I7 Teamverwaltung - Testdokumentation

Inhaltsverzeichnis

1. Allgemeine Informationen

In diesem Dokument werden die genutzten Testfälle (Test Cases) zur Überprüfung der Funktionalität spezifischer Softwarekomponenten beschrieben. Ziel der Test Cases ist es dabei herauszufinden, ob die Funktionsweise der zu testenden Komponenten dem erwarteten Verhalten entspricht.

2. Test Cases - Algorithmus

Aufgrund der Tatsache, dass der Algorithmus für eine bestimmte Gruppe an Testwerten kein ultimatives Ergebnis liefert, welches bei wiederholter Ausführung mit den gleichen Testwerten immer identisch ist, sondern jeweils ein möglichst gutes Ergebnis erzielt, dass bei wiederholter Ausführung unterschiedlich ausfallen kann, lassen sich die Testergebnisse nicht auf eine konkrete Zuordnung überprüfen. Stattdessen werden mit den folgenden Test Cases die harten Restriktionen für die Zuordnung durch den Algorithmus geprüft.

- Test Case 01: Überprüfung, ob alle Studenten in der Lösung berücksichtigt wurden.
- Test Case 02: Überprüfung, ob alle Projekte in der Lösung berücksichtigt wurden.
- Test Case 03: Überprüfung, ob jeder Student exakt einem Projekt zugeordnet wurde.
- Test Case 04: Überprüfung, ob alle Studenten gleichmäßig auf die vorhandenen Projekte aufgeteilt wurden.

- Test Case 05: Überprüfung, ob die Wirtschaftsingenieurwesen-Studenten gleichmäßig auf die vorhandenen Projekte aufgeteilt wurden.
- Test Case 06: Überprüfung, ob jedem Projekt exakt ein Student als Teamleiter zugeordnet wurde.
- Test Case 07: Überprüfung, ob die Studenten innerhalb der Projekte gleichmäßig auf die vorhandenen Rollen veverteilt sind.

2.1. Ergebnisse der Algorithmus-Tests

Bei der Durchführung der Algorithmus-Tests wurden keine überraschenden Fehlfunktionen festgestellt. Der Algorithmus erstellte die Ergebnisse konstant entsprechend der festgelegten harten Restriktionen. Folgend wurden, durch die Umsetzung automatisierter Unit-Tests, die Tests nach jedem Github-Commit automatisch ausgeführt um so eine kontinuierliche Überprüfung des Algorithmus zu ermöglichen. Nachträglich musste hierbei noch die SetUp- und TearDown-Funktion mit einer SetUp- bzw. TearDown-Class ersetzt werden, um eine geringere Laufzeit der Testausführung zu erreichen.

3. Test Cases - Models

Die zur Datenverwaltung verwendeten Django Models müssen zur Sicherung der Funktionalität auf ihre konkreten Eigenschaften und die spezifischen Abhängigkeiten zwischen den unterschiedlichen Models getestet werden. In den zu diesem Zweck erstellten Tests werden Objekte der jeweiligen Models mit im Vorhinein festgelegten Testdaten erstellt. In den Tests werden die erstellten Objekte jedes Models individuell aufgerufen, und die erhaltenen Daten mit den verwendeten Test-Daten verglichen. Im Fall einer Abweichung gilt der Test als fehlgeschlagen.

Zur Testdurchführung wird zuerst ein Model aus der Datei models.py importiert und ein Objekt des Models mit festgelegten Testdaten erstellt:

```
from ..models import Student

@classmethod
def setUpTestData(cls):
    cls.st = Student.objects.create(
       title="h",
       first_name="Max",
       last_name="Mustermann",
       s_num="s12345",
       faculty="ia"
    )
```

Das erstellte Objekt kann wiederum als Bestandteil von Objekten anderer Models genutzt werden, welche eine Fremdschlüssel-Beziehung zum Model des erstellten Objektes haben.

