

**INTERNATIONAL DIPLOMA IN INFORMATION TECHNOLOGY & COMMUNICATION**

**C programming CPG-102**

**Student declaration:**

*I declare that:*

*• I understand what is meant by plagiarism*

*• The implication of plagiarism has been explained to me by my institution*

*• This assignment is all my own work and I have acknowledged any use of the published and unpublished works of other people.*

**Student’s signature: ……………………...... Date:……..............................................**

|  |  |  |
| --- | --- | --- |
|  | **Total number of pages including this cover page** | 68 |
| **Student Full Name** | MAHMUD S. ABDULLAHI | |
| **Student ID** | 2308-1503-0028 | |
| **Submission Date Due Date** | 17-05-2016 | |
| **Class Code / Group** | CPG 102 / GROUP TWO | |
| **Lecturer’s Name** | MR.BASHIR GARBASA | |

**Table of Contents**

CHAPTER ONE ........................................................................................................................................... 4

Introduction .............................................................................................................................................. 4

Current systems ........................................................................................................................................ 5

Proposed New System .............................................................................................................................. 6

System Specification/ Function Modules ................................................................................................. 7

Program (Input/ Output) Specification ………………………………………………………………………………………………. 8

Screen Design............................................................................................................................................ 9

Screen Design.......................................................................................................................................... 10

Processing and Validation ....................................................................................................................... 11

Messages used in the program ............................................................................................................... 12

CHAPTER TWO ........................................................................................................................................ 13

PROGRAM DESIGN PSEUDOCODE ...................................................................................................... 13

Main ........................................................................................................................................................ 14

Module Menu Screen ............................................................................................................................. 15

Module Select ......................................................................................................................................... 16

Module Add ............................................................................................................................................ 17

CHAPTER THREE .................................................................................................................................... 18

PROGRAM LISTING.................................................................................................................................. 18

CHAPTER FOUR....................................................................................................................................... 50

PROGRAM TESTING ............................................................................................................................... 50

TEST LOG ................................................................................................................................................. 50

TEST CASE................................................................................................................................................ 51

CHAPTER FIVE ........................................................................................................................................ 64

IMPLEMENTATION ................................................................................................................................. 64

Hardware and Software Configuration ................................................................................................... 64

USER MANUAL ........................................................................................................................................ 65

Add menu screen .................................................................................................................................... 61

CHAPTER SIX ........................................................................................................................................... 67

CONCLUSION........................................................................................................................................... 67

Program weakness .................................................................................................................................. 67

Program Strength.................................................................................................................................... 67

Program Enhancement ........................................................................................................................... 67

Reference ................................................................................................................................................ 68

**CHAPTER ONE**

**INTRODUCTION**

The Adamawa State library was established in the year 1968, by the Adamawa State Ministry of Education; the library is located in Yola city in Adamawa State. Adamawa State library is one of the great libraries in the state where people from different places and almost all the student in the state come to read and borrow some of the reading materials.

The establishment of this library is to develop the educational standards of the young upcoming generation in Adamawa State particular and Nigeria in general. The library has a set of books reserved in its shield. Many people come and read in the library to get knowledge and most population in the world are saying that “knowledge is power”

The project arrangement is complete and reliable which doesn’t need the user to be trained in the use of this new system in which they will be used to increase the daily operations of the state library. Since the establishment of this library there is no any development that come across to establish the system into the new one that reduce the number of problems that the library is pacing always and they are continuously occurring which lead to the reduction of most of the people that are using the library in advance the. library need to be organised in a modern way that will suit the basic need of the library users and will reduce the number of congestion and some lapses that are occurring in the library and this will make many of the users to have less time wastage not only the users but the workers will find it easier to organise their day to day activities in the library in a more simpler way and easier so that people will not be complaining or having congestion in the library. For the library to engage in this system will make their work to be accurate and diligence and will reduce the number of complaining as far as the library is concern. The program will be created with use of a third generation language called C Programming and software named Borland C++, or Turbo C++.

**CURRENT SYSTEM**

When I visited the Adamawa State library, I had a conversation with the chief Liberian which I found out that activities conducted in the library are executed with the use of pen and paper. For example; information of individuals who come to the library to make use of or borrow books his/her records are been kept which have no copy or backup in the case of loss or damage of such record.

These records is filling by registration of borrowed books, names of the borrowers, their addresses, date and time due to return, and serial number of books all this are done on paper by hand, if there’s any mistake is made when lending the book out of the library, it may not be quickly detected.

All their daily operations are performed manually which makes their transaction slow and complex. Due to these processes, they lose most of their incoming students due to time wasted during the registration process. the library need to be organised in a modern way that will suit the basic need of the library users and will reduce the number of congestion and some lapses that are occurring in the library. Also the staff’s members will find it easy to consult people in easier way instead of them to be giving whatever receipt or registering people manually this system will make them to do their work in a simple and cleared way. this system will more better for the library since people will always register on line without any waste of time because to register manually in the library by the library workers this will consume much of time and will not allow many people do that at a perfect time since they are doing anything manually but with the use of modern technology this will be easier for them to access whatever the library need to do at a particular time.

**PROPOSE NEW SYSTEM**

The planned new system will enable the library’s staffs to defeat the inconveniences faced by the library. The new arrangement is dependable because it is a programmed program. The system program need a Borland C++ or Turbo C++ software program install before it work.

However in the new plan method, there is requirement for the storage space of books name and serial number, name and address of the borrowers. Around will be no more time waste by attend individuals who utilize the library. This program will also allow the user to add record, search record, display record and delete record of regulars. This new program also will give the Adamawa state library a standard because of the computerised means of operation it uses.

This system will also allow people to use the library in a more convenient way and they will satisfy with the system since people now a days doesn’t want to spend much of their time in every aspect of duty. Also the staff’s members will find it easy to consult people in easier way instead of them to be giving whatever receipt or registering people manually this system will make them to do their work in a simple and cleared way which will make their work accurately and diligence.

The system will make many library users to learn how to use computer for their aspect of reading, researching, and downloading many books in the internet which will enhance aspect of reading as well as learning factor.

Many people from several areas will imprecise most of their searching aspect to be the library by consulting the library most of time for their purpose of study or learning culture for their own benefit.

This will suit the need of most users since the system has been computerised and its new and modern system almost all over the world or developed countries

**SYSTEM SPECIFICATION/ FUNCTION MODULES**

The programming language that will be used in coding this work is C programming language, Which can be developed modules of program running, firstly the introduction of the program is displayed and then the main menu which is made up of:

**ADD:** this allows you to enter the information of new student

**EDIT:** this allow you to edit some information of student whose want change something in his record.

**SEARCH:** this allow you to search a student record, and also for the reference to the staffs if they want to view a details of the student in the library

**DELETE:** this allow you to delete to a record of the student when needed.

**EXIT:** if the user want to go for break or is a closing time for the day work then this module will allows the user to quit from the system and close.

**PROGRAM (INPUT AND OUTPUT /OUTPUT) SPECIFICATIONS**

**Input and out Specification**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| SERIAL NUMBER | ITEM NAME | DESCRIPTION | DATA TYPE | SIZE |
| 1 | Add books | To store the record of the customer. | String | 10 |
| 2 | Author’s name | Stores the name of the book’s author. | String | 30 |
| 3 | Registration number | Holds registration number allocated to the customer. | Int | 0-100 |
| 4 | Address | Holds the address of the customer. | String | 30 |
| 5 | Day | Specifies the day customer intends to return book. | Int | 1-91 |
| 6 | Serial number | Holds serial number allocated to the customer. | String | 27 |

# 

# **SCREEN DESIGN**

The filling output designed tables below, shows following from the main menu to add new member page, search for a member page, delete record page, borrow book to a member page, and view books borrowed by member module.

