



# YWEE

Personal Tutoring Service  
Gruppe 2

Alexander Strobl

21. Juni 2014



# Agenda

- 1 Einleitung
- 2 SRA
- 3 SAD
- 4 Implementierung



# Agenda



- 1 Einleitung
  - Die Aufgabe
- 2 SRA
- 3 SAD
- 4 Implementierung



## Aufgabengebiete während des Projektverlaufs

- Teamleitung Content & Layout, bestehend aus:
  - Matthias Birnthaler
  - Christian Dauerer
  - Matthias Götz
  - Nguyen Hoan - später ausgetreten
  - Stefan Holz - später ausgetreten
  - Sven Liebl - später ausgetreten
  - Tobias Schwindl - später in Backend gewechselt
- Fertigstellung und Abgabe des SRA
- Fertigstellung und Abgabe des SAD
- Implementierung verschiedener Inhalte



# Agenda

## 1 Einleitung

## 2 SRA

- Die Vorgabe
- Die Formulierung

## 3 SAD

## 4 Implementierung



## Leitung des Teams Content & Layout

- Verteilung von Aufgaben
- Bildung von Unterteams
- Fertigstellung und Abgabe des SRA-Dokuments
- Kontrolle und Bewertung der Ergebnisse

## Mitarbeit am SRA

- Punkt F4: Lehrer/Mentor mit Christian Dauerer
- Punkt S: Systemanforderungen



# Gruppeneinteilung

Folgende Gruppeneinteilung wurden im Team Content & Layout vorgenommen:

- 1 Matthias Götz, Sven Liebl
- 2 Matthias Birnthaler, Tobias Schwindl
- 3 Nguyen Hoan, Stefan Holz
- 4 Christian Dauerer, Alexander Strobl

Die Aufteilung in 2er Teams soll das Teamwork verbessern und durch die gegenseitige Kontrolle der Teammitglieder die Ergebnisse verbessern und gleichzeitig weniger Fehler hervorrufen.



# Text für die SRA

Zusammen mit Christian Dauerer wurden folgende Texte formuliert und in die SRA eingebaut:

- F41: Profil einstellen
- F42: Lehrmaterialien online stellen und verteilen
- F43: Beurteilungen schreiben
- F44: Schwarzes Brett
- S: Systemanforderungen wurden aus Vorlage übernommen





# Agenda

1 Einleitung

2 SRA

3 SAD

- Analysephase
- Die Formulierung

4 Implementierung



## Leitung des Teams Content & Layout

- Verteilung von Aufgaben
- Fertigstellung des SAD-Dokuments
- Kontrolle und Bewertung der Ergebnisse

## Mitarbeit am SAD

- Kontextdiagramm: Tutor
- Punkt F16: About us (Impressum)
- Punkt F17: Preismodell und Zahlungsinfos
- Punkt F19: Support und Kontaktdaten
- Abschnitt 3.4: Lehrer/Mentor zusammen mit Christian Dauerer



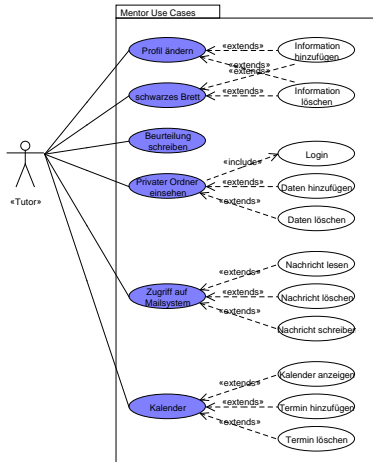
Um einen besseren Überblick zu Erhalten wurden die folgenden Fragen gestellt:

- 1 Welche Grundanforderungen sind nötig?
- 2 Welcher Webserver steht zur Verfügung?
- 3 Welche Möglichkeiten gibt es zur Umsetzung?
- 4 Welche Sprachen werden benötigt?

