ABSTRACT

This project is aimed at developing an online “Bus Management System” for a transportation company. This system can be used to store the details of the various buses, book the tickets by using authorized user ID, generate the ticket fare, check the availability of tickets of a particular bus etc.

The main idea of this application is to computerize the ticket booking functionality and database for bus Transportation Company with huge number of buses. This software will help to manage the information of the database of the bus and its passengers.

The admin will be able to create end user ID, delete end user ID, and add buses to the database. A normal end user can book tickets of a particular bus, cancel tickets of a particular bus, and even view details of any bus they need by entering the bus number. This bus transportation system provides a facility to reserve seats, cancel seats and different types of enquiry which needs an instant and quick reservation. It allows counter employee to check bus ticket availability and make reservation on particular customer.

It helps the passengers to book their tickets at home without moving to the ticket booking office. It also helps the company management to store the database of various buses and the passenger details in a simplified and efficient manner.

The easiest bus booking software to create routes with start, end, manage time traveling and bus schedules. Admin can decide the ticket cost for all the buses. A standard user can book their seat by using a seat map. A new standard user can be authorized or even unauthorized very easily. This is one integrated system that contains both the user component and the admin component used by the administrators for performing admin level functions as well as standard user functions.

MODULE DESCRIPTION

In this Bus Reservation System program, class and functions has been used to display the menu and manipulate the data of the bus and also file handling is also performed to store the user ID and bus data. Here I have used class ID,reserve and functions such as void get_username (); void reserve::display (); void admin_login();.

char ID::get_username();

In this function the input part is coded. This function is used to get the username of a standard user ID when creating or deleting a standard user ID.

char ID::get_password();

In this function the input part is coded. This function is used to get the password of a standard user ID when creating a deleting a standard user ID.

int reserve::check_empty();

This is a member function of class reserve. This function is used to check whether a particular seat of a particular bus is empty or not.

void reserve::display();

This is a member function of class reserve. This function is used to display the details of a particular bus.

void reserve::disp_seat();

This is a member function of class reserve. This function is used to display the seat map of a particular bus.

int reserve::check_empticket();

This is a member function of class reserve. This function is used to check the total number of empty tickets in a given bus.

void vline(char);

In this function the output part is coded. This function is used to display a character repeatedly in a given line.

void loading(void);

In this function the output part is coded. This function is used to delay the next output using the loading screen.

void main_menu(void);

In this function both output and input part is coded. This function is used to display the main menu and select the options to perform the required operations.

void start_screen(void);

In this function the output part is coded. This function is used to display the entry screen while running the program.

void admin_login(void);

In this function both the input and output part is coded. This function is used to get the login details for the admin and interpret the login details and let the user to be authorized.

void admin_mode(void);

In this function both the input and output part is coded. This function displays all the operations that could be performed by an admin and the admin the choose the operations that he/she wants to do.

void create_id(void);

In this function both the input and output part is coded. This function is used to create a standard user ID which authorizes a standard end user to use the application.

void delete_id(void);

In this function both the input and output part is coded. This function is used to delete a standard user ID which remove the authentication of a standard end user to use the application.

void install_bus(void);

In this function both the input and output part is coded. This function is used to add the details of a particular bus to the file.

void standard_login(void);

In this function both the input and output part is coded. This function is used to authenticate the end user for using the application.

void standard_mode(void);

In this function both the input and output part is coded. This function is used to display all the possible operations that a standard end user can perform. He/She can select the operation to be performed with the help of this function.

void book_tickets(void);

In this function both the input and output part is coded. This function is used to book tickets in a particular with the help of bus number and the user ID.

void cancel_ticket(void);

In this function both the input and output part is coded. This function is used to cancel tickets in a particular with the help of bus number and the user ID.

void show_bus(void);

In this function both the input and output part is coded. This function is used to display the details of a particular bus with the help of the bus number.

void show_seats(reserve);

In this function both the input and output part is coded. This function is used to display the details of the seats booked and seats available in a particular bus by using the bus number.

System Supplied Functions:

Getch();

This function is used to read a character from screen.

Exit();

This function is used to terminate program execution and to return to the operating system.

Gets();

This function reads characters from stdin and stores them in string until a newline character or end of file is found.

Strcmp();

This function is used to compare two strings and return 0 only if both the strings are same.

Header Files:

Iostream.h:

Iostream provides basic input and output services for C++ programs. Iostream uses the objects cin, cout, cerr, and clog for sending data to and from the standard streams input, output, error, and log respectively.

Fstream.h:

Fstream object may be used to open a file for writing. And ifstream object is used to open a file for reading purpose only. Following is the standard syntax for open() function, which is a member of fstream, ifstream, and ofstream objects.

Process.h:

Process.h is a header file which contains function declarations and macros used in working with threads and processes.

String.h:

This header files Contains function prototypes for the C-style standard input/output library functions and information used by them.

Stdio.h:

This library uses what are called streams to operate with physical devices such as keyboards, printers, terminals or with any other type of files supported by the system.

EXISTING AND PROPOSED SYSTEM

Existing System:

Existing system allow customers to book tickets only for a single bus. Tickets can be booked using a seat map. All the database would be erased once the user exit from the program. This program has no separation of admin and user mode. The database about the booked tickets are not tracked by the existing system. This program doesn't provide get any information about the person who booked the ticket. So someone may misuse the booked ticket which may spoil the reputation of the company. Since the install bus option is available to everyone, anyone may create a new bus and others may book tickets for that officially unavailable bus which may lead to the huge loss for the company. A booked ticket can be cancelled anyone in this system. So the company may face a huge loss if someone cancels the ticket other than the person who booked it.

Proposed System:

The current system will allow the users to know about their payment details and about their due payment after the reservation. In the current system, the users are authorized by a user ID by the administrator. In the current system, the tickets booked are tracked by a file. The database won't be erased once the user exit from the program. The current system provides the detailed information about the user who booked ticket which makes easier to access the information of the customer. In this system, install bus option is not accessible to everyone. Only the user who booked the ticket can cancel their ticket.

SYSTEM SPECIFICATION

Hardware requirements:

Processor : Pentium(R) Dual core CPUE5400 @2.70GHz

RAM : 2.00GB

Bit : 32-bit operating system

Hard disk : 900MB space required

Software requirements:

Software : Turbo C++ 3.8.0.1

Operating System : Windows 7

ABOUT C++

C++ is general-purpose programming language. It has imperative, objectoriented and generic programming features, while also providing facilities for low-level memory manipulation. It was designed with a bias toward system programming and embedded, resource-constrained and large systems, with performance, efficiency and flexibility of use as its design highlights.

Before the initial standardization in 1998, C++ was developed by Bjarne Stroustrup at Bell Labs since 1979, as an extension of the C language as he wanted an efficient and flexible language similar to C, which also provided high-level features for program organization. In 1979, Bjarne Stroustrup, a Danish computer scientist, began work on "C with Classes", the predecessor to C++.The motivation for creating a new language originated from Stroustrup experience in programming for his Ph.D. thesis.

Stroustrup found that Simula had features that were very helpful for large software development, but the language was too slow for practical use, while BCPL was fast but too low-level to be suitable for large software development. When Stroustrup started working in AT&T Bell Labs, he had the problem of analyzing the UNIX kernel with respect to distributed computing. Remembering his Ph.D. experience, Stroustrup set out to enhance the C language with Simulalike features.

C was chosen because it was general-purpose, fast, portable and widely used. As well as C and Simula's influences, other languages also influenced C++, In 1983, "C with Classes" was renamed to "C++" ("++" being the increment operator in C), adding new features that included virtual functions, function name and operator overloading, references, constants, type-safe free-store memory allocation (new/delete), improved type checking, and BCPL style single-line comments with two forward slashes (//).

Furthermore, it included the development of a standalone compiler for C++, Cfront. In 1985, the first edition of The C++ Programming Language was released, which became the definitive reference for the language, as there was not yet an official standard. The first commercial implementation of C++ was released in October of the same year.

In 1989, C++ 2.0 was released, followed by the updated second edition of The C++ Programming Language in 1991. New features in 2.0 included multiple inheritance, abstract classes, static member functions, const member functions, and protected members.

In 1990, The Annotated C++ Reference Manual was published. Ad ding numerous new features, enlarging the standard library further, and providing more facilities to C++ programmers. After a minor C++14 update released in December 2014, various new additions are planned for 2017 and 2020. The C++ language has two main components: a direct mapping of hardware features provided primarily by the C subset and zero-overhead abstractions based on those mappings.

Stroustrup describes C++ as "a light-weight abstraction programming language [designed] for building and using efficient and elegant abstractions"; and "offering both hardware access and abstraction is the basis of C++. Doing it efficiently is what distinguishes it from other languages" C++ is often considered to be a superset of C, but this is not strictly true. Most C code can easily be made to compile correctly in C++, but there are a few differences that cause some valid C code to be invalid or behave differently in C++.

