`

YONG CHANG YI B.TECH (HONS) INFORMATION & COMMUNICATION TECHNOLOGY SEPT 2012

A TWITTER-INTEGRATED WEB SYSTEM TO AGGREGATE AND PROCESS EMERGENCY-RELATED DATA

YONG CHANG YI

INFORMATION & COMMUNICATION TECHNOLOGY
UNIVERSITI TEKNOLOGI PETRONAS

SEPTEMBER 2012

`

A Twitter-Integrated Web System to Aggregate and Process Emergency-Related Data

By

Yong Chang Yi

Dissertation submitted in partial fulfillment of

the requirements for the

BACHELOR OF TECHNOLOGY (Hons)

(INFORMATION & COMMUNICATION TECHNOLOGY)

SEPTEMBER 2012

Universiti Teknologi PETRONAS

Bandar Seri Iskandar

31750 Tronoh

Perak Darul Ridzuan

`

CERTIFICATION OF APPROVAL

A Twitter-Integrated Web System to Aggregate and Process Emergency-Related Data

By

Yong Chang Yi

A project dissertation submitted to the
Information Technology Programme
Universiti Teknologi PETRONAS
in partial fulfillment of the requirement for the
BACHELOR OF TECHNOLOGY (Hons)
(INFORMATION & COMMUNICATION TECHNOLOGY)

Approved by,

____

(Prof. Dr. Alan Oxley)

UNIVERSITI TEKNOLOGI PETRONAS
TRONOH, PERAK
September 2012

`

CERTIFICATION OF ORIGINALITY

This is to certify that I am responsible for the work submitted in this project, that the original work is my own except as specified in the reference and acknowledgements, and that the original work contained herein has not been undertaken or done by unspecified sources or persons.

____

(YONG CHANG YI)

`

ABSTRACT

A major challenge when encountering time-sensitive, information critical emergencies is to source raw volunteered data from on-site public sources and extract information which can enhance awareness on the emergency itself from a geographical context. This research explores the use of Twitter in the emergency domain by developing a Twitter-integrated web system capable of aggregating and processing emergency-related tweet data. The objectives of the project are to collect volunteered tweet data on emergencies by public citizen sources via the Twitter API, process the data based on geo-location information and syntax into organized informational entities relevant to an emergency, and subsequently deliver the information on a map-like interface. The web system framework is targeted for use by organizations which seek to transform volunteered emergency-related data available on the Twitter platform into timely, useful emergency alerts which can enhance situational awareness, and is intended to be accessible to the public through a user-friendly web interface. Rapid Application Development (RAD) is the methodology of choice for project development. The developed system has a system usability scale score of 84.25, after results were tabulated from a usability survey on 20 respondents. Said system is best for use in emergencies where the transmission timely, quantitative data is of paramount importance, and is a useful framework on extracting and displaying useful emergency alerts with a geographical perspective based on volunteered citizen Tweets. It is hoped that the project can ultimately contribute to the existing domain of knowledge on social media-assisted emergency applications.

`

ACKNOWLEDGEMENTS

First and foremost, the writer would like to take this opportunity to express his greatest gratitude and appreciation to project supervisor, Prof. Dr. Alan Oxley, who had continuously monitored his progress throughout the duration of the project. His constructive comments, advice, and suggestions have guided the project towards its successful outcome.

The utmost gratitude is also expressed towards Universiti Teknologi PETRONAS (UTP), especially the committee of the Final Year Project of the Computer Information Sciences (CIS) department for their excellent organization and management of this course.

Last but not least, the writer would like to thank all the respondents which participated in the survey initiated by the writer as part of the project.

`

TABLE OF CONTENTS

`

`

LIST OF FIGURES

LIST OF TABLES

`

ABBREVIATIONS AND NOMENCLATURES

API	Application Programming Interface
RAD	Rapid Application Development
GIS	Geographic Information System
SDLC	System Development Life Cycle
ICT	Information and Communications Technology
TED	Twitter Earthquake Detector
FYP	Final Year Project
REST	Representational State Transfer
SUS	System Usability Scale
AJAX	Asynchronous JavaScript and XML
PHP	Hypertext Preprocessor
HTML	HyperText Markup Language

`

CHAPTER 1
INTRODUCTION

1.1 Background

The increasing pervasiveness of information and communications technology has radically transformed how people receive, access, and communicates information. One of the prominent trends which have contributed to said phenomena has been the advent of social media platforms. Kaplan and Haenlein (2010) define social media as a group of Internet-based applications that build on the ideological and technological foundations of Web 2.0 that allows the creation and exchange of user generated content. The rising influence of these social media platforms has led to concerted studies on ways to leverage them for non-frivolous applications.

The web-based micro blogging service Twitter is a prime example of a social media platform whose usage has proliferated beyond casual social networking. The Twitter platform allows users to post messages, named as 'Tweets', consisting of text not more than 140 character, and subscribe to other users to view their Tweets as well. The rapidity, accessibility and ubiquity of Twitter are significant enablers of information exchange during time-sensitive and information-critical situations. A domain in which timeliness of data is of extreme importance is emergency communications; there has been recognition of the potentially significant role of Twitter in the emergency domain as reflected by increased efforts to involve Twitter in emergency management efforts. Emergent uses of Twitter in emergencies occurred during the 2007 Southern California wildfires, in which San Diego residents posted rapid-fire updates to Twitter on evacuation, meeting points, places of shelter and so forth (Poulsen, 2007). The trend of Twitter utilization in times of emergency was also illustrated in the Haiti Earthquake, where Twitter became a platform for the dissemination of information on relief efforts (Ali, 2010).

In addition to being a mere channel for the straightforward peer-to-peer transmission of situational updates, Twitter has been explored as a tool to crowd-source emergency-related Tweets to enhance situational awareness during

`

emergencies (Vieweg, Hughes, Starbird & Palen, 2010). Studies have been conducted on the possibilities of harnessing volunteered emergency data on Twitter in a more beneficial, structured manner. These studies vary from leveraging the embedded geo-location features of Twitter in producing visualizations of emergency situations (MacEachren et al, 2011) to the suggestion of a standardized format for tweet messages to support citizen data reporting during emergencies (Starbird & Stamberger, 2010), with the overarching goal of innovating the combined applications of Twitter and emergency communications.

This project provides a proof of concept model on how Twitter can be leveraged in emergency communications, by developing a Twitter-integrated web-system which will aggregate and process emergency-related data on the Twitter platform in order to provide a geo-visualization of emergency situations.

1.2 Problem Statement

A major challenge when encountering time-sensitive, information critical emergencies is to immediately source raw volunteered data from on-site public sources, which can be leveraged as emergency alerts to enhance emergency awareness from a geographical context. Traditional media avenues such as television, newspapers, radio and short messaging systems are useful platforms to passively retrieve and disseminate/circulate emergency-related information to the community, but remain impractical as a means of rapidly collecting valuable emergency-based data, thus failing to deliver information representative of an emergency situation in a timely manner.

A solution which is able to automate the aggregation of emergency-related tweet data available on social media platforms and organize it to produce an emergency-specific informational account with a geographical perspective is required.

`

1.3 Objectives

The development of the proposed system desires to achieve the following objectives:

- Aggregate emergency-related situational data from "Tweets" by public citizen sources on the Twitter platform which fulfils a prescribed syntax format.
- Process the collected tweet data based on geo-location and syntax into organized information entities relevant to an emergency.
- Display the information on a map-like interface.

1.4 Scope of Study

The scope of tweet data to be collected is Tweets which satisfy a pre-determined hashtag-based syntax on a particular emergency, in which the format is prescribed by an emergency management organization. The system seeks to query the Twitter API based on the aforementioned syntax, and does not intend to delve into sentiment analysis or sense-making for tweet content, nor is there any automated verification of the truthfulness of the supplied citizen account or the correctness of geographical information within the tweet.

The system is generally targeted for use by emergency management organizations which seek to retrieve and organize volumes of volunteered emergency-related data available on the Twitter platform, and whilst it can potentially be a tool to support formal decision making processes in aspects of emergency management, as an emergency happens, it is not applicable to pre-emergency preparation, post-emergency recovery or policy communication. The system would also be accessible to public users wishing to geo-visually track emergency situations via a web interface.

`

CHAPTER 2
LITERATURE REVIEW

2.1 Overview

To facilitate understanding of literature related to this project, the research materials examined have been divided into three discrete sections which are concerned with the leveraging of social media in an emergency domain, examples of Twitter use during emergencies and ways to collect and process emergency-related data from Twitter.

2.2 Leveraging Social Media in the Emergency Domain

In examining the use of social media in the context of emergencies, there is a need to understand how social media technologies are specifically harnessed under such situations. In a study of the emergent uses of social media during the 2007 Southern California Wildfire, Sutton, Palen and Shklovski (2008) suggests that social media supports backchannel communications which is collectively resourceful and allows for generation of information which otherwise may not be easily obtained. Sutton et al also notes that stress reactions which results from cases of information dearth can serve as strong catalysts for citizens to proactively find and share information on emergency situations.

Van Leuven (2009) argues that Web 2.0 technologies improves timeliness and access to emergency information, and has the positive attribute of providing greater context. This is in stark contrast to traditional media outlets, which struggles to deliver immediate emergency information in a format which matches citizen needs. A new information pathway which connects official public information office functions with members of the public has emerged due to the need of an organizational destination for citizen-provided data in the context of heightened ICT-related public participation during crisis events (Palen & Liu, 2007). These are indicative of an increasingly prevalent trend of co-opting the social media in sourcing emergency-specific information.

`

2.3 Twitter Use During Emergencies

Twitter has been described as an avenue to provide information not covered on radio and television, such as details and first-hand accounts in the immediate aftermath of an event (Mills, Chen, Lee, & Rao, 2009). Mills et al acknowledge that Twitter is not a complete emergency communications system which can perform management functions, but nevertheless, organizations working at disaster mitigation can potentially obtain highly specific information regarding emergencies. Mills et al continue to propose that Twitter has the ability to, at the very least, function as a complementary tool during time and information-critical situations. While the broader Twitter audience tends to be interested in high-level accounts of an emergency event, re-Tweets of locals affected by an emergency have a tendency to be more specific, which can assist in the propagation of locally relevant information (Starbird & Palen, 2010).

An example of an existing application which leverages Twitter in the emergency domain is the Twitter Earthquake Detector (TED), a software application supported by the U.S Geological Survey which seeks to mine real-time tweet data for indicators of earthquakes (http://recovery.doi.gov/press/us-geological-survey-twitter-earthquake-detector-ted/). Although the focus of this particular research does not lie in pre-hazard preparation, TED nevertheless showcases the rich possibilities afforded by social media in the context of integration with the emergency domain, such as the ability to pool together and make readily accessible citizen contributed information, as well as improving public outreach with regards to timely on-the-ground information (Guy, Earle, Ostrum, Gruchalla & Horvath, 2010).

However, Guy et al notes the limitations of using Twitter as a citizen reporting system such as a potential lack of Tweets with quantitative properties. The presence of quantitative data is necessary in order to produce tactical information which has high utility during emergencies. Naturally, concerns will be raised on the legitimacy and validity of data sourced from citizens during emergencies. Mendoza, Poblete and Castillo (2010) studied the propagation of confirmed truths and false rumors on Twitter in which the results indicated that false rumors tend to be questioned much more than confirmed truths. Similarly, any organization which

`

chooses to co-opt social media data as a source of emergency information has a responsibility to state if the information has been corroborated.

