

Name:Prathamesh Suresh Veer

1.Creation of Backend

```
create table flights(flight_id varchar(100) primary key, name varchar(100) not null ,  
depart varchar(200) not null, arrive varchar(200) not null, departure_date DATE NOT NULL);  
  
create table customers(cust_id int, first_name varchar(200) not null, last_name varchar(200) not null,  
cust_email varchar(200)not null,  
cust_phone varchar(200)not null, pnr varchar(100) primary key ,seat_no int, flight_id varchar(100)  
,foreign key (flight_id) references flights(flight_id)  
) ;
```

Customer Table Data:

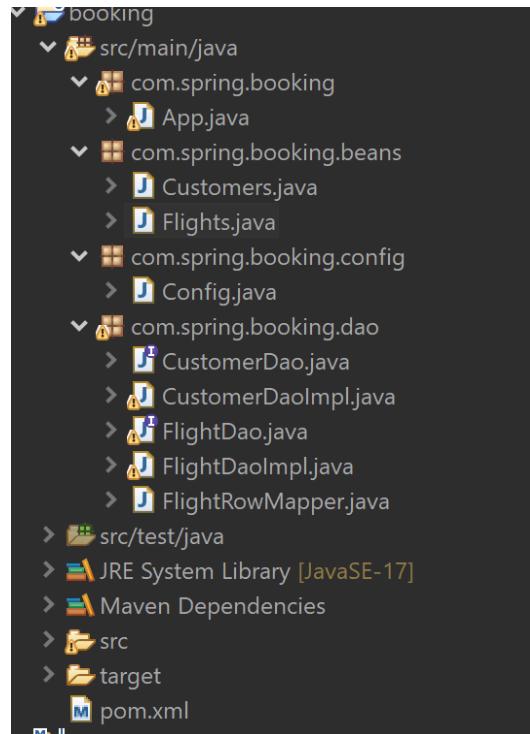
	cust_id integer	first_name character varying (200)	last_name character varying (200)	cust_email character varying (200)	cust_phone character varying (200)	pnr [PK] character varying (100)	seat_no integer	flight_id character varying (100)
4	4	Anjali	Singh	anjali.singh@example.com	6543210987	PNR12348	9	F12348
5	5	Vikram	Rao	vikram.rao@example.com	5432109876	PNR12349	16	F12349
6	6	Sneha	Verma	sneha.verma@example.com	4321098765	PNR12350	2	F12350
7	7	Ravi	Das	ravi.das@example.com	3210987654	PNR12351	8	F12351
8	8	Pooja	Mehta	pooja.mehta@example.com	2109876543	PNR12352	14	F12352
9	9	Anil	Yadav	anil.yadav@example.com	1098765432	PNR12353	10	F12353
10	10	Nisha	Kaur	nisha.kaur@example.com	0987654321	PNR12354	1	F12354
11	1	Rahul	Sharma	rahul.sharma@example.com	9876543210	PNR12380	13	F12347
12	11	Prathamesh	Veer	prathamv31.08@gmail.com	7058090146	PNR12381	22	F12347
13	11	Prathamesh	Veer	prathamv31.08@gmail.com	7058090146	PNR12382	21	F12351

Flights Table Data:

	flight_id [PK] character varying (100)	name character varying (100)	depart character varying (200)	arrive character varying (200)	departure_date date
1	F12345	Air India	Mumbai	Delhi	2024-07-01
2	F12346	IndiGo	Bangalore	Delhi	2024-07-02
3	F12347	SpiceJet	Hyderabad	Kolkata	2024-07-03
4	F12348	Vistara	Delhi	Kolkata	2024-07-04
5	F12349	GoAir	Chennai	Pune	2024-07-05
6	F12350	AirAsia India	Ahmedabad	Pune	2024-07-06
7	F12351	Alliance Air	Kochi	Pune	2024-07-07
8	F12352	TruJet	Pune	Goa	2024-07-08
9	F12353	Star Air	Bhopal	Mumbai	2024-07-09
10	F12354	Air Odisha	Ranchi	Mumbai	2024-07-10

2.Design Appropriate Interface and Classes

Directory Structure:



Customer Class:

```
1 package com.spring.booking.beans;
2
3 import org.springframework.stereotype.Component;
4
5 @Component
6 public class Customers {
7     int custId;
8
9
10    String fname;
11    String lname;
12    String email;
13    String phone;
14    String pnr;
15    public Customers(int custId, String fname, String lname, String email, String phone, String pnr, int seatNo, String flightId);
16
17    public String getFlightId() {}
18    public void setFlightId(String flightId) {}
19    public String getFname() {}
20    public void setFname(String fname) {}
21    public String getLname() {}
22    public void setLname(String lname) {}
23    public String getEmail() {}
24    public void setEmail(String email) {}
25    public String getPhone() {}
26    public void setPhone(String phone) {}
27    public String getPnr() {}
28    public void setPnr(String pnr) {}
29    public int getSeatNo() {}
30    public void setSeatNo(int seatNo) {}
31
32    public int getCustId() {}
33    public void setCustId(int custId) {}
34    public String toString() {}
35
36
37 }
```

Flights Class:

```
1 package com.spring.booking.beans;
2
3 import org.springframework.stereotype.Component;
4
5 @Component
6 public class Flights {
7     String pnr;
8     String name;
9     String depart;
10    String arrive;
11    String date;
12+   public String getPnr() { }
13+   public void setPnr(String pnr) { }
14+   public String getName() { }
15+   public void setName(String name) { }
16+   public String getDepart() { }
17+   public void setDepart(String depart) { }
18+   public String getArrive() { }
19+   public void setArrive(String arrive) { }
20+   public String getDate() { }
21+   public void setDate(String date) { }
22+   public Flights(String pnr, String name, String depart, String arrive, String date) { }
23+   public Flights() { }
24
25
26 }
```

CustomerDao Interface

```
1 package com.spring.booking.dao;  
2  
3 import com.spring.booking.beans.Customers;  
4  
5 public interface CustomerDao {  
6     public int addFlight(Customers c);  
7     public int updateCust();  
8     public int cancelBooking();  
9 }  
10
```

```
import org.springframework.stereotype.Component;  
import com.spring.booking.beans.Customers;  
  
@Component  
public class CustomerDaoImpl implements CustomerDao{  
  
    @Autowired  
    JdbcTemplate jdbcTemplate;  
  
    @Override  
    public int addFlight(Customers c) {  
        // TODO Auto-generated method stub  
  
        String Query="Insert into customers values(?,?,?,?,?,?);";  
        int res=this.jdbcTemplate.update(Query,c.getCustomerId(),c.getFname(),c.getLname(),c.getEmail(),c.getPhone());  
        return res;  
    }  
  
    @Override  
    public int updateCust() {  
        // TODO Auto-generated method stub  
        Scanner sc=new Scanner(System.in);  
        System.out.println("Enter the id of customers who details needs to be updated:");  
        String pnr=sc.next();  
        System.out.println("Enter the new email to be updated:");  
        String email=sc.next();  
        System.out.println("Enter the mobile number to be update:");  
        String phone=sc.next();  
  
        String query="Update customers set cust_email=?,cust_phone=? where pnr=?";  
        return this.jdbcTemplate.update(query,email,phone,pnr);  
    }  
  
    @Override  
    public int cancelBooking() {  
        // TODO Auto-generated method stub  
        Scanner sc=new Scanner(System.in);  
        System.out.println("Enter the PNR of customer whose flight need to be canceled :");  
        String pnr=sc.next();  
        String query="Delete from customers where pnr=?";  
        return this.jdbcTemplate.update(query,pnr);  
    }  
}
```

CustomerDao Class

FlightDao Interface

```
1 package com.spring.booking.dao;
2
3 import java.util.List;
4
5 import com.spring.booking.beans.Customers;
6 import com.spring.booking.beans.Flights;
7
8 public interface FlightDao {
9
10    public void displayFlightByDest(String s);
11    public void displayFlightByCustomer();
12
13
14 }
15
```

