IT-Universitetet i København

Analysis, Design and Software Architecture

Bachelor in Software Development

Calendar system

kvin_thgn_AS3̇8:

Kristian Mohr Nielsen - kvin@itu.dk
Tróndur Høgnason - thgn@itu.dk

Lecturers:

Rasmus Nielsen
Jakob Eyvind Bardram

6. oktober 2014

Indhold

1 Introduction

The purpose of the calendar system is to make a calendar system for a workplace, capable of sharing entries and organizing meetings. Accounts and workgroups will be handled by an administrator.

1.1 Scope of the system

The system is intended to be used in a workplace environment, where people know each other, or at least are acquainted in some way. Thus the program has no need for contact lists and blocking of people, since it is used by people working together, to organize meetings etc.

1.2 Objectives and success criteria of the project

The objective is to implement a sytem that is easy to use and learn for the user. As the system is supposed to replace the paper calendar it should also be quick to create to update entries.

The criteria for this is:

- In a user test 70% of participants must be able to use all of the functions in the system available to them within an hour.
- In a user test 65% of participants must be able to create an entry in their calendar in under one minute.
- From the time an user confirms the creation or update of an entry, the system may at most use 5 seconds to complete the request.

2 Current system

The currents system used is a physical calendar made of paper. The problem with this system is the need to change the calendar every year. The current system also requires the users to bring the calendar with them between destinations, since the calendar is not available through other means. It is also bothersome to share entries with other people, as this must be done with a phone or over email.

3 Proposed system

The proposed system is client-server calendar application capable of syncing with other calendar servers. It's able to have entries for one or multiple persons. Persons are part of workgroups, making it easier to manage large entries, since it's possible to invite a workgroup instead of all individual persons.

3.1 Functional requirements

The system supports two kind of users:

The User: Able to manage entries in the calendar, which includes creating, updating and deleting entries. It should also be possible to add other users to the entry, during creation and after.

The Administrator: Able to manage accounts, which include creating, updating and deleting entries. An administrator should also be able to create, update and delete workgroups.

3.2 Nonfunctional requirements

Category	Nonfunctional requirements
Usability	The UI of the client must resemble a paper calendar to ease the learning process.
Reliability	Loss of connection to the server must only cut off functions requiring connection, but not access to information already loaded.
Performance	• The server must support multiple clients at once, a minimum of 1000 should be able to use the system at the same time. • The communication between client and server, when updating a server should be fast even with a low bandwidth connection.
Supportability	...
Implementation	The system should be implemented in C# and work on all newer versions of Windows.
Operation	...
Legal	...

3.3 System models

3.3.1 Scenarios

Scenario name: accountForNewUser
Participating actors: Jon:User, Ben:Administrator
Flow of events:

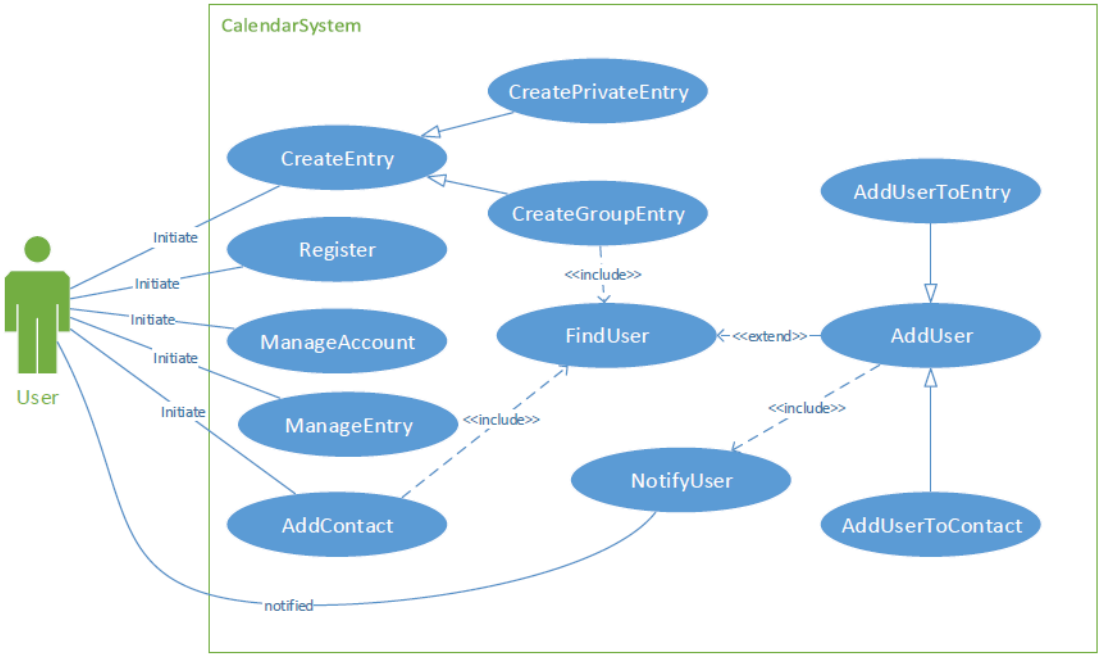
- Jon joins the firm, and thus requires access the the calendar system.
- He contacts Ben, the administrator, by mail or in person to ask for access.
- Ben registers a new account in the calendar system for Jon.
- Ben puts Jons new account in all relevant workgroups, such as the workgroup for Jons department, the whole workplace etc.
- Ben contacts Jon with the login and now Jon has access to the calendar system

