

# SRM BUS RESERVATION SYSTEM

Project submitted to the

SRM University – AP, Andhra Pradesh

For the partial fulfillment of the requirements to award the degree of

Bachelor of Technology

In

Computer Science and Engineering School of Engineering and Sciences

DONE BY-

**D Sai Dhanush**

**AP21110010397**



*Under the Guidance of*

**SUBHANKAR GHATAK**

*SRM University–AP*

*Neerukonda, Mangalagiri,*

*Guntur Andhra Pradesh – 522 240*

*[December, 2022]*

## **Certificate**

DATE: 05-DEC-22

**This is to certify that the work present in this Project entitled “SRM BUS RESERVATION SYSTEM” has been carried out by D Sai Dhanush under supervision. The work is genuine, original, and suitable for submission to the SRM University – AP for the award of Bachelor of Technology in School of Engineering and Sciences.**

**Supervisor**

**(Signature)**

**Dr. SUBHANKAR GHATAK**

**Assistant Professor, CSE Department,**

**SRM UNIVERSITY, AP**

# Acknowledgements

*I'd like to thank my teacher, Subhankar ghatak, for giving me the opportunity to work on this project. This project taught me a lot about different core concepts of Object-Oriented Programming, such as data encapsulation and abstraction. Inheritance, objects and classes etc.. I am grateful to him for his assistance and guidance in completing this project. Finally, I'd like to thank all of my dear teammates for being a part of the team and for assisting me in completing this project.*

**GROUP-I**

# TABLE OF CONTENTS

<b>SRL.NO'S</b>	<b>CONTENTS</b>	<b>PAGE NO'S</b>
<b>1</b>	<b>TITLE PAGE</b>	<b>1</b>
<b>2</b>	<b>CERTIFICATE</b>	<b>2</b>
<b>3</b>	<b>ACKNOWLEDGMENT</b>	<b>3</b>
<b>4</b>	<b>ABSTRACT</b>	<b>5</b>
<b>5</b>	<b>CONTRIBUTIONS</b>	<b>6</b>
<b>6</b>	<b>INTRODUCTION</b>	<b>7</b>
<b>7</b>	<b>FUNCTIONALITIES</b>	<b>8</b>
<b>8</b>	<b>OBJECTIVE</b>	<b>9</b>
<b>9</b>	<b>CODE IMPLEMENTATIONS</b>	<b>10-38</b>
<b>10</b>	<b>OUTPUTS</b>	<b>38-40</b>
<b>11</b>	<b>CONCLUSION</b>	<b>41</b>

## ABSTRACT

- **SRM BUS RESERVATION SYSTEM** deals with the installation of the buses and maintain the record of details of each student. It also includes maintenance of information like schedule and details of each bus.
- We can provide admin to view bus details, add bus details and remove bus details within the system.
- With this project, we can lead an error free, secure, reliable, fast bus reservation system.
- Our project goal is to manage all the information about bus, student name, no of seats and to replace all the paper work and provide better services to the student.

## **INTRODUCTION:-**

- **We observed the working of the “ SRM Bus Reservation System” and after going through it, we get to know that there are many operations, which they have to do manually.**
- **It takes a lot of time and causing many errors while data entry. Due to this, sometimes a lot of problems occur and they were facing many disputes with students.**
- **To solve the above problem, and further maintaining records of students details, seat availability, Mobile number, etc.. we are offering this proposal of computerized reservation system.**
- **By using this software, we can install the buses, and also reserve tickets through this system. student can check availability of bus and reserve seats.**
- **The project provides and checks all sorts of constraints so that admin staff does give only useful data and thus validation is done in an effective way.**

## **FUNCTIONALITIES**

### **SRM BUS RESERVATION SYSTEM HOLDS MAIN MENU INTERFACE**

**HERE, FROM MAIN MENU WE CAN ACCESS-**

- MENU**
- INSTALL BUSES**
- DISPLAY BUSES**
- DELETE BUSES**
- BUS SEATS AVAILABILITY**
- RESERVATION**
- UPDATE BOOKING SYSTEM**
- DELETE BOOKING SYSTEM**
- SHOW BOOKING SYSTEM**

