Präsentation der Umfragewebseite

Modul DBA02

Daniel Falkner + Eugen Grinschuk

AKAD Pinneberg + Stuttgart

3+4.Oktober.2013

Daniel Falkner
Eugen Grinschuk

Über uns

Wer sind wir?

- AKAD Studenten - Bachelor of Science (Wirtschaftsinformatik)

Über uns
Daniel Falkner

Daniel Falkner

- T-Systems International GmbH - Telekom IT
- IT-Architekt - System Analyst
- Projektleiter
- Proof of Concept Engineer
- Debian Linux Administrator

Über uns
Eugen Grinschuk

Eugen Grinschuk

- T-Systems International GmbH
- IT-Architekt
- Projektleiter
- System Engineer

verwendete Werkzeuge

- > PHP 5.3
- MySQL 5
- Eclipse
- Git
- Dia [1]
- `http://www.dba02.studieren-und-arbeiten.de`

[1] `http://live.gnome.org/Dia`

Datenbank

UML Modell

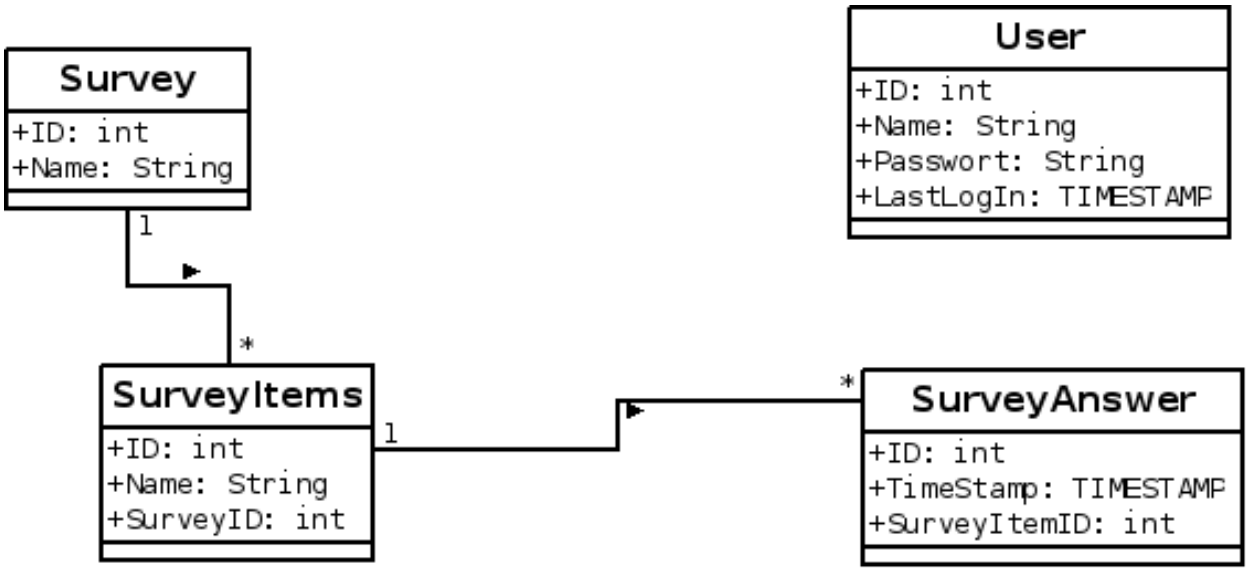

Abb.: UML Modell

- Flache Struktur
- Minimaler Aufbau für Umfragesystem mit Usern

Relationen Modell

- 3te Normalform
- Keine Redundanz

ID	Name
1	Wie findest du dieses Seite?

Table: Survey

Relationen Modell

ID	Name	SurveyID
1	Super	1
2	Fett	1

Table: SurveyItems

ID	TimeStamp	SurveyItemID
1	2013-08-29 19:53:55	2
2	2013-08-29 19:53:55	1

Table: SurveyAnswer

Relationen Modell

ID	Name	Passwort	LastLogIn
1	admin	test	0000-00-00 00:00:00

Table: User

- Gute Performance

SQL Data Definition Language

```
1 CREATE TABLE `Survey` (
2   `ID` int(11) NOT NULL AUTO_INCREMENT,
3   `Name` varchar(255) NOT NULL,
4  PRIMARY KEY (`ID`)
5 ) ENGINE=InnoDB AUTO_INCREMENT=0 DEFAULT CHARSET=latin1
      ;
```

Listing 1: database.sql

SQL Data Definition Language

```
1 CREATE TABLE `SurveyItems` (
2   `ID` int(11) NOT NULL AUTO_INCREMENT,
3   `Name` varchar(255) NOT NULL,
4   `SurveyID` int(11) NOT NULL,
5   PRIMARY KEY (`ID`),
6   KEY `SurveyID` (`SurveyID`),
7   CONSTRAINT `SurveyItems_ibfk_2` FOREIGN KEY (`SurveyID
        `) REFERENCES `Survey` (`ID`) ON DELETE CASCADE ON
      UPDATE CASCADE
8 ) ENGINE=InnoDB AUTO_INCREMENT=0 DEFAULT CHARSET=latin1
        ;
```

