

TeachEval - Web App für die Unterrichtsevaluation

Bachelorarbeit

Abteilung Informatik
Hochschule für Technik Rapperswil

Frühjahrssemester 2014

Autoren:	Peter Camastral, Daniel Marty
Betreuer:	Eduard Glatz
Projektpartner:	OdA G ZH, Zürich
Experte:	Roberto Pajetta
Gegenleser:	Markus Stolze

Inhaltsverzeichnis

1	Aufgabenstellung	6
2	Abstract	8
2.1	Ausgangslage	8
2.2	Ergebnis	8
3	Management Summary	9
3.1	Ausgangslage	9
3.2	Vorgehen/Technologien	9
3.3	Ergebnis	9
3.4	Ausblick	10
4	Einleitung	11
5	Analyse Oda G ZH	12
5.1	Ist Szenarien	12
5.1.1	Kurse verwalten	12
5.1.2	Fragebogen erstellen	13
5.1.3	Umfragen durchführen	15
5.1.4	Umfrage auswerten	15
5.2	Probleme	16
5.2.1	Inkonsistenz der Daten	16
5.2.2	Fehlerhafte Daten	16
5.2.3	Hoher Aufwand für die Auswertung notwendig	16
5.2.4	Archivierung der Umfrage Resultate	16
6	Anforderungsanalyse	17
6.1	Funktionale Anforderungen	17
6.1.1	Rollen	17
6.1.2	Use cases	18
6.1.3	UC1 an anonymer Umfrage teilnehmen	20
6.1.4	UC2 Umfrage durchführen	21
6.1.5	UC3 am System anmelden	23
6.1.6	UC4 vom System abmelden	24
6.1.7	UC5 Passwort zurücksetzen	25
6.1.8	UC6 an personalisierter Umfrage teilnehmen	27
6.1.9	UC7 persönliche Daten verwalten	28
6.1.10	UC8 <i>Kovarianz</i> zweier Fragen betrachten	29
6.1.11	UC9 Auswertung anschauen	30
6.1.12	UC10 Auswertung exportieren	31
6.1.13	UC11 Fragebogen erweitern	32
6.1.14	UC12 Umfragetoken anzeigen	33

6.1.15	UC13 Benutzer verwalten (<i>CRUD</i>)	34
6.1.16	UC14 Fragebogen verwalten (<i>CRUD</i>)	36
6.1.17	UC15 Kurse verwalten (<i>CRUD</i>)	38
6.1.18	UC16 Kursliste importieren	40
6.2	Nichtfunktionale Anforderungen	41
6.2.1	Mengenanforderungen	41
6.2.2	Leistungsanforderungen	41
6.2.3	Qualitätsmerkmale	42
6.3	Randbedingungen	46
6.3.1	Server	46
6.3.2	Client	47
7	Objektorientierte Analyse	49
7.1	Domainmodel	49
7.1.1	Benutzer	49
7.1.2	Benutzergruppe	49
7.1.3	Kurs	49
7.1.4	Umfrage	50
7.1.5	Feedback	50
7.1.6	Frage	50
7.1.7	Antwort	50
8	Technologieentscheid	51
8.1	Technologien Server	51
8.1.1	node.js	51
8.2	Technologien Datenbanken	52
8.2.1	<i>SQL</i> vs noSQL	52
8.2.2	mongodb	54
8.3	Technologien Client	55
8.3.1	JavaScript	55
8.4	Zusätzliche Libraries und Frameworks für den Server	56
8.4.1	express	56
8.4.2	jade	56
8.4.3	mongoose	57
8.4.4	bcrypt	58
8.4.5	async	58
8.4.6	node-xlsx	60
8.4.7	nodemailer	60
8.4.8	password-generator	60
8.4.9	body-parser	60
8.4.10	connect-multiparty	61
8.4.11	console-stamp	61
8.4.12	forever	61
8.5	Zusätzliche Libraries und Frameworks für den Client	62

8.5.1	Bootstrap	62
8.5.2	Bootstrap-datepicker	62
8.5.3	Bootstrap-slider	63
8.5.4	jquery	63
8.5.5	jQuery-Validation-Engine	64
9	Architektur	65
9.1	Systemübersicht	65
9.2	Entity Relationship Model	66
9.2.1	Collections	66
9.3	Verzeichnisstruktur	67
9.4	Routing	68
9.5	Views / Templating	72
9.6	Schemas	74
9.7	Module	75
10	User Interface	76
10.1	Corporate Design	76
10.2	Layout	77
10.3	Login	78
10.4	Umfrage	79
10.5	Auswertungen	80
10.6	Ressourcen verwalten	81
10.7	Fragebogen erstellen	82
10.8	Hinweismeldungen	82
11	Testing	83
11.1	mocha	83
11.2	siege	84
11.3	Usability Test	85
11.3.1	Resultate der Tests mit ausgewählten Benutzern	85
11.3.2	Feedback von Brigitte Keller	86
11.3.3	Feedback der Teilnehmer	86
12	Ergebnis	87
12.1	Ausblick	87
	Abbildungsverzeichnis	88
	Tabellenverzeichnis	89
	Glossary	90

Typografische Konventionen

Durch die ganze Bachelorarbeit werden verschiedene typografische Konventionen verwendet um verschiedene Inhalte zu unterscheiden.

Verweise

- *URLs* werden folgendermassen hervorgehoben: www.google.com
- Verweise auf andere Stellen der Bachelorarbeit sind [blau](#).
- Im Glossar aufgeführte Begriffe und Abkürzungen sind *kursiv und blau*.

Code Beispiele

Code aus einem Dokument wird folgendermassen dargestellt.

Listing 1: helloWorld.js

```
1 function fHallo(){
2   //Ausgabe von Hallo Welt! mit einer Alert-Box
3   alert("Hallo Welt!");
4 }
```

Zum Kontext unpassende Stellen des Codes werden mit ... ausgeblendet.

Listing 2: someCode.js

```
1 ...
2 function someCode(){
3   ...
4   var important = "importantPart";
5   ...
6   return important;
7 }
8 ...
```

1 Aufgabenstellung

Aufgabenstellung zur Bachelorarbeit FS 2014

„TeachEval – Web App für die Unterrichtsevaluation (2)“

Gruppe: Daniel Marty / Peter Camastral

Ausgangssituation

Die Oda G Zürich (<http://www.oda-g-zh.ch/>) ist der Branchenverband für Berufsbildung im Gesundheitswesen des Kantons Zürich. Die Oda G führt unter anderem Kurse im Lernbereich Training & Transfer (LTT) Praxis durch, die systematisch sowohl von den Lernenden und Dozierenden wie auch deren Supervisoren evaluiert werden. Zurzeit erfolgen diese Evaluationen mit Hilfe von gedruckten Formularen, die in manueller Arbeit aufwendig ausgewertet werden müssen.

Aufgabe

Ihre Aufgabe besteht darin eine Webapplikation zu entwickeln, die sowohl die Befragung als auch deren Auswertung automatisiert. Das Beantworten der Fragebögen soll mit Hilfe eines Web-Browsers möglich sein, der eine entsprechende Eingabemöglichkeit in benutzerfreundlicher Art sowohl auf PCs, Tablets als auch Smartphones anbietet.

Zudem besteht ein zweites Hauptziel der Arbeit darin, ansprechende Auswertungen zu realisieren, die auch Diagramme enthalten und als Excel- oder MS Word-Dateien bereit gestellt werden. Der Auswertungsteil der Applikation soll zudem die Möglichkeit bieten, die Antworten auf ausgewählte Fragen zu korrelieren, d.h. die Kovarianz numerischer Auswertungsergebnisse zu berechnen.

Die Webanwendung soll in JavaScript, HTML 5 und CSS erstellt werden. Benutzte Drittkomponenten (z.B. Datenbank) und Tools sollen als OSS vorliegen.

Dieses Thema bietet die vielversprechende Möglichkeit eine praktische Lösung für den Einsatz beim Kunden zu erarbeiten. Die Aufgabe wird parallel in zwei Gruppen in Konkurrenz erarbeitet.

Allgemeine Vorgaben

Die Arbeit ist gemäss den allgemeinen Vorgaben [1] durchzuführen. Dies beinhaltet auch Vorgaben zur Berichtsgestaltung.

Termine

17. Feb. 2014	Arbeitsbeginn
13. Juni 2014	Abgabe des Berichts an den Betreuer (bis 12:00)

Weitere Termine siehe Terminangaben auf dem HSR-Web (intern) bzw. nach Absprache, insbesondere Zwischenbesprechungen mit Experte und Gegenleser sowie für mündliche BA-Schlussprüfung.

Betreuung

Betreuer: Prof. Dr. Eduard Glatz, Email: eglatz@hsr.ch

Während der Durchführung der Arbeit findet nach Möglichkeit regelmässig jede Woche eine Besprechung mit dem Betreuer statt. Dazu werden entsprechende Termine bei Arbeitsbeginn festgelegt.

Referenzen

[1] Glatz, E., „Vorgaben zur Arbeitsdurchführung“, Ausgabe des 16. Februar 2014.

Rapperswil, den 17. Feb. 2014



Eduard Glatz

2 Abstract

2.1 Ausgangslage

Die OdA G Zürich ist der Branchenverband für Berufsbildung im Gesundheitswesen des Kantons Zürich. Sie führt unter anderem Kurse im Lernbereich Training & Transfer Praxis durch, die systematisch sowohl von den Lernenden und Dozierenden, wie auch deren Supervisoren evaluiert werden. Zurzeit erfolgen diese Evaluationen mit Hilfe von gedruckten Formularen, die in manueller Arbeit aufwendig ausgewertet werden müssen.

2.2 Ergebnis

TeachEval realisiert eine Webseite mit *responsive Design*, welches mit JQuery 1.11 und Bootstrap 3 umgesetzt wird. Als Webserver wird Node.js mit dem Express Framework eingesetzt. Die Applikation wird auf einem von der OdA G Zürich zur Verfügung gestellten Server betrieben, welcher über ein *SSL Zertifikat* für seine Domain(teacheval.ch) verfügt. Es wird ausschliesslich *HTTPS* auf Port 40000 unterstützt, da die HTTP Weiterleitung von der Startseite der anderen Gruppe übernommen wird.

Die Anbindung an die MongoDB Datenbank läuft über das Node.js Modul mongoosejs. Der Import und Export von Excel Dateien wird mit xlsx.js von Microsoft realisiert.

Um einfache Passwörter für Benutzer und *Tokens* für Kurse zu generieren, wird das Node.js Modul password-generator genutzt. Zusätzlich wird bcrypt für deren Verschlüsselung eingesetzt.

Die Kommunikation von Server und Client ist *REST-Like*, da auf Browser Seite *Requests* mit *HTML*(get/post) ausgelöst werden und nicht über JQuery Funktionen wie z.B. put.

3 Management Summary

3.1 Ausgangslage

Für die Bewertung der Kurse an der OdA G ZH werden aktuell noch aufwändige Auswertungen auf Papier erfasst und dann von Hand ausgewertet. Durch die grosse Verbreitung von Smartphones und Notebooks ist der Webbrowser ein fester Bestandteil unseres Alltages. Durch die Befragung in einem Webbrowser kann der Prozess der Befragung vereinfacht und die Auswertung automatisiert werden. Im Rahmen der Bachelorarbeit soll ein passender Ansatz gefunden werden um die Befragung auf allen gängigen Desktop- sowie Mobil-Browsern zu ermöglichen und die Daten auf dem Server auszuwerten.

3.2 Vorgehen/Technologien

Eine Anforderung an die Lösung war, dass die eingesetzten Technologien alle als Open Source Software verfügbar sind. Deshalb wurde als Serverplattform Node.js mit dem Express Framework für die Entwicklung der Webapplikation im modernen MVC Muster ausgewählt. Node.js bringt den Node Package Manager mit sich, von dem einige Module, wie z.B. Forever zur Serverüberwachung oder xlsx.js für den Import/Export von Excel Dateien, eingesetzt werden. Auf der Clientseite war eine Anforderung, dass möglichst alte Browser unterstützt werden, damit die Umfrage auf allen möglichen Endgeräten ausgefüllt werden kann. Deshalb wird JQuery 1.11.1 eingesetzt, welches auch alte Internet Explorer Versionen ab 6 unterstützt. Damit sich das Design auf mobilen Geräten Responsive verhält, wird Bootstrap 3 verwendet.

3.3 Ergebnis

Die erstellte Webapplikation vereinfacht die Unterrichtsbefragung stark. Bei der Lösung wurde viel Wert auf Einfachheit gelegt, damit auch unerfahrene Benutzer die Umfragen verwalten können und die Webseiten auf mobilen Geräten nicht überladen wirken. Es ist möglich, die Kurslisten, wie sie bisher in Excel erfasst wurden, direkt in die Webapplikation zu importieren, sowie die ausgewerteten Ergebnisse als Excel Datei zu exportieren um sie weiter zu bearbeiten. Alle Kursteilnehmer melden sich mit einem gemeinsamen 6 stelligen Token auf der Hauptseite an und werden direkt auf die richtige Umfrage weitergeleitet. Die Verwendung eines aktuellen Web Application Frameworks sichert die Wartbarkeit des Produkts ab.

3.4 Ausblick

Es gibt mehrere kleinere Features die noch implementiert oder optimiert werden könnten:

- Automatische Zuordnung von Lehrpersonen beim Kursimport z.B. über Mail-Adressen
- Excel Export schöner formatieren und Formeln sowie Diagramme einfügen
- Mehr Diagramme bei den Auswertungen anzeigen

4 Einleitung

Diese Arbeit hat gemäss Aufgabenstellung zum Ziel, eine Webseite zur Unterrichtsbefragung zu entwickeln. Die Wartung nach der Arbeit, sowie deren Kostenrechnung musste nicht untersucht werden.

Die Applikation soll bestehende Prozesse optimieren. Dies umfasst die Verwaltung der Ressourcen und die Auswertung der Umfragen.

Die Applikation soll auf aktuellen und lizenzfreien Technologien aus dem Webbereich aufbauen. So kann sichergestellt werden, dass die Lösung auch zukünftigen Anforderungen gerecht werden kann.

Das Ausfüllen eines Fragebogens muss für möglichst viele Geräte optimiert sein, die Verwaltungsfunktionen müssen nur mit Desktop Geräten korrekt dargestellt werden.

5 Analyse Oda G ZH

Im folgenden Abschnitt werden die Vorgänge an der Oda G ZH so aufgeführt, wie sie zum Beginn unserer Bachelorarbeit durchgeführt wurden, mit den dadurch entstehenden Problemen.

5.1 Ist Szenarien

5.1.1 Kurse verwalten

Die Kurse werden in einem Word Dokument folgendermassen erfasst. Ein Kurs dauert immer nur einen Tag. Es gibt zwar Themen die mehrere Tage dauern, doch werden diese in mehrere eintägige Kurse aufgeteilt. In diesem Fall wird die Befragung erst am letzten Kurstag durchgeführt.

Monat	Datum	Zeit	Lehrperson	Thema	Hospitation durch
2014 April	07.04.2014	09.00 – 16.15	K. Kuhn	Ausscheidung, Inkontinenz	
	14.04.2014	13.15 – 16.15	B. Keller	Somatische und psychische Notfälle im Pflegealltag (1/2 Tag)	
	16.04.2014	09.00 – 16.15	J. Schmutz	Erkrankungen und Pflege Urogenitaltrakt/DK legen	
	17.04.2014	09.00 – 16.15	C. Artho	Labor, Laborwerte mit ausgewählten Krankheitsbildern	

Abbildung 5-1: Ausschnitt aus dem Word Dokument für die Kurserfassung

Zusätzlich werden die Teilnehmer eines Kurses in einem Excel Dokument mit folgendem Aufbau geführt. Als Beispiel wurde der erste Kurs aus dem Dokument, das in Abbildung 5-1 dargestellt ist, genommen.

Adressliste Teilnehmer/-innen LTT Praxis Oda G ZH							
Kurs:	Nr. 16 - Ausscheidung						
Ausb.jahr:	alle Sem.						
Datum:	4/7/2014						
Dozentin:	K. Kuhn						
Raum:	212						
Name	<input type="checkbox"/> Vorname	<input type="checkbox"/> BZ	<input type="checkbox"/> Semester	<input type="checkbox"/> Praktikumsort	<input type="checkbox"/> Email	<input type="checkbox"/> Bemerkungen	<input type="checkbox"/> Anzahl
Mustermann	Andrea	zag		2 Alter& Pflege Wthur	andrea.mustermann@win.ch		1
Muster	Alexandra	zag		2 Alter& Pflege Wthur	alexandra.muster@win.ch		1
Beispiel	Daniela	zag		2 Alter& Pflege Wthur	daniela.beispiel@win.ch		1
Fake	Marina	zag		2 Alter& Pflege Wthur	marina.fake@win.ch		1
TOTAL							4

Abbildung 5-2: Beispiel einer Teilnehmerliste

5.1.2 Fragebogen erstellen

Die Fragebogen werden ebenfalls im Word erstellt. Es gibt drei verschiedene Fragebogen (Studierende, Dozierende und Supervisoren) und drei verschiedene Typen von Fragen. In jedem Fragebogen gibt es Fragen, bei welchen man wissen will, wie sehr eine Aussage zutrifft. Für den Studierenden-Fragebogen wird bei diesem Fragetyp sogar noch eine Soll-Spalte verwendet, um die Erwartungen der Studierenden an den Kurs in Erfahrung zu bringen. Diese Daten werden für eine Masterarbeit von der Betriebsleiterin Brigitte Keller benötigt.

