Project Report

On

Flight Reservation System

Mphasis Training Project

By

Namrata Singh

Under the Guidance of

Manoj Tekwani

Session oct- nov,2019

INTRODUCTION

Flight ReservationSystem will hold flight schedules and its fare tariffs, passenger reservations and ticket records. It saves time asit allows online procedure as users no longer to wait in a queue to book the flights. It is automaticallygenerated by the server. Admin is the main authority who can do addition, deletion, and modification of flights ifrequired.

The project has been planned to behaving the view of distributed architecture, with centralized storage of the database. The application forthe storage of the data has been planned. Using the constructs of DatabaseMysqland all the user interfaces have been designed using the Adobe Dreamweavertechnologies.

The database connectivity isplanned usingthe "SQL Connection" methodology. The standards of securityand data protective mechanism have been given a big choice for proper usage. The application takes care ofdifferentmodulesandtheirassociated reports, which are produced as per the applicable strategies and standards that are put forwarded by the administrativestaff.

The entire project has been developed keeping in viewof the distributed client server computing technology, in mind.Thespecificationhasbeennormalizedupto2NF toeliminate all the anomalies that may arise dueto thedatabase transaction that are executed by the general users and the organizational administration. The userinterfaces are browser specific togive distributed accessibility for the overall system. The internal database has been selected as DatabaseMysql.

This project also covers various features like onlineregistration of the users, modifying the details of the website by the management staff or administrator of the website, by adding, deleting or modifying the customer details, flights or packages information. In general, this website would be designed to perform like any otherflight ticketing website availableonline.

ProductFunctions

The websitewill allow accessonly toauthorized users with specific roles (Administrator-maintains the websiteCustomr-Register the passengers, Passenger- Fills thedetails).

Following are the System Functions:

Passenger/Customer role:

On the register form, passenger should enter all their detail such as their name, passport number, Email and contactnumber.

Administration role

The system administrator must be able to: add, update and modify flights, routes of the flights and schedules of the flightsand viewthe customerdetails.

3.1. HardwareRequirements:

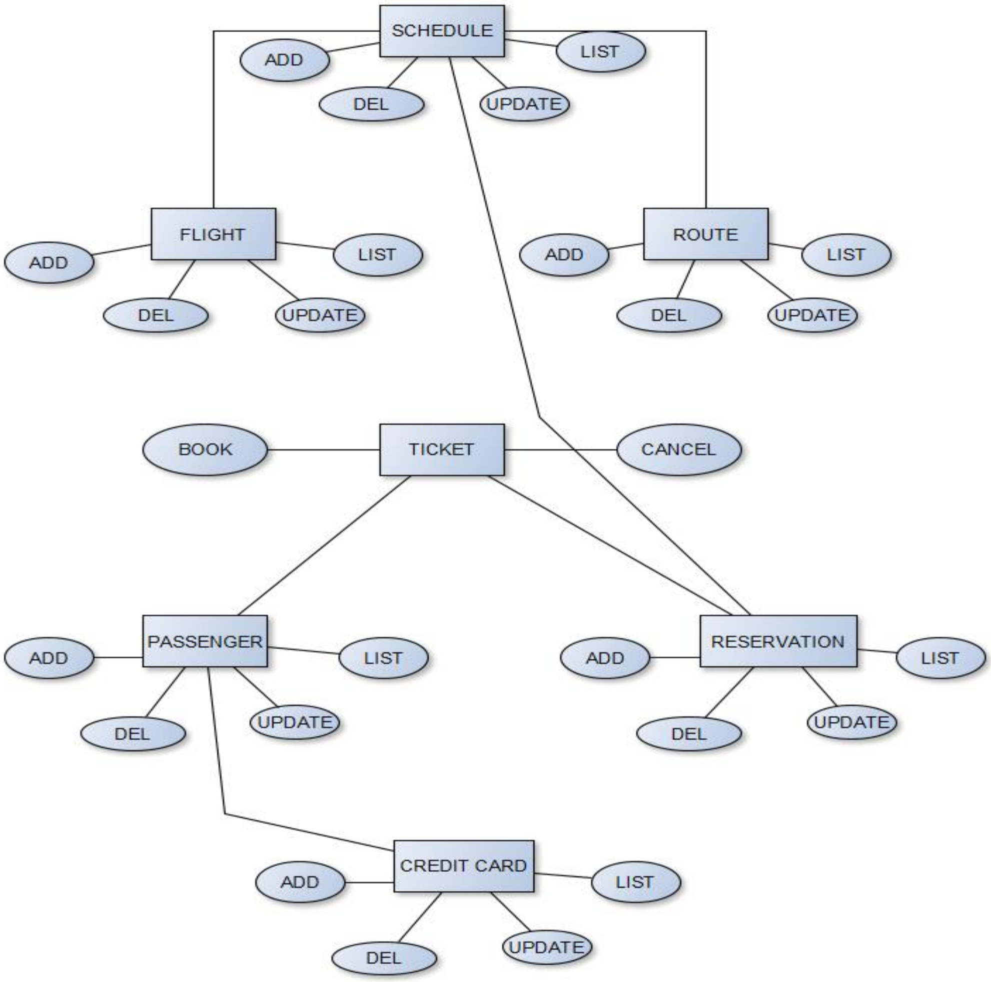
- Intel I3 2.8 GHz Processor andAbove
- RAM 1 GB andAbove
- HDD 20 GB Hard Disk Space andAbove

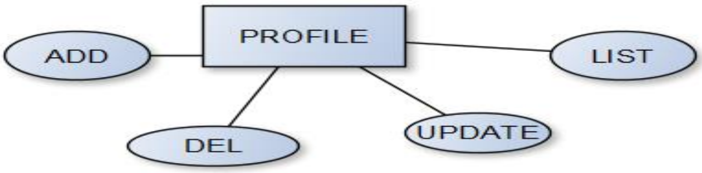
3.2. SoftwareRequirements:

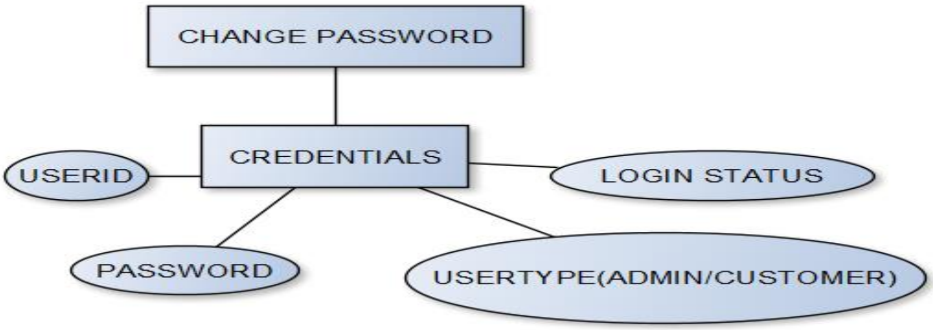
- WINDOWS OS (Windows 7, 8 ,10) OrLinux
- AdobeDreamweaver
- PHP Wamp Server ForWindows(Apache)
- Database sqlForBackend.
- Eclipse Oxygen.
- Microsoft Visual Studio for Frontend

Server side An Apache Web server will accept all requests from the client. A development database will be hosted locally (using MySQL); the production database is hosted centrally.

TableStructure







PACKAGE SPECIFICATION

1. Service: com.mphasis.frs.service
2. Bean: com.mphasis.frs.bean
3. DAO: com.mphasis.frs.dao
4. REST Controller: com.mphasis.frs.control
5. Interfaces: com.mphasis.frs.iFRS

BEAN SPECIFICATION (Package: com.mphasis.frs.bean)

1. ProfileBean (Table: User Profile [FRS_TBL_USER_PROFILE]) {Users}

1. userID (string)
2. firstName (string)
3. lastName (string)
4. dateOfBirth (util.Date)
5. gender (string)
6. street (string)
7. location (string)
8. city (string)
9. state (string)
10. pincode (string)
11. mobileNo (string)
12. emailID (string)
13. password (string)

2. CredentialsBean (Table: Credentials [FRS_TBL_USER_CREDENTIALS]) {Login}

1. userID (string)
2. password (string)
3. userType (string) ['A', 'C', 'FAIL', 'INVALID']
4. loginStatus (string) ['on', 'off']

3. FlightBean (Table: Flights [FRS_TBL_FLIGHT]) {Individual flight}

1. flightID (string)
2. flightName (string)
3. seatingCapacity (int) {Total capacity}
4. reservationCapacity (int) {Current available}

4. ScheduleBean (Table: Schedules [FRS_TBL_SCHEDULE]) {Planes per Flight}

1. scheduleID (string)
2. flightID (string)
3. routeID (string)
4. travelDuration (int)
5. availableDays (string)
6. departureTime (string) {check Validation}

5. ReservationBean (Table: Reservations [FRS_TBL_RESERVATION]) {Booking Details}

1. reservationID (string)
2. userID (string)
3. scheduleID (string)
4. reservationType (string)
5. bookingDate (util.date)
6. journerDate (util.date)
7. noOfSeats (int) {no of seats booked}
8. totalFare (double)
9. bookingStatus (int) [0, 1]

6. PassengerBean (Table: Passengers [FRS_TBL_PASSENGER]) {All passenger list}

1. reservationID (string)
2. name (string) {passenger}
3. gender (string)
4. age (int)
5. seatNo (int)

7. RouteBean (Table: Routes [FRS_TBL_ROUTE]) {Unique flight}

1. routeID (string)
2. source (string)
3. destination (string)
4. distance (int)
5. fare (double)

INTERFACE SPECIFICATION (Package: com.mphasis.frs.iFRS)

1. (INT) IAdminstrator – Service

1. (M) string addFlight(FlightBean flightBean) ['SUCCESS', 'FAIL', 'ERROR']
2. (M) string modifyFlight(FlightBean flightBean)
3. (M) int removeFlight(ArrayList<String> flightID)
4. (M) string addSchedule(ScheduleBean scheduleBean) ['SUCCESS', 'FAIL', 'ERROR']
5. (M) string modifySchedule(ScheduleBean scheduleBean)
6. (M) int removeSchedule(ArrayList<String> scheduleID)
7. (M) string addRoute(RouteBean routeBean) ['SUCCESS', 'FAIL', 'ERROR']
8. (M) string modifyRoute(RouteBean routeBean)
9. (M) int removeFlight(ArrayList<String> routeID)
10. (M) FlightBean viewByFlightId(String flightId)
11. (M) RouteBean viewByRouteId(String routeId)
12. (M) ArrayList<FlightBean> viewByAllFlights()
13. (M) ArrayList<RouteBean> viewByAllRoute()
14. (M) ArrayList<ScheduleBean> viewByAllSchedule()
15. (M) ScheduleBean viewByScheduleId(String scheduleId)
16. (M) ArrayList<PassengerBean> viewPassengersByFlight(String scheduleId)

