Object Oriented Programming

Project Report

On

Library Management Application

Submitted by:

Vishal Sharma(9913103456)

Arjun Singhal (9913103528)

Akash Gupta (9913103459)

Akshit Mittal (9913103477)

Submitted to:

Mr. Raju Pal

(Dept. of CSE)

Acknowledgement

We would like to express our sincere gratitude to our teacher Mr. Raju Pal who gave us the golden opportunity to do this wonderful project on the topic Library Management Application, which also helped us in doing a lot of Research and we came to know about so many new things.

We would also like to thank our parents and friends who helped us a lot in finalizing this project within the limited time frame.

INDEX

- Sofware and Hardware configuration
- Introduction
- Flow of program
- Class diagram

Software and Hardware Requirements

Software configuration:

Operating System: Windows 95\NT\98\2000 and above.

Hardware configuration:

Processor: Pentium(R) Dual-core CPU

RAM : 256 MB or more

Hard Disk: 40 Gb(min)

MAJOR COMPONENTS OF THE APPLICATION

//HEADER FILES

fstream – used for data file handling

iostream- input output operations

cstdlib – used for accessing system(“cls”) function

cstdio – used for accessing getchar() and fflush(stdin)

cstring – string operations

windows.h – used for hiding user input

//CLASSES

```
class book
{
 char bname[50];          // Name of the book
 char aname[30];          // Name of the author of the book
 char isbn[10];           // isbn of the book
 int year;                // publication year
 int copies;              // no. of copies available in the
                          // library
```

```
 public:
 book();                        // constructor for book class
 void displaybook();            // display the book information
 char* getbname();              // accessor function that returns
                                // book name
 char* getisbn();           // accessor function that returns isbn
 int getcopies();           // accessor function that returns no. of
                            // copies of the book
 void bookinfoinput();      // function to input all book details from
                            // user
 void removecopy();         // decreases the no. of copies of the
                            // book by 1.
                            // called when a book is issued
 void addcopy();            // increases the no. of copies of the
                            // book by 1. called when an issued
                            // book is returned
 };


class LDATE
{
 int day;
 int month;
 int year;

 public:

    LDATE();          // constructor
    int getday();     // accessor function that returns day
    int getmonth();   // accessor function that returns month
    int getyear();    // accessor function that returns year
    void inputdate();      // accessor function that returns
                           // function to input date
};
```

```
class issuedbook       // class to maintain record of issued
books
{
 book binfo;                // book information
 char sname[30];          // name of the student issuing the book
 char studroll[11];          // roll no. of the student
 LDATE date;                // date of issuing

 public:

 void dispissued();         // display the details of an issued book
 char* getstudroll();        // accessor function for student rollno
 book getbinfo();         // accessor functin for binfo
 void setbinfo(book bo); // setter function for binfo
 void input();              // function to input details of issued book
};


class libraryrec         // class to be used as a linked list to store
{                        // record of all books in the library
    public:
       book info;
       libraryrec *next;
};


class issuerec             // class to be used as a linked list
{                          // to store record of all issued books
    public:
    issuedbook info;
    issuerec *next;
};

typedef libraryrec node;
typedef issuerec nodei;
```

```
//global objects

node *start=NULL,*ptr;
nodei *starti=NULL,*ptri;

book b;
issuedbook i;


//global function definitions

void displaylibraryrec();       // display details of all books
                                // in the library
void displayissuerec();         // display details of all books
                                // issued from the library
void addnew(book bookinfo);  // to add a new book to the
                                 // library
void issuebook();                // to issue a book
void returnbook();               // return an issued book
void removerec();       // remove record of a book from the
                        // library
void updaterec();       // update record of a book in the library
void lpasschange();     // change password of administrator
void spasschange()      // change password of student account
void librarianlog();    // administrator account
void studentlog();      // student account
void LMA();             // ask for password for desired account
                        // and redirects to the desired account
```

FLOW OF PROGRAM

There are 2 data files library.dat and issuerec.dat which are used to store the record of all books in the library and all issued books in the library.

First of all, all the data from library.dat is extracted and stored to
linked list libraryrec and from issuerec.dat to linked list issuerec.
Then, all operations (insertion, deletion and updation) are done on these two linked lists and not on the data files.

At the last, the data file library.dat is opened in output mode (its contents are truncated). Then all the data in linked list libraryrec is stored in this file. Same with issuerec.dat file (from issuerec linked list).

RELATIONSHIPS

- Composition
  The libraryrec class has a book object

- Aggregation
  The issuedbook class has a book object

- Association
  A book is issued on a particular date. So, issuedbook class is associated with a ldate

- Composition
  The libraryrec class has an issuedbook object

CLASS DIAGRAM