BWolf

Eine webbasierte Plattform zur Einschreibung und Verwaltung des Empiriepraktikums an der FSU Jena

Dokumentation

Christoph Keiner, Matthias Reuse, Ingo Schäfer, Christoph Staudt

16. Dezember 2017

Inhaltsverzeichnis

Kapitel 1

Anforderungsanalyse

1.1 Allgemeine Problemstellung

Im Zuge des Projektes gilt es eine webbasierte Plattform zu schaffen, die die Verwaltung des Empiriepraktikums der FSU Jena ermöglicht. Hauptaufgabe dieser Plattform ist, den verantwortliche Mitarbeitern der FSU Jena 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.

1.2 Genaue Anforderungen an das System

Gemäß der allgemeinen Problemstellung aus Abschnitt 1.1 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 Ab-

laufs eines Praktikumsmoduls dargestellt. Anschließend wird nochmal eine kurze Übersicht über die Anforderungen der jeweiligen Sichtweisen gegeben.

1.2.1 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. 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 soll 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

1.2.2 Verschiedene Sichten im Überblick

Sicht der Administratoren

Die Administratoren haben die Möglichkeit, neue Dozenten zu erstellen, diese zu bearbeiten und zu löschen. Zusätzlich ist er dafür verantwortlich, die Ergebnisse des Verteilungsalgorithmus zu verändern und zu bestätigen.

Sicht der Dozenten

Ein Dozent hat die Möglichkeit, Kurse für das Praktikum zu erstellen. Für einen Kurs müssen sie folgendes angeben:

- Name
- Titel
- Dozenten
- Zeit/Raum
- Maximale Teilnehmer (5 oder 10!)
- Kurzbeschreibung
- Beschreibung
- Literatur
- E-Mail Adresse des Empirie-Praktikums-Leiter

Folgendes ist nicht für den Studenten einsehbar:

- Lehrstuhl
- Lehrauftrag
- Erstes Mal Empiriepraktikum?

Nachdem die Studenten verteilt worden sind, erhalten die Dozenten eine E-Mail mit den Studenten, die in ihrem Kurs sind. Im Verlaufe des Semesters können Dozenten ihren Kurs in zwei Kurse aufteilen.
Sollten Dozenten zu einem anderem Semester erneut Kurse anbieten, so haben sie die Möglichkeit, ihre alten Kurse einzusehen und (ohne Änderung des alten Kurses) einen dieser als Vorlage für einen neuen Kurs zu nutzen.

Sicht der Studenten

1.3 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 insbesonders wichtig, dass die Streuung der gewählten Präferenzen möglichst gering ist. Dies erfolgt beispielsweise durch eine Gewichtung der Präferenzen.
Weiterhin platziert der Algortihmus 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.

1.4 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 2

Entwurf

Nachdem im vorangegangenem Kapitel die Anforderungen für das System spezifiziert wurden, sollen in diesem Kapitel die Überlegungen zum Design der verschiedenen Sichten dargestellt werden.

2.1 Datenmodell

2.2 Design

2.2.1 Frontend

Wie zuvor 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 2.1 werden beide Anforderungen umgesetzt. Zum einen die Login-Oberfläche, in der in zwei Textfelder Benutzername und Passwort für einen erfolgreichen Login eingetragen werden müssen, zum anderen die direkte Weiterleitung zur Kursübersicht als einfacher Knopf darunter.
Die Kursüberischt, die sowohl auf die Weiterleitung als auch auf die Anmeldung folgt, ist in Abbildung 2.2 zu sehen. 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

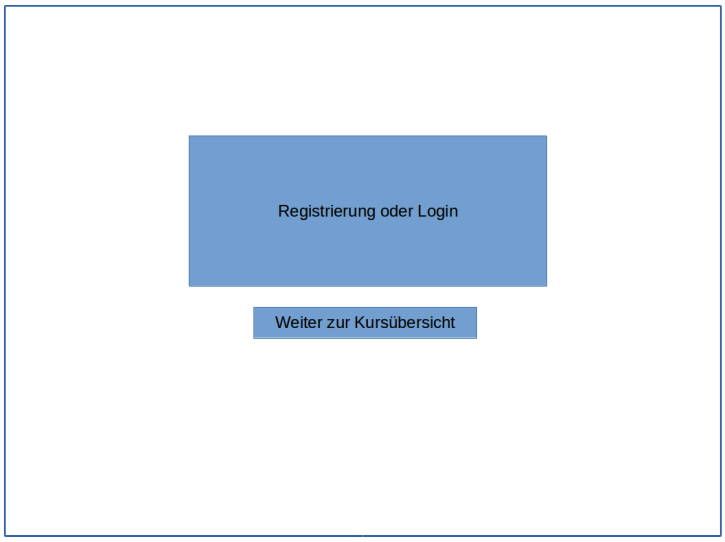

Abbildung 2.1: Entwurf für die Login-Oberfläche

Einsehen der Verteilungsergebnisse, sowie das Tauschen von Kursen. Außerdem soll in der Kopfzeile ein Schalter zum Abmelden aus dem System bereitgestellt werden und eine Anzeige, ab wann das Ende der Einschreibungs- und Tauschphase erreicht ist. 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 2.2 dargestellt, werden darunter die verschiedenen Kurse angezeigt, 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 **??** 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 2.4 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.

