Airline booking System

Name : Alanoud Khaled Bo-Dokhi

GitHub : https://github.com/ialanoudk/Airline-Booking-FlyAway/tree/master

Table of Contents
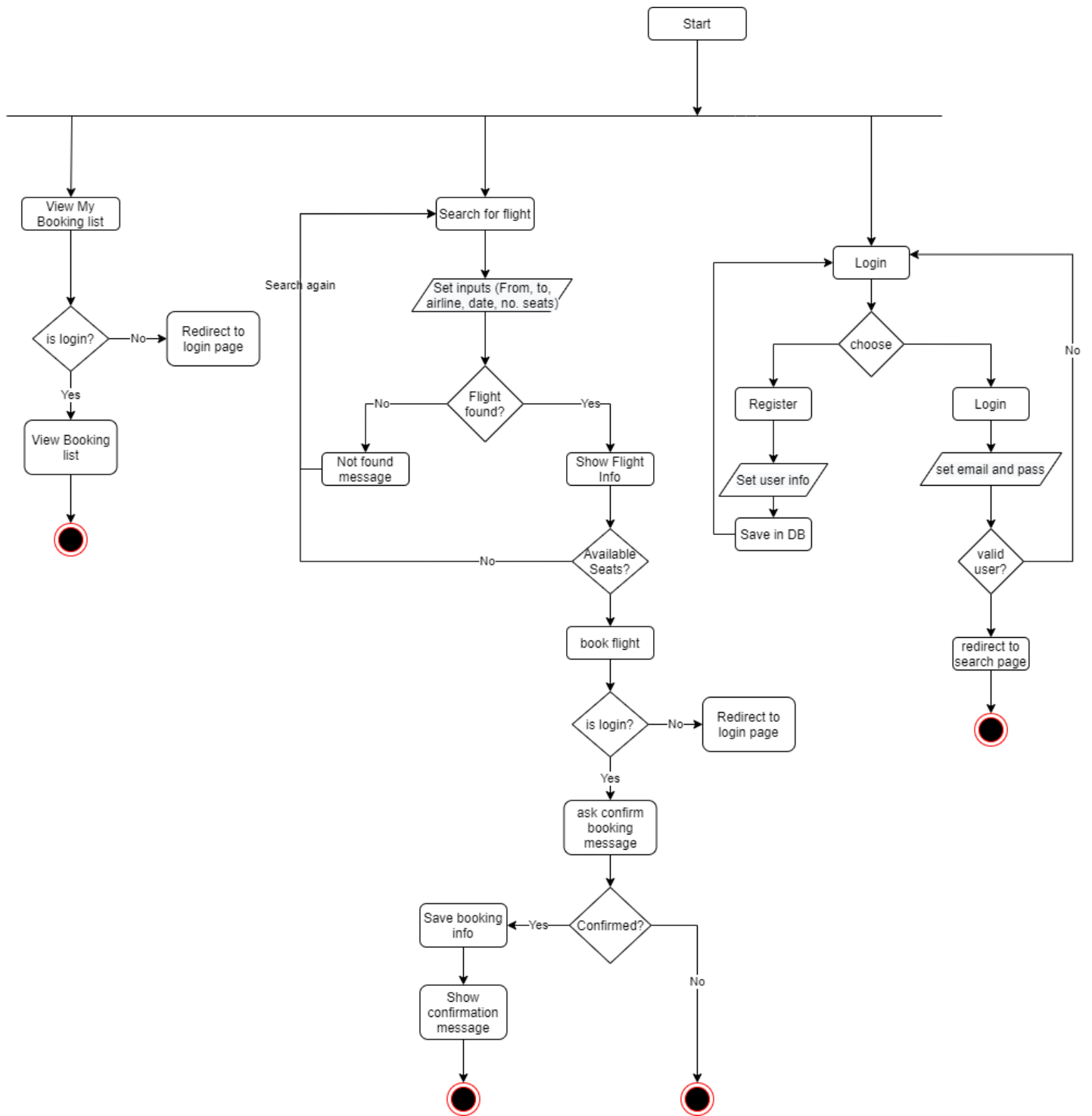
1. Airline Booking System

1.1 Project Description

The project is an online booking system to book flights based on information provided by passengers. The project is a dynamic web application (JSP) created by Maven with MySQL database. Users can select the location of departure, location of arrival or destination, date, which airline, and a number of needed seats in order to find available flights. If a flight is found, the system calculates if the required number of seats are available in the flight or not, if seats are less than required the user cannot book the flight. The user must sign up in order to complete the booking process.
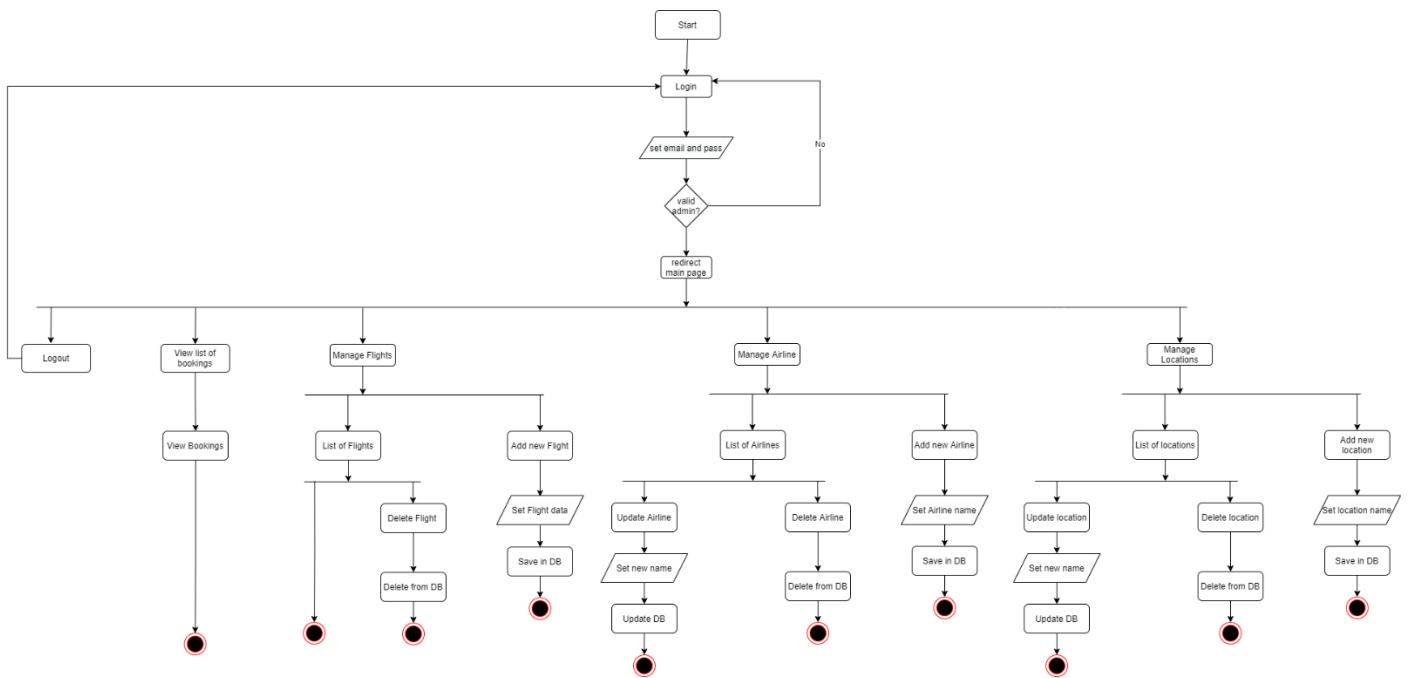
On the other hand, administrator can manage locations, airlines, and flights. Admin can add, edit or delete information, also can view all booking done by users.
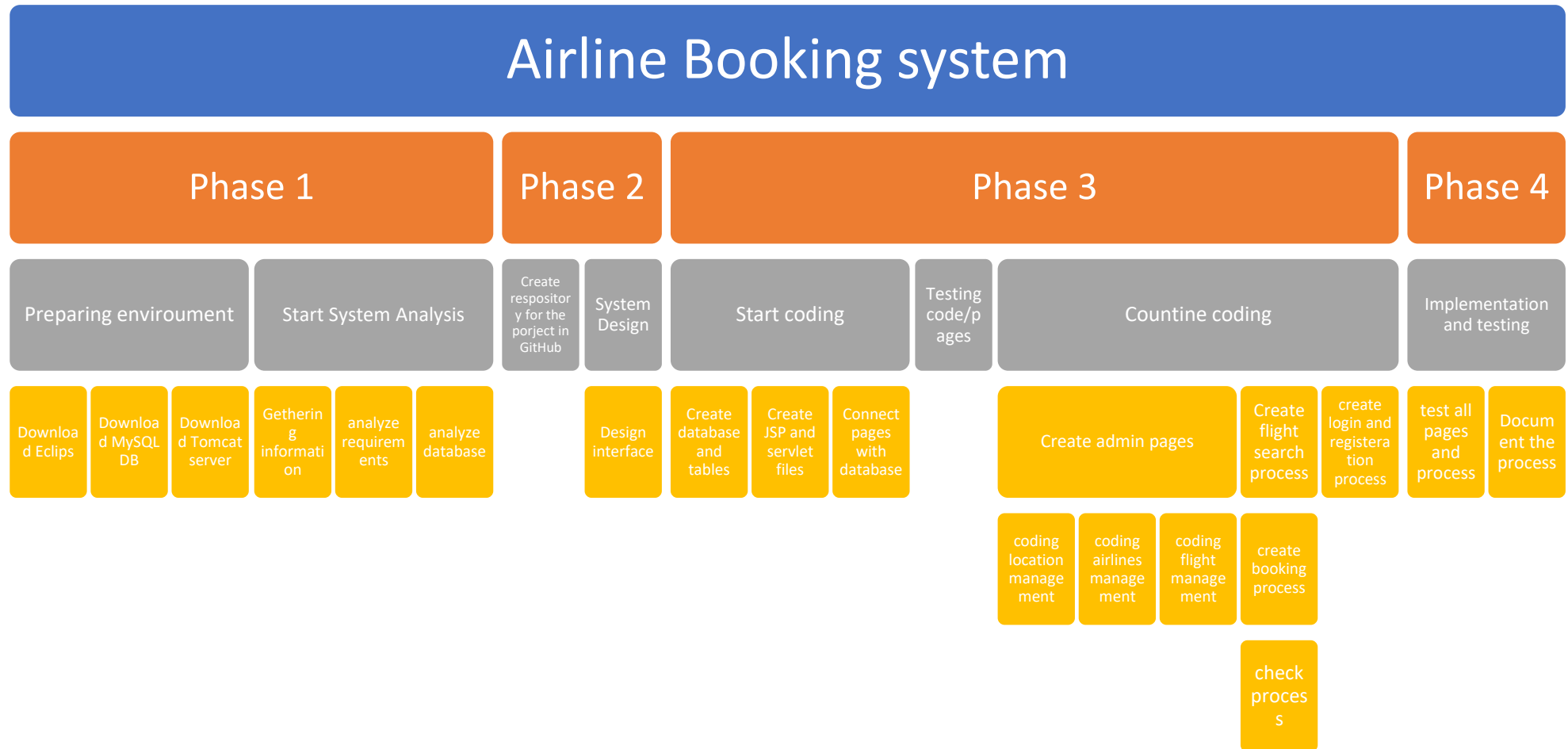
2. Architecture diagram / flow chart

