Technische Universität Braunschweig

ISF

Mediaverse

Software-Entwicklungspraktikum (SEP)
Sommersemester 2023

Testprotokolle

Auftraggeber
Technische Universität Braunschweig
Institut für Robotik und Prozessinformatik - IRP
Mühlenpfordtstraße 23
38106 Braunschweig

Betreuer: Dr. Bertold Bongardt, Heiko Donat, Sven Tittel, Christopher Lösch

Auftragnehmer:

Name	E-Mail-Adresse
Michèle Eger	m.eger@tu-braunschweig.de
Fahd Ferjani	f.ferjani@tu-braunschweig.de
Yassine Kechiche	y.kechiche@tu-braunschweig.de
Yiğit Kemal Çağlar	y.caglar@tu-braunschweig.de
Oussema Ben Smida	o.ben-smida@tu-braunschweig.de
Subing Shen	subing.shen@tu-braunschweig.de
Quynh Tran	quynh.tran@tu-braunschweig.de
Theodore Zebua	t.zebua@tu-braunschweig.de

Braunschweig, 12. Juli 2023

Bearbeiterübersicht

Kapitel	**Autoren**	**Kommentare**
1	Quynh	...
2	Yassine	...
3	Fahd	...
4	Fahd	...
5	Quynh	...
6	Quynh	...
7	Subing	...

Inhaltsverzeichnis

1 Einleitung

In diesem Dokument werden die durchgeführten Testprotokolle von der webbasierten Applikation Mediaverse festgehalten und detailliert beschrieben.
Ziel ist es die Funktionen und Qualität der Software zu testen sowie sicherzustellen. Es sollen dadurch Fehler erkannt und verbessert werden. Hiermit soll sichergestellt werden, dass die Nutzung der Software möglich ist, jedoch aber keine komplette Fehlerfreiheit garantieren.

2 Testdurchführung (2023-07-10-001)

Art des Tests: Abnahmetest
Ausgeführte Testfälle: **⟨T110⟩, <T120>, <T210>**
Beteiligte Tester: Yassine Kechiche
Abgedeckte Funktionen: **F10, F20, F30**

2.1 Testumgebung

Die Testfälle wurden unter Windows 11 auf einem lokalen Webserver ausgeführt. Für den Backend-Code wurde das Django-Framework verwendet und der Webserver wurde mit dem Befehl "python manage.py runserver"gestartet. Der Frontend-Code wurde mit Svelte entwickelt und der Entwicklungsserver wurde mit dem Befehl "npm run dev"gestartet. Es wurde der Chrome-Browser in der Version ”114.0.5735.199 (Official Build) (64-bit)” für die Durchführung der Tests verwendet.

2.2 Testprotokoll

Testfall	***T110: Konto erstellen***
Tester	*Yassine Kechiche*
Eingaben	*Die Webseite aufrufen und auf den Registrierungslink klicken. Benutzername, E-Mail-Adresse und Passwort eingeben und auf ”Konto erstellen” klicken.*
Soll - Reaktion	*Erfolgreiche Registrierung mit Anzeige einer Bestätigungsnachricht und Weiterleitung zur Login-Seite.*
Ist – Reaktion	*Die Ist-Reaktion stimmt mit der Soll-Reaktion überein.*
Ergebnis	*Der Test war erfolgreich.*
Unvorhergesehene Ereignisse	*keine*
Nacharbeiten	*keine*

Testfall	***T120: Einloggen***
Tester	*Yassine Kechiche*
Eingaben	*Die Webseite von Mediaverse aufrufen. Die Login-Seite wird angezeigt. Anmeldedaten (Benutzername und Passwort) in das Login-Feld eingeben. Auf 'Login' klicken.*
Soll - Reaktion	*Erfolgreiche Anmeldung mit Weiterleitung zur Homepage von Mediaverse.*
Ist – Reaktion	*Die Ist-Reaktion stimmt mit der Soll-Reaktion überein.*
Ergebnis	*Der Test war erfolgreich.*
Unvorhergesehene Ereignisse	*keine*
Nacharbeiten	*keine*

