

2008

University of North Carolina Wilmington
Master of Science in
Computer Science and Information Systems
Proceedings

<https://csbapp.uncw.edu/mscsis>

MS CSIS Capstone Research Project

CommunityTips.org:
A System for Anonymous Tipping

By

William J. Shipman

The University of North Carolina Wilmington

Wilmington, North Carolina

October 2008

Project Committee

Dr. Karl Ricanek (Chair)

Dr. Laurie Patterson

Dr. Bryan Reinicke

Dr. Ron Vetter

Table of Contents

List of Tables	iv
List of Figures	v
Acknowledgements	vii
1.0 Introduction	1
2.0 Background	5
2.1 Prior Work	5
2.1.1 Mobile Biometrics	5
2.1.2 Reminder Service	6
2.2 Community Policing Programs	6
2.2.1 Cheshire Public Policing Information System	7
2.2.2 Citizen Observer	7
2.2.3 Crimeline	13
2.2.4 Amber Alert	15
3.0 Methodology	16
3.1 Analysis	16
3.1.1 Choice of Methodologies	16
3.1.2 Desktop Application versus Web-based Application	19
3.1.3 Components Off The Shelf (COTS) versus Custom Development	20
3.1.4 Web-based Application Development Framework	21
3.1.5 Choice of Database	23
3.2 Technologies Used in Development	24
3.2.1 Selection of Environment	24
3.2.2 Cascading Style Sheets	25
3.2.3 jQuery	26
3.2.4 Axis 2	26
3.3 Implementation	26

3.3.1	Database Design	26
3.3.2	Web Application Interface	29
3.3.3	Messaging Interface	38
4.0	Discussion	39
4.1	Comparison To Existing Systems	39
4.1.1	Citizen Observer	40
4.1.2	Crimeline	42
4.1.3	Amber Alert	43
4.2	Response From Law Enforcement	45
4.3	Response From Potential Users	47
5.0	Conclusions and Future Work	52
5.1	Conclusions	52
5.1.1	Project Summary	52
5.1.2	Lessons Learned	54
5.2	Future Work	56
	References	57
	Appendix A: System Entity-Relationship Diagram	59
	Appendix B: Test Plan	60
	Appendix C: Test Cases and User Survey	61
	Appendix D: Note to Users	78

List of Tables

Table 3.1 Description of key user generated data database tables	29
Table 5.1 Summary of key features in online community policing systems	53

List of Figures

Figure 1.1 Messaging System Diagram	3
Figure 2.1 Mobile Biometrics System Architecture	6
Figure 2.2 Citizen Observer™ Suspect Description Form	8
Figure 2.3 Citizen Observer™ Vehicle Information Form	8
Figure 2.4 Citizen Observer™ Crime Notes Form	9
Figure 2.5 Citizen Observer™ Drugs Form	10
Figure 2.6 Citizen Observer™ Upload A Photo Form	10
Figure 2.7 TipSoft data reception and processing	11
Figure 2.8 Citizen Observer's™ SMS tip processing interface	12
Figure 2.9 Crimeline Identify a Person Section	13
Figure 2.10 Crimeline Address of a person section	13
Figure 2.11 Crimeline Address of the crime location section	13
Figure 2.12 Crimeline Description of a victim section	14
Figure 2.13 Crimeline addition information section of a crime tip	14
Figure 2.14 Crimeline CAPTCHA function	14
Figure 2.15 Amber Alert™ SMS Example Broadcast	15
Figure 3.1 Waterfall Model of Software Development	18
Figure 3.2 Spiral Model of Software Development	18
Figure 3.3 Entity relationship diagram of core tip related tables	28
Figure 3.4 Initial Tip Form	29

Figure 3.5 Tip Form with Contact Information	29
Figure 3.6 CommunityTips.org Tip Verification	30
Figure 3.7 CommunityTips.org Describe a person involved form	31
Figure 3.8 CommunityTips.org Describe the place the event involved form	32
Figure 3.9 CommunityTips.org Describe the vehicle if one was involved form	32
Figure 3.10 CommunityTips.org Basic Tip Search Interface	33
Figure 3.11 CommunityTips.org Tip Search With Date Limit	33
Figure 3.12 CommunityTips.org Search Using Personal Details	34
Figure 3.13 CommunityTips.org Search Using Location Details	34
Figure 3.14 CommunityTips.org Search using Vehicle Details	34
Figure 3.15 CommunityTips.org Tip Search Results	35
Figure 3.16 CommunityTips.org Event Details for tip 118	36
Figure 3.17 CommunityTips.org function to add meta-data about an image or video	37
Figure 3.18 CommunityTips.org Automated Textual Translation of tip 105	27

Acknowledgements

First, I must thank my family, particularly Mom and Dad (Marilyn and Dennis) and Grandma Hazel for all their love, support, and all that they have taught me in life, which has brought me here, the completion of my Masters in Computer Science and Information Systems. Also my brothers (Ed and Jim) for always keeping their youngest brother on track and my sister-in-law, Dierdre, for reading and reviewing this document among many. Without my family this task would have not been possible, to them more credit is owed than simple words of thanks I can offer here.

Second my committee for guiding me through this process. My advisor, Dr. Karl Ricanek Jr., a thousand thanks are due for keeping me focused on the end goal, sushi lunches, and guidance in so many different areas. My other committee members, Drs. Laurie Patterson, Bryan Reinicke and Ron Vetter, thank you for your direction, good humor, and always having an open door.

Third, my friends and classmates thank you for many meals, a hundred questions answered, documents edited, and for just being yourselves. You have enriched the last three years more than you know.

Fourth, last but certainly not least, thank you to the Computer Sciences Department and Information and Operations Management Department faculty and staff.

1.0 Introduction

Connectivity is a driving force in American life today and in the world at large. This can be seen in many areas including reality TV where people connect to the reality of the situations they encounter, and the successful social networking sites such as MySpace and Facebook. The desire to “reach out and touch someone” is demonstrated by the myriad of mobile devices one might find strapped to a hip, in a purse or in a pocket. The cell phone represents one kind of mobile device, which can include personal digital assistant functions as well as two-way paging.

The proliferation of the cell phone is highlighted in a February 2007 report by the Federal Communication Commission. There are 217 million cell phone subscribers which is equivalent to approximately two-thirds of the total United States population[18]. Cell phones are used by all segments of modern society. Some reports estimate that seven to nine percent of users have only a cell phone, with no primary landline phone service [3]. There are a variety of reasons that can be attributed to this shift in the usage of cell phone service, including cost of ownership, flexibility of communication (voice, chat, email, web, etc.), and as a tool for physical security. The Pew Foundation, a non-profit research organization, has reported then need to use sampling techniques not previously required for landline surveys to get correct proportions of age and gender to represent the current American population. They attribute the shift in usage patterns of landline versus cell phones: “On many variables, the landline sample was closer to the population parameter than the cell sample, though on some measures the cell sample picks up certain kinds of respondents that the landline samples under-represent”[3]. Specifically the Pew Foundation found landlines over-represented older segments of the population and under-represented younger segments, while cell phones showed the opposite. Cell phones are also seen in the workplace across virtually all industries, including medical and law enforcement. In k-12 schools there has also been a dramatic shift as parents equip their children with phones [1], to better track their movement or to provide a sense of safety. A paper published in the *Journal of Information and Management* provides statistics of SMS (Short Message Service) utilization; on a per-month basis the 11 to 17 years old age group accounts for 27% of texters while the 18 to 25 years old age group account for 43% [2]. Combined 11 to 25 years old accounts for 70% of texters in the survey group.

This project was motivated by a desire for citizens to be more engaged in their community. For example, citizens should be able to send and receive information pertaining to their community through their cell phone. Cell phones typically support one or more of the following services: voice, SMS, multimedia messaging (MMS), instant messaging (IM), Internet access, and/or email. Often these devices have multimedia capabilities that include a still camera, microphone, and video recording. These capabilities are being used in a wide variety of arenas. The popular United States prime time television show, *American Idol*, for example utilizes SMS messages to allow television viewers to vote using different short codes for the program contestant they preferred best. Popular social networking sites such as Facebook™ and Twitter™, allow registered users to post updates or interact with the site utilizing mobile web-browsers or SMS. Google™ provides a wide variety of interactive information tools via SMS including weather, currency conversion, airport information and more [8].

According to the Wireless Communications and Public Safety Act of 1999, wireless carriers are required to support emergency 911 (e911) calling as well as provide geographical coordinates of the calls origin. While extracting exact geographic coordinates of a cell phone user is not generally available to the public, carriers are nonetheless starting to explore location-based services, as most next generation handsets will have GPS capabilities. Given these facts, a large percentage of the population now has the ability to provide direct assistance to law enforcement by means of cell phones. This assistance can come in the form of SMS, MMS, internet forms (web), and/or standard direct dialing (voice calls).

Key terminology used in this paper include SMS/MMS, messaging, aggregator, short code, and push/pull style messaging. SMS, Short Messaging Service, or more colloquially “texting”, is a service offered by most major cellular providers to enable users to send brief (usually a maximum of 160 characters) messages. SMS messages are handled via a Short Message Server Center (SMSC) maintained by the cellular provider to terminate SMS messages. The Short Message Peer to Peer (SMPP) protocol has been adopted by the SMS Forum to enable SMSC’s to interact within and across cellular network providers, and also with third party content providers. MMS, Multimedia Messaging Service is simply a second version of the SMS protocol that allows users to send binary content (images or short videos) in addition to a brief message. The general term *messaging* indicates that either SMS or MMS messaging is

applicable. An *aggregator* is an intermediate party between network carriers SMSC's and external entities that provides services utilizing cellular messaging. Figure 1.1 depicts the general system architecture for sending and receiving SMS messages.

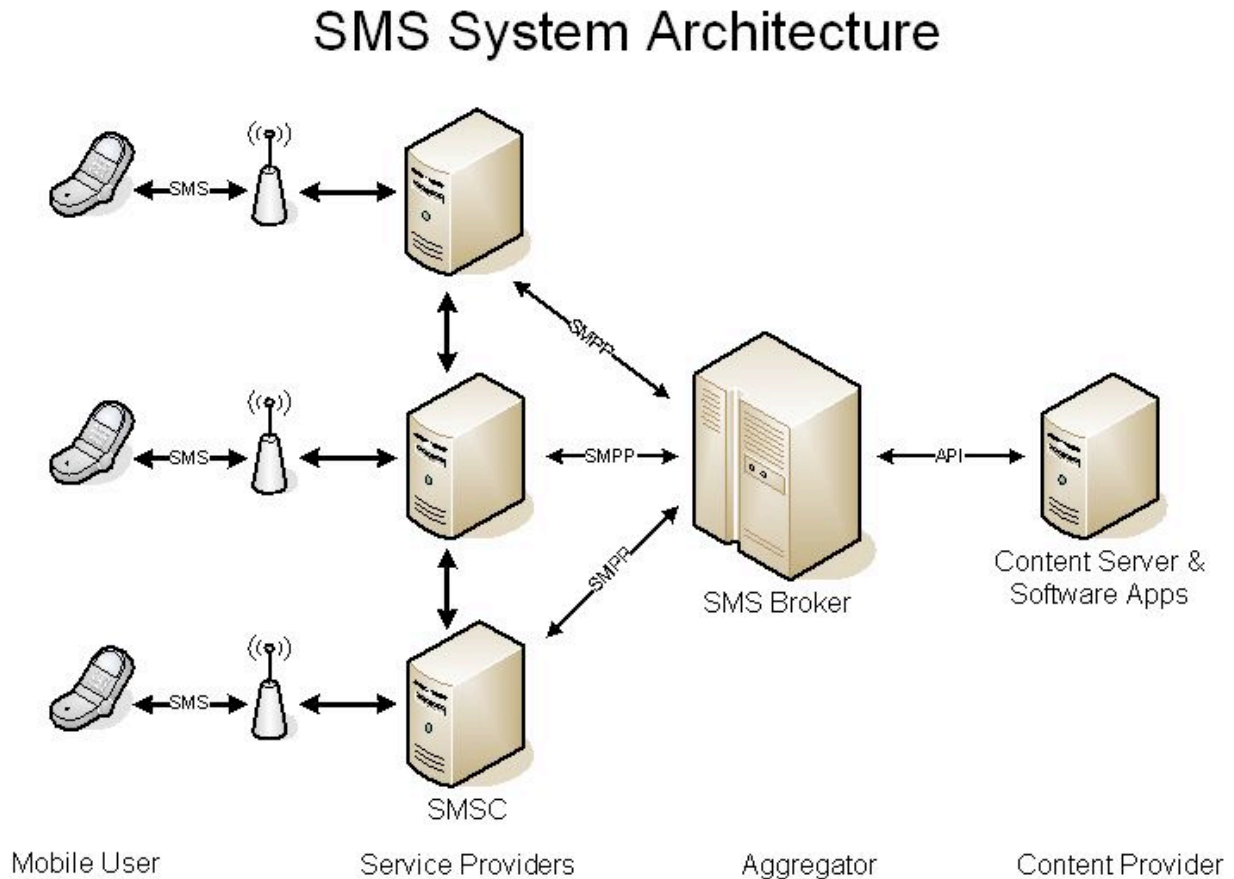


Figure 1.1 Messaging System Diagram [21]

A short code is a five or six digit number recognized by major domestic cellular providers as a unique individually-held number similar to a phone number. The codes are used for sending and receiving messages that do not originate from cell phones.

The terms push and pull are used to describe what form of interaction is taking place. In this situation cell phones are always clients, a push describes sending data across a cellular network to the client. A pull is the process of getting the client to submit information to a system via messaging.

Many public and private entities are poised to leverage the proliferation of cell phones and messaging services for a variety of commercial and public interests. A few examples of

commercial uses of SMS are individual and fleet vehicle tracking, managing personal and business contacts, and providing interactive information services [21]. This proliferation in the commercial sector has enticed public entities to ask, how can this technology benefit their mission.

Law enforcement is a highly visible public service, which is very interested in new modes of communications with the community. Community policing programs regularly solicit information from the public in a wide variety of ways ranging from community watch programs, supports active police operations such as Amber Alert™ and anonymous tip lines. These programs support both community and law enforcement needs and objectives to collect and disseminate appropriate information. A valuable asset would be to allow citizens to communicate with the police department through readily available technology, which a large percentage of the community already owns: mobile devices, and more specifically cell phones.

Community Tips (www.communitytips.org) is a project that produced a comprehensive system for anonymous tips via SMS and a companion web portal. The web portal has a publicly accessible interface for community members to learn about the system as well as to submit tips. Users can provide additional information beyond uploading images and tips to include information about specific persons, vehicles, and locations involved as well as personal identifying information. In addition, the web portal has a law enforcement only section where registered law enforcement agencies login to search and review tips, provide comments and revisions (without altering the original tip) and link related tips together.

The layout of this paper is as follows: Section 2 Background, Section 3 Implementation, Section 4 Discussion, Section 5 Conclusions and Future Work. The Background Section provides information regarding foundational experiences dealing with mobile communications and other electronic systems used in community policing. The Implementation Section covers: Project Analysis (3.1), Technologies Used to build the application (3.2), and Final Implementation (3.3). The Discussion Section deals with choices made in development, responses both from law enforcement and potential users, and comparisons to existing systems. Conclusions and Future Work provides a summary of the project as a whole and lessons learned, in addition to a description of how the system can be expanded and other possibilities in community policing.

2.0 Background

2.1 Prior Work

There are two examples of direct prior work in the area of user centric projects involving SMS within the University of North Carolina Wilmington's Department of Computer Science. The first was a collaborative project demonstrating the integration of biometrics with a mobile device for access control (collaborators are Ms. Amy Curley and Mr. Jaewoong Kim). The second is an example of an information system providing dynamic reminders via a text message to a subscriber's phone created by Bill Shipman—this system was coined the Reminder Service.

