Final Project FlyAway Specifications

**FlyAway**

Class: Become a Back-end Expert

Student: Hector Alarcon

Application: LockedMe.com

Date: 10/1/2021

Contents

[Project Description: 3](#_Toc79327167)

[Java technologies utilized: 3](#_Toc79327168)

[Unique Selling Points 3](#_Toc79327169)

[Sprint breakdown 4](#_Toc79327170)

[Program Details: 10](#_Toc79327171)

[GitHub: 10](#_Toc79327172)

# Project Description:

* This project focused on making a web application

## Java technologies utilized:

* Access modifiers by using the private keyword for methods and variables that the application will need but the user won’t interact with.
* Static keywords for self-references since these are properties of the LockedMe app.
* Scanner was used for handling user input and closed using the finalize keyword at the end of the main method to make sure it closes properly.

Possible enhancements

* Once the app is available for more heavy-duty files, consider switch the array of files to an Array list to keep the fetching times constant.
* More modularity such as breaking down the second menu further into classes will help in future projects where these similar methods are used.

## Unique Selling Points

This excellent application provides the following features to the customer:

* Ease of use, all the menus have user interaction information displayed on the console describing all the options available to the user.
* Security, all of the class variables are set to private, and resources closed and only accessible to the class itself. Only letting the main method access the multiple resources.
* Bullet proof, sentinel values are used for input validation and making sure the user is aware of possible mistakes during inputting data.

# Sprint breakdown

|  |
| --- |
|  |
| Figure 1 PlantUML Sprint Schedule |

# Program Details:

## GitHub:

<https://github.com/ProgrammedPeinado/PracticeJava/tree/main/Simplilearn/Final%20Projects/flyaway>

|  |
| --- |
| Project Folder |
|  |
| AdminServlet |
| package com.controllers;  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 com.DAO.AdminDAO;  import com.DAO.AdminDaoImpl;  import com.dto.Admin;  /\*\*  \* Servlet implementation class AdminServlet  \*/  @WebServlet("/AdminServlet")  public class AdminServlet extends HttpServlet {  private static final long serialVersionUID = 1L;    /\*\*  \* @see HttpServlet#HttpServlet()  \*/  public AdminServlet()  {  super();  }  /\*\*  \* @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse response)  \*/  protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {  response.sendRedirect("admin\_login.jsp");  }  /\*\*  \* @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse response)  \*/  protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException  {  AdminDAO adminCheck = new AdminDaoImpl();  String user = request.getParameter("user");  String pass = request.getParameter("pass");  Admin admin = adminCheck.searchAdminByUser(user);    try  {  if(user.equals(admin.getUser()) & pass.equals(admin.getPass()));  {  response.sendRedirect("admin\_page.jsp");  }  {  request.setAttribute("loginResult", true);  response.sendRedirect("admin\_login.jsp");  }  }  catch(Exception e)  {  System.out.println(e.getMessage());  }  }    } |

