

IU Internationale Hochschule
Studiengang Informatik (M.Sc.)

PJ SkillTrack

Testdokument
im Fach

Projekt: Software Engineering (DLMCSPSE01_D)

Luca Witt (IU14101147)
am 4. Oktober 2025

Tutor:

Davin Kühlen

Inhaltsverzeichnis

1	Teststrategie	1
1.1	Unittests	1
1.2	Integrationstests	1
1.3	System- & Akzeptanztests	2
2	Testprotokoll	3
2.1	Funktionale Anforderungen	3
2.2	Nichtfunktionale Anforderungen	5
2.2.1	Responsive Design (NF01)	5
2.2.2	Intuitive Bedienung (NF02)	6
2.2.3	Performance (NF05)	7
2.2.4	Sonstige (NF03, NF04)	7
A	Tabellenverzeichnis	8
B	Literatur	9

1 Teststrategie

Das Ziel der Qualitätssicherung ist es, sicherzustellen, dass alle definierten funktionalen und nicht-funktionalen Anforderungen erfüllt werden. Die in dieser Phase durchgeführten Tests können manueller und automatisierter Natur sein und haben je nach Teststufe, auf der sie durchgeführt werden, verschiedene Fokuspunkte.

1.1 Unittests

Unittests, oder auch Komponententests, finden, basierend auf der Codeabstraktion, auf der niedrigsten Ebene statt und testen jede einzelne Komponente, welche sinnvoll testbar ist, wie z.B. im vorliegenden Kontext eine Methode. Wichtig hierbei ist, dass der gesamte umgebende Kontext, also z.B. Variablen, die für den Tests benötigt werden, fest gesetzt sind und nicht von dritten Komponenten in Form eines Seiteneffektes manipuliert werden können, um eine Wiederholbarkeit sicherzustellen. Dafür werden sogenannte Mock-Objekte verwendet, welche leere Hüllen um verschiedene konkrete Objekte darstellen und von den Entwickler:innen in Tests in ihren Schnittstellen definiert werden können, um das Verhalten dieser kontrollieren zu können. Bei der Nutzung von Spring bzw. Spring Boot bietet sich das Java-Test-Framework JUnit (The JUnit Team, n. d.) an.

1.2 Integrationstests

Integrationstests finden auf einer höheren Abstraktionsebene als die Unittests statt und testen das Zusammenspiel mehrerer Komponenten, um z.B. komplette logisch sinnvolle Tätigkeiten wie das Update einer Entität über einen Controller-Aufruf bis hin zum eigentlichen Update durch das JPA-Repository zu testen.

In PJ SkillTrack findet die gesamte Kommunikation mit dem Server über eine HTTP-Schnittstelle statt. Die Bibliothek *rest-assured* (Haleby & REST assured Contributors, n. d.) bietet hilfreiche Methoden, um Aufrufe gemäß der REST-Architektur gegen ein laufendes System zu stellen und das zurückgelieferte Ergebnis mithilfe von Assertions auf Korrektheit zu prüfen. Um den Webservice selbst in einen testfähigen Zustand zu bringen, kann das Spring-Modul *Spring Boot Starter Test* (VMware, n. d.) verwendet werden, welches selbst noch weitere Bibliotheken wie JUnit mitliefert. Unter Verwendung verschiedener Interfaces oder Annotationen wird ein Server mit funktionsfähigem Application Context erzeugt, welcher je nach Konfiguration des zugrundeliegenden Profiles (theoretisch lassen sich die erzeugten Beans über festgelegte Profiles wie *dev* und *test* steuern) identisch in der Funktionsweise des tatsächlich betriebenen Servers ist. Da für den Betrieb des Webservers auch eine entsprechende Datenbank existieren muss, kann diese mithilfe von *Testcontainers* (Testcontainers, n. d.) und einer Container-Engine wie Docker erzeugt und nach jedem Testdurchlauf in einen konsistenten Zustand versetzt werden. Die Integrationstests laufen somit automatisiert gegen ein real ausgeführtes System.

1.3 System- & Akzeptanztests

Die System- und Akzeptanztests dienen zum einen der Sicherstellung aller Anforderungen in einem realen Szenario und können zum anderen zur Bedingung für die Abnahme einer bestellten Software eingesetzt werden. Die Durchführung findet nicht automatisiert statt, sondern wird von verschiedenen Personen in der Rolle der tatsächlichen Anwender:innen in Form eines Black-Box-Tests zumeist auf einem dedizierten Testsystem durchgeführt, welches dem Produktivsystem in der Konfiguration möglichst nahe kommt. Es können auch mehrere Testumgebungen existieren, im vorliegenden Fall sollten nach NF01 sowohl mobile (z.B. Smartphone) als auch Desktop-Geräte als Testumgebung eingesetzt werden. Im selben Zuge lässt sich ebenfalls überprüfen, inwiefern die Umsetzung der Benutzeroberfläche dem geplanten Design entspricht.

2 Testprotokoll

2.1 Funktionale Anforderungen

Die Durchführung der Tests erfolgte anhand der beigefügten Demo-Daten, welche sich bei Bedarf bei Serverstart laden lassen.

