**SOFTWARE REQUIREMENTS:**

**Operating system** : Windows Family

# Front End : CSS,JSP,HTML

**Back End** : JAVA,Servlet,JDBC.

DFD for Airline Reservation System



**First Level of Data Flow Diagram for**

**System Login**

Password Verification

User Name

Log In

**SYSTEM**

**ADMINSTRATOR**

Verification

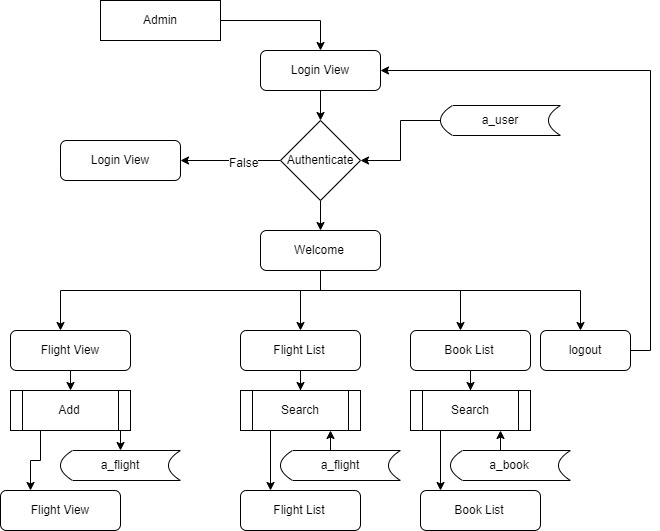
Fail

Verification

Success

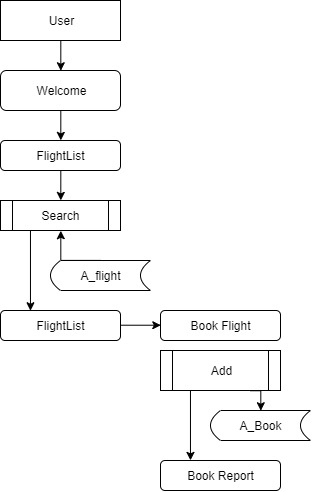
**Second Level of Data Flow Diagram for**

