



Dr. D. Y. PATIL VIDYAPEETH, PUNE
(Deemed to be University)

DR. D. Y. PATIL SCHOOL OF SCIENCE AND TECHNOLOGY
TATHAWADE, PUNE

Mini- Project Report on

--- CONTACT BOOK ---

A Contact Management System Based on C-Language

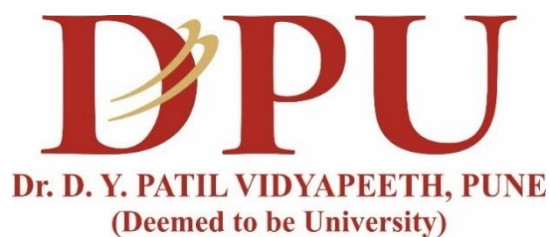
SUBMITTED BY:

NAME OF STUDENT	ROLL NUMBER
1. MUDIT JAIN	BTCSD-27
2. PIYUSH KATOLE	BTCSD-29
3. GOURAV KAR	BTCSD-20

GUIDED BY:

Mrs. Akanksha Goel

B TECH COMPUTER SCIENCE & DESIGN
ACADEMIC YEAR 2022-2023



DR. D. Y. PATIL SCHOOL OF SCIENCE AND TECHNOLOGY

TATHAWADE, PUNE

CERTIFICATE

This is to certify that the Mini- Project Report entitled.

-“--- CONTACT BOOK ---“-

A Contact Management System Based on C-Language

is a bonafide work carried out by **Mr. MUDIT JAIN** under the supervision of **Mrs. Akanksha Goel** and it is submitted towards the partial fulfillment of the requirement Project Based Learning-I.

Mrs. Akanksha Goel
Project Guide

Prof. Manisha Bhende
HOD (DYPSST)

B TECH COMPUTER SCIENCE & DESIGN
ACADEMIC YEAR 2022-2023

ABSTRACT

“CONTACT BOOK” a Contact Management System in C programming language is a Software Application designed to help people managing their Contacts and Communication with them, which allows users to Store and Organize Contact Information, including Names, Phone Numbers, Email Addresses, and other details. It includes features such as Contact Databases, Address Books developed by C programming language. It allows user to access the Emergency Contacts like Police, Ambulance, Fire Station, etc. which are useful at the time of Emergency Situations. It allows users to add, edit and Set Reminders, for important events and search for contacts. It also allows user to make NOTES for Individual Contacts. Also, the user can make a Quick Dial list for Important Contacts, which includes all the Important Contacts imported by User from Contact list.

INDEX

SR. NO.	TOPIC	PAGE NO.
1]	INTRODUCTION	8
	1.1 Objectives	8
	1.2 Problem Statements	8
2]	DESIGN	9-10
	2.1 System Architecture	9
	2.2 Use Case Diagram	10
3]	ALGORITHMS & FLOWCHARTS	11-12
4]	IMPLEMENTATION OF CODE	13-25
5]	OUTPUT	26-35
6]	CONCLUSION	36
7]	REFERENCES	37

LIST OF FIGURES

SR.NO	FIGURE NO.	PAGE NO
1 [2.1]	SYSTEM ARCHITECTURE	9
2 [2.2]	SYSTEM FEATURES	9
3 [2.3]	SYSTEM FEATURES	10
4 [2.4]	USE CASE DIAGRAM	10
5 [3.1]	FLOW CHART	12
6 [5.1]	WELCOME WINDOW	26
7 [5.2]	REGISTRATION	26
8 [5.3]	LOGIN	27
9 [5.4]	USER INTERFACE	27
10 [5.5]	ADD NEW CONTACTS	28
11 [5.6]	VIEW ALL CONTACTS	29
12 [5.7]	SEARCH CONTACT	30
13 [5.8]	DELETE CONTACTS	31
14 [5.9]	ENERGENCY CONTACTS	32
15 [5.10]	QUICK DIAL	33
16 [5.11]	LOGOUT	34
17 [5.12]	EXIT	

Chapter-1: INTRODUCTION

The Mini-Project 'CONTACT BOOK' is the sample project for managing contact details with many other info. Together. The Project Aims to develop Contact Management System using C Language that enable a person to maintain its Library.

There are so many Contacts Management System Application available in market, but what are the Unique Point Which make CONTACT BOOK Different from Other Contact Management System, the following points are described below:

1. **EMERGENCY CONTACTS:** Lots of same Contact system software's have only add, edit, view options but CONTACT BOOK also provides Emergency Contact Numbers like Police, Ambulance, Fire Station, etc.
2. **QUICK DIAL:** This Feature is the feature which make your work faster, this helps user to dial contacts faster by adding important contacts in the priority List, which is again most of applications no providing but CONTACT BOOK do.
3. **ADD NOTES:** This feature helps User to add Notes to individual Contacts, which help user to recall the events related to that contact faster. Also, this helps to add imp. Notes to the contact which is used at the time of use.

[1.1] OBJECTIVES:

A Contact Management System is a software tool that helps Peoples to Organize and Manage their Contacts. It enables users to store, retrieve, and manage contact Information in an organized and efficient manner and also provide Emergency Contacts like Police, Ambulance, Fire Station, etc. It allows user to make a Quick Dial list in which user can Add the Important Contacts from the Contact list. Also user can make a separate notes to the Contact which make System efficient to user and it also search the contact through both First name & Last name

[1.2] PROBLEM STATEMENT:

You have been asked to Develop a Contact Management System in C language. The System should allow users to Add, View, Delete, Search, and Display Contacts. Also shows Emergency Contacts, add Notes to individual contacts, Quick Dial.

Chapter-2: DESIGN

[2.1] SYSTEM ARCHITECTURE:

