Assignment 37

Tróndur Høgnason Kristian Mohr Nielsen

14. september 2014

1 Introduction

The purpose of the calendar system is replace the paper calendar with an online calendar available for user through a client on a computer.

1.1 Scope of the system

The calendar will allow users to access their calendar from any computer with the client installed and an internet connection, since their calendar is bound to an account. Users will be able to create and manage entries in their calendar and share entries with other users.

1.2 Objectives and success criteria of the project

The objective is to implement a easy to use, and learn, system, for the user. The system should also be fast to create/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/update of an entry, the system may at most use 5 seconds to complete the request.

1.3 Definitions, acronyms, and abbreviations

User: In this document the term user covers any regular user with no special privileges in the system.

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.

3 Proposed system

The proposed system is made of a system in two parts. The first part is the client, running on a regular computer, the second part is a server running on a centralised computer, available to all running clients through the internet.

3.1 Functional requirements

The calendar system supports 2 different kinds of users:

- Users who can manage entries in their calendar and manage their account.

3.2 Nonfunctional requirements

Category	Nonfunctional requirements
Usability	The UI of the client must resemble a paper calendar, as to ease 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 (e.g., 25). 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: Create entry
Participating actors: John (user)
Flow of events:

- John is invited to meeting the 21st of July 2015. To remember it he wants to make an entry in his calendar. He accesses the create new entry function in the calendar system.

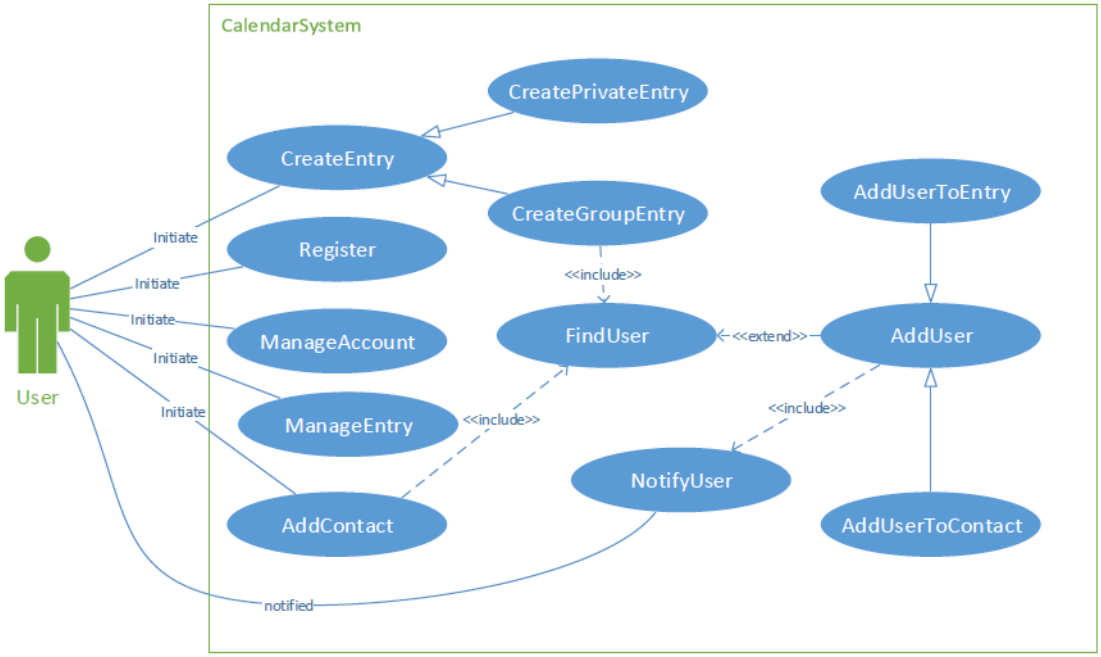
- John plots the information of the meeting and the date/time. He confirms the input and waits for the calendar to create the entry.
- The calendar system tells John that the entry has successfully been created.

Scenario name: Create account
Participating actors: Alice (user)
Flow of events:

- Alice has a lot of appointments, so she wants to create an account to the calendar system. She accesses the createUser function in the calendar system.
- Alice plots in all required information (for example login information and name). She then confirms the information and wait for the system to confirm the creation of her account.
- The calendar system confirms the notify Alice that her account has been successfully created.

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	CreateGroupEntry
Participating actors	Initiated by User - communicates with User
Flow of events	• The User creates a GroupEntry • The User searches for other Users and add them to the GroupEntry (Include use case FindUser and AddUser) • The User confirms input in the entry. • The system confirms entry has been created. • The system sends a notice to all added Users (Include use case NotifyUser).
Entry condition	The User creating the entry is logged in to the CalendarSystem
Exit condition	Entry has been created A notice has been sent to all added Users

Use case name	AddContact
Participating actors	Initiated by User - communicates with User
Flow of events	• The User opens the AddContact function. • The User searches for the User to add (Include use case FindUser). • The User adds the found User, to the ContactList (Include use case AddUser). • The system sends a Notice to the added User.
Entry condition	The User adding another User is logged on
Exit condition	The Users ContactList contains the added User A notice has been sent to the added Users

Use case name	Register
Participating actors	Initiated by User
Flow of events	• The User opens the register function and plots in any required information. • The User confirms the information and waits for the system. • The system notifies the User that the account has been successfully been created.
Entry conditions	The User has installed the Client part of the system
Exit condition	An account has been created for the User

4 Glossary

Entry	An entry in the calendar on a given date and maybe at a specific time . It can be created as a PrivateEntry showing only in the calendar of the creator or as a GroupEntry showing in the calendars of all added Users.
User	An user of the system who has a calendar the user is able to modify with entries.
Account	The account of an User, giving access to the Users calendar.
Notice	A notice is sent to a User whenever the User has been added to an entry or as a contact.
ContactList	A list of a Users contacts.