**General Inquiry System**

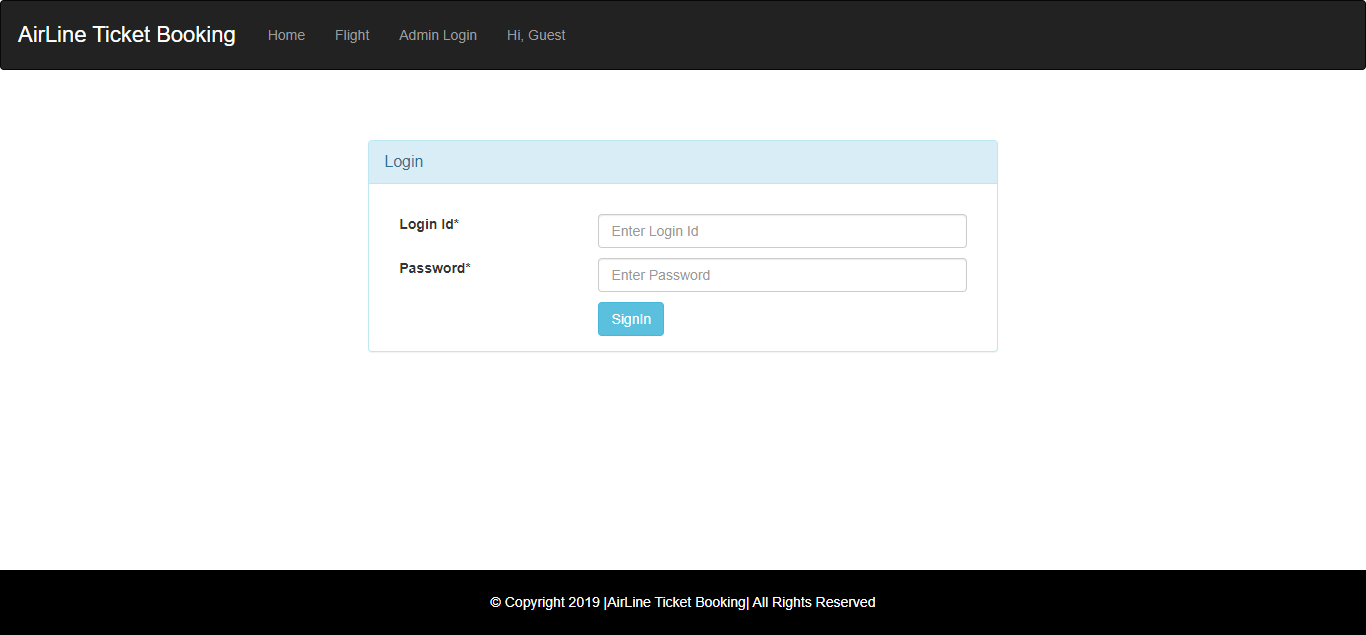
****

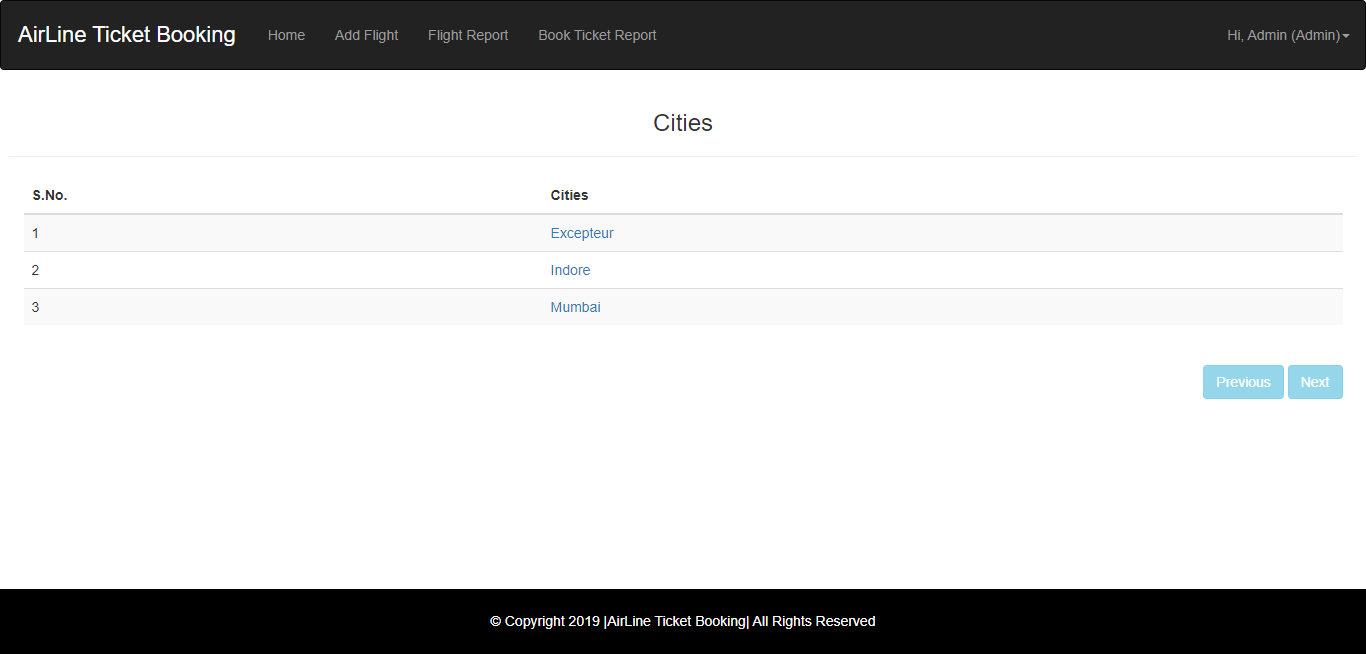
**Third Level DATA FLOW DIAGRAM**

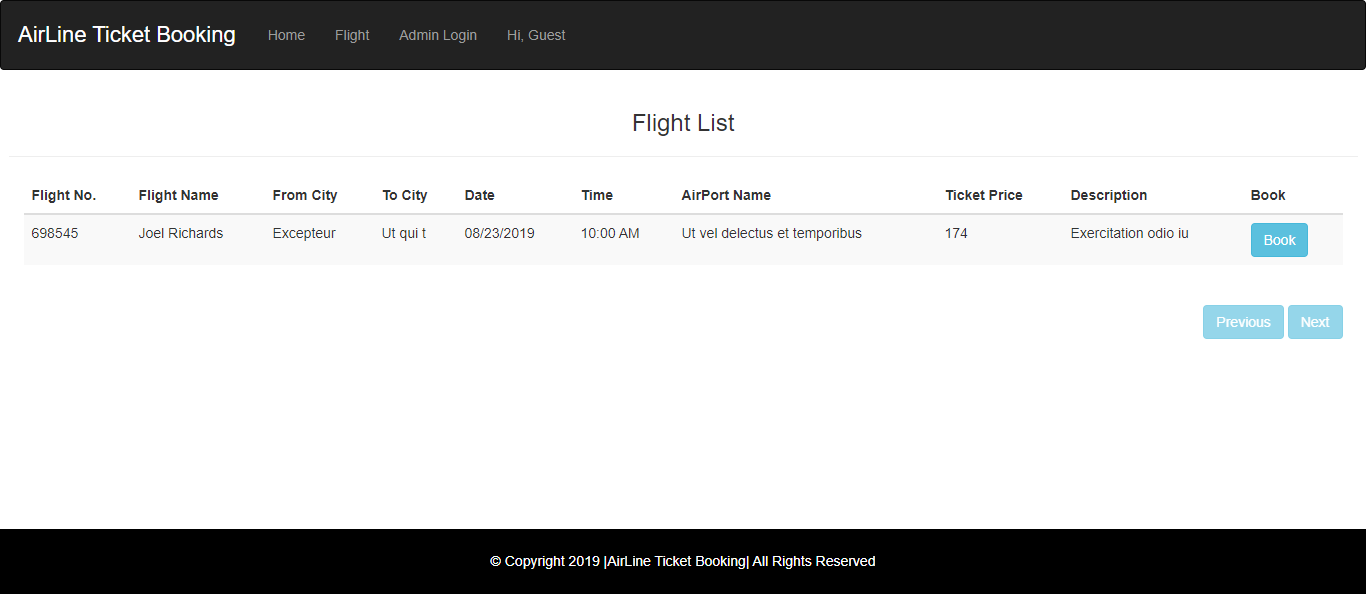
**OF BOOKING SECTION**

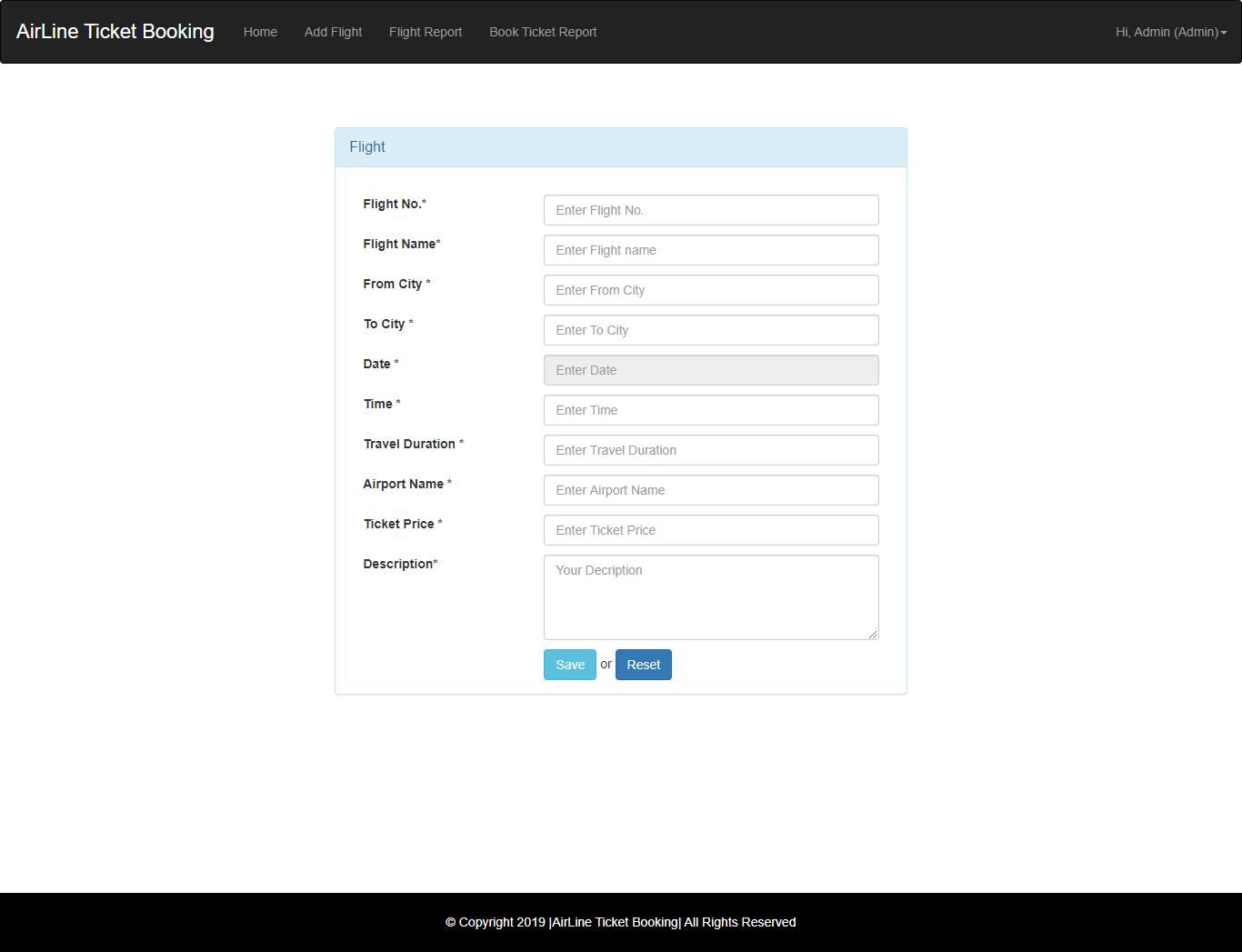


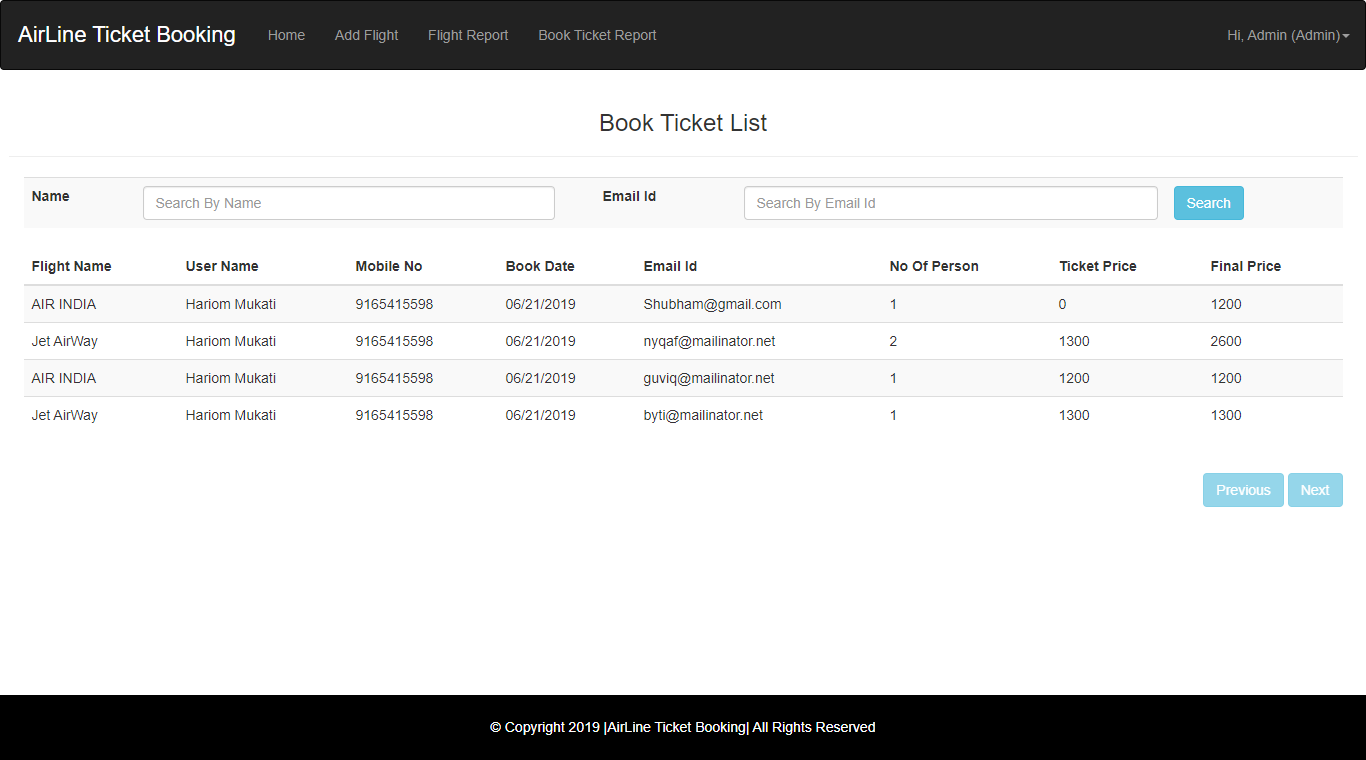
**Screen Short**











**Source Code**

**package** in.co.air.line.ticket.bean;

**import** java.util.Date;

**public** **class** FlightBean **extends** BaseBean {

**private** String flightNo;

**private** String fightName;

**private** String fromCity;

**private** String toCity;

**private** Date date;

**private** String description;

**private** String time;

**private** String travelDuraion;

**private** **long** ticketPrice;

**private** String airPortName;

**public** String getFlightNo() {

**return** flightNo;

}

**public** **void** setFlightNo(String flightNo) {

**this**.flightNo = flightNo;

}

**public** String getFightName() {

**return** fightName;

}

**public** **void** setFightName(String fightName) {

**this**.fightName = fightName;

}

**public** String getFromCity() {

**return** fromCity;

}

**public** **void** setFromCity(String fromCity) {

**this**.fromCity = fromCity;

}

**public** String getToCity() {

**return** toCity;

}

**public** **void** setToCity(String toCity) {

**this**.toCity = toCity;

}

**public** Date getDate() {

**return** date;

}

**public** **void** setDate(Date date) {

**this**.date = date;

}

**public** String getDescription() {

**return** description;

}

**public** **void** setDescription(String description) {

**this**.description = description;

}

**public** String getTime() {

**return** time;

}

**public** **void** setTime(String time) {

**this**.time = time;

}

**public** String getTravelDuraion() {

**return** travelDuraion;

}

**public** **void** setTravelDuraion(String travelDuraion) {

**this**.travelDuraion = travelDuraion;

}

**public** **long** getTicketPrice() {

**return** ticketPrice;

}

**public** **void** setTicketPrice(**long** ticketPrice) {

**this**.ticketPrice = ticketPrice;

}

**public** String getAirPortName() {

**return** airPortName;

}

**public** **void** setAirPortName(String airPortName) {

**this**.airPortName = airPortName;

}

**public** String getKey() {

// **TODO** Auto-generated method stub

**return** **null**;

}

