

# Project Report.

## Twitter Topic Search and Influence Reporting.

Okpala Ikenna N. Jr.  
*u0959353*

5<sup>th</sup> November 2010

**Supervisor:** Nigel Kermode

*IN PARTIAL FULFILMENT OF THE REQUIREMENT FOR THE AWARD  
OF BACHELOR OF SCIENCE IN BUSINESS INFORMATION SYSTEMS FROM  
THE UNIVERSITY OF EAST LONDON*

# Abstract

This report gives details of a project to create a web-enabled twitter topic search and Influence reporting tool tasked with providing public relations or marketing officers with a usable web application in order to help increase their external communications with customers.

The report initially covers the research and analysis involved in developing a system to meet these aims, followed by the process of designing and implementing the system.

It concludes with an assessment of the success of the system in achieving the aims of the project, with suggestions for further development provided.

# Acknowledgements

I would like to thank my project supervisor, Nigel Kermode for his invaluable advice and support over the duration of this project.

In addition, I would also like to thank specially, [Maciej Matyjas](#) and [Richard Dallaway](#) for their technical advise and support.

Also want to thank the guys on stackoverflow.com, the Scala and Lift mailing list for their time and valuable feedback in helping to develop the system.

I also want to thank my wife ‘Blessing’ for her supportive role during this time, I am eternally grateful.

Most of all I would like to thank my parents for providing me with the opportunity to study at university and all of the selfless support given over this period, for which I am eternally grateful.

## Table of Contents

<b>Abstract.....</b>	<b>I</b>
<b>Acknowledgements .....</b>	<b>II</b>
<b>Table of Figures .....</b>	<b>VI</b>
<b>List of Acronyms .....</b>	<b>VII</b>
<b>Chapter One - Introduction .....</b>	<b>1</b>
1.1 An Introduction to the Problem.....	1
<i>1.1.1 The Problem Domain Twitter.....</i>	<i>1</i>
<i>1.1.2 Problem Definition / Rationale.....</i>	<i>1</i>
1.2 Project Aim.....	2
1.3 Project Objectives.....	2
1.4 Deliverables .....	2
1.5 Relevance to Degree.....	2
1.6 Project Schedule (See Appendix) .....	3
1.7 User Feedback.....	3
<i>1.7.1 Consultant 1 Emeka Osuji .....</i>	<i>3</i>
<i>1.7.2 Consultant 2 Jane Uluoma .....</i>	<i>3</i>
1.8 Changes from the Mid-Project Report.....	4
<i>1.8.1 Project Title, Aims and Objectives .....</i>	<i>4</i>
<i>1.8.2 Project Schedule .....</i>	<i>4</i>
1.9 Evaluation Criteria.....	5
1.10 Summary .....	5
<b>Chapter Two –Literature Review .....</b>	<b>6</b>
2.1 Introduction.....	6
2.2 Web 2.0 .....	7
2.3 Social Media and Interaction.....	8
2.4 Social Media Analysis.....	9
<b>Chapter Three - Research Methodology .....</b>	<b>11</b>
3.1 Introduction.....	11
3.2 Methodologies .....	11
<i>3.2.1 Agile Software Development.....</i>	<i>11</i>
<i>3.2.2 Extreme Programming (XP) .....</i>	<i>12</i>
<i>3.2.3 Iterative and Incremental Development .....</i>	<i>12</i>
<i>3.2.4 Iteration Length .....</i>	<i>12</i>
<i>3.2.5 Conclusion .....</i>	<i>12</i>
3.3 Implementation.....	13
<i>3.3.1 Development Technology Requirements .....</i>	<i>13</i>
<i>3.3.2 Server-Side Languages.....</i>	<i>14</i>
<i>3.3.3 Development Tools and Methods .....</i>	<i>17</i>
<i>3.3.4 Web Frameworks.....</i>	<i>18</i>
<i>3.3.5 Web Services .....</i>	<i>21</i>
<i>3.3.5 OAuth .....</i>	<i>22</i>
<i>3.3.5 Other Technologies.....</i>	<i>23</i>
<i>3.3.6 Usability and Standards .....</i>	<i>23</i>
<i>3.3.7 CONCLUSION.....</i>	<i>23</i>
3.4 TESTING .....	24
3.5 SUMMARY .....	24
<b>Chapter Four - Analysis .....</b>	<b>25</b>

4.1 Introduction.....	25
4.1 Social Influence .....	25
4.2 Measuring Social Influence (the klout way) .....	26
4.2.1 <i>KLOUT SCORE</i> .....	26
4.2.2 <i>TRUE REACH</i> .....	26
4.2.3 <i>AMPLIFICATION PROBABILITY</i> .....	26
4.2.4 <i>NETWORK INFLUENCE</i> .....	27
4.3 Requirements .....	27
4.3.1 <i>MUST HAVE</i> .....	27
4.3.2 <i>SHOULD HAVE</i> .....	27
4.3.3 <i>COULD HAVE</i> .....	28
4.3.3 <i>WON'T HAVE</i> .....	29
4.4 Non-Functional Requirements .....	29
4.4.1 <i>USABILITY</i> .....	29
4.4.2 <i>COMPATIBILITY</i> .....	29
4.5 Summary .....	29
<b>Chapter Five - Design .....</b>	<b>30</b>
5.1 Introduction.....	30
5.2 Structure of the Web Application.....	30
5.2.1 <i>FRONT-END STRUCTURE</i> .....	30
5.3 Presentation .....	31
5.2.1 <i>FRONT-END MOCK-UP</i> .....	31
5.4 System Design .....	33
5.4.1 <i>USE DIAGRAM</i> .....	33
5.4.2 <i>LAYERS OVERVIEW</i> .....	34
5.4.3 <i>CLASS DIAGRAM</i> .....	36
5.4.4 <i>SEQUENCE DIAGRAM</i> .....	37
5.4.5 <i>DEPLOYMENT DIAGRAM</i> .....	40
5.5 Iteration Planning .....	40
5.5.1 <i>Iteration One</i> .....	41
5.5.2 <i>Iteration Two</i> .....	41
5.5.2 <i>Iteration Three</i> .....	42
5.6 Summary .....	42
<b>Chapter Six - Implementation .....</b>	<b>43</b>
6.1 Introduction.....	43
6.2 Iteration One (Influencer search by topic).....	43
6.2.1 <i>Deviation from Iteration Plan</i> .....	45
6.3 Iteration Two (twitter user Influencer report page) .....	45
6.3.1 <i>Deviation from Iteration Plan</i> .....	47
6.3.2 <i>User Feedback</i> .....	47
6.4 Iteration Three (OAuth) .....	48
6.4.1 <i>Deviation from Iteration Plan</i> .....	49
6.5 Summary .....	49
<b>Chapter Seven - Evaluation .....</b>	<b>50</b>
7.1 Introduction.....	50
7.2 Features Implemented .....	50
7.2.1 <i>Influencer Search by Topic</i> .....	50
7.2.2 <i>Influencer Report</i> .....	50
7.2.3 <i>Open Authorization (OAuth)</i> .....	51
7.2 Conclusion.....	51
7.3 Non-Functional Requirements .....	51
7.3.1 <i>Usability</i> .....	51

7.3.2 Compatibility .....	52
7.3.2.1 XHTML and CSS validation .....	52
7.4 Development Methodology, Time Management and Planning.....	53
7.7 Conclusion .....	54
7.8 Future Development.....	54
7.9 Summary.....	54
<b>Reference:</b> .....	<b>55</b>
<b>Bibliography:</b> .....	<b>56</b>
<b>Appendix A:</b> .....	<b>A</b>
Error Log.....	A
<b>Appendix B</b> .....	<b>C</b>
Progress Log (Personal Reflection) .....	C
<b>Appendix C</b> .....	<b>D</b>
Code .....	D

# Table of Figures

Figure 1 Google Trends Search Result for Scala based Web Frameworks.	19
Figure 2 View First Approach.	19
Figure 3 Home Page.....	31
Figure 4 Search Results Page .....	32
Figure 5 Influence Report Page.....	32
Figure 6 Use Case Diagrams.....	33
Figure 7 Layers Overview.....	34
Figure 8 CLASS DIAGRAM.....	36
Figure 9 Sequence Diagram: Search for influencers by topic.....	37
Figure 10 Sequence Diagram: View Influence Profile .....	38
Figure 11 Sequence Diagram: Open Authorization .....	39
Figure 12 DEPLOYMENT DIAGRAM .....	40
Figure 13 Front-End Page .....	44
Figure 14 Influencers Search Results.....	44
Figure 15 Influence Summary Reports .....	46
Figure 16 Visual Summary Reports .....	46
Figure 17 Tweets For the Twitter User .....	47
Figure 18 Before Twitter button is clicked .....	48
Figure 19 After Twitter button has been clicked.....	48
Figure 20 W3C XHTML TEST .....	52

# List of Acronyms

AJAX – Asynchronous JavaScript and XML

API – Application Programming Interface

CRM – customer relationship management

HTTP – Hypertext Transfer Protocol

IDE – Integrated Development Environment

JSF – Java Server Faces

JSON – JavaScript Object Notation

MVC – Model-View-Controller

MVP – Model-View-Presenter

OAuth – Open Authorization

REST – Representational State Transfer

SOAP – Simple Object Access Protocol

XML – Extensible Mark-Up Language

# Chapter One - Introduction

## 1.1 An Introduction to the Problem

### 1.1.1 The Problem Domain Twitter

Twitter is a social network and micro blogging service. Registered Users of this service can post and read short messages known as tweets. These tweets are text messages of 140 characters generated by an author. This may be publicly viewed by default, or the author may choose to restrict message delivery only to their friends list . Twitter users can subscribe to tweets from other author, this is called following and subscribers are called followers.

The authors of tweets, use twitter as an avenue to engage each other and in the process vent their opinions on topics in an unfettered manner, which largely reflects the true complexion of the dialogue.

A tweet could refer to one or more topic areas, each with a number of keywords that are normally part of the tweet or sometimes hash tags are appended at the beginning of the keyword.

These topic or keywords and their requirements will be explored in greater detail later in this report.

### 1.1.2 Problem Definition / Rationale

Computers and mobile devices have become ubiquitous in today's gadgetry. The technologies between mobile applications, web browsers and networking standards (e.g. 802.11n and 3G) etc have provided the state of being for social networks to provide informal communities where members generate and share content, interact and respond in an unfettered manner to stimulus especially from product and/or service providers.

Data from social networks has facilitated fundamental changes to business and societal practices. In a 2006 Forrester research, it was observed that societal changes were emanating from social mechanisms like instant messaging, blogs and social networks, where individuals took cues from themselves rather than relying on traditional sources of news, purchasing, and social interaction (Vannoy and Palvia, 2010).

In the light of this, social data can assist businesses and brand owners who require regular feedback from customers. Also this data could help drive innovation from both ends as customer opinions

are valued insights to evolving trends and can serve as an effective decisions making source for businesses to adapt to their customers requirements.

## 1.2 Project Aim

This project will aim to provide the public relations and/or marketing departments of businesses and brands with a usable system, which will attempt to make easier, and concise social topic search, identification of influential customer to foster better customer relationship and make online customer lead generation easier.

## 1.3 Project Objectives

- Find the most relevant influencers on relevant topics or search term.
- Rate influencers based on influence score.
- Provide visual summary on influencers.
- Provide Login and Authorization mechanism via OAuth.

## 1.4 Deliverables

- A web-based topic search and influence reporting system deployed to a Java EE based host.
- A report detailing the process of developing the above system

## 1.5 Relevance to Degree

As this chosen domain area is based on systems analysis and design, it is more of the same kind to the syllabus of the bachelors in business information systems degree, as opposed to a more scientific project which would be required from a bachelors in computer science degree.

Knowledge was therefore drawn upon from technical and system analysis modules such as Software Project Management 'Advanced Information Systems Development', 'Object Modelling Techniques', 'Software Engineering', 'E-Commerce and Web Design', Server Side Web Programming and 'Object Oriented Programming' in order to analyse the problem domain prior to delivering a solution.

However, certain areas of this project presented opportunities to explore areas that were not covered by the course and this required knowledge to be drawn upon from other technical subject areas, such as 'Functional Programming', 'Cloud Computing', 'Web Services (REST)' and 'Actors

Messaging'.

On the whole, the project is comprised of a balance of all the elements of the software development life cycle, allowing for knowledge to be consolidated from two years of learning.

## 1.6 Project Schedule (See Appendix)

This project's schedule is rendered out on a Gantt chart, as displayed in Appendix. This approach was chosen instead of a text-based project plan for its clarity, as it provides graphical representation, highlighting key deadline and/or milestones.

## 1.7 User Feedback

This project is basically an academic endeavour aimed at applying accrued learning. It lacks a particular 'client' for which the system is been built for. It is highly relevant to involve and gather the views of some of likely users. This was done in order to achieve some level of logical coherence and accordance with the certain facts throughout the course of the project. The views of two consultants were used to gain some domain knowledge and provide feedback. The two were chosen as they both have some domain knowledge and also because they represent opposite side of possible users of the system.

### 1.7.1 Consultant 1 Emeka Osuji

Emeka is a public relations officer with Amana Insurance in Nigeria. He has organised several advertising campaigns in the past for Amana. He was chosen to provide a viewpoint on customer relations, as well as a layman's perspective on the system.

Emeka is a not regular user of applications on the web and computers generally. It was expected that he would offer relevant ideas during feedback sessions on particular functionality through experience of using this project's web application from his own point of view.

### 1.7.2 Consultant 2 Jane Uluoma

Jane is an active tweeter and has a history in visual arts and graphics design work with some relevant entrepreneurial experience. She was chosen for her wealth of experience in aspects of the graphic design, and the ability to provide a manager's perspective.

Jane uses the Web very often and is an advance-level computer user. It was hoped that her

management experience combined with her advanced graphics computer know how would make her the right candidate to try on the usability of the system.

The consultants were able to gain access and use this project's web application remotely from Nigeria as the application is hosted on amazon's elastic cloud compute (popularly called EC2) platform for free via stax.net.

Feedback was received via email and telephone conversations.

## 1.8 Changes from the Mid-Project Report

### 1.8.1 Project Title, Aims and Objectives

Following the post project proposal discussion sessions with my supervisor and subsequent peer review with my projects cohort, it became apparently clear that the initial project title, aims and objectives required obscure and required modification to reduce ambiguity and focus on only topic search, Influence reporting, authentication and authorisation.

It was also discovered during this time that klout.com had already exposed an API that can be queried for relevant topics and influence data of twitter users. Based on these findings, the scope of the project had to change.

### 1.8.2 Project Schedule

It was vital to effect alterations to the project timing, which is shown in Appendix. Additional time was needed to learn and research technologies (i.e. Scala programming language and Lift web framework) this took a little longer than expected.

As it became clear that 5th November was set as the final meeting day before the project's deadlines, it was decided that iterations were to be cut down to three weeks each in length, this resulted in the system to be presented in its state of completeness.

## 1.9 Evaluation Criteria

In an attempt to weigh the achievements of the project, a group of evaluation measure have been set:

- Implemented Features This is the group of requirements and improvements initially included in the project proposal but were excluded for a prioritised and explicit set of requirements which came up after proper analysis was carried out. This will be employed during the evaluation.
- Non-Functional Requirements This a list of non-functional requirements that will be clearly outlined in the analysis chapter of this report. Factors like usability and adherence to standards of the web are to be factored into evaluation.
- Planning, Time Management and Development Methodology - The selected method for development will be examined closely, also how well the time was used and the part of planning to effective time management.

## 1.10 Summary

This project has been introduced as a solution to problems faced by business and brands owners today, namely issues around social media management, social search for influencers, CRM and lead generation. This web application has been proposed to address such issues, with an objective of being laid out to guide its completion. On completion, successions of evaluations are to be employed to examine the overall achievements of this project. The next chapter will be a detailed background on the research area, techniques and implementation steps taken in developing the web application.

# Chapter Two –Literature Review

## 2.1 Introduction

Since the inception of computing, our planet has witnessed waves of innovations and technology sweeps. In the 1970s, mainframe computers were used to compute and interpret data for decision-making within organizations. The 1980s, made way for personal computing, this was the beginning of automating manual tasks.

The 1990s, it was Internet computing, this became the first to indigenously facilitate groups and conversation asynchronously. And presently in this decade it is social computing that is permeating across the Internet via social graphs to connect people.

