
Software Requirements Specification

for

Airline Reservation System

Version 1.0

Prepared by

Group Name: TDS

**Fecioru Alina - Maria
Pîrvu Andrei - Cătălin
Popa Radu - Mircea**

**CEN 3.1
CEN 3.2
CEN 3.2**

Instructor:

Sbora Cătălin

Course:

Software Engineering

Date:

20.03.2020

TABLE OF CONTENTS	II
1 INTRODUCTION	1
1.1 DOCUMENT PURPOSE	1
1.2 PRODUCT SCOPE	1
1.3 DEFINITIONS, ACRONYMS AND ABBREVIATIONS	1
1.4 REFERENCES	3
1.5 OVERVIEW	3
2 OVERALL DESCRIPTION	4
2.1 PRODUCT PERSPECTIVE	4
2.2 PRODUCT FUNCTIONS	4
2.3 USER CHARACTERISTICS	4
2.4 OPERATING ENVIRONMENT	5
2.5 CONSTRAINTS	5
2.6 ASSUMPTIONS AND DEPENDENCIES	5
2.7 APPORTIONING OF REQUIREMENTS	6
3 SPECIFIC REQUIREMENTS	7
3.1 EXTERNAL INTERFACE REQUIREMENTS	7
3.1.1 USER INTERFACES	7
3.1.2 HARDWARE INTERFACES	9
3.1.3 SOFTWARE INTERFACES	9
3.1.4 COMMUNICATION INTERFACES	9
3.2 FUNCTIONAL REQUIREMENTS	9
3.2.1 USER CLASS 1 - THE CUSTOMER	9
3.2.2 USER CLASS 2 - THE ADMINISTRATOR	11
3.3 PERFORMANCE REQUIREMENTS	12
3.4 DESIGN CONSTRAINTS	12
3.5 SOFTWARE SYSTEM ATTRIBUTES	12
3.5.1 RELIABILITY	12
3.5.2 AVAILABILITY	13
3.5.3 SECURITY	13
3.5.4 MAINTAINABILITY	13
3.5.5 PORTABILITY	14

1. Introduction

The Airline Reservation System project is an implementation of an Airline Ticketing website. In this section you can find general information about the project, such as the purpose and the scope.

1.1. Document Purpose

The main purpose of this document is to list the requirements of the Airline Reservation System project, which helps the customers to search the availability and prices of various airline tickets. It also describes the functionalities, interface and design of the application. The document is intended to be read and approved by the client and used by the development team.

1.2. Product Scope

The Airline Reservation System project is an implementation of a Airline Ticketing website. This project also covers various features like online registration of the users, reports generation, as well as managing all the data.

In general, this website shall be designed to perform like any other airline ticketing website available online.

Subject facet: User Interface, Searching one-way flights, Searching round trip flights, Searching multiple destinations, Flight reservations, Reservation cancellation, Online payment, Request and response for reservation cancellation, Displaying warning messages.

Usage facet: Searching, Sorting of flights, Reservation of tickets, Managing existing reservation, Managing flight details, Keeping the flights up to date.

IT facet: Database, Web-based software system, AAS for logins, Performance maintenance.

Development facet: Internal policy and culture of the airlines company should be taken under consideration.

1.3. Definitions, Acronyms and Abbreviations

Definitions

#	Term	Definition
1	User	Someone who interacts with the web application
2	Admin/Administrator	System administrator who is given specific permission for managing and controlling the system
3	Web-Portal	A web application which present special facilities for the airline company and admin.

4	Stakeholder	Any person who has interaction with the system who is not a developer
5	PLanguage	Planning language. The basic principle is to use a set of closely defined identifiers to describe and quantify specific elements of the requirements
6	GIST	A short, simple description of the concept contained in a PLanguage statement
7	SCALE	The scale of measure used by the requirement contained in a PLanguage statement
8	METER	The process or device used to establish location on a SCALE contained in a PLanguage statement
9	MUST	The minimum level required to avoid failure contained in a PLanguage statement
10	PLAN	The level at which good success can be claimed contained in a PLanguage statement
11	WISH	A desirable level of achievement that may not be attainable through available means contained in a PLanguage statement
12	TAG	A unique, persistent identifier contained in a PLanguage statement

