

Requirements Analysis Document (RAD)

Prepared for

World Plane, Inc. (WPI)

Prepared by

Team Muse

Yahan Zhu, Marie Solman, Juexing Wang, Jiaoyan Chen

December 5, 2017

Version 3

Requirements Analysis Document (RAD)

1 Introduction

1.1 Purpose of the system

World Plane, Inc. (hereafter identified as WPI) wishes to evaluate the feasibility of transitioning our Travel Agency airline travel reservation system to a Retail Customer airline reservation system. In this project, a proof of concept Airline Reservation System will be built based on a small scale data-set in WPI server. This project will be implemented by using JAVA.

1.2 References

Statement of Work (SOW) provided by World Plane, Inc. 12 September 2017 and dated 09\05\2017.

Online Discussion: https://canvas.wpi.edu/courses/7309/discussion_topics/21824

In class discussion on 9/19/2017, 9/26/17, 10/3/17

1.3 Scope of the system

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, whether or not layovers are desired, 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.

1.4 Core System Functionalities

The application will be able to:

- System shall allow a customer make an airline reservation to travel from a departure airport to a destination airport of their choice.
- The system shall allow a customer to search for trips with a valid three character identifier of the departing and arriving airport.
- The system shall accept a date for departing date and return date, for round trips, in MM/DD/YYYY format.
- The system will ensure the return date is after the departing date.
- 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.

Team Muse Version 1

- 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.
- System shall allow customers to choose their type of seat: first-class seating or coach seating.
- System shall present a list of trips including direct flights and connecting flights with up to two layovers.
- 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.
- System shall allow the user to sort list of trips by price in USD, local departure time, local arrival time and travel time.
- System shall respond when seating is not available.

1.5 Objectives and Success Criteria of the Project

The success of the application depends upon meeting the following core set of objectives:

- The system shall display all trips with available seating that match a customer's search criteria. If a trip contains a connecting flight with no available seating, the entire trip shall not display.
- The system shall allow a customer to reserve seating with zero, one, or two stopovers to travel from departure to destination airport.
- The system shall allow a customer to sort results based on price in USD, departure time in local time for the departing airport, arrival time in local time for the arriving airport, travel time, and class type.
- The system shall have good response time and provide feedback within 10-15 seconds.
- The system shall update the server when a first class or coach seat has been purchased for a flight and update whether a coach or first class seat has been reserved for a specific flight.

2 Current System

2.1 Existing System

The existing system allows the user to use a series of HTTP GET and HTTP POST commands to retrieve information from the server and then post an update. The system allows the user to use a GET command to see a full list of available airports, all types of planes available, what flights leave or arrive at a particular airport, and how many coach or first class tickets that have been reserved on each flight. It also allows the user to reserve a first class or coach seat for an individual flight but it does not link any customer information.

The new system shall improve the customer experience and provide a more streamlined way to search for a one way or round trip on a specific departure and return day. The system shall also calculate and present possible indirect flights and allow the customer to filter these trips by time

Team Muse Version 1

windows on the departure date and return date if any. The new system shall be improved to enhance the customer experience but the existing system will not be replaced.

2.2 Current Operations

The current system presents a large amount of raw data to the user but it does not sort trips or link connecting flights for the user. The new system shall present potential customers with the ability to search for and reserve trips and will be much more robust than the existing functionality.

Because of this, current operations have not been heavily considered in the design.

3 Proposed System

3.1 Overview

This section provides a functional overview of the system. This will again be properly be divided into two parts

3.2 Functional Requirements

- Customers shall be able to specify the departure airport they wish to travel from and the arrival airport they wish to travel to using the 3 character identification code of the respective airports.
- Customers shall be able to specify the date of leaving and return in local time of the departure airport.
- Customers shall be able to select first class or economy reservation for travel between departure and destination airport.
- Customers shall be able to select one way trip or round way trip for travel between departure and destination airport.
- Customers shall be able to confirm their choice of airline before the system reserve the seat(s).
- Customers shall be able to sort results by price, departure time, arrival time, and travel time. But only one strategy can be applied at each sorting.
- The system shall allow a maximum of 2 stopovers when traveling from departure to destination airports. (The system will support 0, 1 or 2 stopovers when traveling from departure to destination airport.)
- System shall present list of trips including direct flights and connecting flights with layovers.
- The system shall display a trip's departure airport name, departure local time, arrival airport name, arrival local time, total flight time and price as a result of customer's search.

Team Muse Version 1

- The system shall update the number of seat reserved in database when ticket has been confirmed.
- The system shall NOT collect customer information.
- The system shall NOT allow refunds once tickets have been purchased.

3.3 Nonfunctional Requirements

3.3.1 Usability

- The system shall provide concise instruction for user's each step.
- 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
- The system shall be stable and validate user inputs to ensure they are correct and as expected.
- The system shall ensure departure time is not after arrival time.
- The system shall notify the user if no trips are available for the user's search criteria.
- The system shall notify the user when the system is searching the airlines

3.3.2 Reliability

- Data input validators will be unit tested using Junit test cases developed in parallel with application software.

