INSTITUTE FOR ADVANCED COMPUTING AND SOFTWARE DEVELOPMENT AKURDI,PUNE

Documentation On

“Online Bus Ticket Booking”

PG-DAC Feb 2020

Submitted By:

Group No:41

Name	***Roll No***
Anjali Roopchandani	1118
Harshada Pimpalkar	1138

Mr.Prashant Karhale
Center Coordinator

Mr. Kashinath Patil
Project Guide

Table of Contents

List of Figures

Title : Online Bus Ticket Booking

List of Actors : 1. Administrator

2. Customer
3. User

1.Introduction

The ***Online Bus Ticket Booking*** application is developed for the Travellers to reserve seats online and to save them from hassles.It will allow the users to enjoy the booking of bus tickets form the current position through internet. They will be provided with the bus routes along with some other facilities like Booking the tickets based on there comfort level,the time of arrival of the bus and the departure of the bus,Cancelation of the tickets.The User can simply enter the source and the destination location and can see all the buses of that route along with time and the price.For the ticket booking the user can login and then book the ticket for the desired ride,One User can booked no. of tickets.The Customer is one who had booked the ticket, can have the choice to cancel there tickets,can view the seats available in the specific bus etc.

The administrator can handle the various aspects like Adding the buses to the specified routes,Adding the routes,Removing the buses from the routes,Updating the route,Updating the bus details,changing the facilities according to price,can monitor various other aspects.

Document Purpose:

This document is meant to delineate the features of SRS,so as to serve as a guide to the developers on one hand and a software validation document for the prospective client on the other.

Problem Statement

Some of the existing system of the Bus Ticket Reservation are based on our traditional way of keeping records and details on paper and registers.This leads to the UnSecurity of all the information stored and leads to the hassle,whenever the retrieval of the data or some information has to be done.It becomes very hard to manage all the contents by using pen and paper.It becomes tedious job to maintain the records and to keep the track of the past records as well as existing records.

Hence we have Developed this Project “Online Ticket Booking System” to overcome all the flaws and hassle of the existing system and giving the power to the admin or Agent of the Travel Agency to smoothly manage the ticket booking system.

Scope of Project

This application can be used by any Travel Agent,Travel Agency to issue the tickets to the customers.It also helps the customer to enquire the availability of seats in a particular bus for the specific date from desired location.It will also provide the facility to check the timings and schedule of the buses along with the ticket price.

This project traverses lot of areas ranging from business concepts to computing field, and required to perform several researches to be able to achieve the project objectives. The area covers include:

1. J2EE Technology used for the Development (Back_End) of this project.
2. Angular (Front_end) technology along with MySQL database is used for the UI and for storing the information.
3. General Customer as well as the Agent can use the system effectively.
4. Web-platform means that the system will be available for access 24*7 except when there is temporary server issue which is expected to the minimal.

Aims & Objectives

Specific Goals are:-

1. This system project is made as user friendly as possible so that any one can use it with little knowledge of system computers.

2. We provide up to date information that is not possible manually.

3. The Online Bus Ticket Booking project will reduce the Ticket Booking tedious job of system paperwork by keeping all the details of bus ticket booking, cancelling tickets ,are stored in the database in computer's hard disk.

4. The objective of my project is to make easy the Ticket Booking system of Ticket Booking Agency simple, reliable, user friendly, and corrective. Moreover less time consuming as compared to manual work.

5. Can Increase The Ticket Booking efficiency.

2. Overall Description

Product Perspective:

Existing System Function:

The existing system of the Bus Ticket Reservation are based on our traditional way of keeping records and details on paper and registers.This leads to the UnSecurity of all the information stored and leads to the hassle,whenever the retrieval of the data or some information has to be done.It becomes very hard to manage all the contents by using pen and paper.It becomes tedious job to maintain the records and to keep the track of the past records as well as existing records.

Proposed System:

The Online Bus Ticket Management System provides the feature for Admin,User and Customer it includes several functionalities:-

Ticket Management:

It will provide the facility to the customer to view the bus on specific routes and the book the tickets on particular date.

Bus Management:

It will provide facility to the user to manage the bus related functions like adding the new bus,updating the bus,removing the bus.The User will be able to view the buses and will be able to book the tickets.

Routes Management:

This function will enable the Admin to add the routes for travelling form some source city to destination city, and can update the routes,can add the bus for the route,delete the route .

Benefits of Online Ticket Booking System

1. This system will help to maximize the number of Reservations.
2. The payment process will take place quickly without hassle.
3. Easy to Manage the calender.
4. Easy to Manage all the records.
5. It This Online Bus Ticket Booking System is fully functional and flexible.
6. It is very easy to use.
7. It saves lot of time,money and Labour.

8. This Application acts as an office that is open 24/7.
9. It increases the efficiency of the management at offering quality services to the customers.
10. It provides custom features development and support with the application.

User and Characteristics

a. Admin

→ Admin can login to the System.

→ Update Profile,Password,Phone number.

→ View list of all the routes .

→ Add the Routes.

→ Delete the Routes.

→ Add the Bus.

→ Delete the Bus.

→ Update Route.

→ Delete Route.

b. Customer

→ Customer can enter the Source and Destination and can view the list of buses Running on the particular routes.

→ Can book the tickets.

→ View the number of seats available in the bus along with other bus

Information.

→ Update the Customer Profile.

→ Update password and Phone Number.

Operational Environment

Server Side:

Processor: Intel® Xeon® processor 3500 series
HDD: Minimum 500GB Disk Space
RAM: Minimum 2GB
OS: Windows 10, Linux 6
Database: Oracle 11g

Client Side (minimum requirement):

Processor: Intel Dual Core
HDD: Minimum 80GB Disk Space
RAM: Minimum 1GB
OS: Windows 10, Linux

Design and Implementation Constraints

⇛ This application will use Angular and Java as main technologies.

⇛ HTTP protocols are used as communication protocols, The system should support various RDMS and Cloud Technologies.

⇛Several types of validations make this web application a secured.

⇛Since Online Bus Reservation System is web based application , internet connection must be established.

⇛The Online Bus Reservation System will be used on PCs and will function via internet in any web browser.

3. Requirements Specifications

External Interface Requirements:

1. **User Interfaces:**

All the users will see the same page when they enter in this website. This page asks the users a username and a password for login.Or else User can see the buses available when the enter the source and destination along with date.

After being authenticated by correct username and password, user will be redirect to their corresponding profile where they can do various activities.

The user interface will be simple and consistence, using terminology commonly understood by intended users of the system. The system will have simple interface, consistence with standard interface, to eliminate need for user training of infrequent users.

2. **Hardware Interfaces:**

No extra hardware interfaces are needed.
The system will use the standard hardware and data communication resources.
This includes, but not limited to, general network connection at the server/hosting site, network server and network management tools.

3. Application Interfaces:

OS: Windows 7, Linux
Web Browser:
The system is a web-based application; clients need a modern web browser such as Mozilla Firebox, Internet Explorer, Opera, and Chrome. The computer must have an Internet connection in order to be able to access the system.

4. Communications Interfaces:

This system uses communication resources which includes but not limited to,HTTP protocol for communication with the web browser and web server and TCP/IP network protocol with HTTP protocol.This application will communicate with the database that holds all the ticket booking information. Users can contact with server side through HTTP protocol by means of a function that is called HTTP Service. This function allows the application to use the data retrieved by server to fulfil the request fired by the user.

Non Functional Requirements

Some typical non-functional requirements are:

1. Reliability:

a. The system provides storage of all databases on redundant computers with automatic switchover.

b. The main pillar of reliability of the system is the backup of the database which is continuously maintained and update to reflect the most recent changes.

2. Availability:

a. The system should be available at all times .meaning the user can access it using web browser,

b. Only restricted by the down time of the server on which the system runs.

In case of a of a hardware failure or database corruption, a replacement page will be shown. uptime : It mean 24 * 7 availability

100%--------------
99.9%
99.999%
99.9999%

3. Maintainability:

a. A commercial database is used for maintaining the database and application server takes care of the site. The maintainability can be done efficiently.

4.Portability:

a. The application is HTML and scripting language based (JavaScript). So the end user part is fully portable and any system using

b. Any web browser should be able to use the features of the system ,including any hardware platform that is available or will be available in the future.

c. An end-user is used this system on an OS either it is Windows or Linux.

d. The System shall run on PC, Laptops and PDA etc.

e. The technology should be transferable to different environments easily.

5. Accessibility:

a. Only registered users should be allowed to book tickets after authentications.
b. Only GUI access of the system should be permitted to end users.

Functional Requirements

This section provides requirement overview of the system. Various functional modules that can be implemented by this system will be
Description:

a. Registration:

Registration if User wants to book the bus ticket then he/she must have to register .Unregistered user will not be allowed to book the tickets.

b.Login:

The User who have registered can login to the system by entering the valid user email id and password.

If Admin logins, they can add or remove bus, they can add or remove the routes ,can change the price of the ticket and the timings of the bus.

If User logged in, they can view the buses from the particular route i.e source to destination and can book the no. of tickets.

c.Selection of Source and destination:

The User will be able to choose the source and destination.

d. Available Buses:

After the selection of the source , destination, date the available buses for the specific route along with the time will be shown to the user.

e.Bus Route:

The Route to be followed by the bus while travelling from source to the destination will be shown to the user.

f.Ticket Booking:

The passenger will be able to book the tickets as per their comfort of price,bus catagory,time,date etc.

g.Add route,Update route,Delete route,Add buses,Update buses,Delete Bus

The admin can add the routes,can update the routes,delete routes,add the buses for the specified route, delete the buses.

m.Update admin password,Update admin phone

The admin can change his/her password,change his/her phone number.

o.Update User phone,Update User password

The User can change his/her phone number,can change his/her password.

q.Logout:

After the payment of the ticket the customer will be logged out.

The term client/server refers primarily to an architecture or logical division of responsibilities,the client is the application (also known as the front-end),and the server is the RDBMS (also known as the back-end).

A client/server system is a distributed system in which,Some sites are client sites and others are server sites. All the data resides at the server sites. All applications execute at the client sites.

Performance Requirement

There is no performance requirement in this system, because the server request and response to client is totally based on internet connection of end user.

Hardware Interface

The System must run over the internet.All the hardware shall require to connect to internet will be hardware interface for the system.e.g. modem, WAN, LAN.
Specialized Server Infrastructure Hardware
The system should use distrubuted servers i.e cloud for managing large amount of data so as to make it appear as single unit for end-user.
The system should have proper clusters for backup.

Operational Scenario

⇒The User will login ,firstly it will be validated whether the User logged in is the Admin,Customer or the normal User.According to the User role the scenario will fall apart.

If the User Logged in ,then he/she will be able to perform following functionalities:-

1. Enter the Source (Leaving from) ,Destination (going to) and Date and after clicking the search bus button the system will show all the buses from that route reaching to the entered destination along with the bus-type and the seats available for booking along with the ticket fair.
2. To Book the ticket the User will have to register and then have to re-login.
3. After the re-login the User will be able to :-
    a. Book the no. of tickets.
    b. To check the seats available.
    c. To check the number of Seats already booked.
    d. The payment will be done and the online bus ticket will be generated.

⇒If the Admin logged in, then the Admin will be able to perform the following functionalities:-

1. Add the bus to the specific route.
2. Add the route.
3. Delete the route.
4. Remove the bus from the route.
5. Update the bus details.
6. Update the route details.
7. View the Total tickets booked,seats booked, etc.

