

hospital management system

Data structure



January 15, 2021

M sameer sohail

15000

**Hospital management system**

**About:**

* concept of recording patient records and their diagnosis information
* . Before stepping into the main system, a user must pass through a login system to get access, then only the user can add a new patient record, diagnosis information and check the full history of the patient.
* Features:
* Add Patient Record
* View Patient History

**Abstract:**

* The purpose and essence of any Records Management system is the right information in the right place in the right order, at the right time for the right person at the lowest cost.
* Hospital management system is a computerized system designed and programmed to deal with day-to-day operations taking place.
* The program can look after inpatients, outpatients, records, database treatments, status illness, billings in the pharmacy and labs.
* It also maintains hospital information such as ward id, doctors in charge and department administering.

**Purpose:**

* The purpose of the project is to computerize the Front Office Management of Hospital to develop software which is user friendly, simple, fast, and cost – effective. It deals with the collection of patient’s information, diagnosis details, etc. Traditionally, it was done manually. The project outlines all the process followed to produce the software that is from analysis to testing the system.
* Hospitals deal with the life and health of their patients.
* Good medical care relies on well- trained doctors and nurses and on high quality facilities and equipment. Good medical care also relies on good record keeping. Without accurate, comprehensive, and up to date and accessible patient notes, medical personnel may not offer

**Code:**

#include<iostream>

#include<conio.h>

#include<process.h>

using namespace std;

class all

{

private:

struct address

{

int house;

char street[30];

char city[30];

char state[30];

char country[30];

};

struct age

{

int day;

int month;

int year;

};

struct patient\_info

{

char name[50];

address AD1; //nested structure implemented

age A1; //nested structure inplemented

int martial\_status;

int reg\_no;

int bld\_group;

int sex;

}PI[100];

int task;

protected:

void enter\_patient\_info();

void show\_patient\_detail();

public:

void software\_detail();

void tasks();

char answer;

char answer1;

char ch;

int serial;

};

class date

{

private:

int date;

int month;

int year;

public:

void enter\_date();

void show\_date();

};

class dob

{

private:

struct dob1

{

int date;

int month;

int year;

int rem;

}DOB11[100];

public:

void enter\_date();

void show\_date();

};

int i = 0;

int rem;

int regis;

int attempt;

int temp;

int show\_count = 0;

all A1; //object declared

date D1; //object declared

dob DOB1; //object declared

void main()

{

int count = 0;

cout << endl ;

cout << endl;

cout << endl;

cout << "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" << endl;

cout << "\*\*\*\*\*\*\*\*\*Welcome to\*\*\*\*\*\*\*\*\*\*\*\*" << endl;

cout << "\*\*HOSPITAL MANAGEMENT SOFTWARE\*\*\*" << endl;

cout << "\*\*\*\*\*\*By Sameer sohail\*\*\*\*\*\*" << endl;

cout << endl;

cout << endl;

cout << endl;

D1.enter\_date();

A1.tasks();

}

void all::tasks()

{

attempt = 0;

D1.show\_date();

cout << endl;

cout << endl;

cout << "\*\*HOSPITAL MANAGEMENT SOFTWARE \* \*\*" << endl;

cout << "By sameer sohail " << endl;

cout << "\* \*Hospital Management Tasks \* \*" << endl;

cout << "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" << endl;

cout << " Please select a task to do...." << endl;

cout << "1. Enter a new patient information ." << endl;

cout << "2. View detail of existing patient ." << endl;

cout << "3. View detail about the program ." << endl;

cout << "4. Exit from the program ." << endl;

cout << endl;

cout << endl;

cout << endl;

//other function remain

cout << " Enter your task serail : " << endl

;

cin >> task;

switch (task)

{

case 1: {

A1.enter\_patient\_info();

break;

}

case 2: {

A1.show\_patient\_detail();

break;

}

case 3: {

A1.software\_detail();

break;

}

case 4: {

cout << "Thank You for trying this program !!!" << endl;

cout << "This is the end of program...." << endl;

cout << "Press any key to exit....." << endl;

//getch();

exit(0);

break;

}

default: {

cout << "Invalid task serial ." << endl;

cout << "Press any key to continue...." << endl;

//getch();

A1.tasks();

