

BUBBLE MATEE



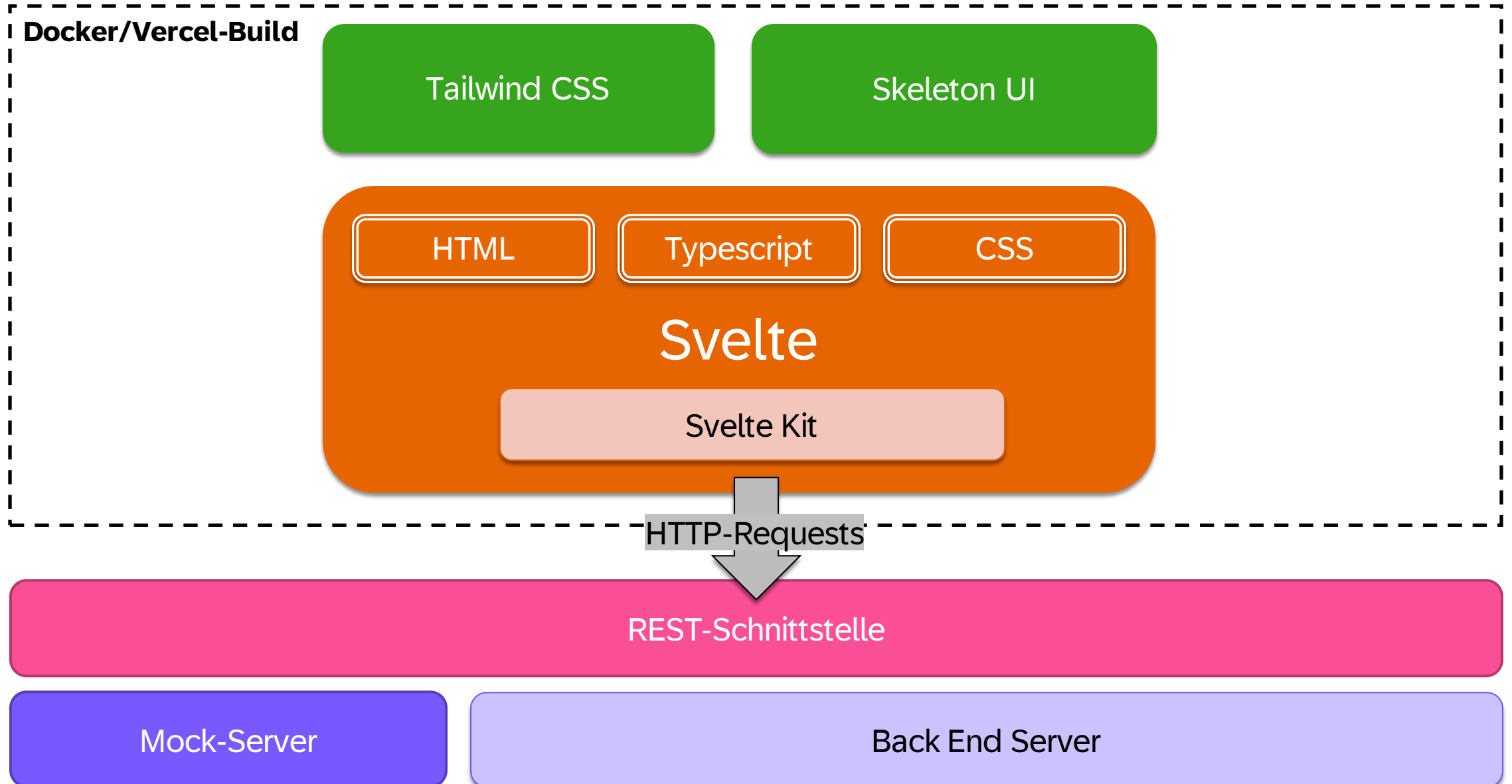
Paul Bahde, Jonas Braun,
Aidan Zimmer, Johanna Deike





