

AnoAsked

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

2023/2024



Inhaltsangabe

1	EIDESSTATTLICHE ERKLÄRUNG DES AUTORS	II
2	KURZBESCHREIBUNG	III
2.1	DEUTSCH	III
2.2	ENGLISCH	III
3	TIMETABLE UND INITIALE ZIELE	IV
3.1	TIMETABLE	IV
3.2	INITIALE ZIELE	IV
4	ANFORDERUNGSMANAGEMENT	V
4.1	USE-CASE-DIAGRAMM	V
4.2	AKTOREN-ANALYSE	V
4.3	ANWENDUNGSFÄLLE	VI
5	SUPPLEMENTARY REQUIREMENTS	X
5.1	USABILITY	X
5.2	RELIABILITY	X
5.3	PERFORMANCE	X
5.4	SUPPORTABILITY	X
5.5	INTERFACE	X
5.6	LEGAL	X
6	DOMÄNENMODELL	XI
7	FUNKTIONALITÄTS-ERKLÄRUNG	XII
7.1	TECHNOLOGIEN	XII
7.2	PROJEKTSTRUKTUR	XIII
7.3	PROJEKTAUSZÜGE	XIV
7.4	TESTEN DER GRUNDFUNKTIONEN	XVI
8	AUSBLICK	XVII
8.2	SCHLUSSBEMERKUNG	XVIII
9	GLOSSAR	XIX
10	QUELLVERZEICHNIS	XX



1 Eidesstattliche Erklärung des Autors

Hiermit erkläre ich an Eides statt, dass ich die vorliegende Arbeit nur mit der angeführten Literatur, selbstständig und ohne Mitwirkung Dritter verfasst habe. Die Projektarbeit umfasst etwa 3300 Wörter.

Masooglu Simon



2 Kurzbeschreibung

2.1 Deutsch

Das Internet und der Meinungsaustausch in sozialen Medien haben die Tür zur Welt und zu inspirierenden Perspektiven geöffnet. Allerdings sollte insbesondere hinsichtlich des Schutzes der Privatsphäre die Möglichkeit bestehen, sich bewusst vor eventuellen persönlichen Angriffen oder Datenmissbrauch zu schützen.

Im Rahmen meiner Projektarbeit habe ich ein Modell einer sicheren Anwendung entwickelt. **AnoAsked** ist eine Social-Media-Plattform. Ihr Schwerpunkt liegt auf der Wahrung der Anonymität. Durch Funktionen wie Kommentieren und Liken von Beiträgen können Diskussionen geführt werden.

AnoAsked ermöglicht die Veröffentlichung von Beiträgen unter maximalem Schutz der Privatsphäre. Sowohl verifizierte als auch anonyme Benutzer können die Funktionen vollumfänglich nutzen.

2.2 Englisch

The internet and social media have opened the door to the world and to inspiring perspectives. However, especially about privacy protection, there should be the possibility of consciously protecting oneself from potential personal attacks or data misuse.

With my project I have developed the model of a secure application. **AnoAsked** is a social media platform. Its focus is on preserving anonymity.

Discussions can be conducted through functions such as commenting and liking posts. **AnoAsked** enables the publication of posts with maximum privacy protection. Both verified and anonymous users can fully utilize the functions.



3 Timetable und initiale Ziele

3.1 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 von 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	Erstellen der Error-Seite Erstellen der Login- und Register-Seiten
5	13.03.2024 03.04.2024	Integrieren des 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 Seitenlayouts mit Raumnavigatio, Erstellen der Raum-Seite mit Nachrichten-Eingabebox
8	30.04.2024 07.05.2024	Hinzufügen der Beitrags-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

