**Air Route Planner**

Software Design Specification

**That “One” Team**

**SE 300 Section 1**

Table of Contents

[**Purpose** 3](#_Toc349673747)

[**Problem Statement** 3](#_Toc349673748)

[**Team Project Information** 3](#_Toc349673749)

[High Level Design 4](#_Toc349673750)

[Detailed Design 7](#_Toc349673751)

# **Purpose**

The purpose of this document is to specify the design details for the Air Route Planner (ARP). The ARP will contain classes that break down the functionality into several key components: user interface, file handling, search algorithms and the manner in which the output will be formatted. A detailed description of what methods and features will be included in this software will be outlined in the later sections of this document.

# **Problem Statement**

The goal of this project is to create an Air Route Planner (ARP) that will allow a user to select an origin and destination based on a database of airports. The system will then search for a route based on time, cost, and airline most frequently used. This information will then be displayed back to the user.

# **Team Project Information**

Course: Spring 2013 SE300 Section 1

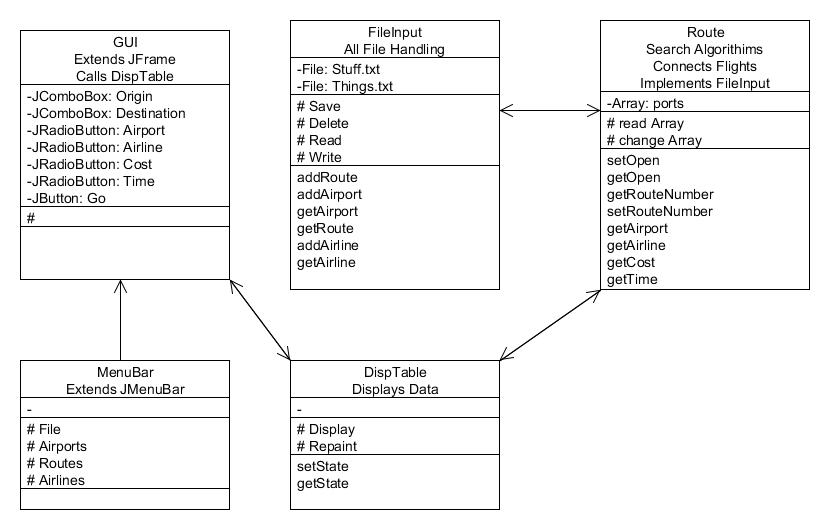
Team: That “One” Team

Members/Roles:

* **Team Leader:** Brian Powell
* **Development Manager:** Brittany Rompa
* **Planning Manager:** Craig Wilkerson
* **Quality Manager:** Yutong Zhu
* **Req/Support Manager:** Muraad Khan

# High Level Design

Introduction

The following is an explanation of the high level design. This includes a diagram to illustrate the way the classes function and work together, and a list of the classes with their purposes and methods to be defined later in the document.

