



**BWOLF**

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

## **Dokumentation**

*Christoph Keiner, Matthias Reuse, Ingo Schäfer, Christoph Staudt*

17. Dezember 2017

# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>1</b>
<b>2</b>	<b>Anforderungsanalyse</b>	<b>2</b>
2.1	Allgemeine Problemstellung . . . . .	2
2.2	Genaue Anforderungen an das System . . . . .	2
2.2.1	Ablauf eines Praktikumsmoduls . . . . .	3
2.2.2	Verschiedene Sichten im Überblick . . . . .	4
2.3	Der Verteilungsalgorithmus . . . . .	6
2.4	Technische Details . . . . .	6
<b>3</b>	<b>Entwurf</b>	<b>8</b>
3.1	Datenmodell . . . . .	8
3.2	Design . . . . .	8
3.2.1	Frontend . . . . .	8
3.2.2	Backend . . . . .	10
<b>4</b>	<b>Umsetzung</b>	<b>12</b>
4.1	Verwendete Software . . . . .	12
4.2	Frontend . . . . .	12
4.3	Backend . . . . .	12
<b>5</b>	<b>Verteilungsalgorithmus</b>	<b>13</b>
5.1	Überlegungen zum Algorithmus . . . . .	13
<b>6</b>	<b>Zusammenfassung</b>	<b>15</b>
6.1	Stand des Projekts . . . . .	15

6.2	Offene Probleme . . . . .	15
6.3	Mögliche weitere Arbeiten . . . . .	15

# **Kapitel 1**

## **Einleitung**

# **Kapitel 2**

## **Anforderungsanalyse**

### **2.1 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.

### **2.2 Genaue Anforderungen an das System**

Gemäß der allgemeinen Problemstellung aus Abschnitt 2.1 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 Ablaufs

eines Praktikumsmoduls dargestellt. Anschließend wird nochmal für die jeweiligen Sichten eine kurze Übersicht gegeben, sowie weitere detailliertere Anforderungen genannt.

### **2.2.1 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. 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**

## **2.2.2 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.



## 2.3 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 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.

## 2.4 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. je-

de 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, sollen in diesem Kapitel die Überlegungen zum Design der verschiedenen Sichten dargestellt werden.

### 3.1 Datenmodell

### 3.2 Design

#### 3.2.1 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 3.1 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 Kursübersicht, die sowohl auf die Weiterleitung als auch auf die Anmeldung folgt, ist in Abbildung 3.2 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

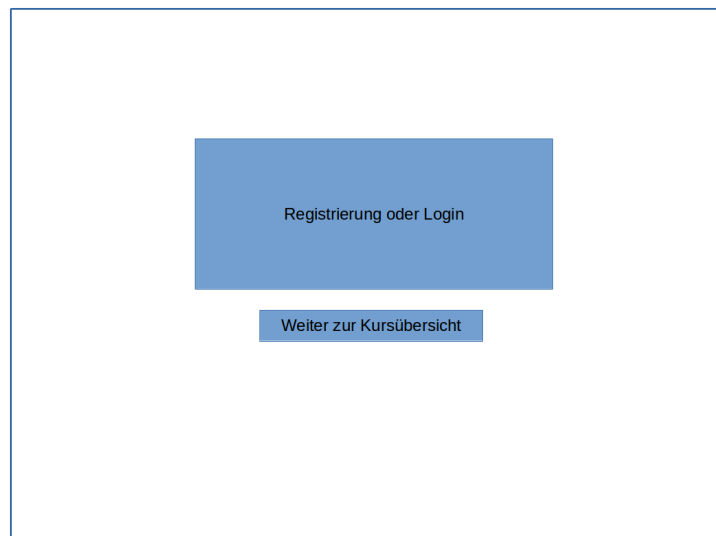


Abbildung 3.1: Entwurf für die Login-Oberfläche

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 3.2 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 Design 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 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.

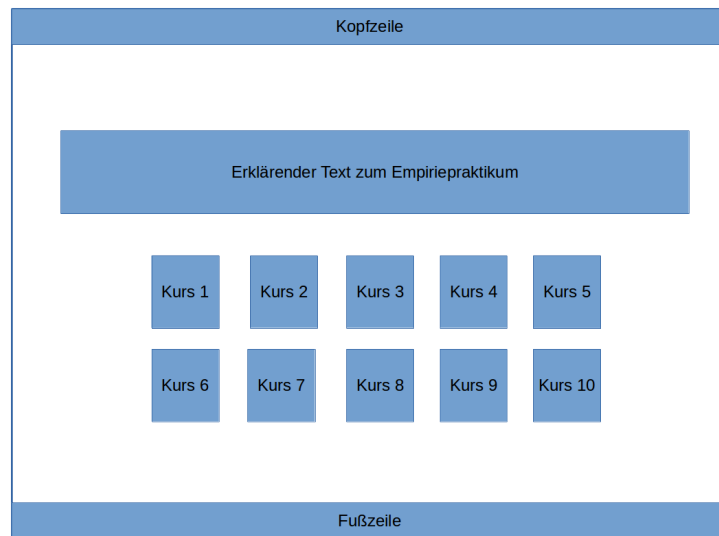


Abbildung 3.2: Entwurf für die Kursübersicht

### 3.2.2 Backend

**Administratoren**

**Dozenten**



Abbildung 3.3: Entwurf für die Kursdetails

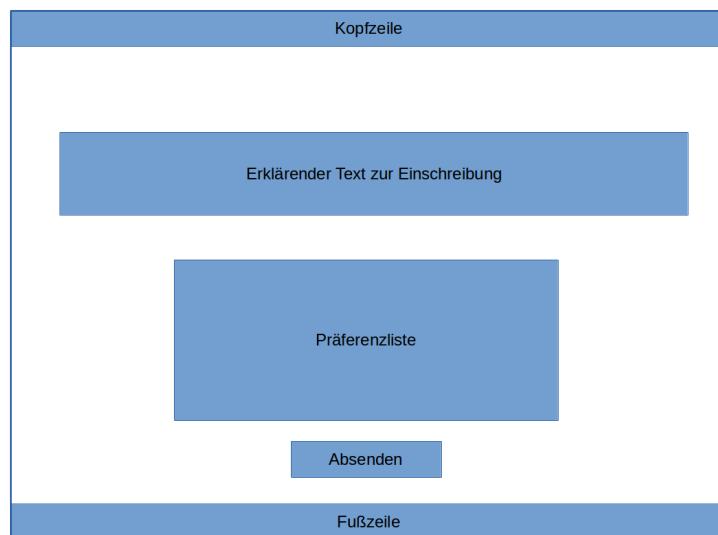


Abbildung 3.4: Entwurf für die Einschreibungs-Oberfläche

# **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:

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

wobei gilt:

- $n$  - Anzahl der Studenten
- $m$  - Anzahl der Kurse
- $c(i, j)$  - Priorität von Student  $i$  für Kurs  $j$
- $\beta$  - 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**