FLYAWAY controller

Addfare Details

```
package com.flyaway.controller;

import java.io.IOException;

import javax.servlet.RequestDispatcher;
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 javax.servlet.http.HttpSession;

import com.flyaway.dao.AdminDAO;
import com.flyaway.model.Fare;

/**
 * Servlet implementation class AddFareDetails
 */
@WebServlet("/addfaredetails")
public class AddFareDetails extends HttpServlet {
	private static final long serialVersionUID = 1L;

	/**
	 * @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse response)
	 */
	protected void doPost(HttpServletRequest request, HttpServletResponse response) throws
ServletException, IOException {
```

```
RequestDispatcher rd;
boolean flag =  false;
HttpSession session = request.getSession(false);
String flightNumber = (String)session.getAttribute("flightnumber");
Integer conFlightNumber = 0;
String travelClass = request.getParameter("travelclass");
String fare = request.getParameter("fare");
Fare fareObj = new Fare();
AdminDAO admin = new AdminDAO();
String status = "";
double classFare = 0.00;
try {
        conFlightNumber = Integer.parseInt(flightNumber);
        classFare = Double.parseDouble(fare);

}catch(Exception e) {
        flag = true;
}



if((conFlightNumber != 0 && conFlightNumber != null) &&
                (travelClass != null && travelClass.trim() != "") &&
                (classFare != 0.00) && (flag == false)) {

        fareObj.setFlightNumber(conFlightNumber);
        fareObj.setTravelClass(travelClass);
        fareObj.setFare(classFare);

        status = admin.addFare(fareObj);
        if(status == "SUCCESS") {
                request.setAttribute("SUCCESS", "Fare Added Successfully");
                rd = getServletContext().getRequestDispatcher("/admindetails.jsp");
                rd.forward(request, response);

            }else if(status == "FAIL") {

                request.setAttribute("FAIL", "ERROR Occured while adding Fare");
                rd = getServletContext().getRequestDispatcher("/admindetails.jsp");
                rd.forward(request, response);


            }

        }else {

            request.setAttribute("FAIL", "ERROR Occured while adding Fare");
            rd = getServletContext().getRequestDispatcher("/admindetails.jsp");
            rd.forward(request, response);
        }

    }

}
```

AddfareDetails Submit

```
package com.flyaway.controller;

import java.io.IOException;
import javax.servlet.RequestDispatcher;
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 javax.servlet.http.HttpSession;

/**
 * Servlet implementation class AddFareDetailSubmit
 */
@WebServlet("/addfaredetailssubmit")
public class AddFareDetailsSubmit extends HttpServlet {
	private static final long serialVersionUID = 1L;

	/**
	 * @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse response)
	 */
	protected void doPost(HttpServletRequest request, HttpServletResponse response) throws
ServletException, IOException {
		RequestDispatcher rd;
		HttpSession session = request.getSession(false);
		String flightNumber = request.getParameter("flightnumber");
		if(flightNumber == null) {
			request.setAttribute("ERROR", "Cannot add Fare details");
			rd = getServletContext().getRequestDispatcher("/admindetails.jsp");
			rd.forward(request, response);
		}else if(flightNumber != null) {
			session.setAttribute("flightnumber", flightNumber);
			rd = getServletContext().getRequestDispatcher("/addfaredetails.jsp");
```

```
            rd.forward(request, response);

        }

    }

}
```

AdminAdd Flight

```
package com.flyaway.controller;

import java.io.IOException;

import javax.servlet.RequestDispatcher;
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 javax.servlet.http.HttpSession;

import com.flyaway.dao.AdminDAO;
import com.flyaway.model.Flight;

/**
 * Servlet implementation class AdminAddFlight
 */
@WebServlet("/addflightadmin")
public class AdminAddFlight extends HttpServlet {
    private static final long serialVersionUID = 1L;
	/**
	 * @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse response)
	 */
	@SuppressWarnings("unused")
	protected void doPost(HttpServletRequest request, HttpServletResponse response) throws 
ServletException, IOException {
		RequestDispatcher rd;
		String airline = request.getParameter("airline");
		String[] days = request.getParameterValues("weekdays");
		String weekdays = String.join("_", days);
		String source = request.getParameter("source");
		String destination = request.getParameter("destination");
		Flight flight = new Flight();
		int flightNumber = 0;
		AdminDAO admin = new AdminDAO();
		HttpSession session = request.getSession(false);
		if((airline != null && airline != "") && (weekdays != null && weekdays != "") &&
				(source != null && source != "") && (destination != null && 
destination != "")) {

			flight.setAirline(airline);
			flight.setWeekdays(weekdays);
			flight.setSource(source);
			flight.setDestination(destination);
			flightNumber = admin.addFlight(flight);

			if(flightNumber != 0) {
				request.setAttribute("SUCCESS", "Flight successfully added");
				rd = getServletContext().getRequestDispatcher("/admindetails.jsp");
				rd.forward(request, response);
			}else {
```

```
                request.setAttribute("Error", "Error Occured while adding flight");
                rd =
getServletContext().getRequestDispatcher("/adminaddflight.jsp");
                rd.forward(request, response);
            }



        }else {
            request.setAttribute("Error2", "Error Occured while adding flight");
            rd = getServletContext().getRequestDispatcher("/adminaddflight.jsp");
            rd.forward(request, response);
        }



    }

}
```

Admin Login

```
package com.flyaway.controller;

import java.io.IOException;

import javax.servlet.RequestDispatcher;
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 javax.servlet.http.HttpSession;
import com.flyaway.dao.AdminDAO;

/**
 * Servlet implementation class AdminLogin
 */
@WebServlet("/adminlogin")
public class AdminLogin extends HttpServlet {
    private static final long serialVersionUID = 1L;

    /**
     * @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse response)
     */
    protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

        String email = request.getParameter("emailaddress");
        String password = request.getParameter("password");

        RequestDispatcher rd;

        AdminDAO admin = new AdminDAO();

        if((email != null && email.trim() != "") &&
                (password != null && password.trim() != "")) {


            int adminId = admin.adminLogin(email, password);

            if(adminId != 0) {
```

```
                                        HttpSession session = request.getSession();
                                        session.setAttribute("adminId", adminId);
                                        rd = getServletContext().getRequestDispatcher("/admindetails.jsp");
                    rd.forward(request, response);
                              }else {

                                        request.setAttribute("loginerr", "Incorrect email or Password");
                                        rd = getServletContext().getRequestDispatcher("/adminlogin.jsp");
                    rd.forward(request, response);
                              }

                      }else {

                              request.setAttribute("loginerr1", "Error Occurred while Loging in");
                              rd = getServletContext().getRequestDispatcher("/adminlogin.jsp");
          rd.forward(request, response);
                    }



           }

}
```

