OXFORD COLLEGE OF ENGINEERING AND MANAGEMENT

Gaindakot-2 , Nawalparasi

"TELEPHONE BILLING SYSTEM"

Submitted By:-

Name	**Symbol No.**
Bishal Dhital	17530095
Dansi Ram Acharya	17530098
Debid Rana Magar	17530099
Khusi Ram Mahato	17530108

Under Guidance

Of

Soba Raj Poudel

Submitted To POKHARA UNIVERSITY (PU)

Bachelor of Computer Application (BCA)

ROLE AND RESPONSIBILITY FORM

Name	Roles and Responsibility
Bishal Dhital	System Analysis
Dansi Ram Acharya	System Design (Algorithm, Flowchart)
Debid Rana Magar	Coding, Testing and Debugging
Khusi Ram Mahato	Typing and Documentation

S.N	Name	Symbol No.	Signature
1.	**Bishal Dhital**	**17530095**	**………………….**
2.	**Dansi Ram Acharya**	**17530098**	**………………….**
3.	**Debid Rana Magar**	**17530099**	**………………….**
4.	**Khusi Ram Mahato**	**17530108**	**………………….**

………………………….

Project Guidance

Mr. Soba Raj Poudel

…………………………

Head of Department

Mr. Suresh Baral

ABSTRACT

This project is based on "Telephone Billing System". Which is an attempt to computerized the existing billing system. This project enables users to perform all the operation that is needed for billing. According to this project users can make entry of the costumer id, displaying their information, modifying their record, deleting their record as well as search their id and payment of telephone bill. Our software has facility to give a unique id for every customer. It includes a search facility to know the current status of all customers.

The Telephone Billing System can be entered by using a password. It is accessible only to the administrative member. Only they can add, modify, and delete all the records. Thus, this system highly secure. The data can be retrieved easily. The interface is user-friendly.

This project mainly uses file handling to perform basic operations like how to add, edit, search and delete record using file. The source code of the system is over 1200 lines

ACKNOWLEDGMENT

This project is developed in order to fulfill the partial requirement of Pokhara University (PU) for the completion of BCA 2nd semester. Although this is the individual project during our study.

We are very thankful to our teacher Mr. Soba Raj Poudel for his guidance and help in study and research. His valuable suggestions were very helpful in this project. So, we would like to give some parts of the credit to Mr. Shoba Raj Poudel Sir without him this project would not be successful. And We would also like to thank our all team members for their hardwork.

TABLES OF CONTENTS

TABLES OF FIGURE

1. INTRODUCTION

“Telephone billing System” is useful for computerizing billing system. This software is useful for viewing customer information, calculating taxes, updating billing details, calculating other charges etc.

“Telephone Billing System ” is developed as per seeing the increasing requirement to speed up the work and incorporate a new work culture. Thus a new software has been proposed to reduce manual work, improving work efficiency, saving time.

1.1 Objectives

The main objective while implementing the project Telephone Billing System were to minimize the work and at the same time increase the speed of the work done.

This system is built with the following objective:

- The objective of Telephone Billing System is to handle the entire billing activity of a Telecom.
- The software keeps track of all the information about the calls and their complete user details.
- The interface of the software will be user friendly.
- The system contains files where all the information will be stored safely.

1.2 Scope of Project

This project is for those organizations who want to keep the record of telephone billing system. This program provides features to insert, display, modify, search and delete the records. This program is developed under the rules and regulations of Telecom.

The current system in use is a paper-based system. It is too slow and cannot provide updated lists of customers within a reasonable timeframe. The intentions of system are to reduce over-time pay and increase the number of customers that cab be treated accurately.

1.3 Purpose of Project

The main purpose of this project is to facilitate the processes of the Telephone Billing System. The users of this program is generally the Administration of the Telephone billing counter and this program helps them to keep records of their customers and prepare the bill on the basis of the total calls made.

1.4 Applicability of project

The released system "Telephone Billing System " can be used in any billing counter of Telecom for the purposes of recording the details of customers. This system enables its users to view list, search, update, delete and modify records so that user can trace up-to-date customer details. That is why it's a simple and easier system for all users, which is much applicable

2. REQUIREMENT ANANALYSIS

After having preliminary works investigate, our next phase was to analyze the program and find the requirement of preparing the program. We interviewed different staffs of the telecom along with the head of the organization in order to determine the requirement like:

- Who is going to use the system?
- How will they use the system?
- What data should be input and what would be output?

We include detailed information about id, name, address and phone number of customers. Finally a requirement specification document is created which serves as the purpose of guideline to the next phase.

In our project there are group of friends who are distinguished as the different post. During the interview we asked the following questions:

- How will you use the system?
- Should our system be available online or standalone?
- In which format should we make our billing system be available?
- What is the current rate of calls across the nation?
- Do you have any scheme available for your customers?

however, the head of telecommunication recommended standalone system as internet is not available in all parts of Nepal to be online.

2.1 Problem Definition

It is very important to maintain efficient software to handle information of a Billing Counter. This application enable it’s users to keeps records and information and to access these in a simple way.

2.1.1 Existing System

The existing systems provide the basic functionalities needed to be handled in a telephone billing system. There is no intelligence of the software in such cases. In the existing system all the customer records are manually prepared and kept by the receptionist or accountant in Paper form. Users was needed to calculate bill amount separately by using calculator or any calculating devices. Thus the billing was very time consuming task at all. These are the main disadvantages of the existing systems that are overcomes in the proposed model.

2.1.2 Proposed system

In our proposed system, we are going to overcome all the problems that is mentioned above by providing the facility of insertion, viewing, deletion, updating, modifying and password changing etc. Effective Search facility to search any type of information related to all customers is provided in this system which reduces the working time. Password changing facilities is also appended to this system which will assure the extra security to Administration.

2.1.3 System Development Life Cycle

System development lifecycle is the total time period of program as well as how long it would be stable into the market. Every system has a life cycle, like that the human beings, plants, new products et. This is called system development life cycle or life period of the program. The system development lifecycle covers all the terms of the system. What is the problem? Feasibility study, Analysis, Design, implementation, and maintenance, consideration of candidates system, planning and control for system success, prototyping etc.

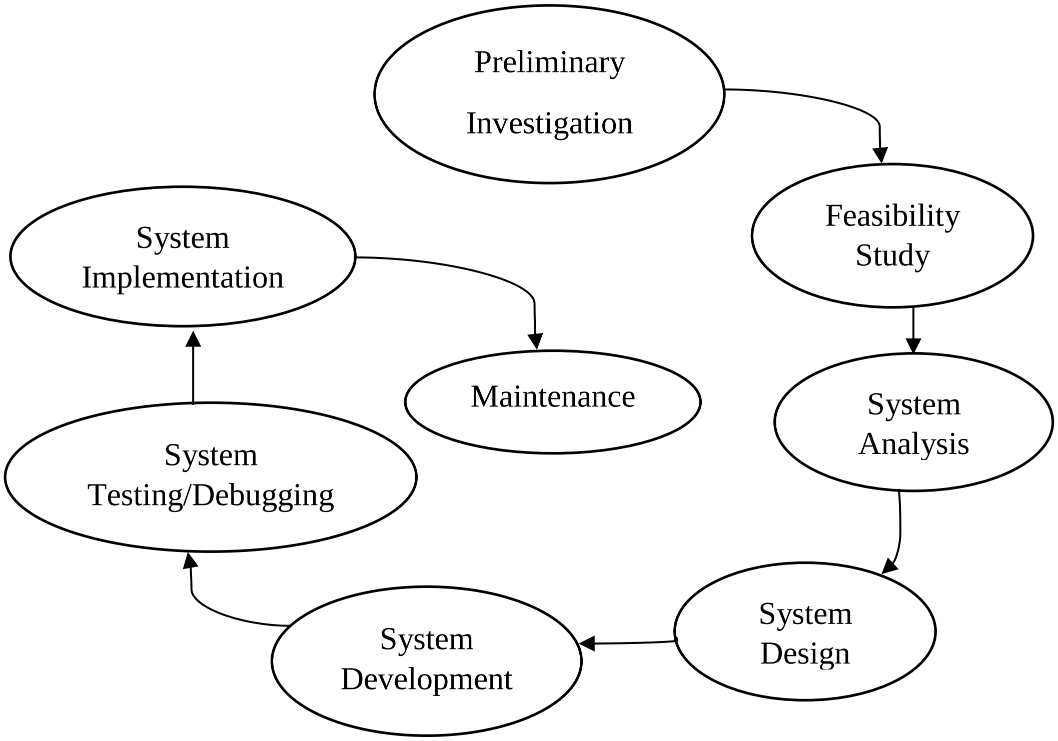


Fig 2.1:- System Development Lifecycle

2.2 Feasibility Analysis

Feasibility is the determination of whether or not a system or project is worth doing. The process followed in making this determination is called a feasibility study. So, feasibility study is to determine whether the whole process of system analysis leading to computerization would be worthwhile for organization or not. It is one of the important analysis about the program (system) which expresses the real image of the project.

The types of feasibility study can be categorized as below:

2.2.1 Technical Feasibility

During technical feasibility study, first of we all the system analyst identified the existing computer system (hardware and software) of counter and we determined that proposed “Telephone Billing System” can run easily in the application area. So, the analyst recommended that the proposed system is more feasible to be used in the Telecom counter.

2.2.2. Economical Feasibility

The proposed system “Telephone Billing System” is more cost effective than we thought because including the cost of hardware, software, development, implementation etc. the system is quite affordable for Telephone Companies. That’s why the system’s goals can be achieved within the allocated fund as well as we recommend that investing on this system can get more benefit.

2.2.3. Operational Feasibility

Now, our system is both economically and technically feasible so the next step is to determine whether it is operationally feasible or not. So, we determined to go head towards our goals because we have the experienced, skill-full and trained-manpower to develop the proposed system. That’s why the system is feasible in operational too. We are confident on developing the proposed system that the user really wants and can handle easily.

2.2.4. Social Feasibility

After investigating in the society, we decided that the proposed system will be acceptable to the people or society because after releasing this system no bad affect will arise. Such as, chance of losing the job of employees is very less as well as environment pollution in society is zero percent.

2.2.5. Management Feasibility

The proposed system will be acceptable to management because the group discussion of management section has decided to go ahead on this project and it’s feasible too.

2.3.6. Legal and contractual Feasibility

The proposed system “Telephone Billing System” is totally based on the rules & regulations of Telecom and government that will never infringe on known acts, statutes, as well as any pending legislation.

Legal effects like taxation, copyright, patent, trademarks, labor laws are also considered. Contractual affects like software and ownership, license agreement are dealt.

2.2.7. Time (schedule) Feasibility

The proposed system can be implemented fully within a stipulated time frame and the whole team will be concerned on prescribed time (schedule) and deadlines to complete whole system in required time.

2.3 Hardware and Software Requirements

To run this program we will required following hardware and software to operate the program effectively and efficiently.

2.3.1. Minimum Hardware Requirements :

- RAM – 512 MB and above
- Hard Disk – 20GB and above
- Processor Speed – 2.4GHZ and above

2.3.2 Minimum Software Requirements :

- Operating System – Microsoft Windows XP With Service Pack 2
- Dev-C++, Turbo C++ or any C-compiler.

