

Utah State University

DigitalCommons@USU

All Graduate Plan B and other Reports

Graduate Studies

5-2015

Crisis Social Media Trainer

Sravani Mamidi

Utah State University

Follow this and additional works at: <https://digitalcommons.usu.edu/gradreports>

Recommended Citation

Mamidi, Sravani, "Crisis Social Media Trainer" (2015). *All Graduate Plan B and other Reports*. 498.
<https://digitalcommons.usu.edu/gradreports/498>

This Report is brought to you for free and open access by the Graduate Studies at DigitalCommons@USU. It has been accepted for inclusion in All Graduate Plan B and other Reports by an authorized administrator of DigitalCommons@USU. For more information, please contact digitalcommons@usu.edu.



CRISIS SOCIAL MEDIA TRAINER

by

Sravani Mamidi

A report submitted in partial fulfillment
of the requirements for the degree

of

MASTER OF SCIENCE

in

Computer Science

Approved:

Dr. Amanda Hughes
Major Professor

Dr. Curtis Dyreson
Committee Member

Dr. Tung Nguyen
Committee Member

Dr. Mark R. McLellan
Vice President for Research and
Dean of the School of Graduate Studies

UTAH STATE UNIVERSITY
Logan, Utah

2015

Copyright © Sravani Mamidi 2015

All Rights Reserved

ABSTRACT

Crisis Social Media Trainer

by

Sravani Mamidi, Master of Science

Utah State University, 2015

Major Professor: Dr. Amanda Lee Hughes

Department: Computer Science

For this project, I designed, developed, and evaluated an application called the Crisis Social Media Trainer (CSMT) which helps emergency responders practice using social media. This application simulates the social media activity that one might encounter during a crisis event.

I began by iteratively designing and prototyping the CSMT application. After I created a stable prototype, I evaluated it. First, I performed cognitive walkthroughs and heuristic analyses on the prototype. After making improvements based on the results of the walkthroughs and analyses, I tested the prototype with four emergency responders. This user testing revealed many areas of improvement for the application and demonstrated that CSMT could be a useful social media training tool for emergency responders. Finally, I modified the CSMT prototype to address the issues encountered by the users during testing.

DEDICATION

To my parents, my sister Sowjanya Mamidi and my major professor Dr. Amanda Lee Hughes who constantly supported me in completing this project successfully.

ACKNOWLEDGMENT

I start by acknowledging God for all the support and strength He gave me to make all my dreams come true.

I also want to acknowledge my major professor Dr. Hughes—without her, this work would not have been possible. I am much obliged to her. Her quiet nature and uplifting state of mind towards exploration and life has constantly encouraged me to work more. I thank her for all the guidance and time she provided for me. She helped at every stage to finish this project successfully. I thank her for having confidence in me. She has additionally been a good example of a strong, caring woman that is profoundly dedicated to both family and career. Dr. Hughes has always been a motivation for me and I wish to end up somebody like her.

I would like to thank my committee members, Dr. Curtis Dyreson and Dr. Tung Nguyen, for their support through the master's program. I also want to thank all the Emergency Responders who agreed to test my application. These test sessions were helpful in making my tool more user-friendly.

I would like to extend my thanks to my family, for all their support in everything I do. Last but not least, I want to thank my sister, Sowjanya for her support throughout my masters program. Her support made this work possible.

Sravani Mamidi

TABLE OF CONTENTS

ABSTRACT.....	iii
DEDICATION.....	iv
ACKNOWLEDGMENT.....	v
LIST OF FIGURES	viii
CHAPTER 1: INTRODUCTION.....	1
1.1 Research Question	2
1.2 Research Design	2
1.3 Project Overview	2
CHAPTER 2: CRISIS SOCIAL MEDIA TRAINER.....	3
2.1 Proof-of-Concept Prototype.....	3
2.2 Architecture & Implementation.....	3
2.2.1. Getting Started	5
2.2.2. Project Layout.....	6
2.2.3. Project Development and Customization.....	6
2.2.4. Prototype Description	7
2.3 User Interface Prototyping and Design.....	9
2.4 Final Prototype.....	10
2.5 Summary.....	13
CHAPTER 3: USER INTERFACE TESTING	15
3.1 Cognitive Walkthroughs	15
3.2 Heuristic Analysis.....	16
3.3 User Testing.....	17
3.3.1. Participants.....	18
3.3.2. Procedure	18
3.3.3. Results.....	19
3.4 Usability Improvements.....	19
3.5 Summary	20
CHAPTER 4: CONCLUSION	21
4.1 Future Work.....	21
4.2 Conclusion.....	22
REFERENCES	23
APPENDICES	24

