Internet Rumour Detection

"WhoSeddit" - CS39440 Major Project Report

Christian Holland (chh57@aber.ac.uk)
Supervisor: Reyer Zwiggelaar (rrz@aber.ac.uk)
Module: CS39440
Degree Scheme: G400 (Computer Science)
Date: February 24, 2017
Revision: 2.0
Status: Draft

Contents

1. Background

1.1 Rumours on the Internet

Over the last 15 years, online social networks have become increasingly more popular within users of the Internet and are the cause of many rumours that have arisen over said time. With the amount of online social networks and the amount of people using them, it is very difficult to trace the rumours back to their source.

An important point to take note when regarding the spread of rumours on the Internet is their reach globally. Will the rumours reach everywhere on earth through means of Internet? Not everywhere has access to the Internet, however nodes holding Internet rumours may spread them via word of mouth and other physical forms. This method of spread cannot be so easily tracked and analysed such as those on the Internet.

One must therefore consider that the above diagrams only show an evaluation from one social network; Twitter, thus with so many rumours circulating from so many different platforms, it is important that a capable model be developed to categorise each sector of rumour.

There have been many studies analysing this spread of rumours, indicating their nature and showing possible predictions towards further spread. During the project, these studies will be used to correctly identify and develop the most accurate model to help analyse the collected rumours.

Section **1.2** will investigate this ideology.

1.2 Abstract

While there have been many studies investigating rumours on the Internet, there have been very little (public) projects that utilize their findings and provides a dynamic, interactive system whereby the user can have their rumour analysed there and then.

This project - namely, WhoSeddit allows the user to enter a hashtag (or string) resembling a rumour, and provides an in-depth breakdown of the first time it was posted. It then shows the spread of the rumour throughout other users; whether it was retweeted or simply posted by another user – and shows the link. It shows this breakdown in a graphical user-friendly manner.

WhoSeddit aims to show the user the nature of the rumour spread without the need for large quantities of text to describe it. The graphical breakdown will provide an analysis of the spread using the studies as the source for its model, displaying the original rumour and the similar rumours following it. As each rumour will be in the form of a ‘node’, the breakdown will also display the popularity of each ‘node’ - whereby if the Tweet has more

‘favorites’ or ‘retweets’, the ‘node’ will be bigger. Each ‘node’ will grow directly proportionate to its popularity.

The above is based on WhoSeddit following the ‘achievable baseline’ as described in section **1.3**, however should there be excess time for project expansion, the user will not only be provided with rumour analysis through Twitter, but also through Facebook/Google. The graphical breakdown will also show how the rumour spread throughout social networks, by providing a timeline showing all of the rumours in chronological order.

1.3 Achievable Baseline

With so many rumours to analyse globally on so many different platforms, it is important that suitable goals are set to ensure that the project is complete by the given deadline. If the goal was set that required WhoSeddit to analyse all rumours within Twitter, Facebook and Google by the deadline, it is highly possible that this goal would not be met. Instead, WhoSeddit will be considered ‘complete’ when Twitter has been fully incorporated into the project and all rumour analysis is fully functional with Tweets.

If the objective has been met early, WhoSeddit will begin to expand its model to investigate rumour spread within Google and Facebook too. This ensures that the Twitter aspect of the project is fully complete by the time Google/Facebook is considered, allowing the user to have a functional interface to use while the other platforms are in development.

2. Research

2.1 Social Network Rumour Studies

One of the studies; "Why Rumors Spread So Quickly in Social Networks" by B. Doer, M. Fouz and T Friedrich [1] states that if a node containing a rumour has less neighbors, the rumour will spread faster from it than a node with many neighbors.

