

Requirements Analysis Document (RAD)

Prepared for
World Plane, Inc. (WPI)

Prepared by
Team jellycat
Dongrui Qi, Wei Xiong, Zhuoqun Yuan, Yueyue Zhong

Feb 28, 2019

[version 1]

Requirements Analysis Document (RAD)

1 Introduction

1.1 Purpose of the system

WPI (World Plane, Inc) has an airline flight database running on an Internet accessible server, which is not comprehensible for common clients who want to book flights based on that. Our proof of concept software is browser-based and is 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.

Commented [BN1]: Proof of concept?

1.2 References

- Statement of Work (SOW) provided by World Plane, Inc. 24 January 2019 and dated 05/10/2019.
- Lecture slides
- Object-Oriented Software Engineering Using UML, Patterns, and Java (3rd Ed) by Bruegge, Bernd, Dutoit, Allen H.

Commented [BN2]: good

1.3 Scope of the system

This system designs and implements concepts of an airplane reservation system. Supported by WPI airline flight database, all available air flight information can be accessed easily through a standard HTTP GET Application Programming Interface (API). A friendly customer interface is provided so that various combinations of search criteria can be fetched from the customer and corresponding database search results can be generated and sorted if prompted by customers. Customers can make reservations on the system based on their specific preferences. The system provides a solution allowing the customer to search for desired travel (like the departure airport and arrival airport), flight date (local date and time for both departure and arrival), one-way trip or round way trip. And customers can sort the results by price (in dollars), departure time, arrival time, travel time, or seat type (coach seat or first-class seat). Then the customer can reserve seats on preferable flights.

1.4 Core System Functionalities

- 1) Create a proof of concept system that can be used by customers to reserve an airline flight.
- 2) Ensure that airline tickets reservation system is easy and customer friendly for customers.
- 3) Widely Search and Precise Search for locating flight list and applying customer preference.
- 4) Display all needed information for each flight based on customers' requests.
- 5) Choose one-way or round-trip flight.

Commented [BN3]: good

- 6) Display the local departure time and arrival time of each leg.
- 7) Customers can sort the flights results based on all-together price.
- 8) Customers can sort the flights results based on departure time.
- 9) Customers can sort the flights results based on arrival time.
- 10) Customers can sort the flights results based on travel time.
- 11) Customers can reserve first class or coach seating for travel through the system.
- 12) Select flights and confirm the selection prior to the reservation being made.
- 13) Select flights and confirm before the reservation being made.
- 14) Customers can exit the system and return to the initial state.

1.5 Objectives and Success Criteria of the Project

Commented [BN4]: Nice job

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

- 1) Deliver a proof of concept system that can be used by customers to reserve an airline flight.
- 2) The system is easy and customer friendly for customers.
- 3) The system can make reasonable response to unsuccessful search.
- 4) Timeliness response to customer's actions is guaranteed.
- 5) Ensure an easy to use and conflict-free reservation system for customers to reserve a flight.

2 Current System

2.1 Existing System

The current system is operated by travel agents employed by World Plane Inc (WPI.) When travelers want to fly from departure airport to destination airport, they will engage with a WPI travel agent who uses existing proprietary software to book the flights for the customer.

And an API to connect with the Database. Through current API, the system allows customers to make query about airports, airplanes and airports arrive and depart airports, and also allow developers to reset the database. After improvement, customers can make queries in a more customer-friendly way, and can also add or modify data through input. The original system will still be used to get the source data.

2.2 Current Operations

Currently, travel agents employed by World Plane Inc (WPI.) When travelers want to fly from departure airport to destination airport, they will engage with a WPI travel agent who uses existing proprietary software to book the flights for the customer.

And there is an API where we can make queries, and get XML containing queried information. More current operations have not been provided.

3 Proposed System

3.1 Overview

This system represents the initial version of the Airline Reservation System. At a high level, this system will allow a customer to check flights, sort flight results, and reserve seats on flights. The goal is to allow customers greater and easier access to the airline's booking system at any time.