3. SYSTEM DESIGN

System Design is the most challenging and creative part of the system development life cycle, where all the logic of all developers come at a place. This phase provides concept of the entire program.

Software design is prepared from the requirement specifications. It helps in specifying hardware and system requirements and also helps in defining overall system. It serves as input for next phase.

System design is rough sketch of any system or program in a piece of paper. Our design included selecting the font color background how the data will be seen after user gives command. We developed a menu driven program which shows insert, display, modify, search, delete, prepare bill and change password etc.

Our design is done on the following basis:

3.1 Algorithm
3.2 Flowchart

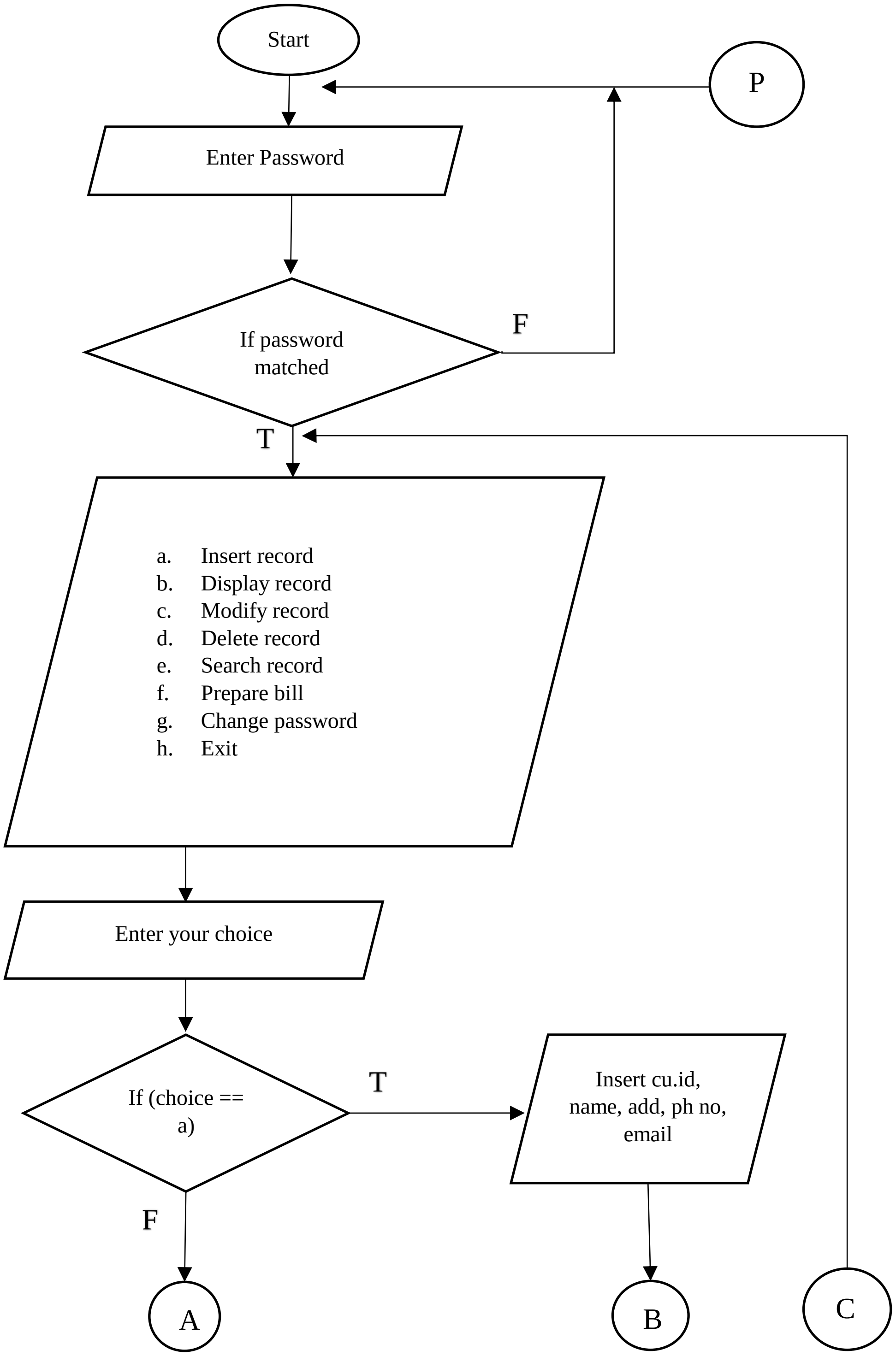
3.1 Algorithm

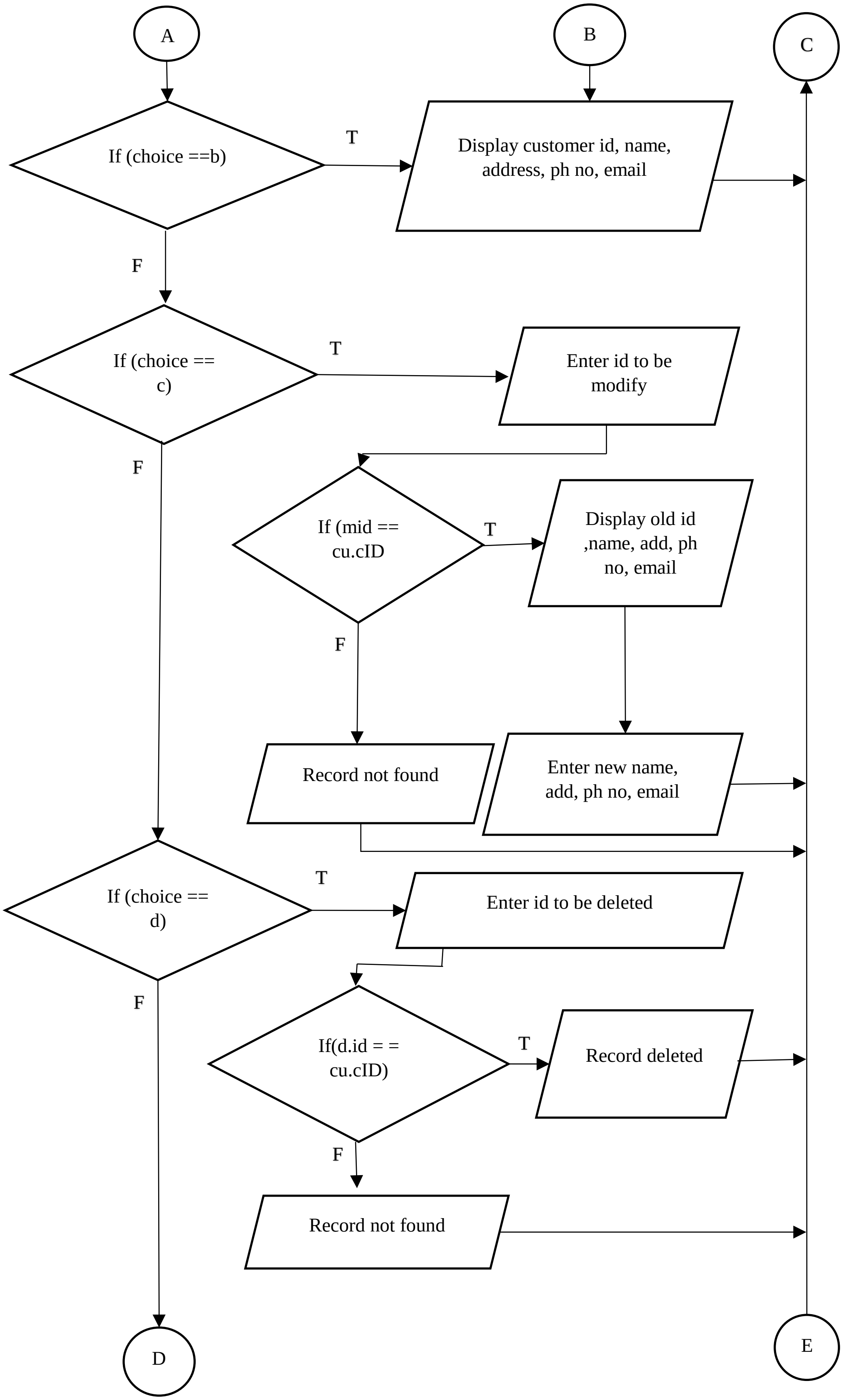
```
Step 1: Start
Step 2: Administration Login
Step 3: Enter a password
Step 4: if ( password  matched )
        Display Administration  Menu :
Insert Customer Record
Display Customer Record
Modify Customer Record
Delete Customer Record
Search Customer Record
Prepare Customer Bill
Change Administration Password
Exit
          Otherwise goto step 2
Step 5: Enter a choice
Step 6: if (choice = = a)
        Read customer ID, name, address, phone, email then go to step 7.
Step 7: if (choice = = b)
        Display customer ID, name, address, phone, email,
        Then goto step 4
Step 8: if (choice = = c)
        Read id to be modified
        if (mid = = cu.cID)
        Display record matched then read name, address, phone, email
        and goto step 4.
        Otherwise display “record not found”
        Then goto step 8.
```

