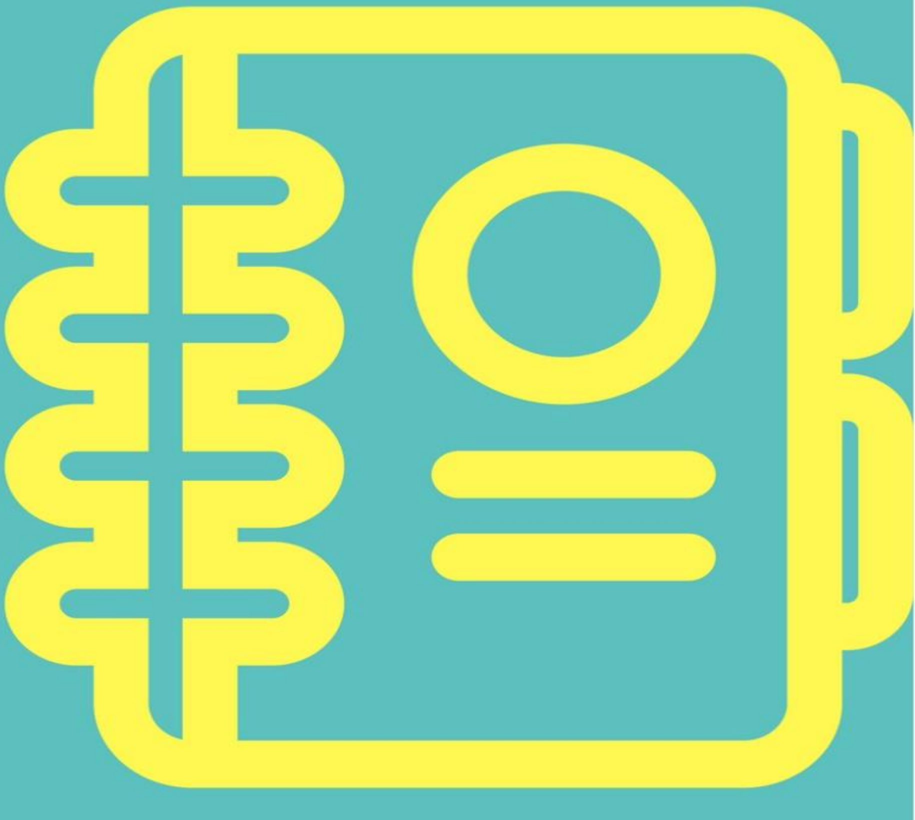
C++ PROGRAMMING DOCUMENTATION

‘C++’Programmming Project on
CONTACT MANAGEMENT SYSTEM



Course code: CSE 130
Course Name : SoftwareDevelopment I Lab
Lecturer: Md. Muhibullah

CONTACT MANAGEMENT SYSTEM

ABSTRACT

The adoption of Electronic contact evolved from the age of computers. Because of time efficiency & storage. It takes only few seconds to search a contact and use it. Also is enables us to rewrite the existing data, transferring data. Contact management system is a great system for recruiters who need access to contacts without the hassle of managing information by hand. The main purpose of contact management is to record and organize contacts. This allows a user to keep a full record of all a user's recruiting communications. Instead of losing data in stacks of paper or spreadsheets, all information is in one, streamlined database. As a recruiter, a user has long relationships with many of people. Furthermore, Peoples are using it more than traditional phonebook.

ACKNOWLEDGEMENT

First of all, I extol to omnipotent Allah for giving me the opportunity to complete the thesis.

I owe a great debt to my supervisor, Md. Mohibullah,lecturer, Department of Computer Science & Engineering, Brittania University, for his contributions, suggestion, guidance and encouragement, this thesis and report has been completed. We consider our self extremely fortunate to be able to work under the guidance of such a dynamic personality.

Also I am greatly indebted to all of my teachers of our department who helped us to pursue our project.

In particular, we are indebted to my classmates for their constant encouragement, kind help and support in doing this thesis and report.

Finally, I thankfully acknowledge the kind assistance of all staffs of our department, my family and friends for giving me constant encouragement throughout the Project.

SUBMISSION

A
Project report on "Contact Management System"

Submitted of the computer practical
2018
Of
Batch 12th

For the practical fulfillment of the requirement for the syllabus.

Submitted by: Thamina Omar Toma,
Safiqul Islam
&
Shahadat Hossain

CERTIFICATE

This is to certify that the project entitled, “Contact Management System" submitted by Thamina Omar Toma, Safiqul Islam & Shahadat Hossain in partial fulfillment of the requirements for the award of "Name of Diploma/Degree" in "Name of stream" at the "Name of institution" is an authentic work carried out by him under my supervision and guidance.

To the best of my knowledge, the matter embodied in the project has not been submitted to any other University / Institute for the award of any Degree or Diploma.

Date: 27.052018
Contact Management system
(Britannia University)

CONTENTS

S.no Topics page

Chapter 1: INTRODUCTION

Chapter 2: Problem Analysis & Solution

Chapter 3: Application Design

S.no Topics page

Chapter 4: IMPLEMENTATION & Its Process

Chapter 5: Testing Phase

Chapter 6: HARDWARE AND SOFTWARE DESCRIPTION

Chapter 7: CONCLUSION AND FUTURE SCOPE

Introduction:

The project 'Contact Management' is the sample project for managing contact details. The project aims at developing contact management system using the C++ language that enables an organization to maintain its library.

The project demonstrates the creation of a user interface of a system, without the use of C++ Graphics library.

The application uses basic C++ function to generate menus, show message boxes and print text on the screen.

To display customized text with colors and fonts according to application requirements, functions have been created in the application, which fetch the exact video memory addresses of a target location, to write text at a particular location.

The application also implements the concept of structures to define the contact items. It also effectively applies the various C++ concepts such as file operations, looping and branching constructs and string manipulation functions.

General Overview of the Problem:

We tried to build a contact management system . Where a user can save his/her contact and update or can delete their contact. We build it by using c++ program for its

object-ordinated feature. We used simple console command for the Ui as it was fast and compact.

Aim for this application:

The project that we have undertaken aims to develop a contact management system that is clean, user-friendly and multi-functional. Development of this application includes a number of fields such that user feels comfortable and the system appears as dynamic to him. The project "Contact Management System" includes the following functionalities:

- *User can save their contact fluently and quickly*
- *Users can manipulate their contact list*
- *Users can Delete data*
- Users can search for certain data.
- *All user's data are stored in files on a stable storage*
- *User have their data password protected*
- *This system possesses password-protected administrative access; thus, preventing the whole management system from unauthorized access*
- Better visuals
- Faster performance

Proposed System and Objectives:

The application will be extremely beneficial for the users intending to use and manage their contact and will get various benefits in the field of management of contact on a clean and user-friendly platform.

"Contact Management System", is a simple application, which is especially generated and designed for the purpose of managing various contact efficiently. It is user name and ID protected as well.

Following are the major objectives behind the new proposed system:

- *It provides security from unauthorized access, only admin is access granted to the system.*
- *It increases efficiency and saves the time.*
- *No any danger and obstacles from external entities.*
- *Easy access of saved data inside the system.*
- *Management operations are efficiently handled by the application*
- *It is highly secured and less time consuming; hence time wastage can be avoided*
- *It creates a user-friendly environment, where a normal user can access through all the benefits of the system.*

Problem description:

The Contact Management System is an application for maintaining a person's contact number in a one place. The system provides the access to the user`s contact numbers, address and many things. The following documentation provides the specification of the system.

We are mainly concerned with developing a contact system where a user can save his/her contacts. if user wants she can edit his / her saved data. they also can add many other data such as mail address, Date of birth many other things. This system is developed for fast and portability.

Module description:

Module	*Description*
Insert Contact	*Selecting this create a new file for the user by accepting input such as name and phone number*
Delete Contact	*User can delete a contact .*
Search Contact	*Enables to search for the details of the given information*
Checking Exiting Contact	*Check if a contact is already inserted In the database .*
Update Contact	*Provides option to edit Contact Details.*
Show All	*Displays only the number detail at a time*

Input Specification:

Struct Information:

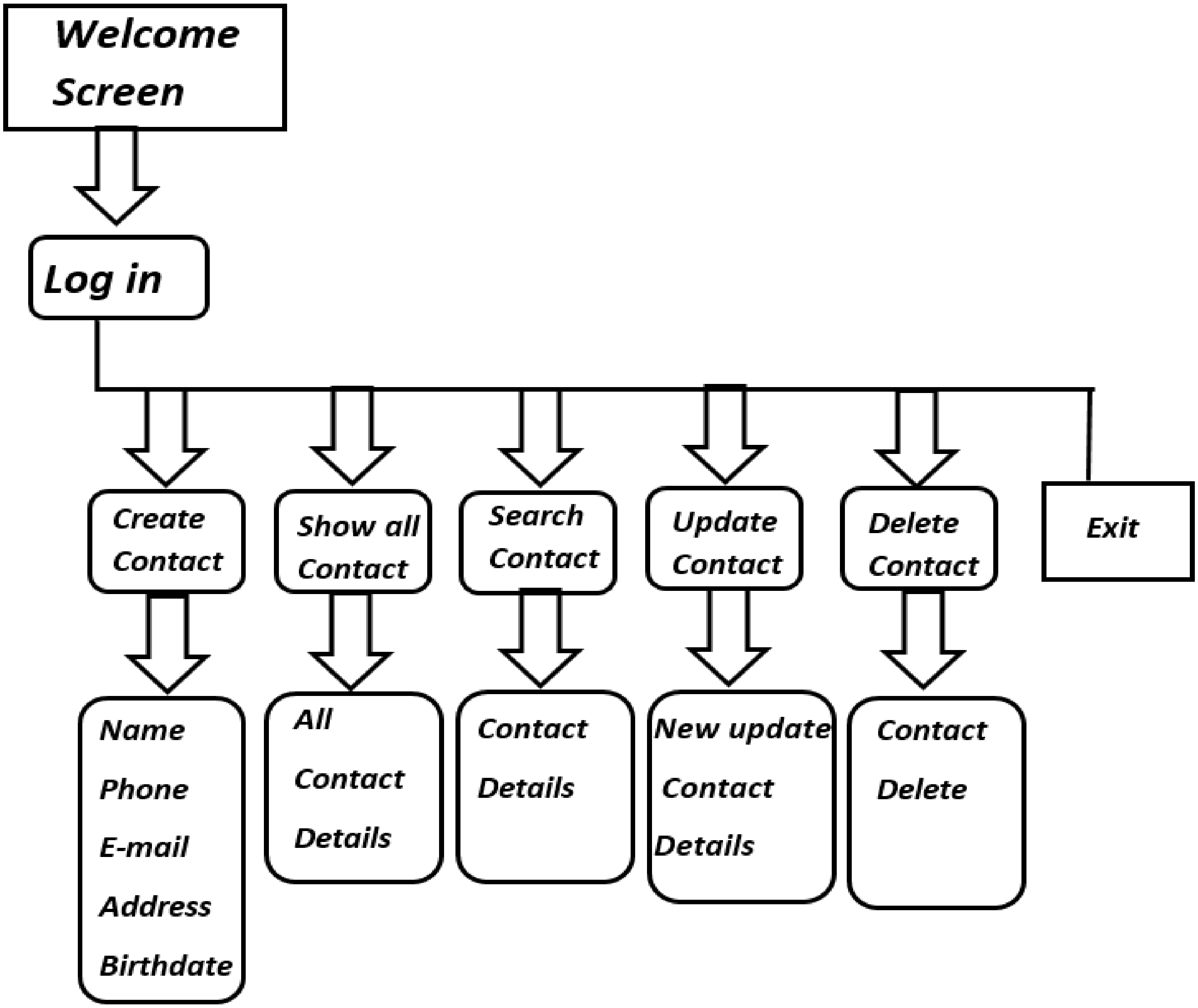
S.no	Variable Name	Description	Data Type	Size
1.	*Name*	*Contact Name*	*Char*	*25*
2.	*Phone*	*Contact Phone number*	*Char*	*25*
3.	*Mail_add*	*Contact Mail address*	*Char*	*25*
4.	*h_address*	*Contact Home address*	*Char*	*25*
5.	*dob*	*Birthdate of Contact*	*Char*	*25*

File Specification:

Name	Description	Data Type
**ff*	*file pointer to create account*	*File Pointer*
**fp*	*file pointer to add Contact*	*File Pointer*
**fp*	*file pointer to Add other data*	*File Pointer*
**fp*	*file pointer to View all details*	*File Pointer*
**fp*	*file pointer to Edit data*	*File Pointer*
**fp*	*file pointer to exit system*	*File Pointer*

Screen Design:

Output screen design:



OBJECTIVES:

Input Design is the process of converting a user-oriented description of the input into a computer-based system. This design is important to avoid errors in the data input process and show the correct direction to the management for getting correct information from the computerized system.

It is achieved by creating user-friendly screens for the data entry to handle large volume of data. The goal of designing input is to make data entry easier and to be free from errors. The data entry screen is designed in such a way that all the data manipulates can be performed. It also provides record viewing facilities.

When the data is entered it will check for its validity. Data can be entered with the help of screens. Appropriate messages are provided as when needed so that the user will not be in a maze of instant. Thus the objective of input design is to create an input layout that is easy to follow.

A quality output is one, which meets the requirements of the end user and presents the information clearly. In output design it is determined how the information is to be displaced for immediate need and also the hard copy output. It is the most important and direct source information to the user. Efficient and intelligent output design improves the system's relationship to help user decision-making. Designing computer output should proceed in an organized, well thought out manner; the right output must be developed while ensuring that each output element is designed so that people will find the system can use easily and effectively.

DATABASE DESIGN:

A database is an organized mechanism that has the capability of storing information through which a user can retrieve stored information in an effective and efficient manner. The data is the purpose of any database and must be protected. The database design is a two level process. In the first step, user requirements are gathered together and a database is designed which will meet these requirements as clearly as possible. This step is called Information Level Design and it is taken independent of any individual DBMS. In the second step, this Information level design is transferred into a design for the specific DBMS that will be used to implement the system in question. This step is called Physical Level Design, concerned with the characteristics of the specific DBMS that will be used. A database design runs parallel with the system design. The organization of the data in the database is aimed to achieve the following two major objectives.

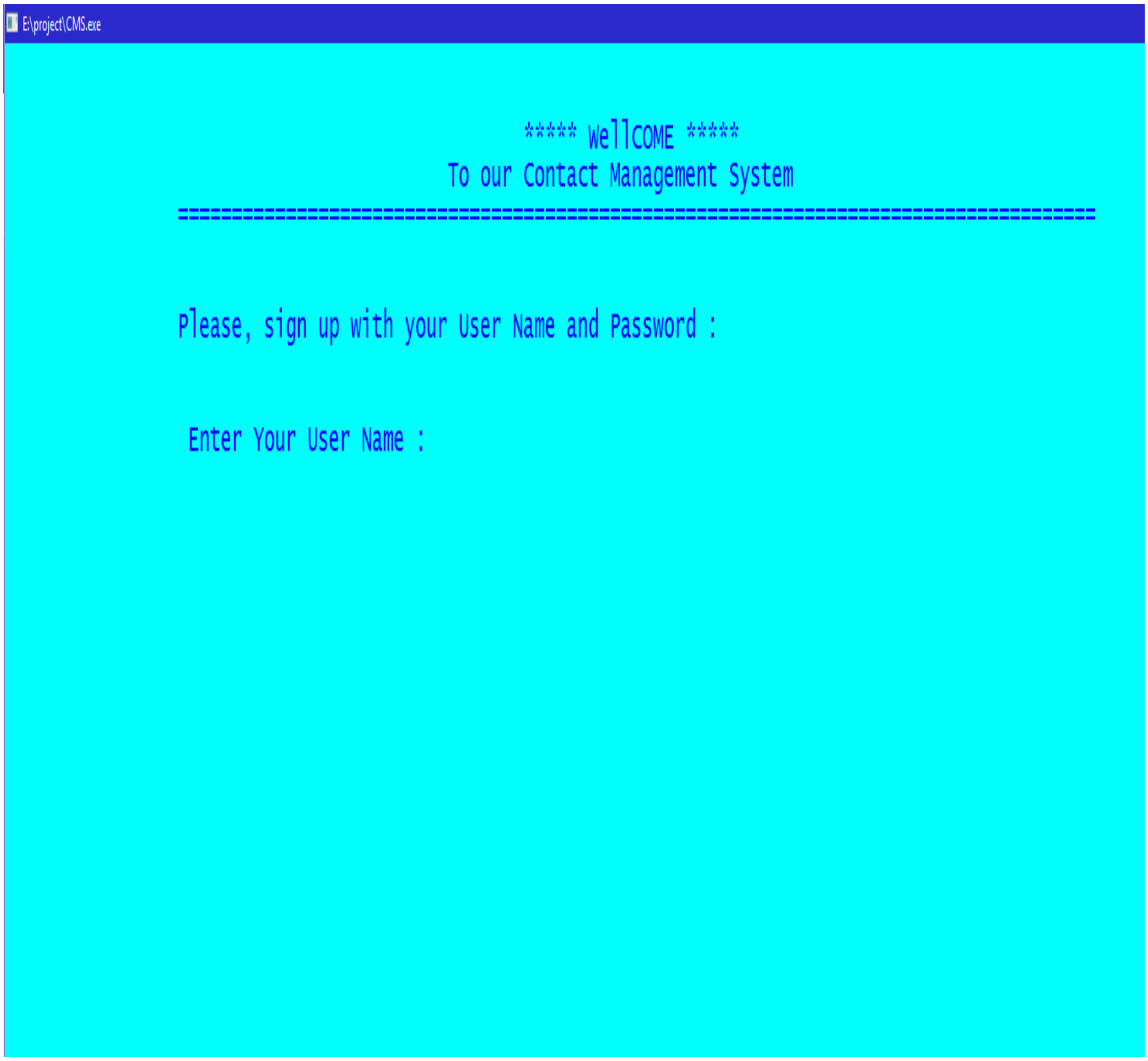
- ✓ *Data Integrity*
- ✓ *Data independence*

Normalization is the process of decomposing the attributes in an application, which results in a set of tables with very simple structure. The purpose of normalization is to make tables as simple as possible. Normalization is carried out in this system for the following reasons.

