



POLITECNICO DI MILANO

MASTER OF SCIENCE IN COMPUTER SCIENCE AND ENGINEERING

# METEOCAL

A WEATHER BASED ONLINE CALENDAR

## TEST CASES ANALYSIS

### Teachers

Raffaella Mirandola  
Marco Miglierina

### Students

Alessandro Negrini 836806  
Andrea Gulino 836681  
Paolo Guglielmino 837055

ACADEMIC YEAR 2014/2015



# Contents

---

<b>1</b>	<b>Introduction</b>	<b>2</b>
1.1	Provided Material . . . . .	2
<b>2</b>	<b>Test Cases specification</b>	<b>3</b>
2.1	Registration . . . . .	5
2.2	Login . . . . .	6
2.3	Logout . . . . .	7
2.4	People Search . . . . .	8
2.5	New Event Creation . . . . .	9
2.6	Repeated Event Creation . . . . .	10
2.7	Calendar export . . . . .	11
2.8	Calendar import . . . . .	12
2.9	Avoid Event date conflicts . . . . .	13
2.10	Restore password in case of password forgotten . . . . .	14
2.11	'Feel free to contact us' mail sending . . . . .	15
2.12	Update image profile . . . . .	16
2.13	Delete profile . . . . .	17
2.14	Private calendar . . . . .	18
2.15	Public calendar . . . . .	19
2.16	Follow user . . . . .	20
2.17	UnFollow user . . . . .	21
2.18	Invite to event an already followed user . . . . .	22
2.19	Acceptance Invite . . . . .	23
2.20	Only user administrator can manage events . . . . .	24
2.21	Edit Event . . . . .	25
2.22	Delete Event . . . . .	26
2.23	User weather updates . . . . .	27
2.24	Find the closest sunny day . . . . .	28
2.25	Weather changes notification . . . . .	29
2.26	Mail Three days before . . . . .	30
<b>3</b>	<b>Performance Test</b>	<b>31</b>
3.1	Request Index Page graph result . . . . .	31
3.2	Login graph result . . . . .	32
3.3	Registration graph result . . . . .	32
<b>4</b>	<b>Appendix A: References</b>	<b>33</b>

# 1

## Introduction

---

The Test Case document documents the functional requirements of the MeteoCal web application test case. The intended audience is the project manager, project team, and testing team. Some portions of this document may on occasion be shared with the client/user and other stakeholder whose input/approval into the testing process is needed.

During the development we have figured out several test cases, but in this document only the most significative will be provided.

### 1.1 Provided Material

In order to test our system, these software were used :

1. RASD Document
2. DD Document
3. Source code of our web application
4. Working application with input as example
5. An up-to-date web browser. We guarantee that our software works completely using **Google Chrome** as browser, using the other browser the main functionalities are met, but it could happen that some graphic details are lost

Before starting, we want to recall that our first versions of RASD Document and DD Document have been updated, so before starting reading this document, please have a look at them.

# 2

## Test Cases specification

---

We chose our test set not in a random way, but following a systematic test procedure according to source code and to application behaviour.

For each test a summary table is presented, containing these following fields:

- **Id number**
- **Goal**
- **Working environment**
- **Inputs**
- **Preconditions**
- **Expected Outcome**
- **Actual Outcome**
- **Final Outcome**
- **Screenshot**



## 2.1 Registration

<b>Id</b>	1
<b>Goal</b>	The system must allow a new user to create a new account and register
<b>Working environment</b>	Web application signup page <i>signup.xhtml</i>
<b>Preconditions</b>	The user must not be already registered using that mail
<b>Inputs</b>	Casual, but well formed data, for name, avatar logo (dimension less than 1 MB), email, password and location. In particular the email must follow the typical email syntax <i>email@domain.domain</i> and not already used, password doesn't have particular requirements and location must be written in the correct way
<b>Expected Outcome</b>	The system should create a user ( and persist it ) and associate him/her the location and the weather forecast associated. In addition the system will also start a timer that updates periodically user weather forecast
<b>Actual Outcome</b>	The user is actually created and the information are stored in the database
<b>Final Outcome</b>	After having typed correctly each field of the signup form, clicking on signup button the system creates the new account associated to that email and password. The system redirects to the home page where user is expected to log in using the apposite form. If some errors occurs ( related to syntax issues, i.e email pattern wrong, password not matching, ... ) the system locks the registration showing apposite error messages.

### Screenshot

**MeteoCal** WHY CODE DOCUMENTATION CONTACTS

JOIN METEOCAL RIGHT NOW

Alessandro Negrini

alnegri92@gmail.com

\*\*\*\*\*

\*\*\*\*\*

Where do you live?  
Milano

Map data ©2015 Google Terms of Use Report a map error

By signing up, you agree to the [Terms of Service](#) and [Privacy Policy](#), including Cookie Use. Others will be able to find you by email.

**MeteoCal** WHY CODE DOCUMENTATION CONTACTS

JOIN METEOCAL RIGHT NOW

Alessandro Negrini

alnegri92@gmail.com

Insert a correct email

\*\*\*\*\*

\*\*\*\*\*

Where do you live?  
Milano

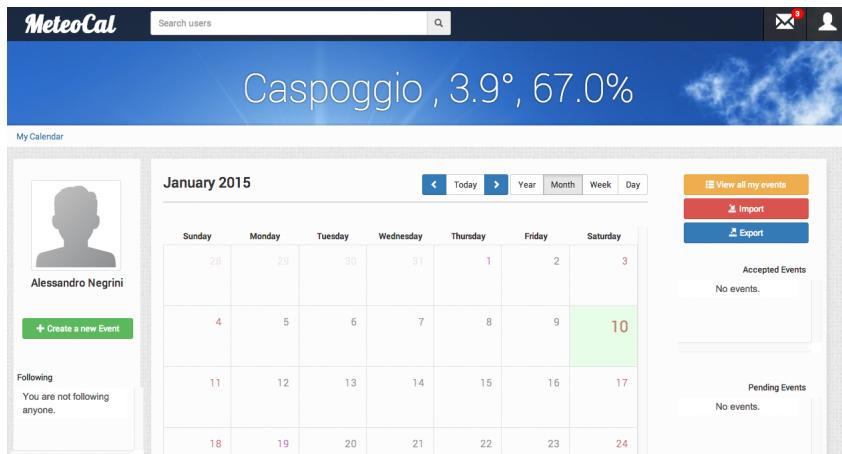
Map data ©2015 Google Terms of Use Report a map error

By signing up, you agree to the [Terms of Service](#) and [Privacy Policy](#), including Cookie Use. Others will be able to find you by email.

