Department of Computer Science and Engineering

Project Report

Course Code: CSE-336

Course Title: Software Project VI

Project Title: Bus Management System

Submitted to:

Fariha Jahan

Lecturer

Department of CSE

Daffodil International University

Submitted by:

Name	Id
Sumaiya Imrose Anika	191-15-12348
Md. Faysal Ahmed	191-15-12294
Imran Hossain	191-15-12722
Aisharjo Chakrobortty	191-15-12334

Section: E

Date of submission: 15th December, 2021

BUS MANAGEMENT SYSTEM

Index page

2. Abstract:

Bus Management System (BMS) project is mainly developed for making our transport system more developed. In this digital era, this kind of digital system is really required. This project is mainly focused in digitalizing our bus management system (BMS). In this project customers have to register first. So the records of the customers can be kept. Then tickets can be booked using phone, computer, and some other devices by the account they have registered. In this project canceling ticket will also be included. Canceling tickets system will also be same as booking tickets. So, it won't be a hectic for the people to book or cancel a bus ticket. Choosing their destination won't be a problem too. Customers can also check another routes' ticket if there is any available. This project is secured too. The information of the bus can be installed. So, people won't have to worry and can review everything just by doing their other works. Storing data and the implementation of this developed system can do proper management and analysis.

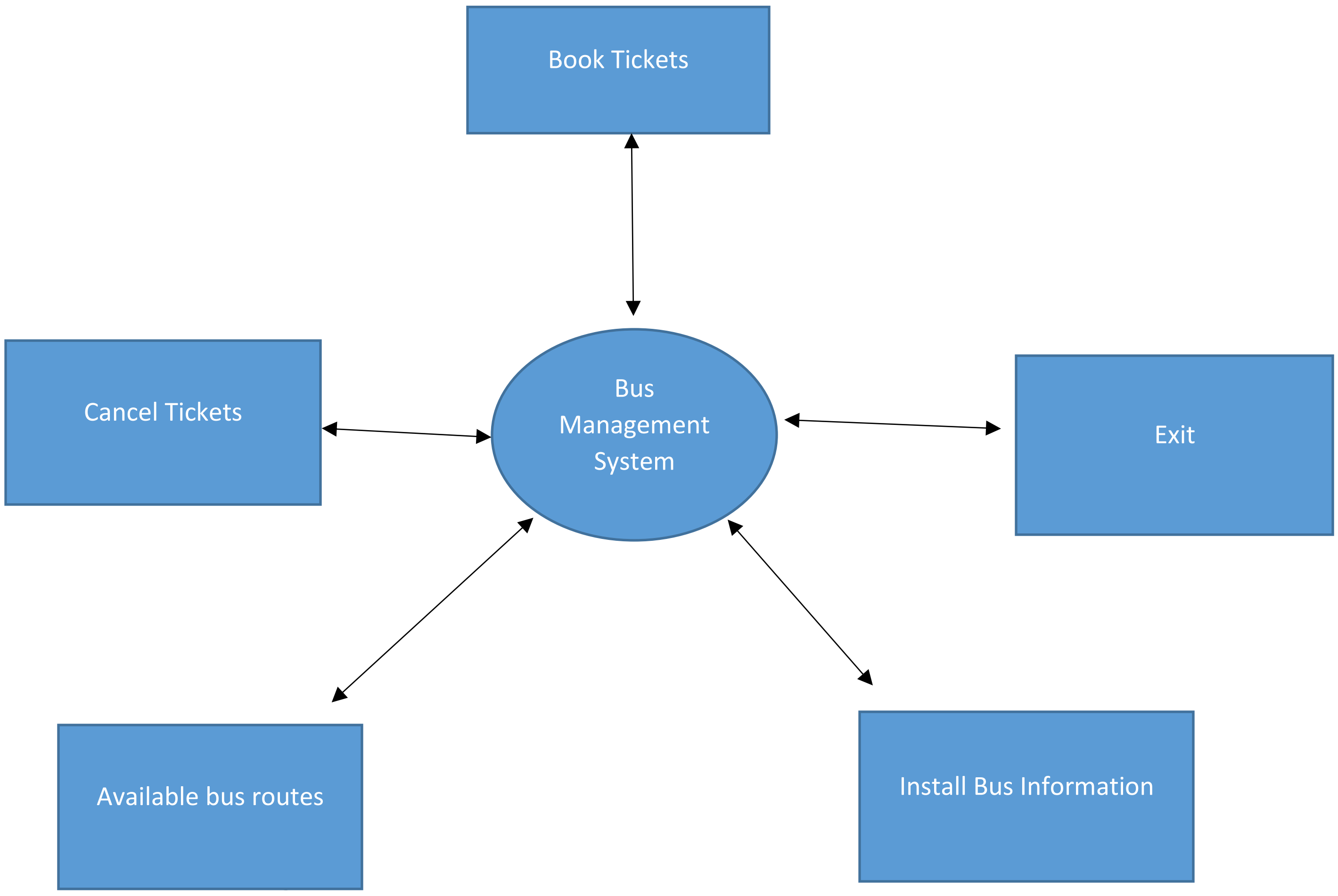
3. Introduction:

Our country is over populated country. So, more transports are needed. But the transport system isn't very good in our country. This sector has many flaws which is hampering our economy and other social sectors. Bus is one of the most required transports in our country. But the system is not organized, not disciplined and in a bad a shape. So, to make the system a little bit more organized, we think our project is be helpful. This will be helpful for people to save their time and energy. The main objectives of our project on bus management system (BMS) are managing the details of the buses, booking ticket more easily, canceling ticket in an easy way, showing available bus routes, install bus information. This project carries the basic step for the people who travel by bus. This system contains booking a ticket by knowing the destination. A ticket can also be canceled. So, any company can handle this system easily. For an over populated country like us, we need more organized system for every step. But that's not happening. That's why we are lacking behind compared to other country. So if we start some new and digitalized system, we can progress. So by starting with improving the transport

system means this bus management system, we can hope for some more organized system later.
If a company gives better service using this system, it can have good reviews and customer will like this system more. So, their management should be more organized and effective. They shouldn’t do any unnecessary things using this system which will make customers feeling insecure. Then they won’t believe this system and there won’t be any response in this system. So, company representation will play a vital role in this system.
Therefore, this project is also a healthy way for booking and canceling tickets and minimize our time and energy. This can be hoped that this project will be a great help for the people.

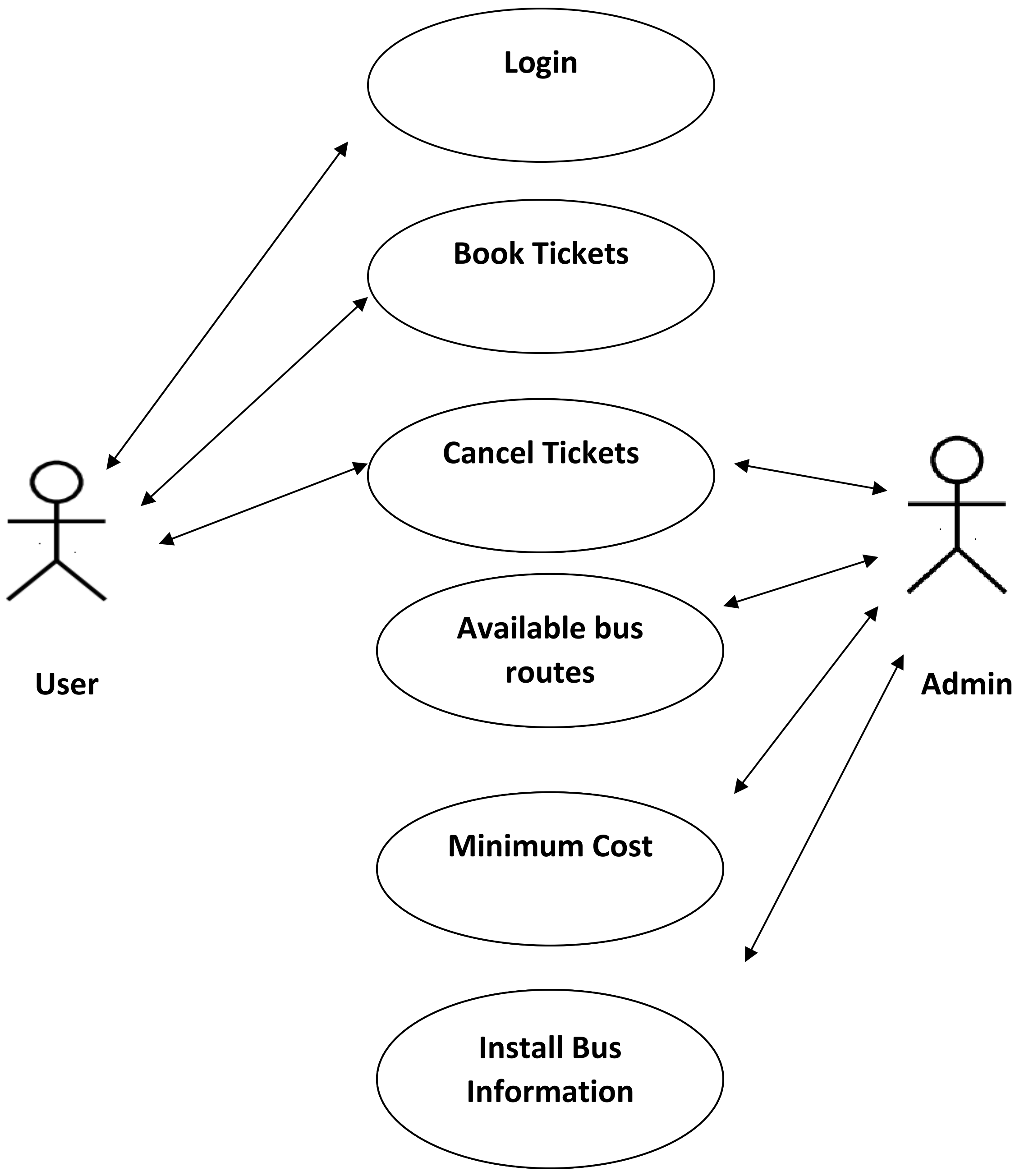
4.Design & Architecture:

Bus Management System Dataflow Diagram



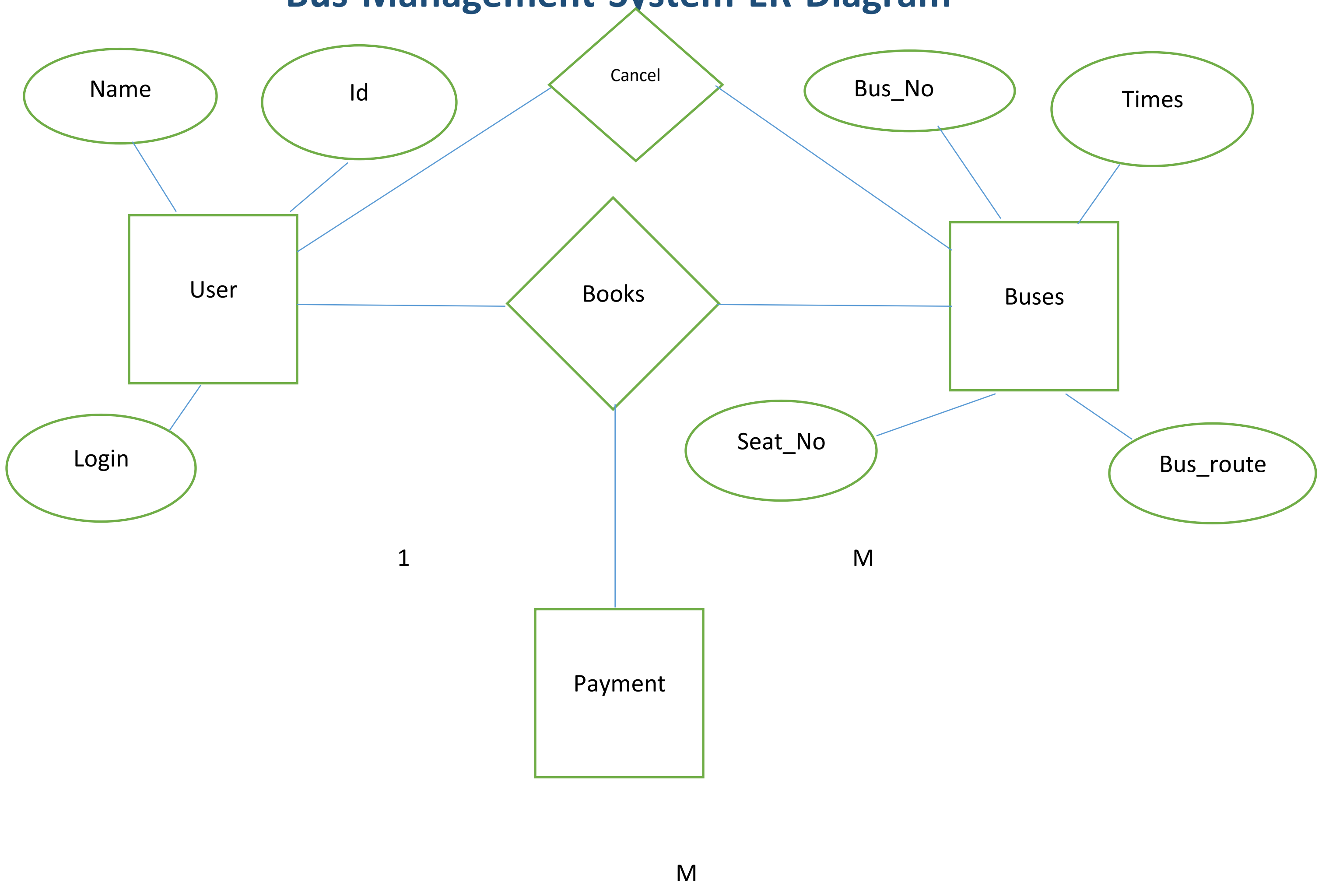
shows how the system is divided into sub-systems, each of which deals with one or more of the data flows to or form and external agent and which together provide all of the functionality of the bus booking system as a whole. It also identifies internal data stores of login, operating, ticket booking, sales, customer that must be present in order for the bus system to do its job, and shows the flow of the data between the various parts of bus, customer, log in. It provides a more detailed breakout of pieces of the level DFD.

Bus Management System Case-Diagram



This is use case diagram is a graphic depiction of the interactions among the elements of bus ticket booking system. It represents the methodology used in the system analysis, clarify and organize system requirement of bus management system. The main actors of bus ticket booking system in this use case diagram are: admin, user who perform the different types of use case such as booking ticket, Cancel Tickets, Available Bus Operators.

Bus-Management-System-ER-Diagram



This ER (Entity Relationship) Diagram represents the model of Bus Management System. The (Entity-Relationship) Diagram of Bus Management System shows all the visual instrument of database table and the relations between booking system. Bus route, seat etc.in this data is used define the relationships between structured data groups of Bus Management system functions. The main entities of the Bus Management system are Book Tickets, Cancel Tickets, Available Bus Operators, Exit.