Businesses and brands have over the years heavily relied on print / digital media, telecommunications services and the Internet to reach out to their target audience. The print / digital media provides a one to many pattern while the telecommunications services provides a one to one pattern. The Internet gives a many to many pattern. The Internet has built-in structure for conveying telecommunication services via VOIP and print / digital media via TCP/IP concurrently. This lead to the migration of phone calls, music, movies and newspapers etc to the Internet (Shirky C., 2009).

Social computing combines the Internet, networking and communication technologies to facilitate participation, social interaction and human innovation. Social interaction denotes the mutual influence a person has on the other's behaviour.

With emergence of social media networks, the way people communicate, collaborate and share data has changed. Groups that see, read or hear about things of interest to them, can share information easily and gather around via social networks to talk to each other.

This has transformed the members of the former target audience from content consumers to content producers. These producers produce unfettered contents, which largely reflects the true complexion of the dialogue.

It appears that in the present scenery, media is global, social, ubiquitous and cheap (Shirky C.,

2009). The aggregation of these user-generated opinions is revolutionary for businesses and brands. This will drive innovation within them and help them make informed decisions (Surowiecki, J. 2004).

## 2.2 Web 2.0

2001 is marked as the beginning of technology sweeps on the web. It appeared at the time that the web was being publicized excessively and the resulting changes and buzzwords appeared to be a common highlight of these technological innovations.

Today Internet users in their alarming numbers are utilizing web technologies especially social web sites to connect with their friends, make new friends, and share the contents they discover or have, like social bookmarks, status message or monologues, news, photos, videos, and articles.

There are so many social web sites out there; at the moment the most popular ones are facebook, myspace, linkedin and twitter. These social networks claim to have user population larger than some countries in the world. Their features have been evolving quite rapidly. These features are heavily based on concepts of Web 2.0.

The name Web 2.0 was given after a brainstorming session at a conference organized by Oreilly and Media Live International, to the new innovations that had evolved on the web.

*“You can visualize Web 2.0 as a set of principles and practices that tie together a veritable solar system of sites that demonstrate some or all of those principles, at a varying distance from that core.”* (Oreilly 2005).

Social media mechanisms (which include instant messaging, blogs and social networks) are established on ideology and fundamental mechanics of Web 2.0, they are mostly a set of Internet based applications facilitating the creation and sharing of contents generated by its users (Kaplan and Haenlein, 2010).

## 2.3 Social Media and Interaction

Social media are communication mediums on the Internet for social interaction, exploiting web based publishing methods to broadcast and transform the monologues from individuals to social dialogues in an unfettered manner.

Social interaction underscores mutual influence we have on each other. Social interactions are the behaviours, gestures and/or actions of two or more persons cooperatively oriented towards each other's subjective intentions, sometimes in a symbiotic manner.

According to Kaplan and Haenlein, (2010) Social media can be classified into six varying forms: Virtual game worlds, virtual social worlds, collaborative projects, blogs and microblogs, content communities and finally the social networking sites. These social services are commonly integrated via social network aggregation platforms. This allows for users to socially interact via blogs (i.e. web logs), vlogs (i.e. video logs), picture sharing, wall postings, email, instant messaging, video sharing, music sharing, crowd sourcing, and voice over IP (e.g. skype) etc.

In November 2008, the North Americans had a presidential election and there were some fears in some parts of the country of voter suppressions. A plan came up to video the voting exercise. The idea was that individual citizens with mobile phones capable of taking photos and/or creating video, would be on the look out to document proceedings of any forms of voter suppression techniques at their local polling station and multimedia file are then uploaded this to the repository of videothevote.org and this served as operate as a form of citizen observation that will help maintain the sanctity of the voting exercise. This is a pattern that assumes that we are all in this together (Shirky, 2009).

What matters here is not the technical capital but the social capital; because now that media is becoming increasingly social, innovation can happen anyway at any time. Thanks to social media collaboration and communication innovation is happening everywhere, moving from one spot to the other.

The Sichuan province of China in May 2008, suffered massive destruction in a wide area to an earthquake. The earthquake was reported as it was happening, people were tweeting on twitter, taking and uploading video and photo footage of buildings shaking to social media web site.

And because of the social connection people around the world were able to get information about the quake. The BBC received hint of the Sichuan earthquake via twitter, ahead of the American geological survey. The earthquake china had prior to this one, it took them three months to admit that anything had happened. This wind of the quake attracted global interest and it was not quite long that donations and aid started pouring from all corners of the globe.

The media ecosystem has transformed as a whole, not just a particular instance. These case studies show how social media facilitates information dissemination to influence social behaviour and interaction, this does not only concern users generating and sharing data, but also affect organizations that want to reach out to their target audience. Most organizations that are trying to send messages to a distributed collection of the audience can use social media to talk to the audience and the audience can now talk back.

It appears that the part of the media landscape, which is primarily made up of the print/digital media and have professionals broadcasting messages to “us” is now gradually slipping away. In a world where media is becoming global, social, ubiquitous and cheap, in a world where the former audience are now increasingly full participants. In this world, it is now no longer about crafting a single message to be consumed by individuals (Shirky, 2009).

## 2.4 Social Media Analysis

Social media mechanisms have continued to turn up across the Internet in the form of p2p networks, discussion forums, instant messaging, blogs and social networks. These mechanisms host users on their databases. Everyone on a social network is a potential consumer of products and/or services. With social media at the disposal of product consumers they have in their possession a soapbox of unprecedented reach for information dissemination. The data they produce on social media networks is huge and provide valuable insight to the evolving trends brand experiences, and opinions (be it positive or negative).

The opinions of consumers on these social networks are capable of influencing other consumers and also their brand advocacy, buying decisions and brand loyalty. An example of this is a satisfied and loyal customer of tesco using the tesco logo as her profile picture on a social network. This often occurs and only tells of how satisfied and happy this customer is with the company.

Businesses require an interface to help them filter the social web of enormous data of content producers. This will give them valuable insights into what their consumers feel about their products.

Social media analysis integrates analyzing, monitoring and measuring data generated on social media networks. These Social networks usually provide Application-programming interfaces that provide set of methods that can be invoked remotely to return required information.

This is most required by marketing and communications/PR teams of organization. This helps them to plan their marketing campaign and choose what channel of communication to use.

Social analytics is very important as it allows companies to capitalize on consumer generated content, which is honest, unfettered and innovative.

# Chapter Three - Research Methodology

## 3.1 Introduction

Kappel et al. (2006) states that the ever evolving nature of web development, owes to three key determinants namely: continual transformation of web frameworks and standards, pressure from competitors and short life of web applications coupled with the quick pace of development. Having said this, a suitable methodology to match the ever-changing nature of web development is required to be implemented. This chapter delves in to evaluate some known software development methodologies.

Firstly, it was considered whether to employ the waterfall model over an iterative line of development. However, Sommerville (2008) highlights in his book, that the waterfall model should be employed when the required outcome are properly understood and is very uncertain to change radically while development is already in progress or put in Nigel's words the requirements are 'Cast Iron'.

As long as the domain in context is already well understood through experience the waterfall model would fit.

This method however contradicts the earlier described evolutionary nature of web application's development. This obviously gives no room for using the waterfall model in developing web application as it is very likely that technology and requirements could change.

It appears that an iterative mode of development seem adoptable for this problem domain, as it would make room for changes while development is on going.

## 3.2 Methodologies

### 3.2.1 Agile Software Development

The focal point for agile software development is on the customer's requirements. The agile manifesto places the customer at the center of it all.

Agile principle advocates after seeking the customer's consent at the end of an iteration and also agile software development in general appears to be more focused around teams or groups of teams.

Agile does say much about in favor of single developers or solo development and also development for one particular client with the allowing the developer to implement his own opinions or

requirements.

### ***3.2.2 Extreme Programming (XP)***

Extreme programming (XP) was also evaluated, however, it appeared to be similar to the agile model. This option was abandoned for the already mentioned reasons. Amongst the principles of the extreme programming methodology is working in close collaboration with the customer and requesting sign off while development progresses from stage to stage. Applying XP practices to this project seem to be very difficult as there existed no particular client to regularly collaborate with.

### ***3.2.3 Iterative and Incremental Development***

This methodology for software development appears to suit this project pretty well. With this, requirement are group into features, and are tagged with priority ratings, then features that will be reused by other components are allocated first to earlier iterations.

The earlier the number of features are made known, the more easier it will be to plan the succession of iterations.

It also appears to be accommodating to team and individual based development.

### ***3.2.4 Iteration Length***

Pilone and Miles advised on the use of shorter iterations, making is more easier to respond and deal with change. Due to the length of the project, three four-week iterations were scheduled.

Although, it was adjudged to keep to medium-length four-week iterations, in order to properly understand the problem domain, and consequently it was hoped features would be kept simple and clear to avoid any changes in the requirements in-between iterations.

Further on in the project is was decided to cut done the iterations lengths to three-weeks due to alteration in the schedule, as discussed in earlier on.

### ***3.2.5 Conclusion***

In conclusion, it is obvious at this point that the chosen method to be employed is the iterative approach, this involves identifying features and prioritizing assigning each of them to an iteration.

Normally, there would be some short cycles on the project's completion this is due to building of web applications are known to be never ending or continuous, this will give chance for features to

be injected and implemented during the course of and on completion of development in future iterations.

## 3.3 Implementation

In building a web-enabled application, there are loads of technologies to be chosen from. Although HTML is the de facto standard for rendering a web page, also JavaScript is very well known and suited for determining browser-based behaviors. Whilst this holds true there also are in existence, programming languages and application frameworks for web and software development that can complement their roles on the server-side.

The considerations made when choosing development tools for this project were mainly based on the requirements of the web application. Which are as follows:

### 3.3.1 Development Technology Requirements

The key concerns that are considered when selecting a development tool, framework or language for web applications, mostly relate to issues revolving from usability, security and performance. This determines the selection of a framework or tool that caters for security and results to quick execution intervals.

This web application is a social network based analytics tool, the data that it obtains from twitter and klout API are publicly available and in some cases the API's require knowledge of the user that is requesting data from it, it may be required to authorise access to data.

At the time of writing this report, very large number of users at any one time is not anticipated, it is expected to cater for at least 150 concurrent users. Therefore high performance responses may be required from a chosen language or framework. Though, the web application is required to load almost in no time.

In a study by Akamai and Jupiter Research (2006) it is highlighted that visitors to online applications are likely to lose interest if the page take as little as 4 seconds to load. The likelihood of this occurring, can be reduced by the design of the presentation layer (an example is by reducing the size of graphics) and discouraging eager evaluation of an object's properties and methods.

For this project, the development language requirements were decided on the process of development, rather than the requirements of the web application. Subsequently, a list of requirements was concluded on:

- The chosen development language should be relatively easy to learn.
- The chosen development language must have a well documented API, books and relevant tutorials to aid development.
- The chosen development language must be expressive, elegant and concise.
- The chosen development language have
- The system being developed will be quite complex, the chosen development language must allow for fast development and minimise repetitions.
- Based on the above point, a chosen language must have several tools to choose from. For instance a number of library, frameworks and plug-ins to possibly help speed up development.

From the already stated requirements, the server-side development language would be chosen first, with any libraries and frameworks to aid the development. This would reduce the time spent reviewing server-side languages for a number of tools for any given language.

### ***3.3.2 Server-Side Languages***

#### ***3.3.2.1 Java***

Java at first appeared most suited to the requirements as it is the one of the languages that was used throughout university; based on this, there existed no issues in the time required to get the cognition to use it.

It would also allow for fast development, also has very good documentation, tutorials, libraries and frameworks to select from, including JSF, Apache Wicket and Tapestry, these may speed up development.

Nevertheless, owing to the prior level of familiarity with the language, it goes against a personal requirement of new skills development during this project.

### **3.3.2.3 PHP**

PHP is a well-known and widely used server-side scripting language for Web application development. It has built-in functions that support sorting algorithms, session handling and a lot more out of the box.

This will reduce the time needed to spend coding and ensures effective algorithms are used. Having obtained some knowledge of PHP before now will cut down the time spent to learn the language, though this already acquired knowledge is not far-reaching; a substantial number of new skills may be required.

Owing to the fact that PHP is an open source scripting language, it has got a lot of examples and good documentation on the PHP.net, a lot of good tutorials on youtube.com and other location on the web and most importantly an active community of committers and users.

This is the reason why there exist a number of web frameworks that help speed up development, such as Symfony, Zend, CakePHP, Joomla and Drupal. Hosting for PHP is widely provided for by hosting companies due to open-source and free nature. This has resulted in very cheap hosting costs.

One of the concerns with PHP is code organization. Namespaces don't exist, exception handling didn't exist till PHP version 5, built-in libraries and API's are not organized and many of the errors are latent or silent. This goes against personal requirements for writing readable and modular applications.

### **3.3.2.4 SCALA**

Scala is an acronym that stands for Scalable Language. It is a statically typed functional and object oriented programming language that runs on the Java Virtual machine (JVM) and the .NET common language runtime (CLR). It has this feel of a truly dynamically typed language but really it is not dynamically typed.

Scala on its functional side is a branch of the ML languages (example Haskell), they are known to be strongly typed unlike their homoiconic and dynamically typed functional language counterpart like LISP or Dylan.

The main principle of Scala is that functional programming complements object oriented programming, despite the surface contradictions. Functional programming places her emphasis on immutable values and side effect free functions as opposed to object oriented programming which is all about state that is mutated in objects and methods that can modify state that are out of their scope.

But combining these two paradigms in Scala has proven to be worthwhile. Scala makes it possible to create immutable objects and also apply object-oriented goodness. In Scala, every value is an object and every function an object (Wampler, D, 2008).

Scala is functional, this helps to write lesser and more concise code making it easier to test and maintain large code base. Because Scala is functional in nature it embraces a lot of mathematical concepts, which emphasize correctness from a logical point of view, which is very important and can help reduce bugs in the software system.

Scala has built –in XML support. It treats XML as first class citizens. In scala XML code can be placed side by side with inline with code as the scala compiler will sensibly type XML elements or nodes (Subramaniam, 2008).

Scala introduces better ways of doing object oriented programming, in terms of composability (using mixins and traits) and scalable design. Scala provides a better model object oriented programming approach. Scala provides a much more principled approach for dealing with concurrency issues.

It has an Actors library that is similar to the one in Erlang programming language. Actors are nothing more than concurrent process actions that communicate by exchanging messages. Actors can also view as active objects, based on this a method call corresponds to sending a message. The Scala Actors can also do asynchronous and synchronous messaging.

### **3.3.2.4 CONCLUSION**

All-in-all, Scala was chosen as the server-side programming language for this project. This choice was based on prior knowledge gathered from personal engagements with JVM based languages and the desire to increase understanding on the functional paradigm, allowing for new skills to be acquired.

It will also prove to be beneficial in building web applications as it has built-in support for XML processing, which is a big plus for web applications development, which are based on HTML a subset of XML.

It allows for very concise code to be written, this makes testing and maintenance easy as the size of the code base is very small when compared to Java based code. Also, it helps to build faster responding web applications; this is due to Scala's Actors concurrency library.

### **3.3.3 Development Tools and Methods**

With Scala chosen as the server-side programming language, the methods for developing the system are bespoke development and development with the use of a framework.

#### **3.3.3.1 BESPOKE DEVELOPMENT**

Bespoke development would possibly give way for skills development with regards to programming and understanding more about the entire system's architecture, although this may to some extent become time-consuming, with the code written repetitively. It could also help achieve faster loading times, for the web application.

#### **3.3.3.2 FRAMEWORK DEVELOPMENT**

A Web framework consists of abstractions and commonly used web development algorithms to avoid wasting time writing repetitive code, and could organize code into logical files, packages or namespaces that attend to the data (model), controller and view tiers of a web application. This is referred to in many literatures as the Model-View-Controller (MVC) pattern or architecture.

This allows for easier maintenance, development and testing of the web application. On a personal note, the skills will be acquired from the framework used may be useful for projects in the future.

### **3.3.3.3 CONCLUSION**

To conclude, of the two methods researched, the employment of a Web framework is most appropriate of the two, as it will ensure as many features as possible can be applied to the web application. Although the learning curve is quite steep, but as the knowledge of the framework became more familiar, the framework will assist in faster development, easy testing and maintenance.

This will help to cut down repetitive code and make developing the system less arduous and allowed time for innovations and possibly implementing some advanced features.

The decision now shifts to which web framework to use, at as the time of writing this report Scala is having a number of options, each with their respective features and complexities.

