Project:	Personal Tutoring Service (PTS)
Team No.:	2
Class:	Web Engineering SS2014
Module:	System Architecture and Design (SAD)
Deliverable:	SAD Document

Version: **[1.0]** **Date:** **[**03.05.2014**]**

Beitragende:

Sven Liebl
Matthias Goetz
Christian Dauerer
Maxmilian Schröter
Stefan Holz
Viet Nguyen
Daniel Tatzel
Nils Weiss
Florian Laufenböck
Alexander Strobl
Matthias Birnthaler
Tobias Schwindl

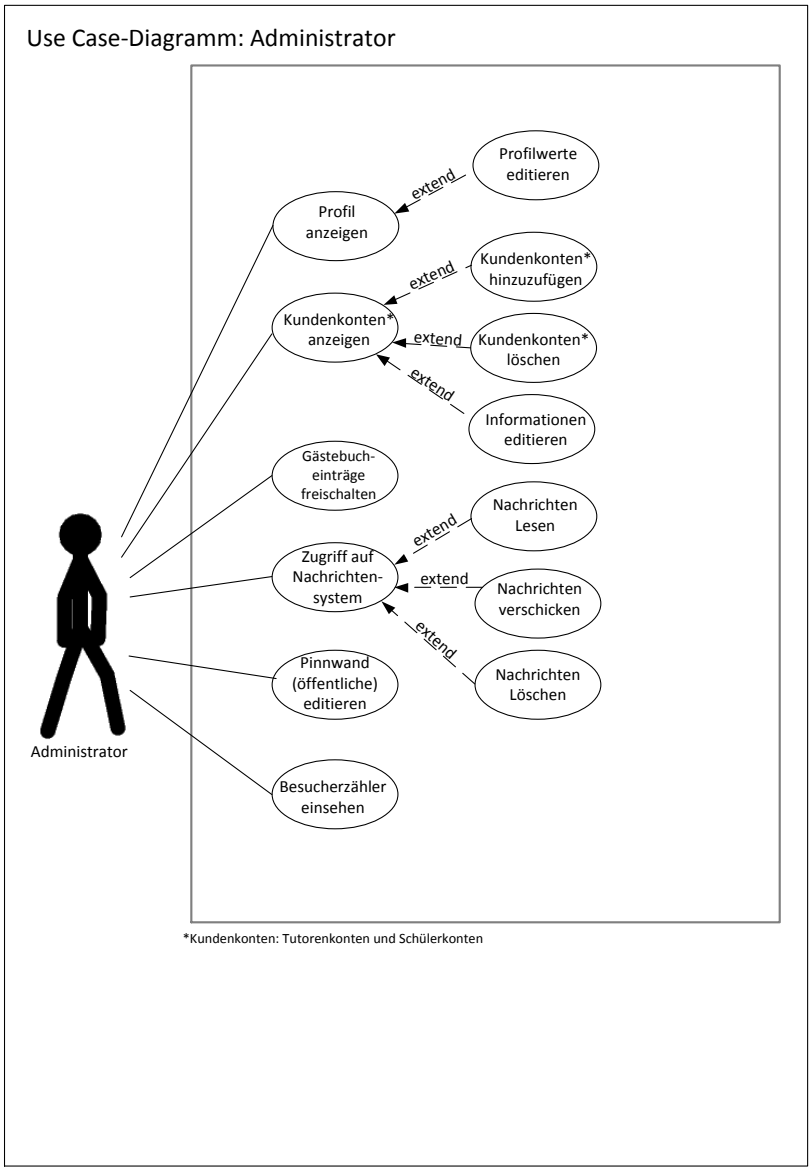
VersionsNr	Datum	Auslöser	Veränderungsgrad	Beschreibung
1.0	23.04.2014	I-wer	Erster Entwurf	First Draft

