Anhang

1 _layout.tpl.php

2 Config.php

3 Controller/Admin/Base.php

4 Controller/Admin/Survey.php

5 Controller/Admin/User.php

6 Controller/Database.php

7 Controller/Survey.php

8 Database.php

9 FrontController.php

10 index.php

11 InvalidUserPassException.php

12 Model/Base.php

13 Model/Survey.php

14 Model/User.php

15 Redirect.php

16 Session.php

17 Stats.php

18 UserNotAuthedException.php

19 UserSession.php

20 View.php

21 admin_stats.tpl.php

22 admin_surveys.tpl.php

23 admin_user.tpl.php

24 classes/Controller/Index.php

25 database_create_table.tpl.php

26 database_diagramm.tpl.php

27 exception.tpl.php

28 index.php

29 index.tpl.php

30 survey.tpl.php

31 survey_result.tpl.php

32 surveys.tpl.php

1 _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             </li>
00053             <?php } ?>
00054           </ul>
00055           <ul class="nav navbar-nav navbar-right">
00056             <?php if (\Session::isUserAuthed()) { ?>
00057                 <li class="dropdown">
```

```
00058                 <a href="#" class="dropdown-toggle" data-toggle="dropdown">Internals<b class="caret"></b><
      /a>
00059                 <ul class="dropdown-menu">
00060                                     <li class="dropdown-header">Code</li>
00061                         <li><a href="docs/html/index.html" target="_blank">Dokumentation</a></li>
00062                         <li class="divider"></li>
00063                         <li class="dropdown-header">Datenbank</li>
00064                         <li><a href="?controller=Database&action=Diagramm">UML Diagramm</a></li>
00065                         <li><a href="?controller=Database&action=CreateTable">Create Table SQL</a></li>
00066               </ul>
00067             </li>
00068                                 <li><a href="?controller=UserSession&action=Logout">Logout</a></li>
00069             <?php } else { ?>
00070                             <li class="dropdown">
00071                               <a class="dropdown-toggle" href="#" data-toggle="dropdown">Login <strong class=
      "caret"></strong></a>
00072                               <div class="dropdown-menu" style="padding: 15px; padding-bottom: 0px;">
00073
00074                                   <link href="css/signin.css" rel="stylesheet">
00075
00076                                                       <form class="form-signin" method="post" action="
      ?controller=UserSession&action=Login">
00077                                                       <input type="text" name="user" class="form-control"
      placeholder="Benutzername" autofocus>
00078                                                       <input type="password" name="pass" class="form-control"
      placeholder="Passwort">
00079                                                       <button class="btn btn-primary btn-block" type="submit">
      Anmelden</button>
00080                                                       </form>
00081
00082                               </div>
00083                             </li>
00084
00085             <?php } ?>
00086           </ul>
00087
00088         </div><!--/.nav-collapse -->
00089       </div>
00090     </div>
00091
00092     <div class="container">
00093         <?php include($this->includetemplate); ?>
00094     </div> <!-- /container -->
00095
00096
00097     <!-- Bootstrap core JavaScript
00098     ================================================== -->
00099     <!-- Placed at the end of the document so the pages load faster -->
00100     <script src="js/jquery.js"></script>
00101     <script src="js/bootstrap.min.js"></script>
00102   </body>
00103 </html>
```

2 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 }
```

3 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         protected $model;
00014         protected $view;
00015
00016         /**
00017          *
00018          * Im Konstruktur generell ueberpruefen ob
00019          * sich der User eingeloggt hat.
00020          *
00021          */
00022         public function __construct() {
00023
00024                 \Session::isUserAuthedCheck();
00025
00026                 $this->view = new \View();
00027
00028         }
00029
00030 }
00031 ?>
```

4 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
00087                 } else {
00088
00089                         throw new \Exception("Illegaler Aufruf");
00090
00091                 }
00092
00093         }
00094 }
00095 ?>
```

5 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 ?>
```

6 Controller/Database.php