Diagram 2.1.1 - Rumours within friend circles [1]

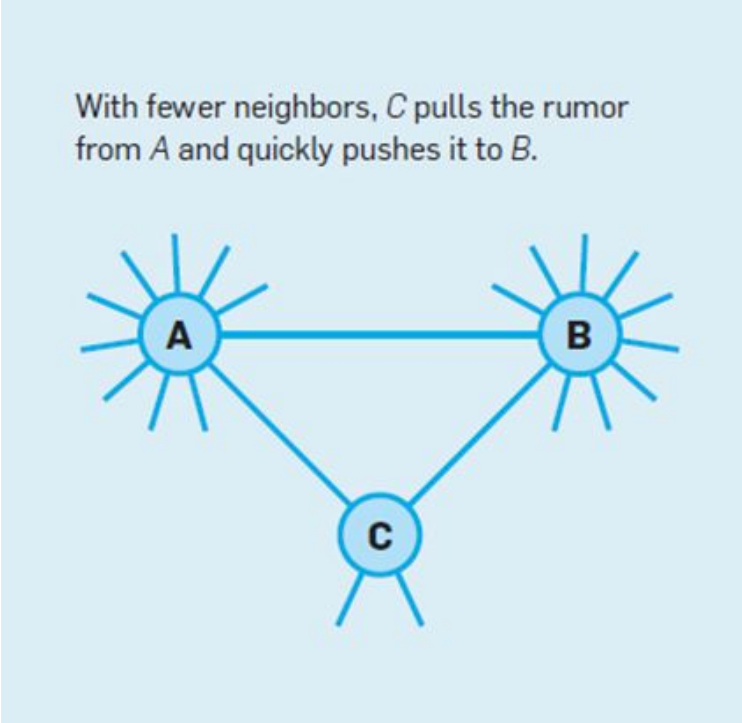


Diagram 2.1.1 shows how quickly rumours can spread when the social circle is small. The nature of the social circle signifies that C will pass on the rumour faster than A or B as there are less people to pass it on to. By the time C has passed on the rumour to its connections, A and B will have only passed the rumour to a small fraction of its connections. By this time, C's connections would have passed on the rumour to their connections.

The study also shows the nature of informed nodes over a given period of time, depending on the platform and its capabilities in regards to discovering the actions of others:

Diagram 2.1.2 - Rumour spread within Twitter [1]

