

Projektaufgabe Webengineering (Teil I)

Eure Aufgabe ist die Entwicklung einer Twitter-ähnlichen Webanwendung unter Verwendung der folgenden Technologien:

- Maven
- Spring-Boot (inklusive aller Bibliotheken aus dem Spring-Umfeld)
- Twitter Bootstrap (falls notwendig mit JQuery)
- Heroku

Solltet Ihr zusätzliche Bibliotheken benötigen, schreibt vor der Verwendung bitte eine E-Mail an lectures@micromata.de, damit wir klären können, ob der Einsatz der Bibliothek für die Lösung der Aufgabe zulässig ist.

Die Bearbeitung der Aufgabe erfolgt in **Einzelarbeit**. Wir behalten uns vor in Verdachtsfällen auf Plagiatismus von allen beteiligten Parteien ein ausführlichen Code-Walkthrough in einem persönlichen Gespräch zu verlangen.

Abgabe

Euer Projekt soll auf den git-Server des Software-Engineering-Fachgebiets committet werden, erreichbar über <http://gitlab.se.cs.uni-kassel.de>.

Das Projekt muss sich über

```
mvn clean package
```

vollständig und fehlerfrei automatisch mit einem JDK 8 bauen lassen. Ein Start muss über den Aufruf

```
java -jar target/<Projektname>-<Version>.jar
```

möglich sein. Auf zusätzliche Besonderheiten, wie z. B. separate Konfigurationsprofile, besondere Einstellungen, etc. muss in einer Datei README.md hingewiesen werden.

Bitte achtet darauf, regelmässig zu commiten. Wir schauen uns in (un)regelmäßigen Abständen den eingereichten Code an und geben Euch direktes Feedback.

Terminplan

08.05.2015 Vorstellung der Projektaufgabe

... Intensivbetreuung in der Übung nach der Vorlesung

03.07.2015 Abgabe bis 23:59 im git-Repository mit Tag release-1.0.0. Maßgeblich für die Abgabe ist der Zeitstempel des Tags, d.h. die Ausgabe von

```
git log -1 --format=%ai release-1.0.0
```

Anforderungen

Allgemeines

Designentscheidungen, sowohl bezüglich der Oberflächengestaltung und -benutzung als auch der Systemarchitektur bleiben euch überlassen. Achtet bitte darauf, dass wir Euch gegebenenfalls nach den Gründen, d.h. Vor- und Nachteilen, einer Entscheidung fragen.

Funktionalitäten, die darüber hinaus implementiert werden, bringen einen von uns im Einzelfall zu bestimmenden Bonus. Für das Erreichen der Bestnote ist die Implementierung der unten beschriebenen Anforderungen in jedem Fall ausreichend.

Registrierung

Ein neuer Benutzer soll sich unter Angabe eines Namens, einer E-Mailadresse und eines Passworts im System registrieren können.

Hinweis: Die Prüfung der E-Mailadresse, z. B. durch den Versand einer Bestätigungsmail ist nicht erforderlich.

Benutzerseite

Auf einer dedizierten Benutzerseite kann der Benutzer

- sein Passwort zurücksetzen
- seine Biografie — einen Kurztext über sich — bearbeiten

Die Biografie kann auch von anderen Leuten auf der Benutzerseite angesehen werden. Die Maximallänge der Biografie beträgt 1024 Zeichen.

Eingabe einer neuen Nachricht

Das System soll die Eingabe einer neuen Nachricht erlauben. Die Maximallänge einer Nachricht beträgt **141** Zeichen in UTF-8. Eine längere Nachricht darf nicht akzeptiert werden.

Auflistung aller Nachrichten

Das System soll die Auflistung aller erstellten Nachrichten erlauben. Es sollen folgende Filtermöglichkeiten existieren:

- Darstellung ausschließlich eigener Nachrichten
- Darstellung aller Nachrichten von verfolgten Leuten
- Anzeige aller existierenden Nachrichten

Die angezeigten Nachrichten sollen jeweils nach Erstellungsdatum sortiert sein. Auf jeder Seite mit einer Auflistung von Nachrichten soll eine Paginierung eingebaut sein, d.h. es wird stets nur ein Teil der existierenden Nachrichten angezeigt.