|  |
| --- |
| FlightList |
| package com.controllers;  import java.io.IOException;  import java.io.PrintWriter;  import java.sql.Connection;  import java.sql.PreparedStatement;  import java.sql.ResultSet;  import java.sql.SQLException;  import java.util.ArrayList;  import java.util.List;  import javax.servlet.ServletException;  import javax.servlet.http.HttpServlet;  import javax.servlet.http.HttpServletRequest;  import javax.servlet.http.HttpServletResponse;  import org.hibernate.SessionFactory;  import org.hibernate.boot.Metadata;  import org.hibernate.boot.MetadataSources;  import org.hibernate.boot.registry.StandardServiceRegistry;  import org.hibernate.boot.registry.StandardServiceRegistryBuilder;  import com.DAO.FlightDAO;  import com.DAO.FlightDaoImpl;  import com.dto.Flight;  import net.bytebuddy.description.type.TypeList.Generic;  /\*\*  \* Servlet implementation class FlightList  \*/  public class FlightList extends HttpServlet  {      private static final long serialVersionUID = 1L;    /\*\*  \* @see HttpServlet#HttpServlet()  \*/  public FlightList() {  super();  }  /\*\*  \* @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse response)  \*/  protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {  response.getWriter().append("Served at: ").append(request.getContextPath());  }  /\*\*  \* @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse response)  \*/  protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException  {  String src = request.getParameter("source\_country");  String dest = request.getParameter("destination\_country");  String date = request.getParameter("book\_date");  int pass = Integer.parseInt(request.getParameter("n\_pass"));    PrintWriter out = response.getWriter();  FlightDAO flightDAO = new FlightDaoImpl();    try  {  List<Flight> res = flightDAO.listFlights(src, dest, date, pass);  request.setAttribute("flightList", res);  }  catch(Exception e)  {  System.out.println("\n\nStack Trace:");  e.printStackTrace();  System.out.println("\n\nMessage:"+e.getMessage());  System.out.println("\n\nThere are no flights matching your search criterion.");  }  finally  {    request.getRequestDispatcher("flightbook.jsp").forward(request,response);  out.close();  }    }      } |

|  |
| --- |
| FlightServlet |
| package com.controllers;  import java.io.IOException;  import java.util.List;  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 com.DAO.FlightDAO;  import com.DAO.FlightDaoImpl;  import com.DAO.PlaneDAO;  import com.DAO.PlaneDaoImpl;  import com.dto.Flight;  import com.dto.Plane;  /\*\*  \* Servlet implementation class FlightServlet  \*/  @WebServlet("/FlightServlet")  public class FlightServlet extends HttpServlet {  private static final long serialVersionUID = 1L;    /\*\*  \* @see HttpServlet#HttpServlet()  \*/  public FlightServlet() {  super();  }  /\*\*  \* @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse response)  \*/  protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException  {  String adminOptions = request.getParameter("selection");  FlightDAO flightDAO = new FlightDaoImpl();  PlaneDAO planeDAO = new PlaneDaoImpl();  String source = null; //to identify the parametrized method  String dest = null; //  List<Flight> res = null;  List<Plane> resAir = null;    if(adminOptions ==null);  adminOptions = "none";    switch(adminOptions)  {  case "listSources":  {  res = flightDAO.listFlights(source, dest);  request.setAttribute("List", res);  response.sendRedirect("admin\_list.jsp");  break;  }  case "listAirlines":  {  resAir = planeDAO.listAirlines();  request.setAttribute("List", resAir);  response.sendRedirect("admin\_list.jsp");  break;  }  default:  {  res = flightDAO.listFlights();  request.setAttribute("List", res);  response.sendRedirect("admin\_list.jsp");  break;  }  }  }  /\*\*  \* @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse response)  \*/  protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException  {  doGet(request, response);  }  } |

|  |
| --- |
| ValidateFlight |
| package com.controllers;  import java.io.IOException;  import javax.servlet.RequestDispatcher;  import javax.servlet.ServletException;  import javax.servlet.http.HttpServlet;  import javax.servlet.http.HttpServletRequest;  import javax.servlet.http.HttpServletResponse;  import com.DAO.FlightDaoImpl;  import com.dto.Flight;  /\*\*  \* Servlet implementation class ValidateUser  \*/  public class ValidateFlight extends HttpServlet {  private static final long serialVersionUID = 1L;  private FlightDaoImpl flightDao;  /\*\*  \* @see HttpServlet#HttpServlet()  \*/  public ValidateFlight() {  super();  flightDao = new FlightDaoImpl();  }  /\*\*  \* @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse response)  \*/  protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException  {  if (request.getParameter("id") == null)  {  System.out.println("id is null");  response.getWriter().append("The id was nulled");  }  else  {  Flight flightID = flightDao.searchFlightById(Integer.parseInt(request.getParameter("id")));  request.setAttribute("flight", flightID);  RequestDispatcher dispatcher = request.getRequestDispatcher("registration.jsp");  dispatcher.forward(request, response);  }    }  /\*\*  \* @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse response)  \*/  protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {  response.getWriter().append("Served at: ").append(request.getContextPath());  }  } |

