**World Plane Inc.**

**CS509 Reservation System**

**TEST PLAN**

Date: 30 Mar 2019

**Prepared by**: jellycat

Table of Contents

Contents

[Introduction 3](#_Toc497221943)

[1.1 Objectives 3](#_Toc497221944)

[1.2 Team Members 3](#_Toc497221945)

[2 Scope 3](#_Toc497221946)

[3 Assumptions / Risks 4](#_Toc497221947)

[3.1 Assumptions 4](#_Toc497221948)

[3.2 Risks 4](#_Toc497221949)

[4 Requirement Verification Test Matrix (RVTM) 4](#_Toc497221950)

[5 Test Cases 6](#_Toc497221951)

[5.1 Test Environment 6](#_Toc497221952)

[5.2 Test Approach 6](#_Toc497221953)

[6 Test Procedures 7](#_Toc497221954)

# Introduction

The Test Plan has been created to communicate the test approach to team members and stakeholders. It includes the objectives, scope, schedule, risks and approach. This document will clearly identify what the test deliverables will be and what is deemed in and out of scope.

## Objectives

CS 509 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.

Our proof of concept software is browser-based and is aimed to transitioning the Travel Agency airline travel reservation system to a Retail Customer airline reservation system with feasibility. And it’s aimed to be conceptually similar to the existing web-based airline reservation systems (like google flights). It’s designed such that all new customers can understand all flights options easily and clearly and make flight reservations based on their personal preferences.

## Team Members

|  |  |
| --- | --- |
| **Name** | **Role** |
| Zhuoqun Yuan | Developer / Tester |
| Wei Xiong | Project Manager / Tester |
| Dongrui Qi | Developer / Tester |
| Yueyue Zhong | Documentation / Tester |

# Scope

This document identifies the testing approach which will be used to very all functional and non-functional requirements identified in the Requirements Analysis Document (RAD) originally delivered 20 Mar 2019 and as updated 28 Mar 2019. Testing will support all Use Cases identified in the RAD. Evidence of test execution will be provided for each requirement and Use Case.

# Assumptions / Risks

## 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 05/01/2019 and 05/30/2019.
2. Customer will only be allowed to input airport code, not airport name or other.
3. System will be used on a reasonable modern laptop.
4. System will be used with access to a stable Internet connection.
5. Server will provide stable connection if requests made by the system.
6. System will only be used by no more than 100 users at once.

## Risks

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| # | 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 | Search complexity: the depth of 3 exhaustive brute force search may be too demanding for real-time use. | Severe | Search function triggered. | Improve search algorithms and local cache flights that have already been retrieved. |
| 3 | The database is accessed by http methods, and has no authentication, so may be adversely attacked by outside | Severe | Database unsecure | Add authentication to the database access. |

# Requirement Verification Test Matrix (RVTM)

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Requirement | Verification | | | | Phase | | |
| T | D | A | I | U | I | S |
| World Plane Inc allow customers to specify departure airport, arrival airport, departure date, class type (coach seat or first-class seat), one-way trip/round-trip, which are all specified by customers. And the system will display all satisfying flights that are no more than 2 layovers and with each stopover time between 30 minutes and 2 hours. | X |  |  |  |  | X |  |
| The system can transform the GMT time to local time according to airport’s longitude and latitude. |  | X |  |  | X |  |  |
| The system can sort trips according to departure time, arrival time, flight time, price based on customer’s request. | X |  |  |  | X |  |  |
| After the customer confirmed the desired legs of flights, the system will be able to update the information on the server side. |  | X |  |  |  | X |  |
| The system may not display the flight if seats are not available for any legs of the flight. |  | X |  |  |  |  | X |
| Customers can cancel the order before they confirmed it and they can exit the system. | X |  |  |  |  | X |  |
| Reasonable and convenient UI shall be provided for customers. |  | X |  |  |  |  | X |
| The system will be able to return information of ‘No seat available’ when requested seat is not available for all legs of flights. |  | X |  |  |  |  | X |
| The system will be able to return the information of ‘server is busy’ to customer when database is locked by others. |  | X |  |  |  |  | X |
| Response time for any requested actions will reasonable. Operations in excess of 3 seconds will provide indication to the customer the system is operating. | X |  |  |  |  |  | X |
| A request of customer shall be answered within seconds, and the system shall be able to process several requests at the same time. |  | X |  |  |  |  | X |
| Accurate and complete results shall be output according to the customer request. |  | X |  |  |  |  | X |
| The application will use the JAVA programming language for platform independence. |  |  |  | X |  |  | X |
| The application shall be run on software environment and get the same performance. |  | X |  |  |  |  | X |
| The application shall have the ability to be maintained to fix defects or add functions. |  |  |  | X |  |  | X |