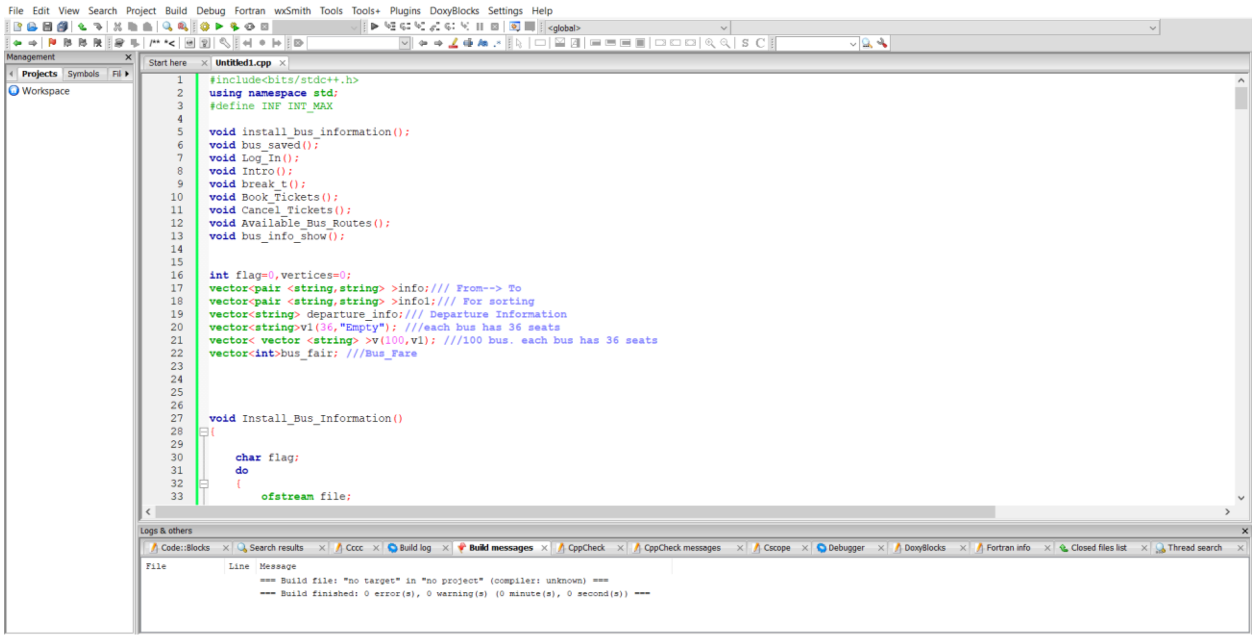
Bus Route Entity: Attributes of bus management system are bus_route, bus_no, seat_no, times.

Booking Entity: User, payment, Buses.

User Entity: Name, id, login.

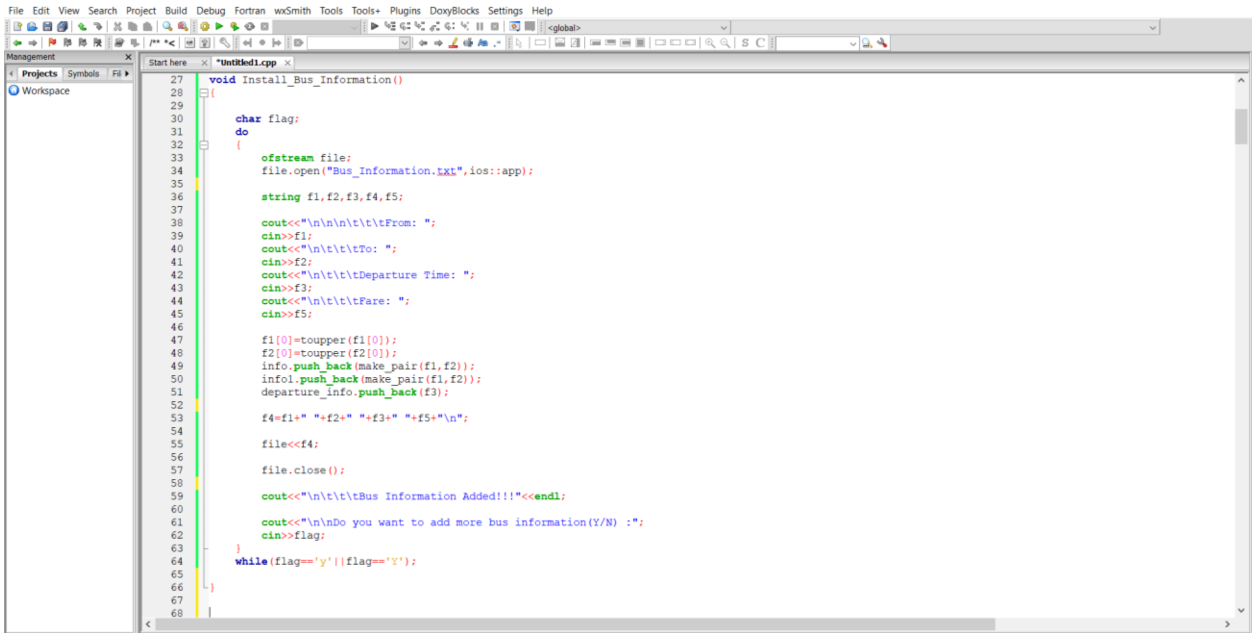
Bus Entity: attributes of Bus are bus_id, bus_no, bus_seat, bus_ticket, bus_description.

5.Source Code

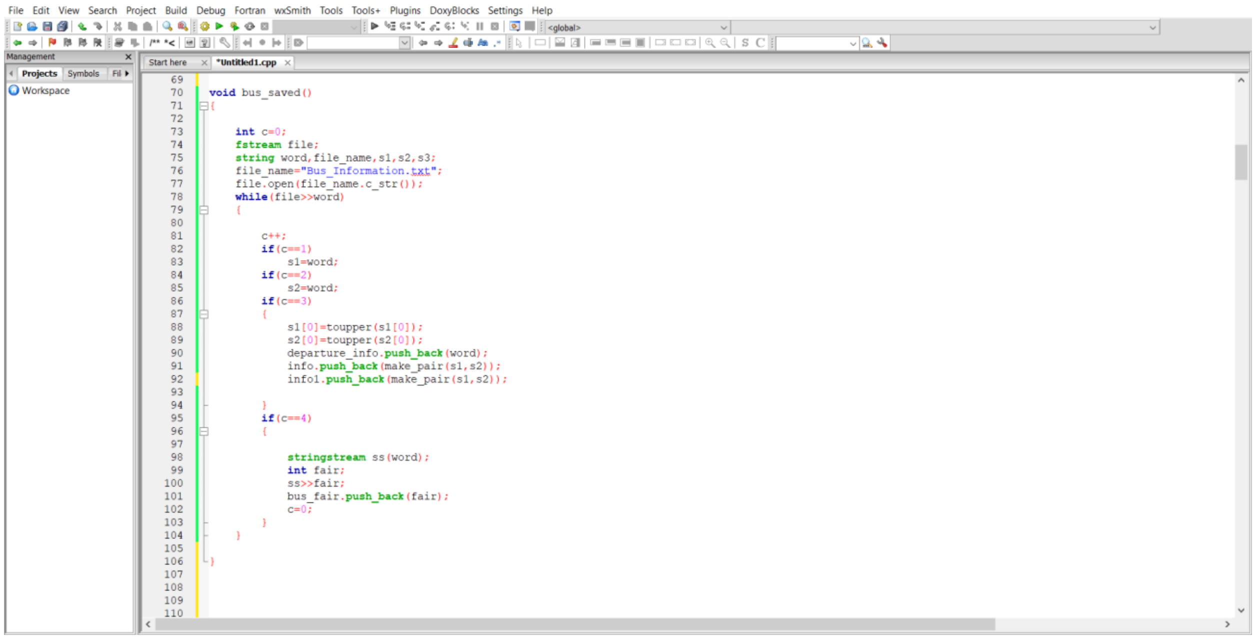


```
#include<bits/stdc++.h>
using namespace std;
#define INF INT_MAX

void install_bus_information();
void bus_saved();
void Log_In();
void Intro();
void break_t();
void Book_Tickets();
void Cancel_Tickets();
void Available_Bus_Routes();
void bus_info_show();


int flag=0,vertices=0;
vector<pair <string,string> >info;/// From--> To
vector<pair <string,string> >info1;/// For sorting
vector<string> departure_info;/// Departure Information
vector<string>v1(36,"Empty"); ///each bus has 36 seats
vector< vector <string> >v(100,v1); ///100 bus. each bus has 36 seats
vector<int>bus_fair; ///Bus_Fare



void Install_Bus_Information()
{

    char flag;
    do
    {
        ofstream file;
```

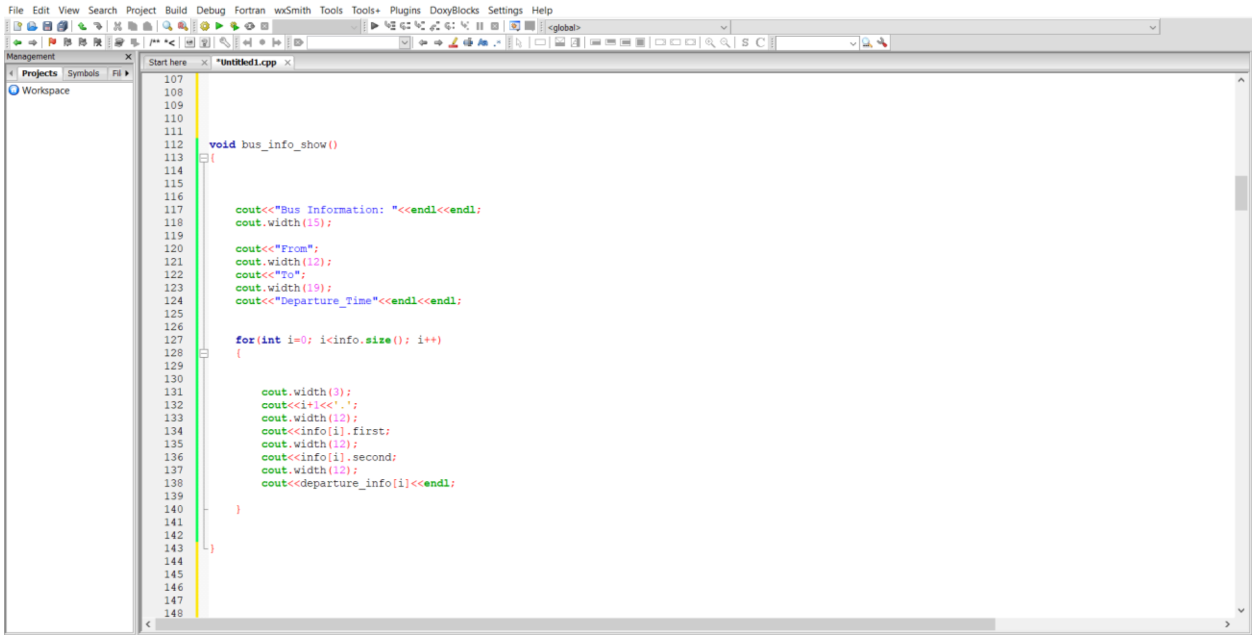
```
File            Line  Message
                      === Build file: "no target" in "no project" (compiler: unknown) ===
                      === Build finished: 0 error(s), 0 warning(s) (0 minute(s), 0 second(s)) ===
```



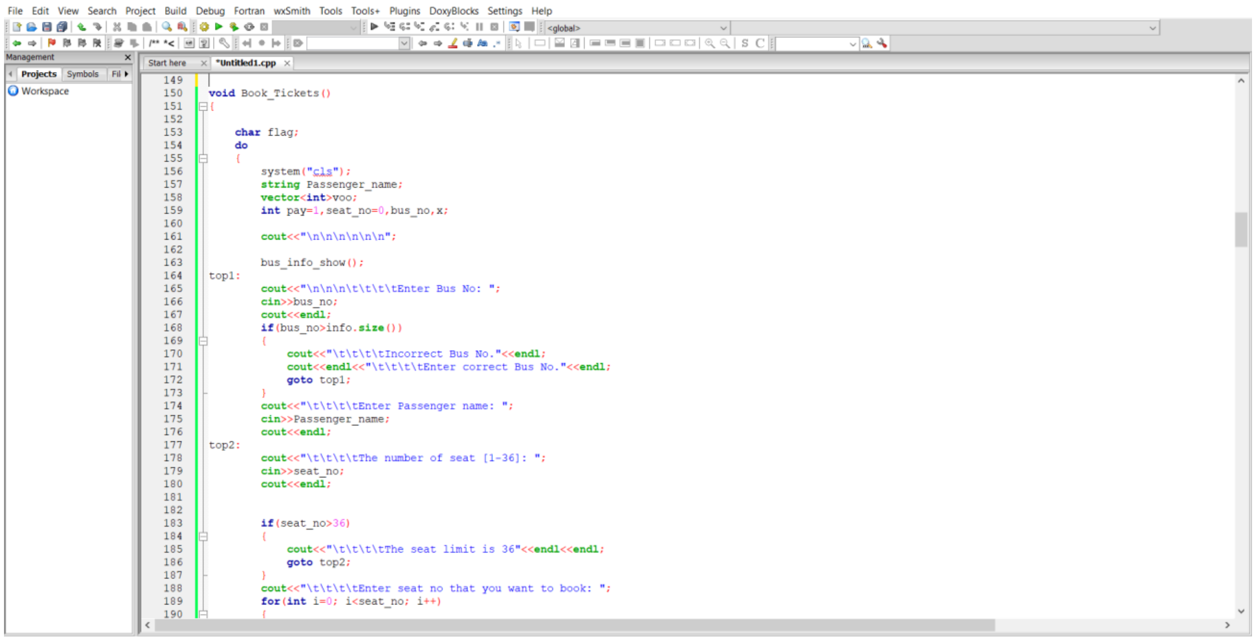
```
void Install_Bus_Information()
{

    char flag;
    do
    {
        ofstream file;
        file.open("Bus_Information.txt",ios::app);

        string f1,f2,f3,f4,f5;

        cout<<"\n\n\n\t\t\tFrom: ";
        cin>>f1;
        cout<<"\n\t\t\tTo: ";
        cin>>f2;
        cout<<"\n\t\t\tDeparture Time: ";
        cin>>f3;
        cout<<"\n\t\t\tFare: ";
        cin>>f5;

        f1[0]=toupper(f1[0]);
        f2[0]=toupper(f2[0]);
        info.push_back(make_pair(f1,f2));
        info1.push_back(make_pair(f1,f2));
        departure_info.push_back(f3);

        f4=f1+" "+f2+" "+f3+" "+f5+"\n";

        file<<f4;

        file.close();

        cout<<"\n\t\t\tBus Information Added!!!"<<endl;

        cout<<"\n\nDo you want to add more bus information(Y/N) :";
        cin>>flag;
    }
    while(flag=='y'||flag=='Y');

}