**OBJECTIVE : -**

**The main objective of “SRM BUS RESERVATION SYSTEM” is to help**

- **Traveling Agency to organize bus schedules more efficiently on the basis of traffic on different route.**
- **All the booking and cancellation process to be easily done with this application.**
- **To provide student to view seat booking status and select their seat in the bus.**
- **Student can book their bus from online by opening SRM student portal. There is no need to visit Transport helpdesk manually in the campus.**



## CODE IMPLEMENTATION

### **CLASS BUS**

```
#include <iostream>

#include <fstream>

using namespace std;

class bus
{
    string bus_no, bus_driver_name, arrival_time, departure_time, to, from;

    int bus_seats;

public:

    void menu();

    void install_bus();

    void display_bus();

    void delete_bus();

    void bus_seats_details();

    void reservation();

    void update_booking();

    void del_booking();

    void show_booking();
```

```
};
```

```
void decorate(char c)
{
    for(int i=0;i<150;i++)
        cout<<c;
    cout<<endl;
}
```

## BUS MENU

```
void bus::menu()
{
    while (1)
    {
        int choice;
        cout<<endl;
        decorate('-');
        cout << "\n\t\t SRM BUS RESERVATION SYSTEM  \n";
        decorate('-');

        cout << "\n\n **MAIN MENU**";
        cout << "\n\n 1. INSTALL BUS DETAILS";
        cout << "\n 2. DELETE BUS DETAILS";
```

```
cout << "\n 3. DISPLAY BUSES";

cout << "\n 4. AVAIABLE SEATS DETAILS ";

cout << "\n 5. RESERVATION";

cout << "\n 6. UPDATE BOOKING RECORD";

cout << "\n 7. DELETE BOOKING RECORD";

cout << "\n 8. SHOW ALL BOOKING RECORD";

cout << "\n 9. EXIT";

cout << "\n\n ENTER YOUR CHOICE : ";

cin >> choice;


switch (choice)
{
case 1:

    install_bus();

    break;

case 2:

    delete_bus();

    break;

case 3:

    display_bus();

    break;

case 4:

    bus_seats_details();
```

```
        break;

    case 5:

        reservation();

        break;

    case 6:

        update_booking();

        break;

    case 7:

        del_booking();

        break;

    case 8:

        show_booking();

        break;

    case 9:

        exit(0);

    default:

        cout << "Invalid choice";

    }

}

}
```

### **Module (install bus)**

```
void bus::install_bus()
```

**{****here:****fstream file;****string test\_no, test\_bus\_name, test\_driver\_name;****string test\_arrival\_time, test\_departure\_time, test\_to, test\_from; // These variables are for to check whether same variables exist or not****int test\_seats;****int found = 0; // When no record or file found then found=0****cout << "\n\n Enter the bus number :";****cin >> bus\_no;****cout << "\n\n Enter the Driver name : ";****cin >> bus\_driver\_name;****cout << "\n\n Enter the arrival time : ";****cin >> arrival\_time;****cout << "\n\n Enter the departure time : ";****cin >> departure\_time;****cout<<"\n\n From : ";**

```
cin>>from;
```

```
cout<<"\n\n To : ";
```

```
cin>>to;
```

```
cout << "\n\n Enter the total Seats : ";
```

```
cin >> bus_seats;
```

```
file.open("bus.txt", ios::in); // To read whether there exist or not to avoid  
duplicate of same bus information
```

```
if (!file) // No file exist-->Creating new file
```

```
{
```

```
file.open("bus.txt", ios::app | ios::out);
```

```
file << bus_no << " " << bus_driver_name<<" "<< arrival_time << " " <<  
departure_time << " " << from <<" "<<to<<" "<< bus_seats<<" "<< endl;
```

```
file.close();
```

```
}
```

```
else // File exists->Check whether same information is there or not
```

```
{
```

```

    file >>
test_no>>test_driver_name>>test_arrival_time>>test_departure_time>>test_fr
om>>test_to>>bus_seats;

    while (!file.eof())
    {
        if (bus_no == test_no)
        {
            found++;
        }

        file >>
test_no>>test_driver_name>>test_arrival_time>>test_departure_time>>test_fr
om>>test_to>>bus_seats;

    }

    file.close();

    if (found == 0) // record no exist(same) -->then copy inside record
    {

        file.open("bus.txt", ios::app | ios::out);

        file << bus_no << " " << bus_driver_name << " " << arrival_time << " " <<
departure_time << " " << from << " " << to << " " << bus_seats << " " << endl;

