

# DEV-Chat

## Projektdokumentation

### Projektarbeit

über die Theoriephase des zweiten Studienjahrs

an der Fakultät für Technik  
im Studiengang Informationstechnik

an der DHBW Ravensburg  
Campus Friedrichshafen

von Gruppe 8, mit den Teilnehmern  
Nico Bayer, Johannes Brandenburger, Lukas Braun, Jan Brutscher, Philipp Patzelt,  
Henry Schuler

23. Februar 2022

Bearbeitungszeitraum: Oktober 2021 - März 2022  
Kurs: TIT20  
Hochschule: DHBW Friedrichshafen

# Inhaltsverzeichnis

<b>1 Planung</b>	<b>1</b>
1.1 Projekthandbuch . . . . .	1
1.1.1 Anzuwendendes Prozessmodell . . . . .	1
1.1.2 Projektorganisation . . . . .	1
1.1.3 Qualitätsmanagement Maßnahmen . . . . .	1
1.1.4 Konfigurationsmanagement Maßnahmen . . . . .	2
1.1.5 Technische Rahmenbedingungen . . . . .	2
1.2 Lastenheft . . . . .	3
1.2.1 Allgemeine Beschreibung . . . . .	3
1.2.2 Anforderungen . . . . .	3
1.2.3 Fachliches Umfeld . . . . .	3
1.2.4 Ausblick . . . . .	3
1.2.5 Erweiterungsmöglichkeiten (optional) . . . . .	3
1.3 Arbeitspaketplan . . . . .	4
1.4 Personalplanung . . . . .	4
1.5 Zeitplan . . . . .	4
1.6 Auswahl kritischer Technologien . . . . .	4
1.6.1 Aktualisierung in der Implementierungsphase am 25.02.2022 . . . . .	5
<b>2 Definition</b>	<b>7</b>
2.1 Use Case Diagramm . . . . .	7
2.2 Use Case Beschreibung . . . . .	8
2.3 Datenbankmodell . . . . .	12
2.3.1 Entity Relationship Model (ERM) . . . . .	12
2.3.2 Data Dictionary . . . . .	13
2.4 HMI . . . . .	15
2.4.1 Seitenhierarchy . . . . .	15
2.4.2 Prototyp Desktop . . . . .	16
2.4.3 Prototyp Mobile . . . . .	23
2.5 DatenFlussdiagramm . . . . .	36
<b>3 Entwurf</b>	<b>37</b>
3.1 Auswahl der Klassenbibliotheken/Frameworks . . . . .	37
3.2 Design Patterns für relevante Problemstellungen . . . . .	37
3.3 Definition der Software-Komponenten . . . . .	38

<b>4 Implementierung</b>	<b>39</b>
4.1 Code Conventions . . . . .	39
4.2 IT-Security . . . . .	39
4.3 Frontend Backend Verbindung . . . . .	39
4.4 Auswertung der Sprints . . . . .	40
4.4.1 Sprint 1 (KW 3+4) . . . . .	40
4.4.2 Sprint 2 (KW 5+6) . . . . .	40
<b>5 Test</b>	<b>42</b>
<b>6 Abnahme</b>	<b>43</b>
<b>7 Betrieb</b>	<b>44</b>
7.1 Zeitliche Dokumentation . . . . .	45
7.1.1 02.11.2021 . . . . .	48
7.1.2 03.11.2021 . . . . .	48
7.1.3 12.11.2021 . . . . .	48
7.2 18.11.2021 . . . . .	48
7.3 23.11.2021 . . . . .	48
7.4 24.11.2021 . . . . .	49
7.5 06.12.2021 . . . . .	49
7.6 19.12.2021 . . . . .	49
7.7 10.01.2021 . . . . .	49
7.8 14.01.2021 . . . . .	49
7.9 31.01.2021 . . . . .	49
7.10 01.02.2021 . . . . .	49
7.11 07.02.2021 . . . . .	50
7.12 11.02.2021 . . . . .	50
7.13 16.02.2021 . . . . .	50
7.14 21.02.2021 . . . . .	50

# 1 Planung

## 1.1 Projekthandbuch

### 1.1.1 Anzuwendendes Prozessmodell

Bei diesem Projekt wird eine modifizierte Version von Scrum verwendet. Der Product Owner und der Scrum Master umfassen vollständig die Aufgaben ihrer Rollen, aber sind zusätzlich auch im Entwickler-Team. Grund hierfür ist, dass es wegen begrenzter humarer Ressourcen ansonsten zu wenig Entwickler geben würde. Somit gibt es jeweils einen mit entwickelnden Product Owner und Scrum Master und vier ausdrücklichen Entwicklern.

### 1.1.2 Projektorganisation

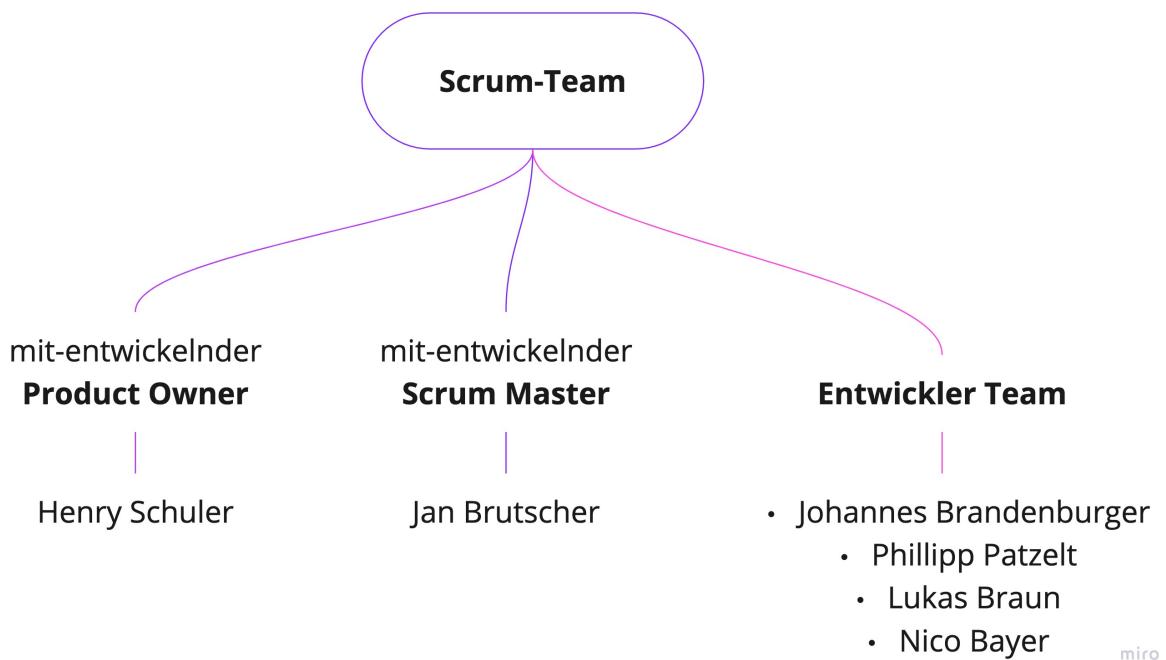


Abb. 1.1: Rollenverteilung

### 1.1.3 Qualitätsmanagement Maßnahmen

Unsere Qualität wird von nicht-Programmierern aus unserem Team durch festgelegte Tests ermittelt. Am Ende der Durchführungsphase wird eine mit ausgewählten Probanden eine geschlossene Beta durchgeführt, welche ebenfalls über die Testkriterien informiert werden. Die Qualität der Tests wird über Muss- und Kann-Kriterien überprüft.

Folgende Kriterien sind entscheidend.

- Funktionalität des Chatraums
- Funktionalität des Logins
- Funktionalität der Benutzerinteraktionen (Erstellen, Login)

Diese Kriterien werden auf einer Skala von 1-10 bewertet und müssen begründet werden. Aufgrund dieses Feedbacks von mehreren unabhängigen Probanden sowie unseren eigenen nicht-Programmierern kann eine maximale Qualität gewährleistet werden.

Eine Zusätzliche Maßnahme zur Sicherung der Qualität ist der sogenannte Bug Befehl. Mit Hilfe des Bug Befehls, können Fehler innerhalb der Website gemeldet werden. Diese können im Anschluss von Administratoren eingesehen und bearbeitet werden.

#### 1.1.4 Konfigurationsmanagement Maßnahmen

Die agile Planung im erweiterten Scrum erfolgt in Microsoft Azure DevOps, hier ist der Backlog angelegt und erfolgt die Sprint Planung. Meetings werden auf einem eigenen Discord Server durchgeführt. Die Dokumentation wird in Overleaf, einem Online LaTeX Tool geschrieben.

Die Versions- und Releaseverwaltung erfolgt in einem GitHub Repository unter dem Git-Branching-Modell Gitflow. Gitflow sieht zwei Branches vor, um den Verlauf des Projekts aufzuzeichnen. Der main-Branch enthält den offiziellen Release-Verlauf und der develop-Branch dient als Integrations-Branch für Features. Der develop-Branch enthält den kompletten Versionsverlauf des Projekts, während der main-Branch eine verkürzte Version enthält.

Jedes neue Feature sollte sich auf seinem eigenen Branch befinden, der zu Sicherungs-/Zusammenarbeitszwecken zum zentralen Repository gepusht werden kann. Ein neuer feature-Branch wird aus dem develop-branch gemitigt. Wenn ein Feature fertig ist, wird es zurück in den develop-Branch gemitigt. Features sollten niemals direkt mit dem main-Branch interagieren.

Wenn der develop-Branch genügend Features für ein Release enthält, wird vom develop-Branch ein release-Branch geforkt. Damit beginnt der neue Release-Zyklus. In diesem Branch sollten ab diesem Punkt keine neuen Features mehr hinzugefügt werden, sondern nur Bugfixes und ähnliche Release-orientierte Änderungen. Ist er zur Auslieferung bereit, wird der release-Branch in den main-Branch gemitigt und mit einer Versionsnummer getaggt. Zusätzlich wird der release-Branch in den develop-Branch gemitigt.

Maintenance- bzw. hotfix-Branches eignen sich für schnelle Patches von Produktions-Releases. Sie werden aus dem main-Branch geforkt. Sobald der Bugfix abgeschlossen ist, wird er sowohl in den main- als auch in den develop-Branch (oder den aktuellen release-Branch) gemitigt.

#### 1.1.5 Technische Rahmenbedingungen

**Umgang mit Open-Source** Das Produkt wird nach der Fertigstellung des Projekts in Betrieb genommen und unter einer Open-Source-Lizenz veröffentlicht.

## 1.2 Lastenheft

### 1.2.1 Allgemeine Beschreibung

Ein Chatportal mit mehreren Chatrooms, denen man über einen Schlüssel beitreten kann. Die Oberfläche soll dabei schlicht im Informatik/Konsolen Stil gehalten werden und alle Operationen werden über eine Kommandozeile ausgeführt. Über die Kommandozeile lassen sich Befehle ausführen, um zum Beispiel Direktnachrichten zu senden.

### 1.2.2 Anforderungen

- Die Lösung muss plattformunabhängig und leicht im Browser über eine eigene, einfache Domain aufrufbar sein.
- Es muss eine eigenen Kommandosprache entwickelt werden, mit der über die Syntax “/command” Befehle direkt in der Eingabe-Zeile gestartet werden können.
- Es sollen so mehrere Funktionalitäten in dem Chat-Room gestartet werden können, die zu implementieren sind
  - Abstimmungen
  - Taschenrechner
- Die Raum-Schlüssel sollen im “ThreeWords”-Stil aus englischen Wörtern generiert werden, sodass eine intuitive Eingabe möglich wird.
- Eine Anmeldung mit Benutzername und Kennwort ist nötig.

### 1.2.3 Fachliches Umfeld

- Plattformunabhängigkeit
- Mobile Lösung
- Datenbanken
- IT-Security
- DSGVO

### 1.2.4 Ausblick

Bei erfolgreichen Entwicklungsergebnissen soll die Lösung in Betrieb genommen werden und wird öffentlich zur Verfügung gestellt.

### 1.2.5 Erweiterungsmöglichkeiten (optional)

- Minispiele (TickTackToe, Schiffe-Versenken) gegen andere Benutzer oder Bots

### 1.3 Arbeitspaketplan

Der Arbeitspaketplan bezeichnet die Aufzählung jedes Arbeitspakets auf Basis des Lastenhefts.

Ein Arbeitspaket wird durch folgendes Definiert

- Ein Definiertes Ergebnis (was soll in diesem Arbeitspaket erreicht werden)
- Der zeitliche Aufwand des Arbeitspakets
- Die Vorbedingungen die beim bearbeiten zu beachten sind
- Die Dauer

### 1.4 Personalplanung

Die Personalplanung soll Aufschluss darüber geben, welche Arbeitspaket von wem, wann bearbeitet werden soll. Diese ist im Backlog festgehalten.

### 1.5 Zeitplan

Die Zeitplanung lässt sich in die folgenden Meilensteine und zugehörige Deadlines unterteilen

Phase	Deadline
Planungsphase	KW 48/2021
Definitionsphase	KW 50/2021
Entwurfsphase	KW 02/2022
Implementierungsphase	KW 08/2022
Testphase	KW 11/2022

### 1.6 Auswahl kritischer Technologien

Der DEV-CHAT wird mit Hilfe von Webtechnologien umgesetzt. Diese bieten im Gegensatz zu klassischen Client/Server Anwendungen vor allem den Vorteil der Plattformunabhängigkeit. Dies reduziert auf der einen Seite den Programmieraufwand, da nicht für jedes System individuelle Anwendungen geschrieben werden müssen, auf der anderen Seite sorgt dies für eine hohe Kompatibilität, da zur Ausführung der Applikation lediglich ein Webbrowser-fähiges Gerät von Nöten ist, da die Hauptrechenarbeit von einem Server geleistet wird.

Hierbei stellt sich die Frage, welches Web-Framework genutzt werden soll. Außerdem muss festgelegt werden, welche Datenbank verwendet werden soll, um Daten in Echtzeit zu speichern und zu verwalten. Als dritten Punkt ist noch die Auswahl des Hostings eine Entscheidung, die getroffen werden musste.

Hierzu wurden zwei Prototypen der App erstellt. Diese Prototypen umreißen nur ganz grob

die Chat-Funktion des DEV-CHAT und dienen einzig und allein dem Zweck, verschiedene Frameworks zu testen. Der erste Prototyp wurde mit Python-Flask realisiert. Flask ist ein Python-Framework, dass es erlaubt Webseiten zu bauen, die serverseitig gerendert werden. Zwar lassen sich sehr schnell Webseiten erstellen, allerdings ist die Kommunikation zwischen Front- und Backend, sowie das erneute Rendern von Seitenkomponenten sehr umständlich. Diese Funktionen sind allerdings essenziell für die geplante Anwendung, um den Chat kontinuierlich zu aktualisieren, während der Rest der Seite unverändert bleibt. Der zweite Prototyp setzt auf ein JavaScript-basiertes Framework. Hier verwendeten wir Next.js in Kombination mit React.js und TypeScript. Next.js basiert auf React und bringt viele Vorteile mit sich, wie das Anbieten von serverless-functions, was das Zusammenspiel von Front- und Backend deutlich erleichtert. TypeScript wurde gewählt, da es ein Typensystem zu JavaScript hinzufügt, was dazu führt, dass Fehler schneller erkannt werden und somit schwerwiegende Laufzeitfehler reduziert werden. Das Zusammenspiel der drei Technologien war sehr intuitiv. Somit ist ein JavaScript-basiertes Framework ideal für die Anwendung.

