Phase 2 - CSE 385 Group M - Political Bias Calculator

Marcellus Mohanna, Dylan Williams, Zachary Sibert

***We attempted to address adding more parameters for our stored procedures, and improving our ER diagram documentation***

1. **What is the goal of your application? Why did you choose this topic? - changed**

The goal of the application is to determine the level of a Twitter user’s political engagement on the site alongside the side on which they lean (liberal, conservative, or moderate). Political engagement is calculated as a percentage of a user’s total tweets and retweets that are political compared to their total count of tweets. Political bias is calculated as the average bias of accounts that they follow. This is calculated by intersecting every user they follow with a set of users that are political by nature (e.g. President Trump’s account, the democratic party’s official account, etc.) and from that, determining if they lean in one direction or if they are moderate. We chose this topic because politics in social media is a huge topic in today’s society.

1. **What are the expectations of the application? What will you complete as well as what is out of scope? - changed**

The expectations are that we will be able to determine the above goals for a single or few users at a time. Being able to determine this for all users simultaneously or quickly will not be considered. Additionally, we initially planned to solely determine political bias based on the language that their tweets and retweets contain, however we quickly realized that was more in the realm of machine learning and natural language processing. We decided to make calculating bias more of a secondary feature and to simplify its calculation.

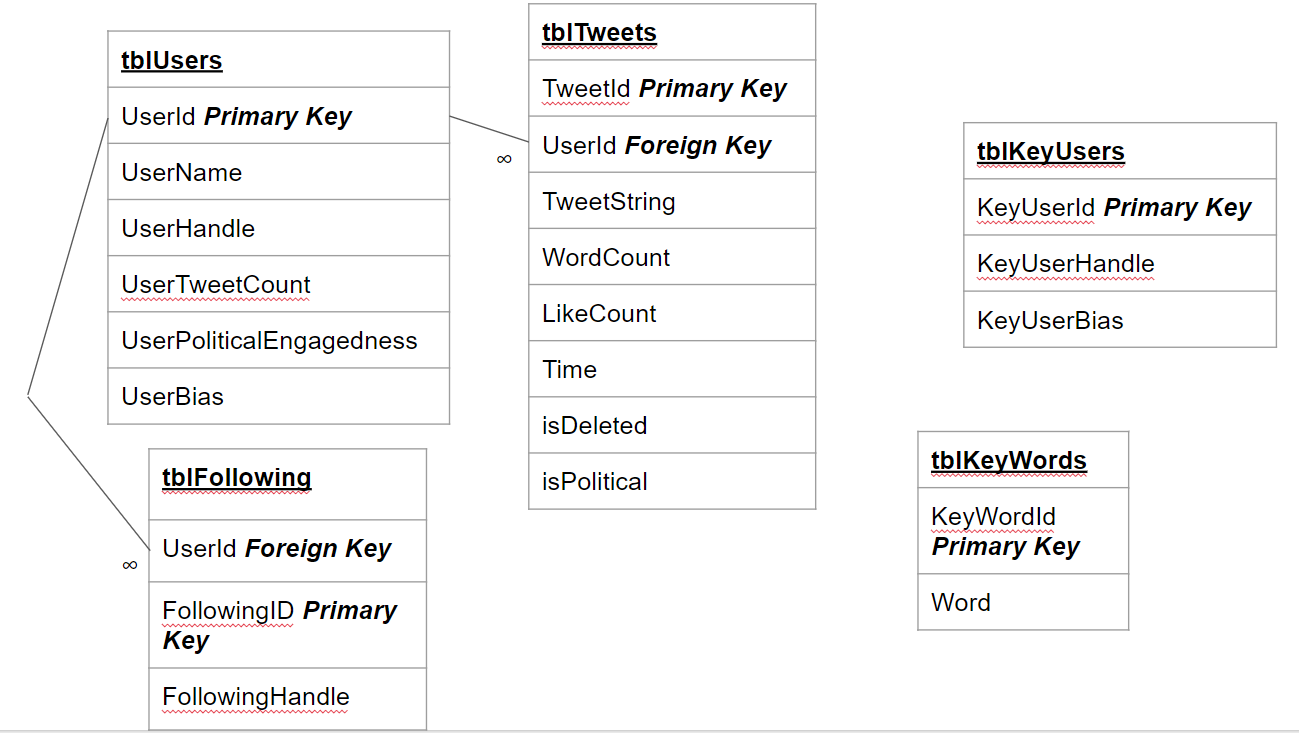
1. **What challenges will you face in completing this application and what is your plan to deal with them?**

The application will most likely not be one-hundred percent accurate but we will try to get as close as possible. Our biggest challenge will be accuracy due to the nature of our calculation techniques. Since we are using a “hard-coded”/predetermined list of political accounts and political topics, our accuracy is limited to their depth. To improve accuracy, we must add more sophisticated topics and more political accounts to compare with as a whole.

1. **Who will use this application and why? Describe at least 2 types of users.**

Advertisers might use this application target specific users for campaigns or political influence. General Twitter users might use this just to see what their political bias is based on their Tweets.

