PROJECT REPORT

TweetMine – Twitter Sentiment Analysis Tool

KRZYSZTOF OBLAK

C00161361

Table of Contents

1. Introduction

1.1. Purpose and Content

The purpose of this document is to outline any and all work that has been completed to date on the TweetMine application. The document will also cover in detail any work that is currently outside the scope, still outstanding on the project.

Also contained within this document are descriptions of all problems that were encountered while creating the TweetMine application along with the solutions found.

Finally any learning outcomes from the implementation will be outlined.

1.2. Project Brief

TweetMine application is a web based application that will allow its users to perform a keyword search to obtain information outlining opinions contained within messages (tweets) posted on Twitter. Those messages will be then categorised based on their nature (positive, negative, neutral) as a result of sentiment analysis and natural language processing techniques. With the use of SQLite3 database the application will be able to store keywords for the purpose of a search cache.

2. Description of Submitted Project

2.1. TweetMine Application

TweetMine application is a web application optimised for desktop browsing with the support for mobile browsing on devices such as smartphones and tablets. The main purpose of the application is to provide a sentiment analysis for Twitter messages known as tweets. The application would be able to support other social media such as Facebook or Google+ but it would require codebase alterations.

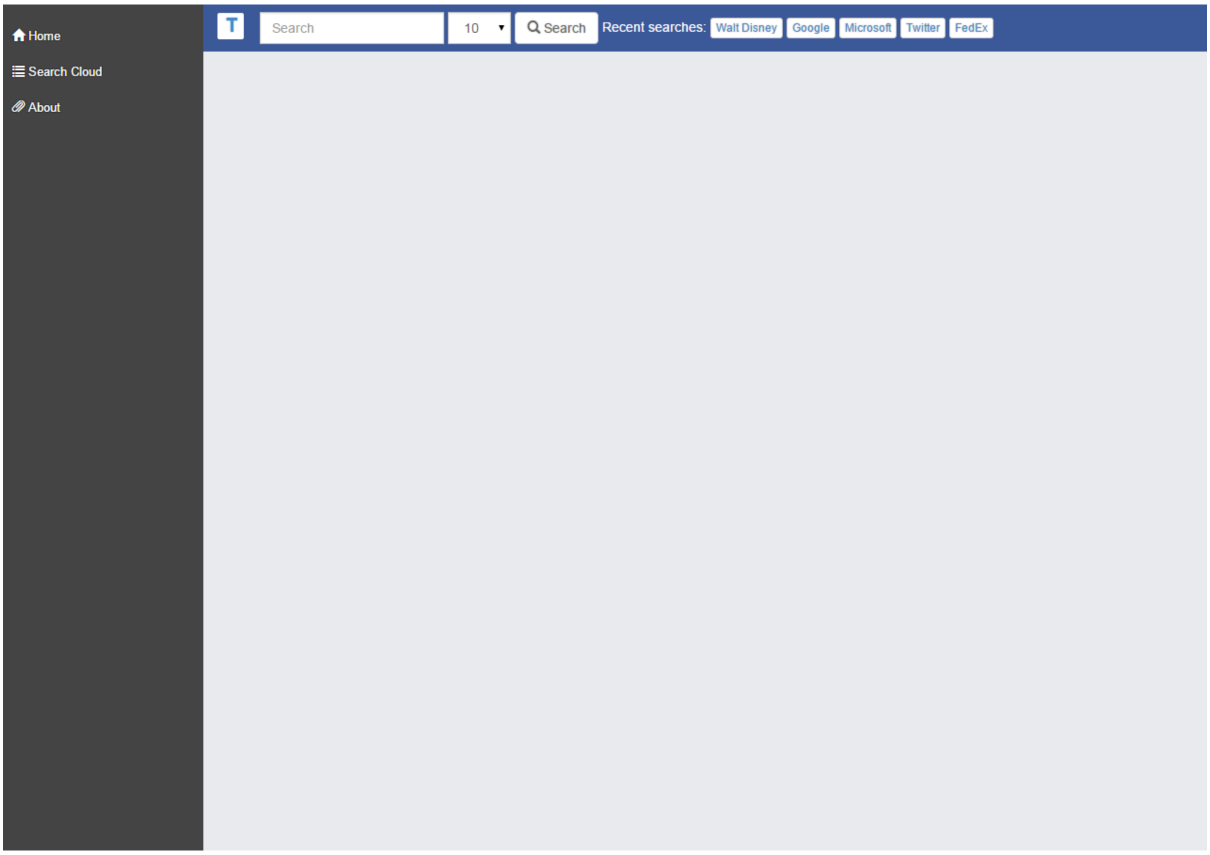
At present state of the application the user can perform a keyword search to retrieve the recent set of tweets which contain the supplied keyword. The application processes the set of tweets by extracting the actual text message from the tweet object and run it through the sentiment analysis. The sentiment analysis is based on pattern matching against a lexicon of adjectives where each lexeme has a predefined sentiment polarity score and final result is the average of all found adjectives in the analysed text. The final result for each tweet is then presented back to the user.