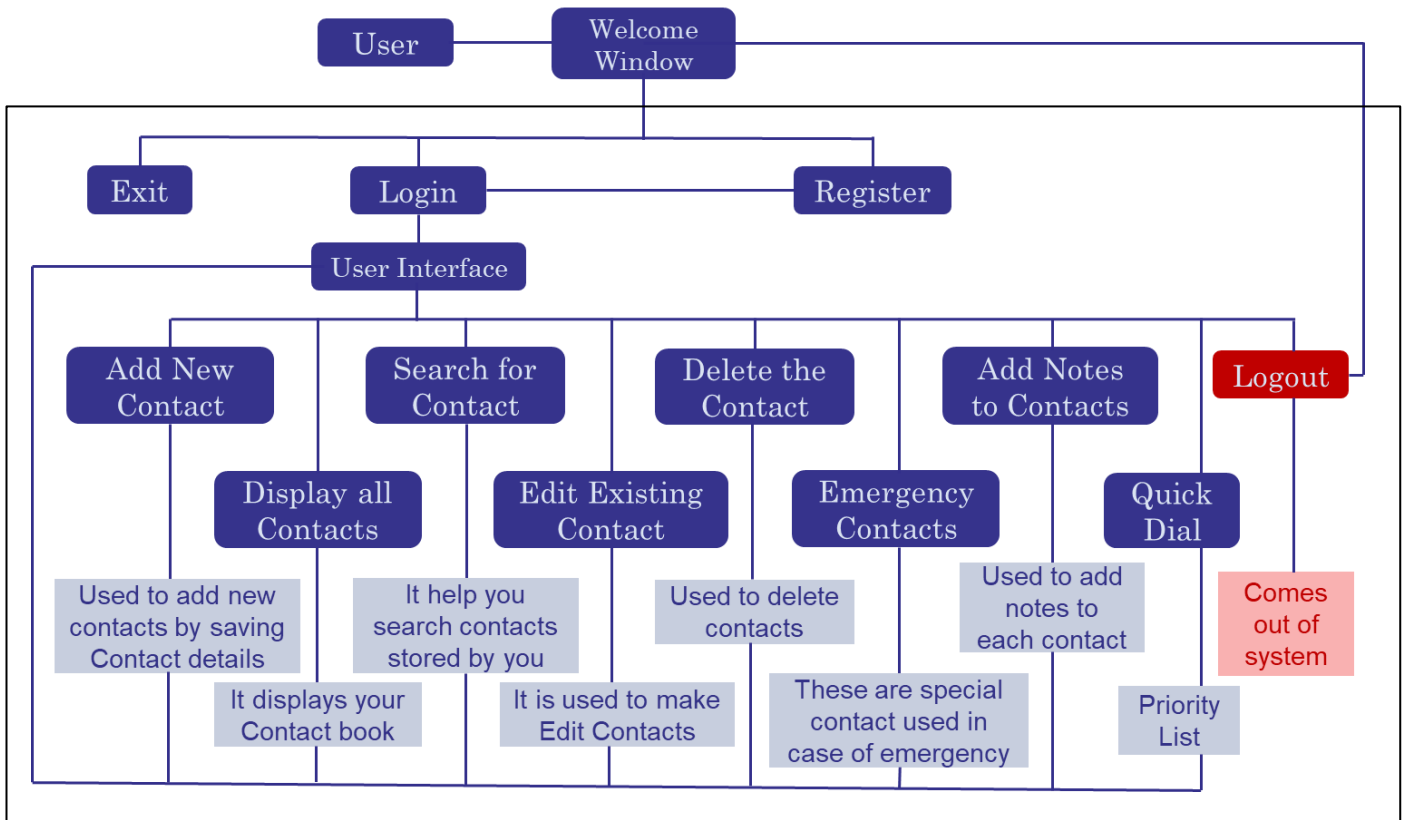


Fig. 2.1

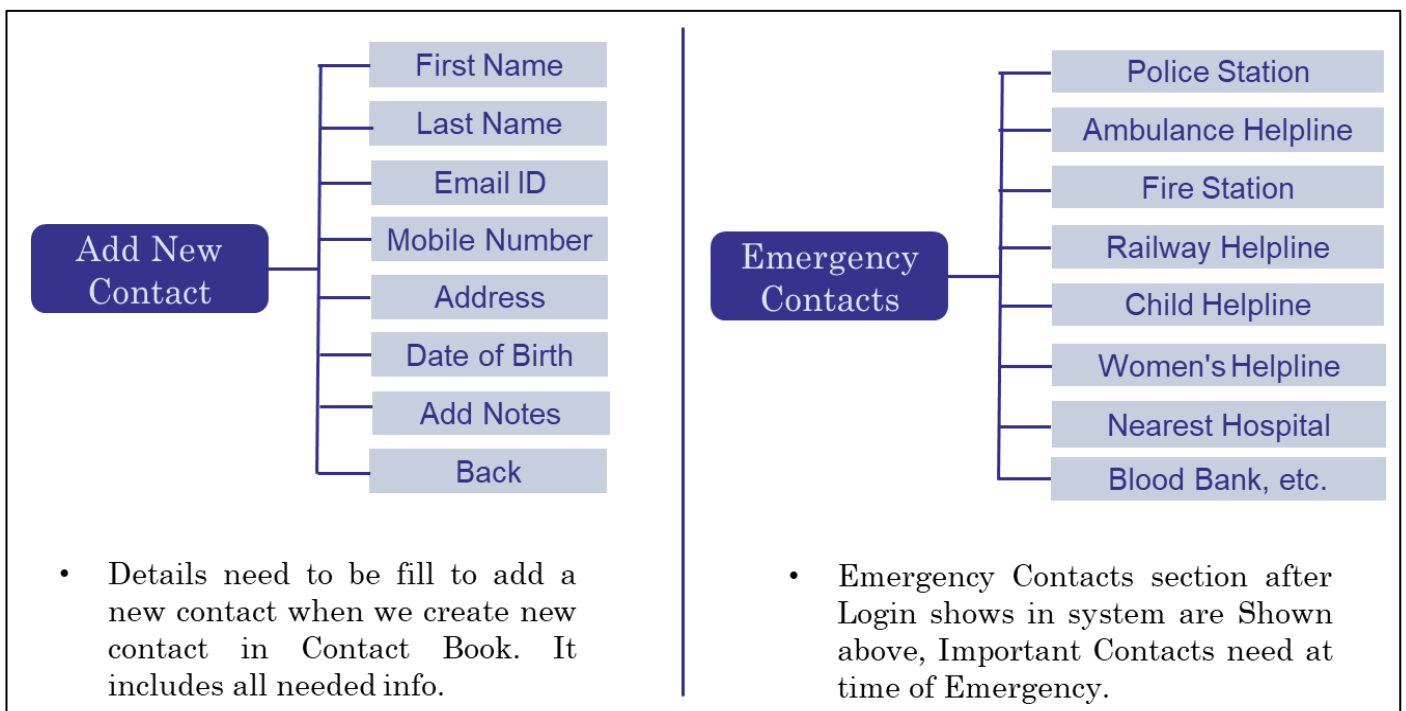


Fig. 2.2

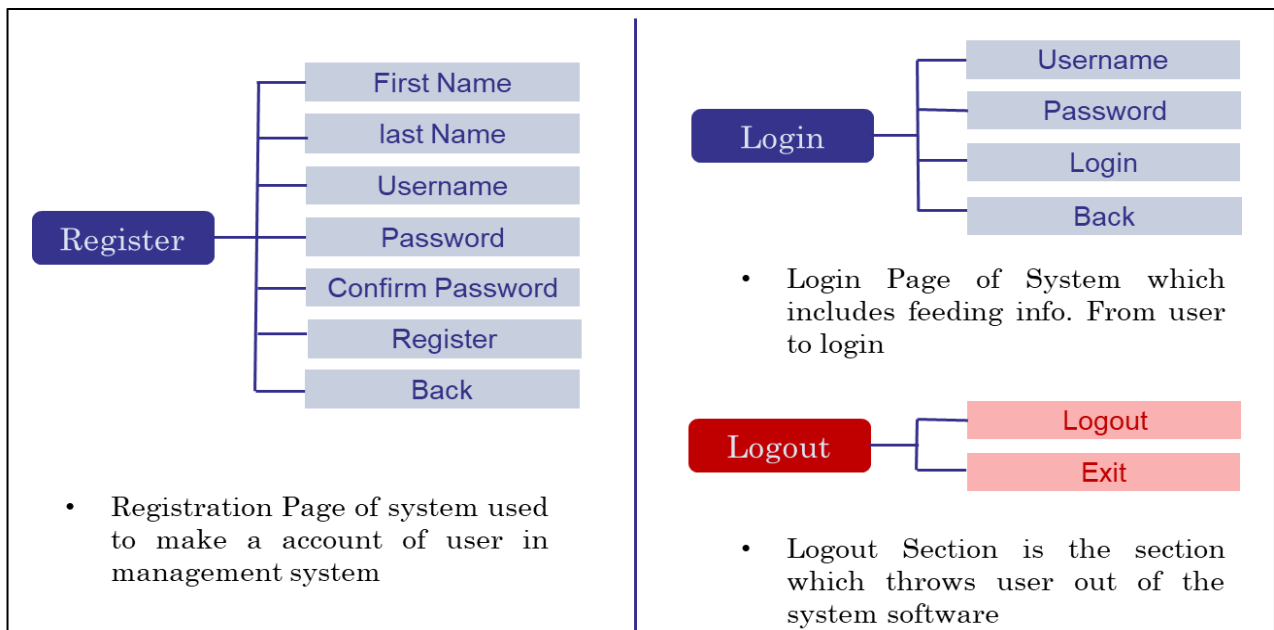


Fig. 2.3

[2.2] USE CASE DIAGRAM:

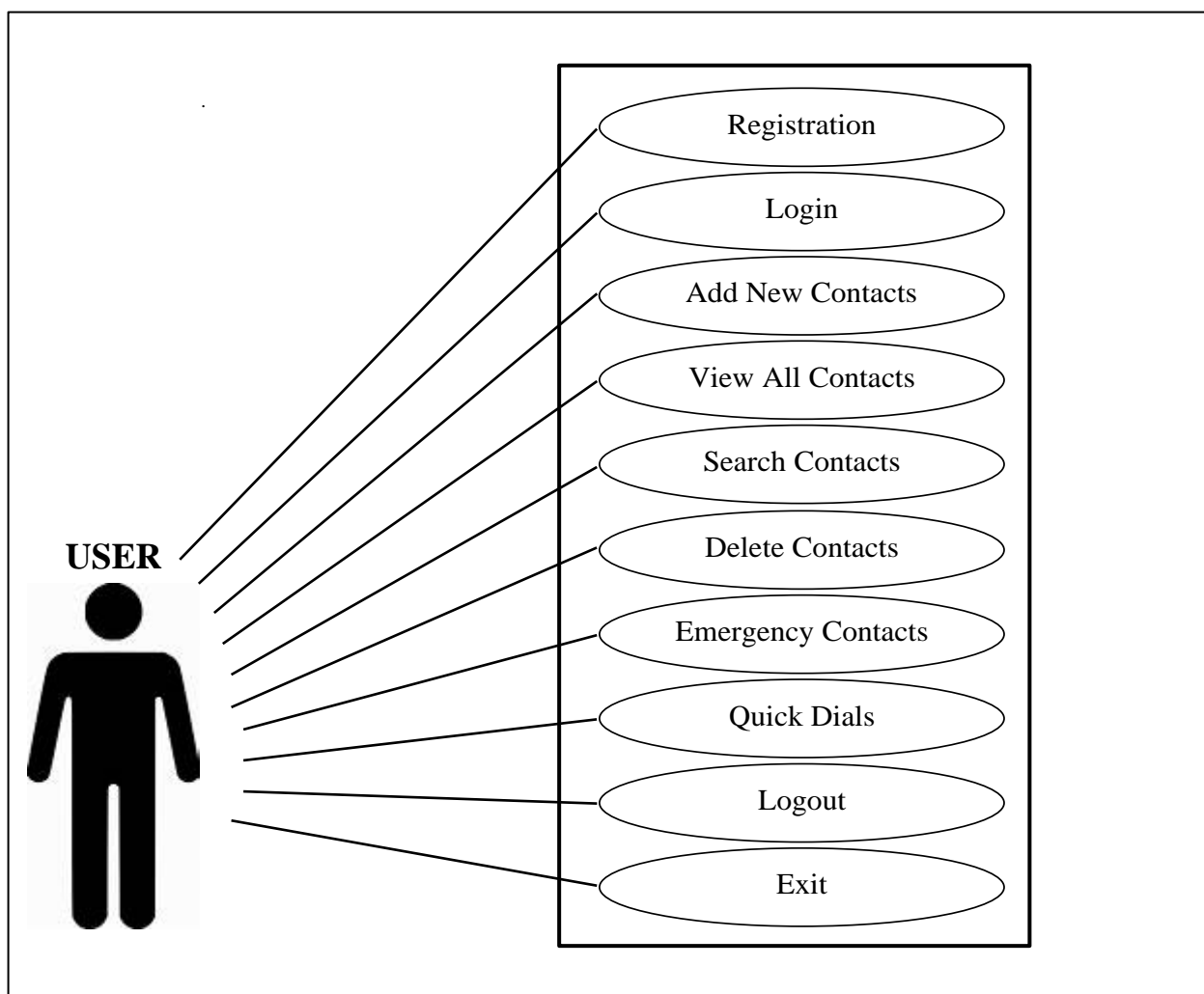


Fig. 2.4

Chapter-3

ALGORITHMS & FLOW CHARTS

[3.1] Algorithm:

An algorithm refers to a step-by-step procedure or set of rules for solving a specific problem or accomplishing a task. It is a logical and systematic approach to solving a problem, independent of any programming language.

- Step-1 — **STARTING** with **WELCOME** window
- Step-2 — after **WELCOME** Window Ask **USER** to Choose between **LOGIN & REGISTER**
- Step-3 —
 - If **REGISTER** is selected create new Account (**for New USER**)
 - If **LOGIN** is selected ask for login credentials (**for Existing USER**)
- Step-4 —
 - **Adding NEW CONTACT** will stored the details of Contact person through **FILE HANDLING**
 - Displaying **ALL CONTACT** which are stored through **FILE HANDLING** from **ADD NEW CONTACT**
 - Searching for **Individual CONTACT** which are stored from **ADD NEW CONTACT**
 - **DELETE CONTACT** will delete the contact chosen from Contacts saved by **ADD NEW CONTACT**
 - **EMERGENCY CONTACT** will display all the **Special imp. CONTACTS** (Police, Ambulance, etc.)
 - **QUICK DIAL** will add **Important or Closed** Contacts to Separate Contact List
 - **EDIT EXISTING CONTACT** will help **USER** to Edit the Saved Contacts
 - **LOGOUT** will take **USER** out to the **WELCOME Window** (again for **LOGIN & REGISTER** choice)
- Step-5 — Stop

[3.1] Flow Chart:

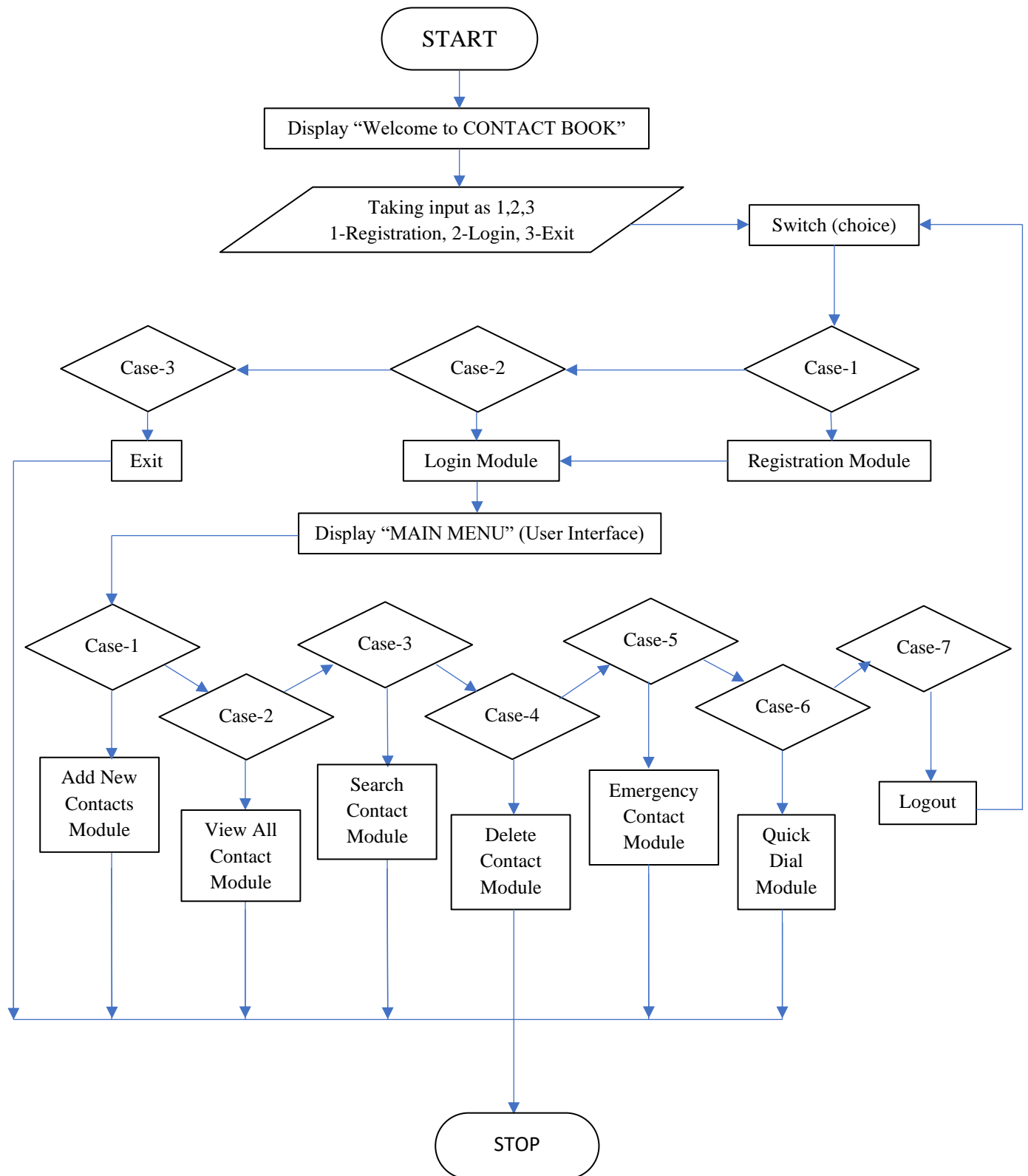


Fig. 3.1

Chapter-4

IMPLEMENTATION OF CODE

File Name: Contact.c

```
#include <stdio.h>
#include <stdlib.h>
#include <windows.h>
#include <unistd.h>
#include <string.h>
#include <stdbool.h>

#define MAX_LINE_LENGTH 100
#define MAX_NAME_LEN 50
#define MAX_CONTACTS 100
#define MAX_PASS_LENGTH 20
#define MAX_DOB_LEN 100
#define MAX 2000

bool isRunning = true;
void registration();
void login();
void add_email_to_file(char *email);
int is_valid_email(char *email);
void customer_interface();
void emergency();
void mail();
void date();
void delete();
void ex_it();
void add_contact();
void view_all();
void search();
void quick_dial();
void view_contact_list();

typedef struct
{
    char name[MAX_NAME_LEN];
    char pass[MAX_PASS_LENGTH];
}Member;
int readMembers(Member members[])
{
    FILE *fptr;
```



```

printf("\t|
printf("\t=====
);
Member members[MAX_CONTACTS];
int count, i;
char name[MAX_NAME_LEN], pass[MAX_PASS_LENGTH];
count = readMembers(members);
printf("\t-----> Enter your USERNAME : ");
scanf("%s", name);
printf("\t-----> Enter your PASSWORD : ");
scanf("%s", pass);
for (i = 0; i < count; i++)
{
    if(strcmp(members[i].name,name)==0&&strcmp(members[i].pass,pass)==0)
    {
        printf("\n\t-----> User Login SUCCESSFULLY! Welcome, %s.",members[i].name);
        printf("\n\t=====
n\n\n\n\n");
        customer_interface();
        return;
    }
}
printf("\n\tInvalid USERNAME or PASSWORD\n\n");
printf("\tLogin Again !?!");
printf("\n\t=====
\n\n\n\n\n");
login();
}

void mail()
{
    char email[50];
    printf("\t-----> Enter E-MAIL : ");
    scanf("%s",email);
    if (!is_valid_email(email)) {
        printf("\t(Invalid email format, Enter Again)\n");
        mail();
        return ;
    }
    FILE *file = fopen("contacts.txt", "a");
    fprintf(file,"%s\t", email);
    fclose(file);
}

int is_valid_email(char *email)
{
    char *at = strchr(email, '@');
    char *dot = strchr(email, '.');

```

```

return (at != NULL && dot != NULL && at < dot);
}
void customer_interface()
{
    int choice;
    printf("\t|||||||[] [] [] [] [] [] [] [] [] [] [] [] [] [] [] [] ||||| \n");
    printf("\t|||||||[] [] [] []          --- MAIN MENU ---          [] [] [] [] ||||| \n");
    printf("\t|||||||[] [] [] [] [] [] [] [] [] [] [] [] [] [] [] [] ||||| \n\n");
};
printf("\t[1]---> ADD NEW CONTACTS\n");
printf("\t[2]---> VIEW ALL CONTACTS\n");
printf("\t[3]---> SEARCH FOR CONTACTS\n");
printf("\t[4]---> DELETE CONTACT\n");
printf("\t[5]---> EMERGENCY CONTACTS\n");
printf("\t[6]---> QUICK DIAL\n");
printf("\t[7]---> LOGOUT\n\n");
printf("\t[_]---> Enter your choice : ");
scanf("%d", &choice);
printf("\n\t===== \n\n\n\n");
switch(choice)
{
    case 1 : add_contact();
            break;
    case 2 : view_all();
            customer_interface();
            break;
    case 3 : search();
            break;
    case 4 : delete();
            break;
    case 5 : emergency();
            break;
    case 6 : quick_dial();
            break;
    case 7 : main();
            break;
    default : printf("\t-----> Invalid CHOICE\n");
             printf("\n\t===== \n\n\n\n\n");
}
}

void add_contact()
{
    FILE *fp;int i;int choice;
    char first_name[20];
    char last_name[20];