Admin Update Password

```
package com.flyaway.controller;

import java.io.IOException;

import javax.servlet.RequestDispatcher;
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 javax.servlet.http.HttpSession;


import com.flyaway.dao.AdminDAO;


/**

 * Servlet implementation class AdminUpdatePassword

 */

@WebServlet("/adminupdatepassword")

public class AdminUpdatePassword extends HttpServlet {

	private static final long serialVersionUID = 1L;


	/**

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

	 */

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


		RequestDispatcher rd;

		HttpSession session = request.getSession(false);

		Integer adminId = (Integer)session.getAttribute("adminId");

		AdminDAO admin = new AdminDAO();

		String status = "";


		if(adminId != null) {


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


			if(password != null && password.trim() != "") {
					status = admin.updatePasswordAdmin(adminId, password);

					if(status == "SUCCESS") {

						request.setAttribute("SUCCESS", "Password Successfully
Updated");

						rd =
getServletContext().getRequestDispatcher("/admindetails.jsp");

						rd.forward(request, response);

					}else if(status == "FAIL") {


						request.setAttribute("FAIL", "Error while Updating
Password");

						rd =
getServletContext().getRequestDispatcher("/admindetails.jsp");

						rd.forward(request, response);

					}

				}else {


					request.setAttribute("FAIL1", "Error while Updating Password");

					rd = getServletContext().getRequestDispatcher("/admindetails.jsp");

					rd.forward(request, response);


				}

			}





		}


}
```

Confirm Booking

```
package com.flyaway.controller;

import java.io.IOException;

import javax.servlet.RequestDispatcher;
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 javax.servlet.http.HttpSession;

/**
 * Servlet implementation class ConfirmBooking
 */
@WebServlet("/confirmbooking")
public class ConfirmBooking extends HttpServlet {
	private static final long serialVersionUID = 1L;

	/**
	 * @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse response)
	 */
	protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

		HttpSession session = request.getSession(false);
		if(session != null) {
			RequestDispatcher rd = getServletContext().getRequestDispatcher("/payment.jsp");
			rd.forward(request, response);

		}else {
```

```
                              request.setAttribute("Error", "Error Occured while confirming ticket");
                              RequestDispatcher rd =
getServletContext().getRequestDispatcher("/confirmbooking.jsp");
                              rd.forward(request, response);
                    }

          }

}
```

Confirm payment Java

```
package com.flyaway.controller;

import java.io.IOException;

import javax.servlet.RequestDispatcher;
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 javax.servlet.http.HttpSession;

import com.flyaway.dao.CustomerDAO;
import com.flyaway.model.Fare;
import com.flyaway.model.Flight;
import com.flyaway.model.Reservation;

/**
 * Servlet implementation class ConfirmPayment
 */
@WebServlet("/confirmpayment")
```

```
public class ConfirmPayment extends HttpServlet {
    private static final long serialVersionUID = 1L;

    /**
     * @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse response)
     */
    @SuppressWarnings("unused")
    protected void doPost(HttpServletRequest request, HttpServletResponse response) throws
ServletException, IOException {

        HttpSession session = request.getSession(false);
        String cardName = request.getParameter("cardname");
        String cardNo = request.getParameter("cardno");
        String date = request.getParameter("paymentdate");

        if(session != null && (cardName != null && cardName.trim() != "") && (cardNo !=
null && cardNo.trim() != "" )
                && date != null) {

            session.setAttribute("cardname", cardName);
            session.setAttribute("cardno", cardNo);
            session.setAttribute("paymentdate", date);

            Flight flight = (Flight)session.getAttribute("flightobject");
            Fare fare = (Fare)session.getAttribute("fareobject");
            String travelDate = (String)session.getAttribute("traveldate");
            int passengers = (int)session.getAttribute("passengers");
            String day = (String)session.getAttribute("day");
            int customerId = (int)session.getAttribute("customerId");
            int bookingId = 0;

            Reservation reservation = new Reservation();
}

            CustomerDAO cust = new CustomerDAO();
            double totalFare = cust.calculateFare(flight.getFlightNumber(),
                                fare.getTravelClass() , passengers);

            reservation.setFlightNumber(flight.getFlightNumber());
            reservation.setTravelClass(fare.getTravelClass());
            reservation.setTravelDate(cust.getDate(travelDate));
            reservation.setPassengers(passengers);
            reservation.setTotalFare(totalFare);
            reservation.setCustomerId(customerId);

            bookingId = cust.addReservation(reservation);

            if(bookingId != 0) {
                session.setAttribute("bookingId", bookingId);
                RequestDispatcher rd = 
getServletContext().getRequestDispatcher("/bookingdetails.jsp");
                rd.forward(request, response);
            }
        }else {
            request.setAttribute("Error", "Error in processing payment please try again 
later");
            RequestDispatcher rd = 
getServletContext().getRequestDispatcher("/payment.jsp");
            rd.forward(request, response);
        }

    }

}
```

Login Java

```
package com.flyaway.controller;

import java.io.IOException;
import java.util.Enumeration;
import java.util.HashMap;
import java.util.Map;

import javax.servlet.RequestDispatcher;
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 javax.servlet.http.HttpSession;

import com.flyaway.dao.CustomerDAO;
import com.flyaway.model.Flight;

/**
 * Servlet implementation class Login
 */
@WebServlet("/login")
public class Login extends HttpServlet {
	private static final long serialVersionUID = 1L;

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

	protected void doPost(HttpServletRequest request, HttpServletResponse response) throws
ServletException, IOException {
        String email = request.getParameter("emailaddress");
        String password = request.getParameter("password");

        RequestDispatcher rd;
        CustomerDAO cust = new CustomerDAO();
        if((email != null && email.trim() != "") && (password != null && password.trim() !=
"")) {
            int customerId = cust.customerLogin(email, password);
            if(customerId != 0) {

                HashMap<String, Object> map = new HashMap<String, Object>();
                HttpSession oldSession = request.getSession(false);
                HttpSession newSession = null;
                if (oldSession != null) {
                    Enumeration keys = oldSession.getAttributeNames();
                    while(keys.hasMoreElements()) {
                        String key = (String)keys.nextElement();
                        map.put(key, oldSession.getAttribute(key));
                        oldSession.removeAttribute(key);

                    }

                    oldSession.invalidate();
                    newSession = request.getSession();
                    for(Map.Entry<String , Object> m : map.entrySet()) {

                        newSession.setAttribute((String)m.getKey(),
m.getValue());

                        map.remove(m);
                    }
```

```
                                    }else if(oldSession == null)  {

                                              newSession = request.getSession();
                                    }



                                    Flight flight = (Flight)newSession.getAttribute("flightobject");
                                    if (flight == null) {

                                              newSession.setAttribute("customerId", customerId);
                                              rd = 
getServletContext().getRequestDispatcher("/customerdetails.jsp");
                                              rd.forward(request, response);

                                    }else {
                                              newSession.setAttribute("customerId", customerId);
                                              rd = 
getServletContext().getRequestDispatcher("/confirmbooking.jsp");
                                              rd.forward(request, response);

                                    }




                          }else {
                                    request.setAttribute("loginerr", "Incorrect email or Password");
                                    rd = getServletContext().getRequestDispatcher("/login.jsp");
                                    rd.forward(request, response);

                          }
```

```
        }else {


            request.setAttribute("loginerr1", "Error Occured while login in");
            rd = getServletContext().getRequestDispatcher("/login.jsp");
            rd.forward(request, response);


        }





    }

}
```