- User Workflow

Start

View My Booking list

is login?
No → Redirect to login page
Yes

View Booking list

Search for flight

Set inputs (From, to, airline, date, no. seats)

Search again

Flight found?
No → Not found message
Yes → Show Flight Info

Available Seats?
No

book flight

is login?
No → Redirect to login page
Yes

ask confirm booking message

Confirmed?
Yes → Save booking info → Show confirmation message
No

Login

choose
Register
Login

Set user info
Save in DB

set email and pass

valid user?
No → Login

redirect to search page

- Admin workflow

3. Project Users Stories: (Agile and Scrum)

Airline Booking system

Phase 1	Phase 2	Phase 3	Phase 4

Preparing enviroument	Start System Analysis	Create respository for the porject in GitHub	System Design	Start coding	Testing code/pages	Countine coding	Implementation and testing

Phase 1:
- Download Eclips
- Download MySQL DB
- Download Tomcat server
- Gethering information
- analyze requirements
- analyze database

Phase 2:
- Design interface
- Create database and tables
- Create JSP and servlet files
- Connect pages with database

Phase 3:
- Create admin pages
- Create flight search process
- create login and registeration process
- coding location management
- coding airlines management
- coding flight management
- create booking process
- check process

Phase 4:
- test all pages and process
- Document the process
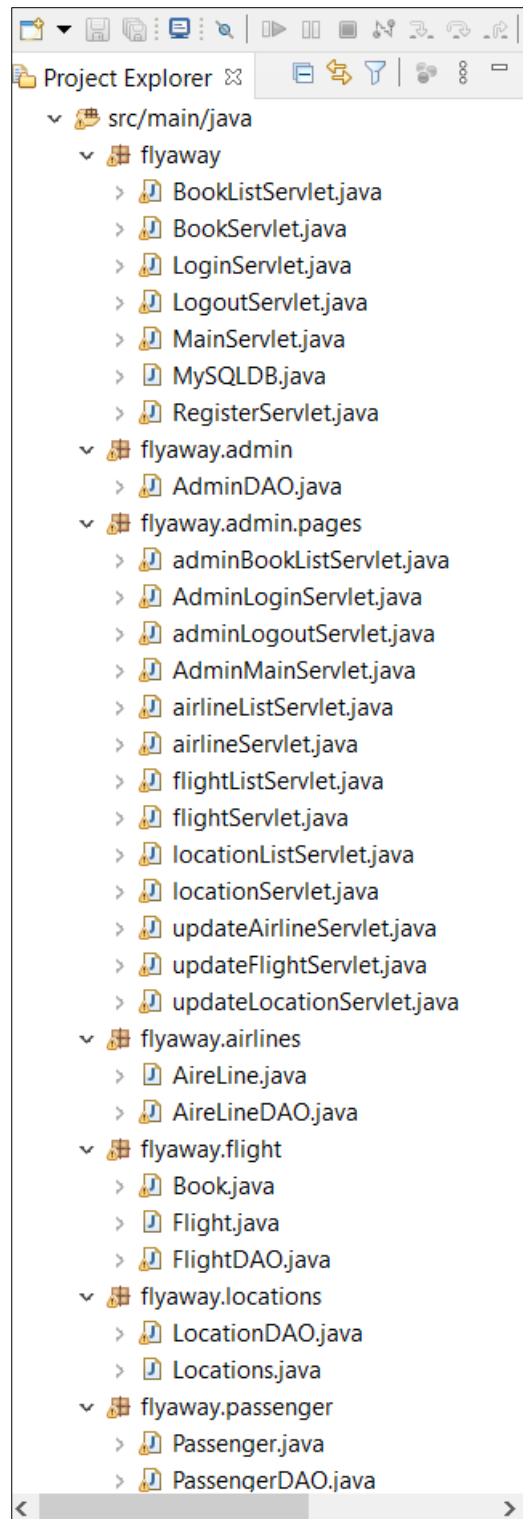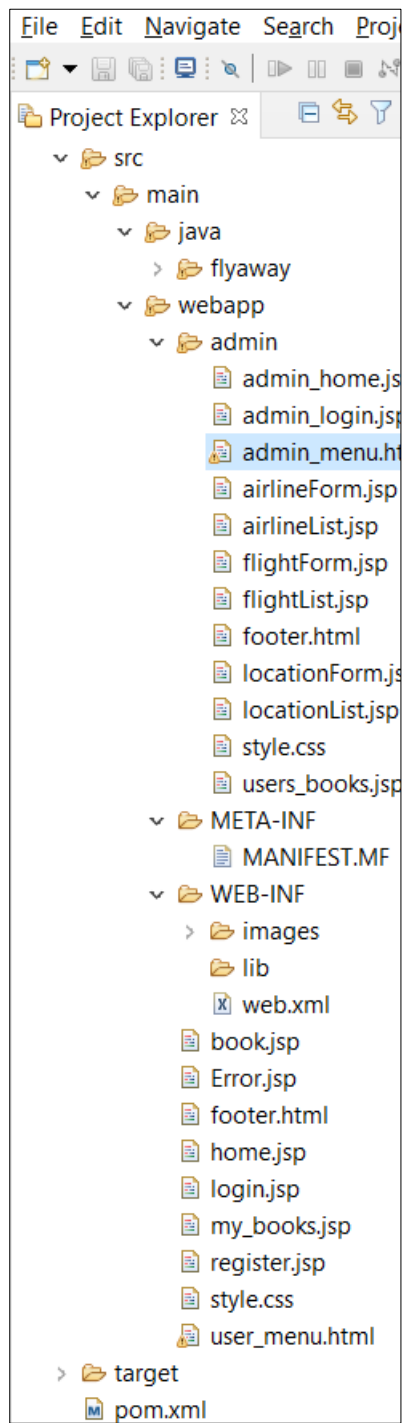
4. Source code files

Screen Shot of the Source Code:

