

PURBANCHAL UNIVERSITY

Biratnagar Nepal



A Project report on

“LIBRARY BOOKS MANAGEMENT SYSTEM”

In the partial fulfillment for the requirement of the 1st Semester Project-I (subject code- BIT 106CO) in the completion of **Bachelor of Information Technology (BIT)** degree at **KIST college of Information Technology**, under **Purbanchal University**.

Submitted By: -

- 1) Anjan Shrestha**
- 2) Pukar Tiwari**
- 3) Sandip Shrestha**
- 4) Saurav Magar**

Submitted To: -

Purbanchal University

Under The Guidance of
Mr. Prawesh Dhungana
Lecturer, BIT

KIST COLLEGE OF INFORMATION AND TECHNOLOGY
KAMALPOKHARI, KATHMANDU, NEPAL

KIST COLLEGE OF INFORMATION AND TECHNOLOGY

KAMALPOKHARI, KATHMANDU NEPAL



CERTIFICATE

This is to certify that the project work entitled “**LIBRARY BOOKS MANAGEMENT SYTEM**” is carried out by **ANJAN SHRESTHA (5418), PUKAR TIWARI (5429), SANDIP SHRESTHA (5437), SAURAV MAGAR (5432)**, bona fide students of **KIST COLLEGE OF INFORMATION AND TECHNOLOGY** in partial fulfillment for the award of **BACHELOR IN INFORMATION AND TECHNOLOGY** of the **PURBANCHAL UNIVERSITY, BIRATNAGAR NEPAL**, during the year **2020-2021**. It is certified that all corrections indicated for internal assessment have been incorporated in the report submitted in the department library. The project report has been approved, as it satisfied the academic requirements in respect of the project work prescribed for the said degree.

The details of the students are as follows: -

NAME	REGISTRATION NO.	SYMBOL NO.
Anjan Shrestha	058-3-2-04716-2020	313418
Pukar Tiwari	058-3-2-04731-2020	313432
Sandip Shrestha	058-3-2-04739-2020	313440
Saurav Magar	058-3-2-04742-2020	313443

Course Semester: - 1st Semester

Subject: - Project-I

Subject Code: - BIT (160CO)

Mr. Deepak Khadka
Program Coordinator, BIT



**KIST COLLEGE OF INFORMATION AND
TECHNOLOGY KAMALPOKHARI, KATHMANDU**

Examiner's Certification

The Project Report
On
“LIBRARY BOOKS MANAGEMENT SYSTEM”

Developed by

**Anjan Shrestha
Pukar Tiwari
Sandip Shrestha
Saurav Magar**

Is approved and is acceptable in qualify form.

Internal Examiner

Name:

Designation:

External Examiner

Name:

Designation:

ACKNOWLEDGEMENT

It is with greatest satisfaction and euphoria that we are submitting our project report entitled “**LIBRARY BOOKS MANAGEMENT SYSTEM**”. We have completed it as a part of the curriculum of **PURBANCHAL UNIVERSITY**.

We also take this opportunity to express a deep sense of gratefulness to our **BIT Coordinator Mr. Deepak Khadka** and **BIT Lecturer Mr. Prawesh Dhungana** for their amiable support, valuable information and guidance which helped us in completing this task throughout its various stages. We are indebted to all members of **KIST College**, for the valuable support and suggestion provided by them using their specific fields’ knowledge. We are grateful for their cooperation during the period of our project.

Finally, we would also like to express our gratefulness towards **Purbanchal University** for designing such a wonderful course structure. It will help us to get more knowledge in the field of Information Technology & help us to have a bright future in the field of technology.

We hope our university will accept this attempt as a successful project.

Last but not the least, our sincere thanks to our parents, teaching and non-teaching staffs of our college and also my friends.

ANJAN SHRESTHA (313418)

PUKAR TIWARI (313432)

SANDIP SHRESTHA (313440)

SAURAV MAGAR (313443)

STUDENT'S DECLARATION

We hereby declare that the project report entitled “**LIBRARY BOOKS MANAGEMENT SYTEM**” is a result of our own work. If we are found guilty of copying any other report or published information and showing as our original work, we understand that we shall be liable and punishable by **Purbanchal University**.

We further certify that this Project submitted in partial fulfillment of the requirement for the award of Bachelor in Information Technology (**BIT**) of the **Purbanchal University** is our original work and has not been submitted for award of any other degree or other similar title or prize.

S.N.	Name	Registration No.	Symbol No.
1	Anjan Shrestha	058-3-2-04716-2020	313418
2	Pukar Tiwari	058-3-2-04731-2020	313432
3	Sandip Shrestha	058-3-2-04739-2020	313440
4	Saurav Magar	058-3-2-04742-2020	313443

TO WHOM IT MAY CONCERN

This is to certify that **Mr. Anjan Shrestha, Mr. Pukar Tiwari, Mr. Sandip Shrestha** and **Mr. Saurav Magar** of **Bachelor in Information Technology (BIT)** has studied as per the curriculum of **BIT 1nd Semester** and completed the project entitled “**Library Books Management System (LBMS)**”. This project is the original work of **Mr. Anjan Shrestha, Mr. Pukar Tiwari, Mr. Sandip Shrestha and Mr. Saurav Magar** and was carried out under the supervision of **Mr. Prawesh Dhungana** as per the guidelines provided by **Purbanchal University** and certified as per the student’s declaration that project “**Library Books Management System**” has not been presented anywhere as a part of any other academic work.

The detail of the student is as follows:

Name of Students	: Anjan Shrestha Pukar Tiwari Sandip Shrestha Saurav Magar
Semester	: 1 nd
Subject Code	: BIT 106C0
Project Title	: Library Books Management System

.....
Mr. Deepak Khadka

Program Coordinator, BIT

KIST College of Information Technology

ABSTRACT

The purpose of “**Library Books Management System**” is to automate the existing manual system by the help of computerized equipment’s and full-fledged computer software, fulfilling their requirements, so that their valuable data/information can be stored for a longer period with easy accessing and manipulation of the same. The required software and hardware are easily available and easy to work with.

Library Books Management System, as declared above, can lead to error free, secure, reliable and fast management system. It can assist the user to concentrate on their other activities rather to concentrate on the record keeping. Thus, it will help organization in better utilization of resources. The organization can maintain computerized records without redundant entries. That means that one need not be distracted by information that is not relevant, while being able to reach the information.

The aim is to automate its existing manual system by the help of computerized equipment and full-fledged computer software, fulfilling their requirements, so that their valuable data/information can be stored for a longer period with easy accessing and manipulation of the same. Basically, the project describes how to manage for good performance and better services for the clients.

According’s: LBMS refers to library books management system.

TABLE OF CONTENTS

Chapter 1.....	1
1.1 INTRODUCTION.....	1
1.2 PROBLEM STATEMENT	2
1.3 OBJECTIVES	2
1.4 SCOPE	2
1.5 ADVANTAGES.....	3
Chapter 2.....	4
SYSTEM DESIGN	4
2.1 ALGORITHM.....	4
2.2 FLOWCHART	6
Chapter 3.....	12
3.1 REQUIREMENTS ANALYSIS & IMPLEMENTATION SYSTEM	12
3.2 SYSTEM METHODOLOGY	13
3.2.1 Requirement and Gathering Analysis.....	14
3.2.2 System Design	15
3.2.3 Implementation	16
3.2.4 Integration and testing.....	16
3.2.5 DEPELOYMNET AND MAINTANANCE	17
Chapter 4.....	18
4.1 CONCLUSION & FUTURE SCOPE	18
4.2 Appendices	19
4.2.1 Screenshots.....	20
4.2.2 Source code.....	23

Chapter 1

INTRODUCTION

The LBMS has been developed to override the problems prevailing in the practicing manual system. This software is supported to eliminate and, in some case, reduce the hardships faced by this existing system. Moreover, this system is designed for the particular need of the library departments to carry out operations in a smooth and effective manner.

Library Books Management System is a desktop-based application. This application is developed by using C programming language. It is an easy and time-efficient way of storing the data. These data can be easily accessed by the administration.

Library Books Management System (LBMS) is a system which helps users to insert, name, edit, rename, issue and time & date. By using this system, it will help in control the loss of book. The library staffs can get the right information in time and make further Planning.

This system is developed by using “C Programming” as its backend and This application based on file handling in C, where I have used a file-related function like fopen, fread, fwrite, etc good thing is that “Library Books Management System project” is password-protected, so only authorized person able to login in this application.

The project demonstrates the creation of a user interface of a system, without the use of C Graphics library. The project uses basic C function to generate menus, show message boxes and print text on the screen.

PROBLEM STATEMENT

A Library books management system has to be developed for automating the Library Management System. Our software provides following facilities: -

- More manual work and effort spent on their respective working areas.
- No backup data record of books.
- Time consuming.

OBJECTIVES

The main objectives behind the development of this project are as follows:

- To utilize the information of Books.
- To store and retrieve books items.
- To manage the particular records of student.
- To provide the details of issue books.

Thus, there are a number of objectives behind developing the “LIBRARY BOOKS MANAGEMENT SYSTEM” and it reduces a lot of manual working of the department. [1]

SCOPE

The scope of this project is as follows:

- To assist the staff in capturing the effort spent on their respective working areas.
- To utilize resources in an efficient manner by increasing their productivity through automation.
- The system generates types of information that can be used for various purposes.

Thus, there are information scopes behind developing the “LIBRARY BOOKS MANAGEMENT SYSTEM” and it reduces a lot of burden of the entry.

ADVANTAGES

The library management system is designed to contribute well management of library functions. It offers ease to perform day to day library operations electronically. This practice being many advantages like:

- Simple and easy to operate
- Saves time and reduces overheads
- Remove manual processes to issue books and maintain records
- Increase librarian's efficiencies
- Increase accuracy

Chapter 2

SYSTEM DESIGN

ALGORITHM

Step 1: Start

Step 2: Welcome Message. Press any key to continue

Step 3: Login! Enter login detail.

