Field of computer Engineering

Data Structure

جامعة الجلالة
GALALA UNIVERSITY

Project Title *(English)*	**In body test**	
Team	1-Mohamed Elnakshbandy 2-Mohamed Yasser Saad 3-Mostafa Mohamed Ali	
Field	Computer Engineering	
Program	Artificial Intelligence	
Instructor	**Dr. Shaker El-sappagh**	
This part for instructor: Notice		**Degree**
Date of Submission		

Content

I. Conclusion...

❖ **Abstract project:**

We make project about Hospital Management System and we make it by using linked list that we study in data structure course

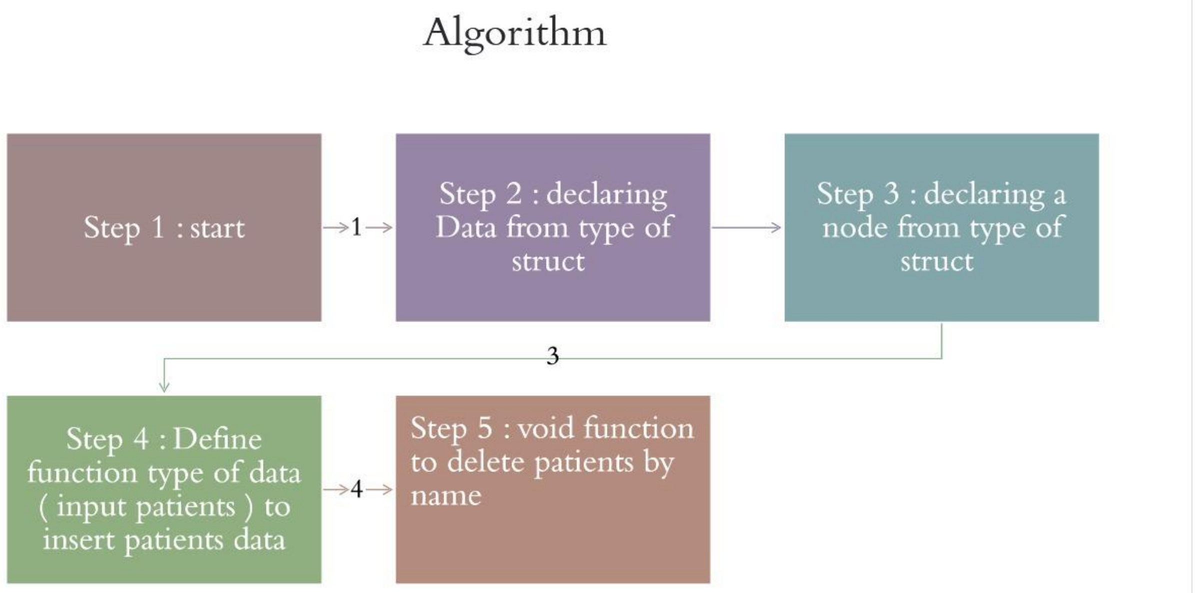
❖ **Introduction:**

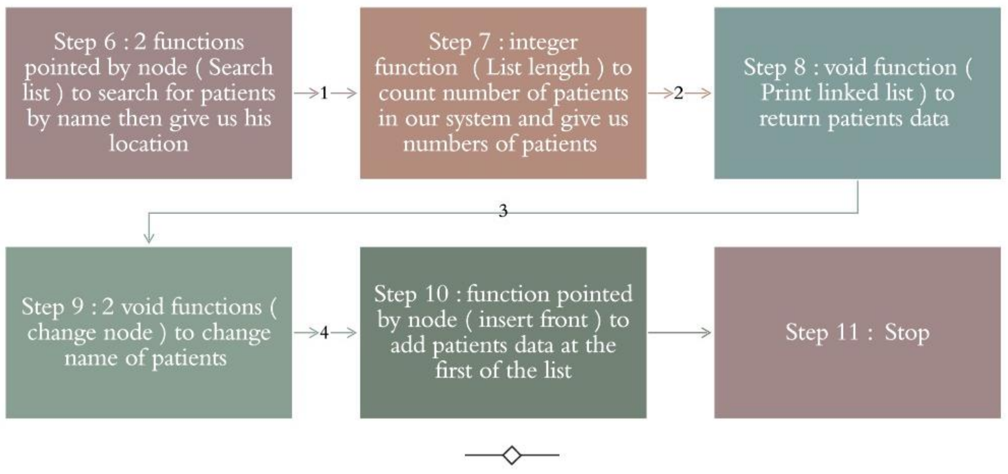
Well-tuned hospital management workflow involves lots of important decisions that should be made in the most efficient and quick way. Nowadays it is hard to implement it without the distinct hospital management system. In this article, we'll explore what is HMS software.