**public** String getValue() {

// **TODO** Auto-generated method stub

**return** **null**;

}

}

package in.co.air.line.ticket.model;

import java.sql.Connection;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.util.ArrayList;

import java.util.List;

import org.apache.log4j.Logger;

import in.co.air.line.ticket.bean.FlightBean;

import in.co.air.line.ticket.bean.UserBean;

import in.co.air.line.ticket.exception.ApplicationException;

import in.co.air.line.ticket.exception.DatabaseException;

import in.co.air.line.ticket.exception.DuplicateRecordException;

import in.co.air.line.ticket.util.JDBCDataSource;

public class FlightModel {

private static Logger log = Logger.getLogger(FlightModel.class);

public Integer nextPK() throws DatabaseException {

log.debug("Model nextPK Started");

Connection conn = null;

int pk = 0;

try {

conn = JDBCDataSource.getConnection();

PreparedStatement pstmt = conn.prepareStatement("SELECT MAX(ID) FROM A\_Flight");

ResultSet rs = pstmt.executeQuery();

while (rs.next()) {

pk = rs.getInt(1);

}

rs.close();

} catch (Exception e) {

log.error("Database Exception..", e);

throw new DatabaseException("Exception : Exception in getting PK");

} finally {

JDBCDataSource.closeConnection(conn);

}

log.debug("Model nextPK End");

return pk + 1;

}

public long add(FlightBean bean) throws ApplicationException, DuplicateRecordException {

Connection conn = null;

int pk = 0;

FlightBean existbean = findByName(bean.getFightName());

if (existbean != null) {

throw new DuplicateRecordException("Flight is already exists");

}

try {

conn = JDBCDataSource.getConnection();

pk = nextPK();

// Get auto-generated next primary key

System.out.println(pk + " in ModelJDBC");

conn.setAutoCommit(false); // Begin transaction

PreparedStatement pstmt = conn.prepareStatement("INSERT INTO A\_Flight VALUES(?,?,?,?,?,?,?,?,?,?,?,?,?,?,?)");

pstmt.setInt(1, pk);

pstmt.setString(2, bean.getFlightNo());

pstmt.setString(3, bean.getFightName());

pstmt.setString(4, bean.getFromCity());

pstmt.setString(5, bean.getToCity());

pstmt.setDate(6, new java.sql.Date(bean.getDate().getTime()));

pstmt.setString(7, bean.getDescription());

pstmt.setString(8, bean.getTime());

pstmt.setString(9, bean.getTravelDuraion());

pstmt.setLong(10, bean.getTicketPrice());

pstmt.setString(11, bean.getAirPortName());