Logout Java

```
package com.flyaway.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;

/**
 * Servlet implementation class Logout
 */
@WebServlet("/Logout")
public class Logout extends HttpServlet {
    private static final long serialVersionUID = 1L;
	@Override
	protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

		if(request.getSession().getAttribute("customerId")!=null)
		{
			request.getSession().invalidate();

			response.sendRedirect("index.jsp");
		}else if(request.getSession().getAttribute("adminId")!=null) {
			request.getSession().invalidate();

			response.sendRedirect("index.jsp");
		}else {

			request.getSession().invalidate();

			response.sendRedirect("index.jsp");

		}
	}



}
```

Register Java

```
package com.flyaway.controller;
```

```
import java.io.IOException;
import java.util.Enumeration;
import java.util.HashMap;
import java.util.Map;

import javax.servlet.RequestDispatcher;
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 javax.servlet.http.HttpSession;

import com.flyaway.dao.CustomerDAO;
import com.flyaway.model.Customer;

/**
 * Servlet implementation class Register
 */
@WebServlet("/register")
public class Register extends HttpServlet {
	private static final long serialVersionUID = 1L;

	/**
	 * @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse response)
	 */
	protected void doPost(HttpServletRequest request, HttpServletResponse response) throws
ServletException, IOException {

		HashMap<String, Object> map = new HashMap<String, Object>();
		HttpSession oldSession = request.getSession(false);
HttpSession newSession = null;
if (oldSession != null) {
        Enumeration<?> keys = oldSession.getAttributeNames();
        while(keys.hasMoreElements()) {
                String key = (String)keys.nextElement();
                map.put(key, oldSession.getAttribute(key));
                oldSession.removeAttribute(key);
        }

        oldSession.invalidate();
        newSession = request.getSession();
        for(Map.Entry<String , Object> m : map.entrySet()) {

                newSession.setAttribute((String)m.getKey(), m.getValue());
                map.remove(m);
        }

}else if(oldSession == null)  {

        newSession = request.getSession();
}

String firstName = request.getParameter("firstname");
String lastName = request.getParameter("lastname");
String email = request.getParameter("emailaddress");
String password = request.getParameter("password");
String phone = request.getParameter("phone");
int customerId = 0;
CustomerDAO cust = new CustomerDAO();
Customer customer = new Customer();
```

```
        if(firstName.trim() != "" && lastName.trim() != "" && email.trim() != ""
                && password.trim() != "" && phone.trim() != ""  ) {
            customer.setFirstName(firstName);
            customer.setLastName(lastName);
            customer.setEmail(email);
            customer.setPassword(password);
            customer.setPhone(phone);
            customerId = cust.addCustomer(customer);
            System.out.println(customerId);

            if(customerId != 0) {
                newSession.setAttribute("customerId", customerId);
                RequestDispatcher rd =
getServletContext().getRequestDispatcher("/confirmbooking.jsp");
                rd.forward(request, response);
            }else {
                request.setAttribute("Error", "Error Occured while adding
customer");
                RequestDispatcher rd =
getServletContext().getRequestDispatcher("/register.jsp");
                rd.forward(request, response);


            }



        }else {
            request.setAttribute("Error1", "Error Occured while adding customer");
            RequestDispatcher rd =
getServletContext().getRequestDispatcher("/register.jsp");
            rd.forward(request, response);


        }
```

```
            }

}
```

Show fare Details

```
package com.flyaway.controller;

import java.io.IOException;
import java.util.ArrayList;
import java.util.List;

import javax.servlet.RequestDispatcher;
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 javax.servlet.http.HttpSession;

import com.flyaway.dao.CustomerDAO;
import com.flyaway.model.Fare;
import com.flyaway.model.Flight;

/**
 * Servlet implementation class ShowFareDetails
 */
@WebServlet("/showfaredetails")
public class ShowFareDetails extends HttpServlet {
    private static final long serialVersionUID = 1L;


    /**
     * @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse response)
     */
    protected void doPost(HttpServletRequest request, HttpServletResponse response) throws
ServletException, IOException {
        CustomerDAO cust = new CustomerDAO();
        String flightNumberStr;
        int flightNumber = 0;
        flightNumberStr = request.getParameter("flightnumber");
        if(flightNumberStr != null && flightNumberStr != "") {
            flightNumber = Integer.parseInt(flightNumberStr);
        }
        List<Fare> fareList = new ArrayList<>();
        Flight flight = new Flight();
        HttpSession session = request.getSession(false);
        if(flightNumber != 0) {
            flight = cust.getFlight(flightNumber);
            fareList = cust.showFareList(flight.getFlightNumber());
            request.setAttribute("farelist", fareList);
            session.setAttribute("flightobject", flight);


            if(fareList.size() == 0 || fareList == null) {


                request.setAttribute("FAIL", "There are no fare list available. Cannot
Book Tickets for " + flightNumber);
                RequestDispatcher rd =
getServletContext().getRequestDispatcher("/flightdetails.jsp");
```

```
                        rd.forward(request, response);

                    }else {

                        RequestDispatcher rd =
getServletContext().getRequestDispatcher("/faredetails.jsp");
                        rd.forward(request, response);
                    }



                }else {

                    request.setAttribute("FAIL1", "Error Occurred while fetching fares");
                    RequestDispatcher rd =
getServletContext().getRequestDispatcher("/flightdetails.jsp");
                    rd.forward(request, response);

                }




        }

}
```