3.2 Conceptual Model - Customer Scenarios

- 1) The customer is permitted to search available flights based on the departure airport, arrival airport, departure date, arrival date.
- 2) The customer can select one-way or round-trip flight.
- 3) The customer can see any matching flights (including connecting flights with a maximum of two stopovers) and their prices based on the search criteria entered.
- 4) The customer can see all the legs of flights in the airport local time.
- 5) The customer can see seats availability (first class or coach seating) and their prices for each leg of flight.
- 6) If no flights or seats are available, customer will be notified by the information of "no available flights" and will be guided to return to a new search page.
- 7) Customer can sort flights by price, departure time, arrival time or travel time and results will display accordingly.
- 8) The customer can modify departure airport, arrival airport, departure date, arrival date, one-way or round-trip flight before choosing the flight.
- 9) The customer can choose an available seat for each leg of flight he/she wants.
- 10) The customer wants no more than 3 stopovers, which means at most 3 legs of flight in each way (from starting airport to destination airport), and each stopover time can be no less than 30 minutes and less than 2 hours.
- 11) Subsequently, the customer will be asked to confirm the selected flight before making a reservation.
- 12) The customer can cancel the selected seats before confirmation.
- 13) The customer actions will be guaranteed to have timeliness of response.

Commented [BN5]: Remember that scenarios are about an actor doing something significant and representative with the system – from the actor's perspective. This would be something such as booking a one way flight after the end of semester. This is a bullet list that reads more like functional requirements.

It doesn't communicate to me that you have your arms around why these functions would be done or how the disparate set of functions fit together to do something meaningful. Revisit Brugge & Dutoit section 4.4.2 and class lessons.

3.3 Functional Model - Use Case Model

3.3.1 Initiate System

Name:	Initiate System
Actor:	Customer
Entry Conditions:	Customer double click the app or browse the website.
Flow of Events:	1.The customer clicks on the app/ tries to enter the website. 2.The system shows front page in reasonable time response.

Commented [BN6]: This should be state of the system instead of an action. You have this action as the first in the flow (which makes sense.) The state of the system would be something like 'system running' or 'system not initialized'

Exit Conditions:	<ul style="list-style-type: none"> The system displays front page and preferences for customers to choose.
-------------------------	---

3.3.2 Entry of search parameters

Name:	Entry of Search Parameters
Actor:	Customer
Entry Conditions:	Customer has opened the main website or app.
Flow of Events:	<ol style="list-style-type: none"> The customer sees the One-Way/Round-Trip selection (default one-way), one departure airport entry and one departure time entry. The customer identifies One-way. Then enter departure airport arrival airport, and departure date and departure time windows. The customer identifies Round-trip. Then one more choice will be displayed to the customer identified as return date and return time windows.
Exit Conditions:	<ul style="list-style-type: none"> All parameters have been identified (except time windows which can be left empty). The customer clicks on the "Search" Button.

Commented [BN7]: What does an actor do, not what an actor sees or thinks... You can write software to react to stimulus. It's not so easy to write software to react based on what an actor experiences or what they hope for.

Commented [BN8]: Aren't there parameters besides those specified? How about first class or economy seating?

In the flow what does the system do in response to the actor stimulus?

Commented [BN9]: This is an action and belongs in a flow. It is not a system state.

3.3.3 Searching for connecting flights

Name:	Searching for connecting flights
Actor:	Database Server, customer
Entry Conditions:	Customer clicked "search" after entering search parameters.
Flow of Events:	<ol style="list-style-type: none"> Customer clicks "search" The system will process the customer's search parameters to HTTP query. Database Server get the parameters from the system and process the raw information including converting time zones and determining available seating. Database Server send back the information based on the query from the system. System shows the list of satisfying flights to the customer (satisfying flights mean there are available seats in first class or coach seating, or

Commented [BN10]: This is very simplified and incorrect. This is most of what the system needs to do (based on customer's input find all potential flights.) So there's a lot of steps to achieve this which are not addressed in the detail they should be.

Commented [BN11]: There's a number of different queries which must happen for the client to get enough information from the server for trying to find connecting flights.

Commented [BN12]: The server does not do this. Your software needs to do this.

	both left for customers to choose from).
Exit Conditions:	<ul style="list-style-type: none"> • Viable legs of flights found • No flights path found • Customer canceled search

3.3.4 Converting time zones

Name:	Converting time zones
Actor:	Database Server
Entry Conditions:	Database sends in flight information including GMT and longitude.
Flow of Events:	<ol style="list-style-type: none"> 1. Database sends in flight information including GMT and longitude. 2. System get flight information from Database server including GMT and airport longitude. 3. System calculates the local time based on the GMT and the airport longitude. 4. System return the local time to where the system calls for it.
Exit Conditions:	<ul style="list-style-type: none"> • System return the local time to where the system calls for it.