**+-----------------------------------+**

**+ ADAMAWA STATE LIBRARY+**

**Main Menu**

**+-----------------------------------+**

**Press [1] add books**

**Press [2] delete books**

**Press [3] Search Books**

**Press [4] Issue books**

**Press [5] view book list**

**Press [6] edit book’s record**

**Press [7] close application**

**Enter Menu Number Here:**

\*----------------------------------------------------------\*

**SELECT CATEGORIES**

**\*---------------------------------------------------------\***

1. **Computer**
2. **Electronics**
3. **Electrical**
4. **Civil**
5. **Mechanical**
6. **Architecture**
7. **Back to main Menu**

**+---------------------------------------------+**

**|\*\*\*\*\*\*\*\*\*\*\*\*search books\*\*\*\*\*\*\*\*\*\*\*\*\*\* |**

**+----------------------------------------------------------------------------+**

1. **Search by id:**
2. **Search by name:**

**Enter search**

**+-----------------------------------+**

**+ ADAMAWA STATE LIBRARY+**

**Main Menu**

**+-----------------------------------+**

**Press [1] add books**

**Press [2] delete books**

**Press [3] Search Books**

**Press [4] Issue books**

**Press [5] view book list**

**Press [6] edit book’s record**

**Press [7] close application**

**Enter Menu Number Here:**

**+-----------------------------------+**

**+ ADAMAWA STATE LIBRARY+**

**Main Menu**

**+-----------------------------------+**

**Press [1] add books**

**Press [2] delete books**

**Press [3] Search Books**

**Press [4] Issue books**

**Press [5] view book list**

**Press [6] edit book’s record**

**Press [7] close application**

**Enter Menu Number Here:**

**+------------------------------------------------------------+**

**ADAMAWA STATE LIBRARY SEARCH MEMBER OPTION|**

**+------------------------------------------------------------+**

**Enter the book id to delete:**

# 

**+-----------------------------------+**

**\*\*\*\*\*\*\*ISSUE SECTION\*\*\*\*\*\***

**+-----------------------------------+**

1. **Issue a book**
2. **View issued book**
3. **Search issued book**
4. **Remove issued book**

**Enter a choice:**

**+-----------------------------------+**

**+ ADAMAWA STATE LIBRARY+**

**Main Menu**

**+-----------------------------------+**

**Press [1] add books**

**Press [2] delete books**

**Press [3] Search Books**

**Press [4] Issue books**

**Press [5] view book list**

**Press [6] edit book’s record**

**Press [7] close application**

**Enter Menu Number Here:**

**+-----------------------------------+**

**BOOK RETURNEE FORM**

**+-----------------------------------+**

**CATEGORY ID BOOK NAME AUTHOR QTY PRICE RackNo**

**Total Books =0**

**Press ENTER to return to main menu**

**+-----------------------------------+**

**+ ADAMAWA STATE LIBRARY+**

**Main Menu**

**+-----------------------------------+**

**Press [1] add books**

**Press [2] delete books**

**Press [3] Search Books**

**Press [4] Issue books**

**Press [5] view book list**

**Press [6] edit book’s record**

**Press [7] close application**

**Enter Menu Number Here:**

**Enter Book Id to be edited:**

**+-----------------------------------+**

**+ ADAMAWA STATE LIBRARY+**

**Main Menu**

**+-----------------------------------+**

**Press [1] add books**

**Press [2] delete books**

**Press [3] Search Books**

**Press [4] Issue books**

**Press [5] view book list**

**Press [6] edit book’s record**

**Press [7] close application**

**Enter Menu Number Here:**

**ADAMAWA STATE LIBRARY MANAGEMENT SYSTEM**

**Have a nice day**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**Are you sure you want to exit...........> press any key to continue**

**+-----------------------------------+**

**+ ADAMAWA STATE LIBRARY+**

**Main Menu**

**+-----------------------------------+**

**Press [1] add books**

**Press [2] delete books**

**Press [3] Search Books**

**Press [4] Issue books**

**Press [5] view book list**

**Press [6] edit book’s record**

**Press [7] close application**

**Enter Menu Number Here:**

**+-----------------------------------+**

**+ ADAMAWA STATE LIBRARY+**

**Main Menu**

**+-----------------------------------+**

**Press [1] add books**

**Press [2] delete books**

**Press [3] Search Books**

**Press [4] Issue books**

**Press [5] view book list**

**Press [6] edit book’s record**

**Press [7] close application**

**Enter Menu Number Here:**

# 

# PROCESSING & VALIDATIONS

1. **Options**

* The acceptable entry for the choices is from 1 – 7. If entry is not within the range, then error message pops up **“Wrong Entry!!Please re-entered correct option.”**

1. **Add books**

* Before adding books there are 6 categories of options to choose from, example all computer related books most be added to computer category. Valid choices is 1-6 if any number exceed number 6 then it will automatically take you to book information insertion without book category details

1. **Delete books**

* Deleting books requires the insertion of correct book ID, without accurate book ID this error will pops up **“No record is found modify the search”**

1. **Search Books**

* The acceptable options on search section is by book ID or book Name, without proper name or ID error message will pops up **“No Record Found”**

1. **Edit Book’s Record**

* Unique book ID need to be inserted to modify a book wrong insertion of book will result in error message will pops up **“Modify another Record?(Y/N) Press ENTER to return to main menu”**

**Messages used in the program**

Main menu module

* 1. Add Books
* 2. Delete books
* 3. Search Books
* 4. Issue Books
* 5. View Book list
* 6. Edit Book's Record
* 7. Close Application

If you enter any number except the above one it will pop up a message **“Wrong Entry!!Please re-entered correct option.”**

**Add Books**

* 1. Computer
* 2. Electronics
* 3. Electrical
* 4. Civil
* 5. Mechanical
* 6. Architecture
* 7. Back to main menu

To add a new book record the user most select from the list from 1 to 6 in other to insert book information after that new window will pop up with **“book ID: Book name: Author: quantity: price: rack no: “**

If all information has been inserted correctly then this message will be displayed **“The record is successfully saved**

**Save any more?(Y / N):”**

**Search Books**

* 1. Search by ID
* 2. Search by Name
* **Try another search?(Y/N)**

**CHAPTER TWO**

**PROGRAM AND DESIGN PSEUDOCODE**

**Main**

*DO*

CALL Menu Screen

CALL Select

ENDDO

**Module Menu Screen**

DO

***DISPLAY options***

ENDDO

**Module Select**

DO

CALL Menu Screen

DO WHILE choice not equal to 0

GET choice

DO CASE

choice 1

DO

CALL Add books

ENDDO

choice 2

DO

CALL Delete books

ENDDO

choice 3

DO

CALL Search books

ENDDO

choice 4

DO

CALL issue books

ENDDO

choice 5

DO

CALL view book list

ENDDO

choice 6

DO

CALL edit book’s record

ENDDO

choice 7

close application

OTHERWISE DO

DISPLAY “Wrong Entry!!Please re-entered correct option”

