

**World Plane Inc.**

# **CS509 Reservation System**

## **TEST PLAN**

Date: December 5, 2017

**Prepared by:**

Team Muse

Yahan Zhu, Marie Solman, Juexing Wang, Jiaoyan Chen

## Table of Contents

### Contents

Introduction.. 3

1.1 Objectives. 3

1.2 Team Members. 3

2 Scope.. 3

3 Assumptions / Risks. 4

3.1 Assumptions. 4

3.2 Risks. 4

4 Requirement Verification Test Matrix (RVTM) 4

5 Test Cases. 6

5.1 Test Environment. 6

5.2 Test Approach. 6

6 Test Procedures. 7

# Introduction

The Test Plan has been created to improve communication between stakeholders and developers and ensure a test plan has been agreed upon. This test plan encapsulates the objectives, scope, schedule, risks and approach. This document will include an analysis of risk, test cases, test procedures, and a requirement verification test matrix.

## 1.1 Objectives

This Airline Reservation System is a ‘proof of concept’ conceptually similar in behavior to existing web-based airline reservation systems such as kayak.com or expedia.com

This Airline Reservation System will allow customers to reserve one way trips or round trips by searching the departure airport, departure date, arrival airport, return date, how many layovers and seating type. Airlines satisfying all requirement information will be returned to customers and sorted by departure time, arrival time, trip duration, or price. However, the system cannot support payment process, customer information storage or trip canceling.

## 1.2 Team Members

Name	Role
Yahan Zhu	Tester/Documentation
Marie Solman	Documentation/Developer
Jiaoyan Chen	Project Manager/Developer
Juexing Wang	Tester/Developer

## Scope

This document identifies the testing approach which will be used to verify all functional and non-functional requirements identified in the Requirements Analysis Document (RAD) originally delivered 10/6/2017 and as updated 11/10/2017. Testing will support all Use Cases identified in the RAD. Evidence of test execution will be provided for each requirement and Use Case.

Section 4 below identifies the test methodology (test, demonstration, analysis or inspection) which will be used for each functional and nonfunctional requirement.

Section 5 below describes the testing plan which will be used to verify operation of each Use Case.

## Assumptions / Risks

### 1.3 Assumptions

This section lists assumptions that are made specific to testing performed for this project. For example:

1. The provided server is available and has flight data for days between 12/06/2017 and 12/31/2017.
2. The Google Time Zone API is available and will return results for all time zone requests.
3. Internet connectivity is available.

### 1.4 Risks

The following risks have been identified and the appropriate action identified to mitigate their impact on the project. The impact (or severity) of the risk is based on how the project would be affected if the risk was triggered. The trigger is what milestone or event would cause the risk to become an issue to be dealt with.

#	Risk	Impact	Trigger	Mitigation Plan
1	Server load from other teams may adversely affect response time	Moderate	Invalid timing results	Multiple test runs with results averaged. Outliers discarded.
2	Google API may have reached its limit for the day	Moderate	Local time will not display	Store time zones for airports locally and import only time zones for airports with no timezone stored
3	Internet is unavailable	High	Software will not run	Ensure internet is available for tests and demonstrations

# Requirement Verification Test Matrix (RVTM)

Each requirement will be tested at the unit, integration or system stage of the software development life cycle. Some requirements will be verified at more than a single stage. The following table identifies the verification test method for each requirement. Each requirement will be verified using one, or more, of the following verification methods:

(T) Test where the system is stimulated with defined input and output is captured, instrumented, measured by some instrumentation and the results are then compared to the expected results.

(D) Demonstration where the system is stimulated with defined input and output is observed using one or more of the senses.

(A) Analysis is where a mathematical model of the system (or simulation) is used to evaluate how the system will perform.

(I) Inspection is where visual examination of the system is used to verify compliance with requirement.