Other works that must be considered are efforts to integrate cell phones and SMS into community policing efforts. Four examples are included, two of which involve pushing information out to community members via SMS: one in Cheshire, England and the other is Wireless Amber Alerts. The other two systems are aimed as gather tips from the community using SMS. These are: Citizen Observer used in several municipalities within the United States and Crimeline which is utilized in South Africa.

2.1.1 Mobile Biometrics

Ms. Curley's work on a mobile biometric system is illustrated in Figure 2.1. In this system, a user takes a self-portrait on any standard camera enabled cell phone and the image is then sent via email to the system for processing. A numeric ID is embedded in the message, which aids in the authentication process. In other words, it is both something the user knows (the numeric ID) and something the user is (physical characteristic). A user submits both pieces of information via an MMS message and the system will analyze the digital image in conjunction with the numeric ID to determine if the picture/ID pair is valid. This type of biometric identification is known as a verification system. This system was implemented before there was an avenue for any form of messaging support; thus it was entirely implemented using email. The goals were to implement and demonstrate the possibilities associated with marrying biometrics to mobile devices.

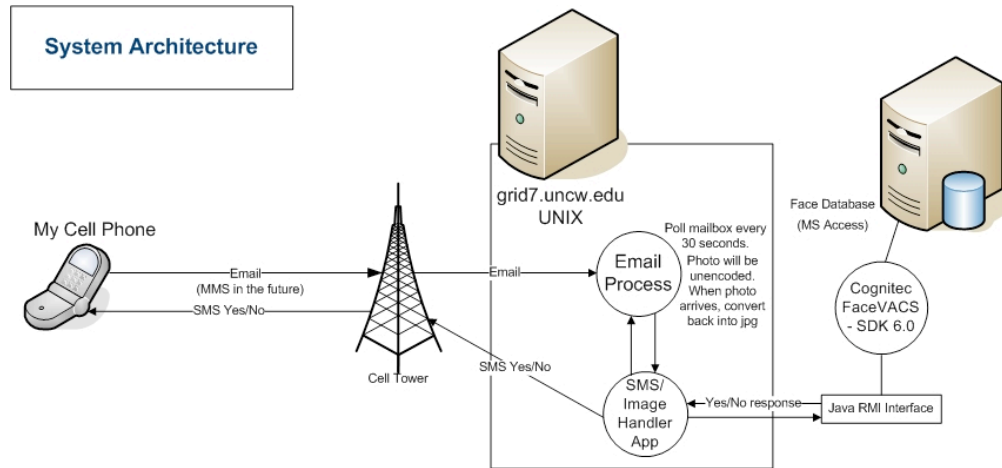


Figure 2.1 Mobile Biometrics System Architecture.

2.1.2 Reminder Service

The Reminder Service was a service that allowed users to pre-schedule SMS messages for delivery to their personal cell phone. The motivating idea for the project was that people are forgetful creatures, but modern technology has provided a perfect tool to aid us – cell phones. Innately, cell phones can do little to aid human memory but when coupled with the ability to send and receive pre-scheduled customized messages they become a powerful tool. In the implementation, the Reminder Service had a simple web page whereby a user scheduled a customized message at a pre-determined time to be sent to their cell phone. A real world example would be a simple reminder at the end of the workday to pick up the dry cleaning. The Reminder Service project has some practical application, but it primarily was an experiment in creating an application to interact with cell phones via SMS. The project was limited by technical issues revolving around SMS transmission. A key benefit was the development of a cell phone verification process to authenticate cell phone users. This model has been adopted by the University of North Carolina Wilmington to ensure that their SMS services are not being used inappropriately. (<http://www.uncw.edu/mobile>).

2.2 Community Policing Programs

2.2.1 Cheshire Public Police Information System

In Cheshire, United Kingdom, a local police department has a pilot project using text messaging as a way of communicating public safety information to the public. This is part of a

larger effort aimed at building and strengthening the relationship between law enforcement and the community it serves [13]. A strong parallel could be drawn to a program by the Wilmington North Carolina Police Department. The City of Wilmington Public Information Report [5] details the aims of a community out-reach program which include providing geographic policing with the lieutenant in charge on foot in their respective areas at least one hour per shift. One way to promote dialog between teenagers and law enforcement is to make law enforcement accessible through channels teenagers are comfortable using, such as SMS and MMS.

The Cheshire project is not extensively detailed, however it describes a push style system for providing community alert/informational notifications. This type of system can be viewed as providing overlap and gap coverage to community members who may not otherwise receive this information.

2.2.2 Citizen Observer

Citizen Observer™ is a commercial product for law enforcement officers to engage in electronic communications with the community they serve through a single entry point. It can provide web-based postings, email, and SMS alerts. In addition it allows community members to submit anonymous tips through a web form that provides a report of the incident they witnessed. The web form has five distinct sections: Suspect Description, Vehicle Description, Crime Notes, Drugs, and Photo Upload.

The first section, Suspect Description, asks for a basic biographical description that includes physical description, height, weight, scars / tattoos, eye color, and clothing as well as home address and first/middle/last names. It also asks about suspects' animals, hangouts, weapons, gang affiliations and employment. Figure 2.2 shows an example of Citizen Observer™ web tip for the Lebanon Tennessee Police Department.

Lebanon Police Department Confidential Web tip Information System If this is an urgent matter do not submit it here alone. Dial 911 if it is an emergency! Fill out as much information below as possible and click "Submit Tip".			
NUMBER OF SUSPECTS 1 <input type="button" value="Go"/>			
Suspect #1			
First Name:	Middle Name:	Last Name:	Alias:
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
Race:	Sex:	Height:	Weight:
<input type="text" value="Unknown"/>	<input type="text" value="Unknown"/>	<input type="text" value="Unknown"/>	<input type="text" value="Unknown"/>
Age (or DOB):	Hair:	Eyes:	
<input type="text"/>	<input type="text"/>	<input type="text"/>	
Suspect's Address, City, State, Zip Code, Country:			
<input type="text"/>			
Scars, Marks, Tattoos:			
<input type="text"/>			
Suspect's Clothing:			
<input type="text"/>			
Dogs or Animals:			
<input type="text"/>			
Weapons:			
<input type="text"/>			
Hangouts:			
<input type="text"/>			
Known Associates:			
<input type="text"/>			
Gang Affiliation:			
<input type="text"/>			
Other Suspect Notes:			
<input type="text"/>			
Suspect's Employment Information:			
<input type="text"/>			
Employer's Address, City, State, Zip Code, Country:			
<input type="text"/>			
Where did you last see this suspect?			
<input type="text"/>			

Figure 2.2 Citizen Observer™ Suspect Description Form [23]

The second section, Vehicle Information, asks for specific information about a single vehicle involved in the tip. The form has fields for users to provide vehicle Make, Model, Color, Year, License, State and other notes. Figure 2.3 is an example of the Vehicle Information section.

VEHICLE INFORMATION			
Make:	Model:		
<input type="text"/>	<input type="text"/>		
Color:	Year:	License:	State:
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
Other Vehicle Notes:			
<input type="text"/>			

Figure 2.3 Citizen Observer™ Vehicle Information Form [23]

The third section, Crime Notes, asks for general information about the crime, including a basic description of the crime, location of the crime, and information about the victim, as well as

specific information such as the offense, Warrant Number, and Case Number. Figure 2.4 is an example of the Crime Notes section.

CRIME NOTES

Type of Offense: Warrant Number: (if known)

Offense City: Case Number: (if known)

Victim's Information:

Crime Description: (Including... Who, What, When, Where and How Do You Know)

How did you become aware of this issue?

It happened to me directly

I observed it happen

I was told by someone

Other

If this tip is related to a scholastic issue, we need the following information:

Campus Name City

Figure 2.4 Citizen Observer™ Crime Notes Form [23]

The fourth section, Drugs, asks for any information regarding drugs which were involved in the crime. General information is asked about drug usage along with questions concerning packaging, transactional information, and place of sale. Figure 2.5 is an example of the Drugs section.

The fifth section which is optional, Upload A Photo, allows a user to submit one photo, either in GIF or JPG format, up to 4 megabytes in size. Figure 2.6 is an example of the Upload A Photo section.

Figure 2.5 Citizen Observer™ Drugs Form [23]

Figure 2.6 Citizen Observer™ Upload A Photo Form [23]

The other key feature of Citizen Observer™ is it enables law enforcement to receive tips via anonymous SMS messages. Tips are submitted by sending an SMS message to a short code, the first word of the message is the word “tip” plus a three digit code identifying the law enforcement agency to whom it is directed. This interface is actually a product of Anderson Software. When a tip is received, the TipSoft™ software generates an alias based on the cell phone number. Citizen Observer™, by way of TipSoft™, protects the identity of the tip submitter, passing only the message and the alias to law enforcement. The use of an alias allows law enforcement to contact the tip submitter to follow up if needed. This aliasing functionality can be described as providing a call back. Figure 2.7 illustrates how TipSoft™ processes data; in this figure the graphic labeled Crime Stoppers Desktop Computer is running the Citizen Observer™ application which utilizes TipSoft™.

It is interesting to note that in the Anderson Software demo of TipSoft™ they use colloquial SMS spelling patterns when replying to a demonstration tip. The message “Tip100 this is a demo” sent to 274637 received a response of “Tnx. Msg rcvd. Your alias is D3D3. Call 911 if emergency! Reply anytime. Delete your tip for safety. Text STOP to block replies” [19]. The abbreviated words “Tnx. Msg rcvd.” translate to “Thanks. Message received.”

TipSoft SMS

How it Works

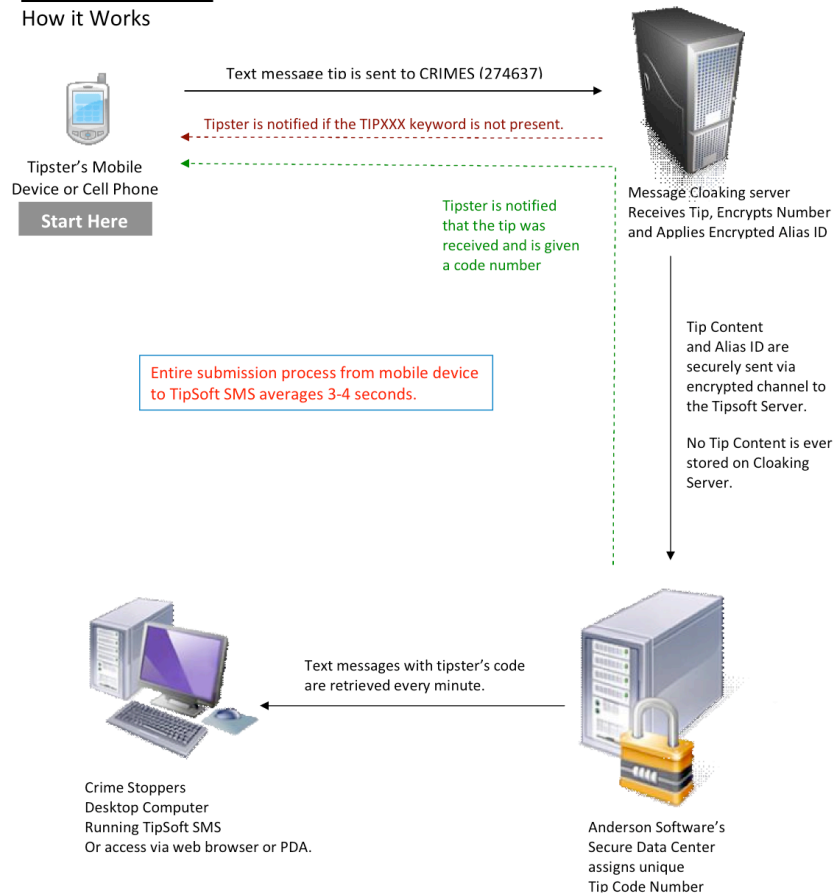


Figure 2.7 TipSoft data reception and processing. [20]

The TipSoft™ software has a very simple user interface allowing law enforcement users to interactively communicate with tip providers, retrieve new tips, delete tips, or search for tips via a variety of fields, by: Tipster [alias], Narrative [Tip Text], Case ID, Other ID, Priority, and Status [Open/Closed]. Figure 2.8 depicts the interface law enforcement uses to interact with TipSoft™.

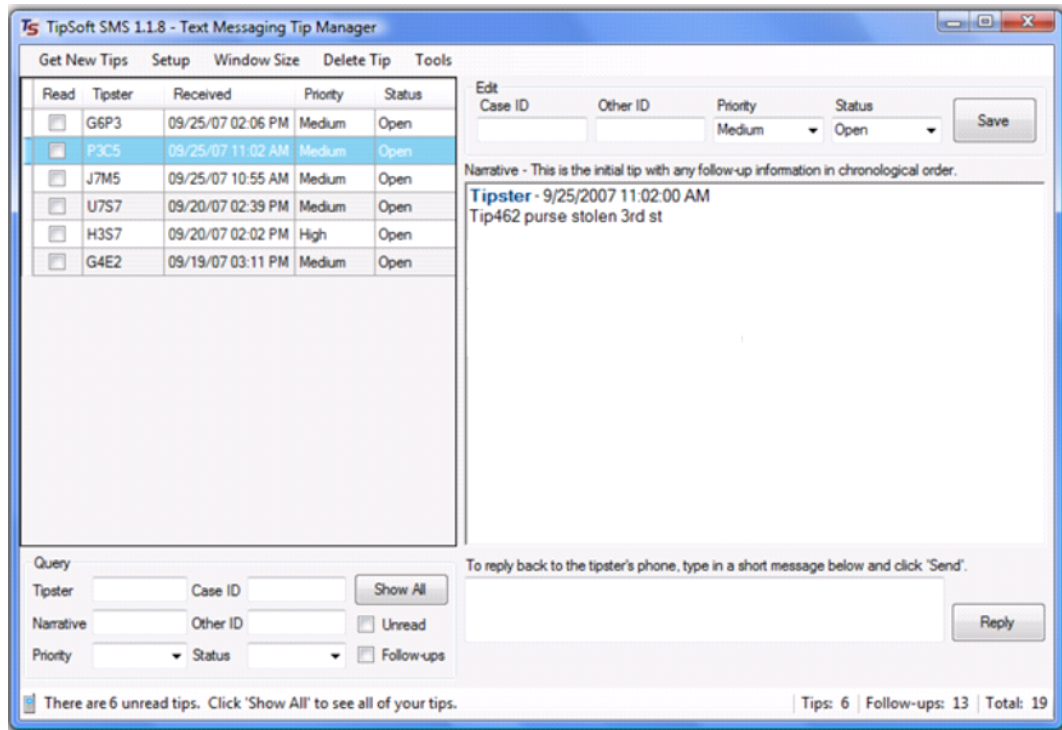


Figure 2.8 Citizen Observer's™ SMS tip processing interface. [4]

2.2.3 Crimeline

A product of New Primedia™ and the South African Police Service is Crimeline (<http://crimeline.co.za>). This website is a companion to the SMS short code 32211 for submitting anonymous tips to law enforcement. The website boasts that in the first weeks of operation more than 1600 SMS and online tips [16] were received. The South African Police Service reported on April 17, 2008 that 441 arrests and approximately two and a half million United States dollars in property had been seized or recovered since Crimeline went live June 7, 2007 [6]. The Crimeline website has five sections for users to fill out to report a tip: 1) identify the person who has/will commit a crime, 2) their address, 3) a physical address of where the offense has/will happened, 4) identify and address of the victim, and 5) general information about the crime.

