

Dokumentation Projekt questMe

Pavithra Sureshkumar, Kevin Sautner, Ralf Zeller

Geändert 11.11.2021

Inhaltsverzeichnis

Abbildungsverzeichnis	3
Tabellenverzeichnis	4
1 Vorwort	5
2 About Us	5
3 UI Designs	6
3.1 Recherche UI Designs	6
3.2 Version 2	8
3.3 Version 1	13
3.4 Kriterien für Usability	14
3.5 Usability Test	15
4 User Stories	16
4.1 Struktur	16
4.2 User Stories Version 1	16
5 Technologien	19
5.1 Entscheidung für die Technologien	19
5.2 Technologie Versionen	19
5.2.1 Node.js 16.13.0 LTS	20
5.2.2 Angular 13.0.1	21
5.2.3 Socket.io 4.3.2	21
5.2.4 MongoDB Community Edition 5.0.3	21
5.2.5 Keycloak 15.0.2	21
5.2.6 RegEx	21
5.2.7 NLP.js 4.22.2 (optional)	21

5.3	Technologie Vergleich	22
5.3.1	Angular vs Vue.js	22
5.3.2	MongoDB vs PostgreSQL	23
5.4	Technologie Diagramme	24
5.4.1	Komponentendiagramme	24
6	Meilensteine	28
6.1	Recherche 13.10.21	28
6.2	Zwischenpräsentation 22.10.21	28
6.3	Implementation 25.10.21	29
6.4	MVP 03.12.21	29
6.5	Endpräsentation 14.01.22	29
6.6	Enddokumentation 14.01.22	29
7	Zeitmanagement	30
7.1	Gantt-Diagramm	30
8	Risikoanalyse	31
	Literatur	34

Abbildungsverzeichnis

1	Komponentendiagramm Client	24
2	Komponentendiagramm Server	25
3	Komponentendiagramm v1.2	26
4	Komponentendiagramm v1.1	26
5	Komponentendiagramm v1.0	27

Tabellenverzeichnis

1	Technologie Liste	20
2	Angular vs Vue.js	22
3	MongoDB vs PostgreSQL	23
4	Meilenstein Liste	28

1 Vorwort

In dieser Dokumentation möchten wir alle relevanten Informationen zum Projekt sammeln.

2 About Us

Hier stellen wir uns kurz vor.

Wer sind wir ?



Pavithra
Sureshkumar
Frontend Dev



Ralf Zeller
Backend Dev



Kevin Sautner
Backend Dev



Wir sind questMe und wir bearbeiten das Thema ChatBot. In unserem Projekt möchten wir nicht nur den ChatBot programmieren oder erstellen, sondern auch neue Technologien kennenlernen. Wir möchten neue Erfahrungen sammeln, Lernen wie man organisiert mit Konflikten umgeht und auch bestimmte Probleme angeht. Wir sind als Team immer offen für Neues.

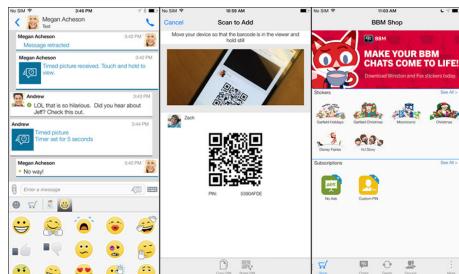
Wir sind das Team questMe.

3 UI Designs

In diesem Kapitel sollen die verschiedenen Versionen unseres UI Designs geführt werden

3.1 Recherche UI Designs

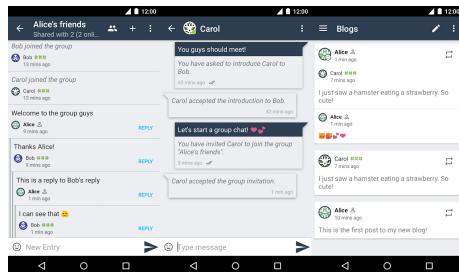
Zuerst haben wir uns mehrere UI Designs von verschiedenen Quellen angeschaut, um einen besseren Überblick über die Designmöglichkeiten zu bekommen.



Bildquelle: *Blackberry-messenger-live-free*, [o. D.]

Blackberry Messenger Live

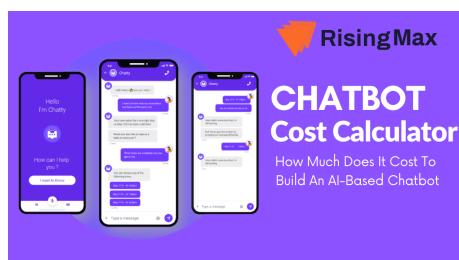
Zuerst haben wir uns umgeschaut und nach Chatfenstern gesucht. Hier haben wir uns die Austauschung von Sprechblasen angeschaut und deren Ausrichtung.



Bildquelle: *Briar*, [o. D.]

Briar

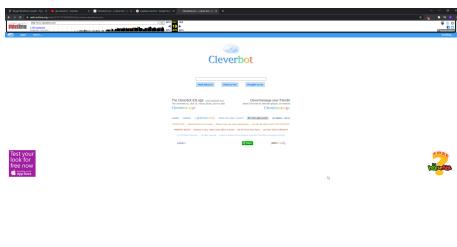
In diesem Beispiel haben wir uns wieder das Chatfenster und die verschiedenen Symbole angeschaut. Wie das Editierbutton oder das Hinzufügebutton.



Bildquelle: *Chatbot-cost-calculator*, [o. D.]

CHATBOT Cost Calculator

Hier haben wir uns ein ChatBot-Fenster angeschaut und dabei die Icons und die verschiedenen Elemente angeschaut, wie den Balken an der Oberseite oder das Sendebutton.



Cleverbot

Diesen Bot haben wir uns genauer angeschaut, weil dieser auch uns bekannt war.

Bildquelle: *Cleverbot*, [o. D.]



Telegram

Telegram haben wir uns wiederrum das Chatfenster und die Icons angeschaut.

Bildquelle: *Telegram Bild*, [o. D.]



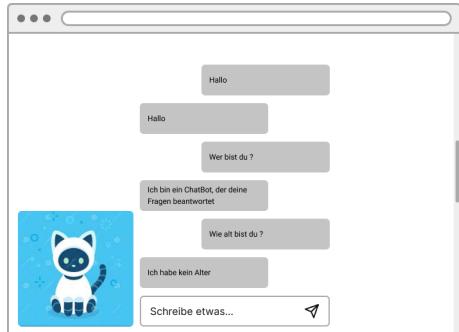
WhatsApp

Hier haben wir uns mehr auf die Ausrichtungen des CHatfensters angeschaut und wie die Sprechblasen ausgerichtet sind. Fast jeder benutzt WhatsApp deswegen haben wir uns die Struktur angeschaut, weil diese viele Benutzer bekannt ist.

Bildquelle: *Tim WhatsApp*, [o. D.]

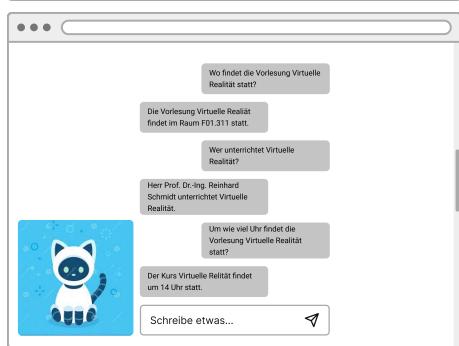
3.2 Version 2

Das sind die neuen Entwürfe des UI Designs. In der neuen Version haben wir uns auf die feineren Elemente konzentriert und uns sogar eine mobile Versionen ausgedacht, weil wir unbedingt "mobilefirst" Programmieren möchten.