Show Flight java

```
package com.flyaway.controller;

import java.io.IOException;
import java.util.ArrayList;
import java.util.Date;
import java.util.List;

import javax.servlet.RequestDispatcher;
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 javax.servlet.http.HttpSession;

import com.flyaway.dao.CustomerDAO;
import com.flyaway.model.Airport;
import com.flyaway.model.Flight;

/**
 * Servlet implementation class ShowFlight
 */
@WebServlet("/bookflight")
public class ShowFlight extends HttpServlet {
	private static final long serialVersionUID = 1L;

	/**
	 * @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse response)
	 */
	protected void doPost(HttpServletRequest request, HttpServletResponse response) throws
ServletException, IOException {
		CustomerDAO cust = new CustomerDAO();

		Airport srcAirport  = null;
		Airport destAirport = null;
		String day = "";
```

```
        String source = request.getParameter("source");

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

        String date = request.getParameter("traveldate");

        Date travelDate = cust.getDate(date);

        int passengers = Integer.parseInt(request.getParameter("passengers"));

        List<Flight> flightList = new ArrayList<>();

        if(source != "" && destination != "" && travelDate != null

                && passengers != 0 ){

            HttpSession session = request.getSession();

            srcAirport = cust.getAirportObject(source);

            destAirport = cust.getAirportObject(destination);

            day = cust.getDay(travelDate);

            flightList = cust.flightList(source, destination, travelDate);

            request.setAttribute("sourceairport", srcAirport);

            request.setAttribute("destairport", destAirport);

            request.setAttribute("flightlist", flightList);

            session.setAttribute("traveldate", date);

            session.setAttribute("passengers", passengers);

            session.setAttribute("day", day);




            if(flightList.size() == 0 || flightList == null) {

                request.setAttribute("FAIL", "There are no flights flying from " +
srcAirport.getAirport() + " to " + destAirport.getAirport() + " on date " + date + " ." );

                RequestDispatcher rd =
getServletContext().getRequestDispatcher("/bookflight.jsp");

                rd.forward(request, response);

            }else {

                RequestDispatcher rd =
getServletContext().getRequestDispatcher("/flightdetails.jsp");

                rd.forward(request, response);
```

```
			}



		}else {
			request.setAttribute("FAIL1", "Error Occurred while searching flights.");
			RequestDispatcher rd = 
getServletContext().getRequestDispatcher("/bookflight.jsp");
			rd.forward(request, response);



		}


	}


}
```

Submit Java

```
package com.flyaway.controller;


import java.io.IOException;


import javax.servlet.RequestDispatcher;
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 javax.servlet.http.HttpSession;


import com.flyaway.dao.CustomerDAO;
import com.flyaway.model.Fare;

import com.flyaway.model.Flight;


/**

 * Servlet implementation class Submit

 */

@WebServlet("/submit")

public class Submit extends HttpServlet {

	private static final long serialVersionUID = 1L;


	/**

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

	 */

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

		RequestDispatcher rd;

		CustomerDAO cust = new CustomerDAO();


		String travelClass = request.getParameter("travelclass");

		HttpSession session = request.getSession(false);

		Flight flight = (Flight)session.getAttribute("flightobject");

		Fare fare = cust.getFareRecord(flight.getFlightNumber(), travelClass);

		session.setAttribute("fareobject", fare);


		Integer customerId = (Integer)session.getAttribute("customerId");


		if(customerId == null) {

		rd = getServletContext().getRequestDispatcher("/register.jsp");

    rd.forward(request, response);

		}else if(customerId != null) {

			rd = getServletContext().getRequestDispatcher("/confirmbooking.jsp");
```

```
            rd.forward(request, response);
                }
        }

}
```

Update password

```
package com.flyaway.controller;

import java.io.IOException;

import javax.servlet.RequestDispatcher;
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 javax.servlet.http.HttpSession;

import com.flyaway.dao.CustomerDAO;

/**
 * Servlet implementation class UpdatePassword
 */
@WebServlet("/updatepassword")
public class UpdatePassword extends HttpServlet {
        private static final long serialVersionUID = 1L;

        /**
```

```
     * @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse response)
     */
    protected void doPost(HttpServletRequest request, HttpServletResponse response) throws
ServletException, IOException {

        RequestDispatcher rd;
        HttpSession session = request.getSession(false);
        Integer customerId = (Integer)session.getAttribute("customerId");
        CustomerDAO cust = new CustomerDAO();
        String status = "";

        if(customerId != null) {

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

            if(password != null && password.trim() != "") {

                status = cust.updatePassword(customerId, password);

                if(status == "SUCCESS") {
                    request.setAttribute("SUCCESS", "Password Successfully
Updated");
                    rd =
getServletContext().getRequestDispatcher("/customerdetails.jsp");
                    rd.forward(request, response);
                }else if(status == "FAIL") {

                    request.setAttribute("FAIL", "Error while Updating
Password");
                    rd =
getServletContext().getRequestDispatcher("/customerdetails.jsp");
                    rd.forward(request, response);
                }
				}else {

					request.setAttribute("FAIL", "Error while Updating Password");
					rd = 
getServletContext().getRequestDispatcher("/customerdetails.jsp");
					rd.forward(request, response);

				}
			}



		}

}
```

