



BWOLF

EINE WEBBASIERTE PLATTFORM ZUR
EINSCHREIBUNG UND VERWALTUNG DES
EMPIRIEPRAKTIKUMS AN DER FSU JENA

Dokumentation

Christoph Keiner, Matthias Reuse, Ingo Schäfer, Christoph Staudt

13. Januar 2018

Inhaltsverzeichnis

1	Einleitung	1
1.1	Motivation	1
1.2	Aufgabenstellung	2
1.3	Aufbau der Arbeit	2
2	Anforderungsanalyse	3
2.1	Anforderungen an das System	3
2.1.1	Ablauf eines Praktikumsmoduls	4
2.1.2	Verschiedene Sichten im Überblick	5
2.2	Der Verteilungsalgorithmus	6
2.3	Technische Details	7
3	Entwurf	9
3.1	Datenmodell	9
3.2	Design	11
3.2.1	Frontend	12
3.2.2	Backend	13
4	Umsetzung	16
4.1	Verwendete Software	16
4.2	Frontend	16
4.3	Backend	16
5	Verteilungsalgorithmus	17
5.1	Überlegungen zum Algorithmus	17

6	Zusammenfassung	19
6.1	Stand des Projekts	19
6.2	Offene Probleme	19
6.3	Mögliche weitere Arbeiten	19
7	Anhang	20
7.1	Anhang - Entwurf	20

Kapitel 1

Einleitung

1.1 Motivation

Jedes Jahr (**Semester?**) wird an der Friedrich-Schiller-Universität Jena das Empiriepraktikum vom Institut für Psychologie angeboten. Dabei handelt es sich um eine Pflichtveranstaltung, die jeder Psychologie-Student absolviert haben muss. Innerhalb eines Praktikums werden mehrere Kurse angeboten. An einem dieser Kurse muss der Studierende teilnehmen, um das Modul erfolgreich abzuschließen. Allerdings ist die Teilnehmerzahl eines jeden Kurses begrenzt. So kann nicht jeder der über einhundert Studenten zu seinem Wunschkurs zugelassen werden. Die naheliegende Lösung ist die Zuweisung zu den Kursen nach Geschwindigkeit der Studenten bei der Anmeldung. Allerdings ist diese Verteilungsstrategie mit viel Stress und Streit verbunden. Die Studenten müssen anders auf die Kurse verteilt werden. Hierzu dient dem Institut für Psychologie eine Zuweisung zu den Kursen mittels Präferenzlisten. Jeder Studierende erstellt eine Liste mit Präferenzen, wie gerne er an welchen Kurs teilnehmen möchte. Es liegt auf der Hand, dass diese Listen nun nicht alle per Hand ausgewertet werden können, sondern automatisch erfolgen muss. Für diese Aufgabe existieren bereits Lösungen. So können mit dem Studienverwaltungstool der FSU Jena Friedolin Studenten bereits Präferenzlisten erstellt und ausgewertet werden. Bei der Verteilung mithilfe von Friedolin kommt es jedoch zu Problemen, da bei dem erstellen der Präferenzliste getrickst werden kann. Aus diesem Grund hat das Institut für Psychologie bereits eine eigene Platt-

form geschaffen, die das aufnehmen der Präferenzliste und das Verteilen auf die Kurse übernimmt. Hierbei ist aber die Verteilung auf die verschiedenen Kurse oft nicht zufriedenstellend und es muss manuell das Ergebniss angepasst werden. Das mit dieser Arbeit verbundene Projekt beschäftigt sich nun mit dem erstellen einer neuen Plattform für das Empiriepraktikum und der Frage, wie sich die Studenten besser auf die Kurse verteilen lassen.

1.2 Aufgabenstellung

Es gilt eine webbasierte Plattform für die Verwaltung des Empiriepraktikums der FSU Jena zu erstellen. Hauptaufgabe dieser Plattform ist, den verantwortliche Mitarbeitern des Instituts für Psychologie zu ermöglichen, für ein kommendes Semester ein neues Praktiumsmodul mit allen zugehörigen Kursen anlegen zu können. Des Weiteren sollen die Studenten die Möglichkeit erhalten eine Präferenzliste zu erstellen, welche der angebotenen Kurse sie am liebsten besuchen möchten. Nach Ablauf einer vorher festzulegenden Frist sollen die Studenten gemäß ihrer Präferenzliste dann automatisch bestmöglich auf die Kurse verteilt werden. Hierzu soll ein geeigneter Verteilungsalgorithmus entwickelt werden.