```
00001 <?php
00002 namespace Controller;
00003
00004 /**
```

```
00005  *
00006  * Datenbankinformationen Controller
00007  *
00008  *
00009  *
00010  */
00011 class Database {
00012
00013         private $view;
00014
00015         /**
00016          *
00017          * Kontruktor
00018          *
00019          * View Initalisieren
00020          *
00021          */
00022         public function __construct() {
00023
00024                 \Session::isUserAuthedCheck();
00025
00026                 $this->view = new \View();
00027
00028         }
00029
00030         /**
00031          *
00032          * UML Diagramm anzeigen
00033          *
00034          */
00035         public function Diagramm_Action() {
00036
00037                 $this->view->setTemplate('database_diagramm');
00038                 $this->view->display();
00039
00040         }
00041
00042         /**
00043          *
00044          * SQL Statements anzeigen
00045          *
00046          */
00047         public function CreateTable_Action() {
00048
00049                 $this->view->setTemplate('database_create_table');
00050                 $this->view->display();
00051
00052         }
00053 }
00054 ?>
```

7 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                 $this->model = new \Model\Survey();
00025                 if (isset($_GET['survey'])) $this->survey = intval($_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                         $this->view->display();
00051
00052
00053                 }
00054
00055         }
00056
00057         /**
00058          *
00059          * Umfrage Ergebnisse speichern
00060          *
00061          *
00062          *
00063          */
00064         public function Save_POST_Action() {
00065
00066                 if ($this->survey > 0 && isset($_POST['answer'])) {
00067
00068                         $answerArr = $_POST['answer'];
```

```
00069
00070                         $this->model->saveNewAnswers($answerArr);
00071
00072                         $this->showSurveyResult();
00073
00074                 } else {
00075
00076                         // Wird nichts ausgewaelt einfach das Ergebnis anzeigen.
00077                         $this->showSurveyResult();
00078
00079                 }
00080
00081
00082         }
00083
00084         /**
00085          *
00086          * Umfrage Ergebnisse anzeigen
00087          *
00088          * Direkt nur fuer Adminstratoren moeglich
00089          *
00090          */
00091         public function Show_Action() {
00092
00093                 if ($this->survey == 0) throw new \Exception("Illegaler Aufruf");
00094
00095                 \Session::isUserAuthedCheck();
00096
00097                 $this->showSurveyResult();
00098
00099         }
00100
00101         /**
00102          *
00103          * Auswertung anzeigen
00104          *
00105          */
00106         private function showSurveyResult() {
00107
00108                 $this->view->setTemplate('survey_result');
00109                 $this->view->assign('survey_name', $this->model->getSurveyName($this->survey));
00110                 $this->view->assign('survey_cnt', $this->model->getSurveyItemCount($this->survey));
00111                 $this->view->assign('survey_result', $this->model->getSurveyResult($this->survey));
00112                 $this->view->display();
00113
00114         }
00115
00116 }
00117
00118
00119 ?>
```

8 Database.php

```
00001 <?php
00002 /**
00003  *
00004  * Database Klasse erweitert PHP PDO
00005  *
```

```
00006  * $db = new Database(); // fuer eine eigene Datenbankverbindung.
00007  * $db = Database::getInstance(); // Empfohlene Methode
00008  *
00009  */
00010 class Database extends PDO {
00011
00012         private $dsn;
00013         private $user;
00014         private $pass;
00015
00016         static private $instance = null;
00017
00018         static public function getInstance() {
00019                 if (null === self::$instance) {
00020                         self::$instance = new self;
00021                 }
00022
00023                 return self::$instance;
00024         }
00025
00026         /**
00027          *
00028          * Konstruktor wird nur einmalig aufgerufen.
00029          *
00030          * Konfiguration laden und der Elternklasse uebergeben.
00031          *
00032          * Die Funktion muss public sein da die PDO Elternklasse auch einen public
00033          * Konstruktur liefert.
00034          *
00035          */
00036         public function __construct() {
00037
00038                 $conf = \Config::getInstance();
00039
00040                 $this->dsn = $conf->database_type . ":host=" . $conf->database_host . ";port=" . $conf->
      database_port . ";dbname=" . $conf->database_name;
00041
00042                 $this->user = $conf->database_user;
00043                 $this->pass = $conf->database_pass;
00044
00045                 parent::__construct($this->dsn, $this->user, $this->pass);
00046
00047
00048         if ($conf->database_verbose == 1) $this->beMoreVerbose();
00049
00050
00051         }
00052
00053         /**
00054          *
00055          * Error Modus erhoehen
00056          *
00057          */
00058         private function beMoreVerbose() {
00059
00060                 $this->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
00061         $this->setAttribute(PDO::ATTR_EMULATE_PREPARES, false);
00062
00063         }
00064
00065
00066 }
```