Step 4: If matches, goto step 6. Else goto step 5.

Step 5: If step 3 is repeated for many times,

 Step 5.1: Display “Login Failed”

 Step 5.2: Close the program.

Step 6: Display “Login is Successful”

Step 7: Show Main Menu

Step 8: Enter choice form menu

Step 9: If the choice is between 1 to 6, goto step 10. Else Display Invalid Choice.

Step 10: If choice is 1 “Insert Record”.

 Step 10.1: Show Enter the Book ID

 Step 10.2: Check the Book ID in Database If match found goto Step 10.3 Else goto Step

10.4

 Step 10.3 Display Enter Unique Book ID. Goto step 10.1

 Step 10.4: Enter the Detail of Book and Student

 Step 10.5: Display “Do you want to insert record of another book” if yes goto step 10.1.

Else goto step 7.

Step 11: If choice is 2 “Display Record”.

 Step 11.1: Display the List of Books.

 Step 11.2: goto step 7.

Step 12: If choice is 3 “Update Record”.

 Step 12.1: Enter the Book ID which you want to update. If Book ID matched with database record goto step 12.2. else display “Record not found” goto step 12.

 Step 12.2: Display the existing data and Display Enter the New data.

 Step 12.3: Display “The record has been successfully updated in the data file”

 Step 12.4: goto step 7.

Step 13: If choice is 4 “Delete Record”.

 Step 13.1: Display “Enter the Book ID which you want to delete”.

 Step 13.2: If record found goto step 13.3. Else display “No record found” goto step

13.1.

 Step 13.2: Display “The record has been successfully deleted form the data file”

 Step 13.3: goto step 7.

Step 14: If choice is 5 “Search Record”

 Step 14.1: Display Search Menu.

Step 14.2: Enter a number between 1 to 4.
 Step 14.3: If choice is 1
 Step 14.3.1: Enter the Book ID.
 Step 14.3.2: If record found Display the record. Else Display no record found
 goto step 14.1
 Step 14.3.3: goto step 7.
 Step 14.4: If choice is 2.
 Step 14.4.1: Enter the Student Name
 Step 14.4.2: If record found Display the desire record. Else Display no record
 found got to step 14.1
 Step
 Step 14.5: If choice is 3.
 Step 14.5.1 Enter the Book Name
 Step 14.6: If choice is 4.
 Step 14.6.1: goto step 7.
 Step 15: If choice is 6 “Quit”;
 Step 15.1: Display “Thank you!”
 Step 15.2: Close the program. Else Display “Invalid choice”
 Step 15.3: goto step 7.