1.3 Aufbau der Arbeit

Zunächst werden in dieser Arbeit die Anforderungen an das die Plattform, sowie den Verteilungsalgorithmus ausgehend von der Aufgabenstellung aus Abschnitt 1.2 näher spezifiziert. Anschließend wird der Entwurf für die Umsetzung der Anforderungen an das zu entwickelnde System dargestellt.

Kapitel 2

Anforderungsanalyse

In diesem Kapitel werden die Anforderungen an die zu schaffende Plattform dargestellt. Zunächst wird im Abschnitt 2.1 behandelt, wie ein Praktikumsmodul abläuft und welche Erfordernisse sich daraus für die einzelnen Teilnehmergruppen ergeben. Darauf folgt eine Beschreibung der Aufgaben des Verteilungsalgorithmus. Zuletzt werden die technischen Anforderungen an das Projekt beschrieben.

2.1 Anforderungen an das System

Gemäß der allgemeinen Aufgabenstellung aus Abschnitt 1.2 existieren verschiedene Sichtweisen auf die Anforderungen des Systems. Zum einen die Sicht der Verantwortlichen für Praktikum und Kurse, zum anderen die der Studenten. Erstere teilt sich wiederum auf in den Blickwinkel der Dozenten der einzelnen Kurse und der übergeordneten Verantwortlichen für das Empiriepraktikum, im Weiteren Administratoren genannt. Im Folgenden werden zunächst die Anforderungen für Studenten, Dozenten und Administratoren anhand des chronologischen Ablaufs eines Praktikumsmoduls dargestellt. Anschließend wird nochmal für die jeweiligen Sichten eine kurze Übersicht gegeben, sowie weitere detailliertere Anforderungen genannt.

2.1.1 Ablauf eines Praktikumsmoduls

Der Ablauf beginnt, indem die Administratoren, nachdem sie sich in einer Login-Oberfläche angemeldet haben, ein neues Praktikumsmodul erstellen. Zu diesem Praktikumsmodul gehören neben generellen Informationen wie Name und Semester auch die besonderen Angaben, ab wann Studenten ihre Präferenzliste erstellen können und zu welchem Zeitpunkt die automatische Verteilung vorgenommen werden soll. Im Anschluss können die Dozenten, nachdem auch sie sich in einer entsprechenden Oberfläche angemeldet haben, ihre Kurse zu dem aktuellen Praktikumsmodul hinzufügen. Dabei sollen Kurse Angaben über Titel, Dozent, Teilnehmerzahl, Ort, Zeit, Beschreibung des Kurses und eine Literaturliste besitzen. Zusätzlich soll ein Kurs Informationen über den Lehrstuhl, sowie die Finanzierung erhalten, die jedoch nicht für Studenten einsehbar sein soll. Nachdem alle Dozenten ihre Kurse eingetragen haben, sollen die Administratoren die aktuelle Kursübersicht online stellen können, sodass jeder die Kurse einsehen kann. Besteht Interesse, dass Empiriepraktikum in diesem Semester zu absolvieren, so sollen sich die Studierenden registrieren können. Nach der Registrierung können die Studenten ihre Präferenzliste erstellen und speichern. Dabei soll die Präferenzliste auch jederzeit vom Studierenden noch verändert werden können. Nach jeder gespeicherten Änderung soll der Student eine Email-Benachrichtigung mit seiner aktuellen Präferenzliste erhalten. Nach Ablauf der zuvor von den Administratoren festgelegten Frist sollen die Studenten automatisch anhand ihrer aktuellen Version der Präferenzliste auf die Kurse verteilt werden. Diese Verteilung sollte möglichst gut sein, d.h. in Fall dieses Projekts eine möglichst gleichmäßige Verteilung der Studenten mit wenigen Ausreißern nach Unten. Nach dieser automatischen Verteilung soll eine Nachbearbeitungsphase folgen, in der die Administratoren den Algorithmus bei Bedarf mit vielleicht anderen Parametern neu starten können. Es soll jedoch auch die Möglichkeit geben, das Ergebnis der Verteilung manuell zu ändern. Nachdem das Ergebnis der Verteilung feststeht, sollen alle Beteiligten Studenten und Dozenten über das Ergebnis der Verteilung mittels einer Email-Benachrichtigung informiert werden. Ab diesem Moment sollen die Teilnehmer jedes Kurses für jeden sichtbar sein. Erwähnung der Tauschperiode? Nur für den Fall, dass wir sie wirklich am Ende implementieren