2.2. Front-end

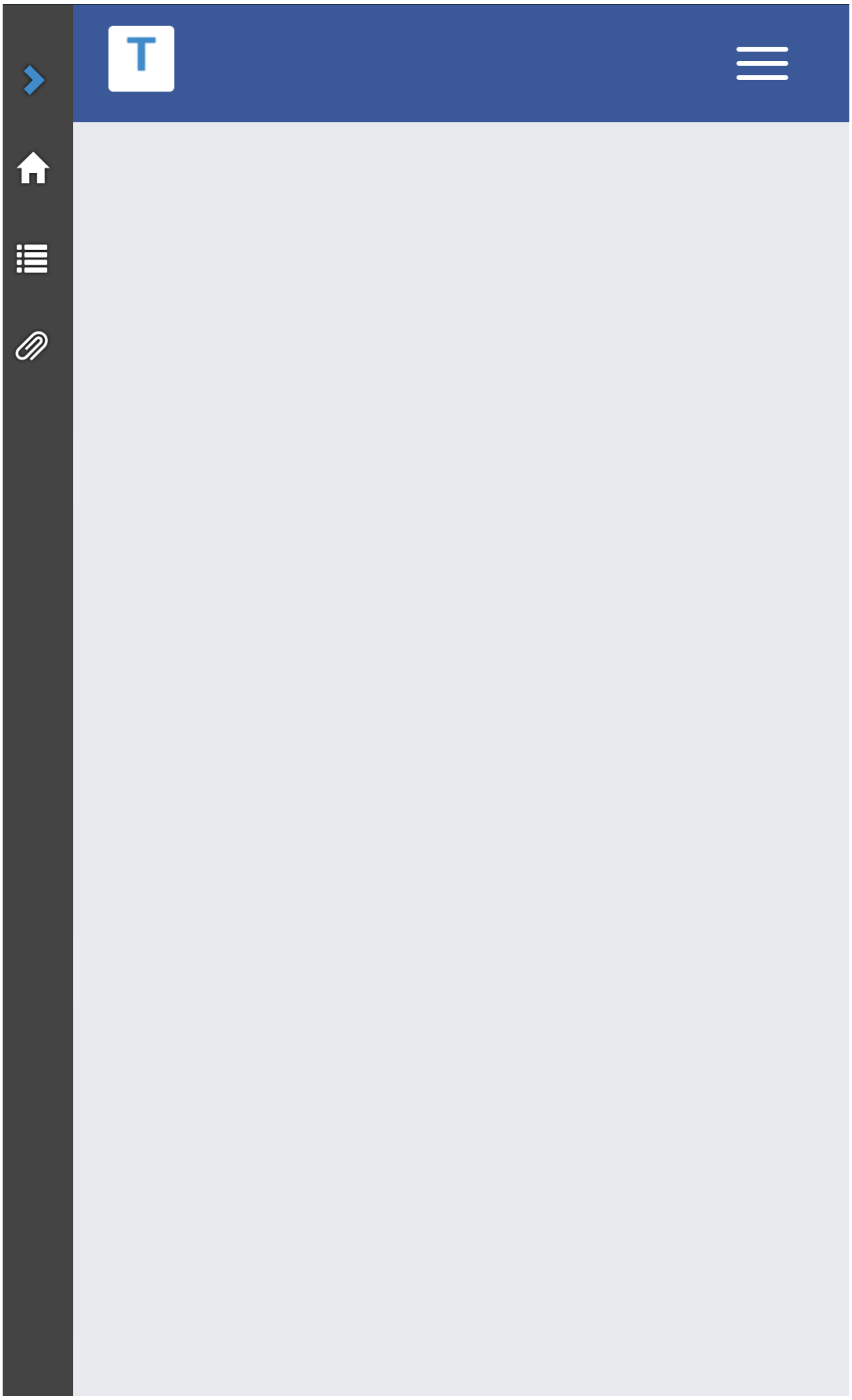
The submitted application currently allows the user to perform a search followed by sentiment analysis, access the search cache containing all the keywords used in the past. The application does not support any login facilities as it is using an application-only authentication. Its functionality was built using the Python 3.4.2 programming language whereas the GUI has been built using combination of HTML5, CSS, jQuery and Flask template framework.