```



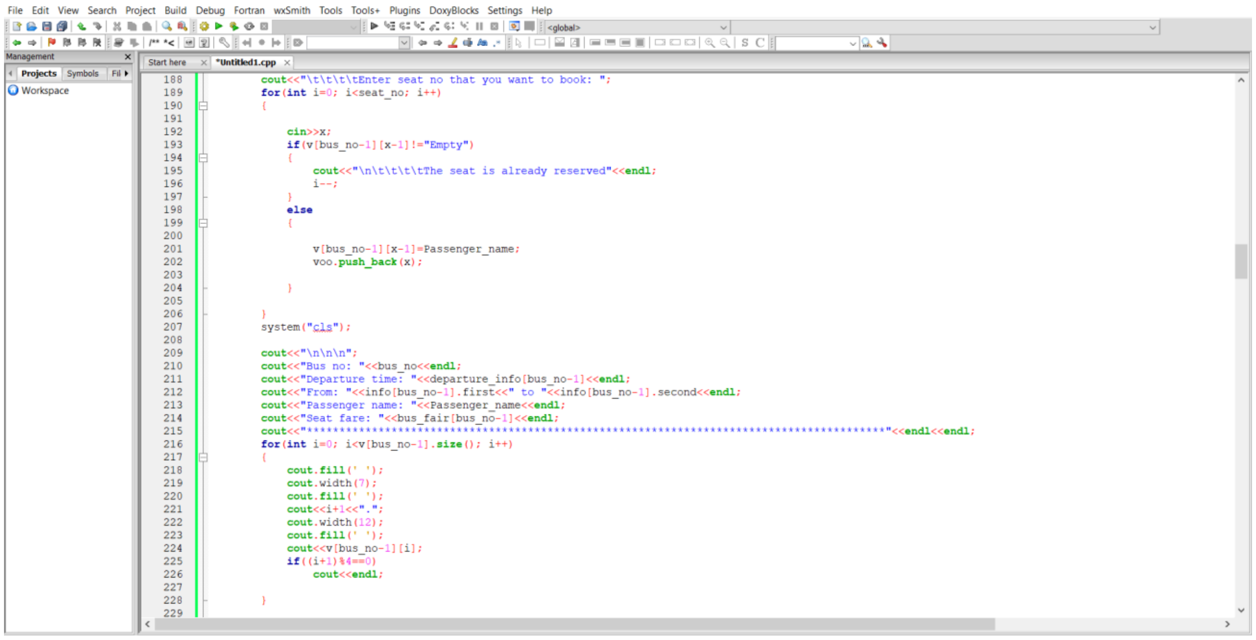
```cpp

void bus_saved()
{

    int c=0;
    fstream file;
    string word,file_name,s1,s2,s3;
    file_name="Bus_Information.txt";
    file.open(file_name.c_str());
    while(file>>word)
    {

        c++;
        if(c==1)
            s1=word;
        if(c==2)
            s2=word;
        if(c==3)
        {
            s1[0]=toupper(s1[0]);
            s2[0]=toupper(s2[0]);
            departure_info.push_back(word);
            info.push_back(make_pair(s1,s2));
            info1.push_back(make_pair(s1,s2));

        }
        if(c==4)
        {

            stringstream ss(word);
            int fair;
            ss>>fair;
            bus_fair.push_back(fair);
            c=0;
        }
    }

}




```



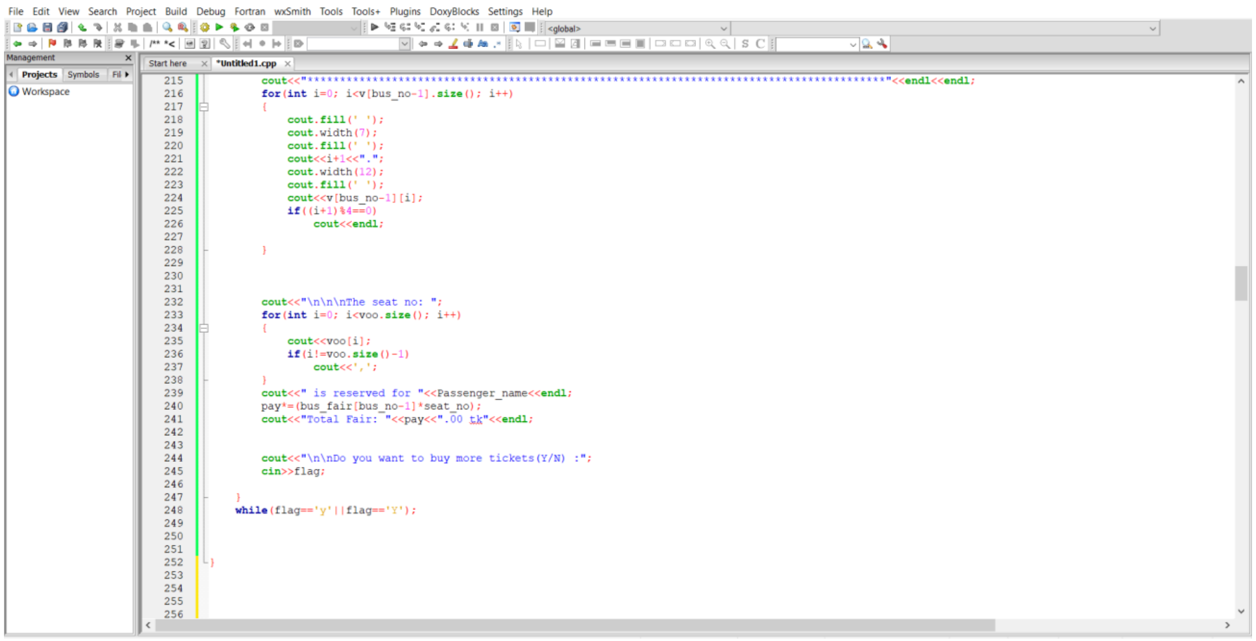
```cpp





void bus_info_show()
{



    cout<<"Bus Information: "<<endl<<endl;
    cout.width(15);

    cout<<"From";
    cout.width(12);
    cout<<"To";
    cout.width(19);
    cout<<"Departure_Time"<<endl<<endl;


    for(int i=0; i<info.size(); i++)
    {


        cout.width(3);
        cout<<i+1<<'.';
        cout.width(12);
        cout<<info[i].first;
        cout.width(12);
        cout<<info[i].second;
        cout.width(12);
        cout<<departure_info[i]<<endl;

    }


}





```



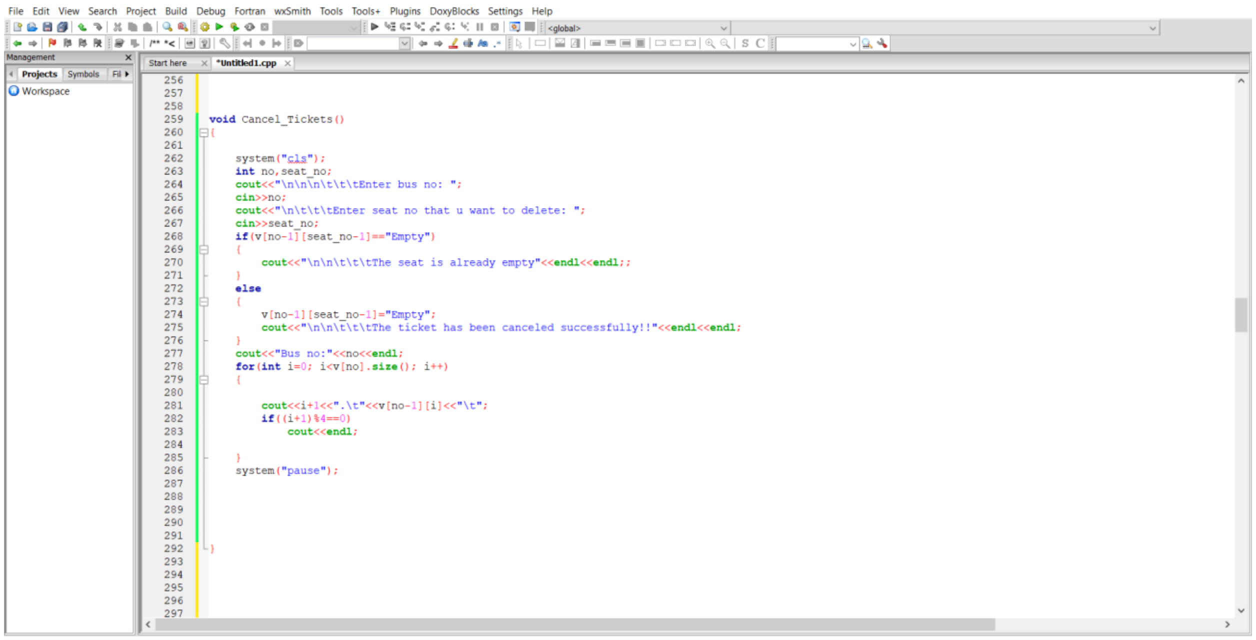
```cpp

void Book_Tickets()
{

    char flag;
    do
    {
        system("cls");
        string Passenger_name;
        vector<int>voo;
        int pay=1,seat_no=0,bus_no,x;

        cout<<"\n\n\n\n\n\n";

        bus_info_show();
top1:
        cout<<"\n\n\n\t\t\t\tEnter Bus No: ";
        cin>>bus_no;
        cout<<endl;
        if(bus_no>info.size())
        {
            cout<<"\t\t\t\tIncorrect Bus No."<<endl;
            cout<<endl<<"\t\t\t\tEnter correct Bus No."<<endl;
            goto top1;
        }
        cout<<"\t\t\t\tEnter Passenger name: ";
        cin>>Passenger_name;
        cout<<endl;
top2:
        cout<<"\t\t\t\tThe number of seat [1-36]: ";
        cin>>seat_no;
        cout<<endl;


        if(seat_no>36)
        {
            cout<<"\t\t\t\tThe seat limit is 36"<<endl<<endl;
            goto top2;
        }
        cout<<"\t\t\t\tEnter seat no that you want to book: ";
        for(int i=0; i<seat_no; i++)
        {
```



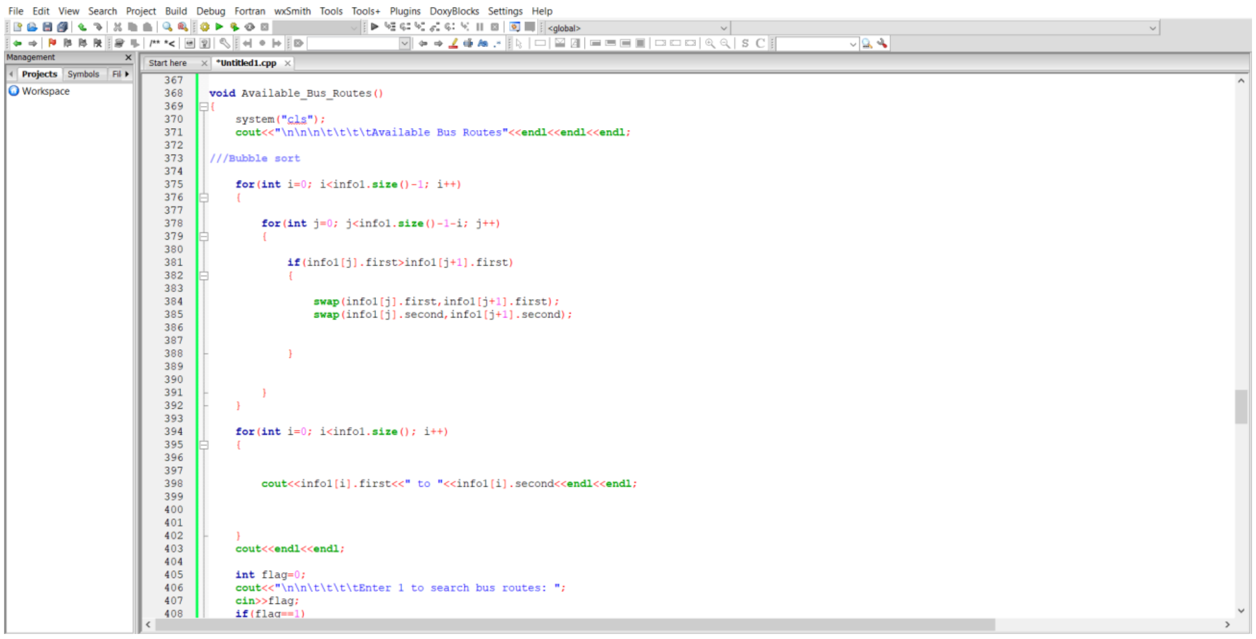
```cpp
        cout<<"\t\t\t\tEnter seat no that you want to book: ";
        for(int i=0; i<seat_no; i++)
        {

            cin>>x;
            if(v[bus_no-1][x-1]!="Empty")
            {
                cout<<"\n\t\t\t\tThe seat is already reserved"<<endl;
                i--;
            }
            else
            {

                v[bus_no-1][x-1]=Passenger_name;
                voo.push_back(x);

            }

        }
        system("cls");

        cout<<"\n\n\n";
        cout<<"Bus no: "<<bus_no<<endl;
        cout<<"Departure time: "<<departure_info[bus_no-1]<<endl;
        cout<<"From: "<<info[bus_no-1].first<<" to "<<info[bus_no-1].second<<endl;
        cout<<"Passenger name: "<<Passenger_name<<endl;
        cout<<"Seat fare: "<<bus_fair[bus_no-1]<<endl;
        cout<<"****************************************************************************************"<<endl<<endl;
        for(int i=0; i<v[bus_no-1].size(); i++)
        {
            cout.fill(' ');
            cout.width(7);
            cout.fill(' ');
            cout<<i+1<<".";
            cout.width(12);
            cout.fill(' ');
            cout<<v[bus_no-1][i];
            if((i+1)%4==0)
                cout<<endl;

        }

