MeteoCal

[Requirements Analysis and Specification Document]

Mite Ristovski
Dushica Stojkoska
Milica Shulevska

Software Engineering 2 project

Politecnico di Milano
5th School of Engineering

Milan, November 16, 2014

CONTENTS

1 Introduction 4

1.1 Purpose 4

1.2 Scope 4

1.3 Definitions, Acronyms and Abbreviations 6

1.3.1 Definitions 6

1.3.2 Acronyms and abbreviations 6

1.4 References 6

1.5 Overview 6

2 Overall Description 8

2.1 Product Perspective 8

2.1.1 System Interfaces 8

2.1.2 User Interfaces 8

2.1.3 Hardware Interfaces 11

2.1.4 Software Interfaces 12

2.1.5 Communication Interfaces 12

2.1.6 Memory 12

2.1.7 Operations 13

2.1.8 Site Adaptation Requirements 13

2.2 Product Functions 13

2.2.1 Functional Requirements 13

2.3 User Characteristics 14

2.4 Constraints 15

2.4.1 Regulatory Policies 15

2.4.2 Hardware Limitations 15

2.4.3 Interfaces to other applications 15

2.4.4 Parallel operation 15

2.4.5 Audit Function 15
2.4.6 Control Functions 15
2.4.7 Higher-order language requirements 15
2.4.8 Signal handshake protocol 15
2.4.9 Reliability requirements 15
2.4.10 Criticality of the application 15
2.4.11 Safety and security considerations 16
2.5 Assumptions and Dependencies 16
2.6 Apportioning of requirements 16
3 Specific requirements 17
3.1 External Interfaces 17
3.1.1 User Interfaces 17
3.1.2 Hardware Interfaces 18
3.1.3 Software Interfaces 18
3.1.4 Communication Interfaces 18
3.2 Functional Requirements 18
3.2.1 Scenarios 18
3.2.2 Analysis model 24
3.2.3 State chart model 25
3.2.4 Activity Model 27
3.2.5 Use cases 28
3.3 Performance Requirements 40
3.4 Logical Database Requirements 40
3.5 Design Constraints 40
3.6 Standard Compliance 40
3.7 Software System Attributes 40
3.7.1 Reliability 40

3.7.2 Availability 40
3.7.3 Security 41
3.7.4 Maintainability 41
3.7.5 Portability 41
3.8 Other Requirements 41
4 Appendices 42
4.1 Alloy Model 42

1 INTRODUCTION

1.1 PURPOSE

The purpose of this Requirements Analysis and Specification Document is to describe the general functionalities of the MeteoCal application assigned as a project in the Software Engineering 2 course. Moreover, it contains all the critical aspects of the application and establishes an agreement between a client and a developer.

The intended audience of this document is all the people actively participating in the software engineering course, including professors and tutors. Not only shall the document serve as a reference for the developers to follow the requirements, but it will also serve to the testers to check whether the stated requirements and goals are met or not.

1.2 SCOPE

The software product that is going to be developed is the MeteoCal application. The application will be a calendar based service for organizing and participating in events. Furthermore, the service will include weather forecast for the organized events. Also, a notification service will be introduced for informing the participants in the event in case of weather change.

The product has the following general objectives:

- Let user register on the system.
- Let user has his/her own calendar.
- Let the user create/update/delete an event. (If a user creates an event it makes him/her an organizer of that event)
- Let the organizer invite people to his/her event and the invited users can either accept or decline an invitation. By accepting the invitation, they are considered as participants on the event.
- Notification service informing the participants of the weather change.
- User can import other calendar or export his/her own calendar.

The product will have the following general functionalities:

- **Manage users**
- **Manage organizers**
- **Manage events**

The product will have the following limitations:

- **Password change**

The user will not be able to change or reset his/her password. The system will not have this option.

- **One event at a time**

The users will not be able to organize multiple events at the same time. The system will only offer the option to create only one event at a given time by one user.

The software will have the following goals:

[G1] Users can register on the system.

[G2] Users can create an event.

[G3] Organizers can update/delete an event.

[G4] Organizers can invite users on the event.

[G5] Invited users can accept or decline an invitation.

[G6] Allow user to make his/her calendar public.

[G7] Allow events to be made public or private.

[G8] Allow event participants to receive a notification about weather forecast.

[G9] The system should propose a closest sunny day to the organizer three days before an event which is outdoors and the weather is bad.

[G10] Allow users to import and export their calendar.

[G11] The weather information is updated periodically.

[G12] Avoid conflicts with existing events (manage time consistency).

[G13] Non-registered users can view public events.

1.3 Definitions, Acronyms and Abbreviations

1.3.1 Definitions

Keyword	Definition
User	All the registered users in the system.
Organizer	A user that has created an event.
Event	A planned public or social occasion, created by a user.
Invited users	Users that have been invited to an event by the organizer
Participants	Invited users that have accepted the invitation.

Table 1: Definitions

1.3.2 Acronyms and abbreviations

Acronym/Abbreviation	Definitions
XML	Extensible Markup Language
RASD	Requirements Analysis and Specification Document
NFR	Non-Functional Requirements
FR	Functional Requirements
QA	Quality Attributes
G	Goal
DBMS	Data Base Management System
AS	Application Server
JEE	Java Enterprise Edition

Table 2: Acronyms and abbreviations

1.4 References

1. IEEE Recommended Practice for Software Engineering Requirements Specification (http://ieeexplore.ieee.org/xpl/mostRecentIssue.jsp?reload=true&punumber=5841)
2. Alloy model file: meteocal.als

1.5 Overview

This document is organized as follows:

1. **Introduction**

   This section provides a synopsis of the software product to be developed.

2. **Overall Description**

   This section describes the general factors that affect the software product and its requirements. For instance, the different interfaces, operations, functions, user characteristics, constraints, assumptions and dependencies and future requirements.

3. **Specific Requirements**

   This section contains all the analysis done to the project requirements. It describes all of the software requirements to a level of detail sufficient to be externally perceivable. It contains the different use cases, interfaces, scenarios and sequence diagrams, among others.

4. **Appendices**

   This section provides supporting information showing how the alloy model contributed to the analysis model and requirement analysis.

2 OVERALL DESCRIPTION

This section describes the general factors that affect the software product and its requirements, and it provides a background for specifying concrete requirements in the next section of this document.

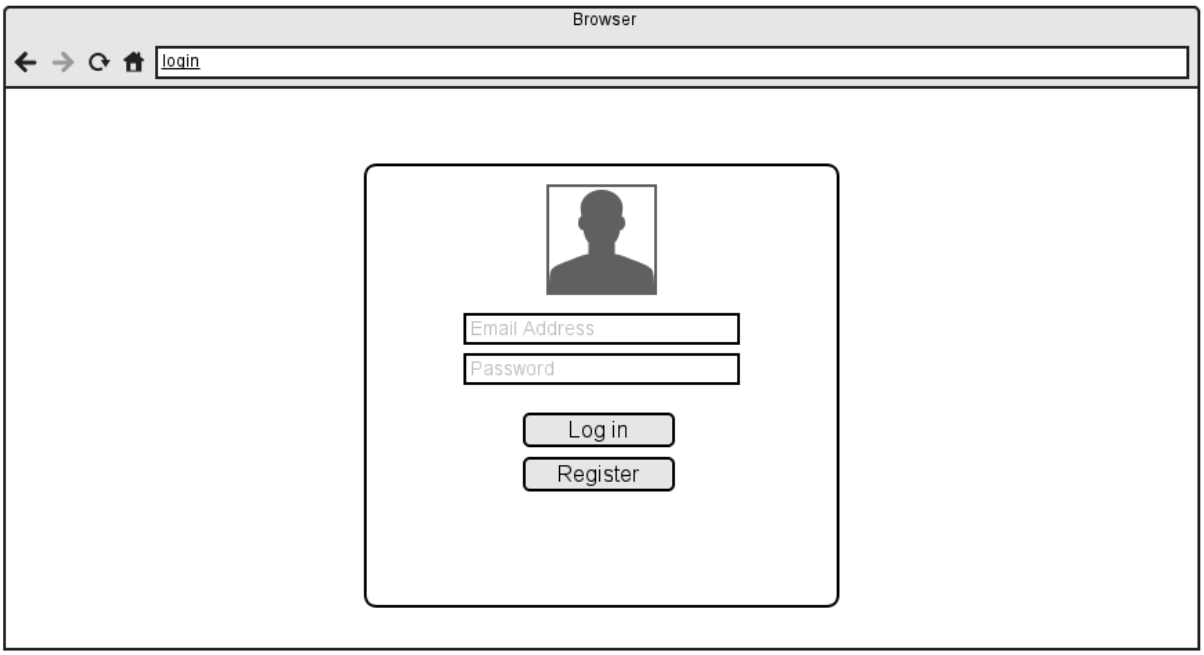
2.1 PRODUCT PERSPECTIVE

The software product is a completely self-contained system, independent from other systems.

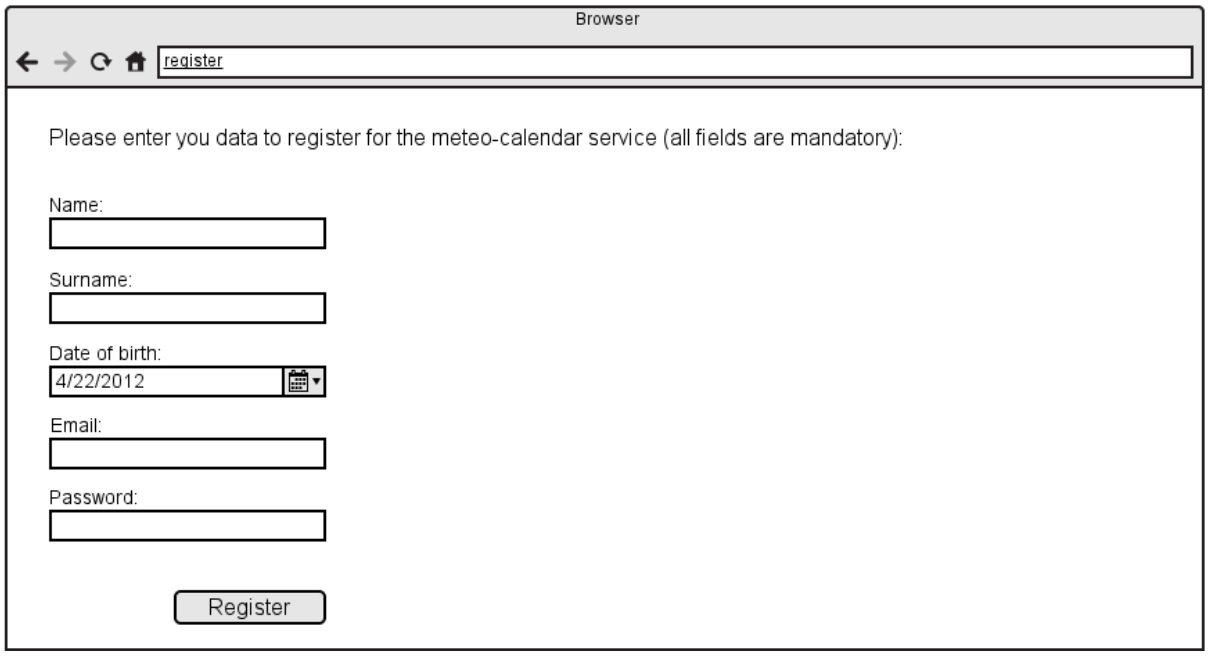
2.1.1 System Interfaces

The software product does not provide any external interfaces.