Architektur

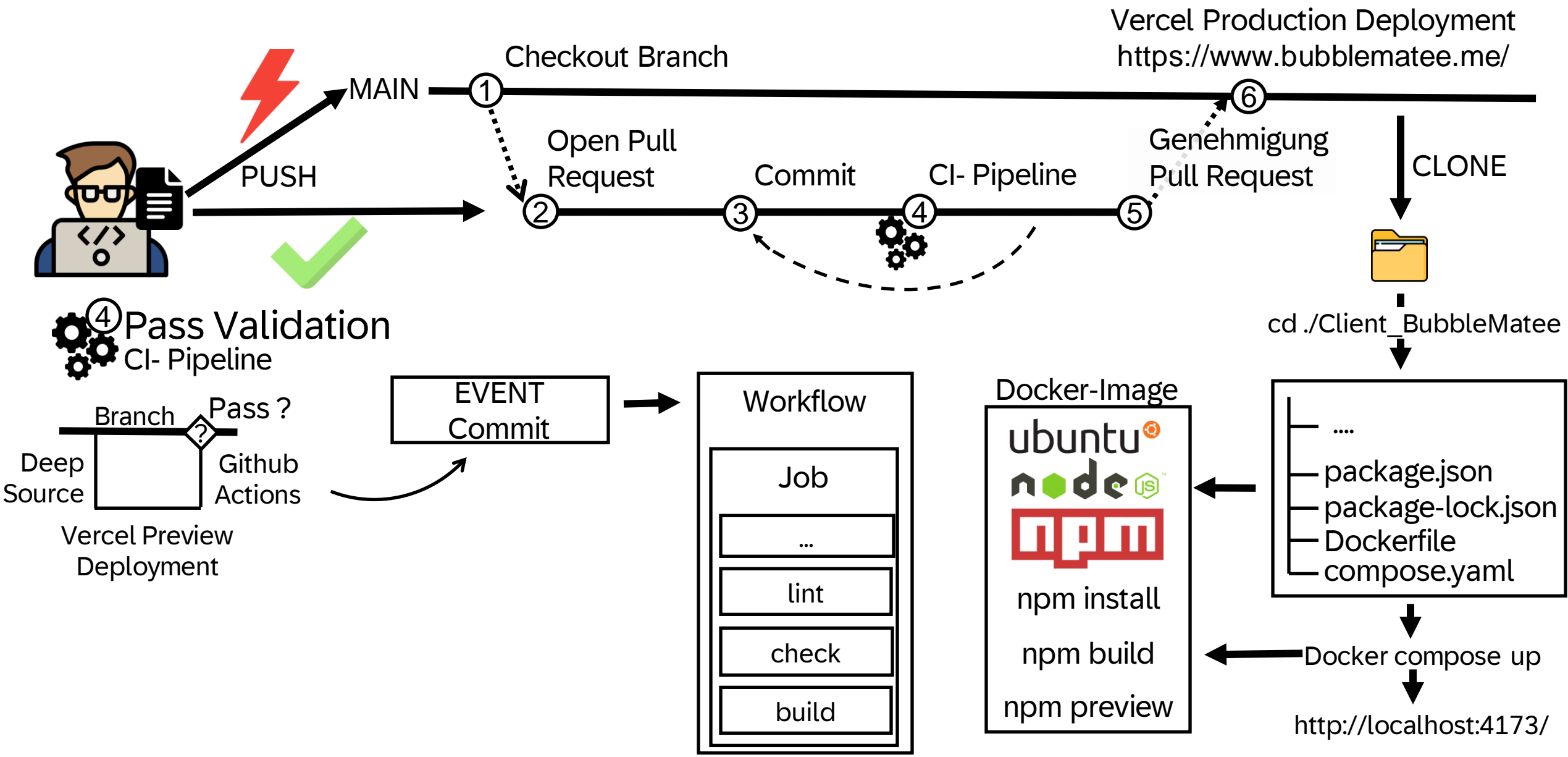
Architektur - Techstack





Build Prozess - CI/CD

Build-Prozess / CI/CD-Pipeline





Qualitätsmaßnahmen

Qualitätsziel: Usability (Benutzerfreundlichkeit)