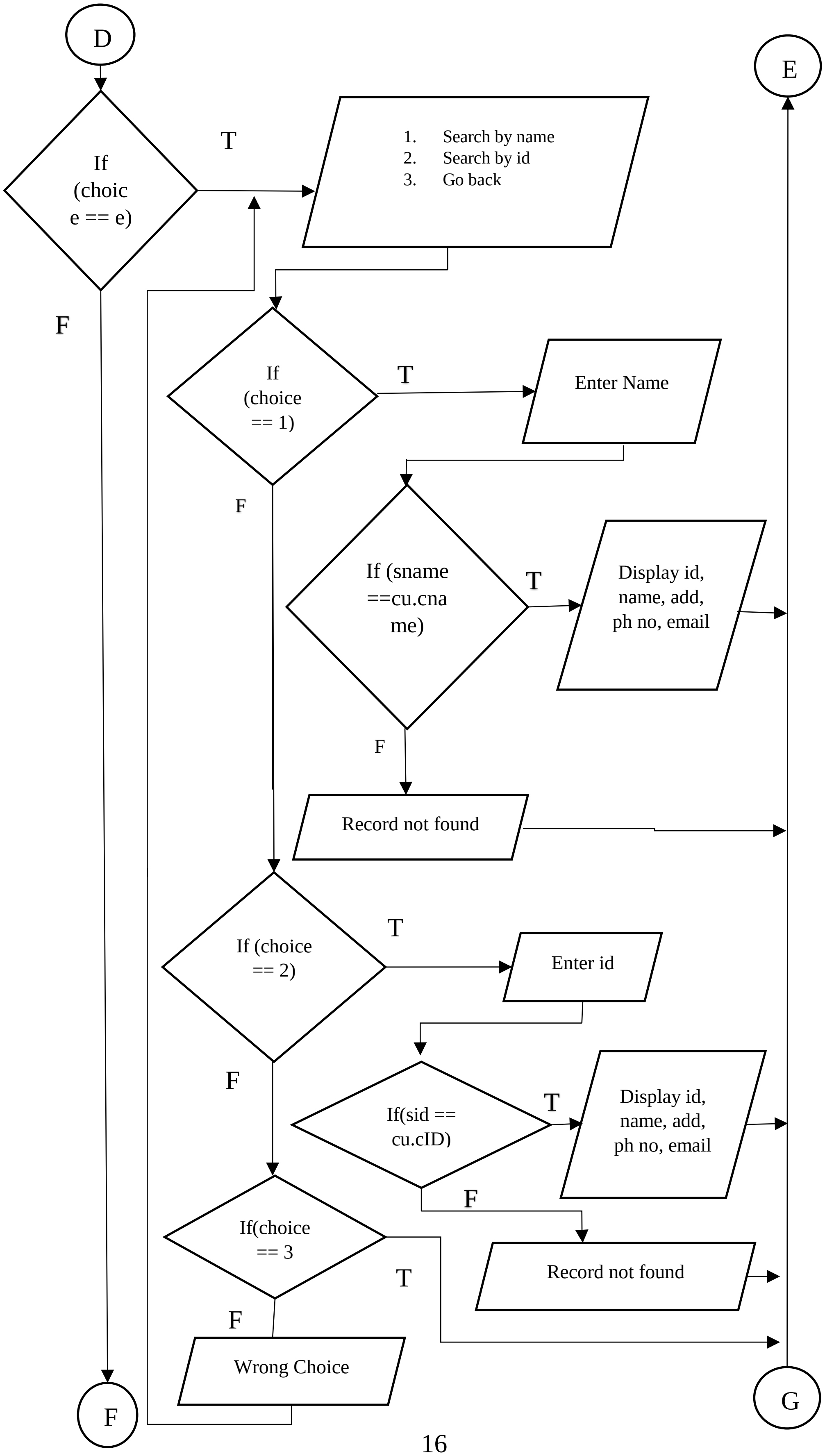
```
Step 9; if  (choice = = d)
    Read id to be deleted then
    if (did = = cu.ID)
    Display "Record deleted successfully"
    Otherwise display "Record not found"
    Then goto step 4.
Step 10: if ( choice = = e)
Search by Name
Search by Id
Goto back
     if (choice = = 1)
         Read name to be searched
         if  (cu.cName = = sname) then
        display record found, id, name, address, phone no, email.
        Otherwise
        display record not found then goto step 4.
        if ( choice = = 2)
        Read id to be searched
    if (cu.cID = = sid)
    then display record found, id, name, address, phone no, email.
    Otherwise
    display record not found then goto step 4.
    if ( choice==3)
     goto step 4.
 Otherwise
Display wrong choice and goto step 4.
Step 11: if (choice == f)
      Read id to be prepare bill
```

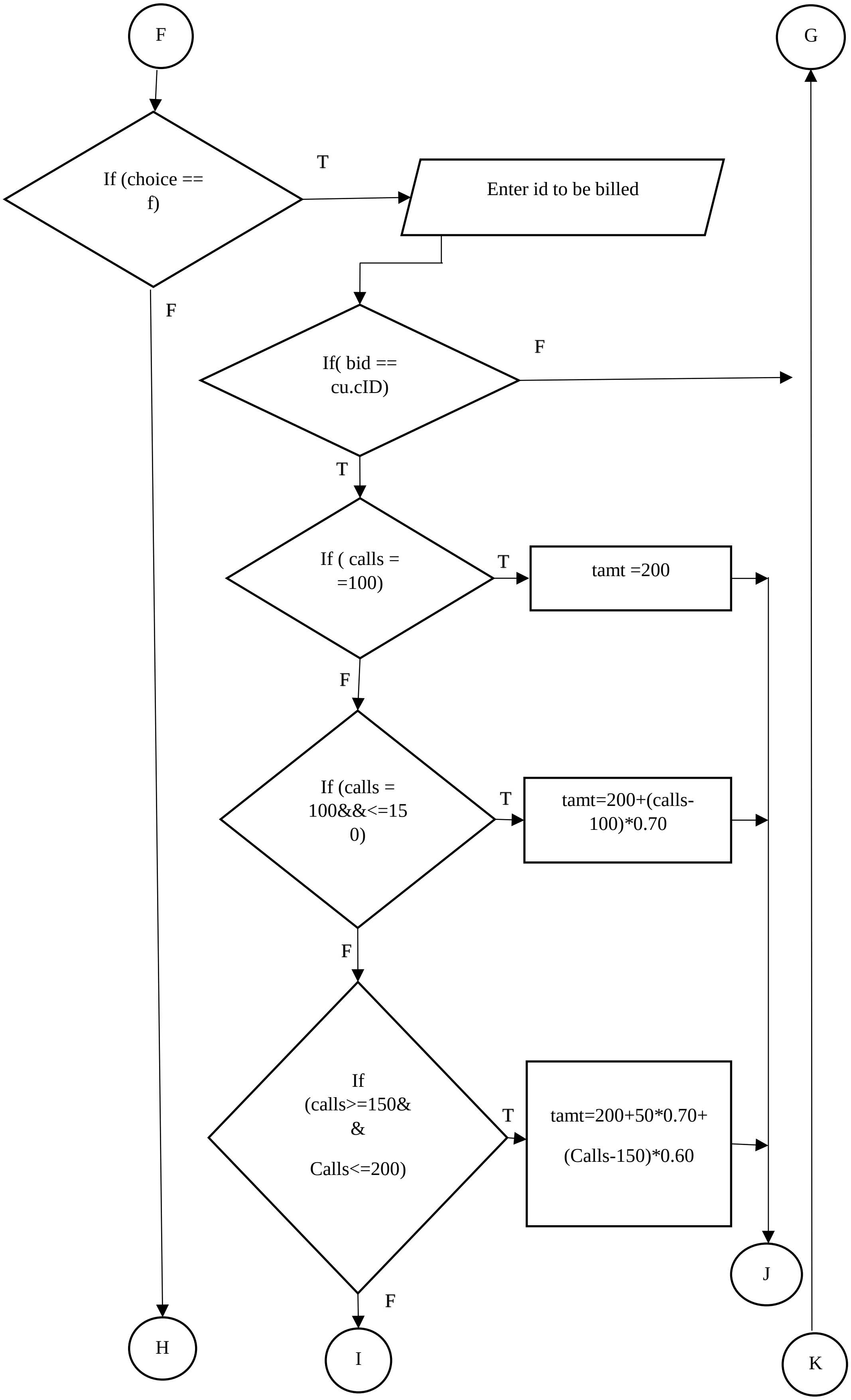
```
            If (bid == cID)
            Then read tax invoice no, total no of calls, payment mode,
            and if(calls<=100)
            tamt=200
            else if(calls>100 && calls<150)
            tamt=200+(calls-100)*0.70
            else if(calls>=150 && calls<=200)
            tamt=200+50*0.70+(calls-150)*0.60
            else if(calls>=200 && calls<=250)
            tamt=200+50*0.70+50*0.60+(calls-200)*0.50
  else
 tamt=200+50*0.70+50*0.60+50*0.50+(call-250)*0.40
  display tax invoice no, customer ID, name, address, phone, email,
  date, payment mode and total amount.
  Otherwise customer id did not match.
Step 12: if (choice = = g)
            Read old password
            if (opass = = cu.pass)
            Read new password, confirm password
                     if npw == cpw)
          display password Change successful. Then goto step 2.
          Otherwise display confirm password did not match. Goto step 12.
            Otherwise display old password did not matched, then goto step 11.
Step 14: if (choice == h)
            goto step15.
            Otherwise display wrong choice then goto step 4.
Step 15:  Stop.
```