```

```

char email[50];
char phone[15];
char notes[100];
char dob[10];
char add[50];

printf("\n\n\t===== \n");
printf("\t|                                ADD NEW CONTACT                                |\n");
printf("\t===== \n");
fp = fopen("contacts.txt", "a");
printf("\n\t-----> Enter FIRSTNAME          : ");
scanf("%s", first_name);
fprintf(fp, "%s ", first_name);
printf("\t-----> Enter LASTNAME          : ");
scanf("%s", last_name);
fprintf(fp, "%s ", last_name);
printf("\t-----> Enter Your PHONE NO.   : ");
scanf("%s", phone);
if(strlen(phone) != 10)
{
    for(i = strlen(phone); i < 10 || i > 10; )
    {
        printf("\t(Invalid Phone no. format, Enter Again)\n");
        printf("\t-----> Enter PHONE NO. again : ");
        scanf("%s", phone);
        if(strlen(phone) == 10)
        {
            fprintf(fp, "%s\t", phone);
            fclose(fp);
            break;
        }
    }
    else if(strlen(phone) == 10)
    {
        fprintf(fp, "%s\t", phone);
        fclose(fp);
    }
}
mail();
FILE *fptr = fopen("contacts.txt", "a");
printf("\t-----> DOB (DD/MM/YYYY*)      : ");
scanf("%s", dob);
fprintf(fptr, "%s ", dob);

printf("\t-----> Enter Address          : ");
scanf("%s", add);
fprintf(fptr, "%s ", add);
printf("\t-----> Add NOTES to Contact   : ");
scanf("%s", notes);
fprintf(fptr, "%s\n", notes);
fclose(fptr);
printf("\n\t-----> Contact Added SUCCESSFULLY");

```



```

printf("\n\t===== \n\n");

printf("\t[1]---> Add another NEW CONTACT\n");
printf("\t[2]---> RETURN to MAIN MENU\n\n");
printf("\t[_]---> Enter your choice : ");
scanf("%d", &choice);
printf("\n\t===== \n\n\n\n");
switch(choice)
{
    case 1 : add_contact();
             break;
    case 2 : customer_interface();
             break;
    default : printf("\t-----> Invalid CHOICE\n");
              printf("\n\t===== \n\n\n\n\n");
}
}

void view_all()
{
    FILE *fp;
    char first_name[20];
    char last_name[20];
    char email[50];
    char phone[15];
    char notes[100];
    char dob[10];
    char add[50];

    printf("\n\n\t===== \n\n");
    printf("\t|                                | \n");
    printf("\t|=====|");
    fp = fopen("contacts.txt", "r");
    printf("\n\t-----");
    printf("\n\t-----");
    while(fscanf(fp, "%s %s %s %s %s %s", first_name, last_name, phone, email, dob, notes, add) != EOF)
    {
        printf("\n\tFIRSTNAME      : %s", first_name);
        printf("\n\tLASTNAME       : %s", last_name);
        printf("\n\tMOBILE NO.     : %s", phone);
        printf("\n\tE-mail Address  : %s", email);
        printf("\n\tDOB (DD/MM/YYYY) : %s", dob);
        printf("\n\tNotes Added     : %s", notes);
        printf("\n\tAddress         : %s\n", add );
    }
}

```

```

    printf("\t-----\n");
    printf("\t-----");
}
fclose(fp);
printf("\n\t===== \n\n\n\n");
}
void search()
{
FILE *fp;int choice;
char search_string[100], line[200];

printf("\n\n\t===== \n");
printf("\t|                                | \n");
printf("\t\t===== \n\n");

printf("\t-----> Enter name to search : ");
scanf("%s", search_string);
printf("\t===== \n\n");
fp = fopen("contacts.txt", "r");
if(fp==NULL)
{
printf("\t-----> Unable to open file\n");
return;
}
printf("\t-----> All contacts with that name are listed below : \n\n");
printf("\t-----\n");
while(fgets(line,sizeof(line),fp)!=NULL)
{
if(strstr(line, search_string) != NULL)
{
printf("\t %s", line);
printf("\t-----\n");
}
}
printf("\n\t===== \n\n\n\n");
printf("\t[1]---> SEARCH another CONTACT\n");
printf("\t[2]---> RETURN to MAIN MENU\n\n");
printf("\t[_]---> Enter your choice : ");
scanf("%d", &choice);

```

```

printf("\n\t===== \n\n");
switch(choice)
{
    case 1 : search();
              break;
    case 2 : customer_interface();
              break;
    default : printf("\t-----> Invalid CHOICE\n");
              printf("\n\t===== \n\n\n\n\n");
}
fclose(fp);
return;
}
void delete()
{
    int lno, ctr = 0; int choice;
    char ch;
    FILE *fptr1, *fptr2;
    char str[MAX], temp[] = "temp.txt";
    printf("\n\n\t===== \n");
    printf("\t|                                     | \n");
    printf("\t=====");
    view_contact_list();
    printf("\n\t-----> DELETE CONTACT : \n");
    printf("\n\t===== \n");
};
fptr1 = fopen("contacts.txt", "r");
if (!fptr1)
{
    printf("\t-----> CONTACT not found or unable to open the CONTACT!!\n");
    return;
}
fptr2 = fopen(temp, "w");
if (!fptr2)
{
    printf("\t-----> Unable to open a temporary file to write!!\n");
    fclose(fptr1);
    return;
}
printf("\t-----> Input the Number of CONTACT you want to DELETE : ");
scanf("%d", &lno);
while (!feof(fptr1))
{
    strcpy(str, "\0");
    fgets(str, MAX, fptr1);

```

```

    if (!feof(fptr1))
    {
        ctr++;
        if (ctr != lno)
        {
            fprintf(fptr2, "%s", str);
        }
    }
}

fclose(fptr1);
fclose(fptr2);
remove("contacts.txt");
rename(temp, "contacts.txt");
view_contact_list();
printf("\n\t===== \n\n\n\n");

printf("\t[1]---> DELETE another CONTACT\n");
printf("\t[2]---> RETURN to MAIN MENU\n\n");
printf("\t[_]---> Enter your choice : ");
scanf("%d", &choice);
printf("\n\t===== \n\n\n");
switch(choice)
{
    case 1 : delete();
             break;
    case 2 : customer_interface();
             break;
    default : printf("\t-----> Invalid CHOICE\n");
              printf("\n\t===== \n\n\n\n\n");
}
}

void view_contact_list()
{
    FILE *fp;
    char line[100];
    int line_num=0;
    fp = fopen("contacts.txt", "r");
    printf("\n\n\t-----> Listed Contacts with numbers :\n");
    printf("\n\t===== \n");
    ;
    printf("\t----- \n");
    while(fgets(line, sizeof(line), fp) != NULL)
    {
        printf("\t%d. %s", ++line_num, line);
    }
}