❖ **Objectives:**

1. Add patient
2. Delete patient
3. Search to any patient by name
4. List number of patient
5. Print all names of patient
6. Change name of any patient
7. Insert ne patient at front
8. Clear screen

❖ **System Analysis and Design:**


Algorithm
Step 1 : start
1
Step 2 : declaring Data from type of struct
Step 3 : declaring a node from type of struct
3
Step 4 : Define function type of data (input patients) to insert patients data
4
Step 5 : void function to delete patients by name


Step 6 : 2 functions pointed by node (Search list) to search for patients by name then give us his location
1
Step 7 : integer function (List length) to count number of patients in our system and give us numbers of patients
2
Step 8 : void function (Print linked list) to return patients data
3
Step 9 : 2 void functions (change node) to change name of patients
4
Step 10 : function pointed by node (insert front) to add patients data at the first of the list
Step 11 : Stop

❖ Implementation and Outputs:

```
#include <iostream>
#include <string>
using namespace std;

struct Data
{
string name;
string address;
string disease;
string gender;
string description;
int specialRoomNo;
int age;
};


struct Node
{
Data data;
Node* next;
};


Node* insertFront(Node* head, Data data)
{
Node* temp = new Node;
temp->data = data;
temp->next = head;
head = temp;
return head;

}//end of insertFront Function

Node* append(Node* head, Data data) {

Node* temp = new Node;

temp->data = data;
temp->next = NULL;
```

```
if (head == NULL)
{
head = temp;
return head;
}

Node* last = head;

while (last->next != NULL)
{
last = last->next;
}

last->next = temp;

cout << "\t\t\t\t\tAppend completed" << endl;

return head;

}
//end of append function

void changeNode(Node* head, Data data, Data newData)
{
while (head != NULL)
{
if (head->data.name == data.name)
{
head->data = newData;
break;
}

head = head->next;
}
}//end of changeNode function

void changeNode(Node* head, string data, string newData)
{
while (head != NULL)
{
if (head->data.name == data)
{
head->data.name = newData;
break;
}

head = head->next;
}
}
//end of changeNode function

void printLinkedList(Node* head)
{
if (head == NULL)
{
cout << "\t\t\t\t\tHead is null" << endl;
return;
}

while (head->next != NULL)
{
cout << "Name: " << head->data.name << endl;
cout << "Address: " << head->data.address << endl;
cout << "Gender: " << head->data.gender << endl;
cout << "Disease: " << head->data.disease << endl;
cout << "Description: " << head->data.description << endl;
cout << "Age: " << head->data.age << endl;
cout << "Specialist No: " << head->data.specialRoomNo << endl;
head = head->next;
}
cout << "Name: " << head->data.name << endl;
cout << "Address: " << head->data.address << endl;
cout << "Gender: " << head->data.gender << endl;
cout << "Disease: " << head->data.disease << endl;
cout << "Description: " << head->data.description << endl;
cout << "Age: " << head->data.age << endl;
cout << "Specialist No: " << head->data.specialRoomNo << endl;

}
//end of printLinkedList function

int listLength(Node* head) {
int temp = 0;
```

```
if (head == NULL)
{
cout << "\t\t\t\t\tNode is empty" << endl;
return 0;
}

while (head->next != NULL)
{
++temp;
head = head->next;
}

return temp + 1;
}
//end of the listLength function

void delElement(Node* head, int loc)
{
Node* temp = new Node;

temp = head;

if (head == NULL) {

cout << "\t\t\t\t\tNod is null" << endl;
return;
}


for (int i = 1; i <= loc; ++i)
{
temp = temp->next;
if (i < loc) {
head = head->next;
}//end of if statement

}//end of for statement

head->next = temp->next;

}
```

```
//end of delElement function

Node* searchList(Node* head, Data v) {

if (head == NULL)
{
cout << "\t\t\t\t\tNode is empty returning null" << endl;
return NULL;
}

int l = 1;

while (head->next != NULL && head->data.name != v.name)
{
head = head->next;
++l;
}

cout << "Element found at location " << l << endl;

return head;

}
//end of searchList function

Node* searchList(Node* head, string v)
{

if (head == NULL)
{
cout << "\t\t\t\t\tNode is empty returning null" << endl;
return NULL;
}

int l = 1;

while (head->next != NULL && head->data.name != v)
{
head = head->next;
++l;
}
```

```
cout << "Element found at location " << l << endl;

return head;

}
//end of searchList function

Data inputPatients()
{
string name, address, disease, gender, description;
int specialRoomNo, age;
Data p;

cout << "Enter Patient Name: ";
cin.ignore();
getline(cin, name);

cout << "Enter Patient Address: ";
getline(cin, address);

cout << "Enter Patient Disease: ";
getline(cin, disease);

cout << "Enter Patient Gender: ";
getline(cin, gender);

cout << "Enter Disease Description: ";
getline(cin, description);

cout << "Enter Patient Special Room No.: ";
cin >> specialRoomNo;

cout << "Enter Patient Age: ";
cin >> age;

p.name = name;
p.address = address;
p.gender = gender;
p.description = description;
p.specialRoomNo = specialRoomNo;
p.age = age;
```

```
cout << "\t\t\t\t\tCompleted input operation" << endl;

return p;

}
//end of inputPatients function

int main()
{

Node* head = NULL;
Data patient;
string nameToSearch;
string oldName, newName;

int op;
cout <<"\n\n\t\t\t\t\tHospital Management System"<<endl;
cout <<"\t\t1-Add Patient:"<<endl;
cout <<"\t\t2-Del Patient:" << endl;
cout <<"\t\t3-Search by Name:" << endl;
cout <<"\t\t4-List Length:" << endl;
cout <<"\t\t5-print List:"<<endl;
cout <<"\t\t6-Change Patient Name:" << endl;
cout <<"\t\t7-Insert new at front:" << endl;
cout <<"\t\t8-clear Screen:" << endl;
cout <<"\t\t\t\t\t(CTRL + Z)To exit:" << endl;
while (cin >> op)
{

switch (op)
{
case 1:
cout << "\t\t\t\t\tEnter Patient Details Below" << endl;
patient = inputPatients();
head = append(head, patient);
break;

case 2:

if (listLength(head) < 2)
{
cout << "Length is less then two.\nTerminating program" << endl;
exit(1);
}
else {
cout << "Enter location where you want to delete a patient, at least three patients must be in list? ";
int l;
cin >> l;
delElement(head, l - 1);
}
break;

case 3:
cout << "Enter name to search patient: ";
cin.ignore();
getline(cin, nameToSearch);
searchList(head, nameToSearch);
break;

case 4:

cout << "\t\t\t\t\tYou have " << listLength(head) << " Patients in your Hospital." << endl;

break;

case 5:
printLinkedList(head);
break;

case 6:
cin.ignore();
cout << "Enter old name ";
getline(cin, oldName);
cout << "Enter New Name ";
getline(cin, newName);

changeNode(head, oldName, newName);
break;

case 7:
patient = inputPatients();
head = insertFront(head, patient);
break;
case 8:
```

