Requirements Analysis Document (RAD)

Prepared for
World Plane, Inc. (WPI)

Prepared by
Team SSR
Yujun Tian,
Gunnar Horve,
Linan Xu,
Xiaoting Cui

28 April 2017
Version 1.2

1 Introduction

1.1 Purpose of the system

World Plane, Inc. (hereafter identified as WPI) serves people as an flight ticket reservation company. It has its own airline flight system, but once it relied on the Travel Agency airline reservation system. WPI decided to develop a small scaled application for customer by Java language, so that customers can order flight on WPI's Retail Customer airline reservation System.

1.2 Scope of the system

This system offers help for customers to create airline reservations with World Plane, Inc.. Also, the system supports sorting flights by either price or time. In addition, the system is to support both one way and round trip flights, initially queryable by departure date. Lastly, when reserving the seat, the system will support selection of which class the seat belongs to and etc.

1.3 Core System Functionalities

The application will be able to:

- Allow a customer make an airline reservation to travel from a departure airport to a destination airport of their choice.
- Select first class or economy seating for travel.
- Provide a series of connecting flights with a maximum of two stopovers between departure and arrival place.
- Choose one-way or round-trip flight.
- Show the departure time and arrival time of each leg.
- Search for flights using departure date, arrival dates and respective time windows within each.
- Sort the list of flight according to price or travel time.
- Select flights and confirm the selection prior to the reservation being made.
- Show the price of flight between the departure and arrival place.
- Select flights and confirm before the reservation being made.

1.4 User Stories

1. As a customer, I want to select a departure airport and arrival airport for all the direct or connecting flights, so I can get a list of results for my search.

2. As a customer, I want to choose one-way or round-trip flight, because I can make sure that I can come back in a certain date.

3. As a customer, if I want a one-way trip, I can enter the date, departure airport and arrival airport to get the flight which is best matched with my schedule.

4. As a customer, if I want a round-trip, I can enter the date, departure airport and arrival airport for the leaving flight and enter the information again for the return flight.

5. As a customer, I want to know the departure time(local time) and arrival time(local time) of each leg, because I need to adjust my flight according to my schedule.

6. As a passenger, I want to choose either first class seating or coach seating, because I want to choose which type of seats I prefer.

7. As a customer, I want to search for flights using departure date, arrival dates and respective time windows within each, because it is much more convenient for me to choose which flight I prefer.

8. As a customer, I want to reserve seats for each leg of the flights.

9. As a customer, I want to sort the list of flight according to price or travel time, because I want to choose one which fits me well.

10. As a customer, I want to select flights and confirm the selection prior to the reservation being made, so that I can review the flight information before my final decision.

11. As a customer, I want to have no more than 2 stopovers, or 3 total flights to get from a starting airport to a destination airport.

12. As a customer, I want to have reasonable stopover times, or stopover times between 30 minutes and 4 hours.

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 will allow a customer to reserve a flight specified by:
    - Flight Type (one-way or round-trip)
    - Departure airport
    - Arrival airport
    - Seating class (consistent throughout all legs of the flight)
    - Departure date
- The System will show customer a list of all results sorted according to price or travel time that customer entered in.
- The System will allow customer to review flight information before making a final reservation decision.

1.6 Definitions, Acronyms, and Abbreviations

Reservation	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.
Stopover time	The time used for customer to transition from the gate of the arrival flight to the gate of the next departing flight.
GTM	Green Mean Time
One-way	From departure to destination
Round-trip	From departure to destination and back to original departure airport
API	Application Programming Interface

1.7 References

· Statement of Work (SOW) provided by World Plane, Inc. 26 January 2017 and dated 05/16/16.

2 Proposed System

2.1 Overview

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

2.2 Functional Requirements

1. Airport Selection: Customers shall be able to specify the departure airport they wish to travel from and the airport they wish to arrive at.
2. Stopover Selection: Customers shall be able to specify the maximum number of stopovers when traveling from departure to destination airports.
3. Stopover Limit: 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.)
4. Class Selection: Customers shall be able to select first class or economy reservation for each flight between departure and destination airport.
5. Layover Range: The system will only display flights whose layover time is between 30 min and 4 hours.
6. Time Display: The system will show the departure time and arrival time of each leg which will be displayed airport local time.
7. Roundtrip Selection: Customers will have the ability to reserve flights to travel either one-way or reserve a round-trip flight.
8. Date Selection: Customers will be able to search for flights using departure date.
9. Roundtrip UI: Customers will be able to perform a round trip flight reservation as a sequence of two single-direction flight reservations
10. Pricing Display: The system will display the pricing of flights from departure to destination.
11. Flight Sorting: The system will allow customer to display flights sorted by price or travel time.
12. Confirmation Screen: The system will allow the customer to select flights and confirm the selection prior to the reservation being made.
13. Confirmation UI: The system will allow customer to select and reserve flights from a searching list.
14. No Deletions: The system will not allow a reservation to be deleted once it is made.
15. Request Method: The system should be able to request data using HTTP Get queries.
16. Server Reading: The system should be able to read data from XML returned from database.

