Requirements Analysis Document (RAD) for 121Calendar

Mads Frederik Madsen - mfrm, Holger Stadel Borum - hstb and Paw Hwsgaard Laursen - pawh

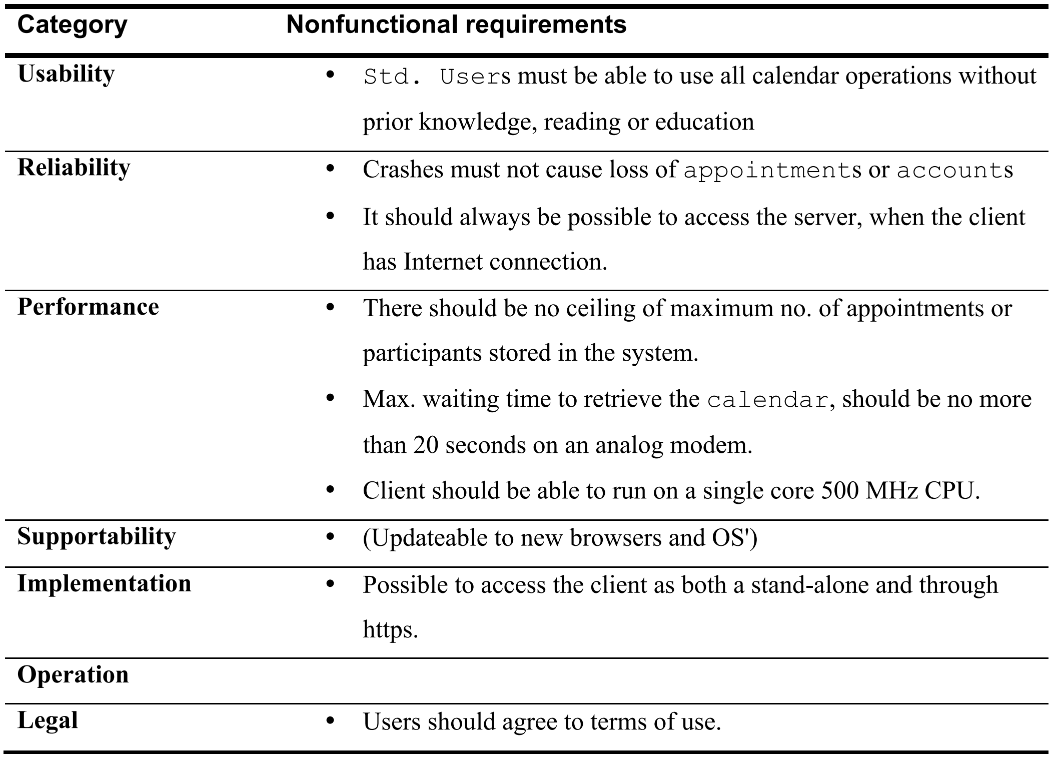
September 15, 2014

Contents

1 Current system

2 Proposed system

2.1 Nonfunctional Requirements

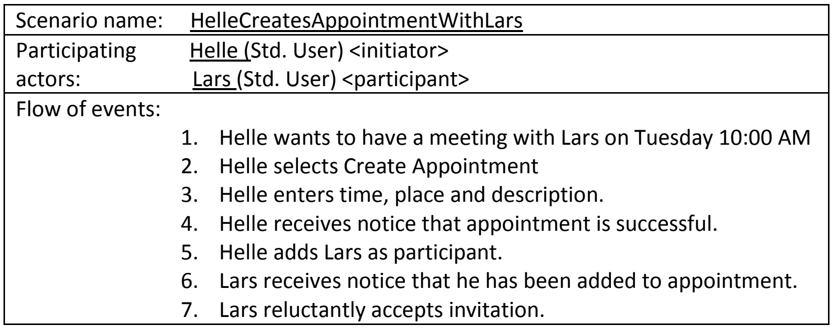


Category	Nonfunctional requirements
Usability	• `Std. User`s must be able to use all calendar operations without prior knowledge, reading or education
Reliability	• Crashes must not cause loss of `appointment`s or `account`s • It should always be possible to access the server, when the client has Internet connection.
Performance	• There should be no ceiling of maximum no. of appointments or participants stored in the system. • Max. waiting time to retrieve the `calendar`, should be no more than 20 seconds on an analog modem. • Client should be able to run on a single core 500 MHz CPU.
Supportability	• (Updateable to new browsers and OS')
Implementation	• Possible to access the client as both a stand-alone and through https.
Operation	
Legal	• Users should agree to terms of use.