Lokale Datenbanken wie MariaDB eignen sich in Kombination mit Webtechnologien nicht. Cloud-Datenbanken bieten den Vorteil der einfachen Zugreifbarkeit, sowie Standortunabhängigkeit der Server. Die Anwendung kann somit auf jedem Server problemlos gehostet werden, ohne eine Migration der Datenbank vornehmen zu müssen. Die Wahl fällt hierbei auf Supabase.io, welche eine Open-Source-Variante der bekannten Cloud-Datenbank Firebase (von Google) ist. Hier lassen sich kostenlos Datenbanken erstellen und einfach über JavaScript darauf zugreifen.

Die Applikation auf einem lokalen Server von uns zu hosten, erachten wir als nicht sinnvoll. Hierzu gibt es besser geeignete Möglichkeiten, wie etwa Vercel. Vercel ist die Firma, die das Next.js-Framework entwickelt hat. Sie bieten Hosting von Web-Apps (mit den unterschiedlichsten Frameworks) an. Das Einstiegsmodell ist hierbei völlig kostenlos und ausreichend für unseren Anwendungsfall. Vercel zieht sich den Source-Code von GitHub und hostet die App dann auf ihren Servern. Zusätzlich bietet Vercel die Möglichkeit, einfache Domains mit der Website zu verknüpfen und stellt somit die perfekte Lösung für unsere Anwendung dar.

### **1.6.1 Aktualisierung in der Implementierungsphase am 25.02.2022**

Die App auf Vercel zu hosten, ergab sich als unmöglich. Grund dafür ist die Realtime-Funktionalität. Diese ist essenziell für die Anforderungen der Chat-App. Realtime-Chat-Updates lassen sich mit offenen Websockets zwischen dem Server und dem Client erreichen. Vercel unterstützt bisher aber noch keine Websockets. Es existieren zwar Workarounds, aber keine native Lösung.

Die App wird aufgrund dessen auf einem Linux-Server gehostet. Dieser ist mit NGINX

und PM2 konfiguriert und stellt so die Web-APP auf der Domain `dev-chat.me` bereit.

## 2 Definition

### 2.1 Use Case Diagramm

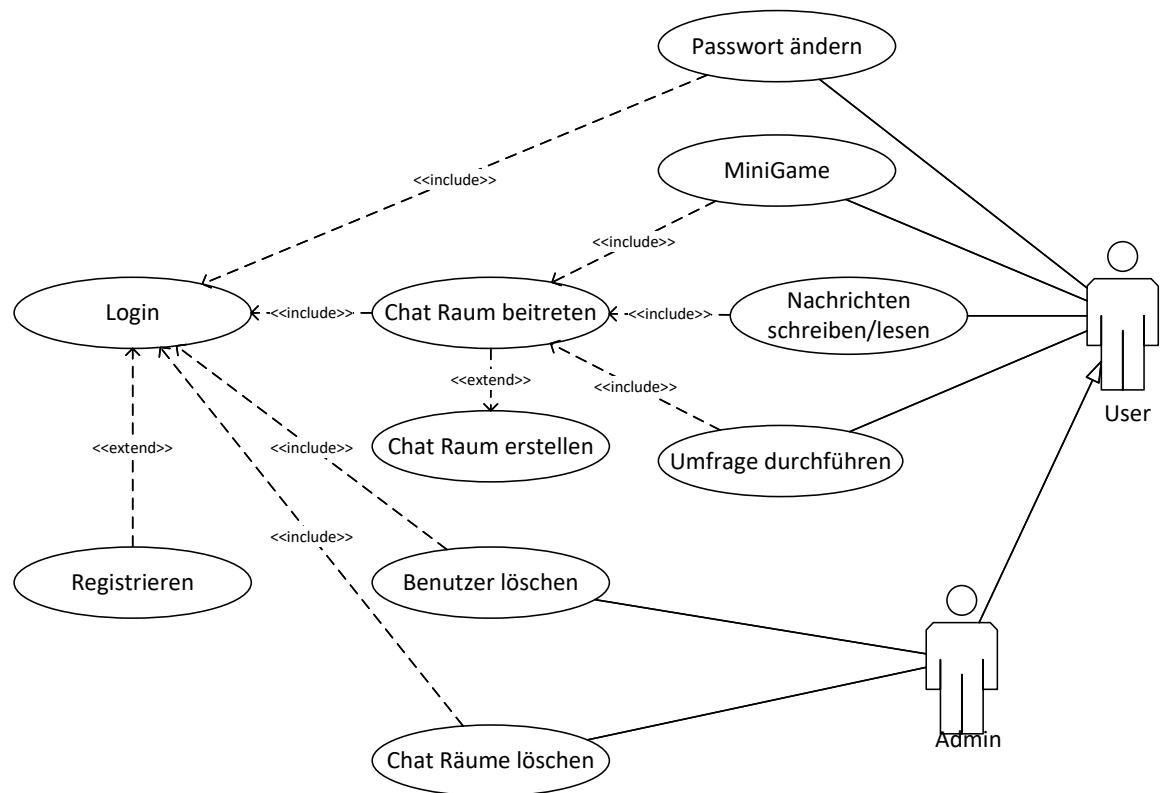


Abb. 2.1: Use Case Diagramm

## 2.2 Use Case Beschreibung

<b>Name:</b>	<b>Login</b>
Ziel:	Anmelden mit bestehenden Logindaten auf der Website
Kategorie:	Primär
Vorbedingung:	Der Benutzer muss bereits einen Account erstellt haben
Nachbedingung:	Der Benutzer ist eingeloggt
Fehlerfälle:	<p>Falsche Logindaten</p> <ul style="list-style-type: none"> <li>• Rückmeldung dass falsche Zugangsdaten verwendet wurden</li> <li>• Möglichkeit das Kennwort über die Email zurückzusetzen</li> </ul> <p>Abbruch durch den Benutzer</p> <ul style="list-style-type: none"> <li>• keine Zustandsänderung</li> </ul>
Akteure:	User
Auslösendes Event:	Benutzer öffnet die Website und klickt auf einloggen
Beschreibung/Erweiterungen:	Der Benutzer meldet sich an, um die verschiedenen Dienste des DevChats zu verwenden
Alternativen:	Benutzer kann sich registrieren, sofern die Email nicht schon registriert ist.

<b>Name:</b>	<b>Registrieren</b>
Ziel:	Anlegen eines neuen Accounts
Kategorie:	Primär
Vorbedingung:	Email ist noch nicht mit einem anderen Account registriert
Nachbedingung:	Der Benutzer ist eingeloggt
Fehlerfälle:	<p>Email bereits verwendet</p> <ul style="list-style-type: none"> <li>• Rückmeldung dass Email bereits verwendet wurde</li> </ul> <p>Abbruch durch den Benutzer</p> <ul style="list-style-type: none"> <li>• keine Zustandsänderung</li> <li>• Account wird nicht angelegt</li> </ul>
Akteure:	User
Auslösendes Event:	Benutzer öffnet die Website und klickt auf registrieren.
Beschreibung/Erweiterungen:	Ein Benutzer möchte einen neuen Account erstellen.
Alternativen:	Login mit bestehendem Account.

<b>Name:</b>	<b>Chat Raum beitreten</b>
Ziel:	Bestehenden Chat Raum beitreten
Kategorie:	Primär
Vorbedingung:	Benutzer muss eingeloggt sein
Nachbedingung:	Der Benutzer befindet sich im Chat Raum
Fehlerfälle:	<p>Schlüssel ist ungültig</p> <ul style="list-style-type: none"> <li>• Rückmeldung dass Schlüssel ungültig ist.</li> </ul>
Akteure:	User
Auslösendes Event:	Benutzer klickt nach dem Einloggen auf bestehenden Chat Raum beitreten
Beschreibung/ Erweiterungen:	Beitreten in einen bereits erstellten Chat Raum über einen Schlüssel
Alternativen:	Einen neuen Chat Raum erstellen und diesem beitreten
<b>Name:</b>	<b>Chat Raum erstellen</b>
Ziel:	Erstellen eines neuen Chat Raums
Kategorie:	Primär
Vorbedingung:	Benutzer muss eingeloggt sein
Nachbedingung:	Der Benutzer befindet sich in dem neuen Chat Raum
Fehlerfälle:	
Akteure:	User
Auslösendes Event:	Benutzer klickt nach dem Einloggen auf neuen Chat Raum erstellen
Beschreibung/ Erweiterungen:	Der Benutzer erstellt einen neuen Chat Raum und erhält dazu den Schlüssel und ist anschließend in dem Chat Raum
Alternativen:	Bestehendem Chat Raum beitreten
<b>Name:</b>	<b>MiniGame</b>
Ziel:	Starten und Spielen eines Minigames
Kategorie:	Sekundär
Vorbedingung:	Benutzer muss einem Chat Raum beigetreten sein
Nachbedingung:	
Fehlerfälle:	<p>Gegenspieler nimmt das Spiel nicht an</p> <ul style="list-style-type: none"> <li>• Rückmeldung dass Gegner spiel nicht angenommen hat.</li> </ul> <p>Kein weiterer Spieler im Chat Raum</p> <ul style="list-style-type: none"> <li>• Rückmeldung dass kein weiterer Spieler im Raum ist.</li> </ul>
Akteure:	User
Auslösendes Event:	Starten des Spiels über einen Kurzbefehl im Chat Raum
Beschreibung/ Erweiterungen:	Der Benutzer kann über einen Kurzbefehl ein Minigame gegen einen Benutzer im Raum starten.
Alternativen:	

<b>Name:</b>	<b>Nachrichten schreiben/lesen</b>
Ziel:	Lesen und Schreiben von Chatnachrichten und Kurzbefehlen
Kategorie:	Primär
Vorbedingung:	Benutzer muss einem Chat Raum beigetreten sein
Nachbedingung:	
Fehlerfälle:	<p>Befehl falsch verwendet</p> <ul style="list-style-type: none"> <li>• Rückmeldung, dass Befehl ungültig ist.</li> </ul>
Akteure:	User
Auslösendes Event:	Nachricht wird abgeschickt/empfangen, Kurzbefehl wird abgeschickt
Beschreibung/ Erweiterungen:	Schreiben und Lesen von Nachrichten in einem Chat Raum. Mittels Kurzbefehlen können weitere Funktionalitäten des DevChats genutzt werden.
Alternativen:	

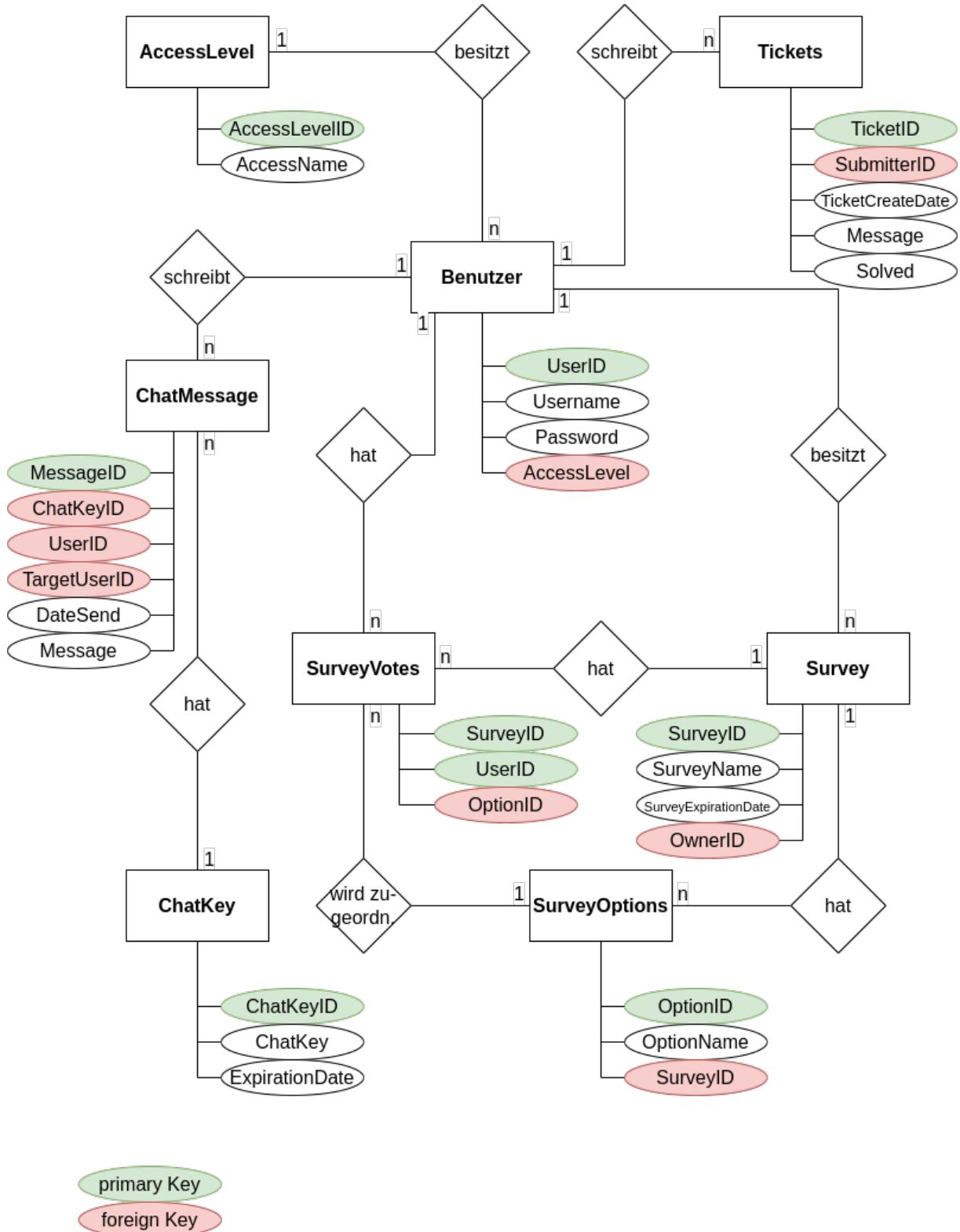
<b>Name:</b>	<b>Umfrage durchführen</b>
Ziel:	Umfrage starten, einstellen und durchführen
Kategorie:	Sekundär
Vorbedingung:	Benutzer muss einem Chat Raum beigetreten sein
Nachbedingung:	Umfrage ist abgeschlossen und Ergebnisse werden dargestellt
Fehlerfälle:	<p>Befehl falsch verwendet</p> <ul style="list-style-type: none"> <li>• Rückmeldung, dass Befehl ungültig ist.</li> </ul>
Akteure:	User
Auslösendes Event:	Kurzbefehl, der die Umfrage startet, wird abgeschickt
Beschreibung/ Erweiterungen:	Durch einen Kurzbefehl wird eine Umfrage gestartet, diese wird anschließend konfiguriert und in Chat Raum veröffentlicht. Alle Chatraumteilnehmer können dann abstimmen. Der Ersteller der Umfrage kann die Umfrage beenden. Die Umfrageergebnisse werden dann im Chat veröffentlicht.
Alternativen:	

