

System Architecture
Sentiments Analysis with Twitter
Team Jazz Men

Anagh Goswami 1217426

Meet Pandya 1214306

Jasman Gill 1211554

Jesse Truong 1222722

Jia Xu 1213268

April 9, 2017

Contents

| | | |
|----------|-------------------------------|----------|
| 1 | Introduction | 1 |
| 1.1 | Overview | 1 |
| 2 | Design Principle | 1 |
| 3 | Anticipated Changes | 1 |
| 3.1 | Likely Changes | 1 |
| 3.2 | Unlikely Changes | 2 |
| 4 | Flow Diagram of Web UI | 2 |
| 5 | Module Decomposition | 2 |
| 5.1 | HTML | 3 |
| 5.1.1 | Index | 3 |
| 5.1.2 | Login | 3 |
| 5.1.3 | Custom Search | 3 |
| 5.2 | PYTHON | 3 |
| 5.2.1 | Web | 3 |
| 5.2.2 | Script | 3 |
| 5.2.3 | Data Object | 4 |
| 5.3 | JAVASCRIPT | 5 |
| 5.3.1 | JS | 5 |
| 5.3.2 | Custom Page JS | 5 |
| 5.4 | Deployment | 5 |
| 5.4.1 | Heroku Web Server | 5 |
| 6 | Traceability | 6 |
| 6.1 | Requirements | 6 |

List of Tables

| | | |
|---|-------------------------------------|---|
| 1 | Revision History | 1 |
| 2 | List of Modules | 6 |
| 3 | Requirements Traceability | 7 |

List of Figures

| | | |
|---|------------------------|---|
| 1 | Flow Diagram | 2 |
|---|------------------------|---|

Table 1: Revision History

| Description of Changes | Author | Date |
|------------------------------------|---------------|---------------|
| Created first draft | Meet Pandya | Jan 8, 2017 |
| Worked on Traceability Requirement | Anagh Goswami | April 7, 2017 |
| Worked on Module Decomposition | Meet Pandya | April 8, 2017 |
| Edited for Revised Final Version | Meet Pandya | Apr 6, 2017 |

1 Introduction

1.1 Overview

This document is provides a detailed description of System Architecture for this project. The document describes Anticipated Changes as the project progresses which is split up into Likely and Unlikely Changes. Next, the document describes the System Architecture using Client-Server model.

2 Design Principle

The Client-Server model has been used as the main principle of design for this project. This model is an obvious choice for this project because the project is based around a WebApp where all the functionalities are present. There is a front-end client side involved where user sends requests and the server side back-end stores user information for future use. Implementing modules becomes easy due to the Client-Server model which allows each module to be put into either/or category.

3 Anticipated Changes

3.1 Likely Changes

The following changes are likely to occur as the project proceeds:

AC1 Currently, the Sentiments Analysis is static, meaning the keyword to be searched for on Twitter is Hard-Coded. Eventually, it will become dynamic, so users will be able to input keyword and thats what the analysis is done on.

AC2 The product only supports Sentiments Analysis on a single keyword. Eventually, the product will support more than one keyword in order to do Comparative Analysis.

AC3 The web UI currently is hosted locally on tester's computer. Eventually, the web UI will be hosted on a live server.

AC4 Based on user feedback, some features will be modified if not, removed.

3.2 Unlikely Changes

The following changes are unlikely to occur:

UC1 It is unlikely that the Sentiments Analysis will be done with any API other than the
Alchemy Language API

UC2 The way the Front End UI is developed is unlikely to change.

4 Flow Diagram of Web UI

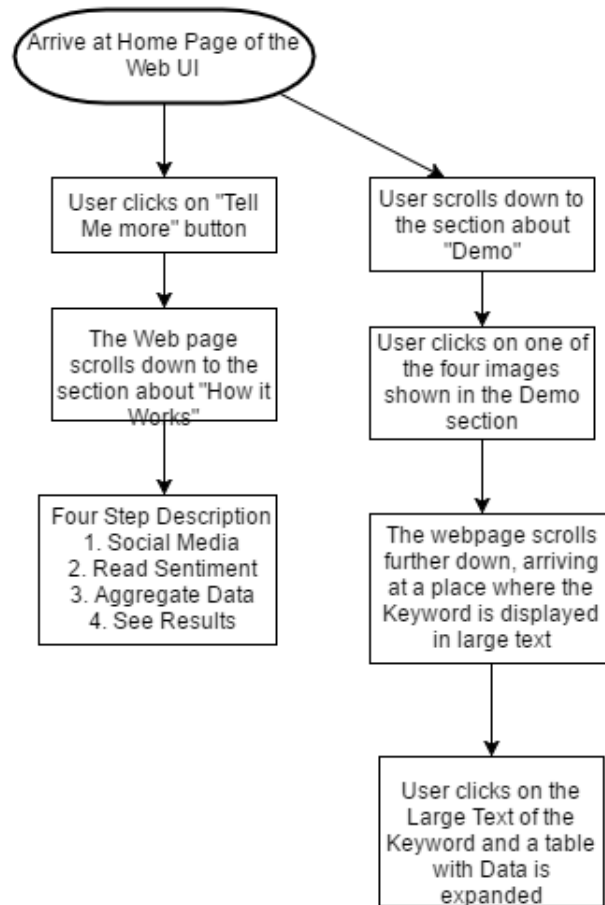


Figure 1: Flow Diagram

5 Module Decomposition

The modules have been decomposed into three categories divided by the respective programming languages utilized. The three categories are HTML, Python and Javascript. Each of the three categories have their respective modules. The categories make it easier to distinguish between different implementations. The categories are used simply for ease of understanding.