Tabelle 1: Überarbeitungshistorie

Inhaltsverzeichnis

1 Software Architecture

1.1 Kontextdiagramm 1 - Administrator



1.2 Kontextdiagramm 2 - Tutor

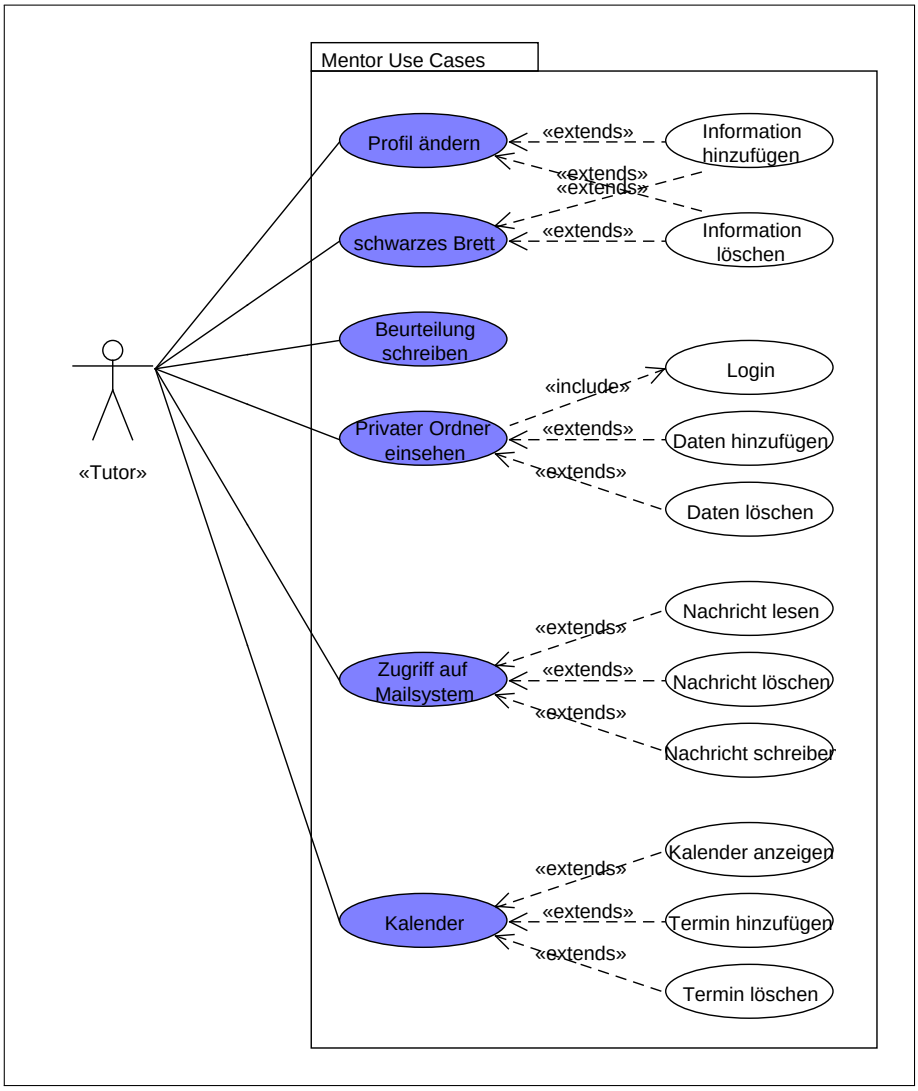


Abbildung 1: Dieses Kontextdiagramm beschreibt die Funktionen, die ein eingeloggter Tutor ausführen kann. Des Weiteren sind die Funktionen auf einzelne Unterfunktionen aufgeschlüsselt.

1.3 Sequenzdiagramm

1.4 Navigationdiagramm

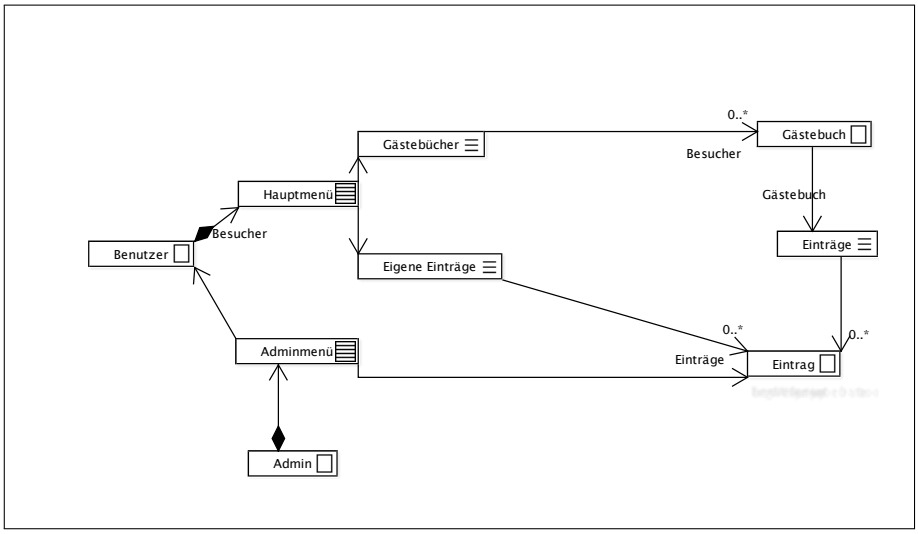


Abbildung 2: Dieses Navigationdiagramm beschreibt die Menüführung für ein Gästebuch und Gästebucheinträge. Es wird zwischen einem Menü für normale Besucher und Administratoren unterschieden. Ausserdem ist eine Navigation zu einem Gästebuch oder zu allen erstellten Einträgen eines Benutzers vorgesehen.