The first section, “identify the person who has / will commit a crime”, asks for the name and alias of the individual, then it asks specifically for what crime, when it was or will be committed and a description of how it will be carried out. The second section, the address of the person being identified, asks for the physical address of the person being identified: Province,

City/Town, Suburb, Street name, and Street number. Figure 2.9 illustrates the first section of the form, Figure 2.10 illustrates the second section of this form.


Who do you think is breaking the law (give their names and / or alias, if you know it)?	
Name	<input type="text"/>
Surname	<input type="text"/>
Alias	<input type="text"/>
What offence was committed / is planned?	<input type="text"/>
When was the offence above committed / planned?	<input type="text"/> 
Describe how the offence was / will be committed	<div style="border: 1px solid #ccc; height: 40px; width: 100%;"></div>

Figure 2.9 Crimeline Identify a Person Section. [7]

Address of person breaking the law	
Province	<input type="text" value="-"/> 
City / Town	<input type="text"/>
Suburb	<input type="text"/>
Street name	<input type="text"/>
Street number	<input type="text"/>

Figure 2.10 Crimeline Address of a person section. [7]

The third section asks for the specific street address of where the crime has or will be committed. This section is illustrated in Figure 2.11.


Where did / will the offence take place? (if known)	
Province	<input type="text" value="-"/> 
City / Town	<input type="text"/>
Suburb	<input type="text"/>
Street name	<input type="text"/>
Street number	<input type="text"/>

Figure 2.11 Crimeline Address of the crime location section. [7]

The fourth section asks for information about the victim of the crime being reported. This section requests: name, alias, province, city/town, suburb, street name, and street number. This section is illustration in Figure 2.12

Victim Information	
Name	<input type="text"/>
Surname	<input type="text"/>
Alias	<input type="text"/>
Province	<input type="text"/>
City / Town	<input type="text"/>
Suburb	<input type="text"/>
Street name	<input type="text"/>
Street Number	<input type="text"/>

Figure 2.12 Crimeline Description of a victim section. [7]

The final section asks for additional information, such as if this tip is in addition to a previous tip, any information about pertinent police stations and any additional general information not otherwise provided. A feature on Crimeline’s tip submission form is a CAPTCHA, or Completely Automated Public Turing test to tell Computers and Humans Apart, this is a feature that attempts to test and validate, that a human, not an electronic process is actually submitting the information. It works by means of displaying random text as a picture that is ideally not parsable by a machine. Figure 2.13 illustrates the fifth section and Figure 2.14 illustrates an instance of the CAPTCHA mechanism.

General	
Is this additional information to a previous tip?	<input type="text"/>
Please supply police case number if known	<input type="text"/>
Please supply police station if known	<input type="text"/>
Please supply any additional information that can assist with the investigation?	
<input type="text"/>	

Figure 2.13 Crimeline addition information section of a crime tip. [7]

	
Please insert the same letters and numbers you see on the image above. This is an added security measure, and ensures we receive only valid tip-offs.	<input type="text"/>

Figure 2.14 Crimeline CAPTCHA function. [7]

Crimeline’s Terms of and Condition [of Service] describe a significantly different functional paradigm than Citizen Observer™. While both Crimeline and Citizen Observer™ protect the identity of the tipster, Crimeline’s explicitly states that the “User acknowledges that due to the anonymity of this system no personal follow-up to the user is possible” [17]. Also, in the Terms and Conditions Crimeline makes it clear they act simply as a pass through agent for data.

2.2.4 Amber Alert

One high profile program utilizing SMS is Amber Alert™ [24]. The goal of the Amber Alert™ is to quickly inform the residents in the geographic region in which a child has been abducted of the situation, as well as providing information about the child and any other relevant facts. Figure 2.15 provides a graphic description of this system. The intuitive reason for conveying Amber Alert™ information via SMS is the simple fact that the majority of cell phone users carry the device on their person most of the time, making it a highly accessible means of communication.

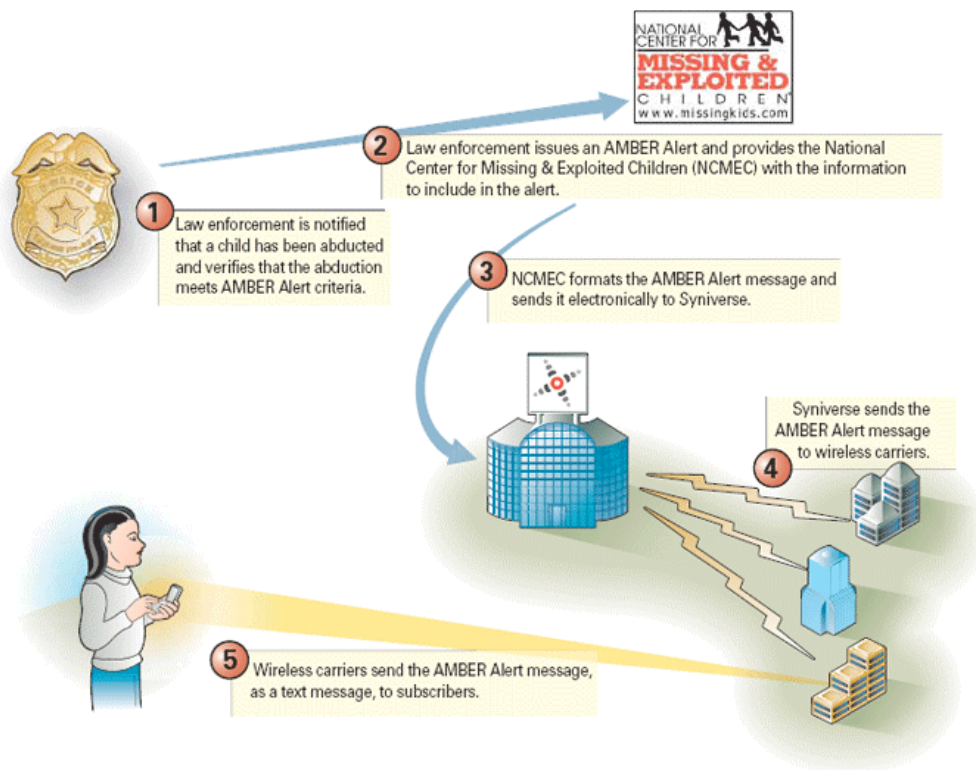


Figure 2.15 Amber Alert™ SMS Example Broadcast [9]

Amber Alert™ works using a very simple broadcast message paradigm. When an Amber Alert™ is issued, the database of registered SMS users is queried for all users' cellular numbers who have a registered zip code in the geographic area of the alert. This result, the set of Amber Alert™ users, is then provided to Syniverse [14] which functions as an SMS aggregator. Syniverse then relays the Amber Alert™ notice and user cellular numbers to the appropriate cellular carriers for final delivery to the user's cellular phones.

3.0 Methodology

Overview of Methodology

The methodology of a project describes the choices that were made during development. By providing a comprehensive analysis of these choices, the end product can be compared to other system and the assumptions used to make choices in development can be evaluated. The methodology for this project can be subdivided into three distinct components. The first two components are the Analysis (3.1) and the Technologies Used in Development (3.2) which layout significant choices made in development and how they affect the Final Implementation (3.3).

3.1 Analysis

3.1.1 Choice of Methodologies

Choosing a suitable software development framework can make or break a project. In choosing the framework several parameters need to be considered. The parameters that a framework needs to be evaluated against include: minimum development period, rate of gathering and completeness of requirements, and impact of mid-development change. For this project the primary development period was originally limited to November 2007 to mid February 2008. The ability to gather requirements was limited by the availability of the key stakeholders. The key stakeholders in the project can be partitioned into two distinct sets: law enforcement and community members who will use the system. The first stakeholders, law enforcement users, were included in the project from its inception. While they could provide specific information about what kinds of information they would like to capture in this system, "how best to obtain (the data)" and "how they would like to manipulate (the data)" were beyond

their experience. As a result, complete requirements were unavailable at inception from law enforcement users. The second set of users, community members, were considered but owing to the novelty of the system and the geographic area, it was decided that a small test group of SMS users in the age range of 18 to 25 would be solicited from the university community. This choice was made on the basis of perceived user inexperience in the larger community with the subject area. As a consequence requirements gathering with the second set of users would not be a fruitful pursuit until there was a tangible product to interact with. The cost for mid-development change for this project was not a monetary issue, but rather the cost of time relating to the development timeline and the significance of change to the system architecture. However, mid-development changes were anticipated due to the inability to gather complete requirement at the project's inception. Due to minimum scope guidance from the two stakeholder groups for which this application was being developed there was need for a fast development timeline and the ability to handle ill-defined and evolving requirements.

Two development frameworks, which have been widely used and researched, are the Waterfall Model and the Spiral Model. These two models were chosen because of the large volume of literature concerning them and their wide spread usage throughout the software development industry and academia. The Waterfall Model, one of the most commonly used methods [10], is a cascading model which moves from phase to phase with a strictly forward looking perspective. It progresses through five phases: Requirement [Gathering], Design, Implementation, Verification, and Maintenance (see Figure 3.1). This model is self-contained, and hence, each step can be completed in its entirety. This model is suitable where a full set of requirements can be gathered and the system can be fully vetted at each step. In this project, the key stakeholders cannot give a complete set of requirements, nor do they have complete availability to evaluate the system at each phase, therefore this model was deemed unsuitable for this project.

The Spiral Model is an alternative iterative prototyping model. It moves inward through four phases of development, where each iteration creates the foundation for the subsequent phase until the software is complete and ready for transition into production, which can be viewed as a final or fifth phase. The four phases are: Determine Objectives, Identify and Resolve Risks, Development and Test, Plan the Next Iteration (see Figure 3.2).

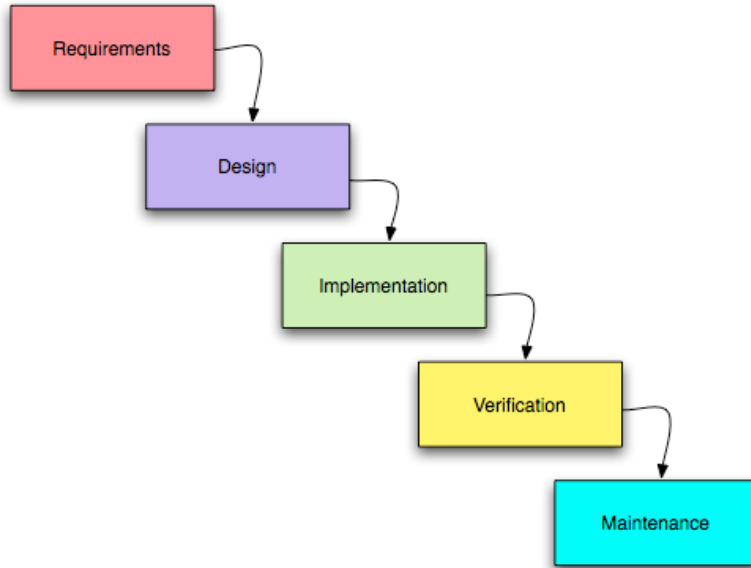


Figure 3.1 Waterfall Model of Software Development. [22]

The iterative development cycle of the Spiral Model is supportive of a rapid application development paradigm, which is suitable for projects which need to be developed on short time lines, and with less than adequate input from stakeholders. During an iteration cycle new features can be added and previewed by key stakeholders. This approach allows stakeholders,

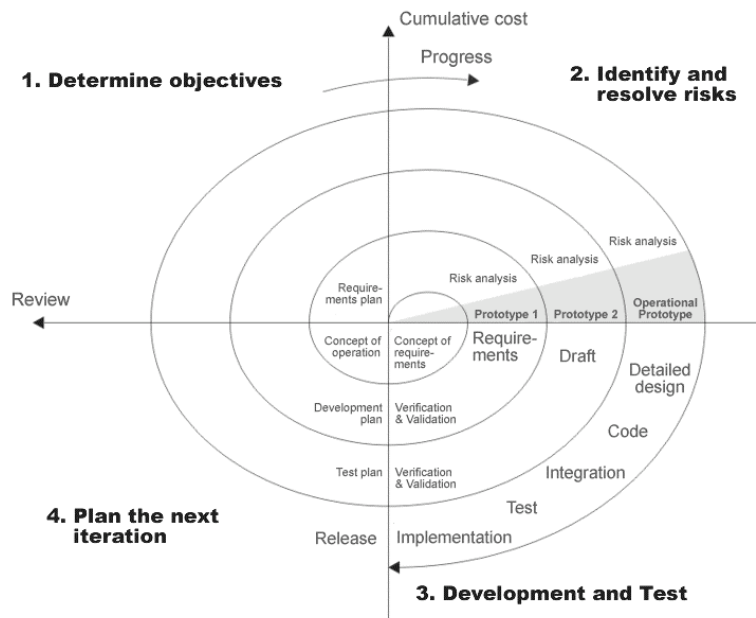


Figure 3.2 Spiral Model of Software Development. [15]

such as law enforcement, to provide additional requirements as development progresses and they can see a tangible product evolve. Another key benefit of this model is that the impact of mid-development changes are mitigated because changes are noted earlier in development and can be better integrated into the existing development. Given the flexibility of the spiral model, it was chosen over the waterfall model as better suited to the requirements of this project.

3.1.2 Desktop Application versus Web-based Application

In developing a modern software application, a key question is how the application will be delivered and interacted with by the end-user. Historically, applications have been developed using an individual install pattern that is colloquially referred to as a *desktop* application. Desktop applications are systems in which the business logic is primarily executed on the computer the user is directly interacting with; a web-based application alternatively uses a centralized computer, a web-server, to execute primary business logic while the user's computer simply acts as a terminal to interact with the server. To create a desktop application, developers must create a minimum set of requirements for the target system, both in terms of hardware and software capabilities, and test the application against a wide variety of configurations. This also means packaging all necessary libraries and ensuring that licensing for the libraries permits redistribution, and determining what financial obligations are required. A key benefit is that the application's functionality and appearance is completely controlled by the developer. The cost of this level of control is writing the application's user interface from the ground up, or integrating it within a user interface framework.

An alternative to desktop applications are web-based applications. Popular examples of web-based applications are Google Docs (providing word processing and spreadsheet functionality) and online ordering and reservations systems (such as Ticket Master or Delta Airlines). These three companies provide a selection of industry leaders, which have high availability revenue generating web-based applications. Web-based applications can provide an easy distribution method owing to the fact there is minimal business logic on the user's machine. Conceptually, a web-based application is very similar to thin client machines interfacing with server applications with the key difference being that the hardware used with a web-based application is not dedicated hardware exclusively running one application, in this case, a web browser. Utilizing a web-browser instead of dedicated hardware in either thin client or a desktop

application allows web-based applications to be portable, and not tied to a specific machine. As with all thin clients, maintenance and upgrades of software are done on the server with a web-based application, which results in a lower burden of supporting legacy code. The legacy code burden is handling different web-browser versions. Another advantage of web-based applications over desktop applications is the relative ease of testing over a wide variety of platforms as the platform is software, not hardware, dependent. One test system can have a wide variety of testing environments concurrently installed. For example, one test machine could have several versions of Microsoft Internet Explorer, Mozilla's Firefox and Opera all installed and available for testing.

For this project a web-based application approach was chosen. The distribution method met the needs of the key stakeholders, because for both law enforcement users and community members the threshold for accessing the application is simply navigating to a website. There is no complex installation procedure. User interface design is also simplified, which was highly desirable due to the constrained timeline the project was to be executed in. To develop a clean user interface, work was initially devoted to simply creating a functional system. The user interface for this application is an HTML cascading style sheet (CSS) that provides a consistent look and feel for all screens.

