

COLLEGE 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

Submitted by
Prasanna Kumar Dheekonda (AP21110010174)
Sahithya Chilukuri (AP21110010181)
Narasimha Yandrapragada (AP21110010190)
Sivamani Gopavarapu (AP21110010195)



Under the Guidance of
RAJIV SENAPATHI
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 “**COLLEGE BUS RESERVATION SYSTEM**” has been carried out by **Group-15(Prasanna Kumar, Sahithya, Narasimha Rao, Sivamani)** under my/our supervision. The work is genuine, original, and suitable for submission to the SRM University – AP for the award of Bachelor of Technology/Master of Technology in **School of Engineering and Sciences**.

Supervisor

(Signature)

Dr. Rajiv Senapati

Assistant Professor, CSE Department,

SRM UNIVERSITY, AP.

Co-supervisor

(Signature)

Venkaiah Chowdary B

Lab Assistant,

SRM UNIVERSITY, AP.

Acknowledgements

I'd like to thank my teacher, Rajiv Senapati, 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-15
CSE-C

Table of Contents

Certificate	3
Acknowledgements	5
Table of Contents	7
Abstract	9
Statement of Contributions	11
1. Introduction	13
2. Methodology	15
2.1 DESIGN	15
2.2 IMPLEMENTATION	16
3. Results	24
4. Concluding Remarks	32

Abstract

COLLEGE BUS RESERVATION SYSTEM deals with the maintenance of records of details of each student. It also includes maintenance of information like details of each bus. We can provide students to book transport , cancel transport and view status and to provide admins to show 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. This reservation system allows us to reserve a bus without having to visit the college in person. Our project goal is to manage all the information about bus, student transportation booking and to replace all the paperwork and provide better services to the user.

Statement of Contributions

Work contributed by team members:

Sahithya Chilukuri -Some contribution in code, Report

Prasanna Kumar Dheekonda - Some contribution in code, Report

Narasimha Yandrapragada - UML and Code

Sivamani Gopavarapu - UML and code

1.Introduction

In previous days, if you want to book bus transportation. We use to go to college and enroll in bus transportation. They used to give us application papers in which we entered the details . After a few days, it changed. They are directly reserving buses on the computer itself. Nowadays there is no use to go near college as we can directly book from our mobiles itself. This project is also the same as that. The objective of the project is to make users more accessible for booking bus and reduce paper work in college

This project demonstrates the implementation of a class and an object is the bus reservation system. The bus information is not stored using file handling in this bus reservation system. When it comes to features, the administrator can add bus information, remove the bus, and print all of the bus's details. Bus reservations, registration status, cancellations of bus registrations, and bus availability are all available to users. Booking a bus only requires a short amount of time.

A bus reservation system online makes it simple to book a bus and travel. This approach is simple to use, quick, and straightforward.

