

Entwicklungsprojekt
Interaktive Systeme
Implementationsdokumentation

vorgelegt an der

Technische Hochschule Köln – Fakultät für
Informatik und Ingenieurwissenschaften

ausgearbeitet von

Leonid Vilents

betreut durch

Prof. Dr. Kristian Fischer

Prof. Dr. Gerhard Hartmann

B. Sc. Sheree Saßmannshausen

B. Sc. Daniela Reschke

Inhaltsverzeichnis

Installationsdokumentation	3
Voraussetzungen Serverlogik	3
Voraussetzungen Client.....	3
Installation und Inbetriebnahme des Servers	4
Installation und Inbetriebnahme des Clients	4
Abweichungen von der Dokumentation	5
Abweichende Alleinstellungsmerkmale	5
Abweichende Selbstständige Aktivitäten des Systems	6
Abweichungen von Strukturen und Architektur	6

Installationsdokumentation

Voraussetzungen Serverlogik

Für das Betreiben des Systems wird ein Computersystem benötigt, auf dessen Betriebssystem das JavaScript-Framework Node.js lauffähig ist. Die offizielle Webseite von NodeJS bietet Installationspakete für folgende Betriebssysteme nativ:

- Microsoft Windows (32bit, 64bit) als Installer (.msi) oder Binary (.zip)
- Apple macOS (64bit) als Installer (.msi) oder Binary (.tar.gz)
- Linux-Distributionen (32bit, 64bit) als Binary oder ARM (6, 7, 8)
- SunOS (32bit, 64bit) als Binary
- Docker-Image
- AIX (64bit)

Ferner wird auch der Sourcecode als .tar.gz zum selber Kompilieren angeboten.

Zusätzlich zu Node.js, wird ein MySQL-Server benötigt. Neben der Enterprise-Edition für Unternehmen, bietet MySQL auch unter der GPL-Lizenz einen Community-Server an.

Voraussetzungen Client

Für das Betreiben der Client-Anwendung für Endbenutzer, wird die Version 8 des Java Runtime Environments benötigt. Java wird von folgenden Betriebssystemen unterstützt:

- Microsoft Windows ab Vista SP2
- Microsoft Server 2009 R2 SP1 (64bit)
- Microsoft Server 2012, 2012 R2 (64bit)
- Apple macOSX ab Version 10.7.3
- Linux / RPM-Linux (32bit, 64bit)
- Solaris (32bit, 64bit)

Installation und Inbetriebnahme des Servers

Für die Installation des Servers wird davon ausgegangen, dass MySQL und Node.js bereits auf dem Zielsystem installiert sind. Es folgen nun die Installationsanweisungen für Windows und für unixoide Systeme.

1. Laden Sie den Server-Teil der Repository herunter (<https://github.com/scarcloud/EISSS17Vilents/tree/master/MS3/Implementation/Server>).
2. Die Schemadatei für die Datenbank befindet sich unter „data“ als „schema.sql“. Starten Sie ihren MySQL-Server und führen Sie diese Datei in dessen Benutzer-Interface (Kommandozeile oder Dritt-Programm) aus. Falls Sie an der Schemadatei Änderungen vornehmen möchten (z.B. Benutzernamen ändern), ändern Sie auch die Zugangsdaten in der „dbam.json“-Konfigurationsdatei (zu finden unter „util/config“).
3. Falls Sie das System mit Testdaten befüllen wollen, existiert unter <https://github.com/scarcloud/EISSS17Vilents/tree/master/MS3/Implementation> eine Rubrik „Daten“, unter der man eine Fixtures-Datei findet.
4. Öffnen Sie die Kommandozeile im Ordner mit der Hauptdatei „server.js“, und führen Sie diese mit Node.js („node server.js“) aus. Der Server wird jetzt versuchen, sich mit dem MySQL-Server zu verbinden. Bei Erfolg akzeptiert der Server jetzt Http-Requests

Installation und Inbetriebnahme des Clients

Für die Installation des Clients wird davon ausgegangen, dass Java bereits auf dem Zielsystem installiert ist.

1. Laden Sie unter <https://github.com/scarcloud/EISSS17Vilents/blob/master/MS3/Implementation/Client/client.zip> den kompilierten Client als ZIP-Datei herunter.
2. Entpacken Sie den ZIP-Ordner und führen sie client.jar aus.
3. Der Client wird jetzt laden, und Sie können sich entweder einloggen oder registrieren.