2.1.2 Verschiedene Sichten im Überblick

Sicht der Administratoren

Administratoren haben die Möglichkeit ein neues Praktikum zu erstellen. Dabei müssen sie Angaben über Name, Semester, Frist für die Anmeldung und Beginn der automatischen Verteilung festlegen. Sie sollen aktiv die Kursübersicht für das aktuelle Praktikum veröffentlichen können. Des Weiteren können Administratoren den Verteilungsalgorithmus in der Nachbearbeitungsphase erneut mit anderen Parametern starten und die Verteilungsergebnisse auch manuell verändern. Zusätzlich sollen Administratoren die angemeldeten Dozenten verwalten können, um z.B. Dozenten, die keine Kurse mehr anbieten zu entfernen. Auch sollen die Administratoren Kurse als „freibleiben“ markieren können, sowie die maximale und minimale Teilnehmerzahl für alle Kurse fließend einstellen können, falls nötig. Die Administratoren sollen für Praktikum auch die Möglichkeit erhalten Filterfunktionen anzuwenden und sich so z.B. die Kurse nach Lehrstühlen sortiert anzeigen zu lassen.

Sicht der Dozenten

Ein Dozent hat die Möglichkeit, Kurse für das aktuelle Praktikum zu erstellen und sich seine Kurse für ältere Praktika nochmals anzusehen, um sie evtl. als Vorlage für neue Kurse zu nutzen. Für einen Kurs müssen sie folgendes angeben:

- Name
- Titel
- Dozenten
- Zeit/Raum
- Teilnehmerzahl, hierbei können Dozenten nur zwischen 5 oder 10 wählen
- Kurzbeschreibung
- Beschreibung

- Literatur
- E-Mail Adresse des Empirie-Praktikums-Leiter

Folgendes ist nicht für den Studenten einsehbar:

- Lehrstuhl
- Lehrauftrag
- **Erstes Mal Empiriepraktikum?**

Der Inhalt von längeren Angaben wie der Beschreibung des Praktikums, soll in einer Textumgebung möglich sein, in der geeignete Textformatierung möglich ist. Außerdem sollen auch Bilder in die Beschreibung eingearbeitet werden können. Des Weiteren sollen Dozenten nur ihre eigenen Kurse editieren können. Nachdem die Studenten verteilt worden sind, sollen die Dozenten eine E-Mail mit den Studenten, die in ihrem Kurs sind, erhalten. **Im Verlaufe des Semesters können Dozenten ihren Kurs in zwei Kurse aufteilen.**

Sicht der Studenten

Die Studenten sollen ohne eine Registrierung die Kursübersicht aufrufen können, sobald die Administratoren die Kursübersicht veröffentlicht haben. Nach der Registrierung sollen die Studenten bis zu einer Frist eine Präferenzliste erstellen und späterhin auch bearbeiten können. Bei jeder angenommenen Änderung der Präferenzliste soll ein Studierender über seine aktuelle Wahl per Email informiert werden. Nachdem die Verteilung vom Algorithmus vorgenommen wurde, sollen die Studenten über ihr Ergebnis informiert werden, sowie die Ergebnisse der gesamten Verteilung einsehen können.

2.2 Der Verteilungsalgorithmus

Der Verteilungsalgorithmus verteilt alle Studenten auf die Kurse. Dabei ist es wichtig, dass die aufaddierte maximale Teilnehmeranzahl der Kurse größer ist als die Anzahl der zu verteilenden Studenten. Sollte dem nicht der Fall sein, so

wird der Administrator informiert. Dieser hat dann die Möglichkeit die maximale Teilnehmeranzahl von Kursen zu erhöhen. Die Eingabe des Algorithmus sind die Gewichte der Präferenzen und (optional) die Auswahl eines Optimierungsalgorithmus.