SOURCE CODE

```
#include<iostream.h>
#include<conio.h>
#include<stdio.h>
#include<string.h>
#include<process.h>
#include<fstream.h>


void vline(char);//Function Prototype

class ID //Class for USER ID
{
 public:
 char user_name[20];
 char password[20];

 char get_username() //Member function to get username
 {
  cout<<"\n\n\t Enter the user name of the ID: ";
  gets(user_name);
  return user_name[20];
 }

 char get_password() //Member function to get password
 {
  cout<<"\n\n\t Enter the password of the ID: ";
  gets(password);
  return password[20];
```

```
 }

}login_id;


class reserve   //Class for the details of the bus
{
  public:
  int busn;
  char driver[20];
  char arrival[5];
  char depart[5];
  char from[20];
  char to[20];
  char seat[8][4][10];
  char user_ID[8][4][20];
  int cost;

  void display();  //Member function to display bus details
  void disp_seat();  //Member function to display seat details
  int check_empticket();  //Member function to check number of empty tickets
in bus
int check_empty(int i,int j) //Member function to check whether the seat is
empty
 {
   if(strcmp(seat[i][j],"Empty")==0)
   {
    return 1;
   }
```

```
   else
   {
    return 0;
   }
  }
}bus[50];

void reserve::display()
  {
   clrscr();
   vline('*');
   cout<<"\n\n\t\t\t     The details of the bus";
   cout<<"\n\n";
   vline('*');
   cout<<"\n\n\n\tBus number: ";
   cout<<busn;
   cout<<"\n\n\tDriver's name: ";
   cout<<driver;
   cout<<"\n\n\tArrival time: ";
   for(int a1=0;a1<5;a1++)
   {
    cout<<arrival[a1];
   }
   cout<<"\n\n\tDeparture time: ";
   for(a1=0;a1<5;a1++)
   {
    cout<<depart[a1];
   }
   cout<<"\n\n\tOrigin: ";
```

```
  cout<<from;
  cout<<"\n\n\tDestination: ";
  cout<<to;
  cout<<"\n\n\tCost: ";
  cout<<cost;
  cout<<"\n\n\n\n\t\t   Press any key to continue to seat map";
  getch();
 }

void reserve::disp_seat()
 {
  int s=0,p=0;
  for(int i=0;i<8;i++)
  {
   cout<<"\n";
   for(int j=0;j<4;j++)
   {
    if(j==2)
    {
     cout<<"\t";
    }
    s++;
    if(strcmp(seat[i][j],"Empty")==0)
    {
     cout.width(5);
     cout.fill(' ');
     cout<<s<<".";
     cout.width(10);
     cout.fill(' ');
```

```
    cout<<seat[i][j];
    p++;
    }
    else
    {
     cout.width(5);
     cout.fill(' ');
     cout<<s<<".";
     cout.width(10);
     cout.fill(' ');
     cout<<seat[i][j];
     }
    }
   }
  cout<<"\n\n\t\t     There are "<<p<<" seats empty in Bus No: "<<busn;
 }

int reserve::check_empticket()
 {
  int e=0;
  for(int i=0;i<8;i++)
  {
   for(int j=0;j<4;j++)
   {
    if(strcmp(seat[i][j],"Empty")==0)
    e++;
   }
  }
  return e;
```

```
 }

//Function prototypes

void main_menu(void);
void admin_login(void);
void admin_mode(void);
void create_id(void);
void delete_id(void);
void standard_login(void);
void standard_mode(void);
void book_tickets(void);
void install_bus(void);
void show_bus(void);
void show_seats(reserve);
void cancel_ticket(void);


void vline(char ch) //Function to insert line
{
  for(int i=1;i<80;i++)
  cout<<ch;
}


void start_screen() //Start Screen Function
{
 clrscr();
 cout<<"\n\n";
```

```
 vline('*');
 cout<<"\n\n\t\t\t     Welcome to VP Transports";
 cout<<"\n\n";
 vline('*');
 cout<<"\n\n\n\n\n\n\n\n\t\t       Only Tatkal Tickets can be booked here" ;
 cout<<"\n\n\n\n\n\n\n\t\t\t     Press any key to continue";
 getch();
}


void main_menu()  //Main menu function
{
 clrscr();
 int choice1;
 cout<<"\n\n";
 vline('*');
 cout<<"\n\n\t\t\t\t\t    MAIN MENU";
 cout<<"\n\n";
 vline('*');
 cout<<"\n\n\t\t\t      Choose the login mode";
 cout<<"\n\n\t\t\t      1.ADMIN Mode";
 cout<<"\n\n\t\t\t      2.STANDARD USER Mode";
 cout<<"\n\n\t\t\t      0.Exit";
 cout<<"\n\n\n\t   Enter your choice (0 ,1 or 2): ";
 cin>>choice1;
 switch(choice1)
 {
   case 1:admin_login();
  break;
```

```
   case 2:standard_login();
  break;
   case 0:cout<<"\n\n\n\n\t\t\t     Press any key to exit";
  getch();
  exit(0);
  break;
   default:cout<<"\n\n\t\t\t     Enter a valid choice";
  cout<<"\n\n\t\t\t   Press any key to continue";
  getch();
  main_menu();
  getch();
  break;
  }
}


void admin_login() //Function for admin login page
{
 clrscr();
 char admin_name[8],admin_pass[10],try_choice;
 cout<<"\n\n";
 vline('*');
 cout<<"\n\n\t\t\t\t  ADMIN Login";
 cout<<"\n\n";
 vline('*');
 cout<<"\n\n\n\t\t Enter the admin name: ";
 gets(admin_name);
 cout<<"\n\n\t\t   Enter the password: ";
 gets(admin_pass);
```

```
 if(strcmp("Pradesh",admin_name)==0 &&
strcmp("123456789",admin_pass)==0)
 {
  cout<<"\n\n\n\n\n\t\t\t       Login Successful";
  cout<<"\n\n\t\t\t   Press any key to continue";
  getch();
  admin_mode();
 }
else
 {
  cout<<"\n\n\n\n\t\t     Either your name or password is incorrect";
  label1:
  cout<<"\n\n\t\t\t Do you want to try again(y/n):";
  cin>>try_choice;
  switch(try_choice)
  {
   case 'y':admin_login();
    break;
   case 'n':cout<<"\n\n\t\t   Press any key to return back to main menu";
    getch();
    main_menu();
    break;
   case 'Y':admin_login();
    break;
   case 'N':main_menu();
    break;
   default :cout<<"\n\n\t\t Enter a valid input";
     goto label1;
  }
```

```
 }
 getch();
}

void admin_mode() //Admin Page
{
 clrscr();
 cout<<"\n\n";
 vline('*');
 cout<<"\n\n\t\t\t\t   ADMIN MENU";
 cout<<"\n\n";
 vline('*');
 cout<<"\n\n\t 1.To create a standard user ID.";
 cout<<"\n\n\t 2.To delete a existing standard user ID.";
 cout<<"\n\n\t 3.Add a bus to the database.";
 cout<<"\n\n\t 4.Return to main menu.";
 cout<<"\n\n\t 0.Exit.";
 cout<<"\n\n\t Enter your choice: ";
 int choice2;
 cin>>choice2;
 switch(choice2)
 {
  case 1:create_id();
 break;
  case 2:delete_id();
 break;
  case 3:install_bus();
 break;
  case 4:cout<<"\n\n\t\t   Press any key to return back to main menu";
```

```
 getch();
 main_menu();
 break;
  case 0:cout<<"\n\n\t\t\t     Press any key to exit";
  getch();
 exit(0);
 break;
  default:cout<<"\n\n\t\t\t     Enter a valid choice";
  cout<<"\n\n\t\t\t   Press any key to continue";
  getch();
  admin_mode();
  break;
 }
 getch();
}


void create_id() //Function to create a user ID
{
 char choice='n';
 do
 {
  label3:
  fstream login("user.dat",ios::in|ios::app|ios::binary);
  fstream login2("user.dat",ios::in|ios::binary);
  clrscr();
  ID user,confirm,check;
  cout<<"\n\n";
  vline('*');
```

```
cout<<"\n\n\t\t\t  Create a standard user ID";
cout<<"\n\n";
vline('*');
user.get_username();
cout<<"\n";
cout<<"\t          Retype the user name: ";
gets(confirm.user_name);
if(strcmp(user.user_name,confirm.user_name)!=0)
{
 cout<<"\n\n\n\n\n\t\tThe usernames that you have entered doesn't match";
 cout<<"\n\n\n\n\t\t\t   Press any key to try again";
 getch();
 goto label3;
}
while(!login2.eof())
{
 login2.read((char*)&check,sizeof(check));
 if(strcmp(user.user_name,check.user_name)==0)
 {
  cout<<"\n\n\n\n\t\tThe username that you have entered already exists";
  cout<<"\n\n\n\n\t\t\t   Press any key to try again";
  getch();
  goto label3;
 }
}
login2.close();
user.get_password();
cout<<"\n";
cout<<"\t          Retype the password: ";
```

```
 gets(confirm.password);
 if(strcmp(user.password,confirm.password)!=0)
 {
  cout<<"\n\n\n\n\n\t\tThe passwords that you have entered doesn't match";
  cout<<"\n\n\n\n\t\t\t   Press any key to try again";
  getch();
  goto label3;
 }
 login.write((char*)&user,sizeof(user));
 cout<<"\n\n\n\t\t     Your ID has been created successfully";
 login.close();
 cout<<"\n\n\t\t      Do you want to create another ID: ";
 cin>>choice;
}
while(choice=='y'||choice=='Y');
cout<<"\n\n\t\t      Press any key to return to main menu";
getch();
admin_mode();
}


void delete_id() //Function to delete a user ID
{
 char choice='n';
 do
 {
  label4:
  clrscr();
  ID del_user,temp_del,confirm;
```

```
cout<<"\n\n";
vline('*');
cout<<"\n\n\t\t\t   Delete a standard user ID";
cout<<"\n\n";
vline('*');
del_user.get_username();
cout<<"\n";
cout<<"\t          Retype the user name: ";
gets(confirm.user_name);
if(strcmp(del_user.user_name,confirm.user_name)!=0)
{
 cout<<"\n\n\n\n\n\t\tThe usernames that you have entered doesn't match";
 cout<<"\n\n\n\n\t\t\t   Press any key to try again";
 getch();
 goto label4;
}
del_user.get_password();
cout<<"\n";
cout<<"\t          Retype the password: ";
gets(confirm.password);
if(strcmp(del_user.password,confirm.password)!=0)
{
 cout<<"\n\n\n\n\n\t\tThe passwords that you have entered doesn't match";
 cout<<"\n\n\n\n\t\t\t   Press any key to try again";
 getch();
 goto label4;
 }
char found='n';
fstream del_org("user.dat",ios::in|ios::out|ios::binary);
```

```
 fstream del_temp("temp.dat",ios::in|ios::out|ios::app|ios::binary);
 while(!del_org.eof())
 {
  del_org.read((char*)&temp_del,sizeof(temp_del));
  if(strcmp(del_user.user_name,temp_del.user_name)!=0)
  {
   del_temp.write((char*)&temp_del,sizeof(temp_del));
  }

if(strcmp(del_user.user_name,temp_del.user_name)==0&&strcmp(del_user.pa
ssword,temp_del.password)==0)
  {
   found='y';
  }
 }
 del_org.close();
 del_temp.close();
 remove("user.dat");
 rename("temp.dat","user.dat");
 if(found=='y')
 {
  cout<<"\n\n\n\t\t    Your ID has been deleted successfully";
  cout<<"\n\n\t\t      Do you want to delete another ID: ";
  cin>>choice;
 }
 if(found=='n')
 {
  cout<<"\n\n\n\t\t          Your ID has not been found";
  cout<<"\n\n\t\t\t Do you want to try again(y/n):";
```

```
  cin>>choice;
  }
 }
 while(choice=='y' || choice=='Y');
 cout<<"\n\n\t\t      Press any key to return to main menu";
 getch();
 admin_mode();
}

void install_bus()
{
 char choice='n';
 do
 {
  clrscr();
  reserve ins;
  cout<<"\n\n";
  vline('*');
  cout<<"\n\n\t\t\t   Add a bus to the database";
  cout<<"\n\n";
  vline('*');
  cout<<"\n\n\n\tEnter bus no: ";
  cin>>ins.busn;
  cout<<"\tEnter Driver's name: ";
  gets(ins.driver);
  cout<<"\n\tArrival time: ";
  gets(ins.arrival);
  cout<<"\tDeparture time: ";
  gets(ins.depart);
```

```
  cout<<"\n\tOrigin: ";
  gets(ins.from);
  cout<<"\tDestination: ";
  gets(ins.to);
  cout<<"\n\tCost: ";
  cin>>ins.cost;
  for(int i=0; i<8;i++)
  {
   for(int j=0;j<4;j++)
   {
    strcpy(ins.seat[i][j], "Empty");
   }
  }
  fstream inst("busdata.dat",ios::app|ios::binary);
  inst.write((char*)&ins,sizeof(ins));
  cout<<"\n\n\t The details of the bus had been added to the database
successfully.";
  inst.close();
  cout<<"\n\n\t\t Do you want to add another bus to the database(y/n): ";
  cin>>choice;
 }while(choice=='Y'||choice=='y');
 cout<<"\n\n\t\t Press any key to return to main menu";
 getch();
 admin_mode();
}

void standard_login() //Function for end user login
{
 ID temp_user;
```

```
fstream std_login;
std_login.open("user.dat",ios::out|ios::in|ios::binary);
clrscr();
cout<<"\n\n";
vline('*');
cout<<"\n\n\t\t\t\tStandard login";
cout<<"\n\n";
vline('*');
login_id.get_username();
login_id.get_password();
char found='n';
while(!std_login.eof())
{
 std_login.read((char*)&temp_user,sizeof(temp_user));
 if(strcmp(temp_user.user_name,login_id.user_name)==0)
 {
  if(strcmp(temp_user.password,login_id.password)==0)
  {
  found='y';
  break;
  }
 }
}
 if(found=='n')
 {
  cout<<"\n\n\n\n\n\n\n\n\n\t\t   Either your name or password is incorrect";
  label2:
  cout<<"\n\n\t\t\t   Do you want to try again(y/n):";
  char try_choice1;
```

```
  cin>>try_choice1;
  switch(try_choice1)
  {
   case 'y':standard_login();
    break;
   case 'n':main_menu();
    break;
   case 'Y':standard_login();
    break;
   case 'N':main_menu();
    break;
   default :cout<<"\n\n\t\t Enter a valid input";
    goto label2;
  }
  }
 std_login.close();
 if(found=='y')
 {
  cout<<"\n\n\n\n\n\n\n\n\n\n\t\t\t   Press any key to continue";
  getch();
  standard_mode();
 }
}

void standard_mode() //Function for end user menu
{
 clrscr();
 cout<<"\n\n";
 vline('*');
```

```
cout<<"\n\n\t\t\t      STANDARD USER MODE";
cout<<"\n\n";
vline('*');
cout<<"\n\n\n\n\t 1.Book Tickets.";
cout<<"\n\n\t 2.View the details of bus by entering number";
cout<<"\n\n\t 3.Cancel your booked Tickets";
cout<<"\n\n\t 4.Main Menu";
cout<<"\n\n\t 0.Exit";
cout<<"\n\n\n\n\t Enter your choice(0 to 4): ";
int choice3;
cin>>choice3;
switch(choice3)
{
 case 1:book_tickets();
break;
 case 2:show_bus();
break;
 case 3:cancel_ticket();
break;
 case 4:main_menu();
break;
 case 0:exit(0);
break;
 default:cout<<"\n\n\n\t\t\t     Enter a valid choice";
 cout<<"\n\n\t\t\t   Press any key to continue";
 getch();
 standard_mode();
}
getch();
```

```
}

void book_tickets()   //Function to book tickets
{
 label15:
 reserve book_ticket,temp,temp1;
 clrscr();
 vline('*');
 cout<<"\n\n\t\t\t\t  Book Tickets\n\n";
 vline('*');
 cout<<"\n\n\t Enter the bus number: ";
 cin>>temp1.busn;
 int n,temp_seat;
 fstream ticket("busdata.dat",ios::in|ios::out|ios::binary|ios::app);
 ticket.seekg(0);
 char choice10='n';
 while(!ticket.eof())
 {
  ticket.read((char*)&book_ticket,sizeof(book_ticket));
  if(book_ticket.busn==temp1.busn)
  {
   if(book_ticket.check_empticket()!=0)
   {
    cout<<"\n\n\tEnter the number of tickets to be booked: ";
    cin>>n;
    if(n>book_ticket.check_empticket())
    {
     cout<<"\n\n\tThere are only "<<book_ticket.check_empticket()<<"
available in the bus";
```

```
 cout<<"\n\n\t\t\t    Do you want to try again(y/n): ";
 cin>>choice10;
 cout<<"\n\n\n\t\t\t     Press any key to continue";
 getch();
 if(choice10=='Y'||choice10=='y')
 {
  goto label15;
 }
 else
 {
  cout<<"\n\n\n\n\t\t      Press any key to return to main menu";
  getch();
  standard_mode();
 }
}
cout<<"\n\n";
top:
for(int i=1;i<=n;i++)
{
 cout<<"\n\t Enter the seat number for passenger "<<i<<": ";
 cin>>temp_seat;
 if(temp_seat>32)
 {
  cout<<"\n\tThere are only 32 seats available in this bus.";
  cout<<"\n\n\tDo you want to try again(y/n): ";
  cin>>choice10;
  cout<<"\n\n\n\t\t\t     Press any key to continue";
  getch();
  if(choice10=='Y'||choice10=='y')
```

```
 {
  goto label15;
 }
 else
 {
  cout<<"\n\n\n\n\t\t      Press any key to return to main menu";
  getch();
  standard_mode();
 }
}
if(book_ticket.check_empty((temp_seat/4),(temp_seat%4)-1)==1)
{
 cout<<"\n\tEnter the name of passenger "<<i<<": ";
 cin>>book_ticket.seat[temp_seat/4][(temp_seat%4)-1];
 strcpy(book_ticket.user_ID[temp_seat/4][(temp_seat%4)1],
  login_id.user_name);
}
else
{
 cout<<"\n\t This seat is reserved for other passenger";
 cout<<"\n\n\n\t\t\t     Press any key to continue";
 getch();
 goto top;
}
}
fstream del_temp("temp.dat",ios::in|ios::out|ios::app|ios::binary);
ticket.seekg(0);
del_temp.seekp(0);
while(!ticket.eof())
```

```
    {
     ticket.read((char*)&temp,sizeof(temp));
     if(book_ticket.busn!=temp.busn)
     {
      del_temp.write((char*)&temp,sizeof(temp));
     }
    }
    del_temp.close();
    ticket.close();
    remove("busdata.dat");
    rename("temp.dat","busdata.dat");
    fstream twrite("busdata.dat",ios::app|ios::binary);
    twrite.write((char*)&book_ticket,sizeof(book_ticket));
    twrite.close();
    cout<<"\n\n\t\t   Your tickets have been booked successfully";
    cout<<"\n\n\t The cost for the ticket is "<<book_ticket.cost;
    cout<<"\n\t Please pay the bill to the conductor before you start the
journey";
    cout<<"\n\t Please carry each of the traveller's ID card for verification";
    cout<<"\n\n\t\t      Press any key to return to main menu";
    break;
   }
  }
  if(book_ticket.check_empticket()==0)
  {
    cout<<"\n\n\t There are no tickets available in this bus";
    cout<<"\n\n\n\n\t\t      Press any key to return to main menu";
    getch();
    standard_mode();
```

```
  }
 }
 getch();
 standard_mode();
}


void show_bus() //Function to show the details of a bus
{
 reserve temp,search;
 clrscr();
 vline('*');
 cout<<"\n\n\t\t\t     The details of the bus";
 cout<<"\n\n";
 vline('*');
 cout<<"\n\n\n\t Enter the bus number: ";
 cin>>temp.busn;
 fstream disp("busdata.dat",ios::in|ios::out|ios::binary);
 while(!disp.eof())
 {
  disp.read((char*)&search,sizeof(search));
  if(search.busn==temp.busn)
  {
   search.display();
   show_seats(search);
   break;
  }
 }
  disp.close();
```

```
 cout<<"\n\n\n\t\t      Press any key to return to main menu";
 getch();
 standard_mode();
}

void show_seats(reserve seat)  //Function to show the seat map of a bus
{
 clrscr();
 cout<<"\n\n";
 vline('*');
 cout<<"\n\n\t\t\t The details of the seats of the bus";
 cout<<"\n\n";
 vline('*');
 cout<<"\n";
 seat.disp_seat();
 cout<<"\n\n\n\t\t      Press any key to return to main menu";
 getch();
 standard_mode();
}

void cancel_ticket() //Function to cancel a booked ticket
{
 label6:
 clrscr();
 reserve del,temp,temp2;
 char choice='n';
 int seats;
 cout<<"\n\n";
 vline('*');
```

