



# PROJEKT MODUL 307



Yago Romero

## Inhalt

1 Formular des Projekts .....	3
2 Einleitung.....	4
3 Anforderungen .....	4
4 Schnittstellen zu anderen Systemen .....	4
5 Architektur .....	5
5.1 Komponenten.....	5
6 Das Projekt .....	7
6.1 Die Vorbereitung der Entwicklungsumgebung .....	7
6.2 Gestaltung des Designs .....	10
6.3 Mockup.....	10
6.4 UML Diagramm .....	19
7 Test.....	19
8 Benutzerhandbuch .....	24
8.1 Sich registrieren.....	24
8.2 Sich einloggen .....	25
8.3 Chatten .....	27
8.4 Navigieren .....	28
8.5 Chats löschen .....	29
8.6 Profil editieren.....	29
9 Arbeitsjournal.....	30
10 Versionen Kontrolle.....	31
Abbildung 1 Auftrag Formular.....	3
Abbildung 2 Node herunterladen .....	8
Abbildung 3 Angular installieren.....	8
Abbildung 4 Angular Version.....	8
Abbildung 5 Firebase Projekt erstellen .....	9
Abbildung 6 Firebase Projekt vorbereiten .....	9
Abbildung 7 Control Panel Firebase Projekt .....	10
Abbildung 8 Typescript Installation.....	10
Abbildung 9 Firebase Installation.....	10
Abbildung 10 Login Mockup.....	11
Abbildung 11 Registrierung Mockup.....	12
Abbildung 12 User List Mockup .....	13
Abbildung 13 User Profile Mockup .....	14
Abbildung 14 Chat Mockup.....	15
Abbildung 15 Profile Mockup.....	16
Abbildung 16 Edit Profile Mockup.....	17
Abbildung 17 Chat liste Mockup .....	18

Abbildung 18 UML Diagramm ChatApp .....	19
Abbildung 19 Profile edditig.....	24
Abbildung 20 Registrieren.....	25
Abbildung 21 Einloggen.....	26
Abbildung 22 Google Login .....	26
Abbildung 23 Password zurücksetzen.....	27
Abbildung 24 Benutzer Liste .....	27
Abbildung 25 Chat Knopf .....	27
Abbildung 26 Chat.....	28
Abbildung 27 Menu.....	28
Abbildung 28 Chat löschen.....	29
Abbildung 29 Menu Profil .....	29
Abbildung 30 Profil.....	29
Abbildung 31 Profil editieren .....	30
Abbildung 32 Authentifikation durch Firebase .....	31
Abbildung 33 Chat Datenbank .....	32
Abbildung 34 Benutzern Datenbank .....	32

# 1 Formular des Projekts

## M307- Interaktive Webseite mit Formular erstellen

### Auftrag von Marcel Arnold an die Schüler der INF17s

Ziel dieses Auftrages ist es sämtliche Handlungsziele vom Modul 307, mit Anwendung der handlungsnotwendigen Kenntnisse, überprüfen zu können. Weiter bietet dieser Auftrag den Schülern eine hervorragende Vorbereitung für ihre PA (Praktische Arbeit), da dieser Auftrag im Projektauftrag eine sehr ähnliche Struktur aufweist.

Kundenanforderungen:

- Der Schüler darf sein Thema frei wählen, muss jedoch sämtliche hier definierten Kundenanforderungen umsetzen.
- Zum Abgabetermin muss eine Dokumentation und eine Applikation abgegeben werden.
- Die Dokumentation muss folgendes beinhalten:
  - o Einleitung und Projektbeschrieb
    - 1:1 Kopie dieses Auftrages
    - Idee und Umsetzungsziele
  - o Anforderungsspezifikation (IEEE 830\*)
    - Ableitung der Entwicklungsanforderungen aus diesen Kundenanforderungen
    - Aufteilung in funktionale und nicht funktionale Anforderungen
  - o Designbeschreibung (IEEE 1016\*)
    - Systemübersicht
    - Struktur- und Verhaltensdiagramme gemäss UML (Unified Modelling Language)
  - o Testkonzept (IEEE 829\*)
    - Testspezifikation mit Testfällen aus den Anforderungen abgeleitet
    - Testbericht mit Testprotokoll und Abweichungen
  - o Benutzerhandbuch
- Die Applikation muss folgendes beinhalten:
  - o Mindestens ein Formular mit Client und Serverseitiger Validierung sämtlicher Eingabefelder:
    - Mehrere einzeilige und mehrzeilige Eingabefelder
    - Mindestens drei verschiedene Typen zur Eingabe von Text
    - Mindestens ein Typ zur Eingabe von Zahlen
    - Mindestens ein Typ zum Treffen einer Auswahl
    - Mindestens zwei Buttons zum Absenden und Zurücksetzen des Formulars
  - o Die Gliederung und Struktur des Codes ist sinnvoll gewählt
- Kommentare erklären Funktionalität und Abschnitte des Codes

Hilfestellung:

- Im bekannten «\_Programm»-Ordner auf GitLab gibt es im «\_programm.txt», sowie im «\_hausaufgabe.txt» den Grob-Projektplan mit aktuellen Wochenzielen. Ich rate euch diese wöchentlichen Ziele stetig zu erarbeiten damit ihr nicht in Verzug gerät.

**Abgabetermin: 07.06.2019 um spätestens 08:00 via Email, zu späte Abgabe gibt 1 Note Abzug.**

Abbildung 1 Auftrag Formular

## 2 Einleitung

Da ich sehr in Angular interessiert bin, habe ich entschlossen eine Angular<sup>1</sup> basierte App zu entwickeln. Da ich noch nicht viel Erfahrung mit Mobile Apps habe, wird es eine einfache Web App sein. Um es ein wenig interessanter zu machen werde ich eine Chat App entwickeln, dafür werde ich verschiedene Funktionalitäten der Plattform Firebase benutzen und das GUI werde ich mithilfe CSS und SASS gestalten.

## 3 Anforderungen

Kundenanforderungen:

Entwicklung einer Chat App mit folgenden Anforderungen:

- Der Benutzer kann sich mit email und Password, sowie mit Google Account einloggen.
- Der Benutzer kann mit anderen Nutzern per Chat schriftlich kommunizieren.
- Jeder Benutzer hat ein eigenes Profil.

Die funktionalen Anforderungen für dieses Projekts lauten:

- Benutzer kann sich mit Benutzername und Password Einloggen.
- Benutzer kann sich mittels Google Account Einloggen.
- Benutzer kann sich ausloggen, indem er auf einen „Sign. Out“ Knopf drückt.
- Nach dem Registrieren wird ein Einloggen erforderlich sein.
- Nach dem Registrieren wird am Nutzer eine Verifikationsemail geschickt.
- Nach dem Einloggen wird der Benutzer zur Chat Seite weitergeleitet.
- In der Chat Seite kann der Benutzer zu allen Nutzern zugreifen, um mit denen zu chatten.
- Benutzer muss folgende Daten angeben, um ein Konto zu erstellen:
  - Email
  - Password: Nicht kürzer als 6 Zeichen und nicht länger als 25 Zeichen
  - Gender: Muss zwischen männliches und weibliches, mittels zwei Radio Buttons, wählen (nicht pflichtig)
- Wenn der Benutzer einer dieser Felder nicht korrekt ausfüllt, kann er kein Konto erstellen.
- Falls der Benutzer ein falsches Password und/oder Email, während den Login eingibt, bekommt er eine Fehler Meldung über.

Nicht funktionale Anforderungen:

- Grafische Oberfläche ist attraktiv und ansprechend.

## 4 Schnittstellen zu anderen Systemen

Diese App wird die Authentifikation, sowie die Datenbank Funktion von Google Firebase benutzen. Sie werden sich um das Back-End kümmern.

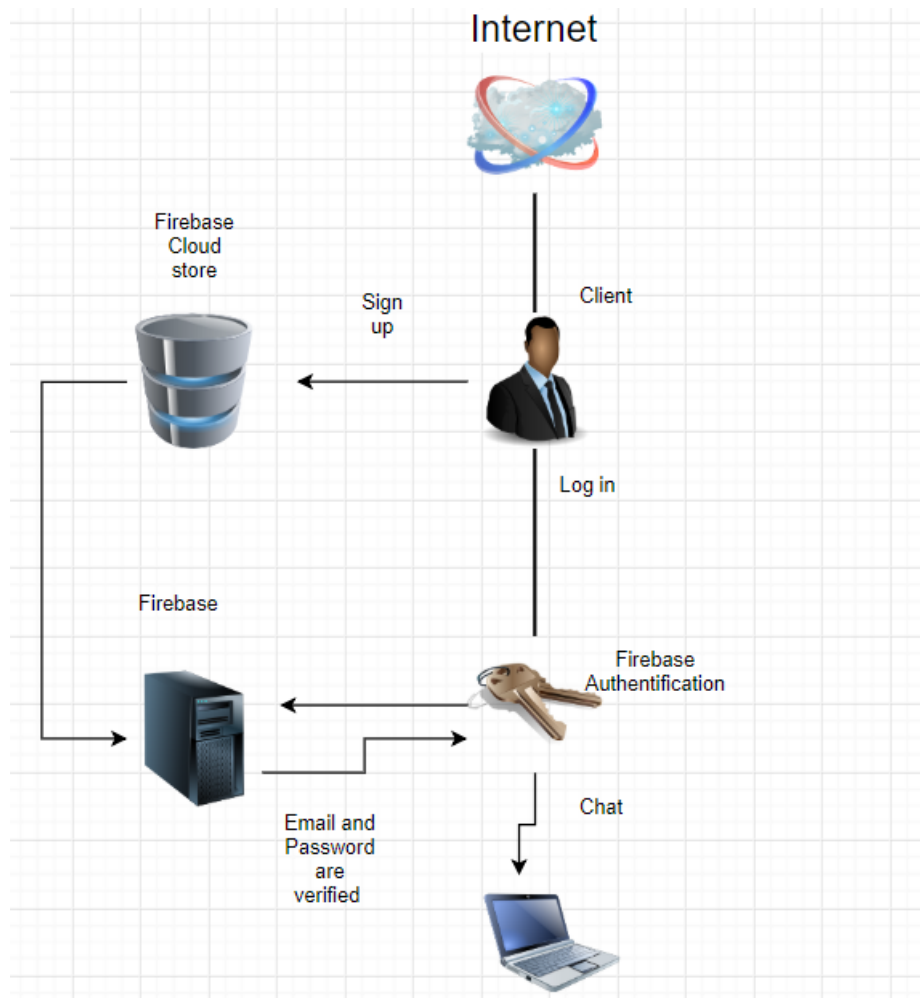
---

<sup>1</sup> Ich werde Angular und nicht Angular.js verwenden. Obwohl sie sehr ähnlich sind, finde ich, dass Angular einfacher ist.