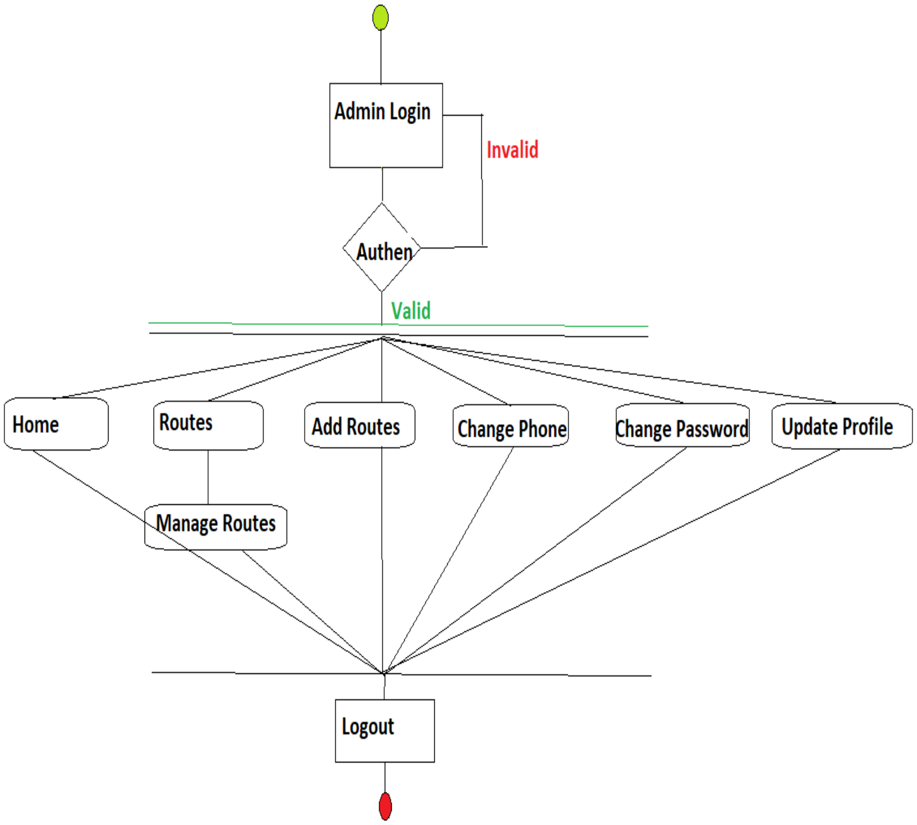
⇒If the Customer logged in, then he/she will be able to perform the following functionalities:-

1. Update the profile.

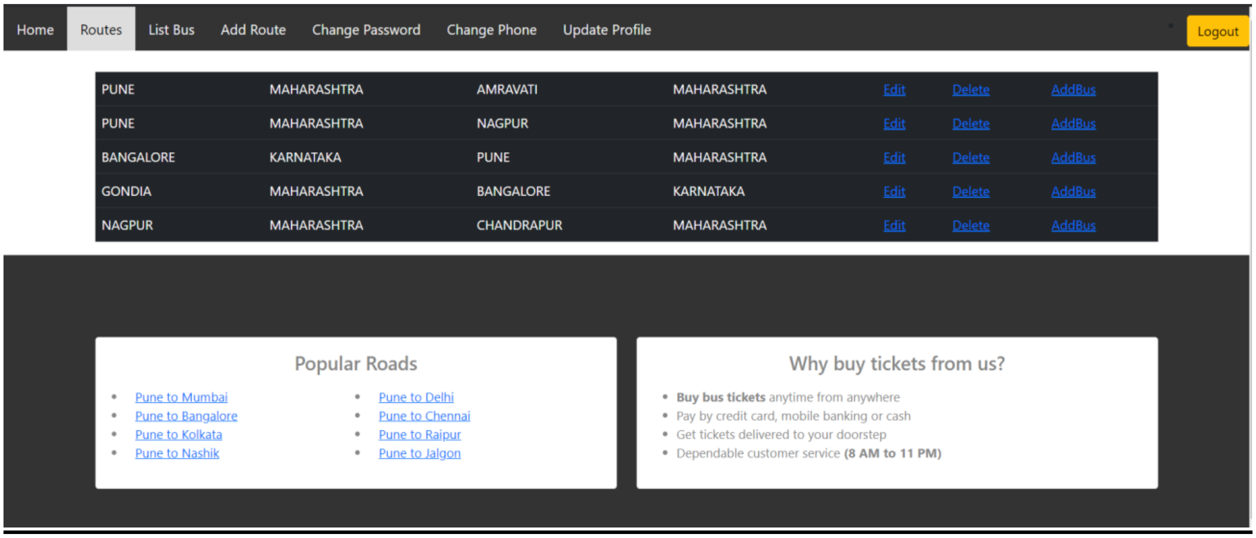
4. SYSTEM DIAGRAMS

4.1.Activity Diagram

a. Admin Activity Diagram



Screenshot of Admin environment



PUNE	MAHARASHTRA	AMRAVATI	MAHARASHTRA	Edit	Delete	AddBus
PUNE	MAHARASHTRA	NAGPUR	MAHARASHTRA	Edit	Delete	AddBus
BANGALORE	KARNATAKA	PUNE	MAHARASHTRA	Edit	Delete	AddBus
GONDIA	MAHARASHTRA	BANGALORE	KARNATAKA	Edit	Delete	AddBus
NAGPUR	MAHARASHTRA	CHANDRAPUR	MAHARASHTRA	Edit	Delete	AddBus

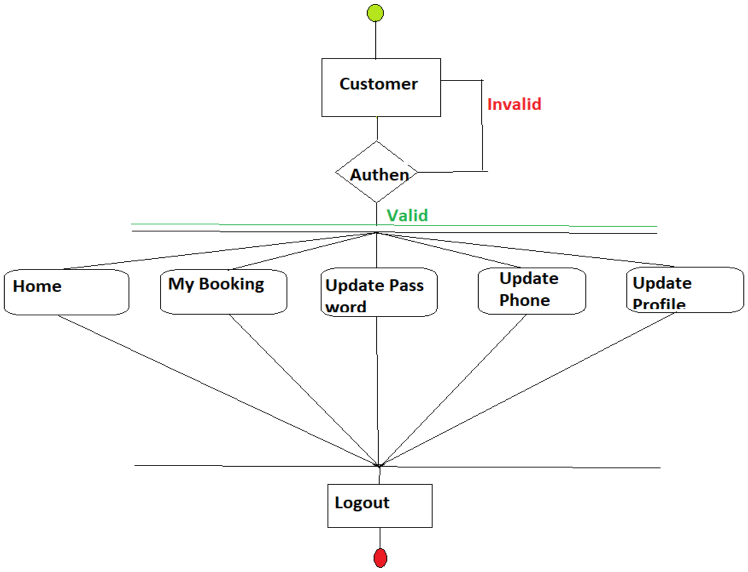
Popular Roads

- Pune to Mumbai
- Pune to Bangalore
- Pune to Kolkata
- Pune to Nashik
- Pune to Delhi
- Pune to Chennai
- Pune to Raipur
- Pune to Jalgon

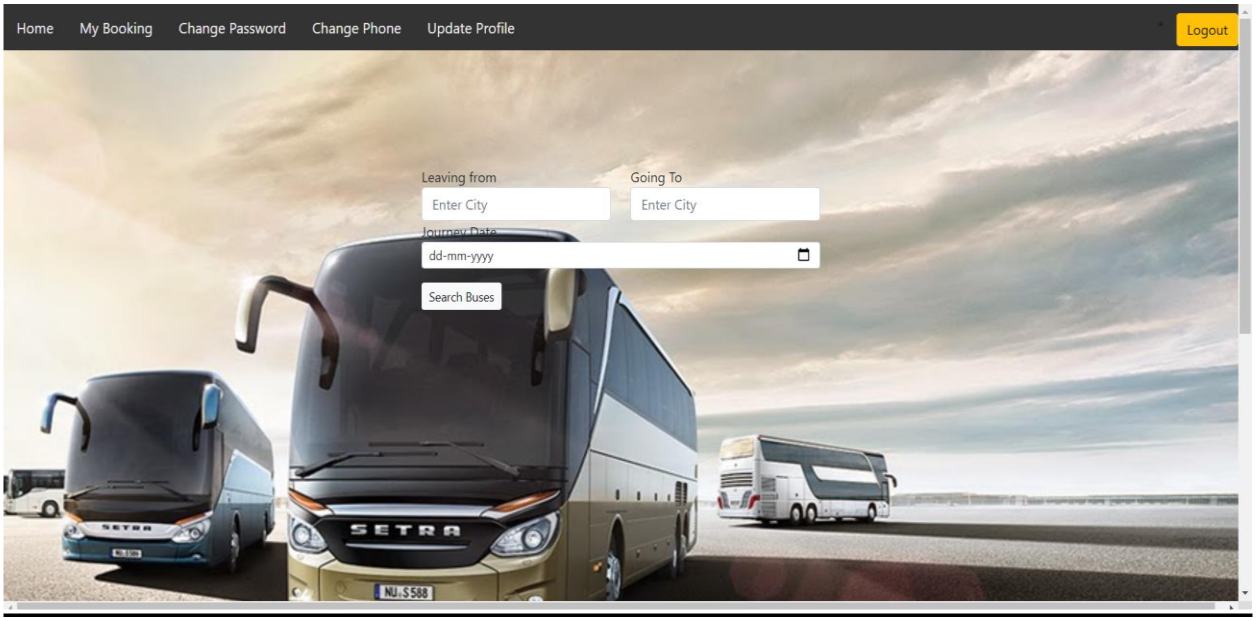
Why buy tickets from us?

- **Buy bus tickets** anytime from anywhere
- Pay by credit card, mobile banking or cash
- Get tickets delivered to your doorstep
- Dependable customer service **(8 AM to 11 PM)**

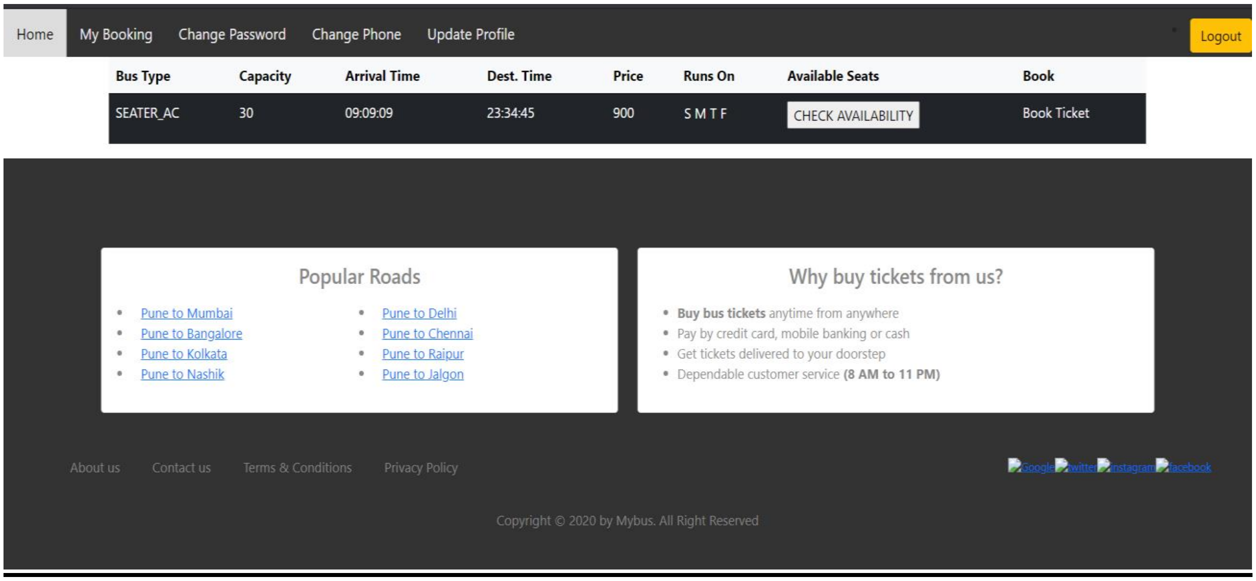
b. Customer Activity Diagram



Home page:



After entering the source and destination:



Bus Type	Capacity	Arrival Time	Dest. Time	Price	Runs On	Available Seats	Book
SEATER_AC	30	09:09:09	23:34:45	900	S M T F	CHECK AVAILABILITY	Book Ticket

Popular Roads

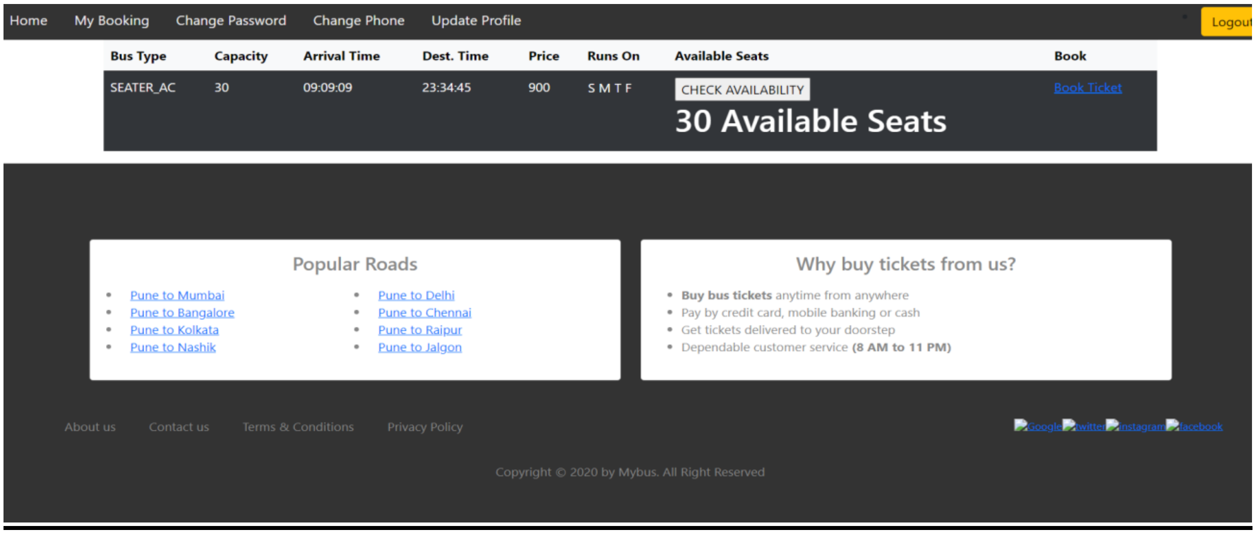
- Pune to Mumbai
- Pune to Bangalore
- Pune to Kolkata
- Pune to Nashik
- Pune to Delhi
- Pune to Chennai
- Pune to Raipur
- Pune to Jalgon

Why buy tickets from us?

- **Buy bus tickets** anytime from anywhere
- Pay by credit card, mobile banking or cash
- Get tickets delivered to your doorstep
- Dependable customer service **(8 AM to 11 PM)**

Copyright © 2020 by Mybus. All Right Reserved

Can check the bus availability:



Bus Type	Capacity	Arrival Time	Dest. Time	Price	Runs On	Available Seats	Book
SEATER_AC	30	09:09:09	23:34:45	900	S M T F	CHECK AVAILABILITY 30 Available Seats	Book Ticket

Popular Roads

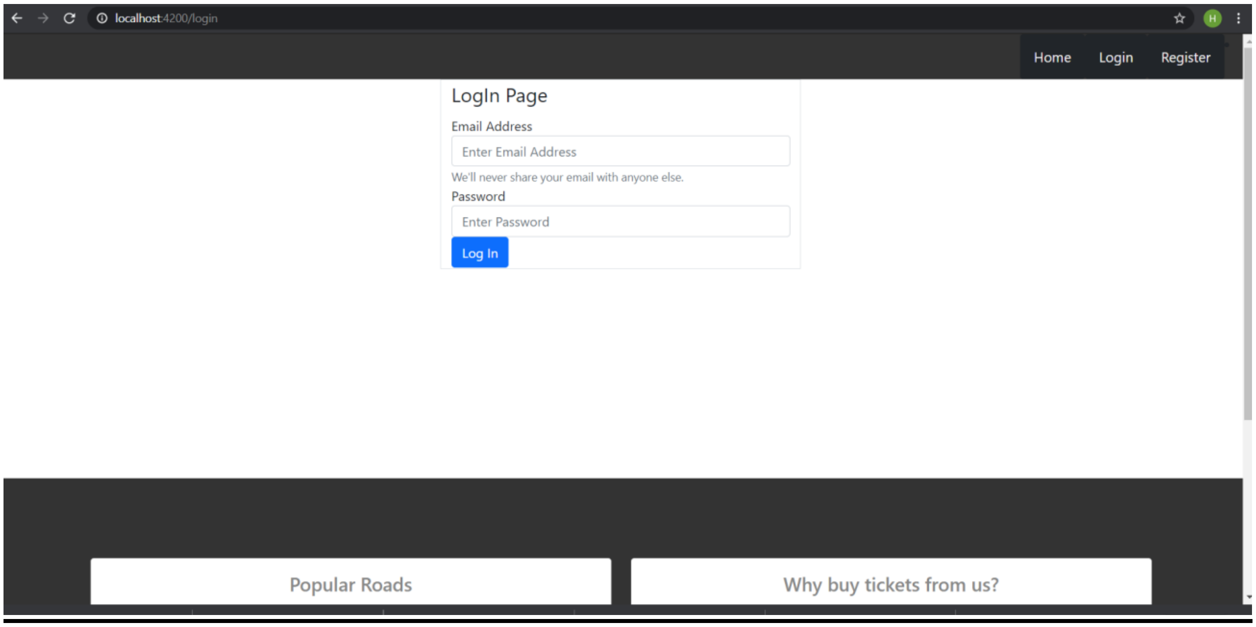
- Pune to Mumbai
- Pune to Bangalore
- Pune to Kolkata
- Pune to Nashik
- Pune to Delhi
- Pune to Chennai
- Pune to Raipur
- Pune to Jalgon

Why buy tickets from us?

- **Buy bus tickets** anytime from anywhere
- Pay by credit card, mobile banking or cash
- Get tickets delivered to your doorstep
- Dependable customer service **(8 AM to 11 PM)**

Copyright © 2020 by Mybus. All Right Reserved

While booking ticket it will ask for login if the Customer is not logged in:



LogIn Page

Email Address

Enter Email Address

We'll never share your email with anyone else.

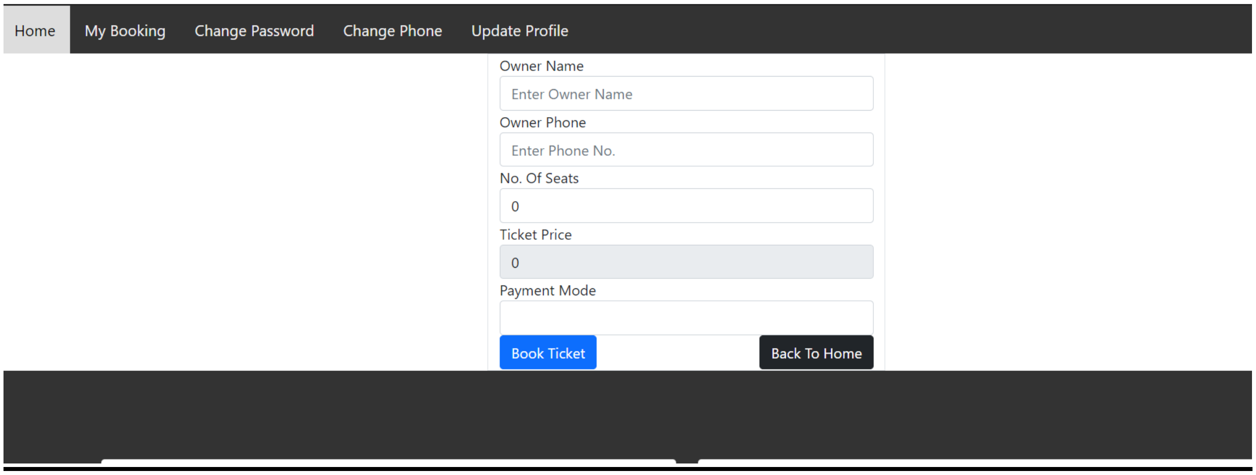
Password

Enter Password

Popular Roads

Why buy tickets from us?

Book Ticket After login:



Owner Name

Enter Owner Name

Owner Phone

Enter Phone No.

No. Of Seats

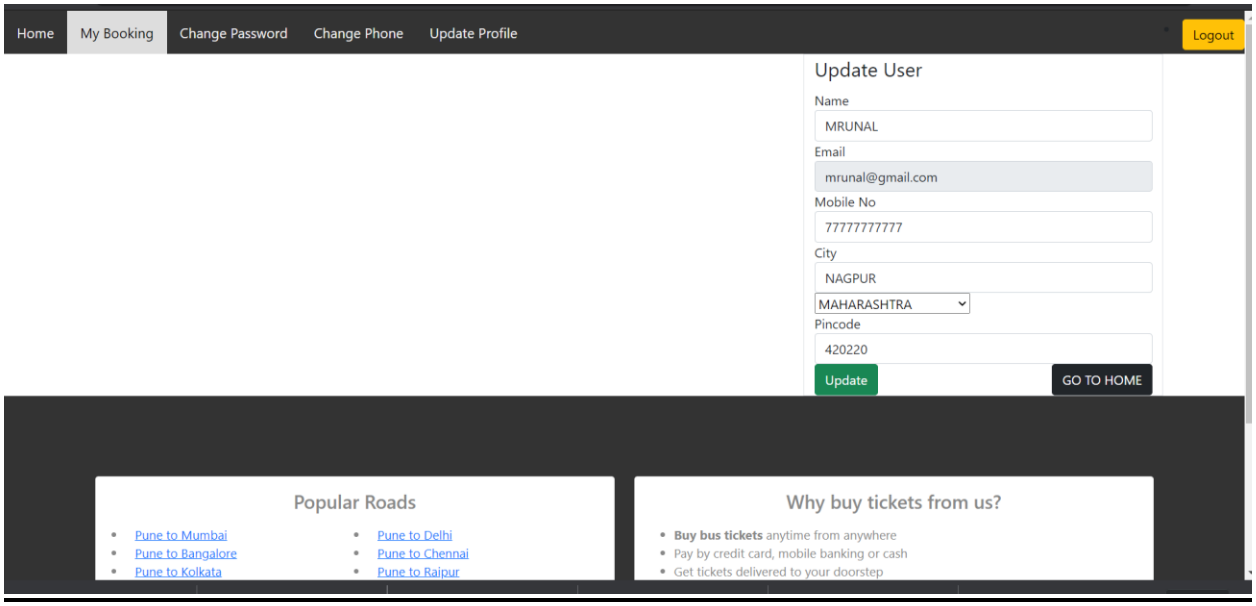
0

Ticket Price

0

Payment Mode

Can update Profile:



Update User

Name

MRUNAL

Email

mrunal@gmail.com

Mobile No

77777777777

City

NAGPUR

MAHARASHTRA

Pincode

420220

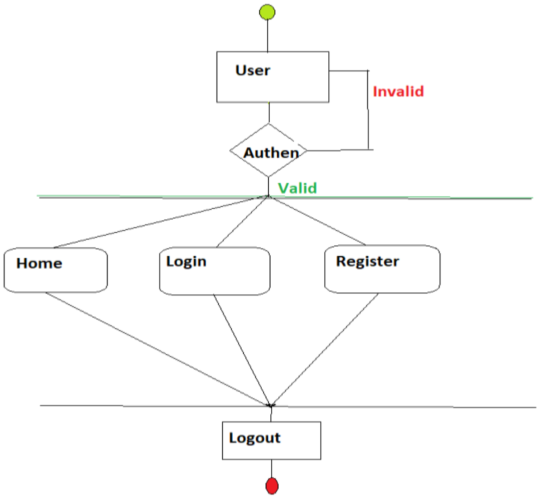
Popular Roads

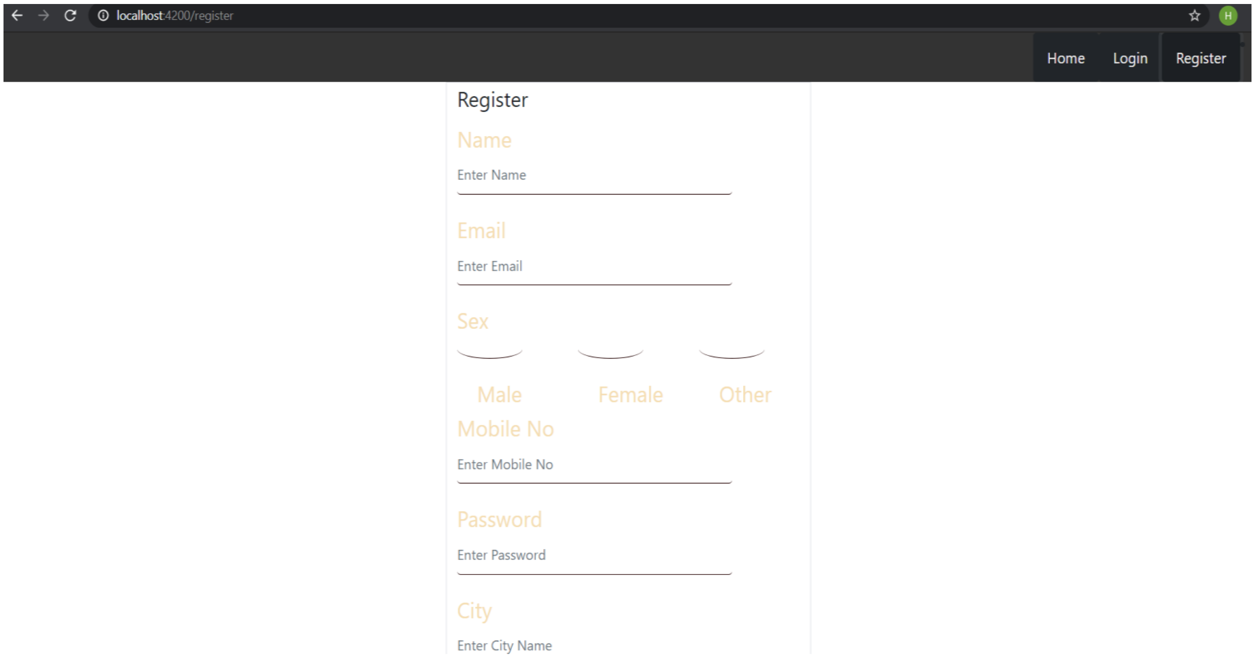
- Pune to Mumbai
- Pune to Bangalore
- Pune to Kolkata
- Pune to Delhi
- Pune to Chennai
- Pune to Raipur

Why buy tickets from us?

- **Buy bus tickets** anytime from anywhere
- Pay by credit card, mobile banking or cash
- Get tickets delivered to your doorstep

C. User Activity Diagram





Register

Name

Enter Name

Email

Enter Email

Sex

Male Female Other

Mobile No

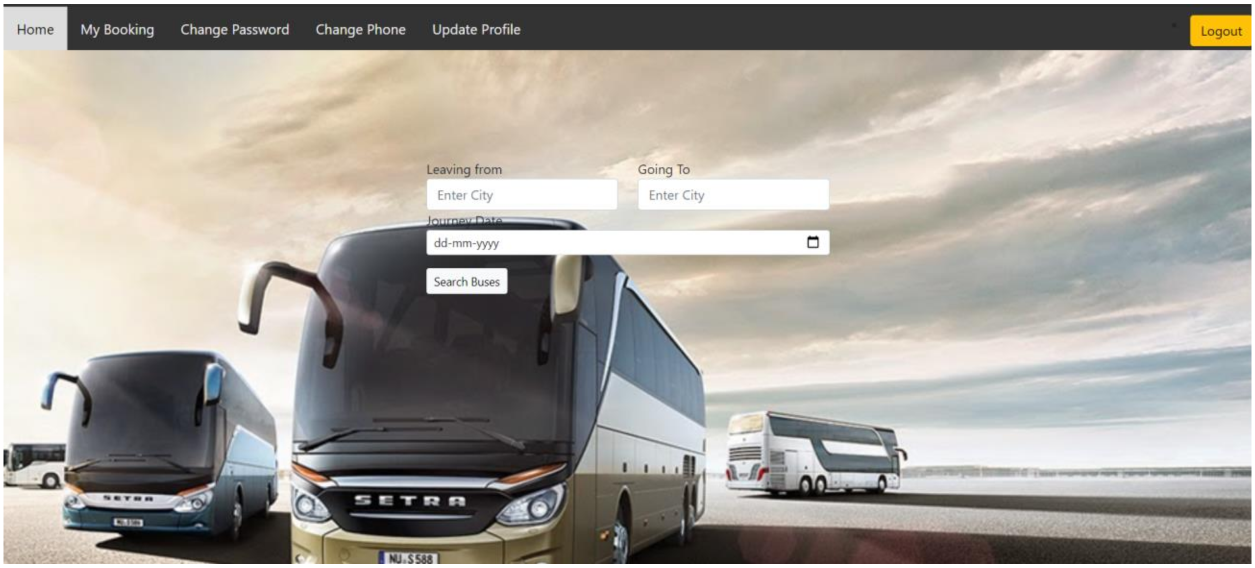
Enter Mobile No

Password

Enter Password

City

Enter City Name



4.2 Data Flow Diagram

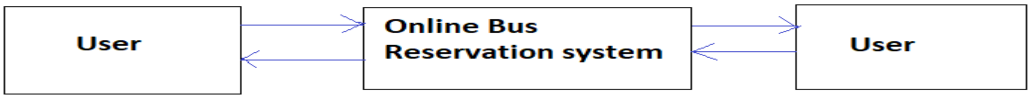


Figure: Level 0 Date Flow Diagram

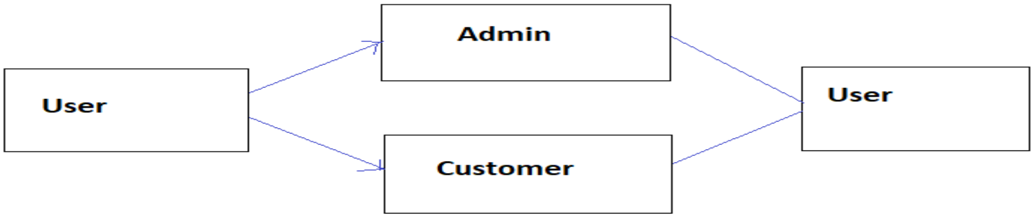


Figure: Level 1 Date Flow Diagram

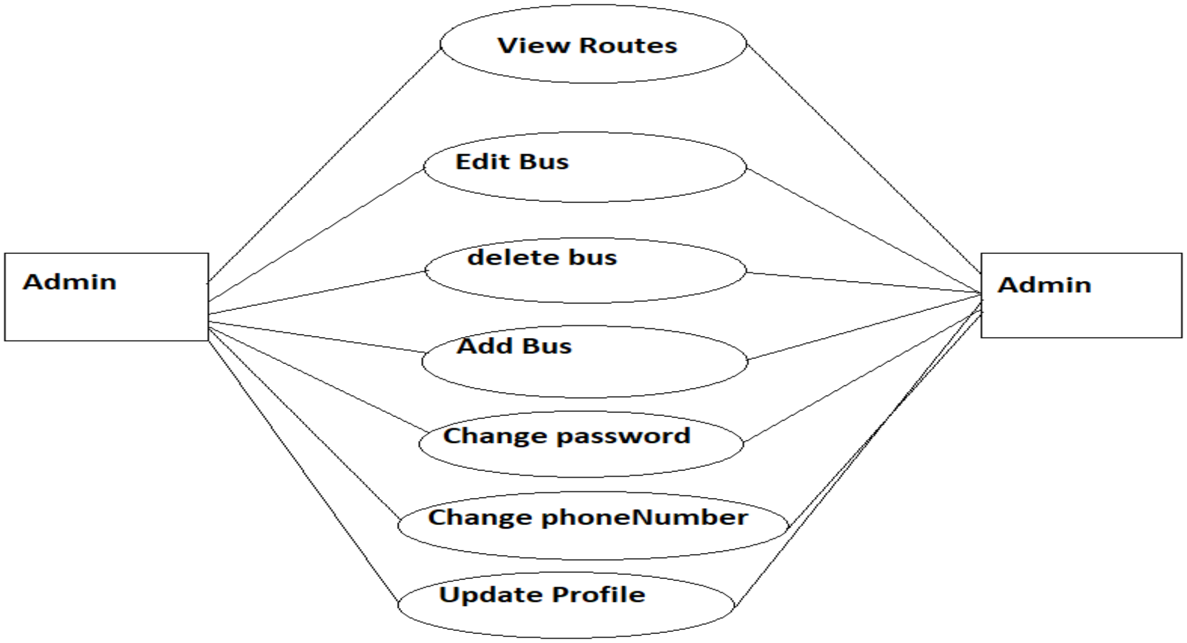
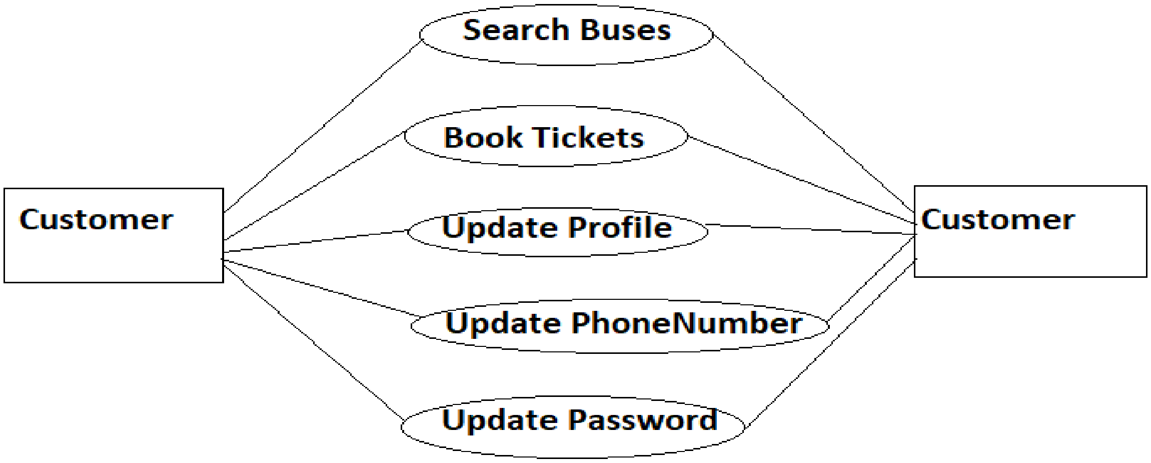