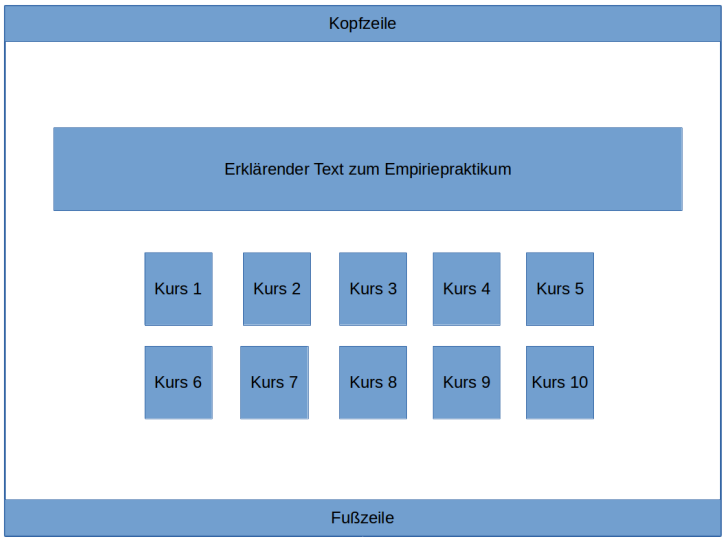

Abbildung 2.2: Entwurf für die Kursübersicht

2.2.2 Backend

Administratoren

Dozenten

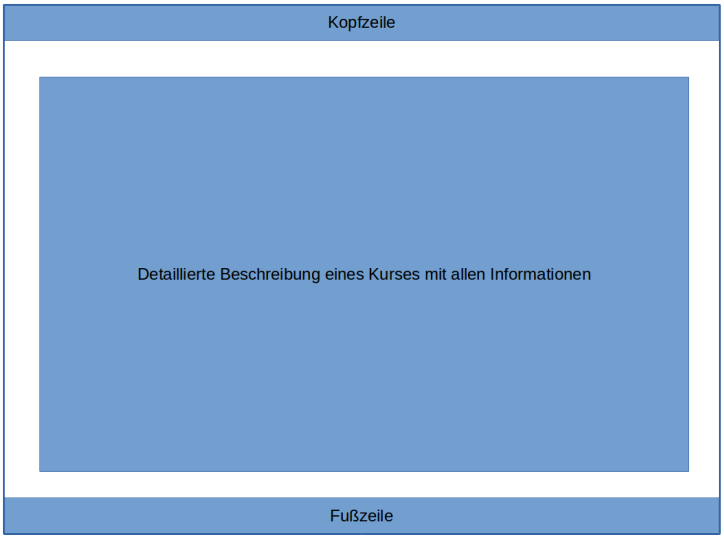

Abbildung 2.3: Entwurf für die Kursdetails

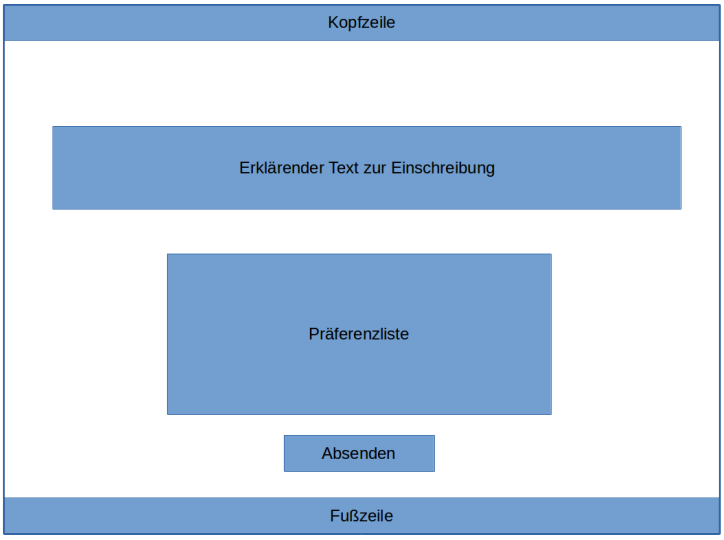

Abbildung 2.4: Entwurf für die Einschreibungs-Oberfläche

Kapitel 3

Umsetzung

3.1 Verwendete Software

3.2 Frontend

3.3 Backend

Kapitel 4

Verteilungsalgorithmus

4.1 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:

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

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:

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

Zuletzt ist die Teilnehmerzahl für die Kurse begrenzt:

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

Kapitel 5

Zusammenfassung

5.1 Stand des Projekts

5.2 Offene Probleme

5.3 Mögliche weitere Arbeiten