2.4 Collecting and Processing Emergency-Related Data from Twitter

It is necessary to develop an understanding of potential strategies which can be applied in the collection and computational processing of Twitter data in order to maximize the utility of extracted emergency-related information. In an examination of hazard-based data sets on Twitter, Vieweg, Hughes, Starbird and Palen (2010) developed an outline of microblog-enhanced situational features for emergencies such as preparation (pre-warning), warning (hazards-focused), responses to warning, (post-hazard), or advice which seeks to inform the development of systems which can support different informational stages in emergency management. Vieweg et al proceeds to caution that the design and implementation of software systems that employ information extraction strategies during emergencies be informed by analysis of specific features of information generated under such circumstances. This is a pertinent point which has to be taken into consideration, as automatic methods to process tweet data during emergencies may vary depending on the type and timeline of emergencies involved. In the context of this research, information extraction is focused on hazards-focused data which is provided on the immediate aftermath of an emergency. Vieweg et al (2010) also state that geo-location, location referencing and situational update data are prevalent entities in Tweets on emergencies. The presence of such entities tends to aid automated retrieval of relevant information in a specific emergency event.

Geographic location is an essential element to inform situational awareness, and arguably a basic building block of emergency management efforts as a whole. The presence of a built in geo-location function on Twitter means that Tweets have the capacity to be location-precise, but the drawback is that the enablement of the function is at the discretion of individual users. To enhance the input provided by Twitter during emergencies, MacEachren et al (2011) suggests various alternative ways in which less explicit geographic information may be extracted, such as hash tags, locations stored in user profiles, and location references in tweet content. The high volume and noisy nature of data provided by people on the ground during emergency events poses a sense-making problem, but instead of training computer algorithms to extract emergency-related information from Tweets, an alternative approach to “train” Twitter users to craft such Tweets in a machine-readable format is suggested (Starbird & Stamberger, 2010). Starbird and Stamberger outlined a model of citizen sensing which places a key emphasis on parsing and processing user-structured messages which originate from the Twitter platform.

Boyd (2010) proposes that emergency message syntax should be structured in such a way that it consists of only defined elements relevant to the emergency itself. **Figure 2.1** illustrates examples of the proposed tweet structure. This research adopts the assumption that citizens are capable of being trained to use emergency-specific Twitter hashtag syntax, which allows them to upload information in a machine-readable form. By marking-up Tweets based on a prescribed format which includes entities relevant to situational awareness in emergencies, computational filtering and classification of such tweet data will be greatly simplified.

EXAMPLE1: #haiti #imok #name John Doe #loc Mirebalais Shelter #status minor injuries
EXAMPLE2: #haiti #need #transport #loc Jacmel #num 10 #info medical volunteers looking for big boat to transport to PAP
EXAMPLE3: #haiti #need #translator #contact @pierrecote
EXAMPLE5: #haiti #ruok #name Raymonde Lafrotune #loc Delmas 3, Rue Menelas #1
EXAMPLE4: #haiti #ruok #name Camelia Siquineau #loc Hotel Montana
EXAMPLE6: #haiti #offering #volunteers #translators #loc Florida #contact @FranceGlobal

Figure 2.1: Examples of Emergency-Specific Tweet Syntax (Source: Boyd, 2010)

`

CHAPTER 3
METHODOLOGY

3.1 Project Development Methodology

In the context a research project where the final deliverable is an information system, a framework to structure, plan and control the system development process is required. The development methodology chosen for this project is **Rapid Application Development (RAD)**. At the core of the RAD methodology is iterative development and prototyping techniques with the aim of accelerating system development. The primary reason of choosing RAD is to enhance flexibility in the face of time constraints. RAD incorporates iterative development practices which interleave the design and construction of software, compensating for lack of extensive pre-planning time. This makes it easier for the system developer to continually refine the system model and produce improved iterations at various time checkpoints over the course of the project. In addition to that, RAD utilizes a data-driven information engineering approach which can adequately model aspects of information flow which occurs in the context of emergency communications, thus making it especially relevant to this project. A potential weakness of the methodology is that certain compromises in functionality and performance may be entailed to complete the project in the stipulated time. Besides, the focus on producing rapid software iterations may lead to very little written documentation during the development process, thus a significant amount of post-project documentation would be required. Development of the project system prototype will be divided into 4 main phases as per **Figure 3.1.**

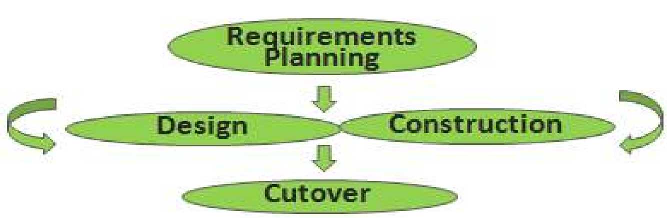

Figure 3.1: Phases of Rapid Application Development

`

3.2 Project Activities

The project activities comprise of the software development methodology processes which make up the project.

3.2.1 Requirements Planning

The requirements planning phase combines elements from both the planning and analysis phases in a traditional System Development Life Cycle. The tasks of requirements planning are to elicit the core problem statement, followed by identification of the scope and objectives of the project.

Literature Review:

Once the aforementioned elements have been documented, a thorough analysis is required to produce a set of desired functional requirements which can effectively solve the problem and achieve the system objectives. Reviewing existing literature is a crucial technique in the requirements planning process, as existing systems, processes, theoretical frameworks can be evaluated, resulting in a synthesis of knowledge required to produce an appropriate solution for the problem statement.

3.2.2 Design

The design stage is where the system architecture is produced for the solution to the problem. This involves designing the information and operation flows within the system and ascertaining the desired system inputs, processes and outputs. The design phase is greatly dependant on the documentation from the requirements planning phase to validate if an effective model of the system is being built. It will continually inform the refinement of the system architecture, which will serve as a basis for system construction.

`

System Architecture:

Figure 3.2 represents the overall system architecture, an output of the design phase in the rapid application development methodology. It provides a high-level overview of the various components and data flow within the system.

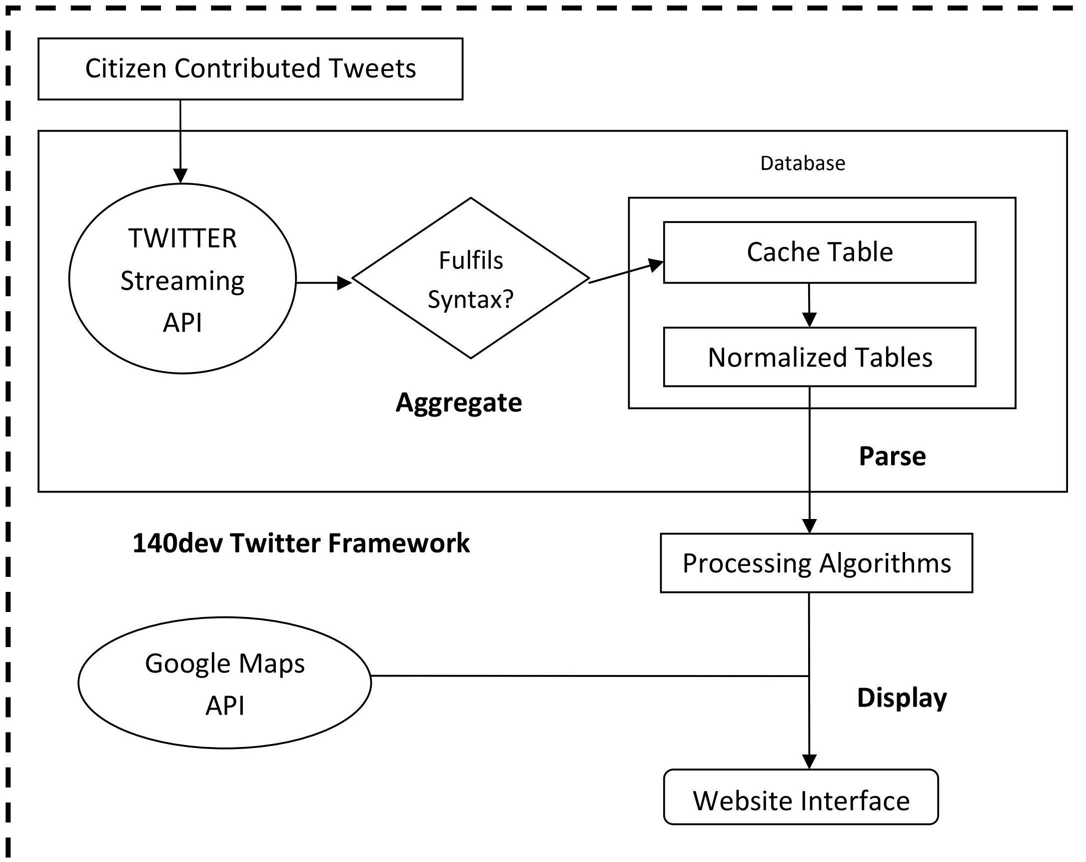

Figure 3.2: System Architecture

The web system consists of 3 major processes: **Aggregate, Parse and Display**. The 140dev Twitter Framework is the backbone of the system which aggregates and parses raw data from the Twitter platform, and the tweet data is further processed with custom algorithms for the system's customized needs before being stored in a MySQL database. The tweet aggregation architecture is designed in a way that the website does not interact directly with the Twitter API, with tweet data being delivered to the website via an intermediary database, the reasons of which will be elaborated later. The

`

website subsequently embeds Google Maps API functionality combined with the tweet data extracted from the database to deliver simple visualizations of emergency-related information.

3.2.3 Construction

The construction stage is executed iteratively with the design phase, and focuses on the programming/ application development and testing aspects of the project. The separate functional modules of the system are developed in accordance to the system architecture, namely the aggregation, parsing and displaying functions. These modules will then be unit tested independently by ascertaining the validity of output data and ensuring correct behavior. Once all modules are determined to be functioning properly, the various components will be integrated to test system functionality as a whole (functionality testing).

System Usability Scale (SUS):

User involvement is co-opted as the developed web interface is subjected to usability testing to ensure that the end product provides a satisfactory user experience. The system usability scale (SUS) is the method of choice, as it is technology independent and a general yet well known measure of the perception of system usability. The results of the SUS questionnaire carried out in this project are tabulated in **Chapter 4**.

3.2.4 Cutover

The cutover phase is where the system is ready to be demonstrated with its intended functionality, followed by finalization of system build and deployment of the system,

Note: *Both 2nd and 3rd phases are interleaved, and are continuous interactive processes executed in parallel during system development.*

`

3.3 Tools Required

The tools required for the implementation of this project are as below:

1) XAMPP
2) Twitter API
3) 140dev Twitter Framework
4) Google Maps API
5) Adobe Dreamweaver CS5

3.3.1 XAMPP

XAMPP (www.apachefriends.org/en/xampp.html) is a web server solution stack package which consists of the Apache HTTP Server, MySQL database and an interpreter for PHP scripts. It is intended as a development tool to test web systems in a local computer, and provides integration with the MySQL database management system through the creation and modification of database elements from an administrative interface. The benefits of the XAMPP solution stack is that it is free, open-source and cross platform. It is also extremely easy to set-up as integrated installation of the various components under the package is supported.

3.3.2 Twitter API

Twitter maintains an API to allow developers to build applications which can be integrated with Twitter. The API is a constantly evolving specification which acts as an interface for external applications to access the data contained within the Twitter network. Twitter data is commonly returned in either XML or JSON, as well as in RSS and ATOM feed formats. Depending on the portions of the API being used, the API may require no authentication, basic authentication (username and password) or OAuth authentication. The multiple facets of the Twitter API are as per **Table 3.1**.

`

Table 3.1: Components of Twitter API

API	**Description**
Twitter for Websites	Provides integration of basic Twitter functions in a website such as the 'Tweet' and 'Follow' buttons.
Search API	Designed to allow queries for Twitter content, the Search API facilitates finding keyword-specific Tweets, Tweets originating from a particular user, or Tweets referencing a particular user from the previous 5-7 days. No authentication required.
REST API	Allows programmatic access to core primitives within Twitter such as timelines, status updates and user information which creates Twitter features on a website or mobile app. Uses OAuth Authentication.
Streaming API	Specifies and tracks large quantities of keywords in Tweets, retrieves geo-tagged Tweets from a certain region, and returns the public status of a user, suitable for developers who require near real-time data and possess aggressive querying needs. Uses basic and OAuth Authentication.

