

Projekt Pinnwand

App-Entwicklung mit Datenbankzugriff unter Android

Hung Nguyen Minh, Anh Nguyen Hoang, Dennis Okos, Eduard Rutz, Daniel Ruppel

09.07.2015

Inhalt

Einleitung.....	3
Idee.....	3
Informelle Beschreibung	4
Konzeptioneller Datenbankentwurf.....	4
Relationaler Datenbankentwurf.....	6
Implementierung in SQL.....	7
Layout Entwurf	8
State Chart.....	9
Implementierung in Java	10
Fazit	10

Einleitung

Ziel dieses Projektes ist die Erstellung einer Android Applikation. Dabei soll die Eclipse Programmieroberfläche zur Realisierung verwendet werden.

Mithilfe der Applikation können in einer Datenbank Dateninhalte erstellt, bearbeitet oder ausgelesen werden.

Idee

Die Idee ist es, eine Android Applikation zu entwickeln, die es einem ermöglicht, sich als Nutzer innerhalb eines Forums mit anderen Nutzern über Problemfragen zu unterhalten und seine Beiträge zu teilen.

Informelle Beschreibung

Der erste Schritt beim Entwerfen einer Datenbank ist eine informelle Problembeschreibung. Diese enthält Bedingungen, welche vom Auftraggeber gestellt werden und die beim Entwerfen einzuhalten sind. In unserem Fall wird die informelle Problembeschreibung von uns selbst verfasst.

Die App wird mit einem Datenbankzugriff zu einer anliegenden Datenbank implementiert. Auf der Pinnwand gibt es einzelne Themen(Threads) mit den dazugehörigen Kommentaren(Notizen).

Für die Datenbank gilt folgende Beschreibung:

- Jede Pinnwand (App Oberfläche) hat N Themen
- Jeder Thread hat einen Namen, einen Verfasser, eine Beschreibung, einen Zeitpunkt und ein Löschdatum(30 Tage nach letztem Kommentar)
- Jeder Thread hat N Notizen
- Jede Notiz hat einen Verfasser, ein Kommentar und einen Zeitpunkt
- Jeder User hat einen Username, ein Passwort, eine E-Mail, einen Vornamen, einen Nachnamen, ein Geburtsdatum und einen Wohnort

Konzeptioneller Datenbankentwurf

Ein gutes Programm entsteht nicht durch einfaches Programmieren nach Erhalt der Aufgabenstellung. Vor der Formulierung des Quelltextes sollte zuerst ein Plan bzw. System entstehen, wonach man sich später beim Programmieren bezieht. Der Entwurf einer Datenbank ist hierbei keine Ausnahme.

Aus diesem Grund erstellen wir zuerst einen Konzeptionellen Datenbankentwurf, um alle Entitäten und Relationen untereinander darzustellen.

Dazu nutzen wir die Notation eines Domänenmodelles, welche uns im Kurs „Software Engineering“ beigebracht wurde.

