PROJECT

VIT Health Care Database Management

Subject: Data Structure and Algorithm (CSE 220)

Faculty: Prof. Chandra Mouli

Slot: D1+TD1

By:

* Sharan Raj (14BCE0612)

**ACKNOWLEDGMENT**

I take this opportunity to present my sincere appreciation to all those guidepost which really acted as lightening pillars to enlighten our way throughout this project that has led to successful and satisfactory completion of this study.

We are really grateful to our faculty Prof. Chandra Mouli for providing us with an opportunity to undertake this project in this university and providing us with all the facilities, he was extremely helpful in offering his professional expertise and bestowing us practical knowledge in all spheres related to the whole organization working.

Lastly, We are thankful to all those, particularly the various friends, who have been instrumental in creating proper, healthy and conductive environment and including new and fresh innovative ideas for us during the project, their help, it would have been extremely difficult for us to prepare in a time bound framework.

Name: Sharan Raj &

Rohit Ranjan.

Reg No. 14BCE0612 &

14BCE0661.

**TABLE OF CONTENTS**

1. Introduction

2. Abstract

3. Objectives

4. Application

5. System flow

6. Source code

7. Testing

8. Conclusion

9. Reference

**INTRODUCTION**

In the existing system, most of the records are maintained on paper. It becomes very inconvenient to modify the data. In the existing system, here is a possibility that the same data in different registers may have different values which means the entries of the same data do not match. This inconsistent state does not supply the concrete information which poses a problem in the case information related to particular search record.

Our project is very useful. User is no longer required to check his register in search of records, as now it can be searched over the software by choosing some options. The user need not to type in most of the information, He/she is just required to enter the desired options. On the whole it liberates the user from keeping lengthy manual records. In a nutshell, it abates the work load of an organization.

**ABSTRACT**

Data can be organized in many ways and a data structure is one of these ways. It is used to represent data in the memory of the computer so that the processing of data can be done in easier way.

The following concepts have been used in this project:

1. Linked List
2. Stack (Implementation through Linked List)
3. Searching
4. OOP Concepts
5. Data files
6. Browser popup using windows.h header file

1. I have used linked list to store the name and details like registration no address and blood group and age also I am dynamically storing this data in data file “patient” so that the data does not get lost when the program runs the data from data file is stored back in the file in the form of linked list

2. I have used stack for storing the logs of each individual person because in real life also logs are taken in LIFO order here also I am storing the data dynamically in data file ”name” and taking it back and pushing to the stack when the programing is running(1 persons log=1 stack+1 data file)

3. I have used linear searching of a linked list to take the record of his details by making the user enter his name

4. I have used class and object for linked list and stack

5. As mentioned earlier I have used data files

6. I have popped the following website for diagnoses of disease using a fun in #include<windows.h> header file <http://symptomchecker.isabelhealthcare.com/suggest_diagnoses_advanced/landing_page>

**OBJECTIVES**

A database is a collection of related data, which are organized so that useful information may be extracted. The effectiveness of databases derives from the fact that from one single, comprehensive database much of the information relevant to a variety of organizational purposes may be obtained. In health care DBMS the same database may be used by various organization personnel for patient care recording, for surveillance of patient status, and for treatment advice; it may be used by researchers in assessing the effectiveness of drugs and clinical procedures; and it can be used by administrative personnel in cost accounting and by management for the planning of service facilities.

**APPLICATION**

The notion of a database encompasses the data themselves, the hardware used to store the data, and the software used to manipulate the data. The application includes multi-platform support, easy configuration, zero administration, high reliability and reliable data repository.

This application is meant for administratives of health center (VIT) also students it stores details of a person who has been admitted to the health center.

It records and shows your current health status.

It calculates your health status by BMI.

It predicts the disease when entered the symptoms.

**SYSTEM FLOW**

N

Y

Y

N

Wish to continue?

Diagnosis

Symptoms

New

User?

