Initial Screen

Sridevi Balaga - sxb140530
Prathika Poludasu - pxp142130
Bala Sai Pavan Trinath Vutukuru - bxv131230

I. Introduction

Today, companies found major security issues that lead to massive data breaches in the cloud environment. Our project is focused on preventing threats in social networking data. We secure the cloud by detecting malware and preventing it before it enters the environment. We perform Sentiment Analysis to identify offensive sentiments in a tweet. For this project, we selected Twitter social networking platform due to the simplicity and richness of the API.

The NIST definition of cloud computing is "Cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction".

II. Design

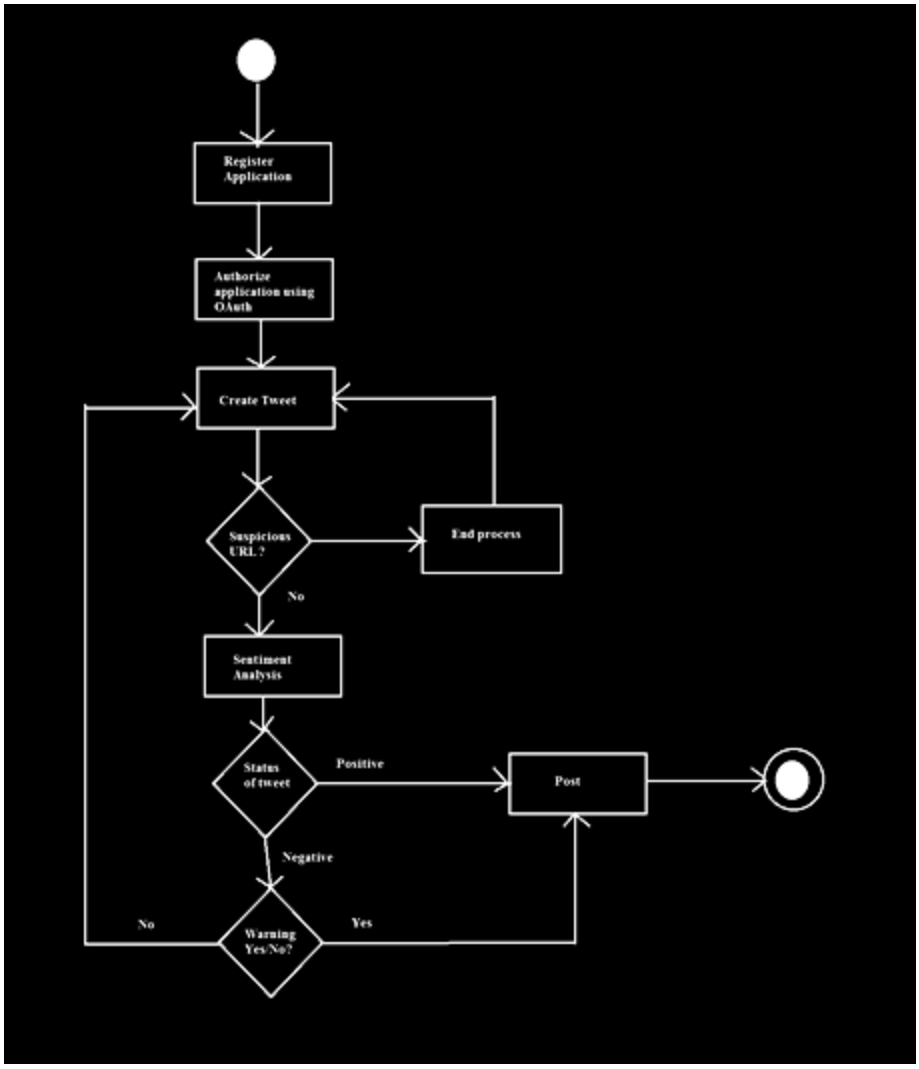


Figure 1: Activity diagram depicting the overall flow of control

Our application communicates successfully with Twitter using Twitter4j Library.
Our detection system was designed in such a way that it detects the malware links before the tweet gets posted. To achieve this, we used Google Safe Browsing (SFB) lookup API. A key should be generated to interact with GSB. This was done by enabling the GSB API in the developers console of our Google account. When the user attempts to post a tweet, we extract the URLs, if there are any, through regular expression pattern match. It is sent to GSB with the help of a key where the link is verified for the presence of any malware. Depending on the server response, we either notify the user about the presence of a suspicious link or just bypass to Twitter.And we used Stanford Core NLP API to analyze the sentiments of the tweet.

