

Testkonzept

ÜK-322_LB-01

Klassifizierung	GEHEIM
Status	in Arbeit
Programmname	LB-01
Projektnummer	450
Projektleiter	Adrian Jasaroska
Version	0.1
Datum	4. Dezember 2024
Auftraggeber	Patrick Michel
Autor/Autoren	Adrian Jasaroska
Verteiler	CsBe

Änderungsverzeichnis

Version	Datum	Änderung	Autor
0.1	04.12.2024	Initialer Start des Testkonzepts	Adrian Jasaroska

Tabelle 1: Änderungskontrolle

Beschreibung

Das Testkonzept beschreibt die Testziele, Testobjekte, Testarten, Testinfrastruktur sowie die Testorganisation. Es umfasst ebenfalls die Testplanung und die Testfallbeschreibungen. Für jeden Testfall wird eine detaillierte Testfallbeschreibung erstellt. Diese stellt die Spezifikation des Tests dar. Die Testplanung legt den logischen und zeitlichen Ablauf der Tests fest. Das Testkonzept bildet die Grundlage, auf der die Testorganisation und die Testinfrastruktur bereitgestellt und die Tests durchgeführt werden. Es wird bei neuen Erkenntnissen stets nachgeführt.

1 Testziele

Globale messbare Testziele über alle Testfälle hinweg:

Nr.	Beschreibung	Messgrösse	Priorität*
1	Usability Tests, Alle-Funktionen Funktionieren? (Einloggen / Ausloggen • Posts aufrufen • Posts erstellen • Posts löschen • Posts bearbeiten • Kommentare erstellen • Kommentare löschen • Kommentare Bearbeiten (• Likes • Dislikes) => Auf Posts und Kommentaren • Benachrichtigungen aufrufen • Admin kann Benutzer sperren/Bannen • Moderator kann Daten von anderen Benutzern löschen • Registrieren	Tester zum Testen holen, Ziele erklären	M
2	Code-Reviews, ist der Code verständlich? Zu kompliziert?	Hier wird ein externer Developer genehmigt um denn Source-Code einen Feedback zu geben, Code Review als Fachgespräch mit Herrn Michel	2
3	Acceptance/Benutzerfreundlichkeits Tests	Tester holen und eine Forms machen, um eine Bewertung der Benutzerfreundlichkeit mitzugeben	1

* Priorität: M = Muss / 1 = hoch, 2 = mittel, 3 = tief

Tabelle 2: Übergeordnete Testziele

2 Teststrategie und Teststufen

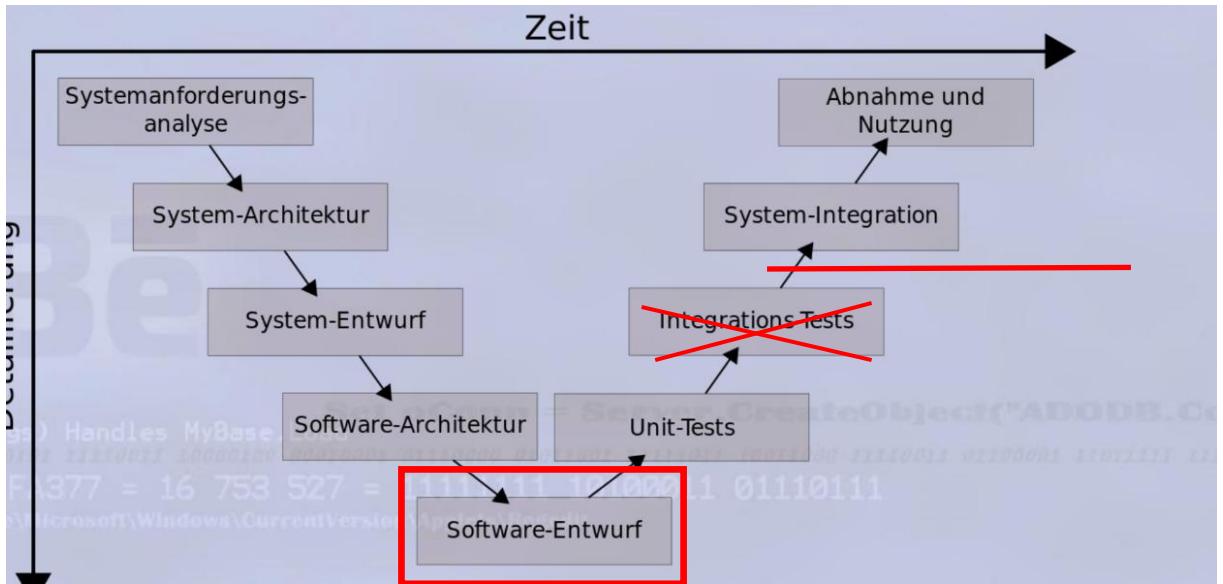
Effiziente Testausführung setzt durchdachte (und evtl. agil einsetzbare) Teststrategie voraus:

