VISVESVARAYA TECHNOLOGICAL UNIVERSITY

"Jnana Sangama", Belgavi, Karnataka -590014

A

Mini Project Report

on

"BUS MANAGEMENT SYSTEM"

Submitted in partial fulfillment of the requirements for the course FILE STRUCTURES LABORATORY WITH MINI PROJECT [18ISL67] of sixth semester of

Bachelor of Engineering in Information Science

2021-2022

Submitted by

N KRUTHIKA	**[4MH19IS054]**
NAGARJUN B P	**[4MH19IS055]**
NIDARSHAN N	**[4MH19IS126]**
RAKSHITH B	**[4MH19IS078]**

2021-2022

DEPARTMENT OF INFORMATION SCIENCE & ENGINEERING

MAHARAJA INSTITUTE OF TECHNOLOGY MYSORE BELAVADI, S.R.

PATNA TALUK, MANDYA DIST-571477

DEPARTMENT OF INFORMATION SCIENCE & ENGINEERING

MAHARAJA INSTITUTE OF TECHNOLOGY MYSORE

Belawadi, S.R. Patna (T), Mandya (D) – 571477.

CERTIFICATE

This is to certify that the mini project work entitled "**BUS MANAGEMENT SYSTEM"** is a bonafide work carried out by **N KRUTHIKA [4MH19IS054], NAGARJUN B P [4MH19IS055], NIDARSHAN N [4MH19IS126]** and **RAKSHITH B [4MH19IS078]** in partial fulfillment for **the FILE STRUCTURES Laboratory with Mini Project (18ISL67)** prescribed **by Visvesvaraya Technological University, Belagavi** during the year 2021-2022 for the sixth semester B.E in **Information Science and Engineering** The mini project report has been approved as it satisfies academic requirements**.**

Signature of Guide
(Prof. DHARMARAJ KB)
Asst. Professor,
Dept. of IS&E

Signature of HOD
(Dr. SHARATH KUMR YH)
Professor & Head,
Dept. of IS&E

Name of the Examiner	External Viva	Signature with Dates

ACKNOWLEDGEMENT

We sincerely owe our gratitude to all the persons who helped and guided us in completing this mini project work.

We are thankful to **Dr. B G NARESH KUMAR, Principal, Maharaja Institute of Technology Mysore**, for having supported us in our academic endeavors.

We are extremely thankful to **Dr. SHARATH KUMAR Y H, Professor &Head, Department of Information Science and Engineering,** for his valuable support and his timely inquiries into the progress of the work.

We are greatly indebted to our guide prof. **DHARMARAJ K B Asst. Professor, Department of Information Science and Engineering,** for the consistent co-operation and support.

We are obliged to all **teaching and non-teaching staff** members of **Department of Information Science and Engineering**, for the valuable information provided by them in their respective field's. We are grateful for their co-operation during the period of our mini project.

N KRUTHIKA	**[4MH19IS054]**
NAGARJUN B P	**[4MH19IS055]**
NIDARSHAN N	**[4MH19IS126]**
RAKSHITH B	**[4MH19IS078]**

ABSTRACT

Bus Management System is an integrated electronics management system. This system ensure that the management process of this can company are smoothly done. This system will also help in decreasing error mostly cause by human mistake. This system was developed as one of the solution to transfer from current manual practices to a more systematic computerized system. This system will be developed by using Microsoft Visual Basic and Microsoft Access. Briefly there are two main modules for this system. The first module will be the Driver management module and the second module will be bus management module. However this thesis will just describe about Bus management module. In this module, the function that will be developed include, bus registration, maintenance record, and remainder.