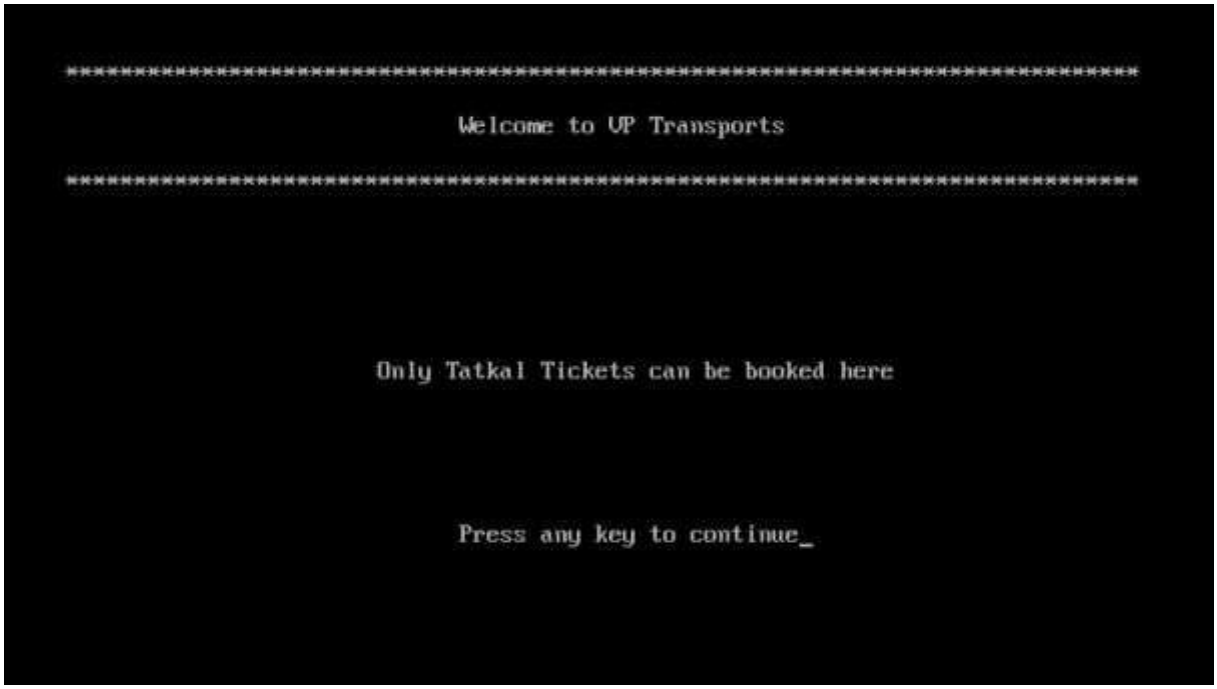
```
cout<<"\n\n\t\t\t   Cancel the booked tickets";
cout<<"\n\n";
vline('*');
cout<<"\n\n";
do
{
 cout<<"\t Enter the bus number: ";
 cin>>temp.busn;
 fstream bus("busdata.dat",ios::binary|ios::in);
 while(!bus.eof())
 {
  bus.read((char*)&del,sizeof(del));
  if(del.busn==temp.busn)
  {
   cout<<"\n\n\t Enter the seat number to be cancelled: ";
   cin>>seats;
   if(strcmp(del.seat[seats/4][(seats%4)-1],"Empty")!=0)
   {
    if(strcmp(del.user_ID[seats/4][(seats%4)-1],login_id.user_name)!=0)
    {
     cout<<"\n\n\t\tThis ticket is not booked from your user id";
     cout<<"\n\n\t\t\tDo you want to try again(y/n): ";
     cin>>choice;
     if(choice=='y'||choice=='Y')
     {
      goto label6;
     }
     else
     {
```

```
  standard_mode();
 }
}
strcpy(del.seat[seats/4][(seats%4)-1],"Empty");
fstream temp1("temp1.dat",ios::app|ios::binary);
bus.seekg(0);
while(!bus.eof())
{
 bus.read((char*)&temp2,sizeof(temp2));
 if(del.busn!=temp2.busn)
 {
  temp1.write((char*)&temp2,sizeof(temp2));
 }
}
temp1.write((char*)&del,sizeof(del));
temp1.close();
bus.close();
remove("busdata.dat");
rename("temp1.dat","busdata.dat");
cout<<"\n\n\t\t Your booked seat have been cancelled successfully";
cout<<"\n\n\t Do yo want cancel one more ticket(Y/N): " ;
cin>>choice;
break;
}
else
{
cout<<"\n\n\t The selected seat havent booked by anyone";
cout<<"\n\n\t\t\t Do yo want to try again(Y/N): " ;
cin>>choice;
```