```

```
        file.close();  
    }  
  
    else  
    {  
        cout << "\n\n Duplicate Record found";  
        goto here;  
    }  
}  
  
cout << "\n\nBus installed successfully";  
}
```

### **Module (display buses)**

```
void bus::display_bus()  
{  
    fstream file;  
  
    file.open("bus.txt", ios::in);  
  
    if (!file) // No file created  
    {  
        cout << "\n\nPlease install the bus details to view.";  
    }  
}
```



```

else
{
    cout << "\n\n BUS NO.\tDRIVER NAME\tARRIVAL TIME\tDEPARTURE
TIME\tFROM\t\tTO\t\tNO. OF SEATS";

    file>>bus_no>>bus_driver_name>>arrival_time>>departure_time>>from>>to>>
bus_seats;

    while (!file.eof())
    {

        cout<<"\n"<<bus_no<<"\t\t"<<bus_driver_name<<"\t\t"<<arrival_time<<"\t\t"<<d
eparture_time<<"\t\t"<<from<<"\t\t"<<to<<"\t\t"<<bus_seats<<endl;

        file>>bus_no>>bus_driver_name>>arrival_time>>departure_time>>from>>to>>
bus_seats;

    }

}

file.close();

```

```
}
```

### **Module (delete buses)**

```
void bus::delete_bus()
```

```
{
```

```
    fstream file,file1;
```

```
    string test_num;
```

```
    int found = 0;
```

```
    file.open("bus.txt",ios::in);
```

```
    if(!file)
```

```
    {
```

```
        cout<<"\n\n Please install the bus details to delete.";
```

```
    }
```

```
    else
```

```
    {
```

```
        cout<<"\n\n Enter the bus number to be deleted : ";
```

```
        cin>>test_num;
```

```
        file1.open("bus1.txt",ios::app|ios::out);
```

```

file>>bus_no>>bus_driver_name>>arrival_time>>departure_time>>from>>to>>
bus_seats;

```

```

while(!file.eof())
{
    if(test_num == bus_no)
    {
        cout<<"\n\n\n Bus has been deleted.";
        found++;
    }
    else{
        file1 << bus_no << " " << bus_driver_name << " " << arrival_time << "
" << departure_time << " " << from <<" " <<to<<" " << bus_seats<<" " << endl;
    }
}

```

```

file>>bus_no>>bus_driver_name>>arrival_time>>departure_time>>from>>to>>
bus_seats;

```

```

}

```

```

file.close();

```

```

file1.close();

```

```

remove("bus.txt");

```

```

        rename("bus1.txt","bus.txt");

        if(found == 0)
        {
            cout<<"\n\nBus number is Invalid";
        }

    }
}

```

## Module(bus seats details)

```

void bus::bus_seats_details()
{
    fstream file,file1; // file --> bus details , file1 -->seats details

    file.open("bus.txt",ios::in);

    file1.open("seat.txt",ios::in);//will come when reservation is used


    string test_no,seats_bus_num,seats_num;

    int count=0; //for reserved seats

    int found=0;


    if(!file || !file1) // File was not found
    {

        cout<<"\n\nPlease install the bus details to view. ";
    }
}