2.1.2 User Interfaces

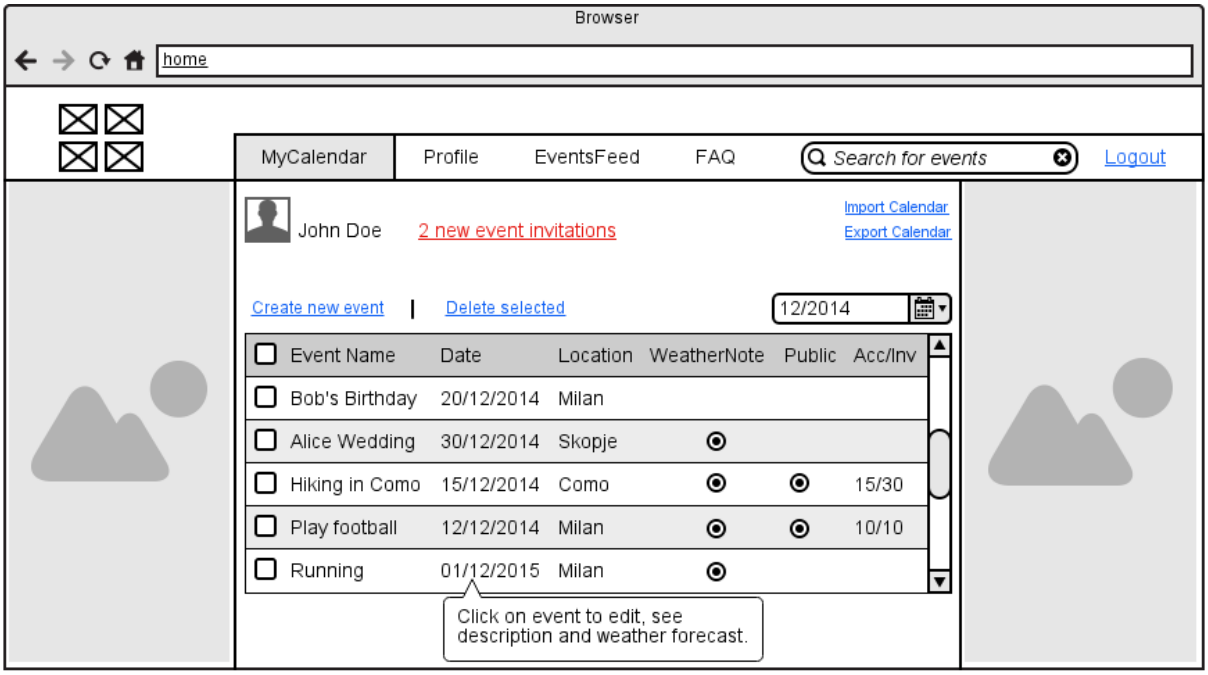


User Interface 1: Login

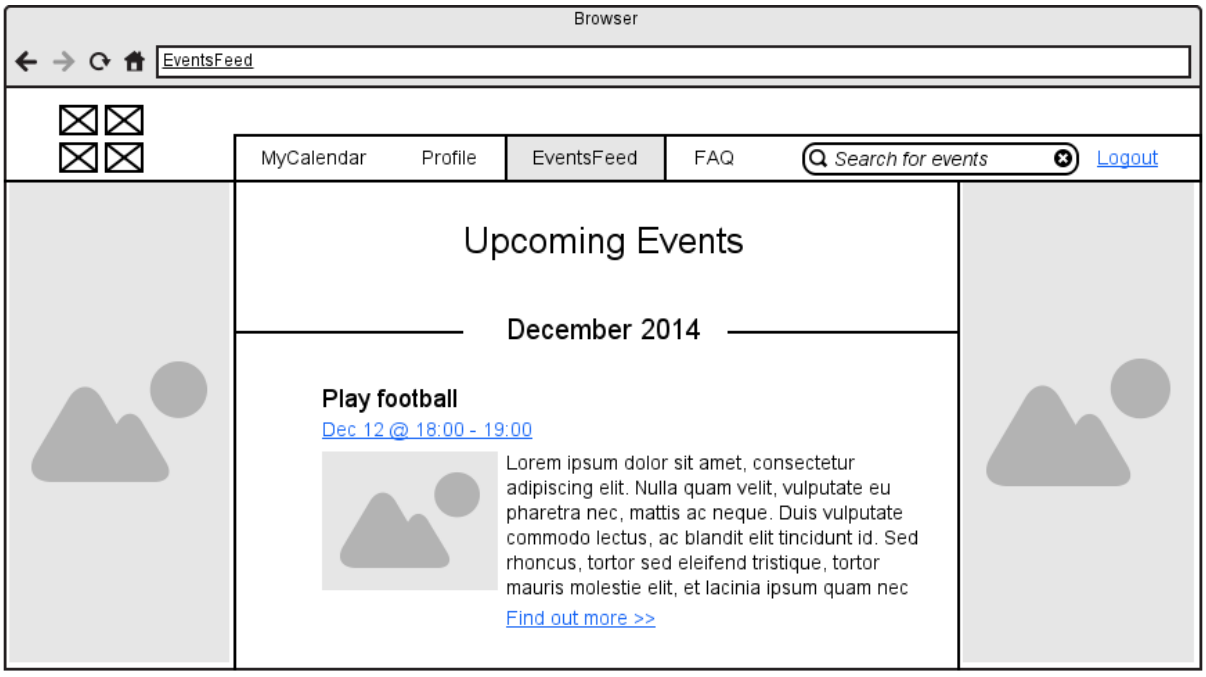


User Interface 2: Register form

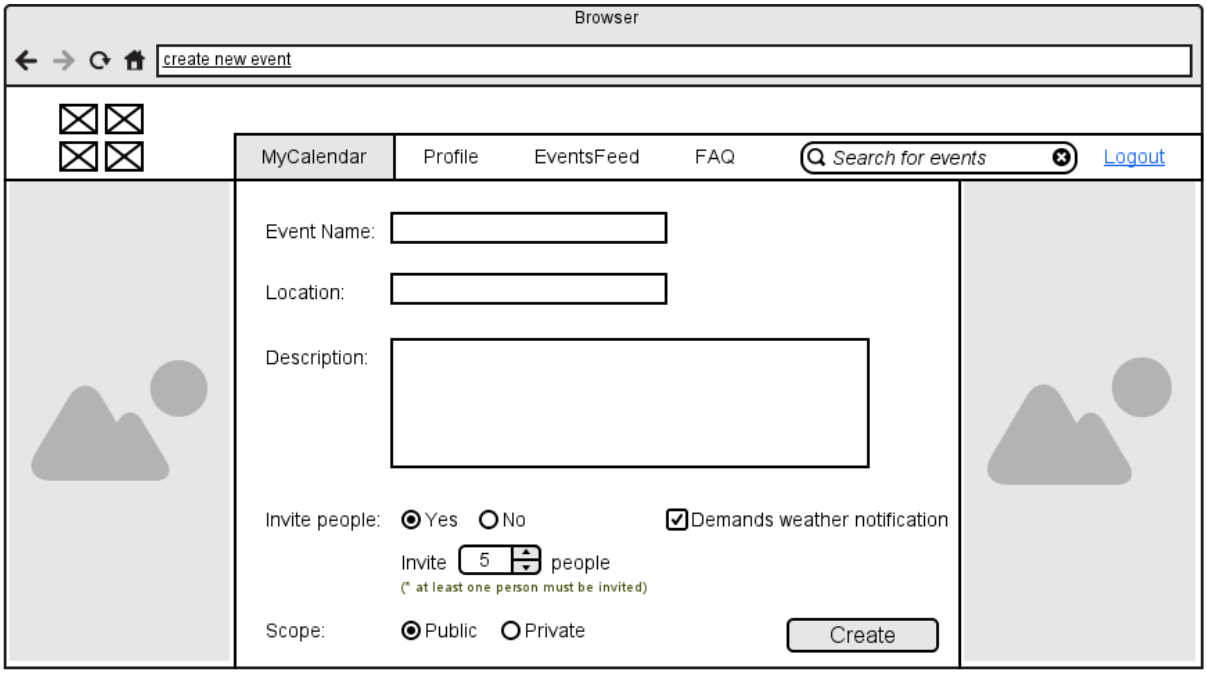
User Interface 3: User profile



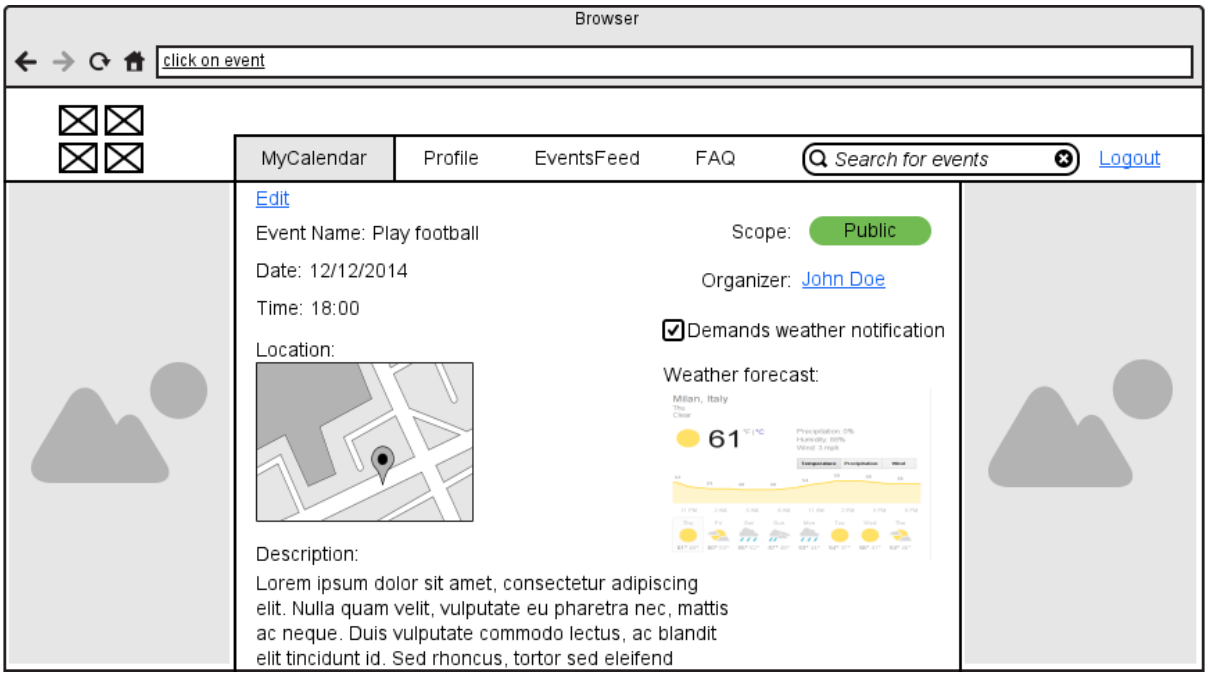
User Interface 4: Homepage of MeteoCal for a registered user.



User Interface 5: Event feed of a registered user.



User Interface 6: Creation of a new event.



User Interface 7: Event page.

2.1.3 Hardware Interfaces

The software product does not provide any hardware interfaces.

2.1.4 Software Interfaces

- *Database Management System*

  Name: MySQL

  Mnemonic: MySQL

  Specification number: Community Server

  Version number: 5.6.21

  Source: http://dev.mysql.com/downloads/mysql/

- *Application Server*

  Name: GlassFish

  Mnemonic: GlassFish

  Version number: 4.1

  Source: https://glassfish.java.net/download.html

- *Operating System*

  The software product will run on any operating system which supports Java Virtual Machine, Database Management System and Application Server described above.

2.1.5 Communication Interfaces

Protocol	**Port**	**Service**
TCP	80	TCP
TCP	3306	MySQL (only if it is in a different physical server)

Table 3: Communication Interfaces

It is important to note that for the development of the first (current) version of the software we will assume that the Database Management System and Application Server reside on the same physical server.

2.1.6 Memory

The minimum memory requirements are:

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

Note that the secondary memory is recommended to be physically on a different server from where the software product is installed, as it can exponentially grow without affecting system's performance. For the current production we assume that they are at least installed on the same server.

2.1.7 Operations

A user can interact with the system as a functional user (registered user – guest or organizer). The functional operations for all the users are described in the product functions section.

2.1.8 Site Adaptation Requirements

The product software requires the following conditions to be satisfied in order to run successfully:

- Java Virtual Machine
- Application Server
- Database Management System
- Primary memory required space
- Secondary memory required space

Furthermore, users are required to have installed any of the following web browsers: IE 7.0+, FF 10+, Chrome 20+ or Safari 5+.

2.2 PRODUCT FUNCTIONS

This subsection describes a summary of major functions, functional and non-functional requirements of the software product.

2.2.1 Functional Requirements

The functional requirements are divided in three principal topics:

- *Manage users*
- *Manage organizers*
- *Manage events*

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

2.2.1.1 Manage users

- Functional Requirements

  [FR1] Register to the system.

  [FR2] Login/Logout.

  [FR3] Consult user profiles.

  [FR41] Consult event information.

  [FR5] Accept or decline an event information.

  [FR6] Receive notification about the weather.

- Non-functional Requirements

  [NFR1] The user password must be stored securely.

  [NFR2] The system must manage high number of users.