Die ganze Logik wird mit Hilfe von Angular/Fire möglich.

## 5 Architektur

Ich verwende für dieses Projekt den MVC sowohl, wie das Client/Server style:



### 5.1 Komponenten

Die Komponenten sind folgende:

- Auth
  - Login: Anmeldekompontenten:
    - login.component.ts
    - login.component.html
    - login.component.css
  - Register: Kontoherstellungskomponenten:
    - register.component.ts
    - register.component.html
    - register.component.css
  - Reset-password: Password Wiederherstellungskomponenten:
    - reset-password.component.ts
    - reset-password.component.html
    - reset-password.component.css

- auth.module: Beinhaltet die Logik für die funktionsweise der anderen Komponenten.
- Chat
  - chat-detail:
    - chat-detail.component
    - chat-detail.component.html
    - chat-detail.component.css
  - chat-feed
    - chat-feed.component
    - chat-feed.component.html
    - chat-feed.component.css
  - chat-input:
    - chat-input.component
    - chat-input.component.html
    - chat-input.component.css
  - chat-list:
    - chat-list.component
    - chat-list.component.html
    - chat-list.component.css
  - chat-message:
    - chat-message.component
    - chat-message.component.html
    - chat-message.component.css
  - chat-messages:
    - chat-messages.component
    - chat-messages.component.html
    - chat-messages.component.css
  - chat-thread:
    - chat-thread.component
    - chat-thread.component.html
    - chat-thread.component.css
  - chat-threads:
    - chat-threads.component
    - chat-threads.component.html
    - chat-threads.component.css
  - chat.module
  - message.model: Classe für message
  - message.service
  - thread.model: Klasse für thread
  - thread.service
- core:
  - auth.service: Authentifikationsfunktionen und Logik
  - core.module
- shared: Alle Komponenten die von anderen Komponenten gebraucht werden:
  - navbar: Navigation Komponenten der Applikation
    - navbar.component
    - navbar.component.html
    - navbar.component.css

- upload: Hochladen von Bildern
    - upload.component
    - upload.component.html
    - upload.component.css
    - upload.service.ts
  - from-now.pipe
  - shared.module
- user
  - user-dashboard
    - user-dashboard.component
    - user-dashboard.component.html
    - user-dashboard.component.css
  - user-detail:
    - user-detail.component
    - user-detail.component.html
    - user-detail.component.css
  - user-list:
    - user-list.component
    - user-list.component.html
    - user-list.component.css
  - user-list-item:
    - user-list-item.component
    - user-list-item.component.html
    - user-item.component.css
  - user.model
  - user.module
  - user.service
- app.component.ts
- app.component.html
- app.component.ts
- app.module.ts
- material.module.ts
- routing.guard.ts
- routing.module.ts

## 6 Das Projekt

### 6.1 Die Vorbereitung der Entwicklungsumgebung

Da ich mit Angular arbeiten möchte, muss ich erst die Entwicklungsumgebung vorbereiten. Dafür muss ich folgendes installieren: Node.js, Angular/cli, Typescript und das Firebase SDK. Ausserdem muss ich eine neue Projektmappe bei Firebase.com herstellen, damit sich die Applikation mit den Firebase Servern kommuniziert.



1. Auf Node.js Homepage gehen und Node.js herunterladen:



Abbildung 2 Node herunterladen

Um Inkompatibilitätsprobleme zu vermeiden, werde ich die Version 10.15.3 LTS benutzen, diese ist die letzte stabile Version.

2. Nach der Installation von Node.js werde ich mittels npm Befehl Angular herunterladen und installieren:

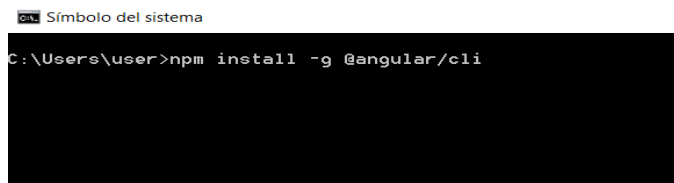


Abbildung 3 Angular installieren

Nachdem ich auf Enter drücke, wird Angular auf dem Rechner Installiert werde.

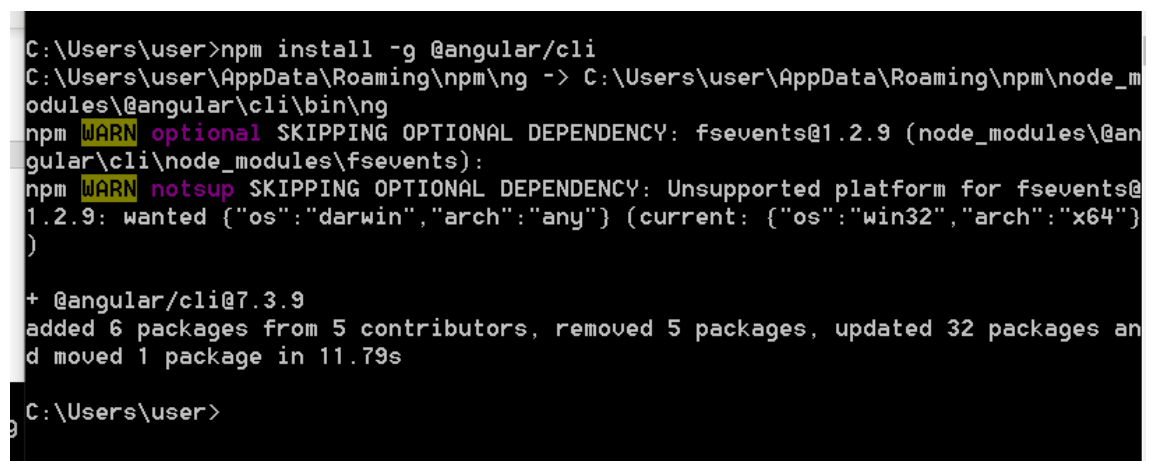


Abbildung 4 Angular Version

Wie man sehen kann, handelt es sich um die Version 7.3.9 von Angular.<sup>2</sup>

<sup>2</sup> Später habe ich das angular/cli auf die Version 8.0.1 aktualisiert, um einige Bugs zu korrigieren.

3. Als nächstes werde ich eine neue Projekt Mappe in Firebase erstellen damit meiner App, all die Firebase Funktionalitäten, die ich brauche, benutzen kann.

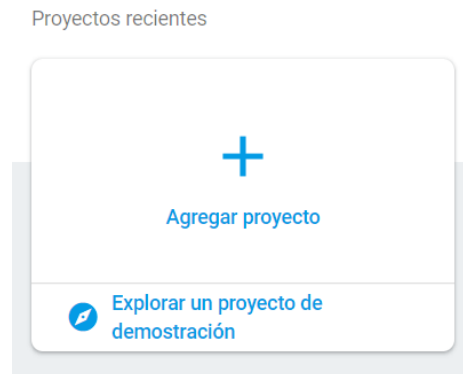


Abbildung 5 Firebase Projekt erstellen

Mit dem Klicken auf dem Kreuz, kann ich ein neues Projekt erstellen.

Abbildung 6 Firebase Projekt vorbereiten

Jetzt muss ich nur einem passenden Namen für meine App wählen und die Geschäftsbedingungen akzeptieren und auf Projekt Erstellen drücken.