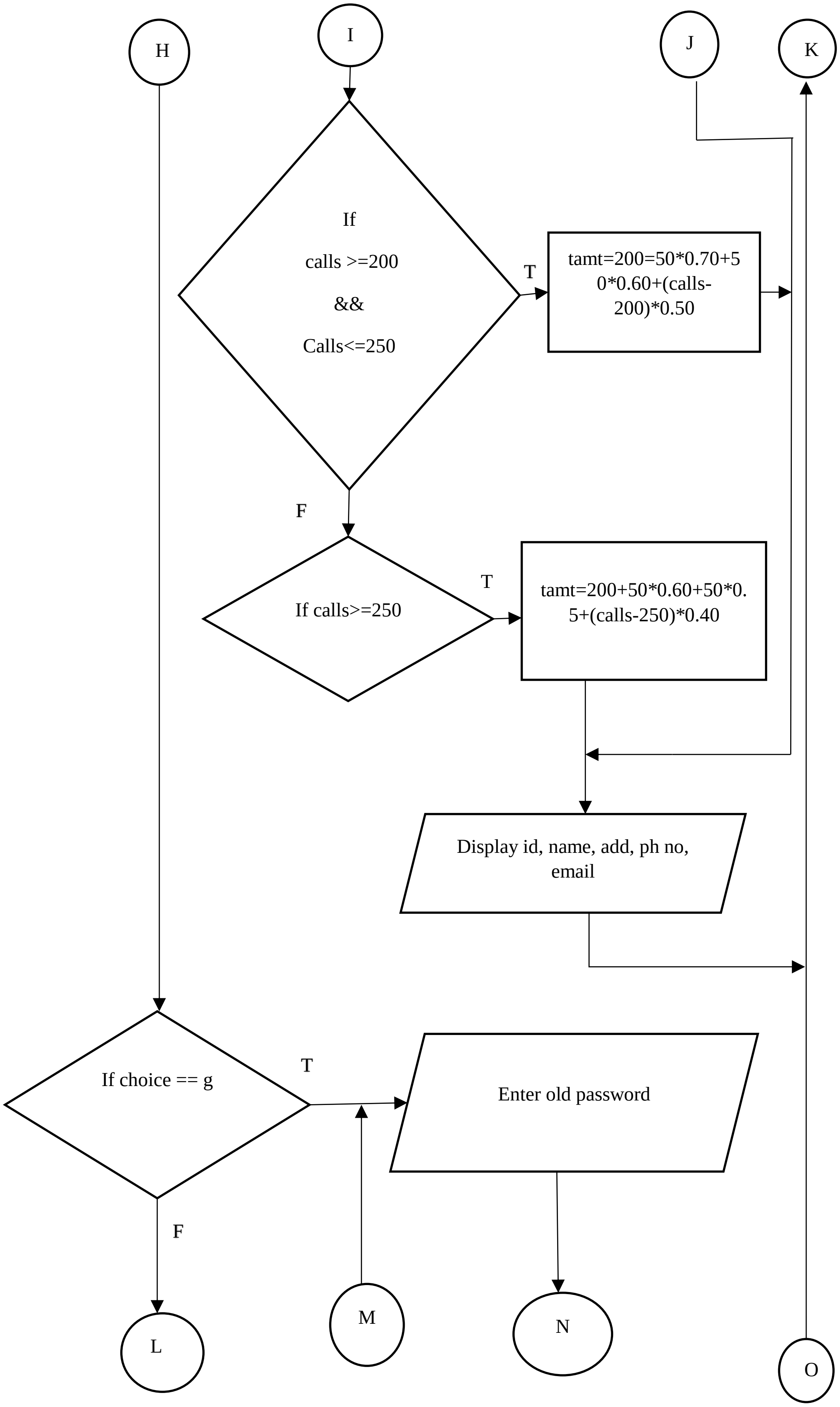
3.2 Flowchart

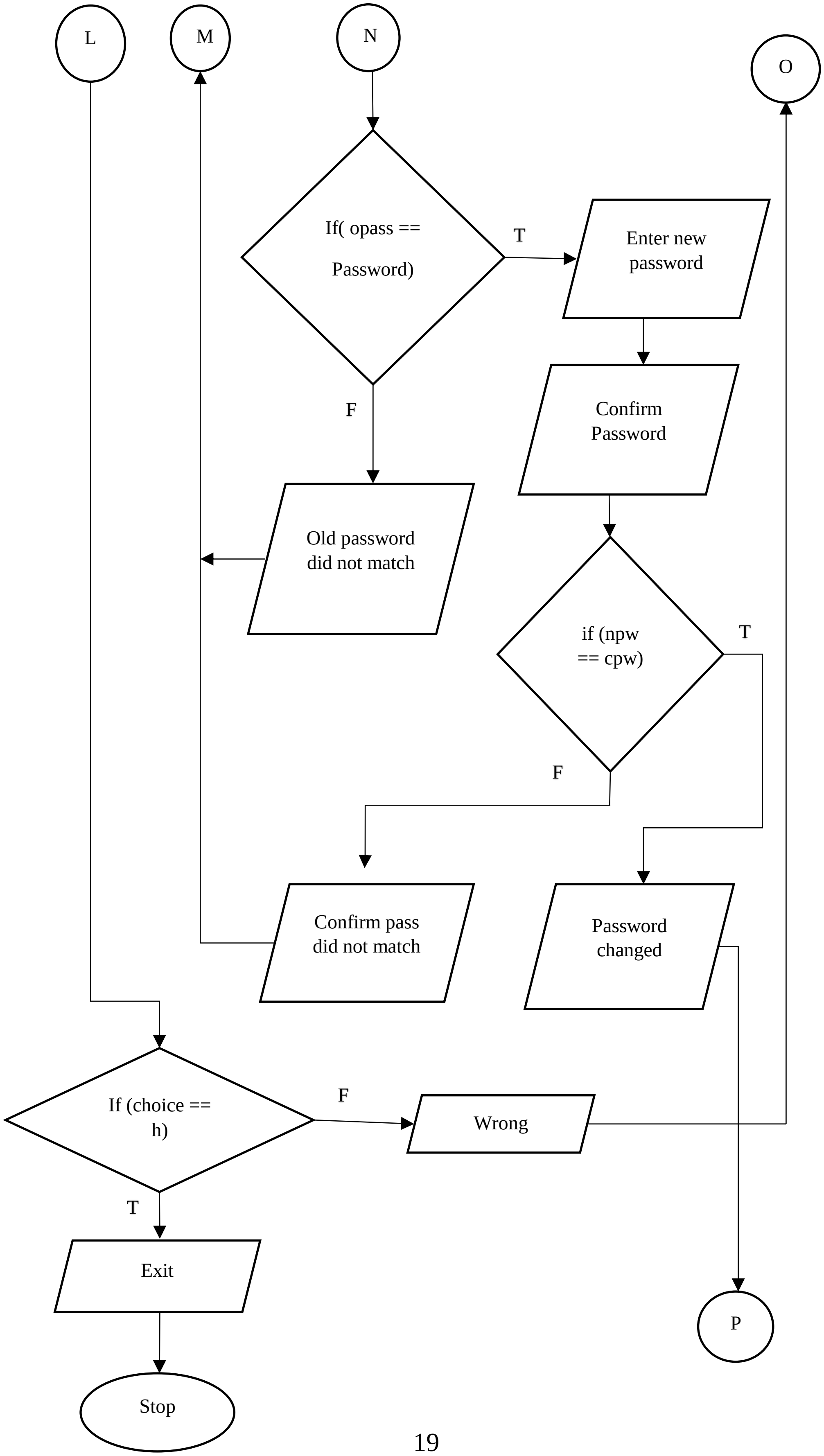



A
B
C
If (choice ==b)
T
Display customer id, name, address, ph no, email
F
If (choice == c)
T
Enter id to be modify
F
If (mid == cu.cID
T
Display old id ,name, add, ph no, email
F
Record not found
Enter new name, add, ph no, email
If (choice == d)
T
Enter id to be deleted
F
If(d.id = = cu.cID)
T
Record deleted
F
Record not found
D
E


D
E
If
(choic
e == e)
T
1. Search by name
2. Search by id
3. Go back
F
If
(choice
== 1)
T
Enter Name
F
If (sname
==cu.cna
me)
T
Display id,
name, add,
ph no, email
F
Record not found
If (choice
== 2)
T
Enter id
F
If(sid ==
cu.cID)
T
Display id,
name, add,
ph no, email
F
Record not found
If(choice
== 3
T
F
Wrong Choice
F
G


F
G
If (choice == f)
T
Enter id to be billed
F
If(bid == cu.cID)
F
T
If (calls = =100)
T
tamt =200
F
If (calls = 100&&<=150)
T
tamt=200+(calls-100)*0.70
F
If (calls>=150& & Calls<=200)
T
tamt=200+50*0.70+ (Calls-150)*0.60
J
F
H
I
K


H
I
J
K
If
calls >=200
&&
Calls<=250
T
tamt=200=50*0.70+50*0.60+(calls-200)*0.50
F
If calls>=250
T
tamt=200+50*0.60+50*0.5+(calls-250)*0.40
Display id, name, add, ph no, email
If choice == g
T
Enter old password
F
L
M
N
O


L
M
N
O
If(opass ==
Password)
T
Enter new
password
F
Confirm
Password
Old password
did not match
if (npw
== cpw)
T
F
Confirm pass
did not match
Password
changed
If (choice ==
h)
F
Wrong
T
Exit
P
Stop

4. SYSTEM DEVELOPMENT

This most important step of program development. In this step designed system is started to built in the real image of the project by writing code. So, it is main focus for us. It is the longest phase of the software development. We have properly written the code dividing into different module and implement these module in the correct way with as much less errors as possible. After completing every module we combine them to form a complete program. The program thus prepared is the backbone of the our system.