1. Project explorer: list of all servlet files

2. Project explorer: list of all JSP files

3. Web.xml file code

```xml
<?xml version="1.0" encoding="UTF-8"?>
<web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns="http://xmlns.jcp.org/xml/ns/javaee"
    xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/javaee
    http://xmlns.jcp.org/xml/ns/javaee/web-app_3_1.xsd" id="WebApp_ID"
    version="3.1">
  <display-name>flyaway</display-name>

    <servlet>
        <servlet-name>MainServlet</servlet-name>
        <servlet-class>flyaway.MainServlet</servlet-class>
    </servlet>

    <servlet-mapping>
        <servlet-name>MainServlet</servlet-name>
        <url-pattern>/</url-pattern>
    </servlet-mapping>

    <servlet>
        <servlet-name>BookServlet</servlet-name>
        <servlet-class>flyaway.BookServlet</servlet-class>
    </servlet>

    <servlet-mapping>
        <servlet-name>BookServlet</servlet-name>
        <url-pattern>/book</url-pattern>
    </servlet-mapping>


    <servlet>
        <servlet-name>LoginServlet</servlet-name>
        <servlet-class>flyaway.LoginServlet</servlet-class>
    </servlet>

    <servlet-mapping>
        <servlet-name>LoginServlet</servlet-name>
        <url-pattern>/login</url-pattern>
    </servlet-mapping>


    <servlet>
        <servlet-name>LogoutServlet</servlet-name>
        <servlet-class>flyaway.LogoutServlet</servlet-class>
    </servlet>

    <servlet-mapping>
        <servlet-name>LogoutServlet</servlet-name>
        <url-pattern>/logout</url-pattern>
    </servlet-mapping>


    <servlet>
        <servlet-name>RegisterServlet</servlet-name>
        <servlet-class>flyaway.RegisterServlet</servlet-class>
    </servlet>

    <servlet-mapping>
```

```xml
57.          <servlet-name>RegisterServlet</servlet-name>
58.          <url-pattern>/register</url-pattern>
59.      </servlet-mapping>
60.
61.
62.
63.      <servlet>
64.          <servlet-name>BookListServlet</servlet-name>
65.          <servlet-class>flyaway.BookListServlet</servlet-class>
66.      </servlet>
67.
68.      <servlet-mapping>
69.          <servlet-name>BookListServlet</servlet-name>
70.          <url-pattern>/mybooking</url-pattern>
71.      </servlet-mapping>
72.
73.
74.
75.      <servlet>
76.          <servlet-name>AdminMainServlet</servlet-name>
77.          <servlet-
    class>flyaway.admin.pages.AdminMainServlet</servlet-class>
78.      </servlet>
79.
80.      <servlet-mapping>
81.          <servlet-name>AdminMainServlet</servlet-name>
82.          <url-pattern>/admin/</url-pattern>
83.      </servlet-mapping>
84.
85.
86.
87.      <servlet>
88.          <servlet-name>AdminLoginServlet</servlet-name>
89.          <servlet-
    class>flyaway.admin.pages.AdminLoginServlet</servlet-class>
90.      </servlet>
91.
92.      <servlet-mapping>
93.          <servlet-name>AdminLoginServlet</servlet-name>
94.          <url-pattern>/admin/login</url-pattern>
95.      </servlet-mapping>
96.
97.
98.      <servlet>
99.          <servlet-name>locationServlet</servlet-name>
100.             <servlet-
    class>flyaway.admin.pages.locationServlet</servlet-class>
101.         </servlet>
102.
103.         <servlet-mapping>
104.             <servlet-name>locationServlet</servlet-name>
105.             <url-pattern>/admin/addlocation</url-pattern>
106.         </servlet-mapping>
107.
108.
109.         <servlet>
110.             <servlet-name>updateLocationServlet</servlet-name>
111.             <servlet-
    class>flyaway.admin.pages.updateLocationServlet</servlet-class>
```

```
112.          </servlet>
113.
114.          <servlet-mapping>
115.              <servlet-name>updateLocationServlet</servlet-name>
116.              <url-pattern>/admin/updatelocation</url-pattern>
117.          </servlet-mapping>
118.
119.
120.
121.          <servlet>
122.              <servlet-name>locationListServlet</servlet-name>
123.              <servlet-
     class>flyaway.admin.pages.locationListServlet</servlet-class>
124.          </servlet>
125.
126.          <servlet-mapping>
127.              <servlet-name>locationListServlet</servlet-name>
128.              <url-pattern>/admin/listlocations</url-pattern>
129.          </servlet-mapping>
130.
131.
132.
133.          <servlet>
134.              <servlet-name>airlineServlet</servlet-name>
135.              <servlet-
     class>flyaway.admin.pages.airlineServlet</servlet-class>
136.          </servlet>
137.
138.          <servlet-mapping>
139.              <servlet-name>airlineServlet</servlet-name>
140.              <url-pattern>/admin/addairline</url-pattern>
141.          </servlet-mapping>
142.
143.
144.          <servlet>
145.              <servlet-name>updateAirlineServlet</servlet-name>
146.              <servlet-
     class>flyaway.admin.pages.updateAirlineServlet</servlet-class>
147.          </servlet>
148.
149.          <servlet-mapping>
150.              <servlet-name>updateAirlineServlet</servlet-name>
151.              <url-pattern>/admin/updateairline</url-pattern>
152.          </servlet-mapping>
153.
154.
155.
156.          <servlet>
157.              <servlet-name>airlineListServlet</servlet-name>
158.              <servlet-
     class>flyaway.admin.pages.airlineListServlet</servlet-class>
159.          </servlet>
160.
161.          <servlet-mapping>
162.              <servlet-name>airlineListServlet</servlet-name>
163.              <url-pattern>/admin/listairlines</url-pattern>
164.          </servlet-mapping>
165.
166.
```

```xml
167.
168.          <servlet>
169.                  <servlet-name>flightServlet</servlet-name>
170.                  <servlet-
      class>flyaway.admin.pages.flightServlet</servlet-class>
171.          </servlet>
172.
173.          <servlet-mapping>
174.                  <servlet-name>flightServlet</servlet-name>
175.                  <url-pattern>/admin/addflight</url-pattern>
176.          </servlet-mapping>
177.
178.
179.          <servlet>
180.                  <servlet-name>updateFlightServlet</servlet-name>
181.                  <servlet-
      class>flyaway.admin.pages.updateFlightServlet</servlet-class>
182.          </servlet>
183.
184.          <servlet-mapping>
185.                  <servlet-name>updateFlightServlet</servlet-name>
186.                  <url-pattern>/admin/updateflight</url-pattern>
187.          </servlet-mapping>
188.
189.
190.
191.          <servlet>
192.                  <servlet-name>flightListServlet</servlet-name>
193.                  <servlet-
      class>flyaway.admin.pages.flightListServlet</servlet-class>
194.          </servlet>
195.
196.          <servlet-mapping>
197.                  <servlet-name>flightListServlet</servlet-name>
198.                  <url-pattern>/admin/listflights</url-pattern>
199.          </servlet-mapping>
200.
201.
202.
203.          <servlet>
204.                  <servlet-name>adminBookListServlet</servlet-name>
205.                  <servlet-
      class>flyaway.admin.pages.adminBookListServlet</servlet-class>
206.          </servlet>
207.
208.          <servlet-mapping>
209.                  <servlet-name>adminBookListServlet</servlet-name>
210.                  <url-pattern>/admin/usersbooking</url-pattern>
211.          </servlet-mapping>
212.
213.
214.          <servlet>
215.                  <servlet-name>adminLogoutServlet</servlet-name>
216.                  <servlet-
      class>flyaway.admin.pages.adminLogoutServlet</servlet-class>
217.          </servlet>
218.
219.          <servlet-mapping>
220.                  <servlet-name>adminLogoutServlet</servlet-name>
```

```
221.               <url-pattern>/admin/logout</url-pattern>
222.          </servlet-mapping>
223.          <error-page>
224.              <exception-type>java.lang.Exception</exception-type>
225.              <location>/Error.jsp</location>
226.          </error-page>
227.     </web-app>
```