Commented [BN13]: How?

Commented [BN14]: I am not sure what this means.

3.3.5 Choosing available seating

Name:	Choosing available seating
Actor:	Customer, Database Server
Entry Conditions:	There are already valid results from previous flight search and customer selects seat preference ("first class" button or "coach seat" button) from page.
Flow of Events:	<ol style="list-style-type: none"> 1. Customer choose a seat preference from the current flight results. 2. System looked for available seat for each flight (to trip and return trip separately) previously shown on the page from Database Server. 3. Only the flights that have the desired seating will be displayed to the customer.
Exit Conditions:	<ul style="list-style-type: none"> • System return valid result. • No valid results.

Commented [BN15]: How does the system look for available seats? There are multiple pieces of information that need to be analyzed. What are they and how are they analyzed?

Commented [BN16]: What does valid result mean?

3.3.6 Sorting of flight data

Name:	Sorting of flight data
Actor:	Customer
Entry Conditions:	Valid flights are displayed, customer select one of the sorting buttons based on: price (low to high), departure time (earliest to latest), arrival time (earliest to latest), travel time (shortest to longest).
Flow of Events:	<ol style="list-style-type: none"> 1.Customer select one of the sorting buttons based on: price, departure time, arrival time, travel time for both to trip and return trip (if round trip applied) respectively. 2.The system will use the flight information to sort the specific term and display the new sorted flight list to the customer.
Exit Conditions:	<ul style="list-style-type: none"> • Valid flight list is returned. • Customer cancel the sort condition.

Commented [BN17]: How? Sorting by flight time is different that sorting by departure time.

Commented [BN18]: What is valid? Who is it returned to?

Commented [BN19]: This is an action. It should be addressed as part of the flow of events and the system would need to react to it in some manner.

3.3.7 Reserving a travel

Name:	Reserving a travel
Actor:	Customer, Database Server
Entry Conditions:	Customer click on the confirmation/ reserve button after he/she choose the desired legs of flight.
Flow of Events:	<ol style="list-style-type: none"> 1.Customer click on the confirmation/ reserve button. 2.System gathers the flight numbers of all the legs in this flight. 3.System locks the Database Server. 4.System send the reserve query one by one. 5.System unlock the Database Server. 6.Customer will be acknowledged of successfully reserving the tickets.
Exit Conditions:	<ul style="list-style-type: none"> • Reservation failed (database locked by others). • Reservation canceled by customer. • Reservation succeeded.

3.3.8 Error processing for no available flights

Name:	Error processing for no available flights
Actor:	Database Server, Customer
Entry Conditions:	After the customer entering all the needed parameters and clicks "search" button.
Flow of Events:	<ol style="list-style-type: none"> 1. System looked up for available flights from database. 2. Database return a blank list or false information.

	<ol style="list-style-type: none">3. System recognize the blank list or false flags.4. System displayed “No flight available found!” information to the customer.5. System displayed a “Restart” button to the customer.
Exit	<ul style="list-style-type: none">• Customer select “Restart” button.
Conditions:	<ul style="list-style-type: none">• Search canceled by customer.

3.3.9 Error processing for no requested seats for all legs of flight

Name:	Error processing for no requested seats for all legs of flight
Actor:	Customer, Database Server
Entry Conditions:	There are already valid flight results for customer’s preferences. Customer wants to select first-class seating for all legs of flight on his/her one-way trip.
Flow of Events:	<ol style="list-style-type: none">1.Customer select the first-class button.2.System find no flight for all legs meets the requested class.3.System shows “No available seat found, please choose another kind of seating or restart.” And a “Restart” button.
Exit Conditions:	<ul style="list-style-type: none">• Customer choose “coach seating”.• Customer click “Restart”.• Search canceled by customer.

3.3.10 Cancel before confirmation

Name:	Cancel Before Confirmation
Actor:	Customer
Entry Conditions:	There are valid flights based on his/her preferences for the customers to choose. Customer select “cancel” before reserving a flight.
Flow of Events:	<ol style="list-style-type: none">1.There are valid flights based on his/her preferences for the customers to choose.2.Customer clicks “reserve” for a desired flight.3.System pop out a dialog with two buttons: “Confirm” and “Cancel”4.Customer select “Cancel” button.5.System returned to valid flights page as in step 1.
Exit Conditions:	<ul style="list-style-type: none">• Customer clicks “Cancel”.• Customer clicks “Confirm”.