|  |
| --- |
| ValidatePassenger |
| package com.controllers;  import java.io.IOException;  import javax.servlet.RequestDispatcher;  import javax.servlet.ServletException;  import javax.servlet.http.HttpServlet;  import javax.servlet.http.HttpServletRequest;  import javax.servlet.http.HttpServletResponse;  import com.DAO.FlightDaoImpl;  import com.DAO.PassengerDaoImpl;  import com.dto.Flight;  import com.dto.Passenger;  /\*\*  \* Servlet implementation class ValidateUser  \*/  public class ValidatePassenger extends HttpServlet {  private static final long serialVersionUID = 1L;  private PassengerDaoImpl passDao;  private FlightDaoImpl flightDao;  /\*\*  \* @see HttpServlet#HttpServlet()  \*/  public ValidatePassenger() {  super();  passDao = new PassengerDaoImpl();  flightDao = new FlightDaoImpl();  }  /\*\*  \* @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse response)  \*/  protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException  {  response.getWriter().append("Served at: ").append(request.getContextPath());  }  /\*\*  \* @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse response)  \*/  protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException  {  if (request.getParameter("id") == null)  {  System.out.println("id is null");  response.getWriter().append("The id was nulled");  }  else  {  Passenger passenger = passDao.searchPassengerById(Integer.parseInt(request.getParameter("id")));  Flight flight = flightDao.searchFlightById(passenger.getFlight\_id());  request.setAttribute("passenger", passenger);  request.setAttribute("flight", flight);  RequestDispatcher dispatcher = request.getRequestDispatcher("summary.jsp");  dispatcher.forward(request, response);  }  }  } |
| AdminDaoImpl |
| package com.DAO;  import java.util.List;  import javax.persistence.TypedQuery;  import org.hibernate.Session;  import org.hibernate.SessionFactory;  import org.hibernate.Transaction;  import org.hibernate.boot.Metadata;  import org.hibernate.boot.MetadataSources;  import org.hibernate.boot.registry.StandardServiceRegistry;  import org.hibernate.boot.registry.StandardServiceRegistryBuilder;  import com.dto.Admin;  public class AdminDaoImpl implements AdminDAO  {  private SessionFactory factory;    public AdminDaoImpl()  {  System.out.println("Config about to load");  StandardServiceRegistry ssr = new StandardServiceRegistryBuilder().configure("hibernate.cfg.xml").build();  Metadata meta = new MetadataSources(ssr).getMetadataBuilder().build();  factory = meta.getSessionFactoryBuilder().build();  System.out.println("Config loaded");  }      @Override  public String addAdmin(Admin admin) {  String administrator = null;  Session session = factory.openSession();  Transaction txn = session.beginTransaction();  administrator = (String) session.save(admin);  txn.commit();  session.close();  return administrator;  }      @Override  public void updateAdmin(String admin, String pass)  {  Session session = factory.openSession();  Transaction txn = session.beginTransaction();  Admin administrator = session.get(Admin.class, admin);  administrator.setPass(pass);  session.update(administrator);  txn.commit();  session.close();  }    @Override  public void deleteAdmin(String admin) {  Session session = factory.openSession();  Transaction txn = session.beginTransaction();  Admin administrator = session.get(Admin.class, admin);  session.delete(admin);  txn.commit();  session.close();  }    @Override  public Admin searchAdminByUser(String admin) {  Session session = factory.openSession();  Transaction txn = session.beginTransaction();  String hql = "SELECT ad.user, ad.pass FROM administrators ad "+"WHERE ad.user="+admin;  TypedQuery<Admin> query = session.createQuery(hql);  Admin administrator = query.getSingleResult();  return administrator;  }  } |