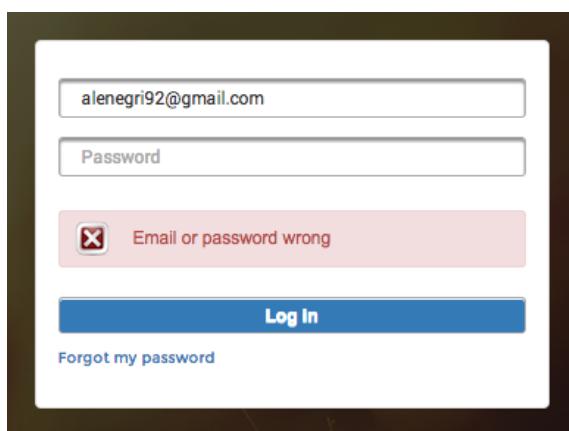
## 2.2 Login

<b>Id</b>	2
<b>Goal</b>	The user can be able to log into the application
<b>Working environment</b>	System home page <i>index.xhtml</i>
<b>Preconditions</b>	The user must be registered
<b>Inputs</b>	User Email and password
<b>Expected Outcome</b>	Clicking on Login Button the system must accept the login and redirect user to its home page <i>home.xhtml</i> . In addition, the system must maintain the session
<b>Actual Outcome</b>	The user is actually logged in and the user home page is shown
<b>Final Outcome</b>	The system shows the user home page. In case of errors (i.e user not registered, wrong password or wrong email), the system shows an error message

Screenshot success



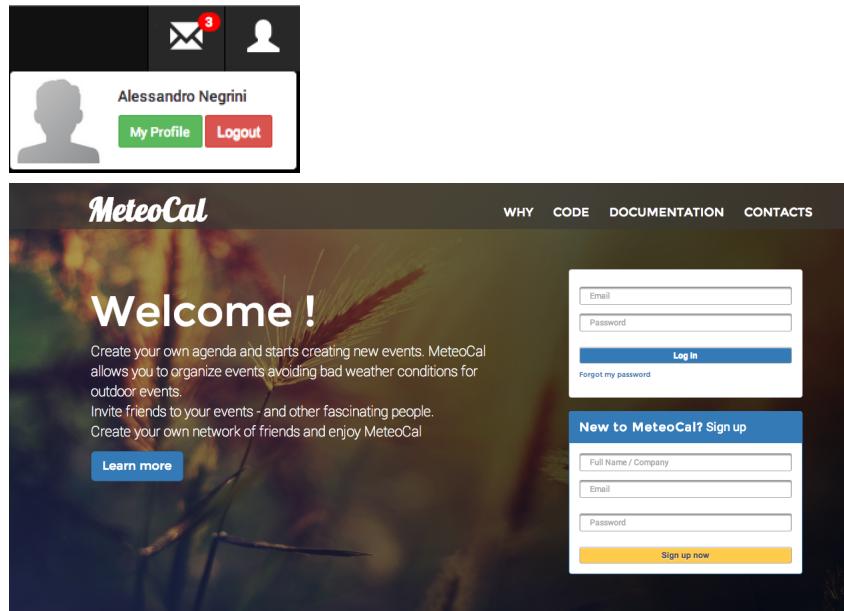
Screenshot failure



## 2.3 Logout

<b>Id</b>	3
<b>Goal</b>	The user can be able to log off the application
<b>Working environment</b>	Any user page <i>user/ pageName.xhtml</i>
<b>Preconditions</b>	The user must be registered and logged in the application
<b>Inputs</b>	User that wants to log off must press on logout button
<b>Expected Outcome</b>	Clicking on Logout Button the system must accept the request of logging out and system must redirect the user to the application home page <i>index.html</i> Session must be closed
<b>Actual Outcome</b>	The user is actually logged off and the application home page is shown
<b>Final Outcome</b>	The system shows the home page.

Screenshot success



Screenshot failure

-

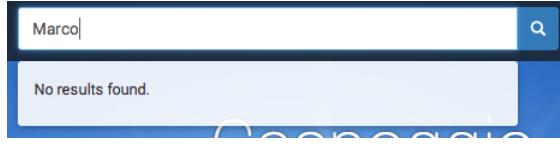
## 2.4 People Search

<b>Id</b>	4
<b>Goal</b>	The user can be able to search for a user according to his name or surname
<b>Working environment</b>	Any user page <i>user/</i> <i>pageName.xhtml</i>
<b>Preconditions</b>	The user must be registered and logged in the application
<b>Inputs</b>	User must input the name or the surname (or a substring of them) of the target user.
<b>Expected Outcome</b>	<p>System dinamically shows the list of user with that name or surname ( or a substring of them).</p> <p>An example if provided in order to be more clear: if the user types the letter 'A' the system shows all user with name or surname starting with 'A', adding 'l' it will show all users whose name or surname starts with 'Al' and so on so forth.</p>
<b>Actual Outcome</b>	The list is correctly shown according to the input
<b>Final Outcome</b>	A list of users who satisfies the name/surname typed is shown. In case of there is anybody with such a name or surname, the list will be empty

### Screenshot Not Empty



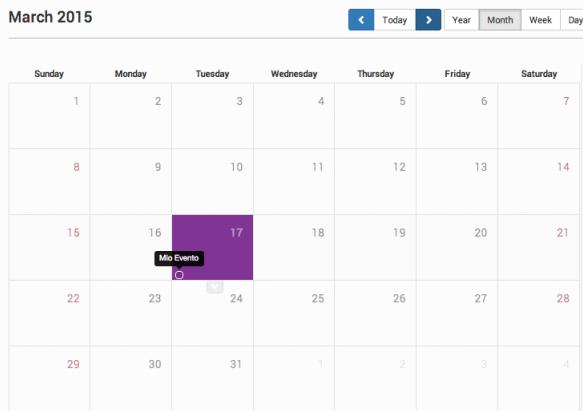
### Screenshot Empty



## 2.5 New Event Creation

<b>Id</b>	5
<b>Goal</b>	The user can create its own event avoiding data conflicts with other events
<b>Working environment</b>	New event creation page <i>signup.xhtml</i>
<b>Preconditions</b>	The user must be logged into the system
<b>Inputs</b>	Random values for event name, the location, a starting date and an ending date, a description, the weather for which the user must be alerted in case of outdoor event and the privacy settings. All fields must be shorter than 40 characters and description shorter than 100.
<b>Expected Outcome</b>	System must create the event adding it to the database avoiding data conflicts with other events. In addition the system must associate it its owner, the location (in terms of latitude and longitude) and the expected weather if available. In case of weather not available a particular weather state is associated.
<b>Actual Outcome</b>	The system creates the event and redirects user to its own page showing him the event correctly scheduled in the calendar.
<b>Final Outcome</b>	The system redirects user to its own page . In case of errors, that can be due to data consistency error or data conflicts with other events, an error feedback is shown, describing which is the error