# Test Cases

## Test Environment

Software: Eclipse

Networks: HTTP GET/POST request

Services: WPI virtual machine

Systems: Windows, Mac OS

Database: WPI virtual machine

Data: XML

## Test Approach

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Methodologies | Type of testing | Description |
| Verify initiating System | Demonstration | Statebased Testing | Enter the url of the product website as input, the website (UI) showing in the browser is observed. |
| Verify parameters entering correctly | Test | equivalence testing, boundary testing | Enter the parameters of trip as input (depart date, departure airport code, arrival airport code), if system get the same parameters from the UI and will tell if they are legal parameters, then it passes the test. |
| Verify trips grabbing | Test | equivalence testing | Given a set of legal parameters, the trips are returned as supposed. |
| Verify Time Transforming | Test | equivalence testing | Given an airport code and a GMT time, assert the transformed time is the same as the local time, if yes, it passes the test. |
| Verify Determining Available Seat | Test | equivalence testing | Given a flight and a seat class (coach/first class), the available seat number should be given as supposed. |
| Verify sorting trips | Test | equivalence testing | Given a list of trips with the same departure, arrival airport code and departure date, a flag of which it wants to sort (price, departure date, arrival date), it should return a sorted list as supposed. |
| Verify reserving trips | Demonstration | Equivalence testing | Reserved a trip, retrieve the available seat of each flight in that trip, assert the available seat is decreasing by 1. |
| Verify error processing for no available trips | Demonstration | Statebased Testing | Given a set of parameters of trip we have identified there is no trip for it in the DB server, if system return false, then it passes the test. |
| Verify error processing for no available seating for trips | Demonstration | Statebased Testing | Given a set of parameters of trip we have identified there is no available seat in the flight of trip for it in the DB server, if system return false, then it passes the test. |
| Verify canceling before confirmation | Demonstration | Statebased Testing | In the UI, if cancel before confirmation, the UI will roll back to the initial UI, if doing so, it passes the test. |
| Verifying error processing for server busy | Demonstration | Statebased Testing | First lock the DB server using other team name, and try to reserve a trip,the system will be able to return the information of ‘Database Busy’, if doing so, it passes the test. |
| Verifying response time | Test | boundary testing | Given a set of legal parameters, record the time it uses to get the list of trips. If the response time less than 20sec, it passes the test. |
| Verifying response time  processing | Demonstration | Statebased Testing | Given a set of legal parameters, record the time it uses to get the list of trips. When the response time is bigger than 3 seconds, the system shows ‘exit’ are ready to click. If doing so, it passes the test. |
| Verifying Java Programming | Inspection | peer inspection | Peek the source code, if the code can be compiled and running by jvm, it passes the test. |
| Verifying platform independently | Demonstration | Equivalence testing | In different PCs, use a same legal trip parameter set to reserve a trip and record the time. If the times difference among them are less than 10sec, it passes the test. |
| Verifying fixing and extending ability | Inspection | peer inspection | If the code is able to extend (heritance, interface...), it passes the test. |

