

AnoAsked

**Projektarbeit für den Fachbereich Wirtschaftsinformatik von Masooglu Simon**

**2023/2024**

Inhaltsangabe

[1 Eidesstattliche Erklärung des Autors ii](#_Toc167059011)

[2 Kurzbeschreibung iii](#_Toc167059012)

[2.1 Deutsch iii](#_Toc167059013)

[2.2 Englisch iii](#_Toc167059014)

[3 Timetable und initiale Ziele iv](#_Toc167059015)

[3.1 Timetable iv](#_Toc167059016)

[3.2 Initiale Ziele iv](#_Toc167059017)

[4 Anforderungsmanagement v](#_Toc167059018)

[4.1 Use Case-Diagramm v](#_Toc167059019)

[4.2 Aktoren Analyse v](#_Toc167059020)

[4.3 Anwendungsfälle vi](#_Toc167059021)

[5 Supplementary Requirements x](#_Toc167059022)

[5.1 Usability x](#_Toc167059023)

[5.2 Reliability x](#_Toc167059024)

[5.3 Performance x](#_Toc167059025)

[5.4 Supportability x](#_Toc167059026)

[5.5 Interface x](#_Toc167059027)

[5.6 Legal x](#_Toc167059028)

[6 Domänenmodell xi](#_Toc167059029)

[7 Funktionalitäts-Erklärung (Essenzielle Auszüge) xii](#_Toc167059030)

[7.1 Technologien xii](#_Toc167059031)

[7.2 Projektstruktur xiii](#_Toc167059032)

[7.3 Projektauszüge xiii](#_Toc167059033)

[7.4 Testen der Grundfunktionen (Bewertung) xiv](#_Toc167059034)

[8 Ausblick xiv](#_Toc167059035)

[9 Glossar xiv](#_Toc167059036)

[10 Quellverzeichnis xiv](#_Toc167059037)

# Eidesstattliche Erklärung des Autors

Hiermit erkläre ich an Eides statt, dass ich die vorliegende Arbeit und nur mit den angeführten Literaturhinweisen selbstständig und ohne Mitwirkung Dritter verfasst habe.  
Die Projektarbeit beinhaltet einen Umfang von 2787 Wörtern.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Masooglu Simon

# Kurzbeschreibung

## Deutsch

AnoAsked ist eine social Media Plattform, die mit dem Schwerpunkt auf Anonymität versucht, die Veröffentlichung von Beiträgen, ohne Einschränkung der eigenen Privatsphäre, zu ermöglichen. Durch Funktionen wie Kommentieren und liken von Beträgen können Diskussionen geführt werden, wobei alle Interaktionen anonym geführt werden können. Zusätzlich können sich Benutzer verifizieren, um ihre Identität zu beweisen.

## Englisch

AnoAsked is a social media platform that, with a focus on anonymity, tries to enable the publication of posts without restricting one's own privacy. Discussions can be conducted through features such as commenting and liking posts, and all interactions can be carried out anonymously. Additionally, users can verify themselves to prove their identity.