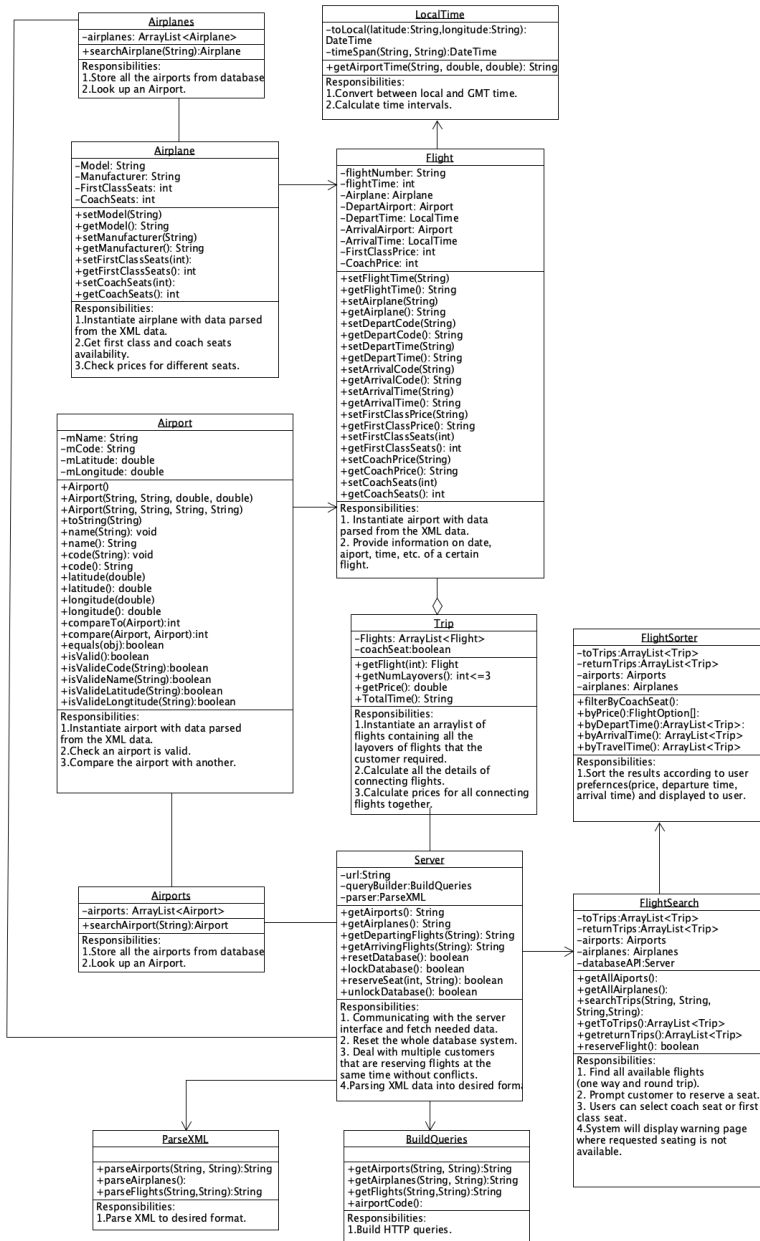
3.4 Analysis Model – Object Model

The picture below is the object model of the system, GUI end not included in this model. Boundary object and control objected are also not included.

Commented [BN20]: You're off to a good start. Most classes have cohesive responsibilities. As you decompose the design you should consider further decomposition of some classes. For example, I don't see the boundary class to support customer and doubt that FlightSearch should be prompting customer for information.

Requirements Analysis Document (RAD) Template Version 2

Airline Reservation System



4 Requirements

4.1 Functional Requirements

- Customers shall be able to specify one-way or round-trip flight they hope to reserve.
- Customers shall be able to specify the departure airport they wish to travel from and the arrival airport they wish to travel to.
- Customers shall be able to specify the departure date and arrival date for the travel.
- Customers shall be able to specify the departure time window and arrival time window for the travel.
- Customers shall be able to see the arrival and departure time in airport's local time zone.
- Customers shall be able to sort qualified flight data according to price, number of stopovers, or flight duration.
- Customers shall be able to get access to flight detailed page from the search page.
- Customers shall be able to reserve flight from flight detailed page.
- Customers shall be able to specify their seating class for the selected flight.
- Customers shall be able to reserve seating number after reserving the flight.
- Customers shall be notified no available data if no flight qualifies customers' requirements.
- Customers shall be able to cancel the order before they confirmed it.
- The system shall adjust time zone according to each customer's location.