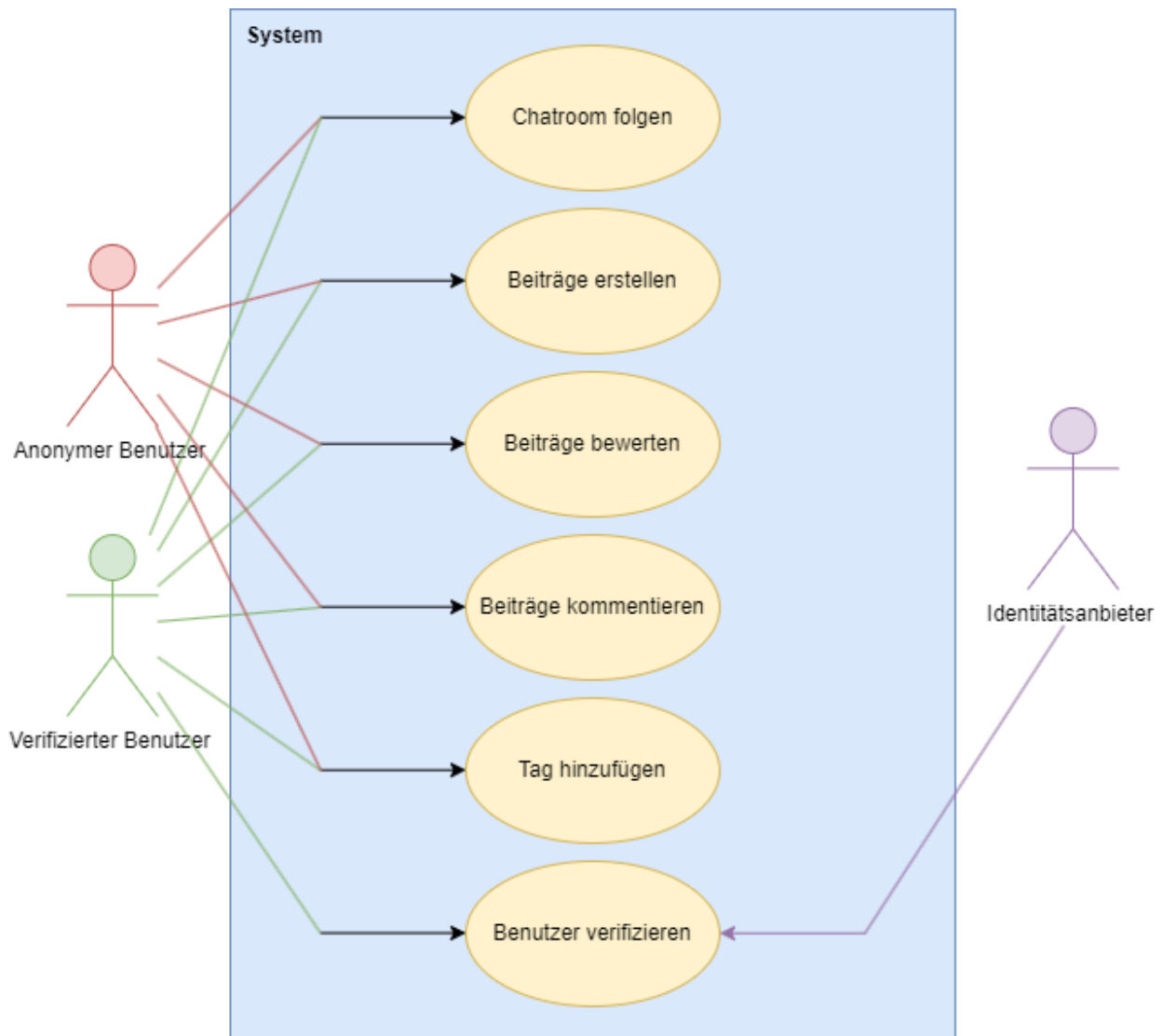
3.2 Initiale Ziele

- 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



4 Anforderungsmanagement

4.1 Use-Case-Diagramm



4.2 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.
- Als Identitätsanbieter (Supporting Actor) besteht die Möglichkeit, sich beispielsweise über einen GitHub-Account zu verifizieren, um seine Identität zu bestätigen.

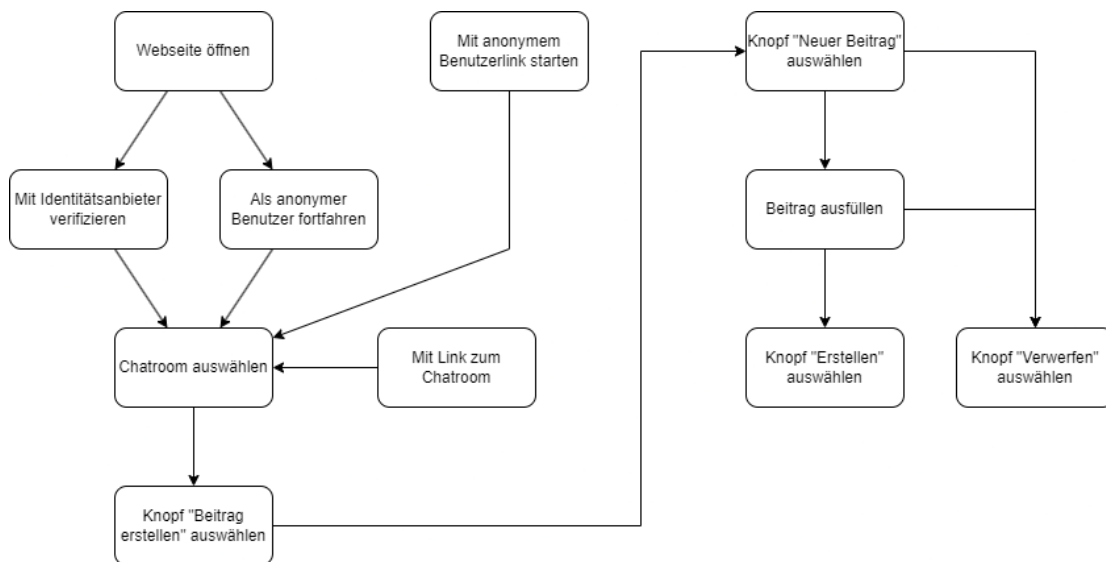


4.3 Anwendungsfälle

Hinweis: Die Storys der Anwendungsfälle werden aus der Perspektive eines fiktiven Benutzers wiedergegeben.