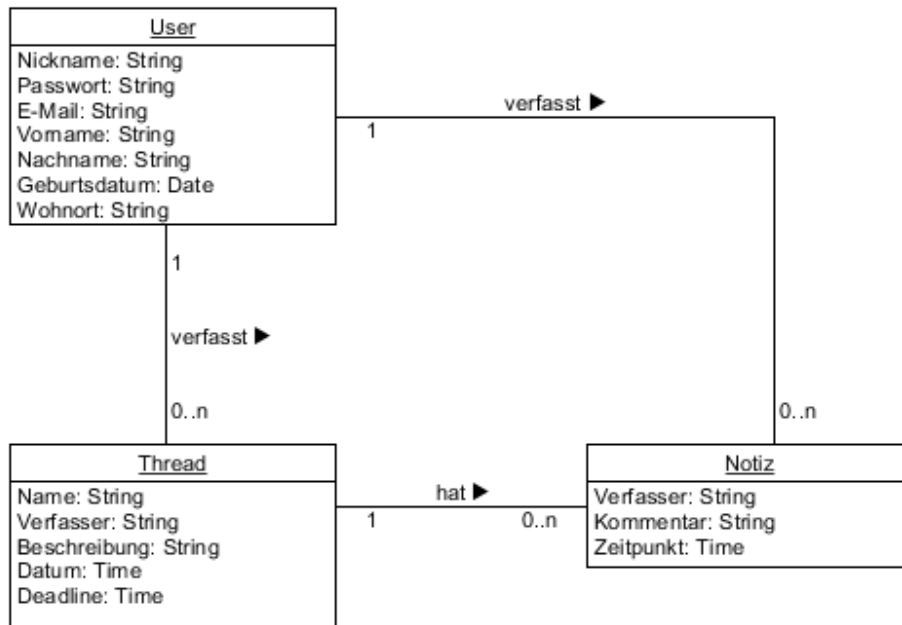
Normalerweise wird für einen konzeptionellen Datenbankentwurf ein ER-Diagramm benutzt, da das verwendete Domänenmodell jedoch eine ähnliche Ausdruckskraft besitzt und bereits bekannt ist, greifen wir auf dieses bereits aufgebaute Wissen zurück.

Beim Entwerfen des Konzeptes achten wir stets auf die vorgegebene informelle Problembeschreibung, die uns gegeben ist, wobei diese in unserem Fall von uns selbst definiert wurde.

Innerhalb unserer Pinnwand werden drei Tabellen benutzt: Der User, der Thread und die Notiz.

Ein Thread ist durch seinen Namen, seinen Verfasser, seine Beschreibung, sein Datum und seine Deadline definiert, d.h. es werden drei Datentypen vom String und zwei Datentypen abhängig von der Zeit verwendet. Die Notiz beinhaltet den Verfasser, das Kommentar und den Zeitpunkt, an dem es verfasst wurde. Ein User wird durch einen Nicknamen, ein Passwort, seine E-Mail, seinen Vor- und Nachnamen, sowie Geburtsdatum und Wohnort definiert.

Ein Thread beinhaltet keine oder mehrere Kommentare, aber Kommentare sind genau einem Thread zugewiesen. Jeder Thread und jeder Kommentar sind jeweils genau einem User zugeordnet, wobei jeder User keine oder mehrere Kommentare und Threads verfassen kann.



Die Assoziationen formalisieren nur 1-zu-N Beziehungen:

- Jede Notiz wird von genau einem User verfasst
- Jeder Thread wird von genau einem User verfasst
- Jede Notiz gehört zu genau einem Thread

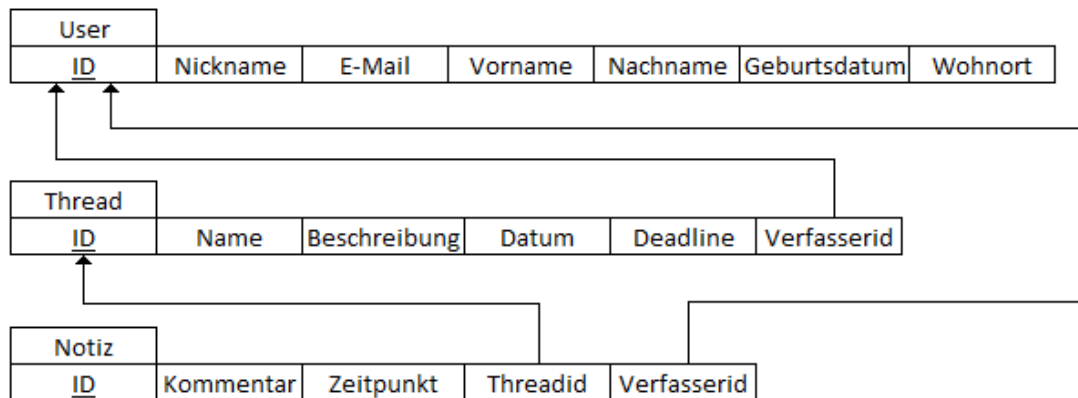
Relationaler Datenbankentwurf

Aus dem erarbeiteten konzeptionellen Datenbankentwurf wird als nächstes der relationale Datenbankentwurf abgeleitet.