ENDDO ENDCASE

ENDO ENDDO **Module Add**

DO

REPEAT UNTIL input is valid

DO

CALL Input String for computer

CALL Input String for electronics

CALL Input Value for electrical

CALL Input Value for civil

CALL Input Value for mechanical

CALL Input Value for architecture

CALL Input Value for back to main menu

ENDDO ENDDO

If record entered for the specified categories THEN DO

CALL Write File

CLOSE

ENDDO ENDIF

ENDDO

MODULE DELETE BOOK

DO

CALL Input value for Book ID

IF ID found

DO

DISPLAY Report

ENDDO

ELSE

DO

DISPLAY “No record is found modify the search”

ENDDO

ENDIF

ENDIF

ENDDO

MODULE DELETE BOOK

DO

CALL Input value for Book ID

IF ID found

DO

DISPLAY Report

ENDDO

ELSE

DO

DISPLAY “No record is found modify the search”

ENDDO

ENDIF

ENDIF

ENDDO

**CHAPTER 3**

PROGRAM LISTENING:

#include<windows.h>

#include<stdio.h>

#include<conio.h>

#include <stdlib.h>

#include<string.h> //contains strcmp(),strcpy(),strlen(),etc

#include<ctype.h> //contains toupper(), tolower(),etc

#include<dos.h> //contains \_dos\_getdate

#include<time.h>

//#include<bios.h>

#define RETURNTIME 15

char catagories[][15]={"Computer","Electronics","Electrical","Civil","Mechnnical","Architecture"};

void returnfunc(void);

void mainmenu(void);

void addbooks(void);

void deletebooks(void);

void editbooks(void);

void searchbooks(void);

void issuebooks(void);

void viewbooks(void);

void closeapplication(void);

int getdata();

int checkid(int);

int t(void);

//void show\_mouse(void);

void Password();

void issuerecord();

void loaderanim();

//list of global files that can be acceed form anywhere in program

FILE \*fp,\*ft,\*fs;

COORD coord = {0, 0};

//list of global variable

int s;

char findbook;

char password[10]={"mahmudabds"};

void gotoxy (int x, int y)

{

coord.X = x; coord.Y = y; // X and Y coordinates

SetConsoleCursorPosition(GetStdHandle(STD\_OUTPUT\_HANDLE), coord);

}

struct meroDate

{

int mm,dd,yy;

};

struct books

{

int id;

char stname[20];

char name[20];

char Author[20];

int quantity;

float Price;

int count;

int rackno;

char \*cat;

struct meroDate issued;

struct meroDate duedate;

};

struct books a;

int main()

{

Password();

getch();

return 0;

}

void mainmenu()

{

//loaderanim();

system("cls");

// textbackground(13);

int i;

gotoxy(20,3);

printf("\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2 WELCOME TO ADAMAWA STATE LIBRARY: MAIN MENU \xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2");

// show\_mouse();

gotoxy(20,5);

printf("\xDB\xDB\xDB\xDB\xB2 1. Add Books ");

gotoxy(20,7);

printf("\xDB\xDB\xDB\xDB\xB2 2. Delete books");

gotoxy(20,9);

printf("\xDB\xDB\xDB\xDB\xB2 3. Search Books");

gotoxy(20,11);

printf("\xDB\xDB\xDB\xDB\xB2 4. Issue Books");

gotoxy(20,13);

printf("\xDB\xDB\xDB\xDB\xB2 5. View Book list");

gotoxy(20,15);

printf("\xDB\xDB\xDB\xDB\xB2 6. Edit Book's Record");

gotoxy(20,17);

printf("\xDB\xDB\xDB\xDB\xB2 7. Close Application");

gotoxy(20,19);

printf("\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2");

gotoxy(20,20);

t();

gotoxy(20,21);

printf("Enter your choice:");

switch(getch())

{

case '1':

addbooks();

break;

case '2':

deletebooks();

break;

case '3':

searchbooks();

break;

case '4':

issuebooks();

break;

case '5':

viewbooks();

break;

case '6':

editbooks();

break;

case '7':

{

system("cls");

gotoxy(16,3);

printf("\tADAMAWA STATE LIBRARY MANAGEMENT SYSTEM");

gotoxy(16,4);

printf("\tHave a nice day");

gotoxy(16,7);

printf("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

gotoxy(16,10);

printf("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

gotoxy(16,11);

printf("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

gotoxy(16,13);

printf("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

gotoxy(10,17);

printf("Exiting in 3 second...........>");

//flushall();

Sleep(3000);

exit(0);

}

default:

{

gotoxy(10,23);

printf("\aWrong Entry!!Please re-entered correct option");

if(getch())

mainmenu();

}

}

}

void addbooks(void) //funtion that add books

{

system("cls");

int i;

gotoxy(20,5);

printf("\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2SELECT CATEGORIES\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2");

gotoxy(20,7);

printf("\xDB\xDB\xDB\xDB\xB2 1. Computer");

gotoxy(20,9);

printf("\xDB\xDB\xDB\xDB\xB2 2. Electronics");

gotoxy(20,11);

printf("\xDB\xDB\xDB\xDB\xB2 3. Electrical");

gotoxy(20,13);

printf("\xDB\xDB\xDB\xDB\xB2 4. Civil");

gotoxy(20,15);

printf("\xDB\xDB\xDB\xDB\xB2 5. Mechanical");

gotoxy(20,17);

printf("\xDB\xDB\xDB\xDB\xB2 6. Architecture");

gotoxy(20,19);

printf("\xDB\xDB\xDB\xDB\xB2 7. Back to main menu");

gotoxy(20,21);

printf("\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2");

gotoxy(20,22);

printf("Enter your choice:");

scanf("%d",&s);

if(s==7)

mainmenu() ;

system("cls");

fp=fopen("Bibek.dat","ab+");

if(getdata()==1)

{

a.cat=catagories[s-1];

fseek(fp,0,SEEK\_END);

fwrite(&a,sizeof(a),1,fp);

fclose(fp);

gotoxy(21,14);

printf("The record is sucessfully saved");

gotoxy(21,15);

printf("Save any more?(Y / N):");

if(getch()=='n')

mainmenu();

else

system("cls");

addbooks();

}

}

void deletebooks() //function that delete items from file fp

{

system("cls");

int d;

char another='y';

while(another=='y') //infinite loop

{

system("cls");

gotoxy(10,5);

printf("Enter the Book ID to delete:");

scanf("%d",&d);

fp=fopen("Bibek.dat","rb+");

rewind(fp);

while(fread(&a,sizeof(a),1,fp)==1)

{

if(a.id==d)

{

gotoxy(10,7);

printf("The book record is available");

gotoxy(10,8);

printf("Book name is %s",a.name);

gotoxy(10,9);

printf("Rack No. is %d",a.rackno);

findbook='t';

}

}

if(findbook!='t')

{

gotoxy(10,10);

printf("No record is found modify the search");

if(getch())

mainmenu();

}

if(findbook=='t' )

{

gotoxy(10,9);

printf("Do you want to delete it?(Y/N):");

if(getch()=='y')

{

ft=fopen("test.dat","wb+"); //temporary file for delete

rewind(fp);

while(fread(&a,sizeof(a),1,fp)==1)

{

if(a.id!=d)

{

fseek(ft,0,SEEK\_CUR);

fwrite(&a,sizeof(a),1,ft); //write all in tempory file except that

} //we want to delete

}

fclose(ft);

fclose(fp);

remove("Bibek.dat");

rename("test.dat","Bibek.dat"); //copy all item from temporary file to fp except that

fp=fopen("Bibek.dat","rb+"); //we want to delete

if(findbook=='t')

