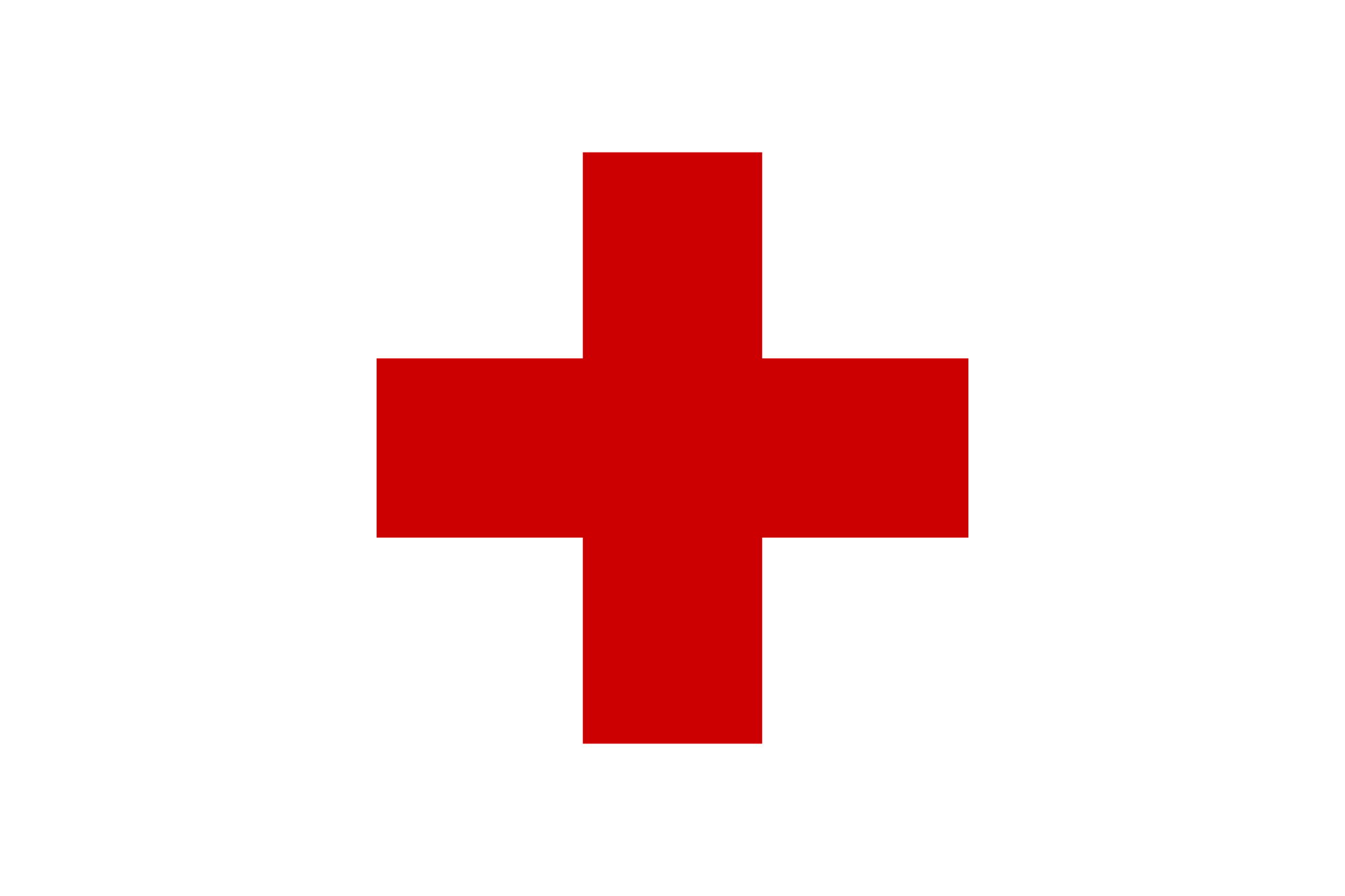
Computer Science Project

**CHECKUP MANAGER 1.0**



***By Nishant Baranwal Somy (Roll No. 12126) and Sagar Kowe (Roll No. 12130) of XII ‘A’, Batch – 2013-14***

Acknowledgement

First and foremost, we wish to extend our sincere thanks to our Physics Teacher Mr. Amit Dave for valuable guidance and instrument in the completion of our project satisfactorily.

I would also like to thank our Principal Mr. N.D Joshi for his constant encouragement, motivation and support.