APPENDIX A: Usability Test Task List	25
APPENDIX B: Post Usability Test Questions.....	27
APPENDIX C: Cognitive Walkthrough Results	28
APPENDIX D: Heuristic Analysis Results	31
APPENDIX E: Usability Test Results	36
APPENDIX F: Post Usability Test Question Responses.....	38

LIST OF FIGURES

Figure 1: File Hierarchy of Project	6
Figure 2: CSV Import Page.....	7
Figure 3: send_tweet Method in tweet_controller	8
Figure 4: Tweet Display Page.....	9
Figure 5: CSMT Welcome Page	10
Figure 6: Login Page of CSMT	11
Figure 7: Welcome Page for Emergency Responder	11
Figure 8 : Page to Upload a Dataset.....	12
Figure 9: Tweet Display Page.....	13

CHAPTER 1: INTRODUCTION

In this project report, I describe the design, implementation, and evaluation of an application called the Crisis Social Media Trainer (CSMT). The CSMT application helps emergency responders practice using social media by simulating the social media activity that one might encounter during a crisis event. The goal of this application is to help emergency responders understand what kinds of information they might see during an event and allow them to practice with the data in a non-emergency setting, rather than first trying to use social media under the chaotic circumstances of a real event.

During times of crisis, members of the public use online media (e.g., Twitter, Facebook, Wikipedia, blogs, etc.) to share photos, spread emergency related information, and update loved ones about their well-being [1], [2]. In turn, emergency responders have found the information that the public provides over online media can be helpful for their response efforts [3], [4]. For example, by monitoring social media activity, responders can better understand what questions and problems members of the public have. They can also find false rumors and misinformation that might be circulating in the public domain. Once responders have this information, they can provide better information that address the public's questions and concerns as well as correct false rumor and misinformation.

The public is coming to expect that emergency responders will share crisis information over social media [5] and that responders will also monitor public social media activity and incorporate the information they find back into emergency response efforts [1], [3]. As a result, many emergency managers have extended their public information responsibilities to include social media. However, it can be challenging for

emergency responders to stay ahead of the rapid advances in social media. Further, there are poor solutions for helping emergency responders practice using social media outside of an actual emergency event. In this project, I seek to design an application that will help emergency responders practice using realistic crisis social media data in a non-emergency setting.

1.1 Research Question

The research question for this project is the following: *How can we design, build, and test an application that helps emergency responders practice working with social media data from a crisis event?*

1.2 Research Design

To address my research question, I take a human-centered approach, one that involves understanding emergency responder needs, building an application based on those needs, and then evaluating the resulting application with emergency responders.

1.3 Project Overview

I describe this project more fully in the following three chapters. Chapter 2 outlines the design, prototyping, and implementation of CSMT. Chapter 3 describes how I evaluated CSMT with expert interface inspection methods and a formal user study. Also in Chapter 3, I report the results of these evaluation activities and describe changes that I made to the application to address issues found in testing. Finally, Chapter 4 concludes with a summary of the project and recommendations for future work.

CHAPTER 2: CRISIS SOCIAL MEDIA TRAINER

In this chapter, I describe the design and implementation of CSMT. I begin by describing my initial, proof-of-concept prototype, which includes a description of the development environment and the tools I used to develop CSMT. Next, I outline the design process, which started with the creation of three paper prototypes of the user interface. Based on feedback, the most promising prototypes were combined to create a final prototype. I conclude this chapter with a description of the final CSMT prototype and its functionality.

2.1 Proof-of-Concept Prototype

During the summer of 2014, I developed the first functional prototype of CSMT. I chose to implement CSMT as a web application, so that emergency responders could access the application from anywhere so long as they have a web browser. To limit the scope of the project, the application only supports Twitter data at this time. The application begins by importing Twitter data (tweets) into a database. Next, the application retrieves the tweets from the database server and displays them one at a time in chronological order. As a new tweet is retrieved from the server, the older tweets slide down and the new tweet appears at the top of the list. This behavior mimics the behavior of other applications that monitor live Twitter data, thus giving the user a more realistic social media experience. The purpose of this prototype was to prove that Twitter data could be imported and displayed in a realistic way using a web application framework.

2.2 Architecture & Implementation

CSMT is built on Ruby on Rails, which is an open source web application framework built upon the Ruby programming language. It uses the MODEL-VIEW-CONTROLLER (MVC) software architectural pattern for implementing user interfaces and the Active Record Pattern for storing data in a database. To understand how Ruby on Rails works, one must understand the MVC pattern. Each layer—model, view, and controller—has a specific responsibility.