|  |
| --- |
| FlightDaoImpl |
| package com.DAO;  import java.io.PrintWriter;  import java.util.List;  import javax.persistence.TypedQuery;  import org.hibernate.Session;  import org.hibernate.SessionFactory;  import org.hibernate.Transaction;  import org.hibernate.boot.Metadata;  import org.hibernate.boot.MetadataSources;  import org.hibernate.boot.registry.StandardServiceRegistry;  import org.hibernate.boot.registry.StandardServiceRegistryBuilder;  import org.hibernate.query.Query;  import com.dto.Flight;  public class FlightDaoImpl implements FlightDAO{  private SessionFactory factory;    public FlightDaoImpl()  {  System.out.println("Config about to load");  StandardServiceRegistry ssr = new StandardServiceRegistryBuilder().configure("hibernate.cfg.xml").build();  Metadata meta = new MetadataSources(ssr).getMetadataBuilder().build();  factory = meta.getSessionFactoryBuilder().build();  System.out.println("Config loaded");  }      @Override  public Integer addFlight(Flight flight) {  Integer flight\_id = null;  Session session = factory.openSession();  Transaction txn = session.beginTransaction();  flight\_id = (Integer) session.save(flight);  txn.commit();  session.close();  return flight\_id;  }      @Override  public void updateFlight(Integer flightID, int seats) {  Session session = factory.openSession();  Transaction txn = session.beginTransaction();    Flight flight = session.get(Flight.class, flightID);  System.out.println(flight);  flight.setSeats(seats);  System.out.println(flight);    session.update(flight);  txn.commit();  session.close();  }    @Override  public void deleteFlight(Integer flightID) {  Session session = factory.openSession();  Transaction txn = session.beginTransaction();  Flight flight = session.get(Flight.class, flightID);  session.delete(flight);  txn.commit();  session.close();  }    @Override  public List<Flight> listFlights(String src, String dest)  {  List<Flight> flights = null;  Session session = factory.openSession();  Transaction txn = session.beginTransaction();  String hql = "SELECT fli.source, fli.destination, FROM Flight AS fli";  TypedQuery<Flight> query = session.createQuery(hql);  query.setParameter("source", src);  query.setParameter("destination", dest);  flights = query.getResultList();  session.close();  return flights;  }    @Override  public List<Flight> listFlights(String src, String dest, String date, int seats)  {  List<Flight> flights = null;  Session session = factory.openSession();  Transaction txn = session.beginTransaction();  String hql = "SELECT fli.id, fli.source, fli.destination, fli.date, fli.seats FROM Flight AS fli WHERE ((fli.source =:source"+  ") AND (fli.destination =:destination"+  ") AND (fli.seats >=:seats"+  ") AND (fli.date >=:date))";  TypedQuery<Flight> query = session.createQuery(hql);  query.setParameter("source", src);  query.setParameter("destination", dest);  query.setParameter("seats", seats);  query.setParameter("date", date);    flights = query.getResultList();  session.close();  return flights;  }    @Override  public List<Flight> listFlights()  {  List<Flight> flights = null;  Session session = factory.openSession();  Transaction txn = session.beginTransaction();  String hql = "From Flight";  TypedQuery<Flight> query = session.createQuery(hql);    flights = query.getResultList();  session.close();  return flights;  }    @Override  public Flight searchFlightById(Integer flightID) {  Session session = factory.openSession();  Transaction txn = session.beginTransaction();  String hql = "FROM Flight = "+ flightID;  TypedQuery<Flight> query = session.createQuery(hql);  Flight flight = query.getSingleResult();  return flight;  }  } |

