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

Version: **[1.1]** **Date:** **[**10.05.2014**]**

Beitragende:

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

VersionsNr	Datum	Auslöser	Veränderungsgrad	Beschreibung
1.0	03.05.2014	AS	Erster Entwurf	First Draft
1.1	10.05.2014	AS	Verbesserung	Missing Data added: Sequencediagram, Annahmen
1.2	25.06.2014	Alle	Anpassung	Final Release

Tabelle 1: Überarbeitungshistorie

Inhaltsverzeichnis

1 Software Architecture

1.1 Anwendungsfalldiagramm 1 - Administrator

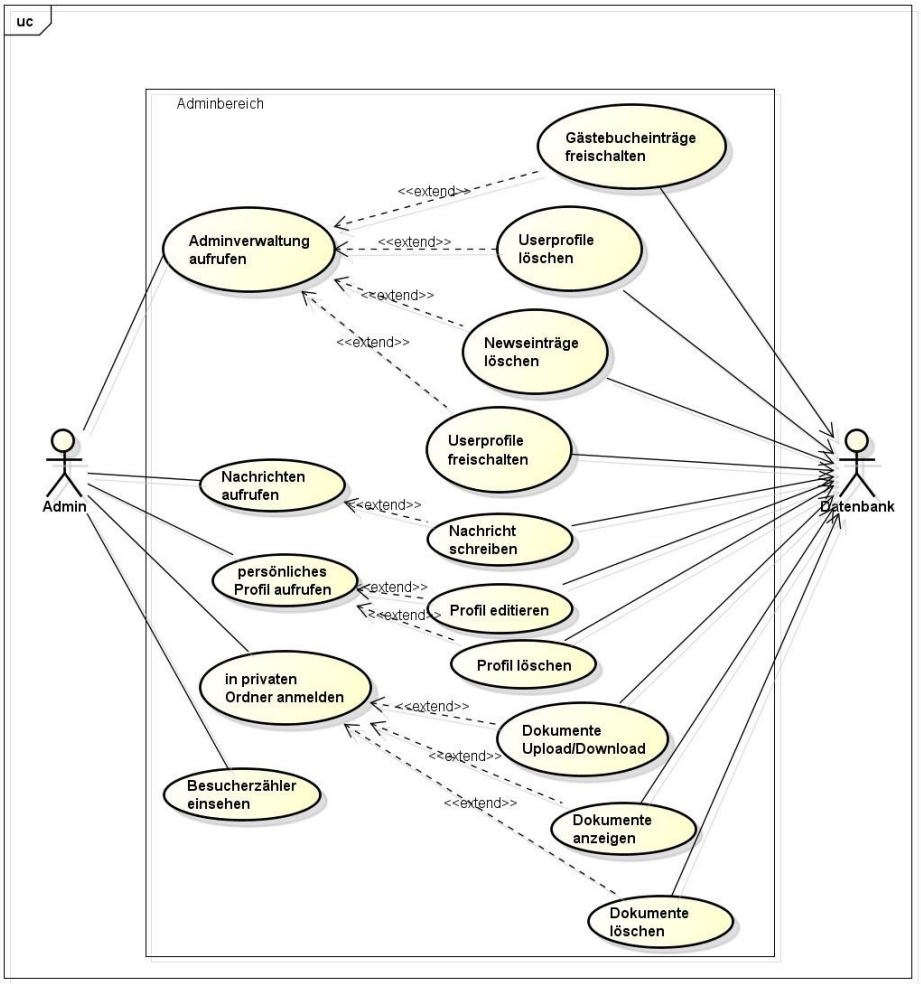

Abbildung 1: Dieses Anwendungsfalldiagramm beschreibt die Funktionen, die ein eingeloggter Administrator ausführen kann. Des Weiteren sind die Funktionen auf einzelne Unterfunktionen aufgeschlüsselt.

