Topic : Online Public Transport Reservation System

Group no : KDY_09

Campus : Kandy

Submission Date : 20/05/2022

We declare that this is our own work and this Assignment does not incorporate without acknowledgment any material previously submitted by anyone else in SLIIT or any other university/Institute. And we declare that each one of us equally contributed to the completion of this Assignment.

Registration No	**Name**	**Contact Number**
IT21312526	R.T.Ahamed	0776271771
IT21333484	M.H.A.Umair	0765588565
IT21318252	R.A.Muahammedh	0770640602
IT21361654	M.R.S.Hussain	0725074306
IT21374524	M.L.Waseek	0762738161

Requirement

The online bus reservation system eZBus has two different users. One is customer and the other is admin. A user has a User id, first name, last name , user name, email and password to be stored in data base.

System also has a set of busses in store, each bus has a route to travel, and each route has many stations. Buses have bus id , seat , features , departure time and arrival time. Route has its route id, source, destination , its details and distance, and each station has a name and price from the source. A person will be able to view available buses.

A customer is able to book a bus to wherever he wants to travel within eZBus's jurisdiction. Each booking has its own id, source station, destination, date of travel, number of seats and price. Payments can be made through bank card via online, by entering card details and choosing the payment type.

Customers are able to leave a feedback and review each having their own id. In addition to that feedback has content. While review has review content and rating.

Rejecting

- Redundant: User
- Event or operation: Request refund
- Out of scope: eZBus , system
- Meta-language: person
- An attribute: User id, first name , last name , user name, email ,password , bus id , seat , features , departure arrival time, route id , source, destination , details and distance, station name and price from the source , booking id, source station, destination , date of travel , number of seats , card details , payment type, refund id , confirmation status ,

reason for asking refund and amount, feedback id , review id ,feedback content, review content and rating

Classes

- Customer
- admin
- Bus
- Route
- Feedback
- Booking

CRC cards

Customer Class	
Responsibilities	Collaborations
Register details	
Reserve buses	Bus
View buses	Bus
Leave feedback	Feedback