3.1.3 Components Off The Shelf versus Custom Development

To develop new software, one should first look at what has been done, to avoid re-inventing the wheel. Components off The Shelf (COTS) is the essence of re-using existing work. For some projects or components it is not possible to use commodity components. An example is the United State's Space Shuttle Fleet. Clearly those craft needed to be custom developed from the ground up. A common choice in software development is that of COTS, custom development, or a combination of the two.

Custom development allows for a great deal of freedom and flexibility because there is a blank canvas on which to work. The price of this freedom is increased time and effort in planning, development, and testing. Planning for custom development should be thorough as it will lay the foundation for the interfaces between different components. Incomplete planning will lead to poor implementation of interfaces between components; these interfaces are prone to duplication and errors when constructed in an ad-hoc manner. An auto-text conversion tool

included in this system required the custom development of a new component because there are no existing text conversion tools which could accomplish this specific conversion and provide the desired explanation of the conversion. A limitation of this tool, and custom developed software in general, is it is only as good as the degree to which it was tested. COTS software has been developed, released, actively maintained. Instead of a single developer or team working with the code, a much wider audience has tested and found bugs to be fixed.

Time spent with COTS is invested in learning the component and integrating it into the larger work. This is not necessarily a trivial task, but the advantage is the burden of initial development has been already paid in advance. COTS software can be very beneficial in a rapid development environment because more time is spent working to developing the end product, rather than producing an intermediate product. In addition COTS software, when built within a framework, operates in patterns. While there is an initial learning curve for the both the components and the framework, the learning cost for the framework is a one-time expense. If care is taken in choosing an appropriate framework, then even if a particular COTS component turns out to be a poor choice, the time lost is only in integration and learning the component, not in the development of the component.

A final benefit to working with COTS components is that within a given framework components can generally be easily integrated through DRAG and DROP mechanisms. This is not to say that components do not have pre-requisites or other requirements but these requirements are often the foundation of the chosen framework and are already present.

3.1.4 Web-based Application Development Framework

Web-base application frameworks are software systems, which expose a wide variety of service components and prescribe how applications are constructed and interact with the server environment. Internet based applications are developed in a wide variety of languages and frameworks, including PHP, Java Server Pages (JSP), Java Server Faces (JSF), Microsoft Active Server Pages (ASP) .NET, Ruby on Rails, and many others. The listed frameworks are all readily available for this project. Ruby on Rails was quickly eliminated because it has never been successfully setup and run onsite and it is also immature as a development framework. PHP and JSP were eliminated because the style of coding required does not cleanly separate

business logic from display, and they both rely heavily on in-line server processing code with HTML.

The two remaining choices, ASP.NET and JSF, represent popular web-based application development technologies which have wide industry acceptance. ASP requires Microsoft's Windows Server product while JSF requires a Java EE compliant server such as Sun's GlassFish, IBM's Websphere, Apache Tomcat, are three examples among many. The wide variety of server options with JSF allows for greater flexibility in that an application is not locked-in to one server environment. If better features are offered by another vendor, or something happens to the availability or maintenance of the server product, moving the application to another server environment would incur some transition cost but not a complete redevelopment as would be required with an ASP.NET application. In terms of overall application design scheme, JSF natively follows a Model-View-Controller (MVC) pattern. ASP.NET has a library for building applications following a MVC pattern, as an alternative to ASP.NET applications native design pattern using web-forms [12]. The choice of using and following an MVC pattern was made because of the clean separation of functionality it creates. An MVC pattern can also be very conducive to reuse of code [11]. JSF divides display code into files which are a mix of HTML tags and JSF tags which generates HTML, or the *view*, for the client's browser. The server bean maintains state information and is the business process *model*, and a file containing conditional navigation rules handles the flow or *control* of the application.

There are two other prime factors in choosing the application framework: cost and associated learning curve. Both choices are freely available in the current academic environment this application is being developed in, but the development tools and server to run an ASP.NET application are encumbered with an academic-educational license. Neither the development tools nor the servers for developing and running a JSF application are encumbered with any licenses requiring compensation for a for-profit venture. The other factor, learning curve, was a significant point considering the timeline of this project. Both frameworks do have an associated learning curve, but developing the Reminder Service in JSP provided an excellent primer for JSF development. In consideration of the cost, learning curve, and availability of server platforms, JSF was chosen for this application.

3.1.5 Choice of Database

In choosing a database, it is important to first determine the required features for the system under development. For this project five key aspects of a database product were identified:

- Support for stored procedures / user defined functions
- Foreign key support and enforcement
- Full text search capabilities
- Wide language support
- Low cost of ownership

Support for stored procedures and user-defined functions were identified for two key reasons. First, for clean abstraction of the data interface layer, this allows for the databases' structure to change without changing the source code of the application. Second, it helps to protect the database from structure query language (SQL) injection attacks. This form of attack is carried out by introducing maliciously composed SQL which will harm the database. SQL injection attacks can cause data corrupting or deletion.

Foreign key support and enforcement was desired to protect the referential integrity of the database. The term referential integrity means to maintain the database in such a way as to prevent data from being lost or unassociated. Database system rely on elements called keys which are how data is accessed and reference, protecting referential integrity ensures the data which is tied to these keys is not delete or disassociated so that data becomes inaccessible.

Full text searching allows the database to index the individual words in a block of text. This enables the database to accept a query, which is a set of words, and find, in a non-brute force manner, texts which contain the set of words and then rank them according to percentage of a match that exists between the query and the found texts. An example would be looking for tips which include: woman, red, truck, not (beach). Using a full text search the database would find text which contains the words woman, red, truck but do not contain the work beach.

Wide language support is in relation to programming languages not human communications. While this project is executed currently in one language, the database should

not be strictly limited to only supporting the original development language. This requirement is to keep a forward-looking perspective with the anticipation this project will be expanded upon and commercialized.

Cost of ownership is a serious consideration for this project looking forward. While the direct monetary development cost are not a significant issue, cost of continuing operations and continued development in a commercial arena are an issue. The continued operation of the database and the system as a whole should not rely upon its original academic origins to be legally and financially solvent.

Practically there are three primary database choices that meet most of the requirements detailed and are readily available: IBM DB2, Microsoft SQLServer, MySQL. All three databases have Java drivers which allows for native communication between the development framework and the database. Both DB2 and SQLServer are available through academic license arrangements. These academic licenses provide for non-commercial research and teaching usage. DB2 and SQLServer have proven track records in industry and represent mature established technology. The downside to both of these options is that neither meets the cost of ownership requirement.

MySQL can meet the four requirements SQLServer and DB2 can provide: stored procedures / user defined functions, foreign key protection, full text search, and wide language support. Additionally MySQL is an open source product of Sun Microsystems and is free to use. There are subscription enterprise support options all well when organizations using MySQL mature to a point where it becomes needed. MySQL was also the database which supported the Reminder Service. Given prior experience with the database and meeting the five requirements it is the database which was chosen for this project.

3.2 Technologies Used in Development

3.2.1 Selection of Environment

Several vendors provide Java Enterprise Edition (JEE) servers, which are software environments, that support JSF: Apache, IBM, and Oracle are just a few. IBM and Oracle both offer rich server products but both are limited for this project to the academic arena due to licensing. Apache's Tomcat product is an open source alternative which has a very flexible license which will allow this project to move into the commercial arena. It should be noted the

Apache Foundation is actively supported and contributed to by both IBM and Oracle. In 2006 Oracle donated its Application Development Framework Faces libraries, which are based on the JSF specification (JSR 127), to the Apache to add to Apache's JSF libraries. Given the industry support and license which Tomcat falls under, Tomcat was the clear choice of JEE server for this project.

In selecting Tomcat for a JEE server it makes sense to choose JSF libraries which were developed for that platform. Apache's MyFaces libraries are just that, JSF specification compliant libraries developed for Tomcat. These libraries could be utilized in any JEE server to provide JSF capabilities but these libraries were developed and tuned utilizing Tomcat. The other primary alternative for use is Java's Reference Implementation JSF libraries. While Java's Reference Implementation was built according to the JSR 127 specification it does not have a very wide palate of built-in components. Nor is this implementation widely supported or utilized in the JSF development community. MyFaces on the other hand is widely used and is being actively developed by its user/developer community. This active development implies the product is actively having bugs fixed and new enhancements with each new release.

In addition to the core MyFaces libraries required to enable for JSF capabilities in Tomcat the Apache Tomahawk and Sandbox libraries are being utilized to gain features not available in the core. A third non-core library, jenia4faces, used was developed by jenia.org it offers a suite of data tools, charting and dynamic HTML components. The component that was not otherwise available is a multi-selection checkbox which integrates into any JSF data table component. This functionality was needed to allow multiple non-contiguous rows in a data table to be individually selected. In other words using a data table with ten rows, a user would like to select a subset of rows, perhaps one, three, and eight, there is no direct built in support for activity in any of the Apache JSF libraries.

3.2.2 Cascading Style Sheets

Cascading Styles Sheets (CSS) are a technique for applying layers of HTML layout and design components on top of a base HTML document, a good analogy would be a mannequin in a store. A store associate could dress the mannequin in any combination of clothes to make many distinct looks all using the same base mannequin. This technique has many advantages over old techniques of having the base HTML document being the final product an end-user

sees. Using the mannequin analogy, a mannequin could only have one outfit to change its display would require changing the mannequin not simply its dressings. Using CSS, the final HTML document being displayed can easily be tune for the end-user's browser. What looks good and polished in Microsoft's Internet Explorer might not display correctly in Mozilla Firefox. Using CSS, a custom CSS for each web-browser can be generated to keep a consistent look and feel across all browsers. Since creating a good looking CSS can be a time consuming task requiring an artistic eye, and the actual color scheme and layout is not mission critical a CSS package from Free CSS Templates was utilized. This website has a completely open licensing arrangement for using and modifying the CSS packages they provide.

3.2.3 jQuery

Javascript is another key display technology being utilized in this project. It enables hide/reveal functionality and Ajax (Asynchronous Javascript and XML) for lightweight data transfer between the browser and server. jQuery was chosen again for the wide feature set and wide acceptance in the web application development community. The wide level of acceptance of this as well as the other technologies using in the project has typically had strong co-relation to the availability and quality of tutorials, explanations and documented bugs and solutions.

3.2.4 Axis 2

To interact with the messaging aggregator necessitated the development of web services to receive both SMS and MMS messages. Axis 2 was selected for several reasons first and for most it is another Apache product that is pure Java. Both setting up Axis 2 and deploying new web services in the Axis 2 framework is little more than drag and drop. This ease of setup and deployment was ideal for this project's timeline and fits neatly in the COTS approach.

3.3 Implementation

3.3.1 Database Design

Database design was a critical issue in developing this system. Several import considerations had to be made, an appropriate structure to capture the data, ensuring referential integrity could be maintained, minimizing duplication of data, correct abstraction of functionality. The data that is at the heart of this system is the *tip*, textual and/or visual data

provided by community members. The event table is the root of the table structure, along with the tip the other vital pieces of information which are maintained within this table are the date / time the system received the tip, and if provided by the user, what date / time the events described in the tip happened. These are the data elements all new tips to the system will have. From there the data model branches off into several tertiary tables: communitytipcontact, eventlocation, multimediamata, persondescription, and vehicle. Table 3.1 provides a brief description of what each table's data models. While this list of tables is not exhaustive for the database it provides comprehensive coverage of the data a community member could optionally provide.

The entity relation diagram, Figure 3.3, illustrates the relationships that existing between the enumerated tables and the supporting tables. (Refer to Appendix A for a complete entity relationship diagram.) Note that all tables connected to the event table have foreign key constraint on event such that, a data row must be present in the event table before it can exist in any other table. This foreign key relationship forms the root of the data model. Two other foreign key relationships are shown in both cases they are to reduce data duplication and wasted spaces. One of the two other foreign, labeled METForeign3 abstracts textual meta tag labels away from a specific event and instead links them via a bridge table by the primary key of the metatag table and the event table.

Table 3.1 Description of key user generated data database tables.

Table Name	Description
events	The actual provided tip and date/time stamp
communitytipcontact	User contact information
eventlocation	Address and description of the event described in the tip
multimediamata	Video or image included with the tip
persondescription	Biographical information of a person involved in the tip
vehicle	Physical description of a vehicle involved in the tip

The second foreign key is for the multimediatatable that holds any binary data submitted with a tip such as a video or a still picture. The foreign key labeled MDForeign4 prevents data duplication of repetitive file types being maintained in the multimediatatable. A total of eighteen stored procedures were created to handle input to the database. These were created to

ensure foreign key constraints between the related tables were appropriately handled and to encapsulate functionality so as to avoid repetitious code to simplify maintenance and changes to the system. A direct example of encapsulated functionality is submitting an anonymous tip verse submitting a tip with contact information for the tip provider. Two different stored procedures are initially called, publicAnonymousInsert to insert the anonymous tip, and publicCommunityInsert to insert the tip with contact information for the community member submitting the tip. In fact, the publicCommunityInsert initially calls publicAnonymousInsert to get a foreign key to the event table so that the contact information in the Community Tip Contact table can be related back to the Events table, which maintains the actual tip. Another benefit of encapsulating functionality in this manner was when full-text search features came online but the tables holding the data did not actually support full-text search doing parallel inserts in to a second table of MyISAM which supports was trivial.

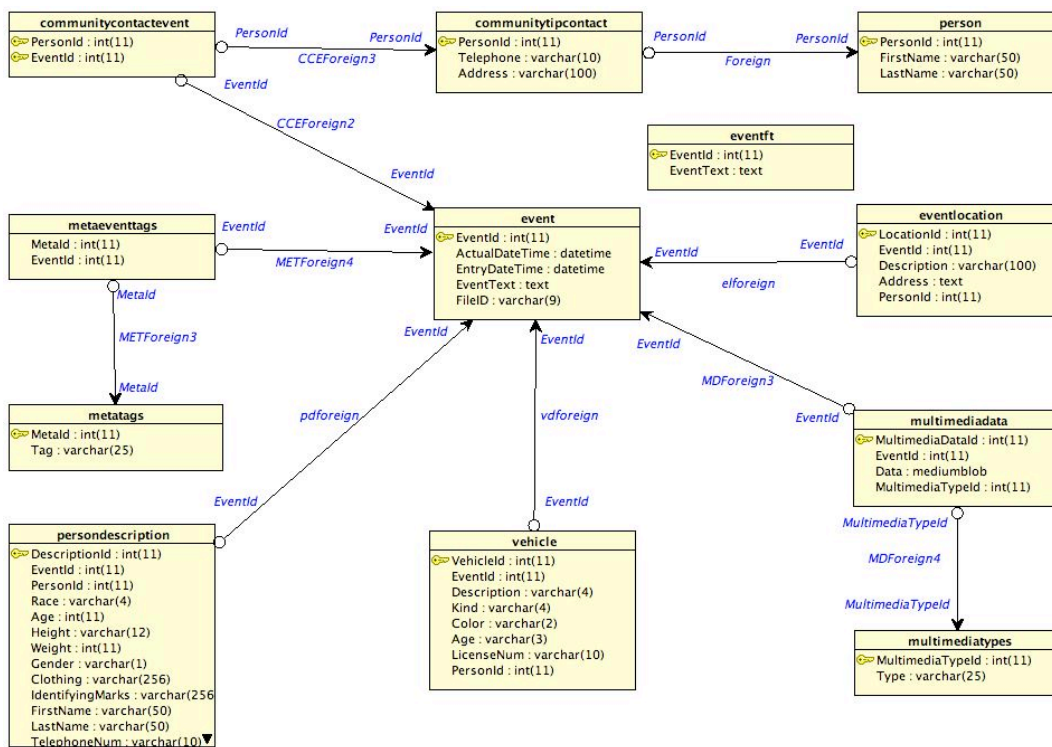


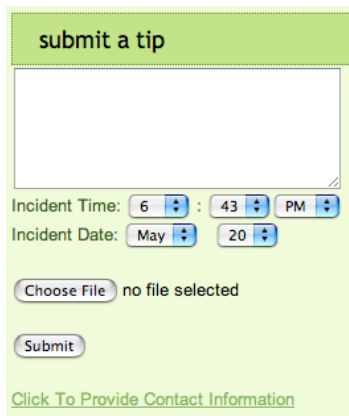
Figure 3.3 Entity relationship diagram of core tip related tables

All tables maintaining foreign key constraints must be of type INNODB in MySQL, which does not support full-text searching. The only data entry points in the system for inserting data into most tables is a stored procedure. For each table which has a stored procedure with an insert

statement, there is only exactly one stored procedure which will insert into that table. For the Events table there is only the publicAnonymousInsert, all other procedures call this procedure to insert into the Events table. When full-text searching over the events table became necessary the only alteration required was to first insert it the Events INNODB format table, get the primary key for that record and do a second insert with that key and the text to be searched into a second parallel table, eventsft (Events Full Text) of type MyISAM. In Figure 3.3 note that there is no explicit foreign key relationship shown between event and eventft because MySQL will not support and enforce a foreign key relationship between a INNODB table and a MyISAM.