2.2.1. Home Screen

2.2.1.1. Desktop

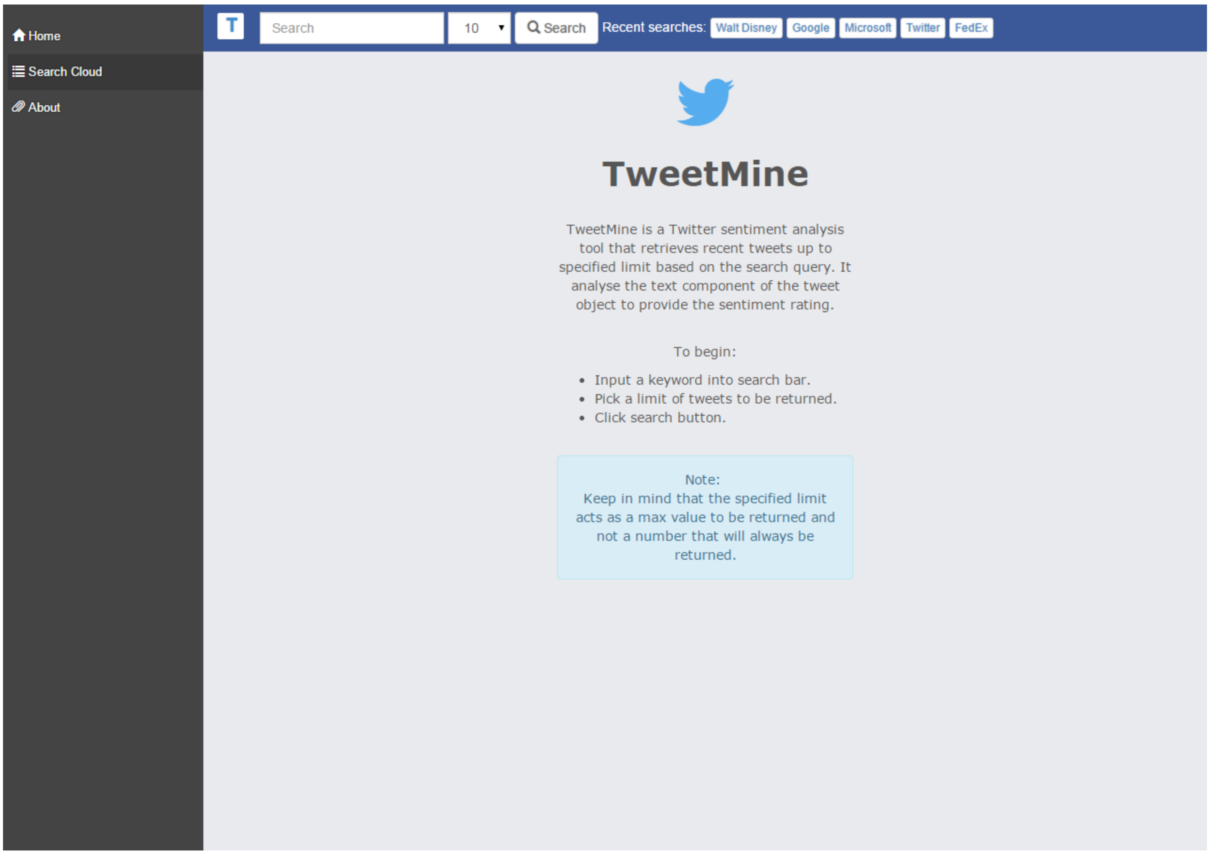


2.2.1.2. Mobile



2.2.2. About Screen

2.2.2.1. Desktop



TweetMine

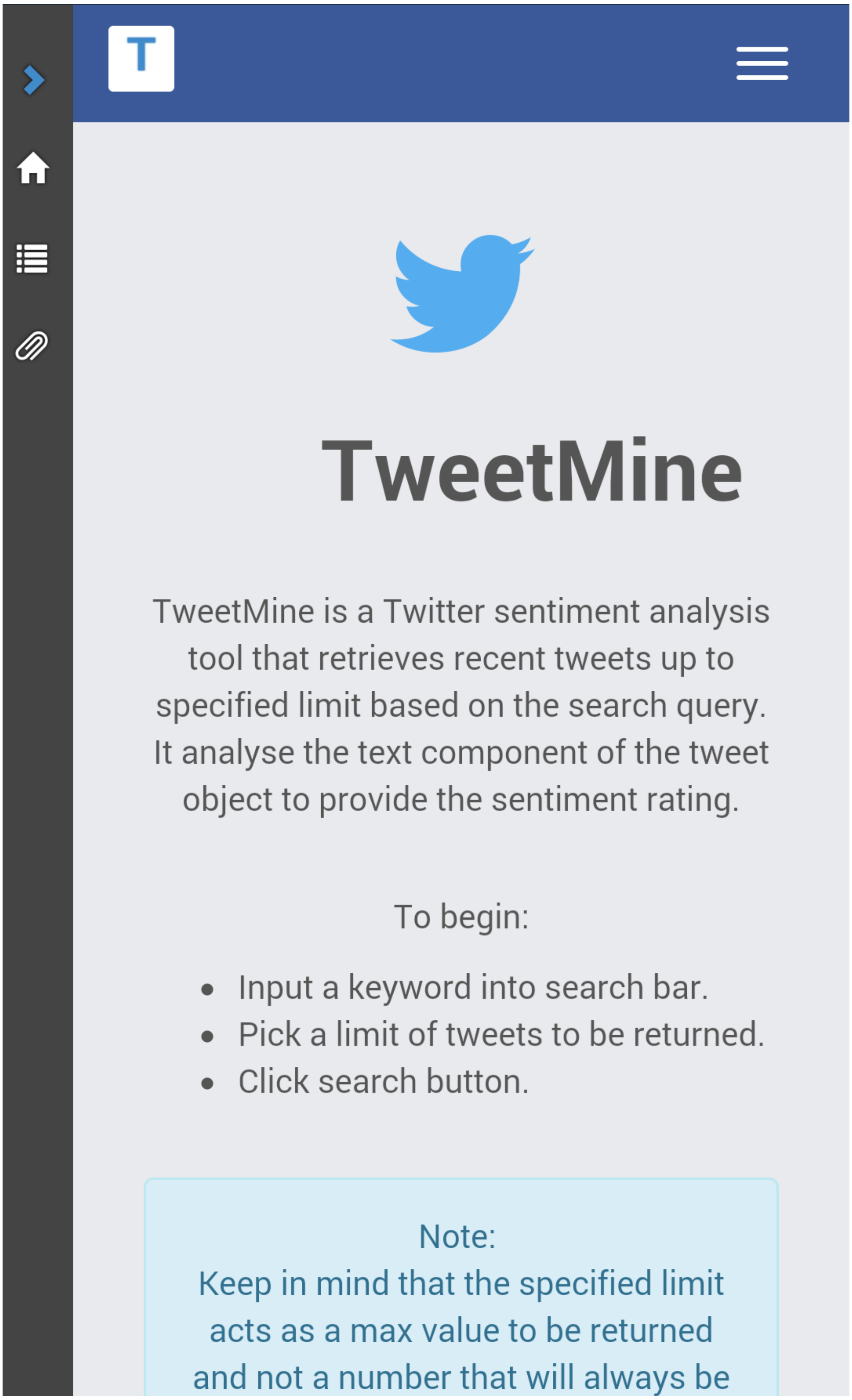
TweetMine is a Twitter sentiment analysis tool that retrieves recent tweets up to specified limit based on the search query. It analyse the text component of the tweet object to provide the sentiment rating.