2.2.1.2 Manage organizers

- Functional Requirements

  [FR7] The organizer can create an event.

  [FR8] The organizer can update an event.

  [FR9] The organizer can delete an event.

  [FR10] The organizer can invite people to an event.

- Non-functional Requirements

  [NFR3] An organizer can manage only one event at a time.

2.2.1.3 Manage events

- Functional Requirements

  [FR11] Consult events.

  [FR12] Events have only one organizer.

- Non-functional Requirements

  [NFR4] Only invited users can attend an event.

2.3 USER CHARACTERISTICS

The user characteristics are:

- Knowledge in using a browser

2.4 CONSTRAINTS

The following constraints apply to the software product:

2.4.1 Regulatory Policies

The project does not have any regulatory policies.

2.4.2 Hardware Limitations

The project does not have any hardware limitations.

2.4.3 Interfaces to other applications

The project does not have any interface with other applications.

2.4.4 Parallel operation

The software product must control different levels of concurrency, for instance concurrent users; creation and management of the events etc. It is necessary the management of parallel operations at the application level, and database level. These operations are important for the general functionality of the product and they will be resolved in the design planning phase of the project.

2.4.5 Audit Function

The software does not perform any audit.

2.4.6 Control Functions

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

2.4.7 Higher-order language requirements

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

2.4.8 Signal handshake protocol

The software product does not manage any handshake protocol.

2.4.9 Reliability requirements

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

2.4.10 Criticality of the application

The software product requires proper support for concurrent users.

2.4.11 Safety and security considerations

The software product does not require any safety and security considerations.

2.5 Assumptions and Dependencies

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

- The Java virtual machine is already installed on the operating system.
- Users have access to a decent Internet connection.
- The software product supports any number of users.

2.6 Apportioning of requirements

Future releases of the software product may provide support for:

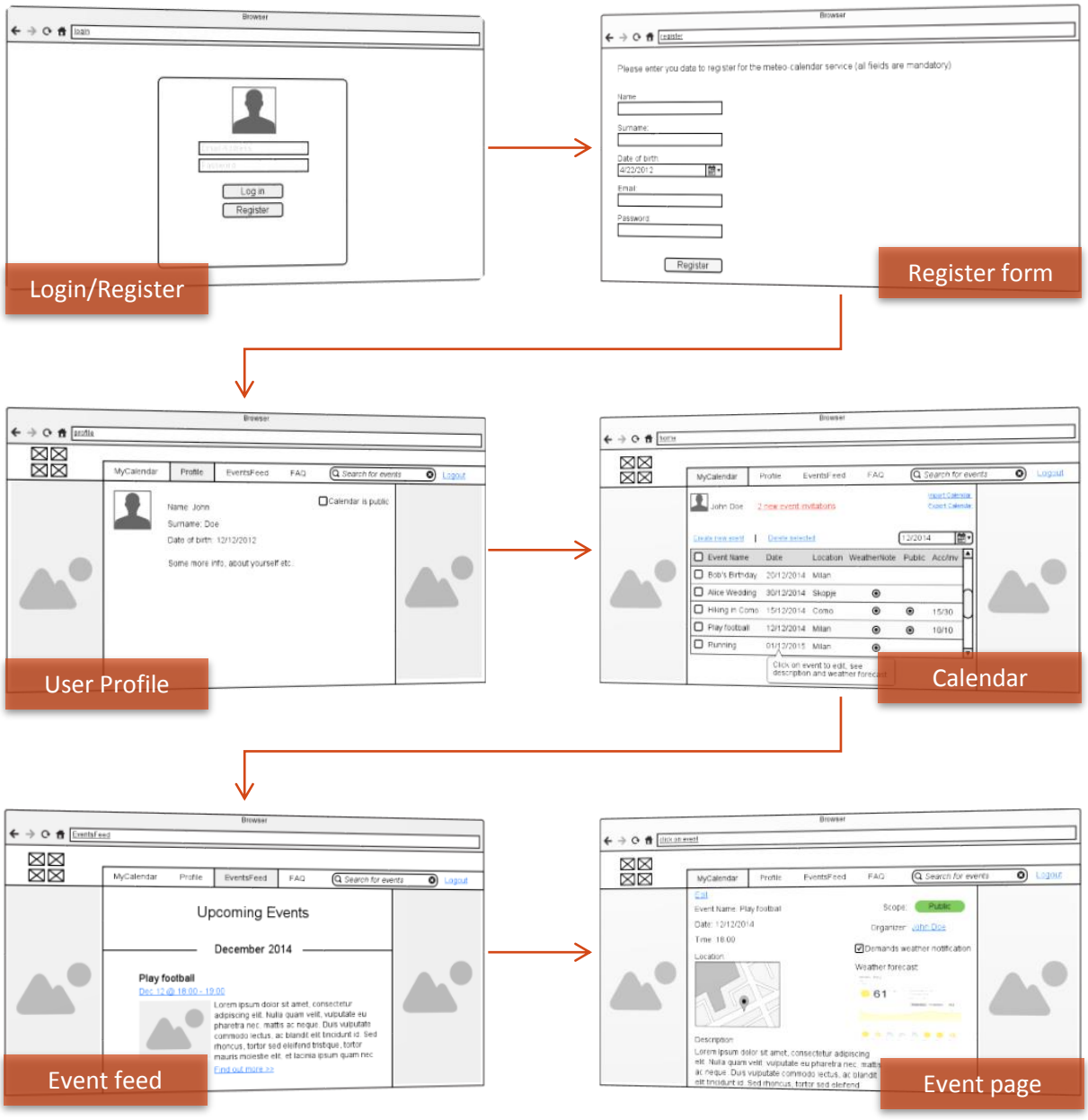
1. Password modifying and resetting.
2. One user can create more events at the same time.
3. Making guest lists in advance for faster invitation to the events.

3 SPECIFIC REQUIREMENTS

3.1 EXTERNAL INTERFACES

3.1.1 User Interfaces

This story board explains how previously mentioned user interfaces are connected with each other.



Storyboard: User navigation through MeteoCal.

3.1.2 Hardware Interfaces

The system does not have hardware interfaces.

3.1.3 Software Interfaces

The system does not have software interfaces.

3.1.4 Communication Interfaces

The system does not have communication interfaces.

3.2 Functional Requirements

3.2.1 Scenarios

User registers to the system	
Code	SC-001
Description	Describe how a user registers to the system
Goal	**[G1]** Users can register on the system.
Assumptions	The user is not yet registered on MeteoCal.
Scenario	
Lara is a student who would like to use a system for managing events and weather forecast. One of her friends told her about MeteoCal, so she decides to use it. Lara connects to the internet and opens up a browser installed on her computer. She goes to the internet address of the system and when the first page loads, Lara notices that there is a register option. After clicking the "Register" button, the register form opens up where she writes her name, surname, email, password and other information. When she finishes filling out the form, she clicks on the button "Confirm", after which she receives a message stating that the registration has been successful. Lara is now registered on the system.	

Scenario 1: User registers to the system

User logs in from the system	
Code	SC-002
Description	Describes how a user logs in the system
Goal	**[G1]** Users can register on the system..
Assumptions	The user is registered but not logged in MeteoCal.
Scenario	
Victor is Lara's friend who told her about the MeteoCal system. He has been using the system for a while now and he is already registered on it. Today Luca told him that there is some kind of a party organized in the Leonardo campus, so Victor decides to check on MeteoCal for the event.	

He connects to the internet and go to the internet address of the system using a browser installed on his computer. When the system loads the first page, Victor notices the form for Login, he writes his username and password and clicks on the "Login" button. He is redirected to his profile on the system where he can see all of the upcoming events.

Scenario 2: User logs in the system.

User logs out from the system	
Code	SC-003
Description	Describes how a user logs out the system
Goal	**[G1]** Users can register on the system.
Assumptions	The user is registered and logged in MeteoCal.
Scenario	
Victor has been checking some events on the MeteoCal system. He is already logged in on the system. Lara comes along and asks him if she can use his computer in order to log in on MeteoCal. To do that, Victor has to log out. Victor navigates to the "Logout" button in the upper right corner of the page and he clicks on it. The system loads the first page again with the Login form, so Lara can login with her username.	

Scenario 3: User logs out from the system.

User creates an event	
Code	SC-004
Description	Describes how a user creates an event
Goal	**[G2, G4, G7]** Users can create an event. Organizers invite users on the event. Allow events to be made public or private.
Assumptions	The user is registered and logged in MeteoCal. The user has not created any other event at this time.
Scenario	
Lara has her birthday next week, so she decides to celebrate it with her friends. She is registered and logged in the MeteoCal system. She can notice a button for creating a new event on the homepage of MeteoCal. After clicking it, a form appears where she needs to enter all the information about the event: date, time, and send the invitations to the friends that are registered. After finishing filling out the form, she clicks on the "Confirm" button bellow the form. By creating this, Lara becomes the organizer of the event and only she can modify or delete this event. She can choose whether she wants to make the event public or private. The page is redirected to the event view where she can see all the necessary information concerning the event.	

Scenario 4: User creates an event.

User updates an event	
Code	SC-005
Description	Describes how a user updates an event
Goal	**[G3, G9, G11]** Organizers can update/delete an event. The system should propose a closest sunny day to the organizer three days before an event which is outdoors if the weather is bad. The weather info is updated periodically.
Assumptions	The user is registered and logged in MeteoCal. The user has already created an event.
Scenario	
Lara has already created an event for her birthday. She is planning to have a small party in the beautiful garden in a friend's house. She wants to log in to check the event and see how many guests have already accepted the invitation. She is already connected to the internet, opens the page of MeteoCal, and logs in by clicking the button "Log in" and entering her username and password. The system loads a page with her profile. She notices that there is a notification and clicks on it. The notification contains information that on the day of her birthday is expecting a heavy rain. Lara decides to have the party indoors, at her house. She navigates to the "Modify" button in the upper part of the page. She can now update the location for the event. She updates it and clicks on "Save" button.	

Scenario 5: User updates an event.

User deletes an event	
Code	SC-006
Description	Describes how a user deletes an event
Goal	**[G3]** Organizers can update/delete an event.
Assumptions	The user is registered and logged in MeteoCal. The user has already created an event.
Scenario	
Luka had created an event for a little party this weekend at his house. In the meantime his boss is calling to tell him that he needs to attend a seminar which is held that same weekend, but in other town. Luka needs to delete the event he had created. Luka navigates to the "Log in" button in the upper right corner of the page of MeteoCal and he clicks on it. He is redirected to his profile on the system. He notices the button for accessing the events he had created and clicks on it. After the system loads the page for the event, he navigates to the "Modify" button in the upper part of the page. He then chooses to delete his event. A form for confirming the action appears, he clicks on "Confirm" button and the event is deleted. Luka then logs out of the system.	

Scenario 6: User deletes an event.

User consults an event	
Code	SC-007
Description	Describes how a user consults an event
Goal	**[G5, G7]** Invited users can accept or decline an invitation. Allow events to be made public or private.
Assumptions	The user is registered and logged in MeteoCal.
Scenario	
Maria wants to see the upcoming events for the weekend. She connects to the internet and opens up a browser installed on her laptop. She goes to the system page, clicks on "Log in" button in the upper right corner of the page. After entering her username and password, the system loads the page with her profile. She can now checks all events for which she received an invitation, as well as all other events that organizers had made them public.	

Scenario 7: User consults an event.

User consults other users calendar	
Code	SC-008
Description	Describes how a user consults an event
Goal	**[G6]** Allow users to make his/her calendar public.
Assumptions	The user is registered and logged in MeteoCal.
Scenario	
Maria is checking some events on MeteoCal. She is already registered and logged in the system. The page with her profile is loaded. Among many options of the system, she can choose to see calendars from other users, which made their calendars public.	

Scenario 8: Users consults other user's calendar.

User accepts an invitation.	
Code	SC-009
Description	Describes how a user accepts an invitation for an event
Goal	**[G5, G12]** Invited users can accept or decline an invitation. Avoid conflicts with existing events.
Assumptions	The user is registered and logged in MeteoCal. Invitations from other organizers are already sent to user's profile.

Scenario
Victor wants to see on which events he is invited for the week. He connects to the internet, opens up a browser installed on his laptop, and goes to the page of the MeteoCal. He clicks on “Log in” button on the right upper corner of the page. The system loads a page with his profile. He can now check the notifications to see on which events is invited to. Among the invitations, he notices that there is a science seminar on Friday. He wants to attend on the event, so he clicks on the “Accept” button for confirming that he is planning to go on that seminar. There is another invitation for a chess tournament and he wants to attend on this event too. He clicks on the “Accept” button, but a notification pops up that he already has accepted other event at that same time. He can choose on which event he prefers to go.