1.2 Anwendungsfalldiagramm 2 - Tutor

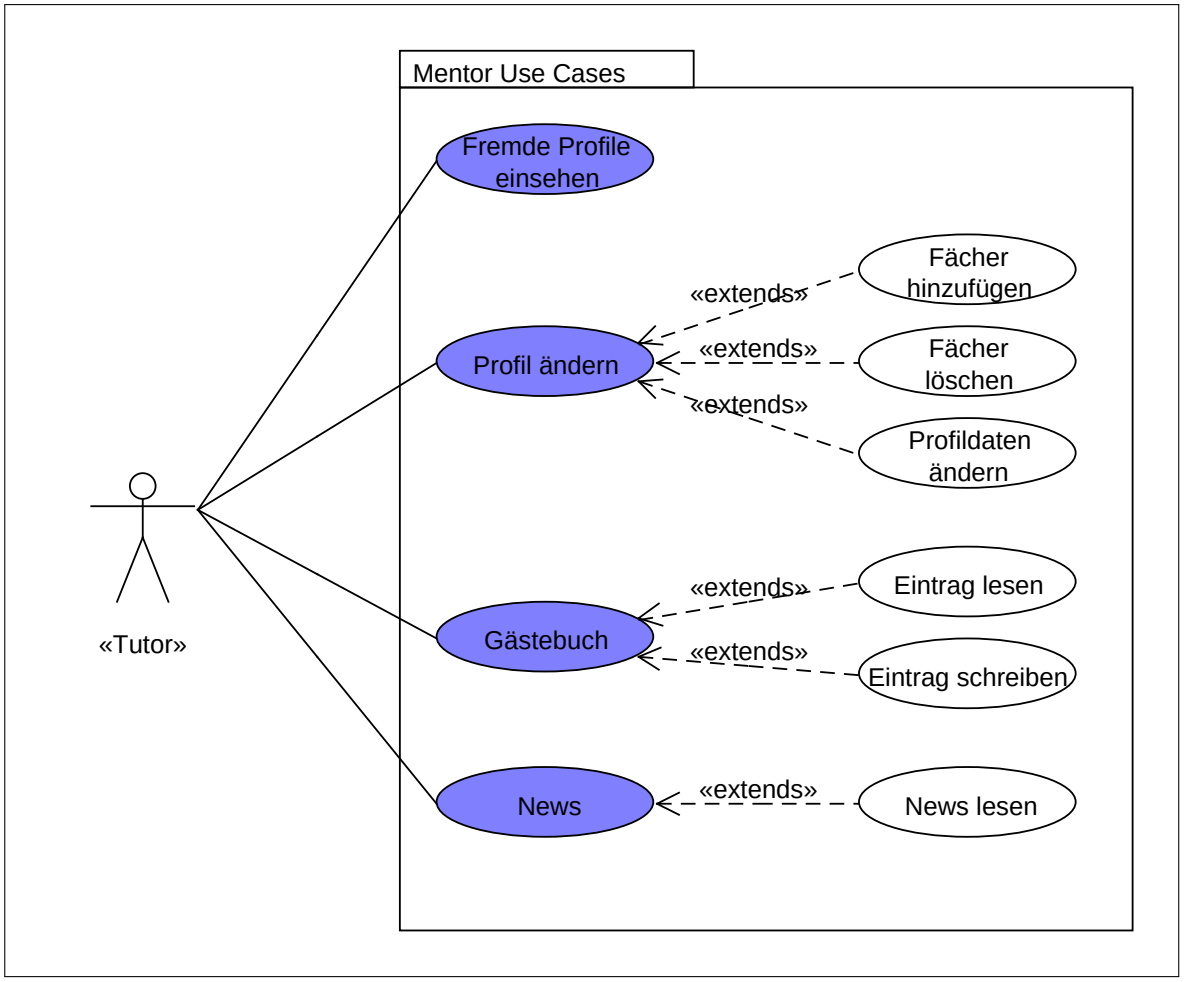

Abbildung 2: Dieses Anwendungsfalldiagramm beschreibt die Funktionen, die ein eingeloggter Tutor ausführen kann. Des Weiteren sind die Funktionen auf einzelne Unterfunktionen aufgeschlüsselt.

