

# **Qualitätssicherung für das Projekt "Schnatter"**

## **Qualitätsziele und Maßnahmen**

Im Rahmen unseres Projekts "Schnatter" haben wir verschiedene Qualitätsziele definiert und Maßnahmen implementiert, um diese zu erreichen. Im Folgenden werden die wichtigsten Qualitätsziele und die dazugehörigen Maßnahmen beschrieben.

### **1. Usability**

**Qualitätsziel:** Eine hohe Benutzerfreundlichkeit der Anwendung sicherstellen.

## **Benutzerstudie 1: Ende Januar**

### **Teilnehmendeninformationen**

Anzahl der Teilnehmende: n = 4

#### Demografische Daten:

- Alter:
  - unter 20: 1 Teilnehmende
  - 21 - 30: 1 Teilnehmende
  - 31 - 40: 1 Teilnehmende
  - über 50: 1 Teilnehmende
- Geschlecht:
  - weiblich: 1 Teilnehmende
  - männlich: 2 Teilnehmende
  - divers: 1 Teilnehmende

### **Durchführung**

Den Nutzern wurden spezifische Aufgaben gestellt, die sie ausführen mussten. Nach der Erledigung der Aufgaben wurde den Teilnehmern Zeit zum freien Erkunden der Webseite "Schnatter" gegeben. Die Nutzung der Seite wurde beobachtet und Auffälligkeiten wurden dokumentiert. Anschließend mussten die Teilnehmer Aussagen auf einer Skala von 0 bis 10 bewerten und Freitextfragen beantworten. Im Nachhinein hätten Ja/Nein-Antworten möglicherweise besser geeignet sein können.

### **Aufgaben:**

1. Lege dir ein Nutzerprofil mit deinen Daten an und logge dich auf der Webseite "Schnatter" ein.
2. Erstelle einen Post mit einem kurzen Text deiner Wahl (nutze Hashtags) und poste diesen.
3. Suche den User "lisa" und folge dem User "lisa".
4. Suche nach Posts mit dem Hashtag "cool".
5. Öffne dein eigenes Profil und lösche deinen eben erstellten Text-Beitrag.
6. Ändere die Anzeige-Sprache auf Deutsch.

7. Ändere den Modus auf Light-Mode.

### **Bewertungsfragen (Skala 0-10):**

8. Ich wusste durchgehend, wo ich hinklicken musste, um die gewünschte Seite zu erreichen. Falls nein, an welchen Stellen war das nicht der Fall?
9. Die Startseite hat mir einen Überblick über alle grundlegenden Funktionen geboten. Falls nein, was hat gefehlt?
10. Das Farbschema, die Schriften und Designelemente waren auf allen Seiten der App konsistent. Falls nein, wo gab es Abweichungen?
11. Alle Links und Buttons haben mich zu den zu erwartenden Seiten geführt. Falls nein, wo gab es eine fehlerhafte Weiterleitung?
12. Bei den Input-Feldern wurden die Eingaben so erwartet, wie ich es intuitiv vermutet habe, oder ich habe das angegebene Format verstanden. Falls nein, wo kam es zu Missverständnissen bei der Eingabe?
13. Bei fehlerhaften Eingaben meinerseits in den Input-Feldern wurden mir verständliche Fehlermeldungen gezeigt, sodass ich den Fehler beheben konnte. Falls nein, wo gab es unverständliche Fehlermeldungen?
14. Interaktive Elemente wie Toggles oder Regler waren klar zu erkennen und die Benennung entsprach meinem Verständnis. Falls nein, wo waren die Toggles etc. unpassend benannt?
15. Wechseln Sie den Browser. Wird die Seite dort auch fehlerfrei angezeigt? Falls nein, in welchen Browsern kommt es zu einer fehlerhaften Darstellung?
16. Die Ladezeiten waren angenehm und ich habe keine benutzerunfreundlichen Wartezeiten erlebt. Falls nein, wo war der Ladevorgang sehr lange?