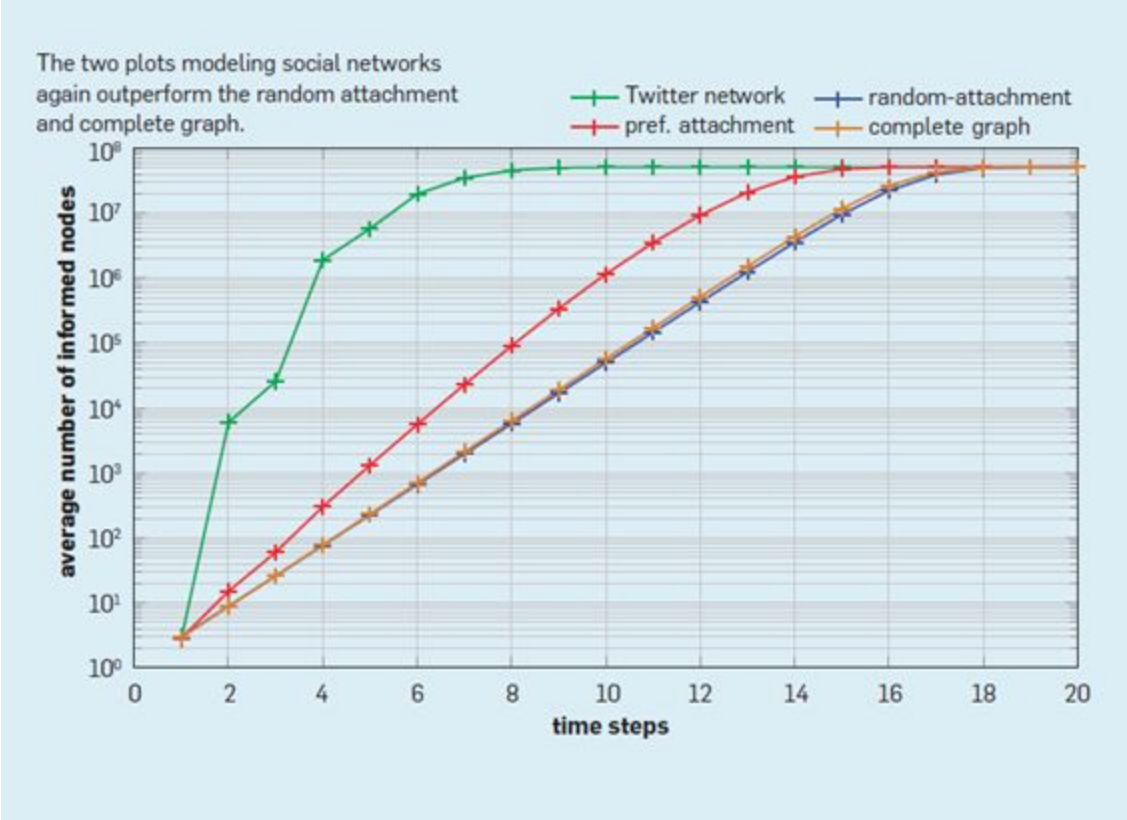


Diagram 2.1.2 shows the amount of informed nodes of a rumour on Twitter. The amount of informed nodes goes up quickly, due to the Discover section whereby the user is able to see rising and "hot" hashtags on the platform. As more people see the hashtag, more people are inclined to use it, resulting in the sharp rise within a small amount of time. The diagram also shows the preferred attachment; whereby the user consciously shared the rumour from the person they found it from (e.g retweets or through the Search feature). The random attachment is shown too; whereby the rumour is coincidently shared between social circles.

This study is very relevant as the base project will be using Twitter to find and analyse rumours; as elaborated in section **1.3**.

Diagram 2.1.3 - Global rumour spread within Twitter [1]

Diagram 2.1.3 shows how news and rumours can be spread globally so easily. The diagram displays how Japan alone has a huge foothold on global communication; the diagram is only showing incoming and outgoing Tweets from a single hour period, in March 2011.

With the diagram only showing the large spread of one region, one can imagine the analysis that must take place in order to track a single rumour within the millions of other Tweets and rumours from every region globally with access to the Internet.

2.2 Similar Projects

2.3 Adapting Existing Rumour Detection

Most online social networking sites have the 'Discover' feature or some variation, whereby the user is able to search for a phrase, hashtag or word. The user is then provided with Tweets, Statuses or more (depending on the platform) descending chronologically. This project will differ as it offers a more visual approach, showing the recent popularity and its connections across the platform on a much wider scale instead of the most recent match.

The current rumour detection within social networking websites does not defer from its own platform. Thus if the user does enter their proposed rumour on Twitter, it will not include Facebook posts in the result (unless the user has provided a link to a Facebook post within a Tweet). WhoSeddit aims to provide a breakdown within Twitter, eventually

2.4 Rumour Verification

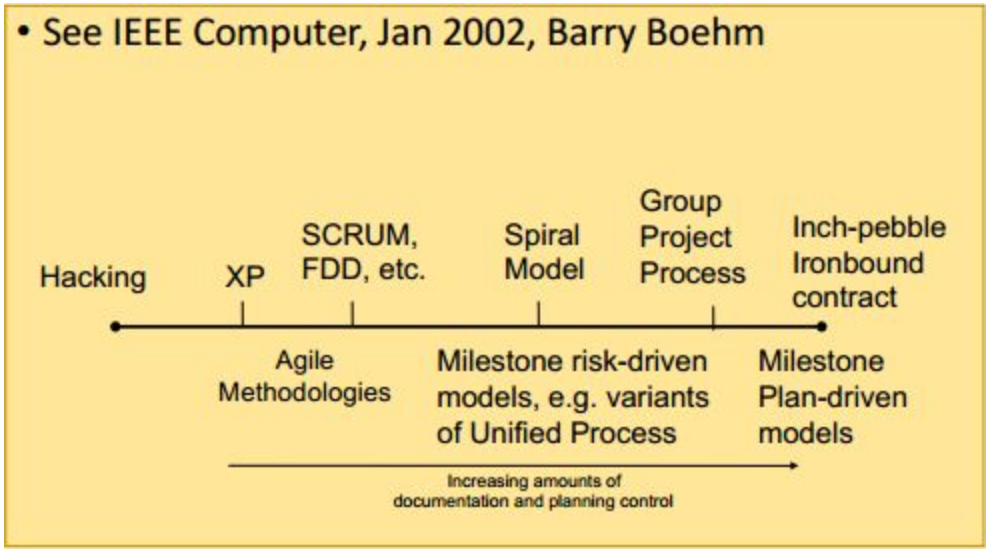
2.5 Platform As A Service

3. Planning

3.1 Methodology

3.1.1 Methodologies Considered

Diagram 3.1.1.1 The Planning Spectrum [2]



