Politecnico di Milano
M. Sc. Course in IT Engineering

Software Engineering 2 – Internal Project
Prof. Elisabetta di Nitto – a.y. 2013/2014

Testing report

VERSION 1

by
Edoardo Mondoni (816283)

Table of contents

Index of test cases

Preamble

This is the Testing Report for the Software Engineering 2 2013 internal project, named *TravelDream*. It contains a summary of the testing conducted on the *TravelDream Agent* application, which was developed during the Implementation phase of the aforementioned project. Such operations have been carried out manually (no automated tool has been used); the tests were run against the functional requirements stated in Version 4 of the Requirements Analysis and Specification Document and the design specifications illustrated in Version 2 of the Design Document.

The remainder of the document is organized as follows:

- **Section 1** contains a report for each one of the 21 test cases conducted on the system;
- **Section 2** indicates what conclusions can be drawn from the results of the test cases illustrated in the previous section and contains a brief report of the application's known bugs.

1 Test cases

The tests listed in the following subsections have mainly been elaborated based on the contents of the Requirements Analysis and Specification Document. In particular, it was deemed reasonable to verify whether the system is actually capable of performing each one of the functions enumerated in Subsection 2.2 of said document, thus ensuring the achievement of all goals in Subsection 1.3. In order to do so, the decision was made to reproduce the use cases in Subsection 3.2 and to note any difference, failure or malfunctioning that might arise during such tests.

The test cases have been developed according to the following structure:

- **Is the flow of events consistent with what is stated in the use case?** The application is tested to check if there are any differences in the flow of events and, if any, whether they influence the satisfaction of the reference goals and functions.
- **Is the exit condition verified if the flow of events is respected and no anomalous input is provided?** The flow of events is reproduced with no voluntary anomaly to test the result of the operation for correctness.
- **Are there any unlisted exceptions?** The flow of events is reproduced again multiple times with different inputs and/or in different conditions to ascertain the completeness of the exception list in the use case.
- **Is the system's reaction to the listed exceptions consistent with what is stated in the use case?** Reports whether the application actually handles the forecast exceptions the way it is supposed to.
- **Are all reference functions satisfied?** Summarizes the findings of the previous questions indicating whether the discrepancies between the use case and the actual program (if any) cause the reference functions not to be satisfied. An affirmative answer implies that the system actually passes the test, while a negative answer means that the system cannot perform one or more functions it is expected to deliver.

1.1 Common functions

This subsection reports on the testing conducted on functions regarding unregistered users or all registered users (independently of their category).

[T1] Registration of new customers					
Reference use case	[U1]	**Reference goals**	[G3] [G4] [G5] [G6]	**Reference functions**	[F1]
Is the flow of events consistent with what is stated in the use case?		✓ **Yes.** No difference has been noted.			

[T1] Registration of new customers	
Is the exit condition verified if the flow of events is respected and no anomalous input is provided?	✓ **Yes.** After filling in the fields with random (but correct and consistent) data, the application allows access to the system with the typed credentials.
Are there any unlisted exceptions?	× **No.** No other exception has been found.
Is the system's reaction to the listed exceptions consistent with what is stated in the use case?	✓ **Yes.** Leaving one or more fields blank, providing inconsistent information (badly-formatted e-mail address…) and choosing an already-taken username all result in non-blocking warning messages barring the finalization of the process.
Are all reference functions satisfied?	✓ **Yes.** An unregistered user is able to sign up as a customer.

TEST CASE 1: REGISTRATION OF NEW CUSTOMERS

[T2] Login to the system					
Reference use case	[U2]	**Reference goals**	[G1] [G2] [G4] [G5] [G6]	**Reference functions**	[F2]
Is the flow of events consistent with what is stated in the use case?			↔ **Somehow.** It was not possible to fully implement the flow of events as it was conceived because of `j_security_check`'s implementation, which apparently requires a request for a reserved page to be submitted before triggering the login process. The user is therefore required to state their category by clicking on one of the three buttons in the homepage before getting the chance to log in to the system.		
Is the exit condition verified if the flow of events is respected and no anomalous input is provided?			✓ **Yes.** Keeping in mind that the flow of events requires an additional step, any registered user can log in with the credentials they possess.		
Are there any unlisted exceptions?			✓ **Yes.** Given the modified flow of events, a user could click on a button that doesn't correspond to their category. In that case, the application server raises a 403 Forbidden error. The same happens if the user correctly logs in to the application, navigates to the site's home page (which is only possible by manually entering its URL in the address bar, since the user interface provides no way to do so for logged users) and then tries to access another category's reserved pages.		
Is the system's reaction to the listed exceptions consistent with what is stated in the use case?			✓ **Yes.** Wrong input results in an error page informing the user of their mistake. The login procedure can be immediately repeated.		
Are all reference functions satisfied?			✓ **Yes.** At the end of the day, the login process does what it is supposed to: it allows access to all registered users, forbids unregistered ones to take advantage of the application and ensures that every user is only admitted to the functions they're granted use of.		