9 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  * Ergebsniss: 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) ? CONTROLLERNAMESPACE . $controller :
      CONTROLLERNAMESPACE . "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 ?>
```

10 index.php

```
00001 <?php
00002 define('CLASS_DIR', 'classes/');
00003 date_default_timezone_set("Europe/Berlin");
00004
00005 /**
00006  *
00007  * PHP Klassen Autoloader definieren
00008  *
00009  * Andere Variante waere mit spl_autoload_register()
00010  * eine oder mehrere Funktionen definieren
00011  *
00012  * @param       String  $class  Klassenname
00013  *
00014  */
00015 function __autoload($class) {
00016
00017           // Namespace Backslashe in Slashe fr das Filesystem umwandeln
00018           $file = str_replace("\\", DIRECTORY_SEPARATOR, $class).".php";
00019
00020           // Klasse einbinden ansonsten Exception werfen.
00021           if (file_exists(CLASS_DIR . $file)) {
00022                   require_once CLASS_DIR . $file;
00023           } else {
00024                   throw new Exception("Autoloaderfehler: " . $file . " not found.");
00025           }
00026 }
00027
00028 try {
00029
00030           // Session Initalisieren
00031           $session = \Session::getInstance();
00032
00033           // Frontcontroller Initalsieren und Aufrufen
00034           $c = new \FrontController();
00035           $c->run();
00036
00037
00038 } catch (CustomException\UserNotAuthedException $e) {
00039
00040           displayException($e->getMessage(), "Illegaler Aufruf");
00041
00042
```

```
00043 } catch (CustomException\InvalidUserPassException $e) {
00044
00045         displayException("Benutzername und/oder Passwort falsch", "Loginfehler");
00046
00047
00048 } catch (Exception $e)  {
00049
00050         displayException($e->getMessage(), "Das h&auml;tte nicht passieren d&uuml;rfen");
00051
00052 }
00053
00054 /**
00055  *
00056  * Exception Anzeigen
00057  *
00058  * @param String        $e      Exceptiontext
00059  * @param String        $h1     Ueberschrift
00060  *
00061  */
00062 function displayException($e, $h1) {
00063
00064         $view = new \View();
00065         $view->setTemplate('exception');
00066         $view->assign('h1', $h1);
00067         $view->assign('exception', $e);
00068         $view->display();
00069
00070 }
00071
00072 $conf = \Config::getInstance();
00073
00074 // Debugging
00075 if ($conf->application_debugging == 1) {
00076         print "<br /><br />";
00077         print "<pre>";
00078         print '$_GET ';
00079         print_r($_GET);
00080         print '$_POST ';
00081         print_r($_POST);
00082         print '$_SESSION ';
00083         print_r($_SESSION);
00084         print '$_SERVER ';
00085         print_r($_SERVER);
00086         print "</pre>";
00087 }
00088
00089
00090 ?>
```

11 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 ?>
```

