NEW HORIZON
COLLEGE OF ENGINEERING
Autonomous College Permanently Affiliated to VTU, Approved by AICTE & UGC
Accredited by NAAC with 'A' Grade. Accredited by NBA

nirf TOP ENGINEERING COLLEGE OF INDIA as per
NATIONAL INSTITUTIONAL RANKING FRAMEWORK, 2018
Ministry of Human Resource Development, Government of India

A

MINI PROJECT REPORT

ON

"TELECOM BILLING SYSTEM"

Submitted in the partial fulfillment of the requirements in the 3rd semester of

BACHELOR OF ENGINEERING

IN

INFORMATION SCIENCE AND ENGINEERING

BY

JNANA P J - [1NH18IS041]

FOR

COURSE NAME: MINI PROJECT

COURSE CODE :19ISE391

Under the guidance of

Prof. Mrs. A Rafega Beham
Senior Assistant Professor,
Dept. of ISE, NHCE

DEPARTMENT OF INFORMATION SCIENCE AND ENGINEERING

NEW HORIZON COLLEGE OF ENGINEERING

(Autonomous College Permanently Affiliated to VTU, Approved by AICTE, Accredited by NAAC
with 'A' Grade & NBA) Ring Road, Bellandur Post, Near Marathahalli, Bengaluru-560103, INDIA.

NEW HORIZON
COLLEGE OF ENGINEERING
Autonomous College Permanently Affiliated to VTU, Approved by AICTE & UGC
Accredited bv NAAC with 'A' Grade. Accredited bv NBA
nirf TOP ENGINEERING COLLEGE OF INDIA as per
NATIONAL INSTITUTIONAL RANKING FRAMEWORK, 2018
Ministry of Human Resource Development, Government of India

CERTIFICATE

Certified that the project work entitled **TELECOM BILLING SYSTEM** carried out by **Ms. JNANA P J**, USN **1NH18IS041**, a bonafide student of **III sem** in partial fulfillment for the award of Bachelor of Engineering in Information Science and Engineering of the Visvesvaraya Technological University, Belgaum during the year 2019-20. It is certified that all corrections/suggestions indicated for Internal Assessment have been incorporated. The project report has been approved as it satisfies the academic requirements in respect of Project work prescribed for the said Degree.

Signature of the Guide	Signature of the HOD	Signature of the Principal
[Mrs. A.Rafega Beham]	[Dr. R.J Anandhi]	[Dr.Manjunatha]

TABLE OF CONTENTS

CHAPTERS

Acknowledgement

Any achievement, be it scholastic or otherwise does not depend solely on the individual efforts but on the guidance, encouragement and corporation of intellectual, elders and friends. A number of personalities, in their own capacities have helped me in carrying out this mini project. I would like to take this opportunity to thank them all.

I thank the management Dr. Mohan Manghnani, Chairman, New Horizon educational institutions for providing necessary infrastructure and creating good environment.

I also record here the constant encouragement and facilities extended to us by Dr.Manjunatha, Principal, New Horizon College of Engineering.

I extend sincere gratitude for constant encouragement and facilities provided to us by Dr.R.J.Anandhi, Head of the Department, Department of Information Science and Engineering.

I sincerely acknowledge the encouragement, timely help and guidance to me by Mrs.A Rafega Beham, Sr.Assistant Professor, Department of Information Science and Engineering to complete the mini project within stipulated time successfully.

Finally, a note of thanks to the teaching and non-teaching staff of Information Science and Engineering Department for their corporation extended to us and our friends, who helped me directly or indirectly in the successful completion of the project.

List of figures

Abstract

Telecom Billing System project performs billing operation that is carried out by the telecommunication companies. This system is based on the concept to add, to update and maintain the daily records of the customer. Telecom companies requires an effective and accurate way to handle the records of the customer and to keep in track with their payment details. This system provides the feature to add records that contains the basic information of the customer such as name and phone number. It also provides the option to view the list of records, search the records, delete the records and shows the payment details of the customer.

The main objective of this system is to make work easier by reducing unnecessary paper works that involved in managing the records. It is difficult to maintain the records of all the customers and hence this system provides a feature to manage the records in a systematic order. It helps to keep in track with the records and hence becomes user friendly. This project is designed using C as the main programming language. Simple functions, if-else statements, switch case statements, structures in association with file handling concepts have been used.

CHAPTER 1

Introduction

Present world is more of competitive in nature and technology has become one of the important assets of it. Time is one of the most precious things and hence it is necessary to evolve with the technology that has become one of the best solutions to save time.

Telecom Billing System is designed using C that makes use of loops, switch case, structures, if-else in association with file handling concepts. This project is mainly designed for its usefulness in maintaining the records without any mistakes. This project mainly focuses to reduce unnecessary paper works that is involved in telecom companies in order to maintain day-to-day information of the customer. It becomes really a heavy task to store the details of the customer manually and is impossible to search them among thousands of records. Hence this project acts as an alternative way to maintain the customer details pertaining to their payments.

Telecom Billing Management System provides an efficient method to maintain customers payment details in telecommunication companies. This system provides the option to add, modify, delete, search the records of the customer and in addition to this there is two special features that is payment and bill generation option. So now at any time in order to get the details pertaining to specific customer, search option searches the record within minutes. Security becomes one of the important topics when it comes to managing the records. Hence this system has evolved with login feature that provides the right user to access the information of these records.

1.1 Problem Statement

Telecom companies provides its services to thousands of people and maintaining the records of these customer becomes one of the most important concern. Hence Telecom billing system provides an efficient method to reduce the workload involved in maintaining the record by providing a better platform for the user to store, modify, search the records, making payment of the customer and their bill generation.

1.2 Methodology to be followed

- In this project in order to perform any operations on the record the user must first login to the account, if by chance the username and password entered by the user is mismatched there will be at most four chances given to a person. Once the user gets logged in, there will be a list of operations which the user can perform.
- The user can add the new records with the respective phone numbers, if the entered phone number is less than ten digits then its throws an error message saying to enter the valid phone number. The user is asked to provide information about the name of the customer and the amount to be paid. The amount will be not taken manually instead an option will be provided to enter the number of std calls which will calculate the total amount a customer has to pay.
- Other than adding the records there are various option such as view the records that provide the total list of all the customers with their name and their amount, modify the record which provides the user to modify the existing record, delete the record of the customer.
- In addition to store the records, the main objective of this project is to generate bill. Hence there is a payment option which shows the user the amount a customer must pay and will provide an option to enter the money they have paid. Bill generation is also an option that generates their bill. The user can search the customer details either by their phone number or by their name.