TEST CASE 2: LOGIN TO THE SYSTEM

[T3] Logout from the system					
Reference use case	[U3]	**Reference goals**	None	**Reference functions**	[F3]
Is the flow of events consistent with what is stated in the use case?	✓ **Yes.** No difference has been noted.				
Is the exit condition verified if the flow of events is respected and no anomalous input is provided?	✓ **Yes.** The flow of events can be initiated by clicking on a button which is absent in case the entry condition (the user is logged in) is not respected. Once terminated the process, the user is automatically returned to the *home* page, which coincided with the login page when the use cases were drafted.				
Are there any unlisted exceptions?	× **No.** No other exception has been found.				
Is the system's reaction to the listed exceptions consistent with what is stated in the use case?	No exception was listed in the use case.				
Are all reference functions satisfied?	✓ **Yes.** A user can gracefully terminate their session by clicking on a button they can reach from anywhere in the site.				

TEST CASE 3: LOGOUT FROM THE SYSTEM

[T4] Modification of one's own personal details					
Reference use case	[U4]	**Reference goals**	None	**Reference functions**	[F4]
Is the flow of events consistent with what is stated in the use case?	✓ **Yes.** No difference has been noted.				
Is the exit condition verified if the flow of events is respected and no anomalous input is provided?	✓ **Yes.** The user's sensitive data is correctly updated in the application's database.				
Are there any unlisted exceptions?	× **No.** No other exception has been found.				
Is the system's reaction to the listed exceptions consistent with what is stated in the use case?	✓ **Yes.** Entering inconsistent information results in overlay messages informing the user of their mistake and inviting them to correct it, highlighting the concerned fields all the while. The same happens when voiding one or more mandatory fields.				
Are all reference functions satisfied?	✓ **Yes.** A user is free to modify their personal details, provided that they do not violate the validation rules built into the application.				

TEST CASE 4: MODIFICATION OF ONE'S OWN PERSONAL DETAILS

[T5] Modification of one's own password					
Reference use case	[U5]	**Reference goals**	None	**Reference functions**	[F5]
Is the flow of events consistent with what is stated in the use case?	× **No.** The implemented procedure skips the request for the current password. Everything else remains the same.				

[T5] Modification of one's own password	
Is the exit condition verified if the flow of events is respected and no anomalous input is provided?	✓ **Yes.** The new password is the only one granting access to the site in combination with the user's username. The old password is no more valid.
Are there any unlisted exceptions?	× **No.** No other exception has been found.
Is the system's reaction to the listed exceptions consistent with what is stated in the use case?	↔ **Somehow.** Actually, it was possible to experiment *fewer* exceptions than those listed (since the current password is not required). Typing two different passwords results in a warning – and does not allow to complete the operation – while the third exception simply never happens, since no constraint has been enforced on passwords.
Are all reference functions satisfied?	✓ **Yes.** All differences that have been found are procedural: the password changing process is actually fully functional and achieves its goals.

TEST CASE 5: MODIFICATION OF ONE'S OWN PASSWORD

1.2 Administrator-related functions

This subsection lists the test cases concerning system functions related to administrators.

[T6] Creation of a new user account					
Reference use case	[U6]	**Reference goals**	[G1] [G2]	**Reference functions**	[F23]
Is the flow of events consistent with what is stated in the use case?		✓ **Yes.** No difference has been found.			
Is the exit condition verified if the flow of events is respected and no anomalous input is provided?		✓ **Yes.** A new user is created and the username-password credentials are enabled for logging in to the reserved area of the user category specified while creating the account.			
Are there any unlisted exceptions?		× **No.** No other exception has been found.			
Is the system's reaction to the listed exceptions consistent with what is stated in the use case?		✓ **Yes.** Failing to fill one or more mandatory fields results in a warning (and in the impossibility to proceed until a value has been specified). The same happens when entering inconsistent information.			
Are all reference functions satisfied?		✓ **Yes.** Administrators can create employee and administrator accounts with the credentials they enter, associated to the personal details they type.			