|  |
| --- |
| PassengerDaoImpl |
| package com.DAO;  import java.util.List;  import javax.persistence.TypedQuery;  import org.hibernate.Session;  import org.hibernate.SessionFactory;  import org.hibernate.Transaction;  import org.hibernate.boot.Metadata;  import org.hibernate.boot.MetadataSources;  import org.hibernate.boot.registry.StandardServiceRegistry;  import org.hibernate.boot.registry.StandardServiceRegistryBuilder;  import com.dto.Passenger;  public class PassengerDaoImpl implements PassengerDAO  {  private SessionFactory factory;    public PassengerDaoImpl()  {  System.out.println("Config about to load");  StandardServiceRegistry ssr = new StandardServiceRegistryBuilder().configure("hibernate.cfg.xml").build();  Metadata meta = new MetadataSources(ssr).getMetadataBuilder().build();  factory = meta.getSessionFactoryBuilder().build();  System.out.println("Config loaded");  }      @Override  public Integer addPassenger(Passenger passenger)  {  Integer pass\_id = null;  Session session = factory.openSession();  Transaction txn = session.beginTransaction();  pass\_id = (Integer) session.save(passenger);  txn.commit();  session.close();  return pass\_id;  }      @Override  public void updatePassenger(Integer passengerID, int seats) {  Session session = factory.openSession();  Transaction txn = session.beginTransaction();    Passenger passenger = session.get(Passenger.class, passengerID);  System.out.println(passenger);  passenger.setSeats\_purchased(seats);  System.out.println(passenger);    session.update(passenger);  txn.commit();  session.close();  }    @Override  public void deletePassenger(Integer passengerID) {  Session session = factory.openSession();  Transaction txn = session.beginTransaction();  Passenger passenger = session.get(Passenger.class, passengerID);  session.delete(passenger);  txn.commit();  session.close();  }      @Override  public List<Passenger> listPassengers(String src, String dest, String date, int seats)  {  List<Passenger> passengers = null;  Session session = factory.openSession();  Transaction txn = session.beginTransaction();    System.out.println("Source: "+src+  "\nDestination: "+dest+  "\nDate: "+date+  "\nSeats: "+seats);    String hql = "SELECT fli.source, fli.destination, fli.date, fli.seats FROM Passenger AS fli WHERE ((fli.source =:source"+  ") AND (fli.destination =:destination"+  ") AND (fli.seats >=:seats"+  ") AND (fli.date >=:date))";    //System.out.println(sql);  //Query query = session.createQuery(hql);  TypedQuery<Passenger> query = session.createQuery(hql);  query.setParameter("source", src);  query.setParameter("destination", dest);  query.setParameter("seats", seats);  query.setParameter("date", date);    //TypedQuery<Passenger> query = session.createSQLQuery(sql);  System.out.println("Query created");  passengers = query.getResultList();  System.out.println("Query completed, leaving the method.");  session.close();  return passengers;  }    @Override  public Passenger searchPassengerById(Integer passengerID) {  Session session = factory.openSession();  Transaction txn = session.beginTransaction();  String hql = "FROM Passenger = "+ passengerID;  TypedQuery<Passenger> query = session.createQuery(hql);  Passenger passenger = query.getSingleResult();  return passenger;  }  } |