Webchat

Hier haben wir einen Beispielchat mit dem Bot dargestellt. In diesem Beispiel haben wir ein Basisgespräch geführt.



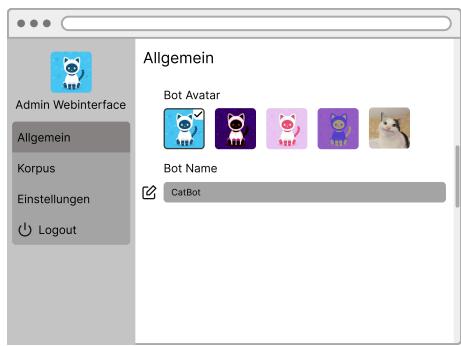
Webchat mit der Hochschuldomäne

Diesmal haben wir konkrete Hochschulfragen gestellt und die Hochschuldomäne dargestellt.



Webchat als mobile Version

Dies ist die mobile Version des Webchats. Wir haben die Darstellung so einfach wie möglich dargestellt. Hier haben wir auch die Hochschulbezogenen Fragen gestellt.



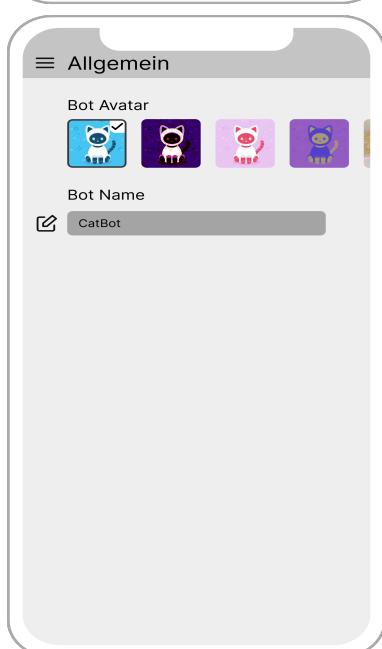
Admin Webinterface: Allgemein

Auf der linken Seite sieht man die Kategorien, die der Admin bearbeiten kann. Im Allgemeinen kann der Admin den Bot Avatar wechseln, diese wird dann mit einem Haken gekennzeichnet. Außerdem kann der Admin den Bot Namen ändern, indem er den Editierbutton drückt.



Admin Webinterface mobil: Allgemein dropdown Menü

In der mobilen Version haben wir die Kategorien, die der Admin bearbeiten kann im dropdown Menü dargestellt.



Admin Webinterface mobil: Allgemein

Die mobile Variante funktioniert genauso wie die Webvariante. Man kann den Bot Avatar wechseln und den ChatBot Namen frei bestimmen.

The screenshot shows the Admin Webinterface with the 'Korpus' section selected. The interface includes a sidebar with 'Admin Webinterface', 'Allgemein', 'Korpus' (which is highlighted in grey), 'Einstellungen', and 'Logout'. The main area is titled 'Korpus' and 'Hochschule'. It lists three questions with their corresponding answers:

- Frage: Wo findet die Vorlesung Virtuelle Realität statt? Antwort: Die Vorlesung Virtuelle Realität findet im Raum F01.311 statt.
- Frage: Wer unterrichtet Virtuelle Realität? Antwort: Herr Prof. Dr.-Ing. Reinhard Schmidt unterrichtet Virtuelle Realität.
- Frage: Um wie viel Uhr findet die Vorlesung Virtuelle Realität statt? Antwort: Der Kurs Virtuelle Realität findet um 14 Uhr statt.

Admin Webinterface: Korpus 00

Im Korpus kann der Admin weitere Domänen, Fragen und Antworten einsehen.

This screenshot shows the same Admin Webinterface as the previous one, but with a new answer added to the first question. The 'Frage' row for 'Wo...' now has two 'Antwort' rows:

- Frage: Wo findet die Vorlesung Virtuelle Realität statt? Antwort: Die Vorlesung Virtuelle Realität findet im Raum F01.311 statt.
- Frage: Wer unterrichtet Virtuelle Realität? Antwort: Herr Prof. Dr.-Ing. Reinhard Schmidt unterrichtet Virtuelle Realität.
- Frage: Um wie viel Uhr findet die Vorlesung Virtuelle Realität statt? Antwort: Der Kurs Virtuelle Realität findet um 14 Uhr statt.

Admin Webinterface: Korpus 01

Mit dem Editierbutton kann er Domänen, Fragen und Antworten hinzufügen und entfernen.

This screenshot shows the Admin Webinterface with a duplicate answer for the first question. The 'Frage' row for 'Wo...' now has three 'Antwort' rows:

- Frage: Wo findet die Vorlesung Virtuelle Realität statt? Antwort: Die Vorlesung Virtuelle Realität findet im Raum F01.311 statt.
- Die Vorlesung findet im Raum F01.311 statt.
- Frage: Wer unterrichtet Virtuelle Realität? Antwort: Herr Prof. Dr.-Ing. Reinhard Schmidt unterrichtet Virtuelle Realität.
- Frage: Um wie viel Uhr findet die Vorlesung Virtuelle Realität statt? Antwort: Der Kurs Virtuelle Realität findet um 14 Uhr statt.

Admin Webinterface: Korpus 02

So kann man wie im Beispiel eine weitere Antwort zu der Frage: "Wo findet die Vorlesung Virtuelle Realität statt?", hinzufügen.

Admin Webinterface: Korpus 03

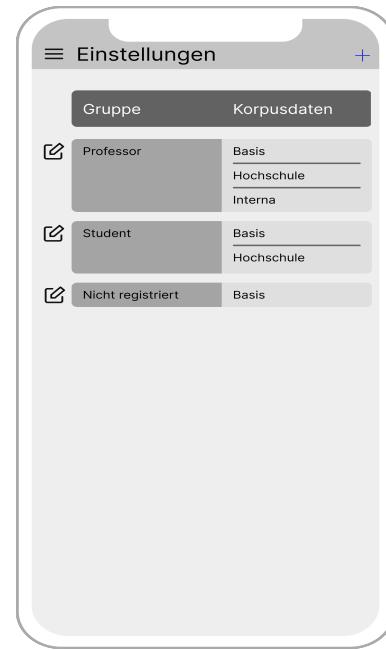
Natürlich kann man auch die hinzugefügte Antwort entfernen. Man klickt auf das Feld und das Feld erscheint rötlich und ein Müllimersymbol entsteht. Wenn man jetzt auf das Eimer-Symbol klickt kann man die Antwort löschen.

Admin Webinterface mobil: Korpus

In der mobilen Version des Korpuses kann man die Elemente genauso editieren wie im Webbrower.

