**StreamTogether**

Inhalt

[**Das Team** 2](#_Toc514182624)

[**Die Idee** 3](#_Toc514182625)

[**Das Portal - Übersicht** 3](#_Toc514182626)

[Home 3](#_Toc514182627)

[Login 3](#_Toc514182628)

[Register 4](#_Toc514182629)

[Account 4](#_Toc514182630)

[Room Overview 4](#_Toc514182631)

[Create Room 4](#_Toc514182632)

[Room 4](#_Toc514182633)

[Übersicht StreamTogether 5](#_Toc514182634)

[**Benutzte Technologien** 6](#_Toc514182635)

[Node.js 6](#_Toc514182636)

[React 6](#_Toc514182637)

[Semantic UI React 6](#_Toc514182638)

[NEXT.JS 6](#_Toc514182639)

[MySQL 6](#_Toc514182640)

[Nginx 6](#_Toc514182641)

[Socket.io 6](#_Toc514182642)

[**Sicherheitsaspekte** 7](#_Toc514182643)

[JSON Web Tokens 7](#_Toc514182644)

[bcrypt 7](#_Toc514182645)

[SSL/HTTPS 7](#_Toc514182646)

[**Projektmanagement** 7](#_Toc514182647)

[**Fazit** 7](#_Toc514182648)

[**Quellen** 8](#_Toc514182649)

# **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 unser 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 das sich jeder einwählen kann und in unterschiedlichen Räumen synchronisiert YouTube Video abgespielt werden können. Zudem existiert ein Chat in den Räumen, um eine bestmögliche Kommunikation zu ermöglichen.  
Und so entstand unsere Projektidee: **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 bereits 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 zu lassen
* Sein Passwort zu ändern
* Seinen Account zu löschen

## Room Overview

Hier kann der Benutzer sich alle aktiven Räume anzeigen lassen, den er beitreten kann.  
Falls ein Room ein Passwort besitzt, wird dies abgefragt.

## Create Room

Hier kann der Benutzer einen Raum anlegen.

Dabei kann er einen Titel, eine Beschreibung, ein Vorschaubild sowie 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.   
Zusätzlich sind auch die Suche nach Youtube-Videos und ein Abstimmungsverfahren über das zu nächst abzuspielende Video implementiert.

## Übersicht StreamTogether

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 ein JavaScript-Framework zur Entwicklung von Benutzeroberflächen sowohl im Browser als auch auf Mobilgeräten. Entwickelt und eingesetzt von Facebook ist es mittlerweile als Open-Source-Projekt verfügbar und hat sich bereits im Einsatz bei diversen namenhaften Websites, wie z. B. Airbnb und Netflix, bewährt.  
Wir haben uns für React aufgrund einer Empfehlung eines Kommilitonen, sowie der Herausforderung einer neuen Technologie entschieden.

## 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

Next.JS ist ein kleines Framework für serverseitig gerenderte JavaScript Webapps, welches auf React aufbaut.

## MySQL

MySQL ist eine leistungsfähige relationale Datenbank für die Abfragesprache SQL (Structured Query Language). Sie zählt zu den verbreitetsten frei erhältlichen Datenbank-Softwares (Open Source).  
MySQL bildet die Grundlage für viele dynamische Webauftritte.  
In diesem Projekt haben wir die Software „phpMyAdmin“ (freie Webanwendung zur Administration von MySQL-Datenbanken) verwendet.

## Nginx

Nginx ist ein schlanker Webserver, welcher im Rahmen dieses Projektes auf einem Ubuntu System lief.

## Socket.io

Socket.IO ist eine JavaScript-Bibliothek für Echtzeit-Webanwendungen. Es ermöglicht bidirektionale Echtzeit-Kommunikation zwischen Webclients und Servern. Es besteht aus zwei Teilen: einer clientseitigen Bibliothek, die im Browser des Benutzers läuft, und einer serverseitigen Bibliothek für Node.js.   
Im Rahmen dieses Projektes wurde die Technologie für den Chat in den jeweiligen Räumen verwendet.

# **Sicherheitsaspekte**

## JSON Web Tokens

Mit Hilfe von JSON Web Tokens, werden in diesem Projekt die Daten für die User Session Verwaltung bzw. die Daten in dem gespeicherten Cookie „verschlüsselt“. Dadurch ist gewährleistet, dass der Cookie nicht so einfach manipuliert werden kann.

## bcrypt

Um Passwörter der Benutzer nicht im Klartext zu übertragen wurde bcrypt verwendet. Dies wandelt die Passwörter in Hashes (SHA256) um, übertragt diese und speichert sie im Anschluss in die Datenbank.

## SSL/HTTPS

Um zu verhindern, dass Dritte Daten mitlesen oder verändern können, wurde SSL/HTTPS verwendet.

# **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 Meilensteine.

Der Quellcode wurde mittels der Versionsverwaltung „GitHub“ verwaltet.

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

# **Fazit**

Rückblickend kann man folgendes Fazit zu unserem Projekt ziehen:

Die Absprachen im Team verliefen durchweg sehr gut, da stets jeder wusste, wer was macht und was er selbst als nächstes machen muss. Bei der Umsetzung der Aufgaben agierten die Teammitglieder stets ehrgeizig, wodurch am Ende sogar mehr umgesetzt werden konnte, als zu Beginn geplant (Vote-Features, temporäre User, etc.). Negativ ist zu nennen, dass es am Anfang ein Missverständnis gab, wodurch einer der Teammitglieder sich im Alleingang durch externe Personen Wissen einholte, welches die anderen beiden auch hätten gebrauchen können. Des Weiteren ist zu nennen, dass am Anfang keine Coderichtlinien festgelegt wurden, wodurch die Code-Nachbereitung am Ende viel Zeit benötigt hat. Abschließend kann man sagen, dass das Projekt in allem doch sehr positiv verlaufen ist, da wir unsere Ziele erreichen konnten und viel neue Technologien kennenlernen konnten.

# **Quellen**

Webseite - <https://gruppe2.testsites.info/>

Nginx - <https://www.nginx.com/>

Next.JS - <https://nextjs.org/learn/>

React - <https://reactjs.org/>

Semantic UI - <https://react.semantic-ui.com/>

Socket.io - <https://socket.io/>

Bcrypt - <https://www.npmjs.com/package/bcrypt>

JSON Web Token - <https://jwt.io/>

MySQL - <https://www.mysql.com/de/>