### Screenshot Success



### Screenshot Failure

**Create a new Event**

Error: Date Consistency Issues Please check if: - 'From' field is bigger than 'To' field <br> - Date fields must be bigger than the current date <br> - In case of repeated event, the ending hour must be bigger than the starting one <br>

**Event Details**

Event Title: Cena con Amici

Location: Milano  
Viale Brianza  
36

Indoor  Outdoor

Additional location info:

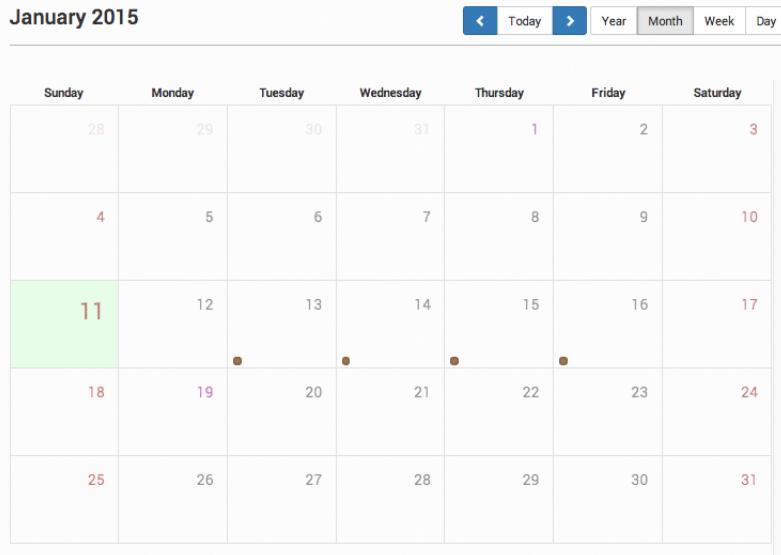
Starts: 20/01/2015 00:00      Ends: 18/01/2015 00:00

This event repeats

## 2.6 Repeated Event Creation

<b>Id</b>	6
<b>Goal</b>	The user can create events that can repeat periodically (daily or weekly)
<b>Working environment</b>	New event creation page <i>signup.xhtml</i>
<b>Preconditions</b>	The user must be logged into the system and the event must not be a straddle event between two days
<b>Inputs</b>	Random values for event name, the location, a starting date and an ending date ( a repeatability option) , a description, the weather for which the user must be alerted in case of outdoor event and the privacy settings.
<b>Expected Outcome</b>	The expected outcome must be the same of the creation event one. The only difference is that the event is repeated according to option set by the user
<b>Actual Outcome</b>	The system creates the event and redirects user to its own page showing him the event correctly scheduled in the calendar.
<b>Final Outcome</b>	The system redirects user to its own page . In case of errors, that can be due to data consistency error or data conflicts with other events, an error feedback is shown, describing which is the error

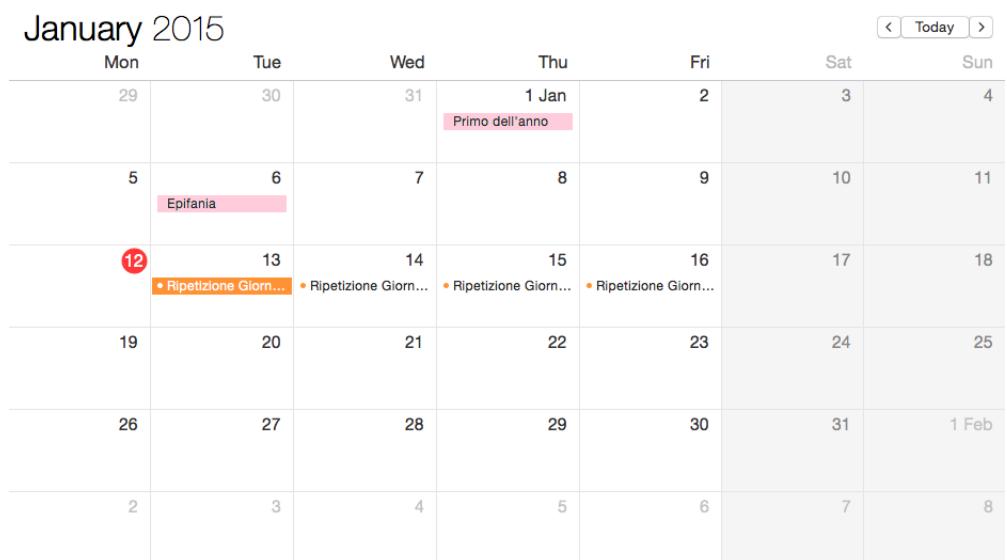
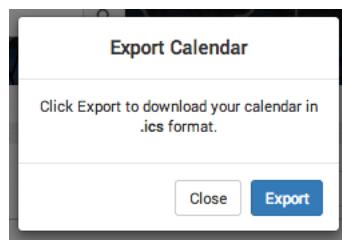
### Screenshot Success



## 2.7 Calendar export

<b>Id</b>	7
<b>Goal</b>	The calendar must be exported in ics format so it can be opened also in others calendar software
<b>Working environment</b>	User home page <i>home.xhtml</i>
<b>Preconditions</b>	The user must be logged into the system
<b>Inputs</b>	-
<b>Expected Outcome</b>	<p>The user calendar must be exported in user computer and it can be opened using a Calendar software that allows ics as calendar format.</p> <p>The calendar exported must be the same of the user calendar, not an event less not an event more.</p> <p><i>all v : Event / v in UserCalendar implies v in CalendarExported</i></p>
<b>Actual Outcome</b>	<p>The user exports and opens correctly the calendar using a calendar software</p> <p>The file is consistent with the user calendar</p>
<b>Final Outcome</b>	User exports the calendar and he/she remains on his/her home page