2.3 Nonfunctional Requirements

Reliability

1. Tests Exist: All classes and use cases will be unit tested through the use of a test plan document
2. No Space UI: The system will be able to return the information of 'there is not seat' when requested seat is not available for all legs of the flight.
3. Servery Busy UI: The system will be able to return the information of 'server is busy' to customer when the database is locked by others.
4. No race cond: The system should be able to lock the database to prevent race conditions

2. Performance

1. Fast UI: Response time for any requested actions will reasonable. Operations in excess of 3 seconds will provide indication to the customer the system is operating.
2. Search Speed: The system should provide results to the customer's searching action within 60 seconds

3. Supportability

1. Code is JAVA: The application will use the JAVA programming language for platform independence.

2.4 System Models

This is the section where you present the use cases your team has developed. For a project of this scope I would expect somewhere between 5 or 6 use cases on the low end and 10 or 12 on the high end.

2.4.1 Use cases

Name:	Search for a flight
Actor:	Customer, server
Entry	+User Clicked "Search" after entering search parameters

Conditions:	
Flow of Events:	1. Customer clicks “search” 2. Server Queried for flights 3. Flights parsed into java Flight objects 4. Flight objects collected into ObservableList<ArrayList<Flight>> 5. (4) searched for valid flight paths
Exit Conditions:	● Viable flight paths found ● No flight paths found ● User Canceled Search

Name:	Reserve a flight
Actor:	Customer, server
Entry Conditions:	User has reviewed flight selection, clicks “reserve”
Flow of Events:	1. customer clicks “reserve” 2. System locks the database 3. System modifies database to reflect reservation 4. System unlocks the database
Exit Conditions:	● Reservation failed -- concurrency issue ● Reservation succeeded

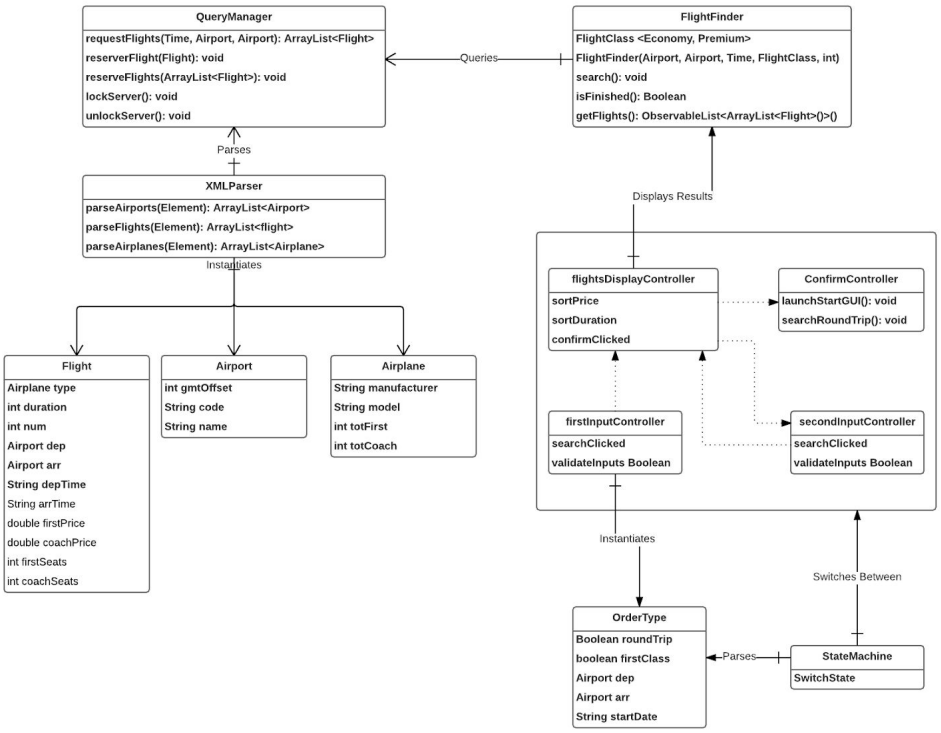
Name:	Sort the flights by (x criteria)
Actor:	Customer
Entry Conditions:	Valid flight paths are computed, customer selects sort type
Flow of	1.Customer chooses sort type (price, duration, layover time, …)