 Step 16: Stop

FLOWCHART

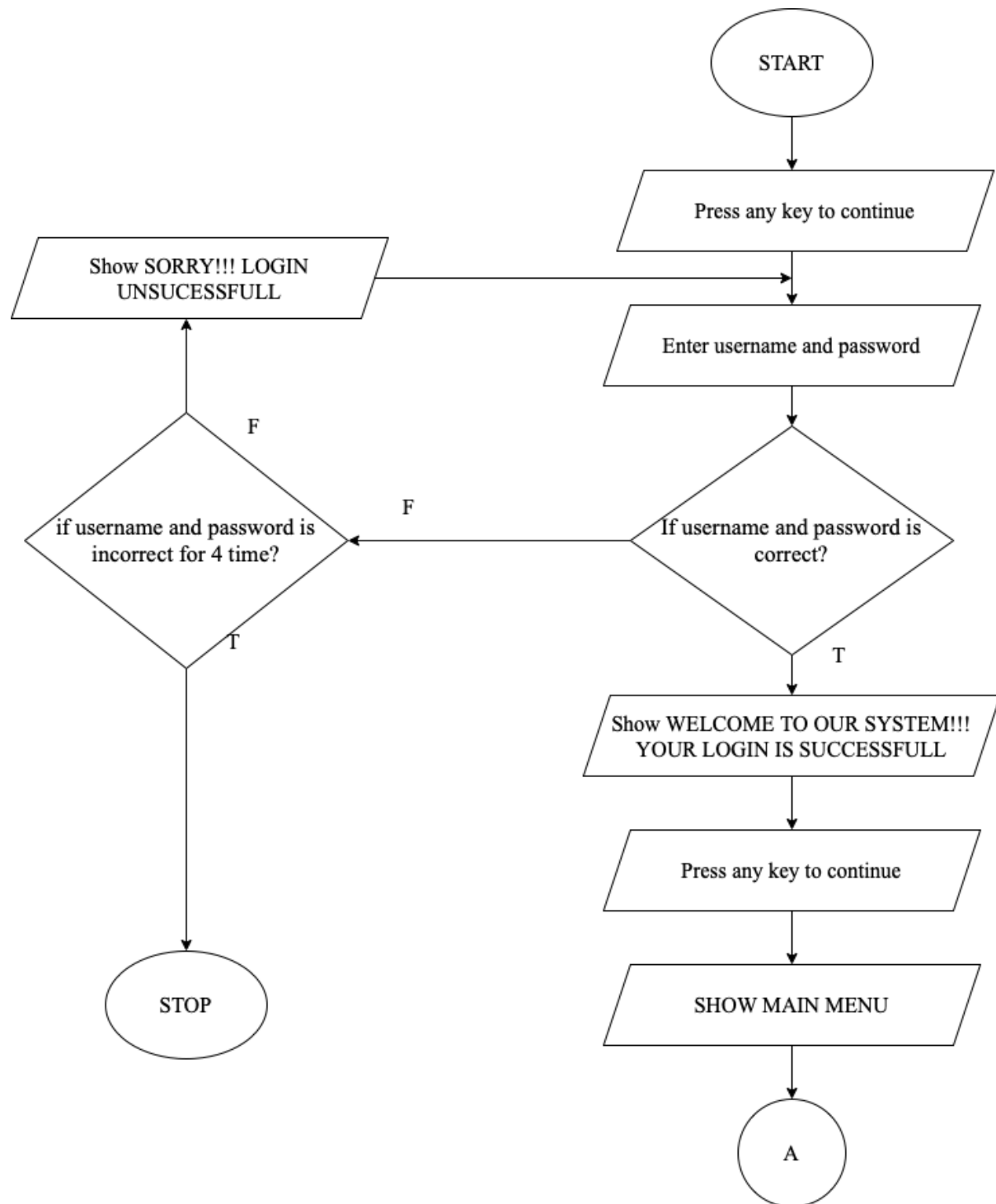


Figure 1 Login Flowchart

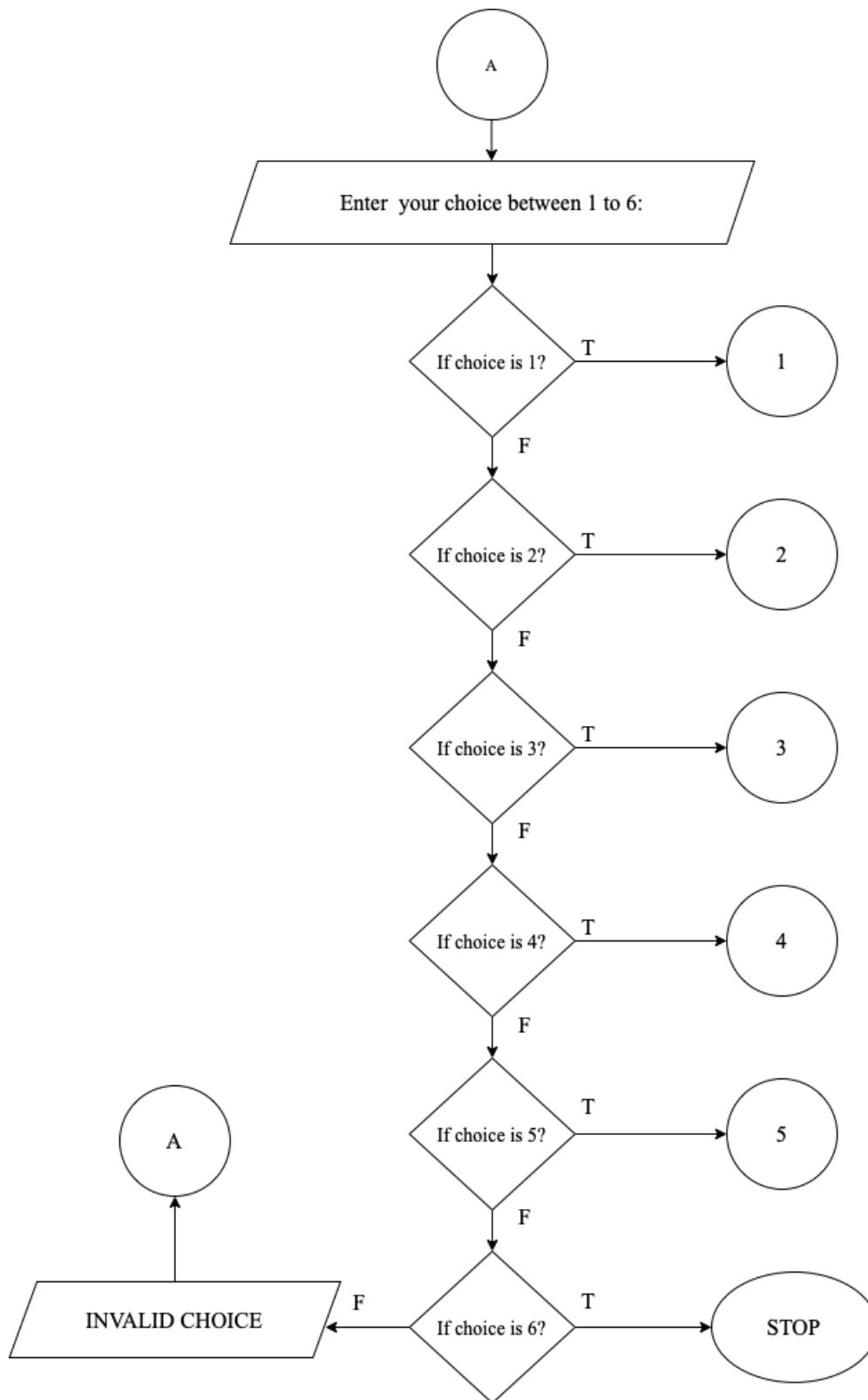


Figure 2 Main menu Flowchart

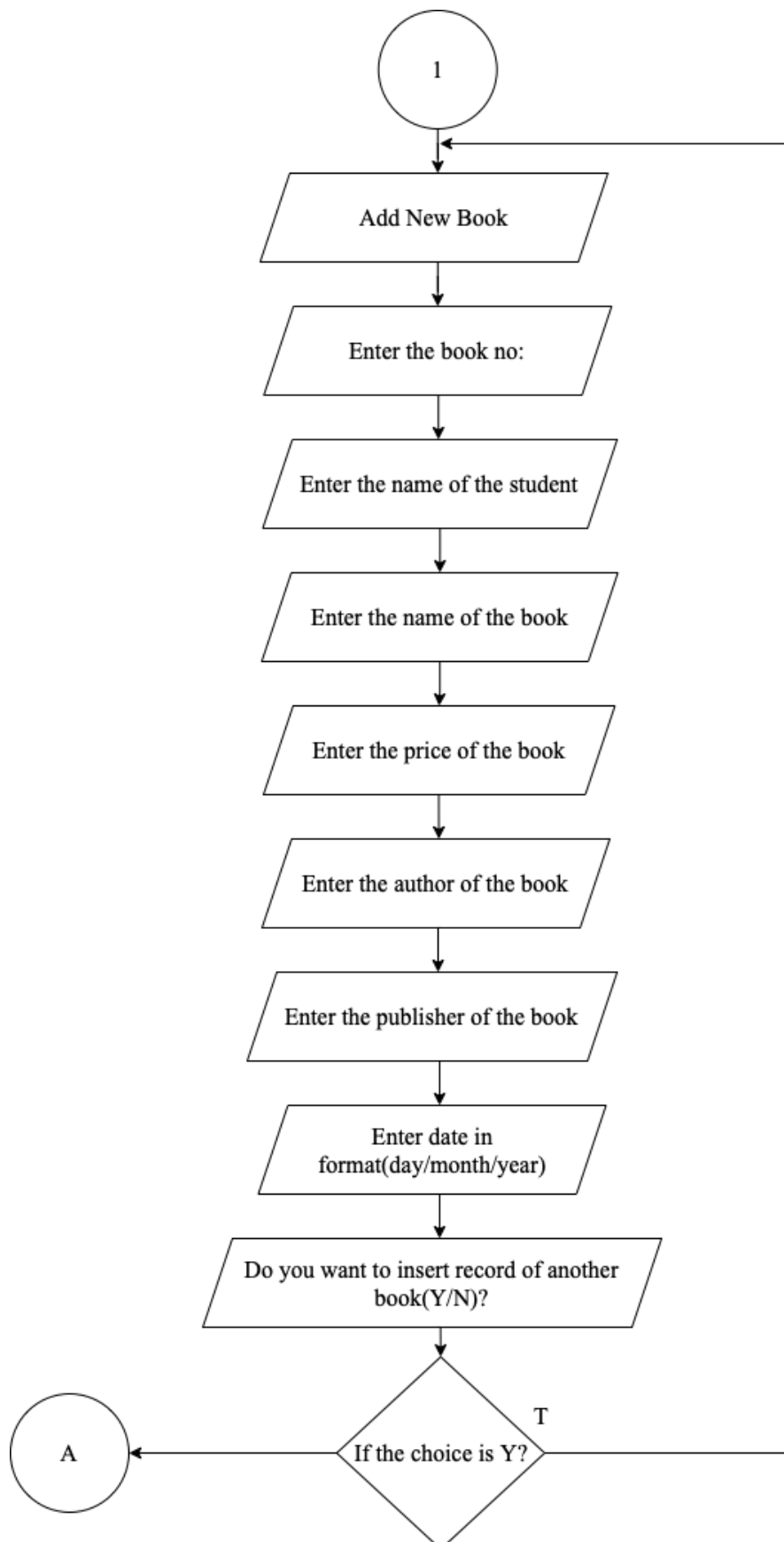


Figure 3 InsertData Flowchart

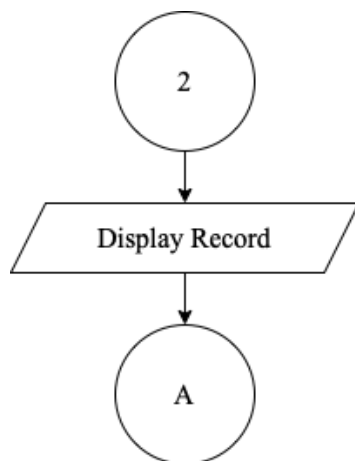


Figure 4 Display Flowchart

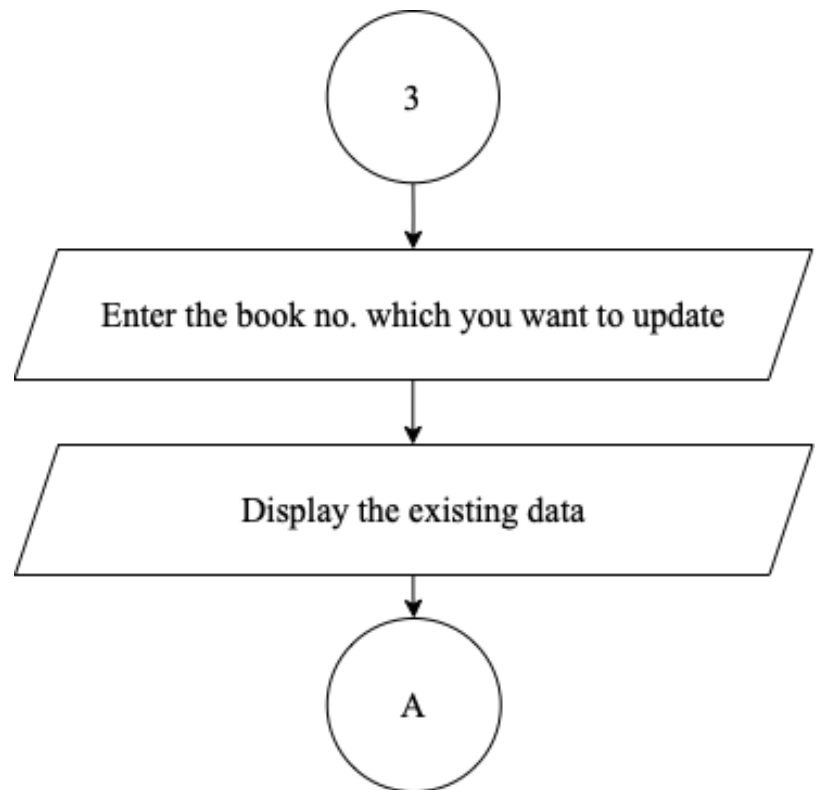


Figure 5 Update Data Flowchart

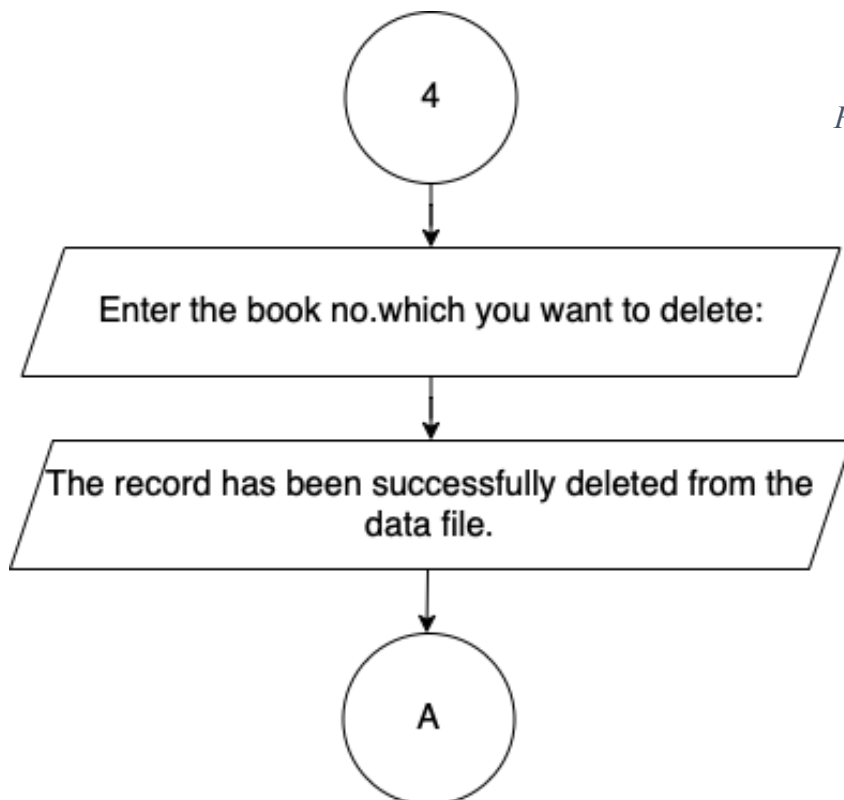


Figure 6 Delete Data Flowchart

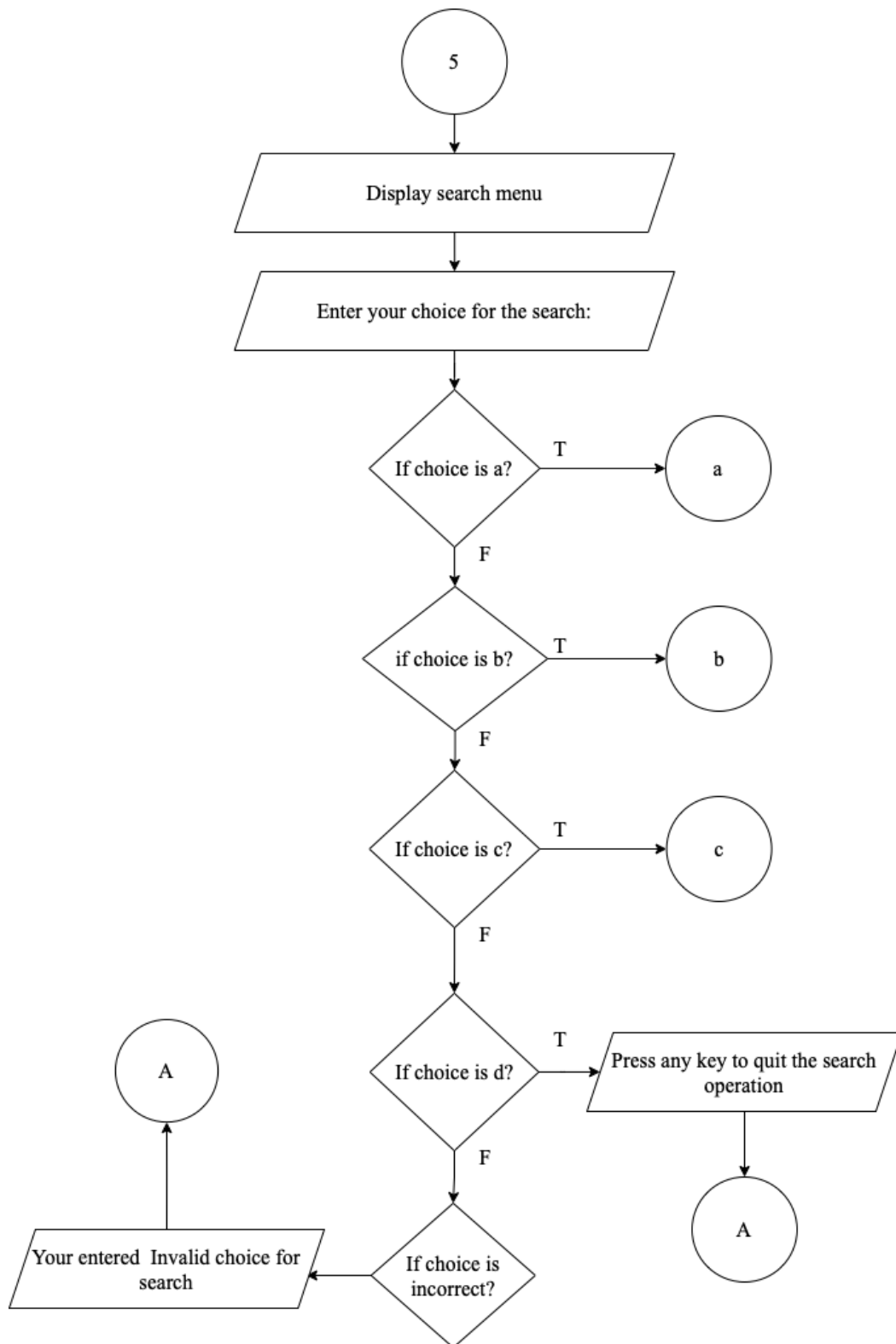


Figure 7 Search Menu Flowchart

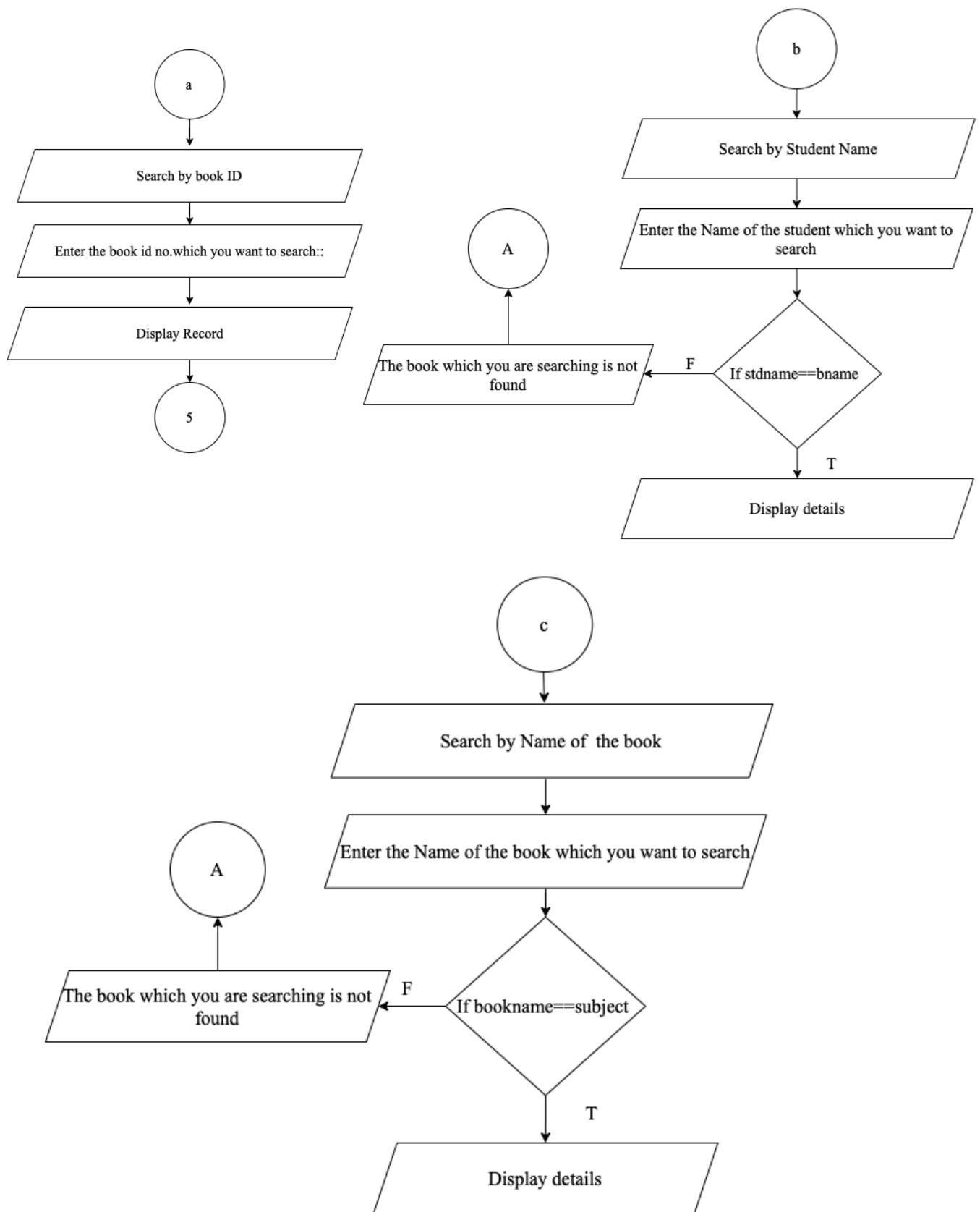


Figure 8 Search Menu choice Flowchart

Chapter 3

REQUIREMENTS ANALYSIS & SYSTEM IMPLEMENTATION

Following hardware and software requirement should be met for flawless running of this system:

Hardware: Hardware is the collection of physical parts of a computer system. This includes the computer case, monitor, keyboard, and mouse. It also includes all the parts inside the computer case, such as the hard disk drive, motherboard, video card, and many others. Computer hardware is what you can physically touch.

MINIMUM REQUIREMENTS:

PROCESSOR: PENTIUM-II

SPEED: 1.5Hz

RAM: 32MB

HARDDISK: 20MB (At least 80MB of free space)

MONITOR: VGA COLOR MONITOR

Software: Software is a set of instructions, data or programs used to operate computers and execute specific tasks. It is the opposite of hardware, which describes the physical aspects of a computer.

OPERATING SYSTEM: WINDOWS XP, 2000 Professional

COMPILER: DEV C++

SYSTEM METHODOLOGY

WATERFALL MODEL

The waterfall model is a classical model used in system development life cycle to create a system with linear and sequential approach. It is termed as waterfall because the model develops systematically from one phase to another in a downward fashion. This model is divided into different phase and the output of one phase is used as the input of the next phase starts and there is no overlapping of the phase. [2]

The sequential phases described in the Waterfall model are:

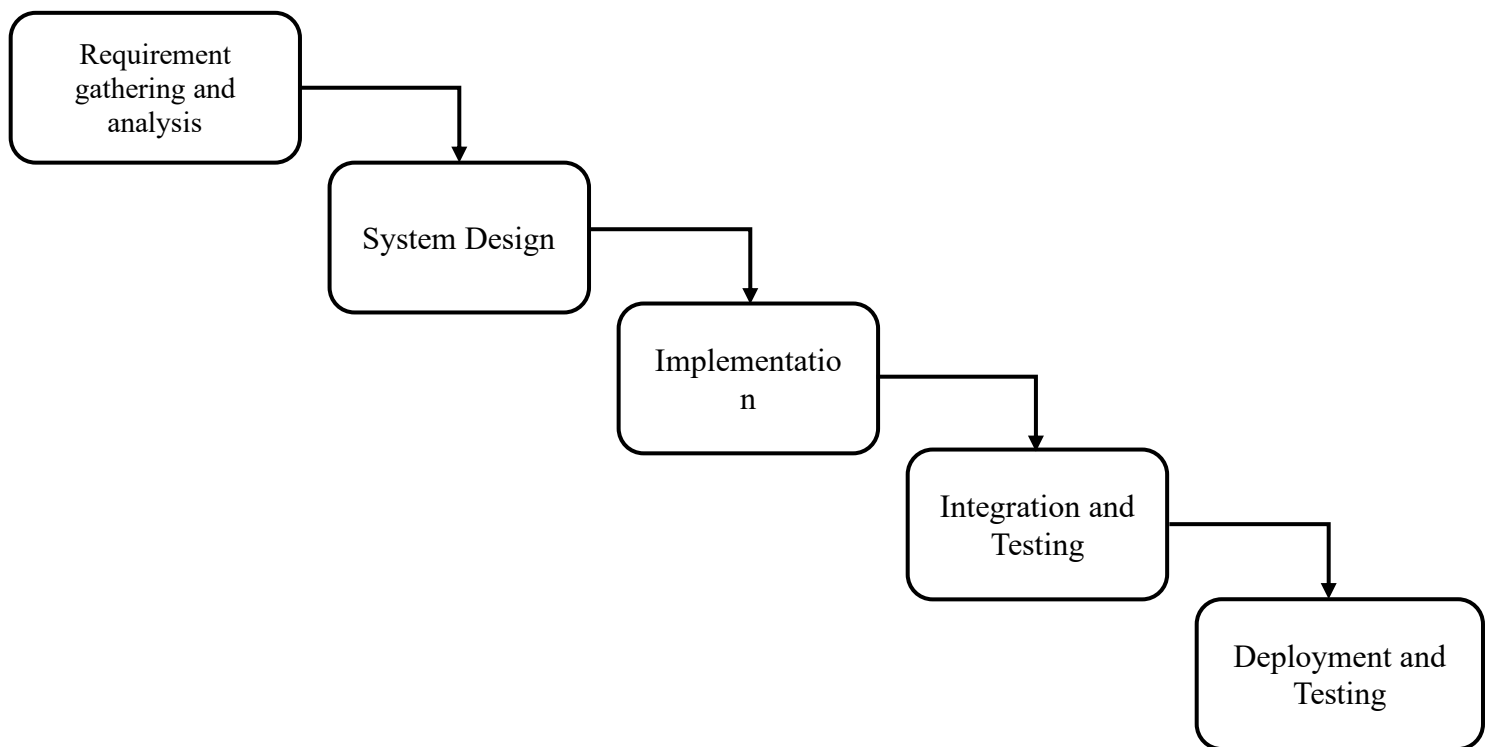


Figure 9 Waterfall Model

3.1.1 Requirement and Gathering Analysis

FUNCTIONAL REQUIREMENT

In software and system engineering, a functional requirement defines a function of a system or its component, where a function is described as a specification of behavior between input and outputs

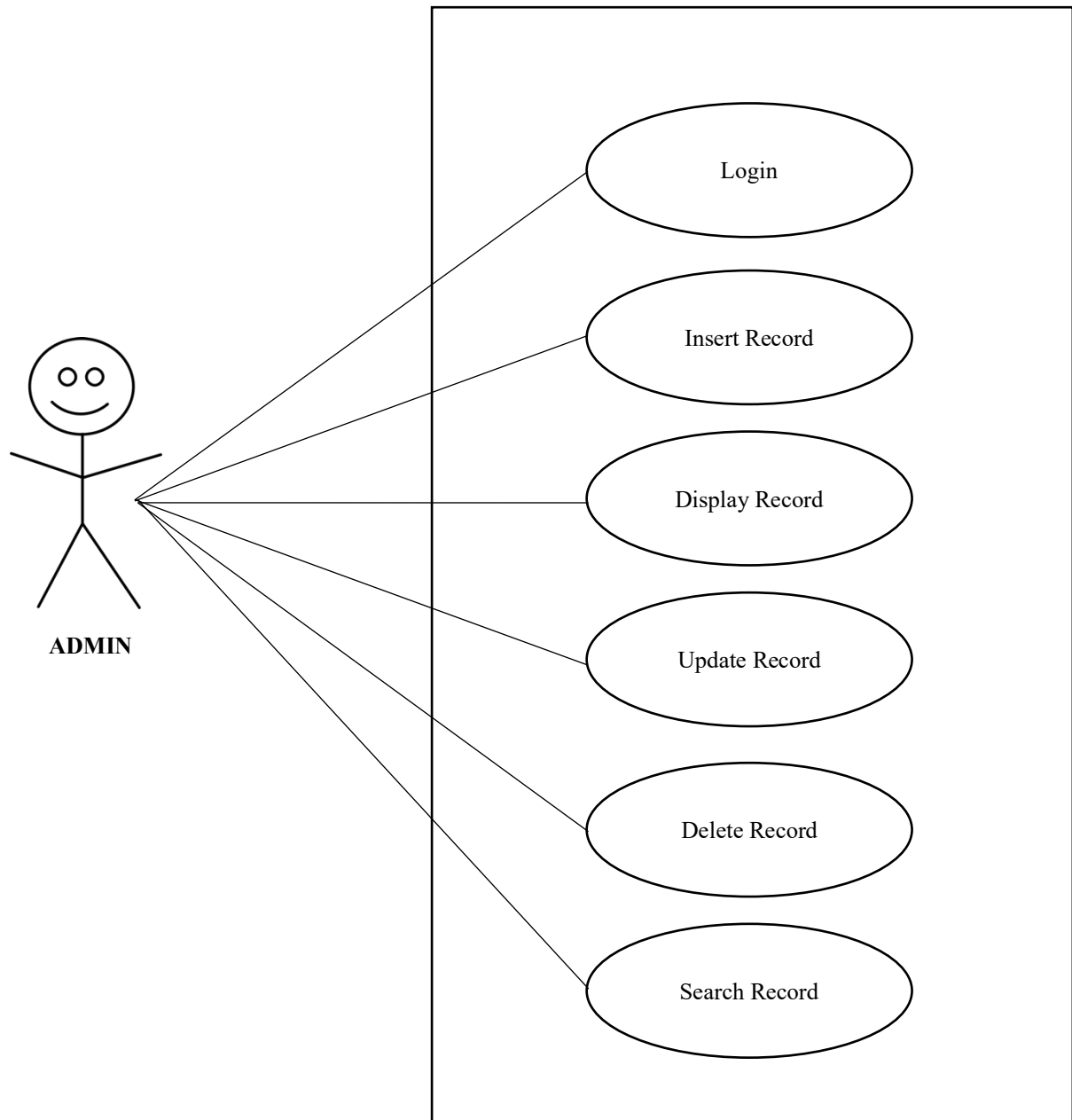


Figure 10 Use Case Diagram

3.1.2 System Design

In this phase we designed the algorithm and flowchart required for the development of the system.

System design is the process of designing the architecture, components, and interfaces for a system so that it meets the end-user requirements. A good system design is to organize the program modules in such a way that are easy to develop and change. There are many strategies or techniques for performing system design.

- **Importance:**
 - If any pre-existing code need to be understood, organized, and pieced together.
 - It is common for the project team to have to write some code and produce original programs that support the application logic of the system.

There are many strategies or techniques for performing system design.

- **Top-down approach:** Top-down integration testing is an integration testing technique used in order to simulate the behavior of the lower-level modules that are not yet integrated. Each system is divided into several subsystems and components. Each of the subsystems is further divided into a set of subsystems and components.
- **Advantages of top-down approach:**
 - The main advantage of the top-down approach is that its strong focus on requirements helps to make a design responsive according to its requirements.

3.1.2.1 Functional Analysis

No.	Function module	Function Description
1.	login ()	This function is for security purpose so that person other than admin cannot manipulate the system or program.
2.	insertData ()	This function opens the binary file in append mode and writes the book and the details.
3.	searchData ()	This function opens the binary file in reading mode and asks the user to enter the book name which wants to search.
4.	displayData ()	It opens the file in reading mode and read and display all the stored book details.
5.	deleteData ()	This function asks the book id form the user for the book want to delete. In this function, I am creating a temporary binary file and copy all the data from the existing file except the book whose book id entered by the user. In the last renamed the temporary bin file with a existing binary file.
6.	updateData ()	This function opens the file in rb+ mode (reading and writing). It asks the user for the new data
7.	checkBookNo ()	This function compares the old book id to new book id. if the book id match, then display the book id which you typed just now, is already exists.
8.	findlastBookNo ()	This function displays the last duplicate book id. and asks to enter a unique book id no.

3.1.3 Implementation

It is the process of using the project in client's computer. After the executive file has been created, this project can be copied from saved source to any secondary storage device and pasted to the required system. The project can be operated by opening it, completely replacing the existing manual system.

3.1.4 Integration and testing

Testing in a project development is a very important task to find out the possible mistakes made by the developers. The system cannot give the correct output until the project contains no errors at all. This project has checked the possible errors by using the following approaches:

- Black Box Testing Approach:** This approach concentrates on the basic requirements of the project. It simply checks direct matching of records of particular book, after we select a book no of a particular student.
- White Box Testing Approach:** This approach concentrates on the actual codes written during the development of the project. It checks every line of codes in all the functions of the program.

This project has fully tested by using both approach's and ensures the correct output

3.1.5 DEPELOYMNET AND MAINTANANCE

When time changes, the requirements of the organization also changes and this project can no longer fulfill its requirements. The changes are necessary to keep the project running and useful to college. Maintenance may be required when the college changes its requirements.

Chapter 4

CONCLUSION & FUTURE SCOPE

CONCLUSION:

Our project is only a humble venture to satisfy the needs to manage the project work. Several user-friendly coding has also adopted. This package shall prove to be a powerful package in satisfying all the requirements of the school and college. The objective of software planning is to provide a frame work that enables the librarian to keep the record of the book within a limited time.