INDEX

1. INTRODUCTION

1.1 Overview 01

1.2 Problem Statement 01

1.3 Solution 01

1.4 Existing System 02

1.5 Proposed System 02

1.6 Advantages 03

2. SOFTWARE REQUIREMENTS SPECIFICATION

2.1 System Requirements 04

2.2 Software Description 04

3. FILE OPERATION

3.1 File Structure 05

3.1.1 Primary Index 05

3.2. Testing 06

3.2.1 Types of Testing 06

3.2.2 Unit Testing 06

3.2.3 Integration Testing 06

3.2.4 System Testing 07

3.2.5 Test Case 07

3.2.6 Unit Testing 07

3.3.7 Code 08

SNAPSHOTS **15**

CONCLUSION **18**

REFERENCES **19**

Chapter 1

INTRODUCTION

Overview

The main objective of this Bus management System is to store the bus details in a file. This, application stores all the details like bus number, driver name, from address and destination in a file. This system was developed to reduce the errors that creep up in manual systems. It was very difficult to store all the details manually there such a system was developed. This software also allows to book the tickets for various passengers. It is secure, easy to use and reliable software system. It also provides a good level of security as there is an admin who can only edit and update details. Earlier where there was only a manual system which required a lot of paperwork but now an automated system is developed which meet all the requirements.

Problem Statement

Bus management System is to automate the existing manual system by the help of computerized equipment and full-fledged computer software, so that their valuable data/information can be stored for a longer period with easy accessing and manipulation of the same. By wasting user’s time entering each bus number, driver name, from address and destination of the people in a particular area. This digitalized system can be a paperless system to provide much information.

Solution

The Bus System allows storing bus details in a file which is handled by the administrator. So, this provides a secure system which could easily book a ticket for a passenger any records. It also allows to easily update the bus details without modifying much. The records can be modified only by the administrator. In this, you can easily log in and view the bus details. This reduces dependence on the manual system which earlier required a lot of paperwork. This also saves time and the cost of paper. All the records are safely stored in a file using C++.

Existing System

Earlier there was a manual system which requires a lot of paperwork and required a lot of time to maintain the bus records in a file. The records stored were also not secured as anyone can view it. So, there was a risk of storing records. Besides this, if any changes were to be made then all things need to be updated which was very difficult to do. There was also a risk for loss of information in this system. This also led to errors so there was a need for the new system to be developed. Customer need more convenient way to get their required results in proper time. In today’s era where technology plays a vital role in each person life, they are bound to use the old-fashioned manual book to search any particular record. Even the publishers have to print the new records in their new volume and published each year. Customers always gets irritated to pay amount for new volume in each year. Even searching process of particular record is also one big problem for the customers, they have to first go through the index page to details on alphabetical letter and after that they have to go through the various pages using page number to get their desired results which also a time- consuming process.

Drawbacks

- Time Consuming
- Complex process
- Cost is high
- Can’t cover from losses

Proposed System

The Bus System allows storing bus details in a file which is handled by the administrator. So, this provides a secure system which could easily search, edit or update any records. It also allows to easily update the bus details without modifying much. The records can be modified only by the administrator.

In this, you can easily log in and view the bus details. This reduces dependence on the manual system which earlier required a lot of paperwork. This also saves time and the cost of paper. All the records are safely stored in a file using C++.

Advantages

- File application for customers
- Data is secure
- Recovery from losses is easy
- Save time and money

Chapter 2

SOFTWARE REQUIREMENT SPECIFICATIONS

2.1 System Requirements

Hardware Requirements:

PROCESSOR: Pentium 3 and higher (1.2GHZ and higher)

RAM: 4GB and higher

HARD DISK: 20GB

Software Requirements:

Operating system: Windows XP/7.

Coding Language: C++

2.2.1 Software Description

C++ (Language used):

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

C++ has also been found useful in many other contexts, with key strengths being software infrastructure and resource-constrained application, including desktop applications, servers (example: e-commerce, web search or C++ servers), and performance-critical applications (e.g.: telephone switches or space probes.). C++ is a complied language, with implementations of it available on many platforms. Many vendors provide C++ compilers, including the free Software Foundation, Microsoft, Intel and IBM.

Chapter 3

FILE OPERATIONS

File Structure

Operation	Input	Output
Insertion	When we want to insert a new bus.	All the buss are inserted.
Deletion	When we don't want that particular bus details, can delete it.	The corresponding bus details will be deleted.
Search	When you are searching for particular bus.	It allows to search for the bus details by just entering its name.
View	When we need to view the bus details we will use.	The corresponding details are viewed.
Bus	When you want to store additional bus details.	All the details like name, phone number, address, account details, will be entered.

