**StreamTogether**

Inhalt

[**Das Team** 2](#_Toc513892511)

[Alexander Klapdor 2](#_Toc513892512)

[Arian Moser 2](#_Toc513892513)

[Christian Car 2](#_Toc513892514)

[**Die Idee** 3](#_Toc513892515)

[**Das Portal - Übersicht** 3](#_Toc513892516)

[Home 3](#_Toc513892517)

[Login 3](#_Toc513892518)

[Register 4](#_Toc513892519)

[Account 4](#_Toc513892520)

[Room overview 4](#_Toc513892521)

[Create room 4](#_Toc513892522)

[Room 4](#_Toc513892523)

[**Benutzte Technologien** 6](#_Toc513892524)

[Node.js 6](#_Toc513892525)

[React 6](#_Toc513892526)

[Semantic UI React 6](#_Toc513892527)

[NEXT.JS 6](#_Toc513892528)

[MySQL 6](#_Toc513892529)

[Nginx 6](#_Toc513892530)

[**Sicherheitsaspekte** 6](#_Toc513892531)

[JSON Web Tokens 6](#_Toc513892532)

[bcrypt 6](#_Toc513892533)

[SSL/HTTPS 6](#_Toc513892534)

[**Projektmanagement** 7](#_Toc513892535)

[**Fazit** 7](#_Toc513892536)

[**Quellen** 7](#_Toc513892537)

# **Das Team**

## Alexander Klapdor

Zuständig für: Frontend, Serverkonfiguration & Backend

## Arian Moser

Zuständig für: Backend, Fixes und Weiterentwicklung (z.B. Chat, SQL)

## Christian Car

Zuständig für: Frontend, Youtube-API, Chat

Datum der Abgabe: 16.05.2018

# **Die Idee**

Wir wohnen zusammen, teilen unsere Socken, teilen die Bandbreite unserer Internetleitung und gemeinsame schauen wir gerne den neuen Blockbuster und hören gemeinsam Musik.

Aber was wird Ende 2019 passieren, wenn unsere Studium vorbei ist?

Eventuell teilen wir unsere Socken fair auf und jeder kann endlich die volle Bandbreite seiner Internetleitung nutzen. Aber wie hören wir gemeinsam Musik und schauen gemeinsam Filme?

Als wir uns diese Frage stellten, wussten wir, dass wir eine Möglichkeit schaffen mussten, um auch nach dem Studium diese Aktivitäten gemeinsam zu erleben.

Natürlich könnten wir ein Kino eröffnen. Aber das kann jeder.

Die Lösung: Ein eigenes Portal, auf dass sich jeder einwählen kann.   
Daraus entstand unsere Idee: StreamTogether.

# **Das Portal - Übersicht**

In diesem Abschnitt wird kurz erläutert, aus welchen Komponenten das Portal besteht und wie diese untereinander verknüpft sind.

## Home

Von der Home-Seite aus kann der Benutzer folgende Aktivitäten ausführen:

* Raumliste bestehender Räume anzeigen lassen
* Wenn er nicht eingeloggt ist:
  + Sich einloggen
  + Sich registrieren
* Wenn er eingeloggt ist:
  + Seinen Account verwalten
  + Sich ausloggen
  + Einen Raum neu eröffnen

## Login

Hier hat der Benutzer die Möglichkeit sich mit einem bestehenden Account anzumelden.

Sollte er sich nicht einloggen wollen, hat er die Möglichkeit auf die Home-Seite zurück zu gelangen.

In dem Fall, dass er keinen Account besitzt, kann er sich auf die Registrierungs-Seite weiterleiten lassen.

## Register

Hier hat der Benutzer die Möglichkeit einen neuen Account anzulegen.

Sollte er keinen Account anlegen wollen, hat er die Möglichkeit auf die Home-seite zurück zu gelangen.

In dem Fall, dass er einen Account besitzt, kann er sich auf die Login-Seite weiterleiten lassen.

## Account

Hier hat der Benutzer drei unterschiedliche Möglichkeiten:

* Sich seine Account-Informationen anzeigen lassen
* Sein Passwort ändern
* Seinen Account löschen

## Room overview

Hier kann der Benutzer sich alle aktiven Räume anzeigen lassen und diesen auch beitreten.

## Create room

Hier kann der Benutzer eine Raum anlegen.

Dabei kann er einen Titel, eine Beschreibung, ein Thumbnail und optional ein Passwort festlegen.

## Room

Auf dieser Seite hat der Benutzer die Möglichkeit zusammen mit anderen Benutzern synchron Youtube-Videos zu schauen und sich dabei über den integrierten Chat zu unterhalten.

Register

Home

Login

Room

Room overview

Create room

Account

Chat

Youtube-API

# **Benutzte Technologien**

## Node.js

Node.js liefert ein Framework zur Entwicklung von serverseitigen Anwendungen. Für Node.js haben wir uns entschieden, da ebenfalls React verwendet wurde und dieses baut auf Node.js auf.

## React

React ist eine JavaScript-Softwarebibliothek für Webdesign.

## Semantic UI React

Semantic UI React ist ein Framework, dass vorgefertigte Komponenten für die Entwicklung des Frontends von Websiten liefert. Diese Technologie wurde benutzt, da das Team Einblick in eine neue Technologie zur Frontend-Entwicklung bekommen wollte und keine üblichen Frameworks wie Bootstrap & Co. benutzen wollte.

## NEXT.JS

## MySQL

## Nginx

Nginx liefert einen Webserver, der in diesem Projekt verwendet wurde.

# **Sicherheitsaspekte**

## JSON Web Tokens

Um zu verhindern, dass nicht registrierte Benutzer Anfragen an den Server stellen können, wurden JSOB Web Tokens verwendet.

## bcrypt

Um Passwörter der Benutzer nicht im Klartext zu übertragen und in der Datenbank zu speichern wurde bcrypt verwendet. Somit werden die Passwörter als Hash übertragen und gespeichert.

## SSL/HTTPS

Um zu verhindern, dass Dritte Daten mitlesen oder verändern, die übertragen werden, wurde SSL/HTTPS benutzt.

# **Projektmanagement**

Um einen ordentlichen und realistischen Zeitplan am Anfang des Projektes zu erstellen, stellte das Team einen Umsetzungs-Plan auf. Dieser Plan beinhaltet drei gestaffelte zu erreichende Ziele.

Der Quellcode wurde mittels der Versionsverwaltung „GitHub“ verwaltet.

Das Repository ist unter <https://github.com/ArianMoser/StreamTogether> erreichbar.

# **Fazit**

# **Quellen**