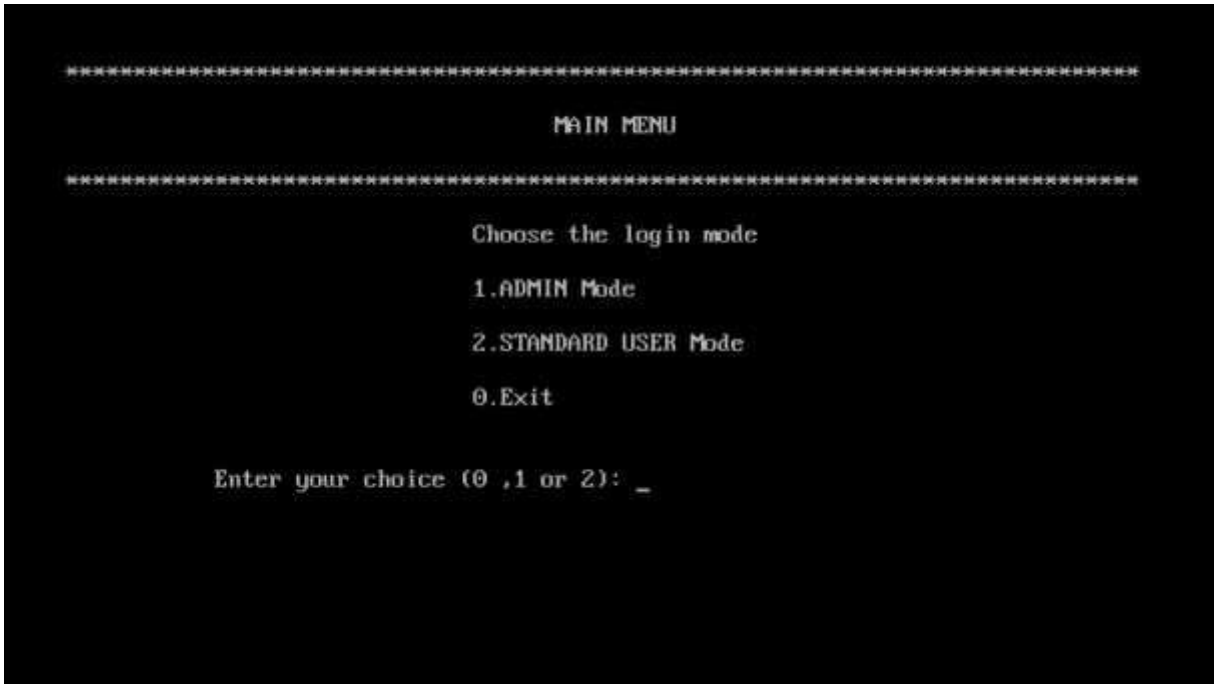
```
    if(choice=='y'||choice=='Y')
    goto label6;
    else
    {
      cout<<"\n\n\n\n\t\t      Press any key to return to main menu";
      getch();
      standard_mode();
    }
   }
  }
 }while(choice=='y'||choice=='Y');
 cout<<"\n\n\n\n\t\t      Press any key to return to main menu";
 getch();
 standard_mode();
}

void main() //Main Function
{
 start_screen();
 main_menu();
}
```