1.3 Requirements

Software requirements:

Code blocks/turbo c++

Operating system windows 10

Hardware requirements:

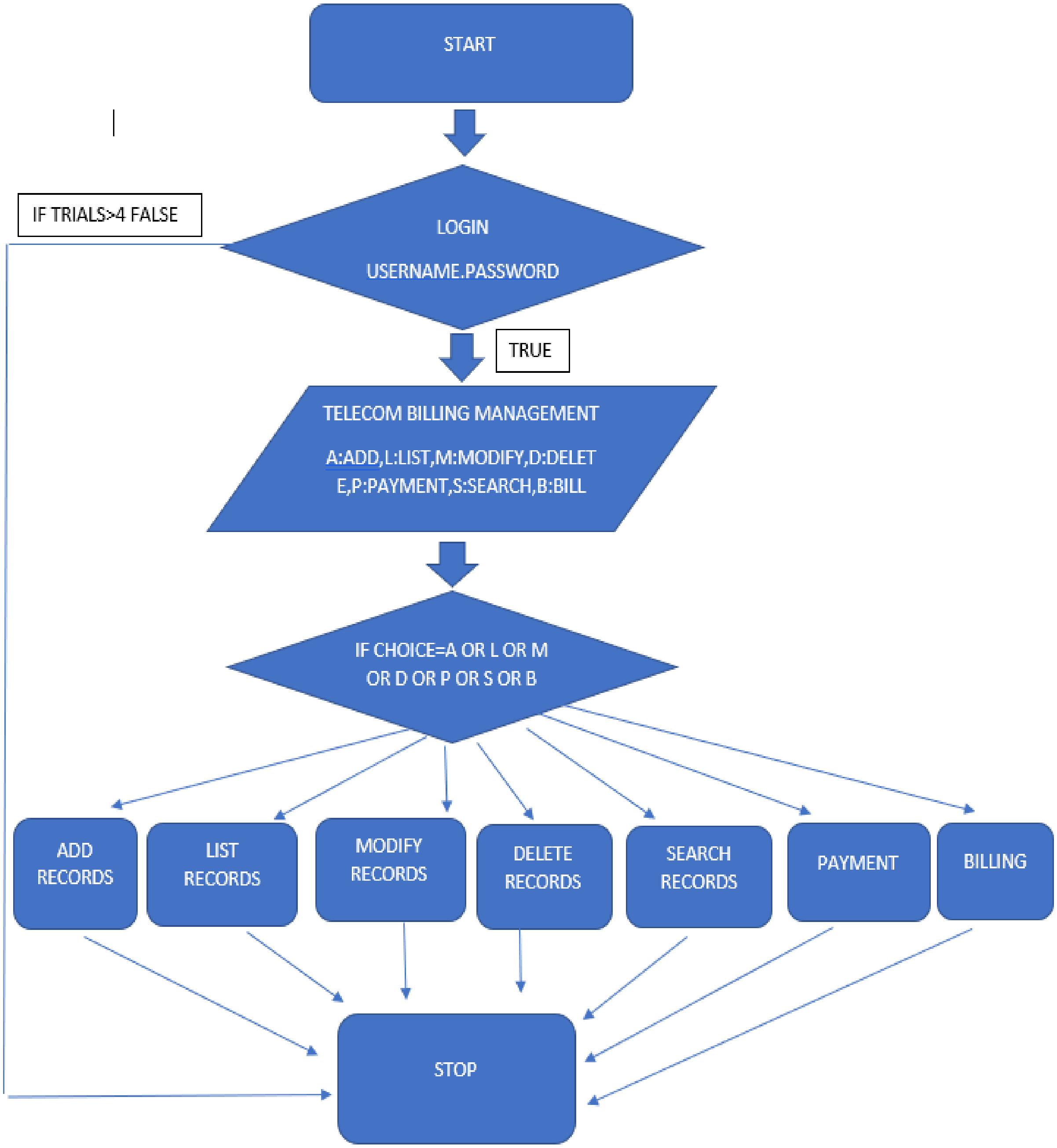
Processor: Intel core i5

RAM: 8GB (4GB can also be used)

Hard Disk:120 GB

Chapter 2

2.1 Flow chart



2.2 Algorithm

- Construct a structure that contains customers phone number, name and amount.
- Construct a login function that allows only the right user to get access to perform the operations on the records of the customer.
- Design a welcome page that asks the user to enter the username and password at most four times. If the username and password is mismatched even after four trials, pop out an error message and end the program else direct the user to the options menu.
- Ask the user to enter the option they require to perform on the records.
- Use switch case statements to perform required operation based on the choice entered by the user.
- Create a file that holds the customer details and allow the user to add necessary information of the customer.
- Once the specific operation is performed by the user return back to the homepage to ask the user if the user likes to perform any other operations.
- During the search operation ask the user whether the searching should be performed based on name or phone number and do the same.
- Use file handling concepts to link files for the operations to be performed.
- Create payment function and bill generation function such that the payment is done instantly by asking the user to enter the amount and in bill generation function, generate the customer bills specified with the date and time.
- If the user wants to end the program, ask the user to enter the required option.
- End the program.

CHAPTER 3

3.1 Switch Case Statements

The switch case in C is alternative for multiple if-else statements. Switch case takes a variable or an expression which equals to different possible values. Whichever value matches that case will be executed against a list of cases. When all the cases are false, the default statement will be executed.

The syntax is as follows:

```
switch(expression)
{
case constant1:
statement(s);
break;
case constant2:
statement(s);
break;
default:
statement(s);
 }
```

The rules to be taken into consideration while using switch cases is shown below:

- The case can accept only integer or character values and no floating values.
- The break keyword is used to separate the cases.
- Multiple cases can be used to evaluate a single statement.
- The break keyword is optional for the default statement.
- The switch case works on the equality operator.

3.2 Functions

A function is a block of code that has specific task to do. The uses of functions are 1) They can be reused.2) They make the code compact.3) It saves time.

There are two types of function:

- Default functions: These functions are already present in C library.
- User defined functions: These are the functions which are written by the user.

