

1 admin_stats.tpl.php

```
00001         <h1>Umfragen Statistik</h1>
00002
00003         <ul class="list-group">
00004             <li class="list-group-item">
00005                 <span class="badge"><?php print $this->view['stats']['user_cnt']; ?></span>
00006                 Benutzer
00007             </li>
00008             <li class="list-group-item">
00009                 <span class="badge"><?php print $this->view['stats']['survey_cnt']; ?></span>
00010                 Umfragen
00011             </li>
00012             <li class="list-group-item">
00013                 <span class="badge"><?php print $this->view['stats']['survey_items_cnt']; ?></span>
00014                 Fragen
00015             </li>
00016             <li class="list-group-item">
00017                 <span class="badge"><?php print $this->view['stats']['answer_cnt']; ?></span>
00018                 Antworten
00019             </li>
00020         </ul>
00021
```

2 admin_surveys.tpl.php

```
00001         <h1>Umfragen Administration</h1>
00002         <table class="table table-hover">
00003             <tr><th>ID</th><th>Umfrage</th><th></th></tr>
00004             <?php
00005                 foreach ($this->view['surveys'] as $arr) {
00006                     print " <tr><td>{$arr['ID']}</td>" .
00007                         " <td><a
00008                             href='?controller=Survey&action=Show&survey={$arr['ID']}'>{$arr['Name']}</a></td>" .
00009                             " <td class='text-right'><a href=\"?controller=Admin\S
00010                             urvey&action=Delete&survey={$arr['ID']}\" class='btn btn-danger'>Löschen</button></td></tr>";
00011                 }
00012             <?>
00013         </table>
00014         <a data-toggle="modal" href="#addSurveyModal" class="btn btn-success">Hinzuf&uuml;gen</a>
00015
00016         <!-- Modal -->
00017         <form class="form-survey" method="post" action="?controller=Admin\Survey&action=Add">
00018             <div class="modal" id="addSurveyModal">
00019                 <div class="modal-dialog">
00020                     <div class="modal-content">
00021                         <div class="modal-header">
00022                             <button type="button" class="close" data-dismiss="modal" aria-hidden="true">&times;</
00023                             button>
00024                             <h4 class="modal-title">Neue Umfrage hinzuf&uuml;gen</h4>
00025                         </div>
00026                         <div class="modal-body">
00027                             <div class="form-group">
00028                                 <label for="name">Umfragenname</label>
00029                                 <input type="text" class="form-control" id="name" name="name" placeholder="
Umfragenname eingeben">
00030                             </div>
00031                         </div>
00032                     </div>
00033                 </div>
00034             </div>
00035         </form>
00036
```

```

00030         <div class="form-group">
00031             <label for="answer">Antwort</label>
00032             <input type="text" class="form-control" id="answer" name="answer[]" placeholder
="Antwort eingeben">
00033         </div>
00034
00035     </div>
00036
00037         <div style="padding: 20px;">
00038             <a class="btn btn-default" onClick="newAnswer()">Weitere Antwort hinzuf&
uuml;gen</a>
00039         </div>
00040
00041     <div class="modal-footer">
00042         <a href="#" class="btn" data-dismiss="modal">Verwerfen</a>
00043         <button class="btn btn-primary" type="submit">Speichern</button>
00044     </div>
00045 </div><!-- /.modal-content -->
00046 </div><!-- /.modal-dialog -->
00047 </div><!-- /.modal -->
00048 </form>
00049
00050 <script>
00051     function newAnswer() {
00052
00053         $(''.modal-body').append('<div class="form-group"><input type="text"
class="form-control" name="answer[]" placeholder="Antwort eingeben"></div>');
00054     }
00055 </script>
00056

```

3 admin_user.tpl.php

```

00001     <h1>Benutzer Administration</h1>
00002     <table class="table table-hover">
00003         <tr><th>Benutername</th><th>Letzer Login</th><th></th></tr>
00004         <?php
00005             foreach ($this->view['users'] as $arr) {
00006                 print " <tr><td>{$arr['Name']}</td>" .
00007                     " <td>{$arr['LastLogin']}</td>" .
00008                     " <td class='text-right'><a href=\"?controller=Admin\U
ser&action=Delete&user={$arr['ID']}" class='btn btn-danger'>Lschen</button></td></tr>";
00009             }
00010         <?>
00011     </table>
00012     <a data-toggle="modal" href="#addUserModal" class="btn btn-success">Hinzuf&uuml;gen</a>
00013
00014 <!-- Modal -->
00015 <form class="form-user" method="post" action="?controller=Admin\User&action=Add">
00016
00017     <div class="modal" id="addUserModal">
00018         <div class="modal-dialog">
00019             <div class="modal-content">
00020                 <div class="modal-header">
00021                     <button type="button" class="close" data-dismiss="modal" aria-hidden="true">&times;</
button>
00022                 <h4 class="modal-title">Neuen Benutzer hinzuf&uuml;gen</h4>
00023                 </div>
00024                 <div class="modal-body">

```

```

00025         <div class="form-group">
00026             <label for="name">Name</label>
00027             <input type="text" class="form-control" id="name" name="name" placeholder="
Benutzername eingeben">
00028         </div>
00029         <div class="form-group">
00030             <label for="pass">Passwort</label>
00031             <input type="text" class="form-control" id="pass" name="pass" placeholder="
Passwort eingeben">
00032         </div>
00033     </div>
00034     <div class="modal-footer">
00035         <a href="#" class="btn" data-dismiss="modal">Verwerfen</a>
00036         <button class="btn btn-primary" type="submit">Speichern</button>
00037     </div>
00038 </div><!-- /.modal-content -->
00039 </div><!-- /.modal-dialog -->
00040 </div><!-- /.modal -->
00041
00042 </form>