4.3.1 Anwendungsfall: Beitrag erstellen (casual)

4.3.1.1 Diagramm



4.3.1.2 Story

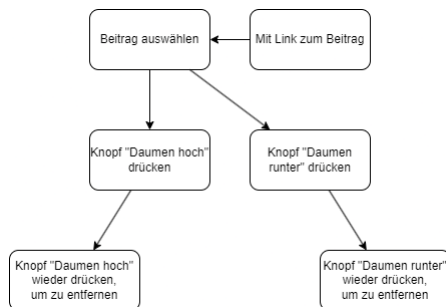
Ein Benutzer befindet sich auf einer Party und spricht mit seinen Freunden über Meinungsfreiheit. Alle seine Freunde sind der gleichen Meinung, aber er will noch andere Meinungen kennenlernen. Deshalb möchte er im Internet einen Beitrag zu diesem Thema erstellen. Doch er will damit nicht seine eigene Privatsphäre gefährden. Ihm fällt ein, dass die App **AnoAsked** verspricht, seine Anonymität zu bewahren und beschließt, auf die **AnoAsked**-Webseite zu gehen.

Nachdem der Benutzer die Webseite gestartet hat, wird er von 2 Optionen auf der Landing-Page, begrüßt. Er hat die Optionen, 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, auf der 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 verfasst einen Beitrag. 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.



4.3.2 Anwendungsfall: Beitrag bewerten (brief)

4.3.2.1 Diagramm

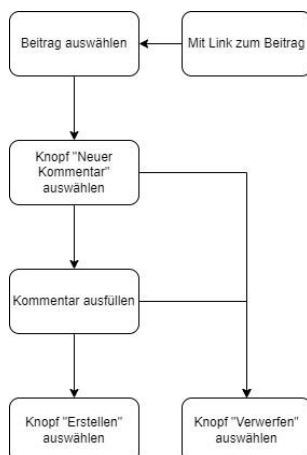


4.3.2.2 User Story

Nach einiger Zeit sieht unser Benutzer, dass jemand unter seinem Beitrag einen Kommentar erstellt hat. Er findet den Kommentar sehr interessant und will ihn bewerten. Unser 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 sichtbar.

4.3.3 Anwendungsfall: Beitrag kommentieren (brief)

4.3.3.1 Diagramm



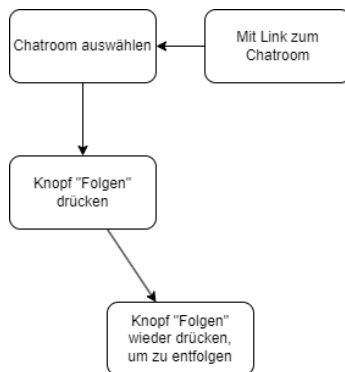
4.3.3.2 Story

Auch ein anderer Benutzer hat einen Kommentar unter dem Beitrag erstellt. Unser Benutzer findet aber, dass der Verfasser dieses Kommentars seinen originalen Beitrag nicht richtig verstanden hat und will ihm dies mitteilen. Dafür drückt er auf die Schaltfläche „Kommentieren“. Unser Benutzer füllt wie bei einem Beitrag die gewünschten Felder aus und drückt, nachdem er seine Meinung nochmal verdeutlicht hat, auf „Erstellen“. Somit können alle anderen seinen Kommentar dazu sehen und die gleichen Funktionen wie bei einem Beitrag verwenden, um mit dem Kommentator zu interagieren.



4.3.4 AnwendungsFall: Chatroom Folgen (brief)

4.3.4.1 Diagramm



4.3.4.2 Story

Nach einer lehrreichen Diskussion im Chatroom „r/discussions“ möchte unser Benutzer dem Chatroom folgen, um immer auf dem neuesten Stand der heißen Beiträge in diesem Chatroom zu bleiben. Er klickt auf „Folgen“ und bekommt damit auf der Homepage die Neuigkeiten angezeigt.

4.3.8 AnwendungsFall: Benutzer verifizieren (brief)

4.3.8.1 Diagramm



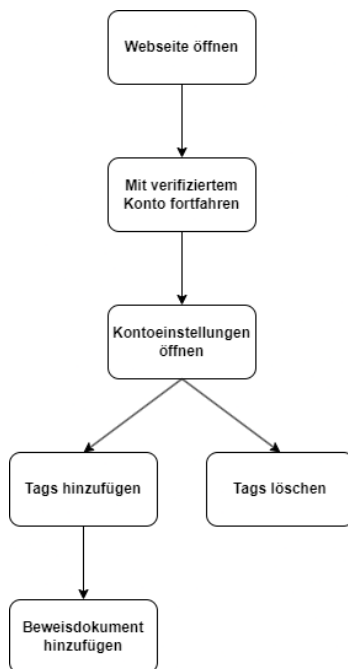
4.3.8.2 Story

Nach einem langen Hin und Her von Argumenten entscheidet sich unser 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“. Daraufhin 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.



