Software Requirements Specification

for

Airline Flight Information and Reservation System (AFIRS)

Table of Contents

Revision History

Name	**Date**	**Reason For Changes**	**Version**
W. Wang, Y. Wang, C. Grant	09/07/08	First Draft	V0.1
W. Wang, Y. Wang, C. Grant, B. Lakshmanan	09/08/08	First Revision	V0.2
W. Wang, Y. Wang, C. Grant, B. Lakshmanan	09/08/08	Final Revision	V1.0

1. Introduction

1.1 Purpose

This software requirement specification (SRS) document describes the functional and nonfunctional requirements of the airline flight information and reservation system (AFIRS) release version 1.0. Business opportunity and objectives is briefly summarized followed by detailed description of the system's scope, vision, use case, features and other related requirement issues. This document satisfies phase 1 of COP 5725 and intended to be used as a reference for group members. In the project's later phases, such as system design, database design, implementation and testing, this document should be referred as functional model of the system for release 1.0.

1.2 Document Conventions

All system development activities should follow the final version of this document. Any discrepancy that found during in later phases should be modified subject to SRS. However, this document may be subject to change dependent on the decision of the group members.

1.3 Business Requirement

1.3.1 Business Opportunity

As Internet and Web-based applications become more and more crucial and widely adapted in today's society, people have found that most everyday businesses can be carried out online in a more flexible and convenient way. Specifically, booking air flights through the Internet gives a lot of advantages when compared to traditional counter or telephone based methods. Matching flights, for instance, can be displayed to the customer in a well-structured manner by simply clicking a button. Customers may also expediently specify their demands through well-designed user interface provided by the website.

1.3.2 Business Objective

The objectives of this project are the following:
BO-1: Reduce the misunderstandings which could happen when customers try to book an air flight tickets through traditional telephone-based processes.
BO-2: Increase the customer satisfactory during air flights booking.
BO-3: Reduce the overall average time each customer spends on booking air flights.

1.4 Project Vision

1.4.1 Vision Statements

Users who wish to travel exclusively within the fifty states may reserve their air flights operated by any airline among all available airports through AFIRS system. The AFIRS system is a Web-based application that can accept client requests, list searched results, process booking, payment, modification and cancellation to existing reservations. Users do not have to personally go to the counter or contact airline representatives through the telephone, but only access AFIRS through any browser to book their flights. By using AFIRS, user can not only save time but also get much larger search space from which they have higher chance to find a suitable air flight.

1.4.2 **Main Features**

FE-1: Update/add/delete airport information (by administrator)
FE-2: Update/add/delete airline company and air flight information (by administrator)
FE-3: Update/add/delete user reservation information (by administrator)
FE-4: Flight inquiry through basic search: by date, time and airport (or city) of departure and return flights, respectively, and number of passengers
FE-5: Flight inquiry through advanced search: by one-way/round-trip/multi-city, preference for nonstop, arrival date/time, airline, flight number, and flight class
FE-6: Flight booking and payment process
FE-7: Flight reservation modification and cancellation
*FE-8[1]: Frequent user login
*FE-9: Frequent user information management
*FE-10: Dashboard: possible interesting flights for login users, last minute deals, etc.

1.5 Project Scope

This project designs and implements AFIRS to fulfill all the vision statements. Supported by a well designed database, all available air flight information is integrated together and can be accessed easily through a single point. A friendly user interface is provided so that various combinations of search criteria can be fetched from user and generates corresponding database search statements. AFIRS provided both customer and administration interfaces with the latter used for administration purposes. If time permits, AFIRS will support frequent user registration and personal information management.