Scenario name: createEntryForMeeting
Participating actors: Jon:User, Mette:User
Flow of events:

- Jon wants to hold a meeting and as such he starts to create an entry in his calendar filling in the information regarding time and date, and subject.
- Since the meeting is for a group of people, Jon wants everyone going to the meeting to see it in their calendar. Thus he invites Mette, and the the people from his department, the workgroup HR.
- Jon then creates the entry, causing the system to show the entry in all invited persons calendar along with a notification.
- Mette unfortunately has another appointment at the time and dismiss the entry, removing it from her calendar.
- The entry is then updated to show who is able to come.

3.3.2 Use case model

Below is the use case model



Figur 1: Use case model

3.3.3 Use cases

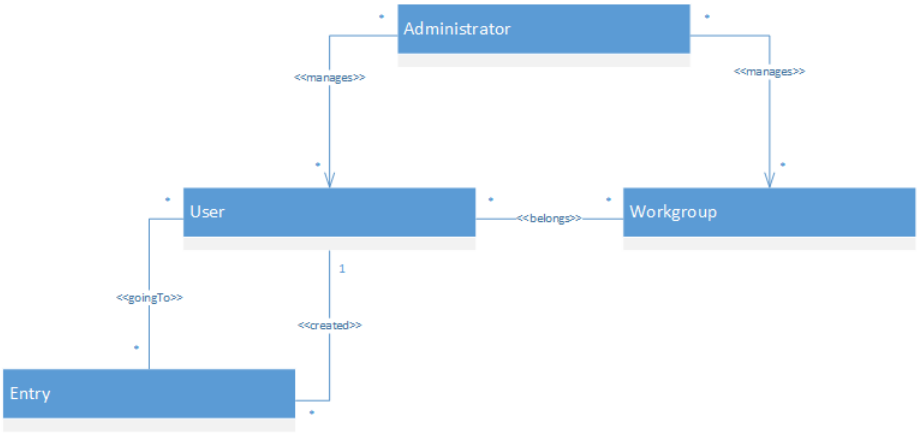
Use case name	CreateAccount
Participating actors	User, Administrator
Flow of events	• The Administrator activates the CreateNewAccount function from a computer. • The calendar system presents a form to the Admin • The Administrator fills out the form by filling out the different kinds of information about the User such as name, phone number, email and adds the user to one or more workgroups. • The calendar system receives the form and notifies the Administrator that the form has been received and the account created.
Entry condition	A request for a new calendar account has been made. Someone in the workplace contacts an Administrator and supplies the Administrator with the required information about the User.
Exit condition	The account has been created for the User, and the User has been notified.

Use case name	CreateEntry
Participating actors	Initiated by User - communicates with User
Flow of events	• The User activates the CreateNewEntry function from a computer. • The calendar system presents a form to the User. • The User fills out the form by filling out the different kinds of information about the new Entry such as name of entry, time, expected duration and optionally location. The User can then choose to add one or more participants (users) or workgroups (groups of users) to the Entry. • The calendar system notifies all the users that the User has added them to an Entry.
	• The calendar system further notifies the User that the form has been received that all Users have been notifed and the Entry created.
Entry condition	A user chooses to create a new Entry, opens the program from a computer and logs into the calendar system.
Exit condition	The Entry has been created and the User has been notified of this. Furthermore all invited users have been notified and their respective calendars have been updated.