For each requirement, identify the software development phase

Requirement	Verification				Phase		
	T	D	A	I	U	I	S
System shall allow a customer make an airline reservation to travel from a departure airport to a destination airport of their choice.		X					X
The system shall allow a customer to search for trips with a valid three character identifier of the departing and arriving airport.	X				X		
The system shall accept a date for departing date and return date, for round trips, in YYYYMMDD format.	X				X		
The system will ensure the return date is after the departing date.	X				X		

System shall allow a customer to book a trip from a departure airport to an arrival airport using a series of connecting flights with a maximum of two stopovers.		X		X	X		
System shall allow a customer to make an airline reservation to travel either one-way between two airports or reserve a round-trip between two airports.		X				X	X
System shall allow customers to choose their type of seat: first-class seating or coach seating.		X				X	X
System shall present a list of trips including direct flights and connecting flights with up to two layovers.		X	X			X	
System shall limit layovers to no less than thirty minutes and no more than four hours. The system shall not display layovers outside these bounds.			X		X	X	
System shall allow the user to sort list of trips by price in USD, local departure time, local arrival time and travel time.		X				X	
System shall respond when seating is not available.		X		X		X	
The system shall provide concise instruction for user's each step.		X					X
The system shall display results in a neat way (shown by departure time, arrival time, travel time, stopovers, and price), make sure every results is easy to distinguish with others		X					X
The system shall be stable and validate user inputs to ensure they are correct and as expected.	X				X		
The system shall ensure departure time is not after return time.	X				X		
The system shall notify the user if no trips are available for the user's search criteria.		X					X

The system shall notify the user when the system is searching the airlines		X				X	
Input validation will be unit tested using Junit test cases developed in parallel with application software.				X		X	X
Response time for any requested actions shall be reasonable. Operations in excess of three seconds shall provide a notification to the customer indicating the system is operating.		X				X	
The software shall be compatible on the client machine.				X		X	X
The system shall use the JAVA programming language for platform independence				X		X	X
The system shall request XML files from the server using HTTP GET commands, and retrieve information from the XML files.				X	X		

# Test Cases

## 1.5 Test Environment

To facilitate testing on the system, a working internet connection must be available for the machine conducting the testing. In addition, the external server with flights, airplanes, and airports and the Google Time Zone API must be working.

## 1.6 Test Approach

(1.) System shall allow a customer make an airline reservation to travel from a departure airport to a destination airport of their choice.

- Demonstration method executed at system level of testing

(2.) The system shall allow a customer to search for trips with a valid three character identifier of the departing and arriving airport.

- Unit test method will be used to ensure validator for departure and arrival airport only accepts valid three character airport codes and will not accept blank values or invalid combinations of characters

(3.) The system shall accept a date for departing date and return date, for round trips, in YYYYMMDD format.

- Unit testing will be conducted
- Testing will ensure that only inputs that match the YYYYMMDD format will be accepted and any other format will be rejected
- Testing will ensure that the year is greater than or equal to 2017
- Testing will ensure that the month is between 1 and 12
- Testing will ensure that the day is between 1 and the expected maximum number of days per month (i.e. 31 days for January, 30 days for April, 28 or 29 days for February)

(4.) The system will ensure the return date is after the departing date.

- Unit testing will be conducted
- If a departure date has been set, the return date must be after
- If a return date has been set, the departure date must be before

(5.) System shall allow a customer to book a trip from a departure airport to an arrival airport using a series of connecting flights with a maximum of two stopovers.

- Unit testing and inspection at a unit level will be conducted
- Testing will ensure that there are a maximum of two stopovers
- Inspection will ensure that for a known group of flights from a variety of airports on a given day, that all expected connecting flights are returned

(6.) System shall allow a customer to make an airline reservation to travel either one-way between two airports or reserve a round-trip between two airports.