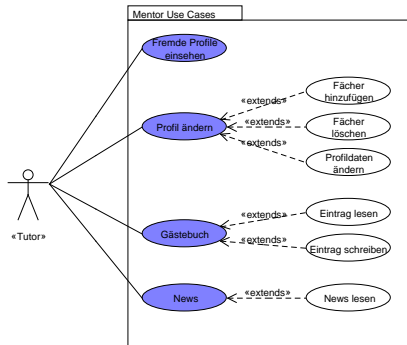


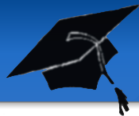
# Kontextdiagramm: Tutor

## Ursprüngliche Version



## Finale Version





## Punkt F16: About us(Impressum)

Der Besucher der Webseite sollte die Möglichkeit haben, Informationen über Die Tutoren AG zu bekommen. Außerdem ist nach §5 Telemediengesetz (TMG) in Deutschland eine Impressumspflicht vorhanden.

Das Impressum enthält:

- Firmenname
- Geschäftsführer (Inhaltlich Verantwortlicher, Vertretungsberechtigter)
- Anschrift der Die Tutoren AG (Straße, PLZ, Ort)
- Kontaktdaten (eMail, Telefon)



## Punkt F17: Zahlungsinformationen

Der Besucher der Webseite sollte die Möglichkeit haben, sich vor der Anmeldung über verschiedenen Preismodelle und Arten der Zahlungsabwicklung zu informieren.

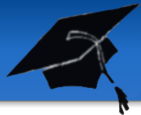
Dafür wurde eine eigene Seite mit statischen Inhalten geplant, welche den Besucher darüber informieren sollte.



## Punkt F19: Support und Kontaktdaten

Der Besucher der Webseite sollte die Möglichkeit haben, sich bei Fragen und Problemen an den Support der Die Tutoren AG zu wenden.

Dafür wurde die Seite "Kontakt" geplant. Diese soll statischen HTML Content beinhalten und ist über die Navigationsleiste zu erreichen.



# Analyse - Schwarzes Brett

Eine geplante Funktion für den Tutor war ein schwarzes Brett, welches der Tutor für die Veröffentlichung von Informationen benutzen kann, wie z.B. eine Krankheitsmeldung oder neue Fächer usw.

Dafür wurde eingeplant:

- Eine DB-Tabelle für die Einträge
- Abfrage der Daten über SQL
- Weitergabe per JSON
- Anzeige über JavaScript





# Analyse - Nachrichtensystem

Eine geplante Funktion für den Tutor (wie für alle Benutzer) war ein systemweites Nachrichtensystem, damit Benutzer sich gegenseitig Nachrichten schicken können.

Dafür wurde eingeplant:

- Eine DB-Tabelle für die Einträge
- Abfragen der Daten über SQL
- Weitergabe per JSON
- Anzeige über jQuery



# Agenda

## 1 Einleitung

## 2 SRA

## 3 SAD

## 4 Implementierung

- Umgesetzt
  - Gästebuch
  - Nachrichten
  - Statische Inhalte
- Nicht umgesetzt
- Demonstration



Implementierung der Inhalte für:

- 1 Gästebuch
- 2 Top-Gästebuch
- 3 Nachrichten
- 4 Adminverwaltung
  - Nachrichteneinträge löschen
- 5 Statische Inhalte
  - Unternehmen
  - Kontakt
  - AGB
  - Impressum



# Gästebucheinträge - Teil 1

## gaestebuch.php

```
1 <form name="addguestbookentry" method="post" action="/scripts/WriteGuestbook.php"
  id="gbform">
2   <input type="hidden" name="sprache" value="<?php echo $_SESSION['sprache'];
    ?>">
3   <input type="hidden" name="username" value="<?php echo $_SESSION['user']; ?>
    ">
4   <div>
5     <label for="nachrichtentext"></label>
6     <p>
7       <textarea name="eintrag" cols="33" rows="5" maxlength="300" id="
        nachrichtentext" placeholder="Schreiben Sie hier ihren
        Gaestebucheintrag..." required></textarea>
8     </p>
9   </div>
10  <div class="line submit">
11    <input type="submit" value="Absenden">
12  </div>
13 </form>
```

## gaestebuch.html

```
1 <h1>Gästebuch</h1>
2 <p id="data"></p>
3 <script type="text/javascript" src="/js/guestbook.js"></script>
```