Admin DAO

```
package com.flyaway.dao;

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

import com.flyaway.model.Admin;
import com.flyaway.model.Fare;
import com.flyaway.model.Flight;

public class AdminDAO {

	private Connection con = null;
```

```
    private PreparedStatement pst = null;

    public int addFlight(Flight flight) {

        int flightNumber = 0;
        String sql = "insert into flight (airline , weekdays , src_airport_code ,
dest_airport_code) "
                + " values(? , ? ,? , ?)";




        try {
            con = DBConnect.getConnection();
            System.out.println("...........................................");
            pst = con.prepareStatement(sql , Statement.RETURN_GENERATED_KEYS);
            pst.setString(1, flight.getAirline());
            pst.setString(2, flight.getWeekdays());
            pst.setString(3, flight.getSource());
            pst.setString(4, flight.getDestination());

            if (pst.executeUpdate() == 1) {
                ResultSet rs = pst.getGeneratedKeys();
                if(rs.next()) {
                    flightNumber = rs.getInt(1);
                }

            }else {
                flightNumber = 0;
            }

        } catch (SQLException e) {
```

```
        flightNumber = 0;
    }finally {
        try {
            con.close();
        }catch (Exception e) {
            e.printStackTrace();
        }

    }

    return flightNumber;
}


public String addFare(Fare fare) {

    String status = "";
    String sql = "insert into fare (flight_number , class , fare) "
            + " values(? , ? ,?)";



    try {
        con = DBConnect.getConnection();
        pst = con.prepareStatement(sql);
        pst.setInt(1, fare.getFlightNumber());
        pst.setString(2, fare.getTravelClass());
        pst.setDouble(3, fare.getFare());

        if (pst.executeUpdate() == 1) {
            status = "SUCCESS";
```

```
                }else {
                        status = "FAIL";
                }

        } catch (SQLException e) {
                status = "FAIL";
        }finally {
                try {
                        con.close();
                }catch (Exception e) {
                        e.printStackTrace();
                }

        }

        return status;

}




public String updatePasswordAdmin(int Id , String password) {
        String sql = "update admin set password=? WHERE admin_id = ?";
        String status = "";

        try {
                con = DBConnect.getConnection();
                pst = con.prepareStatement(sql);
                pst.setString(1, password);
        pst.setInt(2, Id);

        if (pst.executeUpdate() == 1) {
            status = "SUCCESS";
        } else
            status = "FAIL";
    } catch (Exception e) {
        status = "FAIL";
    }finally {
        try {
            con.close();
        }catch (Exception e) {
            e.printStackTrace();
        }

    }
    return status;
}



public int adminLogin(String email , String password) {

    String sql = "select * from admin where email = ? and password = ?";
    int adminId = 0;

    try {
        con = DBConnect.getConnection();
        pst = con.prepareStatement(sql);
        pst.setString(1, email);
        pst.setString(2, password);
        ResultSet rs = pst.executeQuery();
```

```
            if(rs.next()) {
                adminId = rs.getInt(1);
            }else {
                adminId = 0;
            }

        } catch (SQLException e) {
            adminId = 0;
        }finally {
            try {
                con.close();
            }catch (Exception e) {
                e.printStackTrace();
            }

        }

        return adminId;
    }

    public Admin getAdmin(int adminId) {

        String sql = "select * from admin where admin_id = ?";
        Admin admin = new Admin();

        try {
            con = DBConnect.getConnection();
            pst = con.prepareStatement(sql);
            pst.setInt(1, adminId);
            ResultSet rs = pst.executeQuery();
            rs.next();
```

```
                admin.setAdminId(rs.getInt(1));
                admin.setFirstName(rs.getString(2));
                admin.setLastName(rs.getString(3));
                admin.setEmail(rs.getString(4));
                admin.setPassword(rs.getString(5));

            } catch (SQLException e) {
                e.printStackTrace();
            }finally {
                try {
                    con.close();
                }catch (Exception e) {
                    e.printStackTrace();
                }

            }

            return admin;
        }

}
```