### **Freitextfragen:**

17. Bewerten Sie in kurzen Sätzen die allgemeine Benutzerfreundlichkeit und die Intuitivität der App.
18. Diese Aspekte sind mir positiv aufgefallen:
19. Diese Aspekte sind mir negativ aufgefallen:
20. Diese Aspekte würde ich mir anders wünschen:
21. Platz für weiteres Feedback:

### **Ergebnisse und Erkenntnisse**

Die Antworten und die Erkenntnisse aus den Beobachtungen wurden zusammengefasst und bildeten die Grundlage für Verbesserungen. Die [Durchführung](#) und die [Ergebnisse](#) wurden im [Dokumentationsrepository](#) (Link hinzufügen) dokumentiert. Daraus ergaben sich mehrere kleine [Aufgaben](#), die ebenfalls im Doku-Repo zusammengetragen wurden.

### **Allgemeine Verbesserungen:**

- Ist die Seite auch auf dem Handy nutzbar?
- Cookie-Einstellungen erfragen.
- Warum ist die Seite "nicht sicher"?

## **Suchfunktion:**

- Die Suchfunktion ist nicht intuitiv.
- Hashtag-Suche klappt nicht, wenn man das "#" mitschreibt.
- Bei der Suche vielleicht eine grüne Markierung für ausgewählte Elemente?
- Nach Emojis kann nicht gesucht werden.
- Beim Aufrufen der Suche ist der Fokus nicht auf dem Input-Feld.
- Arbeiten mit Records bei "Load More".
- Suchfunktion ist case-sensitive.
- Text statt Icons bei der Suche bevorzugen.

## **Posts:**

- Keine Nachfrage, ob man den Post wirklich löschen möchte.
- Uhrzeit zu Posts hinzufügen.
- Hashtags sind nicht anklickbar.
- Keine Oberfläche für Emojis, nur einfügbar.
- Posts können nicht bearbeitet werden.
- Fokus auf Input-Feld beim Erstellen eines neuen Posts.

## **Profil:**

- Bei Follow: Toast sagt, man kann User nicht folgen und der Button ändert sich nicht, nach Refresh ist der User jedoch gefolgt.
- Man sieht nicht, wem man folgt oder wer einem folgt.
- Seite muss neu geladen werden, um von "Abonniert/Folgen" wegzukommen.
- "Der Nutzer hat noch keine Beiträge" ist zu groß geschrieben.
- Nickname ändern hat keine Max-Length.
- "Load-More" Button erscheint schon bei zwei eigenen Posts (sinnvoller erst bei mehr als 10).
- Kein Hinweis, dass Status möglich ist und wie man ihn anpasst.
- Speicher-Button bei Status bearbeiten nicht intuitiv, Vorschlag: Speicher-Button unten einfügen.
- Mehrere Zeilen im Status werden automatisch zusammengezogen.
- "Load-More" Button zeigt leeren Post, wenn es keine weiteren Posts gibt und macht die Seite kaputt.

## **Passwort ändern:**

- Altes Passwort wird klar angezeigt.
- Beim Ändern des Passworts steht "Registrieren" statt "Ändern".

## **Fehlende/Falsche Übersetzungen:**

- Keine Übersetzung bei Namen ändern.
- E-Mail bei Registrierung ist auf Englisch, auch wenn auf Deutsch eingestellt.
- "Verify Account" ist nicht übersetzt.
- Hover über Nav-Bar ist nur auf Englisch.
- Sollte "Abonnieren" lieber "Folgen" heißen?

### **Login:**

1. Man will E-Mail angeben anstatt Benutzernamen.

### **Nav-Bar:**

- Man klickt erst auf "Messages", wenn man posten möchte.
- Beim Klick auf die Ente sollte ein Geräusch kommen.
- Beim Klick auf das Enten-Icon geht es nicht zurück zur Home-Page.
- Profil-Icon sollte gefüllt sein, damit es einheitlicher aussieht.