The syntax of the function definition is given below:

```
return_type function_name (arguments)
{
body of the function
};
```

• **Return Type**: The return type will tell the compiler what type of value the function returns when it executes. The possible return type is void, int, float, double.

• **Function Name**: It is any valid identifier.

• **Parameters**: The list of arguments are the inputs the function takes to execute. It can be empty also.

Calling a function.

When a function calls another function to execute is known as calling function and a function which is called is called as called function. When the function is called, the arguments can either be passed as the values or as the variables.

Thus, in this project eight functions are defined such as insertrecords (), listrecords (), deleterecords (), searchrecords (), modifyrecords (), payment (), telecombill (), login () and one main () function. The body of these functions will perform the required operation a user wants to perform.

3.3 File Handling

A file is a storage of some information that is present forever in the hard disk. The files can be of any form such as .c,.mp3,.txt. In C, files can be used to store the output of the program permanently on the hard disk.

Opening a file

The fopen() function opens the file by taking the name and mode as inputs. The syntax of this function call is as follows:

fopen (filename, mode);

Here, filename is the name of the file and access mode can have one of the following values:

r-used for reading a file.

w-used for writing a file.

a-used for appending a file.

While handling the binary files, the following access modes has to be used instead of the above ones: "rb", "wb", "ab".

Closing the file

To close a file the fclose () function is used. The syntax of this function is:

int fclose (FILE *fp);

The fclose () function returns zero on success, or EOF if there is an error in closing the file.

Use of fseek() function

The fseek() function is used to set the file pointer to the value of offset and starting point of the file. The syntax of fseek() function is:

fseek(FILE *stream, long int offset, int origin);

Here the offset is number bytes from origin.

There are three macros in fseek() function:

1) SEEK_CUR: Points to the start of the file.
2) SEEK_SET: Points to the current location of the file pointer.
3) SEEK_END: Points to the end of the file.

Use of rewind() function

The function of fseek () and rewind () function is quite similar in use. In fseek() function there are three different ways to point to the required point in the file i.e., it can move the pointer to the starting of the file ,it can move the pointer to the end of the file as well as it can move the pointer to the desired location in the file. Whereas in rewind() function the pointer can be moved towards the starting of the file only. That means rewind() functions performs same function as that of SEEK_CUR.

This project is mainly based upon file handling concepts. The file concepts are used to store the data of the customer that can be accessed at any time. All the function performs necessary tasks on these data keeping its base as file concepts.

3.4 If-else conditions

The if statement has two forms:

Type 1:

```
if(expression)
statement
```

Type 2:

```
if(expression)
statement1
else
statement2
```

In the first form there is only one ‘if’ statements that means if the condition is false then the statement is ignored whereas in type 2 if the condition fails then the control goes to the else statement and the else part gets executed.

There is also nested if-else condition in C:

The syntax is as follows,

```
If(condition)
{
    If(condition_sub)
{
        Statements;
}
}
else{
statements;
}
```

In this case if the condition is said to be true then the control goes to the second if statements and if the sub condition is true then the following statements gets executed otherwise the control goes to else statements.

3.5 Structures

A structure is a user defined datatype that can store a collection of values of different datatypes. It is unlike an array which can store only the elements of same datatype.

Syntax to create a structure:

```
 Struct structure_name
{
   Type 1;
   Type 2;
   Type n;
};
```

In the above declaration, struct is a required keyword followed by the name that identifies the structure, and type1, type2, --- type m are individual member declarations.

The individual members can be of any datatype. Once the structure is defined then the structure variables can be declared as follows:

```
 struct structure_name variable 1, variable 2, --- , variable n;
```

struct is a required keyword followed by the name that appeared in the structure type declaration, and variable 1, variable 2, ---, variable n are structure variables of type structure_name. Here struct_name together represents a datatype.

In the project the customer details such as name, phone number, amount is stored using a structure.

```
Ex:-   struct customer

{char phonenumber[20];

char name[50];

float amount;

}c;
```

The values of the structure can be accessed using a reference or a pointer “.”.

Syntax:

Variable_name.value

For example in the above given snippet the data such as the name of the subscriber can be accessed as c.name. The same syntax is even applicable to assign the values.