1.5 ER-Diagramm

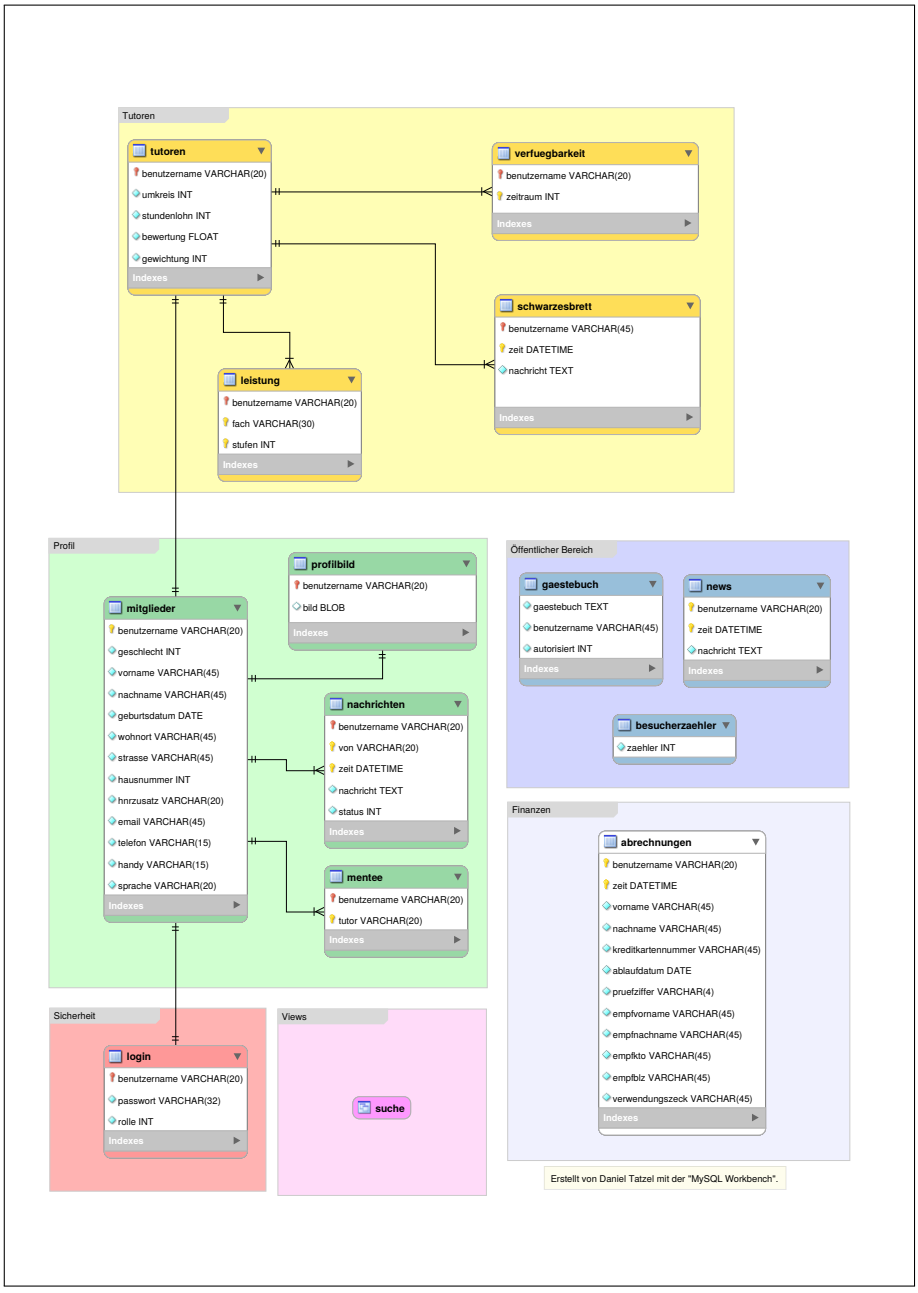
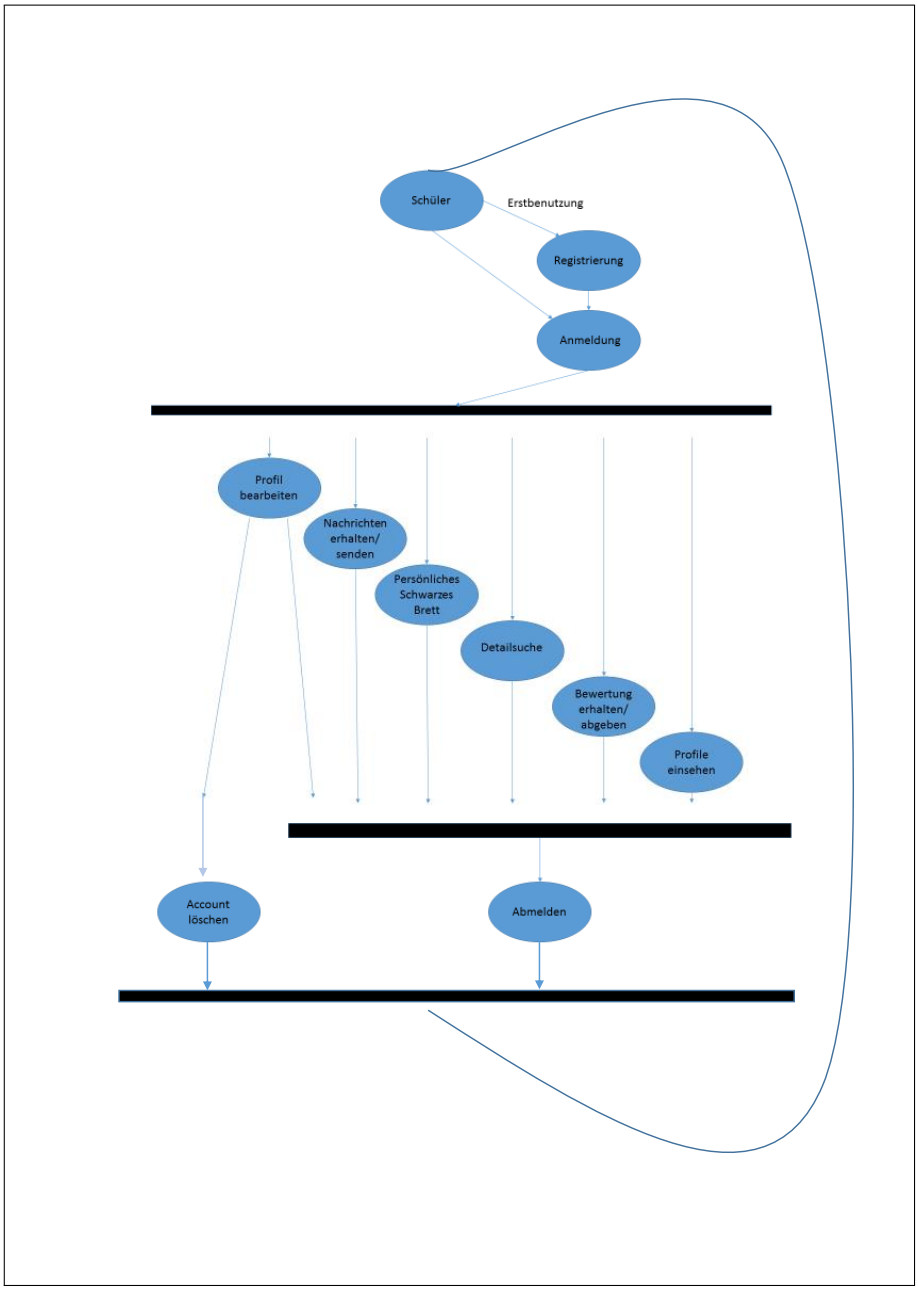


Abbildung 3: Entity-Relation Diagramm der Datenbank

1.6 Activitydiagramm



2 Objectives

2.1 Business Objectives

Ziel der Webseite ist es Tutoren in der Nähe zu finden, und mit Ihnen in Verbindung zu treten. Durch die Möglichkeit ein persönliches Profil erstellen zu können, sollen Nutzer dauerhaft an unser Webangebot gebunden werden. Potentielle Nutzer sollen direkt an Hochschulen geworben werden. Auch Kooperationen mit Universitäten und Hochschulen sollen geschlossen werden.