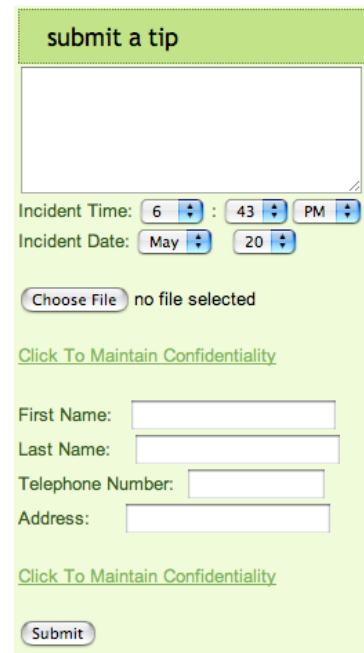
3.3.2 Web Application Interface

To submit a tip through the web, initially the user is simply prompted for a general tip along with drop down boxes to describe the time and date of the incident being reported. Additionally a file, either image or video, can be uploaded, and the user can provide contact information if they choose not to be anonymous. Figure 3.4 depicts the initial form a user is presented with to submit a tip on the web. Figure 3.5 depicts the form after a user has decided to no longer be anonymous.



The screenshot shows a web form titled "submit a tip" with a light green header. Below the header is a large text area for the tip. Underneath the text area are two rows of time and date pickers. The first row is labeled "Incident Time:" and shows "6" for hours, "43" for minutes, and "PM" for the period. The second row is labeled "Incident Date:" and shows "May" for the month and "20" for the day. Below these pickers is a "Choose File" button followed by the text "no file selected". At the bottom of the form is a "Submit" button and a link that says "Click To Provide Contact Information".

Figure 3.4 Initial Tip Form



This screenshot shows the same "submit a tip" form as Figure 3.4, but with additional fields for contact information. Below the "Choose File" button and "Submit" button, there is a link that says "Click To Maintain Confidentiality". Below this link are four input fields: "First Name:", "Last Name:", "Telephone Number:", and "Address:". At the bottom of the form is a "Submit" button.

Figure 3.5 Tip Form with Contact Information

After a tip is submitted using the form in Figure 3.5 the following screen (Figure 3.6) verifies the information initially given and elicits additional information about individuals, location of the incident and vehicles involved. The data being verified is highlighted in red text.

Tip Verification

Please confirm this data is correct before it is submitted to law enforcement.

Your Tip:

Demonstration Tip

Time of Event: 6 : 43 am

Date of Event: May 9

Describe the person involved

Yes No

Describe the place the event involved

Yes No

Describe the vehicle if one was involved

Yes No

Figure 3.6 CommunityTips.org Tip Verification

If the user has additional information and selects “yes” to “Describe the person involved”, “Describe the place the event involved”, or “Describe the vehicle if one was involved” (as seen in Figure 3.6) more questions are presented to the user asking for specific information. Good user interface design will allow users to access relevant content while filtering out extraneous material [11]. If a user selects “yes” to “Describe the person involved”, the user will then be presented with the additional questions regarding the nature of the person being described (Unclear, Suspect, Witness, Victim) as well as standard biographical information name, clothing, gender, race, height, weight, and contact information. Figure 3.7 shows the

additional question a user is presented with after answering “yes” to “Describe the person involved”.

Describe the person involved

Yes No

Was this person: Unclear Suspect Witness Victim

First Name:

Last Name:

Clothing:

Identifying Marks (tattoos, scars):

Weapon:

Sex:

Race:

Age:

Height:

Weight:

Contact Information:

Telephone Number:

Address:

Figure 3.7 CommunityTips.org Describe a person involved form

If a user selects “yes” to “Describe the place the event involved” the user will then be presented with the additional questions for a description and address of the place the event involved. Figure 3.8 shows the new question the user is prompted to answer. If a user selects “yes” to “Describe the vehicle if one was involved” the user will then be presented with the additional questions for the type, manufacturer, color, age and license plate. Figure 3.9 shows the new question the user is prompted to answer.

Describe the place the event involved

Yes No

Describe the Location (landmarks, terrain, etc):

Address of location:

Figure 3.8 CommunityTips.org Describe the place the event involved form

Describe the vehicle if one was involved

Yes No

Vehicle Type:

Vehicle Manufacturer:

Vehicle Color:

Vehicle Age:

Vehicle License:

Figure 3.9 CommunityTips.org Describe the vehicle if one was involved form

The web site has additional functionality for registered law enforcement users to search and review tips, provide revisions and comments about tips, and provide linking within the database of related tips. Presently meta-data about images must be manually supplied with key words describing a video or images, such as car, people, down town, or a specific location. These keywords or meta-data are used to provide text-based searches of non-textual data. Figure 3.10 shows the basic search functionality. A law enforcement user can select through check boxes which sources of data to search over: the original event text (Event check box), comments other law enforcement users have provided (Comment check box), revisions of the original tip text by created by other law enforcement users or by electronic means (Event Revision), or meta-data, annotations, to the pictures/videos (Pictures / Videos).

Tip Search

Event Comment Event Revision Pictures / Videos

Search by date

Yes No

Search on the person involved

Yes No

Search on the location involved.

Yes No

Search on a vehicle involved.

Yes No

Figure 3.10 CommunityTips.org Basic Tip Search Interface

Searches can also be parameterized by limited to a date period (Figure 3.11). Any time data elements (tips, revisions, comments, meta-tags, links between tips) are added to the system a time stamp is made. This time stamp is how data is search over temporally. Searching by date can be limited to before, after, or around (a five day plus or minus window) a given date.

Tip Search

Event Comment Event Revision Pictures / Videos

Search by date

Yes No

Date:

Around Before After

Figure 3.11 CommunityTips.org Tip Search With Date Limit

Search on the person involved

Yes No

Was this person a:
 Unclear
 Suspect
 Witness
 Victim

First Name:

Last Name:

Clothing:

Identifying Marks (tattoos, scars):

Weapon:

Sex:

Race:

Age:

Height:

Weight:

Contact Information:

Telephone Number:

Address:

Figure 3.12 CommunityTips.org Search Using Personal Details

Searches can also be done using the same fields as a community member’s tip provided through the web: using details about a person (Figure 3.12), using details about a location (Figure 3.13), and using details about a vehicle involved (Figure 3.14).

Search on the location involved.

Yes No

Describe the Location (landmarks, terrain, etc):

Address of location:

Figure 3.13 CommunityTips.org Search Using Location Details

Search on a vehicle involved.

Yes No

Vehicle Type:

Vehicle Manufacturer:

Vehicle Color:

Vehicle Age:

Vehicle License:

Figure 3.14 CommunityTips.org Search using Vehicle Details

Searching tips in the system can be accomplished through a basic search for text, or any combination of date parameterization or using characteristics provided by community members such as details about a person, a location or a vehicle involved in the incident.

When a user searches the system the original search page is reloaded, at the bottom the search results are appended. The search results include, the database id of the original tip, any time information the user supplied, the time the tip was actually processed by the system, the original text of the tip, any associated image or video. An illustration of the search results can be seen in Figure 3.15 where no text was provide in the search box and the search was only limited to around the date of May 22, 2008.

Search Results

Link Events

Event Id	Actual Date/Time	Entry Date/Time	Original Text	Image	Link Events
118	5/22/2008	5/22/2008	Underage drinking on campus! Event Details		<input type="checkbox"/>
119	5/22/2008	5/22/2008	drug deal just happened at the gas station on hunters trail and college road Event Details		<input type="checkbox"/>
120	5/22/2008	5/22/2008	Clara was stabbed at Johnny Mercer Pier Event Details		<input type="checkbox"/>

Figure 3.15 CommunityTips.org Tip Search Results

When two or more tips are found and a law enforcement user believes they are related to one another the user can click the check box next to the tip in the search results and press the “Link Events” button to provide a direct relationship between the two tips in the database. This functionality allows a user to view one tip and see not only all details about the tip but also the other tips that are believe to refer to the same incident or are otherwise associated with it.

In the search results by clicking the Event Details button for any found tip, complete details about the tip are provided including other tips which have been linked to it, refer to Figure 3.16. In the same figure you will also see a button, Unlink which can be used to separate out tips which are no longer believed to be relate to the current tip being viewed. Both which tips are linked as well as unlinked records are maintained in the database as to which user created the link as well as which user removed the link.

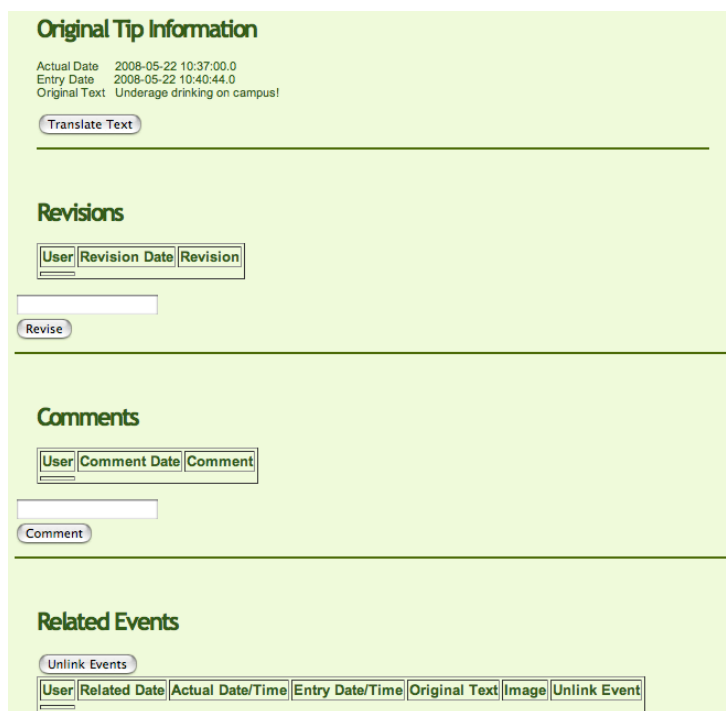


Figure 3.16 CommunityTips.org Event Details for tip 118

When a particular tip is selected from the search results meta-data can be added to provide textual searching capabilities for graphics. This meta-data allows the databases’ native text searching functions to search over binary object data such as jpg images or mpg movies. (Figure 3.17).

Enter a comment separated list of details about the image or video:

Figure 3.17 CommunityTips.org function to add meta-data about an image or video

When dealing with a tip, law enforcement users may want to either provide a revision to the tip through an electronic process or from contextual understanding of the tip. Chatspeak, txtspk, and text language are all synonyms for the informal short hand language that is often used in text messaging exchange as well as on the internet. Some simple example are shorting the work *you* to simply the letter *u* or turning *later* into *l8r*. Additionally whole phrases are sometimes replaced in this short hand, *got to go* is replaced with *g2g* or *be right back* with *brb*. One feature for law enforcement to handle text language is an auto-text conversion tool. Community Tips has a simple tool, which will automatically search and replace 225 common phrases often used in text language. These phases were drawn from two Wikipedia entries: Internet Slang and SMS Language as well as anecdotal experience. Figure 3.18 depicts the output, note that it

Revisions		
User	Revision Date	Revision
AutoTextConvert	5/9/2008	<p>direct translation: tonight there was a fight between a guy & a marine at roxx over his girlfriend</p> <p>conversion mapping: 2nite tonight there not in conversion table was not in conversion table a not in conversion table fight not in conversion table btwn between a not in conversion table guy not in conversion table & not in conversion table a not in conversion table marine not in conversion table @ at roxx not in conversion table ovr over his not in conversion table gf girlfriend</p> <p>This revision was created using a conversion table using data from http://en.wikipedia.org/wiki/SMS_language and http://en.wiktionary.org/wiki/Appendix:Internet_slang and real world exprience.</p> <p>Data compiled 5-7-2008 Unique conversions 225</p>

Figure 3.18 CommunityTips.org Automated Textual Translation of tip 105

provides both the direct mapping of the text language to normalized English as well as an attempt to provide a direct transliteration of the text language message to normal English. This is a revision of tip provided through electronic means. If the language is more nuanced than the auto-text conversion tool can handle or there is other contextual information a law enforcement user can also provide another revision of the tips through the provided text box (Figure 3.16). All revisions are tracked as to which user provided them and when. For the auto-text conversion tool the database has a user with no login access privileges who is recorded as providing the revision.

Also there is functionality for law enforcement users to provide commentary about tips and keep track of the comment thread for each tip (Figure 3.16). This is to enable an electronic collaborative environment allowing multiple users to share and exchange ideas about a tip.

When any data is entered into the system, a tip, revision, comment, linking or unlinking the data is maintained in the system with no delete functionality. This was a purposeful choice to protect the chain of custody of this electronic data, necessitated so that there is no direct way to purge data from the provided user interface. Maintaining the integrity of the system running the database is an issue beyond the scope of this project.

Additional functionality for the registered law enforcement users includes changing a user password, for Administrators there is also functionality to reset user's passwords, activate and deactivate login privileges as well as to assign users to be either regular users or administrators. The difference in the system for users and administrators is the availability of user administration tools.

3.3.3 Messaging Interface

Recall that there are two classes of messaging which Community Tips aims to handle, SMS and MMS. These two classes require two distinct interfaces to be handled appropriately. With an SMS message there are simple string data objects whereas with an MMS message there will be an additional binary data object.

The SMS Interface was built according to the web service interface specified by the content provider, Mobile Education LLC. Their modular architecture parses inbound messages

for a leading keyword which can be viewed as a defining a class of messages. For Community Tips the keyword is tip (case in-sensitive). When this keyword is found a database lookup executes and locates the web service url which is responsible for handling this particular class of messages. The web service takes as formal parameters: phone (the sender's phone number), sentTo (what service the message was routed to, Community Tips in this case), body (the actual message), carrierId (sender's cellular provide's id), registered (a boolean indicator if the sender is registered with Mobile Education), and sessionId (a Mobile Education parameter). Although the cellular number from which this message originated from is transmitted to Community Tips the number is never stored beyond the formal parameter variable and will be lost from active memory as soon as the variable is lost from visible scope. Out of all the formal parameters supplied by Mobile Education only the body is saved by Community Tips. When the web service receives a message it first cleans it via a database sanitizer class to protect the database from malicious input and then calls the publicAnonymousInsert stored procedure to handle actual database the insert. This is the same stored procedure as would be called from the web application in the case of a user submitting a tip without any multimedia data (a picture or video) and choosing not to disclose any contact information.

