



SEP – Sommersemester 2017

Forum für (Java-) Programmierer

Lastenheft

Christian Bachmaier, Thomas Bock

## 1 Motivation

Das Erstellen eines guten Computerprogramms z.B. in Java ist i.d.R. kein einfaches Unterfangen, sondern es ist meistens sehr viel Erfahrung und Profiwissen notwendig. Zudem treten immer wieder Situationen auf, zu denen eine Best-Practice oder einhellige Expertenmeinung existiert. Es macht also Sinn sich als Programmierer mit anderen (auch fremden) Programmierern auszutauschen und diese Kommunikation ggf. persistent und/oder öffentlich wieder auffindbar zu machen, da auch Dritte davon dann profitieren können. Somit ergibt sich ein klassisches Anwendungsszenario für ein web-basiertes Forum, das gut auf die Ansprüche und Wünsche von (Java-)Programmierern abgestimmt ist.

## 2 Aufgabenstellung

Ziel des Software-Engineering Praktikums im Sommersemester 2017 ist es, ein Webforum für den Erfahrungsaustausch unter Programmierern zu entwickeln, im Speziellen mindestens für Java-Programmierer. Um die Anwendung einfach betreiben und verwalten zu können, sollte sie auf einem eigenen Webspaces ausgeführt werden können. Die Anwendung sollte für den 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 Forumplattform 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 oder User-Account-Problemen) einschreiten können. Anonyme Benutzer sollten die Möglichkeit haben sich im System frei zu bewegen, um z.B. Postings/Titel nach Stichwörtern zu durchsuchen oder zu browsen. Geschlossene Foren, deren Ersteller/Moderatoren eine freie öffentliche Nutzung ausgeschlossen haben, sind natürlich davon auszunehmen. Nur angemeldete registrierte Nutzer haben die Möglichkeit, Foren anzulegen und Einträge zu verfassen. Bei geschlossenen Foren müssen sie dazu explizit von einem der Moderatoren ermächtigt worden sein.

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.

Beispiele für ähnliche und bereits existierende Systeme sind *stackoverflow*<sup>1</sup>, *CodeRanch*<sup>2</sup>, *Oracle Java Forum*<sup>3</sup>, *Java Forums*<sup>4</sup>, oder *dream.in.code*<sup>5</sup>. Diese Systeme gehen jedoch teilweise deutlich über die hier geforderte Funktionalitäten hinaus.

Die Anwendung muss mit Java 8 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 Firmen, Non-Profit-Organisationen oder auch private Personen mit eigenem Webspace. Ziel kann z.B. sein, eine vitale Community zu bestimmten Programmierthemen aufzubauen. Der freie Zugang über das Internet soll für ein produktiv eingesetztes System möglich sein, sodass interessierte Programmierer das System benutzen können. Auch ein Modus nur für eine geschlossene Benutzergruppe soll vorhanden sein, um das System z.B. innerhalb einer gewerblich orientierten Firma nur dem eigenen Personal oder Kooperationspartnern zur Verfügung zu stellen. Dazu kann der Administrator das Anlegen öffentlicher Foren zentral verhindern.

## 4 Produktfunktionen

### 4.1 Begriffsdefinitionen

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

---

<sup>1</sup><http://stackoverflow.com/>

<sup>2</sup><http://www.coderanch.com/forums/>

<sup>3</sup><https://community.oracle.com/community/java>

<sup>4</sup><http://www.java-forums.org/>

<sup>5</sup><http://www.dreamincode.net/forums/>

**Betreiber/Manager/Administrator:** Ist für die die Konfiguration des System verantwortlich. Dieser kann beispielsweise das Logo oder den Namen der Webapplikation festlegen. Zudem kann er weitere Einstellungen vornehmen, wie die zu verwendende Datenbank oder ob öffentliche Foren möglich sind. Eine wichtige Aufgabe des Betreibers/Admins ist die Userverwaltung, also das Anlegen, Editieren und Löschen von Nutzern. Ein Admin kann zudem wählen, ob nur bestimmte registrierte Nutzer Foren anlegen dürfen oder alle. Außerdem kann er als "Super"-Moderator über alle Foren agieren.

**(Forums-)Moderator:** Ist für alle Verwaltungsaufgaben eines Forums zuständig, welches er selbst erstellt hat oder zu dem er als Moderator ernannt wurde. Beispielsweise fällt darunter die Einstellung, ob ein Forum öffentlich oder geschlossen ist, oder die Zulassung/Einladung von Nutzern. Moderatoren können andere Moderatoren ernennen oder absetzen, natürlich nur für Foren, die sie selbst moderieren. Desweiteren können sie Beiträge anderer Forumsteilnehmer löschen oder editieren, z.B. wenn sich diese nicht an Benimmregeln o.Ä. halten sollten. Moderatoren können auch für mehr als ein Forum zuständig sein. Optional können Moderatoren bestimmte (registrierte) Benutzer aus dem Forum ausschließen.

**Registrierter Benutzer:** Hat die Möglichkeit, sich im System anzumelden und danach Foren zu erstellen (siehe Konfigurationsoption des Adminitrators) oder Einträge/Postings in für ihn sichtbaren Foren vorzunehmen. Hierfür muss sich dieser zuvor im System mit einer gültigen E-Mail-Adresse registrieren. Benutzer dürfen nur Daten (außer einem Nickname) anderer Benutzer erfahren, die explizit zur Anzeige von diesen freigegeben wurden. Registrierte Nutzer können Erfahrungspunkte ansammeln.