Figure: Level 2 data flow diagram for Admin



Level 2 DataFlow diagram for Customer

Use Case Diagram

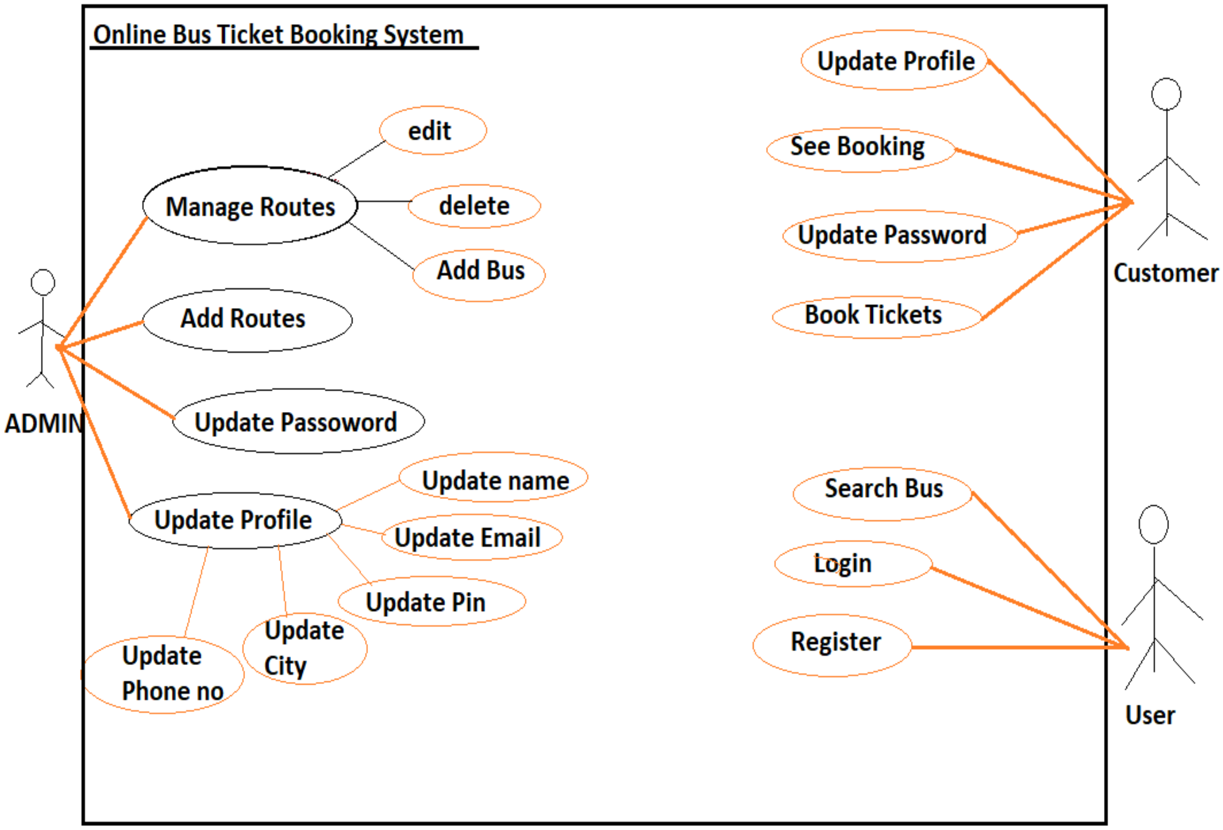
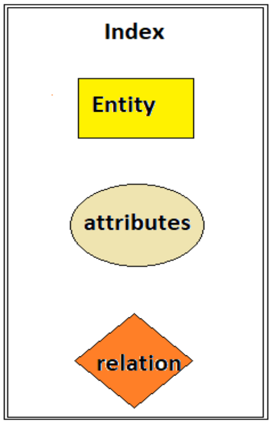
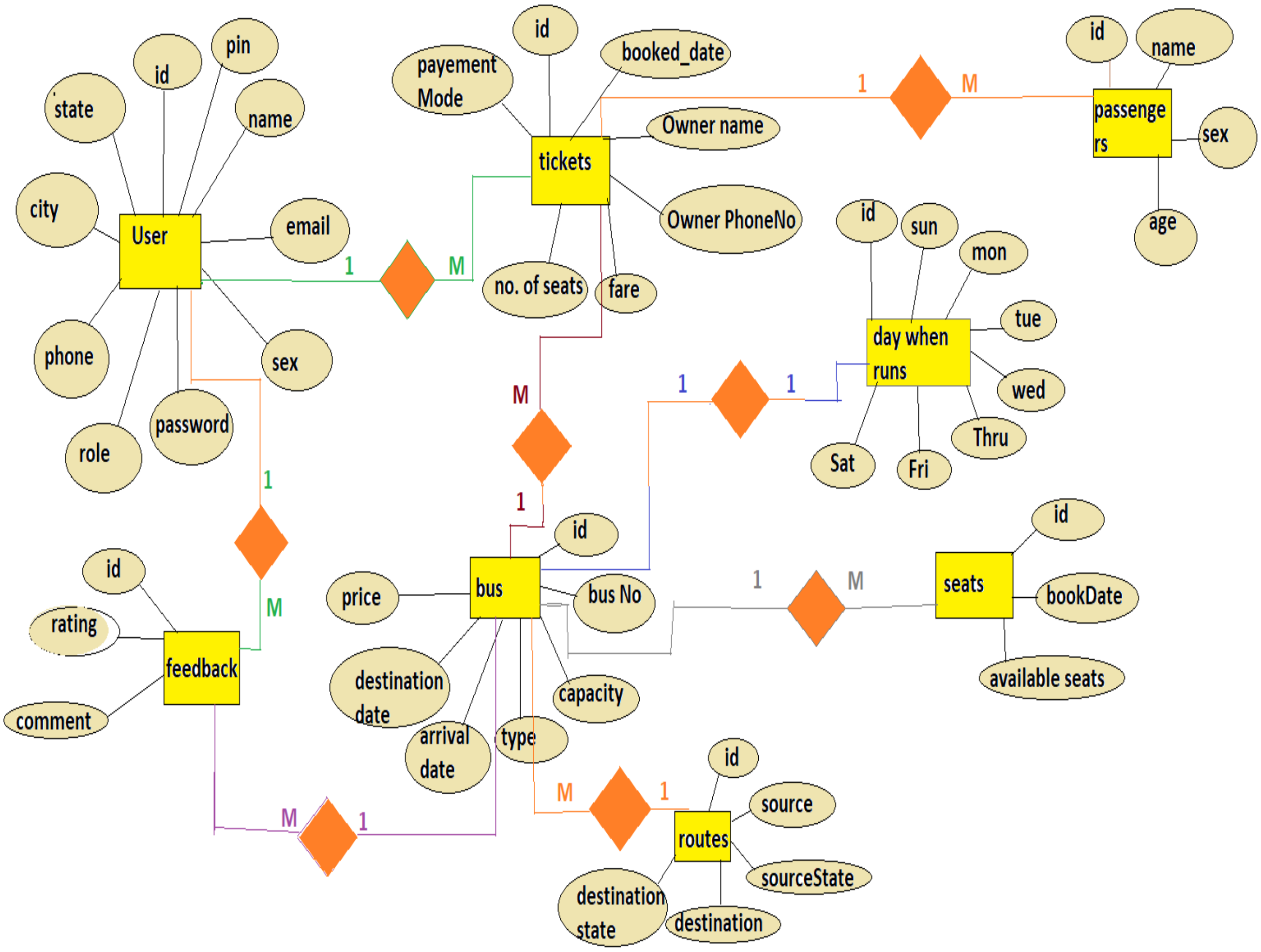


Figure : Use Case Diagram

4.4 ER DIAGRAM





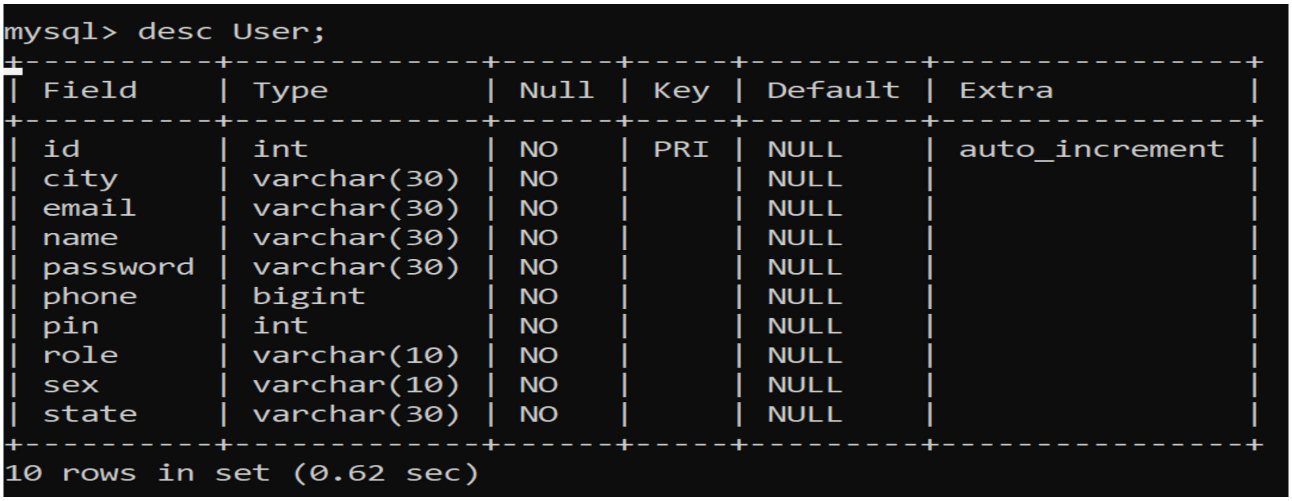
5. Table Structure

1. User Table

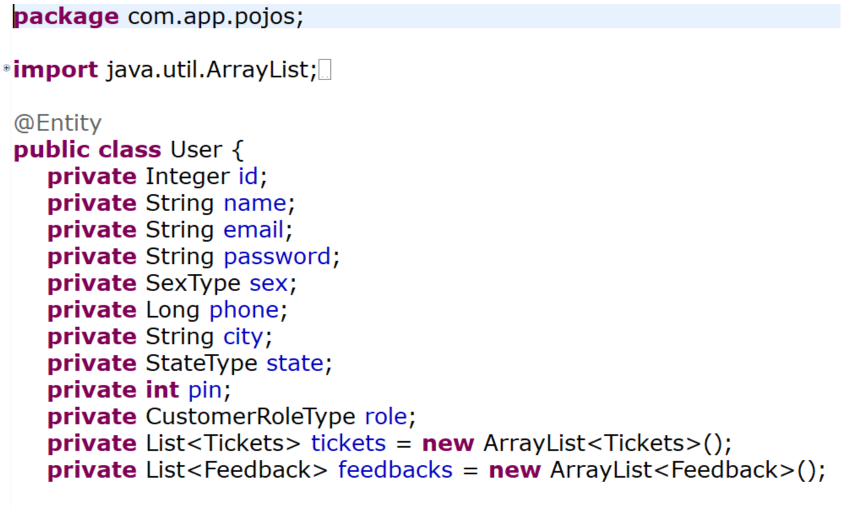
Table Name: user

Primary Key: id

Field	Type	Null	Key	Default	Extra
id	int	NO	PRI	NULL	auto_increment
city	varchar(30)	NO		NULL	
email	varchar(30)	NO		NULL	
name	varchar(30)	NO		NULL	
password	varchar(30)	NO		NULL	
phone	bigint	NO		NULL	
pin	int	NO		NULL	
role	varchar(30)	NO		NULL	
sex	varchar(30)	NO		NULL	
state	varchar(30)	NO		NULL	



```
mysql> desc User;
+----------+-------------+------+-----+---------+----------------+
| Field    | Type        | Null | Key | Default | Extra          |
+----------+-------------+------+-----+---------+----------------+
| id       | int         | NO   | PRI | NULL    | auto_increment |
| city     | varchar(30) | NO   |     | NULL    |                |
| email    | varchar(30) | NO   |     | NULL    |                |
| name     | varchar(30) | NO   |     | NULL    |                |
| password | varchar(30) | NO   |     | NULL    |                |
| phone    | bigint      | NO   |     | NULL    |                |
| pin      | int         | NO   |     | NULL    |                |
| role     | varchar(10) | NO   |     | NULL    |                |
| sex      | varchar(10) | NO   |     | NULL    |                |
| state    | varchar(30) | NO   |     | NULL    |                |
+----------+-------------+------+-----+---------+----------------+
10 rows in set (0.62 sec)
```



