TravelDream Project

Tests

Riccardo B. Desantis - matr. 765106

Contents

Chapter 1

Introduction

1.1 Purpose

This document contains and explains the testing process and the testing results of the TravelDream project developed by one of the other teams taking the course of Software Engineering 2.

1.2 Test Participants

- **Evaluation Team**:
  - Riccardo Benito Desantis (765106)
- **Development Team**:
  - Engida Selamawit Belete (815730)
  - Hu Sile (814739).

1.3 Scope

This document will test the development of the TravelDream project by the assigned team. The testing will include the following functional requirements identified by the development team on the RASD document, divided by user role:

- **Employee**:
  - **FR1**: Adding flight, hotel and excursion product details
  - **FR2**: Delete basic product information the company offers
  - **FR3**: Modify product details
  - **FR4**: Create predefined travel packages
  - **FR5**: View registered customers
  - **FR6**: View basic product details
- **Customer**:
  - **FR7**: Register/login to the system : through a user friendly user interfaces
  - **FR8**: Search for travel packages
  - **FR9**: Personalize travel packages

  - **FR10**: Add/remove basic product
  - **FR11**: Delete package
  - **FR12**: Add departure and return date
  - **FR13**: Invite friends to access their personalized travel package (sharing travel ID)
  - **FR14**: Confirm the travel

- **Customers' friends**:
  - **FR15**: Join the travel package by registering in to the system
  - **FR16**: Visualize the travel package they are invited to access
  - **FR17**: Invite friends to access the package (? Modify package) after they become a customer themselves.

1.4 Definition, acronyms, and abbreviations

The following acronyms will be used through the whole document:

- FR: Functional Requirement
- NFR: Non-Functional Requirement
- G: Goal
- RASD: Requirement And Analysis Specification
- JUnit: Java unitary tests.

1.5 References

- RASD TravelDream814739,815730.pdf
- DD TravelDream 814739,815730.pdf
- testing (2).pdf
- User Manual -2-.pdf

1.6 Overview

This document specifies the testing process, approaches and results of the development of the TravelDream project by the assigned team.

The document is organized in the following sections:

1. **Introduction**: this section describes the purpose of the document including its scope, glossary and related documents used to do it.
2. **Test background**: contains a detailed explanation of the testing environment, test cases and approaches taken into account for executing the development testing.

3. **Development validation**: contains a detailed explanation of the analysis validation of the developed software based on the RASD document, design document, user manual and source code.

4. **Development verification**: contains a detailed explanation of the design test cases and the results obtained based on the outputs of the development and the project specification.

Chapter 2

Test background

2.1 Test objectives

The present testing is for validating and verification the following goals identified by the development group on the project specifications:

- **G1**: To create an interactive website for the travel dream company that will attract several customers to join packages offered by the company
- **G2**: To allow travel dream employee creates different kinds of travel packages that will attract different customers
- **G3**: To allow Employee make modification to the different types of products
- **G4**: Enable customers use packages the TravelDream company provides and personalize their own packages
- **G5**: To allow customers invite their friends to join the packages
- **G6**: To allow many other users who are invited by the customers of the TravelDream Company join packages and be customers.

2.2 Test environment information

Test environment information	
Software tested:	TravelDream
Type	Web application using JSF, JSP, servlets and EJBs
Version	1.0
Goal	The testing is for validating and verification of the development of the TravelDream software defined in the assignment. The testing will be validated with the goals **G1**, **G2**, **G3**, **G4**, **G5**, **G6**.
Author	Engida Selamawit Belete, Hu Sile

2.3 Aspects to be tested

Among the aspects to be tested besides the functional requirements mentioned in the scope of the document are the different documents created during the development process of the project by the development team, these documents also include the source code and the execution of the final software delivered.

The documents verified against the final software are:

- RASD TravelDream814739,815730.pdf
- DD TravelDream 814739,815730.pdf
- testing (2).pdf
- User Manual -2-.pdf.

