Acceptance Testing - TravelDream (buzzoni-alvarez-project)

Paolo Manca - Francesco Nero

February 12, 2014

Contents

1 Introduction and setup

1.1 Introduction

This document will provide setup, test cases and their results for the acceptance testing of the project “buzzoni-alvarez-project” (link).

1.2 Setup

The project EAR should be deployed to a running instance of a glassfish server, configured as instructed by the installation manual (link).

2 Test cases

After analyzing the RASD document, the following test cases were considered in order to cover all of the requirements for the application. Note that, since this is the first release of the product (as pointed out in the document attached with the implementation delivery), no functionality other than registration, login and logout were implemented for the user. With this in mind, no test cases were provided for these use cases, as they would have most certainly failed.

2.1 User test cases

These tests test functionality that is provided to the customer actor.

2.1.1 Test case 1 : Unregistered user registration

Functional requirements tested: [FR1.1] of RASD

Actions performed 1: Clicked on register on homepage. Nickname: testUser | Email: test@test.test | password1: test | password 2 tes

Expected result: registration fails, as passwords entered do not match.

Result: as expected.

Actions performed 2: Clicked on register on homepage. Nickname: testUser | Email: test@test.test | password1: test | password 2 test

Expected result: registration completes.

Result: as expected.

2.1.2 Test case 2 : Registered user login

Functional requirements tested: [FR2.1] of RASD

Actions performed 1: Clicked on login on homepage. Nickname: test | password: test

Expected result: login fails as there is no “test” user. Failure of login should bring to an error page.

Result: there is a redirect to an empty page, instead of the expected error page.

Actions performed 2: Clicked on login on homepage. Nickname: testUser | password: test

Expected result: login completes as expected, and redirects to homepage, with a new session for the user.

Result: as exptected.

2.1.3 Test case 3 : Registered user logout

Functional requirements tested: [FR1.1] of RASD

Assumptions: the customer registered previously as testUser is logged in to the website.

Actions performed: Clicked on logout on homepage.

Expected result: the session associated with the logged in user ends.

Result: logged out as expected. Also redirected to homepage.

2.2 Employee test cases

These tests test functionality that is provided to the employee actor.

2.2.1 Test case 1 : Registered employee login

Functional requirements tested: [FR3.1] of RASD

Assumptions: employee “ele” is registered in the DB.

Actions performed: opened employee homepage login (custom URL). Filled in: Nickname: “ele” | password: password

Expected result: login completes as expected, and redirects to employee homepage, with a new session for the employee.

Result: as expected.

2.2.2 Test case 2 : Registered employee logout

Functional requirements tested: [FR3.1] of RASD

Assumptions: the employee “ele” is logged in to the website.

Actions performed: Clicked on logout on employee homepage.

Expected result: the session associated with the logged in employee ends.

Result: logged out as expected. Also redirected to normal homepage.

2.2.3 Test case 3 : Create product (flight)

Functional requirements tested: [FR3.2] of RASD

Assumptions: the employee “ele” is logged in to the website.

Actions performed: Clicked on “Create flight” on employee homepage. Filled in: Flight number: 1234 | From: “NO ITEMS TO SELECT FROM” | Departure airport: Ciampino | To: “NO ITEMS TO SELECT FROM” | Arrival airport: Linate | Departure date 2/19/14 | Arrival date: 2/19/14 | Price: 90.0 | Company: PigAir | Class: Best

Expected result: the flight product is created. The system notifies the employee that the product was created

Result: failure. The page notifies that there are no departure cities or destinations (the database is populated, but the drop-down list is empty, so there is no way to fulfill the requirement through the page).

Notes: hotel and excursion creation fail for the same reasons.

2.2.4 Test case 4 : Search product (flight)

Functional requirements tested: [FR3.8] of RASD

Assumptions 1: the employee “ele” is logged in to the website. Some flight products are present in DB.

Actions performed 1: Clicked on “Search flight” on employee homepage without filling in constraint details.

