

# Online Air ticket Booking system

## PROBLEM STATEMENT

Online flight booking system provides users the freedom to browse for flights across the globe, view the flight timings, flight fares and the duration of journey before booking the tickets. After the user registration, the user will be able to schedule their flights and make payment for their flight fare. A single transaction can contain tickets for multiple passengers. The administrator will be responsible for updating the flight schedule and prices in the system. The administrator will also have the ability to add new airlines into the system.

### Acronyms and Abbreviations:

SRS - Software Requirements Specification

GUI - Graphical User Interface

### Functional Requirements:

1. Separate authentication screen for users and administrators.
2. User registration screen where new users can register themselves and book flights
3. Home screen for users to book flight, update their profile and save their flight preferences
4. Search screen to search flights between a source and a destination locations, date and time
5. Flight booking screen where user can enter passenger details
6. Users will be able to see their booking history
7. Users will have the ability to modify their search result based on various filters, including class, timing and/or fare
8. After booking and payment, the e-ticket should be displayed to the user for download/print.
9. User will also have an option to email their ticket to their email address.
10. Administrator can will have an option to add new flights to the airline's fleet.
11. Administrator can modify/delete existing flights in the airline's fleet.
12. Admin will have an option to view all passenger's details.
13. Admin will have an option to see the seat availability of any flight.
14. Standard validations will be implemented during the system design.

### User Story

Guest user should be able to surf the system

### Solution:

1. Guest user will be able to search for flights based on locations and time
2. He cannot book ticket without login/registration.
3. He should be redirected to registration page when he attempts to book a flight.

### User Story

New user should be able to register to the system

### Solution:

1. System should accept username and password.
2. Username should be validated for availability.

**User Story**

Registered user should be able to login to the system

**Solution:**

1. The system should authenticate user's username and password.
2. Username and password cannot be left blank.
3. Alert to be raised if the username and/or password are incorrect.
4. If valid username and password is entered, the user should be directed to the search/book flight page.

**User Story**

User should be able to search flight

**Solution:**

1. User should be able to search flight by source and destination locations, and date of travel.
2. User should be able to define class of travel (economy/premium/business).

**User Story**

User should be able to book ticket

**Solution:**

1. Ticket gets booked only when user is logged in.
2. Flight source and destination location is taken from selected search result.
3. Date of travel is taken from selected search result.
4. The session should block the seat(s) until booking confirmed.

**User Story**

User should be able to confirm/cancel ticket before booking

**Solution:**

1. User should be shown previously filled booking details (passenger details, date and time of travel, cost of travel etc.) before confirming booking.
2. User should be shown availability of seats in the chosen flight.
3. User should have option to cancel ticket before booking.

**User Story**

User should be able to make payment

**Solution:**

1. User should be directed to payment option after he confirms ticket details.
3. User should have payment option (by credit card/debit card/internet banking).

**User Story**

User should be able to view booked ticket

**Solution:**

1. After booking and payment, the e-ticket should be displayed to the user for download/print.
2. A copy of e-ticket should be sent to the user by email.

**User Story**

Admin should be able to login to the system

**Solution:**

1. The system should authenticate admin's username and password.
2. Username and password cannot be left blank.
3. Alert to be raised if the username and/or password are incorrect.
4. If valid username and password is entered, the admin should be directed to the dashboard.

**User Story US-FBS-502**

Admin should be able to add new flight operating details

**Business rules:**

1. New flight ID to be auto-generated.
2. Flight arrival and departure times and locations can be added.
3. Flight fares can be defined.

**Solution:**

1. Provide an option to add new flight in the system.
2. On successful addition, show the flight details on dashboard.

**User Story**

Admin should be able to modify/delete flight operating details

**Business rules:**

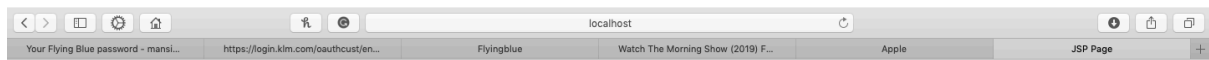
1. Flight arrival and departure times can be modified.
2. Flight locations can be modified.
3. Flight fares can be modified.
4. Current flight details like time can be modified.

**Solution:**

1. Provide an option to modify a flight in the system.
2. On successful modification/deletion, show the details on dashboard.

**USE CASE DIAGRAM**



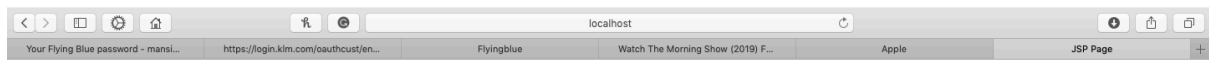


[Go Back](#)

## Register a New Person

Name:

Password:



[Go Back](#)

## Login Page

Name:

Password:

Hi, mansi

[Add Flight](#)  
[View Flights](#)  
[View Passenger List](#)

[logout now](#)



[Home](#)

## Add a New Flight

Flight Name	<input type="text"/>
Airplane Id	<input type="text" value="0"/>
From	<input type="text"/>
Dest	<input type="text"/>
Departure Time	<input type="text"/>
Arrival Time	<input type="text"/>
Travel Class	<input type="text"/>
Total Seats	<input type="text" value="0"/>
Available Seats	<input type="text" value="0"/>
Amount	<input type="text" value="0"/>
Departure Date	<input type="text"/>
Arrival Date	<input type="text"/>

[Add Flight to database](#)