2.2 System models

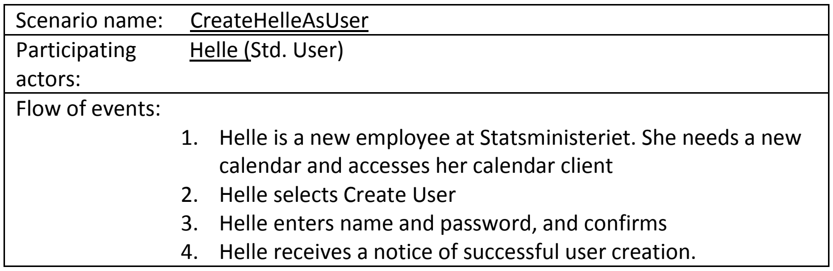
2.2.1 Scenarios

Scenario 1: Create Appointment: -



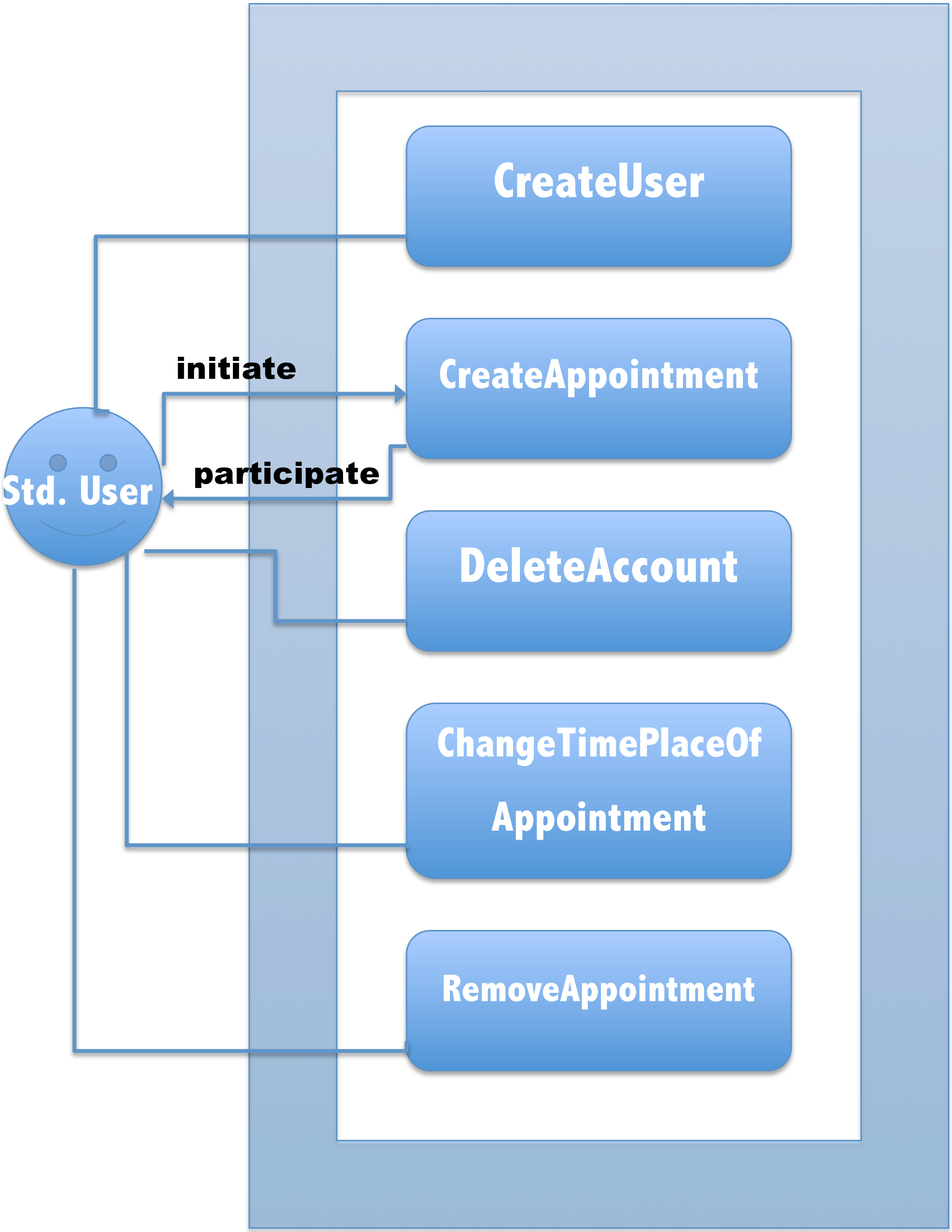
Scenario name:	HelleCreatesAppointmentWithLars
Participating actors:	Helle (Std. User) <initiator> Lars (Std. User) <participant>
Flow of events:	1. Helle wants to have a meeting with Lars on Tuesday 10:00 AM 2. Helle selects Create Appointment 3. Helle enters time, place and description. 4. Helle receives notice that appointment is successful. 5. Helle adds Lars as participant. 6. Lars receives notice that he has been added to appointment. 7. Lars reluctantly accepts invitation.