### **3.3.4 Web Frameworks**

In order to choose a Scala based web framework, a search for available frameworks was made. The bulk of frameworks have very much alike features, this made the selection extremely hard.

Opinions and stats were scarce, and it would be difficult to make a personal decision by given time to become familiar with all the frameworks. Google Trends was used to provide statistical figures of the number of times the frameworks were searched for.

However, this is not only able to give evidence of their popularity, but assist to confirm the conceivable amount of support and discussions available. As at the time of writing this report, it appears that there exists only two stable web frameworks on the web. These are ‘Lift Web’ and ‘Scalatra’.

Owing to the fact that, the web frameworks considered have unambiguous acronyms or names, the results from Google Trends were not probable to be affected by getting mixed up with unrelated subject matters. Therefore, the frameworks considered in the search were ‘Lift Web’ and ‘Scalatra’. The results were as follows:

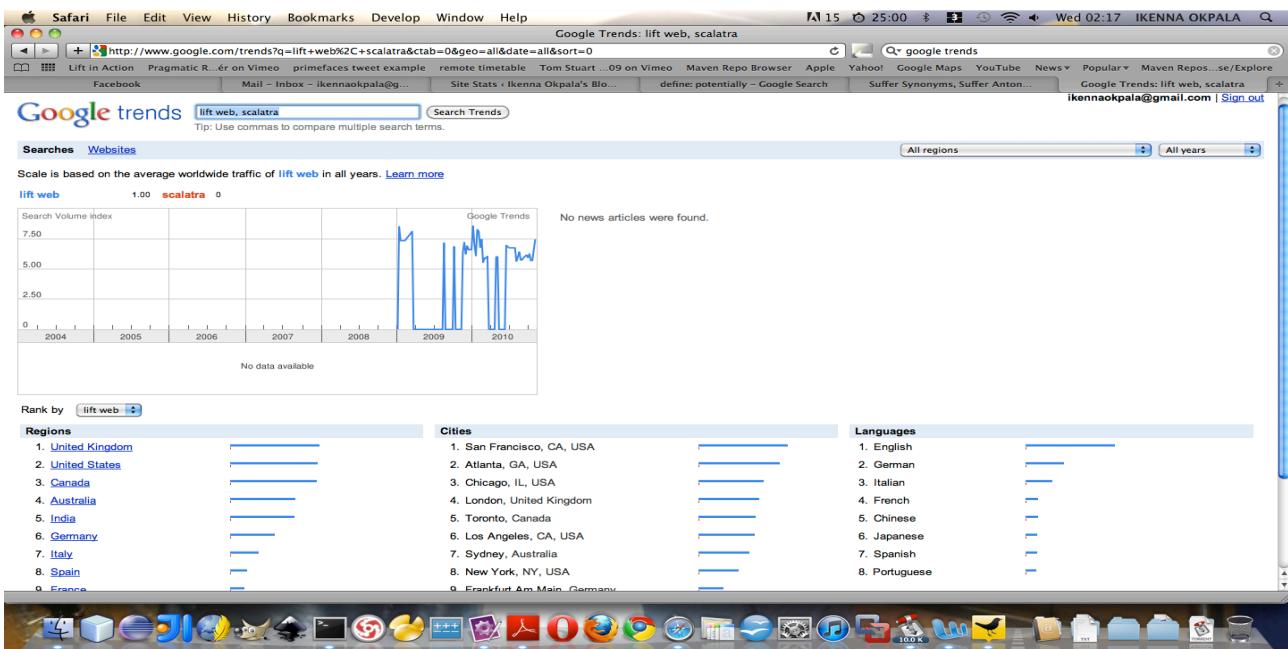


Figure 1 Google Trends Search Result for Scala based Web Frameworks.

The results therefore showed ‘Lift web’ as a clear leader with 1.00 points, with ‘Scalatra’ with no points.

### 3.3.4.1 LIFT WEB FRAMEWORK

Based on the subsequent research done on the web, it appears that ‘Lift Web’ is the most searched of the two, this is as a result of it’s active community of committers and users. This means that code commits, bug fixes, support and discussions were readily available from the community itself. It is a web framework written in Scala. It’s approach to web development, is coined “view first”.

The “view first” approach could be thought of as a modified Model View Presenter, consisting of the ‘View’, ‘View/Model’ and ‘Model’ (Perrett, nd).

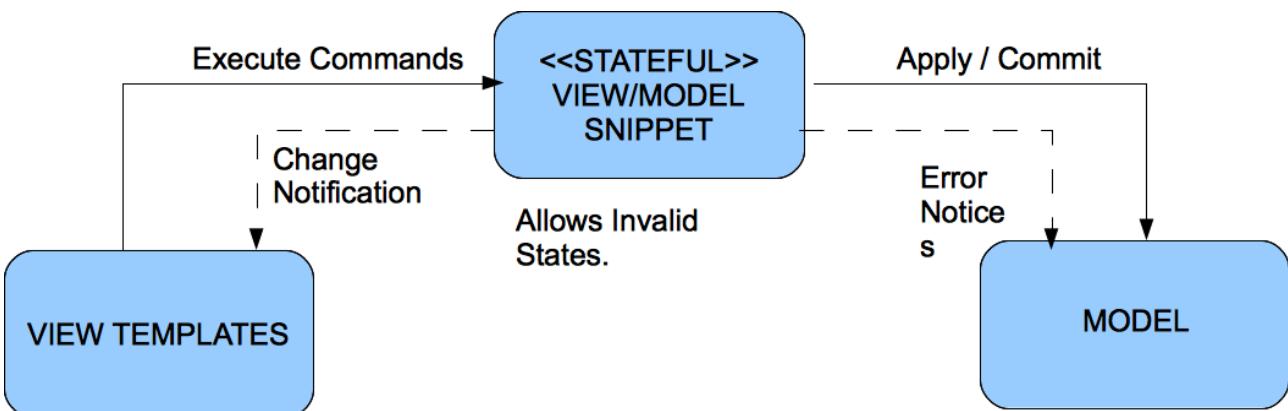


Figure 2 View First Approach.

With this approach a component of web page is an instance of a Scala object, these are called snippets in Lift. Snippets basically are bindings between the view and a function that could contain some business logic and primarily transforms input XML to output XML or put in another way renders dynamic or interactive feedback to the View Template. They could be easily mistaken to be controllers, but they are not.

Lift is expressive, elegant and has a productive sensation. It does a lot of the plumbing on the view side, especially with regards to form rendering and security etc. It leverages the power of Scala's xml and Actors libraries amongst others.

And because Lift is based on Scala, it is possible to play with legacy java code seamlessly and combine both functional and Object Oriented features like immutability, pattern matching, closures and high order functions etc.

### **3.3.4.2 SCALATRA WEB FRAMEWORK**

Scalatra is a web framework written in Scala. Scalatra is based on its own Domain Specific language (DSL). With this is possible to prototype a web application with inline HTML and business logic. It embraces HTTP and avoids session state. It also leverages the Scala's XML capabilities, functional conciseness and appears to be very tiny web framework to test or maintain.

In terms of comparison with 'Lift Web', Lift has been in development since 2007, while Scalatra started in 2009. At the time of writing, it appears that 'Lift Web' has more features and is being used in production by mostly start-up social media companies. 'Lift Web' has got a demo site with code snippets covering various areas of web development and a free online book to serve as a reference guide for developers.

Lift also comes with a template-generating engine called 'Lifty'. This generates template files that allow for rapid development. In addition, Scalatra is similar in many respects to Sinatra a Ruby based web framework, one of the popular frameworks for web development, which has been used to develop famous, large-scale websites like Heroku.com.

While this brings no benefit to the project, on a personal-level, learning 'Lift Web' will expose the new and evolving patterns such as the "View First" that are currently being used in developing

social media based applications, which in turn will be a plus for the future.

### ***3.3.4.3 CONCLUSION***

Whilst on a personal-level, it appears more beneficial to employ ‘Lift Web’ framework, having read the documentation for ‘Scalatra’ and ‘Lift Web’, ‘Lift Web’ seem to be more appealing, active and will be relatively quicker to learn, which agree with the requirements in section 3.2.1.

Upon trying out ‘Lift Web’ to test out its features, the benefits were reaffirmed, as it was easy to get started with and have a collection of all the features that were needed for developing this project.

Therefore, with ‘Lift Web’ it is hoped that it would assist to develop the required functionality for the final system, owing to the limited time available to learn the framework.

## ***3.3.5 Web Services***

Simply put a web service is a Web API of objects and behaviour that can be invoked via HTTP and executed on a remote system. The Web API came into being with in the era of the Web 2.0. Web services can be spilt into two groups the big web services and REST web services. To a very large extents Web API’s normally are connected to one or more remote datastores or databases that are queried for data.

### ***3.3.5.1 BIG WEB SERVICES***

Big web services follow the standards for XML based message exchange as stipulated by SOAP. This is very commonly used in traditional or legacy enterprise systems. In these systems there exist machine-readable operations that are described in a Web Services Description Language (WSDL), this is very important for generating code on the client side and is not required for a SOAP endpoint.

### ***3.3.5.2 REST WEB SERVICES***

REST is an approach to software architecture meant for distributed hypermedia systems like the World Wide Web. It consists of clients and servers. The client’s make requests via a HTTP URL to the server, the servers processes the request and responds with appropriate response usually responses may be in XML or JSON formats. REST basically describes architectures that employ

protocols like HTTP by restraining the interface to a set of already defined, standard operations (such as GET, POST, PUT, DELETE for HTTP). For REST based architectures, they are centred on interacting with resources that are stateful, instead of messages or operations.

### ***3.3.5.2 CONCLUSION***

For this project, we will use REST as the standard for requesting data from twitter and klout API's. This is solely because the twitter and klout API's are based on the REST architecture by default.

With the choices already made, that to use Scala in conjunction with 'Lift Web', and also a requirement to make calls to the twitter and klout API's. It became necessary to use a library that makes it easier to communicate with these APIs. Such features exist in the 'Lift Web' framework, but were found to be generic in nature.

Based on discussions from stackoverflow.com and the scala mailing list, Databinder Dispatch was recommended and will be used for accessing HTTP services for this project. This library is a written totally in Scala and contains already baked algorithms for twitter calls and the 'Open Authorization' popularly referred to as OAuth, which will be discussed in a later section.

### ***3.3.5 OAuth***

OAuth is a service based on open standards for authorization. It serves as way to allow users to share their private resources (like photos, video and friend list etc) without having to give away their credentials like username and password, instead it only requires handing out of tokens to the service provider.

Each token gives access to a specific service provider, for a defined set of resources within a limited time duration. It is very good for granting third party access to the users information hosted by a particular service provider or in this case a social media network like twitter.

For this project we will use the Databinder Dispatch library to provide authorization to view influence reports of twitter users.

### **3.3.5 Other Technologies**

One technology that will be operating behind the scenes is AJAX (Asynchronous JavaScript and XML). It is a technique used to retrieve data asynchronously from the server to the client without re-rendering the entire web page, but rather a snippet of it. The chosen web framework for this project provides AJAX by default.

jQuery is a concise and fast JavaScript based Library that make simple the traversing event handling, animating, and Ajax interactions of HTML documents for rapid web development. The chosen web framework for this project provides jQuery support by default.

### **3.3.6 Usability and Standards**

Usability is an aspect to be taken seriously in designing web-based applications. Nielsen (2000) compares software design to web design; “*in software design, customers pay first and experience usability later; on the web, users experience usability first and pay later*”.

While this web application will not sell products, it must conform to usable standards. Nielsen and Loranger (2006) recommended having a list of ‘dos’ and ‘don’ts’ when designing the front-end of web applications, such as avoiding pop-up windows and using menus to group to categorise data. These will be a point of concern while developing the solution.

Web standards are very important component in web development, owing to the different software and hardware that users accessing the web application will be using. The project will however be developed with XHTML and CSS standards. Testing will also be carried out on a range of web browsers.

### **3.3.7 CONCLUSION**

A number of technologies have been examined so far for this project, some were explicitly explored through ‘hands-on’ evaluation, such as the programming languages, Web frameworks and Web services libraries, others select based on recommendation or as de facto web development standards, such as XHTML and CSS.

Ultimately, the technologies considered to be appropriate in achieving the required functionality from the web application were Scala as the server-side programming language, in conjunction with ‘Lift Web’ as the Web framework, which will interface with REST endpoints.

## 3.4 TESTING

Several testing methods will be used to ensure the condition of the system. While development is being carried out, unit testing will take place with test data testing the unit's response to acceptable, invalid and boundary values. As the iterations are completed, testing will be carried out on the system for usability by inexperienced users, whereby an inexperienced user uses under supervision in order to find flaws.

After the third iteration, it is expected that the system would at Alpha release level, and at this stage the system would be ready for testing by selected users. Regimented and Rigorous testing would occur if the system is required to yield a watertight final release to a customer. Although, testing is very important to provide a quality product, even though the release is only for testing purposes initially, to avoid creating more work in the future.

## 3.5 SUMMARY

This chapter has highlighted out the research, which has been carried out with the objective of identifying the optimum way complete the development of this project. This has lead to an appropriate methodology to be chosen; also including extensive research into the web based development technologies to be employed in developing the system, and concluded with a concise look at testing.

The next chapter will analyse the problem domain, in depth, with the aim of coming up with a list of requirements that will be used to evaluate the success of this project.

# Chapter Four - Analysis

## 4.1 Introduction

In order to develop the system in accordance with the requirements of the project, an analysis of the problem domain was conducted. This analysis used information from several sources; largely from knowledge acquired personally through years of first-hand experience combined with knowledge provided via interviews from the project consultants, but also the bringing together of information from existing social media blog articles, discussions on social networks, and tweets.

On completion of the analysis, the knowledge gathered was used to compose a list of system requirements that is used to design, develop and evaluate the level of success of the system.

## 4.1 Social Influence

Expressing oneself through interacting with others is the significance of being human. With social media sites like Twitter, new modes of interaction have been opened for people to communicate with themselves. In the social media domain, some people play a key role of setting trends and have become influencers that are followed by many (Shirky, 2009).

Using computer based devices to search, capture, quantify and analyze social interaction/behavior patterns grant us the opportunity to identify these pace setters and comprehend the reason behind their influential status. On a personal-level, it has an area of fascination to design and build software application to identify these influencers by their topics or subject areas that they are actively engaged in (Shirky, 2009).

With a recent shift of marketing activities to social media networks, the need for an interactive software application that can assist find the best way to market to online consumers. Influencers sit at the top of the social media knowledge dissemination pyramid. When their tweets are re-tweeted, the tweet cascades down his network of followers and also the network of those following him. It immediately becomes clear how powerful influencers can spread information. They have got the unique capability of surfacing interesting information and present it ways that attract people to want to click and find out more (Shirky, 2009).

Using influencers for information dissemination or marketing revolves around determining key influencers, marketing to these persons and then conducting latent marketing through the influencers to reach potential/existing consumers whose attention are held by them. However, this

method of information dissemination comes with issues, such as how to identify relevant influencers according to subject matter of interest like groceries, fashion, sports or even politics?

Computing algorithms, web based technologies like REST and meaningful make is very possible to locate influencers in any topic area.

## **4.2 Measuring Social Influence (the klout way)**

Currently, a company called klout has taken up the challenge of measuring social influence of twitter users. It has identified and categorized four measurement metrics to be used in evaluating a users social influence. These are klout score, true reach, network influence and amplification probability.

### **4.2.1 KLOUT SCORE**

The Klout Score is the overall measurement of the twitter user's online influence. The klot score ranges between 1 to 100. A high Klout scores simply means that the particular twitter user has a stronger and wider region of influence. The Klout scores are correlated to number of retweets, clicks, and comments (Klout, nd).

### **4.2.2 TRUE REACH**

True Reach is a score indicating how big the audience the twitter user can engage. This is based on how those who follow a twitter user actively respond to tweets from him. The response could be in form of retweets, click the links embedded in the tweet and replying to the tweet etc (Klout, nd).

### **4.2.3 AMPLIFICATION PROBABILITY**

Amplification Score is the likely chances that tweets of a particular twitter user will produce actions (retweets, @replies and clicks etc) and it is scaled on 1 to 100 (Klout, nd).

#### **4.2.4 NETWORK INFLUENCE**

Network score shows to how influential the audience engaged by a particular twitter and it is scaled on 1 to 100 (Klout, nd).

