Mobile Number Directory Management

Project Review - Report

Submitted by

B R ARAVIND RAGAV – CB.EN.U4CCE20008

CHILAKA SOHITH REDDY – CB.EN.U4CCE20012

SANTOSH – CB.EN.U4CCE20053

SUDHAN SARVANAN – CB.EN.U4CCE20061

VUDATTU SREYA SAHITHI – CB.EN.U4CCE20072

Section- Computer and Communication Engineering (CCE) A

Group Number: 12

Subject CODE: 19CSE102

Subject Title: Computer Programming

Problem Definition

Designing an online mobile number directory for your state, that acts as a database and enables the user to implement various tasks by collecting details of the customers: – customer name, house name, locality, district, service provider and mobile number.

Problem Decomposition

Tasks are decomposed as follows: -

1. Enter the details for a new customer.

2. Data Validations:
a. Ensuring the mobile number is exactly 10-digits.
b. Name contains only alphabets.
c. Service Provider is one from the list Airtel, Vi, Jio, BSNL.

3. Searching:
a. Given a mobile number, name of the customer.
b. List of customers who use mobile service by a particular service provider.
c. Given the name, mobile number(s) of that customer.
d. Given the name, address of that customer.

4. Update details:
a. Change in Service Provider – Mobile Number Porting.
b. Change in house name, locality and district.
c. Adding a new number for an existing customer.

5. Reports:
a. Details of a particular customer.
b. Percentage of mobile numbers in a given district for each service provider.

Header Files

```
#include <stdio.h>  // Includes I/O and File Operations Functions

#include <string.h> // Includes String Manipulation Functions

#include <stdlib.h> // Includes functions involving memory allocation, process control, conversions, etc.

#include <conio.h>  // Console I/O Function
```

Structure and File Declaration

(Customer name, house name, locality, district and service provider mobile number is of strings datatype and has been declared as a member of the structure “directory”. A file pointer “*fr” has been declared which is then used as the pointer of the file “dir.txt” to navigate through the records entered in the file.)

```
struct directory

{

   char cname[50];

   char hname[50];

   char loc[50];

   char dis[50];

   char serpro[20];

   char mobileno[20];

} d;


FILE *fr;

long int size = sizeof(d);


// stdin gives a prompt to the program that there will be an input given from the keyboard.
```

main() function

(Identified Serial and Parallel Tasks)

The main() function starts with the opening of file dir.txt, which allows for both reading and writing. This is followed by a welcome note from the team members. There are three services provided namely, admin, general and exit and the type of user is asked.

- Admin **-** Valid username and password are checked. The tasks that can be implemented as an admin is displayed in the menu. Based on the user's choice, the program proceeds to the corresponding switch() case.
- User **-** The tasks that can be implemented as an admin is displayed in the user. Based on the user's choice, the program proceeds to the corresponding switch() case.

```
int main() //Identified Serial and Parallel Tasks
{
    char another, ch;
    int choice;
    int type;
    //Opens the file for both reading and writing.
    fr = fopen("dir.txt", "rb+");

    // showing error if file is
    // unable to open.
    if (fr == NULL)
    { //Opens the file for both reading and writing.
        fr = fopen("dir.txt", "wb+");
        if (fr == NULL)
        {
            printf("\n Cannot open file.");
            exit(1);
        }
    }

    printf("\n\t\t\t\t Amrita Vishwa Vidyapeetham");
    printf("\n\t\t\t\t Department of Computer Science and Engineering");
    printf("\n\t\t\t\t 19CSE102 - Computer Programming");
    printf("\n\t\t\t\t 2020-24 Batch");
    printf("\n\t\t\t\t Second Semester");
    printf("\n\t\t\t\t Project on Mobile Number Directory Management");
    printf

("\n\n\t\t**************************************************************************************
*****\n");
    printf("\n\t\t\t\t Submitted by: CCE A Team 12");
    printf("\n\t\t\t\t B R Aravind Ragav - CB.EN.U4CCE20008");
    printf("\n\t\t\t\t Chilaka Sohith Reddy - CB.EN.U4CCE20012");
    printf("\n\t\t\t\t Santosh - CB.EN.U4CCE20053");

    printf("\n\t\t\t\t Sudhan Sarvanan - CB.EN.U4CCE20061");

    printf("\n\t\t\t\t Vudattu Sreya Sahithi - CB.EN.U4CCE20072");

    printf

("\n\n\t\t*************************************************************************************
*****\n");

    do

    {   system("cls"); //command to clear the output screen.

        printf("\t\t\t A Hearty Welcome to our Project on Mobile Number Directory Management.");

        printf("\n\n Services Offered: -");

        printf("\n 1. Admin \n 2. General");

        printf("\n 3. Exit");

        printf("\n\n Select the type of user: ");

        scanf("%d", &type);

        switch (type)

        {

        case 1: //Admin module

        {

            char againone;

            char un[50], pw[50];

            int cho;


            printf("\n Admin Login: ");

            printf("\n Admin Username: ");

            fflush(stdin);

            gets(un);

            printf("\n Admin Password: ");

            int p = 0;

            do

            {

                pw[p] = getch();

                if (pw[p] != '\r')
                {
                    printf("*");
                }
                p++;
            } while (pw[p - 1] != '\r');
            pw[p - 1] = '\0'; //Morphs the entered password with astricks to avoid visibility
            if ((strcmp(un, "Admindirectory") == 0) && (strcmp(pw, "adminsonlycanlogin") == 0))
            {
                printf("\n You have logged in as the admin.");
                do
                {
                    system("cls");
                    printf("\n Hello Admin!");
                    printf("\n\n Tasks that can be implemented: -");
                    printf("\n 1. New Registration.");
                    printf("\n 2. Change in Service Provider - Mobile Number Porting.");
                    printf("\n 3. Change in House Name, Locality and District.");
                    printf("\n 4. Adding a new number for an existing customer.");
                    printf("\n\n Enter the choice: ");
                    scanf("%d", &cho);
                    switch (cho)
                    {
                    case 1:
                        ch = 'Y';
                        while (ch == 'Y' || ch == 'y')
                        {
                            addcustdetails();
                            printf("\n Want to add another record (Y/N): ");
                            scanf("%s", &ch);
                        }
                        break;
                    case 2:
```

```
    ch = 'Y';
    while (ch == 'Y' || ch == 'y')
    {
        Mobnoport();
        printf("\n\n Do you wish to continue (Y/N): ");
        scanf("%s", &ch);
    }
    break;
case 3:
    ch = 'Y';
    while (ch == 'Y' || ch == 'y')
    {
        changeadd();
        printf("\n\n Do you wish to continue (Y/N): ");
        scanf("%s", &ch);
    }
    break;
case 4:
    ch = 'Y';
    while (ch == 'Y' || ch == 'y')
    {
        addno();
        printf("\n\n Do you wish to continue (Y/N): ");
        scanf("%s", &ch);
    }
    break;
default:
    printf("\n Invalid choice entered.");
    break;
}
printf("\n Do you wish to implement another task (Y/N): ");
scanf("%s", &againone);
```

```
        } while (againone == 'y' || againone == 'Y');
    }
    else
    {
        printf("\n Invalid credentials entered.");
    }
}
break;
case 2: //General User Module
{
    int choi;
    char againtwo;
    do
    {

        printf("\n General Login.\n Hello User!");
        printf("\n\n Tasks that can be implemented: -");
        printf("\n\n Searching:");
        printf("\n 1. Given a mobile number, name of the customer.");
        printf("\n 2. List of customers who uses mobile service by a particular service provider. ");
        printf("\n 3. Given a name, mobile number(s) of that customer.");
        printf("\n 4. Given a name, address of that customer.");
        printf("\n\n Reports:");
        printf("\n 5. Percentage of mobile numbers in given district for each service provider.");
        printf("\n 6. Details of a particular customer.");
        printf("\n\n Enter your choice: ");
        scanf("%d", &choi);
        switch (choi)
        {
        case 1:
            ch = 'Y';
            while (ch == 'Y' || ch == 'y')
    {
        custnamesea();
        printf("\n\n Do you wish to continue (Y/N): ");
        scanf("%s", &ch);
    }
    break;
case 2:
    ch = 'Y';
    while (ch == 'Y' || ch == 'y')
    {
        spdis();
        printf("\n\n Do you wish to continue (Y/N): ");
        scanf("%s", &ch);
    }
    break;
case 3:
    ch = 'Y';
    while (ch == 'Y' || ch == 'y')
    {
        mobilenumsea();
        printf("\n\n Do you wish to continue (Y/N): ");
        scanf("%s", &ch);
    }
    break;
case 4:
    ch = 'Y';
    while (ch == 'Y' || ch == 'y')
    {
        addressdisp();
        printf("\n\n Do you wish to continue (Y/N): ");
        scanf("%s", &ch);
    }
```

```
            break;
        case 5:
            ch = 'Y';
            while (ch == 'Y' || ch == 'y')
            {
                percentagedisp();
                printf("\n\n Do you wish to continue (Y/N): ");
                scanf("%s", &ch);
            }
            break;
        case 6:
            ch = 'Y';
            while (ch == 'Y' || ch == 'y')
            {
                dispdet();
                printf("\n\n Do you wish to continue (Y/N): ");
                scanf("%s", &ch);
            }
            break;
        default:
            printf("\n Invalid choice entered.");
            break;
        }
        printf("\n Do you wish to implement another task (Y/N): ");
        scanf("%s", &againtwo);
    } while (againtwo == 'y' || againtwo == 'Y');
}
break;
case 3:
    fclose(fr); //closes the file
    exit(0);    //terminates the program
    break;
```

```
        default:
           ("\n\n Do you wish to go to the main page (Y/N): ");
        }


        printf("\n Do you wish to go to the main page: ");
        scanf("%s", &another);
    } while (another == 'Y' || another == 'y');


    return 0;
}
```

addcustdetails() function

(Enter the details for a new customer.
Data Validations-
Name contains only Alphabets.
Service Provider is one from the list Airtel, Vi, Jio, BSNL.
Ensuring the mobile number is exactly 10-digits.)