III. Tools & Libraries

- **OAuth**: OAuth is an open standard for authorization. OAuth provides client applications a secure delegated access' to server resources on behalf of a resource owner. It specifies a process for resource owners to authorize third-party access to their server resources without sharing their credentials. Designed specifically to work with Hypertext Transfer Protocol (HTTP), OAuth essentially allows access tokens to be issued to third-party clients by an authorization server, with the approval of the resource owner, or end-user. The client then uses the access token to access the protected resources hosted by the resource server. OAuth is commonly used as a way for web surfers to log into third party web sites using their Microsoft, Google, Facebook or Twitter accounts, without worrying about their access credentials being compromised.

- **Google Safe Browsing:** Google Safe Browsing is a service provided by Google that provides lists of URLs for web resources that contain malware or phishing content. The Google,Chrome, Apple Safari and Mozilla Firefox web browsers use the lists from the Google Safe Browsing service for checking pages against potential threats. Google also provides a public API for the service. Google also provides information to internet service providers, by sending e-mail alerts to autonomous system operators regarding threats hosted on their networks. According to Google, as of June 2012, some 600 million Internet users were using this service, either directly or indirectly. The *Safe Browsing API v2*, has the privacy advantage: "API users exchange data with the server using hashed URLs so the server never knows the actual URLs queried by the clients".

- **Sentiment Analysis with Stanford CoreNLP:** Sentiment analysis (also known as opinion mining) refers to the use of natural language processing, text analysis and computational linguistics to identify and extract subjective information in source materials. Most sentiment prediction systems work just by looking at words in isolation, giving positive points for positive words and negative points for negative words and then summing up these points. That way, the order of words is ignored and important information is lost. In contrast, Stanford CoreNLP uses new deep learning model actually builds up a representation of whole sentences based on the sentence structure. It computes the sentiment based on how words compose the meaning of longer phrases. This way, the model is not as easily fooled as previous models. The

model and dataset for our project is taken from the paper [1]. Semantic word spaces have been very useful but cannot express the meaning of longer phrases in a principled way. The remedy for this is Sentiment Treebank. It includes fine grained sentiment labels for 215,154 phrases in the parse trees of 11,855 sentences and presents new challenges for sentiment compositionality. It is the only model that can accurately capture the effects of negation and its scope at various tree levels for both positive and negative phrases. The Stanford Sentiment Treebank is the first corpus with fully labeled parse trees that allows for a complete analysis of the compositional effects of sentiment in language. The corpus is based on dataset introduced by Pang and Lee (2005) and consists of 11,855 single sentences extracted from movie reviews. It was parsed with the Stanford parser (Klein and Manning, 2003) and includes a total of 215,154 unique phrases from those parse trees, each annotated by 3 human judges. This new dataset allows us to analyze the intricacies of sentiment and to capture complex linguistic phenomena.

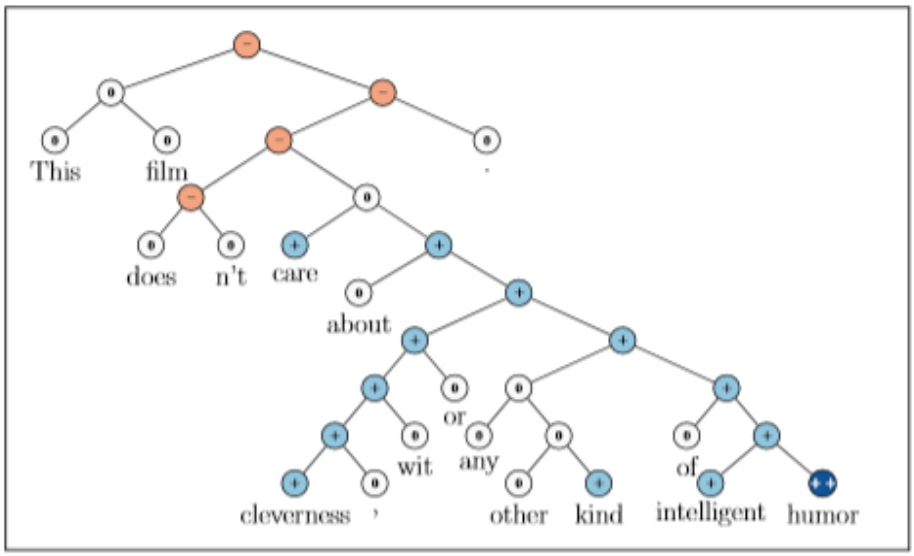


Figure 1

Figure 1 shows one of the many examples with clear compositional structure. The granularity and size of this dataset will enable the community to train compositional models that are based on supervised and structured machine learning techniques. While there are several datasets with document and chunk labels available, there is a need to better capture sentiment from short comments, such as Twitter data, which provide less overall signal per document.

