

Software Requirements Specification

for

News Credibility using Deep Learning

Version 1.0

**Prepared by Ajinkya Sonawane
Girish Bharambe
Ragini Betrabet
Chetan Chauhan**

21st September, 2018

Table of Contents

1. Requirement Analysis	
1.1 Purpose (Problem to be solved)	3
1.2 Audience of the system	3
1.3 Success metrics	3
1.4 Predecessor	3
1.5 References	3
2. Functional Specifications	
2.1 Scenarios	4
2.2 System Overview	4
3. External Interfaces	
3.1 User Interface	5
3.2 Hardware Interfaces & Communications Interface	6
3.3 Software Interfaces	6
3.3.1 FILE FORMAT	6
3.3.2 TWITTER API	6
3.3.3 RSS FEED	6
4. Technical Specifications	
4.1 Hardware Details	7
4.2 Operating System	7
4.3 Performance	7
4.4 Programming languages and Technologies	7
4.5 Version of various components	7

1. Requirement Analysis

1.1 Purpose (Problem to be solved)

The issue of fake news has arisen recently as a potential threat to a high quality journalism and can also create many chaotic situations like mob lynching, etc. The source of this fake news most of the times has been various social networking sites like Twitter, Facebook, Whatsapp, etc. With the incoming 2019 elections in India, there exists a threat by such fake news to make an impact on the results of the elections. We aim to dilute the spread of such fake news using our model.

1.2 Audience of the system

The system will be developed as an Android Application and will be made available free on the Google Play Store. Anyone who wishes to be aware of fake news on social media should use this application.

1.3 Success metrics

Scores provided by the system to trending posts/articles on social media should be as accurate as possible. Number of active users and their reviews for the application will also help to determine the success of the application.

1.4 Predecessor

“Fake Bananas” a system designed by Kastan Day and his team enrolled in Swarthmore College. Fake Bananas finished top 10 at HackMIT 2017 and won awards for 'best use of data' and 'best use of machine learning for the common good.

1.5 References

- [1] <http://www.fakenewschallenge.org/>
- [2] Mrowca, D., Wang, E., & Kosson, A. Stance Detection for Fake News Identification.
- [3] Pfohl, S., Triebe, O., & Legros, F. Stance Detection for the Fake News Challenge with Attention and Conditional Encoding.
- [4] Davis, R., & Proctor, C. Fake News, Real Consequences: Recruiting Neural Networks for the Fight Against Fake News.

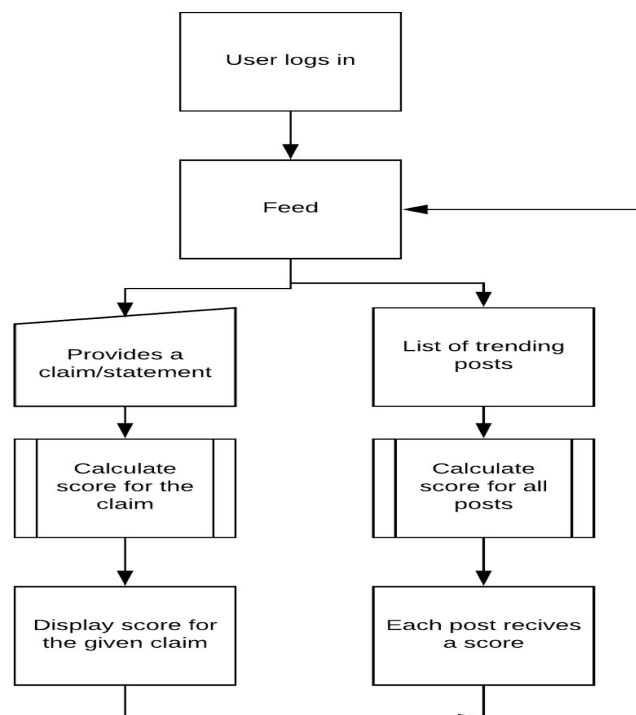
[5] Kusner, M., Sun, Y., Kolkin, N., & Weinberger, K. (2015, June). From word embeddings to document distances. In International Conference on Machine Learning (pp. 957-966).
[6] Chaudhry, A. K., Baker, D., & Thun-Hohenstein, P. Stance Detection for the Fake News Challenge: Identifying Textual Relationships with Deep Neural Nets.
[7] Bourgonje, P., Schneider, J. M., & Rehm, G. (2017). From Clickbait to Fake News Detection: An Approach based on Detecting the Stance of Headlines to Articles. In Proceedings of the 2017 EMNLP Workshop: Natural Language Processing meets Journalism (pp. 84-89).

2. Functional Specifications

2.1 Scenarios

- *User logs into the application and views trending posts on social media along with a credibility score (score of truthfulness).*
- *User can also check score for any given 'claim' or 'statement' by providing it to the application as a simple text input.*
- *Application then sends the claim to the server and our model calculates a credibility score for the 'claim'.*
- *The score is then displayed on the application.*