Unser Webangebot soll kostenfrei sein. Finanziert werden soll das Angebot durch gezielte Werbung.

2.2 System Objectives

Das Webangebot Personal Tutoring Service soll als "Templateäufgebaut werden. Somit soll es möglich sein, leicht andere Vermittlungsportale, wie z.B. eine Mitfahrzentrale aufzubauen. Somit sollen sich mit geringem Aufwand schnell neue Geschäftszweige erschliessen lassen.

3 Systems Requirement

3.1 F1: Öffentlicher Bereich

F11: Startseite

Informiert den Besucher über das Auswahlkonzept bzw. −kriterien der Tutoren. Ebenfalls wird eine Karte mit den Standorten von Tutoren in Deutschland implementiert. Ein Teaser über das Motto und der Angebote der Website soll den Besucher ansprechen und so sein Interesse fördern. Durch die Anzeige von Top Gästebucheinträgen wird dem Gast das Gefühl der Vertrauenswürdigkeit und der Kompetenz übermittelt.

Karte
Es wird eine Karte von Google-Maps eingebettet. Dies funktioniert mit Hilfe des bereitgestellten Iframes von Google.

Slideshow
Die Bilder der Slideshow werden mit HTML5 zur Verfügung gestellt. Mittels CSS3 werden die Bilder formatiert. JQuery sorgt für die Slideanimation.

Video
Das Video wird mit HTML5 implementiert und ist dadurch ohne Flash abspielbar. Die Kontrollfunktionen werden ebenfalls mit HTML5 zur Verfügung gestellt. Controls selbst erstellen

Top Gästebucheinträge

Nach dem Laden der Seite wird eine Javascript-Funktion ausgeführt, welche die Top-Gästebucheinträge über ein Json-Objekt erhält. Diese Einträge werden über JQuery eingeblendet und über CSS3 formatiert.

Profil bearbeiten
- HTML5 Formulare
- Formulare mit HTML5+ JS überprüfen (live)
- PHP Script ? Änderung

F12: Gästebuch

Das Gästebuch ermöglicht den Nutzern eigene Kommentare abzugeben. Diese werden in einer SQL - Datenbank abgespeichert. Zusätzlich wird dazu der Autor abgespeichert und eine Variable, welche erfasst, ob ein Gästebuch eintragt bereits von einem Administrator autorisiert wurde, oder nicht. Mittels PHP, im speziellen mit dem PDO werden die Gästebucheinträge zum Frontend der Website weitergeleitet und dort angezeigt. Einträge, die sehr gute Bewertungen erhalten werden zur Werbung für die Website eingesetzt.

F13: Anzeige von verfügbaren Tutoren in der Nähe

Die Wohnort-Tutorensuche wird mit Hilfe von JavaScript und MySQL umgesetzt. Benutzer gibt einen Ort ein und die Datenbank wird nach Einträgen von Tutoren mit identischem Ort durchsucht und die Anzahl der Einträge wird anschließend zurükkgegeben. Eine implementierte Google Map zeigt den Ort von des eingegebenen Ortes an.

F14: Erweiterte Suchkriterien

Die Suche wird mittels einer View realisiert. Diese View enthält alles notwendigen Informationen, die es dem User ermöglichen das gewünschte Suchergebnis zu finden,

- Benutzername
- Wohnort
- Umkreis
- Stundenlohn
- Bewertung
- Fächer
- Stufen

Die View wird verwendet um in den normalen Tabellen keine Redundanzen zu haben.

F15: Registrierungs- und Anmeldefenster

Die Registrierung erfolgt über ein Formular, in der alle Daten, die unbedingt notwendig sind, ausgefüllt werden müssen. Das OnlineFormular ist im HTML5 standard geschrieben, um gleich Validierungen der Eingaben durchführen zu können, ohne die Informationen erst zum Server schicken zu müssen.

Eingaben, die eindeutig sein sollen(wie z.B. der Benutzername) werden während der Eingabe mittels AJAX,JAVASCRIPT zum Server geschickt und dort überprüft, sodass der Benutzer sofort erfährt(ohne auf einen Button klicken zu müssen), ob dieser Benutzername schon vergeben ist. Wenn alle Daten eingegeben wurden, die Validierungen erfolgreich waren, klickt der Benutzer auf einen Button und schickt die Daten damit zum Server. Dort werden die Daten mittels PHP zur Datenbank geschickt, um dort gespeichert zu werden.