Agile or Plan Driven

I used previous university project experience to decide which methodology to use during the development of WhoSeddit. I found that I am more productive if I spend less time planning and more time developing and refactoring. As displayed in diagram 3.1.1.1, this would signify that I should use a more agile approach towards development, however there are many specific agile methodologies to choose from. Not only this, but in the professional industry, many large technical companies use a more agile approach [3], thus gaining experience with this would be beneficial professionally.

3.1.2 Methodology Chosen

Feature Driven Development (FDD)

'Hacking' is a methodology displayed at the far left side of diagram 3.1.1.1 This involves (mostly) no planning or documentation, which is not ideal for large-scale projects such as this. It is a 'code-as-you-go' approach that could result in more time refactoring code than developing, as features will be coded on the go and later found to interfere with other features.

I plan to use a mix of XP and FDD during the project, as both approaches are dynamic and work well with a change in requirements or situations, which are common in professional large-scale projects. While it does not specifically apply to this project, FDD doesn't require on on-site customer presence throughout the project and therefore is a popular approach to project development. For this reason, it is valuable to gain experience with FDD, even though it will only be myself involved in production.

As this project will be, in short, a website with many features, the FDD approach will suit this perfectly. When WhoSeddit branches out to other social networking websites, this change in requirements will not negatively affect any functionality as the project is based on an agile methodology. FDD in this project will differ from large scale projects as roles will not be delegated to many people - only myself. This is both a challenge and a benefit as I will have more of a workload, but I will know exactly what point each feature of the project is at. The normal roles for FDD are:

- Domain Expert
- Project Manager
- Chief Architect
- Development Manager
- Chief Programmer
- Class Owner
- Various supporting roles

It is important I identify the processes required for each role within this project, to ensure FDD is approached efficiently and effectively.

Diagram 3.1.2.1 The Five FDD Processes [4]

