MeteoCal
Acceptance Testing Document

Simone Graziussi

8th February 2015

Contents

1 Introduction

This document describes the acceptance test cases defined for Stefano Di Carlo's *MeteoCal* project.

It contains some notes, from the customer's point of view, about the documentation and code quality, the fulfilment of the project assignment and some test cases to assess if the application works as excepted.

2 Documentation and code quality

The provided documentation (RASD and DD) is complete and in general clear to understand. However, in the documents there are a few ambiguous or contradictory statements. For example, in the Alloy model described in the RASD there's the constraint that two users cannot have the same mobile phone number: this requirement is not clearly specified in any other part of the documentation so it's unclear if it's a desired behaviour or not. Another example is that in the RASD it is stated that "The recording operation *requires* the entry of personal data among these are: Name, date of birth, home address, phone number", while in the DD we can read "All field (except for address and phone on table user) are not nullable": it's unclear if address and phone are required or not.

The described requirements and goals completely fulfil the initial project assignment: all functions are implemented as described. One exception could be the part saying "Only the organizer will be able to update or delete the event": in the implementation accepting the invitation "clones" the event in the current user calendar (and from that moment the two events are completely independent), so the invited user can update or delete the event (only the one in his/her calendar). Being the requirement ambiguous, it may still be fulfilled (the invited user cannot alter the "original" event), but maybe it should have been mentioned as an assumption in the RASD.

Code quality is poor: there are very few (or none at all) JavaDoc annotations (or simple comments) and this makes the code hard to understand.

3 Test cases: registration and log-in

Goal	User registration
Environment	The "Sign Up" page
Input	User-data: first name, last name, e-mail, phone number, etc.
Expected output	The system saves the new user in the database with all correct data
Obtained output	The procedure works correctly, the user is stored in the database
Possible errors	1. some fields are mandatory and therefore the system should notify the user if they are not set 2. the user email (user-name) must be unique 3. the user phone number must be unique. *Note: like stated in the previous section this requirement is unclear, here I assumed it was a desired behaviour*
Displayed errors	1. the errors are correctly shown. Errors are shown also for address and phone number and, as mentioned in the previous section, it is unclear if this is a desired behaviour or not. 2. the constraint correctly prevents the registration but no message is shown to the user (the only feedback is a MySQL exception printed in the server console saying "Violated primary key constraint") 3. this constraint is *not* verified by the system, two users can have the same phone number

Goal	The user logs into the system
Environment	The "log-in page"
Input	Valid user-name (email) and password
Expected output	The system authenticates the user
Obtained output	The function works properly
Possible errors	1. the user is notified if wrong user-name and/or password are inserted
Displayed errors	1. the error is correctly displayed

4 Test cases: calendar browsing

Goal	Browse the calendar (change displayed month) with weather forecasts and event lists
Environment	The "calendar page"
Expected output	The calendar structure is correctly displayed, with the list of events for each day. Weather forecasts, if available, are shown for the next few days
Obtained output	The function is correctly displayed
Possible errors	None

5 Test cases: event management

Goal	Create a new event or update an existing one
Environment	The "event form"
Input	The event data: name, start, end, etc.
Expected output	The system stores the event in the database and links it the current user
Obtained output	The function works properly
Possible errors	1. all fields are mandatory (in the DD it is stated that "All field (except for address and phone on table user) are not nullable") so the system should notify the user if they are not set 2. the start date must be before the end date. *Note: this requirement is not stated in the RASD. However, in my opinion, it's an implicit (common sense) constraint given by the assignment itself so it should be checked*
Displayed errors	1. this constraint is *not* checked by the system (it allows to create an event with all null parameters). Moreover, when the date is null, the event breaks the visualization in the calendar for all events (NullPointerException) 2. this constraint is *not* checked by the system

Goal	Delete an event
Environment	The "calendar page"
Input	Click on "Delete" button linked to a specific event
Expected output	The system deletes the event from the user calendar
Obtained output	The function works properly
Possible errors	None

6 Test cases: notifications and invitations

Goal	Invite one or more users to a new event
Environment	The "event form"
Input	The selected event and one or more users to invite
Expected output	The system sends the invitations, linked to the correct event, to the selected users
Obtained output	The function works properly
Possible errors	1. all events must have at least one invitation (RASD: "If Someone creates an event, certainly invite other people") so the system should notify the user if no users are selected
Displayed errors	1. this constraint is *not* checked by the system

Goal	Read notifications and invitations
Environment	The "notification page"
Expected output	The system shows the list of unread notifications for the current user
Obtained output	The functions works correctly
Possible errors	None

Goal	Accept or decline an invitation
Environment	The "notification page"
Input	The user clicks on "Accept" or "Decline"
Expected output	If the invitation is accepted the system adds the event to the current user calendar. In both cases the invitation is deleted.
Obtained output	The function works properly
Possible errors	None

Goal	The system sends bad weather alerts
Environment	The "notification page"
Expected output	At the user log-in, the system sends a notification in case of outdoor event and bad weather forecast. The system checks all events that start tomorrow. The notification is sent to all users that have the event in their calendar
Obtained output	This function doesn't work properly. Some notifications are not sent even if the weather forecast is bad. It seems that the problem is that the system always checks the weather forecast for Milan and not the one for the actual event location: for this reason alerts are sent only when the weather is bad in Milan. When this happens, however, the notifications are correctly sent and displayed.
Possible errors	None

7 Conclusion

The general impression on the system is mostly positive. The provided documentation (RASD, DD, Installation and User guides) are in general clear. The implementation works in most situations, except some minor bugs in special situations, and it fulfils the original project assignment.

There could have been a bit more attention on code quality (in particular a complete JavaDoc) and on the documentation (RASD and DD) inconsistencies, though.