# Hospital Management System

**Project Report**

## Created by:

## Saad Ahmed Jamal

**End Semester Project**

**Introduction**

Hospital Management System

**Problem Statement**

Making an automated application for patient’s entering the hospital that provides complete guidance

**Requirements**

Implementing different data structures in project:

* Trees
* Linked List
* Array

Providing an interface for implementing a Hospital.

Including File Handling for saving data.

**Time Complexity**

For Linked List O(n)

For trees O(nlogn)

Hence, Hospital Management System Time Complexity is O(n).

Where n is the number of input i.e. number of patients entering the hospital.

**Procedure**

First we enter the hospital, we are in a node if we are in a state of emergency, then we go directly without any queiries by pressing e, by pressing m you can enter into manager mode,by pressing any other key you enter normal paitent mode, now we enter 2nd node where your name, cnic are asked and your disease once done, you are guided to doctor corresponding to your disease. Then you enter 3rd node for doctor, then you seek medicine by entering 4th node, moving onto 5th node you are asked if you want to play a game which results in decreasing of bill if you score good. In the last node you are told your bill.

All the information is saved using file handling in manager mode in the end.

**Code**

#include <string>

#include <iostream>

#include <limits>

#include <stdio.h>

#include <stdlib.h>

#include <conio.h>

#include<windows.h>

#include<unistd.h>

#include <fstream>

using namespace std;

int i=0;

char ent ;

int paid[10];

string dis;

int j=0;

int k=0;

void medicineplz();

//"Snake Func."

//"Node for Linked List "

struct nod

{int srno;

string name;

int cnc;

int bill;

nod \*next;

};

class linkedlist

{ nod \*head;

nod \*current;

public:

linkedlist()

{

head=new nod();

current=head;

}

void newnod(int cni,string nam)

{ j++; nod \*newnod=new nod(); //cout<<"cool";

current->next=newnod; //cout<<"cooler";

newnod->cnc=cni;

newnod->name=nam; newnod->srno=j;

newnod->bill=0; current=newnod; }

void setbill(int bil)

{ current->bill=bil; }

void display()

{current=head;

current=current->next;

cout<<endl<<"\n\n-The List of patients that visited today are: \n\n";

cout<<"Srno. Patient's Name CNIC Bill\n";

while(current!=NULL)

{cout<<endl<<current->srno<<" "<<current->name<<" "<<current->cnc<<" "<<current->bill;

current=current->next; }

//"File Handling"

string re;

cout<<"\n\nDo you want to save this Important List Permanently??\n";

cin>>re;

if(re=="yes")

{ ofstream myfile;

myfile.open ("HospitalSafe.txt");

current=head; current=current->next;

myfile<<endl<<"\n\n-The List of patients that visited today are: \n\n";

myfile<<"Srno. Patient's Name CNIC Bill\n";

while(current!=NULL)

{myfile<<endl<<current->srno<<" "<<current->name<<" "<<current->cnc<<" "<<current->bill;

current=current->next;

}

myfile.close();

cout<<"Save Successful"; }

else

{ cout<<"OK";}

cout<<endl<<endl; }};

void manager();

linkedlist patientlist;

string name[10];

//"Node for Tree"

struct node

{ int cnic[10]; int data;

struct node\* left;

struct node\* right; void hos()

{ i++;

if(i==1)

cout<<"Welcome to our Hospital";

A:

cout<<"\n\n (use 'e' for emergency and 'n' for normal patient or 'm' to enter manager mode,'q' to exit the program \n";

cout<<"\n\n\n Are you in a state of emergency-1000 or you are a normal patient ";

cin>>ent;

if(ent=='n')

{ }

if(ent=='e')

{ cout<<"we are equally sorrow in this grieve moment ";

patientlist.newnod(0,"emergency");

}

if(ent=='m')

{

manager();

} }

if(i==2)

{

B:

cout<<"\n Enter your CNIC no.(without dashes) ";

cin>>cnic[k];

if (cnic[k]>99999 || cnic[k]<10000)

{ cout<<"\nInvalid CNIC ! Try again\n";

system("pause");

goto B; }

cout<<"\n Enter your cool name "<<endl; cin>>name[k];

cout<<"cool"; patientlist.newnod(cnic[k],name[k]);}

if(i==3)

{ cout<<"\n\nWhich type of disease you are sufferring from : "<<"\nfever - 400rs or heart- 500rs or lungs-500rs or unknown-700rs \n\n" ;

cin.clear();

fflush(stdin); cin>>dis;

cout<<"\nWaiting for doctor to respond\n";

int cc=1,bs=9;

for(int ca=0;ca<9;ca++)