Coding:

```
//list of header file


#include<stdio.h>
#include<conio.h>
#include<string.h>
#include<stdlib.h>
#include<windows.h>
#include<process.h>
#include<time.h>        //for time function


//list of functions
void welcome(void);                 //welcome function
void login(void);                   //sub menu funcation
void menu();
void password(void);                //password fucation
void time(void);                    //time function
void insert_customer_rec();         //insert records function
void view_customer_rec();           //view records function
void search_customer_rec();         //search records menu function
void search_customer_by_id();         //searech records by customer ID function
void search_customer_by_name();    //search records by name function
void modify_customer_rec();    //modify customer record function
void delete_customer_rec();              //delete customer record function
void prepare_customer_bill();   //prepare customer bill
void set_password();                        //change password function
void exit();                                                  //exit function

//for textcolor
#define COLOR_RED     "\x1b[31m" //set red color
#define COLOR_GREEN   "\x1b[32m" //set green color
#define COLOR_RESET   "\x1b[0m"//color reset
#define COLOR_PURPLE  "\x1b[36m"
#define COLOR_MAGENTA "\x1b[35m"

//list of global variable
FILE *fp,*fptr;

//structure for password
struct password
{
        char pass[30];
};
struct password pw;
//structure for customers details
struct customer
{
int cID;
```

```
char cName[20];
char add[30];
long int phNo;
char email[30];
long int total_calls;
long int total_amount;
};
struct customer cu;

COORD coord = {0, 0};
COORD max_res,cursor_size;

void gotoxy (int x, int y) //for space of row and coloum
{
        coord.X = x; coord.Y = y;
        SetConsoleCursorPosition(GetStdHandle(STD_OUTPUT_HANDLE),
coord);
}
void delay(unsigned int mseconds) //for late printing
{
    clock_t goal = mseconds + clock();
    while (goal > clock());
}

main() //main function
{
            welcome();
            password();
            menu();
```

```
}

////////////////////////////////////////////////////////////////////////////////////////

void login(void)  //main function
{
        password();
};
////////////////////////////////////////////////////////////////////////////////////////

void welcome()
{
                printf(" \n \n \n \n \n \n \n ");
        printf(" \t  \t  \t * * * * * * * * * * * * * * * * * * * * * * * * * * \t \n");
        printf(" \t  \t       *  \t \t \t \t \t \t    * \n");
        printf(" \t  \t       * \t \t * -> WELCOME TO <- * \t \t    *  \n");
        printf(" \t  \t       * \t \t \t \t \t \t    * \n");
        printf(" \t  \t       * \t \t      TELEPHONE \t \t    *  \n");
        printf(" \t  \t       * \t \t \t \t \t \t    *  \n");
        printf(" \t  \t       * \t \t   BILLING SYSTEM \t \t    *  \n");
        printf(" \t  \t       *  \t \t \t \t \t \t    *  \n");
        printf("\t\t\t * * * * * * * * * * * * * * * * * * * * * * * * * *  \t \n");
        printf("\n");
        printf(" \n \t \t \t  Press Any Key To Continue... ");
        getch();
}
void menu()
```

```
{
        system("cls");
        int choice;
        label:
        gotoxy(30,6);
        printf("=============Administration Menu================");
        gotoxy(30,9);
        printf("Enter a : to Insert New Customer Record");
        gotoxy(30,10);
        printf("Enter b : to Display All Customer Record");
        gotoxy(30,11);
        printf("Enter c : to Modify Customer Record");
        gotoxy(30,12);
        printf("Enter d : to Delete Customer Record");
        gotoxy(30,13);
        printf("Enter e : to Search Customer Record");
        gotoxy(30,14);
        printf("Enter f : to Prepare bill");
        gotoxy(30,15);
        printf("Enter g : Change Administration Password");
        gotoxy(30,16);
        printf("Enter h : to Exit program");
        gotoxy(30,17);
        printf("====================================================
=");
        gotoxy(30,18);
        printf(">Your Choice[a-h]");
        gotoxy(30,29);
        printf("=================================================
=");
        gotoxy(30,20);
        time();
        gotoxy(30,21);
        printf("=================================================
=");
        gotoxy(30,23);
        printf(">Enter your choice: ");


switch(getch())
        {
        case 'a':
        insert_customer_rec();
        break;
        case 'b':
        view_customer_rec();
        break;
        case 'c':
        modify_customer_rec();
        break;
        case 'd':
        delete_customer_rec();
        break;
        break;
        case 'e':
        search_customer_rec();
        break;
        case 'f':
```

```
        prepare_customer_bill();
        break;
        case 'g':
        set_password();
        break;
        case 'h':
        exit();
        default:

        printf(COLOR_RED"Wrong choice"COLOR_RESET);
        getch();
        system("cls");
}
goto label;
}
/////////////////////////////////////////////////////////////////////////////////////////

void time(void) //time funcation
{
  time_t rawtime;
  struct tm * timeinfo;

  time ( &rawtime );
  timeinfo = localtime ( &rawtime );
  printf ( "Date and Time: %s", asctime (timeinfo) );
}

/////////////////////////////////////////////////////////////////////////////////////////
```

```
void password(void) //password function
{
        system("cls");
        fp=fopen("password.dat","r");
        char ch,opass[30],i;
        gotoxy(30,6);
        printf("====================Administration
Login================");
        gotoxy(30,9);
        printf(COLOR_MAGENTA "Password Required" COLOR_RESET);
        gotoxy(30,10);
        printf("Enter Password : ");
         while(ch!=13)
  {
        ch=getch();

        if(ch!=13 && ch!=8){
        putch('*');
        opass[i] = ch;
        i++;
        }
  }
  opass[i] = '\0';

        fread(&pw,sizeof(pw),1,fp);
        if(strcmp(pw.pass,opass)==0)
        {
        gotoxy(33,13);
        printf(COLOR_GREEN "Password Matched\n" COLOR_RESET);
        gotoxy(33,14);
   printf("Press any key to countinue.....");

   gotoxy(30,16);

        printf("==============================================================");
        getch();
        }
  else
  {
        system("cls");
        gotoxy(30,6);
        printf("====================Administration Login================");
        gotoxy(33,9);
        printf(COLOR_RED " Warning!! Incorrect Password" COLOR_RESET);
        getch();
        password();
}
}

/////////////////////////////////////////////////////////////////////////////////////////

void insert_customer_rec() //insert record function
{
        fp=fopen("customers.dat","a+");
        int c;
```

```
system("cls");
gotoxy(30,6);
printf("============Administration Menu===============");
gotoxy(30,7);
printf("Enter Customer Details");
gotoxy(30,9);
printf("Enter customer ID : ");
scanf("%d",&cu.cID);
gotoxy(30,10);
printf("Enter Customer Name: ");
scanf("%s",cu.cName);
gotoxy(30,11);
printf("Enter Customer Address: ");
scanf("%s",cu.add);
gotoxy(30,12);
printf("Enter Customer Phone No: ");
scanf("%ld",&cu.phNo);
gotoxy(30,13);
printf("Enter Customer Email: ");
scanf("%s",cu.email);
fwrite(&cu,sizeof(cu),1,fp);
system("cls");
gotoxy(30,6);
printf("============Administration Menu===============");
gotoxy(30,8);
printf(COLOR_GREEN "customers record added successfully" COLOR_RESET );
fclose(fp);
getch();
```

```
        view_customer_rec();
}

//////////////////////////////////////////////////////////////////////////////////////////

void modify_customer_rec() //modify record function
{
        fp=fopen("customers.dat","r+");
        int mid,flag=0,size;
        system("cls");
        size=sizeof(cu);
        gotoxy(30,6);
        printf("=============Administration Menu================");
        gotoxy(30,9);
        printf("Enter id to modify record: ");
        scanf("%d",&mid);
        while(fread(&cu,sizeof(cu),1,fp)==1)
        {
                if(mid==cu.cID)
                {
                        flag=1;
                        gotoxy(30,10);
                        printf(COLOR_GREEN"Record
Matched"COLOR_RESET);
                        gotoxy(30,12);
                        printf("ID: %d",cu.cID);
                        gotoxy(30,13);
                        printf("Name: %s",cu.cName);
                        gotoxy(30,14);
```

```
            printf("Address: %s",cu.add);
            gotoxy(30,15);
            printf("Phone No: %ld",cu.phNo);
            gotoxy(30,16);
            printf("Email: %s",cu.email);
            gotoxy(30,17);
            printf("Enter Data To Modify Record");
            gotoxy(30,19);
            printf("Name: ");
            scanf("%s",cu.cName);
            gotoxy(30,20);
            printf("Address: ");
            scanf("%s",cu.add);
            gotoxy(30,21);
            printf("Phone No: ");
            scanf("%ld",&cu.phNo);
            gotoxy(30,22);
            printf("Email: ");
            scanf("%s",cu.email);
            gotoxy(30,24);
            printf(COLOR_GREEN"Record Modified"
COLOR_RESET);
            break;
        }
}
            fseek(fp,-size,1);
            fwrite(&cu,sizeof(cu),1,fp);
            fclose(fp);
            getch();
            system("cls");

    if(flag==0)
    {
        gotoxy(30,12);
        printf(COLOR_RED"Record not found"COLOR_RESET);
        getch();
    }

}

//////////////////////////////////////////////////////////////////////////////////////////

void view_customer_rec() //view record function
{
    fp=fopen("customers.dat","r");
    system("cls");
    int flag=0;
    gotoxy(30,6);
    printf("=============Administration Menu================");
    gotoxy(30,9);
    printf("Displaying Customers Details");
    gotoxy(30,10);
    printf("ID\tName\tAddress\t\tPhone\t\tEmail\n");
    while(fread(&cu,sizeof(cu),1,fp)==1)
    {
        flag=1;
    printf("\t\t\t\t%d\t%s\t%s\t%ld\t%s\n",cu.cID,cu.cName,cu.add,cu.phNo,cu.email);
}
if(flag==0)
{
    system("cls");
    gotoxy(30,6);
    printf("=============Administration Menu================");
    gotoxy(30,9);
    printf(COLOR_RED"Record Not Found\n"COLOR_RESET);
}
    fclose(fp);
    getch();
    system("cls");
}

/////////////////////////////////////////////////////////////////////////////////////////


void search_customer_rec() //search record menu function
{
    again:
    fp=fopen("customers.dat","r");
    char sname[10];
    int flag=0,choice;
    system("cls");
    gotoxy(30,6);
    printf("=============Administration Menu================");
    gotoxy(30,9);
```

```
    printf("Enter 1: to Search Customer By Name");
    gotoxy(30,10);
    printf("Enter 2: to Search Customer By ID");
    gotoxy(30,11);
    printf("Enter 3: to go back");
    gotoxy(30,12);
    printf("Enter Your Choice: ");
    switch(getch())
    {
        case '1':
            search_customer_by_name();
        break;
        case '2':
            search_customer_by_id();
        break;
        case '3':
            menu();
        break;

        default:
            printf(COLOR_RED"Wrong choice"COLOR_RESET);
            getch();
            system("cls");
}
goto again;
}

//////////////////////////////////////////////////////////////////////////////////////////
```

```
void search_customer_by_name() //search record by name function
{
        fp=fopen("customers.dat","r");
        char sname[10];
        int flag=0;
        system("cls");
        gotoxy(30,6);
        printf("============Administration Menu===============");
        gotoxy(30,9);
        printf("Search Customer By Name");
        gotoxy(30,10);

        printf("Enter name : ");
        scanf("%s",sname);
        while(fread(&cu,sizeof(cu),1,fp)==1)
        {
        if(strcmp(cu.cName,sname)==0)
        {
        flag=1;
        system("cls");
        gotoxy(30,6);
        printf("============Administration Menu===============");
        gotoxy(30,9);
        printf(COLOR_GREEN"Record Found"COLOR_RESET);
        gotoxy(30,10);
        printf("Id : %d\n",cu.cID);
        gotoxy(30,11);
        printf("Name : %s\n",cu.cName);
        gotoxy(30,12);
        printf("Address : %s\n",cu.add);
        gotoxy(30,13);
        printf("Phone No : %ld\n",cu.phNo);
        gotoxy(30,14);
        printf("Email : %s\n",cu.email);
        getch();
        system("cls");
        menu();
}
}
if(flag==0)
{
        system("cls");
        gotoxy(30,6);
        printf("=============Administration Menu================");
        gotoxy(30,9);
        printf(COLOR_RED"Record not found"COLOR_RESET);
        getch();
        menu();
}
fclose(fp);
}

//////////////////////////////////////////////////////////////////////////////////////////

void search_customer_by_id() //search record by ID function
```

```
{
        fp=fopen("customers.dat","r");
        int cid;
        int flag=0;
        system("cls");
        gotoxy(30,6);
        printf("=============Administration Menu================");
        gotoxy(30,9);
        printf("Search Customer By ID");
        gotoxy(30,10);

        printf("Enter ID : ");
        scanf("%d",&cid);
        while(fread(&cu,sizeof(cu),1,fp)==1)
        {
        if(cu.cID==cid)
        {
        flag=1;
        system("cls");
        gotoxy(30,6);
        printf("=============Administration Menu================");
        gotoxy(30,9);
        printf(COLOR_GREEN "Record Found" COLOR_RESET);
        gotoxy(30,10);
        printf("Id : %d\n",cu.cID);
        gotoxy(30,11);
        printf("Name : %s\n",cu.cName);
        gotoxy(30,12);
        printf("Address : %s\n",cu.add);
        gotoxy(30,13);
        printf("Phone No : %ld\n",cu.phNo);
        gotoxy(30,14);
        printf("Email : %s\n",cu.email);
        getch();
        system("cls");
        menu();
    }
    }
    if(flag==0)
    {
        system("cls");
        gotoxy(30,6);
        printf("=============Administration Menu================");
        gotoxy(30,9);
        printf(  COLOR_RED "Record not found" COLOR_RESET);
        getch();
        menu();
    }
    fclose(fp);
    }

    //////////////////////////////////////////////////////////////////////////////////////////

    void delete_customer_rec() //delete record function
    {
        system("cls");
```

```
fp=fopen("customers.dat","r+");
int did,flag=0;
fptr=fopen("temp.dat","w+");
gotoxy(30,6);
printf("=============Administration Menu================");
gotoxy(30,9);
printf("Delete Record By Customer ID");
gotoxy(30,10);
printf("Enter Customer ID: ");
scanf("%d",&did);
while(fread(&cu,sizeof(cu),1,fp)==1)
{
        if(did==cu.cID)
{
flag=1;
}
if(did!=cu.cID)
{
fwrite(&cu,sizeof(cu),1,fptr);
}
}
if(flag==1)
{
system("cls");
gotoxy(30,6);
printf("=============Administration Menu================");
gotoxy(30,9);
printf(COLOR_GREEN"Customer Record Deleted Successfully" COLOR_RESET);
```

```
	getch();
	}
	else
	{
	system("cls");
	gotoxy(30,6);
	printf("=============Administration Menu================");
	gotoxy(30,9);
	printf(  COLOR_RED "Record not found" COLOR_RESET);
	getch();
	}
	fclose(fp);
	fclose(fptr);
	remove("customers.dat");
	rename("temp.dat","customers.dat");
}

//////////////////////////////////////////////////////////////////////////////////////////

void prepare_customer_bill()
{
	system("cls");
	fp=fopen("customers.dat","r");
	int bid,flag=0,size;
	int taxinvoiceNo;
	int calls;
	float tamt;
	char ptype[30];
```

```
system("cls");
size=sizeof(cu);
gotoxy(30,6);
printf("=============Administration Menu================");
gotoxy(30,9);
printf("Enter Customer id to prepare bill: ");
scanf("%d",&bid);
while(fread(&cu,sizeof(cu),1,fp)==1)
{
    if(bid==cu.cID)
    {
        flag=1;
        gotoxy(30,10);
        printf(COLOR_GREEN"Record Match"COLOR_RESET);
        gotoxy(30,12);
        printf("Enter Data To prepare bill");
        gotoxy(30,13);
        printf("Tax Invoice No: ");
        scanf("%d",&taxinvoiceNo);
        gotoxy(30,14);
        printf("Total Calls: ");
        scanf("%d",&calls);
        gotoxy(30,15);
        printf("Payment Type(cheque or cash): ");
        scanf("%s",ptype);
        if(calls<=100)
            tamt=200;
        else if(calls>100 && calls<150)
```

```
                    tamt=200+(calls-100)*0.70;
                else if(calls>=150 && calls<=200)
                    tamt=200+50*0.70+(calls-150)*0.60;
                else if(calls>=200 && calls<=250)
                  tamt=200+50*0.70+50*0.60+(calls-200)*0.50;
            else
              tamt=200+50*0.70+50*0.60+50*0.50+(calls-250)*0.40;


                system("cls");
                gotoxy(30,6);
                printf("==================Administration
Menu=====================");
                gotoxy(30,9);
                printf(COLOR_GREEN"bill prepared
successfully"COLOR_RESET);
                gotoxy(50,11);
                printf(COLOR_PURPLE"TELEPHONE BILLING
SYSTEM"COLOR_RESET);
                gotoxy(30,13);
                printf("Tax InvoiceNo: %d",taxinvoiceNo);
                gotoxy(30,14);
                printf("Customer ID: %d",cu.cID);
                gotoxy(60,13);
                time();
                gotoxy(30,15);
                printf("Customer Name: %s",cu.cName);
                gotoxy(30,16);
                printf("Customer Address: %s",cu.add);
                gotoxy(30,17);
                printf("Customer Phone: %ld",cu.phNo);
                gotoxy(30,18);
                printf("Customer Email: %d",cu.email);
                gotoxy(30,19);
                printf("Payment Mode: %s",ptype);
                gotoxy(30,20);
                printf("Totals Calls:\t\t\t\t\t\t    %d",calls);
                gotoxy(30,21);
                printf("-------------------------------------------------------------
------");
                gotoxy(30,22);
                printf("Total Amount(tax included):\t\t\t\t    %f",tamt);
                gotoxy(30,24);
                printf("Received %f in %s Thanks",tamt,ptype);
                getch();
                system("cls");
            }
}

    if(flag==0)
    {

            gotoxy(30,12);
            printf(COLOR_RED"Record not found"COLOR_RESET);
            getch();
    }
    fclose(fp);
}
```

```
//////////////////////////////////////////////////////////////////////////////////////

void set_password() //change password
{
            system("cls");
            char opass[20],npw[20],cpw[20];
            fp=fopen("password.dat","r+");
            gotoxy(30,6);
            printf("============Administration Menu===============");
            gotoxy(30,9);
            printf("Set New Password");
            gotoxy(30,10);
            printf("Enter Old Password : ");
            scanf("%s",opass);
            fread(&pw,sizeof(pw),1,fp);
            if(strcmp(pw.pass,opass)==0)
            {
                        rewind(fp);
                        gotoxy(30,13);
                        printf("Enter New Password : ");
                        scanf("%s",pw.pass);
                        gotoxy(30,14);
                        printf("Enter Conform Password : ");
                        scanf("%s",cpw);
                        fclose(fp);
                        if(strcmp(pw.pass,cpw)==0)
                        {           fp=fopen("password.dat","w");
                                    fwrite(&pw,sizeof(pw),1,fp);
```

```
                    gotoxy(30,15);
                    printf(COLOR_GREEN "change password successfully"
COLOR_RESET );


                    fclose(fp);
                    getch();
                    login();
            }
            else
            {
                    gotoxy(30,14);
                    printf(COLOR_RED "Confirm password didnot match"
COLOR_RESET );
                    getch();
                    set_password();
                    }
}else {
                    gotoxy(30,14);
                    printf(COLOR_RED "old password didnot match"
COLOR_RESET );
                    getch();
                    set_password();
}


}

//////////////////////////////////////////////////////////////////////////////////////////


void exit(void) //exit function
```

```
{
        system("cls");

                gotoxy(30,9);
                delay(600);
                printf("Developed By: ");
                gotoxy(30,11);
                delay(600);
                printf( COLOR_PURPLE "1. Debid Rana Magar");
                gotoxy(30,12);
                delay(600);
                printf("2. Dansi Ram Acharya");
                gotoxy(30,13);
                delay(600);
                printf("3. Khusi Ram Mahato");
                gotoxy(30,14);
                delay(600);
                printf("4. Bishal Dhital");
                gotoxy(30,16);
                delay(600);
                printf(COLOR_GREEN"*****Happy                Coding*****"
COLOR_RESET);
                printf("\nExiting in 5 second...........");
                delay(600);
                printf("\nExiting in 4 second..........");
                delay(600);
                printf("\nExiting in 3 second.........");
                delay(600);
                printf("\nExiting in 2 second........");
```

```
          delay(600);
          printf("\nExiting in 1 second.......");
          exit(0);
}
```

