

Internet Technology Lab Report – Assignment 7

TITLE

Name – Sourav Dutta

Roll – 001610501076

Class – BCSE 4th year

Group – A3

Assignment Numbers – 7

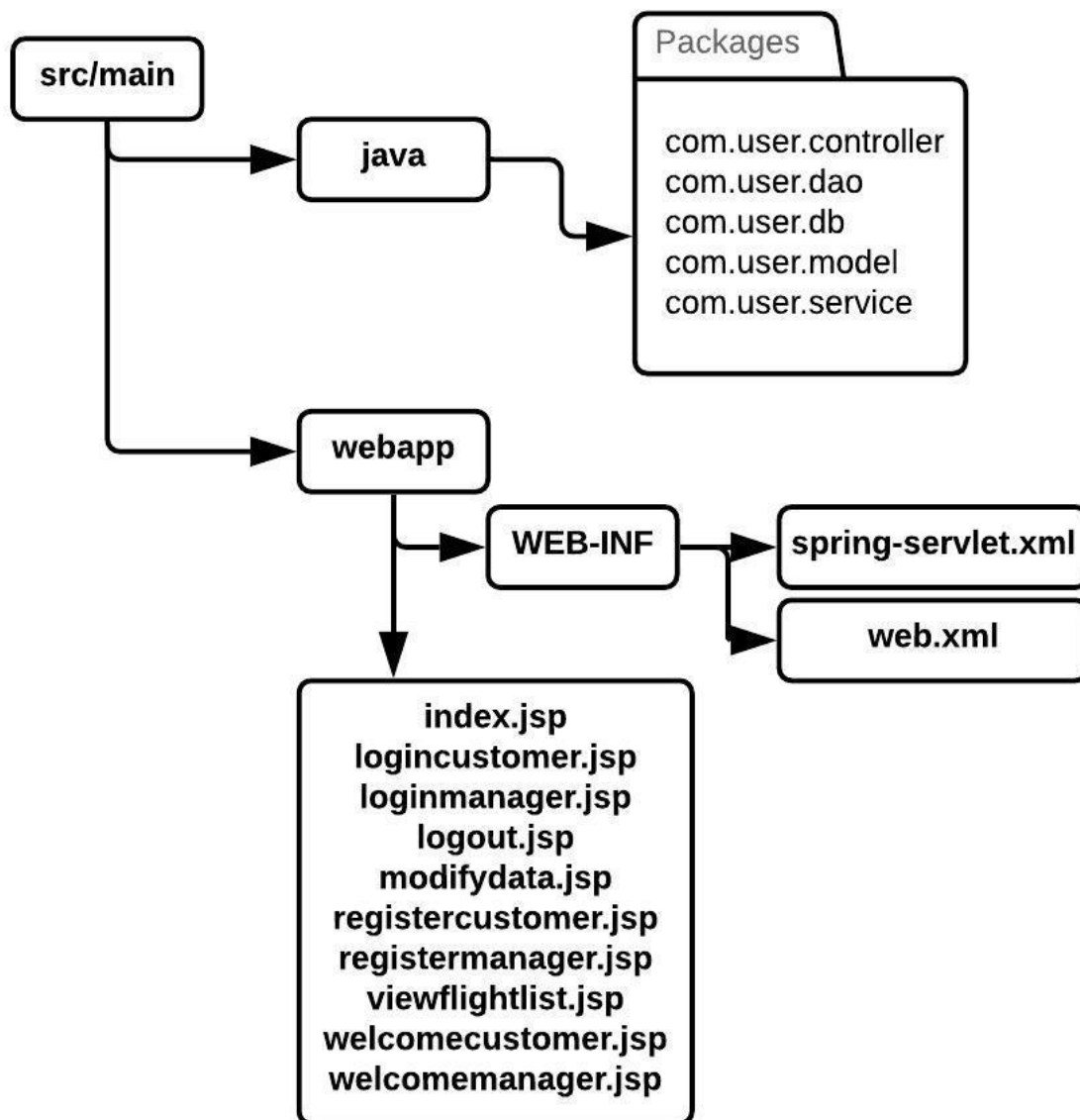
Implement a web application for “Travel Thru Air” based on Spring MVC framework to support any of the following two use cases:

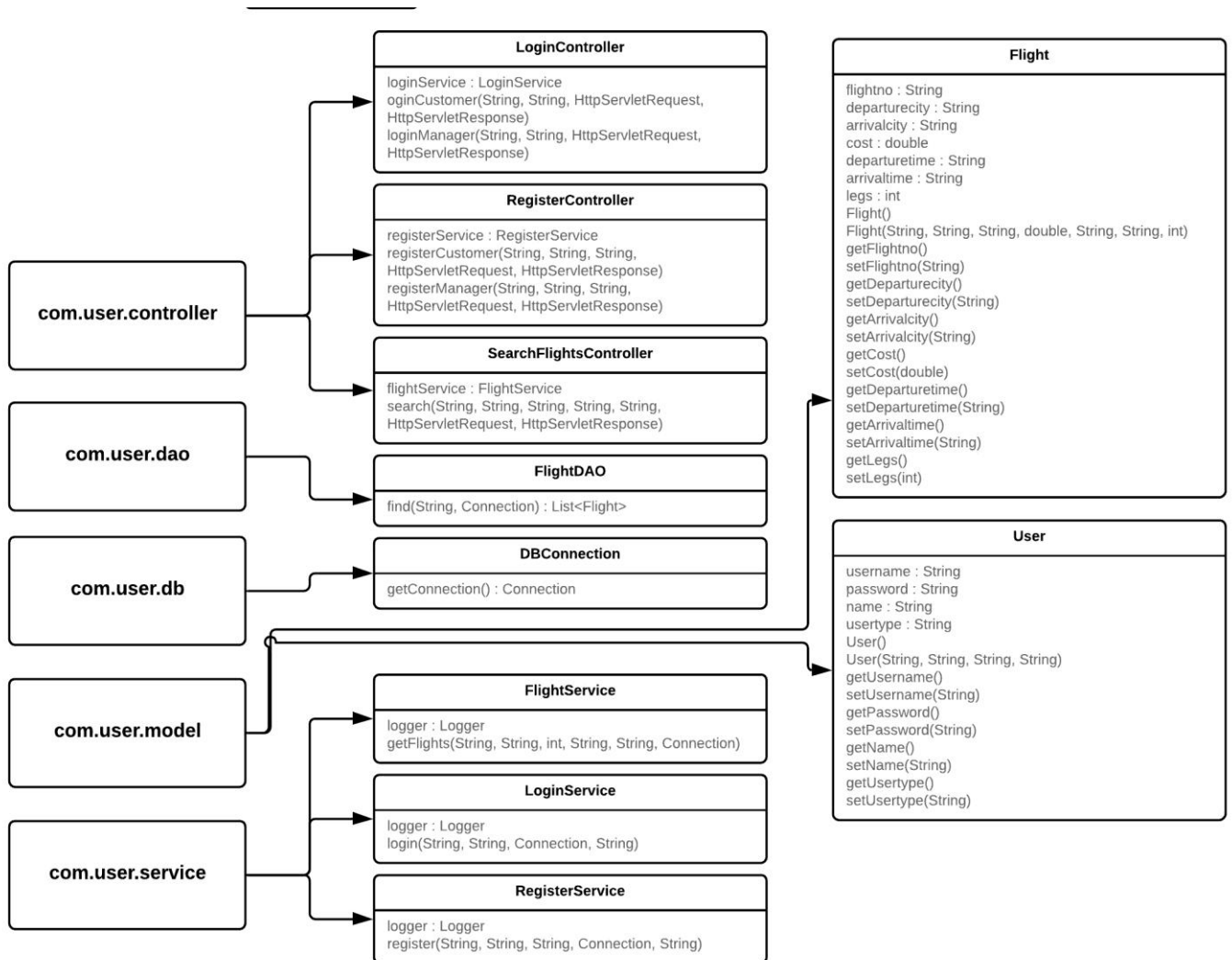
1. A list of current special deals must appear on the home page. Each special deal must display the departure city, the arrival city, and the cost. These special deals are set up by the marketing department and change during the day, so it can't be static. Special deals are only good for a limited amount of time.
2. A user may search for flights, given a departure city, time and an arrival city. The results must display the departure city, the arrival city, the total cost, and how many legs the flight will have.

Provide a login controller for user login facility. The solution should reflect each layer of the spring framework.

Generate trace using log4j.

CLASS DIAGRAMS





CODE SNIPPET

Deployment Descriptor [web.xml] :

```

<!DOCTYPE web-app PUBLIC
"-//Sun Microsystems, Inc.//DTD Web Application 2.3//EN"
"http://java.sun.com/dtd/web-app_2_3.dtd" >

<web-app>
  <display-name>Archetype Created Web Application</display-name>

  <servlet>
    <servlet-name>spring</servlet-name>
    <servlet-class>org.springframework.web.servlet.DispatcherServlet</servlet-class>
  </servlet>

  <servlet-mapping>
    <servlet-name>spring</servlet-name>
    <url-pattern>/</url-pattern>
  </servlet-mapping>
</web-app>
  
```

spring-servlet.xml :

```
<?xml version="1.0" encoding="UTF-8"?>

<beans xmlns="http://www.springframework.org/schema/beans"
       xmlns:ctx="http://www.springframework.org/schema/context"
       xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
       xmlns:mvc="http://www.springframework.org/schema/mvc"
       xsi:schemaLocation="http://www.springframework.org/schema/beans
http://www.springframework.org/schema/beans/spring-beans-4.0.xsd
http://www.springframework.org/schema/mvc
http://www.springframework.org/schema/mvc/spring-mvc-3.0.xsd
http://www.springframework.org/schema/context
http://www.springframework.org/schema/context/spring-context-4.0.xsd">

    <ctx:annotation-config></ctx:annotation-config>
    <ctx:component-scan base-package="com.user"></ctx:component-scan>

</beans>
```

pom.xml :

```
<project xmlns="http://maven.apache.org/POM/4.0.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/maven-
v4_0_0.xsd">
  <modelVersion>4.0.0</modelVersion>
  <groupId>com.sourav</groupId>
  <artifactId>ass7_travel_thru_air_spring</artifactId>
  <packaging>war</packaging>
  <version>0.0.1-SNAPSHOT</version>
  <name>ass7_travel_thru_air_spring Maven Webapp</name>
  <url>http://maven.apache.org</url>
  <dependencies>
    <dependency>
      <groupId>junit</groupId>
      <artifactId>junit</artifactId>
      <version>3.8.1</version>
      <scope>test</scope>
    </dependency>

    <!-- https://mvnrepository.com/artifact/org.springframework/spring-webmvc -->
    <dependency>
      <groupId>org.springframework</groupId>
      <artifactId>spring-webmvc</artifactId>
      <version>5.0.5.RELEASE</version>
    </dependency>

    <!-- https://mvnrepository.com/artifact/mysql/mysql-connector-java -->
    <dependency>
      <groupId>mysql</groupId>
      <artifactId>mysql-connector-java</artifactId>
      <version>8.0.11</version>
    </dependency>

    <!-- https://mvnrepository.com/artifact/org.apache.httpcomponents/httpclient -->
    <dependency>
      <groupId>org.apache.httpcomponents</groupId>
      <artifactId>httpclient</artifactId>
      <version>4.3.3</version>
    </dependency>
  </dependencies>
```

```

<!-- https://mvnrepository.com/artifact/javax.servlet/servlet-api -->
<dependency>
  <groupId>javax.servlet</groupId>
  <artifactId>servlet-api</artifactId>
  <version>3.0-alpha-1</version>
  <scope>provided</scope>
</dependency>
<dependency>
  <groupId>log4j</groupId>
  <artifactId>log4j</artifactId>
  <version>1.2.17</version>
</dependency>
</dependencies>
<build>
  <finalName>ass7_travel_thru_air_spring</finalName>
</build>
</project>