1.6 References

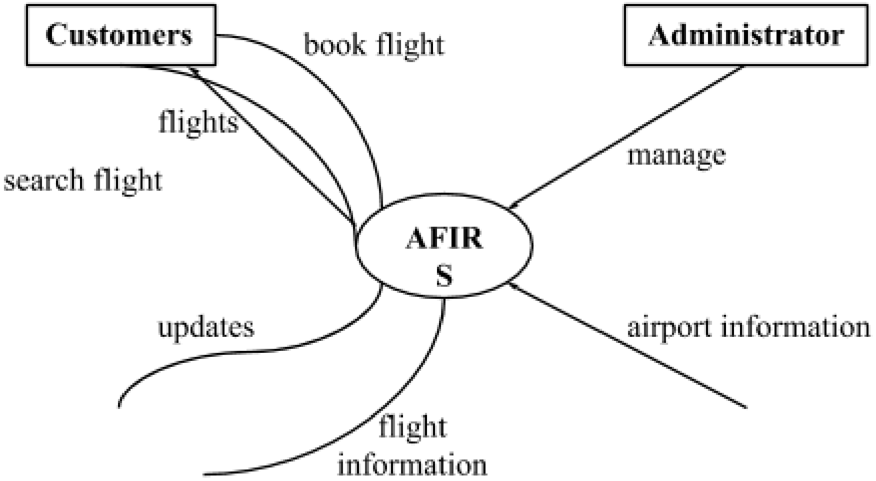
1. AFIRS project specification
http://www.cise.ufl.edu/~mschneid/Teaching/COP5725_Fall2008/project_spec.pdf

2. Overall Description

2.1 Product Perspective

[1] Features with an asterisk (*) means this feather will be implemented if time permits.

AFIRS is an Internet-based application executing on a Web server and connected to enterprise databases. As shown in Figure 1, AFIRS accepts and processes requests from two patrons: end users (customers) and system administrators. Besides the local server database (for storing reservation records), AFIRS also integrates databases from airlines and airports. The system is expected to have a Web user interface for customers and an authorization based Web interface for administrators. Its final release has merits of being efficient and precise in flight searching, consistent in booking transaction processing and secure in credential data transmission.



Airline Database
Airport Database

Figure 1: Context diagram for AFIRS

2.2 User Classes and Characteristics

Customer (favored): A customer is any person who uses AFIRS to book air flight ticket(s). Customers have Internet access to AFIRS Internet-based user interface to book their flights. They may want to travel among all airports in the United States, prefer any date and time, favor any airline as well as various demands such as arrival time, flight class or non-stop. They require an easy-to-use and trustable online air flight reservation system. They need to be fully informed about all available flights before booking, secure assurance when making reservation and flexibility to view, modify or cancel their reservations after having made them.

Administrator: An administrator may be a dedicated staff whose responsibility is managing AFIRS's back-end databases. They may wish to update/add/delete any information in existing reservation database or user registration database. They may need to create and manage temporary views of fetched records from the databases of airlines and airports. They also need to be convinced that their operations will not affect the correctness of user transactions.

2.3 Operating Environment

OE-1: AFIRS should be accessed using any popular versions of the following Web browsers: Microsoft Internet Explorer, Mozilla Firefox, Netscape, Opera and Google Chrome.

OE-2: AFIRS should be able to run on Apache Tomcat Web server configured in a stable Linux/Unix/Windows machine.

OE-3: AFIRS works with Oracle 10i database management system.

2.4 Design and Implementation Constraints

CO-1: The system database design will be based ER modeling which will in turn transfer to database schema formulated using SQL DDL statements.

CO-2: All HTML code should conform to the HTML 4.0 standard.

2.5 User Documentation

UD-1: Final release will be accompanied with a user guide to inform new users how to use AFIRS.

UD-2: The system will be designed as user friendly as possible.

2.6 Assumptions, Dependencies and Limitations

AS-1: Users of AFIRS are assumed to have a computer with internet access.

AS-2: Credit and debit card authorization services are assumed to be available.

DE-1: AFIRS has collaboration with every airline's databases thus it must be able to communicate with these systems. Also it has to obey airline's business rules.

LI-1: Only domestic flights are supported.

3. System Features

3.1 Travel Search Feature

3.1.1 Description and Priority

The search feature allows the user to search for an airline flight on a variety of dimensions.

