

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'][survey_cnt]; ?></span>
00006                 Umfragen
00007             </li>
00008             <li class="list-group-item">
00009                 <span class="badge"><?php print $this->view['stats'][record_cnt]; ?></span>
00010                 Antworten
00011             </li>
00012         </ul>
00013
```

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 $key=>$value) {
00006                     print " <tr><td>$key</td>" .
00007                         " <td><a
00008                             href='?controller=Survey&action=Show&survey=$key'>$value</a></td>" .
00009                             " <td class='text-right'><a href=\"?controller=Admin\S
00010                             urvey&action=Delete&survey=$key\" class='btn btn-danger'>Löschen</button></td></tr>";
00011                 }
00012             </table>
00013             <a data-toggle="modal" href="#addSurveyModal" class="btn btn-success">Hinzufügen</a>
00014
00015             <!-- Modal -->
00016             <form class="form-survey" method="post" action="?controller=Admin\Survey&action=Add">
00017
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 hinzufügen</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="
00030                                     Umfragenname eingeben">
00031                                 </div>
00032
00033                                 <div class="form-group">
00034                                     <label for="answer">Antwort</label>
00035                                     <input type="text" class="form-control" id="answer" name="answer[]" placeholder
00036                                     ="Antwort eingeben">
00037                                 </div>
00038                             </div>
00039                         </div>
00040                     </div>
00041                 </div>
00042             </form>
00043
```

```

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         $('<div class="form-group"><input type="text"
class="form-control" name="answer[]" placeholder="Antwort eingeben"></div>');
00054
00055     }
00056 </script>

```

3 admin_user.tpl.php

```

00001         <h1>Benutzer Administration</h1>
00002         <table class="table table-hover">
00003             <tr><th>Benutzername</th><th>Letzter Login</th><th></th></tr>
00004             <?php
00005                 foreach ($this->view['users'] as $userid=>$lastlogin) {
00006                     print " <tr><td>$userid</td>" .
00007                         "<td>$lastlogin</td>" .
00008                         "<td class='text-right'><a href=\"?controller=Admin\U
ser&action=Delete&user=$userid\" class='btn btn-danger'>Löschen</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
00023                         <h4 class="modal-title">Neuen Benutzer hinzuf&uuml;gen</h4>
00024                     </div>
00025                     <div class="modal-body">
00026                         <div class="form-group">
00027                             <label for="name">Name</label>
00028                             <input type="text" class="form-control" id="name" name="name" placeholder="
Benutzername eingeben">
00029                         </div>
00030                         <div class="form-group">
00031                             <label for="pass">Passwort</label>
00032                             <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     /**
00013      *
00014      * Im Konstruktor generell ueberpruefen ob
00015      * sich der User eingeloggt hat.
00016      *
00017      */
00018     public function __construct() {
00019
00020         \Session::isUserAuthedCheck();
00021     }
00022 }
00023
00024 }
00025 ?>

```

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     * Default Index Get Action
00015     *
00016     */
00017     public function Index_Action() {
00018
00019         $model = new \Model\Survey();
00020
00021         $view = new \View();
00022         $view->setTemplate('admin_surveys');
00023         $view->assign('surveys', $model->getSurveys());
00024         $view->display();
00025
00026     }
00027
00028     /**
00029     *
00030     * neue Umfragen hinzufuegen
00031     *
00032     */
00033     public function Add_POST_Action() {
00034
00035         $this->Index_Action();
00036
00037     }
00038
00039     /**
00040     *
00041     * Umfragen loeschen
00042     *
00043     */
00044     public function Delete_Action() {
00045
00046         $this->Index_Action();
00047
00048     }
00049 }
00050 ?>

```

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     * Default Index Get Action
00015     *

```

```

00016         */
00017     public function Index_Action() {
00018
00019         $model = new \Model\User();
00020
00021         $view = new \View();
00022         $view->setTemplate('admin_user');
00023         $view->assign('users', $model->getUsers());
00024
00025         $view->display();
00026
00027     }
00028
00029     /**
00030     *
00031     * Benutzer hinzufuegen
00032     *
00033     */
00034     public function Add_POST_Action() {
00035
00036         $this->Index_Action();
00037
00038     }
00039
00040     /**
00041     *
00042     * Benutzer loeschen
00043     *
00044     */
00045     public function Delete_Action() {
00046
00047         $this->Index_Action();
00048     }
00049 }
00050 ?>

