BUBBLE MATEE

Paul Bahde, Jonas Braun,
Aidan Zimmer, Johanna Deike

Qualitätsmaßnahmen

Qualtitätsmaßnahmen

Qualitätsziel: Usability (Benutzerfreundlichkeit)

Benutzbarkeit	Erlernbarkeit	Bedienbarkeit
Checkliste bei PR -> nach den 10 Heuristiken von Nielsen	Checkpoints für die Überprüfung der Gesamten Anwendung	Usability-Test

Qualtitätsmaßnahmen

Checkliste bei PR -> nach den 10 Heuristiken von Nielsen

Was ist die Maßnahme?

Abarbeitung einer Checkliste (orientiert an Nielsen)

Warum ist die Maßnahme wichtig?

Sicherstellen, dass alle Änderungen zu einer benutzerfreundlichen Anwendung führen

Wann wird die Maßnahme durchgeführt?

Für jeden PR, bevor er auf den main-branch gemerged wird.

Von wem wird die Maßnahme durchgeführt?

Von einem (möglichst) unbeteiligten Mitglied der Gruppe (wie Code Review)

Was passiert, wenn **Mängel** festgestellt werden?

Merge auf den main-branch wird blockiert.

Qualtitätsmaßnahmen – 10 UI-Heuristiken nach Nielsen

1. Sichtbarkeit des Systemstatus
2. Übereinstimmung zwischen System und realer Welt
3. Benutzerkontrolle und Freiheit
4. Konsistenz und Standards
5. Fehlervermeidung
6. Anerkennung anstelle von Erinnerung
7. Flexibilität und Effizienz der Nutzung
8. Ästhetik und minimalistisches Design
9. Hilfe und Dokumentation
10. Fehlerdiagnose und Wiederherstellung

Qualtitätsmaßnahmen – 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...) sind 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)

Qualtitätsmaßnahmen

Qualitätsziel: Usability (Benutzerfreundlichkeit)

Benutzbarkeit	Erlernbarkeit	Bedienbarkeit
Checkliste bei PR -> nach den 10 Heuristiken von Nielsen	Checkpoints für die Überprüfung der Gesamten Anwendung	Usability-Test

Qualtitätsmaßnahmen

Checkpoints für die Überprüfung der Gesamten Anwendung

Was ist die Maßnahme?

An festen „Checkpoints“ werden Aspekte untersucht, die die ganze Anwendung betreffen.

Warum ist die Maßnahme wichtig?

Nicht alle Aspekte – einheitliches Design etc. können an einzelnen PRs untersucht und behoben werden.

Wann wird die Maßnahme durchgeführt?

Am Ende jedes Sprints.

Von wem wird die Maßnahme durchgeführt?

Gemeinsam im Team (mind. 2 Personen).

Was passiert, wenn **Mängel** festgestellt werden?

Mängel werden im Laufe des nächsten Sprints behoben.

Qualtitätsmaßnahmen – 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

Qualtitätsmaßnahmen

Qualitätsziel: Usability (Benutzerfreundlichkeit)

Benutzbarkeit	Erlernbarkeit	Bedienbarkeit
Checkliste bei PR -> nach den 10 Heuristiken von Nielsen	Checkpoints für die Überprüfung der Gesamten Anwendung	Usability-Test

Qualtitätsmaßnahmen

Usability-Test

Was ist die Maßnahme?

Es werden regelmäßig Usability-Tests mit mehreren Teilnehmern durchgeführt.

Warum ist die Maßnahme wichtig?

Liefern Feedback und Erfahrungen von Außenstehenden.

Wann wird die Maßnahme durchgeführt?

Einen in der 1. Phase + einen Mitte der 2ten

Von wem wird die Maßnahme durchgeführt?

Mit je 4 projektfremden Teilnehmern (jeder mit einem) **- Jeweils verschiedene**

Was passiert, wenn **Mängel** festgestellt werden?