pstmt.setString(12, bean.getCreatedBy());

pstmt.setString(13, bean.getModifiedBy());

pstmt.setTimestamp(14, bean.getCreatedDatetime());

pstmt.setTimestamp(15, bean.getModifiedDatetime());

pstmt.executeUpdate();

conn.commit(); // End transaction

pstmt.close();

} catch (Exception e) {

try {

conn.rollback();

} catch (Exception ex) {

ex.printStackTrace();

throw new ApplicationException("Exception : add rollback exception " + ex.getMessage());

}

throw new ApplicationException("Exception : Exception in add User");

} finally {

JDBCDataSource.closeConnection(conn);

}

return pk;

}

public FlightBean findByName(String name) throws ApplicationException {

log.debug("Model findByLogin Started");

StringBuffer sql = new StringBuffer("SELECT \* FROM A\_Flight WHERE fightName=?");

FlightBean bean = null;

Connection conn = null;

System.out.println("sql" + sql);

try {

conn = JDBCDataSource.getConnection();

PreparedStatement pstmt = conn.prepareStatement(sql.toString());

pstmt.setString(1, name);

ResultSet rs = pstmt.executeQuery();

while (rs.next()) {

bean = new FlightBean();

bean.setId(rs.getLong(1));

bean.setFlightNo(rs.getString(2));

bean.setFightName(rs.getString(3));

bean.setFromCity(rs.getString(4));

bean.setToCity(rs.getString(5));

bean.setDate(rs.getDate(6));

bean.setDescription(rs.getString(7));

bean.setTime(rs.getString(8));

bean.setTravelDuraion(rs.getString(9));

bean.setTicketPrice(rs.getLong(10));

bean.setAirPortName(rs.getString(11));

bean.setCreatedBy(rs.getString(12));

bean.setModifiedBy(rs.getString(13));

bean.setCreatedDatetime(rs.getTimestamp(14));

bean.setModifiedDatetime(rs.getTimestamp(15));

}

rs.close();

} catch (Exception e) {

e.printStackTrace();

log.error("Database Exception..", e);

throw new ApplicationException("Exception : Exception in getting User by login");

} finally {

JDBCDataSource.closeConnection(conn);

}

log.debug("Model findByLogin End");

return bean;

}

/\*\*

\* Find User by PK

\*

\* @param pk

\* : get parameter

\* @return bean

\* @throws DatabaseException

\*/

public FlightBean findByPK(long pk) throws ApplicationException {

log.debug("Model findByPK Started");

StringBuffer sql = new StringBuffer("SELECT \* FROM A\_flight WHERE ID=?");

FlightBean bean = null;

Connection conn = null;

try {

conn = JDBCDataSource.getConnection();