Ist ein Nutzer registriert, so kann er sich über die Login-Maske, die auf jeder Seite eingebunden ist, registrieren. Hierfür wird eine Datenbank Abfrage mittels PHP und SQL über PDO generiert um die Identität des Nutzers zu prüfen. Bei erfolgreicher Authentifizierung der Benutzerdaten werden mehrere Sesseion-Variablen, die angeben, dass der Benutzer angemeldet ist und um welchen Benutzer es sich handelt, initialisiert um immer entsprechende Informationen über den Benutzer zu haben.

F16: About us (Impressum)

Das Impressum enthält folgende Informationen: Firma, Straße, Postleitzahl, Ort, Telefonnummer, E-Mailadresse, Internetadresse, Vertretungsberechtigter, Geschäftsführer, Inhaltlich Verantwortlicher und Haftungshinweis.
Hierbei handelt es sich um statischen Inhalt für den HTML als Formatierung des Textes benutzt wird. Erreichbar ist das Impressum über einen Link in der Navigationsleiste.

F17: Preismodell und Zahlungsinfos

Das Preismodell und Zahlungsinfos enthält Informationen darüber, welche Formen der Abrechnung (z.B. Kreditkarte) unterstützt werden und wie die Abrechnung funktioniert.
Hierbei handelt es sich um statischen Inhalt für den HTML als Formatierung des Textes benutzt wird. Erreichbar ist Preismodell und Zahlungsinfos über einen Link in der Navigationsleiste.

F18: Auswahl der Sprache

Die Auswahl der Sprache erfolgt über eine Serverseitige Sessionvariable, mit Hilfe derer die Serverseitig eingesetzte Skriptsprache PHP sich über den gesamten Verlauf der Interaktion mit dem User "merken"kann, welche Sprache der User benutzen will. Alternativ kann ein zusätzliches Feld in der Datenbank für die Sprache reserviert werden, womit bei jedem neuen Einloggen des Benutzers die Sprache automatisch ausgewählt wird. Sollte der Benutzer die Sprache dann doch einmal ändern, wird diese Änderung

unverzüglich in die Datenbank geschrieben. Umgesetzt wird das erstellen des Formulars so umgesetzt:

- Für jeden Teil der Seite gibt es eine Vorlage in den Sprachen(z.B. Impressum)
- PHP wählt dann zur Laufzeit aus, welche Elemente der Seite in der Sprache geladen werden muss
- und fügt alle Elemente mit Hilfe einer leeren "Dummy"Datei zu einer, korrekten, HTML5 Datei zusammen
- welche dann an den Client geschickt wird.

F19: Support und Kontaktdaten

Der Benutzer erfährt hier die Kontaktdaten des Supports, an den er sich bei Fragen und Problemfällen wenden kann. Hierbei handelt es sich um statischen Inhalt für den HTML als Formatierung des Textes benutzt wird. Erreichbar ist das Impressum über einen Link in der Navigationsleiste.

3.2 F2: Administrator

F21: Anmelden

Der Administrator kann sich über die Login-Maske, die auf jeder Seite eingebunden ist, anmelden. Hierfür wird eine Datenbank Abfrage mittels PHP und SQL über PDO generiert um die Identität des Administrator zu prüfen. Bei erfolgreicher Authentifizierung der Benutzerdaten werden mehrere Session-Variablen, die angeben, dass der Administrator angemeldet ist und um welchen Benutzer es sich handelt, initialisiert um immer entsprechende Informationen über den Benutzer zu haben. Danach hat der Administrator erweiterte Zugriffsrechte um ihm die Verwaltung zu erleichtern.

F22: Persönliche Einstellungen

Wie jeder Nutzer verfügt der Administrator über ein persönliches Profil welches Vorname, Nachname, Geburtsdatum, Profilbild, Straße, Hausnummer, Wohnort, E-Mail, Telefon und Handynummer enthält. Die Bearbeitung und Löschung dieser Informationen ist über entsprechende Masken und Formularen zu er-möglichen.

F23: Bearbeiten von Kundenkonten[1] und Kundeninformationen[2]

Ist ein Nutzer als Administrator eingeloggt, hat dieser die Möglichkeit sich eine Übersicht aller Kundenkonten anzeigen zu lassen. Diese Informationen werden in Form einer Tabelle auf einer separaten Seite angezeigt. Beim Aufruf der Seite werden alle notwendigen Informationen über Ajax aus der Datenbank geladen und über JSON an den Browser des Admins geliefert. Um bei einer grossen Datenbank die Wartezeit gering zu halten, werden die Daten nur teilweise aus der Datenbank geladen. So ist es

beispielsweise nicht sinnvoll mehr wie 20 Datensätze gleichzeitig zu laden, da in der Regel nicht mehr angezeigt werden kann.