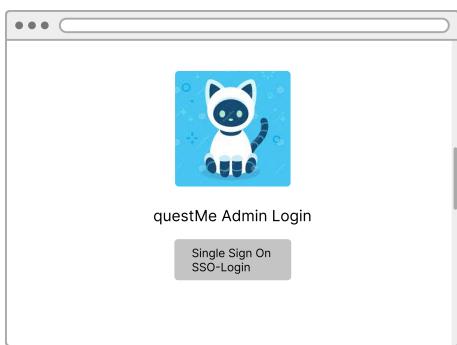
Admin Webinterface: Einstellungen

In Einstellungen kann der Admin seine Gruppen einsehen, hinzufügen und entfernen. Er kann auch die dazugehörigen Korpusdaten sehen und bearbeiten. In den Korpusdaten sind die Domänen der jeweiligen Gruppe eingetragen.



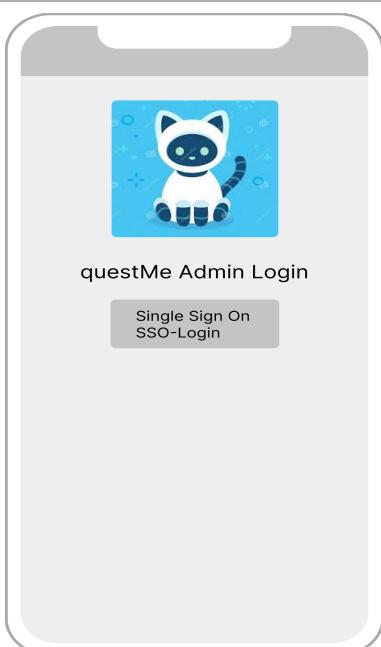
Admin Webinterface mobil: Einstellungen

In der mobilen Version kann man ebenso die gleichen Features nutzen.



Admin Webinterface: Single Sign On

Mit einem Link gelangt der Admin zu der Admin Login Seite, wo er mit Shibboleth sich einloggen kann.

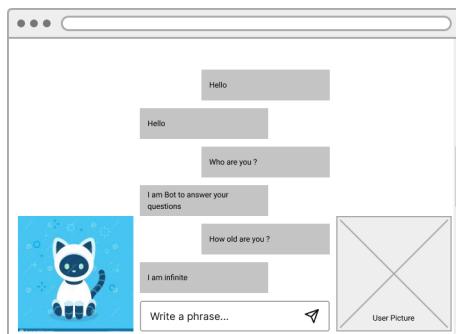


Admin Webinterface mobil: Single Sign On

In der mobilen Version wird die gleiche Prozedur benutzt.

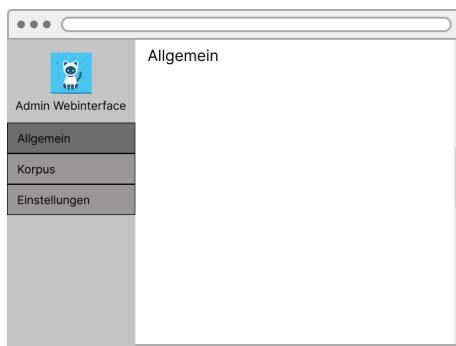
3.3 Version 1

Das sind die älteren Entwürfe des Designs. Bei diesen Entwürfen haben wir uns nur auf die Struktur konzentriert und eine grobe Version erstellt. Uns war es hier hauptsächlich wichtig die Ideen, die wir durch die Recherche aufgenommen haben zu projektiere.



Webchat

In der alten Version unseres Webchat designs haben wir gedacht, auch für den Nutzer ein Profilbild hinzufügen. Diese Idee aber schien zu aufwendig zu sein und wir versuchen zuerst ein Minimal Virable Product zu erschaffen.



Admin Interface: Allgemein

Im Admin Interface kann man auf der linken Seite das Menü sehen. Was aber hier fehlt ist der Logout Button. Wir hatten auch keine richtigen Vorstellungen, was wir einführen möchten.



Admin Interface: Korpus 00

Im Korpus wollten wir schon von Anfang an das Hinzufügen darstellen, wussten aber nicht wie und habe herumprobiert.

Admin Interface: Korpus 01

In diesem Beispiel sieht man ein Basisbeispiel mit einer Frage die im Alltag gestellt wird und die dazugehörigen Antworten. Wie wir diese aber editieren haben wir hier noch nicht gezeigt.

Admin Interface: Korpus 02

Im nächsten Beispiel sieht man eine weitere Frage und die dazugehörigen zwei Antworten. Es ist immer noch nicht bekannt, wie man Antworten und Fragen editiert oder Fragen und Antworten von verschiedenen Domänen bearbeitet.

Admin Interface: Admin Login

Hier haben wir nur einen klassischen Login Eingabefeld dargestellt, weil wir uns nicht klar waren, wie es mit der Authentifizierung und dem Admin Login funktioniert.

3.4 Kriterien für Usability

Unsere Kriterien für Usability, die uns wichtig sind und für unser User Interface essentiell sind, werden wir hier kurz erläutern.

1. Responsive Webdesign

Wir möchten unseren ChatBot auf allen Geräten ohne Probleme ausführen lassen. Das heißt, dass wir auch auf mobilen Geräten unsere Software im Browser laufen lassen wollen. Die mobile Seite sollte dann auch auf die wichtigsten Elemente beschränkt werden, damit der Benutzer es leichter hat die Icons treffsicher mit ihren Fingern zu benutzen.

2. Gute Lesbarkeit

Unsere Anwendung sollte leicht zu lesen sein, weil das Lesen auf dem Bildschirm grundsätzlich schwieriger ist. Deswegen sollten wir auf jeden Fall auf Textgröße und

Kontrastreiche Farben achten. Die Textgröße sollte mindestens 12pt haben. Auch sollten wir lange Textblöcke und Schachtelsätze vermeiden.

3. Gute Navigation

Eine übersichtliche und eine verständliche Navigation ist das wichtigste bei einer Webseite. Eine gute Navigation verhindert Verwirrung und Unterstützt den Benutzer zu seinem Ziel zu gelangen. Bei der Navigation sollte man darauf achten, dass alle Verlinkungen funktionieren.

4. Schnelle Ladezeiten

Die Webseite sollte ihre Inhalte schnell laden und keine großen Verzögerungen aufzeigen. Sie sollte bei ungefähr drei Sekunden liegen, um keine User zu verlieren.

5. Interessantes Design

Eine Konsistente Einhaltung von bestimmten Farben ist sehr wichtig. Der erste Eindruck von einer Webseite zeigt schon, ob die User die Webseite benutzen möchten oder nicht. So können Benutzer durch Bilder mit schlechter Qualität oder zu grellen Farben abgeschreckt werden. Pop-Ups sollten in Grenzen gehalten werden oder vermieden werden.

3.5 Usability Test

Hier beschreiben wir welche Methode wir zum Testen unserer Usability Kriterien benutzen möchten.

Remote User Testing

Zuerst plannen wir was wir testen möchten und dann wie wir testen möchten. In unserem Fall haben wir uns für remote testing entschieden. Als nächstes überlegen wir uns besondere Tasks, die der User absolvieren muss, um zu erkennen, ob unsere Kriterien eingehalten werden und was wir Verbessern sollten. Die Szenarien sollten so einfach und realistisch sein, dass der Benutzer es leicht durchführen kann. Dann müssen wir auch Tester finden, welche in unserer Zielgruppe passen.

4 User Stories

In diesem Kapitel haben wir unsere User Stories gesammelt.

4.1 Struktur

Als <Akteur> möchte ich <Funktion>, um <Nutzen> zu erreichen.