12 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 = \Database::getInstance();
00024
00025         }
00026
00027
00028
00029 }
00030
00031
00032 ?>
```

13 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          * Umfragename 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
      Name");
00057                 $stmt->bindParam(':id', $survey);
00058                 $stmt->execute();
00059
00060                 return $stmt->fetchAll();
00061
00062
00063
00064         }
00065
00066         /**
00067          *
00068          * Anzahl der Abgegeben Umfragewerte
00069          *
00070          * @param       Integer $survey UmfrageId
00071          *
00072          * @return      Integer Anzahl der Ergebnisse zur Umfrage
00073          *
00074          */
00075         public function getSurveyItemCount($survey) {
```

```
00076
00077                 $stmt = $this->dbh->prepare("SELECT COUNT(*) AS cnt FROM SurveyAnswer WHERE SurveyItemID IN
      (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 loeschen
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          *
00161          * neue Umfrage hinzufuegen
00162          *
00163          * @param       String  $name   Name der Umfrage
00164          * @param       Array   $anserArr       Array mit den moeglichen Antworten
00165          *
00166          */
00167         public function addSurvey($name, $answerArr) {
00168
00169                 if (empty($name)) return;
00170
00171                 $this->dbh->beginTransaction();
00172
00173                 $stmt = $this->dbh->prepare("INSERT INTO Survey SET Name = :name");
00174                 $stmt->bindParam(':name', $name);
00175
00176                 $stmt->execute();
00177
00178                 $surveyID = $this->dbh->lastInsertId();
00179
00180                 $stmt = $this->dbh->prepare("INSERT INTO SurveyItems SET Name = :name, SurveyID = :id");
00181
00182                 $stmt->bindParam(':id', $surveyID);
00183
00184                 foreach ($answerArr as $answer) {
00185
00186                         if (!empty($answer)) {
00187                                 $stmt->bindParam(':name', $answer);
00188                                 $stmt->execute();
00189                         }
00190
00191                 }
00192
00193                 $this->dbh->commit();
00194
00195         }
00196
00197         /**
```

```
00198          *
00199          * Anzahl der Tupel zurueckgeben
00200          *
00201          * @param       String  $table  Tabellenname
00202          *
00203          * @return      Integer Anzahl der Tupel
00204          *
00205          */
00206         private function getTableCount($table) {
00207
00208
00209                 switch ($table) {
00210
00211                         case "User":
00212                         case "Survey":
00213                         case "SurveyItems":
00214                         case "SurveyAnswer":
00215                                 $stmt = $this->dbh->query("SELECT COUNT(*) AS cnt FROM $table");
00216                                 return $stmt->fetchObject()->cnt;
00217                                 break;
00218                         default:
00219                                 throw new \Exception("Illegaler Tabellenname $table");
00220
00221                 }
00222
00223         }
00224
00225 }
00226
00227
00228
00229 ?>
```

14 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 = $stmt->fetchObject();
00031
00032                 if (!is_object($res)) return false;
00033
00034                 if ($res->Passwort == $pass) {
00035
00036                         $this->updateLastLogIn($user);
00037                         return true;
00038
00039                 } else {
00040
00041                         return false;
00042
00043                 }
00044
00045         }
00046
00047         /**
00048          *
00049          * Benutzer holen
00050          *
00051          * @return      Array
00052          *
00053          */
00054         public function getUsers() {
00055
00056                 $stmt = $this->dbh->query("SELECT ID, Name, LastLogIn FROM User");
00057                 return  $stmt->fetchAll();
00058
00059         }
00060
00061         /**
00062          *
00063          * Benutzer lschen
00064          *
00065          * @param       Integer $id     BenutzerID
00066          *
00067          */
00068         public function deleteUser($id) {
00069
00070                 $stmt = $this->dbh->prepare("DELETE FROM User WHERE ID = :id");
00071                 $stmt->bindParam(':id', $id);
00072
00073                 $stmt->execute();
00074
00075         }
00076
00077         /**
00078          *
00079          * Benutzer hinzufgen
00080          *
00081          * @param       String  $name   Benutzername
00082          * @param       String  $pass   Passwort
00083          *
00084          */
00085         public function addUser($name, $pass) {
00086
```

```
00087                 $stmt = $this->dbh->prepare("INSERT INTO User SET Name = :name, Passwort = :pass");
00088                 $stmt->bindParam(':name', $name);
00089                 $stmt->bindParam(':pass', $pass);
00090 
00091                 $stmt->execute();
00092 
00093         }
00094 
00095 
00096         /**
00097          *
00098          * Zeitstempel des letzten Logins vom User Updaten
00099          *
00100          * @param       String  $user   Benutzername
00101          *
00102          */
00103         private function updateLastLogIn($user) {
00104 
00105                 $stmt = $this->dbh->prepare("UPDATE User SET LastLogIn = :lastlogin WHERE Name = :name");
00106                 $stmt->bindParam(':name', $user);
00107                 $stmt->bindValue(':lastlogin', date("Y-m-d H:i:s"));
00108 
00109                 $stmt->execute();
00110 
00111         }
00112 
00113 }
00114 
00115 
00116 ?>
```

15 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 ?>
```

16 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 Authentifizeren
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 (array_key_exists('__AUTH', $_SESSION) && $_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("
      Sie haben keine Berechtigung fr diese Seite");
00106 
```

```
00107         }
00108
00109         /**
00110          *
00111          * Session loeschen
00112          *
00113          */
00114         public static function destroy() {
00115
00116                 session_destroy();
00117
00118         }
00119
00120 }
```

17 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 ?>
```