```

8 Controller/Survey.php

```

00001 <?php
00002 namespace Controller;
00003
00004 /**
00005 *
00006 * Umfrage Controller
00007 *
00008 *
00009 */
00010 class Survey {
00011
00012     private $view;
00013     private $model;
00014     private $survey;
00015
00016     /**
00017     *
00018     * Kontruktor
00019     *
00020     */
00021     public function __construct() {

```

```

00022
00023         $this->view = new \View();
00024         $this->model = new \Model\Survey();
00025         $this->survey = intval($_GET['survey']);
00026
00027     }
00028
00029     /**
00030     *
00031     * Default Index Get Action
00032     *
00033     */
00034     public function Index_Action() {
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         //Speichern Wenn > 0
00067
00068         $this->showSurveyResult();
00069
00070     }
00071
00072     /**
00073     *
00074     * Umfrage Ergebnisse anzeigen
00075     *
00076     * Direkt nur fuer Administratoren moeglich
00077     *
00078     */
00079     public function Show_Action() {
00080
00081         if ($this->survey == 0) throw new \Exception("Illegaler Aufruf");
00082
00083         \Session::isUserAuthedCheck();

```

```

00084
00085         $this->showSurveyResult();
00086     }
00087 }
00088
00089 /**
00090  *
00091  * Auswertung anzeigen
00092  *
00093  */
00094 private function showSurveyResult() {
00095
00096     $this->view->setTemplate('survey_result');
00097     $this->view->assign('survey_name', $this->model->getSurveyName($this->survey));
00098     $this->view->assign('survey_cnt', $this->model->getSurveyItemCount($this->survey));
00099     $this->view->assign('survey_result', $this->model->getSurveyResult($this->survey));
00100     $this->view->display();
00101 }
00102 }
00103 }
00104 }
00105
00106
00107 ?>

```

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
00035
00036 }

```

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('CONTROLLERNAMESPACE', '\\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         $controller = $_GET['controller'];
00034         // Workaround magic_quotes() . Ab PHP 5.4 DEPRICATED
00035         if (get_magic_quotes_gpc()) {
00036             $controller = str_replace("\\\\", "\\", $controller);
00037         }
00038
00039         $action = $_GET['action'];
00040
00041
00042         $this->controller = !empty($controller) ? CONTROLLERNAMESPACE . $controller :

```

```

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

```

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     }
00025 }
00026 ?>

```

13 index.php

```
00001 <?php
00002 define('DEBUG', false);
00003
00004 define('CLASS_DIR', 'classes/');
00005
00006 // PHP Autoloader Funktion definieren.
00007 spl_autoload_register('__autoload');
00008
00009 /**
00010  *
00011  * Wir fr den Autoloader von Klassen bentigt
00012  *
00013  * @param      String $class Klassenname
00014  *
00015  */
00016 function __autoload($class) {
00017
00018     // Namespace Backslash in Slash fr das Filesystem umwandeln
00019     $file = str_replace("\\", DIRECTORY_SEPARATOR, $class).".php";
00020
00021     // Klasse einbinden ansonsten Exception werfen.
00022     if (file_exists(CLASS_DIR . $file)) {
00023         require_once CLASS_DIR . $file;
00024     } else {
00025         throw new Exception("Autoloaderfehler: " . $file . " not found.");
00026     }
00027 }
00028
00029 try {
00030
00031     // Session Initialisieren
00032     $session = \Session::getInstance();
00033
00034     // Frontcontroller Initialisieren und Aufrufen
00035     $c = new \FrontController();
00036     $c->run();
00037
00038 } catch (CustomException\UserNotAuthedException $e) {
00039
00040     $view = new \View();
00041     $view->setTemplate('exception');
00042     $view->assign('h1', "Illegalen Aufruf");
00043     $view->assign('exception', $e->getMessage());
00044     $view->display();
00045
00046 } catch (Exception $e) {
00047
00048     $view = new \View();
00049     $view->setTemplate('exception');
00050     $view->assign('h1', "Das h&auml;tte nicht passieren d&uuml;rfe");
00051     $view->assign('exception', $e->getMessage());
00052     $view->display();
00053
00054 }
00055
00056
00057
00058 // Debugging
00059 if (DEBUG) {
```