- ✓ *To structure the data so that there is no repetition of data, this helps in saving.*
- ✓ *To permit simple retrieval of data in response to query and report request. To simplify the maintenance of the data through updates, insertions, deletions.*
- ✓ *To reduce the need to restructure or reorganize data which new application requirements arise.*

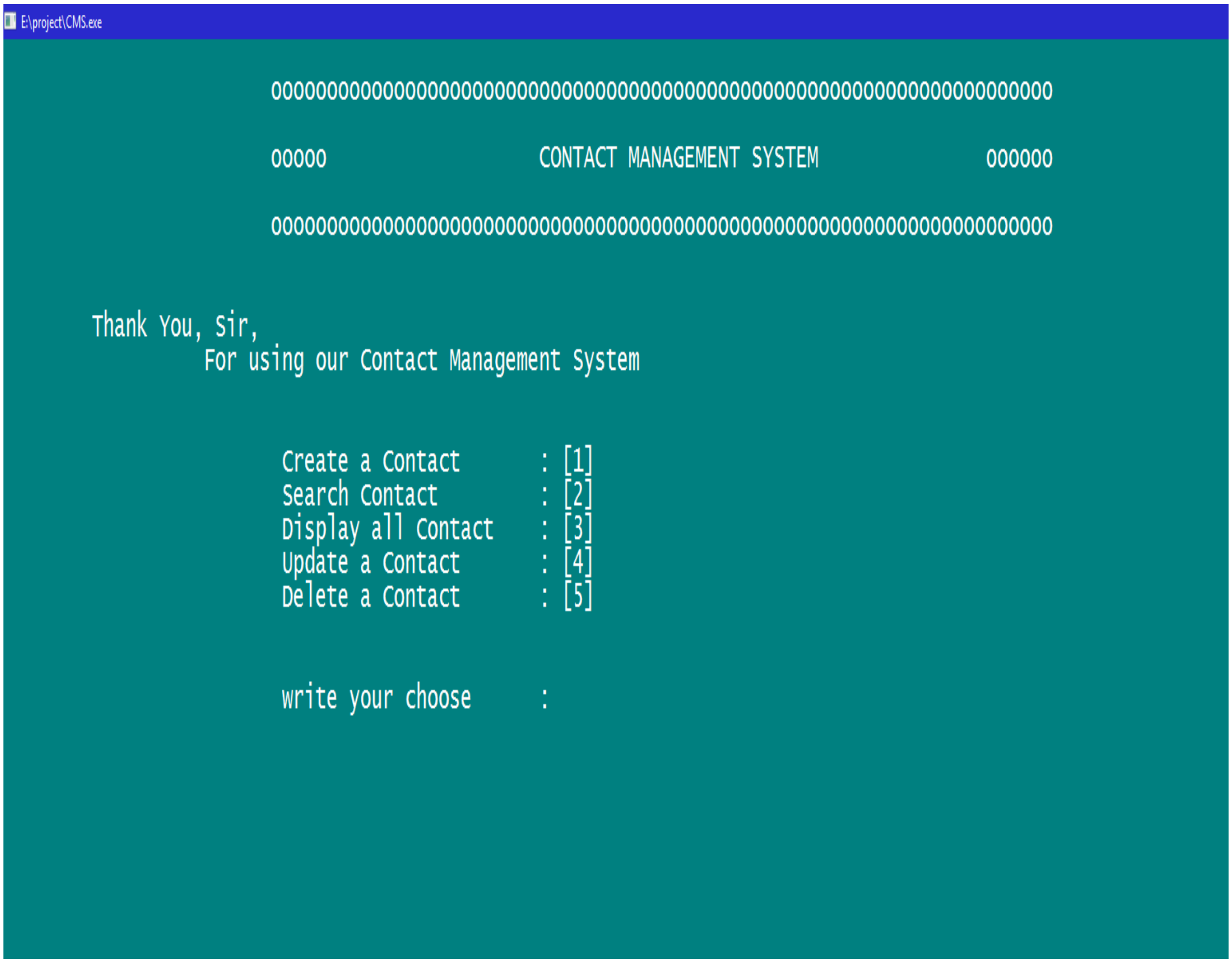
Screen Shots:

Welcome screen



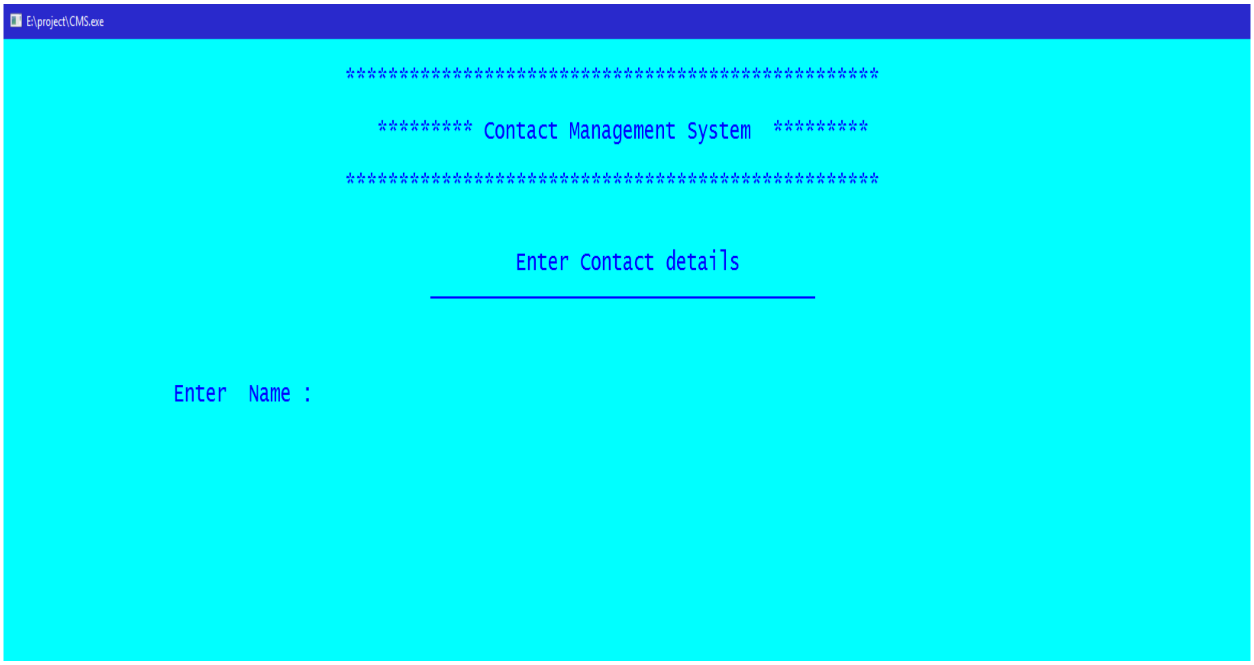
```
                                ***** WellCOME *****
                         To our Contact Management System
=====================================================================================


Please, sign up with your User Name and Password :


 Enter Your User Name :
```

Logging in as User

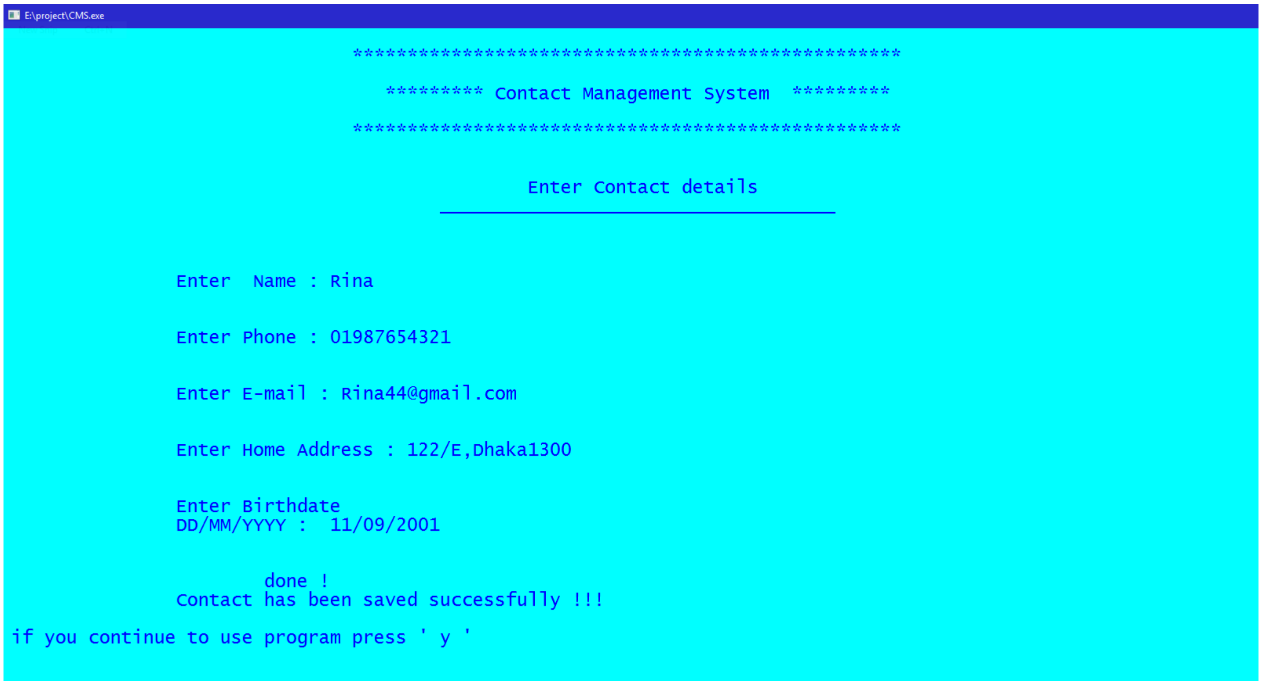


```
                        00000000000000000000000000000000000000000000000000000000000000000000000

                        00000                  CONTACT MANAGEMENT SYSTEM             000000

                        00000000000000000000000000000000000000000000000000000000000000000000000


        Thank You, Sir,
                  For using our Contact Management System


                         Create a Contact       : [1]
                         Search Contact         : [2]
                         Display all Contact    : [3]
                         Update a Contact       : [4]
                         Delete a Contact       : [5]


                         write your choose      :
```