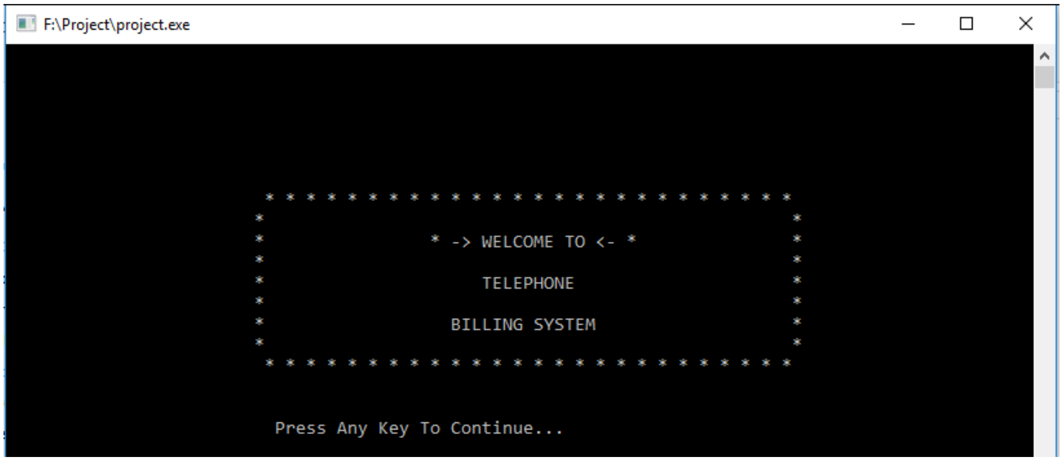
5. SYSTEM TESTING

The System testing is the process of improving the system debugging the problems from the system. The purpose of testing is to identifying and correct errors in the system.

After the code is developed we tested against the requirement to make sure that the product is actually solving the need addressed and gathered during the requirement phase. So we performed various testing and debugging method to minimizing the bugs and errors.

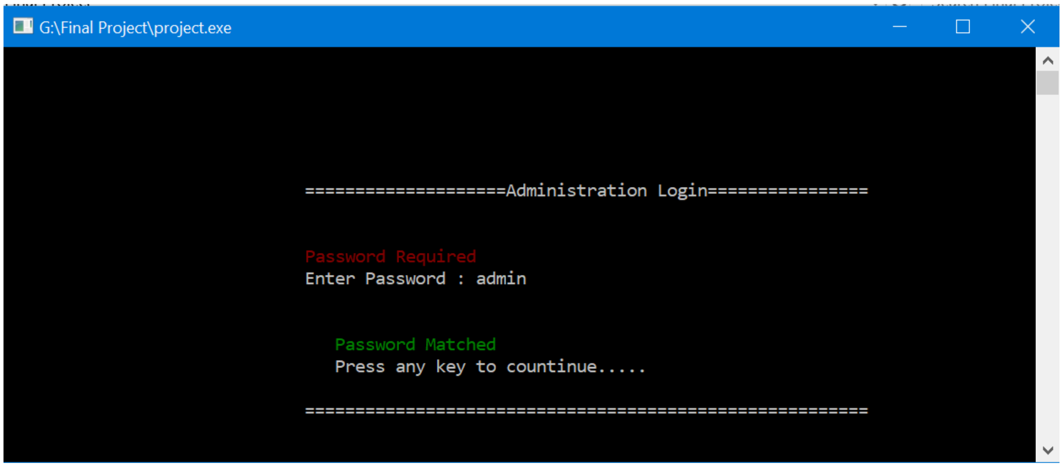
Our term tested the whole data very precisely for number of times and came to final result without errors.

When this program starts its interface is like this……..



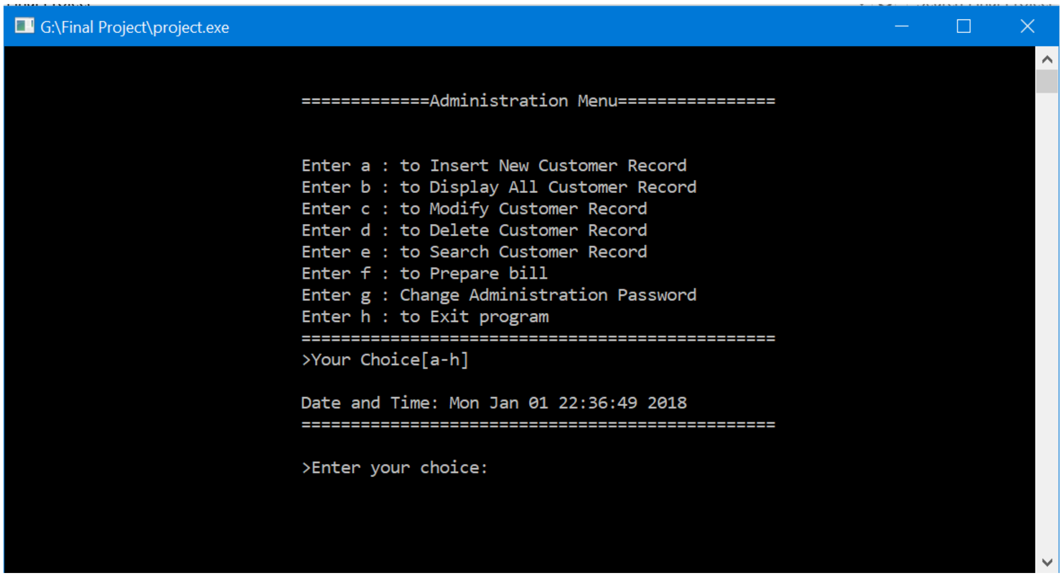
```
 * * * * * * * * * * * * * * * * * * * * * * * * * * *
*                                                       *
*                  * -> WELCOME TO <- *                 *
*                                                       *
*                       TELEPHONE                       *
*                                                       *
*                    BILLING SYSTEM                     *
*                                                       *
 * * * * * * * * * * * * * * * * * * * * * * * * * * *


  Press Any Key To Continue...
```