<b>Name:</b>	<b>Benutzer löschen</b>
Ziel:	Löschen von Benutzerkonten
Kategorie:	Sekundär
Vorbedingung:	Benutzer muss eingeloggt sein und Administrator Rechte haben.
Nachbedingung:	Die Benutzerkonten wurden gelöscht
Fehlerfälle:	
Akteure:	Admin
Auslösendes Event:	Im Admin Menu, die Benutzerkonten auswählen und auf löschen klicken
Beschreibung/ Erweiterungen:	Beim Login als Administrator, wird ein Zugang zum Admin Menu angezeigt, in diesem Menu können Benutzerkonten gelöscht werden.
Alternativen:	

Name:	Chat Räume löschen
Ziel:	Löschen von Chaträumen
Kategorie:	Sekundär
Vorbedingung:	Benutzer muss eingeloggt sein und Administrator Rechte haben.
Nachbedingung:	Chat Räume wurden gelöscht
Fehlerfälle:	
Akteure:	Admin
Auslösendes Event:	Im Admin Menu, die Chat Räume auswählen und auf löschen klicken
Beschreibung/ Erweiterungen:	Beim Login als Administrator, wird ein Zugang zum Admin Menu angezeigt, in diesem Menu können Chat Räume gelöscht werden.
Alternativen:	

## 2.3 Datenbankmodell

### 2.3.1 Entity Relationship Model (ERM)



### 2.3.2 Data Dictionary

#### Benutzer

Attribut	Datentyp	Länge	NULL	Default	Schlüssel	Beschreibung
UserID	INTEGER	-	Nein	-	P	UserID = 0 ist reserviert für Broadcast (siehe 134) UserID = 1 ist reserviert für System (Nachrichten die vom System gesendet werden, etwa bei Commands) UserID = 2 ist reserviert für Admin (Master-Account des DEV-CHAT)
Username	VARCHAR	40	Nein	-	-	-
Password	VARCHAR	30	Nein	-	-	Gehasht
AccessLevel	INTEGER	-	Nein	0	F	-

#### AccessLevel

Attribut	Datentyp	Länge	NULL	Default	Schlüssel	Beschreibung
AccessLevelID	INTEGER	-	Nein	-	P	ID = 0 ist reserviert für Standardrecht ID = 1 ist reserviert für Admin
AccessName	VARCHAR	30	Nein	-	-	-

#### Ticket

Attribut	Datentyp	Länge	NULL	Default	Schlüssel	Beschreibung
TicketID	INTEGER	-	Nein	-	P	Verfasser des Tickets
SubmitterID	INTEGER	-	Nein	-	F	Verfasser des Tickets
TicketCreateDate	DATE	-	Nein	-	-	Zeitpunkt der Ticketerstellung
Message	VARCHAR	255	Nein	-	-	Schilderung des Problems
Solved	BOOL	-	Nein	false	-	Bearbeitungszustand

#### ChatMessage

Attribut	Datentyp	Länge	NULL	Default	Schlüssel	Beschreibung
MessageID	INTEGER	-	Nein	-	P	-
ChatKeyID	INTEGER	-	Nein	-	F	Zugehöriger Chat
UserID	INTEGER	-	Nein	-	F	Verfasser
TargetUserID	INTEGER	-	Nein	-	F	Empfänger
DataSend	DATE	-	Nein	-	-	Sendezzeitpunkt
Message	VARCHAR	255	Nein	-	-	Chatnachricht

#### ChatKey

Attribut	Datentyp	Länge	NULL	Default	Schlüssel	Beschreibung
ChatKeyID	INTEGER	-	Nein	-	P	-
ChatKey		40	Nein	-	-	ThreeWord
ExpirationDate	DATE	-	Nein	-	-	Löschezeitpunkt des Chats

#### Survey

Attribut	Datentyp	Länge	NULL	Default	Schlüssel	Beschreibung
SurveyID	INTEGER	-	Nein	-	P	-
Name	VARCHAR	30	Nein	-	-	Umfrageüberschrift
Description	VARCHAR	255	Nein	-	-	Beschreibung der Umfrage
ExpirationDate	DATE	-	Nein	-	-	Ablaufzeit der Umfrage (Umfrage-Ende)
OwnerId	INTEGER	-	Nein	-	-	Ersteller der Umfrage

#### SurveyOption

Attribut	Datentyp	Länge	NULL	Default	Schlüssel	Beschreibung
OptionID	INTEGER	-	Nein	-	P	-
OptionName	VARCHAR	40	Nein	-	-	-
SurveyID	INTEGER	-	Nein	-	F	Zugehörige Umfrage

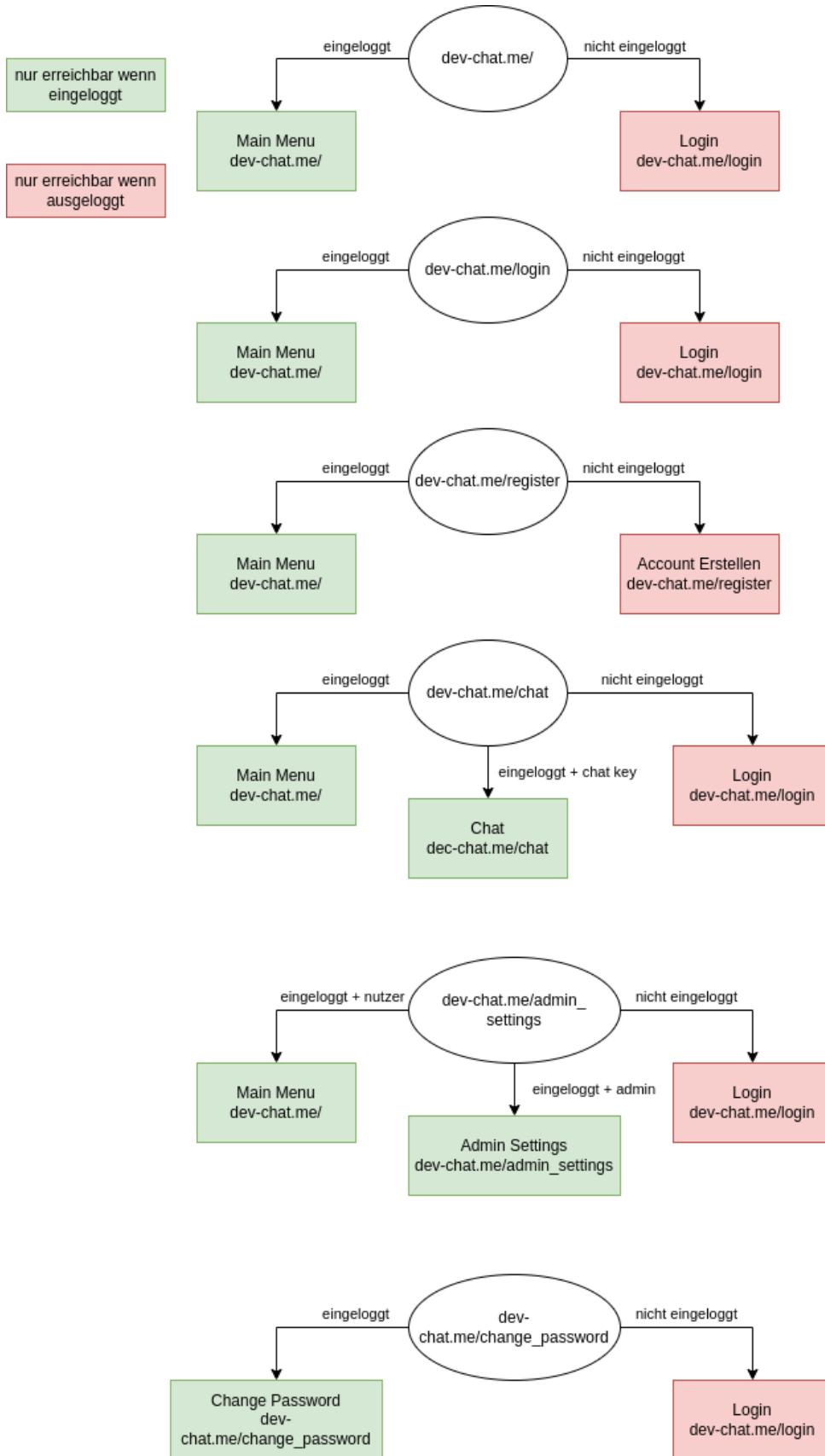
#### SurveyVote

Attribut	Datentyp	Länge	NULL	Default	Schlüssel	Beschreibung
SurveyID	INTEGER	-	Nein	-	P	Abgestimmte Umfrage

Attribut	Datentyp	Länge	NULL	Default	Schlüssel	Beschreibung
UserID	INTEGER	-	Nein	-	P	Abgestimmter Benutzer
OptionID	INTEGER	-	Nein	-	F	Abgestimmte Option