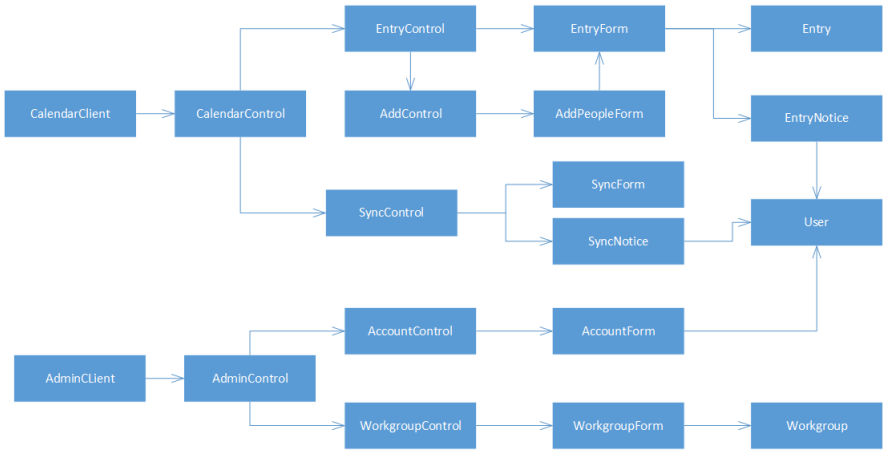
Use case name	SetUpSync
Participating actors	User
Flow of events	• The User activates the SetUpSync function from a computer. • The calendar system presents a form to the User. • The User fills out the form by choosing one of the supported external calendars (such as google calendar, iCal) to synchronize with. The User then provides the calendar system with the username and password for the external calendar system. • The calendar system receives the form and notifies the User that the form has been received and the Calendars are in the process of being synchronized. • Finally the calendar system notifies the user that the synchronization is finished.
Entry conditions	A user who already has an existing external calendar chooses to set up a synchronization with it, opens the program from a computer and logs into the calendar system.
Exit condition	The calendar system has imported all entries from the external calendar and has set up the chosen synchronization time interval with the external calendar.

3.4 Object model

The models below shows a static view of the proposed calendar system with its classes shown as blue squares.



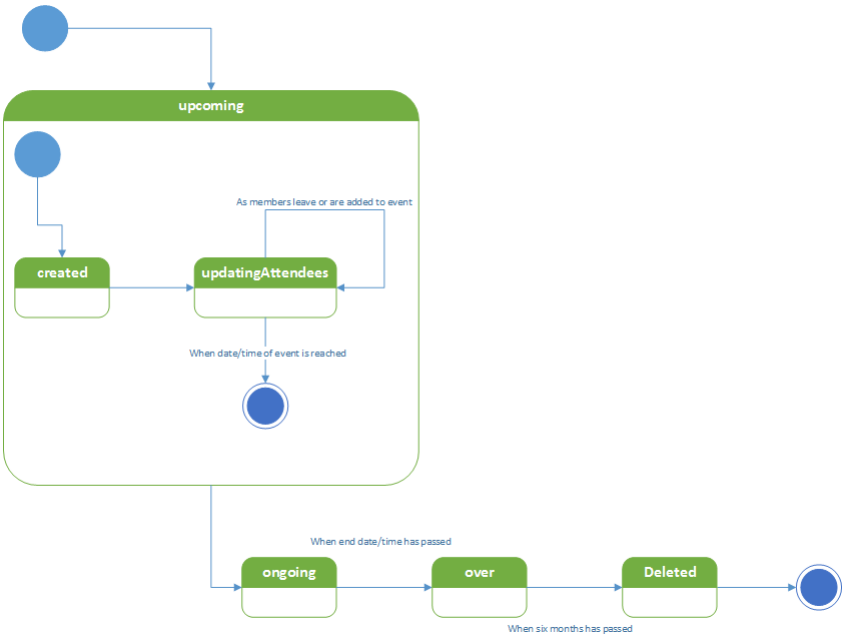
Figur 2: Class diagram, showing entity relations



Figur 3: Class diagram, showing relations between control, boundary and entity objects

