A

MINI PROJECT REPORT

ON

"MEDITRACKER"

Submitted in the partial fulfillment of the requirements in the IV semester of

BACHELOR OF ENGINEERING

IN

INFORMATION SCIENCE AND ENGINEERING

BY

AKHILA S - [1NH17IS008]

FOR

COURSE NAME: MINI PROJECT
COURSE CODE: ISE46

Under the guidance of

Prof. VANDANA C P
Sr.Assistant Professor
Dept. of ISE, NHCE

DEPARTMENT OF INFORMATION SCIENCE AND ENGINEERING

NEW HORIZON COLLEGE OF ENGINEERING
(Autonomous College Permanently Affiliated to VTU, Approved by AICTE, Accredited by NAAC with 'A' Grade & NBA)
Ring Road, Bellandur Post, Near Marathahalli,
Bengaluru-560103, INDIA

NEW HORIZON
COLLEGE OF ENGINEERING
Autonomous College Permanently Affiliated to VTU, Approved by AICTE & UGC
Accredited by NAAC with 'A' Grade. Accredited by NBA
nirf TOP ENGINEERING COLLEGE OF INDIA as per
NATIONAL INSTITUTIONAL RANKING FRAMEWORK, 2018
Ministry of Human Resource Development, Government of India

CERTIFICATE

Certified that the project work entitled Meditracker carried out by Ms., Akhila S USN-1NH17IS008, a bonafied student of IV semester in partial fulfillment for the award of Bachelor of Engineering in Information Science and Engineering of the Visveswaraiah Technological University, Belgaum during the year 2018-19. It is certified that all corrections/suggestions indicated for Internal Assessment have been incorporated. The project report has been approved as it satisfies the academic requirements in respect of Project work prescribed for the said Degree.

Signature of the Guide	Signature of the HOD	Signature of the Principal
Prof. Vandana C P	Dr. R.J Anandhi	Dr.Manjunatha

ABSTRACT

Meditracker is an application which is used by both doctors and patients. It is very tough for some people to remember what medicine has to be taken at what time. So this makes our work easier. The main aim of this application is it acts like a health journal. Not only does it assesses your health but also acts as a great tool when you visit your doctor. The features are monitoring prescriptions given to the patients, fixing doctor's appointment, keeping all your health records in one place including the doctor's contact, details of your health insurance and also give a report of all your health activities. One of the main flaws of the existing system is that the records are maintained manually and there might be errors. Hence the user can track his medical records whenever required. The project is implemented using C programming language using the concepts of file handling.

ACKNOWLEDGEMENT

Any achievement, be it scholastic or otherwise does not depend solely on the individual efforts but on the guidance, encouragement and cooperation of intellectuals, elders and friends. A number of personalities, in their own capacities have helped me in carrying out this mini project. I would like to take this opportunity to thank them all.

I thank the management, **Dr. Mohan Manghnani,** Chairman, New Horizon Educational Institutions for providing necessary infrastructure and creating conducive environment for effective learning.

I also record here the constant encouragement, support and facilities extended to us by **Dr. Manjunatha,** Principal, New Horizon College of Engineering, Bengaluru.

I extent sincere gratitude for constant encouragement and facilities provided to us by **Dr. R.J Anandhi**, Professor and Head of the Department, Department of Information Science and Engineering, New Horizon College of Engineering, Bengaluru.

I sincerely acknowledge the encouragement, timely help and guidance to me by **Prof. Vandana C P, S**r.Assistant Professor, Department of Information Science and Engineering, New Horizon College of Engineering, Bengaluru, to complete the mini project within stipulated time successfully.

Finally, a note of thanks to the teaching and non-teaching staff of Information Science and Engineering Department for their cooperation extended to us and our friends, who helped me directly or indirectly in the successful completion of this mini project.

Akhila S

TABLE OF CONTENTS

Chapter 1

INTRODUCTION

Meditracker is an e service provided to both doctors as well as patients with easy to use customizable options. The application will basically lessen the manual work and improve the quality of maintaining records and other information related to doctors or patients. It reduces time frame in adding any information related to hospital and thereby reduce the complexity too. The application allows patients to book appointment with doctors based on his availability. The application consists of patient module where they can view their records, book an appointment. Doctor module where he can view the patient's medical history and note down their vital as well. And prescribe them with medicines, modify his appointment availability and view his appointments. Admin module where they can add the records of new patient, delete their records, update them.

1.1 Motivation of Project

The purpose of creating a Meditracker is to add, edit, delete and update the patient record. It is also used to book appointments with doctors, view the doctor's availability and the doctor can also change his availability status. This reduces a lot of manual work. If we look at traditional methods, we understand that manual approach is needed where the records are handwritten and there are chances that the data entered might be wrong. The computerized accounting of data offers unlimited storage of memory and the information retrieval is also easy. It is one of the best improvements of the traditional book keeping method.

The data is stored in the form of tables which has proven easy to comprehend and effective in the long run. The hospital also must maintain a proper record of all the patients in a database so that it is easily accessible in the future.

1.2 Problem Statement

To design and implement an application which allows the user to track his/her medical records and fix appointments with the doctor based on his availability. Also supports Doctor to reschedule and manage his appointments.

Chapter 2

SYSTEM REQUIREMENT SPECIFICATION

Purpose: To design an application -this allows the user to track his/her medical records and fix appointments with the doctor based on his availability. Also supports Doctor to reschedule and manage his appointments.

2.1 Hardware System Configuration:

Processor	- Intel core i3
Speed	- 1.8 GHz
RAM	- 256 MB (min)
Hard Disk	- 1 GB

2.2 Software System Configuration:

Operating System	- Windows 8.1
Programming Language	- C
Compiler	- Turbo C outcomes

Chapter 3

METHODOLOGY

3.1 Modules

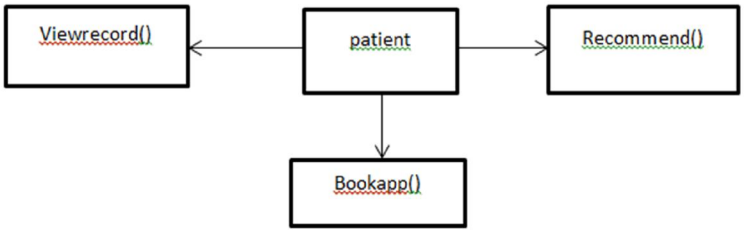


Figure 3.1.1 Pateint module

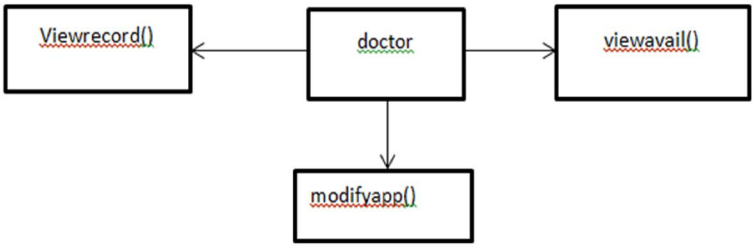
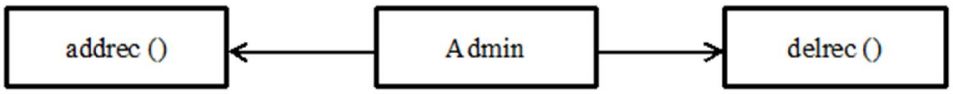


Figure 3.1.2 Doctor module



Figue 3.1.3 Admin module

3.2 Algorithm

Step 1: Create a structure with patients name, age, gender, disease history.