2. (INT) ICustomer – Service

1.(M) ArrayList<ScheduleBean> viewScheduleByRoute(String source, String destination, Date date)

2. (M) String reserveTicket(ReservationBean reservationBean, ArrayList<PassengerBean> passengers)

3. (M) String cancelTicket(String reservationId)

4. (M) Map<ReservationBean,PassengerBean> viewTicket(String reservationId)

5. (M) Map<ReservationBean,PassengerBean> printTicket(String reservationId)

4. (C) xyzDAO

1. (M) String createXYZ(BeanObject)
2. (M) int deleteXYZ(ArrayList<String>)
3. (M) String updateXYZ(BeanObject)
4. (M) BeanObject findByID(String)
5. (M) ArrayList<BeanObject> findAll()

5. (INT) IAuthentication

1. (M) String authenticate(CredentialsBean credentialsBean)
2. (M) String authorize(String userId)
3. (M) String changeLoginStatus(CredentialsBean credentialsBean, int loginStatus)

6. (INT) IUser

1. (M) String login(CredentialsBean credentialsBean) ["A", "C", "FAIL", "INVALID"]
2. (M) String logout(String userId)
3. (M) String changePassword(CredentialsBean credentialsBean, String newPassword) ["SUCCESS", "FAIL", "INVALID"]
4. (M) String register(ProfileBean profileBean) [<userId of length 6>, "FAIL"] {userId-> first 2 letter of first name followed by 4 digit auto generated number}

7. (INT) IPayment

1. String creditCardNumber
2. String validFrom
3. String validTo
4. int balanc
5. (M) String findByCardNumber(String userid, String cardnumber)
6. (M) String process(Payment payment) {Rename to paymentStatus()}

FRS Table Creation Script

```
1. create table FRS_TBL_User_Credentials(
Userid varchar2(6),
Password varchar2(20) NOT NULL,
Usertype varchar2(1),
Loginstatus number(1),
PRIMARY KEY(Userid));
```

```
2.create table FRS_TBL_User_Profile(
Userid varchar2(6) PRIMARY KEY,
Firstname varchar2(20) NOT NULL,
Lastname varchar2(20) NOT NULL,
DateOfBirth date NOT NULL,
Gender varchar2(7) NOT NULL,
Street varchar(30) NOT NULL,
Location varchar2(20) NOT NULL,
City varchar2(20) NOT NULL,
State varchar2(20) NOT NULL,
PinCode varchar2(10),
MobileNo varchar(10),
EmailId varchar2(30) NOT NULL,FOREIGN KEY(Userid) REFERENCES
FRS_TBL_User_Credentials(Userid));
```

```
3. create table FRS_TBL_CreditCard(CreditCardNumber varchar(16),
   ValidFrom varchar(7) NOT NULL,
   ValidTo varchar(7) NOT NULL,
   CreditBalance number NOT NULL,
   Userid varchar2(6), PRIMARY KEY(CreditCardNumber), FOREIGN KEY(Userid) REFERENCES
FRS_TBL_User_Credentials(Userid));
```

```
 4. create table FRS_TBL_Route(Routeid varchar(8) PRIMARY KEY,
   Source varchar(20) NOT NULL,
   Destination varchar(20) NOT NULL,
   Distance number NOT NULL,
   Fare number NOT NULL);
```

```
5. create table FRS_TBL_Flight(FlightId varchar(6) PRIMARY KEY,
   FlightName varchar(15) NOT NULL,
   Seatingcapacity number(5) NOT NULL,
   Reservationcapacity number(5) NOT NULL);
```

```
6. create table FRS_TBL_Schedule(Scheduleid varchar(8) PRIMARY KEY,
  Flightid varchar(6) NOT NULL,
   Routeid varchar(8) NOT NULL,
   TravelDuration number NOT NULL,
   AvailableDays varchar(30) NOT NULL,
   DepartureTime varchar(4) NOT NULL,
   FOREIGN KEY(Flightid) REFERENCES FRS_TBL_Flight(Flightid),
   FOREIGN KEY(Routeid) REFERENCES FRS_TBL_Route(Routeid));
```

```
7.  create table FRS_TBL_Reservation(Reservationid varchar(8) PRIMARY KEY,
    Userid varchar(6) NOT NULL,
    Scheduleid varchar(8) NOT NULL,
    Reservationtype varchar(20) NOT NULL,
    Bookingdate date NOT NULL,
    Journeydate date NOT NULL,
    Noofseats number NOT NULL,
    Totalfare number,
    Bookingstatus number(1),
    FOREIGN KEY(Userid) REFERENCES FRS_TBL_User_Credentials(Userid),
    FOREIGN KEY(Scheduleid) REFERENCES FRS_TBL_Schedule(Scheduleid));
```

```
8. create table FRS_TBL_Passenger(Reservationid varchar(8),
    Passengerid varchar(8) PRIMARY KEY,
    Name varchar(20) NOT NULL,
    Gender varchar(6) NOT NULL,
    Age varchar(6) NOT NULL,
    Seatno varchar(10) NOT NULL,
    FOREIGN KEY(Reservationid) REFERENCES FRS_TBL_Reservation(Reservationid));
```

SEQUENCE

```
create sequence FRS_SEQ_USER_ID
    start with 1000
    increment by 1;

 create sequence FRS_SEQ_ROUTE_ID
    start with 1000
    increment by 1;

create sequence FRS_SEQ_FLIGHT_ID
    start with 1000
    increment by 1;

create sequence FRS_SEQ_SCHEDULE_ID
   start with 1000
    increment by 1;

create sequence FRS_SEQ_RESERVATION_ID
    start with 1000
    increment by 1;
```

CODES:

BEAN CLASS

1.Credentials Bean

```
	@Entity
@Table(name="FRS_TBL_User_Credentials")

public class CredentialsBean {
	@Id
	@Column(name="Userid")
	private String userID;
	@Column(name="passWord")
	private String passWord;
	@Column(name="Usertype")
	private String userType;
	@Column(name="Loginstatus")
	private int loginStatus;

	@OneToOne(mappedBy="Userid")
	ProfileBeanUserid;

	@OneToMany(mappedBy="userID")
	List<ReservationBean> al=new ArrayList<ReservationBean>();

	public CredentialsBean() {
		super();
	}

	public CredentialsBean(String userID, String passWord, String userType, int loginStatus) {
		super();
		this.userID = userID;
		this.passWord = passWord;
		this.userType = userType;
		this.loginStatus = loginStatus;
	}

	public String getUserID() {
		return userID;
	}
	public void setUserID(String userID) {
		this.userID = userID;
	}

	public String getpassWord() {
		return passWord;
	}

	public void setpassWord(String passWord) {
		this.passWord = passWord;
	}

	public String getUserType() {
		return userType;
	}

	public void setUserType(String userType) {
		this.userType = userType;
	}

	public int getLoginStatus() {
		return loginStatus;
	}

	public void setLoginStatus(int loginStatus) {
		this.loginStatus = loginStatus;
	}

	@Override
	public String toString() {
		return "CredentialsBean [userID=" + userID + ", passWord=" + passWord + ",
userType=" + userType
					+ ", loginStatus=" + loginStatus + "]";
	}
```

2.Credit Card Bean

```
	@Entity
@Table(name="FRS_TBL_CreditCard")
public class CreditCardBean {

	@Id
 @Column(name="CreditCardNumber")
 private String creditCardNumber;
 @Column(name="ValidFrom")
 private String validFrom;
 @Column(name="ValidTo")
 private String validTo;
 @Column(name="CreditBalance")
 private int creditBalance;
public String getCreditCardNumber() {
        return creditCardNumber;
}

public void setCreditCardNumber(String creditCardNumber) {
        this.creditCardNumber = creditCardNumber;
}

public String getValidFrom() {
        return validFrom;
}

public void setValidFrom(String validFrom) {
        this.validFrom = validFrom;
}

public String getValidTo() {
        return validTo;
}

public void setValidTo(String validTo) {
        this.validTo = validTo;
}

public int getCreditBalance() {
        return creditBalance;
}

public void setCreditBalance(int creditBalance) {
        this.creditBalance = creditBalance;
}

public String getUserId() {
        return userId;
}
```

```
    public void setUserId(String userId) {
        this.userId = userId;
    }

    @Override
    public String toString() {
        return "CreditCardBean [creditCardNumber=" + creditCardNumber + ",
validFrom=" + validFrom + ", validTo="
                + validTo + ", creditBalance=" + creditBalance + ", userId=" + userId
+ "]";
    }

    public CreditCardBean(String creditCardNumber, String validFrom, String validTo, int
creditBalance, String userId) {
        super();
        this.creditCardNumber = creditCardNumber;
        this.validFrom = validFrom;
        this.validTo = validTo;
        this.creditBalance = creditBalance;
        this.userId = userId;
    }
```

3.Flight Bean

```
    @Entity
@Table(name="FRS_TBL_Flight")
public class FlightBean {

    @Id
    @Column(name="FlightId")
    private String flightID;
    @Column(name="FlightName")
    private String flightName;
    @Column(name="Seatingcapacity")
    private int seatingCapacity;
    @Column(name="Reservationcapacity")
    private int reservationCapacity;

    @OneToMany(mappedBy="flightID")
    List<ScheduleBean> al=new ArrayList<ScheduleBean>();


    public FlightBean() {
```

```
		super();
	}


	public FlightBean(String flightID, String flightName, int seatingCapacity, int
reservationCapacity) {
		super();
		this.flightID = flightID;
		this.flightName = flightName;
		this.seatingCapacity = seatingCapacity;
		this.reservationCapacity = reservationCapacity;
	}


	public String getFlightID() {
		return flightID;
	}


	public void setFlightID(String flightID) {
		this.flightID = flightID;
	}


	public String getFlightName() {
		return flightName;
	}


	public void setFlightName(String flightName) {
		this.flightName = flightName;
	}


	public int getSeatingCapacity() {
		return seatingCapacity;
	}


	public void setSeatingCapacity(int seatingCapacity) {
		this.seatingCapacity = seatingCapacity;
	}
```

```
    public int getReservationCapacity() {
        return reservationCapacity;
    }


    public void setReservationCapacity(int reservationCapacity) {
        this.reservationCapacity = reservationCapacity;
    }


    @Override
    public String toString() {
        return "FlightBean [flightID=" + flightID + ", flightName=" + flightName + ",
seatingCapacity="
                + seatingCapacity + ", reservationCapacity=" + reservationCapacity +
"]";
    }
```

