

Übung 3: Anforderungsanalyse



TECHNISCHE
UNIVERSITÄT
DARMSTADT

Software Engineering
WS 2018/19 - Dr. Michael Eichberg

Abgabe

Die Übung wird als *sbt*-Projekt (<http://www.scala-sbt.org>) bereitgestellt. *sbt* kann verwendet werden, um Java (und Scala) Projekte einfach auszuführen und zu testen. Installieren Sie *sbt* auf Ihrem Rechner und stellen Sie sicher, dass ein Java SDK (min. Java 8) installiert ist. Überprüfen Sie, dass der Java Compiler *javac* auf der Kommandozeile ausführbar ist, auch wenn Sie sich nicht im Verzeichnis mit der ausführbaren Datei befinden. Wenn nicht, passen Sie Ihren *PATH* entsprechend an, Hinweise wie dies bei Ihrem Betriebssystem möglich ist, finden Sie im Internet.

IntelliJ erlaubt das Importieren von *sbt*-Projekten direkt, dafür muss allerdings das Scala Plugin installiert sein. Um ein Eclipse-Projekt zu erzeugen, kann *sbt eclipse* im Projektverzeichnis (das Verzeichnis, das die Datei *build.sbt* enthält) ausgeführt werden, danach kann das Projekt mit Datei > Importieren > Vorhandene Projekte in Arbeitsbereich importiert werden. Wenn Sie eine *main*-Methode geschrieben haben, können Sie Ihr Programm mit *sbt run* ausführen, Tests können Sie unter *src/test/java/* anlegen und mit *sbt test* ausführen.

Das Ausführen des Kommandos *sbt* im Projektverzeichnis startet den interaktiven Modus von *sbt*. Im interaktiven Modus können Kommandos ausgeführt werden, ohne jedes Mal *sbt* eingeben zu müssen.

Um Ihre Lösung abzugeben, melden Sie sich zunächst unter

<https://submission.st.informatik.tu-darmstadt.de/course/se18>

an und erzeugen Sie ein *submission token*. Wenn Sie den interaktiven Modus von *sbt* verwenden, führen Sie dann folgendes Kommando aus:

```
submit <ihreTUID> <submissionToken>
```

wobei *<ihreTUID>* Ihre eindeutige Identifikationsnummer an der TU Darmstadt (**nicht Ihre Matrikelnummer!**) und *<submissionToken>* das zuvor generierte Token ist. Geben Sie die spitzen Klammern nicht mit an, der Befehl sollte etwa so aussehen: *submit ab01cdef 01234567*. Wenn Sie nicht den interaktiven Modus verwenden, muss das Kommando in Anführungszeichen gesetzt werden, also *sbt "submit <ihreTUID> <submissionToken>"*. Sie können (innerhalb der Abgabefrist) beliebig oft eine Lösung einreichen, allerdings wird nur die zuletzt eingereichte Lösung bewertet. Die letzte Lösung, die ein Gruppenmitglied eingereicht hat, wird zur Bewertung für die ganze Gruppe herangezogen. Koordinieren Sie sich daher in Ihrer Gruppe, wer Ihre gemeinsame Lösung einreicht.

Stellen Sie sicher, dass Sie das zur Verfügung gestellte Template nutzen und die Namen und Signaturen der vorgegeben Klassen und Methoden nicht verändern sowie dass von Ihnen hinzugefügte Klassen und Methoden die geforderten Namen und Signaturen verwenden. Ändern Sie außerdem nichts an der vorgegebenen Datei *build.sbt*. Andernfalls wird das System Ihre Lösung nicht bewerten können. Beachten Sie, dass der Zugriff auf die Abgabepattform nur im **internen Netz der Universität** möglich ist. Für einen Zugriff von außerhalb benötigen Sie daher eine VPN-Verbindung.

Einführung

In dieser Übung befassen Sie sich mit der Anforderungsanalyse. Bearbeiten Sie die folgenden vier Aufgaben und erstellen Sie eine einzelne PDF-Datei mit Ihren Antworten. Legen Sie diese Datei in den Ordner *solution* in dem Template, das Ihnen zur Verfügung gestellt wurde, und verwenden Sie zur Abgabe wie oben angegeben *sbt submit*.

Problem 1 Klassifizierung**2P**

Geben Sie für die folgenden Anforderungen an, ob es sich um funktionale oder nicht-funktionale Anforderungen handelt. Geben Sie außerdem an, ob es sich um Nutzer- oder Systemanforderungen handelt. Begründen Sie Ihre Klassifizierung kurz.

- a) Die Webseite muss in alle gängigen Browsern korrekt angezeigt werden.
- b) Nutzer können Routendateien hochladen um Routen darstellen zu können.
- c) Die Import-Funktion darf, unabhängig von der Größe der Eingabedatei, nicht mehr als 100 MB RAM allozieren.
- d) Die Zeit bis zur Darstellung der Webseite darf 150 ms nicht überschreiten.

Problem 2 Überprüfbarkeit**3P**

Bewerten Sie, ob die folgenden nicht-funktionalen Anforderungen geeignet sind, um ihre Erfüllung überprüfen zu können. Wenn eine Anforderung nicht erfüllbar ist, geben Sie eine verbesserte Formulierung an, die die Überprüfbarkeit sicherstellt.

- a) Es darf nicht möglich sein, Kommentare mit beleidigendem Inhalt zu veröffentlichen.
- b) Der Quellcode muss die Java Coding Conventions einhalten.
- c) Die Anwendung muss unter Arch Linux lauffähig sein.
- d) Die Benutzeroberfläche zur Routenplanung muss intuitiv benutzbar sein.

Problem 3 Formulierung nicht-funktionaler Anforderungen**6P**

Die Erfüllung der folgenden nicht-funktionalen Anforderungen ist in dieser Form nicht überprüfbar. Recherchieren Sie geeignete Kriterien, anhand derer die Anforderungen überprüft werden können. Geben Sie eine neue Formulierung der Anforderungen an, die anhand dieser Kriterien überprüfbar ist. Nennen Sie auch die Quellen, auf die Sie sich stützen!

- a) Benutzern einer VR-Brille dürfen durch die Anwendung keine Kopfschmerzen oder Schwindel entstehen.
- b) Die Benutzeroberfläche der Applikation muss auf den allermeisten aktuellen Smartphones benutzbar sein.
- c) Die von der App mit Emojis dargestellten Emotionen sollen unabhängig vom Kulturkreis verständlich sein.

Problem 4 Taxonomie**5P**

Betrachten Sie die in der Vorlesung vorgestellte Taxonomie von Typen nicht-funktionaler Anforderungen.

- a) Klassifizieren Sie die Anforderungen aus Problem 2 gemäß der Taxonomie.
- b) Geben Sie für jede Kategorie der dritten Ebene der Taxonomie (also bspw. *Efficiency Requirements*, *Delivery Requirements*) jeweils eine überprüfbare nicht-funktionale Anforderung an. Der Kontext sei der einer Applikation für (Straßen)Karten und Routenplanung. Für die bereits in Teilaufgabe a) verwendeten Kategorien müssen Sie keine weitere Anforderung angeben.