PreparedStatement pstmt = conn.prepareStatement(sql.toString());

pstmt.setLong(1, pk);

ResultSet rs = pstmt.executeQuery();

while (rs.next()) {

bean = new FlightBean();

bean.setId(rs.getLong(1));

bean.setFlightNo(rs.getString(2));

bean.setFightName(rs.getString(3));

bean.setFromCity(rs.getString(4));

bean.setToCity(rs.getString(5));

bean.setDate(rs.getDate(6));

bean.setDescription(rs.getString(7));

bean.setTime(rs.getString(8));

bean.setTravelDuraion(rs.getString(9));

bean.setTicketPrice(rs.getLong(10));

bean.setAirPortName(rs.getString(11));

bean.setCreatedBy(rs.getString(12));

bean.setModifiedBy(rs.getString(13));

bean.setCreatedDatetime(rs.getTimestamp(14));

bean.setModifiedDatetime(rs.getTimestamp(15));

}

rs.close();

} catch (Exception e) {

e.printStackTrace();

log.error("Database Exception..", e);

throw new ApplicationException("Exception : Exception in getting User by pk");

} finally {

JDBCDataSource.closeConnection(conn);

}

log.debug("Model findByPK End");

return bean;

}

public void delete(FlightBean bean) throws ApplicationException {

Connection conn = null;

try {

conn = JDBCDataSource.getConnection();

conn.setAutoCommit(false); // Begin transaction

PreparedStatement pstmt = conn.prepareStatement("DELETE FROM A\_flight WHERE ID=?");

pstmt.setLong(1, bean.getId());

pstmt.executeUpdate();

conn.commit(); // End transaction

pstmt.close();

} catch (Exception e) {

try {

conn.rollback();

} catch (Exception ex) {

throw new ApplicationException("Exception : Delete rollback exception " + ex.getMessage());

}

throw new ApplicationException("Exception : Exception in delete User");

} finally {

JDBCDataSource.closeConnection(conn);

}

}

public void update(FlightBean bean) throws ApplicationException, DuplicateRecordException {

log.debug("Model update Started");

Connection conn = null;

FlightBean beanExist = findByName(bean.getFightName());

// Check if updated LoginId already exist

if (beanExist != null && !(beanExist.getId() == bean.getId())) {

throw new DuplicateRecordException("Flight is already exist");

}

try {

conn = JDBCDataSource.getConnection();

conn.setAutoCommit(false); // Begin transaction

PreparedStatement pstmt = conn.prepareStatement(

"UPDATE A\_Flight SET FlightNo=?,FightName=?,FromCity=?,ToCity=?,Date=?,Description=?,Time=?,TravelDuraion=?,TicketPrice=?,AirPortName=?,"

+ "CREATEDBY=?,MODIFIEDBY=?,CREATEDDATETIME=?,MODIFIEDDATETIME=? WHERE ID=?");

pstmt.setString(1, bean.getFlightNo());

pstmt.setString(2, bean.getFightName());

pstmt.setString(3, bean.getFromCity());

pstmt.setString(4, bean.getToCity());

pstmt.setDate(5, new java.sql.Date(bean.getDate().getTime()));

pstmt.setString(6, bean.getDescription());

pstmt.setString(7, bean.getTime());

pstmt.setString(8, bean.getTravelDuraion());

pstmt.setLong(9, bean.getTicketPrice());

pstmt.setString(10, bean.getAirPortName());

pstmt.setString(11, bean.getCreatedBy());

pstmt.setString(12, bean.getModifiedBy());

pstmt.setTimestamp(13, bean.getCreatedDatetime());

pstmt.setTimestamp(14, bean.getModifiedDatetime());

pstmt.setLong(15, bean.getId());

pstmt.executeUpdate();

conn.commit(); // End transaction

pstmt.close();

} catch (Exception e) {

e.printStackTrace();

log.error("Database Exception..", e);

try {

conn.rollback();

} catch (Exception ex) {

throw new ApplicationException("Exception : Delete rollback exception " + ex.getMessage());

}

throw new ApplicationException("Exception in updating User ");

} finally {

JDBCDataSource.closeConnection(conn);

}

log.debug("Model update End");

}

public List search(FlightBean bean) throws ApplicationException {

return search(bean, 0, 0);

}

/\*\*

\* Search User with pagination

\*

\* @return list : List of Users

\* @param bean

\* : Search Parameters

\* @param pageNo

\* : Current Page No.

\* @param pageSize

\* : Size of Page

\*

\* @throws DatabaseException

\*/

public List search(FlightBean bean, int pageNo, int pageSize) throws ApplicationException {

log.debug("Model search Started");

StringBuffer sql = new StringBuffer("SELECT \* FROM A\_Flight WHERE 1=1");

if (bean != null) {

if (bean.getId() > 0) {

sql.append(" AND id = " + bean.getId());

}

if (bean.getFightName() != null && bean.getFightName().length() > 0) {

sql.append(" AND FightName like '" + bean.getFightName() + "%'");

}

if (bean.getFlightNo() != null && bean.getFlightNo().length() > 0) {

sql.append(" AND FlightNo like '" + bean.getFlightNo() + "%'");

}

if (bean.getToCity() != null && bean.getToCity().length() > 0) {

sql.append(" AND toCity like '" + bean.getToCity() + "%'");

}

if (bean.getFromCity() != null && bean.getFromCity().length() > 0) {

sql.append(" AND FromCity like '" + bean.getFromCity() + "%'");

}

if (bean.getDate() != null && bean.getDate().getDate() > 0) {

sql.append(" AND Date = "+ new java.sql.Date(bean.getDate().getTime()));

}

}

// if page size is greater than zero then apply pagination

if (pageSize > 0) {

// Calculate start record index

pageNo = (pageNo - 1) \* pageSize;

sql.append(" Limit " + pageNo + ", " + pageSize);

// sql.append(" limit " + pageNo + "," + pageSize);

}

System.out.println("user model search :" + sql);

ArrayList list = new ArrayList();

Connection conn = null;

try {

conn = JDBCDataSource.getConnection();

PreparedStatement pstmt = conn.prepareStatement(sql.toString());