Primary index:

Index on Sequential File, also called Primary Index, when the Index is associated to a Data File which is in turn sorted with respect to the search key.

A Primary Index forces a sequential file organization on the Data File; Since a Data File can have just one order there can be just one Primary Index for Data File. Usually used when the search key is also the primary key of the relation. Usually, these indexes fit in main memory.

Indexes on sequential files can be:

1. Dense: One entry in the index file for every record in the data file.
2. Sparse: One entry in the index file for each block of the data file.

Testing

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub-assemblies and/or a finished product. It is the process of exercising software with the intent of ensuring that a software system meets its requirements and user expectation does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

Types of testing

Unit testing:

Unit testing is a software testing method by which individual units of source code, sets of one or more computer program modules together with associated control data, usage producers, and operating procedures, are tested to determine whether they are fit for use. The primary goal of unit testing is to take the smallest piece of testable software in the application, isolate it from the reminder of the code, and determine whether it behaves exactly as you expect. Each unit is tested separately before integrating them into modules to test the interfaces between modules. Unit testing has proven its value in that a large percentage of defects are identified during its use.

Integrated testing:

Integration testing is the logical extension of unit testing. In its simplest form, two units that have already been tested are combined into a component and the interface between them is tested. A component, in this sense, refers to an integrated aggregate of more than one unit. In a realistic scenario, many units are combined into components, which are in turn aggregated into even larger parts of the program. The idea is to test combinations of pieces and eventually expand the process to test your modules with those of other groups. Eventually all the modules making up the process are tested together. Beyond that, if the program is composed of more than one process, they should be tested in pairs rather than all at once.

System testing:

The process of performing a variety of tests on a system to explore functionality or to identify problems is called System Testing. It is usually required before and after a system is put in place. A series of systematic procedures are referred to while testing is being performed. These procedures tell the tester how the system should perform and where common mistakes may be formed. Testers usually try to “Break the system” by entering data that may cause the system to malfunction or return incorrect information. For example, A tester may put in a city in a search engine design to accept only states, to see how the system responds to the incorrect inputs.

Test case:

Test case is a set of test inputs, execution and expected results developed for a particular objective. An excellent test case satisfies the following criteria:

- Reasonable probability of catching errors.
- Does interesting things.
- Doesn’t do unnecessary things.
- Neither too simple nor too complex.
- Allows isolation and identification of errors.

Unit testing:

Sl.no	Test Case	Input	Expected Output	Actual output	Remarks
1	Test case for adding valid bus	Valid phone number and name	Add bus	Add bus	PASS
2	Test case for adding Invalid bus	Invalid phone number and name	Invalid bus detected	Invalid bus detected	PASS