- Mit welcher Teststrategie kann die Entwicklung begleitet und ergänzt werden, um die vorgegeben Qualität zu erreichen?

Test-Driven-Developement, Höhere Code-Qualität/Schnelle Identifikation von Fehlern/Fördert einfache Designs.

Die Teststrategie basiert sich auf eine Agile Arbeitsgefühl, sodass wenn die Test-Phase wegen einem Kritischen Mangel temporär gestoppt werden muss, verzögert dies nicht noch alles andere

- Welche Teststufen gibt es / ab welcher Teststufe soll getestet werden?



Unit-Tests, Acceptance-Tests, System-Integration-Tests, Usability-Testing, Dazu noch Code-Reviews, ab der Rote Linie gemacht werden, ein anderer Software-Dev soll denn Code durchschauen und ein Feedback geben -> Ist der Code zu kompliziert? Refactoring möglich? Etc. All dies soll als Teststufen gemacht werden

Es soll ab der Software-Entwurf mit Tests begonnen werden. Schlussendlich war es mein Ziel solche Tests zu schreiben aber ich bin sehr weit von dem und konnte diese Tests nicht erstellen, weil ich selber Persönliche Probleme in den letzten Wochen hatte

- Wie können Überschneidungen von verschiedenen Teststufen vermieden werden (Verhinderung von Redundanzen)?

In dem man Schritt für Schritt durch dieses V-Modell durchgeht und jede Phase abchecken mit einem Häklein setzt und somit Redundanzen und Überschneidungen vermindern können, jedoch gibt es immer etwas dummes in jedem Algorithmus und somit gibt es keinen Versprechen dass alles fliessend laufen wird, es wird wahrscheinlich sogar sehr viele Probleme vorkommen, Software-Devs sind nun mal da um ein wenig denn Gehirn anzuschalten und zu debuggen 😊

- Wie können Defizite der verfügbaren Testumgebung umgangen/gemeistert werden? etc.

HARDWARE:

Wenn es Defizite im Hardware Bereich gibt, z.B. zu wenig RAM wenn Sie einen uralten PC verwenden, kann man nicht viel machen außer dass man den RAM erweitern bzw. mehr RAM einzubauen oder ein komplett anderes Gerät zu verwenden die natürlich gut genug ist um die Webseite zu betreiben.

SOFTWARE:

Bei Software können Sie jedoch etwas sinnvolleres erledigen, Sie können schauen dass Sie denn empfohlenen Browser mit der neusten LTS-Version benutzen, sowie Chromium, Mozilla Firefox oder auch Safari (nicht empfohlen)

Jedoch soll gecheckt werden ob Sie die angewandte Versionen von denn Paketen Manager, Node und JEST verwenden, bei denn Anderen Technologien müssen Sie sich keine Sorgen machen, mit NPM INSTALL werden die Dependencies die im package.json definiert sind automatisch installiert. TypeScript wird als Programmiersprache verwendet.

3 Testobjekte

Nr.	Objekt	Beschreibung
1	Frontend	<i>Die Benutzeroberfläche soll Benutzerfreundlich und Funktional funktionieren, sprich es soll alle Funktionen die in den Test-Zielen definiert wurden funktionieren.</i>
2	Backend	<i>Das Backend soll mithilfe der MariaDB Daten bekommen, Users speichern können und je nachdem ob der User ein Admin ist oder nicht, soll der User verschiedene Screens sehen. Auch die Rolle Moderator</i>
3	API	<i>Die Kommunikation zwischen der MariaDB im Docker und des Backends laufen, sodass die Webseite vollständig funktionieren kann. Die ganze Applikation in einem Docker</i>
4	DB	<i>Die DB soll funktional und Laufen, sodass es erfolgreich Daten zugewiesen bekommt.</i>

