



SEP – Sommersemester 2018

Kurs- und Dozentenbewertungssystem

Lastenheft

Christian Bachmaier, Thomas Bock, Kolja Thormann

## 1 Motivation

Das Abhalten eines guten Kurses ist eine Herausforderung. Es gibt dabei verschiedene Zielsetzungen, z.B. möglichst viel Wissen oder möglichst nur das Wissen, was man sich merken kann, zu übermitteln. Der Dozent ist dabei eine Art Entertainer. Natürlich hängt der Spaß- und Lernfaktor auch von den Inhalten und deren Schwierigkeitsgrad ab, sowie auch vom Wissen, der Motivation, dem Zeitbudget, der Benotungshärte und der Fähigkeit des Dozenten, Wissen zu übermitteln. Das System zur Bewertung von Kursen und Dozenten soll potentiellen Kursteilnehmern helfen, die richtigen Kurse zu wählen und Dozenten wichtiges Feedback zu ihren Veranstaltungen geben.

## 2 Aufgabenstellung

Ziel des Software-Engineering-Praktikums im Sommersemester 2018 ist es, eine Webapplikation zur Bewertung von Kursen und/oder deren Dozenten zu entwickeln. Um die Anwendung einfach betreiben und verwalten zu können, sollte sie auf einem eigenen Webspaces gehostet und ausgeführt werden können. Die Anwendung sollte für einen Betreiber leicht einzurichten sein. Außerdem sollte dieser die Möglichkeit haben, gewisse Konfigurationen der Anwendung vornehmen zu können, wie zum Beispiel

das Festlegen des Logos, des Namens der betreibenden Organisation, etc. Da nicht angenommen werden kann, dass der Betreiber der Anwendung alleine eine Plattform mit hohem Traffic auch inhaltlich verwalten kann, soll er die Möglichkeit haben, einen oder mehrere Manager/Administratoren festzulegen, die bei Problemen (z.B. rechtlich nicht einwandfreien Postings, Denunzierung oder User-Account-Problemen) einschreiten können. Anonyme Benutzer sollten die Möglichkeit haben, nach Kursen und Dozenten zu suchen/browsen, aber nur eine numerische Durchschnittsbewertung sehen. Konkrete Bewertungen einzusehen oder zu verfassen soll nur angemeldeten registrierten Benutzern möglich sein. Zur Gruppierung werden Kurse und Dozenten einer oder mehrerer Bildungseinrichtungen/Organisationen zugeordnet.

Die zu erstellende Web-Anwendung soll intuitiv zu bedienen sein, geringe Anwendungshürden aufweisen, zugleich aber fortgeschrittenen Benutzern eine Vielzahl von erweiterten Funktionen mit einfacher Bedienung bieten.

Ein Beispiel für ein ähnliches und bereits existierendes System ist *MeinProf.de*<sup>1</sup> für Studenten und Dozenten von Hochschulen. Dieses Beispiel geht jedoch teilweise deutlich über die hier geforderten Funktionalitäten hinaus.

Die Anwendung muss mit Java und JavaServer Faces (JSF) entwickelt werden. Auf Benutzerseite kann ein moderner Web-Browser vorausgesetzt werden.

### 3 Produkteinsatz

Zur Zielgruppe der zu entwickelnden Anwendung gehören als Betreiber verschiedenste Arten von Bildungseinrichtungen, Non-Profit-Organisationen oder auch private Personen mit eigenem Webspace. Das System wird oft frei zugänglich betrieben werden, aber auch ein Modus nur für eine geschlossene Benutzergruppe soll vorhanden sein, um es z.B. innerhalb einer Schule zu ermöglichen. Hier kann der Administrator das Anlegen von Kursen und Dozenten durch Benutzer zentral verhindern und Benutzer können sich nicht mehr selbst (ohne weitere Maßnahmen wie das Überprüfen von E-Mail-Adressen auf Zugehörigkeit) registrieren.

## 4 Produktfunktionen

### 4.1 Begriffsdefinitionen

Es gibt mehrere Gruppen von Benutzern (= Akteure) des Systems:

**Betreiber/Administrator:** Ist für die Konfiguration des Systems verantwortlich. Dieser kann beispielsweise das Logo oder den Namen der Webapplikation festlegen. Zudem kann er weitere Einstellungen, wie die zu verwendende Datenbank oder ob das System frei zugänglich ist, vornehmen. Eine wichtige Aufgabe des Betreibers/Admins ist die Userverwaltung, z.B. Anlegen, Editieren und Löschen. Ein Admin kann wählen, ob nur bestimmte registrierte Nutzer (= Moderatoren) Bildungseinrichtungen, Kurse und Dozenten anlegen dürfen oder alle.