2.4 Approach

For validating and verifying the development project I've used the analysis approach and the testing approach, therefore a complete view of the development is achieved. The depth of the verification and validation was limited by the tester available time, therefore the execution of the tests are oriented to the functional requirements specified on the project description document.

In the testing approach, black box and glass box tests were executed, that is: the functional requirements were verified not only using automatic test tools and executing the methods implemented by the development team in the business layer of the JEE architecture, but also by interacting with the web interface provided by the web client developed by the development team.

In the analysis approach, a static analysis was executed on the RASD and Design Document published by the development team against the source code delivered for the test phase.

2.5 Hardware and software requirements

The minimum requirements defined in the RASD are here presented.

2.5.1 Hardware requirements

- CPU: Intel Pentium IV 2.5 GHZ
- HDD: 40 GB
- RAM: 128 MB
- FSB: 533 MHz

The hardware used for the tests is:

- CPU: Intel Core i7 2 GHz
- HDD: 250 GB
- RAM: 8 GB
- FSB: 1600 MHz

so it satisfies completely the requirements.

2.5.2 Software requirements

- Software Operating System: Windows 7
- Development Environment: Eclipse (Kepler) for Java EE Developers
- Plugins required:
  - SVN: Subversive
  - GlassFish Tools for Kepler
- Java EE 7 JDK
- Application Server: GlassFish Open Source Edition 4.0
- Database: MySQL.

Even if the stated requirement is Eclipse, the team used Netbeans instead, as reported in other documents.

The software used for the tests is:

- Software Operating System: Mac OS X 10.9.1
- Development Environment: Netbeans 7.4
- Java EE 7 JDK
- Glassfish 4
- MySQL.

The different operating system in the testing system shouldn't be a problem, given the portability of the project and the servers used.

Chapter 3

Development validation

This chapter contains the validation process and results done over the software development assigned. In the last chapter of the document the most important results are summarized.

The elements considered are:

- RASD TravelDream814739,815730.pdf
- DD TravelDream 814739,815730.pdf
- the source code.

3.1 Requirements and Analysis Specification Document

The RASD document is well written, and even if it doesn't follow strictly the standard model it still has all the useful information for understanding the analysis process. Something to add:

- The specification of the project talked about the 3 kind of actors that this project is taking into account; in the analysis of it, also the "unregistered" users had to be identified and considered, because they're also acting on the system (even if it is only for initiating the registration process).
- Some mock-up of the interfaces considered would have been useful, especially for a first understanding of how the system would become when finished.

3.2 Design Document

Also the DD document doesn't follow strictly the standard model, but it still is well written and has the needed informations for the definition of the problem. But:

- Still no mock-ups are shown.

3.3 Source code

The source code comes with a user manual, providing screen shots and informations on how to run the system. The code itself is often understandable (even if arguably not commented). Some things I'd like to note are:

- The approach to the problems sometimes seems a little too complicated, probably because the team had already a background on the JEE platform. The approach followed during the lectures is more understandable, specially for me, the tester, that had no prior knowledge about such platform.

For example: the entities travel around the system in a not so clear way. It would have been much better if they had used Data Transfer Objects (DTOs). More on this in the next chapter.

- The file names defined in the Design Document (DD) aren't followed in the realization of the project, and some elements of the DD aren't present at all. They should have followed better the design they did.
- They didn't use the `@RolesAllowed` annotation ever, so they've only blocked the access to the methods via the different folders of the web pages.

Chapter 4

Development verification

4.1 Functional Requirements testing

I've tried to write JUnit tests for this project, to test the functional requirements without using the graphical Web interface. Sadly, the Enterprise JavaBeans that are needed for this step aren't defined (because the Managed Beans are used everywhere instead).

After hours of trying, I've skipped this part of the tests. I'm sure that it's my fault, but the little time available for the tests, and surely my little knowledge about the topic are really forcing me to go on with only the tests via the GUI.

4.2 Graphic User Interface (GUI) testing

In this section I've checked if the navigability of the Web tier satisfies the functional requirements and therefore the goals of the system.