{

gotoxy(10,10);

printf("The record is sucessfully deleted");

gotoxy(10,11);

printf("Delete another record?(Y/N)");

}

}

else

mainmenu();

fflush(stdin);

another=getch();

}

}

gotoxy(10,15);

mainmenu();

}

void searchbooks()

{

system("cls");

int d;

printf("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Search Books\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

gotoxy(20,10);

printf("\xDB\xDB\xDB\xB2 1. Search By ID");

gotoxy(20,14);

printf("\xDB\xDB\xDB\xB2 2. Search By Name");

gotoxy( 15,20);

printf("Enter Your Choice");

fp=fopen("Bibek.dat","rb+"); //open file for reading propose

rewind(fp); //move pointer at the begining of file

switch(getch())

{

case '1':

{

system("cls");

gotoxy(25,4);

printf("\*\*\*\*Search Books By Id\*\*\*\*");

gotoxy(20,5);

printf("Enter the book id:");

scanf("%d",&d);

gotoxy(20,7);

printf("Searching........");

while(fread(&a,sizeof(a),1,fp)==1)

{

if(a.id==d)

{

Sleep(2);

gotoxy(20,7);

printf("The Book is available");

gotoxy(20,8);

printf("\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2");

gotoxy(20,9);

printf("\xB2 ID:%d",a.id);gotoxy(47,9);printf("\xB2");

gotoxy(20,10);

printf("\xB2 Name:%s",a.name);gotoxy(47,10);printf("\xB2");

gotoxy(20,11);

printf("\xB2 Author:%s ",a.Author);gotoxy(47,11);printf("\xB2");

gotoxy(20,12);

printf("\xB2 Qantity:%d ",a.quantity);gotoxy(47,12);printf("\xB2"); gotoxy(47,11);printf("\xB2");

gotoxy(20,13);

printf("\xB2 Price:Rs.%.2f",a.Price);gotoxy(47,13);printf("\xB2");

gotoxy(20,14);

printf("\xB2 Rack No:%d ",a.rackno);gotoxy(47,14);printf("\xB2");

gotoxy(20,15);

printf("\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2");

findbook='t';

}

}

if(findbook!='t') //checks whether conditiion enters inside loop or not

{

gotoxy(20,8);

printf("\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2");

gotoxy(20,9);printf("\xB2"); gotoxy(38,9);printf("\xB2");

gotoxy(20,10);

printf("\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2");

gotoxy(22,9);printf("\aNo Record Found");

}

gotoxy(20,17);

printf("Try another search?(Y/N)");

if(getch()=='y')

searchbooks();

else

mainmenu();

break;

}

case '2':

{

char s[15];

system("cls");

gotoxy(25,4);

printf("\*\*\*\*Search Books By Name\*\*\*\*");

gotoxy(20,5);

printf("Enter Book Name:");

scanf("%s",s);

int d=0;

while(fread(&a,sizeof(a),1,fp)==1)

{

if(strcmp(a.name,(s))==0) //checks whether a.name is equal to s or not

{

gotoxy(20,7);

printf("The Book is available");

gotoxy(20,8);

printf("\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2");

gotoxy(20,9);

printf("\xB2 ID:%d",a.id);gotoxy(47,9);printf("\xB2");

gotoxy(20,10);

printf("\xB2 Name:%s",a.name);gotoxy(47,10);printf("\xB2");

gotoxy(20,11);

printf("\xB2 Author:%s",a.Author);gotoxy(47,11);printf("\xB2");

gotoxy(20,12);

printf("\xB2 Qantity:%d",a.quantity);gotoxy(47,12);printf("\xB2");

gotoxy(20,13);

printf("\xB2 Price:Rs.%.2f",a.Price);gotoxy(47,13);printf("\xB2");

gotoxy(20,14);

printf("\xB2 Rack No:%d ",a.rackno);gotoxy(47,14);printf("\xB2");

gotoxy(20,15);

printf("\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2");

d++;

}

}

if(d==0)

{

gotoxy(20,8);

printf("\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2");

gotoxy(20,9);printf("\xB2"); gotoxy(38,9);printf("\xB2");

gotoxy(20,10);

printf("\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2");

gotoxy(22,9);printf("\aNo Record Found");

}

gotoxy(20,17);

printf("Try another search?(Y/N)");

if(getch()=='y')

searchbooks();

else

mainmenu();

break;

}

default :

getch();

searchbooks();

}

fclose(fp);

}

void issuebooks(void) //function that issue books from library

{

int t;

system("cls");

printf("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*ISSUE SECTION\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

gotoxy(10,5);

printf("\xDB\xDB\xDB\xDb\xB2 1. Issue a Book");

gotoxy(10,7);

printf("\xDB\xDB\xDB\xDb\xB2 2. View Issued Book");

gotoxy(10,9);

printf("\xDB\xDB\xDB\xDb\xB2 3. Search Issued Book");

gotoxy(10,11);

printf("\xDB\xDB\xDB\xDb\xB2 4. Remove Issued Book");

gotoxy(10,14);

printf("Enter a Choice:");

switch(getch())

{

case '1': //issue book

{

system("cls");

int c=0;

char another='y';

while(another=='y')

{

system("cls");

gotoxy(15,4);

printf("\*\*\*Issue Book section\*\*\*");

gotoxy(10,6);

printf("Enter the Book Id:");

scanf("%d",&t);

fp=fopen("Bibek.dat","rb");

fs=fopen("Issue.dat","ab+");

if(checkid(t)==0) //issues those which are present in library

{

gotoxy(10,8);

printf("The book record is available");

gotoxy(10,9);

printf("There are %d unissued books in library ",a.quantity);

gotoxy(10,10);

printf("The name of book is %s",a.name);

gotoxy(10,11);

printf("Enter student name:");

scanf("%s",a.stname);

//struct dosdate\_t d; //for current date

//\_dos\_getdate(&d);

//a.issued.dd=d.day;

//a.issued.mm=d.month;

//a.issued.yy=d.year;

gotoxy(10,12);

printf("Issued date=%d-%d-%d",a.issued.dd,a.issued.mm,a.issued.yy);

gotoxy(10,13);

printf("The BOOK of ID %d is issued",a.id);

a.duedate.dd=a.issued.dd+RETURNTIME; //for return date

a.duedate.mm=a.issued.mm;

a.duedate.yy=a.issued.yy;

if(a.duedate.dd>30)

{

a.duedate.mm+=a.duedate.dd/30;

a.duedate.dd-=30;

}

if(a.duedate.mm>12)

{

a.duedate.yy+=a.duedate.mm/12;

a.duedate.mm-=12;

}

gotoxy(10,14);

printf("To be return:%d-%d-%d",a.duedate.dd,a.duedate.mm,a.duedate.yy);

fseek(fs,sizeof(a),SEEK\_END);

fwrite(&a,sizeof(a),1,fs);

fclose(fs);

c=1;

}

if(c==0)

{

gotoxy(10,11);

printf("No record found");

}

gotoxy(10,15);

printf("Issue any more(Y/N):");

fflush(stdin);

another=getche();

fclose(fp);

}

break;

}

case '2': //show issued book list

{

system("cls");

int j=4;

printf("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Issued book list\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

gotoxy(2,2);

printf("STUDENT NAME CATEGORY ID BOOK NAME ISSUED DATE RETURN DATE");

fs=fopen("Issue.dat","rb");

while(fread(&a,sizeof(a),1,fs)==1)