4.3.6 Anwendungsfall: Verifizierter Benutzer, Tags hinzufügen (brief)

4.3.6.1 Diagramm



4.3.6.2 Story

Unser Benutzer ist Politikwissenschaftler und hat ein beurkundetes 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, indem er auf sein Profilbild klickt. Nun hat er die Option, Tags hinzuzufügen oder zu löschen. Er drückt auf die „Tag hinzufügen“-Schaltfläche. Anschließend muss er den Namen des Tags definieren und ein Bild seiner Sponsionsurkunde hochladen. Danach drückt er auf die „Tag erstellen“-Schaltfläche. Jetzt ist der Tag hinzugefügt und alle können sehen, dass er ein verifizierter Politikwissenschaftler ist.



8 Supplementary Requirements

8.1 Usability oder Benutzerfreundlichkeit

- Einfache Handhabung
- Große und eindeutige Symbole
- Gut leserliche Schriftart
- Futuristisches Design
- Eindeutige und erkennbare Bedienfeldeinteilung

8.2 Reliability oder Zuverlässigkeit

- Dezentralisiertes Backend, um vor Ausfällen zu schützen
- Zusätzliche Server-Nodes, um vor Backend-Ausfällen zu schützen
- Frontend/UI sollte 24/7 Zugriff bieten

8.3 Performance oder Leistungsfähigkeit

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

8.4 Supportability oder Wartungsfreundlichkeit

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

8.5 Interface oder Interaktionsschnittstellen

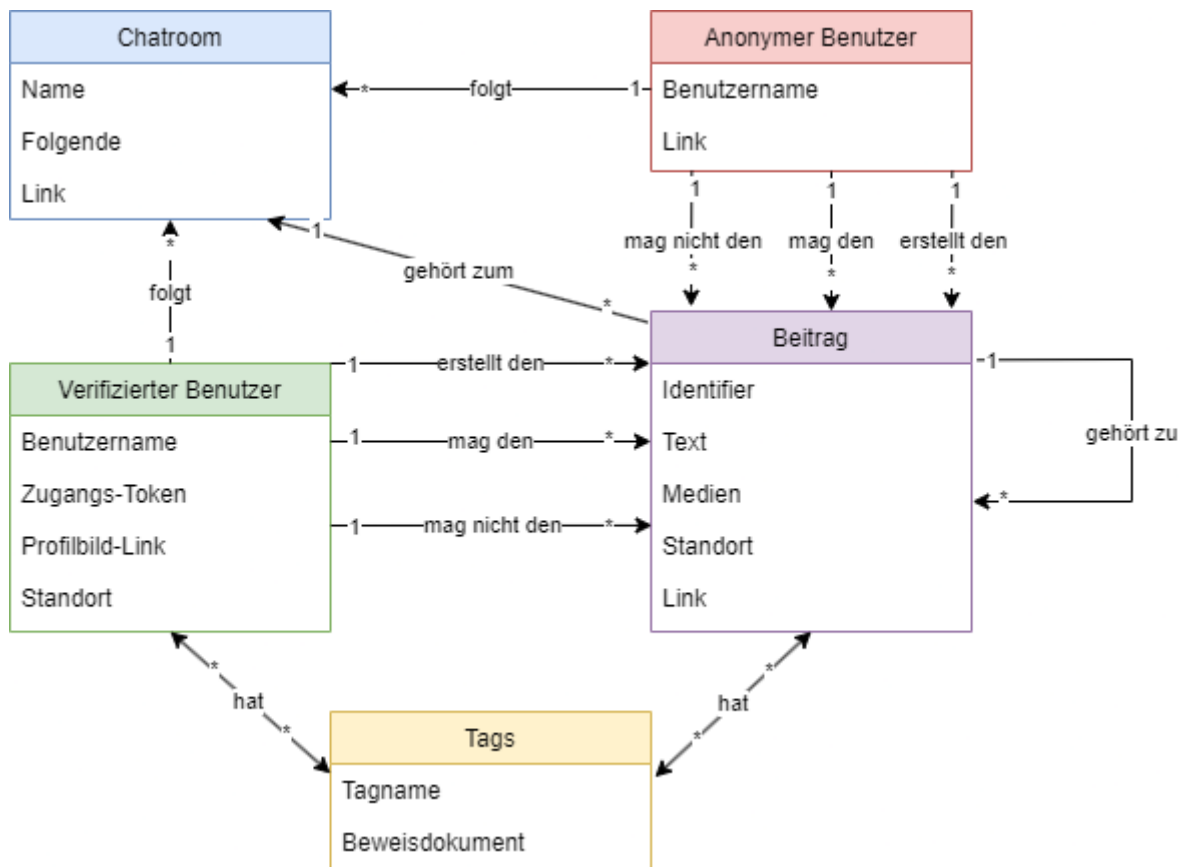
- Modernes und ansprechendes Design mit Skeleton UI
- Stilistische Merkmale global beibehalten
- Info-/ Startscreen und Mainscreen sichtbar trennen
- Projekt-Logo sichtbar und global darstellen