4.2 User Stories Version 1

Hier listen wir unsere ersten Ideen auf, die wir erfüllen möchten.

Chatfenster

1. Als Nutzer möchte ich einen Sendebutton betätigen, um eine Nachricht abzuschicken.
2. Als Nutzer möchte ich ein Eingabefeld für meine Fragen haben, um eine Antwort zu erhalten.
3. Als Nutzer möchte ich einen Chatfenster haben, um eine Nachricht zu erhalten.
4. Als Nutzer möchte ich das Icon vom Bot sehen, um die Nachricht des Bots von meiner zu unterscheiden.
5. Als Nutzer möcht ich verschieden farbige Sprechblasen sehen, um die Nachricht des Bots von meiner zu unterscheiden.
6. Als Nutzer möchte ich hingewiesen werden, wo ich zu Schreiben habe, um eine Nachricht verfassen zu können.

Admin Interface: Allgemein

1. Als Admin möchte ich ein dropdown Menü oder etwas gleichwertiges haben, um auf mein Allgemein zu wechseln.
2. Als Admin möchte ich verschiedene Icons zur Auswahl haben, um mein Bot Avatar zu wechseln.
3. Als Admin möchte ich ein Haken als Bestätigung sehen, um zu sehen, welchen Bot Avatar ich gewählt habe.
4. Als Admin möchte ich ein Editierbutton haben, um mein ChatBot Namen zu ändern.
5. Als Admin möchte ich ein Eingabefeld benutzen können, um mein ChatBot Namen eingeben zu können.

6. Als Admin möchte ich eine gefärbte Fläche sehen, um ein Eingabefeld erkennen zu können.
7. Als Admin möchte ich die ausgewählte Fläche in einer anderen Farbe sehen, um zu erkennen ob ich im Allgemein bin.

KeyCloak

1. Als Admin möchte ich mich in KeyCloak einloggen können, um es zu verwalten.
2. Als Admin möchte ich mich in das Admin-Interface einloggen können, um die Einstellungen des Chatbots zu verwalten.
3. Als Hochschulangehöriger möchte ich mich mit dem Shibboleth SSO der Hochschule einloggen, um relevante Daten mitzuteilen.
4. Als Admin möchte ich schnellen Zugriff auf das KeyCloak-Webinterface über das Admin-Interface, um Zeit zu sparen.
5. Als Admin möchte ich neue Nutzergruppen erstellen, um zielgerichteter Fragen beantworten zu können.
6. Als Admin möchte ich einer Nutzergruppe einen neuen Fragensatz zuweisen, um die möglichen Fragen für diese Gruppe zu erweitern.
7. Als Admin möchte ich einen, zu einer Nutzergruppe zugewiesenen, Fragensatz entfernen, um möglichen Fragen für diese Gruppe einzuschränken.
8. Als Admin möchte ich Nutzer verwalten, um bei Bedarf Änderungen vorzunehmen.
9. Als Admin möchte ich die Login-Seite anpassen, um sie nach meinen Vorstellungen zu ändern.

Node.js Allgemein

1. Als Nutzer möchte ich eine bidirektionale Kommunikation zwischen dem Client und Server, um direkt mit dem Bot kommunizieren zu können.
2. Als Nutzer möchte ich, dass der ChatBot meinen Kontext versteht, um mit dem Bot nach Kontext zu chatten.
3. Als Administrator möchte ich die Möglichkeit den Korpus des ChatBots persistent zu speichern, um auf den Korpus zuzugreifen zu können.
4. Als Nutzer möchte ich, dass der ChatBot über eine Webadresse erreichbar ist, um mit dem ChatBot online zu kommunizieren.

Admin Interface: Korpus

1. Als Administrator möchte ich eine Liste mit allen Fragen und Antworten, um einen Überblick über den Korpus zu haben.
2. Als Administrator möchte ich einen Button, der einen neuen Eintrag hinzufügt, um neue Fragen und Antworten hinzuzufügen zu können.
3. Als Administrator möchte ich eine Möglichkeit zum Hinzufügen von Fragen, um neue Fragen hinzuzufügen zu können.
4. Als Administrator möchte ich eine Möglichkeit zum Hinzufügen von Fragen, um neue Fragen hinzuzufügen zu können.
5. Als Administrator möchte ich eine Möglichkeit zum Hinzufügen von Antworten, um neue Antworten hinzuzufügen zu können.
6. Als Administrator möchte ich eine Möglichkeit zum Entfernen von Fragen, um Fragen entfernen zu können.
7. Als Administrator möchte ich eine Möglichkeit zum Entfernen von Antworten, um Antworten entfernen zu können.
8. Als Administrator möchte ich eine Möglichkeit einen Eintrag „Fragen und Antworten“ bearbeiten zu können, um den Eintrag zu ändern.

5 Technologien

In diesem Kapitel soll alles zum Thema Technologien zusammengefasst sein. Zusätzlich sollen Schaubilder und Diagramme das Verständnis für die Technologien erleichtern. Sowie Beschreibungen und Erklärungen zu der Wahl der einzelnen Technologien.

5.1 Entscheidung für die Technologien

Bei der Wahl der Technologien haben wir nach Möglichkeit LTS Versionen gesucht. Damit wir eine Applikation erstellen können, die für lange Zeit Sicherheitsupdates bekommt. Zusätzlich haben wir als Ziel, dass die Software als ingesamtes Paket sehr lange stabil läuft und dadurch Ausfälle minimiert werden.

5.2 Technologie Versionen

Hier sollen die einzelnen Technologien im Einzelnen beschrieben werden. Sowie ihre geplante Funktion in unserem Projekt.

	Node.js 16.13.0 LTS
	Angular 13.0.1
	Socket.io 4.3.2
	MongoDB Community Edition 5.0.3
	Keycloak 15.0.2
	NLP.js 4.22.2 (optional)

Tabelle 1: Technologie Liste

5.2.1 Node.js 16.13.0 LTS

Als Basis für Angular, den Chatbot und Keycloak nutzen wir eine LTS Version von Node.js. Auf dem Node.js Server verwenden wir zusätzlich die Express Erweiterung, um eine Kommunikation zu den Servern herstellen zu können. Für uns war sehr wichtig, dass wir eine sehr stabile Basis für die Anwendung, die wir entwickeln haben. Deswegen haben wir uns ebenfalls informiert, ob alle geplanten oder in Frage kommenden Technologien mit der LTS Version kompatibel sind. Nach unserem derzeitigen Stand empfehlen die Entwickler der Bibliotheken und Frameworks eine LTS Version von Node.js zu verwenden. Bildquelle: *Node.js Icon*, [o. D.]

5.2.2 Angular 13.0.1

Wir verwenden Angular für die Frontend-Entwicklung. Angular ist die Basis, um das Chat- und das Admin-Interface zu erstellen. Um unsere UIs noch effektiver zu gestalten werden wir vereinzelt auf Angular Material zurückgreifen. Dort gibt es zu häufig genutzten UI Komponenten "Code Snippets", die ausführlich getestet wurden. Bildquelle:*Angular Icon*, [o. D.]

5.2.3 Socket.io 4.3.2

In unserem Projekt nutzen wir Socket.io, um eine bidirektionale Kommunikation zwischen dem Bot und der Benutzeroberfläche herzustellen. Es ermöglicht uns, dass ein User mit dem Chatbot ohne merkbare Verzögerung chatten kann. Bildquelle:*Socket.io Icon*, [o. D.]