Insert Contact

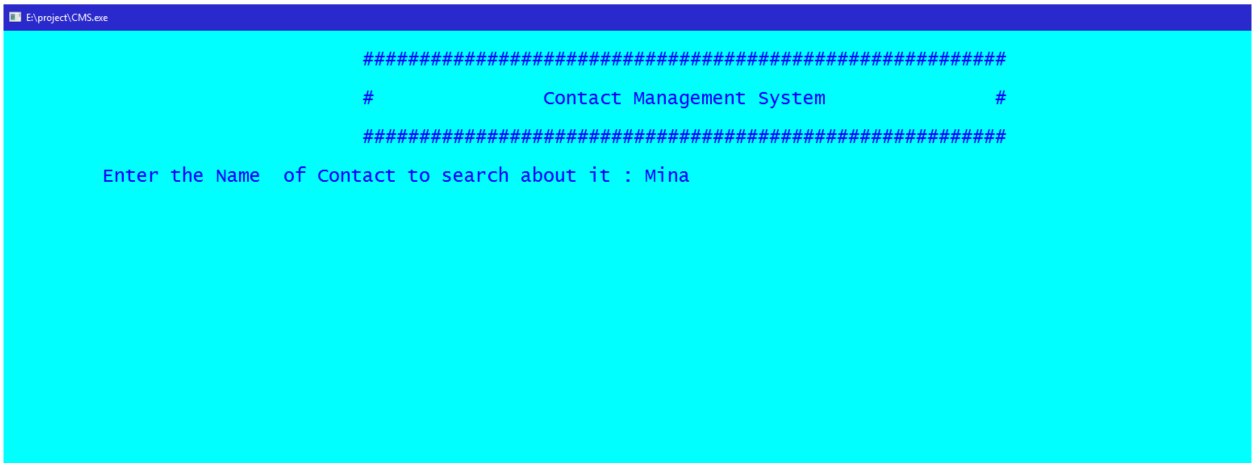


```
                   ***************************************************

                      *********  Contact Management System  *********

                   ***************************************************


                              Enter Contact details
                          _____________________________


        Enter  Name :
```

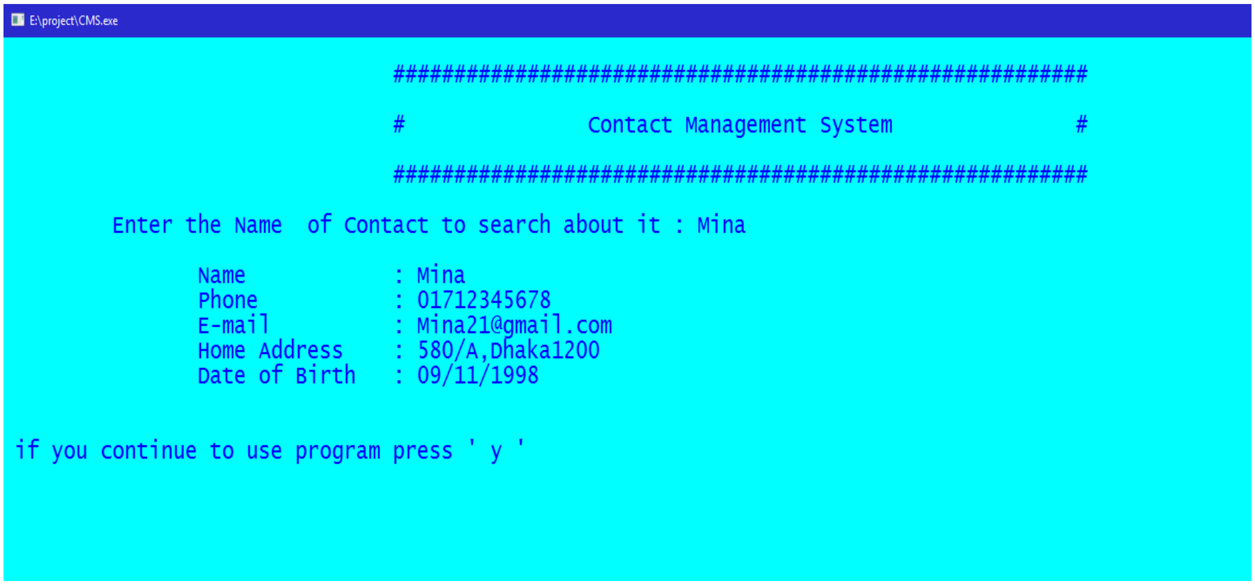


```
                   ***************************************************

                      *********  Contact Management System  *********

                   ***************************************************


                              Enter Contact details
                          _____________________________


        Enter  Name : Rina

        Enter Phone : 01987654321

        Enter E-mail : Rina44@gmail.com

        Enter Home Address : 122/E,Dhaka1300

        Enter Birthdate
        DD/MM/YYYY :  11/09/2001

                done !
        Contact has been saved successfully !!!

if you continue to use program press ' y '
```

Search Contact

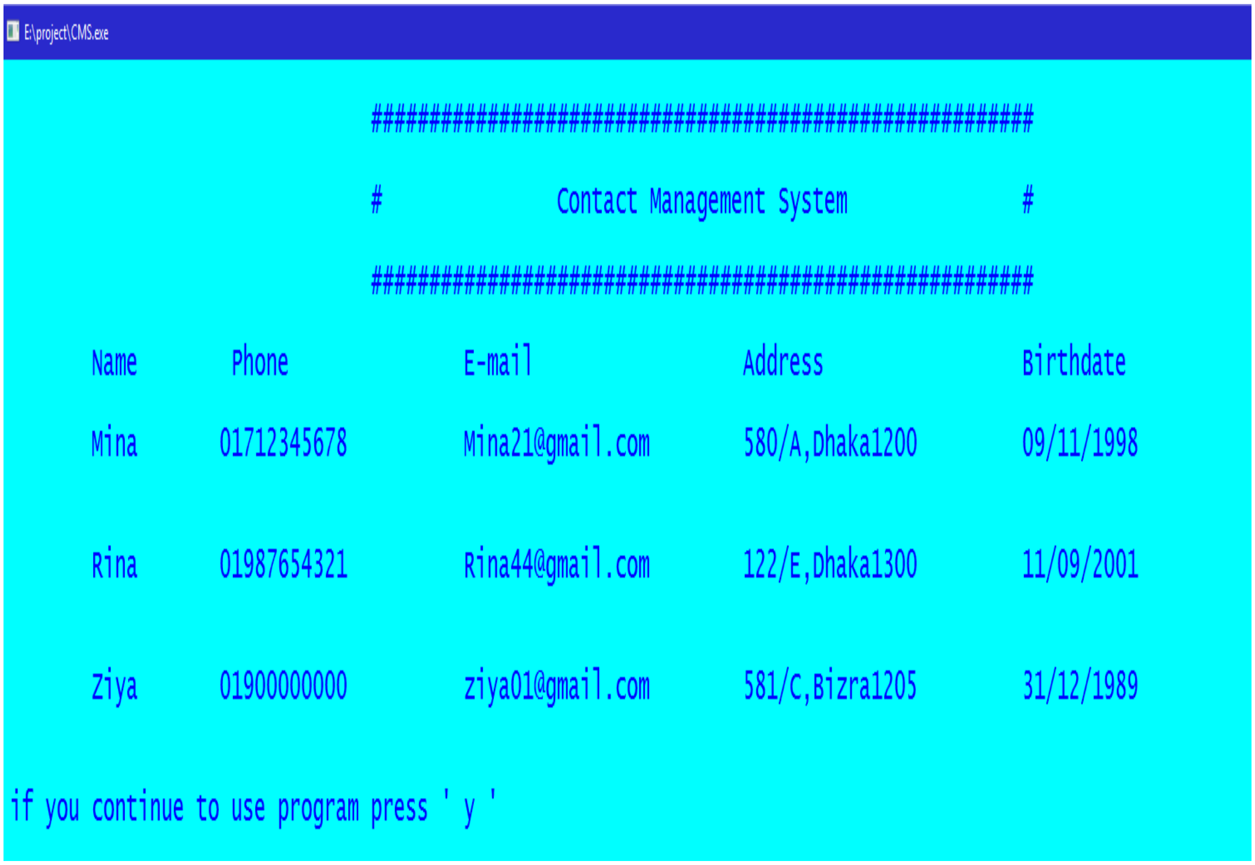


```
                                #########################################################
                                #               Contact Management System               #
                                #########################################################
         Enter the Name  of Contact to search about it : Mina
```