### **Späteres Feedback:**

- Keine Aktualisierung nach Follow.
- Standortanzeige ist zu genau, eventuell unsicher (exakte Koordinaten sind einsehbar).

Nicht alle Punkte wurden als sinnvoll erachtet und daher nicht alle umgesetzt.

## **Benutzerstudie 2 – Eyetracking-Studie: Juni 2024**

### **Teilnehmendeninformationen**

Anzahl der Teilnehmenden: n = 8

#### Demografische Daten:

- Alter:
  - 20 - 30: 8 Teilnehmende
- Geschlecht:
  - weiblich: 2 Teilnehmende
  - männlich: 6 Teilnehmende

### **Durchführung**

Für die Durchführung der Eyetracking-Studie wurde ein Laptop-Set mit Tobii Pro Lab vom ZEEB ausgeliehen und die vorbereiteten Inhalte auf dem Laptop den Teilnehmenden präsentiert. Die Teilnehmenden mussten verschiedene Aufgaben an einem Laptop erfüllen, wobei der Fokus auf den Hauptfunktionalitäten der Anwendung lag. Die Aufgaben wurden schriftlich auf dem Bildschirm angezeigt und hatten in der Regel ein Zeitlimit. Während der Aufgabenbearbeitung wurden die Augenbewegungen der Teilnehmenden getrackt und ihr Verhalten beobachtet.

## Aufgaben

Reihenfolge der Slides und [Aufgabenbeschreibungen](#), die die Teilnehmenden sahen:

- 1) Kalibrierung
- 2) Begrüßung
- 3) Vorstellung des Projektsschnatter
- 4) Erklärung, was jetzt passiert
- 5) Aufgabe 1: Registrierung
- 6) Aufgabe 2: Login und Einstellungen
- 7) Aufgabe 3: Posting
- 8) Aufgabe 4: Nutzersuche und Interaktionen
- 9) Aufgabe 5: Repost
- 10) Aufgabe 6: Hashtag-Suche
- 11) Dank

Die Slides sind hier hinterlegt.

### Beispiel – Aufgabe 1: Registrierung

#### Aufgabe 1/6: Registrierung

Nach dieser Slide öffne bitte **Google Chrome**. Dort findest du zwei **Lesezeichen**.

Gehe zunächst auf das Lesezeichen **Müllmail** und lege dir dort eine neue Müllmail an.

Dann öffne in einem neuen Tab die Seite "**Schnatter**" (zweites Lesezeichen).

Deine Aufgabe ist es nun, dich mit deiner Müllmail bei Schnatter zu **registrieren**. Merke dir deinen Usernamen und Passwort gut!

Bist du registriert, beende diese Aufgabe mit **F10**.