5.2.4 MongoDB Community Edition 5.0.3

Wir haben uns für die Datenbank mongoDB entschieden, da es ein sehr einfaches Datenbank Modell bereitstellt. MongoDB ist eine NoSql Datenbank wodurch auch der Verwaltungsaufwand der Datenbank minimiert wird. In der Datenbank wird der Korpus des Chatbots gespeichert. Bildquelle:*MongoDB Icon*, [o. D.]

5.2.5 KeyCloak 15.0.2

In unserem Projekt verwenden wir KeyCloak, damit sich der Admin sicher in das Admin-Webinterface einloggen kann. Zusätzlich werden wir das Shibboleth Single-Sign-On Verfahren von der Hochschule integrieren. Damit Hochschulangehörige die Möglichkeit haben mit ihrem Hochschul Account sich einloggen zu können. Dadurch können wir zusätzlich begrenzen welche Personengruppen auf welche Daten Zugriff haben. Bildquelle:*KeyCloak Icon*, [o. D.]

5.2.6 RegEx

Wir haben in unserem Projekt anfangs aus Sicherheitsgründen RegEx eingeplant. Um ein Scheitern des Projektes zu verhindern, falls NLP.js sich nicht in unser Projekt integrieren lassen sollte. Somit der Chatbot dennoch Begriffe und Sätze verstehen kann. Nach unserem derzeitigen Stand brauchen wir nicht auf eine RegEx Implementation zurückzugreifen, da NLP.js sich sehr gut in unser Projekt integrieren lässt. Bildquelle:*Regex Icon*, [o. D.]

5.2.7 NLP.js 4.22.2 (optional)

Damit wir NLP.js nutzen können benötigen wir eine LTS Version von Node.js. Diese Begrenzung haben wir zusätzlich in die Technologie Entscheidung einbezogen. Wir

haben den NLP.js ausführlich getestet und werden es als Basis für den Chatbot verwenden. Indem NLP.js ab der Version 4 und höher modular ist. Wird uns ermöglicht leichter eigene Module/Plugins für das Framework zu schreiben. Dadurch können wir einen ChatBot mit verbesserter Spracherkennung gestalten. Bildquelle:*NLP Icon*, [o. D.]

5.3 Technologie Vergleich

In diesem Bereich wollen wir unsere Design Entscheidung für eine Technologie durch einen Vergleich darstellen. Dafür haben wir für die "Themen Angular vs Vue.js" und "MongoDB vs PostgreSQL" eine Tabelle erstellt.

5.3.1 Angular vs Vue.js

	Angular	Vue.js
Framework, Library, Platform	Entwicklungsplattform (Development platform)	Progressive Framework
Gründer	Google	ehemaliger Google Mitarbeiter
Technology Typ	MVC Framework	MVVM Framework
Programmiersprache	TypeScript	JavaScript
Performance	niedrig	hoch
Größe	500 kB	80 kB
Lernkurve	Eine steile Lernkurve	Eine geringe Lernkurve
Dokumentation	vorhanden	vorhanden
Datenbindung	Bi-directional	Bi-directional
Rendering	beim Client	beim Server
Code reuse	möglich	Ja, CSS und HTML
Skalierbarkeit	sehr hoch	hoch
Testbarkeit	mit einem Tool	mehrere Tools benötigt
Vollständige Web App	Kann als standalone Basis verwendet werden	benötigt Third Party Tools
Lizenz	MIT License	MIT License

Tabelle 2: Angular vs Vue.js

5.3.2 MongoDB vs PostgreSQL

	MongoDB	PostgreSQL
Primäres Datenbankmodell	Dokumentation	Relationales DBMS
Entwickler	MongoDB, Inc	PostgreSQL Global Development Group
Datenschema	Schemafrei (NoSQL)	Ja
Programmiersprachen	JavaScript + 28 weitere	JavaScript + 9 weitere
Query Language	MQL	SQL
Maximale Dateigröße	16 MB	1 GB
Lernkurve	Eine geringe Lernkurve	Eine geringe Lernkurve
Dokumentation	vorhanden	vorhanden
Skalierbarkeit	sehr hoch	hoch
Lizenz	Open Source	Open Source

Tabelle 3: MongoDB vs PostgreSQL

5.4 Technologie Diagramme

Hier werden die Zusammenhänge der Technologien in Form von verschiedenen Diagrammen vorgestellt.

5.4.1 Komponentendiagramme

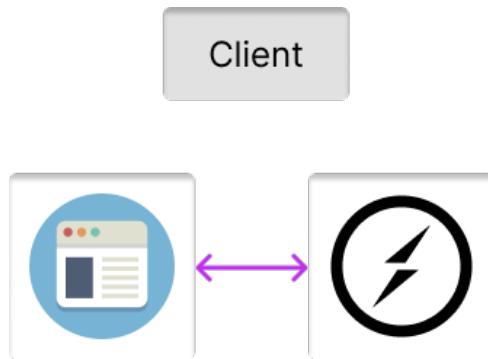


Abbildung 1: Komponentendiagramm Client

Hier wird die Client Seite bildlich dargestellt. Man sieht, dass der Webbrowseer durch die Socket.io Client deployed wird und dadurch wird dann eine Beziehung zur Serverseite aufgebaut. (Siehe Abbildung 1)

