Requirements Analysis Document (RAD)

Prepared for

World Plane, Inc. (WPI)

Prepared by

Team #1

Rodrigo Arguello-Serrano, Jacques Philippe, and Sylar Zhang

February 13th, 2024

1.0.0

Requirements Analysis Document (RAD)

## **Introduction**

## **Purpose of the system**

World Plane, Inc. (hereinafter referred to as WPI) aims to assess the viability of migrating our Travel Agency airline travel reservation system to a Retail Customer airline reservation system. To achieve this objective, WPI seeks the development of a small-scale 'proof of concept' software application.

## **References**

* Statement of Work (SOW) provided by World Plane, Inc.
* Kickoff Meeting, held on January 10th 2024 @ 6:00 pm EST

## **Scope of the system**

WPI currently operates a portion of its airline flight database on an Internet-accessible server, with data retrievable via a standard HTTP GET API returning plain text XML strings. Comprehensive API documentation will be furnished to all project development teams, including preconditions, postconditions, invariants, and side effects. Additionally, sample Java code illustrating API calls and methods for parsing XML results will be supplied, along with result samples to facilitate comprehension of the XML document structure.

## **Core System Functionalities**

The client software allows retail customers to select departure and arrival airports for one-way or round-trip flights, choosing between first class or coach seating. It calculates layover times and displays flight times in local and GMT formats. Concurrent development by multiple contractor teams is supported through database locking. However, the system remains simple as a proof of concept, lacking features such as personal information collection, payment processing, or reservation deletions.

## **Objectives and Success Criteria of the Project**

The application's success depends upon meeting the following core set of objectives and requirements described in Section 4 of this document.

# **Current System**

## **Existing System**

Retail airline reservation systems such as Expedia, Booking.com, and Kayak exist at present. These are web-based airline reservation systems that allow users to book trips and search for various flights on one single web. In these systems, users select a departure airport and an arrival airport, enter the leaving date and returning date (if the user is booking a round trip), and make customizations to the trip (e.g., the number of stopovers and airlines). Currently, given by the Statement of Work, WPI only has a subset of airline flight database running on an Internet accessible server. To augment this travel agency airline travel reservation system to a retail customer airline reservation system, we plan to incorporate front-end, back-end, and database developments to create a user-friendly and technologically advanced platform. This will involve adding functionalities similar to those found on the existing retail airline reservation systems. However, these established retail systems will not be entirely replaced due to their maturity and the extensive systems already deployed and continuously upgraded to meet users’ needs.. Compared with these systems, our proposed system is still at an early stage.

## **Current Operations**

The Statement of Work did not include details of the client’s current business practices regarding their business practices. Without access to specific details of business practices or direct information from the stakeholders, it is difficult to accurately analyze how the client currently addresses the business problem.

# **Proposed System**

## **Overview**

The proposed system will afford the potential traveler a greater degree of flexibility. Specifically, the proposed system will be accessible from anywhere; all that will be required is the use of an Internet-enabled device on which a compatible web browser is installed. And since the proposed system will be available online, it will, for the most part, be offered as an around-the-clock service, making its use far more convenient than those that are agent-based. Keep in mind that agent-based services can be off-putting to potential travelers due to the physical location from which they operate and their hours of operation. Importantly, the proposed system will offer a sufficient degree of transparency and accuracy, as every piece of information that is pertinent to a user-associated transaction will not only be made available to the potential traveler but also be as up-to-date as possible, facilitating and bolstering the overall decision-making process. Speaking of which, the proposed system will be as user-friendly as possible and will even strive to conform to the latest Web Content Accessibility Guidelines (WCAG) and Americans with Disabilities Act (ADA) standards in its design, making navigating the system interface no less than an intuitive experience. Not to mention, the proposed system will grant the potential traveler the ability to willingly cancel an in-process booking transaction at any time. Finally, the proposed system will disallow the collection and storage of any traveler-associated personally identifiable information (PII) as well as the processing of monetary transactions, mitigating the potential risk of the potential traveler being personally exposed.

## **Conceptual Model – User Scenarios**

### **User**

A user is any person who uses the proposed system (henceforth referred to as AirlineTravelReservation or ATR for short) to book a reservation for a flight. Moreover, a user is assumed to have access to an Internet-enabled device on which a compatible web browser is installed. Also, a user requires the ability to travel to and from any ATR-listed airport at any date and time of their choosing. A user requires a user-friendly experience and complete transparency when interacting with the ATR interface – that is, any information resulting from the user interacting with the ATR interface must be pertinent, complete, and accurate. Finally, a user requires the ability to modify and cancel their reservation at any time while interacting with the ATR interface before committing to any in-process transaction.

### **Administrator**

An administrator is any person designated to manage the ATR database. Importantly, an administrator requires an assigned ATR database role that matches their designated duties. As such, an administrator, based on their assigned role, requires the ability to add, delete, view, and update all ATR database-associated tables and/or records. Finally, an administrator requires the assurance that their interaction with the ATR database will not adversely impact any in-process and/or committed user-associated transaction.