5. Output Screen shot

1. Home Screen
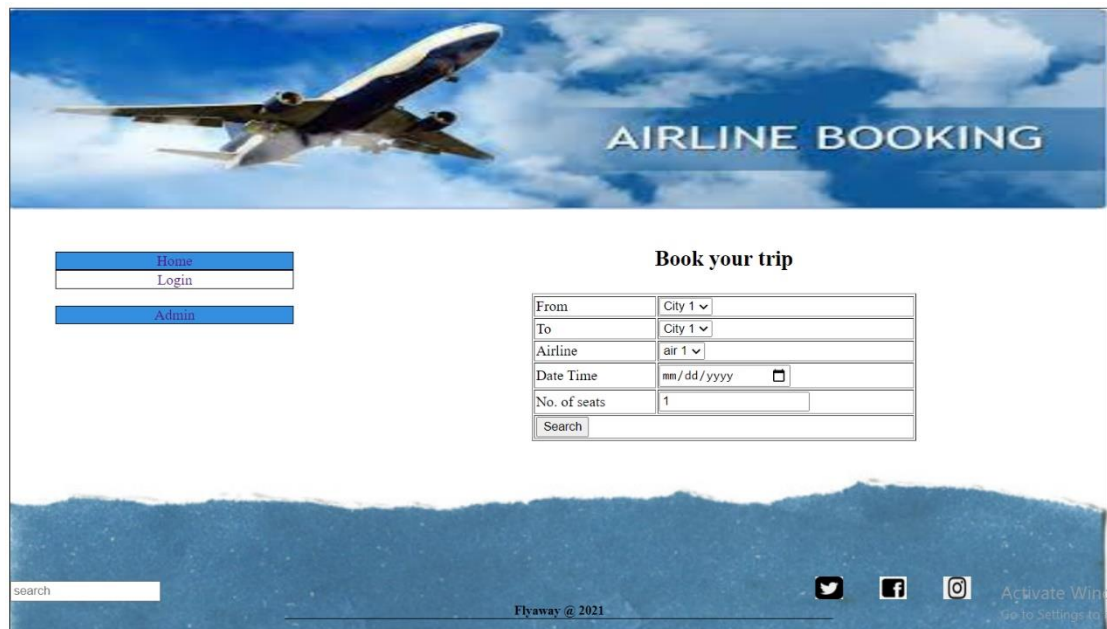
```java
19
20  public class MainServlet extends HttpServlet {
21      LocationDAO locationDB;
22      AireLineDAO airLineDB;
23      FlightDAO flightDB;
24      List<Locations> list;
25      List<AireLine> airList;
26
27      public void init() {
28          locationDB = new LocationDAO();
29          airLineDB = new AireLineDAO();
30          flightDB = new FlightDAO();
31      }
32
33
34      protected void doPost(HttpServletRequest request, HttpServletResponse response)
35              throws ServletException, IOException {
36
37          int fromLoc = Integer.parseInt(request.getParameter("from"));
38          int toLoc = Integer.parseInt(request.getParameter("to"));
39          int airline = Integer.parseInt(request.getParameter("airline"));
40          String flight_date = request.getParameter("date");
41          int no_seats = Integer.parseInt(request.getParameter("no_seats"));
42          boolean availabe_seats = false;
43          Flight flight = null;
44          try {
45              flight = flightDB.findflight(fromLoc, toLoc, airline, flight_date);
46              if(flight != null)
47                  availabe_seats = flightDB.availabeSeats(flight, no_seats);
48
49              list = locationDB.getLocations();
50              airList = airLineDB.getAirlines();
51              request.setAttribute("locations", list);
52              request.setAttribute("airList", airList);
53              request.setAttribute("flight", flight);
54              request.setAttribute("availabe_seats", availabe_seats);
55              request.setAttribute("fromLoc", fromLoc);
56              request.setAttribute("toLoc", toLoc);
57              request.setAttribute("airline", airline);
58              request.setAttribute("no_seats", no_seats);
59              request.setAttribute("flight_date", flight_date);
60              request.setAttribute("submit", true);
61          } catch (SQLException e) {
62              // TODO Auto-generated catch block
```

2.Search result: list of flights/trips founded


```java
32        }
33
34      protected void doPost(HttpServletRequest request, HttpServletResponse response)
35              throws ServletException, IOException {
36
37          HttpSession session = request.getSession();
38          int flight_id = (int)session.getAttribute("flight_id");
39          int seats_no = (int)session.getAttribute("seats_no");
40          int passenger_id = (int)session.getAttribute("id");
41          boolean confirm;
42          try {
43              confirm = flight_db.confirm(passenger_id, flight_id,seats_no);
44              request.setAttribute("confirm", confirm);
45          } catch (SQLException e) {
46              // TODO Auto-generated catch block
47              e.printStackTrace();
48          }
49          RequestDispatcher dispatcher = request.getRequestDispatcher("book.jsp");
50          dispatcher.forward(request, response);
51      }
52
53      protected void doGet(HttpServletRequest request, HttpServletResponse response)
54              throws ServletException, IOException {
55
56          HttpSession session = request.getSession();
57          int seats_no = 0;
58          int flight_id = 0;
59          if(request.getParameter("seats_no") != null) {
60              seats_no = Integer.parseInt(request.getParameter("seats_no"));
61              flight_id = Integer.parseInt(request.getParameter("id"));
62          }else {
63              flight_id = (int)session.getAttribute("flight_id");
64              seats_no = (int)session.getAttribute("seats_no");
65          }
66          Flight flight = null;
67          try {
68              flight = flight_db.getflight(flight_id);
69          } catch (SQLException e1) {
70              // TODO Auto-generated catch block
71              e1.printStackTrace();
72          }
73          session.setAttribute("flight_id", flight_id);
74          session.setAttribute("seats_no", seats_no);
75          int id = -1;
```

3.Login Page: user must log in to complete booking process


```java
 2
 3
 4 ⊕ import java.io.IOException;
21
22 public class LoginServlet extends HttpServlet {
23       PassengerDAO passenger = null;
24 ⊖     public void init() {
25            passenger = new PassengerDAO();
26       }
27
28
29 ⊖     protected void doPost(HttpServletRequest request, HttpServletResponse response)
30             throws ServletException, IOException {
31
32          String email = request.getParameter("email");
33          String pass = request.getParameter("pass");
34
35          boolean valid_user = false;
36          try {
37              if(passenger.login(email, pass))
38              {
39                  HttpSession session=request.getSession();
40                  Passenger p =  passenger.getPassenger(email);
41                  session.setAttribute("id",p.getId());
42                  session.setAttribute("email", p.getemail());
43                  session.setAttribute("name", p.getname());
44
45                  response.sendRedirect("./");
46              }else {
47                  RequestDispatcher dispatcher = request.getRequestDispatcher("login.jsp");
48                  request.setAttribute("valid_user", valid_user);
49                  dispatcher.forward(request, response);
50              }
51          } catch (SQLException | IOException | ServletException e) {
52              // TODO Auto-generated catch block
53              e.printStackTrace();
54          }
55      }
56
57 ⊖     protected void doGet(HttpServletRequest request, HttpServletResponse response)
58             throws ServletException, IOException {
59
60      }
61
```

4.Register Page: If user have no account, user must create an account.


```java
1  package flyaway;
2
3
4  import java.io.IOException;
21
22  public class RegisterServlet extends HttpServlet {
23      PassengerDAO passenger = null;
24      public void init() {
25          passenger = new PassengerDAO();
26      }
27
28
29      protected void doPost(HttpServletRequest request, HttpServletResponse response)
30              throws ServletException, IOException {
31
32          String email = request.getParameter("email");
33          String pass = request.getParameter("pass");
34          String phone = request.getParameter("phone");
35          String name = request.getParameter("name");
36
37          boolean valid_user = false;
38          try {
39              if(passenger.register(name, phone, email, pass))
40              {
41                  HttpSession session=request.getSession();
42
43                  Passenger p =  passenger.getPassenger(email);
44                  session.setAttribute("id",p.getId());
45                  session.setAttribute("email", p.getemail());
46                  session.setAttribute("name", p.getname());
47
48                  RequestDispatcher dispatcher = request.getRequestDispatcher("home.jsp");
49                  dispatcher.forward(request, response);
50              }else {
51                  RequestDispatcher dispatcher = request.getRequestDispatcher("login.jsp");
52                  request.setAttribute("valid_user", valid_user);
53                  dispatcher.forward(request, response);
54              }
55          } catch (SQLException | IOException | ServletException e) {
56              // TODO Auto-generated catch block
57              e.printStackTrace();
58          }
59      }
```

5. Confirmation Page: User must confirm the information of booking.

6.Booking complete confirmation page:

```java
 4⊕ import java.io.IOException;
21
22 public class BookServlet extends HttpServlet {
23     LocationDAO locationDB;
24     AireLineDAO airLineDB;
25     PassengerDAO passenger;
26     FlightDAO flight_db = new FlightDAO();
27
28⊖    public void init() {
29         locationDB = new LocationDAO();
30         airLineDB = new AireLineDAO();
31         passenger= new PassengerDAO();
32     }
33
34⊖    protected void doPost(HttpServletRequest request, HttpServletResponse response)
35             throws ServletException, IOException {
36
37         HttpSession session = request.getSession();
38         int flight_id = (int)session.getAttribute("flight_id");
39         int seats_no = (int)session.getAttribute("seats_no");
40         int passenger_id = (int)session.getAttribute("id");
41         boolean confirm;
42         try {
43             confirm = flight_db.confirm(passenger_id, flight_id,seats_no);
44             request.setAttribute("confirm", confirm);
45         } catch (SQLException e) {
46             // TODO Auto-generated catch block
47             e.printStackTrace();
48         }
49         RequestDispatcher dispatcher = request.getRequestDispatcher("book.jsp");
50         dispatcher.forward(request, response);
51     }
52
53⊖    protected void doGet(HttpServletRequest request, HttpServletResponse response)
54             throws ServletException, IOException {
55
56         HttpSession session = request.getSession();
57         int seats_no = 0;
58         int flight_id = 0;
59         if(request.getParameter("seats_no") != null) {
60             seats_no = Integer.parseInt(request.getParameter("seats_no"));
61             flight_id = Integer.parseInt(request.getParameter("id"));
62         }else {
```

7.Booking list page: user can view his list of booking with information of each booking.

