

Projektskizze

1. Einleitung / Motivation

Einleitung

Facebook, Twitter oder Instagram: Der Trend zu sozialen Netzwerken nimmt zu. Heutzutage surft so gut wie jeder mehrmals täglich durch die Social Media Kanäle. Dies zeigt eine aktuelle Social Trends Studie des Digitalvermarkters Tomorrow Focus Media.

Die Studie hat ergeben, dass Frauen mit 55,3 Prozent die sozialen Netzwerke intensiver nutzen als Männer (46,7 Prozent). Trotz der intensiven Nutzung nimmt das Einstellen von eigenen Inhalten beständig ab. Bereits 70 Prozent der Nutzer surfen nur noch passiv (nur Anschauen von Inhalten) durch die Online Communities.

In Bezug auf Statusmeldungen empfinden 77,8 Prozent der Befragten Posts, die Bilder enthalten, als besonders ansprechend.

Mehr als die Hälfte der Befragten hat angegeben, sich über soziale Netzwerke über Unternehmen zu informieren. 43,6 Prozent gaben an, dass diese ihre Lieblingsmarke aktiv unterstützen.

Unternehmen können das Interesse an ihrem Profil in den sozialen Netzwerken vor allem in Form von Werbung nutzen. Aufgrund von Werbung auf sozialen Netzwerken haben 30,2 Prozent der Befragten schon einmal Einkauf getätigt.

Facebook	Instagram	LinkedIn
Nutzerzahlen / Montag: 1,7 Milliarden (Juli 2016)	Nutzerzahlen: 500 Millionen (Juli 2016)	Nutzerzahlen (international): 433 Millionen (Juli 2016)
Nutzer / Tag: 1,13 Milliarden (Juli 2016)	Nutzerzahlen (Deutschland): 9 Millionen (Januar 2016)	Nutzerzahlen (Europa): 104 Millionen (Juli 2016)
Likes / Tag: 1,6 Milliarden (März 2014)	Nutzer / Tag: 300 Millionen (Juli 2016)	Nutzerzahlen (DACH): 8 Millionen (Juli 2016)
Bilder / Tag: 60 Millionen (März 2014)	Nutzer außerhalb der USA: 80% (Juli 2016)	Unternehmensprofile: 3 Millionen (März 2014)
Facebook-Messenger / Montag: 1 Milliarde (Juli 2016)	Geteilte Bilder: 40 Milliarden (Januar 2016)	Social-Sharing: 1,5 Millionen Websites nutzen den LinkedIn-Share-Button (März 2014)
	Likes / Tag: 4,2 Milliarden (Juli 2016)	Anzahl Gruppen: 2,1 Millionen (März 2014)
	Bilder / Tag: 95 Millionen (Juli 2016)	

Zahlen / Statistiken entnommen:

<http://socialmedia-institute.com/uebersicht-aktueller-social-media-nutzerzahlen/> (23.03.2017)

Die wirtschaftliche Gewichtung der sozialen Netzwerke ist enorm. Viele Unternehmen definieren ihre Unternehmensziele ausschließlich über soziale Netzwerke. Anderen Unternehmen sind soziale Netzwerke mindestens so wichtig, wie das Suchmaschinen Marketing (Werbung in Suchmaschinen, Top Positionen bei den Suchergebnissen, ...).

Kurz gesagt: Um den Erfolg einer Marketing-Strategie zu garantieren, sind die Unternehmen auf Nutzung von Plattformen wie Facebook, Instagram, LinkedIn usw. angewiesen.

Für die Einzelperson ist es unmöglich, ohne die Erfahrungen / Bewertungen anderer Personen, das beste Angebot / das beste Produkt zu finden. Oft sind gute Bewertungen / viele Likes ausschlaggebend. Auch Informationen / Tipps von Bekannten / Freunden sind oft maßgeblich im Entscheidungsprozess.