4. Passenger Bean

```
    @Entity
@Table(name="FRS_TBL_Passenger")
public class PassengerBean {

    @Id
    @Column(name="Passengerid")
    private String passengerID;

//  @Column(name="Reservationid")
//  private String reservationID;
    @Column(name="Name")
    private String name;
    @Column(name="Gender")
    private String gender;
    @Column(name="Age")
    private String age;
    @Column(name="Seatno")
    private String seatNo;

    @ManyToOne
    @JoinColumn(name="Reservationid")
    ReservationBeanreservationID;
    public PassengerBean() {
        super();
    }

    public PassengerBean(String passengerID, String name, String gender, String age, String
seatNo,
            ReservationBeanreservationID) {
        super();
        this.passengerID = passengerID;
        this.name = name;
        this.gender = gender;
        this.age = age;
        this.seatNo = seatNo;
        this.reservationID = reservationID;
    }

    public String getPassengerID() {
        return passengerID;
    }

    public void setPassengerID(String passengerID) {
        this.passengerID = passengerID;
    }

    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }

    public String getGender() {
        return gender;
    }

    public void setGender(String gender) {
        this.gender = gender;
    }

    public String getAge() {
        return age;
    }
```

```
	public void setAge(String age) {
		this.age = age;
	}

	public String getSeatNo() {
		return seatNo;
	}

	public void setSeatNo(String seatNo) {
		this.seatNo = seatNo;
	}

	public ReservationBeangetReservationID() {
		return reservationID;
	}

	public void setReservationID(ReservationBeanreservationID) {
		this.reservationID = reservationID;
	}

	@Override
	public String toString() {
		return "PassengerBean [passengerID=" + passengerID + ", name=" + name + ",
gender=" + gender + ", age=" + age
				+ ", seatNo=" + seatNo + ", reservationID=" + reservationID + "]";
	}
```

5. Profile Bean

```
	@Entity
@Table(name=" FRS_TBL_User_Profile")
public class ProfileBean {

	@Id
	@Column(name="Userid")
	private String userID;
	@Column(name="Firstname")
	private String firstName;
	@Column(name="Lastname")
	private String lastName;
//	@Temporal(TemporalType.DATE)
	@Column(name="DateOfBirth")
	private Date dateOfBirth;
	@Column(name="Gender")
```

```
private String gender;
@Column(name="Street")
private String street;
@Column(name="Location")
private String location;
@Column(name="City")
private String city;
@Column(name="State")
private String state;
@Column(name="Pincode")
private String pincode;
@Column(name="MobileNo")
private String mobileNo;
@Column(name="EmailId")
private String emailID;

@Column(name="Password")
private String passWord;

@OneToOne
@JoinColumn(name="Userid")
CredentialsBeanUserid;

public ProfileBean() {
        super();
}

public String getUserID() {
        return userID;
}

public void setUserID(String userID) {
        this.userID = userID;
}

public String getFirstName() {
        return firstName;
}

public void setFirstName(String firstName) {
        this.firstName = firstName;
}

public String getLastName() {
        return lastName;
}

public void setLastName(String lastName) {
        this.lastName = lastName;
}

public Date getDateOfBirth() {
        return dateOfBirth;
}

public void setDateOfBirth(Date dateOfBirth) {
        this.dateOfBirth = dateOfBirth;
}

public String getGender() {
        return gender;
}

public void setGender(String gender) {
        this.gender = gender;
}

public String getStreet() {
        return street;
}

public void setStreet(String street) {
        this.street = street;
}

public String getLocation() {
        return location;
}

public void setLocation(String location) {
        this.location = location;
}

public String getCity() {
        return city;
}

public void setCity(String city) {
		this.city = city;
	}

	public String getState() {
		return state;
	}

	public void setState(String state) {
		this.state = state;
	}

	public String getPincode() {
		return pincode;
	}

	public void setPincode(String pincode) {
		this.pincode = pincode;
	}

	public String getMobileNo() {
		return mobileNo;
	}

	public void setMobileNo(String mobileNo) {
		this.mobileNo = mobileNo;
	}

	public String getEmailID() {
		return emailID;
	}

	public void setEmailID(String emailID) {
		this.emailID = emailID;
	}

	public String getPassWord() {
		return passWord;
	}

	public void setPassWord(String passWord) {
		this.passWord = passWord;
	}

	public CredentialsBeangetUserid() {
        return Userid;
    }

    public void setUserid(CredentialsBeanuserid) {
        Userid = userid;
    }

    @Override
    public String toString() {
        return "ProfileBean [userID=" + userID + ", firstName=" + firstName + ",
lastName=" + lastName
                + ", dateOfBirth=" + dateOfBirth + ", gender=" + gender + ", street="
+ street + ", location="
                + location + ", city=" + city + ", state=" + state + ", pincode=" +
pincode + ", mobileNo=" + mobileNo
                + ", emailID=" + emailID + ", passWord=" + passWord + ", Userid="
+ Userid + "]";
    }

    public ProfileBean(String userID, String firstName, String lastName, Date dateOfBirth,
String gender, String street,
            String location, String city, String state, String pincode, String mobileNo,
String emailID,
            String passWord, CredentialsBean userid2) {
        super();
        this.userID = userID;
        this.firstName = firstName;
        this.lastName = lastName;
        this.dateOfBirth = dateOfBirth;
        this.gender = gender;
        this.street = street;
        this.location = location;
        this.city = city;
        this.state = state;
        this.pincode = pincode;
        this.mobileNo = mobileNo;
        this.emailID = emailID;
        this.passWord = passWord;
        Userid = userid2;
    }
```

6.Reservation Bean

```
        @Entity
@Table(name="FRS_TBL_Reservation")
public class ReservationBean {
        @Id
        @Column(name="Reservationid")
        private String reservationID;

//      @Column(name="Userid")
//      private String userID;

//      @Column(name="Scheduleid")
//      private String scheduleID;

        @Column(name="Reservationtype")
        private String reservationType;

        @Column(name="Bookingdate")
        private Date bookingDate;

        @Column(name="Journeydate")
        private Date journeyDate;

        @Column(name="Noofseats")
        private int noOfSeats;

        @Column(name="Totalfare")
        private double totalFare;

        @Column(name="Bookingstatus")
        private int bookingStatus;

    @ManyToOne
    @JoinColumn(name="Scheduleid")
ScheduleBeanscheduleID;

    @ManyToOne
    @JoinColumn(name="Userid")
CredentialsBeanuserID;

        public ReservationBean() {
                super();
        }
}

	public ReservationBean(String reservationID, String reservationType, Date bookingDate, 
Date journeyDate,
				int noOfSeats, double totalFare, int bookingStatus, ScheduleBeanscheduleID, 
CredentialsBeanuserID) {
		super();
		this.reservationID = reservationID;
		this.reservationType = reservationType;
		this.bookingDate = bookingDate;
		this.journeyDate = journeyDate;
		this.noOfSeats = noOfSeats;
		this.totalFare = totalFare;
		this.bookingStatus = bookingStatus;
		this.scheduleID = scheduleID;
		this.userID = userID;
	}

	public String getReservationID() {
		return reservationID;
	}

	public void setReservationID(String reservationID) {
		this.reservationID = reservationID;
	}

	public String getReservationType() {
		return reservationType;
	}

	public void setReservationType(String reservationType) {
		this.reservationType = reservationType;
	}

	public Date getBookingDate() {
		return bookingDate;
	}

	public void setBookingDate(Date bookingDate) {
		this.bookingDate = bookingDate;
	}

	public Date getJourneyDate() {
		return journeyDate;
	}
```

```
}

public void setJourneyDate(Date journeyDate) {
        this.journeyDate = journeyDate;
}

public int getNoOfSeats() {
        return noOfSeats;
}

public void setNoOfSeats(int noOfSeats) {
        this.noOfSeats = noOfSeats;
}

public double getTotalFare() {
        return totalFare;
}

public void setTotalFare(double totalFare) {
        this.totalFare = totalFare;
}

public int getBookingStatus() {
        return bookingStatus;
}

public void setBookingStatus(int bookingStatus) {
        this.bookingStatus = bookingStatus;
}

public ScheduleBeangetScheduleID() {
        return scheduleID;
}

public void setScheduleID(ScheduleBeanscheduleID) {
        this.scheduleID = scheduleID;
}

public CredentialsBeangetUserID() {
        return userID;
}

public void setUserID(CredentialsBeanuserID) {
        this.userID = userID;
}
```

```
    @Override
    public String toString() {
        return "ReservationBean [reservationID=" + reservationID + ", reservationType=" +
reservationType
                + ", bookingDate=" + bookingDate + ", journeyDate=" + journeyDate
+ ", noOfSeats=" + noOfSeats
                + ", totalFare=" + totalFare + ", bookingStatus=" + bookingStatus + ",
scheduleID=" + scheduleID
                + ", userID=" + userID + "]";
    }
```