Startet der Algorithmus, so versucht er, für jeden Studenten die größte, mögliche Präferenz zu den Kursen zu wählen. Dabei ist insbesondere wichtig, dass die Streuung der gewählten Präferenzen möglichst gering ist. Dies erfolgt beispielsweise durch eine Gewichtung der Präferenzen.

Weiterhin platziert der Algorithmus in jeden Kurs mindestens drei Studenten, damit der Kurs sinnvoll angeboten werden kann. Es werden aber nie mehr Teilnehmer einem Kurs zugeordnet als die maximale Teilnehmeranzahl des Kurses vorgibt.

Hat der Algorithmus schließlich eine passende Zuordnung von Studenten zu Kursen gefunden, so wird der Administrator zuerst benachrichtigt. Dieser kann die Ergebnisse bearbeiten, Parameter des Algorithmus neu einstellen, Kurse aktivieren/deaktivieren und ihn erneut starten. Ist der Administrator zufrieden mit der Verteilung, so bestätigt dieser das Ergebnis. Anschließend wird an die Dozenten eine E-Mail mit ihren Teilnehmern geschickt, und Studenten erhalten eine E-Mail mit ihrem Kurs.

2.3 Technische Details

Der Login wird über eine E-Mail/Passwort Authentifizierung realisiert. Dafür werden nur E-Mail Adressen der FSU Jena nutzbar sein. Dies bedeutet, dass E-Mails auf ”@uni-jena.de”enden müssen. Man kann sich zu jeder Zeit ein- und ausloggen.

Bei der Wahl der Präferenzen darf keine Präferenz doppelt belegt werden, d.h. jede Präferenz hat einen eindeutigen Kurs.

Der Verteilungsalgorithmus kann mit verschiedenen Optionen angesteuert werden. Optionen können u.a. die Gewichtung der Varianz, oder welche Kurse belegt werden können, sein. Weiterhin ist der Verteilungsalgorithmus innerhalb von 24

Stunden fertig.

- Man kann sich zu einem beliebigen Zeitpunkt einloggen und ausloggen.
- Der Verteilungsalgorithmus braucht maximal 24 Stunden.
- Die Website muss auch auf mobilen Endgeräten funktionieren.
- Sollte ein Browser älter als IE11 sein, so wird den Nutzer angezeigt, dass sie ihren Browser updaten sollen.
- Die Website wird auf einer docker-compose Umgebung aufgesetzt.
- Die Website wird mit OctoberCMS mit Laravel 5.5 umgesetzt.
- Der Webserver wird mit Nginx umgesetzt.
- Die Datenbank wird mit MySQL umgesetzt.
- Der Cache-Server wird mit Redis umgesetzt.
- Tests erfolgen für die wesentlichsten Bestandteile, insbesondere für den Algorithmus.
- Kursinformationen sind öffentlich einsehbar.
- Kurse können von nicht-Studenten insbesondere nach folgenden Kriterien gefiltert werden:
 - Lehrstuhl
 - Finanzierung/Lehrauftrag
 - Erstes Mal Empiriepraktikum?

Kapitel 3

Entwurf

Nachdem im vorangegangenen Kapitel die Anforderungen für das System spezifiziert wurden, soll in diesem Kapitel der Entwurf der verschiedenen Sichten dargestellt werden. Zunächst wird das dem System zugrundeliegende Datenmodell dargestellt. Daraufgehend wird das Design für die webbasierte Plattform erläutert.

3.1 Datenmodell

Der Entwurf für das Datenmodell ist in Abbildung 3.1 zu sehen. Es besteht aus sieben Tabellen. Die Tabelle *bwolfjena_core_modules* beinhaltet die verschiedenen Empiriepraktika. Der Name Module für die Tabelle wurde gewählt, da das System bei Bedarf auch auf andere ausgewählte Module als das Empiriepraktikum erweitert werden kann.

In der Tabelle *bwolfjena_core_courses* sind die verschiedenen Kurse gespeichert. Ein Kurs gehört immer zu einem bestimmten Modul beziehungsweise einem Empiriepraktikum. Um das abzubilden, ist für jeden Kurs mittels eines Fremdschlüssels das Empiriepraktikum, zu dem er gehört, gespeichert.