CODE

```
#include <conio.h>
#include <cstdio>
#include <iostream>
#include <string.h>
#include <cstdlib>
using namespace std;
void intro()
{
     //system("color 03");
     system("cls");
  cout<<"\t\t\t\t*\t*";
  cout<<"\t\t\t\t**\t**";
   cout<<"\t\t\t\t***\t***";
   cout<<"\t\t\t\t****\t****";
   cout<<"\t\t\t\t*****\t*****";
    cout<<"\t\t\t\t******\t******";
   cout<<"\t\t\t\t*******\t*******";
   cout<<"\t\t\t\t*******\t*******";
   cout<<"\t\t\t\t******\t******";
   cout<<"\t\t\t\t*****\t*****";
   cout<<"\t\t\t\t****\t****";
   cout<<"\t\t\t\t***\t***";
   cout<<"\t\t\t\t**\t**";
   cout<<"\t\t\t\t*\t*";
}

static int p ;//= 0;
class Bus_Reservation
{
  char busn[5];
  char driver[10];
  char arrival[5];
  char depart[5];
  char from[10];
  char to[10];
  char seat[8][4][10];

public:
  void add_bus();
  void reserve_bus();
  void empty();
  void show_bus();
  void is_bus_available();
  void position(int i);
}

bus[10];

void Bus_Reservation::add_bus()
{
  cout<<"Enter bus number: ";
  cin>>bus[p].busn;

  cout<<"Enter Driver's name: ";
  cin>>bus[p].driver;

  cout<<"Arrival time: ";
  cin>>bus[p].arrival;

  cout<<"Departure: ";
  cin>>bus[p].depart;

  cout<<"From: ";
  cin>>bus[p].from;

  cout<<"To: ";
  cin>>bus[p].to;

   cout<<"\n Added Successfully....";

  bus[p].empty();
```

```
 p++;

 getch();

}

void Bus_Reservation::reserve_bus()
{
 int seat;
 char number[5];
 top:
 cout<<"Bus number: ";
 cin>>number;
 int n;
 for(n=0;n<=p;n++)
 {
  if(strcmp(bus[n].busn, number)==0)
  break;
 }
 while(n<=p)
 {
  cout<<"\nSeat Number: ";
  cin>>seat;
  if(seat>32)
  {
   cout<<"\nThere are only 32 seats available in this bus.";
  }
  else
  {
   if (strcmp(bus[n].seat[seat/4][(seat%4)-1], "Empty")==0)
   {
    cout<<"Enter passanger's name: ";
    cin>>bus[n].seat[seat/4][(seat%4)-1];
    break;
   }
  else
   cout<<"The seat no. is already reserved.\n";
   }
```

```
     }
    if(n>p)
    {
      cout<<"Enter correct bus no.\n";
      goto top;
    }
    cout<<"\n Added Successfully....";
            getch();
  }



void Bus_Reservation::empty()
{
  for(int i=0; i<8;i++)
  {
    for(int j=0;j<4;j++)
    {
      strcpy(bus[p].seat[i][j], "Empty");
    }
  }
}

void Bus_Reservation::show_bus()
{
  int n;
  char number[5];
  cout<<"Enter bus number: ";
  cin>>number;
  for(n=0;n<=p;n++)
  {
    if(strcmp(bus[n].busn, number)==0)
    break;
  }
  while(n<=p)
  {

            cout<<"\n@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@@@@@@\n";
            cout<<"Bus no: "<<bus[n].busn
```

```
        <<"\nDriver: "<<bus[n].driver<<"\nArrival time: "
        <<bus[n].arrival<<"\nDeparture time: "<<bus[n].depart
        <<"\nFrom: "<<bus[n].from<<"\nTo: "<<
        bus[n].to<<"\n";

        cout<<"@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@@@@@@\n";
        bus[0].position(n);

        int a=1;
        for (int i=0; i<8; i++)
        {
        for(int j=0;j<4;j++)
        {
        a++;
        if(strcmp(bus[n].seat[i][j],"Empty")!=0)
        cout<<"\nThe seat no "<<(a-1)<<" is reserved for "<<bus[n].seat[i][j]<<".";
        }
        }
        break;
        }
        if(n>p)
        cout<<"Enter correct bus no: ";

        getch();
}

void Bus_Reservation::position(int l)
{
  int s=0;p=0;
  for (int i =0; i<8;i++)
  {
    cout<<"\n";
    for (int j = 0;j<4; j++)
    {
      s++;
      if(strcmp(bus[l].seat[i][j], "Empty")==0)
        {
         cout.width(5);
```