TEST CASE 6: CREATION OF A NEW USER ACCOUNT

[T7] Modification of an existing user account					
Reference use case	[U7]	**Reference goals**	None	**Reference functions**	[F24] [F25]
Is the flow of events consistent with what is stated in the use case?		✓ **Yes.** Besides the fact that the administrator must not click on the row, but on a button in the row, the flow of events is fully respected.			

[T7] Modification of an existing user account	
Is the exit condition verified if the flow of events is respected and no anomalous input is provided?	✓ **Yes.** The selected user's account is modified as indicated by the administrator.
Are there any unlisted exceptions?	✓ **Yes.** The administrator cannot transform a customer's profile into a corporate (employee or administrator) one. In the same way, a corporate account cannot be transformed into a customer account: otherwise, administrators would have *de facto* been able to create customer accounts (which is forbidden as a design choice).
Is the system's reaction to the listed exceptions consistent with what is stated in the use case?	✓ **Yes.** Voiding one or more mandatory fields results in a warning (and in the impossibility to proceed until a value has been specified). The same happens when entering inconsistent information.
Are all reference functions satisfied?	✓ **Yes.** Administrators can edit any account's personal details or password; category switching has been limited to conform to a principle established during the design phase.

TEST CASE 7: MODIFICATION OF AN EXISTING USER ACCOUNT

[T8] Deletion of a user account					
Reference use case	[U8]	**Reference goals**	None	**Reference functions**	[F26]
Is the flow of events consistent with what is stated in the use case?		✓ **Yes.** No difference has been noted.			
Is the exit condition verified if the flow of events is respected and no anomalous input is provided?		✓ **Yes.** The selected users are correctly deleted after an affirmative answer has been passed to the confirmation dialog that appears on screen.			
Are there any unlisted exceptions?		× **No.** No other exception has been found.			
Is the system's reaction to the listed exceptions consistent with what is stated in the use case?		↔ **Somehow.** Selecting no account results in an overlay message telling the administrator they selected no account, but only after the operation has been confirmed. Selecting the administrator's own account among those to delete results in all accounts being deleted but the administrator's one; when this happens, an overlay message informs the administrator that the deletion of their account has been prevented.			
Are all reference functions satisfied?		✓ **Yes.** An administrator can delete any user account in the system, except for their own.			

TEST CASE 8: DELETION OF A USER ACCOUNT

1.3 Employee-related functions

This subsection contains all test case reports on the functions assigned to employees.

[T9] Creation of a basic product					
Reference use case	[U10]	**Reference goals**	[G1] [G2] [G5] [G6]	**Reference functions**	[F16] [F17]
Is the flow of events consistent with what is stated in the use case?	✓ **Yes.** No difference has been noted.				
Is the exit condition verified if the flow of events is respected and no anomalous input is provided?	✓ **Yes.** The employee is able to create all types of basic products in the way specified in the use case.				
Are there any unlisted exceptions?	× **No.** No other exception has been found.				
Is the system's reaction to the listed exceptions consistent with what is stated in the use case?	✓ **Yes.** Failing to fill one or more mandatory fields results in a warning (and in the impossibility to proceed until a value has been specified). The same happens when entering inconsistent information.				
Are all reference functions satisfied?	✓ **Yes.** An employee can create excursions, flights and hotels, saving them to the application's database and immediately finding them in the basic products list.				

TEST CASE 9: CREATION OF A BASIC PRODUCT

[T10] Modification of a basic product					
Reference use case	[U11]	**Reference goals**	[G1] [G2] [G5] [G6]	**Reference functions**	[F16] [F18]
Is the flow of events consistent with what is stated in the use case?	✓ **Yes.** Besides the fact that the administrator must not click on the row, but on a button in the row, the flow of events is fully respected.				
Is the exit condition verified if the flow of events is respected and no anomalous input is provided?	✓ **Yes.** The selected basic product's entry in the application's database is modified according to what's been entered by the employee.				
Are there any unlisted exceptions?	× **No.** No other exception has been found.				
Is the system's reaction to the listed exceptions consistent with what is stated in the use case?	✓ **Yes.** Voiding one or more mandatory fields results in a warning (and in the impossibility to proceed until a value has been specified). The same happens when entering inconsistent information.				
Are all reference functions satisfied?	✓ **Yes.** An employee is fully entitled to edit a basic product, provided that the modifications respect the validation constraints.				

