

Das große Ganze

Carina Dolacek

Florian Dienesch

Elias Frantar

Jakob Saxinger

5.5.14

Angabe

Es soll ein kollaboratives Terminvereinbarungssystem (ähnlich wie Doodle) erstellt werden, in dem sich Benutzer koordinieren können. Folgende Anforderungen sollen dabei erfüllt werden:

Benutzer

- Neue Benutzer können sich registrieren
- Existierende Benutzer können sich einloggen
- Man kann nach registrierten Benutzern im System suchen (über ihren Namen).
- Ein Benutzer kann gleichzeitig ein Organisator und Teilnehmer sein.
- Jeder Benutzer kann sich die Events, die er organisiert, oder Events, an denen er teilnimmt, anzeigen lassen.

Organisator

- ist ein Benutzer, der Events mit Namen und mehreren Termin- und Zeitvorschlägen erstellt und die Einladungen an einige registrierte Benutzer schickt
- darf den Namen, die Termine und Zeiten eines Events ändern, aber nur bevor sich einer der Benutzer zu dem Event angemeldet hat
- darf neue Benutzer zu seinen Events zusätzlich einladen
- darf eingeladene Benutzer wieder löschen, bevor sich diese zu dem Event angemeldet haben
- darf die Events jederzeit löschen
- darf zu seinen Events Kommentare posten
- darf Kommentare zu seinen Events löschen (auch die von anderen Benutzern)
- Nachdem sich alle Benutzer zu einer Einladung angemeldet haben, darf der Organisator einen fixen Termin festlegen.

Teilnehmer

- wählt aus den vorgeschlagenen Terminen und Zeiten eines Events (eine Checkbox pro Zeitvorschlag reicht)
- darf seine Wahl ändern, bis ein fixer Termin existiert
- darf Kommentare zu Events, zu denen er eingeladen ist, posten

Notifications

- Ein Teilnehmer wird über jede neue/editierte/gelöschte Eventeinladung notifiziert.
- Weiters wird ein Teilnehmer notifiziert, sobald ein fixer Termin für ein Event festgelegt wird.
- Ein Organisator wird notifiziert, sobald alle Teilnehmer einen Termin gewählt haben.
- Wenn ein Benutzer zur Zeit einer Notification offline ist, darf diese nicht verloren gehen. Der Benutzer bekommt alle seine versäumten Notifications, sobald er online kommt.

Events

Es kann zwei Arten von Events geben

1. Events, bei denen sich die Teilnehmer auf (möglichst) einen Termin einigen sollen (Standardfall). Der Organisator legt letztendlich einen fixen Termin fest.
2. Events, bei denen jeweils nur ein Teilnehmer pro Termin erlaubt ist (z.B. für Elternsprechtage). Der Organisator muss jede Teilnehmer/Termin-Kombination fixieren.

Aufgabenstellung

Entwickeln Sie ein GUI-Programm, welches das Terminvereinbarungssystem realisiert. Bei der Abgabe müssen Sie die Aufgabe auf mindestens drei Rechnern (mit mehreren gleichzeitig gestarteten Clients) präsentieren.

Beim Starten des Programms müssen der gewünschte Benutzername und die Netzwerkadresse des Servers angegeben werden (kein Passwort erforderlich). Die Registrierung kann automatisch bei der ersten Anmeldung erfolgen.

Achten Sie bei der Implementierung auf die transaktionale Sicherheit. Überlegen Sie sich Situationen, in denen z.B. ein Benutzer versucht, eine Terminwahl zu einem in der Zwischenzeit gelöschten Event zu realisieren. Ihr Programm sollte auf solche und ähnliche Situationen entsprechend reagieren.

Beachten Sie bei der Implementierung, dass die Kommentare in derselben Reihenfolge aufgelistet werden müssen, in der diese von den einzelnen Benutzern abgeschickt wurden. Sie müssen sich auch Gedanken über die Persistenz der Informationen machen. Wenn die Serverinstanz herunterfährt, muss der gesamte Inhalt dauerhaft abgelegt worden sein.

Es reicht ein einfaches, aber funktionales GUI. Sie dürfen (sollten!) dafür Frameworks einsetzen.

Vorgehensweise

Es sind die Meta-Regeln zu beachten. Dabei ist zu beachten, dass nur durch eine obligatorische Design-Review durch die unterrichtenden Lehrkräfte, das gewählte Design (realisiert und vorgestellt mittels UML Diagrammen) verwendet werden darf. Nachträgliche Änderungen müssen durch einen Change-Request genehmigt werden. Diese müssen in eine Feature/Requirements Liste münden, die z.B. durch User-Stories definiert werden können. Zu bedenken sind auch nicht-funktionale Anforderungen an das System, wie z.B. die Anzeigegeschwindigkeit der ersten Termine und Kommentare.