```

index.jsp :

```

<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"
    import="com.user.model.*" %>
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Travel Thru Air</title>
</head>
<body>
<%
User user = (User) session.getAttribute("user");
if(user != null) {
    if(user.getUserType() == "customer") response.sendRedirect("welcomecustomer.jsp");
    else if(user.getUserType() == "manager")
response.sendRedirect("welcomemanager.jsp");
    return;
}
%>
<h1>Welcome to Travel Thru Air</h1>
<h3>
Are you from management department?
<a
href="http://localhost:8080/ass7_travel_thru_air_spring/registermanager.jsp">Management
Portal</a>
</h3>
<br> <br>
<h3>
Are you a customer?
<a
href="http://localhost:8080/ass7_travel_thru_air_spring/registercustomer.jsp">Customer
Portal</a>
</h3>

</body>
</html>

```

logincustomer.jsp :

```

<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"
    import="com.user.model.*" %>

```

```

<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Customer Portal</title>
</head>
<body>
<%
User user = (User) session.getAttribute("user");
if(user != null) {
    if(user.getUserType() == "customer") response.sendRedirect("welcomecustomer.jsp");
    else if(user.getUserType() == "manager")
response.sendRedirect("welcomemanager.jsp");
    return;
}
%>
<h1>Welcome to Travel Thru Air</h1>
<h2>Customer Portal</h2>
Want to go back to registration page? <a
href="http://localhost:8080/ass7_travel_thru_air_spring/registercustomer.jsp">Customer
Registration</a>
<br>
<h3>Login here!</h3>
<form action="logincustomer" method="post" >
<pre>
    Username: <input type="text" name="username" placeholder="Enter username here"
required/>

    Password: <input type="text" name="password" placeholder="Enter password here"
required/>

        <input type="submit" value="LOGIN"/>

</pre>
</form>

</body>
</html>

```

loginmanager.jsp :

```

<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"
    import="com.user.model.*" %>
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Management Portal</title>
</head>
<body>
<%
User user = (User) session.getAttribute("user");
if(user != null) {
    if(user.getUserType() == "customer") response.sendRedirect("welcomecustomer.jsp");
    else if(user.getUserType() == "manager")
response.sendRedirect("welcomemanager.jsp");
    return;
}
%>
<h1>Welcome to Travel Thru Air</h1>
<h2>Management Portal</h2>

```

```

Want to go back to registration page? <a
href="http://localhost:8080/ass7_travel_thru_air_spring/registermanager.jsp">Management
Registration</a>
<br>
<h3>Login here!</h3>
<form action="loginmanager" method="post" >
<pre>
    Username: <input type="text" name="username" placeholder="Enter username here"
required/>

    Password: <input type="text" name="password" placeholder="Enter password here"
required/>

        <input type="submit" value="LOGIN"/>
</pre>
</form>

</body>
</html>

```

logout.jsp :

```

<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Logout</title>
</head>
<body>
<%
session.removeAttribute("user");
response.sendRedirect("index.jsp");
%>

</body>
</html>

```

registercustomer.jsp :

```

<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"
    import="com.user.model.*" %>
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Customer Portal</title>
</head>
<body>
<%
User user = (User) session.getAttribute("user");
if(user != null) {
    if(user.getUserType() == "customer") response.sendRedirect("welcomecustomer.jsp");
    else if(user.getUserType() == "manager")
response.sendRedirect("welcomemanager.jsp");
    return;
}
%>
<h1>Welcome to Travel Thru Air</h1>
<h2>Customer Portal</h2>

```

```

Already registered? <a
href="http://localhost:8080/ass7_travel_thru_air_spring/logincustomer.jsp">Customer
Login</a>
<br>
<h3>Register here!</h3>
<form action="registercustomer" >
<pre>
    Username: <input type="text" name="username" placeholder="Enter username here"
required/>

    Password: <input type="text" name="password" placeholder="Enter password here"
required/>

    Name:      <input type="text" name="name" placeholder="Enter name here" required/>

               <input type="submit" value="REGISTER"/>
</pre>
</form>
</body>
</html>

```

registermanager.jsp :

```

<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"
    import="com.user.model.*" %>
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Management Portal</title>
</head>
<body>
<%
User user = (User) session.getAttribute("user");
if(user != null) {
    if(user.getUserType() == "customer") response.sendRedirect("welcomecustomer.jsp");
    else if(user.getUserType() == "manager")
response.sendRedirect("welcomemanager.jsp");
    return;
}
%>
<h1>Welcome to Travel Thru Air</h1>
<h2>Management Portal</h2>

```

```

Already registered? <a
href="http://localhost:8080/ass7_travel_thru_air_spring/loginmanager.jsp">Management
Login</a>
<br>
<h3>Register here!</h3>
<form action="registermanager" method="post" >
<pre>
    Username: <input type="text" name="username" placeholder="Enter username here"
required/>

    Password: <input type="text" name="password" placeholder="Enter password here"
required/>

    Name:      <input type="text" name="name" placeholder="Enter name here" required/>

               <input type="submit" value="REGISTER"/>

```



```

</pre>
</form>
</body>
</html>

```

welcomecustomer.jsp :

```

<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1" import="com.user.model.*"
    import="com.user.dao.*" import="java.sql.*" import="java.util.*" %>
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Welcome Customer</title>

</head>
<body>
<%
User user = (User)session.getAttribute("user");
if(user == null) {
    response.sendRedirect("index.jsp");
    return;
}
out.println("<h2>Hey "+user.getUsertype()+"! Welcome "+user.getName()+"</h2>");
%>
Want to logout?
<a href="http://localhost:8080/ass7_travel_thru_air_spring/logout.jsp">Logout</a>
<br> <br>

<h3>Search flights</h3>
<form name = "searchform" action="searchflights" method="post" onsubmit="return
validate()">
<pre>
        Departure City: <select name="departurecity" >
                                <option
value="Bangalore">Bangalore</option>
                                <option value="Delhi">Delhi</option>
                                <option value="Kolkata">Kolkata</option>
                                <option value="Mumbai">Mumbai</option>
                                </select>
        Arrival City: <select name="arrivalcity" >
                                <option
value="Bangalore">Bangalore</option>
                                <option value="Delhi">Delhi</option>
                                <option value="Kolkata">Kolkata</option>
                                <option value="Mumbai">Mumbai</option>
                                </select>
        Number of persons: <input type="number" name="persons" min="1" max="20"
required>
        Expected Departure Time (Start) : <select name="starttime" >
                                <option value="0000">0000 HRS</option>
                                <option value="0100">0100 HRS</option>
                                <option value="0200">0200 HRS</option>
                                <option value="0300">0300 HRS</option>
                                <option value="0400">0400 HRS</option>
                                <option value="0500">0500 HRS</option>
                                <option value="0600">0600 HRS</option>
                                <option value="0700">0700 HRS</option>
                                <option value="0800">0800 HRS</option>
                                <option value="0900">0900 HRS</option>
                                <option value="1000">1000 HRS</option>