3.6 Project Code

```
#include<stdio.h>
#include<conio.h>
#include<ctype.h>
#include<windows.h>
#include<stdlib.h>
#include<time.h>

struct customer
{
char phone_number[20];
char name[50];
float amount;
}c;

void insertrecords();
void listrecords();
void modifyrecords();
void deleterecords();
void searchrecords();
void payment();
void telecombill();
void login();
char get;
int main()
{
 char choice;
 system("cls");
 system("color 0C");
```

```
printf("\n\n\n\n\n\n\n\n\n\n\t\t**********************************************
*********************************");
printf("\n\t\t\t\t------WELCOME TO THE TELECOM BILLING SYSTEM---");
printf("\n\t\t***************************************************************
****************");
Sleep(2000);
system("cls");
login();
while (1)
{
system("cls");
system("color 0F");
printf("\n********************************************");
printf("\n\n\n********************************************");
printf("\n-----------------------------------------------");
printf("\n Enter\n A -------> for adding new records.\n\n\n L -------> for list  of records");
printf("\n\n\n M -------> for modifying records.\n\n\n P -------> for payment\n\n\n B------->
for bill generation");
printf("\n\n\n S -------> for searching records.");
printf("\n\n\n D -------> for deleting records.\n\n\n E -------> for exit\n");
printf("-------------------------------------------------");
printf("\n\n\n********************************************");
choice=getche();
choice=toupper(choice);
switch(choice)
{
case 'P':
payment();break;
case 'A':
```

```
insertrecords();break;
case 'L':
listrecords();break;
case 'M':
modifyrecords();break;
case 'S':
searchrecords();break;
case 'D':
deleterecords();break;
 case 'B':
  telecombill();break;
case 'E':
system("cls");
printf("\n\n\t\t\t\tTHANK YOU");
printf("\n\n\n\n\n\t\t\tFOR USING OUR SERVICE");
Sleep(2000);
exit(0);
 break;
default:
system("cls");
 printf("Incorrect Input");
 printf("\nAny key to continue");
 _getch();
} } }
void insertrecords()
{
FILE *fp;
char test;
int calls;
```

```
fp=fopen("d:/main.txt","ab+");
while(1)
{
system("cls");
system("color 02");
do{
printf("\n\nENTER A PHONE NUMBER\n");
scanf("%s",c.phone_number);
if(strlen(c.phone_number)!=10)
{
printf("INVALID PHONE NUMBER\n\n\n");
}
}while(strlen(c.phone_number)!=10);
 system("cls");
 printf("\n ENTER NAME IN BLOCK LETTERS ONLY:");
 fflush(stdin);
 scanf("%s",c.name);
 system("cls");
 printf("\nMINIMUM Rs 200 FOR UPTO 100 CALLS\n\n\n Rs 0.60 PER CALL FOR NEXT 50 STD
CALLS\n\n\nPLUS Rs 0.50 PER CALL FOR NEXT 50 STD CALLS\n\n\nPLUS Rs 0.40 FOR
BEYOND 200 STD CALLS \n");
 printf("\n ENTER THE NUMBER OF CALLS");
 scanf("%d",&calls);
 if(calls<=100)
 c.amount=200;
 else if(calls>100 && calls<=150)
 {
   calls=calls-100;
   c.amount=200+(0.60*calls);
}
 else if(calls>150 && calls<=200)
 {
  calls=calls-150;
 c.amount=200+(0.60*50)+(0.50*calls);
 }
 else{
 calls=calls-200;
 c.amount=200+(0.60*50)+(0.50*50)+(0.40*calls);
 }
 fwrite(&c,sizeof(c),1,fp);
 fflush(stdin);
 system("cls");
 printf("\nRECORD SUCCESSFULLY ADDED");
 printf("\n PRESS ESC KEY TO EXIT OR ANY OTHER KEY TO ADD ANOTHER RECORD:");
 test=getche();
 if(test==27)
 break;
 }
 fclose(fp);
}

void listrecords()
{
FILE *fp
int i;
if((fp=fopen("d:/main.txt","rb"))==NULL)
exit(0);
system("cls");
system("color 01");
printf("Phone Number\t\tUser Name\t\t\tAmount\n");
for(i=0;i<79;i++)
printf("-");
while(fread(&c,sizeof(c),1,fp)==1)
{
printf("\n%-10s\t\t%-20s\t\tRs. %.2f /-",c.phone_number,c.name,c.amount);
}
printf("\n");
for(i=0;i<79;i++)
printf("-");
fclose(fp);
_getch();
}
void deleterecords()
{
FILE *fp,*tp;
int g=0;
char phonenumber[20];
int ch=1;
while(ch!=0)
{
 system("cls");
 system("color 06");
 printf("\nENTER THE PHONE NUMBER TO BE DELETED");
 scanf("%s",phonenumber);
 fp=fopen("d:/main.txt","rb+");rewind(fp);
  while(fread(&c,sizeof(c),1,fp)==1)
   {
```

```
   if(strcmp(c.phone_number,phonenumber)==0)
    {
     g=1;
     }
     }
     if(g==1)
     {
     tp=fopen("d:/temp.txt","wb+");
     rewind(fp);
     while(fread(&c,sizeof(c),1,fp)==1)
     {
     if(strcmp(c.phone_number,phonenumber)!=0)
     {                    fseek(tp,0,SEEK_CUR);
       fwrite(&c,sizeof(c),1,tp);
     }
     }
     fclose(tp);
     fclose(fp);
     remove("d:/main.txt");
     rename("d:/temp.txt","d:/main.txt");
     printf("\n Record deleted");
     }
     printf("\n Do you want to continue deletion of records? If no press 0 else press 1\n");
     scanf("%d",&ch);
    }
    }

void searchrecords()
{
```

```
FILE *fp;
char phonenumber[20],name[20];
int op;
fp=fopen("d:/main.txt","rb+");
if(fp==0)
exit(0);
fflush(stdin);
system("cls");
system("color 08");
printf("\nENTER THE SEARCH METHOD OPTION:\n1 FOR PHONENUMBER \n2 FOR NAME");
scanf("%d",&op);
if(op==1)
{
 system("cls");
 printf("\nENTER THE PHONENUMBER");
  scanf("%s",phonenumber);
 }
 else{
 system("cls");
 printf("\nENTER THE NAME");
 scanf("%s",name);
 }
while(fread(&c,sizeof(c),1,fp)==1)
{
if((strcmp(c.phone_number,phonenumber)==0) || (strcmp(c.name,strupr(name))==0))
{
system("cls");
printf("\n----------------------------------");
printf("\n\nRECORD FOUND ");
```

```
          printf("\n----------------------------------");
          printf("\n\nPHONENUMBER:%s\n\n\nNAME:%s\n\n\nAMOUNT:
Rs.%0.2f\n\n\n",c.phone_number,c.name,c.amount);
          printf("\n----------------------------------");
          break;
          }
          }
          _getch();
           fclose(fp);
           }
         void modifyrecords()
          {
          int calls;
          FILE *fp;
          char phonenumber[20],phone[20];
          long int size=sizeof(c);
          if((fp=fopen("d:/main.txt","rb+"))==NULL)
          exit(0);
          system("cls");
          system("color 0F");
          printf("ENTER THE PHONE NUMBER OF THE CUSTOMER TO MODIFY:\n");
          scanf("%s",phonenumber);
          fflush(stdin);
          while(fread(&c,sizeof(c),1,fp)==1)
          {
          if(strcmp(c.phone_number,phonenumber)==0)
          {
           system("cls");
          do{
```

```
 printf("ENTER A NEW PHONENUMBER\n");
 scanf("%s",c.phone_number);
  if(strlen(c.phone_number)!=10)
  {
   printf("INVALID PHONE NUMBER\n\n");
   }
   }while(strlen(c.phone_number)!=10);
   system("cls");
   printf("\nENTER THE NAME IN BLOCK LETTERS ONLY: ");
   fflush(stdin);
   scanf("%s",c.name);
   system("cls");
   printf("\nMINIMUM Rs 200 FOR UPTO 100 CALLS\n\n\nPLUS Rs 0.60 PER CALL FOE NEXT
50 STD CALLS\n\n\nPLUS Rs 0.50 PER CALL FOR NEXT 50 STD CALLS\n\n\nPLUS Rs 0.40 FOR
BEYOND 200 STD CALLS \n");
 printf("\nENTER THE NUMBER CALLS\n");
scanf("%d",&calls);
if(calls<=100)
c.amount=200;
 else if(calls>100 && calls<=150)
 {
   calls=calls-100;
   c.amount=200+(0.60*calls);
  }
  else if(calls>150 && calls<=200)
  {
   calls=calls-150;
   c.amount=200+(0.60*50)+(0.50*calls);
    }
    else{
    calls=calls-200;
    c.amount=200+(0.60*50)+(0.50*50)+(0.40*calls);
    }
    fseek(fp,-size,SEEK_CUR);
    fwrite(&c,sizeof(c),1,fp);
     break;
    }
    }
    fclose(fp);
}
void payment()
{
FILE *fp;
int op;
char phonenumber[20],name[20];
long int size=sizeof(c);
float amt;
int i;
time_t t;
time(&t);
if((f=fopen("d:/main.txt","rb+"))==NULL)
exit(0);
system("cls");
system("color 06");
printf("ENTER THE METHOD OF PAYMENT:\n1 THROUGH PHONENUMBER \n2 THROUGH
NAME");
scanf("%d",&op);
if(op==1)
```

```
 {
  system("cls");
  printf("\nENTER THE PHONENUMBER");
  scanf("%s",phonenumber);
 }
   if(op==2)
   {
    system("cls");
     printf("\nENTER THE NAME\n");
     scanf("%s",name);
     }
 fflush(stdin);
 while(fread(&c,sizeof(c),1,fp)==1)
 {
if((strcmp(c.phone_number,phonenumber)==0) || (strcmp(c.name,strupr(name))==0))
 {
 system("cls");
 for(i=0;i<79;i++)
 {
 printf("*");}
 printf("\n\n.............PAYMENT DETAILS...................   ");
 printf("\n\nDATE/TIME:%s",ctime(&t));
 printf("\n PHONE NO.     :%s",c.phone_number);
 printf("\n NAME          :%s",c.name);
 printf("\n CURRENT AMOUNT:%f",c.amount);
 printf("\n");
 for(i=0;i<79;i++)
 printf("-");
 printf("\n\nENTER THE AMOUNT FOR PAYMENT :");
 fflush(stdin);
 scanf(" %f",&amt);
 c.amount=c.amount-amt;
 fseek(fp,-size,SEEK_CUR);
 fwrite(&c,sizeof(c),1,fp);
 printf("\n\n\nTHANK YOU %s FOR YOUR TIMELY PAYMENTS",c.name);
 break;
 }
 }
 _getch();
 fclose(fp);
 }
 void telecombill()
 {
 FILE *fp;
 int op;
 char phonenumber[20],name[20];
 long int size=sizeof(c);
 float amt;
 int i;
 time_t t;
 time(&t);
 if((fp=fopen("d:/main.txt","rb+"))==NULL)
 exit(0);
 system("cls");
 system("color 06");
 printf("ENTER THE METHOD OF BILL GENERATION:\n\n1 THROUGH PHONENUMBER \n\n2
 THROUGH NAME");
 scanf("%d",&op);
```

```
if(op==1)
  {
  system("cls");
  printf("\nENTER THE PHONENUMBER");
  scanf("%s",phonenumber);
   }
   if(op==2)
   {
   system("cls");
   printf("\nENTER THE NAME\n");
   scanf("%s",name);
   }
    fflush(stdin);
   while(fread(&c,sizeof(c),1,fp)==1)
   {
 if((strcmp(c.phone_number,phonenumber)==0) || (strcmp(c.name,strupr(name))==0))
                        {
system("cls");
for(i=0;i<79;i++)
{
printf("*");}
printf("\n\n      GENERATING BILL     ");
printf("\n\n\nDATE/TIME:%s",ctime(&t));
printf("\n PHONE NO.     :%s",c.phone_number);
printf("\n NAME          :%s",c.name);
printf("\n CURRENT AMOUNT:%f",c.amount);
printf("\n\n********PLEASE PROVIDE THIS COPY DURING PAYMENT***************");
printf("\n");
for(i=0;i<79;i++)
```

```
printf("-");
printf("\n\n\nTHANK YOU %s FOR USING OUR SERVICE",c.name);
break;
}
}
_getch();
fclose(fp);
}
void login()
{
system("color 0F");
int a=0,i=0;
char username[10],c=' ';
char password[10];
char uname[10]="user";
char pass[10]="pass";
do
{
printf("\n   **************************   LOGIN  FORM   **************************
");
printf(" \n                    ENTER USERNAME:-");
scanf("%s",username);
printf(" \n                    ENTER PASSWORD:-");
while(i<10)
{
password[i]=_getch();
 c=password[i];
 if(c==13) break;
else printf("*");
```

```
 i++;
 }
password[i]='\0';
i=0;
if(strcmp(username,"user")==0 && strcmp(password,"pass")==0)
{
printf("  \n\n\n       WELCOME TO OUR SYSTEM LOGIN IS SUCCESSFUL");
printf("\n\n\n\t\tPress any key to continue...");
_getch();
break;
}
else
{
printf("\n        SORRY !!!!  LOGIN IS UNSUCESSFUL");
a++;
_getch();
system("cls");
}
}while(a<=3);
if (a>3)
{
printf("\nSORRY YOU HAVE ENTERED THE WRONG USERNAME AND PASSWORD FOR 4 TIMES!!!!!");
_getch();
exit(0);
}
system("cls");
 }
```