To begin:

- Input a keyword into search bar.
- Pick a limit of tweets to be returned.
- Click search button.

Note:
Keep in mind that the specified limit acts as a max value to be returned and not a number that will always be returned.

2.2.2.2. Mobile



TweetMine

TweetMine is a Twitter sentiment analysis tool that retrieves recent tweets up to specified limit based on the search query. It analyse the text component of the tweet object to provide the sentiment rating.

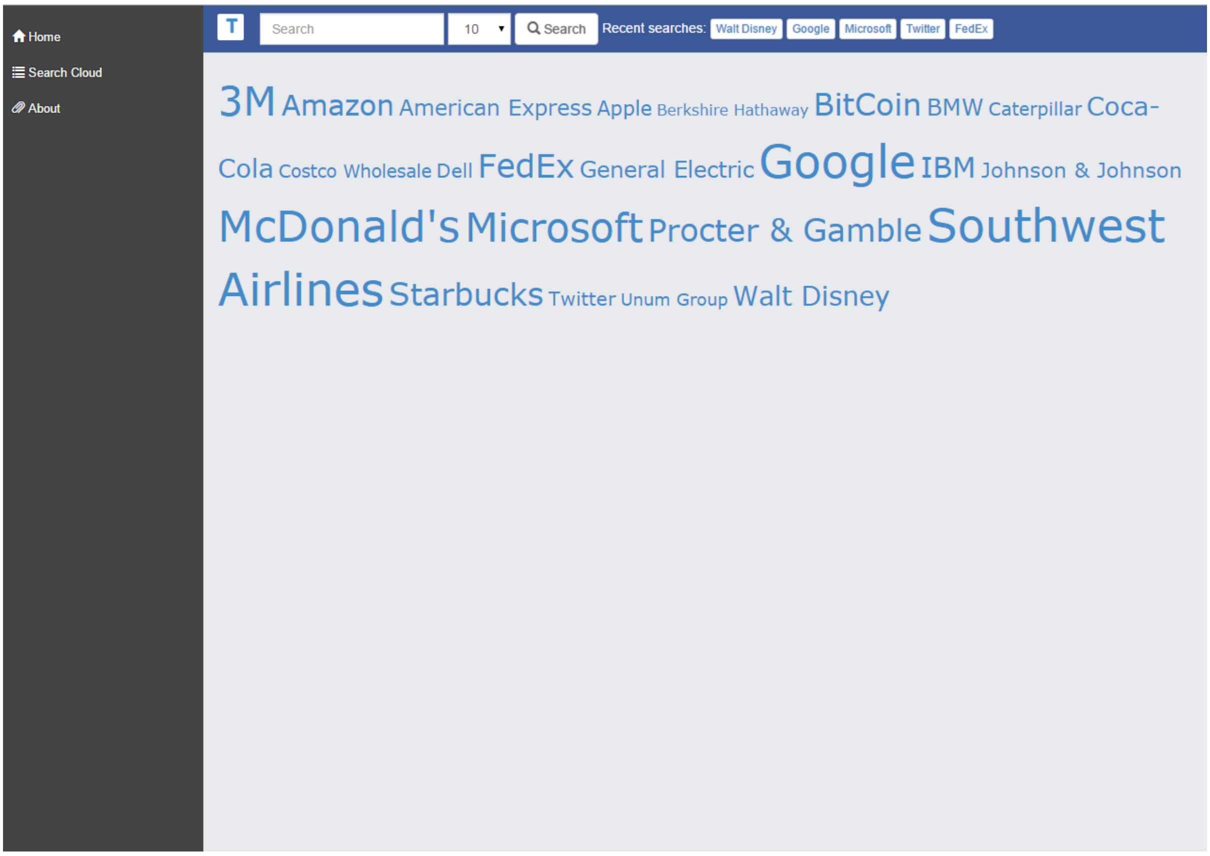
To begin:

- Input a keyword into search bar.
- Pick a limit of tweets to be returned.
- Click search button.