```

```

        <option value="1100">1100 HRS</option>
        <option value="1200">1200 HRS</option>
        <option value="1300">1300 HRS</option>
        <option value="1400">1400 HRS</option>
        <option value="1500">1500 HRS</option>
        <option value="1600">1600 HRS</option>
        <option value="1700">1700 HRS</option>
        <option value="1800">1800 HRS</option>
        <option value="1900">1900 HRS</option>
        <option value="2000">2000 HRS</option>
        <option value="2100">2100 HRS</option>
        <option value="2200">2200 HRS</option>
        <option value="2300">2300 HRS</option>
        <option value="2359">2359 HRS</option>
    </select>
    Expected Departure Time (End) : <select name="endtime" >
        <option value="0000">0000 HRS</option>
        <option value="0100">0100 HRS</option>
        <option value="0200">0200 HRS</option>
        <option value="0300">0300 HRS</option>
        <option value="0400">0400 HRS</option>
        <option value="0500">0500 HRS</option>
        <option value="0600">0600 HRS</option>
        <option value="0700">0700 HRS</option>
        <option value="0800">0800 HRS</option>
        <option value="0900">0900 HRS</option>
        <option value="1000">1000 HRS</option>
        <option value="1100">1100 HRS</option>
        <option value="1200">1200 HRS</option>
        <option value="1300">1300 HRS</option>
        <option value="1400">1400 HRS</option>
        <option value="1500">1500 HRS</option>
        <option value="1600">1600 HRS</option>
        <option value="1700">1700 HRS</option>
        <option value="1800">1800 HRS</option>
        <option value="1900">1900 HRS</option>
        <option value="2000">2000 HRS</option>
        <option value="2100">2100 HRS</option>
        <option value="2200">2200 HRS</option>
        <option value="2300">2300 HRS</option>
        <option value="2359">2359 HRS</option>
    </select>

    <input type="submit" value="SEARCH">

</pre>
</form>

<script>
function validate() {
    var departurecity = document.forms["searchform"]["departurecity"].value;
    var arrivalcity = document.forms["searchform"]["arrivalcity"].value;
    var starttime = document.forms["searchform"]["starttime"].value;
    var endtime = document.forms["searchform"]["endtime"].value;
    if(starttime > endtime) {
        alert("Start time cannot be greater than end time!");
        return false;
    }
    if(departurecity == arrivalcity) {
        alert("Departure city and arrival city cannot be same!");
        return false;
    }
}


```

```

    }
    return true;
}
</script>
</body>
</html>

```

welcomemanager.jsp :

```

<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1" import="com.user.model.*" import="java.sql.*, java.util.*,
    com.user.dao.*"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Welcome Manager</title>
</head>
<body>
<%
User user = (User)session.getAttribute("user");
if(user == null) {
    response.sendRedirect("index.jsp");
    return;
}
out.println("<h2>Hey "+user.getUsertype()+"! Welcome "+user.getName()+"</h2>");
%>
Want to logout?
<a href="http://localhost:8080/ass7_travel_thru_air_spring/logout.jsp">Logout</a>
<br> <br>
<h3>Current Flights</h3>
<table border="1">
    <tbody>
        <%
            Connection conn = (Connection)
request.getSession().getAttribute("DBConnection");
FlightDAO flightDAO = new FlightDAO();
String query = "select * from flights order by departurecity,
arrivalcity, departuretime";
List<Flight> flights = flightDAO.find(query, conn);
boolean flag = false;
%>
        <%
            for(Flight f : flights) {
                if(flag == false) {
                    <%
                        <tr>
                            <th>FLIGHT NUMBER</th>
                            <th>DEPARTURE CITY</th>
                            <th>ARRIVAL CITY</th>
                            <th>COST</th>
                            <th>DEPARTURE TIME</th>
                            <th>ARRIVAL TIME</th>
                            <th>LEGS</th>
                        </tr>
                    <%
                        }
                        flag = true;
                    <%
                        <tr>
                            <th><%=f.getFlightno() %></th>
                            <th><%=f.getDeparturecity() %></th>

```

```

        <th><%=f.getArrivalcity() %></th>
        <th><%= "Rs. " +f.getCost() %></th>
        <th><%=f.getDeparturetime()+" HRS"%></th>
        <th><%=f.getArrivaltime()+" HRS" %></th>
        <th><%=f.getLegs() %></th>
    </tr>
    <%
        }
        if(flag == false) out.println("No flights to display currently");
    %>
</tbody>
</table>
<br>
Want to add, delete or update flights? [Table name is "flights"]
<form action="modifydata.jsp" method="post">
<pre>
    <input type="text" name="query" placeholder="Enter SQL query here"
    style="height:30px;width:600px" required>

    <input type="submit" value="SUBMIT SQL QUERY" />
</pre>
</form>
</body>
</html>

```

modifydata.jsp :

```

<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"
    import="com.user.model.*, java.sql.*"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Modify Data</title>
</head>
<body>
<%
User user = (User)session.getAttribute("user");
if(user == null) {
    response.sendRedirect("index.jsp");
    return;
}
out.println("<h2>Hey "+user.getUsertype()+"! Welcome "+user.getName()+"</h2>");
%>
Want to logout?
<a href="http://localhost:8080/ass7_travel_thru_air_spring/logout.jsp">Logout</a>
<br>
Want to go back to Manager Home Page?
<a href="http://localhost:8080/ass7_travel_thru_air_spring/welcomemanager.jsp">Home
Page</a>
<br> <br>
<%
String query = (String) request.getParameter("query");
out.println("SQL query you entered: " + query + "<br><br>");
try {
    Connection conn = (Connection) request.getSession().getAttribute("DBConnection");
    PreparedStatement statement = conn.prepareStatement(query);
    statement.execute();
    out.println("<b>SQL query executed!</b>");
} catch (Exception e) {
    out.println("<b>Invalid SQL query!</b>");
}