Screenshot Success



## 2.8 Calendar import

<b>Id</b>	8
<b>Goal</b>	The calendar in format ics must be imported and be scheduled in the user calendar taking care of pre existing events and checking of possible conflicts. The expected weather forecasts are not set, but they're updated after update time expires
<b>Working environment</b>	User home page <i>home.xhtml</i>
<b>Preconditions</b>	The user must be logged into the system
<b>Inputs</b>	The ics format file
<b>Expected Outcome</b>	<p>The user calendar must be imported in database and it is shown in user home page as calendar format.</p> <p>The events scheduled must be the same of the events in calendar imported, not an event less not an event more.</p> <p><i>all v : Event / v in UserCalendar implies v in CalendarExported</i></p> <p>In case of errors the system must avoid the import and show the error</p>
<b>Actual Outcome</b>	It is the same of the expected one
<b>Final Outcome</b>	User imports the calendar and he/she remains on his/her home page

## 2.9 Avoid Event date conflicts

<b>Id</b>	9
<b>Goal</b>	The system must avoid event conflicts. So if just a minute of an event is overlapped with a time lapse of an event the system must avoid its creation.
<b>Working environment</b>	New Event creation page <i>newEvent.xhtml</i>
<b>Preconditions</b>	The user must be logged into the system and each field of event creation must be filled in
<b>Inputs</b>	The event starting and ending date
<b>Expected Outcome</b>	In case of error ( date conflict ) system must avoid the creation of event, displaying an error that explains why user is in error, otherwise a new event is created
<b>Actual Outcome</b>	It is the same of the expected one
<b>Final Outcome</b>	User creates the event and user is redirected to the user home page
<b>Screenshot conflict</b>	

☒ **Error: Date Consistency issues** There's a data conflict in the dates chosen. Check available timetables in the calendar.

### Event Details

**Event Title**  
Prova Test

**Location**  
Milano  
Viale Brianza  
36  
 Indoor  Outdoor

**Additional location info**

**Starts** 18/01/2015 00:00    **Ends** 22/01/2015 00:00



Map data ©2015 Google Terms of Use Report a map error

## 2.10 Restore password in case of password forgotten

<b>Id</b>	10
<b>Goal</b>	The system must make possible users to change password
<b>Working environment</b>	System index page <i>index.xhtml</i>
<b>Preconditions</b>	The user must be registered in the system
<b>Inputs</b>	The email associated to the user
<b>Expected Outcome</b>	<p>In case user can't remember his/her password he/she can restore it.</p> <p>By clicking on link 'Forgot password' and after having typed the correct email the system sends to the user an email that allows him/her to open a new page and create a new password.</p> <p>In case of this link is used more than once or more than 12 hours have passed from mail sending, the system must avoid user to create a new password</p>
<b>Actual Outcome</b>	It is the same of the expected one
<b>Final Outcome</b>	User is redirected to the home page, and he/she can login using the new password

Screenshot success



Reset your password



i Password has been reset Password has been reset

Submit

Screenshot failure



Reset your password



☒ Session Expired. The possible reasons are : you have already used this link, or a time lapse greater than 12 hours is passed

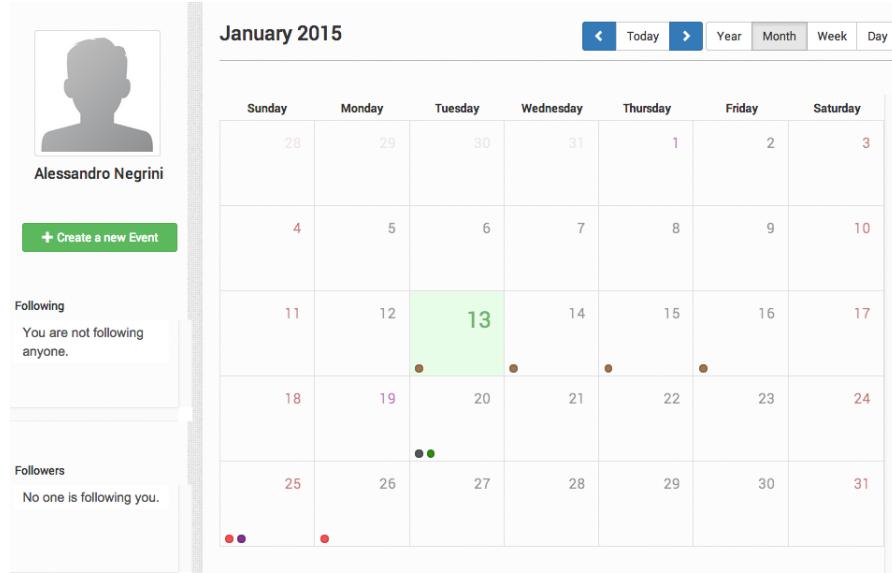
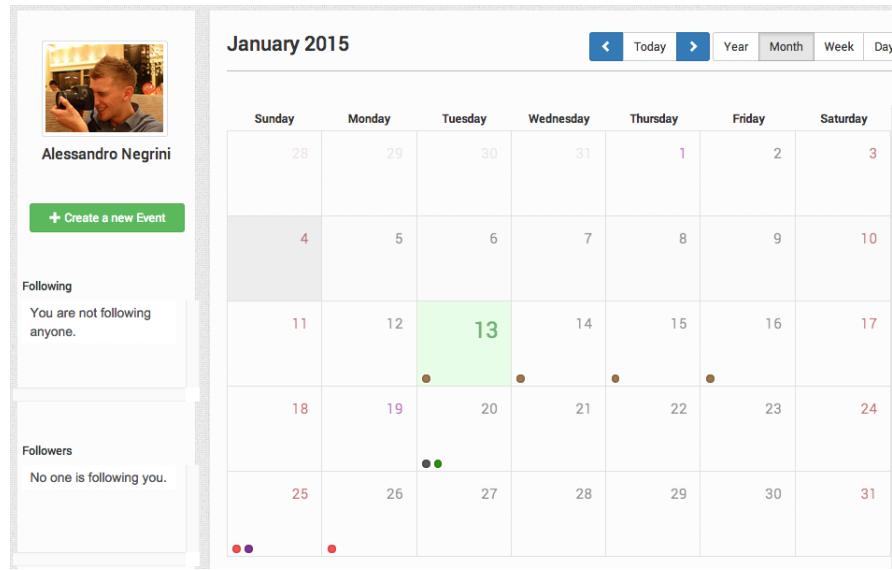
Submit

## 2.11 'Feel free to contact us' mail sending

<b>Id</b>	11
<b>Goal</b>	The system must send the mail from the correct user
<b>Working environment</b>	System index page <i>index.xhtml</i>
<b>Preconditions</b>	User inputs an existing email address
<b>Inputs</b>	A name, an email address and mail content
<b>Expected Outcome</b>	In case users have some requests they can contact the administrator by sending an email filling in a form with his/her name, email and email content. The system must send to the administrator the email with the sender email in order to make possible reply him/her
<b>Actual Outcome</b>	It is the same of the expected one
<b>Final Outcome</b>	User can keep to use the application as before