Index

1. Overview
2. Logic
3. Code
4. Output

Overview

Checkup Manager 1.0 is a program for made for small clinics/hospitals for keeping of records of Doctors, patients and checkups performed.

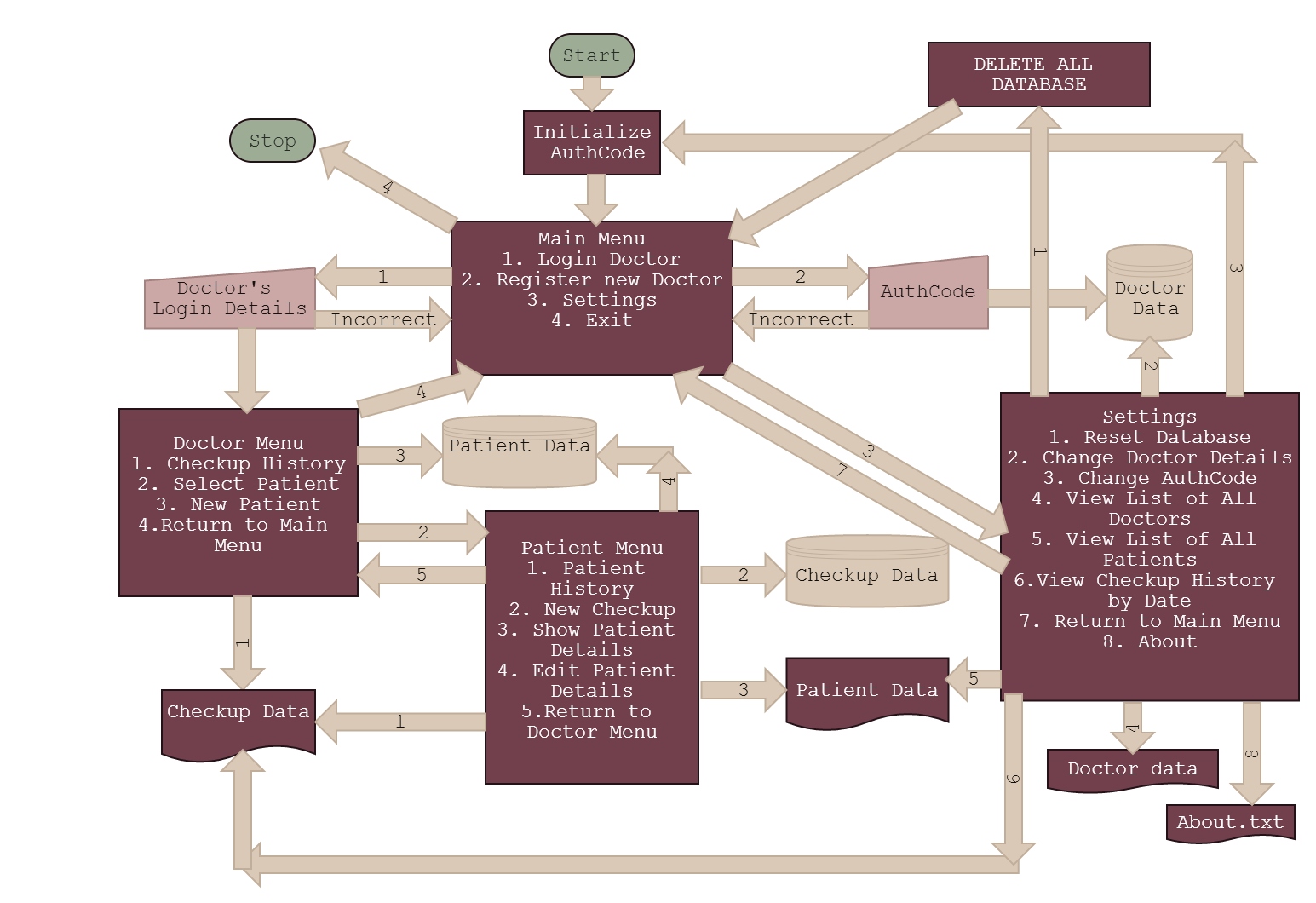
This program is based on simple implementation of basic data handling and uses the concepts of Object Oriented Programming.

Security has been given the highest priority, and every major change is made via a secure administrative code called AuthCode in the program.

As with most record keeping software, this program stores records in binary format (.dat) . Data can be retrieved based on multiple parameters (date, patient, doctor etc.). Records can’t be modified directly, enhancing the security of program.