}

}

}

void all::enter\_patient\_info()

{

int count = 0;

answer = 'y';

if (count == 0)

{

serial = 1;

}

else

{

i = serial;

}

for (i = serial; answer == 'y' || answer == 'Y'; i++)

{

PI[i].reg\_no = i;

temp = serial;

cout << endl;

cout << "\*\*ENTERING INFORMATION FOR PATIENT SERIAL NUMBER " << i << endl;

cin.get(ch);

cout << "Registration Number : " << PI[i].reg\_no;

cout << endl;

cout << "Enter the name of patient :" << endl;

cin.getline(PI[i].name, 50);

cout << "Sex (1-Male 2-Female) :" << endl;

cin >> PI[i].sex;

while (PI[i].sex != 1 && PI[i].sex != 2)

{

cout << "Invalid input for sex of patient!!!" << " ";

cout << "Sex :" << " ";

cin >> PI[i].sex;

}

cout << " \*\*ENTERING ADDRESS \* \*" << endl;

cout << "House number :" ;

cin >> PI[i].AD1.house;

while (PI[i].AD1.house <= 0)

{

cout << "Invalid input for house number ." << endl;

cout << "Again enter the house number ." << endl;

cin >> PI[i].AD1.house;

}

cin.get(ch);

cout << "Street :" << " ";

cin.getline(PI[i].AD1.street, 30);

cout << "City :" << " ";

cin.getline(PI[i].AD1.city, 30);

cout << "State :" << " ";

cin.getline(PI[i].AD1.state, 30);

cout << "Country :" << " ";

cin.getline(PI[i].AD1.country, 30);

DOB1.enter\_date();

//to calculate age

cin.get(ch);

cout << "Martial status(1-Married,2-Not Married ):" << " ";

if (count != 0)

{

}

cin >> PI[i].martial\_status;

while (PI[i].martial\_status < 1 || PI[i].martial\_status>2)

{

cout << "Invalid input for martial status ." << "";

cout << "Enter a valid martial status :" << " ";

cin >> PI[i].martial\_status;

}

cin.get(ch);

if (count != 0)

{

}

cout << "Blood group :" << "";

cout << "1. A+ " << " ";

cout << "2. A- " << " ";

cout << "3. B+ " << " ";

cout << "4. B- " << " ";

cout << "5. AB+ " << " ";

cout << "6. AB- " << " ";

cout << "7. O+ " << " ";

cout << "8. O- " << " ";

cout << "Enter :" << " ";

cin >> PI[i].bld\_group;

switch (PI[i].bld\_group)

{

case 1:

case 2:

case 3:

case 4:

case 5:

case 6:

case 7:

case 8:

{

break;

}

default: {

while (PI[i].bld\_group != 1 && PI[i].bld\_group != 2 && PI[i].bld\_group != 3 &&

PI[i].bld\_group != 4 && PI[i].bld\_group != 5 && PI[i].bld\_group != 6 &&

PI[i].bld\_group != 7 && PI[i].bld\_group != 8)

{

cout << "Invalid input !" << " ";

cout << "Blood Group :" << " ";

cin >> PI[i].bld\_group;

}

break;

}

}

cin.get(ch);

cout << " Want to enter information for another patient ? " << " ";

cin >> answer

;

count++;

serial++;

}

A1.tasks();

}

void dob::enter\_date()

{

int count = 0;

cout << "Date of birth" << " ";

cout << "Year :";

cin >> DOB11[temp].year;

if (DOB11[temp].year <= 0 || DOB11[temp].year > 10000)

{

do

{

cout << "Invalid input for year !" << "";

cout << "Please enter the year correctly :" << " ";

cin >> DOB11[temp].year;

} while (DOB11[temp].year < 0 || DOB11[temp].year>10000);

}

cout << "Month :";

cin >> DOB11[temp].month;

if (DOB11[temp].month <= 0 || DOB11[temp].month > 12)

{

do

{

cout << "Invalid input for month !" << " ";

cout << "Again enter the month :" << " ";

system("cls");

if (count != 0)

{

}

cin >> DOB11[temp].month;

} while (DOB11[temp].month < 0 || DOB11[temp].month>12);

}

cout << "Date :";

switch (DOB11[temp].month)