{

gotoxy(2,j);

printf("%s",a.stname);

gotoxy(18,j);

printf("%s",a.cat);

gotoxy(30,j);

printf("%d",a.id);

gotoxy(36,j);

printf("%s",a.name);

gotoxy(51,j);

printf("%d-%d-%d",a.issued.dd,a.issued.mm,a.issued.yy );

gotoxy(65,j);

printf("%d-%d-%d",a.duedate.dd,a.duedate.mm,a.duedate.yy);

//struct dosdate\_t d;

//\_dos\_getdate(&d);

gotoxy(50,25);

// printf("Current date=%d-%d-%d",d.day,d.month,d.year);

j++;

}

fclose(fs);

gotoxy(1,25);

returnfunc();

}

break;

case '3': //search issued books by id

{

system("cls");

gotoxy(10,6);

printf("Enter Book ID:");

int p,c=0;

char another='y';

while(another=='y')

{

scanf("%d",&p);

fs=fopen("Issue.dat","rb");

while(fread(&a,sizeof(a),1,fs)==1)

{

if(a.id==p)

{

issuerecord();

gotoxy(10,12);

printf("Press any key.......");

getch();

issuerecord();

c=1;

}

}

fflush(stdin);

fclose(fs);

if(c==0)

{

gotoxy(10,8);

printf("No Record Found");

}

gotoxy(10,13);

printf("Try Another Search?(Y/N)");

another=getch();

}

}

break;

case '4': //remove issued books from list

{

system("cls");

int b;

FILE \*fg; //declaration of temporary file for delete

char another='y';

while(another=='y')

{

gotoxy(10,5);

printf("Enter book id to remove:");

scanf("%d",&b);

fs=fopen("Issue.dat","rb+");

while(fread(&a,sizeof(a),1,fs)==1)

{

if(a.id==b)

{

issuerecord();

findbook='t';

}

if(findbook=='t')

{

gotoxy(10,12);

printf("Do You Want to Remove it?(Y/N)");

if(getch()=='y')

{

fg=fopen("record.dat","wb+");

rewind(fs);

while(fread(&a,sizeof(a),1,fs)==1)

{

if(a.id!=b)

{

fseek(fs,0,SEEK\_CUR);

fwrite(&a,sizeof(a),1,fg);

}

}

fclose(fs);

fclose(fg);

remove("Issue.dat");

rename("record.dat","Issue.dat");

gotoxy(10,14);

printf("The issued book is removed from list");

}

}

if(findbook!='t')

{

gotoxy(10,15);

printf("No Record Found");

}

}

gotoxy(10,16);

printf("Delete any more?(Y/N)");

another=getch();

}

}

default:

gotoxy(10,18);

printf("\aWrong Entry!!");

getch();

issuebooks();

break;

}

gotoxy(1,30);

returnfunc();

}

void viewbooks(void) //show the list of book persists in library

{

int i=0,j;

system("cls");

gotoxy(1,1);

printf("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Book List\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

gotoxy(2,2);

printf(" CATEGORY ID BOOK NAME AUTHOR QTY Renting RackNo ");

j=4;

fp=fopen("Bibek.dat","rb");

while(fread(&a,sizeof(a),1,fp)==1)

{

gotoxy(3,j);

printf("%s",a.cat);

gotoxy(16,j);

printf("%d",a.id);

gotoxy(22,j);

printf("%s",a.name);

gotoxy(36,j);

printf("%s",a.Author);

gotoxy(50,j);

printf("%d",a.quantity);

gotoxy(57,j);

printf("%.2f",a.Price);

gotoxy(69,j);

printf("%d",a.rackno);

printf("\n\n");

j++;

i=i+a.quantity;

}

gotoxy(3,25);

printf("Total Books =%d",i);

fclose(fp);

gotoxy(35,25);

returnfunc();

}

void editbooks(void) //edit information about book

{

system("cls");

int c=0;

int d,e;

gotoxy(20,4);

printf("\*\*\*\*Edit Books Section\*\*\*\*");

char another='y';

while(another=='y')

{

system("cls");

gotoxy(15,6);

printf("Enter Book Id to be edited:");

scanf("%d",&d);

fp=fopen("Bibek.dat","rb+");

while(fread(&a,sizeof(a),1,fp)==1)

{

if(checkid(d)==0)

{

gotoxy(15,7);

printf("The book is availble");

gotoxy(15,8);

printf("The Book ID:%d",a.id);

gotoxy(15,9);

printf("Enter new name:");scanf("%s",a.name);

gotoxy(15,10);

printf("Enter new Author:");scanf("%s",a.Author);

gotoxy(15,11);

printf("Enter new quantity:");scanf("%d",&a.quantity);

gotoxy(15,12);

printf("Enter new price:");scanf("%f",&a.Price);

gotoxy(15,13);

printf("Enter new rackno:");scanf("%d",&a.rackno);

gotoxy(15,14);

printf("The record is modified");

fseek(fp,ftell(fp)-sizeof(a),0);

fwrite(&a,sizeof(a),1,fp);

fclose(fp);

c=1;

}

if(c==0)

{

gotoxy(15,9);

printf("No record found");

}

}

gotoxy(15,16);

printf("Modify another Record?(Y/N)");

fflush(stdin);

another=getch() ;

}

returnfunc();

}

void returnfunc(void)

{

{

printf(" Press ENTER to return to main menu");

}

a:

if(getch()==13) //allow only use of enter

mainmenu();

else

goto a;

}

int getdata()

{

int t;

gotoxy(20,3);printf("Enter the Information Below");

gotoxy(20,4);printf("\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2");

gotoxy(20,5);

printf("\xB2");gotoxy(46,5);printf("\xB2");

gotoxy(20,6);

printf("\xB2");gotoxy(46,6);printf("\xB2");

gotoxy(20,7);

printf("\xB2");gotoxy(46,7);printf("\xB2");

gotoxy(20,8);

printf("\xB2");gotoxy(46,8);printf("\xB2");

gotoxy(20,9);

printf("\xB2");gotoxy(46,9);printf("\xB2");

gotoxy(20,10);

printf("\xB2");gotoxy(46,10);printf("\xB2");

gotoxy(20,11);

printf("\xB2");gotoxy(46,11);printf("\xB2");

gotoxy(20,12);

printf("\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2");

gotoxy(21,5);

printf("Category:");

gotoxy(31,5);

printf("%s",catagories[s-1]);

gotoxy(21,6);

printf("Book ID:\t");

gotoxy(30,6);

scanf("%d",&t);

if(checkid(t) == 0)

{

gotoxy(21,13);

printf("\aThe book id already exists\a");

getch();

mainmenu();

return 0;

}

a.id=t;

gotoxy(21,7);

printf("Book Name:");gotoxy(33,7);

scanf("%s",a.name);

gotoxy(21,8);

printf("Author:");gotoxy(30,8);

scanf("%s",a.Author);

gotoxy(21,9);

printf("Quantity:");gotoxy(31,9);

scanf("%d",&a.quantity);

gotoxy(21,10);

printf("Price:");gotoxy(28,10);

scanf("%f",&a.Price);

gotoxy(21,11);

printf("Rack No:");gotoxy(30,11);

scanf("%d",&a.rackno);

return 1;

}

int checkid(int t) //check whether the book is exist in library or not

{

rewind(fp);

while(fread(&a,sizeof(a),1,fp)==1)

if(a.id==t)

return 0; //returns 0 if book exits

return 1; //return 1 if it not

}

int t(void) //for time

{

time\_t t;

time(&t);

printf("Date and time:%s\n",ctime(&t));

return 0 ;

