**KATHFORD INTERNATIONAL COLLEGE OF ENGINEERING AND MANAGEMENT**

**Balkumari, Lalitpur**

**Affiliated to Tribhuvan University**

**Institute of Science and Technology**

**A Project Report on**

**CONTACT MANAGER**

**Submitted to**

**Department of Computer Science and Information Technology**

**Kathford International College of Engineering and Management**

***In Partial Fulfillment for the third semester course Object Oriented Programming (CSC-202) of Bachelor of Science in Computer Science and Information Technology (B.Sc.CSIT)***

**Under Supervision of**

Mr. Bhes Bahadur Thapa

(Lecturer, B.Sc.CSIT, Kathford College)

**Submitted By:**

Ashish Shrestha (5)

Bishal Thapa Shrestha (11)

Namkong Hang Kirat (21)

Neha Adhikari (22)

March 2017

**Kathford International College of Engineering and Management Balkumari, Lalitpur**

**LETTER OF APPROVAL**

This is to certify that this project prepared by Ashish Shrestha, Bishal Thapa Shrestha, Namkong Hang Kirat, Neha Adhikari entitled “**CONTACT MANAGER**” in partial fulfillment of the requirements for the third semester course **Object Oriented Programming (CSC-202)** of B.Sc. in Computer Science and Information Technology has been well studied. In our opinion it is satisfactory in the scope and quality as a project for the required subject.

………….………………. ………….……………….

Project Supervisor HoD of Computer Science and Technology

Kathford International College of Kathford International College of

Engineering and Management Engineering and Management

………………………….. …………………………..

SIGNATURE SIGNATURE

External Examiner Internal Examiner

**Kathford International College of Engineering and Management**

**Balkumari, Lalitpur**

**Supervisor’s Recommendation**

I hereby recommend that this project prepared under my supervision by Ashish Shrestha, Bishal Thapa Shrestha, Namkong Hang Kirat, Neha Adhikari entitled “**CONTACT MANAGER**” in partial fulfillment of the requirements for the third semester course **Object Oriented Programming (CSC-202)** of B.Sc. in Computer Science and Information Technology be processed for evaluation.

………………………...

Project Supervisor

Kathford International College of

Engineering and Management.

**ABSTRACT**

“CONTACT MANAGER” is a simple project for managing contact details. It can be used by any organization and individual to save important contact information of personnel, secured in a database.

Different contact management system has been made digital through several programming appeals today in order to make the system easy, efficient and comprehensible.

So we have thought of designing this project, **CONTACT MANAGER** is based on C++ language. With knowledge of unambiguous syntax commonly used in C++ will make this project completely understandable and easy to implement.

**ACKNOWLEDGMENT**

We are very much thankful to the Department of Computer Science and Information Technology, Kathford International College of Engineering and Management for accepting our report proposal on “CONTACT MANAGER” and granting us the opportunity of work undertaken. We express our earnest gratitude for providing us with all essential cooperation, valuable suggestion and the necessary resources during all the quarters of the completion of the report.

For the accomplishment of this report, a number of people have become a part of it. Firstly, we are deeply indebted to our Supervisor and HoD of CSIT for their kind support, coordination and valuable supervision from starting to end of this report without which this report could not come in this form.

We are also thankful to our friends who have contributed directly or indirectly in the accomplishment of our report.

We would also like to acknowledge and extend our gratitude to everyone for his/her support and encouragement.

**CONTENTS**

**Contents Page Number**

INTRODUCTION 1 OBJECTIVES AND METHODOLOGY 2 SCOPE AND LIMITATIONS 3 IMPLEMENTATION

1. Algorithm 4-6
2. Flowchart 7
3. Header Files 8
4. Class 8
5. User-defined functions 8

CONCLUSION AND ENHANCEMENTS 9

APPENDICES 10-14

SOURCE CODE 15-36

REFERENCES AND BIBLIOGRAPHY

**Contents**

[INTRODUCTION 1](#_Toc476989323)

[OBJECTIVES AND METHODOLOGY 2](#_Toc476989324)

[SCOPE AND LIMITATIONS 3](#_Toc476989325)