## 2.12 Update image profile

<b>Id</b>	12
<b>Goal</b>	The user can have the possibility to change his/her account avatar
<b>Working environment</b>	User profile page <i>myprofile.xhtml</i>
<b>Preconditions</b>	User must be logged in the system
<b>Inputs</b>	A casual photo that met with the format PNG, GIF or JPG
<b>Expected Outcome</b>	User personal photo must be updated in the database and when the user goes back to his/her home page he/she has to see the avatar updated.
<b>Actual Outcome</b>	It is the same of the expected one
<b>Final Outcome</b>	User is redirected to his/her own page and personal photo is updated

**Screenshot Before****Screenshot After**

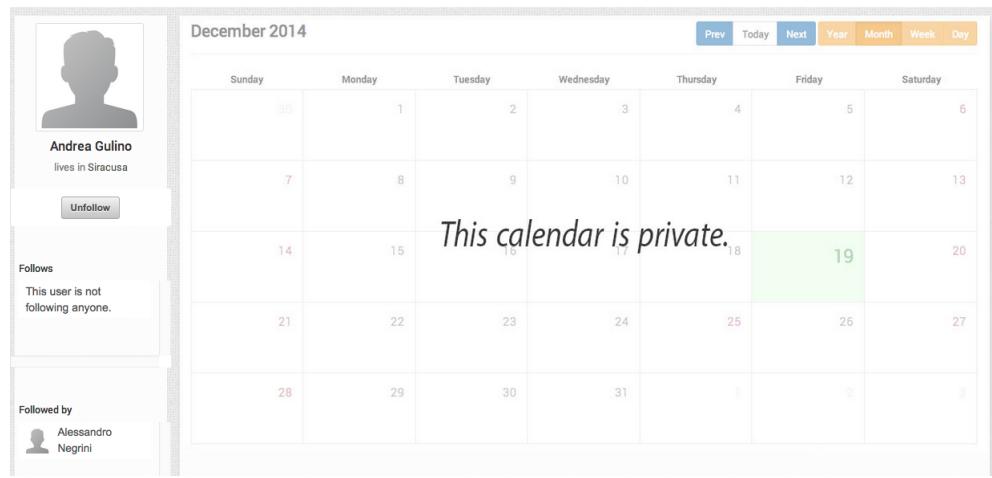
## 2.13 Delete profile

<b>Id</b>	13
<b>Goal</b>	The user can have the possibility to delete his/her account
<b>Working environment</b>	User profile page <i>myprofile.xhtml</i>
<b>Preconditions</b>	User must be logged in the system
<b>Inputs</b>	-
<b>Expected Outcome</b>	System must delete user from the database and it won't allow login with that email any more, unless he/she creates a new account with that email
<b>Actual Outcome</b>	The user is actually deleted form the DB and he can't login using the old credentials.
<b>Final Outcome</b>	The user is redirected to the index page

## 2.14 Private calendar

<b>Id</b>	14
<b>Goal</b>	A user can't see the private calendar of other users.
<b>Working environment</b>	User home page <i>home.xhtml</i>
<b>Preconditions</b>	User must be logged in the system
<b>Inputs</b>	Other user name
<b>Expected Outcome</b>	The logged user after having typed the name of his/her friends using search bar, by clicking on his/her name can have a look on his/her profile and calendar. If his/her calendar is private the user can't see it, and a message saying "This calendar is private"
<b>Actual Outcome</b>	It is the same of the expected one
<b>Final Outcome</b>	-

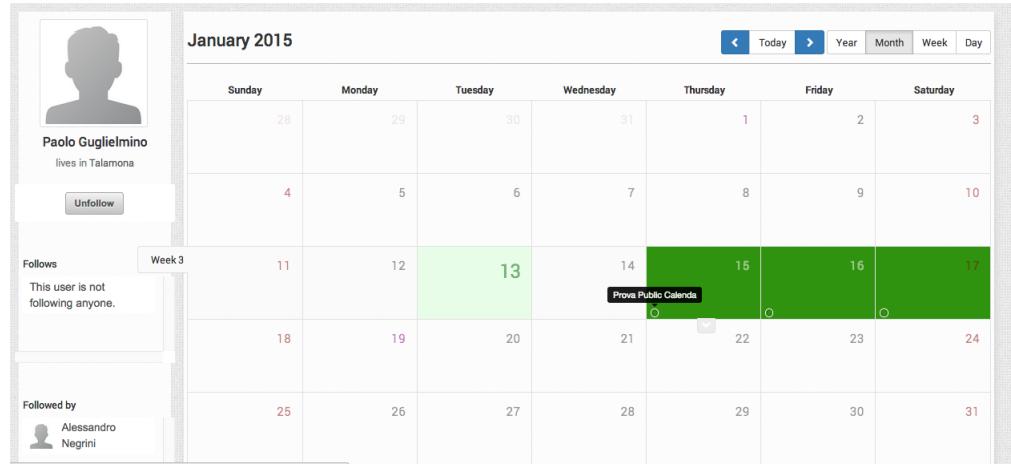
### Screenshot



## 2.15 Public calendar

<b>Id</b>	15
<b>Goal</b>	A user can see the calendar of another people if and only if this user has a public calendar
<b>Working environment</b>	User home page <i>home.xhtml</i>
<b>Preconditions</b>	User must be logged in the system
<b>Inputs</b>	Other user name
<b>Expected Outcome</b>	The logged user after having typed the name of his/her friends using search bar, by clicking on his/her name can have a look on his/her profile and calendar. If his/her calendar is public the user can see it and see only his/her public events scheduled
<b>Actual Outcome</b>	It is the same of the expected one
<b>Final Outcome</b>	-