**Anonymer Benutzer:** Hat die Möglichkeit sich im System zu bewegen und öffentliche Foren/Postings 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.

**Threadansicht** Ein Forum soll durch geeignete hierarchische Gruppierung und Einrückung einen Diskussionsverlauf widerspiegeln. Das heißt konkret, es soll nachvollziehbar sein, welcher Beitrag eine Antwort auf welchen Beitrag ist, und klar ersichtlich sein, wie sich der zeitliche Ablauf gestaltet. Hier soll es grundsätzlich keine Begrenzung der Schachtelungstiefe geben. Ein Syntaxhighlighting von Code-Stücken ist wünschenswert, aber optional.

**Forumserstellung** Erstellen eines öffentlichen oder geschlossenen Forums als registrierter Nutzer und Verwalten wer dieses Forum moderiert. Ob dies möglich ist, richtet sich nach den globalen Vorgaben/Einstellungen des Administrators. Bei geschlossenen Foren kann man andere Nutzer per Nickname oder E-Mail einladen

und/oder bereits fest mit Zugriffsrechten eintragen. Falls per E-Mail eingeladene Nutzer noch kein Nutzerkonto haben, müssen diese sich zunächst am System registrieren und anmelden. Ein Forum besitzt einen Titel/Bezeichnung und einen Untertitel. Foren sollten mit dem Browser durch "feste" URLs leicht bookmarkbar sein.

**Beitrag** Erstellen eines Diskussionsbeitrags (auch als Antwort auf einen anderen). Dateienanhänge sind dabei möglich, deren Größe allerdings durch den Administrator zentral beschränkt sein kann. Diese Dateien können beim Lesen des Beitrags dann heruntergeladen werden. Zusätzlich zur normalen Fließtexteingabe sollen dedizierte Codeblöcke erstellt werden, die dann z.B. mit Typewriterfont o.Ä. gesetzt werden. Es muss mindestens die Option zum Fettschreiben einzelner Wörter oder Phrasen geben. Weitere Textformatierungsoptionen sind wünschenswert, aber optional. Jeder Beitrag hat einen Titel, ein Erstell-/Änderungsdatum und der Ersteller soll (mindestens) über seinen Nickname und Erfahrungswert gekennzeichnet werden. Eigene Beiträge von registrierten Nutzern können nachträglich editiert werden.

**Suche** Die Suche nach Inhalten in Foren ist zusätzlich zum Browsing der Foren und Beiträge ein zentrales Element, um gezielt Informationen zu finden. Sie soll möglichst mächtig sein. Eine Schnellsuche ist beim Betrachten eines Forums verfügbar und bezieht sich nur auf das aktuelle Forum. Die Expertensuche hat mehr Suchparameter (z.B. Datum und Autor des Beitrages) als die Schnellsuche und geht je nach Eingabe über mehrere/alle Foren.

**Benachrichtigung** Sämtliche Kommunikation der Webanwendung mit den Benutzern erfolgt per E-Mail. Bei neuen oder geänderten Einträgen in Foren, die man als registrierter Nutzer abonniert hat, wird man automatisch per E-Mail benachrichtigt. Diese Nachricht enthält einen Direktlink, um an die passende Stelle (ggf. nach einem Login) zu springen. Außerdem soll man sich über Beiträge (aus dem ganzen System) mit bestimmten Schlüsselwörtern/Phrasen, für die man sich vorher registriert, benachrichtigen lassen können. (Natürlich nur für Beiträge, die für einen Nutzer sichtbar sind.) Auch darauf soll man mit einem Direktlink hingewiesen werden. Damit man nicht mit E-Mails überflutet wird, kann man in seinen Nutzereinstellungen festlegen, ob man die E-Mail-Benachrichtigung für abonnierte Foren erhalten möchte oder nicht. Außerdem kann man eine maximale Nachrichtenanzahl pro Tag oder Woche für die Schlüsselwortregistrierung angeben.

**Erfahrungspunkte** Registrierte Nutzer haben einen Erfahrungspunktstand, der sich aus der Anzahl der Likes anderer Benutzer zu eigenen Beiträgen, der Anzahl der eigenen Beiträge und der Anzahl der moderierten Foren zusammensetzt. Diese Punkte sollen einen Indikator bilden, wie relevant ein Beitrag eines Nutzers ist.

**Registrierung** Ein anonymes Benutzerkonto kann sich ein Benutzerkonto anlegen. Die Verifikation eines Benutzers kann entweder durch E-Mail-Verifikation und/oder durch die Bestätigung des Betreibers/Administrators (einstellbar) erfolgen.

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

**Homepage** Registrierte Nutzer haben zwecks schnellem Zugriff eine Art Startseite, auf der alle von ihnen abonnierten oder moderierten Foren aufgelistet sind.

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

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

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

**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.).

**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 Manager oder gar Mitarbeiter informieren.

**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 Administrator kann Moderatoren für ein Forum ernennen oder ihnen das Moderationsrecht entziehen. Eine Funktion zum Suchen nach bestimmten Benutzern erleichtert einem Betreiber die Verwaltung.

## 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 oder Foren, 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 vom Auftraggeber 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 (bevorzugt) 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.