```

4 Config.php

```

00001 <?php
00002 define('CONFIGFILE', 'config/config.ini');
00003 /**
00004  *
00005  * Config Klasse als Singleton Pattern
00006  *
00007  * $conf = \Config::getInstance();
00008  * $value = $conf->key;
00009  *
00010  * Die Werte stehen in der Ini Datei und koennen direkt abgerufen werden.
00011  *
00012  */
00013 class Config {
00014
00015     private $config = array();
00016
00017     static private $instance = null;
00018
00019     static public function getInstance() {
00020         if (null === self::$instance) {
00021             self::$instance = new self;
00022         }
00023
00024         return self::$instance;
00025     }
00026
00027     /**
00028      *
00029      * Konstruktor
00030      *
00031      * Wird nur einmal aufgerufen wegen Singleton Pattern
00032      *
00033      * Konfigurationsdatei einlesen
00034      *
00035      */
00036     private function __construct() {

```

```

00037         $this->config = parse_ini_file(CONFIGFILE);
00038     }
00039
00040     private function __clone(){}
00041
00042     /**
00043      *
00044      * Getter Methode
00045      *
00046      * @param      String  $key      Schlssel
00047      *
00048      * @return     String  Wert
00049      *
00050      */
00051     public function __get($key) {
00052
00053         if (array_key_exists($key, $this->config)) {
00054             return $this->config[$key];
00055         } else {
00056             throw new Exception("Config Schluessel '$key' nicht verfuegbar");
00057         }
00058     }
00059 }
00060
00061 }

```

5 Controller/Admin/Base.php

```

00001 <?php
00002 namespace Controller\Admin;
00003
00004 /**
00005  *
00006  * Admin Base Controller
00007  *
00008  *
00009  */
00010 class Base {
00011
00012     protected $model;
00013     protected $view;
00014
00015     /**
00016      *
00017      * Im Konstruktur generell ueberpruefen ob
00018      * sich der User eingeloggt hat.
00019      *
00020      *
00021      */
00022     public function __construct() {
00023
00024         \Session::isUserAuthedCheck();
00025
00026         $this->view = new \View();
00027     }
00028 }
00029
00030 }
00031 ?>

```

6 Controller/Admin/Survey.php

```
00001 <?php
00002 namespace Controller\Admin;
00003
00004 /**
00005  *
00006  * Admin Umfrage Controller
00007  *
00008  *
00009  */
00010 class Survey extends Base {
00011
00012     /**
00013      *
00014      * Modell initialisieren
00015      *
00016      */
00017     public function __construct() {
00018
00019         $this->model = new \Model\Survey();
00020
00021         parent::__construct();
00022
00023     }
00024
00025     /**
00026      *
00027      * Default Index Get Action
00028      *
00029      */
00030     public function Index.Action() {
00031
00032         $this->view->setTemplate('admin_surveys');
00033         $this->view->assign('surveys', $this->model->getSurveys());
00034         $this->view->display();
00035
00036     }
00037
00038     /**
00039      *
00040      * neue Umfragen hinzufuegen
00041      *
00042      */
00043     public function Add.POST.Action() {
00044
00045         if (isset($_POST['name']) && isset($_POST['answer'])) {
00046
00047             $name = $_POST['name'];
00048             $answer_arr = $_POST['answer'];
00049
00050             $this->model->addSurvey($name, $answer_arr);
00051
00052             $this->Index.Action();
00053
00054         } else {
00055
00056             throw new \Exception("Illegaler Aufruf");
00057
00058         }
00059     }
```

```

00060     }
00061
00062     /**
00063     *
00064     * Umfragen loeschen
00065     *
00066     */
00067     public function Delete_Action() {
00068
00069
00070         if (isset($_GET['survey'])) {
00071
00072             $id = intval($_GET['survey']);
00073
00074             if ($id > 0) {
00075
00076                 $this->model->deleteSurvey($id);
00077
00078                 $this->Index_Action();
00079
00080             } else {
00081
00082                 throw new \Exception("Illegaler Aufruf");
00083
00084             }
00085
00086         } else {
00087
00088             throw new \Exception("Illegaler Aufruf");
00089
00090         }
00091     }
00092 }
00093 }
00094 }
00095 ?>

```

7 Controller/Admin/User.php

```

00001 <?php
00002 namespace Controller\Admin;
00003
00004 /**
00005 *
00006 * Admin Benutzer Controller
00007 *
00008 *
00009 */
00010 class User extends Base{
00011
00012
00013     /**
00014     *
00015     * Modell initialisieren
00016     *
00017     */
00018     public function __construct() {
00019
00020         $this->model = new \Model\User();

```

```

00021
00022         parent::__construct();
00023     }
00024 }
00025
00026 /**
00027  *
00028  * Default Index Get Action
00029  *
00030  */
00031 public function Index_Action() {
00032
00033     $this->view->setTemplate('admin_user');
00034     $this->view->assign('users', $this->model->getUsers());
00035
00036     $this->view->display();
00037 }
00038 }
00039
00040 /**
00041  *
00042  * Benutzer hinzufuegen
00043  *
00044  */
00045 public function Add_POST_Action() {
00046
00047     if (isset($_POST['name']) && isset($_POST['pass'])) {
00048
00049         $name = $_POST['name'];
00050         $pass = $_POST['pass'];
00051
00052         $this->model->addUser($name, $pass);
00053
00054         $this->Index_Action();
00055
00056     } else {
00057
00058         throw new \Exception("Illegaler Aufruf");
00059
00060     }
00061 }
00062 }
00063 }
00064
00065 /**
00066  *
00067  * Benutzer loeschen
00068  *
00069  */
00070 public function Delete_Action() {
00071
00072
00073     if (isset($_GET['user'])) {
00074
00075         $id = intval($_GET['user']);
00076
00077         if ($id > 0) {
00078
00079             $this->model->deleteUser($id);
00080
00081             $this->Index_Action();
00082