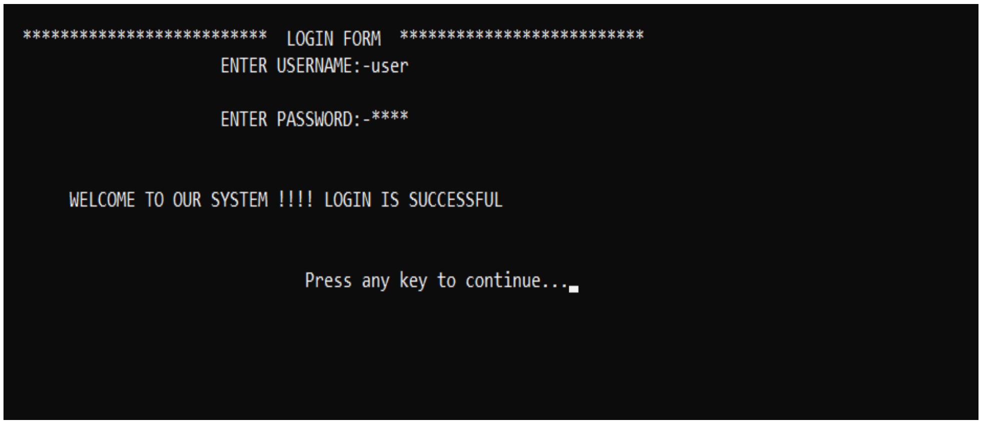
CHAPTER 4

Sample Output

- Login page.