From the analysis conducted for this project, twitter was identified to be primary source for data. The system will have to cater for both types of club, so it is important that it has flexibility built in to do so.

### **4.3 Requirements**

From the enquiry undertaken, the user requirements were identified and drawn up into four groups, ‘must have’, ‘should have’, ‘could have’ and ‘won’t have’. The first three groups, form the core of ‘nice-to-have’ features, these may be included depending on time. The last groups, ‘won’t have’, are features that are not required or unfeasible in the scope of this project. These requirements will be used to plan the three scheduled iterations.

#### **4.3.1 MUST HAVE**

- An influencer search by keywords or topics.
- Graphical visualizations of the klout score for all the top influencer of from a topic search.
- Sort list of all the top influencer of from a topic search by klout score (highest to lowest).
  - This must include the twitter user’s picture and klout score.
- Authenticate and Authorize twitter users via OAuth.
- Display tweets of the current twitter user.
- Provide influence summary report for each twitter user.
- Provide visualization summary report for each twitter user

#### **4.3.2 SHOULD HAVE**

- Administrative section.
- Influence matrix report for each twitter user.
- Content and topics analysis for each twitter user.
- Archiving of tweets and other relevant twitter data.

- Statistical metrics values for the following variables:
  - Total re-tweets.
  - Mention count.
  - Unique messages re-tweeted count.
  - In/Outbound message ratio.
  - Total likes.
  - Total comments.
  - Follower/follow ratio
  - Follow back count.
  - Unique senders.
  - Unique likes.
  - Unique commenters.
- Annotated Time Line graphs.
- Tree Map graphs.
- Map of the world visualization to show the geographical locations of influencers.
- Twitter and Facebook share buttons.

### **4.3.3 COULD HAVE**

- An installer for the initial setup of the web application.
- Ability to create twitter list.
- Ability to tweet directly.
- Ability to favourite a tweet.
- A mobile client version of the web application.
- Facebook integration.

### **4.3.3 WON'T HAVE**

- Ability for users to upload resources.
- The ability to link multiple web application together to share results.
- A content management system.

## **4.4 Non-Functional Requirements**

### **4.4.1 USABILITY**

With an expected variation in computer-literacy levels of prospective users, it is vital to build the system to be very easy to use. This will include rendering a web page to the user that is focused. The usability of the system will be evaluated at the end of each iteration by users; having a user with very little computer literacy to use the system while checking for any conflicting issues. Usability shall be at the forefront of any design and shall be an ongoing consideration during development.

### **4.4.2 COMPATIBILITY**

For every web application, it is imperative that the pages load consistently over all web browsers. In same line of thought like usability, the users dictate what web browsers they be used. The system will consequently conform with various web browsers, like Internet Explorer 6 to 8, Mozilla Firefox versions 2 to 3.6, Apple Safari version 3 and 5.0.2, Google Chrome version 4 to 7 and Opera version 10.

To ensure cross-browser compatibility, the web application shall conform to XHTML and CSS standards, as validated by the World Wide Web Consortium. JavaScript will be used for front-end validation, providing feedback to users.

## **4.5 Summary**

This chapter has delved into the requirements of users and the system as a whole in order to create an itemized list of system requirements. This list will be used in the next phases of this project to prioritise and schedule the implementation of features and to judge the success of the project.

# Chapter Five - Design

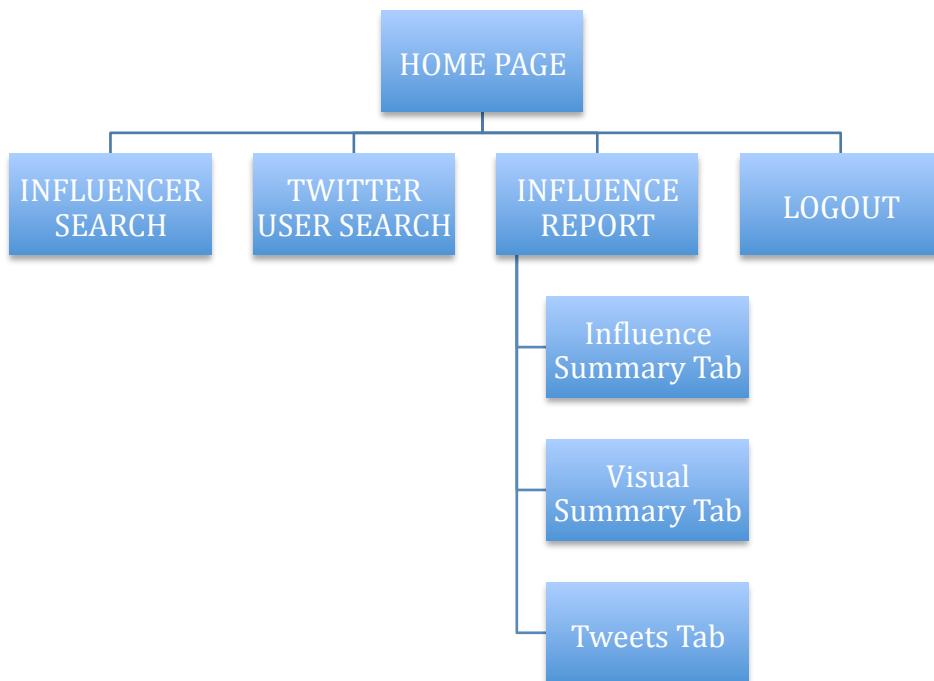
## 5.1 Introduction

Notwithstanding the project's development methodology had already set out three iterations of development, an initial design stage was scheduled to plan out the fundamental structure of the system. This design stage includes a diagram of the structure of the web application, layout of user interface, use case diagram, class diagram, component diagram, sequence diagram and deployment diagram. This chapter concludes with a plan for the three iterations.

## 5.2 Structure of the Web Application

The web application will consist of following web pages and url path; Home Page (/), topic search (/search/index), twitter user search (/search/index2) and twitter user influence report page (/[screenname]).

### 5.2.1 FRONT-END STRUCTURE



## 5.3 Presentation

In addition to logical design and structure of the web application, it was vital to design the structural layout of the application with the help of mock-up diagrams. The following act as the layout guide in which content was placed.

### 5.2.1 FRONT-END MOCK-UP

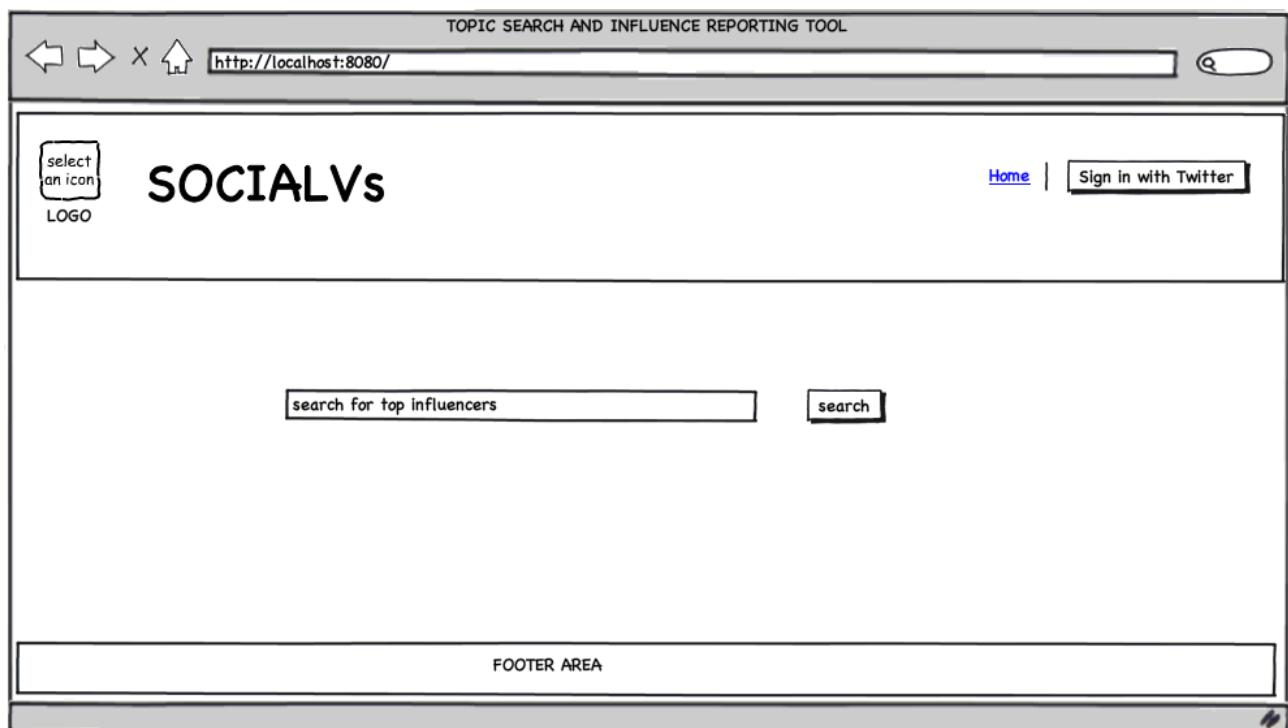


Figure 3 Home Page

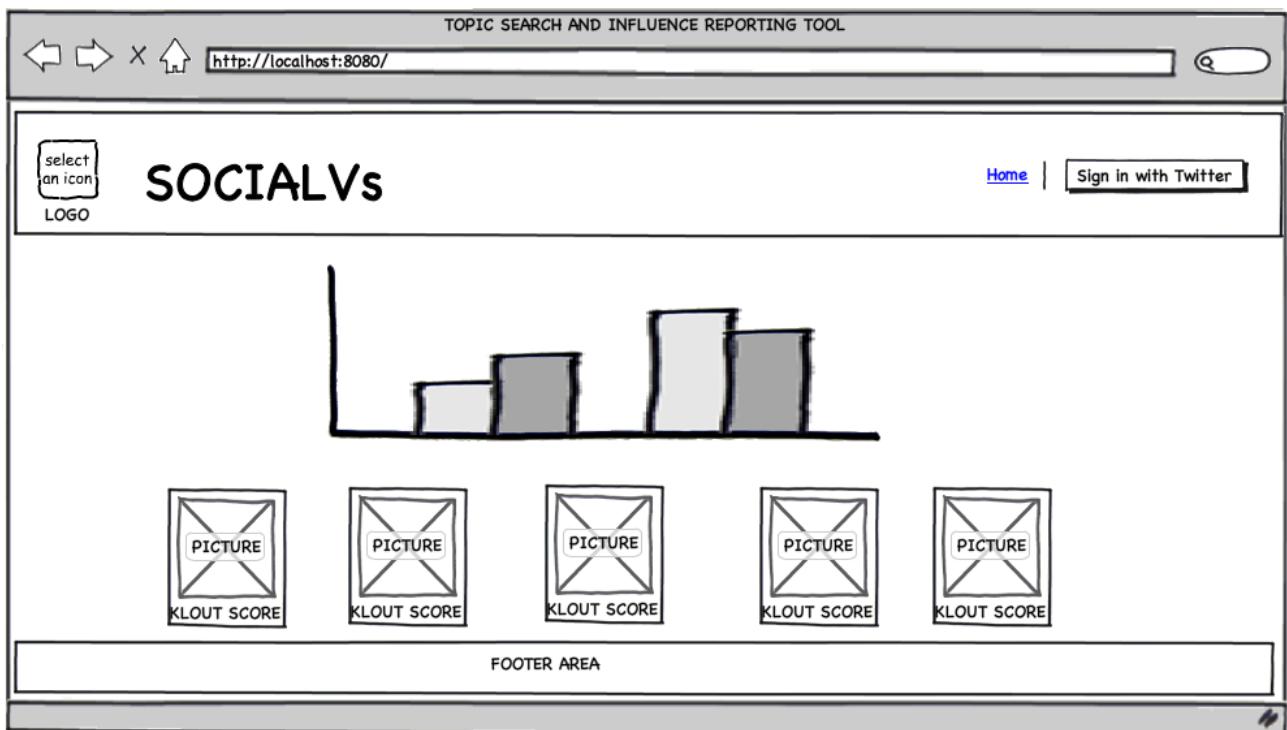


Figure 4 Search Results Page

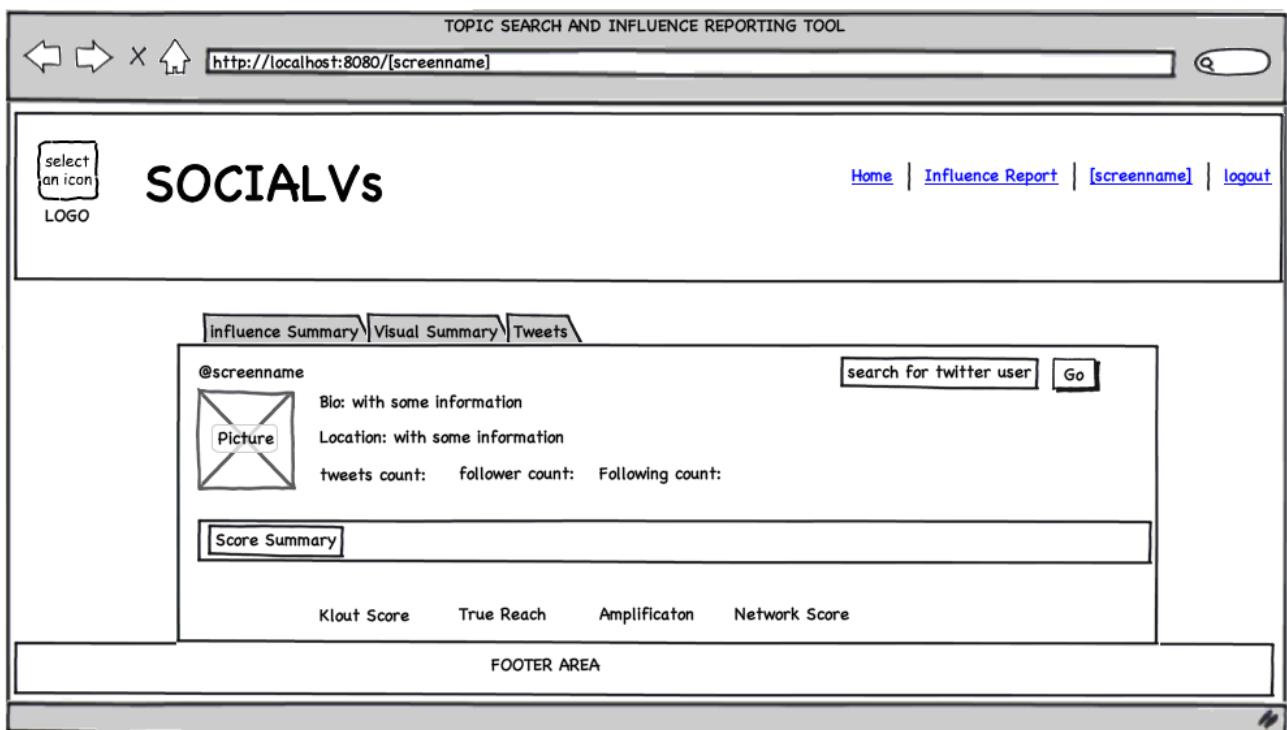


Figure 5 Influence Report Page

## 5.4 System Design

This section provides comprehensive architectural overview of the System. It serves as a communication medium between project owner and other stakeholders regarding System Architecture. The web application will be developed using component based, Java EE Architecture and provides the following UML diagrams to explain the same.

### 5.4.1 USE DIAGRAM

Below is the Use Case Diagram for the system. This diagram presents a graphical summary of the expected functionality of the system regarding its actors; their goals and the dependencies that exist.