### Beispiel – Aufgabe 2: Posting

#### Aufgabe 3/6: Post

**Poste** nun einen Text deiner Wahl. Verwende dabei min. 1 **Hashtag** und gebe auch deinen **Standort** frei.

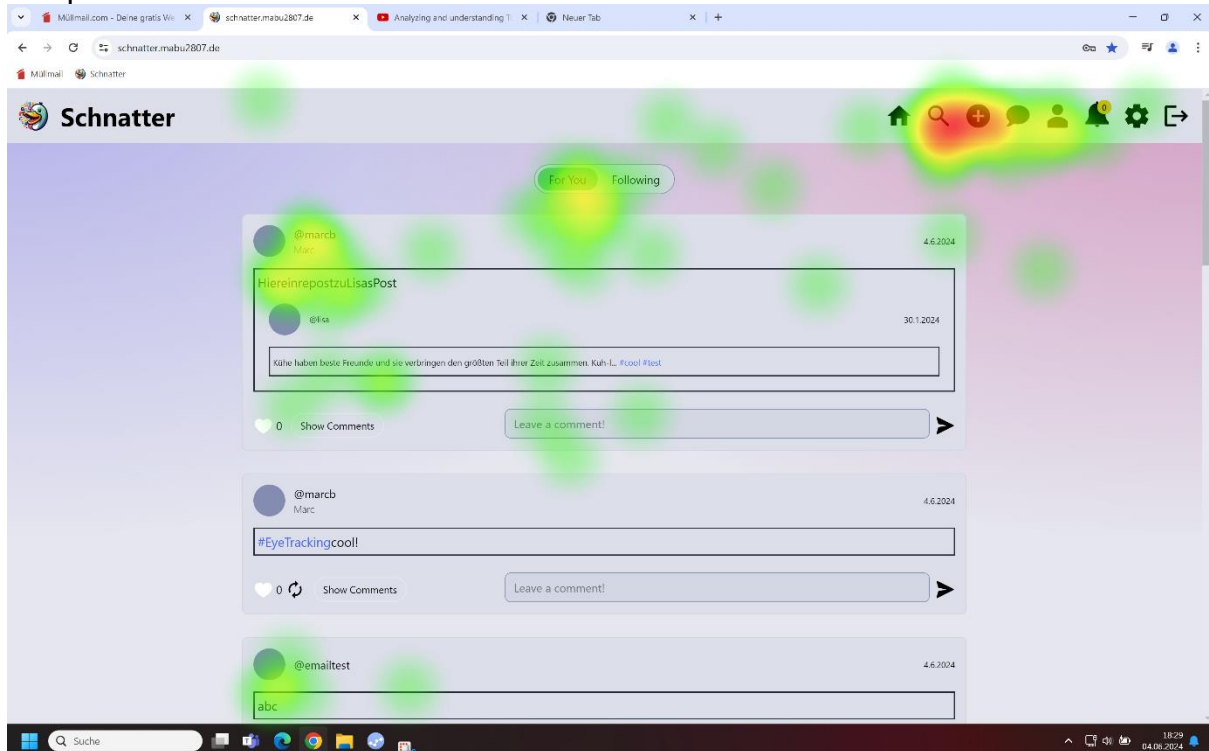
Du hast **1:30 Minuten** Zeit für diese Aufgabe.

Bist du früher fertig, beende diese Aufgabe mit **F10**.