This function is part of the admin module. It asks the admin user to enter the values of the members of the structure variable "d”, ensuring that the data entered are validated based on certain conditions before writing the record into the file "dir.txt".

Conditions:

(i) The Customer name entered have only alphabets, dot, and spaces and does not have any other character.
(ii) The Service Providers entered are among the given 4: "Airtel, BSNL, Vi, Jio".
(iii) The Mobile number entered to have a length of 10 digits.

If any of the above three conditions are not satisfied, the user is asked to enter the respective details again till the above conditions get satisfied thoroughly. After validating all these conditions, the structure variable "d" containing all the values of the members is written into the file "dir.txt".

```
void addcustdetails() //Enter the details for a new customer
{
    system("cls");          //command to clear the output screen.
    fseek(fr, 0, SEEK_END); //seeks the pointer to the end of the file
    char another = 'y';


    int leng, i;
```

```
printf("\n Enter the details of the new customer as asked: -");
printf("\n Customer Name: ");
fflush(stdin);
gets(d.cname);
leng = strlen(d.cname);
int redo;

redo = 1;
while (redo != 0) //condition (i)
{
    for (i = 0; i <= (leng - 1); i++)
    {
        if (!((d.cname[i] >= 'a' && d.cname[i] <= 'z') || (d.cname[i] >= 'A' && d.cname[i] <= 'Z') || d.cname[i] <= ' ' || d.cname[i] <= '.'))
        {
            redo = redo + 1;
        }
    }
    if (redo > 1)
    {
        redo = 1;
        printf("\n Enter a valid input: ");
        scanf("%s", d.cname);
        leng = strlen(d.cname);
    }
    else
    {
        redo = 0;
    }
}
printf("\n House Name: ");
fflush(stdin);
```

```
    gets(d.hname);
    printf("\n Locality: ");
    fflush(stdin);
    gets(d.loc);
    printf("\n District:");
    fflush(stdin);
    gets(d.dis);
    //condition (ii)
    printf("\n Our service providers are Airtel, Vi, Jio and BSNL. Make sure you enter the right spelling along with the uppercase for the first letter. Enter any one of your choice: ");
    scanf("%s", d.serpro);
    if (!((strcmpi(d.serpro, "Airtel") == 0) || (strcmpi(d.serpro, "Vi") == 0) || (strcmpi(d.serpro, "Jio") == 0) || (strcmpi(d.serpro, "BSNL") == 0)))
    {
        do
        {

            printf("\n Enter a service provider based on the above conditions: ");
            fflush(stdin);
            gets(d.serpro);
        } while ((strcmpi(d.serpro, "Airtel") != 0) && (strcmpi(d.serpro, "Vi") != 0) && (strcmpi(d.serpro, "Jio") != 0) && (strcmpi(d.serpro, "BSNL") != 0));
    }
    //condition (iii)
    printf("\n Mobile Number (10 DIGITS ONLY Allowed): ");

    do
    {

        fflush(stdin);
        gets(d.mobileno);
        if (strlen(d.mobileno) != 10)
            printf("\n Enter a mobile number based on the above conditions: ");
    } while (strlen(d.mobileno) != 10);


    fwrite(&d, size, 1, fr); //adds the record at the end of the file

    printf("\n The number %s has been added successfully for the customer (Mr./Ms.) %s.",

        d.mobileno, d.cname);

}
```

custnamesea() function

(Searching - Given a mobile number, name of the customer)

This function is part of the general user module. The custnamesea() asks the user to enter a mobile number, to search through the file "dir.txt". So, that it checks if the number entered by the user matches with the number in any of the customer's records after, reading with the structure variable one by one using linear search. If it matches, it displays the customer’s name who owns this number. If not, then the function prompts the user that the mobile number is not found in the directory.

```
int custnamesea() //Searching - Given a mobile number, name of the customer

{

    char num[10];

    int count = 0;

    system("cls"); //command to clear the output screen.

    // sets pointer to start

    // of the file

    rewind(fr);


    printf("\n Enter the mobile number: ");

    fflush(stdin);

    gets(num);


    while (fread(&d, size, 1, fr) == 1) //checks for the condition if the variable is able to read a record in file

    {


        if (strcmp(d.mobileno, num) == 0)

        {
```

```
      count = 1;
      printf("\n The number %s belongs to the customer (Mr./Ms.) %s.",
         num, d.cname);
    }
  }
  if (count == 0)
  {
    printf("\n There is no customer with number %s registered in the Directory.",
       num);
  }
}
```

mobilenumsea() function

(Searching - Given the name, mobile numbers of that customer)

This function is part of the general user module. The mobilenumsea() asks the user to enter a customer name, to search through the file "dir.txt". So, that it checks if the name entered by the user matches with the name in any of the customer's records after, reading with the structure variable one by one using linear search. If it matches, then displays the mobile number owned by that customer. If not, then the function prompts the user that the customer’s name is not found in the directory.

```
int mobilenumsea() //Searching - Given a name, mobile numbers of that customer
{
  char name[30];
  int count = 0;
  system("cls"); //command to clear the output screen.
  // sets pointer to start
  // of the file
  rewind(fr);

  printf("\n Enter the customer's name: ");
  fflush(stdin);
  gets(name);
  printf("\n Mobile numbers belonging to customer (Mr./Ms.) %s is/are: ",
         name);

  while (fread(&d, size, 1, fr) == 1) //checks for the condition if the variable can read a record in the file
  {

      if (strcmp(d.cname, name) == 0)
      {
          count = 1;
          printf("\n Mobile Number: %s", d.mobileno);
      }
  }
  if (count == 0)
  {
      printf("\n There are no number belonging to the customer %s in the directory.",
             name);
  }
}
```

<u>dispdet() function</u>

(Reports - Details of a particular customer)

This function is part of the general user module. The mobilenumsea() asks the user to enter a customer name, to search through the file "dir.txt". So, that it checks if the name entered by the user matches with the name in any of the customer's records after, reading with the structure variable one by one using linear search. If it matches, then displays the details of that customer. If not, then the function prompts the user that the customer’s name is not found in the directory.

```
void dispdet() //Reports - Details of a particular customer
{
    char name[30];
    int count = 0;
    system("cls"); //command to clear the output screen.

    // sets pointer to start
    // of the file
    rewind(fr);

    printf("\n Enter the customer's name: ");
    fflush(stdin);
    gets(name);
    printf("\n Details of the customer are as follows: ");

    while (fread(&d, size, 1, fr) == 1) //checks for the condition if the variable is able to read a record in file
    {
        if (strcmp(d.cname, name) == 0)
        {
            count = 1;

            printf("\n Customer Name: %s", d.cname);
            printf("\n House Name: %s", d.hname);
            printf("\n Locality: %s", d.loc);
            printf("\n District: %s", d.dis);
            printf("\n Service Provider: %s", d.serpro);
            printf("\n Mobile Number: %s/n", d.mobileno);
        }
    }
    if (count == 0)
    {
        printf("\n Please check the details again as there is no such customer with this name/mobile number registered in the directory.");
    }
}
```

spdis() function

(Searching - List of customers who uses mobile service by a particular service provider)

This function is part of the general user module. The spdis() asks the user to enter a service provider, to search through the file "dir.txt". So, that it checks if the name of the service provider entered by the user matches with the name in any of the customer's records after, reading with the structure variable one by one using linear search. If it matches, then displays the names of the customers who own mobile number that has its service provider as entered by the user. If not, then the function prompts the user that the customer’s name is not found in the directory.

```
void spdis() //Searching - List of customers who use mobile service by a particular service provider

{

   char spchk[30];

   int count = 0;

   system("cls"); //command to clear the output screen.


   // sets pointer to start

   // of the file

   rewind(fr);


   printf("\n Enter the name of the service provider: ");

   fflush(stdin);

   gets(spchk);

   printf("\n List of customer (s) using the service provider %s is / are: ",

        spchk);


   while (fread(&d, size, 1, fr) == 1) //checks for the condition if the variable is able to read a record in file

   {

      if (strcmpi(spchk, d.serpro) == 0)


      {

         count++;

         printf("\n Customer Name (Mr./Ms.) %s", d.cname);

      }

   }

   if (count == 0)

   {

      printf("\n There are no customers with %s as the service provider in the directory.",

           spchk);
```

```
    }
}
```

addressdisp() function

(Searching - Given a name, address of that customer)