[IMPLEMENTATION 4](#_Toc476989326)

[Algorithm 4](#_Toc476989327)

[Flowchart 7](#_Toc476989328)

[Header files 8](#_Toc476989329)

[Class 8](#_Toc476989330)

[User-defined functions 8](#_Toc476989331)

[CONCLUSION AND ENHANCEMENTS 9](#_Toc476989332)

[APPENDICES 10](#_Toc476989333)

[Source code 14](#_Toc476989334)

# INTRODUCTION

**CONTACT MANAGER**

This is a simple project to manage contact details. It can be used by any organization and individual to save important contact information of personnel, secured in a database. This will help any organization and individual to manage contacts accordingly.

Contact Management System is a program based on C++ programming. This project uses file handling methodologies to store contact information of people in a local file. This program is console based. The program uses Command Line Interface (CLI) to interact with the user. Even though CLI is not much user friendly, we have made this program as simple as possible so as to improve the user experience.

The aim of this project is to implement a contact management system which is easy to use and is suitable for every individual and organization to manage contact details.

**PROBLEM STATEMENT**

Managing contact is difficult using traditional methods. Using a manual system is challenging as the records can be scattered, can be redundant and may be very time consuming. All these problems are solved using this project. Throughout the project the focus has been on keeping records in an easy and intelligible manner.

“Contact manager” is an automated version of manual contact details and using this software means eliminating the tedious physical transaction records on paper and making it digital and easier. The main objective of this program is to store contact information in a convenient manner so that it can be retrieved anytime hence solving the problem.

# OBJECTIVES AND METHODOLOGY

**OBJECTIVE**

General:

* The main purpose of this project is to help any user to manage contact details.
* It will facilitate keeping all the records of contacts as Name, Landline number and Cell phone number and email address.
* So, all the contact details will be managed and can be easily available when needed.

Specific:

* To allow the user to enter and save the contact details and to also allow the user to edit those records.
* Deleting the pre-recorded information.

**METHODOLOGY**

We used

* Reference book
* Web surfing

# SCOPE AND LIMITATIONS

**SCOPE**

This system makes easy to record contact details. It provides capabilities for adding contacts details such as name, landline number, mobile number and email address, listing contacts, searching contacts, editing contacts and deleting contacts. This will help an individual to manage his contact accordingly. Further in this project we can add other features as well.

**LIMITATION**

We have done our best to make our project eligible for every kind of purpose but also there are some limitations that we were unable to handle. Some of the limitations are listed below:

* Contact details are executed in off-line mode.
* Off-line information cannot be generated.
* The existing system only provides text-based interface, which is not as user-friendly as Graphical user Interface.
* This program is a console app hence cannot be used in any devices as an app.

# IMPLEMENTATION

## Algorithm

1. Main function

Step 1: Start

Step 2: Display choices

Step 3: Ask user for choice

Step 4: switch (choice)

Case 0: Exit from program

Case 1: Go to Add function add()

Case 2: Go to list function display()

Case 3: Go to Search function \_search()

Case 4: Go to Edit function edit()

Case 5: Go to Delete function \_delete()

Default: Display “Enter a valid choice”

Step 5: Stop

1. Add Function add()

Step 1: Start

Step 2: Open data.txt in append mode

Step 3: Ask name for new data, landline number, cellphone number and email address and store in structure

Step 4: If user enters space only, then goto main menu

Else, store data in data.txt

Step 5: Close data.txt

Step 6: Stop

1. List Function display()

Step 1: Start

Step 2: Open data.txt in read mode.

Step 3: Display data.

Step 4: Wait for user to conform exit.

Step 5: Close data.txt

Step 6: Stop

1. Search Function \_search()

Step 1: Start

Step 2: Open data.txt in read mode.

Step 3: Receive name from user.

Step 4: If name matches display contacts

Else show error message.

Step 5: Wait for user to conform exit.

Step 6: Close data.txt

Step 7: Stop

1. Edit Function edit()

Step 1: Start

Step 2: Open data.txt in read mode and temp.txt in write mode.

Step 3: Receive name from user.

Step 4: Check whether given name matches data in data.txt.