Bitte kreuzen Sie in jeder Zeile das Feld an, das Ihrer Meinung nach am ehesten zutrifft.		So war es					So sollte es sein				
		Trifft zu	Trifft eher zu	Trifft eher nicht zu	Trifft nicht zu	Keine Aussage möglich	Trifft zu	Trifft eher zu	Trifft eher nicht zu	Trifft nicht zu	Keine Aussage möglich
1.	Der LTT Praxis-Tag war insgesamt unterstützend für die Entwicklung meiner beruflichen Kompetenzen.										
2.	Ich erhielt die Möglichkeit mir meine (Lern)Fragen zu beantworten.										
3.	Der LTT Praxis-Tag ermöglichte mir, meine Erfahrungen aus dem Pflegealltag zu reflektieren.										
Bemerkungen zu den oben genannten Fragen:											

Abbildung 5-3: Fragetyp 1 (aus dem Teilnehmer-Fragebogen)

	Bitte kreuzen Sie in jeder Zeile das Feld an, das Ihrer Meinung nach am ehesten zutrifft.	Erfüllung meiner Erwartungen				
		Trifft zu	Trifft eher zu	Trifft eher nicht zu	Trifft nicht zu	Keine Aussage möglich
1.	Den LTT Praxis-Tag erachte ich <u>insgesamt</u> als gelungen.					
2.	Ich fühlte mich genügend kompetent, die Studierenden in ihrem Lernprozess zu unterstützen.					
3.	Ich ermöglichte den Studierenden ihre Erfahrungen zu reflektieren.					
Bemerkungen/Anregungen:						

Abbildung 5-4: Fragetyp 1 (aus dem Dozierenden-Fragebogen)

Ein zweiter Fragetyp hat mehrere Auswahlmöglichkeiten und wird momentan nicht für den Teilnehmer-Fragebogen verwendet.

15.	Am heutigen LTT Praxis benutzte ich die folgenden didaktisch-methodischen Lernformen	
	Bitte kreuzen Sie alle verwendeten Lernformen an	Bemerkungen
	Bearbeitung der persönlich mitgebrachten Fälle bzw. Frage- oder Problemstellungen der Studierenden	
	Bearbeitung eines von einem Studierenden mitgebrachten Falls bzw. Fragestellung	
	Bearbeitung von mir mitgebrachten Fallstudie/Fall-Beispiele oder Fragestellungen	
	Weitere:	

Abbildung 5-5: Fragetyp 2

Um allgemeine Fragen zu stellen, wird der dritte Fragetyp verwendet, bei welchem Platz für die Antwort des Befragten gelassen wird.

18. Bitte beantworten Sie als Abschluss noch die folgenden Fragen

A. Was hat Ihnen an diesem LTT Praxis am meisten Spass gemacht?

.....
.....
.....

B. Was könnte verändert werden, damit Sie den LTT Praxis noch besser nutzen können?

.....
.....
.....

Abbildung 5-6: Fragetyp 3

5.1.3 Umfragen durchführen

Eine Umfrage wird anonym durchgeführt, indem die verschiedenen Fragebogen am Ende eines Kurses ausgeteilt und danach wieder eingesammelt werden. Die Dozierenden und die Supervisoren geben den ausgefüllten Fragebogen innerhalb von zwei Wochen zurück.

5.1.4 Umfrage auswerten

Die ausgefüllten Fragebogen werden aufwändig ausgezählt und von Hand in ein Excel Dokument übertragen.

5.2 Probleme

Anhand der in Kapitel [Ist Szenarien](#) beschriebenen Szenarien wird ersichtlich, dass die aktuelle Lösung einige Probleme mit sich bringt. Nachfolgend werden die Probleme aufgeführt.

5.2.1 Inkonsistenz der Daten

Vergleicht man [Abbildung 5-1](#) und [Abbildung 5-2](#) fällt auf, dass das Datum in verschiedenen Formaten gespeichert wird. Das Feld 07.04.2014 wird zu 4/7/2014, was zu Verwechslungen führen kann.

5.2.2 Fehlerhafte Daten

In den Dokumenten schleichen sich häufig Fehler ein. In dem von uns betrachteten Dokument mit den Kursen hatte es Einträge für das Jahr 2013 sowie 2104, obwohl es nur eine Kursliste vom Jahr 2014 war.

5.2.3 Hoher Aufwand für die Auswertung notwendig

Da die Fragebogen von Teilnehmern und Dozenten auf Papier ausgefüllt werden, muss der Betriebsleiter diese mühsam von Hand auszählen, was einerseits sehr aufwendig ist, andererseits fehleranfällig.

5.2.4 Archivierung der Umfrage Resultate

Die ausgefüllten Umfragen brauchen viel Papier und sind mühsam zu archivieren. Das macht die einzelnen Umfragebogen schwer auffindbar. Auch das Durchstöbern von Excel Dokumenten ist sehr aufwendig.

6 Anforderungsanalyse

6.1 Funktionale Anforderungen

6.1.1 Rollen

In diesem Abschnitt werden verschiedene Rollen beschrieben, welche die Applikation TeachEval verwenden.

Teilnehmer

Ein Kursteilnehmer der über einen anonymen *Token* eine Umfrage am Tag der Kursdurchführung ausfüllen kann.

Dozierender

Ein Kursleiter der sich mit E-Mail und Passwort authentifiziert. Ihm ist es möglich, Umfragen bis zwei Wochen nach dem Kurs auszufüllen. Er kann anonyme Umfragen vor dem Kurs mit eigenen Fragen erweitern und danach deren Auswertung sehen.

Supervisor

Eine Aufsichtsperson die sich mit E-Mail und Passwort authentifiziert. Ihr ist es möglich Umfragen bis zwei Wochen nach dem Kurs auszufüllen.

Bereichsleiter

Eine Führungsperson die sich mit E-Mail und Passwort authentifiziert. Sie kann bei Kursen als Dozent oder Supervisor teilnehmen. Ihr ist es möglich Kurse, Fragebogen und Benutzer zu verwalten. Sie hat volle Einsicht bei Auswertungen und kann diese als Excel Datei exportieren.

6.1.2 Use cases

Die Use Cases werden gemäss des Buches "UML 2 und Patterns angewendet" von Craig Larman [Lar]. erstellt. In den Use Case Diagrammen wurde der include auf den UC 3 absichtlich weggelassen, da dieser von allen Use Cases der Benutzer benötigt wird (ausser UC5) und somit das Diagramm erheblich unübersichtlicher gestalten würde. In den Beschreibungen der Use Cases sind die Abhängigkeiten zu sehen. Einige für unsere Arbeit unwichtige Punkte der Use Cases wurden ebenfalls weggelassen.

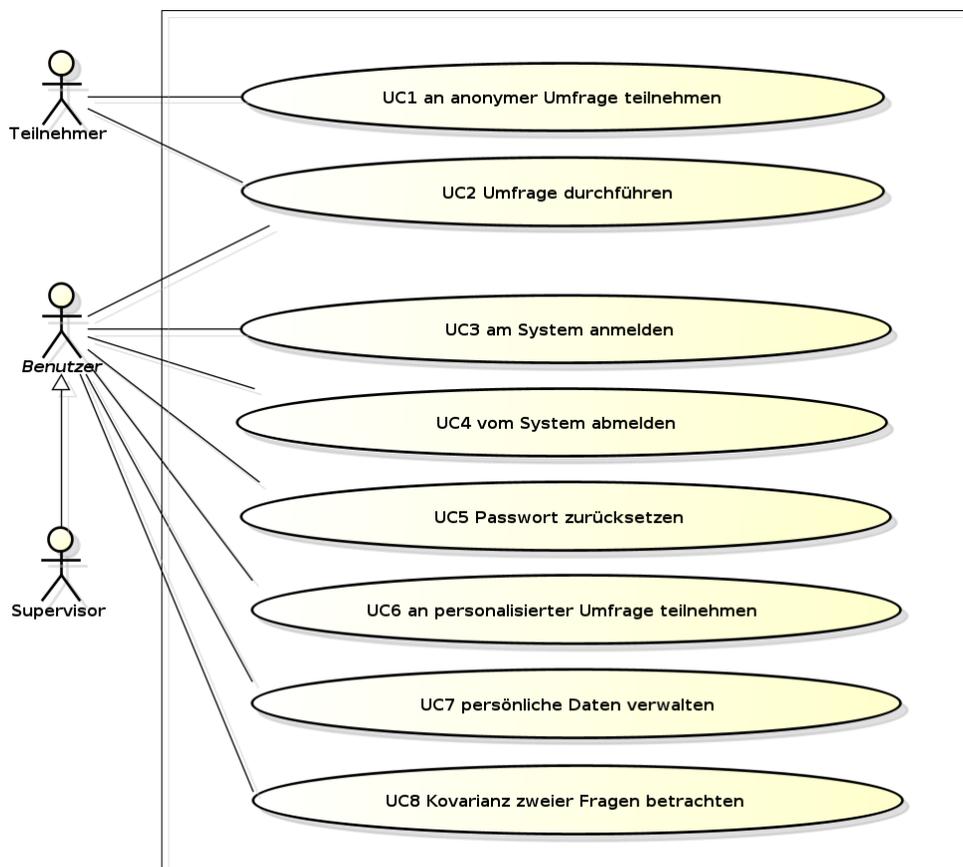


Abbildung 6-7: Use Case Diagramm Teil 1

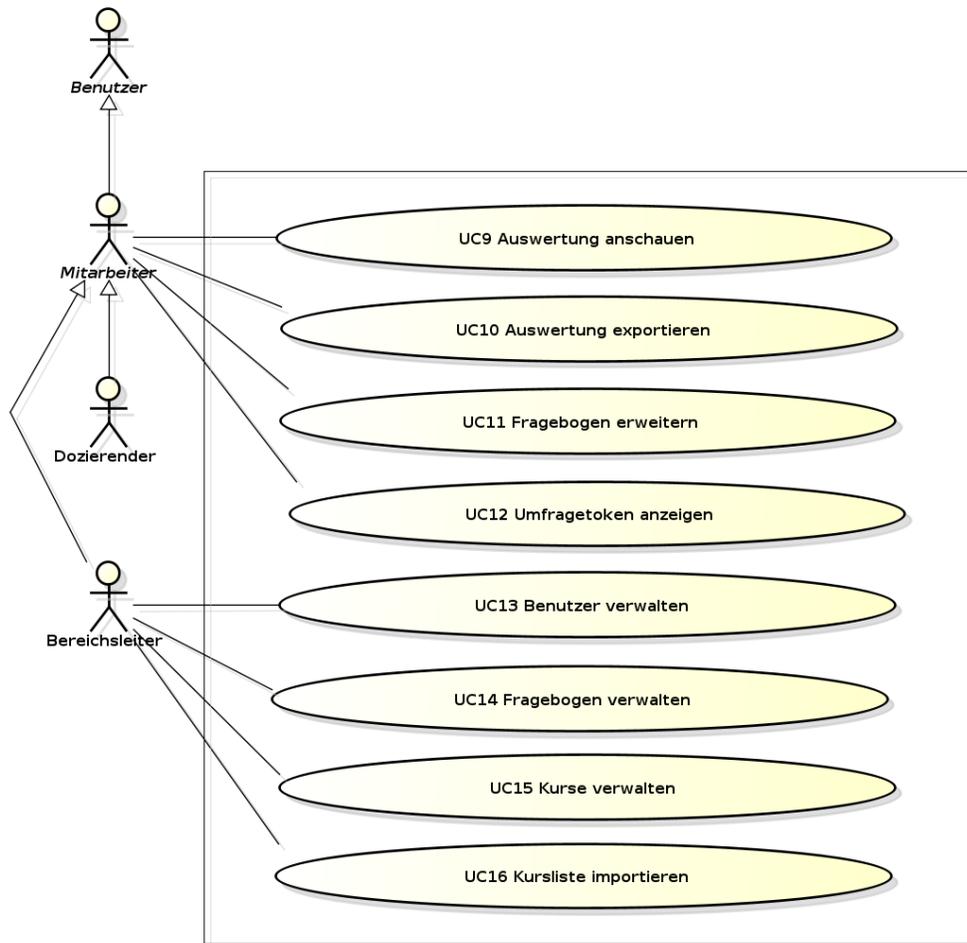


Abbildung 6-8: Use Case Diagramm Teil 2

6.1.3 UC1 an anonymer Umfrage teilnehmen

Primary Actor

- Teilnehmer
-

Stakeholders and Interests

- Teilnehmer: kann an einer Umfrage teilnehmen.
-

Preconditions

- Teilnehmer hat *Token* erhalten.
-

Postconditions

sind im [UC2 Umfrage durchführen](#) beschrieben.

Basic Flow

1. Teilnehmer gibt den *Token* auf der Startseite in das dafür vorgesehene Feld ein.
 2. Teilnehmer wird zum Fragebogen des mit dem *Token* verknüpften Kurses weitergeleitet.
 3. Teilnehmer führt Umfrage durch ->[UC2 Umfrage durchführen](#).
-

Alternative Flows

- 1.a Teilnehmer gibt ungültigen *Token* ein.
 - i) Ein Hinweis, dass der *Token* ungültig ist, wird eingeblendet.
 - ii) Teilnehmer wird nicht weitergeleitet.
-

Frequency of Occurrence

- nach jedem Kurs (ungefähr 50x pro Semester).
-

Tabelle 6-a: UC1 an anonymer Umfrage teilnehmen

6.1.4 UC2 Umfrage durchführen

Primary Actor

- Teilnehmer
 - Benutzer
-

Stakeholders and Interests

- Teilnehmer: kann den Dozierenden / Kurs beurteilen.
 - Dozierender: kann die Lernumgebung / OdA G ZH beurteilen.
 - Supervisor: kann den Dozierenden / Kurs beurteilen.
-

Preconditions

sind im [UC1 an anonymer Umfrage teilnehmen](#) beziehungsweise [UC6 an personalisierter Umfrage teilnehmen](#) beschrieben.

Postconditions

- Antworten sind in der Datenbank gespeichert.
 - Auswertung ist verfügbar.
-

Basic Flow

1. Teilnehmer / Benutzer füllt den Umfragebogen aus.
 2. Teilnehmer / Benutzer klickt auf den Button "Absenden".
 3. Umfragebogen ist korrekt ausgefüllt.
 4. Teilnehmer / Benutzer wird weitergeleitet und ihm wird bestätigt, dass die Teilnahme erfolgreich war.
-

Alternative Flows

3.a Die Umfrage wurde nicht komplett ausgefüllt.

- i) Ein Hinweis, welche Fragen nicht beantwortet wurden, wird eingeblendet.
- ii) Teilnehmer / Benutzer kann die Fragen entsprechend anpassen
>Schritt 1

Frequency of Occurrence

- nach jedem Kurs (ungefähr 50x pro Semester) multipliziert mit der Anzahl zu Befragenden (ungefähr 15)
-

Tabelle 6-b: UC2 Umfrage durchführen

6.1.5 UC3 am System anmelden

Primary Actor

- Benutzer
-

Stakeholders and Interests

- Benutzer: kann sich am System anmelden.
-

Preconditions

- Benutzer wurde erfasst.
 - Benutzer ist nicht am System angemeldet.
-

Postconditions

- Benutzer ist am System angemeldet.
-

Basic Flow

1. Benutzer gibt seine E-Mail-Adresse und sein Passwort in die entsprechenden Felder ein.
 2. Anmeldeinformationen sind korrekt.
 3. Benutzer wird in den persönlichen Bereich weitergeleitet.
-

Alternative Flows

- 2.a Anmeldeinformationen sind nicht korrekt.
 - i) Ein Hinweis, dass die eingegebenen Daten falsch sind, wird eingeblendet.
 - ii) Teilnehmer wird nicht weitergeleitet.
-

Frequency of Occurrence

- täglich
-

Tabelle 6-c: UC3 am System anmelden

6.1.6 UC4 vom System abmelden

Primary Actor

- Benutzer
-

Stakeholders and Interests

- Benutzer: kann sich vom System abmelden.
-

Preconditions

- Benutzer ist am System angemeldet.
-

Postconditions

- Benutzer ist nicht mehr am System angemeldet.
-

Basic Flow

1. Benutzer klickt auf den "Logout" Button.
 2. Benutzer wird vom System abgemeldet.
 3. Benutzer wird zum Login weitergeleitet.
-

Alternative Flows

Frequency of Occurrence

- täglich
-

Tabelle 6-d: UC4 vom System abmelden

6.1.7 UC5 Passwort zurücksetzen

Primary Actor

- Benutzer
-

Stakeholders and Interests

- Benutzer: kann ein neues Passwort anfordern um sich wieder am System anmelden zu können.
-

Preconditions

Postconditions

- Benutzer bekommt ein neues Passwort.
-

Basic Flow

1. Benutzer klickt auf den Link "Passwort vergessen".
 2. Benutzer wird aufgefordert seine Emailadresse einzugeben.
 3. Ein Hinweis, dass das Passwort gesendet wurde, wird angezeigt.
 4. Benutzer erhält per E-Mail ein neues Passwort.
-

Alternative Flows

- 2.a Die eingegebene E-Mail-Adresse war eine dem System nicht bekannte E-Mail-Adresse.
- i) Ein Hinweis, dass die E-Mail-Adresse nicht gefunden wurde, wird eingeblendet.
 - ii) Benutzer kann die E-Mail-Adresse erneut eingeben.
- 2.b Die eingegebene E-Mail-Adresse war eine E-Mail-Adresse eines anderen Benutzers.
- i) Ein Hinweis, dass das Passwort gesendet wurde, wird angezeigt.
 - ii) Benutzer der eingegebenen E-Mail-Adresse bekommt ein neues Passwort zugeschickt. (altes Passwort ist gültig bis man das neue einmal eingibt).

Frequency of Occurrence

- täglich
-

Tabelle 6-e: UC5 Passwort zurücksetzen

6.1.8 UC6 an personalisierter Umfrage teilnehmen

Primary Actor

- Benutzer
-

Stakeholders and Interests

- Benutzer: kann an einer Umfrage teilnehmen.
-

Preconditions

- Benutzer ist am System angemeldet.
 - Kurs wurde erstellt.
-

Postconditions

sind im [UC2 Umfrage durchführen](#) beschrieben.