- *Model* – A Model represents knowledge. The business logic specific to the application is encapsulated here. In Rails, database-backed model classes are derived from the ActiveRecord::Base. This Active Record allows the data from a database to be represented as objects. For example, in this application a tweet is represented by a model.
- *View* – A View is a visual representation of a Model. To create this visual representation, a view is connected to the corresponding Model so it can retrieve the necessary data for presentation. The View layer is composed of “templates.” These templates are ERB files, which are written in HTML and embedded Ruby code (ERB). Views are usually rendered by an action method found in a Controller.
- *Controller* – A Controller provides a link between a user and the system. It decides what the user input was, how to change the Model based on the input, and which View should be used. It is responsible for handling incoming HTTP requests and providing a suitable response by returning HTML. Controllers load and manipulate models, and render view templates to generate the appropriate HTTP response.

Additionally, to make tweets scroll down the website, AJAX is used with JavaScript to make server calls. AJAX sends and retrieves data from the server asynchronously.

2.2.1. Getting Started

I started programming the application by installing Rails. To run the application I had to start the web server on my machine and navigate to `http://localhost:3000`. I then modified the routes to point Rails to where the “`index.html.erb`” page (or the welcome page) was located for the CSMT.

Next, I created a controller for the tweet model. Inside this class, I defined different actions for this controller that retrieve the tweets contained in the database. Now, when we navigate to `http://localhost:3000/tweets` we can see all the tweets sorted in ascending order with respect to time.

The next step was to generate the tweet model. This model can read and write a tweet to the database as well as check to see if the tweet already exists in the database. Since I had already created a tweets table with the appropriate table columns, Rails auto-generated the tweet model based on this table.

2.2.2. Project Layout

Figure 1 shows the file hierarchy of the CSMT application. The dataset directory has many auto-generated files. Most of the files I changed are in the “app” folder. This directory contains the controllers (application_controller.rb, tweets_controller.rb,

welcome_controller.rb), models (tweet.rb) and views

(views/layoutapplication.html.erb and the files in the

view/tweets directory). The “config” directory contains

files that configure the application’s basic settings like its

routes and its database. Lastly, the “db” directory contains

the current database schema, as well as database

migrations.

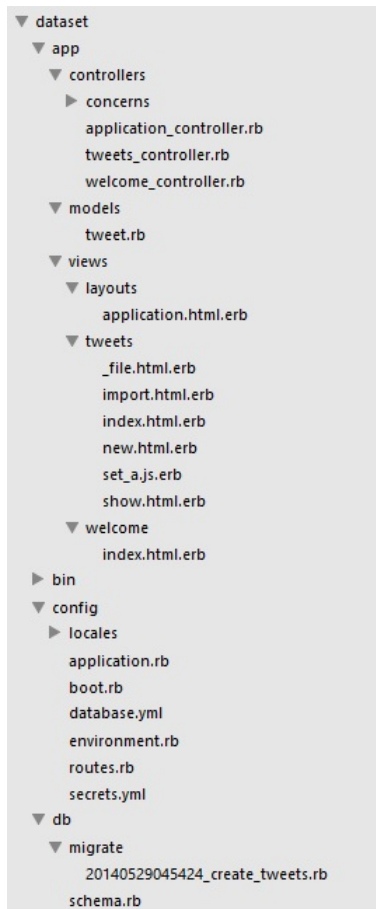


Figure 1: File Hierarchy of Project

2.2.3. Project Development and Customization

After completing the initial project setup and auto-

generation of files, it was time to start modifying and

customizing the program. I began by altering the Tweets

Index View to allow users to upload a CSV file

containing the tweets to display. This also required

changes to the tweet model and controller to accept CSV

files, process them, and put the data in the database.

Next, I modified the Tweets Index View to display the tweets one by one, scrolling down the page. This required use of AJAX to make asynchronous calls back to the server to retrieve the tweets one by one. I describe this proof-of-concept prototype in more detail below.

2.2.4. *Prototype Description*

The first feature I implemented in this prototype was the ability to import a CSV file that contains a list of crisis tweets.



Figure 2 shows a screenshot of the corresponding CSMT page. In this case, there are three different buttons to import CSV files: one to upload tweets associated with a fire event, one for an earthquake event, and one for a flooding event. There is also a link, labeled “Back,” that takes a user back to the welcome page for the application. Once I could successfully import CSV files, I moved on to displaying the tweets.