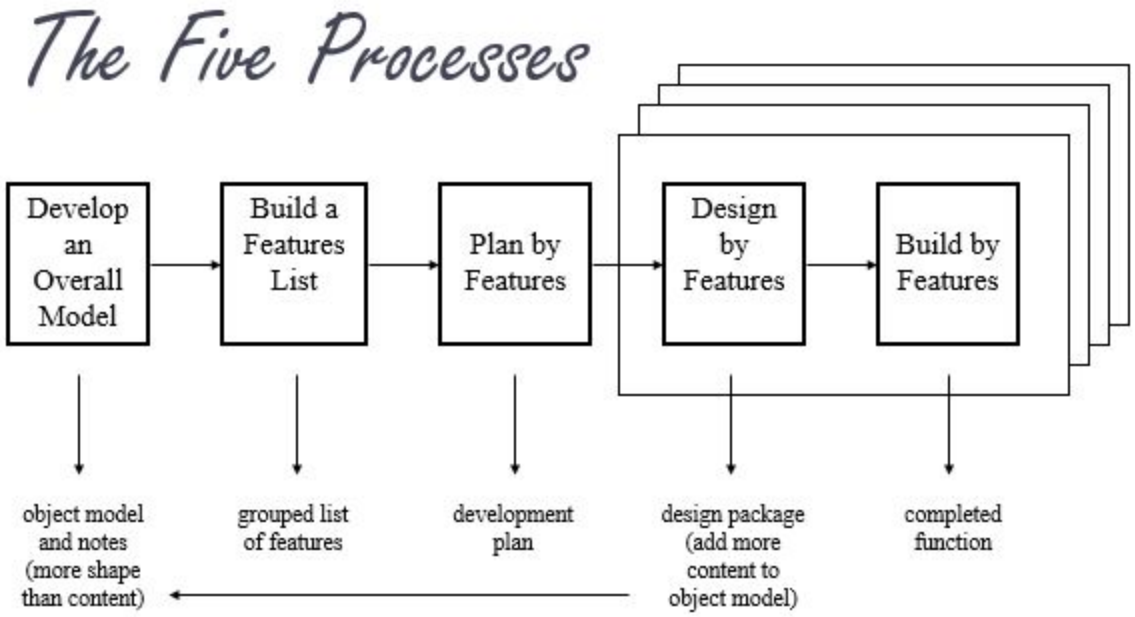


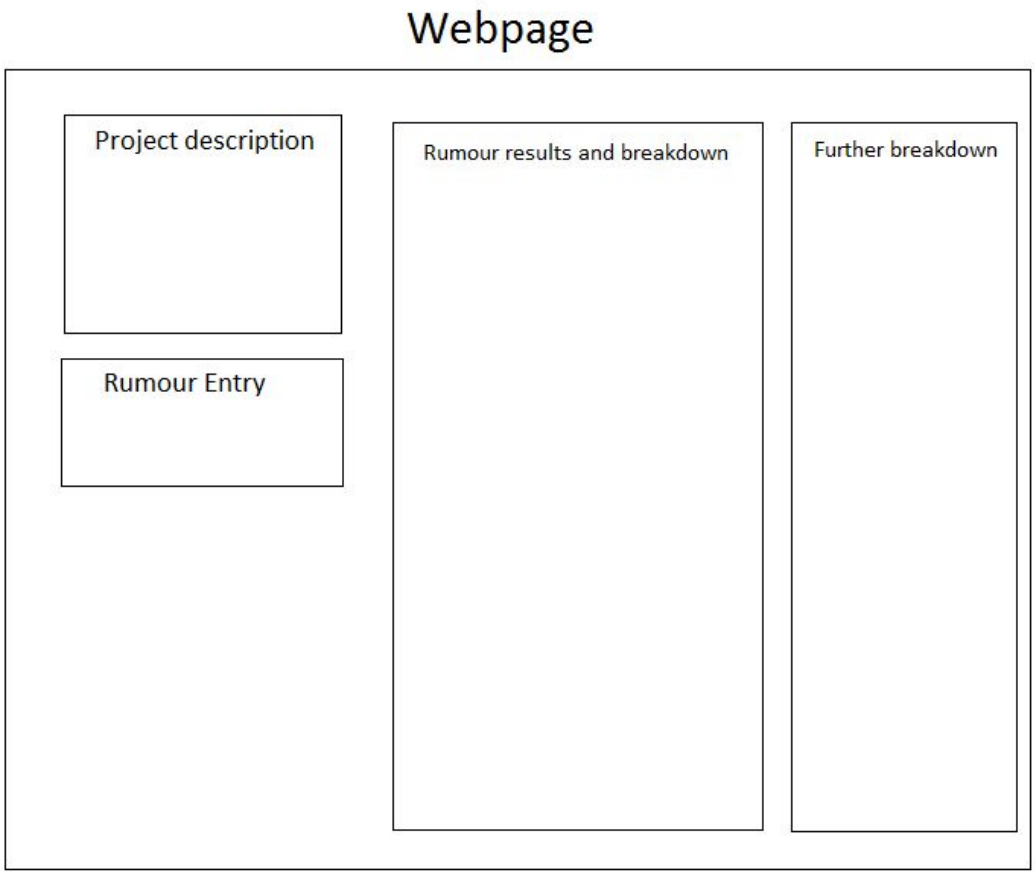
Diagram 3.1.2.1 shows the five processes required for FDD. It is clear to see here that the methodology is almost solely based on developing a feature at a time. This approach is beneficial as it ensures there are not many incomplete features within the project, which could cause issues.

This workflow will be performed throughout the development of WhoSeddit.

