Politecnico di Milano

Software Engineering 2 Project

MeteoCal

Authors:
Andrea Celli
Stefano Cereda

December 11, 2015

Part I
Introduction

1 Purpose

This document introduces the general functionalities of MeteoCal. This project is developed for the Software Engineering 2 course held at Politecnico di Milano. The intended audience is composed by people who want to organize their activities in a calendar and manage them according to the weather forecast. Individuals that will participate actively in the project are Software Engineering students for the moment, and in the future anyone wishing to use such a system. The main functionalities of MeteoCal will be:

- An on-line calendar.
- A system to manage activities according to weather forecast.

2 Scope

The software product that will be delivered is MeteoCal, a web application intended to be used by people to schedule their appointments and rearrange them based on weather forecasts. The main objectives of MeteoCal are:

- Allow a user to manage (create, delete or update) his events.
- Allow a user to invite other users to his events and allow the invited users to either accept or decline the invitation.
- In case of bad weather the system has to notify all the event's participants one day in advance
- Allow a user to define his calendar as public or private, if the calendar is public the other users will be able to see when the user is busy.
- Allow a user to define his events as public or private, if an event is public other users will be able to see its details.
- Three days before the event in case of bad weather the system will propose to the event creator the closest sunny day. The user will be able to move the event to the proposed day.

MeteoCal will provide general functionalities for managing:

- Users: MeteoCal will manage personal data of the users. MeteoCal will manage registering, logging in/out and the modification of personal data.
- Calendar: MeteoCal will manage a calendar for each user. User will be able to create, update and delete an event and to see other people's events. MeteoCal will also manage event invitation and notifications for the event's update.

- Weather: MeteoCal will manage weather forecasts and send notifications to event's participants one day in advance in case of bad weather. It will also have to propose an alternative schedule to the event creator with three day of advance.

MeteoCal will have the following limitations (probably developed in future versions):

- Synchronization with other calendars: MeteoCal will not support the import/export of user's calendar.
- Multiple calendars: a user will have exactly one calendar.
- In case of bad weather, MeteoCal will ask if the proposed schedule is fine only to the event creator, and not to all the event participants.
- Avoid events conflicts: MeteoCal will not check if a new event overlaps with an existing one.
- Periodical weather updates: MeteoCal will not send a periodical update of the weather forecast.

MeteoCal will have the following goals:

G1: Allow the registration of new users

G2: Allow users to view, create, update and delete events in their calendar

G3: Allow users to invite other users to their events

G4: Allow invited users to either accept or decline the invitation

G5: Allow users to see other users public calendar

G6: Allow users to see other users public events details

G7: Send a notification to all the participants one day in advance in case of bad weather

G8: Propose an alternative schedule to the event creator three day in advance in case of bad weather

G9: Notify all the event's participant in case the creator changes some details

G10: Allow users to modify their data

3 Definitions and acronyms

3.1 Definitions

- Calendar: a calendar is the agenda of an user
- Event: a task that a user has into his calendar
- Registered user: a user that has created an account on MeteoCal
- Logged user: a registered user that has performed the login process

- Unlogged user: either a non registered user or a registered user that is logged out of the system
- Participant: a participant to an event is either its creator or an invited user who accepted the invite
- Bad weather alert: the notification send to the user with one day of advance if weather forecasts for outdoor events on the next day are bad
- Date changed notification: the notification send to every participant if the event creator change the event date
- System: the MeteoCal system

3.2 Acronyms and abbreviations

- MeteoCal: Meteorological Calendar
- G: Goal
- JVM: Java Virtual Machine
- JEE: Java Enterprise edition
- DBMS: Database management system
- AS: Application server
- FR: Functional requirement
- NFR: Non-functional requirement
- BWA: Bad weather alert
- DCN: Date changed notification

4 Reference documents

- Alloy model file: `./MeteoCal.als`

5 Overview

This document is organized as follows:

- Part 1, Introduction: provides a synopsis of the software product to be developed.
- Part 2, Overall Description: describes the general factors that affect the software product and its requirements.
- Part 3, Specific Requirements: contains the artifacts generated by the analysis. It describes all of the software requirements to a level of detail sufficient to be externally perceivable.
- Part 4, Appendixes: provides supporting information about how the alloy model contributed to the requirement analysis and analysis model.

Part II
Overall description

This section does not describe specific requirements, but puts the product into perspective and provides a background for specifying concrete requirements in the next section of this document.

6 Product perspective

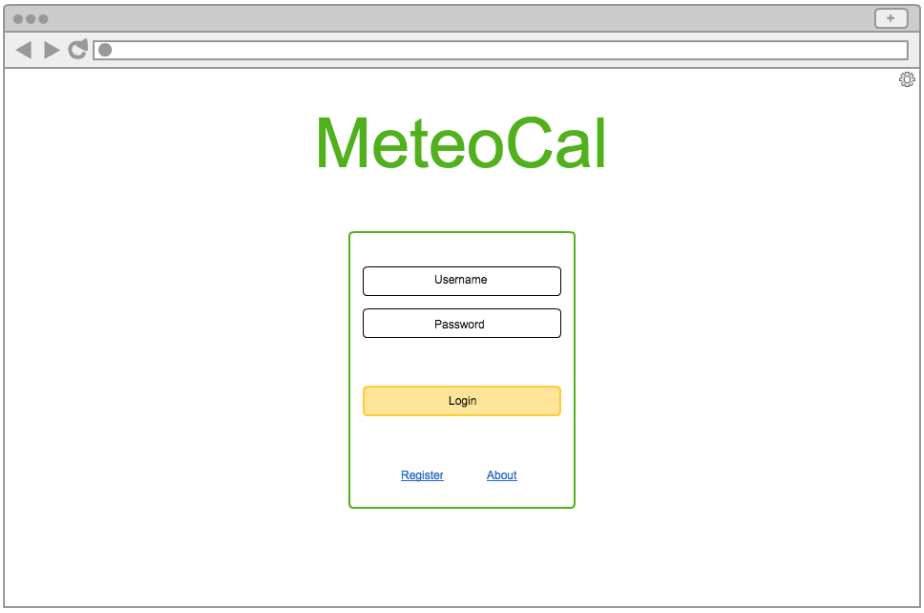
The software product is a complete self-contained system and it is not part of any other larger system. However in the future it may offer external interfaces to other calendars.

6.1 System interfaces

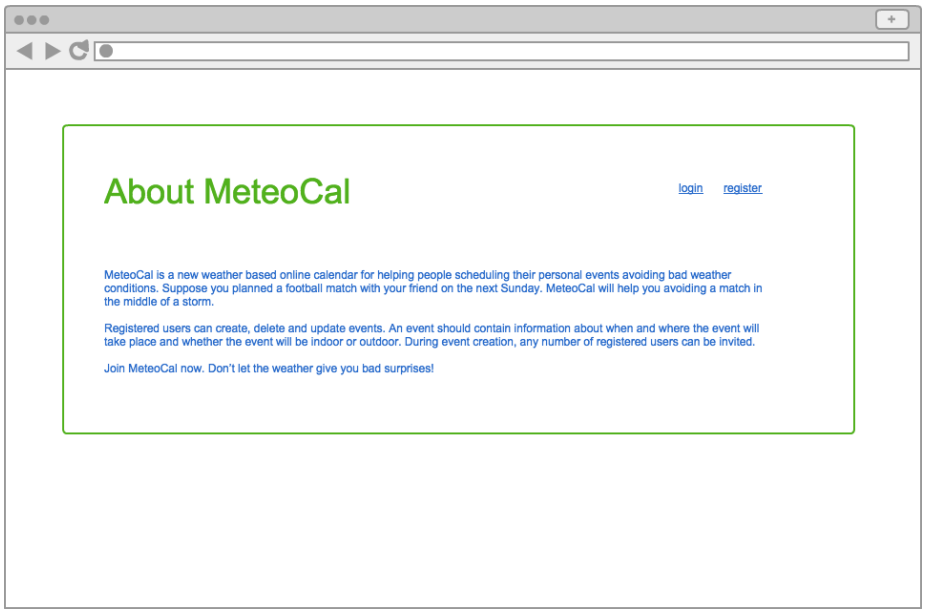
The software product does not provide any external interface.

6.2 User interfaces

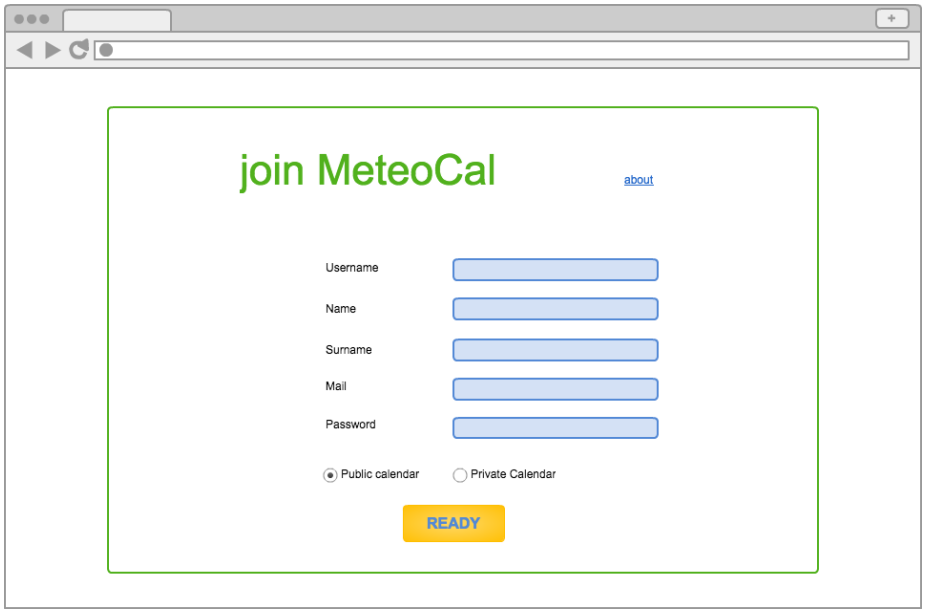
The software product will present the following page layouts as the user interface. These page layouts offer a minimalistic approach to design and navigation. This is the MeteoCal homepage where the user has to login with his data:



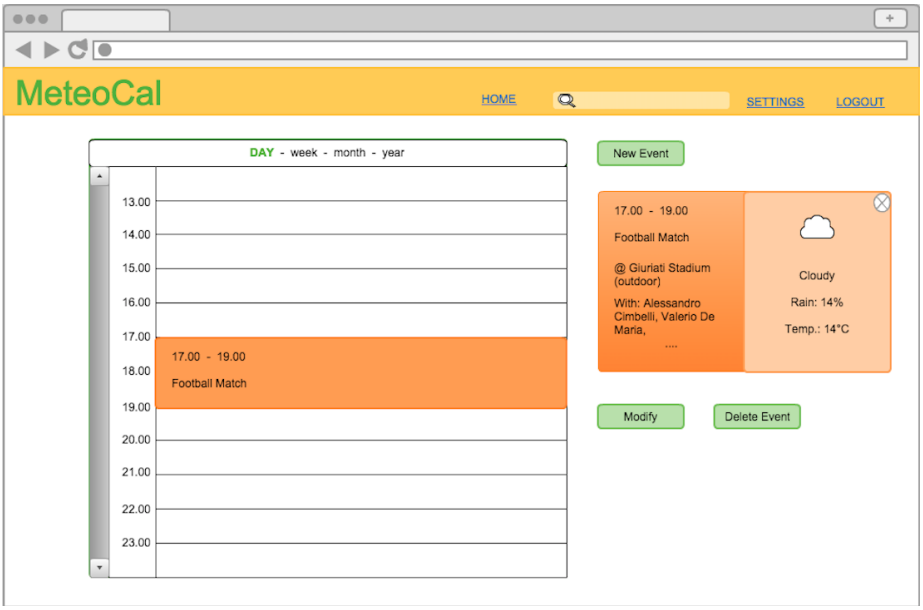
The user can also navigate to the “about” page:



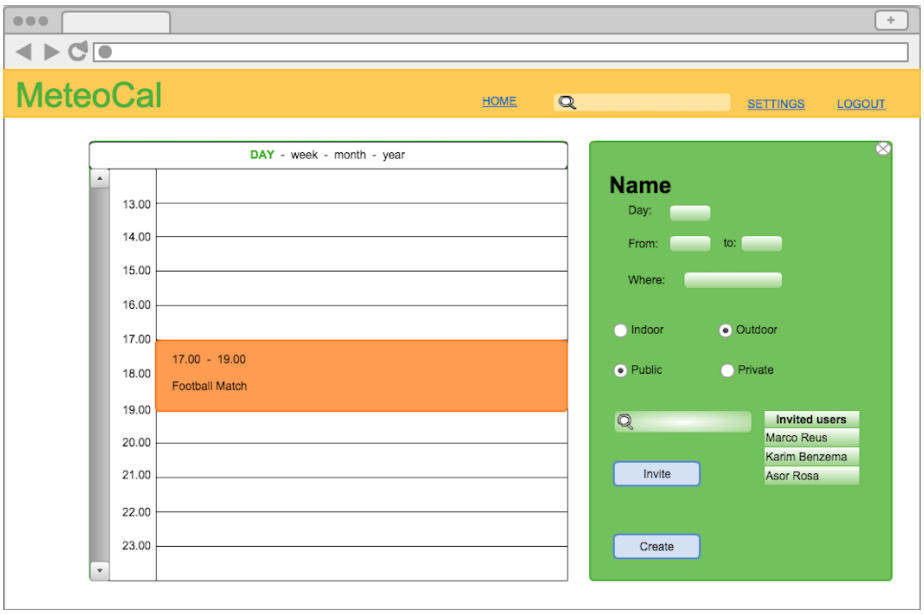
Where a brief description of the MeteoCal service is provided. From both the "homepage" and the "about" pages the user can navigate to "register" page:



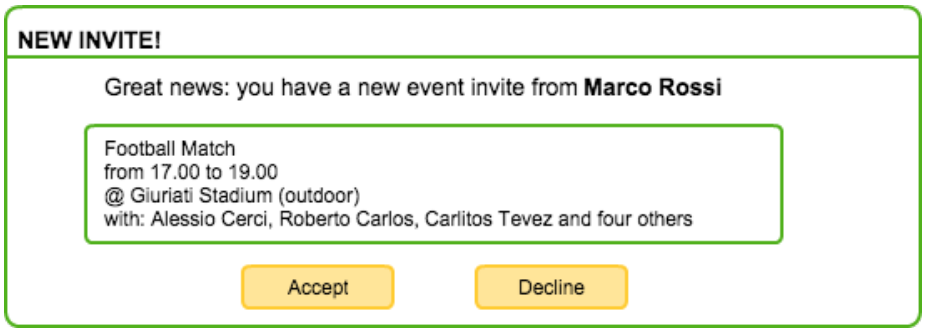
Where new users can register to the system by submitting their data. Once logged the users see the "logged user homepage":



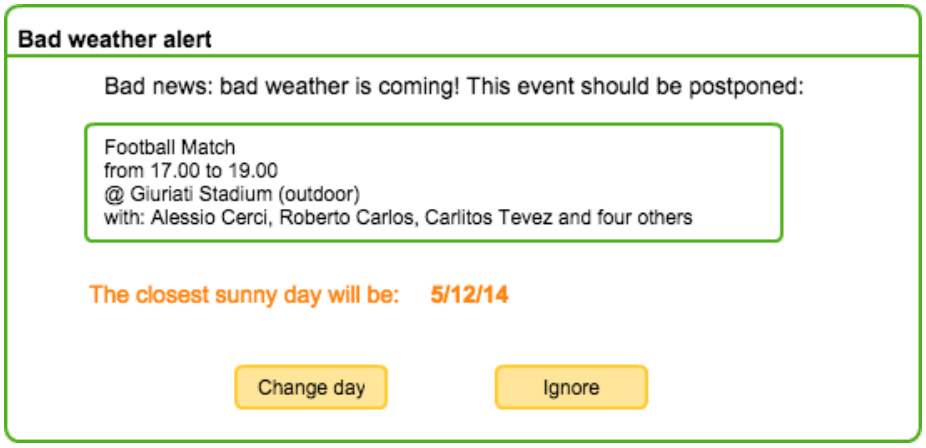
Where they can explore their calendar (on the left) and event details (on the right) by clicking on them on the calendar. They can also choose to create a new event or modify or delete an existing one. If the user chooses to delete the selected event the system will ask for a confirmation. If the user chooses to modify the selected event or create a new one the system will provide the following page:



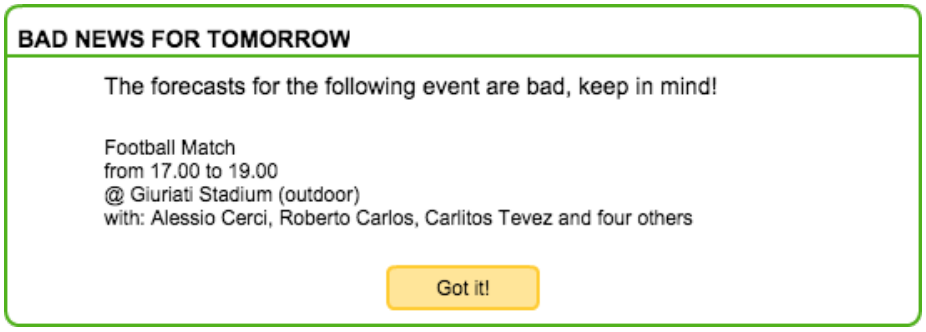
Where he can submit the event details. When the user logs into the system he receives the pending notifications (if any). This is the notification of an event invite:



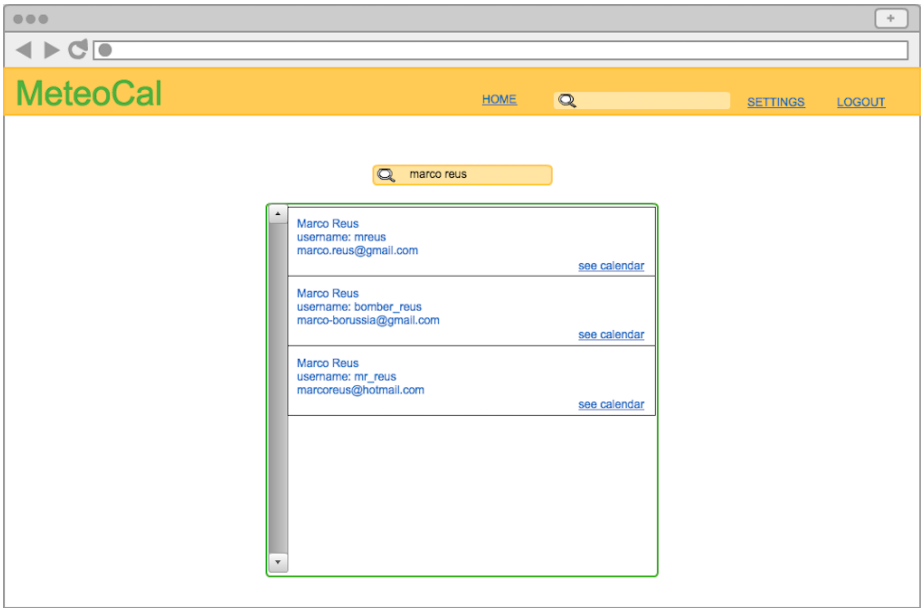
Where the user can see the event details and choose to accept or decline the invitation. This is the notification of bad weather:



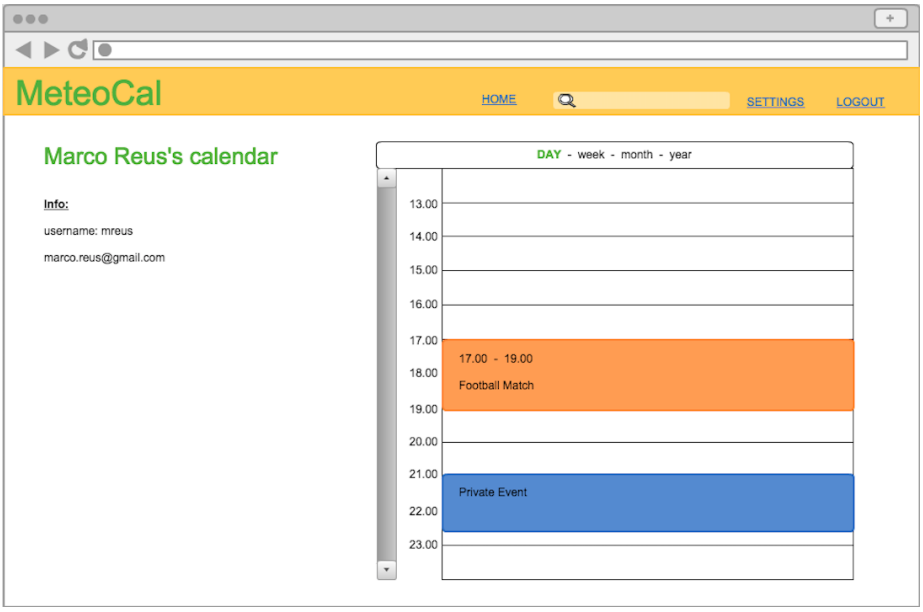
This image is the one presented according to G8 (i.e. the one that proposes a different schedule three days in advance). The notification of G7 (a simple notification of bad weather) is similar except that it does not propose an alternative schedule:



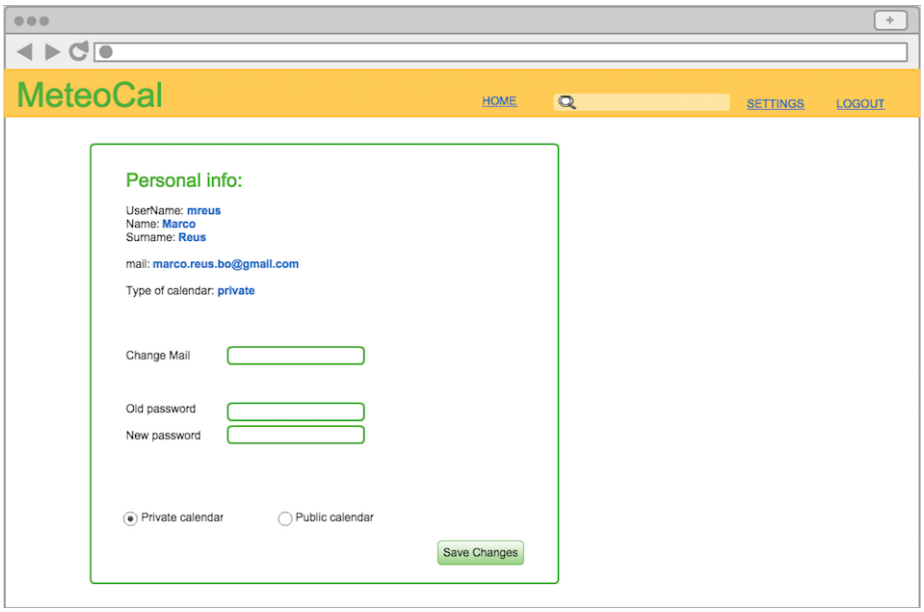
From the "logged user homepage" the user can also search another user by his name, surname or username then the system provides a list of possible matches:



Clicking on "see calendar" will bring the user to the following page:



Where the user can explore the schedule of the chosen user (if he has a public calendar). He will also be able to see public event details by clicking on them. From the "logged user homepage" one can also click on "settings" and navigate to the following page:



Where he can modify his data. From all the pages, by clicking on "logout", the user will log out of the system and return to the MeteoCal homepage.

6.3 Hardware interfaces

The software product does not provide any hardware interface.

6.4 Software interfaces

6.4.1 Database management system

- Name: MySQL
- Mnemonic: MySQL
- Specification number: Community Server
- Version number: 5.6.21
- Source: `http://dev.mysql.com/downloads/mysql/`

6.4.2 Application server

- Name: GlassFish Server
- Mnemonic: GlassFishAS
- Specification number: Open Source Edition
- Version number: 4.1
- Source: `https://glassfish.java.net/download.html`

6.4.3 Operating system

The software product will run on any operating system that supports the JVM and the DBMS and AS described above.

6.5 Communication interfaces

Protocol	Port	Service
TCP	80	World Wide Web
TCP	3306	MySql (only if it is on a different physical server)

6.6 Memory

The minimum memory requirements are:

- Primary memory: 2GB+
- Secondary memory: 40GB+

6.7 Operations

A user can interact with the system as a functional user (unregistered or registered). For all the users, their functional operations are described in the product functions section.

6.8 Site adaptation requirements

The software product requires the following in order to run:

- JVM
- AS
- DBMS
- Primary memory required space
- Secondary memory required space

Users are required to have installed any of the following web browsers: IE6.0+, FF10+, Chrome 20+.

7 Product functions

This section provides a summary of the major functions of the software product.

7.1 General requirements

We have identified 3 main general requirements:

- Managing users
- Managing calendars

- Managing weather forecasts

The functional and non-functional requirements are defined and explained in detail in the following subsections.

7.1.1 Managing users

Functional requirements:

FR 1: Register to system

FR 2: Login

FR 3: Logout

FR 4: Modify password

FR 5: Recover password

FR 6: Update personal data

Non-functional requirements:

NFR 1: User password must be stored securely

NFR 2: System must support high number of users

7.1.2 Managing calendars

Functional Requirements:

FR 1: Add a new event

FR 2: Modify an existing event

FR 3: Delete an existing event

FR 4: View your own schedule

FR 5: View the details of your own event

FR 6: Send an invitation to other users

FR 7: Reply to an invitation

FR 8: See the schedule of other users if their calendar is public

FR 9: See the details of other user's public events

FR 10: Receive a notification when the event details changes

7.1.3 Managing weather forecasts

Functional requirements:

FR 1: Send a notification the day before an event in case of bad weather to all the event's participants

FR 2: Propose an alternative schedule three days before an event in case of bad weather to the event creator

FR 3: Show the weather forecasts for the scheduled events

Non-functional requirements:

NFR 1: The displayed forecasts should be updated every 24 hours

NFR 2: The system has to interface with a meteo service to collect forecasts

8 User characteristics

Intended user should meet the following characteristics:

- Knowledge in using a browser

9 Constraints

The following constraints apply to the software product:

9.1 Regulatory policies

The software product does not have to meet any regulatory policies.

9.2 Hardware limitations

The software product does not have any hardware limitations.

9.3 Software limitations

The system has to be developed using the Java EE platform. The business logic must be implemented using EJBs.

9.4 Interfaces to other applications

The software product has to interface with one Meteorological service to get forecasts.

9.5 Parallel operation

The system has to support simultaneous operation performed by different users. The most frequent situation to handle will be users simultaneously consulting and updating their own calendars.