December 2019

Su	Mo	Tu	We	Th	Fr	Sa
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31				

Next

January 2020

Su	Mo	Tu	We	Th	Fr	Sa
	1	2	3	4		
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30	31	



# View Flights

Hi,

Type to search

Avaiable seats

Flight Name	Airplane Id	From	Dest	Departure Time	Arrival Time	Travel Class	Total Seats	Amount	Departure Date	Arrival Date	
skjn	8	bkj	nk	nk	kn	knk	8	8	12/20/2019	12/20/2019	1
delta	79	boston	delhi	68899	8979	economy	77	100	19/20/30	19/20/30	76
bsakjh	78	khk	bjkbk	88979	9879	jkhkl	68	9	12/20/2019	12/20/2019	63
kamini	8	hbkj	kjhk	78908	980	hkhl	89	89	12/14/2019	12/19/2019	87
sakhl	998	hkhk	knkl	78686	65	gigkj	67	879	12/30/2019	12/31/2019	64
savjhg	87	gj	hk	687	798	ghj	87	787	12/13/2019	12/13/2019	87
sagjsak	89	vjhgb	kjhk	879	98	gvjg	8	698	12/31/2019	12/31/2019	7
vshdh	88	boston	sjnk	g8778	hjb	jhbhj	8	78	12/19/2019	12/19/2019	8

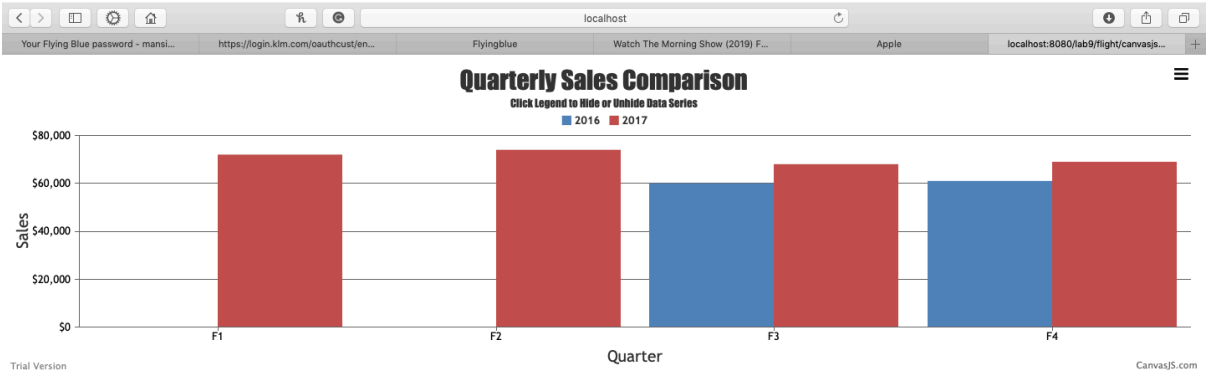
# View Flights

Hi, mansi

Type to search

Avaiable seats

Flight Name	Airplane Id	From	Dest	Departure Time	Arrival Time	Travel Class	Total Seats	Amount	Departure Date	Arrival Date	
skjn	8	bkj	nk	nk	kn	knk	8	8	12/20/2019	12/20/2019	1 <a href="#">Update Flight</a> <a href="#">Delete Flight</a>
delta	79	boston	delhi	68899	8979	economy	77	100	19/20/30	19/20/30	76 <a href="#">Update Flight</a> <a href="#">Delete Flight</a>
bsakjh	78	khk	bjkbk	88979	9879	jkhkl	68	9	12/20/2019	12/20/2019	63 <a href="#">Update Flight</a> <a href="#">Delete Flight</a>
kamini	8	hbkj	kjhk	78908	980	hkhl	89	89	12/14/2019	12/19/2019	87 <a href="#">Update Flight</a> <a href="#">Delete Flight</a>
sakhl	998	hkhk	knkl	78686	65	gigkj	67	879	12/30/2019	12/31/2019	64 <a href="#">Update Flight</a> <a href="#">Delete Flight</a>
savjhg	87	gj	hk	687	798	ghj	87	787	12/13/2019	12/13/2019	87 <a href="#">Update Flight</a> <a href="#">Delete Flight</a>
sagjsak	89	vjhgb	kjhk	879	98	gvjg	8	698	12/31/2019	12/31/2019	7 <a href="#">Update Flight</a> <a href="#">Delete Flight</a>
vshdh	88	boston	sjnk	g8778	hjb	jhbhj	8	78	12/19/2019	12/19/2019	8 <a href="#">Update Flight</a> <a href="#">Delete Flight</a>



## Hello World!

Flight ID	3
Flight Name	skjn
Airplane Id	8
From	bkjbjk
Dest	nk
Departure Time	nkkn
Arrival Time	kn
Travel Class	knk
Total Seats	8
Available Seats	1
Amount	8
Departure Date	12/20/2019
Arrival Date	12/20/2019
<button>Update flight to database</button>	

[Go Back to Menu Page](#)

mansi [com.me.pojo.Ticket@473b5f0b, com.me.pojo.Ticket@2d409771, com.me.pojo.Ticket@5468b490, com.me.pojo.Ticket@7a6ff644, com.me.pojo.Ticket@67b12d05, com.me.pojo.Ticket@4ee3d59d, com.me.pojo.Ticket@5f8e24e5, com.me.pojo.Ticket@5ae63480, com.me.pojo.Ticket@3451e3dc, com.me.pojo.Ticket@644e97f6, com.me.pojo.Ticket@4af90e96, com.me.pojo.Ticket@21d0717d, com.me.pojo.Ticket@1a243998, com.me.pojo.Ticket@ac5fc0d, com.me.pojo.Ticket@2a331d80, com.me.pojo.Ticket@116019a7, com.me.pojo.Ticket@34e0aa12]