Commented [BN21]: Locking server?
HTTP, XML?

Commented [BN22]: When the customer specifies something, what happens? It's good that a customer can specify one-way or round trip, but doesn't the system have to show flights matching the criteria? Just supporting the input choices isn't sufficient.

Commented [BN23]: What is the flight detailed page?

Commented [BN24]: What does this mean? I don't understand.

Commented [BN25]: Not correct. The local time is the airport local time of arriving and departing flights, not the local time where the customer is located.

4.2 Nonfunctional Requirements

4.2.1 Usability

- Customers shall be able to specify the departure airport and arrival airport from a list of possible airports provided by the system.
- Detailed and understandable customer documentation and manual shall be provided.
- The software will follow the customer interface guidelines and principles.

Commented [BN26]: This is not in agreement with your scenarios or use cases. I didn't see where the system was providing a list to choose from

Commented [BN27]: You're going to deliver a user manual? When I use Kayak I don't read a user manual first.

Commented [BN28]: What is this? What guidelines and what principals?

4.2.2 Reliability

- All classes will be unit tested using Junit test cases developed in parallel with application software.
- The system shall function correctly to a reasonable degree in the presence of invalid inputs or stressful environment conditions.

Commented [BN29]: How would you test this for verification you have met the requirement? What is a reasonable degree?

- The system will be able to return information of 'No seat available' when requested seat is not available for all legs of flights.
- The system will be able to return the information of 'server is busy' to customer when database is locked by others.
- The server shall withstand specified security attack and keep high reliance.

Commented [BN30]: This sounds great, but how do you test? What security attacks and what is high reliance?

4.2.3 Performance

- Response time for any requested actions will reasonable. Operations in excess of 3 seconds will provide indication to the customer the system is operating.
- The number of customers request the system can process with a specified amount of time shall be reasonable.
- Accurate and complete results shall be output according to the customer request.

Commented [BN31]: How do you test this for pass / fail? What is reasonable?

4.2.4 Supportability

- The application will use the JAVA programming language for platform independence.
- The application shall be run on different hardware and software environment and get the same performance.
- The application shall have the ability to be maintained to fix defects or add functions.
- The application will be able to adapt to international conventions such as languages, phone numbers and airplane industry conventions.

Commented [BN32]: Really? Different hardware won't effect the performance? How are you going to verify this?

Commented [BN33]: Really? You're going to build in infrastructure to handle internationalization? For a proof of concept I don't want to pay for this!@

5 Glossary

Definitions, Acronyms & Abbreviations

<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>One-way</u>	A one-way flight means a trip with a departure airport and an arrival airport.
<u>Round-trip</u>	Round-trip flights mean two trips: 1. To trip: departure airport to arrival airport. 2. Return trip: destination airport to arrival airport
<u>Stopover</u>	The airport connecting two flights.
<u>GMT</u>	Greenwich Mean Time
<u>First class</u>	Seating will be better and price higher.
<u>Coach seat</u>	Ordinary seating conditions with ordinary price.
<u>Concurrency</u>	Multiple customers will operate on the system at the same time but will not conflict with each other.
<u>Leg of flight</u>	From a certain departure airport to an arrival airport, there may not be a direct flight to choose. Several flights connecting from this specific departure airport to the arrival airport can also satisfies travel preferences. Each of these connecting flights are called leg of flight.
<u>Travel time</u>	The total time (including all legs of flights and layover time) spent on a

Commented [BN34]: Server, Database, time window, customer, timezone (GMT, Local), etc. Go back through the document with a critical eye and pick up the terms used.

Commented [BN35]: Should you include layover time in glossary?

	trip from departure airport to arrival airport.
<u>Contractor</u>	A contractor can be a customer or a realter representing a bunch of customers that wants to reserve seats from the system.
<u>Time window</u>	A time span for customers to choose, including a start time and an end time (they should be in the same day), the flight time of the first leg of flights will be limited to after this start time and before this end time.

Commented [BN36]: Why include this? I don't see any reference to contractor in the document.