```

```

00083             } else {
00084
00085                 throw new \Exception("Illegaler Aufruf");
00086
00087             }
00088
00089
00090         } else {
00091
00092             throw new \Exception("Illegaler Aufruf");
00093
00094         }
00095     }
00096 }
00097 ?>

```

8 Controller/Survey.php

```

00001 <?php
00002 namespace Controller;
00003
00004 /**
00005  *
00006  * Umfrage Controller
00007  *
00008  */
00009 class Survey {
00010
00011     private $view;
00012     private $model;
00013     private $survey;
00014
00015     /**
00016      *
00017      * Konstruktor
00018      *
00019      */
00020     public function __construct() {
00021
00022         $this->view = new \View();
00023         $this->model = new \Model\Survey();
00024         if (isset($_GET['survey'])) $this->survey = intval($_GET['survey']);
00025
00026     }
00027
00028
00029     /**
00030      *
00031      * Default Index Get Action
00032      *
00033      */
00034     public function IndexAction() {
00035
00036
00037
00038         if ($this->survey == 0) {
00039
00040             $this->view->setTemplate('surveys');
00041             $this->view->assign('surveys', $this->model->getSurveys());

```



```

00042             $this->view->display();
00043
00044         } else {
00045
00046             $this->view->setTemplate('survey');
00047             $this->view->assign('survey_id', $this->survey);
00048             $this->view->assign('survey_name', $this->model->getSurveyName($this->survey));
00049             $this->view->assign('survey_items', $this->model->getSurveyItems($this->survey));
00050             $this->view->display();
00051
00052
00053         }
00054     }
00055 }
00056
00057 /**
00058  *
00059  * Umfrage Ergebnisse speichern
00060  *
00061  *
00062  *
00063  */
00064 public function Save_POST_Action() {
00065
00066     if ($this->survey > 0 && $_GET['action'] == 'Save') {
00067
00068         $answerArr = $_POST['answer'];
00069
00070         $this->model->saveNewAnswers($answerArr);
00071
00072         $this->showSurveyResult();
00073
00074     } else {
00075
00076         //Exception werfen
00077
00078     }
00079
00080 }
00081
00082
00083 /**
00084  *
00085  * Umfrage Ergebnisse anzeigen
00086  *
00087  * Direkt nur fuer Administratoren moeglich
00088  *
00089  */
00090 public function Show_Action() {
00091
00092     if ($this->survey == 0) throw new \Exception("Illegaler Aufruf");
00093
00094     \Session::isUserAuthedCheck();
00095
00096     $this->showSurveyResult();
00097
00098 }
00099
00100 /**
00101  *
00102  * Auswertung anzeigen
00103  *

```

```

00104         */
00105         private function showSurveyResult() {
00106
00107             $this->view->setTemplate('survey_result');
00108             $this->view->assign('survey_name', $this->model->getSurveyName($this->survey));
00109             $this->view->assign('survey_cnt', $this->model->getSurveyItemCount($this->survey));
00110             $this->view->assign('survey_result', $this->model->getSurveyResult($this->survey));
00111             $this->view->display();
00112
00113         }
00114
00115     }
00116
00117
00118 ?>

```

9 Database.php

```

00001 <?php
00002 /**
00003  *
00004  * Database Klasse erweitert PHP PDO
00005  *
00006  * $db = new Database();
00007  *
00008  */
00009 class Database extends PDO {
00010
00011     private $dsn;
00012     private $user;
00013     private $pass;
00014
00015     /**
00016      *
00017      * Konstruktor.
00018      *
00019      * Konfiguration laden und der Elternklasse uebergeben.
00020      *
00021      */
00022     public function __construct() {
00023
00024         $conf = \Config::getInstance();
00025
00026         $this->dsn = $conf->database_type . " :host=" . $conf->database_host . " ;port=" . $conf->
database_port . " ;dbname=" . $conf->database_name;
00027
00028         $this->user = $conf->database_user;
00029         $this->pass = $conf->database_pass;
00030
00031         parent::__construct($this->dsn, $this->user, $this->pass);
00032
00033
00034         if ($conf->database_verbose == 1) $this->beMoreVerbose();
00035
00036     }
00037
00038
00039
00040     private function beMoreVerbose() {

```

```

00041
00042             $this->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
00043     $this->setAttribute(PDO::ATTR_EMULATE_PREPARES, false);
00044
00045     }
00046
00047
00048 }

```

10 exception.tpl.php

```

00001     <div class="jumbotron">
00002         <h1><?php print $this->view['h1']; ?></h1>
00003         <p><?php print $this->view['exception']; ?></p>
00004     </div>

```

11 FrontController.php

```

00001 <?php
00002 // Namespace der Controller definieren
00003 define('CONTROLLER_NAMESPACE', '\\Controller\\');
00004
00005 /**
00006  *
00007  * FrontController Klasse
00008  *
00009  * Get Request: ?controller=test&action=machwas
00010  *
00011  * Ergebniss: Automatisches Einbinden der Controllerklasse "test"
00012  * und Aufruf der Methode machwas_Action()
00013  *
00014  * Beim gleichen Request als Post wird die Methode machwas_Action.POST() aufgerufen
00015  *
00016  *
00017  */
00018 class FrontController {
00019
00020     private $controller;
00021     private $action;
00022
00023     /**
00024      *
00025      * Konstruktor
00026      *
00027      * $c = new FrontCrontroller()
00028      * $c->run();
00029      *
00030      */
00031     public function __construct() {
00032
00033         if (isset($_GET['controller'])) $controller = $_GET['controller'];
00034
00035         // Workaround magic_quotes() . Ab PHP 5.4 DEPRICATED
00036         if (get_magic_quotes_gpc()) {
00037             $controller = str_replace("\\\\", "\\", $controller);
00038         }
00039