18 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 ?>
```

19 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 ?>
```

20 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         // Anzuzeigenedes 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  $template       Templatename
00038          *
00039          */
00040         public function setTemplate($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 Exitiert
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 Templatenamen 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 ?>
```

21 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
```

22 admin_surveys.tpl.php

```
00001                 <h1>Umfragen Adminstration</h1>
00002                 <table class="table table-hover">
00003                         <tr><th>ID</th><th>Umfrage</th><th></th></tr>
00004                         <?php foreach ($this->view['surveys'] as $arr) { ?>
00005                                 <tr><td><?php print $arr['ID']; ?></td>
00006                                         <td><a href="?controller=Survey&action=Show&survey=<?php print
 $arr['ID']; ?>"><?php print $arr['Name']; ?></a></td>
00007                                         <td class="text-right"><a href="?controller=Admin\S
urvey&action=Delete&survey=<?php print $arr['ID']; ?>" class='btn btn-danger'>L&ouml;schen</a></td></tr>
00008                         <?php } ?>
00009                 </table>
00010                 <a data-toggle="modal" href="#addSurveyModal" class="btn btn-success">Hinzuf&uuml;gen</a>
00011
00012           <!-- Modal -->
00013           <form class="form-survey" method="post" action="?controller=Admin\Survey&action=Add">
00014
00015           <div class="modal" id="addSurveyModal">
00016             <div class="modal-dialog">
00017               <div class="modal-content">
00018                 <div class="modal-header">
00019                   <button type="button" class="close" data-dismiss="modal" aria-hidden="true">&times;</
button>
00020                   <h4 class="modal-title">Neue Umfrage hinzuf&uuml;gen</h4>
00021                 </div>
00022                 <div class="modal-body">
00023                  <div class="form-group">
00024                           <label for="name">Umfragename</label>
00025                           <input type="text" class="form-control" id="name" name="name" placeholder="
Umfragename eingeben">
00026                         </div>
00027
00028                  <div class="form-group">
00029                           <label for="answer">Antwort</label>
00030                           <input type="text" class="form-control" id="answer" name="answer[]" placeholder
="Antwort eingeben">
00031                         </div>
00032
00033                 </div>
00034
00035                         <div style="padding: 20px;">
00036                                 <a class="btn btn-default" onClick="newAnswer()">Weitere Antwort hinzuf&
uumlgen</a>
00037                         </div>
00038
00039                 <div class="modal-footer">
00040                   <a href="#" class="btn" data-dismiss="modal">Verwerfen</a>
00041                   <button class="btn btn-primary" type="submit">Speichern</button>
00042                 </div>
00043               </div><!-- /.modal-content -->
```

```
00044             </div><!-- /.modal-dialog -->
00045           </div><!-- /.modal -->
00046           </form>
00047
00048           <script>
00049                   function newAnswer() {
00050
00051                                   $('.modal-body').append('<div class="form-group"><input type="text"
       class="form-control" name="answer[]" placeholder="Antwort eingeben""></div>');
00052
00053                   }
00054           </script>
```

23 admin_user.tpl.php

```
00001                   <h1>Benutzer Adminstration</h1>
00002                   <table class="table table-hover">
00003                           <tr><th>Benutername</th><th>Letzer Login</th><th></th></tr>
00004                           <?php foreach ($this->view['users'] as $arr) { ?>
00005                                   <tr>
00006                                           <td><?php print $arr['Name'] ?></td>
00007                                           <td><?php print $arr['LastLogIn'] ?></td>
00008                                           <td class="text-right"><a href="?controller=Admin\U
       ser&action=Delete&user=<?php print $arr['ID'] ?>" class="btn btn-danger">L&ouml;schen</a></td>
00009                                   </tr>
00010                           <?php } ?>
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>
```

24 classes/Controller/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 ?>
```

25 database_create_table.tpl.php

```
00001         <pre><?php include("sql/database.sql");?></pre>
00002
```

26 database_diagramm.tpl.php

```
00001         <img src="sql/database.png" alt="Datenbank UML Diagramm" class="img-thumbnail">
```

27 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>
```