Als fachlicher Schlüssel würden bei unserem Datenbankentwurf nur Strings oder Kombinationen von Strings in Frage kommen. Um dies zu verhindern wird bei jeder Tabelle eine ID als technischer Schlüssel hinzugefügt.

Die Notiz-Tabelle enthält eine Thread-ID, welche als Foreign Key auf die ID der zugehörigen Threads verweist. Außerdem werden innerhalb der Tabellen Thread und Notiz die Verfasser durch eine Verfasser-ID gekennzeichnet, welche dann jeweils auf die ID der Tabelle User verweisen.

Pinnwand



Implementierung in SQL

Durch den vorhandenen relationalen Datenbankentwurf fällt die Implementierung in SQL leichter, lehnt man sich an den Entwurf an.

```
create table benutzer(  
id int not null auto_increment,  
nickname varchar (100) not null,  
vorname varchar (100),  
nachname varchar (100),  
e_mail varchar (100)not null,  
zeitstempel TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP,  
primary key(id)  
);
```

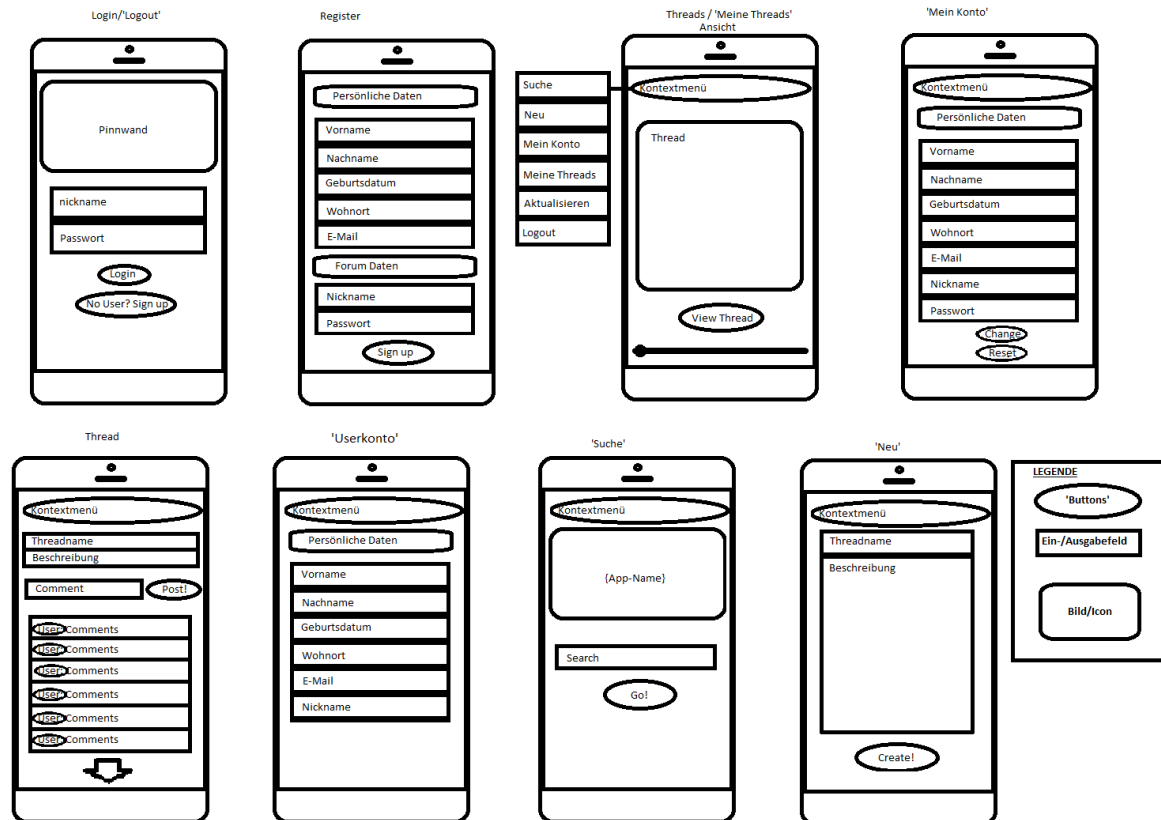
```
create table thread(  
id int not null auto_increment,  
b_id int not null,  
topic text not null,  
datum TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP,  
deadline date,  
beschreibung text,  
primary key (id),  
foreign key (b_id) references benutzer (id) on update cascade on delete cascade  
);
```

```
create table notiz(  
id int not null auto_increment,  
b_id int not null,  
t_id int not null,  
kommentar text,  
datum TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP,  
primary key(id),  
foreign key (b_id) references benutzer (id) on update cascade on delete cascade,  
foreign key (t_id) references thread (id) on update cascade on delete cascade  
);
```