- **Scribe:** Scribe is a server for aggregating log data streamed in real-time from a large number of servers. It is designed to be scalable, extensible without client-side modification, and robust to failure of the network or any specific machine.Scribe servers are arranged in a directed graph, with each server knowing only about the next server in the graph. This network topology allows for adding extra layers of fan-in as a system grows, and batching messages before sending them between data centers, without having any code that explicitly needs to understand datacenter topology, only a simple configuration. Scribe was designed to consider reliability but to not require heavyweight protocols and expansive disk usage. Scribe spools data to disk on any node to handle intermittent connectivity node failure, but doesn't sync a log file for every

message. This creates a possibility of a small amount of data loss in the event of a crash or catastrophic hardware failure.

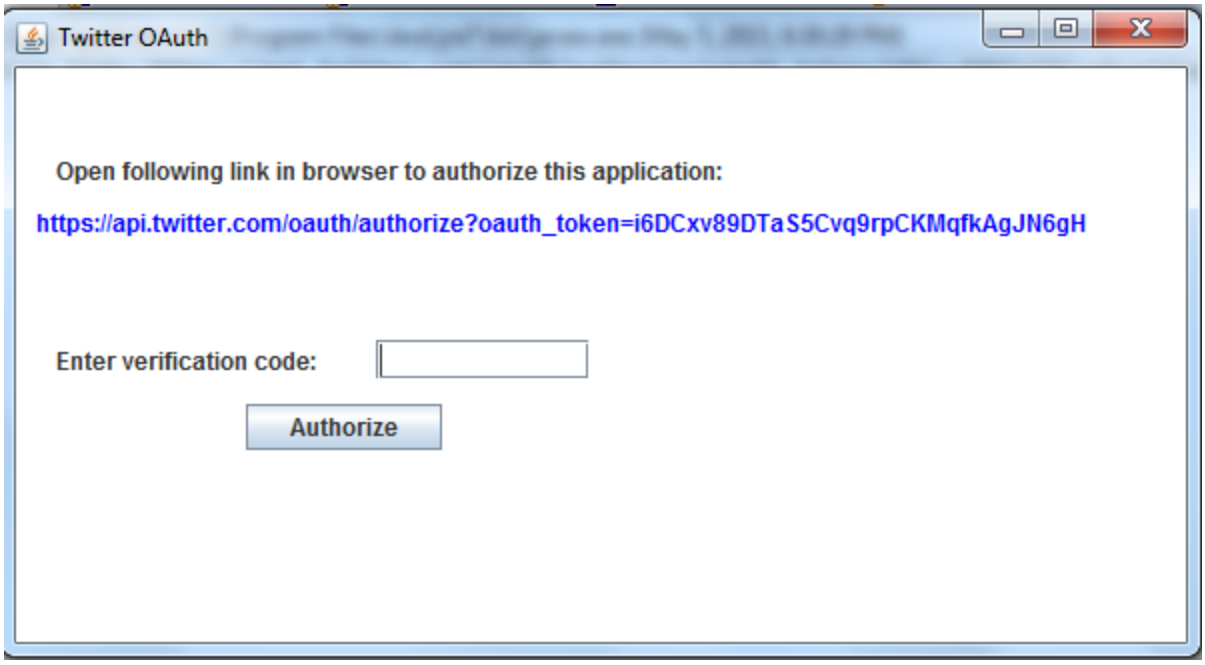
IV. Implementation and Execution

(For more information about execution please refer to readme.txt)