Tabelle 3: Testobjekte

4 Testarten

Nr.	Testart	Beschreibung
1	Usability-Testing	<i>Es werden Tests von User gemacht und die Funktionalität der Buttons werden getestet</i> <i>BT</i>
2	Acceptance-Testing	<i>Es werden Formulare ausgefüllt, die End-Benutzer unsere Benutzerfreundlichkeit bewerten können, es sollen mehrere Tester angewandt werden, wenn möglich</i> <i>BT</i>

Tabelle 4: Testarten

5 Testabdeckung

5.1 Übersicht Testfälle

Nr.	Testobjekt	Testart	Testfälle
1	Frontend	Usability-Testing	<p>Es wurden Tester eingeholt, um möglicherweise nicht funktionale Funktionen zu finden</p> <p>(Einloggen / Ausloggen • Posts aufrufen • Posts erstellen• Posts löschen • Posts bearbeiten • Kommentare erstellen • Kommentare löschen • Kommentare Bearbeiten (• Likes • Dislikes) => Auf Posts und Kommentaren • Benachrichtigungen aufrufen • Admin kann Benutzer sperren/Bannen • Moderator kann Daten von anderen Benutzern löschen • Registrieren</p>
2	Frontend	Acceptance-Testing	<p>Es werden Tester eingeholt, wenn möglich mehrere um die Benutzerfreundlichkeit zu Testen.</p> <p>1. Die Webseite ist nicht schön oder verstörend</p> <p>2. unklar was die Webseite zu bieten hat</p> <p>3. Die Buttons sind zu unklar oder nicht sehr ersichtlich</p>

Tabelle 5: Testabdeckung

6 Testrahmen

6.1 Testvoraussetzungen

Tester, Vorkenntnisse

Tester: Andere Person als der Entwickler, diesem Fall: **Adrian Jasaroska**

Der Tester sollte eine Person sein die nichts mit Applikationsentwicklung zu tun hat.

In diesem Fall: Adrian's Vater

6.2 Mängelklassifizierung

Die festgestellten Mängel, bzw. die nicht erfüllten Anforderungen (Erwartungen), werden in Klassen von M bis 4 eingestuft. Die Klasse 4 findet nur dann Verwendung, wenn ein einwandfreies Ergebnis gesondert ausgewiesen werden soll:

Nr.	Mängelklassen	Beschreibung
4	mängelfrei	Einwandfrei und anforderungsgerecht
3	belangloser Mangel	Verwendung möglich, Brauchbarkeit ist vorhanden, Mängel sollte dennoch nicht vorkommen
2	leichter Mangel	Verwendung möglich, Brauchbarkeit ist nur wenig beeinträchtigt
1	schwerer Mangel	Verwendung ist noch möglich, Brauchbarkeit ist stark verringert
M	kritischer Mangel	Unbrauchbar; Wesentliche Funktionalität ist nicht gegeben; Betrieb ist nicht verantwortbar (z.B. sicherheitsspezifisch)

Tabelle 6: Mängelklassen

Die Klassifizierung spiegelt die Folgenschwere und den Aufwand zur Behebung der möglichen feststellbaren Mängel. Die Zuordnung der festgestellten Mängel zu einer Mängelklasse gibt grob auch die Priorität vor, in welcher Reihenfolge die Behebung der Mängel angegangen werden soll.

Wird eine Mängelklasse zwischen 1-3 erreicht, kann das System/Produkt unter Vorbehalt abgenommen werden. Zur Behebung der Mängel sind jedoch Massnahmen zu definieren. Eine Nachprüfung ist zwingend.

Werden hingegen Mängel der Klasse 4 festgestellt, kann das System/Produkt nicht abgenommen werden und der Auftragnehmer muss umgehend Massnahmen treffen, um diese Mängel zu beheben. Der Auftragnehmer hat zudem die erneute Abnahme zu veranlassen.

6.3 Start- und Abbruchbedingungen

Vorbedingungen für Teststart

START:

Alle Funktionalitäten müssen für Tests optimal und Testsicher sein.

Die Testumgebung muss Stimmen wie ↓

ABBRUCH:

Falls es Funktionalitäten gibt, die nichtfunktionieren, werden die Test-Phasen temporär beendet und die Funktionalitäten repariert.

7 Testumgebung

Beschreibung und Organisation der erforderlichen (praxisnahen) Testumgebung (-en).