### Below are the passengers details

S.No	Passenger Name	Passenger Email	Flight Name	Travelling From	Destination	Date of travel
1	mansi	webtools1994@gmail.com	skjn	bkjbjk	nk	12/20/2019
2	carrie	webtools1994@gmail.com	skjn	bkjbjk	nk	12/20/2019
3	SIMI	MAK@BKS.COM	bsakjh	khk	bjkbbk	12/20/2019
4	gigi	szbakj@bjsdbk.com	sakhl	hkhk	knkl	12/30/2019
5	sakjn	asm@nd.com	sagjgsak	vjhgb	kjhk	12/31/2019
6	213	dsf	delta	boston	delhi	19/20/30
7	skjb	sabkj@bsj.com	skjn	bkjbjk	nk	12/20/2019
8	sjb	webtools1994@gmail.com	bsakjh	khk	bjkbbk	12/20/2019
9	krish	webtools1994@gmail.com	bsakjh	khk	bjkbbk	12/20/2019
10	xbkjm	bam@dnj.com	bsakjh	khk	bjkbbk	12/20/2019
11	SAJH	MABSJ@BJS.CO	kamini	hbkjnk	kjhjk	12/14/2019
12	jhb	bjh2dbj@bsdj	sakhl	hkhk	knkl	12/30/2019
13	snak	sjk@gmail.com	sakhl	hkhk	knkl	12/30/2019
14	xshjzb	bszkjbjk	bsakjh	khk	bjkbbk	12/20/2019
15	jhgk	mas@dgk	skjn	bkjbjk	nk	12/20/2019
16	bdjsbjk	mabjhsg	skjn	bkjbjk	nk	12/20/2019
17	vhhkjgk	gikga	kamini	hbkjnk	kjhjk	12/14/2019

[Go Back to Menu Page](#)

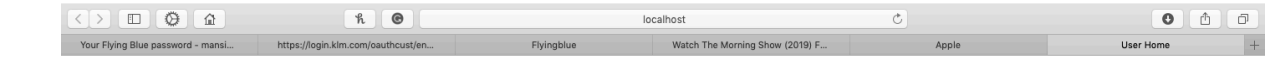
## View Flights

Hi, krish

[Available seats](#)

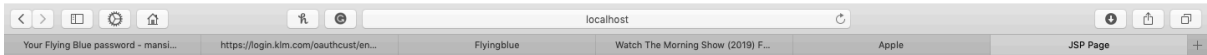
Flight Name	Airplane Id	From	Dest	Departure Time	Arrival Time	Travel Class	Total Seats	Amount	Departure Date	Arrival Date	
skjn	8	bkjbjk	nk	nkkn	kn	knk	8	8	12/20/2019	12/20/2019	1 <a href="#">Book Ticket</a>
delta	79	boston	delhi	68899	8979	economy	77	100	19/20/30	19/20/30	76 <a href="#">Book Ticket</a>
bsakjh	78	khk	bjkbbk	88979	9879	jkhkl	68	9	12/20/2019	12/20/2019	63 <a href="#">Book Ticket</a>
kamini	8	hbkjnk	kjhjk	78908	980	hkhk	89	89	12/14/2019	12/19/2019	87 <a href="#">Book Ticket</a>
sakhl	998	hkhk	knkl	78686	65	gigkj	67	879	12/30/2019	12/31/2019	64 <a href="#">Book Ticket</a>
savjhgbk	87	gj	hk	687	798	ghj	87	787	12/13/2019	12/13/2019	87 <a href="#">Book Ticket</a>
sagjgsak	89	vjhgb	kjhk	879	98	gvjgbk	8	698	12/31/2019	12/31/2019	7 <a href="#">Book Ticket</a>
vsdh	88	boston	sjnk	g8778	hjb	jhbhj	8	78	12/19/2019	12/19/2019	8 <a href="#">Book Ticket</a>





Hi,

[Person register here](#)  
[Person login here](#)  
[View Flights](#)



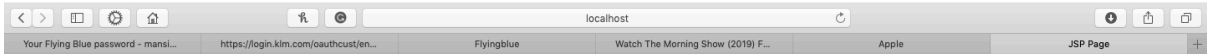
Hello krish

[Home](#)

Your cart contents

[Proceed to enter details and Payment>>](#)

Flight Number	Flight Name	Airplane Id	Price	Departure time	Destination	arrival time	
3	skjn	8	\$8	nkn	kn		<a href="#">[Remove from cart]</a>
Total	8.0						



Hi, krish

Please enter all details correctly

First Name:

Last Name:

Gender ☒ Male ☐ Female

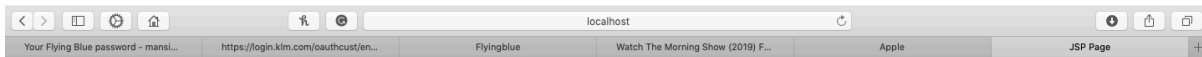
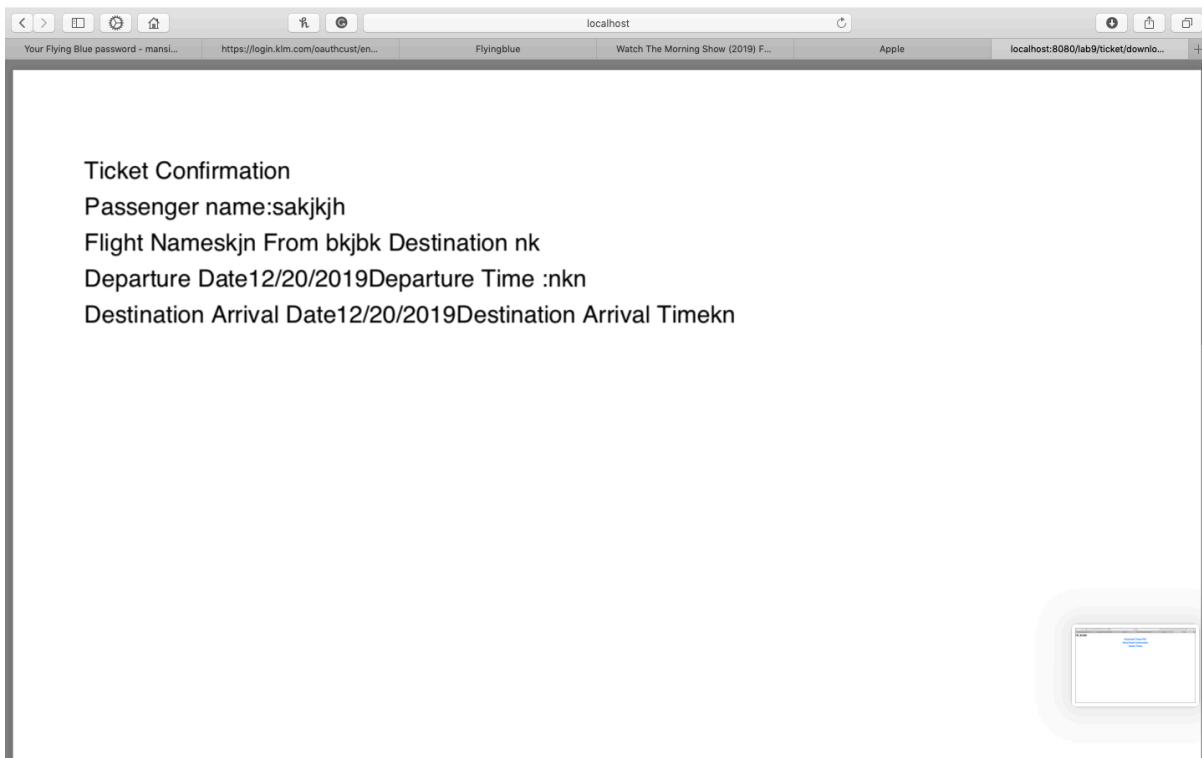
Email:

Date of Birth:

Phone Number:

Address:





Hi, krish

- [Download Ticket PDF](#)
- [Send Email Confirmation](#)
- [Delete Ticket](#)

Hi, krish

Credit Card Number

Bank Name

Name as on card

Expiration Month

Expiration Year

## Controller code

```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package com.me.controller;

import org.json.JSONException;
import org.json.JSONObject;
import com.me.dao.PersonDao;
import com.me.exception.PersonException;
import com.me.pojo.Person;
import com.me.validator.PersonValidator;
import java.io.IOException;
import java.io.PrintWriter;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.beans.factory.annotation.Qualifier;
import org.springframework.stereotype.Controller;
import org.springframework.validation.BindingResult;
import org.springframework.web.bind.WebDataBinder;
```

```

import org.springframework.web.bind.annotation.InitBinder;
import org.springframework.web.bind.annotation.ModelAttribute;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RequestMethod;
import org.springframework.web.servlet.ModelAndView;

/**
 *
 * @author mansiagrawal
 */
@Controller
@RequestMapping("/person/**")
public class PersonController {

    @Autowired
    @Qualifier("personDao")
    PersonDao personDao;

    @Autowired
    @Qualifier("personValidator")
    PersonValidator personValidator;

    @InitBinder("person")
    private void initBinder(WebDataBinder binder) {
        binder.setValidator(personValidator);
    }

    @RequestMapping(value = "/", method = RequestMethod.GET)
    protected String goToUserHome(HttpServletRequest request) throws Exception
    {
        return "user-home";
    }

    @RequestMapping(value = "/person/register", method = RequestMethod.GET)
    protected ModelAndView registerUser() throws Exception {
        System.out.print("registerPerson");

        return new ModelAndView("register-person", "person", new Person());
    }

    @RequestMapping(value = "/person/register", method = RequestMethod.POST)

    public ModelAndView addPerson(HttpServletRequest request) throws
Exception {
        System.out.println("inside Product DAO");
        HttpSession session = (HttpSession) request.getSession();

        String role = "customer";
        String name = request.getParameter("name");
        String password = request.getParameter("password");
        if(personDao.userExists(request.getParameter("name"))){
            return new ModelAndView("error");
        }
        else{
            Person per=new Person(name,password,role);
            Person p = personDao.register(per);