- **OAuth:**

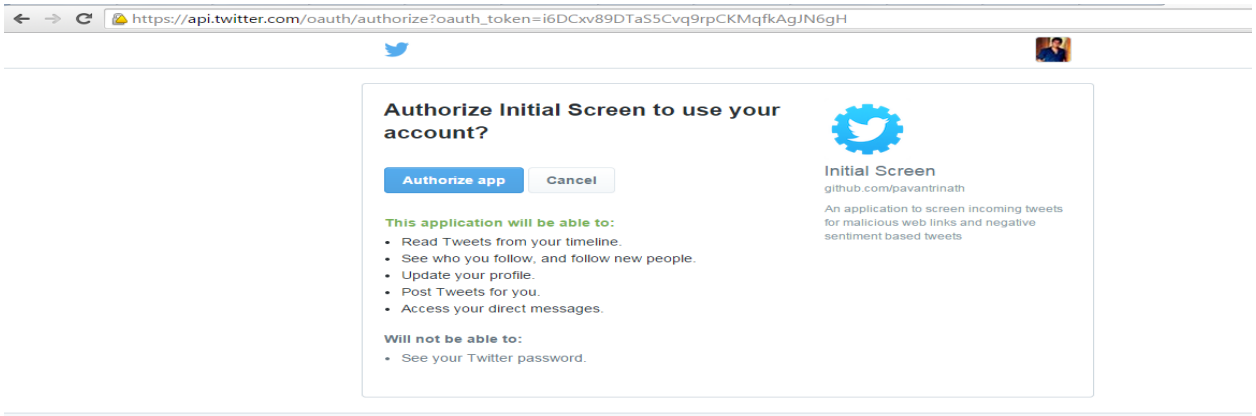
➔ Execute twitterauth.java

➔ Copy the verification link from the twitter Oauth window.

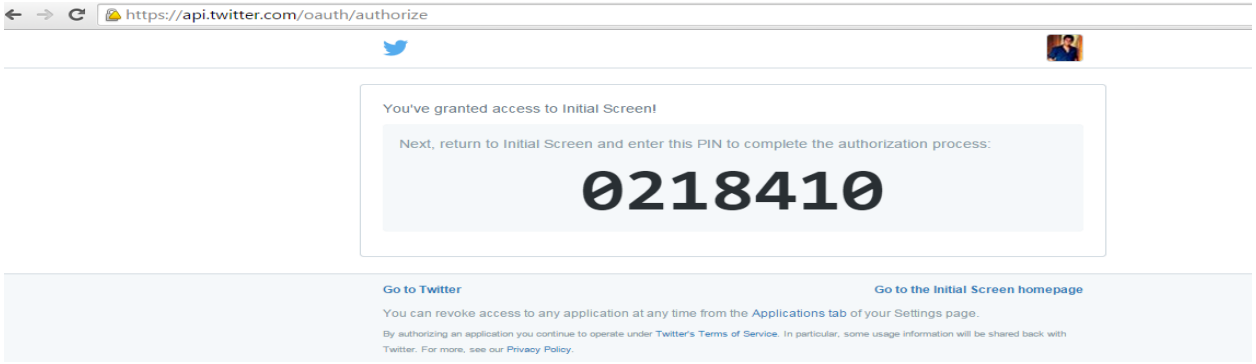


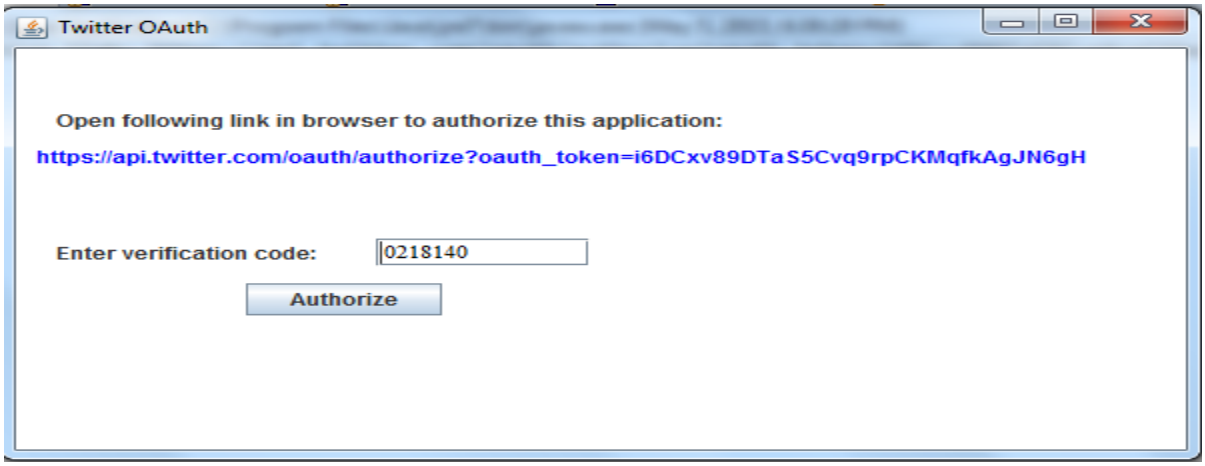
➔ Log in to your twitter account and paste the link in the browser