```

```

printf("\t-----\n");
}
fclose(fp);
}

void quick_dial()
{
    FILE *sourceFile, *destinationFile;
    char line[MAX_LINE_LENGTH];int targetLineNumber;
    char firstString[MAX_LINE_LENGTH];
    char secondString[MAX_LINE_LENGTH];
    int choice;
    printf("\n\n\t=====
=====\\n");
    printf("\t|                                QUICK
DIALS                                |\\n");
    printf("\t-----
=====\\n\\n");
    int lineNumber = 0; view_contact_list();
    printf("\n\t-----> Add CONTACT to QUICK DIALS :");
    printf("\n\t=====
=====\\n");
    printf("\n\t-----> Enter contact no. you want to add to quick dials :");
    scanf("%d",&targetLineNumber);
    sourceFile = fopen("contacts.txt", "r");
    if (sourceFile == NULL)
    {
        printf("Failed to open source file.\\n");
        return;
    }
    destinationFile = fopen("quickdial.txt", "a");
    if (destinationFile == NULL)
    {
        printf("Failed to open destination file.\\n");
        fclose(sourceFile);
        return;
    }
    while (fgets(line, sizeof(line), sourceFile) != NULL)
    {
        lineNumber++;
        if(lineNumber == targetLineNumber)
        {
            sscanf(line, "%s %s", firstString, secondString);
            fprintf(destinationFile, "%s %s\\n", firstString, secondString);
            break;
        }
    }
    fclose(sourceFile);
    fclose(destinationFile);
}

```

```

    printf("-----> Contact added SUCCESSFULLY\n");

    printf("\t=====
=====\\n\\n\\n\\n\\n");

    printf("\t[1]---> Add another CONTACT to QUICK DIALS\n");
    printf("\t[2]---> RETURN to MAIN MENU\\n\\n");
    printf("\t[_]---> Enter your choice : ");
    scanf("%d", &choice);
    printf("\n\t=====
=====\\n\\n\\n");
    switch(choice)
    {
        case 1 : quick_dial();
                break;
        case 2 : customer_interface();
                break;
        default : printf("\t-----> Invalid CHOICE\\n");
                 printf("\n\t=====
=====\\n\\n\\n\\n\\n");
    }

    return;
}

void emergency()
{
    printf("\n\n\t=====
=====\\n");
    printf("\t|                                EMERGENCY
CONTACTS                                |\\n");
    printf("\t=====
=====");
    printf("\n\t-----> National Emergency No. : 112\\n");
    printf("\n\t-----> Police Station          : 100\\n");
    printf("\n\t-----> COVID-19 Helpline           : 011-23978046/1075 \\n");
    printf("\n\t-----> Ambulance Helpline          : 08/102/108\\n");
    printf("\n\t-----> Fire station                 : 101\\n");
    printf("\n\t-----> Railway Helpline             : 1512/139\\n");
    printf("\n\t-----> Child Helpline               : +91-712-2520699\\n");
    printf("\n\t-----> Womens Helpline              : 1091/181\\n");
    printf("\n\t-----> hospital helpline            : 104\\n");
    printf("\n\t-----> Blood Bank                   : 1910\\n");
    printf("\n\t-----> LPG Leak Helpline            : 1906\\n");
    printf("\n\t-----> Diasater Helpline            : 1077\\n");
    printf("\n\t-----> Muncipal Corporation         : 1800 1030 222\\n");
    printf("\n\t-----> Cyber Crime                  : 020-29710097\\n\\n");

```

```

printf("\t=====
=====\\n\\n\\n\\n\\n");
customer_interface();
}
void ex_it()
{
printf("\n\t=====
=====");
printf("\n\t|                               EXITING FROM THE
SYSTEM                               |");
printf("\n\t=====
=====\\n\\n");

char ThankYou[100] = "\t===== THANK YOU =====\\n";
char SeeYouSoon[100] = "\t===== SEE YOU SOON =====\\n";
int i=0;
for(i=0;i<strlen(ThankYou);i++){
printf("%c",ThankYou[i]);
Sleep(40);
}
for(i=0;i<strlen(SeeYouSoon);i++){
printf("%c",SeeYouSoon[i]);
Sleep(40);
}
}
}

```

Chapter-5: OUTPUT

[Fig. 5.1] Welcome Window:



After Welcome Window a choice between Registration & Login is been Asked, if we choose 1 for Registration the Output given below:

[Fig. 5.2] Registration:


```
|||||[] [] [] [] [] [] [] [] [] [] [] [] [] [] [] [] [] [] [] [] [] [] [] [] [] ||||| | | | | |
|||||[] [] [] [] [] [] [] [] [] [] [] [] [] [] [] [] [] [] [] [] [] [] [] [] |||||
|||||[] [] [] [] [] |||||[] [] [] [] [] |||||
|||||[] [] [] [] [] WELCOME TO CONTACT BOOK [] [] [] [] [] |||||
|||||[] [] [] [] [] |||||[] [] [] [] [] |||||
|||||[] [] [] [] [] [] [] [] [] [] [] [] [] [] [] [] [] [] [] [] [] [] [] [] |||||
|||||[] [] [] [] [] [] [] [] [] [] [] [] [] [] [] [] [] [] [] [] [] [] [] [] |||||

=====

[1]---> REGISTRATION
[2]---> LOGIN
[3]---> EXIT

[_]---> Enter your choice : 1

=====

|-----REGISTRATION-----|
|-----|

-----> Enter your FIRSTNAME : Mudit
-----> Enter your LASTNAME : Jain
-----> Enter your USERNAME : jainmudit696
-----> Enter your PASSWORD : Mudit@23

-----> User registered SUCCESSFULLY
=====
```

After Registration User is redirected to the Login window or User can also Login by choosing 2 in the Welcome Window, Login Output is given below:

[Fig. 5.3] Login:

```
=====
|-----LOGIN-----|
|-----|

-----> Enter your USERNAME : jainmudit696
-----> Enter your PASSWORD : Mudit@23

-----> User Login SUCCESSFULLY! Welcome, jainmudit696.
=====
```

After Login the System a User Interface is shown to the User Which include all the features which are going to used later

[Fig. 5.4] User Interface:

```
|||||||[] [] [] [] [] [] [] [] [] [] [] [] [] [] [] [] [] [] |||||
|||||||[] [] [] [] [] --- MAIN MENU --- [] [] [] [] |||||
|||||||[] [] [] [] [] [] [] [] [] [] [] [] [] [] [] [] [] |||||

[1]---> ADD NEW CONTACTS
[2]---> VIEW ALL CONTACTS
[3]---> SEARCH FOR CONTACTS
[4]---> DELETE CONTACT
[5]---> EMERGENCY CONTACTS
[6]---> QUICK DIAL
[7]---> LOGOUT

[_]---> Enter your choice :
```

After coming the Main Menu (User Interface) We are now ready to use all the features in the System So firstly we are going to ADD NEW CONTACTs by choosing 1 from the Main Menu.

[Fig. 5.5] Add New Contacts:

In Add New Contacts the Mob no. and Mail id sections are Authenticated. It means whenever we will add invalid format of email id and mob. No. the system will give that section again for fill without mistake.