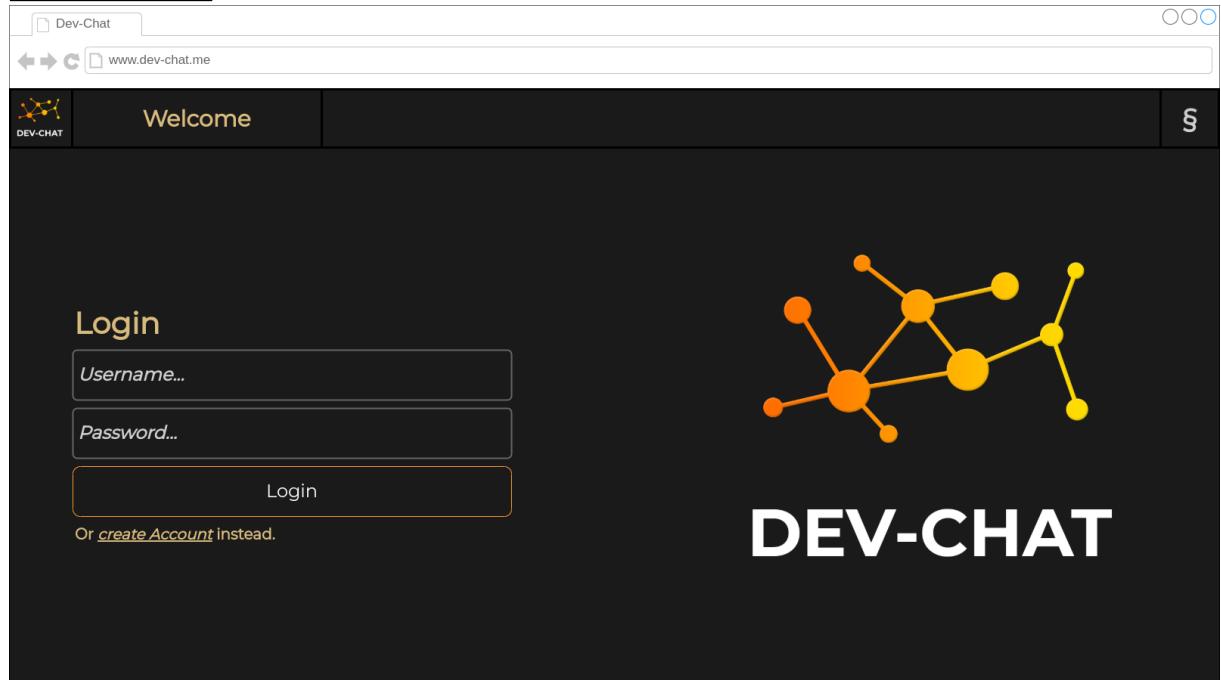
## 2.4 HMI

### 2.4.1 Seitenhirarchie

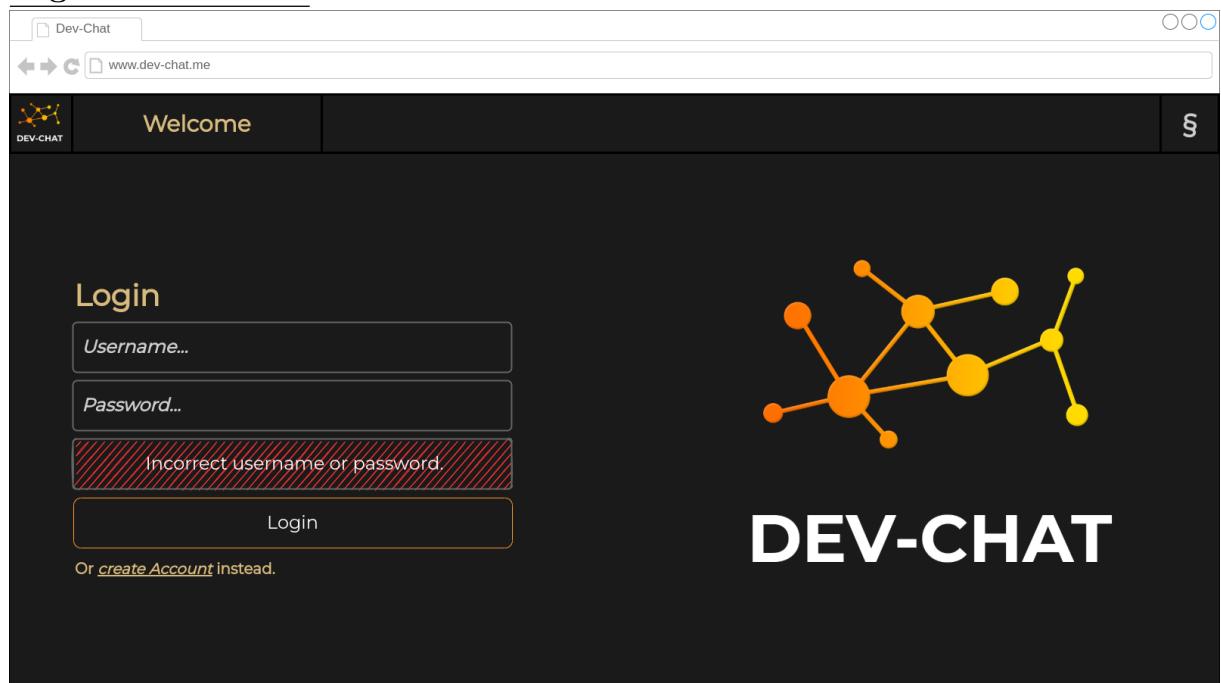


### 2.4.2 Prototyp Desktop

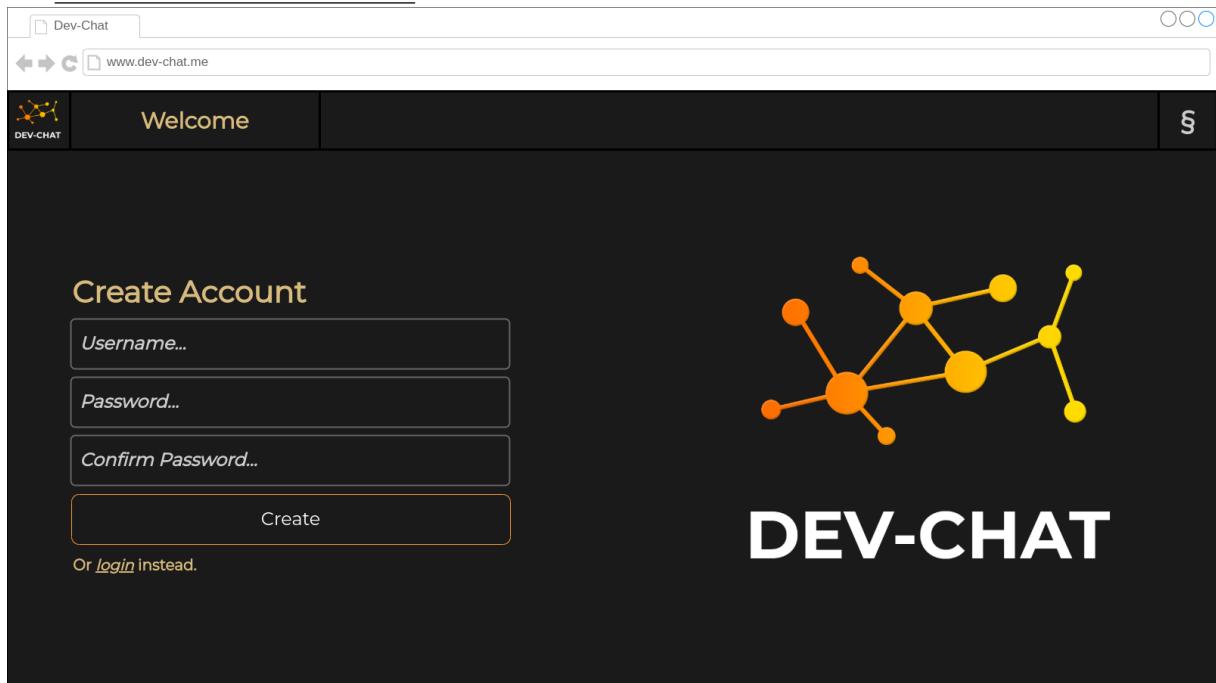
#### Login-Screen



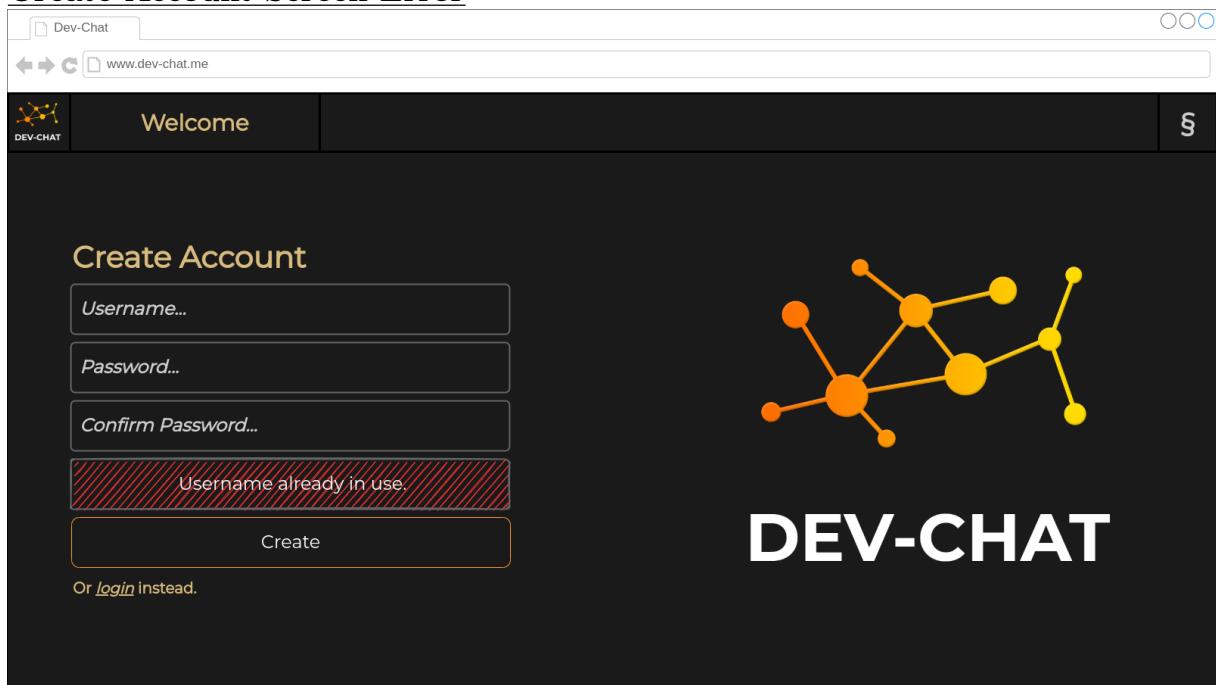
#### Login-Screen-Error

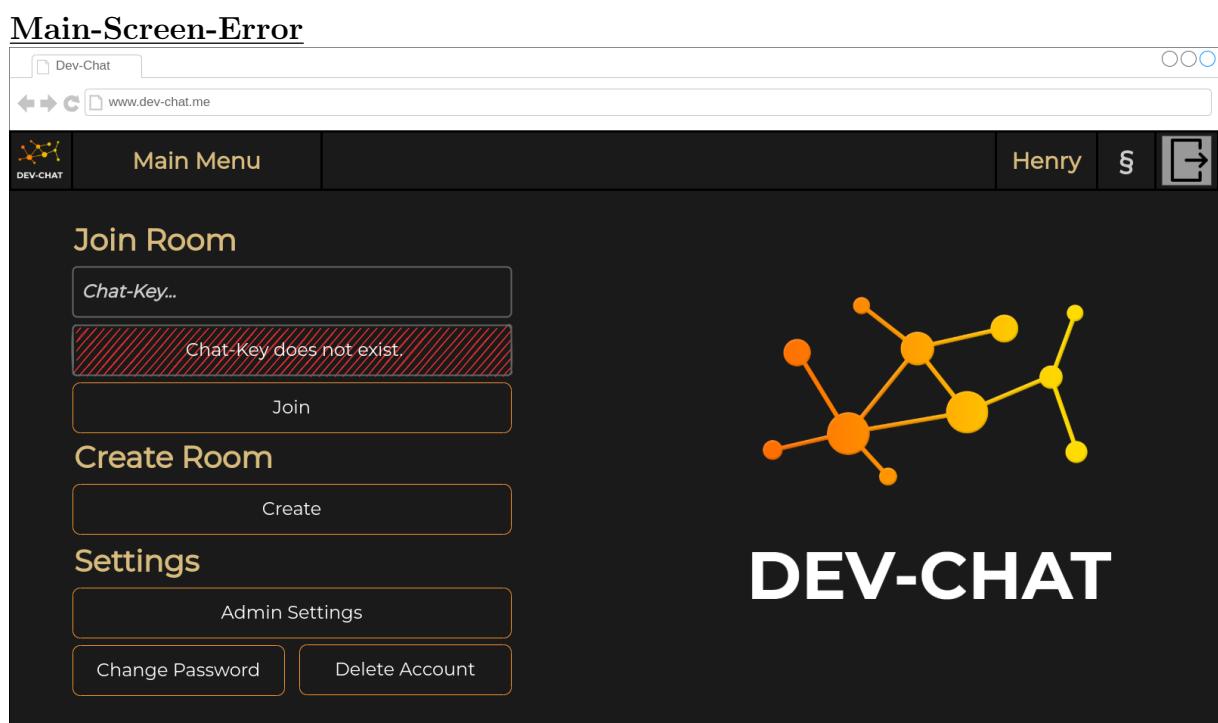
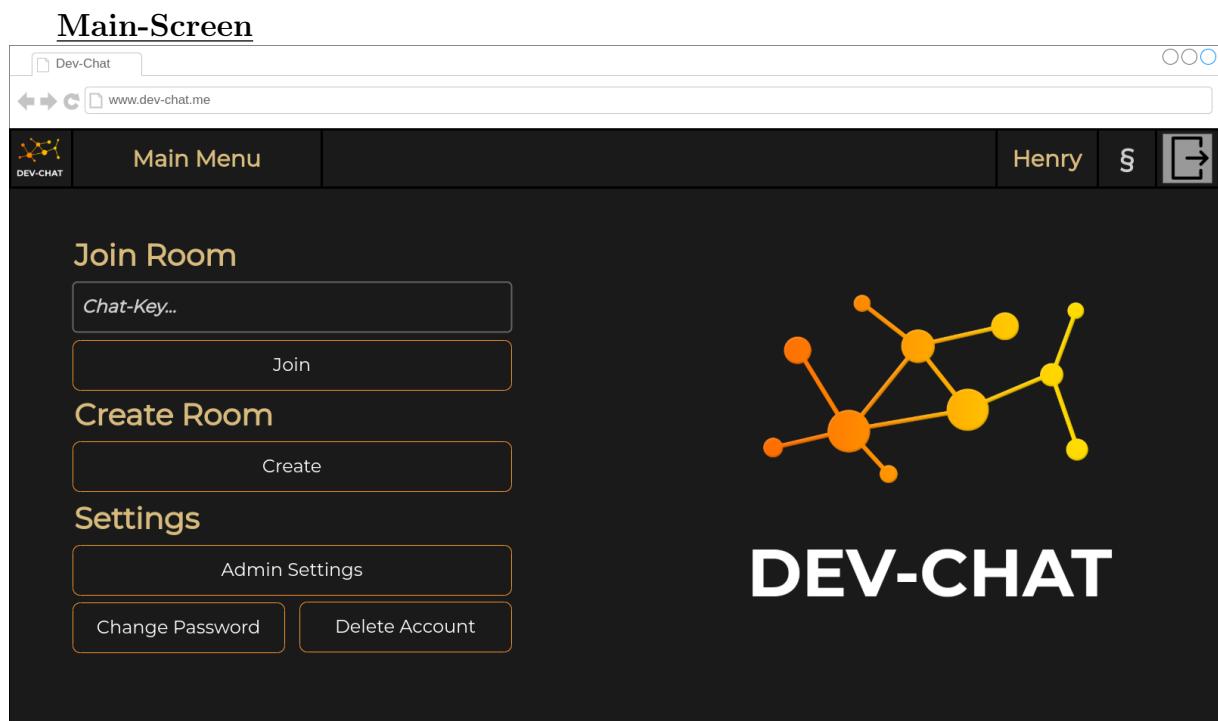


### Create-Account-Screen

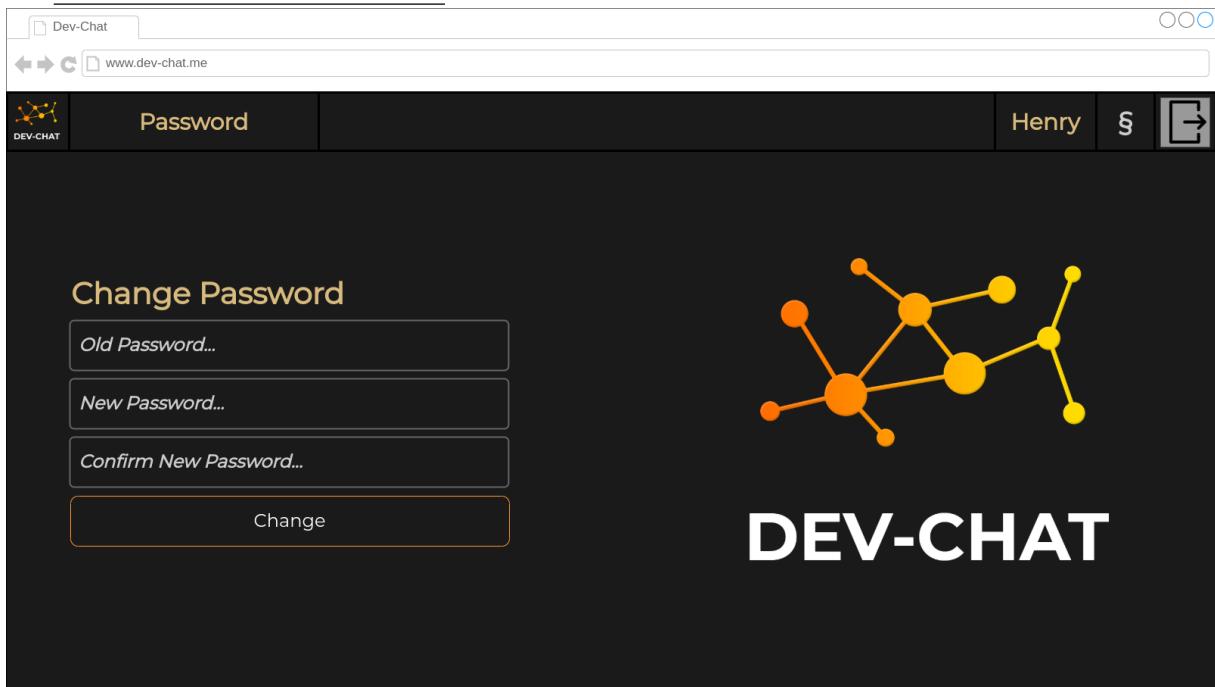


### Create-Account-Screen-Error

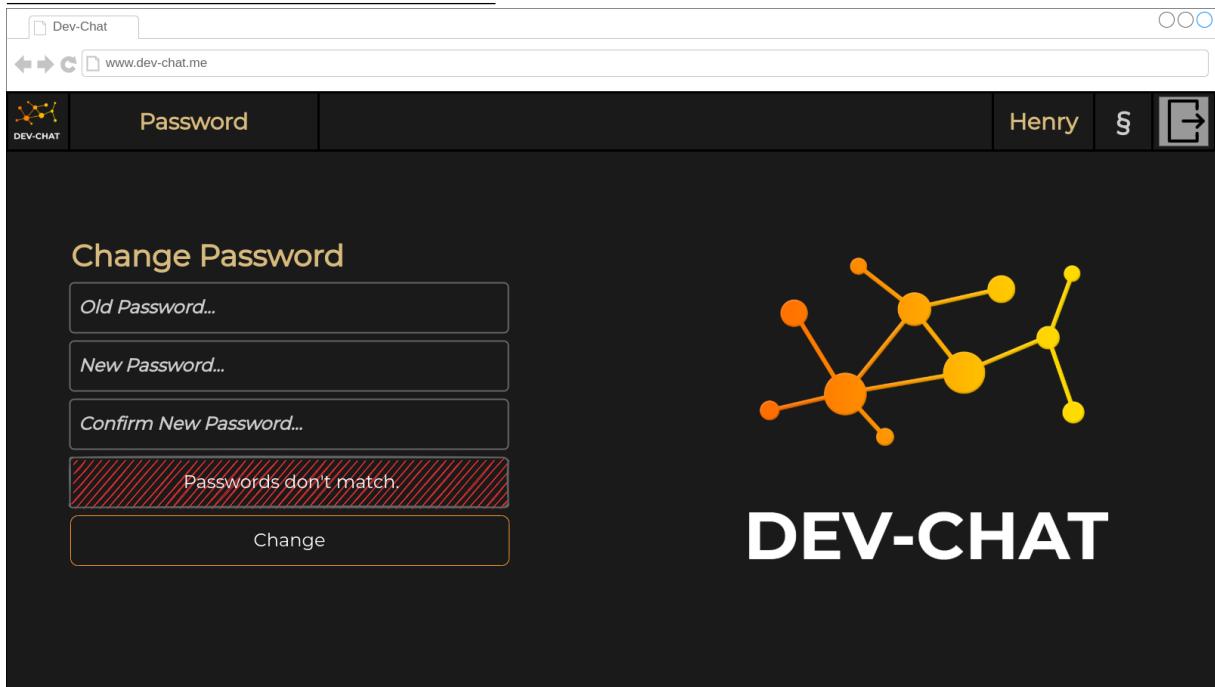




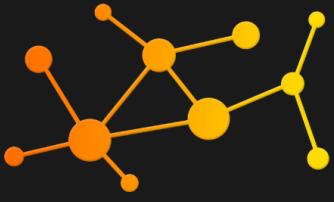
### Change-Password-Screen



### Change-Password-Screen-Error



## Admin-Settings-Screen



**User Settings**

Username	Access Level	Change Level	Password	Account
admin	ADMIN	Promote   Demote	Reset	Delete
henry	USER	Promote   Demote	Reset	Delete