```

```

00040             if (isset($_GET['action'])) $action = $_GET['action'];
00041
00042
00043             $this->controller = !empty($controller) ? CONTROLLER_NAMESPACE . $controller :
CONTROLLER_NAMESPACE . "Index";
00044
00045             $this->action = !empty($action) ? $action : "Index";
00046             if ($_SERVER['REQUEST_METHOD'] == 'POST') {
00047                 $this->action .= "_POST";
00048             }
00049             $this->action .= "_Action";
00050
00051         }
00052
00053         /**
00054          *
00055          * FrontController ausfuehren
00056          *
00057          */
00058         public function run() {
00059
00060             $controller = new $this->controller();
00061
00062             if(in_array($this->action, get_class_methods($controller))) {
00063                 $controller->{$this->action}();
00064             } else {
00065                 throw new Exception("'{$this->controller}' hat keine Aktion '{$this->action}'");
00066             }
00067
00068         }
00069     }
00070 }
00071
00072 ?>

```

12 Index.php

```

00001 <?php
00002 namespace Controller;
00003
00004 /**
00005  *
00006  * Index Controller
00007  *
00008  * Wird default immer Aufgerufen
00009  *
00010  */
00011 class Index {
00012
00013     /**
00014      *
00015      * Default Index Get Action
00016      *
00017      */
00018     public function Index_Action() {
00019
00020         $view = new \View();
00021         $view->setTemplate('index');
00022         $view->display();
00023     }
00024 }

```

```

00023
00024     }
00025 }
00026 ?>

```

13 index.php

```

00001 <?php
00002 define('CLASS_DIR', 'classes/');
00003
00004 // PHP Autoloader Funktion definieren.
00005 spl_autoload_register('__autoload');
00006
00007 /**
00008  *
00009  * Wir fr den Autoloader von Klassen bentigt
00010  *
00011  * @param      String  $class  Klassenname
00012  *
00013  */
00014 function __autoload($class) {
00015
00016     // Namespace Backslash in Slashe fr das Filesystem umwandeln
00017     $file = str_replace("\\", DIRECTORY_SEPARATOR, $class).".php";
00018
00019     // Klasse einbinden ansonsten Exception werfen.
00020     if (file_exists(CLASS_DIR . $file)) {
00021         require_once CLASS_DIR . $file;
00022     } else {
00023         throw new Exception("Autoloaderfehler: " . $file . " not found.");
00024     }
00025 }
00026
00027 try {
00028
00029     // Session Initialisieren
00030     $session = \Session::getInstance();
00031
00032     // Frontcontroller Initialisieren und Aufrufen
00033     $c = new \FrontController();
00034     $c->run();
00035
00036 } catch (CustomException\UserNotAuthedException $e) {
00037
00038     $view = new \View();
00039     $view->setTemplate('exception');
00040     $view->assign('h1', "Illegalen Aufruf");
00041     $view->assign('exception', $e->getMessage());
00042     $view->display();
00043
00044 } catch (CustomException\InvalidUserPassException $e) {
00045
00046     $view = new \View();
00047     $view->setTemplate('exception');
00048     $view->assign('h1', "Loginfehler");
00049     $view->assign('exception', "Benutzername und/oder Passwort falsch");
00050     $view->display();
00051
00052

```

```

00053
00054 } catch (Exception $e) {
00055
00056     $view = new \View();
00057     $view->setTemplate('exception');
00058     $view->assign('h1', "Das h&auml;tte nicht passieren d&uuml;rfe");
00059     $view->assign('exception', $e->getMessage());
00060     $view->display();
00061
00062 }
00063
00064
00065 $conf = \Config::getInstance();
00066
00067 // Debugging
00068 if ($conf->application_debugging == 1) {
00069     print "<br /><br />";
00070     print "<pre>";
00071     print '$_GET ';
00072     print_r($_GET);
00073     print '$_POST ';
00074     print_r($_POST);
00075     print '$_SESSION ';
00076     print_r($_SESSION);
00077     print '$_SERVER ';
00078     print_r($_SERVER);
00079     print "</pre>";
00080 }
00081
00082
00083 ?>

```

14 index.tpl.php

```

00001
00002 <div class="jumbotron">
00003     <h1>Umfrage System</h1>
00004     <p>Zum Teilnehmen an einer Umfrage klicken Sie bitte auf Umfragen.</p>
00005     <p>Haben Sie eine Benutzerkennung? Einfach oben rechts anmelden.</p>
00006     <br /><br />
00007     <p>
00008         <a class="btn btn-lg btn-primary" href="?controller=Survey">zu den Umfragen &raquo;</a>
00009     </p>
00010 </div>
00011

```

15 InvalidUserPassException.php

```

00001 <?php
00002 namespace CustomException;
00003
00004 /**
00005  *
00006  * Diese Exception wird geworfen wenn der Benutzer
00007  * nicht vorhanden oder das Passwort falsch ist.
00008  *
00009  */

```

```

00010 class InvalidUserPassException extends \Exception {}
00011
00012 ?>