8.6 Legal oder Rechtskonformität

- Image-Referenzen, GIF-Referenzen, Font-Referenzen etc.
- Quellverzeichnis muss in Dokumentation angegeben werden
- Keine geschützten Assets ohne Ursprungs-Angabe in Projekt verwenden



6 Domänenmodell



6.1.1 Erkenntnisse:

- Es ist eindeutig erkennbar, dass der anonyme Benutzer genau so wie der verifizierte Benutzer alle Funktionen verwenden kann.
- Es ist zu erkennen, dass ein Chatroom nicht erstellt werden kann. Denn das Prinzip eines Chatrooms ist, dass der Chatroom automatisch erstellt wird, sobald ein Beitragseintrag erfolgt ist.
- Beim Beitrag ist zu erkennen, dass er von sich selbst abhängig sein kann, womit die Kommentar-Funktion implementiert wird.
- Es ist zu erkennen, dass niemand Tags erstellen kann. Zusätzlich können nur anonyme Benutzer sich selbst hinzufügen.



7 Funktionalitäts-Erklärung in Kurzfassung

7.1 Technologien

7.1.1 Sveltekit

Svelte ist ein Framework für die Erstellung von Single-Page-Anwendungen. Es bietet Funktionen wie Routing, Server-Side-Rendering und statische Site-Generierung. Im Projekt wurde SvelteKit verwendet, da es eine einfache und schnell zu lernende Option zwischen den verschiedenen Frameworks darstellt.

7.1.2 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 die Anonymität der Benutzer unterstützt.

7.1.3 Express

Express ist ein Webanwendungs-Framework für Node.js. Sie bietet Funktionen wie Routing, Middleware und Templating. Im Projekt wurde Express verwendet, um ein schnelles und einfaches Backend zum Verschicken von E-Mails und Zugriff auf die MongoDB zu ermöglichen.

7.1.4 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 und um ihre Identität zu garantieren.

7.1.5 Gmail

Gmail ist ein kostenloser webbasierter E-Mail-Dienst von Google. Im Projekt wurde Gmail zum Versenden von Verifizierungs-E-Mails verwendet.

7.1.6 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.

7.1.7 Vercel

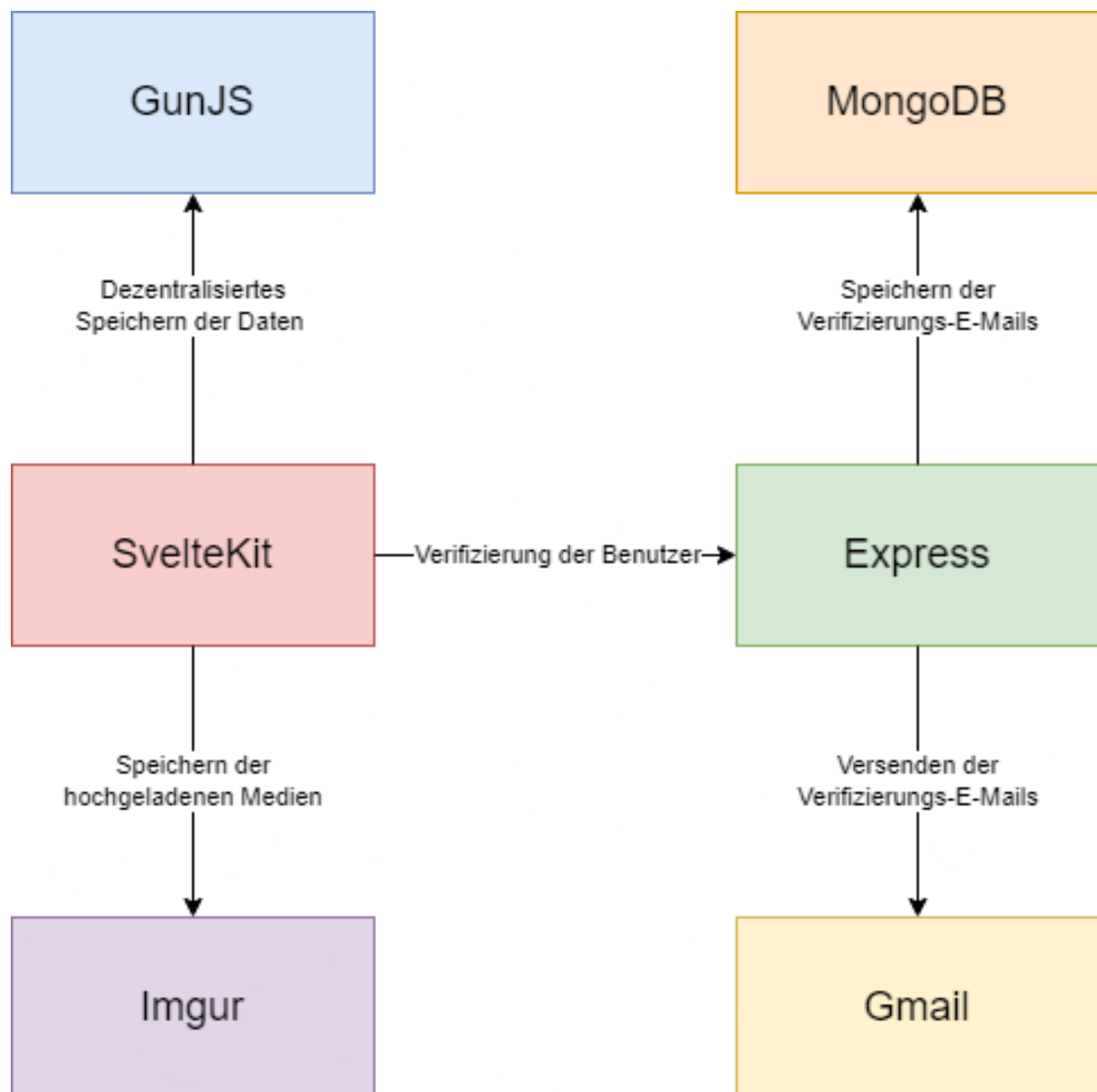
Vercel ist eine Plattform für die Bereitstellung und das Hosting von Webanwendungen. Im Projekt wurde Vercel zum Hosten der Sveltekit-Applikation verwendet.