The user has to enter the username and password and if the given details is matched then the user is allowed to perform other operation else the program exits.

4.1

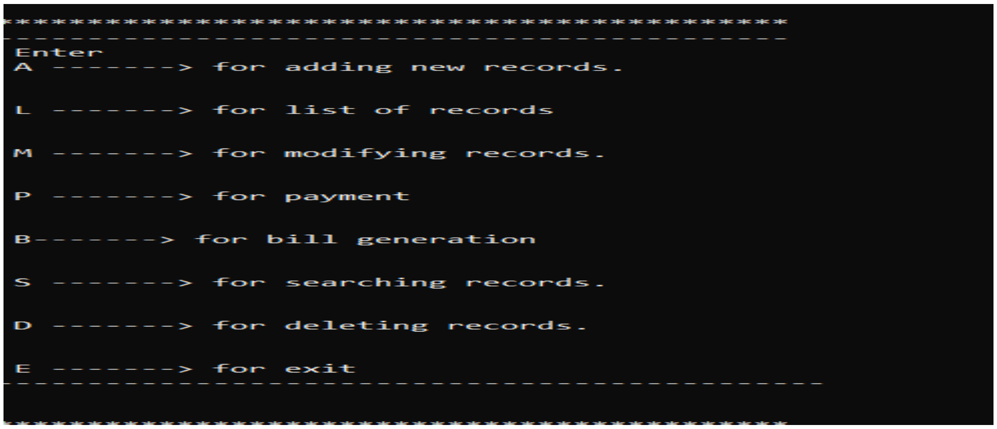


```
**************************  LOGIN FORM  **************************
                     ENTER USERNAME:-user

                     ENTER PASSWORD:-****


     WELCOME TO OUR SYSTEM !!!! LOGIN IS SUCCESSFUL


                              Press any key to continue...
```

- Homepage.

The homepage provides various options the user would like to perform.

4.2

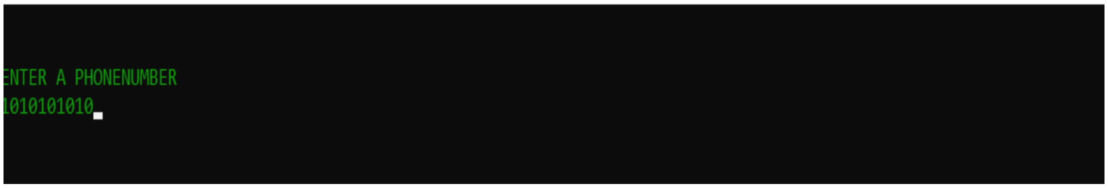


```
********************************************
--------------------------------------------
 Enter
 A -------> for adding new records.

 L -------> for list of records

 M -------> for modifying records.

 P -------> for payment

 B-------> for bill generation

 S -------> for searching records.

 D -------> for deleting records.

 E -------> for exit
----------------------------------------------

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

- Adding records of the customer.

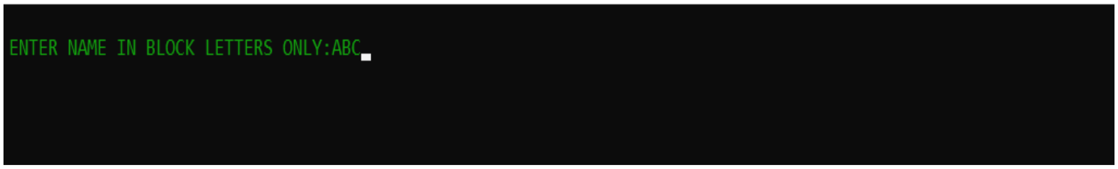
Allows the user to add records of the customer by taking their name, phone number and amount.

4.3



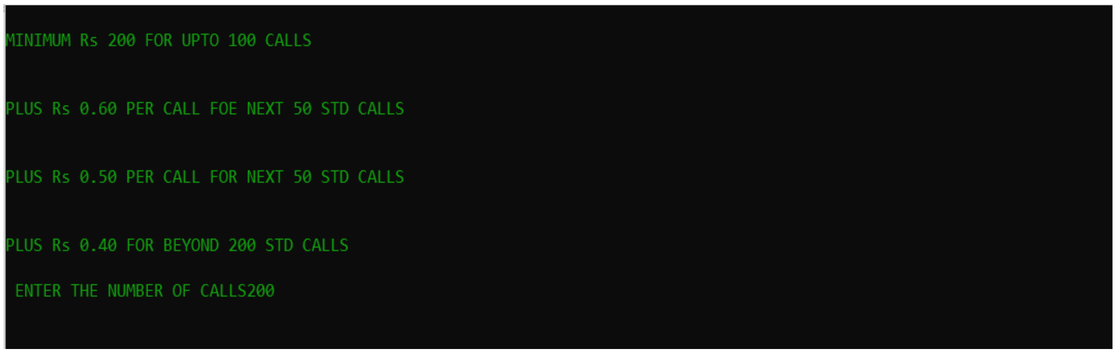
```
ENTER A PHONENUMBER
1010101010
```

4.4



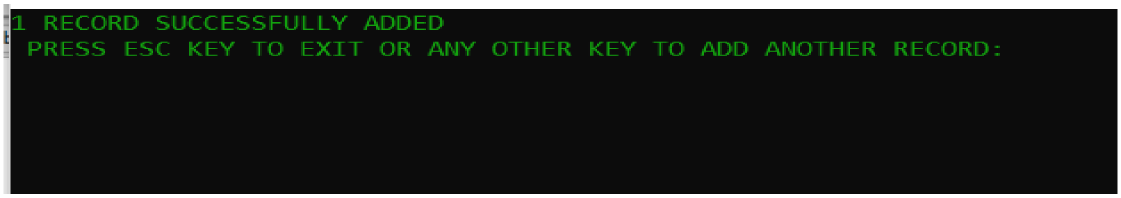
```
ENTER NAME IN BLOCK LETTERS ONLY:ABC
```

4.5