This function is part of the general user module. The addressdisp() asks the user to enter a customer name, to search through the file "dir.txt". So, that it checks if the name entered by the user matches with the name in any of the customer's records after, reading with the structure variable one by one using linear search. If it matches, then displays the customer details such as house name, locality and district. If not, then the function prompts the user that the customer’s name is not found in the directory.

```
void addressdisp() //Searching - Given a name, address of that customer
{
    char name[30];
    int count = 0;
    system("cls"); //command to clear the output screen.
    // sets pointer to start
    // of the file
    rewind(fr);

    printf("\n Enter the customer's name: ");
    fflush(stdin);
    gets(name);
    printf("\n Address belonging to the customer (Mr./Ms.) %s is: ", name);

    while (fread(&d, size, 1, fr) == 1) //check for the condition if the variable is able to read a record in file
    {

        if (strcmp(d.cname, name) == 0)
        {
            count = 1;
            printf("\n House Name: %s", d.hname);
            printf("\n Locality: %s", d.loc);
            printf("\n District: %s", d.dis);
        }
    }
    if (count == 0)
    {
        printf("\n There are no name belonging to the customer %s in the directory.",
              name);
    }
}
```

percentagedisp() function

(Reports - Percentage of mobile numbers in a given district for each service provider)

This function is part of the general user module. The percentagedisp() asks the user to enter a district's name, to search through the file "dir.txt". So, that it checks if the name entered by the user matches with the name in any of the customer's records after, reading with the structure variable one by one using linear search. If it matches, then-

- using if and else statements, it checks for all the valid service providers,
- if yes then it increments a sum counter corresponding to that service provider, and
- displays the Percentage of mobile numbers in a given district for each service provider with the formula (Percentage= (sum/total)*100).

If not, then the function prompts the user that the district name is not found in the directory.

```
void percentagedisp() //Reports - Percentage of mobile numbers in a given district for each service provider
{
    char dist[50];
    float suma = 0, sumj = 0, sumv = 0, sumb = 0, sumtot = 0;
    int count = 0;
    system("cls"); //command to clear the output screen.
    // sets pointer to start
    // of the file
    rewind(fr);
    printf("\n Enter the district name:");
    fflush(stdin);
    gets(dist);
    while (fread(&d, size, 1, fr) == 1) //checks for the condition if the variable is able to read a record in file
    {
```

```
    if (strcmpi(d.dis, dist) == 0)
    {
        count++;
        sumtot++;
        if (strcmpi(d.serpro, "Airtel") == 0)
            suma++;
        else if (strcmpi(d.serpro, "Vi") == 0)
            sumv++;
        else if (strcmpi(d.serpro, "BSNL") == 0)
            sumb++;
        else if (strcmpi(d.serpro, "Jio") == 0)
            sumj++;
    }
}
if (count == 0)
{
    printf("\n There are no district belonging to the customer %s in the directory.",
           dist);
}
else
{
    printf("\n The percentage division of the Service Providers at the district %s:",
           dist);
    printf("\n Airtel Mobile Number Percentage = %.2f %%",
           ((suma / sumtot) * 100));
    printf("\n Vi Mobile Number Percentage = %.2f %%",
           ((sumv / sumtot) * 100));
    printf("\n BSNL Mobile Number Percentage = %.2f %%",
           ((sumb / sumtot) * 100));
    printf("\n Jio Mobile Number Percentage = %.2f %%",
           ((sumj / sumtot) * 100));
}
```

```
}
```

Mobnoport() function

(Update details - Change in Service Provider – Mobile Number Porting)

This function is part of the admin module. The function below asks the admin user to enter a customer name, his mobile number that needs to be ported, to search through the file "dir.txt". It also asks the user to enter the service provider to which the number needs to be ported. So, that it checks if the name and the number entered by the user matches with the name and the number in any of the customer's records after, reading with the structure variable one by one using linear search. If the conditions are validated, then the values of the members of the variable "d" are assigned to the members of the variable "di" except the member "serpro", which is read using the gets function as the member of "di" structure variable. The structure variable "di" is then written into the file. If not, then the function prompts the user that the customer's name or the mobile number is not found in the directory.

```
void Mobnoport() //Update details - Change in Service Provider - Mobile Number Porting
{
    char name[25], no[20];
    int count = 0;
    struct directory di; // different structure variable part of the same structure "directory"
    long int s = sizeof(di);
    system("cls"); //command to clear the output screen.
    // sets pointer to start
    // of the file
    rewind(fr);
    printf("\n Mobile Number Porting: -");
    printf("\n Enter the customer's name: ");
    fflush(stdin);
    gets(name);
    printf("\n Enter the mobile number that needs to be ported: ");
    fflush(stdin);
    gets(no);
    printf("\n Enter the service provider to whom your mobile number must be changed: ");
    fflush(stdin);
    gets(di.serpro);
    while (fread(&d, size, 1, fr) == 1) //checks for the condition if the variable is able to read a record in the file
    {
        if ((strcmpi(d.cname, name) == 0) && (strcmpi(d.mobileno, no) == 0))
        {
            count++;
            strcpy(di.cname, d.cname);
            strcpy(di.mobileno, d.mobileno);
            strcpy(di.loc, d.loc);
            strcpy(di.dis, d.dis);
            strcpy(di.hname, d.hname);
            /*changes the position of the file pointer from the end of the particular record,
                  to the start of the read record */
            fseek(fr, -size, SEEK_CUR);
            fwrite(&di, s, 1, fr); //here it overwrites on the previously written record.
            printf("\n Mobile number has been ported to the requested service provider.");
            break;
        }
    }
    if (count == 0)
    {
        printf("\n Please check the details again as there is no such customer with this name/mobile number registered in the directory.");
    }
}
```

changeadd() function

(Update details - Change in House name, Locality and District)

This function is part of the admin module. The function below asks the admin user to enter a customer name, his mobile number, to search through the file "dir.txt". It also asks the user to enter the address details that need to be changed i.e., house name, district, and locality. So, that it checks if the name and the number entered by the user matches with the name and the number in any of the customer's records after, reading with the structure variable one by one using linear search. If the conditions are validated then, then the values of the members of the variable "d" are assigned to the members of the variable "di", except the members that are needed to be changed, which is read using gets function as the members of "di" structure variable. The structure variable "di" is then written into the file. If not, then the function prompts the user that the customer’s name or the mobile number is not found in the directory.

```
void changeadd() //Update details - Change in House name, Locality and District
```

```
{
    char name[50], no[20];
    int count = 0;
    struct directory di;
    long int s = sizeof(di);
    system("cls"); //command to clear the output screen.
    // sets pointer to start
    // of the file
    rewind(fr);
    printf("\n Enter the customer's name: ");
    fflush(stdin);
    gets(name);
    printf("\n Enter the mobile number: ");
    fflush(stdin);
    gets(no);
    printf("\n Enter the details that must be changed: -");
    printf("\n Enter the house name: ");
    fflush(stdin);
    gets(di.hname);
    printf("\n Enter the locality name: ");
    fflush(stdin);
    gets(di.loc);
    printf("\n Enter the district name: ");
    fflush(stdin);
    gets(di.dis);
    while (fread(&d, size, 1, fr) == 1) //checks for the condition if the variable is able to read a record in the file
    {
        if ((strcmpi(name, d.cname) == 0) && (strcmpi(no, d.mobileno) == 0))
        {
            count++;
            strcpy(di.cname, d.cname);
            strcpy(di.mobileno, d.mobileno);
            strcpy(di.serpro, d.serpro);
            /*changes the position of the file pointer from the end of the record,
                  to the start of the read record */
            fseek(fr, -size, SEEK_CUR);
            fwrite(&di, s, 1, fr); //here it overwrites on the previously written record.
            printf("\n Address details has been changed for the requested customer name.");
            break;
        }
    }
    if (count == 0)
    {
        printf("\n Please check the details again as there is no such customer with this name/mobile number registered in the directory.");
    }
}
```