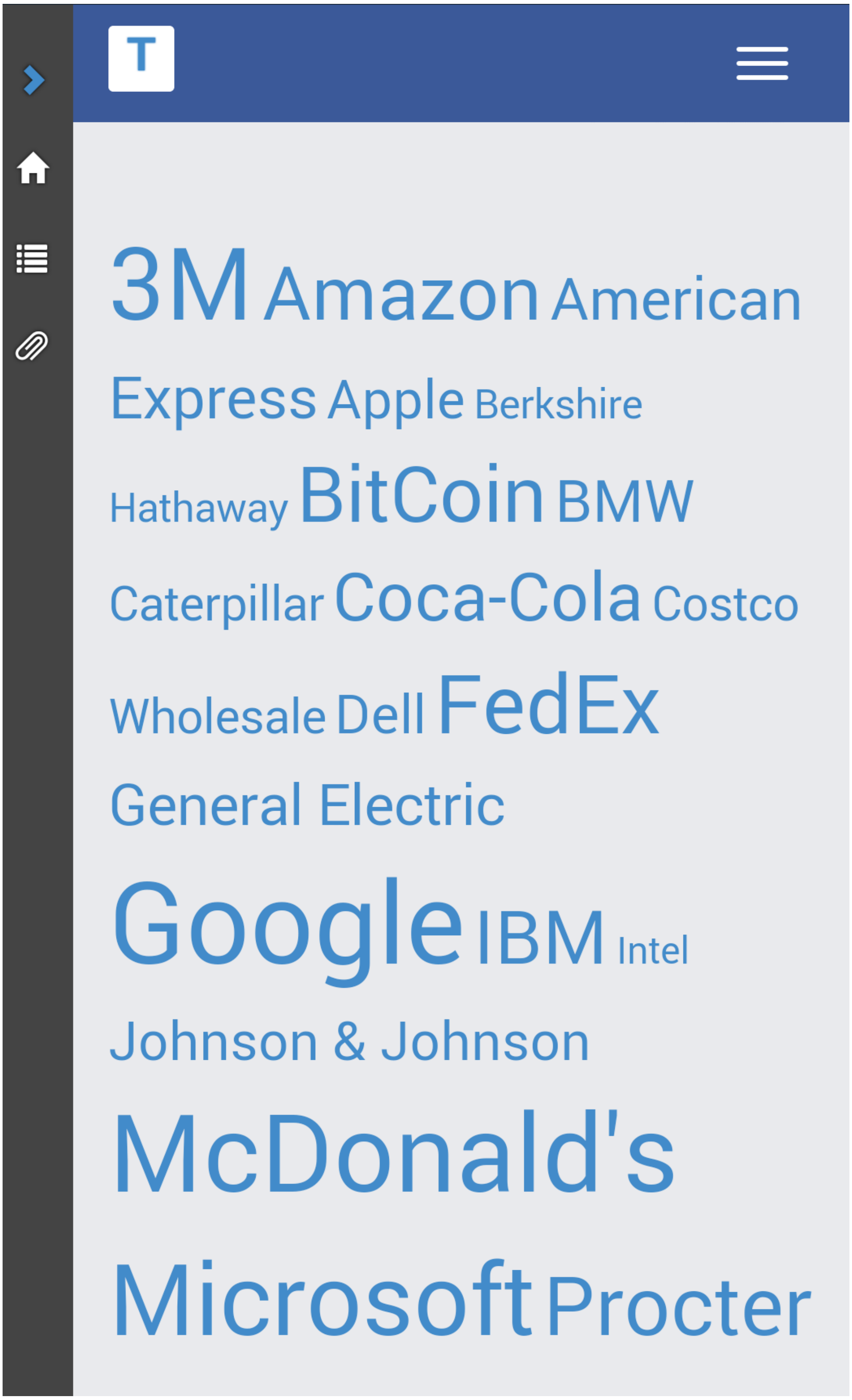
Note:
Keep in mind that the specified limit acts as a max value to be returned and not a number that will always be