Customer DAO

```
package com.flyaway.dao;
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.text.ParseException;
import java.text.SimpleDateFormat;
import java.util.ArrayList;
import java.util.Date;
import java.util.List;
import com.flyaway.model.Airport;
import com.flyaway.model.Customer;
import com.flyaway.model.Fare;
import com.flyaway.model.Flight;
import com.flyaway.model.Reservation;
public class CustomerDAO {
private Connection con = null;
private PreparedStatement pst = null;
public int addCustomer(Customer custBean) {
int customerId = 0;
String sql = "insert into customer (first_name , last_name , email , password , phone)"
+ " values(? , ? , ? , ? , ?)";
try {
con = DBConnect.getConnection();
pst = con.prepareStatement(sql , Statement.RETURN_GENERATED_KEYS);
pst.setString(1, custBean.getFirstName());
pst.setString(2, custBean.getLastName());
pst.setString(3, custBean.getEmail());
pst.setString(4, custBean.getPassword());
pst.setString(5, custBean.getPhone());
if(pst.executeUpdate() == 1) {
ResultSet rs = pst.getGeneratedKeys();
```

```
if(rs.next()) {

customerId = rs.getInt(1);

}

}else {

customerId = 0;

}

} catch (SQLException e) {

customerId = 0;

}finally {

try {

con.close();

}catch (Exception e) {

e.printStackTrace();

}

}

return customerId;

}

public String updatePassword(int Id , String password) {

String sql = "update customer set password=? WHERE customer_id = ?";

String status = "";

try {

con = DBConnect.getConnection();

pst = con.prepareStatement(sql);

pst.setString(1, password);

pst.setInt(2, Id);

if (pst.executeUpdate() == 1) {

status = "SUCCESS";
```

```
} else
status = "FAIL";
} catch (Exception e) {
status = "FAIL";
}finally {
try {
con.close();
}catch (Exception e) {
e.printStackTrace();
}
}
return status;
}
public int customerLogin(String email , String password) {
String sql = "select * from customer where email = ? and password = ?";
int customerId = 0;
try {
con = DBConnect.getConnection();
pst = con.prepareStatement(sql);
pst.setString(1, email);
pst.setString(2, password);
ResultSet rs = pst.executeQuery();
if(rs.next()) {
customerId = rs.getInt(1);
}else {
customerId = 0;
}
```

```
} catch (SQLException e) {
customerId = 0;
}finally {
try {
con.close();
}catch (Exception e) {
e.printStackTrace();
}
}
return customerId;
}
public Customer getCustomer(int customerId) {
String sql = "select * from customer where customer_id = ?";
Customer customer = new Customer();
try {
con = DBConnect.getConnection();
pst = con.prepareStatement(sql);
pst.setInt(1, customerId);
ResultSet rs = pst.executeQuery();
rs.next();
customer.setCustomerId(rs.getInt(1));
customer.setFirstName(rs.getString(2));
customer.setLastName(rs.getString(3));
customer.setEmail(rs.getString(4));
customer.setPassword(rs.getString(5));
customer.setPhone(rs.getString(6));
} catch (SQLException e) {
```

```
e.printStackTrace();
}finally {
try {
con.close();
}catch (Exception e) {
e.printStackTrace();
}
}
return customer;
}
public int addReservation(Reservation resBean) {
int bookingId = 0;
String sql = "insert into reservations (flight_number,class,travel_date,"
+ "no_of_passengers,total_fare,customer_id) values(?,?,?,?,?,?)";
try {
con = DBConnect.getConnection();
pst = con.prepareStatement(sql , Statement.RETURN_GENERATED_KEYS);
pst.setInt(1, resBean.getFlightNumber());
pst.setString(2, resBean.getTravelClass());
pst.setDate(3, getSQLDate(resBean.getTravelDate()));
pst.setInt(4, resBean.getPassengers());
pst.setDouble(5, resBean.getTotalFare());
pst.setInt(6, resBean.getCustomerId());
if (pst.executeUpdate() == 1) {
ResultSet rs = pst.getGeneratedKeys();
if(rs.next()) {
bookingId = rs.getInt(1);
}

}else {

bookingId = 0;

}

} catch (SQLException e) {

bookingId = 0;

}finally {

try {

con.close();

}catch (Exception e) {

e.printStackTrace();

}

}

return bookingId;

}

public List<Reservation> showReservations(int customerId) {

List<Reservation> lrev = new ArrayList<>();

String sql = "select r.booking_id, r.flight_number, r.class, r.travel_date,"

+ " r.no_of_passengers , r.total_fare , r.customer_id from "

+ " reservations r where r.customer_id = ? ";

try {

con = DBConnect.getConnection();

pst = con.prepareStatement(sql);

pst.setInt(1, customerId);

ResultSet rs = pst.executeQuery();

while(rs.next()) {

Reservation rserv = new Reservation();
```

```
rserv.setBookingId(rs.getInt(1));

rserv.setFlightNumber(rs.getInt(2));

rserv.setTravelClass(rs.getString(3));

rserv.setTravelDate(rs.getDate(4));

rserv.setPassengers(rs.getInt(5));

rserv.setTotalFare(rs.getDouble(6));

rserv.setCustomerId(rs.getInt(7));

lrev.add(rserv);

}

} catch (SQLException e) {

e.printStackTrace();

}finally {

try {

con.close();

}catch (Exception e) {

e.printStackTrace();

}

}

return lrev;

}

public Reservation getReservation(int bookingId ) {

Reservation reservation = new Reservation();

String sql = "select r.booking_id, r.flight_number, r.class, r.travel_date,"

+ " r.no_of_passengers , r.total_fare , r.customer_id from "

+ " reservations r where r.booking_id = ? ";

try {

con = DBConnect.getConnection();
```

```java
pst = con.prepareStatement(sql);
pst.setInt(1, bookingId);
ResultSet rs = pst.executeQuery();
rs.next();
reservation.setBookingId(rs.getInt(1));
reservation.setFlightNumber(rs.getInt(2));
reservation.setTravelClass(rs.getString(3));
reservation.setTravelDate(rs.getDate(4));
reservation.setPassengers(rs.getInt(5));
reservation.setTotalFare(rs.getDouble(6));
reservation.setCustomerId(rs.getInt(7));
} catch (SQLException e) {
e.printStackTrace();
}finally {
try {
con.close();
}catch (Exception e) {
e.printStackTrace();
}
}
return reservation;
}
public double getFare(int flightNumber , String travelClass) {
double fare = 0.0;
String sql = "select fare from fare where flight_number = ? and class = ?";
try {
con = DBConnect.getConnection();
```

```
pst = con.prepareStatement(sql);

pst.setInt(1, flightNumber);

pst.setString(2,travelClass);

ResultSet st = pst.executeQuery();

st.next();

fare = st.getDouble(1);

} catch (SQLException e) {

e.printStackTrace();

System.out.println("Cannot find fare");

}finally {

try {

con.close();

}catch (Exception e) {

e.printStackTrace();

}

}

return fare;

}

public double calculateFare(int flightNumber , String travelClass , int passengers) {

double totalFare = 0.0;

double fare = 0.0;

fare = getFare(flightNumber , travelClass);

totalFare = fare * passengers;

return totalFare;

}

public List<Flight>flightList(String src, String dest , Date travelDate){

List<Flight> flightList = new ArrayList<Flight>();
```

```
SimpleDateFormat sdf = new SimpleDateFormat("E");

String day = sdf.format(travelDate);

String sql = "select f.flight_number , f.airline , f.weekdays , f.src_airport_code , "

+ " f.dest_airport_code from flight f where"

+ " f.src_airport_code = ? and"

+ " f.dest_airport_code = ? and (find_in_set(? , replace(f.weekdays , '_' , ',')) > 0) ";

try {

con = DBConnect.getConnection();

pst = con.prepareStatement(sql);

pst.setString(1, src);

pst.setString(2, dest);

pst.setString(3, day);

ResultSet rs = pst.executeQuery();

while(rs.next()) {

Flight flight = new Flight();

flight.setFlightNumber(rs.getInt(1));

flight.setAirline(rs.getString(2));

flight.setWeekdays(rs.getString(3));

flight.setSource(rs.getString(4));

flight.setDestination(rs.getString(5));

flightList.add(flight);

}

} catch (SQLException e) {

e.printStackTrace();

}finally {

try {
```

```
con.close();
}catch (Exception e) {
e.printStackTrace();
}
}
return flightList;
}
public List<Flight>showFlightList(){
List<Flight> flightList = new ArrayList<Flight>();
String sql = "select f.flight_number , f.airline , f.weekdays , f.src_airport_code , "
+ " f.dest_airport_code from flight f ";
try {
con = DBConnect.getConnection();
pst = con.prepareStatement(sql);
ResultSet rs = pst.executeQuery();
while(rs.next()) {
Flight flight = new Flight();
flight.setFlightNumber(rs.getInt(1));
flight.setAirline(rs.getString(2));
flight.setWeekdays(rs.getString(3));
flight.setSource(rs.getString(4));
flight.setDestination(rs.getString(5));
flightList.add(flight);
}
} catch (SQLException e) {
e.printStackTrace();
}finally {
```

```
try {
con.close();
}catch (Exception e) {
e.printStackTrace();
}
}
return flightList;
}
public Flight getFlight(int flightNumber){
Flight flight = new Flight();
String sql = "select f.flight_number , f.airline , f.weekdays , f.src_airport_code , "
+ " f.dest_airport_code from flight f where"
+ " f.flight_number = ? ";
try {
con = DBConnect.getConnection();
pst = con.prepareStatement(sql);
pst.setInt(1, flightNumber);
ResultSet rs = pst.executeQuery();
rs.next();
flight.setFlightNumber(rs.getInt(1));
flight.setAirline(rs.getString(2));
flight.setWeekdays(rs.getString(3));
flight.setSource(rs.getString(4));
flight.setDestination(rs.getString(5));
} catch (SQLException e) {
e.printStackTrace();
}finally {
```

```
try {

con.close();

}catch (Exception e) {

e.printStackTrace();

}

}

return flight;

}

public List<Fare>showFareList(int flightNumber){

List<Fare> fareList = new ArrayList<>();

String sql = "select * from fare where flight_number = ?";

try {

con = DBConnect.getConnection();

pst = con.prepareStatement(sql);

pst.setInt(1, flightNumber);

ResultSet rs = pst.executeQuery();

while(rs.next()) {

Fare fare = new Fare();

fare.setFlightNumber(rs.getInt(1));

fare.setTravelClass(rs.getString(2));

fare.setFare(rs.getDouble(3));

fareList.add(fare);

}

} catch (SQLException e) {

e.printStackTrace();

}finally {

try {
```

```
con.close();

}catch (Exception e) {

e.printStackTrace();

}

}

return fareList;

}

public Fare getFareRecord(int flightNumber , String travelClass){

Fare fare = new Fare();

String sql = "select * from fare where flight_number = ? and class = ?";

try {

con = DBConnect.getConnection();

pst = con.prepareStatement(sql);

pst.setInt(1, flightNumber);

pst.setString(2, travelClass);

ResultSet rs = pst.executeQuery();

rs.next();

fare.setFlightNumber(rs.getInt(1));

fare.setTravelClass(rs.getString(2));

fare.setFare(rs.getDouble(3));

} catch (SQLException e) {

e.printStackTrace();

}finally {

try {

con.close();

}catch (Exception e) {

e.printStackTrace();
```

```java
}
}
return fare;
}
public String getAirport(String airportCode) {
String airport = "";
String sql = "select a.airport from airport a where a.airport_code = ?";
try {
con = DBConnect.getConnection();
pst = con.prepareStatement(sql);
pst.setString(1, airportCode);
ResultSet rs = pst.executeQuery();
rs.next();
airport = rs.getString(1);
} catch (SQLException e) {
e.printStackTrace();
}finally {
try {
con.close();
}catch (Exception e) {
e.printStackTrace();
}
}
return airport;
}
public String getCountryCode(String airportCode) {
String countryCode = "";
String sql = "select a.country_code from airport a where a.airport_code = ?";

try {

con = DBConnect.getConnection();

pst = con.prepareStatement(sql);

pst.setString(1, airportCode);

ResultSet rs = pst.executeQuery();

rs.next();

countryCode = rs.getString(1);

} catch (SQLException e) {

e.printStackTrace();

}

return countryCode;

}

public String getCountry(String airportCode) {

String country = "";

String sql = "select a.country from airport a where a.airport_code = ?";

try {

con = DBConnect.getConnection();

pst = con.prepareStatement(sql);

pst.setString(1, airportCode);

ResultSet rs = pst.executeQuery();

rs.next();

country = rs.getString(1);

} catch (SQLException e) {

e.printStackTrace();

}finally {

try {
```

```
con.close();

}catch (Exception e) {

e.printStackTrace();

}

}

return country;

}

public Date getDate(String date) {

Date theDate = new Date();

try {

theDate = DateUtils.parseDate(date);

} catch (ParseException e) {

e.printStackTrace();

}

return theDate;

}

public java.sql.Date getSQLDate(Date date){

return new java.sql.Date(date.getTime());

}

public List<Airport> listAirport(){

List<Airport> airportList = new ArrayList<>();

String sql = "select * from airport";

try {

con = DBConnect.getConnection();

pst = con.prepareStatement(sql);

ResultSet rs = pst.executeQuery();

while(rs.next()){
```

```
Airport airport = new Airport();

airport.setAirportCode(rs.getString(1));

airport.setAirport(rs.getString(2));

airport.setCountryCode(rs.getString(3));

airport.setCountry(rs.getString(4));

airportList.add(airport);

}

} catch (SQLException e) {

e.printStackTrace();

}finally {

try {

con.close();

}catch (Exception e) {

e.printStackTrace();

}

}

return airportList;

}

public List<String> listAirportCode(){

List<String> codeList = new ArrayList<>();

String sql = "select airport_code from airport";

try {

con = DBConnect.getConnection();

pst = con.prepareStatement(sql);

ResultSet rs = pst.executeQuery();

while(rs.next()){

String code = rs.getString(1);
```

```
codeList.add(code);

}

} catch (SQLException e) {

e.printStackTrace();

}finally {

try {

con.close();

}catch (Exception e) {

e.printStackTrace();

}

}

return codeList;

}

public List<String> listAirline(){

List<String> airList = new ArrayList<>();

String sql = "select * from airline";

try {

con = DBConnect.getConnection();

pst = con.prepareStatement(sql);

ResultSet rs = pst.executeQuery();

while(rs.next()){

String airline = rs.getString(1);

airList.add(airline);

}

} catch (SQLException e) {

e.printStackTrace();

}finally {
```

```
try {
con.close();
}catch (Exception e) {
e.printStackTrace();
}
}
return airList;
}
public Airport getAirportObject(String airportCode){
String sql = "select * from airport where airport_code = ?";
Airport airport = new Airport();
try {
con = DBConnect.getConnection();
System.out.println("flag ------> "+con);
pst = con.prepareStatement(sql);
pst.setString(1, airportCode);
ResultSet rs = pst.executeQuery();
rs.next();
airport.setAirportCode(rs.getString(1));
airport.setAirport(rs.getString(2));
airport.setCountryCode(rs.getString(3));
airport.setCountry(rs.getString(4));
} catch (SQLException e) {
e.printStackTrace();
}finally {
try {
con.close();
}catch (Exception e) {

e.printStackTrace();

}

}

return airport;

}

public String getDay(Date theDate) {

SimpleDateFormat sdf = new SimpleDateFormat("EEEE");

String day = sdf.format(theDate);

return day;

}

}
```