```

```

        request.getSession().setAttribute("person", per);
//
        return new ModelAndView("userHome", "person", per);}

    }
    @RequestMapping(value = "/person/login", method = RequestMethod.GET)
    protected String loginUserHome(HttpServletRequest request) throws Exception {
        return "login-person";
    }

    @RequestMapping(value = "/person/login", method = RequestMethod.POST)
    protected ModelAndView loginUser(HttpServletRequest request) throws
Exception {
        HttpSession session = request.getSession();
        String name = request.getParameter("name");
        String password = request.getParameter("password");

        boolean flag=false;
        boolean check = false;
        try{

            flag = personDao.validateAdmin(name, password);
            check = personDao.validateUser(name, password);

        }
        catch(PersonException e){

            System.out.println("Exception: " + e.getMessage());

        }

        if(flag)
        {
            Person u = personDao.get(request.getParameter("name"),
request.getParameter("password"));
            System.out.println("Login successful" +u.getName());
            session.setAttribute("person", u);
            return new ModelAndView("adminHome");

        }

        else if(check){
            Person u = personDao.get(request.getParameter("name"),
request.getParameter("password"));
            System.out.println("Login successful");
            session.setAttribute("person", u);
//            return "userHome";
            return new ModelAndView("userHome", "person", u);

        }
        else{
            return new ModelAndView("error");
        }
//

```

```

    }
    @RequestMapping(value = "/person/logout", method = RequestMethod.GET)
    protected String logout(HttpServletRequest request) throws Exception {
        HttpSession session = (HttpSession) request.getSession();
        session.setAttribute("person", null);
        return "user-home";
    }
}

/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package com.me.controller;

import com.me.dao.FlightDao;
import com.me.pojo.Flight;
import com.me.pojo.FlightDetail;
import com.me.pojo.Person;
import com.me.pojo.User;
import java.io.IOException;
import java.io.PrintWriter;
import java.util.ArrayList;
import java.util.List;
import java.util.Map;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.beans.factory.annotation.Qualifier;
import org.springframework.stereotype.Controller;
import org.springframework.ui.ModelMap;
import org.springframework.validation.BindingResult;
import org.springframework.web.bind.annotation.ModelAttribute;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RequestMethod;
import org.springframework.web.servlet.ModelAndView;

/**
 *
 * @author mansiagrawal
 */
@Controller
@RequestMapping("/flight/*")
public class FlightController {
    @Autowired
    @Qualifier("flightDao")
    FlightDao flightDAO;
    private static int startpoint = 1;
    private static int pagesize = 3;
    private static int pagecount = 1;

    // @RequestMapping(value = "/flight/add", method = RequestMethod.GET)
    // public ModelAndView initializeForm(HttpServletRequest request) throws Exception
    // {
    //     HttpSession session = (HttpSession) request.getSession();

```

```

//      User user = (User)session.getAttribute("user");
//      if (user == null) {
//          return new ModelAndView("login-user");
//      }
//      return new ModelAndView("flight-form", "flight", new Flight());
//  }
@RequestMapping(value = "/flight/add", method = RequestMethod.GET)
public ModelAndView initializeForm(HttpServletRequest request) throws Exception {
    HttpSession session = (HttpSession) request.getSession();
    Person user = (Person)session.getAttribute("person");
    if (user == null) {
        return new ModelAndView("login-user");
    }
    return new ModelAndView("flight-form", "flight", new FlightDetail());
}

@RequestMapping(value = "/flight/add", method = RequestMethod.POST)
protected String doSubmitAction(@ModelAttribute("flight") FlightDetail
fd, BindingResult result) throws Exception
{
    //
    //      validator.validate(fd, result);
    //      if(result.hasErrors()){
    //          return "addFlights";
    //      }

    try
    {
        long airplane_id = fd.getAirplane_id();
        String from=fd.getFrom();
        //      from = from.replaceAll("[^A-Za-z]+$", "");
        String dest=fd.getDest();
        //      dest = dest.replaceAll("[^A-Za-z]+$", dest);
        String deptTime=fd.getDeptTime();
        //      deptTime = deptTime.replaceAll("[^\\d:]", "");
        String arrivalTime=fd.getArrivalTime();
        //      arrivalTime = arrivalTime.replaceAll("[^\\d:]", "");
        String travelClass=fd.getTravelClass();
        //      travelClass = travelClass.replaceAll("[^A-Za-z]+$", "");
        int totalSeats=fd.getTotalSeats();
        int amount=fd.getAmount();
        String deptDate=fd.getDeptDate();
        String arrDate=fd.getArrDate();
        String flight_name=fd.getFlight_name();

        //      FlightDao fdao = new FlightDao();
        FlightDetail flight =
flightDAO.createFlight(flight_name,airplane_id,from,dest,deptTime,arrivalTime,travel
Class,totalSeats,totalSeats,amount,deptDate,arrDate);
        //DAO.close();

    }
    catch (Exception e)
    {
        System.out.println("Exception: " + e.getMessage());
    }
}