1.3 Sequenzdiagramm

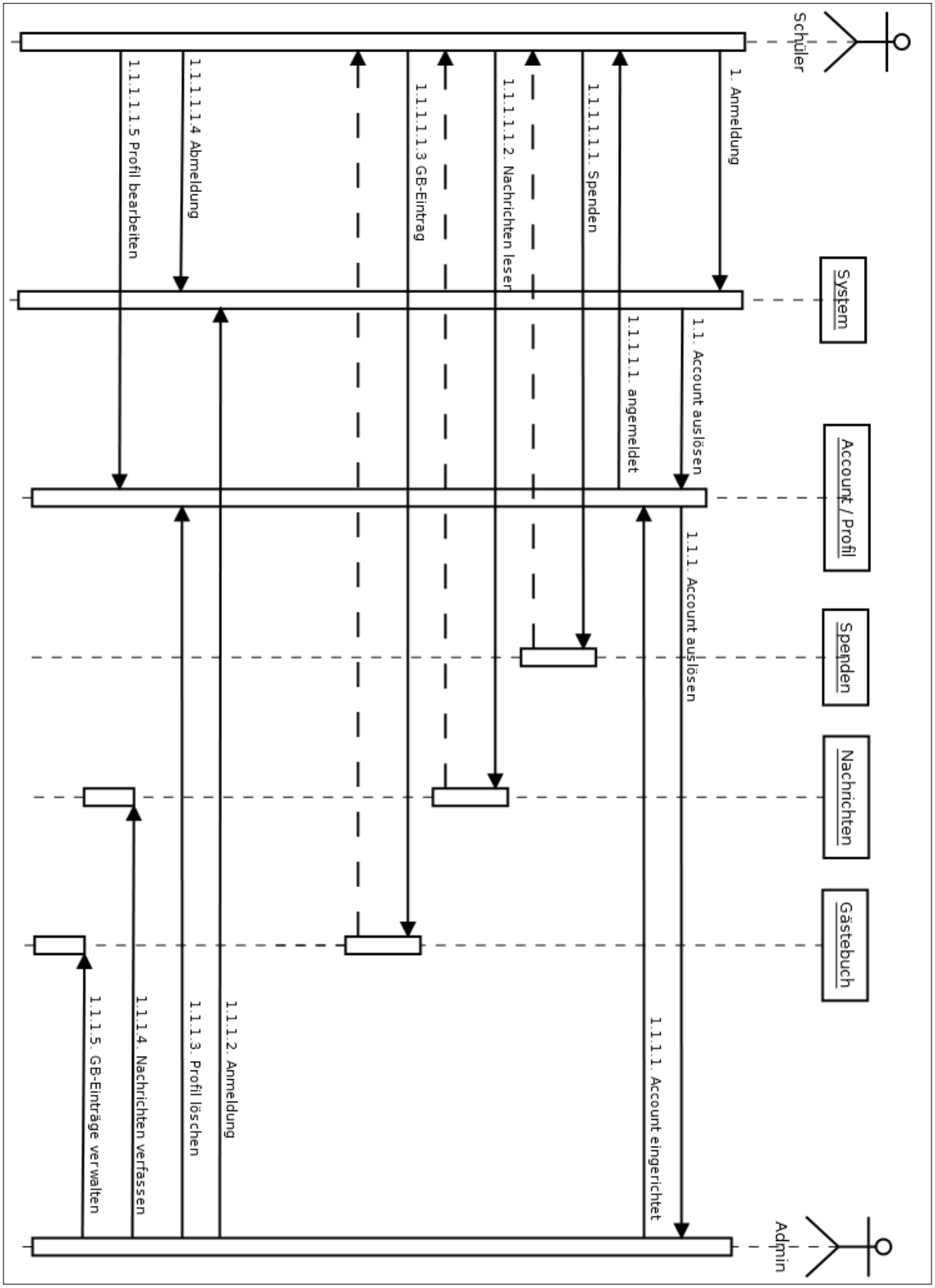

Abbildung 3: Dieses Sequenzdiagramm beschreibt, in welcher Reihenfolge Befehle ausgeführt werden müssen, um eine Aktion vollständig und korrekt auszuführen

