Calendar System

ButterFree

September 26, 2012

Contents

1 Vision

Our vision for the Calendar System is to create an easy to use multiuser scheldule for organizing meetings. We realise that multiple simulair services already exist. What we want to do, is create a bridge between calendar systems. We want people, who use differnt calendar applications, to plan together.

2 Use cases

ID	Description	Notes	Priority
1	The user creates an appointment with another user.	The target can either be in our calendar system or in another iCal compatible system.	High
2	The user accepts an invite to an appointment with another user	Same as above	High

Table 1: Our use cases

2.1 Use case 1

The user wish to create an appointment with another user. The user must first create the appointment in his own calendar system. He then selects the event on our website, and chooses "Invite Attendee". A list of contacts are shown, and the user can select on ore more contacts he wants to invite. The user can add a text to the invitation. The people invited will be notified. The user will receive notification when each invited attendee accepts or declines.

2.2 Use case 2

The user has gotten an invitation to an appointment with another user, and he wants to reply. The user logs on to our website, and finds the invitation in his notifications. The user can press either 'Accept' or 'Decline'. The user can add a text to his reply.

3 Use Cases UML Diagram

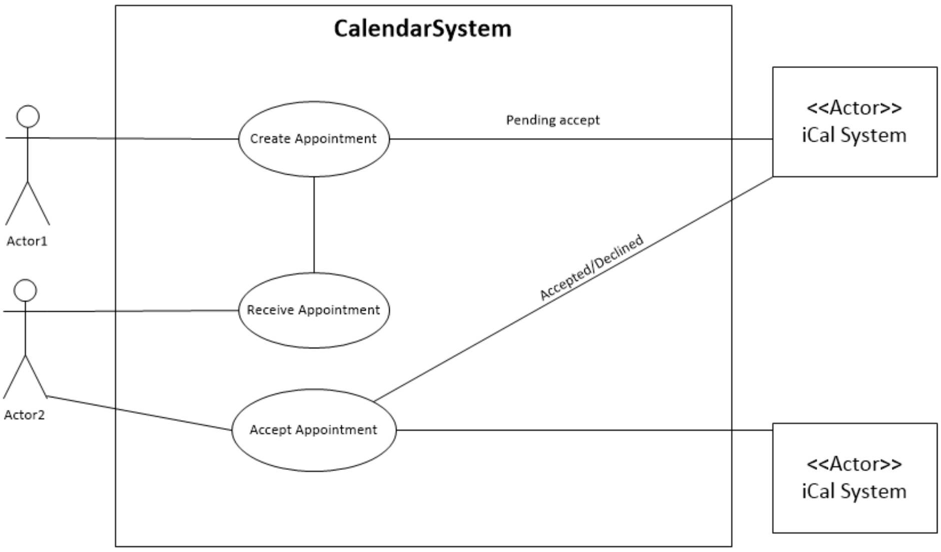

Figure 1: The UML diagram for our use cases

4 Glossary

Word	Description
Appointment	An appointment with the data that can be stored in the iCal format
iCal	See iCalendar
iCalendar	iCalendar is a computer file format which allows Internet users to send meeting requests and tasks to other Internet users
Share	Send an appointment to another user via the iCal format
User	A user of an iCal compatible system. When several users are mentioned the first one is the user of our system

Table 2: Our glossary explaining the terms we use

5 Supplementary Requirements

In this section we will cover any supplementary requirements.

5.1 Functionality

No addition requirements.

5.2 Usability

No addition requirements.

5.3 Reliability

- Our system must be available to our users at least 90% of the time

5.4 Performance

- Our system must respond in less than 1 sec in 90% of the cases. (External system are out of our control, and this time is therefore subtracted)

5.5 Supportability

- Compatible with all iCal-systems.

