Software Requirements Specification

for

Pocket Plane

Version <3.0> -APPROVED

Prepared by

Group Name: Echipa Racheta

Efrem Ion-Ștefan	**CEN 3.1**
Hagiu Teodora-Adriana	**CEN 3.1**
Mitre Flavia-Antonia	**CEN 3.1**
Barbu Mircea-Alexandru	**CEN 3.1**

Instructor:	***Stoica Spahiu-Cosmin***
Course:	**Project II – Use of Databases**
Date:	**10.03.2020**

Table of Contents

1 Introduction

This part of our document gives an overview of everything that is written in this Software Requirements Specification document and describes the process of developing our application.

1.1 Document Purpose

The purpose of this document is to give a detailed description of the requirements for our software. It will illustrate the purpose and complete description of the development of the system. It will also explain system constraints, interface, interactions with others external users and how user-friendly the application was made.

This document is primarily intended to be used by the client and by the developing team, to understand the development of our application. It will present the scope of our application, its functionality alongside the end users privileges on the app.

1.2 Product Scope

The application "Pocket Plane" was made for managing the flights of an Airline. The Airline admins will be able to manage the list of flights available for the airline. They will have the permission to add new flights for which tickets can be purchased by clients or cancel planned flights. Clients will be able to buy tickets to different flights and manage them on the application.

The goal of this software application is to facilitate the selling and buying of tickets, thus helping the stakeholder's business on the commercial flights market.

1.3 Definitions, Acronyms and Abbreviations

No.	Definitions	Description
1	Application server	An application server is a software framework that provides both facilities to create web applications and a server environment to run them.
2	Web server	A web server is server software, or hardware dedicated to running this software, that can satisfy client requests on the World Wide Web.
3	Database	A database is an organized collection of data, generally stored and accessed electronically from a computer system.
4	User	Someone who interacts with the web application
5	Admin/Administrator	System administrator who is given specific permission for managing and controlling the system
6	Stakeholder	Any person who has interaction with the system who is not a developer.

No.	Acronyms	Description

1	SRS	Software Requirement Specifications
2		
3		

No.	Abbreviations	Description
1	App	Application
2		
3		

1.4 References and Acknowledgments

N.o.	Reference	Description
1	https://www.tutorialspoint.com/asp.net/index.htm	This site was used to accomplish a base of knowledge over asp.net core, that shall be used in developing the application later
2	MISRA-C: 2004	Used for software quality attributes.
3	*IEEE830 standard*	The current document uses this standard template.

2 Overall Description

This section will present an overview of whole system. The system will be explained in its context to show how the system interacts with users and introduce the basic functionality of it. It will

also describe what functionality is available for each type of user. Also, the constraints and assumptions for the system will be presented.

2.1 Product Functionality

The application "Pocket Plane" was developed for the selling of flight tickets and administration of passengers and the flights of an airline. The Application has client users and administrators. Every type of user has a different kind of account and a different set of permissions.

Every user can access his account with a username and a password, this connects the application to a database. It allows a client user to buy tickets for a flight/flights, to cancel a reservation, to manage/change their data, to delete their account.

The application allows an administrator to create, post, modify, cancel flights and to manage/change their data.

2.2 End Users and Characteristics

Each user has a unique account (with a username and a password) with two different types of users' levels: admin and client user. That includes multiple actions for 2 types of user:

- Client Users:
  - Buy tickets for flights
  - Cancel a reservation
  - Manage/change their data
  - Manage reservations
  - Deletes their client

  The client can be any computer-user that navigates on the internet. The person needs to have basic computer skills and knowledge about internet browsing. Also, the client can be any person that wishes to buy a plane ticket.

- Admin Users:
  - Create Flights
  - Modify/Delete Flights
  - Manage/change their data

  The admin must be a person hired by the airline company to maintain the current flights updated on the site and add new ones. The admin must have a good understanding of computer usage and basic data-base management skills.

2.3 System Stakeholders

The owner of this application is the Pocket Plane Airline Company. Below is a comprehensive list of the stakeholders of this software application. The stakeholders will have the privileges of potential profits made from this app alongside the potential risk that should be managed by them.