# Timetable und initiale Ziele

## Timetable

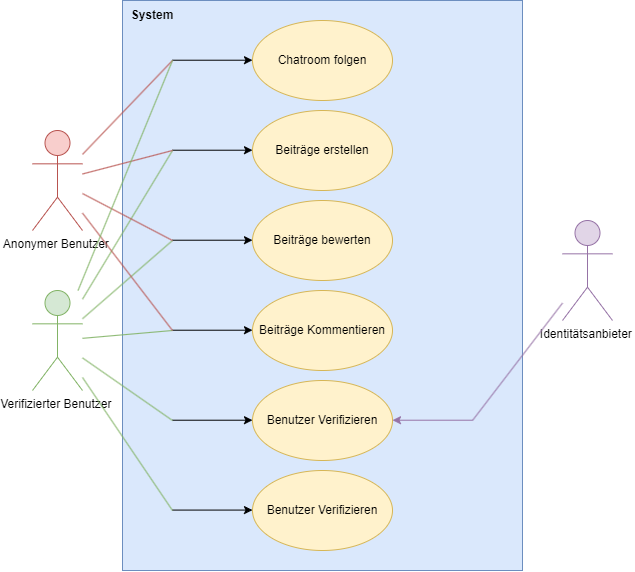
|  |  |  |
| --- | --- | --- |
| Sprint | Datum von/bis | Arbeitsauftrag |
| 1 | 10.01.2024  12.01.2024 | Erstellen der Github Organisation,  Erstellen eines Figma Prototyps |
| 2 | 13.01.2024  21.02.2024 | Festlegen Projektname und Logo,  Abgabe des Anforderungsmanagements |
| 3 | 22.02.2024  06.03.2024 | Erstellen der SvelteKit Applikation,  Erstellen der Landing Seite |
| 4 | 07.03.2024  12.03.2024 | Error Seite erstellen,  Login und Register Seiten erstellen |
| 5 | 13.03.2024  03.04.2024 | Intigrieren den GunJS backends,  Erstellen des postinstall scripts um gunJS Fehler zu beheben |
| 6 | 04.04.2024  16.04.2024 | Erstellen des Verifikations Servers,  Hinzufügen der Login, Registrier und Einstellungs Logik |
| 7 | 17.04.2024  29.04.2024 | Erstellen des Seiten Layouts mit Raumnavigation,  Erstellen der Raum Seite mit Nachrichten Eingabebox |
| 8 | 30.04.2024  07.05.2024 | Hinzufügen der Beitrag Logik,  Hinzufügen der Logik für Likes und Dislikes |
| 9 | 08.05.2024  15.05.2024 | Hinzufügen der Logik für Kommentare,  Hinzufügen der Logik für Tags, Verschlüsselung und Uploads |
| 10 | 16.05.2024  22.05.2024 | Fertigstellen des Projektes  Fertigstellen der Dokumentation |

## Initiale Ziele

* Festlegen des Designs und Erstellen eines Prototyps
* Erstellen der Svelte Frontend Applikation
* Erstellen der Frontend Komponenten in Svelte
* Erstellen des gunJS backend Servers
* Implementierung der Frontend Funktionalität
* Dokumentation der gesamten Applikation

# Anforderungsmanagement

## Use Case-Diagramm



## Aktoren Analyse

* Ein anonymer Benutzer (Primary Actor), braucht kein Passwort, um sich zu registrieren, und kann mit einem Link wieder auf seinen anonymen Benutzer zugreifen.
* Ein verifizierter Benutzer (Primary Actor), muss sich verifizieren und ein Konto erstellen, er kann dann mit seinem Drittanbieter Benutzer wieder auf sein verifiziertes Konto zugreifen. Zusätzlich kann er sich selbst Tags hinzufügen.
* Identitätsanbieter (Supporting Actor), bietet eine Möglichkeit sich über einen z.B. Github-Account zu verifizieren, um seine Identität zu bestätigen.

## Anwendungsfälle

(Die Stories der Anwendungsfälle werden durch die Perspektive eines Benutzers wiedergegeben)

### Anwendungsfall Beitrag erstellen (casual)

#### Diagramm

Ein Bild, das Text, Diagramm, Plan, parallel enthält.

Automatisch generierte Beschreibung

#### Story

Ein Benutzer ist gerade auf einer Party und redet mit seinen Freunden über Meinungsfreiheit. Alle seine Freunde sind der gleichen Meinung, doch er will noch andere Meinungen kennenlernen. Somit will er einen Beitrag zu diesem Thema im Internet erstellen, doch will nicht seine eigene Privatsphäre gefährden. Ihm fällt ein das AnoAsked seine Anonymität bewahren kann und entscheidet sich auf die AnoAsked Webseite zu gehen.

Nachdem der Benutzer die Webseite startet, wird er von 2 Optionen auf der Landing Page begrüßt. Er hat die Option entweder Anonym fortzufahren oder sich zu verifizieren. Da der Benutzer anonym bleiben will, wählt er die Option anonym fortzufahren. Nun kommt er auf die Homepage der Webseite bei denen ihm Chatrooms angezeigt werden, die gerade im Trend sind. Da er aber selbst einen Beitrag erstellen will navigiert er in den r/discussions Chatroom. Hier klickt er auf „Neuer Beitrag“ und stellt seine Frage da der Benutzer aus Österreich kommt stellt er die Standort-Sichtbarkeit auf Österreichweit ein. Zusätzlich hat er die Option bestimmte „Tags“ hinzuzufügen, die signalisieren an welche Benutzer dieser Beitrag gerichtet ist. Hier fügt er den Tag „Politiker“ hinzu. Nun klickt er auf „Erstellen“ um den Beitrag im r/discussions Chatroom zu erstellen.