Fig 5.1 System Welcome Menu



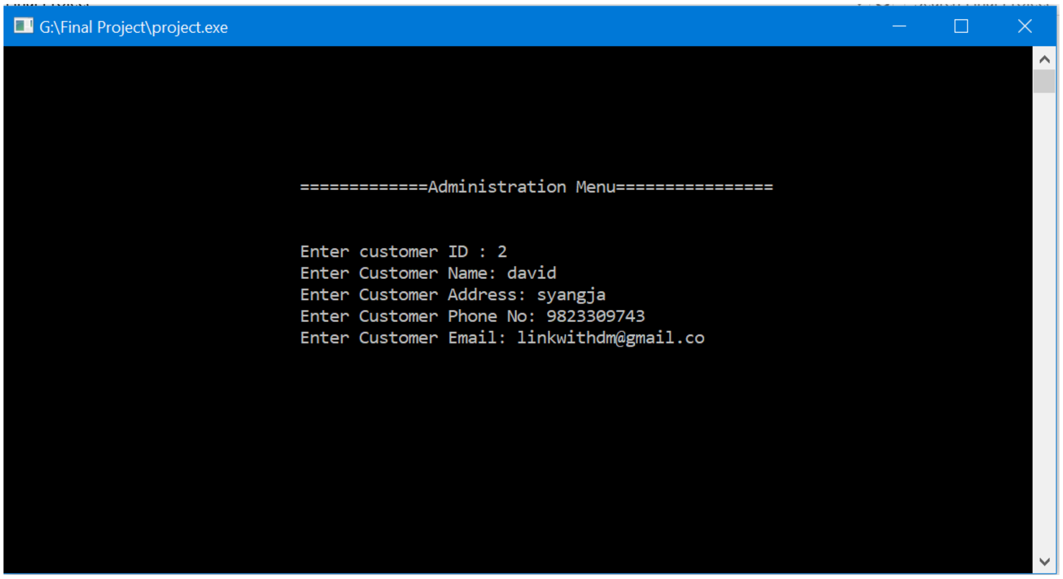
```
===================Administration Login================


Password Required
Enter Password : admin


   Password Matched
   Press any key to countinue.....

======================================================
```

Fig 5.2 Administration Login



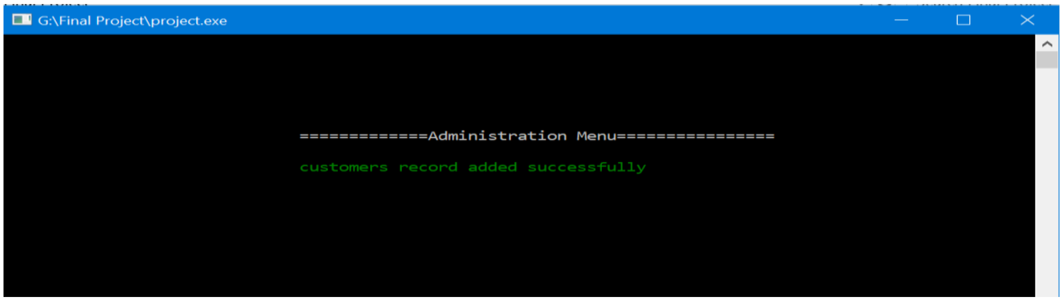
```
=============Administration Menu================


Enter a : to Insert New Customer Record
Enter b : to Display All Customer Record
Enter c : to Modify Customer Record
Enter d : to Delete Customer Record
Enter e : to Search Customer Record
Enter f : to Prepare bill
Enter g : Change Administration Password
Enter h : to Exit program
================================================
>Your Choice[a-h]

Date and Time: Mon Jan 01 22:36:49 2018
================================================

>Enter your choice:
```

Fig 5.3 Administration Menu



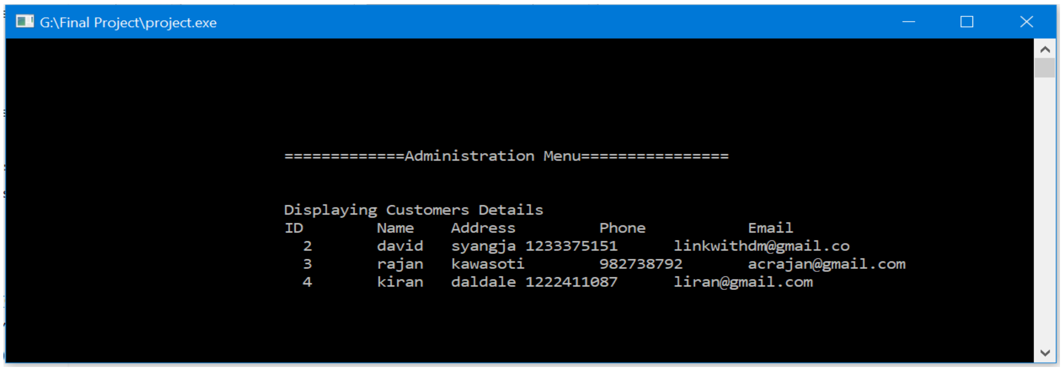
```
=============Administration Menu================


Enter customer ID : 2
Enter Customer Name: david
Enter Customer Address: syangja
Enter Customer Phone No: 9823309743
Enter Customer Email: linkwithdm@gmail.co
```

Fig 5.4 Insert Customer Record



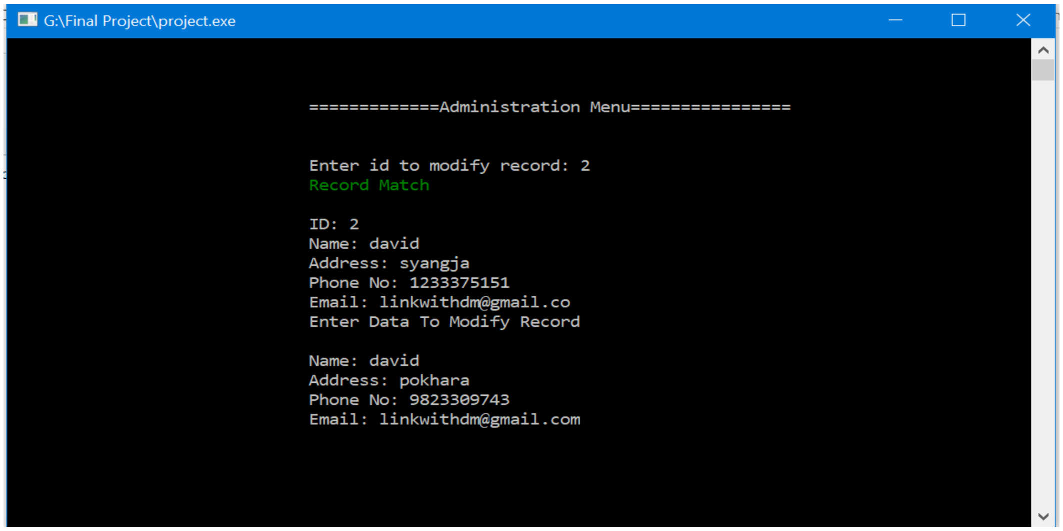
```
=============Administration Menu================
customers record added successfully
```

Fig 5.5 Record Inserting Added Successfully



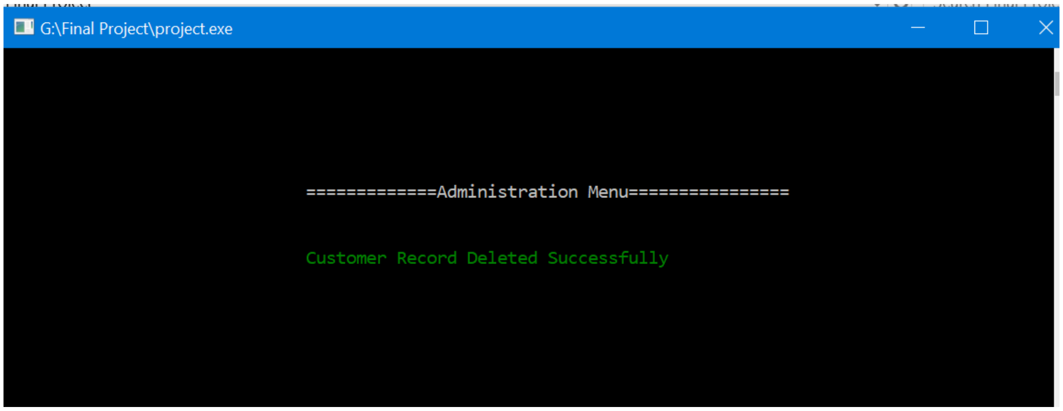
```
=============Administration Menu================


Displaying Customers Details
ID         Name    Address         Phone           Email
  2        david   syangja 1233375151      linkwithdm@gmail.co
  3        rajan   kawasoti        982738792       acrajan@gmail.com
  4        kiran   daldale 1222411087      liran@gmail.com
```

Fig 5.6 Displaying Record



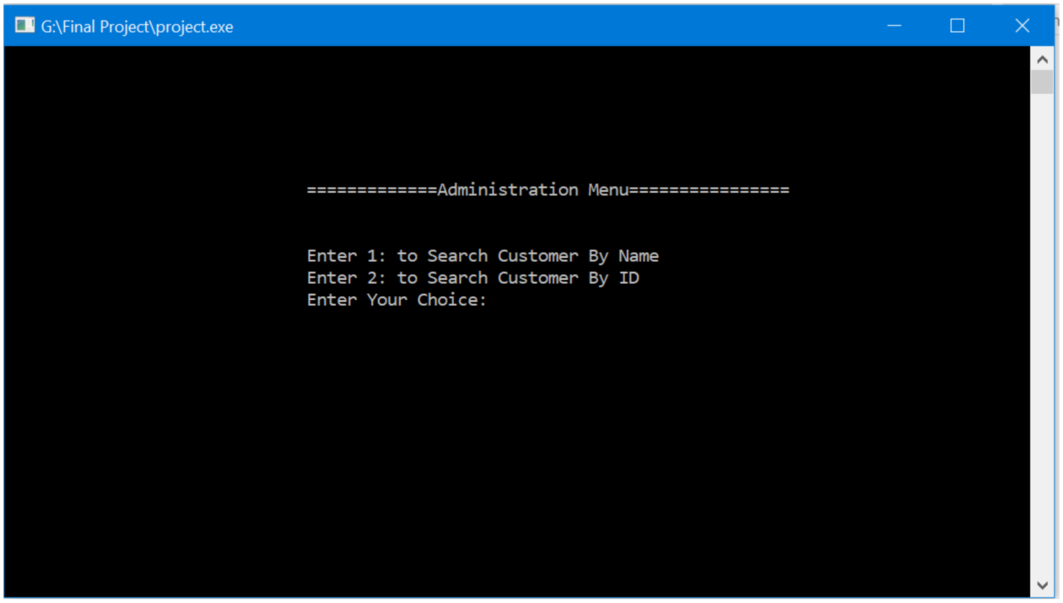
```
=============Administration Menu================


Enter id to modify record: 2
Record Match

ID: 2
Name: david
Address: syangja
Phone No: 1233375151
Email: linkwithdm@gmail.co
Enter Data To Modify Record

Name: david
Address: pokhara
Phone No: 9823309743
Email: linkwithdm@gmail.com
```

