



REPUBLIC OF YEMEN
SANA'A UNIVERSITY
FACULTY OF ENGINEERING
MECHATRONICS ENGINEERING DEPARTMENT



Contact Management System.

Done by:

Osama Abdulelah Alward

202174040

Group 1.

System (Parallel)

Supervised by:

Aseel Taher

Abstract:

Contact management is the process of storing and tracking data on customers and leads. In addition to providing quick access to customer data, investing in contact management can help you grow and diversify your customer base.

Contents

Abstract:	2
Introduction:.....	5
Tools of Structured analysis:	5
Data Flow Diagrams (DFDs):	5
Data Dictionary:.....	5
Header Files:.....	6
• stdio.h: Defines types and macros needed for the Standard I/O Package defined in Kernighan and Ritchie and extended under UNIX System V. Defines the standard I/O predefined streams stdin, stdout, stdprn, and stderr, and declares stream-level I/O routines	6
• conio.h:	6
• string.h:	6
• process.h:	6
• stdlib.h:	6
• dos.h:	6
System Defined Functions:	6
• getch:	6
• Gets:	6
• Fwrite:	6
• Fclose:	7
• fflush:	7
• long:	7
• char:	7
• *fp:	7
• Switch:	7
• break:	7
• System:	7
• While:	7
• remove:	7
• goto:	7
• default:	7
• stricmp:	7
• Struct:	7
Contact management code:	8
Flowchart:	15

Discussion and output:	19
Conclusion:	22
Future work:	22
References:	22

Figure 1 choice number	19
Figure 2 add new contact	19
Figure 3 list of contacts	20
Figure 4 search about contact.....	20
Figure 5 Edit contact.....	21
Figure 6 delete a contact.....	21

Introduction:

Contact management is the process of recording contacts' details and tracking their interactions with a business. which allow businesses to improve sales and service levels leveraging a wider range of data also A contact management system enables businesses to store customers' names and contact details in a desktop database.

Tools of Structured analysis:

Data Flow Diagrams (DFDs): Data flow diagrams are widely used graphic tools for describing the movement of data within or outside the system. As a DFD consists of a series of bubbles joining by lines, it is also known as 'bubble chart'.

Data Dictionary: Data dictionary is an organized list of terms and their definitions for all the data elements and data structures that are pertinent to the system. It stores names along with their descriptions of all data used in a system.

Header Files:

- **stdio.h**: Defines types and macros needed for the Standard I/O Package defined in Kernighan and Ritchie and extended under UNIX System V. Defines the standard I/O predefined streams stdin, stdout, stderr, and declares stream-level I/O routines
- **conio.h**: Declares various functions used in calling the DOS console I/O routines.
- **string.h**: Declares several strings- and memory-manipulation routines.
- **process.h**: To perform mathematical operations in the code.
- **stdlib.h**: Declares several commonly used routines: conversion routines, search/sort routines, and other miscellany.
- **dos.h**: Defines various constants and gives declarations needed for DOS and 8086-specific calls.

System Defined Functions:

- **getch**: reads a single character directly from the keyboard, without echoing to the screen.
- **Gets**: gets collects a string of characters terminated by a new line from the standard input stream stdin and puts it into s.
- **Fwrite**: writes a record from an array into a files

- **Fclose**: close a file stream
- **fflush**: flushes an output stream
- **long**: A long int typically uses twice as many bits as a regular int, allowing it to hold much larger numbers
- **char**: char uses type to store characters and letters.
- ***fp**: a pointer of FILE type and FILE is a structure that store following information about opened file.
- **Switch**: is an alternate to if-else-if ladder statement which allows us to execute multiple operations for the different possible values of a single variable called switch variable.
- **break**: is a loop control statement which is used to terminate the loop.
- **System**: It is used to pass the commands that can be executed in the command processor or the terminal of the operating system, and finally returns the command after it has been completed.
- **While**: A while loop in C programming repeatedly executes a target statement as long as a given condition is true.
- **remove**: delete the file identified the string pointer
- **goto**: is used to transfer the program control to a predefined label
- **default**: The default statement is executed if no case constant-expression value is equal to the value of expression
- **stricmp**: The stricmp function compares string1 and string2 without sensitivity to case.
- **Struct**: a struct (or structure) is a collection of variables (can be of different types) under a single name.