➔ Authorize Initial Screen App in the browser



➔ Copy the generated verification PIN and enter it in the Verification Code Field of our application

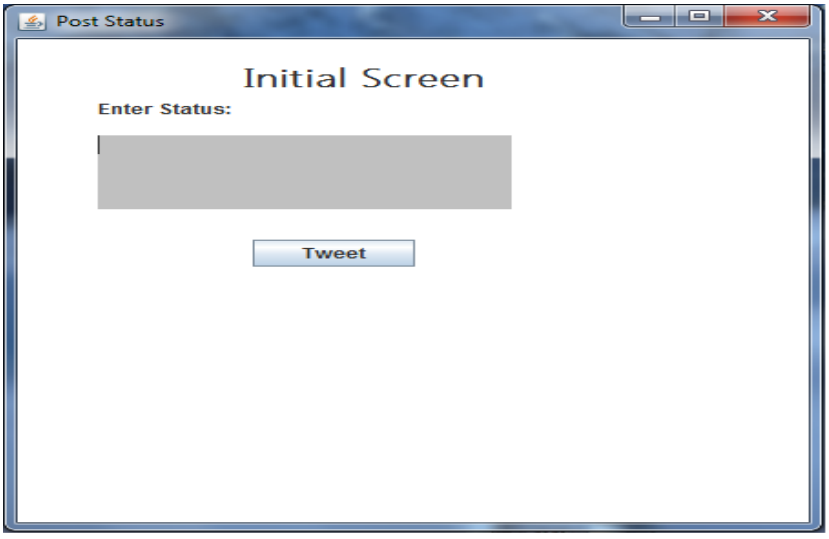




This app is as of now authorized by the user.

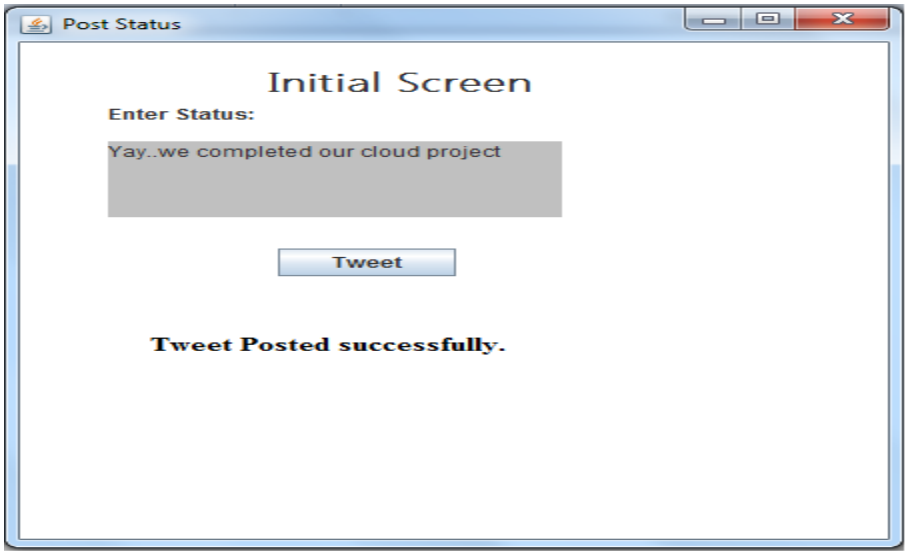
Post Status:

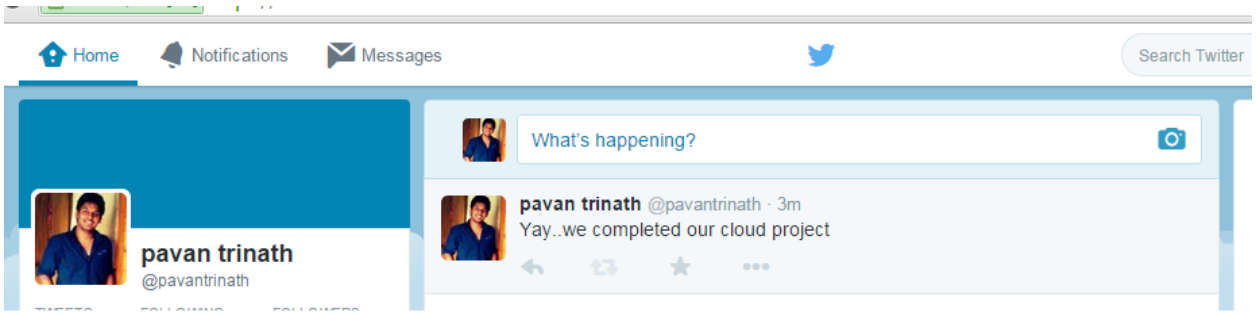
➔ On authorization the user will be navigated to the post status screen



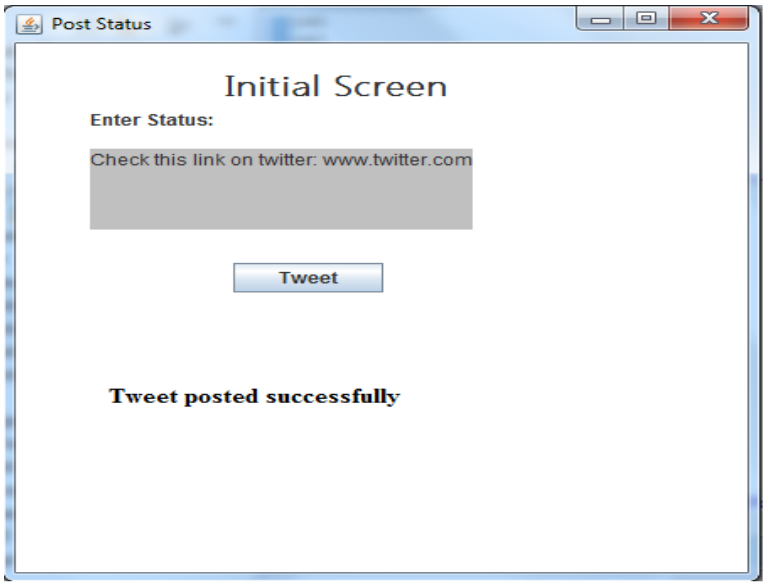
The following screenshots describes various facets of our implementation