Testfall	***T210: Suchfunktion nach beliebigen Ressourcen (mit bestimmten Suchfilter)***
Tester	*Yassine Kechiche*
Eingaben	*Zur Homepage von Mediaverse gehen. In das Suchfeld klicken und einen Suchbegriff eingeben. Den Suchbegriff bestätigen.*
Soll - Reaktion	*Eine Liste mit der gesuchten Ressource wird auf der Seite angezeigt.*
Ist – Reaktion	*Die Ist-Reaktion stimmt mit der Soll-Reaktion überein.*
Ergebnis	*Der Test war erfolgreich.*
Unvorhergesehene Ereignisse	*Keine*
Nacharbeiten	*Keine*

2.3 Zusammenfassung

- Das Aussehen der Homepage befindet sich noch in Bearbeitung und wird fortlaufend verbessert. Es wurden jedoch keine funktionalen Abweichungen festgestellt.
- Die durchgeführten Tests deckten die grundlagenden Funktionen ab, nämlich das Erstellen eines Kontos, das Anmelden und die Suche nach Ressourcen
- Das Aussehen der Homepage befindet sich noch in Bearbeitung und wird fortlaufend verbessert. Es wurden jedoch keine funktionalen Abweichungen festgestellt

- Basierend auf den erfolgreichen Testergebnissen kann die Softwarequalität als zufriedenstellend bewertet werden. Die durchgeführten Tests haben gezeigt, dass die grundlegenden Funktionen korrekt funktionieren.

3 Testdurchführung (2023-07-11)

Art des Tests: Unit-tests
Ausgeführte Testfälle: ⟨**T210**⟩, <**T220**>, <**T230**>, <**T240**>
Beteiligte Tester: Fahd Ferjani
Abgedeckte Funktionen: **F60:Hinzufügen/Bearbeiten/Löschen von Ressourcen**

3.1 Testumgebung

Die Testfälle wurden in Visual Studio Code Version 1.79.2 durchgeführt, Das Betriebssystem ist Windows 11 64-Bit,Die Python-Version ist 3.11 und Das Backend-Code basiert auf dem Django-Framework (Rest Framework).

3.2 Testprotokoll

Testfall	***T2100 : test_ create_ resource***
Tester	*Fahd Ferjani*
Eingaben	*Ausführung des Testfalls python manage.py test resource.tests.test_ resource_ api.PrivateResourceeApiTests.test_ create_ resource im Projektverzeichnis.*
Soll - Reaktion	*Konsole zeigt "OK" an.*
Ist – Reaktion	*Die Ist-Reaktion entspricht der Soll-Reaktion.*
Ergebnis	*Der Test wurde erfolgreich durchgeführt.*
Unvorhergesehene Ereignisse	*keine*
Nacharbeiten	*keine*

Testfall	***T2200 :test_ delete_ resource:***
Tester	*Fahd Ferjani*

Eingaben	*Ausführung des Testfalls python manage.py test resource.tests.test_resource_api.PrivateResourceeApiTests.test_delete_resource im Projektverzeichnis.*
Soll - Reaktion	*Konsole zeigt "OK" an.*
Ist – Reaktion	*Die Ist-Reaktion entspricht der Soll-Reaktion.*
Ergebnis	*Der Test war erfolgreich.*
keine	*keine*
Nacharbeiten	*keine*

Testfall	*Beispiel:* ***T2300:test_update_resource:***
Tester	*Fahd Ferjani*
Eingaben	*Ausführung des Testfalls python manage.py test resource.tests.test_resource_api.PrivateResourceeApiTests.test_update_resource im Projektverzeichnis.*
Soll - Reaktion	*HKonsole zeigt "OK" an.*
Ist – Reaktion	*Die Ist-Reaktion entspricht der Soll-Reaktion.*
Ergebnis	*Der Test wurde erfolgreich durchgeführt*
Unvorhergesehene Ereignisse	*keine*
Nacharbeiten	*keine*