```

00060     print "<hr>Debug<br>";
00061     print "<pre>";
00062     print_r($_GET);
00063     print_r($_POST);
00064     print_r($_SESSION);
00065     print "</pre>";
00066 }
00067
00068
00069 ?>

```

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 _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             <?php if (\Session::isUserAuthed()) { ?>
00058                 <li><a href="?controller=UserSession&action=Logout">Logout</a></li>
00059             <?php } else { ?>
00060                 <li class="dropdown">
00061                     <a class="dropdown-toggle" href="#" data-toggle="dropdown">Login <strong class=
"caret"></strong></a>
00062                     <div class="dropdown-menu" style="padding: 15px; padding-bottom: 0px;">
00063                         <link href="css/signin.css" rel="stylesheet">
00064                         <form class="form-signin" method="post" action="
?controller=UserSession&action=Login">
00065                             <input type="text" name="user" class="form-control"
placeholder="Benutzername" autofocus>
00066                             <input type="password" name="pass" class="form-control"
placeholder="Passwort">
00067                             <button class="btn btn-primary btn-block" type="submit">
Anmelden</button>
00071                             </form>
00072                         </div>
00073                     </li>
00074                 </li>
00075             <?php } ?>
00076             </ul>
00077         </div><!--/.nav-collapse -->
00078     </div>
00079 </div>
00080 </div>
00081 </div>
00082 <div class="container">
00083     <?php include($this->includetemplate); ?>
00084

```

```

00085     </div> <!-- /container -->
00086
00087
00088     <!-- Bootstrap core JavaScript
00089     ===== -->
00090     <!-- Placed at the end of the document so the pages load faster -->
00091     <script src="js/jquery.js"></script>
00092     <script src="js/bootstrap.min.js"></script>
00093 </body>
00094 </html>

```

16 Model/Survey.php

```

00001 <?php
00002 namespace Model;
00003
00004 /**
00005  *
00006  * Umfrage Datenmodell
00007  *
00008  */
00009 class Survey {
00010
00011     /**
00012      *
00013      * Umfragen holen
00014      *
00015      * @return      Array
00016      *
00017      */
00018     public function getSurveys() {
00019
00020         return array(1 => "Umfrage 1", 2 => "Umfrage 2");
00021
00022     }
00023
00024     /**
00025      *
00026      * Umfragenamen holen
00027      *
00028      * @param      Integer $survey UmfrageId
00029      *
00030      * @return      String Name der Umfrage
00031      *
00032      */
00033     public function getSurveyName($survey) {
00034
00035         return "Name der Umfrage $survey";
00036
00037     }
00038
00039     /**
00040      *
00041      * Umfrage Punkte holen
00042      *
00043      * @param      Integer $survey UmfrageId
00044      *
00045      * @return      Array
00046      *

```

```

00047         */
00048     public function getSurveyItems($survey) {
00049
00050         return (array(10 => "Antwort 1", 11 => "Antwort 2"));
00051     }
00052 }
00053
00054 /**
00055  *
00056  * Anzahl der Abgegeben Umfragewerte
00057  *
00058  * @param      Integer $survey UmfrageId
00059  *
00060  * @return      Integer Anzahl der Ergebnisse zur Umfrage
00061  */
00062 */
00063 public function getSurveyItemCount($survey) {
00064
00065     return 564;
00066 }
00067 }
00068
00069 /**
00070  *
00071  * Umfrage Ergebnisse holen
00072  *
00073  * @param      Integer $survey UmfrageId
00074  *
00075  * @return      Array
00076  */
00077 */
00078 public function getSurveyResult($survey) {
00079
00080     return (array(232 => "Antwort 1", 332 => "Antwort 2"));
00081 }
00082 }
00083
00084 /**
00085  *
00086  * Statistiken holen
00087  *
00088  */
00089 */
00090 public function getStats() {
00091
00092     $stat[survey_cnt] = 17;
00093     $stat[record_cnt] = 343;
00094
00095     return $stat;
00096 }
00097 }
00098 ?>