4.2.1 User Login

User Login	
Code	T001
Purpose	Checks if a client can login with an existent username and password, and if it is redirected from the login page to the right user page.
Exec. Environment	The login page, reachable clicking on the “login” link from the homepage.
Input 1	1432146241, password
Input 2	0, admin
Input 3	1234, 1234
Expected Output 1	Navigate to the consumer homepage.
Expected Output 2	Navigate to the employee homepage.
Expected Output 3	Return an error (wrong data).
Current Output 1	Navigate to the consumer homepage.
Current Output 2	Navigate to the employee homepage.
Current Output 3	Return an error (wrong data).
Result	The form correctly redirect to the right page or returns error when needed.
Note	Even if the form asks for the “Email” of the user, it actually wants the user id, that is assigned automatically at the registration.

4.2.2 User Logout

User Logout	
Code	T002
Purpose	Checks if a logged user can log out of the system, and go back to the homepage.
Exec. Environment	The user home page.
Input	Press the “Logout” link on the upper menu.
Expected Output	The user is logged out and goes back to the common home page.
Current Output	The user is logged out and goes back to the common home page.
Result	The logout link works, invalidating correctly the user session.
Note	-

4.2.3 Listing of the Products

Listing of the Products	
Code	T003
Purpose	Checks if an employee can show the list of already defined products.
Exec. Environment	The user home page of the employee logged in.
Input	Click on the link “Product” in the voice “Menu” in the menubar.
Expected Output	The products page is showed.
Current Output	The products page is showed.
Result	The list of the available products is showed.
Note	-

4.2.4 Creation of a new Product

Creation of a new Product	
Code	T004
Purpose	Checks if an employee can create a new product.
Exec. Environment	The products page, reachable by clicking on the link “Product” in the menubar, from the user home pageof the employee logged in.
Input 1	After clicking on the “Create” button, i put the ProductId 11 (an id that is already used), the AmountAvailable 11, I select from the UserId list the id 0 and I click “Save”.
Input 2	After clicking on the “Create” button, i put the ProductId 342 (an id that isn’t already used), the AmountAvailable 11, I select from the UserId list the id 0 and I click “Save”.
Expected Output 1	No product is added to the list.
Expected Output 2	The product is added to the list.
Current Output 1	The product with the same id is updated!
Current Output 2	The product is added to the list.
Result	The creation process works, and adds also to the database.
Note	First of all, the id should be automatically assigned. It isn’t clear what some of the fields in the form are. It would have been better to automatically select the id of the logged user, and it doesn’t make sense to ask for, e.g, a flight id because we first need to create this product before creating a new flight to link to this.

4.2.5 Handling of a Product

Handling of a Product	
Code	T005
Purpose	Checks if an employee can view, edit or remove a product.
Exec. Environment	The products page, reachable by clicking on the link “Product” in the menubar, from the user home page of the employee logged in.
Input 1	After selecting a product in the list, I click the “Delete” button.
Input 2	After selecting a product in the list, I click the “View” button.
Input 3	After selecting a product in the list, I click the “Edit” button, changing the details of the product (e.g. the AmountAvailable) and saving the modification via the “Save” button.
Expected Output 1	The product is deleted from the list.
Expected Output 2	The details about the product are showed.
Expected Output 3	The product is modified as requested.
Current Output 1	The product is deleted from the list.
Current Output 2	The product is modified as requested.
Current Output 3	The details about the product are showed.
Result	The products can be viewed, modified and removed.
Note	The previous note still stands.

