Broadband Billing System

By

Md. Arham Tabib

ID: 221-15-5707

Sultanus Salehin
ID: 221-15-5357

Sumyia Sabrin Liza
ID: 221-15-5768

Course code : CSE 135
Course title : Data Structure Lab

Submitted To

Indrani Sen Toma
Lecturer
Department of CSE
Daffodil International University

DAFFODIL INTERNATIONAL UNIVERSITY
DHAKA, BANGLADESH
18 NOVEMBER 2023

ABSTRACT

The broadband billing system aims to develop and deliver a robust and efficient billing system for a broadband service provider. It can store the customer information such as number, pending bills. It will help the service provider to keep a note of pending payments and track records. It also helps to modify the information. It simplifies the billing process for any customer having payment option.

TABLE OF CONTENTS

CONTENTS	PAGE
Abstract	1
Table of contents	2
CHAPTER	
CHAPTER 1	**3-4**
1.1 Introduction	3
1.2 Literature review	3-4
1.3 Problem Statement	4
CHAPTER 2	**5-10**
2.1 Methodology	5
2.2 Implementation	5-9
2.3 Testing and Evaluation	9-10
CHAPTER 3	**11-14**
3.1 Results	11-14
3.2 Discussion	14
CHAPTER 4	**15**
4.1 Conclusion	15
4.2 Future Work	15
REFERENCES	**15**

CHAPTER 1

1.1 Introduction

This report will express about the build and procedure of this project. We applied data structure methods to build this project. In this modern-day world user of internet is increasing day by day. So, the number of broadband connection also increasing.

Our project will help and reduce the hassle by tracking and keeping notes of consumer payment, information, phone number.

1.2 Literature Review

To understand the context of implementing a broadband billing system in C, it is essential to review the existing literature on broadband billing systems. Research by Smith et al. (2018) provides a comprehensive overview of the challenges and requirements in developing billing systems for broadband services, emphasizing the need for real-time processing.

Security is a paramount concern in billing systems, particularly in the context of broadband services. The work of Chen and Liu (2019)

discusses security vulnerabilities and potential threats in billing systems and proposes strategies for implementing secure billing solutions in C, addressing issues such as data encryption and user authentication.

The strength of this project is it is built in C language. C is widely used mother language of programing so it is easy to understand and easy to handle.

The weakness of the project is it has some security concern. Which can be improved and it has large time complexity.

1.2 Problem Statement

In the age of information and technology everything is updating. In western countries every thing is being updated with a touch of technologies. Somehow our country billing process has not updated as such way. Most of the provider uses the analog way for his billing process and pending payment calculation. Sometimes the record got lost and keeping notes in paper is hassle because specific data cannot be search when it is needed.

Here where our projects aim to solve those problems but updating the system of tracking customer information in manner way. Keep notes about pending process also payment process. Specific data can be search when it is needed.

CHAPTER 2

2.1 Methodology

The system has various data structure algorithms. At first we used a linked list struct to secure all variables such as phone-number, name, amount. We used stack to keep the variable stored. We use a OJS file to keep records save. The OJS file works as a local database for this project.

The tools used in the projects are:

1.Code blocks: It is an IDE for c programing language.

2.Pc: A well decorated pc to do all the steps done properly.

3.OJS file: It's a file extension working as local database.

4.Programming language: We selected C as our programing language and applied data structure method as implementation.

2.2 Implementation

We implemented the linked list , stack to store information.

Raw code:

```
#include<stdio.h>
#include<string.h>
#include<ctype.h>
#include<stdlib.h>
struct subscriber
{
char phonenumber[20];
char name[50];
float amount;
}s;
void addrecords();
void listrecords();
void modifyrecords();
void searchrecords();
void payment();
char get;
```

```
int main()
{
 int phonenumber;
 int choice;
 printf("\n\n\n\n\n\n\n\n\n\n\t\t**********************************************************************");
 printf("\n\t\t\t ---WELCOME TO THE BILLING MANAGEMENT SYSTEM---");
 printf("\n\t\t**********************************************************************");
 while (choice!=6)
 {
 printf("\n Enter what you want:\n\n 1 : add new records.\n 2 : list of records");
 printf("\n 3 : modify records\n 4 : payment");
 printf("\n 5 : search records");
 printf("\n 6 : exit\n");
 scanf("%d",&choice);
 switch(choice)
 {
 case 1:
 addrecords();break;
 case 2:
 listrecords();break;
 case 3:
 modifyrecords();break;
 case 4:
 payment();break;
 case 5:
 searchrecords();break;
 case 6:
 printf("\n\n\t\t\t\tTHANK YOU FOR USING OUR SERVICE");
 exit(0);
 break;
 default:
 printf("Incorrect Input");
 printf("\nAny key to continue");
 break;
 }
 }
 }
```