28 index.php

```
00001 <?php
00002 define('CLASS_DIR', 'classes/');
00003
00004 /**
00005  *
00006  * PHP Klassen Autoloader definieren
00007  *
00008  * Andere Variante waere mit spl_autoload_register()
00009  * eine oder mehrere Funktionen definieren
00010  *
00011  * @param       String  $class  Klassenname
00012  *
00013  */
00014 function __autoload($class) {
00015
00016         // Namespace Backslashe 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 Initalisieren
00030         $session = \Session::getInstance();
00031
00032         // Frontcontroller Initalsieren und Aufrufen
00033         $c = new \FrontController();
00034         $c->run();
00035
00036
00037 } catch (CustomException\UserNotAuthedException $e) {
00038
00039         displayException($e->getMessage(), "Illegaler Aufruf");
00040
00041
00042 } catch (CustomException\InvalidUserPassException $e) {
00043
00044         displayException("Benutzername und/oder Passwort falsch", "Loginfehler");
00045
00046
00047 } catch (Exception $e)  {
00048
00049         displayException($e->getMessage(), "Das h&auml;tte nicht passieren d&uuml;rfen");
00050
00051 }
00052
00053 /**
00054  *
00055  * Exception Anzeigen
00056  *
00057  * @param String        $e      Exceptiontext
00058  * @param String        $h1     Ueberschrift
00059  *
```

```
00060  */
00061 function displayException($e, $h1) {
00062
00063         $view = new \View();
00064         $view->setTemplate('exception');
00065         $view->assign('h1', $h1);
00066         $view->assign('exception', $e);
00067         $view->display();
00068
00069 }
00070
00071 $conf = \Config::getInstance();
00072
00073 // Debugging
00074 if ($conf->application_debugging == 1) {
00075         print "<br /><br />";
00076         print "<pre>";
00077         print '$_GET ';
00078         print_r($_GET);
00079         print '$_POST ';
00080         print_r($_POST);
00081         print '$_SESSION ';
00082         print_r($_SESSION);
00083         print '$_SERVER ';
00084         print_r($_SERVER);
00085         print "</pre>";
00086 }
00087
00088
00089 ?>
```

29 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
```

30 survey.tpl.php

```
00001
00002       <div class="jumbotron">
00003                 <form role="form" method="post" action="?controller=Survey&action=Save&survey=<?php print
      $this->view['survey_id']; ?>">
00004                 <div class="panel panel-default">
00005                         <div class="panel-heading">
00006                         <h3 class="panel-title"><?php print $this->view['survey_name']; ?></h3>
00007                         </div>
00008                         <div class="panel-body">
```

```
00009                                 <?php foreach ($this->view['survey_items'] as $arr) { ?>
00010                                         <div class='checkbox'>
00011                                                 <label>
00012                                                         <input type="checkbox" value="<?php print
      $arr['ID']; ?>" name="answer[]"><?php print $arr['Name']; ?>
00013                                                 </label>
00014                                         </div>
00015                                 <?php } ?>
00016                                 <button type="submit" class="btn btn-primary">Absenden</button>
00017                         </div>
00018                 </div>
00019                 </form>
00020       </div>
```

31 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                                 <tr>
00017                                         <td><?php print $arr['Name']; ?></td>
00018                                         <td>
00019                                 <?php if ($arr['cnt'] > 0) { ?>
00020                                                 <div style="background-color:#428BCA; width:<?php print
      $percent; ?>%; padding-left:10px; border-radius: 18px;"><?php print $arr['cnt']; ?> <small>(<?php print
     $percent; ?>%)</small></div>
00021                                 <?php } else { ?>
00022                                                 <div>bisher keine Stimmen</div>
00023                                 <?php } ?>
00024                                         </td>
00025                                 </tr>
00026
00027                         <?php } ?>
00028
00029                         </table>
00030                         </div>
00031                 </div>
00032       </div>
```

32 surveys.tpl.php

```
00001
00002       <div class="jumbotron">
00003         <h1>Bitte Umfrage ausw&auml;hlen</h1>
```

```
00004                 <ul class="list-group">
00005                         <?php foreach ($this->view['surveys'] as $arr) { ?>
00006                                 <li class="list-group-item"><a href="?controller=Survey&survey=<?php print
      $arr['ID'] ?>"><?php print $arr['Name'] ?></a></li>
00007                         <?php } ?>
00008                 </ul>
00009       </div>
```