Step 2: Display a menu- Admin, Doctor, Patient

Input user choice

switch (ch)

case 1: admin()

case 2: doctor()

case 3: patient()

Step 3: In admin module:

Display menu

if(ans==1) addrec()

if(ans==2) delrec()

In doctor module

Display menu

if(ans==1) viewrec()

if(ans==2) modifyavail()

In patient module

Display menu

if(ans==1) viewrec()

if(ans==2) viewapp()

if(ans==3) bookapp()

Step 4: In add record module:

Open a file (pdetails.txt) in append mode

Ask user to enter the necessary details

Close the file

Step 5: In delete record module:

input the name of the patient whose records has to be deleted

Open the file (pdetails.txt) in read mode

Open the file (temp.txt) in write mode

while(!feof)

if (p.pname != name)

fwrite(&p, sizeof(p), 1, pa1);

close the files

Step 6: In view record module:

open the file (pdetails.txt) in read mode

Input the name of the patient whose records you need to view

while(!feof)

if(strcmp(name,p.pname)==0)

display the details

close the file

Step 7: In book appointment module:

Open the file (avail.txt) in read mode

Input the date you want to book appointment on

while(!feof)

if(strstr(line,date))

display appointment booked

close the file

Step 8: In view availability module:

Open the file (avail.txt) in read mode

if(fp==NULL)
display error

else

read the file contents

close the file

Step 9: In modify availability module:

Open the file (avail1.txt) in read mode and the file (temp.txt) in write mode

Input the data you want to replace in the file

while(!feof)

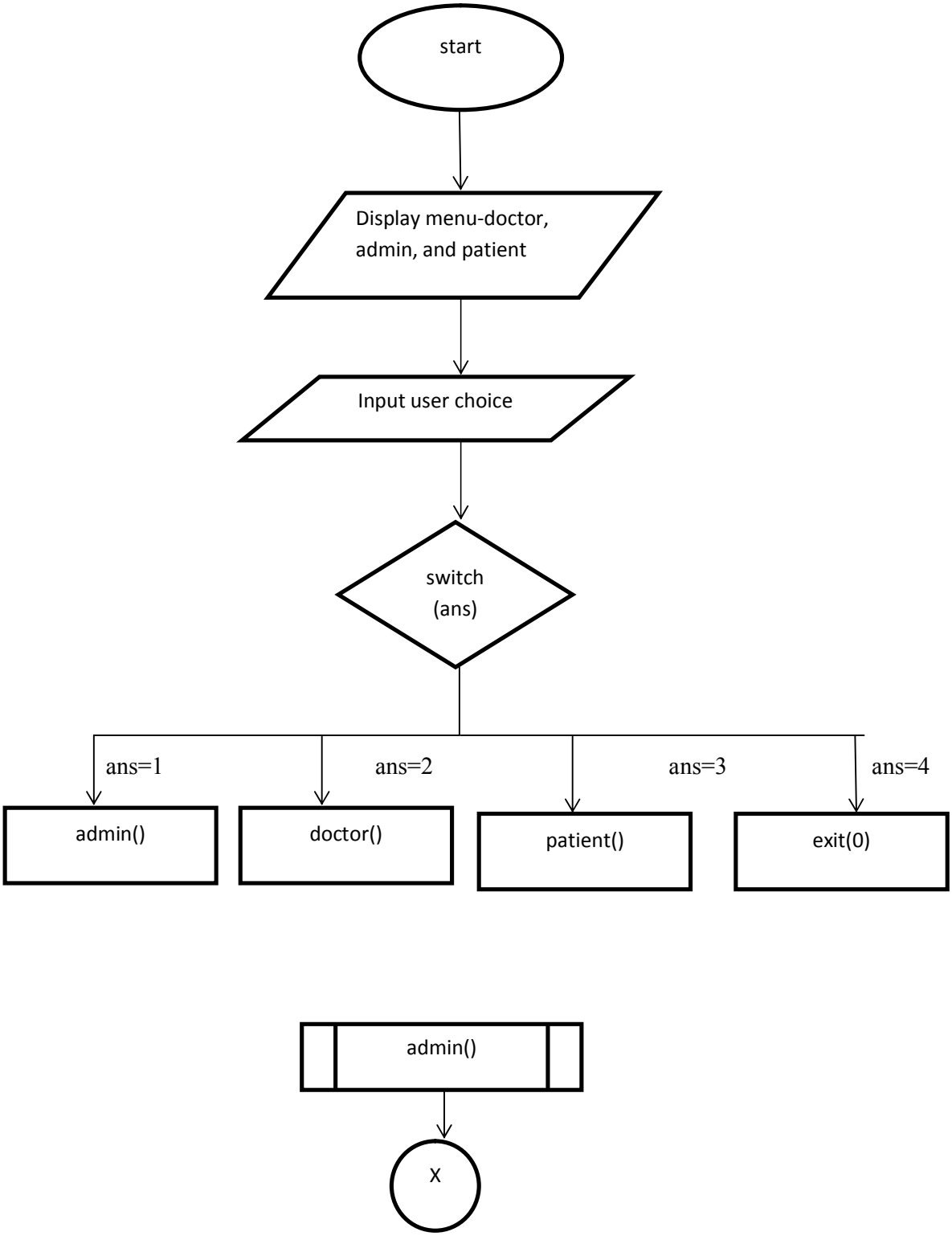
copy the content of avail.txt file except for the line you want to replace