Durch eine Tabellen-PlugIn für jQuery werden die Rohdaten übersichtlich dargestellt. Die Tabelle soll ausserdem die Möglichkeit bieten die Datensätze nach Spalten zu sortieren.

Änderungen der Daten werden ebenfalls über jQuery an die Datenbank übergeben, wenn der Administrator einen SSpeichern"Knopf drückt.

Die Validierung der Daten wird über HTML5 und gegebenenfalls mit einer PHP-Funktion durchgeführt.

F24: Zugriff auf Nachrichtensystem

In der Datenbank befinden sich eine grosse Tabelle in der alle Nachrichten als Rohdaten abgelegt werden. Jeder Datensatz hat die Felder: Absender, Empfänger, Datum, Zeit, Nachricht, gelesen. Über SQL-Abfragen werden die jeweiligen Nachrichten aus der Datenbank geladen und über JSON an ein jQuery-Script übergeben. Möchte ein Nutzer Beispielsweise alle gesendeten Nachrichten einsehen, wird der aktuell angemeldete Nutzername in die "WHEREKlausel der Datenbankabfrage eingefügt, um so nur die relevanten Nachrichten abzufragen.

Zum Abrufen der Nachrichten wird dem angemeldeten Nutzer eine spezielle Webseite zur Verfügung gestellt, auf welcher er Nachrichten abrufen und erstellen kann. Die Anzeige wird in Form einer Tabelle über ein jQuery-Plugin erstellt.

Zum Versenden von Nachrichten muss ein Nutzer den Benutzernamen eines Empfängers in ein Textfeld eingeben. Bevor eine Nachricht abgesendet werden kann, erfolgt eine Validierung des Benutzernamens des Empfängers. Diese Validierung wird durch ein Javascript vorgenommen. Für das Eingabefeld des Empfängers ist es denkbar eine Autovervollständigung über Ajax zur Verfügung zu stellen.

F25: Freischaltung von Gästebucheinträge

Der Administrator ruft das Gästebuch auf und sieht alle nicht autorisierten Einträge. Diese kann er dann autorisieren. Dann werden diese in der Datenbank aktualisiert.

F26: Abmelden

Jeder angemeldete Nutzer kann sich auch wieder abmelden, dies geschieht durch löschen einer Session-Variable, die bei der erfolgreichen Anmeldung gesetzt wurde. Das löschen der Variable hat keinen Einfluss auf die restlichen Inhalte der Session.

F27: Besucherzähler

Für jeden Besucher der Seite wird eine PHP Session in Form eines Cookies, der drei Stunden aktiv ist, initialisiert. Bei jedem Aufruf wird geprüft, ob eine Session-Variable, die zur Erkennung benutzt wird, gesetzt ist. Falls diese nicht gesetzt ist, dann wird der Besucherzähler um eins erhöht. Ist die Variable jedoch gesetzt, dann wurde der Besucher schon gezählt und es wird nichts unternommen.

[1] Kundenkonten: Tutorenkonten und Schülerkonten
[2] Kundeninformationen: Persönliche Informationen der Kundenkonten

3.3 F3: Schüler

Jeder Besucher hat die Möglichkeit ein Profil anzulegen und sich mit den erhaltenen Zugangsdaten einzuloggen. Dadurch erhält er Zugriff auf seinen Privaten Bereich, in dem er seine bei der Registrierung hinterlegten persönlichen Daten ändern kann. Die persönlichen Daten setzen sich aus Vorname, Nachname, Alter, Anschrift, Schultyp, Jahrgangsstufe, Passwort Kontaktdaten und Profilbild zusammen. Es besteht die Option seinen Account zu löschen.

F31: Detaillierte Suche

Die Suche wird mittels einer View realisiert. Diese View enthält alles notwendigen Informationen, die es dem User ermöglichen das gewünschte Suchergebnis zu finden,

- Benutzername
- Wohnort
- Umkreis
- Stundenlohn
- Bewertung
- Fächer
- Stufen

Die View wird verwendet um in den normalen Tabellen keine Redundanzen zu haben.