Stichwort-Suche

Das System soll die Suche nach einem Stichwort erlauben. Es werden alle Nachrichten angezeigt, die das angegebene Stichwort (unabhängig von Gross- und Kleinschreibung) enthalten. Ein Stichwort ist eine Zeichenfolge, die dem regulären Ausdruck `\b#\w\w+` entspricht.

Deployment

Die komplette Anwendung soll auf Heroku über den Aufruf

```
git push -f heroku master
```

deployed werden können. Zum Zeitpunkt der Abgabe soll eine bereits laufende Instanz mit der aktuellen Programmversion aktiv sein. Die URL der Instanz soll in der Datei README.md stehen.

Logging

Alle zur Nachvollziehbarkeit der verschiedenen Systemaktionen relevanten Informationen sollen über die in der Vorlesung vorgestellte Logging-Schnittstelle protokolliert werden.

Sicherheit

Ein Benutzer darf alle oben genannten Aktionen nur unter seinem Namen ausführen, d.h. es muss technisch verhindert werden, dass ein nicht-angemeldeter bzw. unter anderem Namen angemeldeter Benutzer z. B.

- Nachrichten oder Biografien als ein anderer Benutzer schreibt
- die Liste verfolgter Leute für jemand anderen außer sich selber ändert

Mehrbenutzerfähigkeit

Das System muss von mehreren Benutzern gleichzeitig fehlerfrei bedienbar sein. Wenngleich Effizienz und Skalierbarkeit keine explizite Rolle spielt, soll das System mit wenig parallel arbeitenden Benutzern (subjektiv) akzeptable Antwortzeiten bieten.

Bewertungskriterien

Wir bewerten den abgegebenen Quellcode nach den folgenden Kriterien (in absteigender Gewichtung)

Funktionalität	Erfüllt die Abgabe alle in den Anforderungen vorgegebenen Punkte?
Lesbarkeit	Ist der Code gut lesbar, d.h. <ul style="list-style-type: none"> • modularer Aufbau (z.B. Packages) • einheitliche Codeformatierung • kleine Klassen und kurze Methoden • Kommentierung von Schnittstellen und komplexen Methoden
Erweiterbarkeit	Ist der Code einfach zu ändern, d.h. um neue Funktionalitäten zu ergänzen bzw. existierende Features zu erweitern?
Commit-Kommentare	Beschreiben die Commit-Kommentare sinnvolle Änderungen an der Codebasis?

Bitte beachtet, dass die oben angegebenen Kriterien natürlich subjektiv sind. Im Zweifel interpretieren wir aber zu Euren Gunsten, insbesondere wenn ihr auf Nachfrage Entscheidungen durch eine verständliche Abwägung der Vor- und Nachteile Eurer Lösung begründen könnt.

Ergänzungen und Erratum

08.05.2015 An dieser Stelle werden in Zukunft Ergänzungen und Korrekturen zu den oben gemachten Angaben vorgenommen.

20.05.2015 Offizielle Bootstrap-Erweiterungen sind generell erlaubt.

29.05.2015 Bitte benutze für die Registrierung bei gitlab (siehe oben) das Namensschema **web-<Nachname>**.

16.06.2015 Mit Namen sind in der obigen Aufgabenstellung eindeutige Benutzernamen, d.h. 'mlesniak' (und nicht Klartextnamen, d.h. 'Michael Lesniak', gemeint)

16.06.2015 Der Einsatz von JavaScript ist zulässig.

16.06.2015 Zur Anzeige von User-Icons darf Gravatar benutzt werden.

28.06.2015 es sollen nur exakte Matches gefunden werden, d.h. bei

Tweet1: #hollywood

Tweet2: #hollywoodIsGreat

und der Suche nach #hollywood soll nur Tweet 1 gefunden werden.

28.06.2015 Es muss die Möglichkeit geben, Benutzer zu folgen bzw. das Folgen wieder zu entfernen. Wie dies aus Interface-Sicht ermöglicht wird bleibt euch überlassen.