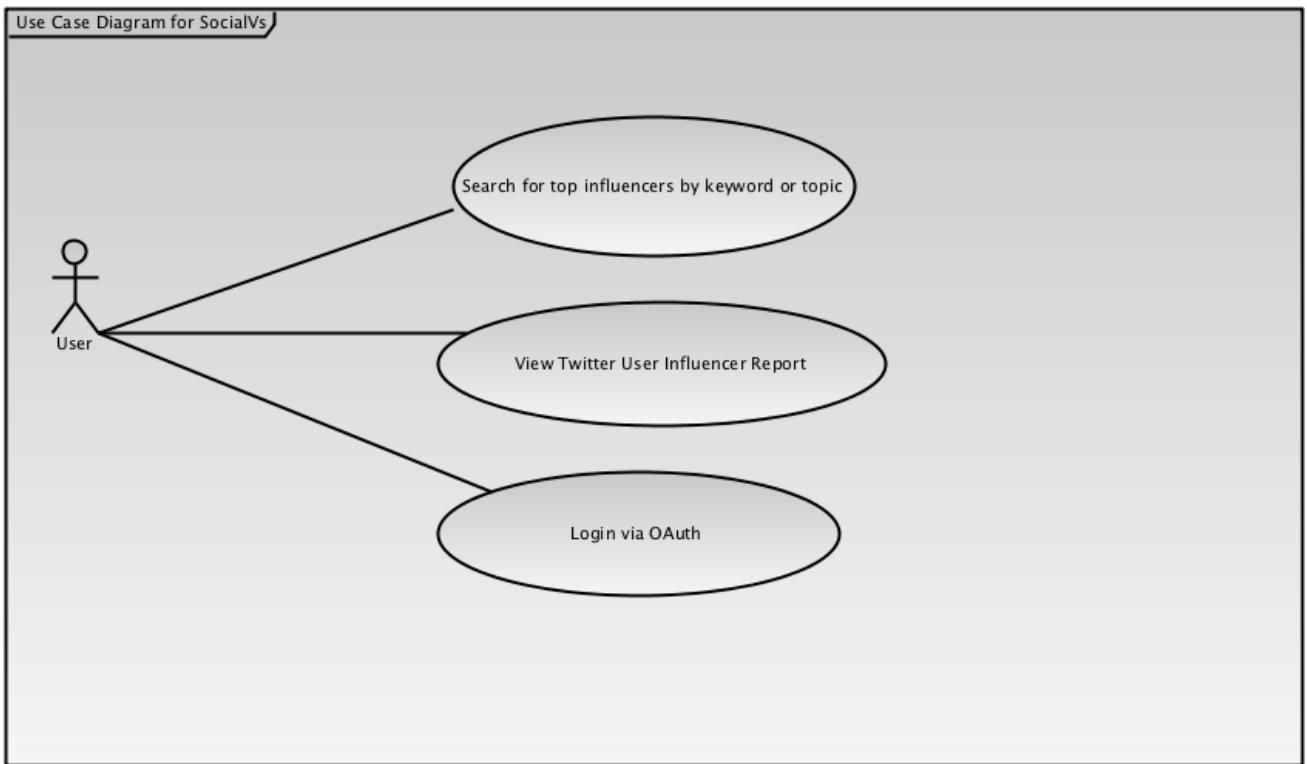


Figure 6 Use Case Diagrams

## 5.4.2 LAYERS OVERVIEW

The layers for the system is broken down into three layer as shown below:

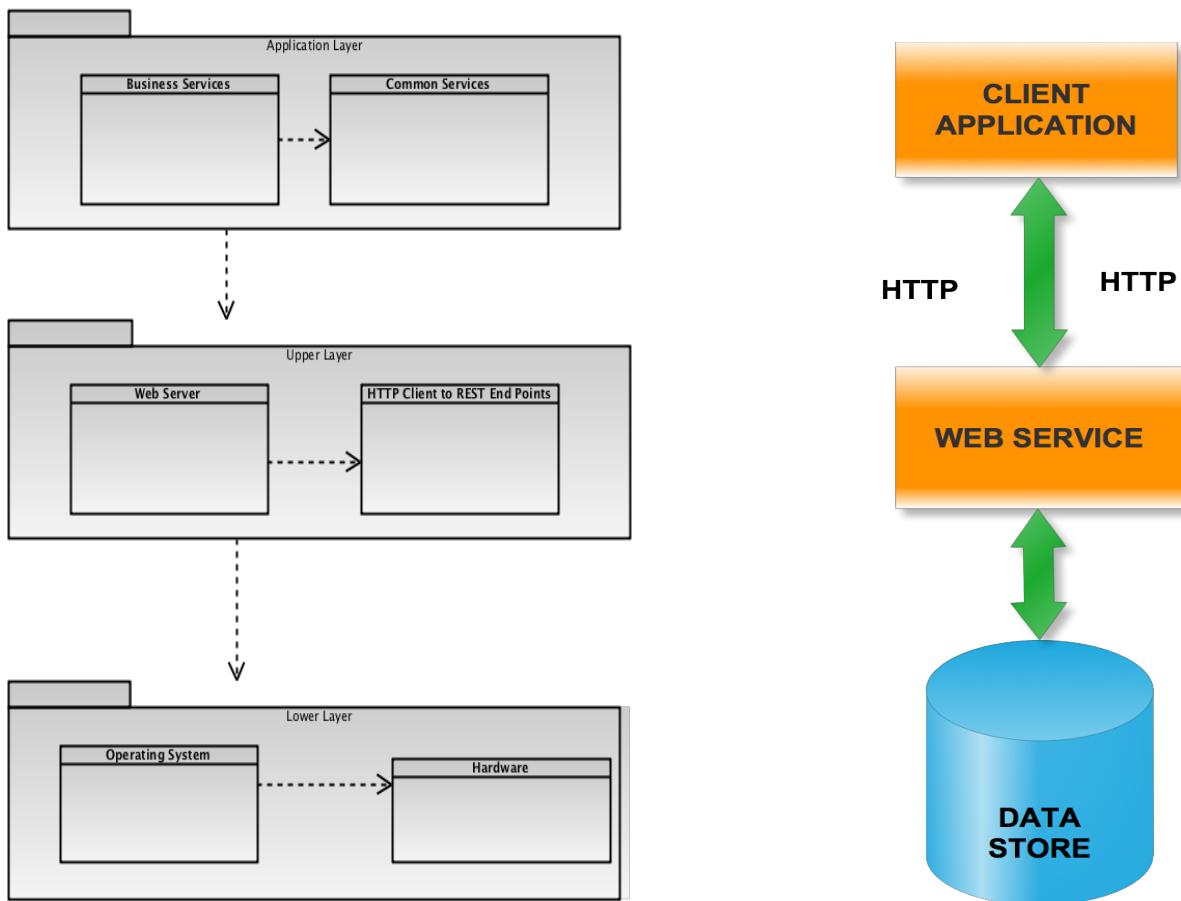


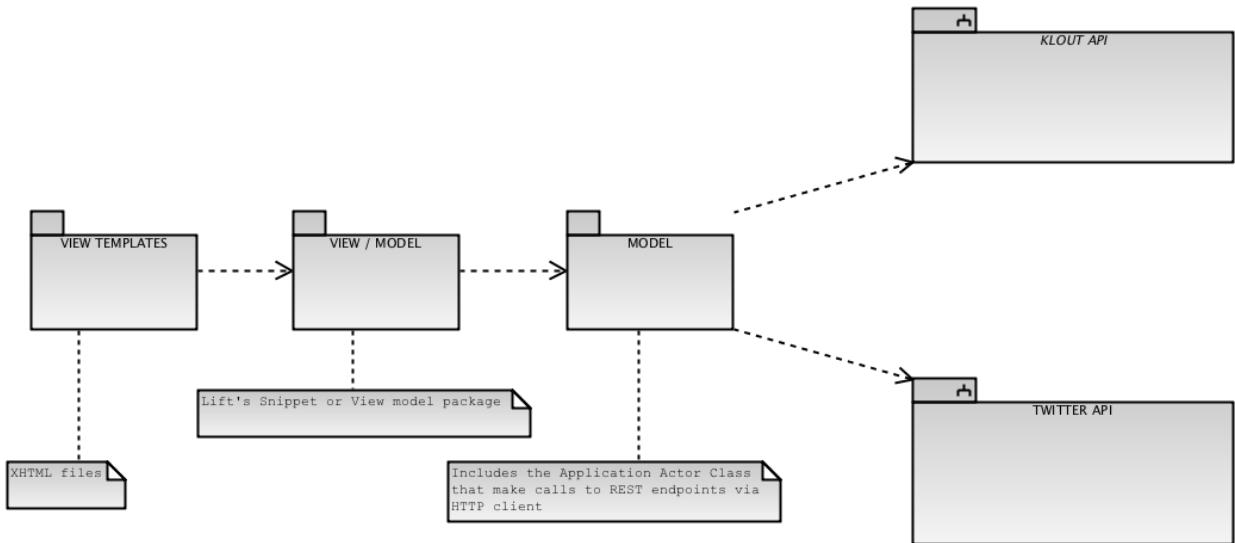
Figure 7 Layers Overview

**Application Layer:** Application Business Services and Common Services code.

**Upper Layer:** Web Server, HTTP Client to REST web services end points and other infrastructure type of software needed to support Application Layer.

**Lower Layer:** Operating System and Hardware.

**Application Packages / Modules:** Based on functionality, the application is divided into the following modules / subsystems. By identifying the interfaces between modules, it is possible then to develop the modules independently to improve productivity. The layer diagram below follows the “View First” pattern described by Lift web Framework.



**View Templates Module:** This is composed of XHTML templates files with Lift tags.

**View / Model Module:** This holds the function that render feedback to the View Template.

**Model Module:** This consists of the domain model and Actor classes.

**Twitter API Sub-System:** This accepts valid endpoint urls and replies to requests from the application appropriately.

**Klout API Sub-System:** This accepts valid endpoint urls and replies to requests from the application appropriately.

### 5.4.3 CLASS DIAGRAM

The following class diagram shows the business components along with the business object model.

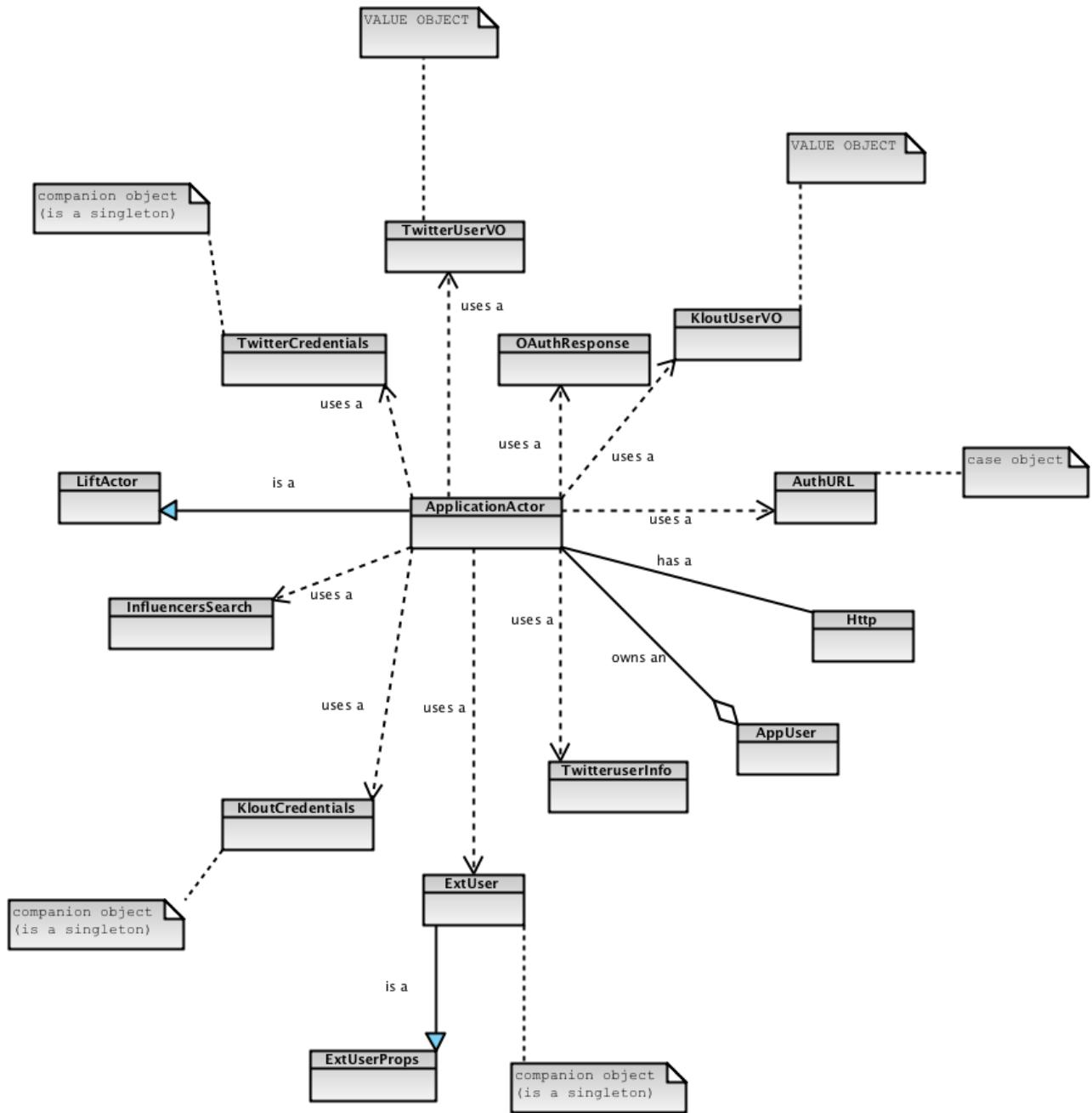


Figure 8 CLASS DIAGRAM

## 5.4.4 SEQUENCE DIAGRAM

The interaction diagram shows the flow of messages between objects. The following diagrams explain the use case realization of all the use cases.