7.1.8 Render

Render ist eine Plattform für die Bereitstellung und das Hosting von Serverless-Anwendungen. Im Projekt wurde Render zum Hosten der Express-Applikation verwendet.



7.2 Projektstruktur





7.3 Projektauszüge

7.3.1 Funktionen mit Abweichungen

7.3.1.1 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“ die 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.

Für das Abmelden wird die „user.leave“-Methode verwendet, die 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.

7.3.1.2 Verifiziertes Login (Verifikations-Server)

Um sicherzustellen, dass ein Benutzer der echte ist, wurde beschlossen, keinen Drittidentitätsanbieter zu verwenden, sondern ein eigenes System zur Identifikation zu entwickeln. Dies stellt sicher, dass Benutzerinformationen durch Dritte nicht missbraucht werden können. Die Verifizierung wird durch einen zusätzlichen Verifizierungs-Server verwaltet. Durch das Angeben einer E-Mail-Adresse und das Anfordern eines Verifizierungscode wird eine E-Mail verschickt, bei der sich der Benutzer durch das Bestätigen des Codes verifiziert. Diese E-Mail wird auf einer zusätzlichen MongoDB-Datenbank mit dem jetzigen Status der Verifizierung gespeichert.

7.3.1.3 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 Zwischenablage des Benutzers kopiert, mit dem er sich wieder einloggen kann.



7.3.1.4 Routing-Struktur des Frontends

Das Frontend ist grundsätzlich in 2 Hälften eingeteilt: Die Routen „/auth“ und „/app“.

Unter „/auth“ stehen alle Optionen zur Anmeldung zur Verfügung.

Unter „/app“ muss der Benutzer angemeldet sein und kann erst dann auf die Funktionen der App **AnoAsked** zugreifen.

Unter der „/auth“-Route gibt es noch zusätzlich weiterführende Routen, „/auth/login“ und „/auth/register“, bei denen man sich respektiv anmelden und registrieren kann. Außerdem steht noch eine „/auth/link“-Route zur Verfügung, bei der man sich mit einem Link anmelden kann.

Unter der „/app“-Route gibt es noch zusätzlich weiterführende Routen, „/app/r/“, „/app/b/“ und „/app/t/“, bei denen man respektiv auf die Chat-Raum-Seite, Beitrags-Seite und Tag-Seite kommt.

7.3.1.5 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.

7.3.2 Lernpunkte

7.3.2.1 GunJS-Fehler der Sea-Bibliothek

Eine der zusätzlichen Bibliotheken, die von GunJS verwendet wird, 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.

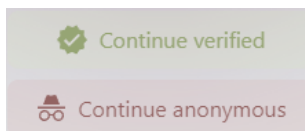
Um diesen Fehler zu beheben, wurde ein Post-Install-Script erstellt, bei dem der Inhalt der „sea.js“-Datei so abgeändert wird, dass Instanzen der Encoder-Objekte von der globalen Instanz übernommen werden. Die Instanzen werden dann noch zusätzlich als globale Instanz in der „auth.js“-Datei zur Verfügung gestellt.



7.4 Testen der Grundfunktionen

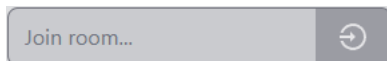
7.4.1 Registrieren

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-Adresse, Benutzername und Passwort eingeben. Durch das Drücken der Registrieren-Schaltfläche wird der Benutzer erstellt. Nach dem Registrieren wird man auf die Home-Seite weitergeleitet und die Einstellungs-Seite des Benutzers öffnet sich. Dort können Tags hinzugefügt und ein Verifizierungs-Code angefordert werden.