If yes, ask for new data and write it in temp.txt

If no, write data from data.txt in temp.txt

Step 5: Close both files.

Step 6: Delete data.txt and rename temp.txt to data.txt.

Step 7: Stop

1. Delete Function \_delete()

Step 1: Start

Step 2: Open data.txt in read mode and temp.txt in write mode.

Step 3: Receive name from user.

Step 4: Check whether given name matches data in data.txt.

If yes, skip the data

If no, write data from data.txt in temp.txt

Step 5: Close both files.

Step 6: Delete data.txt and rename temp.txt to data.txt.

Step 7: Stop

## Flowchart

Start

Display Choices

Ask user for choice

Switch   
Choice

Goto List function

Goto Search function

Display “Enter a Valid Choice”

Stop

Goto Add function

Goto Edit function

Choice=1

Choice=2

Choice=3

Choice=4

Choice=0

default

## Header files

The header files used in the program are:

#include <iostream>

#include <string.h>

#include <windows.h>

#include <conio.h>

#include <fstream>

## Class

Two Class are used in the program:

* Class Draw
* Class People

## User-defined functions

* Void display
* Void add
* Int \_find
* Void read
* Void write
* Void \_delete
* Void \_search
* Void edit
* Void box
* Void divide\_horizontal

# CONCLUSION AND ENHANCEMENTS

**Conclusion**

From this project we are able to program in a detailed way by evaluating our skills and this has boost up our confidence for future projects. This report is not quite so successful but it will surely make the user to use this system which is quite user friendly

* Helpful to perform paperless work and manage all data.
* Provides easy, accurate, unambiguous and faster data access.

**Enhancement**

Contact Manager can be further enhanced to add other functionality also. Some of them are listed below:

* Make updates in design and features
* Provision of service to mobile phones with other mobile OS.
* Online accessible