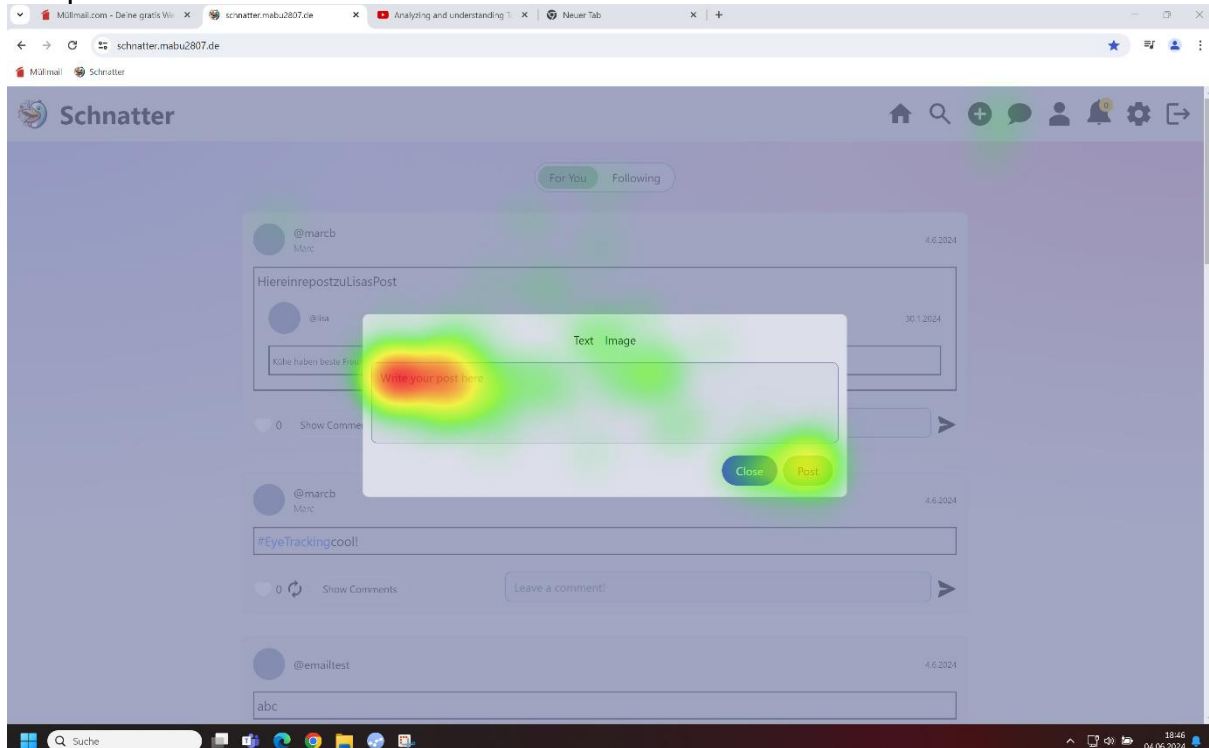
## Ergebnisse und Erkenntnisse

Die Augenbewegungen wurden im Anschluss auf Snapshots gemappt und in [Heatmaps](#) zusammengefasst, die alle im Dokumentationsrepository hinterlegt sind.

### Beispiel – Suche nach Such-Icon



### Beispiel – Post



Retrospektiv hätte eine A/B-Analyse wertvollere und detailliertere Erkenntnisse liefern können. Allerdings war dies aus Zeitgründen nicht möglich und hätte den Aufwand der Auswertung erheblich erhöht.

Bei der Beobachtung der Aufgabenbearbeitung wurden zusätzlich folgende Erkenntnisse gewonnen:

- Fehler bei der E-Mail-Registrierung
- Probleme beim Posten festgestellt

Diese Fehler wurden im Anschluss behoben.

## **Benutzerstudie 3: Ende Juni/Anfang Juli**

### **Teilnehmendeninformationen**

Anzahl der Teilnehmende: n = 4

#### Demografische Daten:

- Alter:
  - unter 20: 1 Teilnehmende
  - 21 - 30: 1 Teilnehmende
  - 31 - 40: 1 Teilnehmende
  - über 50: 1 Teilnehmende
- Geschlecht:
  - weiblich: 1 Teilnehmende
  - männlich: 2 Teilnehmende
  - divers: 1 Teilnehmende

### **Durchführung**

Die Durchführung glich der der ersten Benutzerstudie.

### **Aufgaben:**

1. lege dir ein Nutzerprofil mit Profilbild an
2. logge dich auf der Webseite Schnatter ein
3. erstelle einen Bild-Post mit einem kurzen Text deiner Wahl (nutze Hashtag) und poste diesen
4. suche den User "lisa" und folge dem User "lisa"
5. suche nach Posts mit dem Hashtag "cool"
6. gehe auf "Following"; like und kommentiere einen Post
7. reposte einen Post deiner Wahl
8. Finde heraus, warum du eine Benachrichtigung erhalten hast
9. Beginne einen Chat mit dem User "lisa"

**Bewertungsfragen (Skala 0-10):** glichen der der ersten Benutzerstudie

**Freitextfragen:** glichen der der ersten Benutzerstudie

## Ergebnisse und Erkenntnisse

Die Antworten und die Erkenntnisse aus den Beobachtungen wurden zusammengefasst und bildeten die Grundlage für Verbesserungen. Die [Durchführung](#) und die [Ergebnisse](#) wurden im [Dokumentationsrepository](#) (Link hinzufügen) dokumentiert. Daraus ergaben sich mehrere kleine [Aufgaben](#), die ebenfalls im Doku-Repo zusammengetragen wurden.