2.2.3. Search Cloud Screen

2.2.3.1. Desktop

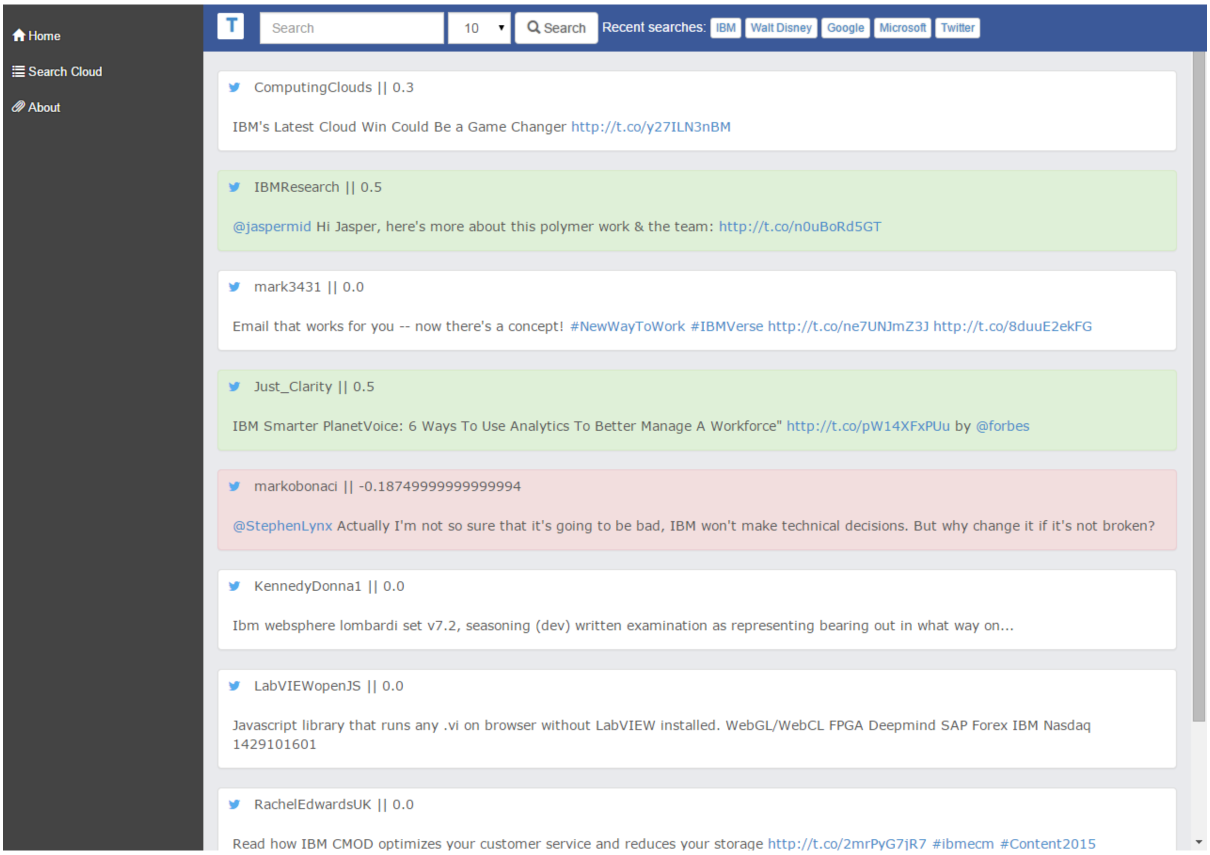

Home
Search Cloud
About
T
Search
10
Search
Recent searches:
Walt Disney
Google
Microsoft
Twitter
FedEx
3M Amazon American Express Apple Berkshire Hathaway BitCoin BMW Caterpillar Coca-Cola Costco Wholesale Dell FedEx General Electric Google IBM Johnson & Johnson McDonald's Microsoft Procter & Gamble Southwest Airlines Starbucks Twitter Unum Group Walt Disney

2.2.3.2. Mobile


T
3M
Amazon
American Express
Apple
Berkshire Hathaway
BitCoin
BMW
Caterpillar
Coca-Cola
Costco Wholesale
Dell
FedEx
General Electric
Google
IBM
Intel
Johnson & Johnson
McDonald's
Microsoft
Procter

2.2.4. Result Screen

2.2.4.1. Desktop



ComputingClouds || 0.3

IBM's Latest Cloud Win Could Be a Game Changer http://t.co/y27ILN3nBM

IBMResearch || 0.5

@jaspermid Hi Jasper, here's more about this polymer work & the team: http://t.co/n0uBoRd5GT

mark3431 || 0.0

Email that works for you -- now there's a concept! #NewWayToWork #IBMVerse http://t.co/ne7UNJmZ3J http://t.co/8duuE2ekFG

Just_Clarity || 0.5

IBM Smarter PlanetVoice: 6 Ways To Use Analytics To Better Manage A Workforce" http://t.co/pW14XFxPUu by @forbes

markobonaci || -0.18749999999999994

@StephenLynx Actually I'm not so sure that it's going to be bad, IBM won't make technical decisions. But why change it if it's not broken?

KennedyDonna1 || 0.0

Ibm websphere lombardi set v7.2, seasoning (dev) written examination as representing bearing out in what way on...