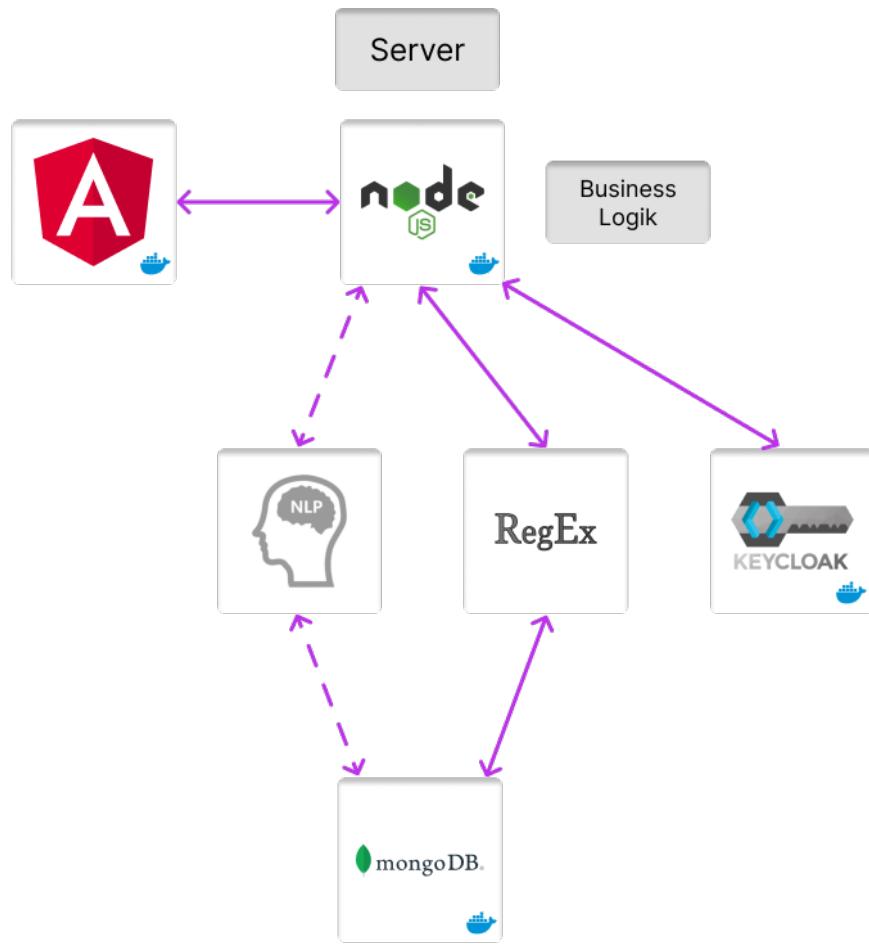


Abbildung 2: Komponentendiagramm Server

In der Serverseite wird dann durch den Socket.io Server die Verbindung zum Client aufrecht gehalten. Im Server befindet sich Angular, node.js, KeyCloak und die Datenbank mongoDB und haben jeweils einen eigenen Dockercontainer. Die Erkennung von der eingegebenen Sprache möchten wir zunächst mit RegEx ermöglichen, um das Minimal Viable Product hinzubekommen. Optional dann mit NLP (Natural Language Processing) erweitern.