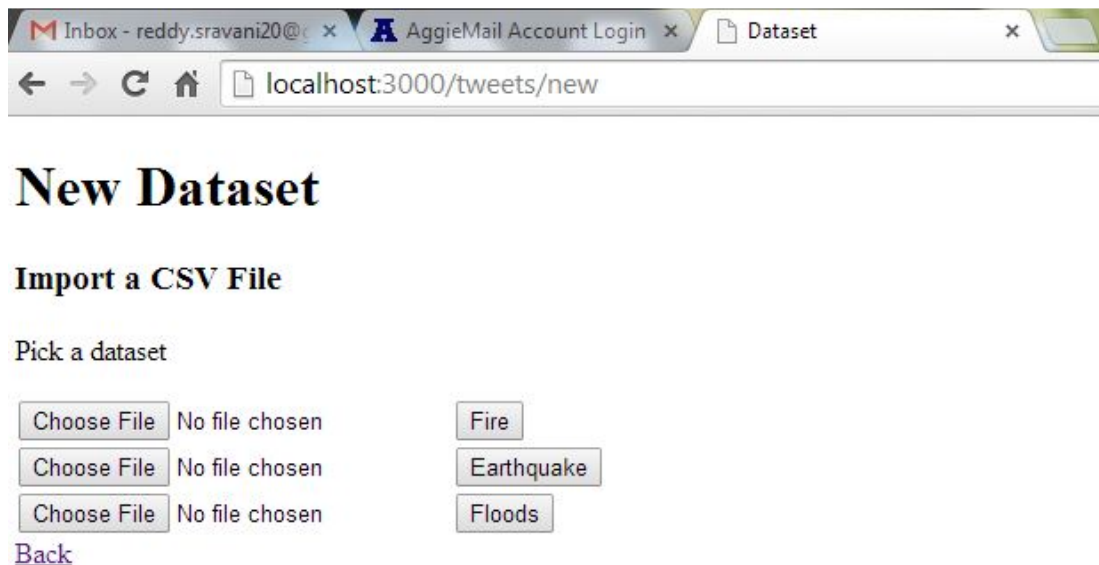


Figure 2: CSV Import Page

I wanted to display tweets in a slide down manner, where the newest tweet appears at the top while the older tweets slide down the web page. This type of display is a common way of displaying new tweets in other Twitter search applications. I initially tried to create this behavior solely in the VIEW page using JavaScript, JQuery, Html, and CSS, however, I was unable to dynamically display the tweets using this approach.

After some investigation, the approach I used to scroll tweets makes dynamic calls to the server to get each tweet. Repeated calls to the “display_tweets” action in the tweets_controller allowed the application to retrieve each tweet from the server. The calls to the server were made asynchronously using AJAX. Once the call returned with a tweet, this new tweet displays at the top of the page, while the older tweets slide down the page.

```

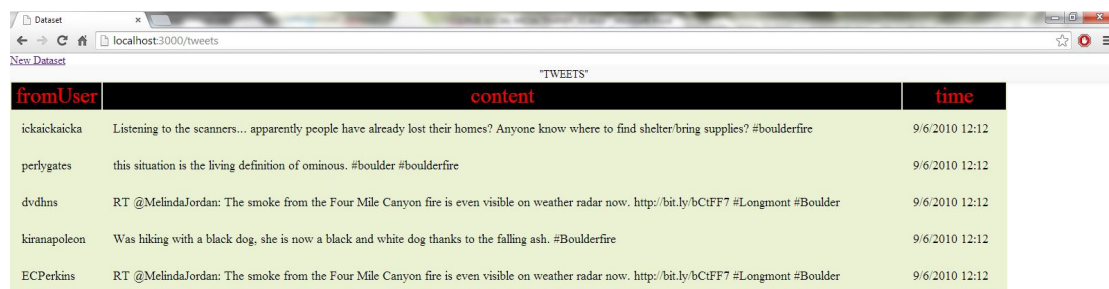
def send_tweet
  @twee = Tweet.all
  @oldesttweet = @twee.last
  render layout: false, json: @twee[$counter]
  $counter = $counter + 1
end

```

Figure 3: send_tweet Method in tweet_controller

Figure 3 contains the code for the “send_tweet” action in the tweet_controller. This action is responsible for sending the appropriate tweet back to the user interface. Inside the tweet_controller, there is a global variable “counter” which acts as an index, telling the send_tweet method which tweet to retrieve. The send_tweet method returns a tweet model object. This object contains values for the 4 tweet attributes (“id,” “fromUser,” “content,” and “time”). In the view for the tweet displaying page, an AJAX call is made to the server with the type set as “get” and the URL “/tweets/send_tweet.” Once a tweet object is returned from the AJAX call, it passes the response to the Javascript function “display_tweet()” in the page view. This function displays the tweets. The tweets begin to display when a user clicks the “Start” link on the welcome page.

The page that displays tweets in the final proof-of-concept prototype can be seen in Figure 4 below. As a new tweet is retrieved from the server, the older tweets slide down and the new tweet appears at the top of the list of tweets.



fromUser	content	time
ickaickaicka	Listening to the scanners... apparently people have already lost their homes? Anyone know where to find shelter/bring supplies? #boulderfire	9/6/2010 12:12
perlygates	this situation is the living definition of ominous. #boulder #boulderfire	9/6/2010 12:12
dvdhns	RT @MelindaJordan: The smoke from the Four Mile Canyon fire is even visible on weather radar now. http://bit.ly/bCtFF7 #Longmont #Boulder	9/6/2010 12:12
kiranapoleon	Was hiking with a black dog, she is now a black and white dog thanks to the falling ash. #Boulderfire	9/6/2010 12:12
ECPerkins	RT @MelindaJordan: The smoke from the Four Mile Canyon fire is even visible on weather radar now. http://bit.ly/bCtFF7 #Longmont #Boulder	9/6/2010 12:12

Figure 4: Tweet Display Page

2.3 User Interface Prototyping and Design

After completing the proof-of-concept prototype, I was ready to further improve the application and its usability. In this section, I describe how I designed and prototyped the user interface of CSMT.