Scenario 9: User accepts an invitation.

User declines an invitation	
Code	SC-010
Description	Describes how a user declines an invitation.
Goal	**[G5]** Invited users can accept/decline/ an invitation.
Assumptions	The user is registered and logged in MeteoCal. Invitations from other organizers are already sent to user’s profile.
Scenario	
Victor is already logged in on his profile on MeteoCal. He is checking the invitations on which he is invited to attend. Among the invitations, he notices that there is a hiking event. He clicks on the button for accessing to the information for that event. He would like to go, but the duration of the event is in the time when he is at work. He clicks on the “Decline” button for confirming that he will not go on that event.	

Scenario 10: User declines an invitation.

User receives a notification.	
Code	SC-011
Description	Describes how a user receives a notification about the weather forecast.
Goal	**[G8, G11]** Allow event participants to receive a notification about weather forecast. The weather info is updated periodically.
Assumptions	The user is registered and logged in MeteoCal. The user has accepted invitations for one or more events.
Scenario	

Marko wants to log in on his profile on MeteoCal to see if there are some news. He connects to the internet, opens up a browser installed on his computer, opens the MeteoCal page and logs in on his profile. He notices that there is one notification. He clicks on the button for notifications and new page is loaded, which takes him to the details about that notification. He can see that the party for celebrating his friend's graduation is postponed, because it is expecting a rain that day and the party is outdoor.

Scenario 11: User receives a notification.

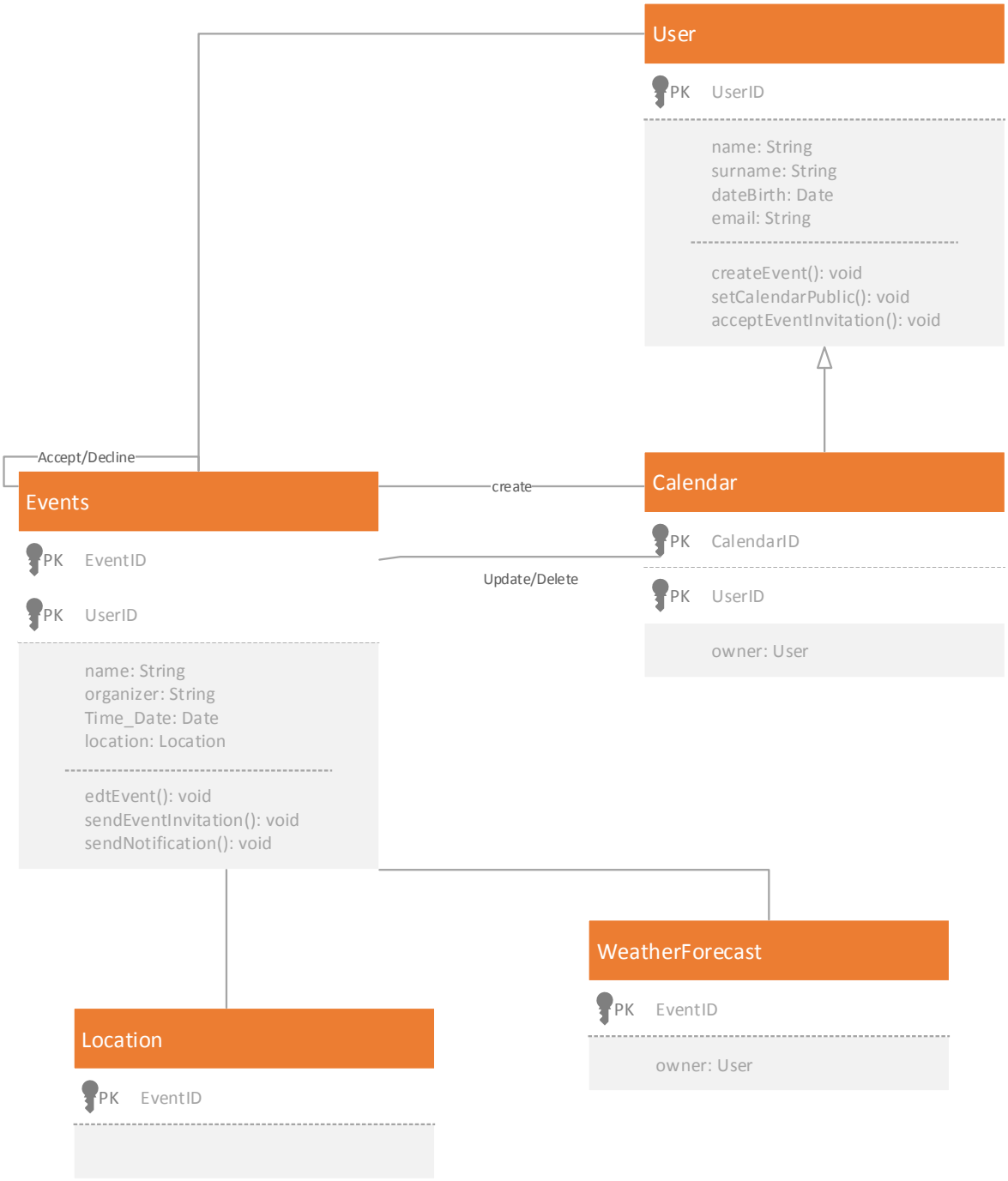
User can export calendar.	
Code	SC-012
Description	Describes how a user can export calendars.
Goal	**[G10]** Allow users to import and export their calendar.
Assumptions	The user is registered and logged in MeteoCal. The user has accepted invitations for one or more events.
Scenario	
Matej and Luka need to make a school project. They want to see both of their calendars on one place, so they will know when both of them are free and can work on the project. Matej logs in to his profile on MeteoCal by clicking on the "Log in" button. He is redirected to the home page of his profile, where he notices a link for exporting calendars. He clicks on the link and the system allows him to export his calendar.	

Scenario 12: User can export calendar.

User can import calendar.	
Code	SC-013
Description	Describes how a user can import calendars.
Goal	**[G10]** Allow users to import and export their calendar.
Assumptions	The user is registered and logged in MeteoCal. The user has accepted invitations for one or more events.
Scenario	
Luka logs in to the MeteoCal, by clicking on the "Log in" button and entering his username and password. Then he navigates to the link for importing calendars and clicks on it. The system gives an option to import other calendars. He chooses to import the calendar from Matej.	

Scenario 13: User can import calendar.

