

/-----Entity_Class-----/

/-----Booking_Class-----/

```
package com.espire.cabbooking1;
```

```
import java.sql.Timestamp;
```

```
public class Booking {  
    private int bookingId;  
    private Timestamp bookingDateTime ;  
    private String bookingStatus ;  
    private boolean check;  
  
    public Booking() {}  
    public Booking(int bookingId, Timestamp bookingDateTime,  
String bookingStatus, boolean check) {  
        this.bookingId = bookingId;  
        this.bookingDateTime = bookingDateTime;  
        this.bookingStatus = bookingStatus;  
        this.check = check;  
    }  
  
    public int getBookingId() {  
        return bookingId;  
    }  
    public void setBookingId(int bookingId) {  
        this.bookingId = bookingId;  
    }  
    public Timestamp getBookingDateTime() {  
        return bookingDateTime;  
    }  
    public void setBookingDateTime(Timestamp bookingDateTime)  
{  
        this.bookingDateTime = bookingDateTime;  
    }  
    public String getBookingStatus() {  
        return bookingStatus;  
    }  
    public void setBookingStatus(String bookingStatus) {  
        this.bookingStatus = bookingStatus;  
    }  
    public boolean isCheck() {
```

```

        return check;
    }
    public void setCheck(boolean check) {
        this.check = check;
    }
    @Override
    public String toString() {
        return "Booking [bookingId=" + bookingId + ",
bookingDateTime=" + bookingDateTime + ", bookingStatus="
        + bookingStatus + ", check=" + check +
        "];
    }
}

```

/-----cab_classs-----/

```

package com.espire.cabbooking1;

public class Cab {
    private int cabNo;
    private int driverId;
    private String cabModel;
    private String cabColor;
    private String availibility;
    private boolean check;

    public Cab() {}

    public Cab(int cabNo, int driverId, String cabModel,
String cabColor, String availibility, boolean check) {
        this.cabNo = cabNo;
        this.driverId = driverId;
        this.cabModel = cabModel;
        this.cabColor = cabColor;
        this.availibility = availibility;
        this.check = check;
    }

    public int getCabNo() {
        return cabNo;
    }
    public void setCabNo(int cabNo) {
        this.cabNo = cabNo;
    }
    public int getDriverId() {
        return driverId;
    }
    public void setDriverId(int driverId) {
        this.driverId = driverId;
    }
}

```

```

    }
    public String getCabModel() {
        return cabModel;
    }
    public void setCabModel(String cabModel) {
        this.cabModel = cabModel;
    }
    public String getCabColor() {
        return cabColor;
    }
    public void setCabColor(String cabColor) {
        this.cabColor = cabColor;
    }
    public String getAvailability() {
        return availability;
    }
    public void setAvailability(String availability) {
        this.availability = availability;
    }
    public boolean isCheck() {
        return check;
    }
    public void setCheck(boolean check) {
        this.check = check;
    }
}

@Override
public String toString() {
    return "Cab [cabNo=" + cabNo + ", driverId=" +
driverId + ", cabModel=" + cabModel + ", cabColor=" + cabColor
        + ", availability=" + availability + ",
check=" + check + "]";
}

}

// Passenger Class

package com.espire.cabbooking1;

public class Passenger {
    private int passengerId;
    private String passengerName;
    private String passengerPhone;
    private String passengerEmail;

    public Passenger() {}
    public Passenger(int passengerId, String passengerName,
String passengerPhone, String passengerEmail) {
        this.passengerId = passengerId;

```

```

        this.passengerName = passengerName;
        this.passengerPhone = passengerPhone;
        this.passengerEmail = passengerEmail;
    }
    public int getPassengerId() {
        return passengerId;
    }
    public void setPassengerId(int passengerId) {
        this.passengerId = passengerId;
    }
    public String getPassengerName() {
        return passengerName;
    }
    public void setPassengerName(String passengerName) {
        this.passengerName = passengerName;
    }
    public String getPassengerPhone() {
        return passengerPhone;
    }
    public void setPassengerPhone(String passengerPhone) {
        this.passengerPhone = passengerPhone;
    }
    public String getPassengerEmail() {
        return passengerEmail;
    }
    public void setPassengerEmail(String passengerEmail) {
        this.passengerEmail = passengerEmail;
    }
    @Override
    public String toString() {
        return "Passenger [passengerId=" + passengerId + ",
passengerName=" + passengerName + ", passengerPhone="
            + passengerPhone + ", passengerEmail=" +
passengerEmail + "]";
    }
}

```

```

}

```

```

//User class

```

```

package com.espire.cabbooking1;

```

```

public class User {
    private String username;
    private String password;
    private String role;
    private boolean checked;

    public User() {}
    public User(String username, String password, String
role, boolean checked) {