3.1.3 Functional Requirements

Req #	**Description**	**Priority**
REQ-S1	The user will be able to search for flights through a standardized screen. Advanced options will be available by clicking appropriate links.	[Priority = High]
REQ-S2	Through the standard flight search method the user will be able to search one-way, round-trip, or multi-destination flights. The screen by default will display options for searching round trip flights. The search criteria can be modified by the user by selecting one-way and multi-destination options which would be displayed on a new window.	[Priority = Medium]
REQ-S3	Through the standard flight search method the user shall be able to specify the departure and return date of their flight.	[Priority = High]
REQ-S4	Through an advanced flight search method the user shall be able to specify the arrival flight times.	[Priority = Medium]
REQ-S5	The standard flight search method will enable the user to search both precise dates as well as a range of arrival and departure dates.	[Priority = Medium]
REQ-S6	The standard flight search method will allow the user to specify a preferred airline. This is optional, i.e. the user may or may not specify the airline of preference.	[Priority = Low]
REQ-S7	The user will have the option to express a preference of non-stop flights; otherwise both non-stop flights and flights with stopovers will be treated equally.	[Priority = Low]
REQ-S8	User can only search for a flight in the future and within one year of the current date. (The cutoff time can be applied, i.e. user cannot make the reservation for the flight which will take off in 2 hours.)	[Priority = High]

REQ-S9	The user to select any hour on the hour (i.e. flights at 12am, 11am, 2pm etc.).	[Priority = High]
REQ-S10	When the user is searching for a flight he may select one of the following preset time ranges: Morning, Afternoon, Evening, Night, or Anytime.	[Priority = Medium]
REQ-S11	User should be able to search by Flight number and airline. This will allow him to see the particular flight for the particular airline.	[Priority = Low]
REQ-S12	The user may specify anywhere from 1 (one) to 6 (six) passengers. Passengers will not be treated separately as minors or seniors.	[Priority = High]
REQ-S13	Any error in entry of the system will stop the system from processing the search. An error message will be presented to the user.	[Priority = High]

3.2 Result Browsing Feature

3.2.1 Description and Priority

After a user performs a search, their search results are displayed. The object of this feature is to display the flights which closely match the desires the user expressed during the search. In turn the user should select a flight in order to proceed to book the flight.

3.2.3 Functional Requirements

Req #	**Description**	**Priority**
REQ-BR1	The user will view the results of the search performed in a graphical-tabular-like format.	[Priority = High]
REQ-BR2	The user will be able to sort the results by price, duration, departure time, and arrival time.	[Priority = High]
REQ-BR3	When the user selects a particular flight the system will send them to the booking option.	[Priority = High]

3.3 Flight Booking Feature

3.2.1 Description and Priority

This Feature allows users to book a reservation for a trip.

3.2.3 Functional Requirements

Req #	Description	Priority
REQ-B1	User will be able to specify the Flight Class which will be with respect to the flight chosen.	[Priority = Low]
REQ-B2	The user will be able to choose from a listing of available seats.	[Priority = Low]
REQ-B3	The system shall accept reservations for 1 person per flight and seat number.	[Priority = High]
REQ-B4	The user must complete all the necessary steps to book the flight for the seat reservation to be guaranteed.	[Priority = High]
REQ-B5	The user must specify a valid e-mail address when booking a flight.	[Priority = High]
REQ-B6	Users may only book flights within the United States.	[Priority = High]
REQ-B7	All times displayed are expressed in local time, specific to the location.	[Priority = High]
REQ-B8	The user will only be able to book flight within one year of the present date. (Cutoff time may be applied)	[Priority = High]
REQ-B9	After booking, the user may cancel/modify their flight.	[Priority = Low]

4. Use Cases

4.1 Description

This section list use cases for AFIRS. The various user classes identified the following use cases and primary actors for the AFIRS:

Primary Actor	Use Cases
Customer	1. Place reservation 2. Change reservation 3. Cancel reservation 4. Search flight information (included in 1)
Administrator	5. Update/Add/Delete Flight information 6. Update/Add/Delete User Reservations

