Test-Dokumentation: StudIQ

Inhaltsverzeichnis

1. Allgemeine Informationen

Dieses Dokument beschreibt die Test Cases für das Projekt StudIQ. Es enthält detaillierte Informationen zu den einzelnen Testfällen, deren Durchführung und die erzielten Ergebnisse. Ziel ist es, durch systematische Tests die funktionale Korrektheit, Stabilität und Wartbarkeit der Software sicherzustellen sowie kritische Fehler frühzeitig zu identifizieren.

Die Teststrategie orientiert sich an der Testpyramide, wobei der Fokus auf Unit-Tests liegt, gefolgt von Integrationstests und abschließend Systemtests. Dies gewährleistet eine umfassende Abdeckung der Softwarekomponenten und deren Zusammenspiel.

2. Testkonzept

2.1. Testarten und Testmethoden

Bei der Entwicklung der Testfälle für StudIQ wurde sich stark an der Testpyramide orientiert. Der Schwerpunkt liegt demnach auf Unit-Tests zum Testen der einzelnen Komponenten. Ergänzt werden diese durch Integrationstests zum Testen der zentralen Schnittstellen.

Testart	**Testobjekt**	**Ziel**	**Beispiel**
Unit-Test	einzelne Softwarekomponenten	Sicherstellung der Funktionalität auf Komponentenebene	Quiz muss mindestens drei Fragen enthalten

Testart	Testobjekt	Ziel	Beispiel
Integrationstest	Testen der APIs und Datenbankzugriffe	Sicherstellung der Interaktion zwischen Komponenten	Ein Quiz wird korrekt aus der Datenbank geladen
Systemtest	Testen des gesamten Systems	Sicherstellung der Erfüllung der Anforderungen	Ein Nutzer kann ein Quiz erfolgreich absolvieren und erhält eine Bewertung

Unit-Tests wurden als Whiteboxtests implementiert und prüfen die Funktionen isoliert. Integrationstest prüfen die REST-API-Endpunkte und Authentifizierung. Sie werden als BlackboxTests durchgeführt.

3. Testumgebung und Testwerkzeuge

Die Tests wurden in einer lokalen Entwicklungsumgebung durchgeführt.

Verwendete Werkzeuge und Frameworks:

- Programmiersprache: Python
- Testframework: Django Test Framework (TestCase, APIClient)
- Ausführung der Tests: pytest

4. Testen im Entwicklungsprozess

Die Testphase hat erst gegen Ende der Entwicklungsphase begonnen, als die meisten Funktionen implementiert waren. Dies ermöglichte es, die Tests auf eine stabile Codebasis anzuwenden und sicherzustellen, dass die Kernfunktionen der Software abgedeckt sind (kein Test Driven Design).

Die Testfälle wurden vom Tester der Gruppe geschrieben und durchgeführt. Sie sind abgelegt unter `src/server/tests/`. Bei jeder Durchführung laufen die Tests automatisch durch. Es kann angegeben werden welcher Unterordner getestet werden soll, um z.B. nur Modultests durchzufuehren: python manage.py test tests.integration Zusätzlich werden automatische Testlogs angelegt, welche im Falle von Fehlern zur Analyse herangezogen werden können. Sie sind abgelegt unter `src/server/test_logs/`.

5. Bekannte Abweichungen / Bugs

Während der Tests wurden mehrere Bugs identifiziert, welche in Github dokumentiert sind unter dem Label 'Bug'. https://github.com/users/Gorg-tech/projects/7/views/4 Einer der wichtigsten gefunden Bugs ist ein Fehler in der Quiz-update-Funktionalität, welcher dazu führt, dass Änderungen an einem Quiz nicht korrekt gespeichert werden. Dieser Bug konnte behoben werden und ist in der aktuellen Version nicht mehr vorhanden. Die in Github dokumentierten Bugs können sich die Developer assignen und beheben. Manche Bugs sind für die finale Version nicht kritisch und können aktuell vernachlässigt werden.

6. Bewertung des Softwarequalitätsstands

Der aktuelle Qualitätsstand der Software ist insgesamt zufriedenstellend. Die Kernfunktionalitäten des Systems werden durch Unit- und Integrationstests abgedeckt und zeigen ein stabiles Verhalten. Kritische Bugs wurden identifiziert, dokumentiert und größtenteils behoben. Restrisiken bestehen vor allem in Randfällen und nicht-kritischen Funktionen, die in zukünftigen Iterationen adressiert werden können. Aktuelle Bugs wurden dem Team mitgeteilt und in der Priorisierung vorerst niedrig eingestuft.

7. Test Case 1

Dieser Test Case überprüft die Grundfunktionalität der Benutzerregistrierung. Es wird getestet, ob ein neuer Benutzer erfolgreich registriert werden kann und ob die erforderlichen Felder korrekt validiert werden.

7.1. Testschritte

1. Navigiere zur Registrierungsseite.
2. Fülle das Registrierungsformular mit gültigen Daten aus (eindeutigen Benutzername, E-Mail, Passwort und Studiengang wählen).
3. Klicke auf die Schaltfläche "Registrieren".

7.2. Erwartetes Ergebnis

Der Benutzer wird erfolgreich registriert und zur Login-Seite weitergeleitet.

Ergebnis: Bestanden

8. Test Case 2

Dieser Test Case überprüft die Funktionalität der Quiz-Durchführung. Es wird getestet, ob ein Benutzer ein Quiz erfolgreich durchführen kann und ob die Antworten korrekt ausgewertet werden.

8.1. Testschritte

1. Melde dich mit einem registrierten Benutzerkonto an.
2. Navigiere zum Profil (untere Leiste ganz rechts) und überprüfe welcher IQ-Score angezeigt wird.
3. Navigiere zur Suche.
4. Filter nach Quiz und wähle ein Quiz aus der Liste aus.
5. Starte das Quiz und beantworte alle Fragen.

8.2. Erwartetes Ergebnis

Das Quiz wird erfolgreich abgeschlossen und die Ergebnisse werden korrekt angezeigt. Der IQ-Score erhoht sich entsprechend der erreichten Punktzahl.