

Automatic evaluation[+]

FlightMgmt/src/com/dao/BookFlightDAO.java

```
1 package com.dao;
2
3 import java.time.LocalDate;
4 import java.util.ArrayList;
5 import java.util.List;
6
7 import com.model.Flight;
8 import com.exception.InvalidFlightException;
9 import com.model.BookFlight;
10 import com.model.Customer;
11
12 public class BookFlightDAO {
13
14     List<BookFlight> bookingList = new ArrayList<>();
15
16     public boolean    bookAFlight(final Customer customer,final Flight flight,final
LocalDate dateOfbooking, LocalDate flightDate,int noOfPassengers){
17
18         BookFlight          bookObj          =          new
BookFlight(customer,flight,dateOfbooking,flightDate,noOfPassengers);
19
20         //boolean flag=true;
21         //boolean result = bookingList.add(bookObj);
22         //return result;
23         return bookingList.add(bookObj);
24     }
25
26     public List<BookFlight>viewBookingByFlight(Flight flightObj) throws
InvalidFlightException {
27         final List<BookFlight> temp = new ArrayList<>();
28
29         for(final BookFlight booking : bookingList){
30             if(booking.getFlight().getFlightId()==flightObj.getFlightId()) {
31                 temp.add(booking);
32             }
33         }
34         if(temp.isEmpty()){
35             throw new InvalidFlightException("No booking for Flight
"+flightObj.getFlightId());
36         }
37         return temp;
38     }
39 }
40 }
41
42
43
```

FlightMgmt/src/com/dao/CustomerDAO.java

```
1 package com.dao;
2
3 import java.util.*;
4
5 import com.exception.InvalidCustomerException;
6 import com.model.Customer;
7
```



```

8 public class CustomerDAO {
9
10     List<Customer>customerList = new ArrayList<>();
11
12     public void addCustomer(Customer customer){
13         customerList.add(customer);
14     }
15
16     public Customer viewCustomerByUserName(String userName){
17         try{
18             if(customerList.isEmpty()){
19                 throw new InvalidCustomerException("User Name is invalid");
20             }
21             else{
22                 for(Customer c : customerList) {
23                     if(c.getUserName().equals(userName)){
24                         return c;
25                     }
26                 }
27             }
28         }
29         catch(Exception e){
30             System.out.println(e.getMessage());
31         }
32         return null;
33     }
34
35     public boolean validateCustomer(String userName,String password){
36         //boolean flag=false;
37
38         for(Customer c : customerList) {
39             if(c.getUserName().equals(userName)                &&
c.getPassword().equals(password)) {
40                 return true;
41             }
42         }
43
44         return false;
45     }
46
47 }
48

```

FlightMgmt/src/com/dao/FlightDAO.java

```

1 package com.dao;
2
3 import java.util.*;
4
5 import com.exception.*;
6 import com.model.*;
7
8 public class FlightDAO {
9
10     List<Flight>flightList = new ArrayList<>();
11
12     public void addCourse(Flight flightObj){
13         flightList.add(flightObj);
14     }
15

```

```

15
16 public List<Flight>viewFlightBySourceDestination(String source,String destination)
throws InvalidFlightException {
17     List<Flight>temp=new ArrayList<>();
18     for(Flight f : flightList) {
19         if(f.getSource().equals(source) && f.getDestination().equals(destination)) {
20             temp.add(f);
21         }
22     }
23     if(temp.isEmpty()){
24         throw new InvalidFlightException("No Flight with source "+source+" and
destination "+destination);
25     }
26     else{
27         return temp;
28     }
29 }
30
31 }
32

```

FlightMgmt/src/com/exception/InvalidCustomerException.java

```

1 package com.exception;
2
3 public class InvalidCustomerException extends Exception {
4
5     public InvalidCustomerException(String msg){
6         super(msg);
7     }
8
9 }
10

```

FlightMgmt/src/com/exception/InvalidFlightException.java

```

1 package com.exception;
2
3 public class InvalidFlightException extends Exception {
4
5     public InvalidFlightException(String msg){
6         super(msg);
7     }
8
9 }
10

```

FlightMgmt/src/com/model/BookFlight.java

```

1 package com.model;
2
3 import java.time.LocalDate;
4 //import java.util.Date;
5
6 public class BookFlight {
7
8     private Customer customer;
9     private Flight flight;
10    private LocalDate dateOfbooking;
11    private LocalDate flightDate;
12    private int noOfPassengers;
13    private double totalFare;
14