4.2 Reservation Use case

Use Case ID:	1		
Use Case Name:	Place reservation		
Created By:	Yibin Wang	Last Updated By:	Weixun Wang
Date Created:	09/07/2008	Date Last Updated:	09/10/2008
Actors:	Customer		
Description:	A customer accesses the AFIRS from the Internet, optionally search for specific ticket/flight information of interest, selects ticket(s), and places reservation.		
Preconditions:	None		
Post conditions:	1. Database of available tickets is updated to reflect items in this order. 2. Remaining tickets number is updated.		
Normal Flow:	**1.0 Order a Single Ticket** 1. Customer uses the web interface to enter a certain query to view flight information for		

	a specified interest. 2. System displays available flight information satisfied the query. 3. Customer selects one or more items from page. Customer can also click on a particular ticket to see the detailed information. 4. System displays reservation with detailed price information including all taxes. 5. Customer confirms reservation or requests to modify reservation (back to step 3). 6. Customer specifies payment method. 7. Customer indicates that reservation is complete. 8. System confirms acceptance of the order. 9. System sends Customer an e-mail confirming order details, price, and		

	additional links to access the ticket details or for potential modification. 10. System stores order in database, and updates available ticket information (database).		
Alternative Flows:	**1.1 Order multiple tickets** (branch after step 8) 1. Customer asks to place another reservation. 2. Return to step 2. **1.2. Order the Last minute deals** (after step 2) 1. Customer orders the daily special from the menu. 2. Return to step 5.		
Exceptions:	**1.0.E.1 Concurrent access from multiple users (when there is less available ticket than potential users, demand surpass supply)** (at step 1) 1. System informs Customer that ticket no longer available. 2a. Customer cancels the ticket order. 2b. System		

	terminates use case. 3a. Customer requests to select another ticket. 3b. System restarts use case. **1.0.E.2 Cutoff time for available ticket (the cutoff time is usually 5 hours before the departure time of the flight)** (at step 1) 1. System informs Customer that the cutoff time policy occurs. 1a. System denies the access to the particular ticket information terminates use case. **1.2.E.1 the user input query is not reasonable (e.g. departure time is behind arrival time)** (at step 1) 1. System informs Customer of right form of query to input. 2. Customer changes query.		
Includes:	None		
Priority:	High		

Frequency of Use:	Approximately 400 users, average of one usage per day		
Special Requirements:	1. Customer shall be able to cancel the order at any time prior to confirming the order. 2. Customer shall be able to view all tickets he reserved within the previous six months. (Priority = medium)		
Notes and Issues:	1. The default time zone of departure/arrival information is the local time zone of specific city. 2. If customer doesn't need to have an account until reservation is placed.		

4.3 Change or Cancellation Use Case

Use Case ID:	2,3		
Use Case Name:	Customer change or cancel reservation (before cutoff time)		
Created By:	Yibin Wang	Last Updated By:	Weixun Wang
Date Created:	09/07/2008	Date Last Updated:	09/11/2008

Actors:	Customer		
Description:	Customers who have reservations in AFIRS should be able to modify or cancel these reservations before a certain cutoff time.		
Preconditions:	Customer is logged into AFIRS.		
Postconditions:	Customer has placed certain actions on existing reservations.		
Normal Flow:	**2.0 Reservation modification or cancellation** 1. Customer requests to change or cancel reservation. 2. System invokes Authenticate User's Identity use case. 3. System verifies Customer's identity and provides the login view menu for customer. 4. Customer clicks on the reservation section and chooses one of the reservations to modify or		

	cancel. 5. Customer confirms desire to do modification or cancellation. 6. System checks the cutoff time and permit the modification/cancellation requested by customer. 7. System asks Customer to confirm his or her decision. 8. System sends corresponding update information to the database of ticket/flight information. 9. System informs Customer the change and provides confirmation number of the transaction.		
Alternative Flows:	None		
Exceptions:	**5.0.E.1 Customer identity authentication fails** (at step 2) 1. System gives user two more opportunities for correct identity authentication. 2a. If authentication is successful,		