Testfall	***T2400: test_create_resource_only_mediathekar:***
Tester	*Fahd Ferjani*
Eingaben	*Ausführung des Testfalls python manage.py test resource.tests.test_resource_api.PrivateResourceeApiTests.test_create_resource_only_mediathekar im Projektverzeichnis.*
Soll - Reaktion	*Konsole zeigt "OK" an.*
Ist – Reaktion	*Die Ist-Reaktion entspricht der Soll-Reaktion.*
Ergebnis	*Der Test wurde erfolgreich durchgeführt.*
Unvorhergesehene Ereignisse	*Keine*
Nacharbeiten	*Keine*

3.3 Zusammenfassung

- Es wurden keine Abweichungen festgestellt.

- Die Testergebnisse deuten darauf hin, dass die Funktionen zum Hinzufügen, Bearbeiten und Löschen von Ressourcen erfolgreich implementiert wurden.
- Die Softwarequalität wurde nicht weiter bewertet, da weitere geplante Tests nicht durchgeführt wurden.

4 Testdurchführung (2023-07-11)

Art des Tests: Unit-tests
Ausgeführte Testfälle: ⟨**T1700**⟩, <**T1800**>, <**T1810**>, <**T1900**>, <**T2000**>
Beteiligte Tester: Fahd Ferjani
Abgedeckte Funktionen: <**F10**>**:Konto erstellen,** <**F20**>**:Konto anmelden**

4.1 Testumgebung

Die Testfälle wurden in Visual Studio Code Version 1.79.2 durchgeführt,Das Betriebssystem ist Windows 11 64-Bit,Die Python-Version ist 3.11 und Das Backend-Code basiert auf dem DjangoFramework (Rest Framework).

4.2 Testprotokoll

Testfall	***⟨T1700⟩:test_create_user_success***
Tester	*Fahd Ferjani*
Eingaben	*Ausführung des entsprechenden DjangoUnittests mit dem Befehl: python manage.py test user.tests.PublicUserApiTests. test_create_user_success*
Soll - Reaktion	*In der Konsole wird "OK" ausgegeben, wenn der Test erfolgreich ist*
Ist – Reaktion	*Die Ist-Reaktion entspricht der Soll-Reaktion..*
Ergebnis	*Der Test konnte erfolgreich durchgeführt werden.*
Unvorhergesehene Ereignisse	*keine*
Nacharbeiten	*keine*

Testfall	***⟨T1800⟩:test_create_token_for_non_activated_user:***
Tester	*Fahd Ferjani*

Eingaben	*Ausführung des entsprechenden DjangoUnittests mit dem Befehl: python manage.py test user.tests.AuthenticationApiTests.test_ create _ token_ for_ non_ activated_ user*
Soll - Reaktion	*In der Konsole wird "OK" ausgegeben, wenn der Test erfolgreich ist.*
Ist – Reaktion	*Die Ist-Reaktion entspricht der Soll-Reaktion*
Ergebnis	*Der Test konnte erfolgreich durchgeführt werden.*
keine	*keine*
Nacharbeiten	*keine*

Testfall	*Beispiel:* ***⟨T1810⟩:test_ create_ token_ for_ non_ activated_ user***
Tester	*Fahd Ferjani*
Eingaben	*Ausführung des entsprechenden DjangoUnittest smit dem Befehl: python manage.py test user.tests.AuthenticationApiTests.test_ create_ token_ for _ non_ activated_ user*
Soll - Reaktion	*In der Konsole wird "OK" ausgegeben, wenn der Test erfolgreich ist.*
Ist – Reaktion	*Die Ist-Reaktion entspricht der Soll-Reaktion.*
Ergebnis	*Der Test konnte erfolgreich durchgeführt werden*
Unvorhergesehene Ereignisse	*keine*
Nacharbeiten	*keine*