9.6 Audit functions

The software product does not perform any audit.

9.7 Control functions

The software product does not control any device or any other system.

9.8 Higher-order language requirements

The software product requires basic knowledge of HTML, Java and JEE technologies.

9.9 Signal handshake protocols

The software product does not manage any handshake protocol.

9.10 Reliability requirements

The software product does not require any specific requirements to perform and maintain its functions under normal operation.

9.11 Criticality of the application

The software product requires proper support for concurrent users.

9.12 Safety and security considerations

User data has to be stored and managed in a secure way.

10 Assumptions and dependencies

The requirements in this document are grounded on the following assumptions:

- The JVM is already installed on the OS
- Users have a decent and acceptable Internet connection.

11 Apportioning of requirements

Future releases of the software product may provide support for:

- Synchronization with other calendars
- Multiple calendars
- Avoid conflicting events
- Periodical weather updates

Part III
Specific requirements

12 External interface requirements

12.1 User interfaces

The storyboard in figure 1 describes the work flow to get from the login page to the exploration of the calendar of another user.

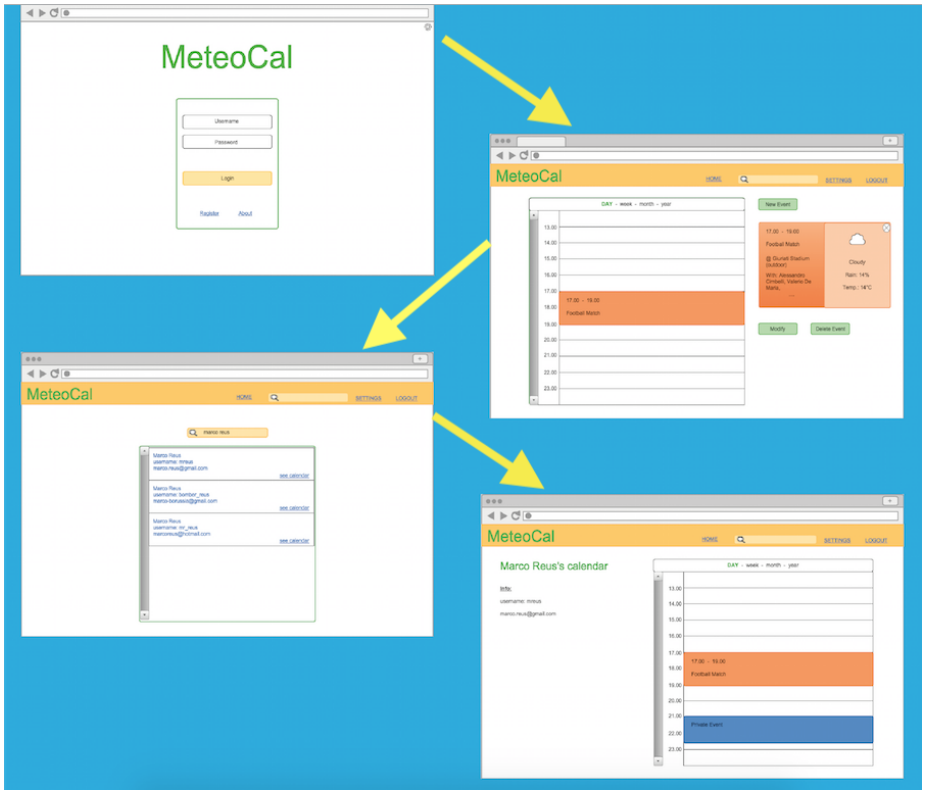


Figure 1: Storyboard from login to another user's calendar

12.2 Hardware interfaces

The software product does not provide any hardware interfaces.

12.3 Software interfaces

The software product does not provide any software interfaces.

12.4 Communications interfaces

The software product does not provide any communications interfaces.

13 Functional Requirements

13.1 Scenarios

13.1.1 Registering in the system

Registering in the system	
Code	SC001
Description	Describing how a user registers in the system
Goal	• G1: Allow the registration of new users
Assumptions	1. User is not registered
Valerio wants to find a way to manage the schedule of his outdoor workouts, he asks a friend and becomes aware of the existence of MeteoCal. He navigates to the MeteoCal website and decides to register, so he clicks on "Register" button. The system provides him a form to be filled with mandatory information: his username, password, email address, name and surname. He also has to specify whether his calendar is public or private. At the end he can press the "Save" button or the "Cancel" button. He saves and then logs out since he has to go out running.	

13.1.2 User logs into the system

User logs into the system	
Code	SC002
Description	Describing how a user logs into the system
Goal	• G2: Allow users to view, create, update and delete events in their calendar
Assumptions	1. User is registered 2. User correctly inserts the data
Valerio is back from his workout and wants to try his new calendar. He opens a web browser and navigates to the website of MeteoCal, where he clicks on a "Login" button. The system provides him a form where he inserts his username and his password, then he clicks on a "submit" button. Now he sees a page with his calendar. He is interrupted by a phone call.	

13.1.3 User creates an event

User creates an event	
Code	SC003
Description	Describing how a user creates a new event
Goal	• G2: Allow users to view, create, update and delete events in their calendar
Assumptions	1. User is logged
Valerio finishes his phone call and returns to the MeteoCal website. He decides to schedule a running session for the next week: he clicks on an "Add event" button. The system provides a form where he is asked to provide a name for his appointment, a date, the starting and ending time. He is also asked to specify whether the event will take place outdoor or indoor. He specifies that it has to be a private event. Then he clicks on a "Save event" button and closes the website.	

13.1.4 User modifies an event

User modifies an event	
Code	SC004
Description	Describing how a user modifies an existing event
Goal	• G2: Allow users to view, create, update and delete events in their calendar
Assumptions	1. User is logged 2. User has at least one event in his calendar
Valerio suddenly realizes that he scheduled the workout on a wrong day, so he goes back to the MeteoCal website and logs into the system. The system provides him a calendar where he can sees his appointments, he clicks on the interesting event and appears a description of the event and two buttons: "modify" and "delete". He clicks on "modify" and the system provides a form with the data of the event, he modifies the day and clicks "save". Now on the calendar the event is scheduled on the new date.	

13.1.5 User views his calendar

User views his calendar	
Code	SC005
Description	Describing how a user views the details of his appointments
Goal	• G2: Allow users to view, create, update and delete events in their calendar
Assumptions	1. User is logged 2. User has at least one event in his calendar
Valerio forgot when he has to run, so he decides to check on MeteoCal. After the login the system provides an overview of his calendar, with a visual representations of his appointments, he clicks on the desired activity and the system visualize the details of the event: when it is scheduled, where it will take place (the place and if it's indoor or/outdoor) and the invited users. The systems also shows the forecast for the selected event.	

13.1.6 User deletes an event

User deletes an event	
Code	SC006
Description	Describing how a user deletes an existing event
Goal	• G2: Allow users to view, create, update and delete events in their calendar
Assumptions	1. User is logged 2. User has at least one event in his calendar
Valerio broke a leg during his last workout, so he decides to remove the scheduled running from MeteoCal. From his homepage (after the log in) he clicks on the event and from the new page he clicks on a "delete" button, the system asks for a confirmation, and after the confirm the event disappears from the calendar.	

13.1.7 User invites another user to his event

User invites another user to his event	
Code	SC007
Description	Describing how a user invites another user to his event
Goal	• G3: Allow users to invite other users to their events
Assumptions	1. User is logged 2. User is visualizing the details of one of his events 3. User doesn't try to invite himself 4. User correctly inserts the name of the other user
While visualizing the details of his scheduled running workout Valerio has a great idea: inviting to the workout his friend Ilario. Valerio clicks on an "Invite" button and the system provides a form where he can insert the username of the desired user, he inserts the username of his friend and clicks on a "Send invitation" button. While waiting for the answer he sends an sms to his friend.	

13.1.8 Invited user accepts

Invited user accepts	
Code	SC008
Description	Describing how a user accepts an invitation
Goal	• G4: Allow invited users to either accept or decline the invitation
Assumptions	1. User is registered 2. User has received an invite 3. User has not logged to system since he has received the invite
Ilario is a friend of Valerio and is a new user of MeteoCal. He receives a sms by Valerio and decides to check if he has received the event invitation. He logs into MeteoCal and the system presents him a notification of the invitation where he can see the event details. Since he likes the idea, he clicks on an "Accept" button. The event appears on his calendar.	

13.1.9 User declines an event invite

User declines an event invite	
Code	SC009
Description	Describing how a user declines an event invite.
Goal	• G4: Allow invited users to either accept or decline the invitation
Assumptions	1. The user is registered 2. The user has been invited to another user's event
Valerio has just returned home after a long training session. After a shower, he turns on his computer to check what are his appointments for the following day. In order to do so he opens the browser, navigate to the MeteoCal website and login into the system. Valerio starts to browse his events when an invite notification appears in the middle of the page. Leonardo, his south-american friend, invited him to try a Samba lesson later in the evening, at 21.30. Valerio feels very tired because of his training so he clicks on the "decline" button. The notification disappears and Valerio finishes to check his appointments.	

13.1.10 User search the calendar of another user

User search the calendar of another user	
Code	SC010
Description	Describing how a user can search the calendar of another user.
Goal	• G5: Allow users to see other users public calendar
Assumptions	1. Both user are registered 2. The second user's calendar is public
Valerio has changed his mind and he decided that he wants to take a dance lesson with Leonardo. Unfortunately it's already 21.45 and the lesson is already started. Valerio decides to check whether Leonardo will go to other lessons during the week. He opens his MeteoCal homepage, writes Leonardo Tanque (his friend's name) in the search bar in the top right and press enter. He could have used also Leonardo's username but, at the moment, it didn't come to his mind. After performing the search the system shows a page displaying all the results. There are a lot of users called Leonardo Tanque. Valerio scrolls down the bar on the left until he finds his friend by recognizing his email address (which is displayed along with the name and the userName). Valerio clicks on the "see calendar" link on the right of the result and gets to Leonardo's calendar.	

13.1.11 A user looks through the calendar of another user

A user looks through the calendar of another user	
Code	SC011
Description	Describing how a user navigate through another user's calendar
Goal	• G5: Allow users to see other users public calendar • G6: Allow users to see other users public events details
Assumptions	1. Both user are registered 2. The second user's calendar is public
Valerio has reached Leonardo's calendar. He selects the week view. Leonardo has some events programmed for next Wednesday. He has a private event going from 13.00 to 15.00 and a public event called "Samba" going from 21.30 to 22.30. The calendar shows only public events name. Valerio wants to be sure that "Samba" is a dance lesson so he clicks on the event to get further details. The event is public so a window showing all the event information appears on the right. Valerio sees that it will take place indoor in the "Samba dance school" so he has just found what he was looking for. Now he's quite curious about what Leonardo has to do from 13.00 to 15.00. He tries to see the details but the event is private and, when he clicks on it, nothing happens.	

13.1.12 The user receives the bad weather alert

The user receives the bad weather alert	
Code	SC012
Description	After entering MeteoCal the user receives a bad weather alert with one day of advance from the event
Goal	• G7: Send a notification to all the participants one day in advance in case a bad weather
Assumptions	1. The user is registered 2. The user is going to take part in an outdoor event on the following day 3. The weather forecasts for the following day are bad.
It's Friday and Valerio decides to check his programs for the weekend. His friend Guglielmo invited him to a trekking trip on the mountains near Como. Valerio is worried about the weather conditions, he heard that it might rain. He logins and, as soon as he gets to his homepage, a notification appears. It warns Valerio that the weather forecasts for the trip (planned for the next day) are bad. There's a high chance of rain. Thus Valerio will put a raincoat in his rucksack.	

13.1.13 The system propose a sunny day to the user

The system propose a sunny day to the user	
Code	SC013
Description	If three days before the event its weather forecasts are bad, the system warns the creator and suggests to him to change the date to the closest sunny day. The user can move the event or leave it on the scheduled day.
Goal	• G8: Propose an alternative schedule to the event creator three day in advance in case of bad weather
Assumptions	1. The user is registered 2. The user created an event 3. There are bad weather forecasts for the event 4. The event will take place after three days
Valerio planned a trip to the seaside with his friend Alessandro and created the related event on MeteoCal. Three days before the event Valerio wants to check whether he invited his friend to the event or not. He logins in MeteoCal and, as soon as he enters the homepage, a notification appears. It says that the weather forecasts for the seaside trip are bad. It also points out that the closest sunny day will be Wednesday. Valerio can ignore the message ("ignore" button) or move the event to Wednesday ("move" button). He thinks about it and finally he decides to click on the move button. The event is postponed and scheduled for Wednesday, the other event details (time, place ecc..) remain the same.	

13.1.14 A user gets notified that an event date has been changed

A user gets notified that an event date has been changed	
Code	SC014
Description	When the creator of an event changes its details all the participants (but not the creator) are notified.
Goal	G9: Notify all the event's participant in case the creator changes some details
Assumptions	1. The user is registered 2. He accepted the invite to an event 3. The creator changed the date of the event for bad weather conditions.
Alessandro wants to check the leaving time for his trip to the seaside with Valerio. He logins and access his MeteoCal homepage. While browsing through his events he gets a notification. It states that Valerio (the event creator) chanced the date of the seaside trip. The event is now planned for Wednesday. Alessandro returns to his calendar page.	