# ER Diagram



* **tblUsers**
  + UserID int (Primary Key)
  + UserName varchar(30) - a non-unique name for an account
  + UserHandle varchar(30) - a unique username for an account
  + UserTweetCount int
  + UserPoliticalEngagement float - a percentage of tweets and retweets from a user that are political
  + UserBias float - Represents a political bias on a scale from 0(liberal) to 1(conservative). This is calculated with a stored procedure
* **tblFollowing *(One user has many followed accounts)***
  + A table of users that follow other twitter accounts
  + FollowingID int - an identifier for an account that a User may follow
  + FollowingHandle varchar(30) - an account’s identifier
* **tblTweets *(One user to many tweets)***
  + TweetID int (Primary Key) - ID for a Tweet
  + TweetString - full tweet in a string
  + WordCount int
  + LikeCount int
  + Time smalldatetime
  + isPolitical
  + isDeleted - bit that tells if the tweet is deleted or not
* **tblKeyUsers** 
  + Popular political accounts (which we have defined) that a user might follow. These accounts will have a political bias value (determined set by us) as a bit type. This is used as a proxy to calculate a user’s political bias.
* **tblKeyWords**
  + Popular political words/topics that a user may tweet or retweet about. These are pre-defined by us and are used as a proxy to calculate a user’s political engagement.
  + KeyWordID (Primary Key) - ID for the Key Word
  + Word - the actual word that is the key word

# Stored Procedures

*spGetFollowing (handle) - removed*

*spGetTweets (handle) - removed*

*spCalculateBias (status) - removed*

*spCalculateBias (UserID) - changed*

* Returns a float between 0 (left wing) and 1 (right wing) representing a scale of political bias. If the user does not exists, returns -1

*spCalculatePoliticalEngagement (UserID)*

* A procedure that uses vwPoliticalTweets to calculate the level of a user's political engagement
* as a percentage expressed as a float (decimal value).
* Returns the level of political engagement and updates PoliticalEngagement in tblUsers

*spAddUser (UserName, UserHandle, UserTweetCount) -added*

* Adds the user into database, politicalEngagement and userBias are left to have default values initially

*spAddTweet (UserID, TweetString, WordCount, LikeCount, RetweetCount , Time) -added*

* Adds Tweets with all relevant information to the database

*spAddFollowing (UserID, FollowingHandle) - added*

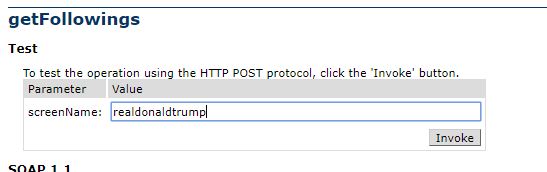
* Adds a following relationship. 1 user can follow many accounts
* UserID is of the user whose relationship you want to add and FollowingHandle is the handle that the UserID will follow

# Web App progress / How to use

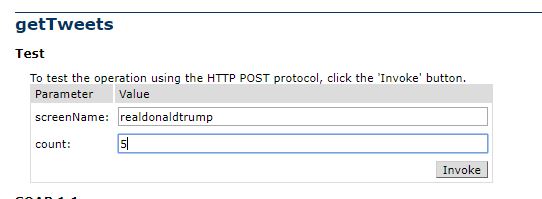
We have working website solution with a working Twitter API implementation with oAuth validation.

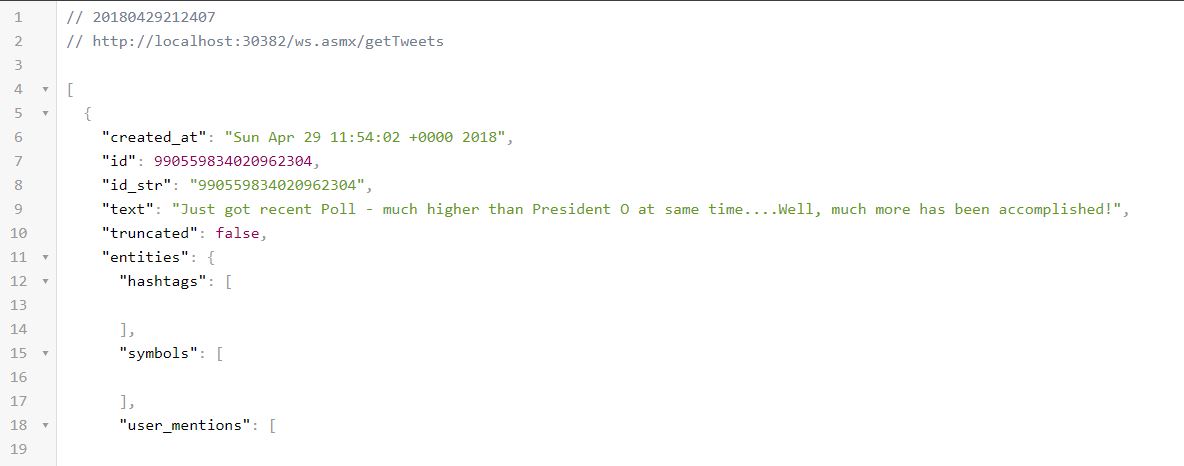
These two working methods below just output json straight from the twitter api, and **does not yet pull/push data to/from our SQL database.** To use these methods**, run the application with the “wc.cs” file open, NOT the “twitter\_index.html”**

* getTweets (screenName, count)
  + Takes in a screen name / @name, and returns the nth most recent tweets
* getFollowing (screenName)
  + Takes in a screen name / @name, and returns the nth most recent accounts that the particular user has followed









# Work to be done...

* Integrate stored procedure calls in our app code / access our database from C# code
* Possibly add more stored procedures, and implementing more twitter api function calls as we see fit
* Design an html page that gets data from our api