**Room Settings**

Key	Expiration Time	Change Time	Room
TIT20	18/01/2021 12:00:00	Set Time	Delete
BreadFatherMother	20/01/2021 13:46:14	Set Time	Delete
Enter Custom Key...	Auto	Set Time	Add

**Survey Settings**

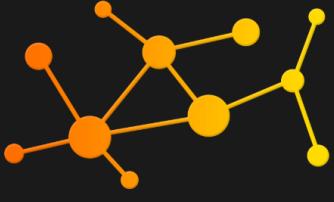
Name	Expiration Time	Owner	Change Time	Survey
Survey 1	16/01/2021 21:23:12	admin	Set Time	Delete
Survey 2	25/01/2021 16:59:59	henry	Set Time	Delete

**Tickets**

Submitter	Created on	State	Change State	Info
admin	16/01/2021 21:23:12	Done	Set To Do	View
henry	25/01/2021 16:59:59	To Do	Set Done	View

**DEV-CHAT**

## Admin-Settings-Screen-Error



**User Settings**

Username	Access Level	Change Level	Password	Account
admin	ADMIN	Promote   Demote	Reset	Delete
henry	USER	Promote   Demote	Reset	Delete

**Room Settings**

Key	Expiration Time	Change Time	Room
TIT20	18/01/2021 12:00:00	Set Time	Delete
BreadFatherMother	20/01/2021 13:46:14	Set Time	Delete
Enter Custom Key...	Auto	Set Time	Add

**Survey Settings**

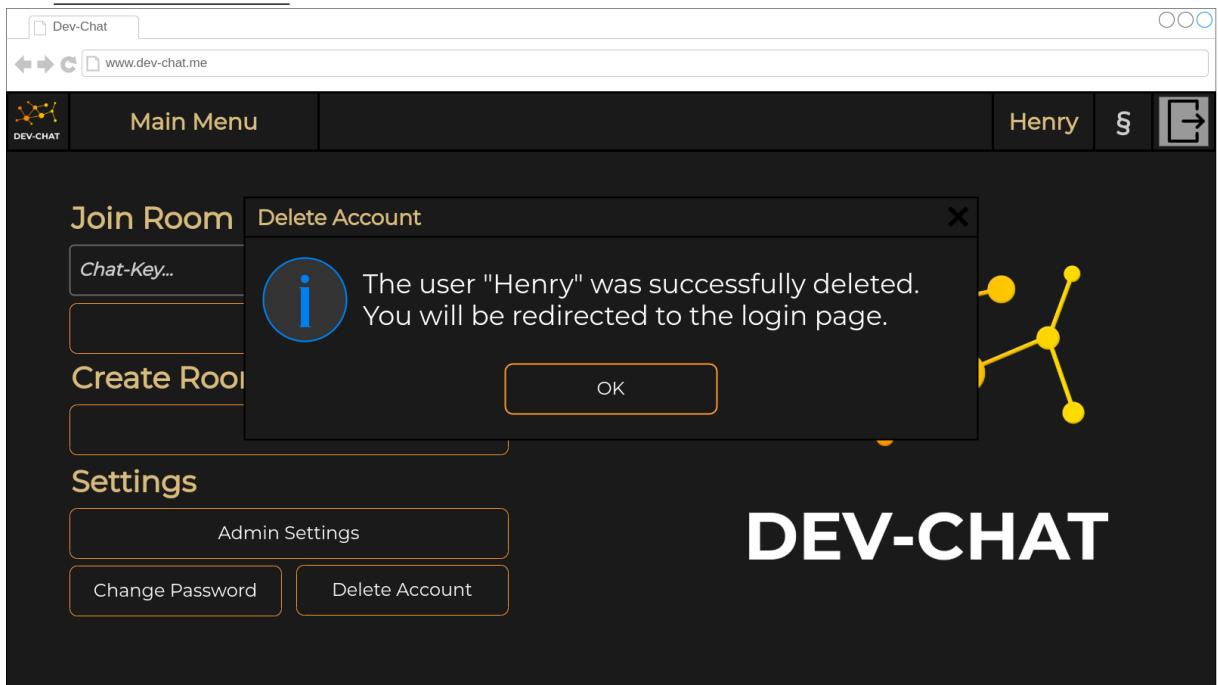
Name	Expiration Time	Owner	Change Time	Survey
Survey 1	16/01/2021 21:23:12	admin	Set Time	Delete
Survey 2	25/01/2021 16:59:59	henry	Set Time	Delete

**Tickets**

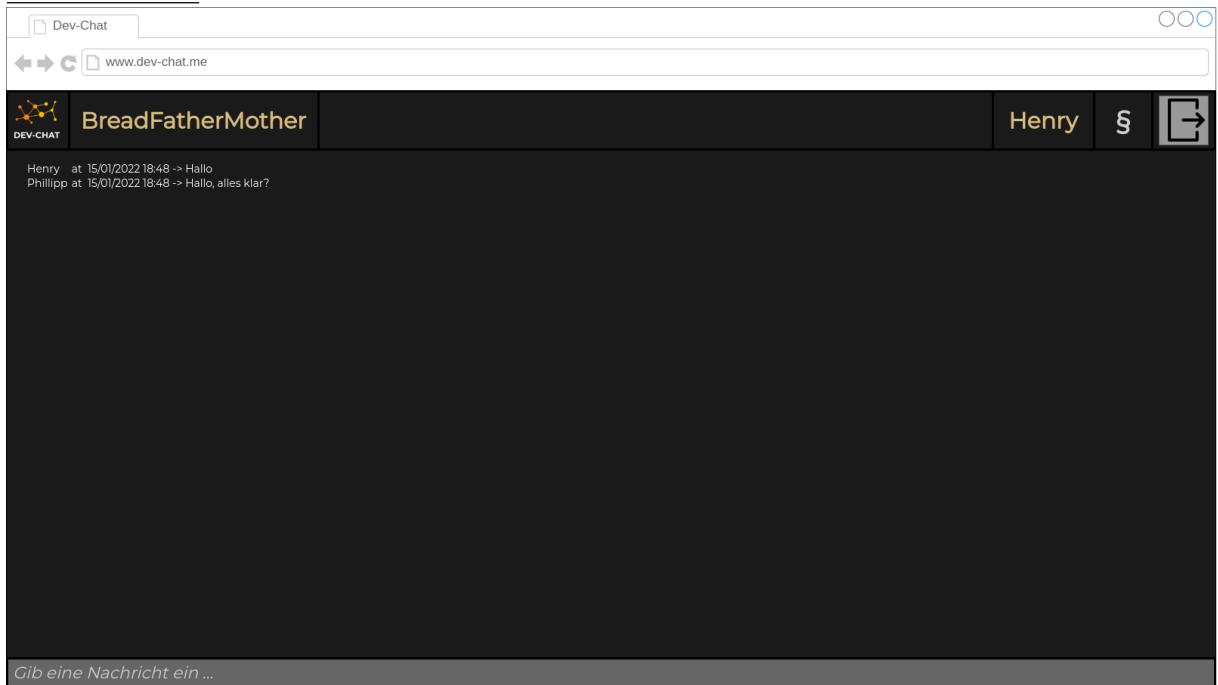
Submitter	Created on	State	Change State	Info
admin	16/01/2021 21:23:12	Done	Set To Do	View
henry	25/01/2021 16:59:59	To Do	Set Done	View

**DEV-CHAT**

### Pop-Up-Screen



### Chat-Screen



**Impressum-Screen**

The screenshot shows a web browser window for 'www.dev-chat.me'. The title bar says 'Dev-Chat'. The address bar shows the URL. The top navigation bar includes a logo, 'Impressum', 'Henry', a section for '§', and a search icon. Below this, there's a dark background with white text sections:

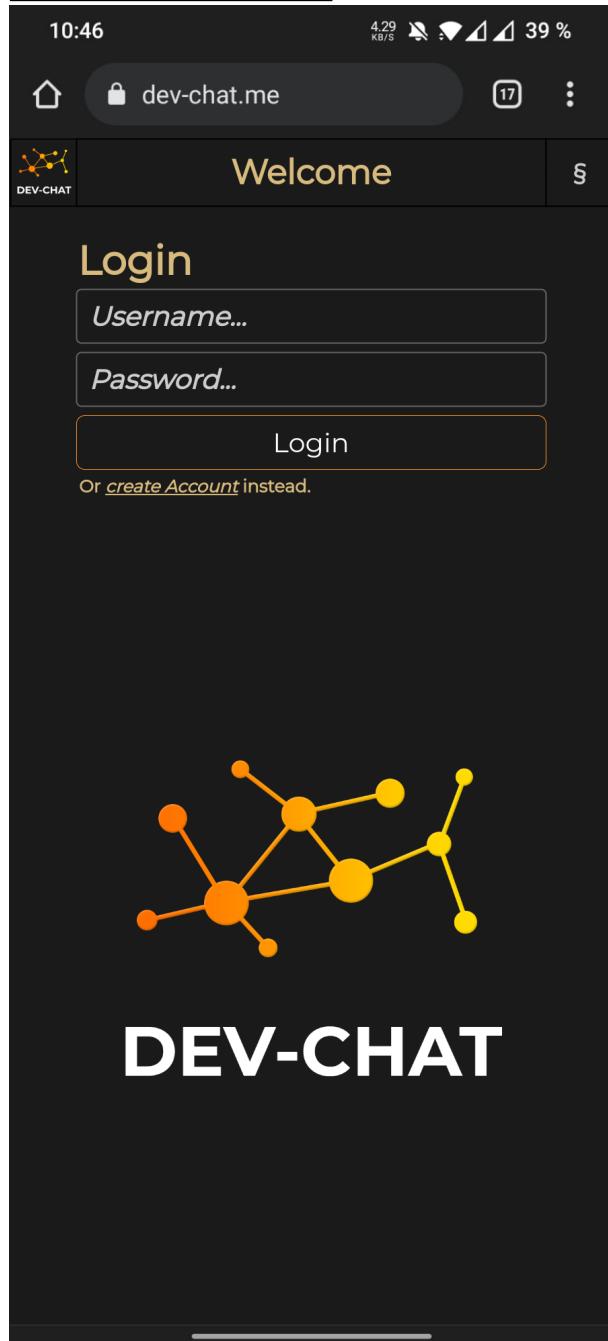
- Verantwortlich**: Henry Schuler
- Team**: Nico Bayer, Johannes Brandenburger, Lukas Braun, Jan Brutscher, Philipp Patzelt
- Kontakt**: DEV-CHAT, STRASSE, PLZ ORT; TELEFON, E-MAIL

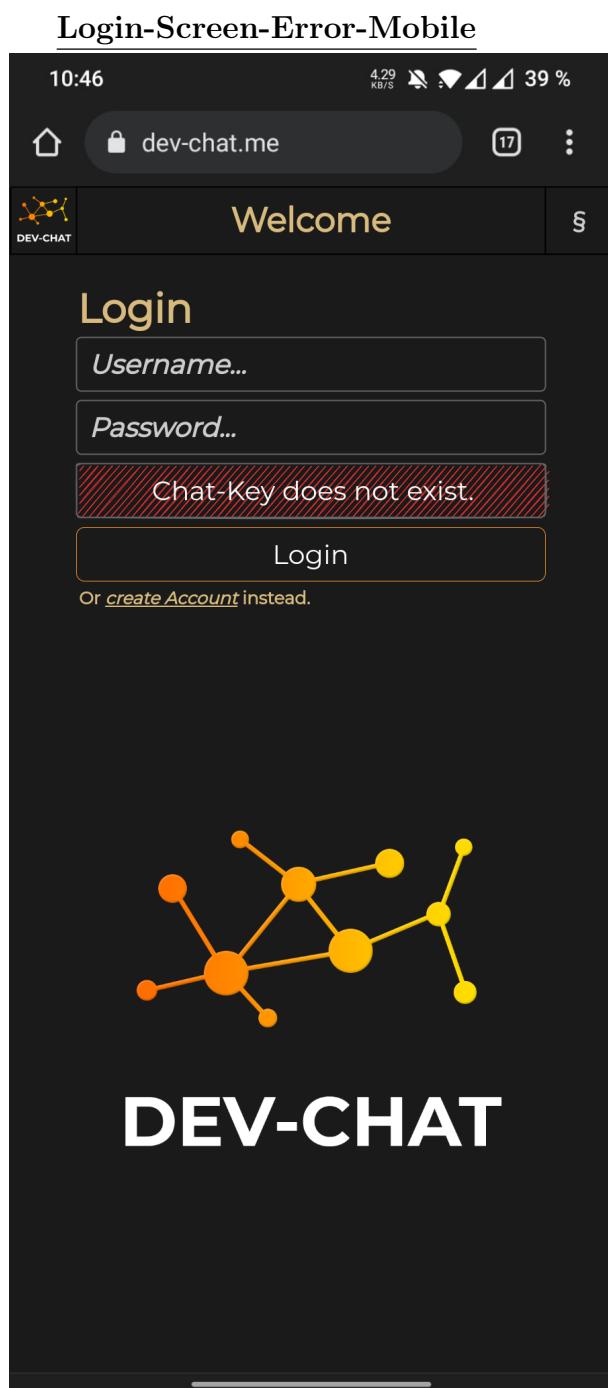
To the right of the text is a network graph consisting of orange and yellow nodes connected by lines.