An Output with both example is given below:

```

|||||[] [] [] [] [] [] [] [] [] [] [] [] [] [] [] [] [] [] [] |||||
|||||[] [] [] [] [] --- MAIN MENU --- [] [] [] [] |||||
|||||[] [] [] [] [] [] [] [] [] [] [] [] [] [] [] [] [] [] |||||

[1]----> ADD NEW CONTACTS
[2]----> VIEW ALL CONTACTS
[3]----> SEARCH FOR CONTACTS
[4]----> DELETE CONTACT
[5]----> EMERGENCY CONTACTS
[6]----> QUICK DIAL
[7]----> LOGOUT

[_]----> Enter your choice : 1

=====

=====
|                                     ADD NEW CONTACT                                     |
=====

-----> Enter FIRSTNAME      : Gourav
-----> Enter LASTNAME       : Kar
-----> Enter Your PHONE NO. : 984728132
(Invalid Phone no. format, Enter Again)
-----> Enter PHONE NO. again : 9876543210
-----> Enter E-MAIL        : abcdefghijkl
(Invalid email format, Enter Again)
-----> Enter E-MAIL        : abcdef@ghijk.com
-----> DOB (DD/MM/YYYY*)   : 06/06/2006
-----> Enter Address       : Arusai_homes,Pune
-----> Add NOTES to Contact : Developer

-----> Contact Added SUCCESSFULLY
=====

[1]----> Add another NEW CONTACT
[2]----> RETURN to MAIN MENU

[_]----> Enter your choice : |

```

After adding a Contact Successfully in system, a choice for Adding Another contact and Returning the User Interface is Given by System to the user if User Chooses 1 then system will be redirected to the Add New Contact Section Again for adding another Contact

If User will select 2 then the he/she will be redirected to the User Interface (Main Menu)

[Fig. 5.6] View All Contacts:

View all contact feature will show all the contacts, to using this feature we have to choose 2 in the choice and then all the contacts saved in system are shown below:

```
|||||[] [] [] [] [] [] [] [] [] [] [] [] [] [] [] [] [] [] [] [] [] [] ||||| | | | | | |
|||||[] [] [] [] |||      MAIN MENU      |||[] [] [] [] [] [] [] [] [] |||||
|||||[] [] [] [] [] [] [] [] [] [] [] [] [] [] [] [] [] [] [] [] [] |||||

[1]--> ADD NEW CONTACTS
[2]--> VIEW ALL CONTACTS
[3]--> SEARCH FOR CONTACTS
[4]--> DELETE CONTACT
[5]--> EMERGENCY CONTACTS
[6]--> QUICK DIAL
[7]--> LOGOUT

[.]--> Enter your choice : 2

=====

=====
|                                     LIST ALL CONTACT                                     |
=====

FIRSTNAME      : Gourav
LASTNAME       : Kar
MOBILE NO.     : 9876543210
E-mail Address : abcdef@ghijk.com
DOB (DD/MM/YYYY) : 06/06/2006Arusai_homes,Pune
Notes Added    : Arusai_homes,Pune
Address        : Developer
=====

FIRSTNAME      : Mudit
LASTNAME       : Jain
MOBILE NO.     : 1234567890
E-mail Address : jainmudit@gmail.com
DOB (DD/MM/YYYY) : 09/09/2008ivy
Notes Added    : ivy
Address        : doctor
=====

FIRSTNAME      : Mudit
LASTNAME       : Shah
MOBILE NO.     : 1234567980
E-mail Address : asdfghj123@gmail.com
DOB (DD/MM/YYYY) : 06/04/2001pune
Notes Added    : pune
Address        : plumber
=====

FIRSTNAME      : Khushi
LASTNAME       : Kumar
MOBILE NO.     : 9876543210
E-mail Address : khushi_k12@gmail.com
DOB (DD/MM/YYYY) : 03/03/2003mumbai
Notes Added    : mumbai
Address        : software
=====
```

[Fig.5.7] Search for Contacts:

After viewing all Contacts, now we will Search the Contact through Search Contact. For which we have select 3 in the user interface

In Search Contact We Can search through both First Name and Last Name

```
|||||||[] [] [] [] [] [] [] [] [] [] [] [] [] [] [] [] [] [] [] [] [] |||||
|||||||[] [] [] [] [] --- MAIN MENU --- [] [] [] [] |||||
|||||||[] [] [] [] [] [] [] [] [] [] [] [] [] [] [] [] [] [] [] [] [] |||||

[1]---> ADD NEW CONTACTS
[2]---> VIEW ALL CONTACTS
[3]---> SEARCH FOR CONTACTS
[4]---> DELETE CONTACT
[5]---> EMERGENCY CONTACTS
[6]---> QUICK DIAL
[7]---> LOGOUT

[_]---> Enter your choice : 3

=====

| SEARCH CONTACTS |
=====

-----> Enter name to search : Mudit
=====

-----> All contacts with that name are listed below :

-----
Mudit Jain 1234567890 jainmudit@gmail.com 09/09/2008 ivy doctor
-----
Mudit Shah 1234567980 asdfghj123@gmail.com 06/04/2001 pune plumber
-----

=====

[1]---> SEARCH another CONTACT
[2]---> RETURN to MAIN MENU

[_]---> Enter your choice : |
```

Here as there are two Contacts with First Name is Saved in System so at time of Searching the Contact Both the Contact with the name Mudit are displayed.

After Search Contact process overs then again, a choice asked for Search Another Contact or Return to User Interface (Main Menu).

[Fig. 5.8] Delete Contacts:

Choose 4 in the user interface to use Delete Contact feature. Firstly, all the contacts are shown listed the numbers before every contact, user asked for a number which will be deleted from the shown list above and the show a updated list.

```

|||||[] [] [] [] [] [] [] [] [] [] [] [] [] [] [] [] [] ||||| | |
|||||[] [] [] [] |----- MAIN MENU -----| [] [] [] [] |||||
|||||[] [] [] [] [] [] [] [] [] [] [] [] [] [] [] [] [] |||||

[1]---> ADD NEW CONTACTS
[2]---> VIEW ALL CONTACTS
[3]---> SEARCH FOR CONTACTS
[4]---> DELETE CONTACT
[5]---> EMERGENCY CONTACTS
[6]---> QUICK DIAL
[7]---> LOGOUT

[.]---> Enter your choice : 4

=====

|-----> DELETED CONTACTS |

=====

-----> Listed Contacts with numbers :

=====
1. Mudit Jain 1234567890      jainmudit@gmail.com      09/09/2008 ivy doctor
2. Aditya Rawat 9876543210   aditya!e@gmail.com       08/07/2002 pauri ai
3. Vns Lakhwani 1234567890   asdfghjkl@gmail.com      03/03/2003 ivy dypatil
4. Abhishek Jha 1234509876   abhi_jha@gmail.co        01/01/2001 ivy friend
5. Tanu Pawar 1234567890     tanupawar@gmail.com      23/03/2003 ivy csd
=====

-----> DELETE CONTACT :

=====
-----> Input the Number of CONTACT you want to DELETE : 4

-----> Listed Contacts with numbers :

=====
1. Mudit Jain 1234567890      jainmudit@gmail.com      09/09/2008 ivy doctor
2. Aditya Rawat 9876543210   aditya!e@gmail.com       08/07/2002 pauri ai
3. Vns Lakhwani 1234567890   asdfghjkl@gmail.com      03/03/2003 ivy dypatil
4. Tanu Pawar 1234567890     tanupawar@gmail.com      23/03/2003 ivy csd
=====

[1]---> DELETE another CONTACT
[2]---> RETURN to MAIN MENU

[.]---> Enter your choice :

```

Again, after deleting a contact user asked for a choice between Delete Another Contact or Return to User interface (Main Menu)