4.2.6 Listing of the Excursions

Listing of the Excursions	
Code	T006
Purpose	Checks if an employee can show the list of already defined excursions.
Exec. Environment	The user home page of the employee logged in.
Input	Click on the link “Excursion” in the voice “Menu” in the menubar.
Expected Output	The excursions page is showed.
Current Output	The excursions page is showed.
Result	The list of the available excursions is showed.
Note	-

4.2.7 Creation of a new Excursion

Creation of a new Excursion	
Code	T007
Purpose	Checks if an employee can create a new excursion.
Exec. Environment	The excursions page, reachable by clicking on the link “Excursion” in the menubar, from the user home page of the employee logged in.
Input 1	After clicking on the “Create” button, i put the ExcursionId 234, the Route Milano and the Product with an id already used for another excursion, and I click “Save”.
Input 2	After clicking on the “Create” button, i put the ExcursionId 234, the Route Milano and the Product that we’ve just created, and I click “Save”.
Input 3	After clicking on the “Create” button, i put the ExcursionId 11 (which is already used), the Route Milano and the Product that we’ve just created, and I click “Save”.
Expected Output 1	No excursion is added to the list.
Expected Output 2	The excursion is added to the list.
Expected Output 3	No excursion is added to the list.
Current Output 1	No excursion is added to the list.
Current Output 2	The excursion is added to the list.
Current Output 3	The excursion with the same id is updated!
Result	The creation process works.
Note	The id should be assigned automatically! The product id requested should automatically be associated with a new product: the user shouldn’t create first a “bland” product and then the realization.

4.2.8 Handling of an Excursion

Handling of an Excurion	
Code	T008
Purpose	Checks if an employee can view, edit or remove an excursion.
Exec. Environment	The excursions page, reachable by clicking on the link “Excursion” in the menubar, from the user home page of the employee logged in.
Input 1	After selecting an excursion in the list, I click the “Delete” button.
Input 2	After selecting an excursion in the list, I click the “View” button.
Input 3	After selecting an excursion in the list, I click the “Edit” button, changing the details of the excursion and saving the modification via the “Save” button.
Expected Output 1	The excursion is deleted from the list.
Expected Output 2	The details about the excursion are showed.
Expected Output 3	The excursion is modified as requested.
Current Output 1	The excursion is deleted from the list.
Current Output 2	The details about the excursion are showed.
Current Output 3	The excursion is modified as requested.
Result	It is possible to show/edit/modify the excursions.
Note	-

4.2.9 Listing of the Flights

Listing of the Flights	
Code	T009
Purpose	Checks if an employee can show the list of already defined flights.
Exec. Environment	The user home page of the employee logged in.
Input	Click on the link “Flight” in the voice “Menu” in the menubar.
Expected Output	The flights page is showed.
Current Output	The flights page is showed.
Result	The list of the available flights is showed.
Note	-

4.2.10 Creation of a new Flight

Creation of a new Flight	
Code	T010
Purpose	Checks if an employee can create a new flight.
Exec. Environment	The flights page, reachable by clicking on the link “Flight” in the menubar, from the user home page of the employee logged in.
Input 1	After clicking on the “Create” button, i put the FlightId 234, the FlightOrigin Milano, the FlightDestination Roma, and the Product with an id already used for another flight, and I click “Save”.
Input 2	After clicking on the “Create” button, i put the FlightId 234, the FlightOrigin Milano, the FlightDestination Roma, and the Product with an id of a product that we’ve just created, and I click “Save”.
Input 3	After clicking on the “Create” button, i put the FlightId 11 (which is already used), the FlightOrigin Milano, the FlightDestination Roma, and the Product with an id of a product that we’ve just created, and I click “Save”.
Expected Output 1	No flight is added to the list.
Expected Output 2	The flight is added to the list.
Expected Output 3	No flight is added to the list.
Current Output 1	No flight is added to the list.
Current Output 2	The flight is added to the list.
Current Output 3	The flight with the same id is updated!
Result	The creation process works.
Note	The id should be assigned automatically. The product id requested should automatically be associated with a new product: the user shouldn’t create first a “bland” product and then the realization.