BUS RESERVATION SYSTEM HOLDS MAIN MENU INTERFACE

Here, from main menu we can access

- Student Menu Interface
- Admin Interface

Student menu contains 4 operations:-

- Book transport
- View status
- Cancel transport
- Exit

Admin menu contains 4 operations:-

- Add Bus
- Show Bus
- Remove Bus
- Logout

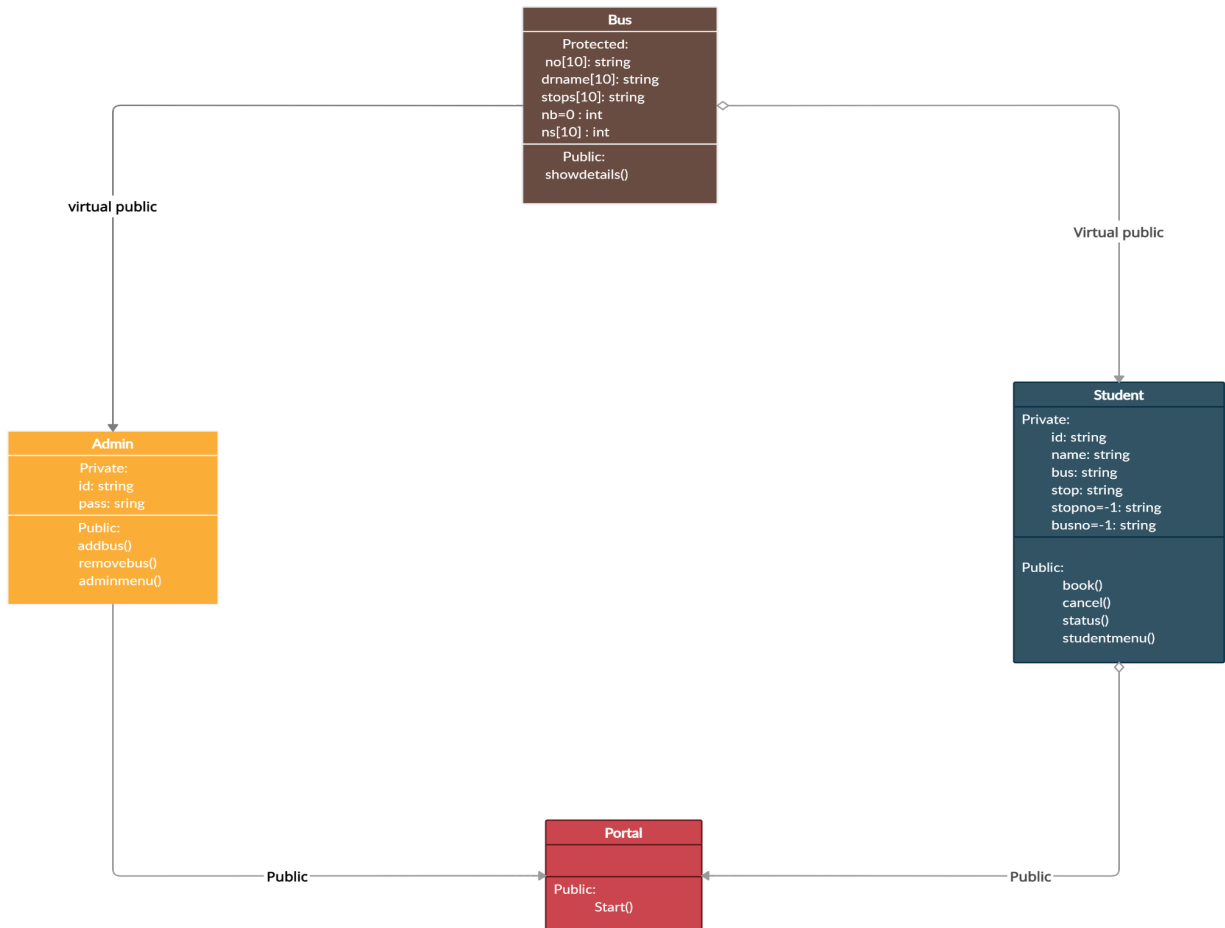
2.Methodology

2.1 DESIGN

The UML Class diagram is a graphical notation used to construct and visualize object-oriented systems. A class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system:

- Classes
- Their data members
- member functions (or methods)
- and the relationships among objects

Class diagram of our project:



2.2 IMPLEMENTATION

Class bus:-

```
#include<iostream>
#include<string>
#include<cstdlib>
#include<cstring>
using namespace std;

class Bus{
protected:
    string no[10], drname[10];
    string stops[10][10];
    int nb=0,ns[10];
public:
    void showdetails()
    {
        system("cls");
        cout<<" ----- BUS DETAILS ----- "<<endl;
        if(nb==0)
        {
            cout<<"                No Buses Available"<<endl;
            cout<<"-----"<<endl;
        }
        else
        {
            for(int i=0;i<nb;i++)
            {
                cout<<"-----"<<endl;
                cout<<" Bus "<<(i+1)<<endl;
                cout<<" Bus Code   : "<<no[i]<<endl;
                cout<<" Driver Name : "<<drname[i]<<endl;
                cout<<" Stops      : "<<endl;
                for(int j=0;j<ns[i];j++)
                {
                    cout<<(j+1)<<" . "<<stops[i][j]<<endl;
                }
                cout<<"-----"<<endl;
            }
        }
    }
};
```

This is one of the base classes used in our program. The data members of this class are all protected so that they can only be accessed by the class which is inheriting it. The data members are string no[10] used to store bus code, drname[10] stores driver name, and stop[10][10] is a multiple array which is used to store the number of stops for each bus. And nb(no of buses=0) and ns[10] are int data types which store the number of stops.

The member function **showdetails()** is publicly derived and can be accessed throughout the program. In this showdetails() function it prints bus details. Here we used an if-else statement to print bus details based on nb (number of buses). Here If nb=0, it prints "No Buses Available" and if nb value is other than zero then it executes the else statement.