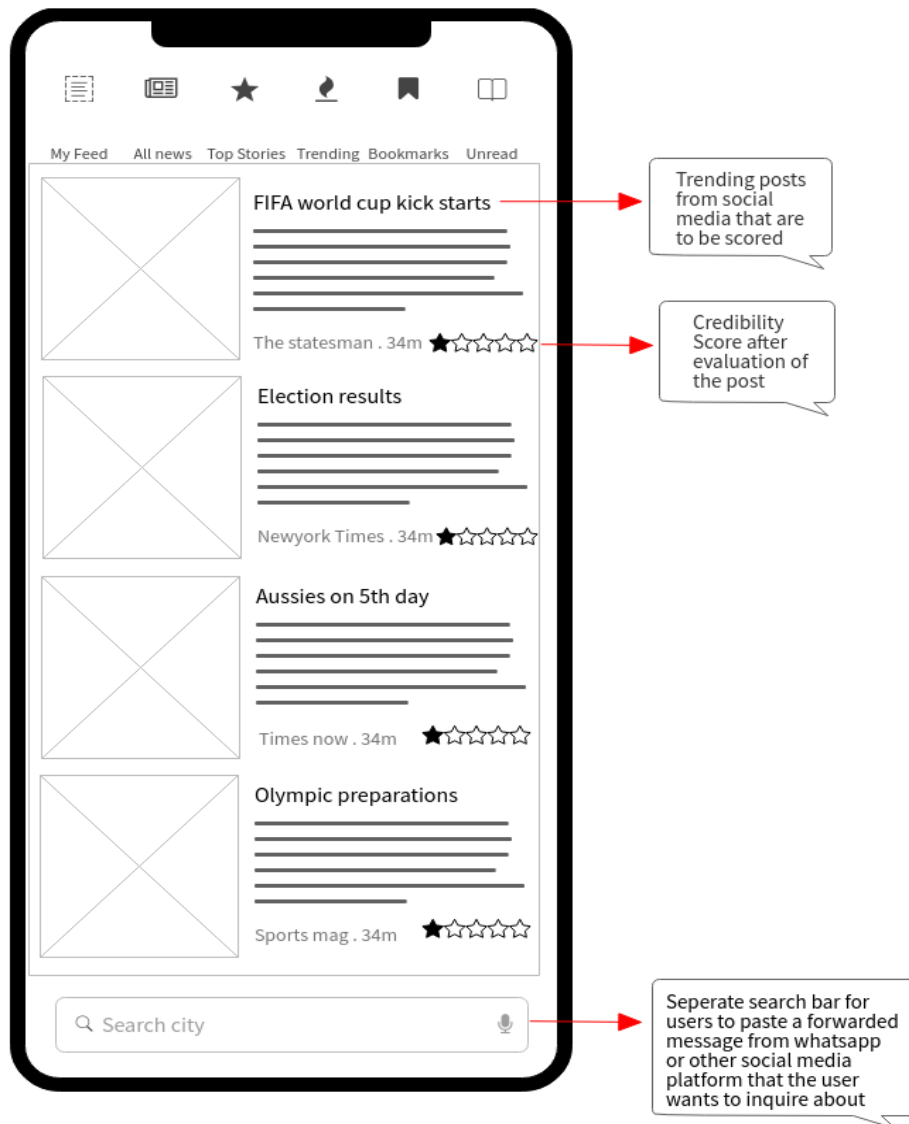
2.2 System Overview



3. External Interfaces

3.1 User Interface

We have currently not decided upon whether we will implement a web app or a mobile application but the basic UI will be as follows:



3.2 Hardware Interfaces & Communications Interface

The system consists no extra hardware interfaces or communications thereby making it portable and adheres with the programming concept of “System portability”.

3.3 Software Interfaces

System interfaces are essentially and hardware or software component that helps in the development and accomplishment of the software. As stated above, there are no hard and fast external hardware components in the development of this product, but there is a use of external software resources in the form of APIs. In computer programming, an Application Programming Interface (API) is a set of subroutine definitions, protocols, and tools for building application software. In general terms, it is a set of clearly defined methods of communication between various software components. A good API makes it easier to develop a computer program by providing all the building blocks, which are then put together by the programmer. An API may be for a web-based system, operating system, database system, and computer hardware or software library.

3.3.1 FILE FORMAT

RSS feeds can be in either XML or JSON format.

3.3.2 TWITTER API

Standard Search API to extract trending tweets.

3.3.3 RSS FEED

We'll use RSS feeds of reputed news houses to extract news articles that'll serve as evidences for our Stance Detection Model.

4. Technical Specifications

4.1 Hardware Details

We'll be using standard laptops that'll be running on **intel core i5/i7**. Android smartphones running on **ARM64** chip may also be used to test our app.

4.2 Operating Systems

Windows is the primary operating system. Building an environment to develop a mobile app for Android devices is rather easy. It only requires installation of Eclipse, Android SDK and Android emulator to initiate the development process -although more software and developer tools can be installed later during the process. Android SDK is a software development kit that enables a developer to create applications for Android platforms. Android SDK includes application development tools, sample projects with source codes and required libraries to build Android application. The Android emulator is a virtual mobile device running on the computer. The software emulates an Android device, running the Android OS, for debugging applications without needing a variety of devices and OS versions.

4.3 Performance

Our priority is to keep the system response to as minimal as possible, currently we are aiming to have a output latency of 5 seconds. We'll try to bring it down to 3 seconds if possible.

4.4 Programming languages and Technologies

Python will be our primary programming language.**JAVA** will be used to build the android application. We might also use **TensorFlow** library to implement the CNN's, LSTM etc. We'll be using **Spyder** IDE for coding.Spyder is a powerful scientific environment written in Python, for Python, and designed by and for scientists, engineers and data analysts. It offers a unique combination of the advanced editing, analysis, debugging, and profiling functionality of a comprehensive development tool with the data exploration, interactive execution, deep inspection, and beautiful visualization capabilities of a scientific package.**Github** will be used as our version control system.

4.5 Versions of various components

- Eclipse - 4.8(Photon)
- Spyder - 3.3.1
- TensorFlow - 1.7.0
- Android Studio - 3.1.4