Ein Kurs wird immer von einem Lehrstuhl angeboten, die in der Tabelle *bwolfjena_core_chairs* abgelegt sind. Auch auf diese Tabelle existiert wieder ein entsprechender Fremdschlüssel in *bwolfjena_core_courses*.

Alle angemeldeten Studenten werden in der Tabelle *users* gespeichert. Jeder der

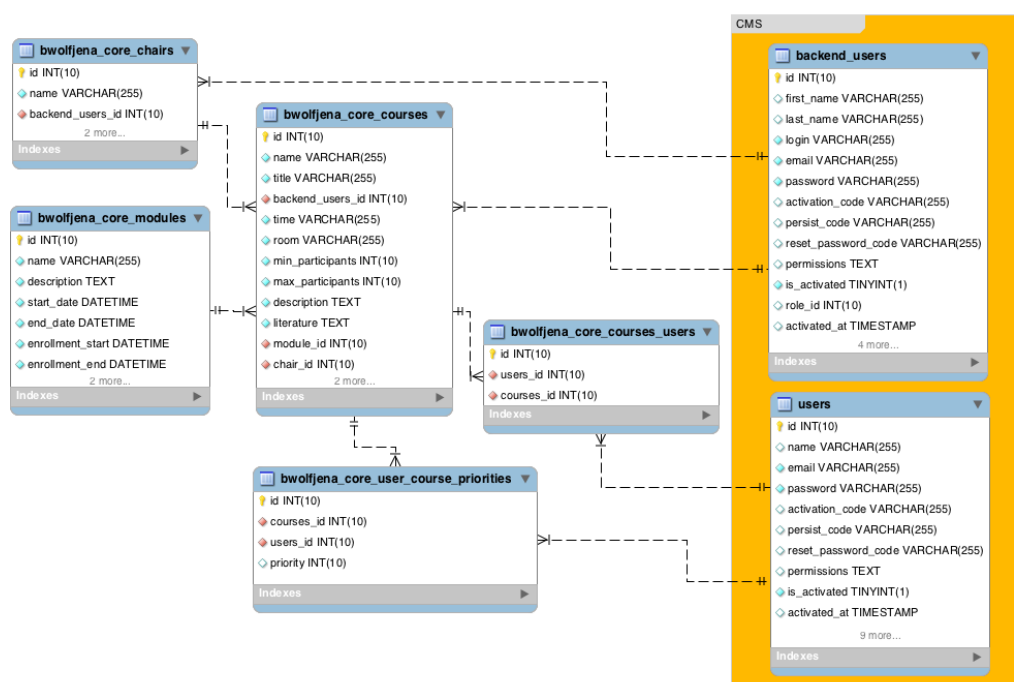


Abbildung 3.1: Entwurf des Datenmodells. Farbcode für Spalten der Tabellen: Gelb Primärschlüssel, Rot Fremdschlüssel, Blau gewöhnliche Variable, Weiß nul-lable Variable






	Eingabefeld
	Button
	Drag&Drop-Element
	Textfeld
	Optisches Element

Tabelle 3.1: Farbcode der Mock-Ups

Studenten hat eine Präferenzliste. Die Einträge der Präferenzliste für jeden Studenten und jeden Kurs werden in Tabelle *bwolfjena_core_priorities* gespeichert. Dementsprechend beinhaltet *bwolfjena_core_priorities* Fremdschlüssel von *users* und *bwolfjena_core_courses*.

In der Tabelle *bwolfjena_core_users* ist abgelegt, welcher Student welchem Kurs zu geteilt wurde. Aus diesem Grund beinhaltet *bwolfjena_core_users* Fremdschlüssel auf *bwolfjena_core_courses* und *users*.

Die letzte Tabelle *backend_users* beinhaltet die Backendbenutzer. Damit sind die Dozenten und Administratoren gemeint. In *bwolfjena_core_courses* ist ein Fremdschlüssel für die Backendbenutzer gespeichert, um den Dozenten anzugeben, der den Kurs leitet. Auch die Tabelle *bwolfjena_core_chairs* hat einen Fremdschlüssel zu der Tabelle *backend_users*, um den Lehrstuhlinhaber anzugeben.