```



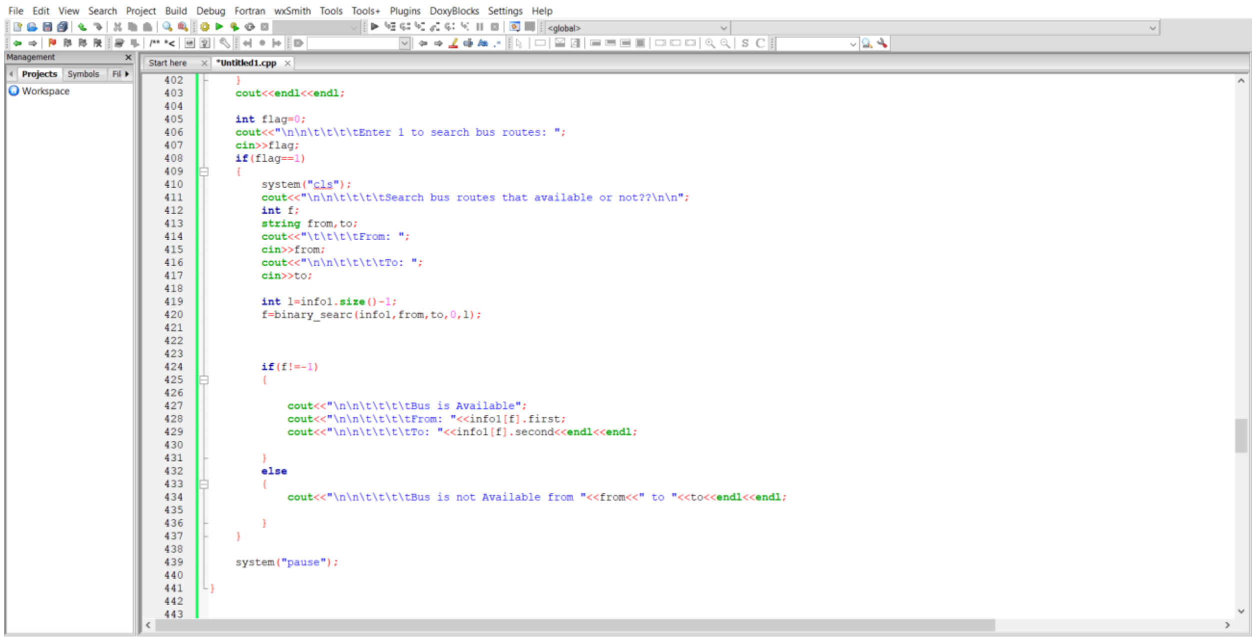
```cpp
        cout<<"*****************************************************************************************"<<endl<<endl;
        for(int i=0; i<v[bus_no-1].size(); i++)
        {
            cout.fill(' ');
            cout.width(7);
            cout.fill(' ');
            cout<<i+1<<".";
            cout.width(12);
            cout.fill(' ');
            cout<<v[bus_no-1][i];
            if((i+1)%4==0)
                cout<<endl;

        }



        cout<<"\n\n\nThe seat no: ";
        for(int i=0; i<voo.size(); i++)
        {
            cout<<voo[i];
            if(i!=voo.size()-1)
                cout<<',';
        }
        cout<<" is reserved for "<<Passenger_name<<endl;
        pay*=(bus_fair[bus_no-1]*seat_no);
        cout<<"Total Fair: "<<pay<<".00 tk"<<endl;


        cout<<"\n\nDo you want to buy more tickets(Y/N) :";
        cin>>flag;

    }
    while(flag=='y'||flag=='Y');



}
```



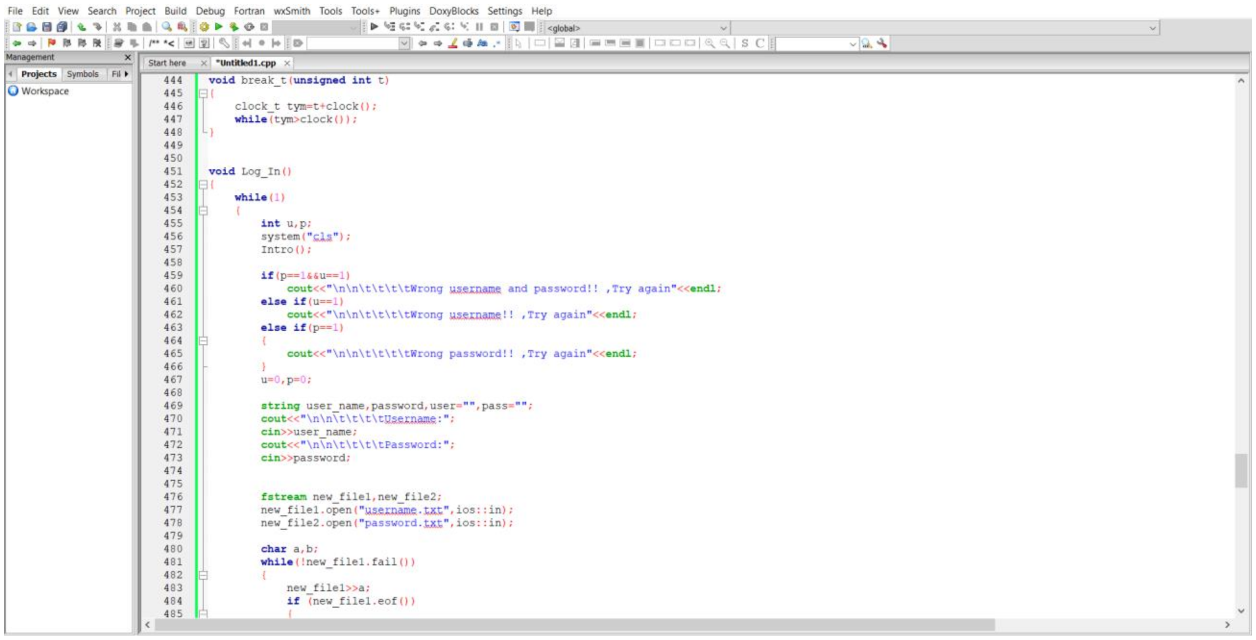
```cpp



void Cancel_Tickets()
{

    system("cls");
    int no,seat_no;
    cout<<"\n\n\n\t\t\tEnter bus no: ";
    cin>>no;
    cout<<"\n\t\t\tEnter seat no that u want to delete: ";
    cin>>seat_no;
    if(v[no-1][seat_no-1]=="Empty")
    {
        cout<<"\n\n\t\t\tThe seat is already empty"<<endl<<endl;;
    }
    else
    {
        v[no-1][seat_no-1]="Empty";
        cout<<"\n\n\t\t\tThe ticket has been canceled successfully!!"<<endl<<endl;
    }
    cout<<"Bus no:"<<no<<endl;
    for(int i=0; i<v[no].size(); i++)
    {

        cout<<i+1<<".\t"<<v[no-1][i]<<"\t";
        if((i+1)%4==0)
            cout<<endl;

    }
    system("pause");





}
```



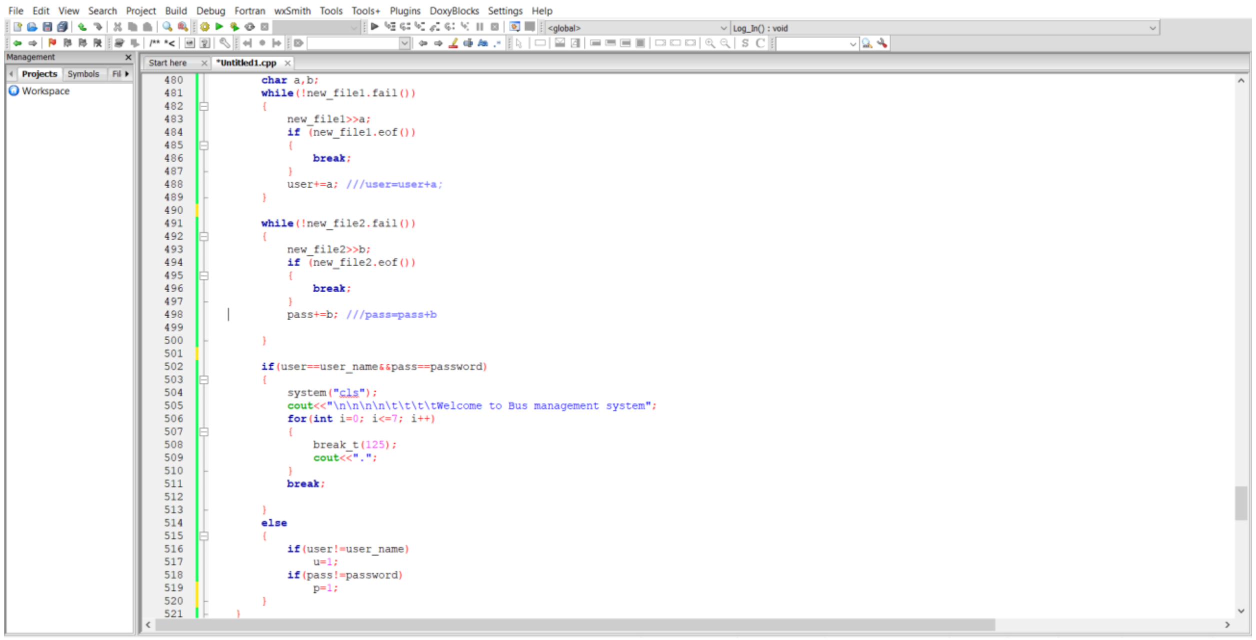
```cpp

void Available_Bus_Routes()
{
    system("cls");
    cout<<"\n\n\n\t\t\t\tAvailable Bus Routes"<<endl<<endl<<endl;

///Bubble sort

    for(int i=0; i<info1.size()-1; i++)
    {

        for(int j=0; j<info1.size()-1-i; j++)
        {

            if(info1[j].first>info1[j+1].first)
            {

                swap(info1[j].first,info1[j+1].first);
                swap(info1[j].second,info1[j+1].second);


            }


        }
    }

    for(int i=0; i<info1.size(); i++)
    {


        cout<<info1[i].first<<" to "<<info1[i].second<<endl<<endl;



    }
    cout<<endl<<endl;

    int flag=0;
    cout<<"\n\n\t\t\t\tEnter 1 to search bus routes: ";
    cin>>flag;
    if(flag==1)
```



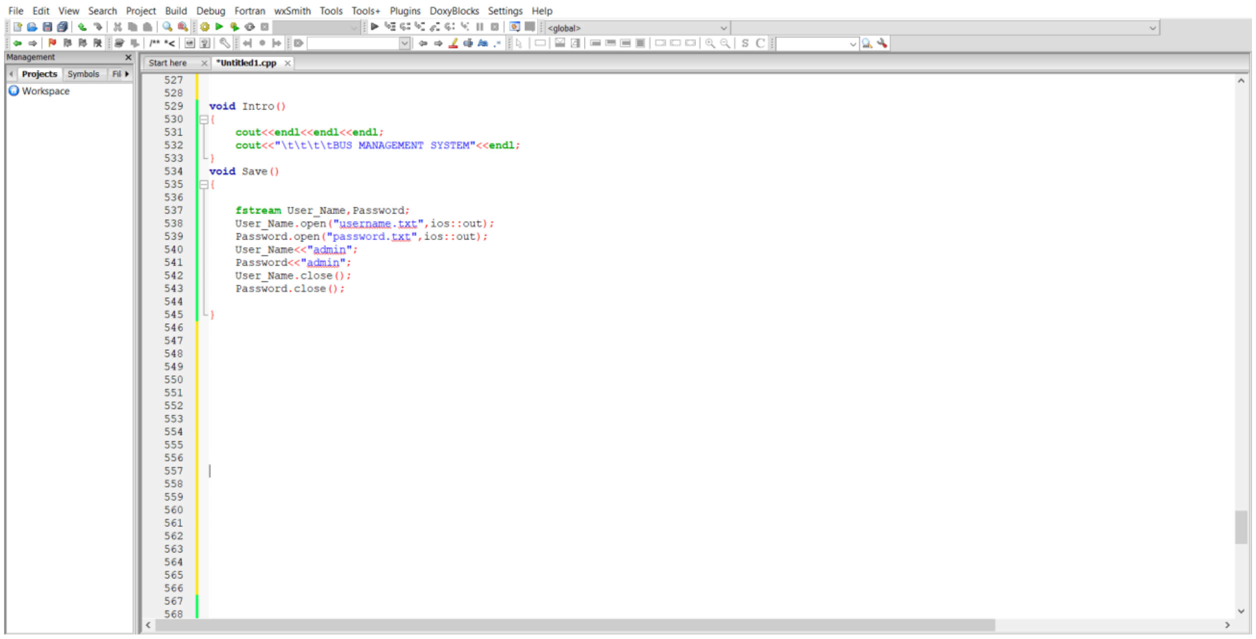
```cpp
    }
    cout<<endl<<endl;

    int flag=0;
    cout<<"\n\n\t\t\t\tEnter 1 to search bus routes: ";
    cin>>flag;
    if(flag==1)
    {
        system("cls");
        cout<<"\n\n\t\t\t\tSearch bus routes that available or not??\n\n";
        int f;
        string from,to;
        cout<<"\t\t\t\tFrom: ";
        cin>>from;
        cout<<"\n\n\t\t\t\tTo: ";
        cin>>to;

        int l=info1.size()-1;
        f=binary_searc(info1,from,to,0,l);



        if(f!=-1)
        {

            cout<<"\n\n\t\t\t\tBus is Available";
            cout<<"\n\n\t\t\t\tFrom: "<<info1[f].first;
            cout<<"\n\n\t\t\t\tTo: "<<info1[f].second<<endl<<endl;

        }
        else
        {
            cout<<"\n\n\t\t\t\tBus is not Available from "<<from<<" to "<<to<<endl<<endl;

        }
    }

    system("pause");

}


```



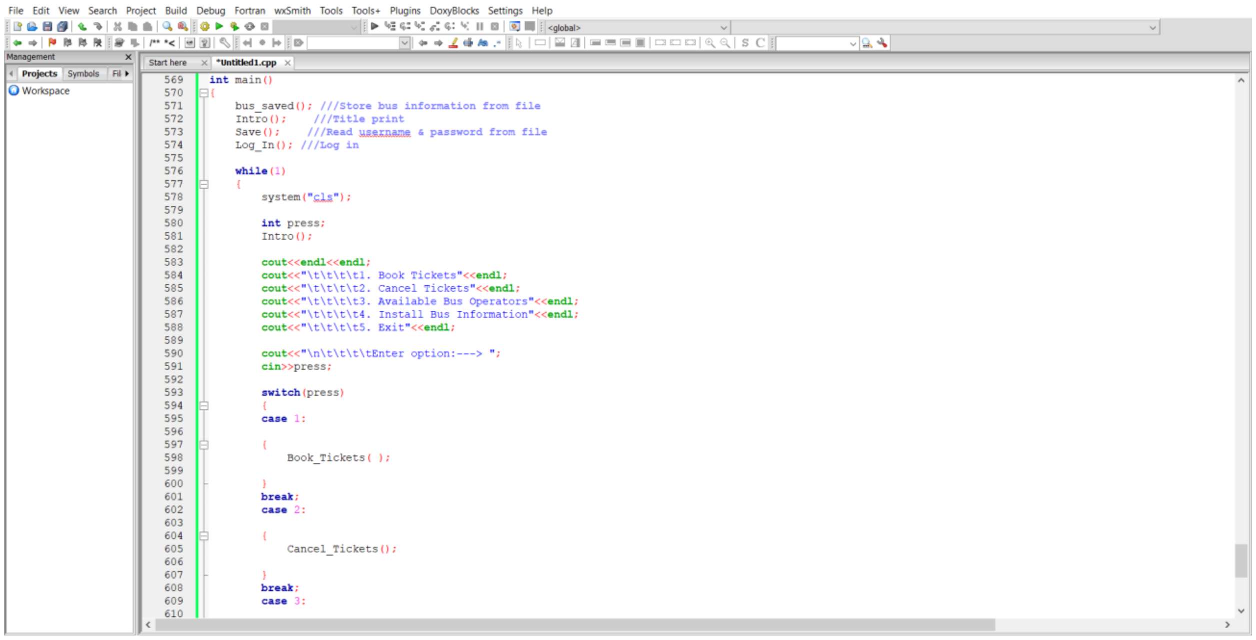
```cpp
void break_t(unsigned int t)
{
    clock_t tym=t+clock();
    while(tym>clock());
}


void Log_In()
{
    while(1)
    {
        int u,p;
        system("cls");
        Intro();

        if(p==1&&u==1)
            cout<<"\n\n\t\t\t\tWrong username and password!! ,Try again"<<endl;
        else if(u==1)
            cout<<"\n\n\t\t\t\tWrong username!! ,Try again"<<endl;
        else if(p==1)
        {
            cout<<"\n\n\t\t\t\tWrong password!! ,Try again"<<endl;
        }
        u=0,p=0;

        string user_name,password,user="",pass="";
        cout<<"\n\n\t\t\t\tUsername:";
        cin>>user_name;
        cout<<"\n\n\t\t\t\tPassword:";
        cin>>password;


        fstream new_file1,new_file2;
        new_file1.open("username.txt",ios::in);
        new_file2.open("password.txt",ios::in);

        char a,b;
        while(!new_file1.fail())
        {
            new_file1>>a;
            if (new_file1.eof())
            {
```