My Booking List

Passenger Name	Book date	Flight Number	From	To	Airline	No. of persons
Farah	2021-07-28	1	City 1	City 2	air 1	6
Farah	2021-07-29	1	City 1	City 2	air 1	4
Farah	2021-07-30	1	City 1	City 2	air 1	1

BookListServlet.java

```java
23
24 public class BookListServlet extends HttpServlet {
25     LocationDAO locationDB;
26     AireLineDAO airLineDB;
27     PassengerDAO passenger;
28     FlightDAO flight_db = new FlightDAO();
29
30     public void init() {
31         locationDB = new LocationDAO();
32         airLineDB = new AireLineDAO();
33         passenger= new PassengerDAO();
34     }
35
36     protected void doPost(HttpServletRequest request, HttpServletResponse response)
37             throws ServletException, IOException {
38
39         HttpSession session = request.getSession();
40         RequestDispatcher dispatcher = request.getRequestDispatcher("my_books.jsp");
41         dispatcher.forward(request, response);
42     }
43
44     protected void doGet(HttpServletRequest request, HttpServletResponse response)
45             throws ServletException, IOException {
46
47         HttpSession session = request.getSession();
48         int id = -1;
49         if(session.getAttribute("id") != null) {
50             id= (int)session.getAttribute("id");
51         }
52
53         if(id==-1){
54             response.sendRedirect(request.getContextPath() + "/login.jsp");
55         }else {
56             RequestDispatcher dispatcher = request.getRequestDispatcher("my_books.jsp");
57             String email= (String)session.getAttribute("email");
58             try {
59                 Passenger p =  passenger.getPassenger(email);
60                 request.setAttribute("passenger", p);
61             } catch (SQLException e) {
62                 // TODO Auto-generated catch block
63                 e.printStackTrace();
64             }
65             List<Book> listBooks= new ArrayList<Book>();
66             try {
```