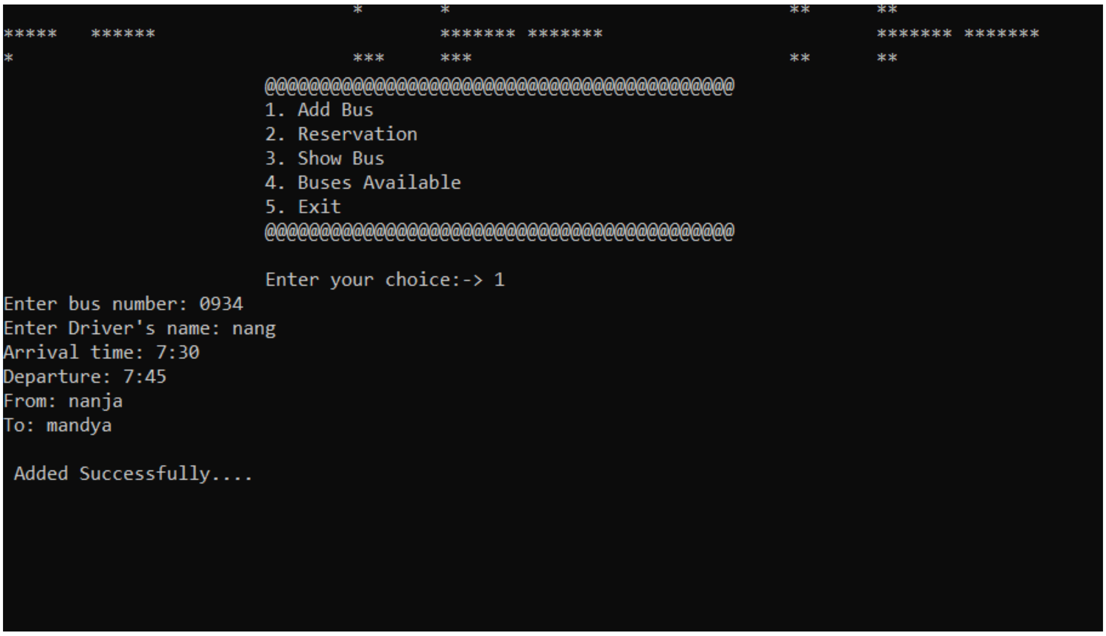
```
          cout.fill(' ');
          cout<<s<<".";
          cout.width(10);
          cout.fill(' ');
          cout<<bus[l].seat[i][j];
          p++;
        }
        else
        {
        cout.width(5);
        cout.fill(' ');
        cout<<s<<".";
        cout.width(10);
        cout.fill(' ');
        cout<<bus[l].seat[i][j];
        }
      }
    }
    cout<<"\n\nThere are "<<p<<" seats empty in Bus Number: "<<bus[l].busn;
  }

void Bus_Reservation::is_bus_available()
{
  for(int n=0;n<p;n++)
  {
            cout<<"\n@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@\n";
            cout<<"Bus number: "<<bus[n].busn<<"\nDriver: "<<bus[n].driver
            <<"\nArrival time: "<<bus[n].arrival<<"\nDeparture Time: "
            <<bus[n].depart<<"\nFrom: "<<bus[n].from<<"\nTo: "
            <<bus[n].to;

            cout<<"\n@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@\n";
  }
getch();
}

int main()
```

```
{
system("cls");
int w;
while(1)
{
system("cls");
    intro();
  cout<<"\n";
cout<<"\t\t\t@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@@@@\n";
  cout<<"\t\t\t1. Add Bus\n\t\t\t"
  <<"2. Reservation\n\t\t\t"
  <<"3. Show Bus\n\t\t\t"
  <<"4. Buses Available \n\t\t\t"
  <<"5. Exit";

cout<<"\n\t\t\t@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@@@@@\n";
  cout<<"\n\t\t\tEnter your choice:-> ";
  cin>>w;
  switch(w)
  {
    case 1:  bus[p].add_bus();
      break;
    case 2:  bus[p].reserve_bus();
      break;
    case 3:  bus[0].show_bus();
      break;
    case 4:  bus[0].is_bus_available();
      break;
    case 5:  exit(0);
  }
}
return 0;
}
```

SNAPSHOTS

Snapshot 1:

To add the bus details to the file like bus number, driver name, arrival time and departure