Benutzbarkeit

Erlernbarkeit

Bedienbarkeit

Checkliste bei PR -> nach den
10 Heuristiken von Nielsen

Checkpoints für die
Überprüfung der Gesamten
Anwendung

Usability-Test

Qualitätsmaßnahmen – Checkliste

BubbleMatee - PR Checkliste

Usability Checklist nach den UI-Prinzipien von Nielsen - PR Checkliste

- ☐ Der Nutzer wird über jede Zustandsänderung der Anwendung informiert (z. B. Beitrag wird gepostet, Post fehlgeschlagen).
- ☐ Die verwendete Sprache und Symbole sind einfach verständlich und entsprechen den Erwartungen der Benutzer.
- ☐ Das Design (Farben/Icons...) stimmen mit denen der restlichen Anwendung überein.
- ☐ Der Benutzer kann alle ausgeführten Aktionen jederzeit einfach beenden und ggf. rückgängig machen.
- ☐ Der Nutzer sieht jederzeit, in welchem Bereich der Anwendung er sich befindet und wie er auf die Hauptseiten zurückgelangt.
- ☐ Der Nutzer sieht jederzeit, in welchem Bereich der Anwendung er sich befindet und wie er auf die Hauptseiten zurückgelangt.
- ☐ Klare Anweisungen und Hinweise werden bereitgestellt, um Benutzerfehler zu minimieren.
- ☐ Validierung und Feedback sind vorhanden, um Benutzer vor Fehlern zu warnen (Überprüfung der Logik von Eingaben etc.).
- ☐ Das Design ist klar, minimalistisch und lenkt nicht von den Hauptaufgaben ab - Unwichtige Informationen werden vermieden.
- ☐ Informationen und Hilfestellungen finden sich an Stellen, an denen sie eventuell benötigt werden. (Besonders bei Benutzereingaben)

Qualitätsmaßnahmen – Checkliste



pbahde commented on Dec 13, 2023

Member



Bei der Registrierung scheint die Überprüfung der Mail und des Nutzernamens kein passendes Feedback zu geben. Vielleicht nochmal überarbeiten, damit Punkt 7 der UI Checkliste erfüllt ist.

 [Uploading MicrosoftTeams-image \(13\).png...](#)




1


Qualitätsmaßnahmen – Checkliste

`src/components/about/karussellProjektbeteiligte.svelte` Outdated Hide resolved

```
fragment slot="summary"  
><h2 class="h2">  
  <span  
    class="bg-gradient-to-br from-red-500 to-yellow-500 bg-clip-text text-transparent box-decoration-clone"
```

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

Das hatten wir ja heute schon erwähnt. Wir benutzen hier bitte die skeleton Farb-Klassen, um mit der Farbpalette konform zu sein. Suche dir gerne [hier](#) was aus