Streaming API:

<u>Benefits</u>

The **Streaming API** (https://dev.twitter.com/docs/streaming-apis), or more specifically the public streaming API endpoint, is the method of choice for this project to integrate with Twitter. The main reason for choosing the Streaming API in the context of this project is timeliness. The low latency of the Streaming API means that there is minimal lag between creation of a tweet and delivery to the API. The near real-time nature of obtained data is very important given the time-sensitive property of emergency-based information.

Challenges

There are some challenges in implementing the Streaming API as well. Connecting to the Streaming API requires keeping a persistent HTTP connection open, and on top of that an account may create only one standing connection to the public streaming endpoint. In addition to that, rate limiting dictate that the number of messages sent to our application will have to be capped to a fraction of the entire volume of Tweets on the Twitter network. Thus, if the total number of Tweets which satisfy the search criteria exceeds the streaming cap, a certain subset of Tweets will not be delivered. The potential lack of completeness of data means that the proposed system will need to overcome the aforementioned challenges before it can be implemented in a production setting. However, the rate limits currently afforded by the public Streaming API are more than adequate in the context of our project.

3.3.3 140dev Twitter Framework

The 140dev Twitter framework (http://140dev.com/free-twitter-api-source-code-library/) is a free source code library written in PHP and JavaScript by Adam Green which is released under the General Public License. The core module of the framework is the Twitter Database Server, which connects to the Streaming API to collect Tweets based on specific keywords and caches them in a MySQL database. It utilizes the Phirehose PHP library to easily connect and consume a stream of Twitter data while hiding the complexity of connection handling, persistent HTTP connection and filter predicate updates. The framework aggregates tweet data in a database, as this approach enhances flexibility and potential for expansion by allowing additional functional modules which are written to access Twitter data solely through the database without coming into direct contact with the Twitter API. In short, the 140dev Twitter Framework was chosen for the project as it provides the web system with a simplified interface to the Twitter API.

`

3.3.4 Google Maps API

The Google Maps API is a free service available for use on any website as long as it is freely accessible to the end-user. It allows developers to not only embed maps from the Google Maps website in their web pages but customize, annotate or add content within the map. In our project, we have specifically chosen to utilize the **Google Maps JavaScript API Version 3** (https://developers.google.com/maps/documentation/javascript/reference) which allows the manipulation of map elements based on our emergency tweet data to enhance interactivity, beyond deploying static map images. The reason for using the Google Maps API is that it allows the project to tap into an existing, comprehensive map engine to plot the tweet data and provide a geospatial view of emergencies.

3.3.5 Adobe Dreamweaver CS5

The source code editor of choice is the Adobe Dreamweaver CS5, (http://www.adobe.com/products/dreamweaver.html), a web development application which supports web technologies such as HTML, CSS, JavaScript and server-side scripting languages such as PHP. This software was chosen for the project due to its ability to support syntax highlighting, built in CSS style features as well as dual code and design windows for easy development.

`

3.4 Key Milestones

Key Milestones for the project in FYP2 are as per **Table 3.2** below:

Table 3.2: FYP2 Milestones

Week	FYP2 Milestone	Date
4	Submission of Progress Report	10 Oct 2012
11	Dissertation Submission	26 Nov 2012
11	Pre-SEDEX	28 Nov 2012
12	FYP2 VIVA	5 Dec 2012
14	Technical Report Submission	19 Dec 2012

3.5 Gantt Chart

Table 3.3: Combined Gantt Chart for FYP 1 and FYP 2

Milestones

No.	Details / Week No.	FYP 1														FYP 2													
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	Project Title Selection																												
2	Submission of Project Title Proposal																												
3	Literature Review																												
4	Submission of Extended Proposal																												
5	Requirements Planning Phase																												
	a. Identify Functional Requirements																												
	b. Analysis of Required Tools and Costs																												
	c. Production of Basic System Model																												
6	Proposal Defense																												
7	Design Phase																												
	a. Analysis of System Data Flow																												
	b. Visual Representations of System																												
	c. User Interface Design																												
8	Submission of Interim Report																												
9	Construction Phase																												
	a. Programming Functional System Units																												
	a. Unit Testing																												
	c. System Integration Testing																												
10	Design Phase (2nd Iteration)																												
11	Submission of Progress Report																												
12	Construction Phase (2nd Iteration)																												
13	Pre-SEDEX presentation																												
14	Cutover Phase																												
15	Dissertation Submission																												
16	FYP2 VIVA																												
17	Technical Report Submission																												

`

CHAPTER 4
RESULTS AND DISCUSSION

4.1 Use Case Diagram

Figure 4.1 represents the use case diagram of the Twitter-integrated web system for emergency alerts.

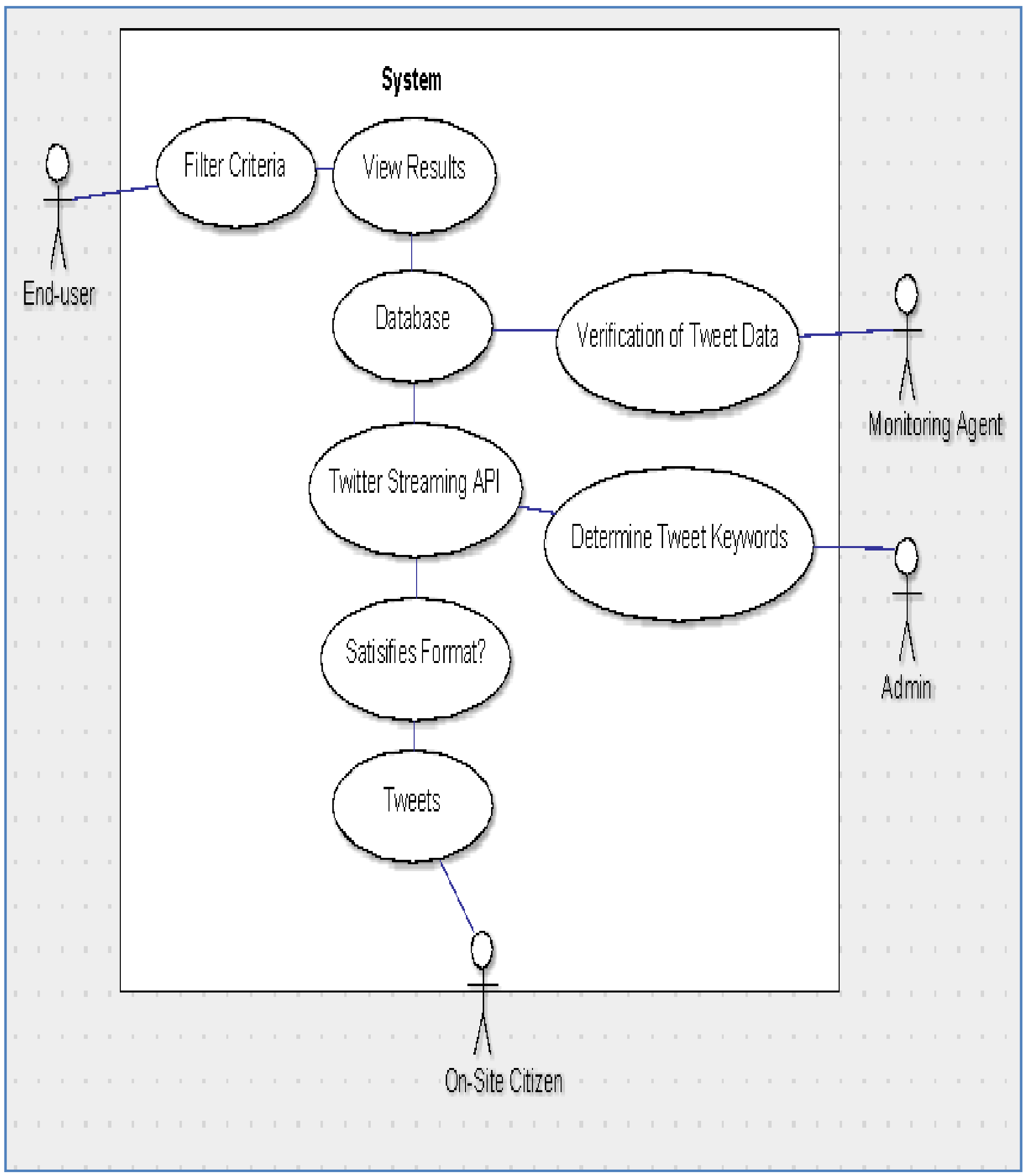

Figure 4.1: Use Case Diagram

`

The main actors in the system are the:

- **Admin:** The administrator is responsible for determining the keywords to control which Tweets are collected from the Streaming API. The keywords will be based on a prescriptive syntax format for an emergency happening.

- **End-user:** The end-user represents any user who enters the publicly accessible web interface. They can control the results displayed on the map interface by using the filtering criteria via the select boxes. The web interface will be further discussed in **Section 4.4**.

- **On-site citizen:** The on-site citizen may be either a part of or casual observer of the emergency who provides volunteered Tweets containing emergency-related data. The on-site citizen has to construct the tweet on such a way that it fulfills the prescribed syntax format.

- **Monitoring agent:** The monitoring agent represents a member of the emergency management organization implementing the system who monitors incoming Tweets for events. The agent plays a role in verifying the tweet data received. If the information is corroborated by reliable sources, the corresponding tweet will be marked as verified in the results on the web interface.

`

4.2 Aggregation Process

The current design relies on the Twitter platform as the main source of data input for the system. The username and password of a valid Twitter account is configured within the Phirehose Library, which uses basic authentication to establish and maintain a connection to the Twitter Streaming API. A set of keywords are provided to a ***get_tweets.php*** script to filter the incoming Tweets, with the keywords based on the prescribed syntax for emergency events. The syntax is, in turn, determined by the emergency management organization implementing the system based on the nature of the emergency and type of data required. The proposed tweet format is as follows:

#[keyword] #[alert type] #dct [description] #loc [location]

Notice that with the exception of the compulsory placement of certain hash tags, the tweet format above can be tweaked to cater to different emergencies which have different alert types, with differing description and location. Below are two italicized examples of tweets which cater to two different (imaginary) use cases, but nevertheless fulfill the format above.

#thaifloods #flood #dct 0.5m deep #loc Bangkok Thailand

#hurricanephilly #shelter #dct 20 spaces left for refuge #loc Philadelphia US

There are four main elements present in the tweet syntax, namely the keyword, alert type, description and location. As mentioned earlier, the keyword hashtag serves as a unique identifier of which emergency event the tweet is referring to. The alert type hashtag determines the category of alert within the event, while the #dct tag precedes a text description of the alert. Twitter allows a user's Tweets to be automatically geo-tagged, provided that the user enables the option. The purpose of the location tag #loc is to allow geocoding to be performed on the location text should geo-tagging functionality be disabled by the tweet user. If geo-tagging is enabled, however, the geo-tagged location takes precedence over the location specified in the tweet text when processed by the system. The length of the description and location text is solely at the discretion of the user, as long as it is within the 140 character limit enforced on Twitter messages. The location text

`

should ideally be as specific as possible in the event that geo-tagging is not enabled, to avoid wrong geocoding results for locations which may share identical names.

The script for the tweet collection process is initiated as a background process, where raw tweet data which satisfies the filter parameters is returned in the JSON format and the full JSON payloads are added to the cache table in the MySQL database. **Figure 4.2** is a data flow diagram of the aggregation process.