I began the design process by creating three paper prototypes (see Appendix G). Each prototype explored different user interface ideas. These three prototypes were then evaluated by an experienced user interface designer—my major professor. I took the three paper prototypes to her and demonstrated the functionality of each prototype, after which I requested her feedback. The main goal was to find which prototype proved the most useful for further development. The areas in need of improvement include the following:

- The interface needs better descriptions of the application and its purpose, so that users understand how to use it.
- There should be a separate screen for uploading data sets.
- The data sets should have a name and a description field.
- Only one button is needed to “start” and to “pause” the streaming display of tweets.
- The tweet entries need to look less like a table entry and more like the way that they are displayed using Twitter’s search feature.
- An authenticated user should have a different view than a user who is not authenticated.
- The user should have an option to restart the display of tweets.

Based on this feedback, I implemented a new functional prototype of CSMT.

2.4 Final Prototype

In this section, I describe the final CSMT prototype and its functionality. When users first load the tool in a web browser by navigating to the welcome page they can perform multiple tasks. They can login to become an authenticated user or they can view the Twitter data of an event. Figure 5 shows the new welcome page of the application.

Crisis Social Media Trainer

Crisis Social Media Trainer is a tool that helps emergency responders practice monitoring social media in a non-emergency setting. The goal of this tool is to help emergency responders understand the types of information they might encounter over social media during an emergency event.

Name	Description	View
Fire	Uncontrolled blazes fueled by weather, wind, and dry underbrush, wildfires can burn acres of land and consume everything in their path in mere minutes.	View Data
floods	an overflowing of a large amount of water beyond its normal confines, especially over what is normally dry land.	View Data
Hurricane	a storm with a violent wind, in particular a tropical cyclone in the Caribbean.	View Data
disaster	a sudden event, such as an accident or a natural catastrophe, that causes great damage or loss of life	View Data
windstorm	a storm with very strong wind but little or no rain or snow; a gale.	View Data
Cyclone	In meteorology, a cyclone is an area of closed, circular fluid motion rotating in the same direction as the Earth.	View Data

Figure 5: CSMT Welcome Page

The application does not allow a user to upload a dataset if he/she has not logged in using a valid account. They need to login first with a username and password (See Figure 6).

Figure 6: Login Page of CSMT

After successfully logging in, an upload button appears at the top of the welcome page as shown in Figure 7. A user can sign-up to become a valid user by providing a user name, an email, and a password. The signup option is provided at the bottom of the login page.

Crisis Social Media Trainer is a tool that helps emergency responders practice monitoring social media in a non-emergency setting. The goal of this tool is to help emergency responders understand the types of information they might encounter over social media during an emergency event.

Logged in as sravani@gmail.com

Name	Description	View
Fire	Uncontrolled blazes fueled by weather, wind, and dry underbrush, wildfires can burn acres of land and consume everything in their path in mere minutes.	View Data
floods	an overflowing of a large amount of water beyond its normal confines, especially over what is normally dry land.	View Data
Hurricane	a storm with a violent wind, in particular a tropical cyclone in the Caribbean.	View Data
disaster	a sudden event, such as an accident or a natural catastrophe, that causes great damage or loss of life	View Data
windstorm	a storm with very strong wind but little or no rain or snow; a gale.	View Data
Cyclone	In meteorology, a cyclone is an area of closed, circular fluid motion rotating in the same direction as the Earth.	View Data

Figure 7: Welcome Page for Emergency Responder

After logging in, a user can upload a custom file that contains emergency social media messages. To work with CSMT, the CSV file for upload must be formatted in a

particular way. The csv file should have the following fields “Twitter Text,” ”Created,” ”fromuser,” ”name,” and ”profileimg.” The “profileimg” field should contain a URL to a valid image otherwise CSMT will display a message that says “image not found.”

Each time a user uploads a dataset, he/she needs to provide a name and description for that data set (See Figure 8). Once this information has been provided, the user clicks the “upload” button. CSMT will upload and create a new dataset that will then appear on the welcome page as a new entry.

The upload page also contains a description of the page’s purpose. Within that description, a user can click on the “Click Here” link to see more details about the format of the custom csv file.