```

```

}

%>
</body>
</html>

```

viewflightlist.jsp :

```

<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"
    import="java.util.*"
    import="java.sql.*"
    import="com.user.model.*"
    import="com.user.dao.*" %>
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Flight list</title>
</head>
<body>
    <%
        User user = (User)session.getAttribute("user");
        if(user == null) {
            response.sendRedirect("index.jsp");
            return;
        }
        out.println("<h2>Hey "+user.getUsertype()+"! Welcome "+user.getName()+"</h2>");
    %>
    Want to logout?
    <a href="http://localhost:8080/ass7_travel_thru_air_spring/logout.jsp">Logout</a>
    <br>
    Want to search flights again?
    <a
href="http://localhost:8080/ass7_travel_thru_air_spring/welcomecustomer.jsp">Profile
Page</a>
    <br> <br>
    <h3>Flight list</h3>
    <%
        int persons = Integer.parseInt((String) request.getAttribute("persons"));
        List<Flight> flights = (List<Flight>) request.getAttribute("flights");
        boolean flag = false;
    %>
    <table border="1">
        <tbody>
            <%
                int i = 0;
                for(Flight f : flights) {
                    i++;
                    if(flag == false) {
                        %>
                        <tr>
                            <th><b>CHOOSE</b></th>
                            <th><b>FLIGHT NUMBER</b></th>
                            <th><b>DEPARTURE CITY</b></th>
                            <th><b>ARRIVAL CITY</b></th>
                            <th><b>TOTAL COST</b></th>
                            <th><b>DEPARTURE TIME</b></th>
                            <th><b>ARRIVAL TIME</b></th>
                            <th><b>LEGS</b></th>
                            <th><b>PERSONS</b></th>
                        </tr>
                    <%

```

```

        }
        flag = true;
    }
    %>
    <tr>
        <th><input type="radio" name="index" value="<%=i%>"><%=i
    %></input></th>
        <th><%=f.getFlightno() %></th>
        <th><%=f.getDeparturecity() %></th>
        <th><%=f.getArrivalcity() %></th>
        <th><%= "Rs. " + (f.getCost()*persons) %></th>
        <th><%=f.getDeparturetime()+" HRS"%></th>
        <th><%=f.getArrivaltime()+" HRS" %></th>
        <th><%=f.getLegs() %></th>
        <th><%=persons %></th>
    </tr>
    <%
    }
    if(flag == false) out.println("No flights to display
currently");
    %>
    </tbody>
</table>
<button id="buyButton">BUY</button>
<br><br>
<div id="log"></div>
</body>

<script>
    var log = document.getElementById('log');
    document.getElementById("buyButton").onclick = function() {
        var index = document.getElementsByName('index');
        var flag = 0;
        log.innerHTML = "";
        for(var i=0;i<index.length;i++) {
            if(index[i].checked) {
                flag = 1;
                var idx = parseInt(index[i].value);
                log.innerHTML += "You bought ticket of Flight "+idx;
                break;
            }
        }
        if(flag == 0) log.innerHTML += "You did not choose a flight to buy!";
    };
</script>
</html>

```

LoginController.java :

```

package com.user.service;

import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;

import org.springframework.stereotype.Service;

import com.user.model.User;

@Service
public class LoginService {

```

```

        public User login(String username, String password, Connection conn, String type)
    {
        User user = null;
        try {
            String query;
            if(type == "customer") query = "select * from customers where
username='" + username + "'";
            else query = "select * from managers where username='" + username +
            "'";

            PreparedStatement statement = conn.prepareStatement(query);
            ResultSet resultSet = statement.executeQuery();
            if (resultSet.next()) {
                if (resultSet.getString("password").equals(password)) {
                    user = new User(username, password,
resultSet.getString("name"), type);
                    System.out.println("info : " + type + " " + username +
                    " login successful");
                    return user;
                } else {
                    System.out.println("error: " + type + " " + username +
                    " login failed : password mismatch");
                    return null;
                }
            } else {
                System.out.println("error: " + type + " " + username + " login
failed : username not found");
                return null;
            }
        } catch (Exception e) {
            e.printStackTrace();
        }
        System.out.println("error: " + type + " " + username + " login failed");
        return null;
    }
}

```

RegisterController.java :

```

package com.user.controller;

import java.sql.Connection;

import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Controller;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RequestParam;
import org.springframework.web.servlet.ModelAndView;

import com.user.db.DBConnection;
import com.user.service.RegisterService;

@Controller
public class RegisterController {

    @Autowired
    RegisterService registerService;

    /*
     * Registering a customer

```

```

        */
        @RequestMapping("registercustomer")
        public ModelAndView registerCustomer(@RequestParam("username") String username,
        @RequestParam("password") String password
        , @RequestParam("name") String name, HttpServletRequest req,
        HttpServletResponse res) {
            ModelAndView mv = new ModelAndView();
            Connection conn = null;
            if(req.getSession().getAttribute("DBConnection") == null) {
                conn = DBConnection.getConnection();
                req.getSession().setAttribute("DBConnection", conn);
            } else {
                conn = (Connection) req.getSession().getAttribute("DBConnection");
            }
            if(registerService.register(username, password, name, conn, "customer")) {
                mv.setViewName("logincustomer.jsp");
            } else {
                mv.setViewName("registercustomer.jsp");
            }
            return mv;
        }

        /*
        * Registering a manager
        */
        @RequestMapping("registermanager")
        public ModelAndView registerManager(@RequestParam("username") String username,
        @RequestParam("password") String password
        , @RequestParam("name") String name, HttpServletRequest req,
        HttpServletResponse res) {
            ModelAndView mv = new ModelAndView();
            Connection conn = null;
            if(req.getSession().getAttribute("DBConnection") == null) {
                conn = DBConnection.getConnection();
                req.getSession().setAttribute("DBConnection", conn);
            } else {
                conn = (Connection) req.getSession().getAttribute("DBConnection");
            }
            if(registerService.register(username, password, name, conn, "manager")) {
                mv.setViewName("loginmanager.jsp");
            } else {
                mv.setViewName("registermanager.jsp");
            }
            return mv;
        }
    }
}