	Customer proceeds with use case. 2b. If authentication fails after three tries, System notifies Customer, logs invalid authentication attempt, and terminates use case. **5.0.E.2 The cutoff time policy is applied** (at step 6) 1. System informs Customer that he cannot make the modification/cancellation and explains why. 2. System terminates use case.		
Includes:	Authenticate User’s Identity		
Priority:	High		
Frequency of Use:	Once per user on average		
Special Requirements:	1. User authentication is performed per corporate standards for medium-security applications.		

Assumptions:	None		
Notes and Issues:	1. Expect low frequency of executing this use case. But relatively high frequency during the hot season (Christmas)		

4.4 Administrative Use Case

Use Case ID:	5,6		
Use Case Name:	Update/Add/Delete Flight information/User reservations		
Created By:	Yibin Wang	Last Updated By:	Weixun Wang
Date Created:	09/07/2008	Date Last Updated:	09/11/2008
Actors:	Administrator		
Description:	The Administrator may modify the flight information and prices for a specified date to reflect changes in availability or prices or to define last minute deal. Administrator can also Update/Add/Delete User Reservations in some cases.		

Preconditions:	Database already exists in the system.		
Post conditions:	Modified database has been saved.		
Normal Flow:	**11.0 Update/Add/Delete Flight information/User reservations** 1. Menu Manager requests to view the menu for specific ticket/flight information. 2. System displays the menu. 3. Menu Manager modifies the menu to add new information, remove or change items, create or change deal, or change prices, number of seats available ect. (invoke the database management language module through interface) 4. Menu Manager requests to save the modified menu. 5. System saves modified menu. 6. If the change is about user reservations,		

	send notification to users by e-mail		
Alternative Flows:	None		
Exceptions:	**11.0.E.1 No item exists for specified information** (at step 1) 1. System informs Administrator that no menu exists for the specified date. 2. System asks Administrator if he would like to add a new item. 3a. Administrator says yes. 3b. System invokes Database interface. 4a. Menu Manager says no. 4b. System terminates use case. **11.0.E.2 Item specified is the past information** (at step 1) 1. System informs Administrator that the item requested cannot be modified. 2. System terminates use case.		

Includes:	None		
Priority:	High		
Frequency of Use:	Approximately 20 times per week by one user		
Special Requirements:	1. The Administrator may cancel out of the modification function at any time. If any item has been changed, the system shall request confirmation of the cancellation.		
Assumptions:	None		
Notes and Issues:	1. If the Administrator is doing modification of certain information, that information should be temporally invisible/inaccessible for customers.		

5. External Interface Requirements

5.1 User Interfaces

(1) The system shall provide a help (explanation) link from each displayed HTML page to explain how to use that page.
(2) The Web page of AFIRS shall permit complete navigation, flight selection, ticket reservation and ticket/flight information query view using keyboard and mouse combinations.

5.2 Hardware Interfaces

No hardware interface specified for AFIRS.

5.3 Software Interfaces

5.3.1 Flight/ticket inventory/database system

(1) The AFIRS should transmit and store the detailed properties (quantities, flight number, departure/return time/date, seat, location) of ticket items for the potential requests from users.
(2) The Flight/ticket inventory/database system should keep updated information about the availability of ticket and flight information to determine whether a requested reservation is available or not.
(3) When the AFIRS make the query about ticket information with constrains, Flight/ticket inventory/database system should provide all the information that satisfy the query.
(4) The Flight/ticket inventory/database system should accept information update operation flows coming from the administrator interface system.