# APPENDICES

**Snapshots**

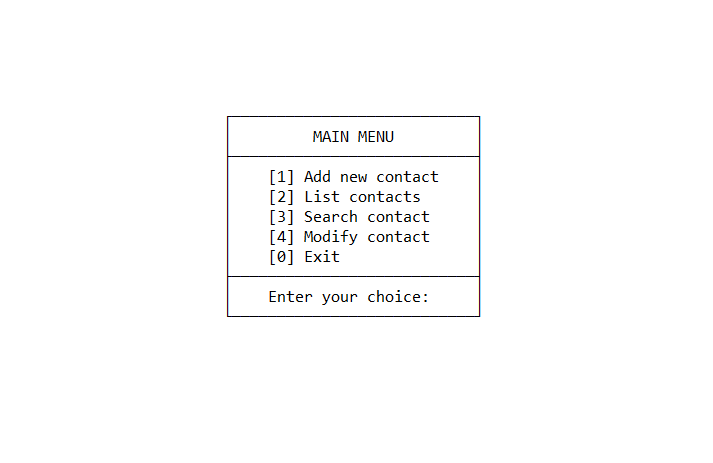
****

Fig: Main menu

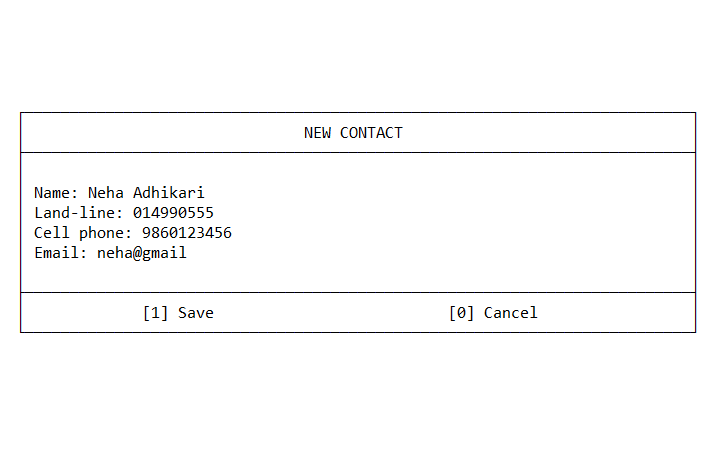
****

Fig: Add contacts

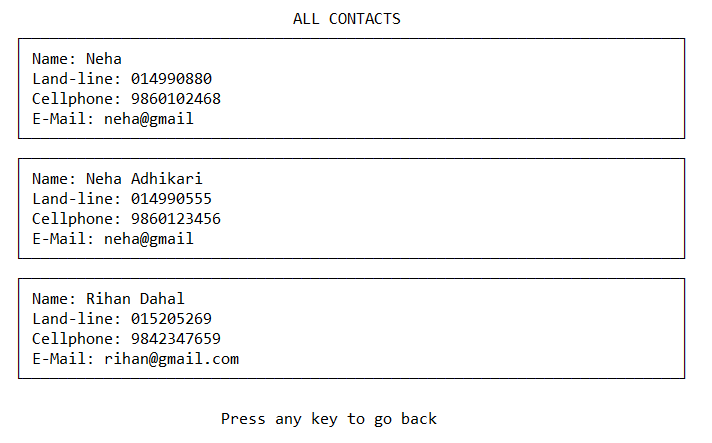
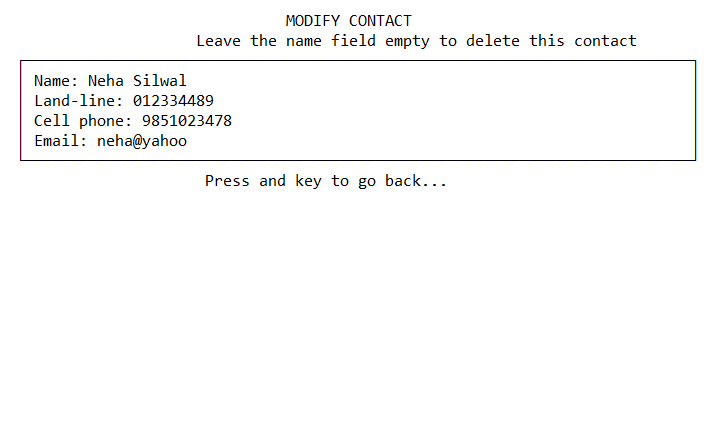
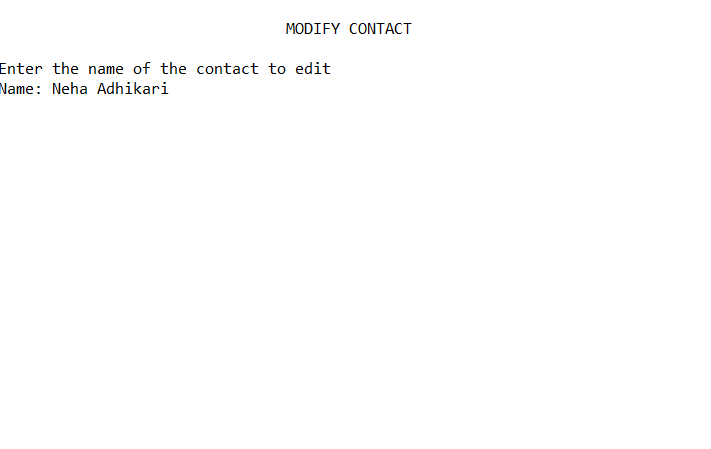
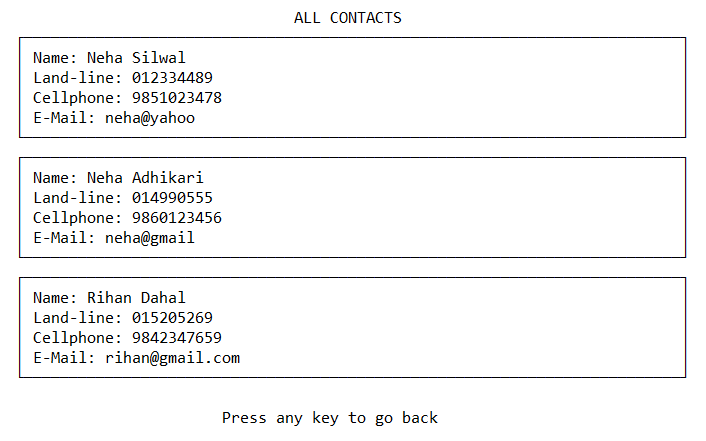
****Fig: List contacts****Fig: Modifying contacts****