OUTPUT

Start Screen:



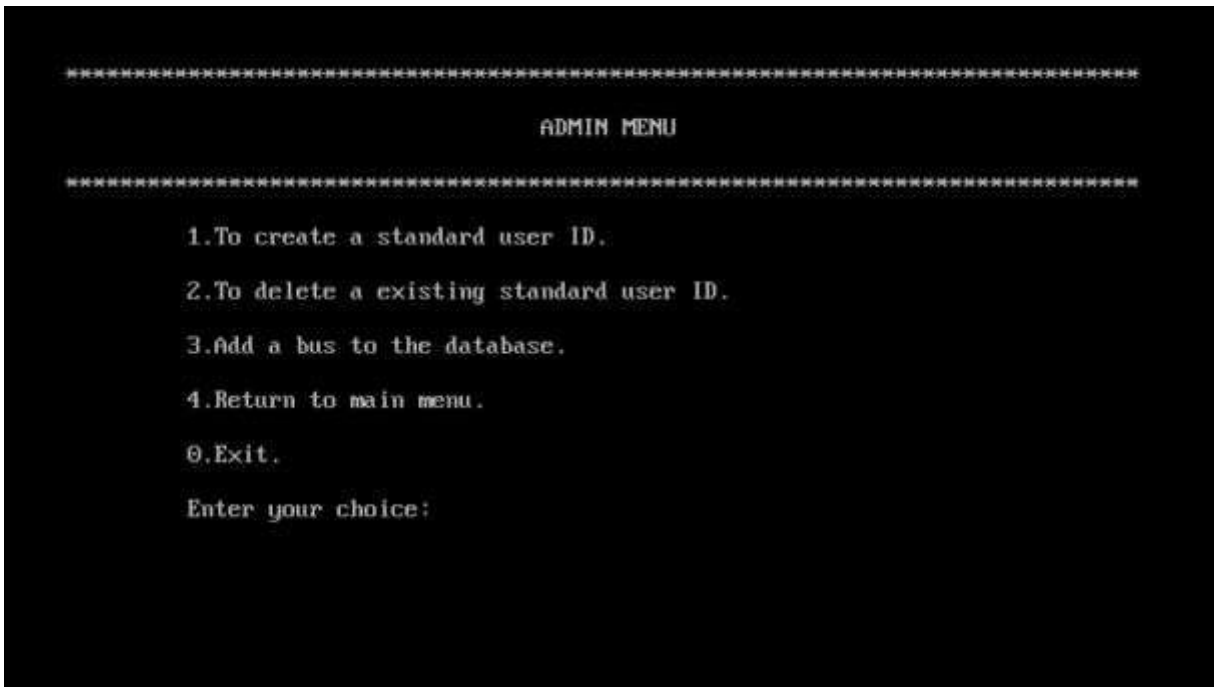
Main Menu:



Admin login Successful:

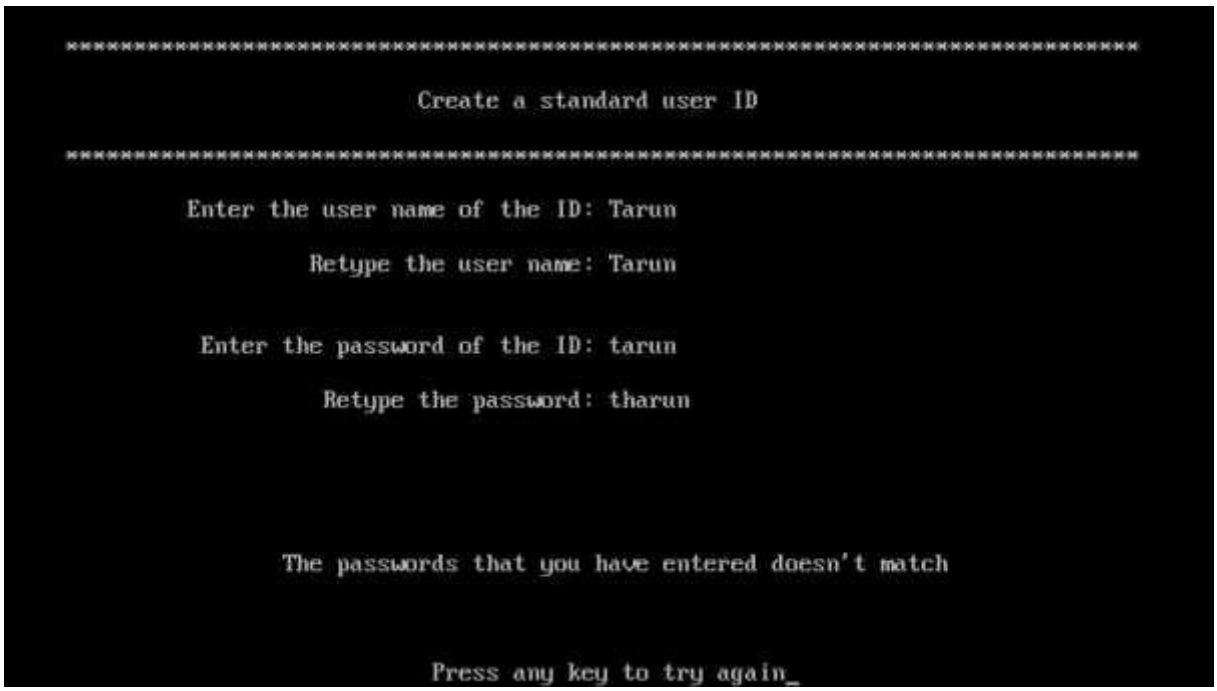
Admin login Unsuccessful:

Admin Mode Screen:



Create User ID (Name Incorrect):

Create User ID (Password Incorrect):



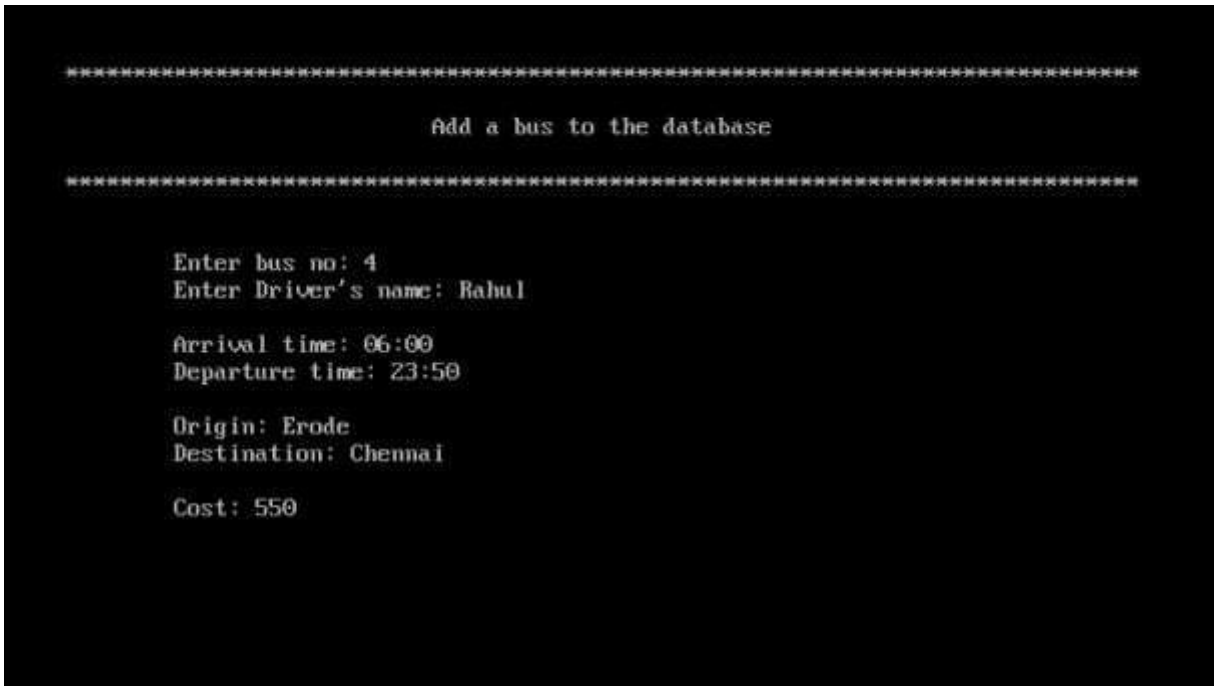
Create User ID (ID already exists):