Main function of the project ;

```
void addrecords()
{
FILE *f;
int test=1;
f=fopen("file.gjs","ab+");
if(f==0)
{   f=fopen("file.gjs","wb+");
printf("please wait while we configure your computer");
printf("/npress any key to continue");
}
while(test!=57)
{
printf("\n Enter phone number: ");
scanf("%s",s.phonenumber);
printf("\n Enter name: ");
scanf("%s[^\n]",s.name);
printf("\n Enter amount: ");
scanf("%f",&s.amount);
fwrite(&s,sizeof(s),1,f);
printf("1 record successfully added");
printf("\n Press 57 to exit, any other key to add other record: ");
scanf("%d",&test);
if(test==57)
break;
}
fclose(f);
}
```

Adding information function ;

```
void listrecords()
{
FILE *f;
int i;
if((f=fopen("file.dis","rb"))==NULL)
exit(0);
system("cls");
printf("Phone Number\t\tUser Name\t\t\tAmount\n");
for(i=0;i<79;i++)
printf("-");
while(fread(&s,sizeof(s),1,f)==1)
{
printf("\n%-10s\t\t%-20s\t\ttaka. %.2f /-",s.phonenumber,s.name,s.amount);
}
printf("\n");
for(i=0;i<79;i++)
printf("-");
fclose(f);
}
```

Listing records;

```
void searchrecords()
{
FILE *f;
char phonenumber[20];
int flag=1;
f=fopen("file.dis","rb+");
if(f==0)
exit(0);
fflush(stdin);
printf("Enter Phone Number to search in our database: ");
scanf("%s", phonenumber);
while(fread(&s,sizeof(s),1,f)==1)
{
if(strcmp(s.phonenumber,phonenumber)==0)
{
printf(" Record Found ");
printf("\n\nPhonenumber: %s\nName: %s\nAmount: taka.%0.2f\n",s.phonenumber,s.name,s.amount);
flag=0;
break;
}
else if(flag==1)
{
printf("Requested Phone Number Not found in our database");
}
}
fclose(f);
}
```

Search the record function;

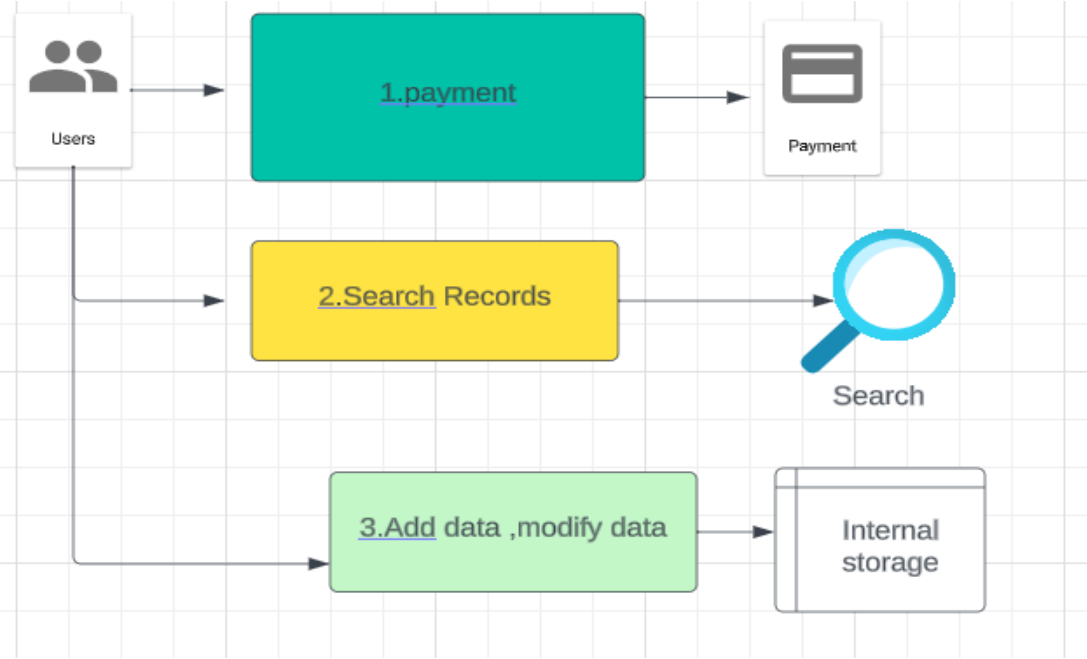
```
void modifyrecords()
{
FILE *f;
char phonenumber[20];
long int size=sizeof(s);
if((f=fopen("file.cjs","rb+"))==NULL)
exit(0);
printf("Enter phone number of the subscriber to modify:");
scanf("%s[^\n]",phonenumber);
fflush(stdin);
while(fread(&s,sizeof(s),1,f)==1)
{
if(strcmp(s.phonenumber,phonenumber)==0)
{
printf("\n Enter phone number:");
scanf("%s",s.phonenumber);
printf("\n Enter name: ");
fflush(stdin);
scanf("%s[^\n]",s.name);
printf("\n Enter amount: ");
scanf("%f",&s.amount);
fseek(f,-size,SEEK_CUR);
fwrite(&s,sizeof(s),1,f);
break;
}
}
fclose(f);
}
```