TEST CASE 10: MODIFICATION OF A BASIC PRODUCT

[T11] Deletion of basic products					
Reference use case	[U12]	**Reference goals**	[G1]	**Reference functions**	[F16] [F19]
Is the flow of events consistent with what is stated in the use case?	↔ **Somehow.** There actually is no "basic products list", but three separate lists, each one for a different kind of basic product. Deleting products belonging to more than one category thus requires repeating the process for each category.				

[T11] Deletion of basic products	
Is the exit condition verified if the flow of events is respected and no anomalous input is provided?	✓ **Yes.** The selected basic products are correctly deleted and are no more available for election as package components; all packages containing the selected basic products are deleted from the system as well.
Are there any unlisted exceptions?	× **No.** No other exception has been found.
Is the system's reaction to the listed exceptions consistent with what is stated in the use case?	↔ **Somehow.** Selecting no basic product before clicking on the button results in a warning message, but only after the employee confirms the operation.
Are all reference functions satisfied?	✓ **Yes.** An employee is completely able to delete any basic product in the database.

TEST CASE 11: DELETION OF BASIC PRODUCTS

[T12] Creation of a travel package					
Reference use case	[U13]	**Reference goals**	[G2] [G4] [G5] [G6]	**Reference functions**	[F20]
Is the flow of events consistent with what is stated in the use case?	✓ **Yes.** There are three different product lists (one for each basic product type) instead of a single one, but the flow of events remains valid.				
Is the exit condition verified if the flow of events is respected and no anomalous input is provided?	✓ **Yes.** A new travel package featuring the selected components and details is immediately available for purchase and customization on the part of a customer.				
Are there any unlisted exceptions?	× **No.** No other exception has been found.				
Is the system's reaction to the listed exceptions consistent with what is stated in the use case?	✓ **Yes.** The *Create package* button is actually disabled unless at least one component is part of the package, so the employee can't even attempt to create an empty package, but the listed exceptions are correct.				
Are all reference functions satisfied?	✓ **Yes.** A travel package can be created starting from already-existing basic products, and immediately appears in the packages list.				

TEST CASE 12: CREATION OF A TRAVEL PACKAGE

[T13] Modification of a travel package					
Reference use case	[U14]	**Reference goals**	[G2] [G4] [G5] [G6]	**Reference functions**	[F21]
Is the flow of events consistent with what is stated in the use case?	✓ **Yes.** Apart from the employee having to click on a button in the row rather than on the row itself, no differences have been noted.				
Is the exit condition verified if the flow of events is respected and no anomalous input is provided?	✓ **Yes.** The modifications are correctly stored and the old version of the package is nowhere to be found.				
Are there any unlisted exceptions?	× **No.** No other exception has been found.				

[T13] Modification of a travel package	
Is the system's reaction to the listed exceptions consistent with what is stated in the use case?	✓ **Yes.** The *Submit changes* button is actually disabled if the employee removes all components from the package, so the employee can't even attempt to save such a change, but the listed exceptions are correct.
Are all reference functions satisfied?	✓ **Yes.** An employee is fully capable of modifying a pre-defined travel package in any way they deem appropriate, provided that the new package respects the validation constraints.

TEST CASE 13: MODIFICATION OF A TRAVEL PACKAGE

[T14] Deletion of travel packages					
Reference use case	[U15]	**Reference goals**	[G2]	**Reference functions**	[F22]
Is the flow of events consistent with what is stated in the use case?	✓ **Yes.** No difference has been noted.				
Is the exit condition verified if the flow of events is respected and no anomalous input is provided?	✓ **Yes.** The selected packages are no longer available for customers to buy or customize, and do not appear in the employee's packages list anymore.				
Are there any unlisted exceptions?	× **No.** No other exception has been found.				
Is the system's reaction to the listed exceptions consistent with what is stated in the use case?	↔ **Somehow.** Selecting no package before clicking on the button results in a warning message, but only after the employee confirms the operation.				
Are all reference functions satisfied?	✓ **Yes.** An employee is fully capable of deleting any travel packages they want to remove.				

TEST CASE 14: DELETION OF TRAVEL PACKAGES

1.4 Customer-related functions

This subsection reports on the testing conducted on the application's customer-related functions.