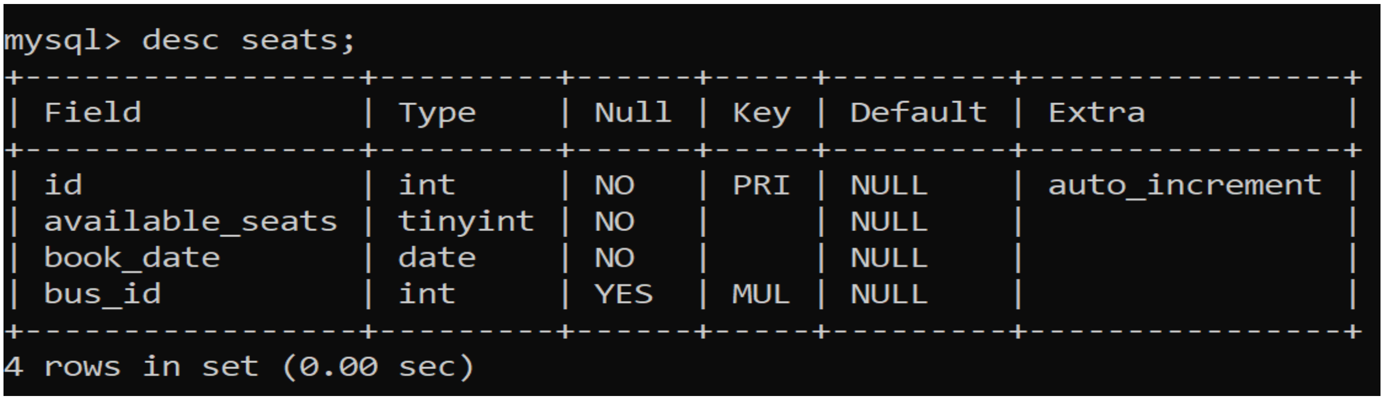
```
package com.app.pojos;

import java.util.ArrayList;

@Entity
public class User {
    private Integer id;
    private String name;
    private String email;
    private String password;
    private SexType sex;
    private Long phone;
    private String city;
    private StateType state;
    private int pin;
    private CustomerRoleType role;
    private List<Tickets> tickets = new ArrayList<Tickets>();
    private List<Feedback> feedbacks = new ArrayList<Feedback>();
```

2.Seats Table

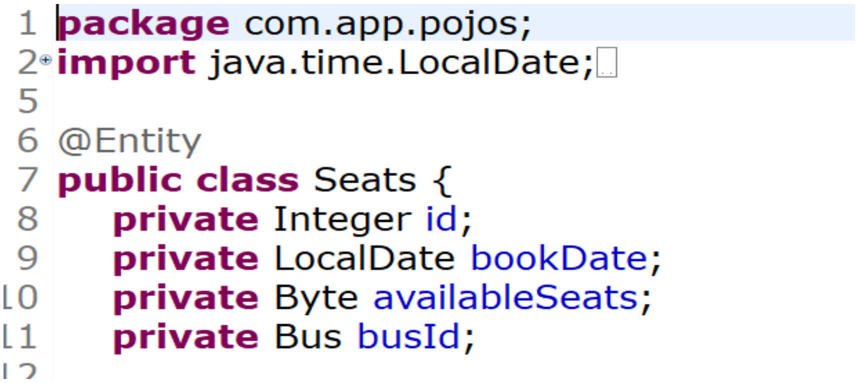
Table Name: seats

Primary Key:id

Field	Type	Null	Key	Default	Extra
id	int	NO	PRI	NULL	auto_increment
available_seats	tinyint	NO		NULL	
book_date	date	NO		NULL	
bus_id	int	YES	FKEY	NULL	



```
mysql> desc seats;
+-----------------+---------+------+-----+---------+----------------+
| Field           | Type    | Null | Key | Default | Extra          |
+-----------------+---------+------+-----+---------+----------------+
| id              | int     | NO   | PRI | NULL    | auto_increment |
| available_seats | tinyint | NO   |     | NULL    |                |
| book_date       | date    | NO   |     | NULL    |                |
| bus_id          | int     | YES  | MUL | NULL    |                |
+-----------------+---------+------+-----+---------+----------------+
4 rows in set (0.00 sec)
```



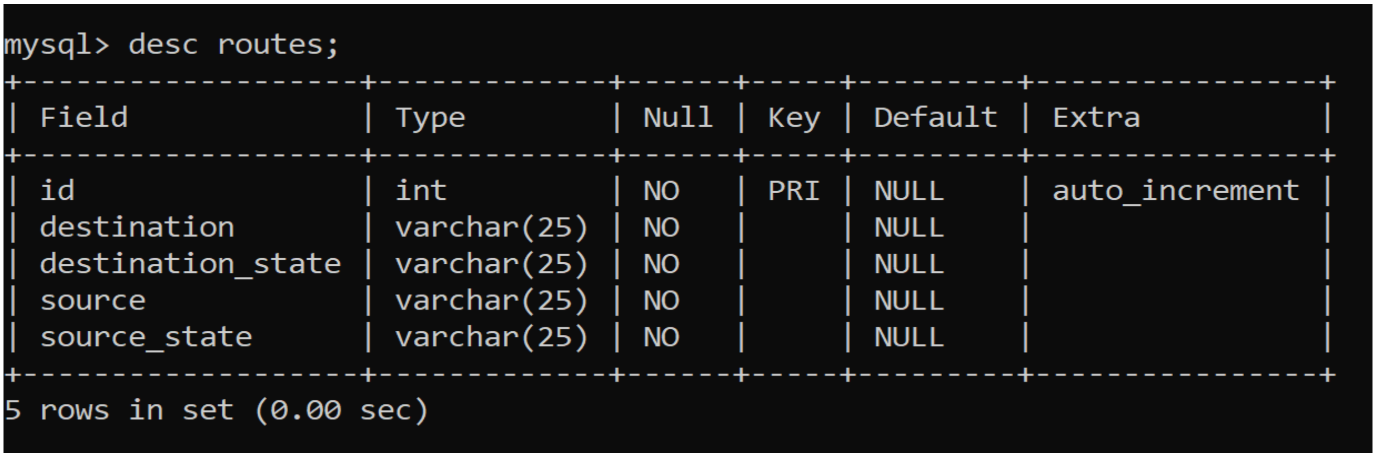
```
package com.app.pojos;
import java.time.LocalDate;

@Entity
public class Seats {
    private Integer id;
    private LocalDate bookDate;
    private Byte availableSeats;
    private Bus busId;
```

3.Routes Table

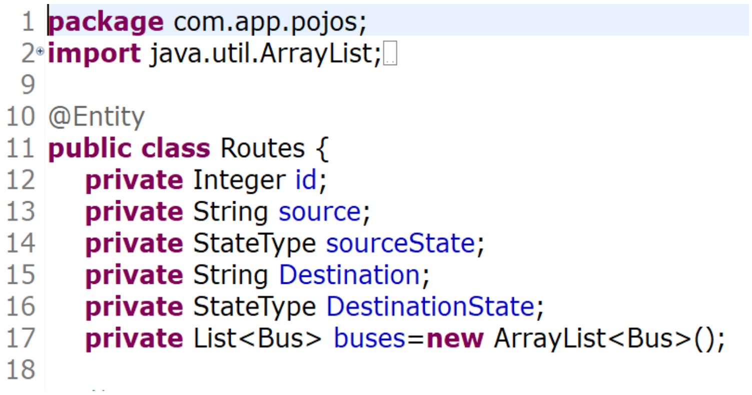
Table Name: routes

Primary Key:id

Field	Type	Null	Key	Default	Extra
id	int	NO	PRI	NULL	auto_increment
destination	varchar(25)	NO		NULL	
destination_state	varchar(25)	NO		NULL	
source	varchar(25)	NO		NULL	
source_state	varchar(25)	NO		NULL	



```
mysql> desc routes;
+-------------------+-------------+------+-----+---------+----------------+
| Field             | Type        | Null | Key | Default | Extra          |
+-------------------+-------------+------+-----+---------+----------------+
| id                | int         | NO   | PRI | NULL    | auto_increment |
| destination       | varchar(25) | NO   |     | NULL    |                |
| destination_state | varchar(25) | NO   |     | NULL    |                |
| source            | varchar(25) | NO   |     | NULL    |                |
| source_state      | varchar(25) | NO   |     | NULL    |                |
+-------------------+-------------+------+-----+---------+----------------+
5 rows in set (0.00 sec)
```



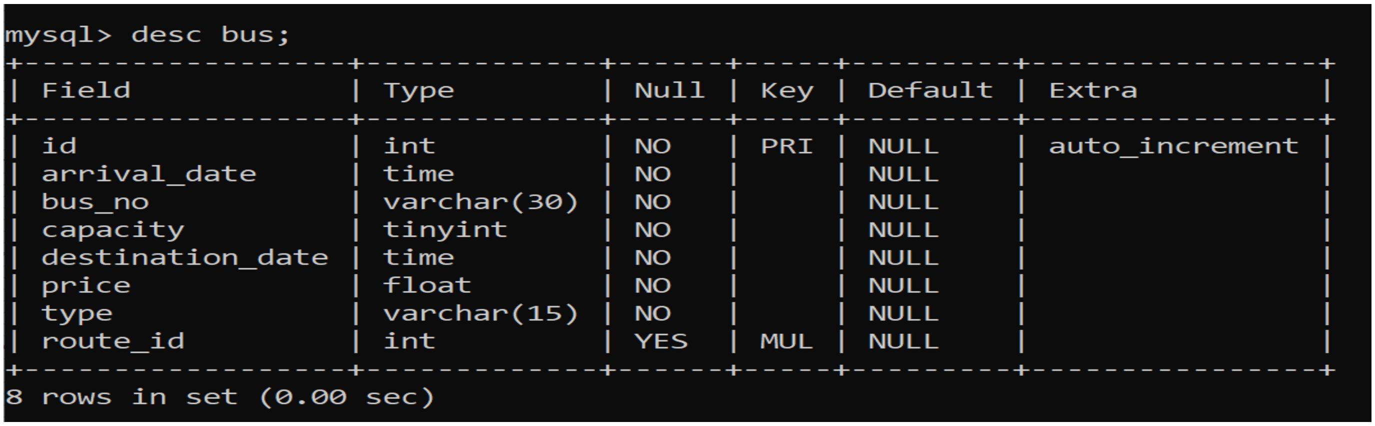
```
package com.app.pojos;
import java.util.ArrayList;

@Entity
public class Routes {
    private Integer id;
    private String source;
    private StateType sourceState;
    private String Destination;
    private StateType DestinationState;
    private List<Bus> buses=new ArrayList<Bus>();