```

SearchFlightsController.java :

```
package com.user.controller;
```

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

```
import java.util.List;
```

```
import javax.servlet.http.HttpServletRequest;
```

```
import javax.servlet.http.HttpServletResponse;
```

```
import org.springframework.beans.factory.annotation.Autowired;
```

```
import org.springframework.stereotype.Controller;
```

```
import org.springframework.web.bind.annotation.RequestMapping;
```

```
import org.springframework.web.bind.annotation.RequestParam;
```

```
import org.springframework.web.servlet.ModelAndView;
```



```

import com.user.model.Flight;
import com.user.service.FlightService;

@Controller
public class SearchFlightsController {

    @Autowired
    FlightService flightService;

    @RequestMapping("searchflights")
    public ModelAndView search(@RequestParam("departurecity") String departurecity,
                              @RequestParam("arrivalcity") String arrivalcity,
                              @RequestParam("persons") String persons,
                              @RequestParam("starttime") String starttime,
                              @RequestParam("endtime") String endtime,
                              HttpServletRequest req, HttpServletResponse res) {

        ModelAndView mv = new ModelAndView();
        Connection conn =
(Connection)req.getSession().getAttribute("DBConnection");
        List<Flight> flights = flightService.getFlights(departurecity, arrivalcity,
Integer.parseInt(persons), starttime, endtime, conn);
        mv.addObject("flights", flights);
        mv.addObject("persons", persons);
        mv.setViewName("viewflightlist.jsp");
        return mv;
    }
}

```

FlightDAO.java :

```

package com.user.dao;
import java.sql.*;
import java.util.*;
import com.user.model.*;

public class FlightDAO {
    public List<Flight> find(String query, Connection conn) {
        List<Flight> flightlist = new ArrayList<Flight>();
        try {
            PreparedStatement statement = conn.prepareStatement(query);

            ResultSet resultset = statement.executeQuery();
            while(resultset.next()) {
                Flight flight = new Flight();
                flight.setFlightno(resultset.getString("flightno"));
                flight.setDeparturecity(resultset.getString("departurecity"));
                flight.setArrivalcity(resultset.getString("arrivalcity"));
                flight.setCost(resultset.getDouble("cost"));
                flight.setDeparturetime(resultset.getString("departuretime"));
                flight.setArrivaltime(resultset.getString("arrivaltime"));
                flight.setLegs(resultset.getInt("legs"));
                flightlist.add(flight);
            }
        } catch (Exception e) {
            e.printStackTrace();
        }
        return flightlist;
    }
}

```

DBConnection.java :

```
package com.user.db;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;

public class DBConnection {

    public static Connection getConnection() {
        Connection conn = null;
        try {
            Class.forName("com.mysql.jdbc.Driver");
            conn =
DriverManager.getConnection("jdbc:mysql://localhost:3306/travelldb", "root", "root");
        } catch (ClassNotFoundException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        } catch (SQLException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }
        return conn;
    }

}
```

FlightService.java :

```
package com.user.service;

import java.sql.Connection;
import java.util.ArrayList;
import java.util.List;

import org.apache.log4j.Logger;
import org.springframework.stereotype.Service;

import com.user.dao.FlightDAO;
import com.user.model.Flight;

@Service
public class FlightService {

    final static Logger logger = Logger.getLogger(FlightService.class);

    public List<Flight> getFlights(String departurecity, String arrivalcity, int
persons, String starttime, String endtime, Connection conn) {
        logger.info("Getting the flight details");
        List<Flight> flights = new ArrayList<Flight>();
        FlightDAO flightDAO = new FlightDAO();
        String query = "select * from flights where departurecity='" +
departurecity + "' and arrivalcity='" + arrivalcity + "' order by departurertime";
        List<Flight> tempflights = flightDAO.find(query, conn);
        for(Flight f : tempflights) {
            if(Integer.parseInt(f.getDeparturertime()) <
Integer.parseInt(starttime)) continue;
            if(Integer.parseInt(f.getDeparturertime()) >
Integer.parseInt(endtime)) continue;
            flights.add(f);
        }
    }

}
```

```

        return flights;
    }
}

```

LoginService.java :

```

package com.user.service;

import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;

import org.apache.log4j.Logger;
import org.springframework.stereotype.Service;

import com.user.model.User;

@Service
public class LoginService {

    final static Logger Logger = Logger.getLogger(LoginService.class);

    public User login(String username, String password, Connection conn, String type)
    {
        User user = null;
        try {
            String query;
            if(type == "customer") query = "select * from customers where
username='" + username + "'";
            else query = "select * from managers where username='" + username +
            "'";

            PreparedStatement statement = conn.prepareStatement(query);
            ResultSet resultset = statement.executeQuery();
            if (resultset.next()) {
                if (resultset.getString("password").equals(password)) {
                    user = new User(username, password,
resultset.getString("name"), type);
                    Logger.info(type + " " + username + " login
successful");
                    return user;
                } else {
                    Logger.error(type + " " + username + " login failed :
password mismatch");
                    return null;
                }
            } else {
                Logger.error(type + " " + username + " login failed : username
not found");
                return null;
            }
        } catch (Exception e) {
            e.printStackTrace();
        }
        Logger.error(type + " " + username + " login failed");
        return null;
    }
}

```

RegisterService.java :

```

package com.user.service;