Modify records;

```
void payment()
{
FILE *f;
char phonenumber[20];
long int size=sizeof(s);
float amt;
int i;
if((f=fopen("file.cjs","rb+"))==NULL)
exit(0);
printf("Enter phone number of the subscriber for payment");
scanf("%s[^\n]",phonenumber);
fflush(stdin);
while(fread(&s,sizeof(s),1,f)==1)
{
if(strcmp(s.phonenumber,phonenumber)==0)
{
printf("\n Phone No.: %s",s.phonenumber);
printf("\n Name: %s",s.name);
printf("\n Current amount: %f",s.amount);
printf("\n");
for(i=0;i<79;i++)
printf("-");
printf("\n\nEnter amount of payment: ");
fflush(stdin);
scanf(" %f",&amt);
s.amount=s.amount-amt;
fseek(f,-size,SEEK_CUR);
fwrite(&s,sizeof(s),1,f);
break;
}
}
printf("THANK YOU %s FOR YOUR TIMELY PAYMENTS",s.name);
fclose(f);
}
```

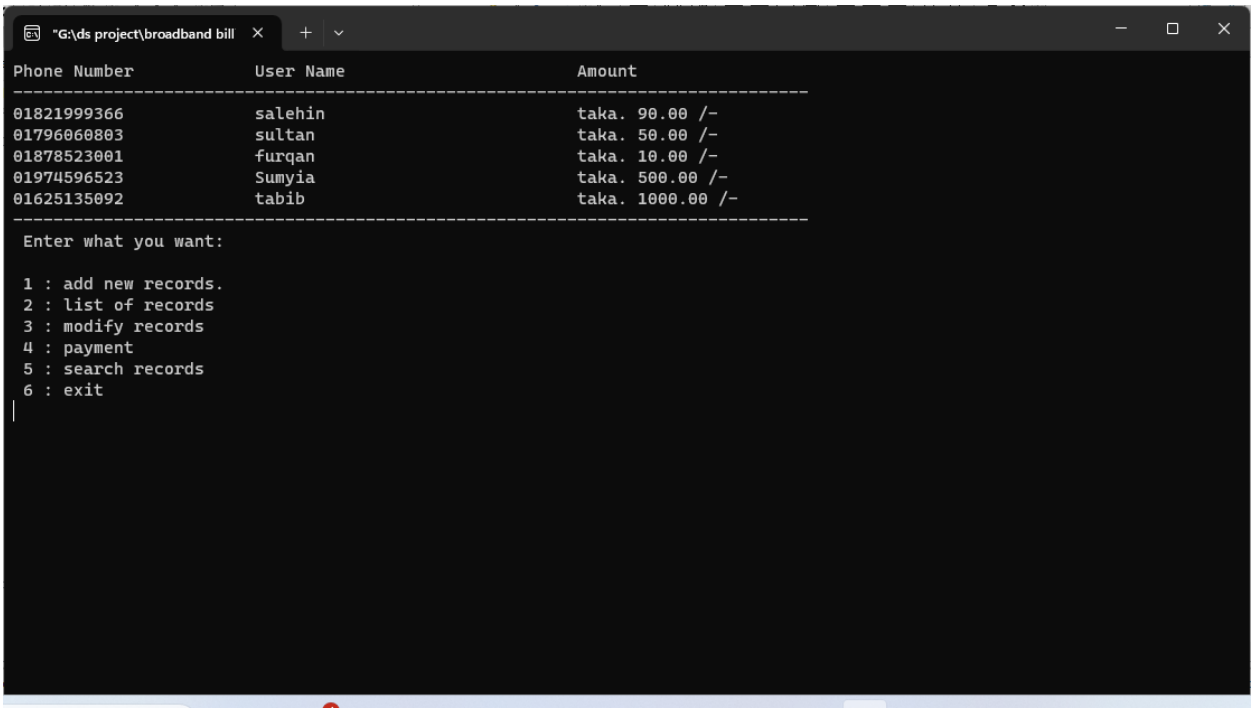
Payment function;