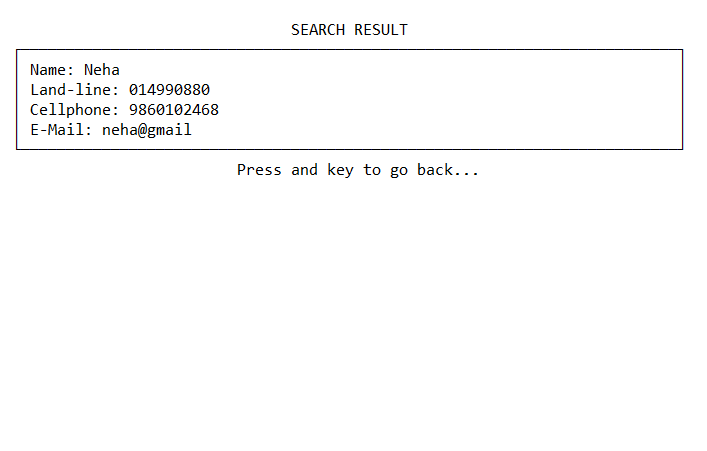
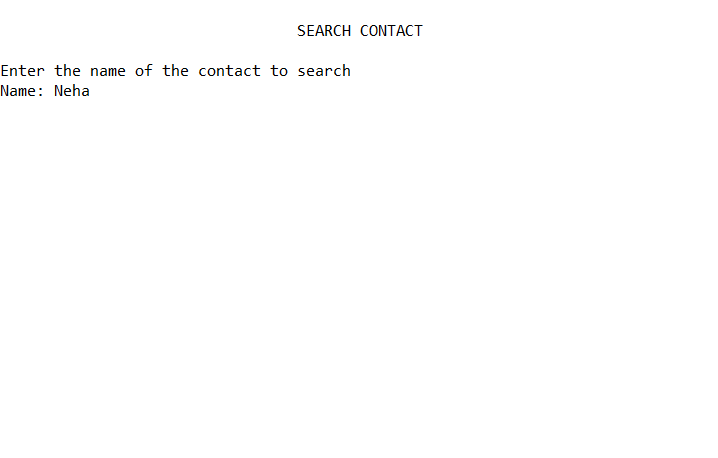
Fig: Listing contacts after modifying****

Fig: Searching contacts

# Source code

#include <iostream>

#include <string.h>

#include <conio.h>

#include <windows.h>

#include <fstream>

using namespace std;

class Draw {

public:

void box (int x, int y, int h, int w) {

gotoxy (x, y); //top

cout<<(char)(-38);

for(int i = 1; i <= (w - 2); i++) {

cout<<(char)(-60);

}

cout<<(char)(-65);

gotoxy(x, y + h - 1); //bot

cout<<(char)(-64);

for(int i = 1; i <= (w - 2); i++) {

cout<<(char)(-60);

}

cout<<(char)(-39);

gotoxy(x, y+1); //left vertical

for(int i = 0; i < h - 2; i++) {

gotoxy(x, (y + 1 + i));

cout<<(char)(-77)<<endl;

}

gotoxy(x + w - 1, y + 1); //right vertical

for(int i = 0; i < h - 2; i++) {

gotoxy(x + w - 1, (y + 1 + i));

cout<<(char)(-77)<<endl;

}

}

void divide\_horizontal (int x, int y, int w) {

gotoxy(x, y);

cout<<(char)(-61);

for (int i = 0; i < w-2; i++) {

cout<<(char)(-60);

}

cout<<(char)(-76);

}

void gotoxy(int x, int y) {

COORD coordinate;

coordinate.X = x;

coordinate.Y = y;

SetConsoleCursorPosition(GetStdHandle(STD\_OUTPUT\_HANDLE), coordinate);