## **Functional Model - Use Case Model**

The varying functionalities that the proposed system will provide and the way they relate to one another are described using the below use-cases.

### **Flight Search**

|  |  |
| --- | --- |
| **Name** | Flight Search |
| **Description** | This use case describes the scenario during which the user searches for available flights based upon a desired set of criteria. |
| **Participating Actor** | User |
| **Flow of Events** | 1. The user selects the type of flight (i.e., one-way, round-trip, and/or multi-city) and seat (i.e., economy, premium economy, business, and/or first class) desired. 2. The user inputs their desired departure and arrival airports. 3. The user selects their desired departure date. 4. The user quantifies the number of travelers by type (i.e., adult, children, and/or infant).    1. If a traveler is a child between 2 and 17 years of age, the ATR interface requires the user to quantify the age of the traveler.    2. If a traveler is an infant younger than 2 years of age, the ATR interface requires the user to quantify the age of the traveler and whether the traveler will travel on the lap of an older traveler or in a properly secured safety seat. 5. The user executes the search for available flights. 6. The results of the search are displayed on the ATR interface. |
| **Entry Conditions** | * The user navigates to the ATR website. * The user has a desired departure and arrival airport. * The user has a desired departure date and time as well as arrival date and time. * The user has a desired type of flight and seat. |
| **Exit Conditions** | * The ATR application displays information that is pertinent to the search that the user executed. |
| **Quality Requirements** | * The ATR website must be available to the user over the Internet no less than 90 percent of the time (and 99.9 percent of the time if hosted across multiple AWS server instances across two or more availability zones). * The search results must be displayed in 15 seconds or less. * The search results must list the following:   + The cost of the flight per traveler.   + The departure and arrival airports.   + The departure date and time.   + The arrival date and time.   + The flight operator and flight number.   + The duration of the flight.   + Whether the flight is nonstop and if not, the number of stops that will occur between the listed departure and arrival airports (with each stop listing their duration and the airport where they occur). * All displayed time values must match the time zone of the user. |

### **Flight Review**

|  |  |
| --- | --- |
| **Name** | Flight Review |
| **Description** | This use case describes the scenario during which the user reviews the information that resulted from their search for available flights using their selected set of desired criteria. |
| **Participating Actor** | User |
| **Flow of Events** | 1. The user sorts the results of their search by a desired criterion (i.e., price, departure time, arrival time, flight duration, number of stops, and flight operator) in ascending or descending order. 2. The user scrolls through the list of available flights. 3. The user selects a potential flight for further review. 4. A modal window that features additional details for the selected flight is displayed. 5. The user makes one of the following choices:    1. The user exits the modal window and repeats steps 2 through 5 for another potential flight.    2. The user elects to book the selected flight. |
| **Entry Conditions** | * The user is on the ATR website. * The results from the search for available flights are displayed on the ATR interface. |
| **Exit Conditions** | * The user elects to book a selected flight. |
| **Quality Requirements** | * The ATR website must be available to the user over the Internet no less than 90 percent of the time (and 99.9 percent of the time if hosted across multiple AWS server instances across two or more availability zones). * The customer must have access to information that is complete, accurate, and up-to-date. * The modal window with the additional details for the selected flight must be displayed in 15 seconds or less. * The modal window must list the following:   + The cost of the flight per traveler.   + The number of travelers by type.   + The number of seats available by type.   + The departure and arrival airports along with their geographic coordinates and local weather (if time permits).   + The departure date and time.   + The arrival date and time.   + The flight operator and flight number.   + The duration of the flight.   + Whether the flight is nonstop and if not, the number of stops that will occur between the listed departure and arrival airports (with each stop listing their duration and the airport where they occur).   + Any special conditions pertaining to the booking transaction (e.g., refundability, cancellations allowance and penalties, discounts extended by the flight operator, etc.). * All displayed time values must match the time zone of the user. |

### **Flight Booking**

|  |  |
| --- | --- |
| **Name** | Flight Booking |
| **Description** | This use case describes the scenario during which the user modifies, cancels and/or finalizes a booking transaction after reviewing the additional details for a selected flight. |
| **Participating Actor** | User |
| **Flow of Events** | 1. The user reviews the additional details for the selected flight. 2. The user makes one or a combination of the following choices:    1. The user reviews the additional details for the selected flight.    2. The user exits the modal window and repeats steps 2 through 5 from the Review Flights use case for another potential flight.    3. The user reviews and alters the modifiable details for the selected flight.    4. The user commits to and finalizes the booking transaction for the selected flight.       1. The ATR website requests that the user performs a final review of and confirmation of the additional details before finalizing the booking transaction for the selected flight.    5. The user abandons the transaction and navigates away from the ATR website. 3. The user |
| **Entry Conditions** | * The user is on the ATR website. * The user scrolled through the list resulting from their search of available flights and selected a potential flight. * A modal window that features additional details for the selected flight is displayed. |
| **Exit Conditions** | * The user receives a confirmation message on the modal window that their transaction has been processed. * The user exits the modal window and repeats steps 2 through 5 from the Review Flights use case for another potential flight. * The user abandons the transaction and navigates away from the ATR website. |
| **Quality Requirements** | * The ATR website must be available to the user over the Internet no less than 90 percent of the time (and 99.9 percent of the time if hosted across multiple AWS server instances across two or more availability zones). * The customer must have access to information that is complete, accurate, and up-to-date. * In the event of an extended loss of Internet connectivity, any user-associated transaction that is in the process of being finalized must be completed. |