# Test Procedures

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Test Case: Verify initiating System | | | | Test Organization: jellycat | | | |
| Test Version: version 1 | | | | Execution Date: Apr 7 2019 | | | |
| Description: Enter the url of the product website as input, the website (UI) showing in the browser is observed. | | | | Executed By: jellycat | | | |
| SW Baseline: | | | |
| Preconditions: Server is online. | | | | | | | |
| Dependencies: A normal browser such as chrome | | | | | | | |
|  | | | | | | | |
| Step | Actions | Data | Expected Result | | Actual Results | Pass/Fail | Notes |
| 1 | Enter “localhost:8080/index” in the browser | “localhost:8080/index” | Homepage showed in the browser | | Homepage showed in the browser | Pass |  |
|  | | | | | | | |
| Postconditions: Homepage displayed in the browser | | | | | | | |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Test Case: Verify parameters entering correctly | | | | Test Organization: jellycat | | | |
| Test Version: version 1 | | | | Execution Date: Apr 7 2019 | | | |
| Description: Enter the parameters of trip as input (depart date, departure airport code, arrival airport code), if system get the same parameters from the UI and will tell if they are legal parameters, then it passes the test. | | | | Executed By: jellycat | | | |
| SW Baseline: | | | |
| Preconditions: the parameters of trip | | | | | | | |
| Dependencies: Homepage displayed in the browser | | | | | | | |
|  | | | | | | | |
| Step | Actions | Data | Expected Result | | Actual Results | Pass/Fail | Notes |
| 1 | Enter a valid parameter set | Depart BOS  Arrive MEM  One way  05/06/2019  Coach seat | True is returned from the assert | | True is returned from the assert | Pass |  |
|  | | | | | | | |
| Postconditions: All valid flights are displayed | | | | | | | |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Test Case: searchOneWayFlight | | | | Test Organization: jellycat | | | |
| Test Version: version 1 | | | | Execution Date: Apr 7 2019 | | | |
| Description: verify trips can be returned if search data filled | | | | Executed By: jellycat | | | |
| SW Baseline: | | | |
| Preconditions: Valid airport code and departure/arrival time are input. | | | | | | | |
| Dependencies: 1) Ability to search all possible flights based on input 2)Ability to display the flights which are found | | | | | | | |
|  | | | | | | | |
| Step | Actions | Data | Expected Result | | Actual Results | Pass/Fail | Notes |
| 1 | Search information are selected | Depart BOS  Arrive SFO  One way  05/05/2019  Coach seat | all the information can be selected | | all the information can be selected | Pass |  |
| 2 | Click the search button |  | A list of valid trips displayed | | A list of valid trips displayed | Pass | Displayed in 3 seconds |
| 3 | Verify each trip contains no more than 3 legs of flights |  | All trips contain no more than 3 legs of flights | | Each trip contains no more than 3 legs of flights | Pass |  |
| 4 | Verify departure airport and arrival airport |  | For each of the trips: first leg’s departure airport = “BOS”  Last leg’s arrival airport = “SFO” | | For each of the trips: first leg’s departure airport = “BOS”  Last leg’s arrival airport = “SFO”  If there it’s a direct flight, departure airport = “BOS” arrival airport = “SFO” | Pass |  |
| 5 | Verify seat type |  | For each trip the left coach seat number > 0 | | Each flight the left coach seat number > 0 | Pass |  |
| 6 | Verify departure date |  | For each trip the departure date is 05/05/2019 | | For each trip, the departure date is 05/05/2019 | Pass |  |
| 7 | Verify layover time |  | For a trip with multiple legs of flights, layover time is between 30 minutes and 2 hours | | For a trip with multiple legs of flights, layover time is between 30 minutes and 2 hours | Pass |  |
|  | | | | | | | |
| Postconditions: All valid flights are displayed | | | | | | | |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Test Case: localTimeConversion | | | | Test Organization: jellycat | | |
| Test Version: version 1 | | | | Execution Date: Apr 5 2019 | | |
| Description: verify that the system can convert a GMT time to local time | | | | Executed By: jellycat | | |
|  | | | | SW Baseline: | | |
| Preconditions: A GMT time and an airport code are provided. | | | | | | |
| Dependencies: 1) Ability to calculate timezone based on airport code2) Ability to get GMT offset based on time zone | | | | | | |
|  | | | | | | |
| Step | Actions | Data | Expected Result | Actual Results | Pass/Fail | Notes |
| 1 | Identify a valid airport code, GMT time | BOS, 2019 MAY 05 00:28GMT | Longitude and latitude of BOS airport | 42.366  -71.010 |  |  |
| 2 | Verify GMT offset | Longitude and latitude of BOS airport | -14400 | -14400 | Pass | -4 hours |
| 3 | Verify local time | -14400, 2019 MAY 05 00:28GMT | 2019 MAY 04 20:28 | 2019 MAY 04 20:28 | Pass |  |
|  | | | | | | |
| Postconditions: local time returned | | | | | | |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Test Case: Verify Determining Available Seat | | | | Test Organization: jellycat | | |
| Test Version: version 1 | | | | Execution Date: Apr 10 2019 | | |
| Description: Given a flight and a seat class (coach/first class), the available seat number should be given as supposed. | | | | Executed By: jellycat | | |
|  | | | | SW Baseline: | | |
| Preconditions: A flight number and a seat class | | | | | | |
| Dependencies: 1) Ability to retrieve airplane information | | | | | | |
|  | | | | | | |
| Step | Actions | Data | Expected Result | Actual Results | Pass/Fail | Notes |
| 1 | Identify a valid flight number and a seat calss | 2809 and Coach | True is returned from the method | True | Pass | 2809 have sufficient coach seat to reserve  31(reserved) 100(all) |
