

Präsentation der Umfragewebsite Modul DBA02

Daniel Falkner + Eugen Grinschuk

AKAD Pinneberg + Stuttgart

3+4.Okttober.2013



Über uns

Wer sind wir?

- AKAD Studenten - Bachelor of Science (Wirtschaftsinformatik)

- 1 Über uns
- 2 Werkzeuge
- 3 Datenbank
- 4 Programmcode
- 5 Live Demo
- 6 Anhang

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

Datenbank

- Über uns
- Werkzeuge
- Datenbank**
 - UML Modell
 - Relationen Modell
 - SQL DDL
- Programmcode
- Live Demo

verwendete Werkzeuge

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

¹<http://live.gnome.org/Dia>

UML Modell

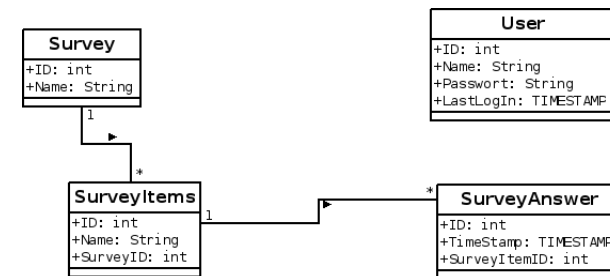


Abb.: UML Modell

- 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	Passwort	LastLogin
1	admin	test	0000-00-00 00:00:00

Table: User

- Gute Performance

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

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 '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 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 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

- 1 Über uns
- 2 Werkzeuge
- 3 Datenbank
- 4 Programmcode**
 - PHP
 - Design Pattern
 - Controller
 - View
 - Model
- 5 Live Demo
- 6 Anhang

Design Pattern

- Singleton für Konfiguration
- MVC ¹

¹Modell View Controller

> PHP 5.3

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

¹Objektorientierte Programmierung

Singleton und Konfiguration in INI Datei

```
1 database_type = mysql
2 database_port = 3306
3 database_name = dba02
4 database_host = localhost
5 database_user = dbuser
6 database_pass = supersicherundextremgeheim
7 database_verbose = 0
8 application_debugging = 0
```

Listing 6: config.ini

```
1 $conf = \Config::getInstance();
2 $debug = $conf->application_debugging;
```

Listing 7: Config Klasse

MVC

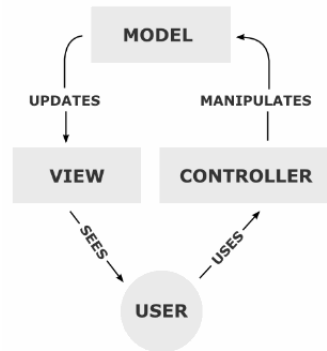


Abb.: Wikipedia MVC

<http://en.wikipedia.org/wiki/File:MVC-Process.png>

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

Die Steuerung

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

Modell

- PDO ¹
- Prepared Statements
- SQL-Injection Vorbeugung

¹PHP Data Objects

SQL-Injection Beispiel

```
1 $user = $_GET['user'];  
2 $sql = "SELECT * FROM user WHERE name = '$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;`

Anhang

Vielen Dank für Ihre Aufmerksamkeit.
Fragen?

Links

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

Live Demo

Please wait ... we are
connecting ;-)

Quellen

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