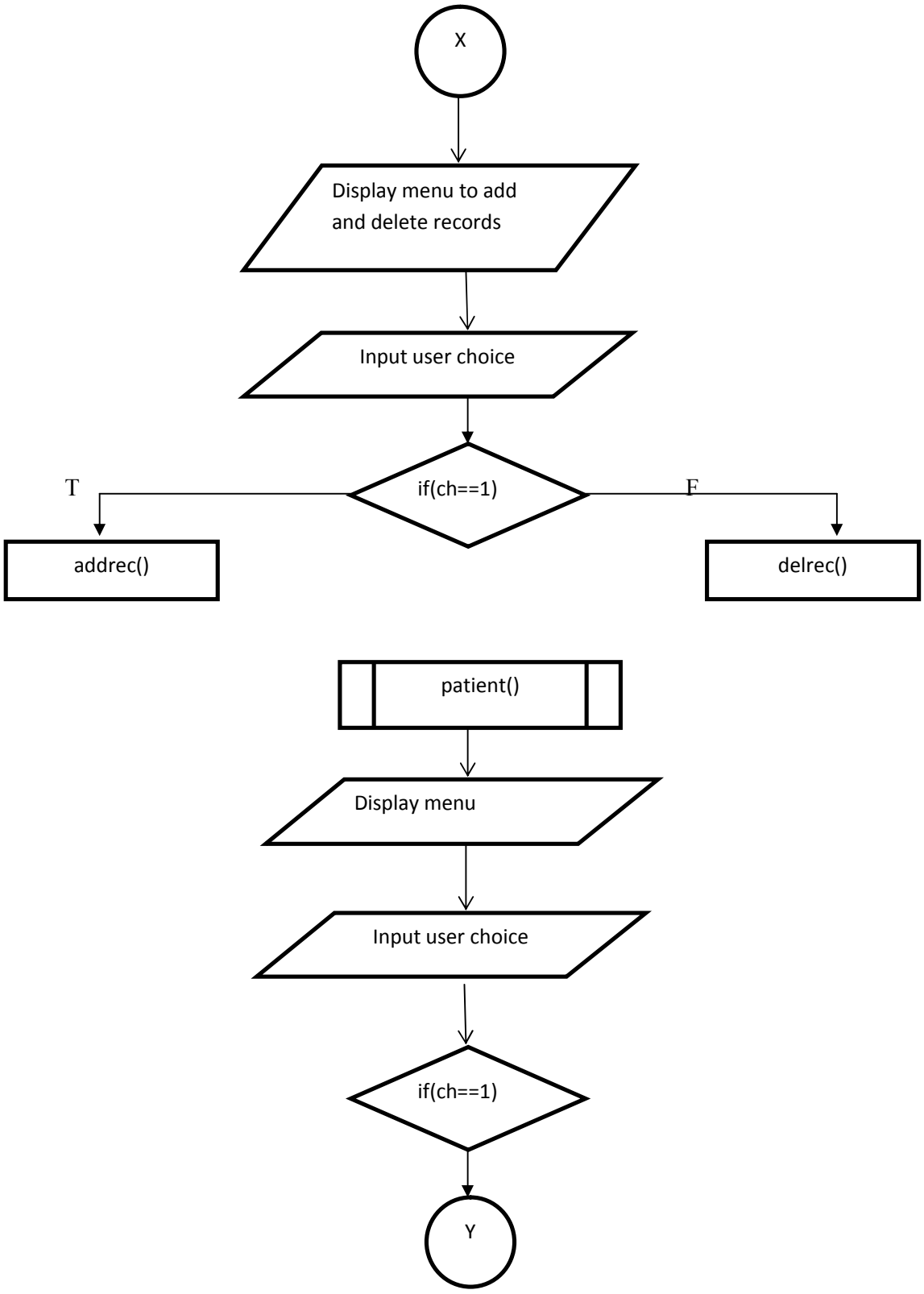
close the files

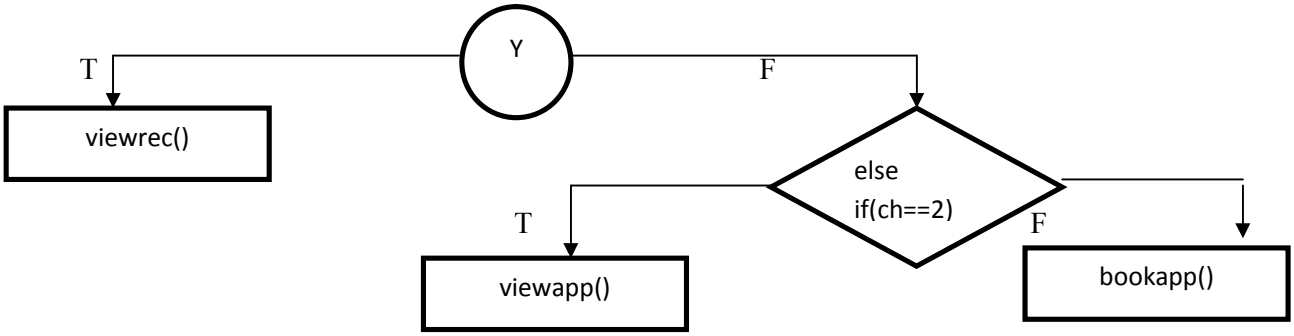
rename the temp.txt file as avail1.txt

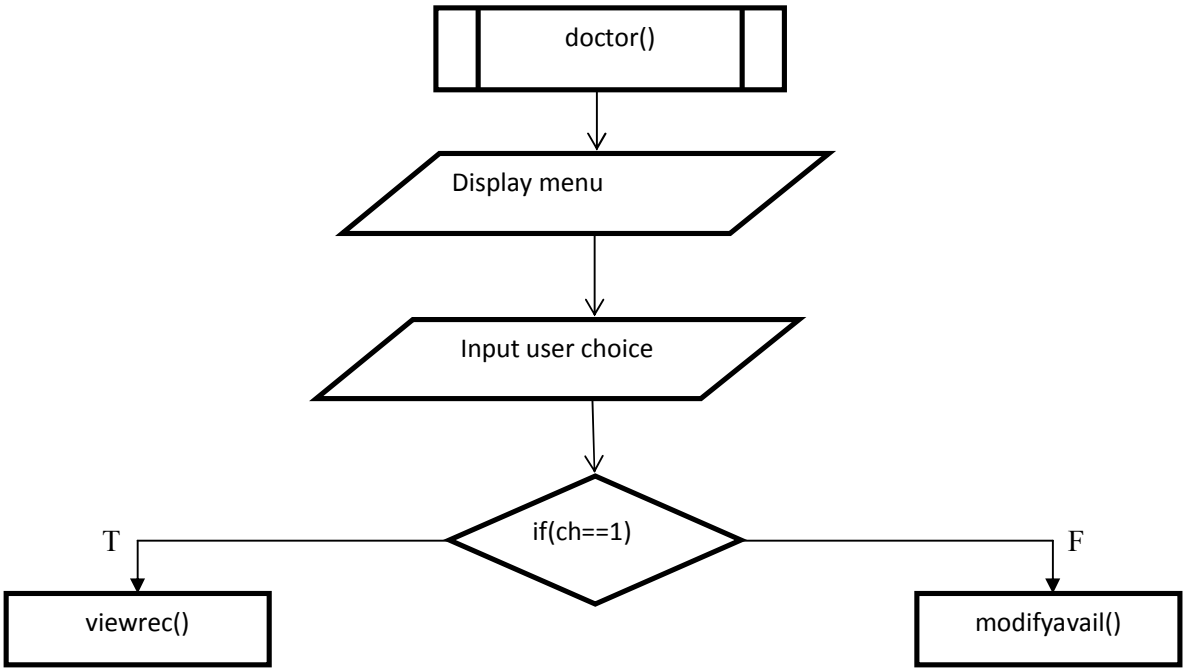
Step: 10 Stop

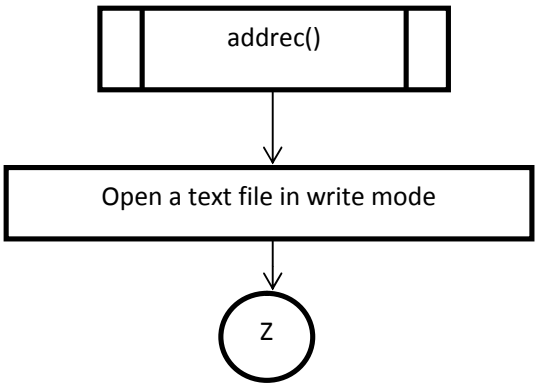