Our project provides a computerized version of library management system which will be beneficial for the students as well as the staff of the library. It makes entire process easy where student can search books, staff can generate reports and do book transactions. It also has a facility for login where library staff can login and can see status of books issued.

FUTURE SCOPE:

There is a future scope of the project with addition to the facilities as follows:-

1. We can provide online facilities for ordering and searching the availability of books.
2. We can add login page for individual students.
3. User interfaced can be made attractive.

References

- [1] FREEPROJECT, "freeprojectz," 20 2 2015. [Online]. Available: freeprojectz.com/projects/.
- [2] R. S. PRESSMAN, SOFTWARE ENGINEERING, McGraw-Hill Companies, 1997.

Appendices

4.1.1 Screenshots



Figure 11 Welcome Screen



Figure 12 Login Menu

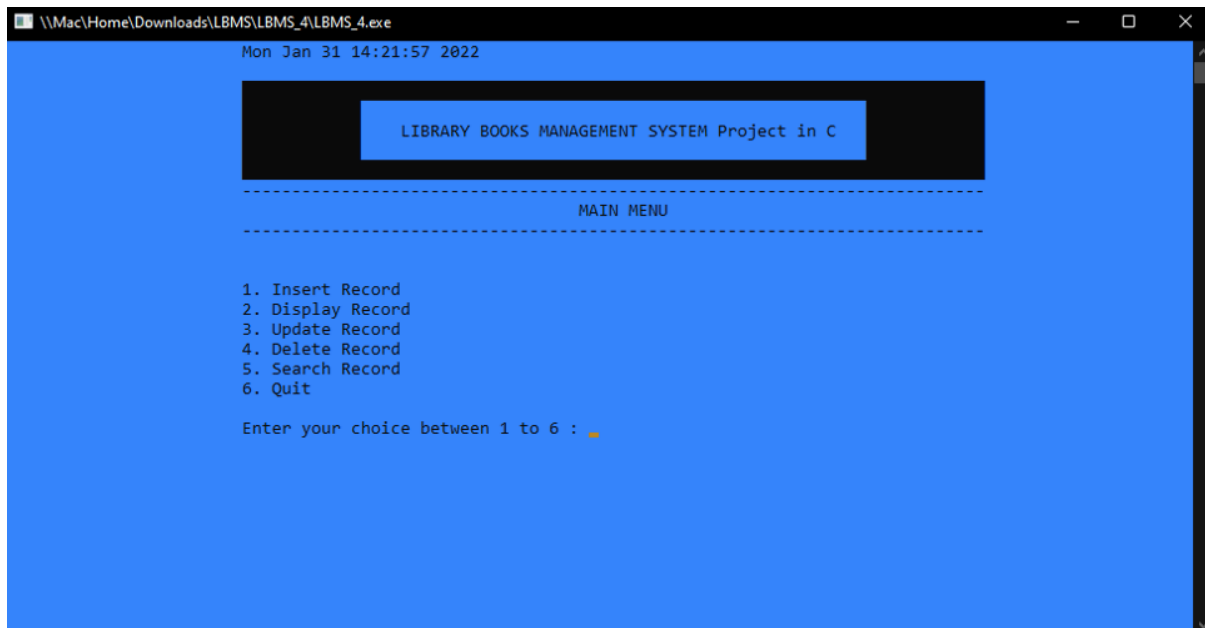


Figure 13 Main Menu

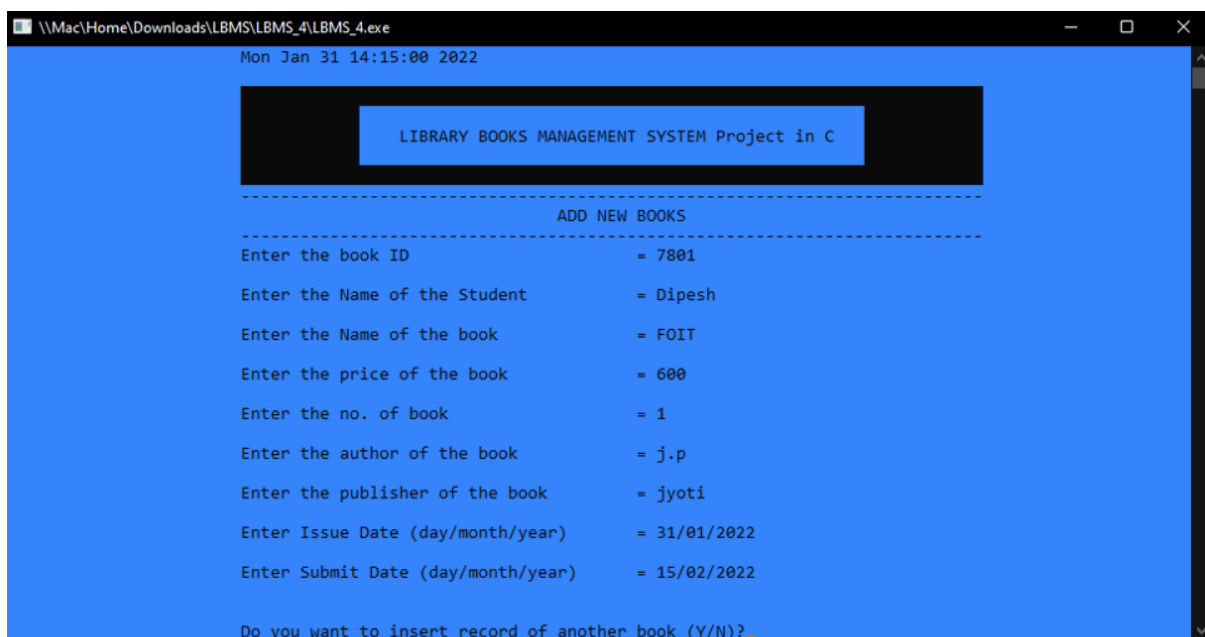
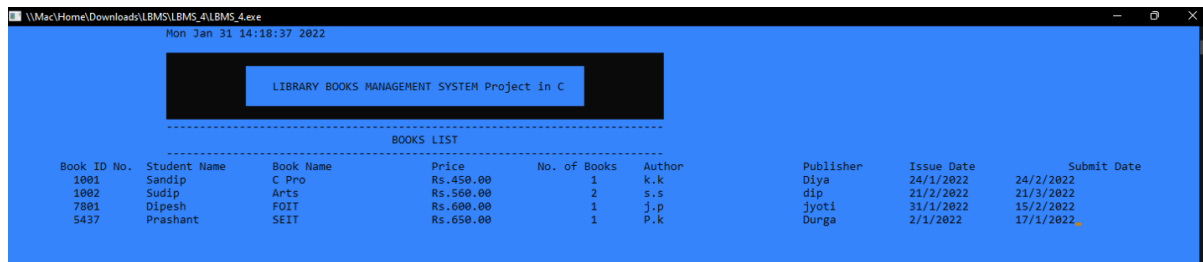


Figure 14 Insert Record



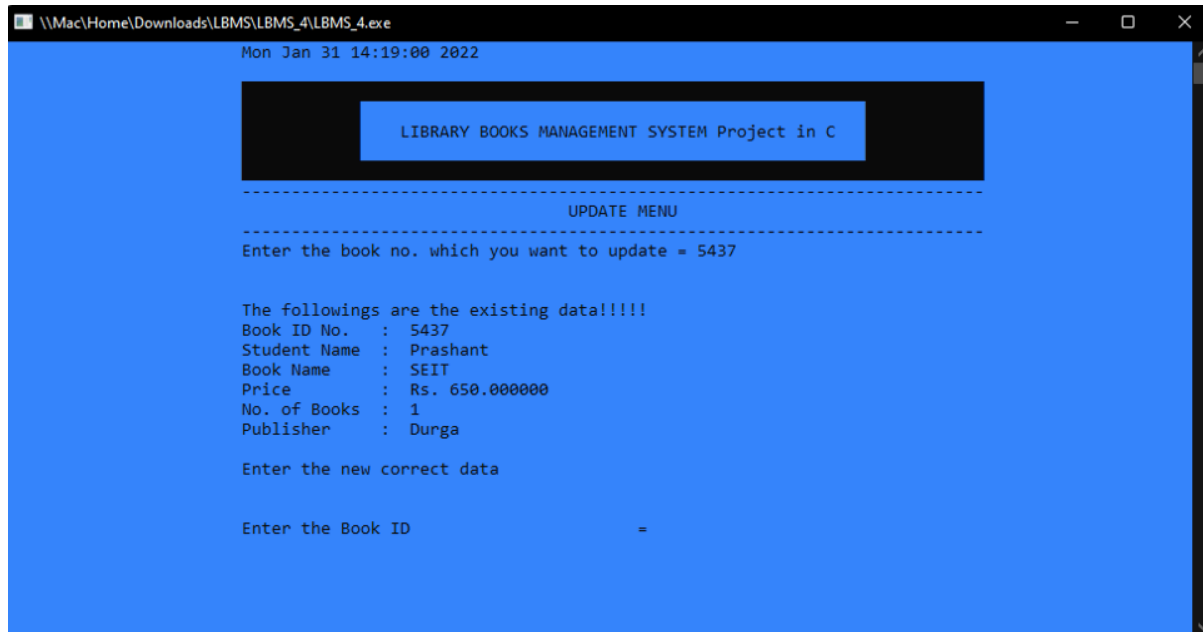
Mon Jan 31 14:18:37 2022

LIBRARY BOOKS MANAGEMENT SYSTEM Project in C

BOOKS LIST

Book ID No.	Student Name	Book Name	Price	No. of Books	Author	Publisher	Issue Date	Submit Date
1001	Sandip	C Pro	Rs.450.00	1	k.k	Diya	24/1/2022	24/2/2022
1002	Sudip	Arts	Rs.500.00	2	s.s	dip	21/2/2022	21/3/2022
7801	Dipesh	FOIT	Rs.600.00	1	j.p	jyoti	31/1/2022	15/2/2022
5437	Prashant	SEIT	Rs.650.00	1	P.k	Durga	2/1/2022	17/1/2022