```

```

        return "adminHome";
    }

    @RequestMapping(value="/flight/list", method=RequestMethod.GET)
    public ModelAndView listForm(HttpServletRequest request){
//    {
//
        try {

            List<FlightDetail> flight = flightDAO.list();

            return new ModelAndView("ListFlights", "flight", flight);

        } catch (Exception e) {
            System.out.println(e.getMessage());
            return new ModelAndView("error", "errorMessage", "error
while login");
        }
    }

    @RequestMapping(value = "/flight/pagination", method =
RequestMethod.GET)
    public ModelAndView GetFlightPageWise(HttpServletRequest request)
throws Exception {

        try {
            List<FlightDetail> flight;
            pagecount++;
            startpoint = (pagesize * pagecount) + 1 ;

            flight = flightDAO.listf(startpoint,pagesize);
            System.out.println("flight "+flight.size());
            return new ModelAndView("ListFlights", "flight", flight);

        } catch (Exception e) {
            System.out.println(e.getMessage());
            return new ModelAndView("error", "errorMessage", "error while
fetching products");
        }

    }

    @RequestMapping(value="/canvasjschart",method = RequestMethod.GET)
    public String springMVC(ModelMap modelMap) {
        List<List<Map<Object, Object>>> canvasjsDataList = flightDAO.getData();
        modelMap.addAttribute("dataPointsList", canvasjsDataList);
        return "chart";
    }

    @RequestMapping(value="/updateFlights.htm", method=RequestMethod.GET)
    public String updateFlights(@ModelAttribute("fd") FlightDetail fd,
HttpServletRequest request) throws Exception
    {
        String id = request.getParameter("id");
        int flight_id = Integer.parseInt(id);

        String action = request.getParameter("action");

```



```

        HttpSession session = request.getSession();

        if(action!=null){
            if(action.equalsIgnoreCase("update"))
            {
                FlightDetail flight = flightDAO.searchFlightByID(flight_id);
                //int oldAvailable = flight.getAvailableSeats();
                request.setAttribute("flight", flight);
                //session.setAttribute("oldAvailable", oldAvailable);
                return "updateFlight";
            }

        }

        return "ListFlights";
    }

    @RequestMapping(value="/updateFlight.htm", method=RequestMethod.POST)
    public String update(@ModelAttribute("flight") FlightDetail flight, BindingResult
result, HttpServletRequest request)
    {
        //HttpSession session = request.getSession();

        try{

            flightDAO.updateFlight(flight);
            System.out.println("Now Seats available
are"+flight.getAvailableSeats()+flight.getFlight_name());
            System.out.println("Flight saved/updated successfully");

        }
        catch(Exception e)
        {
            System.out.println("Exception: " + e.getMessage());
        }

        return "adminHome";
    }

    @RequestMapping(value="/deleteFlights.htm", method=RequestMethod.GET)
    public String deleteFlights(@ModelAttribute("fd") FlightDetail fd,
HttpServletRequest request) throws Exception
    {
        try{

            String id = request.getParameter("id");

            int flight_id = Integer.parseInt(id);

            HttpSession session = request.getSession();

            FlightDetail flight = flightDAO.searchFlightByID(flight_id);

            //
            tdao.deleteTickets(flight_id);

            flightDAO.deleteFlight(flight);

        }
        catch(Exception e)
        {

```

```

        System.out.println("Exception: " + e.getMessage());
    }

    return "adminHome";
}
//
//
@RequestMapping(value="/addToCart.htm", method=RequestMethod.GET)
public String initialize(HttpServletRequest request, HttpServletResponse
response) throws Exception, IOException{

    try{
        HttpSession session = request.getSession();
        System.out.println("flight id" + request.getParameter("fid"));
        int flightid = Integer.parseInt(request.getParameter("fid"));
        System.out.println("Flight ID is"+flightid);
        List<FlightDetail> cart;

        FlightDetail fd = flightDAO.searchFlightByID(flightid);

        System.out.println("Cart Controller:"+fd.getAvailableSeats());

        int noOfSeats = fd.getAvailableSeats();
        PrintWriter out = response.getWriter();
        if(noOfSeats>0)
        {

            out.println("Seats are available");
            if (session.getAttribute("cart") != null) {
                cart = (ArrayList<FlightDetail>) session.getAttribute("cart");
            } else {
                cart = new ArrayList<FlightDetail>();
            }

            cart.add(fd);
            session.setAttribute("cart", cart);
            session.setAttribute("flight", fd);

            float total = 0;
            int quantity=0;
            for (FlightDetail f : cart) {

                total = total + f.getAmount();
            }

            session.setAttribute("total", total);

            return "viewCart";

        }

        else
        {

            return "notAvailable";

        }
    }
}

```

```

    }
    catch(Exception e)
    {
        System.out.println(e.getMessage());
        return null;
    }
}

```

```

@RequestMapping(value="/removeFromCart.htm", method=RequestMethod.GET)
public String removeItems(HttpServletRequest request) throws Exception{

    HttpSession session = request.getSession();
    try{

        List<FlightDetail> cart =(ArrayList<FlightDetail>)
session.getAttribute("cart");
        String id = request.getParameter("id");
        int flight_id = Integer.parseInt(id);

        for(FlightDetail fd:cart){
            if(fd.getFlight_id()==flight_id){
                cart.remove(fd);
                break;
            }
        }

        session.setAttribute("cart", cart);

        float total = 0;
        for (FlightDetail f : cart) {
            total = total + f.getAmount();
        }

        session.setAttribute("total", total);
    }

    catch(Exception e)
    {
        System.out.println("Could not remove flight object"+ e);
    }

    return "viewCart";
}
}

```

```

/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.