7.Route Bean

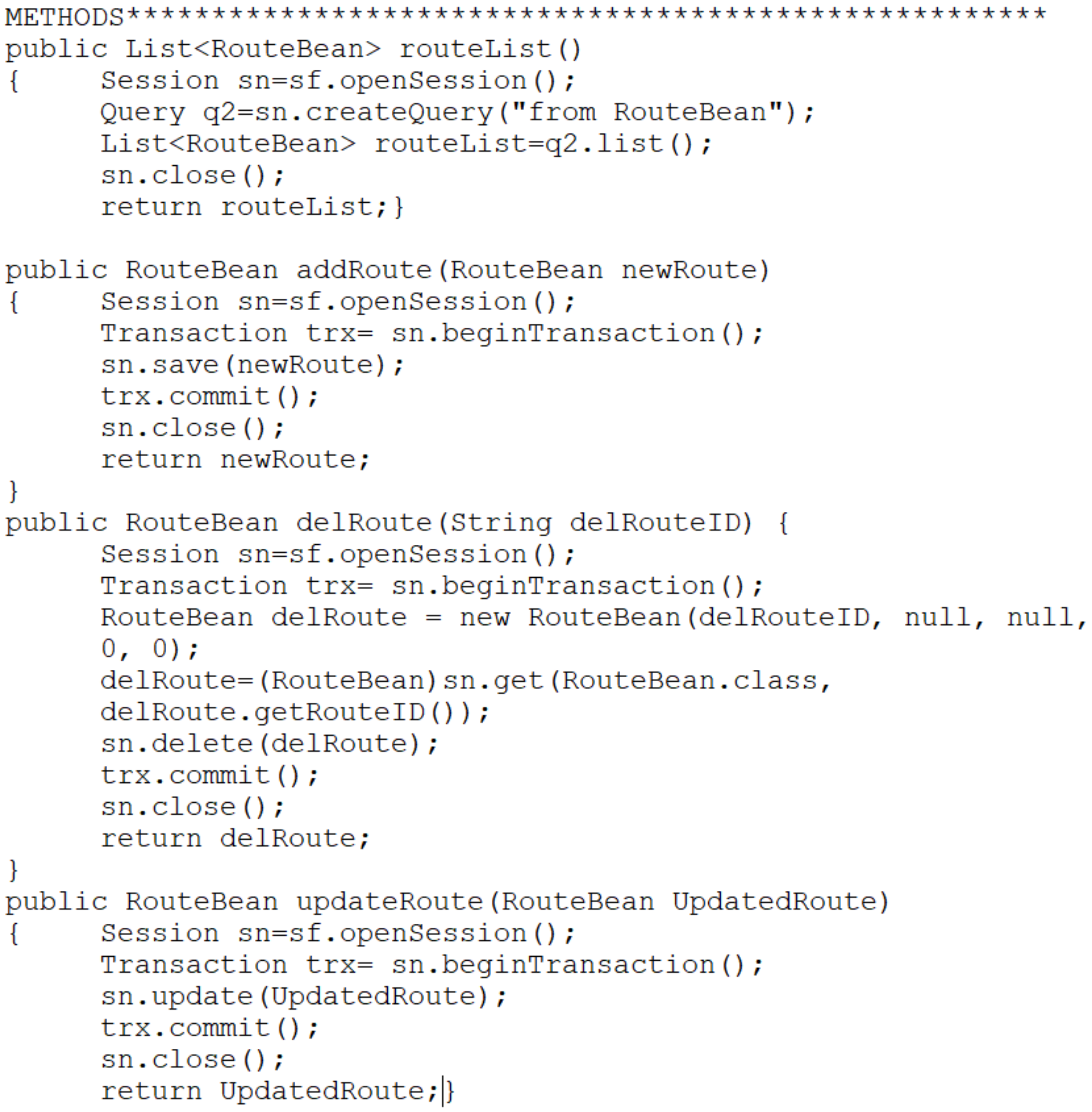
```
    @Entity
@Table(name="FRS_TBL_Route")
public class RouteBean {

    @Id
    @Column(name="Routeid")
    private String routeID;
    @Column(name="Source")
    private String source;
    @Column(name="Destination")
    private String destination;
    @Column(name="Distance")
    private int distance;
    @Column(name="Fare")
    private double fare;

    @OneToMany(mappedBy="routeID")
    List<ScheduleBean> al=new ArrayList<ScheduleBean>();

    public RouteBean() {
        super();
    }

    public RouteBean(String routeID, String source, String destination, int distance, double
fare) {
        super();
        this.routeID = routeID;
        this.source = source;
```

```
		this.destination = destination;
		this.distance = distance;
		this.fare = fare;
	}

	@Override
	public String toString() {
		return "RouteBean [routeID=" + routeID + ", source=" + source + ", destination=" +
destination + ", distance="
				+ distance + ", fare=" + fare + "]";
	}

	public String getRouteID() {
		return routeID;
	}

	public void setRouteID(String routeID) {
		this.routeID = routeID;
	}

	public String getSource() {
		return source;
	}

	public void setSource(String source) {
		this.source = source;
	}

	public String getDestination() {
		return destination;
	}

	public void setDestination(String destination) {
		this.destination = destination;
	}

	public int getDistance() {
		return distance;
	}

	public void setDistance(int distance) {
		this.distance = distance;
	}
	public double getFare() {
		return fare;
	}

	public void setFare(double fare) {
		this.fare = fare;
	}
```

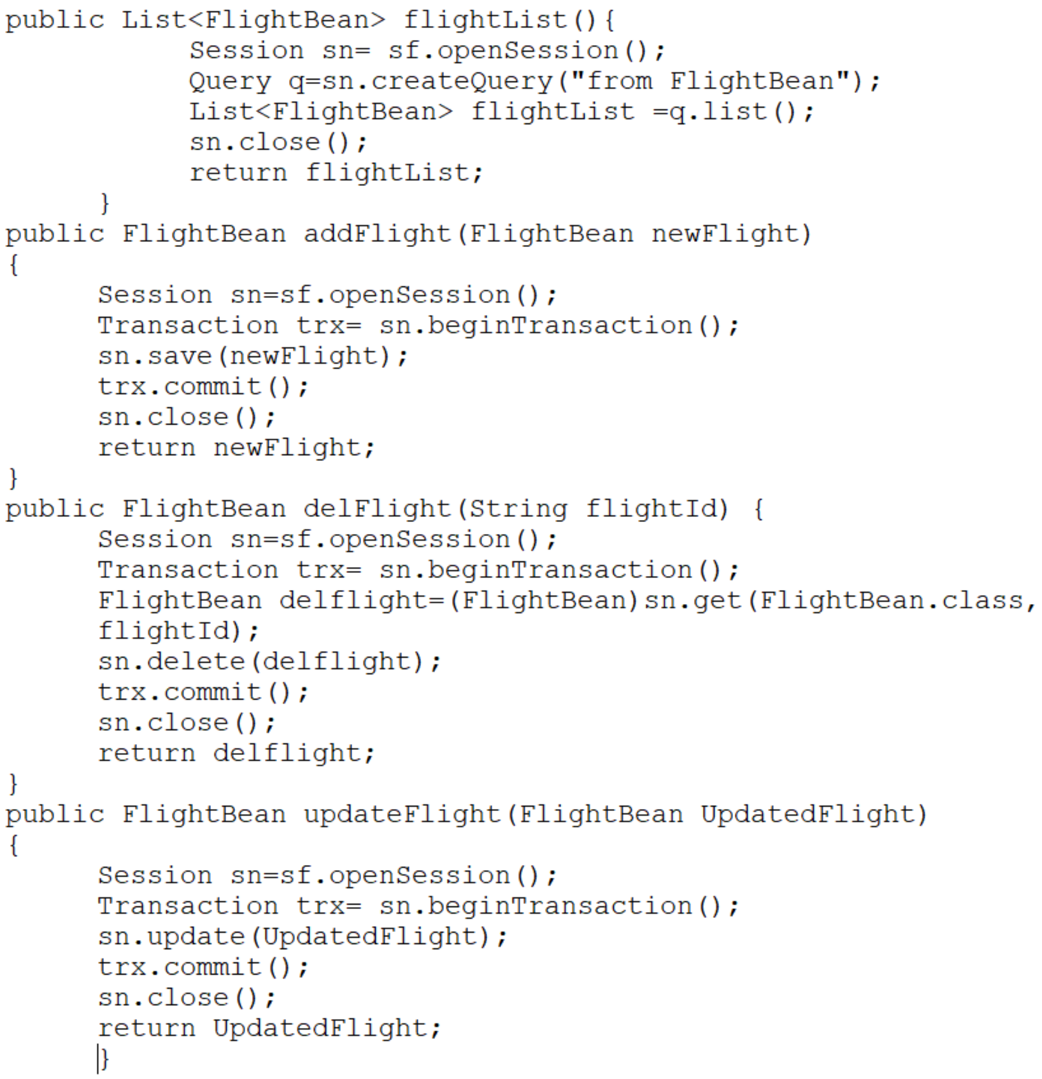
8.Schedule Bean

```
	@Entity
@Table(name="FRS_TBL_Schedule")
public class ScheduleBean {

	@Id
	@Column(name="Scheduleid")
	private String scheduleID;
//	@Column(name="Flightid")
//	private String flightID;
//	@Column(name="Routeid")
//	private String routeID;
	@Column(name="TravelDuration")
	private int travelDuration;
	@Column(name="AvailableDays")
	private String availableDays;
	@Column(name="DepartureTime")
	private String departureTime;

	@ManyToOne
	@JoinColumn(name="Routeid")
	RouteBeanrouteID;

	@ManyToOne
	@JoinColumn(name="Flightid")
	FlightBeanflightID;

	@OneToMany(mappedBy="scheduleID")
	List<ReservationBean> al=new ArrayList<ReservationBean>();

	public ScheduleBean() {
		super();
	}

	public ScheduleBean(String scheduleID, int travelDuration, String availableDays, String
departureTime,
			RouteBeanrouteID, FlightBeanflightID) {
		super();
		this.scheduleID = scheduleID;
		this.travelDuration = travelDuration;
		this.availableDays = availableDays;
		this.departureTime = departureTime;
		this.routeID = routeID;
		this.flightID = flightID;
	}

	@Override
	public String toString() {
		return "ScheduleBean [scheduleID=" + scheduleID + ", travelDuration=" +
travelDuration + ", availableDays="
				+ availableDays + ", departureTime=" + departureTime + ", routeID="
+ routeID.getRouteID() + ", flightID=" + flightID.getFlightID()
				+ "]";
	}

	public String getScheduleID() {
		return scheduleID;
	}

	public void setScheduleID(String scheduleID) {
		this.scheduleID = scheduleID;
	}

	public int getTravelDuration() {
		return travelDuration;
	}

	public void setTravelDuration(int travelDuration) {
		this.travelDuration = travelDuration;
	}

	public String getAvailableDays() {
		return availableDays;
	}

	public void setAvailableDays(String availableDays) {
		this.availableDays = availableDays;
	}

	public String getDepartureTime() {
		return departureTime;
	}

	public void setDepartureTime(String departureTime) {
		this.departureTime = departureTime;
	}

	public RouteBeangetRouteID() {
		return routeID;
	}

	public void setRouteID(RouteBeanrouteID) {
		this.routeID = routeID;
	}

	public FlightBeangetFlightID() {
		return flightID;
	}

	public void setFlightID(FlightBeanflightID) {
		this.flightID = flightID;
	}
```