Contact management code:

```
#include<stdio.h>

#include<conio.h>

#include<string.h>

#include<process.h>

#include<stdlib.h>

#include<dos.h>

struct contact

{

    long ph;

    char name[20],add[20],email[30];

} list;

char query[20],name[20];

FILE *fp, *ft;

int i,n,ch,l,found;

int main()

{

main:

    system("cls");    /* *****Main menu
***** */

    printf("\n\t **** Welcome to Contact Management System
*****");

    printf("\n\n\n\t\tMAIN
MENU\n\t\t=====
\n\t\t[1] Add a new
Contact\n\t\t[2] List all Contacts\n\t\t[3] Search for
contact\n\t\t[4] Edit a Contact\n\t\t[5] Delete a
Contact\n\t\t[0] Exit\n\t\t=====
\n\t\t");

    printf("Enter the choice:");

    scanf("%d",&ch);

    switch(ch)

    {

case 0:
```



```

printf("\n\n\t\tAre you sure you want to exit?");

break;

/* *****Add new contacts***** */

case 1:

    system("cls");

    fp=fopen("contact.dll","a");

    for (;;)

    {
        fflush(stdin);

        printf("To exit enter blank space in the name
input\nName (Use identical):");

        scanf("%[^\\n]", &list.name);

        if(strcmp(list.name, "")==0 || strcmp(list.name, "
")==0)

            break;

        fflush(stdin);

        printf("Phone:");

        scanf("%ld", &list.ph);

        fflush(stdin);

        printf("address:");

        scanf("%[^\\n]", &list.add);

        fflush(stdin);

        printf("email address:");

        gets(list.email);

        printf("\n");

        fwrite(&list, sizeof(list), 1, fp);

    }

    fclose(fp);

    break;

/* *****list of
contacts***** */

```

```

case 2:

    system("cls");

printf("\n\t\t===== \n\t\t\tLIST OF
CONTACTS\n\t\t\t===== \n\nName\t\tPhone
No\t\tAddress\t\tE-mail
ad.\n=====
===== \n\n");

    for(i=97; i<=122; i=i+1)

    {

        fp=fopen("contact.dll","r");

        fflush(stdin);

        found=0;

        while(fread(&list,sizeof(list),1,fp)==1)

        {

            if(list.name[0]==i || list.name[0]==i-32)

            {

                printf("\nName\t: %s\nPhone\t:
%d\nAddress\t: %s\nEmail\t: %s\n",list.name,
                                list.ph,list.add,list.email);

                found++;

            }

        }

        if(found!=0)

        {

printf("=====
== [%c]-(%d) \n\n",i-32,found);

            getch();

        }

        fclose(fp);

    }

    break;

/* *****search
contacts***** */

```

```

case 3:

    system("cls");

    do

    {

        found=0;

        printf("\n\n\t...:CONTACT
SEARCH\n\t===== \n\t...:Name of contact to
search: ");

        fflush(stdin);

        scanf("%[^\\n]", &query);

        l=strlen(query);

        fp=fopen("contact.dll", "r");

        system("cls");

        printf("\n\n\t...:Search result for '%s'
\n===== \n", query);

        while(fread(&list, sizeof(list), 1, fp)==1)

        {

            for(i=0; i<=l; i++)

                name[i]=list.name[i];

            name[l]='\0';

            if(stricmp(name, query)==0)

            {

                printf("\n...:Name\t: %s\n...:Phone\t:
%ld\n...:Address\t: %s\n...:Email\t:
%s\n", list.name, list.ph, list.add, list.email);

                found++;

                if (found%4==0)

                {

                    printf("...:Press any key to
continue...");

                    getch();

                }