Class admin:


```

class Admin:public virtual Bus
{
private:
    string id, pass;
public:
    void addbus()
    {
        system("cls");
        cout<<" ----- ADD BUS ----- "<<endl;
        cout<<" ----- "<<endl;
        cout<<"Enter bus code      : ";cin>>no[nb];
        cout<<"Enter driver name    : ";cin>>drname[nb];
        cout<<"Enter number of stops : ";cin>>ns[nb];
        for(int j=0;j<ns[nb];j++)
        {
            cout<<"Enter stop"<<(j+1)<<" : ";cin>>stops[nb][j];
        }
        nb++;
        return;
    }
    void removebus()
    {
        string key;
        int index=-1;
        system("cls");
        showdetails();
        cout<<"Which bus do you want to remove?\nEnter bus number : ";cin>>key;
        for(int i=0;i<nb;i++)
        {
            if(key==no[i])
            {
                index=i;
            }
        }
        if(index==--1)
        {
            cout<<"BUS NOT FOUND!"<<endl;
            return;
        }
        else
        {
            for(int i=index;i<nb;i++)
            {

```

```

                else
                {
                    for(int i=index;i<nb;i++)
                    {
                        no[i]=no[i+1];
                        drname[i]=drname[i+1];
                        ns[i]=ns[i+1];
                        for(int j=0;j<ns[i];j++)
                        {
                            stops[i][j]=stops[i+1][j];
                        }
                    }
                    nb--;
                }
            }
            return;
        }
    }
    void adminmenu()
    {
        system("cls");
        adminlogin;
        cout<<" ----- ADMIN MENU ----- "<<endl;
        cout<<" ----- "<<endl;
        cout<<"      Login to continue"<<endl;
        cout<<"      Enter your ID   : ";cin>>id;
        cout<<"      Enter password  : ";cin>>pass;
        if(id=="admin" & pass=="admin")
        {
            int ch;
            system("cls");
            cout<<"      LOGIN SUCCESSFUL!      "<<endl;
            while(1)
            {
                cout<<" ----- ADMIN MENU ----- "<<endl;
                cout<<" ----- "<<endl;
                cout<<"      1. Add Bus      "<<endl;
                cout<<"      2. Remove Bus   "<<endl;
                cout<<"      3. Show Buses   "<<endl;
                cout<<"      4. Logout      "<<endl;
                cout<<" ----- "<<endl;
                cout<<"      Enter your choice : ";cin>>ch;
                switch(ch)
                {
                    case 1:

```

```

switch(ch)
{
    case 1:
        addbus();
        break;
    case 2:
        removebus();
        break;
    case 3:
        showdetails();
        break;
    case 4:
        system("cls");
        cout<<"LOGGED OUT"<<endl;
        return;
        break;
    default:
        system("cls");
        cout<<"Invalid Choice, Try Again!"<<endl;
}
}
else
{
    cout<<"Incorrect Credentials"<<endl;
    goto adminlogin;
}
};

```

Admin class is a base class which is public,virtual derived from a base class named bus.

ID and Pass are the private data members of the class which is used to login for admin menu.We can login into admin menu only if ID and password are correct.

Member function:-

void adminmenu():-

If password and ID are correct ,the admin menu will be displayed.From the Admin menu using switch case we can select one option.If password and ID are incorrect it will print "Incorrect Credentials" and go to the adminlogin.

If the choice is 3, then it will call showdetails function which is a member function of a base class.

void addbus():-

This member function is used to add the bus .The administration should add the bus so that students can see the bus details and book the bus accordingly. So,here the administrator will add bus details like bus code,bus driver name,Number of stops for each bus.

void removebus():-

This member function is used to remove the particular bus in the list.In this function,

String key and int index=-1 are variables of this function. Here showdetails() function is called and details are printed. Admin will enter the bus number of the bus which they want to remove and it is stored as key. If the key is found then index value is updated to i(index of array of bus). If index=-1 then it prints "No bus found".