```

```

}

else{

    cout<<"\n\n Enter the Bus Number : ";

    cin>>test_no;

    file1>>seats_bus_num>>seats_num;

    while(!file1.eof())

    {

        if(test_no == seats_bus_num)

        {

            count++;

        }

        file1>>seats_bus_num>>seats_num;

    }

    file1.close();

```

```

file>>bus_no>>bus_driver_name>>arrival_time>>departure_time>>from>>to>>
bus_seats;

while(!file.eof())

{

    if(test_no == bus_no)

    {

        cout<<"\n\n Total number of seats : "<<bus_seats;

```

```

        cout<<"\n\n Reserved number of seats : "<<count;

        cout<<"\n\n Empty number of seats : "<<bus_seats-count;


        found++;

        cout<<endl<<endl;

    }

    file>>bus_no>>bus_driver_name>>arrival_time>>departure_time>>from>>to>>
    bus_seats;

}

file.close();

if(found == 0)
{
    cout<<"\n\n Bus number is invalid ";
}

}

}

```

### **Module(Reservation)**

```

void bus:: reservation()

{

    p:

```

```

fstream file;

file.open("bus.txt",ios::in);

string t_no,s_b_no,student_name,phone;

int found =0,s_no,seats,count=0,s_s_no; //s_no selected seat no to book,
seats -> total no of seats , count-> To check all seats are reserved or not.

if(!file)
{

    cout<<"\n\n Please install the bus details to do reservation.";

    cin>>t_no;

}

else

{

    cout<<"\n\n Enter the bus number : " ;

    cin>>t_no;

    file.close();

    file.open("seat.txt",ios::in);

    if(file) // File exists

    {

        file>>s_b_no>>s_s_no;

```

```
while(!file.eof())  
{  
    if(t_no == s_b_no)  
        count++;  
  
    file>>s_b_no>>s_s_no;  
}  
  
file.close();  
}
```

```
file.open("bus.txt",ios::in);
```

```
file>>bus_no>>bus_driver_name>>arrival_time>>departure_time>>from>>to>>  
bus_seats;
```

```
while(!file.eof())  
{  
    if(t_no == bus_no)  
    {  
        seats = bus_seats;  
        found++;  
    }  
}
```



```
}
```

```
file>>bus_no>>bus_driver_name>>arrival_time>>departure_time>>from>>to>>  
bus_seats;  
  
}
```

```
file.close();
```

```
if(seats - count == 0)
```

```
    cout<<"\n\n All seats are reserved";
```

```
else if(found == 1)
```

```
{
```

```
    h:
```

```
    cout<<"\n\n Enter the Seat number : ";
```

```
    cin>>s_no;
```

```
    if(s_no > seats)
```

```
    {
```

```
        cout<<"\n\n Seat number is Invalid.";
```

```
        goto h;
```

```

    }

    file.open("seat.txt",ios::in);

    if(!file) //new file ->all seats empty
    {
        file.open("seat.txt",ios::app | ios:: out);

        file<<t_no<<" "<<s_no<<"\n";

        file.close();
    }

    else // File exist ->To check if the seat is reserveed or not
    {
        file>>s_b_no>>s_s_no;

        while(!file.eof())
        {
            if(t_no == s_b_no && s_no == s_s_no) // bus no and seat no
matches
            {

                cout<<"\n\n This seat is already reserved";

                goto h;
            }
        }
    }

```

```
        file>>s_b_no>>s_s_no;

    }

    file.close();

    file.open("seat.txt",ios::app | ios:: out);

    file<<t_no<<" "<<s_no<<"\n";

    file.close();

}

cout<<"\n\n Enter your name : ";

cin>>student_name;

cout<<"\n\n Enter your phone number : ";

cin>>phone;

file.open("student.txt",ios::app | ios::in);

file<<student_name<<" "<<t_no<<" "<<phone<<" "<<s_no<<endl;

file.close();
```

```

    decorate('*');

    cout<<"\n\n Booking Information \n";

    decorate('*');

    cout<<"\n\n Student Name :      "<<student_name;

    cout<<"\n\n Phone Number :      "<<phone;

    cout<<"\n\n Seat number :      "<<s_no;

    cout<<"\n\n Booking Succesful";

}

else{

    cout<<"\n\n Bus number is Invalid";

    goto p;

}

}

}