➔ You entered a non-offensive tweet



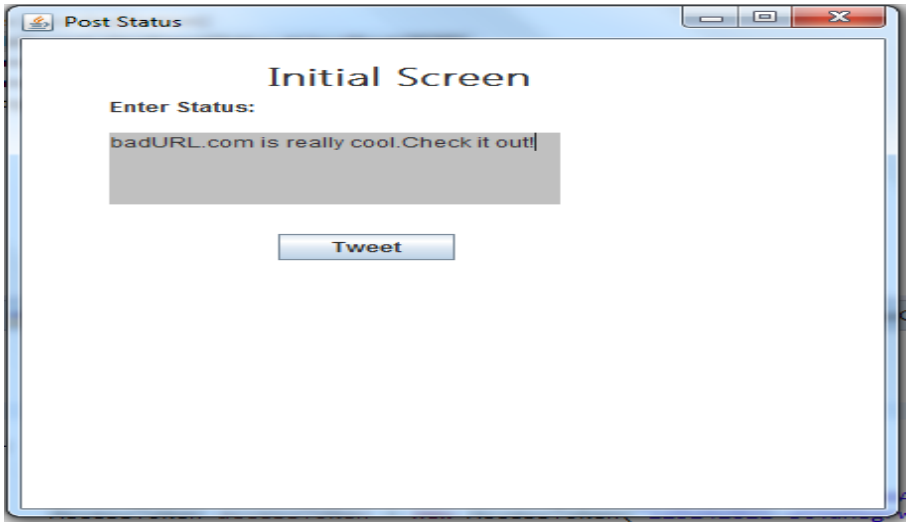


➔ Tweet is malware free and hence successfully posted

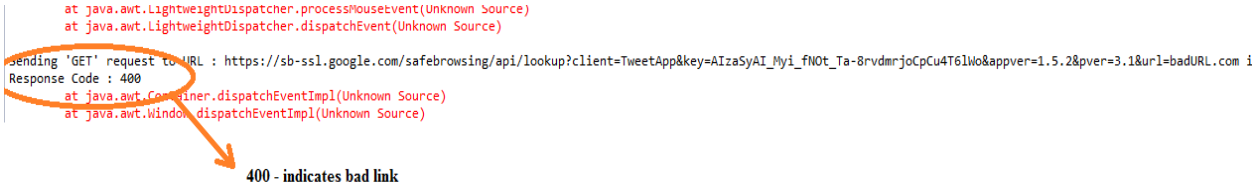


Malware affected Tweet :

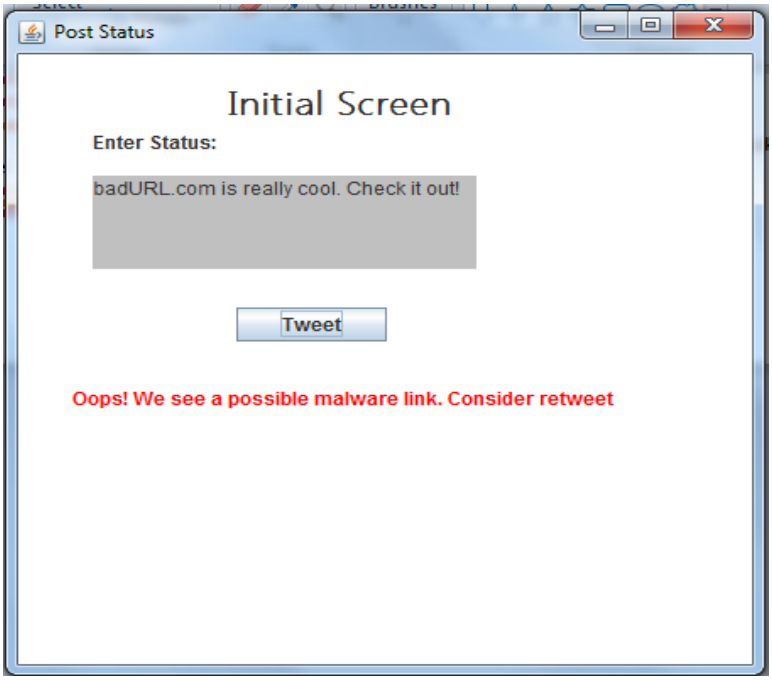
➔ A bad URL is present in the tweet.



➔ Google safe browsing lookup API detected a malware web Link, Error code - 400

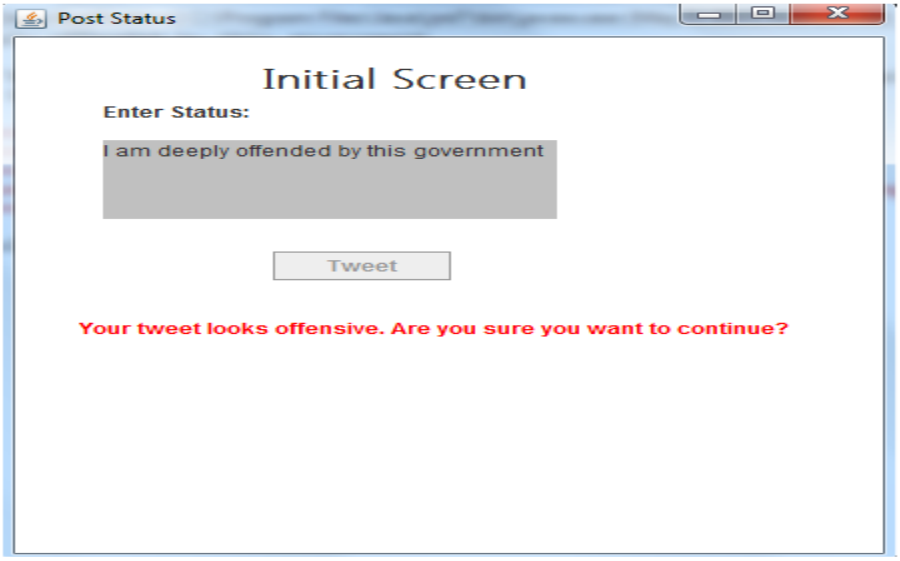


➔ User is prompted to re-tweet.