5.1 HTML

The HTML category has 3 main modules, Index, Login and Custom and each are described in detail below.

5.1.1 Index

This module defines the home page of the web application. It serves as a more visually appealing sitemap for users. The Index module communicates and accesses modules from the Javascript category of modules.

5.1.2 Login

This module defines the view of the Login and Registration page of the web application.

5.1.3 Custom Search

This module creates a view where user can access the custom search bar functions of the web application.

5.2 PYTHON

The Python category has 3 modules as well and their respective functions are described below.

5.2.1 Web

The Web module allows communication between all pages and serves as an access point/cross-roads between different functionalities of the web application.

5.2.2 Script

The Script module serves the purpose of retrieving twitter data based on either custom search or predefined categories. It includes retrieving data from the Twitter API and passing it on to the Alchemy API in order to get the sentiments scores. Below are a list of key functions of this module that help achieve its tasks.

5.2.2.1 API Access

This function stores the authentication keys that allow the web application to access the third party APIs.

5.2.2.2 Data

This function is used to retrieve twitter data based on inputted keywords. The inputted keyword is passed onto the Twitter API module and it becomes the input for that module.

5.2.2.3 Search

The function of this module is to take a keyword input and utilize the Twitter API to retrieve tweets and assign their sentiment scores. The resulting score and the corresponding tweets are presented on the webpage in a tabular format.

5.2.2.4 Twitter API

This function is part of the search module and retrieves a specific number of tweets (100) corresponding to the user inputted string. The module also refers to the "API Access" module to complete its task.

5.2.2.5 Alchemy API

This function uses the list created from the Twitter API sub-module and sends it to the Alchemy API to retrieve their respective sentiments score. The module also refers to the "API Access" module to complete its task.

5.2.3 Data Object

This module deals with all the backend functionalities. These functionalities includes saving data, user registration, user login, user logout, saving twitter data associated with each user and user authentication for login. All the different functions used to create this module are described briefly below.

5.2.3.1 Store Tweets

This function saves the data retrieved from the Search module and saves it in the database while accounting for repeated entries.

5.2.3.2 Get Tweets

This function parses through the database and returns a list of historically saved data.

5.2.3.3 Custom Search Storage

This function simply stores twitter data associated with the user that has logged in.

5.2.3.4 Custom Data Retrieval

This function serves the purpose of returning data associated with a user as long as a user has logged in. The restriction is that only one result is returned per subject.

5.2.3.5 User Creation

This function creates a new user account as long as it does not already exists and saves it to the database.

5.2.3.6 User Authentication

This function returns an existing username by searching through the database which is then used to verify during the login process.

5.3 JAVASCRIPT

The Javascript category has 2 modules and they are described in detail below

5.3.1 JS

This module holds the dynamic functionalities of the home page of the web application. The homepage has predefined categories that user can select to see sentiments scores of. The predefined categories are Restaurants, TV Shows, Athletes and Universities. Under each of these, there are 4 specific popular items pertaining to the category. Once a user clicks on one of the categories, the module then updates the sentiments scores by using the Script module and it displays the sentiments scores for the 4 popular items of the selected category. The module also displays the Most Positive and Most Negative tweets in a tabular format. The table shows 10 Most positive and 10 Most Negative tweets along with the score, the user and the actual tweet.

If a user decides to open up a different predefined category then the code is reset and repeats the process for the newly selected category.

5.3.2 Custom Page JS

This module is based off of the JS module described earlier. The main purpose of this module is for additional views of search results based on custom search by the user. There are two views implemented by this module, a bubble view and a table view of the resulting tweets on a custom topic searched by the user. The Bubble view shows the 1 Most Positive Tweet and 1 Most Negative Tweet from the custom topic. While the Tables view shows a list of 10 Most Positive and 10 Most Negative tweets.

5.4 Deployment

5.4.1 Heroku Web Server

The Heroku web server module serves as the final point of where the web application will be accessed from and where it will go live from. The Heroku web server will be used to deploy the web application for the general public. Heroku is a cloud Platform-as-a-Service(PaaS) and allows for instant deployment and has very good performance. The link is provided below for the web application.

<https://jazzmen-capstone.herokuapp.com/>

Table 2: List of Modules

| Module | Function | System |
|---------------------|--|---------------|
| 1 Index | Defines the visual features of the homepage | Client |
| 2 Login | Defines the visual features of the login page | Client |
| 3 Custom Search | Defines the visual features of the custom search page | Client |
| 4 Web | Serves the page routing functions on the web application | Server |
| 5 Script | Facilitates the usage of the Twitter and the Alchemy API functions and ensures accuracy of results | Server |
| 6 Data Object | Builds and manages communication with the backend database | Server |
| 7 JS | Defines the functionalities for the static search categories and Homepage traversing actions | Client |
| 8 Custom Page JS | Defines the visual features of the dynamic search results from user inputted keyword | Client |
| 9 Heroku Web Server | Web Application deployed via the heroku web server | Client-Server |

6 Traceability

6.1 Requirements

Table 3: Requirements Traceability

| Requirement | Module(s) |
|-------------|-----------|
| 1 | 5 |
| 2 | 5 |
| 3 | 3,8 |
| 4 | 2,6 |
| 5 | 2,6 |
| 6 | 2,6 |
| 7 | 1,7 |
| 8 | 7 |
| 9 | 7 |
| 10 | 3 |
| 11 | 5 |
| 12 | 1 |
| 13 | 3 |
| 14 | 4 |
| 15 | 5 |
| 16 | 9 |
| 18 | 9 |
| 20 | 9 |