

Social Media Analytics Application

Title	Author	Date
Initial version	Evangelos Kyritsis	02/2020

Scope of this document

The scope of this document, is to present the “*Social Media Analytics Application*” and explain its main functionalities.

Overview

The “*Social Media Analytics Application*” is an application designed for extracting and analysing data of the same user from its social media accounts.

More specifically, the application requires a .csv file as an input containing usernames of user's accounts for the following social media:

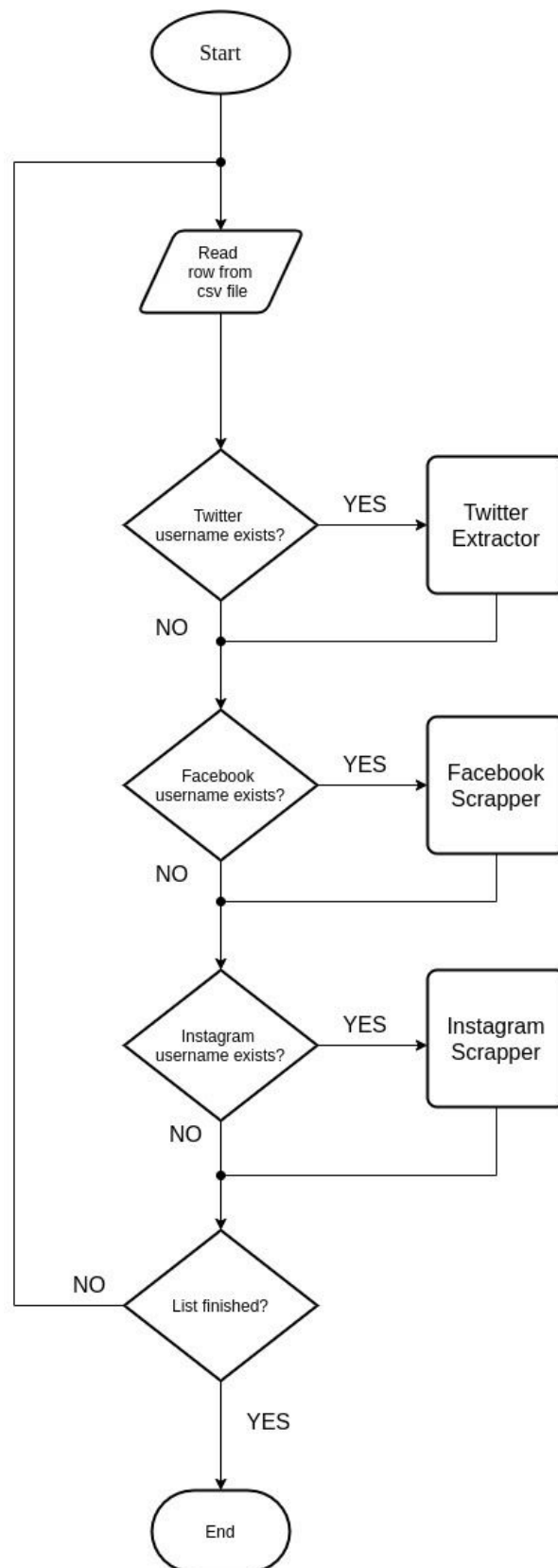
- Twitter
- Facebook
- Instagram

The application utilizes various tools for extracting the raw data from user's social media and stores them in a local database.

The application is written in Python and uses the framework django. In the front end it uses Html, CSS, Javascript and the libraries JQuery and Plotly.js.

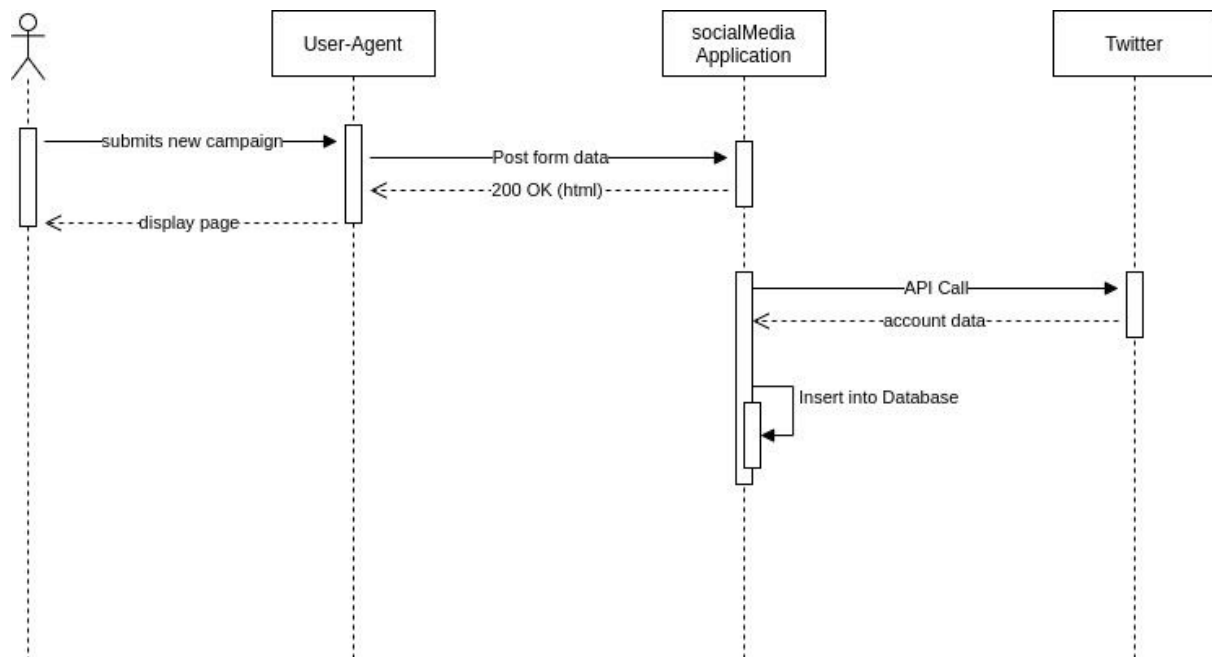
For the backend it uses the MySQL Database, the library python-twitter for the communication with the Twitter API, BeautifulSoup for processing html data, and the library selenium with Firefox web browser to scroll down in Facebook and Instagram.