### Anwendungsfall Beitrag bewerten (brief)

#### Diagramm

Ein Bild, das Text, Screenshot, Diagramm, Reihe enthält.

Automatisch generierte Beschreibung

#### Story

Nach einiger Zeit sieht der Benutzer das jemand unter seinem Beitrag einen Kommentar erstellt hat. Er findet den Kommentar, sehr interessant und will ihn Bewerten, der Benutzer hat jetzt die Option den Beitrag zu „Liken“ oder zu „Disliken“, da er den Beitrag mag klickt er auf „Like“. Diese Bewertung wird nun für alle anderen Benutzer auch angezeigt.

### Anwendungsfall Beitrag kommentieren (brief)

#### Diagramm

Ein Bild, das Text, Diagramm, Screenshot, Reihe enthält.

Automatisch generierte Beschreibung

#### Story

Ein anderer Benutzer hat auch ein Kommentar unter dem Beitrag erstellt. Der Benutzer findet aber, dass der Verfasser dieses Kommentares seinen originalen Beitrag nicht richtig verstanden hatte und will ihm dies auch klarstellen. Um das zu machen, drückt er auf den Knopf „Kommentieren“. Der Benutzer füllt wie beim Beitrag die gewünschten Felder aus und drückt, nachdem er seine Meinung nochmal verdeutlicht, hat auf „Erstellen“. Somit können alle anderen den Kommentar sehen und die gleichen Funktionen wie bei einem Beitrag verwenden, um mit dem Kommentar zu interagieren.

### Anwendungsfall Chatroom folgen (brief)

#### Ein Bild, das Text, Screenshot, Schrift, Reihe enthält. Automatisch generierte BeschreibungDiagramm

#### Story

Nach einer lehrreichen Diskussion im Chatroom r/discussions entscheidet sich der Benutzer dem Chatroom zu folgen, um immer auf dem neusten Stand der heißen Beiträge in diesem Chatroom zu bleiben. Um das zu machen, klickt er auf „Folgen“ und bekommt somit auf der Homepage die Neuigkeiten angezeigt.

### Anwendungsfall Benutzer Verifizieren (brief)

#### Diagramm

Ein Bild, das Text, Schrift, Screenshot, Reihe enthält.

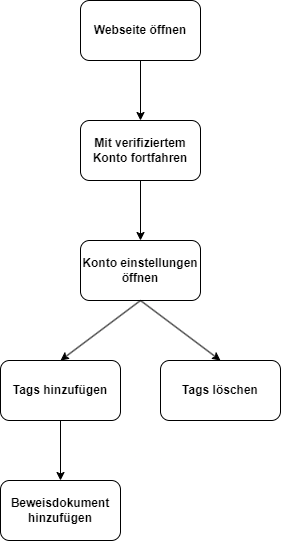
Automatisch generierte Beschreibung

#### Story

Nach einem langen hin und her von Argumenten entscheidet sich der Benutzer ein verifiziertes Konto zu verwenden, um seine Aussagen zu stärken. Er geht zurück auf die Landing Page und wählt die Option „Mit GitHub fortfahren“ aus, somit meldet er sich mit dem gewünschten Konto an und wird zur Homepage von AnoAsked zurückgeleitet. Nun ist sein Konto ein verifiziertes Konto und wird für alle anderen Benutzer auch als verifiziertes Konto angezeigt.

### Anwendungsfall Verifizierter Benutzer Tags hinzufügen (brief)

#### Diagramm



#### Story

Der Benutzer ist ein Politikwissenschaftler und hat ein Studium in diesem Bereich absolviert. Um weiterhin zu zeigen, dass er Ahnung vom Gesprächsthema hat, entscheidet er sich, ein Konto-Tag hinzuzufügen. Dafür geht er auf die Kontoeinstellungen in dem er auf sein Profilbild klickt. Nun hat er die Option Tags hinzuzufügen oder zu löschen. Er drückt auf den „Tag hinzufügen“ Knopf, um ein neues Tag hinzuzufügen. Nun muss er den Namen des Tags definieren und ein Bild seiner Urkunde hochladen. Um den Tag hinzuzufügen, drückt er auf „Tag erstellen“. Jetzt können alle sehen das er ein verifizierter Politikwissenschaftler ist, sie Hochgeladene Urkunde ist für jeden sichtbar.