addno() function

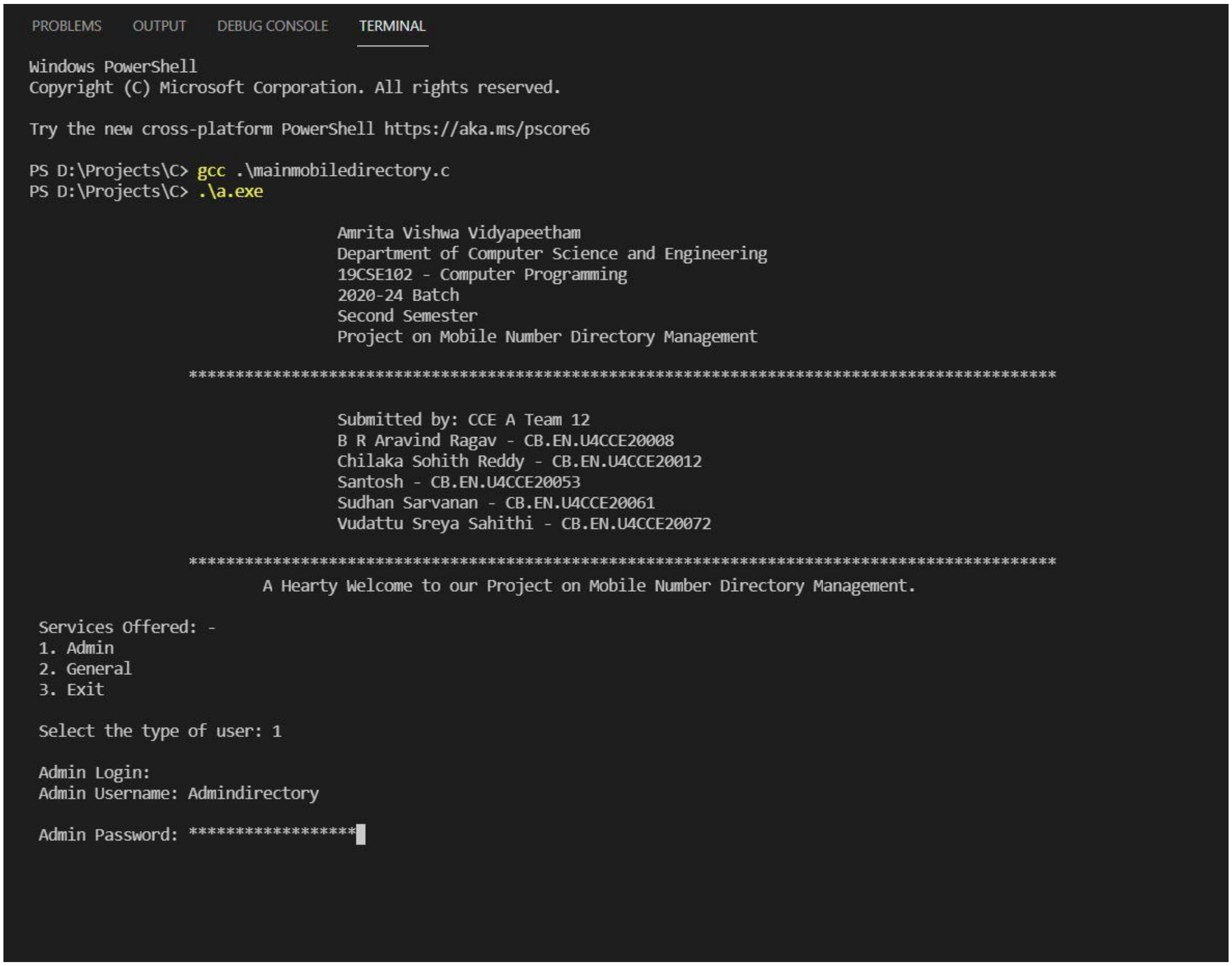
(Update details - Adding a new number for an existing customer)

This function is part of the admin module. The function below asks the admin user to enter a customer name, to search through the file "dir.txt". It also asks the user to enter the mobile number that needs to be added. So, that it checks if the name and the number entered by the user matches with the name and the number in any of the customer's records after, reading with the structure variable one by one using linear search. If the conditions are validated then, the values of the members of the variable "d" are assigned to the members of the variable "di", so that the structure variable "di" is then written as a separate record into the file. If not, then the function prompts the user that the customer’s name or the mobile number is not found in the directory.

```
void addno() //Update details - Adding a new number for an existing customer
{
    struct directory di;
    long int s = sizeof(di);
    char name[25], no[20], newno[20];
    int count = 0;
    // sets pointer to start
    // of the file
    rewind(fr);
```

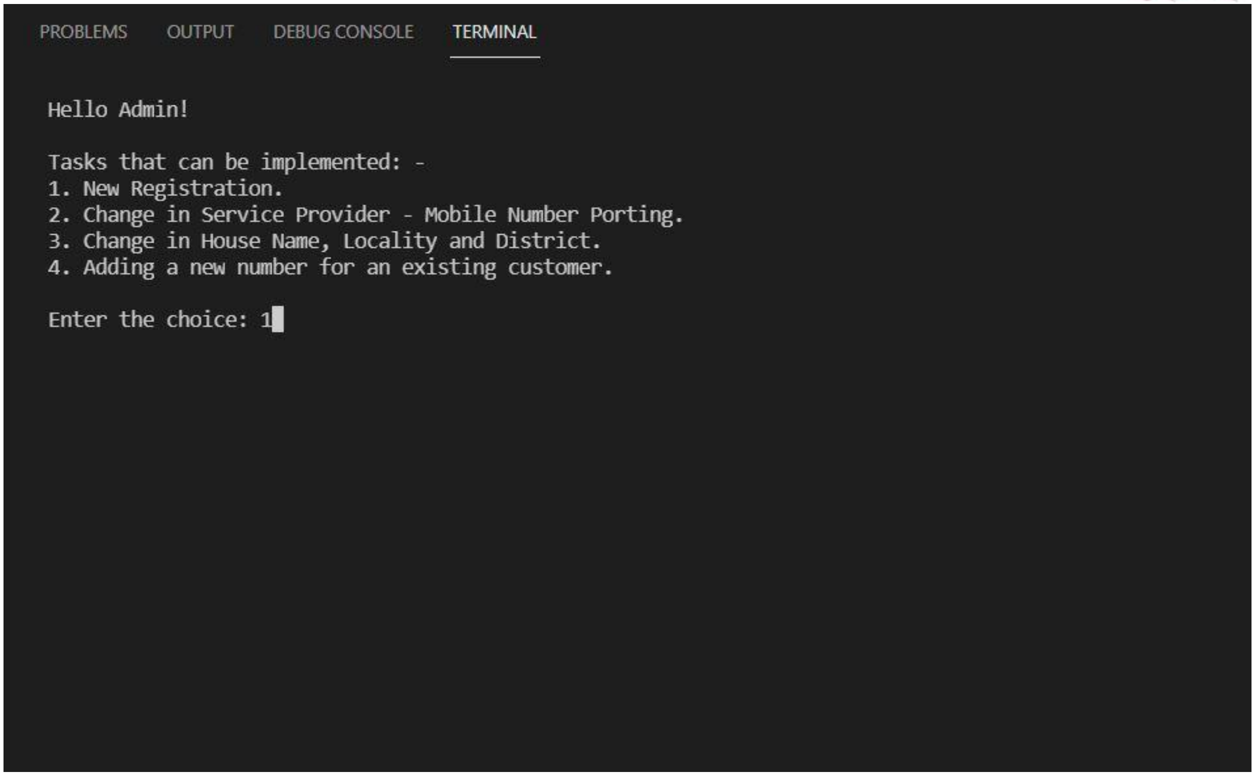
```
system("cls"); //command to clear the output screen.
printf("\n Enter the customer's name: ");
fflush(stdin);
gets(name);
printf("\n Enter the mobile number: ");
fflush(stdin);
gets(no);
printf("\n Enter the new mobile number that needs to be added: ");
fflush(stdin);
gets(di.mobileno);
printf("\n Enter the service provider of the number to be added: ");
fflush(stdin);
gets(di.serpro);
while (fread(&d, size, 1, fr) == 1) //checks for the condition if the variable is able to read a record in the file

{
   if ((strcmpi(d.cname, name) == 0) && (strcmpi(d.mobileno, no) == 0))
   {
      count++;
      strcpy(di.cname, d.cname);
      strcpy(di.loc, d.loc);
      strcpy(di.dis, d.dis);
      strcpy(di.hname, d.hname);

      fseek(fr, 0, SEEK_END); //seeks the pointer to the end of the file
      fwrite(&di, s, 1, fr);  //adds the record at the end of the file
      printf("\n New mobile number has been added to the requested customer.");
   }
}
if (count == 0)
{
        printf("\n Please check the details again as there is no such customer with this name/mobile number registered in the directory.");

    }

}
```