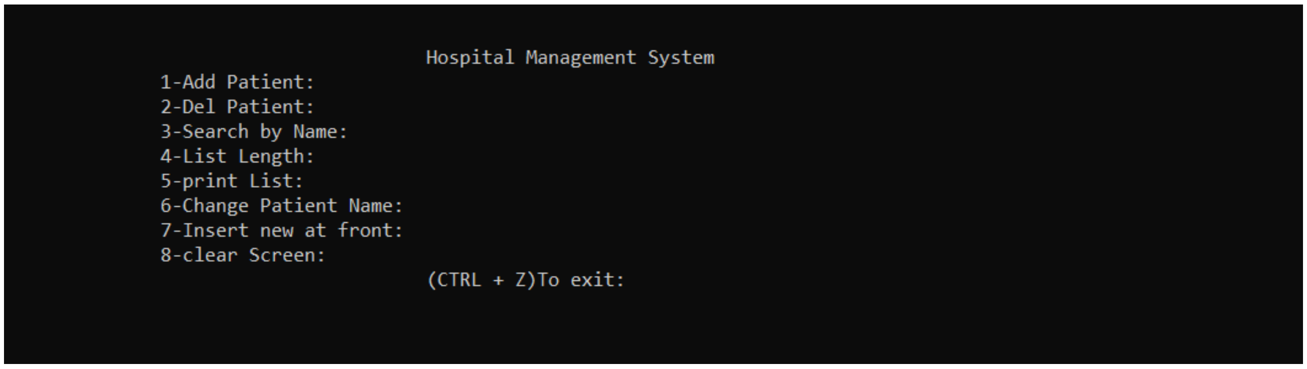
```
system("cls");
break;

default:
cout << "\t\t\t\t\tWrong option Selected" << endl;
}

cout << "\t\t1-Add Patient:" << endl;
cout << "\t\t2-Del Patient:" << endl;
cout << "\t\t3-Search by Name:" << endl;
cout << "\t\t4-List Length:" << endl;
cout << "\t\t5-print List:" << endl;
cout << "\t\t6-Change Patient Name:" << endl;
cout << "\t\t7-Insert new at front:" << endl;
cout << "\t\t8-clear Screen:" << endl;
cout << "\t\t\t\t\t(CTRL + Z)To exit:" << endl;
}

}
```

❖ **outputs:**



```
                                        Hospital Management System
                1-Add Patient:
                2-Del Patient:
                3-Search by Name:
                4-List Length:
                5-print List:
                6-Change Patient Name:
                7-Insert new at front:
                8-clear Screen:
                                        (CTRL + Z)To exit:
```

- **Conclusion:**

And here we reach an end of our project today. The hospital management system as presented could save the patients data, change name, delete patients, present the data of the patients in the hospital, by using the linked list concept.
Thank you for your attention