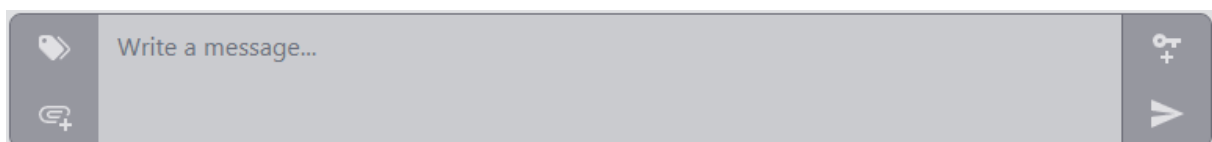
7.4.2 Raum beitreten

Nach dem Registrieren oder Anmelden kann man über die Raumnavigation auf der linken Seite einem Chat-Raum beitreten. Entweder kann man oben durch das Eingeben eines Namens und das Drücken der Join-Schaltfläche einem neuen Chat-Raum beitreten oder einen schon favorisierten Chat-Raum aus der Liste wählen.



7.4.3 Beitrag erstellen

Nach dem Beitreten zu einem Chat-Raum kann man auf der unteren Seite einen neuen Beitrag erstellen. Nach dem Eingeben einer Nachricht ist es möglich, zusätzlich Medien, Schlüssel zum Verschlüsseln und Tags hinzuzufügen. Danach wird durch das Drücken der Senden-Schaltfläche die Nachricht an alle anderen Benutzer verschickt.





8 Ausblick auf Weiterentwicklungen

8.1.1 Erwähnung von Benutzern in Beiträgen

Beim Erstellen eines Beitrags soll es zusätzlich ermöglicht werden, mithilfe eines @-Zeichens einen anderen Benutzer zu erwähnen. Dieser Benutzer kann dann in der Raumnavigation unter dem Mentions-Tab die Beiträge sehen, in denen er erwähnt worden ist.

8.1.2 Laden von „nur den letzten 100 Beiträgen“

Momentan wird beim Beitreten zu einem Chat-Raum die gesamte Beitrags-Historie geladen. Dies könnte bei Räumen mit viel Aktivität zum Problem werden. Darum sollten beim Beitreten nur die letzten 100 Nachrichten geladen werden. Bei Bedarf kann mehr nachgeladen werden.

8.1.3 Anzahl der Kommentare anzeigen

Momentan kann man zwar durch das Drücken der Kommentar-Schaltfläche auf die Kommentar-Seite wechseln, die Anzahl der Kommentare wird jedoch nicht angezeigt. Man weiß also nicht, ob ein Beitrag Kommentare hat oder nicht.

8.1.4 Beweis-Dokument Für Tags hochladen

Momentan können Benutzer zwar Tags hinzufügen, doch noch keine Urkunden anhängen. Die Tag-Funktion soll also so erweitert werden, sodass man diese Beweisdokumente hinzufügen kann.

8.1.5 Hosten aller Services auf eigenen Servern

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



8.2 Schlussbemerkung

In Sozialen Medien präsent zu sein, kann belastend werden, wenn Benutzer Grenzüberschreitungen oder Datenmissbrauch ausgesetzt sind. Um sich möglichen negativen Folgen der freien Meinungsäußerung im öffentlichen Raum sozialer Medien nicht aussetzen zu müssen, ist es wichtig, Benutzern die Option, die Privatsphäre zu schützen, ohne lange Umwege zur Verfügung zu stellen.

Mein Projekt **AnoAsked** soll in diesem Sinne ein Beitrag dazu sein. Ich habe mich dafür tiefer mit den Themen Verschlüsselung, sichere Datenübertragung und Nutzerauthentifizierung auseinandergesetzt. Nach sich wiederholenden Testphasen und der darauffolgenden Fehlerbehebung habe ich festgestellt, dass mein Projekt doch sehr ambitioniert und zeitaufwendig ist. Die größte Herausforderung allerdings war für mich das eigentliche Projektmanagement mit der notwendigen Selbstorganisation.

Persönlich konnte mit diesem Projekt meine Kreativität, meine Programmierkenntnisse und meine Problemlösungsfähigkeit trainieren und unter Beweis stellen.