{

for(int ba=0;ba<bs;ba++)

{ cout<<" "; }

for(int cb=0;cb<cc;cb++)

{

cout<<"\*"; }

cc=cc+2;

bs--;

cout<<endl;

Sleep(1000); }

int ti;

ti=rand()%10;

ti=ti\*5000;

Sleep(ti);

cout<<"n\nYes, The required doctor is available now \n\n";

if (dis=="fever")

{ cout<<"\nPlz proceed towards room no. 1 \n";

paid[k]=paid[k]+400; }

else if (dis=="heart")

{ cout<<"\nProceed towards room no.2 to meet our world best~! heart speacialist ";

paid[k]=paid[k]+500; }

else if (dis=="lungs")

{ cout<<"\n PRoceed towards room 3 our pulmo expert doctor is sitting thier ";

paid[k]=paid[k]+500; }

else{

cout<<"\nProceed towards room .4 to meet our Haqeem sahab ";

paid[k]=paid[k]+700;}

cout<<endl;

system("pause");}

if(i==4)

{string med;

cout<<"\n\n\n The medicines are available at discount/lower rate from the market";

cout<<"\n\nDo you want medicine also from hospital ?";

cin>>med;

if(med=="yes")

{

medicineplz();

}

else

{

cout<<"\n As you wish dear patient "; } }

if(i==5){ char gam;

cout<<"\n\nPlay a game and win cash prize nOw!!!"<<endl<<"Press Y for yes or anyother key to continue..";

cin>>gam;

if(gam=='Y')

{int cash;

run();

cash= score\*10;

cout<<"Congratulations, you have earned "<<cash<<"\n Its yours Enjoy!";

paid[k]=paid[k]-cash;

Sleep(2000);

}

else {

cout<<"\nOkay , No problem! You can continue to the counter \n";} }

if(i==6)

{ cout<<"\n\nSir Your total bill is "<<paid[k];

patientlist.setbill(paid[k]);

cin.ignore();

cout<<"\n\nThank you for visiting us, God Bless you";

cout<<"\n\nFor any Quiries, Contact:"<<endl<<"Saad Ahmed, Noman Abbasi "<<endl<<"0340-8157349";

system("pause");

}

};

/\* Helper function that allocates a new node \*/

struct node\* newNode(int data)

{

struct node\* node = (struct node\*)

malloc(sizeof(struct node));

node->data = data;

node->left = NULL;

node->right = NULL;

return(node);

}

void PostorderTraversal( node\* node)

{

if (node == NULL)

{Z: return;}

PostorderTraversal(node->left);

PostorderTraversal(node->right);

if(ent=='e'||ent=='m')

{

goto Z;

}

printf("%d ", node->data);

node->hos();

}

int main()

{

struct node \*root = newNode(1);

root->left = newNode(2);

root->right = newNode(3);

root->left->left = newNode(4);

root->left->right = newNode(5);

root->right->left =newNode(6);

A:

printf("\nPostorder traversal of binary tree is \n");

PostorderTraversal(root);

getchar();

string exe;

cout<<"\n Do you wanna exit \n";

cin>>exe;

if(exe=="yes")

{

system("pause");

return 0;

}

else{i=0;

k++;

ent='o';

system("cls");

goto A;

}

}

void medicineplz()

{

S:

int dose,notes;

string medtype;

cout<<" Medicines available are "<<endl<<" 1.'Panadol' for fever , rs 100 per dose "<<endl<<" 2.'Buffin' for fever , es 112 per dose "<<endl<<"3.'DIsprene' for heart , rs 100 per dose"<<endl<<"4.'Caltex' for heart , rs 15 per dose"<<endl<<"5.'Ogmantun' for lungs , rs 300 per dose"<<endl<<"6.'Honey' for Everthing, rs 500 per dose ";

cout<<endl;

cin>>medtype;

cout<<"\nHow much dose of the above medicine do you want ? : ";

cin>>dose;

if(medtype=="Panadol")

{

notes=100\*dose;

paid[k]=paid[k]+notes;

}

if(medtype=="Buffin")

{

notes=112\*dose;

paid[k]=paid[k]+notes;

}

if(medtype=="DIsprene")

{

notes=100\*dose;

paid[k]=paid[k]+notes;

}

if(medtype=="Caltex")