3.2 Overall Model

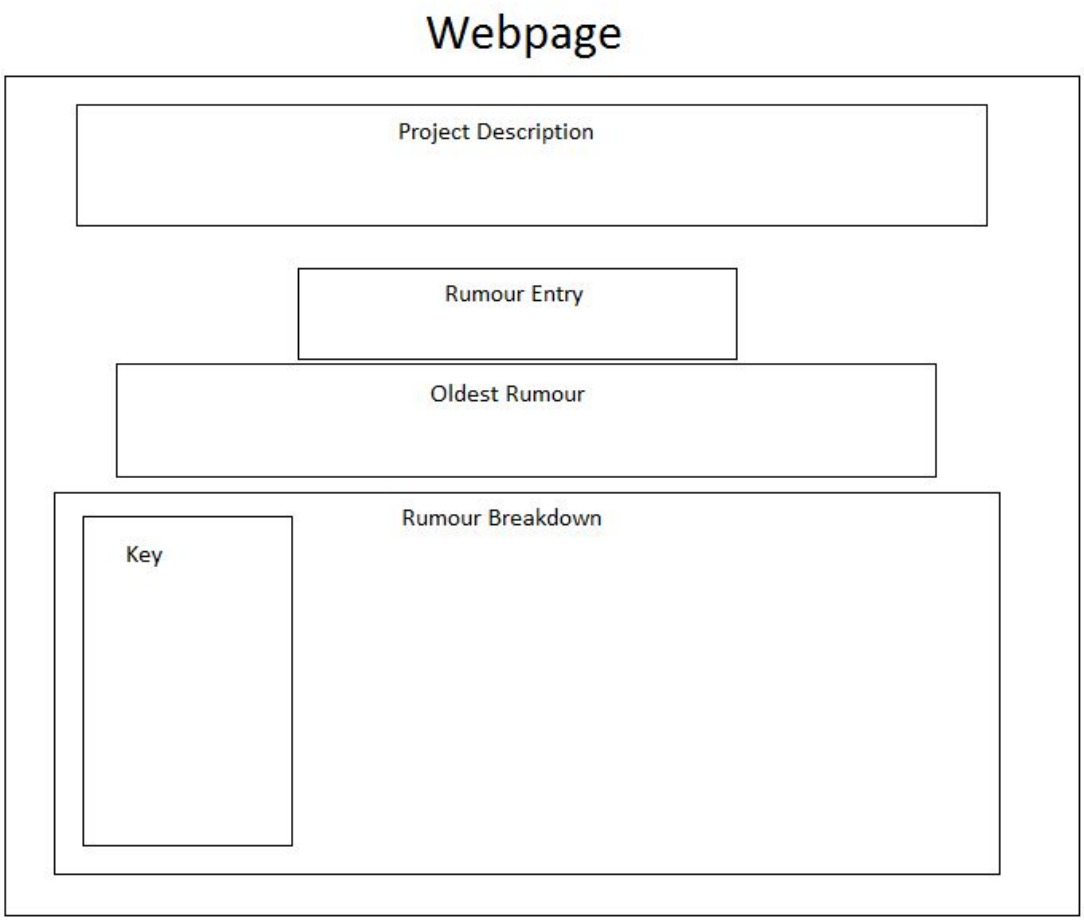
3.2.1 Model Designs

Diagram 3.2.1.1 - Design #1



This was the first model design I had, before I had my final decision on representation of rumours. After I had established that my data representation would be in a circular format (refer to section **3.4**), it was clear that Design #1 would not be effective as the box was too narrow for a full display of the circular network. This design also does not consider displaying the oldest rumour that WhoSeddit can gather - which should be a centre point for all of the data returned (as further rumours displayed would, in theory, stem from this rumour).