4. Hier habe ich den Control Panel meiner App:

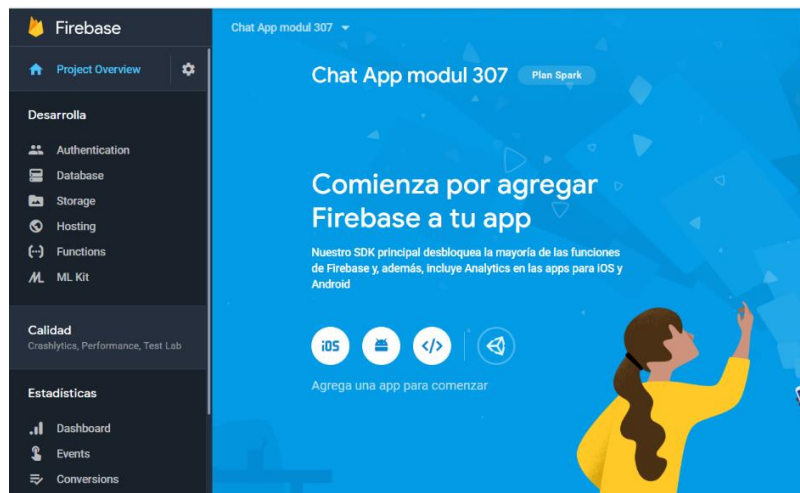


Abbildung 7 Control Panel Firebase Projekt

## 5. Installation von Typescript

```
> npm install -g typescript
```

Abbildung 8 Typescript Installation

## 6. Installation von Firebase

```
> npm i firebase
```

Abbildung 9 Firebase Installation

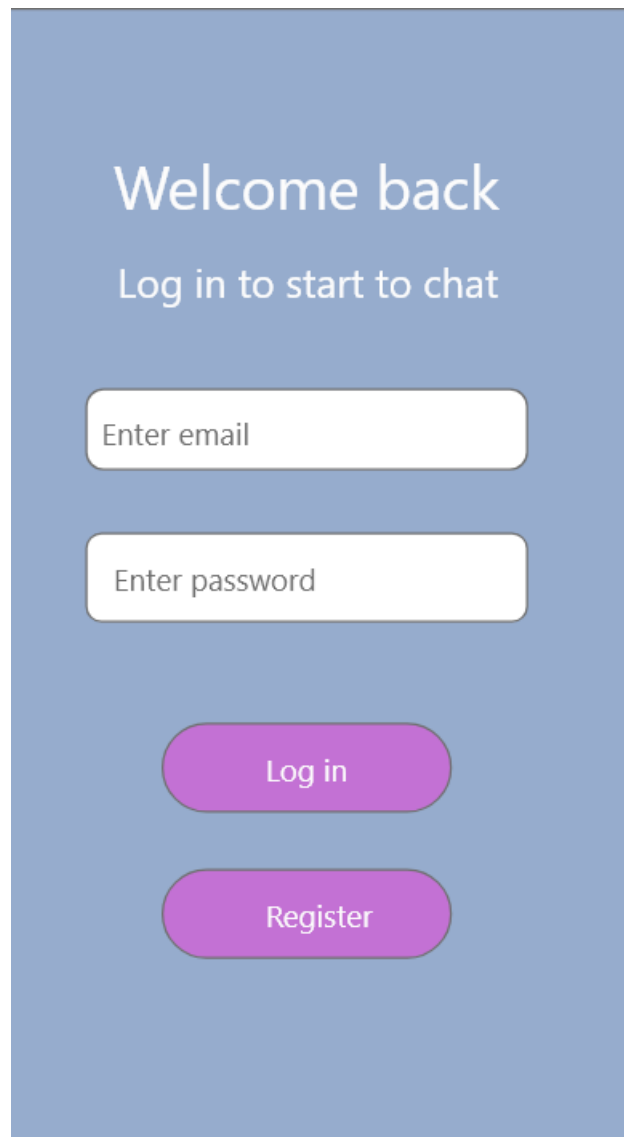
## 6.2 Gestaltung des Designs

Ich wollte ein modernes und attraktives Design gestalten, deshalb habe ich die angular-material Komponenten für meine App benutzt. Da das Design dieser Komponenten sich nach die Google-Material Standards orientieren, denn Angular wird von Google betreut, kann ich davon ausgehen, dass diese Komponenten eine hohe Qualität haben, sowie Kompatibel. Ausserdem vermeide ich so mögliche Fehler, die durch falsch programmierte Formulare entstehen können.

## 6.3 Mockup

Damit es mir leichter fällt meine App zu programmieren, habe ich einige Mockups gemacht, die den Fertigen Design entsprechen sollen.

- Log In:

A login screen mockup with a blue background. It features the text "Welcome back" in a large white font, followed by "Log in to start to chat" in a smaller white font. Below this are two white input fields with rounded corners, labeled "Enter email" and "Enter password". At the bottom are two purple buttons with rounded corners, labeled "Log in" and "Register".

Welcome back

Log in to start to chat

Enter email

Enter password

Log in

Register

*Abbildung 10 Login Mockup*

Nachdem der Benutzer sich eingeloggt hat, wird er zu dem Benutzer liste weitergeleitet und wenn er auf Register drückt, kommt er zur Registrierung Seite.

- Registrierung:

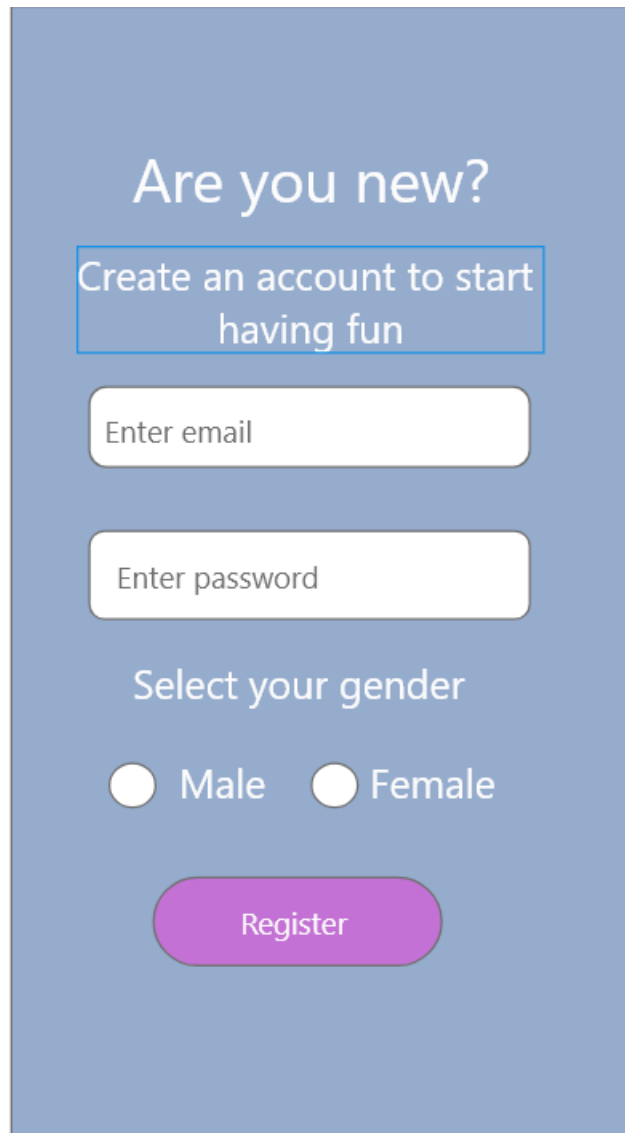
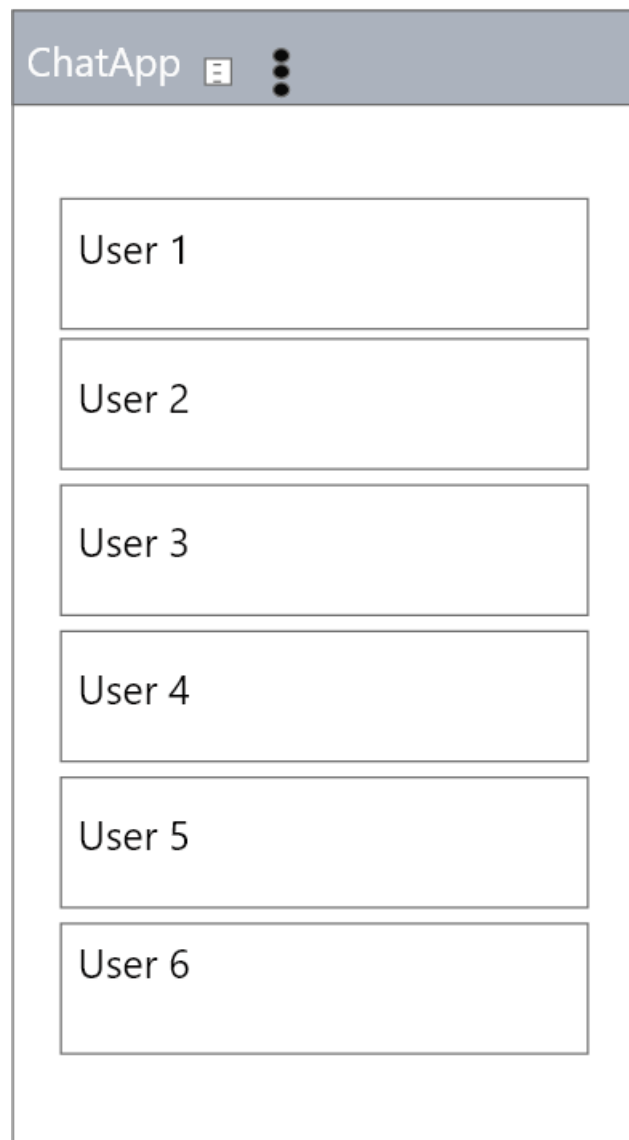
A registration form mockup on a blue background. At the top, it asks 'Are you new?' in white text. Below this, a blue-bordered box contains the text 'Create an account to start having fun'. There are two white input fields with rounded corners: the first is labeled 'Enter email' and the second is labeled 'Enter password'. Below the password field, it says 'Select your gender' in white text. Underneath, there are two radio buttons, one labeled 'Male' and one labeled 'Female'. At the bottom, there is a purple rounded rectangular button with the word 'Register' in white text.

Abbildung 11 Registrierung Mockup

Wenn der Benutzer auf Register drückt, bekommt er ein verifikationsemail und wird zur Login Seite weitergeleitet.

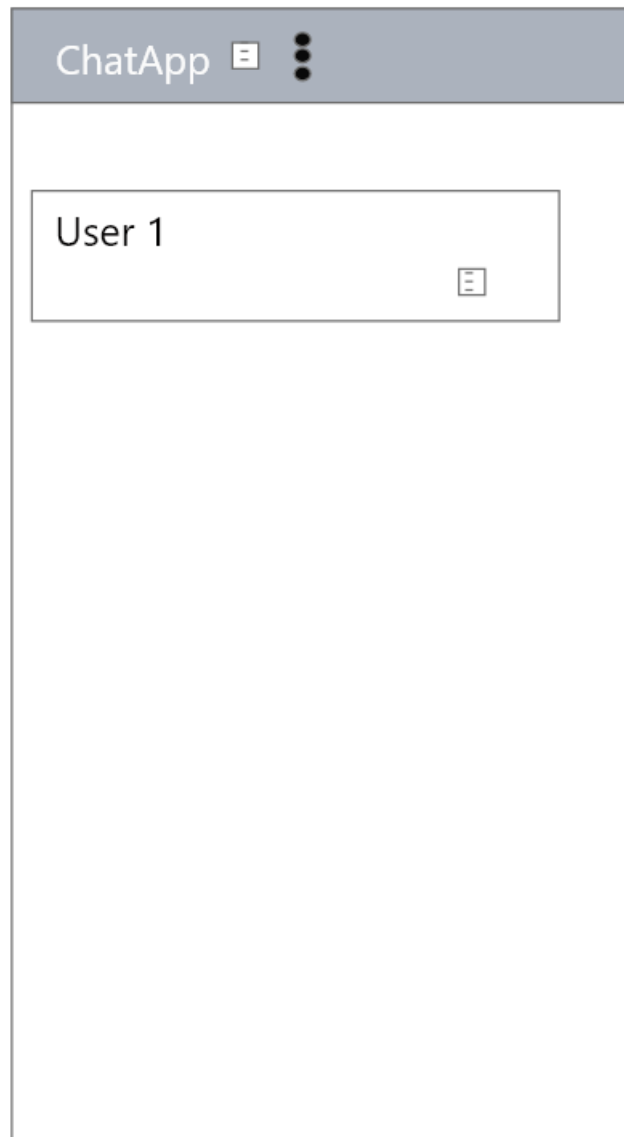
- Benutzer liste



*Abbildung 12 User List Mockup*

Hier kann der Nutzer auf einen beliebigen User der Liste klicken, um an der Profil Seite dieses Users zu gelangen. Von dort aus, kann er auf den Chat Knopf drücken, um ein Chat zu öffnen.