Date Utils

```
package com.flyaway.dao;

import java.text.ParseException;
import java.text.SimpleDateFormat;
import java.util.Date;

public class DateUtils {

    private static SimpleDateFormat formatter = new SimpleDateFormat("yyyy-MM-dd");

    // read a date string and parse/convert to a date
    public static Date parseDate(String dateStr) throws ParseException {
        Date theDate = formatter.parse(dateStr);
        return theDate;
    }

    // read a date and format/convert to a string
    public static String formatDate(Date theDate) {

        String result = null;

        if (theDate != null) {
            result = formatter.format(theDate);
        }

        return result;
    }

}
```

DBC connect java

```
package com.flyaway.dao;

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;

public class DBConnect {

    public static Connection getConnection() throws SQLException {

        Connection con = null;
        String jdbcUrl = 
"jdbc:mysql://localhost:3306/flight_reservation_tracker?useSSL=false";
        String driver = "com.mysql.cj.jdbc.Driver";
        String user = "root";
        String password = "Sujisk@0096";



        try
        {
            Class.forName(driver);
            con = DriverManager.getConnection(jdbcUrl, user, password);
            System.out.println("con");

        }catch(Exception e)
        {

            e.printStackTrace();
        }

        return con;
    }



//  public static void main(String[] args) {
//
//          getConnection();
//          System.out.println("Connection Established");
//
//  }
```

```
}
```