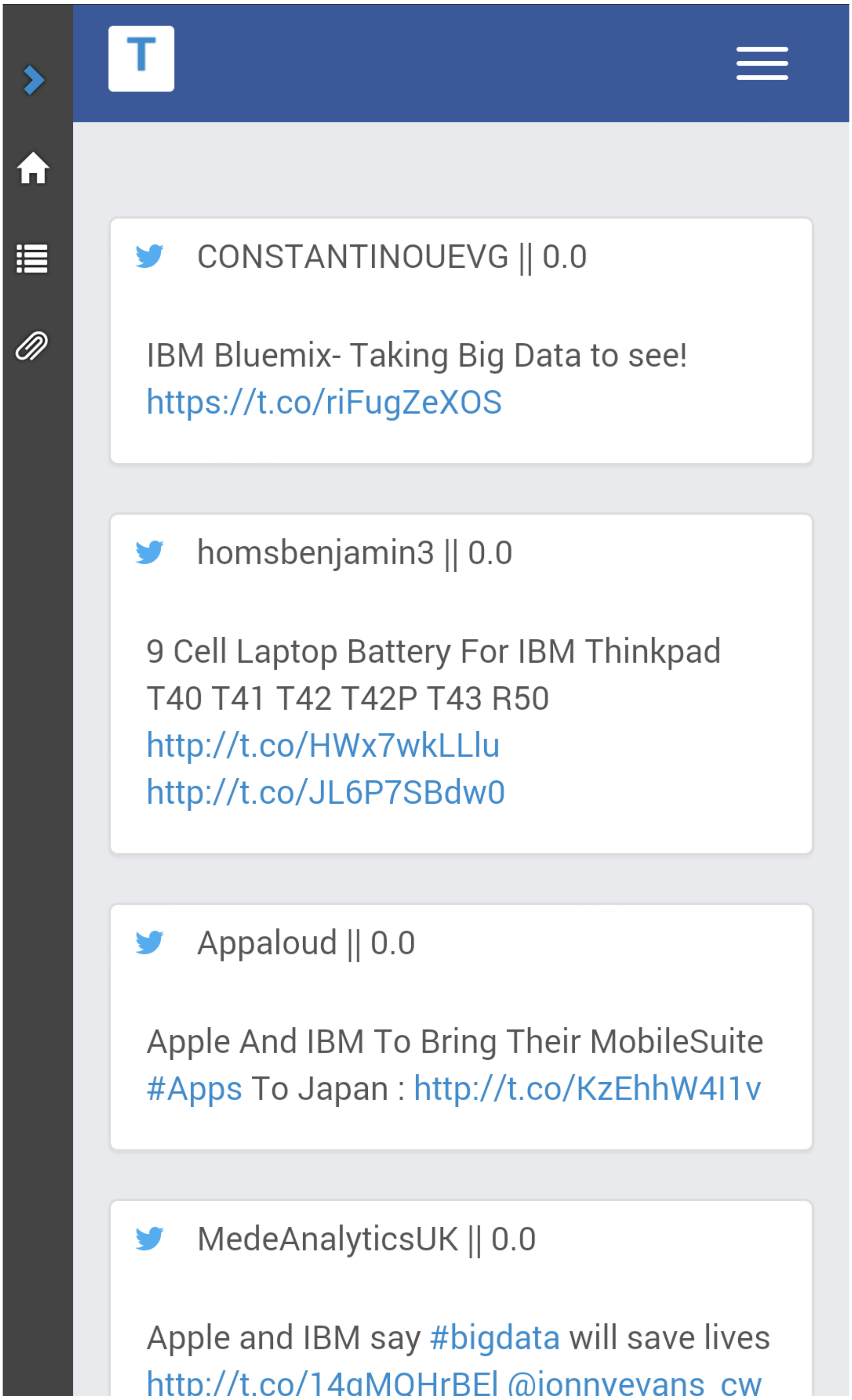
LabVIEWopenJS || 0.0

Javascript library that runs any .vi on browser without LabVIEW installed. WebGL/WebCL FPGA Deepmind SAP Forex IBM Nasdaq 1429101601

RachelEdwardsUK || 0.0

Read how IBM CMOD optimizes your customer service and reduces your storage http://t.co/2mrPyG7jR7 #ibmecm #Content2015

2.2.4.2. Mobile



CONSTANTINOUEVG || 0.0

IBM Bluemix- Taking Big Data to see! https://t.co/riFugZeXOS

homsbenjamin3 || 0.0

9 Cell Laptop Battery For IBM Thinkpad T40 T41 T42 T42P T43 R50 http://t.co/HWx7wkLLlu http://t.co/JL6P7SBdw0

Appaloud || 0.0

Apple And IBM To Bring Their MobileSuite #Apps To Japan : http://t.co/KzEhhW4I1v

MedeAnalyticsUK || 0.0

Apple and IBM say #bigdata will save lives http://t.co/14qMQHrBEl @jonnyevans_cw

2.3. Back-end

At the time of submission of this project the back-end of the application consist of SQLite3 database system holding the search cache of the application accesses through Python module. The communication with Twitter is handled by a Python wrapper for Twitter API called Twython.