Basic Flow

1. Benutzer klickt auf den Navigationspunkt "Teilnehmen".
 2. Benutzer werden alle Kurse, an denen er teilnimmt, angezeigt.
 3. Benutzer klickt auf den Button "teilnehmen" des entsprechenden Kurses.
 4. Benutzer wird auf die entsprechende Umfrage weitergeleitet.
 5. Benutzer führt Umfrage durch ->[UC2 Umfrage durchführen](#).
-

Alternative Flows

- 2.a Es werden keine Kurse angezeigt, obwohl ein Kurs stattfindet.
 - i) Benutzer wurde dem Kurs nicht zugeteilt oder das Startdatum wurde falsch erfasst. ->[UC15 Kurse verwalten \(CRUD\)](#).
 - 4.a Die Umfrage kann nicht geladen werden.
 - i) Dem Kurs wurde kein Fragebogen zugeteilt. ->[UC15 Kurse verwalten \(CRUD\)](#).
-

Frequency of Occurrence

- nach jedem Kurs (ungefähr 50x pro Semester)
-

Tabelle 6-f: UC6 an personalisierter Umfrage teilnehmen

6.1.9 UC7 persönliche Daten verwalten

Primary Actor

- Benutzer
-

Stakeholders and Interests

- Benutzer: kann seine Daten verändern.
-

Preconditions

- Benutzer ist am System angemeldet.
-

Postconditions

- Benutzerdaten sind aktualisiert.
-

Basic Flow

1. Benutzer klickt auf den Navigationspunkt "Profil".
 2. Benutzer ändert die Angaben wunschgemäss.
 3. Benutzer klickt auf den Button "Aktualisieren".
 4. Daten sind korrekt.
 5. Ein Hinweis, dass die Daten aktualisiert wurden, wird eingeblendet.
-

Alternative Flows

- 4.a Daten sind nicht korrekt.
 - i) Ein Hinweis, welche Felder nicht korrekt ausgefüllt wurden, wird eingeblendet.
 - ii) Benutzer kann die Felder entsprechend anpassen ->Schritt 2
-

Frequency of Occurrence

- first
-

Tabelle 6-g: UC7 persönliche Daten verwalten

6.1.10 UC8 *Kovarianz* zweier Fragen betrachten

Primary Actor

- Benutzer
-

Stakeholders and Interests

- Benutzer: kann Auswertungen von Kursen, die ihn betreffen, sowie Statistiken einsehen.
 - Bereichsleiter: kann Auswertungen aller Kurse einsehen.
-

Preconditions

- Benutzer ist am System angemeldet.
 - Mindestens eine Umfrage eines Kurses mit der Beteiligung der Teilnehmer und des Benutzers wurde durchgeführt.
 - Benutzer betrachtet eine Auswertung
-

Postconditions

Basic Flow

1. Benutzer klickt auf den Navigationspunkt "Kovarianzrechner".
 2. Benutzer wählt zwei Fragen aus.
 3. Dem Benutzer wird die Kovarianz der zwei Fragen angezeigt.
-

Alternative Flows

Frequency of Occurrence

- monatlich
-

Tabelle 6-h: UC8 Kovarianz zweier Fragen betrachten

6.1.11 UC9 Auswertung anschauen

Primary Actor

- Benutzer
-

Stakeholders and Interests

- Benutzer: kann Auswertungen von Kursen, die ihn betreffen, sowie Statistiken einsehen.
 - Bereichsleiter: kann Auswertungen aller Kurse einsehen.
-

Preconditions

- Benutzer ist am System angemeldet.
 - Mindestens ein Kurs mit der Beteiligung des Benutzers wurde durchgeführt.
-

Postconditions

Basic Flow

1. Benutzer klickt auf den Navigationspunkt "Auswertungen".
 2. Benutzer wählt einen Kurs aus.
 3. Dem Benutzer werden Auswertungen zum gewählten Kurs gezeigt.
-

Alternative Flows

Frequency of Occurrence

- nach jedem Kurs (ungefähr 50x pro Semester)
-

Tabelle 6-i: UC9 Auswertung anschauen

6.1.12 UC10 Auswertung exportieren

Primary Actor

- Bereichsleiter
-

Stakeholders and Interests

- Bereichsleiter: kann Auswertung eines Kurses als Excel-Dokument exportieren lassen.
-

Preconditions

- Bereichsleiter ist am System angemeldet.
 - Mindestens eine Umfrage des Kurses wurde durchgeführt.
-

Postconditions

Basic Flow

1. Benutzer klickt auf den Navigationspunkt "Auswertungen".
 2. Benutzer wählt einen Kurs aus.
 3. Benutzer klickt auf den Navigationspunkt "Excel Datei generieren".
 4. Download der Excel Datei startet.
-

Alternative Flows

Frequency of Occurrence

- nach jedem Kurs (ungefähr 50x pro Semester)
-

Tabelle 6-j: UC10 Auswertung exportieren

6.1.13 UC11 Fragebogen erweitern

Primary Actor

- Dozierender
-

Stakeholders and Interests

- Dozierender: kann einen Anonymen Fragebogen, der von einem Bereichsleiter erstellt wurde, erweitern.
-

Preconditions

- Benutzer ist am System angemeldet.
-

Postconditions

- Erweiterter Fragebogen ist im System vorhanden.
-

Basic Flow

1. Der Dozierende klickt auf den Navigationspunkt "Umfrage erweitern".
 2. Dem Dozierenden wird eine Liste der Kurse angezeigt.
 3. Der Dozierende wählt einen Kurs aus.
 4. Der Dozierende fügt seine eigenen Fragen zum Fragebogen hinzu.
 5. Die Felder sind ordnungsgemäss ausgefüllt.
 6. Der Dozierende klickt auf den Button "Speichern".
-

Alternative Flows

- 5.a Die Felder sind nicht ordnungsgemäss ausgefüllt.
 - i) Ein Hinweis, welche Felder nicht korrekt ausgefüllt wurden, wird eingeblendet.
 - ii) Der Dozierende kann die Felder entsprechend anpassen ->Schritt 4
-

Frequency of Occurrence

- monatlich
-

Tabelle 6-k: U11 Fragebogen erweitern

6.1.14 UC12 Umfragetoken anzeigen

Primary Actor

- Dozierender
-

Stakeholders and Interests

- Dozierender: kann Umfragetoken den Teilnehmern zeigen, damit diese Zugang zur Umfrage bekommen.
-

Preconditions

- Benutzer ist am System angemeldet.
 - Kurs wurde erstellt.
 - Benutzer wurde dem Kurs zugeteilt.
-

Postconditions

Basic Flow

1. Benutzer klickt auf den Navigationspunkt "Teilnehmen".
 2. Eine Liste aus dem Benutzer zugeteilten Kurse erscheint.
 3. Benutzer klickt auf den Link "anzeigen" für den gewünschten Kurs.
 4. Der dazu passende Umfragetoken wird angezeigt.
-

Alternative Flows

Frequency of Occurrence

- nach jedem Kurs (ungefähr 50x pro Semester)
-

Tabelle 6-1: UC12 Umfragetoken anzeigen

6.1.15 UC13 Benutzer verwalten (*CRUD*)

Primary Actor

- Betriebsleiter
-

Stakeholders and Interests

- Betriebsleiter: kann einen Benutzer im System einsehen, erstellen, löschen oder verändern.
-

Preconditions

- Betriebsleiter ist im System angemeldet.
 - **Read, Update, Delete:** Es ist mindestens ein Benutzer ausser dem Betriebsleiter im System vorhanden.
-

Postconditions

- **Create:** Neuer Benutzer ist im System vorhanden.
 - **Update:** Veränderungen am Benutzer sind persistent gespeichert.
 - **Delete:** Benutzer ist gelöscht.
-

Basic Flow

1. Bereichsleiter klickt auf den Navigationspunkt "Benutzerverwaltung".

Create

2. Bereichsleiter klickt auf den Menüpunkt "Benutzer erstellen"
3. Bereichsleiter erfasst Daten für den Benutzer.
4. Bereichsleiter klickt auf den Button "Erstellen".
5. Daten sind korrekt.
6. E-Mail mit Passwort wird an die E-Mail-Adresse des neuen Benutzer geschickt.
7. System speichert den Benutzer.
8. Bereichsleiter wird zur Übersicht der Benutzer weitergeleitet.

Read

2. Bereichsleiter sieht alle Benutzerdaten der erfassten Benutzer in einer Tabelle.

Update

2. Bereichsleiter klickt in der Tabelle bei dem gewünschten Benutzer auf den Link "ändern".
3. Daten des Benutzers werden angezeigt
4. Bereichsleiter verändert Daten für den Benutzer.
5. Bereichsleiter klickt auf den Button "Aktualisieren".
6. Daten sind korrekt.
7. System speichert die Änderungen.
8. Bereichsleiter wird zur Übersicht der Benutzer weitergeleitet.

Delete

2. Bereichsleiter klickt in der Tabelle bei dem gewünschten Benutzer auf den Link "löschen".
 3. Bereichsleiter bestätigt Frage ob dieser Benutzer wirklich gelöscht werden soll.
 4. System löscht ausgewählten Benutzer.
 5. Bereichsleiter wird zur Übersicht der Benutzer weitergeleitet.
-

Alternative Flows

Create

- 5.a Die Angaben sind nicht korrekt oder leer.
 - i) Ein Hinweis, welche Felder nicht korrekt ausgefüllt wurden, wird eingeblendet.
 - ii) Bereichsleiter kann die Felder entsprechend anpassen ->Schritt 3.

Update

- 5.a Die Angaben sind nicht korrekt oder leer.
 - i) Ein Hinweis, welche Felder nicht korrekt ausgefüllt wurden, wird eingeblendet.
 - ii) Bereichsleiter kann die Felder entsprechend anpassen ->Schritt 4.

Delete

- 3.a Bereichsleiter bestätigt den Löschvorgang nicht
 - i) Bereichsleiter wird zur Übersicht der Benutzer umgeleitet.
-

Frequency of Occurrence

- monatlich
-

6.1.16 UC14 Fragebogen verwalten (*CRUD*)

Primary Actor

- Betriebsleiter
-

Stakeholders and Interests

- Betriebsleiter: kann einen Fragebogen im System einsehen, erstellen, löschen oder verändern.
-

Preconditions

- Betriebsleiter ist im System angemeldet.
 - **Read, Update, Delete:** Es ist mindestens ein Fragebogen im System vorhanden.
-

Postconditions

- **Create:** Neuer Fragebogen ist im System vorhanden.
 - **Update:** Veränderungen am Fragebogen sind persistent gespeichert.
 - **Delete:** Fragebogen ist gelöscht.
-

Basic Flow

1. Bereichsleiter klickt auf den Navigationspunkt "Fragebogenverwaltung".

Create

2. Bereichsleiter klickt auf den Menüpunkt "Fragebogen erstellen"
3. Bereichsleiter erstellt einen Fragebogen.
4. Bereichsleiter klickt auf den Menüpunkt "Speichern".
5. Daten sind korrekt.
6. System speichert den Fragebogen.
8. Bereichsleiter wird zur Übersicht der Fragebogen weitergeleitet.

Read

2. Bereichsleiter sieht alle erfassten Fragebogen in einer Tabelle.
3. Bereichsleiter klickt auf den Link "bearbeiten" des gewünschten Fragebogens.
4. Der Fragebogen wird angezeigt.

Update

2. Bereichsleiter klickt in der Tabelle bei dem gewünschten Fragebogen auf den Link "ändern".
3. Der Fragebogen wird angezeigt.
4. Bereichsleiter nimmt Änderungen am Fragebogen vor.
5. Bereichsleiter klickt auf den Menüpunkt "Aktualisieren".
6. Daten sind korrekt.
7. System speichert die Änderungen.
8. Bereichsleiter wird zur Übersicht der Fragebogen weitergeleitet.

Delete

2. Bereichsleiter klickt in der Tabelle bei dem gewünschten Fragebogen auf den Link "löschen".
3. Bereichsleiter bestätigt Frage ob dieser Fragebogen wirklich gelöscht werden soll.
4. System löscht ausgewählten Fragebogen.
5. Bereichsleiter wird zur Übersicht der Fragebogen weitergeleitet.

Alternative Flows

Create

- 5.a Die Angaben sind nicht korrekt oder leer.
 - i) Ein Hinweis, welche Felder nicht korrekt ausgefüllt wurden, wird eingeblendet.
 - ii) Bereichsleiter kann die Felder entsprechend anpassen ->Schritt 3

Update

- 5.a Die Angaben sind nicht korrekt oder leer.
 - i) Ein Hinweis, welche Felder nicht korrekt ausgefüllt wurde, wird eingeblendet.
 - ii) Bereichsleiter kann die Felder entsprechend anpassen ->Schritt 4

Delete

- 3.a Bereichsleiter bestätigt den Löschvorgang nicht
 - i) Bereichsleiter wird zur Übersicht der Fragebogen umgeleitet.

Frequency of Occurrence

- halbjährlich
-

Tabelle 6-n: UC14 Fragebogen verwalten (CRUD)

6.1.17 UC15 Kurse verwalten (*CRUD*)

Primary Actor

- Betriebsleiter
-

Stakeholders and Interests

- Betriebsleiter: kann einen Kurs im System einsehen, erstellen, löschen oder verändern.
-

Preconditions

- Betriebsleiter ist im System angemeldet.
 - **Read, Update, Delete:** Es ist mindestens ein Kurs im System vorhanden.
-

Postconditions

- **Create:** Neuer Kurs ist im System vorhanden.
 - **Update:** Veränderungen am Kurs sind persistent gespeichert.
 - **Create / Update:** Mindestens ein Fragebogen und ein Benutzer pro Benutzergruppe ist vorhanden (Supervisor optional).
 - **Delete:** Kurs ist gelöscht.
-

Basic Flow

1. Bereichsleiter klickt auf den Navigationspunkt "Kursverwaltung".

Create

2. Bereichsleiter klickt auf den Menüpunkt "Kurs erstellen"
3. Bereichsleiter erfasst Daten für den Kurs.
4. Bereichsleiter klickt auf den Button "Erstellen".
5. Daten sind korrekt.
6. System speichert den Kurs.
7. Bereichsleiter wird zur Übersicht der Benutzer weitergeleitet.

Read

2. Bereichsleiter sieht alle Kurse in einer Tabelle.
3. Bereichsleiter klickt auf den Link "bearbeiten" des gewünschten Kurses.
4. Die Kursdaten werden angezeigt.

Update

2. Bereichsleiter klickt in der Tabelle bei dem gewünschten Kurs auf den Link "ändern".
3. Bereichsleiter erfasst Daten für den Kurs.
4. Bereichsleiter klickt auf den Button "Aktualisieren".
5. Daten sind korrekt.
6. System speichert die Änderungen.
7. Bereichsleiter wird zur Übersicht der Kurse weitergeleitet.

Delete

2. Bereichsleiter klickt in der Tabelle bei dem gewünschten Kurs auf den Link "löschen".
3. Bereichsleiter bestätigt Frage ob dieser Kurs wirklich gelöscht werden soll.
4. System löscht ausgewählten Kurs.
5. Bereichsleiter wird zur Übersicht der Benutzer weitergeleitet.

Alternative Flows

Create

- 5.a Die Angaben sind nicht korrekt oder leer.
 - i) Ein Hinweis, welche Felder nicht korrekt ausgefüllt wurden, wird eingeblendet.
 - ii) Bereichsleiter kann die Felder entsprechend anpassen ->Schritt 3

Update

- 5.a Die Angaben sind nicht korrekt oder leer.
 - i) Ein Hinweis, welche Felder nicht korrekt ausgefüllt wurden, wird eingeblendet.
 - ii) Bereichsleiter kann die Felder entsprechend anpassen ->Schritt 4

Delete

- 3.a Bereichsleiter bestätigt den Löschvorgang nicht
 - i) Bereichsleiter wird zur Übersicht der Kurse umgeleitet.

Frequency of Occurrence

- monatlich

Tabelle 6-o: UC15 Kurse verwalten (CRUD)

6.1.18 UC16 Kursliste importieren

Primary Actor

- Bereichsleiter
-

Stakeholders and Interests

- Bereichsleiter: kann Kurse aus einem Excel-Dokument importieren.
-

Preconditions

- Bereichsleiter ist im System angemeldet.
-

Postconditions

- Kurse sind im System persistent gespeichert.
-

Basic Flow

1. Benutzer klickt auf den Navigationspunkt "Kursverwaltung".
 2. Benutzer klickt auf den Menüpunkt "Aus Excel importieren".
 3. Die Daten im Excel-Dokument sind korrekt.
 4. System fügt Kurse hinzu
 5. Hinzugefügte Kurse werden in der Tabelle angezeigt.
 6. Betriebsleiter fügt Fragebogen etc. den Kursen hinzu -/textgreater [UC15](#)
[Kurse verwalten \(CRUD\)](#)
-

Alternative Flows

- 2.a Die Daten im Excel-Dokument sind nicht korrekt.
 - i) Betriebsleiter wird zur Übersicht der Kurse weitergeleitet.
-

Frequency of Occurrence

- halbjährlich
-

Tabelle 6-p: UC16 Kursliste importieren

6.2 Nichtfunktionale Anforderungen

6.2.1 Mengenanforderungen

An den Kursen im Bereich Psychiatrie nehmen maximal 20 Teilnehmer teil und es finden maximal zwei Kurse am gleichen Tag statt (Stand 2014). Weiterhin sollte es möglich sein, ungefähr 25 Dozierende, 5 Supervisoren und einen Bereichsleiter anzulegen und zu verwalten.

6.2.2 Leistungsanforderungen

Der einzige Usecase, der eine grössere Anzahl Teilnehmer hat und dadurch eine grössere Anzahl gleichzeitiger Verbindungen entstehen, ist die anonyme Teilnahme. Der Fragebogen soll bei 25 gleichzeitigen Teilnehmern jeweils in maximal 5 Sekunden geladen werden.