Mängel werden im Laufe des nächsten Sprints behoben / Verbesserungsvorschläge angepasst.

Qualtitätsmaßnahmen – Usability-Test

Beispielaufgabe

Startpunkt: Home-Seite

Aufgabe: Logge dich mit deinem Account ein und ändere deinen Nickname

Erwartete Zeit: max. 1:30 min

Erwartete Anzahl Button Klicks: max. 6

Aufgaben werden jeweils am Anfang eines Sprints neu definiert/ angepasst

Qualtitätsmaßnahmen – Usability-Test

Protokoll

1) Quantitative Auswertung

1. Aufgabe geschafft?
2. Zeit
3. Benötigte Button Klicks

Qualtitätsmaßnahmen – Usability-Test

Protokoll

1) Quantitative Auswertung

2) Fragebogen

1. Auf einer Skala von 1 bis 10, wie intuitiv/ einfach war der Vorgang?
2. An welcher Stelle warst du dir am unsichersten, wie du weitermachen musst? – Wo willst du mehr Unterstützung haben?
3. Welche allgemeinen Verbesserungsvorschläge hast du?

Qualtitätsmaßnahmen – Usability-Test

Protokoll

1) Quantitative Auswertung

2) Fragebogen

3) Beobachtung

1. Stellen an denen der Benutzer besonders lange braucht/ zögert:
2. Fehler des Nutzers/ Wahl eines anderen Wegs, als der erwartete:

Qualtitätsmaßnahmen

Qualitätsziel: Wartbarkeit (Maintainability)

Nutzung und manuelle Überprüfung von Best Practices

Automatische Überprüfung in der Pipeline

Qualtitätsmaßnahmen

Nutzung und manuelle Überprüfung von Best Practices

Was ist die Maßnahme?

Es werden Best Practices definiert zum coden und reviewen

Warum ist die Maßnahme wichtig?

Koordinierte Überprüfung des Codes

Wann wird die Maßnahme durchgeführt?

Während der Entwicklung jedes Feature Branches

Von wem wird die Maßnahme durchgeführt?

- Vom Entwickler und Reviewer

Was passiert, wenn **Mängel** festgestellt werden?

Direkte Behebung im Feature Branch

Qualtitätsmaßnahmen

Nutzung und manuelle Überprüfung von Best Practices

Typisierung und Deklaration

Namenskonventionen

Wiederverwendbarkeit

Verständlichkeit

Qualtitätsmaßnahmen – 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.

Qualtitätsmaßnahmen – 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.

Qualtitätsmaßnahmen

Automatische Überprüfung in der Pipeline

Was ist die Maßnahme?

Unterstützende Code Überprüfungen in der CI-Pipline

Warum ist die Maßnahme wichtig?

Um die manuelle Überprüfung zu ergänzen und dadurch die Qualität erhöhen

Wann wird die Maßnahme durchgeführt?

Bei Erstellung eine Pull-Request und jedem weiteren Commit

Von wem wird die Maßnahme durchgeführt?

- Automatischer Job

Was passiert, wenn **Mängel** festgestellt werden?

Direkte Behebung im Feature Branch

Qualtitätsmaßnahmen

Automatische Überprüfung in der Pipeline

EsLint

- Stilregeln
- Best Practices
- Identifizierung von potenziellen Bugs oder unsicherem Code