Admin Class	
Responsibilities	Collaborations
Reply feedback	Feedback

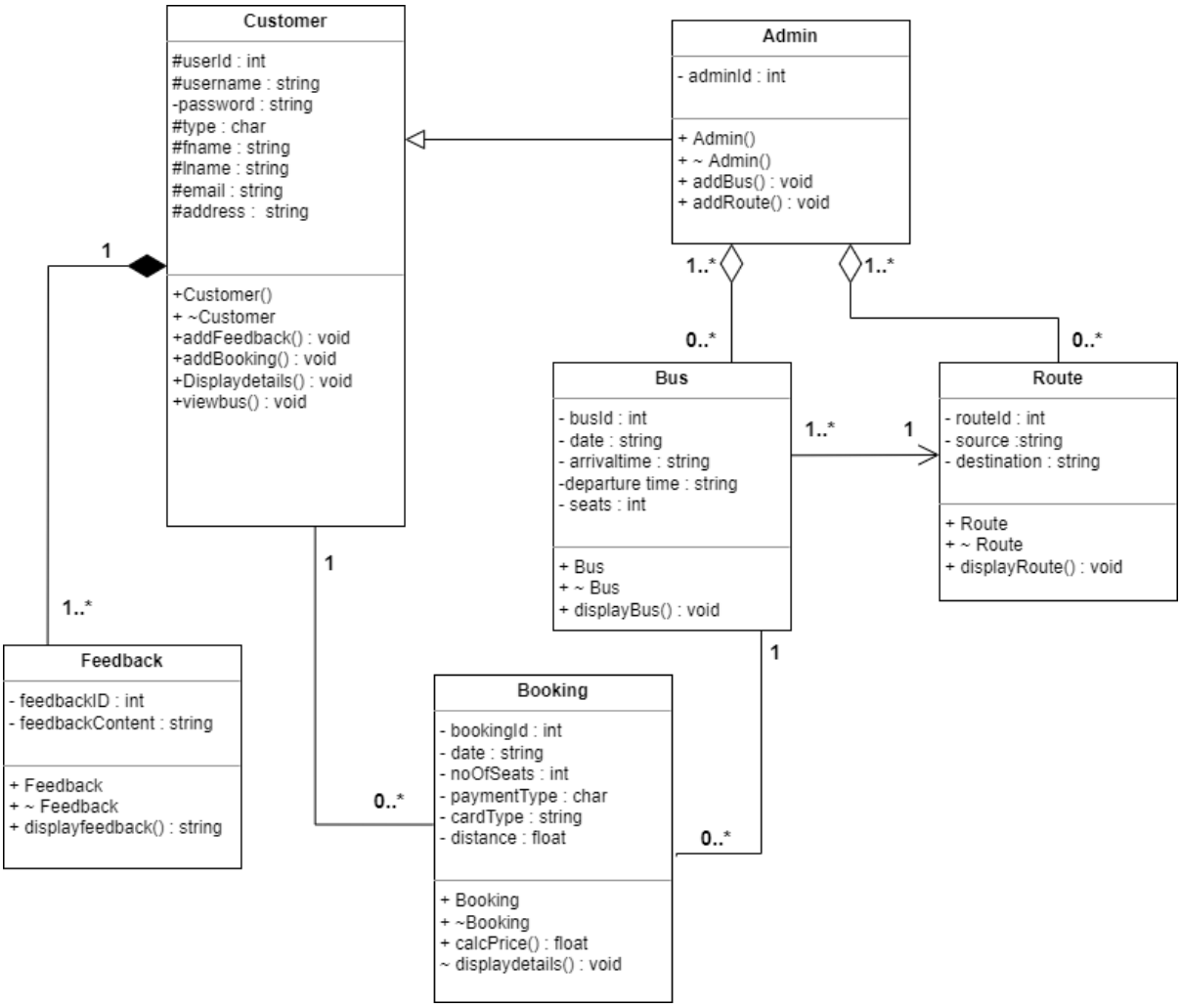
Bus Class	
Responsibilities	Collaborations
Add bus Edit bus	Route, Admin

Route Class	
Responsibilities	Collaborations
Add route	Admin

Booking Class	
Responsibilities	Collaborations
Store booking details Reserve seats	 Bus, customer

Feedback Class	
Responsibilities	Collaborations
Store feedback	

UML class diagram



- Route, Main: R.T.Ahamed - IT21312526
- Customer, Admin: M.H.A.Umair - IT21333484
- Bus: R.A.Muahmmedh - IT21318252
- Booking: M.R.S.Hussain - IT21361654
- Feedback: M.L.Waseek - IT21374524

Code

M.H.A.Umair – IT21333484

Customer class header file

```
#include "Feedback.h"
#include "Booking.h"

define SIZE = 5;

class Customer
{
protected:
  int userid;
  string username;
  char type[1];
  string fname;
  string lname;
  string address;
  string email;

private:
  string password;
  Feedback *feed[SIZE];
  Booking *book[SIZE];

public:
  Customer();
  Customer(int id, char tp[], string fn, string ln, string addr, string mail, string un, string pass);

  void setUserid(int i);
  void setType(char typ[]);
  void setFname(string fnme);
  void setLname(string lnme);
  void setAddress(string adrs);
  void setEmail(string mil);
  void setUsername(string usern);
  void setPassword(string ps);

  int getUserid();
  char getType();
  string getFname();
  string getLname();
  string getAddress();
```

```
  string getEmail();
  string getUsername();
  string getPassword();

  void addFeedback(int fd_no1, int fbid1, string cntnt1);
  void displayFeedback();

  void viewBus();
  void addBooking(Booking *b);
  void displayCustomer();

  ~Customer();
};
```