Abweichungen von der Dokumentation

Abweichende Alleinstellungsmerkmale

Von den in der Dokumentation angeführten Alleinstellungsmerkmalen, wurde das Merkmal „Erstellungstool für Case Modding-Teams für Sponsoren“ in seiner Funktionalität abgeändert. Um einen komplexeren Aspekt der Auswahl als den bloßen Durchschnitt der Gesamtreputation des theoretischen Teams zu betrachten, wurde die Überlegung getroffen, eine Art sekundäres Ranking in drei verschiedenen Bereichen zu berechnen:

- Kommentare: Casemodder mit einem hohen Kommentare-Ranking gelten als profiliert in sozialer Interaktion. Ihre Stärke soll vor allem in einer möglichen Kommunikation sein, was sich dadurch auszeichnet, dass sie viel kommentieren und ihre Kommentare viel Reputation generieren.
- Projekte: Ein hohes Projekte-Ranking steht für viele Projekte, die ein Casemodder in seiner Zeit als Systembenutzer begonnen hat. Die zusätzlichen Reputationspunkte aus seinen Projekten machen ihn zu einem erfahrenen Macher.
- Projekt-Update-Kombinationen: Dieses Ranking soll einen Projektmanagement-Aspekt beinhalten – ein Casemodder steigert sein P-U-Kombinationsranking dank konsistenter Dokumentation und Pflege seiner Projekte mittels Projektupdates.

Die Rankings berechnen sich aus folgenden Formeln:

$$R_{\text{kommentare}} = \frac{(n_{\text{kommentare}} + n_{\text{kommentarUpvotes}})}{2}$$

$$R_{\text{projekte}} = \frac{(n_{\text{projekte}} + n_{\text{projektUpvotes}})}{2}$$

$$R_{\text{projektUpdateKombination}} = \frac{\left(R_{\text{projekte}} + \left(\frac{(n_{\text{projektUpdates}} + n_{\text{projektUpdatesUpvotes}})}{2} \right) \right)}{2}$$

Ein vorgeschlagenes Casemodder-Team soll auf diese Weise auf seinen 3 Casemodder-Positionen alle 3 „Schwerpunkte“ bedienen können.

Das derzeitig implementierte System verfährt folgendermaßen bei der Generation der Teamvorschläge:

1. Der Client erhält bei Betreten der Sponsoringwelt vom Server einen Datensatz mit aktuellen Casemoddern mit Suchstatus. Jeder Casemodder-Datensatz enthält neben den anzuzeigenden Daten des Casemodders, eine Liste mit gezählten Elementen, zu denen der Casemodder in der Datenbank eine Referenz besitzt – also Projekte, Projektupdates, Kommentare und Upvotes für die jeweiligen Elemente.
2. Die Client-Komponente „CandidatePicker“ kloniert die Liste und sortiert diese mittels QuickSort-Algorithmus aufsteigend nach Gesamtreputation der Casemodder.
3. Von der sortierten Liste wird rekursiv das obere Quartil der Datensätze entnommen, bis sich die Zahl der Datensätze zwischen 6 und 17 befindet (willkürlich genommene Werte).
4. Für alle Casemodder-Objekte der gekürzten Liste, werden die Rankings aus den übergebenen Zählungen nach obigen Formeln berechnet.
5. Der CandidatePicker kloniert die gekürzte Liste dreimal, und sortiert das erste Klon-Array mittels BubbleSort aufsteigend nach Projekt-Update-Kombinationsranking.
6. Aus dem Array mit den sortierten Projekt-Update-Kombinationsrankings wird der Casemodder mit dem letzten Index gezogen.

7. Im Array mit den sortierten Projekte-Rankings wird zunächst der Top-Kandidat aus dem vorigen Array entfernt, bevor auch dort BubbleSort sortiert wird, diesmal aber aufsteigend nach Projekte-Rankings.
8. Die Schritte 6 und 7 werden nochmals für das dritte Array und das Kommentar-Ranking wiederholt.
9. Die drei ausgewählten Kandidaten werden als gesondertes Array zurückgegeben, und werden von der Controllerklasse in die Rubrik gerendert.

Abweichende Selbstständige Aktivitäten des Systems

Die nachfolgenden Abweichungen betreffen die in der Dokumentation erstellten Anforderungen an das System.

Die Anforderung /A0003/ wurde nicht umgesetzt. Grund dafür ist der Zeitpunkt der Teamvorschlag - zusammenstellung, bei der es zu Fehlern kommen kann. Die Ausweichmöglichkeit besteht darin, den Tab neu zu öffnen, da das System beim Wechseln der Tabs für den selektierten Tab automatisch die aktuellsten Informationen abrufen.