6.2.3 Qualitätsmerkmale

Die Qualitätsmerkmale werden nach ISO/IEC 9126¹ spezifiziert. Qualitätsmerkmale welche für unsere Webapplikation nicht von Wichtigkeit sind, werden nicht aufgeführt.

Funktionalität

Angemessenheit

Funktionen sollen so eingeschränkt werden, dass sie auch auf mobilen Geräten sinnvoll sind. Die Benutzer sollen zum Beispiel auch auf einem mobilen Gerät angezeigt werden können. Hingegen muss das Verwalten der Benutzer, welches nur selten durchgeführt wird, auf mobilen Geräten nicht ausgeführt werden können.

Richtigkeit

Die Auswertungen sollen zu jeder Zeit den aktuellen Stand reflektieren. Nur Umfragen für Kurse, die auch gerade durchgeführt werden, dürfen gestartet werden.

Interoperabilität

Die Applikation soll im Browser auf Desktop-Computern, Smartphones und Tablets lauffähig sein. Das Layout soll so entwickelt werden, dass die Darstellung auf die Auflösung / Bildschirmgröße des jeweiligen Gerätes angepasst wird (Responsive Design). Auswertungen sollen im Format xlsx exportiert werden können, das mit entsprechender Software auf gängigen Betriebssystemen geöffnet werden kann.

Sicherheit

Die Verwendung der Applikation soll keine Sicherheitsdefizite verursachen. Benutzer mit einem Account haben über den Login Zugriff auf das System. TeilnehmerInnen können nur an Umfragen teilnehmen. Über die verschiedenen Rollen sollen die Funktionalitäten sowie Berechtigungen zugeteilt werden. Die Datenübertragung soll mittels TLS sichergestellt werden.

¹ http://de.wikipedia.org/wiki/ISO/IEC_9126

Zuverlässigkeit

Reife

Es soll zu 95 Prozent eine korrekte Kommunikation mit dem Server zustande kommen, wenn die Internetanbindung funktioniert.

Fehlertoleranz

Bei einer Falscheingabe soll keine Aktion vorgenommen werden und der Benutzer dementsprechend benachrichtigt werden.

Wiederherstellbarkeit

Bei einem Absturz der Datenbank oder der Applikation sollen die Daten innerhalb von zwei Stunden wiederhergestellt werden können.

Benutzbarkeit

Verständlichkeit

Die grafische Oberfläche soll einfach zu bedienen sein. Es sollen bereits bekannte Konzepte verwendet werden. Um die Verständlichkeit zu testen, wird die Applikation laufend mit im Projekt nicht involvierten Personen getestet und verbessert.

Erlernbarkeit

Die Applikation soll so gestaltet werden, dass auf jeder Seite der gleiche Aufbau verwendet wird und um zu den Hauptfunktionen zu gelangen nur zwei Klicks benötigt werden.

Bedienbarkeit

Die Applikation soll mit der Maus sowie auch mit Touchscreens bedienbar sein.

Attraktivität

Die grafische Oberfläche der Applikation soll schlicht und nicht überladen wirken. Zudem soll sie Ähnlichkeiten mit der Hauptseite der Oda G ZH aufweisen und einen professionellen Eindruck hinterlassen.

Effizienz

Zeitverhalten

Die Webapplikation überträgt nur eine geringe Menge von Daten. Deshalb sollte es möglich sein, die maximale Antwortzeit der Applikation auf <500ms zu optimieren. Diese Zeit ist abhängig von der Internetanbindung. Für die Überprüfung wird von einer Verbindung von mindestens 10Mbps ausgegangen. Express logt bei jedem Aufruf einer Seite, wie lange die Antwortzeit ist. Diese Werte werden von uns laufend kontrolliert.

Verbrauchsverhalten

Auf einigen mobilen Geräten ist der Speicher, den die Webanwendung für Javasript benutzen darf, beschränkt. Damit keine Probleme auftauchen sollten die Funktionen, die auf mobilen Geräten verfügbar sind, weniger als 10 MB benutzen.

Wartbarkeit

Analysierbarkeit

Auftretende Fehler sollen auf Serverseite so geloggt werden, dass die Ursache und der Zeitpunkt des Fehlers klar ersichtlich sind.

Modifizierbarkeit

Die Applikation soll so programmiert und dokumentiert werden, dass Änderungen an der Client- oder Serverlogik innerhalb von einem Arbeitstag implementiert werden können. Desweiteren sollen Module so eingesetzt werden, dass sie durch andere ausgetauscht werden können.

Stabilität

Die Anwendung sollte 95 Prozent der Zeit laufen (Downtime wegen Updates oder Wartung). Bei einem Fehler oder einem Neustart des Systems, soll die Applikation automatisch neugestartet werden.

Testbarkeit

Die Architektur auf dem Server soll nach MVC entwickelt werden um das automatische Testen zu ermöglichen. Ein automatisches Testen der grafischen Oberfläche wird nicht durchgeführt und wird von Hand mit verschiedenen zu unterstützenden Browsern getestet.

Übertragbarkeit

Anpassbarkeit

Es sollen nur opensource und plattformunabhängige Komponenten eingesetzt werden, damit die Applikation auf allen modernen Betriebssystemen eingesetzt werden kann.

Installierbarkeit

Auf der Clientseite ist keine Installation notwendig, da die Applikation im Browser ausgeführt wird und standardmässig schon ein Browser vorinstalliert ist. Auf der Serverseite soll die Applikation innerhalb von zwei Stunden auf allen modernen Systemen installiert sowie die Umgebung eingerichtet werden können.

Koexistenz

Die Koexistenz kann auf der Serverseite ignoriert werden, da die Applikation in einem Browser läuft. Auf der Serverseite soll die Applikation so konfiguriert werden können, dass sie ohne Probleme neben einer anderen Applikation laufen kann. Dies ist sowieso der Fall, da die Applikation der zweiten Gruppe, die zu diesem Thema eine Arbeit schreibt, ebenfalls auf demselben Server laufen wird.

6.3 Randbedingungen

Im folgenden Abschnitt werden Einschränkungen für den Server und den Client gemacht, damit Entscheidungen bezüglich einzusetzender Versionen von z.B. Bootstrap und jQuery, getroffen werden können.

6.3.1 Server

Die Serversoftware ist frei verfügbar und Cross-platform fähig.

- Betriebssystem: Linux, Mac OS X, Windows, Solaris
- Webserver: Node.js \geq v0.10.26
- Datenbank: mongoDB \geq v2.4.9
- Module:
 - async \geq v0.8.0
 - bcrypt \geq v0.7.7
 - body-parser \geq v1.2.0
 - connect-multiparty \geq v1.0.3
 - console-stamp \geq v0.1.3
 - express \geq v3.4.8
 - forever \geq v0.10.11
 - jade \geq v1.3.0
 - mongoose \geq v3.8.8
 - node-mailer \geq v0.6.3
 - node-xlsx \geq v0.4.0
 - password-generator \geq v0.2.2

Die oben aufgelisteten Mindestversionen sind die Versionen, mit denen die Applikation entwickelt wurde.

6.3.2 Client

Aufgrund der vielen verschiedenen Geräten, die auf unsere Applikation zugreifen, sollte nicht zu sehr eingeschränkt werden. Alle Smartphones, Tablets, Desktop-Computer der letzten fünf Jahre zuzüglich Windows XP sollten unterstützt werden. Auf Smartphones muss nur die Umfrage fehlerfrei dargestellt werden können. Um dies zu erreichen, wird die Umfrage so gestaltet, dass sie auch mit dem in mobile-Browser üblichen 320px dargestellt werden kann. Unter Berücksichtigung der aktuell noch eingesetzten Browser [w3s] wird eine Liste der zu unterstützenden Browser erstellt.

Browser Statistics



2014	Chrome	Internet Explorer	Firefox	Safari	Opera
April	58.4 %	9.4 %	25.0 %	4.0 %	1.8 %

Google Chrome Statistics



The following table is a breakdown of the Google Chrome numbers from our [Browsers Statistics](#):

2014	Total	C 34	C 33	C 32	C 31	C 30	Older
April	58.4%	31.7%	21.4%	1.0%	0.9%	0.3%	3.1%

2013	Total	C 32	C 31	C 30	C 29	C 28	C 27	C 26	C 25	C 24	C 23	Older
December	55.8%	1.0%	49.0%	1.6%	0.8%	0.8%	0.6%	0.3%	0.2%	0.1%	0.2%	1.2%
November	54.8%	0.5%	26.3%	23.7%	1.1%	1.0%	0.6%	0.4%	0.3%	0.2%	0.2%	0.5%

2012	Total	C 25	C 24	C 23	C 22	C 21	C 20	C 19	C 18	C 17	C 16	Older
December	46.9%	0.4%	0.7%	41.7%	1.1%	0.5%	0.5%	0.2%	0.5%	0.2%	0.1%	1.0%
November	46.3%	0.3%	0.6%	30.3%	10.7%	1.1%	0.7%	0.4%	0.7%	0.3%	0.2%	1.0%

Firefox Statistics



The following table is a breakdown of the Firefox numbers from our [Browsers Statistics](#):

2014	Total	FF 29	FF 28	FF 27	FF 26	FF 25	Older
April	25.0%	1.1%	18.1%	1.2%	0.7%	0.3%	3.6%

2013	Total	FF 26	FF 25	FF 24	FF 23	FF 22	FF 21	FF 20	FF 19	FF 18	FF 17	Older
December	26.8%	6.9%	13.3%	1.1%	0.5%	0.4%	0.3%	0.4%	0.2%	0.2%	0.5%	3.0%
November	26.8%	1.0%	16.2%	3.5%	0.7%	0.5%	0.4%	0.4%	0.3%	0.2%	0.7%	2.9%

2012	Total	FF 17	FF 16	FF 15	FF 14	FF 13	FF 12	FF 11	FF 9-10	FF 4-8	FF 3.6	Older
December	31.1%	16.3%	5.6%	1.3%	1.0%	0.7%	1.1%	0.6%	1.2%	1.2%	0.8%	1.3%
November	31.2%	2.5%	19.4%	1.8%	1.2%	0.8%	1.2%	0.6%	1.3%	1.3%	0.8%	0.3%

Internet Explorer Statistics



The following table is a breakdown of the Internet Explorer numbers from our [Browsers Statistics](#):

2014	Total	IE 11	IE 10	IE 9	IE 8	IE 7	IE 6
April	9.4 %	2.7 %	1.6 %	2.0 %	2.4 %	0.3 %	0.1 %

Safari Statistics



The following table is a breakdown of the Safari numbers from our [Browsers Statistics](#):

2014	Total	S 7	S 6	S 5
April	4.0 %	2.8 %	0.8 %	0.4 %

Opera Statistics



The following table is a breakdown of the Opera numbers from our [Browsers Statistics](#):

2014	Total	O 20	O 19	O 18	O 12	O Mini	Other
April	1.8 %	0.6 %	0.0 %	0.0 %	0.4 %	0.8 %	0.0 %

Abbildung 6-9: Statistiken der Browser von w3schools

Aufgrund der Statistiken in Abbildung 6-9 wurden folgende Minimalversionen festgelegt:

- Google Chrome ab Version 18
- Mozilla Firefox ab Version 4
- Internet Explorer ab Version 7
- Safari ab Version 5
- Opera ab Version 12

Da ältere Browserversionen mitunterstützt werden, muss bei jQuery auf Version 2 verzichtet werden.

7 Objektorientierte Analyse

7.1 Domainmodell

Das Modell soll einen Überblick über die verschiedenen Objekte einer Befragung geben.

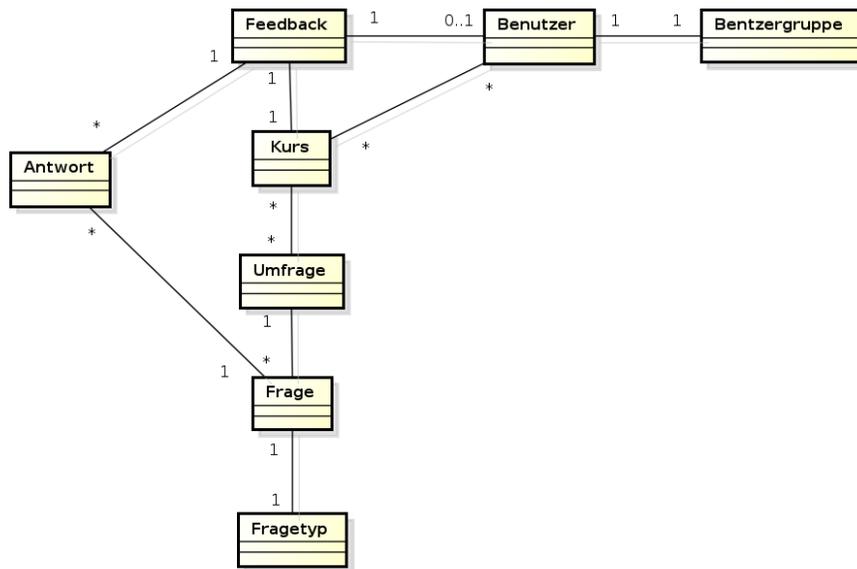


Abbildung 7-10: Domainmodell

7.1.1 Benutzer

Beispiel: Brigitte Keller
Ein Angestellter der Oda G ZH.

7.1.2 Benutzergruppe

Beispiel: Bereichsleiter, Dozent
Entscheidet, welche Aktionen der Benutzer durchführen darf, z.B. Kurse bewerten, verwalten oder auswerten.

7.1.3 Kurs

Beispiel: Medikamente und typische Krankheitsbilder, 20.06
Ein Kurs ist die Ausbildung, bei der eine Befragung durchgeführt wird.

7.1.4 Umfrage

Eine Umfrage ist eine Vorlage des Fragebogens, der bei einem Kurs ausgeteilt wird.

7.1.5 Feedback

Ein Feedback ist eine ausgefüllte Umfrage.

7.1.6 Frage

Eine Frage die zu einer Umfrage gehört.

7.1.7 Antwort

Eine mögliche Antwort auf eine Frage

8 Technologieentscheid

Um eine optimale Wahl der Technologien zu ermöglichen, wurden verschiedene Technologien für den Server sowie für den Client betrachtet. Der von der OdA G ZH zur Verfügung gestellte Server setzt das Debian Betriebssystem ein und wird bei einem externen Anbieter gehostet.

8.1 Technologien Server

Wir haben uns gegen *PHP* entschieden, da PHP eine Skriptsprache ist und somit den Inhalt nicht direkt an den Aufrufer ausliefern kann. Der Webserver startet das PHP-Skript, welches die Seite dann generiert. Wird eine Webseite 100mal aufgerufen, wird auch das Skript 100mal ausgeführt, was schlecht für die Performance ist. Ein weiterer Nachteil ist, dass PHP synchron arbeitet. Falls nun eine Abfrage auf die Datenbank abläuft wird das Skript angehalten bis die Datenbank antwortet.

JSF kommt für uns nicht in Frage, weil die ganze Umgebung ziemlich aufgebläht ist und die Dokumentation nicht sehr benutzerfreundlich ist.

Auch asp.net fällt weg, da man dafür ein Microsoft-Betriebssystem auf dem Server benötigt. Zusätzlich handelt es sich dabei um eine proprietäre Technologie, wofür in der Regel Lizenzgebühren anfallen.

Die Technologie unserer Wahl für den Server ist dementsprechend folgende:

8.1.1 node.js



Node.js ist eine serverseitige Plattform aufbauend auf der JavaScript Engine V8, welche von Chrome verwendet wird. Node.js ist vorallem dazu geeignet um Webserver zu realisieren. Die Plattform ist event-driven und bietet nur non-blocking Funktionen an.

- + Einfacheres Handhaben von mehreren gleichzeitigen Verbindungen
- + Keine zusätzliche Webserveranwendung notwendig
- + Kann überall installiert werden
- + Client und Server können in JavaScript programmiert werden
- Wir haben noch keine Erfahrungen mit node.js
- Bietet nur lowlevel *API* an -> Frameworks die auf node.js aufbauen können verwendet werden um Entwicklung zu vereinfachen.

8.2 Technologien Datenbanken

8.2.1 *SQL* vs noSQL

Für die effiziente Speicherung von Daten stehen unzählige Datenbankstrukturen zur Verfügung. Grundsätzlich gibt es zwei Typen der Datenspeicherung: SQL und noSQL. Der Begriff noSQL steht aber nicht für "kein SQL", sondern für "nicht nur SQL" (not only SQL). Das Ziel von noSQL ist es relationale Datenbanken sinnvoll zu ergänzen. In der untenstehenden Tabelle werden die Unterschiede der zwei Modelle aufgelistet.

	SQL-Datenbank	noSQL-Datenbank
Typ	SQL Datenbank	Verschiedene Arten: key-value stores, Document Databases, widecolumn stores, graph databases
Entwicklung	1970	2000
Beispiel	MySQL, Postgres, Oracle	MongoDB, Cassandra, HBase, Neo4j
Speicherart	Spalten mit festen Werten (String, Integer etc.)	JSON oder XML
Schemas	Datenbanken, Tabellen	Datenbanken, Collections
Skalierbarkeit	Es können inkonsistente Daten entstehen, sollte eine Tabelle mit einem oder mehreren Feldern erweitert wird.	Die Daten werden in JSON gespeichert, wobei eine hohe Skalierbarkeit verfügbar ist.
Lizenzen	Mix von open-source (Postgres, MySQL) und closed source (Oracle)	Open-source

Tabelle 8-q: Vergleich von SQL und noSQL

Um den Unterschied noch besser aufzeigen zu können, wird die Speicherung von Daten aus einer Form für eine SQL- und eine noSQL-Datenbank gezeigt.