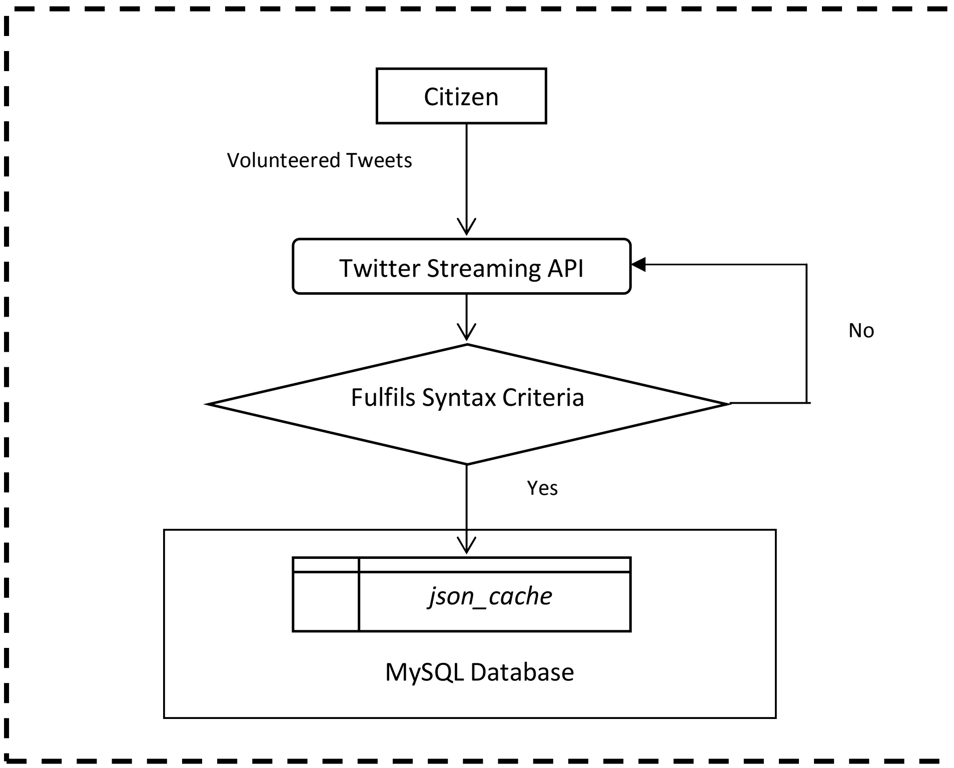

Figure 4.2: Data Flow Diagram for Aggregation Process

`

4.3 Parsing Process

The next process is the parsing process, which parses each raw tweet in the cache table into readable tweet data, to be distributed into multiple MySQL tables. A parsing script, ***parse_tweets.php*** in the 140dev Twitter Framework is initiated as a background process to parse each tweet stored in the cache, to extract tweet text as well as metadata such as latitude and longitude from the raw tweet payload. Once a raw tweet in the cache table has been processed it will be marked as parsed, so subsequent parsing processes will bypass that payload. These data are stored in a *tweets* table as a way of organizing the information, as per **Figure 4.3**. **Figure 4.4** is a data flow diagram of the parsing process.

Tweets
tweet_id
tweet_text
entities
created_at
geo_lat
geo_lng
tweet_location
user_id
screen_name
name
event
alert
verified

Figure 4.3: Tweets Table Fields

`

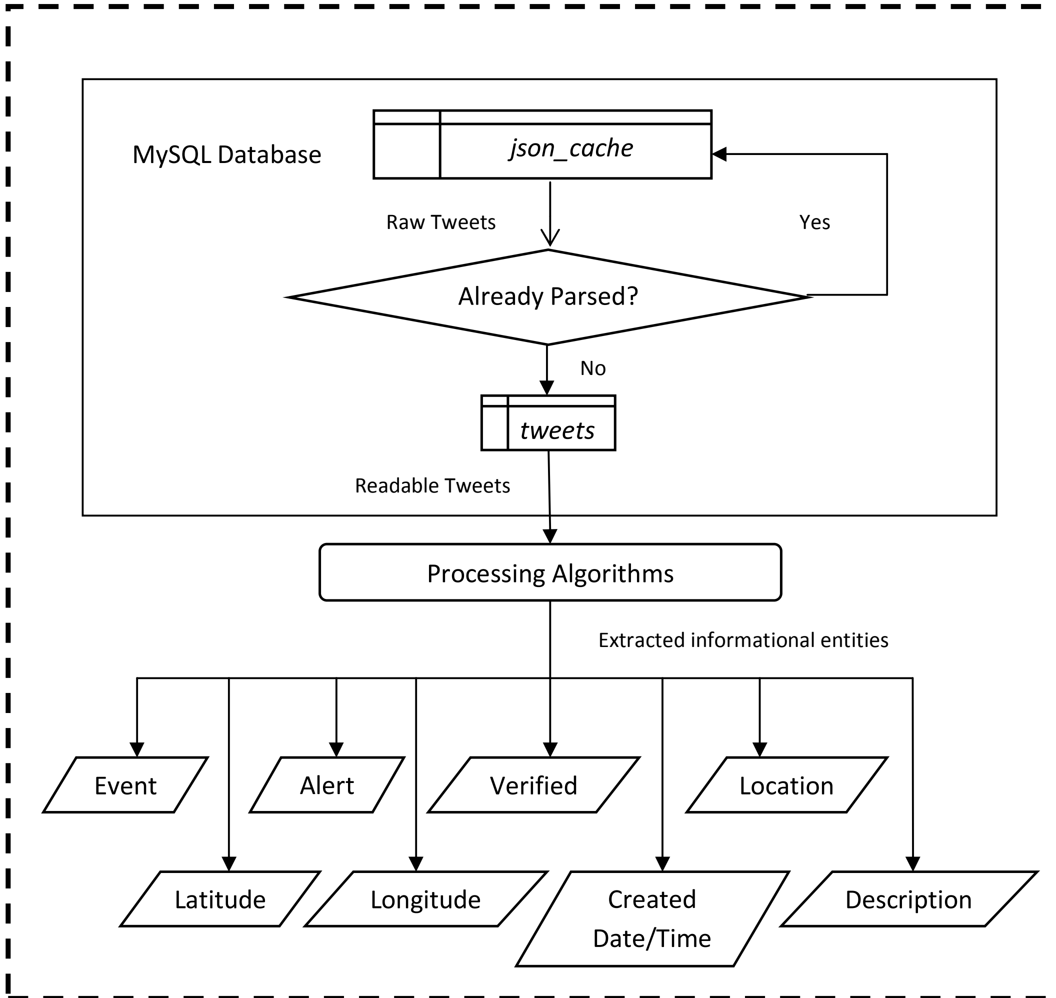

Figure 4.4: Data Flow Diagram for Parsing Process

The *tweets* table is essential to the web system as an input for the scope of content displayed which will be detailed later. The *tweets* table is connected to ***processquery.php*** script which contains algorithms designed to process the tweet format. The processing algorithm performs programmatic operations on the tweet text to extract the relevant informational entities within the tweet. The algorithm behaves strictly in the sense that it does not tolerate faulty spacing, spelling mistakes or incorrectly-ordered tags which cause the message to differ from the prescribed syntax. Any formatting error within the tweet will result in ***processquery.php*** disregarding it. The following is a code snippet for ***processquery.php***.

```
while ($row = @mysql_fetch_assoc($result)){

$value = $row['tweet_text'];
$tokens = explode(" ", $value);
$total=count($tokens);

$keydct = array_search('#dct', $tokens);
$keyloc = array_search('#loc', $tokens);

if (($keydct+1!=$keyloc)&&($keyloc+1!=$total)&&(!in_array('', $tokens))){

$node=$dom->createElement("marker");
$newnode=$parentnode->appendChild($node);
$newnode->setAttribute("created",$row['created_at']);
$newnode->setAttribute("lat",$row['geo_lat']);
$newnode->setAttribute("lng",$row['geo_long']);
$newnode->setAttribute("verified",$row['verified']);
$newnode->setAttribute("keyword",$row['event']);
$newnode->setAttribute("alert",ucfirst($row['alert']));
$location="";
$description="";

for($i=$keydct+1; $i<$keyloc; $i++){
$newString = ucfirst($tokens[$i]);
$description=$description.$newString." ";
$newnode->setAttribute("description",$description);
}

for($i=$keyloc+1; $i<$total; $i++){
$newString = ucfirst($tokens[$i]);
$location=$location.$newString." ";
$newnode->setAttribute("location",$location);
}
}

}
```

On a separate note, notice that the aggregation process as in **Section 4.2** and parsing processes are decoupled. It ensures that the aggregation code does not perform heavy processing tasks on tweet data which arrives at a fast rate, thus reducing the chances of losing incoming Tweets. The separation of these two operations protects the user interface layer when the Twitter API fails. By caching all the API responses in the database, the parsing processes can still function normally to deliver processed results to the user interface. In short, the web application built on top of the data collection framework can safely rely on the database of Tweets without being subject to any Twitter API performance issues.

`

4.4 Display Process

This is the stage where information extracted from the aggregation and parsing processes are delivered to a visual interface which can be accessed by an end-user. A web page which mashes up the Google Map and tweet data is created using a combination of Google API, JavaScript, AJAX, PHP and HTML techniques. **Figure 4.5** displays the full page view of the web interface.

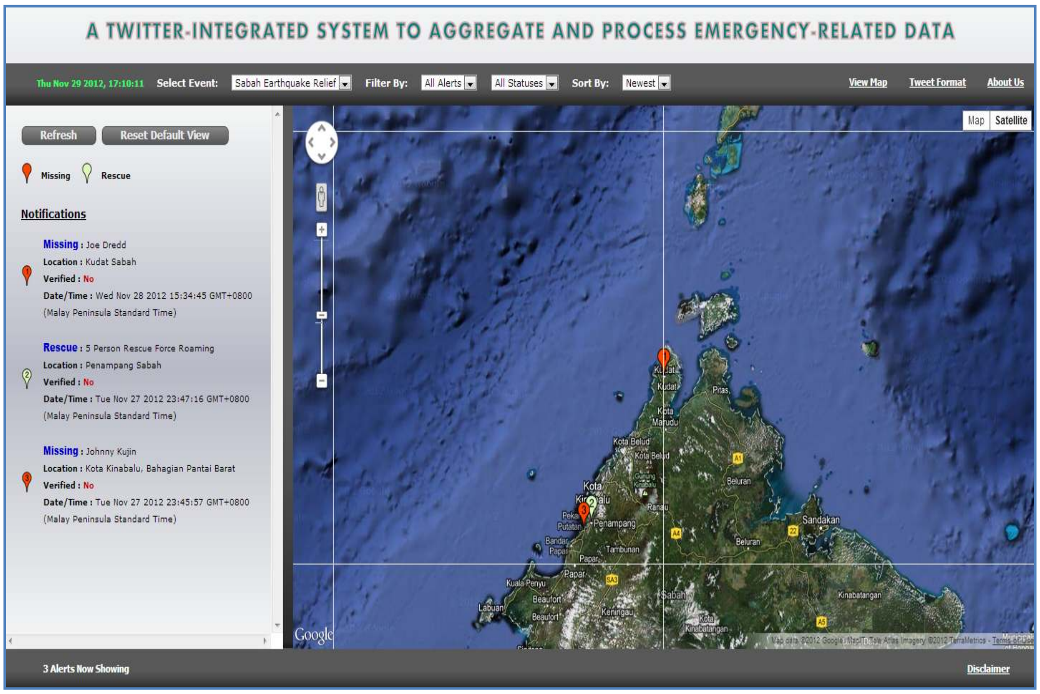

Figure 4.5: Full Page View of Website Interface

Select boxes which are coded into the web interface allow the user to filter or sort the type emergency information displayed based on the desired attributes. The attributes which can be filtered and sorted are as per data flow diagram in **Figure 4.6**. A query is constructed based on the filter parameters provided, and the results are subsequently exported into an XML format via ***processquery.php***. The XML output from the database are then retrieved through asynchronous JavaScript calls on the web page. The fetched data consists of informational entities which are specified in the prescribed syntax format, including the type of alert, description, and location, accompanied by additional data, namely the created date/time of the tweet and verification status to indicate if the tweet information has been corroborated (refer to

`

Figure 4.7). After loading the Google Maps API function, the extracted tweets are subsequently placed as marker points on the map with their respective information windows, as per **Figure 4.8.** The marker points are color coded to distinguish between different types of alerts on the map. The location of the marker point is dependent on whether the tweet is geo-tagged. If geo-tagged functionality is switched on, the latitude and longitude metadata within the tweet will take precedence, otherwise the geo-coding result based on the location text provided in the tweet will be utilized. The interface also has a sidebar to list color codes and emergency alerts which correlates with the map markers, with refresh and reset buttons available to refresh the displayed content and to reset the map and select boxes to its default settings, as per **Figure 4.9**. Clicking on a notification on the sidebar will expand the associated information window on the Google Map.