Create User ID (Confirmation Page):

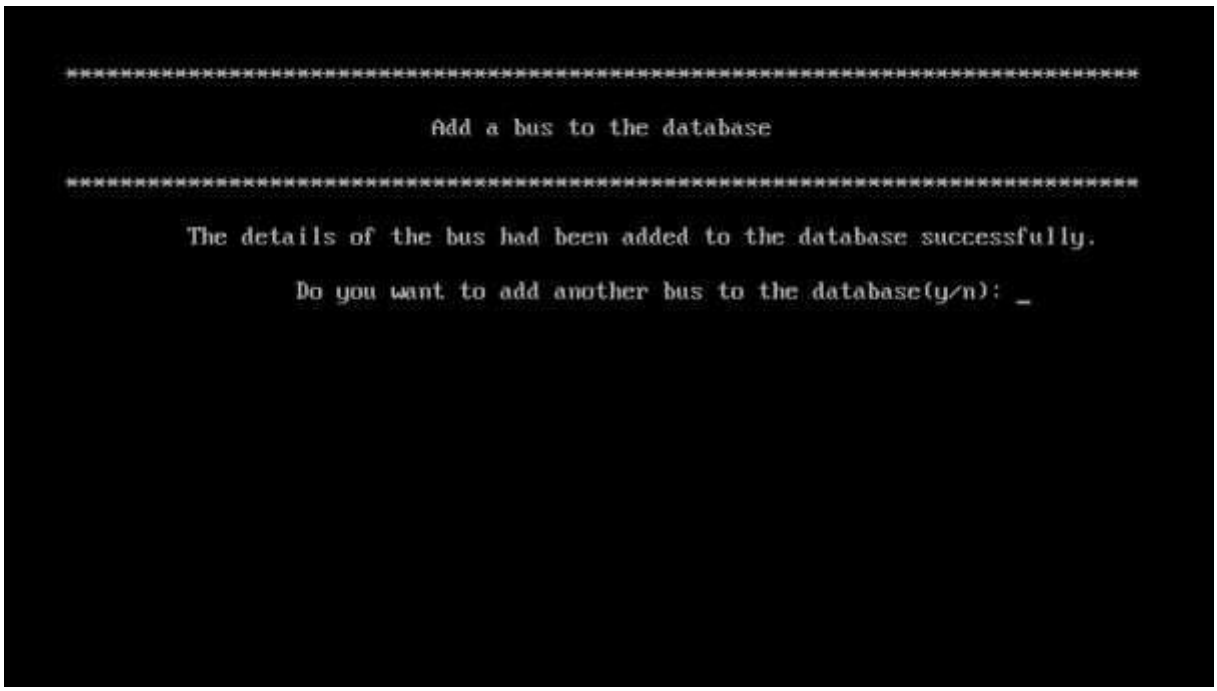
Delete User ID (ID not found):

Delete User ID (Confirmation Page):

Add a bus to the database (Input Page):



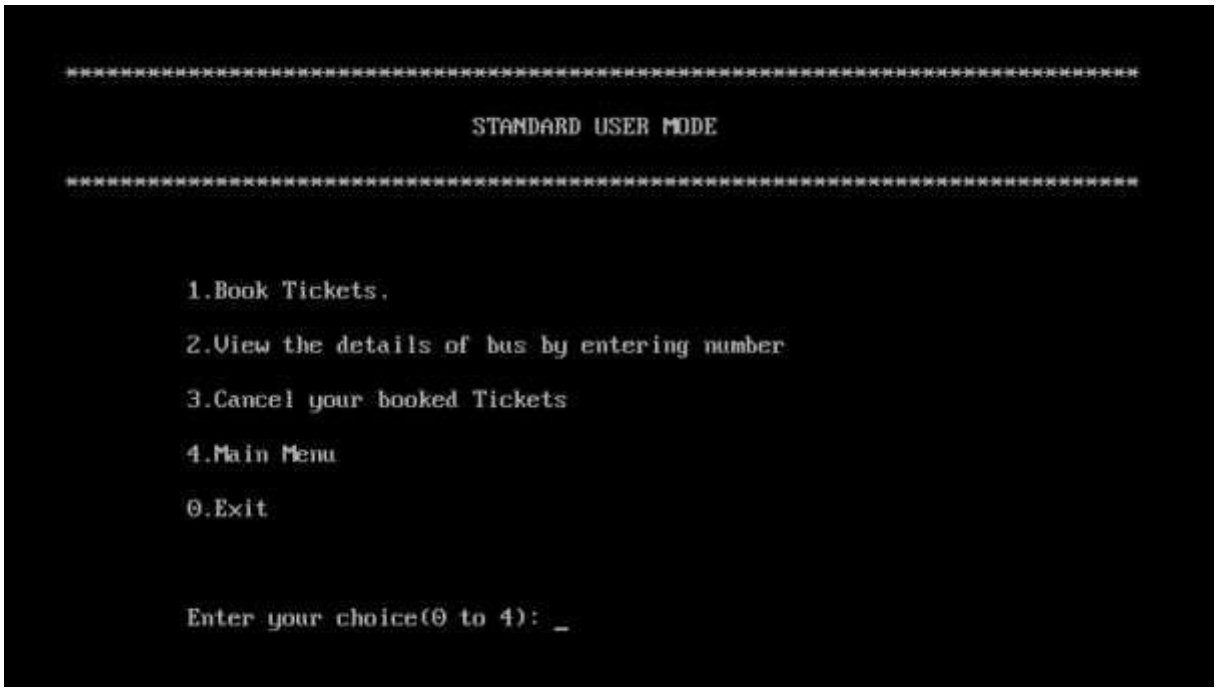
Add a bus to the database (Confirmation Page):



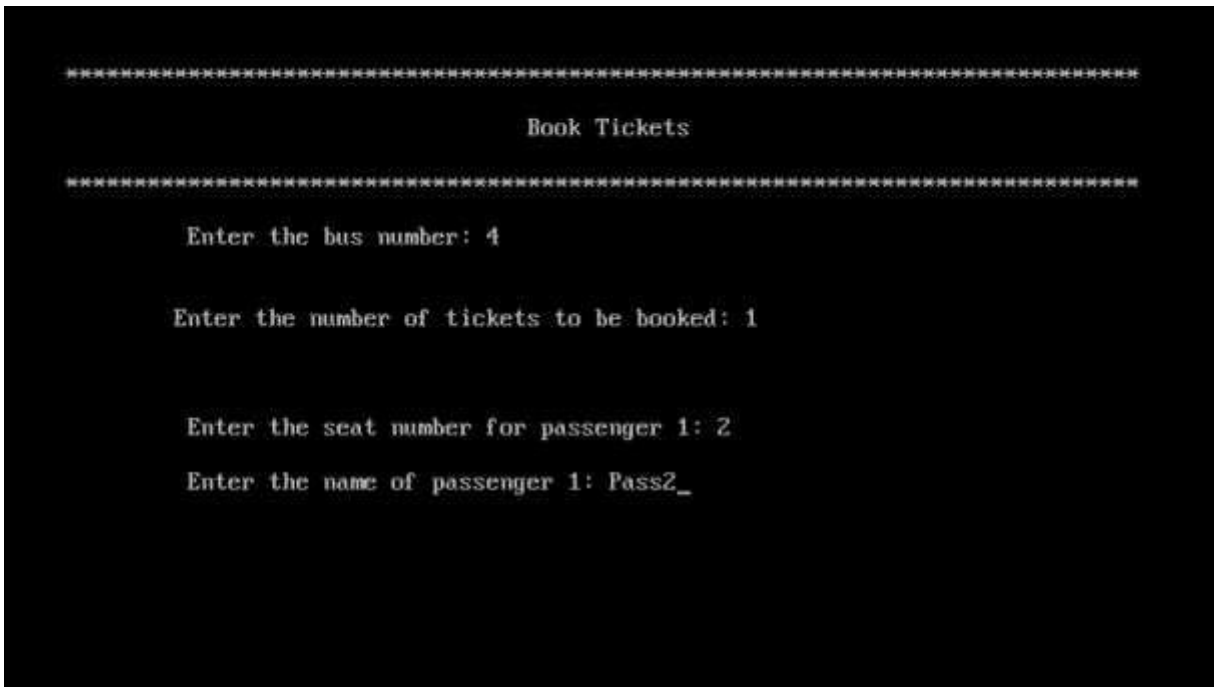
Standard User login Successful:

Standard User ID login Unsuccessful:

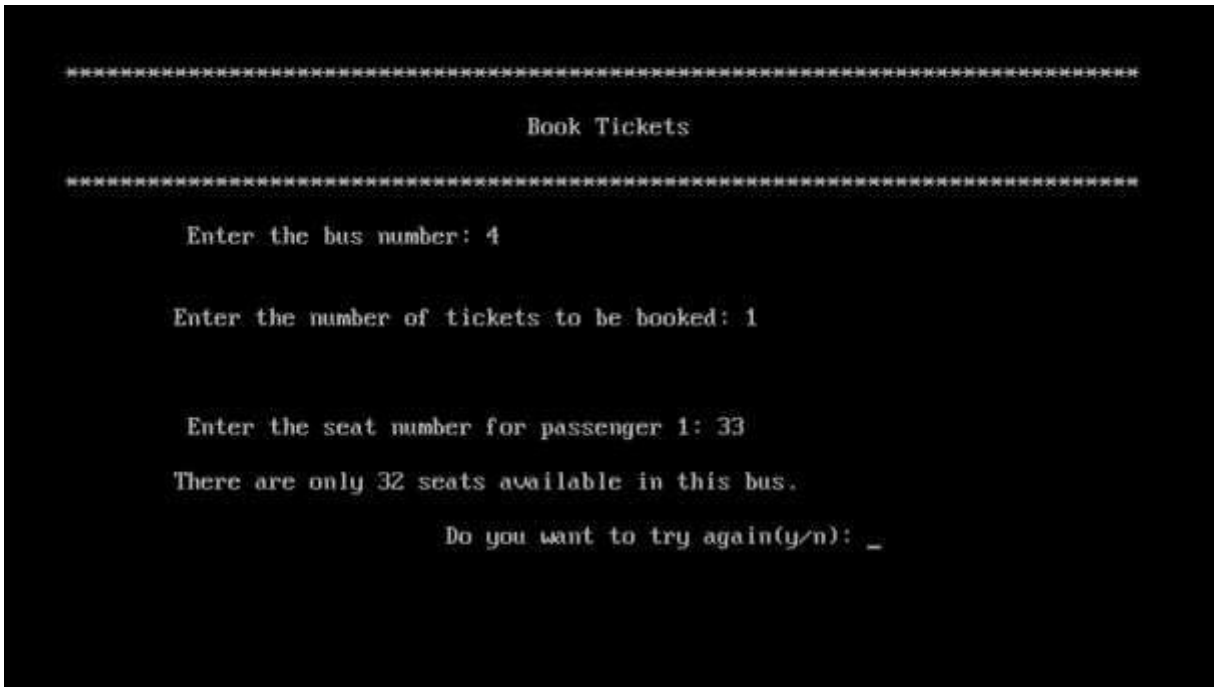
Standard User Menu:



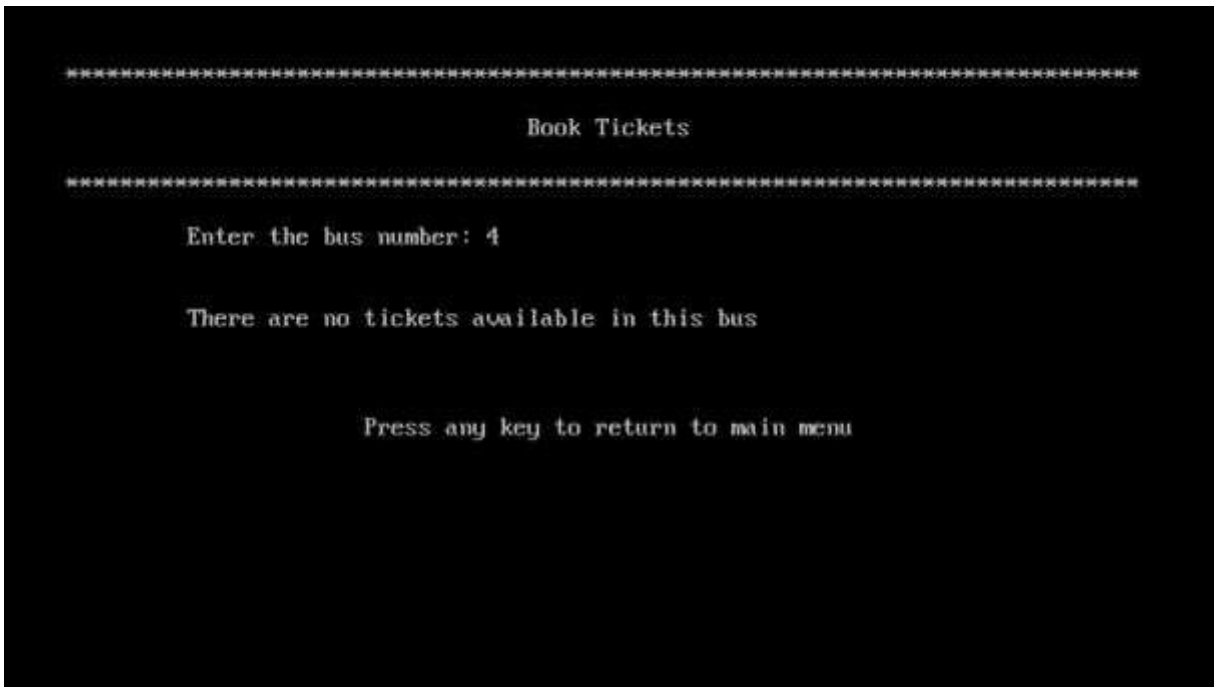
Book tickets (Details Page):



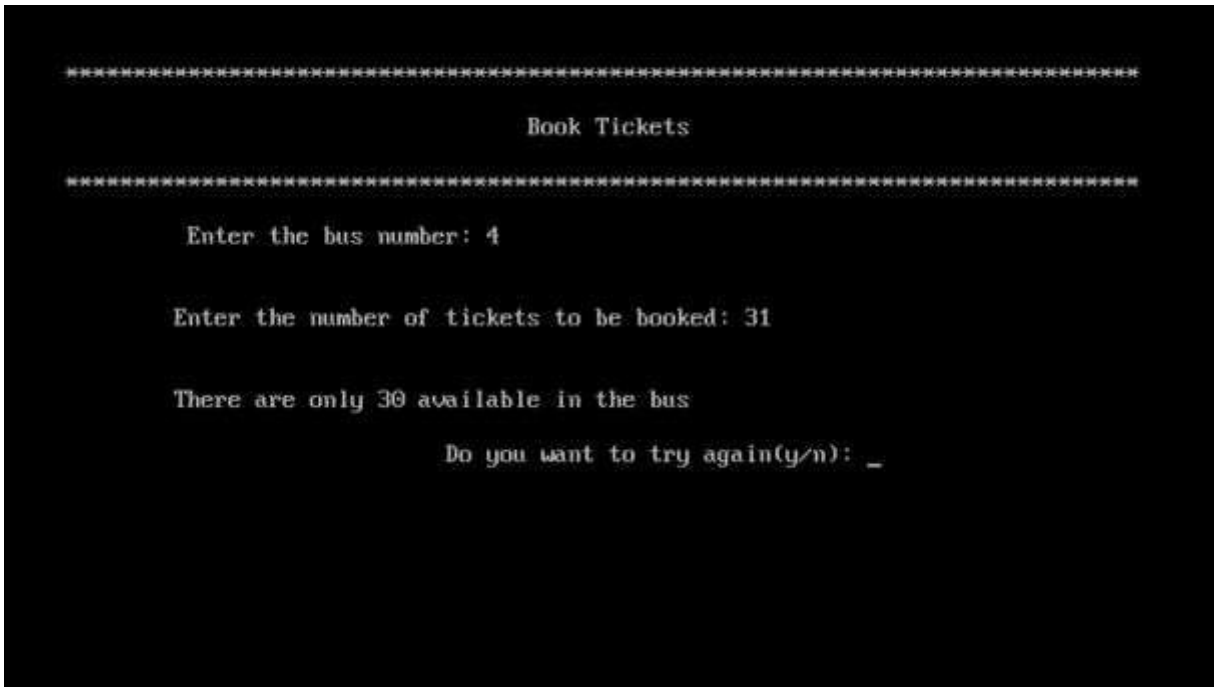
Book Tickets(Selected seat number is not in the bus):



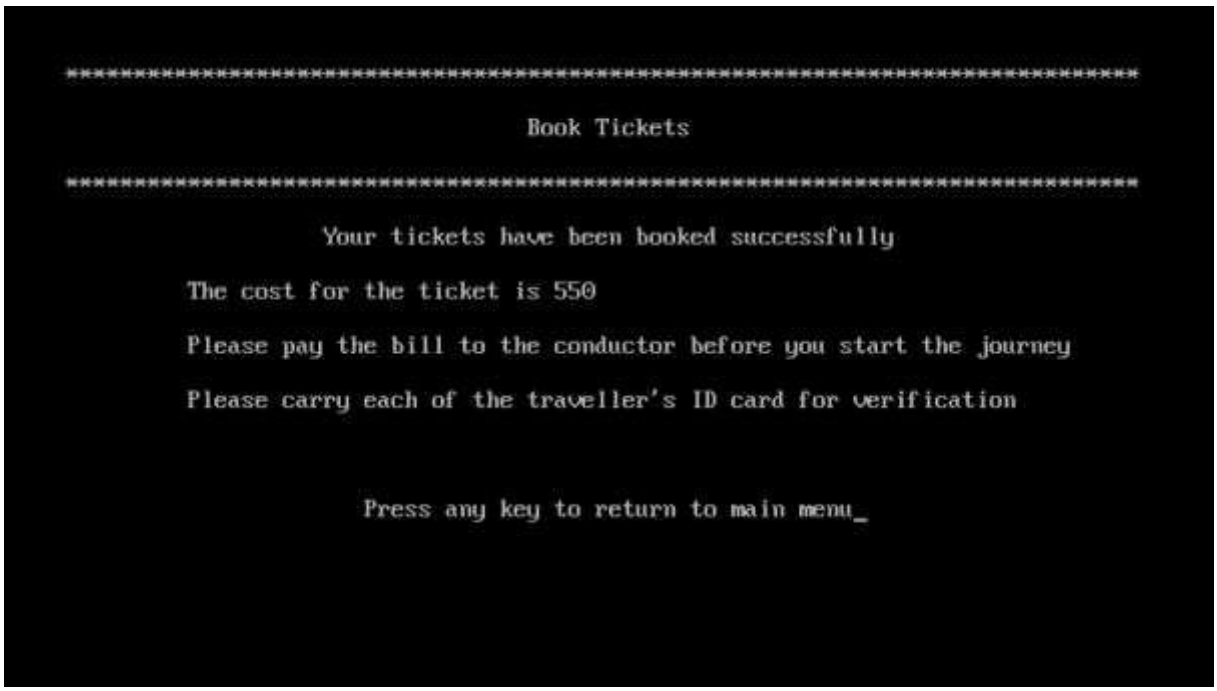
Book Tickets (No seats are available in the bus):