The MMS interfaces mirrors the SMS interfaces with the afore-mentioned exception that it handles an additional parameter of a binary data object or more particularly a byte array. This necessitates calling a different stored procedure, publicAnonymousMediaInsert which is the same as would be called from the web application in the case of a user submitting a tip with a picture (or video) but choosing not to disclose any contact information. Mobile Education does not currently support MMS messaging so this feature while built and tested is not yet available.

4.0 Discussion

4.1 Comparison To Existing Systems

Three community policing systems built around cellular messaging capabilities have been presented. One, Crimeline, operates exclusively in a pull style; another, Amber Alert™ operates exclusively in a push style. The remaining system, Citizen Observer™ operates in with dual modality, pull and push. The following sections will discuss, compare, and contrast these systems and how they relate to the design and implementation of Community Tips.

4.1.1 Citizen Observer

Citizen Observer™ and Community Tips are very similar in terms of the information that is asked of users when submitting a tip. Both systems gather biographical information about persons involved in the tip, the location the tip happened at, vehicles involved, and both accept digital photographs uploads. Community Tips additionally supports video upload and allows users to identify the class (Unclear, Suspect, Victim, Witness) of the person whose biographical information is being provided. Citizen Observer™ solicits this information as one long contiguous form which may detract from usability of the system by providing content which is a distraction [11]. For example a user may not have any biographical information on the persons involved in a tip but may be able to identify a vehicle. Some users may be confused about using the system since they cannot answer all the questions being posed. Community Tips instead structures the questions in a hierarchical nature to allow for a natural progression of questions which a user can clearly choose to answer or not [11].

The interface law enforcement uses to review tips in Citizen Observer™ has a wide variety of functionality: searching for tips, retrieving new tips, deleting tips, interactive dialog with tip providers. While Citizen Observer's™ tip searching facilities are limited to searching over Tipster [alias], Narrative [tip text], Case ID, Other ID, Priority, and Status, Community Tips can search over the tip, comments, revisions, meta-data if multimedia data is included, and date parameterization. It is worth noting that the user interface for Citizen Observer™ will allow user's to delete a tip. In developing Community Tips a distinct choice not to allow any user interface functionality to delete tips, revision, or comments to ensure data preservation is maintained for all information. A key difference in Citizen Observer's™ interface for handling and Community Tips' is the support facilities for managing tips. Community Tips allows law enforcement to generate both automatic and human produced revisions to user supplied tips, also law enforcement can provide comments allowing for a collaborative work space. Citizen Observer™ has none of these facilities.

One very distinct feature which differentiates Citizen Observer™ and Community Tips in the messaging services which are offered is Citizen Observer's™ call back function. A concept from theater that is very apt in the context of anonymous tipping, "Willing Suspension of Disbelief". In theater actors do not acknowledge the audience's presence directly so that the

audience can maintain the perspective of being a fly on the wall simply observing fictitious events as though they were real. The same holds true in an expected environment of anonymity. Law enforcement are the actors and the anonymous tippers function as the audience. Community Tips provides a general acknowledgement message in reply to a submitted tip, noting specifically that the message was received and there is no identifying information stored with it. This would be akin to an actor pausing for applause. Citizen Observer™ replies that the message has been received and provides a user alias. This is not necessarily intuitive to the end user that their identity is being protected. Particularly, it may confuse end users about their anonymity if there is a call back at some indeterminate point in the future, with specific questions regarding the anonymous tip. To complete the theater analogy, this kind of reply would be akin to an actor directly addressing the audience. Providing an alias and potentially following up with specific questions related to the tip has a serious risk of harming the efficacy of an anonymous tipping system by causing the end users to lose faith in their anonymity.

Additionally Citizen Observer's™ call back could be detrimental to the user if the message originates from a phone which is shared. Both Citizen Observer™ and Community Tips, as has been stated, provide an acknowledgement of receipt that is transmitted immediately. There is no assurance that Citizen Observer's™ call back will be close in time to the original message, a long time differential could result in the user no longer be in possession of the phone, putting the user in some degree of risk.

A technical note regarding the anonymity provided through Citizen Observer's™ alias. There is no explanation of how the alias is generated, simply that the alias is the mapping between the tip provider's number and law enforcement. This alias could be a randomly generated id or it could be generated through a two-way hash function. If it is randomly generated, then anonymity from law enforcement is more likely, if it is a two-way hash, anonymity is less secure.

It is worth noting in Citizen Observer™ Privacy Policy regarding personally identifying information “when we believe in good faith that the law requires it”. Not under court order, simply the belief in a good faith request. Also Terms [of Service] has a curious language:

Submissions of Content by Community Group

As a participating Community Group, you will be offered Services that accommodate or require user-supplied information, materials or communications (collectively, "User Content"). You grant CitizenObserver.com a non-exclusive, worldwide, royalty-free license, in any media now existing or existing in the future, to link to, use, copy, distribute, modify, publicly display, transmit and perform publicly the submitted User Content; provided that we will use the User Content in accordance with our posted Privacy Policy. You retain all other rights in and ownership of User Content submitted by you, subject to the license granted to us in this paragraph.

The concern about this paragraph is the language: “use, ... distribution, ... public display” of “User-Content”, there are no specific limitations or exclusion regarding tips being transmitted to law enforcement. The issue is that this content could implicate the otherwise anonymous user. For example displaying a picture provided with an anonymous tip of interior room of a private residence could affect the perceived anonymity of the user.

4.1.2 Crimeline

Crimeline™ is very similar to Community Tips in end user functionality; it has a simple web interface for users to submit anonymous tips and a SMS interface for cell phone based tips. Like Citizen Observer, Crimeline™, solicits tip information on the web in a long contiguous form. Unlike either Citizen Observer™ or Community Tips, Crimeline™ does not have the ability for a user to upload any digital media nor does it ask for any information regarding automobiles. These could be cultural differences between the American audience which Citizen Observer™ and Community Tips are implicitly geared to, having been developed in the United States, and the South African audience which Crimeline™ is geared towards. Crimeline™ does have a section requesting information about existing police records that the user may be aware of.

One interesting information question that is common between Crimeline™, Community Tips and Citizen Observer™ is asking for the location of where the events described in the tip happened. Crimeline™ requests very specific information, down to the street number of the location. Citizen Observer™ asks the same question in very open general term in a large text area, “Crime Description: (Including... Who, What, When, Where and How Do You Know)”. Community Tips has taken a middle road approach to issue of location, asking for a street

address as well as a spatial description of the location. This approach was chosen because the physical address may not be known, but its location in relation to known landmarks could be just as valuable. The relative value of the approach will have to be evaluated at a later date when Community Tips is seeing regular use by community members.

Both Citizen Observer™ and Community Tips both have back-end applications which receive and manage tip data for law enforcement; Crimeline™ does not offer these facilities. Instead Crimeline™ acts as a transmission conduit for user supplied data, though the actual means of delivery to law enforcement is unclear.

Crimeline™ is identical to the default settings of Community Tips in its approach to anonymity, it maintains no user related data. Crimeline's™ Terms and Conditions [of Services] state:

The User shall provide only the tip-off on the website and not own personal information. The User understands and agrees that New Primedia (Pty) Ltd shall forward such tip-offs to the SAPS who in turn, will use the tip-offs accordingly.

New Primedia (Pty) Ltd will ensure that the user's contact information will not be forwarded to the SAPS with the tip-off, except in those instances where the User provides their own personal contact information in the tip-off.

The User acknowledges that due to the anonymity of this system no personal follow-up to the user is possible.

Crimeline's™ approach to anonymity is in line with discussions held with law enforcement during the development of Community Tips. This style of relations with end users is thought to be the most conducive to regular and active use by the community.

4.1.3 Amber Alert™

There are two limitations of Amber Alert™ via SMS, geographic range as well as under utilization of end user devices. Amber Alert™ requires SMS users to provide up to five postal zip codes which they are typically in, while network providers can locate a cellular phone within their network they do not at present share this information with Amber Alert. Thus Amber Alert™ will send notifications to users which potentially are not in the geographically relevant area potentially causing confusion due to erroneous reports. For example any alerts for Wilmington, NC will be sent to a subscriber regardless of the supplied zip code, 28403, 28405,

or 28412. But if the user was out of the area, perhaps in New York City, NY and registered in New York City, the user will still receive alerts pertaining to Wilmington.

Community Tips at present also is limited by geographic range. All tips are treated as being from the same geographic origin. There are three strategies which can be used to differentiate geographic origin of tips: unique geographic specific short code, unique top-level keyword, or a secondary keyword identifying geographic origin. A unique geographic specific short code was infeasible for this project and for many areas which the system might be used due to the monthly costs associated with maintaining a short code and the finite number of short codes available. Having a unique top-level keyword is a workable solution, but it is limited by the ability and willingness of Mobile Education to support multiple keywords for this application. The third option of a secondary keyword places additional burden on end users to know their geographic specific secondary keyword. Either of the first two options would be preferable from a usability point of view because they place no additional burden on the end user than the current system. The second option, of a unique top-level keyword is how Citizen Observer™ manages this issue. In developing Community Tips the issue of geographic range was recognized and appropriate modifications to the database to support geographic identification of tip origin have been planned but not implemented due to the narrow project schedule.

The second limitation related to the Amber Alert™ system is that it is exclusively a push style service. Considering the wide spread acceptances and availability of cellular phones with cameras, and the ability of computers to identify persons in a photograph it is logical that a system like Amber Alert™ would utilize these capabilities to provide enhanced services to users having camera phones to quickly identify if a child they see is in fact the subject of the alert. Additionally, Amber Alert™ could send in addition to SMS messages, MMS messages to provide an actual reference image of the subject to help users better recognize the subject.

Push messaging services can be very helpful for communities as evident by the Amber Alert™ system and other systems be experimented with internationally. Community Tips presently does not support any push style services because all push style communications exist outside the core business of Community Tips to enable anonymous tipping via SMS, MMS and

the internet. As Community Tips evolves and matures push-messaging services may be added but at present there are no specific plans for any services of that nature.

4.2 Response From Law Enforcement

Since the initial meeting to propose the idea of this project law enforcement has been excited and offered what limited help they can provide to see this project come to fruition. The initial concept of this system anticipated the basic functional needs of receiving a tip and connecting tips together via electronic records in a database. Tips received via a message service would be anonymous, tips received via the web application users would be by default anonymous but could choose to identify themselves. Additionally the concept included the functionality to return a record identifier to the tip submitter so that they could return to it and provide additional information. This record identifier was not an attempt to identify the tip provider, simply a mechanism to support follow up information which could be easily connected to the original message.

In the first interaction to layout this basic concept of system the concept as a whole was well received. The exception was that having a record identifier sent back to the tip provider was perceived as a negative feature which would hinder public acceptance of the system. The motivation for this perception was that if the tip was expected to be anonymous getting a specific reply with an identification number it could be easily misconstrued to be linking the tip provider to the tip. While this was not the case there was not a clean solution to allow the user direct access back to the tip in the database. The compromise was to have a generic reply message, “Thank you for your anonymous tip. Please visit <http://www.communitytips.org> for more information about this service.”. This message ideally solidifies the anonymousness of the interaction and will enable the tip provider to provide additional information even if they cannot link it to the original tip.

In the second interaction with law enforcement the system’s two key components, the SMS interface and web application were demonstrated. The web application had basic abilities to tip (Figure 3.3 and Figure 3.4), prompt the user to verify the tip (Figure 3.5 without the additional questions), and finally thanking the user for using the system. Both the SMS interface and web application were well received but for both components a request for additional

information from the user was put forth. The additional information being sought from the user was more specific information such as describe a person involved, provide a location, provide a description of a vehicle involved. For the SMS interface this request is impossible due to the technological limitations of SMS, in that they are analogous to email. A recipient of an email cannot a priori prompt the sender for specific information within the original email. That is to say if Jane sends Joe an email, Joe cannot enumerate a set of questions for Jane to reply to without some initial exchange. And due to the judgment made in the original discussion for the system, the user should not receive a specific reply back, such as asking for the specifically enumerated information. For the web application adding additional questions did not pose a significant problem to the model, it was simply a choice of where did the additional questions belong. The end result was placing the questions on the verification page (Figure 3.5) so as to not overwhelm a potentially hesitant user with a multitude of questions. Bearing the potential nervousness of a user in mind it was also decided in development to use a JavaScript accordion feature to ask high level questions which the user could drill down into if they could provide additional information. As a result the user is presented with the high level questions *Describe the [Person / Place / Vehicle]* in Figure 3.5 which can then drill down accordingly to Figure 3.6 for Person, Figure 3.7 for Place and Figure 3.8 for Vehicle.

During the third interaction with law enforcement the web application was again well received and the implement choice of asking high-level questions to probe for additional information was validated. A police detective who was reviewing the system stated from personal experience of interviewing witnesses and receiving tips that not inundating the user with many questions some of which may either be inapplicable or unable to answer was a positive choice. A few additions to the set of questions being ask was requested, specifically in the Person Description (Figure 3.6), a prompt to describe any weapons involved and a prompt to characterize the person being described: victim, suspect, witness or unclear. The Vehicle description had two modifications, one a clear omission, license number. The second modification was from practical experience of the detective, the drop down box Vehicle Type initially had the selections: Coup, Sedan, Truck, Van, etc. The problem with these selections is the general public at times can be unclear that a coup in automotive jargon is a vehicle with two side mounted doors for passengers to enter, and a sedan has four side mounted passenger entry doors. The solution was to substitute Coup with Coup (2 doors) and Sedan with Sedan (4 doors).

On the surface this may seem a trivial change of language, but bear in mind the tip provided may be used in court as evidence in legal proceedings. If the user states in the tip a two door red car was involved but the users then in describing the vehicle selects a red sedan, it would appear that the user is confused. By providing additional description for these automotive terms this misunderstanding can be mitigated.

4.3 Response From Potential Users

Acquiring real user feedback during development is similar to the chicken and the egg problem. Does the system get built and go into production and then ask for feed back from actual users or should person with prior anonymous tipping experience be recruited to test the system during development. Neither of these scenarios is particularly viable. The first option of asking actual users to be the guinea pigs has a high potential for hamper future use by creating a negative perception in the user base. The second choice of soliciting users with prior anonymous tipping experience is not feasible because there is no practical way to determine if they have the requisite prior experience. The solution is to create “model” users.

Model users of this system are community members who may or may not have submitted tips to law enforcement in the past but have enough computer literacy or SMS experience to be able to carry out a defined task with this system. The defined tasks were actually the set of test cases developed for this system, which provide comprehensive coverage of the actions and selections a user may make. The test cases, which model users executed provided two key benefits: direct user feed back on experience with the system and providing additional bug testing.

In total sixteen test cases were developed to test the primary the web application and messaging interface. These test cases covered a wide variety of permutations of providing a tip, a few examples are:

- Provide an anonymous tip
- Provide an anonymous tip with a digital picture
- Provide a non-anonymous tip
- Provide an anonymous tip with a suspect description

The test cases provide high-level directions as to which sections the user should fill out and whether or not to include a picture from a provided library of pictures. What particular data is entered into which fields is left to the user's discretion for all test cases. This approach was taken to reflect the variability in real world usage. Some users may provide all the details requested some may provide none or a small subset. Also actual input will vary depending on user, while two people may see the same event they are not guaranteed to describe it in identical terms nor necessarily interpret what they have seen in the same way. To prepare model users a brief note explaining the purpose of the test case was provided to all users. Refer to Appendix B for the user interface test plan for Community Tips, Appendix C for the test cases, and Appendix D for the note to users regarding the test cases.