# Supplementary Requirements

## Usability

* Einfache Handhabung
* Große und eindeutige Symbole
* Gut leserliche Schriftart
* Futuristisches Design
* Eindeutige & Erkennbare Bedienfeldeinteilung

## Reliability

* Dezentralisiertes backend, um vor Ausfällen zu schützen
* Zusätzliche Server-Nodes um vor backend ausfällen zu schützen
* Frontend/UI sollte 24/7 Zugriff Bieten

## Performance

* Aktionen müssen innerhalb von 1 Sekunde durchführbar sein
* Schnelle Abarbeitung von Anweisungen und Tätigkeiten

## Supportability

* Einfache Updates von Entwicklerseite möglich
* „Einstellungen“ implementieren: Spracheinstellungen, etc.

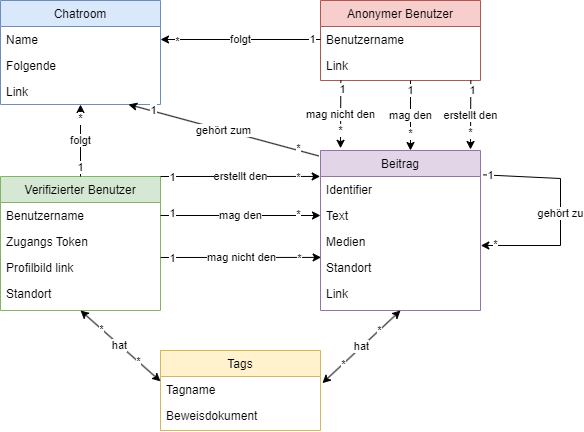
## Interface

* Modernes und ansprechendes Design mit Skeleton UI
* Selbe stilistische Merkmale global beibehalten
* Info-/ Startscreen und Mainscreen sichtbar trennen
* Projekt-Logo sichtbar und global darstellen

## Legal

* Image-Referenzen, GIF-Referenzen, Font-Referenzen, etc.
* -> Muss in Dokumentation angegeben werden (Quellverzeichnis)
* Keine geschützten Assets ohne Ursprungs-Angabe in Projekt verwenden

# Domänenmodell



### Erkenntnisse:

* Es ist eindeutig erkennbar das der anonyme Benutzer genau wie der verifizierte Benutzer alle Funktionen verwenden kann.
* Es ist zu erkennen das ein Chatroom nicht erstellt werden kann, denn das Prinzip des Chatrooms ist, dass ein Chatroom automatisch erstellt wird, sobald ein Beitrag in einem der möglichen Chatrooms erstellt wird.
* Beim Beitrag ist zu erkennen das er von sich selbst Abhängig sein kann, womit die Kommentar Funktion implementiert wird.
* Es ist zu erkennen das niemand Tags erstellen kann. Zusätzlich können nur anonyme Benutzer sich selbst hinzufügen.

# Funktionalitäts-Erklärung (Essenzielle Auszüge)

## Technologien

### Sveltekit

Ein Framework für die Erstellung von Single-Page-Anwendungen mit Svelte. Sie bietet Funktionen wie Routing, Server-Side-Rendering und statische Site-Generierung.

Im Projekt wurde SvelteKit verwendet da sie eine einfache und schnell zu lernende Option zwischen den verschiedenen Frameworks darstellt.

### Gunjs

Gunjs ist eine JavaScript-Bibliothek für Echtzeit-Webanwendungen. Sie bietet Funktionen wie Datenbindung, Datenreplikation und Offline-Unterstützung.

Im Projekt wurde GunJS zum Speichern der Daten verwendet, da sie durch das dezentralisierte Design dir Anonymität der Benutzer unterstützt.

### Express

Express ist ein Webanwendungs-Framework für Nodejs. Es bietet Funktionen wie Routing, Middleware und Templating.