Ein einfaches Beispiel hierfür sind Online-Shops (eBay, Amazon, ...). Beim Vergleich von Produkten beziehen Nutzer nicht nur den Preis in Betracht. Die Bewertungen von anderen Nutzern / Käufern haben meist einen großen Einfluss auf die Entscheidung des Käufers. Dabei werden häufig auch etwas teurere Produkte bevorzugt, wenn diese die besseren Bewertungen / mehr Likes / Empfehlungen haben.

Das selbe Prinzip funktioniert auf sozialen Netzwerken. Personen, die auf Facebook, Instagram, usw. mehr Follower / Abonnenten / Likes haben tauchen häufiger in Suchergebnissen auf und werden eher von den Nutzern gefunden.

Wer zum Beispiel schon einmal ein eigenes Projekt auf sozialen Netzwerken veröffentlicht hat, weiß selbst nur zu gut: *je mehr Likes das Projekt hat, umso größer die Besucherzahl und auch die Bekanntheit des Projektes*. Einmal angestoßen, steigt das Wachstum an Likes / Abonnenten / etc. exponentiell, da durch mehr Besucher immer mehr Likes / Abonnenten hinzukommen.

Ein weiterer Aspekt der Bekanntheit (und damit auch großen Zahl Likes / Abonnenten / etc.) ist, dass den Besuchern unterbewusstes Vertrauen suggeriert wird.

Zahlreiche Studien haben diese ausgestellten Theorien bewiesen. Je nach Umfeld, Angebot, Seriosität, und weiteren Faktoren bedeuten viele Likes / Follower / Abonnenten bis zu 50 Prozent mehr Erfolg.

Motivation

Unsere Motivation für das Projekt, einen eigenen Like-Bot / Follower-Bot umzusetzen beruht auf folgenden Fakten:

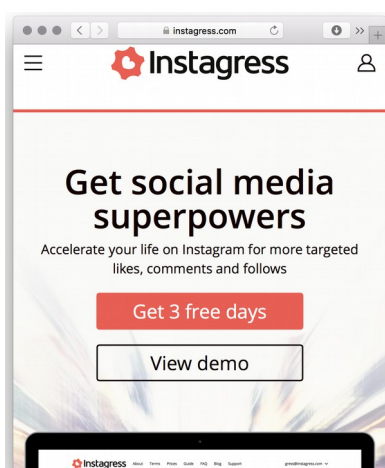
Im Internet zu findende Anbieter für Like-Bots / Follower-Bots sind abhängig von der jeweiligen Website. Es gibt zahlreiche Bots für solche Webseiten wie Facebook, Instagram, usw. allerdings gibt es keinen Bot, der beliebige soziale Netzwerke gleichzeitig unterstützt. Die meisten Anbieter konzentrieren sich nur auf eine konkrete Ziel.

Das System, dass wir entwickeln möchten, ermöglicht dem Nutzer einen eigenen Bot aufzubauen. Somit erschaffen wir ein System, das unabhängig von der jeweilige Webseite die Aufgaben eines Bots erfüllt.

Ein solcher Bot soll es dem Nutzer ermöglichen, eigene Inhalten, wie zum Beispiel Kommentare, Nachrichten, etc. zu den Posts hinzuzufügen. Dies geht meist über den Umfang der bereits existierenden Bots hinaus, da diese oft nur Posts liken oder Nutzern folgen / Nutzer abonnieren.

Auch die Tatsache, dass die Software die Arbeit für den Nutzer übernimmt ist ein weiterer Aspekt für die Motivation. Eine große Zahl an Likes / Followern / Abonnenten in Eigenarbeit zu erreichen, benötigt viel Zeit und ist oft nicht einfach für den Nutzer. Die Software übernimmt die Arbeit (durch Generierung von Likes / Followern / Abonnenten) und erleichtert dem Nutzer das Wachstum auf den sozialen Netzwerken.

Um die Funktionalitäten und Aufgaben zu verstehen haben wir bereits existierende Bots untersucht und ausprobiert. Im speziellen handelt es sich dabei um Instagress, Twitfox und Cyberlikes.