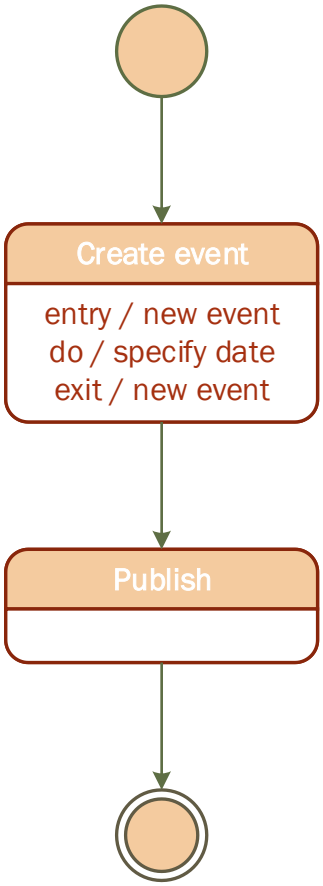
3.2.2 Analysis model



Class diagram 1: Analysis model

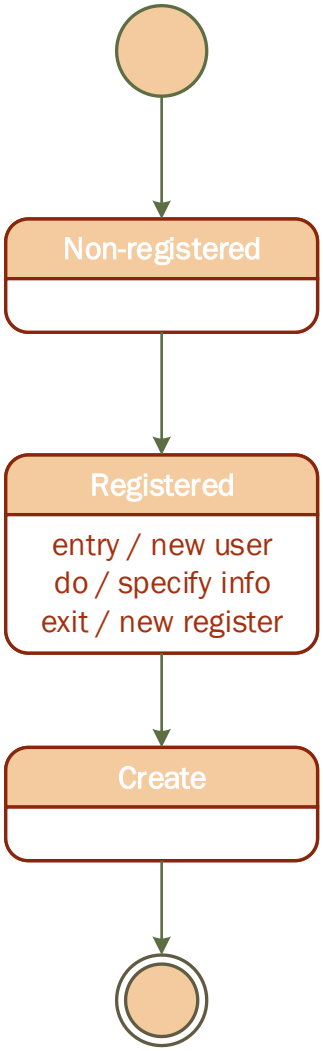
3.2.3 State chart model

3.2.3.1 State diagram of an event



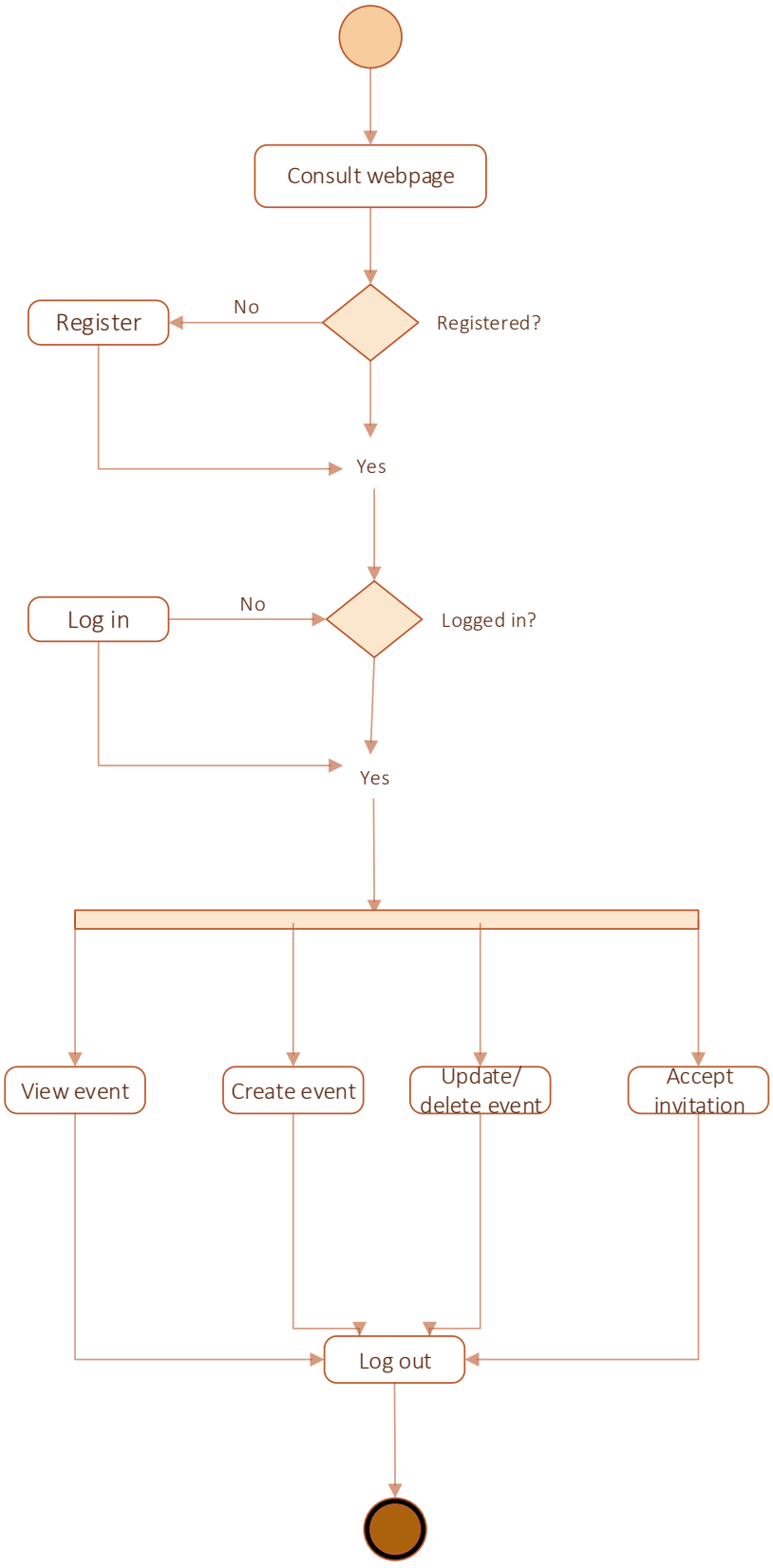
State diagram 1: Event

3.2.3.2 State diagram of a user



State diagram 2: User

3.2.4 Activity Model

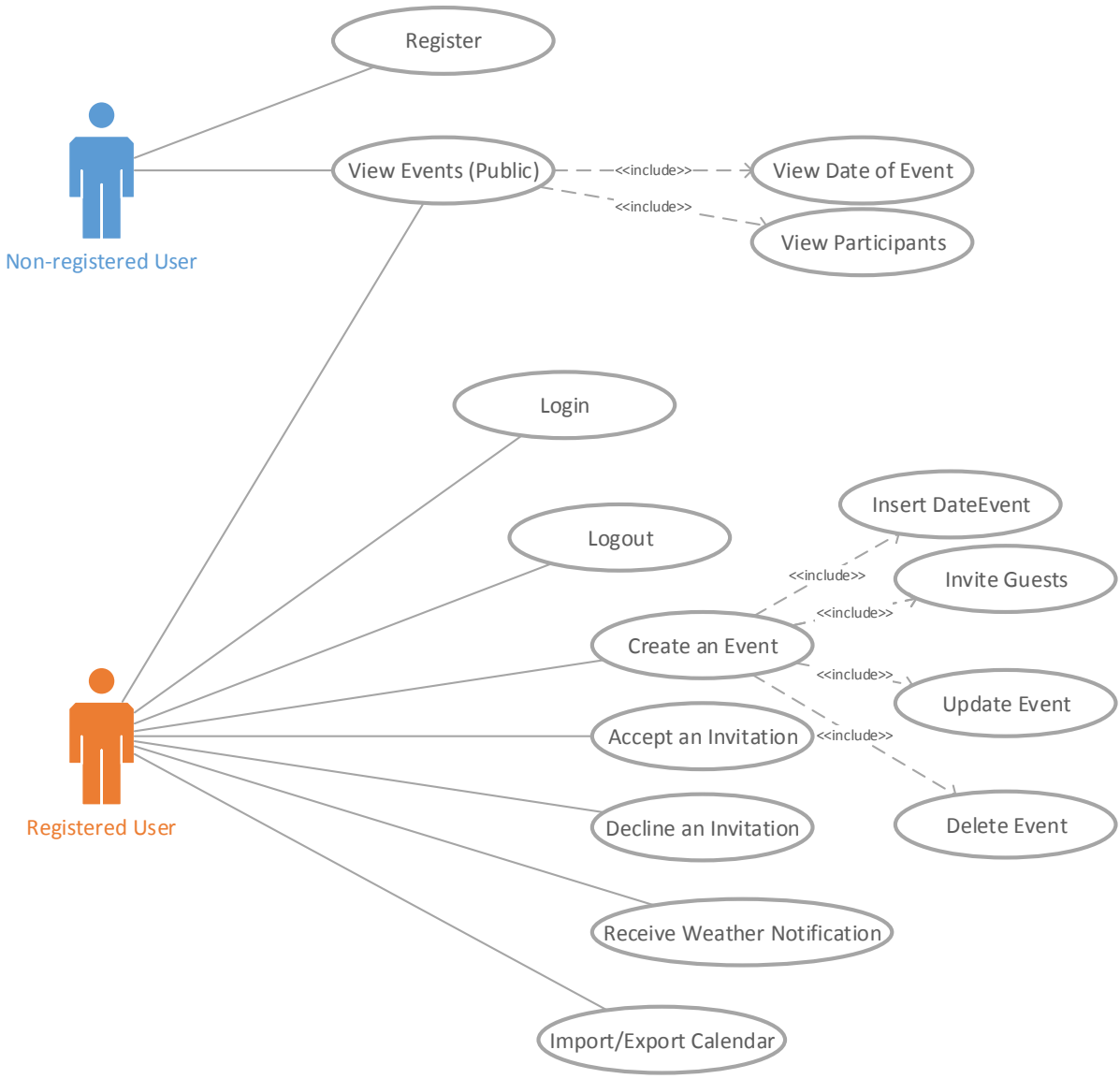


Activity diagram 1: Actions of the user

3.2.5 Use cases

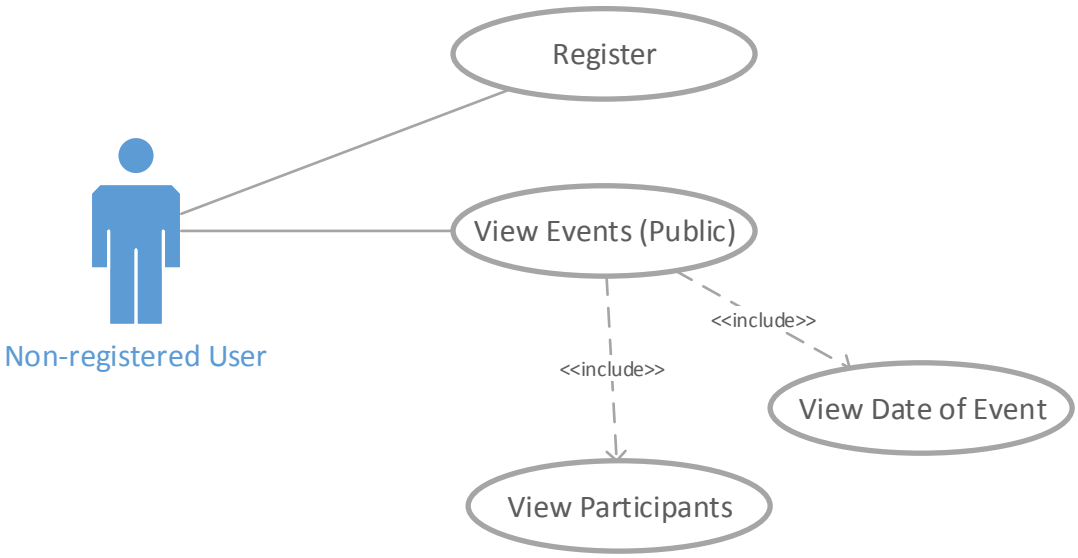
The software contains two main actors:

- Non-registered user
- Registered user (citizen)



Use Case Diagram 1: General use case model.

3.2.5.1 Non-registered User



Use Case Diagram 2: Non-registered user

- **Registering in the system**
  Code: USC-001

  Description: Registering in the system

  Goal: [G1]

  Actors: Non-registered user

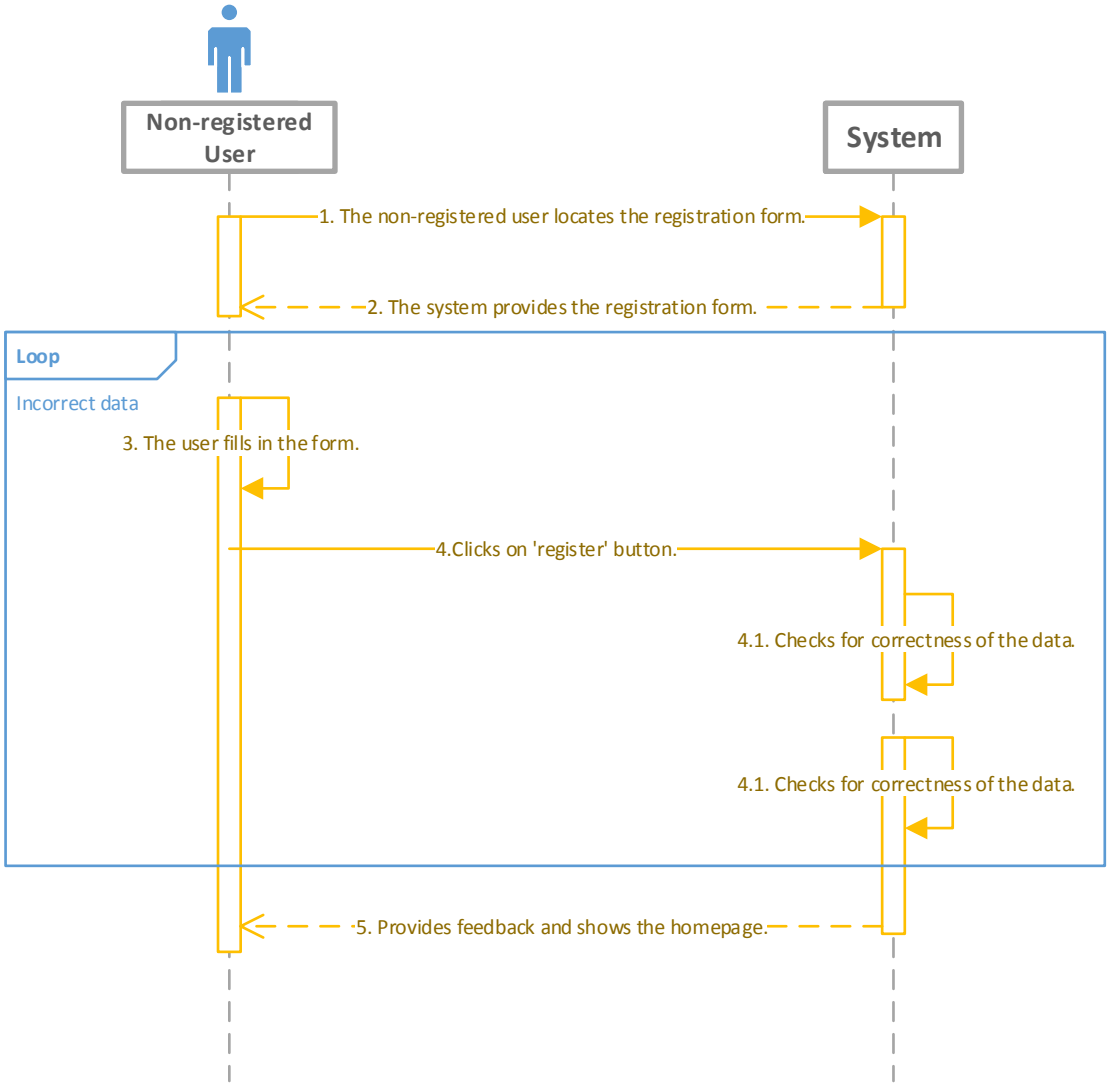
  Entry condition: Non-registered user navigates to the homepage of MeteoCal

  Exit condition: Non-registered user is registered.

  Flow of events:

  1. Non-registered user locates the registration form
  2. The system provides the registration form
  3. Non-registered user fills in the form
  4. Non-registered-user clicks “Register” button
  5. The system provides feedback and shows the homepage

  Exceptions: the user adds incorrect data



Sequence Diagram 1: Registering in the system. (USC-001)

- **View public events**
  Code: USC-002

  Description: View public events

  Goal: [G13]

  Actors: Non-registered user

  Entry condition: Non-registered user navigates to the homepage of MeteoCal

  Exit condition: View a public event.

  Flow of events:

  1. Non-registered user locates the public event.

2. The system provides information about the event.

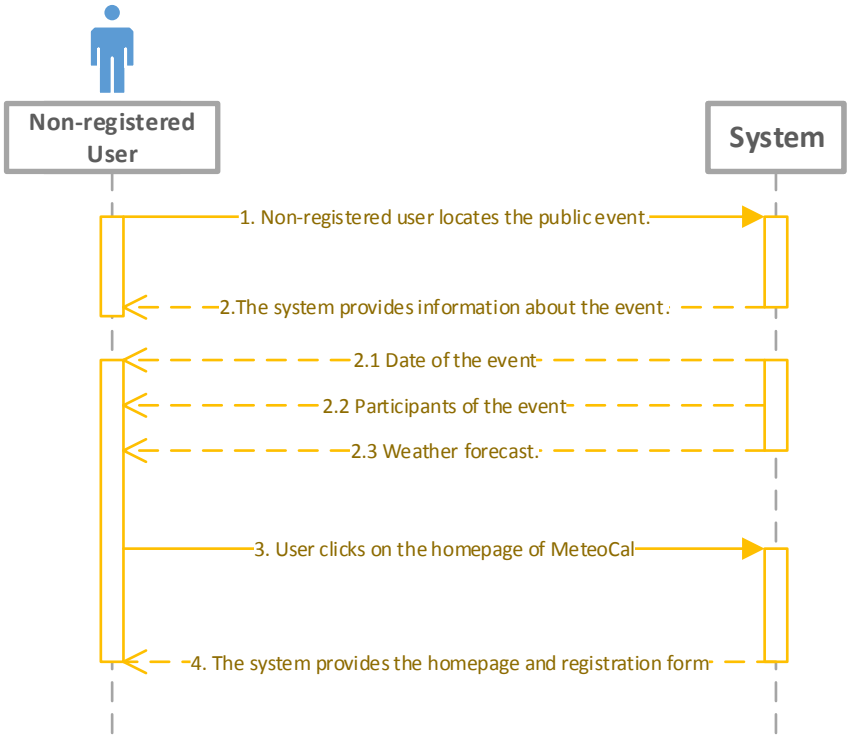
2.1. The system provides the event date.