}

}draw;

class People{

private:

string landline, cell, name, email;

bool valid;

public:

void display(int contact, int location) {

read (contact);

draw.box(2, (location \* 6) - 4, 6, 75);

draw.gotoxy(4, (location \* 6) - 3);

cout<<"Name: -\b";

cout<<name<<endl;

draw.gotoxy(4, (location \* 6) - 2);

cout<<"Land-line: -\b";

cout<<landline<<endl;

draw.gotoxy(4, (location \* 6) - 1);

cout<<"Cellphone: -\b";

cout<<cell<<endl;

draw.gotoxy(4, (location \* 6));

cout<<"E-Mail: -\b";

cout<<email<<endl;

}

int \_find(string \*querry){

ifstream file;

string buffer;

int location;

bool match = false;

file.open("data.txt");

if (!file.is\_open()) {

draw.gotoxy(2,1);

cout<<"Error accessing data file.\n";

getch();

exit(-1);

}

location = 0;

while(!file.eof() && !match) {

getline (file, buffer);

if (buffer.substr (0, 5) == "BEGIN") {

location++;

while (buffer.substr (0, 3) != "END") {

getline(file, buffer);

if (buffer.substr (0, 4) == "name" && \*querry == buffer.substr (5, buffer.length() - 6)) {

match = true;

break;

}

}

}

}

if (!match) {

return -1;

}

else {

return location;

}

file.close();

}

void add () {

system ("cls");

int y = 5;

draw.box(2, y, 12, 76);

draw.divide\_horizontal(2, y+2, 76);

draw.divide\_horizontal(2, y+9, 76);

draw.gotoxy (34, y+1);

cout<<"NEW CONTACT";

draw.gotoxy (4, y+4);

cout<<"Name: -";

draw.gotoxy (4, y+5);

cout<<"Land-line: -";

draw.gotoxy (4, y+6);

cout<<"Cell phone: -";

draw.gotoxy (4, y+7);

cout<<"Email: -";

draw.gotoxy (16, y+10);

cout<<"[1] Save";

draw.gotoxy (50, y+10);

cout<<"[0] Cancel";

valid=false;

while (!valid) {

draw.gotoxy(10, y+4);

valid = false;

{

char temp[70];

ifstream file;

string buffer;

file.open("data.txt");

if (!file.is\_open()) {

cout<<"Error accessing data file.\n";

getch();

exit(-1);

}

cin.getline(temp, 70);

name=(string)temp;

cin.sync();

while(!file.eof()) {

getline (file, buffer);

if(buffer.substr(0, 4)=="name") {

if(name == buffer.substr (5, buffer.length() - 6)) {

draw.gotoxy (10, y+4);

for (unsigned int i = 0; i < name.length(); i++) {

cout<<" ";

}

draw.gotoxy(62, y+4);

cout<<"Already exists!";

draw.gotoxy (10, y+4);

cout<<"-";

valid = false;

break;

}

else {

valid = true;

}

}

}

file.close();

}

}

draw.gotoxy(15, y+5);

{

char temp[70];

cin.getline(temp, 70);

landline=(string)temp;

cin.sync();

}

draw.gotoxy(16, y+6);

{

char temp[70];

cin.getline(temp, 70);

cell=(string)temp;

cin.sync();

}

valid=false;

while (!valid) {

draw.gotoxy(11, y+7);

valid = false;

{

char temp[70];

cin.getline(temp, 70);

email=(string)temp;

cin.sync();

}

int count = 0;

for (int unsigned i = 0; i < email.length(); i++) {

if(email.substr(i,1) == "@") {

count++;

}

}

if (count != 1 && email.length() != 0) {

draw.gotoxy(11, y+7);

int temp = email.length();

while (temp > 0) {

cout<<" ";

temp--;

}

draw.gotoxy(11, y+7);

cout<<"-";

draw.gotoxy(73, y+7);

cout<<"!!!";

}

else {

valid = true;

draw.gotoxy(75, y+7);

cout<<" ";

}

}

if (name.length() != 0) {

bool loop = true;

while (loop) {

draw.gotoxy (51, y+10);

{

char ch;

ch = getch();

switch (ch) {

case '1':

write("data.txt");

loop = false;

break;

case '0':

loop = false;

break;