Im Allgemeinen unterscheidet sich die Funktionsweise der Tools kaum. Es wird eine Verbindung zu der jeweiligen API aufgebaut. Anschließend kann geliked, gefollowed und abonniert werden. Darüber hinaus folgen / abonnieren / liken die Tools automatisch auch andere(n) Nutzern. Dadurch sollen andere Nutzer auf den eigenen Account aufmerksam gemacht werden und gegebenenfalls zurück folgen / abonnieren / liken. Hier kommt der psychologische Trick „ **Folge ich dir, folgst du mir**“ zum Einsatz.

2. Grundbegriffe

API – engl. für application programming interface – Programmierschnittstelle; bietet Zugriff auf Funktionen / Attribute / Eigenschaften von Websites / Programmen

Bot – Computerprogramm, dass (weitgehend) automatisch wiederholende Aufgaben automatisch abarbeitet

Clickjacking – Technik, bei der ein Computerhacking die Darstellung einer Internetseite überlagert und dann deren Nutzer dazu veranlasst, scheinbar harmlose Mausklicks und / oder Tastatureingaben zu tätigen

DOM – engl. für document object model – Schnittstelle zwischen HTML und dynamischem JavaScript; ermöglicht Aufruf, Zugriff und Veränderung von Objekten auf Websites

GUI – engl. für graphical user interface – Benutzeroberfläche eines Programms / Systems / usw.

X-Frame-Options – Gegenmaßnahme zu Clickjacking; Möglichkeit, dass Webanwendungen einen Header „X-Frame-Options“ senden, der dem Browser anzeigt, ob die Webseite in einem Frame angezeigt werden darf oder nicht; Mögliche Werte sind „DENY“, Seite darf nicht in Frame angezeigt werden; „SAMEORIGIN“, Seite darf nur von Frames auf derselben Domain angezeigt werden und „ALLOW-FROM“, Seite darf von festgelegter / (n) Domäne(n) / URL(s) angezeigt werden. (Unterstützt in Internet Explorer, Firefox, Opera, Safari und Chrome)

3. Problem

Im Internet sind verschiedenste Like-Bots / Follower-Bots usw. zu finden. Wir haben jedoch keinen finden können, der die von uns gestellten Anforderungen zu 100 Prozent erfüllt.

Es gibt Bots, die unterschiedliche Funktionalitäten liefern. Diese sind jedoch oft an ein konkretes soziales Netzwerk gebunden und können nicht bei verschiedenen sozialen Netzwerken zum Einsatz kommen. Dies ist der Abhängigkeit zu der jeweiligen API des sozialen Netzwerks zu verschulden.

Ein Bot, der unabhängig vom sozialen Netzwerk Posts liken, Nutzern folgen oder diese abonnieren soll, darf nicht auf API Funktionen angewiesen sein.

Im folgenden Kapitel „Tool & Web-Anwendung“ wird ausführlich beschrieben, welche Probleme auftreten können, wenn man ein solches System in Form einer Web-Anwendung realisiert.

Der Funktionsumfang vieler Bots ist unserer Ansicht nach nicht ausreichend. Wir möchten dem Nutzer die maximale Freiheit bei zum Beispiel der Gestaltung der Kommentare ermöglichen.

Auch die Sicherheit des Nutzeraccounts ist bei (fast) allen Bots nicht garantiert, da die meisten Bots serverseitige Anwendungen sind. Hier ist der Nutzer gezwungen, seine Accountdaten (Nutzername, Passwort) anzugeben und muss darauf vertrauen, dass diese gut geschützt sind und kein Zugriff Dritter darauf erfolgen kann.

Plattformunabhängigkeit (im Bezug auf verschiedene Anwendersysteme) ist bei vielen Bots auch nicht garantiert. Somit kann beispielsweise ein Bot, der auf einem Mac funktioniert, auf einem Windows Rechner nicht eingesetzt werden.