Bei der SQL-Datenbank ist jede Reihe ein Eintrag mit Spalten voller Daten. Jedes Feld hat seine eigene Spalte. Es ist nun einfach ersichtlich, dass es Probleme gibt, Daten zu halten, wenn eine neue Spalte hinzugefügt werden muss. Es muss alles geändert werden. Möchte man zum Beispiel das Feld "EYES" ändern zu einer Selektion HAIR oder EYES könnte man die Daten daraus nicht in einem Feld speichern. Man wüsste dann nicht mehr, ob der gespeicherte Wert die Haar- oder die Augenfarbe ist.

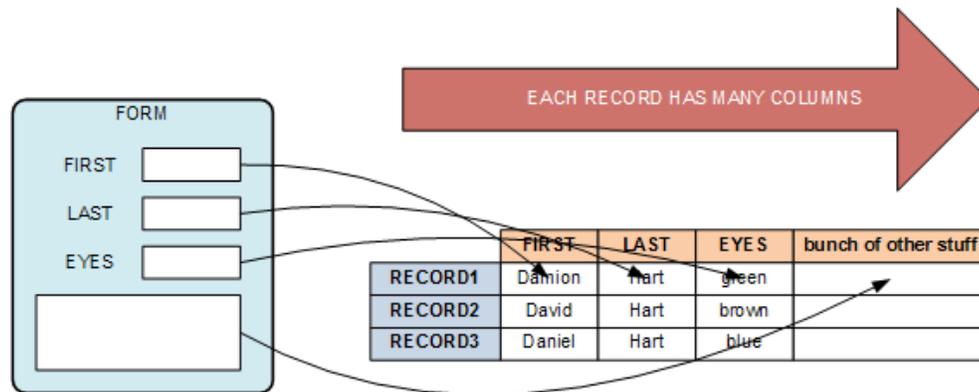


Abbildung 8-11: Beispiel einer Speicherung in einer SQL-Datenbank

Die noSQL-Datenbank hingegen speichert die Daten mittels Key/Value-Paaren. Die Einträge müssen nun nicht mehr die gleiche Anzahl von Einträgen oder den gleichen Key haben. Beim 3. Eintrag sieht man, dass der Key einfach zu HAIR geändert wurde. Am Schluss wurde sogar noch ein neuer Key hinzugefügt. Dies wäre nicht möglich bei einer SQL-Datenbank ohne bei jedem Eintrag ein neues Feld zu erstellen.

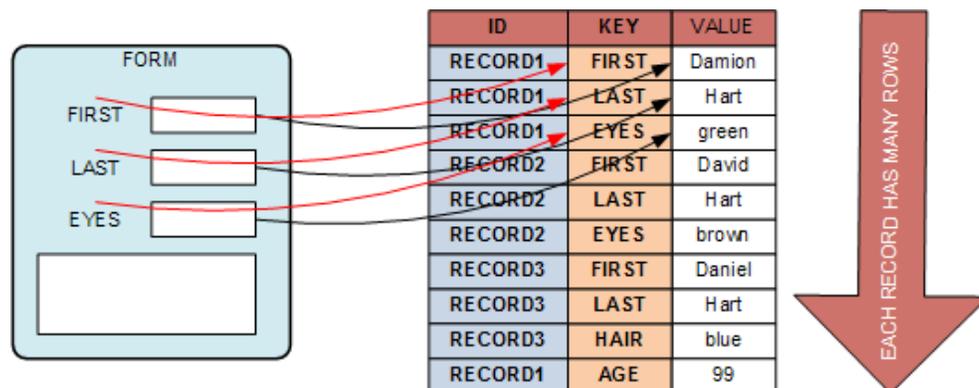


Abbildung 8-12: Beispiel einer Speicherung in einer noSQL-Datenbank

8.2.2 mongodb



MongoDB ist eine Open-Source-Lösung, die 2009 von 10gen nach zwei Jahren Entwicklung vorgestellt wurde. Die Implementierung ist in C++ geschrieben und bietet Unterstützung sowohl als 32-Bit als auch als 64-Bit Variante für Windows, Linux, Max OSX. MongoDB ist ausgelegt auf hohe Leistung, grosse Datenmengen, einfache Skalierbarkeit sowie hohe Flexibilität. Die nicht-relationale Datenbank arbeitet Dokumentorientiert. Es ist also auf eine schemafreie Struktur ausgelegt, was bedeutet, dass es keine festen Tabellenschemas und dadurch keine Relationstabellen und Joins gibt. Relationen lassen sich entweder direkt im Datensatz speichern oder individuell bei der Abfrage erstellen. Dies macht die Struktur um einiges flexibler und lässt sich bei Bedarf auf einfache Weise horizontal skalieren. Dokumente werden im *BSON*-Format ausgegeben und gespeichert. Somit kann jedes einzelne Dokument beliebig viele Felder besitzen und eine verschachtelte Array-Struktur ist ebenfalls möglich. MongoDB verwendet für *CRUD*-Operationen und Datenabfragen keine Abfragesprache wie *SQL*. Diese werden durch objektspezifische Methoden der jeweiligen unterstützten Programmiersprache (JavaScript, PHP, Perl, Ruby, Python, Java) realisiert.

8.3 Technologien Client

Auch auf der Client-Seite gibt es viele verschiedene Technologien. Da die Applikation sowohl auf Desktop Computern, als auch auf mobilen Geräten laufen muss, kann Flash als clientseitige Technologie den Zweck nicht erfüllen. Auch Java Applets haben einen enormen Nachteil, da zuerst immer die *JVM* geladen und das Applet heruntergeladen wird. Somit ist die Reaktionsgeschwindigkeit mit dieser Technologie auch eingeschränkt.

Für die Client-Seite unserer Applikation verwenden wir folgende Technologie:

8.3.1 JavaScript



JavaScript ist eine Programmiersprache, die syntaktisch an Java angelehnt ist und für das Entwickeln von dynamischen Webseiten geeignet ist.

- + sehr verbreitet
- + viele Libraries erhältlich
- es gibt keine Klassen

8.4 Zusätzliche Libraries und Frameworks für den Server

Über den Node Package Manager können beliebige Module installiert werden um die Arbeit zu vereinfachen. Nachfolgend sind Module aufgelistet, die wir für die Arbeit einsetzen.

8.4.1 express



Express ist ein minimales, performantes, auf node.js aufbauendes Framework, das robuste Funktionen anbietet um Single-, Multi-Pages oder gemischte Webapplikationen zu entwickeln. Es hilft die Applikation auf der Serverseite in eine *MVC*-Architektur zu organisieren. Zusätzlich gibt es eine grosse Auswahl an Template Engines, die mit Express eingesetzt werden können. Express hilft grundsätzlich die Routes, das Handling von Requests sowie die Views zu verwalten.

8.4.2 jade



Jade ist eine *Template* Engine für *HTML* oder *XML*. Sie macht HTML schlanker, da sie eine deutlich vereinfachte Schreibweise verwendet. Schliessende Elemente werden nicht verwendet, da dies eigentlich nur Overhead ist. Die Position im *DOM* Baum wird in Jade durch Einrückung bestimmt und Attribute werden in Klammern hinter das Element angehängt. Zudem bietet Jade eine Möglichkeit zur Template-Vererbung und die Benutzung von Variablen, Abfragen und Schleifen an. Express bietet zudem die Funktion an Jade Templates als Response zu rendern und Variablen zu übergeben.

Mit folgendem kleinen Beispiel wird der Unterschied von HTML und Jade aufgezeigt.

Listing 3: Beispiel in HTML

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <meta charset="UTF-8">
5     <link type="text/css" rel="stylesheet" href="/site.css" />
6     <title>Title of the document</title>
7   </head>
8   <body>
9     <h1>Hello world!</h1>
10  </body>
11 </html>
```

Listing 4: Beispiel in Jade

```
1 !!! 5
2 html
3   head
4     meta(charset='UTF-8')
5     link(type='text/css', rel='stylesheet', href='/site.css')
6     title Title of the document
7   body
8     h1 Hello world!
```

8.4.3 mongoose



Mongoose ist eine Abstraktionsschicht zwischen Node.js und MongoDB, mit der Objekte und Schematas modelliert, sowie eingebaute Funktionen zum Validieren von Daten gebraucht werden können. Die populate Funktion in Mongoose hilft dabei, Relationen, ähnlich wie ein *SQL JOIN*, zwischen Daten in einer dokumentorientierten Datenbank darzustellen. Durch Mongoose wird den dynamischen Collections, die keine Struktur besitzen, eine Struktur gegeben.

8.4.4 bcrypt

bcrypt.js

Bcrypt ist eine für die Erstellung von Passwort *Hashs*, kryptologische Hashfunktion, die auf dem Blowfish-Algorithmus basiert. Das Ziel von Bcrypt ist den Aufwand für das Hashen möglichst hoch zu halten, damit ein Angriff möglichst verlangsamt wird, was jedoch bei normalem Gebrauch nicht spürbar ist. Es bietet zusätzlich einen Work Factor an, um den Aufwand für eine Berechnung zu erhöhen, damit auch in Zukunft sichergestellt werden kann, dass sich ein Angriff mit schnellerer Hardware nicht lohnt.

8.4.5 async

Async.js

Die Node.js Plattform arbeitet asynchron, das bedeutet, dass das Ergebnis nicht als Rückgabewert vorliegt sondern mit einem Parameter an den Callback zurückgegeben wird. Durch diesen Aufbau ergeben sich schnell unüberschaubare Verschachtelungen. Mit async kann dies deutlich vereinfacht werden.

Mit folgendem Code-Beispiel wird gezeigt, inwiefern sich drei Abfragen auf die Datenbank mit async vereinfachen lassen.

Listing 5: Beispiel ohne async (einzelne Abfragen synchron)

```
1 query('one', function(resultOne){
2   query('two', function(resultTwo){
3     query('three', function(resultThree){
4       console.log('one: ' + resultOne + ' two: ' +
5         resultTwo + ' three: ' + resultThree);
6     });
7   });
8 });
```

Listing 6: Beispiel mit async (einzelne Abfragen synchron)

```
1 var async = require('async');
2 async.series({
3   one: function(callback){
4     query('one', function(result){
5       callback(null, result);
6     });
7   },
8   two: function(callback){
9     query('two', function(result){
10      callback(null, result);
11    });
12  },
13  three: function(callback){
14    query('three', function(result){
15      callback(null, result);
16    });
17  }
18 }, function(err, result){
19   console.log('one: ' + result.one + ' two: ' + result.two + '
20   three: ' + result.three);
};
```

Listing 7: Beispiel mit async (einzelne Abfragen asynchron)

```
1 var async = require('async');
2 async.parallel({
3   one: function(callback){
4     query('one', function(result){
5       callback(null, result);
6     });
7   },
8   two: function(callback){
9     query('two', function(result){
10      callback(null, result);
11    });
12  },
13  three: function(callback){
14    query('three', function(result){
15      callback(null, result);
16    });
17  }
18 }, function(err, result){
19   console.log('one: ' + result.one + ' two: ' + result.two + '
20   three: ' + result.three);
};
```

8.4.6 node-xlsx



Node-xlsx ist ein Excel Parser und Builder der auf der xlsx.js library von Microsoft aufbaut.

8.4.7 nodemailer



Nodemailer ist ein einfach zu gebrauchendes Modul um Emails mit Node zu versenden. Mit Nodemailer können alle Charaktere (Unicode), *HTML*-Inhalte sowie Klartext, Anhänge und eingebettete Bilder in HTML benutzt werden. Zudem wird SSL/STARTTLS für sicheres Emailsenden und verschiedene Transportmethoden (*SMTP*, Amazon SES, sendmail, directly) unterstützt.

8.4.8 password-generator

password-generator

Password-generator ist ein Modul, das ein zufälliges Passwort mit gewünschter Länge, nach einem gewünschten Muster, erstellen kann. Es ist ebenfalls möglich, einprägsame Passwörter, wie zum Beispiel "juxesi", zu generieren.

8.4.9 body-parser

body-parser

Body-parser erleichtert den Zugriff auf Informationen, die bei einem Clientrequest an den Server geschickt werden, indem er diese als *JSON* Objekt an req.body anhängt.

Listing 8: Beispiel mit body-parser bei survey/create

```
1 console.log(req.body);
2 /*output
3 { surveyname: 'Neue Umfrage',
4   surveytype: '0',
5   question1: { type: '0', text: 'Frage 1' },
6   question2: { type: '2', text: 'Frage 2' } }
7 */
```

8.4.10 connect-multiparty

connect-multiparty

Connect-multiparty erleichtert den Zugriff auf Dateien, die bei einem Clientrequest an den Server geschickt werden, indem diese in ein temporäres Verzeichnis gespeichert werden und die Informationen über die Datei an req.body.files angehängt wird. Die so erstellten Dateien müssen manuell gelöscht werden.

Listing 9: Beispiel mit connect-multiparty bei /course