```

16 _layout.tpl.php

```

00001 <!DOCTYPE html>
00002 <html lang="en">
00003     <head>
00004         <meta charset="utf-8">
00005         <meta name="viewport" content="width=device-width, initial-scale=1.0">
00006         <meta name="description" content="">
00007         <meta name="author" content="">
00008         <link rel="shortcut icon" href="ico/favicon.gif">
00009
00010         <title>DBA02</title>
00011
00012         <!-- Bootstrap core CSS -->
00013         <link href="css/bootstrap.css" rel="stylesheet">
00014
00015         <!-- Custom styles for this template -->
00016         <link href="css/navbar-fixed-top.css" rel="stylesheet">
00017
00018         <!-- HTML5 shim and Respond.js IE8 support of HTML5 elements and media queries -->
00019         <!--[if lt IE 9]>
00020             <script src="js/html5shiv.js"></script>s
00021             <script src="js/respond.min.js"></script>
00022         <![endif]-->
00023     </head>
00024
00025     <body>
00026
00027         <!-- Fixed navbar -->
00028         <div class="navbar navbar-default navbar-fixed-top">
00029             <div class="container">
00030                 <div class="navbar-header">
00031                     <button type="button" class="navbar-toggle" data-toggle="collapse" data-target=".navbar-collapse">
00032                         <span class="icon-bar"></span>
00033                         <span class="icon-bar"></span>
00034                         <span class="icon-bar"></span>
00035                     </button>
00036                     <a class="navbar-brand" href="index.php">DBA02</a>
00037                 </div>
00038                 <div class="navbar-collapse collapse">
00039                     <ul class="nav navbar-nav">
00040                         <li><a href="?controller=Survey">Umfragen</a></li>
00041                         <?php if (\Session::isUserAuthed()) { ?>
00042                         <li class="dropdown">
00043                             <a href="#" class="dropdown-toggle" data-toggle="dropdown">Administration <b class="caret"></b></a>
00044                             <ul class="dropdown-menu">
00045                                 <li class="dropdown-header">Verwaltung</li>
00046                                 <li><a href="?controller=Admin\User">Benutzer</a></li>
00047                                 <li><a href="?controller=Admin\Survey">Umfragen</a></li>
00048                                 <li class="divider"></li>
00049                                 <li class="dropdown-header">Statistik</li>
00050                                 <li><a href="?controller=Admin\Stats">Umfragen</a></li>
00051                             </ul>

```

```

00052         <?php } ?>
00053     </li>
00054 </ul>
00055     <ul class="nav navbar-nav navbar-right">
00056         <li><a href="docs/html/index.html" target="_blank">Code Dokumentation</a></li>
00057     </ul>
00058     <?php if (\Session::isUserAuthenticated()) { ?>
00059         <li><a href="?controller=UserSession&action=Logout">Logout</a></li>
00060     <?php } else { ?>
00061         <li class="dropdown">
00062             <a class="dropdown-toggle" href="#" data-toggle="dropdown">Login <strong class="
00063                 "caret"></strong></a>
00064             <div class="dropdown-menu" style="padding: 15px; padding-bottom: 0px;">
00065                 <link href="css/signin.css" rel="stylesheet">
00066                 <form class="form-signin" method="post" action="
00067                     ?controller=UserSession&action=Login">
00068                     <input type="text" name="user" class="form-control"
00069                     placeholder="Benutzername" autofocus>
00070                     <input type="password" name="pass" class="form-control"
00071                     placeholder="Passwort">
00072                     <button class="btn btn-primary btn-block" type="submit">
00073                         Anmelden</button>
00074                     </form>
00075                 </div>
00076             </li>
00077         </ul>
00078     </div><!--/.nav-collapse -->
00079 </div>
00080 </div>
00081 </div>
00082 <div class="container">
00083     <?php include($this->includetemplate); ?>
00084 </div> <!-- /container -->
00085 </div>
00086 <!-- Bootstrap core JavaScript
00087 ===== -->
00088 <!-- Placed at the end of the document so the pages load faster -->
00089 <script src="js/jquery.js"></script>
00090 <script src="js/bootstrap.min.js"></script>
00091 </body>
00092 </html>

```

17 Model/Base.php

```

00001 <?php
00002 namespace Model;
00003
00004 /**
00005  *
00006  * Basis Datenmodell
00007  *

```



```

00008 */
00009 class Base {
00010
00011     // Datenbank Handler
00012     protected $dbh;
00013
00014     /**
00015      *
00016      * Konstruktor
00017      *
00018      * Datenbank Verbindung herstellen.
00019      *
00020      */
00021     public function __construct() {
00022
00023         $this->dbh = new \Database();
00024
00025     }
00026
00027
00028
00029 }
00030
00031
00032 ?>

```

18 Model/Survey.php

```

00001 <?php
00002 namespace Model;
00003
00004 /**
00005  *
00006  * Umfrage Datenmodell
00007  *
00008  */
00009 class Survey extends Base{
00010
00011     /**
00012      *
00013      * Umfragen holen
00014      *
00015      * @return      Array
00016      *
00017      */
00018     public function getSurveys() {
00019
00020         $stmt = $this->dbh->query("SELECT * FROM Survey");
00021         return $stmt->fetchAll();
00022
00023     }
00024
00025     /**
00026      *
00027      * Umfragenname holen
00028      *
00029      * @param      Integer $survey UmfrageId
00030      *
00031      * @return      String Name der Umfrage

```

```

00032      *
00033      */
00034      public function getSurveyName($survey) {
00035
00036          $stmt = $this->dbh->prepare("SELECT Name FROM Survey WHERE ID = :id");
00037          $stmt->bindParam(':id', $survey);
00038          $stmt->execute();
00039          $res = $stmt->fetchObject();
00040
00041          return $res->Name;
00042
00043      }
00044
00045      /**
00046      *
00047      * Umfrage Punkte holen
00048      *
00049      * @param      Integer $survey UmfrageId
00050      *
00051      * @return      Array
00052      *
00053      */
00054      public function getSurveyItems($survey) {
00055
00056          $stmt = $this->dbh->prepare("SELECT ID, Name FROM SurveyItems WHERE SurveyID = :id ORDER BY
00057      Name");
00058          $stmt->bindParam(':id', $survey);
00059          $stmt->execute();
00060
00061          return $stmt->fetchAll();
00062
00063      }
00064
00065      /**
00066      *
00067      * Anzahl der Abgegeben Umfragewerte
00068      *
00069      * @param      Integer $survey UmfrageId
00070      *
00071      * @return      Integer Anzahl der Ergebnisse zur Umfrage
00072      *
00073      */
00074      public function getSurveyItemCount($survey) {
00075
00076          $stmt = $this->dbh->prepare("SELECT COUNT(*) AS cnt FROM SurveyAnswer WHERE SurveyItemID IN
00077      (SELECT ID FROM SurveyItems WHERE SurveyID = :id)");
00078          $stmt->bindParam(':id', $survey);
00079          $stmt->execute();
00080          $res = $stmt->fetchObject();
00081
00082          return $res->cnt;
00083
00084      }
00085
00086      /**
00087      *
00088      * Umfrage Ergebnisse holen
00089      *
00090      * @param      Integer $survey UmfrageId
00091      *