FlightDaoImpl Class

```
import org.springframework.jdbc.core.JdbcTemplate;
import org.springframework.jdbc.core.RowMapper;
import org.springframework.stereotype.Component;

import com.spring.booking.beans.Customers;
import com.spring.booking.beans.Flights;

@Component
public class FlightDaoImpl implements FlightDao {
    @Autowired
    JdbcTemplate jdbcTemplate;
    @Override
    public void displayFlightByDest(String s) {
        String query="Select * from flights where arrive=?";
        List<Flights> list=this.jdbcTemplate.query(query,new FlightRowMapper(),s);

        for(Flights flight:list) {
            System.out.println("Flight PNR :"+flight.getPNr());
            System.out.println("Flight Name :"+flight.getName());
            System.out.println("Flight From :"+flight.getDepart());
            System.out.println("Departure Date :"+flight.getDate());
            System.out.println();
        }
    }
    @Override
    public void displayFlightByCustomer() {
        // TODO Auto-generated method stub
        Scanner sc=new Scanner(System.in);
        System.out.println("Enter the customer id");
        int id=sc.nextInt();
        String Query="Select f.* from flights f join customers c on f.flight_id=c.flight_id where cust_id=?";
        System.out.println("Flights booked by customer with Customer Id :"+id+" :");
        List<Flights>list=this.jdbcTemplate.query(Query, new FlightRowMapper(), id);
        for(Flights flight:list) {
            System.out.println("Flight PNR :"+flight.getPNr());
            System.out.println("Flight Name :"+flight.getName());
            System.out.println("Flight From :"+flight.getDepart());
            System.out.println("Departure Date :"+flight.getDate());
            System.out.println();
        }
    }
}
```

RowMapperClass

```
1 package com.spring.booking.dao;
2
3 import java.sql.ResultSet;
4 import java.sql.SQLException;
5
6 import org.springframework.jdbc.core.RowMapper;
7
8 import com.spring.booking.beans.Flights;
9
10
11 public class FlightRowMapper implements RowMapper<Flights>{
12
13     @Override
14     public Flights mapRow(ResultSet rs, int rowNum) throws SQLException {
15         // TODO Auto-generated method stub
16
17         Flights flight=new Flights();
18         flight.setPnr(rs.getString(1));
19         flight.setName(rs.getString(2));
20         flight.setDepart(rs.getString(3));
21         flight.setArrive(rs.getString(4));
22         flight.setDate(rs.getString(5));
23
24         return flight;
25     }
26 }
27
28 }
```

Config Class

```
1 package com.spring.booking.config;
2
3 import javax.sql.DataSource;
4
5 import org.springframework.context.annotation.Bean;
6 import org.springframework.context.annotation.ComponentScan;
7 import org.springframework.context.annotation.Configuration;
8 import org.springframework.jdbc.core.JdbcTemplate;
9 import org.springframework.jdbc.datasource.DriverManagerDataSource;
10
11 @Configuration
12 @ComponentScan("com.spring.booking.*")
13 public class Config {
14
15     private String dbuser="postgres";
16     private String dbpassword="3108";
17     //api:database://hostname:portno/databasename
18     private String dburl="jdbc:postgresql://localhost:5432/flight";
19
20     @Bean("ds")
21     public DataSource getDataSource() {
22         DriverManagerDataSource ds=new DriverManagerDataSource();
23         ds.setDriverClassName("org.postgresql.Driver");
24         ds.setUsername(dbuser);
25         ds.setPassword(dbpassword);
26         ds.setUrl(dburl);
27         return ds;
28     }
29
30
31     @Bean("JdbcTemplate")
32     public JdbcTemplate getJdbcTemplate() {
33         JdbcTemplate jd=new JdbcTemplate();
34         jd.setDataSource(getDataSource());
35         return jd;
36     }
37
38
39
40 }
```

