************

fstream fp,fp1;
book bk;
student st;

//***************************************************************
//        functions to write in file
//***************************************************************
```

```
void write_book()
{
    char ch;
    fp.open("book.dat",ios::out|ios::app);
    do
    {
        cout<<"\n\n\n\n";
        bk.create_book();
        fp.write((char*)&bk,sizeof(book));
        cout<<"\n\nDo you want to add more record..(y/n?)";
        cin>>ch;
    }while(ch=='y'||ch=='Y');
    fp.close();
}

void write_student()
{
    char ch;
    fp.open("student.dat",ios::out|ios::app);
    do
    {
        st.create_student();
        fp.write((char*)&st,sizeof(student));
 cout<<"\n\ndo you want to add more record..(y/n?)";
        cin>>ch;
    }while(ch=='y'||ch=='Y');
    fp.close();
}
```

```
//*************************************************************
*****
//        functions to read specific record from file
//*************************************************************
******

void display_spb(char n[])
{
    cout<<"\nBOOK DETAILS\n";
    int flag=0;
    fp.open("book.dat",ios::in);
    while(fp.read((char*)&bk,sizeof(book)))
    {
        if(strcmpi(bk.retbno(),n)==0)
        {
            bk.show_book();
             flag=1;
        }
    }

    fp.close();
    if(flag==0)
        cout<<"\n\nBook does not exist";
    getch();
}

void display_sps(char n[])
{
```

```
cout<<"\nSTUDENT DETAILS\n";
    int flag=0;
    fp.open("student.dat",ios::in);
    while(fp.read((char*)&st,sizeof(student)))
    {
        if((strcmpi(st.retadmno(),n)==0))
        {
            st.show_student();
            flag=1;
        }
    }

 fp.close();
    if(flag==0)
            cout<<"\n\nStudent does not exist";
     getch();
}

//***************************************************************
//        functions to modify record of file
//***************************************************************

void modify_book()
```

```
char n[6];
    int found=0;
    cout<<"\n\n\n\n";
    cout<<"\n\n\tMODIFY BOOK REOCORD....
";
    cout<<"\n\n\tEnter The book no. of The
book";
    cin>>n;
    fp.open("book.dat",ios::in|ios::out);
    while(fp.read((char*)&bk,sizeof(book)) &&
found==0)
    {
        if(strcmpi(bk.retbno(),n)==0)
        {
            bk.show_book();
            cout<<"\nEnter The New Details of
book"<<endl;
            bk.modify_book();
            int pos=-1*sizeof(bk);
                fp.seekp(pos,ios::cur);
                fp.write((char*)&bk,sizeof(book));
                cout<<"\n\n\t Record Updated";
                found=1;
        }
    }
```

```
 fp.close();
        if(found==0)
            cout<<"\n\n Record Not Found ";
        getch();
}


void modify_student()
{
    char n[6];
    int found=0;
    cout<<"\n\n\n\n";
    cout<<"\n\n\tMODIFY STUDENT RECORD... ";
    cout<<"\n\n\tEnter The admission no. of The
student";
    cin>>n;
    fp.open("student.dat",ios::in|ios::out);
    while(fp.read((char*)&st,sizeof(student)) &&
found==0)
    {
        if(strcmpi(st.retadmno(),n)==0)
        {
st.show_student();
            cout<<"\nEnter The New Details of
student"<<endl;
            st.modify_student();
            int pos=-1*sizeof(st);
            fp.seekp(pos,ios::cur);
            fp.write((char*)&st,sizeof(student));
            cout<<"\n\n\t Record Updated";
            found=1;
```

```
 }
   }

   fp.close();
   if(found==0)
      cout<<"\n\n Record Not Found ";
   getch();
}

//***************************************************************
//        functions to delete record of file
//***************************************************************

void delete_student()
{
   char n[6];
   int flag=0;
   cout<<"\n\n\n\n";
      cout<<"\n\n\n\tDELETE STUDENT...";
      cout<<"\n\nEnter The admission no. of the Student You
Want To Delete : ";
      cin>>n;
      fp.open("student.dat",ios::in|ios::out);
      fstream fp2;
      fp2.open("Temp.dat",ios::out);
      fp.seekg(0,ios::beg);
      while(fp.read((char*)&st,sizeof(student)))
```

```
 {
       if(strcmpi(st.retadmno(),n)!=0)
              fp2.write((char*)&st,sizeof(student));
       else
             flag=1;
    }

    fp2.close();
       fp.close();
       remove("student.dat");
       rename("Temp.dat","student.dat");
       if(flag==1)
           cout<<"\n\n\tRecord Deleted ..";
       else
           cout<<"\n\nRecord not found";
       getch();
}


void delete_book()
{
    char n[6];
    cout<<"\n\n\n\n";
    cout<<"\n\n\n\tDELETE BOOK ...";
    cout<<"\n\nEnter The Book no. of the Book You Want To
Delete : ";
    cin>>n;
    fp.open("book.dat",ios::in|ios::out);
    fstream fp2;
    fp2.open("Temp.dat",ios::out);
    fp.seekg(0,ios::beg);
    while(fp.read((char*)&bk,sizeof(book)))
```

```
 fp2.close();
        fp.close();
        remove("book.dat");
        rename("Temp.dat","book.dat");
        cout<<"\n\n\tRecord Deleted ..";
        getch();
}

//***************************************************************
*****
//        function to display all students in list
//***************************************************************
******

void display_alls()
{
    cout<<"\n\n\n\n";
         fp.open("student.dat",ios::in);
         if(!fp)
         {
              cout<<"ERROR!!! FILE COULD NOT BE
OPEN ";
              getch();
              return;
         }

    cout<<"\n\n\t\tSTUDENT LIST\n\n";

cout<<"=================================
==================================\n";
    cout<<"\tAdmission
No."<<setw(10)<<"Name"<<setw(20)<<"Book
Issued\n";
```