3. Description of Conformance to Specification & Design

3.1. What Was Achieved from Specification and Design

3.1.1. Connection to the Twitter API

The essential connection with Twitter has been achieved by the use of Twython, a Python wrapper for Twitter API developed by Ryan McGrath. The current release supports both Python 2.6+ and Python 3. It is actively maintained by its creator but input from the outside is always welcomed.

3.1.2. Sentiment Analysis

Sentiment analysis which is the primary focus and ultimate goal of the application has been achieved with the implementation of the TextBlob Python library used in the processing of the textual data. It's sentiment capabilities and implementation provides a basic analysis through pattern matching based on the *pattern* library currently used in the application and also more advanced analysis using the Naïve Bayes method with a trained classifier.

3.1.3. HTML DOM Parser

HTML DOM Parser was an additional requirement that surfaced after the connection to the Twitter API was established and the tweets were received by the application. Source links as well as topics and hashtags were retrieved in base text format not as those originally appear on Twitter. The function of the parser was to scan the web document for such elements of the tweet and convert them back to their original state.

3.2. What Wasn't Achieved from Specification and Design

3.2.1. Smart Statistics

The scope of Smart Statistics involved a graphical representation of the trend line of the sentiment from the retrieved tweets. The development of this functionality has been started with the use of Charts.js and although the development process outcomes seemed promising this functionality has been put on hold as issues raised with the <*canvas*> element of HTML5 upon the resizing of the screen or switching between portrait/landscape on mobile devices rendering the canvas blank and unresponsive.

3.2.2. Cloud Deployment

The application was developed locally in its entirety and cloud deployment was scheduled as a final touch to the application once it was fully functional with all previous goal achieved. The time window available proved to be too short to fully convert the application to be deployed on the cloud as the support for technologies used was miniscule.

4. Description of Learning

4.1. Technical Learning

During the course of this project a greater understanding and knowledge has been gained on the web application development using different programming and scripting languages. By initially investigating the possible technologies that could potentially be used in the development of the application discovered the primary solutions were discovered and further investigation allowed the development cycle to begin. Having previously touched on the subject of web application development in this year's course it was an opportunity to further develop coding abilities in the area of web development.

Exposure to an external APIs during the course of the development cycle introduced a steep learning curve due to the lack of background in such complex systems. A lot of learning had to be achieved in order to understand the concept, structure and use of the system components.

4.2. Personal Learning

Throughout the course of this project I have discovered that I am capable of combining variety of programming and scripting languages as well as to adapt and learn languages I had no or little previous experience working with. That aside I have also discovered that despite the fact of being a capable Python programmer my personal preference lies with the front end technologies such as HTML, CSS and JavaScript (jQuery) as it is the area I enjoy working with the most.

Thanks to the presentation held at the end of each iteration I have further developed my presentation and communication skills as well as build up confidence required to successfully present.

5. Review of Project

5.1. Problems Encountered

5.1.1. Twitter API Wrapper

With the start of the development cycle I have initially used *tweepy* as the Python wrapper for Twitter. After the setup and registration of the application with Twitter I have begun the development of one of the core functionalities that is the connection to the Search API of Twitter with the several attempts on the use of both OAuth 1.0 and OAuth 2.0 through the wrapper and final success it was revealed that the application would require a PIN number to be submitted through the command line input which would not suit a web application. This step back was caused by the lack of proper documentation available at the time in regards to the capabilities of the *tweepy* wrapper. The development using *tweepy* had to be discontinued and with that decision I have switched to the *Twython* wrapper which turned out to be a perfect solution.

5.1.2. Chart.js

With the start of 3rd iteration of the development I was beginning to work on a small functionality called Smart Statistics. It involved the creation of graphical representation of the tweets sentiment scores as a trend line to be presented to the end user for the purpose of providing a quicker feedback without relying on the user to manually scan through the results. After trying out few available tools I have decided to begin the development using *Chart.js* written in JavaScript for the use of the HTML5 <*canvas*> element. Despite promising outcomes the development was put on hold as the graph became blank and unresponsive once the dimensions of either desktop or mobile browser changed.

5.2. Future Advice

Advice for anyone doing this or a similar project would be to make a finite decision on the technology to use as soon as possible as a lot of development could be ahead and time is very important.

The use of Git would be helpful in this project as it allows an efficient version control in case issues arise and code needs to be restored to its previous state. It also allows the branching of the code base allowing certain parts being developed independently.

5.3. General Feedback

Overall I am pleased with the design the application has taken as it is clean and easy to navigate around allowing the users to quickly familiarise themselves with the application. Also despite the fact than certain functionalities were not completely developed the main focus and its functionality is available and performs without any issues.

6. Acknowledgements

I would like to thank my project supervisor Dr. Greg Doyle for giving me the opportunity to work on this project, expand my knowledge and skillset as well as for being available at times I required assistance, insight or general feedback on the approach and the course of the development. Also I would like to thank my Web and Cloud Development lecturer Paul Barry for the introduction to a set of vital web technologies and database systems that I have used in this project and for the suggestion on how to tackle the sentiment analysis using Python. My final thanks go to Joseph Kehoe the acting head of the project module this year for all the help he had provided me with in relation to the project.