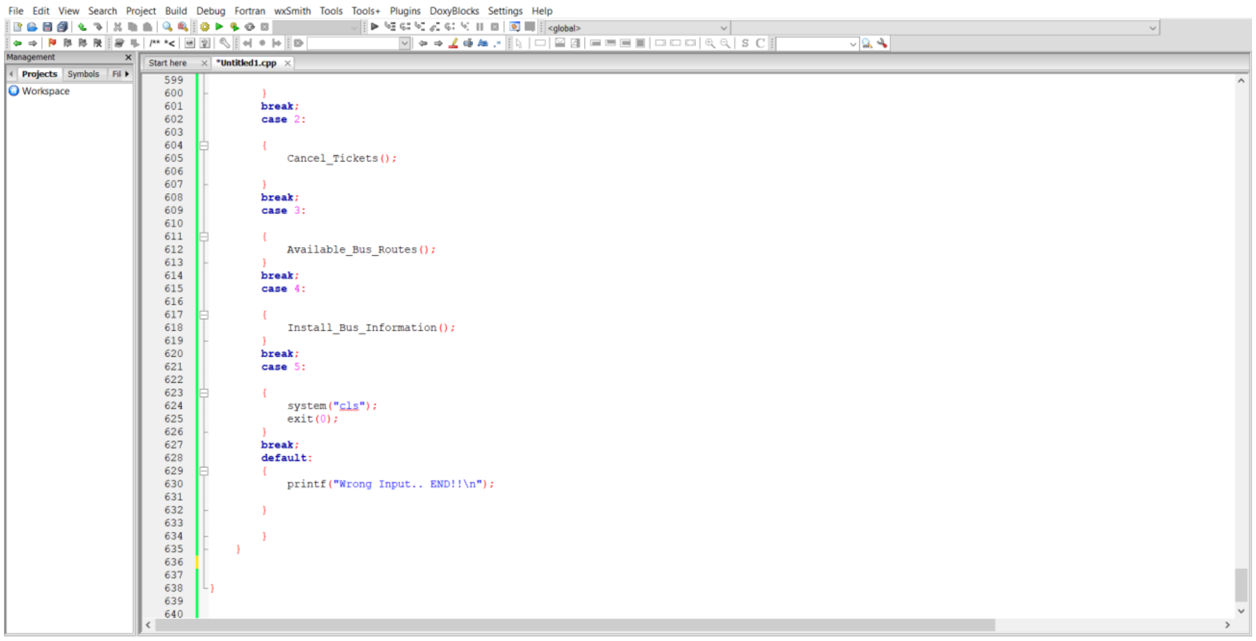
```cpp
        char a,b;
        while(!new_file1.fail())
        {
            new_file1>>a;
            if (new_file1.eof())
            {
                break;
            }
            user+=a; ///user=user+a;
        }

        while(!new_file2.fail())
        {
            new_file2>>b;
            if (new_file2.eof())
            {
                break;
            }
            pass+=b; ///pass=pass+b

        }

        if(user==user_name&&pass==password)
        {
            system("cls");
            cout<<"\n\n\n\n\t\t\t\tWelcome to Bus management system";
            for(int i=0; i<=7; i++)
            {
                break_t(125);
                cout<<".";
            }
            break;

        }
        else
        {
            if(user!=user_name)
                u=1;
            if(pass!=password)
                p=1;
        }
    }
```



```cpp


void Intro()
{
    cout<<endl<<endl<<endl;
    cout<<"\t\t\t\tBUS MANAGEMENT SYSTEM"<<endl;
}
void Save()
{

    fstream User_Name,Password;
    User_Name.open("username.txt",ios::out);
    Password.open("password.txt",ios::out);
    User_Name<<"admin";
    Password<<"admin";
    User_Name.close();
    Password.close();

}
```



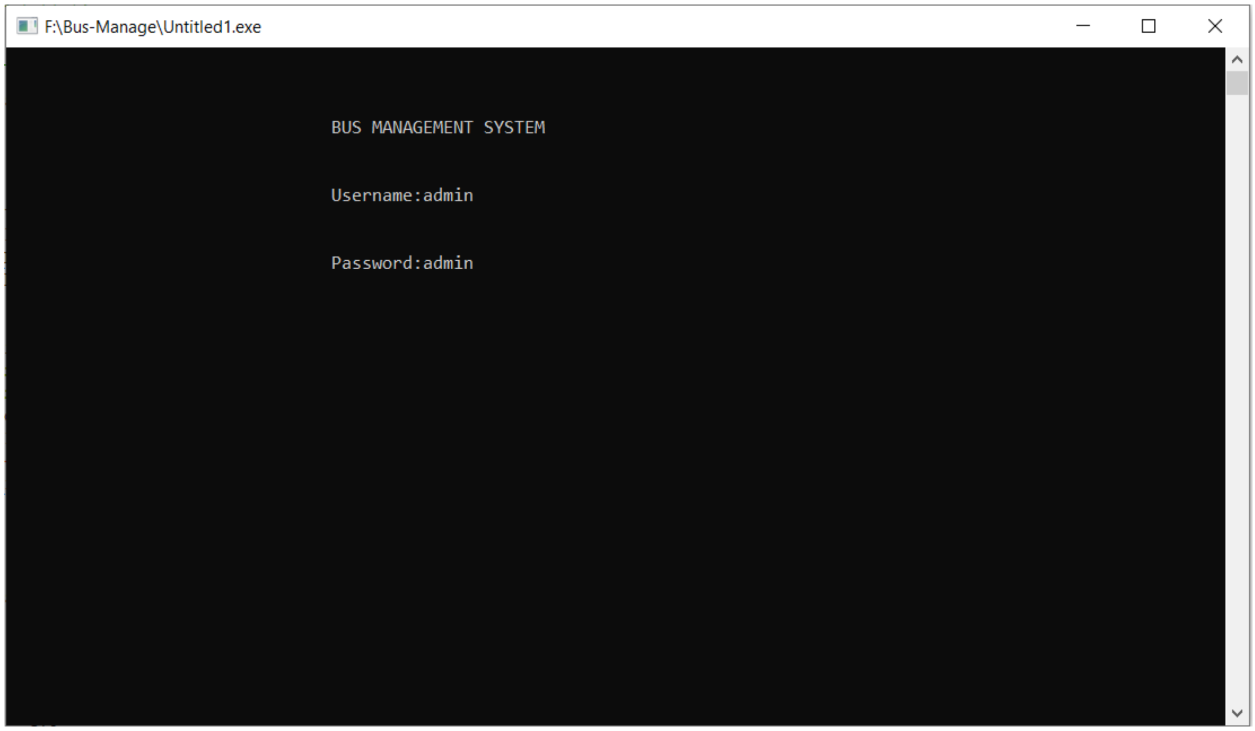
```cpp
int main()
{
    bus_saved(); ///Store bus information from file
    Intro();    ///Title print
    Save();    ///Read username & password from file
    Log_In(); ///Log in

    while(1)
    {
        system("cls");

        int press;
        Intro();

        cout<<endl<<endl;
        cout<<"\t\t\t\t1. Book Tickets"<<endl;
        cout<<"\t\t\t\t2. Cancel Tickets"<<endl;
        cout<<"\t\t\t\t3. Available Bus Operators"<<endl;
        cout<<"\t\t\t\t4. Install Bus Information"<<endl;
        cout<<"\t\t\t\t5. Exit"<<endl;

        cout<<"\n\t\t\t\tEnter option:---> ";
        cin>>press;

        switch(press)
        {
        case 1:

        {
            Book_Tickets( );

        }
        break;
        case 2:

        {
            Cancel_Tickets();

        }
        break;
        case 3:

```



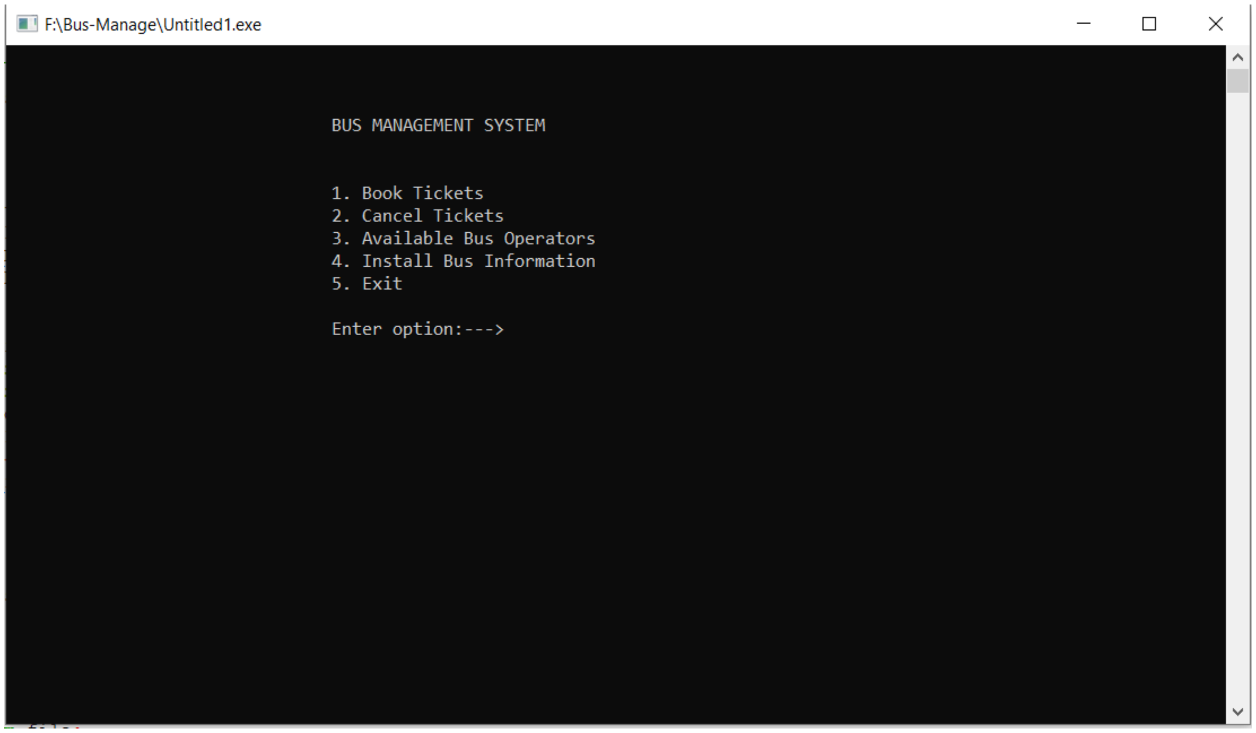
```
        }
        break;
        case 2:

        {
            Cancel_Tickets();

        }
        break;
        case 3:

        {
            Available_Bus_Routes();
        }
        break;
        case 4:

        {
            Install_Bus_Information();
        }
        break;
        case 5:

        {
            system("cls");
            exit(0);
        }
        break;
        default:
        {
            printf("Wrong Input.. END!!\n");

        }

        }
    }


}

```

6. Validation, verification and Testing:

➢ Validation, verification :

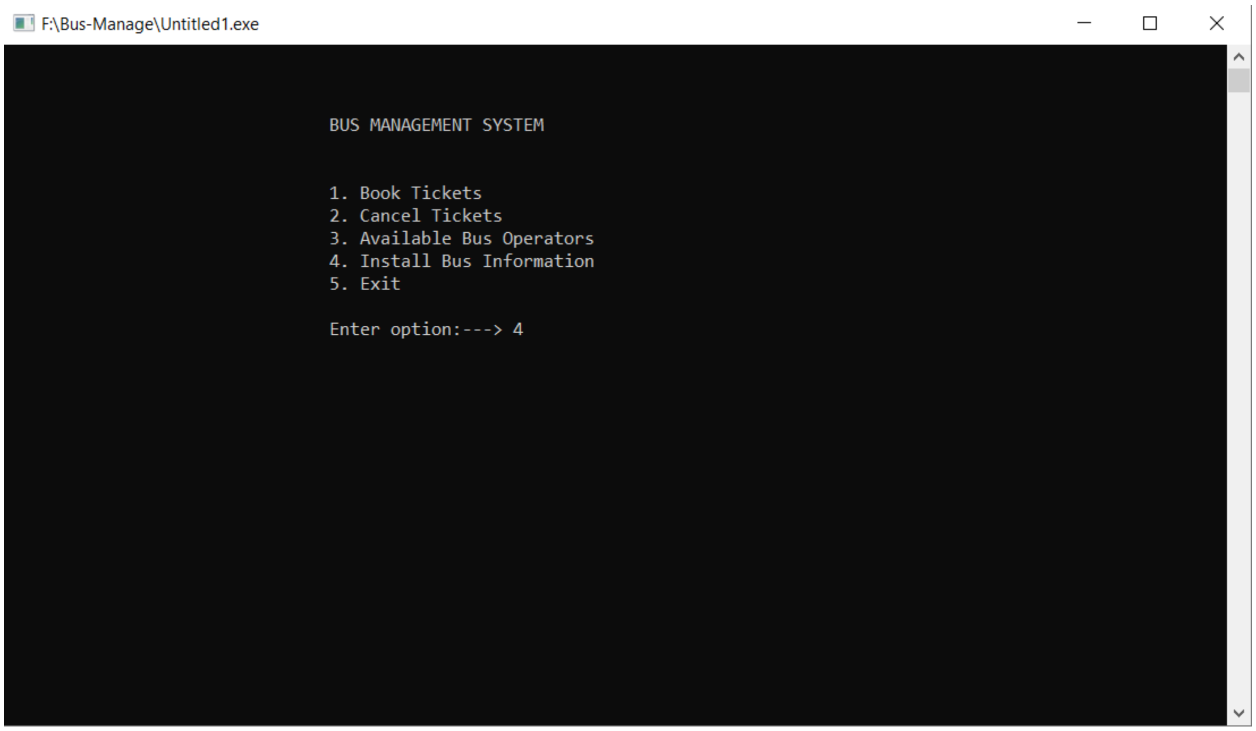

F:\Bus-Manage\Untitled1.exe

```
BUS MANAGEMENT SYSTEM

Username:admin

Password:admin
```