```

```

        this.username = username;
        this.password = password;
        this.role = role;
        this.checked = checked;
    }
    public String getUsername() {
        return username;
    }
    public void setUsername(String username) {
        this.username = username;
    }
    public String getPassword() {
        return password;
    }
    public void setPassword(String password) {
        this.password = password;
    }
    public String getRole() {
        return role;
    }
    public void setRole(String role) {
        this.role = role;
    }
    public boolean isChecked() {
        return checked;
    }
    public void setChecked(boolean checked) {
        this.checked = checked;
    }
    @Override
    public String toString() {
        return "User [username=" + username + ", password="
+ password + ", role=" + role + ", checked=" + checked
        + "]";
    }
}

```

```

}

```

```

//Dao Interfaces

```

```

//Creating AdminDao Interface

```

```

package com.espire.cabbooking1.dao;

```

```

import java.sql.SQLException;

```

```

public interface AdminDao {
    public void viewAllCabs() throws SQLException;
    public void viewAllPassengers() throws SQLException;
    public void viewAllBookings() throws SQLException;
}

```

```

//PassengerDao Interface

package com.espire.cabbooking1.dao;

import java.sql.SQLException;

public interface PassengerDAO {
    void bookTheRide() throws SQLException;
    void cancelTheBooking() throws SQLException;
}

//Implementing of dao interfaces

//Creating AdminDaoImpl

package com.espire.cabbooking1.daoimpl;

import java.sql.SQLException;

import com.espire.cabbooking1.dao.AdminDao;
import com.espire.cabbooking1.serviceimpl.AdminServiceImpl;

public class AdminDaoImpl implements AdminDao{
    AdminServiceImpl adminSerImpl=new AdminServiceImpl();
    @Override
    public void viewAllCabs() throws SQLException{
        adminSerImpl.getAllCabs();
        return;
    }

    @Override
    public void viewAllPassengers() throws SQLException{
        adminSerImpl.getAllPassengers();
        return;
    }

    @Override
    public void viewAllBookings() throws SQLException{
        adminSerImpl.getAllBookings();
        return;
    }
}

//creating PassengerDaoImpl

package com.espire.cabbooking1.daoimpl;

import java.sql.SQLException;
import java.sql.Timestamp;

```

```

import java.util.Scanner;
import com.espire.cabbooking1.dao.PassengerDAO;
import
com.espire.cabbooking1.serviceimpl.PassengerServiceImpl;

public class PassengerDaoImpl implements PassengerDAO{
    PassengerServiceImpl serImpl=new PassengerServiceImpl();
    Scanner scan=new Scanner(System.in);
    public void bookTheRide() throws SQLException {
        System.out.println("Enter booking id");
        int bId=scan.nextInt();
        scan.nextLine();
//        System.out.println("Enter booking Date and time");
        Timestamp ts=new
Timestamp(System.currentTimeMillis());
//        SimpleDateFormat sdf = new SimpleDateFormat("dd-mm-
yyyy");
//        String bDate=scan.nextLine();
        System.out.println("Enter booking Status");
        String bStatus=scan.nextLine();
//        boolean bn = "true" != null;
        serImpl.add(bId,ts,bStatus);
        return;
    }

    public void cancelTheBooking() throws SQLException {
        System.out.println("Enter booking id");
        int bId=scan.nextInt();
        serImpl.delete(bId);
        return;
    }

}

//Creating Service Interface

//Creating AdminService Interface

package com.espire.cabbooking1.service;

import java.sql.SQLException;

import com.espire.cabbooking1.Booking;
import com.espire.cabbooking1.Cab;
import com.espire.cabbooking1.Passenger;

public interface AdminService {
    public Cab getAllCabs()throws SQLException;
    public Passenger getAllPassengers()throws SQLException;
    public Booking getAllBookings()throws SQLException;
}

```

```

//Creating Passenger service Interface

package com.espire.cabbooking1.service;

import java.sql.SQLException;
import java.sql.Timestamp;

public interface PassengerService {
    void add(int bId, Timestamp bDate, String bStatus)throws
SQLException;
    void delete(int bId)throws SQLException;
}

//AdminServiceImpl

package com.espire.cabbooking1.serviceimpl;

import java.sql.Connection;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import com.espire.cabbooking1.Booking;
import com.espire.cabbooking1.Cab;
import com.espire.cabbooking1.Passenger;
import com.espire.cabbooking1.connection.DBConnection;
import com.espire.cabbooking1.service.AdminService;

public class AdminServiceImpl implements AdminService{
    DBConnection connection = new DBConnection();

    @Override
    public Cab getAllCabs() throws SQLException{
        String sql="select*from Cab";
        Connection con = connection.getCon();
        Statement st = con.createStatement();
        ResultSet rs=st.executeQuery(sql);
        while(rs.next()) {
            System.out.println(rs.getInt(1)+"
"+rs.getInt(2)+" "+rs.getString(3)+" "+rs.getString(4)+"
"+rs.getString(5));
        }
        return null;
    }

    @Override
    public Passenger getAllPassengers() throws SQLException {
        String sql="select*from Passenger";
        Connection con = connection.getCon();
        Statement st = con.createStatement();
        ResultSet rs=st.executeQuery(sql);
        while(rs.next()) {