Figure 8 : Page to Upload a Dataset

Once an emergency responder has identified a dataset that they would like to view, they can select the “View Data” option for that event (see Figure 7). This option will take the user to the Tweet Display Page (see Figure 9) where s/he can view the emergency event tweets that scroll down the page in the order they were sent. New tweets show up at the top and push the already existing tweets down the list—creating a

scrolling effect. Each tweet shows the user name, text message, profile image and a timestamp when the tweet was sent (See Figure 9).

Sometimes, an emergency responder may find that the tweet stream moves too rapidly. Consequently, I implemented a feature that lets users pause and resume the flow of tweets. Additionally, the user can restart the flow of tweets by clicking the restart button provided at the top of the page. Also, a user can Logout or navigate to the home page if he/she is done viewing messages.



Figure 9: Tweet Display Page

2.5 Summary

This chapter describes the design, and implementation of a functional CSMT prototype. To create this prototype I engaged in several iterations of prototyping, evaluation, and refinement. I conclude the chapter with a description of the final prototype. In the next chapter, I evaluate and improve this prototype based on feedback from user interface inspection methods and a user study.

CHAPTER 3: USER INTERFACE TESTING

In this chapter, I describe the different types of interface testing I conducted with

CSMT. The testing process began with two expert-inspection methods for evaluating the user interface; namely, I performed cognitive walkthroughs and heuristic analyses. After these methods, I conducted user testing of CSMT with emergency responders. I describe these testing methods and their results below and conclude with a description of the changes made to the application as a result of this testing.

3.1 Cognitive Walkthroughs

After developing a functional prototype, I performed a number of cognitive walkthroughs to uncover usability issues. The cognitive walkthrough method [6] can identify usability issues in interactive systems. A cognitive walkthrough consists of a list of tasks that a user must perform to use a system. For each of these tasks, the person evaluating the interface will consider the following questions:

- Are the actions that the user should take obvious?
- Does the application change states to indicate the status of the application?
- Does the interface offer good feedback for this task?
- Will the user know what to try and do and how to do it?
- Are there any wrong actions that may look correct?

The answers to these questions form the basis of this type of evaluation and help interface designers understand the difficulties that users might encounter when using the system.

I performed cognitive walkthroughs on the two main tasks for using the CSMT application—uploading a data set and viewing the tweets in a dataset. Each of these tasks were divided into smaller subtasks and evaluated using the cognitive walkthrough method. The testing sheets I used to record the results of these walkthroughs can be found in Appendix C. I list the most important issues found using

this method below:

- On the “View” page a few of the buttons were placed at the bottom making it difficult and more time consuming for users to access them.
- When the “View” page updates with a large number of tweets, the buttons (“Home” and “Restart”) were not visible, thus confusing the users.
- On the “Upload” page, the users can upload a valid file in csv format. Data file formats might be confusing for users and may result in invalid data if a file in the wrong format is uploaded.

3.2 Heuristic Analysis

Next, I performed a heuristic analysis of CSMT’s user interface. A heuristic analysis [6] identifies usability problems through the evaluation of 10 heuristics for good interface design (see Table 1). This type of analysis is performed by visiting each interface page and considering whether each of the 10 heuristics have (or have not) been met. Heuristic analyses are good for the general analysis of a user interface and they typically reveal different types of issues than cognitive walkthroughs do because they are not task based.

10 Heuristics for Good User Interface Design
1. Visibility of system status.
2. Match between system and the real world
3. User control and freedom
4. Consistency
5. Error prevention
6. Recognition rather than recall
7. Flexibility and efficiency of use
8. Aesthetic and minimalist design
9. Help users recognize diagnose and recover from errors
10. Help and documentation

Table 1: 10 Heuristics for Good User Interface Design

The heuristic analysis provided quick feedback and resulted in a list of usability issues. It helped me discover design issues and determine the usability of the application. Detailed results of this analysis can be found in Appendix D. Below, I highlight the most important findings from this analysis:

- On the “Login screen” page, if a user did not enter valid credentials, no error message appeared. In addition, anyone could access the system without being an authenticated user.
- The buttons in the application lacked consistency because they were placed haphazardly on the page, sometimes at the bottom of the page, sometimes at the top. This button-placement strategy made it difficult, at times, for users to know that a button existed.

Following completion of the cognitive walkthroughs and the heuristic analyses, I made a few improvements to the existing prototype to ready the application for user testing.

3.3 User Testing

To ensure that CSMT is helpful for emergency responders, I tested it with them. This testing not only addressed the functionality of CSMT, but also investigated whether the idea behind this application is useful to them. Between test sessions, I iteratively made improvements to the tool based on test results and suggestions made by emergency responders. I summarize the user testing process below (a more detailed description can be found in Appendix E).