```

4. Bus Table

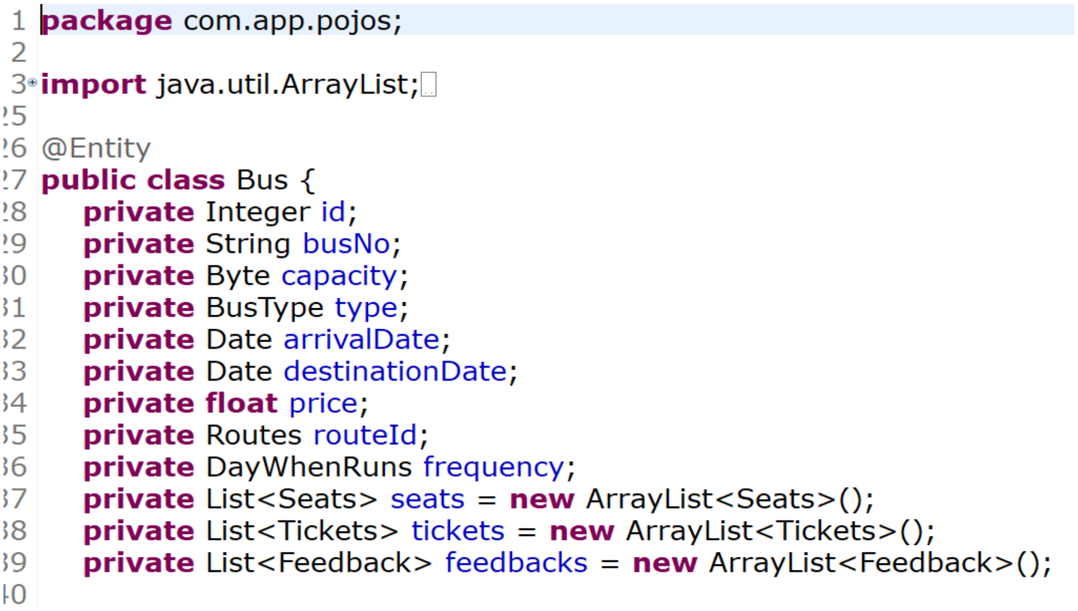
Table Name:bus

Primary Key:id

Field	Type	Null	Key	Default	Extra
id	int	NO	PRI	NULL	auto_increment
arrival_date	time	NO		NULL	
bus_no	varchar(30)	NO		NULL	
capacity	tinyint	NO		NULL	
destination_date	time	NO		NULL	
price	float	NO		NULL	
type	varchar(15)	NO		NULL	
route_id	int	YES	FKEY	NULL	



```
mysql> desc bus;
+------------------+-------------+------+-----+---------+----------------+
| Field            | Type        | Null | Key | Default | Extra          |
+------------------+-------------+------+-----+---------+----------------+
| id               | int         | NO   | PRI | NULL    | auto_increment |
| arrival_date     | time        | NO   |     | NULL    |                |
| bus_no           | varchar(30) | NO   |     | NULL    |                |
| capacity         | tinyint     | NO   |     | NULL    |                |
| destination_date | time        | NO   |     | NULL    |                |
| price            | float       | NO   |     | NULL    |                |
| type             | varchar(15) | NO   |     | NULL    |                |
| route_id         | int         | YES  | MUL | NULL    |                |
+------------------+-------------+------+-----+---------+----------------+
8 rows in set (0.00 sec)
```



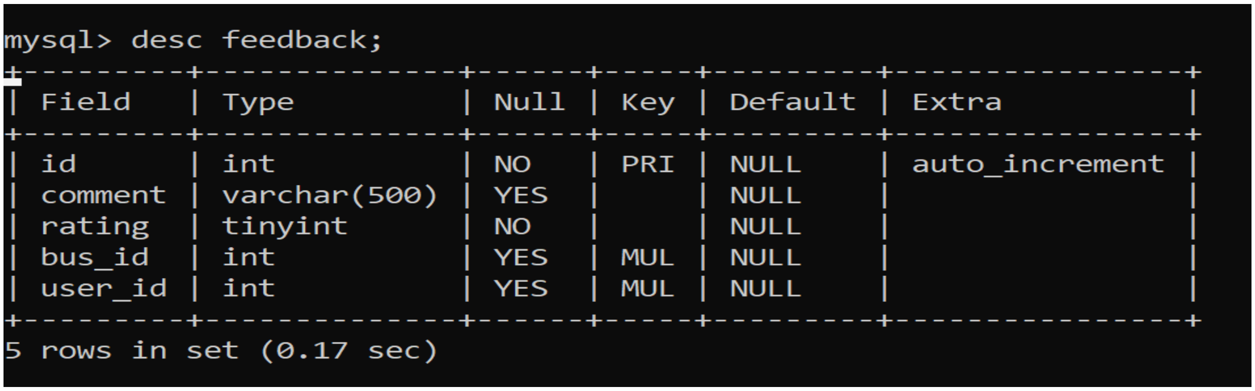
```java
package com.app.pojos;

import java.util.ArrayList;

@Entity
public class Bus {
    private Integer id;
    private String busNo;
    private Byte capacity;
    private BusType type;
    private Date arrivalDate;
    private Date destinationDate;
    private float price;
    private Routes routeId;
    private DayWhenRuns frequency;
    private List<Seats> seats = new ArrayList<Seats>();
    private List<Tickets> tickets = new ArrayList<Tickets>();
    private List<Feedback> feedbacks = new ArrayList<Feedback>();
```

5. Feedback Table

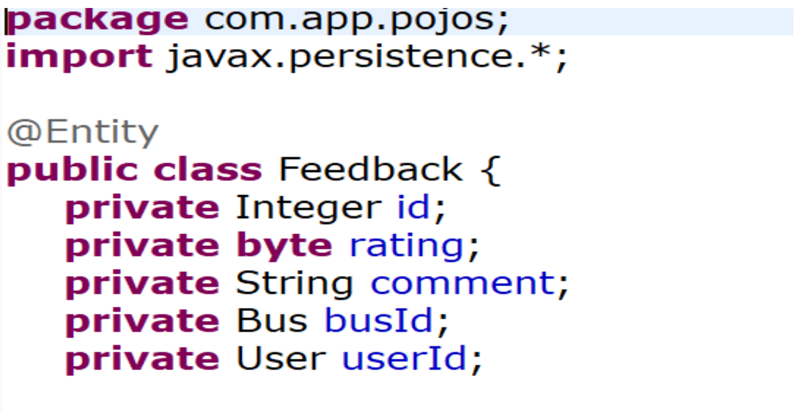
Table Name:feedback

Primary Key:id

Field	Type	Null	Key	Default	Extra
id	int	NO	PRI	NULL	auto_increment
comment	varchar(500)	YES		NULL	
rating	tinyint	NO		NULL	
bus_id	int	YES	FKEY	NULL	
user_id	int	YES	FKEY	NULL	



```
mysql> desc feedback;
+---------+--------------+------+-----+---------+----------------+
| Field   | Type         | Null | Key | Default | Extra          |
+---------+--------------+------+-----+---------+----------------+
| id      | int          | NO   | PRI | NULL    | auto_increment |
| comment | varchar(500) | YES  |     | NULL    |                |
| rating  | tinyint      | NO   |     | NULL    |                |
| bus_id  | int          | YES  | MUL | NULL    |                |
| user_id | int          | YES  | MUL | NULL    |                |
+---------+--------------+------+-----+---------+----------------+
5 rows in set (0.17 sec)
```



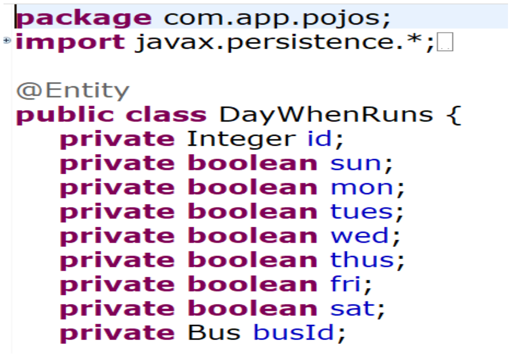
```
package com.app.pojos;
import javax.persistence.*;

@Entity
public class Feedback {
    private Integer id;
    private byte rating;
    private String comment;
    private Bus busId;
    private User userId;
```

6. Day_when_runs Table

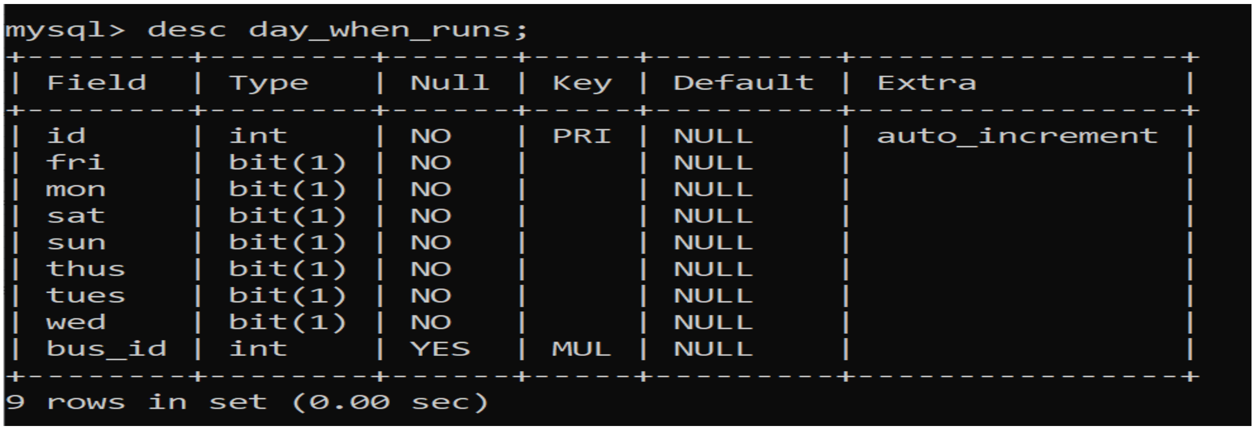
Table Name:day_when_runs

Primary Key:id

Field	Type	Null	Key	Default	Extra
id	int	NO	PRI	NULL	auto_increment
fri	bit(1)	NO		NULL	
mon	bit(1)	NO		NULL	
sat	bit(1)	NO		NULL	
sun	bit(1)	NO		NULL	
thus	bit(1)	NO		NULL	
tues	bit(1)	NO		NULL	
wed	bit(1)	NO		NULL	
bus_id	int	YES	FKEY	NULL	



```java
package com.app.pojos;
import javax.persistence.*;

@Entity
public class DayWhenRuns {
    private Integer id;
    private boolean sun;
    private boolean mon;
    private boolean tues;
    private boolean wed;
    private boolean thus;
    private boolean fri;
    private boolean sat;
    private Bus busId;
```