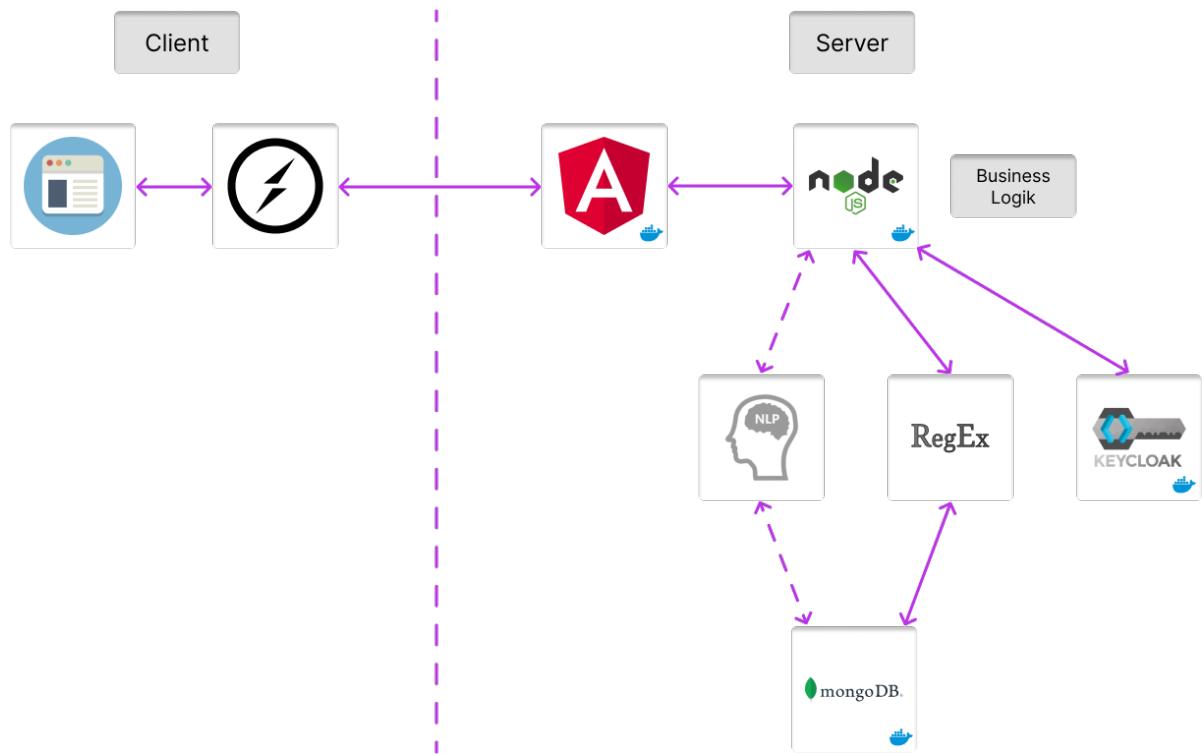


Abbildung 3: Komponentendiagramm v1.2

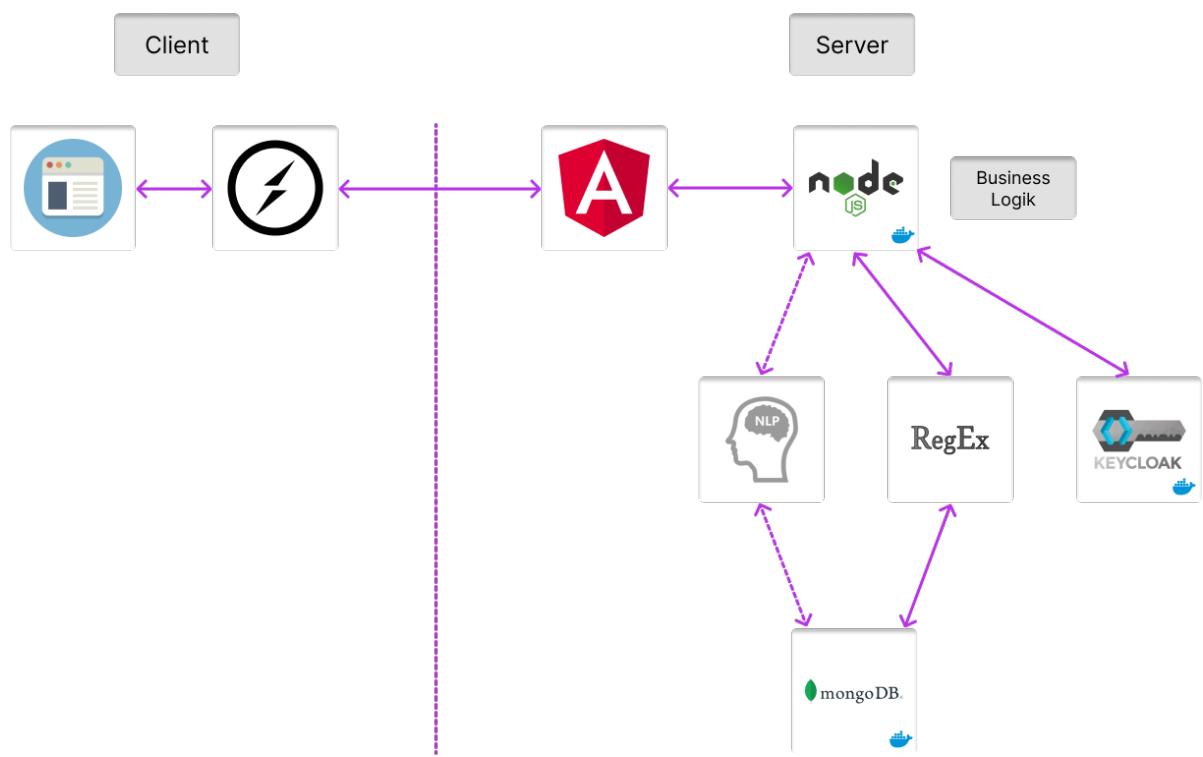


Abbildung 4: Komponentendiagramm v1.1

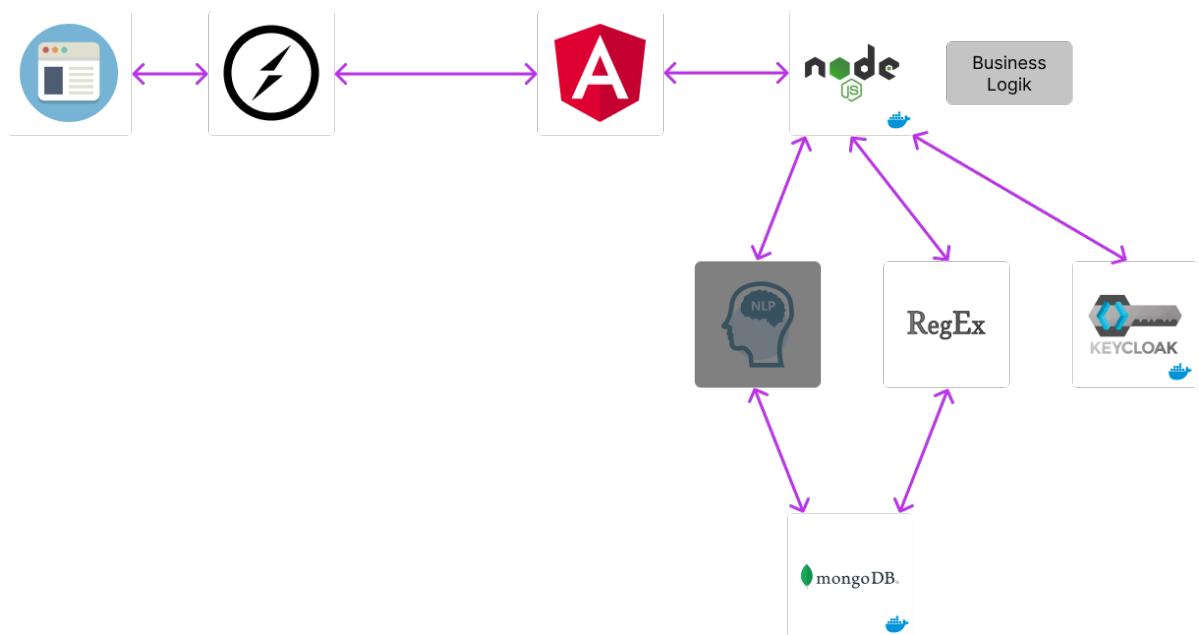


Abbildung 5: Komponentendiagramm v1.0

6 Meilensteine

Recherche	13.10.21
Zwischenpräsentation	22.10.21
Implementation	25.10.21
MVP	03.12.21
Endpräsentation	14.01.22
Enddokumentation	14.01.22

Tabelle 4: Meilenstein Liste

In der obigen Tabelle 4 sind alle einzelnen Meilensteine aufgelistet. Jeder Meilenstein wird noch etwas genauer erklärt.

6.1 Recherche 13.10.21

Während der Recherche haben wir alle Informationen gesammelt, die wir für das Projekt benötigen. Die Themen Node.js, Keycloak, Angular, Socket.io und NLP waren vorrangige Themen, um Risiken zu minimieren. Deswegen haben wir uns in dieser Phase möglichst ausführlich informiert und Bücher, Webadressen und weitere Materialien besorgt. Außerdem haben wir nach Möglichkeit alle Unklarheiten geklärt.

6.2 Zwischenpräsentation 22.10.21

Wir haben eine Woche früher (12.10.21) angefangen alle relevanten Themen zu sammeln, um Materialien für die Präsentation zu haben. Damit die Präsentation sehr interessant für alle Teilnehmer ist, haben wir die wichtigsten Themen optisch ansehbar gestaltet. In unserer Zeitplanung ist auch die praktische Übung der Folien im Team eingeplant.

6.3 Implementation 25.10.21

In der Implementierung wollen wir die recherchierten Materialien umsetzen und praktische Erfahrung sammeln. Während wir versuchen alle relevanten Informationen in einen MVP umzusetzen. Zusätzlich wird in dieser Phase ein Teil der Recherche in das L^AT_EX-Format übertragen.

6.4 MVP 03.12.21

Der MVP ist unser angestrebtes Ziel, dass wir ein Produkt zum Präsentieren haben. Während wir unser angesammeltes Wissen in die Praxis umsetzen versuchen wir frühzeitig ein funktionierendes Produkt mit den Mindestanforderungen umzusetzen.

6.5 Endpräsentation 14.01.22

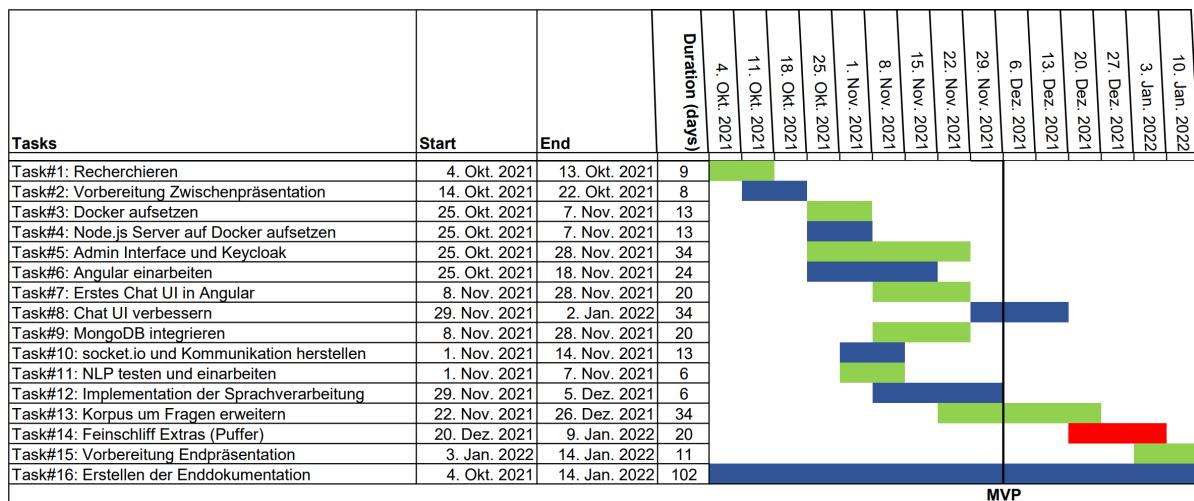
Für die Endpräsentation ist eine Woche früher (07.01.22) der Beginn der Erstellung der Präsentation eingeplant. Ziel ist hierbei, dass ein Prototyp mit interessanten Features und Funktionen vorgestellt werden kann.

6.6 Enddokumentation 14.01.22

Für die Enddokumentation werden wir ab dem Start der Implementation anfangen, alle wichtigen Informationen zu dokumentieren. Sehr wichtig ist hierbei für unser Team, dass wir alle kontinuierlich wichtige Themen zu unserem Projekt dokumentieren. Für unsere Dokumentation wählen wir das L^AT_EX-Format, da es uns hilft besser kooperativ über Gitlab zu arbeiten.