1.4 Navigationdiagramm

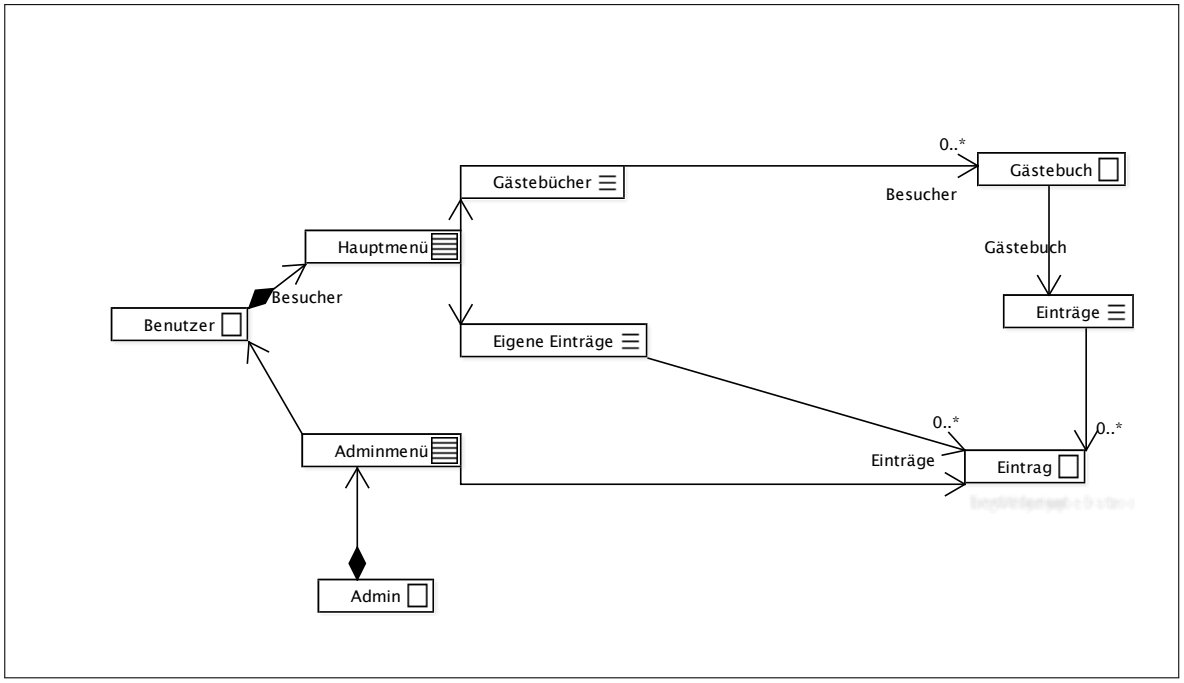

Abbildung 4: Dieses Navigationdiagramm beschreibt die Menüführung für ein Gästebuch und Gästebucheinträge. Es wird zwischen einem Menü für normale Besucher und Administratoren unterschieden. Ausserdem ist eine Navigation zu einem Gästebuch oder zu allen erstellten Einträgen eines Benutzers vorgesehen.