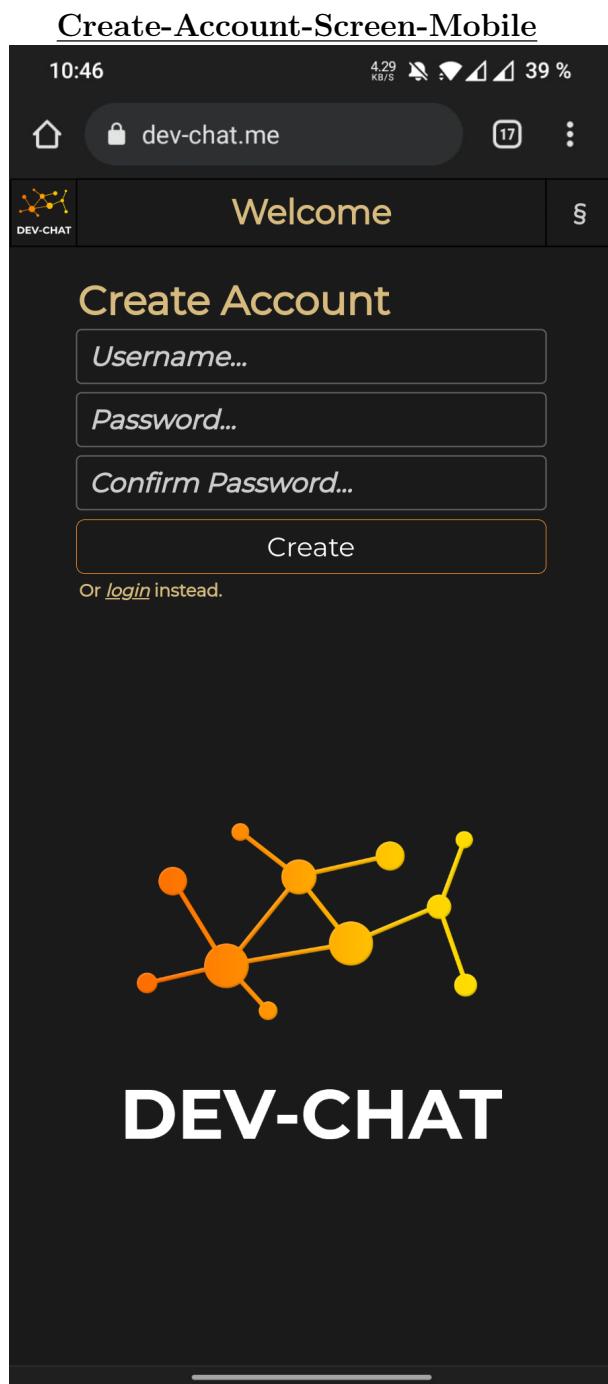
**DEV-CHAT**

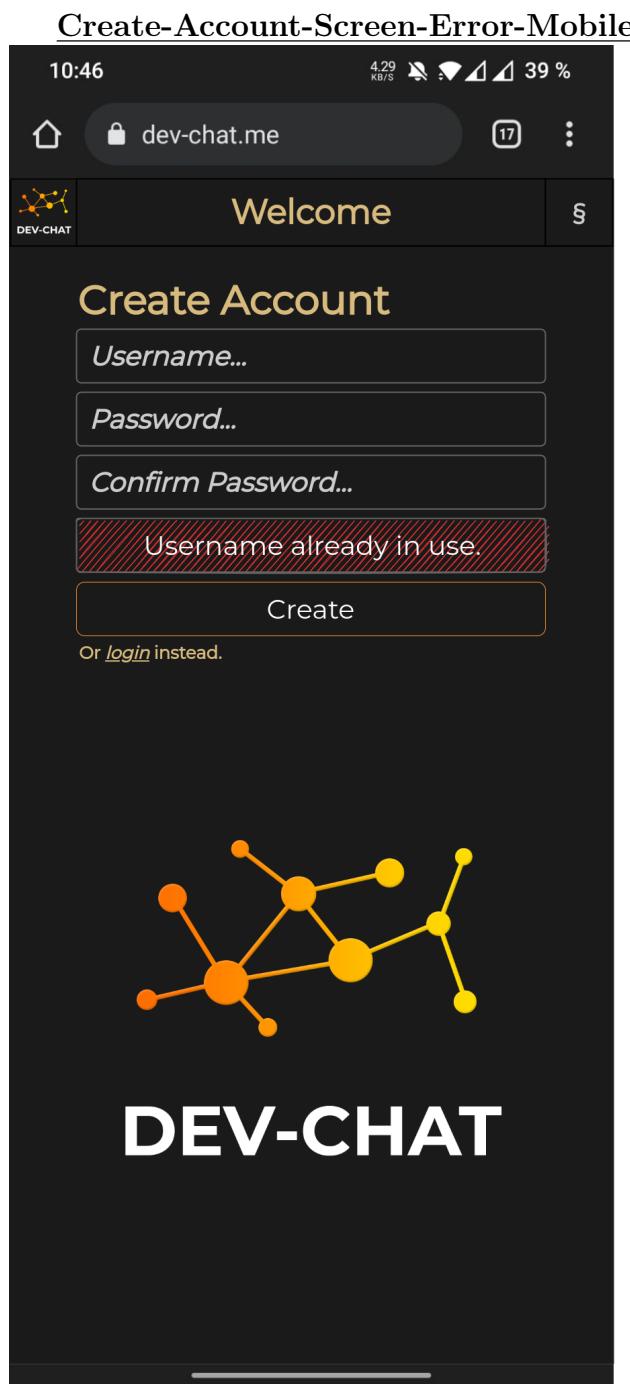
### 2.4.3 Prototyp Mobile

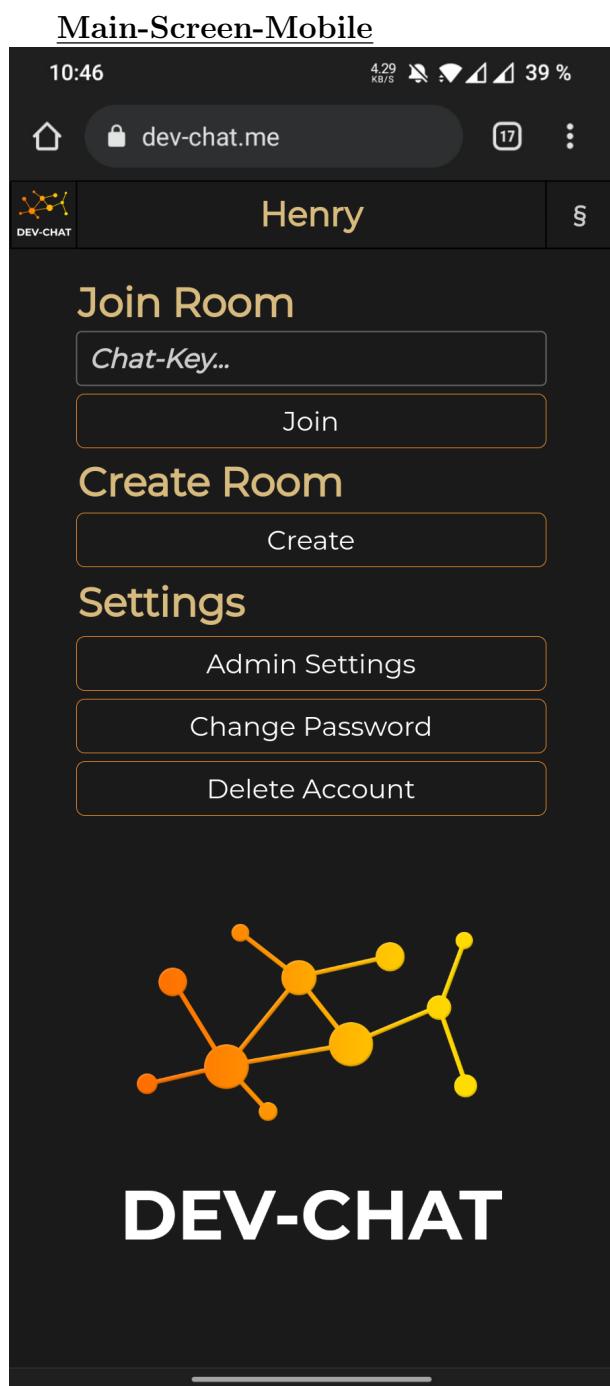
#### Login-Screen-Mobile

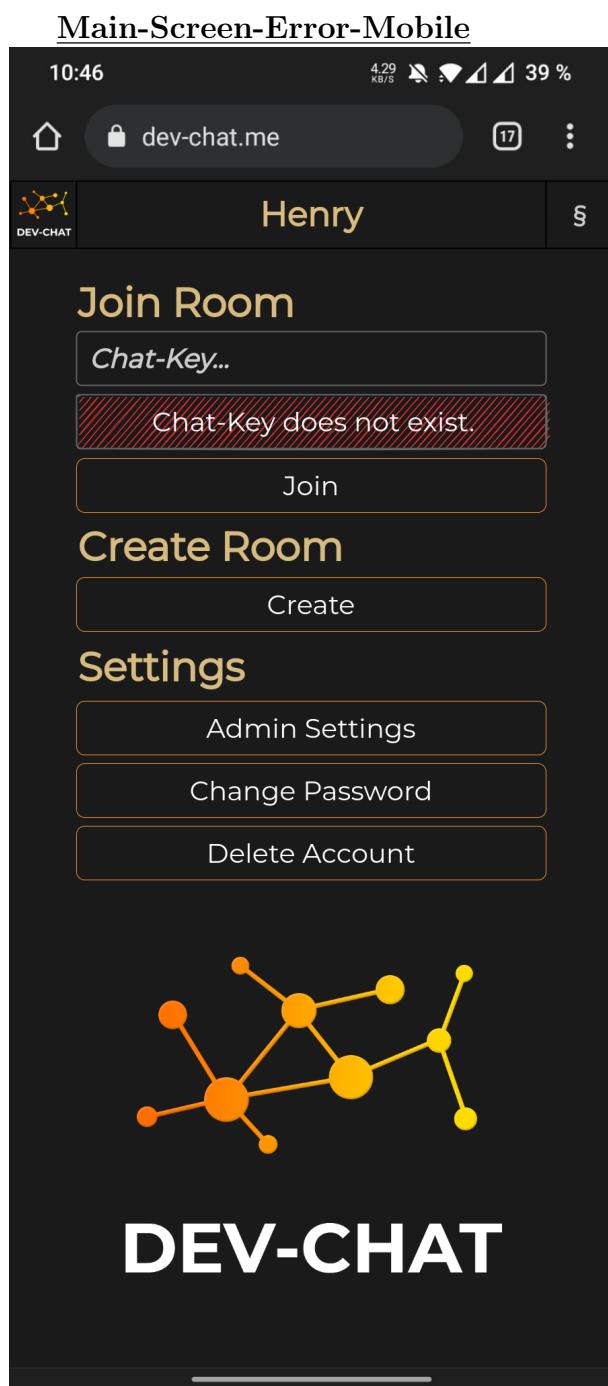


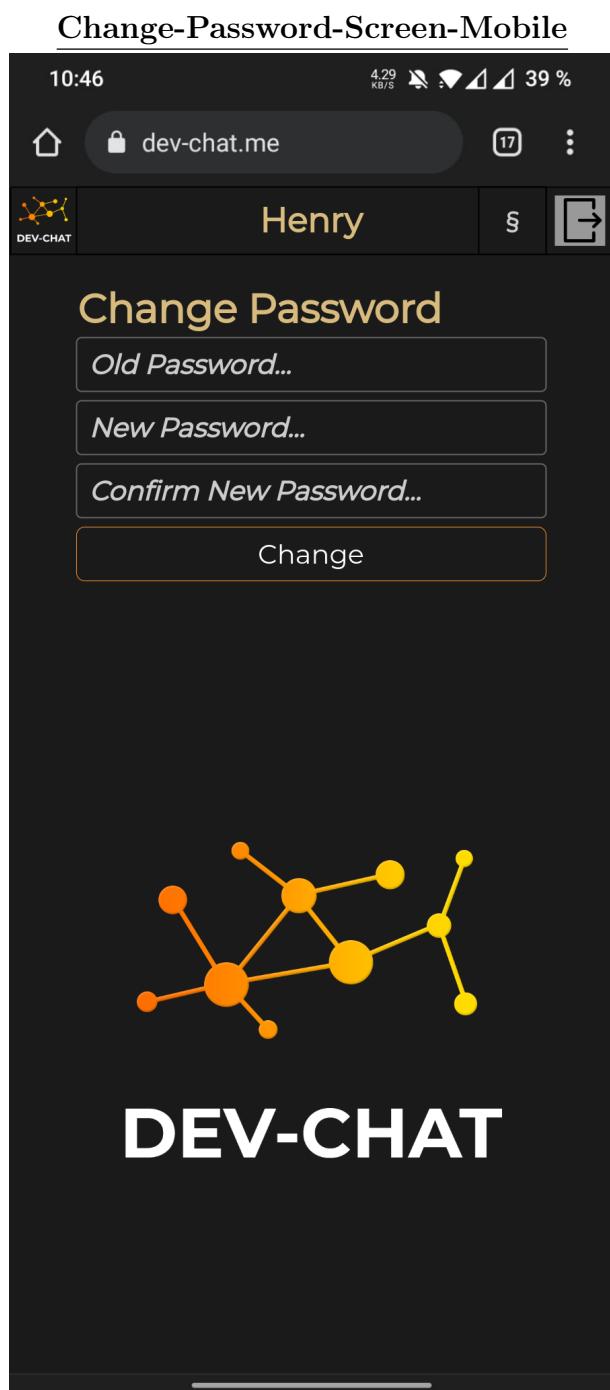


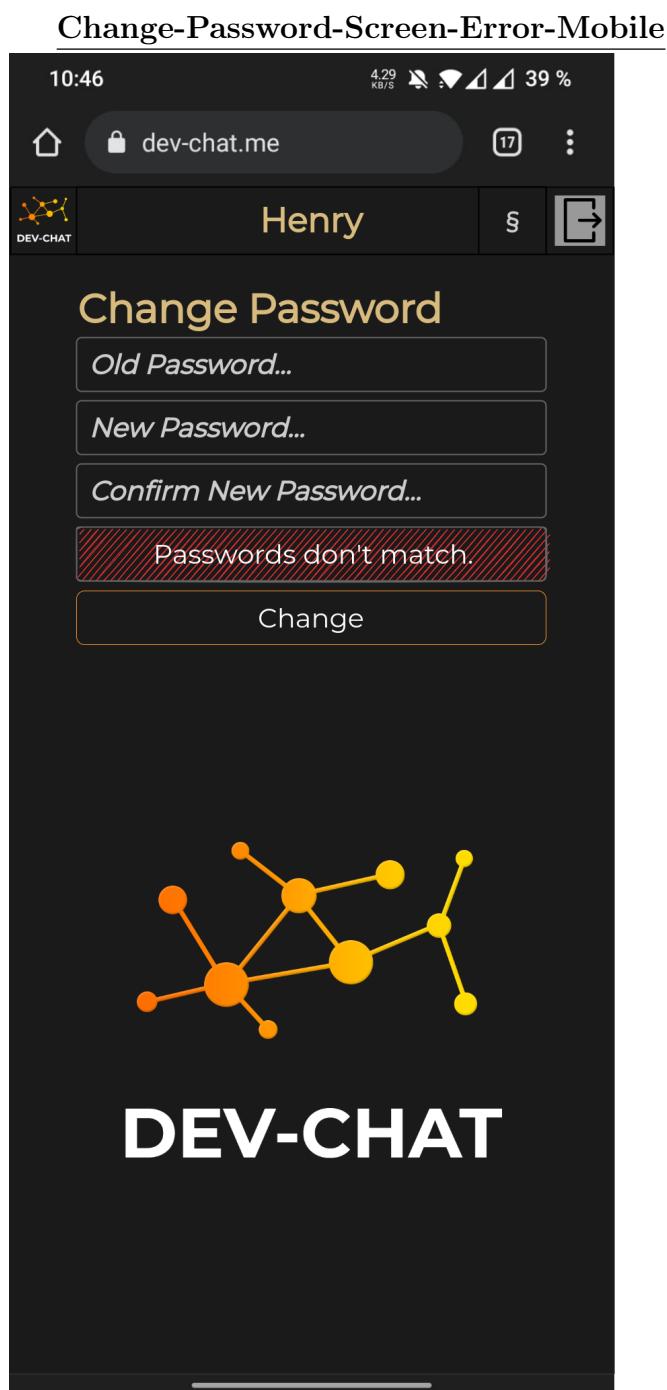












**Admin-Settings-Screen-Mobile**

10:46 4.29 KB/S 39 %

dev-chat.me 17 :