Utils java

```
package com.flyaway.dao;

import java.util.ArrayList;
import java.util.LinkedHashMap;
import java.util.List;
import java.util.Map;

public class Utils {

    public static Map<String,String> getWeekDays() {

        Map<String,String>weekMap = new LinkedHashMap<>();
        weekMap.put("Sun", "Sunday");
        weekMap.put("Mon", "Monday");
        weekMap.put("Tue", "Tuesday");
        weekMap.put("Wed", "Wednesday");
        weekMap.put("Thu", "Thursday");
        weekMap.put("Fri", "Friday");
        weekMap.put("Sat", "Saturday");

        return weekMap;

    }




    public static List<String> getClasses(){
        List<String> classList = new ArrayList<>();
        classList.add("Economy");
        classList.add("Premium");
        classList.add("Business");

        return classList;
    }

}
```

ADMIN

```
package com.flyaway.model;
public class Admin {
private int adminId;
private String firstName;
private String lastName;
private String email;
private String password;
public int getAdminId() {
return adminId;
}
public void setAdminId(int adminId) {
this.adminId = adminId;
}
public String getFirstName() {
return firstName;
}
public void setFirstName(String firstName) {

this.firstName = firstName;

}

public String getLastName() {

return lastName;

}

public void setLastName(String lastName) {

this.lastName = lastName;

}

public String getEmail() {

return email;

}

public void setEmail(String email) {

this.email = email;

}

public String getPassword() {

return password;

}

public void setPassword(String password) {

this.password = password;

}

@Override

public String toString() {

return "Admin [adminId=" + adminId + ", firstName=" + firstName + ", lastName=" +
lastName + ", email=" + email

+ ", password=" + password + "]";

}

}
```

AIRPORT

```
package com.flyaway.model;
public class Airport {
private String airportCode;
private String airport;
private String countryCode;
private String country;
public String getAirportCode() {
return airportCode;
}
public void setAirportCode(String airportCode) {
this.airportCode = airportCode;
}
public String getAirport() {
return airport;
}
public void setAirport(String airport) {
this.airport = airport;
}
public String getCountryCode() {
return countryCode;
}
public void setCountryCode(String countryCode) {
this.countryCode = countryCode;
}
public String getCountry() {
return country;
}
public void setCountry(String country) {
this.country = country;
}
@Override
public String toString() {
return "Airport [airportCode=" + airportCode + ", airport=" + airport + ",
countryCode=" + countryCode
+ ", country=" + country + "]";
}
}
```

CUSTOMER

```
package com.flyaway.model;
public class Customer {
private int customerId;
private String firstName;
private String lastName;
private String email;
private String password;
private String phone;
public int getCustomerId() {
return customerId;
}
public void setCustomerId(int customerId) {
```

```java
this.customerId = customerId;
}
public String getFirstName() {
return firstName;
}
public void setFirstName(String firstName) {
this.firstName = firstName;
}
public String getLastName() {
return lastName;
}
public void setLastName(String lastName) {
this.lastName = lastName;
}
public String getEmail() {
return email;
}
public void setEmail(String email) {
this.email = email;
}
public String getPassword() {
return password;
}
public void setPassword(String password) {
this.password = password;
}
public String getPhone() {
```

```java
return phone;

}

public void setPhone(String phone) {

this.phone = phone;

}

@Override

public String toString() {

return "Customer [customerId=" + customerId + ", firstName=" + firstName + ",
lastName=" + lastName + ", email="

+ email + ", password=" + password + ", phone=" + phone + "]";

}

}
```

```java
package com.flyaway.model;

public class Fare {

private int flightNumber;

private String travelClass;

private double fare;

public int getFlightNumber() {

return flightNumber;

}

public void setFlightNumber(int flightNumber) {

this.flightNumber = flightNumber;

}

public String getTravelClass() {

return travelClass;

}

public void setTravelClass(String travelClass) {
```

```
this.travelClass = travelClass;
}
public double getFare() {
return fare;
}
public void setFare(double fare) {
this.fare = fare;
}
@Override
public String toString() {
return "Fare [flightNumber=" + flightNumber + ", travelClass=" + travelClass + ", fare=" + fare + "]";
}
}
```

FLIGHT

```
package com.flyaway.model;
public class Flight {
private int flightNumber;
private String airline;
private String weekdays;
private String source;
private String destination;
public int getFlightNumber() {
return flightNumber;
}
public void setFlightNumber(int flightNumber) {
```

```java
this.flightNumber = flightNumber;
}
public String getAirline() {
return airline;
}
public void setAirline(String airline) {
this.airline = airline;
}
public String getWeekdays() {
return weekdays;
}
public void setWeekdays(String weekdays) {
this.weekdays = weekdays;
}
public String getSource() {
return source;
}
public void setSource(String source) {
this.source = source;
}
public String getDestination() {
return destination;
}
public void setDestination(String destination) {
this.destination = destination;
}
@Override
```

```
public String toString() {

return "Flight [flightNumber=" + flightNumber + ", airline=" + airline + ",
weekdays=" + weekdays + ", source="

+ source + ", destination=" + destination + "]";

}

}
```

RESERVATION

```
package com.flyaway.model;

import java.util.Date;

public class Reservation {

private int bookingId;

private int flightNumber;

private String travelClass;

private Date travelDate;

private int passengers;

private double totalFare;

private int customerId;

public Reservation() {

super();

// TODO Auto-generated constructor stub

}

public Reservation(int flightNumber, String travelClass, Date travelDate, int
passengers, double totalFare,

int customerId) {

super();

this.flightNumber = flightNumber;
```

```java
this.travelClass = travelClass;
this.travelDate = travelDate;
this.passengers = passengers;
this.totalFare = totalFare;
this.customerId = customerId;
}
public int getBookingId() {
return bookingId;
}
public void setBookingId(int bookingId) {
this.bookingId = bookingId;
}
public int getFlightNumber() {
return flightNumber;
}
public void setFlightNumber(int flightNumber) {
this.flightNumber = flightNumber;
}
public String getTravelClass() {
return travelClass;
}
public void setTravelClass(String travelClass) {
this.travelClass = travelClass;
}
public Date getTravelDate() {
return travelDate;
}
```

```
public void setTravelDate(Date travelDate) {

this.travelDate = travelDate;

}

public int getPassengers() {

return passengers;

}

public void setPassengers(int passengers) {

this.passengers = passengers;

}

public double getTotalFare() {

return totalFare;

}

public void setTotalFare(double totalFare) {

this.totalFare = totalFare;

}

public int getCustomerId() {

return customerId;

}

public void setCustomerId(int customerId) {

this.customerId = customerId;

}

@Override

public String toString() {

return "Reservation [bookingId=" + bookingId + ", flightNumber=" + flightNumber +
", travelClass=" + travelClass

+ ", travelDate=" + travelDate + ", passengers=" + passengers + ", totalFare=" +
totalFare

+ ", customerId=" + customerId + "]";
}

}
```