```

### Module(update booking)

```

void bus:: update_booking()

{

    fstream file,file1;

    int r_seats,found = 0,i=0;

    string name,no,phone,t_name,t_phone,del_no;//no-bus no

```

```
file.open("student.txt",ios::in);

if(!file)

{

    cout<<"\n\n Please install the bus details to update.";

}

else

{

    cout<<"\n\n Enter your phone number : ";

    cin>>t_phone;

    file>>name>>no>>phone>>r_seats;

    while(!file.eof())

    {

        if(t_phone == phone)

        {

            file.close();

            file.open("student.txt",ios::in);

            file1.open("student1.txt",ios::app|ios::out);

            file>>name>>no>>phone>>r_seats;
```

```

while(!file.eof())

{

    if(t_phone == phone) // To clear previous seats which were
reserved in the previous bookings

    {

        del_no = no;

    }

    if(t_phone != phone)

    {

        file1<<name<<" "<<no<<" "<<phone<<" "<<r_seats<<endl;

    }

    file>>name>>no>>phone>>r_seats;

}

file.close();

file1.close();


remove("student.txt");

rename("student1.txt","student.txt");


file.open("seat.txt",ios::in);

file1.open("seat1.txt",ios::app | ios:: out);

```

```
file>>no>>r_seats;

while(!file.eof())
{
    if(del_no != no )
    {
        file1<<no<<" "<<r_seats<<endl;
    }

    file>>no>>r_seats;
}

file.close();

file1.close();


remove("seat.txt");

rename("seat1.txt","seat.txt");


reservation();

cout<<"\n\n Updated Reserved Seats";


found++;

goto h;
}

file>>name>>no>>phone>>r_seats;

}
```

```

        file.close();

    h:
    if(found == 0)
    {
        cout<<"\n\n Phone number is Invalid.";
    }
}

}

Module(delete booking)

void bus:: del_booking()
{
    fstream file,file1;

    int r_seats,found = 0;

    string name,no,phone,t_name,t_phone,del_no;//no-bus no

    file.open("student.txt",ios::in);

    if(!file)
    {
        cout<<"\n\n Please do the reservation to delete.";
    }
}

```



```
}  
  
else  
  
{  
  
    cout<<"\n\n Enter your phone number : ";  
  
    cin>>t_phone;  
  
  
    file>>name>>no>>phone>>r_seats;  
  
  
    while(file.eof())  
    {  
  
        if(t_phone == phone)  
        {  
  
            file.close();  
  
            file.open("student.txt",ios::in);  
  
            file1.open("student1.txt",ios::app|ios::out);  
  
            file>>name>>no>>phone>>r_seats;  
  
  
            while(!file.eof())  
            {  
  
                if(t_phone == phone) // To clear previous seats which were  
reserved in the previous bookings  
                {  
  
                    del_no = no;
```

```

    }

    if(t_phone != phone)
    {
        file1<<name<<" "<<no<<" "<<phone<<" "<<r_seats<<endl;
    }

    file>>name>>no>>phone>>r_seats;
}

file.close();

file1.close();

remove("student.txt");

rename("student1.txt","student.txt");

file.open("seat.txt",ios::in);

file1.open("seat1.txt",ios::app | ios:: out);

file>>no>>r_seats;

while(!file.eof())
{
    if(del_no != no )
    {
        file1<<no<<" "<<r_seats<<endl;
    }
}

```

```
        file>>no>>r_seats;

    }

    file.close();

    file1.close();

    remove("seat.txt");

    rename("seat1.txt","seat.txt");

    cout<<"\n\n Deleted Reserved Seats.";

    found++;

    goto h;

}

    file>>name>>no>>phone>>r_seats;

}

file.close();

h:

if(found == 0)

{

    cout<<"\n\n Phone number is Invalid.";

}

}

}
```