Prettier

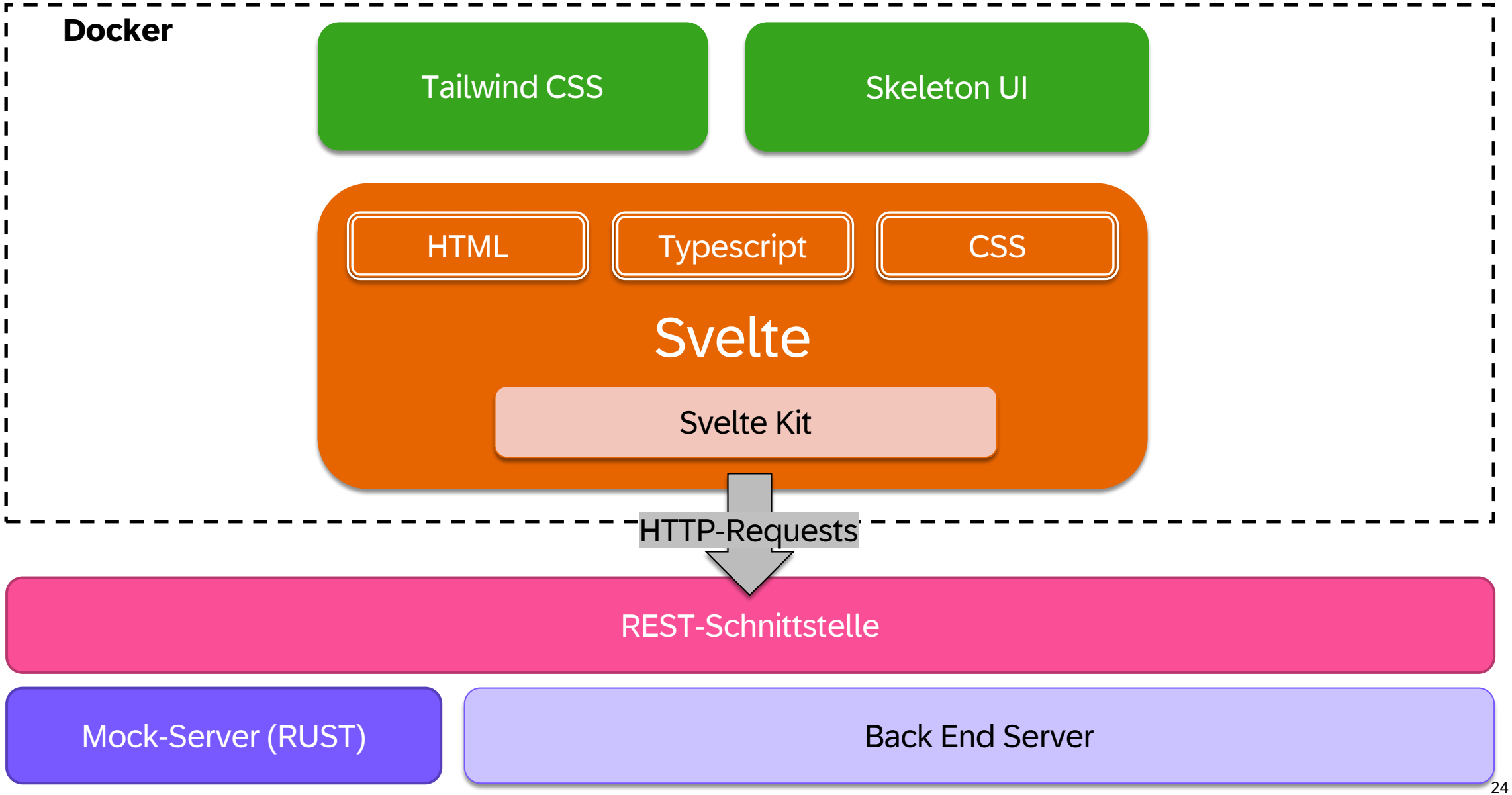
- Konsistenz der Codeformatierung.