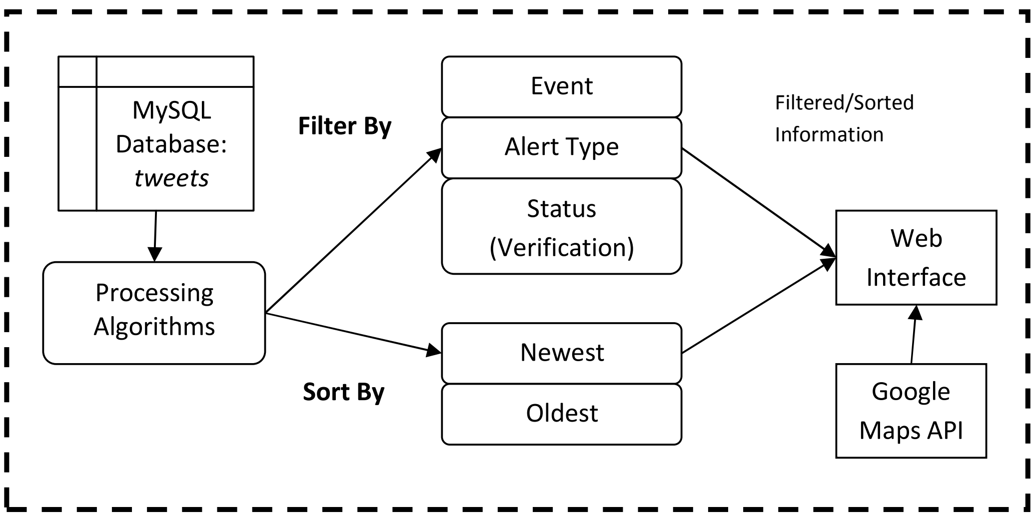

Figure 4.6: Data Flow Diagram for Display Process

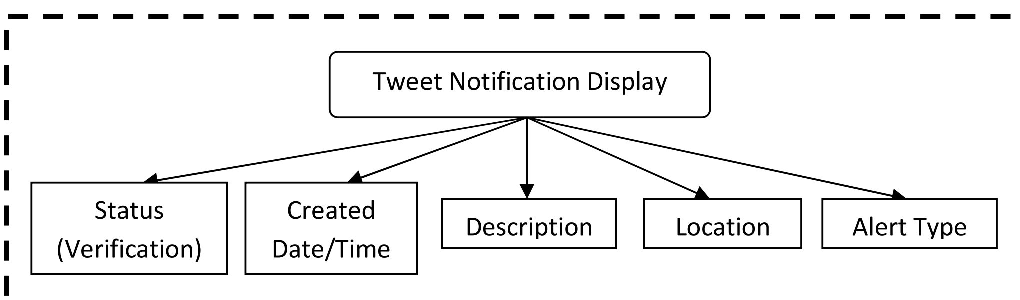

Figure 4.7: Elements of Tweet Notification Display

Figure 4.8: Tweet Information Window for a Map Marker Point

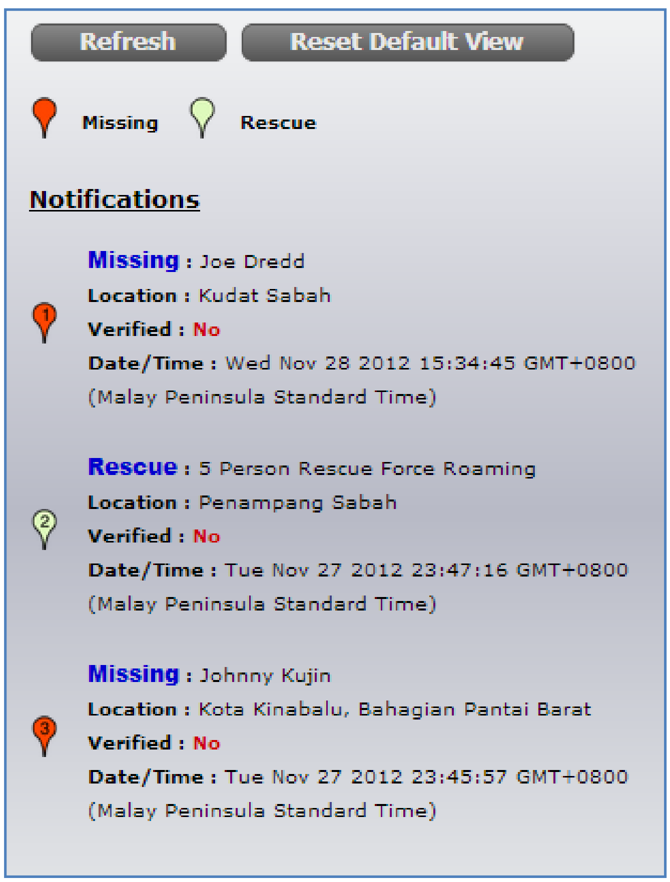

Figure 4.9: Sidebar List of Emergency Tweet Information

`

4.5 Sample of Functional Testing Results

To ensure the system has the desired end-to-end functionality as outlined within the objective and scope of study, functional testing was performed on the completed system. Functional testing represents a form of black box testing, which uses test cases based on the system specifications under test. In order to perform functional testing, the following five steps were undertaken:

4.5.1 Identifying the Function that the Software is Expected to Perform

The function to be tested is the correct display of a submitted tweet which fulfils the syntax criteria on the web interface. In short, it covers the end-to-end process of feeding a tweet to the Twitter platform and ensuring the tweet is displayed on the system web interface with the right information. This represents a collective combination of the aggregation, parsing and displaying functions as outlined in **Section 4.2, 4.3** and **4.4** respectively.

4.5.2 Creating Input Data Based on the Function's Specifications

The desired input data would depend on the keywords specified within the aggregation script. For the purpose of the test case, we specify #phgfloods, #loc and #dct as the set of keywords to be collected within the aggregation script.

```
$stream->setTrack(array('#phgfloods #flood #dct #loc'));
```

Thus, the input data should contain all the keywords as mentioned above, in line with the proposed tweet format in Section 4.2. An input tweet is drafted for our test case as below:

#phgfloods #floods #dct depth approaching 1.0m #loc temerloh pahang

`

4.5.3 Determining the Output Based on the Function's Specifications

On receipt of the above tweet by the system, the tweet appears on the notifications sidebar of the web interface on a default display for the #phgfloods event filter, with the appropriate values for the respective fields. **Table 4.1** details the expected output values for the test case.

Table 4.1: Expected Output Values for Test Case

Field	Expected Output Value
Alert	Flood
Description	Depth approaching 1.0m
Location	Temerloh Pahang
Verification Status	No

4.5.4 Execution of Test Case

The test case was executed by first initiating ***get_tweets.php*** (**Figure 4.10**) and ***parse_tweets.php*** (**Figure 4.11**), the aggregation and parsing scripts respectively, through Windows command prompt. The input data (tweet) was then fed into the Twitter platform. A Twitter account was created especially for the project to post the tweet (**Figure 4.12).**

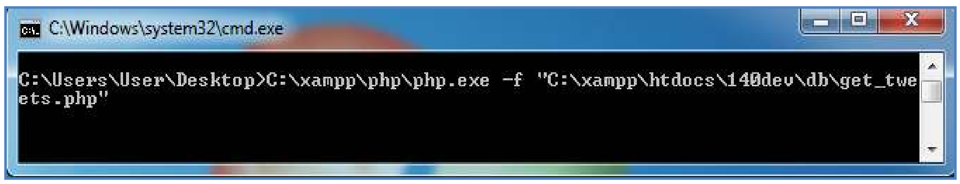

Figure 4.10: Running the Aggregation Script on Command Prompt

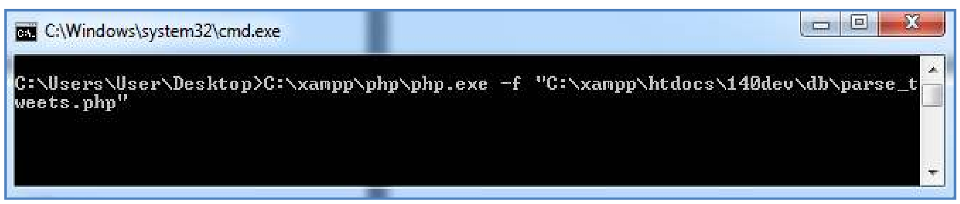

Figure 4.11: Running the Parsing Script on Command Prompt

Figure 4.12: Part of Twitter Screenshot Showing Posted Tweet

`

4.5.5 Comparison of Actual and Expected Output

In order to ascertain if the system functionality is working as desired, the actual output values was compared with the expected output values. The screenshot of the output tweet displayed on the notifications sidebar is as per **Figure 4.13**.

Figure 4.13: Screenshot of Output Tweet Data

From **Figure 4.13**, we can see that each item of the tweet has been processed correctly by the system.

`

4.6 System Usability Scale Survey Results

A system usability scale (SUS) survey was conducted to measure the perception of usability for the web interface of the developed system as described in **Section 4.4**. The survey was conducted online using Google Documents form to facilitate distribution of the questionnaire, where a total of 20 respondents were surveyed. The questionnaire consisted of a ten-item attitude Likert scale to gauge the user's perception of system usability.

Once all responses were obtained for each item within the questionnaire, scores for each question were scaled from 0 to 4. For odd numbered questions 1, 3, 5, 7 and 9, the score contributions is the scale position minus 1. As for the even-numbered questions 2, 4, 6, 8 and 10, the score contribution is 5 minus the scale position. The tabulated results of the survey are as per **Table 4.2** in the next page. The overall SUS score was then calculated using the SUS conversion formula (http://www.measuringusability.com/sus.php). Average scores for each question were added up and the total multiplied by 2.5. This converts the range of possible values from 0 to 100 instead of from 0 to 40. As the sum of the score was 33.7, the resulting overall SUS score is:

$$\mathbf{33.7 \times 2.5 = 84.25}$$

A score of **84.25** was obtained, which marks positive usability as per SUS conventions. It would be prudent to emphasize, however, that any SUS score measures perception of usability and is not intended to diagnose usability problems.