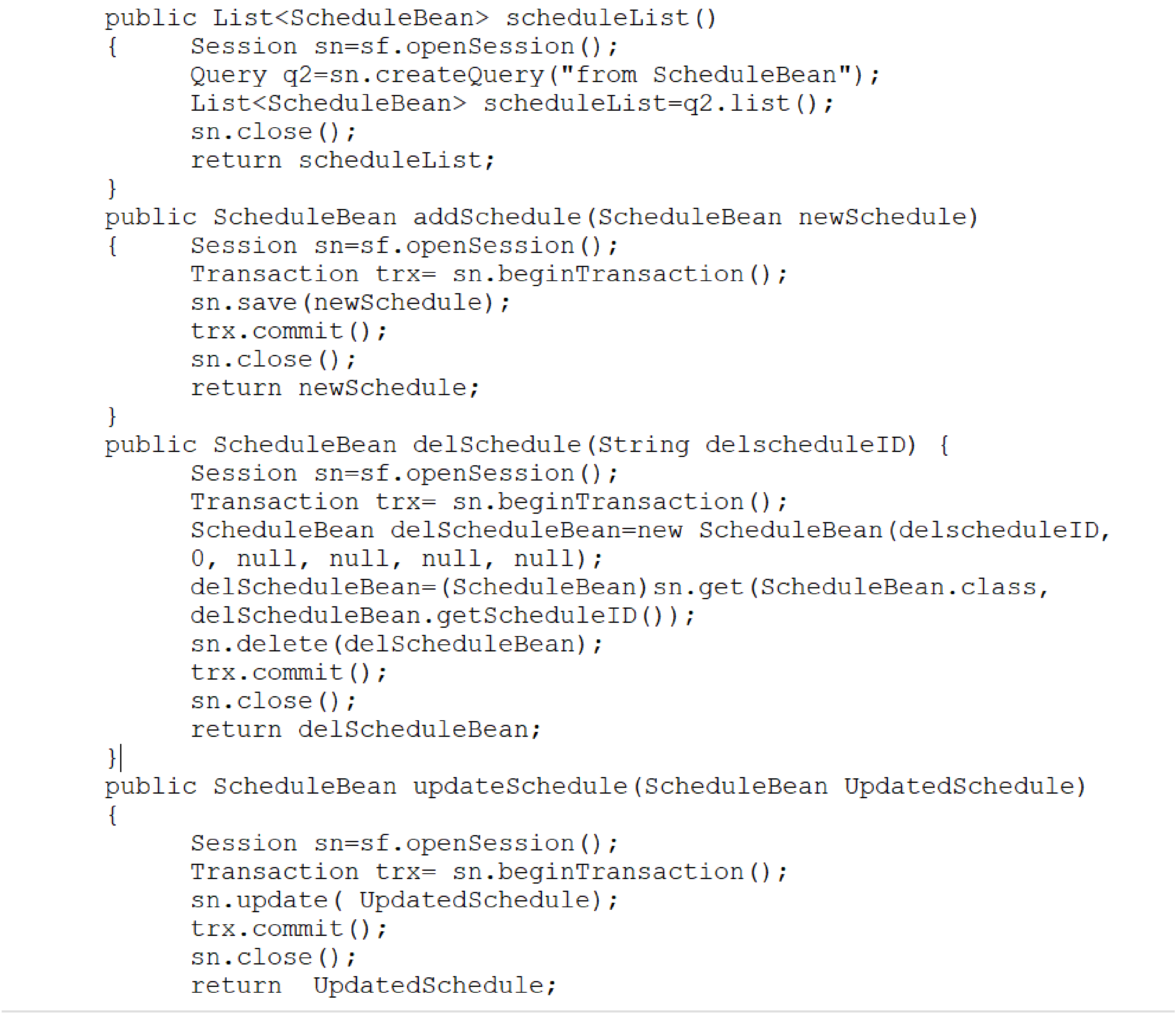
DAO CLASS



```
METHODS************************************************************
public List<RouteBean> routeList()
{       Session sn=sf.openSession();
        Query q2=sn.createQuery("from RouteBean");
        List<RouteBean> routeList=q2.list();
        sn.close();
        return routeList;}

public RouteBean addRoute(RouteBean newRoute)
{       Session sn=sf.openSession();
        Transaction trx= sn.beginTransaction();
        sn.save(newRoute);
        trx.commit();
        sn.close();
        return newRoute;
}
public RouteBean delRoute(String delRouteID) {
        Session sn=sf.openSession();
        Transaction trx= sn.beginTransaction();
        RouteBean delRoute = new RouteBean(delRouteID, null, null,
        0, 0);
        delRoute=(RouteBean)sn.get(RouteBean.class,
        delRoute.getRouteID());
        sn.delete(delRoute);
        trx.commit();
        sn.close();
        return delRoute;
}
public RouteBean updateRoute(RouteBean UpdatedRoute)
{       Session sn=sf.openSession();
        Transaction trx= sn.beginTransaction();
        sn.update(UpdatedRoute);
        trx.commit();
        sn.close();
        return UpdatedRoute;}
```

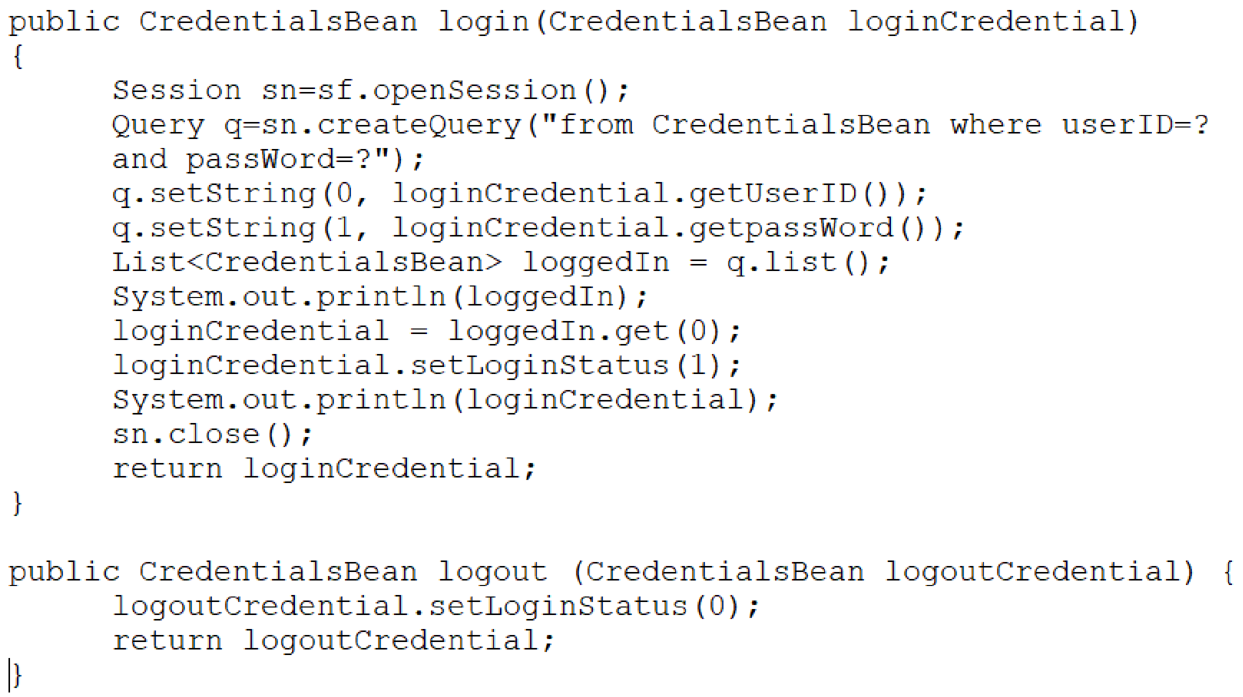


```
public List<FlightBean> flightList(){
           Session sn= sf.openSession();
           Query q=sn.createQuery("from FlightBean");
           List<FlightBean> flightList =q.list();
           sn.close();
           return flightList;
      }
public FlightBean addFlight(FlightBean newFlight)
{
      Session sn=sf.openSession();
      Transaction trx= sn.beginTransaction();
      sn.save(newFlight);
      trx.commit();
      sn.close();
      return newFlight;
}
public FlightBean delFlight(String flightId) {
      Session sn=sf.openSession();
      Transaction trx= sn.beginTransaction();
      FlightBean delflight=(FlightBean)sn.get(FlightBean.class,
      flightId);
      sn.delete(delflight);
      trx.commit();
      sn.close();
      return delflight;
}
public FlightBean updateFlight(FlightBean UpdatedFlight)
{
      Session sn=sf.openSession();
      Transaction trx= sn.beginTransaction();
      sn.update(UpdatedFlight);
      trx.commit();
      sn.close();
      return UpdatedFlight;
      }
```

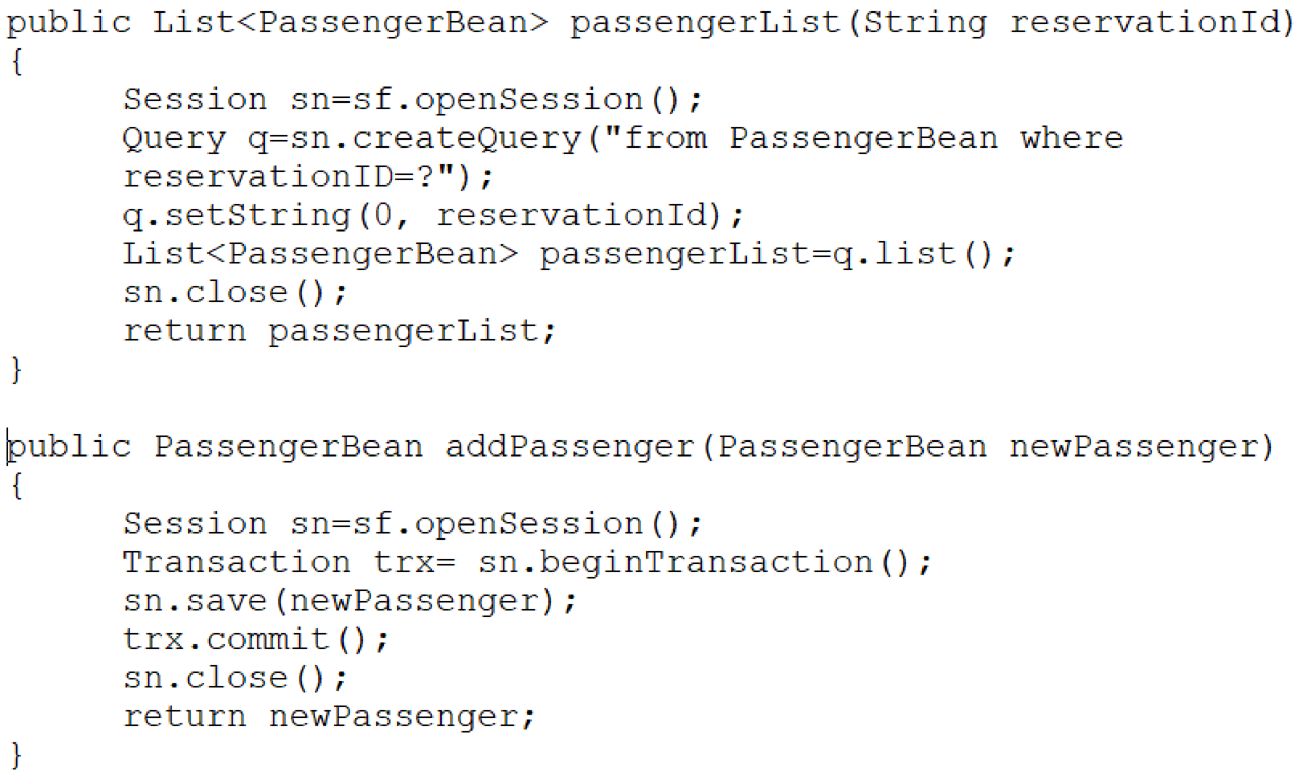


```
public List<ScheduleBean> scheduleList()
{       Session sn=sf.openSession();
        Query q2=sn.createQuery("from ScheduleBean");
        List<ScheduleBean> scheduleList=q2.list();
        sn.close();
        return scheduleList;
}
public ScheduleBean addSchedule(ScheduleBean newSchedule)
{       Session sn=sf.openSession();
        Transaction trx= sn.beginTransaction();
        sn.save(newSchedule);
        trx.commit();
        sn.close();
        return newSchedule;
}
public ScheduleBean delSchedule(String delscheduleID) {
        Session sn=sf.openSession();
        Transaction trx= sn.beginTransaction();
        ScheduleBean delScheduleBean=new ScheduleBean(delscheduleID,
        0, null, null, null, null);
        delScheduleBean=(ScheduleBean)sn.get(ScheduleBean.class,
        delScheduleBean.getScheduleID());
        sn.delete(delScheduleBean);
        trx.commit();
        sn.close();
        return delScheduleBean;
}
public ScheduleBean updateSchedule(ScheduleBean UpdatedSchedule)
{
        Session sn=sf.openSession();
        Transaction trx= sn.beginTransaction();
        sn.update( UpdatedSchedule);
        trx.commit();
        sn.close();
        return  UpdatedSchedule;
```

```
public List<CredentialsBean> credentialList()
{     Session sn=sf.openSession();
      Query q2=sn.createQuery("from CredentialsBean");
      List<CredentialsBean> credList=q2.list();
      sn.close();
      return credList;
}
public CredentialsBean addCredential( CredentialsBean
newCredentials)
{     Session sn=sf.openSession();
      Transaction trx= sn.beginTransaction();
      sn.save(newCredentials);
      trx.commit();
      sn.close();
      return newCredentials;
}
public CredentialsBean delCredential(CredentialsBean
delCredential) {
      Session sn=sf.openSession();
      Transaction trx= sn.beginTransaction();
      delCredential=(CredentialsBean)sn.get(CredentialsBean.class,
      delCredential.getUserID());
      sn.delete(delCredential);
      trx.commit();
      sn.close();
      return delCredential;}

public CredentialsBean updateCredential(CredentialsBean
UpdateCredential)
{     Session sn=sf.openSession();
      Transaction trx= sn.beginTransaction();
      sn.update(UpdateCredential);
      trx.commit();
      sn.close();
      return UpdateCredential;
}
```