A total of thirty-five model users that completed the test cases, they were drawn from introductory computer sciences classes teaching basic computer literacy skills. The only two test cases which these users did not complete tested the SMS and MMS interface due to difficulty of cellular reception inside the building which the tests were conducted in. In the process of executing the test cases users identified some deficiencies in the system which needed to be remedied. The deficiencies were:

- Failure to appropriately handle all cases of user punctuation marks within input which caused database insert exceptions
- Database field sizes which were too narrow for the user's input
- An incorrect method for validating file upload types

All the deficiencies were remedied after the source of the problem was identified and a solution to resolve the problem without generating other errors was found.

The issue involved with user punctuation marks was that the database input sanitizer class does not handle all database input violations. The goal of the sanitizer class was to preserve user input as originally supplied, but at the same time to protect the database. The sanitizer input is processed to replace certain control characters with HTML equivalent encoding so that the original text will be displayed while simultaneously protecting the database. One control character, a single forward-slash, (\), is replaced with the equivalent HTML encoding, `\`, the punctuation mark which was not handled correctly was a single quotation mark, ('), which should be replaced with an `'`. A single quote is a string terminator in the database

language SQL unless it is escaped via a control character, rather than altering the text by adding additional characters it makes more sense to preserve the original document but simply to re-encode some characters. For example the first character which is re-encoded is an ampersand, (&), so that other re-encodings which follow are not over written with an ampersand re-encoding.

The second issue, field sizes, was simply a poor choice at anticipating the verbosity of some users. The initial field size for a suspect's clothing was fifty characters; this was barely half of what one particular user wanted to write. Given the relatively low cost of memory and negligible effect on database performance by enlarging a fixed size field, it was altered from fifty characters to two hundred and fifty-five. Several other fields were also re-evaluated and enlarged from fifty characters to two hundred and fifty-five.

The third issue was a poor design choice on validating correct file upload types. Model users were utilized on two different occasions, on the second a change to the system attempted to better validate file uploads using the file content type attribute of the file object during upload. This attribute is not correctly set at the time of validation; instead the system was changed to validate files by checking the file extension type rather than content type.

Additionally after the users completed their test case they were asked to take a survey of nineteen questions. Instead of providing a test plan document, user's were provided with a brief note explaining what they were about to do. In addition a brief oral introduction explaining the purpose of Community Tips was given before the test cases were distributed. The first section covered the web application followed by two questions about the user's acceptance of messaging in community policing, and finally three questions providing feedback about the user's experience with the test case.

The first section, questions one through four, of the survey covering the web application asked questions related to content, layout, navigation, and cross-browser support. The first question presented on the survey 88% of respondents agreed that the site's content made sense, that is to say the content was coherent and relevant. Of the four respondents who did not feel agree, two supplied textual explanations to their response to the content related survey questions. One of the two respondents was confused by the directions in the test case, the second

respondents was confused by an error in the system's handling of file uploads. The same two respondents both agreed that the content was appropriate and was not ambiguous. Given these two additional replies the actual positive user response to the site's content is likely better than 90%.

The second set of questions in section one, questions five, six, and seven, dealt with the layout of the site. This set of questions problem re-affirmed a known issue with Microsoft's Internet Explorer failing to correctly render the CSS. Out of twelve users who provided written commentary about the layout, ten comments were related to the failure of Internet Explorer with CSS. Of the other two comments, one somewhat irrelevant, it was simply a comment comparing the layout to another unrelated website. The second comment suggested changing the flow of questions, moving the description questions to the main tip section on the front page. This suggestion highlights a purposeful design choice. When a user submits a tip, the goal is to obtain as much information about the tips while at the same time creating a minimum amount of overhead for the user while supplying the tip. The search engine Google's success has been attributed, to a certain degree, to its interface design in that it is simple, straightforward and uncluttered. By initially presenting the user with a direct means to submit an anonymous tip and then following up with supplement questions, Community Tips is emulating the user interface design that Google has been so successful with.

The third and fourth sections, questions eight through thirteen provided little substantive value do to a lack commentary from users. Section three asked two questions regarding site navigation; section four asked about inter-browser appearance and functionality. Only one survey respondent accessed Community Tips from more than one browser. This single user only noted the correct CSS rendering in Firefox and incorrect rendering in Internet Explorer.

The fourteenth question asked users if they would use this site. More than 80% of respondents indicated that they would. In future uses of this survey expanding this question into several other questions may be more a more constructive insight. For example, instead asking only specifically about this site asking about a site similar to this, asking if they have every submitted an anonymous tip or event a police report. Also providing a short answer section which for other sections of the survey provided informative insight.

The second section regarding user's acceptance of messaging in community policing had interesting results. The first question, "Would you use anonymous SMS (text messaging) or MMS (picture/video messaging) to communicate with law enforcement", more than 80% of users agreed that they would. But when asked, "Would you like to receive information via SMS about public safety issues", barely 40% of users agreed that they would. This is a significant difference and merits further investigation. Some questions which can be explored are: is this difference due to type of service (pull vs. push), was this difference due to the context in which the questions were being asked, would providing examples of informational message affect user's response.

The third section covering users experience performing the test cases to improve and refine them. Question seventeen asked, "Did the test directions make sense", 31 of 34, or slightly better than 90% of respondents stated that they did. Of the three respondents who didn't all three provide some written description of what the source of confusion was. The first respondent with commentary states, "I was confused about what to describe about the outcome", all test cases both in intermediate steps and as a final question ask the user what happened after they performed some action, to gauge their perception of what the system is doing. Typically users described what the screen was currently displaying: error messages (the test cases force users to provide erroneous input to verify system functionality), subsequent screens in the process, or in some cases of a system malfunction they describe no change in after the action. This phrasing of this question will be investigated to see if there is a better alternative phrasing. The second comment was " [I] was confused about #5 and 6 and why I had to do thos [sic] steps" highlighted a deficiency in the survey; there was no indication which test case was being commented on. In the future a required radio button or drop down box enumerating the test cases will be included so comments like this can be properly evaluated in context. The third comment, "They need to be more clear" is likewise not illuminating because there is little context of which test case's directions were unclear. This comment was made after the user was given an explanation of the test cases with the statements, "... some instructions [in the test case] may seem redundant, incomplete, or otherwise ill-advised. Please be assured that the steps in the instructions have been carefully chosen." Some instructions were crafted purposely to check validation routines in the software, those perhaps are the instructions that were perceived as unclear.

5.0 Conclusions and Future Work

5.1 Conclusions

5.1.1 Project Summary

Change is a natural part of the world. Two generations ago college students boarding jet planes to spend spring break on sunny beaches would have been unimaginable. A decade ago the Pew Foundation did not have to use special sampling techniques to get correct age and gender proportions for surveys of the American populace in telephone surveys. Law enforcement has transitioned from primitive bludgeoning weapons, billyclubs, to fire arms to non-lethal weapons to keep pace with changing life-styles, attitudes and technology. It is imperative for community policing efforts to keep pace with modern life styles to maintain their relevance and efficacy. One specific aspect of modern life which needs to be embraced and utilized to its fullest extent are cell phones which are widely available and a vital part of everyday life for many age groups.

Three examples of community policing efforts utilizing SMS messaging in conjunction with the internet have been presented, Amber Alert, Citizen Observer™™ and Crimeline™. Two of these, Amber Alert™ and Crimeline™, are nationwide endeavors. Citizen Observer™™ operates at a local level interfacing directly with local law enforcement agencies. Amber Alert™ functions solely as a means of transmitting information out to relevant communities using SMS messaging. Citizen Observer™ and Crimeline™ actively solicit community involvement to solve crimes by providing an SMS and internet based interface to law enforcement. Citizen Observer™ offers additional functionality over Crimeline™ by providing an interface which law enforcement can use to manage the data that is provided.

Community Tips provides, from the community user's perspective a similar service to Crimeline. The key differences between the two applications are that Community Tips offers more functionality from the internet by allowing users to upload videos and provide additional information that is not directly requested or supported by Crimeline™, such as a physical description of the location, vehicle description and the ability to describe persons other than

suspects involved. Beyond allowing anonymous tipping, Community Tips also gives law enforcement an interface to manage, analyze, and link tips.

Comparing Community Tips to Citizen Observer™ outwardly they have the same functionality for user submission. Community Tips differs from Citizen Observer™ from the user’s perspective significantly in how anonymity is handled and law enforcement’s ability to have follow-up contact with users. Citizen Observer™ creates an alias for all anonymous tip providers to protect their identity from law enforcement; Community Tips maintains no user specific data. For law enforcement to find the origin of the tip, they would have to contact the SMS aggregator with a court order. From the law enforcement perspective Community Tips provides a more comprehensive interface for analyzing and managing tips. An overview of the key features available in the three systems is presented in Table 5.1.

Table 5.1 Summary of key features in online community policing systems

	Crimeline	Citizen Observer	CommunityTips.org
Anonymous SMS tip support	X	X	X
Anonymous internet tip support	X	X	X
Bi-direction SMS support		X	
Support for digital pictures / videos		X	X
Clear privacy statement	X		X
Law enforcement interface to review tips		X	X

Local law enforcement, since the outset of this project, has been a constant supporter. They have provided invaluable direction and insight into what their needs are; by providing practical experience of interviewing community members to obtain necessary information concerning crimes. During law enforcement’s review of system functionality they were able to provide cross validation of user interface design choices, e.g. extensive use of drop-down selection boxes and providing users high-level questions which can then be drilled down into for more specific questions.

Test users responded favorably to the Community Tips system. Eighty-eight percent of surveyed users responded favorably to the user interface content. Of those who did not like the layout, two-thirds provided feedback that was used to improve the interface, or described known browser display problems that were resolved. While some response to the user survey merit more investigation, the survey results as a whole in combination with the errors and solutions identified in user testing support the goal for moving the system forward into alpha release.

Community Tips is still a young system, currently approaching alpha release, with an ultimate goal of transition into production and then commercial licensing or sale. It follows in the same vein as other systems being actively developed as both commercial and non-commercial products, and offers additional functionality over all of them. In the short term the system will continue to be hosted on university equipment but as long term plans evolve continuity of service will be maintained at the URL www.communitytips.org.

5.1.2 Lessons Learned

Development of Community Tips was undertaken as a final capstone project for a Masters of Computer Science and Information Systems. As such, developing a coherent system with real world applications was of great importance, but an implicit goal of the project was to demonstrate competencies arising from the course work execute as part of the degree program. This course work included topics such as software engineering, database design, analysis and design of algorithms, and analysis modeling and design of software systems to name a few key areas.

The course work dealing with the analysis modeling and design of software systems was concerned with the life cycle process of developing software from the initial inception phase clear through to the terminal phase, maintenance and support. Community Tips has negotiated requirements analysis, initial development, and testing, in the software life cycle. The skills drawn on from this course work were primarily the requirements gathering and analysis. Requirements from both of the stakeholder groups could never be fully elicited, but enough could be gathered to weave together a coherent functional system that meets the user needs and known requirements. The system analysis gave good direction on how to proceed in terms of making correct technological choices to allow this system to move forward. One practical lesson

that could not be adequately learned from course work has been time and project management. At certain points in the development process miscommunications of responsibilities or priorities occurred leading to timeline slippage. These occurred both in scheduling interactions with the stakeholder groups and with interim development goals.

Also while the technology chosen has been a positive choice, it has led to some issues due to the fact some libraries used are the bleeding edge of development. In one instance a library, Apache MyFaces Core, which is publicly stated to be in stable release 1.2 had a flaw in a Tag Library Description for input objects that most other MyFaces Core objects were based off. The flaw prevented the Tomcat from successfully processing the input objects and caused a complete rendering failure. This flaw resulted in the originally library being un-usable due to the severity of the flaw. Determining the flaw was in the library and not in the specific application of the library and finally finding a solution required almost a week's worth of effort. The solution was found after significant searching that there was in fact a defect in the stable library that was fixed in the source code repository, as a result the library used in development of Community Tips was generated from the open source code repository hosting the development of that library.

Database design and analysis was another key set of skills that were required to develop this system. To implement this system a total of twenty tables and eighteen stored procedures were required. Two direct examples of this lessons learned in this course work are how the data model takes on object abstraction approach to handling "people". Both non-anonymous tipper providers and police users are rooted in the person table so that there is one person identifier key being generated and searched through rather than two, one for each kind of person being maintained in the system. Another key lesson learned from the course work was abstraction and compartmentalization of stored procedure functionality so that if the underlying table structure changes in the database a minimal number of stored procedures must be altered to manage the change. A specific example is the change of the system to support full text searching of tips. All forms of tips, (anonymous, anonymous with digital media, non-anonymous, non-anonymous with digital media) all are rooted in a stored procedure publicAnonymous which completes the insert of the tip into the events table. To support full text searching a secondary table eventsft was required which was a different table type (MyISAM as opposed to INNODB) as seen in

Figure 3.3. This second table simply required add a single insert statement into the publicAnonymous stored procedure and no modification to any calls to the publicAnonymous procedure. If all or any other statements executed inserts into the events table, the events and eventsft table could easily become out of synchronization, which is assumed and required for correct functionality.

5.2 Future Work

There are several areas which this system can be expanded. A few are automatic photographic and videographic analysis and tagging, visualization of geographic information, integration with third party systems, system segmentation for expansion to new geographic areas or municipalities. Some of these areas are beyond the expertise of the current developer, specifically photographic and videographic analysis. These analysis hinge on pattern recognition algorithms to locate key features in data, for example locating and reading license plates in photographs or locating and extracting faces. The ability to extract this data would dovetail well with connectivity to third party systems, such as a state department of motor vehicles which have complete information about license plates. Visualization of geographic information could be accomplished through the Google Maps API to display maps with physical addresses provided in tips to see emergent patterns or trends that may not be apparent in textual tips. Additionally some refinements to the system are: archiving for tips that are resolved and additional testing to resolve an intermittent page rendering failure by Tomcat.