1.5 ER-Diagramm

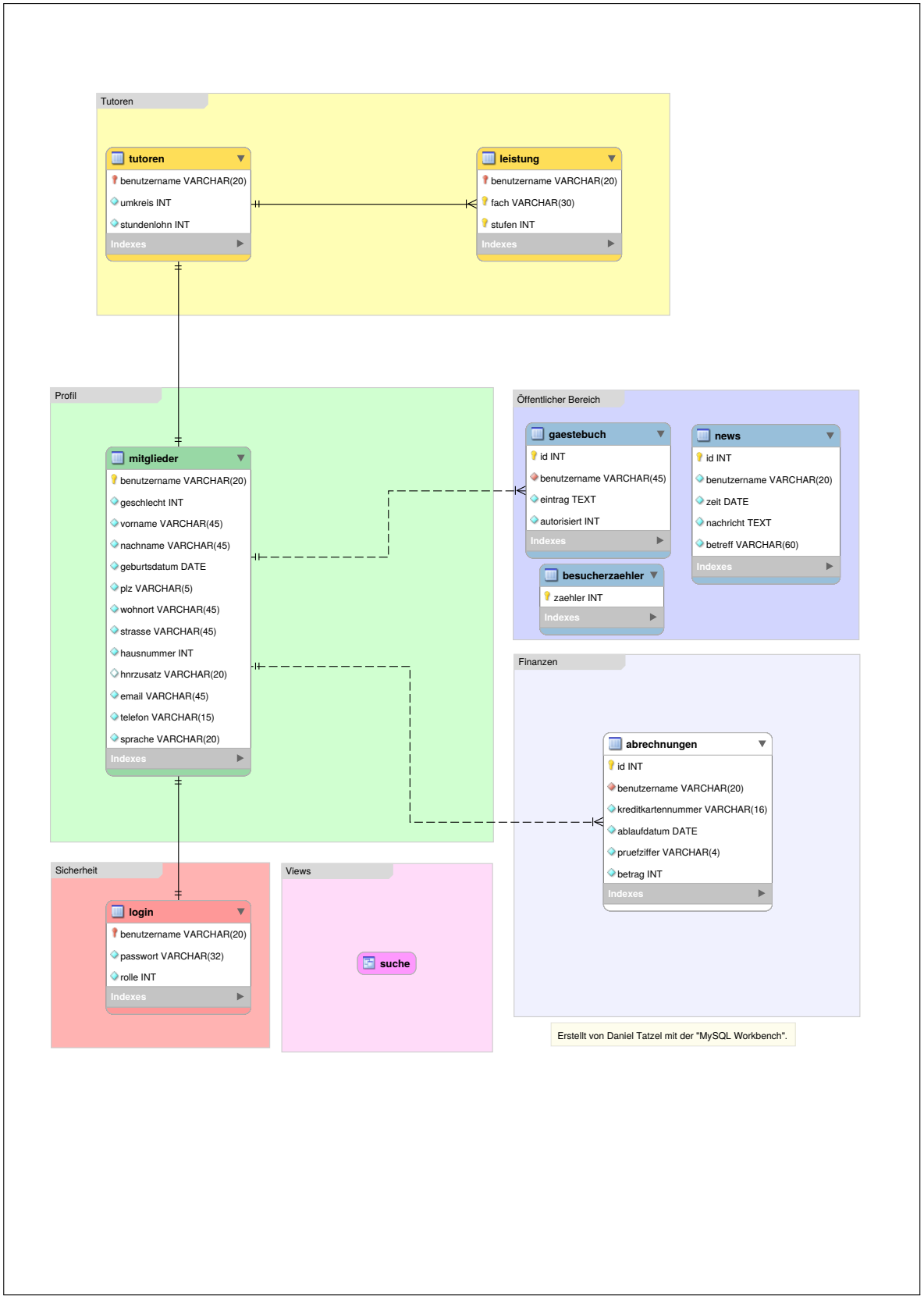

Abbildung 5: Entity-Relation Diagramm der Datenbank

1.6 Activitydiagramm

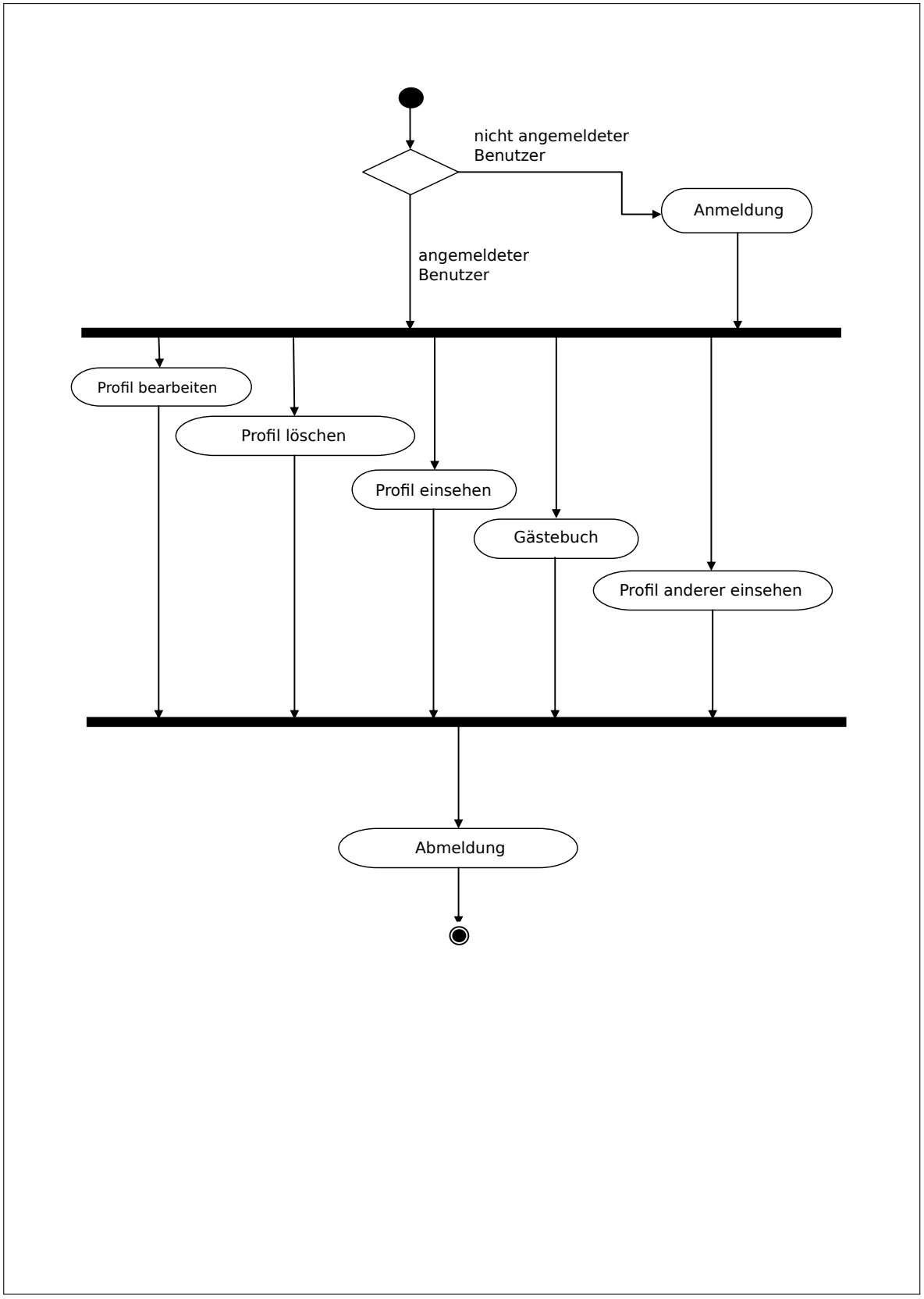

Abbildung 6: User muss sich anmelden, um die erweiterten Funktionen nutzen zu können. Bei erfolgreicher Anmeldung hat der User die Möglichkeit folgende Aktivitäten zu nutzen: Profil bearbeiten, Nachrichten senden und empfangen, Zugriff auf das Schwarze Brett, erweiterte Detailsuche, Bewertung erhalten und abgeben und in Profile einsehen. Um den persönlichen Bereich wieder zu verlassen, muss der User sich abmelden.

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 'Template' aufgebaut 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 Funktion und Zweck der Webseite. Ebenfalls wird eine Karte mit der Route zum Hauptsitz der Tutoren AG 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
Mit Hilfe der Google Maps JavaScript API wird eine Karte eingebettet. Außerdem besteht für den Besucher die Möglichkeit, sich die Route von seinem jetzigen Standort bis in die Siemensstraße 12 in Regensburg anzeigen zu lassen. Um nicht wiederholt die Standortdaten preisgeben zu müssen werden diese nach erstmalig erlaubten Zugriff in einem Cookie gespeichert. Dies geschieht mit Hilfe des jQuerry-Cookie-Plugins.

Slideshow
Die Bilder der Slideshow werden mit HTML5 zur Verfügung gestellt. Mittels CSS3 werden die Bilder formatiert und animiert. Eine Fallback Lösung, welche in JavaScript implementiert ist, wird durch den Modernizr realisiert.

Video