Neben den beschriebenen Fremdschlüsseln existieren für die Einträge in jeder Tabelle IDs als Primärschlüssel. Die weiteren Spalten der Tabellen sind entsprechend den Anforderung aus Kapitel 2 gewählt.

3.2 Design

Für den Desing-Entwurf der Seite wurden Mock-Ups erstellt, die den groben Aufbau der Website mit den entsprechenden Funktionen zeigen. Im folgenden werden diese Mock-Ups für das sogenannte Frontend, also aus Sicht der Studenten, und für das Backend, die Sicht der Dozenten beziehungsweise der Administratoren, vorgestellt. Dabei gilt der in Tabelle 3.1 angegebene Farbcode für die verschiedenen Elemente der Mock-Ups.

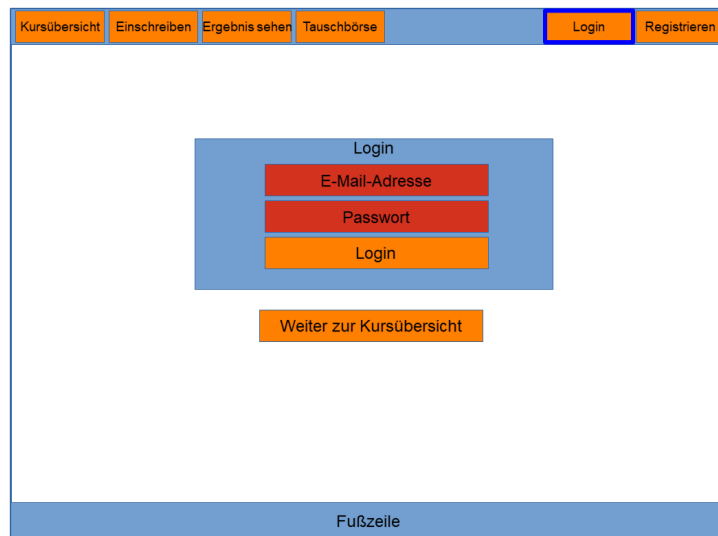


Abbildung 3.2: Entwurf für die Login-Oberfläche

3.2.1 Frontend

Wie in Kapitel 2 bereits ausgeführt, sollen die Studenten zunächst eine Registrierungs- bzw. Login-Oberfläche sehen. Jedoch sollen die verschiedenen Kurse auch ohne eine Anmeldung einsehbar sein.

In Abbildung 3.2 werden beide Anforderungen umgesetzt. Zum einen die Login-Oberfläche, in der in zwei Textfeldern Benutzername und Passwort für einen erfolgreichen Login eingetragen werden müssen, zum anderen die direkte Weiterleitung zur Kursübersicht als einfacher Knopf darunter.

Die Kopfzeile umfasst einige Reiter. So kann mithilfe der Kopfzeile auf die verschiedenen Funktionen des Frontends gewechselt werden. Dazu zählen die Kursübersicht, die Erstellung der Präferenzliste, das Einsehen der Verteilungsergebnisse, sowie das Tauschen von Kursen. Außerdem soll in der Kopfzeile ein Schalter zum Abmelden aus dem System bereitgestellt werden, sobald ein Benutzer die sich erfolgreich eingeloggt hat. Auf allen weiteren Seiten soll die Kopfzeile die gleiche Funktion und das gleiche Aussehen haben. Unter der Kopfzeile folgt eine textuelle Erklärung des Ablaufs des Empiriepraktikums und der für die Studenten relevanten Schritte, um sich erfolgreich für die Kurse einzuschreiben. Wie in Abbildung 7.3 dargestellt, werden darunter die verschiedenen Kurse angezeigt,



Abbildung 3.3: Entwurf für die Kursdetails

jeweils mit einer kurzen Beschreibung. Durch einen Klick auf ein Kursfeld, sollen sich die detaillierten Informationen zu dem Kurs einsehen lassen. Das Desing hierfür ist in Abbildung 3.3 zu sehen und bedarf keiner weiteren Erläuterung. Die Fußzeile soll weitere allgemeine Informationen bereitstellen, sofern diese von Nöten sein sollten, aber vor allem als optischer Abschluss der Seite dienen.