```

```

00092         * @return      Array
00093     */
00094     public function getSurveyResult($survey) {
00095
00096         $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");
00097         $stmt->bindParam(':id', $survey);
00098         $stmt->execute();
00099
00100         return $stmt->fetchAll();
00101     }
00102 }
00103
00104 /**
00105  *
00106  * Statistiken holen
00107  *
00108  */
00109 public function getStats() {
00110
00111     $stat = array();
00112
00113     $stat['user_cnt'] = $this->getTableCount("User");
00114     $stat['survey_cnt'] = $this->getTableCount("Survey");
00115     $stat['survey_items_cnt'] = $this->getTableCount("SurveyItems");
00116     $stat['answer_cnt'] = $this->getTableCount("SurveyAnswer");
00117
00118     return $stat;
00119 }
00120
00121 /**
00122  *
00123  * Antworten Speichern
00124  *
00125  * @param      Integer $answerArr      Array mit den Antwort IDs
00126  *
00127  */
00128 public function saveNewAnswers($answerArr) {
00129
00130     $stmt = $this->dbh->prepare("INSERT INTO SurveyAnswer SET SurveyItemID = :id");
00131
00132     foreach ($answerArr as $answer) {
00133
00134         $stmt->bindParam(':id', $answer);
00135         $stmt->execute();
00136
00137     }
00138 }
00139
00140
00141
00142 /**
00143  *
00144  * Umfrage lschen
00145  *
00146  * @param      Integer $id      UmfragenID
00147  *
00148  */
00149 public function deleteSurvey($id) {
00150
00151     $stmt = $this->dbh->prepare("DELETE FROM Survey WHERE ID = :id");
00152     $stmt->bindParam(':id', $id);

```

```

00153
00154         $stmt->execute();
00155     }
00156 }
00157
00158 /**
00159  *
00160  * neue Umfrage hinzufügen
00161  *
00162  * @param      String  $name      Name der Umfrage
00163  * @param      Array   $answerArr  Array mit den mglichen Antworten
00164  *
00165  */
00166
00167 public function addSurvey($name, $answerArr) {
00168
00169
00170         $stmt = $this->dbh->prepare("INSERT INTO Survey SET Name = :name");
00171         $stmt->bindParam(':name', $name);
00172
00173         $stmt->execute();
00174
00175         $surveyID = $this->dbh->lastInsertId();
00176
00177         $stmt = $this->dbh->prepare("INSERT INTO SurveyItems SET Name = :name, SurveyID = :id");
00178
00179         $stmt->bindParam(':id', $surveyID);
00180
00181         foreach ($answerArr as $answer) {
00182
00183             $stmt->bindParam(':name', $answer);
00184             $stmt->execute();
00185
00186         }
00187     }
00188 }
00189
00190 /**
00191  *
00192  * Anzahl der Tupel zurckgeben
00193  *
00194  * @param      String  $table  Tabellenname
00195  *
00196  * @return      Integer  Anzahl der Tupel
00197  *
00198  */
00199 private function getTableCount($table) {
00200
00201
00202         switch ($table) {
00203
00204             case "User":
00205             case "Survey":
00206             case "SurveyItems":
00207             case "SurveyAnswer":
00208                 $stmt = $this->dbh->query("SELECT COUNT(*) AS cnt FROM $table");
00209                 return $stmt->fetchObject()->cnt;
00210                 break;
00211             default:
00212                 throw new \Exception("Illegaler Tabellenname $table");
00213
00214         }

```

```

00215
00216     }
00217
00218 }
00219
00220
00221
00222 ?>

```

19 Model/User.php

```

00001 <?php
00002 namespace Model;
00003
00004 /**
00005  *
00006  * Benutzer Datenmodell
00007  *
00008  */
00009 class User extends Base {
00010
00011
00012     /**
00013      *
00014      * Berechtigung ueberprfen
00015      *
00016      * @param      String $user  Benutzername
00017      * @param      String $pass  Passwort
00018      *
00019      * @return      Boolean
00020      *
00021      */
00022     public function checkCredentials($user, $pass) {
00023
00024         if (empty($user) || empty($pass)) return false;
00025
00026         $stmt = $this->dbh->prepare("SELECT Passwort FROM User WHERE Name = :name");
00027         $stmt->bindParam(':name', $user);
00028
00029         $stmt->execute();
00030         $res = $stmt->fetchObject();
00031
00032         if ($res->Passwort == $pass) {
00033
00034             $this->updateLastLogIn($user);
00035             return true;
00036
00037         } else {
00038
00039             return false;
00040
00041         }
00042     }
00043
00044
00045     /**
00046      *
00047      * Benutzer holen
00048      *