```

17 Model/User.php

```

00001 <?php
00002 namespace Model;
00003
00004 /**

```

```

00005  *
00006  * Benutzer Datenmodell
00007  *
00008  */
00009 class User {
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         return true;
00025
00026     }
00027
00028     /**
00029     *
00030     * Benutzer holen
00031     *
00032     * @return     Array
00033     */
00034     */
00035     public function getUsers() {
00036
00037         return array("sepp" => date("d.m.Y H:i", time()), "franz" => date("d.m.Y H:i", time()), "
00038         test" => date("d.m.Y H:i", time()));
00039     }
00040 }
00041
00042
00043 ?>

```

18 ORM/Base.php

```

00001 <?php
00002 namespace ORM;
00003
00004 /**
00005  *
00006  * Sehr einfache und minimale
00007  * Objekt Relationen Abbildung
00008  *
00009  * Abstrakte Basis Klasse
00010  *
00011  * Alle ORM Klassen Erben von dieser Klasse
00012  */
00013 */
00014 abstract class Base {
00015
00016     // Datenbank Handler

```



```

00017     protected $dbh;
00018
00019     /**
00020      *
00021      * Konstruktor
00022      *
00023      * Datenbank Verbindung herstellen.
00024      *
00025      */
00026     public function __construct() {
00027
00028         $this->dbh = new \Database();
00029
00030     }
00031
00032     /**
00033      *
00034      * Abstrake Funktion mit dem Insert Statements des Datensatzes
00035      *
00036      * @return      string  Insert Statement
00037      *
00038      */
00039     abstract protected function getInsertStmt();
00040
00041     /**
00042      *
00043      * Getter Methode
00044      *
00045      */
00046     public function __get($key) {
00047         return $this->{$key};
00048     }
00049
00050     /**
00051      *
00052      * Setter Methode
00053      *
00054      */
00055     public function __set($key, $val) {
00056         $this->{$key} = $val;
00057     }
00058
00059     /**
00060      *
00061      * Speichern Methode
00062      *
00063      * Daten werden peristent auf die Datenbank geschrieben
00064      *
00065      */
00066     public function save() {
00067
00068         $stmt = $this->getInsertStmt();
00069         $stmt->execute();
00070
00071     }
00072 }
00073 }
00074 ?>

```

19 ORM/Survey.php

```
00001 <?php
00002 namespace ORM;
00003
00004 /**
00005  *
00006  * Objekt Relationen Abbildung fr die Umfragen
00007  *
00008  */
00009 class Survey extends Base {
00010
00011     // Mapping der Datenbank Attribute
00012     protected $id;
00013     protected $name;
00014
00015     /**
00016      *
00017      * Definieren der Abstrakten Funktion
00018      *
00019      * @return      Object   PDF Statement
00020      *
00021      */
00022     protected function getInsertStmt() {
00023
00024         $stmt = $this->dbh->prepare("INSERT INTO survey (name) VALUES (:name)");
00025         $stmt->bindParam(':name', $this->name);
00026
00027         return $stmt;
00028     }
00029 }
00030 }
00031 ?>
```

20 ORM/User.php

```
00001 <?php
00002 namespace ORM;
00003
00004 /**
00005  *
00006  * Objekt Relationen Abbildung fr die Umfragen
00007  *
00008  */
00009 class User extends Base {
00010
00011     // Mapping der Datenbank Attribute
00012     protected $id;
00013     protected $username;
00014     protected $password;
00015     protected $lastonline;
00016
00017     /**
00018      *
00019      * Definieren der Abstrakten Funktion
00020      *
00021      * @return      Object   PDF Statement
00022      *
00023      */
```

```

00024         protected function getInsertStmt() {
00025
00026             $stmt = $this->dbh->prepare("INSERT INTO user (name, pass) VALUES (:name, :pass)");
00027             $stmt->bindParam(':name', $this->username);
00028             $stmt->bindParam(':pass', $this->password);
00029
00030             return $stmt;
00031         }
00032     }
00033 }
00034 }
00035 ?>