}

}

}

}

}

unsigned int how\_many () {

unsigned int total = 0;

string dump;

ifstream file;

file.open ("data.txt");

if (!file.is\_open()) {

cout<<"Error accessing data file.\n";

getch ();

exit (-1);

}

while (!file.eof()) {

getline(file, dump);

total++;

}

file.close();

return total / 6;

}

void read (unsigned int contact) {

ifstream file;

string buffer;

file.open("data.txt");

if (!file.is\_open()) {

cout<<"Error accessing data file.\n";

getch();

exit(-1);

}

if(!file.eof()) {

for (int i = (contact \* 6) - 6; i > 0; i--) {

getline (file, buffer);

}

getline (file, buffer);

if (buffer.substr (0, 5) == "BEGIN") {

while (buffer.substr (0, 3) != "END") {

getline(file, buffer);

if (buffer.substr (0, 4) == "name") {

name = buffer.substr (5, buffer.length() - 6);

}

else if (buffer.substr (0, 4) == "land") {

landline = buffer.substr (5, buffer.length() - 6);

}

else if (buffer.substr (0, 4) == "cell") {

cell = buffer.substr (5, buffer.length() - 6);

}

else if (buffer.substr (0, 4) == "mail") {

email = buffer.substr (5, buffer.length() - 6);

}

}

}

}

file.close();

}

void write (char \*filename) {

ofstream file;

file.open (filename, ios::app);

if (!file.is\_open ()) {

cout<<"Error accessing data file.\n";

getch();

exit (-1);

}

file<<"BEGIN"<<endl;

file<<"name:"<<name<<";"<<endl;

file<<"land:"<<landline<<";"<<endl;

file<<"cell:"<<cell<<";"<<endl;

file<<"mail:"<<email<<";"<<endl;

file<<"END"<<endl;

file.close();

}

void \_delete () {

system ("cls");

string tempn;

ifstream file;

string buffer;

file.open("data.txt");

if (!file.is\_open()) {

cout<<"Error accessing data file.\n";

getch();

exit(-1);

}

draw.gotoxy (33, 1);

cout<<"DELETE CONTACT";

cout<<"\n\nEnter the name of the contact to delete";

cout<<"\nName: -\b";

cin>>tempn;

}

void \_search () {

system("cls");

string tempn;

int location;

draw.gotoxy (33, 1);

cout<<"SEARCH CONTACT";

cout<<"\n\nEnter the name of the contact to search";

cout<<"\nName: -\b";

cin>>tempn;

cin.sync();

system ("cls");

draw.gotoxy (33, 1);

cout<<"SEARCH RESULT";

location = \_find(&tempn);

if (location == -1) {

draw.gotoxy(32, 3);

cout<<"No match found!!";

draw.gotoxy(27, 4);

}

else {

display (location, 1);

draw.gotoxy(27, 8);

}

cout<<"Press and key to go back...";

getch ();

}

void edit () {

system("cls");

string tempn;

int location;

draw.gotoxy (32, 1);

cout<<"MODIFY CONTACT";

cout<<"\n\nEnter the name of the contact to edit";

cout<<"\nName: -\b";

cin>>tempn;

cin.sync();

system ("cls");

draw.gotoxy (32, 1);

cout<<"MODIFY CONTACT";

location = \_find(&tempn);

if (location == -1) {

draw.gotoxy(32, 3);

cout<<"No match found!";

draw.gotoxy(27, 4);

}

else {

int y = 3;

draw.gotoxy (22, 2);

cout<<"Leave the name field empty to delete this contact";

draw.box(2, y, 6, 76);

draw.gotoxy (4, y+1);

cout<<"Name: -";

draw.gotoxy (4, y+2);

cout<<"Land-line: -";

draw.gotoxy (4, y+3);

cout<<"Cell phone: -";

draw.gotoxy (4, y+4);

cout<<"Email: -";

valid=false;

while (!valid) {

draw.gotoxy(10, y+1);

valid = false;

{

char temp[70];

ifstream file;

string buffer;

file.open("data.txt");

if (!file.is\_open()) {

cout<<"Error accessing data file.\n";

getch();

exit(-1);