Alle Versionen der Pakete, im package.json und package-lock.json ersichtlich.

Empfohlene Hardwarestandarden:

M1 Pro (MacOS)

16GB RAM

1TB Speicher

<https://docs.docker.com/desktop/install/windows-install/>

Schauen Sie dass Sie auch diese oder bessere Hardware verwenden

LENOVO 21DH	
Microsoft Windows 11 Education (10.0, Build 22631)	
Processor:	12th Gen Intel(R) Core(TM) i7-1255U 10/12
Motherboard:	LENOVO LNVNB161216
RAM:	16 GB
Storage:	SAMSUNG MZAL4512HBLU-00BL2 (476.9 GB/Fixed hard disk media)
Graphics card:	Intel(R) Iris(R) Xe Graphics (1 GB)
Display:	Lenovo LEN9022 (1920x1080 / 13.9 Inch)
Audio1:	Proxy für Microsoft Streaming Service
Audio2:	Realtek(R) Audio

Software-Standarden

Docker:

<https://docs.docker.com/desktop/install/windows-install/>

- WSL version 1.1.3.0 or later.

Npm version: 10.8.2

Node version: v.20.17.0

Chromium Browser, Bei mir Google Chrome

Brave (Chromium), beim Entwickler Patrick Michel

8 Testinfrastruktur

8.1 Testsystem

Das Testsystem soll denn entsprechenden Hardware Standard und Software Standard treffen oder noch besser sein. Es soll genau die Technologien verwenden die im Repository verwendet werden

8.2 Testdaten

Bevor die Applikation gelaunched wird, werden Fake-Daten benutzt und bei jedes neues Feature soll auch Testdaten verwendet werden. Bsp wenn ein User deleted sein soll, sollte man dies zuerst mit Testdaten umfangreich Testen. Eine DB soll gemockt werden.

8.3 Testhilfsmittel

Software für das Testmanagement

Liste der Reports

Jest – Unit-Testing

E2E - Cypress

GitHub Repository Pushing and Pulling

9 Testorganisation

Wer hat welche Rolle im Testprozess.

Entwickler: Adrian Jasaroska

Tester: Adrian's Vater

Andere Tester: End-Benutzer

10 Quellenverzeichnis

Projekt Repo vom Entwickler	https://github.com/SwitzerChees/simple-typescript-template
Docker Installation und mindest Hardware	https://docs.docker.com/desktop/install/windows-install/
Sonstige Informationen aus dem ÜK-223 in der CsBe, Dozent: Patrick Michel	

Abkürzungen und Glossar

Abkürzung / Fachwort	Erläuterung
HERMES	Vorgehensmethodik für Projekte und Programme HERMES 5 ist ein eCH Standard

Tabelle 7: Abkürzungen und Glossar

Inhaltsverzeichnis

Änderungsverzeichnis.....	1
Beschreibung.....	1
1 Testziele	2
2 Teststrategie und Teststufen	2
3 Testobjekte	4
4 Testarten.....	5
5 Testabdeckung	6
5.1 Übersicht Testfälle	6
5.2 Beurteilung Testziele und Testabdeckung ..	Error! Bookmark not defined.
6 Testrahmen.....	6
6.1 Testvoraussetzungen.....	6
6.2 Mängelklassifizierung	6
6.3 Start- und Abbruchbedingungen	6
7 Testumgebung	8
8 Testinfrastruktur.....	9
8.1 Testsystem.....	9
8.2 Testdaten	9
8.3 Testhilfsmittel.....	10
9 Testorganisation	10
10 Testfallbeschreibungen.....	Error! Bookmark not defined.
11 Testplan	Error! Bookmark not defined.
<i>Abkürzungen und Glossar.....</i>	<i>10</i>
Inhaltsverzeichnis	11
Tabellenverzeichnis	12

Tabellenverzeichnis

Tabelle 1: Änderungskontrolle.....	1
Tabelle 2: Übergeordnete Testziele.....	2
Tabelle 2: Testobjekte	4
Tabelle 3: Testarten.....	5
Tabelle 4: Testabdeckung	6
Tabelle 5: Mängelklassen.....	Error! Bookmark not defined.
Tabelle 6: Testfallbeschreibung	Error! Bookmark not defined.
Tabelle 7: Testplan.....	Error! Bookmark not defined.
Tabelle 5: Abkürzungen und Glossar	10