```
MINIMUM Rs 200 FOR UPTO 100 CALLS

PLUS Rs 0.60 PER CALL FOE NEXT 50 STD CALLS

PLUS Rs 0.50 PER CALL FOR NEXT 50 STD CALLS

PLUS Rs 0.40 FOR BEYOND 200 STD CALLS
 ENTER THE NUMBER OF CALLS200
```

4.6

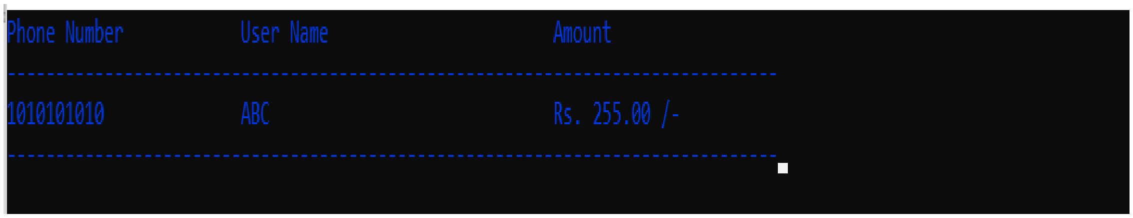


```
1 RECORD SUCCESSFULLY ADDED
 PRESS ESC KEY TO EXIT OR ANY OTHER KEY TO ADD ANOTHER RECORD:
```

- List of the records.

Show the list of the customer added by the user.

4.7

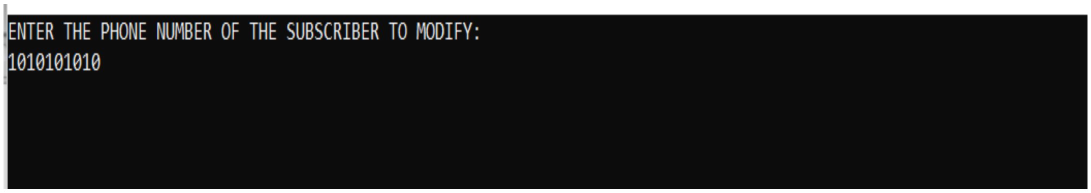


```
Phone Number          User Name                  Amount
--------------------------------------------------------------------------
1010101010            ABC                        Rs. 255.00 /-
--------------------------------------------------------------------------
```

- Modify the records.

Allows the user to modify the records of the customer using their phone number. Here the user can enter the new phone number, can change the name and the amount to be paid.

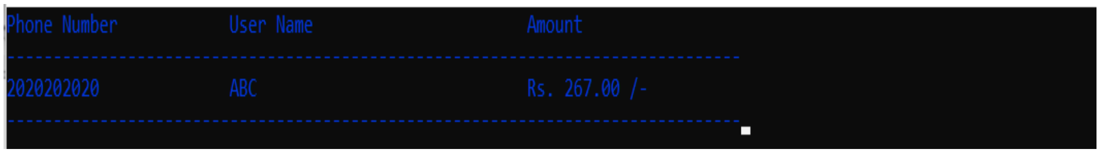
4.8



```
ENTER THE PHONE NUMBER OF THE SUBSCRIBER TO MODIFY:
1010101010
```

The above given number is modified as shown here:

4.9

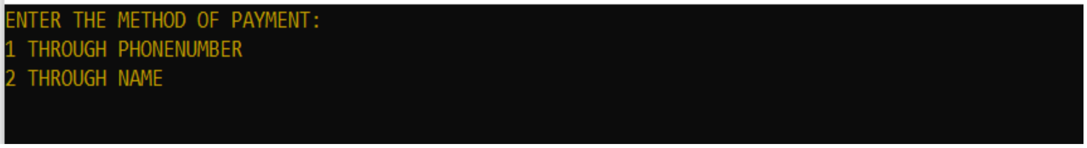


```
Phone Number          User Name                  Amount
--------------------------------------------------------------------------
2020202020            ABC                        Rs. 267.00 /-
--------------------------------------------------------------------------
```

- Payment Option

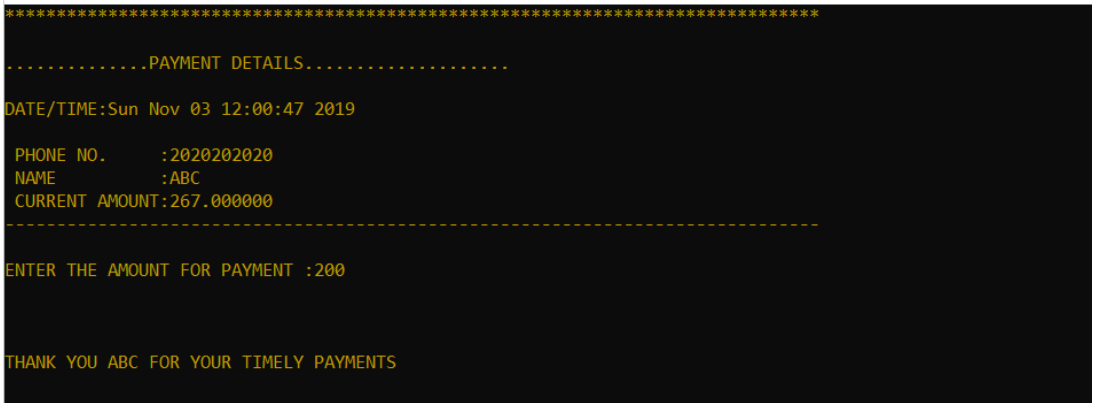
Allows the customer to perform the payment either by entering their name or phone number.

4.10