ResultSet rs = pstmt.executeQuery();

while (rs.next()) {

bean = new FlightBean();

bean.setId(rs.getLong(1));

bean.setFlightNo(rs.getString(2));

bean.setFightName(rs.getString(3));

bean.setFromCity(rs.getString(4));

bean.setToCity(rs.getString(5));

bean.setDate(rs.getDate(6));

bean.setDescription(rs.getString(7));

bean.setTime(rs.getString(8));

bean.setTravelDuraion(rs.getString(9));

bean.setTicketPrice(rs.getLong(10));

bean.setAirPortName(rs.getString(11));

bean.setCreatedBy(rs.getString(12));

bean.setModifiedBy(rs.getString(13));

bean.setCreatedDatetime(rs.getTimestamp(14));

bean.setModifiedDatetime(rs.getTimestamp(15));

list.add(bean);

}

rs.close();

} catch (Exception e) {

log.error("Database Exception..", e);

throw new ApplicationException("Exception : Exception in search user");

} finally {

JDBCDataSource.closeConnection(conn);

}

log.debug("Model search End");

return list;

}

public List list() throws ApplicationException {

return list(0, 0);

}

/\*\*

\* Get List of User with pagination

\*

\* @return list : List of users

\* @param pageNo

\* : Current Page No.

\* @param pageSize

\* : Size of Page

\* @throws DatabaseException

\*/

public List list(int pageNo, int pageSize) throws ApplicationException {

log.debug("Model list Started");

ArrayList list = new ArrayList();

StringBuffer sql = new StringBuffer("select \* from A\_Flight");

// if page size is greater than zero then apply pagination

if (pageSize > 0) {

// Calculate start record index

pageNo = (pageNo - 1) \* pageSize;

sql.append(" limit " + pageNo + "," + pageSize);

}

System.out.println("sql in list user :" + sql);

Connection conn = null;

try {

conn = JDBCDataSource.getConnection();

PreparedStatement pstmt = conn.prepareStatement(sql.toString());

ResultSet rs = pstmt.executeQuery();

while (rs.next()) {

FlightBean bean = new FlightBean();

bean.setId(rs.getLong(1));

bean.setFlightNo(rs.getString(2));

bean.setFightName(rs.getString(3));

bean.setFromCity(rs.getString(4));

bean.setToCity(rs.getString(5));

bean.setDate(rs.getDate(6));

bean.setDescription(rs.getString(7));

bean.setTime(rs.getString(8));

bean.setTravelDuraion(rs.getString(9));

bean.setTicketPrice(rs.getLong(10));

bean.setAirPortName(rs.getString(11));

bean.setCreatedBy(rs.getString(12));

bean.setModifiedBy(rs.getString(13));

bean.setCreatedDatetime(rs.getTimestamp(14));

bean.setModifiedDatetime(rs.getTimestamp(15));

list.add(bean);

}

rs.close();

} catch (Exception e) {

log.error("Database Exception..", e);

throw new ApplicationException("Exception : Exception in getting list of users");

} finally {

JDBCDataSource.closeConnection(conn);

}

log.debug("Model list End");

return list;

}

}

package in.co.air.line.ticket.controller;

import java.io.IOException;

import javax.servlet.ServletException;

import javax.servlet.annotation.WebServlet;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import org.apache.log4j.Logger;

import in.co.air.line.ticket.bean.BaseBean;

import in.co.air.line.ticket.bean.CityBean;

import in.co.air.line.ticket.bean.FlightBean;

import in.co.air.line.ticket.exception.ApplicationException;

import in.co.air.line.ticket.exception.DuplicateRecordException;

import in.co.air.line.ticket.model.CityModel;

import in.co.air.line.ticket.model.FlightModel;

import in.co.air.line.ticket.util.DataUtility;

import in.co.air.line.ticket.util.DataValidator;

import in.co.air.line.ticket.util.PropertyReader;

import in.co.air.line.ticket.util.ServletUtility;

/\*\*

\* Servlet implementation class FlightCtl

\*/

@WebServlet(name="FlightCtl",urlPatterns={"/ctl/FlightCtl"})

public class FlightCtl extends BaseCtl {

private static final long serialVersionUID = 1L;

private static Logger log=Logger.getLogger(FlightCtl.class);

/\*\*

\* Validate input Data Entered By User

\*

\* @param request

\* @return

\*/

@Override

protected boolean validate(HttpServletRequest request) {

log.debug("FlightCtl validate method start");

boolean pass = true;

if (DataValidator.isNull(request.getParameter("name"))) {

request.setAttribute("name",

PropertyReader.getValue("error.require", "Flight Name"));

pass = false;

}

if (DataValidator.isNull(request.getParameter("flightNo"))) {

request.setAttribute("flightNo",

PropertyReader.getValue("error.require", "Flight No"));

pass = false;

}

if (DataValidator.isNull(request.getParameter("fromCity"))) {

request.setAttribute("fromCity",

PropertyReader.getValue("error.require", "From City"));

pass = false;

}

if (DataValidator.isNull(request.getParameter("toCity"))) {

request.setAttribute("toCity",

PropertyReader.getValue("error.require", "To City"));

pass = false;

}

if (DataValidator.isNull(request.getParameter("date"))) {

request.setAttribute("date",

PropertyReader.getValue("error.require", "Date"));

pass = false;

}

if (DataValidator.isNull(request.getParameter("time"))) {

request.setAttribute("time",

PropertyReader.getValue("error.require", "Time"));

pass = false;

}

if (DataValidator.isNull(request.getParameter("duration"))) {

request.setAttribute("duration",

PropertyReader.getValue("error.require", "Travel Duration"));

pass = false;

}

if (DataValidator.isNull(request.getParameter("price"))) {

request.setAttribute("price",

PropertyReader.getValue("error.require", "Ticket Price"));

pass = false;

}

if (DataValidator.isNull(request.getParameter("airPort"))) {

request.setAttribute("airPort",

PropertyReader.getValue("error.require", "Air Port Name"));

pass = false;

}

if (DataValidator.isNull(request.getParameter("description"))) {

request.setAttribute("description",

PropertyReader.getValue("error.require", "Description"));

pass = false;

}

log.debug("FlightCtl validate method end");

return pass;

}

@Override

protected BaseBean populateBean(HttpServletRequest request) {

log.debug("FlightCtl populateBean method start");