Diagram:

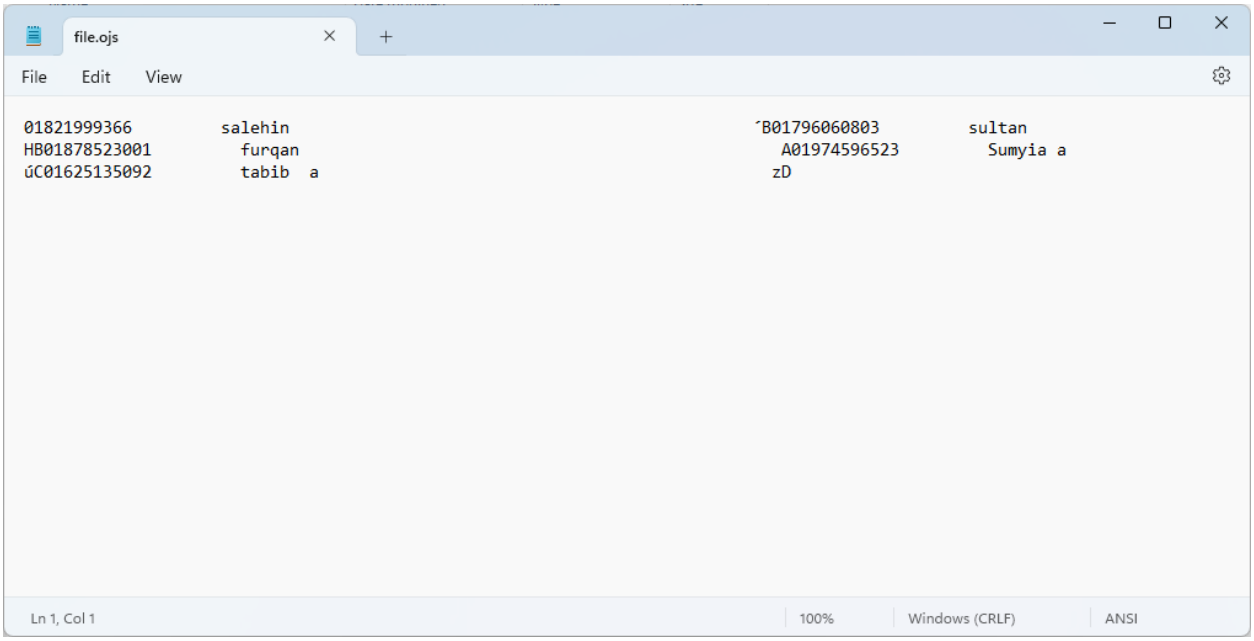


2.3 Testing and Evaluation

After building the project we tested by input some data on it

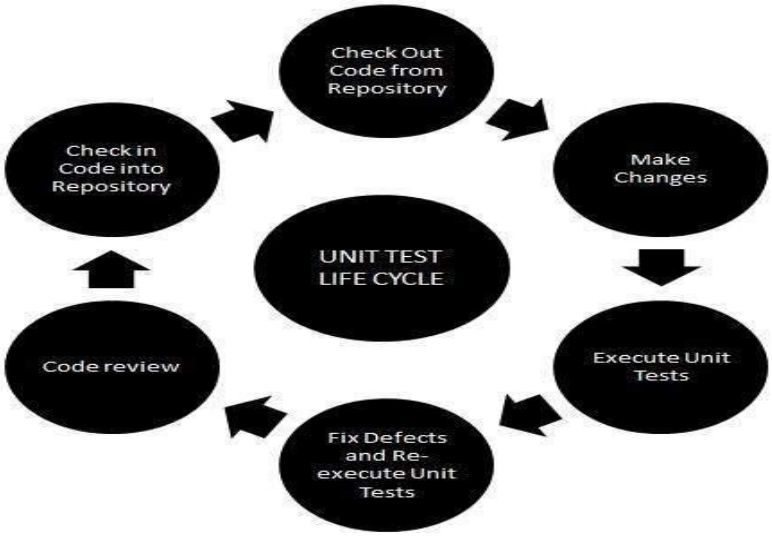


Here is the OJS file that works as database.



The execution worked well. No bug detected. User friendly.