Testfall	***⟨T1900⟩:test_ update_ user_ profile:***
Tester	*Fahd Ferjani*
Eingaben	*Ausführung des entsprechenden DjangoUnittests mit dem Befehl: python manage.py test user.tests.PrivateUserApiTests.test_ update_ user_ profile*
Soll - Reaktion	*In der Konsole wird "OK" ausgegeben, wenn der Test erfolgreich ist.*
Ist – Reaktion	*Die Ist-Reaktion entspricht der Soll-Reaktion*
Ergebnis	*Der Test wurde erfolgreich durchgeführt.*
Unvorhergesehene Ereignisse	*Keine*
Nacharbeiten	*Keine*

Testfall	***⟨T2000⟩:test_ retrieve_ profile_ success:***
Tester	*Fahd Ferjani*

Eingaben	*Ausführung des entsprechenden DjangoUnittests mit dem Befehl: python manage.py test user.tests.UserProfileApiTests.test_retrieve_profile_success*
Soll - Reaktion	*In der Konsole wird "OK" ausgegeben, wenn der Test erfolgreich ist.*
Ist – Reaktion	*Die Ist-Reaktion entspricht der Soll-Reaktion*
Ergebnis	*Der Test wurde erfolgreich durchgeführt.*
Unvorhergesehene Ereignisse	*Keine*
Nacharbeiten	*Keine*

4.3 Zusammenfassung

- Die Unit-Tests wurden erfolgreich durchgeführt.
- Es traten keine Abweichungen auf.
- Die Funktionen für das Erstellen von Benutzern, Erzeugen von Tokens, Aktualisieren von Benutzerprofilen und Abrufen von Profilen wurden wie geplant getestet.
- Die Softwarequalität ist gut.

5 Testdurchführung (2023-07-11-003)

Art des Tests: Unittests
Ausgeführte Testfälle: **T2500**, **T2600**, **T2700**
Beteiligte Tester: Quynh Tran
Abgedeckte Funktionen: **F30**, **F120**, **Q30**

5.1 Testumgebung

Die Testfälle wurden unter MacOs Ventura 13.3 auf einem lokalen Webserver ausgeführt. Für den Backend-Code wurde das Django-Framework verwendet und der Webserver wurde mit dem Befehl "python manage.py runserver"gestartet. Der Frontend-Code wurde mit Svelte entwickelt und der Entwicklungsserver wurde mit dem Befehl "npm run dev"gestartet. Es wurde der Safarie-Browser in der Version 16.4 (64-bit) für die Durchführung der Tests verwendet.

5.2 Testprotokoll

Testfall	***T2500***
Tester	*Quynh Tran*
Eingaben	*Ausführung des entsprechenden Django-Unittests mit dem Befehl: python manage.py test resource.tests.SearchResourceTestCase.test_search_resource*
Soll - Reaktion	*Konsole zeigt 'Ok' an.*
Ist - Reaktion	*Die Ist-Reaktion entspricht der Soll-Reaktion.*
Ergebnis	*Test wurde erfolgreich ausgeführt.*
Unvorhergesehene Ereignisse	*Keine*
Nacharbeiten	*Keine*

Testfall	***T2600***
Tester	*Quynh Tran*
Eingaben	*Ausführung des entsprechenden Django-Unittests mit dem Befehl: python manage.py test resource.tests.SearchResourceTestCase.test_search_resource_filter*
Soll - Reaktion	*Konsole zeigt 'Ok' an.*
Ist - Reaktion	*Die Ist-Reaktion entspricht der Soll-Reaktion.*
Ergebnis	*Test wurde erfolgreich durchgeführt.*
Unvorhergesehene Ereignisse	*Keine*
Nacharbeiten	*Keine*