2.2. The system provides the event participants.

2.3. The system provides the weather forecast.

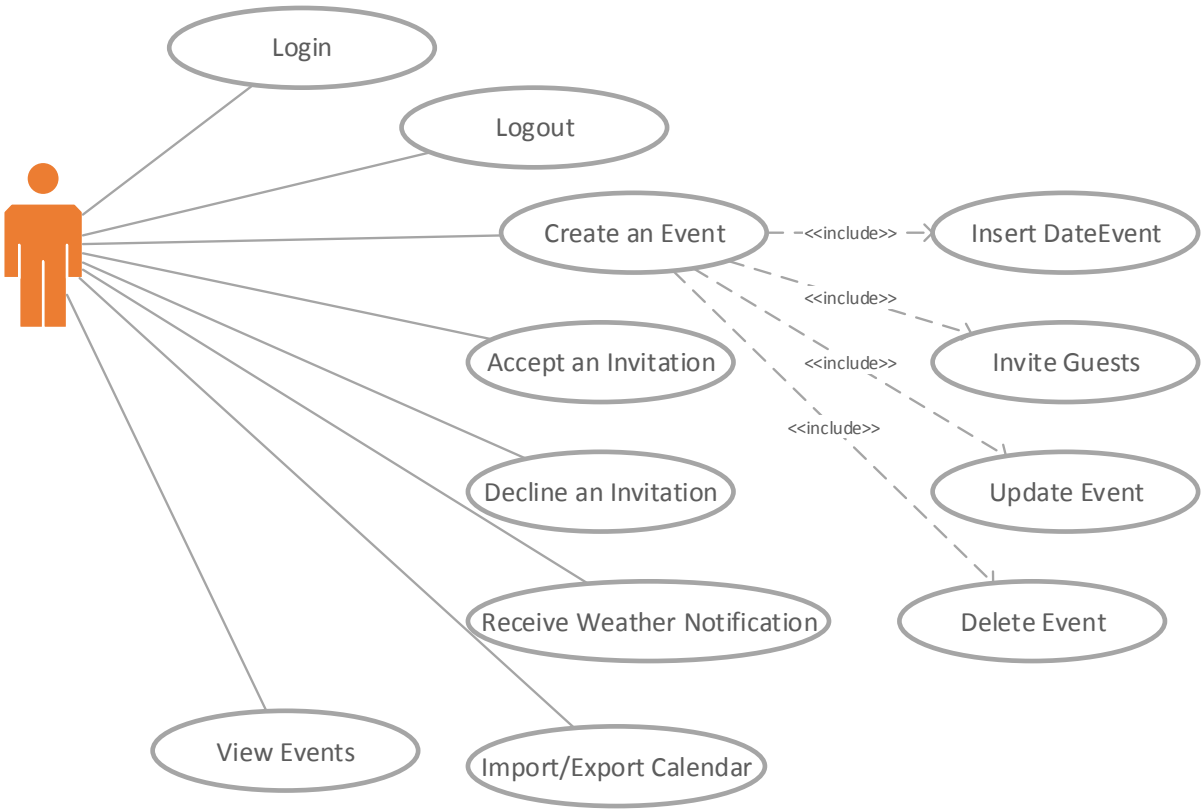
3. User clicks on the homepage of MeteoCal.

4. The system provides a registration form.



Sequence Diagram 2: View public events. (USC-002)

3.2.5.2 Registered User



Use Case Diagram 3: Registered user

- **Registered user login**

  Code: USC-003

  Description: Registered user login.

  Goal:

  Actor: Registered user

  Entry condition: Registered user navigates to the homepage of MeteoCal.

  Exit condition: User is logged in.

  Flow of events:

  1. The registered user locates the login form.

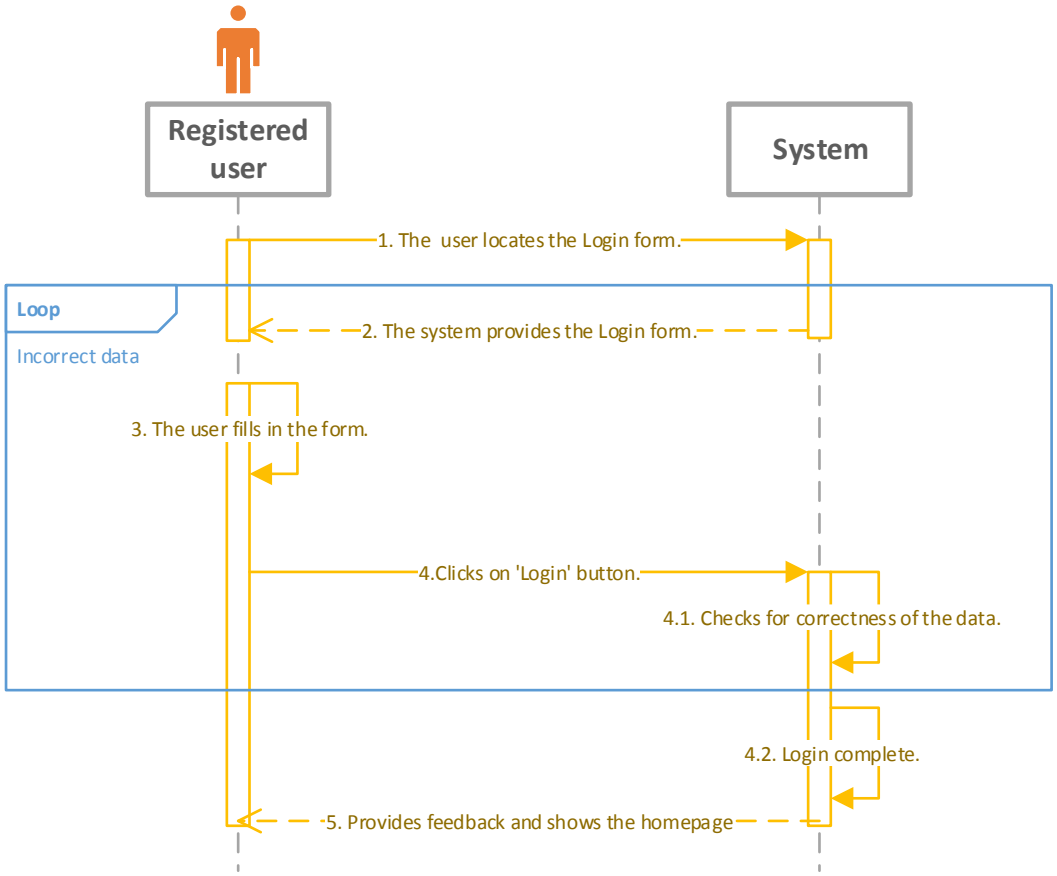
  2. The system provides the login form.

  3. The registered user fills in the form.

4. The registered user clicks “Login” button.

5. The system provides feedback and shows the homepage of MeteoCal.

Exceptions: the user adds incorrect data.



Sequence Diagram 3: Login of already registered user. (USC-003)

- **Registered user logs out from the system**

  Code: USC-004

  Description: Registered user logout.

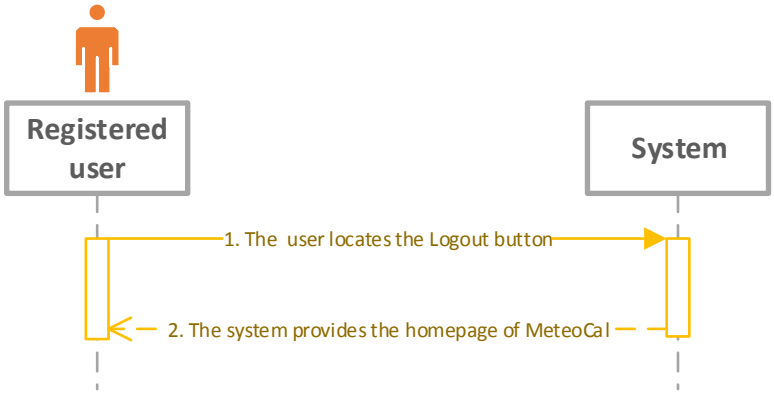
  Goal:

  Entry condition: The user is logged in.

  Exit condition: The user is logged out.

  Flow of events:

  1. The registered user clicks on the logout button.
  2. The system provides the homepage for non-registered users.



Sequence Diagram 4: Registered User logout (USC-004)

- **User creates a new event**

  Code: USC-005

  Description: User creates a new event (becomes an organizer)

  Goal: [G2]

  Entry condition: User is logged in and there is not an event organized by that event at that time.

  Exit condition: A new event is created

  Flow of events:

  There is no ongoing event created by that user at that time.

  1. The user clicks the button “New event”.

  2. The system provides the form for a new event.

  3. The user enters the information about the event (date, time, place).

  4. The user invites guests to the event.

  5. The user submits the event.

  5.1. The system checks for correctness of the data.

  6. Confirmation that the event is created.

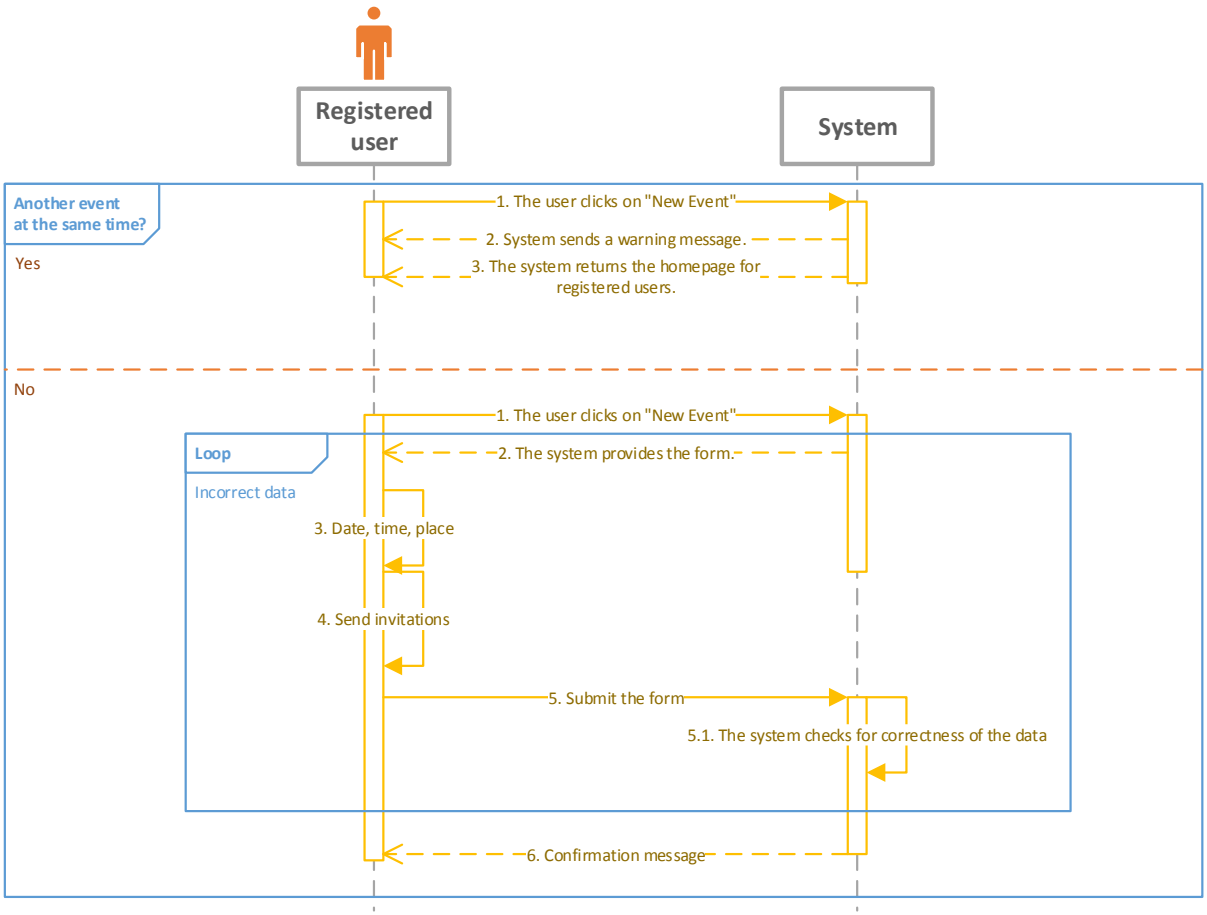
  The user already organizes an event at the same period.