```

```

import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;

import org.apache.log4j.Logger;
import org.springframework.stereotype.Service;

@Service
public class RegisterService {

    final static Logger logger = Logger.getLogger(RegisterService.class);

    public boolean register(String username, String password, String name, Connection
conn, String type) {
        try {
            String query;
            if(type == "customer") query = "select * from customers where
username='" + username + "'";
            else query = "select * from managers where username='" + username
+ "'";

            PreparedStatement statement = conn.prepareStatement(query);
            ResultSet resultset = statement.executeQuery();
            if (resultset.next()) {
                logger.warn(type + " " + username + " already exists");
                return false;
            } else {
                if(type == "customer") query = "insert into customers
values('" + username + "', '" + password + "', '" + name + "')";
                else query = "insert into managers
values('" + username + "', '" + password + "', '" + name + "')";
                statement = conn.prepareStatement(query);
                statement.execute();
                logger.info(type + " " + username + " registered
successfully");
                return true;
            }
        }
        catch(Exception e) {
            e.printStackTrace();
        }
        logger.error(type + " " + username + "registration failed");
        return false;
    }
}

```

log4j.properties :

```

# Root logger option
log4j.rootlogger=INFO, file

# Direct log messages to a log file
log4j.appender.file=org.apache.log4j.RollingFileAppender

#Redirect to a folder
log4j.appender.file.File=C:\\log\\logging.log
log4j.appender.file.MaxFileSize=10MB
log4j.appender.file.MaxBackupIndex=10
log4j.appender.file.layout=org.apache.log4j.PatternLayout
log4j.appender.file.layout.ConversionPattern=%d{yyyy-MM-dd HH:mm:ss} %-5p %c{1}:%L -
%m%n

```

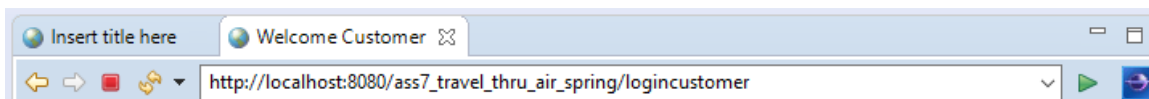
OUTPUT

The image displays three sequential browser screenshots of a web application titled "Travel Thru Air".

First Screenshot: The browser address bar shows `http://localhost:8080/ass5_travel_thru_air/`. The page content includes a heading "Welcome to Travel Thru Air", a question "Are you from management department?" with a link to "Management Portal", and another question "Are you a customer?" with a link to "Customer Portal".

Second Screenshot: The browser address bar shows `http://localhost:8080/ass7_travel_thru_air_spring/registercustomer.jsp`. The page content includes the heading "Welcome to Travel Thru Air", a sub-heading "Customer Portal", a link "Already registered? Customer Login", and a section "Register here!". Below this is a registration form with fields for "Username" (containing "sourav"), "Password" (containing "1234"), and "Name" (containing "Sourav Dutta"). A "REGISTER" button is at the bottom.

Third Screenshot: The browser address bar shows `http://localhost:8080/ass7_travel_thru_air_spring/logincustomer`. The page content includes the heading "Welcome to Travel Thru Air", a sub-heading "Customer Portal", a link "Want to go back to registration page? Customer Registration", and a section "Login here!". Below this is a login form with fields for "Username" (containing "sourav") and "Password" (containing "1234"). A "LOGIN" button is at the bottom.



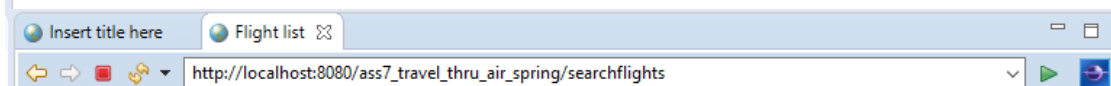
Hey customer! Welcome Sourav Dutta

Want to logout? [Logout](#)

Search flights

Departure City:
Arrival City:
Number of persons:
Expected Departure Time (Start) :
Expected Departure Time (End) :

SEARCH



Hey customer! Welcome Sourav Dutta

Want to logout? [Logout](#)

Want to search flights again? [Profile Page](#)

Flight list

CHOOSE	FLIGHT NUMBER	DEPARTURE CITY	ARRIVAL CITY	TOTAL COST	DEPARTURE TIME	ARRIVAL TIME	LEGS	PERSONS
<input type="radio"/> 1	5WR	Delhi	Mumbai	Rs. 59990.0	1100 HRS	1500 HRS	2	10
<input checked="" type="radio"/> 2	E32	Delhi	Mumbai	Rs. 65000.0	1700 HRS	2245 HRS	4	10

BUY

You bought ticket of Flight 2

Management Portal

http://localhost:8080/ass7_travel_thru_air_spring/loginmanager.jsp

Welcome to Travel Thru Air

Management Portal

Want to go back to registration page? [Management Registration](#)

Login here!

Username:

Password:

Welcome Manager

http://localhost:8080/ass7_travel_thru_air_spring/loginmanager

Log file generated using log4j:

```

2019-12-03 10:00:29,205 INFO [org.springframework.web.servlet.DispatcherServlet] - FrameworkServlet
'spring': initialization started
2019-12-03 10:00:29,380 INFO [org.springframework.web.context.support.XmlWebApplicationContext] -
Refreshing WebApplicationContext for namespace 'spring-servlet': startup date [Tue Dec 03 10:00:29 IST
2019]; root of context hierarchy
2019-12-03 10:00:29,517 INFO [org.springframework.beans.factory.xml.XmlBeanDefinitionReader] - Loading
XML bean definitions from ServletContext resource [/WEB-INF/spring-servlet.xml]
2019-12-03 10:00:30,726 INFO
[org.springframework.web.servlet.mvc.method.annotation.RequestMappingHandlerMapping] - Mapped
"/logincustomer" onto public org.springframework.web.servlet.ModelAndView
com.user.controller.LoginController.loginCustomer(java.lang.String,java.lang.String,javax.servlet.http.HttpSe
rvletRequest,javax.servlet.http.HttpServletResponse)
2019-12-03 10:00:30,730 INFO
[org.springframework.web.servlet.mvc.method.annotation.RequestMappingHandlerMapping] - Mapped
"/loginmanager" onto public org.springframework.web.servlet.ModelAndView
com.user.controller.LoginController.loginManager(java.lang.String,java.lang.String,javax.servlet.http.HttpSer
vletRequest,javax.servlet.http.HttpServletResponse)