```

```

        }

    }

    if(found==0)

        printf("\n...:No match found!");

    else

        printf("\n...:%d match(s) found!",found);

        fclose(fp);

        printf("\n ...:Try again?\n\n\t[1] Yes\t\t[0]
No\n\t");

        scanf("%d",&ch);

    }
    while(ch==1);

    break;

    /* *****edit
contacts***** */

    case 4:

        system("cls");

        fp=fopen("contact.dll","r");

        ft=fopen("temp.dat","w");

        fflush(stdin);

        printf("...:Edit
contact\n=====
of contact to edit:");

        scanf("%[^\\n]",name);

        while(fread(&list,sizeof(list),1,fp)==1)

        {

            if(strcmp(name,list.name)!=0)

                fwrite(&list,sizeof(list),1,ft);

        }

        fflush(stdin);

        printf("\n\n...:Editing '%s'\n\n",name);

```

```

printf("...:Name(Use identical):");

scanf("%[^\\n]", &list.name);

fflush(stdin);

printf("...:Phone:");

scanf("%ld", &list.ph);

fflush(stdin);

printf("...:address:");

scanf("%[^\\n]", &list.add);

fflush(stdin);

printf("...:email address:");

gets(list.email);

printf("\\n");

fwrite(&list, sizeof(list), 1, ft);

fclose(fp);

fclose(ft);

remove("contact.dll");

rename("temp.dat", "contact.dll");

break;

/* *****delete
contacts***** */

case 5:

    system("cls");

    fflush(stdin);

    printf("\\n\\n\\t...:DELETE A
CONTACT\\n\\t=====\\n\\t...:Enter the name of
contact to delete:");

    scanf("%[^\\n]", &name);

    fp=fopen("contact.dll", "r");

    ft=fopen("temp.dat", "w");

    while(fread(&list, sizeof(list), 1, fp) !=0)

        if (strcmp(name, list.name) !=0)