Des weiteren sind Programmier-Teams verpflichtend. Diese sind durch eine/n Tester/in und eine/n Programmierer/in definiert. Angenommene Tasks der einzelnen Stories werden gleichzeitig(!) vom Tester und Programmierer behandelt, wobei der Tester die Anforderungen in z.B. Unit-Tests und der Programmierer in den entsprechenden Codeteilen implementiert. Dabei soll sichergestellt sein, dass sofort geeignete Testfälle den gerade eben implementierten Code auf dessen Funktionstüchtigkeit überprüfen.

Integrations- und Systemtests sind verpflichtend. Dabei sind in diesem Fall auch automatisierte GUI-Tests zu verwirklichen. Der Testbericht im Protokoll muss auch eine kontinuierliche Verbesserung der zu erzielenden Storypoints ersichtlich machen.

Termine:

8.5.2014 AdS: Design Review - Deadline für Design der Applikation!

4.6.2014 23:55: Deadline Abgabe (fix - keine Verlängerung möglich!)

5.6.2014: Abnahme Interview (15min / Gruppe)

Dazwischen müssen die Teamleader wöchentlich einen Reviewtermin selbstständig wahrnehmen.

Bewertungskriterien:

Dokumentation:

Design:

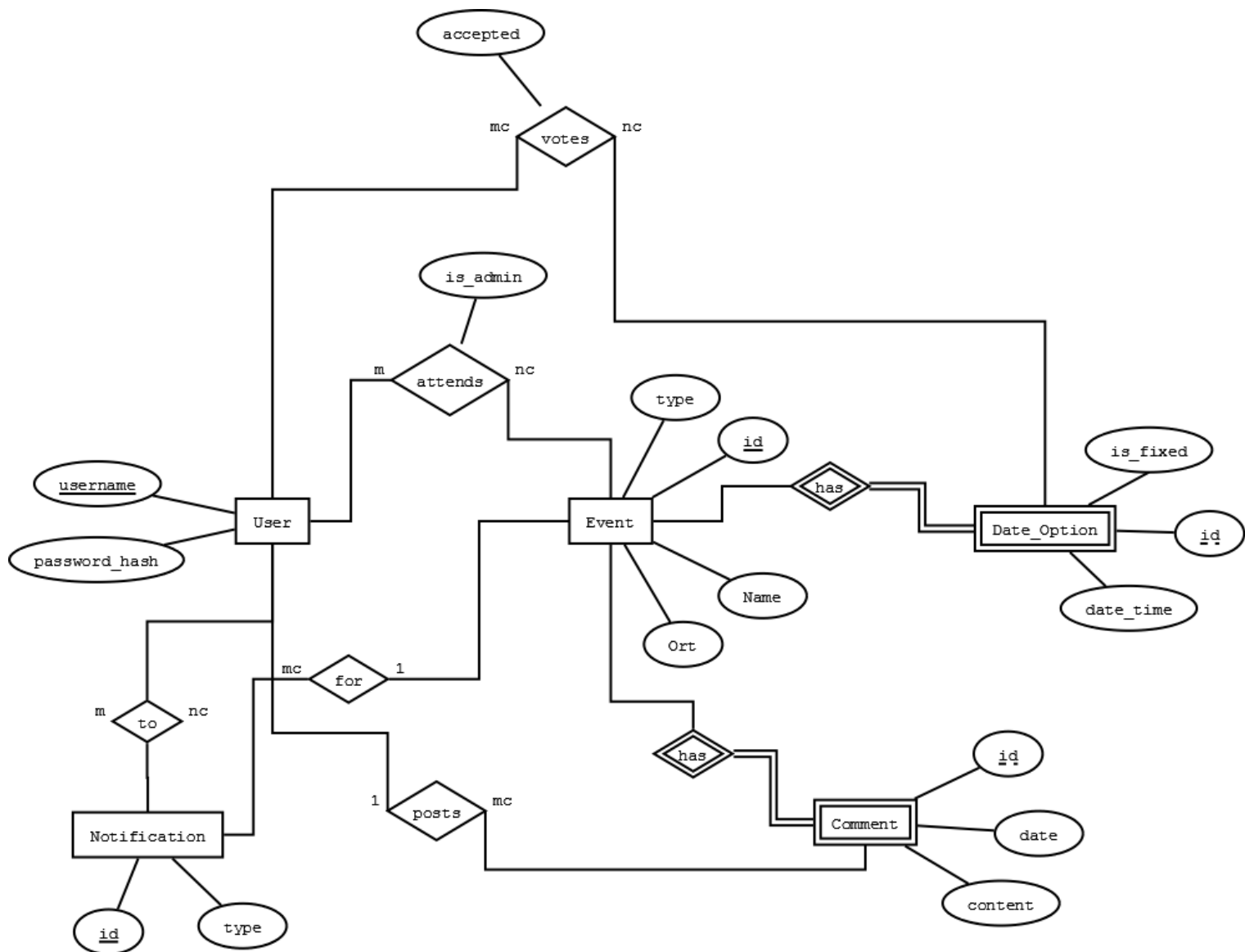
Implementierung:

Funktionalität:

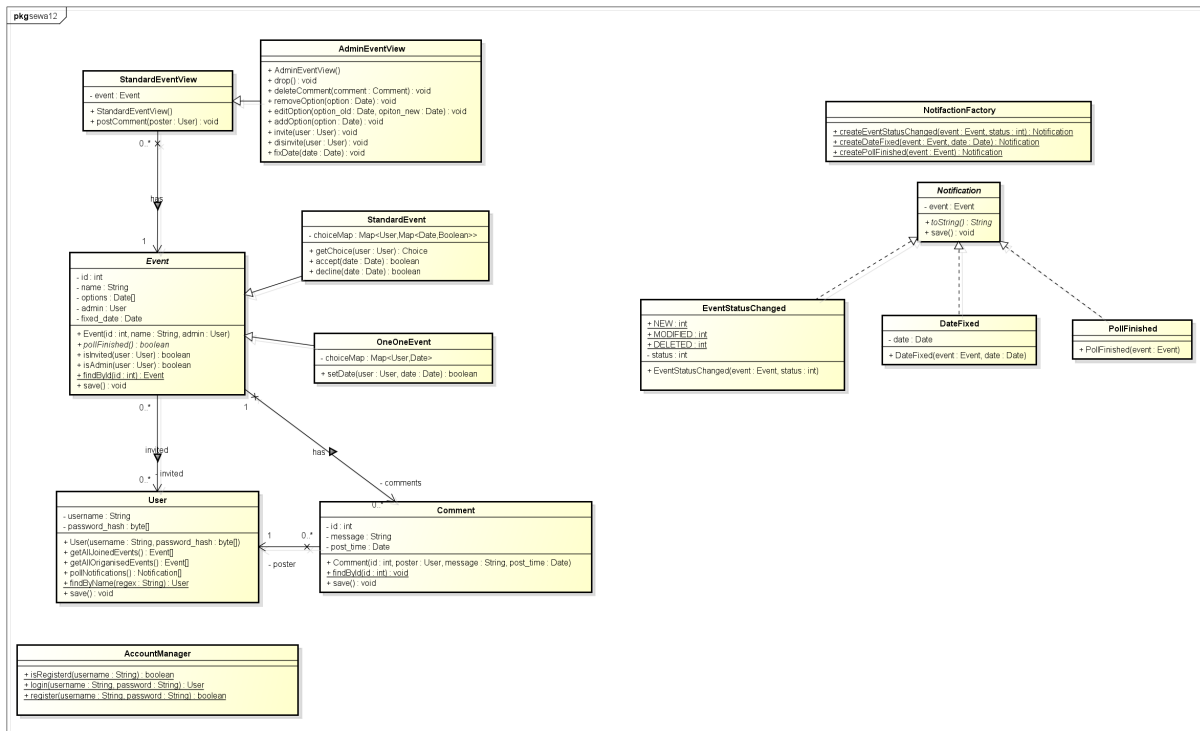
Tests/System:

Design

ER-Diagramm



UML- Diagramm



Grafik- Design

Eventübersicht

Meine Events

Neuer Termin

DAY MON 00, 00:00 AM

I'M AN EVENT 1

DAY MON 00, 00:00 AM

I'M AN EVENT 2

DAY MON 00, 00:00 AM

I'M AN EVENT 3

Events an denen ich teilnehme

DAY MON 00, 00:00 AM

I'M AN EVENT 1

DAY MON 00, 00:00 AM

I'M AN EVENT 2

DAY MON 00, 00:00 AM

I'M AN EVENT 3

Daten der Veranstaltung

Veranstaltung erstellen

Termin

Zeit

Name

Email

Veranstaltung

Nachricht

Termin auswählen

Termin

Veranstaltung erstellen

Termin

Zeit

Today ◀ May 2014 ▶						
SUN	MON	TUE	WED	THU	FRI	SAT
27	28	29 21:00 Te	30	1 21:00 Te	2	3 21:00 Test ev
4	5	6 21:00 Te	7	8 21:00 Te	9	10 21:00 Test ev
11	12	13 21:00 Te	14	15 21:00 Te	16	17 21:00 Test ev
18	19	20 21:00 Te	21	22 21:00 Te	23	24 21:00 Test ev
25	26	27 21:00 Te	28	29 21:00 Te	30	31 21:00 Test ev
1	2	3 21:00 Te	4	5 21:00 Te	6	7

Zeiten für den Termin

Veranstaltung erstellen Termin **Zeit**

	Zeit	Zeit	Zeit	Zeit
Termin				
Day MON 00, 00:00 AM				
Termin				
DAY MON 00, 00:00 AM				
Termin				
DAY MON 00, 00:00 AM				

Fertig