Figure 15 Display Record



Mon Jan 31 14:19:00 2022

LIBRARY BOOKS MANAGEMENT SYSTEM Project in C

UPDATE MENU

Enter the book no. which you want to update = 5437

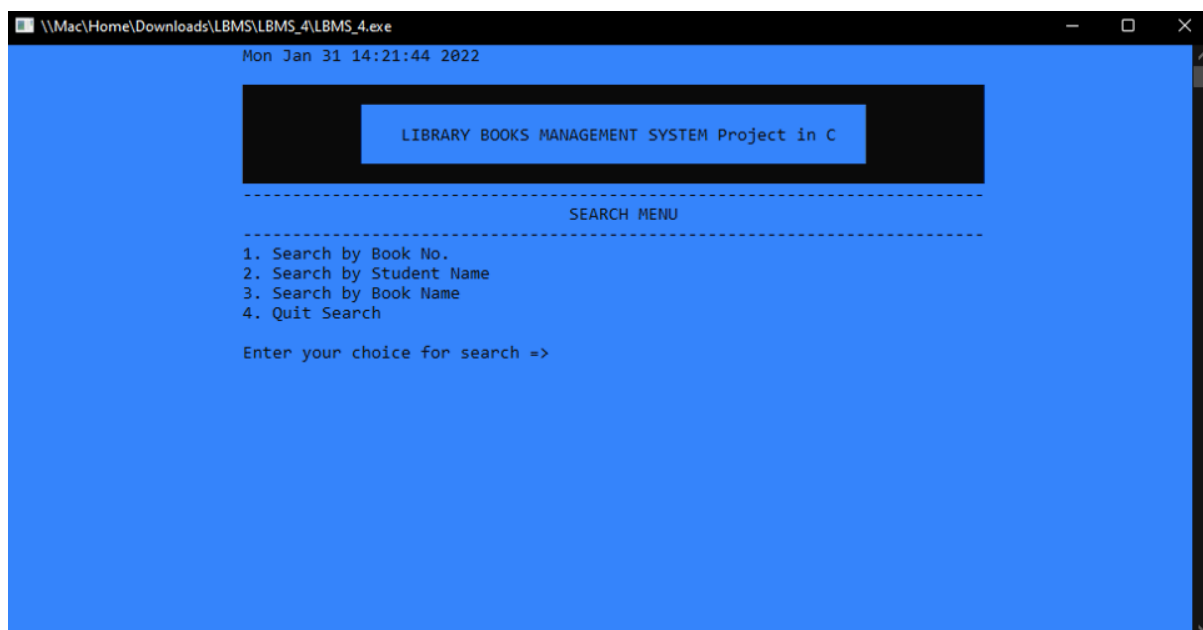
The followings are the existing data!!!!

Book ID No. : 5437
 Student Name : Prashant
 Book Name : SEIT
 Price : Rs. 650.000000
 No. of Books : 1
 Publisher : Durga

Enter the new correct data

Enter the Book ID =

Figure 16 Update Record



Mon Jan 31 14:21:44 2022

LIBRARY BOOKS MANAGEMENT SYSTEM Project in C

SEARCH MENU

1. Search by Book No.
 2. Search by Student Name
 3. Search by Book Name
 4. Quit Search

Enter your choice for search =>

Figure 17 Search Record

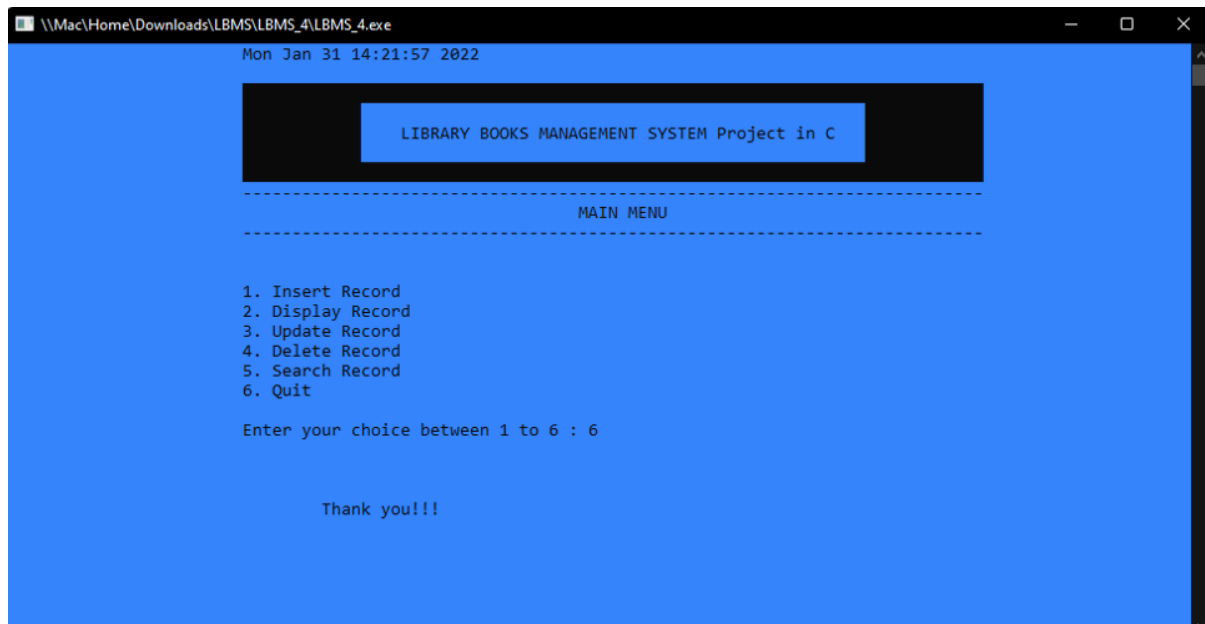


Figure 18 Exit Menu

4.1.2 Source code

```
#include<stdio.h>
#include<conio.h>
#include<time.h>
#include<process.h>
#include<string.h>
#include<stdlib.h>
#include<windows.h>

#define MAX_YR 9999
#define MIN_YR 1900

//structure to store date
typedef struct
{
    int yyyy;
    int mm;
    int dd;
} Date;

void printMessageCenter(const char* message)
{
    int len = 0;
    int pos = 0;

    len = (78 - strlen(message))/2;
    printf("\t\t\t");
    for(pos=0;pos<len;pos++)
    {
        printf(" ");
    }
}
```



```

float price;
int nob;
char author[100];
char publisher[100];
Date bookIssueDate;
Date bookSubmitDate;
};

//Code to declare structure variable
struct book b;
//Code to declare main function
/*##### main() #####*/
main()
{
    welcomeMessage();
    system("cls");
    login();
    int choice;
    int bn;
    while(1)
    {
        system("cls");
        //Code to display menu
        headMessage("MAIN MENU");

        printf("\n\n\t\t1. Insert Record");
        printf("\n\t\t2. Display Record");
        printf("\n\t\t3. Update Record");
        printf("\n\t\t4. Delete Record");
        printf("\n\t\t5. Search Record");
        printf("\n\t\t6. Quit");
        printf("\n\n\t\tEnter your choice between 1 to 6 : ");

        scanf("%d",&choice);
        switch(choice)
        {
            case 1:
                //Code to call insertData function
                insertData();
                break;
            case 2:
                //Code to call displayData function
                displayData();
                break;
            case 3:
                //Code to call updateData function
                headMessage("UPDATE MENU");
                printf("\n\t\tEnter the book no. which you want to update = ");
                scanf("%d",&bn);

```



```

    }
    fflush(stdin);
    printf("\n\t\tEnter the Name of the Student      = ");
    gets(b.stdName);
    printf("\n\t\tEnter the Name of the book          = ");
    gets(b.bookName);
    printf("\n\t\tEnter the price of the book          = ");
    scanf("%f",&b.price);
    printf("\n\t\tEnter the no. of book                = ");
    scanf("%d",&b.nob);
    fflush(stdin);
    printf("\n\t\tEnter the author of the book          = ");
    gets(b.author);
    printf("\n\t\tEnter the publisher of the book        = ");
    gets(b.publisher);
    printf("\n\t\tEnter Issue Date (day/month/year)      = ");

    scanf("%d/%d/%d",&b.bookIssueDate.dd,&b.bookIssueDate.mm,&b.bookIssueDate.
yyyy);

    printf("\n\t\tEnter Submit Date (day/month/year)   = ");

    scanf("%d/%d/%d",&b.bookSubmitDate.dd,&b.bookSubmitDate.mm,&b.bookSubmi
tDate.yyyy);

    fp=fopen("Book.dat","a");
    if(fp==NULL)
    {
        printf("\n\t\tFile creation error has occurred!!!");
    }
    else
    {
        fwrite(&b,sizeof(b),1,fp);
        fclose(fp);
        printf("\n\n\t\tDo you want to insert record of another book (Y/N)?");
        next=getche();
    }
}

/*#####displayData()#####*/
void displayData(void)
{
    system("cls");
    headMessage("BOOKS LIST");
    fp=fopen("Book.dat","r");
    rewind(fp);
    if(fp==NULL)
    {
        printf("\n\n\t\tRead operation failure as the file which your are searching
does not exixt!!!");
    }
}