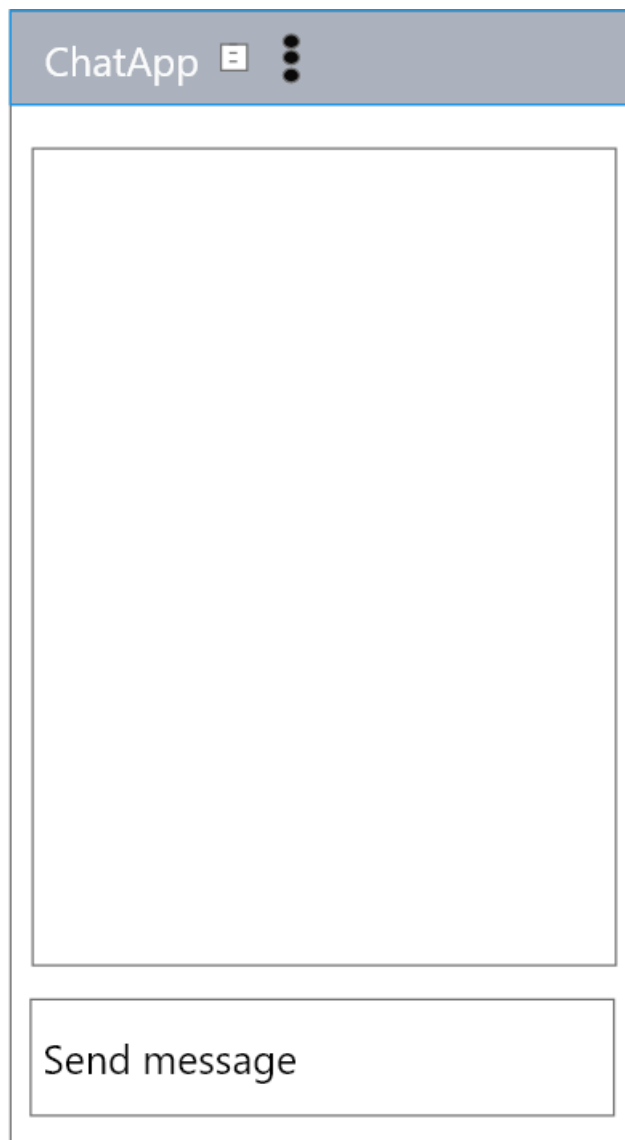
- User-profile



*Abbildung 13 User Profile Mockup*

Hier hat der Benutzer auf einen der User geklickt und wenn er nun auf das Chat Knopf drückt, beim User unten rechts, kann er einen Chat mit diesem User Beginen.

- Chat

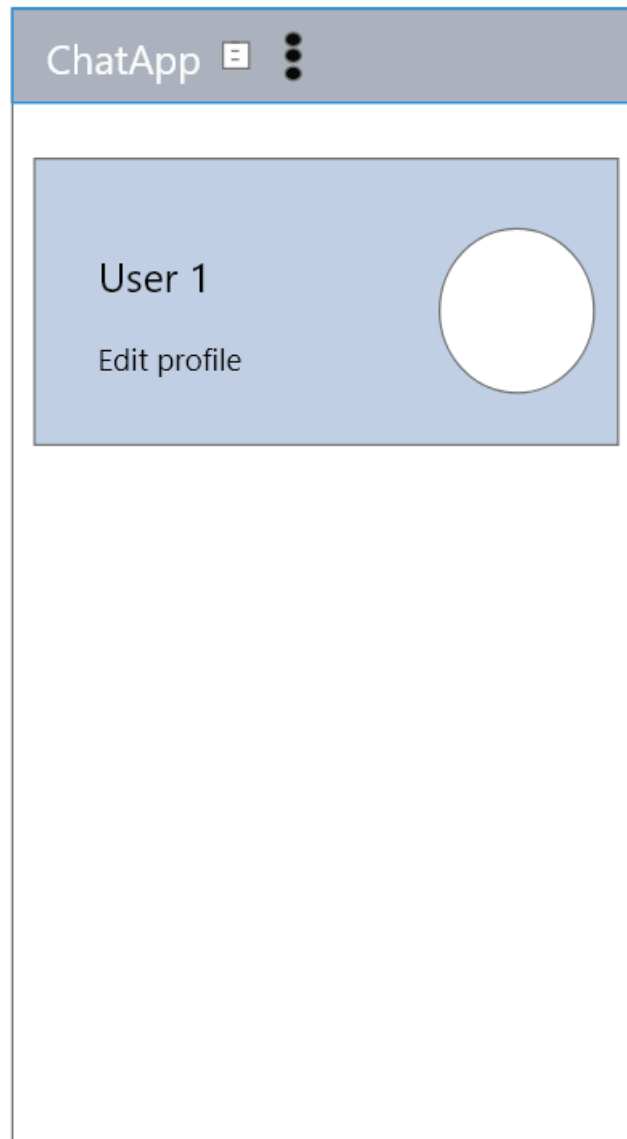


*Abbildung 14 Chat Mockup*

Das ist die Chat Seite, hier kann der Benutzer Nachrichten senden und Empfangen.



- Profile



*Abbildung 15 Profile Mockup*

Dies ist die Profile Seite des Eingeloggten Benutzers, er kann hier ein neues Profilbild zufügen oder die Profile Information ändern, indem er auf Edit Profile klickt.

- Edit-profile

ChatApp

Update name

Update email

Update username

Update username

Update location

Update age

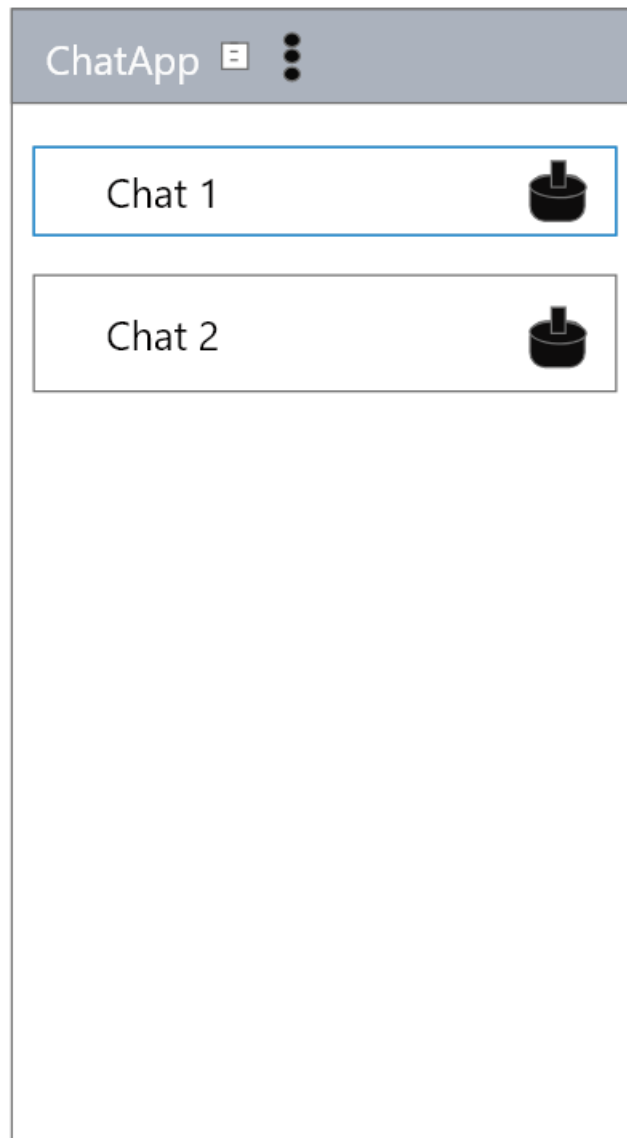
Update gender

Save

Cancel

Abbildung 16 Edit Profile Mockup

- Chat liste



*Abbildung 17 Chat liste Mockup*

Hier kann der Benutzer alle aktiven Chats ansehen und sie löschen.



**Erwartetes Resultat:** Console.log() email is badly formatted und Input soll rot angezeigt werden.

**Test Nr:** 1.0.1

**Typ:** White box

**Art:** Unit Test

**Was wird gemacht:** Register Knopf wird angeklickt während beide Inputs leer sind.

**Resultat:** Console.log() email is badly formatted, Inputs werden rot angezeigt.

**Erwartetes Resultat:** Console.log() email is badly formatted und beide Inputs sollen rot angezeigt werden.

**Test Nr:** 1.1.0

**Typ:** White box

**Art:** Unit Test

**Was wird gemacht:** Asdrdssarr wird im Email Input geschrieben, während Password Input leer ist. Danach wird auf Registrieren geklickt.

**Resultat:** Console.log() email is badly formatted, Input wird rot angezeigt.

**Erwartetes Resultat:** Console.log() email is badly formatted und Input soll rot angezeigt werden und Registrierung soll nicht möglich sein.