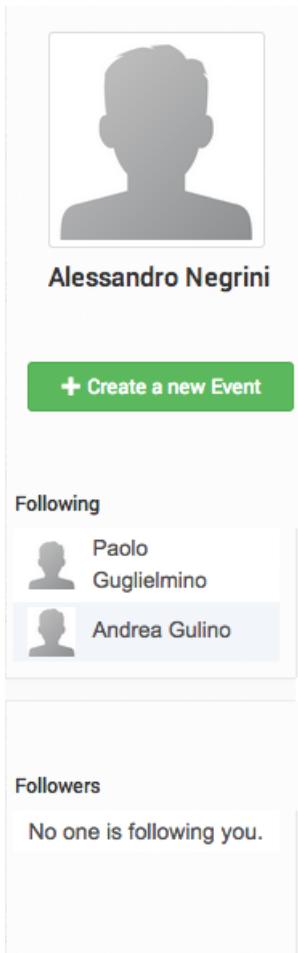
### Screenshot



## 2.16 Follow user

<b>Id</b>	16
<b>Goal</b>	A user can follow another user
<b>Working environment</b>	Each user home page
<b>Preconditions</b>	User must be logged in the system an the other user must not be already followed by him
<b>Inputs</b>	Other user name
<b>Expected Outcome</b>	<p>Let's call user A the logged user and user B the "target" user, the one that A wants to follow.</p> <p>A clicks on 'Follow' button and in A following section the B name must be added.</p> <p>Thus, in B followers section, A name must be added as well.</p>
<b>Actual Outcome</b>	It is the same of the expected one
<b>Final Outcome</b>	-

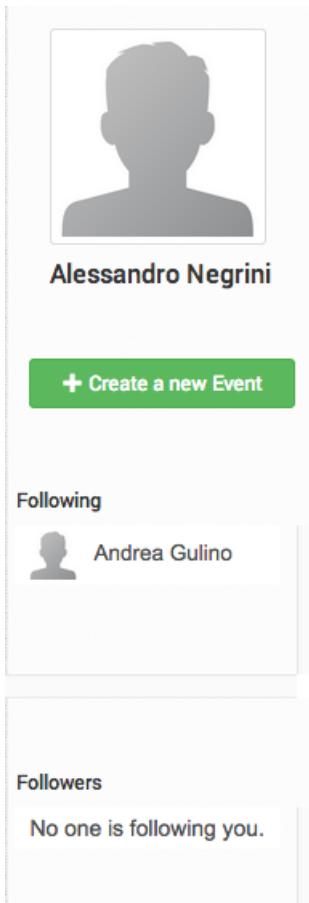
### Screenshot



## 2.17 UnFollow user

<b>Id</b>	17
<b>Goal</b>	A user can unfollow a followed user
<b>Working environment</b>	Each user home page
<b>Preconditions</b>	User must be logged in the system an the other user must already followed by him
<b>Inputs</b>	Other user name
<b>Expected Outcome</b>	<p>Let's call user A the logged user and user B the "target" user, the one that A wants to follow.</p> <p>A clicks on 'UnFollow' button and in A following section the B name must be deleted.</p> <p>Thus, in B followers section, A name must be deleted as well.</p>
<b>Actual Outcome</b>	It is the same of the expected one
<b>Final Outcome</b>	-

### Screenshot



## 2.18 Invite to event an already followed user

<b>Id</b>	18
<b>Goal</b>	A user can invite to his/her events some friends that are already in his/her following list
<b>Working environment</b>	Each event page
<b>Preconditions</b>	The user must already follow the user who wants to invite and he/she must have an event in calendar
<b>Inputs</b>	The name of users that want to invite
<b>Expected Outcome</b>	The user selects the name of users who want to invite and the system must send the email and a notification that invites him/her to the event. Then, the name of users invited is added to the event PENDING section
<b>Actual Outcome</b>	It is the same of the expected one
<b>Final Outcome</b>	The system redirects the user to the event detail page

### Screenshot

Prova 1 giorno

WHEN: Wed Jan 14 00:00:00 CET 2015 -> Wed Jan 14 02:00:00 CET 2015  
 ADDRESS: Milano  
 EXTRA:  
 DESCRIPTION:  
 FACEBOOK:  
 TWITTER:  
 ALERT IF: SUN  
 EXPECTED WEATHER: Clouds  
 OWNER: Alessandro Negrini  
 PENDING:

	Andrea Gulino	PENDING
--	---------------	---------

GOING: No accepted invitations.

DECLINED: No declined invitations.

### Screenshot Mail



## 2.19 Acceptance Invite

<b>Id</b>	19
<b>Goal</b>	An invited user can accept event
<b>Working environment</b>	-
<b>Preconditions</b>	The user must have been invited and so he/she must have received the invitation mail or notification
<b>Inputs</b>	-
<b>Expected Outcome</b>	When the user clicks on the notification (email), the system opens the invitation module where he/she can accept the invite After having accepted, that event must be added to his/her calendar and he can receive weather notification about that event
<b>Actual Outcome</b>	It is the same of the expected one
<b>Final Outcome</b>	-

### Screenshot



## 2.20 Only user administrator can manage events

<b>Id</b>	20
<b>Goal</b>	Only event administrator can manage his/her event. Invited users are allowed only to see details if accept, and receive weather notifications.
<b>Working environment</b>	Event page
<b>Preconditions</b>	The administrator must have an event, and he/she must have some invited to events
<b>Inputs</b>	-
<b>Expected Outcome</b>	When the administrator opens the page event he can manage the event (update and delete). When an invited open the same page the system must avoid him/her to manage it
<b>Actual Outcome</b>	It is the same of the expected one
<b>Final Outcome</b>	-

### Screenshot administrator

**Event Details**

**Prova 1 giorno**

Invite Friends

Edit

Cancel Event

**WHEN** Wed Jan 14 00:00:00 CET 2015 → Wed Jan 14 02:00:00 CET 2015

**ADDRESS** Milano

**EXTRA**

**DESCRIPTION**

**FACEBOOK**

**TWITTER**

**ALERT IF** SUN

**EXPECTED WEATHER** Clouds

**OWNER** Alessandro Negrini

### Screenshot invited

**Event Details**

**Your invitation**

**Prova 1 giorno**

Wed Jan 14 00:00:00 CET 2015 → Wed Jan 14 02:00:00 CET 2015

Milano

EXTRA

DESCRIPTION

FACEBOOK

TWITTER

ALERT IF SUN

EXPECTED WEATHER Clouds

OWNER Alessandro Negrini

## 2.21 Edit Event

<b>Id</b>	21
<b>Goal</b>	Administrator can update all event info
<b>Working environment</b>	Event page
<b>Preconditions</b>	The user who wants to update event's info must be his/her administrator
<b>Inputs</b>	-
<b>Expected Outcome</b>	User changes event's information, and all updates must be performed in the database
<b>Actual Outcome</b>	It is the same of the expected one
<b>Final Outcome</b>	User is redirected to his/her home page