5.3.2 User query/view system

The user query/view system should communicate with Flight/ticket inventory/database system through a programming interface for the following operations.
(1) To allow a user post various query about ticket information available at present. User can use any combination available
(2) To show the user the resulting list of query. This should allow users to further sort the list by subfields of the ticket information (e.g. price). The view should be friendly and flexible.
(3) To allow a user place a reservation
(4) To detect the conflict if a user placed two incompatible reservations.
(5) To allow a user view his/her reservation
(6) To allow user modify his/her reservation
(7) To allow user to manage his/her account in AFIRS

5.3.3 Administrator interface system

The administrator interface system should communicate with Flight/ticket inventory/database system through a programming interface for the following operations.
(1) To allow administrator Update/Add/Delete Flight information
(2) To allow administrator Update/Add/Delete User Reservations

5.4 Communications Interfaces

CI-1: AFIRS should send an e-mail message to the user to confirm the acceptance of acceptance of reservation.

CI-2: because the AFIRS is a Web-database system, all kinds of user-system interaction is presented by user-side web browser. So the communication standard is HTTP protocol

6. Other Nonfunctional Requirements

6.1 Performance Requirements

PE-1: The system shall accommodate 400 users during the peak usage time window of 8:00am to 10:00am local time, with estimated average session duration of 45 minutes.

PE-2: All Web pages generated by the system shall be fully downloadable in no more than 10 seconds over a 40KBps modem connection.

PE-3: Responses to queries shall take no longer than 10 seconds to load onto the screen after the user submits the query.

PE-4: The system shall display confirmation messages to users within 4 seconds after the user submits information to the system.

6.2 Safety Requirements

No safety requirements have been identified.

6.3 Security Requirements

SE-1: Users shall be required to log in to the AFIRS for their own reservation information and modification with e-mail address and password.

SE-2: The system shall permit only authorized members who are on the list of authorized Menu Managers to do administrator’s task.

SE-3: The system shall permit customers to view only their own previously placed orders, not orders placed by other customers.

6.4 Software Quality Attributes

Availability-1: The AFIRS shall be available to users on the Internet 99.9% of the time between 5:00am and midnight local time and 95% of the time between midnight and 5:00am local time. And AFIRS should be available to administrators all the time.

Robustness-1: If the connection between the user and the system is broken prior to an order being either confirmed or canceled, the AFIRS shall enable the user to recover an incomplete reservation.

Consistancy-1: The ticket availability in AFIRS shall be updated when a transaction (more, administrator’s update operation) is finalized to ensure that the most updated information is delivery to any following queries from other customers. When administrator is updating the information, the consistency should also hold among user view.

7. Other Requirements

7.1 System Development Requirements

7.1.1 Description

This section describes what resources will be utilized in the development and use of the software.

7.1.2 Requirements

Req #	**Description**
REQ-SR1	The database backend system in use will be Oracle 10i.
REQ-SR2	The Front-end and middle logic will be written using JavaEE.
REQ-SR3	Code will be stored on the Google Code SVN repository. [http://code.google.com/p/afirs/]
REQ-SR4	Our development environment will be the latest NetBeans Integrated Development Environment.
REQ-SR5	We will use Tomcat as our web server.
REQ-SR6	We may write scripts to create synthetic code in Python/Perl.
REQ-SR7	We may make changes to any of the above system requirement at anytime and for any reason.

7.2 Data Categories Requirements

7.2.1 Description

This section describes the category of data required by the system. Because there is no actual complete data set available for use we will produce the needed data synthetically. This data will be more formally represented in our entity relational design data model.

7.2.2 Requirements List

- A List of Airports including:
    - Airport Name
    - Abbreviation
    - Location

        - City
        - State
        - zip code (If possible)
        - Time zone
- The information about several Flights, more specifically:
    - Flight id
    - Airline
    - Flight number
    - Departure/Arrival Date/Time
    - Departure/Destination Airport
    - Seats
    - Total / Vacant
    - Seat Number
    - First Seats
    - Coach Seats
    - Type plane
    - Fare
- Information of Reservations made:
    - Flights id
    - Email
    - Passenger names
    - Credit card type/number
    - Address
    - Total Price
- User Information, most importantly containing:
    - Email
    - Password
    - Reservations