2.3.1 Internal Stakeholders:

- The Owners of the Airline
- Employees who own shares/bonds of the Airline

2.4 Operating Environment

Below are the minimum system requirements so that a user can run our application:

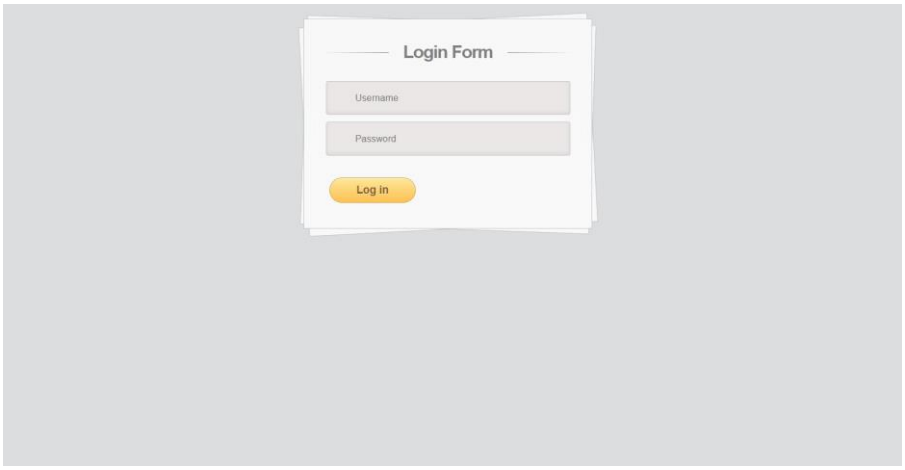
- Microsoft Windows 7 at least installed
- Minimum Internet Speed: Above 10 Mb/s
- Microsoft .NET Core 3.1
- One of the next modern browsers:
  - Google Chrome 42 or newer
  - Opera 36 or newer
  - Microsoft Edge 12 or newer
  - Firefox 68 or newer

3 Specific Requirements

3.1 User Interface

Before logging on:

S



Register

First Name: *First Name ...*
Last Name: *Last Name ...*
CI/Passport: *CI/Passport ...*
Username: *Username...*
Password: *New Password...*
Confirm Password: *Confirm New Password...*
Address: *Address ...*
E-mail: *E-mail ...*
Phone Number: *Phone Number ...*

Account Type:
(•) User
( ) Admin
If Admin selected please enter: *Admin Verification Code ...*

Log In | Register

After the log-in for client user:

RESERVATION PAGE

LOG OUT

LOG OUT

MENU

HOME PAGE
FUTURE FLIGHTS
EDIT ACCOUNT
PAST FLIGHTS

Buy Ticket

FLIGHT NO.	DEPARTURE	DEPARTURE AIRPORT	DEPARTURE HOUR	DATE D.M.Y	ARRIVAL	ARRIVAL AIRPORT	ARRIVAL HOUR
PP2020	CRAIOVA	CRAIOVA AIRPORT	07:30	10.02. 2020	LONDON	LUTON AIRPORT	11:15

CANCEL

Class:
Price:

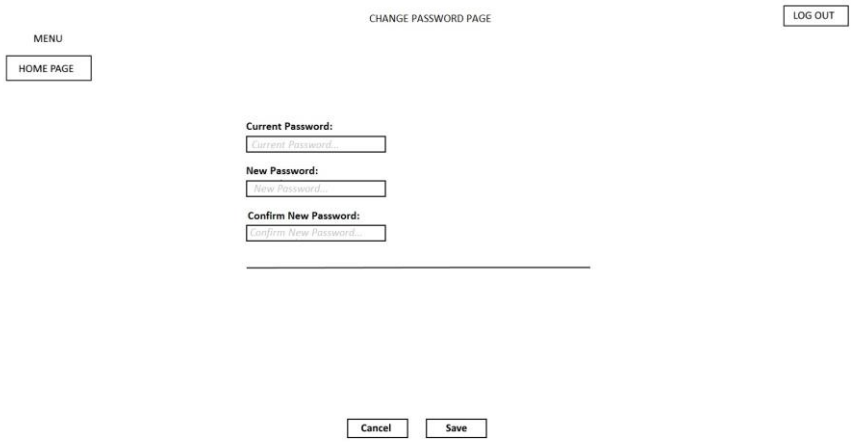
CI/Passport: *CI/Passport ...*
Nationality: *Nationality ...*
First Name: *First Name:*
Last Name: *Last Name:*

Select Payment Method: *Visa / Master Card / etc*

Card Number: *Card Number ...*
Expiration Date: *Expiration Date ...*
Cardholder Name: *Cardholder Name ...*
CVV: *CVV ...*