Zur Testdurchführung wird das erstellte Objekt aufgerufen und die erhaltenen Daten mit den genutzten Testdaten verglichen:

```
def test_st(self):
    self.assertEqual(self.st.title,"h")
    self.assertEqual(self.st.first_name,"Max")
    self.assertEqual(self.st.last_name,"Mustermann")
    self.assertEqual(self.st.s_num,"s12345")
    self.assertEqual(self.st.faculty,"ia")
    self.assertEqual(self.st.__str__(),"Max Mustermann - s12345")
```

Es ergeben sich folgende Test-Cases:

3.1. Test-Cases Model Student

- Test Case 01: Überprüfung des Wertes von *title*
- Test Case 02: Überprüfung des Wertes von *first_name*
- Test Case 03: Überprüfung des Wertes von *last_name*
- Test Case 04: Überprüfung des Wertes von *s_num*
- Test Case 05: Überprüfung des Wertes von *faculty*
- Test Case 06: Überprüfung des Rückgabewertes der Stringfunktion

3.2. Test-Cases Model Project

- Test Case 07: Überprüfung des Wertes von *name*
- Test Case 08: Überprüfung des Wertes von *description*
- Test Case 09: Überprüfung des Wertes von *responsible*
- Test Case 10: Überprüfung des Wertes von *documentfile_url*
- Test Case 11: Überprüfung des Rückgabewertes der Stringfunktion

3.3. Test-Cases Model Poll

- Test Case 12: Überprüfung der Fremdschlüsselbeziehung zum Model *Student*
- Test Case 13: Überprüfung des Wertes von *is_wing*
- Test Case 14: Überprüfung des Rückgabewertes der Stringfunktion

3.4. Test-Cases Model Role

- Test Case 15: Überprüfung des Wertes von *role*
- Test Case 16: Überprüfung des Rückgabewertes der Stringfunktion

3.5. Test-Cases Model Assignment

- Test Case 17: Überprüfung der Fremdschlüsselbeziehung zum Model *Student*

- Test Case 18: Überprüfung der Fremdschlüsselbeziehung zum Model *Project*
- Test Case 19: Überprüfung der Fremdschlüsselbeziehung zum Model *Role*
- Test Case 20: Überprüfung des Rückgabewertes der Stringfunktion

3.6. Test-Cases Model Skill

- Test Case 21: Überprüfung des Wertes von *skill*
- Test Case 22: Überprüfung des Wertes von *abbreviation*
- Test Case 23: Überprüfung des Rückgabewertes der Stringfunktion

3.6.1. Test-Cases Model Projectanswer

- Test Case 24: Überprüfung der Fremdschlüsselbeziehung zum Model *Project*
- Test Case 25: Überprüfung der Fremdschlüsselbeziehung zum Model *Poll*
- Test Case 26: Überprüfung des Wertes von *score*
- Test Case 27: Überprüfung des Rückgabewertes der Stringfunktion

3.7. Test-Cases Model Skillanswer

- Test Case 28: Überprüfung der Fremdschlüsselbeziehung zum Model *Skill*
- Test Case 29: Überprüfung der Fremdschlüsselbeziehung zum Model *Poll*
- Test Case 30: Überprüfung des Wertes von *score*
- Test Case 31: Überprüfung des Rückgabewertes der Stringfunktion

3.8. Test-Cases Model Roleanswer

- Test Case 32: Überprüfung der Fremdschlüsselbeziehung zum Model *Role*
- Test Case 33: Überprüfung der Fremdschlüsselbeziehung zum Model *Poll*
- Test Case 34: Überprüfung des Wertes von *score*
- Test Case 35: Überprüfung des Rückgabewertes der Stringfunktion

3.9. Ergebnisse der Model-Tests

Bei der Durchführung der Model-Tests wurden ebenfalls keine überraschenden Fehlfunktionen festgestellt. Das Festlegen von Member-Werten sowie die Fremdschlüsselbeziehungen zwischen den Klassen funktionierten einwandfrei. Auch die Ausgabe der Werte der Models lief wie erwartet ab. Folgend wurden, durch die Umsetzung automatisierter Unit-Tests, die Tests nach jedem Github-Commit automatisch ausgeführt um so eine kontinuierliche Überprüfung der Models zu ermöglichen.