}

cin.getline(temp, 70);

name=(string)temp;

cin.sync();

while(!file.eof()) {

getline (file, buffer);

if(buffer.substr(0, 4)=="name") {

if(name == buffer.substr (5, buffer.length() - 6)) {

draw.gotoxy (10, y+1);

for (unsigned int i = 0; i < name.length(); i++) {

cout<<" ";

}

draw.gotoxy(62, y+1);

cout<<"Already exists!";

draw.gotoxy (10, y+1);

cout<<"-";

valid = false;

break;

}

else {

valid = true;

}

}

}

file.close();

}

}

draw.gotoxy(15, y+2);

{

char temp[70];

cin.getline(temp, 70);

landline=(string)temp;

cin.sync();

}

draw.gotoxy(16, y+3);

{

char temp[70];

cin.getline(temp, 70);

cell=(string)temp;

cin.sync();

}

valid=false;

while (!valid) {

draw.gotoxy(11, y+4);

valid = false;

{

char temp[70];

cin.getline(temp, 70);

email=(string)temp;

cin.sync();

}

int count = 0;

for (int unsigned i = 0; i < email.length(); i++) {

if(email.substr(i,1) == "@") {

count++;

}

}

if (count != 1 && email.length() != 0) {

draw.gotoxy(11, y+4);

int temp = email.length();

while (temp > 0) {

cout<<" ";

temp--;

}

draw.gotoxy(11, y+4);

cout<<"-";

draw.gotoxy(73, y+4);

cout<<"!!!";

}

else {

valid = true;

draw.gotoxy(75, y+4);

cout<<" ";

}

}

{

fstream tempfile;

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

tempfile.close();

}

if (name.length() != 0) {

write ("temp.txt");

}

for (unsigned int i = 1; i <= how\_many(); i++) {

if (i == location) {

continue;

}

else {

read (i);

write ("temp.txt");

}

}

system ("del /f /q data.txt");

system ("ren temp.txt data.txt");

}

draw.gotoxy(23, 9);

cout<<"Press and key to go back...";

getch ();

}

}p;

int main () {

int ch;

system ("COLOR f0");

system ("title Contact Manager");

while (true) {

system ("cls");

system ("mode con:cols=80 lines=23");

draw.box (25, 5, 11, 29);

draw.gotoxy (35, 6);

cout<<"MAIN MENU";

draw.divide\_horizontal(25, 7, 29);

draw.gotoxy(30, 8);

cout<<"[1] Add new contact";

draw.gotoxy (30, 9);

cout<<"[2] List contacts";

draw.gotoxy (30, 10);

cout<<"[3] Search contact";

draw.gotoxy (30, 11);

cout<<"[4] Modify contact";

draw.gotoxy (30, 12);

cout<<"[0] Exit";

draw.divide\_horizontal (25, 13, 29);

draw.gotoxy (30, 14);

cout<<"Enter your choice: ";

ch = getch();

switch (ch) {

case '1':

p.add();

break;

case '2':

system("cls");

if ((6\*p.how\_many())+4 > 30) {

char script[100];

sprintf (script, "mode con:cols=80 lines=%d", (6\*p.how\_many())+5);

system (script);

}

draw.gotoxy(33,1);

cout<<"ALL CONTACTS";

for(unsigned int i=1; i <= p.how\_many();i++) {

p.display(i,i);

}

draw.gotoxy (25, (6\*p.how\_many())+3);

cout<<"Press any key to go back"<<endl;

draw.gotoxy (0, 0);

getch();

break;

case '3':

p.\_search();

break;

case '4':

p.edit ();

break;

case '0':

cout<<endl<<endl;

exit (0);

break;

default:

draw.gotoxy (29, 14);

cout<<"Enter a valid number!";

getch ();

}

}

}

**REFERENCES AND BIBLIOGRAPHY**

**References**

**Bibliography**

* <https://www.tutorialspoint.com/cplusplus/cpp_files_streams.htm>
* <http://www.theasciicode.com.ar/>
* <http://www.cplusplus.com>