---

<sup>1</sup><http://www.meinprof.de/>

**Moderator:** Ist für alle Verwaltungsaufgaben einer oder mehrerer Bildungseinrichtungen zuständig, welche er selbst erstellt hat oder zu der er als ein Moderator ernannt wurde. Moderatoren können (für zugeordnete Bildungseinrichtungen) andere Moderatoren ernennen oder absetzen. Desweiteren können sie Bewertungen anderer Teilnehmer in einer ihnen zugeordneten Einrichtung löschen oder editieren, z.B. wenn sich diese nicht an Benimmregeln o.ä. halten sollten. Es gibt auch Moderatoren, die für mehr als eine Einrichtung zuständig sind. Moderatoren können auch Dozenten der eigenen Einrichtung verwalten.

**Registrierter Benutzer:** Hat die Möglichkeit, sich im System anzumelden und danach Einzelbewertungen zu sehen. Zudem kann er Einrichtungen, Kurse und Dozenten erstellen (siehe Konfigurationsoption des Administrators), oder Bewertungen (numerisch und mit Text) für Kurse/Dozenten abgeben. Hierfür muss sich ein registrierter Benutzer zuvor im System mit einer gültigen E-Mail-Adresse registrieren. Benutzer dürfen nur Daten (außer einem Nickname) anderer Benutzer erfassen, die explizit zur Anzeige von diesen freigegeben wurden.

**Anonymer Benutzer:** Hat die Möglichkeit, sich im System zu bewegen und numerische Durchschnittsbewertungen von Dozenten in Bildungseinrichtungen zu lesen. Die Suchfunktion steht ihm dabei zur Verfügung.

## 4.2 Minimale Anforderungen

Im Folgenden ist mit Erstellen gleichzeitig auch nachträgliches Editieren und Löschen gemeint.

**Toplisten** Es gibt ein Ranking der Bildungseinrichtungen und Toplisten: Topkurse und Topdozenten, und Listen mit Veranstaltungen pro Dozent

**Kennungsart** Es wird zwischen Kursteilnehmer- und Dozentenkennungen unterschieden. Dozenten (sowie Moderatoren) können auch passwortgeschützte Kurse anlegen und Kommentare zu erhaltenen Bewertungen (nur eigene) abgeben.

**Geschlossene Gruppen** Das System ist so konfigurierbar, dass es von einer Organisation wie z.B. einer Universität geschlossen nur für ihre Mitglieder betrieben werden kann. Einrichtungen wären dann z.B. die Fakultäten.

**Infrastrukturangaben** Erstellung einer Einrichtung mit zugehörigen Dozenten und Kursen, die von einem oder mehreren Dozenten abgehalten werden. Einrichtungen haben einen Namen und einen optionalen Beschreibungstext. Kurse an Einrichtungen haben einen Titel und ggf. einen Zeitraum, wie z.B. ein Semester, einen optionalen Beschreibungstext und eine optionale URL auf Originalwebseiten des Kurses. Dozenten haben zumindest einen Namen, ein optionales Bild und eine optionale URL auf deren Homepage.

**Bewertung** Erstellen einer Bewertung mit mindestens der Angabe des numerischen Werts. Optional kann ein Fließtext dazu erstellt werden. Bei anonymer Abgabe ist der Ersteller der Bewertung nur für Administratoren und Moderatoren

ersichtlich. Ein Bewerter kann angeben, ob die Bewertung anonym oder mit Erstellerdaten erfolgt, die in seinem Profil freigeschaltet sind. Jeder Teilnehmer kann pro Kurs/Dozent nur eine Bewertung abgeben, die er aber nachträglich jederzeit editieren kann. Jede Bewertung hat ein ersichtliches Erstell-/Bearbeitungsdatum.

**Bewertungskriterien** Es gibt ein numerisches Bewertungskriterium, z.B. anhand Schulnoten von 1 bis 6, mit dem ein Teilnehmer einen Kurs und/oder dessen Dozenten bewerten kann.

**Ansicht** Ein Browsing des Systems soll anhand Einrichtungen, Kursen der Einrichtung und Dozenten der Einrichtung hierarchisch möglich sein. Registrierte Nutzer sehen dabei zusätzlich zu aggregierten Bewertungswerten auch die einzelnen Bewertungen zusammen mit deren Texteintrag.

**Suche** Die Suche nach Einrichtungen, Kursen, Dozenten ist zusätzlich zum Browsing ein zentrales Element, um gezielt Informationen zu finden. Sie soll möglichst mächtig sein. Eine Schnellsuche ist beim Betrachten einer Ansicht verfügbar und bezieht sich nur auf die aktuelle (Listen-)Ansicht. Die Expertensuche hat mehr Suchparameter (z.B. Datumseingrenzung) als die Schnellsuche und geht über mehrere/alle Einrichtungen/Kurse/Dozenten.