Tool & Web-Anwendung

Eine Webseite, die ein solches System repräsentiert, soll über Zahlreiche Bedienelemente verfügen. Es handelt sich also um eine Grafische Bedienoberfläche die das Erstellen eines Bots ermöglicht. Die jeweilige Webseite, die von dem erstellten Bot bedient wird, muss ebenso als ein html-Element eingebettet werden (z.B mit <iframe>).

Mittels X-Frame-Options in html-header wird die oben beschriebene Möglichkeit verhindert. X-Frame-Options ist eine Gegenmaßnahme gegen Clickjacking und wird von vielen Sozialen Netzwerken wie Facebook, Instagram, Twitter usw. ergriffen.

Wir entscheiden uns deswegen das vorgestellte System in Form eines Tool zu realisieren. In folgenden Kapiteln werden die Anforderungsanalyse und die Lösungsvorschläge eines Programms besprochen.

4. Anforderungsanalyse / Priorisierung

Das Tool soll den Standard Browser des Betriebssystems integrieren. Die GUI des Tools zeigt sowohl die Funktionen des Tools als auch den Browser in dem der Nutzer navigieren kann an.

Das Tool muss als Tool umgesetzt werden, das nicht vom gewünschten sozialen Netzwerk abhängig ist.

Das Tool muss mehrere Eigenschaften anderer Bots kombinieren. Es muss im Tool möglich sein, Posts zu liken, Nutzern zu folgen oder diese zu abonnieren, etc.

Die Programmiersprache, in der das Tool umgesetzt wird, soll Plattform (Betriebssystem) unabhängig sein.

Im Tool soll es dem Nutzer möglich sein, die Kommentare, die das Tool verwendet, frei zu gestalten.

Das Tool soll die Kommentare entsprechend der Sprache des Original Posts verwenden. Der Nutzer gibt Kommentare in zum Beispiel drei Sprachen an. Der Bot entscheidet dann anhand der bereits vorhandenen Kommentare oder des Titel des Posts oder anderen Parametern, in welcher Sprache der Kommentar eingefügt werden soll.

Die GUI des Tools soll möglichst einfach gehalten sein. Der Nutzer kann bei erster Verwendung alle Funktionen des Tools ohne Anleitung verwenden.

Der Bot kann dem Nutzer die Funktion bieten den Aufbau der Website zu untersuchen.

5. Lösungsvorschläge

Zur Integration des Browsers in das Tool kann eine Bibliothek verwendet werden, die diese Funktionalität bietet. Vorschläge wären zum Beispiel JavaFX für Java, WebKit für Swift oder (Qt-)WebKit in Python. Die GUI wird in der entsprechenden Sprache umgesetzt.

Um die gewünschte Unabhängigkeit von den sozialen Netzwerken zu garantieren verzichtet der Bot auf Zugriffe auf die APIs der sozialen Netzwerke. Stattdessen werden Strukturen und Aufbau und damit auch Kommentarfelder, Like-Buttons oder ähnliches durch Analyse der Website erkannt. Ein Beispiel hierfür wäre die Verwendung des DOM-Baums der Website.

Durch die Verwendung des obigen Vorschlages, wird es ermöglicht, verschiedene Funktionen in einem Tool zu kombinieren. Wie bereits erwähnt, ist es möglich Like-Buttons zu finden und damit einen Like-Bot zu simulieren aber auch Kommentarfelder zu finden und auszufüllen und damit einen Kommetar-Bot zu simulieren.

Zur Gestaltung eigener Kommentare, die das Tool verwendet, kann der Nutzer Kommentare direkt im Programm eingeben. Die Kommentare werden dann nach Sprache sortiert in verschiedenen Dateien gespeichert.