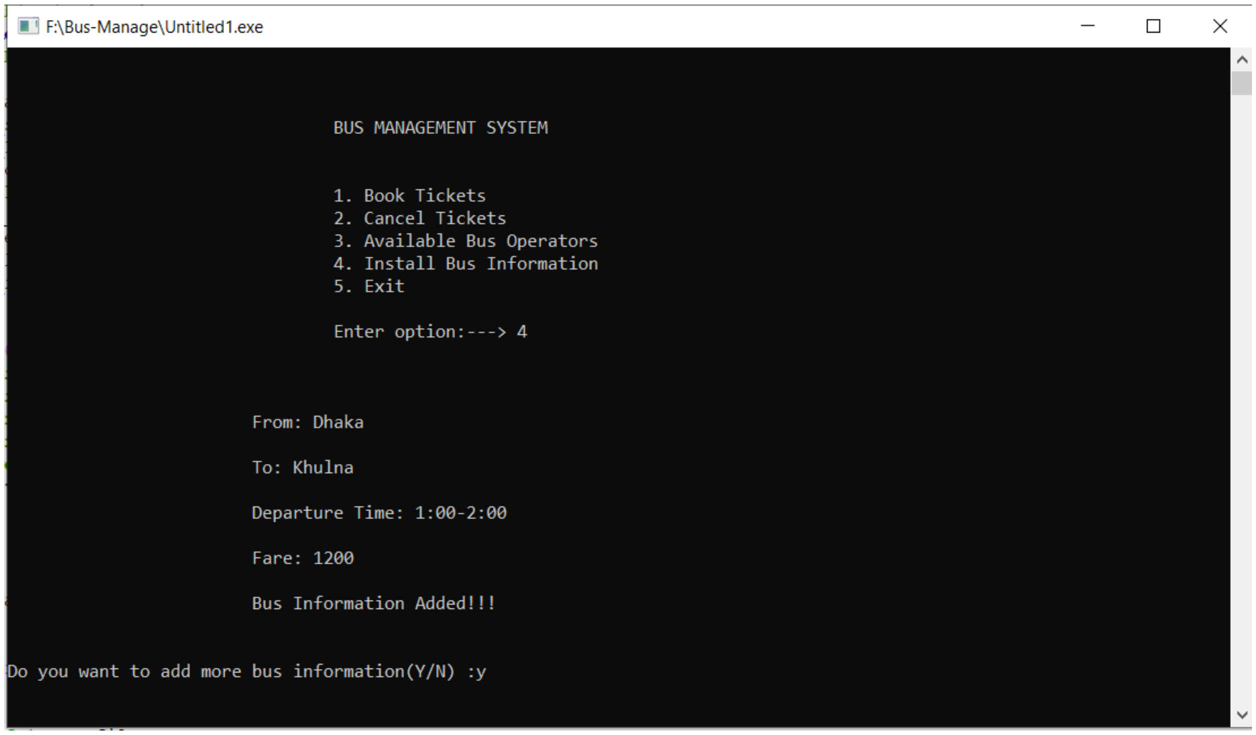
F:\Bus-Manage\Untitled1.exe

```
BUS MANAGEMENT SYSTEM


1. Book Tickets
2. Cancel Tickets
3. Available Bus Operators
4. Install Bus Information
5. Exit

Enter option:--->
```

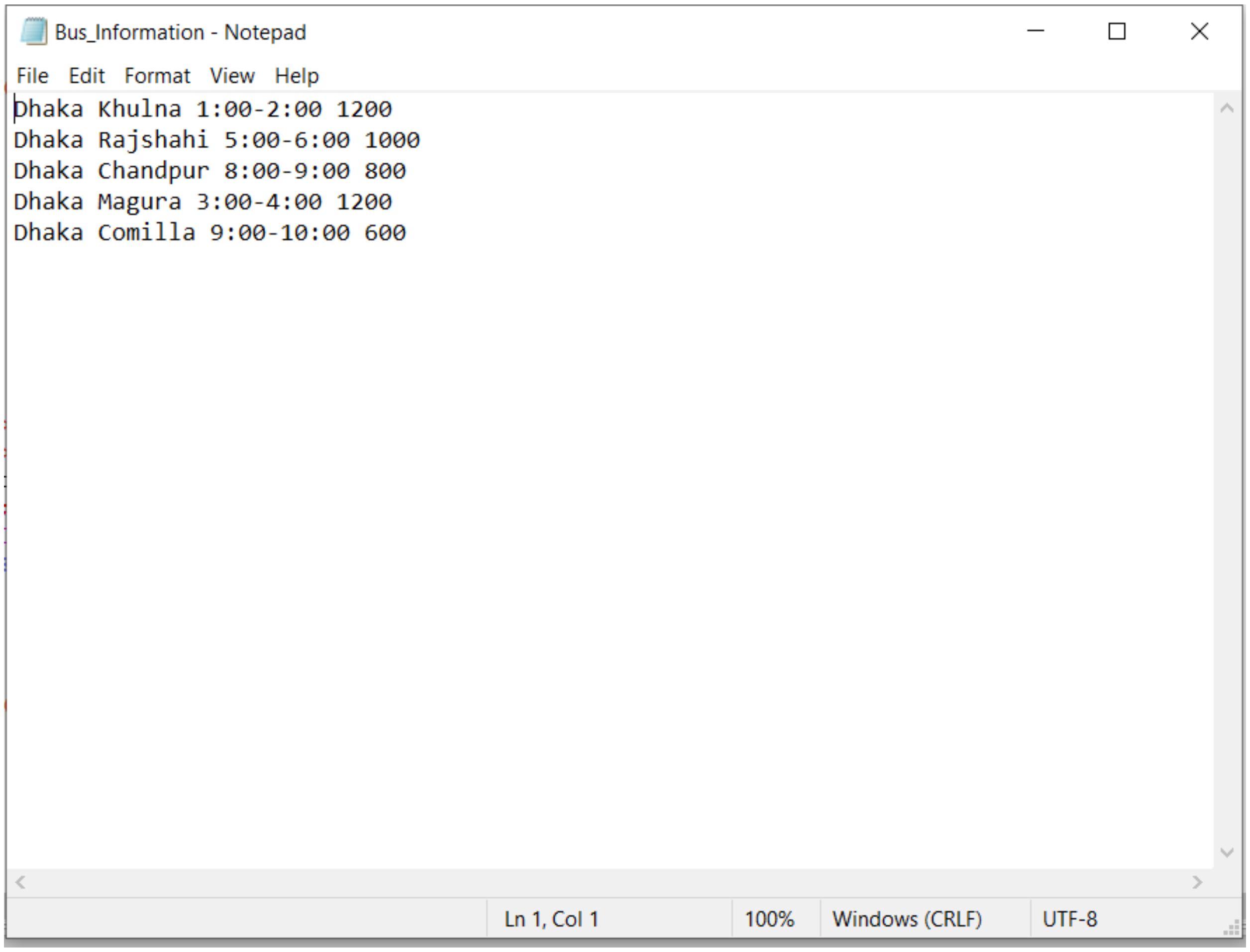


F:\Bus-Manage\Untitled1.exe

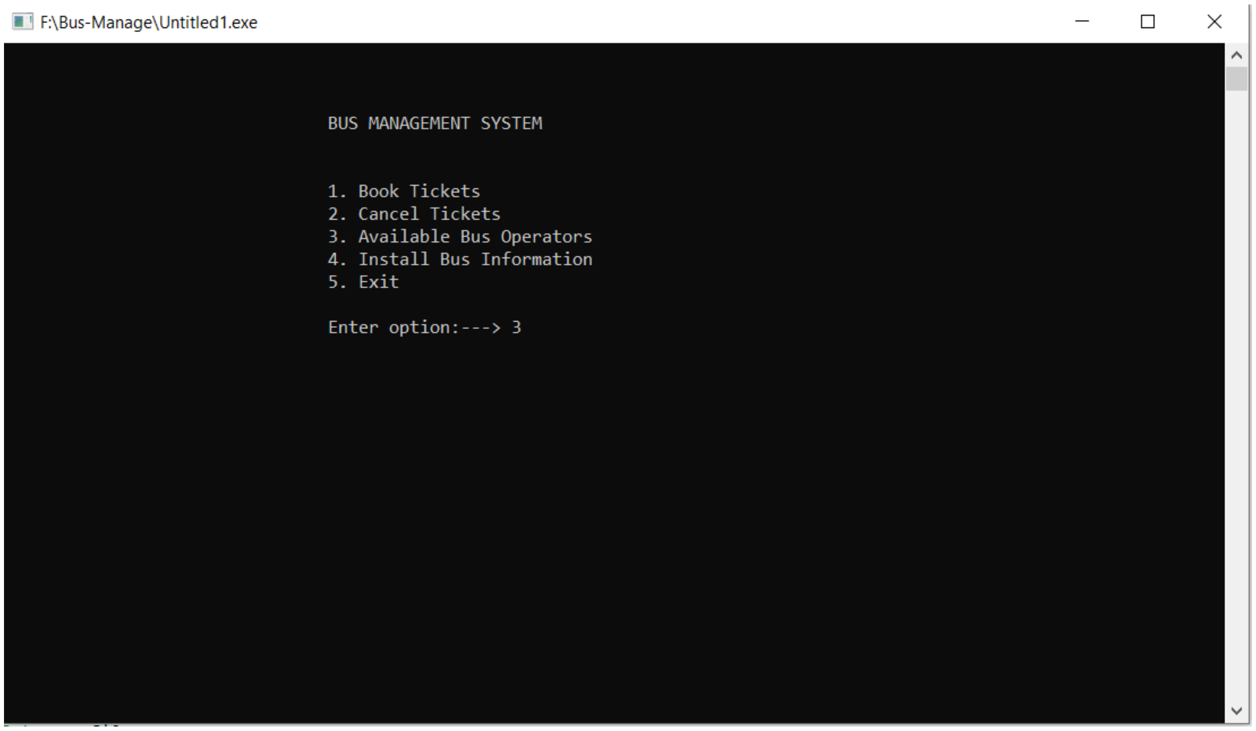
```
BUS MANAGEMENT SYSTEM


1. Book Tickets
2. Cancel Tickets
3. Available Bus Operators
4. Install Bus Information
5. Exit

Enter option:---> 4
```



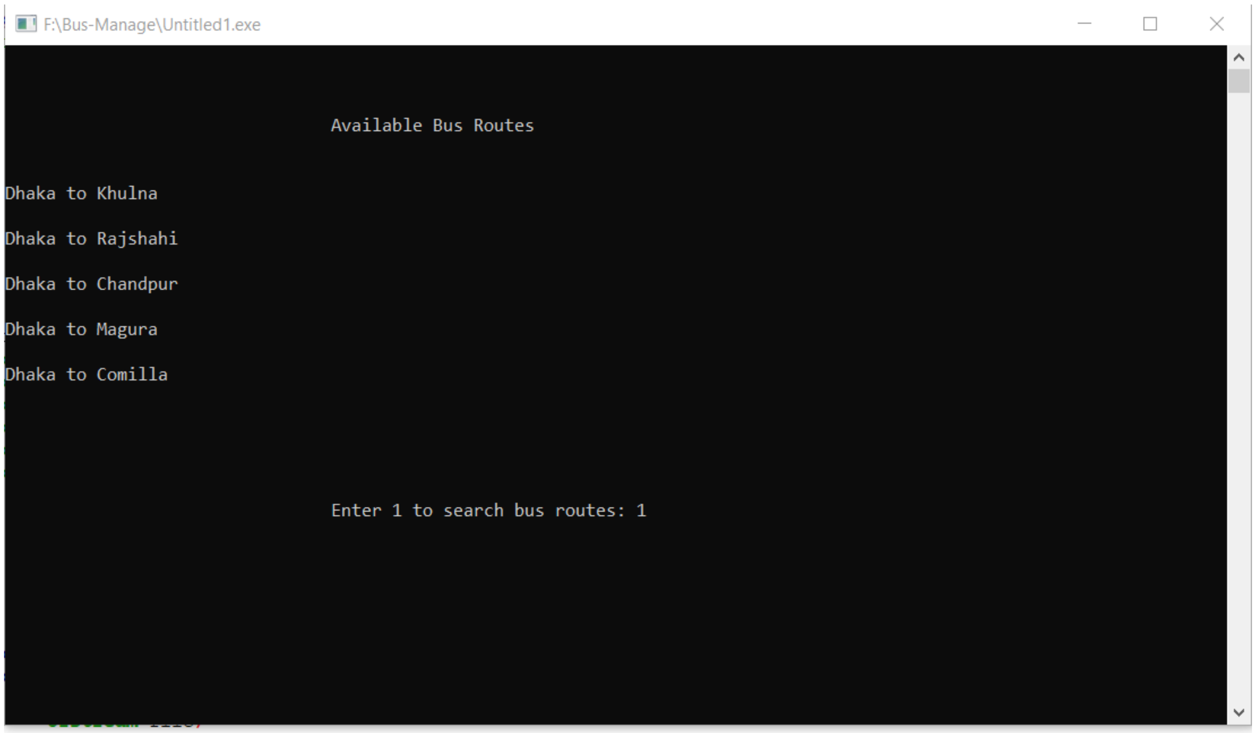
```
                                BUS MANAGEMENT SYSTEM


                                1. Book Tickets
                                2. Cancel Tickets
                                3. Available Bus Operators
                                4. Install Bus Information
                                5. Exit

                                Enter option:---> 4



                        From: Dhaka

                        To: Khulna

                        Departure Time: 1:00-2:00

                        Fare: 1200

                        Bus Information Added!!!


Do you want to add more bus information(Y/N) :y
```