Finally, the program allows modification of major authentication data (AuthCode, Doctor Passwords etc.) through settings. Database can be cleared in case of emergencies conveniently as well.

Logic



CODE

#include <fstream.h>

#include <conio.h>

#include <ctype.h>

#include <stdio.h>

#include <string.h>

#include <math.h>

#include <dos.h>

date enterdate()

{

datein:

date datep;

char td[11];

short unsigned int n;

gets(td);

if(td[10]!='\0')

{

cout<<'\a';

goto datein;

}

for(int i=0;i<10;i++)

if((i==2||i==5)&&td[i]=='/')

continue;

else if(isdigit(td[i]))

continue;

else

{

cout<<'\a';

goto datein;

}

n=(td[0]-48)\*10+td[1]-48;

if(n>=1&&n<=31)

datep.da\_day=n;

else

{

cout<<'\a';

goto datein;

}

n=(td[3]-48)\*10+td[4]-48;

if(n>=1&&n<=12)

datep.da\_mon=n;

else

{

cout<<'\a';

goto datein;

}

n=(td[6]-48)\*1000+(td[7]-48)\*100+(td[8]-48)\*10+td[9]-48;

if(n<=2013)

datep.da\_year=n;

else

{

cout<<'\a';

goto datein;

}

return datep;

}

unsigned long int enter6digit(int c=1) //0 to show digits

{

//cout<<" [ONLY DIGITS WILL BE ACCEPTED] : ";

char ch;

int i=0,temp;

long unsigned int no=0;

re:

for (; i<6; ++i)

{

ch=getch();

if (isdigit(ch)&&!(ch=='0'&&i==0))

{

if(c)

cout<<'\*';

else

cout<<ch;

temp = (int)ch-48;

no = no\*10 + temp;

}

else

goto re;

}

return no;

}

void passenter (char password[17])

{

char ch;

for (int i=0; i<16; i++)

{

top:

ch=getch();

if (ch=='\b')

{

if(!i)

goto top;

cout<<"\b";

i--;

goto top;

}

if (ch=='\r')

break;

password[i]=ch;

cout<<"\*";

}

password[i]='\0';

}

struct medicallist

{

char mediname[100];

int quantity;

};

class authcode

{

long unsigned int code;

char secq[400];

char seca[400];

public:

void changeauthcode(int c);

int checkauthcode();

}ac;

void authcode::changeauthcode(int c=1) //0 for first time

{

rein:

if(c)

{

clrscr();

gotoxy (25, 6); cout<<"Enter new AuthCode : ";

}

else

{

clrscr();

cout<<"\t\t\tWelcome to Checkup Manager 0.6 (Dev)"

<<"\n\nNote: This software's administrative rights are managed with \nthe help of a code called AuthCode. You will be asked \nfor AuthCode whenever making singnificant changes in the Database."

<<"\n\nPlease Initialize the the AuthCode and Security Question now."

<<"\n\nEnter a 6 digit AuthCode : ";

}

code=enter6digit(0);

cout<<"\n\nEnter Security Question : ";

gets (secq);

cout<<"\nEnter Securty Answer : ";

gets (seca);

clrscr();

cout<<"\n\n\t\t\t Confirmation "

<<"\nAuthCode : "<<code

<<"\nQuestion : "<<secq

<<"\nAnswer : "<<seca

<<"\n\nPress [TAB] to change. \nPress any key to continue . . .";

if (getch()=='\t')

{

clrscr();

goto rein;

}

}

int authcode::checkauthcode()

{

codein:

clrscr();

gotoxy(25, 6); cout<<"Enter Authcode : ";

if (code == enter6digit())

{

gotoxy(30, 8); cout<<"Code Matched";

return 1;

}

else

{

int i=3;

cout<<"\nCode Invalid!";

invalidcodein:

cout<<"\n\n1. Answer security questions."

<<"\n\n2. Try again ... ";

switch(getch())

{

case '1':

{

char str[400];

cout<<"\n\nPlease answer the following question : "<<endl;

cout<<"Q : "<<secq<<endl<<"A : ";

gets(str);

if (!strcmp(str,seca))

{

cout<<"\nSuccess."