}

/\*void show\_mouse(void) //show inactive mouse pointer in programme

{

union REGS in,out;

in.x.ax = 0x1;

int86(0x33,&in,&out);

}\*/

void Password(void) //for password option

{

system("cls");

char d[25]="Password Protected";

char ch,pass[10];

int i=0,j;

//textbackground(WHITE);

//textcolor(RED);

gotoxy(10,4);

for(j=0;j<20;j++)

{

Sleep(50);

printf("\*");

}

for(j=0;j<20;j++)

{

Sleep(50);

printf("%c",d[j]);

}

for(j=0;j<20;j++)

{

Sleep(50);

printf("\*");

}

gotoxy(10,10);

gotoxy(15,7);

printf("Enter Password:");

while(ch!=13)

{

ch=getch();

if(ch!=13 && ch!=8){

putch('\*');

pass[i] = ch;

i++;

}

}

pass[i] = '\0';

if(strcmp(pass,password)==0)

{

gotoxy(15,9);

//textcolor(BLINK);

printf("Password match");

gotoxy(17,10);

printf("Press any key to countinue.....");

getch();

mainmenu();

}

else

{

gotoxy(15,16);

printf("\aWarning!! Incorrect Password");

getch();

Password();

}

}

void issuerecord() //display issued book's information

{

system("cls");

gotoxy(10,8);

printf("The Book has taken by Mr. %s",a.stname);

gotoxy(10,9);

printf("Issued Date:%d-%d-%d",a.issued.dd,a.issued.mm,a.issued.yy);

gotoxy(10,10);

printf("Returning Date:%d-%d-%d",a.duedate.dd,a.duedate.mm,a.duedate.yy);

}

/\*void loaderanim()

{

int loader;

system("cls");

gotoxy(20,10);

printf("LOADING........");

printf("\n\n");

gotoxy(22,11);

for(loader=1;loader<20;loader++)

{

Sleep(100);printf("%c",219);}

}\*/

//End of program

**CHAPTER FOUR**

**PROGRAM TESTING**

**TEST LOG**

|  |  |
| --- | --- |
| Test Case | Test Objectives |
| Case 1 | To check application for security constraint, password insertion was protected by asterisks |
| Case 2 | To check for the correct entry of the options in the main menu |
| Case 3 | To check that after selecting add books in menu No. 1 will display book categories before book details insertion. |
| Case 4 | To check that after selecting Delete books in menu No. 2 will display where to insert the book ID for deletion to be carried and after result. |
| Case 5 | To check that after selecting Search books in menu No. 3 will display how to search for a book using 1. Search by ID or 2. Search by Name. (Testing option 1 search by ID) |
| Case 6 | To check that after selecting Search books in menu No. 3 will display how to search for a book using 1. Search by ID or 2. Search by Name. (Testing option 2 search by Name) |
| Case 7 | To check that after selecting “issue books” in menu No. 4 will display this options 1. Issue a book 2. View issued book 3. Search issued book 4. Remove issued book (Testing option 1. Issue a book) |
| Case 8 | To check that after selecting “issue books” in menu No. 4 will display this options 1. Issue a book 2. View issued book 3. Search issued book 4. Remove issued book (Testing option 2. View issued book) |
| Case 9 | To check that after selecting “issue books” in menu No. 4 will display this options 1. Issue a book 2. View issued book 3. Search issued book 4. Remove issued book (Testing option 3. Search issued book) |
| Case 10 | To check that after selecting “issue books” in menu No. 4 will display this options 1. Issue a book 2. View issued book 3. Search issued book 4. Remove issued book (Testing option 4. Remove issued book) |
| Case 11 | To check that after selecting “View book list” in menu No. 5 will display all the available books stored on the application will be displayed, with category, ID, Book Name, Author, qty, Renting, Rack no |
| Case 12 | To check that after selecting “Edit book’s record” in menu No. 6 will request for the user to input book ID to be edited. |
| Case 13 | To check that the application will abort itself after usage, after pressing any key it will terminate the application. |

Test Case

|  |  |
| --- | --- |
| **Test case** | 1 |
| **Objectives** | To check application for security constraint, password insertion was protected by asterisks |
| **Test Data** | Starting application requires access code which is : mahmudabds  Invalid password requires you to only press enter to reinsert the password again |
| **Expected Result** | If password match, it will display “Password match Press any key to countinue.....”  But if password mismatch it will display “Warning!! Incorrect Password” |
|  |  |
| **Before**  **Test** |  |
| **After**  **Test** |  |
| **Conclusion** | Test succesfull |

Test Case

|  |  |
| --- | --- |
| **Test case** | 2 |
| **Objectives** | To check for the correct entry of the options in the main menu |
| **Test Data** | Enter number 8 to test |
| **Expected Result** | It will display error message “Wrong Entry!!Please re-entered correct option” |
|  |  |
| **Before**  **Test** |  |
| **After**  **Test** |  |
| **Conclusion** | Test succesfull |

Test Case

|  |  |
| --- | --- |
| **Test case** | 3 |
| **Objectives** | To check that after selecting add books in menu No. 1 will display book categories before book details insertion. |
| **Test Data** | Enter number 1 from the menu to test |
| **Expected Result** | It will display books categories First |
|  |  |
| **Before**  **Test** |  |
| **After**  **Test** |  |
| **Conclusion** | Test succesfull |

Test Case

|  |  |
| --- | --- |
| **Test case** | 4 |
| **Objectives** | To check that after selecting Delete books in menu No. 2 will display where to insert the book ID for deletion to be carried, and after result. |
| **Test Data** | Enter number 2 from the menu to test |
| **Expected Result** | It will display a result instructing a user to input Book ID before deletion |
|  |  |
| **Before**  **Test** |  |
| **After**  **Test** |  |
| **Conclusion** | Test succesfull |

Test Case

|  |  |
| --- | --- |
| **Test case** | 5 |
| **Objectives** | To check that after selecting Search books in menu No. 3 will display how to search for a book using 1. Search by ID or 2. Search by Name. (Testing option 1 search by ID) |
| **Test Data** | Enter number 3 from menu to test |
| **Expected Result** | It will display and prompt the user to input book ID: after the book ID is inserted correctly then it will return all the book information to the user. |
|  |  |
| **Before**  **Test** |  |
| **After**  **Test** |  |
| **Conclusion** | Test succesfull |

Test Case

|  |  |
| --- | --- |
| **Test case** | 6 |
| **Objectives** | To check that after selecting Search books in menu No. 3 will display how to search for a book using 1. Search by ID or 2. Search by Name. (Testing option 1 search by name) |
| **Test Data** | Enter number 3 from menu to test |
| **Expected Result** | It will display and prompt the user to input book name: after the book name is inserted correctly then it will return all the book information to the user. |
|  |  |
| **Before**  **Test** |  |
| **After**  **Test** |  |
| **Conclusion** | Test succesfull |

Test Case

|  |  |
| --- | --- |
| **Test case** | 7 |
| **Objectives** | To check that after selecting “issue books” in menu No. 4 will display this options 1. Issue a book 2. View issued book 3. Search issued book 4. Remove issued book (Testing option 1. Issue a book) |
| **Test Data** | Enter number 4 from menu then 1 to test |
| **Expected Result** | It will display to enter the book ID before you can insert student information |
|  |  |
| **Before**  **Test** |  |
| **After**  **Test** |  |
| **Conclusion** | Test succesfull |