`

Table 4.2: System Usability Scale Survey Results

*R = Respondent

No.	Question	Scaled User Scores (0-4)																				Average Score
		R1	***R2***	***R3***	***R4***	***R5***	***R6***	***R7***	***R8***	***R9***	***R10***	***R11***	***R12***	***R13***	***R14***	***R15***	***R16***	***R17***	***R18***	***R19***	***R20***	
1	I think that I would like to use this system frequently.	3	3	3	3	4	3	2	3	3	3	3	3	4	3	4	3	2	3	4	3	3.1
2	I found the system unnecessarily complex.	3	3	3	3	4	4	3	4	4	3	3	4	4	4	4	4	4	4	4	4	3.65
3	I thought the system was easy to use.	3	3	4	4	3	3	4	3	4	3	3	3	4	3	4	3	4	3	4	3	3.4
4	I think that I would need the support of a technical person to be able to use this system.	4	4	3	4	4	4	4	4	3	4	4	3	3	4	3	4	4	3	4	4	3.7
5	I found the various functions in this system were well integrated.	3	4	2	3	4	3	3	3	4	3	3	3	2	4	3	3	4	3	3	3	3.15
6	I thought there was too much inconsistency in this system.	3	3	3	4	3	4	3	4	4	3	4	4	4	3	4	4	3	3	3	4	3.5
7	I would imagine that most people would learn to use this system very quickly.	3	3	3	4	3	2	3	3	2	3	3	3	4	2	4	3	3	4	4	4	3.15
8	I found the system very cumbersome to use.	3	4	3	3	3	3	3	3	3	4	4	4	4	3	4	3	3	4	4	4	3.45
9	I felt very confident using the system.	3	3	2	3	3	4	3	4	3	3	3	2	3	4	3	4	4	3	3	2	3.1
10	I needed to learn a lot of things before I could get going with this system.	3	4	4	4	3	4	3	4	3	3	4	3	4	3	4	3	4	3	3	4	3.5
																			Sum Score			**33.7**
																			Overall Score			**84.25**

`

4.7 Limitations

Over the course of project implementation, numerous limitations have been encountered. The major limitation with respect to the system is that it relies solely on citizen-contributed tweets as its input. Whilst it is a timely and efficient method to crowd-source data on emergency situations, it might also lead to some challenges in terms of data quality. A problem arises if the citizen-provided data is untrustworthy. To counter this limitation, a verification status field has been appended to each tweet data received to indicate if the information has been corroborated, where manual intervention is required to verify the provided information. Sourcing data from a large pool of citizens may also mean that the system may have duplicates of the same piece of information provided to it, which leads to data redundancy.

The system is also limited in the sense that it still has to rely on Twitter users to customize their tweets in such a way that the tweets comply with the extremely rigid specifications of the tweet format which was set beforehand. It assumes the discipline of users in adhering to syntax rules, such as correct spelling, spacing and placement of hash tags. It would be ideal if further research could be put into strengthening the processing algorithms to enable citizens to provide more loosely typed tweets which can still be recognized by the system.

Nevertheless, despite the limitations above, it should be noted that the system only provides a visualization of emergency situations, and is not meant to be an authoritative source of reference for decision-making in the field.

`

CHAPTER 5
CONCLUSION & RECOMMENDATIONS

5.1 Conclusion

In the age of the interconnected world, it is more important than ever to leverage on the affordances provided by social media technologies to innovate and improve on emergency management.

Utilizing Twitter to crowd-source data on emergencies allows for immediacy in generating information which cannot be matched by traditional media outlets. Despite contentions that Twitter accounts of events lack rigorous fact checking, Twitter's ubiquity, ease of use and the rapidity of micro-blogging communications means that it is still an ideal tool to collect volunteered data during time-sensitive emergencies where first-hand accounts are extremely valuable. The Twitter-integrated web system developed in this project is envisioned as a solution which is able to automate the aggregation of emergency-related tweets available on social media platforms and organize it to produce an emergency-specific informational account. Said system adopts a prescriptive syntax approach to facilitate the aggregation of emergency-related tweet, processes the data based on geo-location information and syntax into organized informational entities relevant to an emergency, and subsequently deliver alerts on a map-like interface using the Google Maps API. The aforementioned approach is best for use in emergencies where the transmission timely, quantitative data is of paramount importance.

It is hoped that the project can provide a framework on extracting and displaying useful emergency alerts with a geographical perspective based on volunteered citizen Tweets, and ultimately contribute to the domain of social media-assisted emergency applications.

`

5.2 Recommendations

The Twitter-integrated web system to aggregate and process emergency-related data is developed as part of a proof of concept framework based on literature research over the constrained duration of this project. Therefore, it is inevitable that the system prototype can be further improved in terms of performance and features to be enhanced into a system ready for production purposes.

Among the notable recommendations for the continued development of this project is to enhance the map visualizations displayed. In addition to simple marker point overlays, the web interface may include more annotations on maps such as polylines, polygons and custom overlays to convey more complex representations of the emergency from a geographical context. An additional recommendation would be to enhance the quality of input received by the system. The aggregation and parsing algorithms would need to be strengthened to ensure that redundancy of citizen-provided data does not occur. In addition to Twitter integration, the system may also be integrated with formal emergency-based organizations such as the national weather service to include alert data from authoritative sources, to ensure that the data is more reliable. The framework may even be extended for use in other domains. Currently, the system prototype only supports the processing of a custom syntax which contains entities which are relevant specifically for emergencies. However, with suitable tweaks to the prescribed syntax and processing algorithms, the framework could ostensibly be applied in other time-sensitive, information-critical areas such as crime reporting, citizen activism, and election coverage where first-accounts of events are of high utility.

The prototype has been developed as a web system to be accessed on desktop browsers. To provide flexibility to end-users who wish to view the interface on a mobile device, the web interface may be optimized for mobile browsers or the entire system may be rebuilt as a dedicated application on a mobile platform.

`

REFERENCES

Ali, S. (2010). Twitter Helps in Haiti Quake Coverage, Aid. Retrieved from: http://blogs.wsj.com/digits/2010/01/14/twitter-helps-in-haiti-quake-coverage-aid/

Boyd, S. (2010). Disaster Microsyntax: Project EPIC, Tweak the Tweet, and Emergency Codes. Retrieved from: http://stoweboyd.com/post/786084841/disastermicrosyntax

Guy, M., Earle, P., Ostrum, C., Gruchalla, K. & Horvath, S. (2010). Integration and Dissemination of Citizen Reported and Seismically Derived Earthquake Information via Social Network Technologies. In P.R. Cohen, N.M. Adams, and M.R. Berthold (Eds.)*: IDA 2010, LNCS 6065, 2010,* (pp. 42–53).

Kaplan, A.M., Haenlein, M. (2010). "Users of the world, unite! The challenges and opportunities of Social Media". *Business Horizons* 53(1): 59–68.

MacEachren, A.M., A.C. Robinson, A. Jaiswal, S. Pezanoski, A.Savelyev, J. Blanford and P. Mitra (2011). Geo-Twitter Analytics: Applications in Crisis Management, *Proceedings of the 25th International Cartographic Conference, Paris, France*

Mendoza, M., Poblete, B. and Castillo, C. (2010). Twitter Under Crisis: Can we trust what we RT? , *1st Workshop on Social Media Analytics (SOMA '10), Washington, DC, USA.*

Mills, A., Chen, R., Lee, J.K., & Rao, H.R. (2009). Web 2.0 Emergency applications: How useful can Twitter be for emergency response?. *Journal of Information Privacy & Security* 5(3): 3-26.

`

Palen, L. & Liu, S.B. (2007). Citizen Communications in Crisis: Anticipating a Future of ICT-Supported Public Participation. *Proceedings of the SICGHI Conference on Human Factors in Computing Systems, April 28 - May 3, 2007,* (pp.727-736).

Poulsen, K. (2007). Firsthand Reports From California Wildfires Pour Through Twitter. Retrieved from: http://www.wired.com/threatlevel/2007/10/firsthand-repor/

Starbird, K. & Palen, L. (2010). Pass It On? Retweeting in Mass Emergency. *Proceedings of the 7th International ISCRAM conference – Seattle, USA, May 2010.*

Starbird, K. & Stamberger, J. (2010). Tweak the Tweet: Leveraging Microblogging Proliferation with a Prescriptive Syntax to Support Citizen Reporting. *Proceedings of the 7th International ISCRAM Conference – Seattle, USA, May 2010.*

Sutton, J., Palen, L. & Shklovski, I. (2008). Backchannels on the Front Lines: Emergent Uses of Social Media in the 2007 Southern California Wildfires. *Proceedings of the 5th International ISCRAM Conference – Washington, DC, USA, May 2008*, (pp. 624-631).

Van Leuven, L. (2009). *Optimizing Citizen Engagement During Emergencies Through the Use of Web 2.0 Technologies*. Master's Thesis, Naval Postgraduate School, Monterey, CA.

Vieweg, S., Hughes, A.L., Starbird, K. & Palen, L. (2010). Microblogging During Two Natural-Hazard Events: What Twitter May Contribute to Situational Awareness. *Proceedings of the 28th International Conference on Human Factors in Computing Systems, 2010,* (pp. 1079-1088).

APPENDICES

Appendix 1: SUS Questionnaire

1. I think that I would like to use this system frequently.
2. I found the system unnecessarily complex.
3. I thought the system was easy to use.
4. I think that I would need the support of a technical person to be able to use this system.
5. I found the various functions in this system were well integrated.
6. I thought there was too much inconsistency in this system.
7. I would imagine that most people would learn to use this system very quickly.
8. I found the system very cumbersome to use.
9. I felt very confident using the system.
10. I needed to learn a lot of things before I could get going with this system.

Response Format:

Strongly Disagree 1	2	3	4	Strongly Agree 5
○	○	○	○	○

`

A Twitter-Integrated Web System to Aggregate and Process Emergency-Related Data

Alan Oxley
Computer and Information Sciences Department
UniversitiTeknologi PETRONAS
Bandar Seri Iskandar, 31750, Tronoh, Perak, Malaysia
alanoxley@petronas.com.my

Yong Chang Yi
Computer and Information Sciences Department
UniversitiTeknologi PETRONAS
Bandar Seri Iskandar, 31750, Tronoh, Perak, Malaysia
yongchangyi@gmail.com

ABSTRACT

A major challenge when encountering time-sensitive, information critical emergencies is to source raw volunteered data from on-site public sources and extract information which can enhance awareness on the emergency itself from a geographical context. This research explores the use of Twitter in the emergency domain by developing a Twitter-integrated web system capable of aggregating and processing emergency-related tweet data. The objectives of the project are to collect volunteered tweet data on emergencies by public citizen sources via the Twitter API, process the data based on geo-location information and syntax into organized informational entities relevant to an emergency, and subsequently deliver the information on a map-like interface. The web system framework is targeted for use by organizations which seek to transform volunteered emergency-related data available on the Twitter platform into timely emergency alerts which can enhance situational awareness, and is intended to be accessible to the public through a user-friendly web interface. The developed system has a system usability scale score of 84.25, after results were tabulated from a usability survey on 20 respondents. Said system is best for use in emergencies where the transmission timely, quantitative data is of paramount importance, and is a useful framework on extracting and displaying useful emergency alerts with a geographical perspective based on volunteered citizen Tweets.

Keywords: *Twitter, Volunteered Tweets, Emergency-Related Data*

I. INTRODUCTION

The increasing pervasiveness of information and communications technology has radically transformed how people receive, access, and communicates information. One of the prominent trends which have contributed to said phenomena has been the advent of social media platforms. Social media is defined as a group of Internet-based applications that build on the ideological and technological foundations of Web 2.0 that allows the creation and exchange of user generated content [1]. The rising influence of these social media platforms has led to concerted studies on ways to leverage them for non-frivolous applications.

The web-based micro blogging service Twitter is a prime example of a social media platform whose usage has proliferated beyond casual social networking. The Twitter platform allows users to post messages, named as 'Tweets', consisting of text not more than 140 character, and subscribe to other users to view their Tweets as well. The rapidity, accessibility and ubiquity of Twitter are significant enablers of information exchange during time-sensitive and information-critical situations. A domain in which timeliness of data is of extreme importance is emergency communications; there has been recognition of the potentially significant role of Twitter in the emergency domain as reflected by increased efforts to involve Twitter in emergency management efforts. Emergent uses of Twitter in emergencies occurred during the 2007 Southern California wildfires, in which San Diego residents posted rapid-fire updates to Twitter on evacuation, meeting points, places of shelter and so forth [2].