Flow Chart



Twitter Extractor

The twitter extractor is the process used for retrieving user's data from his twitter account. The process uses the python-twitter library for calling the Twitter API. The following sequence diagram depicts the basic function of the process:



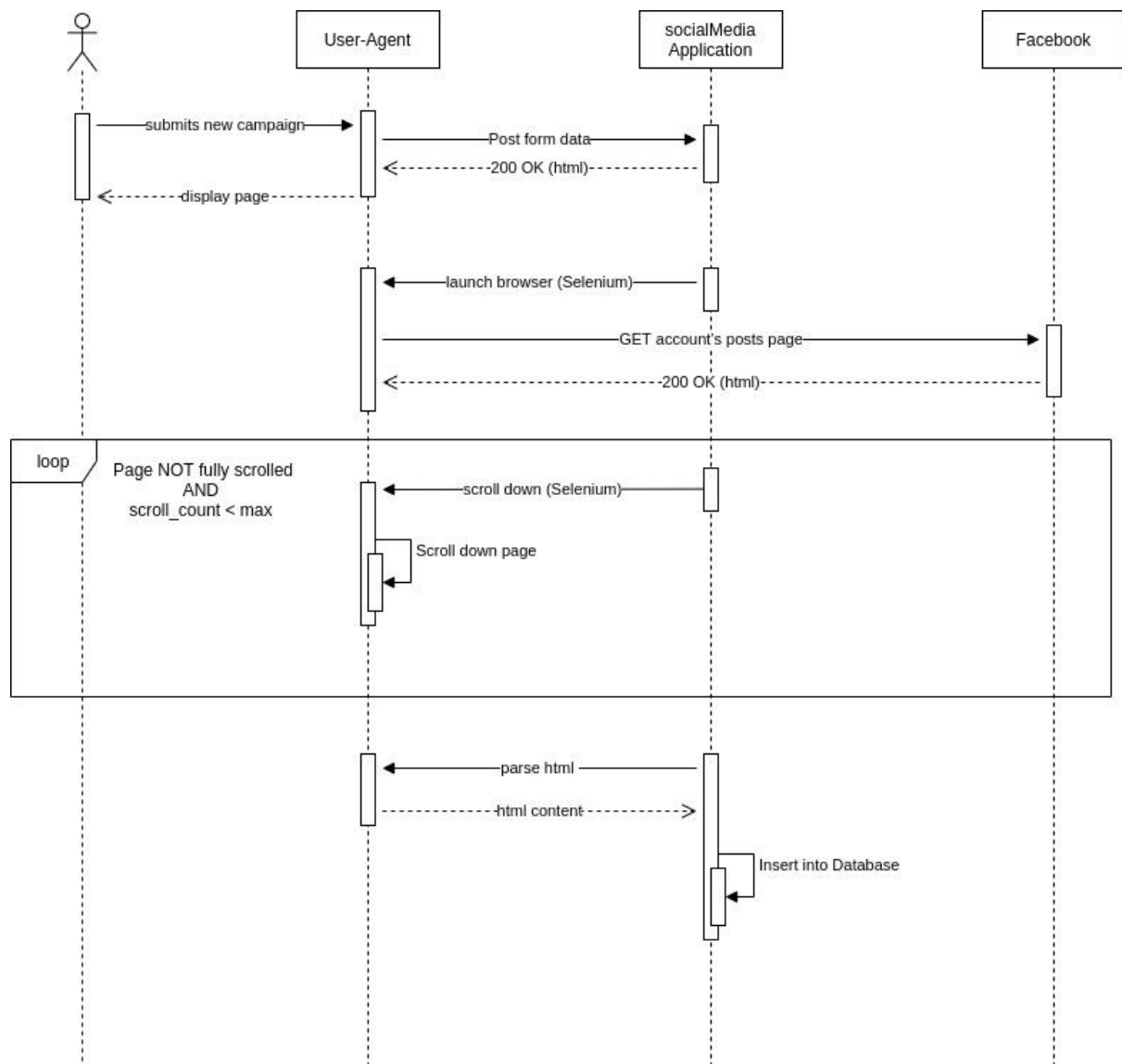
Facebook Scraper

Facebook scraper is the process used for extracting user's data from his Facebook account. The process launched a Mozilla web-browser with the help of Selenium and visits user's posts page on Facebook.

The page is scrolled down as long there are still posts available or the defined maximum scroll count has been reached.

Then, the html source code is parsed by using BeautifulSoup.

Finally, the information for every post is stored in the local MySQL database. The following sequence diagram depicts the basic function of the process:



Instagram Scraper

The functionality of this process is similar to Facebook Scraper presented in the previous paragraph. Again, with the help of Selenium, the home page of the user is visited.

After the page is scrolled down, the URLs of the posts found on the home page are stored in the database.

Then, a loop runs where Selenium visits each post URL one-by-one and the html source code of each post page is parsed with BeautifulSoup. The post data are saved in the database for every iteration of the loop.