Expected result: all flight products present in the DB are presented to the employee.

Result: as expected.

Notes: hotel and excursion products work as expected also.

Assumptions 2: the employee “ele” is logged in to the website. Only one flight with number ER14891H are present in DB.

Actions performed 2: Clicked on “Search flight” on employee homepage. Filled in: Flight number: ER14891H.

Expected result: the flight product associated to flight number ER14891H is presented.

Result: as expected.

Notes: hotel and excursion products work as expected also.

Assumptions 3: the employee “ele” is logged in to the website. At least one flight with departure from Milan is present in DB. The city Milan is present in the DB as a locality.

Actions performed 3: Clicked on “Search flight” on employee homepage. Try to select Milan fro From: field drop-down list.

Expected result: the flight products associated to departure city Milan are presented.

Result: drop-down list is empty. Cannot enter the search.

Notes: hotel and excursion products fail in the same way.

2.2.5 Test case 5 : Delete product (flight)

Functional requirements tested: [FR3.3] of RASD

Assumptions: the employee "ele" is logged in to the website.

Actions performed: Clicked on "Search flight" on employee homepage without filling in constraint details, so as to retrieve all flights. Clicked "Delete" button associated to first entry in the list.

Expected result: the flight product is created. The system notifies the employee that the product was deleted

Result: as expected.

Notes: hotel and excursion products work as expected also.

2.2.6 Test case 6 : Modify product (flight)

Functional requirements tested: [FR3.4] of RASD

Assumptions: the employee "ele" is logged in to the website. Some flight products are present in the DB.

Actions performed: Clicked on "Search flight" on employee homepage without filling in constraint details, so as to retrieve all flight products. Clicked on "Edit"next to first flight in list presented. Filled in: Flight number: 1234 | From: "NO ITEMS TO SELECT FROM" | Departure airport: Ciampino | To: "NO ITEMS TO SELECT FROM" | Arrival airport: Linate | Departure date 2/19/14 | Arrival date: 2/19/14 | Price: 90.0 | Company: PigAir | Class: Best

Expected result: the flight product is modified with the new info provided. The system notifies the employee that the product was modified correctly

Result: failure. The page notifies that there are no departure cities or destinations (the database is populated, but the drop-down list is empty, so there is no way to fulfill the requirement through the page).

Notes: hotel and excursion modification fail for the same reasons.

2.2.7 Test case 7 : Create packet

Functional requirements tested: [FR3.5] of RASD

Assumptions: the employee "ele" is logged in to the website.

Actions performed: Clicked on "Create packet" on employee homepage. Filled in: Packet name: TestPacket | Preview photo: pig.png | Starting validity date: 2/11/14 | Departure date: 2/20/14 | Description: La vispa Teresa / aveva tra l'erbetta / A volo sorpresa / gentil farfalletta Clicked "Add flights". Searched for all flights. Selected first from list.

Expected result: the packet is inserted in the system with the details provided. The system notifies the employee of the creation.

Result: as expected.

2.2.8 Test case 8: Modify packet

Functional requirements tested: [FR3.6] of RASD

Assumptions: the employee “ele” is logged in to the website.Some packets are present in the DB.

Actions performed: Clicked on “Search packet” on employee homepage without filling in constraint details, so as to retrieve all packets. Clicked “Edit” button associated to first entry in the list.Changed Packet name to “Test2”.

Expected result: the packet is modified with the new details. The system notifies the employee of the creation.

Result: as expected.

Notes: every field is correctly updated (not only Packet name).

2.2.9 Test case 9: Delete packet

Functional requirements tested: [FR3.7] of RASD

Assumptions: the employee “ele” is logged in to the website. Some packets are present in the DB.

Actions performed: Clicked on “Search packet” on employee homepage without filling in constraint details, so as to retrieve all packets. Clicked “Delete” button associated to first entry in the list.

Expected result: the packet is removed from the system. The system notifies the employee of the removal.

Result: as expected.