```
ENTER THE METHOD OF PAYMENT:
1 THROUGH PHONENUMBER
2 THROUGH NAME
```

4.11

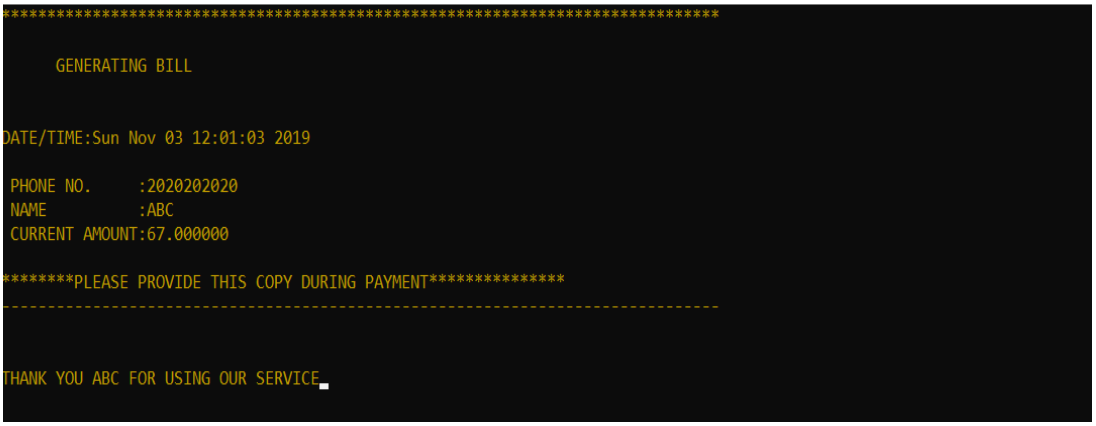


```
*******************************************************************************

.............PAYMENT DETAILS..................

DATE/TIME:Sun Nov 03 12:00:47 2019

 PHONE NO.     :2020202020
 NAME          :ABC
 CURRENT AMOUNT:267.000000
-------------------------------------------------------------------------------

ENTER THE AMOUNT FOR PAYMENT :200


THANK YOU ABC FOR YOUR TIMELY PAYMENTS
```

- Bill generation.

Allows the user to generate the bill of the customer either by the customer's name or phone number.

4.12

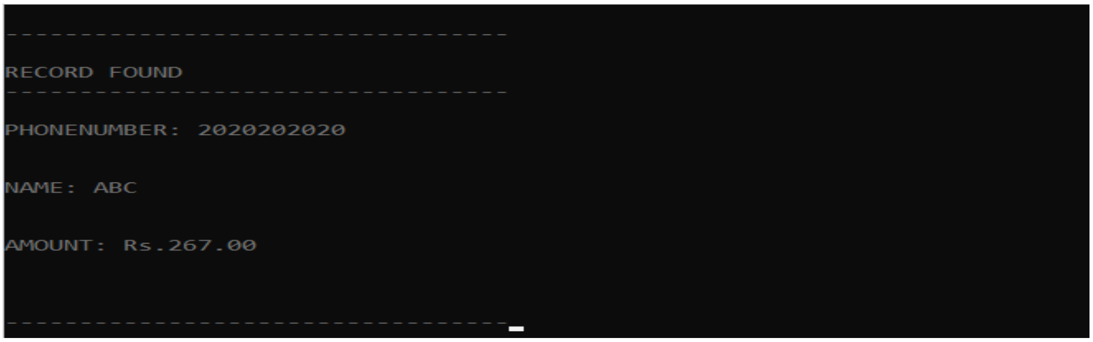


```
*******************************************************************************

     GENERATING BILL


DATE/TIME:Sun Nov 03 12:01:03 2019

 PHONE NO.     :2020202020
 NAME          :ABC
 CURRENT AMOUNT:67.000000

********PLEASE PROVIDE THIS COPY DURING PAYMENT***************
-------------------------------------------------------------------------------


THANK YOU ABC FOR USING OUR SERVICE_
```

- Search the records of the customer.

Allows the user to search the record of the customer either using their name or phonenumber.

4.13

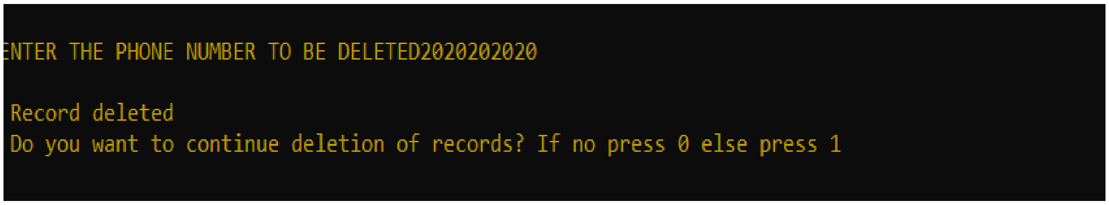


```
----------------------------------

RECORD FOUND
----------------------------------

PHONENUMBER: 2020202020


NAME: ABC


AMOUNT: Rs.267.00



----------------------------------_
```

- Deletion of the record.

Allows the user to delete the record by using their phone number.

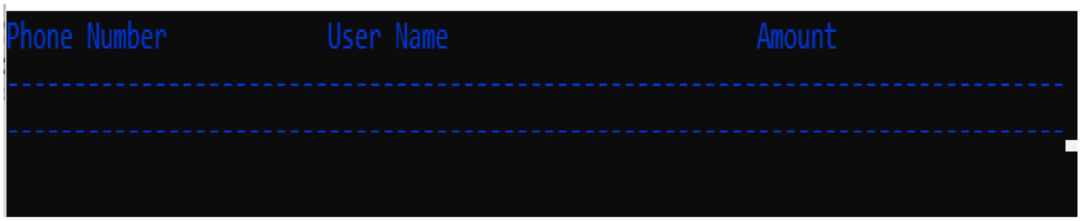
4.14



```
ENTER THE PHONE NUMBER TO BE DELETED2020202020

 Record deleted
 Do you want to continue deletion of records? If no press 0 else press 1
```

After the deletion of the record the list doesn’t contain the deleted record.

4.15



```
Phone Number            User Name                          Amount
--------------------------------------------------------------------------------

--------------------------------------------------------------------------------
```

CHAPTER 5

CONCLUSION

This project becomes one of the most efficient method to handle the records and generate the bills in telecom companies. This has made work easier by creating an efficient method in handling with the records so that now the companies need not struggle in maintaining thousands of records manually and put themselves at a risk for searching the given record of the customer. This becomes a time saviour for the telecom companies as they need not struggle in keeping track of the payment details of the customer. The records of the customer will be holding some confidential details and hence it becomes the responsibility of the company to safeguard their details. Thus, this project provides a high level of security to their documents. This system will make sure that there is efficient error free billing operation performed for both in the welfare of company as well as the customer. Finally, this project aims at time saving, making work easier and more accurate and also provides the security.

CHAPTER 5

BIBLIOGRAPHY

Introduction to programming with c (textbook) by Padma Reddy

Data structures with C by Seymour Lipschutz

Programming with C by Brian W.Kernighan and Dennis Ritchie.

www.wikipedia.com