Test Case

|  |  |
| --- | --- |
| **Test case** | 8 |
| **Objectives** | To check that after selecting “issue books” in menu No. 4 will display this options 1. Issue a book 2. View issued book 3. Search issued book 4. Remove issued book (Testing option 2. View issued book) |
| **Test Data** | Enter number 4 from menu then 2 to test |
| **Expected Result** | It will display the number of student that rent books, their names, book category, ID, book name, issued date, Return date |
|  |  |
| **Before**  **Test** |  |
| **After**  **Test** |  |
| **Conclusion** | Test succesfull |

Test Case

|  |  |
| --- | --- |
| **Test case** | 9 |
| **Objectives** | To check that after selecting “issue books” in menu No. 4 will display this options 1. Issue a book 2. View issued book 3. Search issued book 4. Remove issued book (Testing option 3. Search issued book) |
| **Test Data** | Enter number 4 from menu then 3 to test |
| **Expected Result** | It will display “enter the book ID” after inserting a book ID it will show the student that has taken the book, issued date, returning date. |
|  |  |
| **Before**  **Test** |  |
| **After**  **Test** |  |
| **Conclusion** | Test succesfull |

Test Case

|  |  |
| --- | --- |
| **Test case** | 10 |
| **Objectives** | To check that after selecting “issue books” in menu No. 4 will display this options 1. Issue a book 2. View issued book 3. Search issued book 4. Remove issued book (Testing option 4. Remove issued book) |
| **Test Data** | Enter number 4 from menu then 4 to test |
| **Expected Result** | It will display “enter the book ID” after inserting a book ID it will show the student that has taken the book, issued date, returning date then it will prompt “do you want to remove it?(Y/N) if yes, the record of the student that taken the book will be deleted. |
|  |  |
| **Before**  **Test** |  |
| **After**  **Test** |  |
| **Conclusion** | Test succesfull |

Test Case

|  |  |
| --- | --- |
| **Test case** | 11 |
| **Objectives** | To check that after selecting “View book list” in menu No. 5 will display all the available books stored on the application will be displayed, with category, ID, Book Name, Author, qty, Renting, Rack no |
| **Test Data** | Enter number 5 from menu to test |
| **Expected Result** | It will display all the available books in the library by Categories, ID, Book Name, Author, QTY, Renting, RackNo |
|  |  |
| **Before**  **Test** |  |
| **After**  **Test** |  |
| **Conclusion** | Test succesfull |

Test Case

|  |  |
| --- | --- |
| **Test case** | 12 |
| **Objectives** | To check that after selecting “Edit book’s record” in menu No. 6 will request for the user to input book ID to be edited. |
| **Test Data** | Enter number 6 from menu to test |
| **Expected Result** | It will display “Enter Book ID to be edited:” after the book ID has been successfully inserted then it will display “enter new book name:” “Enter new author:” Enter new Quantity:” “Enter new price:” “Enter new rackNo:” |
|  |  |
| **Before**  **Test** |  |
| **After**  **Test** |  |
| **Conclusion** | Test succesfull |

Test Case

|  |  |
| --- | --- |
| **Test case** | 13 |
| **Objectives** | To check that the application will abort itself after usage, after pressing any key it will terminate the application. |
| **Test Data** | Enter number 7 from menu to test |
| **Expected Result** | It will display “Are you sure you want to exit?..................>press any key to continue…. |
|  |  |
| **Before**  **Test** |  |
| **After**  **Test** |  |
| **Conclusion** | Test succesfull |

**CHAPTER FIVE**

**IMPLEMENTATIONS**

**Hardware and Software Configuration**

**Hardware Configuration:**

For the Program to be run the set of computer hardware must be prepared with the following specifications:

* Pentium II higher
* 733 MGHZ or higher
* 32MB RAM or higher
* 10 GB HDD 0r higher Mouse
* QWERTY keyboard

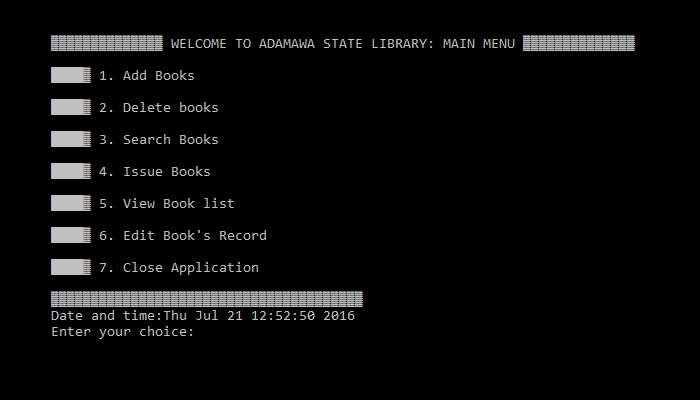
**Software Requirement:**

For the Program to be run the set of the following software must be prepared with the following specifications:

* Borland C++/Turbo C++
* Disk Operating System (DOS) Environment
* The Main Program Code Blocks
* MS Word or Other Text Editor

**USER MANUAL**

Figure 1 Main Menu

1. This is the view of the main menu of the application that can be started by double clicking the application icon.

There are 1 to 7 options in the menu, if a user want to add a new book record then he/she have to select 1 from the menu which will take the user to a book categories, the user will select a category among the list and proceed with the insertion.

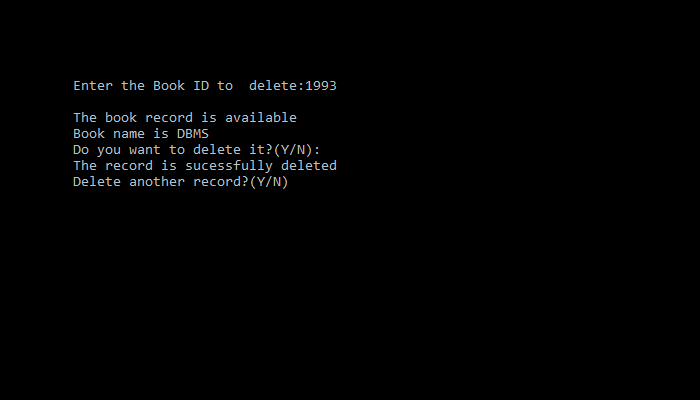
1. This is the view of option 2 which is **“Delete books”** in other any user to delete any book from the book list, book ID is extremely important without any unique book ID no book will be deleted. In case a user don’t know the actual book ID to delete, it is very simple all the user need to do is to go for option 4 from the main menu to “view book list” and copy the book ID he/she want to delete.

Figure 2 delete books example after pressing 2 from the main menu

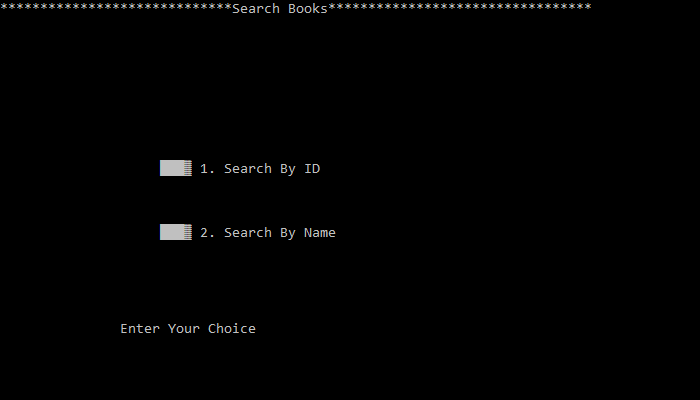
1. This is the view of option 3 which is **“Search books”** in other any user to search for books record two more options is available, option 1 is to search for the book using book ID or option 2 to search for the book using book Name. No book record will be displayed without proper unique stored record of the books. If user cannot still find book then it’s very simple to go back to Main menu and View book list.