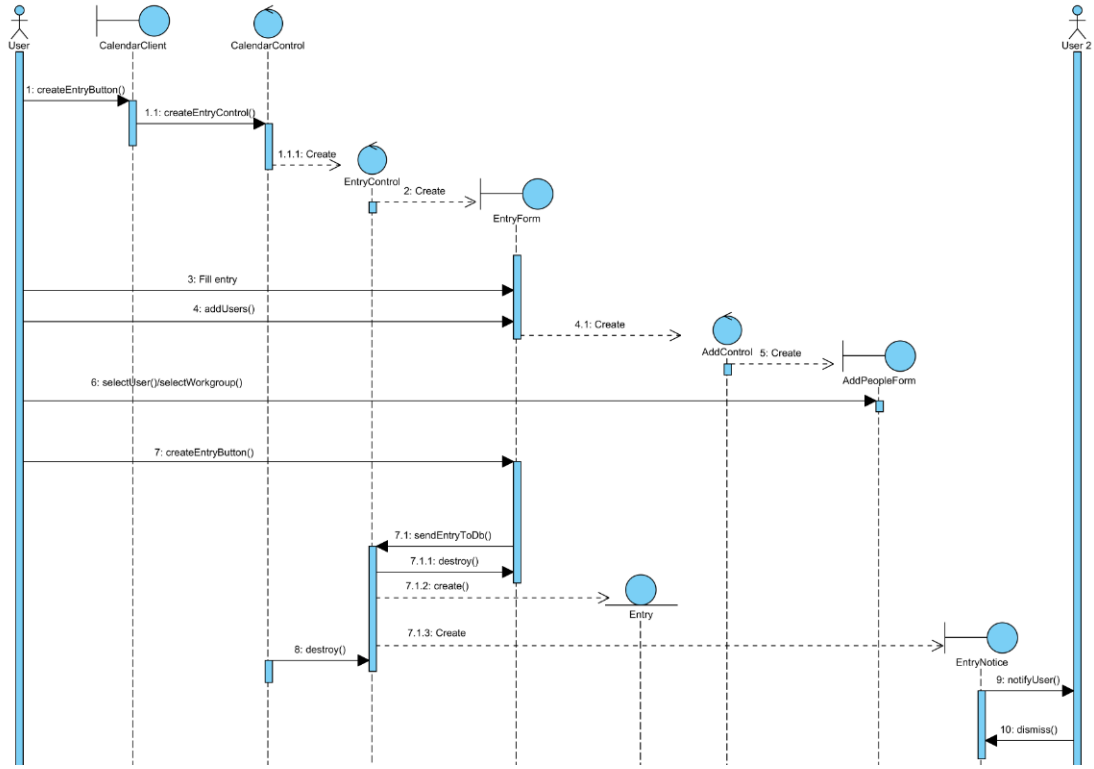
3.5 Dynamic models

Below is a statemachine diagram for an entry, showing which states it goes through.

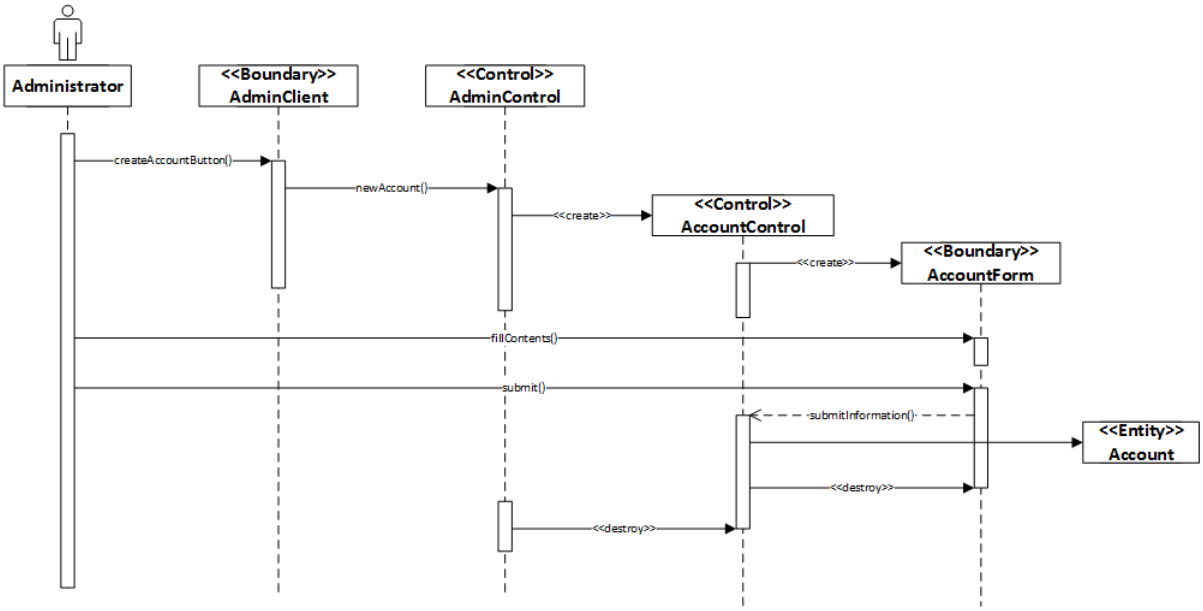


Figur 4: State machine diagram - entry

The models below show two squence diagram of the use cases CreateEntry and CreateAccount. They show how the proposed calendar system will behave on runtime when creating an entry and creating an account.



Figur 5: Sequence diagram - CreateEntry



Figur 6: Sequence diagram - CreateAccount

4 Glossary

Entry	An event written in the calendar, containing information regarding the event, and a date and time.
User	A user who manages entries, the user created. The user is also able to see entries in the calendar, which the user is invited to. The user is also able to set up automatic synchronization with a personal calendar.
Administrator	A user managing accounts in the calendar system, as well as managing workgroups.
Workgroup	A collection of users connected in some way, e.g. department of a firm, who can be added to an event collectively.