Testprotokoll

ÜK223_LB-01

Klassifizierung GEHEIM
Status in Arbeit
Programmname LB-01
Projektnummer 332
Projektleiter Adrian Jasaroska
Version 0.1
Datum 13. September 2024
Auftraggeber Patrick Michel
Autor/Autoren Microsoft Office User
Verteiler CsBe

Änderungsverzeichnis

Version	Datum	Änderung	Autor
0.1	04.12.2024	Initialer Start des Testprotokolls	Adrian Jasaroska

Tabelle 8: Änderungskontrolle

Beschreibung

Das Testprotokoll hält die Testergebnisse fest. Die Testergebnisse sind gemäss den im Testkonzept definierten Mängelklassen bewertet.

Übersicht der Testfälle / Testdurchführungen

Hinweis auf das Testkonzept

ID	Bezeichnung	Testdatum	Tester	FK*
T-001	Login/Logout	04.12.2024	Adrian Jasaroska	M
T-002	Registration	04.12.2024	Adrian Jasaroska	M
T-003	Posts erstellen	04.12.2024	Adrian Jasaroska	M
T-004	Posts löschen	04.12.2024	Adrian Jasaroska	1
T-005	Posts Bearbeiten	04.12.2024	Adrian Jasaroska	2
T-006	Kommentare erstellen	04.12.2024	Adrian Jasaroska	1
T-007	Benutzerprofil einsehen und bearbeiten	04.12.2024	Adrian Jasaroska	3
T-008	Kommentare löschen	04.12.2024	Adrian Jasaroska	3
T-009	Kommentare Bearbeiten	04.12.2024	Adrian Jasaroska	1
T-010	Feed einlesen	04.12.2024	Adrian Jasaroska	M
T-011	Benachrichtigungen bekommen	04.12.2024	Adrian Jasaroska	1
T-012	Posts und Kommentare Liken/Disliken	04.12.2024	Adrian Jasaroska	2
T-013	Admin Dashboard mit Admin Funktionen	04.12.2024	Adrian Jasaroska	2
T-014	Moderator Dashboard mit Moderator Funktionen	04.12.2024	Adrian Jasaroska	2

11 Testfall T-001: Login/Logout

ID / Bezeichnung	T-001 Login/Logout
Beschreibung	Eine Login-Maske soll existieren und es soll im Hauptmenü ausgeloggt werden können.
Testvoraussetzung	Die Applikation muss starten, die Login-Maske muss erscheinen. 1. Applikation starten. 2. Login-Daten eingeben. 3. Auf den Login-Button klicken. 4. Logout im Hauptmenü ausführen.
Testschritte	
Erwartetes Ergebnis	Erfolgreicher Login bringt den Nutzer zur Hauptseite, Logout bringt ihn zurück zur Login-Seite.
Testdatum	04.12.2024
Tester	Adrian Jasaroska
Mängelklasse	M
Mangelbeschreibung	Login funktioniert korrekt, Logout ebenso.
Bemerkungen	Alles in Ordnung, Login-Daten: Benutzer: adi, Passwort: supersecret123.

12 Testfall T-002: Registrierung

ID / Bezeichnung	T-002 Registrierung
Beschreibung	Neue Benutzer sollen sich registrieren können.
Testvoraussetzung	Eine Registrierungsmaske muss existieren. 1. Registrierungsmaske aufrufen. 2. Daten eingeben. 3. Registrieren-Button drücken.
Testschritte	
Erwartetes Ergebnis	Registrierung erfolgreich, Nutzer kann sich direkt einloggen.
Testdatum	04.12.2024
Tester	Adrian Jasarovska
Mängelklasse	1
Mangelbeschreibung	Registrierung ist nicht funktionsfähig.
Bemerkungen	Die Funktion wurde nicht implementiert.

13 Testfall T-003: Posts erstellen

ID / Bezeichnung	T-003 Posts erstellen
Beschreibung	Benutzer können neue Posts erstellen, die im Feed sichtbar sind.
Testvoraussetzung	"Post erstellen"-Funktion muss sichtbar sein.
	1. "Post erstellen" öffnen. 2. Titel und Inhalt eingeben. 3. "Erstellen"-Button drücken.
Testschritte	
Erwartetes Ergebnis	Der Post wird im Feed angezeigt.
Testdatum	04.12.2024
Tester	Adrian Jasarovska
Mängelklasse	M
Mangelbeschreibung	Posts können nicht erstellt werden.
Bemerkungen	Die Funktion wurde nicht implementiert.