## Interaktionen

- Nach dem Abschicken eines Kommentars springt man automatisch auf den Anfang der Seite, auch wenn man das nicht will
- Erster Repost wird doppelt gepostet
- Erkenntlich machen ob etwas ein Repost ist: Über das Modal "Repost" schreiben
- Hashtags klickbar machen
- Anzeigen, wie viele/ob Kommentare geschrieben wurde; Vorschlag: Kommentarbutton unterlegen (filled), falls Kommentare existieren
- Repost ohne weiteren Inhalt geht nicht (ohne Text)

## Chat

- Chat starten ist unintuitiv
- Chat: Leere Nachricht abschicken sendet Nachricht mit Text undefined und Zeit NAN NAN NAN NAN NAN, dann ist die Reihenfolge des Chats zerstört. Bei erneutem Öffnen ist die Nachricht wieder weg
- Person A schreib Person B zum ersten Mal eine Nachricht. Person B hat das Chat-Fenster geöffnet. Der neue Chat wird der Person B nicht angezeigt, dazu muss zunächst das Fenster geschlossen und danach wieder geöffnet werden.
- Nachricht im Chat: Nach dem Abschicken einer Nachricht ist Schreibfeld nicht mehr aktiviert/fokussiert, man muss wieder reinklicken --> bitte drin bleiben
- Such-Button erst aber mehr als 2 Chats anzeigen, sonst zu verwirrend

## Notifications

- Notification wird nicht angezeigt, wenn Seite nicht aktualisiert wird
- Wenn man im eigenen Profil ist, funktioniert die Benachrichtigungsweiterleitung zu anderen Profilen nicht
- Weiterleitung zu Profil nicht nur wenn man auf Namen klickt, sondern ganze Kachel

## Login

- Bug: Man loggt sich ein, die Seite aktualisiert sich aber nicht (anscheinend keine Reaktion auf Button-Klick), erst bei Neuladen der Seite ist man eingeloggt
- Möglichkeit einbauen, das Passwort sichtbar zu machen



## **Post**

- Schreibt man im Post einen Hashtag und macht danach einen Zeilenumbruch, wird im Post das nächste Wort auch Blau (Anschauungsbeispiel: #schnatterfeeling)
- svg aus möglichen Bildformaten rausnehmen
- Bei jeden Post nach der Ortungsfreigabe, darauf hinweisen, dass Ort mit veröffentlicht wird. Manuelles Ein- und Ausschalten wäre gut.

## **About Us**

- Auf About us Footer hängt in der Luft (nicht unten)

## **Suche**

- Bei Post-Suche Hashtag-Icon verwenden, Suche immer noch intuitiv

## **Profilbild**

- svg aus möglichen Bildformaten rausnehmen
- Nach Änderung des Profilbild im Profil: Erneutes Speichern ist unintuitiv

## **Passwort zurücksetzen**

- Nach Absenden des Benutzernamens: Ausdrücklich sagen, dass ein Code an Email gesendet wurde
- Passwort ändern Icon im Profil ist nicht verständlich

## **Registrierung**

- Ungültiger Character-Fehlermeldung geht nicht weg, auch wenn man Zeichen wieder löscht
- Verify Code: Bei Navigieren mit Tab über die Seite wird auch quackquack, und send new code ausgelöst, nicht nur bei klick, genauso bei neuen code senden

## **Sonstiges**

- Abonniert (Startseite) und Abonniert in Profile heißen gleich

Nicht alle Punkte wurden als sinnvoll erachtet und daher nicht alle umgesetzt.

## **Gegenseitige Überprüfung**

Um ein konsistentes Benutzererlebnis sicherzustellen, überprüften Teammitglieder kontinuierlich die Benutzeroberfläche. Wöchentlich fanden Meetings statt, in denen wichtige Punkte und Verbesserungsmöglichkeiten besprochen wurden.

## 1. Code Qualität

**Qualitätsziel:** Sicherstellung eines hohen Standards im Quellcode durch kontinuierliche Überprüfung und Einhaltung von Coding Standards.