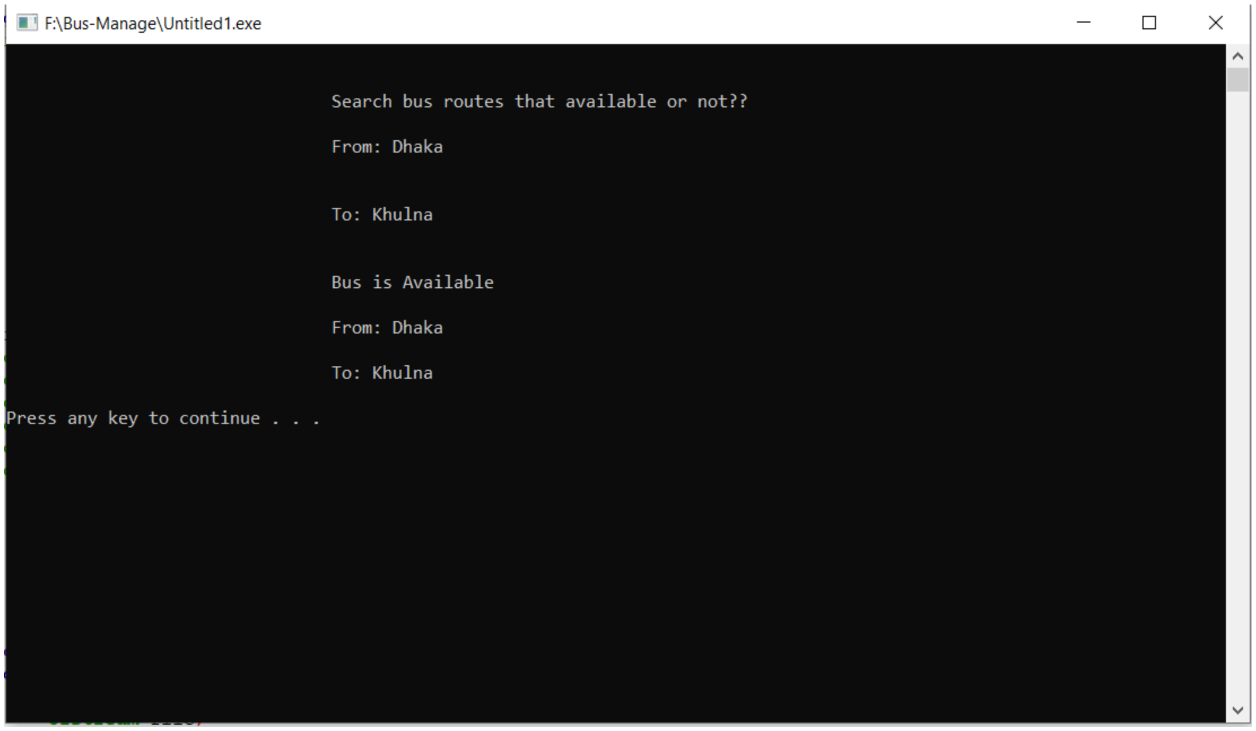
```
Dhaka Khulna 1:00-2:00 1200
Dhaka Rajshahi 5:00-6:00 1000
Dhaka Chandpur 8:00-9:00 800
Dhaka Magura 3:00-4:00 1200
Dhaka Comilla 9:00-10:00 600
```



```
F:\Bus-Manage\Untitled1.exe

                                BUS MANAGEMENT SYSTEM


                                1. Book Tickets
                                2. Cancel Tickets
                                3. Available Bus Operators
                                4. Install Bus Information
                                5. Exit

                                Enter option:---> 3
```

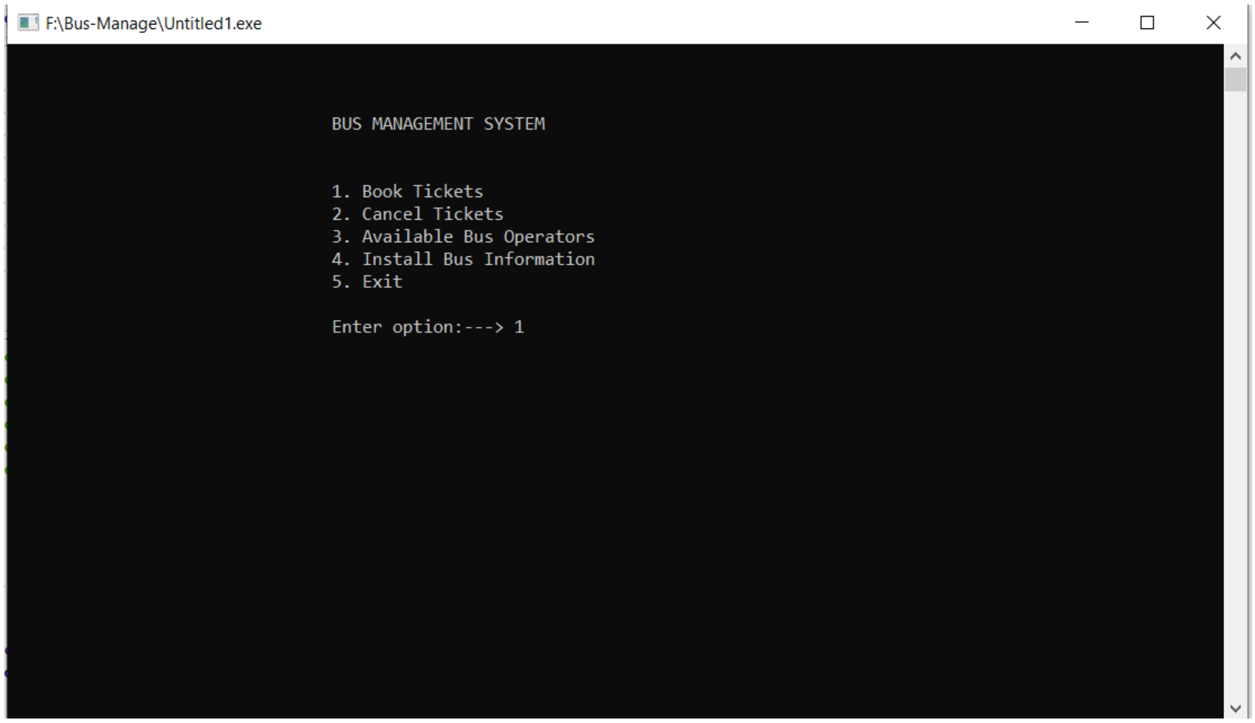


```
F:\Bus-Manage\Untitled1.exe

                                Available Bus Routes


Dhaka to Khulna

Dhaka to Rajshahi

Dhaka to Chandpur

Dhaka to Magura

Dhaka to Comilla




                                Enter 1 to search bus routes: 1
```



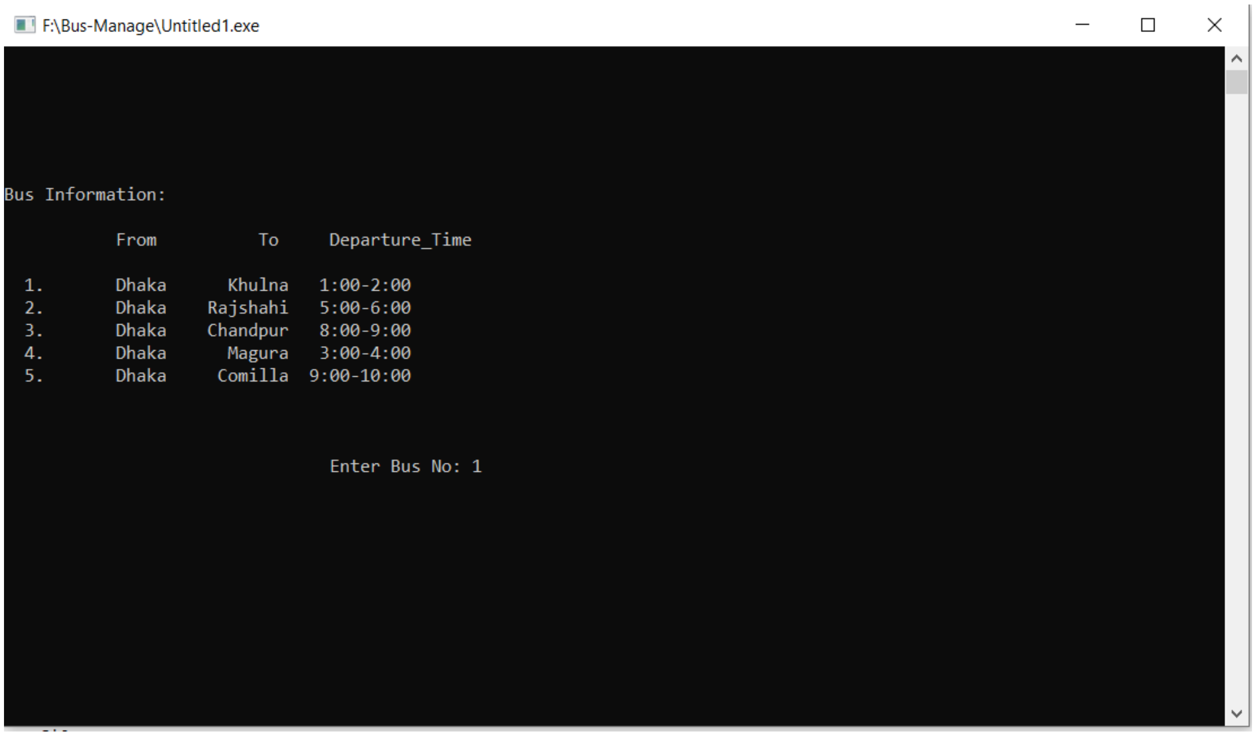
F:\Bus-Manage\Untitled1.exe

```
                                Search bus routes that available or not??

                                From: Dhaka


                                To: Khulna


                                Bus is Available

                                From: Dhaka

                                To: Khulna

Press any key to continue . . .
```



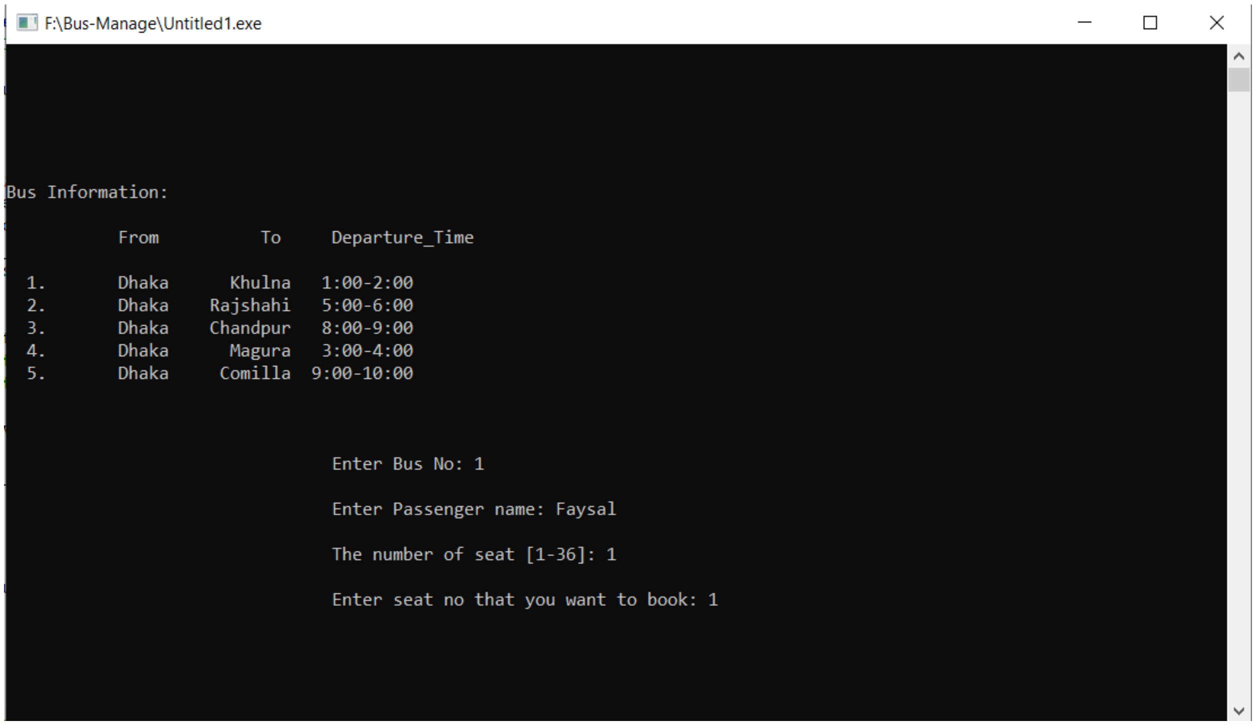
F:\Bus-Manage\Untitled1.exe

```
                                BUS MANAGEMENT SYSTEM


                                1. Book Tickets
                                2. Cancel Tickets
                                3. Available Bus Operators
                                4. Install Bus Information
                                5. Exit

                                Enter option:---> 1
```



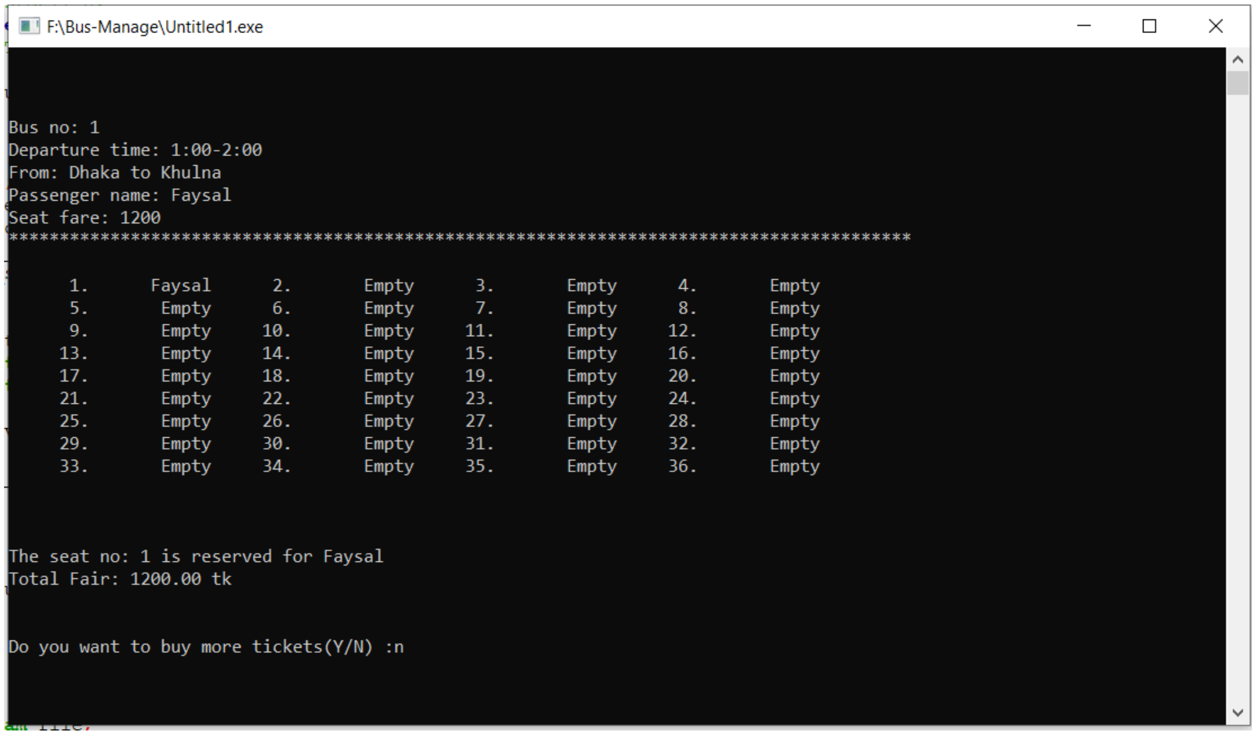
F:\Bus-Manage\Untitled1.exe

```
Bus Information:

           From          To     Departure_Time

  1.       Dhaka      Khulna   1:00-2:00
  2.       Dhaka    Rajshahi   5:00-6:00
  3.       Dhaka    Chandpur   8:00-9:00
  4.       Dhaka      Magura   3:00-4:00
  5.       Dhaka     Comilla  9:00-10:00



                                Enter Bus No: 1
```