|  |
| --- |
| PlaneDaoImpl |
| package com.DAO;  import java.util.List;  import javax.persistence.TypedQuery;  import org.hibernate.Session;  import org.hibernate.SessionFactory;  import org.hibernate.Transaction;  import org.hibernate.boot.Metadata;  import org.hibernate.boot.MetadataSources;  import org.hibernate.boot.registry.StandardServiceRegistry;  import org.hibernate.boot.registry.StandardServiceRegistryBuilder;  import com.dto.Passenger;  import com.dto.Plane;  public class PlaneDaoImpl implements PlaneDAO  {  private SessionFactory factory;    public PlaneDaoImpl()  {  System.out.println("Config about to load");  StandardServiceRegistry ssr = new StandardServiceRegistryBuilder().configure("hibernate.cfg.xml").build();  Metadata meta = new MetadataSources(ssr).getMetadataBuilder().build();  factory = meta.getSessionFactoryBuilder().build();  System.out.println("Config loaded");  }      @Override  public Integer addPlane(Plane plane) {  Integer plane\_id = null;  Session session = factory.openSession();  Transaction txn = session.beginTransaction();  plane\_id = (Integer) session.save(plane);  txn.commit();  session.close();  return plane\_id;  }      @Override  public void updatePlane(String airline, Integer flightID)  {  Session session = factory.openSession();  Transaction txn = session.beginTransaction();    Plane plane = session.get(Plane.class, airline);  System.out.println(plane);  plane.setFlight\_id(flightID);  System.out.println(plane);    session.update(plane);  txn.commit();  session.close();  }    @Override  public void deletePlane(Integer planeID)  {  Session session = factory.openSession();  Transaction txn = session.beginTransaction();  Plane plane = session.get(Plane.class, planeID);  session.delete(plane);  txn.commit();  session.close();  }    @Override  public List<Plane> listPlanes()  {  List<Plane> planes = null;  Session session = factory.openSession();  Transaction txn = session.beginTransaction();  String hql = "From Plane";  TypedQuery<Plane> query = session.createQuery(hql);    planes = query.getResultList();  session.close();  return planes;  }    @Override  public List<Plane> listAirlines()  {  List<Plane> planes = null;  Session session = factory.openSession();  Transaction txn = session.beginTransaction();  String hql = "Select Plane.airline From Plane";  TypedQuery<Plane> query = session.createQuery(hql);    planes = query.getResultList();  session.close();  return planes;  }  @Override  public Plane searchPlaneByID(Integer flightID)  {  Session session = factory.openSession();  Transaction txn = session.beginTransaction();  String hql = "FROM Passenger = "+ flightID;  TypedQuery<Plane> query = session.createQuery(hql);  Plane plane = query.getSingleResult();  return plane;  }  } |

|  |
| --- |
| Admin |
| package com.dto;  import javax.persistence.Column;  import javax.persistence.Entity;  import javax.persistence.GeneratedValue;  import javax.persistence.GenerationType;  import javax.persistence.Id;  import javax.persistence.Table;  @Entity  @Table(name="administrators")  public class Admin  {  @Id  @GeneratedValue(strategy=GenerationType.SEQUENCE)  @Column(name="username")  private String user;  @Column(name="password")  private String pass;    public String getUser() {  return user;  }  public void setUser(String user) {  this.user = user;  }  public String getPass() {  return pass;  }  public void setPass(String pass) {  this.pass = pass;  }    @Override  public String toString() {  return "Admin [user=" + user + ", pass=" + pass + "]";  }  } |