Scenario 2: Create User: -



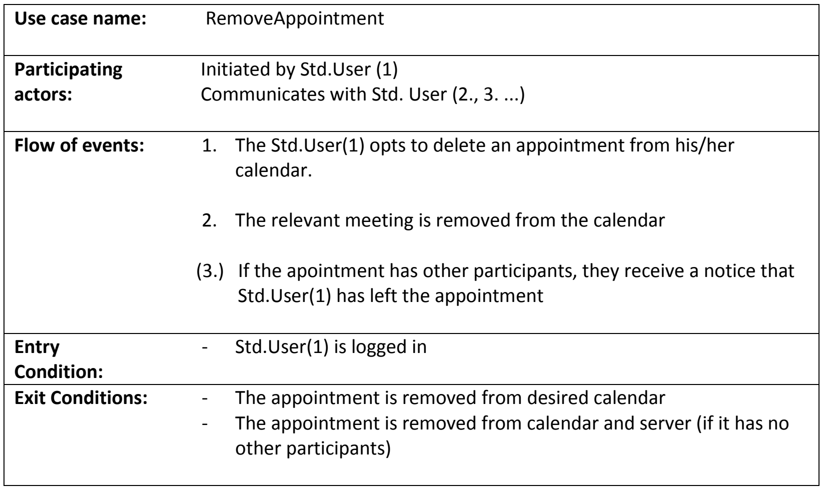
Scenario name:	CreateHelleAsUser
Participating actors:	Helle (Std. User)
Flow of events:	1. Helle is a new employee at Statsministeriet. She needs a new calendar and accesses her calendar client 2. Helle selects Create User 3. Helle enters name and password, and confirms 4. Helle receives a notice of successful user creation.

2.2.2 Use case model


CreateUser
initiate
CreateAppointment
Std. User
participate
DeleteAccount
ChangeTimePlaceOf
Appointment
RemoveAppointment

Use case name:	ChangeTime(And/OrPlace)OfAppointment
Participating actors:	Initiated by Std.User (1) Communicates with Std. User (2., 3. ...)
Flow of events:	1. The Std.User(1) opts to change the time and/or place of appointment in his/her calendar. 2. The relevant meeting is moved to the new time and place in the calender (3.) If the apointment has other participants, they receive a notice that Std.User(1) has changed the meeting
Entry Condition:	- Std.User(1) is logged in
Exit Conditions:	- The appointment is moved in all calenders