### 5.4.4.1 Search for top influencers by keyword or topic

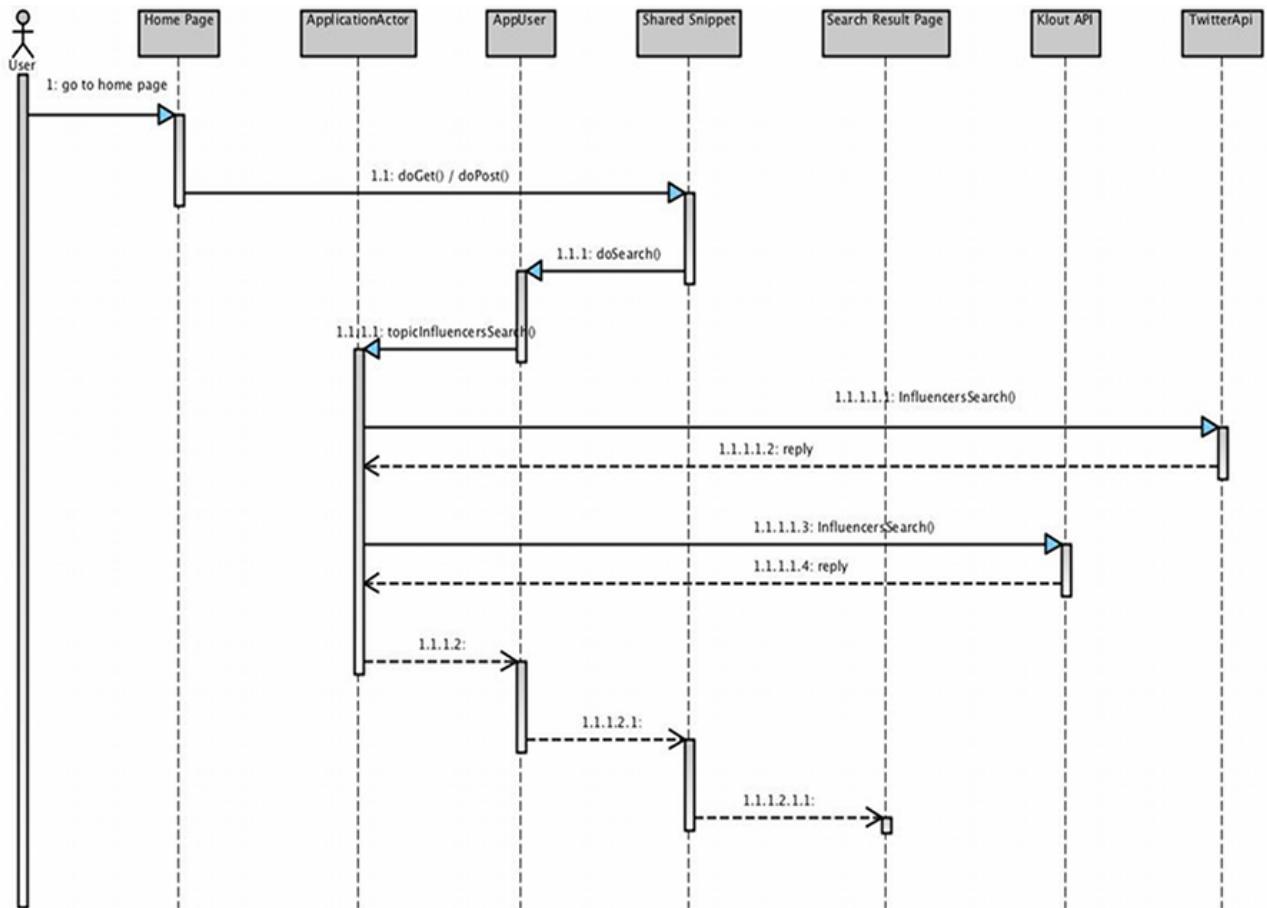


Figure 9 Sequence Diagram: Search for influencers by topic

#### 5.4.4.2 View Twitter User Influencer Report

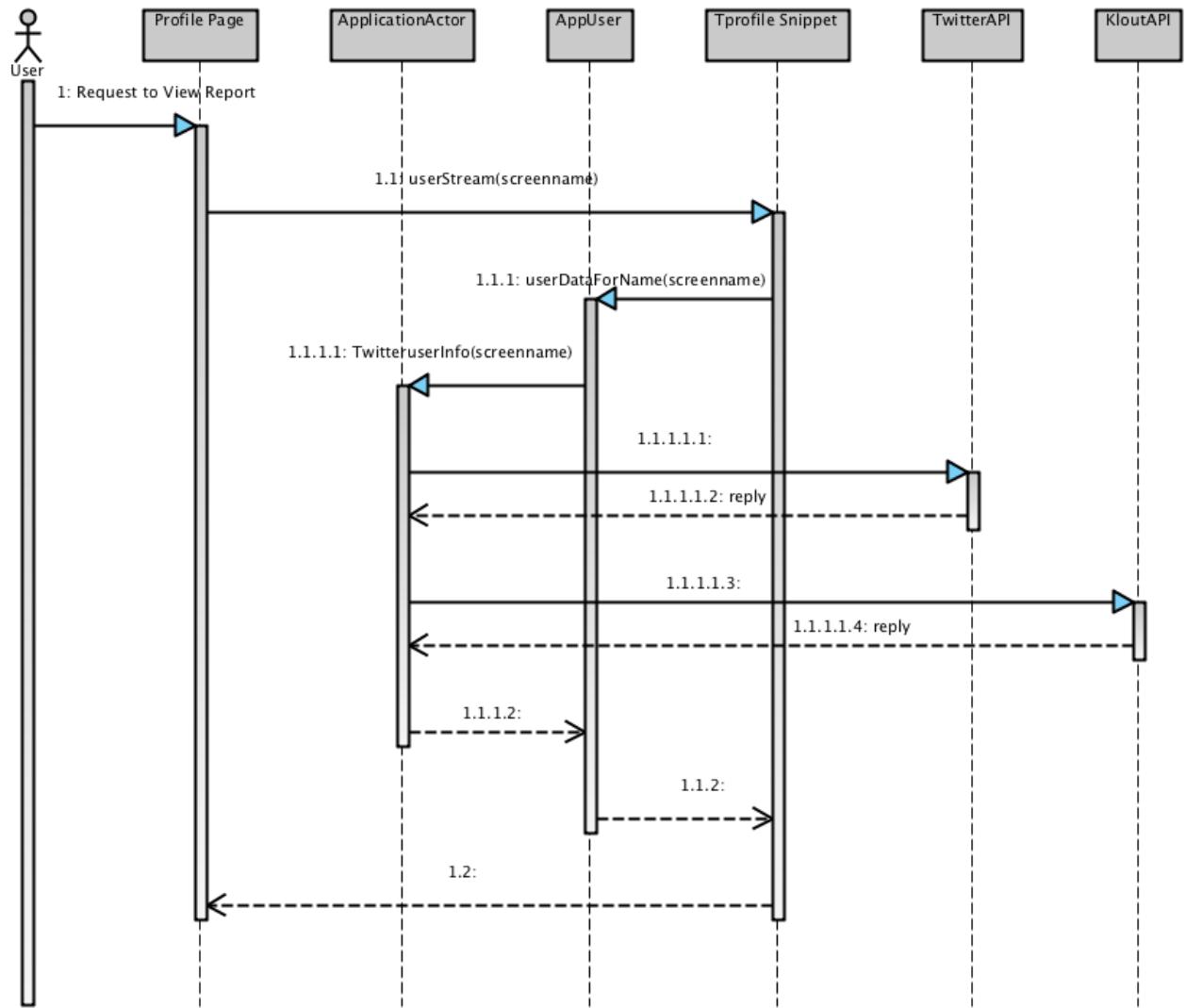


Figure 10 Sequence Diagram: View Influence Profile

#### 5.4.4.3 Search for top influencers by keyword or topic

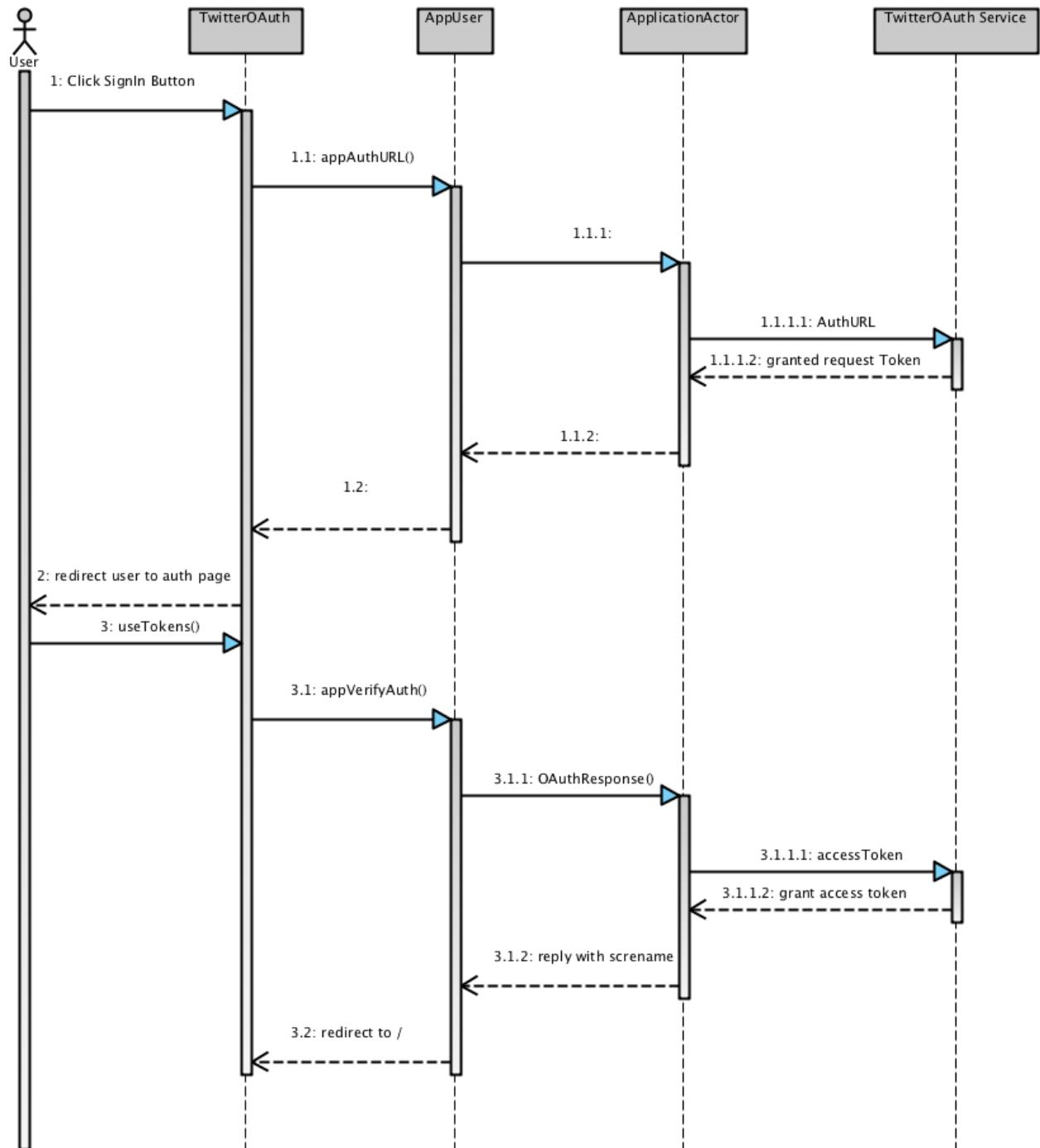


Figure 11 Sequence Diagram: Open Authorization

#### 5.4.5 DEPLOYMENT DIAGRAM

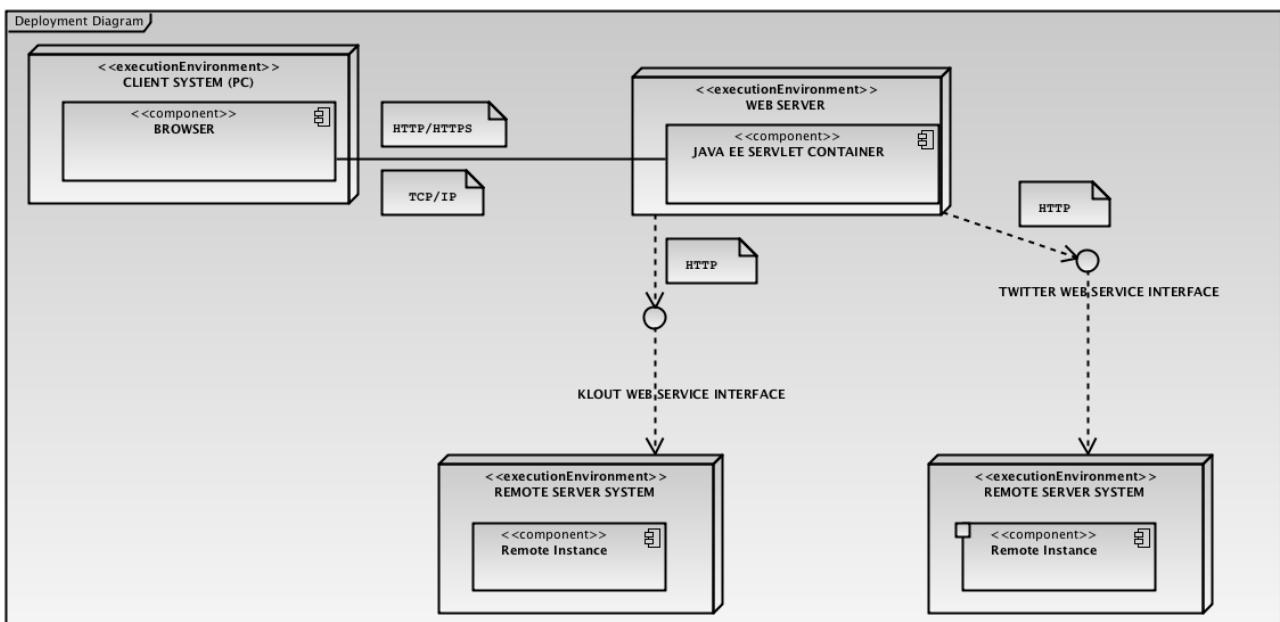


Figure 12 DEPLOYMENT DIAGRAM

#### 5.5 Iteration Planning

In line with the chosen methodology for the project, it was vital to schedule the tasks need to be achieved for each iteration before development starts. This was carried out by prioritizing the requirements list within the analysis section and allocating them to an iteration. Main features, i.e. ‘must-haves’, were selected for iteration one; with desirable features being added in iterations two and three.

Each iteration was planned for approximately thirty hours worth of work, giving ninety hours worth of development time in total. This figure was calculated with the requirements of the marking scheme in mind, whereby twenty marks out of one hundred are awarded for delivering a solution, meaning eighty hours of work for a four-hundred hour project. The extra ten hours scheduled for would include system testing, a form of evaluation.

### **5.5.1 Iteration One**

In this iteration the feature that is to be accomplished is the influencer search by topic.

#	TASKS	ESTIMATED TIME (hours)	PREREQUISITE TASK (S)	COMMENTS
1.1	PROJECT SET-UP	4	-	Created the project's build environment and IDE plugins.
1.2	Design mock-ups for the overall layout.	6	-	Use Mock up editor to create mocks.
1.3	Transfer of mock-up to XHTML and CSS	4	1.2	Create css, images and view templates files and composed them matched the mocks.
1.4	Create the snippet implementation for View template. (SharedSnippet.class)	8	1.3	Create the Snippet Class and added basic function to submit a simple search form.
1.5	Create the Supporting classes	12	1.4	Created the AppUser, ApplicationUser class and ApplicationActor class.
1.6	Test the entire flow and seek users feed back.	6	1.5	Test the web browser compatibility and entire work flow

### **5.5.2 Iteration Two**

In this iteration the feature that is to be accomplished is the twitter user Influence report page.

#	TASKS	ESTIMATED TIME (hours)	PREREQUISITE TASK (S)	COMMENTS
2.1	Review previous iteration and review class and object relationship design.	9	-	Review the last iteration and then effective changes.
2.2	Review mock-ups for the profile page	2	-	Use Mock up editor to review mocks.
2.3	Effect changes of mock-up to XHTML and CSS	2	2.2	Create css, images and view templates files and composed them matched the mocks.
2.4	Create, modify and test snippet class for View template. (TProfile.class, SharedSnippet.class and SearchSnippet.class)	11	2.1, 2.3	Add basic twitter user search functions on form submit. Display tweets of the current twitter user. Provide influence summary report, visualizations with google chart and flot for each twitter user.

2.5	Modify the Supporting classes	15	2.4	Modify and add appropriate backing functions, case class and companion object to AppUser, ApplicationUser class and ApplicationActor class.
2.6	Test the entire flow and seek users feed back.	2	2.5	Test functionality and the web browser compatibility and entire work flow

### 5.5.2 Iteration Three

In this iteration the feature that is to be accomplished is the twitter OAuth feature.

#	TASKS	ESTIMATED TIME (hours)	PREREQUISITE TASK (S)	COMMENTS
3.1	Review previous iteration and redesign class and object relationship diagram	9	-	Created the project's build environment.
3.2	Review mock-ups for the profile page	2	-	Use Mock up editor to create mocks.
3.3	Effect changes of mock-up to XHTML and CSS	2	3.2	Create css, images and view templates files and composed them matched the mocks.
3.4	Create, modify and test snippet class for View template. (TwitterOAuth.class, TProfile.class, SharedSnippet.class and SearchSnippet.class)	11	3.1, 3.3	Add a twitter oauth function to request and access tokens. Review and modify influence summary report, visualizations with google chart and flot for each twitter user.
3.5	Modify the Supporting classes	15	3.4	Add and modify appropriate backing functions, case class and companion object to AppUser, ApplicationUser class and ApplicationActor class.
3.6	Test the entire flow and seek users feed back.	2	3.5	Test functionality and the web browser compatibility and entire work flow

### 5.6 Summary

This chapter has set the basis for the intended system, with the user interface design set and the initial system design decided. While it is anticipated that these designs may change while the development well on course, the commencing designs will assist to guide the project in its beginning phases.

# Chapter Six - Implementation

## 6.1 Introduction

The purpose of the three iterations was to set-up the project development environment to give it solid base to operate on. This involves creating the build environment, install IDE plug-ins, download dependencies, designing / production of the majority of pages and class file that were required and finally testing the web pages.

## 6.2 Iteration One (Influencer search by topic)

The majority of tasks scheduled this iteration for the influencer search by topic feature are very similar, both in terms of the code needed to write their functionality and the layout of the pages to display results and forms.

These followed the design laid out in the mocks, whereby the front page contained a snippet called the ‘shared snippet’, which contained two functions that rendered a text field and a button and also provides implementation to display a list of results from the search, each with an option to view their Influencer report, along with graphical visualisation of all the Klout scores of the respective influencer gotten from the search.