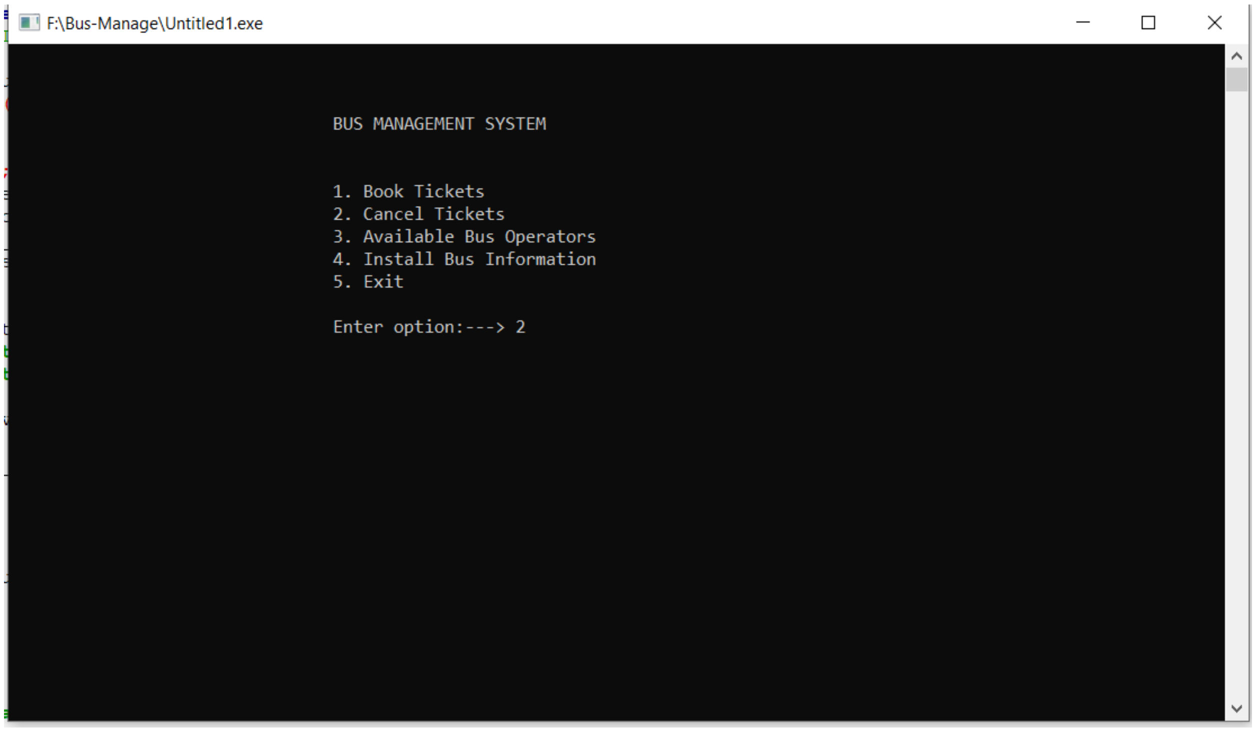


F:\Bus-Manage\Untitled1.exe

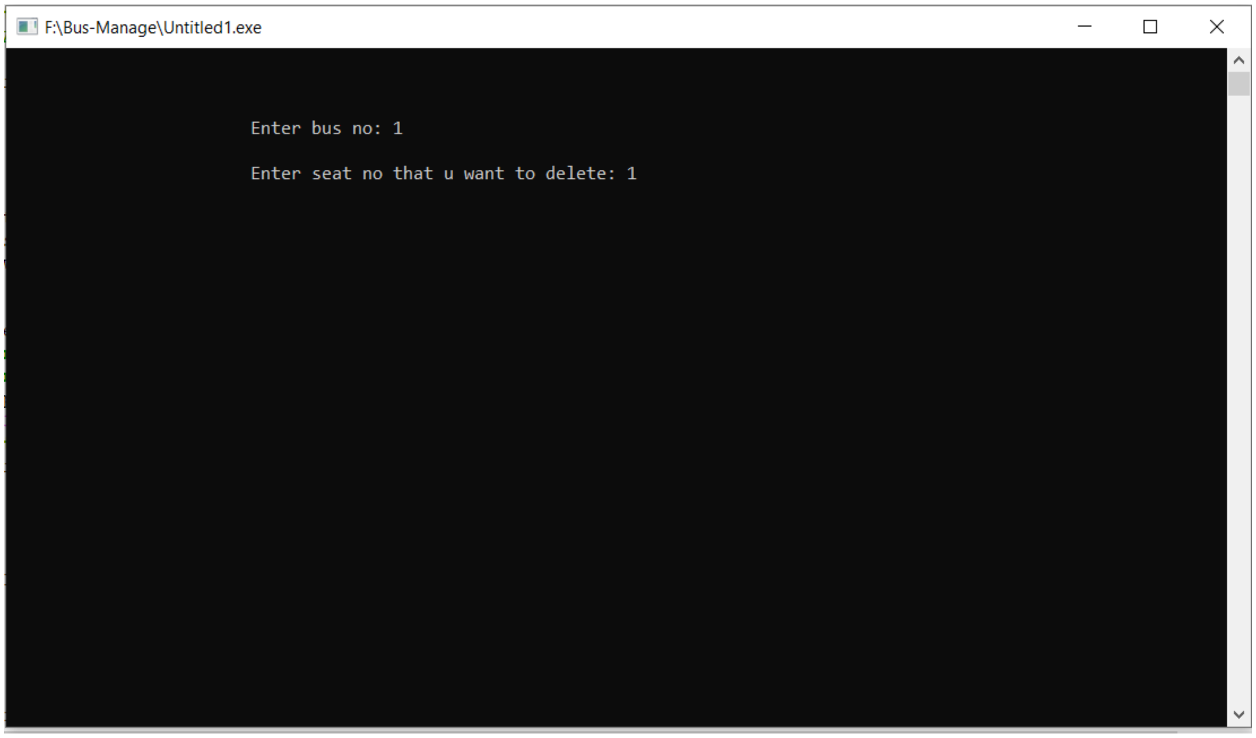
```
Bus Information:

           From          To     Departure_Time

  1.       Dhaka      Khulna   1:00-2:00
  2.       Dhaka    Rajshahi   5:00-6:00
  3.       Dhaka    Chandpur   8:00-9:00
  4.       Dhaka      Magura   3:00-4:00
  5.       Dhaka     Comilla  9:00-10:00



                                Enter Bus No: 1

                                Enter Passenger name: Faysal

                                The number of seat [1-36]: 1

                                Enter seat no that you want to book: 1
```


F:\Bus-Manage\Untitled1.exe

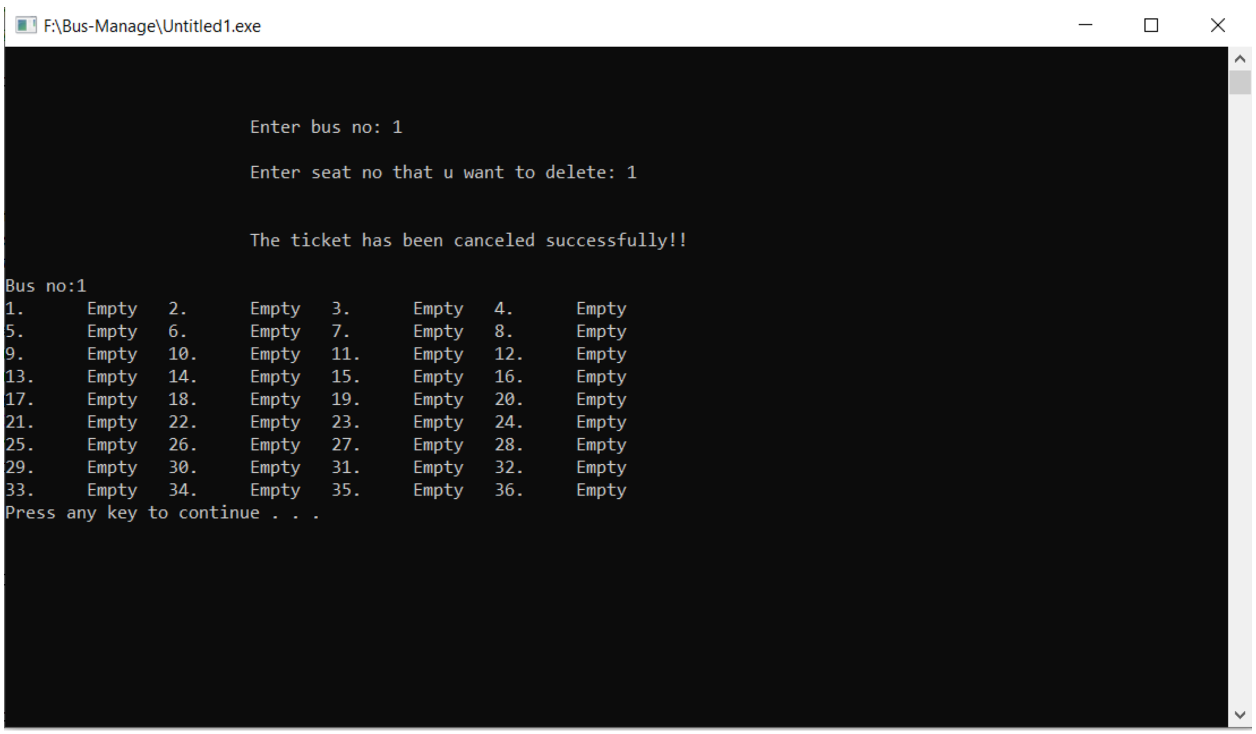
```
Bus no: 1
Departure time: 1:00-2:00
From: Dhaka to Khulna
Passenger name: Faysal
Seat fare: 1200
*****************************************************************************************

      1.      Faysal      2.       Empty      3.       Empty      4.       Empty
      5.       Empty      6.       Empty      7.       Empty      8.       Empty
      9.       Empty     10.       Empty     11.       Empty     12.       Empty
     13.       Empty     14.       Empty     15.       Empty     16.       Empty
     17.       Empty     18.       Empty     19.       Empty     20.       Empty
     21.       Empty     22.       Empty     23.       Empty     24.       Empty
     25.       Empty     26.       Empty     27.       Empty     28.       Empty
     29.       Empty     30.       Empty     31.       Empty     32.       Empty
     33.       Empty     34.       Empty     35.       Empty     36.       Empty


The seat no: 1 is reserved for Faysal
Total Fair: 1200.00 tk


Do you want to buy more tickets(Y/N) :n
```


F:\Bus-Manage\Untitled1.exe

```
                                BUS MANAGEMENT SYSTEM


                                1. Book Tickets
                                2. Cancel Tickets
                                3. Available Bus Operators
                                4. Install Bus Information
                                5. Exit

                                Enter option:---> 2
```


F:\Bus-Manage\Untitled1.exe

```
                        Enter bus no: 1

                        Enter seat no that u want to delete: 1
```


F:\Bus-Manage\Untitled1.exe

```
                        Enter bus no: 1

                        Enter seat no that u want to delete: 1


                        The ticket has been canceled successfully!!

Bus no:1
1.      Empty   2.      Empty   3.      Empty   4.      Empty
5.      Empty   6.      Empty   7.      Empty   8.      Empty
9.      Empty   10.     Empty   11.     Empty   12.     Empty
13.     Empty   14.     Empty   15.     Empty   16.     Empty
17.     Empty   18.     Empty   19.     Empty   20.     Empty
21.     Empty   22.     Empty   23.     Empty   24.     Empty
25.     Empty   26.     Empty   27.     Empty   28.     Empty
29.     Empty   30.     Empty   31.     Empty   32.     Empty
33.     Empty   34.     Empty   35.     Empty   36.     Empty
Press any key to continue . . .
```

- **Testing:**

Here we are doing testing to check software adaptability, to avoid risks and to identify errors. We know there are four types of testing but for our software, we are doing unit testing.

Reason behind choosing unit testing:

Basically, our software has four different functions and here we want to test them individually. So, unit testing will be perfect for that.

Unit testing

There are two kinds of unit testing but for our software we will choose white box testing.

White box testing allows to find the hidden errors, to find internal errors because it checks and works by internal functionality.

It helps to find issues and optimize code to adopt different techniques of White Box Testing to test a developed application or website.

It requires internal knowledge to do testing that's why it helps in maximum coverage of the code. White Box test requires programming knowledge.

Log In page testing: In our log in page we will enter the username and password. If both are correct, the dashboard page will open. Otherwise, if any data is incorrect, it will be redirected to the login page. And again, ask for the username and password.

So basically, if we enter wrong password or username or both wrong and then if our software doesn't show any errors, that's mean our testing is failed but if it shows error that's mean testing is successful.

7. Maintenance or Help Guide

- **Maintenance:** Basically, Software maintenance is part of the software development life cycle. The purpose of the service is to modify and continuously update software applications to eliminate all possible errors, malfunctions, to improve work efficiency and better system.
- **Corrective maintenance**: It is essential either to rectify some bugs observed while the system is in use.
  - **Seat Issue**: In our system seat issue is creating bugs. When two customers with the same name using book tickets, some problems arise. Again, if one of them cancels the ticket, it may confuse the other one and also cancel the other's one ticket.
- **Adaptive maintenance**: This includes modifications and updations when the customers need the product to run on new platforms.
  - **Running this project on flutter:** We are trying to make our software android based. So, our next target is to run this software on flutter.
- **Perfective maintenance:** It supports new features that the users want or to change different types of functionalities of the system according to the customer demands. So we are using two new features.
  - **Forgot Password:** Our software still doesn't have this "Forgot Password" feature. As some customers can forget their password, this feature is actually an important feature for a software. So, this new feature will be added.
  - **Online Payment:** Now everything has been digitalized and many people feel very comfortable with online payment. And online payment also saves times and energy. So our next target is to build this feature.
- **Preventive maintenance:** This type of maintenance includes modifications and updations to prevent future problems of the software.
  - **Update seat Issue:** So basically, as there is an issue about seat, we will update this feature.

➢ **Help Guide:**

We want you to have an easy and safe journey and you can get that experience by using our software.

For you to know how to use our software, we are giving the guidelines you need to know.

- Login to dashboard
- From navigation bar press "3" to find Available Bus Operators.
- Press "1" for Book Tickets & seat reservation
- Press "2" for Cancel Tickets