3.3.3 Performance

- All the phases during the reservation should return all results accords to user's requirement,
- 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.
- The software shall be compatible on the client machine.

3.3.4 Supportability

- The system shall use the JAVA programming language for platform independence.
- The system shall request XML files from the server using HTTP GET commands, and retrieve information from the XML files.

3.4 System Models

3.4.1 User Scenarios

Team Muse Version 1

This section lists the scenarios defining the detailed operations the system will need to support. Each user scenario will define a role, an action, and potentially a reason the role wants to perform the action.

1. As a student I want to purchase the least expensive ticket available to travel home after my school semester completes.
2. As a businessman I want to purchase a round trip ticket within very specific time spans for my business trip.
3. As a couple I want to purchase a luxurious first class trip for our honeymoon.
4. As a family member experiencing an emergency, I want the next available trip to my desired destination so I can get there as soon as possible.
5. As a traveler who has never seen Chicago, I want to reserve an indirect trip through Chicago with a maximum layover time so I can see the city.
6. As a pet owner, I want to purchase a ticket with no layovers to ensure my pet is on the plane for as little time as possible.
7. As a frequent traveler, I want to purchase the least expensive round trip ticket for my departure and return dates to save money for future trips.
8. As a patient flier, I want to purchase a one way trip with two layovers to see multiple airports.
9. As a CEO, I want to purchase a trip at a specific time since I don't have much time for travel.
10. As a person with lots of time, I want to reserve a direct flight to my destination to get there quickly but an indirect flight home so I can see multiple cities on my way back.
11. As a flier depending on a family member picking me up, I want a trip that will arrive at an airport at a specific time so I won't make them wait.

3.4.2 Use case model

Name:	Purchase Least Expensive Ticket
Actor:	Customer Data Server
Entry Conditions :	Logged into the web

Flow of Events:	<ol style="list-style-type: none"> 1. Customer inputs departure date, departure airport, and arrival airport for one way trip. 2. System validate user's inputs 3. System will retrieve flight information from the data server and filter trips that meet the search requirements 4. System will convert times to local times and show all the available reservations. 5. Customer will sort by price 6. Customer will select least expensive ticket 7. System will show the detail of the reservation and wait for client's confirmation. 8. Customer will confirm the reservation and submit purchase 9. System will update data server that a ticket has been purchased
Exit Conditions :	Customer purchased ticket successfully

Name:	Purchase Round Trip Ticket
Actor:	Customer Data Server
Entry Conditions :	Logged into the web
Flow of Events:	<ol style="list-style-type: none"> 1. Customer inputs departure and return dates and airports for round trip 2. System validate user's inputs 3. System will retrieve flight information from the data server and filter trips that meet the search requirements 4. System will convert times to local times and show all the available reservations. 5. Customer will select desired departure trip 6. Customer will select desired return trip 7. System will show the detail of the reservation and wait for client's confirmation. 8. Customer will confirm the reservation and submit purchase 9. System will update data server that a ticket has been purchased
Exit Conditions :	Customer purchased ticket successfully

Name:	Purchase a luxurious first class
Actor:	Customer Data Server
Entry Conditions :	Logged into the web
Flow of Events:	<ol style="list-style-type: none">1. Customer inputs departure and return dates and airports for round trip2. System validate user's inputs3. System will retrieve flight information from the data server and filter trips that meet the search requirements4. System will convert times to local times and show all the available reservations.5. Customer will filter the trips with first class tickets available.6. Customer will sort departure trips by time7. Customer will select desired departure trip8. Customer will sort the return trips by time9. Customer will select desired return trip10. Customer will confirm the reservation and submit purchase11. System will update data server that a ticket has been purchased
Exit Conditions :	Customer purchased ticket successfully

3.4.3

Name:	Purchase Next Available Trip
Actor:	Customer Data Server
Entry Conditions :	Logged into the web
Flow of Events:	<ol style="list-style-type: none">1. Customer inputs departure date as today and departure and arrival airports2. System validate user's inputs3. System will retrieve flight information from the data server and filter trips that meet the search requirements

	<ol style="list-style-type: none">4. System will convert times to local times and show all the available reservations.5. Customer will sort by departure time and determine the next available trip6. Customer will select the next available trip7. Customer will confirm the reservation and submit purchase8. System will update data server that a ticket has been purchased
Exit Conditions :	Customer purchased ticket successfully