ID	Kurzbeschreibung	Vorbedingungen	Schritte / Eingaben	Erwartetes Ergebnis	Tatsächliches Ergebnis
TC-F01-01	Skills je Fachrichtung anzeigen	Benutzer eingeloggt; es existieren Fachrichtungen und Skills	Webseite (neu) laden	Liste der Skills der gewählten Fachrichtung wird geladen & angezeigt	Erfüllt
TC-F01-02	Skills je Fachrichtung anzeigen nach Anmeldung	Benutzer nicht eingeloggt; es existieren Fachrichtungen und Skills	Anmeldung vornehmen	Liste der Skills der gewählten Fachrichtung wird geladen & angezeigt	Erfüllt
TC-F02-01	Initialer Status auf „Offen“	Benutzer authentifiziert; ein Skill S wurde noch nie im Status aktualisiert	Webseite (neu) laden, entsprechende Fachrichtung auswählen	Der Status des Skills S entspricht „Offen“	Erfüllt
TC-F02-02	Status & Zeitstempel ändern	Benutzer authentifiziert; ein Skill S hat Status „Offen“	Status durch Klick auf das Status-Symbol auf „Gesehen“ ändern, Seite neu laden	Der Status von Skill S entspricht „Gesehen“; Der Zeitstempel in der Detailansicht entspricht der Änderungszeit	Erfüllt
TC-F02-03	Ungültiger Status abfangen	Skill mit ID X existiert & Nutzer ist authentifiziert	Anfrage an <code>/api/skill/X</code> mit fehlerhaftem Status „FALSCH“ senden; dann über GUI den Skill laden	Der Skill wurde nicht aktualisiert; der Server liefert bei der Anfrage einen 400-Bad-Request-Fehlercode	Erfüllt

TC-F03-01	Notiz hinterlegen	Skill S sichtbar und in Detailansicht; Notiz-Feld ist leer	Text abändern und per Button speichern; Seite neu laden und Skill-Detailansicht öffnen	Das Notizfeld enthält den aktualisierten Text	Erfüllt
TC-F03-02	Lange Notiz einfügen	Skill S sichtbar und in Detailansicht; Notiz-Feld enthält beliebigen Text	Sehr langen Text (mind. 3000 Zeichen) einfügen	Der gesamte Text wird angezeigt und nicht abgeschnitten	Erfüllt
TC-F03-03	Sonderzeichen einfügen	Skill S sichtbar und in Detailansicht; Notiz-Feld enthält beliebigen Text	Das Unicode-Symbol U+1F601 eingeben; speichern & neu laden	Das Symbol wird korrekt angezeigt	Erfüllt
TC-F04-01	Skills nach Fachrichtung filtern	Skill S vorhanden; dieser ist Teil von Fachrichtung F1, aber nicht von F2; F1 ist ausgewählt und S ist sichtbar	Im Dropdown in der Oberfläche F2 auswählen	Skill S ist in der Liste der Skills nicht vorhanden	Erfüllt
TC-F05-01	Fortschritt pro Fach anzeigen	Fachrichtungen F1 & F2, beide haben jeweils mind. einen Skill pro Status mit entsprechendem Status	Fortschrittsübersicht öffnen	In Listenansicht wird für F1 & F2 (alphabetisch) jeweils ein Fortschrittsbalken mit allen Status angezeigt; jeder Status hat mind. einen Eintrag	Erfüllt

TC-F05-02	Fortschritt für unbearbeiteten Skill ist „Offen“	Fachrichtung F hat genau einen Status S, der noch nie bearbeitet wurde	Fortschrittsübersicht öffnen	In Listenansicht wird für F genau ein Eintrag unter der Sektion „Offen“ angezeigt	Erfüllt
TC-F05-03	Fortschritt pro Fach wird ebenfalls aktualisiert	Skill S mit Status „Offen“ für Fachrichtung F vorhanden, einziger Skill mit diesem Status für F	Skill-Status ändern auf „Gesehen“; wechseln zur Fortschrittsübersicht	In Listenansicht für Status F ist die Sektion für „Offen“ aus dem Balken verschwunden	Erfüllt

Tabelle 2.1: Funktions-Testfälle (F01–F05).

2.2 Nichtfunktionale Anforderungen

2.2.1 Responsive Design (NF01)

Die Durchführung der Responsiveness Tests erfolgte unter Nutzung der Developer Tools(Google LLC, n. d.) von Google Chrome, welche die Darstellung einer Webseite mithilfe verschiedener Geräte/Viewports simulieren lassen.

ID	Geräte/Viewport	Schritte	Erwartung	Tatsächliches Ergebnis
TC-NF01-01	kl. Smartphone; 320x568	Webseite im Browser (Safari oder Chrome) öffnen, alle Hauptseiten prüfen	Keine Horizontal-Scrolls, Navigation über Burger-Menü, Inhalte lesbar	Erfüllt
TC-NF01-02	Tablet; 820x1180	Webseite im Browser (Safari oder Chrome) öffnen, alle Hauptseiten prüfen	Keine Horizontal-Scrolls, Navigation über Header-Leiste, Inhalte lesbar	Erfüllt
TC-NF01-03	Desktop; 1920x1080	Webseite im Browser (Safari oder Chrome) öffnen, alle Hauptseiten prüfen	Keine Horizontal-Scrolls, Navigation über Header-Leiste, kein Burger-Menü vorhanden	Erfüllt

Tabelle 2.2: Testfälle für die Untersuchung der Responsiveness.