Test Cases



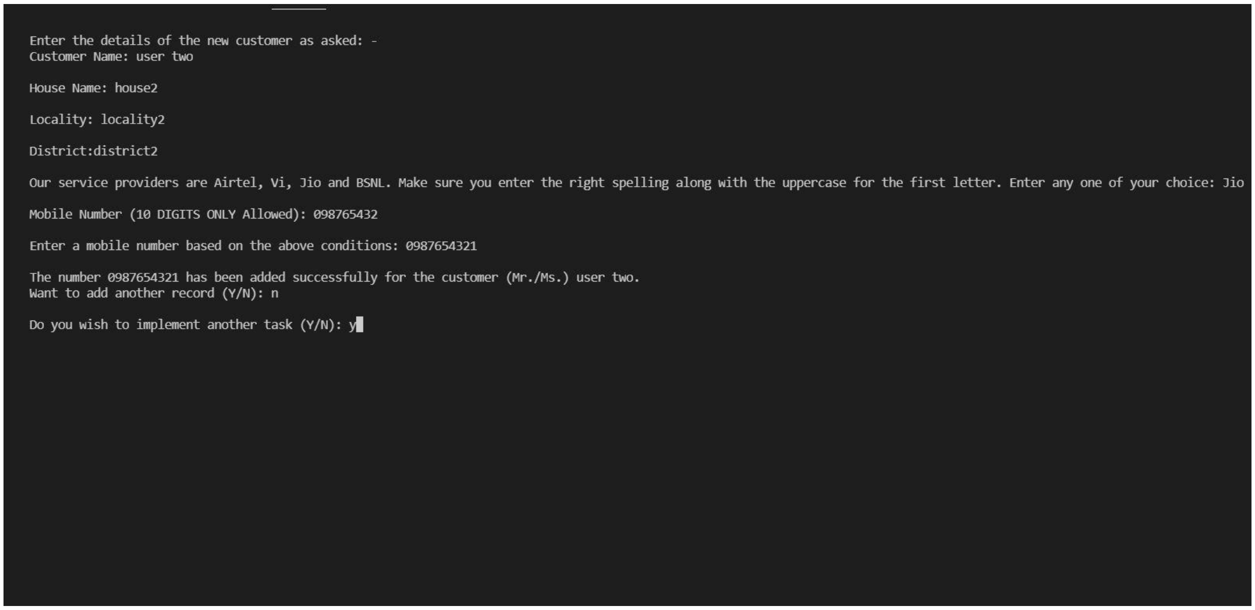
```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS D:\Projects\C> gcc .\mainmobiledirectory.c
PS D:\Projects\C> .\a.exe

                                 Amrita Vishwa Vidyapeetham
                                 Department of Computer Science and Engineering
                                 19CSE102 - Computer Programming
                                 2020-24 Batch
                                 Second Semester
                                 Project on Mobile Number Directory Management

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

                                 Submitted by: CCE A Team 12
                                 B R Aravind Ragav - CB.EN.U4CCE20008
                                 Chilaka Sohith Reddy - CB.EN.U4CCE20012
                                 Santosh - CB.EN.U4CCE20053
                                 Sudhan Sarvanan - CB.EN.U4CCE20061
                                 Vudattu Sreya Sahithi - CB.EN.U4CCE20072

                 *********************************************************************************************
                         A Hearty Welcome to our Project on Mobile Number Directory Management.

 Services Offered: -
 1. Admin
 2. General
 3. Exit

 Select the type of user: 1

 Admin Login:
 Admin Username: Admindirectory

 Admin Password: ******************
```

Figure 1 - Welcome Note



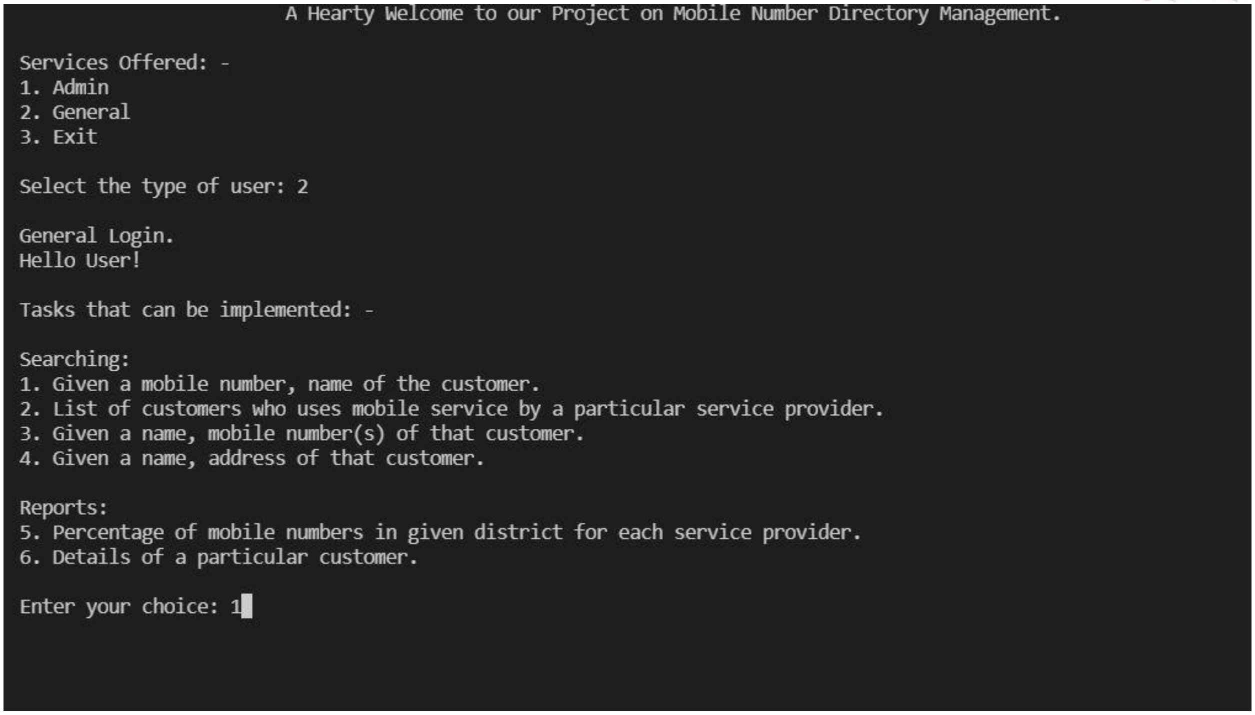
```
Hello Admin!

Tasks that can be implemented: -
1. New Registration.
2. Change in Service Provider - Mobile Number Porting.
3. Change in House Name, Locality and District.
4. Adding a new number for an existing customer.

Enter the choice: 1
```

Figure 2 - Admin User Tasks



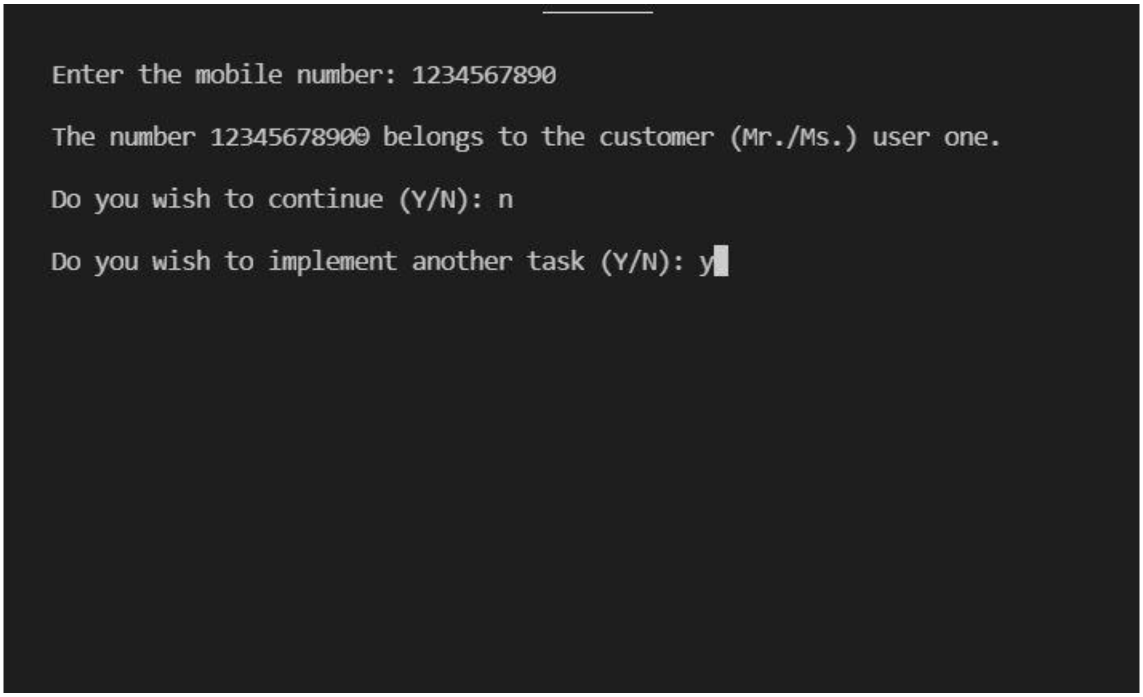
```
Enter the details of the new customer as asked: -
Customer Name: user two

House Name: house2

Locality: locality2

District:district2

Our service providers are Airtel, Vi, Jio and BSNL. Make sure you enter the right spelling along with the uppercase for the first letter. Enter any one of your choice: Jio

Mobile Number (10 DIGITS ONLY Allowed): 098765432

Enter a mobile number based on the above conditions: 0987654321

The number 0987654321 has been added successfully for the customer (Mr./Ms.) user two.
Want to add another record (Y/N): n

Do you wish to implement another task (Y/N): y
```