**Admin Settings**

**User Settings**

Username	Access Level	Change Level	Password	Account	
admin	ADMIN	Promote	Demote	Reset	Delete
henry	USER	Promote	Demote	Reset	Delete

**Room Settings**

Key	Expiration Time	Change Time	Room
TIT20	18/01/2021 12:00:00	Set Time	Delete
BreadFatherMother	20/01/2021 13:46:14	Set Time	Delete
Enter Custom Key...	Auto	Set Time	Add

**Survey Settings**

Name	Expiration Time	Owner	Change Time	Survey
Survey 1	16/01/2021 21:23:12	admin	Set Time	Delete
Survey 2	25/01/2021 16:59:59	henry	Set Time	Delete

**Tickets**

Submitter	Created on	State	Change State	Info
admin	16/01/2021 21:23:12	Done	Set To Do	View
henry	25/01/2021 16:59:59	To Do	Set Done	View

DEV-CHAT

**Admin-Settings-Screen-Error-Mobile**

10:46 4.29 KB/S 39 %

dev-chat.me 17 :

**Admin Settings**

**User Settings**

Username	Access Level	Change Level	Password	Account	
admin	ADMIN	Promote	Demote	Reset	Delete
henry	USER	Promote	Demote	Reset	Delete

**Room Settings**

Key	Expiration Time	Change Time	Room
TIT20	18/01/2021 12:00:00	Set Time	Delete
BreadFatherMother	20/01/2021 13:46:14	Set Time	Delete
<i>Enter Custom Key...</i>	Auto	Set Time	Add

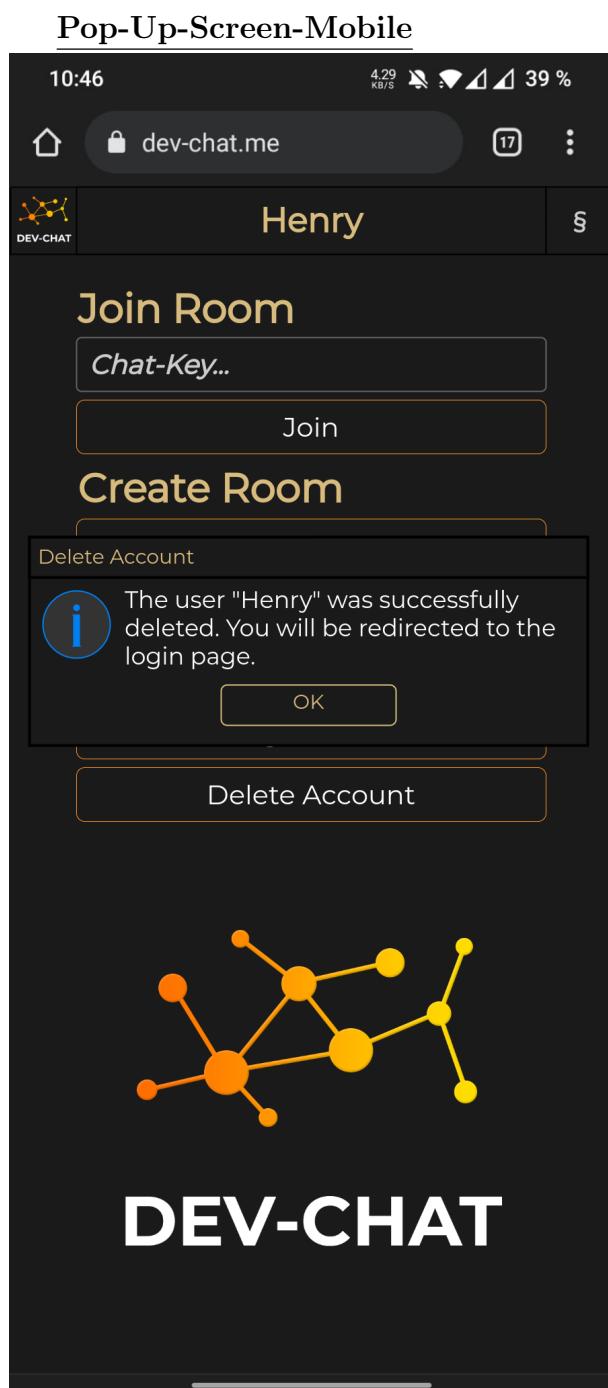
**Survey Settings**

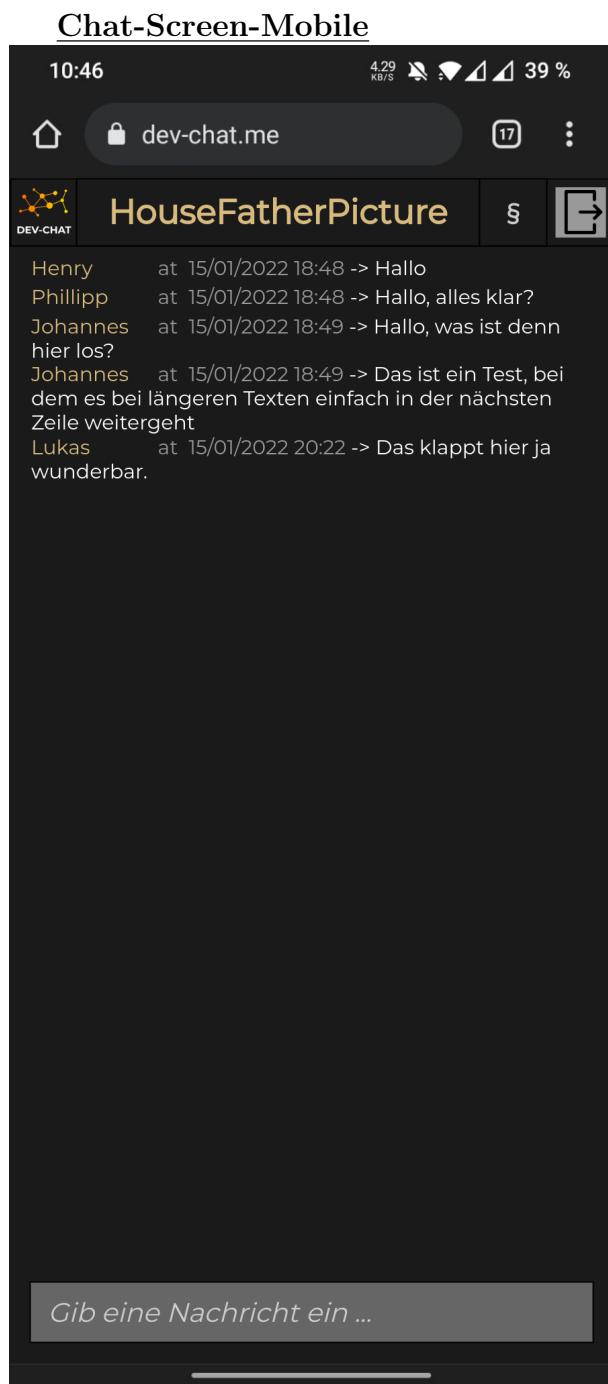
Name	Expiration Time	Owner	Change Time	Survey
Survey 1	16/01/2021 21:23:12	admin	Set Time	Delete
Survey 2	25/01/2021 16:59:59	henry	Set Time	Delete

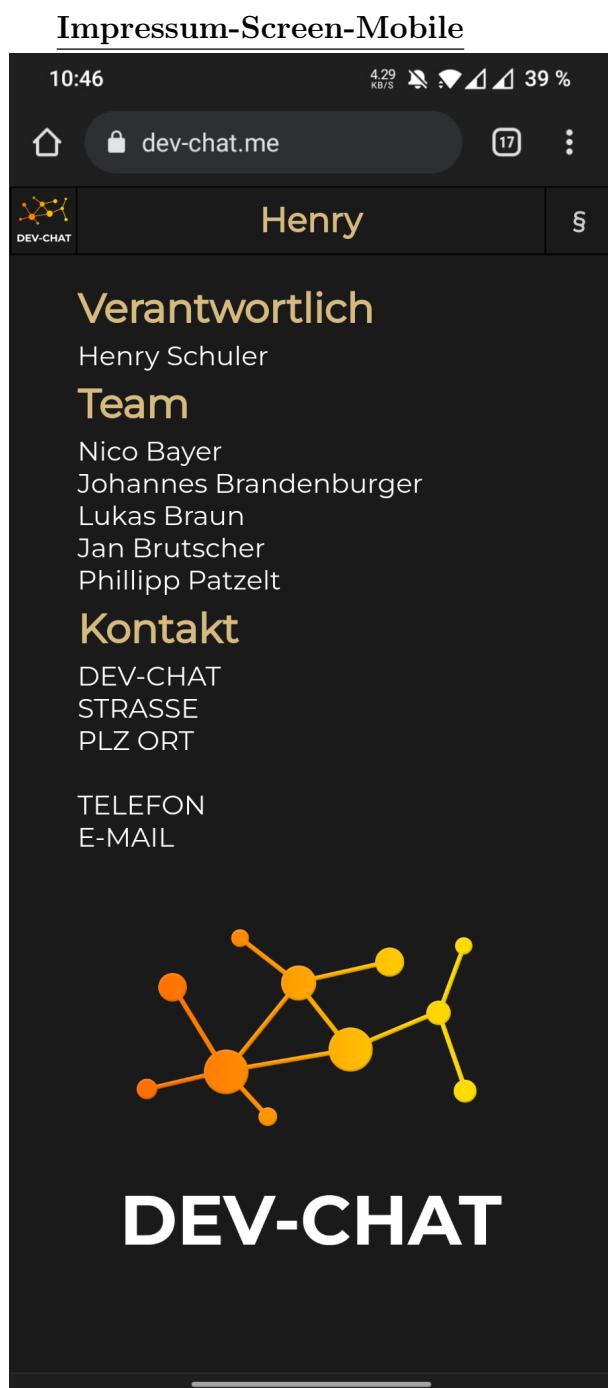
**Tickets**

Submitter	Created on	State	Change State	Info
admin	16/01/2021 21:23:12	Done	Set To Do	View
henry	25/01/2021 16:59:59	To Do	Set Done	View

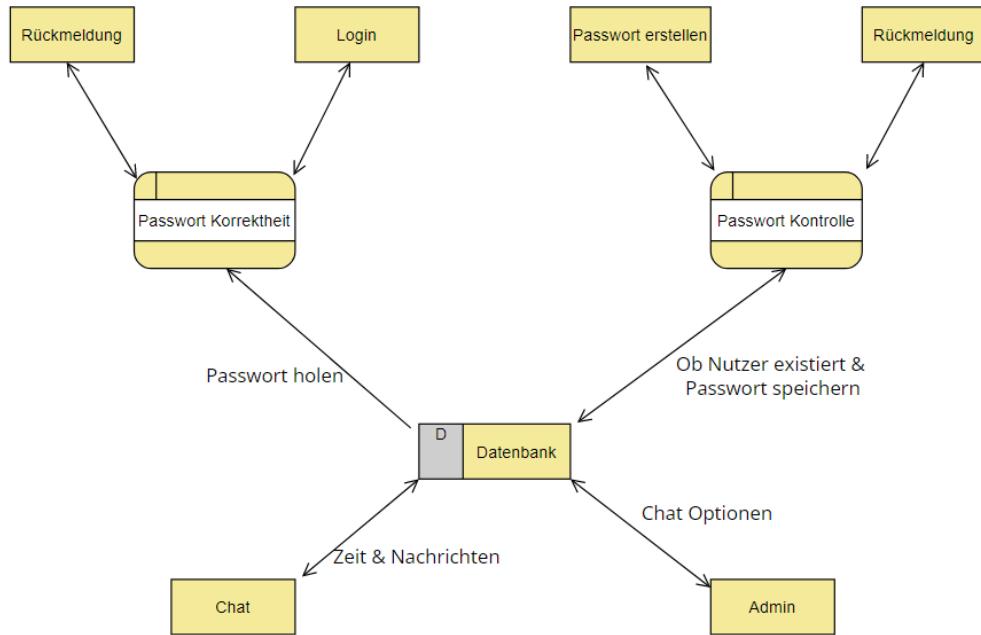
DEV-CHAT







## 2.5 DatenFlussdiagramm



## 3 Entwurf

### 3.1 Auswahl der Klassenbibliotheken/Frameworks

Die Webanwendung wird in HTML und JavaScript programmiert. Um Serverseitige Scripts auszuführen wird das Node.js Framework genutzt. Für Ausgabe von User-Interface-Komponenten wird die JavaScript-Bibliothek React genutzt. Um weitere Funktionalitäten von React zu nutzen und für eine verbesserte Performance wird zusätzlich das Next.js Framework eingesetzt.

### 3.2 Design Patterns für relevante Problemstellungen

Allgemein wird die Struktur des Framework Next.JS übernommen. Hier werden für die verschiedenen Unterseiten der APP Klassen erstellt, die von der Basisklasse `React.Component` erben. Diese Klassen stellen die View dar. Da die verschiedenen Seiten oft ähnliche Events im Hintergrund auslösen müssen, gibt es zusätzlich den FrontEnd-Controller. Alle UI-Events in den React-Komponenten besitzen eine Instanz des FrontEnd-Controllers und rufen dessen Funktionen auf.

Der FrontEnd-Controllers triggert die https-Requests, welche mit den in Next.JS integrierten API-Routern abgefangen werden. Diese fungieren somit als Bridge zwischen Front- und BackEnd. Auf dem Node-JS-Server (BackEnd) gibt es eine BackEnd-Controller-Klasse. Die Bridge-Routen rufen Methoden dieses BackEnd-Controllers auf. Diese Controller-Klasse übernimmt alle Logikoperationen (z.B. Benutzer verifizieren) und Datenbankzugriffe, sodass sicherheitskritische Daten nicht im FrontEnd eingesehen oder modifiziert werden können. Der BackEnd-Controller fungiert somit als Controller, aber auch als Schnittstelle zur Datenbank.