Die erstellte Datenbank wird im SQL-Code nicht befüllt und es werden keine Queries beschrieben da diese Funktionen von der App übernommen werden sollen.

Layout Entwurf

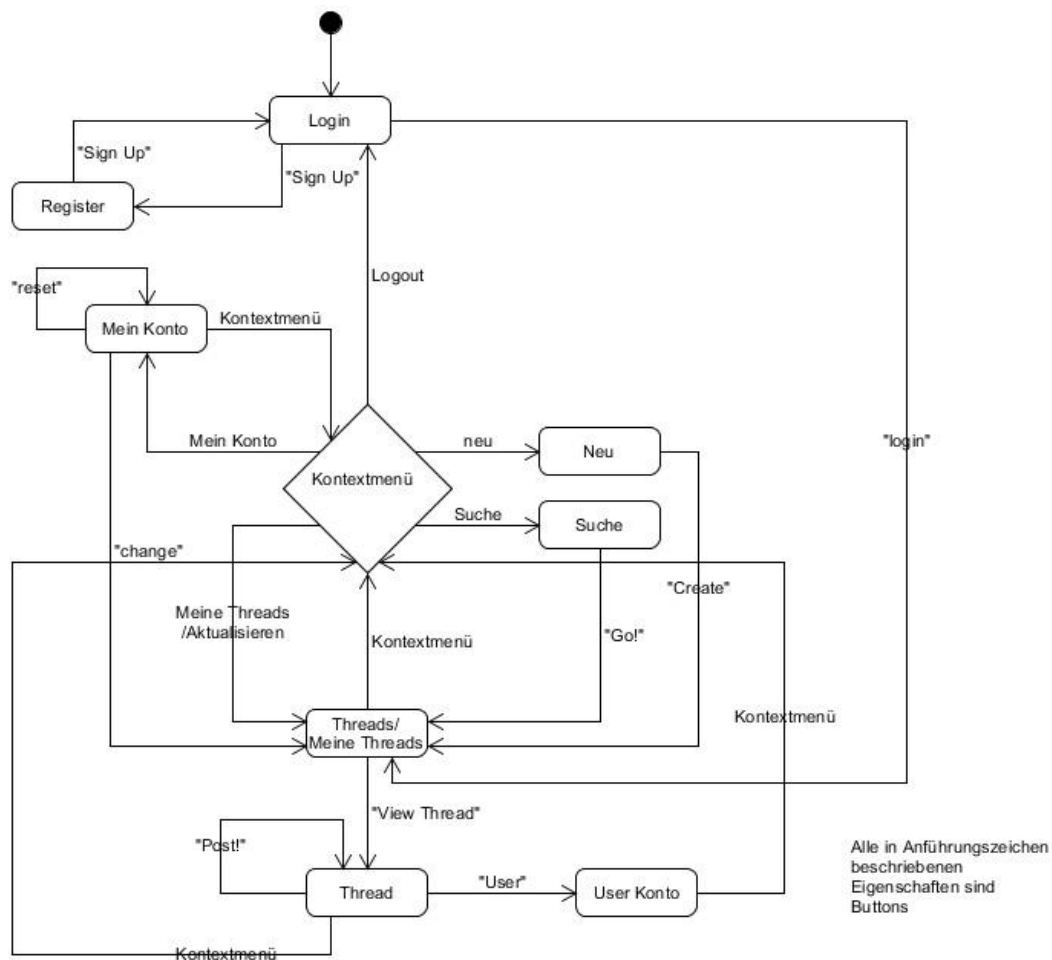
Der Entwurf des Layouts ist eine wichtige Aufgabe, die vor der Entwicklung des Quellcodes der Applikation erstellt werden muss. Das Layout zeigt alle möglichen Ausgabebildschirme. Beim Start kommt der Benutzer sofort zum Login-Fenster; ohne eine Anmeldung ist die Anwendung nicht zugänglich. Nach einer erfolgreichen Anmeldung können eigene, sowie öffentliche Attribute eines anderen Users eingesehen werden und Threads können erstellt, gesucht, geöffnet oder Kommentare geschrieben werden. Die Steuerung zwischen den Ausgabefenstern findet hauptsächlich durch das Kontextmenü statt. Da dieses jedoch nicht von allen Fenstern zugänglich ist und auch nicht zu jedem Fenster führt, werden auch Buttons benutzt um das ganze Spektrum des Layouts zu decken.