Qualitätsmaßnahmen – Checkliste



pbahde commented on Dec 13, 2023

Member

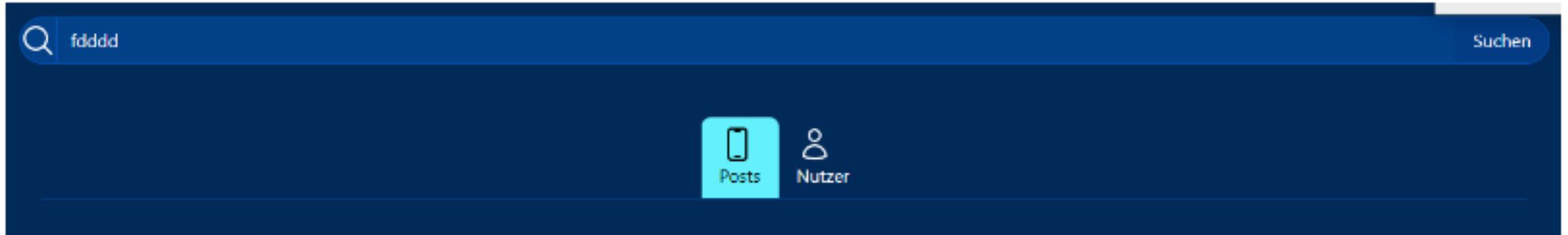


Fehlermeldungen (Code abgelaufen etc.) Error Message in Error-Farbe aus Theme, damit Punkt 3 der UI-checkliste erfüllt ist.

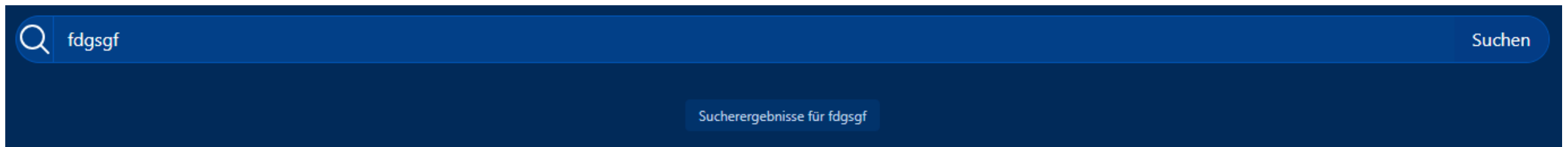


1

Qualitätsmaßnahmen – Checkliste



Ich denke, dass es gut wäre, wenn das aktuelle suchwort irgendwo zu sehen ist. Weil die suche ja erst umspringt, wenn man auf die Lupe klickt. Wenn ich also ein neues Wort eingebe, passt die Anzeige eventuell nicht so gut zusammen.



Qualitätsziel: Usability (Benutzerfreundlichkeit)

Benutzbarkeit

Erlernbarkeit

Bedienbarkeit

Checkliste bei PR -> nach den
10 Heuristiken von Nielsen

Checkpoints für die
Überprüfung der Gesamten
Anwendung

Usability-Test

Qualitätsmaßnahmen – Checkliste



Das Design ist in der gesamten Anwendung einheitlich (Farbschema/ Icon-Design/ Schriftarten...)



Dieselben Icons/Begriffe werden für dieselben Dinge verwendet



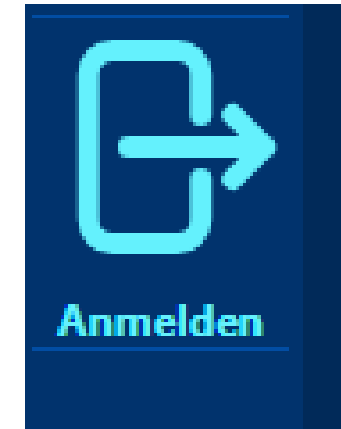
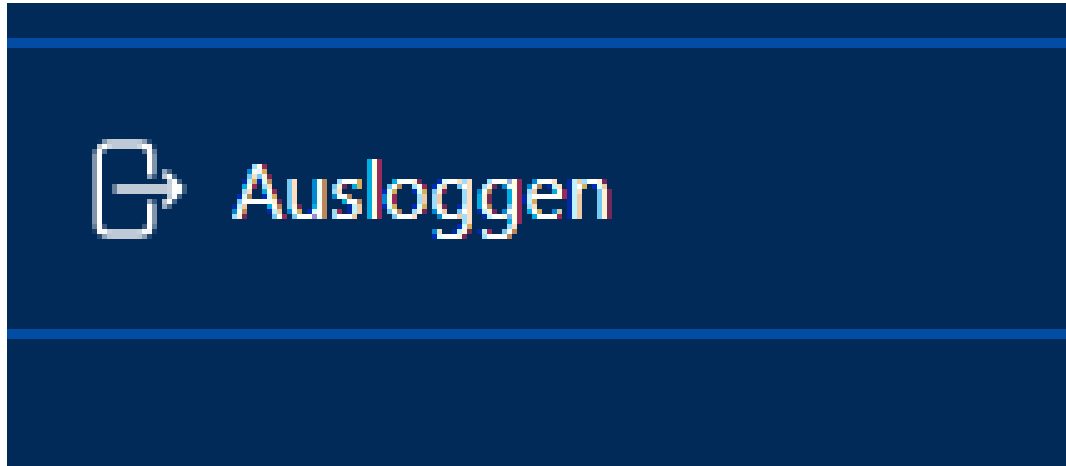
Keine Doppelbelegung von Icons/ Begriffen für verschiedene Dinge