| 2 | Identify a valid flight number and a seat calss | 6258 and Coach | False is returned from assert | False | Pass | 6258 have insufficient coach seat to reserve  200(reserved) 200(all) |
|  | | | | | | |
| Postconditions: A boolean is returned | | | | | | |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Test Case: Verify sorting trips | | | | Test Organization: jellycat | | |
| Test Version: version 1 | | | | Execution Date: Apr 11 2019 | | |
| Description: Given a list of trips with the same departure, arrival airport code and departure date, a flag of which it wants to sort (price, departure date, arrival date), it should return a sorted list as supposed. | | | | Executed By: jellycat | | |
|  | | | | SW Baseline: | | |
| Preconditions: A list of trips | | | | | | |
| Dependencies: 1) Ability to retrieve trips from the DB server. | | | | | | |
|  | | | | | | |
| Step | Actions | Data | Expected Result | Actual Results | Pass/Fail | Notes |
| 1 | Retrieve trip list from DB server | Depart BOS  Arrive SFO  One way  05/05/2019  Coach seat | A list of trips with the same departure, arrival airport code and departure date | A list of trips with the same departure, arrival airport code and departure date |  |  |
| 2 | Click button of sorting by price |  | The list is sorted by price as more expensive as going down | The list is sorted by price as more expensive as going down | Pass |  |
| 3 | Click button of sorting by departure time |  | The list is sorted by departure time as more later as going down | The list is sorted by departure time as more later as going down | Pass |  |
| 4 | Click button of sorting by fight time |  | The list is sorted by fight time as longer as going down | The list is sorted by fight time as longer as going down | Pass |  |
|  | | | | | | |
| Postconditions: Sorting list is dislayed | | | | | | |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Test Case: Verify error processing for no available trips | | | | Test Organization: jellycat | | |
| Test Version: version 1 | | | | Execution Date: Apr 10 2019 | | |
| Description: verify that the system can respond correctly when no available trips for the request. | | | | Executed By: jellycat | | |
|  | | | | SW Baseline: | | |
| Preconditions: Legal parameter but no available trips exist for it | | | | | | |
| Dependencies: 1) Ability to do the search and realize there is no available trips exist 2) Ability to return notification about no available trips. | | | | | | |
|  | | | | | | |
| Step | Actions | Data | Expected Result | Actual Results | Pass/Fail | Notes |
| 1 | Input the legal parameters which lead to no available trips | "BOS","SFO","2019\_05\_05",0,2 |  |  |  |  |
| 2 | Verify the result and the user interface |  | No available trips showed and notification appears | as expected | Pass |  |
|  | | | | | | |
| Postconditions: local time returned | | | | | | |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Test Case: Verify after reserving | | | | Test Organization: jellycat | | |
| Test Version: version 1 | | | | Execution Date: Apr 10 2019 | | |
| Description: verify that the system can correctly do the reserving and show it to the user. | | | | Executed By: jellycat | | |
|  | | | | SW Baseline: | | |
| Preconditions: Legal reserving parameters are provided | | | | | | |
| Dependencies: 1) Ability to do the reserve and post it to the server 2) Proper UI to notify user about the result | | | | | | |
|  | | | | | | |
| Step | Actions | Data | Expected Result | Actual Results | Pass/Fail | Notes |
| 1 | Input the legal reserving parameters | Click reserve button on the web page. |  |  |  |  |
| 2 | Check whether changes reserved in Server |  | Related flight’s seat decrease. | As expected | pass |  |
| 3 | Check whether UI provide the result information correctly |  | “Successfully reserved” showed on screen | As expected | pass |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Test Case: Verifying response time | | | | Test Organization: jellycat | | |
| Test Version: version 1 | | | | Execution Date: Apr 20 2019 | | |
| Description: Provide legal parameters and search trips, record the response time and see whether it’s below 20 sec. | | | | Executed By: jellycat | | |
|  | | | | SW Baseline: | | |
| Preconditions: Legal parameters for searching trips are provided | | | | | | |
| Dependencies: 1) Ability to search trip when provided legal parameters. 2) Ability to record response time | | | | | | |
|  | | | | | | |
| Step | Actions | Data | Expected Result | Actual Results | Pass/Fail | Notes |
| 1 | Input the same legal parameters and record the response time. | "BOS","SFO","2019\_05\_05",0,2 | Below 20 sec | 3 sec | Pass |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Test Case: Verifying platform independently | | | | Test Organization: jellycat | | |
| Test Version: version 1 | | | | Execution Date: Apr 20 2019 | | |
| Description: verify the system can be run on different pc and platform and get similar result | | | | Executed By: jellycat | | |
|  | | | | SW Baseline: | | |
| Preconditions: Legal parameters for searching trips are provided | | | | | | |
| Dependencies: 1) Ability to search trip when provided legal parameters. 2) Several pc set the correct environment for the system and install it. | | | | | | |
|  | | | | | | |
| Step | Actions | Data | Expected Result | Actual Results | Pass/Fail | Notes |
| 1 | Input the same legal parameters on different pc. | "BOS","SFO","2019\_05\_05",0,2 |  |  |  |  |
| 2 | Compare the results |  | The systems on different PCs can all search trips correctly | As expected | Pass |  |
| 3 | Compare the response time |  | Response time difference is within 10 sec | Response time difference within 2 sec | Pass |  |
|  | | | | | | |
| Postconditions: local time returned | | | | | | |