Acronyms

#	Acronym	Meaning
1	ARS	Airline Reservation System
2	AAS	Authentication and Authorization System
3	SQL	Structured Query Language
4	SSL	Secure Sockets Layer
5	IEEE	The Institute of Electrical and Electronics Engineers
6	SRS	Software Requirements Specification
7	CPU	Central Processing Unit
8	OS	Operating System

Abbreviation

#	Abbreviation	Meaning
1	RAT	Rational
2	DEP	Dependency
3	DESC	Description

1.4. References

1. IEEE Software Engineering Standards Committee, "IEEE Std 830-1998, IEEE Recommended Practice for Software Requirements Specifications", October 20, 1998.
2. http://www.syque.com/quality_tools/tools/Tools104.htm
3. http://micro-os-plus.sourceforge.net/wiki/SRS_Guide

1.5. Overview

This document contains all of the software requirement specifics. It contains a general description of the types of users who will be using our application, how it is going to work, and what technologies we are using to make it work.

The remainder of this document includes two chapters and appendices. The second one provides information about the application functionalities. This chapter also presents different types of stakeholders and constraints.

The third chapter includes detailed description about the requirements specification.

2. Overall Description

This section will give an overview of the whole system. The second one provides information about the application functionalities. This chapter also presents different types of stakeholders and constraints. At last, the constraints and assumptions for the system will be presented.

2.1. Product Perspective

This system will consist of one web portal. A Web portal will be used for managing the information about the flights and the system as a whole. Since this is a data-centric product it will need somewhere to store the data. For that, a database will be used. The web portal will use the database to get data, but also to add and modify data.

2.2. Product Functions

The application is intended for two types of audiences. One is the customer and the other is the administrator of the website.

The users will be able to search for flights. Flight details includes the originating flight terminal and destination terminal, along with the stops in between and the arrival and departure times, the number of seats booked/available seats between two destinations and the type of plane.

The result of the search will be viewed in a list view

A user can create, edit and delete an online account. the user can access the home page where they can search for flights and see their price and the number of remaining seats for that price and they can buy tickets for multiple flights. The website also provides the contact details of the company and instructions on how to book a flight. Moreover, clients can create an account so they are able to make reservations and receive offers and deals. The reservation includes customer details, code number, flight number, date of booking, date of travel and the chosen type of payment.

Administrator - this type of user can login/logout, manage the lists of reservations, flights and clients, modify details of the web application and send email confirmations to the customers.

2.3. User Characteristics

The system is intended to anyone who wants to search or buy plane tickets.

There are two kinds of users for the Airline Reservation System. One is the customer and the other is the administrator.

The system is designed to be user friendly, so the customer doesn't need any training or technical background to use the application and can successfully use the application regardless of their educational level or experience. The administrator does need training to use the application and is responsible to provide the customers with instructions on how to make flight reservations.

2.4. Operating Environment

2.4.1 Hardware

- **Web application server:**

Memory: 8GB

CPU: 4 core intel xenon@2.2GHz

Storage: HDD 10GB

Network: 10Gbps

- **Database server:**

Memory: 16GB

CPU: 4 core intel xenon@2.2GHz

storage: HDD 100GB

2.4.2 Software

- **Web application server**

OS: Windows Server 2016

Application Server: Microsoft IIS

Other applications: .NET Core 3.1 Runtime

- **Database server**

OS: Linux RedHat Server

Database Engine: SQL Server

2.5. Constraints

Regulatory policies: No field must be left empty

Control functions: The application must be user-friendly and display appropriate error messages

Parallel operations: The application must allow many users simultaneously

Reliability requirements: Data redundancy must be avoided

Higher order language requirements: C#

The application fetches data from the database over the internet, so the internet connection is also a constraint for the application.

The database should allow a big amount of data. The application will be designed in such a way that it can be run on the latest versions of the most used operating systems and browser. The .NET technology shall be used to implement the web application and SQL Server shall be used to manage the database. The user should have a browser installed on his system.

2.6. Assumptions and Dependencies

A booking/cancellation of a flight from any source to any destination, giving connected flights in case no direct flight between the specified Source-Destination pair exist.

It is assumed that the user has an internet access and can do online payments. The performance of ARS depends on the quality and speed of the internet connection.

2.7. Apportioning of Requirements

Some functions may be implemented in the future, such as the development of a mobile application that will allow users to store their boarding passes offline.

3. Specific Requirements

This section of the SRS contains the detailed description of the software requirements and allows designers to design a system to satisfy the requirements, and testers to test that the system satisfies the requirements.