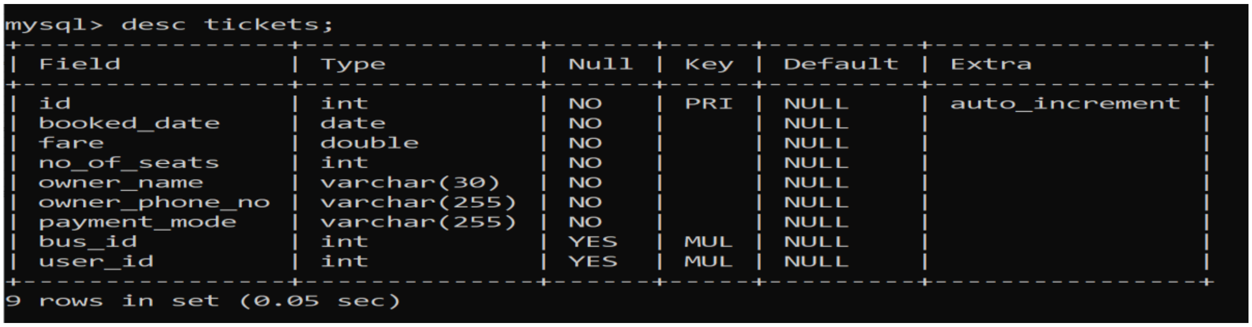
```
mysql> desc day_when_runs;
+---------+--------+------+-----+---------+----------------+
| Field   | Type   | Null | Key | Default | Extra          |
+---------+--------+------+-----+---------+----------------+
| id      | int    | NO   | PRI | NULL    | auto_increment |
| fri     | bit(1) | NO   |     | NULL    |                |
| mon     | bit(1) | NO   |     | NULL    |                |
| sat     | bit(1) | NO   |     | NULL    |                |
| sun     | bit(1) | NO   |     | NULL    |                |
| thus    | bit(1) | NO   |     | NULL    |                |
| tues    | bit(1) | NO   |     | NULL    |                |
| wed     | bit(1) | NO   |     | NULL    |                |
| bus_id  | int    | YES  | MUL | NULL    |                |
+---------+--------+------+-----+---------+----------------+
9 rows in set (0.00 sec)
```

7. Tickets Table

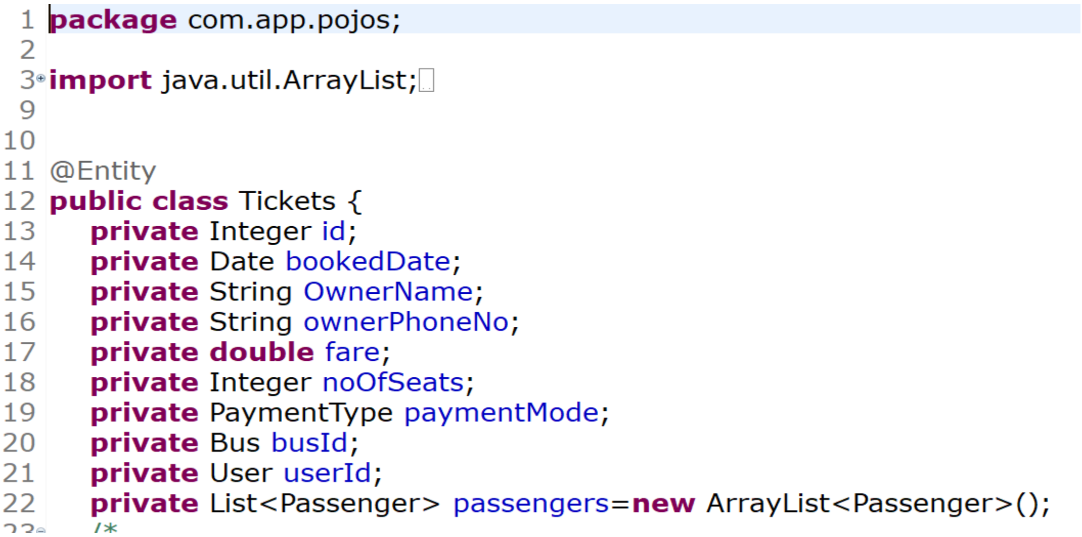
Table name: tickets

Primary Key: id

Field	Type	Null	Key	Default	Extra
id	int	NO	PRI	NULL	auto_increment
booked_date	date	NO		NULL	
fare	double	NO		NULL	
no_of_seats	int	NO		NULL	
owner_name	varchar(30)	NO		NULL	
owner_phone_no	varchar(255)	NO		NULL	
payment_mode	varchar(255)	NO		NULL	
bus_id	int	YES	FKEY	NULL	
user_id	int	YES	FKEY	NULL	



```
mysql> desc tickets;
+-----------------+--------------+------+-----+---------+----------------+
| Field           | Type         | Null | Key | Default | Extra          |
+-----------------+--------------+------+-----+---------+----------------+
| id              | int          | NO   | PRI | NULL    | auto_increment |
| booked_date     | date         | NO   |     | NULL    |                |
| fare            | double       | NO   |     | NULL    |                |
| no_of_seats     | int          | NO   |     | NULL    |                |
| owner_name      | varchar(30)  | NO   |     | NULL    |                |
| owner_phone_no  | varchar(255) | NO   |     | NULL    |                |
| payment_mode    | varchar(255) | NO   |     | NULL    |                |
| bus_id          | int          | YES  | MUL | NULL    |                |
| user_id         | int          | YES  | MUL | NULL    |                |
+-----------------+--------------+------+-----+---------+----------------+
9 rows in set (0.05 sec)
```

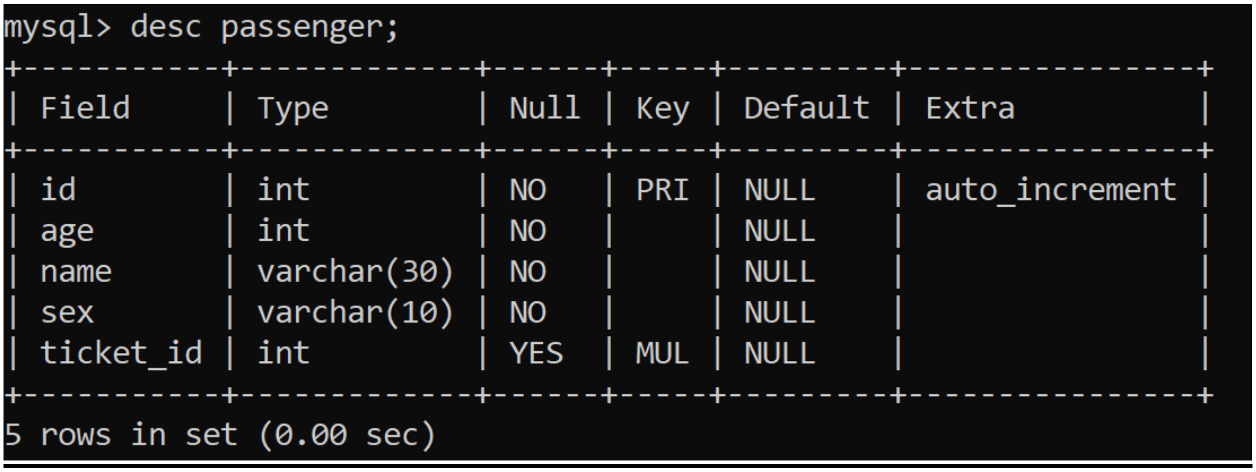


```java
package com.app.pojos;

import java.util.ArrayList;


@Entity
public class Tickets {
    private Integer id;
    private Date bookedDate;
    private String OwnerName;
    private String ownerPhoneNo;
    private double fare;
    private Integer noOfSeats;
    private PaymentType paymentMode;
    private Bus busId;
    private User userId;
    private List<Passenger> passengers=new ArrayList<Passenger>();
    /*
```

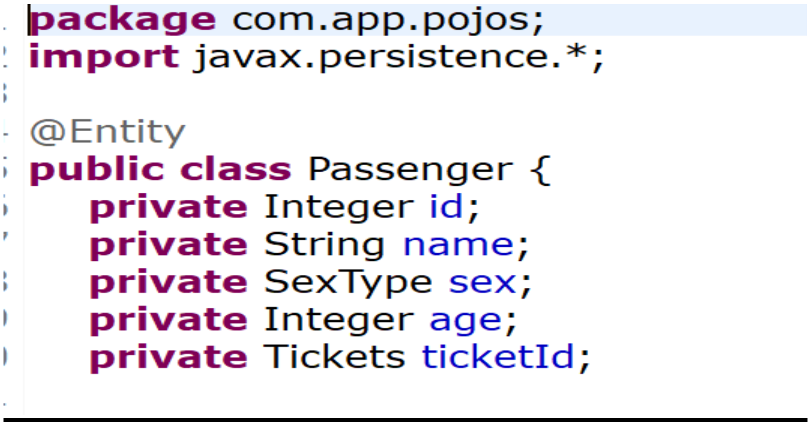
8. Passenger Table

Table name: passenger

Field	Type	Null	Key	Default	Extra
id	int	NO	PRI	NULL	auto_increment
age	int	NO		NULL	
name	varchar(30)	NO		NULL	
sex	varchar(10)	NO		NULL	
ticket_id	int	YES	MUL	NULL	



```
mysql> desc passenger;
+-----------+-------------+------+-----+---------+----------------+
| Field     | Type        | Null | Key | Default | Extra          |
+-----------+-------------+------+-----+---------+----------------+
| id        | int         | NO   | PRI | NULL    | auto_increment |
| age       | int         | NO   |     | NULL    |                |
| name      | varchar(30) | NO   |     | NULL    |                |
| sex       | varchar(10) | NO   |     | NULL    |                |
| ticket_id | int         | YES  | MUL | NULL    |                |
+-----------+-------------+------+-----+---------+----------------+
5 rows in set (0.00 sec)
```



```java
package com.app.pojos;
import javax.persistence.*;

@Entity
public class Passenger {
    private Integer id;
    private String name;
    private SexType sex;
    private Integer age;
    private Tickets ticketId;
```

6 . Conclusion:

In **Online Bus Ticket Booking**, we have developed a secure, user-friendly Bus Reservation System.This Project basically provides a Bus-Ticket information.In our website any user or visitors, can view our system and search the buses on the route and how many seats are available in the buses.

The user can also register there seats in the bus .But user have to compulsorily registered first in the system.To access this system you only need a web browser, internet connection.We observed the working of the Bus reservation system and after going through it, we get to know that there are many operations, which they have to do manually.

To solve the above problem, and further maintaining records of passenger details, seat availability, price per seat, bill generation and other things, we have developed computerized reservation system.

By using this software, we can reserve tickets, via the internet. The customer can check the availability of bus-ticket and reserve selective seats.The validation is done in an efficient manner.

a. ***Future Scope:***

This project can be enhanced further by adding the Agent Module to perform the agent related functionalities,The generated ticket can be sent on the email id of the customer.The website is flexible enough to be modified and implemented as per future requirements.We have tried our best to present this website.Messages and Email alerts for various things can be sent to the Users so that they cannot miss anything.The offers information for various festiv can be send to the User.The payment related things can be upgraded .

7. Reference

1. https://stackoverflow.com
2. https://www.javatpoint.com
3. https:docs//oracle.com
4. Geeksforgeeks
5. https://angular.io/api