

# PROJECT REPORT

on

## **BUS RESERVATION SYSTEM**

Submitted by

**Rohan Habu (1032191005)**

**Rohil Nataraj (1032191016)**

**Vishal Singh (1032191177)**

in

**Object Oriented Programming**

**SY B. TECH**

**Under the Guidance of**

**(Mrs. Mrunal Annadate)**

**School of Electronics & Communication Engineering**

**Dr. Vishwanath Karad**

**MIT WORLD PEACE UNIVERSITY, PUNE.**

**[2020-2021]**

## Table of Contents

<b>Acknowledgement</b>	
<b>List of Tables</b>	
<b>CH. 1</b>	<b>Introduction.....</b>
<b>1.1</b>	<b>Introduction.....</b>
<b>1.2</b>	<b>Aim and Objectives.....</b>
<b>CH. 2</b>	<b>Methodology</b>
<b>2.1</b>	<b>Problem statement</b>
<b>2.2</b>	<b>System requirements</b>
<b>2.3</b>	<b>Class Diagram of the System</b>
<b>CH. 3</b>	<b>Results</b>
<b>Ch. 4</b>	<b>Conclusion</b>
	<b>References</b>

## ACKNOWLEDGEMENT

Our sincere thanks to our teacher Mrs. Mrunal Annadate for guiding us in the process of project making. We have learnt many new things throughout the making of this OOP project. We wish to continue learning and creating new applications. During this project, we have experienced many formatting errors due to very small mistakes, though due to this our focus and patience to bear the errors and remain calm has drastically increased and thus it has also improved our productivity in normal studies.

## LIST OF TABLES

Table name	Description
Table 1	Storing bus records.
Table 2	Storing customer records.
Table 3	Storing login records.

## 1. Table 1

File=" bus.txt"

Variables	Datatype
number	string
name	string
seats	int
driverName	string
freeseats	int
dp	string
ds	string

## 1. Table 2

File=" cus.txt"

Variables	Datatype
name	string
number	string
seat	string
id	string
contact	string

## 2. Table 3

File="login.txt"

Variables	Datatype
userName	string
pass	string

## **CHAPTER 1**

### **1.1 Introduction**

This is a bus ticket booking system that stores the bus data on the hard drive.

It takes user input for storing details of user and reserving the seat for the user.

There is an admin login option to change some parameters related to the buses and view all the customer details from database.

Admin login is specific and no one can create new account unless it is mentioned by the programmer in the code file.

After successful booking of the ticket by the user, there is a reference number provided which is used to keep track of the user.

Also, the user can cancel the ticket and thus the reference number associated with that ticket is deleted from database.

There is a facility of adding new buses if in case the company buys a new bus and wants to operate on this app.

Also, the admin can edit the existing bus details like the driver's name and number of seats.

### **1.2 Aim and Objectives of Project**

To implement concept of class in OOP.

To make use of the member functions with use of class object.

To store and access the data on hard drive using file handling.

## CHAPTER 2

### Methodology

For this system there is a class created named 'Bus'.

In the class there are all the functions related to the bus reservation system.

So, in the main function we create an object of the class 'Bus' and then call it's member function named 'Start' from which the code execution starts and everything is initialized.

### 2.1 Problem statement

To create a bus reservation system using objects and file handling algorithm in C++.

