

AnoAsked

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

**2023/2024**

# 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 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.

# Inhaltsverzeichnis

[1 Eidesstattliche Erklärung des Autors i](#_Toc159798889)

[2 Kurzbeschreibung ii](#_Toc159798890)

[2.1 Deutsch ii](#_Toc159798891)

[2.2 Englisch ii](#_Toc159798892)

[3 Inhaltsverzeichnis iii](#_Toc159798893)

[4 Ziel iv](#_Toc159798894)

[5 Anforderungsmanagement iv](#_Toc159798895)

[5.1 Use Case-Diagramm iv](#_Toc159798896)

[5.2 Aktoren Analyse v](#_Toc159798897)

[5.3 Anwendungsfälle v](#_Toc159798898)

[5.3.1 Anwendungsfall Beitrag erstellen (casual) v](#_Toc159798899)

[5.3.2 Anwendungsfall Beitrag bewerten (brief) vi](#_Toc159798900)

[5.3.3 Anwendungsfall Beitrag kommentieren (brief) vii](#_Toc159798901)

[5.3.4 Anwendungsfall Chatroom folgen (brief) vii](#_Toc159798902)

[5.3.5 Anwendungsfall Benutzer Verifizieren (brief) viii](#_Toc159798903)

[5.3.6 Anwendungsfall Verifizierter Benutzer Tags hinzufügen (brief) ix](#_Toc159798904)

[6 Supplementary Requirements ix](#_Toc159798905)

[6.1.1 Usability ix](#_Toc159798906)

[6.1.2 Reliability ix](#_Toc159798907)

[6.1.3 Performance ix](#_Toc159798908)

[6.1.4 Supportability x](#_Toc159798909)

[6.1.5 Interface x](#_Toc159798910)

[6.1.6 Legal x](#_Toc159798911)

[6.2 Domänenmodell x](#_Toc159798912)

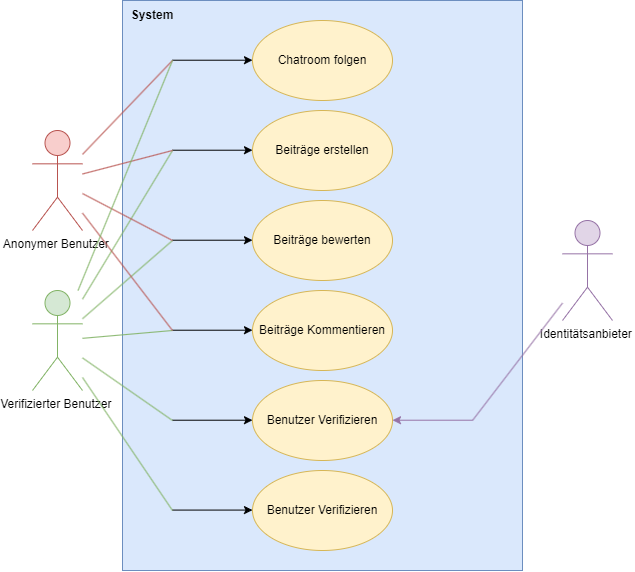
[6.2.1 Erkenntnisse: x](#_Toc159798913)

# Ziel

* 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 c/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 c/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 c/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.

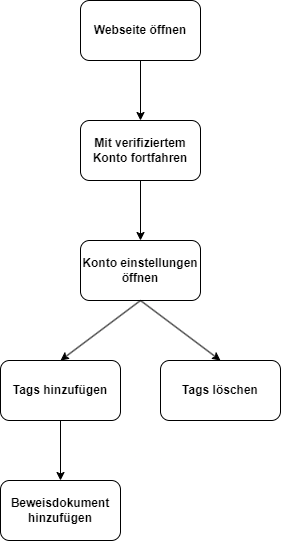
Automatisch generierte Beschreibung

#### 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.

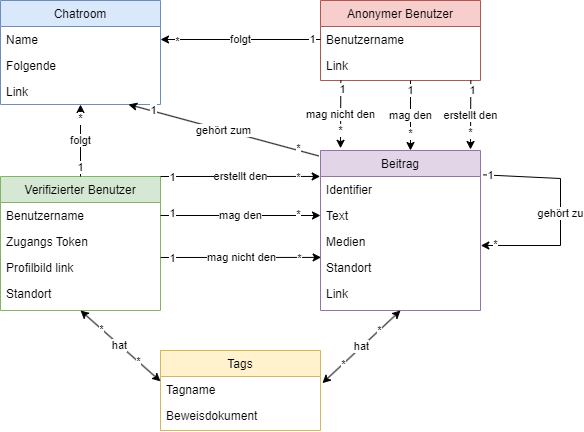
### Interface

* 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.