Listing 2: database.sql

SQL Data Definition Language

```
1 CREATE TABLE `SurveyAnswer` (
2   `ID` int(11) NOT NULL AUTO_INCREMENT,
3   `TimeStamp` timestamp NOT NULL DEFAULT
      CURRENT_TIMESTAMP,
4   `SurveyItemID` int(11) NOT NULL,
5   PRIMARY KEY (`ID`),
6   KEY `SurveyItemID` (`SurveyItemID`),
7   CONSTRAINT `SurveyAnswer_ibfk_2` FOREIGN KEY (`
      SurveyItemID`) REFERENCES `SurveyItems` (`ID`) ON
      DELETE CASCADE ON UPDATE CASCADE
8 ) ENGINE=InnoDB AUTO_INCREMENT=0 DEFAULT CHARSET=latin1
      ;
```

Listing 3: database.sql

SQL Data Definition Language

```
1 CREATE TABLE `User` (
2   `ID` int(11) NOT NULL AUTO_INCREMENT,
3   `Name` varchar(64) NOT NULL,
4   `Passwort` varchar(64) NOT NULL,
5   `LastLogIn` timestamp NOT NULL DEFAULT '0000-00-00
      00:00:00',
6   PRIMARY KEY (`ID`),
7   UNIQUE KEY `Name` (`Name`)
8 ) ENGINE=InnoDB AUTO_INCREMENT=0 DEFAULT CHARSET=latin1
      ;
```

Listing 4: database.sql

SQL Data Definition Language

```
1 INSERT INTO `User` (
2   `ID` ,
3   `Name` ,
4   `Passwort` ,
5   `LastLogIn`
6 ) VALUES (
7   NULL , 'admin', 'test', '0000-00-00 00:00:00'
8 );
```

Listing 5: database.sql

Programmcode

> PHP 5.3

- OOP [1]
- Namespaces
- Klassen Autoloader
- 1426 Zeilen
  - 525 Zeilen Code
  - 526 Zeilen Kommentare
  - 375 Leerzeilen ;-)

[1] Objektorientierte Programmierung

Design Pattern

- Singleton für Konfiguration
- MVC [1]

[1] Modell View Controller

Singleton und Konfigruation in INI Datei

```
database_type = mysql
database_port = 3306
database_name = dba02
database_host = localhost
database_user = dbuser
database_pass = supersicherundextremgeheim
database_verbose = 0
application_debugging = 0
```

Listing 6: config.ini

```
$conf = \Config::getInstance();
$debug = $conf->application_debugging;
```

Listing 7: Config Klasse

MVC

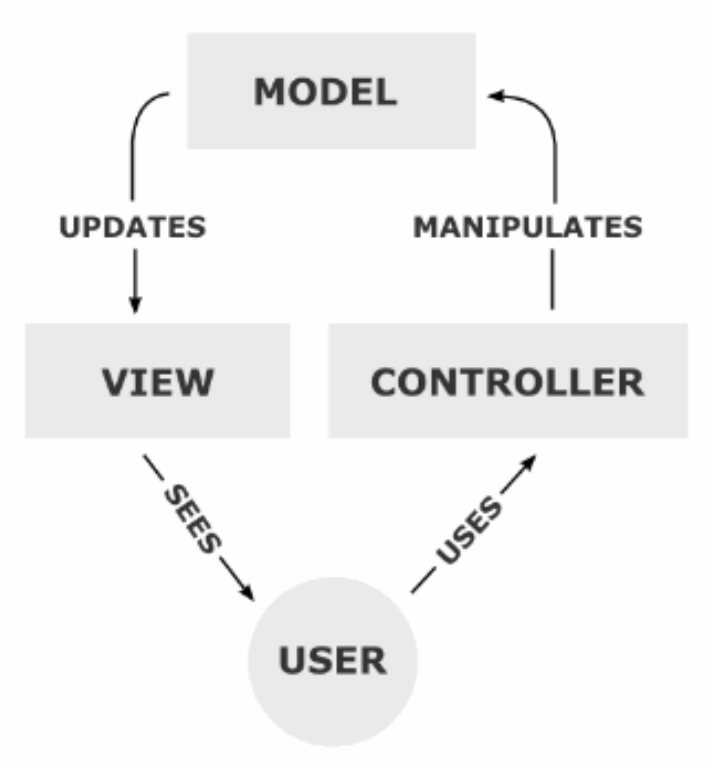

Abb.: Wikipedia MVC
http://en.wikipedia.org/wiki/File:MVC-Process.png

Die Steuerung

- Interface zwischen Model und View
- Saubere Codekapselung
- kein Spagetti Code und Springen zwischen Dateien

Präsentation

- eigene Template Engine
- CSS Twitter Bootstrap [1]

[1] http://getbootstrap.com/

```
1 $this->view->setTemplate('surveys');
2 $this->view->assign('surveys', $this->model->getSurveys
      ());
3 $this->view->display();
```

Listing 8: Template Engine

Modell

- PDO [1]
- Prepared Statements
- SQL-Injection Vorbeugung

[1] PHP Data Objects

SQL-Injection Beispiel

```php
$user = $_GET['user'];
$sql = "SELECT * FROM user WHERE name = '$user'";
```

Listing 9: PHP Code der nicht Existieren sollte!!

URL Aufruf

http://meineseite.de/index.php?user=owned'; DROP TABLE user;

Das Ausgeführte SQL Query

SELECT * FROM table WHERE key = 'owned'; DROP TABLE user;

Live Demo

Please wait ... we are connecting ;-)

Anhang

Vielen Dank für Ihre Aufmerksamkeit.
Fragen?

Links

- https://github.com/derdanu/akad-dba02-beamer
- http://www.dba02.studieren-und-arbeiten.de

Quellen

- http://www.w3.org/
- http://git-scm.com/
- http://de.selfhtml.org/
- http://php.net/
- http://openbook.galileocomputing.de/javascript_ajax/