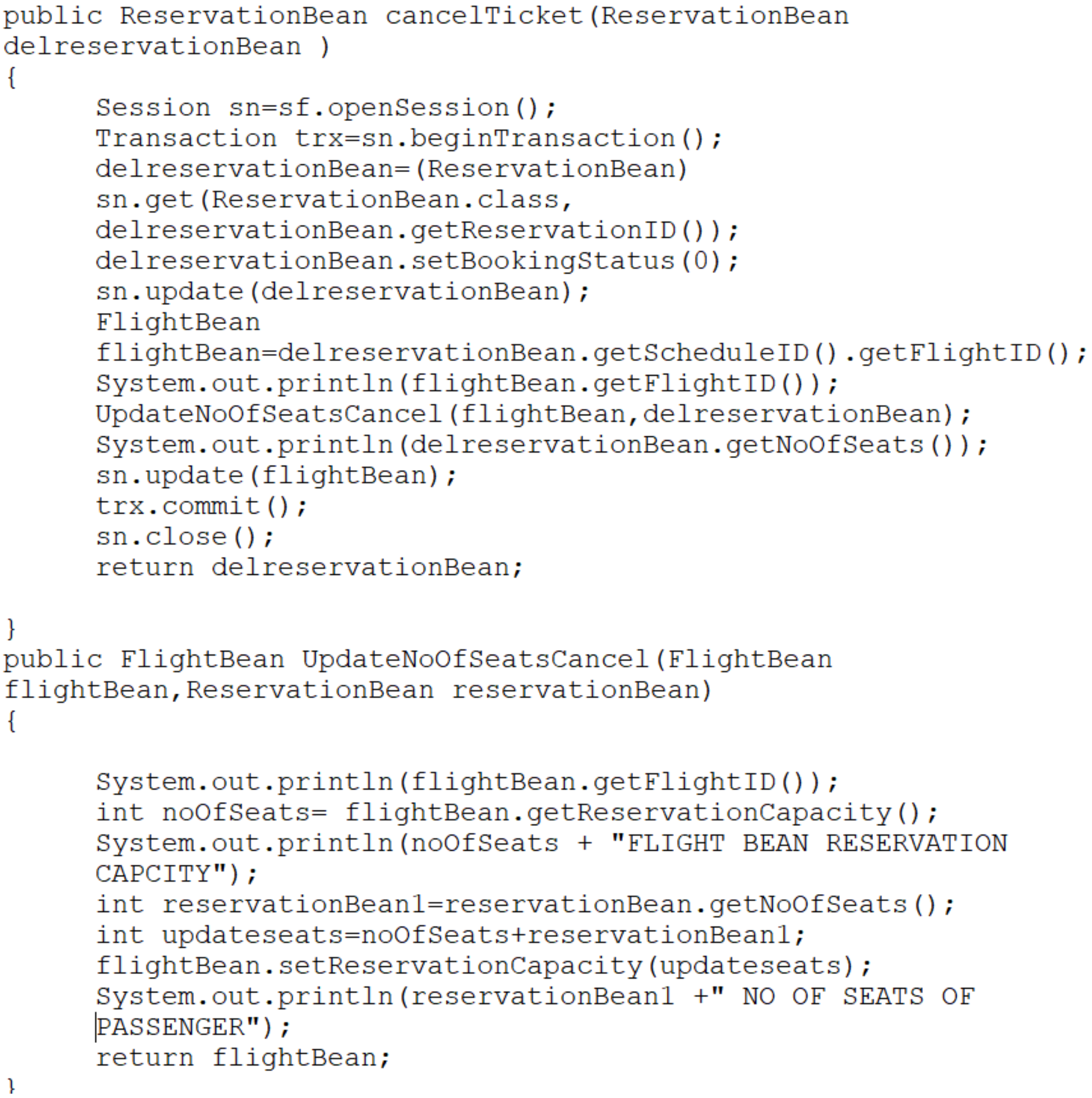
```
public CredentialsBean login(CredentialsBean loginCredential)
{
        Session sn=sf.openSession();
        Query q=sn.createQuery("from CredentialsBean where userID=?
        and passWord=?");
        q.setString(0, loginCredential.getUserID());
        q.setString(1, loginCredential.getpassWord());
        List<CredentialsBean> loggedIn = q.list();
        System.out.println(loggedIn);
        loginCredential = loggedIn.get(0);
        loginCredential.setLoginStatus(1);
        System.out.println(loginCredential);
        sn.close();
        return loginCredential;
}

public CredentialsBean logout (CredentialsBean logoutCredential) {
        logoutCredential.setLoginStatus(0);
        return logoutCredential;
}
```



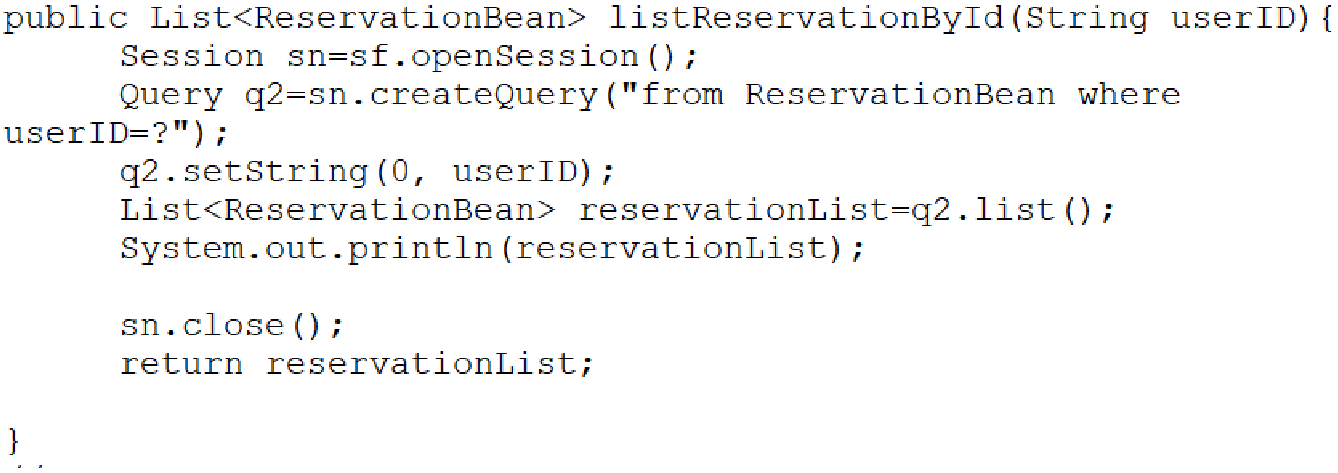
```
public List<PassengerBean> passengerList(String reservationId)
{
        Session sn=sf.openSession();
        Query q=sn.createQuery("from PassengerBean where
        reservationID=?");
        q.setString(0, reservationId);
        List<PassengerBean> passengerList=q.list();
        sn.close();
        return passengerList;
}

public PassengerBean addPassenger(PassengerBean newPassenger)
{
        Session sn=sf.openSession();
        Transaction trx= sn.beginTransaction();
        sn.save(newPassenger);
        trx.commit();
        sn.close();
        return newPassenger;
}
```

```
public List<CreditCardBean> creditList()
{
       Session sn=sf.openSession();
       Query q2=sn.createQuery("from CreditCardBean");
       List<CreditCardBean> creditList=q2.list();
       sn.close();
       return creditList;
}
public CreditCardBean addCreditCard(CreditCardBean creditBean)
{
       Session sn=sf.openSession();
       Transaction trx= sn.beginTransaction();
       sn.save(creditBean);
       trx.commit();
       sn.close();
       return creditBean;
}
public CreditCardBean delCreditcard(CreditCardBean creditCard) {
       Session sn=sf.openSession();
       Transaction trx= sn.beginTransaction();
       creditCard =(CreditCardBean) sn.get(CreditCardBean.class,
       creditCard.getCreditCardNumber());
       sn.delete(creditCard);
       trx.commit();
       sn.close();
       return creditCard;
}
public CreditCardBean updateCreditCard(CreditCardBean
creditcardnumber)
{      Session sn=sf.openSession();
       Transaction trx= sn.beginTransaction();
       sn.update(creditcardnumber);
       trx.commit();
       sn.close();
       return creditcardnumber;
}
```

