Über uns
Werkzeuge
Datenbank
Programmcode
Live Demo
Anhang

# Präsentation der Umfragewebseite
## Modul DBA02

Daniel Falkner + Eugen Grinschuk

AKAD Pinneberg + Stuttgart

3+4.Oktober.2013

AKAD

Über uns
Werkzeuge
Datenbank
Programmcode
Live Demo
Anhang

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

Daniel Falkner
Eugen Grinschuk

# Über uns
Eugen Grinschuk

## Eugen Grinschuk

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

Über uns
**Werkzeuge**
Datenbank
Programmcode
Live Demo
Anhang

## verwendete Werkzeuge

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

---

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

Über uns
Werkzeuge
**Datenbank**
Programmcode
Live Demo
Anhang

UML Modell
Relationen Modell
SQL DDL

# Datenbank

1. Über uns

2. Werkzeuge

3. Datenbank
   - UML Modell
   - Relationen Modell
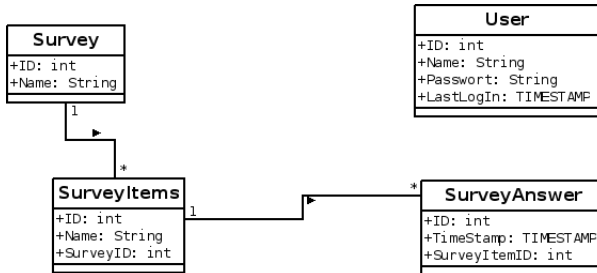   - SQL DDL

4. Programmcode

5. Live Demo

Über uns
Werkzeuge
**Datenbank**
Programmcode
Live Demo
Anhang

**UML Modell**
Relationen Modell
SQL DDL

# UML Modell



Abb.: UML Modell

- Flache Struktur
- Minimaler Aufbau für Umfragesystem mit Usern

Über uns
Werkzeuge
**Datenbank**
Programmcode
Live Demo
Anhang

UML Modell
Relationen Modell
SQL DDL

## Relationen Modell

- 3te Normalform

- Keine Redundanz

| ID | Name |
|----|------|
| 1 | Wie findest du dieses Seite? |

Table: Survey

Über uns
Werkzeuge
**Datenbank**
Programmcode
Live Demo
Anhang

UML Modell
Relationen Modell
SQL DDL

## 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

Über uns
Werkzeuge
**Datenbank**
Programmcode
Live Demo
Anhang

UML Modell
Relationen Modell
SQL DDL

## Relationen Modell

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

Table: User

- Gute Performance

Über uns
Werkzeuge
**Datenbank**
Programmcode
Live Demo
Anhang

UML Modell
Relationen Modell
**SQL DDL**

# SQL Data Definition Language

```sql
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

Über uns
Werkzeuge
**Datenbank**
Programmcode
Live Demo
Anhang

UML Modell
Relationen Modell
**SQL DDL**

# SQL Data Definition Language

```sql
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

Über uns
Werkzeuge
**Datenbank**
Programmcode
Live Demo
Anhang

UML Modell
Relationen Modell
**SQL DDL**

# SQL Data Definition Language

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

Listing 3: database.sql

Über uns
Werkzeuge
**Datenbank**
Programmcode
Live Demo
Anhang

UML Modell
Relationen Modell
**SQL DDL**

# SQL Data Definition Language

```sql
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

Über uns
Werkzeuge
**Datenbank**
Programmcode
Live Demo
Anhang

UML Modell
Relationen Modell
**SQL DDL**

# SQL Data Definition Language

```sql
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

Über uns
Werkzeuge
Datenbank
**Programmcode**
Live Demo
Anhang

PHP
Design Pattern
Controller
View
Model

# Programmcode

1 Über uns

2 Werkzeuge

3 Datenbank

4 Programmcode
   - PHP
   - Design Pattern
   - Controller
   - View
   - Model

5 Live Demo

6 Anhang

Über uns
Werkzeuge
Datenbank
**Programmcode**
Live Demo
Anhang

**PHP**
Design Pattern
Controller
View
Model

## > PHP 5.3

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

[1]Objektorientierte Programmierung

Über uns
Werkzeuge
Datenbank
**Programmcode**
Live Demo
Anhang

PHP
**Design Pattern**
Controller
View
Model

# Design Pattern

- Singleton für Konfiguration
- MVC [1]