```

```

        fwrite(&list, sizeof(list), 1, ft);

fclose(fp);

fclose(ft);

remove("contact.dll");

rename("temp.dat", "contact.dll");

break;

default:

    printf("Invalid choice");

    break;

}

printf("\n\n\n...:Enter the Choice:\n\n\t[1] Main Menu\t\t[0]
Exit\n");

scanf("%d", &ch);

switch (ch)

{

case 1:

    goto main;

case 0:

    break;

default:

    printf("Invalid choice");

    break;

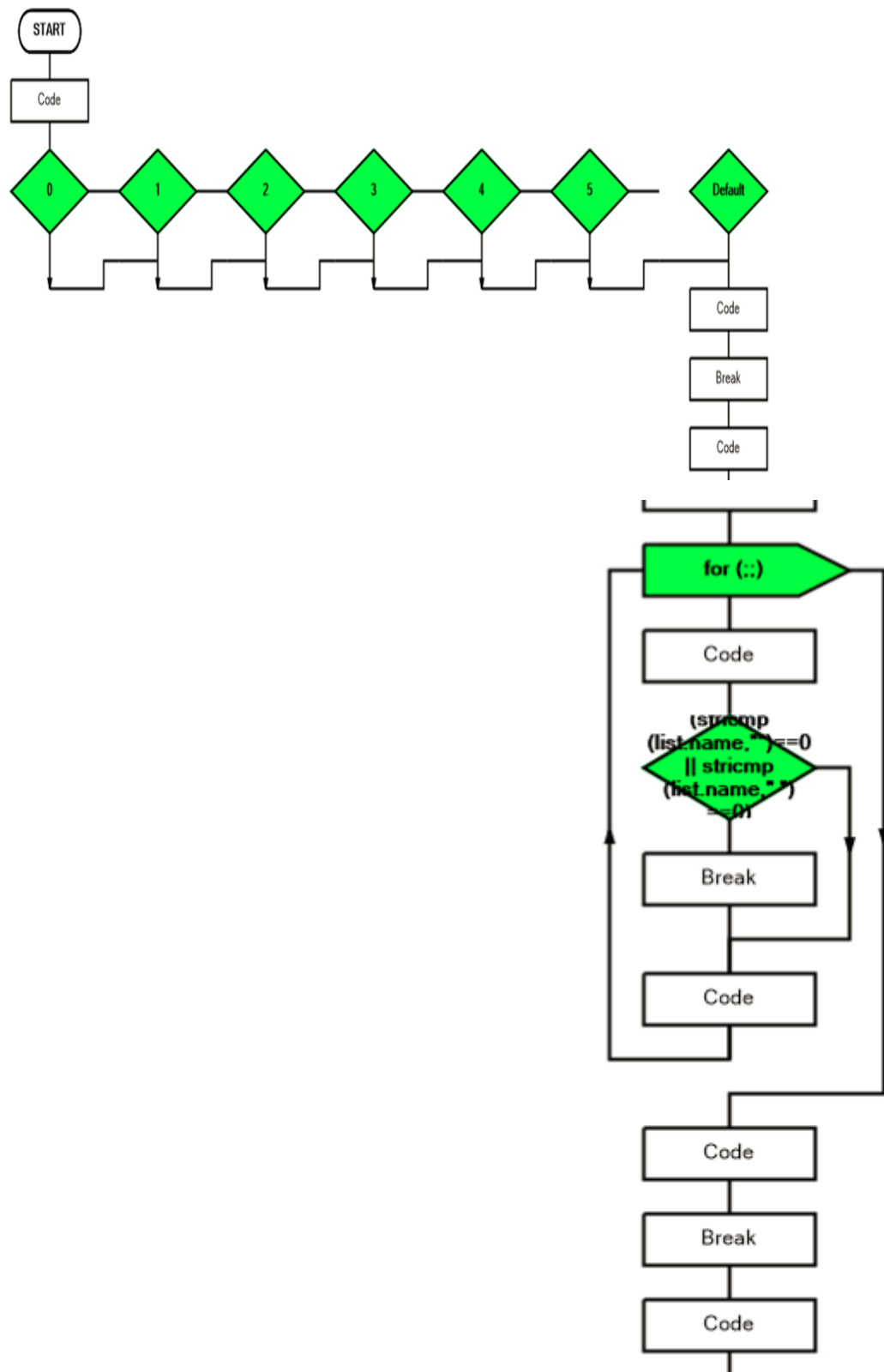
}

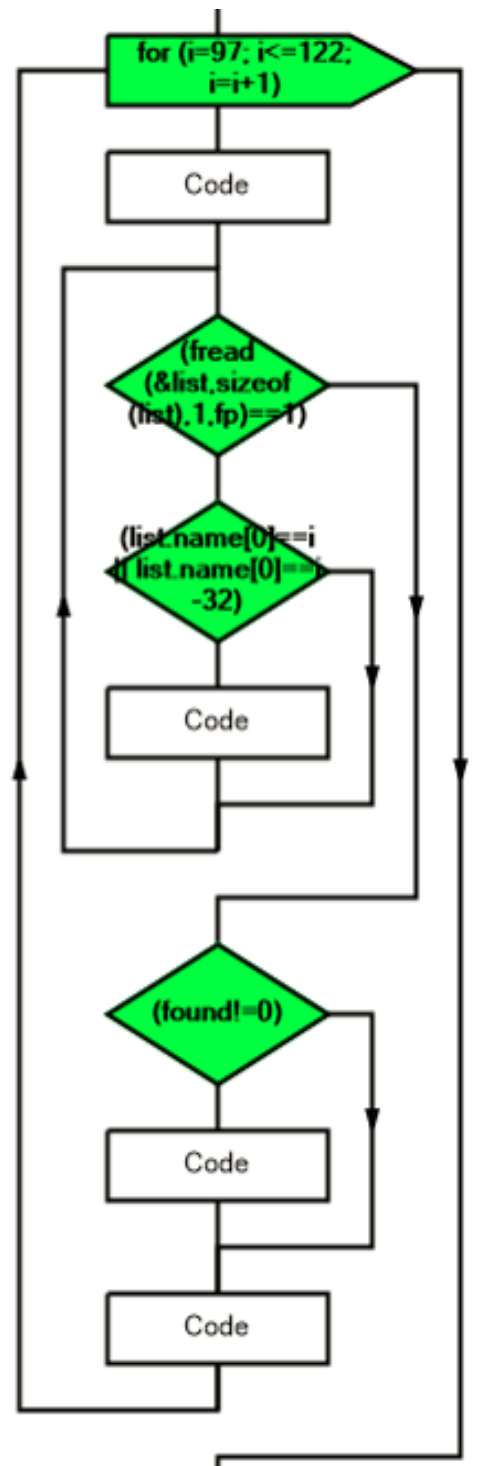
return 0;

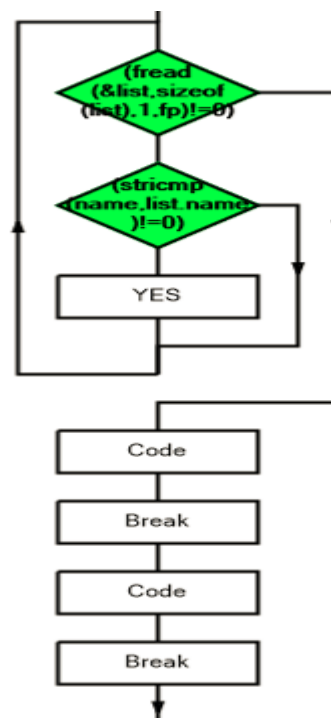
}

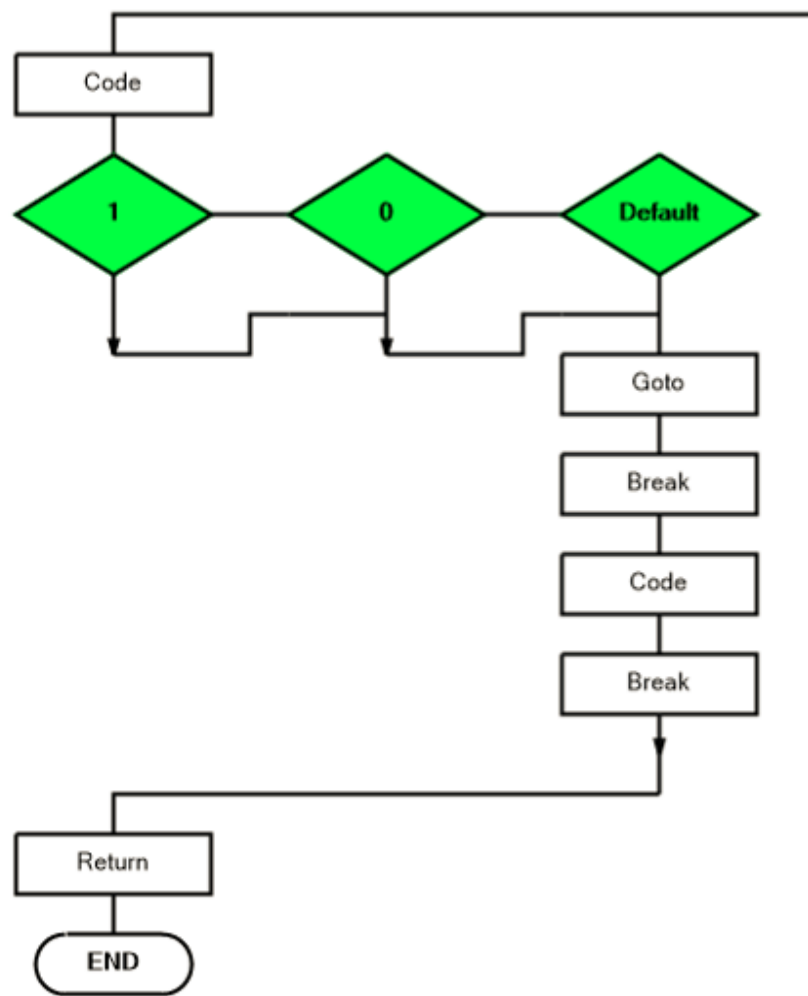
```