13.1.15 A user changes his data

A user changes his data	
Code	SC015
Description	A user accesses his settings page and change his password and his email address.
Goal	G10: Allow users to modify their data
Assumptions	1. The user is registered 2. The user types a valid email account and the right old password
Valerio has just created a new email account on gmail. He decides that he wants to use it in MeteoCal instead of his previous email address. He logs into the system and gets to his homepage. He clicks on the settings button on the top right of the page. The settings page shows his personal details. He enters his new email in the appropriate box. Valerio wants to be sure that his private events are safe so he decides to change the password. He types in the appropriate boxes his old password and the new one. Finally Valerio presses the save button. His information are updated by system and the settings page is changed consequently.	

13.1.16 The system updates the meteo forecasts

The system updates the meteo forecasts	
Code	SC016
Description	MeteoCal asks an external meteo service for updated weather forecasts for all the scheduled events.
Goal	• G7: Send a notification to all the participants one day in advance in case of bad weather • G8: Propose an alternative schedule to the event creator three day in advance in case of bad weather.
Assumptions	1. The systems has some user events in his database 2. The system has an interface to "speak" with an external weather forecasts service
The system has to update the weather forecasts of some outdoor events because they're obsolete (they haven't been updated for more than one day). To do so MeteoCal ask an external service for the forecasts that he needs and waits for the each answer. When the external service answers the systems updates the events. Users are now able to see the up to date forecasts for their events.	

14 Analysis model

The analysis model represents the core concepts; the diagram in figure 2 introduces the conceptual classes that we have decided to include in the software product.

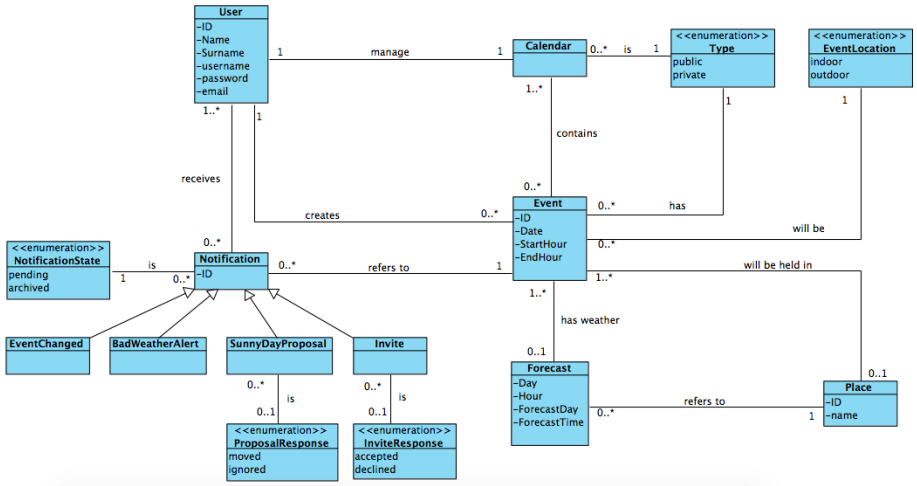


Figure 2: Class diagram

15 State chart model

The following diagrams represents the evolution of some objects in our system.
Figure 3 depicts the evolution of the notification of an event invite.

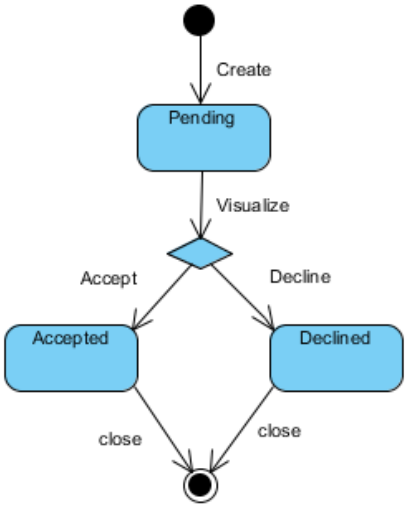


Figure 3: Invite notification state chart

Figure 4 depicts the evolution of the notification sent to the event creator to propose an alternative schedule.

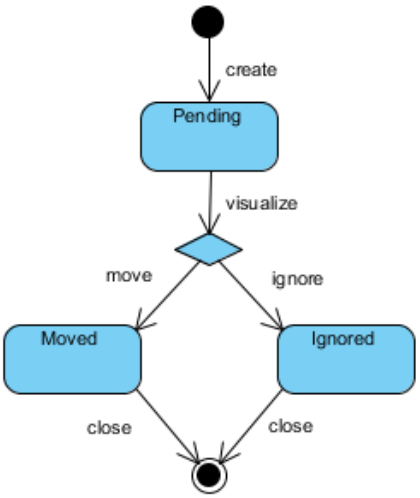


Figure 4: New schedule proposal state chart

Figure 5 depicts the evolution of the notification of bad weather sent to all the event's participants one day in advance. It also describes a notification sent when the event details are changed.

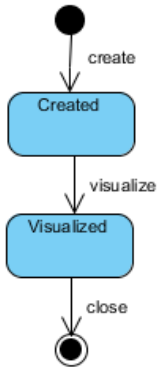


Figure 5: Bad weather and event changed notification state chart

16 Activity model

Since the main goal of the software product is to create new event, in figure 6 we introduce the activity model of the creation of a new event.

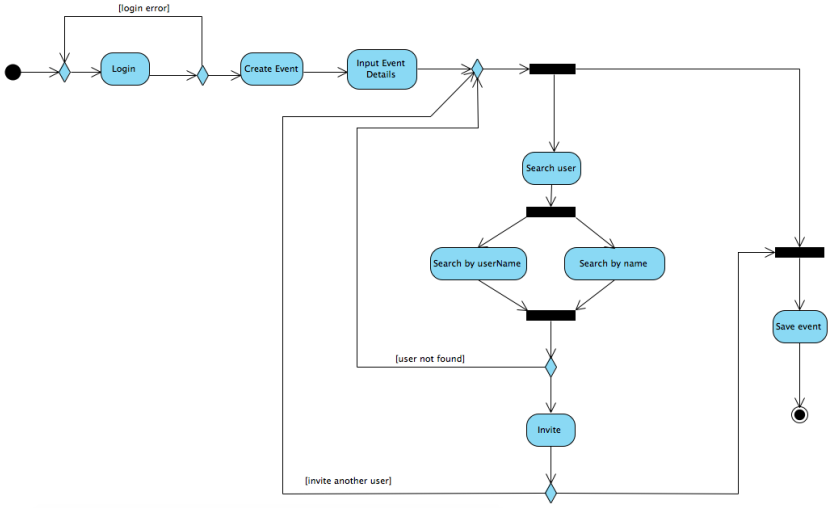


Figure 6: Event creation activity diagram

17 Use case model

Below we separately present the use cases associated with our two actors:

- Unlogged user
- Logged user

17.1 Unlogged user

Figure 7 depicts the use cases of an unlogged user.

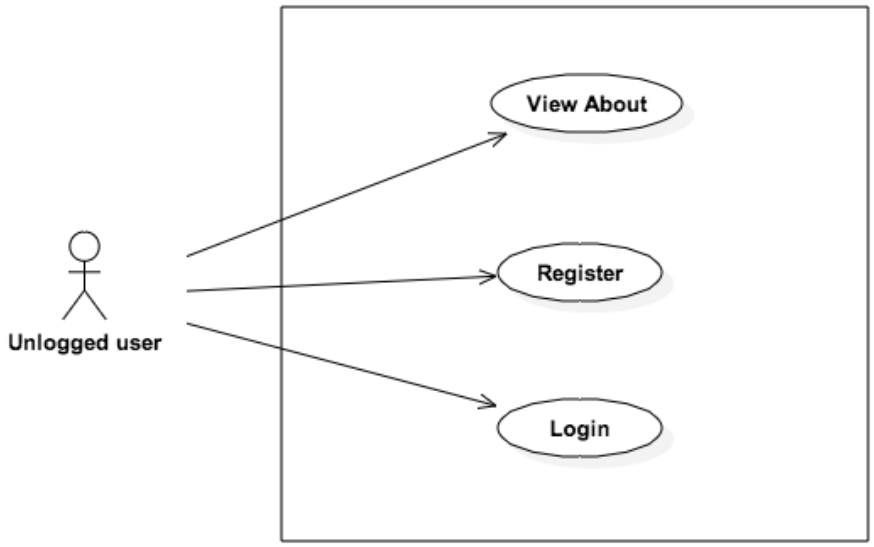


Figure 7: Unlogged user use case

17.1.1 User register to the system

User register to the system	
Code	USC001
Description	Describing how unregistered users can register to the system
Goal	G1: Allow the registration of new users
Assumptions	1. The user is not registered to the system
Actors	1. The unregistered user
Entry condition	The unregistered user navigates to the register page
Exit condition	User data are correctly saved a confirmation is displayed
Flow of events	1. The unregistered user navigates to the MeteoCal registering page 2. The system provides a form to be filled with Username, Name, Surname, Mail, Password 3. The unregistered user fills the form and clicks “Ready” 4. The system saves the new user data and display a confirmation window
Exceptions	• If some data are missing the system shows an error • If the username or the password have already been used by another user the system shows an error
Special requirements	
Nonfunctional requirements	

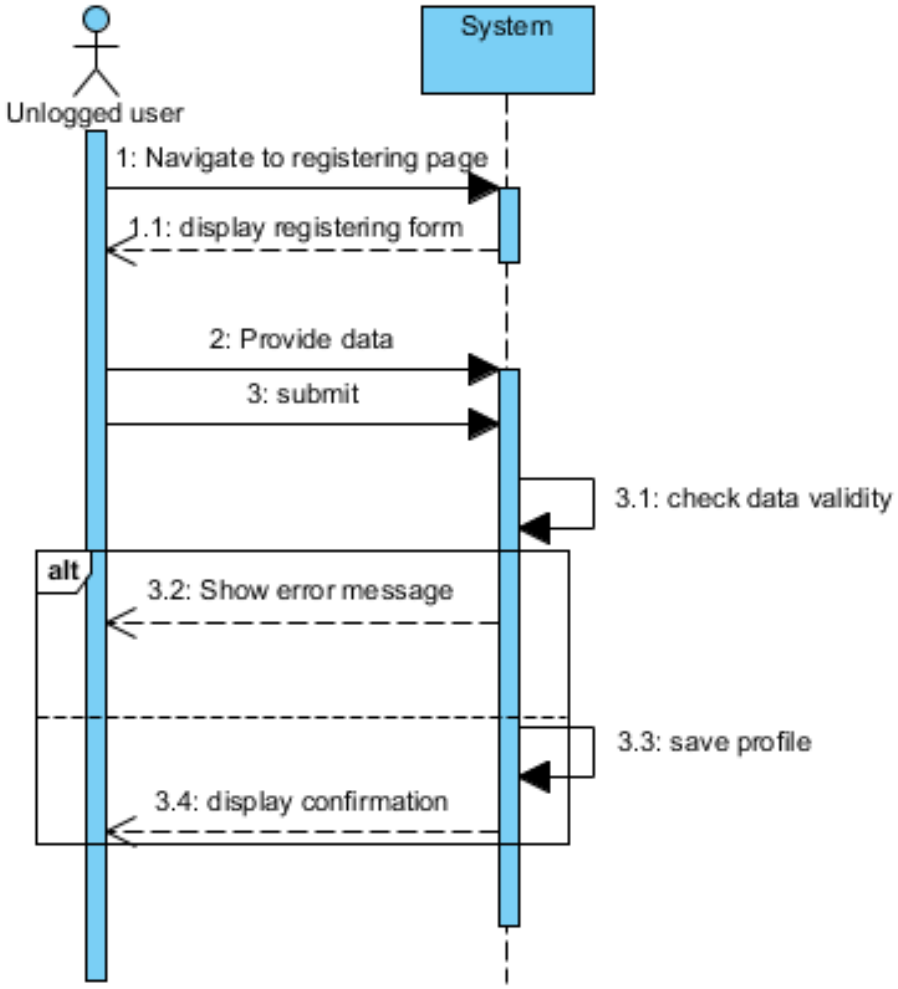


Figure 8: USC001 - User register to the system

17.1.2 A registered user logs into the system

A registered user logs into the system	
Code	USC002
Description	Describing how a registered user logs into the system
Goal	G1: Allow users to create update and delete events in their calendar
Assumptions	1. User is registered to the system 2. User is not logged to the system
Actors	1. The registered user
Entry condition	The user navigates to the MeteoCal homepage
Exit condition	The user is logged into the system
Flow of events	1. The registered user navigates to the MeteoCal homepage 2. The system provides a form to be filled with username and password 3. The user inserts his data and clicks login 4. The system logs the user and redirect the user to the "Registered user homepage"
Exceptions	If the inserted data are not correct the system shows an error message
Special requirements	
Nonfunctional requirements	

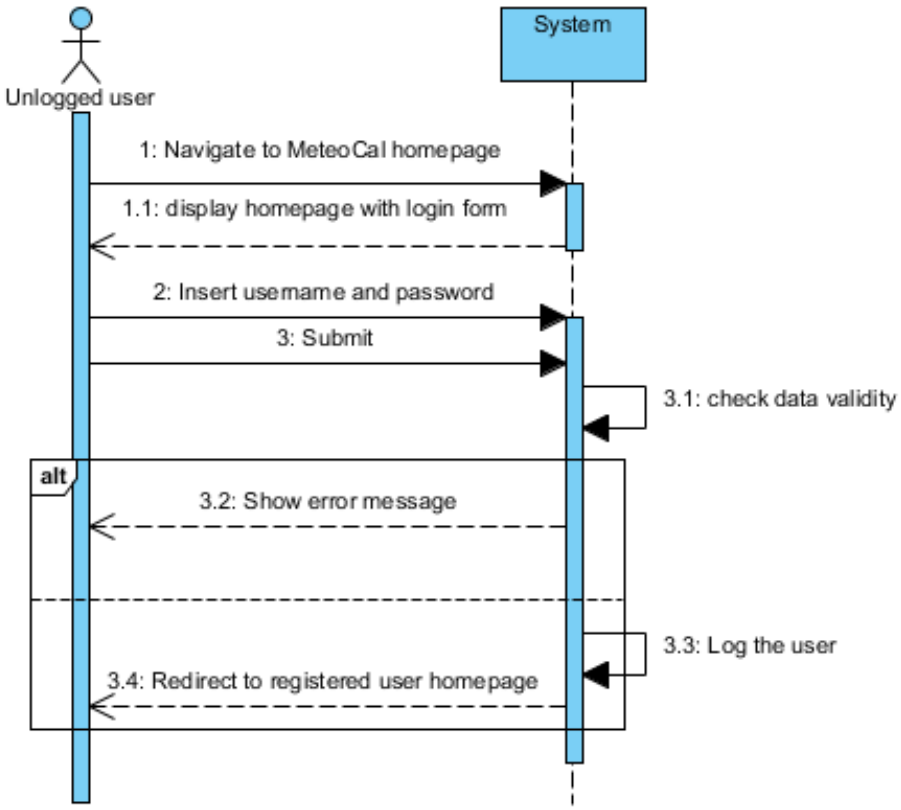


Figure 9: USC002 - A registered user logs into the system

17.2 Logged user

Figure 10 contains the simplified version of the logged user use case, while figure 11 depicts the complete one.