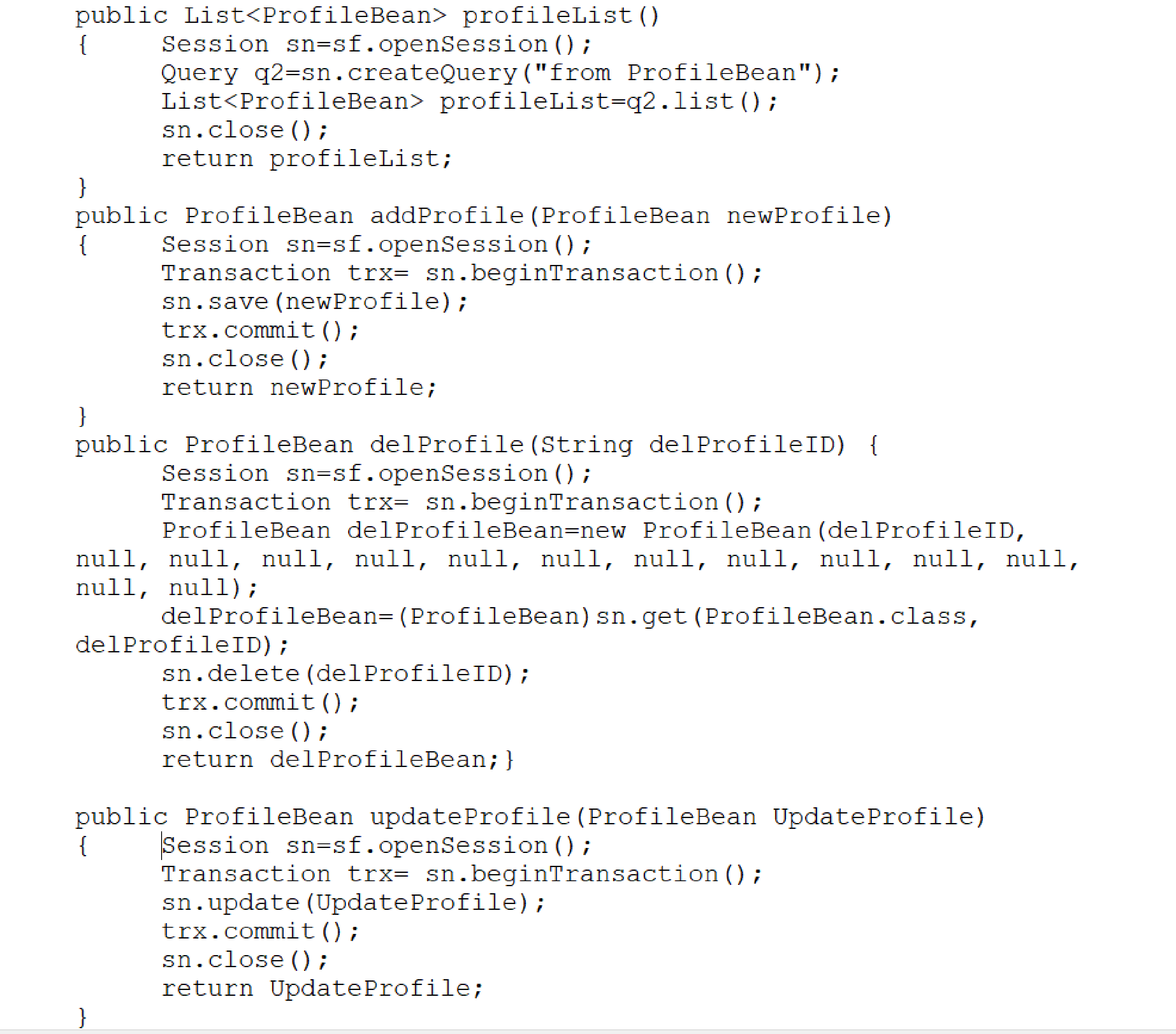
```
public CredentialsBean changePassword(String userId,String
oldPassword,String newPassword)
{
        Session sn=sf.openSession();
        Transaction trx= sn.beginTransaction();
        CredentialsBean credentialsBean=(CredentialsBean)
        sn.get(CredentialsBean.class,userId);
        System.out.println(credentialsBean.getpassWord()+"
        "+newPassword);
        if(credentialsBean.getpassWord().trim().equalsIgnoreCase(old
        Password.trim()))
        {
                System.out.println("password match");
                credentialsBean.setpassWord(newPassword);
                sn.update(credentialsBean);
                trx.commit();
                sn.close();
                return credentialsBean;
        }
        return null;
}


public ReservationBean bookTicket(ReservationBean reservationBean)
{
        Session sn=sf.openSession();
        Transaction trx=sn.beginTransaction();
        System.out.println(reservationBean.getScheduleID());
        sn.save(reservationBean);
        FlightBean
        flightBean=reservationBean.getScheduleID().getFlightID();
        System.out.println(flightBean.getFlightID());
        UpdateNoOfSeats(flightBean,reservationBean);
        System.out.println(reservationBean.getNoOfSeats());
        sn.update(flightBean);
        trx.commit();
        sn.close();
        return reservationBean;

}

public FlightBean UpdateNoOfSeats(FlightBean
flightBean,ReservationBean reservationBean)
{

        System.out.println(flightBean.getFlightID());
        int noOfSeats= flightBean.getReservationCapacity();
        System.out.println(noOfSeats + "FLIGHT BEAN RESERVATION
        CAPCITY");
        int reservationBean1=reservationBean.getNoOfSeats();
        int updateseats=noOfSeats-reservationBean1;
        flightBean.setReservationCapacity(updateseats);
        System.out.println(reservationBean1 +" NO OF SEATS OF
        PASSENGER");
        return flightBean;
}
```



```
public ReservationBean cancelTicket(ReservationBean
delreservationBean )
{
        Session sn=sf.openSession();
        Transaction trx=sn.beginTransaction();
        delreservationBean=(ReservationBean)
        sn.get(ReservationBean.class,
        delreservationBean.getReservationID());
        delreservationBean.setBookingStatus(0);
        sn.update(delreservationBean);
        FlightBean
        flightBean=delreservationBean.getScheduleID().getFlightID();
        System.out.println(flightBean.getFlightID());
        UpdateNoOfSeatsCancel(flightBean,delreservationBean);
        System.out.println(delreservationBean.getNoOfSeats());
        sn.update(flightBean);
        trx.commit();
        sn.close();
        return delreservationBean;

}
public FlightBean UpdateNoOfSeatsCancel(FlightBean
flightBean,ReservationBean reservationBean)
{

        System.out.println(flightBean.getFlightID());
        int noOfSeats= flightBean.getReservationCapacity();
        System.out.println(noOfSeats + "FLIGHT BEAN RESERVATION
        CAPCITY");
        int reservationBean1=reservationBean.getNoOfSeats();
        int updateseats=noOfSeats+reservationBean1;
        flightBean.setReservationCapacity(updateseats);
        System.out.println(reservationBean1 +" NO OF SEATS OF
        PASSENGER");
        return flightBean;
}
```



```
public List<ReservationBean> listReservationById(String userID){
      Session sn=sf.openSession();
      Query q2=sn.createQuery("from ReservationBean where
userID=?");
      q2.setString(0, userID);
      List<ReservationBean> reservationList=q2.list();
      System.out.println(reservationList);

      sn.close();
      return reservationList;

}
```



```
public List<ProfileBean> profileList()
{       Session sn=sf.openSession();
        Query q2=sn.createQuery("from ProfileBean");
        List<ProfileBean> profileList=q2.list();
        sn.close();
        return profileList;
}
public ProfileBean addProfile(ProfileBean newProfile)
{       Session sn=sf.openSession();
        Transaction trx= sn.beginTransaction();
        sn.save(newProfile);
        trx.commit();
        sn.close();
        return newProfile;
}
public ProfileBean delProfile(String delProfileID) {
        Session sn=sf.openSession();
        Transaction trx= sn.beginTransaction();
        ProfileBean delProfileBean=new ProfileBean(delProfileID,
null, null, null, null, null, null, null, null, null, null, null,
null, null);
        delProfileBean=(ProfileBean)sn.get(ProfileBean.class,
delProfileID);
        sn.delete(delProfileID);
        trx.commit();
        sn.close();
        return delProfileBean;}

public ProfileBean updateProfile(ProfileBean UpdateProfile)
{       Session sn=sf.openSession();
        Transaction trx= sn.beginTransaction();
        sn.update(UpdateProfile);
        trx.commit();
        sn.close();
        return UpdateProfile;
}
```

ADMIN SERVICE CLASS

```
@Service
public class AdminService {

      @Autowired
      Dao AdminDao;
//    *********LOGIN***********
      public CredentialsBean login(CredentialsBean 
loginCredential)
      {
            CredentialsBean 
credentialsBean=AdminDao.login(loginCredential);
            return credentialsBean;
      }



//    ***********ADD Flight by Admin

      public FlightBean addFlight(FlightBean newFlight) {
            FlightBean flightbean = AdminDao.addFlight(newFlight);
            return flightbean;
      }

//    ***********Delete Flight by Admin

      public FlightBean delFlight(String flightId ) {
            FlightBean flightbean=AdminDao.delFlight(flightId);
            return flightbean;
}

      //************View by ADMIN
      public List<FlightBean> flightList(){
            List<FlightBean> flightbean=AdminDao.flightList();
            return flightbean;

      //*****************Modify by ADMIN
      public FlightBean updateFlight(FlightBean UpdatedFlight)
      {
            FlightBean 
flightBean=AdminDao.updateFlight(UpdatedFlight);
            return flightBean;

      }
```

```
//***********SCHEDULE OPERATION

      //*********************ADD SCHEDULE Details by ADMIN

      public ScheduleBean addschedule(ScheduleBean newschedule)
      {
           ScheduleBean scheduleBean
=AdminDao.addSchedule(newschedule);
           return newschedule;
      }



      //***************Delete SCHEDULE Details by ADMIN
      public ScheduleBean deleteSchedule(String delScheduleID)
      {
           ScheduleBean
scheduleBean=AdminDao.delSchedule(delScheduleID);
           return scheduleBean;
      }

      //*************View SCHEDULE Details by ADMIN

      public List<ScheduleBean> scheduleList()
      {
           List <ScheduleBean>
scheduleBean=AdminDao.scheduleList();
           return (List<ScheduleBean>) scheduleBean;
      }
```

```
        //****************Modify SCHEDULE Details by ADMIN

        public ScheduleBean updateSchedule(ScheduleBean
UpdatedSchedule)
        {
                ScheduleBean
scheduleBean=AdminDao.updateSchedule(UpdatedSchedule);
                return scheduleBean;

        }




//      *********************ROUTE DETAILS

//      *********************ADD ROUTES


        public RouteBean addRoute(RouteBean newRoute)
        {
                RouteBean routeBean=AdminDao.addRoute(newRoute);
                return routeBean;
        }

//      *********************DELETE ROUTES
        public RouteBean delRoute(String delRouteID)
        {
                RouteBean routeBean=AdminDao.delRoute(delRouteID);
                return routeBean;
        }
```

```
//	**********************View ROUTES

	public List<RouteBean> routeList()
	{
		List<RouteBean> routeBeans=AdminDao.routeList();
		return routeBeans;
	}



//	//	****************View ROUTES

	public RouteBean updateRoute(RouteBean UpdatedRoute)
	{
		RouteBean
routeBean=AdminDao.updateRoute(UpdatedRoute);
		return routeBean;
	}



//	*********************ADD PASSENGER DETAILS

	public PassengerBean addPassenger(PassengerBean
newPassenger)
	{
		PassengerBean
passengerBean=AdminDao.addPassenger(newPassenger);
		return passengerBean;
	}
```

```
//	******************VIEW/List PASSENGER/UserProfile DETAILS

	public List<PassengerBean> passengerList(String 
reservationId)
	{
		List<PassengerBean> 
passengerList=AdminDao.passengerList(reservationId);
		return passengerList;
	}



//	******************CHANGE PASSWORD



	public CredentialsBean changePassword(String userId,String 
oldPassword,String newPassword) {
		CredentialsBean 
credentialBean=AdminDao.changePassword(userId, oldPassword, 
newPassword);
		return credentialBean;
	}
//	*******************LOGOUT
	public CredentialsBean logout (CredentialsBean 
logoutCredential)
	{
		CredentialsBean 
credentialsBean=AdminDao.logout(logoutCredential);
		return credentialsBean;


	}

//--------------------------------------------
	public List<ProfileBean> profileList()
	{
		List<ProfileBean> profileList=AdminDao.profileList();
		return profileList ;

	}
```