State Chart

Aus unserem Layout-Entwurf haben wir ein State Chart erstellt, um die Implementierung der App in Java zu vereinfachen und es in der Gruppe verständlicher zu machen. Ausserdem sind fehlerhafte Verbindungen im Layout einfacher zu Entdecken und zu bearbeiten.

Unser Startzustand ist das Login Fenster, da man bei jedem Programmstart sich zuerst einloggen muss. Die App hat keinen Endzustand, da diese ohne Eingriff von Android nie terminiert wird. Das Programm kann also nur durch Android Vorgänge abgebrochen werden.



Implementierung in Java

Während der Entwicklung des Java-Codes sind einige Probleme aufgetreten:

Nach dem Login war der App nicht bekannt, welcher User eingeloggt wurde. Vor allem beim Erstellen eines Kommentars war dies ein Problem. Um den einmalig eingeloggteten User zu speichern, wurde die globale Variable `CurrentUserID` in der Pinnwand Applikation deklariert, um den Benutzer zu speichern.

Um eine Registrierung mit selbem Username zu verhindern, wurde eine Methode `checkIfExist` eingefügt. Diese gibt den Wert `TRUE` zurück wenn bereits ein Nutzer mit dem selben Usernamen in der Datenbank vorhanden ist.

Damit der zuletzt erstellte Thread als erstes in der App angezeigt wird, wurde eine SQL-Abfrage mit dem Inhalt `ORDER BY DESC` hinzugefügt, um bei der Ansicht immer mit dem aktuellsten Thread zu beginnen.

Das größte Problem bei der Implementierung in Java war die Verbindung zu dem zentralen Datenbankserver. Um dieses Problem zu umgehen, wurde eine Lokale Datenbank durch die App implementiert und mit dieser weitergearbeitet. Da in dieser keine anderen Typen als Strings möglich waren, wurde für jede Spalte der Tabellen dieser Typ genutzt.

Fazit

Die Entwickelte Android-App ist definitiv erweiterbar. Mögliche Funktionen wären z.B.:

- die eigenen Threads zu löschen
- die eigenen Threads anzuzeigen
- die Verwendung einer zentralen Datenbank
- erweiterte Thread-Suche
- eine 'Passwort vergessen'-Funktion
- das eigene Kommentar zu löschen
- das Löschen von Threads ab einer vorgegebenen Zeit

Desweiteren wäre eine Kommunikation mit einer zentralen Datenbank eine Möglichkeit die App nicht nur auf ein Gerät zu beschränken.