Die Seite für die Einschreibung in die Kurse soll wie in Abbildung 3.4 gestaltet sein. Wieder bilden Kopfzeile und Fußzeile den Rahmen der Seite. Unter der Kopfzeile befindet sich auch hier eine kurze Erklärung, wie man die Präferenzliste genau erstellt. Das Erstellen soll im Feld *Präferenzliste* über ein "Drag&Drop"-System vorgenommen werden und über den Knopf *Absenden* fixiert werden können.

3.2.2 Backend

Administratoren

Dozenten

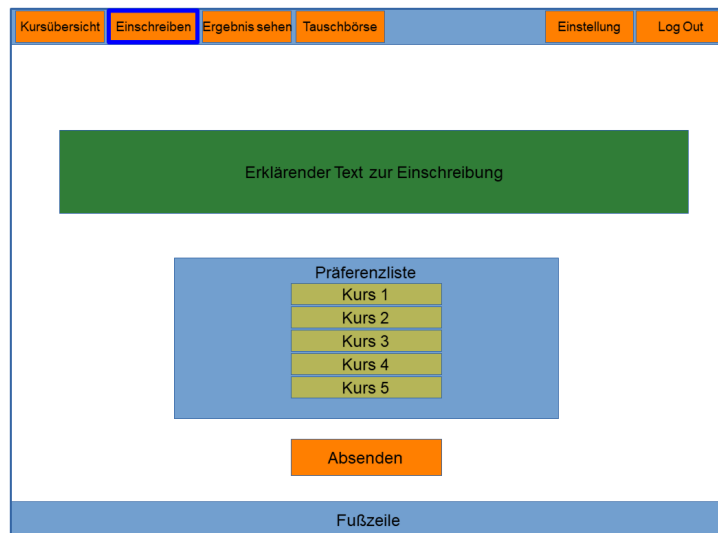


Abbildung 3.4: Entwurf für die Einschreibungs-Oberfläche

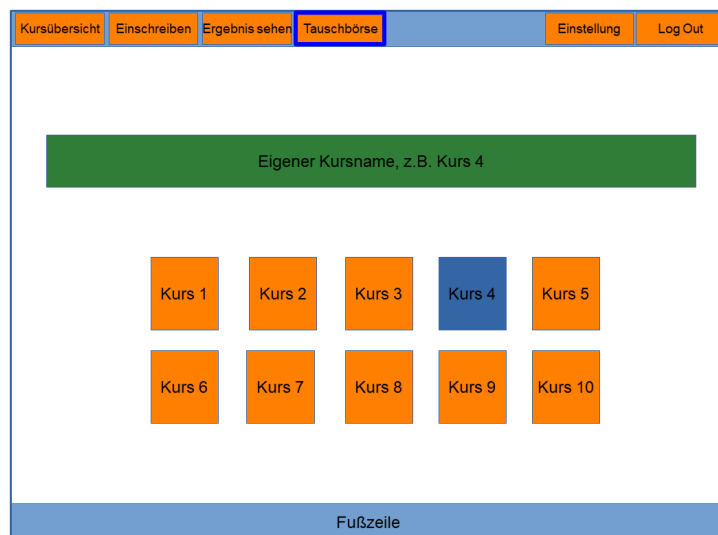


Abbildung 3.5: Entwurf für die Tauschauswahl

Kursübersicht	Einschreiben	Ergebnis sehen	Tauschbörse		Einstellung	Log Out
<div>Tauschanfrage an {Kursname}</div> <div>Begründung</div> <div><div>Absenden</div><div>Cancel</div></div>						
Fußzeile						

Abbildung 3.6: Entwurf für die Tauschanfrage

Kapitel 4

Umsetzung

4.1 Verwendete Software

4.2 Frontend

4.3 Backend

Kapitel 5

Verteilungsalgorithmus

5.1 Überlegungen zum Algorithmus

Die Grundlegende Idee der Zielfunktion hat die Form:

$$\max \text{ Summe der Prioritäten} - \text{Gewicht} \cdot \text{Varianz} \quad .$$

Genauer ausformuliert ergibt sich:

$$\max \sum_{i=1}^n \sum_{j=1}^m c(i, j) x_{ij} - \frac{\beta}{n} \sum_{i=1}^n \left[\left(\sum_{j=1}^m c(i, j) x_{ij} \right) - \frac{1}{n} \sum_{i=1}^n \sum_{j=1}^m c(i, j) x_{ij} \right]^2 \quad ,$$

wobei gilt:

- n - Anzahl der Studenten
- m - Anzahl der Kurse
- $c(i, j)$ - Priorität von Student i für Kurs j
- β - Gewichtung der Varianz
- $t_{\min}(j)$ - Minimale Anzahl der Teilnehmer für Kurs j
- $t_{\max}(j)$ - Maximale Anzahl der Teilnehmer für Kurs j .