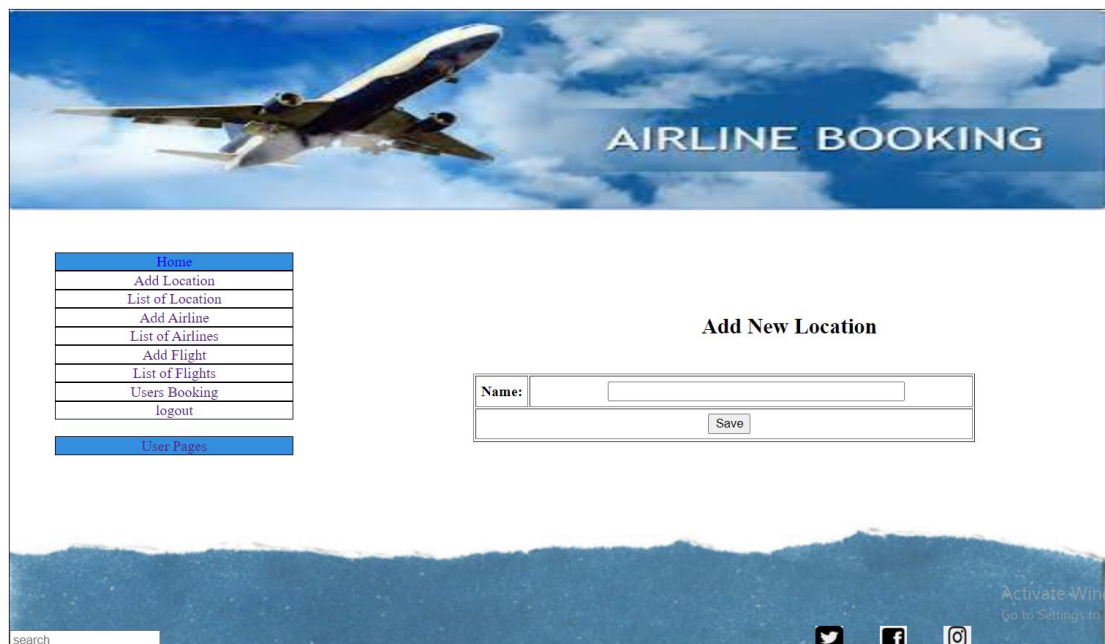8.Admin Login Page:


```java
AdminLoginServlet.java ⊠
1       package flyaway.admin.pages;
2⊕ import java.io.IOException;
14
15 public class AdminLoginServlet extends HttpServlet {
16     AdminDAO admin = null;
17⊖    public void init() {
18         admin = new AdminDAO();
19     }
20
21
22⊖    protected void doPost(HttpServletRequest request, HttpServletResponse response)
23             throws ServletException, IOException {
24
25         String email = request.getParameter("email");
26         String pass = request.getParameter("pass");
27
28         boolean valid_user = false;
29         try {
30             if(admin.adminLogin(email, pass))
31             {
32                 HttpSession session=request.getSession();
33                 session.setAttribute("adminLogin", true);
34
35                 response.sendRedirect("./");
36             }else {
37                 RequestDispatcher dispatcher = request.getRequestDispatcher("admin_login.jsp");
38                 request.setAttribute("valid_user", valid_user);
39                 dispatcher.forward(request, response);
40             }
41         } catch (SQLException | IOException | ServletException e) {
42             // TODO Auto-generated catch block
43             e.printStackTrace();
44         }
45     }
46
47⊖    protected void doGet(HttpServletRequest request, HttpServletResponse response)
48             throws ServletException, IOException {
49         RequestDispatcher dispatcher = request.getRequestDispatcher("admin_login.jsp");
50         dispatcher.forward(request, response);
51
52     }
53
54 }
```
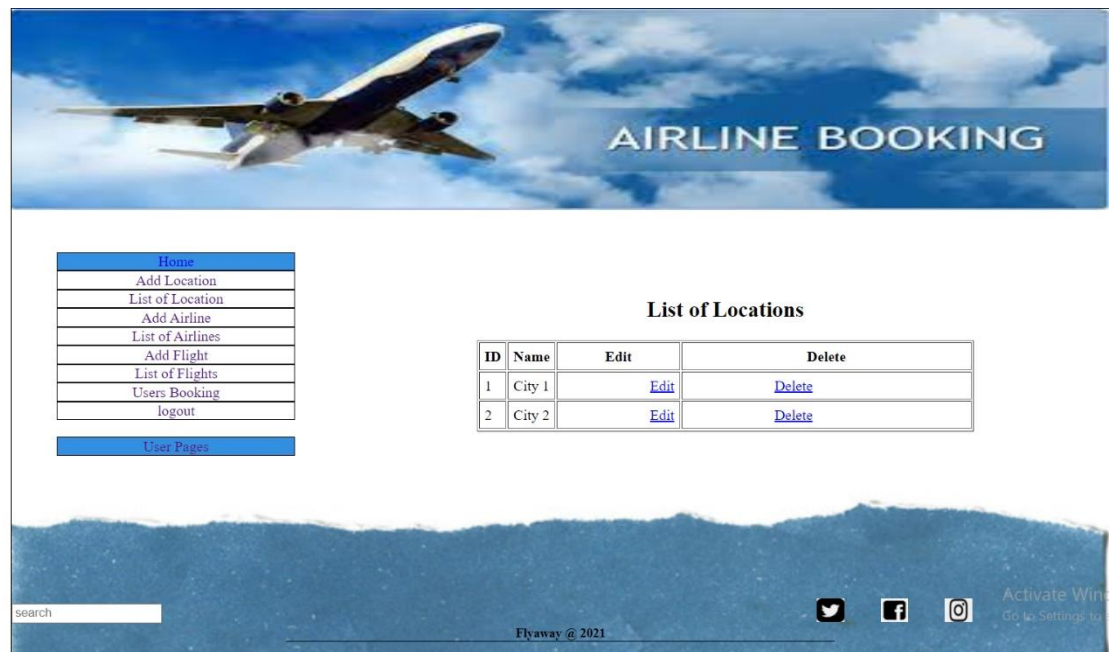
9.Admin Home page :.
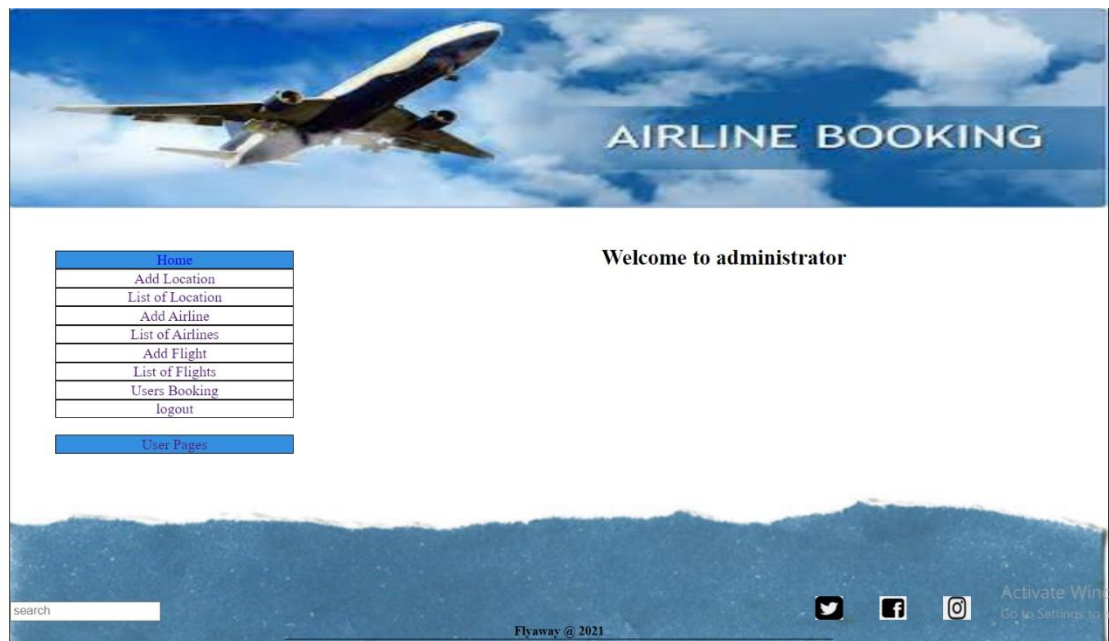

```java
AdminLoginServlet.java    AdminMainServlet.java ⊠
  1      package flyaway.admin.pages;
  2
  3
  4⊕ import java.io.IOException;□
 20
 21 public class AdminMainServlet extends HttpServlet {
 22      LocationDAO locationDB;
 23      AireLineDAO airLineDB;
 24      FlightDAO flightDB;
 25      List<Locations> list;
 26      List<AireLine> airList;
 27
 28      public void init() {
 29          locationDB = new LocationDAO();
 30          airLineDB = new AireLineDAO();
 31          flightDB = new FlightDAO();
 32      }
 33
 34
 35      protected void doPost(HttpServletRequest request, HttpServletResponse response)
 36              throws ServletException, IOException {
 37
 38
 39
 40          RequestDispatcher dispatcher = request.getRequestDispatcher("admin_home.jsp");
 41          dispatcher.forward(request, response);
 42      }
 43
 44      protected void doGet(HttpServletRequest request, HttpServletResponse response)
 45              throws ServletException, IOException {
 46
 47          HttpSession session = request.getSession();
 48          if(session.getAttribute("adminLogin") == null) {
 49              response.sendRedirect(request.getContextPath() + "/admin/login");
 50          }else {
 51              RequestDispatcher dispatcher = request.getRequestDispatcher("admin_home.jsp");
 52              dispatcher.forward(request, response);
 53          }
 54      }
 55
 56 }
 57
 58
```
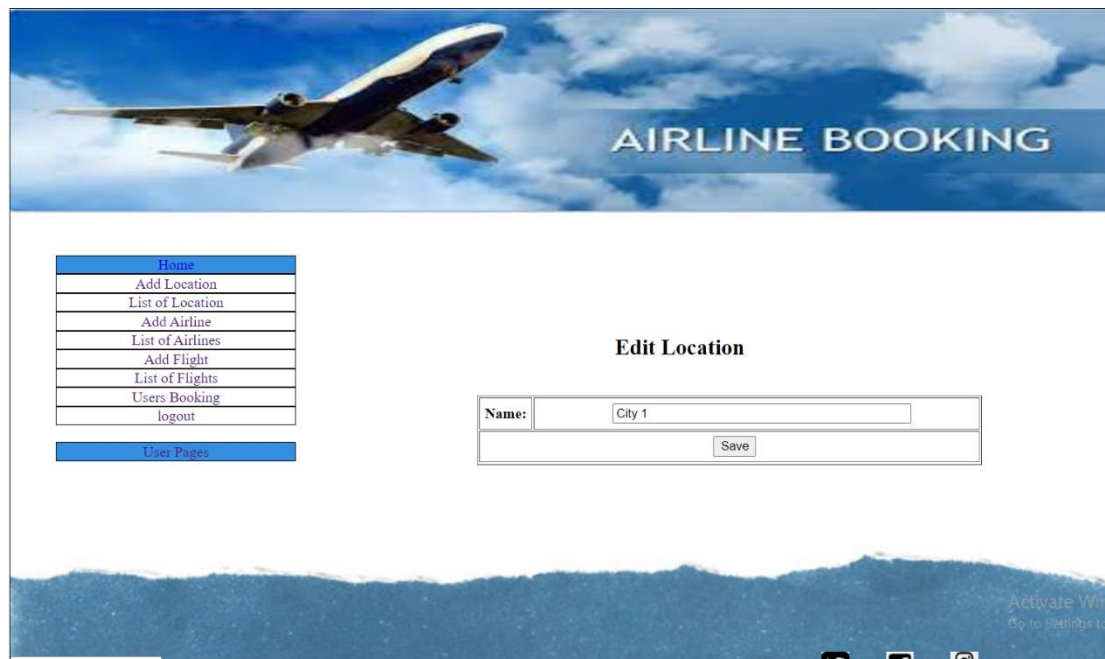
10. Add new location page


```java
public class locationServlet extends HttpServlet {
    LocationDAO loc = null;
    public void init() {
        loc = new LocationDAO();
    }


    protected void doPost(HttpServletRequest request, HttpServletResponse response)
            throws ServletException, IOException {

        String name = request.getParameter("name");
        boolean error = false;
        try {
            if(loc.add(name))
            {
                HttpSession session=request.getSession();
                session.setAttribute("adminLogin", true);

                response.sendRedirect("./listlocations");
            }else {
                RequestDispatcher dispatcher = request.getRequestDispatcher("locationForm.jsp");
                request.setAttribute("error", error);
                dispatcher.forward(request, response);
            }
        } catch (SQLException | IOException | ServletException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }
    }

    protected void doGet(HttpServletRequest request, HttpServletResponse response)
            throws ServletException, IOException {

        HttpSession session = request.getSession();
        boolean adminLogin = false;
        if(session.getAttribute("adminLogin") != null) {
            adminLogin=  (boolean)session.getAttribute("adminLogin");
        }

        if(!adminLogin){
            response.sendRedirect(request.getContextPath() + "/admin/login");
        }else {
            RequestDispatcher dispatcher = request.getRequestDispatcher("locationForm.jsp");
```
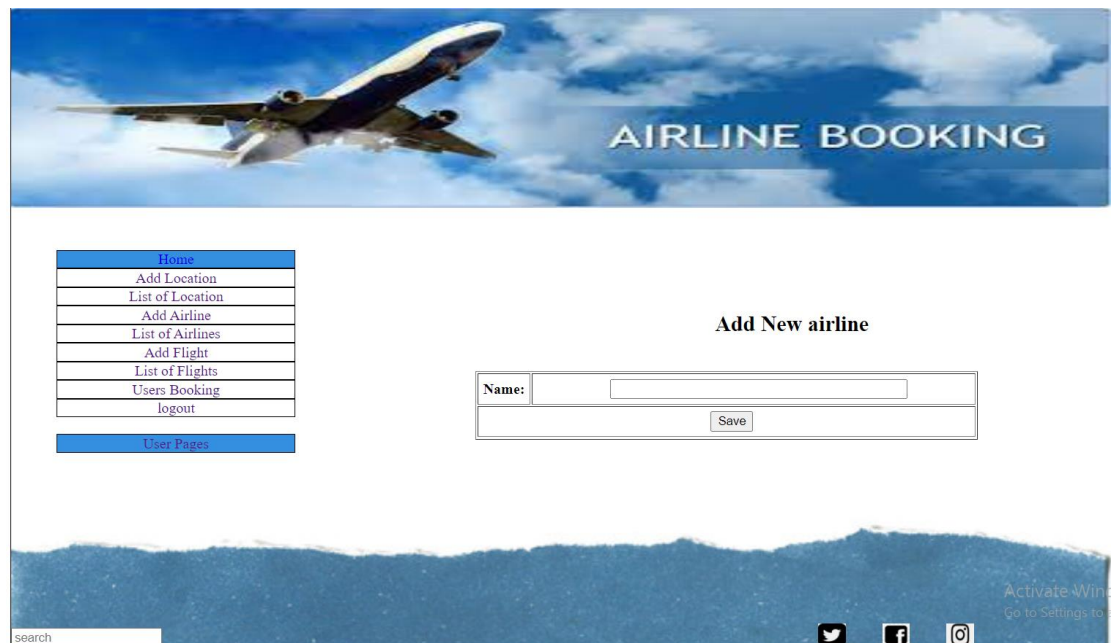
11. List of saved locations


```java
package flyaway.admin.pages;


import java.io.IOException;

public class locationListServlet extends HttpServlet {
    LocationDAO loc = null;
    public void init() {
        loc = new LocationDAO();
    }


    protected void doPost(HttpServletRequest request, HttpServletResponse response)
            throws ServletException, IOException {

    }

    protected void doGet(HttpServletRequest request, HttpServletResponse response)
            throws ServletException, IOException {

        HttpSession session = request.getSession();
        boolean adminLogin = false;
        if(session.getAttribute("adminLogin") != null) {
            adminLogin=  (boolean)session.getAttribute("adminLogin");
        }

        if(!adminLogin){
            response.sendRedirect(request.getContextPath() + "/admin/login");
        }else {
            RequestDispatcher dispatcher = request.getRequestDispatcher("locationList.jsp");
            List<Locations> list;
            try {
                list = loc.getLocations();
                request.setAttribute("listLocations", list);
                dispatcher.forward(request, response);
            } catch (SQLException e) {
                // TODO Auto-generated catch block
                e.printStackTrace();
            }
        }
    }
}
```