**Benachrichtigung** Sämtliche Kommunikation der Webanwendung mit den Benutzern erfolgt per E-Mail. Bei neuen Einträgen in einer Einrichtung, einem Kurs oder zu einem Dozenten wird man als registrierter Nutzer nach Abonnement automatisch benachrichtigt. Diese Nachricht enthält einen Direktlink, um an die passende Stelle (ggf. nach einem Login) zu springen. Damit man hiermit nicht mit Mails überflutet wird, kann man in seinen Nutzereinstellungen eine maximale Nachrichtenanzahl pro Tag oder Woche angeben.

**Registrierung** Ein anonymes Benutzerkonto kann sich ein Benutzerkonto anlegen. Die Verifikation eines Benutzers kann entweder durch E-Mailverifikation und/oder durch die Bestätigung des Betreibers oder Moderator (einstellbar) erfolgen. Falls im System durch den Administrator erlaubt, kann bei der Registrierung bereits bei der Anmeldung am System angegeben werden, ob man selbst ein Dozent einer bestimmten (oder optional mehreren) Bildungseinrichtung ist oder nicht.

**Anmeldung** Ein Benutzer kann sich in ein Benutzerkonto einloggen und ist damit angemeldet. Hierfür ist eine Authentifizierung notwendig.

**Startseite** Registrierte Nutzer haben eine Art Startseite, auf der alle von ihnen abonnierten Inhalte oder Moderatortätigkeiten aufgelistet sind zwecks schnellem Zugriff. Zudem sollen Ansichten zu Einrichtungen, Kursen und Dozenten bookmarkbar sein.

**Kontoverwaltung** Ein angemeldeter Benutzer kann sein Benutzerkonto editieren oder löschen. Ein Profilbild/Thumbnail, welches bei eigenen Beiträgen angezeigt wird,

kann hochgeladen werden. Es kann bestimmt werden, welche Informationen des Profils registrierten Anwendern zugänglich gemacht werden.

**Abmeldung** Ein angemeldeter Benutzer kann sich abmelden.

**Benutzerverwaltung** Im System gibt es verschiedene Benutzerrollen: Betreiber, Moderatoren, registrierte Nutzer und anonyme Nutzer. Der Betreiber kann alle Benutzer und die damit verbundenen Rollen verwalten: Anlegen neuer Benutzer, Bearbeiten/Löschen existierender Benutzer, Zuordnen/Entziehen von Rollen. Beispielsweise werden Administratoren durch einen Betreiber ernannt oder ein Moderator kann Moderatoren für eine Bildungseinrichtung ernennen oder ihnen das Moderationsrecht entziehen. Eine Funktion zum Suchen nach bestimmten Benutzern erleichtert einem Betreiber die Verwaltung.

**Statistiken** Dem Betreiber soll die Möglichkeit geboten werden, sich Statistiken über Aktivität der (registrierten) Nutzer anzeigen zu lassen. So kann er sich z.B. über potentielle neue Moderatoren informieren.

**Customizing** Administratoren/Betreiber sollen das System an bestimmte Vorgaben anpassen können, z.B. Look & Feel, Registrierungsmodalitäten von Benutzern (E-Mails müssen verifiziert werden), Kontaktmöglichkeiten, Impressum, usw.

**Präsentation der Organisation** Die Anwendung sollte die Möglichkeit bieten, die jeweilige betreibende Organisation zu präsentieren (Beschreibung, Bild, Kontaktmöglichkeiten, Spezialgebiete, etc.).

**Online-Hilfe** Ein Benutzer muss zu jedem Zeitpunkt schnellen Zugriff auf die für die aktuelle Seite relevante Online-Hilfe haben.

## 5 Nicht-Funktionale Anforderungen

Folgend die Produktleistungen und Qualitätsanforderungen.

### 5.1 Usability

1. Einfache und intuitive Bedienbarkeit des Systems.
2. Die Benutzeroberfläche soll sich an allgemein geläufigen Bedienkonzepten und den damit verbunden Funktionen orientieren; alle Tabellen sollen z.B. nach den dargestellten Spalten sortierbar sein; Tabellen, die eine gewisse Größe überschreiten, sollen, um kurze Seitenladezeiten zu ermöglichen, auf mehrere Seiten aufgeteilt werden (Pagination).
3. Häufig wiederkehrende Aufgaben sind durch das System möglichst benutzerfreundlich zu unterstützen; häufig genutzte Funktionen sind möglichst einfach zugänglich zu machen.