4.2.11 Handling of a Flight

Handling of a Flight	
Code	T011
Purpose	Checks if an employee can view, edit or remove a flight.
Exec. Environment	The flights page, reachable by clicking on the link “Flight” in the menubar, from the user home page of the employee logged in.
Input 1	After selecting a flight in the list, I click the “Delete” button.
Input 2	After selecting a flight in the list, I click the “View” button.
Input 3	After selecting a flight in the list, I click the “Edit” button, changing the details of the flight and saving the modification via the “Save” button.
Expected Output 1	The flight is deleted from the list.
Expected Output 2	The details about the flight are showed.
Expected Output 3	The flight is modified as requested.
Current Output 1	The flight is deleted from the list.
Current Output 2	The details about the flight are showed.
Current Output 3	The flight is modified as requested.
Result	The flights can be showed/modified/removed.
Note	Even if the date is modified on the database, it keep showing the previous one.

4.2.12 Listing of the Hotels

Listing of the Hotels	
Code	T012
Purpose	Checks if an employee can show the list of already defined hotels.
Exec. Environment	The user home page of the employee logged in.
Input	Click on the link “Hotel” in the voice “Menu” in the menubar.
Expected Output	The hotels page is showed.
Current Output	The hotels page is showed.
Result	The list of the available hotels is showed.
Note	-

4.2.13 Creation of a new Hotel

Creation of a new Hotel	
Code	T013
Purpose	Checks if an employee can create a new hotel.
Exec. Environment	The hotels page, reachable by clicking on the link “Hotel” in the menubar, from the user home page of the employee logged in.
Input 1	After clicking on the “Create” button, i put the HotelId 234, the HotelName Miramare, the Product with an id already used for another hotel, and I click “Save”.
Input 2	After clicking on the “Create” button, i put the HotelId 234, the HotelName Miramare, the Product of a product that we’ve just created, and I click “Save”.
Input 3	After clicking on the “Create” button, i put the HotelId 11 (which is already used), the HotelName Miramare, the Product of a product that we’ve just created, and I click “Save”.
Expected Output 1	No hotel is added to the list.
Expected Output 2	The hotel is added to the list.
Expected Output 3	No hotel is added to the list.
Current Output 1	No hotel is added to the list.
Current Output 2	The hotel is added to the list.
Current Output 2	The hotel with the same id is updated!
Result	The creation process works.
Note	The id should be assigned automatically! The product id requested should automatically be associated with a new product: the user shouldn’t create first a “bland” product and then the realization.

4.2.14 Handling of a Hotel

Handling of a Hotel	
Code	T014
Purpose	Checks if an employee can view, edit or remove a hotel.
Exec. Environment	The hotels page, reachable by clicking on the link “Hotel” in the menubar, from the user home page of the employee logged in.
Input 1	After selecting a hotel in the list, I click the “Delete” button.
Input 2	After selecting a hotel in the list, I click the “View” button.
Input 3	After selecting a hotel in the list, I click the “Edit” button, changing the details of the product and saving the modification via the “Save” button.
Expected Output 1	The hotel is deleted from the list.
Expected Output 2	The details about the hotel are showed.
Expected Output 3	The hotel is modified as requested.
Current Output 1	The hotel is deleted from the list.
Current Output 2	The details about the hotel are showed.
Current Output 3	The hotel is modified as requested.
Result	The hotels can be modified/showed/removed.
Note	Even if the date is modified on the database, it keep showing the previous one.

4.2.15 Listing of the General Packages

Listing of the General Packages	
Code	T015
Purpose	Checks if an employee can show the list of already defined general packages.
Exec. Environment	The user home page of the employee logged in.
Input	Click on the link “Generalpackage” in the voice “Menu” in the menubar.
Expected Output	The general packages page is showed.
Current Output	The general packages page is showed.
Result	The list of the available general packages is showed.
Note	-

4.2.16 Creation of a new General Package

Creation of a new General Package	
Code	T016
Purpose	Checks if an employee can create a new general package.
Exec. Environment	The general package page, reachable by clicking on the link "Generalpackage" in the menubar, from the user home page of the employee logged in.
Input 1	After clicking on the "Create" button, i put the GPackageId 555 (an id that is already used), the price 100, the UserId 0 (that is the id of the employee logged in), and I click "Save".
Input 2	After clicking on the "Create" button, i put the GPackageId 234 (an id that isn't already used), the price 100, the UserId 0 (that is the id of the employee logged in), and I click "Save".
Expected Output 1	No general package is added to the list.
Expected Output 2	The general package is added to the list.
Current Output 1	The previous record is updated!
Current Output 2	The general package is added to the list.
Result	The creation process works if the given id isn't already used.
Note	The package id should be automatically computed, not requested.