3.1. External Interface Requirements

This section of the SRS includes detailed description of all inputs into and outputs from the software system.

3.1.1 User Interfaces

The user will see the home page when accessing the web application. (Figure 1)

The wireframe illustrates the layout of the home page. At the top, there is a header section containing a 'Social media icons' box on the left, and 'Currency' and 'Language' dropdown menus followed by a 'Sign in/ Create an account' button on the right. Below this is a navigation bar with a 'Logo' on the left and a list of links: 'HOME', 'ABOUT', 'FLIGHT', 'BLOG', 'CONTACT', and 'ACCOUNT'. The main content area is divided into several sections: a large 'Image' placeholder, a search and filter section with input fields for 'From', 'To', 'Dep Date', 'Arr Date', 'Passengers', and 'Class', along with a 'Search' button; a large 'Flights' section; and a 'Footer' section at the bottom.

Figure 1

The 'Booking Flight' form is structured as follows:

- Navigation Bar:** Includes a 'Logo' and a menu with 'HOME', 'ABOUT', 'FLIGHT', 'BLOG', 'CONTACT', and 'MY NAME'.
- Form Header:** A box labeled 'Booking Flight'.
- Personal Information:** A section containing input fields for 'Full Name', 'Address', 'Date of Birth', 'Nationality', 'Sex', 'Phone Number', and 'Email Address'.
- Flight Summary:** A section containing sub-sections for 'Departure Flight Details' and 'Arrival Flight Details'.
- Baggage:** A section labeled 'Baggage' with a sub-label 'Types of baggage'.
- Payment:** A large section labeled 'Payment'.
- Submit:** A 'Submit' button located below the 'Payment' section.
- Footer:** A box labeled 'Footer' at the bottom of the form.

Figure 2

The 'My Flights' form is structured as follows:

- Navigation Bar:** Includes a 'Logo' and a menu with 'HOME', 'ABOUT', 'FLIGHT', 'BLOG', 'CONTACT', and 'MY NAME'.
- Form Header:** A box labeled 'My Flights'.
- Upcoming Flights:** A section labeled 'Upcoming Flights' containing:
 - Flight #1:** A sub-section containing a 'Flight details' box.
 - Buttons:** Two buttons, 'Manage Reservation' and 'Cancel Reservation', located below the 'Flight details' box.

Figure 3

The user needs to have an account to be able to book tickets and manage reservations. The page where the customer will complete a reservation is shown in Figure 2. The customers will have a profile page where they can manage reservations (Figure 3).

Figure 4

3.1.2 Hardware Interfaces

3.1.3 Software Interfaces

3.1.4 Communication Interfaces

FTP is used to upload or download the files from the server.

3.2. Functional Requirements

3.2.1 User Class 1 - The Customer

3.2.1.1 Functional requirement 1.1

ID: FR1

TITLE: *User Registration*

DESC: The customer should be able to register. The user must provide username, password and email address.

RAT: In order for a user to register.

DEP: None

3.2.1.2 Functional requirement 1.2

ID: FR2

TITLE: User Login

DESC: After the user has registered then he/she will be able to log in to the application. The information will be saved and the user will be logged in automatically.

RAT: In order for a user to register.

DEP: FR1

3.2.1.3 Functional requirement 1.3

ID: FR3

TITLE: Search

DESC: Customer should be able to search flights for a specific date for one-way trips, for multiple destinations after they logged in. Search results should enable customers to find the most recent and relevant booking options.

RAT: In order for a customer to search for ticket flights.

DEP: FR2

3.2.1.4 Functional requirement 1.4

ID: FR4

TITLE: Search result

DESC: Search results can be viewed as a list. Customer should be able to see all the possible flights based on the information he entered and sort the list of possible flights by price, by flight duration.

RAT: The way results are displayed in a list.

DEP:FR3

3.2.1.5 Functional requirement 1.5

ID: FR5

TITLE: Book a flight

DESC: Given that a user is logged in, he/she can choose a particular destination and make a reservation

RAT: To enable the users to view the different flights available and make a reservation

DEP:FR2, FR3

3.2.1.6 Functional requirement 1.6

ID: FR6

TITLE: Reservation cancellation

DESC: Customer should be able to request reservation cancellation. Customer should be able to see given response to reservation cancellation request.