Customer class implementation

```
#include <iostream>
#include "Customer.h"
#include <string.h>

using namespace std;

define SIZE = 5;

Customer::Customer()
{
  userid = 0;
  type = '';
  fname = '';
  lname = '';
  address = '';
  email = '';
  username ='';
  password ='';
}

Customer::Customer(int id, char tp[], string fn, string ln, string addr, string mail, string un, string
pass)
{
  userid = id;
  strcpy(type, tp);
  fname = fn;
  lname = ln;
  address = addr;
  email = mail;
  username = un;
  password = pass;
}

Customer::~Customer()
```

```
{
  cout << "Deleting Customer" << endl ;

  for(int i ; i < SIZE ; i++)
    delete feed[i] ;
}
```

Admin Class header file

```
#include "Route.h"
#include "Bus.h"

class Admin : public customer
{
private:
    int admin_id;
    Route *rt[SIZE];
    Bus *bus[SIZE] ;

public:
    Admin();
    Admin( ind aid ,int id, char tp[], string fn, string ln, string addr, string mail, string un, string
pass) : Customer( id, tp,  fn, ln,  addr,  mail,  un,  pass);

    void addRoute(Route *rt1, Route *rt2);
    void addBus(Bus *bs1 , Bus *bs2) ;
    void setadminId(int id ) ;
    int getadminId () ;

    void displayAdmin();
    ~Admin();
};
```

Admin Class implementation

```
#include<iostream>
#include "Admin.h"
#include <cstring>

using namespace std;
Admin::Admin()
    {
        admin_id = 0;
    }

Admin::Admin( ind aid ,int id, char tp[], string fn, string ln, string addr, string mail, string un,
string pass) : Customer( id, tp,  fn, ln,  addr,  mail,  un,  pass)
    {
        admin_id = aid;
    }
```

```
Admin::~Admin()
   {
      cout << "Deleting Admin" <<endl<< endl;
   }
```

R.T.Ahamed – IT21312526

Route class header file

```
class Route
{
private:
   int route_id;
   string source;
   string destination;


public:
   Route();
   Route(int id, string src, string dest);

   void setRoute_id(int id);
   void setSource(string src);
   void setDestination(string dest);

   int getRoute_id();
   string getSource();
   string getDestination();

   void displayRoute();

   ~Route();
};
```

Route class implementation

```
#include <iostream>
#include <string.h>
#include "Route.h"

Route::Route()
{
  route_id = 0;
  source = '';
  destination = '';
}
```

```
Route::Route(int id, string src, string dest)
{
  route_id = id;
  source = src;
  destination = dest;
}

Route::~Route()
{
  cout << "Deleting Route" << route_id << endl;
}
```

Main

```
#include <iostream>
#include "Admin.h"
#include "Customer.h"
#include "Bus.h"
#include "Route.h"
#include "Booking.h"
#include "Feedback.h"
#include <cstring>

using namespace std ;

int main(){

  Admin *a1 ;
  a1 = new Admin(100,101, A , "Peter" ,"parker", "No 128 ,main road", "parker123@gmail.com"
, "parker123" , "spiderman@678") ;

  Customer *c1 ;
  c1 = new Customer(102 , C , "perera" , "thisara" , "No 64,new town , Kandy " ,
"tp25@gmail.com" , "tp45" , "Tperera@123" ) ;



  Route *rt1 ;
  rt1 = new Route(10 , "kandy" , "Colombo") ;

  Bus *bs1 ;
  bs1 = new Bus(105 , 45 , "07.30" , "12.30" , rt1) ;

  Booking *bk1 ;
  bk1 = new Booking(200 , "23-05-2022" , 2 , "Visa" , "Full" , 100 , bs1 , c1) ;


  return 0 ;
}
```