<<"\n\nAuthCode : "

<<code;

cout<<"\nPress any key to continue ... ";

getch();

return 1;

}

break;

}

case '2':

{

clrscr();

gotoxy (25, 6); cout<<"Enter Authcode : ";

if (code == enter6digit())

{

gotoxy (30, 8); cout<<"Code Matched";

return 1;

}

break;

}

default :

{

cout<<"\a";

goto invalidcodein;

}

}

--i;

if (!i)

{

cout<<"\aIncorrect Answer 3 times.";

getch();

return 0;

}

cout<<"\n\n\aIncorrect answer. "<<i<<" Try(s) Remaining. . .";

getch();

goto invalidcodein;

}

}

class checkup

{

unsigned long int cardno,n;

medicallist medicine[20];

char doctorname[50];

date datep;

time timep;

char remark[200];

public:

date getdate\_of\_cu();

unsigned long int get\_cardno();

char\* get\_doctorname();

void do\_checkup(char dname[],unsigned long int newcn);

void show\_without\_date();

void show\_without\_cardno();

void show\_without\_doctor();

}cu;

date checkup::getdate\_of\_cu()

{

return datep;

}

unsigned long int checkup::get\_cardno()

{

return cardno;

}

char\* checkup::get\_doctorname()

{

return doctorname;

}

void checkup::do\_checkup(char dname[],unsigned long int newcn)

{

getdate (&datep);

gettime (&timep);

strcpy(doctorname,dname);

cardno=newcn;

int i=0, y=3,x=1;;

do

{

gotoxy(10,0);cout<<"Medicine "; gotoxy(30,0); cout<<"|";

gotoxy(0,2); cout<<"-----------------------------";gotoxy(30,2); cout<<"|";gotoxy(31,2);cout<<"-----------------------------";

gotoxy(40,0);cout<<"Quantity "; gotoxy(30,y); cout<<"|";

gotoxy(0,y); cout<<x<<". ";

x++;

gotoxy(4,y);gets(medicine[i].mediname);

gotoxy(44,y);cin>>medicine[i].quantity;

y++;

i++;

gotoxy(0, (y+2));cout<<" ";

gotoxy(0, (y+3));cout<<"( Press [ENTER] to continue. Press [TAB] to exit. )";

}while (getch()!='\t');

n=i;

cout<<"\nEnter Remarks : ";

gets(remark);

}

void checkup::show\_without\_date()

{

cout<<endl<<"----------------------------------------------------------------------------------------------------------------------------------------------------------------"<<endl;

printf("\nTime of Checkup : %2d:%02d:%02d\n", timep.ti\_hour, timep.ti\_min, timep.ti\_sec);

cout<<"\nCard no. : \t"<<cardno;

for (int i=0; i<n; i++)

cout<<"\nMedicine : \t"<< medicine[i].mediname<<" Quantity : "<< medicine[i].quantity;

cout<<"\nRemarks : \t"<<remark;

cout<<"\nDoctor : \t" << doctorname;

}

void checkup::show\_without\_cardno()

{

cout<<endl<<"----------------------------------------------------------------------------------------------------------------------------------------------------------------"<<endl;

printf("\nDate of Checkup: %02d/%02d/%04d\n", datep.da\_day, datep.da\_mon, datep.da\_year);

printf("\nTime of Checkup : %2d:%02d:%02d\n", timep.ti\_hour, timep.ti\_min, timep.ti\_sec);

for (int i=0; i<n; i++)

cout<<"\nMedicine : \t"<< medicine[i].mediname<<" Quantity : "<< medicine[i].quantity;

cout<<"\nRemarks : \t"<<remark;

cout<<"\nDoctor : \t" << doctorname;

}

void checkup::show\_without\_doctor()

{

cout<<endl<<"----------------------------------------------------------------------------------------------------------------------------------------------------------------"<<endl;

printf("\nDate of Checkup: %02d/%02d/%04d\n", datep.da\_day, datep.da\_mon, datep.da\_year);

printf("\nTime of Checkup : %2d:%02d:%02d\n", timep.ti\_hour, timep.ti\_min, timep.ti\_sec);

cout<<"\nCard no : \t"<<cardno;

for (int i=0; i<n; i++)

cout<<"\nMedicine : \t"<< medicine[i].mediname<<" Quantity : "<< medicine[i].quantity;

cout<<"\nRemarks : \t"<<remark;

}

int showcu(date tempdate)

{

int flag=0;

ifstream fch("PCH.dat", ios::in | ios::binary);

fch.seekg(0);

while (fch.read((char\*)&cu, sizeof(cu)))