```



```

15 //public BookFlight(){
16 //
17 //}
18
19 public BookFlight(Customer customer, Flight flight, LocalDate dateOfbooking,
LocalDate flightDate,
20         int noOfPassengers) {
21     super();
22     this.customer = customer;
23     this.flight = flight;
24     this.dateOfbooking = dateOfbooking;
25     this.flightDate = flightDate;
26     this.noOfPassengers = noOfPassengers;
27     this.totalFare = totalFare;
28 }
29
30 public Customer getCustomer() {
31     return customer;
32 }
33
34 public void setCustomer(Customer customer) {
35     this.customer = customer;
36 }
37
38 public Flight getFlight() {
39     return flight;
40 }
41
42 public void setFlight(Flight flight) {
43     this.flight = flight;
44 }
45
46 public LocalDate getDateOfbooking() {
47     return dateOfbooking;
48 }
49
50 public void setDateOfbooking(LocalDate dateOfbooking) {
51     this.dateOfbooking = dateOfbooking;
52 }
53
54 public LocalDate getFlightDate() {
55     return flightDate;
56 }
57
58 public void setFlightDate(LocalDate flightDate) {
59     this.flightDate = flightDate;
60 }
61
62 public int getNoOfPassengers() {
63     return noOfPassengers;
64 }
65
66 public void setNoOfPassengers(int noOfPassengers) {
67     this.noOfPassengers = noOfPassengers;
68 }
69
70 public double getTotalFare() {

```

```

71     return totalFare;
72 }
73
74 public void setTotalFare(double totalFare) {
75     this.totalFare = totalFare;
76 }
77
78 public float calculateTotalFare(){
79     return (float)noOfPassengers*flight.getFlightFare();
80     //return totalFare;
81 }
82
83 public boolean validateNoOfPassengers(){
84     boolean flag=false;
85     if(noOfPassengers >0 && noOfPassengers <= 30) {
86         flag=true;
87     }
88     return flag;
89 }
90 }
91
FlightMgmt/src/com/model/Customer.java
1 package com.model;
2
3 public class Customer {
4
5     private String customerId;
6     private String customerName;
7     private String emailId;
8     private String userName;
9     private String password="FH782";
10
11     public Customer(String customerName, String emailId, String userName) {
12         this.customerId = customerId;
13         this.customerName = customerName;
14         this.emailId = emailId;
15         this.userName = userName;
16     }
17
18     public String getCustomerId() {
19         return customerId;
20     }
21
22     public void setCustomerId(String customerId) {
23         this.customerId = customerId;
24     }
25
26     public String getCustomerName() {
27         return customerName;
28     }
29
30     public void setCustomerName(String customerName) {
31         this.customerName = customerName;
32     }
33
34     public String getEmailId() {
35         return emailId;

```



```

36 }
37
38 public void setEmailId(String emailId) {
39     this.emailId = emailId;
40 }
41
42 public String getUser_name() {
43     return user_name;
44 }
45
46 public void setUser_name(String user_name) {
47     this.user_name = user_name;
48 }
49
50 public String getPassword() {
51     return password;
52 }
53
54 public void setPassword(String password) {
55     this.password = password;
56 }
57
58
59
60 }
61
FlightMgmt/src/com/model/Flight.java
1 package com.model;
2
3 public class Flight {
4
5     public int flightId;
6     private String flightName;
7     private String source;
8     private String destination;
9     private float flightFare;
10    private int noOfSeats = 40;
11    static final String COMPANYNAME = "Aviva Airlines";
12
13    public Flight(int flightId, String flightName, String source, String destination, float
fare) {
14
15        this.flightId = flightId;
16        this.flightName = flightName;
17        this.source = source;
18        this.destination = destination;
19        this.flightFare = fare;
20
21    }
22
23    public int getFlightId() {
24        return flightId;
25    }
26
27    public void setFlightId(int flightId) {
28        this.flightId = flightId;
29    }

```



```

30
31 public String getFlightName() {
32     return flightName;
33 }
34
35 public void setFlightName(String flightName) {
36     this.flightName = flightName;
37 }
38
39 public String getSource() {
40     return source;
41 }
42
43 public void setSource(String source) {
44     this.source = source;
45 }
46
47 public String getDestination() {
48     return destination;
49 }
50
51 public void setDestination(String destination) {
52     this.destination = destination;
53 }
54
55 public float getFlightFare() {
56     return flightFare;
57 }
58
59 public void setFare(float fare) {
60     this.flightFare = fare;
61 }
62
63 public int getNoOfSeats() {
64     return noOfSeats;
65 }
66
67 public void setNoOfSeats(int noOfSeats) {
68     this.noOfSeats = noOfSeats;
69 }
70
71 public static String getCompanyname() {
72     return COMPANYNAME;
73 }
74 }
75

```

Grade

Reviewed on Wednesday, 12 May 2021, 10:43 PM by Automatic grade

Grade 100 / 100

Assessment report

[+]Grading and Feedback