```

21 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 ?>

```

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

```

23 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      * Default Index Get Action
00016      *
00017      */
00018     public function Index.Action() {
00019
00020         $model = new \Model\Survey();
00021
00022         $view = new \View();
00023         $view->setTemplate('admin_stats');
00024         $view->assign('stats', $model->getStats());
00025         $view->display();
00026
00027     }
00028 }
00029 ?>
```

24 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 $this->view['survey_name']; ?></h3>
00008         <div class="panel-body">
00009             <?php
00010                 foreach ($this->view['survey_items'] as $key=>$value) {
00011                     print "<div class='checkbox'>" .
00012                         "<label>" .
00013                         "<input type='checkbox' value='$key'
00014 name='items[]'>$value" .
00015                         "</label>" .
00016                         "</div>";
00017                 }
00018             ?>
00019             <button type="submit" class="btn btn-primary">Absenden</button>
00020         </div>
00021     </form>
00022 </div>
```

25 SurveyItems.php

```
00001 <?php
00002 namespace ORM;
00003
00004 /**
00005  *
00006  * Objekt Relationen Abbildung fr die Umfragen
00007  *
00008  */
00009 class SurveyItem extends Base {
00010
00011     // Mapping der Datenbank Attribute
00012     protected $id;
00013     protected $survey_id;
00014     protected $name;
00015
00016     /**
00017      *
00018      * Definieren der Abstrakten Funktion
00019      *
00020      * @return      Object   PDF Statement
00021      *
00022      */
00023     protected function getInsertStmt() {
00024
00025         $stmt = $this->dbh->prepare("INSERT INTO survey_items (name, survey_id) VALUES (:name,
00026 :survey_id)");
00027         $stmt->bindParam(':name', $this->name);
00028         $stmt->bindParam(':survey_id', $this->survey_id);
00029
00030         return $stmt;
00031     }
00032 }
00033 ?>
```

26 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 $key=>$value) {
00009                 $percent = round($key / $this->view['survey_cnt'] * 100);
00010                 print "<tr><td>$value</td>" .
00011                     "<td>" .
00012                     "<div style='background-color:#428BCA; width:{$percent}%;
padding-left:10px; border-radius: 18px;'>$key ($percent %)</div>" .
00013                     "</td>" .
00014                     "</tr>";
00015             }
00016             ?>
00017
00018         </table>
00019     </div>
```

```

00020         </div>
00021     </div>

```

27 surveys.tpl.php

```

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

```

28 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 ?>

```

29 UserSession.php

```

00001 <?php
00002 namespace Controller;
00003
00004 /**
00005  *
00006  * Login Controller
00007  *
00008  *
00009  */
00010 class UserSession {
00011
00012
00013     /**
00014      *
00015      * Login Formular anzeigen
00016      *
00017      */
00018     public function Login_Action() {
00019

```



```

00020         $view = new \View();
00021         $view->setTemplate('login');
00022         $view->display();
00023
00024
00025     }
00026
00027     /**
00028     *
00029     * Verarbeitung der Logindaten
00030     *
00031     */
00032     public function Login_POST_Action() {
00033
00034         $user = $_GET['user'];
00035         $pass = $_GET['pass'];
00036
00037         $model = new \Model\User();
00038
00039         if ($model->checkCredentials($user, $pass)) {
00040
00041             \Session::authUser();
00042             \Redirect::toController("Index");
00043
00044         } else {
00045
00046             $this->Login_Action();
00047
00048         }
00049
00050     }
00051
00052
00053     /**
00054     *
00055     * Logout Action
00056     *
00057     */
00058     public function Logout_Action() {
00059
00060         \Session::destroy();
00061         \Redirect::toController("Index");
00062
00063     }
00064
00065 }
00066
00067
00068 ?>

```

30 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     } else {
00058
00059         throw new Exception("Template '{$this->template}' nicht gefunden");
00060     }
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  ?>

```