{

if (cu.getdate\_of\_cu().da\_day == tempdate.da\_day && cu.getdate\_of\_cu().da\_mon == tempdate.da\_mon && cu.getdate\_of\_cu().da\_year == tempdate.da\_year)

{

cu.show\_without\_date();

flag=1;

cout<<endl;

}

}

fch.close();

if(flag)

return 1;

else

return 0;

}

int showcu(unsigned long int cardno)

{

int flag=0;

ifstream fch("PCH.dat", ios::in | ios::binary);

fch.seekg(0);

while (fch.read((char\*)&cu, sizeof(cu)))

{

if (cu.get\_cardno() == cardno)

{

cu.show\_without\_cardno();

flag=1;

cout<<endl;

}

}

fch.close();

if(flag)

return 1;

else

return 0;

}

int showcu(char doctorname[])

{

int flag=0;

ifstream fch("PCH.dat", ios::in | ios::binary);

fch.seekg(0);

while (fch.read((char\*)&cu, sizeof(cu)))

{

if (!strcmp(cu.get\_doctorname(),doctorname))

{

cu.show\_without\_doctor();

flag=1;

cout<<endl;

}

}

fch.close();

if(flag)

return 1;

else

return 0;

}

class patient

{

unsigned long int cardno;

int age;

date dob;

char bg[4];

char name[100];

char gender;

char category[100];

void enter\_card\_no();

void enter\_without\_cardno();

public:

unsigned long int get\_cardno();

void enter\_new\_patient();

void show\_patient();

void edit\_patient\_data();

}p;

int chkpat(unsigned long int cardno)

{

ifstream fp("PD.dat", ios::in | ios::binary);

fp.seekg(0);

while (fp.read((char\*)&p, sizeof(patient)))

if( p.get\_cardno() == cardno )

{

fp.close();

return 1;

}

fp.close();

return 0;

}

/\*void wrtpat(patient& newpat) //it doesn't work?

{

fstream fp("PD.dat", ios::in | ios::out | ios::binary);

fp.seekg(0);

long int pos=fp.tellg();

int flag=0;

while (fp.read((char\*)&p, sizeof(patient)))

{

if( p.get\_cardno() == newpat.get\_cardno() )

{

flag=1;

break;

}

pos=fp.tellg();

}

if(!flag)

{

fp.close();

fp.open("PD.dat", ios::app | ios::binary);

fp.seekp(0,ios::end);

}

else

{

fp.close();

fp.open("PD.dat", ios::app | ios::binary);

fp.seekp(pos);

}

fp.write((char\*)&newpat,sizeof(newpat));

fp.close();

}\*/

void wrtpat(patient& newpat,int flag=1) //0 for new patient

{

if(!flag)

{

ofstream fp("PD.dat", ios::app | ios::binary);

fp.seekp(0,ios::end);

fp.write((char\*)&newpat,sizeof(newpat));

fp.close();

}

else

{

ifstream fp1("PD.dat", ios::in | ios::binary);

ofstream fp2("temp.dat", ios::out | ios::binary);

fp1.seekg(0);

fp2.seekp(0);

while(fp1.read((char\*)&p,sizeof(p)))

{

if( p.get\_cardno() == newpat.get\_cardno() )

fp2.write((char\*)&newpat,sizeof(newpat));

else

fp2.write((char\*)&p,sizeof(p));

}

fp1.close();

fp2.close();

remove("PD.dat");

rename("temp.dat","PD.dat");

}

}

void patient::enter\_card\_no()

{

retry:

cout<<"\n\nEnter 6 digit Card Number : ";

cardno=enter6digit(0);

if(chkpat(cardno))

{

cout<<"Patient Already Exist. Redirecting . . . ";

getch();

goto retry;

}

cout<<"\n\nCARD NO. REGISTERED SUCCESSFULLY.";

}

void patient::enter\_without\_cardno()

{

char tempbg[4];

start:

clrscr();

gotoxy(20, 0);cout<<"Enter the following Patient Details - ";

gotoxy(20, 3);cout<<"Enter Name : ";

cin>>name;

gotoxy(20, 4);cout<<"Enter Age : ";

cin>>age;

gotoxy(20, 5);cout<<"Enter Gender : ";

cin>>gender;

gotoxy(20, 6);cout<<"Enter Date of Birth : ";

dob=enterdate();

bgs:

gotoxy(20, 7);cout<<"Enter Bloodgroup : ";

cin>>bg;

strcpy(tempbg, bg);

for (int i=0; i<4; i++)

bg[i]=toupper(bg[i]);

if(!(((bg[0]=='A'||bg[0]=='B'||bg[0]=='O')&&(bg[1]=='+'||bg[1]=='-')&&(bg[2]=='\0'))||(bg[0]=='A'&&bg[1]=='B'&&(bg[2]=='+'||bg[2]=='-')&&bg[3]=='\0')))