9 Glossar

account Konto	Hier wird damit die Identität und die Berechtigungen eines Benutzers verwaltet und der Zugriff auf Ressourcen und Funktionen kontrolliert.
actor	Benutzer
assets	Damit bezeichnet man gestalterische Ressourcen wie Bilder, Videos, Schriftarten etc.
auth	Damit bezeichnet man den Prozess der Überprüfung der Identität eines Benutzers.
brief	kurz und knapp zusammengefasst
backend	Damit bezeichnet man einen Teil einer Hintergrund-Softwareanwendung, die sich um die serverseitige Logik und Datenverarbeitung kümmert. Es kommuniziert über Schnittstellen mit dem Frontend.
chatroom	Damit bezeichnet man einen virtuellen Raum, in dem Benutzer in Echtzeit miteinander kommunizieren können.
Domäne	Hier bezieht sich der Begriff auf die Entwicklungsschritte einer Softwarelösung im Kontext der Benutzersicherheit.
encoder	Datumumwandler
encryption	Verschlüsselung von Daten für Unbefugte
framework	Der Begriff bezeichnet einen wiederverwendbaren Rahmen von vorgefertigten Codes, der Entwicklern hilft, schneller und einfacher Anwendungen zu erstellen.
frontend	Damit bezeichnet man den Teil der Softwareanwendung, der die Benutzeroberfläche präsentiert und die Interaktion mit dem Benutzer managt. Es kommuniziert über Schnittstellen mit dem Backend.
GitHub	Plattform zur Verwaltung verschiedener Versionen von Softwareprojekten
hash	Der Begriff bezeichnet die kryptografisch sichere Umwandlung von Daten beliebiger Länge in eine Zeichenfolge fester Länge.
homepage	Damit bezeichnet man die Hauptseite einer Webseite, die beim Öffnen als erstes angezeigt wird und dem Benutzer ein Spektrum an Funktionen anbietet.
host	Dienstleister, der über ein Netzwerk Daten und Funktionen bereitstellt
link	Mit diesem Begriff bezeichnet man die Zugriffsmöglichkeit auf verknüpfte Ressourcen.
log in	Anmeldung
landing page	Damit bezeichnet man eine Seite einer Webseite, die beim Öffnen meist als erstes angezeigt wird und dem Benutzer nur eine zweckgebundene Funktion anbietet.
like, dislike	Mit den Begriffen aus sozialen Medien können Benutzer angeben, ob ihnen ein Inhalt gefällt (like) oder nicht gefällt (dislike).
mentions	Damit werden Erwähnungen in sozialen Medien bezeichnet: Wenn jemand in einem Beitrag auf einen anderen Benutzer hinweist, geschieht das in der Regel mit einem @-Symbol vor dem Benutzernamen.
middleware	Damit wird Software bezeichnet, die es ermöglicht sich über verschiedene komplexe Systeme hinweg auszutauschen. Sie liegt zwischen Anwendungsprogrammen, Betriebssystem oder Hardware.
NoSQL Datenbank	Datenbank mit flexibler Struktur
post install	Der Begriff bezieht sich auf den Zeitraum oder Vorgang, nachdem Software erfolgreich installiert worden ist.
primary secondary	Damit wird die primäre (Haupt-) und sekundäre (Neben-) Bedeutung oder Funktion von etwas bezeichnet.
script	Damit bezeichnet man eine Reihe von Befehlen, die von einem Computerprogramm ausgeführt werden.
server	Damit bezeichnet man Computer, die anderen Dienste, Ressourcen oder Daten im Netzwerk bereitstellen.
serverless	Der Begriff bezieht sich auf Anwendungen, die aufgrund von Anfragen automatisch aktiviert werden. Infrastruktur und Wartung übernehmen Cloud-Anbieter.
server nodes	Damit bezeichnet man die einzelnen Server in einem Server Cluster.
supplementary requirements	Das sind zusätzliche Anforderungen, die nicht unbedingt zum Kern einer Software gehören, aber ihre Funktionalität erweitern können.
rendering	Prozess der Visualisierung von Daten
routing	Weiterleitung von Daten auf optimalen Routen in einem Netzwerk
signalling	Der Begriff bezieht sich auf die Übertragung von Informationen zwischen Systemen oder Benutzern.
single page	Eine SPA lädt nur einmal eine Haupt-HTML-Seite.
store	Speichereinheit
tag	Dabei handelt es sich um ein Schlagwort oder eine Markierung, die verwendet wird, um Inhalte zu kategorisieren und die Suche zu erleichtern.
templating	Verwendung von Vorlagen für Wiederkehrendes
timetable	Damit wird der Zeit- und Ablaufplan der Entwicklung eines Projekts nachvollziehbar dokumentiert.
UI	Der Begriff bezeichnet eine Benutzeroberfläche als Teil einer Software, mit der der Benutzer interagiert.
use case	Damit wird ein Anwendungsfall beschrieben: Wie interagiert ein Benutzer mit einem System, um ein bestimmtes Ziel zu erreichen.
user story	Damit wird eine kurze Beschreibung einer Funktionalität aus der Sicht des Benutzers beschrieben.



10 Quellverzeichnis

10.1.1.1 Icon-Paket MDI

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

10.1.1.2 Pixelify-Sans-Font Fürs Logo

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

10.1.1.3 Dicebear zur Generation der Avatare

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

10.1.1.4 Gitter-GunJS-Community

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

10.1.1.5 GunJS-Dokumentation

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

10.1.1.6 SkeletonUI zum Erstellen der UI

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

10.1.1.7 Fragen zum Programmieren

<https://stackoverflow.com/>

10.1.1.8 Imgur: speichern der Medien

<https://imgur.com/>

10.1.1.9 Vercel: hosten des Frontends

<https://vercel.com/>

10.1.1.10 Render: hosten des Backends

<https://render.com/>

10.1.1.11 MongoDB: speichern der verifizierten E-Mails

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