14 Testfall T-004: Posts löschen

ID / Bezeichnung	T-004 Posts löschen
Beschreibung	Benutzer sollen ihre eigenen Posts löschen können.
Testvoraussetzung	Ein existierender Post muss vorhanden sein.
Testschritte	<ol style="list-style-type: none">1. Post auswählen.2. "Löschen" klicken.3. Bestätigung ausführen.
Erwartetes Ergebnis	Post wird erfolgreich gelöscht.
Testdatum	04.12.2024
Tester	Adrian Jasarovska
Mängelklasse	1
Mangelbeschreibung	Posts können nicht gelöscht werden.
Bemerkungen	Die Funktion wurde nicht implementiert.

15 Testfall T-005: Posts bearbeiten

ID / Bezeichnung	T-005 Posts bearbeiten
Beschreibung	Benutzer sollen ihre Posts nachträglich bearbeiten können.
Testvoraussetzung	Ein existierender Post muss vorhanden sein.
Testschritte	<ol style="list-style-type: none">1. Post auswählen.2. "Bearbeiten" klicken.3. Änderungen speichern.
Erwartetes Ergebnis	Änderungen werden übernommen und angezeigt.
Testdatum	04.12.2024
Tester	Adrian Jasarovska
Mängelklasse	2
Mangelbeschreibung	Posts können nicht bearbeitet werden.
Bemerkungen	Die Funktion wurde nicht implementiert.

16 Testfall T-006: Kommentare erstellen

ID / Bezeichnung **T-006 Kommentare erstellen**

Beschreibung Benutzer sollen Kommentare unter Posts erstellen können.

Testvoraussetzung Ein Post muss existieren, unter dem kommentiert werden kann.

1. Post öffnen.

Testschritte 2. Kommentar eingeben.

3. "Posten"-Button drücken.

Erwartetes Ergebnis Kommentar wird unter dem Post angezeigt.

Testdatum **04.12.2024**

Tester Adrian Jasarovska

Mängelklasse 1

Mangelbeschreibung Kommentare können nicht erstellt werden.

Bemerkungen Die Funktion wurde nicht implementiert.

16 Testfall T-007: Kommentare löschen

ID / Bezeichnung	T-007 Kommentare löschen
Beschreibung	Benutzer sollen ihre eigenen Kommentare löschen können.
Testvoraussetzung	Ein eigener Kommentar muss existieren.
Testschritte	<ol style="list-style-type: none">1. Kommentar auswählen.2. "Löschen"-Button klicken.3. Bestätigung ausführen.
Erwartetes Ergebnis	Kommentar wird erfolgreich entfernt.
Testdatum	04.12.2024
Tester	Adrian Jasarovska
Mängelklasse	1
Mangelbeschreibung	Kommentare können nicht gelöscht werden.
Bemerkungen	Die Funktion wurde nicht implementiert.

17 Testfall T-008: Likes vergeben

ID / Bezeichnung	T-008 Likes vergeben
Beschreibung	Benutzer sollen Posts oder Kommentare liken können.
Testvoraussetzung	Ein Post oder Kommentar muss existieren.
Testschritte	<ol style="list-style-type: none">Post oder Kommentar öffnen."Like"-Button klicken.
Erwartetes Ergebnis	Like wird erfolgreich hinzugefügt und angezeigt.
Testdatum	04.12.2024
Tester	Adrian Jasaroska
Mängelklasse	1
Mangelbeschreibung	Likes können nicht vergeben werden.
Bemerkungen	Die Funktion wurde nicht implementiert.

18 Testfall T-009: Likes entfernen

ID / Bezeichnung	T-009 Likes entfernen
Beschreibung	Benutzer sollen vergebene Likes wieder entfernen können.
Testvoraussetzung	Ein vorher vergebener Like muss existieren.
Testschritte	1. Post oder Kommentar mit Like öffnen. 2. "Like entfernen"-Button klicken.
Erwartetes Ergebnis	Like wird erfolgreich entfernt.
Testdatum	04.12.2024
Tester	Adrian Jasarovska
Mängelklasse	2
Mangelbeschreibung	Likes können nicht entfernt werden.
Bemerkungen	Die Funktion wurde nicht implementiert.