FlightBean bean=new FlightBean();

bean.setId(DataUtility.getLong(request.getParameter("id")));

bean.setFightName(DataUtility.getString(request.getParameter("name")));

bean.setFlightNo(DataUtility.getString(request.getParameter("flightNo")));

bean.setToCity(DataUtility.getString(request.getParameter("toCity")));

bean.setFromCity(DataUtility.getString(request.getParameter("fromCity")));

bean.setTravelDuraion(DataUtility.getString(request.getParameter("duration")));

bean.setTicketPrice(DataUtility.getLong(request.getParameter("price")));

bean.setTime(DataUtility.getString(request.getParameter("time")));

bean.setAirPortName(DataUtility.getString(request.getParameter("airPort")));

bean.setDate(DataUtility.getDate(request.getParameter("date")));

bean.setDescription(DataUtility.getString(request.getParameter("description")));

populateDTO(bean, request);

log.debug("FlightCtl populateBean method end");

return bean;

}

/\*\*

\* @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse response)

\*/

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

log.debug("FlightCtl doGet method start");

String op = DataUtility.getString(request.getParameter("operation"));

FlightModel model = new FlightModel();

long id = DataUtility.getLong(request.getParameter("id"));

ServletUtility.setOpration("Add", request);

if (id > 0 || op != null) {

System.out.println("in id > 0 condition");

FlightBean bean;

try {

bean = model.findByPK(id);

ServletUtility.setOpration("Edit", request);

ServletUtility.setBean(bean, request);

} catch (ApplicationException e) {

ServletUtility.handleException(e, request, response);

return;

}

}

ServletUtility.forward(getView(), request, response);

log.debug("FlightCtl doGet method end");

}

/\*\*

\* @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse response)

\*/

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

log.debug("FlightCtl doPost method start");

String op=DataUtility.getString(request.getParameter("operation"));

FlightModel model=new FlightModel();

long id=DataUtility.getLong(request.getParameter("id"));

if(OP\_SAVE.equalsIgnoreCase(op)){

FlightBean bean=(FlightBean)populateBean(request);

try {

if(id>0){

model.update(bean);

ServletUtility.setOpration("Edit", request);

ServletUtility.setSuccessMessage("Data is successfully Updated", request);

ServletUtility.setBean(bean, request);

}else {

CityModel cModel=new CityModel();

CityBean cBean=cModel.findByName(bean.getFromCity());

if(cBean==null) {

CityBean ccBean=new CityBean();

ccBean.setName(bean.getFromCity());

cModel.add(ccBean);

}

long pk=model.add(bean);

//bean.setId(id);

ServletUtility.setSuccessMessage("Data is successfully Saved", request);

ServletUtility.forward(getView(), request, response);

}

} catch (ApplicationException e) {

e.printStackTrace();

ServletUtility.forward(ATBView.ERROR\_VIEW, request, response);

return;

} catch (DuplicateRecordException e) {

ServletUtility.setBean(bean, request);

ServletUtility.setErrorMessage(e.getMessage(),

request);

}

}else if (OP\_DELETE.equalsIgnoreCase(op)) {

FlightBean bean= (FlightBean)populateBean(request);

try {

model.delete(bean);

ServletUtility.redirect(ATBView.FLIGHT\_LIST\_CTL, request, response);

} catch (ApplicationException e) {

ServletUtility.handleException(e, request, response);

e.printStackTrace();

}

}else if (OP\_CANCEL.equalsIgnoreCase(op)) {

ServletUtility.redirect(ATBView.FLIGHT\_LIST\_CTL, request, response);

return;

}else if (OP\_RESET.equalsIgnoreCase(op)) {

ServletUtility.redirect(ATBView.FLIGHT\_CTL, request, response);

return;

}

ServletUtility.forward(getView(), request, response);

log.debug("FlightCtl doPost method end");

}

@Override

protected String getView() {

// TODO Auto-generated method stub

return ATBView.FLIGHT\_VIEW;

}

}

<%@page import=*"in.co.air.line.ticket.controller.FlightCtl"*%>

<%@page import=*"in.co.air.line.ticket.util.DataUtility"*%>

<%@page import=*"in.co.air.line.ticket.util.ServletUtility"*%>

<%@ page language=*"java"* contentType=*"text/html; charset=ISO-8859-1"*

pageEncoding=*"ISO-8859-1"*%>

<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">

<html >

<head>

<meta http-equiv=*"Content-Type"* content=*"text/html; charset=ISO-8859-1"*>

<title>Flight</title>

</head>

<body>

<%@ include file=*"Header.jsp"*%>

<div class=*"container"*>

<div id=*"signupbox"* style="margin-top: *50px*"

class=*"mainbox col-md-6 col-md-offset-3 col-sm-8 col-sm-offset-2"*>

<div class=*"panel panel-info"*>

<div class=*"panel-heading"*>

<div class=*"panel-title"*>Flight</div>

<b><font color=*"red"*> <%=ServletUtility.getErrorMessage(request)%>

</font></b> <b><font color=*"Green"*> <%=ServletUtility.getSuccessMessage(request)%>

</font></b>

</div>

<div class=*"panel-body"*>

<form method=*"post"* action=*"*<%=ATBView.FLIGHT\_CTL%>*"*>

<jsp:useBean id=*"bean"*

class=*"in.co.air.line.ticket.bean.FlightBean"* scope=*"request"*></jsp:useBean>

<input type=*"hidden"* name=*"id"* value=*"*<%=bean.getId()%>*"*>

<input type=*"hidden"* name=*"createdBy"*

value=*"*<%=bean.getCreatedBy()%>*"*> <input type=*"hidden"*

name=*"modifiedBy"* value=*"*<%=bean.getModifiedBy()%>*"*> <input

type=*"hidden"* name=*"createdDatetime"*

value=*"*<%=DataUtility.getTimestamp(bean.getCreatedDatetime())%>*"*>

<input type=*"hidden"* name=*"modifiedDatetime"*

value=*"*<%=DataUtility.getTimestamp(bean.getModifiedDatetime())%>*"*>

<div class=*"form-group required"*>

<label class=*"control-label col-md-4 requiredField"*>Flight

No.<span class=*"asteriskField"*>\*</span>

</label>

<div class=*"controls col-md-8 "*>

<input class=*"input-md textinput textInput form-control"*

maxlength=*"30"* name=*"flightNo"* placeholder=*"Enter Flight No."*

style="margin-bottom: *10px*" type=*"text"*

value=*"*<%=DataUtility.getStringData(bean.getFlightNo())%>*"* />

<font color=*"red"*><%=ServletUtility.getErrorMessage("flightNo", request)%></font>