12. Edit Location Name

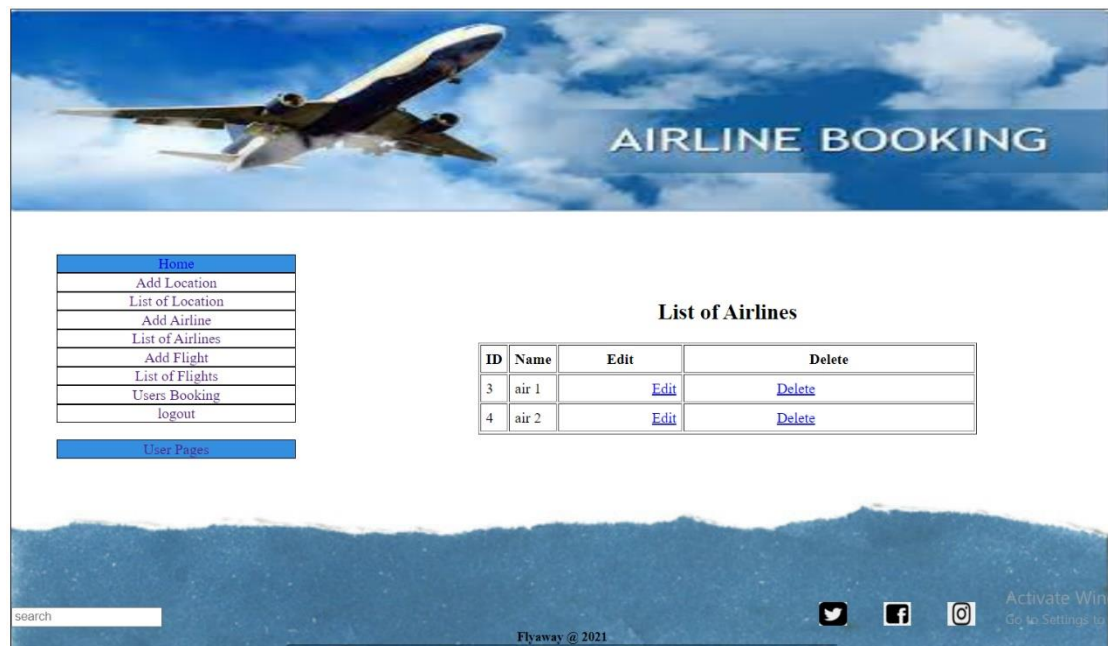```java
1   package flyaway.admin.pages;
2
3
4   import java.io.IOException;
18
19   public class updateLocationServlet extends HttpServlet {
20   LocationDAO loc = null;
21   public void init() {
22   loc = new LocationDAO();
23   }
24
25
26   protected void doPost(HttpServletRequest request, HttpServletResponse response)
27   throws ServletException, IOException {
28
29   String name = request.getParameter("name");
30   String id = request.getParameter("id");
31   boolean error = false;
32   try {
33   if(loc.update(id, name))
34   {
35
36   response.sendRedirect("./listlocations");
37   }else {
38   RequestDispatcher dispatcher = request.getRequestDispatcher("locationForm.jsp");
39   request.setAttribute("error", error);
40   dispatcher.forward(request, response);
41   }
42   } catch (SQLException | IOException | ServletException e) {
43   // TODO Auto-generated catch block
44   e.printStackTrace();
45   }
46   }
47
48   protected void doGet(HttpServletRequest request, HttpServletResponse response)
49   throws ServletException, IOException {
50
51   HttpSession session = request.getSession();
52   boolean adminLogin = false;
53   if(session.getAttribute("adminLogin") != null) {
54   adminLogin= (boolean)session.getAttribute("adminLogin");
55   }
56
57   if(!adminLogin){
```

## 13. Add new airline page



```java
package flyaway.admin.pages;

import java.io.IOException;

public class airlineServlet extends HttpServlet {
    AireLineDAO loc = null;
    public void init() {
        loc = new AireLineDAO();
    }


    protected void doPost(HttpServletRequest request, HttpServletResponse response)
            throws ServletException, IOException {

        String name = request.getParameter("name");
        boolean error = false;
        try {
            if(loc.add(name))
            {
                response.sendRedirect("./listairlines");
            }else {
                RequestDispatcher dispatcher = request.getRequestDispatcher("airlineForm.jsp");
                request.setAttribute("error", error);
                dispatcher.forward(request, response);
            }
        } catch (SQLException | IOException | ServletException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }
    }

    protected void doGet(HttpServletRequest request, HttpServletResponse response)
            throws ServletException, IOException {

        HttpSession session = request.getSession();
        boolean adminLogin = false;
        if(session.getAttribute("adminLogin") != null) {
            adminLogin= (boolean)session.getAttribute("adminLogin");
        }

        if(!adminLogin){
            response.sendRedirect(request.getContextPath() + "/admin/login");
        }else {
```

13. List of saved airlines