Figure 3 - Enter the details for a new customer and data validations



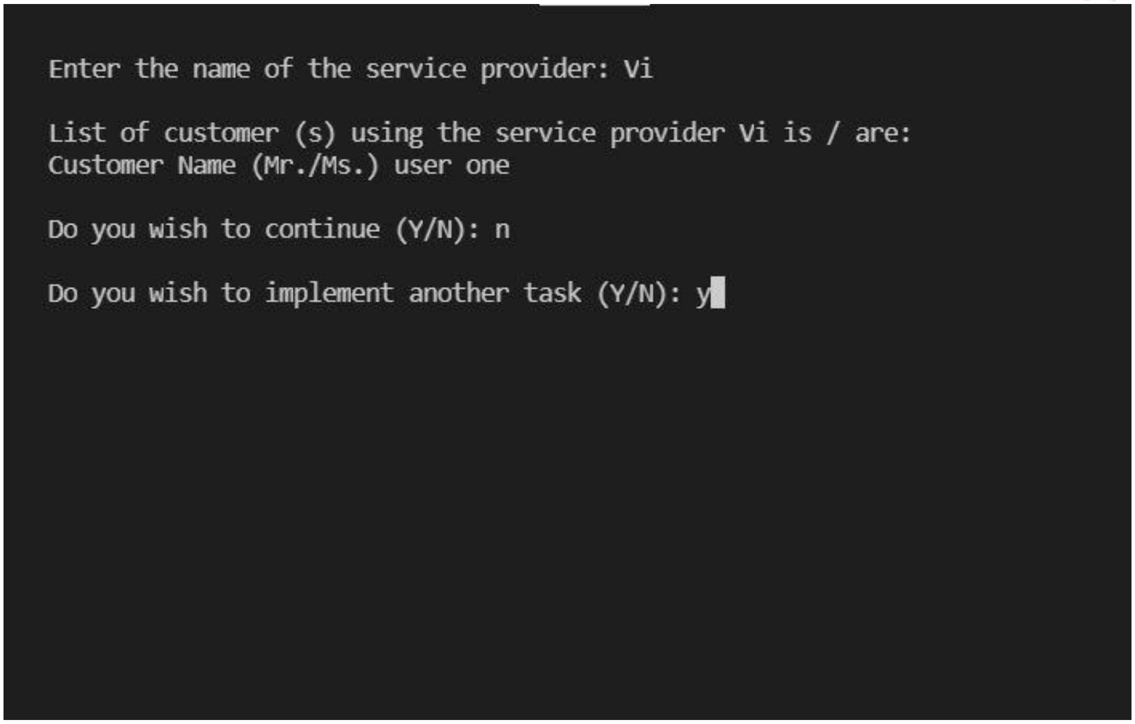
```
                            A Hearty Welcome to our Project on Mobile Number Directory Management.

Services Offered: -
1. Admin
2. General
3. Exit

Select the type of user: 2

General Login.
Hello User!

Tasks that can be implemented: -

Searching:
1. Given a mobile number, name of the customer.
2. List of customers who uses mobile service by a particular service provider.
3. Given a name, mobile number(s) of that customer.
4. Given a name, address of that customer.

Reports:
5. Percentage of mobile numbers in given district for each service provider.
6. Details of a particular customer.

Enter your choice: 1
```

Figure 4 - General User Tasks



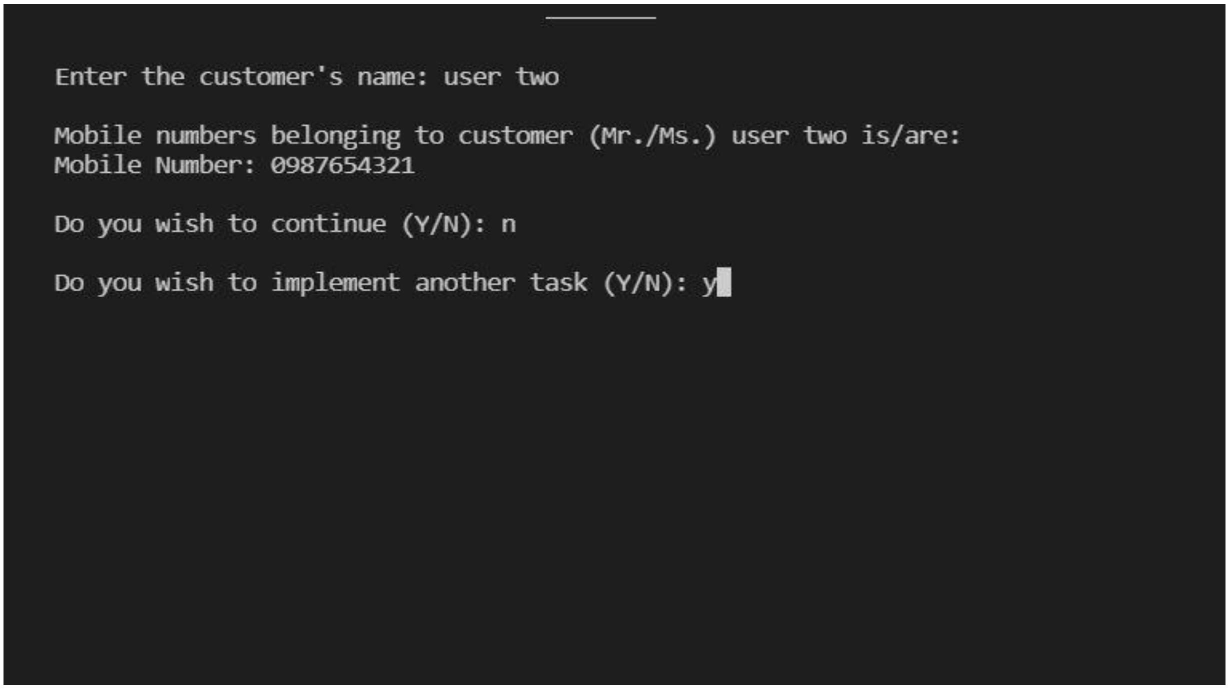
```
Enter the mobile number: 1234567890

The number 12345678900 belongs to the customer (Mr./Ms.) user one.

Do you wish to continue (Y/N): n

Do you wish to implement another task (Y/N): y
```

Figure 5 - Given a mobile number, name of the customer



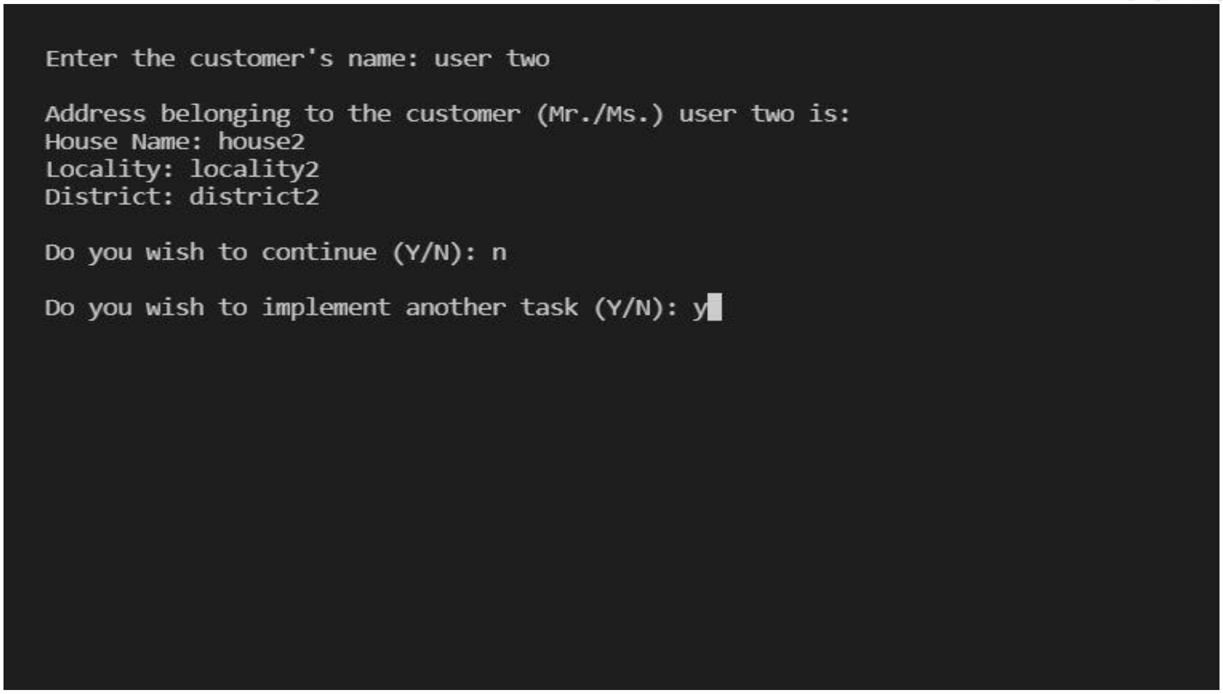
```
Enter the name of the service provider: Vi

List of customer (s) using the service provider Vi is / are:
Customer Name (Mr./Ms.) user one

Do you wish to continue (Y/N): n

Do you wish to implement another task (Y/N): y
```

Figure 6 - List of the customers who use mobile service by a particular service provider