3.3 Flowchart

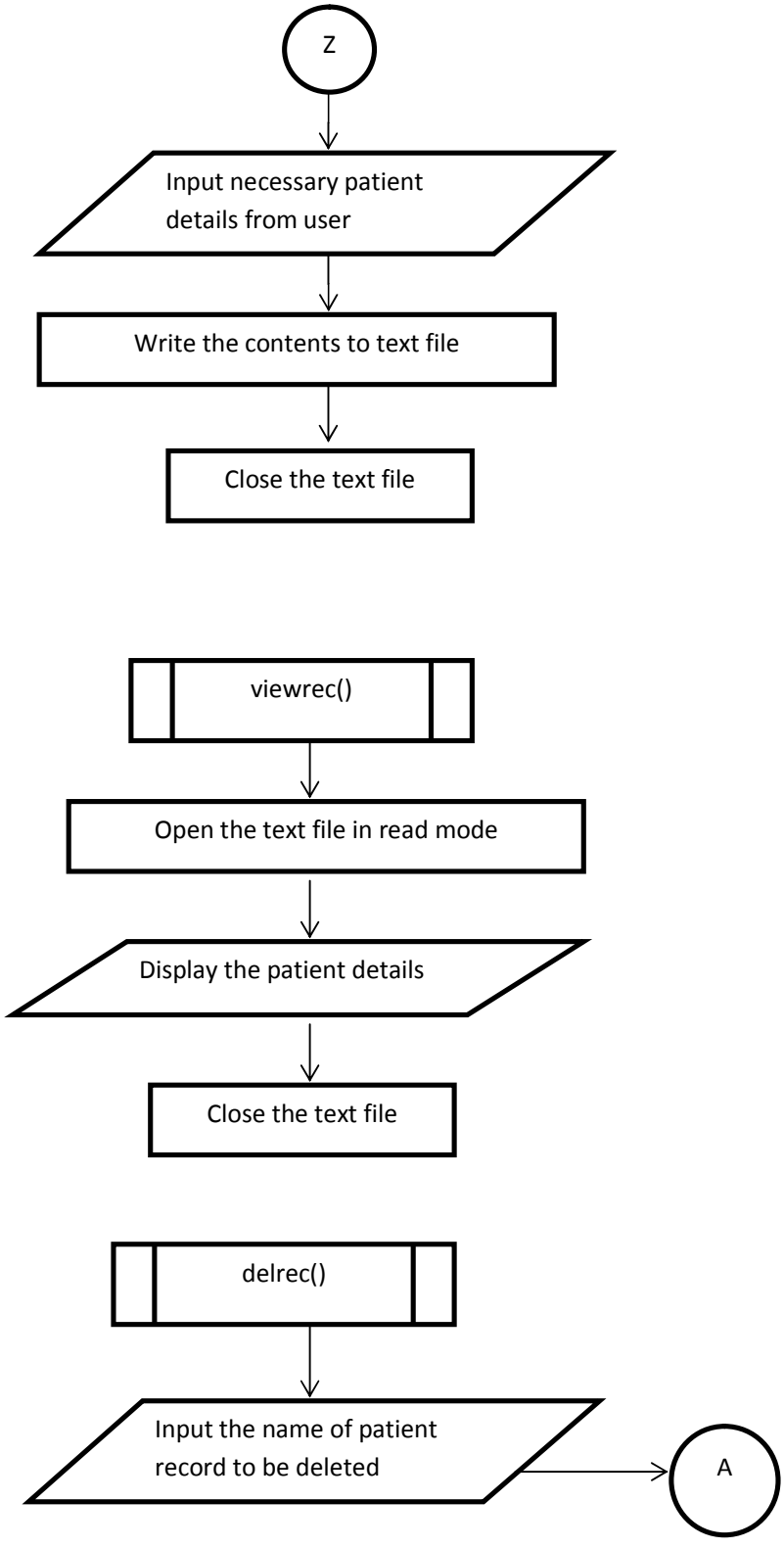


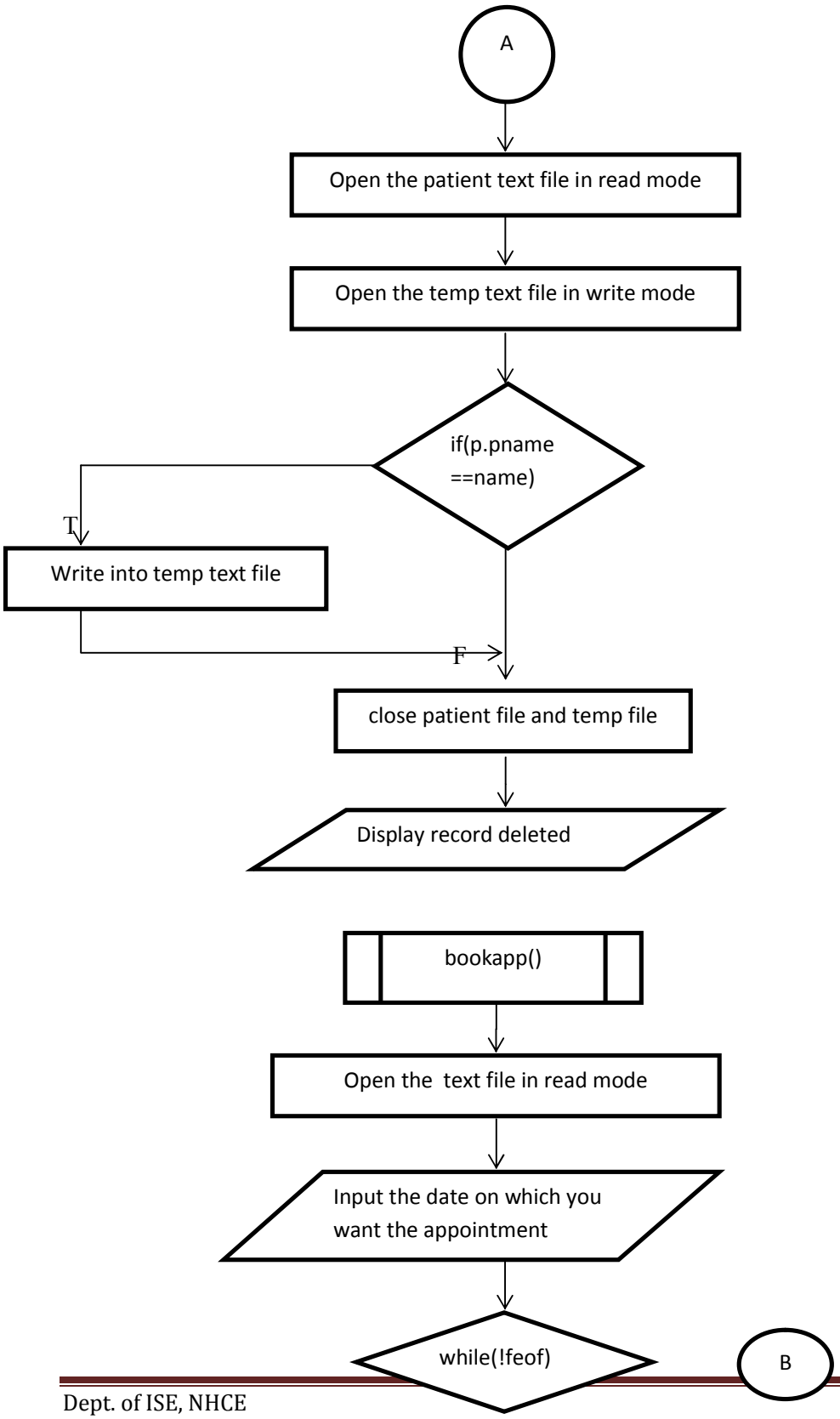

X
Display menu to add
and delete records
Input user choice
if(ch==1)
T
F
addrec()
delrec()
patient()
Display menu
Input user choice
if(ch==1)
Y