```

2019-12-03 10:00:30,731 INFO

[org.springframework.web.servlet.mvc.method.annotation.RequestMappingHandlerMapping] - Mapped "{[/registercustomer]}" onto public org.springframework.web.servlet.ModelAndView com.user.controller.RegisterController.registerCustomer(java.lang.String,java.lang.String,java.lang.String,javax.servlet.http.HttpServletRequest,javax.servlet.http.HttpServletResponse)

2019-12-03 10:00:30,732 INFO

[org.springframework.web.servlet.mvc.method.annotation.RequestMappingHandlerMapping] - Mapped "{[/registermanager]}" onto public org.springframework.web.servlet.ModelAndView com.user.controller.RegisterController.registerManager(java.lang.String,java.lang.String,java.lang.String,javax.servlet.http.HttpServletRequest,javax.servlet.http.HttpServletResponse)

2019-12-03 10:00:30,733 INFO

[org.springframework.web.servlet.mvc.method.annotation.RequestMappingHandlerMapping] - Mapped "{[/searchflights]}" onto public org.springframework.web.servlet.ModelAndView com.user.controller.SearchFlightsController.search(java.lang.String,java.lang.String,java.lang.String,java.lang.String,java.lang.String,javax.servlet.http.HttpServletRequest,javax.servlet.http.HttpServletResponse)

2019-12-03 10:00:30,871 INFO

[org.springframework.web.servlet.mvc.method.annotation.RequestMappingHandlerAdapter] - Looking for @ControllerAdvice: WebApplicationContext for namespace 'spring-servlet': startup date [Tue Dec 03 10:00:29 IST 2019]; root of context hierarchy

2019-12-03 10:00:31,138 INFO [org.springframework.web.servlet.DispatcherServlet] - FrameworkServlet 'spring': initialization completed in 1932 ms

2019-12-03 10:00:31,668 WARN [com.user.service.RegisterService] - customer sourav already exists

2019-12-03 10:00:38,035 ERROR [com.user.service.LoginService] - customer soruab login failed : username not found

2019-12-03 10:00:46,673 INFO [com.user.service.LoginService] - customer sourav login successful

2019-12-03 12:29:58,730 INFO [org.springframework.web.servlet.DispatcherServlet] - FrameworkServlet 'spring': initialization started

2019-12-03 12:29:58,897 INFO [org.springframework.web.context.support.XmlWebApplicationContext] - Refreshing WebApplicationContext for namespace 'spring-servlet': startup date [Tue Dec 03 12:29:58 IST 2019]; root of context hierarchy

2019-12-03 12:29:59,020 INFO [org.springframework.beans.factory.xml.XmlBeanDefinitionReader] - Loading XML bean definitions from ServletContext resource [/WEB-INF/spring-servlet.xml]

2019-12-03 12:30:00,145 INFO

[org.springframework.web.servlet.mvc.method.annotation.RequestMappingHandlerMapping] - Mapped "{[/logincustomer]}" onto public org.springframework.web.servlet.ModelAndView com.user.controller.LoginController.loginCustomer(java.lang.String,java.lang.String,javax.servlet.http.HttpServletRequest,javax.servlet.http.HttpServletResponse)

2019-12-03 12:30:00,148 INFO

[org.springframework.web.servlet.mvc.method.annotation.RequestMappingHandlerMapping] - Mapped "{[/loginmanager]}" onto public org.springframework.web.servlet.ModelAndView com.user.controller.LoginController.loginManager(java.lang.String,java.lang.String,javax.servlet.http.HttpServletRequest,javax.servlet.http.HttpServletResponse)

2019-12-03 12:30:00,150 INFO

[org.springframework.web.servlet.mvc.method.annotation.RequestMappingHandlerMapping] - Mapped "{[/registermanager]}" onto public org.springframework.web.servlet.ModelAndView com.user.controller.RegisterController.registerManager(java.lang.String,java.lang.String,java.lang.String,javax.servlet.http.HttpServletRequest,javax.servlet.http.HttpServletResponse)

2019-12-03 12:30:00,150 INFO

[org.springframework.web.servlet.mvc.method.annotation.RequestMappingHandlerMapping] - Mapped "{[/registercustomer]}" onto public org.springframework.web.servlet.ModelAndView com.user.controller.RegisterController.registerCustomer(java.lang.String,java.lang.String,java.lang.String,javax.servlet.http.HttpServletRequest,javax.servlet.http.HttpServletResponse)

2019-12-03 12:30:00,151 INFO

[org.springframework.web.servlet.mvc.method.annotation.RequestMappingHandlerMapping] - Mapped "{[/searchflights]}" onto public org.springframework.web.servlet.ModelAndView com.user.controller.SearchFlightsController.search(java.lang.String,java.lang.String,java.lang.String,java.lang.String,java.lang.String,java.lang.String,javax.servlet.http.HttpServletRequest,javax.servlet.http.HttpServletResponse)

2019-12-03 12:30:00,308 INFO

[org.springframework.web.servlet.mvc.method.annotation.RequestMappingHandlerAdapter] - Looking for @ControllerAdvice: WebApplicationContext for namespace 'spring-servlet': startup date [Tue Dec 03 12:29:58 IST 2019]; root of context hierarchy

2019-12-03 12:30:00,664 INFO [org.springframework.web.servlet.DispatcherServlet] - FrameworkServlet 'spring': initialization completed in 1915 ms

2019-12-03 12:30:01,490 INFO [com.user.service.LoginService] - customer sourav login successful

2019-12-03 12:30:01,490 INFO [com.user.service.LoginService] - customer sourav login successful

2019-12-03 12:31:30,686 WARN [com.user.service.RegisterService] - customer sourav already exists

2019-12-03 12:31:51,277 ERROR [com.user.service.LoginService] - customer asfasf login failed : username not found

2019-12-03 12:31:59,207 INFO [com.user.service.LoginService] - customer sourav login successful

2019-12-03 12:32:09,747 INFO [com.user.service.LoginService] - customer sourav login successful

2019-12-03 12:32:44,408 INFO [com.user.service.FlightService] - Getting the flight details

2019-12-03 12:33:22,982 INFO [com.user.service.LoginService] - manager sourav login successful

2019-12-03 12:33:36,537 INFO [com.user.service.LoginService] - manager sourav login successful