Die Navigation zwischen verschiedenen Teilen ist logisch, intuitiv und kurz

Qualitätsmaßnahmen – Checkliste

1) Übereinstimmende Symbole für Ein- und Ausloggen



Qualitätsmaßnahmen – Checkliste

2) Übereinstimmendes Design für die Überschriften

A dark blue rectangular header with the text "ABOUT US" in a light blue, sans-serif font. Two thin, horizontal light blue lines are positioned at the bottom of the header.

ABOUT US

A dark blue rectangular header with the text "TRENDING POSTS" in a light blue, sans-serif font. Two thin, horizontal light blue lines are positioned at the bottom of the header.

TRENDING POSTS

A dark blue rectangular header with the text "TRENDING POSTS" in a light blue, sans-serif font. Two thin, horizontal light blue lines are positioned at the bottom of the header.

TRENDING POSTS

Qualitätsziel: Usability (Benutzerfreundlichkeit)

Benutzbarkeit

Erlernbarkeit

Bedienbarkeit

Checkliste bei PR -> nach den
10 Heuristiken von Nielsen

Checkpoints für die
Überprüfung der Gesamten
Anwendung

Usability-Test

Usability-Test



Usability-Tests wurden mit dem Stand nach dem letzten Sprint durchgeführt



5 verschiedene Aufgaben, die die gesamte damalige Funktionalität abdecken



4 einzelne Test – Handy/Desktop LightMode/
DarkMode

Usability-Test

User Test 2: Registrierungsprozess und Profildatenänderungen

Aufgabenbeschreibung:

Startpunkt ist die Homepage.

Protokoll:

1. Gehe auf die Login-Seite und erstelle dir einen neuen Account und logge dich ein.
(Bestätige deine Email Adresse) 1 min 34 Sekunden
2. Gehe in deine Einstellungen und ändere deinen Nicknamen und deinen Status. 18 Sekunden
3. Ändere in deinen Einstellungen dein Passwort. 26 Sekunden
4. Logge dich aus. 5 Sekunden

Usability-Test

Testumgebung:

1. Verwendetes Gerät: PC-Monitor 1920x1080px
2. Dark- oder Light-Mode: Dark Mode

Quantitative Kriterien:

Kriterium	Erwarteter Wert	Tats. Wert
Gesamtzeit	Max. 180 Sekunden	143 Sekunden
Benötigte Button-Klicks	23	27
Aufgabe geschafft?	Ja	ja

Usability-Test

Beobachtungen:

1. An welchen Stellen braucht der Nutzende länger als erwartet/ wo zögert er?
 - a. |Registrieren der Nutzerdaten -> Erkennen, bei welchen Eingaben Fehler vorliegen/ wo die Fehler vorliegen (z.B -> Nutzernamen)
2. An welcher Stelle wählt der Nutzende einen ungewöhnlichen Navigationsweg?
 - a. Nichts aufgefallen
3. An welchen Stellen sind mir Verbesserungsvorschläge eingefallen?
 - a. Nutzname → Nutzernamen (bei Nutzerdaten eingeben)
 - b. Rückmeldung/ Toast das man ausgeloggt ist