### Code Reviews

- **Review-Prozess:** Code darf erst nach einem genehmigten Code Review in den Hauptentwicklungszweig (Merge) integriert werden.
- **Checkliste:** Der Reviewer muss eine Checkliste ausfüllen, die alle relevanten Qualitätskriterien abdeckt, um die Konsistenz und Qualität des Codes zu gewährleisten.
- **Wöchentliche Meetings:** Wöchentliche Besprechungen (meist mittwochs oder montags) zur Festlegung der Review-Verantwortlichkeiten nach einem Rotationsprinzip, um eine gleichmäßige Verteilung der Review-Aufgaben sicherzustellen und unterschiedliche Perspektiven zu integrieren.

### **Checkliste Code Review:**

- Gibt es keine offensichtlichen Logikfehler im Code?
- Sind alle Variablen/Methoden etc. richtig und sprechend benannt (siehe Coding Guidelines -> Benennung)?
- Sind alle relevanten Tests geschrieben ("relevant" siehe Code-Guidelines -> Tests)?
- Sind alle zuvor besprochenen Fehlerfälle abgedeckt? (StatusCodes)
- Gibt es keine Code-Duplikationen (z.B. bereits implementierte Teile in Komponenten)?
- Ist die UI einheitlich (siehe Mockups) und sind die Farben wie im Theme beschrieben verwendet?
- Ist die Codelänge in Ordnung? Wenn nein, wo gibt es Verbesserungspotenzial?
- Ist der Code verständlich geschrieben? Wenn nein, an welchen Stellen ist er zu kompliziert?
- Allgemeine Kommentare:

## Beispiele für ausgefüllte Checklisten und Kommentare in PRs:

### Beispiel 1: ausgefüllte Checkliste nach Review des „[New Profile Feed](#)“-PRs

Nora01K left a comment

Member

...

#Checkliste Code

- ☒ Gibt es keine offensichtliche Logikfehler im Code
  - [nein, siehe Kommentare ] Sind alle Variablen/Methoden etc. richtig und sprechend benannt (siehe Coding Guidelines -> Benennung)
  - [ nicht relevant] Sind alle relevant Tests geschrieben ("relevant" siehe Code-Guidelines -> Tests)
  - [nicht relevant] Sind alle zuvor besprochenen Fehlerfälle abgedeckt? (StatusCodes)
- ☒ Gibt es keine Code-Duplikationen (bsp schon implementierte Teile in Komponenten)
- ☒ UI ist einheitlich (siehe Mockups) und Farben sind verwendet wie in Theme beschrieben
- ☒ Ist die Codelänge in Ordnung? Wenn nein, wo gibt es Verbesserungspotenzial:
- ☒ Ist der Code verständlich geschrieben, wenn Nein an welchen Stellen ist er zu kompliziert:
- Allgemeine Kommentare:

### Beispiel 2: ausgefüllte Checkliste nach Review des „[Feed](#)“-PRs

Nora01K left a comment

Member

...


#Checkliste Code

- [ siehe Kommentar] Gibt es keine offensichtliche Logikfehler im Code
- [ siehe Kommentar] Sind alle Variablen/Methoden etc. richtig und sprechend benannt (siehe Coding Guidelines -> Benennung)
- [ nicht relevant] Sind alle relevant Tests geschrieben ("relevant" siehe Code-Guidelines -> Tests)
- [kommt noch ] Sind alle zuvor besprochenen Fehlerfälle abgedeckt? (StatusCodes)
- ☒ Gibt es keine Code-Duplikationen (bsp schon implementierte Teile in Komponenten)
- ☒ UI ist einheitlich (siehe Mockups) und Farben sind verwendet wie in Theme beschrieben
- ☒ Ist die Codelänge in Ordnung? Wenn nein, wo gibt es Verbesserungspotenzial:
- ☒ Ist der Code verständlich geschrieben, wenn Nein an welchen Stellen ist er zu kompliziert:
- Allgemeine Kommentare:
  - der load more Button funktioniert nicht
  - das png vom Logo kann weg
  - wenn man eingeloggt ist, hat man zwei scroll-Leisten, da musst du mal schauen was du falsch gesetzt hast