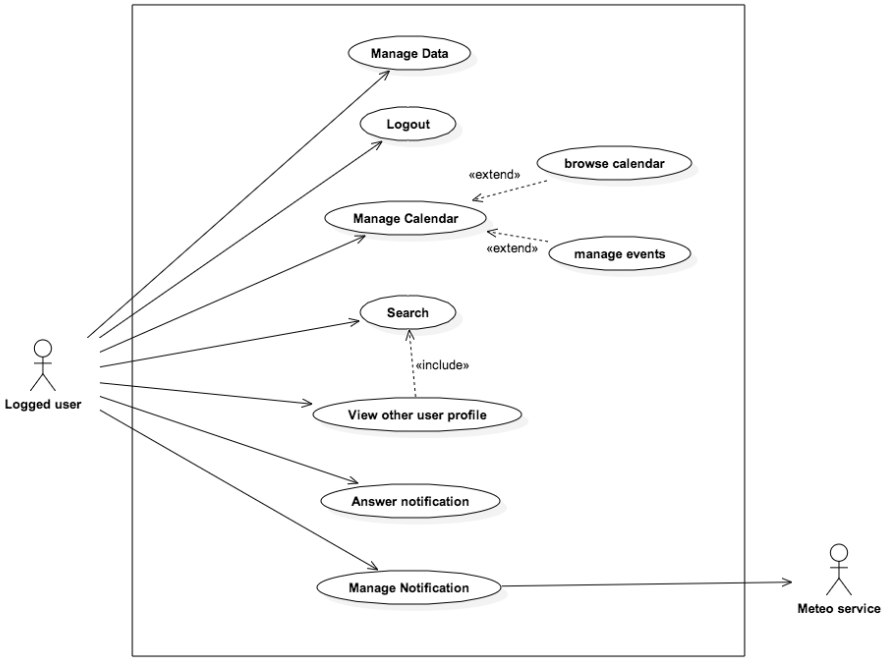


Figure 10: Logged user use case. Simplified

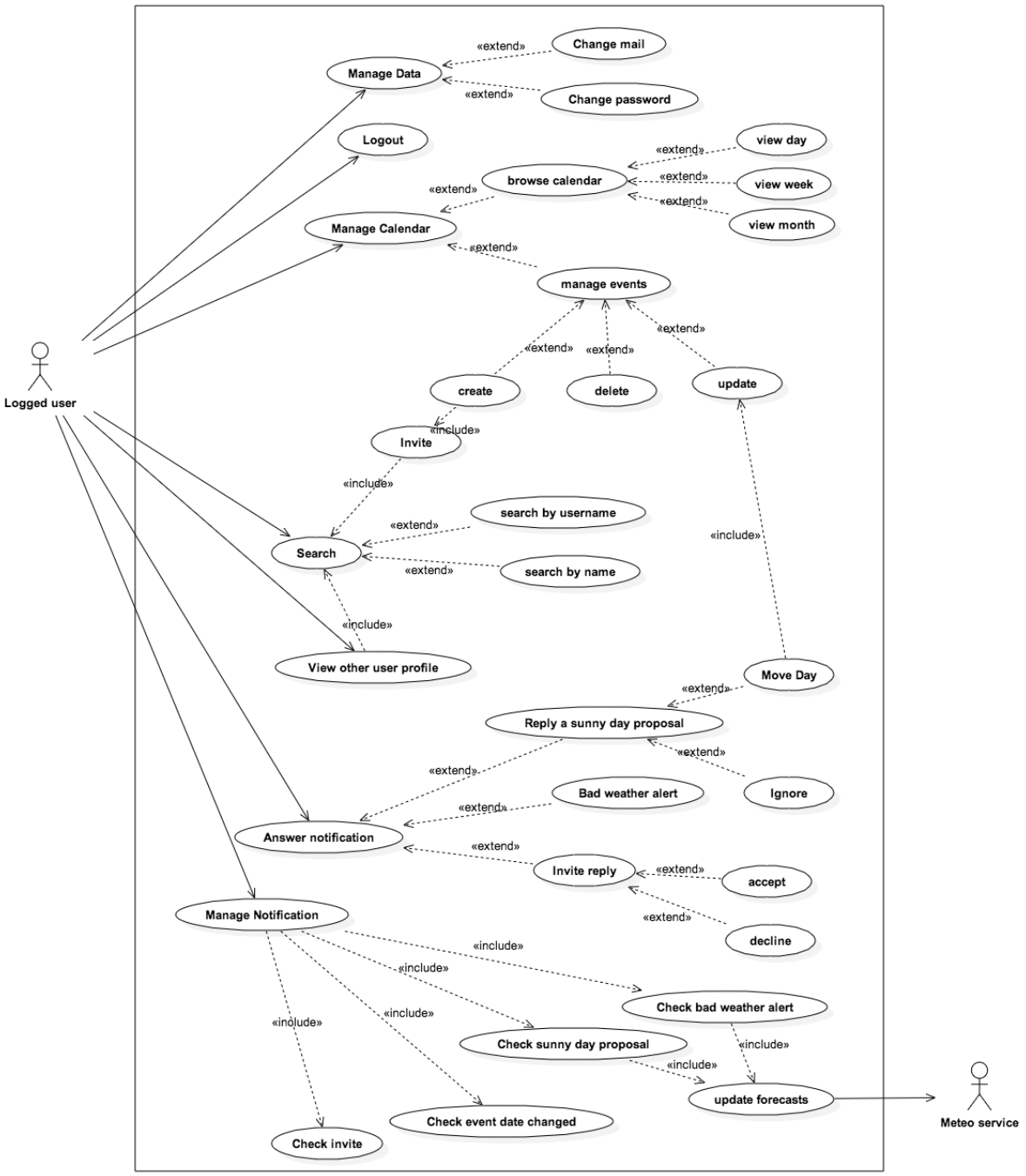


Figure 11: Logged user use case

17.2.1 Logged user creates a new event

Logged user creates a new event	
Code	USC003
Description	Describing how a logged user can create new events
Goal	G2: Allow users to view, create, update and delete events in their calendar
Assumptions	1. User is logged 2. User is visualizing the "registered user homepage"
Actors	1. The logged user
Entry condition	The user clicks on "New event"
Exit condition	The event is created
Flow of events	1. The user clicks on "New event" 2. The system provides a form to be filled with the event data (Day, From, To, Where, Indoor/Outdoor, Public/Private and a possibly empty list of invited users) 3. The user fills the form and clicks on "Create" 4. The system saves the event and shows a confirmation
Exceptions	• If some data is missing the system shows an error • If the beginning hour is before the ending hour the system shows an error • In case of an outdoor event the use must indicate a place recognized by the meteo service
Special requirements	
Nonfunctional requirements	

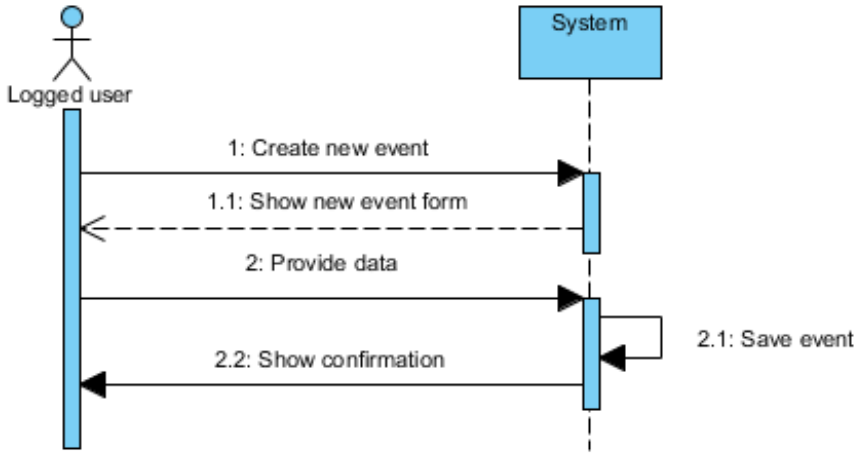


Figure 12: USC003 - Logged user creates a new event

17.2.2 Logged user modifies an existing event

Logged user modifies an existing event	
Code	USC004
Description	Describing how a logged user can modify the data of an event that he have created previously
Goal	G2: Allow users to view, create, update and delete events in their calendar
Assumptions	1. The user is logged 2. The user tries to modify an event that he have created before
Actors	1. The logged user
Entry condition	From the "registered user homepage" the user selects an event and clicks on "Modify"
Exit condition	The system modifies the event data and show a confirmation
Flow of events	1. From his "registered user homepage" the user selects an event 2. The system displays the selected events data and a "Modify" button 3. The user clicks on "Modify" 4. The system provides a form with the old data 5. The user modifies the data and clicks "Save" 6. The system modifies the event data and show a confirmation
Exceptions	• If some data is missing the system shows an error • If the beginning hour is before the ending hour the system shows an error • In case of an outdoor event the use must indicate a place recognized by the meteo service
Nonfunctional requirements	

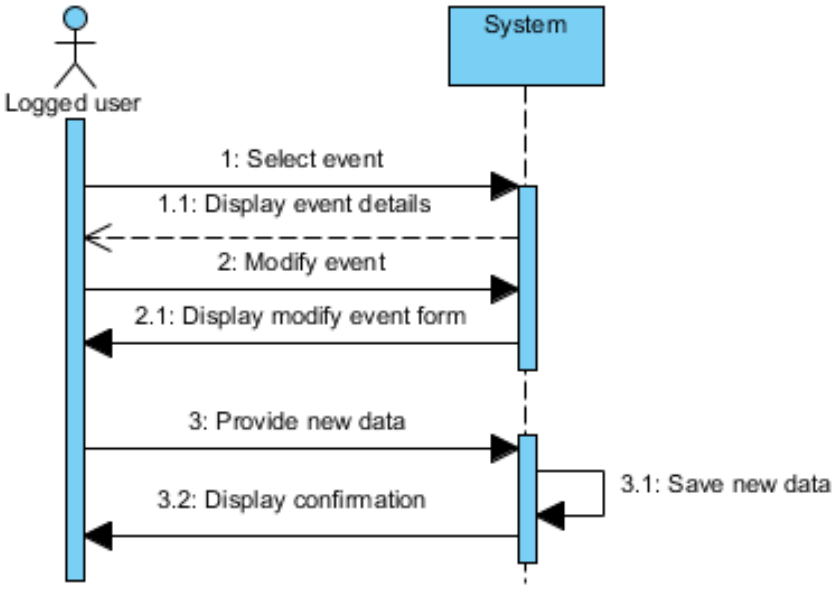


Figure 13: USC004 - Logged user modifies an existing event

17.2.3 Logged user views the details of his events

Logged user views the details of his events	
Code	USC005
Description	Describing how a logged user can view his calendar and event details
Goal	G2: Allow users to view, create, update and delete events in their calendar
Assumptions	1. User is logged 2. User has at least one event in his calendar
Actors	1. The logged user
Entry condition	The user reaches the "registered user homepage"
Exit condition	The user clicks on "X"
Flow of events	1. The user reaches his "registered user homepage" 2. The system shows him a representation of his event, showing the name and the schedule of events 3. The user clicks on an event 4. The system shows him the event details: Starting and ending time, name, place, invited users and weather forecast 5. The user closes the event details by clicking on an "X"
Exceptions	
Special requirements	
Nonfunctional requirements	