Im Projekt wurde Express verwendet um ein schnelles und einfaches Backend zum verschicken von Emails und zugriff auf die MongoDB zu ermöglichen.

### MongoDB

MongoDB ist eine NoSQL-Datenbank für die Speicherung von Dokumenten. Sie ist skalierbar, performant und einfach zu verwenden.

Im Projekt wurde MongoDB verwendet, um die Daten der verifizierten Benutzer zu speichern, um ihre Identität zu garantieren.

### Gmail

Gmail ist ein kostenloser webbasierter E-Mail-Dienst von Google.

Im Projekt wurde Gmail zum Versenden von Verifizierungs-Emails verwendet.

### Imgur

Imgur ist eine Image-Hosting-Website zum Hochladen, Speichern und Teilen von Bildern.

Im Projekt wurde Imgur als Hosting Plattform zum Speichern der an Beiträge angehängten Dokumente verwendet.

### Vercel

Vercel ist eine Plattform für die Bereitstellung und das Hosting von Webanwendungen.

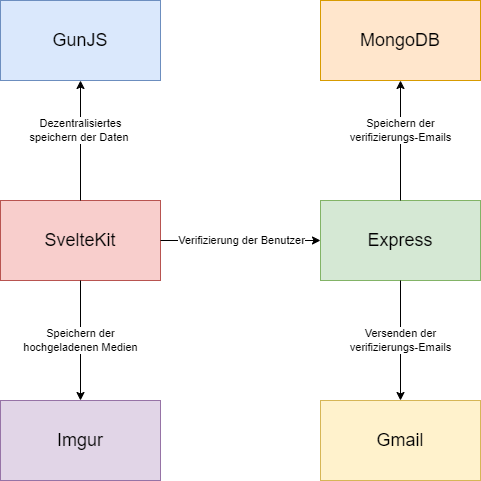
Im Projekt wurde Vercel zum Hosten der Sveltekit Applikation verwendet.

### Render

Render ist eine Plattform für die Bereitstellung und das Hosting von Serverless-Anwendungen.

Im Projekt wurde Render zum Hosten der Express Applikation verwendet.

## Projektstruktur



## Projektauszüge

### Funktionen mit Abweichungen

#### Authentifikations-System

Das Authentifizieren der Benutzer wird durch die GunJS Bibliothek ermöglicht.

Um einen Benutzer zu erstellen verwendet man die user.create Funktion bei der man mit einem einzigartigen Benutzernamen und einem Passwort einen Benutzer registrieren kann. Zusätzlich wird mit user.put dir E-Mail des Benutzers auf dem Benutzerobjekt gespeichert.

Zum Anmelden wird die user.auth Funktion verwendet bei der man mit dem Benutzernamen und dem Passwort ein Benutzerobjekt mit Informationen zum Benutzer zurückbekommt.

Das Abmelden wird die user.leave Methode verwendet der den momentan angemeldeten Benutzer abmeldet.

Das gesamte Authentifizieren wird auf der Frontend Seite mit Svelte-Stores ermöglicht. Der aktuelle Benutzer und Benutzername werden als Store global zur Verfügung gestellt.

#### Verifiziertes Login (Verifikations Server)

Um sicherzustellen das ein Benutzer der echte ist wurde entschlossen keine Drittidentitätsanbieter zu verwenden, sondern ein eigenes System zur Identifikation zu entwickeln. Die Verifizierung wird durch einen zusätzlichen Verifizierungs-Server verwaltet. Durch das Angeben einer E-Mail und anfordern eines Verifizierungscodes wird eine E-Mail verschickt, bei der der Benutzer durch das Bestätigen des Codes sich verifiziert. Diese Email wird auf einer Zusätzlichen MongoDB Datenbank mit dem jetzigen Status der Verifizierung gespeichert.

#### Anonymes Login

Das anonyme Login wird im Hintergrund genau gleich wie beim normalen verifizierten Benutzer abgewickelt. Im Gegensatz zum verifizierten Benutzer wird der Benutzername und das Passwort durch einen zufälligen Hash ersetzt. Nach dem Anonymen registrieren wird ein link in die zwischen Ablage des Benutzers kopiert, mit dem er sich wieder einloggen kann.