App Class

```
package com.spring.booking;

import java.util.Scanner;

import org.springframework.context.ApplicationContext;
import org.springframework.context.annotation.AnnotationConfigApplicationContext;

import com.spring.booking.beans.Customers;
import com.spring.booking.config.Config;
import com.spring.booking.dao.CustomerDaoImpl;
import com.spring.booking.dao.FlightDaoImpl;

/**
 * Hello world!
 *
 */
public class App
{
    public static void main( String[] args )
    {
        AnnotationConfigApplicationContext context=new AnnotationConfigApplicationContext(Config.class);

        CustomerDaoImpl Cust=context.getBean(CustomerDaoImpl.class);
        FlightDaoImpl flight=context.getBean(FlightDaoImpl.class);
        flight.displayFlightByDest("Mumbai");

        Scanner sc=new Scanner(System.in);

        System.out.println("Enter 1 to Book a flight ");
        System.out.println("Enter 2 to update customer details ");
        System.out.println("Enter 3 to get all flights for a particular destination");
        System.out.println("Enter 4 to cancel a booking ");
        System.out.println("Enter 5 the flights booked by a particular customer");
        System.out.println();
        int ch=sc.nextInt();

        switch (ch) {
        case 1: {
            System.out.println("Enter id:");
            int id=sc.nextInt();
            System.out.println("Enter the first Name:");
            String fname=sc.next();
            System.out.println("Enter the last Name:");
            String lname=sc.next();
            System.out.println("Enter the email");
            String email=sc.next();

            System.out.println("Enter the flight Number :");
            String flightNumber=sc.next();
            Customers C=new Customers(id,fname,lname,email,phone,PNR,seatNo,flightNumber);
            int res=Cust.addFlight(C);
            if(res==1) {
                System.out.println("Customer successfully added");
            }
            else {
                System.out.println("Operation failed");
            }
            break;
        }
        case 2:{ 
            int res=Cust.updateCust();
            if(res==1) {
                System.out.println("Customer details successfully updated");
            }
            else {
                System.out.println("Operation failed");
            }
            break;
        }
        case 3:{ 
            System.out.println("Enter the destination:");
            String dest=sc.next();
            flight.displayFlightByDest(dest);
            break;
        }
        case 4:{ 
            int res=Cust.cancelBooking();
            if(res==1) {
                System.out.println("Booking canceled");
            }
            else {
                System.out.println("Operation failed");
            }
            break;
        }
        case 5:{ 
            flight.displayFlightByCustomer();
            break;
        }
        default:
            throw new IllegalArgumentException("Unexpected value: " + ch);
        }
    }
}
```

3 Book A New Flight

```
Enter 2 to update customer details
Enter 3 to get all flights for a particular destination
Enter 4 to cancel a booking
Enter 5 the flights booked by a particular customer

1
Enter id:
11
Enter the First Name:
Prathamesh
Enter the last Name:
Veer
Enter the email
prathamv31.08@gmail.com
Enter the phone number:
7058090146
Enter Unique PNR :

PNR12382\
Enter the seat number
23
Enter the flight Number :
F12349
Customer successfully added
```

4.Update customer Details

```
Flight PNR :F12353
Flight Name :Star Air
Flight From :Bhopal
Departure Date :2024-07-09

Flight PNR :F12354
Flight Name :Air Odisha
Flight From :Ranchi
Departure Date :2024-07-10

Enter 1 to Book a flight
Enter 2 to update customer details
Enter 3 to get all flights for a particular destination
Enter 4 to cancel a booking
Enter 5 the flights booked by a particular customer

2
Enter the id of customers who details needs to be updates:
2
Enter the new email to be updated:
Pratham@gmail.com
Enter the mobile number to be update:
707789999
```

5.Display files for a particular destination

```
Enter 1 to Book a flight
Enter 2 to update customer details
Enter 3 to get all flights for a particular destination
Enter 4 to cancel a booking
Enter 5 the flights booked by a particular customer

3
Enter the destination:
Mumbai
Flight PNR :F12353
Flight Name :Star Air
Flight From :Bhopal
Departure Date :2024-07-09

Flight PNR :F12354
Flight Name :Air Odisha
Flight From :Ranchi
Departure Date :2024-07-10
```

6.Cancel Booking

```
Enter 1 to Book a flight
Enter 2 to update customer details
Enter 3 to get all flights for a particular destination
Enter 4 to cancel a booking
Enter 5 the flights booked by a particular customer

4
Enter the PNR of customer whose flight need to be canceled :
PNR12346
Booking Cancelled
```

7.Display the files by a particular customer

```
Enter 1 to Book a flight
Enter 2 to update customer details
Enter 3 to get all flights for a particular destination
Enter 4 to cancel a booking
Enter 5 the flights booked by a particular customer

5
Enter the customer id
11
Flights booked by customer with Customer Id :11 :
Flight PNR :F12347
Flight Name :SpiceJet
Flight From :Hyderabad
Departure Date :2024-07-03

Flight PNR :F12351
Flight Name :Alliance Air
Flight From :Kochi
Departure Date :2024-07-07

Flight PNR :F12349
Flight Name :GoAir
Flight From :Chennai
```