```
while(fp.read((char*)&st,sizeof(student)))
    {
        st.report();
    }

    fp.close();
    getch();
}


//***************************************************************
//        function to display all Books in list
//***************************************************************

void display_allb()
{
    cout<<"\n\n\n\n";
    fp.open("book.dat",ios::in);
    if(!fp)
    {
        cout<<"ERROR!!! FILE COULD NOT BE OPEN ";
              getch();
              return;
         }
    cout<<"\n\n\t\tBook LIST\n\n";

cout<<"==========================================
==============================\n";
```

```
 cout<<"Book Number"<<setw(20)<<"Book
Name"<<setw(25)<<"Author\n";

cout<<"==========================================
==============================\n";

    while(fp.read((char*)&bk,sizeof(book)))
    {
        bk.report();
    }
         fp.close();
         getch();
}



//***************************************************************
//        function to issue a book
//***************************************************************

void book_issue()
{
    char sn[6],bn[6];
    int found=0,flag=0;
    cout<<"\n\n\n\n";
    cout<<"\n\nBOOK ISSUE ...";
    cout<<"\n\n\tEnter The student's admission no.";
    cin>>sn;
    fp.open("student.dat",ios::in|ios::out);
        fp1.open("book.dat",ios::in|ios::out);
        while(fp.read((char*)&st,sizeof(student)) && found==0)
           {
        if(strcmpi(st.retadmno(),sn)==0)
        {
```

```
 found=1;
          if(st.rettoken()==0)
          {
                cout<<"\n\n\tEnter the book no. ";
             cin>>bn;

while(fp1.read((char*)&bk,sizeof(book))&& flag==0)
             {
                  if(strcmpi(bk.retbno(),bn)==0)
                {
                    bk.show_book();
                    flag=1;
                    st.addtoken();
                    st.getstbno(bk.retbno());
                          int pos=-1*sizeof(st);
                    fp.seekp(pos,ios::cur);
                    fp.write((char*)&st,sizeof(student));
                    cout<<"\n\n\t Book issued
successfully\n\nPlease Note: Write current date in
backside of book and submit within 15 days fine Rs.
1 for each day after 15 days period";
                 }
                 }
 if(flag==0)
                    cout<<"Book no does not exist";
          }
             else
               cout<<"You have not returned the last
book ";
```

```
}
    }
         if(found==0)
        cout<<"Student record not exist...";
    getch();
      fp.close();
      fp1.close();
}

//***************************************************************
//        function to return a book
//***************************************************************

void book_deposit()
{
    char sn[6],bn[6];
    int found=0,flag=0,day,fine;
    cout<<"\n\n\n\n";
    cout<<"\n\nBOOK DEPOSIT ...";
    cout<<"\n\n\tEnter The student's admission no.";
    cin>>sn;
    fp.open("student.dat",ios::in|ios::out);
    fp1.open("book.dat",ios::in|ios::out);
    while(fp.read((char*)&st,sizeof(student)) && found==0)
       {
        if(strcmpi(st.retadmno(),sn)==0)
        {
            found=1;
            if(st.rettoken()==1)
            {
            while(fp1.read((char*)&bk,sizeof(book))&& flag==0)
            {
               if(strcmpi(bk.retbno(),st.retstbno())==0)
            {
```

```
bk.show_book();
                flag=1;
                cout<<"\n\nBook deposited in no. of days";
                cin>>day;
                if(day>15)
                {
                   fine=(day-15)*1;
                   cout<<"\n\nFine has to deposited Rs. "<<fine;
                }
                    st.resettoken();
                    int pos=-1*sizeof(st);
                    fp.seekp(pos,ios::cur);
                    fp.write((char*)&st,sizeof(student));
                    cout<<"\n\n\t Book deposited successfully";
             }
             }
           if(flag==0)
             cout<<"Book no does not exist";
              }
          else
             cout<<"No book is issued..please check!!";
        }
       }
      if(found==0)
    cout<<"Student record not exist...";
    getch();
  fp.close();
  fp1.close();
```

```
//***************************************************************
//        THE MAIN FUNCTION OF PROGRAM
//***************************************************************

int main()
{
    char ch;
    intro();
    do
    {
        cout<<"\n\n\n\n";
        cout<<"\n\n\n\tMAIN MENU";
        cout<<"\n\n\t01. BOOK ISSUE";
        cout<<"\n\n\t02. BOOK DEPOSIT";
          cout<<"\n\n\t03. ADMINISTRATOR MENU";
          cout<<"\n\n\t04. EXIT";
          cout<<"\n\n\tPlease Select Your Option (1-4) : ";
          ch=getche();
          switch(ch)
          {
            case '1':cout<<"\n\n\n\n";
                 book_issue();
                    break;
              case '2':book_deposit();
```

```
 break;
            case '3':admin_menu();
               break;
            case '4':exit(0);
            default :cout<<"\a";
        }
        }while(ch!='4');
        return 0;
}

//****************************************************************
//                END OF PROJECT
//****************************************************************
```

CONCLUSION:

Hence after the completion of the project we got familiar with C++ programming and its features. Complete and helpful library management can only be developed with a lot of intensive effort and time. Due to the lack of time and we are beginners in programming a program such as that can't be developed by us. Our library management may not be useful for most libraries in our college but it will be very useful for studying and programming practice using C++. As a whole, the project has been a good learning experience for us. We have gained knowledge about various aspects of C++ programming. At the same time, we have developed a deep understanding of file handling in C++. We still want to emphasize that the program is not complete by itself. There is still a lot of room for improvement. Graphics may be added to the program to make it more attractive. The mouse cursor may be initiated in order to make the program even more interactive.