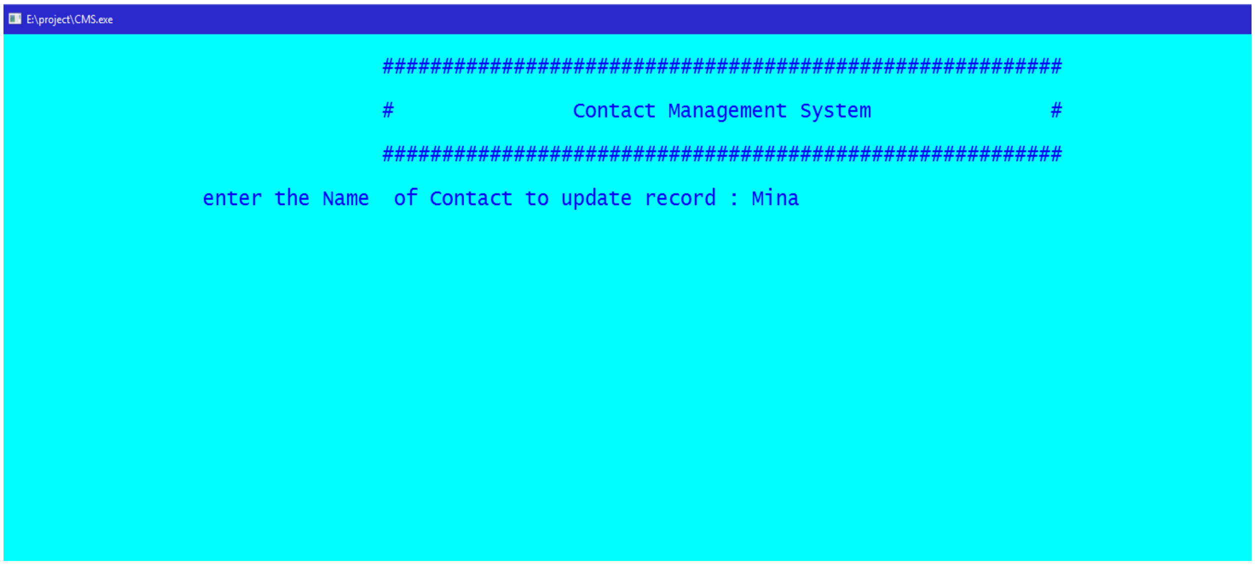
```
                                #########################################################

                                #               Contact Management System               #

                                #########################################################

         Enter the Name  of Contact to search about it : Mina

                Name            : Mina
                Phone           : 01712345678
                E-mail          : Mina21@gmail.com
                Home Address    : 580/A,Dhaka1200
                Date of Birth   : 09/11/1998


 if you continue to use program press ' y '
```

Display All Contact

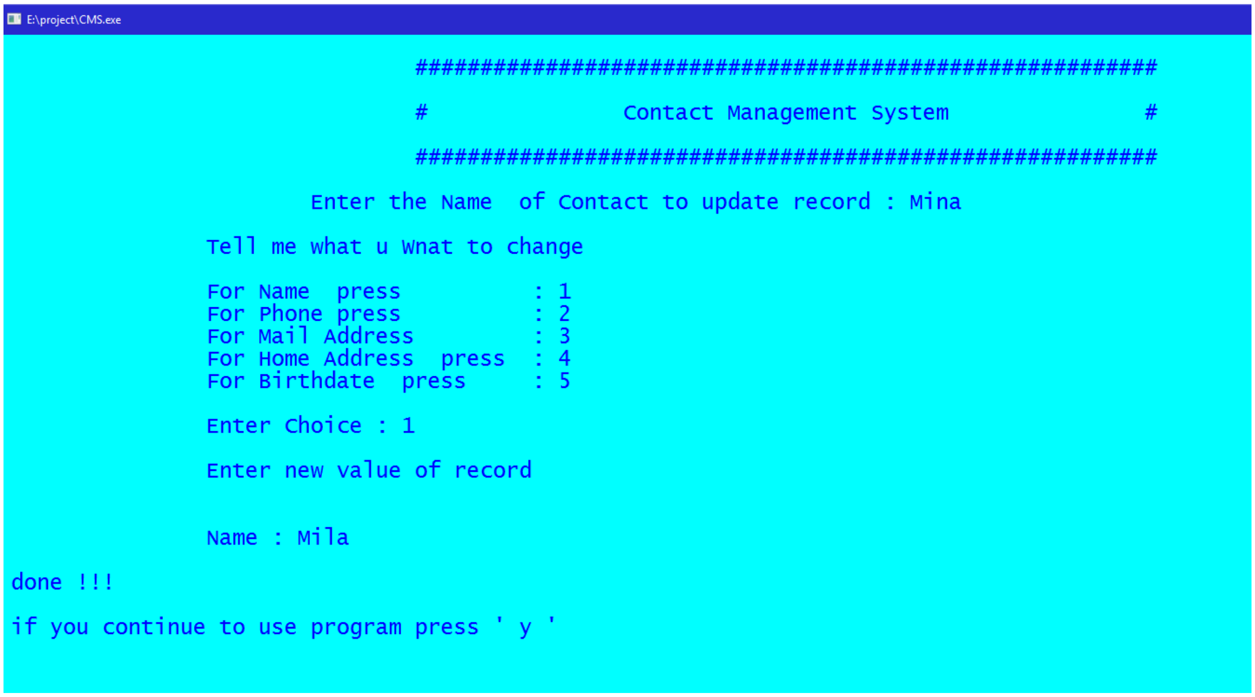


```
                                  #########################################################

                                  #               Contact Management System               #

                                  #########################################################

        Name         Phone                 E-mail                    Address                   Birthdate

        Mina        01712345678            Mina21@gmail.com          580/A,Dhaka1200           09/11/1998


        Rina        01987654321            Rina44@gmail.com          122/E,Dhaka1300           11/09/2001


        Ziya        01900000000            ziya01@gmail.com          581/C,Bizra1205           31/12/1989


if you continue to use program press ' y '
```

Update Contact

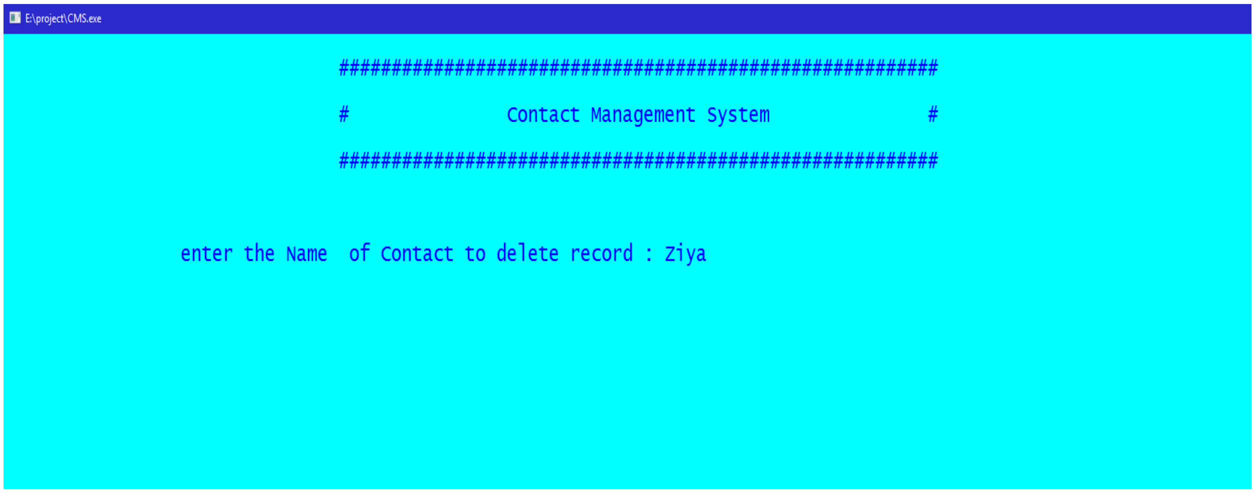


```
                               #########################################################

                               #               Contact Management System               #

                               #########################################################

                enter the Name  of Contact to update record : Mina
```

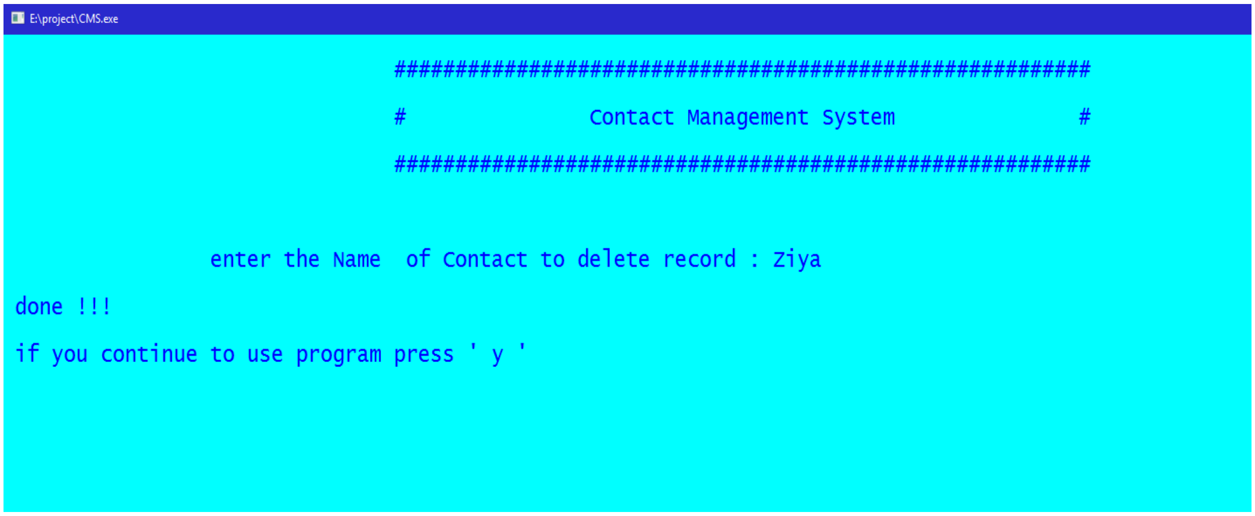


```
                               #########################################################

                               #               Contact Management System               #

                               #########################################################

                       Enter the Name  of Contact to update record : Mina

               Tell me what u Wnat to change

               For Name  press          : 1
               For Phone press          : 2
               For Mail Address         : 3
               For Home Address  press  : 4
               For Birthdate  press     : 5

               Enter Choice : 1

               Enter new value of record


               Name : Mila

done !!!

if you continue to use program press ' y '
```

Delete Contact



```
                               #########################################################
                               #               Contact Management System              #
                               #########################################################


                enter the Name  of Contact to delete record : Ziya
```



```
                               #########################################################
                               #               Contact Management System              #
                               #########################################################


                enter the Name  of Contact to delete record : Ziya
done !!!
if you continue to use program press ' y '
```

Processing and validation:

Main menu:

User have to Enter his user name and his password to login into his system.