3.3.1. Participants

I tested this tool with four emergency responders from the Logan, Utah area. One participant is a local emergency manager with many years of experience in a variety of emergency situations. The other three participants are from a local Citizen Emergency Response Team (CERT). These three do not have as much experience, but they have been trained in emergency response procedures and it is likely that they may be asked to help monitor social media during future emergency events. All four participants completed the testing individually.

3.3.2. Procedure

Test sessions were held at each participant's place of work and lasted approximately 30 minutes. I started every session by asking each participant about their background so that I could better understand his/her experience as an emergency responder. After talking about their experience, I demonstrated the format and purpose of the test session and asked them if they had any questions.

Next, I briefly demonstrated the features of the tool and asked the participants to complete a list of tasks designed to test the functionality CSMT (see Appendix A). Participants took approximately one minute to perform each task. While performing these tasks, I asked the participants to "think aloud"—that is to verbalize their thoughts so I could understand what they were thinking. I took notes around my observations and how the participants interacted with CSMT during this time.

After completing the tasks, I asked each participant a few follow up questions about their experience with the application (see Appendix B). Finally, I thanked the participants for their help and gave them a chance to ask me questions.

3.3.3. *Results*

The user testing revealed insights into how CSMT might be used by real emergency responders. The detailed notes from the user testing sessions can be found in Appendix E and Appendix F. Overall, the application was well received by the participants and they thought the application would be useful for social media training exercises. However, the application has room for improvement. Some of the problems highlighted by user testing are the following:

- Several test participants felt that the “Login” button was too small.
- Participants had difficulty understanding the “dataset upload” option, and they didn’t understand what the .csv file format was.
- The function of the “restart” button was confusing. Participants did not understand the icon used to represent the “restart” button.
- The naming convention for uploading a dataset also caused problems. For example, there are two fields where CSMT asks a user to provide a name and description for the dataset they upload. Test participants were confused about the purpose of those fields.

3.4 Usability Improvements

In response to the usability issues found in the cognitive walkthroughs, heuristic analyses, and user testing, I made improvements to the CSMT application. Below, I list the changes I made to the system:

- The application was modified to only allow users to access authenticated content if they have valid login credentials.
- The “upload” page was modified to include additional information regarding

the naming convention of the dataset. I also included an extra link with information to help users better understand the file format for datasets.

- The “Login” button has been reduced in size.
- Button consistency, which was reported as a drawback to the application, has now been modified and all the buttons are placed at the top of the respective pages.
- The “restart” button has been changed to a new one where the text “Restart” has been written on the button. Now, the functionality of the button is more easily understood by users.
- Tool tips have been provided on every button of the application, which helps users understand the purpose of each button.
- I have also added better descriptions to each page so that users new to the application can navigate the application easier.

3.5 Summary

In this chapter, I outlined the steps I took to evaluate and improve the usability of the CSMT application. I began by conducting cognitive walkthroughs and heuristic analyses on the application prototype. Next, I conducted a user study. Based on the results of these evaluation methods, I made usability improvements to the CSMT application prototype.

CHAPTER 4: CONCLUSION

In this project report, I describe the design, development, and evaluation of an application that helps emergency responders practice using social media—CSMT. The goal of this application is to help emergency responders understand the different kinds of information they might see during an event and allow them to practice with the data in a non-emergency setting.

To begin my efforts on this project, I created a proof-of-concept prototype. This prototype allowed me to test the feasibility of CSMT and helped me to think through the best way to implement CSMT's features. After completing this initial prototype, I iteratively designed the user interface for the application using paper prototypes. These interface prototypes were then combined with the proof-of-concept prototype to make a functional prototype that could be tested with real users.

The next step in this project was to evaluate the CSMT prototype. First, I performed cognitive walkthroughs and heuristic analyses on the prototype. After making improvements based on the results of the walkthroughs and analyses, I tested the prototype with four emergency responders. This user testing revealed many areas of improvement for the application and demonstrated that CSMT could be a useful social media training tool for emergency responders. Finally, I modified the CSMT prototype to address the issues encountered by the users during testing.

4.1 Future Work

For any software tool there are always new features and improvements that can be added. A natural extension of this work would be to make CSMT publically available on

a web server for anybody to use. In addition, I discovered several features that would enhance the existing project and improve its ability to help emergency responders gain experience with the types of social media they might encounter during a crisis event. The emergency responders who tested the application also had ideas for improving CSMT's user experience and for additional features that could improve and expand the usefulness of CSMT. These future feature improvements include the following:

- Add support for other types of social media, such as Facebook.
- Provide capability to add a comment to a tweet.
- Add ways to filter the tweets based on various criteria.
- The aesthetics of the tool could use improvement. The focus of the user testing and prototyping for this project was on the functionality of the tool not on the attractiveness of the application.
- The tool was not intended to monitor tweets in real-time, but several users thought the tool would be more useful if it could.
- Several test participants wanted to see live links in the tweets, so they could click on the links and follow them for more information.