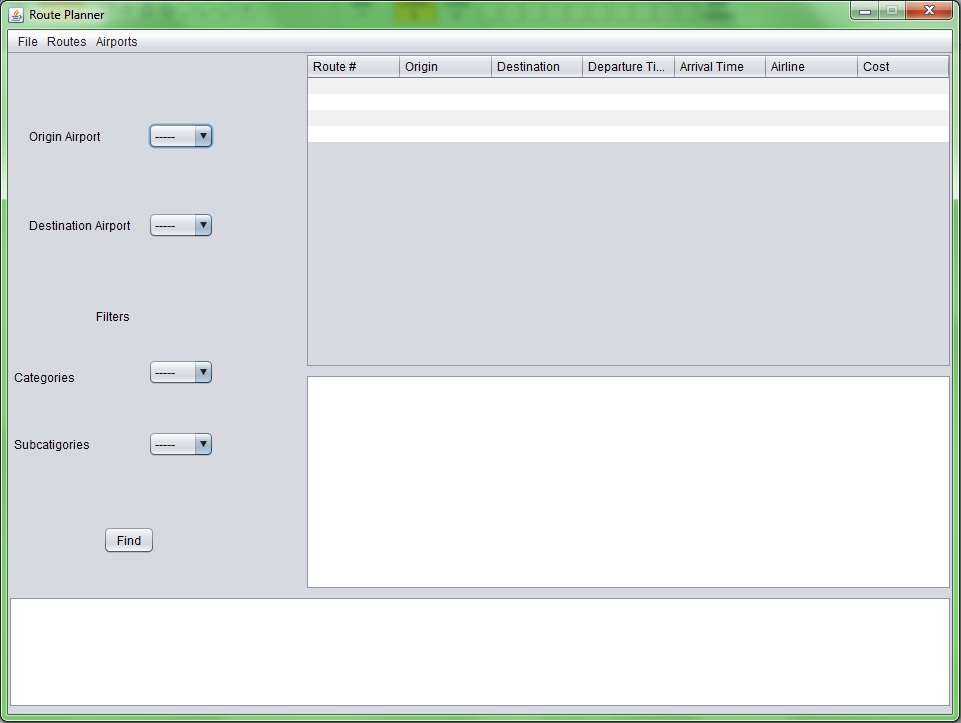
Class Diagram

Class Specifications

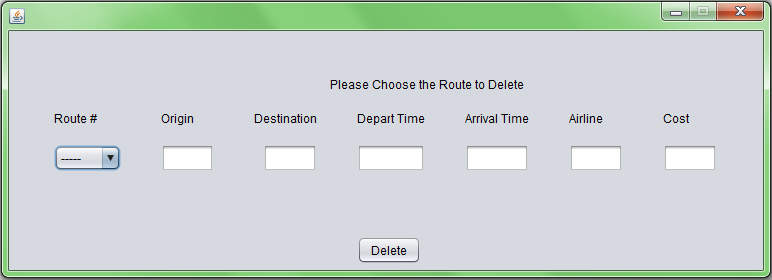
**Classes:**

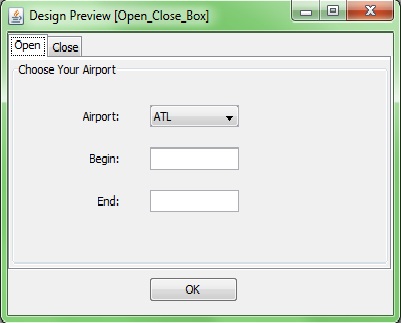
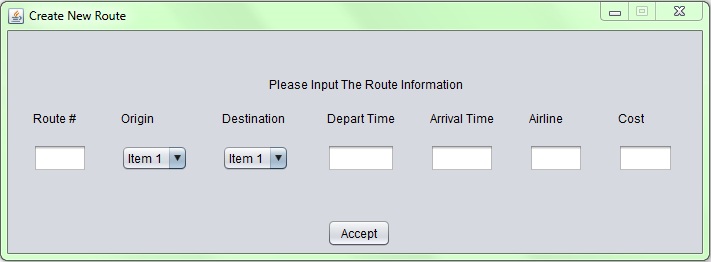
* FileInput
  + All file handling
    - Save
    - Delete
    - Modify (maybe)
  + Write to temp file
    - Saved to master once save is clicked
  + Methods
    - addRoute
    - addAirport
    - etc.
* DispTable extends JPanel
  + Default display all entries
  + Create an object of FileInput
  + Create an object of Filter
* Route (search algorithims)
  + Array
    - Airport by open/closed by route numbers
* Performs filter functions
* Connects Flights
* GUI extends JFrame
  + 2 combo boxes
    - Origin
    - Destination
  + 4 radio buttons
    - Filters
  + JButton
    - Go
  + Call DispTable class
* MenuBar extends JMenuBar
  + File
    - Save
    - Open (maybe)
    - Close (maybe)
    - Save to different file types (maybe)
  + Airports
    - Add
    - Delete
    - Open
    - Close
    - Modify (maybe)
  + Routes
    - Add
    - Delete
    - Modify (maybe)
  + Airlines
    - Add
    - Delete
    - Modify (maybe)
  + Adding and delete will be calling FileInput methods
  + Opening and closing calls Route which will have open and close parameters (be careful about this. Since we are writing to arrays which are unique to each instance of Route. Might want to find a better way to open and close)

User Interfaces









File and Report Formats

#comment \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

#comment Example Input file

#comment Version 1.0

#comment Authors: Evan Richardson, Keith Garfield

#comment

#comment This file represents the state of the air network upon first system start-up.

#comment Use this file to ensure that your system has basic functionality with respect to loading files.

#comment Use this file as an example of how to create other input files you may need during development and test.

#comment \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

#comment All airports must be listed first, per three-letter code, one airport per line

#comment Format: XXX

#airports

ATL

DCA

ORL

JFK

#comment All routes to be listed here, one route per line.

#comment Format: route\_id, carrier\_id, ap\_dep, time\_dep, ap\_arr, time\_arr, price, [val8, val9, val10, ...]

#routes

R001, DELTA, ATL, 0800, JFK, 1100, 184.00

R002, UNITED, ATL, 1430, JFK, 1800, 230.00

R003, DELTA, ORL, 0830, JFK, 1230, 325.00

R004, UNITED, ORL, 1000, ATL, 1100, 150.00

R005, DELTA, ORL, 0820, ATL, 0930, 125.00

R006, UNITED, ORL, 0730, DCA, 1100, 230.00

R007, DELTA, ORL, 0900, DCA, 1120, 240.00

R008, UNITED, DCA, 1200, JFK, 1330, 140.00

R009, DELTA, JFK, 0800, ATL, 1100, 184.00

R010, UNITED, JFK, 1430, ATL, 1800, 230.00

R011, DELTA, JFK, 0830, ORL, 1230, 325.00

R012, UNITED, ATL, 1000, ORL, 1100, 150.00

R013, DELTA, ATL, 0820, ORL, 0930, 125.00

R014, UNITED, DCA, 0730, ORL, 1100, 230.00

R015, DELTA, DCA, 0900, ORL, 1120, 240.00

R016, UNITED, JFK, 1200, DCA, 1330, 140.00

#comment All closures to be listed here.

#comment Format: airport\_code, time\_c, time\_o

#closures

#comment No closures in initial data set.

#comment Example: ATL, 0800, 1000

#end

Files are to be in .txt format.

# Detailed Design

Introduction

This portion of the document describes the purpose and function of each method. This is to be used to develop an understanding for how each method will interact with another method.

Method Specification

addRoute

Adds Route to an Airport

addAirport

Adds an Airport to the list

getAirport

Returns the Airport according to the information provided

getRoute

Returns the Route according to the information provided

addAirline

Adds an Airline to the list

getAirline

Returns the Airline according to the information provided

setOpen

Sets the Airport provided to open

getOpen

Returns if the Airport provided is open

getRouteNumber

Returns the Route number of the route selected

setRouteNumber

Sets selected Route Number

getFilterAirport

Returns whether the Airport filter is selected

getFilterAirline

Returns whether the Airline filter is selected

getFilterCost

Returns whether the Cost filter is selected

getFilterTime

Returns whether the Time filter is selected

setState

Sets the state of the display

getState

Returns the state of the display