User Menu:

Here user can choose what he wants to do. like insert contact, delete or Edit.

Creating A Contact:

Here user has to ***fill all the data*** *which is require to create a contact like name, phone number etc.*

Viewing Contact:

Here user enables to ***view*** *his / her all the contacts.*

Search Contact:

*Here user **search** his/her **desired** contact by Name.*

Editing Contact:

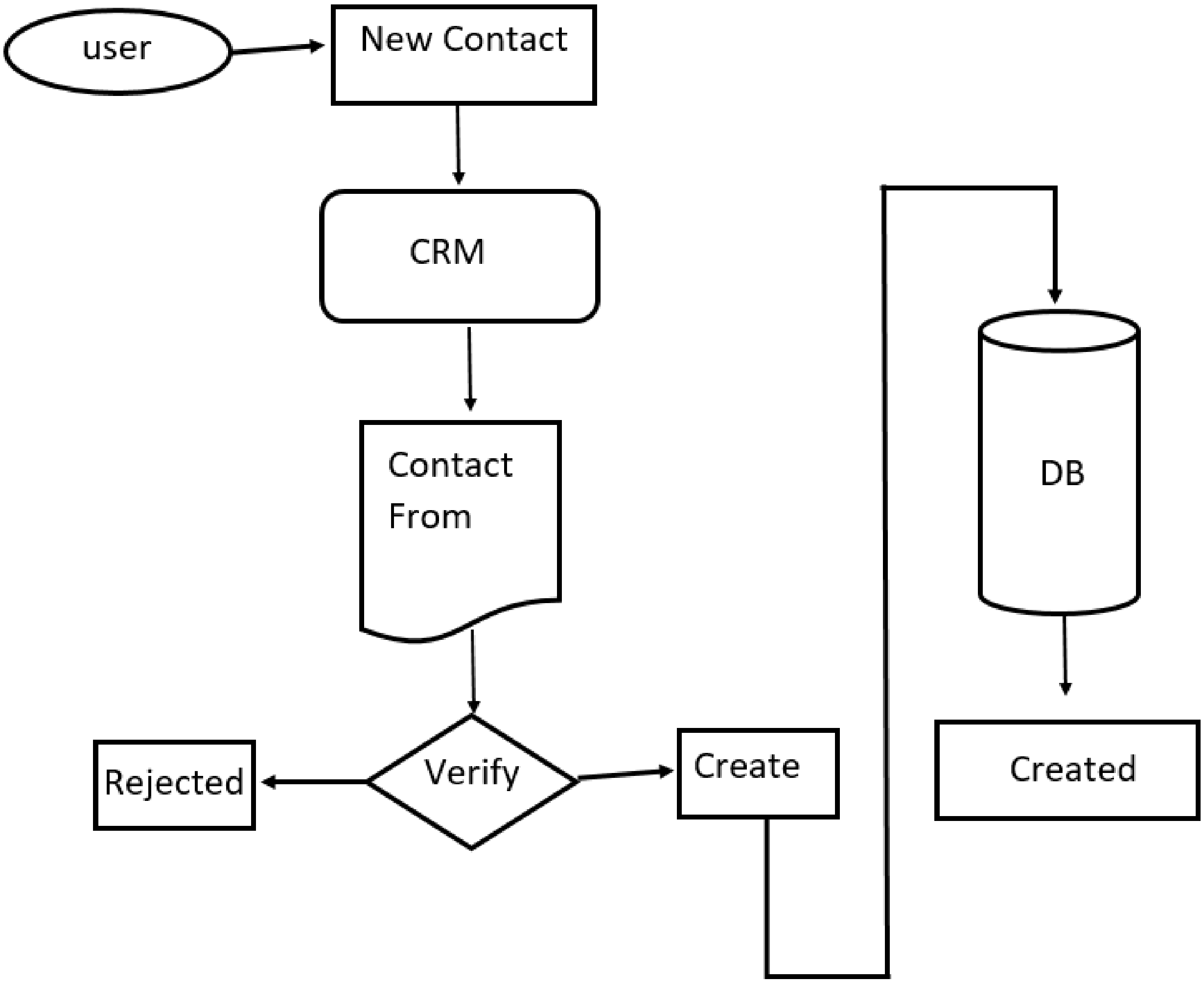
*Here user can **edit** his / her already saved contact.*

Uniqueness:

*Each Contact is unique from each other. There is a small chance for a **duplicate** contact. If user insert the 1st character of contact's name in lowercase automatically converted into **uppercase**. User can change **specific** item from contact record.*

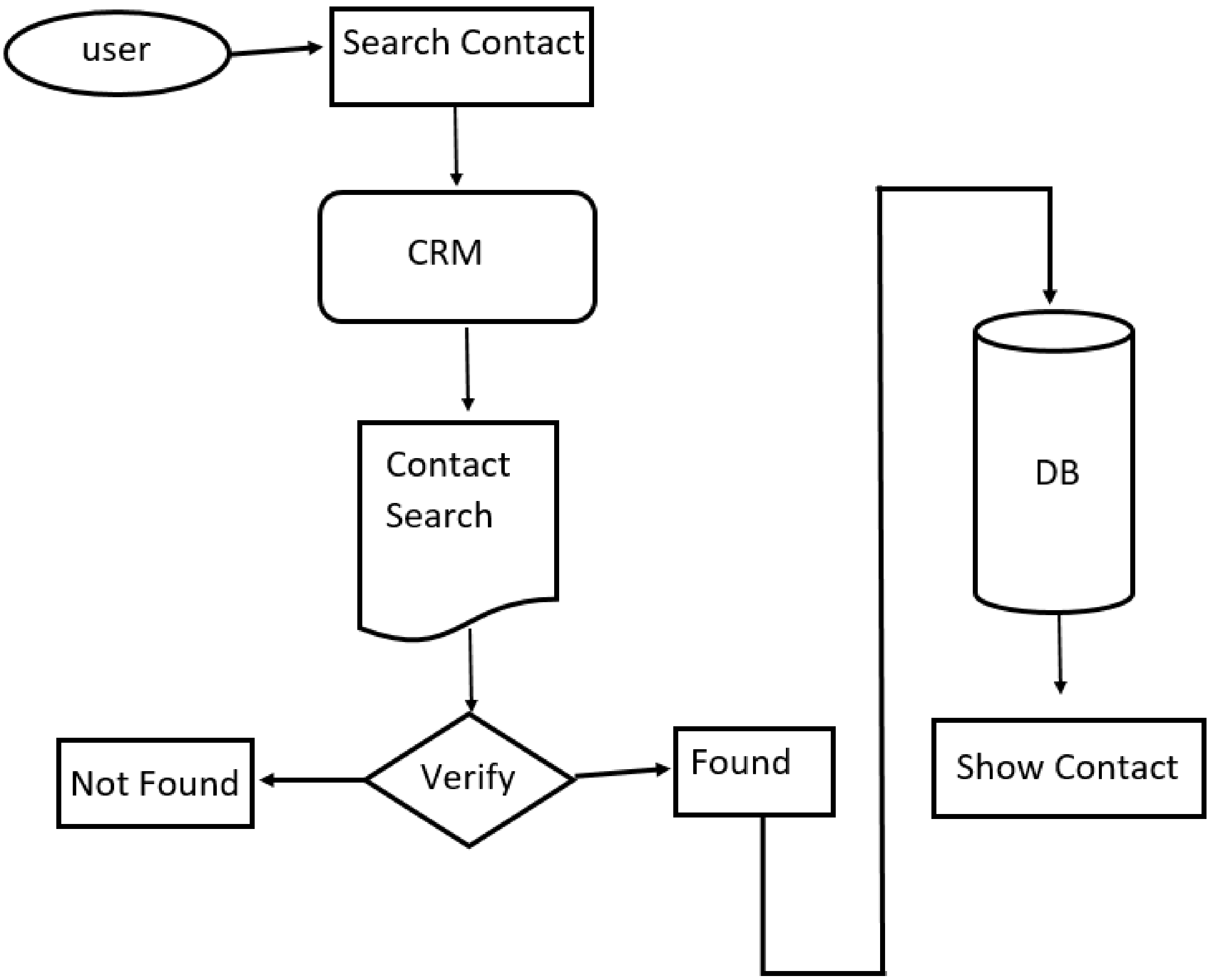
Flowchart Diagrams

For Creating Contact:



For Searching Contact:

42



Programming Testing

S.no	***Test Objective***
1	*To check whether program runs or not.*
2	*To check if the program menu displays all menu options or not.*
3	*To check all options are right or not.*
4	*To check a Contact was created or not.*
5	*To check if the program Edit Contact Working or not .*
6	*To check if the Delete Contact Working or not .*
7	*To check if 'view Contact' option shows the info or not.*
8	*To check if checking existing content works properly or not.*
9	*To check if the exit option works or not.*
10	*To check whether menu options return back or not.*
11	*To check whether menu is opened after entering correct password*
12	*To check whether mgs of "name or Phone. is not valid" prompt or not.* *when incorrect number or phone. is entered.*
13	*To check if the color text are displayed or not.*

Tests:

Test Case	***1***
Test Objective	*To check whether program runs or not.*
Test Data	*Running the program.*
Expected Result	*Main screen should display successfully.*
Test Result	*Main screen appears correctly.*
Conclusion	*Expected result matches actual result.*

Test Case	***2***
Test Objective	*To check if the program menu displays all menu options or not.*
Test Data	*Open the main menu*
Expected Result	*All the options should be displayed correctly with serial other*
Test Result	*Menu options are displayed in right order*
Conclusion	*Expected result matches actual result.*

Test Case	***3***
Test Objective	*To check all options are right or not.*
Test Data	*Open menus*
Expected Result	*All options are right*
Test Result	*All options are right and displayed correctly.*
Conclusion	*Expected result matches actual result.*

Test Case	***4***
Test Objective	*To check Contact was created or not.*
Test Data	*Enter Data .*
Expected Result	*contact must be created.*
Test Result	*Contact was successfully created*
Conclusion	*Expected result matches actual result.*

Test Case	5
Test Objective	To check if the program Edit Contact after Saving .
Test Data	New data
Expected Result	New data must be added in Contact
Test Result	New data must be added in Contact
Conclusion	Expected result matches actual result.