19 Testfall T-010: Suche nach Posts

ID / Bezeichnung	T-010 Suche nach Posts
Beschreibung	Benutzer sollen Posts anhand von Keywords suchen können.
Testvoraussetzung	Eine Suchleiste muss existieren. 1. Suchleiste aufrufen.
Testschritte	2. Keyword eingeben. 3. Auf "Suchen" klicken.
Erwartetes Ergebnis	Relevante Posts werden angezeigt.
Testdatum	04.12.2024
Tester	Adrian Jasarovska
Mängelklasse	1
Mangelbeschreibung	Suche liefert keine Ergebnisse.
Bemerkungen	Die Funktion wurde nicht implementiert.

20 Testfall T-011: Passwort zurücksetzen

ID / Bezeichnung	T-011 Passwort zurücksetzen
Beschreibung	Benutzer sollen ihr Passwort über die "Passwort zurücksetzen"-Funktion ändern können.
Testvoraussetzung	Eine "Passwort zurücksetzen"-Funktion muss vorhanden sein. 1. "Passwort zurücksetzen" öffnen. 2. E-Mail-Adresse eingeben. 3. Anweisungen befolgen.
Erwartetes Ergebnis	Passwort wird erfolgreich geändert.
Testdatum	04.12.2024
Tester	Adrian Jasaroska
Mängelklasse	M
Mangelbeschreibung	Funktion ist nicht implementiert.
Bemerkungen	Die Funktion wurde nicht implementiert.

21 Testfall T-012: Profil bearbeiten

ID / Bezeichnung	T-012 Profil bearbeiten
Beschreibung	Benutzer sollen persönliche Daten im Profil ändern können.
Testvoraussetzung	Profilbearbeitungsseite muss existieren. 1. Profil öffnen. 2. Änderungen vornehmen. 3. Änderungen speichern.
Erwartetes Ergebnis	Änderungen werden übernommen und gespeichert.
Testdatum	04.12.2024
Tester	Adrian Jasarovska
Mängelklasse	2
Mangelbeschreibung	Profil kann nicht bearbeitet werden.
Bemerkungen	Die Funktion wurde nicht implementiert.

22 Testfall T-013: Benachrichtigungen anzeigen

ID / Bezeichnung	T-013 Benachrichtigungen anzeigen
Beschreibung	Benutzer sollen Benachrichtigungen für Aktivitäten sehen.
Testvoraussetzung	Es muss Benachrichtigungen im System geben.
Testschritte	1. Benachrichtigungscenter öffnen. 2. Benachrichtigungen überprüfen.
Erwartetes Ergebnis	Benachrichtigungen werden korrekt angezeigt.
Testdatum	04.12.2024
Tester	Adrian Jasaroska
Mängelklasse	2
Mangelbeschreibung	Benachrichtigungen werden nicht angezeigt.
Bemerkungen	Die Funktion wurde nicht implementiert.

23 Testfall T-014: Abmeldung erzwingen

ID / Bezeichnung	T-014 Abmeldung erzwingen
Beschreibung	Administratoren sollen Benutzer zwangsabmelden können.
Testvoraussetzung	Admin-Berechtigungen müssen vorhanden sein.
Testschritte	1. Benutzer auswählen. 2. Auf "Abmelden erzwingen" klicken.
Erwartetes Ergebnis	Benutzer wird abgemeldet.
Testdatum	04.12.2024
Tester	Adrian Jasaroska
Mängelklasse	M
Mangelbeschreibung	Funktion ist nicht implementiert.
Bemerkungen	Die Funktion wurde nicht implementiert.

Inhaltsverzeichnis

1 Testfall.....	Error! Bookmark not defined.
2 Testfall.....	Error! Bookmark not defined.
3 Testfall.....	Error! Bookmark not defined.
4 Testfall.....	Error! Bookmark not defined.
5 Testfall.....	Error! Bookmark not defined.
6 Testfall.....	Error! Bookmark not defined.
7 Testfall.....	Error! Bookmark not defined.
8 Testfall.....	Error! Bookmark not defined.
9 Testfall.....	Error! Bookmark not defined.
10 Testfall.....	Error! Bookmark not defined.
11 Testfall.....	Error! Bookmark not defined.
12 Testfall.....	Error! Bookmark not defined.
13 Testfall.....	Error! Bookmark not defined.