References

1. A Quarter of Elementary School Kids Have Cell Phones. (2005, September 28). Mobiledia from <<http://www.mobiledia.com/news/37113.html>>
2. Bontis, N., Serenko, A., Tural, O. (2007, January). User acceptance of wireless short messaging services: Deconstructing perceived value. **Information & Management Volume 44, Issue 1**, 63-73.
3. The Cell Phone Challenge to Survey Research National Polls Not Undermined By Growing Cell-Only Population. (2006, May 15). The Pew Research Center for the People & the Press from <<http://people-press.org/reports/display.php3?ReportID=276>>
4. Citizen Observer Do More With Less. Citizen Observer. <http://server4.citizenobserver.com/training/demo/1_easytouse/1_easytouse/shell.swf>
5. City of Wilmington Public Information Report. (Fall 2007). Volume 4, Number 4.
6. Community participation leads to 71 crime line arrests, Pretoria. (2007, April 17). South African Police Service. <<http://www.info.gov.za/speeches/2008/08041811451004.htm>>
7. CRIME LINE :: Your Anonymous Crime Tip-off Line. Crimeline. <<https://www.crimeline.co.za/Tipoff.asp>>
8. Google Mobile. Google. <http://www.google.com/intl/en_us/mobile/sms/>
9. How Wireless AMBER Alerts™ are sent. Wireless Amber Alerts <<http://wirelessamberalerts.adcouncil.org/howwirelessamberalertswork.htm>>
10. Laplante, P., Neill, C. "The Demise of the Waterfall Model Is Imminent" and Other Urban Myths. (2004, February). ACM Queue Volume 1, Number 10. <<http://www.acmqueue.com/modules.php?name=Content&pa=showpage&pid=110>>
11. Lowe, D., Pressman, R. (2008). Web engineering a practitioner's approach. New York: McGraw-Hill Higher Education.
12. Microsoft ASP.NET MVC <<http://www.asp.net/mvc/>>

13. Myhill, A., Rudat, K. Community engagement in policing: case-study evaluation of a demonstration project in Cheshire. <<http://police.homeoffice.gov.uk/news-and-publications/publication/police-reform/Cheshire-case-study.pdf?view=Binary>>
14. Partners and Sponsors. Wireless Amber Alerts
<<http://wirelessameralerts.adcouncil.org/partners.htm>>
15. Spiral Model. (2008, June 4). Wikipedia.
<http://en.wikipedia.org/wiki/Spiral_model>
16. Ten Suspects. (2007 June 15). Crimeline.
<https://www.crimeline.co.za/ten_suspects.asp>
17. Terms. Crimeline.
<<https://www.crimeline.co.za/terms.asp>>
18. Trends In Telephone Service. Federal Communications Commission. (2007, February).
Table 11-2.
19. TipSoft SMS. Anderson Software.
<<http://www.smscrimetips.com/>>
20. TipSoft SMS “How It Works” Anderson Software.
<http://www.smscrimetips.com/sms_process_flow.pdf>
21. Vetter, R. Brown, J. Shipman, S. “SMS: The Short Message Service” (2007 December)
IEEE Computer. Volume 40, Issue 12, 106-110.
22. Waterfall Model. (2008, June 18). Wikipedia.
<http://en.wikipedia.org/wiki/Waterfall_model>
23. Web Tips. Citizen Observer.
<<https://www.tipsubmit.com/citobs/WebTips.aspx?AgencyID=618>>
24. Wireless Amber Alerts.
<<https://www.wirelessameralerts.org/index.jsp>>

Appendix B

Test Plan

Purpose:

To fully test the web application www.communitytips.org and associated services to prove correct functionality as well as to locate any runtime errors in preparation for the alpha release in June 2008.

Scope:

This test will cover all user interface components viewable to all types of users at www.communitytips.org as well as the application interface for SMS (texting), which is the short code 90947 (keyword “tip”).

Objectives:

- Identify all runtime errors in the system which can be generate via the primary user interfaces; both the website and SMS.
- Solicit user feedback on: design, layout and functionality
- Gather sufficient information in regards to the aforementioned characteristics of the system to make appropriate changes to enable the highest quality product to be rolled out as a alpha release.

Method:

Test cases will be generated to cover all aspects of the user interface to enable black-box testers to provide complete coverage for the scope of this document. These tests will include inputs for both valid and invalid inputs as well as boundary cases where appropriate. These test cases will also be representative of real world use cases. All tests will be performed on both of the major web browsers in use today, Microsoft Internet Explorer and Mozilla Firefox.

Additionally a follow-up survey will be utilized to gather uniform data across all participants performing the test. The survey will include questions relating to:

- Content
- Design
- Navigability
- Cross-browser compatibility

Appendix C

Test Case 1

Scenario A

Purpose:

Test for a non-registered user submitting an anonymous text only tip.

Precondition:

Use Microsoft Internet Explorer, navigate to <http://dev.communitytips.org>

Test:

1. Click "Submit".
 2. Describe out come:
-
3. Provide a fictional tip of a crime in the text box under the heading "submit a tip".
 4. Click "Submit" [Continue forward until either a Thank You message or an error occurs].
 5. Describe out come:
-

Test Case 1

Scenario B

Purpose:

Test for a non-registered user submitting a text tip with contact information.

Precondition:

Use Microsoft Internet Explorer, navigate to <http://dev.communitytips.org>

Test:

1. Click the link "Click To Provide Contact Information"
 2. Provide some fictional information
 3. Click "Submit".
 4. Describe out come:
-
5. If the contact information provided in step 2 is no longer visible or present please repeat steps 1 and 2.
 6. Provide a fictional tip of a crime in the text box under the heading "submit a tip".
 7. Click "Submit" [Continue forward until either a Thank You message or an error occurs].
 8. Describe out come:
-

Test Case 1

Scenario C

Purpose:

Test for a non-registered user submitting an anonymous text with a media file.

Precondition:

Navigate to <http://photos.communitytips.org> and download any image to your desktop, remember the name of the file.

Use Microsoft Internet Explorer, navigate to <http://dev.communitytips.org>

Test:

1. Click “Browse”
 2. Select the image you downloaded in the **Precondition** above.
 3. Click “Submit”.
 4. Describe out come:
-
5. If the file name is no longer listed in the text box next to browse please Click “Browse” again and select the image you downloaded in **Preconditions**.
 6. Provide a fictional tip of a crime in the text box under the heading “submit a tip”.
 7. Click “Submit” [Continue forward until either a Thank You message or an error occurs].
 8. Describe out come:
-

Test Case 1

Scenario D

Purpose:

Test for a non-registered user submitting a text tip with contact information and a media file.

Precondition:

Navigate to <http://photos.communitytips.org> and download any image to your desktop, remember the name of the file.

Use Microsoft Internet Explorer, navigate to <http://dev.communitytips.org>

Test:

1. Click “Browse”
 2. Select the image you downloaded in the **Precondition** above.
 3. Click the link “Click To Provide Contact Information”
 4. Provide some fictional information
 5. Click “Submit”.
 6. Describe out come:
-
7. If the file name is no longer listed in the text box next to browse please Click “Browse” again and select the image you downloaded in **Preconditions**.
 8. If the contact information provided in step 4 is no longer visible or present please repeat steps 3 and 4.
 9. Provide a fictional tip of a crime in the text box under the heading “submit a tip”.
 10. Click “Submit” [Continue forward until either a Thank You message or an error occurs].
 11. Describe out come:
-

Test Case 2

Scenario A

Purpose:

Test for a non-registered user submitting a text tip with contact information.

Precondition:

Use Microsoft Internet Explorer, navigate to <http://dev.communitytips.org>

Click the “About” tab in the upper right corner.

Test:

1. Click “Submit”.
 2. Describe out come:
-
3. Provide a fictional tip of a crime in the text box under the heading “submit a tip”.
 4. Click “Submit” [Continue forward until either a Thank You message or an error occurs].
 5. Describe out come:
-

Test Case 2

Scenario B

Purpose:

Test for a non-registered user submitting a text tip with contact information.

Precondition:

Use Microsoft Internet Explorer, navigate to <http://dev.communitytips.org>

Click the “About” tab in the upper right corner.

Test:

1. Click the link “Click To Provide Contact Information”
 2. Provide some fictional information
 3. Click “Submit”.
 4. Describe out come:
-
5. If the contact information provided in step 2 is no longer visible or present please repeat steps 1 and 2.
 6. Provide a fictional tip of a crime in the text box under the heading “submit a tip”.
 7. Click “Submit” [Continue forward until either a Thank You message or an error occurs].
 8. Describe out come:
-

Test Case 2

Scenario C

Purpose:

Test for a non-registered user submitting an anonymous text with a media file.

Precondition:

Navigate to <http://photos.communitytips.org> and download any image to your desktop, remember the name of the file.

Use Microsoft Internet Explorer, navigate to <http://dev.communitytips.org>

Click the “About” tab in the upper right corner.

Test:

1. Click “Browse”
 2. Select the image you downloaded in the **Precondition** above.
 3. Click “Submit”.
 4. Describe out come:
-
5. If the file name is no longer listed in the text box next to browse please Click “Browse” again and select the image you downloaded in **Preconditions**.
 6. Provide a fictional tip of a crime in the text box under the heading “submit a tip”.
 7. Click “Submit” [Continue forward until either a Thank You message or an error occurs].
 8. Describe out come:
-

Test Case 2

Scenario D

Purpose:

To test for a non-registered user submitting a text tip with contact information and a media file.

Precondition:

Navigate to <http://photos.communitytips.org> and download any image to your desktop, remember the name of the file.

Use Microsoft Internet Explorer, navigate to <http://dev.communitytips.org>

Click the “About” tab in the upper right corner.

Test:

1. Click “Browse”
 2. Select the image you downloaded in the **Precondition** above.
 3. Click the link “Click To Provide Contact Information”
 4. Provide some fictional information
 5. Click “Submit”.
 6. Describe out come:
-
7. If the file name is no longer listed in the text box next to browse please Click “Browse” again and select the image you downloaded in **Preconditions**.
 8. If the contact information provided in step 4 is no longer visible or present please repeat steps 3 and 4.
 9. Provide a fictional tip of a crime in the text box under the heading “submit a tip”.
 10. Click “Submit” [Continue forward until either a Thank You message or an error occurs].
 11. Describe out come:
-

Test Case 3

Scenario A

Purpose:

Test for a non-registered user submitting a text tip with contact information.

Precondition:

Use Microsoft Internet Explorer, navigate to <http://dev.communitytips.org>

Click the “Contact” tab in the upper right corner.

Test:

1. Click “Submit”.
 2. Describe out come:
-
3. Provide a tip of a fictional criminal in the text box under the heading “submit a tip”
 4. Click “Submit” [Continue forward until either a Thank You message or an error occurs].
 5. Describe out come:
-

Test Case 3

Scenario B

Purpose:

Test for a non-registered user submitting a text tip with contact information.

Precondition:

Use Microsoft Internet Explorer, navigate to <http://dev.communitytips.org>

Click the “Contact” tab in the upper right corner.

Test:

1. Click the link “Click To Provide Contact Information”
 2. Provide some fictional information
 3. Click “Submit”.
 4. Describe out come:
-
5. If the contact information provided in step 2 is no longer visible or present please repeat steps 1 and 2.
 6. Provide a fictional tip of a crime in the text box under the heading “submit a tip”.
 7. Click “Submit” [Continue forward until either a Thank You message or an error occurs].
 8. Describe out come:
-

Test Case 3

Scenario C

Purpose:

Test for a non-registered user submitting an anonymous text with a media file.

Precondition:

Navigate to <http://photos.communitytips.org> and download any image to your desktop, remember the name of the file.

Use Microsoft Internet Explorer, navigate to <http://dev.communitytips.org>

Click the “Contact” tab in the upper right corner.

Test:

1. Click “Browse”
 2. Select the image you downloaded in the **Precondition** above.
 3. Click “Submit”.
 4. Describe out come:
-
5. If the file name is no longer listed in the text box next to browse please Click “Browse” again and select the image you downloaded in **Preconditions**.
 6. Provide a fictional tip of a crime in the text box under the heading “submit a tip”.
 7. Click “Submit” [Continue forward until either a Thank You message or an error occurs].
 8. Describe out come:
-

Test Case 3

Scenario D

Purpose:

To test for a non-registered user submitting a text tip with contact information and a media file.

Precondition:

Navigate to <http://photos.communitytips.org> and download any image to your desktop, remember the name of the file.

Use Microsoft Internet Explorer, navigate to <http://dev.communitytips.org>

Click the “Contact” tab in the upper right corner.

Test:

1. Click “Browse”
 2. Select the image you downloaded in the **Precondition** above.
 3. Click the link “Click To Provide Contact Information”
 4. Provide some fictional information
 5. Click “Submit”.
 6. Describe out come:
-
7. If the file name is no longer listed in the text box next to browse please Click “Browse” again and select the image you downloaded in **Preconditions**.
 8. If the contact information provided in step 4 is no longer visible or present please repeat steps 3 and 4.
 9. Provide a fictional tip of a crime in the text box under the heading “submit a tip”.
 10. Click “Submit” [Continue forward until either a Thank You message or an error occurs].
 11. Describe out come:
-

Test Case 4

Navigational Test

Purpose:

To test inter-page navigation.

Precondition:

Use Microsoft Internet Explorer, navigate to <http://dev.communitytips.org>

Test:

Please mark through correct browser in use for any steps which do not navigate to the indicated page.

Ex. 51. Click the CNN tab	Internet Explorer	Firefox
---------------------------	------------------------------	---------

If the step works correctly in both browsers please indicate with a check mark

Ex. 52. Click the ESPN tab	Internet Explorer	Firefox√ (OK)
1. Click the "CONTACT" tab	Internet Explorer	Firefox
2. Click the "ABOUT" tab	Internet Explorer	Firefox
3. Click the "ADMIN ACCESS" tab	Internet Explorer	Firefox
4. Click the "HOMEPAGE" tab	Internet Explorer	Firefox
5. Click the "ABOUT" tab	Internet Explorer	Firefox
6. Click the "CONTACT" tab	Internet Explorer	Firefox
7. Click the "ADMIN ACCESS" tab	Internet Explorer	Firefox
8. Click the "CONTACT" tab	Internet Explorer	Firefox
9. Click the "HOMEPAGE" tab	Internet Explorer	Firefox
10. Click the "ADMIN ACCESS" tab	Internet Explorer	Firefox
11. Click the "ABOUT" tab	Internet Explorer	Firefox
12. Click the "HOMEPAGE" tab	Internet Explorer	Firefox
13. Click the "HOMEPAGE" tab	Internet Explorer	Firefox
14. Click the "ADMIN ACCESS" tab	Internet Explorer	Firefox

Navigational Test

- | | | |
|-----------------------------------|-------------------|---------|
| 15. Click the “ADMIN ACCESS” tab | Internet Explorer | Firefox |
| 16. Click the “ABOUT” tab | Internet Explorer | Firefox |
| 17. Click the “CONTACT” tab | Internet Explorer | Firefox |
| 18. Click the “CONTACT” tab | Internet Explorer | Firefox |
| 19. Provide any addition comments | | |

Test Case 5

Scenario A

Purpose:

To test the functionality of cellular message via SMS.

Precondition:

SMS enabled cell phone.

Test:

1. Create an SMS (text message) addressed to the short code 90947.
 2. The first word must be “tip” exactly.
 3. Following the word tip compose a tip about a fictional criminal event.
 4. Describe the reply text message
-

Test Case 5

Scenario B

Purpose:

To test the functionality of cellular message via MMS.

Precondition:

MMS enabled cell phone.

Test:

1. Create an MMS (multimedia message) addressed to the short code 90947.
 2. The first word must be “tip” exactly.
 3. Following the word tip compose a fictional tip about a crime.
 4. Attach a picture which is related to the fictional tip.
 5. Describe the reply text message
-

NOTE:

THIS SCENARIO IS NOT VALID AS OF 4/10/2008.

MMS service is still unavailable from Mobile Education, LLC.

When MMS becomes available revise accordingly and implement this test.

Test Case 5

Scenario B

W.Shipman

Test Case Survey

Purpose:

To gather feedback from test users regarding their experience with www.communitytips.org as well as with the testing process.

Directions:

1. Go to <http://dev.communitytips.org>
2. Click the “CONTACT” tab on the upper right corner.
3. Click the link for “SURVEY” and complete the survey.

Appendix D

A note to the Testers,

Today you are being asked to perform user testing on the web application at <http://dev.communitytips.org>. This web application is being developed along with an SMS (text message) interface to provide a new form of anonymous tip service to law enforcement. This project will serve as a reference for future work involving inter-connecting data services available on the web with mobile computing environments.

Today you will be testing the web application, the SMS interface, or possibly both. These tests are to ensure correct operation of the system. Bearing this in mind, some instructions may seem redundant, incomplete, or otherwise ill advised. Please be assured that the steps in the instructions have been carefully chosen. Please perform the test exactly as stated.

This application is the capstone project for Bill Shipman who is in his final semester in the UNCW MSCSIS program. Your help in this process is greatly appreciated both for creating a better system and providing test data which will be incorporated in the final write up of this project.

Thank you again for you time and help.

Bill Shipman