Events:	2. System performs sorting operation 3. System updates GUI to reflect resulting sort
Exit Conditions:	Valid flight paths are computed (same as entry condition)

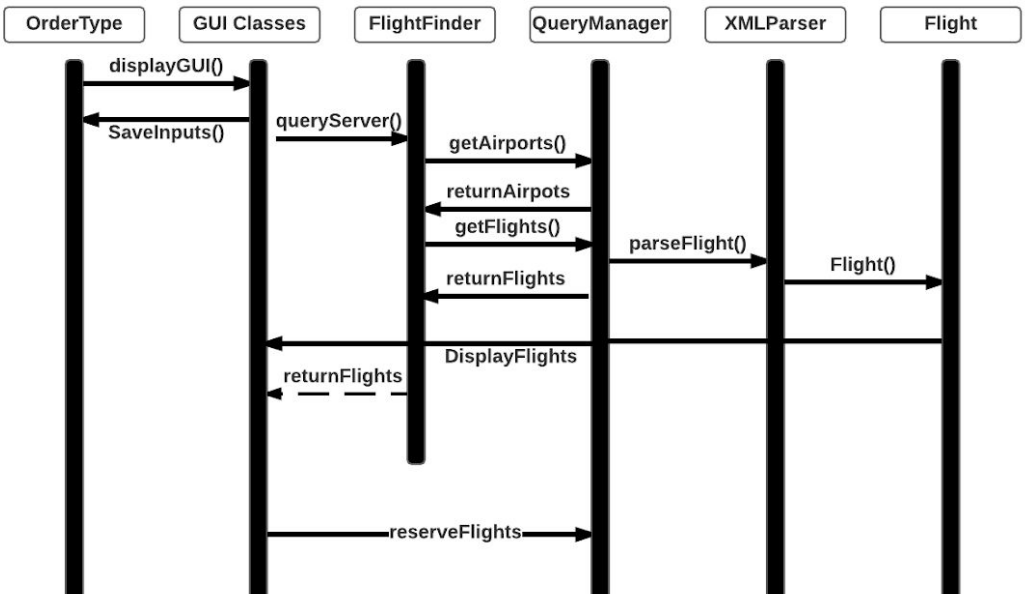
Name:	Review flight information
Actor:	Customer, server
Entry Conditions:	Customer selects desired flight
Flow of Events:	1.customer can browse the order that he/she chose 2.customer can review the flight information before he/she makes a confirmation
Exit Conditions:	● Customer clicks “reserve” ● Customer cancels order ● Customer clicks “back”

2.4.2 Object models

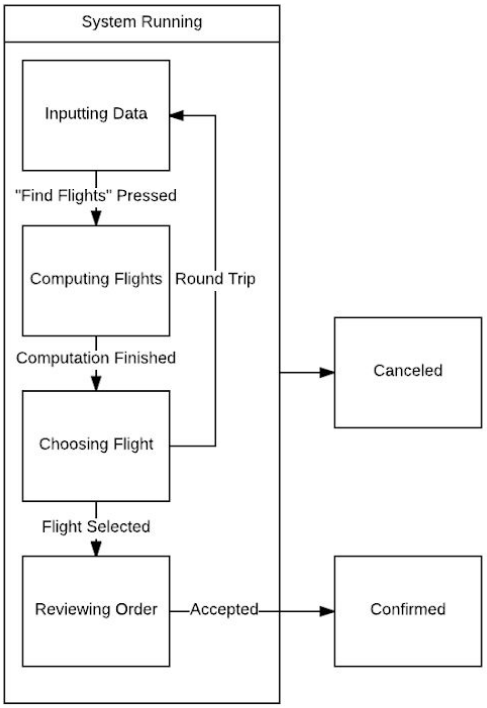
2.4.2.1 Class Diagram



2.4.2.2 Interaction Diagram



2.4.2.3 State Diagram



2.4.2.4 Activity Diagram