{

gotoxy(20, 7);goto bgs;

}

strcpy(bg, tempbg);

gotoxy(20, 8);cout<<"Enter Category : ";

gets(category);

}

unsigned long int patient::get\_cardno()

{

return cardno;

}

void patient::enter\_new\_patient()

{

enter\_card\_no();

enter\_without\_cardno();

}

void patient::show\_patient()

{

cout<<"\n\nPatient Details :"

<<"\nCard no. : "<<cardno

<<"\nName : "<<name;

printf("\nDate of Birth: %02d/%02d/%04d", dob.da\_day, dob.da\_mon, dob.da\_year);

cout<<"\nAge : "<<age

<<"\nBlood Group : "<<bg

<<"\nGender : "<<gender

<<"\nCategory : "<<category;

}

void patient::edit\_patient\_data()

{

show\_patient();

enter\_without\_cardno();

}

class doctor

{

char doctorname[100];

char password[17];

public:

void enter\_new\_doctor();

char\* get\_password();

char\* get\_doctorname();

void edit\_doctor();

}d;

int chkdoc(char doctorname[])

{

ifstream fd("DD.dat", ios::in | ios::binary);

fd.seekg(0);

while (fd.read ((char\*)&d, sizeof(doctor)))

{

if (!strcmp (d.get\_doctorname(), doctorname))

{

fd.close();

return 1;

}

}

fd.close();

return 0;

}

void wrtdoc(doctor& newdoc,int flag=1) //0 for new doctor 1 to edit doctor

{

if(!flag)

{

ofstream fd("DD.dat", ios::app | ios::binary);

fd.seekp(0,ios::end);

fd.write((char\*)&newdoc,sizeof(newdoc));

fd.close();

}

else

{

ifstream fd1("DD.dat", ios::in | ios::binary);

ofstream fd2("temp.dat", ios::out | ios::binary);

fd1.seekg(0);

fd2.seekp(0);

while(fd1.read((char\*)&d,sizeof(d)))

{

if(!strcmp(d.get\_doctorname(),newdoc.get\_doctorname()))

{

newdoc.edit\_doctor();

fd2.write((char\*)&newdoc,sizeof(newdoc));

}

else

fd2.write((char\*)&d,sizeof(d));

}

fd1.close();

fd2.close();

remove("DD.dat");

rename("temp.dat","DD.dat");

}

}

void doctor::enter\_new\_doctor()

{

char pass[17];

upar:

clrscr();

gotoxy(25,6); cout<<"Enter Doctor Name : Dr.";

gotoxy(49,6); gets(doctorname);

if(chkdoc(doctorname))

{

gotoxy(20,9); cout<<"Doctor Already Exists. \nPlease try again. . . ";

getch();

clrscr();

goto upar;

}

repass:

clrscr();

gotoxy(25,6); cout<<"Enter Doctor Name : Dr.";

gotoxy(49,6); cout<<doctorname;

gotoxy(25,7); cout<<"Enter Password : ";

gotoxy(44,7); passenter (password);

gotoxy(25,8); cout<<"Confirm Password : ";

gotoxy(44,8); passenter (pass);

if (strcmp (pass, password))

{

gotoxy(20, 10); cout<<"Passwords didn't match. Please Retry . . . ";

goto repass;

}

}

char\* doctor::get\_password()

{

return password;

}

char\* doctor::get\_doctorname()

{

return doctorname;

}

void doctor::edit\_doctor()

{

reenter:

char pass[17];

gotoxy(25,6); cout<<"Username : "<<doctorname;

gotoxy(25,7); cout<<"Password : ";

gotoxy(36,7); passenter(pass);

if (!strcmp (pass, password))

{

enter\_new\_doctor();

}

else

{

cout<<"Wrong Password. Press [TAB] to try again. \nPress any key to continue. . .";

if (getch()=='\t')

goto reenter;

}

}

void listalldoc()

{

clrscr();

int ctr=0;

ifstream fd("DD.dat", ios::in | ios::binary);

fd.seekg(0);

gotoxy(0,4);

while (fd.read((char\*)&d, sizeof(doctor)))