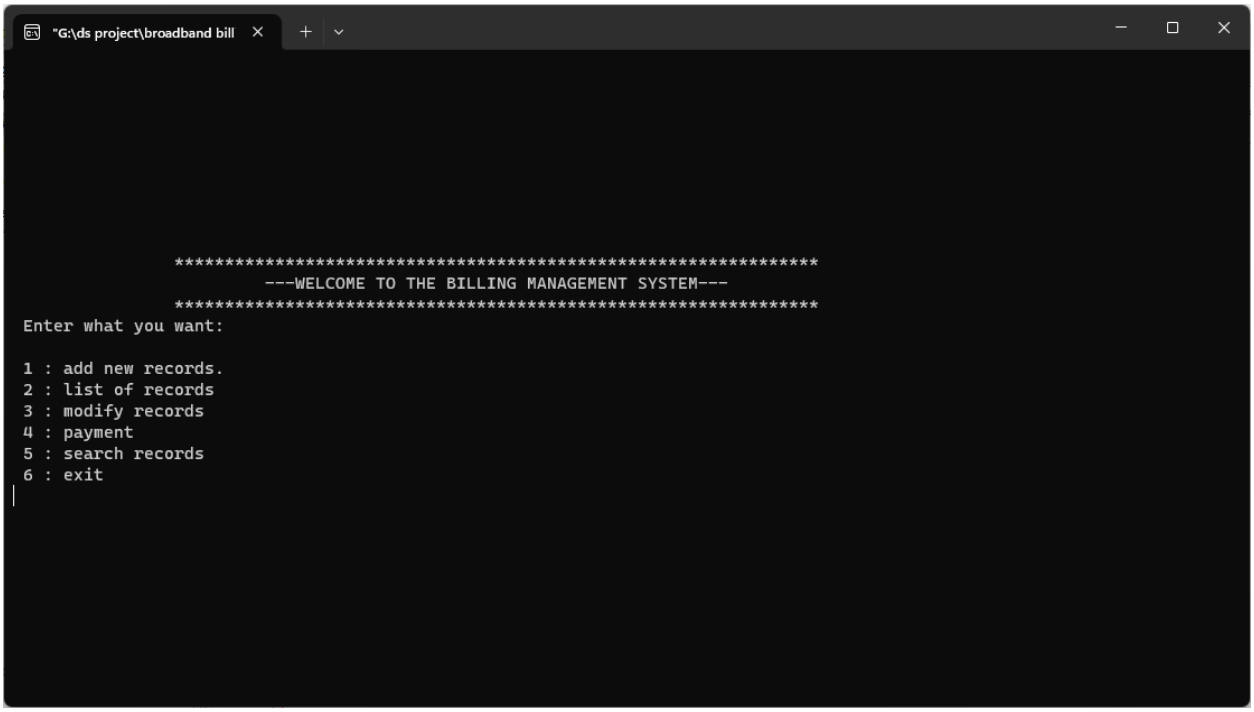
The data structure method implemented here optimizes whole projects as all the projects base depends on linked list and stack.