|  |
| --- |
| Flight |
| package com.dto;  import javax.persistence.Column;  import javax.persistence.Entity;  import javax.persistence.GeneratedValue;  import javax.persistence.GenerationType;  import javax.persistence.Id;  import javax.persistence.Table;  @Entity  @Table(name="avail\_flights")  public class Flight  {  @Id  @GeneratedValue(strategy=GenerationType.SEQUENCE)  @Column(name="flight\_id")  private int id;  @Column(name="src\_point")  private String source;  @Column(name="dest\_point")  private String destination;  @Column(name="travel\_date")  private String date;  @Column(name="time\_to\_board")  private String time;  @Column(name="price")  private double price;  @Column(name="seat\_vacancy")  private int seats;    public Flight()  {  }    public Flight(int id, String source, String destination, String date, String time, String price, String seats)  {  super();  this.id = id;  this.source = source;  this.destination = destination;  this.date = date;  this.time = time;  this.price = Double.parseDouble(price);  this.seats = Integer.parseInt(seats);  }    public Flight(int id, String source, String destination, String seats, String date)  {  super();  this.source = source;  this.destination = destination;  this.seats = Integer.parseInt(seats);  this.date = date;  }    public Flight(String source, String destination, String seats, String date)  {  super();  this.source = source;  this.destination = destination;  this.seats = Integer.parseInt(seats);  this.date = date;  }    public int getId() {  return id;  }  public void setId(int id) {  this.id = id;  }  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;  }  public String getDate() {  return date;  }  public void setDate(String date) {  this.date = date;  }  public String getTime() {  return time;  }  public void setTime(String time) {  this.time = time;  }  public double getPrice() {  return price;  }  public void setPrice(double price) {  this.price = price;  }  public int getSeats() {  return seats;  }  public void setSeats(int seats) {  this.seats = seats;  }  @Override  public String toString() {  return "Flight [id=" + id + ", source=" + source + ", destination=" + destination + ", date=" + date + ", time="  + time + ", price=" + price + "]";  }  } |
| Passenger |
| package com.dto;  import javax.persistence.Column;  import javax.persistence.Entity;  import javax.persistence.GeneratedValue;  import javax.persistence.GenerationType;  import javax.persistence.Id;  import javax.persistence.Table;  @Entity  @Table(name="passengers")  public class Passenger  {  @Id  @GeneratedValue(strategy=GenerationType.SEQUENCE)  @Column(name="passenger\_id")  private int passenger\_id;  @Column(name="flight\_id")  private int flight\_id;  @Column(name="firstname")  private String firstname;  @Column(name="lastname")  private String lastname;  @Column(name="seats\_purchased")  private int seats\_purchased;    public Passenger()  {  }    public Passenger(String firstname, String lastname, int flight\_id, int seats\_purchased)  {  this.firstname = firstname;  this.lastname = lastname;  this.flight\_id = flight\_id;  this.seats\_purchased = seats\_purchased;  }    public int getPassenger\_id() {  return passenger\_id;  }  public void setPassenger\_id(int passenger\_id) {  this.passenger\_id = passenger\_id;  }  public int getFlight\_id() {  return flight\_id;  }  public void setFlight\_id(int flight\_id) {  this.flight\_id = flight\_id;  }  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 int getSeats\_purchased() {  return seats\_purchased;  }  public void setSeats\_purchased(int seats\_purchased) {  this.seats\_purchased = seats\_purchased;  }    @Override  public String toString() {  return "Passenger [passenger\_id=" + passenger\_id + ", flight\_id=" + flight\_id + ", firstname=" + firstname  + ", lastname=" + lastname + ", seats\_purchased=" + seats\_purchased + "]";  }    } |

|  |
| --- |
| Plane |
| package com.dto;  import javax.persistence.Column;  import javax.persistence.Entity;  import javax.persistence.GeneratedValue;  import javax.persistence.GenerationType;  import javax.persistence.Id;  import javax.persistence.Table;  @Entity  @Table(name="plane")  public class Plane  {  @Id  @GeneratedValue(strategy=GenerationType.SEQUENCE)  @Column(name="flight\_id")  private int flight\_id;  @Column(name="plane\_id")  private int plane\_id;  @Column(name="airline")  private String airline;    public int getFlight\_id() {  return flight\_id;  }  public void setFlight\_id(int flight\_id) {  this.flight\_id = flight\_id;  }  public int getPlane\_id() {  return plane\_id;  }  public void setPlane\_id(int plane\_id) {  this.plane\_id = plane\_id;  }  public String getAirline() {  return airline;  }  public void setAirline(String airline) {  this.airline = airline;  }    @Override  public String toString() {  return "Plane [flight\_id=" + flight\_id + ", plane\_id=" + plane\_id + ", airline=" + airline + "]";  }  } |

# Screenshots

|  |
| --- |
| Admin Page |
| Text  Description automatically generated with medium confidenceGraphical user interface, application  Description automatically generated |

|  |
| --- |
| Passenger pages |
| Graphical user interface, application  Description automatically generatedGraphical user interface, application  Description automatically generated |

|  |
| --- |
| Database hierarchy |
| A picture containing graphical user interface  Description automatically generated |