```java
1    package flyaway.admin.pages;
2
3
4⊕ import java.io.IOException;
20
21 public class airlineListServlet extends HttpServlet {
22     AireLineDAO a = null;
23⊖    public void init() {
24         a = new AireLineDAO ();
25     }
26
27
28⊖    protected void doPost(HttpServletRequest request, HttpServletResponse response)
29             throws ServletException, IOException {
30
31     }
32
33⊖    protected void doGet(HttpServletRequest request, HttpServletResponse response)
34             throws ServletException, IOException {
35
36         HttpSession session = request.getSession();
37         boolean adminLogin = false;
38         if(session.getAttribute("adminLogin") != null) {
39             adminLogin=  (boolean)session.getAttribute("adminLogin");
40         }
41
42         if(!adminLogin){
43             response.sendRedirect(request.getContextPath() + "/admin/login");
44         }else {
45             RequestDispatcher dispatcher = request.getRequestDispatcher("airlineList.jsp");
46             List<AireLine> list;
47             try {
48                 list = a.getAirlines();
49                 request.setAttribute("listAirlines", list);
50                 dispatcher.forward(request, response);
51             } catch (SQLException e) {
52                 // TODO Auto-generated catch block
53                 e.printStackTrace();
54             }
55         }
56     }
57
58 }
```
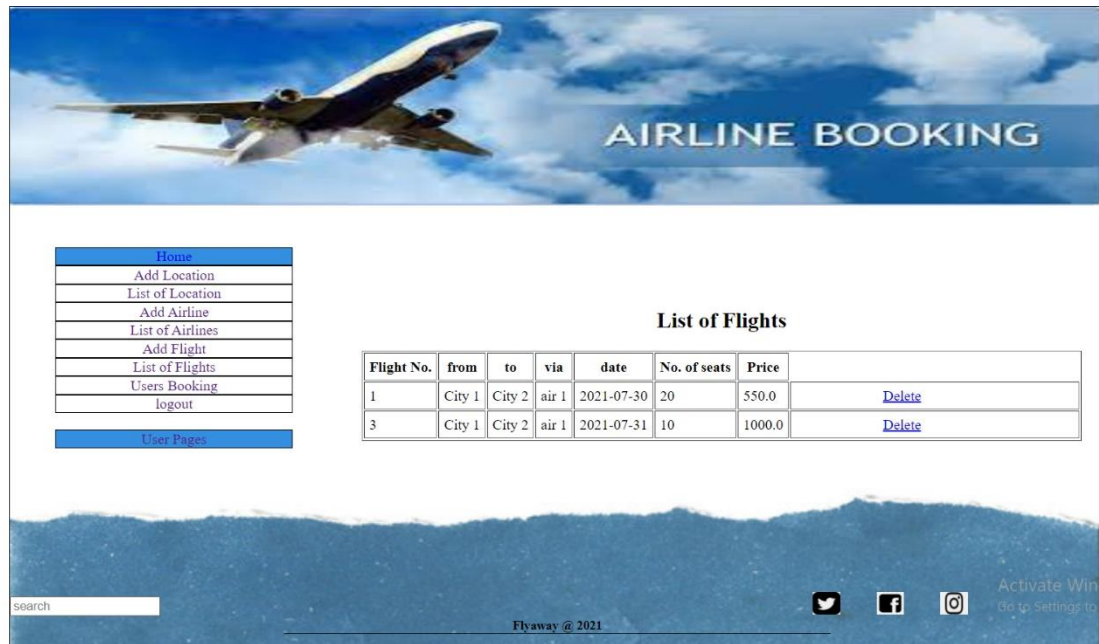
14. Add new flight page:


```java
flightServlet.java ⊠   flightListServlet.java
23  public class flightServlet extends HttpServlet {
24      LocationDAO locationDB;
25      AireLineDAO airLineDB;
26      FlightDAO flightDB;
27      List<Locations> list;
28      List<AireLine> airList;
29
30      public void init() {
31          locationDB = new LocationDAO();
32          airLineDB = new AireLineDAO();
33          flightDB = new FlightDAO();
34      }
35
36      protected void doPost(HttpServletRequest request, HttpServletResponse response)
37              throws ServletException, IOException {
38
39          String fromLoc = request.getParameter("from");
40          String toLoc = request.getParameter("to");
41          String airline = request.getParameter("airline");
42          String flight_date = request.getParameter("date");
43          String no_seats = request.getParameter("no_seats");
44          String price = request.getParameter("price");
45
46          try {
47              Flight f = new Flight(fromLoc, toLoc, Integer.parseInt(airline) , Integer.parseInt(no_seats), flight_date, price);
48              if(flightDB.add(f))
49              {
50                  response.sendRedirect("./listflights");
51              }else {
52                  RequestDispatcher dispatcher = request.getRequestDispatcher("flightForm.jsp");
53                  dispatcher.forward(request, response);
54              }
55          } catch (SQLException | IOException | ServletException e) {
56              // TODO Auto-generated catch block
57              e.printStackTrace();
58          }
59      }
60
61      protected void doGet(HttpServletRequest request, HttpServletResponse response)
62              throws ServletException, IOException {
63
64          HttpSession session = request.getSession();
65          boolean adminLogin = false;
66          if(session.getAttribute("adminLogin") != null) {
```
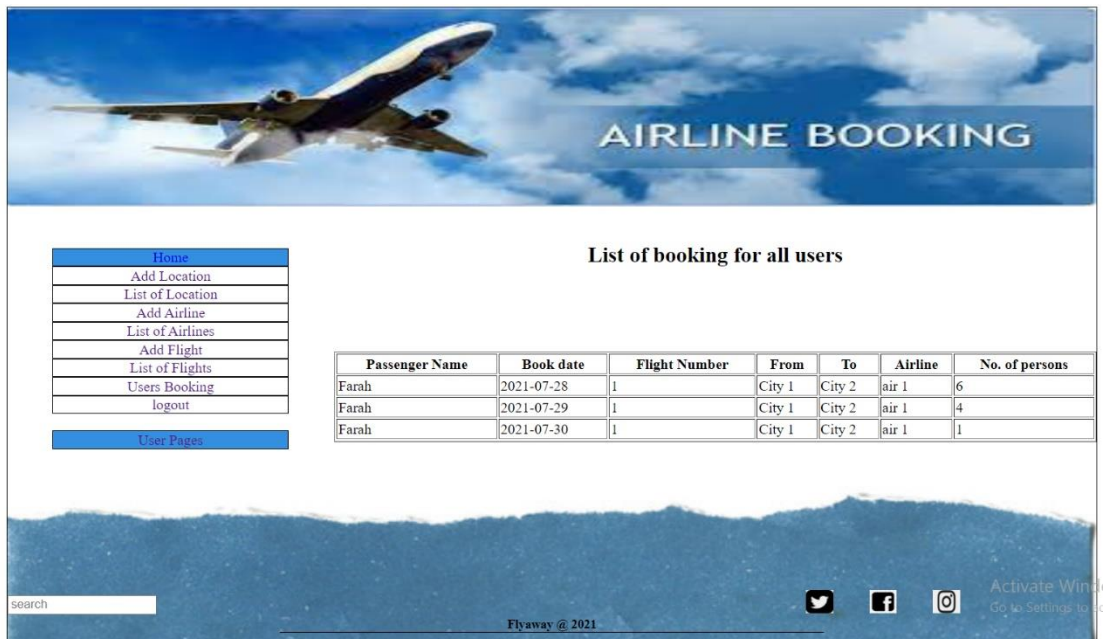
15. list of all flights


```java
package flyaway.admin.pages;

import java.io.IOException;

public class flightListServlet extends HttpServlet {
    FlightDAO fdb = null;
    public void init() {
        fdb = new FlightDAO ();
    }


    protected void doPost(HttpServletRequest request, HttpServletResponse response)
            throws ServletException, IOException {

    }

    protected void doGet(HttpServletRequest request, HttpServletResponse response)
            throws ServletException, IOException {

        HttpSession session = request.getSession();
        boolean adminLogin = false;
        if(session.getAttribute("adminLogin") != null) {
            adminLogin=  (boolean)session.getAttribute("adminLogin");
        }

        if(!adminLogin){
            response.sendRedirect(request.getContextPath() + "/admin/login");
        }else {
            RequestDispatcher dispatcher = request.getRequestDispatcher("flightList.jsp");
            List<Flight> list;
            try {
                list = fdb.getflight();
                request.setAttribute("listFlights", list);
                dispatcher.forward(request, response);
            } catch (SQLException e) {
                // TODO Auto-generated catch block
                e.printStackTrace();
            }
        }
    }
}
```

16. List of all booking done by users

List of booking for all users

Passenger Name	Book date	Flight Number	From	To	Airline	No. of persons
Farah	2021-07-28	1	City 1	City 2	air 1	6
Farah	2021-07-29	1	City 1	City 2	air 1	4
Farah	2021-07-30	1	City 1	City 2	air 1	1

Flyaway @ 2021

```jsp
users_books.jsp
1  <%@ page language="java" contentType="text/html; charset=UTF-8"  pageEncoding="UTF-8"%>
2  <%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>
3
4  <html>
5      <head>
6          <style>
7              <%@ include file='style.css' %>
8          </style>
9      </head>
10
11      <body>
12          <div class="body">
13              <div class="header">
14                  <img src="https://i.ibb.co/QkRMC5g/banner.jpg"/>
15              </div>
16
17
18              <div class="mainContent">
19                  <div class="leftSide">
20                      <ul>
21                          <%@ include file='admin_menu.html' %>
22                      </ul>
23
24                  </div>
25                  <div class="main">
26
27                      <c:if test="${listBooks != null}">
28
29                          <h2>List of booking for all users</h2>
30                              <table border="1" width="100%">
31                                  <tr>
32                                      <th>Passenger Name</th>
33                                      <th>Book date</th>
34                                      <th>Flight Number</th>
35                                      <th>From</th>
36                                      <th>To</th>
37                                      <th>Airline</th>
38                                      <th>No. of persons</th>
39                                  </tr>
40                              <c:forEach var="book" items="${listBooks}">
41                                  <tr>
42                                      <td><c:out value="${book.passenger.name}" /></td>
43                                      <td><c:out value="${book.bookdate}" /></td>
44                                      <td><c:out value="${book.flight_id}" /></td>
```