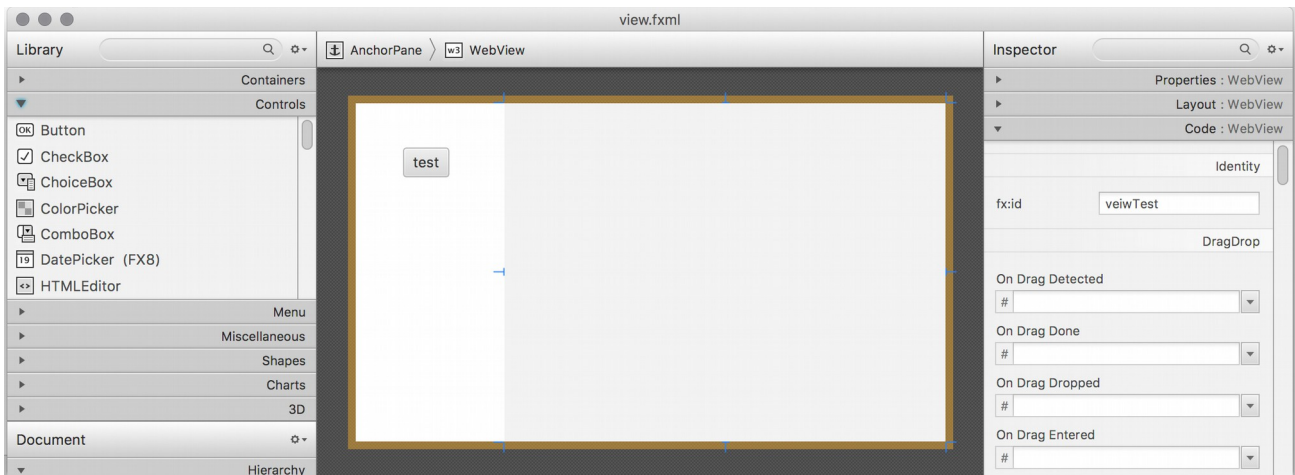
Das Tool analysiert die bisher vorhandenen Kommentare eines Posts. Wenn keine Kommentare vorhanden sind, können auch andere Posts oder der Titel des Posts in die Analyse einbezogen werden. Nach einer erfolgreichen Analyse weiß das Tool, in welcher Sprache die anderen Kommentare verfasst sind. Nun kann es aus den vorhandenen Kommentar-Dateien der passenden Sprachen einen Kommentar auswählen und diesen Posten.

Um die GUI möglichst einfach zu halten, sollten alle Buttons die in der GUI vorhanden sind eindeutige Bezeichnungen haben. Es wäre möglich, die Buttons mit Tooltips zu verknüpfen um dem Nutzer eine kurze Beschreibung zur Verfügung zu stellen.

Um die Struktur / den Aufbau der Website zu untersuchen kann im Tool ein interaktives Schaubild des DOM-Baumes der Website angezeigt werden. Hier kann der Nutzer verschiedene Elemente anklicken oder sich anzeigen lassen und deren Attribute auslesen.

Programmiersprache

Als Programmiersprache für das Tool wäre Java eine mögliche Option. Java ist eine der am weitesten verbreiteten Programmiersprachen und ist auf den verschiedensten Betriebssystemen lauffähig. Java bietet eine umfangreiche Klassenbibliothek und ist in der Verwendung und Gestaltung für sowohl Nutzer als auch Programmierer einfach. Für Programmierer steht es ein komfortabler UI-Designer zur Verfügung, namens „Scene Builder“. Dieser kann direkt in die Entwicklungsumgebung, zum Beispiel Eclipse, eingebunden werden.



Eine weitere Option ist Python. Python ist ebenfalls sehr weit verbreitet und auch auf verschiedensten Betriebssystemen lauffähig. Python Programme sind meist kürzer und weniger umfangreich, was sich positiv auf die Laufzeiten des Tools auswirken kann.

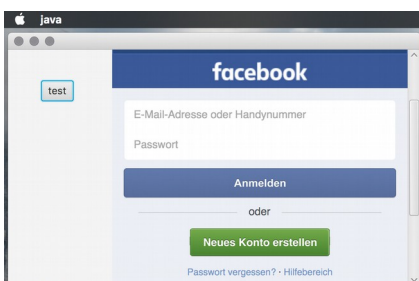
Auch eine Umsetzung in C / C++ wäre eine Option für das Tool. Allerdings bieten C / C++ weniger Komfort für den Programmierer / Nutzer. Die Gestaltung einer GUI ist in C sehr schwer. Außerdem ist C keine Objektorientierte Programmiersprache und damit nicht sehr geeignet für die Umsetzung des Tools.