R.A.Muahammedh -IT21318252

Bus class header file

```
#include "Route.h"
#include "Booking.h"

define SIZE= 5;

Class Bus
{
  private:
    int Bus_id ;
    string Seats ;
    string arrivalTime ;
    string departureTime ;
    Route *rt ;
    Booking *bk[SIZE] ;

  public:
    Bus();
    Bus(int bus_id, string seats, string aTime, string dTime, Route *r);

    void setBus_id(int id);
    void setSeats(int st);
    void setArrivalTime(string arr);
    void setDepartureTime(string dep);

    int getBus_id();
    string getSeats();
    string getArrivalTime();
    string getDepartureTime();

    void displayBus();
    void BookBus(Booking *bk);

    ~Bus();


}
```

Bus class implementation

```
#include<iostream>
#include<string.h>
#include "Bus.h"
#include "Route.h"

using namespaces std;

Bus::Bus()
   {
     Bus_id = 0;
     Seats = 0;
     arrivalTime = "";
     depatureTime = "";
   }

Bus::Bus(int bus_id, string seats, string aTime, string dTime, Route *r)
   {
     Bus_id= bus_id ;
     Seats= seats ;
     arrivalTime=aTime ;
     depatureTime=dTime ;
     rt = r ;
   }

Bus::~Bus()
   {
    cout<<"Deleting Bus " <<  Bus_id << endl;
   }
```

M.R.S.Hussain – IT21361654

Booking class header file

```
#include<iostream>
#include<cstring>
#include"Bus.h"
#include"Customer.h"

using namespace std;

class Booking
{
  private:
    int bookingId;
          string bookingDate;
```

```
        int noOfSeats;
        string cardType;
        char paymentType;
        float travelDistance;
    Bus *bookedBus; //an object of Bus as attribute
    Customer *bookedcustomer ; //an object of customer

  public:
    Booking ();
        Booking (int id, string date , int seats,
        string ctype, char ptype, float distance ,Bus *bus , Customer *cus) ;

    void setId (int id);
    void setDate (string date);
    void setSeat (int seats);
    void setCType (string ctype);
    void setPtype (char ptype);
    void setDistance (float distance);
    void setBus (Bus *bus);
    void setCustomer (Customer *cus);

    int getId ();
    string getDate ();
    int getSeat ();
    string getCType ();
    char getPtype ();
    float getDistance ();
    int getBus ();
    int getCustomer ();
        double calcPrice ();
        void displayDetails ();
    ~Booking ();
};
```

Booking class implementation

```
#include<iostream>
#include<cstring>
#include"Booking.h"

using namespace std;

  Booking::Booking ()
        {
            bookingId = 0;
                bookingDate = "";
                noOfSeats = 0;
                cardType = "";
                paymentType = "";
                travelDistance = 0;
```

```
        }
Booking::Booking (int id, string date , int seats,
        string ctype, char ptype, float distance ,Bus *bus, Customer *cus)
        {
                bookingId = id;
                        bookingDate = date;
                        noOfSeats = seats;
                        cardType = ctype;
                        paymentType = ptype;
                        travelDistance = distnace;
          bookedBus = bus;
  bookedcustomer = cus ;

        }

Booking::~Booking () {
  cout << "Cancelling Booking " << bookingId
  << endl;
 }
```

M.L.Waseek – IT21374524

Feedback Class header file

```
#include <cstring>
#include <iostream>
#include"Customer.h"

using namespace std;

class Feedback {
 private:
   int feedbackId;
   string feedbackContent;

 public:
  Feedback(){};
  Feedback(int id, string content );
  void setId(int id);
  void setContent( string content );
  int getId();
  string getContent();
  void display();
  ~Feedback() {};
};
```

Feedback class implementation

```
#include<iostream>
#include<cstring>
#include"Feedback.h"

Feedback::Feedback(){
      feedbackId = 0;
      feedbackContent ="";


}
  Feedback:: Feedback(int id, string content ) {

       feedbackId = id;
       feedbackContent = content;


    }

   Feedback::~Feedback() {
     cout << "Deleting Feedback given " << feedbackId       << endl;
    }
```