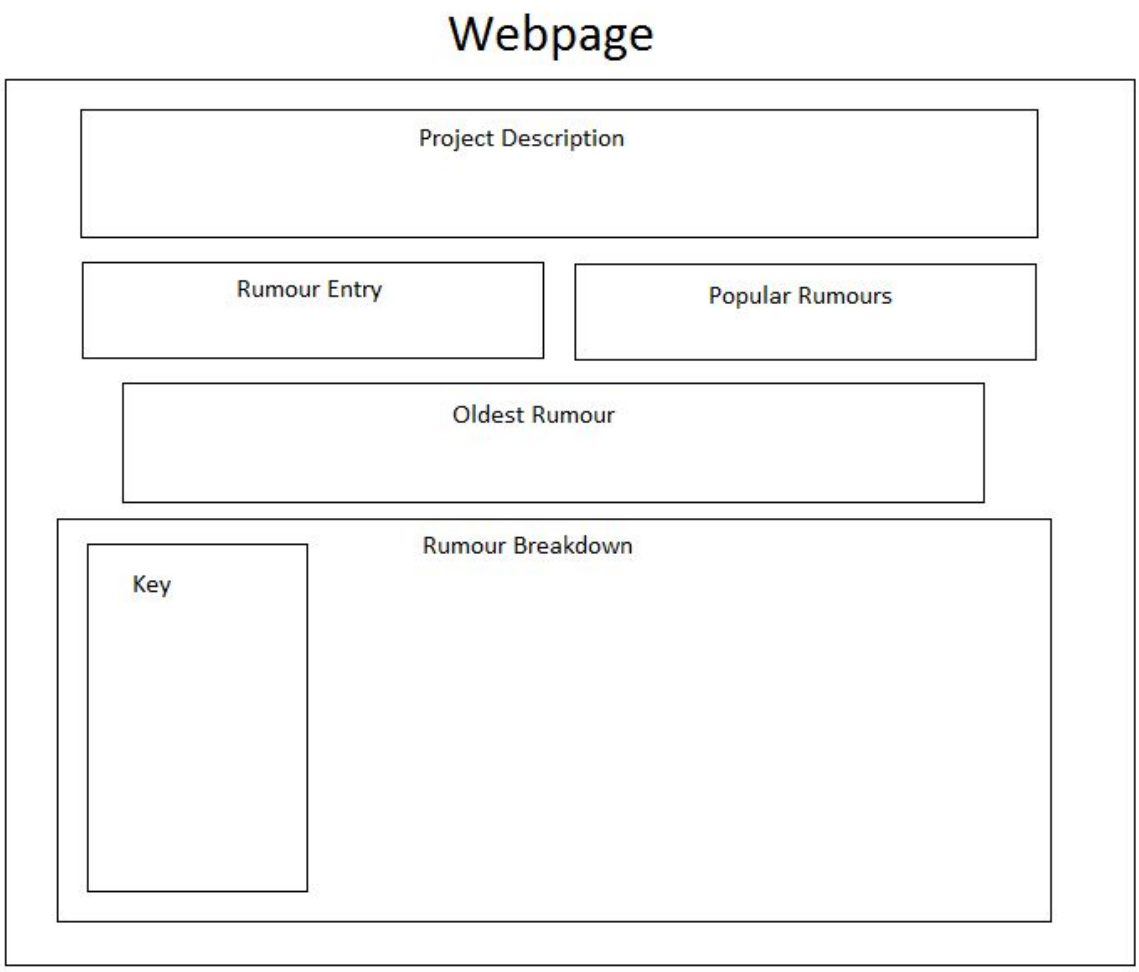
Diagram 3.2.1.2 - Design #2



Design #2 accounts for the layout of the rumour network and its circular nature. As it was established that this network would have different types of nodes representing each rumour, a key is required. This layout is also more mobile friendly, due to its more portrait nature. This design was created before it was established that a ‘popular rumours’ feature would be added.

3.2.2 Final Model Design

Diagram 3.2.2.1 - Final Design



The final design ensures that the page is both mobile friendly and informative. Many websites will put too much data on the page, overwhelming the user, however this design allows the user to absorb each bit of information before presented with the next. The “Oldest rumour” and “Rumour Breakdown” sections are not visible until the user has entered their desired rumour.

3.3 Gathering Data

3.3.1 Twitter API

The first choice I had when considering retrieving the rumours from Twitter was to use their official API.

Advantages

- Free to use. All the developer needs is a Twitter account to tie their credentials to each request to the API.
- Abundance of online support.
- Fast response to each request, as Twitter fund their API extensively to ensure its support and functionality

Disadvantages

- 6-9 day limit on historical Tweets [5]. This means that each request will only provide results for up to 6-9 days from the request time. This is the largest disadvantage and must be reviewed to establish whether this will prevent the full project potential.
- 100 Tweet limit for each day. The API allows the user to specify which day they wish to retrieve Tweets for (up to 6-9 days ago) but does not allow any more than 100 Tweets for each given day. While this is a disadvantage, it does not affect the project as much as the 6-9 day cap, as a sample of about 100 Tweets per day is all that is required for efficient rumour tracking.

3.3.2 Third Party API's

In terms of alternate solutions to retrieve the Tweets, the only viable solution that offers a full archive of historical Tweets is GNIP [6]. This is an API used most commonly by large businesses for data mining.