Das Video wird mit HTML5 implementiert und ist dadurch ohne Flash abspielbar. Die Kontrollfunktionen werden ebenfalls mit HTML5 zur Verfügung gestellt. Gesteuert wird der HTML5 Videoplayer mit JavaScript.

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 AJAX eingeblendet und über CSS3 formatiert.

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 eines JSON-Objektes werden die Gästebucheinträge zum Frontend der Website weitergeleitet und dort über eine JavaScript-Funktion angezeigt. Besonders positive Einträge werden zur Werbung für die Website eingesetzt.

F13: 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. Zusätzlich werden die Eingaben per Javascript überprüft. Wenn alle Daten eingegeben wurden, die Validierungen erfolgreich waren, klickt der Benutzer auf einen Button und schickt die Daten damit zum Server. Dort wird der Benutzername überprüft und falls dieser noch nicht vorhanden ist, dann werden die Daten mittels POST zur Datenbank geschickt, um dort

gespeichert zu werden. Der Aufruf des Scripts zum Eintragen der Informationen in die Datenbank wird mittels JQuery ausgeführt. Vor dem Eintragen der Daten in die Datenbank erfolgt eine Validierung dieser Daten durch JQuery-Validate.

Ist ein Nutzer registriert, so kann er 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 Nutzers zu prüfen. Bei erfolgreicher Authentifizierung der Benutzerdaten werden mehrere Session-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. Das Script zum Anmelden wird ebenfalls per JQuery aufgerufen.