## 2.22 Delete Event

<b>Id</b>	22
<b>Goal</b>	Only event administrator can delete his/her event. Invited users will receive a notification about it
<b>Working environment</b>	Event page
<b>Preconditions</b>	The user who wants to delete event must be his/her administrator
<b>Inputs</b>	-
<b>Expected Outcome</b>	When the administrator opens the page event he can delete the event. The event is not deleted from the database but its attribute 'IsCanceled' is set to 1. Anyway, the event is not visible on user calendar any more. All users that had been invited will receive a notification about event deletion.
<b>Actual Outcome</b>	It is the same of the expected one
<b>Final Outcome</b>	-

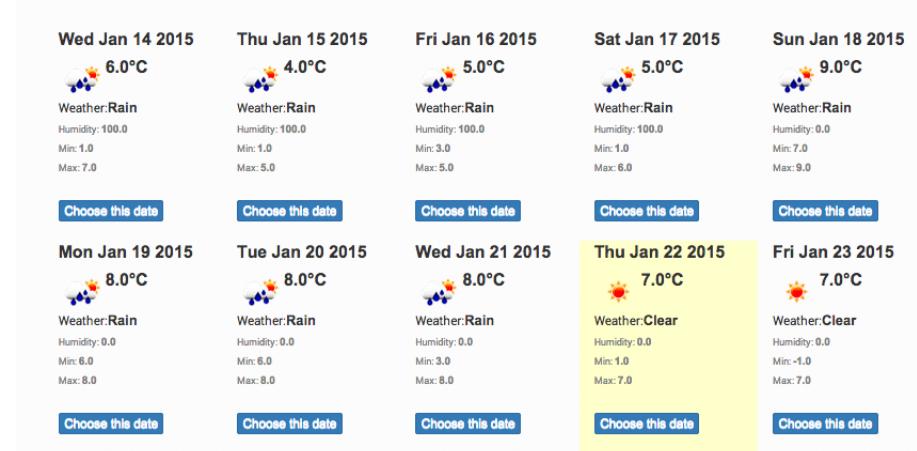
## 2.23 User weather updates

<b>Id</b>	23
<b>Goal</b>	Every (1,5,12 hours, depending on administrator settings) the weather associated to the user is updated
<b>Working environment</b>	Any page
<b>Preconditions</b>	Internet connection
<b>Inputs</b>	-
<b>Expected Outcome</b>	Periodically weather forecasts are updated and the banner at the top of any page is updated according to the actual weather forecasts
<b>Actual Outcome</b>	It is the same of the expected one
<b>Final Outcome</b>	-
<b>Screenshot administrator</b>	

## 2.24 Find the closest sunny day

<b>Id</b>	24
<b>Goal</b>	Administrator event must find the closest sunny day (if any) in case of expected bad weather
<b>Working environment</b>	-
<b>Preconditions</b>	Administrator must have the email with the link and must be already logged
<b>Inputs</b>	-
<b>Expected Outcome</b>	If the event administrator receives the email regarding to expected weather condition and he/she wants to change date, the system should propose him the closest sunny day within 10 days.
<b>Actual Outcome</b>	It is the same of the expected one
<b>Final Outcome</b>	-

Screenshot administrator



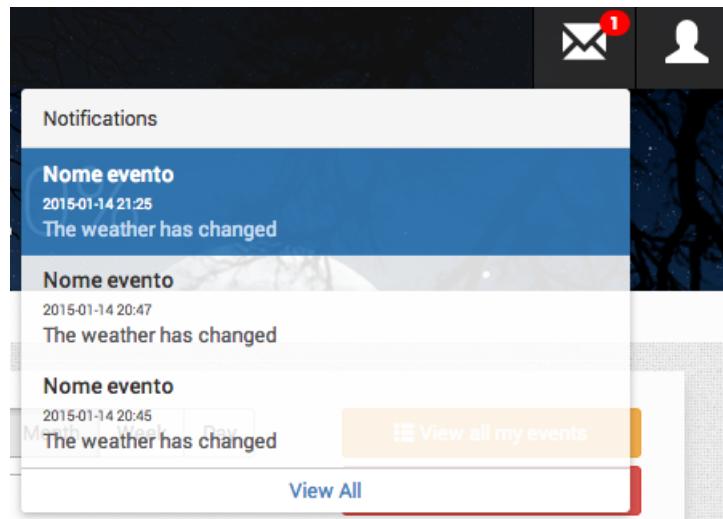
## 2.25 Weather changes notification

---

<b>Id</b>	25
<b>Goal</b>	All the participants to an event must receive a notification in case of the expected weather forecast change
<b>Working environment</b>	Any user page
<b>Preconditions</b>	User must have not declined an invitation to an outdoor event
<b>Inputs</b>	-
<b>Expected Outcome</b>	The system updates periodically the weather and if it is changed with respect to the previous one it send to all participants a notification.
<b>Actual Outcome</b>	It is the same of the expected one
<b>Final Outcome</b>	A user clicking on that notification is redirected to the event page

---

### Screenshot notification

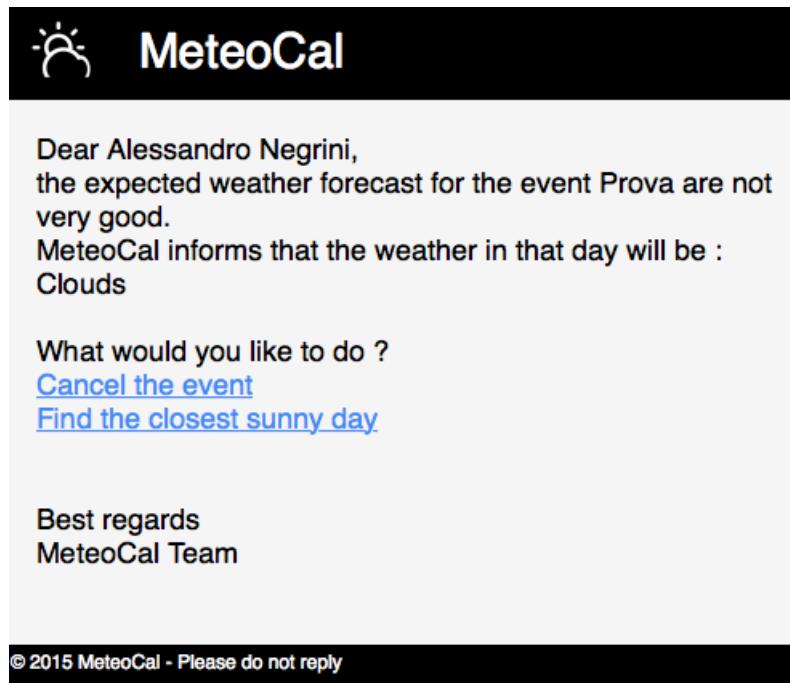



---

## 2.26 Mail Three days before

<b>Id</b>	26
<b>Goal</b>	The event administrator must be informed by mail if the expected event is still bad three days before it
<b>Working environment</b>	-
<b>Preconditions</b>	The user must have at least an event three days after the current date
<b>Inputs</b>	-
<b>Expected Outcome</b>	The system checks if the weather is bad, and in case the event is programmed for three days after ( or less ), it sends him/her the mail asking him/her what he wants to do
<b>Actual Outcome</b>	It is the same of the expected one
<b>Final Outcome</b>	A user can cancel the event or change date

Screenshot mail



# 3

## Performance Test

In software engineering, performance testing is in general testing performed to determine how a system performs in terms of responsiveness and stability under a particular workload. It can also serve to investigate , measure, validate or verify other quality attributes of the system, such as scalability, reliability and resources usage.