7 Zeitmanagement

7.1 Gantt-Diagramm



8 Risikoanalyse

Risiko	Eintritts-wahrscheinlichkeit	Auswirkung	Maßnahme
Team schafft es nicht schnell genug Typescript zu lernen	niedrig	hoch	Alle Teammitglieder beginnen frühzeitig sich mit Typescript zu beschäftigen
Team schafft es nicht Keycloak zu integrieren	niedrig	hoch	Alle Teammitglieder schauen sich rechtzeitig die Einführungsvideos von Herr Rößler zu Keycloak an
Das Team hat Schwierigkeiten das Userinterface in Angular zu entwickeln	niedrig	hoch	Das Team greift auf gut bewährte Designs zurück
Das Team hat Schwierigkeiten eine Verbindung mit socket.io herzustellen	niedrig	hoch	Das Team informiert sich rechtzeitig auf der Socket.io Website, wie eine Verbindung aufgebaut wird
Das Admin Interface lässt sich nicht flexibel genug anpassen	niedrig	niedrig	Das Team informiert sich rechtzeitig welche Möglichkeiten es in das UI einbauen möchte, um Flexibilität zu garantieren
Der ChatBot antwortet stark verzögert	niedrig	mittel	Das Team muss mit-einplanen, dass der ChatBot in kleine saubere Module aufgeteilt wird
Hohe Latenz durch alle Komponenten	unwahr-scheinlich	mittel	Das Team muss den ChatBot testen, um z.B. Endlosschleifen zu verhindern

Risiko	Eintritts-wahrscheinlichkeit	Auswirkung	Maßnahme
Der ChatBot hat Schwierigkeiten Sätze zu verstehen	mittel	mittel	Das Team muss bei einem Regex Ansatz mehrere Regex Befehle vordefinieren, um ein großes Spektrum abzudecken
Die Hardware des Kunden ist nicht kompatibel mit der Software	niedrig	hoch	Das Team muss frühzeitig mit dem Kunden klären für welche Hardware der ChatBot entwickelt werden soll
Das Team scheitert einen MVP zu entwickeln	niedrig	hoch	Das Team muss sehr früh mit der Implementierung beginnen und ausführlich genug recherchieren
Das Team scheitert rechtzeitig genug alle Technologien zu lernen	niedrig	hoch	Das Team recherchiert frühzeitig und versucht für alle möglichen Probleme Lösungen zu finden

Literatur

Angular Icon, [o. D.] **online**[besucht am 2021-11-11]. Abger. unter: <https://angular.io/assets/images/logos/angular/angular.svg#>.

Blackberry-messenger-live-free, [o. D.] **online**[besucht am 2021-11-11]. Abger. unter: https://www.google.com/search?sa=G&hl=en_GB&tbs=simg:CAQSigIJhkYKQeZpo2ga_1gELELCMpwga0go4CAQSFNwjrhOGH5Yp4ArFNOYozArhA70rGhqMV02g86-J4LEEe7o2NhCoXzI40s0avczyMSAFMAQMCxCOrv4IGgoKCAGBEgS8yhvGDAsQne3BCRqeAQobCgh2ZXJ0aWNhbNqliPYDCwoJL2EvNGhoM3AwChYKA2RvdNqliPYDCwoJL20vMDI3Y3RnChsKCGxhbmd1YWdl2qWI9gMLCgkvai8yc2hfeTQKKwoXdGVjaG5vbG9neSBhcHBsaWNhdGlvbnPapYj2AwwKCi9tLzBjMHNxOXQKHQoKaG9yaXpvnRhbNqliPYDCwoJL2EvMm1xdnpjDA&sxsrf=AOaemvIqBE_CBQnc5LbSVstD8aZ_tE0m5A:1635361908652&q=vertical&tbo=isch&ved=2ahUKEwjnj5PtpevzAhVESvEDHdixBtAQwg4oAHoECAEQMg&biw=958&bih=967&dpr=1#imgrc=WbybuUumcvn6WM

Briar, [o. D.] **online**[besucht am 2021-11-11]. Abger. unter: https://www.google.com/search?sa=G&hl=en_GB&tbs=simg:CAESigIJsqta7umNdmEa_1gELELCMpwga0go4CAQSFNwj4QOhHYYfxTTAPuYokibTFM0LGhooxkKxDJXZDHqsS0d-jgch2_1GBemxZC9KE5yAFMAQMCxCOrv4IGgoKCAGBEgRtBijYDAsQne3BCRqeAQobCghsYW5ndWFnZdqliPYDCwoJL2ovMnNoX3k0ChsKCHZ1cnRpY2Fs2qWI9gMLCgkvYS80aGgzcDAKKwoXdGVjaG5vbG9neSBhcHBsaWNhdGlvbnPapYj2AwwKCi9tLzBjMHNxOXQKHQoKaG9yaXpvnRhbNqliPYDCwoJL2EvMm1xdnpjChYKA2RvdNqliPYDCwoJL20vMDI3Y3RnDA&sxsrf=AOaemvKD2G5sh2A-CGziL7iecZAZ4Im_gg:1635365049125&q=Briar&tbo=isch&ved=2ahUKEwjKxdLGsevzAhVLN0wKHTKuDcsQjJkEegQIAhAC&biw=958&bih=967&dpr=1#imgrc=OZdYEumydcv8_M

Chatbot-cost-calculator, [o. D.] **online**[besucht am 2021-11-11]. Abger. unter: <https://risingmax.com/blog/ai-based-chatbot-cost-calculator/>.

Cleverbot, [o. D.] **online**[besucht am 2021-11-11]. Abger. unter: <https://www.cleverbot.com/>.

KeyCloak Icon, [o. D.] **online**[besucht am 2021-11-11]. Abger. unter: <https://www.katacoda.com/sebastienblanc/avatar>.

MongoDB Icon, [o. D.] **online**[besucht am 2021-11-11]. Abger. unter: https://webimages.mongodb.com/_com_assets/cms/kpo5kblefbjq79065-Horizontal_Default.svg?auto=format%252Ccompress.

NLP Icon, [o. D.] **online**[besucht am 2021-11-11]. Abger. unter: <https://ecomschool.uk/wp-content/uploads/2020/01/NLP.png>.

Node.js Icon, [o. D.] **online**[besucht am 2021-11-11]. Abger. unter: <https://cdn.freebiesupply.com/logos/thumbs/2x/nodejs-1-logo.png>.

Regex Icon, [o. D.] **online**[besucht am 2021-11-11]. Abger. unter: https://upload.wikimedia.org/wikipedia/commons/thumb/d/d3/Toolbaricon_RegEx.svg/1280px-Toolbaricon_RegEx.svg.png.

Socket.io Icon, [o. D.] **online**[besucht am 2021-11-11]. Abger. unter: <https://socket.io/images/logo.svg>.

Telegram Bild, [o. D.] **online**[besucht am 2021-11-11]. Abger. unter: <https://www.google.com/search?hl=en&tbs=simg&sa=G&q=telegram+bild>.
[besucht am 2021-11-11]. Abger. unter: <https://www.google.com/search?hl=en&tbs=simg&sa=G&q=telegram+bild>.

Tim WhatsApp, [o. D.] **online**[besucht am 2021-11-11]. Abger. unter: <https://www.google.com/search?hl=en&tbs=simg&sa=G&q=tim+whatsapp>.
[besucht am 2021-11-11]. Abger. unter: <https://www.google.com/search?hl=en&tbs=simg&sa=G&q=tim+whatsapp>.