# Gästebucheinträge - Teil 2

## guestbook.js

```
1 $(document).ready(function myFunction() {
2     $.ajaxSetup({cache: false});
3     var url = "/scripts/ReadGuestbook.php";
4
5     $.ajax(    {
6         type: 'post',
7         url: url,
8         dataType: 'json',
9         cache: false,
10        success: function(data)
11        {
12            var SingleEntry = new Array(data.length);
13
14            for (var i = 0; i < data.length; i++)
15            {
16                SingleEntry.push("<div class='basic-wrapper'><div class='
17                    GBauthor'>" + data[i]['benutzername'] + "</div><div class='
18                    GBmessage'>" + data[i]['eintrag'] + "</div></div>");
19            }
20
21            var AllEntries = SingleEntry.join('');
22            document.getElementById("data").innerHTML = AllEntries;
23        }
24    });
25 }
```



# Nachrichten - Teil 1

## ReadNewsDelete.php

```
1 <?php
2 include_once($_SERVER["DOCUMENT_ROOT"] . "/test_02/scripts/ConToDB.php");
3     // Inkludiert die Funktion zur Anmeldung an der DB
4     // Baue Verbindung auf
5     try
6     {
7         $dbConnection = ConnectToDB();
8     } catch (Exception $e)
9     {
10        die("keine Verbindung moeglich: " . $e->getMessage());
11    }
12    $query = $dbConnection->prepare("SELECT * FROM news where 1 order by id desc");
13    ;
14    if($query->execute() )
15        $result = $query->fetchAll(PDO::FETCH_ASSOC);
16    $dbConnection = null;
17    ?>
```



# Nachrichten - Teil 2

## admin.php

```
1  if(isset($_GET['deleteNews']) and $_GET['deleteNews'] == true) {
2      if(isset($_GET['fired']) and $_GET['fired'] == true and isset($_POST["NewsID"]
3          )){
4          $deleteNews = $_POST["NewsID"];
5          include_once($_SERVER["DOCUMENT_ROOT"] . "/test_02/scripts/
6              DeleteNewsfromAdmin.php");
7      }
8      include_once($_SERVER["DOCUMENT_ROOT"] . "/test_02/scripts/ReadNewsDelete.php"
9          );
10     $newsresults = $result;
11     $news_print = "";
12     foreach($newsresults as $eintrag) {
13         $news_print .= "<tr> ";
14         $news_print .= "<th><div class='NewsDelEntry'> " . $eintrag["nachricht"] . "
15             </div></th>";
16         $news_print .= "<th><div class='NewsDelTopic'> " . $eintrag["betreff"] . "</
17             div></th>";
18         $news_print .= "<th><div class='NewsDelAuthor'> " . $eintrag["benutzername"]
19             . "</div></th> ";
20         $news_print .= '<th><div class="NewsCheckbox"><input type="checkbox" name="
21             NewsID[]" value="' . $eintrag["id"] . '"></div></th>';
22         $news_print .= "</tr>";
23     }
24     include_once($_SERVER["DOCUMENT_ROOT"] . "/test_02/de/content/ShowNewstoDelete
25         .html");
26 }
```



# Statische Inhalte - Beispiel: Kontakt

## kontakt.html

```
1 <h1>Kontakt:</h1>
2 <table>
3 <tr>
4 <td><p>Telefon:</p></td>
5 <td><p>0123456789</p></td>
6 </tr>
7 <tr>
8 <td><p>E-Mail:</p></td>
9 <td><p><a href="mailto:support@tutoren-ag.de">An den Support wenden</a></p></td>
10 </tr>
11 </table>
12 Unser Support ist taeglich 24h zu erreichen.<br>
```





# Nicht umgesetzt



Aus Zeitgründen konnten folgende Features nicht umgesetzt werden:

- 1 Punkt F17: Preismodell und Zahlungsinfos  
Stattdessen wurde eine Spendenseite eingerichtet
- 2 Punkt F42: Eigenes Schwarzes Brett für Tutoren
- 3 Punkt F24/F34/F43: Zugriff aufs Nachrichtensystem



Es folgt eine Demonstration ...



**Vielen Dank für Ihre Aufmerksamkeit**