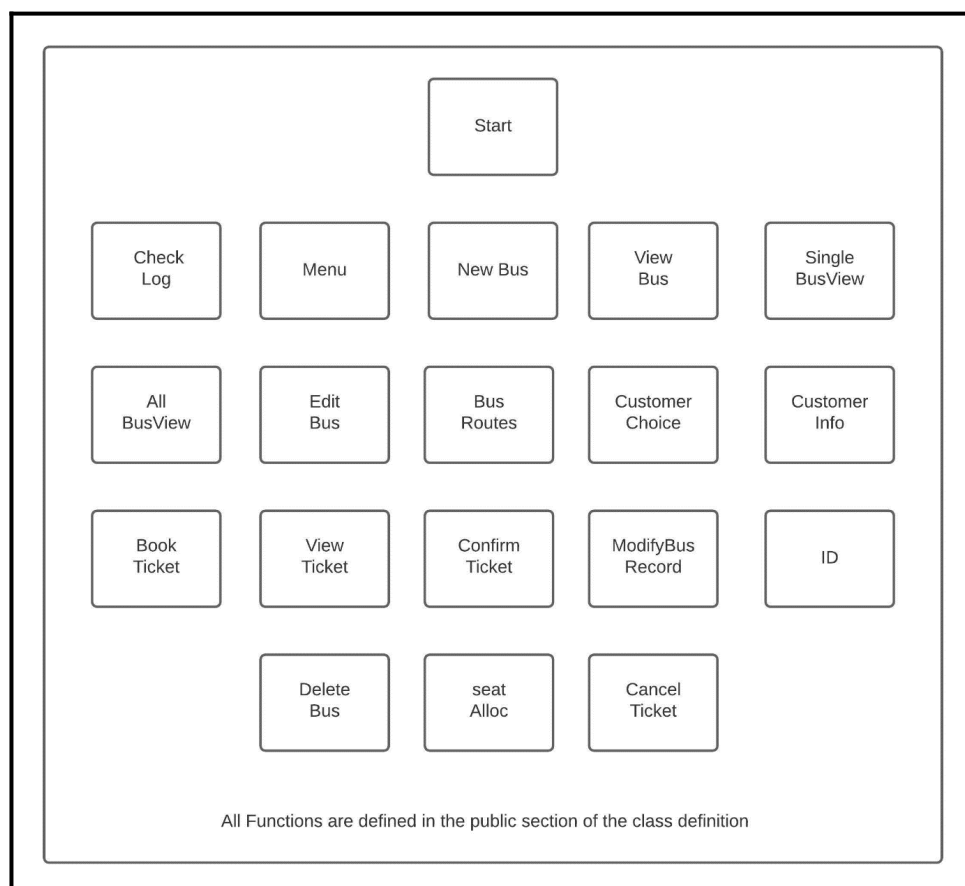
### 2.2 System requirements

Software used is Visual Studio Code.

Input is taken from user and the output is displayed in the terminal window.

There are some libraries required other than 'iostream' to execute the file handling statements and some windows function.

### 2.3 Class diagram of System



### 3. Results

## Starting of the application

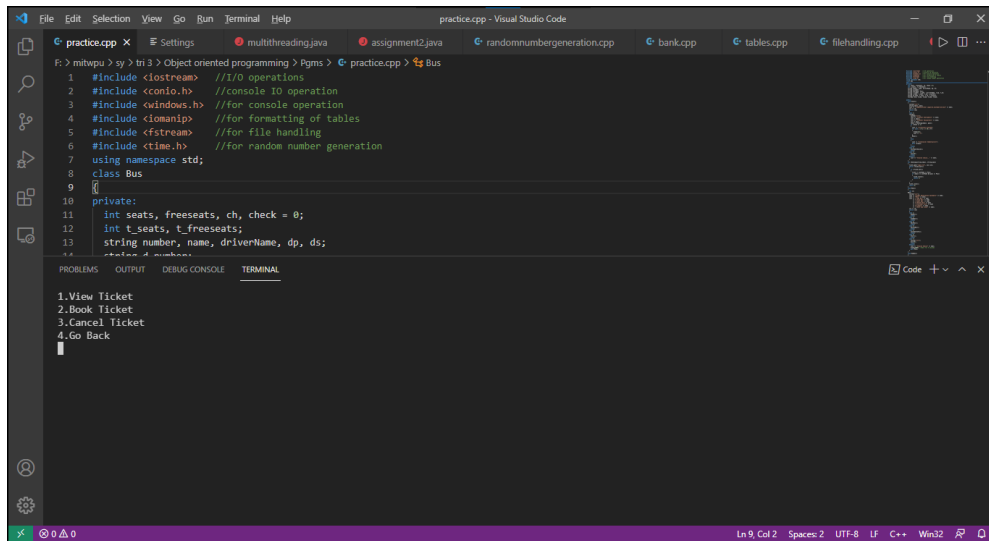
The screenshot shows the Visual Studio Code interface with a C++ file named 'practice.cpp' open. The code defines a 'Bus' class with various attributes and methods. The 'Terminal' tab is active, displaying the output of a program execution, which shows a menu with options: '1.Administrator Login', '2.Customer', and '3.Exit'.