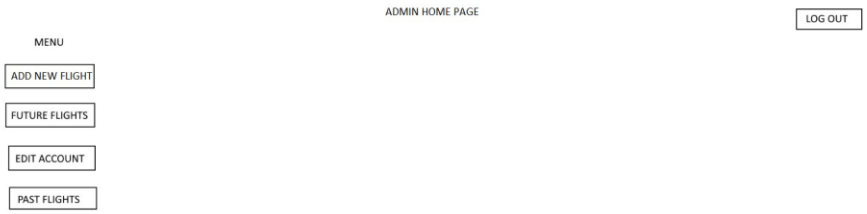
Cancel | Buy

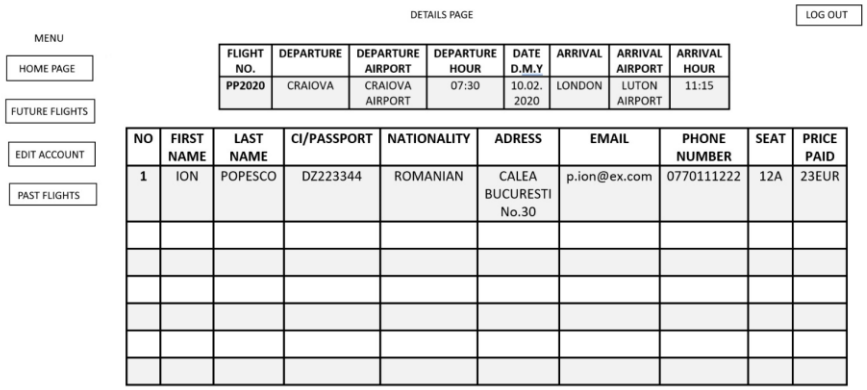

LOG OUT
MENU
HOME PAGE
FUTURE FLIGHTS
EDIT ACCOUNT
PAST FLIGHTS
Edit Account
First Name:
First Name ...
Address:
Address ...
Last Name:
Last Name ...
E-mail:
E-mail ...
CI/Passport:
CI/Passport ...
Phone Number:
Phone Number ...
Change password?
Delete my Account
Cancel
Save


CHANGE PASSWORD PAGE
LOG OUT
MENU
HOME PAGE
Current Password:
Current Password...
New Password:
New Password...
Confirm New Password:
Confirm New Password...
Cancel
Save


SEARCH FLIGHTS PAGE
LOG OUT
MENU
HOME PAGE
RESERVATIONS
EDIT ACCOUNT
FLIGHT NO.
DEPARTURE
DEPARTURE AIRPORT
DEPARTURE HOUR
DATE D.M.Y
ARRIVAL
ARRIVAL AIRPORT
ARRIVAL HOUR
PRICES FROM
PP2020
CRAIOVA
CRAIOVA AIRPORT
07:30
10.03. 2020
LONDON
LUTON AIRPORT
11:15
23 EUR
BUY TICKET

After the log-in for administrator-user:






PAST FLIGHTS PAGE
LOG OUT
MENU
HOME PAGE
FUTURE FLIGHTS
EDIT ACCOUNT
FLIGHT NO.
DEPARTURE
DEPARTURE AIRPORT
DEPARTURE HOUR
DATE D.M.Y
ARRIVAL
ARRIVAL AIRPORT
ARRIVAL HOUR
PRICE FROM
PP2020
CRAIOVA
CRAIOVA AIRPORT
07:30
10.02. 2020
LONDON
LUTON AIRPORT
11:15
23 EUR
DETAILS


FUTURE FLIGHTS PAGE
LOG OUT
MENU
HOME PAGE
PAST FLIGHTS
EDIT ACCOUNT
FLIGHT NO.
DEPARTURE
DEPARTURE AIRPORT
DEPARTURE HOUR
DATE D.M.Y
ARRIVAL
ARRIVAL AIRPORT
ARRIVAL HOUR
PRICES FROM
PP2020
CRAIOVA
CRAIOVA AIRPORT
07:30
10.03. 2020
LONDON
LUTON AIRPORT
11:15
23 EUR
EDIT
DELETE


LOG OUT
MENU
HOME PAGE
FUTURE FLIGHTS
EDIT ACCOUNT
PAST FLIGHTS
Edit Flight
Flight Number:
Flight Number ...
Date:
Date ...
Departure:
Departure ...
Departure Airport:
Departure Airport ...
Arrival:
Arrival ...
Arrival Airport:
Arrival Airport ...
Departure Hour:
Departure Hour ...
Arrival Hour:
Arrival Hour ...
Number of Seats:
Number of Seats ...
Price:
Price ...
Delete This Flight
Cancel
Save

3.2 Functional Requirements

3.2.1 User Class 1 – The User

3.2.1.1 Functional requirement 1.1

Title: User registration

Description: In order to use the application, each new user must go through the registration process. In order to create a new account, the user must provide his full name, a username, a password, etc… as well as his type of user.