```

```

*/
package com.me.controller;

import com.me.dao.PassengerDao;
import com.me.dao.TicketDao;
import com.me.pojo.Passenger;
import com.me.pojo.Person;
import com.me.pojo.Ticket;
import java.util.ArrayList;
import java.util.List;
import java.util.regex.Matcher;
import java.util.regex.Pattern;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpSession;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.beans.factory.annotation.Qualifier;
import org.springframework.stereotype.Controller;
import org.springframework.validation.BindingResult;
import org.springframework.web.bind.annotation.ModelAttribute;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RequestMethod;
import org.springframework.web.servlet.ModelAndView;

/**
 *
 * @author mansiagrawal
 */
@Controller
@RequestMapping(value="/*passenger*.htm")
public class PassengerController {
    @Autowired
    @Qualifier("passengerDao")
    PassengerDao passengerDAO;
    @Autowired
    @Qualifier("ticketDao")
    TicketDao ticketDAO;

    @RequestMapping(value = "/passenger1.htm", method = RequestMethod.GET)
    public String checkUserInSession(HttpServletRequest request,
    @ModelAttribute("passenger") Passenger passenger, BindingResult result ) {

        HttpSession session = request.getSession();
        Person username = (Person) session.getAttribute("person");
        System.out.println(username);

        if (username == null) {
            return "message";
        }

        else {
            return "passenger";
        }

    }

    @RequestMapping(value="/passenger.htm", method=RequestMethod.GET)
    public String initialize(@ModelAttribute("passenger")Passenger
passenger,HttpServletRequest request, BindingResult result)

```

```

    {
        HttpSession session = request.getSession();
        int noOfTravellers=1;
        session.setAttribute("noOfTravellers",noOfTravellers);
        return "passenger";
    }

    @RequestMapping(value="/passenger.htm", method=RequestMethod.POST)
    public String doSubmit(@ModelAttribute("passenger")Passenger
passenger,BindingResult result, HttpServletRequest request)
    {
        HttpSession session = request.getSession();
        // validator.validate(passenger, result);
        // if(result.hasErrors()){
        //     return "passenger";
        // }

        try{

            String firstName=passenger.getFirstName();
            firstName = firstName.replaceAll("[^A-Za-z]+$", "");
            String lastName=passenger.getLastName();
            // lastName = lastName.replaceAll("[^A-Za-z]+$", "");
            String gender=passenger.getGender();
            String email=passenger.getEmail();
            // String EMAIL_PATTERN = "^[_A-Za-z0-9-\\+](\\.[_A-Za-z0-9-\\+])*@"
            // + "[A-Za-z0-9-]+(\\.[A-Za-z0-9-]+)*(\\.[A-Za-z]{2,})$";
            // Pattern pattern = Pattern.compile(EMAIL_PATTERN);
            // Matcher matcher = pattern.matcher(email);
            String dob=passenger.getDob();
            String phonenum=passenger.getPhonenum();
            phonenum = phonenum.replaceAll("[^0-9]", "-");
            String address=passenger.getAddress();

            Passenger pas =
passengerDAO.createPassenger(firstName,lastName,gender,email,dob,phonenum,addr
ess);

            long passenger_id = pas.getId();
            session.setAttribute("passenger_id", passenger_id);

        }
        catch(Exception e)
        {
            System.out.println("Could not create Passenger"+e.getMessage());
        }

        //Payment payment = new Payment();

        return "payment";
    }

    @RequestMapping(value="/viewpassengers.htm", method=RequestMethod.GET)
    public ModelAndView viewPassenger(HttpServletRequest request)
    {
        HttpSession session = request.getSession();
        List<Ticket> ticketList= new ArrayList<Ticket>();
        try{

```

```

        ticketList = ticketDAO.TicketList();
        System.out.println("----->" + ticketList.size());
        for(Ticket u: ticketList){
            System.out.println(u.getFlightDetails().getDeptDate());
        }
    }

    catch(Exception e)
    {
        System.out.println("Could not list passengers" + e.getMessage());
    }
    ModelAndView mv = new ModelAndView("passengerList", "ticketList",
ticketList);
    return mv;
}

}
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package com.me.controller;

import com.me.dao.FlightDao;
import com.me.dao.PassengerDao;
import com.me.dao.TicketDao;
import com.me.pojo.FlightDetail;
import com.me.pojo.Passenger;
import com.me.pojo.Payment;
import com.me.pojo.Ticket;
import javax.servlet.http.Cookie;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.beans.factory.annotation.Qualifier;
import org.springframework.stereotype.Controller;
import org.springframework.web.bind.annotation.ModelAttribute;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RequestMethod;

/**
 *
 * @author mansiagrawal
 */
@Controller
@RequestMapping(value = "/*payment*.htm")
public class PaymentController {

    @Autowired
    @Qualifier("passengerDao")
    PassengerDao passengerDAO;
    @Autowired
    @Qualifier("flightDao")

```

```

        FlightDao flightDAO;
        @Autowired
        @Qualifier("ticketDao")
        TicketDao ticketDAO;

        @RequestMapping(value = "/payment.htm", method = RequestMethod.GET)
        public String initialize(@ModelAttribute("payment") Payment payment) {

            return "payment";
        }
        @RequestMapping(value = "/payment.htm", method = RequestMethod.POST)
        public String addPayment(HttpServletRequest request, HttpServletResponse
response) throws Exception {

            HttpSession session = request.getSession();
            long passenger_id = ((Long) session.getAttribute("passenger_id"));

            try {

                String number = request.getParameter("creditCardNumber");
//                number = number.replaceAll("[^0-9]", "");
                long creditCardNumber = Long.parseLong(number);

                String bankName = request.getParameter("bankName");
//                bankName = bankName.replaceAll("[^\\dA-Za-z]", "");
                String fullName = request.getParameter("fullName");
//                fullName = fullName.replaceAll("[^A-Za-z]+$", "");
                String expiration_month =
request.getParameter("expiration_month");
//                expiration_month = expiration_month.replaceAll("[^0-9]", "");
                Cookie exp_month = new Cookie("month", expiration_month);
                String expiration_year =
request.getParameter("expiration_year");
//                expiration_year = expiration_year.replaceAll("[^0-9]", "");
                Cookie exp_year = new Cookie("year", expiration_year);
                Payment paymnt =
passengerDAO.createPayment(creditCardNumber, bankName, fullName,
expiration_month,
                                expiration_year);

                passengerDAO.updatePassenger(passenger_id, paymnt);

                Passenger passenger =
passengerDAO.searchPassenger(passenger_id);
                FlightDetail flightDetail = (FlightDetail)
session.getAttribute("flight");

                Ticket t=ticketDAO.bookTicket(passenger, flightDetail);
                int availSeats = flightDetail.getAvailableSeats();
                flightDAO.updateAvailableSeats(flightDetail, availSeats,
availSeats-1);

//flightDetail.setAvailableSeats(availSeats - 1);

                response.addCookie(exp_month);
                response.addCookie(exp_year);
                session.setAttribute("ticket", t);
            }
        }
    }
}