Student Class:

```
class Student:public virtual Bus
{
    private:
        string id, name, bus, stop;
        int stopno=-1, busno=-1;
    public:
        void book()
        {
            showdetails();
            cout<<" ----- BOOK BUS ----- "<<endl;
            cout<<"-----" <<endl;
            cout<<"Enter student ID      : ";cin>>id;
            cout<<"Enter student name      : ";cin>>name;
            cout<<"Enter Bus Number        : ";cin>>busno;
            cout<<"Enter Stop Number       : ";cin>>stopno;
            bus=no[busno-1];
            stop=stops[busno-1][stopno-1];
        }
        void cancel()
        {
            cout<<" ----- CANCEL BUS ----- "<<endl;
            cout<<"-----" <<endl;
            id="Not registered";
            name="Not registered";
            busno=-1;
            stopno=-1;
        }
        void status()
        {
            cout<<" ----- REGISTRATION STATUS ----- "<<endl;
            if(busno== -1)
```

```

void status()
{
    cout<<" ----- REGISTRATION STATUS ----- "<<endl;
    if(busno==-1)
    {
        cout<<"          NOT REGISTERED"<<endl;
    }
    else
    {
        cout<<"          Student ID   : "<<id<<endl;
        cout<<"          Student Name  : "<<name<<endl;
        cout<<"          Bus Code     : "<<bus<<endl;
        cout<<"          Bus Stop     : "<<stop<<endl;
    }
}

void studentmenu()
{
    int ch;
    while(1)
    {
        cout<<" ----- STUDENT MENU ----- "<<endl;
        cout<<"-----"<<endl;
        cout<<"          1. Book Transport      "<<endl;
        cout<<"          2. Cancel Transport    "<<endl;
        cout<<"          3. View Status         "<<endl;
        cout<<"          4. Exit                "<<endl;
        cout<<"-----"<<endl;
        cout<<"          Enter your choice : ";cin>>ch;
        switch(ch)
        {

```

```

            case 1:
                system("cls");
                book();
                break;
            case 2:
                system("cls");
                cancel();
                break;
            case 3:
                system("cls");
                status();
                break;
            case 4:
                cout<<"EXITED"<<endl;
                return;
                break;
            default:
                system("cls");
                cout<<"Invalid Choice, Try Again!"<<endl;
        }
    }
}

};

```

Student class is a base class which is public,virtual derived from a base class named bus.

Here this class has private data members: string id,name,bus,stop which is used to store student information like id,name, bus number, stop number. And stopno, busno which is initialized to -1 of int type

Member function:-

void book():-

Here the showdetails() function is called and details are printed. From the details, students can book the bus by entering the details.

void cancel():-

If we choose option 2 then it goes to void cancel(). Then Id and name will be updated to “not registered” and bus number and stop number will be updated to -1.

void status():-

If busno=-1 then it will print “Not registered” else it will print details like student Id, name, bus code, bus stop.

void studentmenu():-

From the Student menu using the switch case we can select any option and according to the option the function is called.

Class portal:

```
class Portal:public Admin, public Student
{
public:
    void start()
    {
        int ch;
        while(1)
        {
            cout<<endl;
            system("cls");
            cout<<"----- MAIN MENU ----- "<<endl;
            cout<<"----- "<<endl;
            cout<<"          1. Login as Admin          "<<endl;
            cout<<"          2. Login as Student        "<<endl;
            cout<<"          3. Exit                    "<<endl;
            cout<<"----- "<<endl;
            cout<<"          Enter your choice : ";cin>>ch;
            switch(ch)
            {
                case 1:
                    system("cls");
                    adminmenu();
                    break;
                case 2:
                    system("cls");
                    studentmenu();
                    break;
                case 3:
                    cout<<"EXITED"<<endl;
                    return;
                    break;
                default:
                    system("cls");
                    cout<<"Invalid Choice, Try Again!"<<endl;
            }
        }
    }
};
```

This class is publicly derived from two base classes admin and student .It has only one member function void start()

Void start():-

It prints the details of Main Menu and asks the user to enter the choice.Using the switch case function is called according to selected choice.

main() function:-

```
int main()
{
    Portal p;
    p.start();
    return 0;
}
```

Here we created an object p of class portal. Here we call the start() function of the class portal using Object p.The execution of the program starts from the int main().