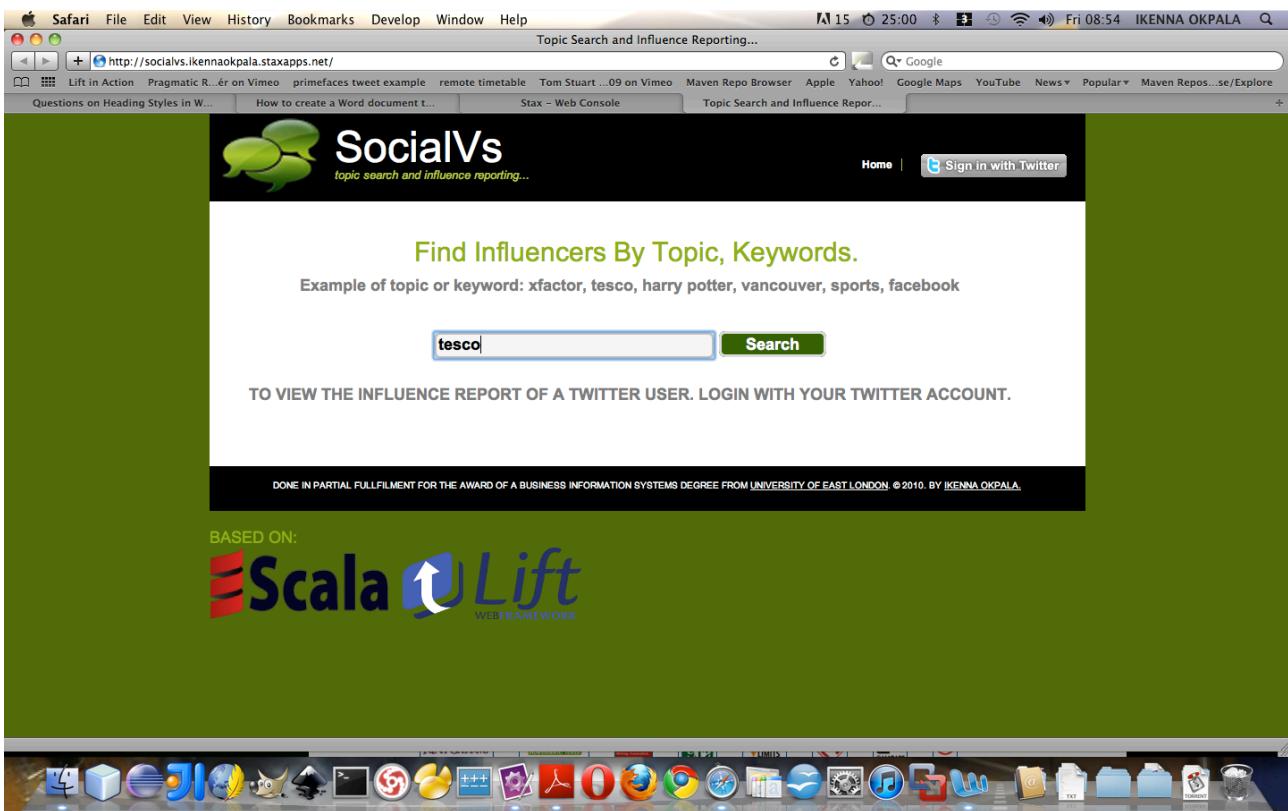


Figure 13 Front-End Page

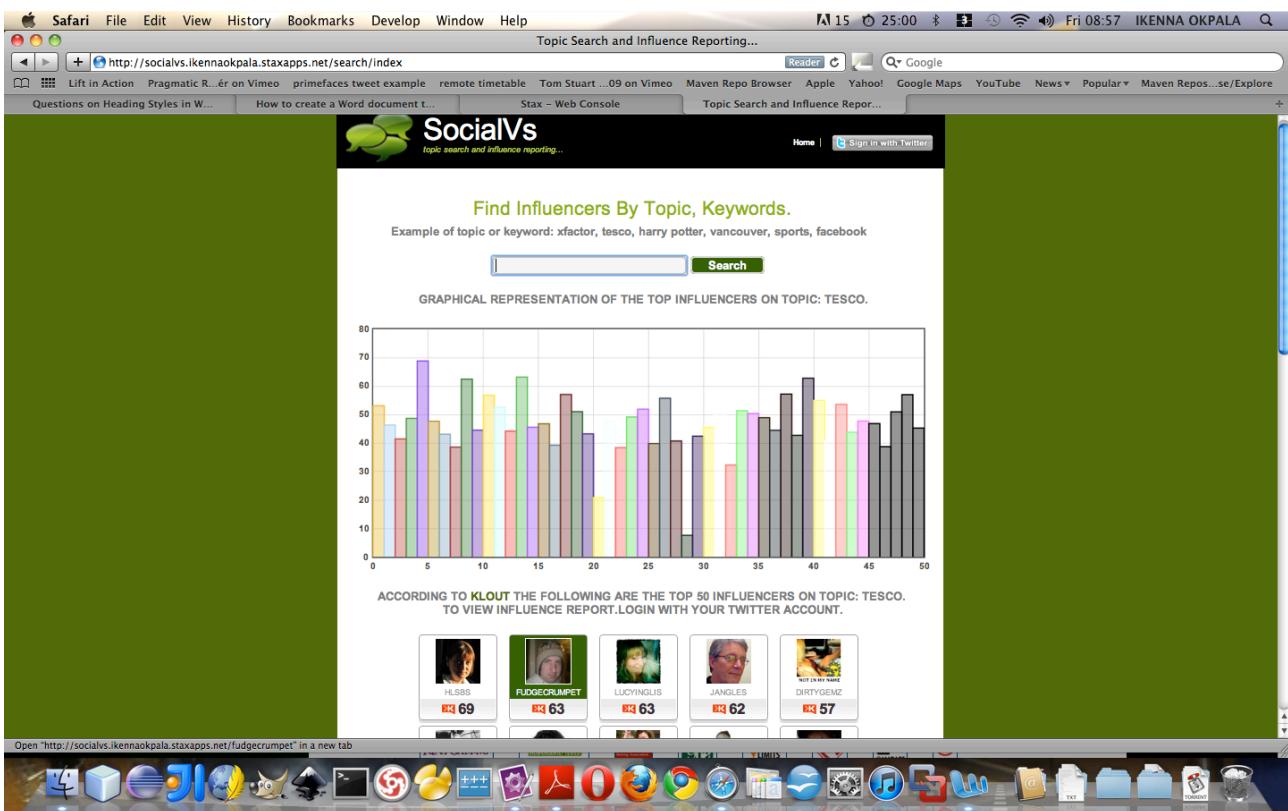


Figure 14 Influencers Search Results.

### ***6.2.1 Deviation from Iteration Plan.***

For the most part, the iteration went as planned, with all of the features scheduled for completion being completed. However, time became a major issue; the total time to complete iteration one way exceeded the time allocated to it. This was due to additional time required to become more familiar with handling and parsing JSON data from twitter and Klout APIs.

Due to the allocated development time being exceeded, a user feedback session that was scheduled for the completion of iteration one had to be cancelled to ensure that the project remained on schedule as per the overall project plan. Whilst this may have caused additional work down the line from future feedback sessions, it was envisaged that due to the system being in its infancy, with only basic features implemented, user feedback would not identify many, if any, major changes that needed to be carried out.

## **6.3 Iteration Two (twitter user Influencer report page)**

With the basics of the influencer search side of the system complete, iteration two was scheduled to focus on improving functionality of the ApplicationActor and it's associated objects / classes and in effect developing the Influencer report page, but in the most part to create a front-end for this to display the information required.

The plan for iteration two remained unchanged at the outset, owing to the completion of all features planned in iteration one, and also the fact that the planned user feedback session was not carried out, meaning no changes from the user feedback needed to be factored in, only slight spontaneous changes were noticed.

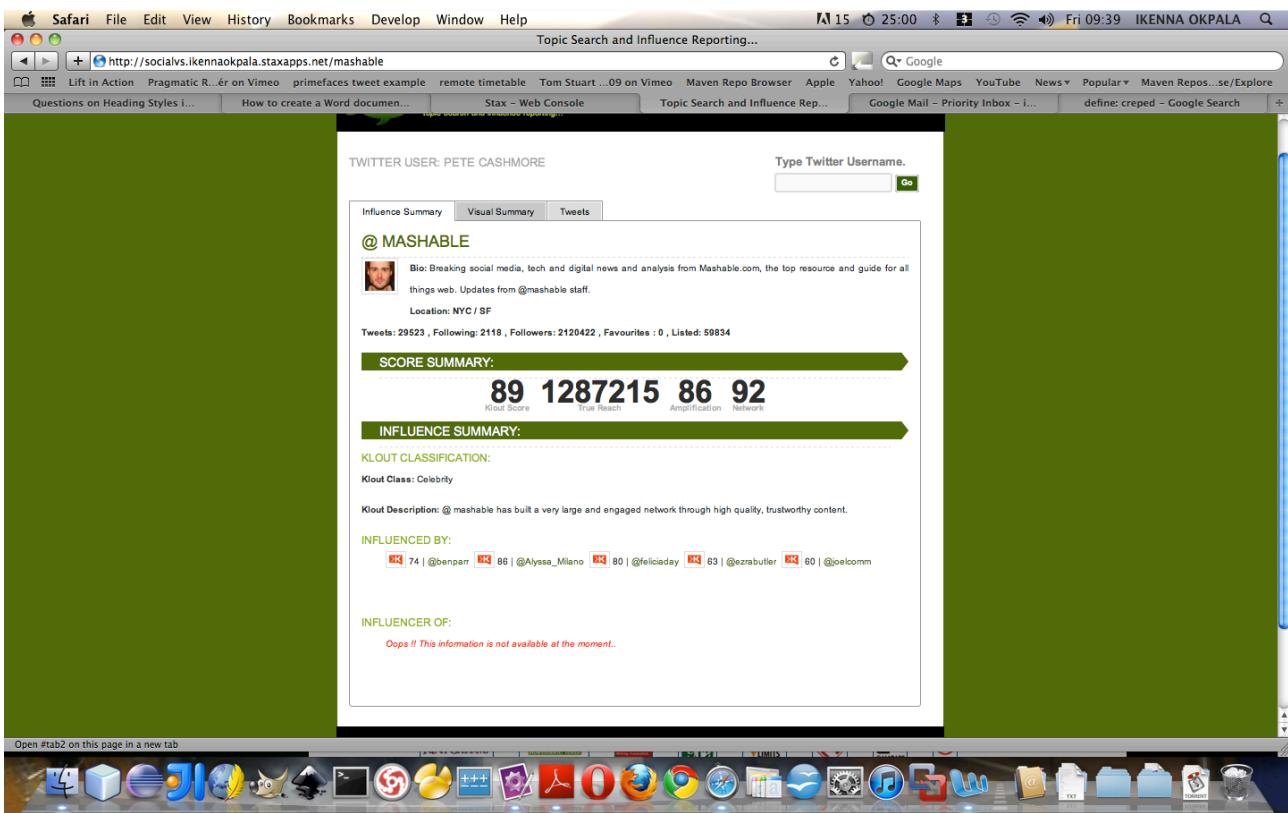


Figure 15 Influence Summary Reports

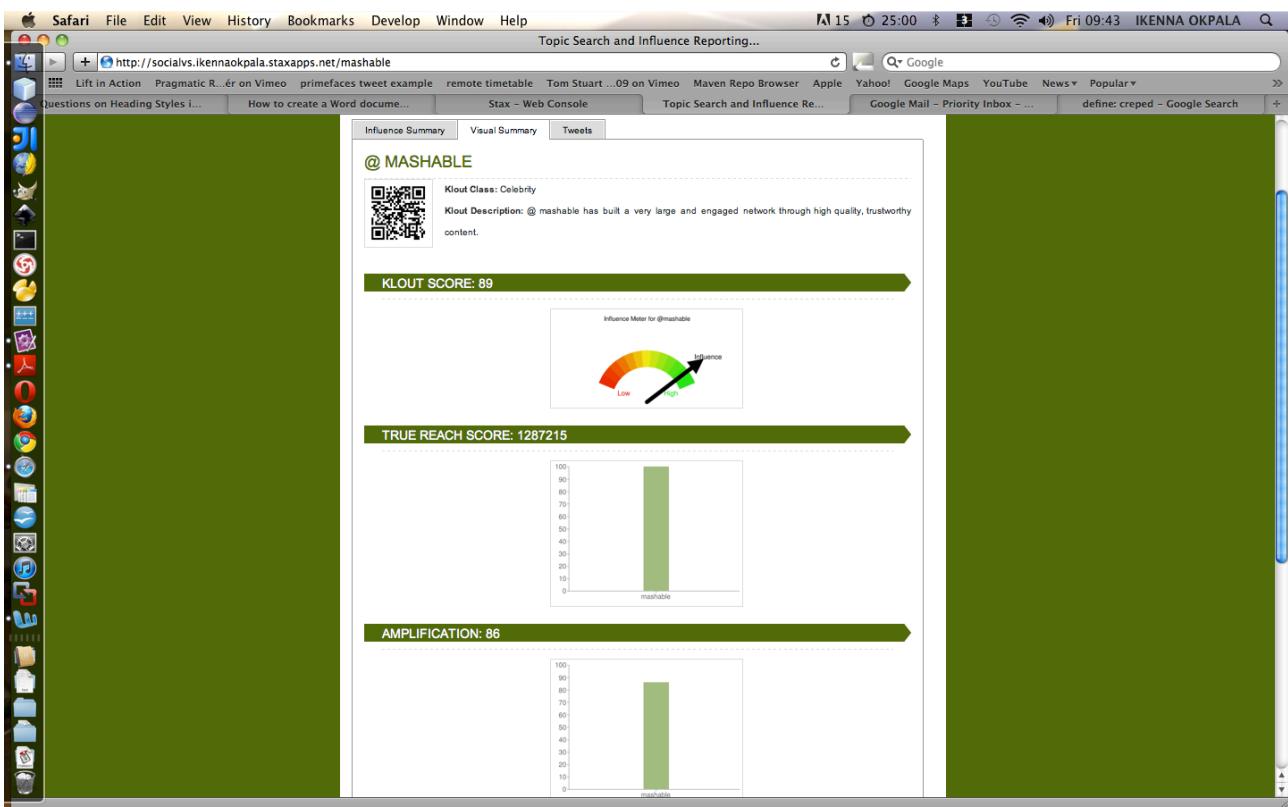


Figure 16 Visual Summary Reports

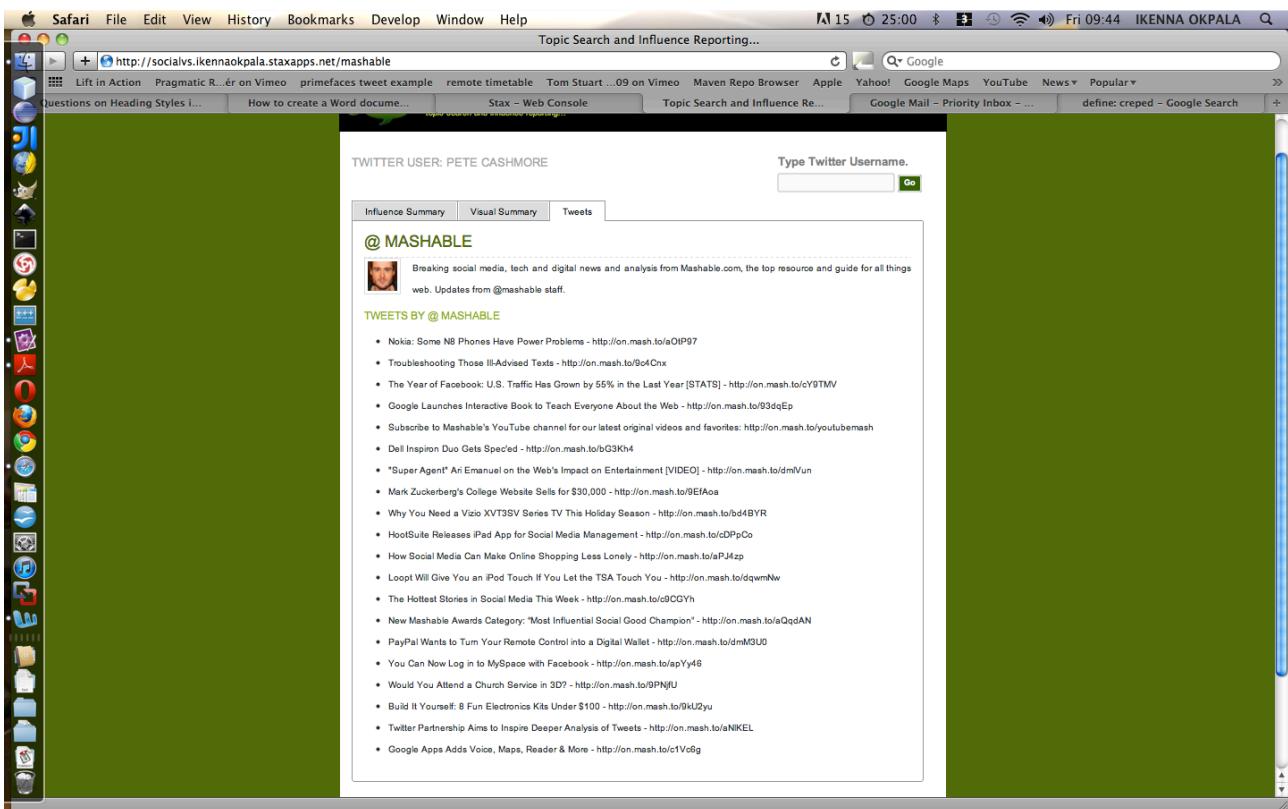


Figure 17 Tweets For the Twitter User

### 6.3.1 Deviation from Iteration Plan.

Overall, the schedule for iteration two was followed and the majority of features were completed on time. However, the graphing was greatly hampered by runtime response with status ‘code 500 Internal Server Error’, this meant that klout had problems with a few of their statistical end points.