#### Routing Struktur des Frontends

Das Frontend ist grundsätzlich in 2 Hälften eingeteilt. Die Route /app und /auth, dabei sind unter /auth alle Funktionen bei der der Benutzer sich noch nicht angemeldet hat. Bei /app muss der Benutzer angemeldet sein und kann auf die Funktionen der App zugreifen.

Unter der /auth Route gibt es noch zusätzlich /auth/login und /auth/register bei denen man respektiv sich Anmelden und Registrieren kann. Eine /auth/link route gibt es auch noch bei der man mit einem link sich anmelden kann.

Unter der /app Route gibt es noch zusätzlich /app/r/, /app/b/ und /app/t/ bei denen man respektiv auf dir Raum Seite, Beitrags Seite und Tag Seite kommt.

#### Community Peer Server Struktur

GunJS basiert auf WebRTC Technologie. Um Kommunikation zwischen den Benutzern zu ermöglichen, werden sogenannte „signalling“ Server benötigt. Einige Server werden von der Community zur Verfügung gestellt. Eine Liste von solchen Servern ist auf der GunJS Dokumentationsseite verlinkt.

### Lernpunkte

#### Gunjs fehler der sea bibliothek

Eine der zusätzlichen Bibliotheken, die von GunJS verwendet werden, ist SeaJS. SeaJS kümmert sich um Security, Encryption und Autorisation. Durch das Verwenden von SvelteKit und dem SSR entsteht ein Fehler, der die gesamte Kommunikation unterbricht.

Um diesen Fehler zu beheben, wurde ein post install script erstellt bei der der Inhalt der sea.js Datei so abgeändert wird das Instanzen der Encoder Objekte von der globalen Instanz übernommen wird. Die Instanzen werden dann noch zusätzlich als globale Instanz in der auth.js Datei zur Verfügung gestellt.

## Testen der Grundfunktionen

### Registrieren

Um sich zu registrieren, muss man sich zuerst entscheiden, ob man ein verifiziertes Konto oder ein Anonymes Konto erstellen will. Dies wählt man auf der Landing Seite aus. Wenn man das anonyme Konto wählt, wird für den Benutzer alles automatisch generiert und er kann sofort fortfahren. Wenn man den verifizierten Weg wählt muss man zunächst auf verifiziert Fortfahren drücken, dann auf Registrieren. Nun kann man E-Mail, Benutzername und Passwort eingeben. Durch das Drücken des Registrieren Buttons wird der Benutzer erstellt. Nach dem Registrieren wird man auf die Home Seite weitergeleitet, und die Einstellungs Seite des Benutzers öffnet sich, bei der tags hinzugefügt und ein verifizierungs Code angefordert werden kann.



### Raum beitreten

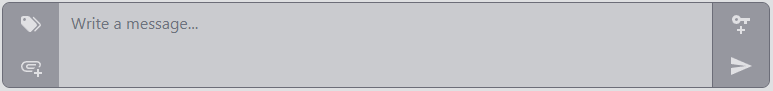
Nach dem Registrieren oder Anmelden kann man einem Raum über die Raumnavigation auf der linken Seite beitreten. Entweder kann man oben durch das Eingeben eines Namens und dem Drücken des Join Buttons einem neuen Raum beitreten oder ein schon favorisiertem Raum aus der Liste wählen.

Ein Bild, das Text, Schrift, Screenshot, Logo enthält.

Automatisch generierte Beschreibung

### Beitrag erstellen

Nach dem Beitreten eines Raums kann man auf der unteren Seite einen neuen Beitrag erstellen. Nach dem eingeben einer Nachricht kann man zusätzlich Medien, Schlüssel zum verschlüsseln und Tags hinzufügen. Danach wird durch das Drücken des Send Buttons die Nachricht an alle anderen Benutzer verschickt.



# Ausblick

### Erwähnung von Benutzern in Beiträgen

Beim Erstellen eines Beitrags soll es zusätzlich ermöglicht werden durch das Schreiben von einem @ einen anderen Benutzer zu erwähnen. Dieser Benutzer kann dann in der Raumnavigation unter dem Mentions Tab die Beiträge sehen bei denen er erwähn wurde.