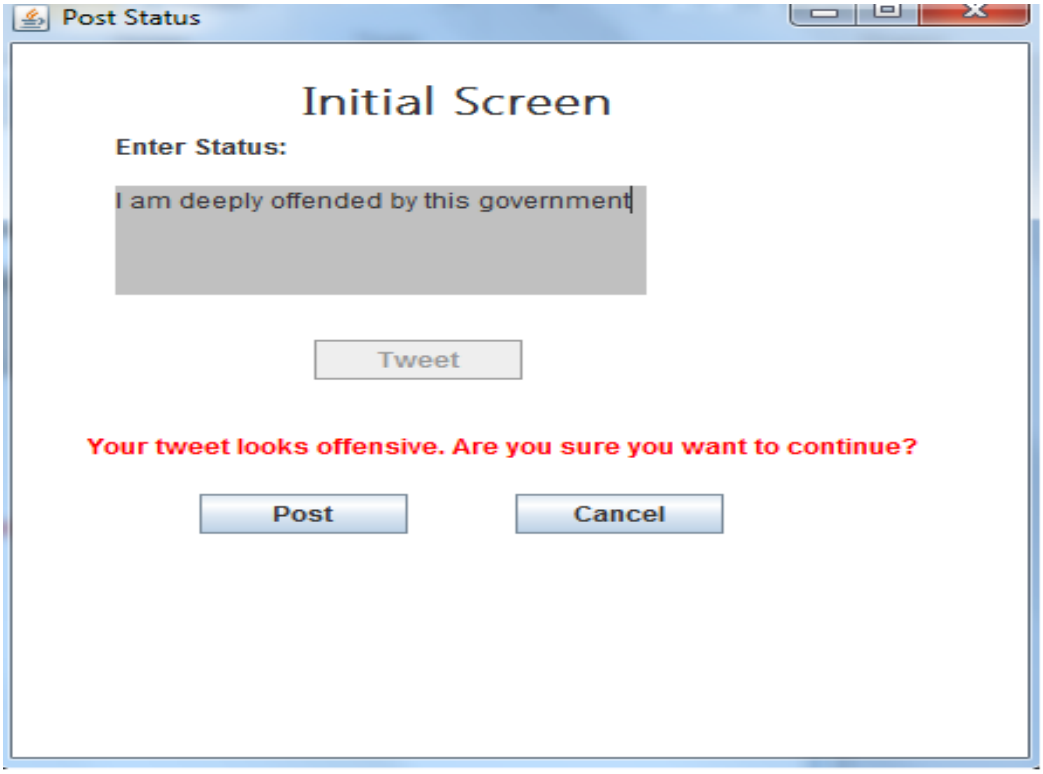
Offensive tweet :

➔ The Sentiment of the tweet is deemed negative by our sentiment analysis model.



The user is warned about the sensitivity of the tweet

➔ User can decide to either proceed with posting the tweet or he can cancel the action



V. Sample Outputs

Tweet	Malware	Sentiment
badURL.org is really cool	Yes	N/A
www.google.com is great!	No	Very Positive
I'm offended by this government	No	Very Negative
I had a bad day	No	Negative
Its a good day	No	positive

VI. System Requirements:

- Maven(Please refer readme.txt for installation instructions)
- For Ideal time complexity,Ram should be no less than 4GB
- Operating System: Windows/ubuntu
- Programming Language: Java 1.8+, Java Swing (Windows Builder plug-in)
- Download space: 300 MB

VII. Limitations/Challenges:

Our project is confined to Twitter social networking platform. If a malware link is not listed in the GSB lookup, then the detection might fail. We have worked on single user account. Complexities/overheads in integrating this application to support multiple user requests in a distributed environment is not considered.

VIII. Future Work:

This project can be extended to other social networking platforms with the ability to handle multiple client requests at a time.

IX. References:

1. http://nlp.stanford.edu/~socherr/EMNLP2013_RNTN.pdf
2. http://nlp.stanford.edu/sentiment/index.html
3. http://en.wikipedia.org/wiki/OAuth
4. http://en.wikipedia.org/wiki/Sentiment_analysis
5. http://en.wikipedia.org/wiki/Google_Safe_Browsing
6. http://en.wikipedia.org/wiki/Scribe_%28log_server%29