Usability-Test

Fragen für die Testperson:

1. Auf einer Skala von 1 bis 10, wie intuitiv einfach war der gesamte Prozess?

1. 8

2. An welchen Stellen warst du dir unsicher? Wo hättest du mehr Unterstützung gebraucht?

1. Was ist falsch ist bei Nutzerdaten?

3. Welche allgemeinen Verbesserungsvorschläge hast du?

1. Siehe oben

1) Interaktion als unangemeldeter Nutzer

1. Light- / DarkMode Symbol besser/ anders kennzeichnen
2. Globlen Feed besser kennzeichnen / anders benennen auf Desktop

2) Registrierungsprozess und Profildatenänderung

1. Bessere Rückmeldung bei der Eingabe der Registrierungsdaten, wenn Fehler auftreten (z.B. warum Nutzernamen nicht erlaubt)
2. Rückmeldung, wenn erfolgreich ausgeloggt
3. Rechtschreibfehler "Nutzname" bei Registrierung
4. Woher kommt der Code? -> Toast, dass per Email gesendet

3) User im Globalen Feed

1. Unterschied zwischen Privatem und Globalen Feed kennzeichnen
2. Globaler Feed und Suche Trennen

4) Suchen, Folgen und Entfolgen

1. Kennzeichnen der Personensuche gegenüber der Postsuche. (Idee -> bei # automatisch zur Postsuche sonst Nutzersuche)

5) Post erstellen, suchen und löschen

1. Evtl. Posten von „meinProfil“ aus?
2. Eigene Posts von überall löschen können
3. Kennzeichnen, dass nach Hashtags gesucht wird/ kennzeichnen ob das #-Symbol miteingegeben werden muss.

Qualitätsziel: Wartbarkeit (Maintainability)

Nutzung und manuelle Überprüfung von Best Practices

Automatische Überprüfung in der Pipeline

Qualitätsmaßnahmen – Checkliste

Code Checkliste - Developer

Überprüfe alle unten stehenden Punkte gewissenhaft

- ☐ Ich habe konsequent Variablen, Rückgabewerte von Funktionen, Arrays usw. typisiert.
- ☐ Wo notwendig, habe ich spezifische Interfaces genutzt, um eine klarere Typisierung zu ermöglichen.
- ☐ Ich habe "let" anstelle von "var" für die Deklaration von lokalen Variablen verwendet.
- ☐ Ich habe die Standard Naming Conventions eingehalten (camelCase für Variablen, Funktionen und Dateien - PascalCase für Interfaces, Typen, Enums und Klassen)
- ☐ Ich habe aussagekräftige Bezeichner für Funktionen und Variablen verwendet, um die Verständlichkeit des Codes zu verbessern (z. B. `let customerId` statt `let numA`).
- ☐ Ich habe Files, die mehr als 200 Code Zeilen haben in wiederverwendbare Komponenten aufgeteilt
- ☐ Ich habe unnötige Code-Duplikation vermieden und Code, der an verschiedenen Stellen benötigt wird, in Funktionen oder Komponenten ausgelagert.
- ☐ Ich habe meinen Code angemessen kommentiert, insbesondere bei komplexen Logiken oder unklaren Stellen.
- ☐ Wo immer möglich, habe ich Async/Await statt Promises mit `.then()` für asynchrone Operationen verwendet, um den Code lesbarer zu machen.