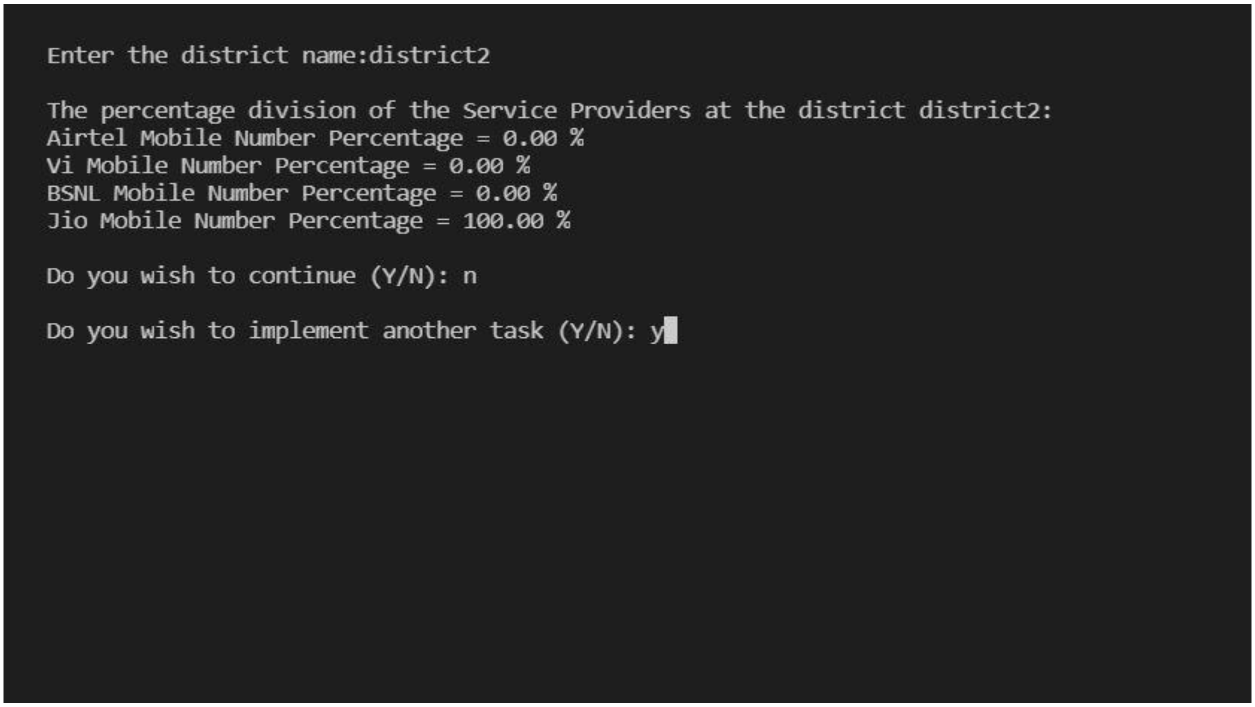
```
Enter the customer's name: user two

Mobile numbers belonging to customer (Mr./Ms.) user two is/are:
Mobile Number: 0987654321

Do you wish to continue (Y/N): n

Do you wish to implement another task (Y/N): y
```

Figure 7 - Given a name, mobile number(s) of that customer



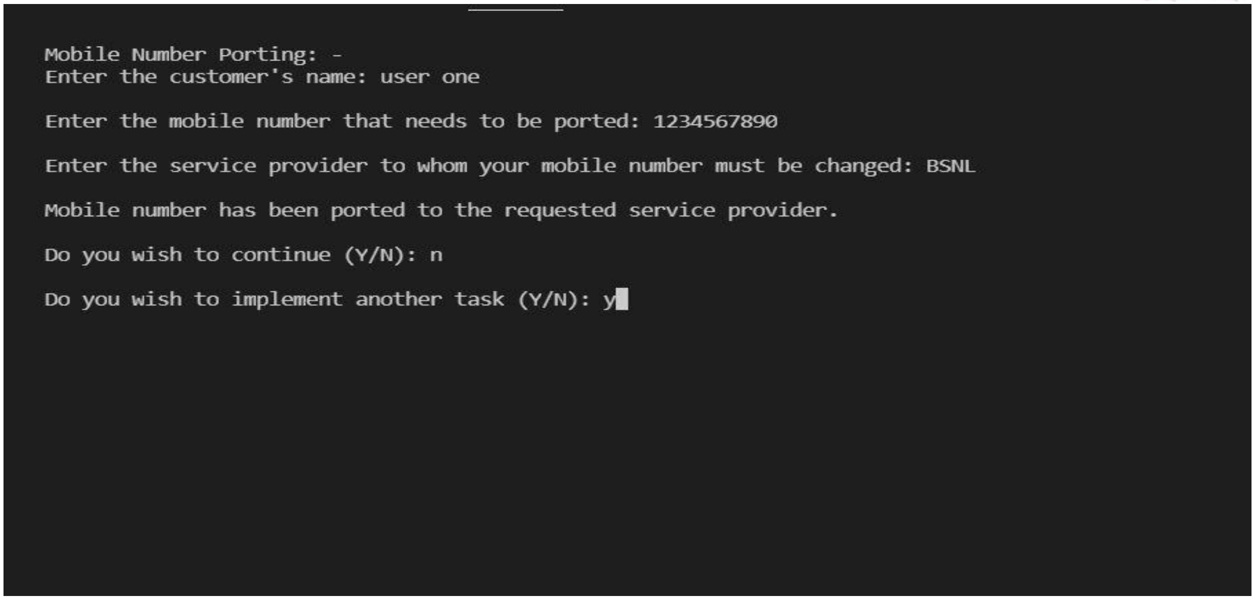
```
Enter the customer's name: user two

Address belonging to the customer (Mr./Ms.) user two is:
House Name: house2
Locality: locality2
District: district2

Do you wish to continue (Y/N): n

Do you wish to implement another task (Y/N): y
```

Figure 8 - Given the name, address of that customer



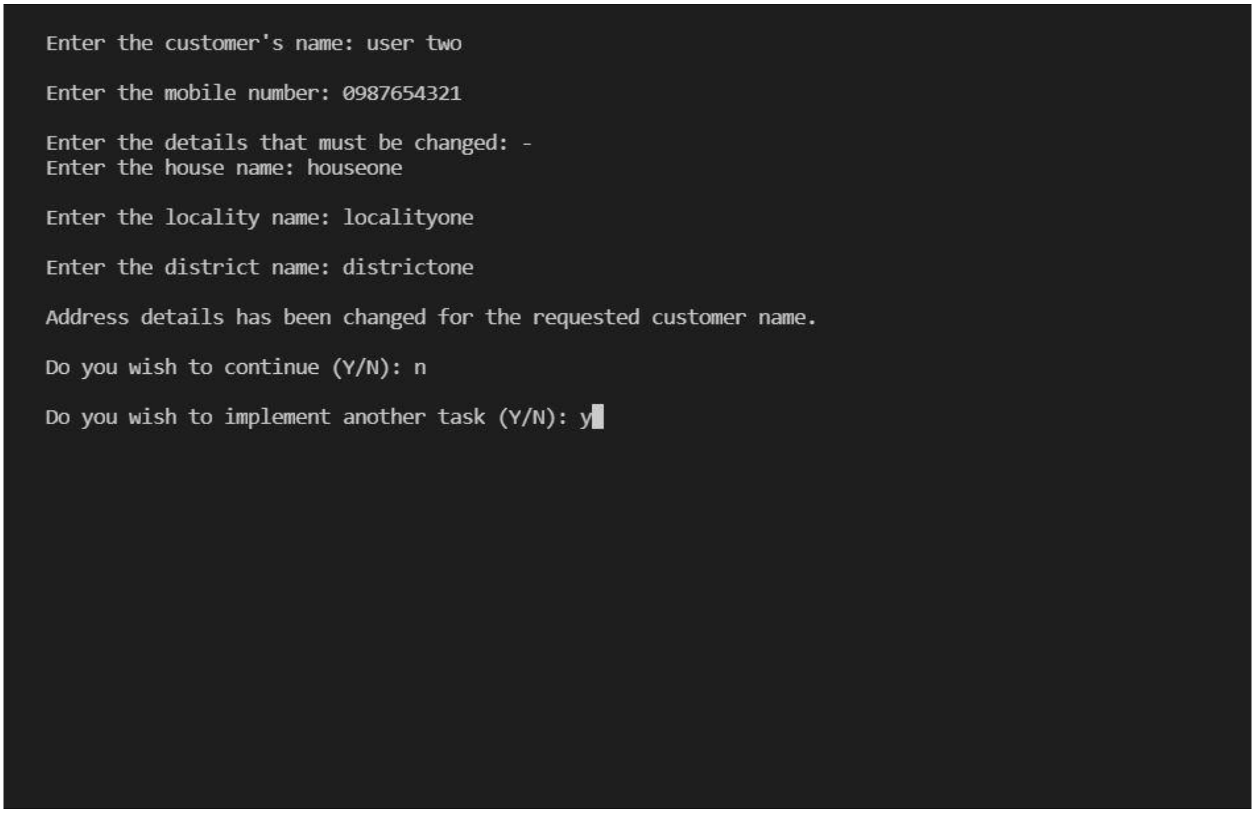
```
Enter the district name:district2

The percentage division of the Service Providers at the district district2:
Airtel Mobile Number Percentage = 0.00 %
Vi Mobile Number Percentage = 0.00 %
BSNL Mobile Number Percentage = 0.00 %
Jio Mobile Number Percentage = 100.00 %

Do you wish to continue (Y/N): n

Do you wish to implement another task (Y/N): y
```