[Fig. 5.9] Emergency Contacts:

For Emergency Contacts we have to select 5 in the User Interface and then all the Emergency Contacts Shown to the user in the System.

```
|||||[] [] [] [] [] [] [] [] [] [] [] [] [] [] [] [] [] [] [] [] [] |||||
|||||[] [] [] [] []      --- MAIN MENU ---      [] [] [] [] [] |||||
|||||[] [] [] [] [] [] [] [] [] [] [] [] [] [] [] [] [] [] [] [] |||||

[1]----> ADD NEW CONTACTS
[2]----> VIEW ALL CONTACTS
[3]----> SEARCH FOR CONTACTS
[4]----> DELETE CONTACT
[5]----> EMERGENCY CONTACTS
[6]----> QUICK DIAL
[7]----> LOGOUT

[_]----> Enter your choice : 5

=====

=====
|                                     EMERGENCY CONTACTS                                     |
=====
-----> National Emergency No. : 112

-----> Police Station           : 100

-----> COVID-19 Helpline       : 011-23978046/1075

-----> Ambulance Helpline     : 08/102/108

-----> Fire station           : 101

-----> Railway Helpline       : 1512/139

-----> Child Helpline         : +91-712-2520699

-----> Womens Helpline        : 1091/181

-----> hospital helpline     : 104

-----> Blood Bank             : 1910

-----> LPG Leak Helpline      : 1906

-----> Diasater Helpline     : 1077

-----> Muncipal Corporation  : 1800 1030 222

-----> Cyber Crime           : 020-29710097

=====
```

[Fig. 5.10] Quick Dial:

For Quick Dial Feature we have to select 6 in the User Interface and then the list of contact with numbers shown to the user, user just have to select the no. of contact which is going to be stored in Quick Dial list and the used an Important Contact.

```
|||||[] [] [] [] [] [] [] [] [] [] [] [] [] [] [] [] [] [] [] [] [] [] [] ||||| | | | | | | | | | |
|||||[] [] [] [] |||||      --- MAIN MENU ---      |||||[] [] [] [] |||||
|||||[] [] [] [] [] [] [] [] [] [] [] [] [] [] [] [] [] [] [] [] [] |||||

[1]--> ADD NEW CONTACTS
[2]--> VIEW ALL CONTACTS
[3]--> SEARCH FOR CONTACTS
[4]--> DELETE CONTACT
[5]--> EMERGENCY CONTACTS
[6]--> QUICK DIAL
[7]--> LOGOUT

[.]--> Enter your choice : 6

=====

|                                     |
|                               QUICK DIALS                               |
|                                     |
=====

-----> Listed Contacts with numbers :

=====

1. Mudit Jain 1234567890      jainmudit@gmail.com      09/09/2008 ivy doctor
2. Aditya Rawat 9876543210    adityale@gmail.com      08/07/2002 pauri ai
3. Vns Lakhwani 1234567890    asdfghjkl@gmail.com     03/03/2003 ivy dypatil
4. Tanu Pawar 1234567890      tanupawar@gmail.com     23/03/2003 ivy csd

-----> Add CONTACT to QUICK DIALS :

=====

-----> Enter contact no. you want to add to quick dials :4
-----> Contact added SUCCESSFULLY

=====

[1]--> Add another CONTACT to QUICK DIALS
[2]--> RETURN to MAIN MENU

[.]--> Enter your choice : |
```

Here again after feature end a choice between Add another Contact to Quick Dial and Return to Main Menu (User Interface) is Asked to the user

[Fig. 5.11] Logout:

For Logout feature we have to select 7 in the user interface and then the user will get out of the user interface and redirected to the Welcome Window for Login or Register Again or also to get Exit of the system.

```
|||||[] [] [] [] [] [] [] [] [] [] [] [] [] [] [] [] [] [] |||||
|||||[] [] [] [] [] --- MAIN MENU --- [] [] [] [] |||||
|||||[] [] [] [] [] [] [] [] [] [] [] [] [] [] [] [] [] |||||

[1]---> ADD NEW CONTACTS
[2]---> VIEW ALL CONTACTS
[3]---> SEARCH FOR CONTACTS
[4]---> DELETE CONTACT
[5]---> EMERGENCY CONTACTS
[6]---> QUICK DIAL
[7]---> LOGOUT

[_]---> Enter your choice : 7

=====

|||||[] [] [] [] [] [] [] [] [] [] [] [] [] [] [] [] [] ||||| |
|||||[] [] [] [] [] [] [] [] [] [] [] [] [] [] [] [] [] |||||
|||||[] [] [] [] | [] [] [] [] |||||
|||||[] [] [] [] | WELCOME TO CONTACT BOOK [] [] [] [] |||||
|||||[] [] [] [] | [] [] [] [] |||||
|||||[] [] [] [] [] [] [] [] [] [] [] [] [] [] [] [] [] |||||
|||||[] [] [] [] [] [] [] [] [] [] [] [] [] [] [] [] [] |||||

=====

[1]---> REGISTRATION
[2]---> LOGIN
[3]---> EXIT

[_]---> Enter your choice : |
```

[Fig. 5.12] Exit:

For Exit feature we have to select 3 in the Welcome Window to get Exiting From the System

```
|||||[] [] [] [] [] [] [] [] [] [] [] [] [] [] [] [] [] [] [] [] [] [] ||||| | |
|||||[] [] [] [] [] [] [] [] [] [] [] [] [] [] [] [] [] [] [] [] [] |||||
|||||[] [] [] [] |                                     |[] [] [] [] |||||
|||||[] [] [] []   WELCOME TO CONTACT BOOK   [] [] [] [] |||||
|||||[] [] [] []   |[] [] [] [] |||||
|||||[] [] [] [] [] [] [] [] [] [] [] [] [] [] [] [] [] [] [] [] [] |||||
|||||[] [] [] [] [] [] [] [] [] [] [] [] [] [] [] [] [] [] [] [] [] |||||

=====

[1]---> REGISTRATION
[2]---> LOGIN
[3]---> EXIT

[_]---> Enter your choice : 3

=====

|                                     |
|                               EXITING FROM THE SYSTEM                               |
|                                     |
=====

===== THANK YOU =====
===== SEE YOU SOON =====
```

Chapter-6: CONCLUSION

In conclusion, the contact management system developed in the C language provides a user-friendly interface for storing and managing contacts. It allows users to add new contacts with relevant information such as name, phone number, and email address. The system also supports searching for contacts based on various criteria, such as name or phone number, enabling users to quickly retrieve specific contacts. Furthermore, the system allows for updating contact details and deleting unwanted contacts, ensuring the contact list remains up to date.

Throughout the development process, we utilized fundamental programming concepts in C, such as file handling, data structures, and functions, to create a robust and efficient contact management system. The system demonstrates the power and flexibility of the C language in building practical applications.

Overall, the contact management system in C language serves as an excellent foundation for organizing and maintaining contacts efficiently. It can be customized and extended to meet specific requirements, providing a reliable solution for personal or professional contact management needs.

REFERENCES

- [1.] <https://www.javatpoint.com/c-programming-language-tutorial>
- [2.] <https://www.w3schools.com>
- [3.] [C Tutorial \(tutorialspoint.com\)](#)