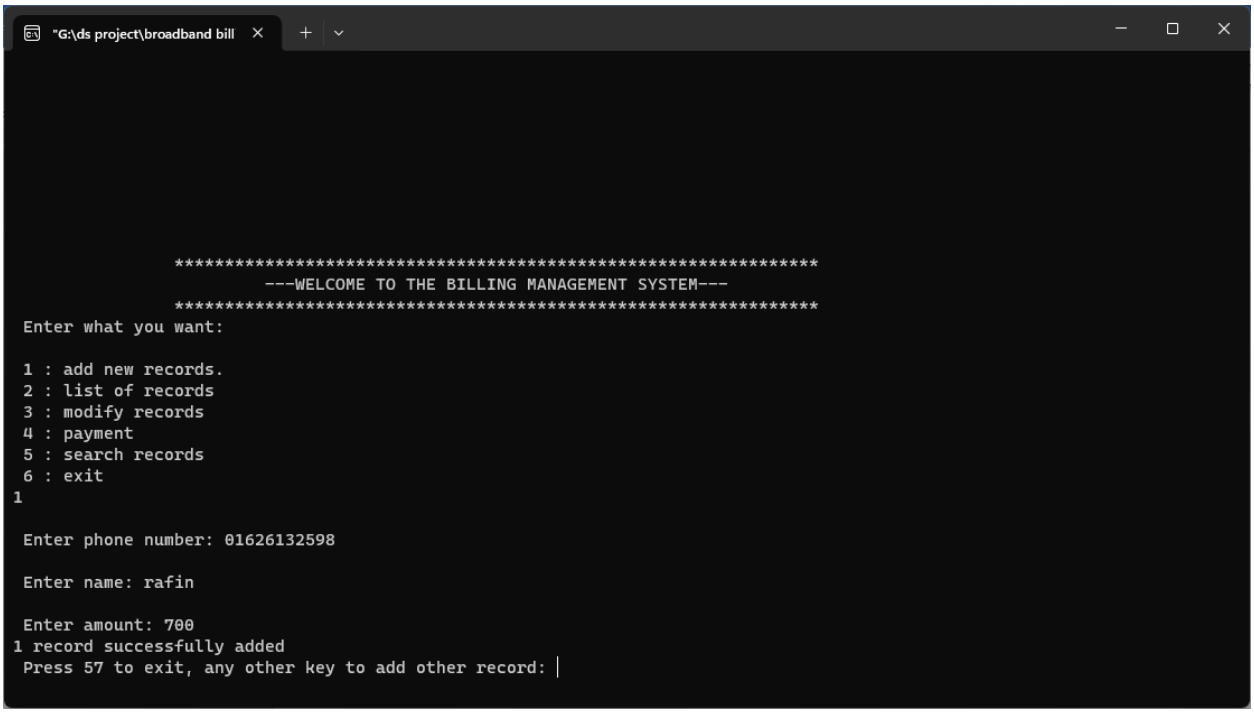
CHAPTER 3

3.1 Results

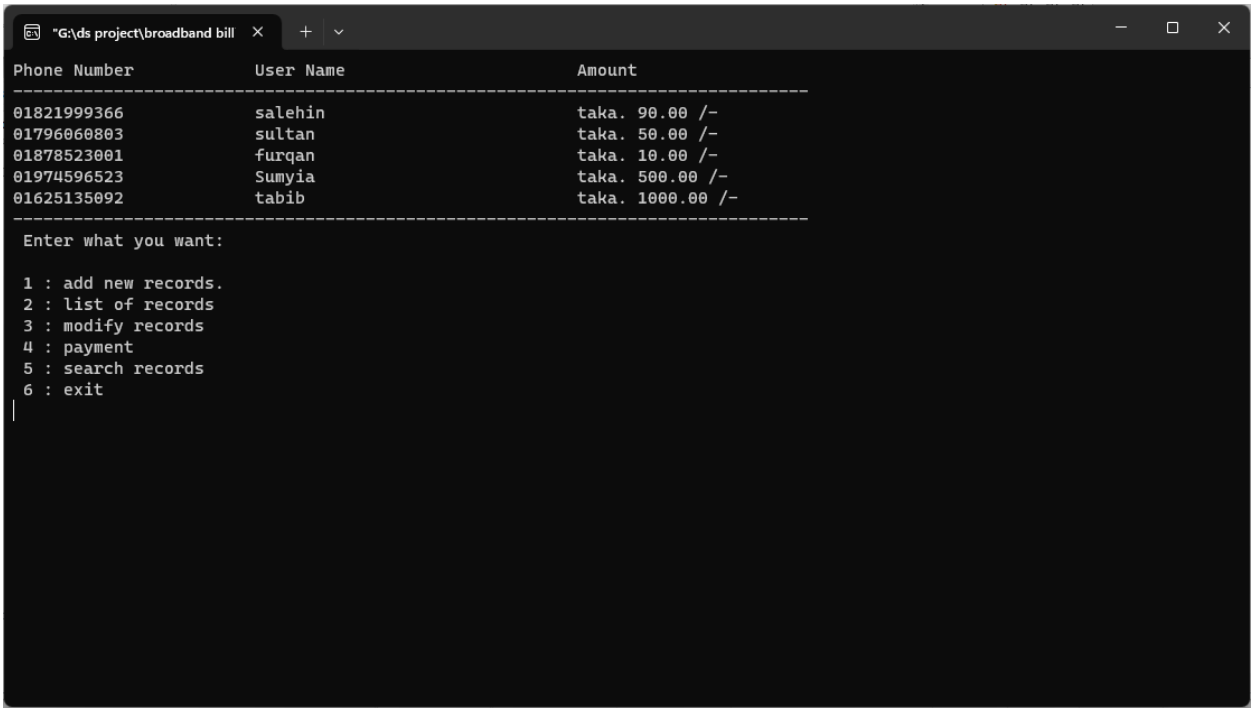
1.First interface:



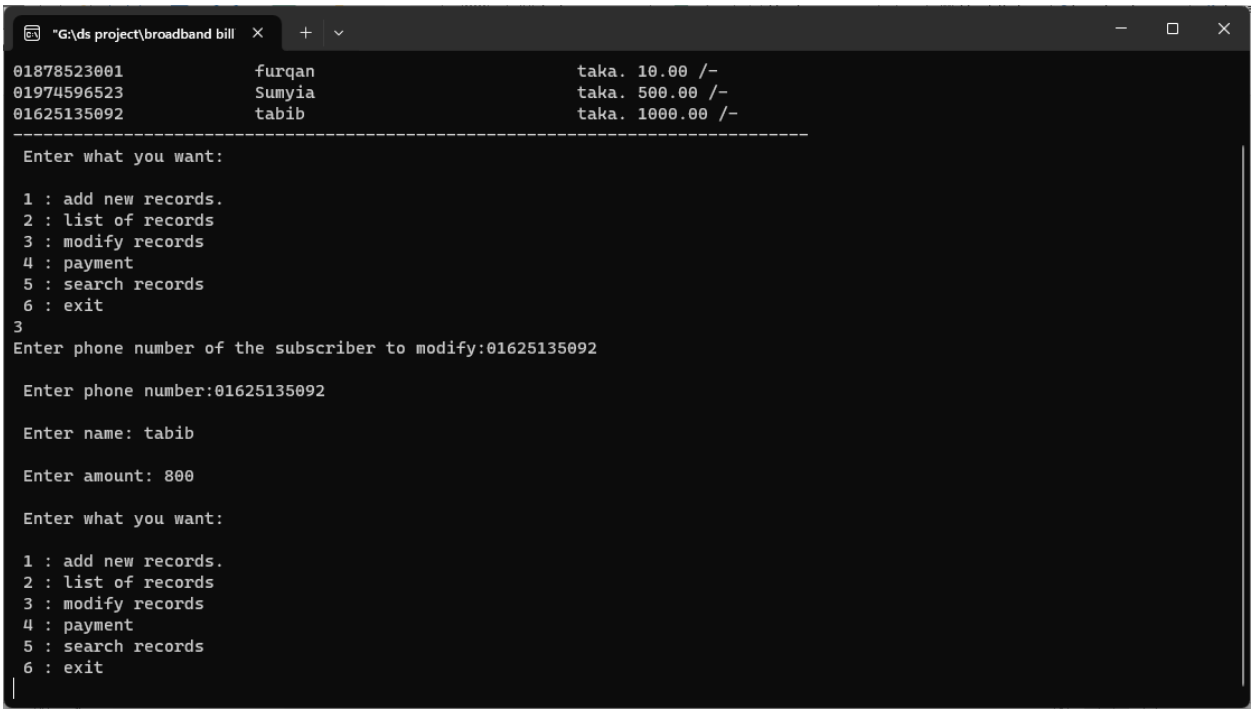
2.Add new records:



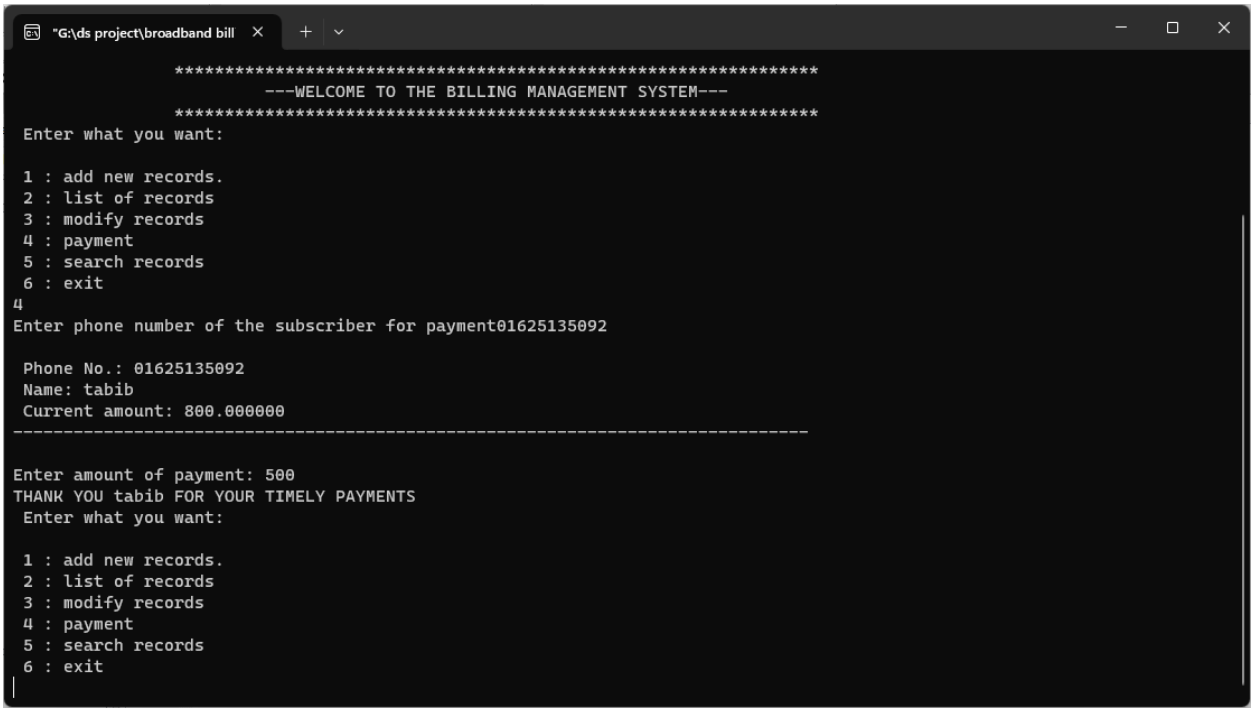
3.List of records:



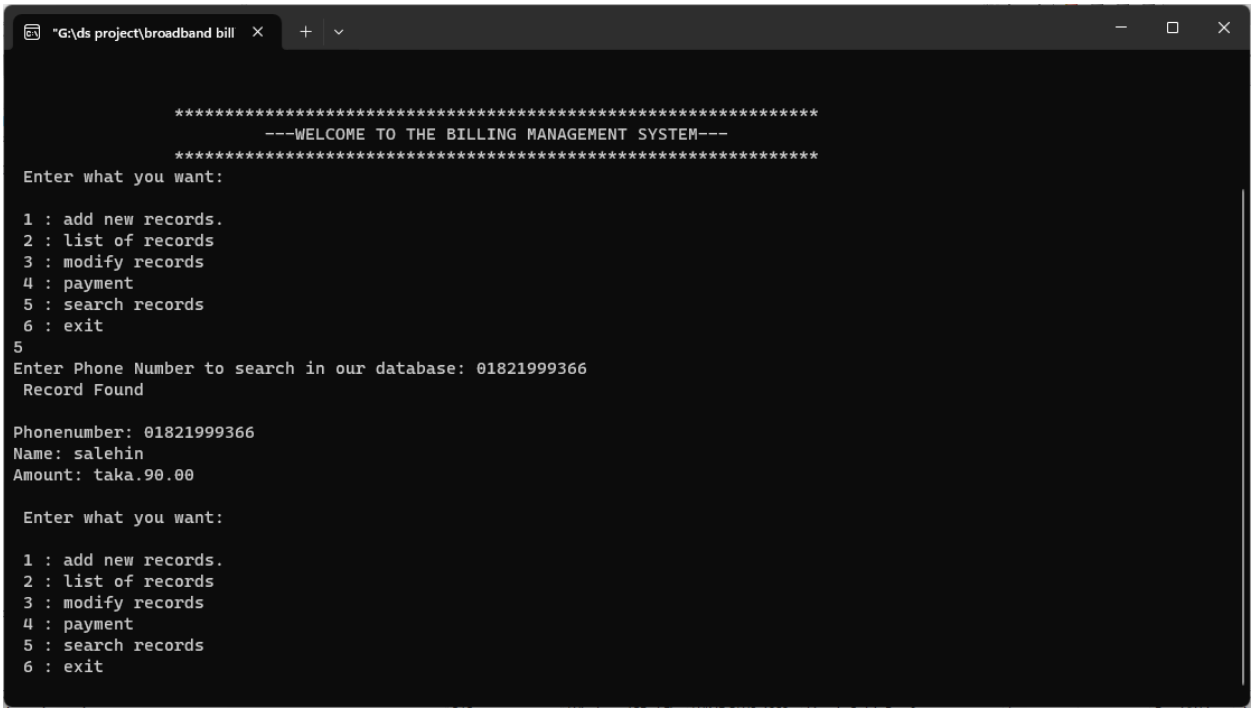
4.Modify records:



5.payment:



6.Search Records:



3.2 Discussion:

The whole output summarize the project details which is user friendly, easy to implement.

CHAPTER 4

4.1 Conclusion:

Although the billing system is easy to use it has some boundaries also. There is no security feature added to the system. The information runs on the basis of OJS file, if the file corrupted the whole project will crush. Otherwise the projects runs smooth and easy to implement all the available features.

4.1 Future Work:

1.integration with a graphical interface.

2.Cloudbase data storage option.

3.Enhanced Security features.

4.Mobile application development.

Reference

1. Telecom Billing System in C - GeeksforGeeks
2. Customer Billing System Project Using C Language - Studytonight