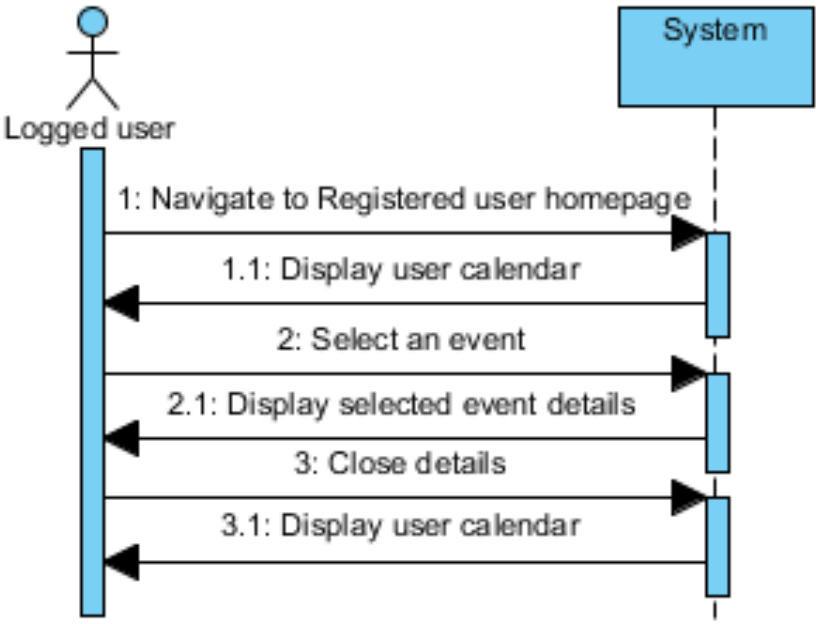


Figure 14: USC005 - Logged user views the details of his events

17.2.4 Logged user deletes an event

Logged user deletes an event	
Code	USC006
Description	Describing how a logged user removes an appointment from his calendar
Goal	G2: Allow users to view, create, update and delete events in their calendar
Assumptions	1. User is logged and visualizing the details of one event in his calendar 2. User has at least one event in his calendar
Actors	1. The logged user
Entry condition	The user clicks on "delete event"
Exit condition	The system removes the appointment from the user's calendar
Flow of events	1. While visualizing the details of an event the user clicks on "delete event" 2. The system removes the appointment from the user's calendar. If the user created the event and there are other users invited the system removes the event also from their calendars. If another user created the event and the current user is only a guest the system removes the event only from the current user's calendar.
Exceptions	
Special requirements	
Nonfunctional requirements	

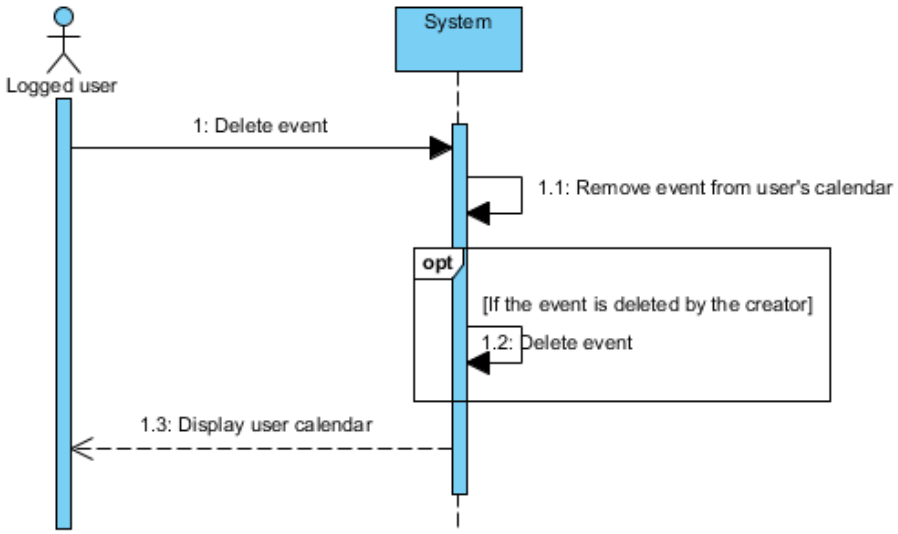


Figure 15: USC006 - Logged user deletes an event

17.2.5 Logged user invites another user to his event

Logged user invites another user to his event	
Code	USC007
Description	Describing how a logged user can invite another user to his events
Goal	G3: Allow users to invite other users to their events
Assumptions	1. User is logged and modifying the details of one of his events 2. User has at least one event in his calendar
Actors	1. The logged user
Entry condition	User starts typying in an “invite” textbox
Exit condition	The system sends an invite notification and shows a confirmation
Flow of events	1. The user starts typing in an “invite” textbox 2. The system searches for an user that matches the typed text by username, name, surname or e-mail and proposes a list of possible matches. 3. The user selects an user and clicks on “invite” 4. The system saves the invitation and shows a confirmation
Exceptions	If there are no possible matches the system shows an error
Special requirements	
Nonfunctional requirements	

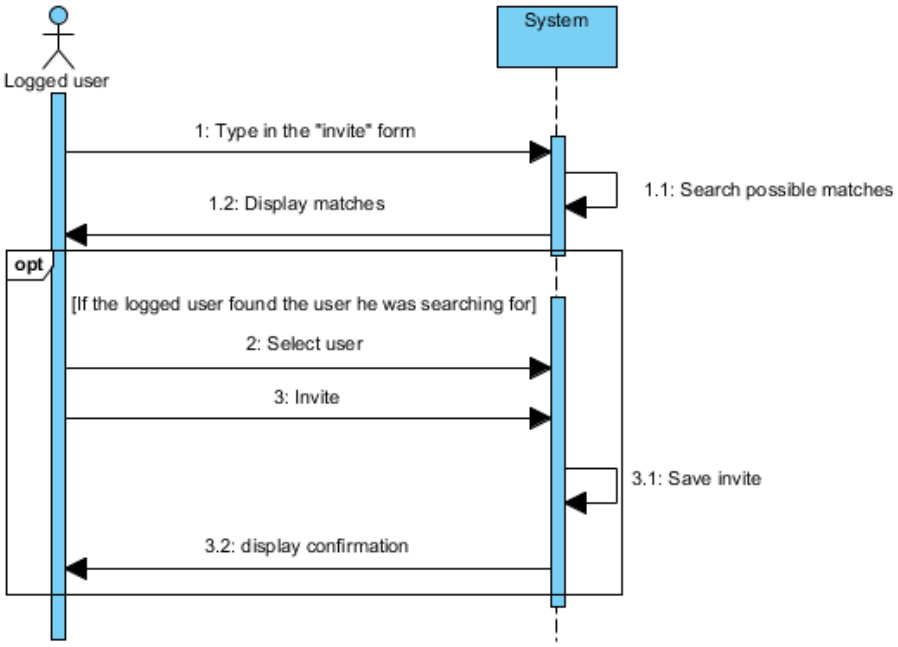


Figure 16: USC007 - Logged user invites another user to his event

17.2.6 Registered user accepts an invitation

Registered user accepts an invitation	
Code	USC008
Description	Describing how a registered user receives the notification of an invite and accepts it
Goal	G4: Allow invited users to either accept or decline the invitation
Assumptions	1. User is registered 2. User has received an invite 3. User has not logged to system since he has received the invite 4. The user accepts the invitation
Actors	1. Registered user
Entry condition	User logs into the system
Exit condition	System adds the user to the list of event's participants
Flow of events	1. The registered user logs into the system 2. The system notifies him that he has received an invitation. It shows the event creator's name and surname, the event name, when and where it will take place and the event's participants. 3. The user clicks on "Accept" 4. The system add the event to the user's calendar an the user to the event's participants.
Exceptions	
Special requirements	
Nonfunctional requirements	

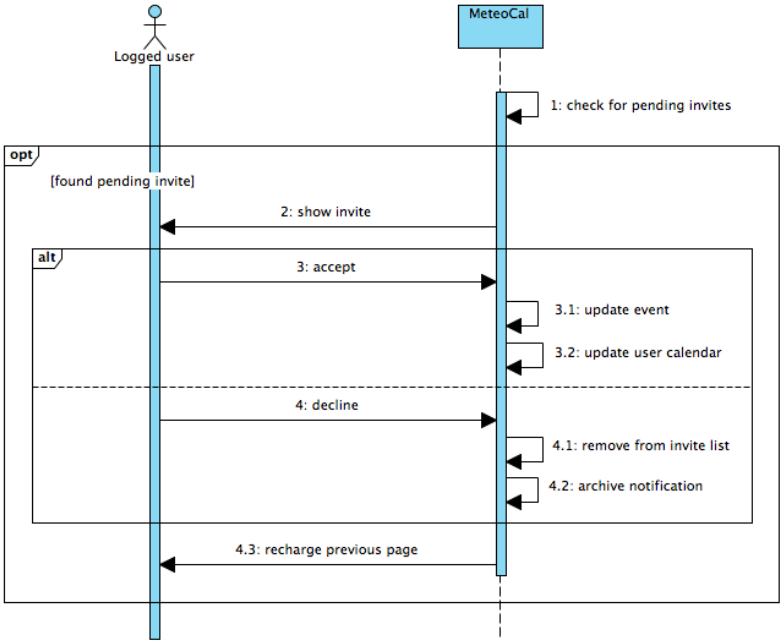


Figure 17: USC008 - Registered user accepts an invitation

17.2.7 Decline an event invite

Decline an event invite	
Code	USC009
Description	A user declines an event invite from another user
Goal	G3: Allow invited users to either accept or decline the invitation
Assumptions	1. the user is registered 2. the user has been invited to another user's event 3. the invited user's already logged in and he's on a MeteoCal page
Actors	1. Two registered users
Entry condition	A registered user has sent an invite to another registered user
Exit condition	The systems has archived the request
Flow of events	1. The registered user B accesses a MeteoCal page 2. As soon as A invites B the systems shows to B a notification 3. B declines the invite using the "decline button" 4. The notification disappears, B returns to his previous activity 5. The system removes B from the list of invited people to the event and archives the invite 6. A no longer sees B among the invited people
Exceptions	• The user exits MeteoCal before after receiving the notification but before clicking on the "got it " button. • The user exits MeteoCal before he receives the notification.
Special requirements	
Nonfunctional requirements	

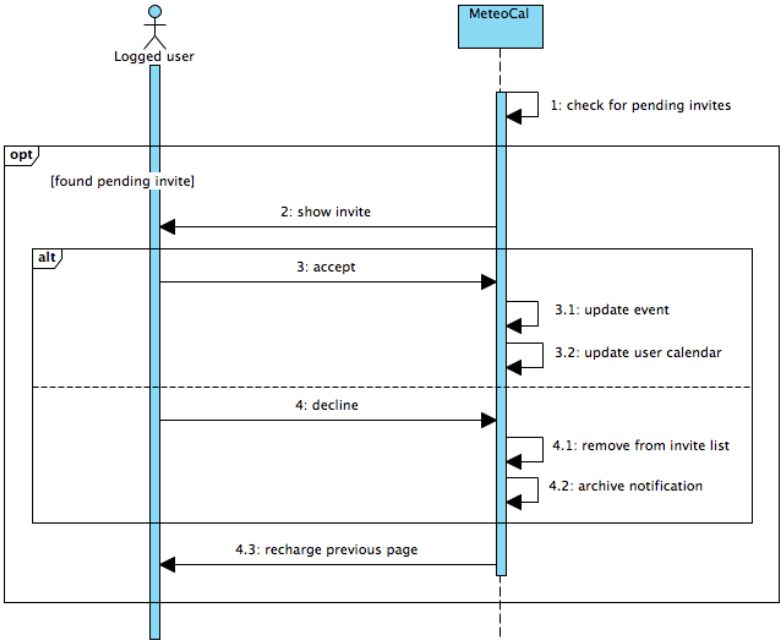


Figure 18: USC009 - Decline an event invite

17.2.8 Search the calendar of another user

Search the calendar of another user	
Code	USC010
Description	A user searches the page of another user to see his calendar
Goal	G4: Allow users to see other users public calendar
Assumptions	1. both users are registered 2. the user who performs the search is logged in 3. the user who performs the search knows the name/username of the other
Actors	1. Logged user
Entry condition	The user is on a MeteoCal page
Exit condition	The user finds the desired calendar
Flow of events	1. The user clicks on the search bar and writes the other user name (or user name) 2. He presses enter 3. The systems shows a result page with all the people with the selected name 4. The user scrolls the bar and search the other one looking at the displayed details (name, username, mail) 5. The user finds the desired user and press the related "see calendar" button
Exceptions	No users found during the search
Special requirements	
Nonfunctional requirements	

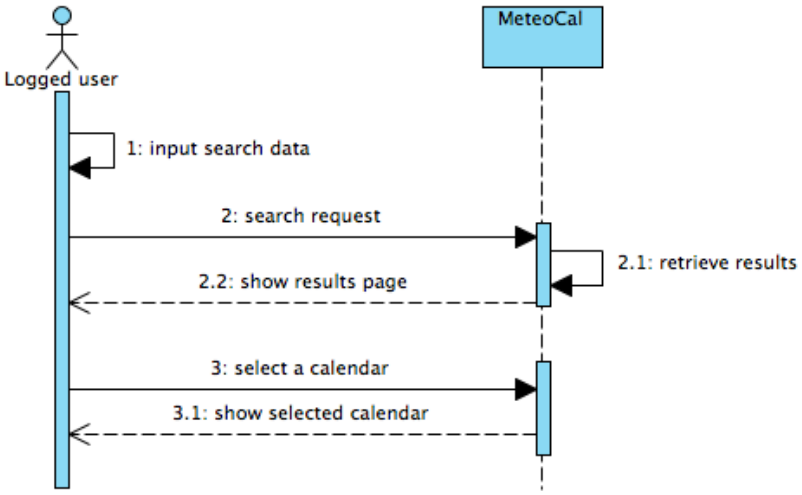


Figure 19: USC010 - Search the calendar of another user

17.2.9 Browse another user calendar

Browse another user calendar	
Code	USC011
Description	Describing how a user navigate through another user's calendar
Goal	• G4: Allow users to see other users public calendar • G5: Allow users to see other users public events details
Assumptions	1. both users are registered 2. the second user has a public calendar with at least a public event 3. the first user has performed a search
Actors	1. Logged user
Entry condition	The user has clicked on the "see calendar" button of a search result
Exit condition	The user who performs the search finds the information he needs
Flow of events	1. The user goes through the selected user calendar 2. The user accesses the event details of a public event by clicking on the event box in the calendar 3. The systems retrieves the information about the event and shows them to the user in a box on the right of the calendar
Exceptions	• The user clicks on a private event • The "searched" user has a private calendar
Special requirements	
Nonfunctional requirements	

