Software Requirements Specification

for

Airline Reservation System

Version 1.0

Prepared by

Group Name: TDS

Fecioru Alina - Maria	**CEN 3.1**
Pîrvu Andrei - Cătălin	**CEN 3.2**
Popa Radu - Mircea	**CEN 3.2**

Instructor:	**Sbora Cătălin**
Course:	**Software Engineering**
Date:	**20.03.2020**

TABLE OF CONTENTS II

3 SPECIFIC REQUIREMENTS 1

3.1 External Interface Requirements 1
3.1.1 User Interfaces 1
3.1.2 Hardware Interfaces 3
3.1.3 Software Interfaces 3
3.1.4 Communication Interfaces 3
3.2 Functional Requirements 3
3.2.1 User Class 1 - The Customer 3
3.2.2 User Class 2 - The Administrator 5
3.3 Performance Requirements 6
3.4 Design Constraints 6
3.5 Software System Attributes 6
3.5.1 Reliability 6
3.5.2 Availability 7
3.5.3 Security 7
3.5.4 Maintainability 7
3.5.5 Portability 8

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)

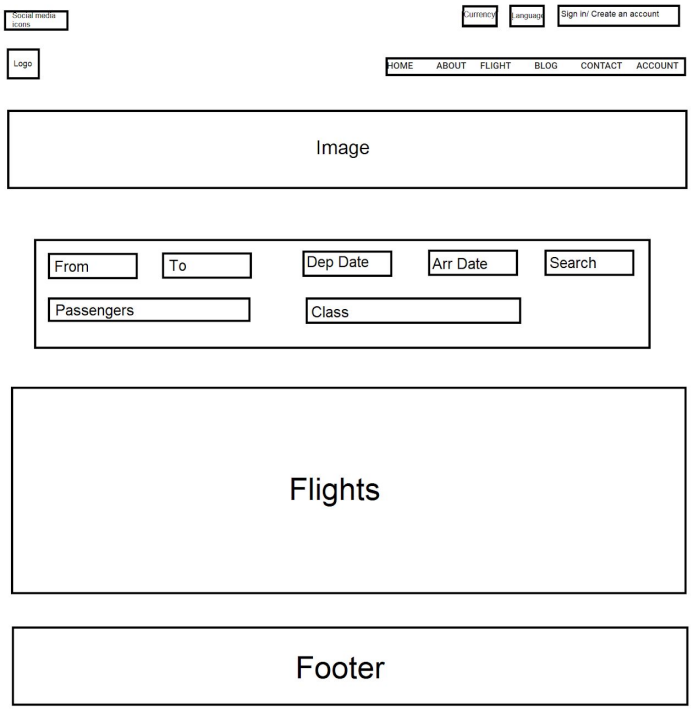

Figure 1

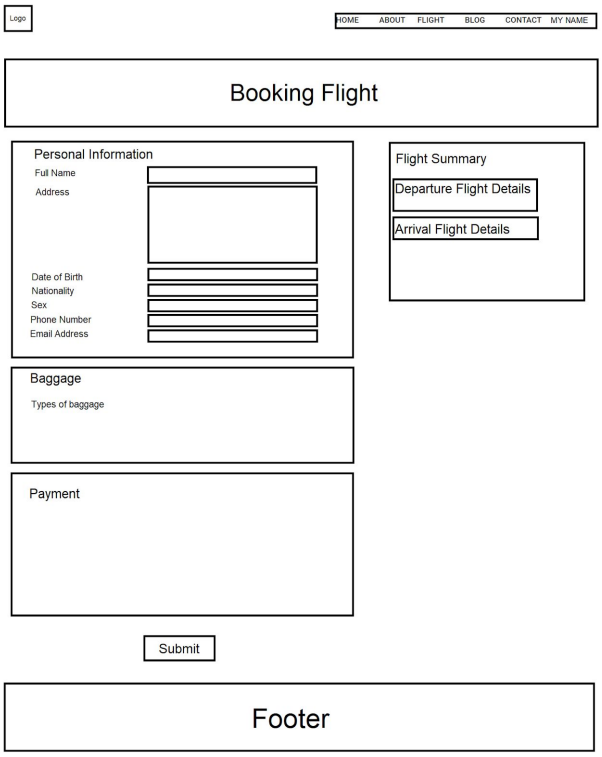

Figure 2

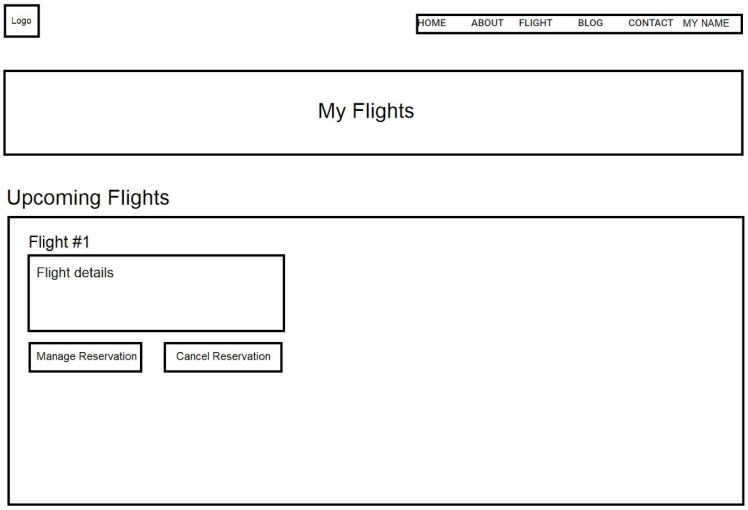

Figure 3

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).

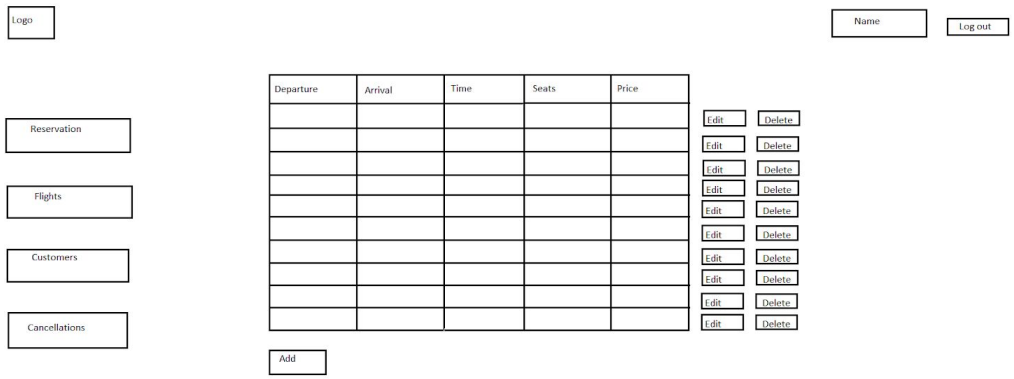

Figure 4

The interface of the administrator showing the list of flights. The administrator can add, edit and delete flights. He/ she can also see the list of reservations, clients and cancellations. (Figure 4)

3.1.2 Hardware Interfaces

The application does not have any direct hardware interfaces. It can be run on any device with the latest versions of the most used operating systems and browsers.

3.1.3 Software Interfaces

Entity Framework Core will be used to simplify the mapping between objects to the tables and columns of a relational database.

3.1.4 Communication Interfaces

The application will use HTTPS with an SSL certificate (2048 Bit SHA2 SSL/TLS Public Key Encryption) for enhanced security.
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

Scenario: Accept cancellations

The administrator should be able to accept reservation cancellation requests

Scenario: 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