4. Die Seiten der Applikation sind übersichtlich und einfach verständlich zu gestalten.
5. Daten sollen nicht nur leicht auffindbar und gut lesbar sein, sondern auch leicht einzugeben.
6. Bei Fehleingaben in ein HTML-Formular und der darauf folgenden Korrektur sollen die zuvor eingetragenen Felder nicht erneut einzugeben sein, sondern schon vorbesetzt sein. Außerdem sollte die Überprüfung der Eingaben nicht nach der ersten fehlerhaften Eingabe abbrechen, sondern alle Eingaben überprüfen und eine akkumulierte Fehlermeldung an den Benutzer zurückgeben.
7. Das System soll mit steigender Last skalieren.
8. Als Referenzplattform für die Implementierung dienen die Rechner im CIP-Pool unter Apache Tomcat.

## 5.2 Datensicherheit

1. Alle im System erfassten Daten sind persistent in einer Datenbank abzulegen; die Konsistenz der Daten ist sicherzustellen (Mehrbenutzerbetrieb!). Speziell, wenn Änderungen über mehrere Datenbanktabellen hinweg vorgenommen werden, sind Transaktionen zu nutzen.
2. Für die persistente Speicherung der Daten soll eine Datenbank (PostgreSQL) verwendet werden, die auf den Infosun-Rechnern zur Verfügung gestellt wird.
3. Beim Löschen von Daten sind die Abhängigkeiten zwischen Daten einzelner Tabellen zu berücksichtigen; hat das Löschen eines Datensatzes das Löschen anderer Datensätze zur Folge, muss der Benutzer vorher deutlich darauf hingewiesen werden!

## 5.3 Datenschutz

1. Es muss sichergestellt werden, dass durch das System zu keinem Zeitpunkt sensible Daten für unberechtigte Dritte zugänglich sind.
2. Es sollen möglichst wenige technische Informationen über das System nach außen gegeben werden.
3. Alle personenbezogenen Daten, wie z.B. Login-Daten, sind sensibel und dürfen daher nur per SSL-Verbindung übertragen werden.
4. Passwörter dürfen nicht im Klartext gespeichert werden.
5. Die Nutzerdaten sind so zu speichern, dass kein unautorisierter Zugriff durch Dritte oder durch andere Nutzer im System stattfinden kann.

6. Das System darf nur im Rahmen der oben genannten oder vorgesehenen Funktionalitäten verändert werden.
7. Änderungen durch Manipulationen mit bekannten Angriffsmethoden wie SQL-Injection oder Cross-Site-Scripting müssen ausgeschlossen werden. Es müssen außerdem Maßnahmen ergriffen werden, um die Sitzungen der einzelnen Nutzer zu schützen (Session-Hijacking).

#### **5.4 Internationalisierbarkeit**

1. Für die Texte auf der Website ist die Zeichenkodierung UTF-8 zu wählen.
2. Die Sprache des Systems kann Deutsch oder Englisch sein; eine mehrsprachige Implementierung ist optional.

#### **5.5 Weiterentwickelbarkeit**

Das zu erstellende System soll flexibel gegenüber zukünftigen Erweiterungen sein. Eine einfache und kostengünstige Weiterentwickelbarkeit des Systems ist sicherzustellen.

#### **5.6 Installation**

Es soll eine komfortable Installation für Systembetreiber bereitgestellt werden. Die Installation sollte einfach und schnell sein und automatisch das entsprechende Datenbank-Setup vornehmen.

### **6 Ergänzungen**

Die Benutzung des Systems sollte mit allen gängigen Web-Browsern möglich sein. Wir raten allerdings davon ab, für jeden Browsertyp unterschiedlichen HTML-Code zu generieren. Der HTML-Code soll logisches Markup darstellen und nicht dazu missbraucht werden, eine bestimmte graphische Darstellung zu erzwingen. Deswegen dürfen Features wie Frames nicht und explizites Javascript nur nach Absprache eingesetzt werden. Gleiches gilt für den Einsatz von Third-Party-Libraries. Der HTML-Code muss HTML-konform sein und z.B. durch <http://validator.w3.org/> validierbar sein. Die Verwendung von Cascading Stylesheets (CSS) wird dringend angeraten.

Die Sessionverwaltung darf die Verwendung von Cookies nicht erzwingen.

Das System muss ein Log über alle Fehler führen, um das Debugging und den Betrieb der Anwendung zu vereinfachen. Achten Sie darauf, dass die Fehlerbeschreibungen detailliert genug sind, um auf einen Fehler bzw. dessen Ursache schließen zu können.

### **7 Leistungsnachweis**

Mit der funktionalen Umsetzung o.g. Musskriterien kann die Note 1,0 erreicht werden. Wunschriterien sind (wie der Name sagt) wünschenswert, aber deren Umsetzung nicht

zwingend notwendig für das Erreichen der vollen Punktzahl.