### **Add/Delete/Modify Record**

|  |  |
| --- | --- |
| **Name** | Add/Delete/Modify Record |
| **Description** | This use case describes the scenario during which the administrator adds, deletes, and modifies the ATR database flight records. |
| **Participating Actor** | Administrator |
| **Flow of Events** | 1. The administrator accesses the ATR database management system. 2. The ATR database management system authenticates the administrator and grants them access. 3. The administrator queries the database for a record of interest and reviews said record for content. 4. The administrator makes one or a combination of the following choices:    1. The administrator executes the appropriate command to add a record.    2. The administrator executes the appropriate command to modify a record of interest.    3. The administrator executes the appropriate command to delete a record of interest.    4. The administrator repeats steps a, b, and c as required for additional records.    5. The administrator queries the ATR database for the record and confirms the successful execution of steps a, b, and c as required. 5. The administrator logs off the ATR database management system. |
| **Entry Conditions** | * The administrator is assigned a role based upon their designated duties. * The administrator is assigned a list of records to add to, modify, and/or delete from the ATR database. * The administrator accesses the ATR database management system during their designated work hours. |
| **Exit Conditions** | * The ATR database is update as required. |
| **Quality Requirements** | * The ATR database management system must always be available to the administrator. * In the event an administrator adds, modifies, and/or deletes an ATR database record, the result of the transaction should be nearly instantaneously viewable by the user. |

## **Analysis Model – Object Model**

The following represents the initial static object model and illustrates the varying classes and the manner in which these classes are associated as well as support the function use cases.

|  |  |
| --- | --- |
| **Participating Actor** | **Use Cases** |
| User | 1. Flight Search 2. Flight Review 3. Flight Booking |
| Administrator | 1. Add/Delete/Modify Record |

A diagram of a computer

Description automatically generated

# **Requirements**

## **Functional Requirements**

|  |
| --- |
| **Requirement** |
| The User must be able to reserve travel from the destination airport to an arrival airport using a series of connecting. |
| The AirlineTravelReservation must allow User to reserve travel from the destination airport to an arrival airport with stepovers of sufficient time to allow Users to transiton between gates. |
| The User must be able to reserve flights to travel either one-way (from departure to destination), or reserve a round-trip flight |
| The User must be able to reserve either first class seating or coach seating for travel. |
| The User must be able to search for flights using departure date, arrival dates and respective time |
| The User must be able to select and save flights. |

## **Nonfunctional Requirements**

|  |
| --- |
| **Requirement** |
| The AirlineTravelReservation must be accessed by standard HTTP GET API |
| The AirlineTravelReservation must provide documentation for the APIs, including preconditions, post conditions, invariants and side effects |
| The AirlineTravelReservation must provide a Java subset of functionality similar to existing web-based airline reservation systems (Proof-Of-Concept) |
| The AirlineTravelReservation must allow User to reserve travel from the destination airport to an arrival airport using a series of connecting flights with a maximum of two stopovers |
| The AirlineTravelReservation must allow User to reserve travel from the destination airport to an arrival airport with stepovers of sufficient time to allow the airline to transfer the passenger’s bags from one flight to the next. |
| The AirlineTravelReservation must allow User to reserve travel from the destination airport to an arrival airport using a series of connecting flights with a maximum of two stopovers |
| The AirlineTravelReservation must restrict situations where requested seating is not available for all legs of the flight. |
| The AirlineTravelReservation must not offer flight choices with unreasonable layover times. |
| The AirlineTravelReservation must allow multiple Users running against the same database server |
| The AirlineTravelReservation must support concurrency by locking the server database and prohibiting changes during the User reservation session and will be released within a reasonable period. |
| The AirlineTravelReservation must not support any payment processing. |
| The AirlineTravelReservation must should remain simple |
| The AirlineTravelReservation must have the ability to reserve seats (first class or coach) for each leg of flights |
| The AirlineTravelReservation must not store personal User information into the server |
| The AirlineTravelReservation must not allow a reservation to be deleted once it is made. |
| The AirlineTravelReservation must display flights sorted by price, departure time, arrival time or travel time. |
| The AirlineTravelReservation must display local time on selected leg airports. |
| The AirlineTravelReservation must confirm flights selection before saving the User reservation. |
| The AirlineTravelReservation must be responsive on User actions. |
| The AirlineTravelReservation must support testing and developing by having a finite number of flights and airports |

# **Glossary**

Definitions, Acronyms & 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. |
| *Software Application* | Refers as AirlineTravelReservation (ATR) |
| *User* | Refers as person using the Application (Customer) |