```

```

00049         * @return      Array
00050         *
00051         */
00052     public function getUsers() {
00053
00054         $stmt = $this->dbh->query("SELECT ID, Name, LastLogIn FROM User");
00055         return $stmt->fetchAll();
00056     }
00057
00058     /**
00059     *
00060     * Benutzer lschen
00061     *
00062     * @param      Integer $id      BenutzerID
00063     *
00064     */
00065     public function deleteUser($id) {
00066
00067         $stmt = $this->dbh->prepare("DELETE FROM User WHERE ID = :id");
00068         $stmt->bindParam(':id', $id);
00069
00070         $stmt->execute();
00071     }
00072
00073     /**
00074     *
00075     * Benutzer hinzufügen
00076     *
00077     * @param      String $name      Benutzername
00078     * @param      String $pass      Passwort
00079     *
00080     */
00081     public function addUser($name, $pass) {
00082
00083         $stmt = $this->dbh->prepare("INSERT INTO User SET Name = :name, Passwort = :pass");
00084         $stmt->bindParam(':name', $name);
00085         $stmt->bindParam(':pass', $pass);
00086
00087         $stmt->execute();
00088     }
00089
00090     /**
00091     *
00092     * Zeitstempel des letzten Logins vom User Updaten
00093     *
00094     * @param      String $user      Benutzername
00095     *
00096     */
00097     private function updateLastLogIn($user) {
00098
00099         $stmt = $this->dbh->prepare("UPDATE User SET LastLogIn = :lastlogin WHERE Name = :name");
00100         $stmt->bindParam(':name', $user);
00101         $stmt->bindValue(':lastlogin', date("Y-m-d H:i:s"));
00102
00103         $stmt->execute();
00104     }
00105
00106 }
00107
00108
00109
00110

```

```

00111 }
00112
00113
00114 ?>

```

20 Redirect.php

```

00001 <?php
00002 /**
00003  *
00004  * Redirect Klasse
00005  *
00006  */
00007 class Redirect {
00008
00009     /**
00010      *
00011      * Header Redirect durchfuehren
00012      *
00013      * @param      String $url      URL
00014      *
00015      */
00016     public static function Header($url) {
00017
00018         header("Location: $url");
00019     }
00020
00021     /**
00022      *
00023      * Controller Redirect durchfuehren
00024      *
00025      * @param      String $controller      Controllername
00026      *
00027      */
00028     public static function toController($controller) {
00029
00030         header("Location: index.php?controller=$controller");
00031     }
00032
00033 }
00034
00035
00036
00037 ?>

```

21 Session.php

```

00001 <?php
00002
00003 use \CustomException as Ex;
00004
00005 /**
00006  *
00007  * Session Klasse als Singleton Pattern
00008  *
00009  * $session = \Session::getInstance();
00010  * $session->key = "value";

```

```

00011 *
00012 *
00013 * \Session::destroy();
00014 *
00015 *
00016 */
00017 class Session {
00018
00019     private $session = array();
00020
00021     static private $instance = null;
00022
00023     static public function getInstance() {
00024         if (null === self::$instance) {
00025             self::$instance = new self;
00026         }
00027
00028         return self::$instance;
00029     }
00030
00031     private function __construct() {
00032         session_start();
00033     }
00034
00035     private function __clone(){}
00036
00037     /**
00038     *
00039     * Getter Methode
00040     *
00041     * @param      String  $key      Schlssel
00042     *
00043     * @return     String  Wert
00044     *
00045     */
00046     public function __get($key) {
00047
00048         if (array_key_exists($key, $_SESSION)) {
00049             return $_SESSION;
00050         } else {
00051             throw new Exception("Session Schluessel '$key' nicht verfuegbar");
00052         }
00053     }
00054 }
00055
00056 /**
00057 *
00058 * Setter Methode
00059 *
00060 * @param      String  $key      Schlssel
00061 * @param      String  $value    Wert
00062 *
00063 */
00064 public function __set($key, $value) {
00065
00066     $_SESSION[$key] = $value;
00067 }
00068 }
00069
00070 /**
00071 *
00072 * Benutzer Authentifizieren

```



```

00073      *
00074      */
00075      public static function authUser() {
00076
00077          $_SESSION['__AUTH'] = 1;
00078      }
00079
00080      /**
00081      *
00082      * Ueberprfen ob Benutzer Authentifiziert ist
00083      *
00084      * @return      Boolean
00085      *
00086      */
00087      public static function isUserAuthed() {
00088
00089          if (array_key_exists('__AUTH', $_SESSION) && $_SESSION['__AUTH'] == 1) return true;
00090
00091          return false;
00092      }
00093
00094      /**
00095      *
00096      * Ueberprfen ob Benutzer Authentifiziert ist
00097      *
00098      * Ansonsten Exception werfen
00099      *
00100      */
00101      public static function isUserAuthedCheck() {
00102
00103          if (!self::isUserAuthed()) throw new Ex\UserNotAuthedException("
00104          Sie haben keine Berechtigung fr diese Seite");
00105      }
00106
00107      /**
00108      *
00109      * Session loeschen
00110      *
00111      */
00112      public static function destroy() {
00113
00114          session_destroy();
00115      }
00116
00117      }
00118
00119  }
00120 }

```

22 Stats.php

```

00001 <?php
00002 namespace Controller\Admin;
00003
00004 /**
00005 *
00006 * Admin Statistik Controller
00007 *

```

```

00008  *
00009  */
00010  class Stats extends Base {
00011
00012
00013      /**
00014      *
00015      * Modell initialisieren
00016      *
00017      */
00018      public function __construct() {
00019
00020          $this->model = new \Model\Survey();
00021
00022          parent::__construct();
00023
00024      }
00025
00026      /**
00027      *
00028      * Default Index Get Action
00029      *
00030      */
00031      public function Index_Action() {
00032
00033          $this->view->setTemplate('admin_stats');
00034          $this->view->assign('stats', $this->model->getStats());
00035          $this->view->display();
00036
00037      }
00038  }
00039  ?>