  1. The user clicks the button “New event”.

  2. The system replies a warning message: “You are already organizing another event at the moment”.

  3. The system provides the homepage for registered users.

  Exceptions: wrong user data.



Sequence Diagram 5: Creation of a new event (USC-005)

- **User accepts an invitation**

  Code: USC-006

  Description: User accepts an invitation

  Goal: [G5]

  Entry condition: User is logged in and is invited to an event.

  Exit condition: The user becomes a participant in the event.

  Flow of events:

  1. The user receives an invitation.
  2. The system provides information about the invitation.

  2.1 The system provides the date of the event.

  2.2 The system provides type of the event.

  2.3 The system provides the participants in the event.

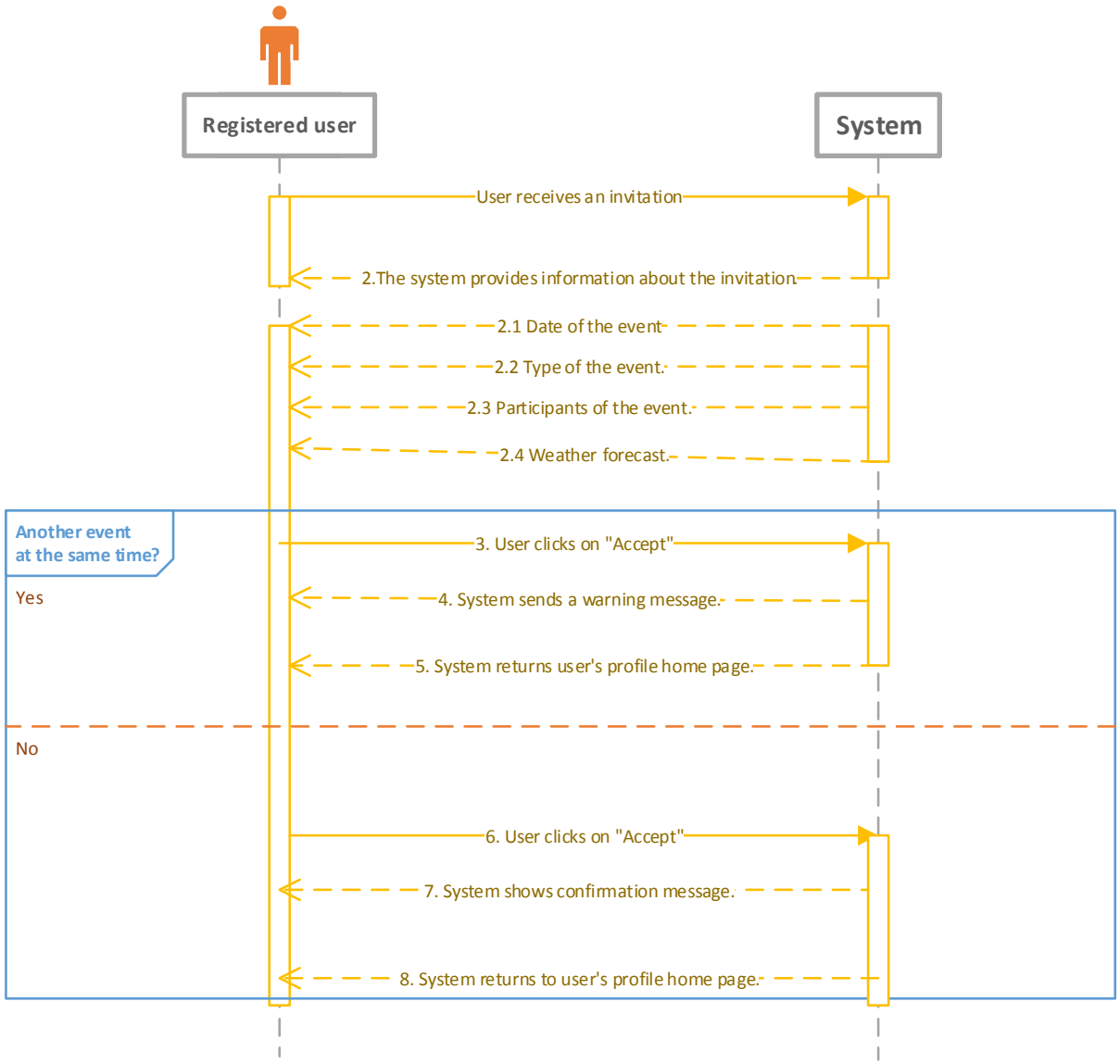
2.4 The system provides the weather forecast for that day.

There is another event at the same time where the user already is attending:

3. User clicks on the ‘Accept’ button.
4. System sends a warning message.
5. The system returns the user’s homepage.

There isn’t another event at the same time where the user already is attending:

3. User clicks on the ‘Accept’ button.
4. System sends a confirmation message.
5. The system returns the user’s homepage.



Sequence Diagram 6: User accepts an invitation.

- **User declines an invitation**

  Code: USC-007

  Description: User declines an invitation

  Goal: [G5]

  Entry condition: User is logged in and is invited to an event.

  Exit condition: The user declines the invitation to the event.

  Flow of events:

  1. The user receives an invitation.
  2. The system provides information about the invitation.

  2.1 The system provides the date of the event.

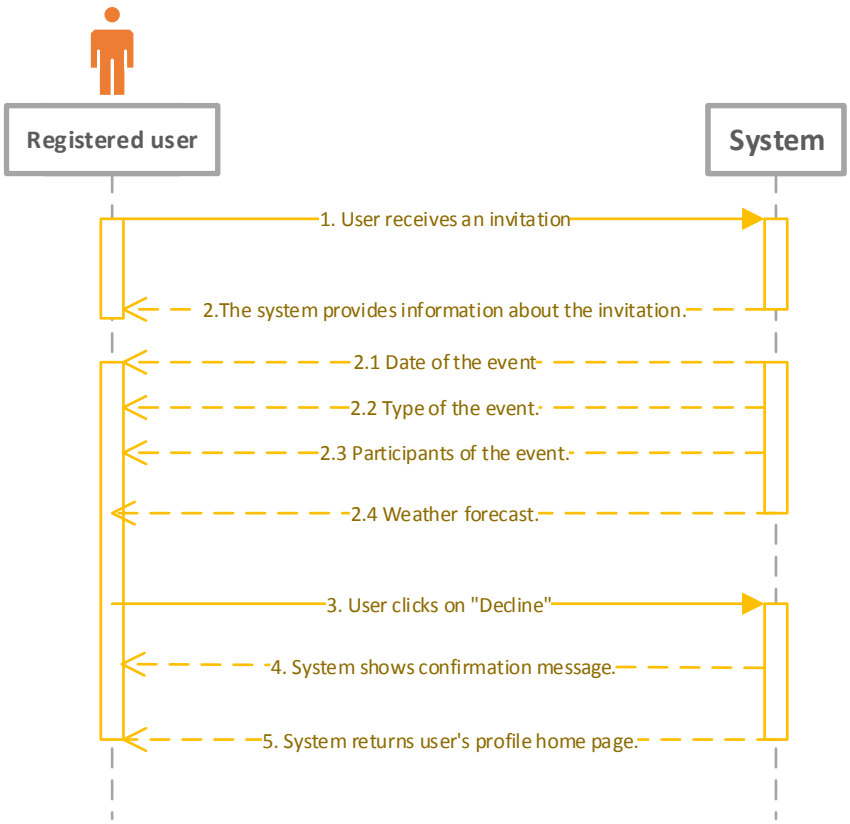
  2.2 The system provides type of the event.

  2.3 The system provides the participants in the event.

  3. The user clicks on the ‘Decline’ button.

  4. The system shows confirmation message.

  5. The system returns the user’s profile home page.



Sequence Diagram 7: User declines an invitation.

- **User receives a weather notification**

  Code: USC-008

  Description: User receive a weather notification.

  Goal: [G8] [G9]

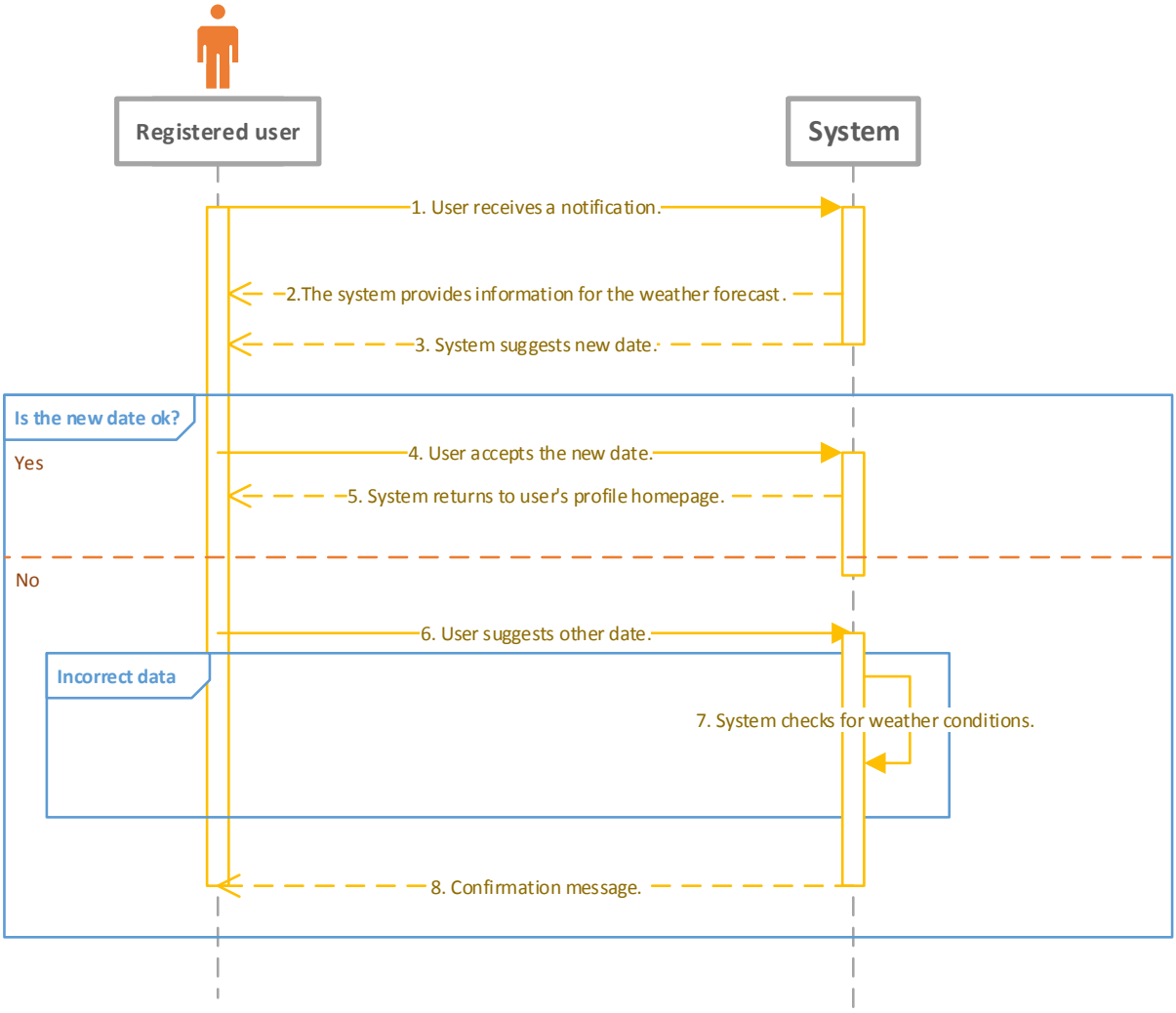
  Entry condition: User is logged in and is participating to an event.

  Exit condition: The user receives a weather notification.