Prescription

Appointment

Health Status

Logs

Sign Up

Sign In

**SOURCE CODE**

**#include<fstream>**

**#include<iostream>**

**#include<sstream>**

**#include<string>**

**#include <windows.h>**

**using namespace std;**

**// Creating a NODE Structure**

**struct logs**

**{**

**string log;**

**struct logs \*next;**

**};**

**// Creating a class STACK**

**class stack**

**{**

**struct logs \*top;**

**public:**

**stack() // constructure**

**{**

**top=NULL;**

**}**

**void push(int,string,string); // to write a log**

**logs\* pop(); // to read the log**

**};**

**// PUSH Operation**

**void stack::push(int c,string nam1,string loge)**

**{**

**if(c==1)**

**{**

**ofstream file;**

**file.open(nam1.c\_str(),ios::out|ios::app|ios::binary);**

**string lg;**

**cin.ignore();**

**cout<<"Enter the log ";**

**getline(cin,lg);**

**file<<lg;**

**struct logs \*ptr=new logs;**

**ptr->log=lg;**

**ptr->next=NULL;**

**if(top!=NULL)**

**ptr->next=top;**

**top=ptr;**

**cout<<"\nNew item is inserted to the stack!!!";**

**file<<endl;**

**file.close();**

**}**

**else**

**{**

**struct logs \*ptr;**

**ptr=new logs;**

**ptr->log=loge;**

**ptr->next=NULL;**

**if(top!=NULL)**

**ptr->next=top;**

**top=ptr;**

**}**

**}**

**// POP Operation**

**logs\* stack::pop()**

**{**

**struct logs \*temp=top;**

**if(top==NULL)**

**{**

**cout<<"\nThe log is empty!!!\n";**

**}**

**cout<<temp->log<<endl;**

**temp=temp->next;**

**delete temp;**

**}**

**struct node**

**{**

**string name,regno,address,bloodgrp;**

**int age;**

**struct node \*next;**

**};**

**class list**

**{**

**struct node \*head,\*tail;**

**public :**

**list()**

**{**

**head=tail=0;**

**}**

**int insert(int ,string);**

**node\* search(string);**

**string ret();**

**void display(node \*);**

**};**

**int list::insert(int x,string line)**

**{**

**if(x==1)**

**{**

**node \*temp=new node;**

**ofstream fout;**

**fout.open("patient.txt",ios::out|ios::app|ios::binary);**

**cout<<"Enter the details \nNAME:";**

**cin>>temp->name;**

**fout<<endl<<temp->name<<" ";**

**cout<<"\nREGISTRATION NO:";**

**cin>>temp->regno;**

**fout<<temp->regno<<" ";**

**cout<<"\nADDRESS:";**

**cin>>temp->address;**

**fout<<temp->address<<" ";**

**cout<<"\nAGE:";**

**cin>>temp->age;**

**fout<<temp->age<<" ";**

**cout<<"\nBLOOD GROUP:";**

**cin>>temp->bloodgrp;**

**fout<<temp->bloodgrp<<endl;**

**fout.close();**

**temp->next=0;**

**if(temp==0)**

**return -1;**

**if(head==0)**

**head=tail=temp;**

**else**

**{**

**tail->next=temp;**

**tail=temp;**

**}**

**}**

**else**

**{node \*temp=new node;**

**istringstream iss(line);**

**iss>>temp->name;**

**iss>>temp->regno;**

**iss>>temp->address;**

**iss>>temp->age;**

**iss>>temp->bloodgrp;**

**if(temp==0)**

**return -1;**

**else**

**if(head==0)**

**head=tail=temp;**

**else**

**{**

**tail->next=temp;**

**tail=temp;**

**}**

**}**

**return 1;**

**}**

**node\* list::search(string sch)**

**{ node \*temp;**

**temp=head;**

**if(temp==0)**

**return 0;**

**else**

**if(temp->name==sch)**

**{**

**cout<<"\nitem found at first node";**

**return temp;**

**}**

**for( ;temp!