

ACCEPTANCE TEST

Tested Code: agostini-cantini-dignani-se2 [\[link\]](#)

Test Group: Angelo, Ceriani, Di Fabrizio. Reference email: v.ceriani92@gmail.com

Introduction

The test cases in the testing document provided in the folder “Deliveries” have been all successfully executed.

Here there are the other test cases that we have identified.

Every time we detected an inconsistent behavior that could compromise the database, we did reset the database so that the following tests wouldn't be affected.

Colors Legend:

	Successful
	Warning
	Error

Test Cases

Login redirect	
Goal	verify the redirect
Enviroment	the user has been already logged in
Input	the url “localhost:8080/MeteoCal”
Expected output	Redirect to the calendar page
Obtained output	Login page
Note	Even if this is not a real bug, it may interfere with the user experience.

Delete event with participants	
Goal	The event should be deleted from every participants calendars
Enviroment	The owner of the event is logged in and in the edit page of the event.
Input	The owner clicks “remove participation”
Expected output	The event is deleted from every participants’ calendars.
Obtained output	The event is removed only from the owner’s calendar and the in the event’s details the owner is still a participant.
Note	This causes an inconsistency because all the participants have no chance to know that the event is been canceled. In case the idea was to only to remove the participation, the delete feature is missing.

Edit event	
Goal	Verify consistency of navigation
Enviroment	The user is logged in, in the calendar page.
Input	The user insert the url “localhost:8080/MeteoCal/user/edit-event.xhtml”
Expected output	An error message, a page for the creation of an event or a redirect to the homepage
Obtained output	The page for editing the first event created by the user, even if the event has been deleted.
Errors	<ul style="list-style-type: none"> • The owner can edit a deleted event • The owner can invite multiple times the same user • If the user accepts the invites for the same event more than once the system crashes

Edit event	
Goal	Change location
Enviroment	The user is logged in, in the edit event page
Input	The user modify the location
Expected output	The location is correctly updated and also the weather forecast
Obtained output	Correct as expected
Note	<p>The user cannot understand if the system is checking the right locations between all the possible solutions (e.g. "San Jose")</p> <p>There is no indication whether the location is understood by the system before the saving.</p> <p>The user doesn't know the location format to make the system understand correctly.</p>

Edit privacy	
Goal	Edit the privacy of an event
Enviroment	The user is logged in and in the edit page of an event. His calendar is public
Input	The owner switch the privacy
Expected output	The privacy is changed. The event is visible with its details by another user accordingly to the privacy.
Obtained output	As expected

Answer to an invitation	
Goal	Remove participation from an event
Enviroment	The user is logged in, accepts an invite
Input	The user changes his mind and clicks on "remove participation"
Expected output	The event is removed from his calendar and his decision is available to the owner and the other participants.
Obtained output	The event is removed but the user is still in the participants list.
Note	Once a user decline or remove his participation, he cannot change his mind anymore

Invite an user to an event	
Goal	Create an event and invite user
Enviroment	A user X creates an event.
Input	He edit the event inviting Y 4 times and then clicks "Save"
Expected output	The user Y is invited only once and is the only invitee
Obtained output	Y is invited one time but after saving the event, when the owner edits the event, the invited users list shows two times Y and one time another user Z (who didn't receive nothing)
Note	The problem is that owner shouldn't be allowed to invite multiple times the same users. This causes a set of inconsistencies (e.g. notifications for events that were already in the calendar, users in both declined and participants list, etc) that can lead to the system fault.

Check time conflict	
Goal	Check time conflict
Enviroment	The user has 2 events
Input	The user overlaps the events by editing one of them
Expected output	The system alerts the user of the time conflict
Obtained output	As expected

Modify an event dragging it	
Goal	Modify event day and time
Enviroment	The user is logged in, in the homepage calendar
Input	The user drags an event and drops it in a different day and/or time
Expected output	The event details are updated
Obtained output	As expected

Modify an event resizing	
Goal	Modify event day and time
Enviroment	The user is logged in, in the homepage calendar
Input	The user resizes an event changing day and/or time
Expected output	The event details are updated
Obtained output	As expected

Import	
Goal	Import a deleted event
Enviroment	The user exported a calendar with one event and some participants.
Input	The user deletes the event and then he imports the calendar
Expected output	The event is correctly imported
Obtained output	As expected
Note	The calendar is exported without participants to the events so it's a weak form of backup

Notification when weather forecast change	
Goal	Check receive weather forecast changed notification
Enviroment	The user X has an event with a valid location. The weather forecast changes
Input	-
Expected output	The user is notified
Obtained output	As expected
Note	To test this system behavior we restarted the server to make the system rechecking the weather forecast (instead of waiting)

Final Consideration

The system behavior is almost correct if the user follows the expected navigation and doesn't force the system. The system meets the given specification and it has been improved with additional features, such as token confirmation and setting preferences. Just in few cases we expected a more robust behavior. The user interface is well-finished and user friendly.