Qualitätsmaßnahmen – Checkliste

Code Checkliste - Reviewer

Überprüfe alle unten stehenden Punkte gewissenhaft

- ☐ Es wurden konsequent Variablen, Rückgabewerte von Funktionen, Arrays usw. typisiert.
- ☐ Es wurden Interfaces genutzt, um eine klarere Typisierung zu ermöglichen, wo dies notwendig war.
- ☐ Es wurde "let" anstelle von "var" für die Deklaration lokaler Variablen verwendet.
- ☐ Es wurden die Standard Naming Conventions eingehalten (camelCase für Variablen, Funktionen und Dateien; PascalCase für Interfaces, Typen, Enums und Klassen).
- ☐ Es wurden aussagekräftige Bezeichner für Funktionen, Variablen und Komponenten verwendet, um die Verständlichkeit des Codes zu verbessern (z. B. `let customerId` statt `let numA`).
- ☐ Es wurden Files, die mehr als 200 Codezeilen enthalten, in wiederverwendbare Komponenten aufgeteilt.
- ☐ Es wurde unnötige Code-Duplikation vermieden, indem Code, der an verschiedenen Stellen benötigt wird, in Funktionen oder Komponenten ausgelagert wurde.
- ☐ Der Code wurde angemessen kommentiert, insbesondere bei komplexen Logiken oder unklaren Stellen.
- ☐ Wo immer möglich, wurde Async/Await anstelle von Promises mit `.then()` für asynchrone Operationen verwendet, um den Code lesbarer zu gestalten.

Qualitätsmaßnahmen – Checkliste

src/components/posts/posts.svelte Outdated ⚙️ Hide resolved

```
1 + <!-- List component that contains multiple posts. -->
2 + <script lang="ts">
3 +     //TODO: Anpassen an Interface
4 +     export let data: [];
```

 **johannadke** on Dec 20, 2023 Member ...

Typisierung: Kann man da vielleicht in domain ein interface anlegen für einen Post und es dann wie vorgeschlagen typisieren?

Suggested change

-	export let data: [];
+	export let data: Array<Post>;

Commit suggestion ▾



Qualitätsmaßnahmen – Checkliste

Ich würde das hier nicht mit plain html machen, sondern mit [svelte assignments](#)
Beispielsweise:

```
let moreClicked = false
function changeMoreClicked() {
  moreClicked = !moreClicked ;
}
```



Dann kann man das in den Button hauen und mithilfe von `moreClicked ? 'Weniger' : Mehr` den Button umbenennen und du kannst dann auch die Redundanz mit den Kommentaren verringern mit `moreClicked ? 'comment' : shortComment`

Qualitätsmaßnahmen – Checkliste

src/components/postComponent/components/fileInput/filedropzone.svelte Outdated ⚙️ Hide resolved

18 +

19 +

20 +

21 +

```
function onChangeHandler(e: Event) {  
    console.log(localFiles.length !== 1);  
    if ($files === null || $files.length !== 1) {  
        alert('Please upload only one file.');
```



pbahde on Dec 16, 2023 Member Author ...

Ist dieser Fall überhaupt möglich? Hab ausprobiert mehrere Files hochzuladen und es ist immer nichts passiert.



Qualitätsmaßnahmen – Checkliste



BraunJonas commented 3 weeks ago • edited ▾

Member



es wäre denke ich auch ganz gut noch mehr Kommentare in den Code einzubauen (besonders im HTML Code bei posts/comments...)



Qualitätsmaßnahmen – Checkliste



BraunJonas commented 3 weeks ago

Member



Als Idee im Post Header die Zeitemrechnung nochmals auslagern, sodass sie in anderen Komponenten gegebenenfalls wiederverwendet werden kann



zimmerai 5 days ago

Member



Das Popup kann ne eigene Komponente sein



Qualitätsziel: Wartbarkeit (Maintainability)