4.2.17 Handling of a General Package

Handling of a General Package	
Code	T017
Purpose	Checks if an employee can view, edit or remove a general package.
Exec. Environment	The general package page, reachable by clicking on the link "Generalpackage" in the menubar, from the user home page of the employee logged in.
Input 1	After selecting a general package in the list, I click the "Delete" button.
Input 2	After selecting a general package in the list, I click the "View" button.
Input 3	After selecting a general package in the list, I click the "Edit" button, changing the details of the product and saving the modification via the "Save" button.
Expected Output 1	The general package is deleted from the list.
Expected Output 2	The details about the general package are showed.
Expected Output 3	The general package is modified as requested.
Current Output 1	The general package is deleted from the list.
Current Output 2	The details about the general package are showed.
Current Output 3	The general package is modified as requested.
Result	The general packages can be modified/showed/removed.
Note	Even if the date is modified on the database, it keep showing the previous one.

4.2.18 Listing of the Product Containments

Listing of the Product Containments	
Code	T018
Purpose	Checks if an employee can show the list of already defined product containments.
Exec. Environment	The user home page of the employee logged in.
Input	Click on the link "Productcontainment" in the voice "Menu" in the menubar.
Expected Output	The product containments page is showed.
Current Output	The product containments page is showed.
Result	The list of the available product containments is showed.
Note	-

4.2.19 Creation of a new Product Containment

Creation of a new Product Containment	
Code	T019
Purpose	Checks if an employee can create a new product containment.
Exec. Environment	The product containment page, reachable by clicking on the link “Productcontainment” in the menubar, from the user home page of the employee logged in.
Input 1	After clicking on the “Create” button, i put the PCId 23 (an id that is already used), the ProductId 11 (selecting from the list), the GPackageId 670 (selecting from the list), and I click “Save”.
Input 2	After clicking on the “Create” button, i put the PCId 23 (an id that isn’t already used), the ProductId 11 (selecting from the list), the GPackageId 670 (selecting from the list), and I click “Save”.
Expected Output 1	No product containment is added to the list.
Expected Output 2	The product containment is added to the list.
Current Output 1	The previous record is updated!
Current Output 2	The product containment is added to the list.
Result	The creation process works if the given id isn’t already used.
Note	The id should be automatically computed, not requested.

4.2.20 Handling of a Product Containment

Handling of a Product Containment	
Code	T020
Purpose	Checks if an employee can view, edit or remove a product containment.
Exec. Environment	The product containment page, reachable by clicking on the link “Productcontainment” in the menubar, from the user home page of the employee logged in.
Input 1	After selecting a product containment in the list, I click the “Delete” button.
Input 2	After selecting a product containment in the list, I click the “View” button.
Input 3	After selecting a product containment in the list, I click the “Edit” button, changing the details of the product and saving the modification via the “Save” button.
Expected Output 1	The product containment is deleted from the list.
Expected Output 2	The details about the product containment are showed.
Expected Output 3	The product containment is modified as requested.
Current Output 1	The product containment is deleted from the list.
Current Output 2	The details about the product containment are showed.
Current Output 3	The product containment is modified as requested.
Result	The product containments can be modified/showed/removed.
Note	-

4.2.21 Listing of the Users

Listing of the Users	
Code	T021
Purpose	Checks if an employee can show the list of already defined users.
Exec. Environment	The user home page of the employee logged in.
Input	Click on the link “User” in the voice “Menu” in the menubar.
Expected Output	The users page is showed.
Current Output	The users page is showed.
Result	The list of the available users is showed.
Note	-

