



POLITECNICO DI MILANO

SOFTWARE ENGINEERING 2 PROJECT

---

# MeteoCal

## Acceptance Testing Document

---

*Authors:*  
Andrea CELLI  
Stefano CEREDA

February 8, 2015

# Contents

I	Introduction	2
II	RASD	3
1	Weather forecast	3
2	Bad weather alert for the event creator	3
III	Design document	4
3	Date of birth and city	4
4	Closest day with expected weather	4
IV	Automated test	5
V	Manual test	5
5	Registration	5

## Part I

# Introduction

The assigned project is <https://code.google.com/p/meteocal-iodicefinardi/>.

In this document we will start describing what, in our opinion, does not match the project assignment and then we will document the various phases of the acceptance testing.

## Part II

# RASD

### 1 Weather forecast

In the RASD (page 4 section 1.1 for the first time) it's stated that the user can choose a kind of weather, while the project assignment clearly states that the system has to handle *bad* weather forecast. This is a problem because it's impossible to obtain a behavior that matches the one of the assigned system, suppose a scenario like this:

Event weather	assigned behavior	wanted=sunny	wanted=cloudy
sunny	good	good	bad
cloudy	good	bad	good

This is assuming that cloudy is not a bad weather. The only situation where the two behavior matches is when we consider only sunny to be a good weather, but we think that this is a great limitation, because in most of the cases we don't reschedule our appointments only because the weather is partially cloud. A more reasonable approach is to consider sunny, partially cloud and cloud as good weather and everything else as bad weather.

This approach is reflected in the domain assumptions (page 6 2.2) where it's stated that "A person desires at most one type of weather for an event."

The real problem is how the partially cloud forecast is handled, and that's not clearly stated in any part of the rasd.

### 2 Bad weather alert for the event creator

The RASD states (page 10 section 3.2.1) that registered users should "Be notified (one day before the event) whenever an outdoor event they accepted an invitation for has an unfavourable weather forecast."

We don't think that this is the assigned behavior. The assignment states that the first type of notification should be sent to all the event's participant, but we think that even the event's creator should be considered a participant.

### 3 Different cities with the same name

The document does not clarify how the system will deal will different cities that have the same name (for example [http://openweathermap.org/help/city\\_list.txt](http://openweathermap.org/help/city_list.txt) contains 7 different Dublin). Neither the DD explains this issue.

Looking at the code of WeatherHttpClient.java we saw that the meteo is obtained passing the direct name of the city. Therefore if the user inserts Dublin as event location the forecast will be downloaded for Dublin, US (it comes first in the city list) instead of the more reasonable Dublin, IE. The user could overcome that by inserting "Dublin, IE" as location, but that's not stated anywhere and the application does not specify in any way which is the picked city, so the user can't be aware of the error.

## Part III

# Design document

## 4 Date of birth and city

In the dd (page 7 section 3.1.1-User) it's stated that a user should insert his date of birth and the city where he lives. These attributes are not necessary for any of the requirements in the rasd. Moreover, using the application we noticed that a user can only be older than 10 or without a date of birth (during the registration the date of birth is not mandatory but if present can only be set prior to 10 years ago). We think that this makes no sense.

## 5 Closest day with expected weather

The dd assumes (page 7 section 3.1.1-WeatherNotification) that a Weather-Notification (the one sent three days in advance to the event organizer) should contain an information about the closest day with the expected weather. We don't think this is a good choice, because from the time when the notification is created to the time when the notification is visualized the forecast could change, making the notification incorrect. We think that the closest good day should be searched when the notification is visualized from the user.

## **Part IV**

# **Automated test**

The unit and integration automated test does not raise any problem. Their code seems correct.

## **Part V**

# **Manual test**

We tried the manual tests proposed in the System Testing document and we also tried all the rasd's requirements with "strange" input. Here we will report only the things that don't work as expected, everything that is not stated works well.

## **6 Registration**

After completing the registration the browser proposes to save the login credentials, but for username it takes the date of birth instead of the email.