</div>

</div>

<div class=*"form-group required"*>

<label class=*"control-label col-md-4 requiredField"*>Flight

Name<span class=*"asteriskField"*>\*</span>

</label>

<div class=*"controls col-md-8 "*>

<input class=*"input-md textinput textInput form-control"*

maxlength=*"30"* name=*"name"* placeholder=*"Enter Flight name"*

style="margin-bottom: *10px*" type=*"text"*

value=*"*<%=DataUtility.getStringData(bean.getFightName())%>*"* />

<font color=*"red"*><%=ServletUtility.getErrorMessage("name", request)%></font>

</div>

</div>

<div class=*"form-group required"*>

<label class=*"control-label col-md-4 requiredField"*>From

City <span class=*"asteriskField"*>\*</span>

</label>

<div class=*"controls col-md-8 "*>

<input class=*"input-md textinput textInput form-control"*

maxlength=*"30"* name=*"fromCity"* placeholder=*"Enter From City"*

style="margin-bottom: *10px*" type=*"text"*

value=*"*<%=DataUtility.getStringData(bean.getFromCity())%>*"* />

<font color=*"red"*><%=ServletUtility.getErrorMessage("fromCity", request)%></font>

</div>

</div>

<div class=*"form-group required"*>

<label class=*"control-label col-md-4 requiredField"*>To

City <span class=*"asteriskField"*>\*</span>

</label>

<div class=*"controls col-md-8 "*>

<input class=*"input-md textinput textInput form-control"*

maxlength=*"30"* name=*"toCity"* placeholder=*"Enter To City"*

style="margin-bottom: *10px*" type=*"text"*

value=*"*<%=DataUtility.getStringData(bean.getToCity())%>*"* /> <font

color=*"red"*><%=ServletUtility.getErrorMessage("toCity", request)%></font>

</div>

</div>

<div class=*"form-group required"*>

<label class=*"control-label col-md-4 requiredField"*>Date

<span class=*"asteriskField"*>\*</span>

</label>

<div class=*"controls col-md-8 "*>

<input class=*"input-md textinput textInput form-control"*

maxlength=*"30"* name=*"date"* placeholder=*"Enter Date"* readonly=*"readonly"*

style="margin-bottom: *10px*" type=*"text"* id=*"datepicker"*

value=*"*<%=DataUtility.getDateString(bean.getDate())%>*"* /> <font

color=*"red"*><%=ServletUtility.getErrorMessage("date", request)%></font>

</div>

</div>

<div class=*"form-group required"*>

<label class=*"control-label col-md-4 requiredField"*>Time

<span class=*"asteriskField"*>\*</span>

</label>

<div class=*"controls col-md-8 "*>

<input class=*"input-md textinput textInput form-control"*

maxlength=*"30"* name=*"time"* placeholder=*"Enter Time"*

style="margin-bottom: *10px*" type=*"text"*

value=*"*<%=DataUtility.getStringData(bean.getTime())%>*"* /> <font

color=*"red"*><%=ServletUtility.getErrorMessage("time", request)%></font>

</div>

</div>

<div class=*"form-group required"*>

<label class=*"control-label col-md-4 requiredField"*>Travel

Duration <span class=*"asteriskField"*>\*</span>

</label>

<div class=*"controls col-md-8 "*>

<input class=*"input-md textinput textInput form-control"*

maxlength=*"30"* name=*"duration"*

placeholder=*"Enter Travel Duration"* style="margin-bottom: *10px*"

type=*"text"*

value=*"*<%=DataUtility.getStringData(bean.getTravelDuraion())%>*"* />

<font color=*"red"*><%=ServletUtility.getErrorMessage("duration", request)%></font>

</div>

</div>

<div class=*"form-group required"*>

<label class=*"control-label col-md-4 requiredField"*>Airport

Name <span class=*"asteriskField"*>\*</span>

</label>

<div class=*"controls col-md-8 "*>

<input class=*"input-md textinput textInput form-control"*

maxlength=*"30"* name=*"airPort"* placeholder=*"Enter Airport Name"*

style="margin-bottom: *10px*" type=*"text"*

value=*"*<%=DataUtility.getStringData(bean.getAirPortName())%>*"* />

<font color=*"red"*><%=ServletUtility.getErrorMessage("airPort", request)%></font>

</div>

</div>

<div class=*"form-group required"*>

<label class=*"control-label col-md-4 requiredField"*>Ticket

Price <span class=*"asteriskField"*>\*</span>

</label>

<div class=*"controls col-md-8 "*>

<input class=*"input-md textinput textInput form-control"*

maxlength=*"30"* name=*"price"* placeholder=*"Enter Ticket Price"*

style="margin-bottom: *10px*" type=*"text"*

value=*"*<%=(bean.getTicketPrice()==0)?"":bean.getTicketPrice()%>*"* />

<font color=*"red"*><%=ServletUtility.getErrorMessage("price", request)%></font>

</div>

</div>

<div class=*"form-group required"*>

<label class=*"control-label col-md-4 requiredField"*>Description<span

class=*"asteriskField"*>\*</span>

</label>

<div class=*"controls col-md-8 "*>

<textarea rows=*"4"* cols=*"5"* name=*"description"*

placeholder=*"Your Decription "*

class=*"input-md textinput textInput form-control"*

style="margin-bottom: *10px*"><%=DataUtility.getStringData(bean.getDescription())%></textarea>

<font color=*"red"*> <%=ServletUtility.getErrorMessage("description", request)%></font>

</div>

</div>

<div class=*"form-group"*>

<div class=*"aab controls col-md-4 "*></div>

<div class=*"controls col-md-8 "*>

<input type=*"submit"* name=*"operation"*

value=*"*<%=FlightCtl.OP\_SAVE%>*"*

class=*"btn btn-primary btn btn-info"* />

or <input type=*"submit"* name=*"operation"* value=*"*<%=FlightCtl.OP\_RESET %>*"* class=*"btn btn btn-primary"* />

</div>

</div>

</form>

</div>

</div>

</div>

</div>

<div style="margin-top: *198px*">

<%@ include file=*"Footer.jsp"*%>

</div>

</body>

</html>