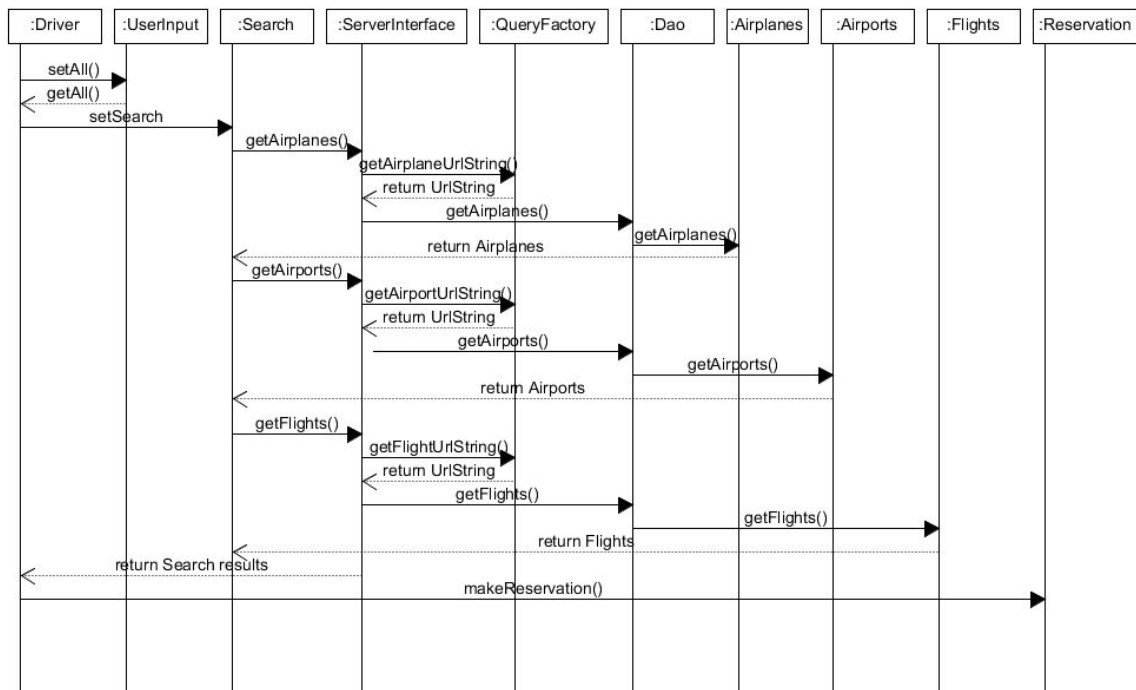
3.4.4

Name:	Purchase Fastest Trip
Actor:	Customer Data Server
Entry Conditions :	Logged into the web
Flow of Events:	<ol style="list-style-type: none">1. Customer inputs departure date and airport and destination for one way trip, and sets the stopovers times to 0. Then click search.2. System validate user's inputs3. System retrieves flight information from the data server and filter trips that meet the search requirements.4. System will convert times to local times and show all the available reservations.5. Customer click "duration" on the page.6. System sort the trips by duration and display the sorted result.7. Customer buy the fastest airplane.8. Customer will confirm the reservation and submit purchase9. System will update data server that a ticket has been purchased
Exit Conditions :	Customer purchased ticket successfully

3.4.5 Object model

3.4.5.1 Static model

3.4.5.2 Dynamic model



Sequence Diagram Based on “Purchase Next Available Trip” use case model

Communication Diagram based on “Purchase the Cheapest Ticket” user case

4 Glossary

<u>Reservation</u>	A seat on a specific flight specifying either 'First Class' or 'Economy' seating section of the plane. A reservation does not specify a particular seat number for the flight.
<u>Customer</u>	Actor purchasing a flight reservation
<u>System</u>	The software which allows a customer to look up and purchase reservations stored on the database
<u>Stopover</u>	A period of time between connecting flights at an airport
<u>Trip</u>	Travelling from a departure airport to an arrival airport, whether there are stopovers or not
<u>Flight</u>	Travel between two airports on an airplane, stored in database
<u>Direct Flight</u>	A trip with no stopovers
<u>Indirect Flight</u>	A trip with one or two stopovers
<u>Connecting Flight</u>	Leg of a trip involving a stopover
<u>Local Time</u>	Time as standard within the time zone of the airport
<u>GMT Time</u>	Time as standard at Greenwich Mean Time, the time zone in which time is stored on the database
<u>First Class</u>	Premier seating on a flight
<u>Coach</u>	Standard seating on a flight
<u>Price</u>	Cost of a trip in USD.
<u>Departure Time</u>	Time a flight leaves an airport displayed in local time
<u>Return Time</u>	Time a return flight leaves its departing airport in local time
<u>Arrival Time</u>	Time a flight arrives at an airport displayed in local time
<u>Departure Airport</u>	Airport where a trip begins

Requirements Analysis Document (RAD)
Team Muse Version 1

Airline Reservation System

<u>Arrival Airport</u>	Airport where a trip ends
<u>One Way Trip</u>	Trip purchased with no return trip
<u>Round Trip</u>	Trip purchased with a return trip
<u>Travel Time</u>	Duration of entire trip