F14: Impressum

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.

F15: Auswahl der Sprache

Jeder Besucher bekommt eine Serverseitige Sessionvariable, in der die Sprache gespeichert wird. Im Kopf einer jeden Seite wird diese Variable mit der Sprache der aktuellen Seite definiert. Diese Variable wird im weiteren Verlauf der Seite genutzt um in den entsprechenden Bereichen wie in der „Navigation“ genutzt um die Inhalte der entsprechenden Sprache zu laden. Dies wird nur für die Elemente genutzt, die auf jeder Seite eingebunden werden. Jede Seite besitzt eine „Dummy-Datei“ in der der entsprechende Inhalt geladen wird. Diese „Dummy-Datei“ wird für entsprechende Überprüfungen und für das ausführen von Scripten benutzt.

Wechselt ein Benutzer die Sprache, dann wird er mithilfe von RewriteRules in einer htaccess Datei auf die entsprechende Seite der jeweils anderen Sprache geleitet. Es befindet sich eine htaccess Datei mit entsprechenden RewriteRules in den jeweiligen Ordner für die Sprachen. PHP prüft zur Laufzeit welche Sprache gesetzt ist und lädt die entsprechende Sprache für das Template.

F16: 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.

F17: Nachrichten

Die Nachrichten der Webseite sind für jedermann einsehbar. Diese werden von einem PHP-Skript per JSON-File vom Server übertragen und über Javascript mithilfe von AJAX angezeigt. Die Formatierung übernimmt CSS3.

F18: Anzeige von verfügbaren Tutoren an einem bestimmten Ort

Die Wohnort-Tutorensuche wird mit Hilfe von JavaScript und MySQL umgesetzt. Benutzer gibt einen Ort oder ein Fach ein und die Datenbank wird nach Einträgen von Tutoren mit identischem Ort oder Fach durchsucht und die Anzahl der Einträge wird anschließend zurükkgegeben. Suchvorschläge erhält der Nutzer unmittelbar während der Suche, sobald mehr als 2 Zeichen eingegeben wurden.

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. Der Aufruf des Scripts zum Anmelden wird mittels JQuery ausgeführt.

F22: Persönliche Einstellungen

Der Admin kann durch Ausfüllen von HTML5 Formularen seine Profildaten bearbeiten. Die Eingabe des Users wird mittels HTML5 und JavaScript überprüft. Die Daten in der Datenbank werden über ein PHP Skript geändert.

F23: Adminverwaltung

Ist ein Nutzer als Administrator eingeloggt, hat dieser die Möglichkeit sich eine Übersicht aller Kundenkonten anzeigen, freischalten und über einen extra Menüpunkt auch löschen zu lassen. Diese Informationen werden in Form einer Tabelle auf einer separaten Seite angezeigt. Beim Aufruf der Seite werden alle notwendigen Informationen über ein PHP-Skript aus der Datenbank geladen und für die Darstellung im Browser vorbereitet.

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

Der Administrator kann über die Administrator-Verwaltung alle nicht autorisierten Gästebucheinträge sehen und gegebenenfalls autorisieren. Diese werden daraufhin in der Datenbank aktualisiert. Der Aufbau und Funktion dieser Tabelle entspricht im Wesentlichen dem Grundgerüst der Kundendaten.

Der Administrator kann über die Administrator-Verwaltung alle nicht Newseinträge sehen und gegebenenfalls löschen. Diese werden daraufhin in der Datenbank aktualisiert. Der Aufbau und Funktion dieser Tabelle entspricht im Wesentlichen dem Grundgerüst der Gästebucheinträge.

F24: Nachrichten

Der Administrator hat die Möglichkeit auf der News-Seite neue Nachrichten zu schreiben. Dazu gibt er den Betreff und die Nachricht in ein HTML 5 Formular ein. Nach Eingabe der Daten werden diese durch JQuery Validate auf ihre Richtigkeit geprüft. Anschließend werden diese in der Datenbank gespeichert und für alle sichtbar (auch für nicht angemeldete User).

F25: 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.

F26: Besucherzähler

Jeder Aufruf der Startseite wird über eine enstprechende Zählvariable, die in der Datenbank gespeichert ist, mitprotokolliert. Hierfür wird beim Aufbau der Startseite der aktuelle Zählerstand in der Datenbank um eins erhöht, dies geschieht mittels PHP und PDO zum Ansprechen der Datenbank.