Fig 5.7 Modifying Record



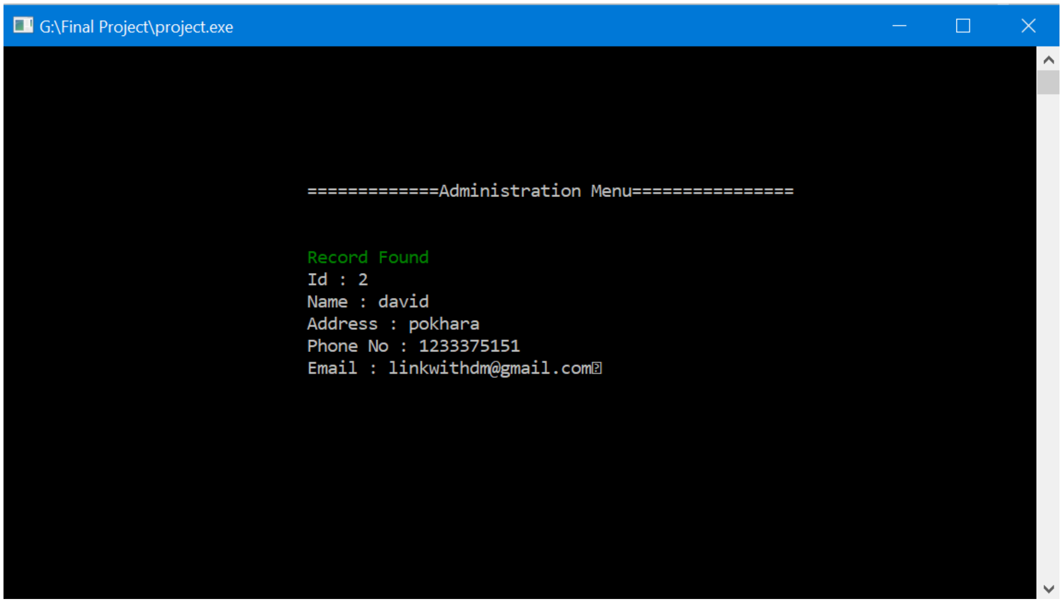
```
=============Administration Menu================


Customer Record Deleted Successfully
```

Fig 5.8 Record Deleted Successfully



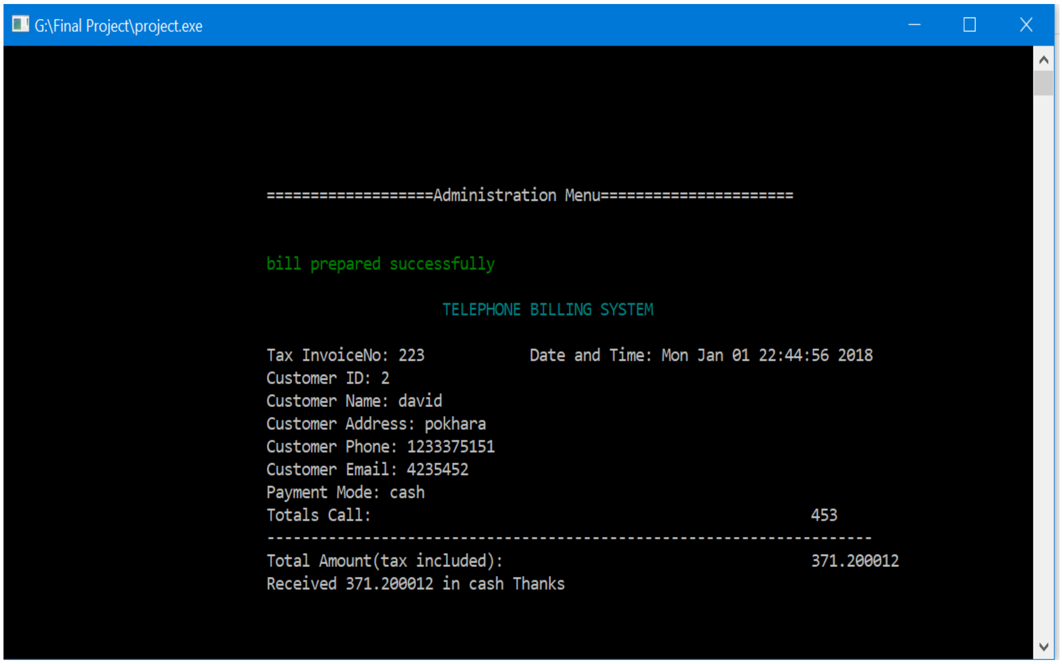
```
=============Administration Menu================


Enter 1: to Search Customer By Name
Enter 2: to Search Customer By ID
Enter Your Choice:
```

Fig 5.9 Record Search Menu



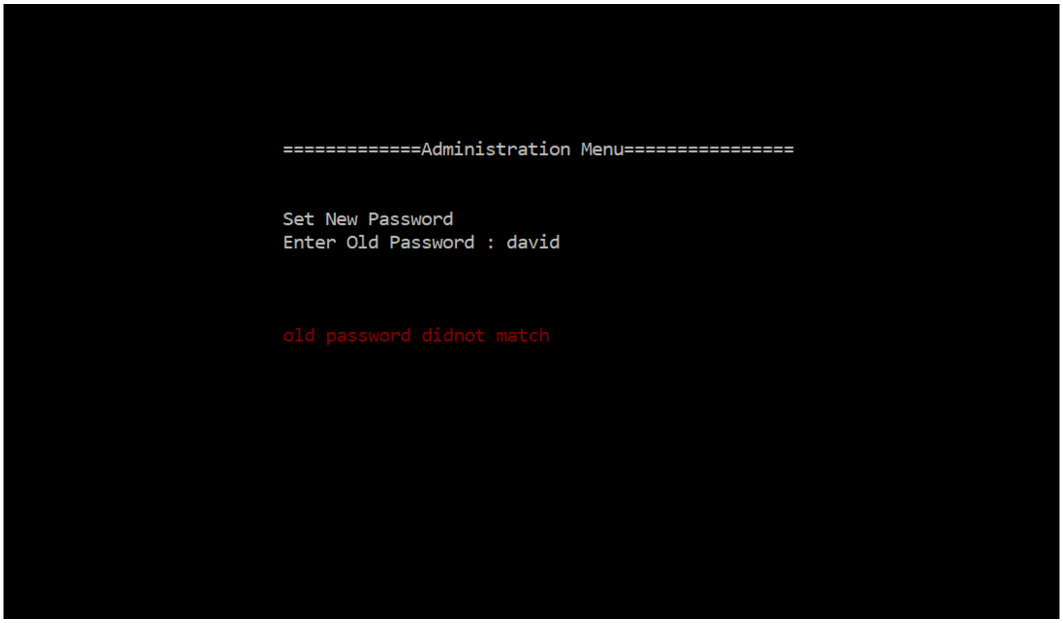
```
=============Administration Menu================


Record Found
Id : 2
Name : david
Address : pokhara
Phone No : 1233375151
Email : linkwithdm@gmail.com
```

Fig 5.10 Search Record Matched



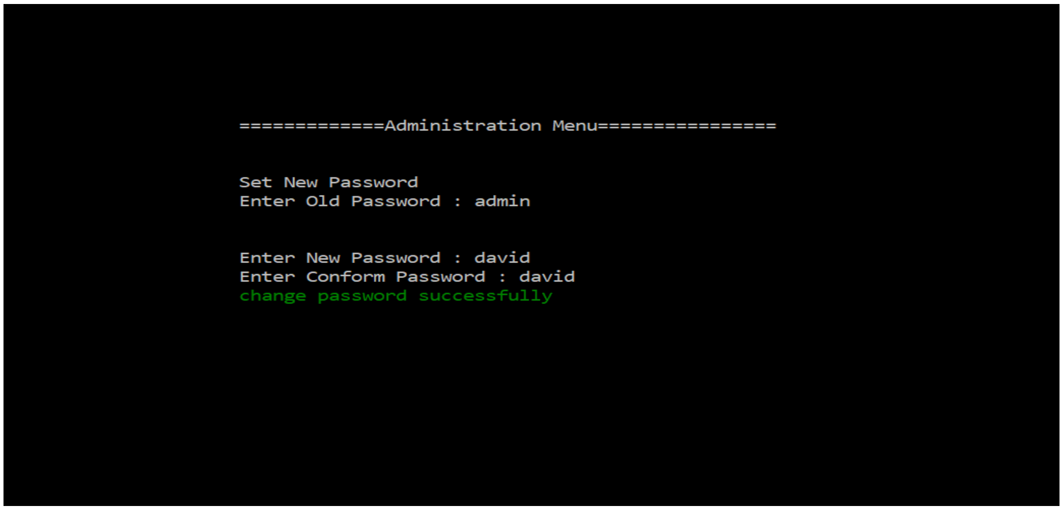
```
===================Administration Menu======================


bill prepared successfully

                    TELEPHONE BILLING SYSTEM

Tax InvoiceNo: 223           Date and Time: Mon Jan 01 22:44:56 2018
Customer ID: 2
Customer Name: david
Customer Address: pokhara
Customer Phone: 1233375151
Customer Email: 4235452
Payment Mode: cash
Totals Call:                                                  453
---------------------------------------------------------------------
Total Amount(tax included):                                   371.200012
Received 371.200012 in cash Thanks
```

Fig 5.11 Displaying Prepared Bill



```
=============Administration Menu================


Set New Password
Enter Old Password : david



old password didnot match
```

Fig 5.12 Old Password Did Not Match



```
=============Administration Menu================


Set New Password
Enter Old Password : admin


Enter New Password : david
Enter Conform Password : david
change password successfully
```

Fig 5.13 Password Change Successfully

6. SYSTEM IMPLEMENTATION

In general sense, the implementation means taking the action of the system to the real use but in converse wide range of meaning. The system or program implementation process starts from the conversion of the basic application and ends at complete computer system replacement. Upgrading system or program so seriousness conversion system is applied otherwise the problem may occur into the system. Implementation of modified application to replace the existing system by using the same computer. It is earliest implementation method.

The relationship between users and information systems specialists has traditionally been a problem area for information system implementation efforts. Users and information systems specialists trend to have different backgrounds, interests and priorities. This is referred to as the user-designer communications gap. These differences lead to divergent organizational loyalties, approaches to problem solving and vocabularies.

7. CONCLUSION

Development of this Telephone Billing System was necessary to keep the records accurately, to replace most of the paper works which has risk of data loss and occupy large space and time.

Since, the project is on CUI based environment, it does not contain any visual forms or such interface. So, the first interface screen for running the project is through the menu based where the user will be provided a set of particular menu facilities in order to select the required operation in program.

7.1. Limitation of the System

- This system is based on the stand alone programming. So, it doesn't have its own database.
- This system cannot hold much detail information about the costumer and is made for the billing purpose only.
- This system does not support any graphics.
- This System is not multiuser.
- Data loss can occur due to technical issue so it is necessary to make to backup at regular interval of time

7.2 Future Scope of the Project

- We can design the better system by using latest programming language.
- This project will be useful for the more organization of similar nature after editing some blocks of codes.
- We can convert this standalone system into the client server system in the course of action.

REFERENCES

While making this project we took help and guidance from the difference sources. We took help from our teacher, seniors and books.

We took guidance and help to complete our project report from the following sources:-

Web References:-

- https://stackoverflow.com
- https://github.com
- https://www.w3schools.in