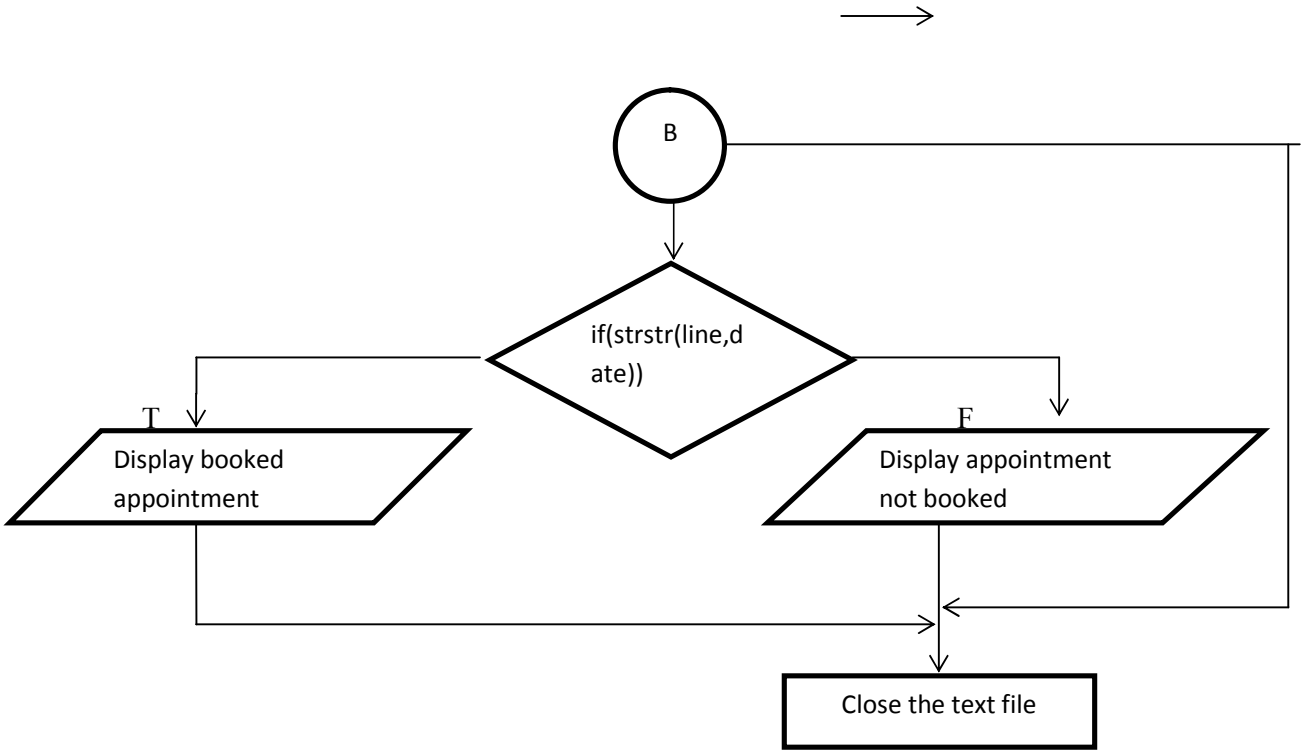

Y
T
F
viewrec()
else
if(ch==2)
T
F
viewapp()
bookapp()

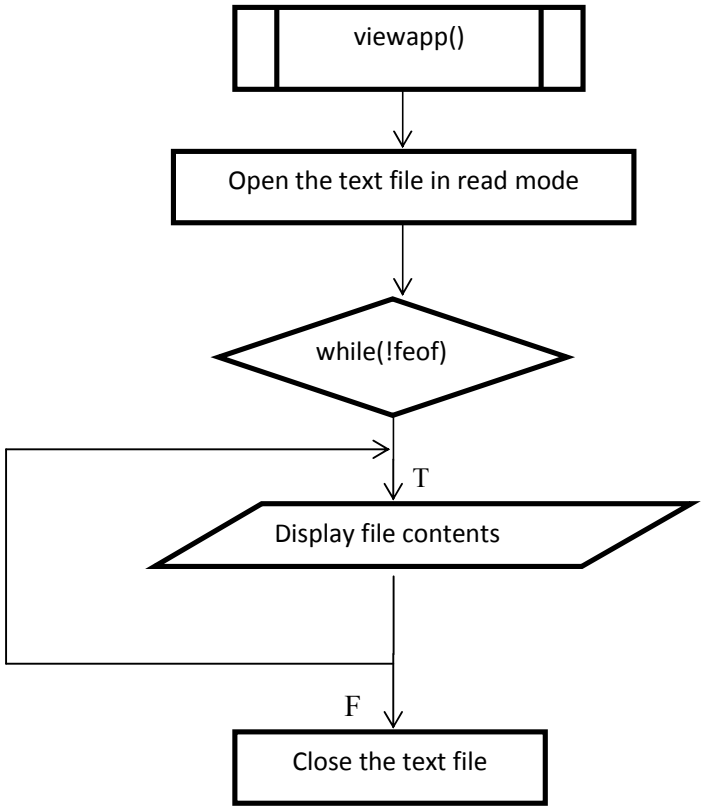

doctor()
Display menu
Input user choice
if(ch==1)
T
F
viewrec()
modifyavail()

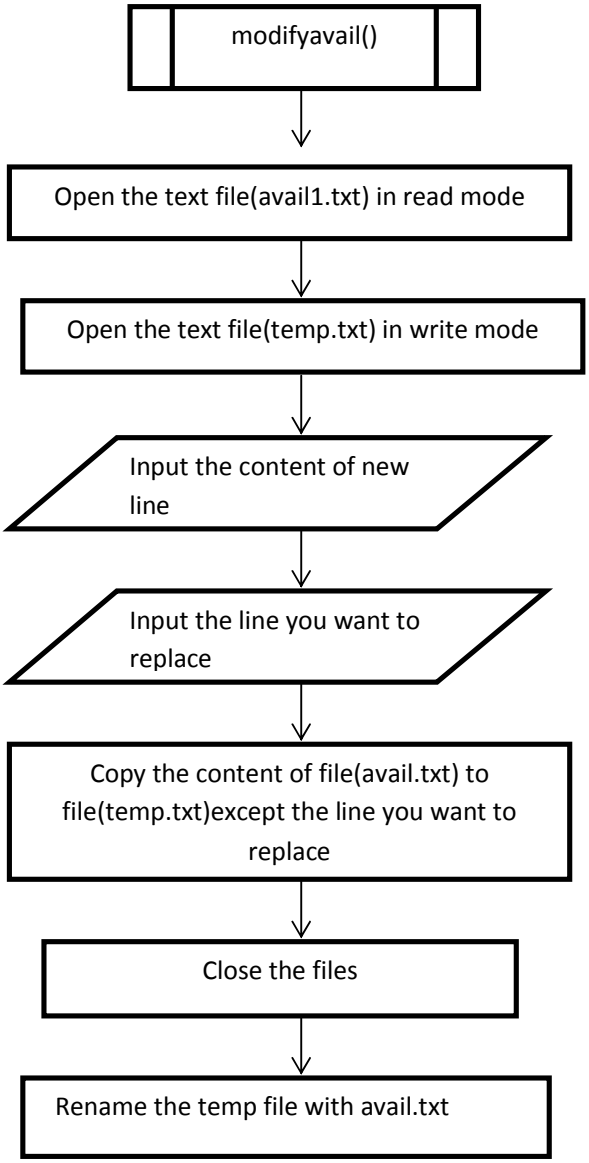

addrec()
Open a text file in write mode
Z


Z
Input necessary patient details from user
Write the contents to text file
Close the text file
viewrec()
Open the text file in read mode
Display the patient details
Close the text file
delrec()
Input the name of patient record to be deleted
A


A
Open the patient text file in read mode
Open the temp text file in write mode
if(p.pname
==name)
T
Write into temp text file
F
close patient file and temp file
Display record deleted
bookapp()
Open the text file in read mode
Input the date on which you
want the appointment
while(!feof)
B


B
if(strstr(line,d
ate))
T
Display booked
appointment
F
Display appointment
not booked
Close the text file


viewapp()
Open the text file in read mode
while(!feof)
T
Display file contents
F
Close the text file


modifyavail()
Open the text file(avail1.txt) in read mode
Open the text file(temp.txt) in write mode
Input the content of new line
Input the line you want to replace
Copy the content of file(avail.txt) to file(temp.txt)except the line you want to replace
Close the files
Rename the temp file with avail.txt