### Beispiel 3: Kommentar im „Login“-PR

src/lib/components/Login.svelte Outdated Hide resolved

```
32 + <div class="p-2 h4 text-center font-bold">Log In</div>
33 + <div class="h-[50vh] flex flex-col">
34 +   <form class="flex flex-col justify-around items-center w-full" on:submit={handl
35 +     <div class="flex flex-col w-full pr-3 pl-3 pb-2">
```

 **mabu2807** on Dec 13, 2023 Member ...

pr-3 und pl-3 --> da gibts einen command der das in einem macht oder du zentrierst das ganze einfach



### Beispiel 4: Kommentar im „ForgotPwd“-PR

src/components/modals/ModalForgotPwd.svelte Outdated Hide resolved

```
87 +
88 +   $: tokenSent = tokenSent;
89 +
90 +   async function handleSubmit() {
```

 **mabu2807** on Jun 6 Member ...

in utils auslagern



 Reply...

### Beispiel 5: Kommentar im „Add Likes“-PR

src/lib/PostFunctions.ts Outdated

```
15 +   message: '',
16 +   code: ''
17 + };
18 + serverURL.subscribe((prev_val) => {serverUrl = prev_val});
```

 **Nora01K** on May 17 Member ...

da kommt ein GET hin und kein subscribe, dann ist es einheitlich



 Reply...

## Beschreibung des Testkonzepts

**Ziel:** Sicherstellung, dass alle Eingabefelder korrekt funktionieren und den Qualitätsanforderungen entsprechen.

- **Automatisierte Tests:**
  - **Testframework:** Playwright
  - **Integration in CI/CD-Pipeline:** Die Tests werden automatisch ausgeführt, sobald ein Pull Request (PR) erstellt wird. Falls ein Test fehlschlägt, ist ein Merge nicht möglich.
  - **Testumfang:** Alle Eingabefelder und kritischen Systemkomponenten werden getestet.
- **Manuelle Überprüfung:**
  - **Initiale Tests:** Zunächst wurden die Implementierungen mit dem Mockserver (sofern Testen mit Server Beta/Alpha nicht möglich waren) getestet.
  - **Folgetests:** Nach erfolgreicher initialer Überprüfung erfolgen weitere Tests mit dem Server Alpha und Server Beta.

Unser Testkonzept umfasst sowohl manuelle als auch automatisierte Tests. Dies gewährleistet eine umfassende Überprüfung aller Eingabefelder.

## Fazit

Die Qualitätssicherung für unser Projekt "Schnatter" war umfassend und zielgerichtet, um eine hohe Benutzerfreundlichkeit und Codequalität sicherzustellen. Durch sorgfältige Benutzerstudien und Eyetracking-Analysen konnten wir wertvolle Erkenntnisse gewinnen, die in gezielte Verbesserungen der Anwendung einfließen. Unsere systematischen Code Reviews und die Einhaltung von Coding Standards haben zur Konsistenz und Stabilität des Quellcodes beigetragen.

Die Kombination aus manuellen und automatisierten Tests hat es uns ermöglicht, alle Eingabefelder und kritischen Systemkomponenten gründlich zu überprüfen und sicherzustellen, dass sie den Qualitätsanforderungen entsprechen. Die Integration von Tests in die CI/CD-Pipeline hat den Entwicklungsprozess effizienter und fehlerfreier gemacht.

Dank wöchentlicher Meetings und gegenseitiger Überprüfungen konnte ein konsistentes Benutzererlebnis gewährleistet werden. Diese kontinuierliche Zusammenarbeit und der Austausch im Team waren entscheidend für den Erfolg des Projekts.

Insgesamt hat die systematische und umfassende Qualitätssicherung maßgeblich dazu beigetragen, dass "Schnatter" eine benutzerfreundliche und stabile Anwendung geworden ist. Die gewonnenen Erkenntnisse und Erfahrungen werden uns auch in zukünftigen Projekten von großem Nutzen sein.