F32: Tutorprofil/Tutorbewertung

Der registrierte Anwender ist berechtigt das Profil des Tutors einzusehen, das für den Anwender folgende Informationen bereit hält:

- Geschlecht
- Stundenlöhne
- Verfügbarkeiten
- Fächer / Jahrgangsstufen
- Verfügbare Orte / Regionen - Eintrag auf Karte
- Profilbild
- Kontaktdaten

- Lehrerbewertung

Er ist in der Lage das Schwarze Brett des Tutors einzusehen.

F33: Zahlungsinformationen

Der Schüler zahlt mit Kreditkarte. Dazu gibt er seine Kreditkarteninformationen in ein HTML5 Formular u. a. Name, Vorname, Benutzername, Kreditkartennummer, Ablaufdatum der Karte, die Prüfziffer und den zu zahlenden Betrag. Diese werden auf Richtigkeit der Informationen geprüft (HTML5 Formular) und dann für die weitere Verwendung in der MySQL Datenbank gespeichert.

3.4 F4: Lehrer / Mentor

F41 Profil einstellen

Der Tutor kann ein Profil erstellen, um sich kurz vorzustellen und Werbung in eigener Sache machen zu können. Schüler können ihn bewerten (z.B. Verbesserung der Note).
Das Profil soll beinhalten / anbieten:

- Geschlecht
- Stundenlöhne verlangen
- Verfügbarkeiten angeben
- Fächer / Jahrgangsstufen einstellen
- Verfügbare Orte / Regionen anbieten - Eintrag auf Karte
- Zahlungsdaten zur Verfügung stellen
- Sieht in einer Übersicht seine Schüler und einen Kalender mit den nächsten Terminen
- Rechnungen / Zahlungserinnerungen verschicken
- Profilbild
- Kontaktdaten
- Lehrerbewertung / Gästebuch

F42 Schwarzes Brett

Das Schwarze Brett wird mit JavaScript und MySQL umgesetzt. Die Ankündigungen werden in die Datenbank geschrieben und können von dort wieder abgerufen werden.

4 Annahmen und Beschränkungen

4.1 Annahmen

4.2 Beschränkungen

Leistung

Einer der wichtigesten Dinge, auf die zu achten ist, ist die Kommunikation zwischen javascript und PHP über AJAX. Wenn man mit Hilfe von javascripts Daten dynamisch über PHP bzw. PDO aus einer Datenbank laden will, sollte man sehr genau auf die Performance dieses Prozesses achten. Wenn man also *.php Skripte ansteuert, sollte man in diesen darauf achten, dass das Skript an sich sehr performant ist(z.B. nicht viele, grosse Klassenobjekte generieren, Operationen durchführen die sehr lange brauchen, wie z.B. aufwendige Hash-Operationen), da sonst die Datengenerierung im PHP-Skript sehr lange dauert und dies dazu führt, dass es relativ lange dauert, bis aktualisierte Informationen auf dem Bildschirm des End-Users angezeigt werden.

Um diesem Umstand gerecht zu werden, sollen die Files so angelegt werden, dass pro File möglichst wenig Funktionen enthalten sind. Damit soll ein möglichst hoher Grad an Unabhängigkeit von anderen Funktionen erreicht werden, sodass in jedem *.php File nur die für die darin enthaltenen Funktionen benötigten Komponenten ausgeführt werden müssen.

PHP Funktionen aus Javascript

Da man mit Hilfe javascript nur einzelne *.php-Skripte ansprechen kann und nicht einzelne Funktionen aus php wird es für die meisten javascript-Funktionen eine *.php-Datei geben, die dann die Dinge ausführt, die Serverseitig gemacht werden sollen. Damit kann man auch mehrere Serverseitige Methoden auf ein Mal abarbeiten.

Datenbank

1. Durch das Einsetzen einer MySQL Datenbank sind wir durch das Relationale Design in der Optimalen Aufteilung, wie man es bei einem NoSQL System umsetzen würde um bessere Reaktionszeiten zu erhalten, eingeschränkt.

2. Durch den Einsatz eines einzigen Entitätstyp für den Besucherzähler ist die maximale Anzahl an neuen Nutzer, die gleichzeitig auf die Seite zugreifen, die gezählt werden können begrenzt.

3. Benutzernamen dürfen maximal eine Länge von 20 Zeichen haben um unnötigen Speicherplatz zu vermeiden.

4. Um so wenig NULL Werte wie möglich zu bekommen, ist darauf zu achten, dass Aufteilung entsprechende gewählt wird