Test Case	6
Test Objective	To check if the program Delete Contact after Saving .
Test Data	Enter Name
Expected Result	Contact must be deleted from the database
Test Result	Data was successfully updated
Conclusion	Expected result matches actual result.

Test Case	*7*
Test Objective	*To check if 'view contact ' option shows the info or not..*
Test Data	*View contact*
Expected Result	*Details must be show*
Test Result	*Details appear correctly.*
Conclusion	*Expected result matches actual result.*

Test Case	*8*
Test Objective	*To check if Checking Existing Contact works properly or not..*
Test Data	*Enter the same old Data .*
Expected Result	*Program must recognized that the data is already saved e*
Test Result	*Showing Warning successfully.*
Conclusion	*Expected result matches actual result.*

Test Case	9
Test Objective	To check if the exit option works or not.
Test Data	Run program.
Expected Result	Exit program using option.
Test Result	Program exit.
Conclusion	Expected result matches actual result.

Test Case	10
Test Objective	To check whether menu options return back or not.
Test Data	Run program
Expected Result	Menu must be returns.
Test Result	Menu returns.
Conclusion	Expected result matches actual result.

Test Case	11
Test Objective	To check whether menu is opened after entering correct password.
Test Data	Enter password.
Expected Result	Menu must be returns.
Test Result	Menu returns.
Conclusion	Expected result matches actual result.

Test Case	12
Test Objective	To check if the color text are displayed or not.
Test Data	Run program.
Expected Result	Color test must be seen.
Test Result	Color test seen.
Conclusion	Expected result matches actual result.

Results:

s.no	Test Objective	Results
1.	To check whether program runs or not.	successful
2.	To check if the program menu displays all menu options or not.	successful
3.	To check all options are right or not.	successful
4.	To check account was created or not.	successful
5.	To check if the program Edit Contact Working or not.	successful
6.	To check if the program Delete Contact working or not.	successful
7.	To check if 'view Contact' option shows the info or not.	successful
8.	To check if Checking Exist Contact works properly or not.	successful
9.	To check if the exit option works or not.	successful
10.	To check whether menu options return back or not.	successful
11.	To check whether menu is opened after entering Correct password.	successful
12.	To check if the color text are displayed or not.	successful
13.	To check if the color text are displayed or not.	successful

Hardware and Software Specification:

Configuration for hardware and software:

It does not need any additional hardware or software to operate the program, but the following requirements should be strongly maintained:

Requirements for hardware:

- *Pentium II and hardware.*
- *512MB of RAM or higher.*
- *800MHz processor or above.*
- *CD ROM.*
- *20 MB of hard disk space.*

Requirements for software:

- *Operating System WINDOWS 98 or higher.*
- *Program Code-blocks or Any other IDE needs to be installed If Anyone Wants to make Change.*
- *The content of DB files should not be deleted as they are the database of this software.*
- *Necessary hardware needed to operate I/O operation.*

Implementation

Platform:

The whole code was done by Code Blocks IDE. Which is a versatile ide for both c and c++ .

Code:

Below here is the full source code of the project. For better understanding we used tab line for describe the whole code.

Header file :

```
     #include<iostream>
     #include<fstream>
    #include<cstring>
    #include<cstdlib>
```

Create Account :

```
void contact :: insert_contact()
{
system("cls");
system("color b9");
label :
     cout<<"\n\t\t\t\t***************************************************"<<endl;
    cout<<"\n\t\t\t\t   ********* Contact Management System   *********"<<endl;
    cout<<"\n\t\t\t\t***************************************************"<<endl;
    cout<<"\n\n\t\t\t\t\t\tEnter Contact details
\n\t\t\t\t\t________________________________________ "<<endl;
   fstream stud;
    stud.open("contact.txt", ios::app);
    cin.ignore();
    cout<<" \n\n\n\t\tEnter  Name : ";
    cin.getline(name,25);
    if(name[0]>=97 && name[0] <= 122)
    {
        cout<<"\n\n\t\tThe 1st char of Name must be uppercase "<<endl; ;
        cout<<"\t\tplease try Again' "<<endl ;
      goto label;
 }
    else
 {
         string c_name ;
        c_name= name ;
        int a ;
      a =checking_existing_name( c_name);
      if(a==1)
      {
          cout<<"\n\n\t\tThis Name   is already Taken  "<<endl; ;
         cout<<"\t\tplease try Again' "<<endl ;
        goto label;
      }
   }
phone :
   cout<<" \n\n\t\tEnter Phone : ";
    cin.getline(phone,25);
   string  c_ph ;
   c_ph=phone;
   while(c_ph.length() != 11)
   {
      cout<<"\n\t\tPhone number must be 11 digit\n";
      cout<<"\t\tplease press  'Enter key' to try Again' "<<endl ;
      goto phone;
   }
     int d;
   d=checking_existing_phone (c_ph);
   if(d == 1)
   {
      cout<<"\n\t\tThis Phone Number  is already Taken  "<<endl; ;
      cout<<"\t\tplease press  'Enter key' to try Again' "<<endl ;

      goto phone;
   }
   cout<<" \n\n\t\tEnter E-mail : ";


   cin.getline(mail_add,25);
   cout<<" \n\n\t\tEnter Home Address : ";
    cin.getline(h_address,25);
   cout<<" \n\n\t\tEnter Birthdate"<<endl;
   cout<<"\t\tDD/MM/YYYY :  ";
    cin.getline(dob,25);
     stud<< name<<' '<<phone<<' '<<mail_add<<' '<<h_address<<' '<<dob<<'\n';
    cout<< "\n\n\t\t\tdone !";
    cout<<"\n\t\tContact has been saved successfully !!!"<<endl;
    stud.close();
}
```

Show All Contact:

```
void contact :: show_all()

{

system("cls");

system("color b9");

cout<<"\n\t\t\t\t#####################################################
#######"<<endl;

 cout<<"\n\t\t\t\t#\t\tContact Management System\t\t#"<<endl;

cout<<"\n\t\t\t\t#####################################################
#######"<<endl;

   fstream stud;

   stud.open("contact.txt",ios::in);

   cout<<"\n\tName\t    Phone\t\tE-mail\t\t\tAddress\t\t\tBirthdate"<<endl;

   while(!stud.eof())

   {

      stud.getline(name,25,' ');

      stud.getline(phone,25,' ');

      stud.getline(mail_add,25,' ');

      stud.getline(h_address,25,' ');

      stud.getline(dob,25);

      if(stud.eof())

 {

         break;

      }

    cout<< "\n \t"<< name << "\t
"<<phone<<"\t\t"<<mail_add<<"\t"<<h_address<<"\t\t"<<dob<<"\n"<<endl;

 }

}
```

Search Contact:

```
void contact :: search_contact()

{

   system("cls");

   system("color b9");


cout<<"\n\t\t\t\t###################################################
#######"<<endl;

   cout<<"\n\t\t\t\t#\t\tContact Management System\t\t#"<<endl;

cout<<"\n\t\t\t\t###################################################
#######"<<endl;

   fstream stud;

   stud.open("contact.txt",ios::in);

  char a[25];

   cout<<" \n\t Enter the Name  of Contact to search about it : ";

   cin.ignore();

   cin.getline(a,25);

   int x=0;

 while(!stud.eof())

   {

       stud.getline(name,25,' ');

       stud.getline(phone,25,' ');

       stud.getline(mail_add,25,' ');

       stud.getline(h_address,25,' ');

       stud.getline(dob,25);

       if(strcmp(name, a)==0)

       {
```

```
cout<< "\n \t\tName\t\t: "<< name << "\n\t\tPhone\t\t: "<<phone<<"\n\t\tE-mail\t\t: "<<mail_add<<"\n\t\tHome Address\t: "<<h_address<<"\n\t\tDate of Birth\t: "<<dob<<"\n"<<endl;

        x=1;

           break;

        }

 }

   if(x==0)

   {

       cout<<" \n\n\n\t\t not found !!!!\n";

   }

   stud.close();

}
```