```

```

        } catch (Exception e) {
            System.out.println("Could not add payment/ticket" +
e.getMessage());
        }

        return "printTicket";
    }
}

/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package com.me.controller;

import com.lowagie.text.Document;
import com.me.dao.PassengerDao;
import com.me.dao.TicketDao;
import com.me.pojo.FlightDetail;
import com.me.pojo.Passenger;
import java.io.ByteArrayOutputStream;
import javax.servlet.ServletOutputStream;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.beans.factory.annotation.Qualifier;
import org.springframework.stereotype.Controller;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RequestMethod;
import com.lowagie.text.Paragraph;
import com.lowagie.text.pdf.PdfWriter;
import java.util.Properties;
import javax.mail.Authenticator;
import javax.mail.Message;
import javax.mail.Session;
import javax.mail.Transport;
import javax.mail.internet.InternetAddress;
import javax.mail.internet.MimeMessage;
import org.springframework.mail.MailException;
import org.springframework.mail.SimpleMailMessage;
import org.springframework.mail.javamail.JavaMailSender;

/**
 *
 * @author mansiagrawal
 */
@Controller
@RequestMapping(value="/*ticket*")
public class TicketController {
    @Autowired
    @Qualifier("ticketDao")
    TicketDao ticketDAO;
    @Autowired
    @Qualifier("passengerDao")

```



```

        PassengerDao passengerDAO;
        @Autowired
        private JavaMailSender mailSender;

    /**
     * This method will send compose and send the message
     */
    public void sendEmail(String to, String subject, String body) {
        try {
            SimpleMailMessage message = new SimpleMailMessage();
            message.setTo(to);
            message.setSubject(subject);
            message.setText(body);
            mailSender.send(message);
        } catch (MailException ex) {
            to = "";
        }
    }
    //
    @RequestMapping(value="/downloadTicket.pdf", method=RequestMethod.GET)
    public void downloadTicket(HttpServletRequest request, HttpServletResponse
response) throws Exception
    {
        HttpSession session = request.getSession();
        long passenger_id = ((Long) session.getAttribute("passenger_id"));
        Passenger passenger = passengerDAO.searchPassenger(passenger_id);
        //Payment payment = (Payment)session.getAttribute("payment");
        FlightDetail flightDetail = (FlightDetail)session.getAttribute("flight");

        try{

            response.setContentType("application/pdf");

            Document document = new Document();
            ByteArrayOutputStream baos = new ByteArrayOutputStream();
            PdfWriter.getInstance(document, baos);
            document.open();
            Paragraph title = new Paragraph("Ticket Confirmation");
            Paragraph name = new Paragraph("Passenger name:"+passenger.getFirstName()+
+""+passenger.getLastName());
            Paragraph flight = new Paragraph("Flight Name"+flightDetail.getFlight_name()+"
From "+flightDetail.getFrom()+" Destination "+flightDetail.getDest());
            Paragraph deptDetails = new Paragraph("Departure
Date"+flightDetail.getDeptDate()+"Departure Time :"+flightDetail.getDeptTime());
            Paragraph arrDetails = new Paragraph("Destination Arrival
Date"+flightDetail.getArrDate()+"Destination Arrival
Time"+flightDetail.getArrivalTime());

            document.add(title);
            document.add(name);
            document.add(flight);
            document.add(deptDetails);
            document.add(arrDetails);

```

```

document.close();

ServletOutputStream out = response.getOutputStream();
baos.writeTo(out);
out.flush();

    }

    catch(Exception e)
    {
        System.out.println("Could not add ticket object"+e.getMessage());
    }

}
@RequestMapping(value="/emailTicket.htm", method=RequestMethod.GET)
public String emailTicket(HttpServletRequest request) throws Exception
{
    sendEmail("webtools1994@gmail.com","hi","b");
//
//
    return "confirmation";
}
@RequestMapping(value="/deleteTicket.htm", method=RequestMethod.GET)
public String deleteTicket(HttpServletRequest request) throws Exception
{
    try{
        HttpSession session = request.getSession();
        long passenger_id = ((Long) session.getAttribute("passenger_id"));
        Passenger passenger = passengerDAO.searchPassenger(passenger_id);
        FlightDetail flightDetail =
(FlightDetail)session.getAttribute("flightdetail");
        ticketDAO.cancelTicket(passenger, flightDetail);

    }
    catch(Exception e)
    {
        System.out.println("Could not cancel Ticket "+e.getMessage());
    }

    return "confirmation";
}
}

```