Claudio Mondillo

ZÜrcher Lehrbetriebsverband ICT | CMONDILLO

Modul 426

Erkenntnisse

Inhaltsverzeichnis

[Einleitung 2](#_Toc89419124)

[Ursachen für Probleme 3](#_Toc89419125)

[Tag 1: 3](#_Toc89419126)

[Tag 2: 3](#_Toc89419127)

[Tag 3: 3](#_Toc89419128)

[Tag 4: 3](#_Toc89419129)

[Verbesserungsmassnahmen 3](#_Toc89419130)

[Tag 1: 3](#_Toc89419131)

[Tag 2: 3](#_Toc89419132)

[Tag 3: 3](#_Toc89419133)

[Tag 4: 3](#_Toc89419134)

[Gewonnene Erkenntnisse 3](#_Toc89419135)

[Tag 1: 3](#_Toc89419136)

[Tag 2: 4](#_Toc89419137)

[Tag 3: 4](#_Toc89419138)

[Tag 4: 4](#_Toc89419139)

# Einleitung

Ziel ist es eine Applikation zu erstellen, welche es ermöglicht die Follower Anzahl verschiedener Plattformen zu bekommen und diese dem User anzuzeigen. Dies ermöglicht dem Benutzer eine zentrale Stelle zu haben, welche alle Follower der verschiedenen Plattformen an einem Ort anzeigt, damit der Benutzer nicht mehrere Applikationen öffnen muss und schnell und einfach erkennt, ob seine Follower mehr oder weniger werden.

# Ursachen für Probleme

## Tag 1:

* Instagram und Facebook API sind sehr umständlich

## Tag 2:

* 7 verschiedene Programmiersprachen sind unnötig

## Tag 3:

* Daten werden im local storage gespeichert, -> besser in einem JSON abspeichern
* promise macht Probleme

## Tag 4:

* Funktion wird immer ausgeführt bevor sie soll. Deswegen ist es erforderlich, 2x den Button zu drücken, bis die Seite aktualisiert wird

## Tag 5:

* Finden von anderen API’s oder Scrapper.
* Youtube Scrapper..

# Verbesserungsmassnahmen

## Tag 1:

Instagram und Facebook API sind sehr umständlich: deshalb werden wir eine eigene API erstellen auf welcher wir alle Daten speichern und von dieser dann holen werden.

## Tag 2:

Anstatt 7 verschiedene Programmiersprachen zu verwenden haben wir uns geeinigt, eine Electron App zu erstellen. Dies erspart und den Server Teil mit dem Erstellen einer eigenen API.

## Tag 3:

JSON Datei erstellen, welche die Daten speichert, anstatt local storage.

## Tag 4:

Problem mit der Reihenfolge des Codes (Alex arbeitet an einer Lösung)

## Tag5:

Zusätzliche API’s oder Scrapper finden.

HTML anpassen

# Gewonnene Erkenntnisse

## Tag 1:

Am ersten Tag haben wir uns Gedanken gemacht, was wir genau machen wollen und mit welchen Programmiersprachen wir arbeiten.

Das Ziel ist eine Anwendung zu schreiben, welche die Follower von verschiedenen Plattformen zusammenzählt, damit der Benutzer einen schnellen Überblick über seine Follower bekommt und sofort sieht ob diese mehr oder weniger geworden sind.

Wir bekommen die Follower Anzahl über verschiedene Wege (API, HTML) und erstellen eine eigene API auf welcher wir die Follower pro Plattform speichern.

Mit Hilfe dieser API ist es uns dann möglich diese Zahlen auf unserer PHP Seite zu veranschaulichen.

**Neu gelerntes:**

-API erstellen mithilfe einer automatisch generierten db.json Datei und über Server laufen lassen

-Follower über Twitter API bekommen

-Replit kennengelernt

## Tag 2:

Den Weg, alles mithilfe einer Electron App zu visualisieren ist viel besser, da man dann nicht viele verschiedene Programmiersprachen benötigt und neben der Web App auch noch gleich eine Applikation für Android und Iphone erstellen kann. Da wir nun mit Electron arbeiten, müssen wir das meiste wieder verwerfen, können jedoch Codesegmente wiederverwenden.

Ich habe mit Dennler die Funktionen von Python auf Node umgeschrieben. Alex hat die Electron App erstellt und arbeitet gerade daran.

**Neu gelerntes:**

-Wenige Programmiersprachen zu benutzen macht alles viel übersichtlicher und spätere Änderungen sind einfacher

-Node

## Tag 3:

Wir werden eine JSON-Datei erstellen um die Daten zu speichern, da diese dann auch bei einem Storage Clean behaltet werden.

Das promise macht Probleme da nicht mit der API Abfrage gewartet wird, bis die Userdaten eingegeben wurden.

Yassin arbeitet am Frontend und ich bin ihm behilflich, wenn er nicht weiterkommt.

**Neu gelerntes:**

-Funktion von promise

## Tag 4:

Wir versuchten einen Fehler zu beheben, damit die Daten mit nur einem Klick aktualisiert werden.

Yassin arbeitet am Frontend und Alex am Backend. Ich habe in dieser Zeit keine wichtigen Aufgaben, welche ich erledigen kann, weshalb ich mich damit beschäftige für die anderen Sozialen Medien schon heraus zu finden, wie ich dort die Follower Anzahl bekomme. Ich lese mich in die Doc’s ein und bereite die Abfragen vor, welche wir später dann einbauen werden sobald die Hauptprobleme gelöst sind. Ich würde lieber aktiv an der Problemlösung helfen, jedoch ist dies schwierig da ich nicht genau weiss was Alex schon alles probiert hat. Manchmal probiere ich meine Lösungsvorschläge einzubringen aber Alex hat das meiste schon probiert. Bei Yassin kann ich gut mithelfen da CSS oder HTML Probleme meist nicht schwierig zu lösen sind.

Um 10:50 hatten wir das Problem mit Hilfe von Dennler gelöst. Zu diesem Zeitpunkt starteten wir unser Daily Meeting und verteilten neue Aufgaben.

Meine Aufgabe war es das db.json File richtig zu generieren. Dazu hatte ich Hilfe von Dennler, da es ziemlich schwierig war mit dem «Promise» zu arbeiten. Letztendlich haben wir es aber hingekriegt.

Ich habe alles gepusht und Yassin macht holt jetzt die Daten und zeigt sie im HTML an.

Damit wäre die Hauptfunktion der App abgeschlossen.

**Neu gelerntes:**

-API wissen verbessert

## Tag 5:

Am fünften Tag starteten wir mit einem kleinen Sprint-Meeting um die letzten Aufgaben noch zu verteilen. Alex hat am Sonntag noch ein paar fixes gemacht und die Applikation läuft nun ohne Probleme. Twitter und Instagram funktionieren super.

Heute wird Yassin sich noch um das Frontend kümmern, damit die Daten schön angezeigt werden.

Alex kümmert sich um die ReadMe Datei.

Ich bin daran, andere API’s oder Scrapper für noch nicht vorhandene Soziale Medien zu suchen und zu testen. Diese werden wir später noch einbauen da wir das Projekt auch nach dem Modul noch weiterführen werden.

Die Abonnenten von Youtube zu bekommen hat leider nicht gklappt, werde ich aber noch weiter probieren.