{

case 1:

case 3:

case 5:

case 7:

case 8:

case 10:

case 12: {

cin >> DOB11[temp].date;

while (DOB11[temp].date < 1 || DOB11[temp].date>31)

{

cout << "Invalid date !" << " ";

cout << "Again enter the date :" << " ";

cin >> DOB11[temp].date;

}

break;

}

case 2: {

cin >> DOB11[temp].date;

if (DOB11[temp].year % 4 == 0)

{

while (DOB11[temp].date < 0 || DOB11[temp].date>29)

//for leap year

{

cout << "Invalid date !" << " ";

cout << "Again enter the date :" << " ";

cin >> DOB11[temp].date;

}

}

else

{

while (DOB11[temp].date < 0 || DOB11[temp].date>28)

//for non-leap year

{

system("cls");

cout << "Invalid date !" << " ";

cout << "Again enter the date :" << " ";

cin >> DOB11[temp].date;

}

}

break;

}

default: {

cin >> DOB11[temp].date;

while (DOB11[temp].date < 1 || DOB11[temp].date>30)

{

cout << "Invalid date !" << " ";

cout << "Again enter the date :" << " ";

cin >> DOB11[temp].date;

}

break;

}

} //end of switch

}

void date::enter\_date()

{

cout << "First of all I need the current date ..." << endl;

cout << "Year :";

cin >> year;

if (year <= 0 || year > 10000)

{

do

{

cout << "Invalid input for year !" << endl;

cout << "Please enter the year correctly :" << endl;

cin >> year;

} while (year < 0 || year>10000);

}

cout << "Month :";

cin >> month;

if (month <= 0 || month > 12)

{

do

{

cout << "Invalid input for month !" << endl;

cout << "Again enter the month :" << endl;

cin >> month;

} while (month < 0 || month>12);

}

cout << "Date :";

switch (month)

{

case 1:

case 3:

case 5:

case 7:

case 8:

case 10:

case 12: {

cin >> date;

while (date < 1 || date>31)

{

cout << "Invalid date !" << endl;

cout << "Again enter the date :" << endl;

cin >> date;

}

break;

}

case 2: {

cin >> date;

if (year % 4 == 0)

{

while (date < 0 || date>29) //for leap year

{

cout << "Invalid date !" << endl;

cout << "Again enter the date :" << endl;

cin >> date;

}

}

else

{

while (date < 0 || date>28) //for non-leap year

{

cout << "Invalid date !" << endl;

cout << "Again enter the date :" << endl;

cin >> date;

}

}

break;

}

default: {

cin >> date;

while (date < 1 || date>30)

{

cout << "Invalid date !" << endl;

cout << "Again enter the date :" << endl;

cin >> date;

}

break;

}

} //end of switch

}

void date::show\_date() //remove the goto ststements in this function

{

cout << "Hello....It's ";

cout << date;

rem = date % 10;

switch (date)

{

case 11:

case 12:

case 13:

case 14:

case 15:

case 16:

case 17:

case 18:

case 19:

case 20: {

cout << "th ";

goto over;

}

}

switch (rem)

{

case 1: {

cout << "st ";

break;

}

case 2: {

cout << "nd ";

break;

}

case 3: {

cout << "rd ";

break;

}

default: {

cout << "th ";

break;

}

}

over:

switch (month)

{

case 1: {

cout << "January , ";

break;

}

case 2: {

cout << "February , ";

break;

}

case 3: {

cout << "March , ";

break;

}

case 4: {

cout << "April , ";

break;

}

case 5: {

cout << "May , ";

break;

}

case 6: {

cout << "June , ";

break;

}

case 7: {

cout << "July , ";

break;

}

case 8: {

cout << "August , ";

break;

}

case 9: {

cout << "September , ";

break;

}

case 10: {

cout << "October , ";

break;

}

case 11: {

cout << "November , ";

break;

}

case 12: {

cout << "December , ";

break;

}

}

cout << year << " ";

}

void all::show\_patient\_detail()

{

do

{

cout << "Enter registration number : " << "";

//system("cls");

cin >> regis;

cin.get(ch);

show\_count++;

if (regis > 0 & regis < serial)