3. RESULTS:

Output :-

```
----- MAIN MENU -----  
-----  
1. Login as Admin  
2. Login as Student  
3. Exit  
-----  
Enter your choice : 1_
```

```
----- ADMIN MENU -----  
-----  
Login to continue  
Enter your ID : admin  
Enter password : admin
```

```
LOGIN SUCCESSFUL!  
----- ADMIN MENU -----  
-----  
1. Add Bus  
2. Remove Bus  
3. Show Buses  
4. Logout  
-----  
Enter your choice : 1
```



```
----- ADD BUS -----  
-----  
Enter bus code      : G1  
Enter driver name   : sunil  
Enter number of stops : 2  
Enter stop1        : Guntur  
Enter stop2        : gardens
```

```
----- ADD BUS -----  
-----  
Enter bus code      : V14  
Enter driver name   : sathvik  
Enter number of stops : 2  
Enter stop1        : PVP  
Enter stop2        : Guntur  
  
----- ADMIN MENU -----  
-----  
1. Add Bus  
2. Remove Bus  
3. Show Buses  
4. Logout  
-----  
Enter your choice : 3_
```

```
----- BUS DETAILS -----  
-----  
Bus 1  
Bus Code      : G1  
Driver Name   : sunil  
Stops         :  
1. Guntur  
2. gardens  
-----  
  
Bus 2  
Bus Code      : V14  
Driver Name   : sathvik  
Stops         :  
1. PVP  
2. Guntur  
-----  
  
----- ADMIN MENU -----  
-----  
1. Add Bus  
2. Remove Bus  
3. Show Buses  
4. Logout  
-----  
  
Enter your choice : 4_
```

```
----- MAIN MENU -----  
-----  
1. Login as Admin  
2. Login as Student  
3. Exit  
-----  
  
Enter your choice : 2_
```

```
----- STUDENT MENU -----  
-----  
1. Book Transport  
2. Cancel Transport  
3. View Status  
4. Exit  
-----  
Enter your choice : 1
```

```
----- BUS DETAILS -----  
-----  
Bus 1  
Bus Code : G1  
Driver Name : sunil  
Stops :  
1. Guntur  
2. gardens  
-----  
Bus 2  
Bus Code : V14  
Driver Name : sathvik  
Stops :  
1. PVP  
2. Guntur  
-----  
----- BOOK BUS -----  
-----  
Enter student ID : AP21110010181  
Enter student name : Sahithya  
Enter Bus Number : 1  
Enter Stop Number : 1  
-----  
----- STUDENT MENU -----  
-----  
1. Book Transport  
2. Cancel Transport  
3. View Status  
4. Exit  
-----  
Enter your choice : 3
```

```
----- REGISTRATION STATUS -----  
-----  
Student ID : AP21110010181  
Student Name : Sahithya  
Bus Code : G1  
Bus Stop : Guntur  
-----  
----- STUDENT MENU -----  
-----  
1. Book Transport  
2. Cancel Transport  
3. View Status  
4. Exit  
-----  
Enter your choice : 2_
```

```
----- CANCEL BUS -----  
-----  
  
----- STUDENT MENU -----  
-----  
1. Book Transport  
2. Cancel Transport  
3. View Status  
4. Exit  
-----  
  
Enter your choice : 4_
```

```
----- MAIN MENU -----  
-----  
1. Login as Admin  
2. Login as Student  
3. Exit  
-----  
  
Enter your choice : 1_
```

```
----- ADMIN MENU -----  
-----  
Login to continue  
Enter your ID : admin  
Enter password : admin
```

C:\Users\sunil\Downloads\bus (2).exe

LOGIN SUCCESSFUL!

----- ADMIN MENU -----

- 1. Add Bus
- 2. Remove Bus
- 3. Show Buses
- 4. Logout

Enter your choice : 2_

----- BUS DETAILS -----

Bus 1
Bus Code : G1
Driver Name : sunil
Stops :
1. Guntur
2. gardens

Bus 2
Bus Code : V14
Driver Name : sathvik
Stops :
1. PVP
2. Guntur

Which bus do you want to remove?
Enter bus number : G1

----- ADMIN MENU -----

- 1. Add Bus
- 2. Remove Bus
- 3. Show Buses
- 4. Logout

Enter your choice : 3_

----- BUS DETAILS -----

Bus 1
Bus Code : V14
Driver Name : sathvik
Stops :
1. PVP
2. Guntur

----- ADMIN MENU -----

- 1. Add Bus
- 2. Remove Bus
- 3. Show Buses
- 4. Logout

Enter your choice : 4_

```
----- MAIN MENU -----  
-----  
      1. Login as Admin  
      2. Login as Student  
      3. Exit  
-----  
      Enter your choice : 3  
EXITED  
-----  
Process exited after 684.2 seconds with return value 0  
Press any key to continue . . . █
```


4. Concluding Remarks:

In the process of completing this project, we have understood the oops concept and learned more about classes and inheritance. By this project the students can easily book the bus from mobile. At last, we conclude that due to this project user can easily book bus and we too learnt a lot regarding this programming language