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

F27: Privater Ordner

Der Administrator ist in der Lage über das Menü auf den Privaten Ordner zu zugreifen. Nach Aufruf der Seite kommt eine Login-Maske, in der sich der Administrator Authentifizieren muss. Diese Authentifizierung geschieht über eine .htaccess, in der die entsprechenden Optionen zum Schutz eines Verzeichnisses gesetzt sind und eine .htpasswd Datei, in der sich die Zugangsdaten befinden. Auf der Seite des Privaten Ordners hat der Admin die Möglichkeit das Passwort für den Ordner zu ändern sowie neue Dateien hochzuladen oder zu löschen. Diese Möglichkeiten werden mithilfe von HTML5 Formularen und PHP Scripten gelöst. Zur Übersichtlichkeit werden die Bereichen zum ändern des Passworts und zum hochladen von Dateien mittels CSS verborgen. Zum Anzeigen der Bereiche ist ein Button vorhanden, der mithilfe von Javascript die Bereiche sichtbar macht.

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, Passwort und Kontaktdaten zusammen. Es besteht die Option seinen Account zu löschen.

F31: Schülerprofil

Der Schüler kann durch Ausfüllen von HTML5 Formularen seine Profildaten bearbeiten. Die Eingabe des Users wird mittels HTML5 und JavaScript überprüft. Die Daten in der Datenbank werden über ein PHP Skript geändert.

3.4 F4: Lehrer / Mentor

F41 Profil einstellen

Der Tutor kann durch Ausfüllen von HTML5 Formularen seine Profildaten bearbeiten. Die Eingabe des Users wird mittels HTML5 und JavaScript überprüft. Die Daten in der Datenbank werden über ein PHP Skript geändert.

3.5 F5: Registrierte Benutzer

F51: Spenden per Kreditkarten

Ein registrierter Benutzer kann per Kreditkarte Spenden. Dazu gibt er seine Kreditkarteninformationen in ein HTML5 Formular Kreditkartennummer, Ablaufdatum der Karte, die Prüfziffer und den zu zahlenden Betrag. Diese werden mit Javascript auf Richtigkeit der Informationen geprüft (HTML5 Formular) und dann für die weitere Verwendung in der MySQL Datenbank gespeichert.

F52: Fremde Profile

Der angemeldete Benutzer ist in der Lage die Profile von anderen Benutzern einzusehen.

4 Annahmen und Beschränkungen

4.1 Annahmen

Zahlungsverkehr

Wir nehmen an, dass alle unsere Besucher über ein Mindestalter von 14 Jahre verfügen, um so keine gesonderten rechtlichen Bestimmungen für den Zahlungsverkehr erfüllen zu müssen. Wir führen keine Bezahlvorgänge durch, sondern vermitteln diese nur zwischen den jeweiligen Vertragspartnern. Diese führen die Bezahlvorgänge über externe Dienstleister durch. Die weiteren Bestimmungen des Vertrages bleiben hiervon unberührt.

Mindestalter

Das Mindestalter für die Vermittlung auf unserer Webseite beträgt 14 Jahre.

4.2 Beschränkungen

Leistung

Einer der wichtigsten Dinge, auf die zu achten ist, ist die Kommunikation zwischen Javascript und PHP über AJAX. Wenn man mit Hilfe von Javascript Daten dynamisch über PHP bzw. PDO aus einer Datenbank laden will, sollte man sehr genau auf die Performance dieses Prozesses achten. Wenn man *.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 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

5 Delivery und Schedule

VersionsNr	Datum	Auslöser	Veränderungsgrad	Beschreibung
SRA	26.03.14	09.04.14	Abgeschlossen	
SAD	16.04.14	07.05.14	Abgeschlossen	
Datenbank Design	16.04.14	03.05.14	Abgeschlossen	
Navigation und Design	14.05.14	11.06.14	Abgeschlossen	
Implementierung	25.06.14	11.06.14	Abgeschlossen	
Test	18.06.14	18.06.14	Abgeschlossen	Testbericht zugestellt
Update	25.06.14	25.06.14	Abgeschlossen	Implementierung Ende
Projekt Ende	25.06.14	26.06.14	Abgeschlossen	
Präsentation	18.06.14	02.07.14	Geplant	

Tabelle 2: Zeitplan des Projekts