{

//system("cls");

cout << " \* \*\*INFORMATION FOR PATIENT REGISTRATION NUMBER" << regis << " \* \*\*";

//system("cls");

cout << "Name : " << PI[regis].name << " ";

//system("cls");

cout << "Sex : ";

//system("cls");

if (PI[regis].sex == 1)

{

cout << "Male " << " ";

//system("cls");

}

if (PI[regis].sex == 2)

{

cout << "Female " << " ";

//system("cls");

}

cout << "Blood Group : ";

//system("cls");

switch (PI[regis].bld\_group)

{

case 1: {

//system("cls");

cout << "A+";

break;

}

case 2: {

//system("cls");

cout << "A-";

break;

}

case 3: {

//system("cls");

cout << "B+";

break;

}

case 4: {

//system("cls");

cout << "B-";

break;

}

case 5: {

//system("cls");

cout << "AB+ ";

break;

}

case 6: {

//system("cls");

cout << "AB-";

break;

}

case 7: {

//system("cls");

cout << "O+ ";

break;

}

case 8: {

//system("cls");

cout << "O-";

break;

}

}

//system("cls");

cout << "Date of birth : ";

//system("cls");

DOB1.show\_date();

cout << "Martial Status : ";

//system("cls");

if (PI[i].martial\_status == 1)

{

cout << "Married " << " ";

//system("cls");

}

else

{

cout << "Not married " << "";

//system("cls");

}

//system("cls");

cout << "\* \*ADDRESS \* \*" << "";

//system("cls");

cout << " House no. : " << PI[regis].AD1.house;

//system("cls");

cout << " Street : " << PI[regis].AD1.street;

//system("cls");

cout << "City : " << PI[regis].AD1.city;

//system("cls");

cout << "State : " << PI[regis].AD1.state;

//system("cls");

cout << " Country : " << PI[regis].AD1.country;

}

else

{

if (regis == 1)

{

cout << " Database is empty !!!" << endl;

cout << "Press any key to exit to main task menu..." << endl;

//getch();

A1.tasks();

}

attempt++;

if (attempt == 3)

{

cout << "You have entered wrong registration number 3 ." << endl;

cout << "Access Denied!!! " << endl;

cout << "Please try again later. " <<endl;

cout << "Press any key to exit to main task menu..." << endl

;

//getch();

A1.tasks();

}

cout << " Sorry, the registration number is invalid ." << " ";

cout << "Press any key to continue...." << " ";

//getch();

A1.show\_patient\_detail();

}

cout << " Want to see information of another patient : " << "";

cin >> answer1;

} while (answer1 == 'y' || answer1 == 'Y');

A1.tasks();

}

void dob::show\_date()

{

cout << DOB11[regis].date;

rem = DOB11[regis].date % 10;

switch (DOB11[regis].date)

{

case 11:

case 12:

case 13:

case 14:

case 15:

case 16:

case 17:

case 18:

case 19:

case 20: {

cout << "th ";

goto over;

}

}

switch (rem)

{

case 1: {

cout << "st ";

break;

}

case 2: {

cout << "nd ";

break;

}

case 3: {

cout << "rd ";

break;

}

default: {

cout << "th ";

break;

}

}

over:

switch (DOB11[regis].month)

{

case 1: {

cout << "January , ";

break;

}

case 2: {

cout << "February , ";

break;

}

case 3: {

cout << "March , ";

break;

}

case 4: {

cout << "April , ";

break;

}

case 5: {

cout << "May , ";

break;

}

case 6: {

cout << "June , ";

break;

}

case 7: {

cout << "July , ";

break;

}

case 8: {

cout << "August , ";

break;

}

case 9: {

cout << "September , ";

break;

}

case 10: {

cout << "October , ";

break;

}

case 11: {

cout << "November , ";

break;

}

case 12: {

cout << "December , ";

break;

}

}

}

void all::software\_detail()

{

cout << "\*\*\*\*SOFTWARE DETAILS\*\*\*\*";

cout << " Developer :muhammad sameer sohail " << endl;

cout << "Programming Language : C++ " << endl;

cout << " Press any key to return to the main task menu......." << endl;

//getch();

A1.tasks();

}

**Output**

**Text

Description automatically generated**

Text

Description automatically generated