Update Contact:

```
void contact ::update_contact()

{

   system("cls");

   system("color b9");

cout<<"\n\t\t\t\t###################################################
#######"<<endl:

 cout<<"\n\t\t\t\t#\t\tContact Management System\t\t#"<<endl;


cout<<"\n\t\t\t\t###################################################
#######"<<endl;

fstream stud;

   fstream temp;

stud.open("contact.txt",ios::in);

   temp.open("temp.txt",ios::out);

   char a[25];

   cin.ignore();

   cout<<" \n\t\t\tEnter the Name  of Contact to update record : ";

   cin.getline(a,25);

   while(!stud.eof())

   {

       stud.getline(name,25,' ');

       stud.getline(phone,25,' ');

       stud.getline(mail_add,25,' ');

       stud.getline(h_address,25,' ');

       stud.getline(dob,25);

 if(strcmp(name,a)==0)

       {
```

```
    char choice ;
    fstream file_read;
     file_read.open("contact.txt", ios::app);
    cout<<"\n\t\tTell me what u Wnat to change "<<endl ;
    cout<<"\n\t\tFor Name  press          : 1";
    cout<<"\n\t\tFor Phone press          : 2";
    cout<<"\n\t\tFor Mail Address         : 3";
    cout<<"\n\t\tFor Home Address  press  : 4 ";
    cout<<"\n\t\tFor Birthdate  press     : 5";
    cout<<"\n\n\t\tEnter Choice : ";
    cin>>choice;
    string c_name, c_ph ;
    cin.ignore();
    int a=0;
    switch (choice)
    {
    case '1' :
      cout<<" \n\t\tEnter new value of record  \n ";
label :
      cout<<" \n\n\t\tName : ";

      cin.getline(name,25);
      if(name[0]>=97 && name[0] <= 122)
      {
        name[0] = name[0]-32;
        string c_name ;
        c_name= name ;
```

```
                  int a ;
                  a =checking_existing_name( c_name);
                  if(a==1)
                  {
                      cout<<"This Name   is already Taken  "<<endl; ;
                      cout<<"please try Again' "<<endl ;
                     goto label;
                  }
               }
               else
               {
                  string c_name ;
                  c_name= name ;
                  int a ;
                  a =checking_existing_name( c_name);
                  if(a==1)
                  {
                      cout<<"This Name   is already Taken “<<endl;
                      cout<<"please try Again' "<<endl ;
                       goto label;
                  }
               }
temp<<name<<' '<<phone<<' '<<mail_add<<' '<<h_address<<' '<<dob<<'\n';
        break ;


           case '2':
phone :
    cout<<" \n\n\tPhone : ";
    cin.ignore();
    cin.getline(phone,25);
     c_ph = phone;
     int d;
    d=checking_existing_phone (c_ph);
    if(d == 1)
    {
        cout<<"\n\t\tThis Phone Number  is already Taken  "<<endl; ;
        cout<<"please press  'Enter key' to try Again' "<<endl ;
         goto  phone ;
      }
    temp<<name<<' '<<phone<<' '<<mail_add<<' '<<h_address<<' '<<dob<<'\n';
    break ;
case '3':
    cout<<" \n\n\tMail Address : ";
    cin.ignore();
    cin.getline(mail_add,25);
    temp<<name<<' '<<phone<<' '<<mail_add<<' '<<h_address<<' '<<dob<<'\n';
    break;
case '4' :
    cout<<" \n\n\tHome Address : ";
    cin.ignore();
    cin.getline(h_address,25);
    temp<<name<<' '<<phone<<' '<<mail_add<<' '<<h_address<<' '<<dob<<'\n';
    break;
case '5' :
```

```
            cout<<" \n\n\t\t Date of Birth : ";
            cin.ignore();
            cin.getline(dob,25);
            temp<<name<<' '<<phone<<' '<<mail_add<<' '<<h_address<<' '<<dob<<'\n';
            break;
        default:
        cout << "Error! operator is not correct";
            break;
        }
    }
    else
    {
        temp<<name<<' '<<phone<<' '<<mail_add<<' '<<h_address<<' '<<dob<<'\n';
    }
}
temp.close();
stud.close();
stud.open("contact.txt",ios::out);
temp.open("temp.txt",ios::in);
while(!temp.eof())
{
    temp.getline(name,25,' ');
    temp.getline(phone,25,' ');
    temp.getline(mail_add,25,' ');
    temp.getline(h_address,25,' ');
    temp.getline(dob,25);
    stud<<name<<' '<<phone<<' '<<mail_add<<' '<<h_address<<' '<<dob<<'\n';
```

```
        if(temp.eof())
        {
            remove("temp.txt");
            temp.close();
            break;
        }
    }
    stud.close();
    cout<<"\n done !!! \n";
}
```

Delete Contact:

```
void contact :: delete_contact()
{
   system("cls");
   system("color b9");
cout<<"\n\t\t\t###################################################
#######"<<endl;
   cout<<"\n\t\t\t\t#\t\tContact Management System\t\t#"<<endl;
cout<<"\n\t\t\t###################################################
#######"<<endl;
   fstream stud;
   fstream temp;
   stud.open("contact.txt",ios::in);
   temp.open("temp.txt",ios::out);
   char a[25];
   cin.ignore();
   cout<<" \n\n\n\t\t enter the Name  of Contact to delete record : ";
   cin.getline(a,25);
   while(!stud.eof())
   {
      stud.getline(name,25,' ');
      stud.getline(phone,25,' ');
      stud.getline(mail_add,25,' ');
      stud.getline(h_address,25,' ');
      stud.getline(dob,25);
      if(strcmp(name,a)==0)
      {
            continue;
         }
         else
         {
            temp<<name<<' '<<phone<<' '<<mail_add<<' '<<h_address<<' '<<dob<<'\n';
         }
      }
      temp.close();
      stud.close();
      stud.open("contact.txt",ios::out);
      temp.open("temp.txt",ios::in);
      while(!temp.eof())
      {
         temp.getline(name,25,' ');
         temp.getline(phone,25,' ');
         temp.getline(mail_add,25,' ');
         temp.getline(h_address,25,' ');
         temp.getline(dob,25);
         stud<<name<<' '<<phone<<' '<<mail_add<<' '<<h_address<<' '<<dob<<'\n';
      }
      temp.close();
      stud.close();
      remove("temp.txt");
      cout<<"\n done !!! \n";
   }
```

Main Menu:

```
int main()
{
    contact ob;
    string User = "admin";
    string pass= "1234";
    bool flag=true;
    string User_name;
    string Passcode;
label :
    system("cls");
    system("color b9");
    cout<<"\n\n\t\t\t\t\t\t***** WellCOME *****\n\t\t\t\t\t To our Contact Management System"<<endl;
cout<<"\t\t==============================================================="<<endl;
    cout<<"\n\n\t\tPlease, sign up with your User Name and Password :"<< endl;
    cout<<"\n\n\t\t Enter Your User Name : ";
    cin>>User_name;
    cout<<"\n\t\t Enter Your User Password : ";
    cin>>Passcode;
    cout<<"\n\n\tThank You, Sir, \n\t\t  for using our Contact Management System "<<endl;
    if(User_name == User && Passcode == pass)
    {
        while (flag)
        {
            system("cls");
            system("color 3f");
```

```
cout<<"\n\t\t\tooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooo
"<<endl;

        cout<<"\n\t\t\tooooo\t\t\tCONTACT MANAGEMENT SYSTEM\t\tooooooo"<<endl;


cout<<"\n\t\t\tooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooo
"<<endl;

        cout<<"\n\n\tThank You, Sir, \n\t\t  For using our Contact Management System "<<endl;

        cout << "\n\n\t\t\t Create a Contact\t: [1]\n\t\t\t Search Contact\t\t: [2] \n\t\t\t Display all
Contact\t: [3] \n\t\t\t Update a Contact\t: [4] \n\t\t\t Delete a Contact\t: [5] \n\n\n\t\t\t write
your choose\t: ";

        int c;

        cin>>c;

        switch (c)

        {

        case 1 :

          ob.insert_contact();

          break;

        case 2:

          ob.search_contact();

          break;

        case 3:

          ob.show_all();

          break;

        case 4 :

          ob.update_contact();

          break;

        case 5 :

          ob.delete_contact();

          break;

        case 6 :
```

```
      default :
        cout << " \n choose right number \n";
      }
      cout << "\n if you continue to use program press ' y ' \n ";
      char f;
      cin>>f;
      if(f!='y')
      {
        flag=false;
      }
    }
  }
  else
  {
    goto label;
  }
  return 0;
}
```

CONCLUSION:

Program Weakness:

As we know that, there is no program which can be 100% reliable and efficient. So there are also some drawbacks from our system which are as follows:

- *It cannot perform all the required functions as Contact Management required, it's simply a record of contacts.*
- *System is not sharply a graphical user interface. There is just use of some text color & Background color.*
- *It's not a multiuser and multitasking program. It can't perform various task at a single time.*
- *It's only an Offline program*

Program Strength:

There are many advantages of using this program as it contains various features like:

- *It is actually a user friendly software, as it is just easy to use by just following the instruction which are appeared on the screen.*
- *This program needs user password to access user information, so that only authorized users are only allowed to accessed through the internal main system.*
- *Once user a save a data it can't be duplicated.*
- Manual editing of data is not authorized.
- *It was cool visual color and easy to use up.*
- *It can be used in many computers as it doesn't need powerful computers.*

Future Scope

We would like to expend our application to further more. we would like to implement complicated data structure. Also want to use sql database method to store data in table method. As we want to make it online base banking system, we will implement script language in it, to access over the air. And finally if we have proper funding we would like to make a fresh GUI for the application   to achieve the fresh look.

REFERENCE

*1. **Wikipedia***
*2. **www.stackoverflow .com***
*3. **cppforschool.com***
*4. **google.com***
*5.**www.youtube.com***
*6.**www.quora.com***
*7. **Teach yourself c++ by Herbert Schildt***