Testfall	***T2700***
Tester	*Quynh Tran*
Eingaben	*Ausführung des entsprechenden Django-Unittests mit dem Befehl: python manage.py test resource.tests.SearchResourceTestCase.test_search_resource_qrcode*
Soll - Reaktion	*Konsole zeigt 'Ok' an.*
Ist - Reaktion	*Die Ist-Reaktion entspricht der Soll-Reaktion.*
Ergebnis	*Der Testdurchlauf war erfolgreich.*
Unvorhergesehene Ereignisse	*Keine*
Nacharbeiten	*Keine*

5.3 Zusammenfassung

- Alle durchgeführten Tests waren erfolgreich und deckten die entsprechenden Testfälle ab
- Es wurden keine funktionalen Abweichungen festgestellt.
- Basierend auf den Testergebnissen kann die Komponenten für den beabsichtigten Zweck eingesetzt werden.
- Softwarequalität wurde als zufriedenstellend bewertet. Die durchgeführten Tests haben gezeigt, dass die grundlegenden Funktionen funktionieren.

6 Testdurchführung (2023-07-10-005)

Art des Tests: Unittests
Ausgeführte Testfälle: **T3200**, **T3300**, **T3400**
Beteiligte Tester: Quynh Tran
Abgedeckte Funktionen: **F70**

6.1 Testumgebung

Die Testfälle wurden unter MacOs Ventura 13.3 auf einem lokalen Webserver ausgeführt. Für den Backend-Code wurde das Django-Framework verwendet und der Webserver wurde mit dem Befehl "python manage.py runserver"gestartet. Der Frontend-Code wurde mit Svelte entwickelt und der Entwicklungsserver wurde mit dem Befehl "npm run dev"gestartet. Es wurde der Safarie-Browser in der Version 16.4 (64-bit) für die Durchführung der Tests verwendet.

6.2 Testprotokoll

Testfall	***T3200***
Tester	*Quynh Tran*
Eingaben	*Ausführung des entsprechenden Django-Unittests mit dem Befehl: python manage.py test favourite.tests.FavouriteResourceTestCase.test_add_resource.*
Soll - Reaktion	*Konsole zeigt 'Ok' an.*
Ist – Reaktion	*Die Ist-Reaktion entspricht der Soll-Reaktion.*
Ergebnis	*Der Testdurchlauf war erfolgreich.*
Unvorhergesehene Ereignisse	*Keine*
Nacharbeiten	*Keine*

Testfall	***T3300***
Tester	*Quynh Tran*
Eingaben	*Ausführung des entsprechenden Django-Unittests mit dem Befehl: python manage.py test favourite.tests.FavouriteResourceTestCase.test_share_resource _favouritelist.*
Soll - Reaktion	*Konsole zeigt 'Ok' an.*
Ist – Reaktion	*Die Ist-Reaktion entspricht der Soll-Reaktion.*
Ergebnis	*Test wurde erfolgreich durchgeführt.*
Unvorhergesehene Ereignisse	*Keine*
Nacharbeiten	*Keine*

Testfall	***T3400***
Tester	*Quynh Tran*
Eingaben	*Ausführung des entsprechenden Django-Unittests mit dem Befehl: python manage.py test favourite.tests.FavouriteResourceTestCase.test_remove_resource _favouritelist.*
Soll - Reaktion	*Konsole zeigt 'Ok' an.*
Ist – Reaktion	*Die Ist-Reaktion entspricht der Soll-Reaktion.*
Ergebnis	*Test wurde erfolgreich durchgeführt.*
Unvorhergesehene Ereignisse	*Keine*
Nacharbeiten	*Keine*

6.3 Zusammenfassung

- Alle durchgeführten Tests waren erfolgreich und deckten die entsprechenden Testfälle ab.
- Es wurden keine funktionalen Abweichungen festgestellt.
- Basierend auf den Testergebnissen kann die Komponenten für den beabsichtigten Zweck eingesetzt werden.
- Softwarequalität wurde als zufriedenstellend bewertet. Die durchgeführten Tests haben gezeigt, dass die grundlegenden Funktionen funktionieren.