3.4 Code and Implementation

```
#include<stdio.h>

#include<conio.h>

#include<stdlib.h>

#include<string.h>

#define MAX 256

struct patient

{

    //char pid[20];

    char pname[50];

    int age;

    char gender[10];

    char diseasehistory[50];

};

struct patient p;

void addrec( )

{

   FILE *pa;

    pa=fopen("pdetails.txt","w");

    //printf("\nEnter patient id.\n");
```

```
    //scanf("%s",p.pid);
    printf("Enter patient Name.\n");
    scanf("%s",p.pname);
    printf("Enter patients age.\n");
    scanf("%d",&p.age);
    printf("Enter patient gender.\n");
    scanf("%s",p.gender);
    printf("Enter patient diseasehistory.\n");
    scanf("%s",p.diseasehistory);
    fwrite(&p,sizeof(p),1,pa);
    fclose(pa);
}
void viewrec( )
{
   FILE *pa;
   char name[10];
   pa = fopen("pdetails.txt","r");
   printf("enter the name of the patient whose file to be opened    :->");
   scanf("%s",name);
   while(fread(&p,sizeof(p),1,pa))
     {
        if(strcmp(name,p.pname)==0)
```

```
            {
                printf("                details of patient %s is",p.pname);
                printf("\n--------------------------------------------------------------------------\n");
                printf( "\n  pname \t age \t gender \t diseasehistory \n" );
                printf("--------------------------------------------------------------------------\n");
                printf("%s\t%d\t%s\t%s\n",p.pname,p.age,p.gender,p.diseasehistory);
                fwrite(&p,sizeof(p),1,pa);
            }
        }
    fclose(pa);
}
void delrec()
{
    FILE *pa;
    FILE *pa1;
    char name[10];
    printf("Enter the name of the patient:");
    scanf("%s", name);
    pa = fopen("pdetails.txt", "r");
    pa1 = fopen("temp.txt", "w");
15
    while (fread(&p, sizeof(p), 1, pa))
     {
```

```
        if (p.pname != name)
            fwrite(&p, sizeof(p), 1, pa1);
    }
   fclose(pa);
   fclose(pa1);
   pa = fopen("pdetails.txt", "w");
   pa1 = fopen("temp.txt", "r");
   while (fread(&p, sizeof(p), 1, pa1))
    fwrite(&p, sizeof(p), 1, pa);
   printf("\nRECORD DELETED\n");
    fclose(pa);
   fclose(pa1);
}
void bookapp()
{
    FILE *fp,*fp1;
   char line[200];
  char date[10];
   fp=fopen("avail.txt","r");
   if(!fp)
   {
     printf("could not find the file");
```

```
     exit(0);
    }
    printf("\n enter the date you want to book appointment on(in dd format)\n");
    scanf("%s",date);
   while ( fgets ( line, 200, fp ) != NULL ) /* read a line */
   {
      if(strstr(line,date))
        {
          printf("%s has booked an appointment on %s",p.pname,date);
          }
        else
         printf("\n appointment not available");
        break;
      }
  fclose ( fp );
}
void viewapp()
{
   char ch;
   FILE *fp;
   fp = fopen("avail.txt", "r"); // read mode
   if (fp == NULL)
```

```
  {
    perror("Error while opening the file.\n");
    exit(0);
  }
    while((ch = fgetc(fp)) != EOF)
    printf("%c", ch);
    fclose(fp);
}
void modifyavail()
{
   FILE *fptr1, *fptr2;
  int lno, linectr = 0;
  char str[50];
  char newln[50];
  fptr1 = fopen("avail1.txt", "r");
  if (!fptr1)
   {
    printf("Unable to open the input file!!\n");
    }
    fptr2 = fopen("temp.txt", "w");
    if (!fptr2)
      {
```

```
  printf("Unable to open a temporary file to write!!\n");
  fclose(fptr1);
 }
printf(" Input the content of the new line :\n”);
scanf("%s",newln);
printf(" Input the line no you want to replace : ");
scanf("%d", &lno);
lno++;
while (!feof(fptr1))
{
   strcpy(str, "\0");
  fgets(str, MAX, fptr1);
    if (!feof(fptr1))
     {
       linectr++;
       if (linectr != lno)
          {
             fprintf(fptr2, "%s", str);
          }
          else
          {
             fprintf(fptr2, "%s", newln);
```

```
                }
            }
        }
        fclose(fptr1);
        fclose(fptr2);
        remove("avail1.txt");
        rename("temp.txt", "avail1.txt");
        printf(" Appointment modified successfully..!! \n");
}
void doctor()
{
  int ch;
  printf("\n 1 View records\n 2 Modify appointments\n");
 printf("Choose one:\n");
 scanf("%d",&ch);
 if(ch==1)
  viewrec();
  else
   modifyavail();
}
void patient()
{
```

```
  int ch;
  printf("1 View records\n 2 View availability\n3 Book appointments\n");
 printf("Choose one:\n");
 scanf("%d",&ch);
 if(ch==1)
 viewrec();
 else if(ch==2)
 viewapp();
  else
 bookapp();
}
void admin()
{
   int  ch;
  printf("1 Add records\n 2 Delete records\n");
 printf("Choose one:\n");
 scanf("%d",&ch);
 if(ch==1)
 addrec();
 else
  delrec();
}
```

```
void main( )
{
   int ans;
   char ch;
    clrscr();
    printf("\n*****WELCOME!!!*****\n");
   do
   {
     printf("1 ADMIN\n 2 DOCTOR \n 3 PATIENT\n 4 EXIT\n");
     printf("Choose one:\n");
    scanf("%d",&ans);
    switch(ans)
     {
        case 1: admin();
                  break;
         case 2: doctor();
                  break;
        case 3:  patient();
                  break;
        case 4: printf("exit\n");
                 exit(0);
        default: printf("choice not valid!\n");
```

```
        }
    } while (ans!= 4);
 getch();
}
```

Chapter 4

RESULTS AND DISCUSSION

4.1 Explanation

The admin is given 2 options:

1. To add new records of patient
2. To delete the records of patient

The doctor is given 2 options:

1. To view patient records
2. To modify the schedule

The patient is given 3 options:

1. To view the records
2. To vie doctors availability
3. To book an appointment

4.2 Output (Snapshots)

```
******WELCOME!!!******
1 ADMIN
 2 DOCTOR
 3 PATIENT
 4 EXIT
Choose one:
1
1 Add records
 2 Delete records
Choose one:
1

Enter patient id.
1321
Enter patient Name.
akhila
Enter patients age.
19
Enter patient gender.
female
Enter patient diseasehistory.
none
```