DeepSource

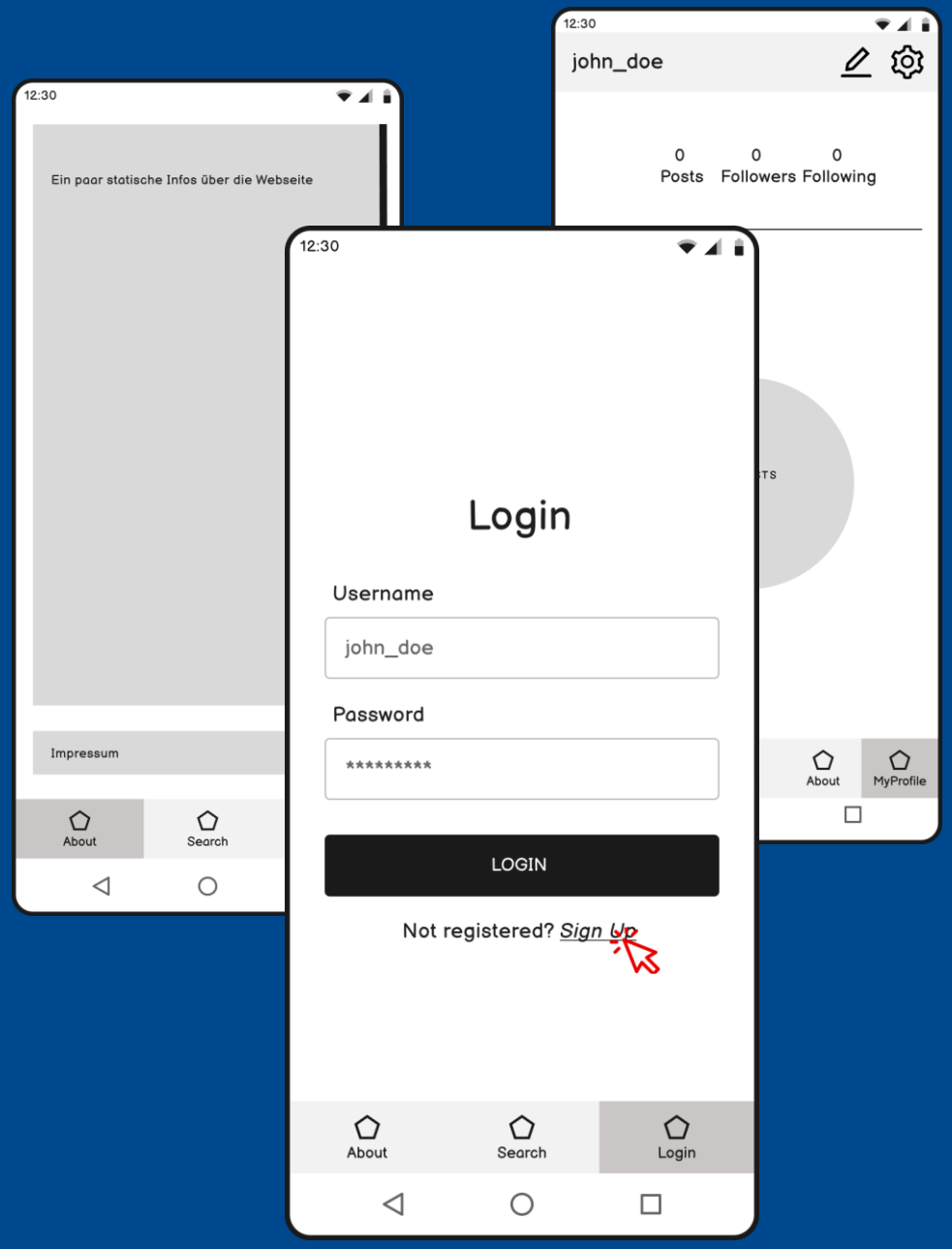
- Code-Stilprüfungen
- Identifikation von Code-Smells und Verbesserungsmöglichkeiten.
- tiefere Code-Analyse