- Demonstration testing will be conducted at the integration and system level
- Testing should ensure a customer can select a one way or round trip.
- Testing should ensure that if a one way trip has been selected, only one set of flights will return and the user can only select a single trip.
- Testing should ensure that if a round trip has been selected, two sets of flights will return and the user can select a first trip for the outgoing trip and a second trip for the return trip

(7.) System shall allow customers to choose their type of seat: first-class seating or coach seating.

- Demonstration testing will be conducted at both integration and system levels
- Testing should ensure that flights that have both first class and coach seats available indicate that they have both available



- If both first class or coach seats are available, testing should ensure a user can select either first class or coach
- After flight updates to the server have been completed, test should be rerun and ensure that the flight updates the server and will reflect upon further tests.

(8.) System shall present a list of trips including direct flights and connecting flights with up to two layovers.

- Demonstration and inspection testing will be conducted at an integration level
- Test requires that flights are being returned for user inputs or hard coded values and that connecting flights are being calculated
- Test requires analysis of a known group of values that all expected flights are being displayed

(9.) System shall limit layovers to no less than thirty minutes and no more than four hours. The system shall not display layovers outside these bounds.

- Unit testing and analysis testing will be conducted at the unit and integration level
- Testing should account for midnight boundaries
- Testing will verify that for a known flight leaving late in the day, after 8, that all expected connecting flights will be included, including connecting flights after the midnight boundary

(10.) System shall allow the user to sort list of trips by price in USD, local departure time, local arrival time and travel time.

- Demonstration testing will be conducted at the integration phase
- Flights must be returning from server and sorting must be enabled
- Test should be rerun for regression testing after connecting flights have been implemented
- Testing should ensure sorting is available by price, departure time, arrival time, and travel time
- Testing should verify that sorting by price, departure time, arrival time, and travel time sort as expected and all flights appear each time a sort is conducted

(11.) System shall respond when seating is not available.

- Demonstration and inspection will both be conducted at the integration phase.
- The testing here requires that flights can be imported, manipulated and updated to the server.
- Testing should ensure that when a flight has been filled, it no longer appears when a search would otherwise return it.

(12.) The system shall provide concise instruction for user's each step.

- Demonstration testing will be conducted at the system level
- The tester should ensure at each phase that there are displayed directions and a designated way to interact with the system.

(13.) The system shall display results in a neat way (shown by departure time, arrival time, travel time, stopovers, and price), make sure every results is easy to distinguish with others

- Demonstration testing will be conducted at the system level

- The tester should ensure that when flights display, the flight number, the departure time, arrival time, travel time, stopovers, and price should display in clearly delineated columns.

(14.) The system shall be stable and validate user inputs to ensure they are correct and as expected.

- Unit testing will be conducted
- Testing will ensure that all inputs only accept valid inputs
- Testing will cover date/time formats, airport codes, and Y/N value for first class

(15.) The system shall ensure departure time is not after return time.

- Unit testing will be conducted
- If a departure time has been set, the return time must be after
- If a return time has been set, the departure time must be before

(16.) The system shall notify the user if no trips are available for the user's search criteria.

- Demonstration testing will be conducted at a system level
- Test will input known dates outside the valid date range and ensure the flights return indicate no flights are available for that day
- Test will exhaust and fill all flights for a certain day and ensure all flights that return are at capacity, are filtered, and indicate no flights are available for that day

(17.) The system shall notify the user when the system is searching the airlines

- Demonstration test at an integration level
- Tester should ensure that immediately upon inputting search criteria, the system updates the user that the system is searching and filtering flights

(18.) Input validation will be unit tested using Junit test cases developed in parallel with application software.

- Inspection testing will be conducted at every phase of the project
- Inspection will ensure that any new classes and code has attached Junit testing

(19.) Response time for any requested actions shall be reasonable. Operations in excess of three seconds shall provide a notification to the customer indicating the system is operating.

- Demonstration test at an integration level
- Tester should ensure that immediately upon inputting search criteria, the system updates the user that the system is searching and filtering flights

(20.) The software shall be compatible on the client machine.