Das Tool kann ebenso in Swift umgesetzt werden. Swift ist eine neu objektorientierte Programmiersprache von Apple. Die integrierte Entwicklungsumgebung Xcode bietet sehr einfache und komfortable Werkzeuge um GUI eines Tools zu gestalten. Das Programm wird in dem Fall nur auf die Mac Geräte ausführbar und somit Plattform abhängig.

Die folgende Abbildung repräsentiert die Integration eines Browsers in der entsprechenden Sprache

IOS → Java

Framework : JavaFX



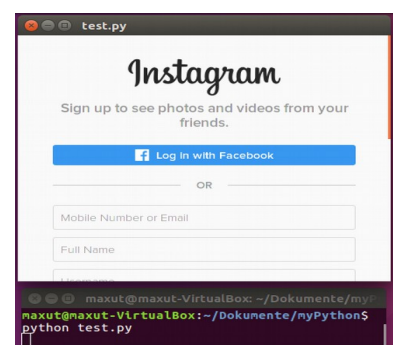
IOS → Swift

Framework: WebKit



Linux (Ubuntu) → Python

Framework: QtWebKit



6. Auswahl der Lösungen anhand der Anforderungen

Um die Integration des Browsers in das Tool zu gewähren verwenden wir JavaFX. JavaFX stellt ein Framework zur Erstellung plattformübergreifender Java-Applikationen dar. Es ermöglicht die Einbettung von Web-Inhalten in eine JavaFX-Applikation. Dazu gehört unter anderem HTML-Rendering (Visualisierung von HTML Inhalten) und die Option des Zugriffs auf den DOM-Baum der Website. Auch eine Manipulation des DOM-Baums wird durch JavaFX machbar.

Als Programmiersprache verwenden wir somit Java. Java ist im Vergleich zu C / C++ einfacher in der Verwendung für den Programmierer. Die Laufzeiten des Programms unterscheiden sich dabei nicht großartig, da das Tool keine komplexen Aufgaben erfüllen muss. Python fällt als Lösung weg, da JavaFX im Vergleich zu der äquivalenten Bibliothek von Python einen größeren Funktionsumfang liefert und in der Bedienung einfacher und komfortabler ist. Außerdem benötigt Python einen Compiler der das Programm für ein bestimmtes Betriebssystem ausführbar macht. Da auf fast allen Systemen die Java Virtual Machine laufen kann, ist Java die bessere Option.

Die Kommentare, die der Nutzer eingeben kann, werden über ein Textfeld im Tool eingegeben. Dabei gibt der Nutzer in einem separaten Feld die Sprache des Kommentars an. Nach Bestätigung des Nutzers wird der Kommentar, je nach Sprache, in einer anderen XML-Datei gespeichert.

Zur Analyse der bisher vorhandenen Kommentare verwendet das Tool DOM-Scripting. DOM-Scripting ermöglicht es dem Tool bereits vorhandene Kommentare auszulesen. Dann werden die ausgelesenen Kommentare mit einem vordefinierten Wortschatz der vorhandenen Sprachen verglichen und anhand der Anzahl an Überschneidungen von Wörtern entschieden um welche Sprache es sich handelt. Sollte es keine eindeutige Lösung geben, wird der Nutzer gebeten, die Sprache des Kommentars selbst auszuwählen.

Für die Gestaltung der GUI verwenden wir den JavaFX Scene Builder. Der JavaFX Scene Builder bietet eine GUI zur Gestaltung eigener GUIs. Diese können einfach per Drag&Drop zusammengebastelt werden. Der vom Scene Builder erzeugte Code kann anschließend bearbeitet und nach Wunsch angepasst werden um verschiedene weitere Funktionen zu realisieren.

Auch zur Darstellung des interaktiven Schaubilds kann der JavaFX Scene Builder verwendet werden.