=0;temp=temp->next)**

**{**

**if(temp->next->name==sch)**

**{**

**cout<<"\nitem found\n";**

**return temp->next;**

**}**

**}**

**}**

**string list::ret()**

**{**

**return(tail->name);**

**}**

**void list::display(node \*temp)**

**{**

**cout<<"\t\t PATIENT DETAILS \nNAME:"<<temp->name<<endl;**

**cout<<"\nREGISTRATION NO:"<<temp->regno<<endl;**

**cout<<"\nADDRESS:"<<temp->address<<endl;**

**cout<<"\nAGE:"<<temp->age<<endl;**

**cout<<"\nBLOOD GROUP:"<<temp->bloodgrp<<endl;**

**}**

**void bmi()**

**{**

**float height,bmi,weight;**

**cout<<"Enter your height(cm)\n";**

**cin>>height;**

**height/=100;**

**cout<<"Enter your weight\n";**

**cin>>weight;**

**bmi=weight/(height\*height);**

**if(bmi<18.5)**

**cout<<"NOT HEALTHY UNDER WEIGHT\nBMI="<<bmi<<endl;**

**else**

**if(bmi>18.5&&bmi<24.9)**

**cout<<"HEALTHy NORMAL WEIGHT\nBMI-"<<bmi<<endl;**

**else**

**if(bmi>24.9&&bmi<29.9)**

**cout<<"NOT HEALTHY OVER WEIGHT\nBMI="<<bmi<<endl;**

**else**

**cout<<"NOT HEALTHY OBESE\nBMI="<<bmi<<endl;**

**}**

**int main()**

**{ l:**

**list obj;**

**stack st;**

**string s,nam,loger;**

**char op;**

**int choice;**

**/\* ofstream fout;**

**fout.open("patient.txt",ios::out|ios::app|ios::binary);**

**fout.close();\*/**

**ifstream fin;**

**fin.open("patient.txt",ios::in|ios::binary);**

**while(!(getline(fin,s).eof()))**

**{**

**obj.insert(0,s);**

**}**

**fin.close();**

**cout<<"\t\t\tVIT-HEALTH CARE DBMS\n";**

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

**cout<<"CAPSLOCK SHOULD BE ON\n\n";**

**cout<<"Is this your first visit to health center?(Y/N)\n";**

**cin>>op;**

**if(op=='Y')**

**{ obj.insert(1,s);**

**s=obj.ret();**

**string nam=s+".txt";**

**cout<<"\n1.HEALTH STATUS\n2.LOGS\n3.NEW APPOINMENT\n";**

**cin>>choice;**

**if(choice==1)**

**{**

**bmi();**

**}**

**else**

**if(choice==2)**

**{ int i=1;**

**do//logs**

**{**

**if(i==1)**

**{**

**++i;**

**st.push(1,nam,nam);**

**}**

**else**

**if(i>1)**

**{**

**cout<<"1:new log\n2:read log\n";**

**cout<<"\nEnter your choice: ";**

**cin>>choice;**

**switch(choice)**

**{**

**case 1:**

**st.push(choice,nam,nam);**

**break;**

**case 2:**

**st.pop();**

**--i;**

**break;**

**default:**

**cout<<"Please enter correct choice!!";**

**break;**

**}**

**}**

**cout<<"\nwould you like to do again\n";**

**cin>>op;**

**}while(op=='Y');**

**}**

**else**

**if(choice==3)**

**{**

**ShellExecute(NULL, "open", "http://symptomchecker.isabelhealthcare.com/suggest\_diagnoses\_advanced/landing\_page",NULL, NULL, SW\_SHOWNORMAL);**

**}**

**}**

**else**

**if(op=='N')**

**{**

**node \*ptr;**

**cout<<"Enter your name\n";**

**cin>>nam;**

**ptr=obj.search(nam);**

**obj.display(ptr);**

**nam=nam+".txt";**

**do**

**{**

**cout<<"\n1.HEALTH STATUS\n2.LOGS\n3.NEW APPOINMENT\n";**

**cin>>choice;**

**if(choice==1)**

**{**

**bmi();**

**}**

**else**

**if(choice==2)**

**{**

**ifstream file;**

**file.open(nam.c\_str(),ios::in|ios::binary);**

**while(!(getline(file,loger)).eof())**

**{**

**st.push(choice,loger,loger);**

**}**

**fin.close();**

**do//logs**

**{**

**cout<<"1:new log\n2:read log\n";**

**cout<<"\nEnter your choice: ";**

**cin>>choice;**

**switch(choice)**

**{**

**case 1:**

**st.push(choice,nam,nam);**

**break;**

**case 2:**

**st.pop();**

**break;**

**default:**

**cout<<"Please enter correct choice!!";**

**break;**

**}**

**cout<<"\nwould you like to check your logs again?\n";**

**cin>>op;**

**}while(op=='Y');**

**}**

**else**

**if(choice==3)**

**{**

**ShellExecute(NULL, "open", "http://symptomchecker.isabelhealthcare.com/suggest\_diagnoses\_advanced/landing\_page",NULL, NULL, SW\_SHOWNORMAL);**

**}**

**cout<<"would like to go back to options\n";**

**cin>>op;**

**}while(op=='Y');**

**}**

**cout<<"Would like to reuse the app\n";**

**cin>>op;**

**if(op=='Y')**

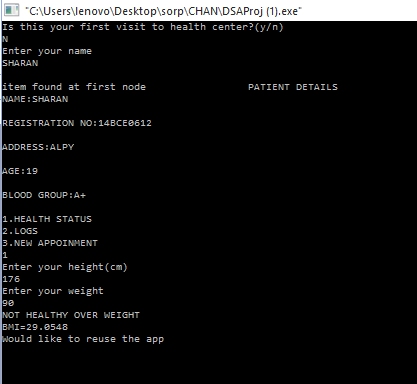
**goto l;**

**}**

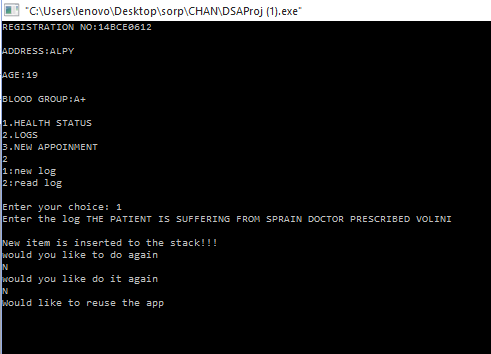
**TESTING**

After the coding phase, computer programs are available that can be executed for testing purpose. So after testing, the outputs of my project are as follows:

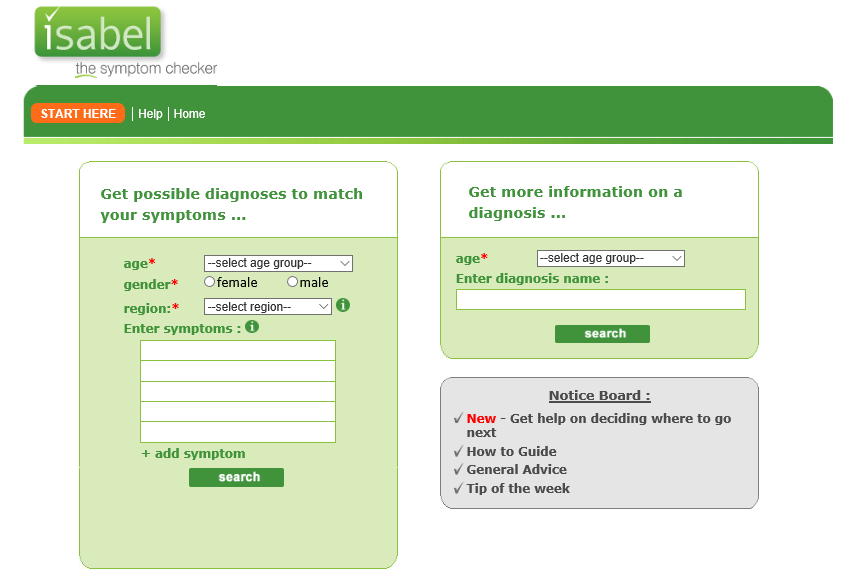
Health status:-



Logs:-



Symptom checker:-



**CONCLUSION**

Here we would like brief about the difficulties faced while doing this project:

1. Lot of syntax error typed in was very difficult to identify do to the length of the code and caused program terminations.
2. Splitting a string read from a data file.
3. While reading from a file wrong input came.
4. Using “.eof()” in file handling was wrong way as it caused the last entry in the data file to repeat and cause an additional node in the linked list.
5. To link each node of the linked list with a stack for each one.
6. To open a website in c++.

The following is what we did to rectify the above difficulties and thusly gained new knowledge:

1. Since program was terminating at different junctures had to manually test individual parts of the code to rectify each error one by one learned to be more careful while typing the code because most was do too carelessness.
2. Here we used #include<sstream> header file which contained an inbuilt function to do splitting of string(iss fun()).
3. Using binary file rectified the error.
4. This was the toughest to debug because we didn’t know the above mentioned fact so we used this syntax instead “if(!(getline(file,s)).eof()) this prevents this mistake.
5. For this used header file windows.h which had an inbuilt fun ShellExecute(NULL, "open", "website link",NULL, NULL, SW\_SHOWNORMAL); all other methods we saw was long and complex.
6. Here we wanted to make a complex data structure of our own which is in the following form so here the top chain represents a inked list and bottom is a stack for doing so in data file we used the name which is a content of linked list as the name of the data file for which is used “.c\_str()” fun to name the data file as a variable.

So by doing the above I was able to success fully complete our project.

Databases are used in a variety of industries, specifically speaking in health care. This project will address how databases can be used across the health care industry. It will examine the architectures and discuss the relational model in detail. The needs of databases users in health care will also be explored. Healthcare databases consist of data that reflect clinical and clinically related information. The data are usually collected through the routine processes and activities of patient care; however, its usefulness goes beyond the operational applications that generate the data. The health care databases project can be used for quality improvement, patient safety and security, improved decision making, and preventing redundancy. Healthcare databases facilitate the organization by facilitating, organizing, storing, and processing large amounts of data. They are also useful for billing, keeping patient records, and automating sensitive information.

**REFERENCE**

* <https://en.wikipedia.org/wiki/Linked_data_structure>
* <http://www.geeksforgeeks.org/data-structures/#LinkedList>
* <http://www.slideshare.net>
* Introduction to Algorithm by Thomas H. Cormen
* Mastering C++ by K.R Venugopal