2.2.2 Intuitive Bedienung (NF02)

Rahmenbedingung für Usability-Test:

- mind. 30 Teilnehmende
- Grundmenge sind PJ-Studierende, Teilnehmende ermittelt über einfache Zufallsstichprobe
- Durchführung der Aufgaben jeweils in Google Chrome (Desktop) und Safari (Mobile)

Aufgabenkatalog:

1. Skills einer Fachrichtung finden (inkl. Anmeldung) (F01)
2. Status ändern (F02)
3. Notiz „Test“ speichern (F03)
4. Nach Fachrichtung filtern (F04)
5. Fortschritt prüfen (F05)

Metriken & Akzeptanzkriterien:

- Erfolgsquote „ohne Hilfe“ pro Aufgabe: über 90% über alle Teilnehmenden.
- Durchschnittliche Bearbeitungszeit pro Aufgabe: unter 90 s.

Tatsächliche Durchführung:

Im Rahmen der vorliegenden Projektarbeit wurde kein Nutzertest mit mindestens 30 Teilnehmenden durchgeführt. Nachfolgend sind die Ergebnisse für eine weitere Person, welche die angegebenen Grundmenge-Kriterien erfüllt und nicht am Entwicklungsprozess beteiligt war, aufgeführt: 24,61s

Aufgabe	Gerät	Bearbeitungszeit	Ohne Hilfe gelöst?
1	Mobil	11,28s	Ja
1	Desktop	9,40s	Ja
2	Mobil	8,37s	Ja
2	Desktop	5,83s	Ja
3	Mobil	7,35	Ja
3	Desktop	5,11s	Ja
4	Mobil	11,10s	Ja
4	Desktop	5,27s	Ja
5	Mobil	7,86s	Ja
5	Desktop	6,11s	Ja

Tabelle 2.3: Ergebnisse für die Untersuchung der Usability.

2.2.3 Performance (NF05)

ID	Beschreibung	Schritte	Erwartung	Tatsächliches Ergebnis
TC-NF05-01	Lighthouse-Report ¹	Anmeldung im Browser; Start des Lighthouse-Reports	Performance liegt bei mind. 75/100 Punkten	72
TC-NF05-02	Geringe Ladezeit	Anmeldung im Browser; Reload durchführen	Gemessene Ladezeit zwischen Reload und Skill-Liste liegt bei unter einer Sekunde ²	400ms

Tabelle 2.4: Testfälle für die Untersuchung der Performance.

Obwohl TC-NF05-01 fehlgeschlagen ist, wird die Software aufgrund der tatsächlichen geringen Ladezeit abgenommen. Nichts desto Trotz muss die Performance anhand der Analyseergebnisse des Lighthouse-Reports in naher Zukunft untersucht werden, um potenzielle Performance-Probleme in der Zukunft zu mitigieren.

2.2.4 Sonstige (NF03, NF04)

Die Anforderungen nach Datenschutz (NF03) durch das Speichern von nur unbedingt benötigten persönlichen Informationen muss durch die Entwicklung erfolgen. Benötigt werden eine Email sowie ein Passwort für die Anmeldung. Alle weiteren Informationen wie in den Skill-Notizen werden von den Anwender:innen freiwillig angegeben.

Die Datensicherheit (NF04) wird ebenfalls technisch sichergestellt durch die Anwendung eines Hashing-Algorithmus (bcrypt) im Servercode.

¹<https://developer.chrome.com/docs/lighthouse?hl=de>

²Gemittelt über 10 Durchführungen, da Schwankungen in der Netzwerkverbindung möglich sind

A Tabellenverzeichnis

2.1	Funktions-Testfälle (F01–F05).	5
2.2	Testfälle für die Untersuchung der Responsiveness.	6
2.3	Ergebnisse für die Untersuchung der Usability.	6
2.4	Testfälle für die Untersuchung der Performance.	7

B Literatur

Google LLC. (n. d.). Chrome-Entwicklertools | Chrome DevTools. Verfügbar 4. Oktober 2025 unter <https://developer.chrome.com/docs/devtools?hl=de>.

Haleby, J., & REST assured Contributors. (n. d.). REST Assured. Verfügbar 4. Oktober 2025 unter <https://rest-assured.io/>.

Testcontainers. (n. d.). Testcontainers. Verfügbar 4. Oktober 2025 unter <https://testcontainers.com/>.

The JUnit Team. (n. d.). JUnit. Verfügbar 3. Oktober 2025 unter <https://junit.org/>.

VMware. (n. d.). Spring Boot Starter Test. Verfügbar 3. Oktober 2025 unter <https://docs.spring.io/spring-boot/docs/1.5.7.RELEASE/reference/html/boot-features-testing.html>.