**Test Nr:** 1.1.1

**Typ:** White box

**Art:** Unit Test

**Was wird gemacht:** Asdrdssarr wird im Password Input geschrieben, während Email Input leer ist. Danach wird auf Registrieren geklickt.

**Resultat:** Console.log() email is badly formatted, Input wird rot angezeigt.

**Erwartetes Resultat:** Console.log() email is badly formatted und Input soll blau angezeigt werden und Registrierung soll nicht möglich sein.

**Test Nr:** 2

**Typ:** White box

**Art:** Unit Test

**Was wird gemacht:** [yagoromero50@gmail.com](mailto:yagoromero50@gmail.com) wird in Email Input geschrieben, während Password Input leer ist. Danach wird auf Registrieren geklickt.

**Resultat:** Console.log() The password must be 6 characters long or more, Input wird blau angezeigt.

**Erwartetes Resultat:** Console.log() The Password must be 6 characters long or more und Input soll blau angezeigt werden. Registrierung soll nicht möglich sein.

**Test Nr.:** 2.0

**Typ:** White box

**Art:** Unit Test

**Was wird gemacht:** 1 wird in Password Input geschrieben, während Email Input leer ist. Danach wird auf Registrieren geklickt.

**Resultat:** Console.log() email is badly formatted, Input wird rot angezeigt.

**Erwartetes Resultat:** Console.log() email is badly formatted und Input soll rot angezeigt werden und Registrierung soll nicht möglich sein.

**Test Nr: 2.0.1**

**Typ:** White box

**Art:** Unit Test

**Was wird gemacht:** password1 wird in Password Input geschrieben, während Email Input [yagoromero50@gmail.com](mailto:yagoromero50@gmail.com) beinhaltet. Danach wird auf Registrieren geklickt.

**Resultat:** Beide Inputs werden blau angezeigt. Console.log() Welcome, your account has been created! Und We sent you an email verification. E-Mail mit verifikationslink wird geschickt. Danach wird man zur Login Seite weitergeleitet.

**Erwartetes Resultat:** Inputs sollen blau angezeigt werden und ein einen verifikationsemail soll an die eingegebene Email Adresse geschickt werden und man soll zur Login Seite weitergeleitet werden.

**Test Nr: 2.1.1**

**Typ:** White box

**Art:** Unit Test

**Was wird gemacht:** Im Login Formular Registrieren wird geklickt.

**Resultat:** Man wird zur Registrierung Seite weitergeleitet.

**Erwartetes Resultat:** Man soll zur Registrierung Seite weitergeleitet werden.

**Test Nr: 3**

**Typ:** White box

**Art:** Unit Test

**Was wird gemacht:** Im Login Formular gibt man die Anmeldungsdaten ein und klickt auf Sign In.

**Resultat:** Man wird zu dem Benutzerliste Seite weitergeleitet.

unsafe:nullhttps://www.gravatar.com/avatar/cb60af7770d84063b66bb39edbd81fcd?d=identicon:1 GET

unsafe:nullhttps://www.gravatar.com/avatar/cb60af7770d84063b66bb39edbd81fcd?d=identicon net::ERR\_UNKNOWN\_URL\_SCHEME. Profilbilder werden nicht angezeigt

**Erwartetes Resultat:** Man soll zu dem Benutzerliste Seite weitergeleitet werden. Profilbilder sollen angezeigt werden.

**Test Nr: 3**

**Typ:** White box

**Art:** Unit Test

**Was wird gemacht:** In der Benutzerlist Seite wird ein Nutzer angeklickt.

**Resultat:** Man wird zum Profil dieses Nutzers weitergeleitet.

unsafe:nullhttps://www.gravatar.com/avatar/cb60af7770d84063b66bb39edbd81fcd?d=identicon:1 GET

unsafe:nullhttps://www.gravatar.com/avatar/cb60af7770d84063b66bb39edbd81fcd?d=identicon net::ERR\_UNKNOWN\_URL\_SCHEME. Profilbilder werden nicht angezeigt

**Erwartetes Resultat:** Man soll zum Profil dieses Nutzers Seite weitergeleitet werden. Profilbilder sollen angezeigt werden.

**Test Nr:** 3.0

**Typ:** White box

**Art:** Unit Test

**Was wird gemacht:** Beim Nutzer Profil klickt man auf Chat Knopf.

**Resultat:** Man wird zur Chat Seite weitergeleitet. Console.log() Thread created!

**Erwartetes Resultat:** Man soll zur Chat Seite weitergeleitet werden. Console.log() Thread created!

**Test Nr:** 3.0.1

**Typ:** White box

**Art:** Unit Test

**Was wird gemacht:** Man schickt eine Nachricht am Nutzer.

**Resultat:** ChatInputComponent.html:2 ERROR TypeError: Cannot read property 'doc' of undefined

at ChatInputComponent.saveLast (chat-input.component.ts:42)

at ChatInputComponent.send (chat-input.component.ts:35)

at ChatInputComponent.handleSubmit (chat-input.component.ts:47)

at Object.eval [as handleEvent] (ChatInputComponent.html:2)

at handleEvent (core.js:34777)

at callWithDebugContext (core.js:36395)

at Object.debugHandleEvent [as handleEvent] (core.js:36031)

at dispatchEvent (core.js:22519)

at core.js:33709

at HTMLInputElement.<anonymous> (platform-browser.js:1789)

View\_ChatInputComponent\_0 @ ChatInputComponent.html:2

proxyClass @ compiler.js:19199

logError @ core.js:36330

handleError @ core.js:7239

dispatchEvent @ core.js:22523

(anonymous) @ core.js:33709

(anonymous) @ platform-browser.js:1789

invokeTask @ zone-evergreen.js:391

onInvokeTask @ core.js:30873

invokeTask @ zone-evergreen.js:390

runTask @ zone-evergreen.js:168

invokeTask @ zone-evergreen.js:465

invokeTask @ zone-evergreen.js:1603

globalZoneAwareCallback @ zone-evergreen.js:1629

ChatInputComponent.html:2 ERROR CONTEXT DebugContext\_ {view: {...}, nodeIndex: 11,

nodeDef: {...}, elDef: {...}, elView: {...}}

View\_ChatInputComponent\_0 @ ChatInputComponent.html:2

proxyClass @ compiler.js:19199

logError @ core.js:36330

handleError @ core.js:7244

dispatchEvent @ core.js:22523

(anonymous) @ core.js:33709

(anonymous) @ platform-browser.js:1789

invokeTask @ zone-evergreen.js:391

onInvokeTask @ core.js:30873  
invokeTask @ zone-evergreen.js:390  
runTask @ zone-evergreen.js:168  
invokeTask @ zone-evergreen.js:465  
invokeTask @ zone-evergreen.js:1603  
globalZoneAwareCallback @ zone-evergreen.js:1629  
message.service.ts:43 Message sent  
Nachricht wird gesendet

**Erwartetes Resultat:** Nachricht soll ohne Fehler gesendet werden. Console.log() Message sent

**Test Nr:** 3.1.1

**Typ:** White box

**Art:** Unit Test

**Was wird gemacht:** Man klickt im Menu auf Profil.

**Resultat:** Man wird zur Profil Seite von [yagoromero50@gmail.com](mailto:yagoromero50@gmail.com) weitergeleitet. Profilbild wird nicht angezeigt.

GET nullhttps://www.gravatar.com/avatar/cb04ff574e27bcd6f557c0ce90d29ecb?d=identicon  
net::ERR\_UNKNOWN\_URL\_SCHEME

**Erwartetes Resultat:** Man soll zur Profil Seite des Eingeloggtes Nutzers weitergeleitet werden. Profilbild soll angezeigt werden.

**Test Nr:** 4

**Typ:** White box

**Art:** Unit Test

**Was wird gemacht:** Bei der Profil Seite von [yagoromero50@gmail.com](mailto:yagoromero50@gmail.com), klickt man auf dem Profilbild, um es zu ändern.

**Resultat:** core.js:7187 ERROR FirebaseStorageError {code\_: "storage/object-not-found",  
message\_: "Firebase Storage: Object  
'users/MXARggehIEgFiVOMM8cQFmNAob33/photos/Momon.jpg' does not exist.",  
serverResponse\_

**Erwartetes Resultat:** Man soll das Profilbild anderen können und danach soll es als Profilbild bleiben.

**Test Nr:** 4.0

**Typ:** White box

**Art:** Unit Test

**Was wird gemacht:** Bei der Profil Seite von [yagoromero50@gmail.com](mailto:yagoromero50@gmail.com), klickt man auf Edit Profil.

**Resultat:** Man wird zur Edit Profil Seite weitergeleitet.

**Erwartetes Resultat:** Man soll zur Edit Profil Seite weitergeleitet werden.

**Test Nr:** 4.1

**Typ:** White box

**Art:** Unit Test

**Was wird gemacht:** Bei der Profil Seite von [yagoromero50@gmail.com](mailto:yagoromero50@gmail.com), wird folgendes eingetragen:



The image shows a profile editing interface with three separate form sections. The first section is for updating the name, with the current value 'yago' and a 'Save' button. The second section is for updating the email, with the current value 'yagoromero50@gmail.com' and a 'Save' button. The third section is for updating the username, location, age, bio, and gender, with current values 'momon', 'schweiz', '16', 'hello there', and 'male' respectively. This section has a large pink 'Save' button and a grey 'Cancel' button at the bottom.

Update your name  
yago  
Save

Update your email  
yagoromero50@gmail.com  
Save

Update your username  
momon

Update your location  
schweiz

Update your age  
16

Update your bio  
hello there

Update your gender  
male

Save

Cancel

Abbildung 19 Profile editing

Bei update your name und Update your email, wird auf dem Save Knopf geklickt. Danach wird auf Save geklickt.

**Resultat:** Console.log() Your profile has been updated!

user.service.ts:43 Your email has been updated to: yagoromero50@gmail.com

user.service.ts:47 We sent you an email verification

**Erwartetes Resultat:** Console.log() Your profile has been updated!

user.service.ts:43 Your email has been updated to: yagoromero50@gmail.com

user.service.ts:47 We sent you an email verification

Ein Verifikation Email soll gesendet werden.