Zusätzlich sind drei Nebenbedingungen notwendig, um das Problem angemessen darzustellen. Zum einen sollen die x_{ij} nur die Werte 0 oder 1 annehmen können:

$$x_{ij} \in \{0, 1\} \quad .$$

Des Weiteren soll jeder Student nur einem Kurs zugeteilt werden:

$$\forall i \in \{1, \dots, n\} : \sum_{j=1}^m x_{ij} = 1 \quad .$$

Zuletzt ist die Teilnehmerzahl für die Kurse begrenzt:

$$\forall j \in \{1, \dots, m\} : t_{\min}(j) \leq \sum_{i=1}^n x_{ij} \leq t_{\max}(j) \quad .$$

Kapitel 6

Zusammenfassung

6.1 Stand des Projekts

6.2 Offene Probleme

6.3 Mögliche weitere Arbeiten

Kapitel 7

Anhang

7.1 Anhang - Entwurf

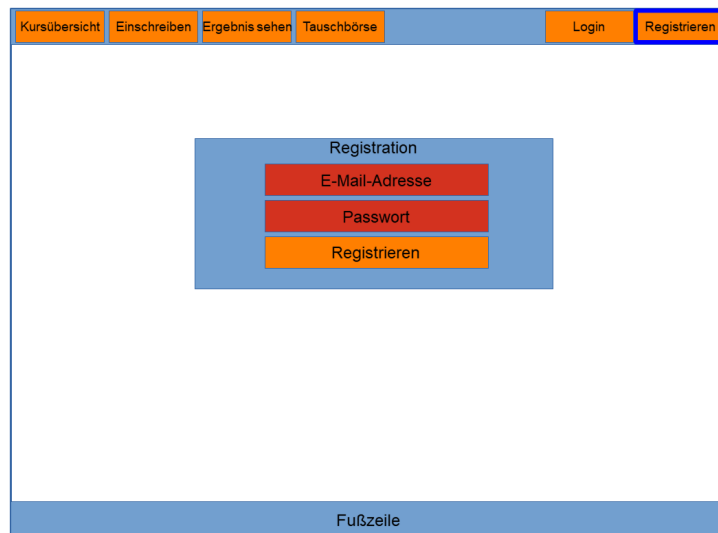


Abbildung 7.1: Entwurf für Registrierungs-Oberfläche



Abbildung 7.2: Entwurf für die Ergebnisübersicht

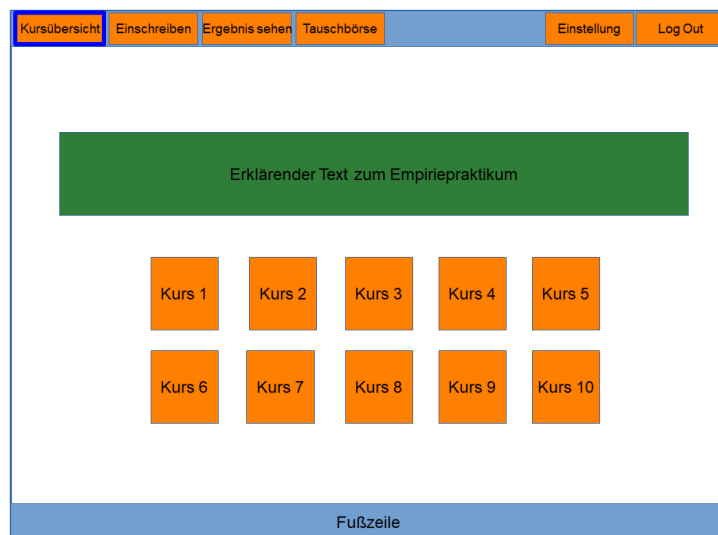


Abbildung 7.3: Entwurf für die Kursübersicht