```
1 app.post('/course', multipartMiddleware, function(req, res) {
2   course.create(excel.parseFile(req.files.excelFile.path),
3     function(err, c) {
4       fs.unlink(req.files.excelFile.path, function(err) {});
    });
  });
```

8.4.11 console-stamp

console-stamp

Console-stamp hängt einen benutzerdefinierten Zeitstempel an alle Zeilen in der Konsole an, was das Analysieren von Logfiles erheblich erleichtert.

8.4.12 forever

forever

Das Kommandozeilen-Tool forever stellt sicher, dass ein Skript ständig läuft. In unserem Fall wird forever beim Serverstart automatisch gestartet und stellt sicher, dass die Applikation nach einem Absturz neugestartet wird.

8.5 Zusätzliche Libraries und Frameworks für den Client

8.5.1 Bootstrap

B Bootstrap

Bootstrap ist eine Library für die Gestaltung von Tabellen, Buttons und Formularen. Bootstrap verwendet *CSS*, *HTML* und optional auch JavaScript. Es kann dazu verwendet werden um ein *responsive Design* zu gestalten.

- +** Die Webseite kann damit auf verschiedenen Geräten optimal angezeigt werden
- Ab Version 3.0 werden Webbrowser wie zum Beispiel Firefox 3.6 oder Internet Explorer 7 nicht mehr offiziell unterstützt

8.5.2 Bootstrap-datepicker

bootstrap-datepicker.js

Bootstrap-datepicker ist eine Library um das Arbeiten mit Daten zu erleichtern. Sie bietet verschiedene Formate an und ein Dropdown für das Auswählen eines Datums an.

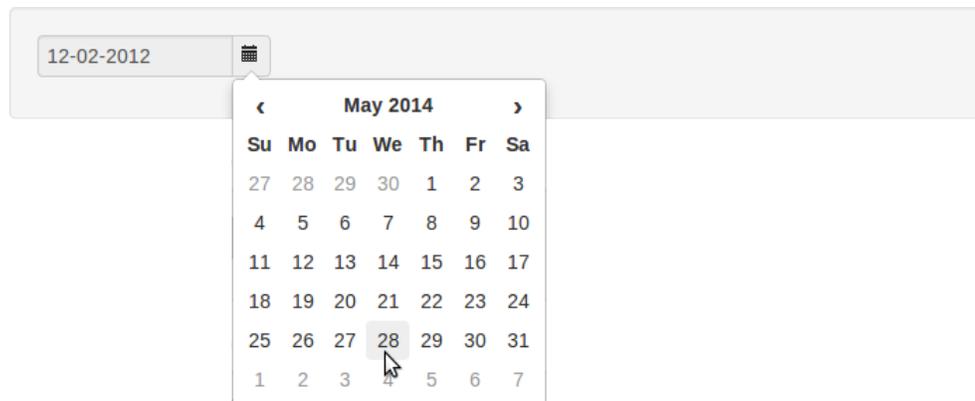


Abbildung 8-13: Bootstrap-datepicker Inputfeld

8.5.3 Bootstrap-slider

bootstrap-slider.js

Die Library Bootstrap-slider bietet vertikale oder horizontale touchfähige Slider, bei welchen man die minimum und maximum Werte, sowie die Schrittweite definieren kann, an.

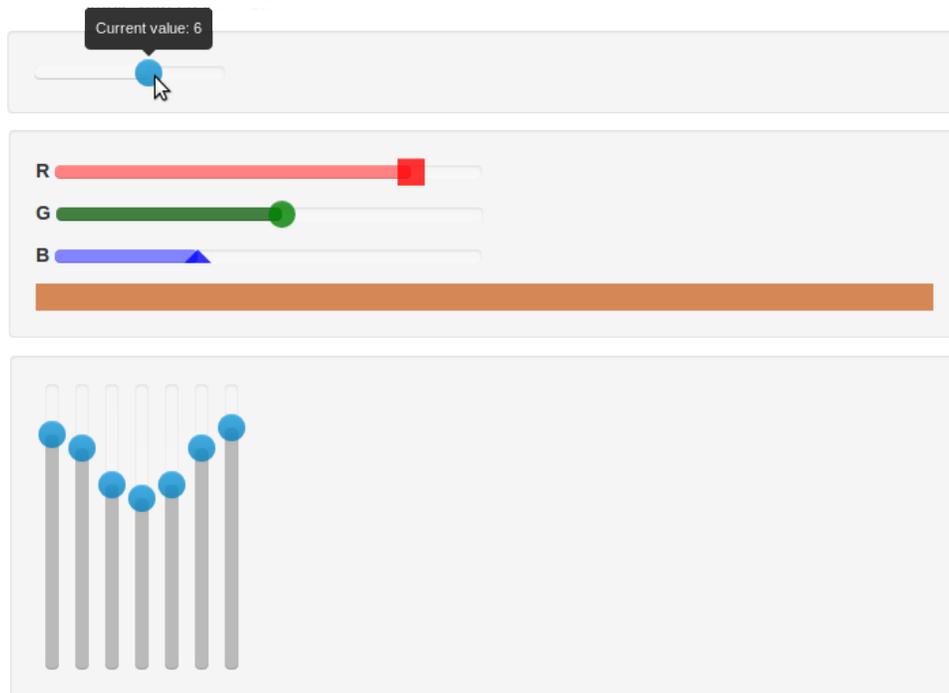


Abbildung 8-14: Verschiedene Slider der Library Bootstrap-slider

8.5.4 jquery

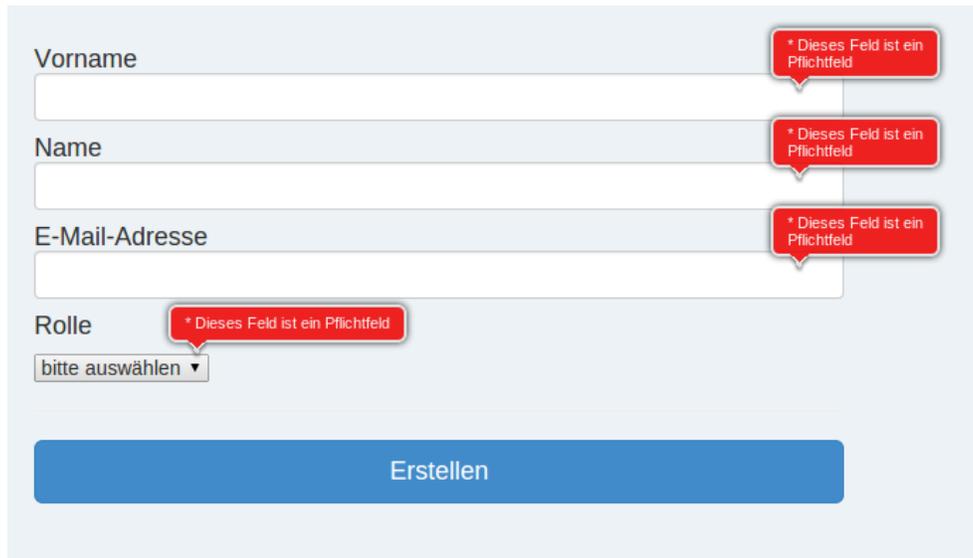


jQuery ist eine clientseitige Library, die die *DOM*-Manipulation und -Navigation vereinfacht. Zusätzlich bietet jQuery Ajax-Funktionalitäten, ein erweitertes Event-System sowie Animationen und Effekte.

- + Weniger Code, dank einfacheren DOM-Manipulationen
- jQuery 2.0 unterstützt nur Browser ab Internet Explorer 9, deshalb verwenden wir Version 1.11.1, die auch ältere Browser unterstützt

8.5.5 jQuery-Validation-Engine

Diese jQuery Erweiterung wird zur clientseitigen Überprüfung von Formularen verwendet.



The image shows a web form with four input fields: 'Vorname', 'Name', 'E-Mail-Adresse', and 'Rolle'. Each field has a red tooltip message that says '* Dieses Feld ist ein Pflichtfeld'. The 'Rolle' field is a dropdown menu with the text 'bitte auswählen'. Below the fields is a blue button labeled 'Erstellen'.

Abbildung 8-15: Validierung der Inputfelder mit der jQuery-Validation-Engine

9 Architektur

9.1 Systemübersicht

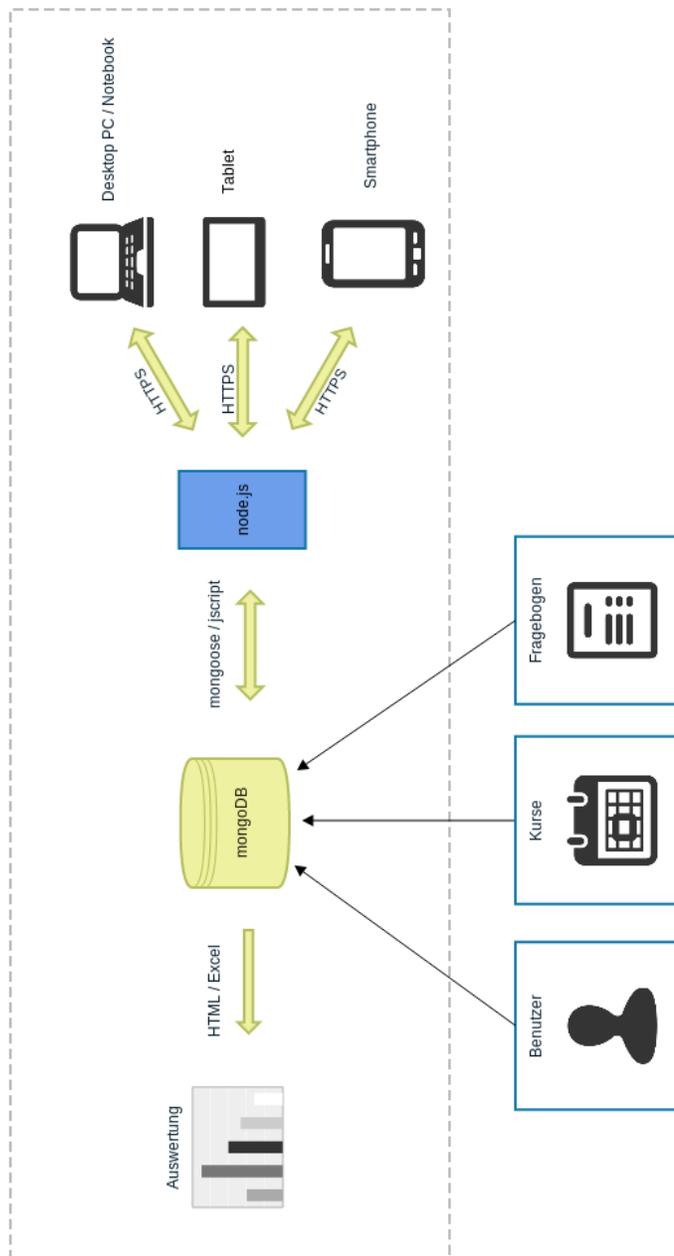


Abbildung 9-16: Systemübersicht

9.2 Entity Relationship Model

Das Entity Relationship Model wird aufgrund des Domainmodells erarbeitet. Einige Entitäten mit nur einer 1:1 Verbindung wurden direkt in andere integriert. Bei noSQL Datenbanken können auch n:n Verbindungen direkt abgebildet werden, weshalb nur noch 1:1 Beziehungen zwischen Entitäten bestehen die mehrere Verbindungen besitzen. Mit den integrierten Verbindungen ist es schwierig mit *UML* Tools die Datenbank ohne Zwischentabellen aufzuzeichnen.

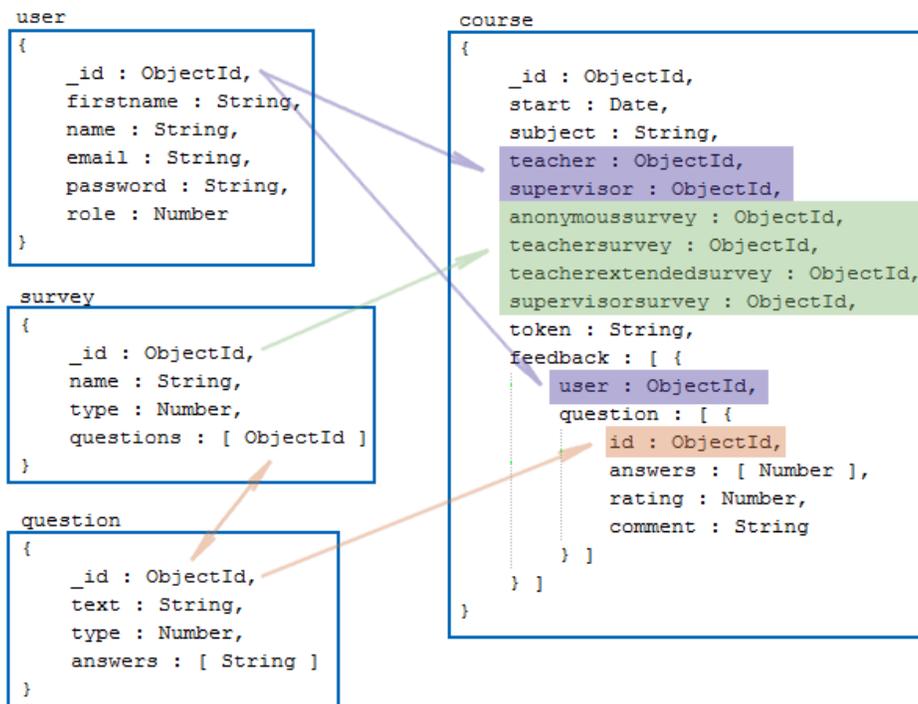


Abbildung 9-17: Entity Relationship Model

9.2.1 Collections

Name	Beschreibung
user	Beinhaltet alle Benutzer sowie deren Rolle
survey	Beinhaltet alle Umfragen sowie deren Fragen
question	Beinhaltet alle Fragen sowie deren Antworten
course	Beinhaltet alle Kurse mit ihren Mitarbeitern, Umfragen und Antworten

9.3 Verzeichnisstruktur

Die Applikation wird nach folgender Struktur aufgebaut.

	Ebene 1	Ebene 2	Beschreibung
Server-Scripts	/server	/https /modules /routes /views	Zertifikate die für eine sichere Verbindung benötigt werden. Javascript Code, der serverseitig ausgeführt wird. Implementierung der Routen. Die Routen sind im Kapitel Routing beschrieben. Jade-Templates, die serverseitig gerendert werden.
Client-Scripts	/public	/images /scripts /libs /stylesheets	Grafiken, die für die Gestaltung der Webseite verwendet werden. Javascript Code, der clientseitig ausgeführt wird. Mit dem Projekt verknüpfte Libraries. Stylesheets für die Darstellung der Webseite.
Tests	/test	/mocha	Tests, um die API zu testen.
app.js			Startet Server und die Applikation.
package.json			Beschreibt die Abhängigkeiten der Applikation.

Tabelle 9-s: Verzeichnisstruktur der Applikation

9.4 Routing

Das Routing wird in `/server/routes` definiert und beschreibt welche Funktionen und Views bei einem Aufruf eines Links ausgeführt beziehungsweise angezeigt werden.

Listing 10: Beispiel eines Routings (`server/routes/survey.js`)

```

1 app.post('/survey/create', function(req, res) {
2   if (!req.session.role === 3) {
3     res.redirect('/');
4     return;
5   }
6
7   survey.saveParse(survey.parse(req), function(err) {
8     res.redirect('/survey/admin');
9   });
10 });

```

In folgender Tabelle wird das Routing, so wie es implementiert wurde, beschrieben.

Rolle	Ebene 1	Ebene 2	Daten
*	/		get: Falls der Benutzer nicht angemeldet ist, wird <code>index.jade</code> gerendert, sonst wird auf <code>/survey</code> weitergeleitet.
*			post: Die geschickten Daten werden überprüft und mit den Einträgen in der Datenbank verglichen. Wenn sie korrekt sind wird der Benutzer auf <code>/survey</code> weitergeleitet ansonsten auf <code>/</code> .
B	/course		get: Die View <code>course/index.jade</code> wird gerendert.
B			post: Im Post angegebenes Excel-Dokument wird ausgelesen und verarbeitet. Danach wird die View <code>course/index.jade</code> mit den ausgelesenen Daten gerendert.
B		/create	get: Die View <code>course/create.jade</code> wird gerendert.
B			post: Der Kurs mit den im Post geschickten Daten wird in der Datenbank gespeichert.
B		/delete/:id	get: Der Kurs mit der im Link angegebenen <code>id</code> wird gelöscht.

B		/update/:id	get: Holt die Daten des Kurses mit der im Link angegebenen id und rendert course/create.jade. Gibt es keinen Kurs mit angegebener id, wird auf /course weitergeleitet
B			post: Der Kurs mit der im Link angegebenen id wird mit den im Post gelieferten Daten aktualisiert.
DSB	/logout		get: Die Session wird zerstört ->der Benutzer wird abgemeldet.
DB	/evaluation		get: Die View evaluation/index.jade wird gerendert.
DB		/view/:id	get: Falls die Umfrage mit der im Link angegebenen id existiert, wird die View evaluation/view.jade mit den Daten der Auswertungen gerendert, sonst wird auf /evaluation weitergeleitet.
DB		/covariant/:id	get: Falls die Umfrage mit der im Link angegebenen id existiert, wird die View evaluation/covariant.jade mit den Daten der Auswertungen gerendert, sonst wird auf /evaluation weitergeleitet.
B		/generate/:id	get: Wenn die Umfrage mit der im Link angegebenen id existiert, wird ein Excel-Dokument mit der entsprechenden Auswertung generiert und in der Response mitgeliefert, sonst wird auf /evaluation weitergeleitet.
DSB	/survey		get: Die View survey/index.jade wird gerendert.
B		/admin	get: Die View survey/admin.jade wird gerendert.
B		/create	get: Die View survey/create.jade wird gerendert.
B			post: Die im Post mitgeschickten Daten eines Fragebogens werden als neuer Fragebogen in der Datenbank gespeichert. Danach wird auf /survey/admin weitergeleitet.

B		/delete/:id	get: Der Fragebogen mit der im Link angegebenen id wird gelöscht.
DB		/extend	get: Die View survey/extend.jade wird gerendert.
DB		/extend/:id	get: Die View surevy/create.jade wird mit der Umfrage der im Link angegebenen id gerendert.
DB			post: Erweitert den Kurs mit der im Link angegebenen id mit den im Post mitgeschickten Fragen. Danach wird auf /survey/extend weitergeleitet.
*		/participate/:id	get: Die View survey/participate.jade wird mit der Umfrage mit der im Link angegebenen id gerendert.
*			post: Die Resultate der Umfrage, die mit dem Post gesendet werden, werden in der Datenbank gespeichert. Danach wird auf /survey/successful weitergeleitet.
*		/successful	get: Die View survey/successful wird gerendert.
*		/successfulFb	get: Die View survey/successful wird ohne die Weiterleitung auf den Usability Test gerendert.
B		/update/:id	get: Holt die Daten des Fragebogens mit der im Link angegebenen id und rendert survey/create.jade. Gibt es keinen Fragebogen mit angegebener id, wird auf /survey/admin weitergeleitet.
B			post: Der Fragebogen mit der im Link angegebenen id wird mit den im Post gelieferten Daten aktualisiert.
*	/token		post: Es wird auf /survey/participate/+ c.id weitergeleitet, wobei c durch den Kurs, der mit dem im Link angegebenen Token verknüpft ist, ersetzt wird.
*		/:token	get: Gleich wie post, token übertragen über get parameter

*		/display/:token	get: Die View survey/token wird mit dem im Link angegebenen Token gerendert.
B	/user		get: Die View user/index.jade wird gerendert.
B		/create	get: Die View user/create.jade wird gerendert.
B			post: Der Benutzer mit dem im Post gesendeten Daten wird erstellt, ein Passwort generiert und an seine Email verschickt. Danach wird auf /user weitergeleitet.
B		/delete/:id	get: Der Benutzer mit der im Link angegebenen id wird gelöscht.
DBS		/profile	get: Die View user/profile.jade wird gerendert.
DBS			post: Die im Post gesendeten Daten werden überprüft, und wenn sie richtig sind beim angemeldeten User überschrieben. Danach wird user/profile.jade gerendert.
*		/restore	get: Die View user/restore.jade wird gerendert.
*			post: Dem zur im Post gesendeten Email passende User wird ein Email mit einem neu generierten Passwort gesendet. Danach wird user/restore.jade gerendert.
B		/update/:id	get: Holt die Daten des Benutzers mit der im Link angegebenen id und rendert user/create.jade. Gibt es keinen Benutzer mit angegebener id, wird auf /user weitergeleitet.
B			post: Der Benutzer mit der im Link angegebenen id wird mit den im Post gelieferten Daten aktualisiert.

Tabelle 9-t: Beschreibung des Routings

Die Spalte "Rolle" beschreibt, welche Benutzerrolle auf den Link zugreifen kann. Wird auf den Link zugegriffen ohne die Zugriffsrechte zu besitzen wird auf / weitergeleitet.

* = Jeder, D = Dozierende, S = Supervisoren, B = Bereichsleiter

9.5 Views / Templating

Ebene 1	Ebene 2	Beschreibung
index.jade		Loginscreen <i>Template</i>
course/	create.jade	Template für die Erstellung und das Verändern von Daten über den Kurs.
	index.jade	Template für die Übersicht aller Kurse.
evaluation/	index.jade	Template für die Übersicht aller Auswertungen, die ein Benutzer sehen darf.
	covariant.jade	Template für die Berechnung von Kovarianten.
	view.jade	Template für die Auswertung eines Kurses.
layout/	footer.jade	Template für die Fusszeile.
	header.jade	Template für die Kopfzeile.
	layout.jade	Basistemplate das durch die anderen erweitert wird.
	navigation.jade	Template für die Navigation auf der linken Seite.
	scripts.jade	Scripts die für alle Dokumente geladen werden.
survey/	admin.jade	Template für die Übersicht der erstellten Fragebogen.
	create.jade	Template für die Erstellung und das Verändern von Fragebogen.
	index.jade	Template für die Übersicht aller offenen Kurse des jeweiligen Benutzers.
	participate.jade	Template für die Umfrage.
	successful.jade	Template für die Rückmeldung nach dem erfolgreichen Teilnehmen an einer Umfrage.
user/	create.jade	Template für die Erstellung und das Verändern von Daten der Benutzer.
	index.jade	Template für die Übersicht aller registrierten Benutzer, aufgeteilt auf ihre Benutzergruppen.
	profile.jade	Template für die Veränderung der eigenen Profildaten.
	restore.jade	Template für die Rücksetzung des Passwortes.

Tabelle 9-u: Beschreibung der Views

Listing 11: Basislayout layout/layout.jade