RAT: To allow the customer to cancel the reservation

DEP: FR2, FR3, FR5

3.2.1.7 Functional requirement 1.7

ID: FR7

TITLE: Payment method

DESC: System should demand customer to choose the payment method. System should offer payment via manually entering card details as default payment method to customer. System should be able to process the payments done by the customer via manually entering the card details.

RAT: To allow the customer to choose the payment method

DEP: FR2, FR5

3.2.1.8 Functional requirement 1.8

ID: FR8

TITLE: Booking confirmation

DESC: Booking confirmation should be sent to user to the specified contact details.

RAT: In order for a customer to receive the booking confirmation

DEP: FR2, FR5, FR7

3.2.2 User Class 2 - The Administrator

3.2.2.1 Functional requirement 2.1

ID: FR9

Feature: Administrator log in

The administrator should be logged in in order to access the account

Scenario: Successful log-in

Given the administrator wants to log in When the administrator logs in with an administrator account Then the administrator should be logged in as an administrator

3.2.2.2 Functional requirement 2.2

ID: FR10

Feature: Manage flights

The administrator should be able to manage the flights

Scenario: Add a flight

Given the administrator is logged in, he should be able to add new flights to the system

Scenario: Modify a flight

Given the administrator is logged in, he should modify the details of existing flights

3.2.2.3 Functional requirement 2.3

ID: FR11

Feature: Manage cancellations

The administrator should see reservations cancellation requests

Feature: Accept cancellations

The administrator should be able to accept reservation cancellation requests

Feature: Reject cancellations

The administrator should be able to reject reservation cancellation requests

3.3. Performance requirements

ID: QR1

TAG: Response Time

GIST: The fastness of the page load.

SCALE: Seconds.

METER: Measurements done during testing.

MUST: No more than 3 seconds.

WISH: No more than 1 second.

3.4. Design constraints

ID: QR2

TAG: Application Memory Usage

GIST: The amount of memory occupied by the application.

SCALE: MB.

METER: Measurements done during testing.

MUST: No more than 256 MB.

PLAN: No more than 128 MB.

WISH: No more than 100 MB.

MB: DEFINED: Megabyte.

3.5. Software system attributes

3.5.1 Reliability

ID: QR3

TAG: System Reliability

GIST: The reliability of the system

SCALE: The reliability that the system gives the right result on a search.

METER: Measurements done during testing.

MUST: More than 98% of the searches.

PLAN: More than 99% of the searches.

WISH: 100% of the searches.

3.5.2 Availability

ID: QR4

TAG: System Availability

GIST: The availability of the system when it is used. All software upgrades, patches and fixes should be done without shutting down the application. There should be disaster recovery environment to handle natural disasters.

SCALE: The average system availability (not considering network failing).

METER: Measurements done during testing.

MUST: More than 98% of the time.

PLAN: More than 99% of the time.

WISH: 99.999% of the time.

ID: QR5

TITLE: Internet Connection

DESC: The application should be connected to the Internet.

RAT: In order for the application to communicate with the database.

DEP: none

3.5.3 Security

ID: QR6

TAG: Communication Security

GIST: Security of the communication between the system and server.

SCALE: It must be ensured that access will be provided to the authorized persons through user ID and password. The system, at any time, should be accessed only by the authenticated users.

Network communications should use cryptographic protocols such as SSL. The system is required to end the session automatically, when an open session is not used for a specific period of time.

METER: Attempts to get user-name and password through obtained messages during testing.

MUST: 100% of the exchanged of information between client and servers in the communication of a login session should be encrypted.

3.5.4 Maintainability

ID: QR7

TITLE: Application extensibility

DESC: The software will be developed by implementing the concept of modularity which in turn reduces the complexity involved in maintaining it. The application should adopt standards based integration for extensibility and scalability. The code should be written in a way that it favors implementation of new functions.

RAT: In order for future functions to be easily implemented to the application.

DEP: none

ID:QR8

TITLE: Application testability

DESC: All code components should be thoroughly tested and the test coverage should be more than 95%. There are four levels of testing: unit testing, integration testing, system testing, user acceptance testing.

RAT: In order to test the application.

DEP: none

3.5.5 Portability

ID:QR9

TITLE: Application portability

DESC: The web application interface should support any modern browser (Microsoft Edge, Google Chrome, Mozilla Firefox, Opera and Safari). The server side should be run on either Windows server or Linux server.

RAT: In order to ensure the adaptability of the application.

DEP: none