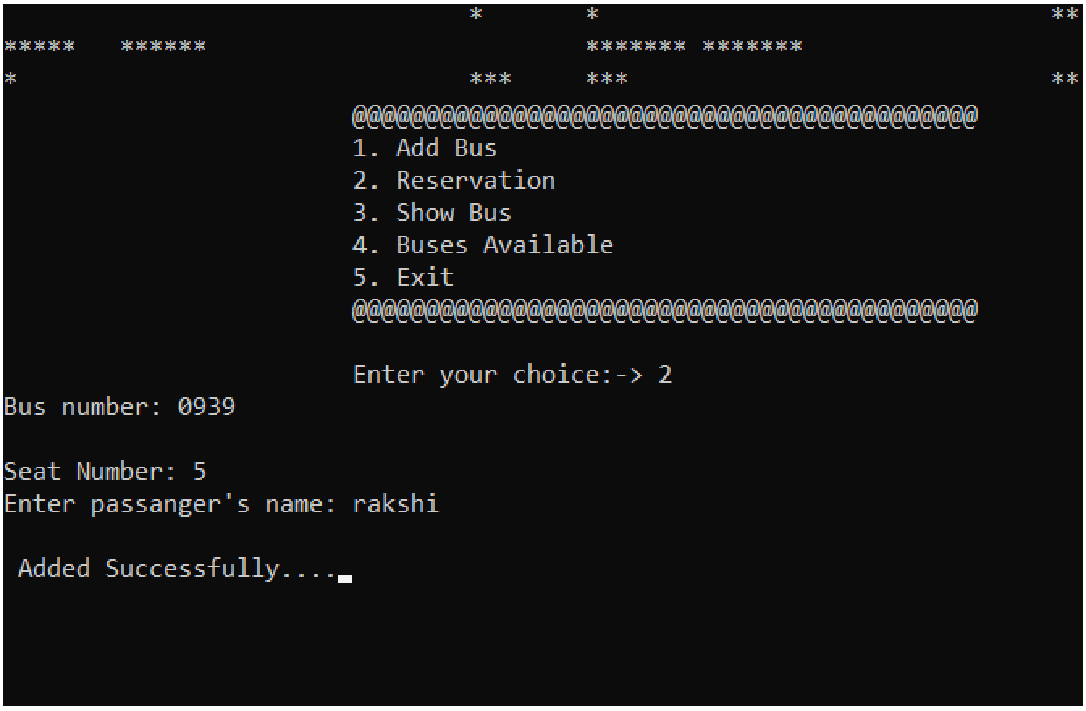


```
                                 *       *                              **      **
*****   ******                           ******* *******                        ******* *******
*                               ***     ***                             **      **
                        @@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
                        1. Add Bus
                        2. Reservation
                        3. Show Bus
                        4. Buses Available
                        5. Exit
                        @@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@

                        Enter your choice:-> 1
Enter bus number: 0934
Enter Driver's name: nang
Arrival time: 7:30
Departure: 7:45
From: nanja
To: mandya

 Added Successfully....
```

Fig No:1 Add Bus

Snapshot 2:

To reserve the ticket for the passenger

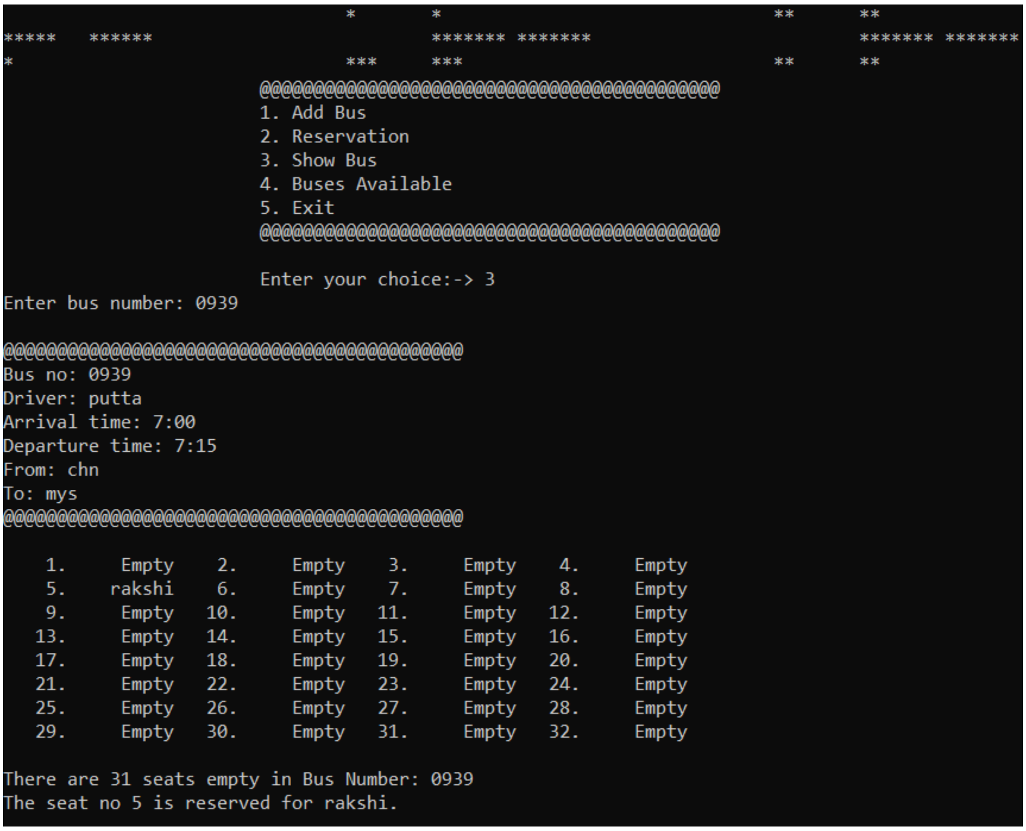


```
                                *       *                               **
*****   ******                          ******* *******
*                               ***     ***                             **
                        @@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
                        1. Add Bus
                        2. Reservation
                        3. Show Bus
                        4. Buses Available
                        5. Exit
                        @@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@

                        Enter your choice:-> 2
Bus number: 0939

Seat Number: 5
Enter passanger's name: rakshi

 Added Successfully....
```