Advantages

- Offers access to Tweets of any age.
- Offers powerful and fast ways to filter Tweets
- Offers "Real-time Trend Detection" and Discovery

Disadvantages

- After contacting them, they confirmed that the cost of usage would be $1100 a month. They would not allow me to use it free of charge. This is capital that I do not own for this project.

3.3.3 Final Choice

Based on this, while GNIP offers almost exactly what is needed for this project to thrive a meet its full potential, I do not have the funds for its usage and therefore have to use the Twitter API. The 6-9 day cap has been carefully considered. While older rumours will not be included in the rumour analysis, many rumours that people wish to trace are in fact more recent, therefore this limit is not preventing the rumour analysis and the Twitter API is still viable.

3.4 Presenting Data

3.4.1 Optimal Method of Presentation

There are many ways that the Tweets can be represented to the user in order to show the spread of their popularity. The first idea considered was in a Gantt Chart format, whereby the first Tweet retrieved is at the start, and the spread is shown as time goes on. While this can be an efficient way of presenting the data, it doesn't capture the user's eye and does not look impressive on larger scale when there are hundreds of Tweets. Very quickly, the chart would need to clump the Tweets together to fit them all in.

This lead me on to the 'network' concept. In this concept, the representation is not linear, but circular; thus hundreds of Tweets can fit inside the circle with room to even show the usernames respectively. This method

3.4.1 Representation Concepts

3.4.1 Technical Options for Representation

As the representation would be done client side, it must be in Javascript. I could code this manually, however there is an abundance of Javascript libraries for graph representation online that can be used free of charge. Some examples I considered:

- Chart.js [7]. Impressive aesthetics, however lacks the ability to create large connections between many nodes (which is what is required to represent when users have retweeted a Tweet).

- Plotly.js [8]. This library is aesthetically pleasing and is ideal for network representation, but it's method of implementation can be very complicated- with the code requiring large functions for small features.
- Sigma.js [9]. Simple library for network representation, however can seem a bit messy with larger networks.

3.4.2 Final Choice

I have chosen Vis.js as the Javascript library to represent the network.

3.5 Technical Limitations

3.5.1 6-9 Day Limit

3.5.2 100 Tweet Daily Limit

3.5.3 Client-Side Processing

3.6 Plan Evolution

4. Development

4.1 Local Experimentation

4.2 Front End

4.3 Back End

4.4 Feature Addition

5. Testing

5.1 Tests Considered

5.2 Test Creation

5.3 Results

6. Analysis

6.1 Requirements Met

6.2 Issues Encountered

7. Conclusion

8. Bibliography

[1] Benjamin Doer, Mahmoud Fouz, Tobias Friedrich - “Why Rumours Spread So Quickly in Social Networks”, https://cacm.acm.org/magazines/2012/6/149793-why-rumors-spread-so-quickly-in-social-networks/abstract , accessed March 2017

[2] Neil Taylor - “The Planning Spectrum”, https://blackboard.aber.ac.uk/bbcswebdav/pid-753728-dt-content-rid-1109749_1/courses/CS31310_AB0_2016-17/CS31310_L2_The_Agile_Landscape.pdf , accessed March 2017

[3] Marcia Riefer Johnston “Agile Content Development: Five companies tell how they do it”, http://contentmarketinginstitute.com/2015/04/agile-content-development-icc-panel/ , accessed March 2017

[4] Edel Sherratt “Feature-Driven Development”, https://blackboard.aber.ac.uk/bbcswebdav/pid-753785-dt-content-rid-1109723_1/xid-1109723_1 , accessed March 2017

[5] Twitter - “Twitter Developer Documentation”, https://dev.twitter.com/rest/public/search , accessed March 2017

[6] GNIP - “Enterprise Access to Twitter Data”, https://gnip.com/sources/twitter/ , accessed March 2017

[7] Chart.js “Chart.js”, http://www.chartjs.org/ , accessed March 2017

[8] Plotly.js “plotly.js” https://plot.ly/javascript/ , accessed March 2017

[9] Sigma.js “sigma.js” http://sigmajs.org/ , accessed March 2017