6 Domain Model

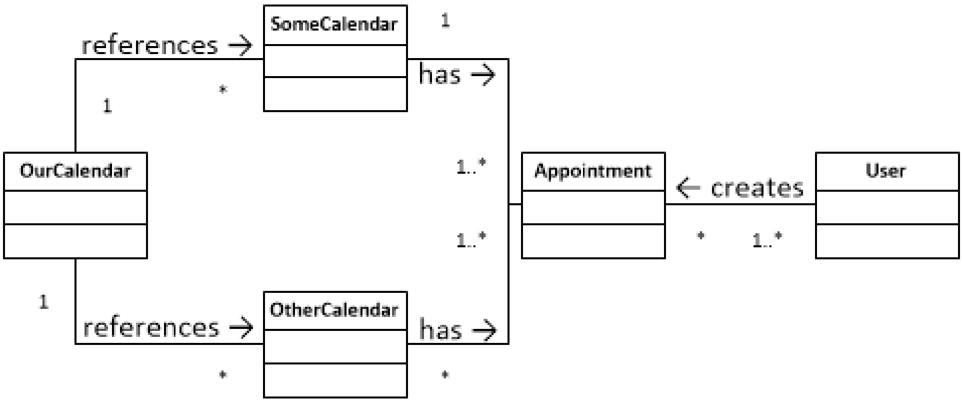

Figure 2: Model illustrating our domain and its relations

7 System Sequence Diagram

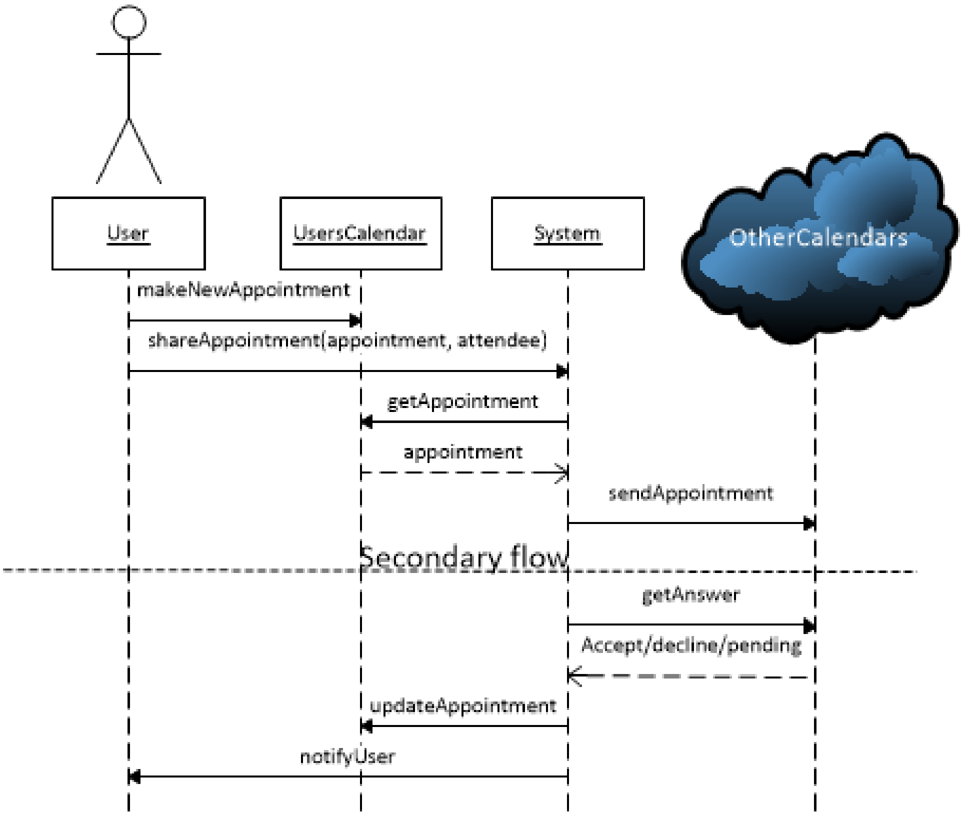

Figure 3: The sequence diagram for our system

8 Operation Contract Template

8.1 OCT 1

Operation:	shareAppointment(appointment, attendee)
Cross references:	Primary flow in SSD diagram
Preconditions:	User is registered in our system. Appointment must exist Other users calendar must be registered in our system
Postconditions:	- OtherCalendar now has the appointment - SomeCalendars appointment is synchronized with the appointment in OtherCalendar - The appointment now has the status of the attendees

8.2 OCT 2

Operation:	getAnswer()
Cross references:	Secondary flow in SSD diagram
Preconditions:	There is an appointment that needs monitoring (For status changes or synchronization)
Postconditions:	- The appointment in the users calendar is updated - The other attendees has been notified