Politecnico di Milano

Scuola di Ingegneria Industriale e dell'Informazione

Computer Science and Engineering

Software Engineering 2 Project – A.Y. 2014/15

Acceptance

Testing

Document

Authors

Francesco Lattari (838380)

Alessandro Rimoldi (835506)

Summary

1. Document's purpose

This document has the purpose to verify and validate the MeteoCal project developed by Andrea Bignoli and Leonardo Cella for the 2014 Software Engineering 2 course. This document will provide a report of all the test cases described in the *Test Cases* document available at this address:

https://code.google.com/p/meteocal-bignoli-cella/source/browse/

To deploy and use the application in a proper way, we have followed their indications written in the *Installation Guide* and in the *User Guide*. However there is an error in the *Installation Guide*: the JDBC Resource is reported as *jdbc/connectionToMeteocalDB* but it must be *jdbc/connectionToMeteoCalDB*. Without this fix, the server cannot deploy the application.

2. Test cases

2.1 Registration

Goal	Perform a new registration.
Environment	Index page.
Assumptions	The user is not already registered.
Expected Output	The system validates the data and creates a new user.
Flow of Events	• The guest fulfills the form on the left of the page with a username and a couple of equal passwords. • The guest clicks on signup.
Real Output	Same as expected. Furthermore the user receives a feedback from the system about the positive outcome of the operation. The database correctly stores the information.
Test Notes	All the possible errors are properly handled: empty mandatory fields, miss-match between password and confirmation, username and password to short or badly formed and username already in use generate error messages.

2.2 Log in

Goal	Perform a login.
Environment	Index page.
Assumptions	The user is already registered to MeteoCal.
Expected Output	The system analyzes the data and redirects the user calendar page.
Flow of Events	• The user fulfills the form on the right of the page with his username and his password. • The user clicks on login.
Real Output	Same as expected.
Test Notes	We have tried to perform a login with all possible combinations of bad information: empty username and password, correct username and wrong password, random username and correct password, both wrong username and password. In all these cases the system doesn't perform the login action and provides error messages. However if a user, once logged, manually inserts the address of the *Index page*, he can perform again the login, even if he's already logged. This must be avoided.

2.3 Access settings

Goal	Access to personal settings.
Environment	Any page with the top bar available.
Assumptions	The user is logged.
Expected Output	The user gets redirect to Setting.xhtml and can see his information.
Flow of Events	• The user clicks on "Account → Settings", visible on the top bar menu.
Real Output	Same as expected.
Test Notes	Nothing to report. The function works well in every page of the application.

2.4 Edit settings

Goal	Edit personal information and preferences.
Environment	Setting page.
Assumptions	The user is logged.
Expected Output	The user settings are updated with the new values.
Flow of Events	• The user clicks on "edit". • The system loads SettingsEdit.xhtml. • The user changes what he wants. • The user clicks on "save". • The system loads the user calendar page.
Real Output	Same as expected. The database correctly saves the new information.
Test Notes	We have tried to insert a not well formed email. The system recognizes this error and doesn't allow the user to save this information. The system also avoids modifications to the password if the "previous password" field is empty or contains the wrong word. In this case the system loads Error.xhtml. Finally, if the user leaves the page without press "save" button, the system doesn't save the new information. In this way, nothing changes into the database.

2.5 Log out

Goal	Log out from the system.
Environment	Any page with the top bar available.
Assumptions	The user is logged.
Expected Output	The user session gets closed.
Flow of Events	• The user clicks on "Account → Logout". • The system load the index page.
Real Output	Same as expected.
Test Notes	Nothing to report.

2.6 Own calendar page

Goal	Visualize user calendar page.
Environment	Any page with the top bar available.
Assumptions	The user is logged.
Expected Output	The user views his calendar page.
Flow of Events	• The user clicks on the first button from left of the top bar. • The system loads the calendar page.
Real Output	Same as expected.
Test Notes	Nothing to report. The function works well in every page of the application.

2.7 View event on own calendar page

Goal	Visualize the details of an event in own calendar page.
Environment	User home page.
Assumptions	The user is logged.
Expected Output	The system shows the details of the event.
Flow of Events	• The user clicks on an event visible in the calendar page. • The system loads a new page with all the details
Real Output	Same as expected.
Test Notes	Nothing to report.

2.8 Search

Goal	Search users and events based on username or event name.
Environment	Any page with the top bar available.
Assumptions	The user is logged.
Expected Output	The user gets redirected on a search results page.
Flow of Events	• The user writes something into the "search" input text. • The user press the lens button. • The system loads a list of results.
Real Output	Same as expected.
Test Notes	Nothing to report. The functionality works well: it finds results if available, otherwise signalizes "No records found."

2.9 Event creation

Goal	Create a new event.
Environment	Any page with the top bar available.
Assumptions	The user is logged.
Expected Output	A new event is created on the system using the provided data.
Flow of Events	• The user clicks on "Events → Create". • The system loads EventCreate.xhtml. • The user inserts all the mandatory information. • The user clicks on "create" button. • The system saves the event into the database and redirects the user to his own calendar page.
Real Output	Same as expected.
Test Notes	The system checks that all the mandatory information are present and doesn't allow to create events without them. The system checks also the consistency of the start time and the end time. They can't be previous of the current data and the start time can't be late of the end time. However the system doesn't alert in a proper way the user about that, it only refreshes the event creation page. If the organizer tries to invite a user that doesn't exist, the system will not add him to the participant list but it will not display any type of error. The user can select to avoid sun, rain, snow and clouds at the same time.