Nutzung und manuelle Überprüfung von Best Practices

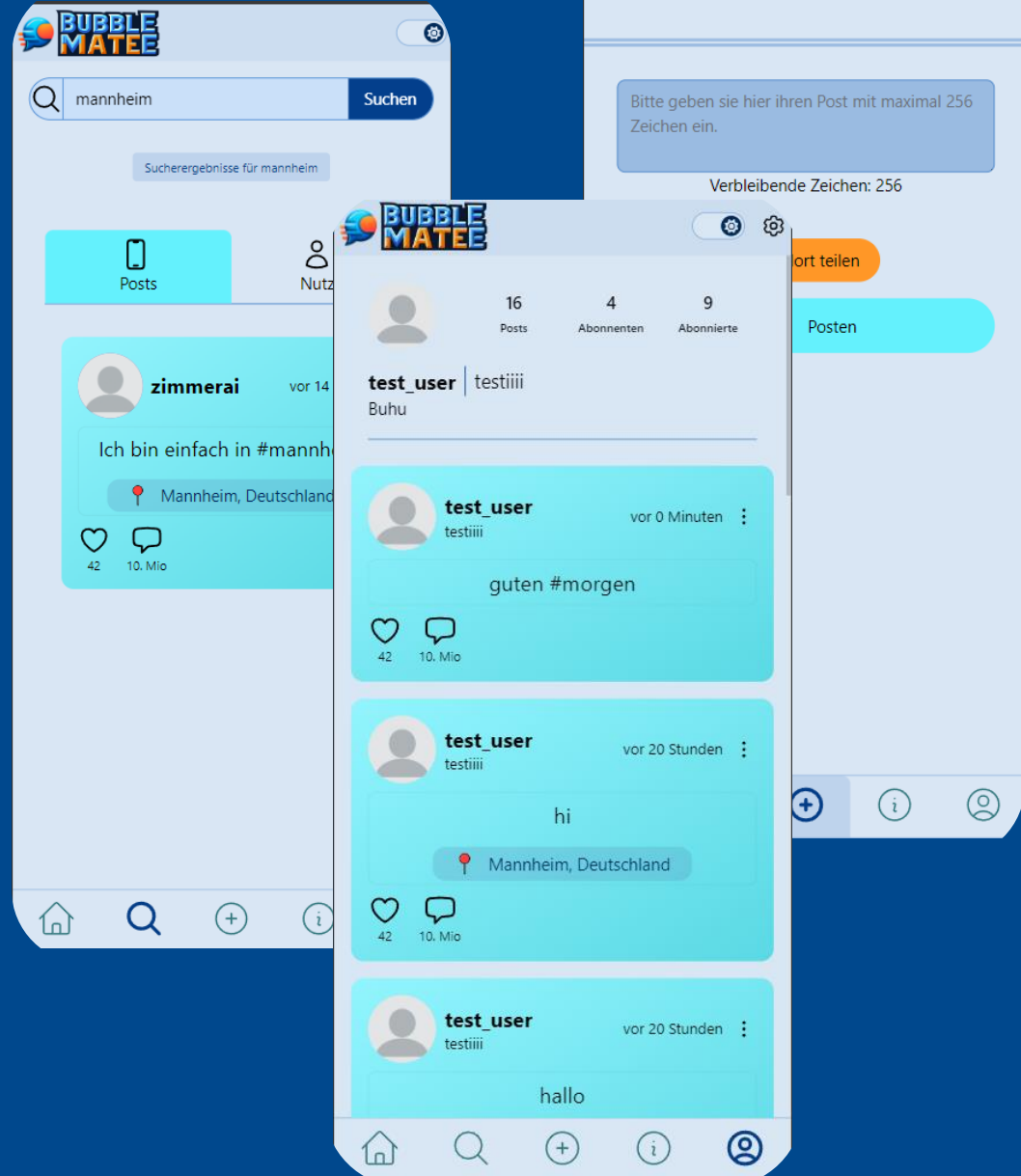
Automatische Überprüfung in der Pipeline

Automatische Überprüfung in der Pipeline





Demo



Danke!

Fragen?



**BUBBLE
MATEE**