Einige der Anforderungen konnten zum Zeitpunkt der Deadline der Implementation nicht umgesetzt werden:

Abweichungen von Strukturen und Architektur

Das aus der Dokumentation bekannte ER-Diagramm wurde um einige Elemente erweitert.

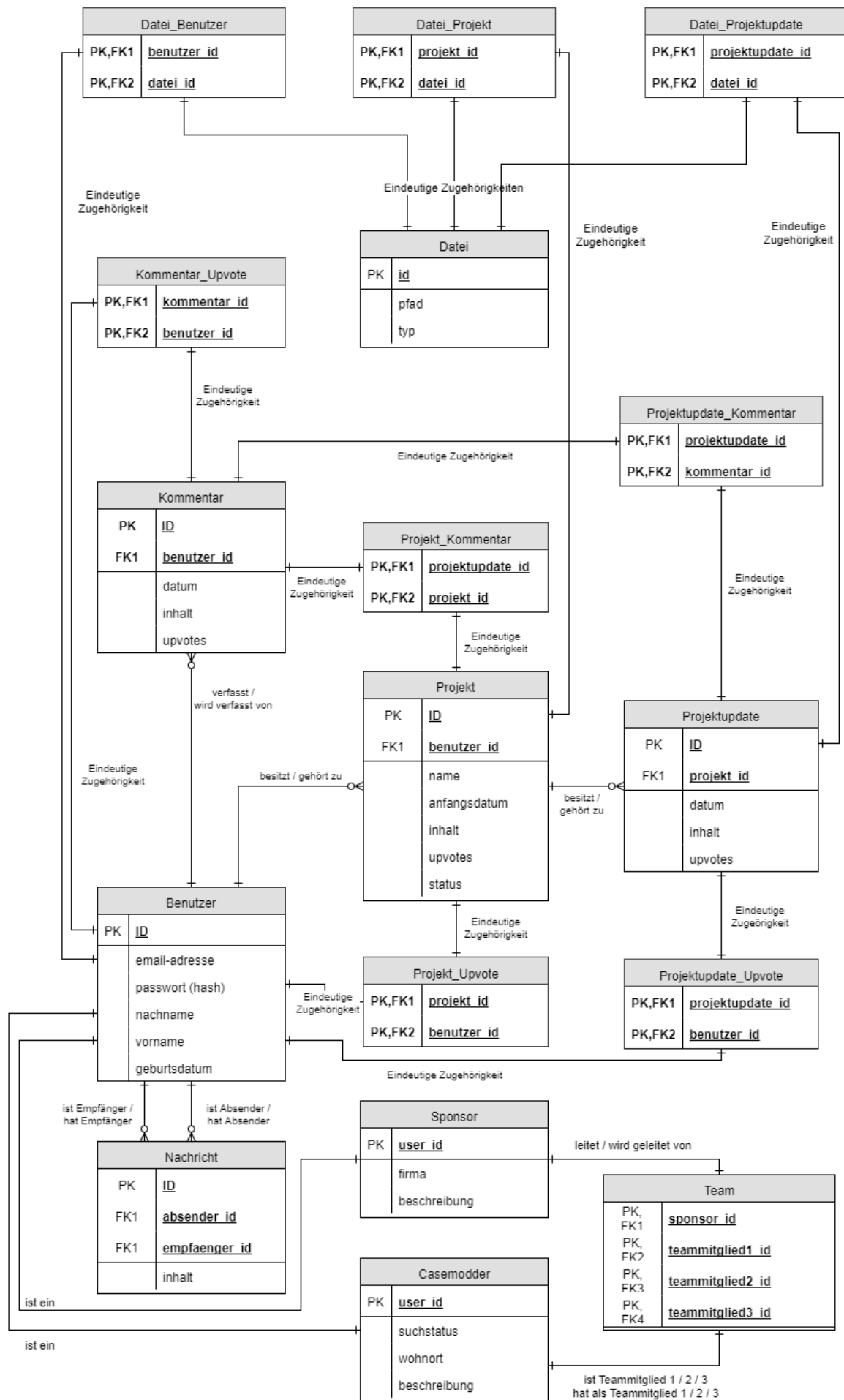


Abbildung 1: Aktualisiertes ER-Diagramm

Die Benutzer-Tabelle speichert jetzt redundante Informationen, während die Casemodder- und Sponsor-Tabellen rollenspezifische Informationen mit einer Fremdschlüsselreferenz speichern. Durch letztere wird die Verwendung eines künstlichen Primärschlüssels für die Benutzerrollen obsolet, da jeder Benutzer nur einen Primärschlüssel besitzen kann.