## 8 Benutzerhandbuch

### 8.1 Sich registrieren

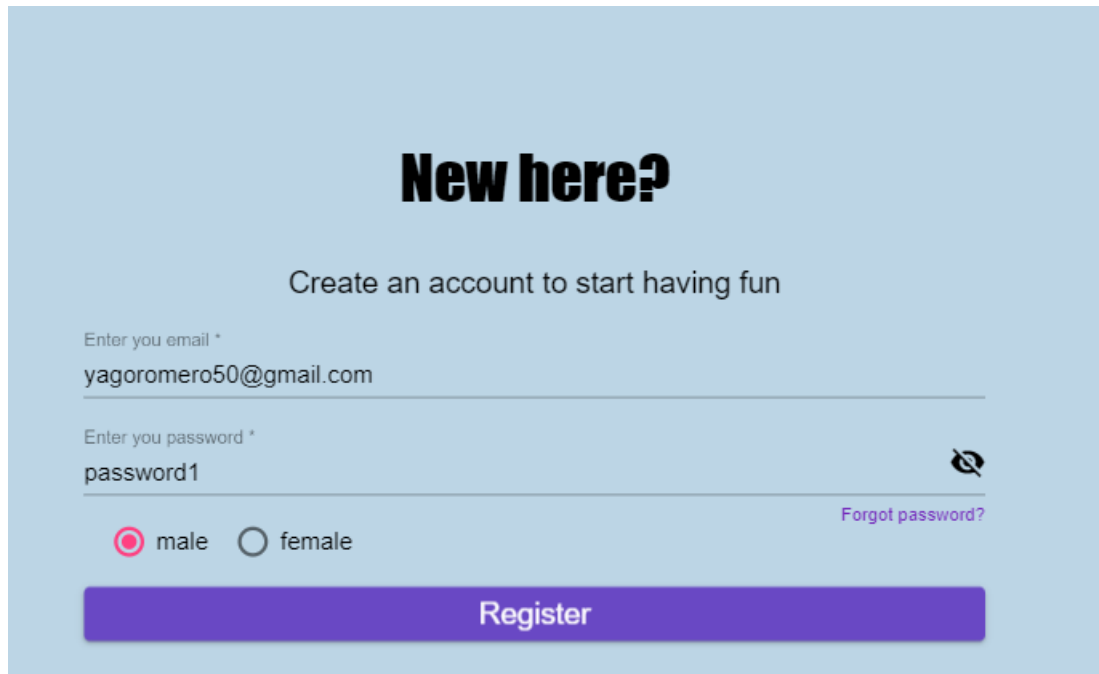
Um sich erfolgreich zu registrieren, müssen Sie ein gültiges Email eingeben wie zum Beispiel: [yagoromero@gmail.com](mailto:yagoromero@gmail.com).

Danach müssen Sie ein Password eingeben, der minimal 6 Zeilen lang ist und maximal 25 Zeilen lang ist. Ausserdem muss es mindestens eine Zahl beinhalten, um das gewählte

Password zu überprüfen können Sie auf den Augen Knopf drücken, so wird Ihrem Password angezeigt und Sie können es korrigieren.

Das Gendern ist freiwillig. Sie müssen es nicht angeben, wenn Sie nicht möchten.

Wenn Sie fertig sind, drücken Sie auf Register. Sie werden eine Verifikationsemail bekommen.



The image shows a registration form on a light blue background. At the top, the text 'New here?' is displayed in a large, bold, black font. Below this, the instruction 'Create an account to start having fun' is centered. The form includes two input fields: 'Enter your email \*' with the value 'yagoromero50@gmail.com' and 'Enter your password \*' with the value 'password1'. To the right of the password field is an eye icon for toggling visibility. Below the password field is a link that says 'Forgot password?'. Underneath the password field, there are two radio buttons for gender selection: 'male' (which is selected) and 'female'. At the bottom of the form is a large purple button labeled 'Register'.

Abbildung 20 Registrieren

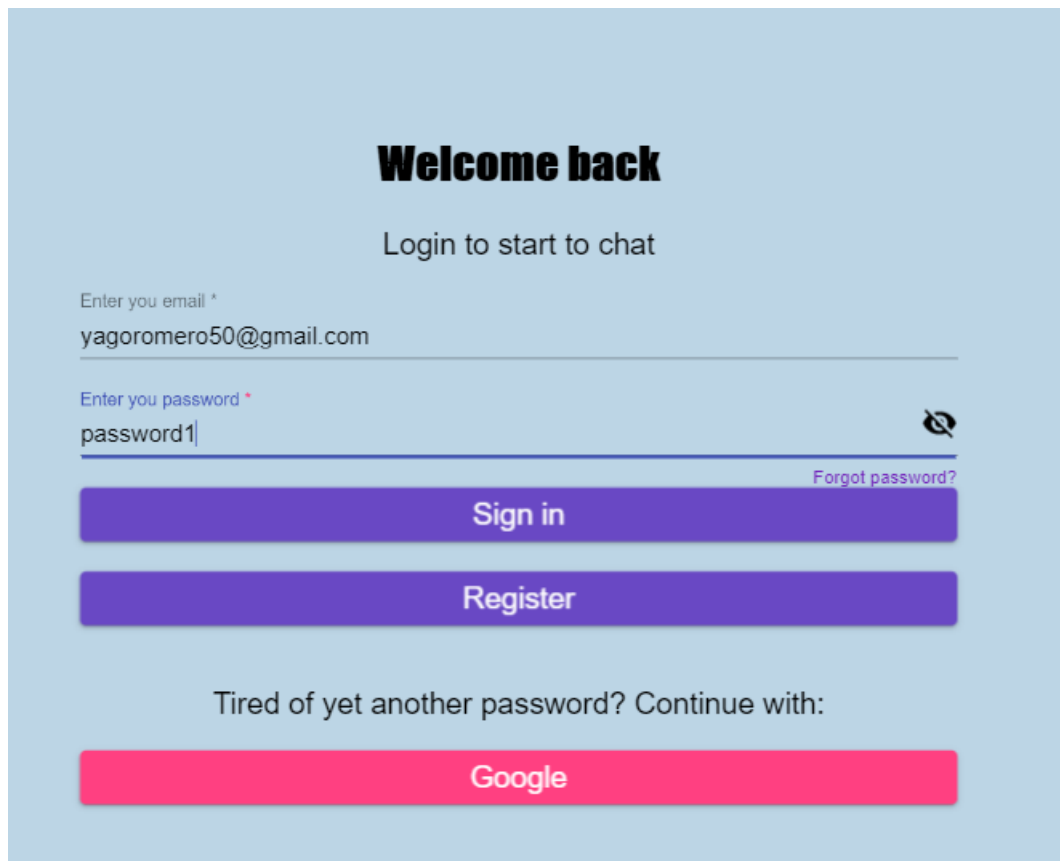
Nachdem Sie auf dem Link, der in der Verifikationsemail zu finden ist, geklickt haben, können Sie sich einloggen.

Falls Sie schon ein Account haben, können Sie oben links, neben dem Namen der App, auf Log in klicken und sich einloggen.

## 8.2 Sich einloggen

Um sich einloggen zu können, müssen Sie Ihre Anmeldungsdaten eingeben oder Sie können sich mit Ihrem Gmail Account einloggen.

Wenn Sie Email und Password eingegeben haben, drücken Sie auf Sign In. Falls Sie Ihr Password oder Email vergessen haben, können Sie auf Forgot Password? klicken, um es zurückzustellen.



The image shows a login interface for 'ChatApp'. At the top, it says 'Welcome back' in bold black text, followed by 'Login to start to chat'. Below this, there are two input fields. The first is labeled 'Enter you email \*' and contains the text 'yagoromero50@gmail.com'. The second is labeled 'Enter you password \*' and contains 'password1'. To the right of the password field is an eye icon for toggling visibility. Below the password field is a link that says 'Forgot password?'. There are two large purple buttons: 'Sign in' and 'Register'. Below these buttons, it says 'Tired of yet another password? Continue with:' followed by a large pink button with the 'Google' logo.

Abbildung 21 Einloggen

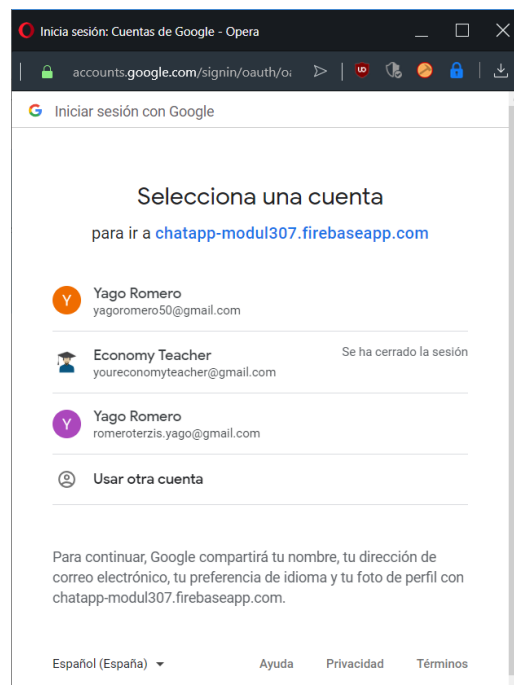


Abbildung 22 Google Login

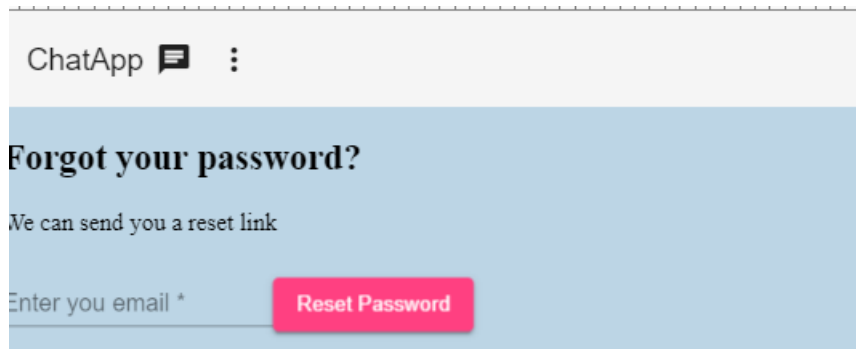


Abbildung 23 Password zurücksetzen

Nach dem einloggen, werden Sie automatisch zur Benutzern Liste weitergeleitet.