4.2.22 Creation of a new User

Creation of a new User	
Code	T022
Purpose	Checks if an employee can create a new user.
Exec. Environment	The user page, reachable by clicking on the link “User” in the menubar, from the user home page of the employee logged in.
Input 1	After clicking on the “Create” button, i put the UserId 0 (an id that is already used), the password password, the Role user, and I click “Save”.
Input 2	After clicking on the “Create” button, i put the UserId 0 (an id that isn’t already used), the password password, the Role user, and I click “Save”.
Expected Output 1	No user is added to the list.
Expected Output 2	The user is added to the list.
Current Output 1	The previous record is updated!
Current Output 2	The user is added to the list.
Result	The creation process works if the given id isn’t already used.
Note	The id should be automatically computed, not requested. Plus, the password should be hashed as requested (MD5 + SHA-512), and not saved in clear.

4.2.23 Handling of a User

Handling of a User	
Code	T023
Purpose	Checks if an employee can view, edit or remove a user.
Exec. Environment	The user page, reachable by clicking on the link “User” in the menubar, from the user home page of the employee logged in.
Input 1	After selecting a user in the list, I click the “Delete” button.
Input 2	After selecting a user in the list, I click the “View” button.
Input 3	After selecting a user in the list, I click the “Edit” button, changing the details of the product and saving the modification via the “Save” button.
Expected Output 1	The user is deleted from the list.
Expected Output 2	The details about the user are showed.
Expected Output 3	The user is modified as requested.
Current Output 1	The user is deleted from the list.
Current Output 2	The details about the user are showed.
Current Output 3	The user is modified as requested.
Result	The users can be modified/showed/removed.
Note	-

4.2.24 Registration of a User

Registration of a User	
Code	T024
Purpose	Checks if visitor can register to the system.
Exec. Environment	The register page, reachable clicking the “register” link from the homepage.
Input	After filling the data, I click the “Register” button.
Expected Output	The is registered.
Current Output	Nothing happens.
Result	It isn’t possible to register at the moment.
Note	-

4.2.25 Listing of the Personal Packages

Listing of the Personal Packages	
Code	T025
Purpose	Checks if an employee/user can show the list of already defined personal packages.
Exec. Environment	The user home page of the employee/user logged in.
Input	Click on the link “Personalpackage” in the voice “Menu” in the menubar.
Expected Output	The personal packages page is showed.
Current Output	The personal packages page is showed.
Result	The list of the available personal packages is showed.
Note	-

4.2.26 Creation of a new Personal Package

Creation of a new Personal Package	
Code	T026
Purpose	Checks if an employee/user can create a new personal package.
Exec. Environment	The personal package page, reachable by clicking on the link “Personalpackage” in the menubar, from the personal package home page of the employee/user logged in.
Input 1	After clicking on the “Create” button, i put the PPackageId 66 (an id that is already used), the GPackageId 555 (selecting from the list), the UserId 123 (selecting from the list), and I click “Save”.
Input 1	After clicking on the “Create” button, i put the PPackageId 1435 (an id that isn’t already used), the GPackageId 555 (selecting from the list), the UserId 123 (selecting from the list), and I click “Save”.
Expected Output 1	No personal package is added to the list.
Expected Output 2	The personal package is added to the list.
Current Output 1	The previous record is updated!
Current Output 2	The personal package is added to the list.
Result	The creation process works if the given id isn’t already used.
Note	The id should be automatically computed, not requested.

4.2.27 Handling of a Personal Package

Handling of a Personal Package	
Code	T027
Purpose	Checks if an employee/user can view, edit or remove a personal package.
Exec. Environment	The personal package page, reachable by clicking on the link “Personal-package” in the menubar, from the personal package home page of the employee/user logged in.
Input 1	After selecting a personal package in the list, I click the “Delete” button.
Input 2	After selecting a personal package in the list, I click the “View” button.
Input 3	After selecting a personal package in the list, I click the “Edit” button, changing the details of the product and saving the modification via the “Save” button.
Expected Output 1	The personal package is deleted from the list.
Expected Output 2	The details about the personal package are showed.
Expected Output 3	The personal package is modified as requested.
Current Output 1	The personal package is deleted from the list.
Current Output 2	The details about the personal package are showed.
Current Output 3	The personal package is modified as requested.
Result	The personal packages can be modified/showed/removed.
Note	-