{

ctr++;

cout<<ctr<<")\t Dr. "<<d.get\_doctorname()<<endl;

}

if(!ctr)

cout<<"No doctor exists.";

cout<<"\n\nPress any key to continue...";

getch();

fd.close();

}

void init()

{

fstream fac("AC.dat", ios::nocreate | ios::in | ios::binary);

if (!fac)

{

fac.close();

ac.changeauthcode(0);

fac.open ("AC.dat", ios::out | ios::binary);

fac.write((char\*)&ac,sizeof(ac));

}

else

fac.read((char\*)&ac,sizeof(ac));

fac.close();

}

main()

{

restart:

init();

start:

clrscr();

cout<<"\t\t Checkup Manager 1.0 (Alpha)";

cout<<"\n\n\t\t--------- Main Menu ---------- \n\n\n\n\t\t1. Login doctor \n\n\t\t2. Register as New Doctor \n\n\t\t3. Settings \n\n\t\t4. Exit"<<endl<<endl;

switch(getch())

{

case '1':

{

start1:

clrscr();

doctor maind;

char doctorname[100],password[17];

gotoxy(25,6); cout<<"Username : ";

gotoxy(25,7); cout<<"Password : ";

gotoxy(36,6); gets(doctorname);

if(!chkdoc(doctorname))

{

reindnf:

gotoxy(23, 9); cout<<"Doctor not found. \n\n1. Try again \n2. Register as New Doctor \n3. Return to Main Menu";

switch(getch())

{

case '1': goto start1;

case '2': { clrscr(); goto regnewdoc; }

case '3': goto start;

default : goto reindnf;

}

}

gotoxy(25, 7); cout<<"Password : ";

gotoxy(36, 7); passenter(password);

maind=::d;

if(strcmp(maind.get\_password(),password))

{

cout<<"\n\n\aInvalid Password.\nPress any key to retry. . .";

getch();

goto start1;

}

cout<<"\n\n\t\t\tWelcome Dr. "<<maind.get\_doctorname();

getch();

docmen:

clrscr();

cout<<"\t\t\t\t\t\tLogged in as Dr. "<<doctorname<<"\n\n\t\t--------- Doctor's menu --------- \t \n\n\n\t\t1. See doctor history \n\n\t\t2. Select Patient \n\n\t\t3. View History by Date \n\n\t\t4. New Patient \n\n\t\t5. Return to Main Menu ";

switch (getch())

{

case '1':

{

clrscr();

gotoxy(30,2); cout<<"Showing Checkup records of Dr. "<<maind.get\_doctorname()<<endl;

if (!showcu(maind.get\_doctorname()))

{

cout<<"0 record found.\n\nPress any key to continue . . .";

getch();

}

else

{ cout<<endl<<"----------------------------------------------------------------------------------------------------------------------------------------------------------------"<<endl;

cout<<"\n\nPress any key to Return to Doctor Menu . . . ";

getch();

}

break;

}

case '2':

{

ic1:

patient mainp;

clrscr();

gotoxy(25,6); cout<<"Enter card number : ";

if(!chkpat(enter6digit(0)))

{

gotoxy(23,9); cout<<"Patient not found\n"

<<"Register as New Patient? [Y/N].";

if(toupper(getch())=='Y')

goto newp;

else

goto docmen;

}

mainp=::p;

pmen:

clrscr();

gotoxy(40,0); cout<<"Active Patient No : "<<mainp.get\_cardno();

cout<<"\n\n\t\t--------- Patient Menu --------- \n\n\n\t\t1. Patient's medical history \n\n\t\t2. New Checkup \n\n\t\t3. Show Patient Details \n\n\t\t4. Edit Patient details\n\n\t\t5. Return to doctor's menu";

switch (getch())

{

case '1':

{

clrscr();

mainp.show\_patient();

if (!showcu(mainp.get\_cardno()))

cout<<"0 record found";

else

{

cout<<endl<<"----------------------------------------------------------------------------------------------------------------------------------------------------------------"<<endl;

cout<<"\n\nPress any key to return to Patient Menu . . . ";

getch();

}

break;

}

case '2':

{

clrscr();

checkup maincu;

maincu.do\_checkup(maind.get\_doctorname(),mainp.get\_cardno());

ofstream fch("PCH.dat", ios::app | ios::binary);

fch.seekp(0, ios::end);

fch.write((char\*)&maincu, sizeof(checkup));

fch.close();

break;