### 8.3 Chatten

Damit Sie anderen Benutzern eine Nachricht senden können, müssen sie folgende Schritte folgen:

- Erstens wählen Sie ein Nutzer aus der Benutzer Liste aus, indem sie auf den Namen dieses Nutzers klicken.

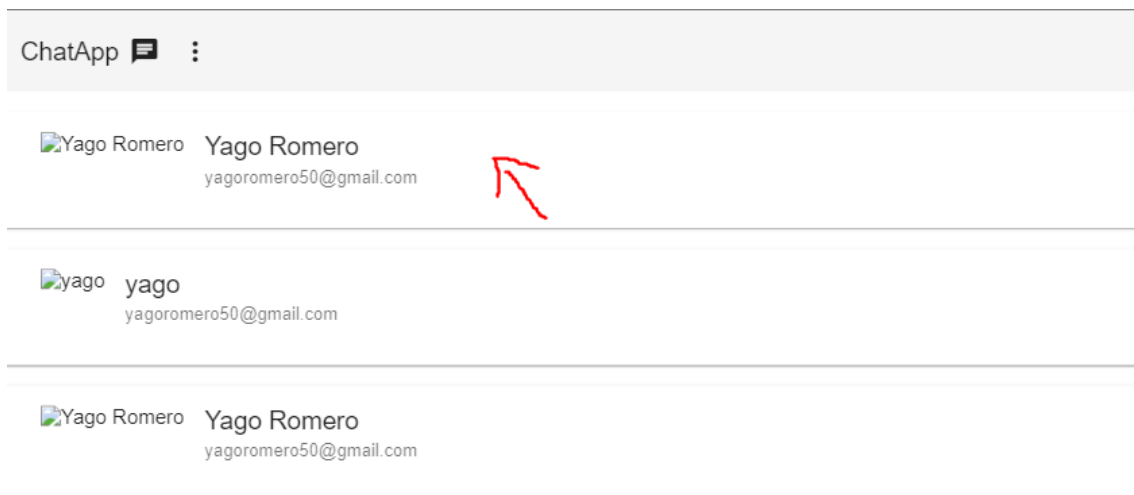


Abbildung 24 Benutzer Liste

- Sie werden zur Profil Seite dieses Nutzers weitergeleitet, dort können sie auf dem Nachricht Knopf, unten rechts beim Nutzer Profil, um einen neuen Chat anzufangen.

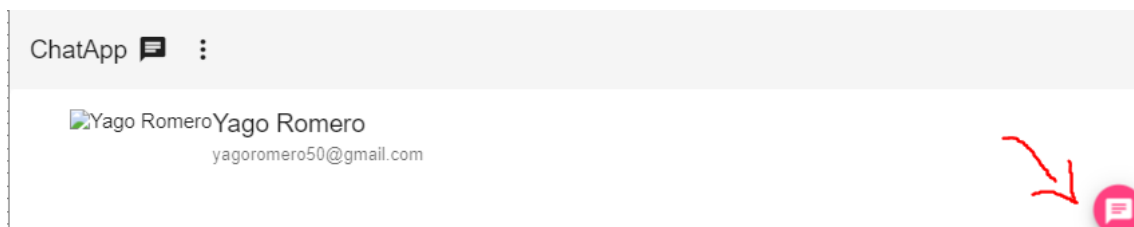


Abbildung 25 Chat Knopf

Sie werden durch den klicken auf diesen Knopf zur Chat Seite gelangen.

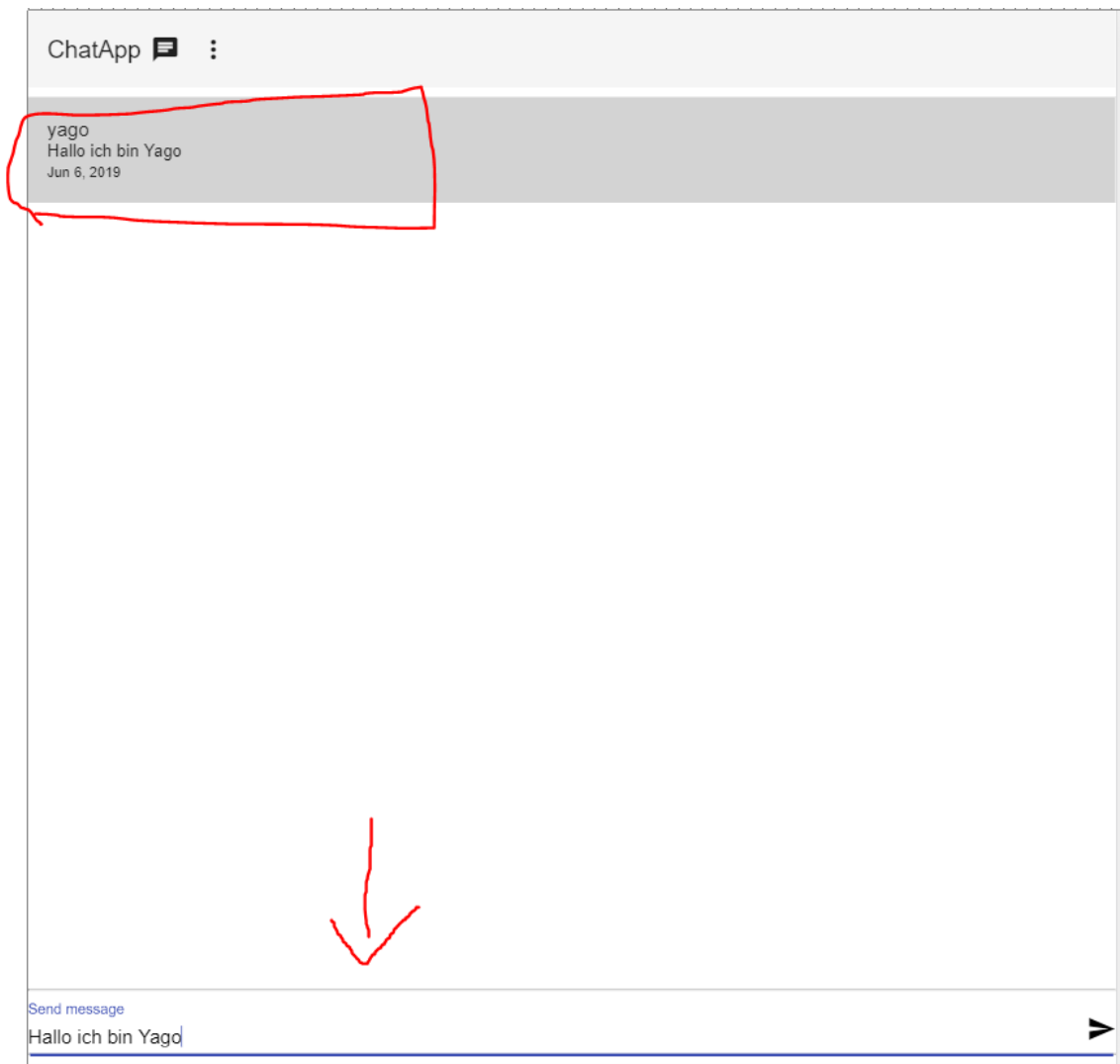


Abbildung 26 Chat

Ganz unten können sie einen Text eingeben, der versendet werden soll.

## 8.4 Navigieren

Um durch die App zu navigieren, können sie oben links neben den Namen der App, auf den Menu Knopf drücken.

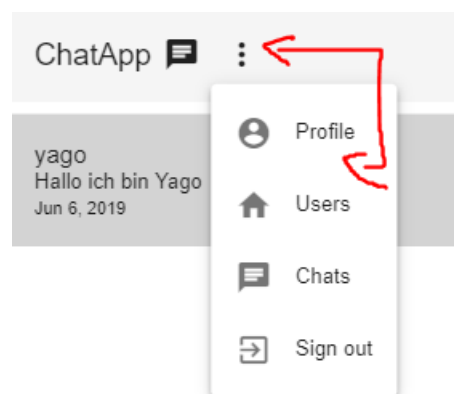


Abbildung 27 Menu

So können sie zum Profil, Benutzer List oder Chats gelangen, sowie sich auch Ausloggen.

## 8.5 Chats löschen

Falls Sie einen Chatverlauf löschen möchten, können sie auf oben links auf den Chat Icon drücken, so gelangen sie zu Ihrem aktiven Chat. Danach müssen Sie nur noch beim ausgewählten Chatverlauf, auf dem Papierkorb drücken.

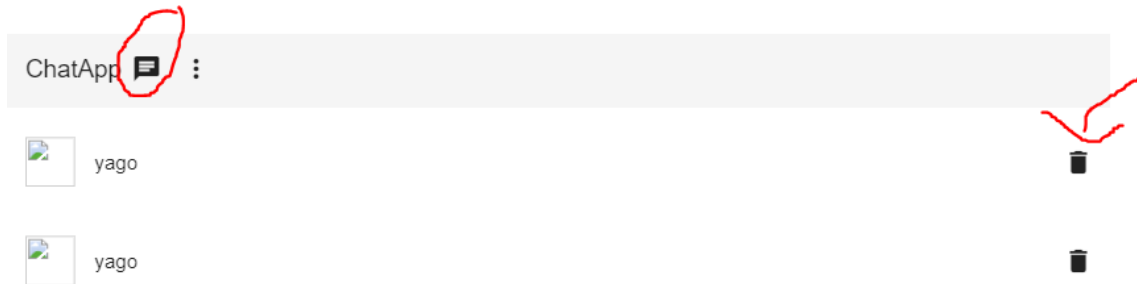


Abbildung 28 Chat löschen

## 8.6 Profil editieren

Klicken sie beim Menu auf Profil, um an ihre Profil Seite zu gelangen. Dort können sie auf Edit Profile drücken, um die Profil daten zu verändern.