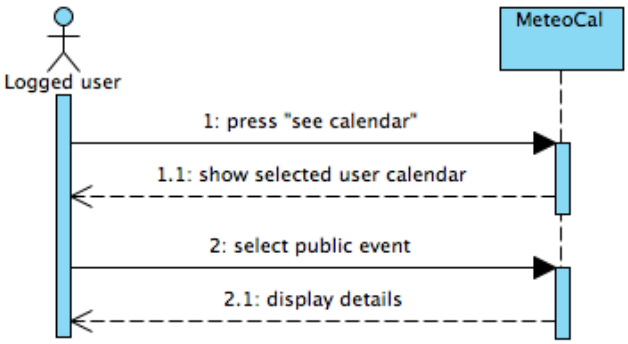


Figure 20: USC011 - Browse another user calendar

17.2.10 Receive bad weather alert

Receive bad weather alert	
Code	USC012
Description	The user receives a bad weather alert with one day of advance from the event
Goal	G6: Send a notification to all the participants one day in advance in case a bad weather
Assumptions	1. the user is registered 2. the user is going to take part in an outdoor event on the following day 3. the weather forecasts for the following day are bad 4. the user has not answered to the notification yet
Actors	1. Logged user
Entry condition	The user is on a MeteoCal page and he receives a notification
Exit condition	The notification disappears
Flow of events	1. The user logins into MeteoCal 2. The system checks the user's outdoor events: if an event of the following day has bad weather he sends a notification to the user 3. The user receives the notification (which is shown "over" the MeteoCal page he's currently on) 4. The user presses the "got it" button 5. The notification disappears
Exceptions	• The user exits MeteoCal before after receiving the notification but before clicking on the "got it " button • The user exits MeteoCal before he receives the notification
Special requirements	
Nonfunctional requirements	

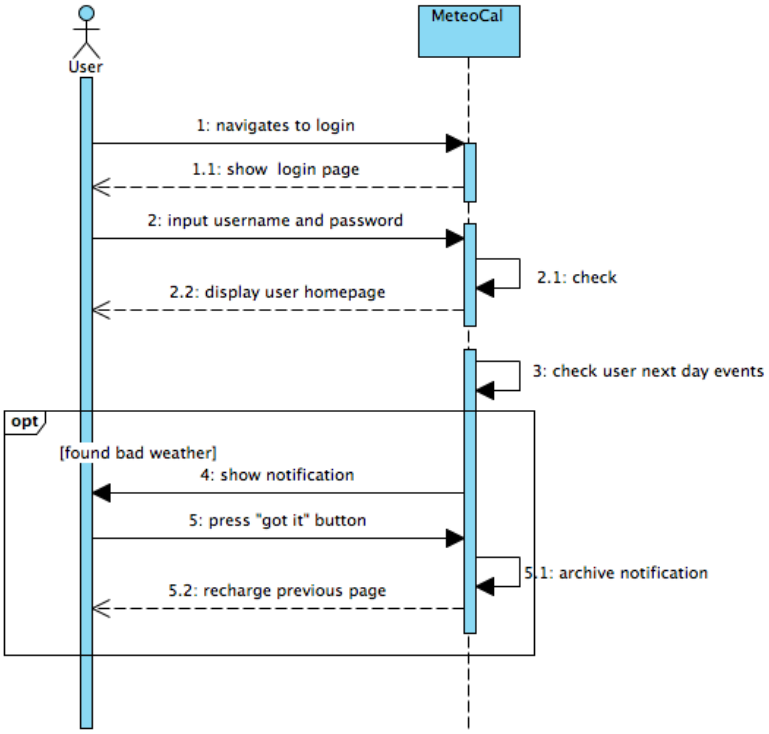


Figure 21: USC012 - Receive bad weather alert

17.2.11 Propose a sunny day

Propose a sunny day	
Code	USC013
Description	If three days before the event its weather forecasts are bad, the system warns the creator and suggests him to change the date to the closest sunny day. The user can move the event or leave it on the scheduled day
Goal	G7: Propose an alternative schedule to the event creator three day in advance in case of bad weather
Assumptions	1. the user is registered 2. the user created an event 3. there are bad weather forecasts for the event 4. the event will take place after three days 5. the user accesses MeteoCal for the first time in that day
Actors	1. registered user
Entry condition	The user accesses MeteoCal
Exit condition	The systems has updated the event information or archived the notification
Flow of events	1. The user accesses MeteoCall 2. The systems checks the forecasts for the user outdoor events that will take place after three days 3. If it finds one event with bad weather forecasts it looks for the closest sunny day 4. Once found the sunny day the system sends the notification 5. The user, who's still on a MeteoCal page, receives the notification (shown over his current page) 6. The user selects whether to postpone the event or ignore the notification ("move" button, "ignore" button) 7. If necessary the systems updates the event details and generates the notifications for the participants

Exceptions	• The user exits MeteoCal after receiving the notification but before clicking on the “got it“ button. • The user exits MeteoCal before he receives the notification. • The system is not able to find any close sunny day. • The closest sunny event is full of other events.
Special requirements	
Nonfunctional requirements	

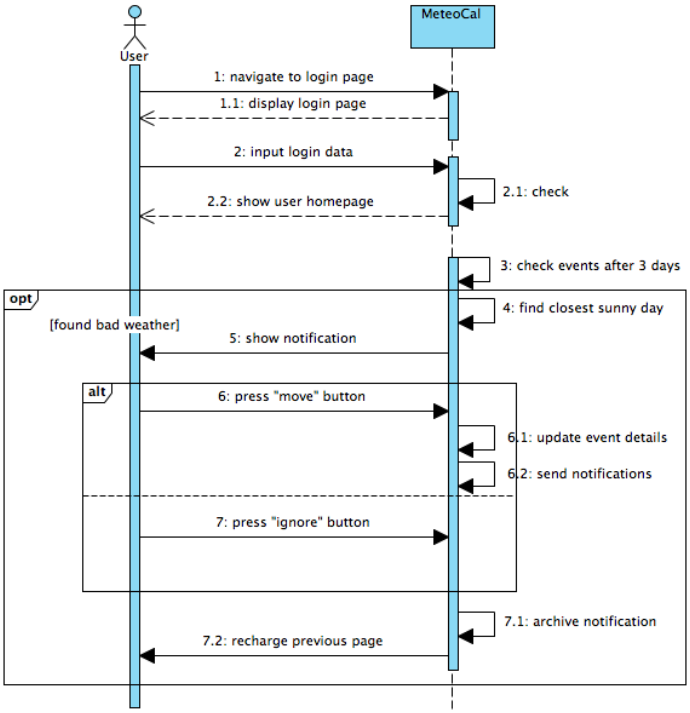


Figure 22: USC013 - Propose a sunny day

17.2.12 Receive date changed notification

Receive date changed notification	
Code	USC014
Description	When the creator of an event changes its scheduled date due to bad weather all the participants are notified
Goal	G9: Notify all the event's participant in case the creator changes some details
Assumptions	1. the user is registered 2. he accepted the invite to an event 3. the creator changed the date of the event for bad weather conditions 4. the user accesses MeteoCal for the first time after the creator moved the event
Actors	1. registered user
Entry condition	The user receives a notification
Exit condition	The systems archives the notification
Flow of events	1. The user logins and enters in MeteoCal for the first time after the event creator moved its date 2. The system checks if there are "date-changed" notification pending for the user 3. The systems shows the notification 4. The user clicks on the "got it" button 5. The system deletes the notification
Exceptions	• The user exits MeteoCal before after receiving the notification but before clicking on the "got it" button • The user exits MeteoCal before he receives the notification
Special requirements	
Nonfunctional requirements	

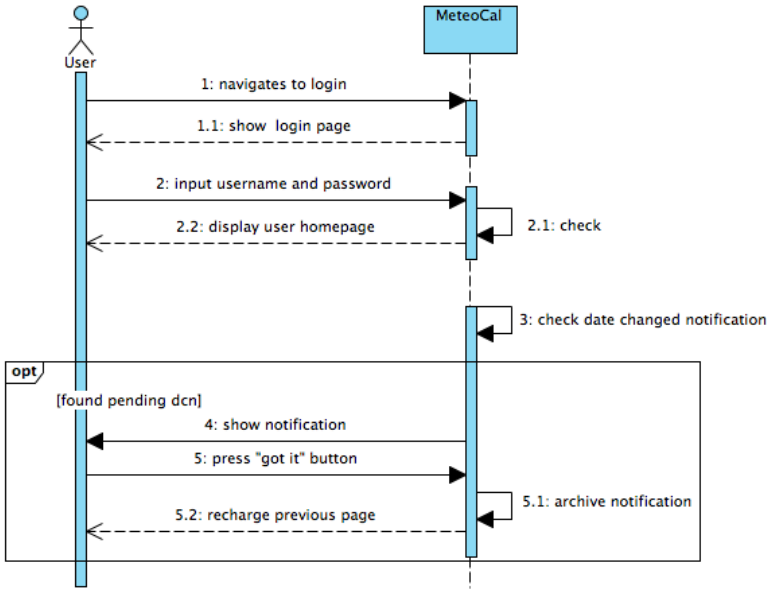


Figure 23: USC014 - Receive date changed notification

17.2.13 Change data

Change data	
Code	USC015
Description	A user accesses his settings page and change his password and his email address
Goal	G10: Allow users to modify their data
Assumptions	1. the user is logged
Actors	1. Logged user
Entry condition	The user decides to change his personal details
Exit condition	The system has successfully updated the user information
Flow of events	1. The user press “settings button” 2. The system shows him the personal information page 3. The user types in the relative box the new mail 4. The user types in the relative box the old password and the new one 5. The user presses the “save” button 6. The system checks the old password 7. The system saves the updated information 8. The system shows to personal information page with the new details
Exceptions	• The user does not type a valid email address. • The user uses the wrong boxes. • The user writes the old password. • The user exits before saving
Special requirements	
Nonfunctional requirements	

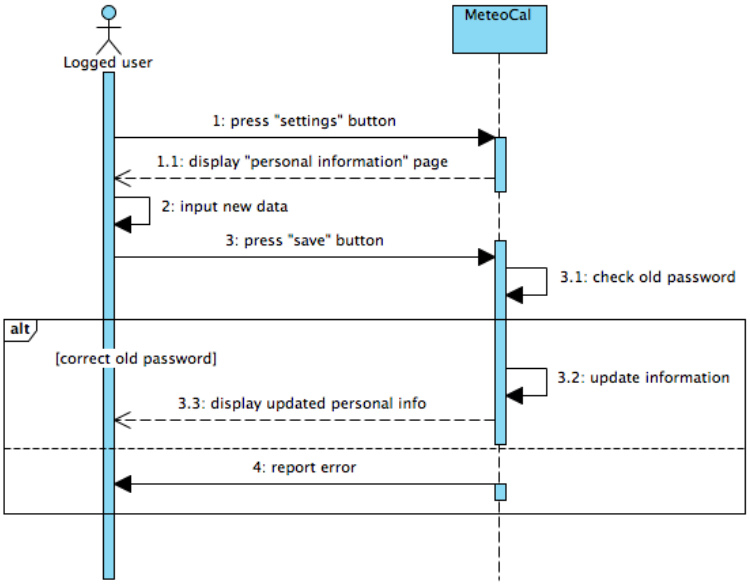


Figure 24: USC015 - Change data

17.2.14 Update weather forecasts

Update weather forecasts	
Code	USC016
Description	MeteoCal asks an external meteo service for updated weather forecasts for all the scheduled events
Goal	• G7: Send a notification to all the participants one day in advance in case of bad weather • G8: Propose an alternative schedule to the event creator three day in advance in case of bad weather
Assumptions	1. the systems has some user events in his database 2. the system has an interface to "speak" with an external weather forecasts service
Actors	1. external meteo service
Entry condition	The system has to update some forecasts
Exit condition	The system saved the updated forecasts that are now available to users
Flow of events	1. The MeteoCal system generates a forecast request for the event that has to be updated 2. The forecast service answers providing new forecasts 3. The system update its record with the new data 4. The procedure is repeated for all the event that has to be updated
Exceptions	• The external meteo service is not available • The external meteo service can't provide new forecasts • The external meteo service can't provide forecasts for the relevant place
Special requirements	
Nonfunctional requirements	

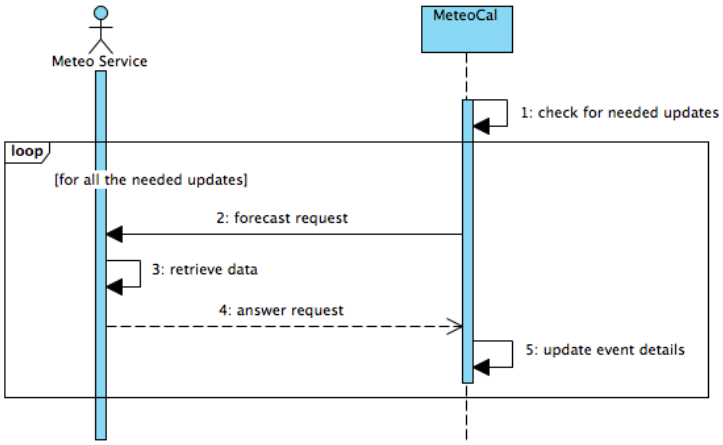


Figure 25: USC016 - Update weather forecast

18 Performance requirements

The software product requires that every web page shall download in 15 seconds or less.