}

case '3':

{

clrscr();

mainp.show\_patient();

cout<<"\n\n\nPress any key to continue. . .";

getch();

break;

}

case '4':

{

clrscr();

mainp.edit\_patient\_data();

wrtpat(mainp);

break;

}

case '5':

{

clrscr();

goto docmen;

}

default:

{

cout<<"\a";

break;

}

}

goto pmen;

}

case '3':

{

if (!showcu(enterdate()))

cout<<"0 record found";

break;

}

case '4':

{

newp:

patient mainp;

mainp.enter\_new\_patient();

wrtpat(mainp,0);

cout<<"\nPatient entered successfully.\n\nPress any key to return to doctor menu . . . ";

getch();

break;

}

case '5':

{

goto start;

}

default :

{

cout<<"\a";

break;

}

}

goto docmen;

}

case '2':

{

regnewdoc:

clrscr();

if(ac.checkauthcode())

{

doctor maind;

maind.enter\_new\_doctor();

wrtdoc(maind,0);

gotoxy(20, 11); cout<<"You have been successfully registered";

getch();

break;

}

else

{

cout<<"\nIncorrect AuthCode. . . . ";

cout<<"\nRegistration unsuccessful. \n\n1. Try again \n2. Return to Main Menu";

switch(getch())

{

case '1': goto regnewdoc;

case '2': goto start;

}

}

}

case '3':

{

settings:

clrscr();

cout<<"\n\n\t\t--------- Settings ---------";

cout<<"\n\n\n\t\t1. Reset Database \n\n\t\t2. Change Doctor Details \n\n\t\t3. Change AuthCode \n\n\t\t4. View List of All Doctors \n\n\t\t5. Return to Main Menu \n\n\t\t6. About CheckUp Manager 1.0";

switch (getch())

{

case '1':

{

if (ac.checkauthcode())

{

re1:

clrscr();

cout<<"\n\n\tResetting Database will PERMANENTLY DELETE ALL RECORDS STORED. \n\n\t\t\tAre you sure? [Y/N] : ";

switch(getch())

{

case 'y':

{

remove ("AC.dat");

remove ("PD.dat");

remove ("DD.dat");

remove ("PCH.dat");

cout<<"\n\n\tDATABASE CLEANED. ALL FILES DELETED. PRESS ANY KEY TO RESTART . . .";

getch();

goto restart;

}

case 'Y':

{

remove ("AC.dat");

remove ("PD.dat");

remove ("DD.dat");

remove ("PCH.dat");

cout<<"\n\n\tDATABASE CLEANED. ALL FILES DELETED. PRESS ANY KEY TO RESTART . . .";

getch();

goto restart;

}

case 'n':

{

cout<<"Redirecting....";

getch();

break;

}

case 'N':

{

cout<<"Redirecting....";

getch();

break;

}

default : goto re1;

}

}

else

cout<<"Incorrect authcode. Redirecting...";

break;

}

case '2':

{

if (ac.checkauthcode())

{

char doctorname[100];

clrscr();

gotoxy (25, 6); cout<<"Username : ";

gotoxy (36, 6); gets(doctorname);

if (chkdoc(doctorname))

{

doctor maind=::d;

wrtdoc(maind);

}

else

{

cout<<"\n\nDoctor not found. Redirecting...";

getch();

}

}

else

{

cout<<"\n\nIncorrect authcode. Redirecting...";

getch();

}

break;

}

case '3':

{

if (ac.checkauthcode())

{

ac.changeauthcode();

ofstream fac("AC.dat", ios::out | ios::binary);

fac.write((char\*)&ac,sizeof(ac));

fac.close();

}

else

cout<<"Incorrect authcode. Redirecting...";

break;

}

case '4':

{

if (ac.checkauthcode())

listalldoc();

else

cout<<"Incorrect authcode. Redirecting...";

break;

}

case '5':

{

goto start;

}

case '6':

{

clrscr();

char reader;

ifstream fa("About.txt", ios::in);

while (fa.read((char\*)&reader, sizeof(reader)))

cout<<reader;

cout<<"\n\n\nPress any key to return to Settings . . . ";

getch();

break;

}

default :

{

cout<<"\a";

break;

}

}

goto settings;

}

case '4':

{

clrscr();

gotoxy(20, 4); cout<<"Thank you for using Checkup Manager 1.0";

return 1;

}

default:

{

cout<<"\a";

break;

}

}

goto start;

}

Output Screenshots