This lead to use of column chart to be used to display Amplification, Network and True Reeach score as opposed to the earlier planned use of Annotated Time Line graphs to show the respective scores over a period of 14 days, this feature is part of the ‘should have’. Authorization was another feature planned for this iteration, but was move to the next iteration.

### 6.3.2 User Feedback

#### 6.3.2.1 Usability Observations

From the skype conversation with project consultant Jane, it was clear that the design was usable as only minor issues were raised with locating all the influencers at once. This is because her computer screen size is 12” wide only big enough to capture a few influencers on her screen This would be

common with any system of this type when used for the first time until the user becomes familiar with its operation, so these minor issues were not a concern.

### 6.3.2.2 Functionality Observations

The feedback session with project consultant Emeka at the end of iteration two yielded a major suggestion for improvement. The first suggestion was that as the search by topic did not always return fifty influencers for most of the topic he searched for. This is contradicts what was recorded in the API documentation on the Klout web site. Therefore for the next iteration this will be dynamically evaluated to improve the functionality.

Other than the corrections from Emeka, he was pleased with the basic functionality of the web application; in particular he like the thumbnail presentation of the influencers and said the QRCode barcode was an excellent addition.

## 6.4 Iteration Three (OAuth)

Iteration three, the final iteration scheduled in the scope of this project, dealt with changes to features from the previous iterations and primarily the OAuth feature which would add value to the system.

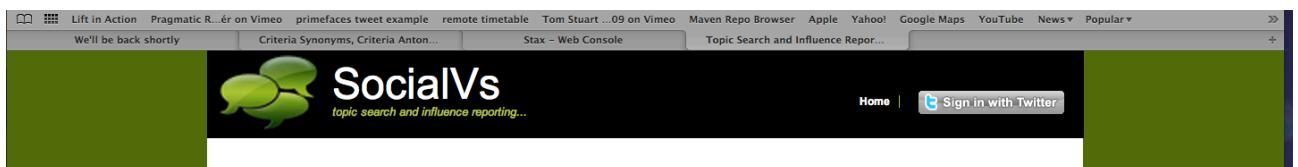


Figure 18 Before Twitter button is clicked

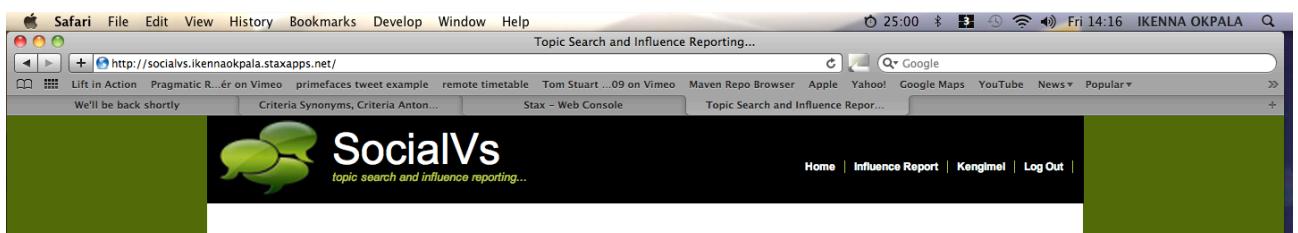


Figure 19 After Twitter button has been clicked

### ***6.4.1 Deviation from Iteration Plan.***

The scheduled plan for iteration three was followed and the feature was completed on time, having time to do some more enhancement on other areas of the project. However, during implementation it was noticed that the button delayed the page from loading and a few other quirky behaviors.

This lead to a sight change to how the class properties and functions were eagerly evaluated my the scala compiler, hence the intorductions of the private and lazy keywords to some parts of the project. These modifier avoids the class fields from being eagerly evaluated by the complier.

## **6.5 Summary**

Iteration three introduced a number of better design and programming concepts to the requirements to the system. The OAuth functionality was review and made to work better along with a review on using google chart API for graphing and the flot graphing library of the lift web framework for statistics. Whilst a number of other design concepts were added, a number of features had benefited from this iteration and this iteration kept within the planned schedule with a lot of velocity.

# Chapter Seven - Evaluation

## 7.1 Introduction

The evaluation criteria for the project were set in the introduction of this report. They combine both criteria, which evaluate the the project as a whole. This chapter will delve into each of the criteria in turn, concluding with an overall assessment of the project's success.

## 7.2 Features Implemented

The analysis section of this report set out a list of features, which the system 'must have', 'should have', 'could have' and 'won't have'. The number of these features implemented and their respective functionality will provide some indication as to the success of the system.

The system at the conclusion of the third iteration was feature-rich, with many of the planned features being implemented. With the web application split into three sections, the features implemented in each are discussed in turn:

### 7.2.1 Influencer Search by Topic

The main function of this feature is to search for top influencer based on the topic or key word entered in tot the text. This is part of fundamental 'must have' features, with the graphical visualizations of the klout score for respective influencer of from the same topic search. This however was included to increase its usability with the use of a colour separating each score on the graph.

The result when retrieves from klout an is also sorted from Highest to Lowest according to the klot scores, which is also part of the 'must have' features.

### 7.2.2 Influencer Report

The influencer report area allowed for users to get a further insight on a twitter user that appear relevant to the particular search in question. Of the 'must-have' features, this feature's operations were implemented as mentioned, along with influence summary report and visual summary report for the respective or currently examined twitter user.

The ability to provide further statistical information with the aid of an annotated time line was however not included due to response problems from the internal klout server, though this ‘should have’ features but had to be abandoned due to time constraints and lack of response to emails sent to klout.

The rest of the ‘should have’ and ‘could have’ features were not also implemented due to time constraints.

### ***7.2.3 Open Authorization (OAuth)***

As with the other two features, all of the planned ‘must haves’ were implemented into the system; the front-end had separate areas for different club teams, along with displaying fixtures, results, news, events, squads and statistics. A detailed match page, incorporating a pitch showing the match line-up with profile thumbnails allowed for users to be presented with clear information relating to matches.

However none of the ‘should have’ and ‘could have’ features was implemented in this iteration.

## ***7.2 Conclusion***

On the whole, the number of features implemented provided a feature-rich system. The fact that every ‘must have’ feature was implemented meant that the system incorporating all of the fundamental features required.

## ***7.3 Non-Functional Requirements***

### ***7.3.1 Usability***

From the usability responses obtained from user observations, it became apparent that the system is usable. The consultant mostly commented on the thumbnail presentation style from the influencers search. This thumbnail display the user’s information in a concise manner, so users can easily filter through based on klout scores. Also the graph gives an overall view of the klout scores on a particular topic area, this is as it was reported by the consultant.

On the whole, the system tried to maintain a uniform feel. It avoided getting cluttered with too many elements on a page. Though vertical scrolling was unavoidable due to the elements that required to be spaced out, to ensure information is displayed clearly to the user.

## 7.3.2 Compatibility

Compatibility was tested with the web application's compliance to W3Cs standards for strict XHTML and CSS and cross browser compatibility testing.

### 7.3.2.1 XHTML and CSS validation

In order to validate the XHTML compliance, the web pages were tested with the W3c validation tool, the front-end screen for search was used for the test, with the results shown below:

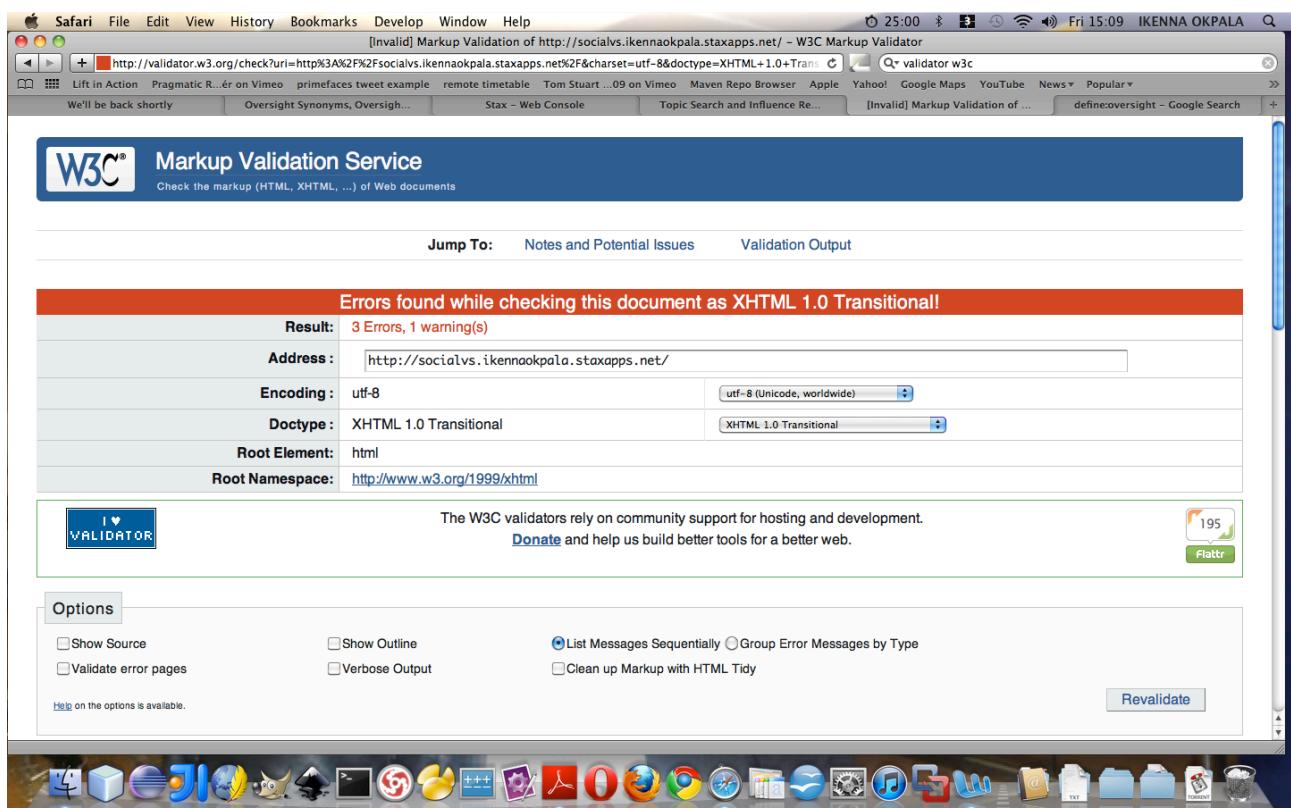


Figure 20 W3C XHTML TEST

Three errors however were found this was as a result of the automatic generation of the field ids from the lift web framework. The reason why lift does this is to avoid web form replays by a hacker on the web. This makes the form even more secure and hard to hack. For this project we have considered this area to be overlooked. This test also goes a long way to suggest that this web application will appear the same way across a majority of the well known and used web browsers.

## 7.4 Development Methodology, Time Management and Planning

The methodology chosen dictated the planning and time management required, whereby planning was to be carried out down to the level of individual-features through prioritization. This was an effective way of working as it allowed the development of the system to be broken down into achievable goals of implementing features.

The all the planned ‘must have’ features were implemented, so planning proved to be effective. However, the underestimation of time required to pick up the skills required in developing in Scala meant that valuable evaluation had to be foregone in iteration one, and total development time was increased significantly.

On the other hand, the problems encountered in iteration one were the only issues in a plan which otherwise ran smoothly in the most part, with the odd feature not being implemented as planned. This was one of the features of using the selected methodology, however, with the ability to reprioritize and reschedule features into future iterations possible.

In conclusion, the methodology was generally a success, with time being managed and planned well. The familiarity with the problem domain meant less interaction was required with the two project consultants, which if was not the case, a more agile methodology would need to be chosen; something to keep in mind for future development, for instance if the system were to be expanded to more unfamiliar grounds in social media analytics.

On the whole, the project ran to the overall schedule reasonably well once the schedule was revised to take into account the time required to become familiar with the chosen development tools. More time possibly should have been allocated to writing the final report, as this took longer than expected, but other than that, the scheduling for the project was a success.

## 7.7 Conclusion

All-in-all, the project is a success, having created a social media web based system. The critical success factor for the project was most likely the use of the web framework, Lift web framework; without it, time would have been spent performing repetitive, routine tasks as opposed to developing innovative features for clubs to use. The only area of evaluation in which the project struggles is its comparison to existing systems, which already exist. However, the system can compete on its usability even in its infancy.

## 7.8 Future Development

Despite the sheer number of features developed, there is plenty of room for future development. Especially possible addition of new features, improvement of existing features (like historic data or timeline analysis) and inclusion of a mobile version of the web application, which were defined in ‘should haves’ and ‘could haves’ in chapter four.

## 7.9 Summary

The concluding this chapter of the report has assessed the success of both the system, and the project as a whole, against a set of pre-set criteria in chapter one, in which it fared quite well. The chapter was completed with suggestions for further development of the system, including mobile version of the software, along with the addition of historic data or timeline analysis.

## Reference:

- Bruegge B, Dutoit H., (2004), Object-Oriented Software Engineering using UML, Patterns and Java, Second Edition. Inc: Pearson Education Publishing, NJ
- Kaplan A. M. and Haenlein M., (2010), Users of the world, unite! The challenges and opportunities of social media, Business Horizons, Vol. 53, Issue 1, p. 59-68.
- Odersky M. et al, (2008) Programming in Scala First Edition, Artima.
- Kappel G. et al., (2006), Web Engineering. Chichester: John Wiley & Sons,.
- Nielsen J., (2000), Designing Web Usability: The Practice of Simplicity. Berkley : New Riders Publishing.
- Nielsen J. and Loranger H., (2006), Prioritizing Web Usability . Berkley : New Riders Publishing.
- Oreilly T. (2005) What Is Web 2.0, Available at: <http://oreilly.com/web2/archive/what-is-web-20.html>, [Accessed 18<sup>th</sup> June 2010]
- Perrett T., Available at: [http://www.manning.com/perrett/MEAP\\_CH\\_01.pdf](http://www.manning.com/perrett/MEAP_CH_01.pdf) [Accessed 12<sup>th</sup> October 2010]
- Shirky C., (2009): How social media can make history, Available at: [http://www.ted.com/talks/clay\\_shirky\\_how\\_cellphones\\_twitter\\_facebook\\_can\\_make\\_history.html](http://www.ted.com/talks/clay_shirky_how_cellphones_twitter_facebook_can_make_history.html), [Accessed 9<sup>th</sup> June 2010]
- Sommerville, I., (2006), Software Engineering (7th Edition) . Harlow : Pearson Education Ltd.
- Subramaniam V., (2008), Programming Scala, The Pragmatic Programmers Bookshelf.
- Surowiecki, J., (2004). The Wisdom of Crowds, Doubleday Anchor Books
- Wampler D., (2008), Available at: <http://blog.objectmentor.com/articles/2008/08/03/the-seductions-of-scala-part-I>, [Accessed 11<sup>th</sup> September 2010]

## Bibliography:

- Bocij P., Greasley A., and Hickie, A. (2008), Business Information Systems - Technology, Development & Management (4th edition). Essex: Pearson Education Limited.
- Hunt, J., (2006), Agile Software Construction. London : Springer-Verlag.
- Pilone D., and Miles R., (2008), Head First Software Development. Sebastopol: O'reilly Media Inc.