Architektur & Mockups

Architektur - Techstack

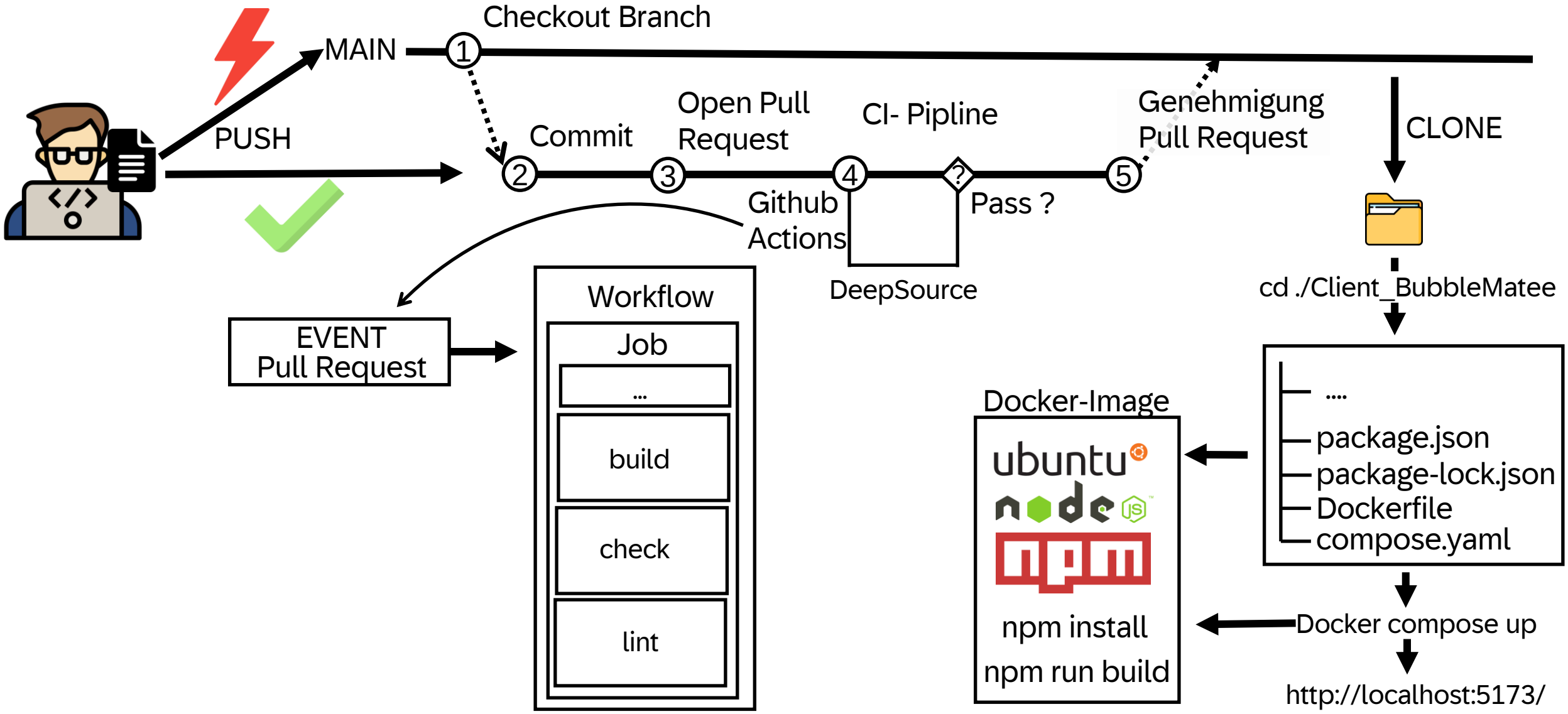


Mockups


12:30
Ein paar statische Infos über die Webseite
Impressum
About
Search
12:30
john_doe
0
Posts
0
Followers
0
Following
About
MyProfile
12:30
Login
Username
john_doe
Password
****
LOGIN
Not registered? Sign Up
About
Search
Login

Build Prozess / CICD

Build-Prozess / CI-Pipeline



Danke!

Fragen?

BUBBLE MATE