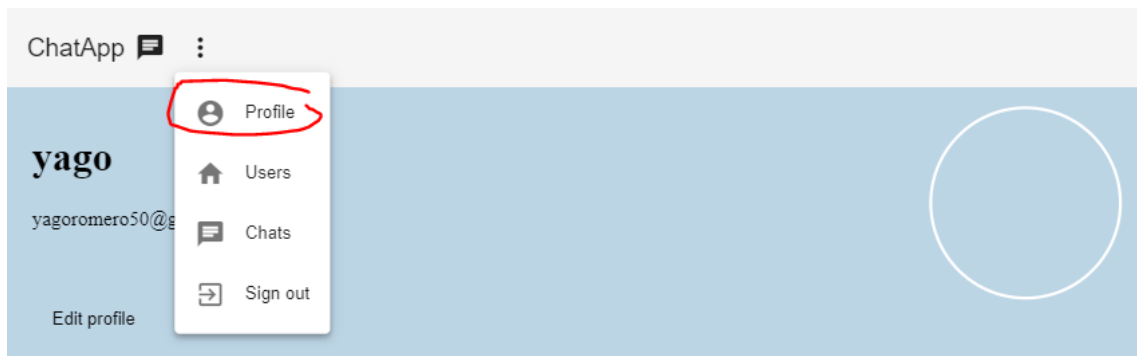


Abbildung 29 Menu Profil

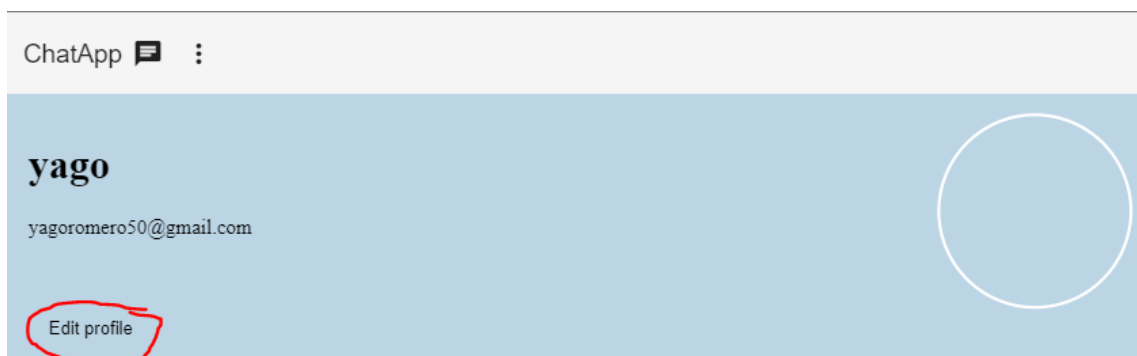


Abbildung 30 Profil

ChatApp

Update your name  
yago

Save

Update your email  
yagoromero50@gmail.com

Save

Update your username  
momon

Update your location  
schweiz

Update your age  
16

Update your bio  
hello there

Update your gender  
male

Save

Cancel

Abbildung 31 Profil editieren

Wenn Sie fertig sind, drücken Sie auf Save.

## 9 Arbeitsjournal

Was wurde gemacht?	Wenn
Thema gesucht	19/04/2019
Ich habe mich für ein Thema entschieden	26/04/2019
Umsetzung recherchieren	29/04/2019
Ich habe mich für Angular um das Projekt um zu setzen entschieden	01/05/2019
Entwicklungsumgebung wird vorbereitet	10/05/2019
Dokumentation wird geschrieben	10/05/2019
Mockups gemacht	11/05/2019
Erste Angular Komponenten und Services werden vorbereitet	14/05/2019
Authentifikationsprobleme mit Gitlab	17/05/2019
Routing der verschiedenen Komponenten	17/05/2019
Hauptfunktionen programmiert	18/05/2019

<b>Login und Registrierung</b>	19/05/2019
<b>Formular für Login und Registrierung</b>	21/05/2019
<b>Login und Registrierung Funktionen/Methoden</b>	21/05/2019
<b>Password-reset programmiert</b>	22/05/2019
<b>Users Komponenten programmiert</b>	23/05/2019
<b>Chat Komponenten programmiert</b>	26/05/2019
<b>Gesamtes Design überprüft</b>	02/06/2019
<b>Dokumentation geschrieben</b>	05/06/2019
<b>Tests durchgeführt</b>	05/06/2019
<b>Bugs korrigiert</b>	05/06/2019
<b>Dokumentation geschrieben</b>	06/06/2019
<b>Navbar und letzte Änderungen durchgeführt</b>	07/06/2019
<b>Test durchgeführt</b>	07/06/2019

## 10 Versionen Kontrolle

<i>Chatapp</i>	<i>Version</i>
	1.0.0

## 11 Anhang

<input type="text"/> Buscar por dirección de correo electrónico, número de teléfono o UID de usuario <button>Añadir usuario</button> <span>↺</span> <span>⋮</span>				
Identificador	Proveedores	Fecha de creación	Inicio de sesión	UID de usuario ↑
yagoromero50@gmail.com	✉	6 jun. 2019	6 jun. 2019	MXARggehIEgFIVOmM8cQFmNAo...
yagot@gmail.com	✉	6 jun. 2019	6 jun. 2019	aDfQahNdEJM9ed90Knz3R6Ry2J...
yago43@gmail.com	✉	5 jun. 2019	5 jun. 2019	cROkzDjVfDghMWgE6UvteciMAvR2
ahsbdh@gmail.com	✉	4 jun. 2019	4 jun. 2019	pH6kaol7xsSzhN29Z8YsC9GYeCG3
Filas por página: 50 ▾ 1-4 de 4 < >				

Abbildung 32 Authentifikation durch Firebase



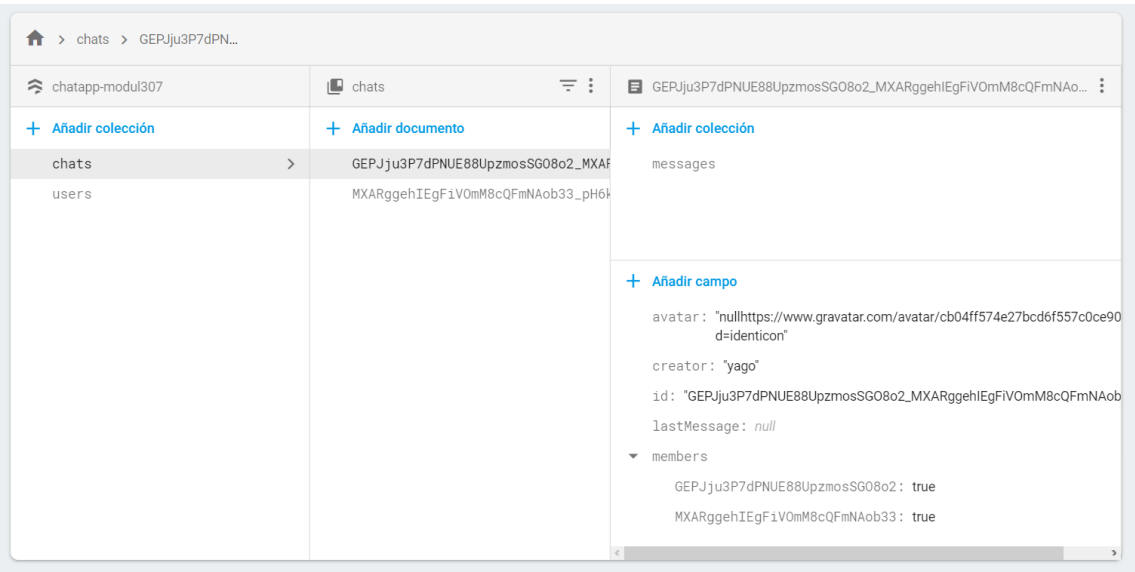


Abbildung 33 Chat Datenbank

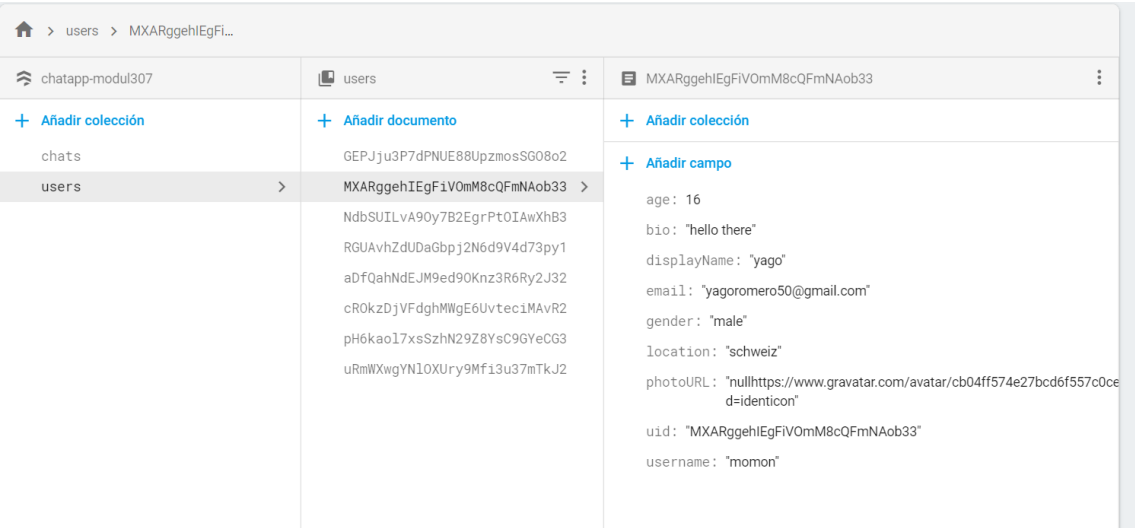


Abbildung 34 Benutzern Datenbank