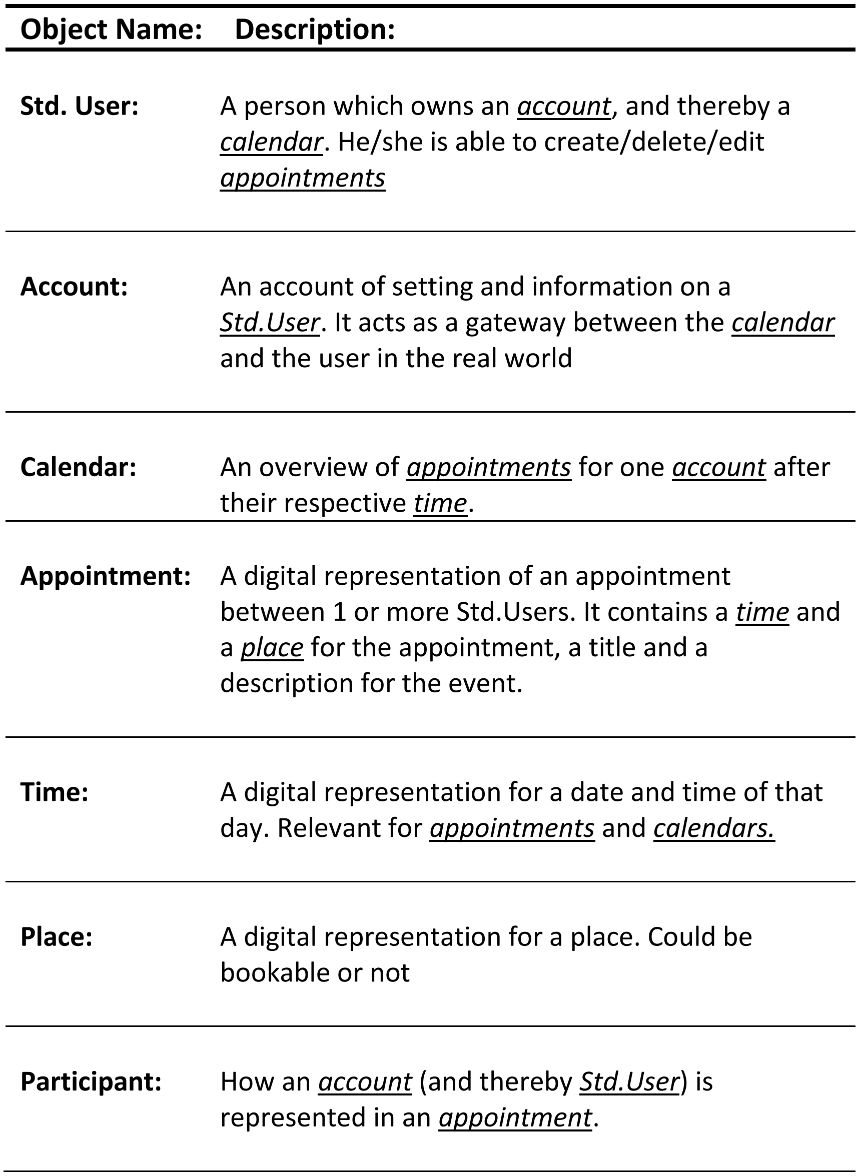
Use case name:	DeleteAccount
Participating actors:	Initiated by Std.User (1) Communicates with Std. User (2., 3. ...)
Flow of events:	1. The Std.User(1) opts to leave the calendarsystem, and deletes his/her account 2. All Std.User(1)’s appointments is removed (3.) If one of the appointments has other participants, they receive a notice that Std.User(1) has left the meeting
Entry Condition:	- Std.User(1) is logged in
Exit Conditions:	- The appointment is moved in all calenders



Use case name:	RemoveAppointment
Participating actors:	Initiated by Std.User (1) Communicates with Std. User (2., 3. ...)
Flow of events:	1. The Std.User(1) opts to delete an appointment from his/her calendar. 2. The relevant meeting is removed from the calendar (3.) If the apointment has other participants, they receive a notice that Std.User(1) has left the appointment
Entry Condition:	- Std.User(1) is logged in
Exit Conditions:	- The appointment is removed from desired calendar - The appointment is removed from calendar and server (if it has no other participants)

3 Glossary

3.1 Initial Analysis Objects:



Object Name:	Description:
Std. User:	A person which owns an *account*, and thereby a *calendar*. He/she is able to create/delete/edit *appointments*
Account:	An account of setting and information on a *Std.User*. It acts as a gateway between the *calendar* and the user in the real world
Calendar:	An overview of *appointments* for one *account* after their respective *time*.
Appointment:	A digital representation of an appointment between 1 or more Std.Users. It contains a *time* and a *place* for the appointment, a title and a description for the event.
Time:	A digital representation for a date and time of that day. Relevant for *appointments* and *calendars.*
Place:	A digital representation for a place. Could be bookable or not
Participant:	How an *account* (and thereby *Std.User*) is represented in an *appointment*.