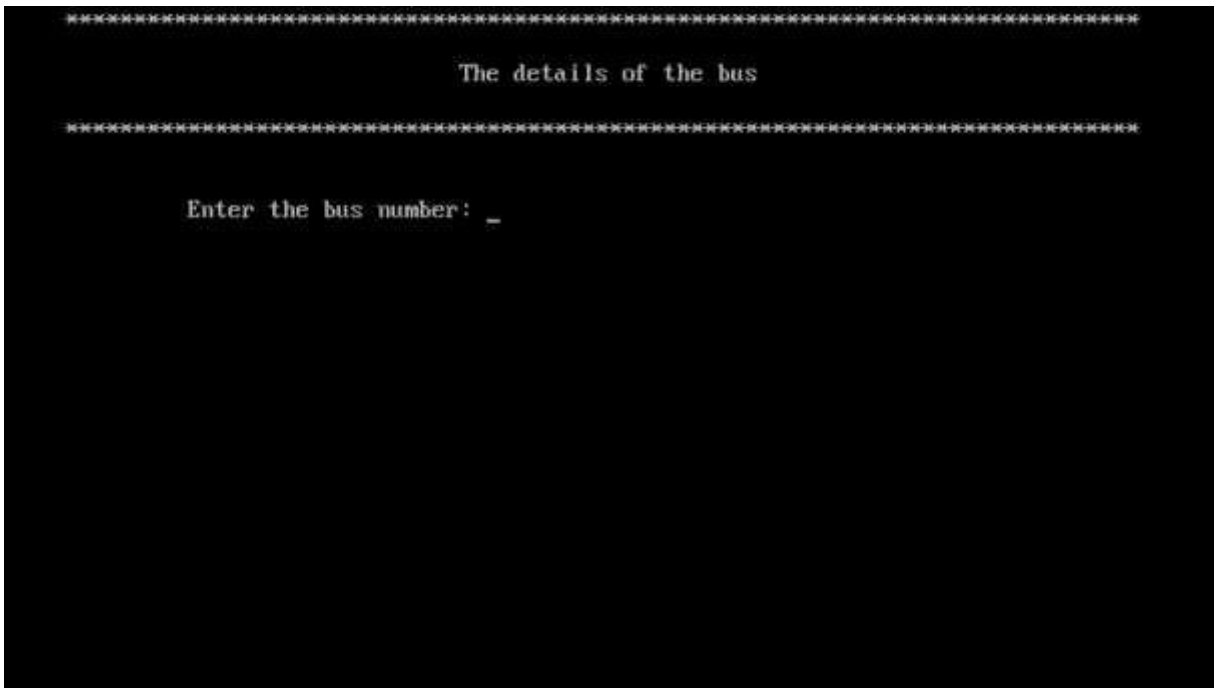
Book Tickets (No sufficient seats are available):



Book Tickets (Confirmation Page):



Show bus details (Get Info Page):



Show Bus Details (Details Page):

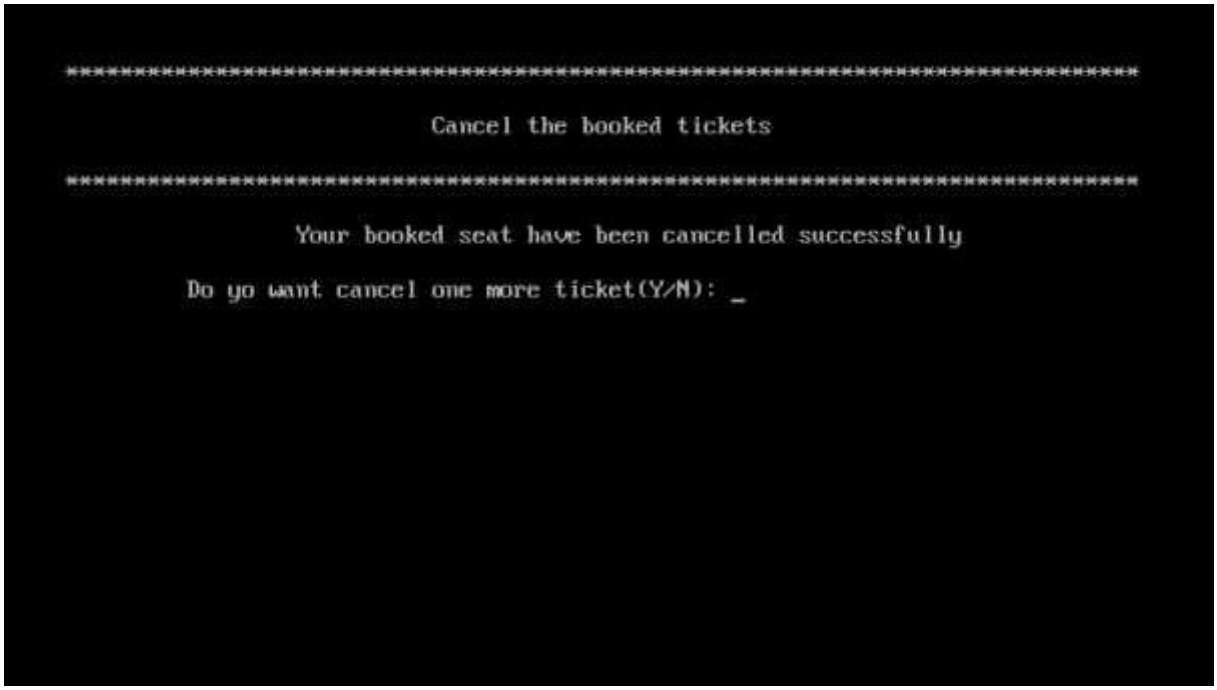
Show Bus Details (Seat Map):

Cancel Tickets (Details Page):

Cancel Tickets (Already Unreserved):

Cancel Tickets (Booked with another User ID):

Cancel Tickets (Confirmation Page):



CONCLUSION

The accurate knowledge of the Bus data and management allow the user to expand and grow his business. Bus Reservation System is the perfect tool to assist in business management. Our project is only a humble venture to satisfy the needs to manage their project work. Several user friendly coding have also adapted. The package shall prove to be a powerful package in satisfying all the requirement in school. The objective of software planning is to provide a framework that enables the manager to make reasonable estimate made within a limited time frame at the beginning of the software project and should be updated regularly as the project progresses.

At the end it is concluded that we have made effort on following points.

- ✓ A description of the background and context of the project and its relation to work already done in the area.
- ✓ Made statement of the aims and objectives of the project.
- ✓ The description of purpose, scope and applicability.
- ✓ We define the problem on which we are working in the project.
- ✓ We describe the requirement specification of the system and the actions that can be done on these things.
- ✓ We understand the problem domain and produce a model of the system, which describe operations that can be performed on the system.
- ✓ We included features and operation in detail, including screen layouts.
- ✓ Finally, the system is implemented and tested according to test cases.

FUTURE ENHANCEMENTS

The system can be further enhanced in future by including the following features:

- By making the entire process online which helps the passengers to search buses and do transactions.
- By implementing the facility for access for more than one admin where more than one admin can login and modify the details of the company.
- By implementing the concept of booking of tickets for a particular bus for the next three months.

The new system is designed such that these enhancements can be integrated with the current modules easily with less integration work. The enhanced features make project more interactive, more user friendly and can fulfill each users need in the best way possible.

BIBLIOGRAPHY

Books:

- ✓ Computer Science with C++ for class XII by Sumita Arora
- ✓ Computer Science with C++ for class XI by Sumita Arora
- ✓ CBSE All In One Computer Science CBSE Class 12 (Old edition).

Websites:

- ✓ www.wikipedia.com
- ✓ www.slideshare.com
- ✓ www.projectsgeek.com
- ✓ www.google.com
- ✓ www.codeproject.com
- ✓ www.geeksforgeeks.org
- ✓ www.stackoverflow.com
- ✓ www.zetcode.com