  ---
  [1] Modell View Controller

Über uns
Werkzeuge
Datenbank
Programmcode
Live Demo
Anhang

PHP
Design Pattern
Controller
View
Model

# Singleton und Konfigruation 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;
```
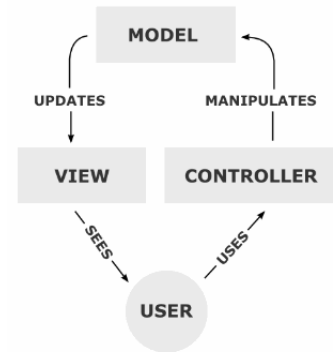
Listing 7: Config Klasse

Über uns
Werkzeuge
Datenbank
**Programmcode**
Live Demo
Anhang

PHP
**Design Pattern**
Controller
View
Model

# MVC



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

Über uns
Werkzeuge
Datenbank
**Programmcode**
Live Demo
Anhang

PHP
Design Pattern
**Controller**
View
Model

## Die Steuerung

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

Über uns
Werkzeuge
Datenbank
**Programmcode**
Live Demo
Anhang

PHP
Design Pattern
Controller
**View**
Model

## Präsentation

- eigene Template Engine
- CSS Twitter Bootstrap [1]

---

[1] http://getbootstrap.com/

```php
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

Über uns
Werkzeuge
Datenbank
**Programmcode**
Live Demo
Anhang

PHP
Design Pattern
Controller
View
**Model**

## SQL-Injection Beispiel

```php
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;

Über uns
Werkzeuge
Datenbank
**Programmcode**
Live Demo
Anhang

PHP
Design Pattern
Controller
View
Model

# Umfragename holen

```php
1  function getSurveyName($survey) {
2
3      $stmt = $this->dbh->prepare("SELECT Name FROM
       Survey WHERE ID = :id");
4      $stmt->bindParam(':id', $survey);
5      $stmt->execute();
6      $res = $stmt->fetchObject();
7
8      return $res->Name;
9
10 }
```

Listing 10: getSurveyName Funktion

Über uns
Werkzeuge
Datenbank
**Programmcode**
Live Demo
Anhang

PHP
Design Pattern
Controller
View
**Model**

# Umfrage Ergebnisse holen

```php
1  public function getSurveyResult($survey) {
2
3      $stmt = $this->dbh->prepare("SELECT i.Name, COUNT(a
       .ID) as cnt FROM SurveyItems i LEFT JOIN
       SurveyAnswer a ON i.ID = a.SurveyItemID WHERE i.
       SurveyID = :id GROUP BY Name ORDER BY i.Name");
4      $stmt->bindParam(':id', $survey);
5      $stmt->execute();
6
7      return $stmt->fetchAll();
8
9  }
```

Listing 11: getSurveyResult Funktion

Über uns
Werkzeuge
Datenbank
**Programmcode**
Live Demo
Anhang

PHP
Design Pattern
Controller
View
Model

## Abgegebene Antworten Speichern

```php
 1  public function saveNewAnswers($answerArr) {
 2
 3      $stmt = $this->dbh->prepare("INSERT INTO
        SurveyAnswer SET SurveyItemID = :id");
 4
 5      foreach ($answerArr as $answer) {
 6
 7          $stmt->bindParam(':id', $answer);
 8          $stmt->execute();
 9
10      }
11
12  }
```

Listing 12: saveNewAnswers Funktion

Über uns
Werkzeuge
Datenbank
**Programmcode**
Live Demo
Anhang

PHP
Design Pattern
Controller
View
**Model**

# neue Umfrage hinzufügen

```php
1  public function addSurvey($name, $answerArr) {
2
3      if (empty($name)) return;
4
5      $stmt = $this->dbh->prepare("INSERT INTO Survey SET
       Name = :name");
6      $stmt->bindParam(':name', $name);
7
8      $stmt->execute();
9
10     $surveyID = $this->dbh->lastInsertId();
11
12     $stmt = $this->dbh->prepare("INSERT INTO
       SurveyItems SET Name = :name, SurveyID = :id");
13
14     $stmt->bindParam(':id', $surveyID);
15
16     foreach ($answerArr as $answer) {
17
18         if (!empty($answer)) {
19             $stmt->bindParam(':name', $answer);
20             $stmt->execute();
21         }
22
23     }
24
25 }
```

Listing 13: addSurvey Funktion

## Live Demo

# Please wait ... we are connecting ;-)

Über uns
Werkzeuge
Datenbank
Programmcode
Live Demo
Anhang

Quellen

# Anhang

Vielen Dank für Ihre Aufmerksamkeit.
Fragen?

## Links

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

Über uns
Werkzeuge
Datenbank
Programmcode
Live Demo
**Anhang**

Quellen

## Quellen

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