19 Design constraints

The software product must be designed and implemented with JEE technologies, in particular EJBs for the business logic.

20 Software system attributes

20.1 Reliability

The software product does not have any reliability related issue because malfunctioning would cause minor inconveniences.

20.2 Availability

The system shall be available 24 hours per day, 365 days per year.

20.3 Security

The software product must provide secure storage of the passwords of its users. This can be achieved by using any cryptographic techniques.

20.4 Maintainability

The database must be backed up periodically, so that new events and users data are not lost in case of malfunction.

20.5 Portability

The software product can be installed in any operating system that supports the JVM and its dependent components.

21 Other requirements

The software product must provide understandable messages in text form in the event of errors, and instruct the user on what to do.

Part IV
Appendixes

22 Alloy

In this section we try to understand if our Class Diagram can be consistent using Alloy Analyzer.

22.1 Alloy code

The alloy code can be seen in the attached file `./MeteoCal.als`

22.2 Alloy output

Figure 26 is the output of Alloy Analyzer showing that our model is consistent.

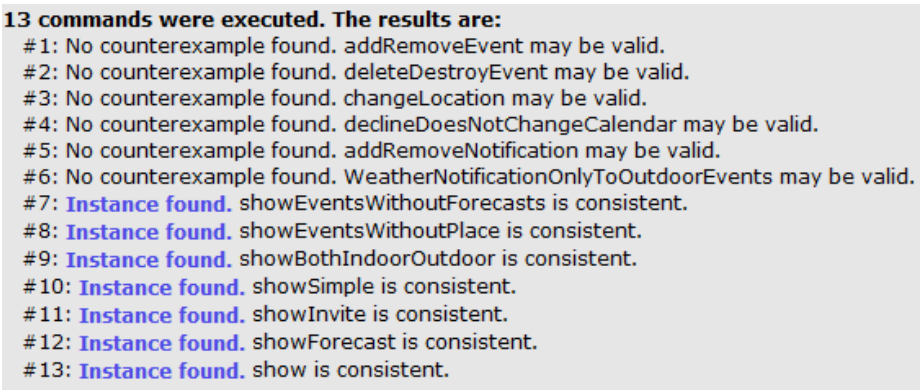


Figure 26: Alloy Analyzer output

22.3 Worlds generated

Below we attach some worlds generated with our model in order to make understand that our model correctly describes the problem.

Figure 27 describes a really simple world.

Figure 28 describes a simple world with an invite notification.

Figure 29 describes a simple world with a weather forecast.

Figure 30 describes a rather complex world

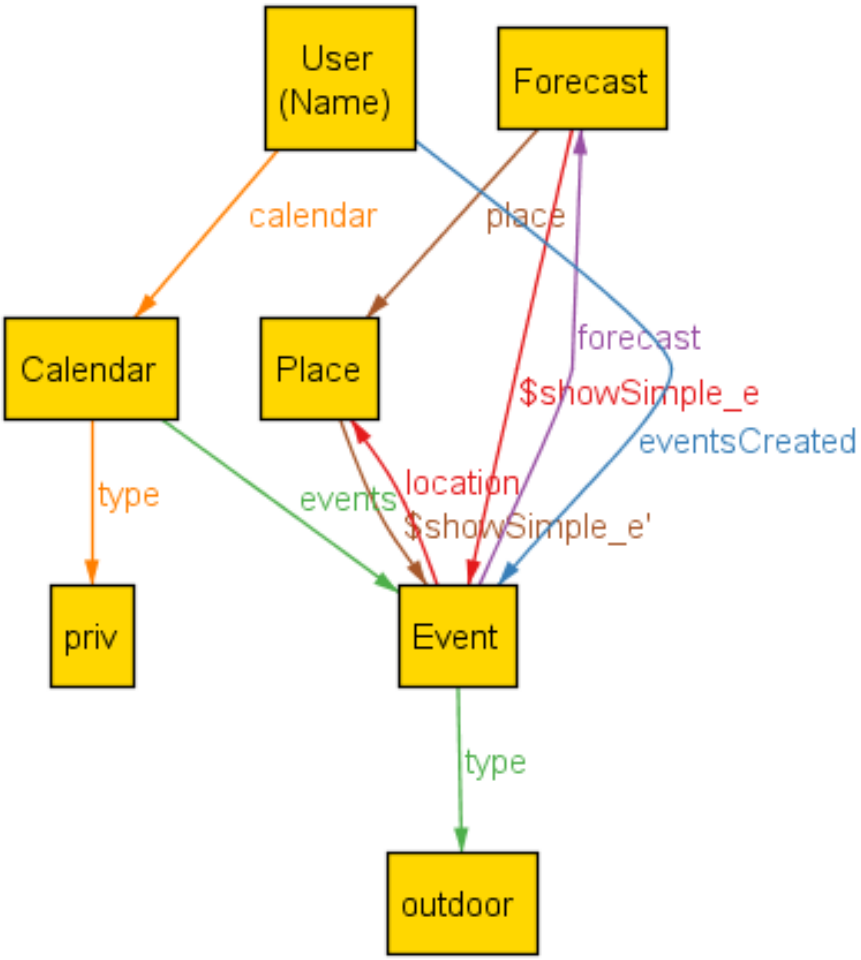


Figure 27: Alloy simple world

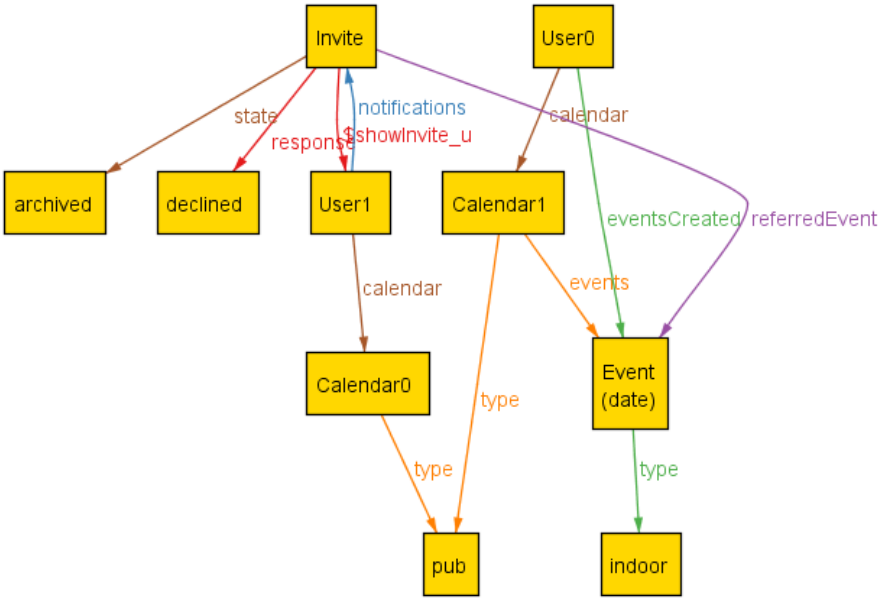


Figure 28: Alloy simple world with an invite

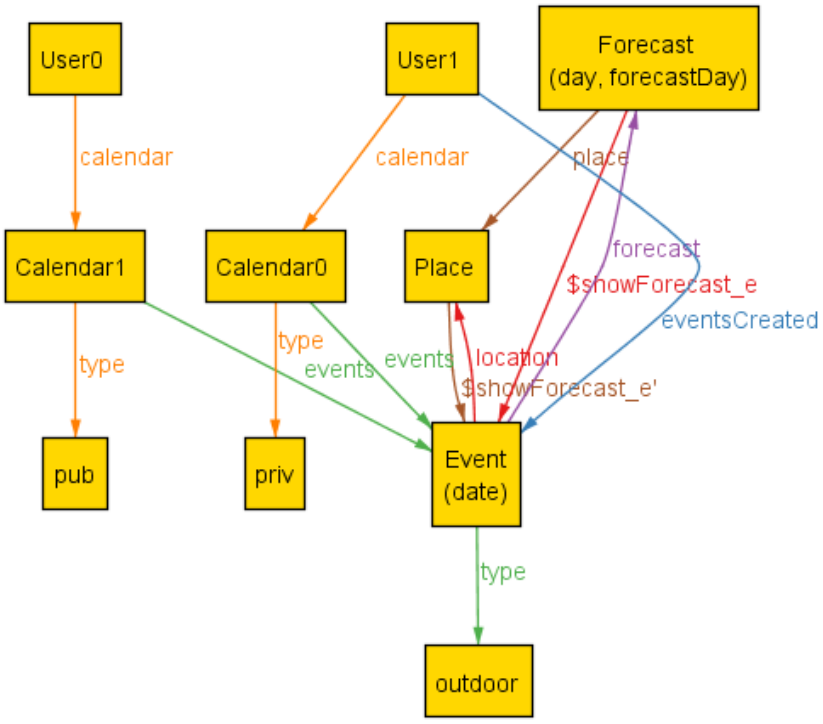


Figure 29: Alloy simple world with a weather forecast

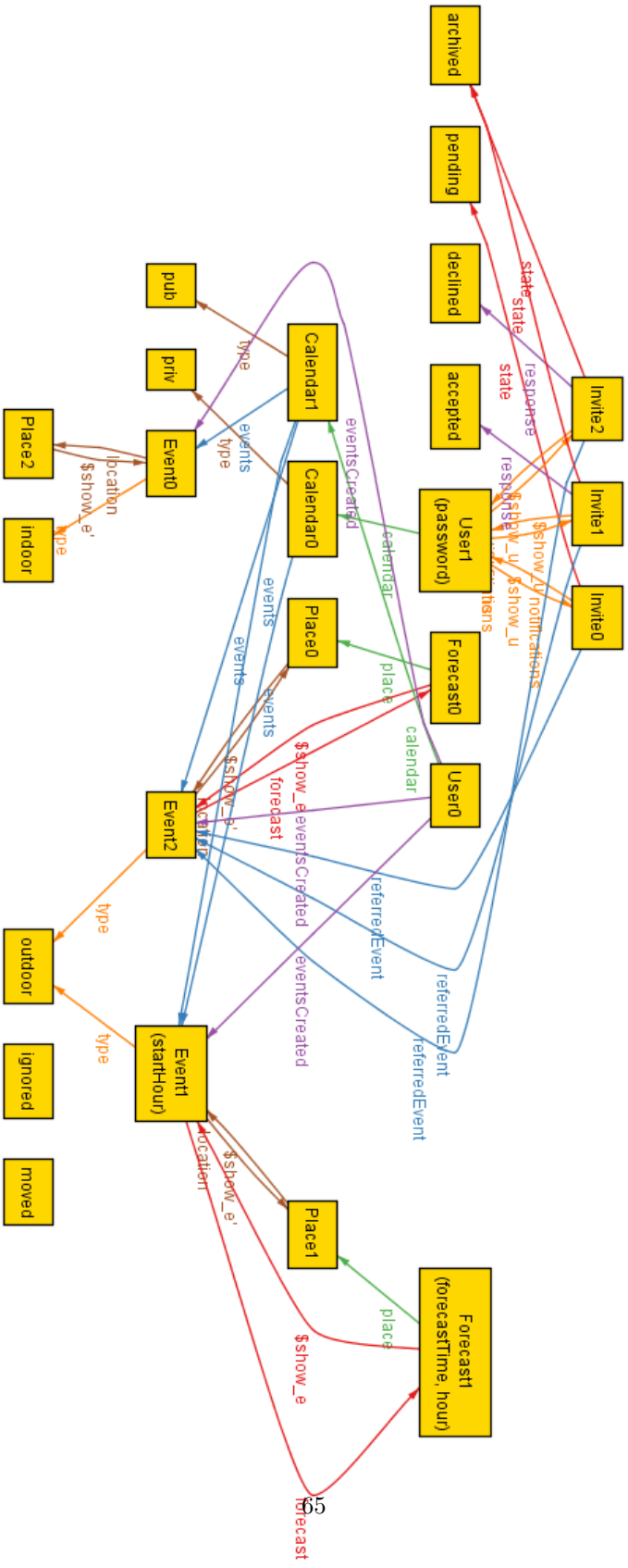


Figure 30: Alloy complex world

23 Worked hours

To produce this document we have worked 60 hours, so distributed:

- Andrea Celli: 30 hours
- Stefano Cereda: 30 hours

24 Used tools

To create this document we have used to following tools:

- Office 365: to collaboratively redact the document and to keep track of the worked hours
- LaTeX: to format the document
- Cacoo: to create the UI sketches
- StarUML 2.0.0: to create the use case diagram
- Visual Paradigm 11.2: to create the class diagram, the sequence diagrams and the state charts
- Alloy Analyzer 4.2: to prove the model consistency

Contents