**Software Requirements and Design Document**

**For**

**Sentiment Analysis on Twitter**

Version 1.0

**Authors**:

Oscar Kosar-Kosarewicz

Andre Guiraud

Jacob Wharton

# Overview (5 points)

Our project uses a machine learning algorithm to discover how people feel about certain topics on the internet, or more specifically, hashtags on Twitter. We will use a website to get information from the user such as the topic to search for, and possibly the location to search in as well. Then the UI will output the data in a graph. For example, the user could search for the topic “impeachment” and get information about how people's feelings about this have changed over the past couple of years.

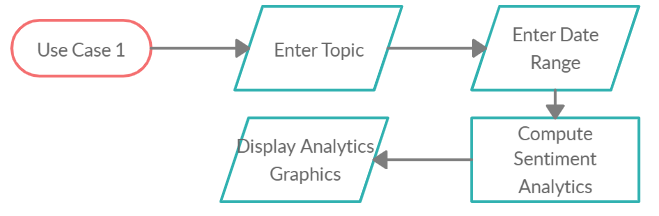
# Functional Requirements (10 points)

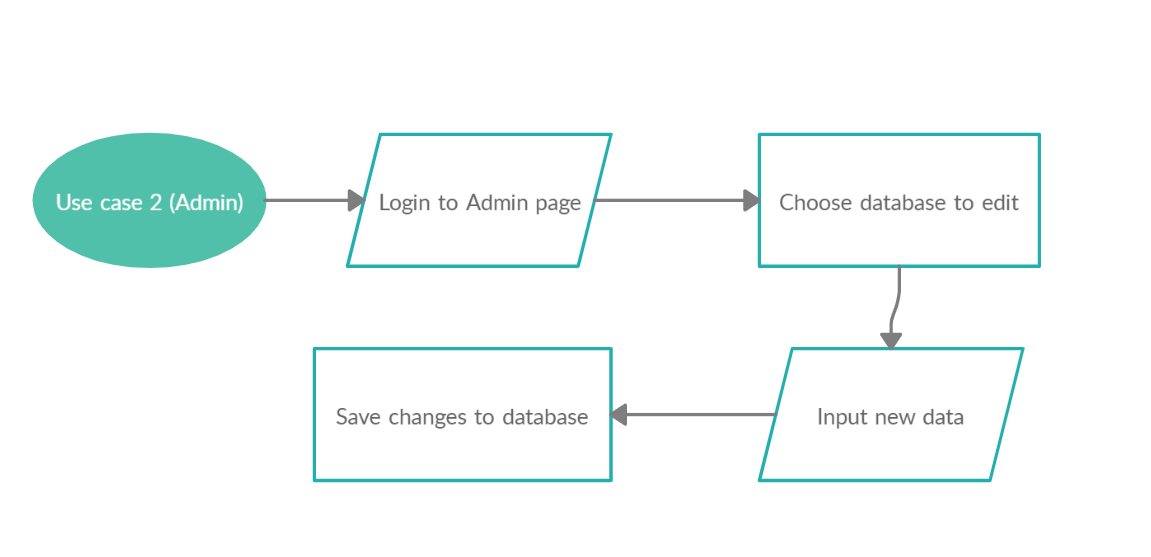
* High priority: Create a way to access tweets for analysis
* High priority: Create ML model that is able to analyze tweets and predict their sentiment
* High priority: Finish integration of ML model with webapp
* Medium priority: Generate a dataset of tweets with more nuanced descriptions than positive or negative so we can create better analytics
* Medium priority: Email is sent as soon as new data is generated for a specific query requested by x user
* low priority: Create additional visualizations of our data
* Low priority: Create additional database tables for our data

# Non-functional Requirements (10 points)

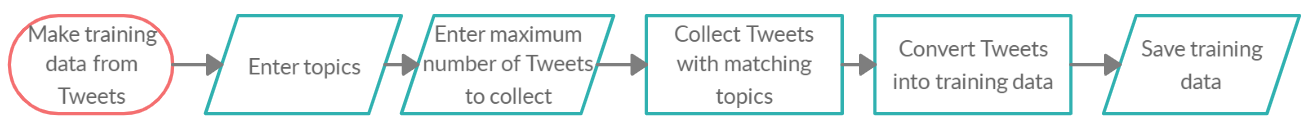
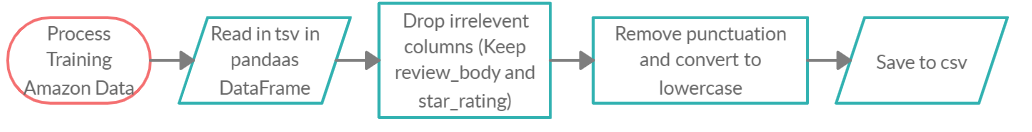
* Have the generated analytics be reasonably accurate
* Test database integrity to avoid errors when querying/saving
* Verify database relations for best performance/quality
* Ensure that the database of tweets is secure from modification outside our functions

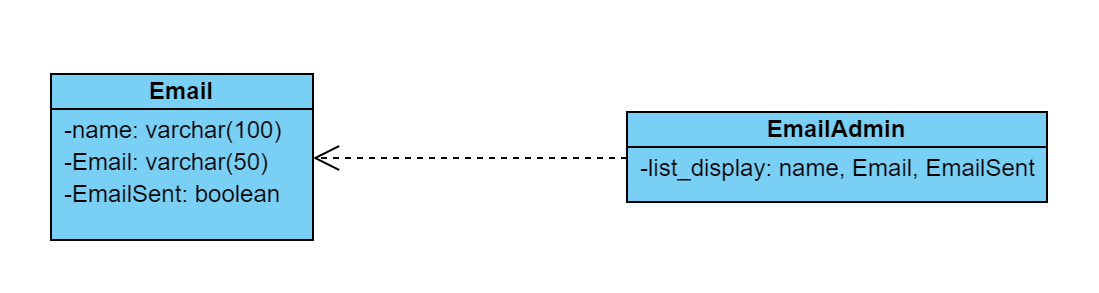
# Use Case Diagram (10 points)

**

**

# Class Diagram and/or Sequence Diagrams (15 points)





# Operating Environment (5 points)

The application will run in a (Chromium) web browser, using Python, JS, and HTML. Tested in Chrome.

At the moment the app only runs in a local server. To run the application the user must navigate to the application folder and in a command window run the following command:

python manage.py runserver

Then localhost:8000 (or a different port depending on the user) on the web browser.

Necessary for development:

>>>import textblob

>>>import nltk # required by textblob

>>>nltk.download('brown') # required by textblob

>>>nltk.download('punkt') # required by textblob

>>>import tweepy

>>>pip install django

Training model:

Python 3

import pandas, numpy, sklearn

# Assumptions and Dependencies (5 points)

We can assume that one factor that could affect our project is the company Twitter itself, which owns the rights to their tweets; if they decide not to let us access their API for any reason, that will affect our ability to gather training data, and will ultimately render our application nonfunctional.

Dependencies: Chromium, Twitter API, Google Translate API, textblob, tweepy, django