CIS 435 Capstone Project Report

Lucky Draw System for Retweets

By Ruoyu Li
UMID: 87760768

1. Introduction

This application enables users to set a lucky draw for retweeters of a specific tweet on Twitter, and the idea is inspired from Twitter's counterpart in China, Sina Weibo, where setting up lucky draws for a post (i.e. tweet) have become a vastly common way for earning followers and number of reposts. Generally, when a company or business wants to gain popularity, they would make an ad post on Sina Weibo stating that one or several of those who reposts this one have the chance of winning presents prepared by the author, and those presents are usually quite attractive ones, for instance, a Macbook or iPhone, or a free treat at a top restaurant. Such is human nature, that most people tend to repost these advertisements under the hope that he might win the lottery, and yet there is much less pain than really buying lotteries. On the other hand, companies and shops get to pay as little as the gifts' worth to earn themselves hundreds of thousands of repost, hence advertising themselves easily. Meanwhile, when I browse through twitter, never have I seen anything similar like this, so I got the idea of building some sort of luck draw game with Twitter retweets, through their APIs. And I have been able to realize drawing lucky users that retweets a particular tweet via PHP, and save the drawing result in a MySQL server.

2. Program Logic

2.1 Obtaining Retweets

2.1.1 Problems and Solutions

As soon as I had looked up Twitter's API, I found that the biggest problem is that Twitter ridiculously only allows querying the most recent 100 retweets of a specific tweet, and that apparently made this topic impossible to proceed from the very beginning. However, I soon came up with the idea of utilizing Twitter's search API to grab all the retweets of any post, but there were several limitations:

1. Twitter returns a maximum of 100 retweet statistics per query
2. Twitter may only search tweets dated back up to 1 week ago, and there is no other way to overcome this issue
3. There is also a rate limit, so that a single API may be called a limited number of times

As for the first one, each tweet has a unique ID that is a series of numbers, and the fact is that later tweets have bigger IDs. With this, I noticed that twitter also provides in its tweet search API a parameter called "**`max_id`**", so that twitter returns results with tweets whose IDs are less than or equal to a specific ID, which gives room for a loop algorithm to grab until the earliest

possible retweet in time order.

Then it comes to the second problem, which I have frankly not had any solutions, so the project gets limited by only being able to check all the retweets posted in the past week.

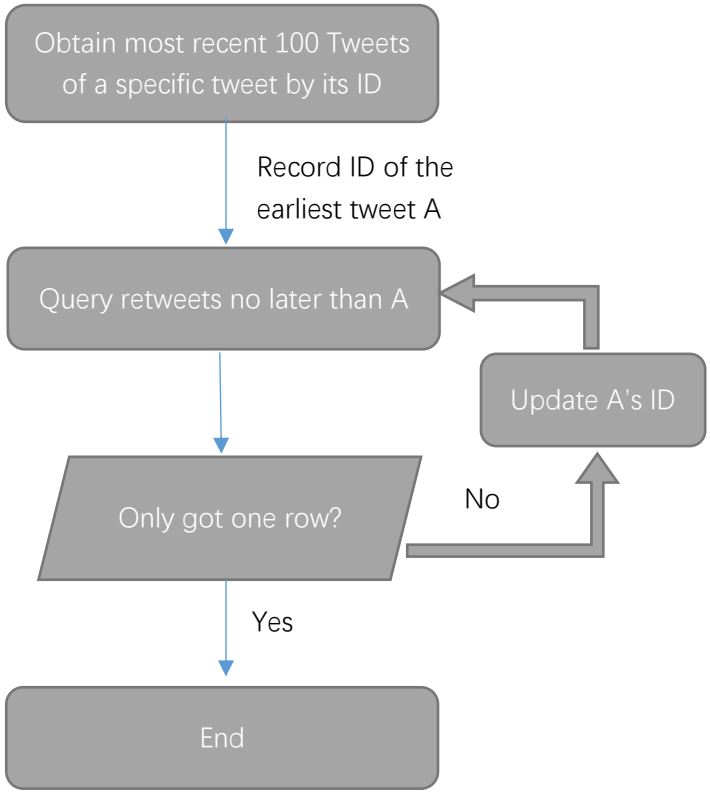
The third problem should have more importance if this application is actually going to the public market, but for an initial test, I put this issue away.

Twitter's GET search/tweets API sample *https://api.twitter.com/1.1/search/tweets.json*

Key parameters:

q	A UTF-8, URL-encoded search query of 500 characters maximum, including operators. Queries may additionally be limited by complexity.
result_type	Optional. Specifies what type of search results you would prefer to receive. In this implementation, ***recent*** is used, forcing timely ordered results.