We performed a stress test on few pages of our application. In particular we tested the response time for login, registration and get home page, using JMeter as tool (<http://jmeter.apache.org/>)

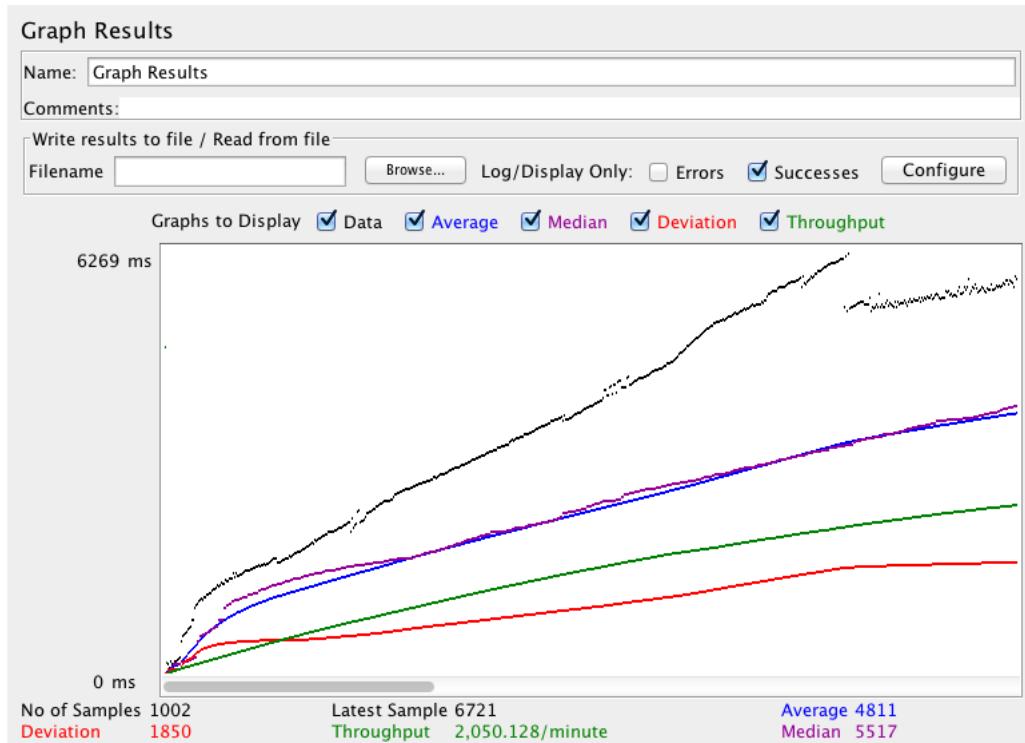
In this page we just want to show briefly the outcomes that we got. The reader won't find any detailed aspects in these few lines.

Any further detail you can find in our file (`MeteoCalJMeterTest.jmx`).

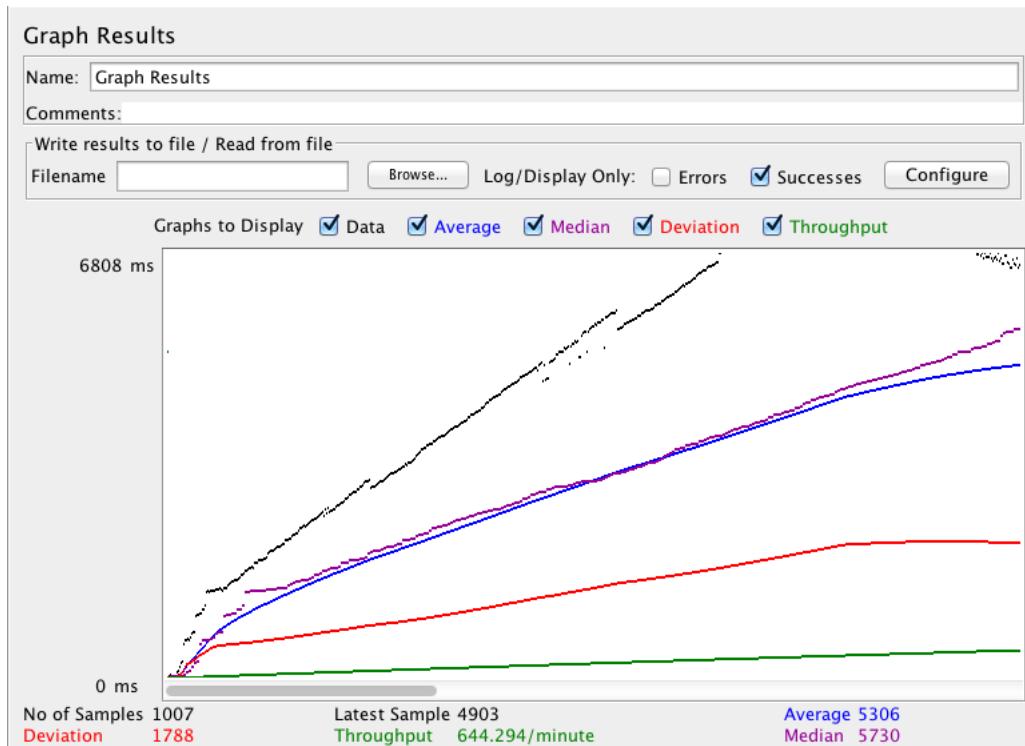
### 3.1 Request Index Page graph result



## 3.2 Login graph result



## 3.3 Registration graph result



# 4

## Appendix A: References

---

1. Past example of Test Cases document
2. Lecture slides
3. <http://users.csc.calpoly.edu/~jdalbey/206/Assign/HowToTestCase.html>