### Laden von nur den letzten 100 Beiträgen

Momentan wird beim Beitreten eines Raums die gesamte Beitrags Historie geladen. Dies könnte bei Räumen mit viel Aktivität zum Problem werden. Darum sollte beim beitreten nur die letzten 100 Nachrichten geladen werden. Bei bedarf kann mehr nachgeladen werden.

### Anzahl der Kommentare Anzeigen

Momentan kann man zwar durch das Drücken den Kommentare Buttons auf die Kommentar Seite wechseln. Die Anzahl der Kommentare wird jedoch nicht angezeigt, wodurch man nicht weiß, ob ein Beitrag Kommentare hat oder nicht.

### Hosten alles Services auf eigenen Servern

Momentan laufen alle Applikationen auf „free to use“ Hosts (Vercel und Render). Diese sollten durch eigene Server ersetzt werden, um den Limitationen der Hosts beschränkt zu werden.

# Glossar

|  |  |
| --- | --- |
| script | Ein Skript ist eine Reihe von Befehlen, die von einem Computerprogramm ausgeführt werden. |
| post install | Post-Install bezieht sich auf den Zeitraum oder Vorgang, nachdem Software erfolgreich installiert wurde. |
| tag | Ein Tag ist ein Schlagwort oder eine Markierung, die verwendet wird, um Inhalte zu kategorisieren und die Suche zu erleichtern. |
| like, dislike | Begriffe aus sozialen Medien, mit denen Benutzer angeben können, ob ihnen ein Inhalt gefällt (Like) oder nicht gefällt (Dislike). |
| primary, secondary | Bezeichnet die primäre (Haupt-) und sekundäre (Neben-) Bedeutung oder Funktion von etwas. |
| framework | Ein Framework ist ein wiederverwendbarer Rahmen vorgefertigten Codes, der Entwicklern hilft, schneller und einfacher Anwendungen zu erstellen. |
| supplementary requirements | Zusätzliche Anforderungen, die nicht unbedingt zum Kern einer Software gehören, aber ihre Funktionalität erweitern können. |
| timetable | Ein Zeitplan oder Fahrplan, der die Ankunfts- und Abfahrtszeiten von Dingen wie Bussen, Zügen oder Flugzeugen angibt. |
| brief | Kurz und knapp zusammengefasst. |
| casual | Informell und ungezwungen. |
| user story | Eine kurze Beschreibung einer Funktionalität aus der Sicht des Benutzers. |
| use case | Ein Anwendungsfall beschreibt, wie ein Benutzer mit einem System interagiert, um ein bestimmtes Ziel zu erreichen. |
| UI | Die Benutzeroberfläche ist der Teil einer Software, mit dem der Benutzer interagiert. |
| free to use | Kostenlos nutzbar. |
| mentions | Erwähnungen sind in sozialen Medien, wenn jemand in einem Beitrag auf einen anderen Benutzer hinweist, in der Regel mit einem @-Symbol vor dem Benutzernamen. |
| join | Beitreten. |
| signalling | Signalgebung bezieht sich auf die Übertragung von Informationen zwischen Systemen oder Benutzern. |
| create | Erstellen. |
| put | Einfügen oder speichern. |
| auth | Authentifizierung ist der Prozess der Überprüfung der Identität eines Benutzers. |
| leave | Verlassen. |
|  |  |

# Quellverzeichnis

#### Icon Paket MDI

<https://icon-sets.iconify.design/mdi/>

#### Pixelify Sans Font fürs Logo

<https://fonts.google.com/specimen/Pixelify+Sans>

#### Dicebear zur Generation der Avatare

<https://www.dicebear.com/>

#### Gitter GunJS community

<https://app.gitter.im/#/room/#amark_gun:gitter.im>

#### GunJS Dokumentation

<https://gun.eco/docs/API>

#### SkeletonUI zum erstellen der UI

<https://www.skeleton.dev/>

#### Fragen zum Programmieren

<https://stackoverflow.com/>

#### Imgur speichern der medien

<https://imgur.com/>

#### Vercel hosten des Frontends

<https://vercel.com/>

#### Render hosten des Backends

<https://render.com/>

#### MongoDB speichern der verifizierten Emails

<https://www.mongodb.com/>