After successful login of admin

```

1 #include <iostream> //I/O operations
2 #include <conio.h> //console I/O operation
3 #include <windows.h> //for console operation
4 #include <iomanip> //for formatting of tables
5 #include <fstream> //for file handling
6 #include <time.h> //for random number generation
7 using namespace std;
8 class Bus
9 {
10 private:
11     int seats, freeseats, ch, check = 0;
12     int t_seats, t_freeseats;
13     string number, name, driverName, dp, ds;
14     static Bus* bus;
15 public:
16     Bus() {}
17     ~Bus() {}
18     static Bus* getInstance()
19     {
20         if (bus == nullptr)
21             bus = new Bus();
22         return bus;
23     }
24     void reservationSystem()
25     {
26         cout << "*****Bus Reservation System*****" << endl;
27         cout << "Menu" << endl;
28         cout << "1.Add Bus" << endl;
29         cout << "2.View Bus" << endl;
30         cout << "3.Edit Bus" << endl;
31         cout << "4.Delete Bus" << endl;
32         cout << "5.Customer Info" << endl;
33         cout << "6.Logout" << endl;
34         cout << "7.Exit And Close" << endl;
35         int choice;
36         do
37         {
38             choice = 0;
39             while (choice < 1 || choice > 7)
40             {
41                 cout << "Enter your choice: ";
42                 choice = getch();
43             }
44             switch (choice)
45             {
46                 case 1:
47                     addBus();
48                     break;
49                 case 2:
50                     viewBus();
51                     break;
52                 case 3:
53                     editBus();
54                     break;
55                 case 4:
56                     deleteBus();
57                     break;
58                 case 5:
59                     customerInfo();
60                     break;
61                 case 6:
62                     logout();
63                     break;
64                 case 7:
65                     exit(0);
66                     break;
67             }
68             cout << "Press any key to continue: ";
69             getch();
70         } while (choice != 7);
71     }
72     void addBus()
73     {
74         cout << "Add Bus" << endl;
75         cout << "Enter number: ";
76         string number;
77         while (true)
78         {
79             number = getNumber();
80             if (number.length() != 10)
81                 continue;
82             if (isNumber(number))
83                 continue;
84             break;
85         }
86         cout << "Enter name: ";
87         string name;
88         while (true)
89         {
90             name = getName();
91             if (name.length() < 3)
92                 continue;
93             if (isNumber(name))
94                 continue;
95             break;
96         }
97         cout << "Enter driverName: ";
98         string driverName;
99         while (true)
100         {
101             driverName = getDriverName();
102             if (driverName.length() < 3)
103                 continue;
104             if (isNumber(driverName))
105                 continue;
106             break;
107         }
108         cout << "Enter dp: ";
109         string dp;
110         while (true)
111         {
112             dp = getDate();
113             if (dp.length() != 10)
114                 continue;
115             if (isNumber(dp))
116                 continue;
117             break;
118         }
119         cout << "Enter ds: ";
120         string ds;
121         while (true)
122         {
123             ds = getTime();
124             if (ds.length() != 10)
125                 continue;
126             if (isNumber(ds))
127                 continue;
128             break;
129         }
130         cout << "Enter seats: ";
131         int seats;
132         while (true)
133         {
134             seats = getSeats();
135             if (seats < 10 || seats > 100)
136                 continue;
137             break;
138         }
139         cout << "Enter freeseats: ";
140         int freeseats;
141         while (true)
142         {
143             freeseats = getFreeseats();
144             if (freeseats < 10 || freeseats > 100)
145                 continue;
146             break;
147         }
148         cout << "Enter check: ";
149         int check;
150         while (true)
151         {
152             check = getCheck();
153             if (check < 10 || check > 100)
154                 continue;
155             break;
156         }
157         cout << "Enter t_seats: ";
158         int t_seats;
159         while (true)
160         {
161             t_seats = getT_seats();
162             if (t_seats < 10 || t_seats > 100)
163                 continue;
164             break;
165         }
166         cout << "Enter t_freeseats: ";
167         int t_freeseats;
168         while (true)
169         {
170             t_freeseats = getT_freeseats();
171             if (t_freeseats < 10 || t_freeseats > 100)
172                 continue;
173             break;
174         }
175         cout << "Enter number, name, driverName, dp, ds, seats, freeseats, check: ";
176         cout << endl;
177         cout << "Enter number: ";
178         string number;
179         while (true)
180         {
181             number = getNumber();
182             if (number.length() != 10)
183                 continue;
184             if (isNumber(number))
185                 continue;
186             break;
187         }
188         cout << "Enter name: ";
189         string name;
190         while (true)
191         {
192             name = getName();
193             if (name.length() < 3)
194                 continue;
195             if (isNumber(name))
196                 continue;
197             break;
198         }
199         cout << "Enter driverName: ";
200         string driverName;
201         while (true)
202         {
203             driverName = getDriverName();
204             if (driverName.length() < 3)
205                 continue;
206             if (isNumber(driverName))
207                 continue;
208             break;
209         }
210         cout << "Enter dp: ";
211         string dp;
212         while (true)
213         {
214             dp = getDate();
215             if (dp.length() != 10)
216                 continue;
217             if (isNumber(dp))
218                 continue;
219             break;
220         }
211         cout << "Enter ds: ";
222         string ds;
223         while (true)
224         {
225             ds = getTime();
226             if (ds.length() != 10)
227                 continue;
228             if (isNumber(ds))
229                 continue;
230             break;
231         }
232         cout << "Enter seats: ";
233         int seats;
234         while (true)
235         {
236             seats = getSeats();
237             if (seats < 10 || seats > 100)
238                 continue;
239             break;
240         }
241         cout << "Enter freeseats: ";
242         int freeseats;
243         while (true)
244         {
245             freeseats = getFreeseats();
246             if (freeseats < 10 || freeseats > 100)
247                 continue;
248             break;
249         }
250         cout << "Enter check: ";
251         int check;
252         while (true)
253         {
254             check = getCheck();
255             if (check < 10 || check > 100)
256                 continue;
257             break;
258         }
259         cout << "Enter t_seats: ";
260         int t_seats;
261         while (true)
262         {
263             t_seats = getT_seats();
264             if (t_seats < 10 || t_seats > 100)
265                 continue;
266             break;
267         }
268         cout << "Enter t_freeseats: ";
269         int t_freeseats;
270         while (true)
271         {
272             t_freeseats = getT_freeseats();
273             if (t_freeseats < 10 || t_freeseats > 100)
274                 continue;
275             break;
276         }
277         cout << "Enter number, name, driverName, dp, ds, seats, freeseats, check: ";
278         cout << endl;
279         cout << "Enter number: ";
280         string number;
281         while (true)
282         {
283             number = getNumber();
284             if (number.length() != 10)
285                 continue;
286             if (isNumber(number))
287                 continue;
288             break;
289         }
290         cout << "Enter name: ";
291         string name;
292         while (true)
293         {
294             name = getName();
295             if (name.length() < 3)
296                 continue;
297             if (isNumber(name))
298                 continue;
299             break;
300         }
301         cout << "Enter driverName: ";
302         string driverName;
303         while (true)
304         {
305             driverName = getDriverName();
306             if (driverName.length() < 3)
307                 continue;
308             if (isNumber(driverName))
309                 continue;
310             break;
311         }
312         cout << "Enter dp: ";
313         string dp;
314         while (true)
315         {
316             dp = getDate();
317             if (dp.length() != 10)
318                 continue;
319             if (isNumber(dp))
320                 continue;
321             break;
322         }
323         cout << "Enter ds: ";
324         string ds;
325         while (true)
326         {
327             ds = getTime();
328             if (ds.length() != 10)
329                 continue;
330             if (isNumber(ds))
331                 continue;
332             break;
333         }
334         cout << "Enter seats: ";
335         int seats;
336         while (true)
337         {
338             seats = getSeats();
339             if (seats < 10 || seats > 100)
340                 continue;
341             break;
342         }
343         cout << "Enter freeseats: ";
344         int freeseats;
345         while (true)
346         {
347             freeseats = getFreeseats();
348             if (freeseats < 10 || freeseats > 100)
349                 continue;
350             break;
351         }
352         cout << "Enter check: ";
353         int check;
354         while (true)
355         {
356             check = getCheck();
357             if (check < 10 || check > 100)
358                 continue;
359             break;
360         }
361         cout << "Enter t_seats: ";
362         int t_seats;
363         while (true)
364         {
365             t_seats = getT_seats();
366             if (t_seats < 10 || t_seats > 100)
367                 continue;
368             break;
369         }
370         cout << "Enter t_freeseats: ";
371         int t_freeseats;
372         while (true)
373         {
374             t_freeseats = getT_freeseats();
375             if (t_freeseats < 10 || t_freeseats > 100)
376                 continue;
377             break;
378         }
379         cout << "Enter number, name, driverName, dp, ds, seats, freeseats, check: ";
380         cout << endl;
381         cout << "Enter number: ";
382         string number;
383         while
```