```

23 survey.tpl.php

```

00001
00002      <div class="jumbotron">
00003          <form role="form" method="post" action="?controller=Survey&action=Save&survey=?php print
00004      $this->view['survey_id']; ?>">
00005          <div class="panel panel-default">
00006              <div class="panel-heading">
00007                  <h3 class="panel-title"><?php print $this->view['survey_name']; ?></h3>
00008              </div>
00009              <div class="panel-body">
00010                  <?php
00011                  foreach ($this->view['survey_items'] as $arr) {
00012                      print "<div class='checkbox'>" .
00013                          "<label>" .
00014                          "name='answer[]'>{$arr['Name']}" .
00015                          "<input type='checkbox' value='{$arr['ID']}'>" .
00016                          "</label>" .
00017                          "</div>";
00018                      }
00019                  ?>
00020                  <button type="submit" class="btn btn-primary">Absenden</button>
00021              </div>
00022          </div>
00023      </form>
00024  </div>

```

24 survey_result.tpl.php

```
00001 <div class="jumbotron">
00002 <div class="panel panel-default">
00003 <div class="panel-heading">
00004 <h3 class="panel-title"><?php print $this->view['survey_name']; ?></h3>
00005 </div>
00006 <table class="table table-bordered table-survey">
00007 <?php
00008     foreach ($this->view['survey_result'] as $arr) {
00009
00010         if ($this->view['survey_cnt'] > 0) {
00011             $percent = round($arr['cnt'] / $this->view['survey_cnt'] * 100);
00012         } else {
00013             $percent = 0;
00014         }
00015
00016         print "<tr><td>{$arr['Name']}</td>" .
00017             "<td>" .
00018             "<div style='background-color:#428BCA; width:{$percent}%;"
padding-left:10px; border-radius: 18px;'>{$arr['cnt']} ($percent %)</div>" .
00019             "</td>" .
00020             "</tr>";
00021     }
00022     ?>
00023
00024 </table>
00025 </div>
00026 </div>
00027 </div>
```

25 surveys.tpl.php

```
00001 <div class="jumbotron">
00002 <h1>Bitte Umfrage ausw&auml;hlen</h1>
00003 <ul class="list-group">
00004 <?php
00005     foreach ($this->view['surveys'] as $arr) {
00006         print "<li class='list-group-item'><a
00007 href='?controller=Survey&survey={$arr['ID']}'>{$arr['Name']}</a></li>";
00008     }
00009     ?>
00010 </ul>
00011 </div>
```

26 UserNotAuthedException.php

```
00001 <?php
00002 namespace CustomException;
00003
00004 /**
00005  *
00006  * Diese Exception wird geworfen wenn der Benutzer
00007  * nicht eingeloggt ist.
00008  */
```

```

00009  */
00010 class UserNotAuthedException extends \Exception {}
00011
00012 ?>

```

27 UserSession.php

```

00001 <?php
00002 namespace Controller;
00003
00004 use \CustomException as Ex;
00005
00006 /**
00007  *
00008  * Login Controller
00009  *
00010  *
00011  */
00012 class UserSession {
00013
00014
00015     /**
00016      *
00017      * Verarbeitung der Logindaten
00018      *
00019      */
00020     public function Login_POST_Action() {
00021
00022         $user = $_POST['user'];
00023         $pass = $_POST['pass'];
00024
00025         $model = new \Model\User();
00026
00027         if ($model->checkCredentials($user, $pass)) {
00028
00029             \Session::authUser();
00030             \Redirect::toController("Index");
00031
00032         } else {
00033
00034             throw new Ex\InvalidUserPassException();
00035
00036         }
00037
00038     }
00039
00040
00041     /**
00042      *
00043      * Logout Action
00044      *
00045      */
00046     public function Logout_Action() {
00047
00048         \Session::destroy();
00049         \Redirect::toController("Index");
00050
00051     }
00052

```

```

00053 }
00054
00055
00056 ?>

```

28 View.php

```

00001 <?php
00002 /**
00003  *
00004  * View Klasse
00005  *
00006  */
00007 class View {
00008
00009     // Template Pfad
00010     private $templatePath = 'templates';
00011     // Default Template index.tpl.php
00012     private $template = 'index';
00013     // Layout Template
00014     private $layouttemplate = '_layout';
00015     // Anzuzeigendes Template inkl Pfad;
00016     private $includetemplate;
00017     // Variablen innerhalb des Templates
00018     private $view = array();
00019
00020
00021     /**
00022      *
00023      * Template Variable zuweisen
00024      *
00025      * @param      String $key      Schluessel
00026      * @param      String $value     Wert
00027      *
00028      */
00029     public function assign($key, $value) {
00030         $this->view[$key] = $value;
00031     }
00032
00033     /**
00034      *
00035      * Template setzen
00036      *
00037      * @param      String $template     Templatenamen
00038      *
00039      */
00040     public function setTemplate($template) {
00041         $this->template = $template;
00042     }
00043
00044     /**
00045      *
00046      * Template anzeigen
00047      *
00048      */
00049     public function display() {
00050
00051         $this->includetemplate = $this->getTemplateOnFilesystem($this->
template);

```

```

00052
00053         // berprfen ob das angeforderte Template Existiert
00054         if (file_exists($this->includetemplate)) {
00055             // Layouttemplate einbinden
00056             include $this->getTemplateOnFilesystem($this->layouttemplate
    );
00057
00058         } else {
00059
00060             throw new Exception("Template '{$this->template}' nicht gefunden");
00061
00062         }
00063     }
00064 }
00065
00066 /**
00067  *
00068  * Anhand des Templatennamen den vollen Pfad inkl Datei zurueckgeben
00069  *
00070  * @param      String $template      Templatename
00071  *
00072  * @return     String Pfad inkl Dateiname zum Template
00073  *
00074  */
00075 private function getTemplateOnFilesystem($template)    {
00076
00077     return $this->templatePath . DIRECTORY_SEPARATOR . $template . ".tpl.php";
00078
00079 }
00080 }
00081 ?>

```