Besonders auf Modularität geachtet wird bei den Kurzbefehlen/Commands. Es gibt eine Basis-Klasse `Command`, von der neue Commands erben und dabei mindestens zwei Bestandteile ausformuliert werden müssen:

- `callString`: Ein String-Property, welches als Trigger für den Command dienen soll (z. B.: /survey für das Erstellen einer Umfrage)
- `execute()`: Eine Methode, welche den eigentlichen Befehl ausführt und die Antwort davon in Form eines Arrays von Chat-Nachrichten zurückgibt.

Der Backend-Controller besitzt ein `command`-Attribut in Form eines Arrays, in dem jeweils ein Objekt von jeder Command-Klasse enthalten ist. Außerdem hat er eine `executeCommand()`-Methode, welche alle Objekte in dem `command`-Array durchgeht und beim Erkennen eines passenden Callstrings den jeweiligen Befehl ausführt.

Somit ist Modularität gegeben, da die Commands alle den gleichen Aufbau haben und nur in den Code importiert und dem `.command`-Array hinzugefügt werden müssen.

### 3.3 Definition der Software-Komponenten

- Frontend (Client)
  - Main React-Component (Seite)
  - Login React-Component (Seite)
  - Register React-Component (Seite)
  - Admin Settings React-Component (Seite)
  - Password React-Component (Seite)
  - Chat React-Component (Seite)
  - Impressum React-Component (Seite)
  - Pop-Up React-Component
  - Header React-Compoent (auf jeder Seite eingebunden)
- Backend (Server)
  - Node.js-Server
  - Supabase-Datenbank

## 4 Implementierung

### 4.1 Code Conventions

Die Programmierparadigmen und Code Conventions werden aus dem „Google TypeScript Style Guide“ übernommen. Dies ist ein Regelwerk für TypeScript-Code und wird unter anderem von Google benutzt. (Es ist jedoch kein offizielles Google-Produkt, sondern wurde von den Node.js-Entwicklern des Konzerns veröffentlicht). Die Guidelines können hier nachgelesen werden.

### 4.2 IT-Security

Um den Usern maximale Sicherheit zu ermöglichen, werden folgende Technologien eingesetzt. Die Passwörter werden gehashet in der Datenbank abgelegt. Eine Verbindung auf die Website ist nur mit HTTPS möglich. Userverifizierung zur wird ein JSON Web Token genutzt.

Ein JSON Web Token ist ein nach RFC 7519 genormtes Acess Token, es ermöglicht, Daten sicher zwischen zwei Parteien auszutauschen. Es enthält dabei alle wichtigen Informationen über eine Entität, wodurch keine Datenbankabfrage erforderlich ist und die Sitzung nicht auf dem Server gespeichert werden muss. Dieses Token wird beim Nutzer im Browser gespeichert. Des Web Token wird vom Server der es generiert mit einer Signatur versehen. Diese stellt bei einem Autentifizierungsvorgang sicher, dass das Web Token nicht verändert wurde.

### 4.3 Frontend Backend Verbindung

Die Abbildung unten stellt die Struktur von FrontEnd und BackEnd gegliedert nach Dateien dar.

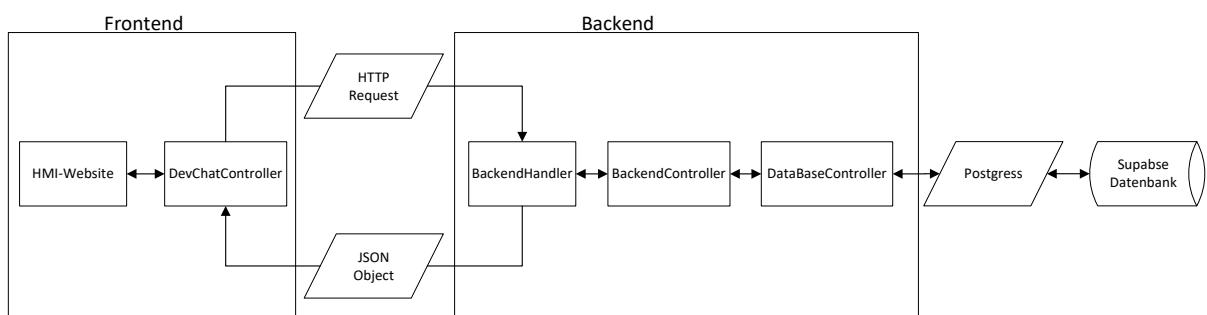


Abb. 4.1: FrontEnd-BackEnd-Connection Diagramm

## 4.4 Auswertung der Sprints

Nach Abschluss der drei Sprints, zur Fertigstellung des Produkts, werden Sprintreviews durchgeführt. Dabei werden die Ergebnisse des vergangenen Sprints durch die jeweiligen Programmierer vorgestellt. Außerdem wird der abgeschlossene Sprint reflektiert und Verbesserungen für den kommenden Sprint gesammelt und zusammengefasst.

### 4.4.1 Sprint 1 (KW 3+4)

Mit Abschluss des Sprint 1 stehen folgende Funktionen nun zur Verfügung:

- UI-Mockups für alle HTML Pages
- Datenbankstruktur für sämtliche Funktionalitäten
- Ein- und Ausloggen, sowie Registrieren von Benutzern
- Erstellen eines ThreeWords für den ChatKey
- Schreiben von Nachrichten in einem Chatraum (ohne Differenzierung nach ChatKey)
- Automatischer Abruf und Anzeige von Nachrichten im Chatraum

Folgende Verbesserungen werden für den nächsten Sprint umgesetzt:

- Das Mergen aller bearbeiteten Tasks in den develop Branch auf GitHub am Ende des Sprints wird ersetzt durch ein neues Merge-Konzept
  - Nach Bearbeiten eines Task wird direkt ein Pull Request auf develop eröffnet
  - Nach Abschluss des Reviews durch einen weiteren Programmierer wird unmittelbar auf develop gemerget
  - Es ist auch während dem bearbeiten des Sprints fortlaufend ein aktueller Stand mit aktuellen Änderungen auf Develop, von dem neue Branches erstellt werden können
  - Ein Chaos am Ende des Sprints wird vermieden, da während des Sprints bereits auf Änderungen reagiert werden kann
- Es müssen feste Zeitpunkte definiert werden, zu denen alle Änderungen des Sprints auf der develop Branch vorliegen müssen
- Zu Beginn des folgenden Sprints müssen alle dafür notwendigen Anforderungen im Dev-Ops angelegt sein (alle Tasks müssen geschätzt und zugewiesen sein)
- Eventuell müssen Prioritäten im Dev-Ops angelegt werden, damit Abhängigkeiten während des Sprints beachtet werden

### 4.4.2 Sprint 2 (KW 5+6)

Mit Abschluss des Sprint 2 stehen folgende Funktionen zur Verfügung:

- Überprüfung Username + Password requirements, Pop-Up, Button auf Main-Page, Befehle sehen nur der Nutzer selbst

- Befehle Survey Show Vote, Template Kurzbefehl
- Raumerstellen: Chat-Key speichern, Beitreten chat-Key überprüfung, chat-key aus vergangenheit werden gelöscht in DB, Password ändern implementiert, /expire (auf datum, nicht Zeit + Amerikanisches Datum)
- Befehlserlaubnis via AccessLevel, Help-Text (schon fertig aus S1), /calc, /report (Bug einreichen) + change solved state (testweise)
- CSS für Registrieren, PW ändern, Admin
- Seitenbasierter Zugriff nach Login/Chat-Key (token), Hover-Funktion für optisches Feedback

Folgende Verbesserungen werden für den nächsten Sprint umgesetzt:

- Bugfixes ins Dev-Ops aufnehmen
- Teaminterne Kommunikation

## 5 Test

## 6 Abnahme

## 7 Betrieb

## 7.1 Zeitliche Dokumentation

In diesem Kapitel werden die Treffen vor dem eigentlichen Entwicklungsprozess dokumentiert.

Datum	Teilnehmer	Themen
02.11.2021	Nico Bayer, Johannes Brandenburger, Lukas Braun, Jan Brutscher, Phillipp Patzelt, Henry Schuler	Planungsbeginn Projekthandbuch (Kapitel 1), Rollenvergabe (Abb. 1.1), Aufsetzen der Dokumentationsumgebung
03.11.2021	Nico Bayer, Johannes Brandenburger, Lukas Braun, Jan Brutscher, Phillipp Patzelt, Henry Schuler	Klarstellung Lasten-/Pflichtenheft, Beginn Backlog-Erstellung
12.11.2021	Nico Bayer, Johannes Brandenburger, Lukas Braun, Jan Brutscher, Phillipp Patzelt, Henry Schuler	Backlog ausformulieren, User-Storys ausfüllen, Einführung in React, Arbeitspaket- und Zeitplan
18.11.2021	Nico Bayer, Johannes Brandenburger, Lukas Braun, Jan Brutscher, Phillipp Patzelt, Henry Schuler	Scrum-Poker
23.11.2021	Nico Bayer, Johannes Brandenburger, Lukas Braun, Jan Brutscher, Phillipp Patzelt, Henry Schuler	Scrum-Poker, Kapitel 1.1.3
24.11.2021	Lukas Braun, Henry Schuler	To-Do Excel Liste erstellt

Datum	Teilnehmer	Themen
06.12.2021	Nico Bayer, Johannes Brandenburger, Lukas Braun, Jan Brutscher, Phillipp Patzelt, Henry Schuler	Vorstellen der Ergebnisse der Planungsphase, Aufteilen der Aufgaben zur Definitionsphase
19.12.2021	Nico Bayer, Johannes Brandenburger, Lukas Braun, Jan Brutscher, Phillipp Patzelt, Henry Schuler	Fragen klären
10.01.2022	Nico Bayer, Johannes Brandenburger, Lukas Braun, Jan Brutscher, Phillipp Patzelt, Henry Schuler	Endbesprechung Definitionsphase
14.01.2022	Nico Bayer, Johannes Brandenburger, Lukas Braun, Jan Brutscher, Phillipp Patzelt, Henry Schuler	Besprechung Beginn erster Sprint, Überprüfen der Aufgabenverteilung für ersten Sprint
31.01.2022	Nico Bayer, Johannes Brandenburger, Lukas Braun, Jan Brutscher, Phillipp Patzelt, Henry Schuler	Besprechung Beginn zweiter Sprint, Überprüfen der Aufgabenverteilung für zweiten Sprint, Scrum-Poker
01.02.2022	Nico Bayer, Johannes Brandenburger, Lukas Braun, Jan Brutscher, Phillipp Patzelt,	Sprint Review, Retrospective

Datum	Teilnehmer	Themen
	Henry Schuler	
07.02.2022	Nico Bayer, Johannes Brandenburger, Lukas Braun, Jan Brutscher, Phillipp Patzelt, Henry Schuler	Scrum-Meeting
07.02.2022	Nico Bayer, Johannes Brandenburger, Lukas Braun, Jan Brutscher, Phillipp Patzelt, Henry Schuler	Scrum Meeting, Scrum-Poker
16.02.2022	Nico Bayer, Johannes Brandenburger, Lukas Braun, Jan Brutscher, Phillipp Patzelt, Henry Schuler	Scrum Meeting, Scrum Review, Retrospective
21.02.2022	Nico Bayer, Johannes Brandenburger, Lukas Braun, Jan Brutscher, Phillipp Patzelt, Henry Schuler	Scrum Meeting

### 7.1.1 02.11.2021

- Rollenverteilung
  - Henry Schuler: Project-Owner
  - Jan Brutscher: Scrum-Master
- Overleaf – L<sup>A</sup>T<sub>E</sub>X als Dokumentationsumgebung

### 7.1.2 03.11.2021

- Lastenheft ist ein Dokument (Bereits per E-Mail an Kegreiß gegangen (Projektentreichung))
- Backlog ist unser Pflichtenheft – User-Stories + Darstellung als Usecase-Diagramm (Vorlesung 27.10.2021)
  - Backlog werden gepflegt, sobald Rückmeldung zu Lastenheft da ist
  - Hierarchie (Epic, User-Story, Task) festgelegt, erste allgemeine Punkte bereits eingepflegt
- Ziel ist, Backlog/Pflichtenheft/... mal gemacht zu haben -> nicht übertrieben ins Detail gehen

### 7.1.3 12.11.2021

- Scrum-Poker + Aufgabenzuordnung im kommenden Termin am Donnerstag 18.11. 9 Uhr
- Henry Schuler füllt die User-Storys bis zum nächsten Termin und erstellt eventuell die zugehörigen Tasks
- Johannes Brandenburger informiert sich weiter über React zur späteren Evaluierung der zu nutzenden Plattform

## 7.2 18.11.2021

- Scrum-Poker mit allen. Zuweisung von Wertigkeiten für die ersten Tasks.

## 7.3 23.11.2021

- Scrum-Poker mit allen. Zuweisung von Wertigkeiten für die restlichen Tasks.
  - Eventuell Rauswurf der Minigames da zu zeitaufwändig (Maximal Tick Tack Toe) -> Frage Kegreiß wird.
- Nachfrage was unter Qualitätsmanagement und Konfigurationsmanagement verstanden (Punkte im Projekthandbuch)

## 7.4 24.11.2021

- Excel To-Do Liste für Planungs- und Definitionsphase erstellt.
  - Nächstes Treffen: Aufteilen der To-Dos auf das Team (Planungs-/Definitionsphase)

## 7.5 06.12.2021

- Übertragen der Ergebnisse der Planungsphase ins Latex -> Abschicken
- QMM
  - Freigabe für den Kurs / Test durch Henry Schuler/Jan Brutscher
  - Unterstützt der DEV-CHAT mehrere Browser gleichzeitig / mehrere Login-Sessions eines Benutzers?
- Es soll eine Bug-Submit-Funktion geben -> Einreichen von Bugs durch Benutzer

## 7.6 19.12.2021

- Gibt es Unit-Tests (QMM)?

## 7.7 10.01.2021

- Fragen klären

## 7.8 14.01.2021

- Besprechung Beginn erster Sprint
- Überprüfen der Aufgabenverteilung für zweiten Sprint

## 7.9 31.01.2021

- Besprechung Beginn zweiter Sprint
- Scrum-Poker für neue Aufgaben

## 7.10 01.02.2021

- Sprint Review:
  - Vorstellung der erledigten Aufgaben durch die jeweiligen Entwickler für das Team und Vorstellung des aktuellen Stands durch Product Owner
- Retrospective:
  - Änderung der Branchaufteilung in Github um Übersichtlichkeit zu fördern und um Mergen am Sprintende zu vereinfachen.

- Neues Vorgehen: Nach jedem Vollenden eines Tasks durch anderes Teammitglied abnehmen lassen und dann direkt mergen.

### **7.11 07.02.2021**

- Scrum-Poker für neue Aufgaben
- Scrum Meeting

### **7.12 11.02.2021**

- Scrum Meeting
- Scrum-Poker für neue Aufgaben

### **7.13 16.02.2021**

- Scrum Meeting
- Sprint Review
- Retrospective

### **7.14 21.02.2021**

- Scrum Meeting