Fig No:2 Reservation

Snapshot 3:

To fetch the bus details and the reserved seats

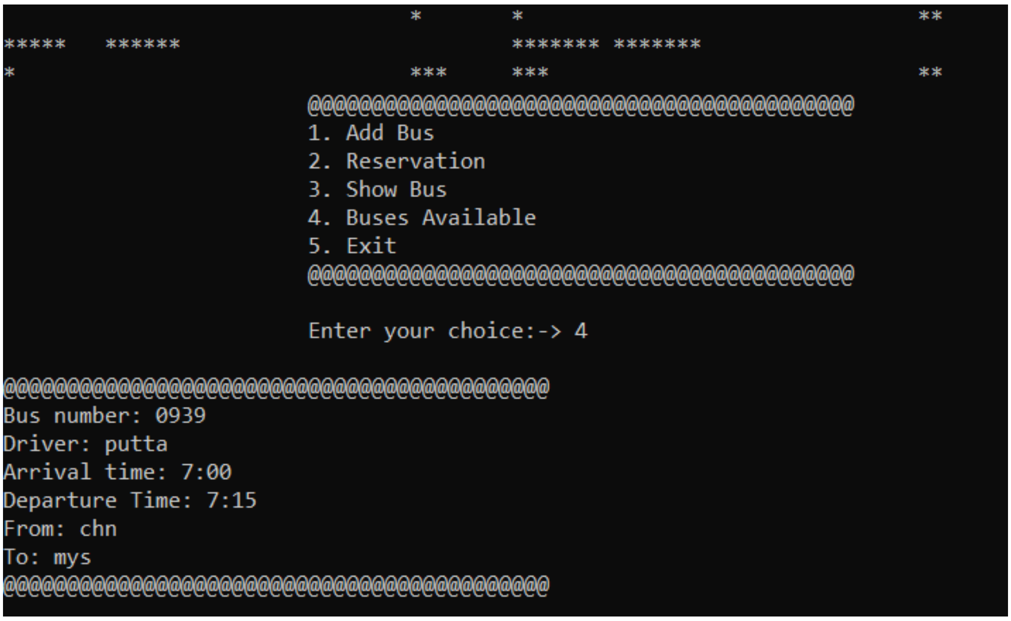


```
                                *       *                              **      **
*****   ******                          ******* *******                        ******* *******
*                               ***     ***                            **      **
                        @@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
                        1. Add Bus
                        2. Reservation
                        3. Show Bus
                        4. Buses Available
                        5. Exit
                        @@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@

                        Enter your choice:-> 3
Enter bus number: 0939

@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
Bus no: 0939
Driver: putta
Arrival time: 7:00
Departure time: 7:15
From: chn
To: mys
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@

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

There are 31 seats empty in Bus Number: 0939
The seat no 5 is reserved for rakshi.
```

Fig No:3 Show buses

Snapshot 4:

To fetch the details of no of buses and their destination

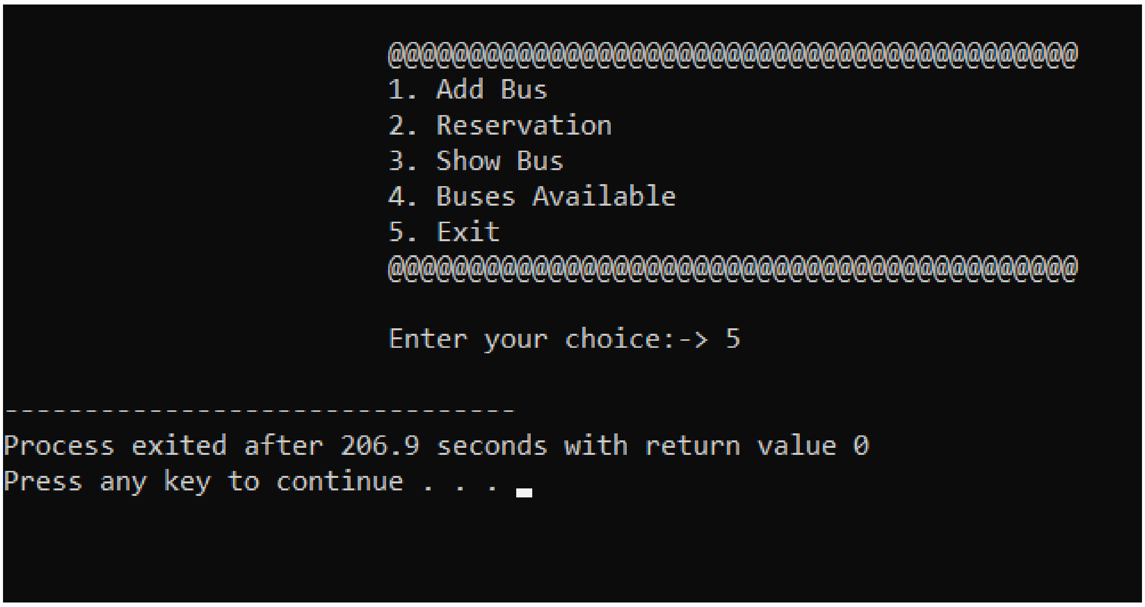


```
                                *       *                               **
*****   ******                          ******* *******
*                               ***     ***                             **
                        @@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
                        1. Add Bus
                        2. Reservation
                        3. Show Bus
                        4. Buses Available
                        5. Exit
                        @@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@

                        Enter your choice:-> 4

@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
Bus number: 0939
Driver: putta
Arrival time: 7:00
Departure Time: 7:15
From: chn
To: mys
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
```

Fig No:4 Buses Available

Snapshot 5:

Exit from the output .



```
                        @@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
                        1. Add Bus
                        2. Reservation
                        3. Show Bus
                        4. Buses Available
                        5. Exit
                        @@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@

                        Enter your choice:-> 5

--------------------------------
Process exited after 206.9 seconds with return value 0
Press any key to continue . . .
```

Fig No:5 Exit

CONCLUSION

It can be observed that computer applications are very important in every field of human endeavor. Here all the information about customer that made reservation can be gotten just by clicking a button with this new system, some of the difficulties encountered with the manual system are overcome. It will also reduce the workload of the staff, reduce the time used for making reservation at the bus terminal and also increase efficiency. The application also has the ability to update records in various files automatically thereby relieving the company's staff the stress of working from file security of data. Our project online bus reservation system provides an easy way for booking the bus tickets. Our project has succeeded in managing the data and providing the best service to the users

REFERENCES

Book:

Michael J. Folk, Bill Zoellick, Greg Riccardi: File Structures-An Object-Oriented Approach with C++, 3rd Edition, Pearson Education,1998.

Websites:

[1] https://code-projects.org/telephone-directory-in-c-with-source-code/

[2] https://projectsgeek.com/2014/09/phone-directory-system-project-c.html

[3] https://code-blocks.en.uptodown.com/windows/download

[4] https://www.youtube.com/watch?v=8Qht1JtEw1M\

[5] https://sourcecodehero.com/telephone-directory-in-c-project-with-source-code/