In addition to being a channel for the straightforward peer-to-peer transmission of situational updates, Twitter is a potential tool to crowd-source emergency-related Tweets to enhance situational awareness during emergencies. Studies have been conducted on the possibilities of harnessing volunteered emergency data on Twitter in a more beneficial, structured manner. These studies vary from utilizing embedded geo-location features of Twitter in producing visualizations of emergency situations to standardized formats for tweet messages which support citizen data reporting, with the overarching goal of combining the applications of Twitter and emergency communications. This project provides a proof of concept model on how Twitter can be leveraged in emergency communications, by developing a Twitter-integrated web-system which will aggregate and process emergency-related data on the Twitter platform in order to provide a geo-visualization of emergency situations.

A. Problem Statement

A major challenge when encountering time-sensitive, information critical emergencies is to immediately source raw volunteered data from on-site public sources, which can be leveraged as emergency alerts to enhance emergency awareness from a geographical context.

`

Traditional media avenues such as television, newspapers, radio and short messaging systems are useful platforms to passively retrieve and disseminate emergency-related information to the community, but remain impractical as a means of rapidly collecting valuable emergency-based data, thus failing to deliver information representative of an emergency situation in a timely manner.

B. Objectives

The development of the proposed system desires to achieve the following objectives:

- Aggregate emergency-related situational data from "Tweets" by public citizen sources on the Twitter platform which fulfils a prescribed syntax format.
- Process the collected tweet data based on geo-location and syntax into organized information entities relevant to an emergency.
- Display the information on a map-like interface.

C. Scope of Study

The scope of data to be collected by the system is tweets on the Twitter platform which satisfy a pre-determined hash tag-based syntax on a particular emergency, in which the format is prescribed by an emergency management organization. The system seeks to query the Twitter API based on the aforementioned syntax, and does not intend to delve into sentiment analysis or sense-making for tweet content, nor is there any automated verification of the truthfulness of the supplied citizen account or the correctness of geographical information within the tweet.

II. LITERATURE REVIEW

A. Leveraging Social Media in the Emergency Domain

In examining the use of social media in the context of emergencies, there is a need to understand how social media technologies are specifically harnessed under such situations. A study of the emergent uses of social media during the 2007 Southern California Wildfire suggests that social media supports backchannel communications which is collectively resourceful and allows for generation of information which otherwise may not be easily obtained [3]. Web 2.0 technologies improve timeliness and access to emergency information, and have the positive attribute of providing greater context [4]. This is in stark contrast to traditional media outlets, which struggles to deliver immediate emergency information in a format which matches citizen needs. A new information pathway which connects official public information office functions with members of the public has emerged due to the need of an organizational destination for citizen-provided data in the context of heightened ICT-related public participation during crisis events [5].

B. Twitter Use During Emergencies

Twitter has been described as an avenue to provide information not covered on radio and television, such as details and first-hand accounts in the immediate aftermath of an event [6]. It is acknowledged that Twitter is not a complete emergency communications system which can perform management functions, but has the ability to, at the very least, function as a complementary tool during time and information-critical situations. While the broader Twitter audience tends to be interested in high-level accounts of an emergency event, re-Tweets of locals affected by an emergency have a tendency to be more specific, which can assist in the propagation of locally relevant information [7]. An example of an existing application which leverages Twitter in the emergency domain is the Twitter Earthquake Detector (TED), a software application supported by the U.S Geological Survey which seeks to mine real-time tweet data for indicators of earthquakes (http://recovery.doi.gov/press/us-geological-survey-twitter-earthquake-detector-ted/). TED showcases the rich possibilities afforded by social media in the context of integration with the emergency domain, such as the ability to pool together and make readily accessible citizen contributed information, as well as improving public outreach with regards to timely on-the-ground information [8]. However, Twitter possesses limitations as a citizen reporting system such as a potential lack of Tweets with quantitative properties. The presence of quantitative data is necessary in order to produce tactical information which has high utility during emergencies. Naturally, concerns will be raised on the legitimacy and validity of data sourced from citizens during emergencies as well. A study of the propagation of confirmed truths and false rumors on Twitter indicated that false rumors tend to be questioned much more than confirmed truths [9]. Any organization which chooses to co-opt social media data as a source of emergency information has a responsibility to state if the information has been corroborated.

C. Collecting and Processing Emergency-Related Data from Twitter

It is necessary to develop an understanding of potential strategies which can be applied in the collection and computational processing of Twitter data in order to maximize the utility of extracted emergency-related information. A study on hazard-based data sets on Twitter cautions that the design and implementation of software systems that employ information extraction strategies during emergencies has to be informed by analysis of specific features of information generated under such circumstances [10]. The study also states that geo-location, location referencing and situational update data are prevalent entities in Tweets on emergencies which tend to aid automated retrieval of relevant information in a specific emergency event.

The presence of a built in geo-location function on Twitter means that Tweets have the capacity to be location-precise, but the drawback is that the enablement of the function is at the discretion of individual users. To enhance the input provided by Twitter during emergencies, various alternative ways in which less explicit geographic information may be extracted are hash tags, locations stored in user profiles, and location references in tweet content [11]. The high volume and noisy nature of data provided by people on the ground during emergency events poses a sense-making problem, but instead of training computer algorithms to extract emergency-related information from Tweets, an alternative approach to "train" Twitter users to craft such Tweets in a machine-readable format is suggested [12]. This approach represents a model of citizen sensing which places a key emphasis on parsing user-structured messages which originate from the Twitter platform. Emergency message syntax should be structured in such a way that it consists of only defined elements relevant to the emergency itself. **Figure 1** [13] illustrates examples of a prescriptive tweet structure. This research adopts the assumption that citizens are capable of being trained to use emergency-specific Twitter hash tag syntax, which allows them to upload information in a machine-readable form. By marking-up Tweets based on a prescribed format, computational filtering and classification of such tweet data will be greatly simplified.

EXAMPLE1: #haiti #imok #name John Doe #loc Mirebalais Shelter #status minor injuries
EXAMPLE2: #haiti #need #transport #loc Jacmel #num 10 #info medical volunteers looking for big boat to transport to PAP
EXAMPLE3: #haiti #need #translator #contact @pierrecote
EXAMPLE5: #haiti #ruok #name Raymonde Lafrotune #loc Delmas 3, Rue Menelas #1
EXAMPLE4: #haiti #ruok #name Camelia Siquineau #loc Hotel Montana
EXAMPLE6: #haiti #offering #volunteers #translators #loc Florida #contact @FranceGlobal

Figure 1: Examples of Emergency-Specific Tweet Syntax

III. METHODOLOGY

A. Project Development Methodology

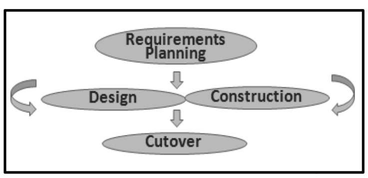

Figure 2: Phases of Rapid Application Development

The development methodology chosen for this project is Rapid Application Development (RAD) as shown in **Figure 2**. At the core of the RAD methodology is iterative development and prototyping techniques with the aim of accelerating system development. The primary reason of choosing RAD is to enhance flexibility in the face of time constraints. RAD incorporates iterative development practices which interleave the design and construction of software, compensating for lack of extensive pre-planning time. This makes it easier for the system developer to continually refine the system model and produce improved iterations at various time checkpoints over the course of the project. A potential weakness of the methodology is that certain compromises in functionality and performance may be entailed to complete the project in the stipulated time.

B. System Architecture

Figure 3 represents the overall system architecture which provides a high-level overview of the various components and data flow within the system. The web system consists of 3 major processes: Aggregate, parse and display. The 140dev Twitter Framework is the backbone of the system which aggregates and parses raw data from the Twitter platform, and the tweet data is further processed with custom algorithms for the system's customized needs before being stored in a MySQL database. The tweet aggregation architecture is designed in a way that the website does not interact directly with the Twitter Streaming Application Programming Interface (API), with tweet data being delivered to the website via an intermediary database. This is to reduce the chances of system failure should the Twitter Streaming API encounter problems. The website subsequently embeds Google Maps API functionality combined with tweet data extracted from the database to deliver a visualization of the emergency information.

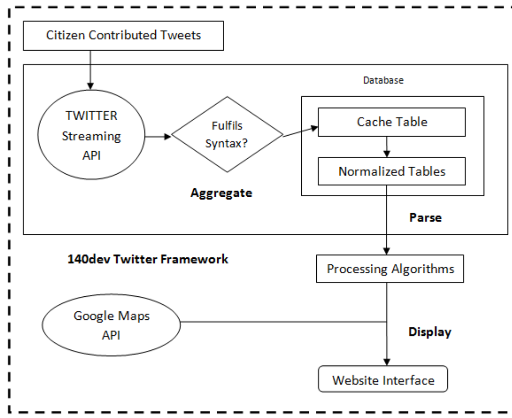

Figure 3: System Architecture

C. Development Tools Required

6) XAMPP
7) Twitter API
8) 140dev Twitter Framework
9) Google Maps API
10) Adobe Dreamweaver CS5

IV. RESULTS AND DISCUSSION

This chapter contains 5 chapters which details the use case diagram, aggregation process, parsing process, display process and the results of a system usability scale survey. Each of these sections will be discussed.

A. Use Case Diagram

The main actors in the system are the:

- **Administrator:** The administrator is responsible for determining the keywords to control which Tweets are collected from the Streaming API. The keywords will be based on a prescriptive syntax format for an emergency happening.

- **End-user:** The end-user represents any user who enters the publicly accessible web interface. They can control the results displayed on the map interface by using the filtering criteria via the select boxes.

- **On-site citizen:** The on-site citizen may be either a part of or casual observer of the emergency who

`

provides volunteered Tweets containing emergency-related data. The on-site citizen has to construct the tweet on such a way that it fulfills the prescribed syntax format.

- **Monitoring agent:** The monitoring agent represents a member of the emergency management organization implementing the system who monitors incoming Tweets for events. The agent plays a role in verifying the tweet data received. If the information is corroborated by reliable sources, the corresponding tweet will be marked as verified in the results on the web interface.

Figure 4 represents the use case diagram of the Twitter-integrated web system for emergency alerts.

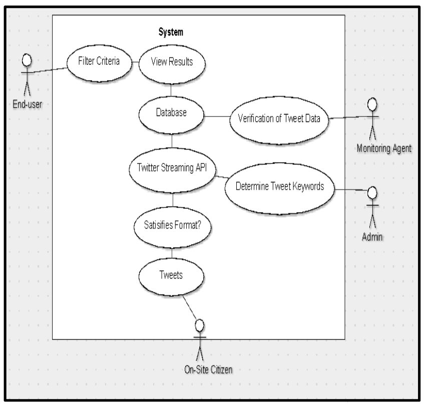

Figure 4: Use Case Diagram

B. Aggregation Process

The system design relies on the Twitter platform as the main source of data input for the system. The username and password of a valid Twitter account is configured within the Phirehose Library, which uses basic authentication to establish and maintain a connection to the Twitter Streaming API. A set of keywords are provided to a *get_tweets.php* script to filter the incoming Tweets, with the keywords based on the prescribed syntax for emergency events. The proposed tweet format is as follows:

#[keyword] #[alert type] #dct [description] #loc [location]

With the exception of the compulsory hash tags, the tweet format above can be tweaked to cater to different emergencies which have different alert types, with differing description and location. Below are two examples of tweets which cater to two different use cases and fulfill the format above.