```



```

        System.out.println(rs.getInt(1)+"
"+rs.getString(2)+" "+rs.getString(3)+" "+rs.getString(4));
    }
    return null;
}

```

```

@Override
public Booking getAllBookings() throws SQLException {
    String sql="select*from Booking";
    Connection con = connection.getCon();
    Statement st = con.createStatement();
    ResultSet rs=st.executeQuery(sql);
    while(rs.next()) {
        System.out.println(rs.getInt(1)+"
"+rs.getString(2)+" "+rs.getString(5));
    }
    return null;
}

```

```

}

```

```

//PassengerServiceImpl

```

```

package com.espire.cabbooking1.serviceimpl;
import java.sql.Connection;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.sql.Timestamp;
import java.util.Date;
import java.util.Scanner;
import com.espire.cabbooking1.connection.DBConnection;
import com.espire.cabbooking1.service.PassengerService;

```

```

public class PassengerServiceImpl implements PassengerService{
    DBConnection connection = new DBConnection();
    Scanner scan=new Scanner(System.in);
    public void add(int bId, Timestamp date, String bStatus)
throws SQLException {
        String sql = "insert into
booking(bookingId,bookingDateTime,bookingStatus) values(" +
bId + "," +
                + date + "','"+ bStatus + "');"
        Connection con = connection.getCon();
        Statement st = con.createStatement();
        int n = st.executeUpdate(sql);
        if (n >= 0)
            System.out.println(" Cab Booked");
        return;
    }
}

```

```

        public void delete(int bId) throws SQLException {
            String sql = "delete from booking where
bookingId="+bId;
            Connection con = connection.getCon();
            Statement st = con.createStatement();
            int n = st.executeUpdate(sql);
            if (n >= 0)
                System.out.println("Ride Cancelled");
            return;
        }
    }

//Creating UserInterface
//Including Main class
package com.espire.cabbooking1.ui;
import java.sql.SQLException;
import java.util.Scanner;

import com.espire.cabbooking1.connection.DBConnection;
import com.espire.cabbooking1.daoimpl.AdminDaoImpl;
import com.espire.cabbooking1.daoimpl.PassengerDaoImpl;

public class LoginUI {
    public static void main(String[] args) throws SQLException
    {
        DBConnection connection = new DBConnection();
        PassengerDaoImpl impl=new PassengerDaoImpl();
        AdminDaoImpl adminImpl=new AdminDaoImpl();
        Scanner scan=new Scanner(System.in);
        int n;
        do {
            System.out.println("Select the Number\n"
                + "1. Admin\n"
                + "2. User\n");
            n=scan.nextInt();
            switch(n) {
            case 1:
                System.out.println("Enter id:");
                int id=scan.nextInt();
                System.out.println("Enter password");
                int pass=scan.nextInt();
                if(id==1 && pass== 1)
                {
                    char ch;
                    System.out.println("Select the Chararacter\n"
                        + "A. View all cabs\n"
                        + "B. View all passengers\n"
                        + "C. View all bookings\n");
                    ch=scan.next().charAt(0);

```

```

        switch(ch) {
        case 'A':
            adminImpl.viewAllCabs();
            break;
        case 'B':
            adminImpl.viewAllPassengers();
            break;
        case 'C':
            adminImpl.viewAllBookings();
            break;
        }
    }
    else {
        System.out.println("Wrong credentials");
    }
    break;
case 2:
    char ch;
    System.out.println("Select the Character\n"
        + "A. Book The Ride\n"
        + "B. Cancel The Ride\n");
    ch=scan.next().charAt(0);
    switch(ch) {
    case 'A':
        impl.bookTheRide();
        break;
    case 'B':
        impl.cancelTheBooking();
        break;
    }
    break;
default:
    System.out.println("Not a valid choice");
    break;
    }
    }while(true);
}
}

```

//creating DBConnection to connect with the database

```
package com.espire.cabbooking1.connection;
```

```
import java.sql.Connection;
```

```
import java.sql.DriverManager;
```

```
import java.sql.SQLException;
```

```

public class DBConnection {

    String driver="com.mysql.cj.jdbc.Driver";

    String dburl="jdbc:mysql://localhost:3306/cabbooking";

    String user="root";

    String password="root";

    Connection con;

    public void createConnection() {

        try {

            Class.forName(driver);

            con=DriverManager.getConnection(dburl,user,password);

            //Statement st=con.createStatement();

            if(con!=null) {

                System.out.println("Connection successfull");

                //Statement st=con.createStatement();

                // return con;

            }

            else {

                System.out.println("Connection failed");

            }

        }

        catch(ClassNotFoundException | SQLException e) {

            System.out.println(e.getMessage());

        }

    }

    public Connection getCon() {

        try {

            Class.forName(driver);

            con=DriverManager.getConnection(dburl,user,password);

        }

        catch(ClassNotFoundException | SQLException e) {

            System.out.println(e.getMessage());

        }

    }

}

```

```
    }  
    return con;  
}  
}
```