```
1 doctype html
2 html
3   head
4     meta(name='viewport', content='width=device-width, initial-
5         scale=1.0')
6     title= title
7     link(rel='stylesheet', href='/libs/bootstrap/css/bootstrap.
8         min.css')
9     block head
10    link(rel='stylesheet', href='/stylesheets/style.css')
11  body
12    include header
13    if session.user
14      .col-md-3
15        include navigation
16        block menu
17      block content
18      include footer
19      include scripts
20      block scripts
```

Listing 12: Beispiel einer Erweiterung des Layouts

```
1 extends ../layout/layout
2
3 append head
4   link(rel='stylesheet', type='text/css', href='/libs/slider/css
5       /slider.css')
6   ...
7 append menu
8   if(session.role === 3)
9     .ta_container
10    ul.list-group
11    li.list-group-item
12      a(href='/evaluation/generate/#{course}')
13      span.glyphicon.glyphicon-export
14      | Excel Datei generieren
15    ...
16
17 append content
18   .col-md-8
19   .ta_container
20     h3 Anonyme Antworten
21     h5 Anzahl Teilnehmer: #{anonymousparticipants}
22     ...
```

9.6 Schemas

Ein Mongoose Schema bildet eine Node.js Klasse auf eine Collection einer MongoDB ab.

Listing 13: Beispiel eines Schemas (question.js)

```
1 var question = new schema({
2   text : {
3     type : String,
4     required : true
5   },
6   type : {
7     type : Number,
8     required : true
9   },
10  answers : [ {
11    type : String
12  } ]
13 });
```

Die Schema Klasse(auch Modell genannt) verfügt so über alle Mongoose Standardfunktionen wie z.B. create, update, find oder delete.

Listing 14: Beispiel einer Standardfunktion

```
1 question.create({text : 'bsp', type : 0}, function(err, q) {
2   //q ist die erstellte Instanz der question Klasse
3 });
```

Diese können erweitert werden mit Funktionen die über das Modell(statics) oder auf ein Dokument(methods) aufgerufen werden können.

Listing 15: Beispiel einer static Funktion

```
1 question.statics.updateMultiple = function(questions, callback)
2   {
3   async.each(questions, function(q, asynccallback) {
4     question.findByIdAndUpdate(q.id, q, function(err) {
5       asynccallback(err);
6     });
7   }, function(err) {
8     callback(err);
9   });
```

Listing 16: Beispiel einer static Verwendung

```
1 var parse = survey.parse(req);
2 question.updateMultiple(parse.oldquestions, function(err) {
3
4 });
```

9.7 Module

Module sind JavaScript Dateien die ihre Funktionen Exportieren, damit sie in anderen JavaScripts über die node.js Funktion "require(filename)" verwendet werden können.

Listing 17: Beispiel eines Moduls (excel.js)

```
1 exports.buildDocument = function(data){
2   var buffer = xlsx.build({
3     worksheets : [ {
4       "name" : "Auswertung",
5       "data" : data
6     } ]
7   });
8   return buffer;
9 }
```

Listing 18: Beispiel einer Verwendung (evaluation.js)

```
1 var excel = require('../modules/excel');
2 var excelFile = excel.buildDocument(excelData);
3 res.send(excelFile);
```

10 User Interface

10.1 Corporate Design

Um das Prinzip des *Corporate Design* nicht zu verletzen wird das Design an die Webseite der Oda G ZH [oda] angelehnt. Die Webpage wird aber so gestaltet, damit sie auch auf Smartphones benutzerfreundlich und attraktiv wirkt. Deshalb werden einige Dinge, wie zum Beispiel die Navigation, abgeändert.

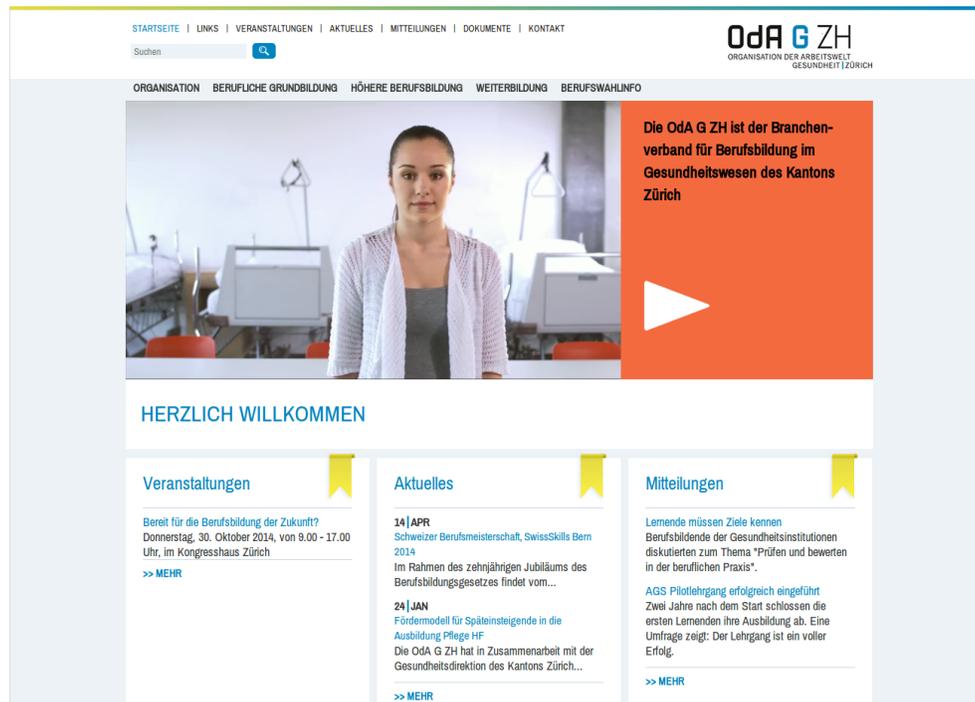


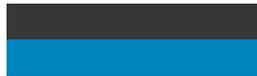
Abbildung 10-18: Webseite der Oda G ZH

	Farbcode	Farbe
Primärfarben	0084be ffffff	
Hintergrundfarbe	ecf2f5	

Tabelle 10-v: benutzte Farbkombinationen auf der Webseite von Oda G ZH

Schrift :

- Schriftart: 'Archivo Narrow', verdana
- Schriftgrösse: 16px
- Schriftfarben:
0x373737
0x0084be



10.2 Layout

Das Layout von TeachEval wird grob in die vier Bereiche header, navigation, menu und content aufgeteilt. In der Abbildung 10-19 sind die einzelnen Bereiche hervorgehoben worden.

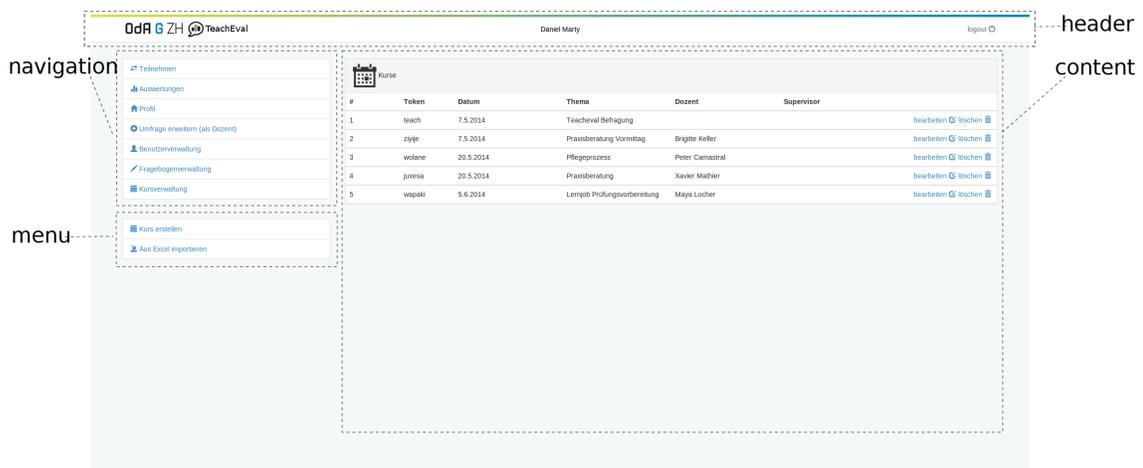
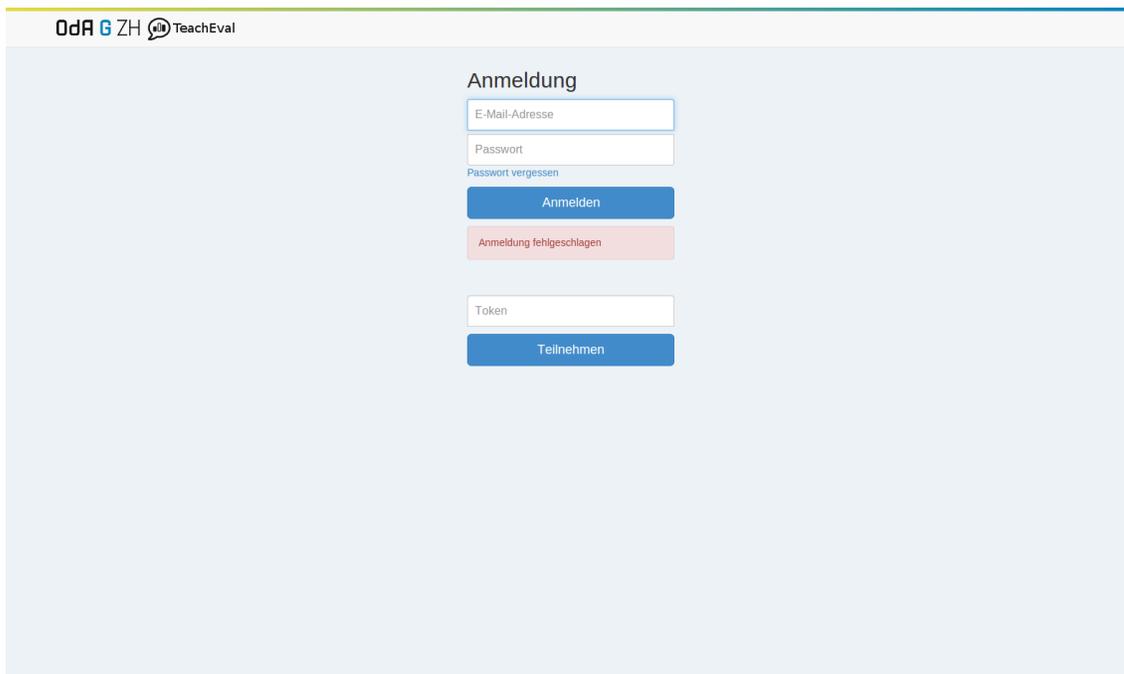


Abbildung 10-19: Layout unserer Webapplikation

Im Bereich **header** befindet sich das Logo, was zugleich einen Link zur Startseite ist. Der Header wird immer angezeigt, wobei das Logo bei einer zu niedrigen Auflösung gewechselt wird, da es sonst zu viel Platz bräuchte. Ist man am System angemeldet, wird in diesem Bereich auch der Logout Button angezeigt sowie der Name der angemeldeten Person. Die Bereiche **navigation** und **menu** sind nur für angemeldete Benutzer sichtbar. Die Navigation wird je nach Benutzerrolle eingeschränkt, verändert sich aber nicht. Welche Funktionen eine Benutzerrolle nutzen kann, ist im Use Case Diagramm in Abbildung 6-7 beziehungsweise 6-7 sichtbar. Das Menu unterscheidet sich von der Navigation dadurch, dass es sich verändert. Das Menu zeigt immer die zum aktiven Navigationspunkt gehörenden Funktionen an. Bei einer niedrigeren Auflösung ($\leq 992\text{px}$) wird die Navigation und das Menu oberhalb vom Bereich **content** angezeigt und nicht links davon. Im Bereich **content** wird der Inhalt der aufgerufenen Seite angezeigt.

10.3 Login



The screenshot shows a web interface for login. At the top left, there is a logo for 'OöA 6 ZH' and the text 'TeachEval'. The main heading is 'Anmeldung'. Below the heading, there are two login sections. The first section contains an input field for 'E-Mail-Adresse', an input field for 'Passwort', a link for 'Passwort vergessen', and a blue button labeled 'Anmelden'. Below this section is a red error message that says 'Anmeldung fehlgeschlagen'. The second section contains an input field for 'Token' and a blue button labeled 'Teilnehmen'.

Abbildung 10-20: Anmeldebildschirm

Der Anmeldebildschirm ist so gestaltet, dass sich die Benutzer anmelden können, aber auch so, dass es eine direkte Möglichkeit zur Weiterleitung zu einer Teilnehmerumfrage mittels Token gibt. Vergessene Passwörter können auch über den Anmeldebildschirm zurückgesetzt werden.

10.4 Umfrage

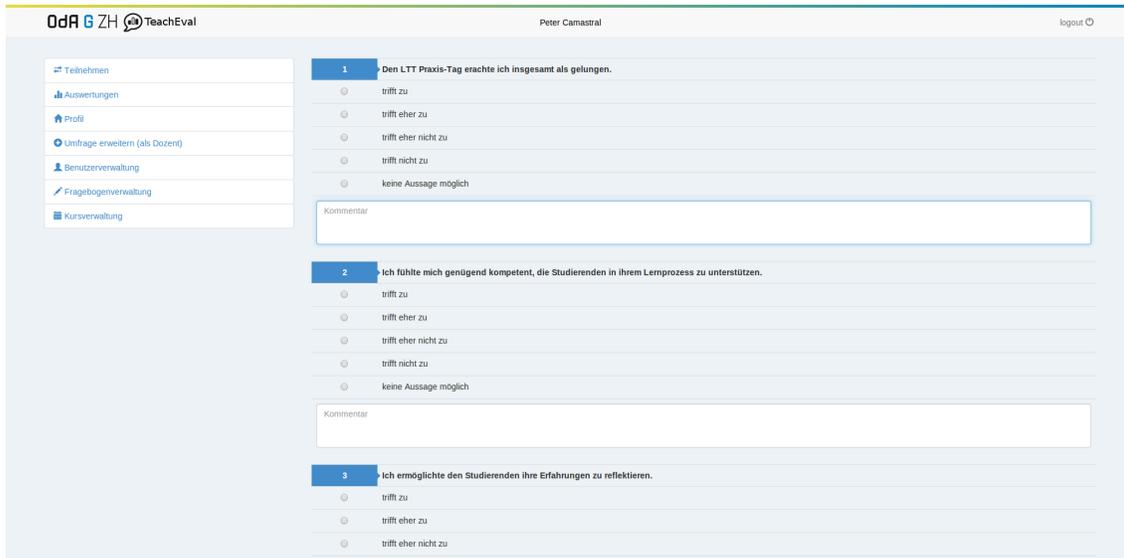


Abbildung 10-21: Ausschnitt eines Dozierenden-Fragebogens

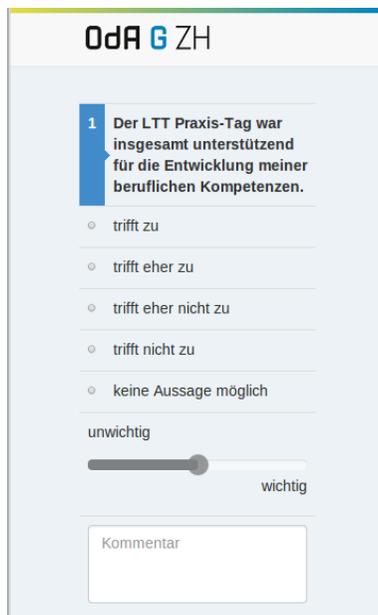


Abbildung 10-22: Fragebogen auf einem Smartphone

Die Umfrage kann als Dozierender oder Supervisor, wie in der Abbildung 10-21, oder als Teilnehmer, wie in der Abbildung 10-22, ausgefüllt werden. Als angemeldeter Benutzer kommt man automatisch auf den Fragebogen, der dem Benutzer zugeteilt wurde. Bei der Umfrage wurde darauf geachtet, dass sie auch für geringe Auflösungen in den Browsern der Smartphones ($\geq 320\text{px}$) noch zu bedienen ist.

10.5 Auswertungen

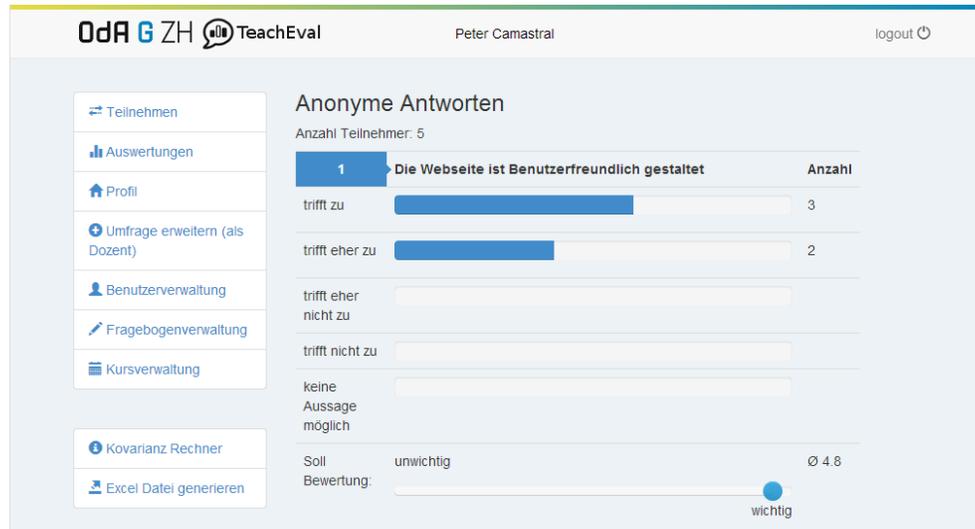


Abbildung 10-23: Ausschnitt einer Auswertung

Auswertungen werden mit einer Progress-Bar pro Antwort angezeigt, die darstellt, wie viele Teilnehmer diese selektiert haben.

Bei anonymen Umfragen wird zusätzlich noch ein Slider mit der Durchschnittlichen Soll Bewertung angezeigt.



Abbildung 10-24: Ausschnitt einer Kovarianz Berechnung

Die Kovarianz wird aus den Antwortdaten von zwei Fragen berechnet, welche über ein Dropdown selektiert werden. Es ist möglich, die Kovarianz einer Frage mit sich selber zu berechnen.

Zusätzlich wird eine Erklärung zur Bedeutung des Werts angezeigt.

10.6 Ressourcen verwalten

#	Token	Datum	Thema	Dozent	Supervisor
1	teach	7.5.2014	Teacheval Befragung		
2	ziyije	7.5.2014	Praxisberatung Vormittag	Brigitte Keller	
3	wolane	20.5.2014	Pflegeprozess	Peter Camastral	
4	juxesa	20.5.2014	Praxisberatung	Xavier Mathier	
5	wapaki	5.6.2014	Lernjob Prüfungsvorbereitung	Maya Locher	
6	noyeba	11.6.2014	Gruppen leiten	Patrick Lenzin	
7	kifoke	11.6.2014	Gruppen leiten	Brigitte Bourke	

Abbildung 10-25: Übersicht aller Kurse

Das Verwalten der Ressourcen "Kurse", "Fragebogen" und "Benutzer" ist überall gleich gestaltet. In der Verwaltung werden alle entsprechenden Ressourcen in einer Tabelle, wie in Abbildung 10-25 dargestellt. Die einzelnen Einträge können entweder bearbeitet, was gleich dargestellt ist wie das Erstellen einer neuen Ressource, (Abbildung 10-26) oder gelöscht werden. Beim Löschen wird zudem mit einem Modal, wie in Abbildung 10-27 sichergestellt, dass es nicht unabsichtlich zu einem Löschvorgang kommt.

Abbildung 10-26: Benutzerinformationen

Abbildung 10-27: Löschvorgang

10.7 Fragebogen erstellen

The screenshot shows the TeachEval interface for creating a questionnaire. At the top, the user is logged in as 'Daniel Marty'. The interface is divided into a left sidebar with navigation options and a main content area. The sidebar includes options like 'Teilnehmen', 'Auswertungen', 'Profil', 'Umfrage erweitern (als Dozent)', 'Benutzerverwaltung', 'Fragebogenverwaltung', and 'Kursverwaltung'. The main content area shows a questionnaire titled 'Testbogen Dozent' with a 'Fragebogentyp' of 'Dozent'. It contains three questions: 1. A Likert scale question about the LTT Praxis-Tag. 2. A question about didactic-methodical learning forms, with a checkbox for 'Rollenspiel'. 3. A question 'Was ist mir gelungen?'. The interface also includes buttons for adding new questions and deleting existing ones.

Abbildung 10-28: Fragebogengenerator

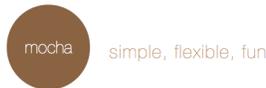
Mit dem Fragebogengenerator ist es möglich, Fragebogen online zu erstellen und direkt zu sehen, wie der Fragebogen nachher aussehen wird. In der Abbildung 10-28 sind die drei Fragetypen, die im Kapitel [Fragebogen erstellen](#) behandelt werden, zu sehen. Dozierende haben die Möglichkeit einen Teilnehmer-Fragebogen zu erweitern. Sie haben aber nur die Möglichkeit zusätzliche Fragen hinzuzufügen, alle anderen Funktionen sind nicht verfügbar.

10.8 Hinweismeldungen

Hinweise wie zum Beispiel ein erfolgreiches Zurücksetzen des Passwortes oder eine fehlgeschlagene Anmeldung, wie in Abbildung 10-20 zu sehen ist, werden durch Meldungen angezeigt.

11 Testing

11.1 mocha



Mocha ist ein JavaScript Testframework entwickelt für node.js. Es führt Tests seriell aus und unterstützt asynchrone Funktionen. Synchroner Funktionen sind erfolgreich, solange keine Exception geworfen wird und keine Asserts fehlschlagen.

Listing 19: Beispiel eines synchronen Mocha tests