#thaifloods #flood #dct 0.5m deep #loc Bangkok Thailand

#hurricanephilly #shelter #dct 20 spaces #loc Philadelphia US

There are four main elements present in the tweet syntax, namely the keyword, alert type, description and location. The keyword hash tag serves as a unique identifier of which emergency event the tweet is referring to. The alert type hash tag determines the category of alert within the event, while the #dct tag precedes a text description of the alert. Twitter allows a user's Tweets to be automatically geo-tagged, provided that the user enables the option. The purpose of the location tag #loc is to allow geo-coding to be performed on the location text should geo-tagging functionality be disabled by the tweet user. If geo-tagging is enabled, however, the geo-tagged location takes precedence over the location specified in the tweet text when processed by the system. The length of the description and location text is solely at the discretion of the user, as long as it is within the 140 character limit enforced on Twitter messages. The location text should ideally be as specific as possible in the event that geo-tagging is not enabled, to avoid wrong geo-coding results for locations which may share identical names.

The script for tweet collection is initiated as a background process, where raw tweet data which satisfies the filter parameters is returned in the JSON format and the full JSON payloads are added to the cache table in the MySQL database. **Figure 5** is a data flow diagram of the aggregation process.

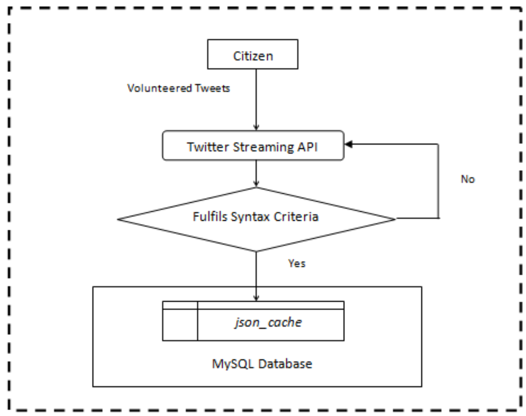

Figure 5: Data Flow Diagram for Aggregation Process

C. Parsing Process

The next process is the parsing process, which parses each raw tweet in the *json_cache* table into readable tweet data, to be transferred into a *tweets* table. The aggregation and parsing processes are decoupled to ensure that the aggregation code does not perform heavy processing tasks on tweet data which arrives at a fast rate, thus reducing the chances of losing incoming Tweets. P*arse_tweets.php* in the 140dev Twitter Framework is initiated as a background process to extract tweet text as well as metadata such as latitude and longitude from the raw tweet payload. Once a raw tweet in the cache table has been processed it will be marked as parsed, so subsequent parsing processes will bypass that payload. The *tweets* table is connected to a *processquery.php* script which contains algorithms designed to perform programmatic operations on the tweet text to extract the relevant informational entities within the tweet. The algorithm behaves strictly in the sense that it does not tolerate faulty spacing, spelling mistakes or incorrectly-ordered tags which cause the message to differ from the prescribed syntax. Any formatting error within the tweet will result in *processquery.php* disregarding it. **Figure 6** is a data flow diagram of the parsing process.

`

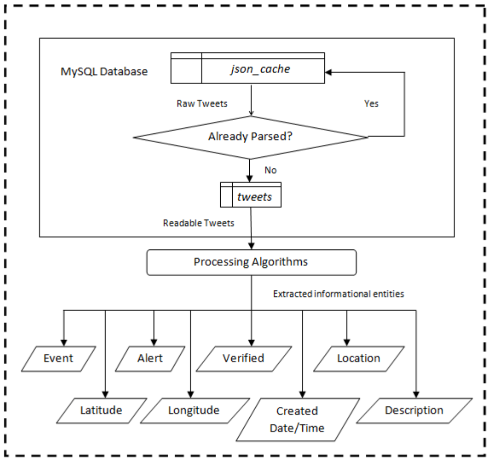

Figure 6: Data Flow Diagram for Parsing Process

D. *Display Process*

The information extracted from the aggregation and parsing processes is delivered to a visual interface during the display process. A web page which mashes up a Google Map and tweet data is created. Select boxes are coded into the web interface to allow the user to filter or sort the emergency information displayed. The attributes which can be filtered and sorted are as per data flow diagram in **Figure 7**. A query is constructed based on the filter parameters provided, and the results are exported to the web page via *processquery.php*. The output data consists of informational entities specified in the prescribed syntax format. They include the type of alert, description, and location, accompanied by the created date/time of the tweet and a verification status to indicate if the tweet has been corroborated. The extracted tweets are subsequently placed as marker points on the map with their respective information windows, as per **Figure 8.** The marker points are color coded to distinguish between different types of alerts on the map. The location of the marker point is dependent on whether the tweet is geo-tagged. If geo-tagged functionality is switched on, the latitude and longitude metadata within the tweet will take precedence, otherwise the geo-coding result based on the location text provided in the tweet will be utilized. The interface also has a sidebar to list color codes and emergency alerts which correlates with the map markers, with refresh and reset buttons available to refresh the displayed content and to reset the map and select boxes to its default settings, as per **Figure 9**.

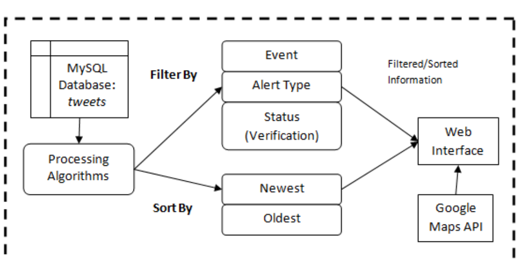

Figure 7: Data Flow Diagram of Display Process

Figure 8: Tweet Information Window for a Map Marker Point

Figure 9: Sidebar List of Emergency Tweet Information

E. *System Usability Scale Survey Results*

A system usability scale (SUS) survey was conducted to measure the perception of usability for the web interface of the developed system, where a total of 20 respondents were surveyed. The questionnaire consisted of a ten-item attitude Likert scale to gauge the user's perception of system usability. Once all responses were obtained for each item, scores for each question were scaled from 0 to 4. For odd numbered questions 1, 3, 5, 7 and 9, the score contributions is the scale position minus 1. As for the even-numbered questions 2, 4, 6, 8 and 10, the score contribution is 5 minus the scale position. The tabulated results of the survey are as per **Table 1**.

Table 1: System Usability Score Survey Results

No.	Question	Average Score
1	I think that I would like to use this system frequently.	3.1
2	I found the system unnecessarily complex.	3.65
3	I thought the system was easy to use.	3.4
4	I think that I would need the support of a technical person to be able to use this system.	3.7
5	I found the various functions in this system were well integrated.	3.15
6	I thought there was too much inconsistency in this system.	3.5
7	I would imagine that most people would learn to use this system very quickly.	3.15
8	I found the system very cumbersome to use.	3.45
9	I felt very confident using the system.	3.1
10	I needed to learn a lot of things before I could get going with this system.	3.5
	Sum Score	33.7
	Overall Score	84.25

Average scores for each question were added up and the total multiplied by 2.5. This converts the range of possible values from 0 to 100 instead of from 0 to 40. As

`

the sum of the score was 33.7, the resulting overall SUS score is:

33.7 x 2.5 = 84.25

A score of **84.25** was obtained, which marks positive usability as per SUS conventions.

V. CONCLUSION

Utilizing Twitter to crowd-source data on emergencies allows for immediacy in generating information which cannot be matched by traditional media outlets. Despite contentions that Twitter accounts of events lack rigorous fact checking, Twitter's ubiquity, ease of use and the rapidity of micro-blogging communications means that it is still an ideal tool to collect volunteered data during time-sensitive emergencies where first-hand accounts are extremely valuable. The Twitter-integrated web system developed in this project is envisioned as a solution which is able to automate the aggregation of emergency-related tweets available on social media platforms and organize it to produce an emergency-specific informational account. The system is best for use in emergencies where the transmission timely, quantitative data is of paramount importance. It is hoped that the project can provide a framework on extracting and displaying useful emergency alerts with a geographical perspective based on volunteered citizen Tweets, and ultimately contribute to the domain of social media-assisted emergency applications.

VI. RECOMMENDATIONS

A recommendation for the project would be to enhance the quality of input received by the system. The aggregation and parsing algorithms would need to be strengthened to ensure that redundancy of citizen-provided data does not occur. The system may also be integrated with formal emergency-based organizations and location-sensitive devices to include alert data from authoritative sources, to ensure that the data is more reliable. The prototype has been developed as a web system to be accessed on desktop browsers, so in order to provide flexibility to end-users who wish to view the interface on a mobile device, the entire system may be rebuilt as a dedicated application on a mobile platform.

VII. ACKNOWLEDGMENT

The writer would like to take this opportunity to express his greatest gratitude and appreciation to project supervisor, Prof. Dr. Alan Oxley, who had continuously monitored his progress throughout the duration of the project. His constructive advice and suggestions have guided the project towards its successful outcome. Gratitude is also expressed towards Universiti Teknologi PETRONAS (UTP), especially the committee of the Final Year Project of the Computer Information Sciences (CIS) department for their excellent management of this course. The writer would also like to thank all the respondents which participated in the survey initiated by the writer as part of the project.

REFERENCES

[1] Kaplan, A.M., Haenlein, M. (2010). "Users of the world, unite! The challenges and opportunities of Social Media". *Business Horizons* 53(1): 59–68.

[2] Poulsen, K. (2007). Firsthand Reports From California Wildfires Pour Through Twitter. Retrieved from: http://www.wired.com/threatlevel/2007/10/firsthand-repor/

[3] Sutton, J., Palen, L. & Shklovski, I. (2008). Backchannels on the Front Lines: Emergent Uses of Social Media in the 2007 Southern California Wildfires. *Proceedings of the 5th International ISCRAM Conference – Washington, DC, USA, May 2008*, (pp. 624-631).

[4] Van Leuven, L. (2009). *Optimizing Citizen Engagement During Emergencies Through the Use of Web 2.0 Technologies*. Master's Thesis, Naval Postgraduate School, Monterey, CA.

[5] Palen, L. & Liu, S.B. (2007). Citizen Communications in Crisis: Anticipating a Future of ICT-Supported Public Participation. *Proceedings of the SICGHI Conference on Human Factors in Computing Systems, April 28 - May 3, 2007,* (pp.727-736).

[6] Mills, A., Chen, R., Lee, J.K., & Rao, H.R. (2009). Web 2.0 Emergency applications: How useful can Twitter be for emergency response?. *Journal of Information Privacy & Security* 5(3): 3-26.

[7] Starbird, K. & Palen, L. (2010). Pass It On? Retweeting in Mass Emergency. *Proceedings of the 7th International ISCRAM conference – Seattle, USA, May 2010.*

[8] Guy, M., Earle, P., Ostrum, C., Gruchalla, K. & Horvath, S. (2010). Integration and Dissemination of Citizen Reported and Seismically Derived Earthquake Information via Social Network Technologies. In P.R. Cohen, N.M. Adams, and M.R. Berthold (Eds.)*: IDA 2010, LNCS 6065, 2010,* (pp. 42–53).

[9] Mendoza, M., Poblete, B. and Castillo, C. (2010). Twitter Under Crisis: Can we trust what we RT? , *1st Workshop on Social Media Analytics (SOMA '10), Washington, DC, USA.*

[10] Vieweg, S., Hughes, A.L., Starbird, K. & Palen, L. (2010). Microblogging During Two Natural-Hazard Events: What Twitter May Contribute to Situational Awareness. *Proceedings of the 28th International Conference on Human Factors in Computing Systems, 2010,* (pp. 1079-1088).

[11] MacEachren, A.M., A.C. Robinson, A. Jaiswal, S. Pezanoski, A.Savelyev, J. Blanford and P. Mitra (2011). Geo-Twitter Analytics: Applications in Crisis Management, *Proceedings of the 25th International Cartographic Conference, Paris, France*

[12] Starbird, K. & Stamberger, J. (2010). Tweak the Tweet: Leveraging Microblogging Proliferation with a Prescriptive Syntax to Support Citizen Reporting. *Proceedings of the 7th International ISCRAM Conference – Seattle, USA, May 2010.*

[13] Boyd, S. (2010). Disaster Microsyntax: Project EPIC, Tweak the Tweet, and Emergency Codes. Retrieved from: http://stoweboyd.com/post/786084841/disastermicrosyntax