CUSTOMER SERVICE CLASS

```
@Service
public class CustomerService {

      @Autowired
      Dao customerDao;

      public ArrayList<ScheduleBean> viewScheduleByRoute(String
source, String destination, Date date)
      {
            List<ScheduleBean>scheduleList=
customerDao.scheduleList();
            return (ArrayList<ScheduleBean>) scheduleList ;

      }

      //--------------------------------------
ListReservationById-----------------------------------------------
--------

      public List<ReservationBean> listResrvationById(String
userID){

            List<ReservationBean>
listreservationById=customerDao.listReservationById(userID);
            return listreservationById;
      }
      //------------------------------------------------------
LOGIN ----------------------------------------------------------

      public CredentialsBean login(CredentialsBean
loginCredential)
      {
            CredentialsBean
credentialsBean=customerDao.login(loginCredential);
            return loginCredential;
      }

      //------------------------------------------------------
CHANGE
PASSWORD----------------------------------------------------------
//    public CredentialsBean updateCredential(CredentialsBean
UpdateCredential)
//    {
//          CredentialsBean
credentialbean=customerDao.updateCredential(UpdateCredential);
//          return credentialbean;
//    }

      public CredentialsBean changePassword(String userId,String
oldPassword,String newPassword)
      {
            CredentialsBean
credentialBean=customerDao.changePassword(userId, oldPassword,
newPassword);
            return  credentialBean;
      }



      //
      *************************LOGOUT*****************************
*****************************************************************
      public CredentialsBean logout (CredentialsBean
logoutCredential)
      {
            CredentialsBean
credentialsBean=customerDao.logout(logoutCredential);
            return credentialsBean;
      //---------------------------------CREDITCARD
METHODS-------------------------------

      public List<CreditCardBean> viewCreditList()
      {
            List<CreditCardBean> creditList=
customerDao.creditList();
            return creditList;

      }

      //------------------------------------ADD
PASSENGER------------------------------------------------------
--------

           public PassengerBean addpassenger(PassengerBean
passengerbean) {

                PassengerBean
addPassenger=customerDao.addPassenger(passengerbean);
           return addPassenger;
           }

           //------------------------------------PROFILE
SERVICE-------------------------------------------------

           public ProfileBean addProfile(ProfileBean newProfile) {

                ProfileBean
profileBean=customerDao.addProfile(newProfile);
                return profileBean;
           }

           public ProfileBean delProfile(String delUserID)
           {
                ProfileBean
profileBean=customerDao.delProfile(delUserID);
                return profileBean;
           }
            public ProfileBean updateProfile(ProfileBean
  profileBean)
            {
                 ProfileBean
  profilebean=customerDao.updateProfile(profileBean);
                 return profilebean;
            }
```

CONTROLLER CLASS

```
@RestController
@CrossOrigin("*")
public class FrsController {


        @Autowired
        CustomerService customerService;

        @Autowired
        AdminService adminService;

        @Autowired
        Dao dao;

        @GetMapping("/Customer/ScheduleList")
        public List<ScheduleBean> getSchedule(String source, String
destination, Date date)
        {
                List<ScheduleBean> li =
customerService.viewScheduleByRoute(source, destination, date);
                                return li;

        }

        //--------------------------------------
ListReservationById-----------------------------------------------
--------

        @GetMapping("/Customer/ListReservationById/{id}")
        public List<ReservationBean>
listReservationById(@PathVariable("id") String userId){
                System.out.println(userId);
                List<ReservationBean>
listReservationById=customerService.listResrvationById(userId);
                return listReservationById;
        }

        @GetMapping("/Customer/CreditCardList")
        public List<CreditCardBean> getCreditCardList()
        {
                List<CreditCardBean> creditcardList =
customerService.viewCreditList();
                                return creditcardList;

        }
        @PostMapping("/Customer/AddCreditCard")
        public CreditCardBean addCreditCardList(@RequestBody
  CreditCardBean creditcardbean)
        {
                System.out.println(creditcardbean);
                CreditCardBean creditcard =
  dao.addCreditCard(creditcardbean);
                return creditcard;

        }
```

ADMIN CONTROLLER

```
//      --------------------------ADMIN
CONTROLLER---------------------------


        @GetMapping("/Admin/ListProfile")
        public List<ProfileBean> listProfile(){
                List<ProfileBean>
profileList=adminService.profileList();
                return profileList;
        }

        @GetMapping("/Admin/PassengerList/{id}")
        public List<PassengerBean>
getPassengerList(@PathVariable("id") String reservationId)
        {
                List<PassengerBean>
passengerList=adminService.passengerList(reservationId);
                return passengerList;

        }

@PostMapping("/Admin/AddFlight")
public FlightBean addFlight(@RequestBody FlightBean addflight) {
        System.out.println(addflight);
        FlightBean flightbean=adminService.addFlight(addflight);
        return flightbean;

}

@DeleteMapping("/Admin/DelFlight/{Id}")
public FlightBean delFlight(@PathVariable("Id") String flightid) {

        FlightBean flightbean=adminService.delFlight(flightid);
        return flightbean;
}
```

```
@GetMapping("/Admin/ListFlight")
public List<FlightBean> listFlight(){
      List<FlightBean> flightbean=adminService.flightList();
      return flightbean;
}

@PutMapping("/Admin/UpdateFlight")
public FlightBean updateFlight(@RequestBody FlightBean
updateFlight) {
      System.out.println(updateFlight);
      FlightBean
flightbean=adminService.updateFlight(updateFlight);
      return flightbean;
}

//--------------------------------ADMIN
ROUTE----------------------------

@PostMapping("/Admin/AddRoute")
public RouteBean addRoute(@RequestBody RouteBean addroute) {
      RouteBean routebean=adminService.addRoute(addroute);
      return routebean;

}


@DeleteMapping("/Admin/DelRoute/{Id}")
public RouteBean delRoute(@PathVariable("Id") String delRouteID) {
      RouteBean routebean=adminService.delRoute(delRouteID);
      return routebean;
}

  @GetMapping("/Admin/ListRoute")
  public List<RouteBean> listRoute(){
        List<RouteBean> routebean=adminService.routeList();
        return routebean;
  }
 @PutMapping("/Admin/UpdateRoute")
 public RouteBean updateRoute(@RequestBody RouteBean updateRoute) {
       RouteBean routebean=adminService.updateRoute(updateRoute);
       return routebean;
 }
```

```
//---------------------------------ADMIN
SCHEDULE-----------------------------

@PostMapping("/Admin/AddSchedule")
public ScheduleBean addSchedule(@RequestBody ScheduleBean
addschedule) {
      System.out.println(addschedule);
      ScheduleBean
schedulebean=adminService.addschedule(addschedule);
      return schedulebean;

}


@DeleteMapping("/Admin/DelSchedule/{Id}")
public ScheduleBean delSchedule(@PathVariable("Id") String
delScheduleID) {
      ScheduleBean
schedulebean=adminService.deleteSchedule(delScheduleID);
      return schedulebean;
}


@GetMapping("/Admin/ListSchedule")
public List<ScheduleBean> listSchedule(){
      List<ScheduleBean> schedulebean=adminService.scheduleList();
      return schedulebean;
}

@PutMapping("/Admin/UpdateSchedule")
public ScheduleBean updateSchedule(@RequestBody ScheduleBean
updateSchedule) {
      ScheduleBean
schedulebean=adminService.updateSchedule(updateSchedule);
      return schedulebean;
}
```

```
//-----------------------------LOGIN ADMIN
CONTROLLER---------------------------


      @GetMapping("/login/{username}/{password}")
      public CredentialsBean login(@PathVariable("username")
String userName, @PathVariable("password") String password) {

            CredentialsBean loginCredential = new
CredentialsBean(userName, password, null, 0);
            return adminService.login(loginCredential);
      }

      @PostMapping("/logout")
      public CredentialsBean logout(@RequestBody CredentialsBean
logoutCredential)
      {

      return adminService.logout(logoutCredential);
}

      @PutMapping("/Admin/changePassword/{userId}/{oldPassword}/{n
ewPassword}")
      public CredentialsBean
adminChangePassword(@PathVariable("userId") String userId,
@PathVariable("oldPassword") String oldPassword,
@PathVariable("newPassword") String newPassword)
      {
            return adminService.changePassword(userId, oldPassword,
newPassword);
      }
```

CUSTOMER CONTROLLER

```
      @PutMapping("/Customer/changePassword/{userId}/{oldPassword}
/{newPassword}")
      public CredentialsBean
customerChangePassword(@PathVariable("userId") String userId,
@PathVariable("oldPassword") String oldPassword,
@PathVariable("newPassword") String newPassword)
      {
            return customerService.changePassword(userId,
oldPassword, newPassword);
      }
      //--------------------------------------
BookTicket----------------------------------

      @PostMapping("/bookTicket")
      public ReservationBean bookTicket(@RequestBody
ReservationBean reservationBean )
      {
            ReservationBean reservationbeanBook=
dao.bookTicket(reservationBean);
            return reservationbeanBook;
      }

      @PutMapping("/cancelTicket")
      public ReservationBean cancelTicket(@RequestBody
ReservationBean reservationBean )
      {
            ReservationBean reservationbeanBook=
dao.cancelTicket(reservationBean);
            return reservationbeanBook;
      }

     //-------------------------------------------ADD
PASSENGER----------------------------------

          @PostMapping("/Customer/AddPassenger")
          public PassengerBean addPassenger(@RequestBody
PassengerBean addPassenger) {
                System.out.println(addPassenger);
                PassengerBean
addnewPassenger=customerService.addpassenger(addPassenger);
                return addnewPassenger;
          }

          //-------------------------------------PROFILE
 CONTROLLER------------------------------------------------------

          @PostMapping("/Customer/AddProfile")
          public ProfileBean addProfile(@RequestBody ProfileBean
 addProfile) {
                 System.out.println(addProfile);
                 ProfileBean
 addNewProfile=customerService.addProfile(addProfile);
          return addNewProfile;
          }
          @DeleteMapping("/Customer/DelProfile/{Id}")
          public ProfileBean delProfile(@PathVariable("Id")
String userID) {

           ProfileBean
delProfile=customerService.delProfile(userID);
           return delProfile;
          }
```

```
            @PutMapping("/Customer/updateProfile")
            public ProfileBean updateProfile(@RequestBody
ProfileBean updateProfile) {

              ProfileBean
updateprofile=customerService.updateProfile(updateProfile);
              return updateProfile;
            }

      }
```