Figure 4.2.1 Admin –to add new records

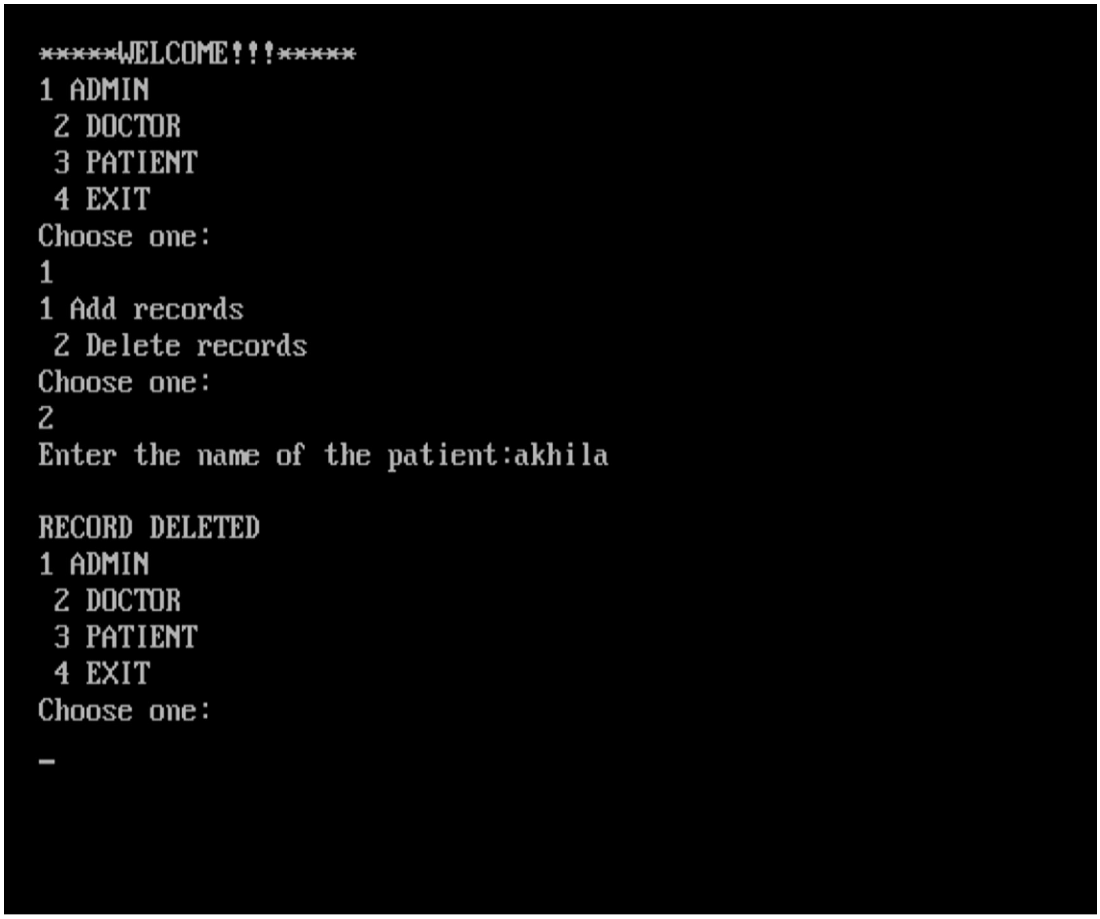


Figure 4.2.2 Admin-to delete the records

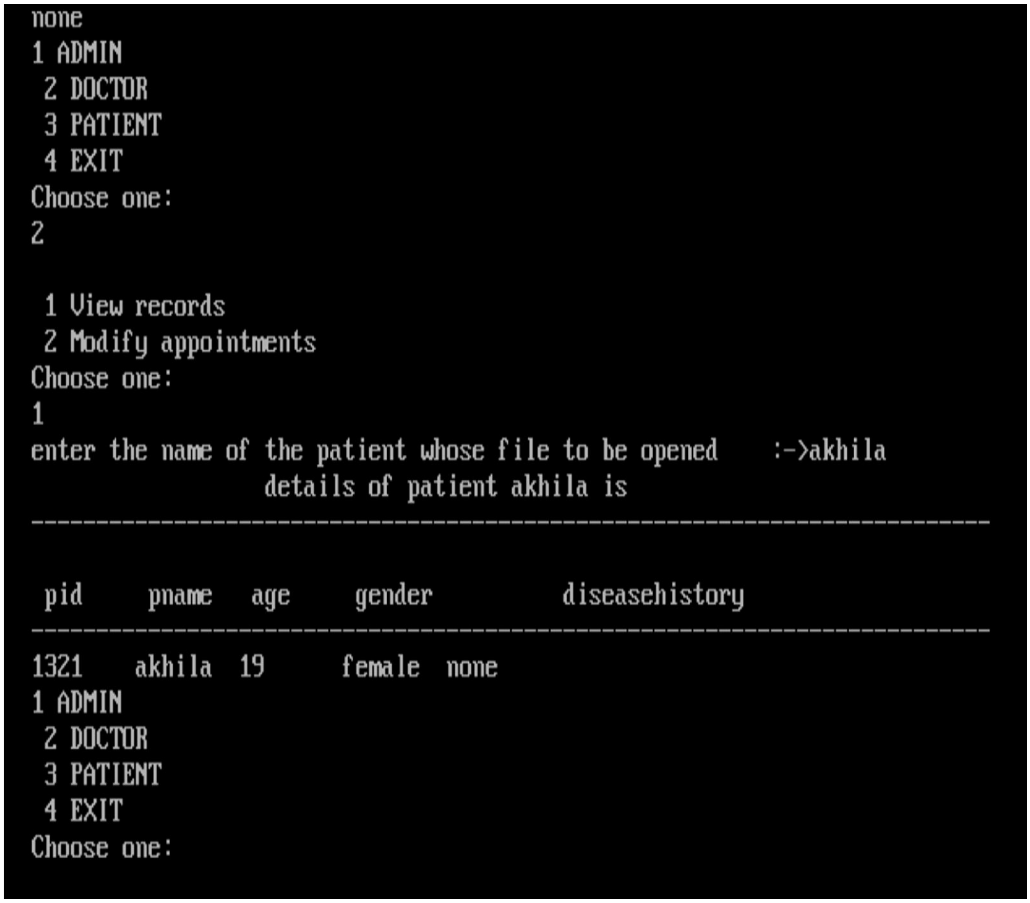


Figure 4.2.3 Doctor-to view patient records

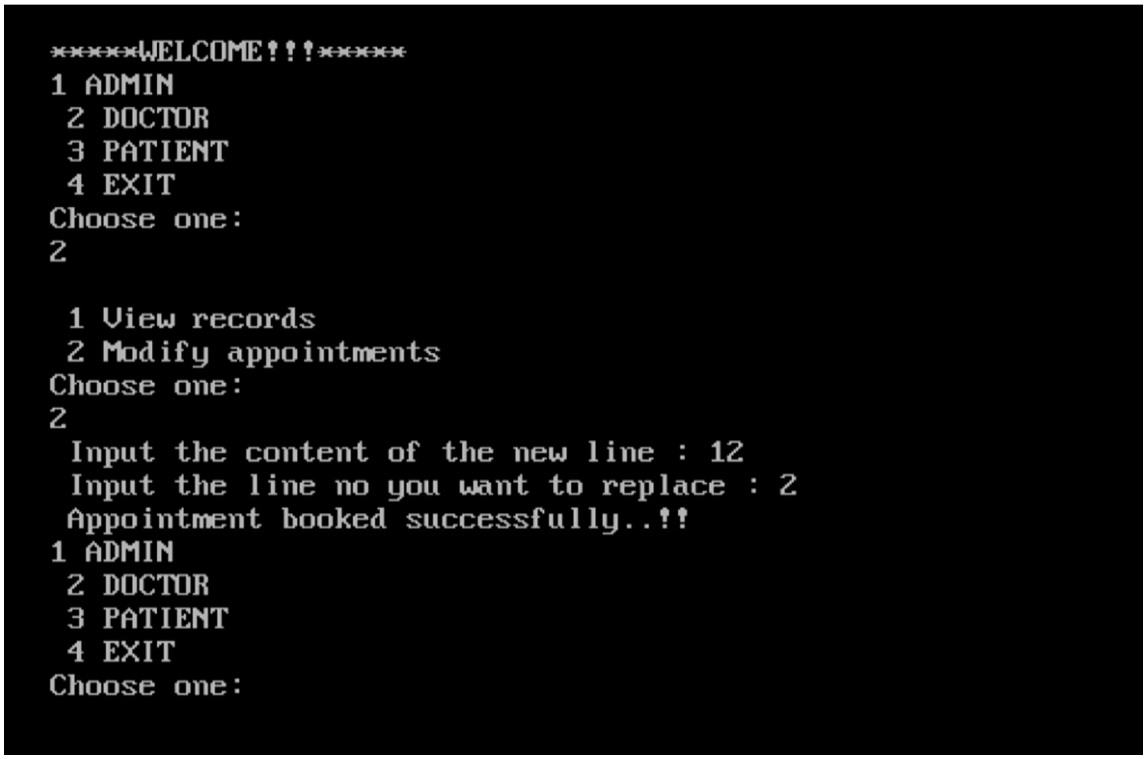


Figure 4.2.4 Doctor-to modify appointment availability

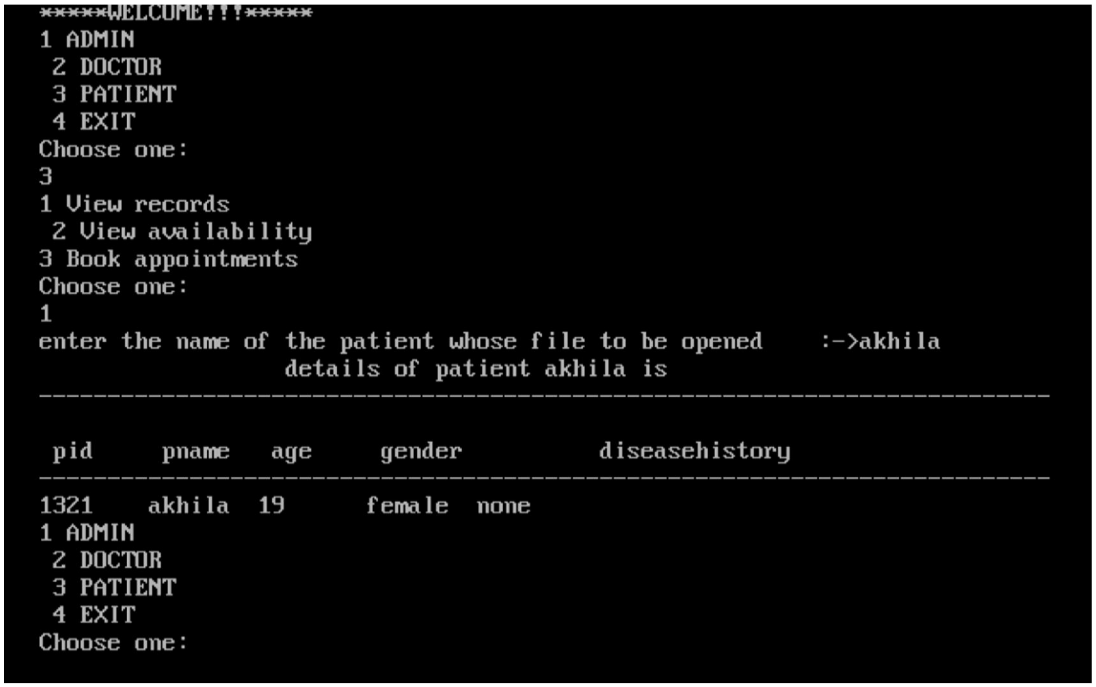


Figure 4.2.5 Patient-to view records

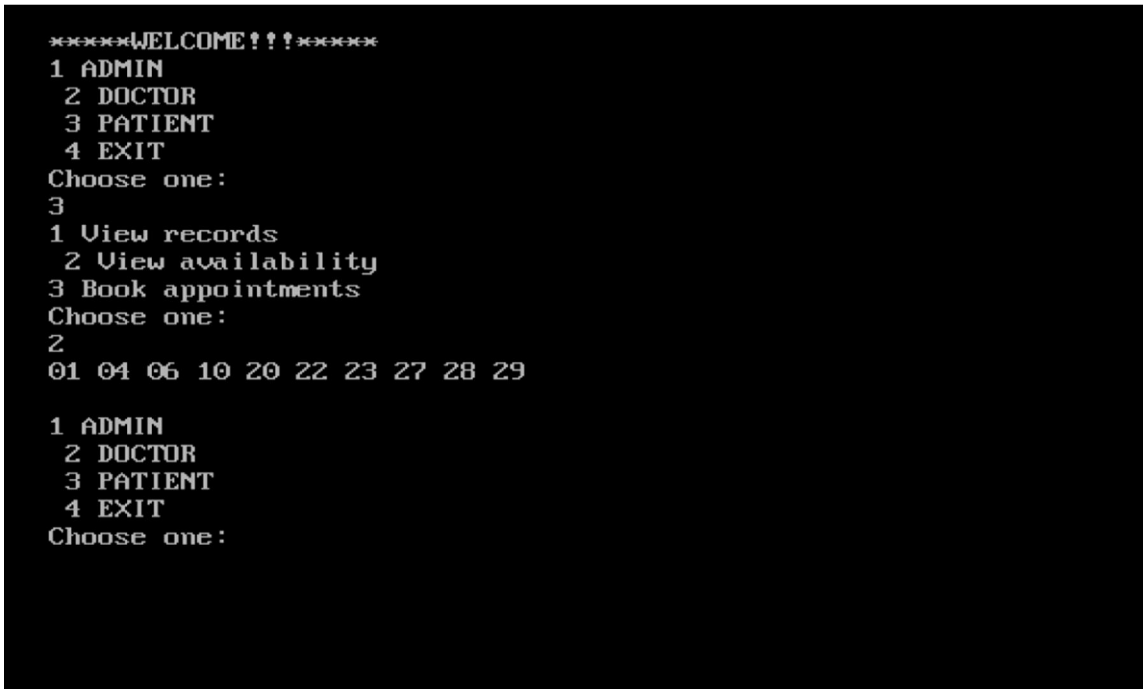


Figure 4.2.6 Patient-to view doctor's availability

```
*****WELCOME!!!*****
1 ADMIN
 2 DOCTOR
 3 PATIENT
 4 EXIT
Choose one:
3
1 View records
 2 View availability
3 Book appointments
Choose one:
3

 enter the date you want to book appointment on(in dd format)
12

 appointment not available1 ADMIN
 2 DOCTOR
 3 PATIENT
 4 EXIT
Choose one:
_
```

Figure 4.2.7 Patient-to book appointment with the doctor(unavailable)

```
-------------------------------------------------------------------------

 pid     pname   age     gender          diseasehistory
-------------------------------------------------------------------------
1321    akhila  19      female  none
1 ADMIN
 2 DOCTOR
 3 PATIENT
 4 EXIT
Choose one:
3
1 View records
 2 View availability
3 Book appointments
Choose one:
3

 enter the date you want to book appointment on(in dd format)
04
akhila has booked an appointment on 041 ADMIN
 2 DOCTOR
 3 PATIENT
 4 EXIT
Choose one:
_
```

Figure 4.2.8 Patient-to book an appointment(available)

CONCLUSION

Hence we can conclude that this project can widely be used in hospitals. It is also used by the patients as well as doctor. It removes the flaws in the existing system. This project requires the usage of database which is easily executed using the concept of file handling in C.

REFERENCES

The following books were very helpful during the completion of project:

1. The C Programming Language (Ansi C Version)

-Brain W Kernighan and Dennis M. Ritchie

2. C in Depth

-Deepali Srivatsava and S. K Srivatsava

3. Introduction to Programming in C

- Padma Reddy