  Flow of events:

  1. The user receives a notification.
  2. The system provides information about the weather forecast.
  3. The system suggests a new date.
  4. The user accepts the new date.
  5. The system returns to the user's profile page.
  6. User suggests other date.
  7. System checks for weather conditions.
  8. The system sends a confirmation message.

Exceptions: User adds incorrect data.



Sequence Diagram 8: User receives a weather notification.

- **User imports/exports calendar.**

  Code: USC-009

  Description: User imports/exports calendar.

  Goal: [G10]

  Entry condition: User is logged in.

  Exit condition: The user imports/exports calendar.

  Flow of events:

  1. User clicks on “Import” or “Export”.
  2. System responds with importing or exporting calendar.

Sequence Diagram 9: User imports/exports calendar.

3.3 Performance Requirements

The software product requires that each web page shall load in less than 15 seconds. Also, as mentioned before, it is necessary to support a high level of concurrent users.

3.4 Logical Database Requirements

The database must save all the entities identified in the analysis diagram, because they are necessary for managing the software product information.

3.5 Design Constraints

The software product must be designed and implemented in JEE and its related technologies.

3.6 Standard Compliance

The software product must be developed following recommended standards in order to be easily readable and updatable.

3.7 Software System Attributes

3.7.1 Reliability

The system shall assure the integrity of the users.

3.7.2 Availability

The system should be available 24 hours per day, 7 days per week, and 365 days per year.

3.7.3 Security

The software product must encrypt users' password. This can be achieved using cryptographic techniques.

3.7.4 Maintainability

The database of the software product should be backed up periodically, so that in case of failure existing data can be reconstructed easily. The system needs maintainability in clearing the users that are inactive for longer periods, server files system backup and periodical cleaning of old events, in order not to affect the performance of the system.

3.7.5 Portability

The software product can be installed on any operating system which supports Java Virtual Machine and its dependent components.

3.8 OTHER REQUIREMENTS

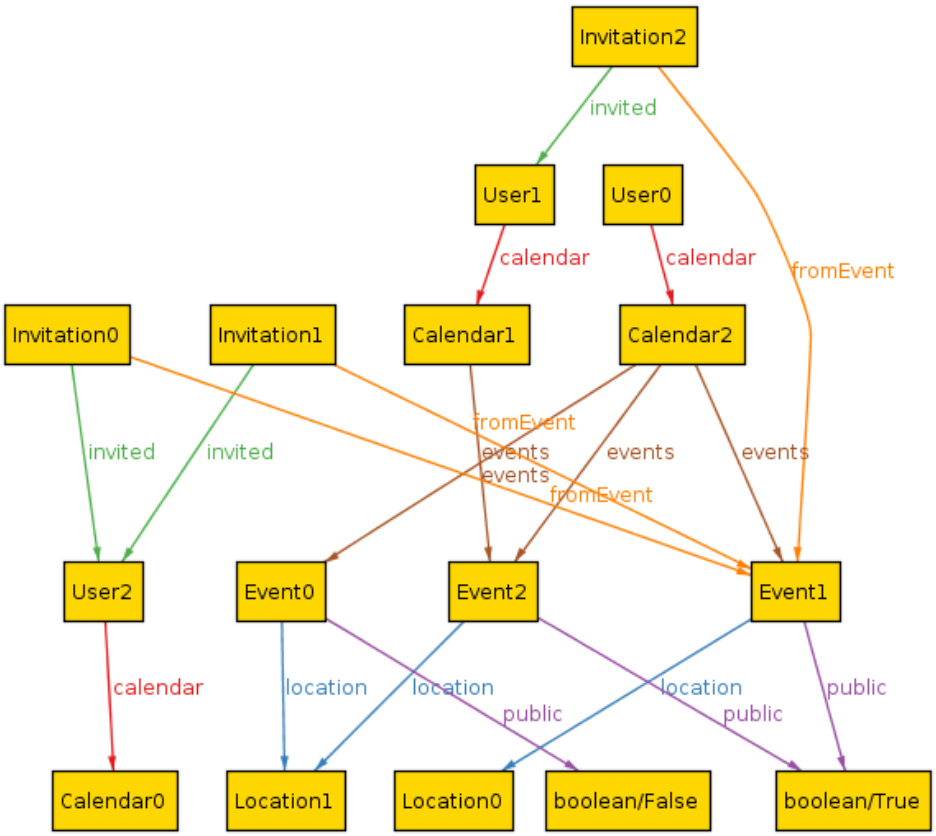
The software product must provide understandable messages in text form in the event of an error and instruct the user on what to do.

4 APPENDICES

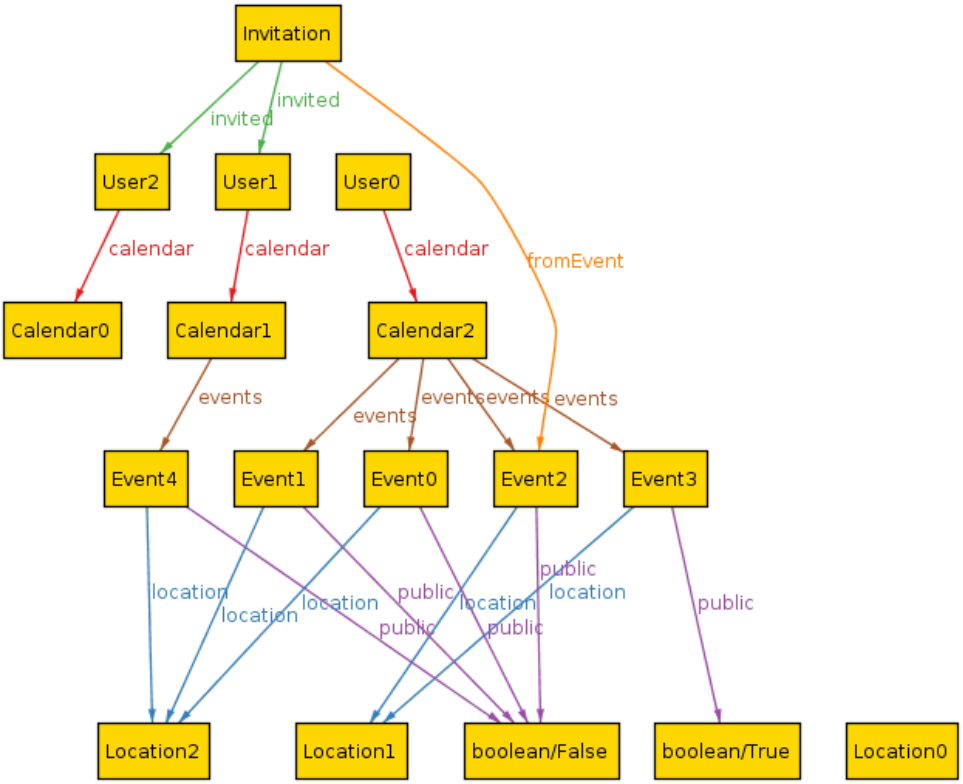
4.1 ALLOY MODEL

Alloy model file: meteocal.als

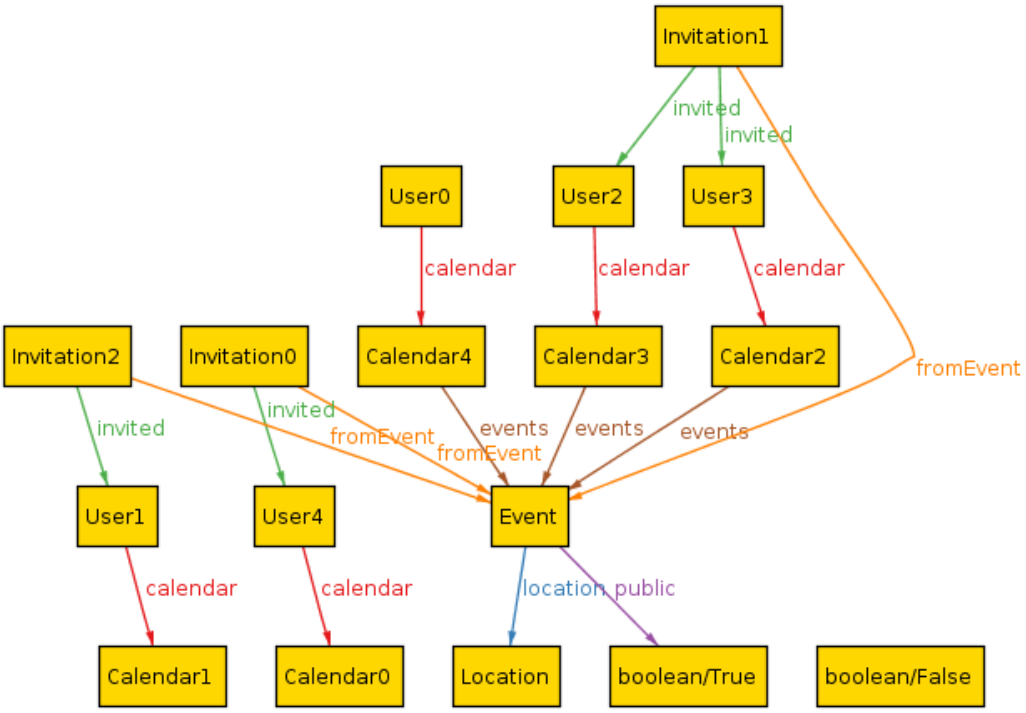
Following there are some screenshots from the alloy model.



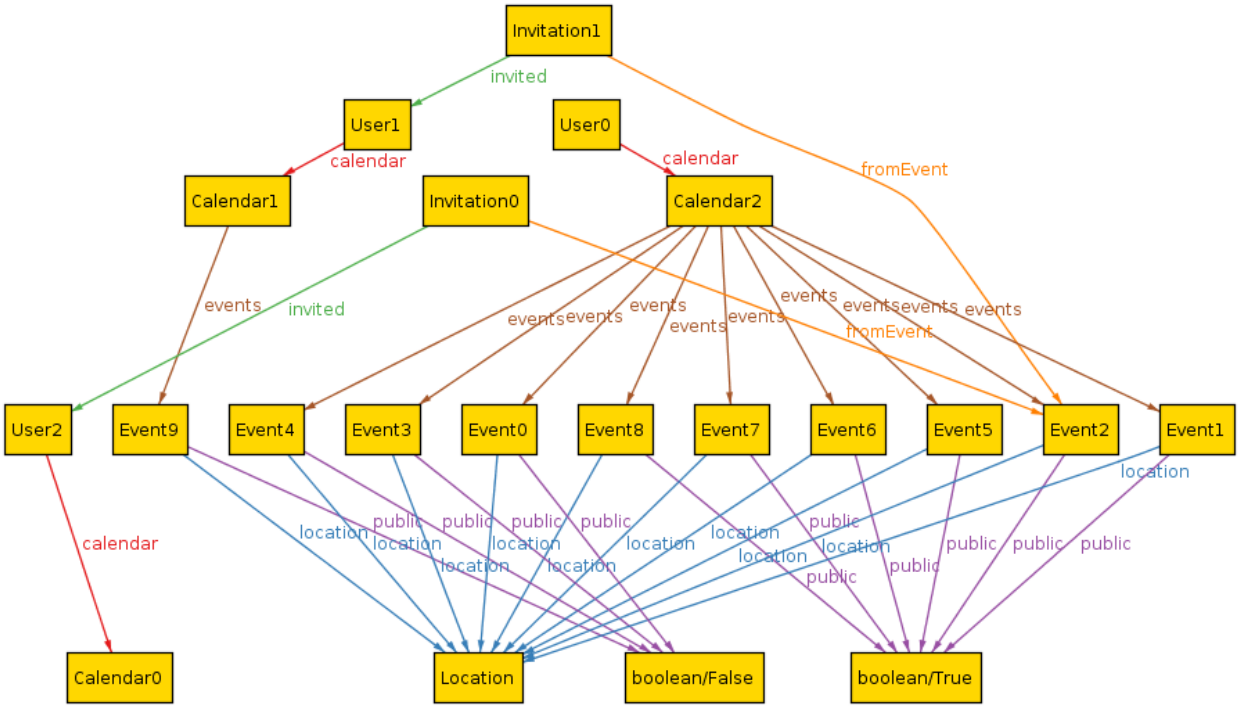
Alloy instance 1: Five instances of every signature.



Alloy instance 2: One invitation, multiple users, multiple events.



Alloy instance 3: One event and multiple users.



Alloy instance 4: The number of events is greater than the number of users.