[T15] Browsing the package catalogue					
Reference use case	[U16]	**Reference goals**	[G4] [G5] [G6]	**Reference functions**	[F6] [F7] [F8]
Is the flow of events consistent with what is stated in the use case?	× **No.** The tools for narrowing the package catalogue down are nowhere to be found: the customer has to manually browse through the catalogue to find a specific package.				
Is the exit condition verified if the flow of events is respected and no anomalous input is provided?	× **No.** Since the customer has no way of entering their search criteria, the exit condition cannot be met.				
Are there any unlisted exceptions?	× **No.** No other exception has been found.				

[T15] Browsing the package catalogue	
Is the system's reaction to the listed exceptions consistent with what is stated in the use case?	× **No.** The listed exceptions are based on the hypothesis that the customer can input their criteria to filter the list; the system does not offer this functionality, so it is impossible to evaluate its behaviour in such exceptional conditions.
Are all reference functions satisfied?	× **No.** The customer cannot browse through the package catalogue.

TEST CASE 15: BROWSING THE PACKAGE CATALOGUE

[T16] Visualization of a travel package's details					
Reference use case	[U17]	**Reference goals**	[G5] [G6]	**Reference functions**	[F9] [F10]
Is the flow of events consistent with what is stated in the use case?			✓ **Yes.** Apart from the customer having to click on a button in the row rather than on the row itself, no differences have been noted.		
Is the exit condition verified if the flow of events is respected and no anomalous input is provided?			✓ **Yes.** The customer is redirected to a page where the components of the package are clearly listed and classified by type, along with the package's name, price and description.		
Are there any unlisted exceptions?			× **No.** No other exception has been found.		
Is the system's reaction to the listed exceptions consistent with what is stated in the use case?			No exception was listed in the use case.		
Are all reference functions satisfied?			✓ **Yes.** A customer can see the details of any travel package they find in the catalogue.		

TEST CASE 16: VISUALIZATION OF A TRAVEL PACKAGE'S DETAILS

[T17] Customization of a pre-defined travel package					
Reference use case	[U18]	**Reference goals**	[G5] [G6]	**Reference functions**	[F11] [F12]
Is the flow of events consistent with what is stated in the use case?			✓ **Yes.** No difference has been noted.		
Is the exit condition verified if the flow of events is respected and no anomalous input is provided?			✓ **Yes.** A customized package is created based on what the customer entered in the customization dialog. Furthermore, the system redirects the customer to the newly-created bundle's details page upon saving.		
Are there any unlisted exceptions?			✓ **Yes.** The customer must also enter a non-empty name for the bundle, otherwise the system does not allow them to proceed (no message is issued, but the correspondent field is highlighted).		
Is the system's reaction to the listed exceptions consistent with what is stated in the use case?			✓ **Yes.** The *Save customized package* button is actually disabled if the customer removes all components from the package, so the customer can't even attempt to save such a bundle, but the listed exceptions are correct.		

[T17] Customization of a pre-defined travel package	
Are all reference functions satisfied?	✓ **Yes.** A customer can modify any package in the catalogue in any way they want to, provided they respect the non-emptiness constraint.

TEST CASE 17: CUSTOMIZATION OF A PRE-DEFINED TRAVEL PACKAGE

[T18] Visualization of the customized packages list					
Reference use case	[U19]	**Reference goals**	[G5] [G6]	**Reference functions**	[F12] [F13] [F14]
Is the flow of events consistent with what is stated in the use case?	✓ **Yes.** No difference has been noted.				
Is the exit condition verified if the flow of events is respected and no anomalous input is provided?	✓ **Yes.** The customer is redirected to a page containing a list of all customized packages (except the ones that have been subsequently deleted).				
Are there any unlisted exceptions?	× **No.** No other exception has been found.				
Is the system's reaction to the listed exceptions consistent with what is stated in the use case?	No exception was listed in the use case.				
Are all reference functions satisfied?	✓ **Yes.** A customer can access their customized packages list from any page in the website.				

TEST CASE 18: VISUALIZATION OF THE CUSTOMIZED PACKAGES LIST

[T19] Modification of a customized package					
Reference use case	[U20]	**Reference goals**	[G5] [G6]	**Reference functions**	[F13]
Is the flow of events consistent with what is stated in the use case?	↔ **Somehow.** The *Edit it!* button is actually located in the package details page, which the user can access by clicking on the appropriate button in the customized packages list.				
Is the exit condition verified if the flow of events is respected and no anomalous input is provided?	✓ **Yes.** The changes are stored in the application's database and the old version of the package is nowhere to be found.				
Are there any unlisted exceptions?	✓ **Yes.** The customer must also enter a non-empty name for the bundle, otherwise the system does not allow them to proceed (no message is issued, but the correspondent field is highlighted).				
Is the system's reaction to the listed exceptions consistent with what is stated in the use case?	✓ **Yes.** The *Submit changes* button is actually disabled if the customer removes all components from the package, so the customer can't even attempt to save such a bundle, but the listed exceptions are correct.				
Are all reference functions satisfied?	✓ **Yes.** The customer is fully entitled to modify a customized package in any way they wish, provided that the package still contains at least one component.				