2.10 Visualize event details

Goal	Event visualization
Environment	Event page
Assumptions	The user is logged and he reached the event page by using the search engine.
Expected Output	The system has to show the event details according to the privacy policy of the event.
Flow of Events	No flow of events.
Real Output	Same as expected.
Test Notes	The functionality works well. The system shows all the details only if the event is public or, in any case, only if the user that is checking the event he's also the organizer.

2.11 Modify event

Goal	Edit event data.
Environment	Edit event page.
Assumptions	The user is logged and he's the organizer of the event.
Expected Output	The system saves the new information into the database.
Flow of Events	• The user clicks on "edit"; • The system loads EventPageCreator.xhtml. • The user inserts all the mandatory information. • The user clicks on "save editing".
Real Output	Same as expected.
Test Notes	View *Event Creation* test case (2.9). The invitation form is not available as expected.

2.12 Invite more users

Goal	Invite other users.
Environment	Event page.
Assumptions	The user is logged and he's the organizer of the event.
Expected Output	The system sends invitation to the users chosen by the organizer.
Flow of Events	• The user insert the username of the guest. • The user clicks on "add participant". • The system adds him to the invited users list.
Real Output	Same as expected.
Test Notes	If the organizer tries to invite the same users more than one time, the system doesn't send again the invitation and it doesn't add him twice to the invited users list.

2.13 Participate to an event/Cancel participation

Goal	Participate to an event or cancel participation.
Environment	Event page.
Assumptions	• The user is logged and the event is public (for the participation case); • The user is logged and he's in the participants list of the event.
Expected Output	The user joins the event/The user leaves the event.
Flow of Events	• The user click on "participate" or "cancel", based on his current situation. • The system *adds him to*/*removes him from* the participants list.
Real Output	Same as expected.
Test Notes	Nothing to report. The user can also perform this action more than one time: for example, he removes himself, then he adds himself then he removes himself again.

2.14 Suggested change

Goal	Edit event schedule, following a suggested change.
Environment	Suggested change page.
Assumptions	The user is logged and he's the creator of the event.
Expected Output	No modification to the event details if the user rejects the suggestion provided by the system, otherwise the database stores the new information.
Flow of Events	• The system suggests a change, if available, and the user has the possibility to confirm or manually change the schedule.
Real Output	Same as expected.
Test Notes	Nothing to report.

2.15 Accept/Refuse a request to join an event

Goal	Accept or reject a request to join an event.
Environment	Invitation management page.
Assumptions	The user is logged and he has received an invitation to an event.
Expected Output	The user is added to the participants list if he accepts the invitation, otherwise the invitation will be deleted.
Flow of Events	First path: • The user clicks on “Join”. • The system add him to participants list and starts to display the event on his calendar. Second path: • The user clicks on “Refuse”. • The system remove the invitation from the user list of invitations.
Real Output	Same as expected but with some errors. View the test notes.
Test Notes	If the user accepts the invitation, the invitation still be visible on his invitations list and the “accept” and “reject” buttons are still available. So a user can accept the invitation continuously even if the system won’t do anything about that. The worst thing is that a user can first accept and then reject the invitation.

2.16 Unseen invitations/notifications count increase/decrease

Goal	Check that the counters available in the top menu bar work well.
Environment	Any page with the top bar available.
Assumptions	The user is logged.
Expected Output	The unseen invitations counter increases/decreases. The unseen notifications counter increases/decreases.
Flow of Events	• The user receives a notification/invitation. • The counter automatically increases. • The user checks the new notification/invitation. • The counter automatically decreases.
Real Output	Same as expected.
Test Notes	Nothing to report.

2.17 Notifications view

Goal	Notification visualization
Environment	Notification management page
Assumptions	The user is logged.
Expected Output	The system displays all the notifications the user has received.
Flow of Events	No flow of events.
Real Output	Same as expected, but there are some inaccuracies. View test notes.
Test Notes	If the user clicks on notification id the system loads the error page. Sometimes no event name visible in “Event name” field and when available it’s not clickable.

2.18 Visualize a generic user calendar

Goal	Visualize the calendar of another user and see his scheduled events.
Environment	Any page with the top bar available.
Assumptions	The user is logged.
Expected Output	The requested calendar page is shown if and only if his creator set it as public in his account settings. The private events that would be represented on this page are masked nevertheless and no information is shown about them other than the fact that the user is busy in that time frame.
Flow of Events	The user finds, using the search bar, the calendar page of another user.
Real Output	The system always shows the user calendar page, even if it’s private. At least, in this case the system doesn’t show his appointments. If the calendar privacy is set to “visible”, the user can see his public events but he can’t click on them and view their details. If the calendar privacy is set to “visible” and an event is set to private, the user can view that the request user is busy, but clicking on the event will redirect to the error page.
Test Notes	Nothing to add.