{

notes=15\*dose;

paid[k]=paid[k]+notes;

}

if(medtype=="Ogmantun")

{

notes=300\*dose;

paid[k]=paid[k]+notes;

}

if(medtype=="Honey")

{

notes=500\*dose;

paid[k]=paid[k]+notes;

}

string abcd;

cout<<"If you want anyother medicine also ";

cin>>abcd;

if(abcd=="yes")

{

goto S;

}

else

{

cout<<"Here are your medicines go oN!! much program still left ";

}

}

void manager()

{

int givenkey;

int key=1111;

cout<<"\nWarning you are entering manager mode if you are foung guility, you will be sent to jail, so you still have time enter 1 in key to retreive \n";

system("pause");

cout<<"\nEnter the secret key ";

cin>>givenkey;

if (givenkey==1)

{

return;

}

if (key==givenkey)

{

patientlist.display();

system("pause");

}

else

{

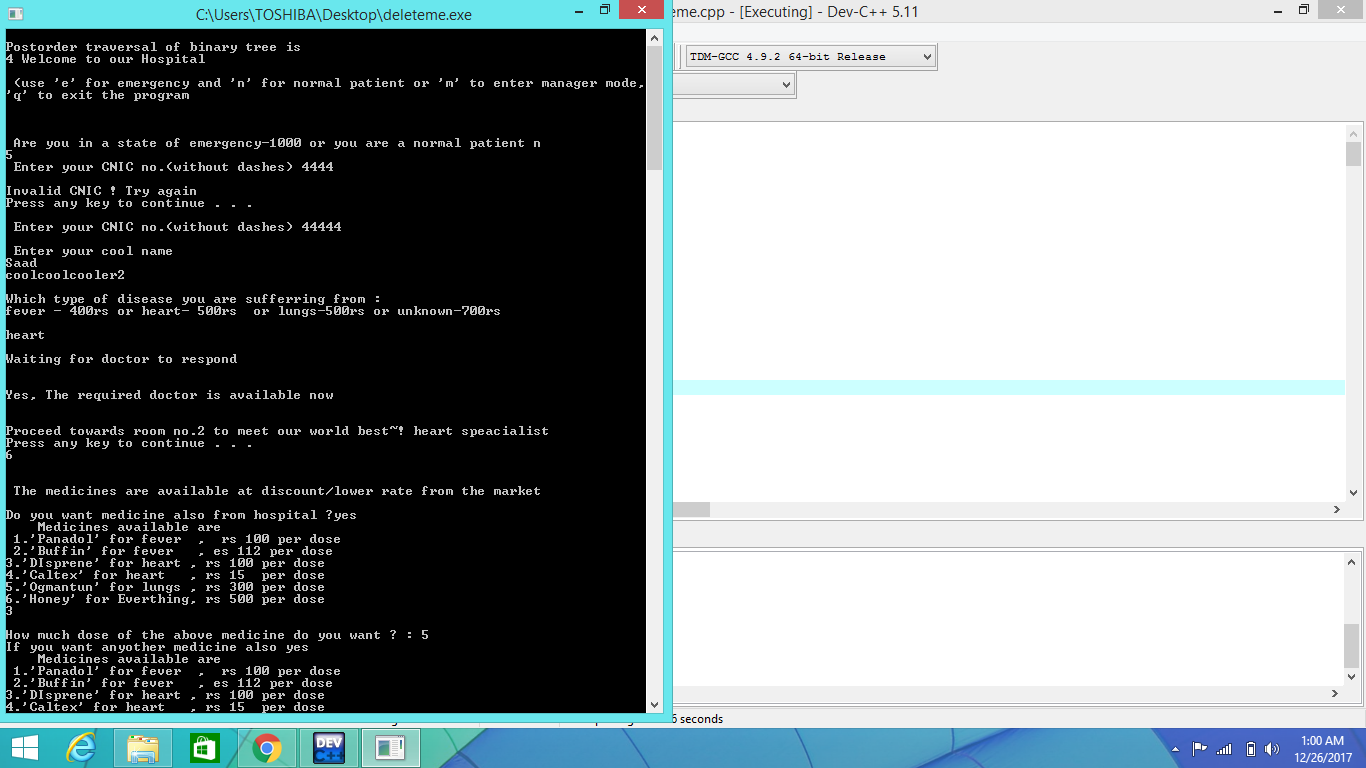
cout<<"\n Run! the police is after you ";

}

}

//"Snake Func."

**Output**

****