After choosing customer option from 1<sup>st</sup> screen



```
File Edit Selection View Go Run Terminal Help
practice.cpp - Visual Studio Code
R: > mitwpu > sy > tn 3 > Object oriented programming > Pgms > practice.cpp > Bus
1 #include <iostream> //I/O operations
2 #include <conio.h> //console IO operation
3 #include <windows.h> //for console operation
4 #include <iomanip> //for formatting of tables
5 #include <fstream> //for file handling
6 #include <time.h> //for random number generation
7 using namespace std;
8 class Bus
9 {
10 private:
11     int seats, freeseats, ch, check = 0;
12     int t_seats, t_freeseats;
13     string number, name, driverName, dp, ds;
14     static Bus * buses;
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

1.View Ticket  
2.Book Ticket  
3.Cancel Ticket  
4.Go Back

Ln 9, Col 2 Spaces: 2 UTF-8 LF C++ Win32



## CHAPTER 4

### 4. Conclusion

Bus reservation system using C++ was implemented successfully using file handling features to manage the data of the app.

#### Learning outcome

The major thing we learned from this project was the concept and advantages of using file handling in development of an application.

Also, we learned the approach to use for development of any application and the parameters to consider for successful implementation.

We learned memory management and how to make its use efficient and avoid duplication of data in the database.

#### References

<https://www.javatpoint.com/>

<https://www.geeksforgeeks.org/>

<https://www.tutorialspoint.com/index.htm>

<https://www.learncpp.com/>