**Module(show booking)**

```
void bus:: show_booking()
{
    fstream file;

    int r_seats,found=0;

    string name,no,phone;

    file.open("student.txt",ios::in);

    if(!file)
    {
        cout<<"\n\n Please do the reservation to view.";
    }
    else
    {
        file>>name>>no>>phone>>r_seats;

        while(!file.eof())
        {
            cout<<"\n\n\n Student Name : "<<name;

            cout<<"\n\n Bus Number : "<<no;

            cout<<"\n\n Phone Number : "<<phone;

            cout<<"\n\n Reserved Seats : "<<r_seats;

            found++;
        }
    }
}
```

```
        file>>name>>no>>phone>>r_seats;

    }

    file.close();

    if(found == 0)

        cout<<"\n\n No Booking Record Found.";

    }

}
```

## OUTPUTS

THESE ARE THE MAIN MODULE OUTPUTS OF OUR PROJECT

```

SRM BUS RESERVATION SYSTEM
-----

*MAIN MENU*

1. INSTALL BUS DETAILS
2. DELETE BUS DETAILS
3. DISPLAY BUSES
4. AVAIBLE SEATS DETAILS
5. RESERVATION
6. UPDATE BOOKING RECORD
7. DELETE BOOKING RECORD
8. SHOW ALL BOOKING RECORD
9. EXIT
```

```

ENTER YOUR CHOICE : 1

Enter the bus number :v2

Enter the Driver name : john

Enter the arrival time : 7:15

Enter the departure time : 7:25

From : mangalagiri

To : srmuniv

Enter the total Seats : 30

Bus installed successfully

```

```

2. DELETE BUS DETAILS
3. DISPLAY BUSES
4. AVAIBLE SEATS DETAILS
5. RESERVATION
6. UPDATE BOOKING RECORD
7. DELETE BOOKING RECORD
8. SHOW ALL BOOKING RECORD
9. EXIT

ENTER YOUR CHOICE : 2
Buses Available to delete...

BUS NO.      DRIVER NAME  ARRIVAL TIME  DEPARTURE TIME  FROM          TO          NO. OF SEATS
v1           srinu       7:00         7:10          vijayawada    srmuniv     30
v2           john        7:15         7:25          mangalagiri   srmuniv     30
v3           ramu        7:30         7:40          guntur        srmuniv     30

Enter the bus number to be deleted : v2

Bus has been deleted.

```

```

3. DISPLAY BUSES

4. AVAIBLE SEATS DETAILS

5. RESERVATION

6. UPDATE BOOKING RECORD

7. DELETE BOOKING RECORD

8. SHOW ALL BOOKING RECORD

9. EXIT

ENTER YOUR CHOICE : 3

BUS NO.      DRIVER NAME  ARRIVAL TIME  DEPARTURE TIME  FROM          TO          NO. OF SEATS
v1           srinu       7:00         7:10          vijayawada    srmuniv     30
v3           ramu        7:30         7:40          guntur        srmuniv     30

```

```

ENTER YOUR CHOICE : 5

Enter the bus number : v1

Enter the Seat number : 15

Enter your name : phani

Enter your phone number : 8985923888
*****
*****

Booking Information
*****
*****

Student Name :          phani
Phone Number :          8985923888
Seat number :          15
Booking Succesful

```

```

ENTER YOUR CHOICE : 6

Enter your phone number : 8985923888

Enter the bus number : v1

Enter the Seat number : 15

Enter your name : phani

Enter your phone number : 898592388
*****
*****

Booking Information
*****
*****

Student Name :          phani
Phone Number :          898592388
Seat number :          15
Booking Succesful

Updated Reserved Seats

```

## CONCLUSION

In the conclusion of this project we had understood the oops concept and learned more about the classes and Inheritance. By this project the students can easily book their buses from online portal. At last we conclude this project student can easily book bus and we too learnt a lot regarding this programming language.