4.2.28 Listing of the Modifications

Listing of the Modifications	
Code	T028
Purpose	Checks if an employee/user can show the list of already defined modifications.
Exec. Environment	The user home page of the employee/user logged in.
Input	Click on the link “Modification” in the voice “Menu” in the menubar.
Expected Output	The modifications page is showed.
Current Output	The modifications page is showed.
Result	The list of the available modifications is showed.
Note	-

4.2.29 Creation of a new Modification

Creation of a new Modification	
Code	T029
Purpose	Checks if an employee/user can create a new modification.
Exec. Environment	The modification page, reachable by clicking on the link “Modification” in the menubar, from the modification home page of the employee/user logged in.
Input 1	After clicking on the “Create” button, i put the Mid 77 (an id that is already used), the ProductId 11 (selecting from the list), the PPackageId 66 (selecting from the list), and I click “Save”.
Input 1	After clicking on the “Create” button, i put the Mid 234 (an id that isn’t already used), the ProductId 11 (selecting from the list), the PPackageId 66 (selecting from the list), and I click “Save”.
Expected Output 1	No modification is added to the list.
Expected Output 2	The modification is added to the list.
Current Output 1	The previous record is updated!
Current Output 2	The modification is added to the list.
Result	The creation process works if the given id isn’t already used.
Note	The id should be automatically computed, not requested.

4.2.30 Handling of a Modification

Handling of a Modification	
Code	T030
Purpose	Checks if an employee/user can view, edit or remove a modification.
Exec. Environment	The modification page, reachable by clicking on the link “Modification” in the menubar, from the modification home page of the employee/user logged in.
Input 1	After selecting a modification in the list, I click the “Delete” button.
Input 2	After selecting a modification in the list, I click the “View” button.
Input 3	After selecting a modification in the list, I click the “Edit” button, changing the details of the product and saving the modification via the “Save” button.
Expected Output 1	The modification is deleted from the list.
Expected Output 2	The details about the modification are showed.
Expected Output 3	The modification is modified as requested.
Current Output 1	The modification is deleted from the list.
Current Output 2	The details about the modification are showed.
Current Output 3	The modification is modified as requested.
Result	The modifications can be modified/showed/removed.
Note	-

4.2.31 Adding of a Friend to a Personal Package

Adding of a Friend to a Personal Package	
Code	T031
Purpose	Checks if an employee/user can add a friend to a personal package.
Exec. Environment	The personal package page, reachable by clicking on the link "Personal-package" in the menubar, from the personal package home page of the employee/user logged in.
Input 1	After clicking on the "Create" button, I put the PPackageId 66 (an id that is already used), the GPackageId 555 (selecting from the list), the UserId of my friend, 123 (selecting from the list), that should already be registered, and I click "Save".
Input 1	After clicking on the "Create" button, i put the PPackageId 1435 (an id that isn't already used), the GPackageId 555 (selecting from the list), the UserId of my friend, 123 (selecting from the list), that should already be registered, and I click "Save".
Expected Output 1	No friend is added to the personal package.
Expected Output 2	The friend is added to the personal package.
Current Output 1	The original record is updated, making the friend the new owner of the personal package!
Current Output 2	The friend is added to the personal package.
Result	The adding of a friend works if the given id isn't already used.
Note	The id should be automatically computed, not requested. There's no difference with the adding of a new personal package.

4.3 Conclusions

Other than the suggestions that I've written in the tests above, I'd suggest to *hide the links* to the elements that the customers aren't allowed to see, because the error page isn't a nice solution. Not only this: the customer is even allowed to modify excursions, flights, and so on (that is everything he/she can see without the 403 error page).

Chapter 5

Appendix

5.1 Hours spent for each part of the project

	Riccardo B. Desantis
RASD	15 h
DD	32 h
Implementation	120 h
Testing	20 h