Dependencies: None

3.2.1.2 Functional requirement 1.2

Title: User Log-in

Description: Given that the user has registered, in order to use the application, each user must go through the log in process. In order to enter in account, the user must provide username and his password.

Dependencies: FR1

3.2.1.3 Functional requirement 1.3

Title: User changes password

Description: In case the user wants to change his password, he must go through the change password process. In order to change the password, the user must provide his old password and his new password.

Dependencies: FR2

3.2.1.4 Functional requirement 1.4

Title: User can search for a ticket

Description: The client can search for a ticket to buy.

Dependencies: FR2

3.2.1.5 Functional requirement 1.5

Title: User can buy a ticket

Description: The client user can buy tickets. In order to buy a ticket, the user must provide the details needed. He can also buy tickets for other persons, as long as he provides the required information.

Dependencies: FR2

3.2.1.6 Functional requirement 1.6

Title: User can manage reservations
Description: The client can see his reservations. He can choose to see details or cancel each reservation through specific buttons.
Dependencies: FR2

3.2.1.7 Functional requirement 1.7

Title: User can log out.
Description: The user can log out of his account.
Dependencies: FR2

3.2.1.8 Functional requirement 1.8

Title: User can delete account
Description: The user can delete his account.
Dependencies: FR2

3.2.1.9 Functional requirement 1.9

Title: User can edit account
Description: The user can edit his account.
Dependencies: FR2

3.2.2 User Class 2 – The Administrator

3.2.2.1 Functional requirement 2.1
Title: User registration
Description: In order to use the application, each new user must go through the registration process. In order to create a new account, the user must provide his full name, a username, a password, etc… as well as his type of user.
Dependencies: None

3.2.2.2 Functional requirement 2.2

Title: User Log-in
Description: Given that the user has registered, in order to use the application, each user must go through the log in process. In order to enter in account, the user must provide username and his password.
Dependencies: FR1

3.2.2.3 Functional requirement 2.3

Title: User changes password
Description: In case the user wants to change his password, he must go through the change password process. In order to change the password, the user must provide his old password and his new password.
Dependencies: FR2

3.2.2.4 Functional requirement 2.4

Title: User can display flights.
Description: The administrator can display all flights.
Dependencies: FR2

3.2.2.5 Functional requirement 2.5

Title: User can add a flight
Description: The administrator can add new flights.
Dependencies: FR2

3.2.2.6 Functional requirement 2.6

Title: User can change a flight
Description: The administrator can change the information of a specific flight. Given that the user displayed all flights, from that page he can add the button edit to change a flight.
Dependencies: FR4

3.2.2.7 Functional requirement 2.7

Title: User can delete a flight
Description: The administrator can delete a specific flight. Given that the user has enterd in the edit mode, from there he can press the”delete” button to delete the flight.
Dependencies: FR 5

3.2.2.8 Functional requirement 2.8

Title: User can log out.
Description: The administrator can log out of his account.
Dependencies: FR2

3.2.2.9 Functional requirement 2.9

Title: User can delete account
Description: The administrator can delete his account.
Dependencies: FR2

3.2.2.10 Functional requirement 2.10

Title: User can edit account
Description: The user can edit his account.
Dependencies: FR2

3.3 Use Case Diagram

N/A.

4 Non-functional Requirements

4.1 Software Quality Attributes

This section will present the rules used in creating this application. This shall avoid future software problems/bugs by having a professional approach towards development:

- Rule 1: Source code shall only use /* … */ style comments
- Rule 2: Identifiers in an inner scope shall not use the same name as an identifier in an outer scope, and therefore hide that identifier
- Rule 3: All libraries used in production code shall be written to comply with the provisions of this document, and shall have been subject to appropriate validation
- Rule 4: For each function parameter the type given in the declaration and definition shall be identical, and the return types shall also be identical.
- Rule 5: A typedef name shall be a unique identifier.
- Rule 6: The value of a complex expression of floating type shall only be cast to a floating type that is narrower or of the same size.
- Rule 7: The sizeof operator shall not be used on expressions that contain side effects.
- Rule 8: There shall be no unreachable code.
- Rule 9: All if … else if constructs shall be terminated with an else clause.
- Rule 10: The final clause of a switch statement shall be the default clause
- Rule 11: The three expressions of a for statement shall be concerned only with loop control.
- Rule 12: All exit paths from a function with non-void return type shall have an explicit return statement with an expression.

5 Other Requirements

5.1 Security

We are assuring the security of our Web Application through the use of certain methods:

- "https" will be used instead of "http"

- Login and Password for the accounts of every user. It has different type of users and every user has access constraints because every type of user can make specific actions.

5.2 Safety

N/A.