```

1 describe('parseFile', function() {
2   it('should parse an excel file', function() {
3     var data = excel.parseFile('test/excelfixture.xlsx');
4     if(data[0].subject !== 'TeachEval Testing')
5       throw new Error('file was not parsed correctly!');
6   });
7 });

```

Asynchrone Funktionen sind erfolgreich, wenn die Funktion `done(err)` aufgerufen wird bevor Fehler passieren und der Übergabeparameter null ist.

Listing 20: Beispiel eines asynchronen Mocha tests

```

1 describe('authenticate', function() {
2   it('should authenticate user', function(done) {
3     user.authenticate('hans@muster.test', 'test1', function(u) {
4       if (!u)
5         throw new Error('user could not be authenticated
6           correctly!');
7       else
8         assert.equal(u.name, 'Muster');
9       done();
10    });
11 });

```

Um mocha mit output zu starten muss im Projektverzeichnis "mocha -R spec" aufgerufen werden.

```
course
  generateUniqueToken
    U should create a token
  pre(save)
    U token should be attached on save
  addFeedback
    U should add a feedback entry to a course
  prepareCovariantData
    U should prepare answer arrays for covariants
  covariant
    U should calculate covariants
  evaluate
    U should evaluate a complete course

6 passing (58ms)
```

Abbildung 11-29: Tests mit mocha

11.2 siege

Siege ist ein Tool mit dem es möglich ist Stresstests für HTTP und [HTTPS](#) durchzuführen. Das Tool kann unter Linux mittels `apt-get install siege` installiert werden. Die Option `-d` steht für das Intervall von 0 Sekunden bis zur angegebenen Zeit, in dem jeweils ein neuer Request ausgeführt wird. Mit der Option `-c` kann die Anzahl Benutzer festgelegt werden. Wie in [Abbildung 11-30](#) zu sehen ist, braucht die Webapplikation für das Darstellen einer Umfrage bei 25 gleichzeitigen Benutzer maximal 2.89 Sekunden.

```
dani@daniMint ~ $ sudo siege -d0.2 -c25 https://teacheval.ch:40000/survey/participate/536945f6c708d2d46f65d61b
** SIEGE 3.0.1
** Preparing 25 concurrent users for battle.
The server is now under siege...^C
Lifting the server siege... done.

Transactions:          600 hits
Availability:          100.00 %
Elapsed time:          60.20 secs
Data transferred:     10.95 MB
Response time:         2.46 secs
Transaction rate:      9.97 trans/sec
Throughput:            0.18 MB/sec
Concurrency:          24.49
Successful transactions: 600
Failed transactions:   0
Longest transaction:   2.89
Shortest transaction:  0.44
```

Abbildung 11-30: Stresstest mit siege

11.3 Usability Test

Um die Benutzbarkeit der Applikation zu testen wird zuerst ein Test mit ausgewählten Personen durchgeführt, die nichts mit der Oda G ZH zu tun haben, und danach wird die Benutzbarkeit durch Feedbacks von Brigitte Keller und durch eine an die kursbezogene Umfrage angehängte freiwillige Umfrage verbessert.

11.3.1 Resultate der Tests mit ausgewählten Benutzern



Abbildung 11-31: Benutzerinformationen

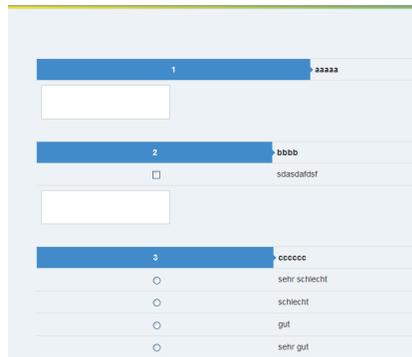


Abbildung 11-32: Umfrage auf den getesteten Browsern



Abbildung 11-33: Fehler bei der Anzeige des bootstrap-sliders

Bei diesem Test ist uns aufgefallen, dass noch viel ältere Browser im Gebrauch sind, als wir angenommen haben. Die Applikation sah nicht aus wie auf Abbildung 11-31, so wie wir es gewohnt waren, sondern wie auf Abbildung 11-32. Auch der Slider wurde nicht richtig dargestellt (Abbildung 11-33). Daraufhin wurden die Verteilung der Browser im Kapitel 6.3.2 genauer angeschaut und die Applikation auf jQuery in der Version 1 aufgebaut. Somit konnte das Problem behoben werden.

11.3.2 Feedback von Brigitte Keller



Abbildung 11-34: Problem beim Login

Es gab Probleme mit dem Login. Der Internet Explorer prüft Felder mit dem Typ "email" automatisch und zeigt bei E-Mail-Adressen mit dem Charakter '-' an, dass es sich um eine ungültige handelt. Das Problem dabei ist, dass dieses Feld das Inputfeld für das Passwort überdeckt, wie in Abbildung 11-34 zu sehen ist, und somit für Verwirrung sorgt. Als Konsequenz wurde die Validierung der E-Mail-Felder entfernt.

11.3.3 Feedback der Teilnehmer

- Skalierung stimmt nicht mit Fragebogen überein.
- wenn Fehler bei Auswertung auftreten bzw, Fragen nicht beantwortet werden sollte eine Fehlermeldung erscheinen
- nach Abschluss des Feedbacks nicht wieder Verlinkung auf Umfrage-link

Die Skalierung der Fragebogen wurde überarbeitet und an den Fragetyp angepasst. Für die Validierung der Felder wird die im Kapitel 8.5.5 vorgestellte Engine benutzt. Die zirkuläre Abhängigkeit der Umfrage wurde gelöst, da die freiwillige Umfrage entfernt wurde.

12 Ergebnis

Gemäss der Analyse wurde im Rahmen dieser Bachelorarbeit unter Einsatz moderner Webtechnologien eine Webapplikation zur Unterrichtsbewertung entwickelt.

Das dabei entstandene Produkt deckt alle Features gemäss den erarbeiteten Use Cases ab.

Die Verwendung eines aktuellen Web Application Frameworks sichert die Wartbarkeit des Produkts ab. Die responsive Weboberfläche erlaubt es den Teilnehmern, auf verschiedenen Geräten wie Tablets, Smartphones und Desktop PCs mit der Applikation interagieren zu können. Das Deployment auf das produktive System wurde durchgeführt und kann mit der Installationsanleitung selber vorgenommen werden.

12.1 Ausblick

Es gibt mehrere kleinere Features die noch implementiert oder optimiert werden könnten:

- Automatische Zuordnung von Lehrpersonen beim Kursimport z.B. über E-Mail-Adressen
- Excel Export schöner formatieren und Formeln sowie Diagramme einfügen
- Mehr Diagramme bei den Auswertungen anzeigen

Abbildungsverzeichnis

5-1	Ausschnitt aus dem Word Dokument für die Kurserfassung	12
5-2	Beispiel einer Teilnehmerliste	12
5-3	Fragetyp 1 (aus dem Teilnehmer-Fragebogen)	13
5-4	Fragetyp 1 (aus dem Dozierenden-Fragebogen)	14
5-5	Fragetyp 2	14
5-6	Fragetyp 3	15
6-7	Use Case Diagramm Teil 1	18
6-8	Use Case Diagramm Teil 2	19
6-9	Statistiken der Browser von w3schools	48
7-10	Domainmodell	49
8-11	Beispiel einer Speicherung in einer SQL-Datenbank	53
8-12	Beispiel einer Speicherung in einer noSQL-Datenbank	53
8-13	Bootstrap-datepicker Inputfeld	62
8-14	Verschiedene Slider der Library Bootstrap-slider	63
8-15	Validierung der Inputfelder mit der jQuery-Validation-Engine	64
9-16	Systemübersicht	65
9-17	Entity Relationship Model	66
10-18	Webseite der OdA G ZH	76
10-19	Layout unserer Webapplikation	77
10-20	Anmeldebildschirm	78
10-21	Ausschnitt eines Dozierenden-Fragebogens	79
10-22	Fragebogen auf einem Smartphone	79
10-23	Ausschnitt einer Auswertung	80
10-24	Ausschnitt einer Kovarianz Berechnung	80
10-25	Übersicht aller Kurse	81
10-26	Benutzerinformationen	81
10-27	Löschvorgang	81
10-28	Fragebogengenerator	82
11-29	Tests mit mocha	84
11-30	Stresstest mit siege	84
11-31	Benutzerinformationen	85
11-32	Umfrage auf den getesteten Browsern	85
11-33	Fehler bei der Anzeige des bootstrap-sliders	85
11-34	Problem beim Login	86

Tabellenverzeichnis

6-a	UC1 an anonymer Umfrage teilnehmen	20
6-b	UC2 Umfrage durchführen	22
6-c	UC3 am System anmelden	23
6-d	UC4 vom System abmelden	24
6-e	UC5 Passwort zurücksetzen	26
6-f	UC6 an personalisierter Umfrage teilnehmen	27
6-g	UC7 persönliche Daten verwalten	28
6-h	UC8 Kovarianz zweier Fragen betrachten	29
6-i	UC9 Auswertung anschauen	30
6-j	UC10 Auswertung exportieren	31
6-k	U11 Fragebogen erweitern	32
6-l	UC12 Umfragetoken anzeigen	33
6-m	UC13 Benutzer verwalten (CRUD)	35
6-n	UC14 Fragebogen verwalten (CRUD)	37
6-o	UC15 Kurse verwalten (CRUD)	39
6-p	UC16 Kursliste importieren	40
8-q	Vergleich von SQL und noSQL	52
9-s	Verzeichnisstruktur der Applikation	67
9-t	Beschreibung des Routings	71
9-u	Beschreibung der Views	72
10-v	benutzte Farbkombinationen auf der Webseite von Oda G ZH	77

Literaturverzeichnis

[lar] Craig Larman. UML 2 und Patterns angewendet: Objektorientierte Softwareentwicklung (1. Auflage 2005). ISBN 978-3-8266-1453-8. URL: <http://www.amazon.de/UML-Patterns-Angewendet-mitp-Professional/dp/3826614534/>

[w3s] w3schools <http://www.w3schools.com/browsers/default.asp>.
(Besucht am 14.5.2014)

[oda] OdA G ZH, Zürich. URL: <http://http://www.oda-g-zh.ch/>.
(Besucht am 26.4.2014)

Glossary

A

API

”Application Programming Interface”: Ein Programmteil, der von einem Softwaresystem anderen Programmen zur Anbindung an das System zur Verfügung gestellt wird. [51](#)

B

BSON

”Binary JSON”: Datenformat, das zum Sichern und Transferieren von Daten genutzt wird. [53](#)

C

Corporate Design

Einheitliches Erscheinungsbild eines Unternehmens. [76](#)

CRUD

”Create, Read, Update and Delete”: Die Zusammenfassung der Datenmanipulationsoperationen. [2](#), [3](#), [26](#), [33](#), [36](#), [38](#), [40](#), [53](#)

CSS

”Cascading Style Sheet”: Deklarative Sprache für Stilvorlagen von strukturierten HTML-Dokumenten. [62](#)

D

DOM

”Document Object Model”: Eine Spezifikation einer Schnittstelle für den Zugriff auf HTML-Dokumente. [56](#), [63](#)

H

Hash

Funktion, die eine Zeichenfolge beliebiger Länge auf eine Zeichenfolge mit fester Länge abbildet. [57](#)

HTML

”Hypertext Markup Language”: Textbasierte Auszeichnungssprache zur Strukturierung von Webseiten. [8](#), [56](#), [60](#), [62](#)

HTTPS

”Hypertext Transfer Protocol Secure”: Kommunikationsprotokoll um Daten abhörsicher zu übertragen. [8](#), [84](#)

J

JSF

”JavaServer Faces”: Ein Java Framework-Standard zur Entwicklung von grafischen Benutzeroberflächen für Webapplikationen. [51](#)

JSON

”JavaScript Object Notation”: Datenformat in einer einfach lesbaren Textform zum Zweck des Datenaustauschs zwischen Anwendungen. [60](#)

JVM

”Java Virtual Machine”: Virtuelle Umgebung für die Ausführung von Java-Programmen. [55](#)

K

Kovarianz

Mass an Zusammenhang zweier Variablen. [2](#), [28](#)

M

MVC

”Model View Controller”: Muster zur Strukturierung von Software-Entwicklung. [56](#)

P

PHP

”Hypertext Preprocessor”: Programmiersprache, die hauptsächlich zur Erstellung dynamischer Webseiten oder Webapplikationen verwendet wird. [51](#)

R

Request

Anfrage eines Clients an einen Server. [8](#)

responsive Design

Eine Darstellungart, die je nach Gerät eine optimale Ansicht gestaltet. [8](#), [62](#)

REST

”Representational State Transfer ”: Eine URL stellt genau eine serverseitige Aktion dar. [8](#)

S

SMTP

”Simple Mail Transfer Protocol”: Protokoll zum Austausch von E-Mails. [60](#)

SQL

”Structured Query Language”: Eine Datenbanksprache zur Definition von Datenstrukturen in relationalen Datenbanken sowie zum Bearbeiten (Einfügen, Verändern, Löschen) und Abfragen von darauf basierenden Datenbeständen. [3](#), [52](#), [53](#), [57](#)

SSL Zertifikat

Ein Datensatz mit dem Authentizität und Integrität eines Objekts geprüft werden kann. [8](#)

T

Template

Vorlage die mit Inhalt gefüllt werden kann. [56](#), [72](#)

Token

Methode zur Autorisierung von Benutzern. [8](#), [17](#), [19](#)

U

UML

”Unified Modeling Language”: Eine grafische Modellierungssprache zur Spezifikation, Konstruktion und Dokumentation von Systemen. [18](#), [66](#)

URL

”Uniform Resource Locator”: Identifiziert eine Ressource in Netzwerken. [5](#)

X

XML

”Extensible Markup Language”: Auszeichnungssprache zur Darstellung hierarchisch strukturierter Daten in Form von Textdateien. [56](#)