7 Testdurchführung (2023-07-11-006)

Art des Tests: Unit-tests
Ausgeführte Testfälle: <**T2800**>, <**T2900**>, <**T3000**>, <**T3100**>
Beteiligte Tester: Subing Shen
Abgedeckte Funktionen: **F40 Verfügbarkeit, Ausleihen und Ausleih-Verlängerung**

7.1 Testumgebung

Die Testfälle wurden unter Windows 10 64-bit Version 22H2 ausgeführt. Die Python-Version war 3.9.2.Es wurde der Chrome-Browser in der Version "114.0.5735.199 (Official Build) (64-bit)" für die Durchführung der Tests verwendet.

7.2 Testprotokoll

Testfall	***T2800: Test_ borrow_ resource***
Tester	*Subing Shen*
Eingaben	*Ausführung des Testfalls python manage.py test Borrow.tests.BorrowResourceTestCase.test_ borrow_ resource*
Soll - Reaktion	*Die Funktion "test_ borrow_ resource"führt die erforderlichen Schritte aus und gibt das erwartete "OK" zurück.*
Ist – Reaktion	*Die Ist-Reaktion stimmt mit der Soll-Reaktion überein.*
Ergebnis	*Der Test war erfolgreich durchgeführt.*
Unvorhergesehene Ereignisse	*keine*
Nacharbeiten	*keine*

Testfall	***T2900: Test_ borrow_ resource_ not_ available***
Tester	*Subing Shen*

Eingaben	*Ausführung des Testfalls python manage.py test borrow.tests.BorrowResourceTestCase.test_borrow_resource_not_available*
Soll - Reaktion	*Die Funktion "test_borrow_resource"führt die erforderlichen Schritte aus und gibt das erwartete "OK" zurück.*
Ist – Reaktion	*Die Ist-Reaktion stimmt mit der Soll-Reaktion überein.*
Ergebnis	*Der Test war erfolgreich durchgeführt*
Unvorhergesehene Ereignisse	*keine*
Nacharbeiten	*keine*

Testfall	***T3000: Test_borrow_resource_invalid_resource***
Tester	*Subing Shen*
Eingaben	*Ausführung des Testfalls python manage.py test borrow.tests.BorrowResourceTestCase.test_borrow_resource*
Soll - Reaktion	*Die Funktion "test_borrow_resource"führt die erforderlichen Schritte aus und gibt das erwartete "OK" zurück.*
Ist – Reaktion	*Die Ist-Reaktion stimmt mit der Soll-Reaktion überein.*
Ergebnis	*Der Test war erfolgreich.*
Unvorhergesehene Ereignisse	*Keine*
Nacharbeiten	*Keine*

Testfall	***T3100: Test_borrow_resource_due_date***
Tester	*Subing Shen*
Eingaben	*Ausführung des Testfalls python manage.py test borrow.tests.BorrowResourceTestCase.test_borrow_resource_due_date*
Soll - Reaktion	*Die Funktion "test_borrow_resource"führt die erforderlichen Schritte aus und gibt das erwartete "OK" zurück.*
Ist – Reaktion	*Die Ist-Reaktion stimmt mit der Soll-Reaktion überein.*
Ergebnis	*Der Test war erfolgreich.*
Unvorhergesehene Ereignisse	*Keine*
Nacharbeiten	*Keine*

7.3 Zusammenfassung

- Alle durchgeführten Tests waren erfolgreich und deckten die entsprechenden Testfälle ab.
- Es wurden keine funktionalen Abweichungen festgestellt.
- Basierend auf den Testergebnissen kann die Komponenten für den beabsichtigten Zweck eingesetzt werden.
- Die Softwarequalität wurde als zufriedenstellend bewertet. Die durchgeführten Tests haben gezeigt, dass die grundlegenden Funktionen korrekt funktionieren.