```



```

        printf("\n\t\t\tBook Name   : %s",b.bookName);
        printf("\n\t\t\tPrice       : Rs. %2f",b.price);
        printf("\n\t\t\tNo. of Books  : %d",b.nob);
        printf("\n\t\t\tPublisher    : %s",b.publisher);
    }
}
//code to enter new data
rewind(fp);
//headMessage("Enter the new correct data");
printf("\n\n\t\t\tEnter the new correct data");
while(fread(&b,sizeof(b),1,fp)==1)
{
    if(bn==b.bn)
    {
        printf("\n\n\n\t\t\tEnter the Book ID           = ");
        scanf("%d",&b.bn);
        fflush(stdin);
        printf("\n\t\t\tEnter the Name of the Student       = ");
        gets(b.stdName);
        printf("\n\t\t\tEnter the Name of the book           = ");
        gets(b.bookName);
        printf("\n\t\t\tEnter the price of book           = ");
        scanf("%f",&b.price);
        printf("\n\t\t\tEnter the no. of book           = ");
        scanf("%d",&b.nob);
        fflush(stdin);
        printf("\n\t\t\tEnter the author of the book           = ");
        gets(b.author);
        printf("\n\t\t\tEnter the publisher of the book       = ");
        gets(b.publisher);
        fwrite(&b,sizeof(b),1,fp1);
    }
    else
    {
        fwrite(&b,sizeof(b),1,fp1);
    }
}
fclose(fp);
fclose(fp1);
remove("Book.dat");
rename("newBook.dat","Book.dat");
printf("\n\n\t\t\tThe record has been successfully updated in the data file");
}
getch();
}
/*#####deleteData()#####*/
void deleteData(int bn)
{
    int found = 0;

```

```

headMessage("DELETE MENU");
fp=fopen("Book.dat","r");
fp1=fopen("newBook.dat","w");

if(fp==NULL||fp1==NULL)
{
    printf("\nFile Operation failed");
}
else
{
    while(fread(&b,sizeof(b),1,fp))
    {
        if(bn!=b.bn)
        {
            fwrite(&b,sizeof(b),1,fp1);
            //continue;
        }
        else
        {
            found = 1;
        }
    }
    fclose(fp);
    fclose(fp1);
    remove("Book.dat");
    rename("newBook.dat","Book.dat");
    (found)? printf("\n\t\t\tRecord deleted successfully....."):printf("\n\t\t\tRecord
not found");
    //printf("\n\n\t\t\tThe record has been successfully deleted from the date file.");
}
getch();
}
/*##### searchData() #####*/
void searchData(void)
{
    int bn,found=0;
    char bname[100],subject[100];
    unsigned int countBook = 1;
    int ch;
    system("cls");
    headMessage("SEARCH MENU");
    printf("\n\t\t\t1. Search by Book No.");
    printf("\n\t\t\t2. Search by Student Name");
    printf("\n\t\t\t3. Search by Book Name");
    printf("\n\t\t\t4. Quit Search");
    printf("\n\n\t\t\tEnter your choice for search => ");
    scanf("%d",&ch);
    switch(ch)
    {
        case 1:

```

```

//search on the basis of book no.
system("cls");
headMessage("SEARCH BY BOOK ID ");
printf("\n\n\t\t\tEnter the book id no. which you want to search : ");
scanf("%d",&bn);
fp=fopen("Book.dat","r");
if(fp==NULL)
{
    printf("\n\t\t\tFile seach operation failed!!!!");
}
else
{
    while(fread(&b,sizeof(b),1,fp)==1)
    {
        if(b.bn==bn)
        {
            headMessage("RECORD");
            printf("\n\t\t\tBook ID No   : %d",b.bn);
            printf("\n\t\t\tStudent Name : %s",b.stdName);
            printf("\n\t\t\tBook Name   : %s",b.bookName);
            printf("\n\t\t\tPrice      : Rs. %2f",b.price);
            printf("\n\t\t\tNo. of Books : %d",b.nob);
            printf("\n\t\t\tAuthor     : %s",b.author);
            printf("\n\t\t\tPublisher  : %s",b.publisher);
            printf("\n\t\t\tBook Issue Date(day/month/year)
: (%d/%d/%d)\n",b.bookIssueDate.dd,b.bookIssueDate.mm, b.bookIssueDate.yyyy);
            printf("\n\t\t\tBook Submit Date(day/month/year)
: (%d/%d/%d)\n",b.bookSubmitDate.dd,b.bookSubmitDate.mm, b.bookSubmitDate.yyyy);

            found=1;
        }
    }
    if(found==0)
    {
        headMessage("WARNING!!!");
        printf("\n\t\t\t!!!!!!!!!!!!!!!!!!!!");
        printf("\n\t\t\tThe book which you are searching is not
found");
    }
    fclose(fp);
}
break;
case 2:
    //seach on the basis of Student Name
    system("cls");
    fflush(stdin);
    headMessage("SEARCH BY STUDENT NAME ");
    printf("\n\n\t\t\tEnter the Name of the Student which you want
to search = ");
    gets(bname);

```



```

fp=fopen("Book.dat","r");
if(fp==NULL)
{
    printf("\n\t\t\tFile search operation failed!!!!");
}
else
{
    while(fread(&b,sizeof(b),1,fp)==1)
    {
        if(strcmpi(b.stdName,bname)==0)
        {
            printf("\n\t\t\tBook Count =
%d\n",countBook);

            printf("\n\t\t\tBook ID No. : %d",b.bn);
            printf("\n\t\t\tBook Name :
%s",b.bookName);

            printf("\n\t\t\tStudent Name :
%s",b.stdName);

            printf("\n\t\t\tPrice :
Rs. %2f",b.price);

            printf("\n\t\t\tNo. of Books :
%d",b.nob);

            printf("\n\t\t\tAuthor : %s",b.author);
            printf("\n\t\t\tPublisher :
%s",b.publisher);

            printf("\n\t\t\tBook Issue
Date(day/month/year) : (%d/%d/%d)\n",b.bookIssueDate.dd,b.bookIssueDate.mm,
b.bookIssueDate.yyyy);

            printf("\n\t\t\tBook Submit
Date(day/month/year) : (%d/%d/%d)\n",b.bookSubmitDate.dd,b.bookSubmitDate.mm,
b.bookSubmitDate.yyyy);

            ++countBook;
            found=1;
        }
        if(found==0)
        {
            printf("\n\t\t\t!!!!!!!!!!!!");
            printf("\n\t\t\tThe book which you are
searching is not found");
        }
        fclose(fp);
    }
    break;
}
case 3:
    //search on the basis of Book Name
    system("cls");
    headMessage("SEARCH BY NAME OF THE BOOK ");
    fflush(stdin);

```

```

search = "");

printf("\n\n\t\t\tEnter the Name of the book which you want to

gets(subject);
fp=fopen("Book.dat","r");
if(fp==NULL)
{
    printf("\n\t\t\tFile search operation failed!!!!");
}
else
{
    while(fread(&b,sizeof(b),1,fp)==1)
    {
        if(strcmpi(b.bookName,subject)==0)
        {
            printf("\n\t\t\tBook Count =

%d\n",countBook);

            printf("\n\t\t\tBook ID No. : %d",b.bn);
            printf("\n\t\t\tStudent Name :

%s",b.stdName);

            printf("\n\t\t\tBook Name :

%s",b.bookName);

            printf("\n\t\t\tPrice :

Rs.%2f",b.price);

            printf("\n\t\t\tNo of Books :

%d",b.nob);

            printf("\n\t\t\tAuthor : %s",b.author);
            printf("\n\t\t\tPublisher :

%s\n",b.publisher);

            ++countBook;
            found=1;
        }
    }
    if(found==0)
    {
        printf("\n\t\t\t!!!!!!!!!!!!");
        printf("\n\t\t\tThe book which you are searching

is not found");
    }
    fclose(fp);
}
break;
case 4:
    //code to close search window
    printf("\n\t\t\tPress any key to quit the search operation");
    break;
default:
    printf("\n\t\t\tYou entered invalid choice for search");
}
getch();
}

```

```

/*##### checkBookNo() #####*/

int checkBookNo(int bn)
{
    int repeated=0,lastbookno;
    struct book b1;
    fp=fopen("Book.dat","r");
    if(fp==NULL)
    {
        printf("\n\n\t\t\tRead operation failure as the file which you are searching does
not exist!!!");
    }
    else
    {
        while(fread(&b1,sizeof(b1),1,fp)==1)
        {
            if(b1.bn==bn)
            {
                repeated=1;
                break;
            }
        }
        fclose(fp);
    }
    return(repeated);
}

/*##### findLastBookNo() #####*/
int findLastBookNo()
{
    int lastbookno;
    struct book b2;
    fp=fopen("Book.dat","r");
    if(fp==NULL)
    {
        printf("\n\n\t\t\tFile open operation failure!!!");
    }
    else
    {
        while(fread(&b2,sizeof(b2),1,fp)==1)
        {
            lastbookno=b2.bn;
        }
        fclose(fp);
    }
    return(lastbookno);
}

/*##### Login menu() #####*/
void login()
{
    int a=0,i=0;

```

```

char uname[10],c=' ';
char pword[10],code[10];
char user[10]="user";
char pass[10]="pass";
do
{
    headMessage("Login");
    printf("\n\n\t\t\tUsername: ");
    scanf("%s", &uname);
    printf("\n\t\t\tPassword: ");
    while(i<10)
    {
        pword[i]=getch();
        c=pword[i];
        if(c==13) break;
        else printf("*");
        i++;
    }
    pword[i]='\0';
    i=0;
    if(strcmp(uname,"kist")==0 && strcmp(pword,"kist")==0)
    {
        printf("\n\n\n\t\t\tLOGIN IS SUCCESSFUL");
        printf("\n\n\n\t\t\tPress any key to continue...");
        getch();
        break;
    }
    else
    {
        printf("\n\n\n\t\t\tLogin Failed! Enter Again Username & Password");
        a++;
        getch();
        system("cls");
    }
}
while(a<=2);
if (a>2)
{
    headMessage("Login Failed");
    printf("\n\t\t\tYou have attempted to Login with wrong password too many
times.");
    getch();
    exit(0);
}
system("cls");
}

```