Figure 9 - Percentage of mobile numbers in a given district for each service provider



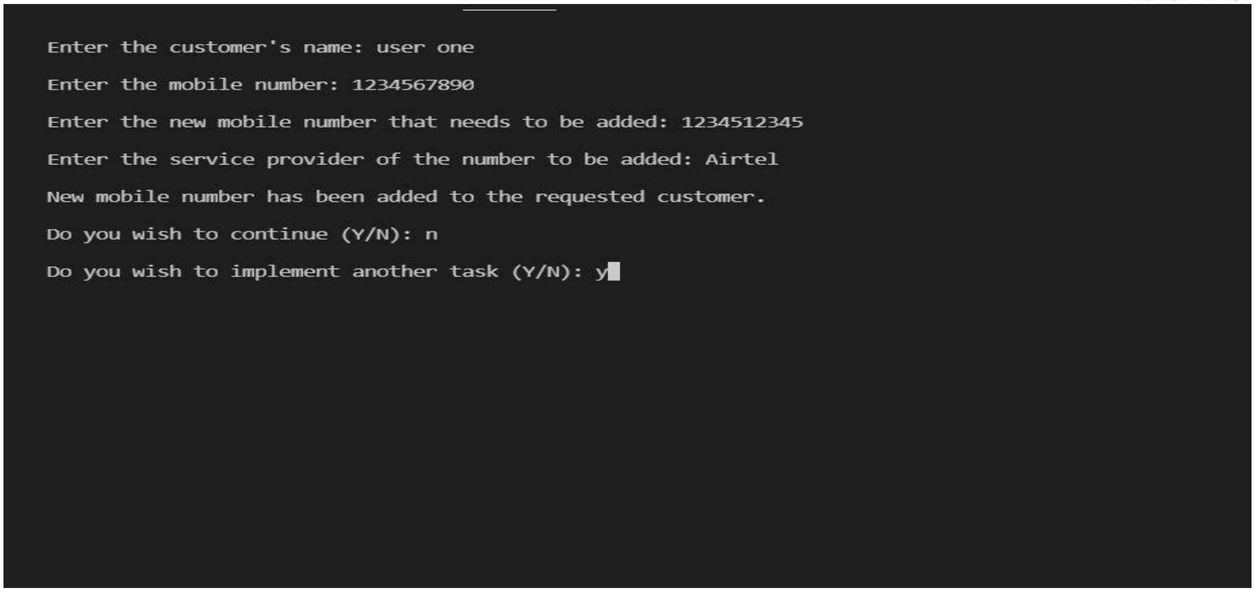
```
Mobile Number Porting: -
Enter the customer's name: user one

Enter the mobile number that needs to be ported: 1234567890

Enter the service provider to whom your mobile number must be changed: BSNL

Mobile number has been ported to the requested service provider.

Do you wish to continue (Y/N): n

Do you wish to implement another task (Y/N): y
```

Figure 10 - Change in Service Provider – Mobile Number Porting



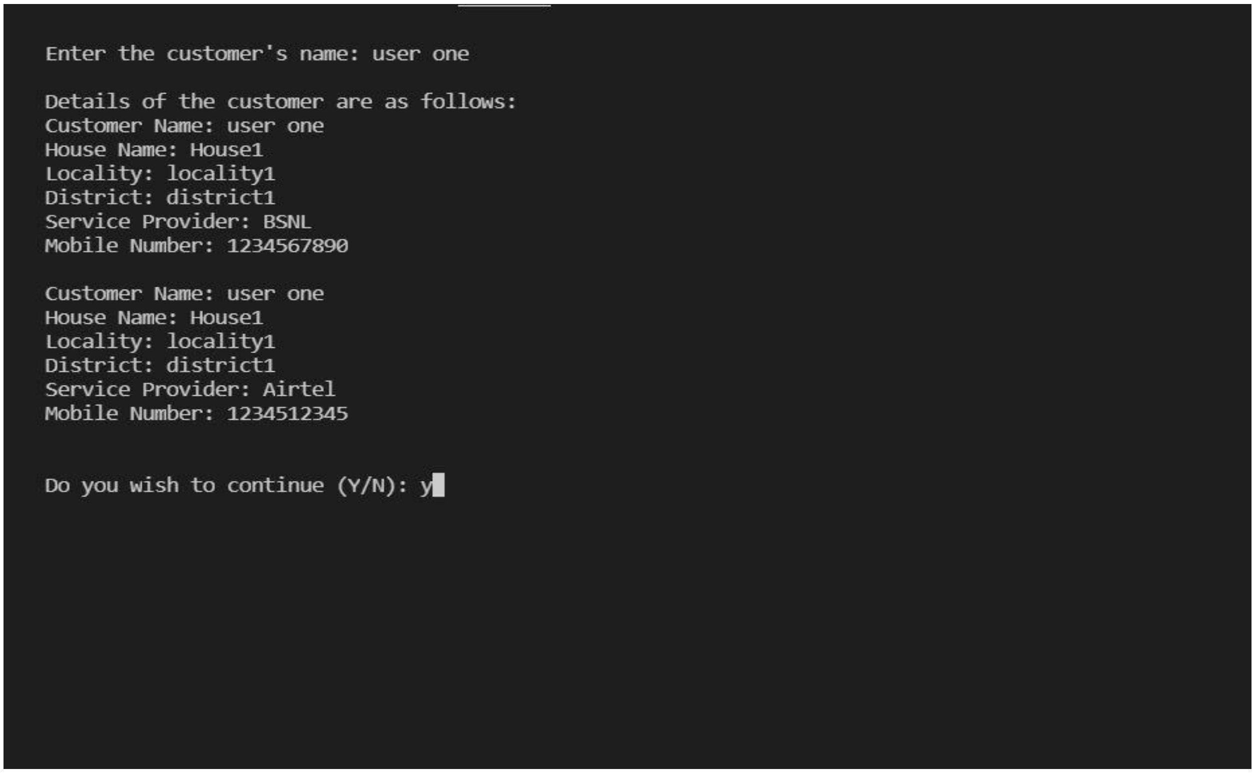
```
Enter the customer's name: user two

Enter the mobile number: 0987654321

Enter the details that must be changed: -
Enter the house name: houseone

Enter the locality name: localityone

Enter the district name: districtone

Address details has been changed for the requested customer name.

Do you wish to continue (Y/N): n

Do you wish to implement another task (Y/N): y
```

Figure 11 - Change in House Name, Locality and District



```
Enter the customer's name: user one

Enter the mobile number: 1234567890

Enter the new mobile number that needs to be added: 1234512345

Enter the service provider of the number to be added: Airtel

New mobile number has been added to the requested customer.

Do you wish to continue (Y/N): n

Do you wish to implement another task (Y/N): y
```

Figure 12 - Adding a new number for an existing customer



```
Enter the customer's name: user one

Details of the customer are as follows:
Customer Name: user one
House Name: House1
Locality: locality1
District: district1
Service Provider: BSNL
Mobile Number: 1234567890

Customer Name: user one
House Name: House1
Locality: locality1
District: district1
Service Provider: Airtel
Mobile Number: 1234512345


Do you wish to continue (Y/N): y
```

Figure 13 - Details of a particular customer

Pattern Recognition

- ✓ Usage of while loop for asking the user whether he needs to execute a particular task again.
- ✓ Linear Search is implemented in all the algorithms in the program.
- ✓ Updating of details as per the request based on the user.

Summary

Corner Stones	Topics	Sub-Topics
Problem Definition	Designing an Online Mobile Number Directory for your state, that acts as a Database and enables the user to implement various tasks by collecting details of the customers: – Customer Name, House Name, Locality, District, Service Provider and Mobile Number.	
Problem Decomposition	Enter the details for a new customer.	
	Data Validations	Ensuring the mobile number is exactly 10-digits.
		Name contains only Alphabets.
		Service Provider is one from the list Airtel, Vi, Jio, BSNL.
	Searching	Given a mobile number, name of the customer.
		List of the customers who use mobile service by a particular service provider.
		Given a name, mobile number(s) of that customer.
		Given the name, address of that customer
	Update Details	Change in Service Provider – Mobile Number Porting.
		Change in House Name, Locality and District.
		Adding a new number for an existing customer.
	Reports	Details of a particular customer.
		Percentage of mobile numbers in a given district for each service provider.
Pattern Recognition	New Registration	
	Data Validations	
	Searching	
	Update Details	
	Reports	
Identified Serial & Parallel Tasks	Display Introduction	
	Invalid Input	
	Do you want to continue?	
	Conclusion	