TEST CASE 19: MODIFICATION OF A CUSTOMIZED PACKAGE

[T20] Deletion of customized packages					
Reference use case	[U21]	**Reference goals**	None	**Reference functions**	[F14]
Is the flow of events consistent with what is stated in the use case?	✓ **Yes.** No difference has been noted.				
Is the exit condition verified if the flow of events is respected and no anomalous input is provided?	✓ **Yes.** The selected packages are deleted from the system and are no more available for purchase or modification.				
Are there any unlisted exceptions?	✗ **No.** No other exception has been found.				
Is the system's reaction to the listed exceptions consistent with what is stated in the use case?	↔ **Somehow.** Selecting no package before clicking on the button results in a warning message, but only after the customer confirms the operation.				
Are all reference functions satisfied?	✓ **Yes.** It is totally possible for a customer to delete their own customized packages.				

TEST CASE 20: DELETION OF CUSTOMIZED PACKAGES

[T21] Purchase of a travel package					
Reference use case	[U22]	**Reference goals**	[G6]	**Reference functions**	[F15]
Is the flow of events consistent with what is stated in the use case?	✓ **Yes.** No difference has been noted.				
Is the exit condition verified if the flow of events is respected and no anomalous input is provided?	✓ **Yes.** The package appears in the customer's purchases list with the selected date assignments.				
Are there any unlisted exceptions?	✓ **Yes.** The calendars for the selection of the dates do not save the date if the next/previous month arrows are used.				
Is the system's reaction to the listed exceptions consistent with what is stated in the use case?	✓ **Yes.** Missing or inconsistent date assignments do not allow the customer to proceed to the billing screen. Billing data cannot be checked for validity, so a simple random boolean switch has been implemented that simulates errors during the billing process.				
Are all reference functions satisfied?	✓ **Yes.** A customer can purchase any package appearing in the package catalogue or in their customized packages list.				

TEST CASE 21: PURCHASE OF A TRAVEL PACKAGE

2 Conclusions

2.1 Test results

The purpose of the testing phase is to verify whether the system can perform all functions it has been developed for. It is easy to check that 20 out of 21 test cases terminated with an affirmative answer, and are thus to be considered passed.

On the other hand, test case [T15] failed: the reason for this is the lack of a way for the user to input their constraints. This does not, however, prevent them from exploring the catalogue as it is. As a consequence, function [F6] is to be considered satisfied; functions [F7] and [F8] can in no way be considered satisfied, since the customer cannot restrict the list of packages that's presented to them, either based on a keyword or on any other criterion.

Available functions	Unavailable functions
All common functions: [F1] to [F5] All administrator-related functions: [F23] to [F26] All employee-related functions: [F16] to [F22] Customer-related functions: [F6], [F9] to [F15]	Customer-related functions: [F7] and [F8]

2.2 Known bugs

The calendars in the date selection page might not work correctly if the next/previous month arrows are used. This misbehaviour is to be ascribed to Primefaces, whose `<p:calendar>` component does not interact as expected when used in combination with `<p:dataTable>`'s in-cell editing. For more information, see `https://code.google.com/p/primefaces/issues/detail?id=5230`: the bug report was filed about a year ago and quickly dismissed as "Can't replicate", but many are still reporting the bug recently, judging from the comments. The fact that Primefaces' developer Cagatay Civici could not replicate the bug might hint that this is browser-dependent.

A function that is entirely missing from the program is the possibility to upload photos and associate them to excursions, hotels and travel packages. The reason for this is a lack of time: I preferred to strip photo-related functionalities off the application rather than risking not to deliver the User and Install Guides in time.

Miscellanea

Changelog

Version 1 (February 3rd, 2014)

First release of the Testing report.

Effort

A total of *5 hours* were spent on the production of this document. Bear in mind that the actual testing was carried out during the implementation and that the corresponding effort is included in the implementation phase's *Effort report*.