Figure 3 Searching for a book records by ID or Name display op3 in the menu

Figure 4 issue book display after pressing option 4 from the main menu

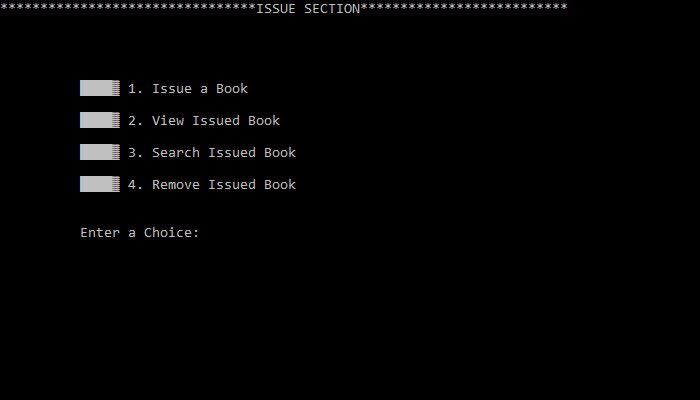
1. This is the view of option 4 which is **“Issue books”** in other to issue a book to a student, book ID is highly important. In this section there are 4 options available, 1. Issue a book allows you to insert the information of student that want to rent a particular book. 2. View issued book allows you to see all the student that books has been issued to them. 3. Search issued books allows you to search for a particular book that student have been issued previously. 4. Remove issued book allows you to remove any student who’s issued books has been returned.

Figure 5 view book list display after pressing 5 from the main menu

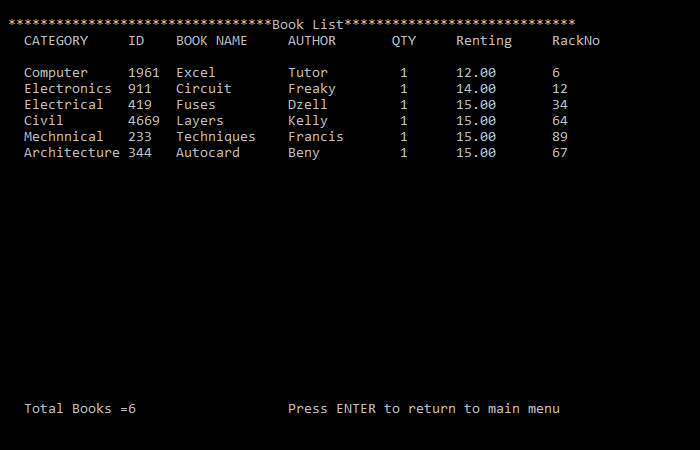
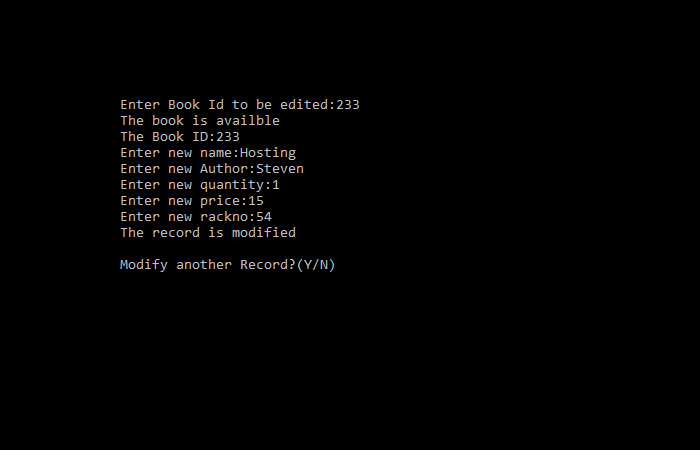
1. This is the view of option 5 which is **“view book list”** in other to view all the available books that has been stored in the application by book category, by book ID, by book name, book author, book qty, book renting price, and book rack no. is simply by going for option 5 from the main menu everything will be displayed for any user that need to see the overall summary of the books that has been inputted to the program.
2. This is the view of option 6 which is **“edit book’s record”** in other any user to update any book information, book ID is important before making any modifications to a book record, when book ID is valid it will allow a user to update the book name, update the book author, update the book qty, update the renting price, and rack number.

Figure 6 editing book’s record example after entering a valid book ID

1. Application **termination**, which is no 7 on the main menu allows a user to quit the program after usage. It’s recommended to press any key to quit. This is what the application will pops up after pressing seven from the main menu “**Are you sure you want to exit?...........>Press any key to continue . . .”**

**CHAPTER SIX**

**CONCLUSION**

Definitively this project has been finished inside the stipulated time and the accompanying topics have illustrated:

**Program weakness**

* The program is not protected with user name only password protection only one person can operate the application
* The application can only show the numbers of student that books has been issued to them but cannot display the renting price, the application user have to check for all the books record to know the renting price which is very inconvenience.
* The application have only 1 to 5 book categories which is very frustrating if a user want to insert new book categories.

**Program Strength**

* The application is password protected, upon insertion of the password asterisks are shown in case a 3rd party might be looking at the screen
* All the books in the library will be easily managed in the application
* Adding new books will be very easy which will allow the user to input unlimited numbers of book per day.
* By having computerize system, it represent data in good format
* It can easily be recover/backup
* It can easily remove record of a book using book ID hassle free
* If there are huge numbers of Book records, it is more quicker and simple to search a record in just a few seconds

**Program Enhancement**

* More book categories need to be added when the library require new book categories
* Student who book has been issued to him/her need to have renting price and penalty fee
* The application have password upon start-up but it need have username as well for more security constrains
* Only 1 administrator can have the access code but in future it need to have multiple administrator in case of staffs time shifting duties

Reference

* Fabio, Adam (6 April 2014). "CPLD Tutorial: Learn programmable logic the easy way". Hackaday. Retrieved 19 May 2015.
* A Tutorial for the Go Programming Language. Archived July 26, 2010, at the Wayback Machine. The Go Programming Language. Retrieved July 26, 2011.
* Kaner, Cem; Falk, Jack; Nguyen, Hung Quoc (1999). Testing Computer Software, 2nd Ed. New York, et al: John Wiley and Sons, Inc. p. 480. ISBN 0-471-35846-0.
* Leitner, A., Ciupa, I., Oriol, M., Meyer, B., Fiva, A., "Contract Driven Development = Test Driven Development – Writing Test Cases", Proceedings of ESEC/FSE'07: European Software Engineering Conference and the ACM SIGSOFT Symposium on the Foundations of Software Engineering 2007, (Dubrovnik, Croatia), September 2007. PDF available online at <http://se.inf.ethz.ch/people/leitner/publications/cdd_leitner_esec_fse_2007.pdf>
* Adamson, Veronica, et al. (2008). JISC & SCONUL Library Management Systems Study PDF (1 MB). Sheffield, UK: Sero Consulting. p. 51. Retrieved on 21 January 2009. "... a Library Management System (LMS or ILS 'Integrated Library System' in US parlance)." Some useful library automation software are: KOHA ,Grennstone .LIBsis, and granthlaya. PDF available online at <http://www.jisc.ac.uk/media/documents/programmes/resourcediscovery/lmsstudy.pdf>