2.1.2 Algorithm UML Diagram



2.2 Data Storage

The data storage is quite simple for this implementation, as all that needs to be kept are the winners of every retweet lucky draw, and enabling others to check if those records are real. I therefore used a single MySQL table, with the original tweet ID(**"tweetID"**), the winner's twitter ID(**"winnerID"**), and the winner's twitter account name(**"winnerName"**) on it, while **tweetID** is set as the primary key.

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
tweetID	VARCHAR(18)	☑	☑	☑	☐	☐	☐	☐	☐	
winnerID	VARCHAR(45)	☐	☐	☐	☐	☐	☐	☐	☐	NULL
winnerName	VARCHAR(45)	☐	☐	☐	☐	☐	☐	☐	☐	NULL

2.3 Integration of Querying Twitter and MySQL Database

It is designed that whenever the user tries to search the retweeting results of a tweet, the application first checks with the database to see if this retweet has already been Luck-Drawn, and if **true**, the page shows the winner's twitter id, and stops from further making queries to Twitter. This meanwhile acts as a validation system for anyone who have doubts about a previously drawn game. The application only has rights to **READ** and **ADD** records to the database, and no rights to **UPDATE** or **DELETE**, so that records cannot be modified.

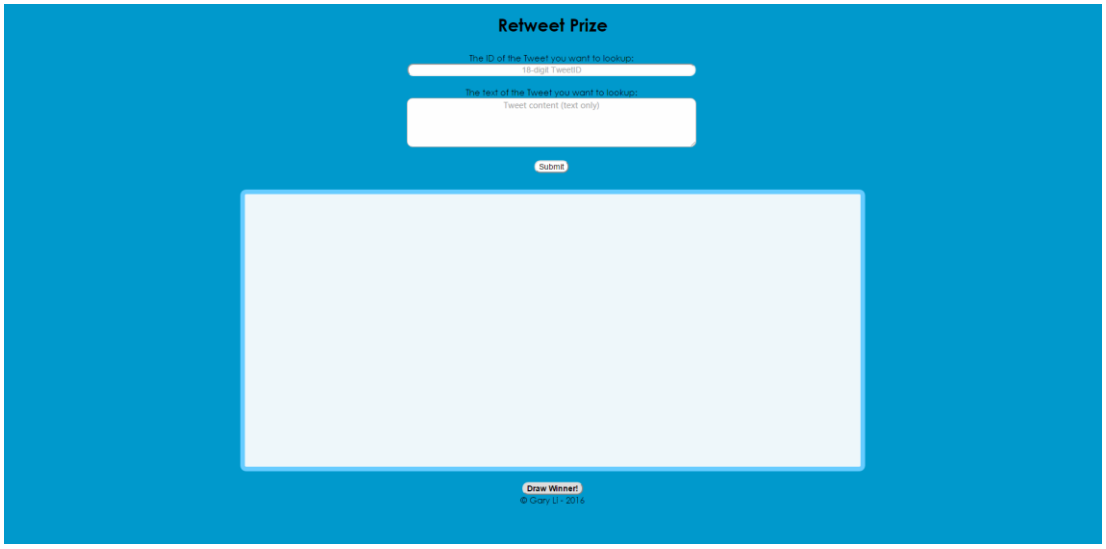
Then, if there is no record in the database, the application queries Twitter to grab all the retweets possible, and lists every retweets' unique ID and retweeters' unique ID. All retweeters' unique ID are pushed into an array, which is shuffled at the end of the query, hence acting as a ***Draw***, and when the user clicks the button at the bottom of the webpage, the winner's info prompts.

3. User Interface

The user interface is quite concise, with 2 fields both required to fill out before the user can submit (**http** method used: **post**).

When the submission is done, the server suns corresponding background **PHP** and displays in the box below either a "record exists" message, or a table of retweet information.

Additionally, there is no need to refresh the page before starting a new query, as the submit button also functions for clearing up the page.



4. Publish Over VPS

The application has been published on a VPS hosted by VPeasy.com, running Ubuntu 14.04 LTS, with manual setup of LAMP (**Linux**, **Apache**, **MySQL**, and **PHP**).
URL: http://104.238.223.24/index.php

5. Credits and Techniques

I finished the whole project as individual, and used Matt Harris's tmhOAuth to realize OAuth authentication with Twitter's API.
I have in this application obtained much deeper understanding of especially PHP and the **http GET** method for API utilizations, as well as first official practice of publishing my own web application.
Personally, I am deeply interested in web technologies, and this course has been very helpful in preparing me in the fundamentals, I will be exploring more into such technologies that provides much convenience for actual experience with communication without borders.