Flowchart:









Discussion and output:

the main menu <<choice number

Choose an option from the list viewed. Each option will view a specific menu except option 0 will close the program

```
**** Welcome to Contact Management System ****

      MAIN MENU
=====
[1] Add a new Contact
[2] List all Contacts
[3] Search for contact
[4] Edit a Contact
[5] Delete a Contact
[0] Exit
=====
Enter the choice:
```

Figure 1 choice number

Option# [1] <<Add a new contact

Options enter your (name, phone, address, email)

Fill up the information

```
To exit enter blank space in the name input
Name (Use identical):osama alward
Phone:774412944
address:sanaa
email address:alwardosama7@gmail.com

To exit enter blank space in the name input
Name (Use identical):

...:Enter the Choice:

      [1] Main Menu      [0] Exit
```

Figure 2 add new contact

Option# [2] <<list of contacts

Options donor, the contacts which they saved in the system

```
=====
LIST OF CONTACTS
=====
Name           Phone No      Address      E-mail ad.
=====
Name   : ahmed
Phone  : 775581000
Address : taza
Email  : ahmed20@gmail.com
===== [A]-(1)

Name   : osama alward
Phone  : 774412944
Address : sanaa
Email  : alwardosama7@gmail.com
===== [0]-(1)

...:Enter the Choice:

        [1] Main Menu      [0] Exit
```

Figure 3 list of contacts

Option# [3] <<search about contacts

Search about any contacts you saved in the system

```
...:Search result for 'ahmed'
=====
...:Name       : ahmed
...:Phone      : 775581000
...:Address    : taza
...:Email      : ahmed20@gmail.com
...:1 match(s) found!
...:Try again?

        [1] Yes      [0] No
```

Figure 4 search about contact

Option# [4] <<edit contact

You can edit any things you want from (names, phone, address or email)

```
...:Edit contact
=====
...:Enter the name of contact to edit:ahmed

...:Editing 'ahmed'
...:Name(Use identical):fawaz
...:Phone:777799104
...:address:aden
...:email address:fawaz20@gmail.com

...:Enter the Choice:

        [1] Main Menu        [0] Exit
```

Figure 5 Edit contact

Option# [5] << delete a contact

```
...:DELETE A CONTACT
=====
...:Enter the name of contact to delete:ahmed

...:Enter the Choice:

        [1] Main Menu        [0] Exit
```

Figure 6 delete a contact

Option# [0] << exit from the system

Conclusion:

- The user can connect with company by using program in his phone or calling customer service
- The conclusion is that we have a better system which will help in better interaction between the blood donors and the blood banks.

Future work:

There is a scope of further improvement of the system. The system can be expanded with the following ideas:

- It is recommended to develop password and user name for every new user which prevent make confusion in the data
- At present, the company can only see the list of contacts, but in future we will also include features by connect with anther companies

References:

- <https://www.salesforce.com/eu/learning-centre/sales/contact-management/>.
- www.msdn.com is the Microsoft website which was referred to understand about windows programming and for better understanding of application development
- www.expressionblend.com to find information about the tool expression blend