- Inspection testing will be conducted at every phase of the project
- Testing will examine code to ensure JAVA programming language is in use to ensure compatibility

(21.) The system shall use the JAVA programming language for platform independence

- Inspection testing will be conducted at every phase of the project

- Testing will examine code to ensure JAVA programming language is in use

(22.) The system shall request XML files from the server using HTTP GET commands, and retrieve information from the XML files.

- Inspection testing will be conducted at a unit level
- The analysis tester/developer will examine that the flight, airport, and airline requests are made with HTTP GET commands

## Test Procedures

Test procedures are detailed descriptions of how a specific test will be performed. The detail should be sufficient for someone familiar with the application domain and operation of the software and software development environment would be able to conduct the test using the instruction steps provided.

Test Case: Sort flights				Test Organization:World Plane Inc.		
Test Version: Version 1				Execution Date:		
Description: System shall allow the user to sort list of trips by price in USD, local departure time, local arrival time and travel time.				Executed By:		
				SW Baseline: SW version 1		
Preconditions: Program has started, departing and arrival airports and dates have been inputted, and round trip has been selected						
Dependencies: DAO and Server Interface have been developed and can return flights to user						
Step	Actions	Data	Expected Result			
1.	System will request departure airport					

2.	Input “BOS”	“BOS”	System accepts date input			
3.	System will request arrival airport		System has prompted user for input			
4.	Input “JFK”	“JFK”	System accepts date input			
5.	System will request departure date		System has prompted user for input			
6.	Input “12/10/2017”	12/10/2017	System accepts date input			
7.	System will ask for seat preference		System has prompted user for input			
8.	Input “coach”	“coach”	System accepts date input			
9.	System will ask if you want round trip		System has prompted user for input			
10.	Input “no”	“no”	System accepts date input			
11.	System will return flights		System displays flights			
12.	System will ask if you want to sort		System has prompted user for input			
13.	Input “yes”	“yes”	System accepts date input			

14.	System will ask if you want to sort by price, duration, departure time, or arrival time		System has prompted user for input			
15.	Input “price”	“price”	System accepts date input			
16.	System will display flights in ascending sort order of price		System displays results			
17.	Repeat steps 1-14					
18.	Input “time”	“time”	System accepts date input			
19.	System will display flights in ascending time duration		System displays results			
20.	Repeat steps 1-14					
21.	Input “departure”	“departure”	System accepts date input			
22.	System will display flights in ascending departure time		System displays results			

23.	Repeat steps 1-14					
24.	Input “arrival”	“arrival”	System accepts date input			
25.	System will display flights in ascending arrival time		System displays results			
Postconditions: System has sorted flights and displayed them						

Test Case: Display list of flights				Test Organization:World Plane Inc.		
Test Version: Version 1				Execution Date:		
Description: System shall present a list of trips including direct flights and connecting flights with up to two layovers.				Executed By:		
				SW Baseline: SW version 1		
Preconditions: Program has started, departing and arrival airports and dates have been inputted, and round trip has been selected						
Dependencies: DAO and Server Interface have been developed and can return flights to user						
Step	Actions	Data	Expected Result			
1.	System will request departure airport		System has prompted user for input			
2.	Input “BOS”	“BOS”	System accepts date input			
3.	System will		System has			

	request arrival airport		prompted user for input			
4.	Input “JFK”	“JFK”	System accepts date input			
5.	System will request departure date		System has prompted user for input			
6.	Input “12/10/2017”	12/10/2017	System accepts date input			
7.	System will ask for seat preference		System has prompted user for input			
8.	Input “coach”	“coach”	System accepts date input			
9.	System will ask if you want round trip		System has prompted user for input			
10.	Input “no”	“no”	System accepts date input			
11.	System will return flights		System displays flights			
Postconditions: System has returned and displayed flights						