4.2 Conclusion

In this project report, I describe the design, implementation, and evaluation of an application that will help emergency responders practice monitoring the type of social media activity that occurs during an emergency event. Initial user testing shows that the application is useable and fits a real need for emergency responders.

REFERENCES

- [1] L. Palen and S. B. Liu, "Citizen Communications in Crisis: Anticipating a Future of ICT-supported Public Participation," in *Proceedings of the 2007 Conference on Human Factors in Computing Systems (CHI 2007)*, New York, NY, 2007, pp. 727–736.
- [2] A. L. Hughes, L. Palen, J. Sutton, S. B. Liu, and S. Vieweg, "'Site-Seeing' in Disaster: An Examination of On-Line Social Convergence," in *Proceedings of the Information Systems for Crisis Response and Management Conference (ISCRAM 2008)*, Washington D.C., 2008.
- [3] A. L. Hughes and L. Palen, "The Evolving Role of the Public Information Officer: An Examination of Social Media in Emergency Management," *J. Homel. Secur. Emerg. Manag.*, vol. 9, no. 1, 2012.
- [4] M. Latonero and I. Shklovski, "Emergency Management, Twitter, and Social Media Evangelism," *Int. J. Inf. Syst. Crisis Response Manag.*, vol. 3, no. 4, pp. 1–16, 2011.
- [5] American Red Cross, "Social Media in Disasters and Emergencies," 2011.
- [6] C. Lewis and J. Rieman, *Task-Centered User Interface Design*. University of Colorado, Boulder, Department of Computer Science, 1993.

APPENDICES

APPENDIX A: Usability Test Task List

- Task 1:
Register as a user, providing an email address and password.
- Task 2:
Login with your email address and password.
- Task 3:
Choose to upload a dataset.
- Task 4:
Browse and choose a csv file (dataset) with required format.
- Task 5:
Enter a name and description for the dataset selected.
- Task 6:
Upload dataset.
- Task 7:
Select a dataset to view the messages in it.
- Task 8:
Pause the message flow.
- Task 9:
Read messages on screen and comment on the usefulness of the messages for emergency response.

- Task 10:
Resume the message flow.
- Task 11:
Restart the flow of messages from the beginning.
- Task 12:
Return to the homepage
- Task 13:
Logout

APPENDIX B: Post Usability Test Questions

- How comfortable do you feel using this application?
- What do you think about the appearance of the application?
- Do you think this application could be useful for emergency responders? If yes why?
- Do you have any ideas for additional application features?

APPENDIX C: Cognitive Walkthrough Results

Task: View page for Crisis Social Media Trainer

Instructions: write the Steps. Checkbox when things work. Note only when there are problem spots and why

1.Steps of Task (These are micro)	2. User Action Obvious	3. Change in States	4.Good Feedback	Critique: Will the user know: What to try to do? How to do it? Are there any wrong actions that may look correct
Go to specified address.	Obvious.	Goes to home page.	Yes.	He/she needs to know the address (URL).
Emergency responder can login and see data or can directly view data.	Can select any task present on screen.	Goes to either view page or login page.	Yes.	There are different tasks present on the screen to select .The user may select login, view data or help.
Click on login button provided.	Obvious.	Navigate to login page.	Yes.	The user should login to upload a dataset. Yes user know how and what to do it.
Enter user details if he/she is an existing user.	If existing user obvious.	Redirect to home page with more options.	Yes.	User should have valid username and password to login.
If new user click on "signup".	Obvious.	Goes to signup page.	Yes.	Needs to signup to become a valid user. Yes, user know how and what to do it.
Enter details and click on sign up.	Obvious	After signup goes to homepage with more options.	Yes	Should enter valid details about user.
Select view data for corresponding dataset.	Can select any dataset.	Selected dataset is displayed.	Yes	The user should know the dataset for which he/she is trying to view it.

1.Steps of Task (These are micro)	2. User Action Obvious	3. Change in States	4.Good Feedback	Critique: Will the user know: What to try to do? How to do it? Are there any wrong actions that may look correct
Start/Pause flow of tweets.	Obvious if user wants to pause it.	Will pause till user clicks start button and will play until user clicks pause button.	Yes.	Yes the user know how and what to do it. Since he/she just have to click the button.
Restart tweets	Obvious if he/she wants to restart tweets.	Tweets displayed from very first one.	Yes.	Button at the bottom of page, have to scroll down to see it.
Go-back to home page	Obvious	Goes to homepage	Yes.	Yes user know how and what to do it. But the home button is not visible since it is at the bottom of the page.
Logout	Obvious. Logout once user is done.	Goes to homepage with Login button	Yes.	Yes the user know what and how to do it.

Task: Upload page for Crisis Social Media Trainer

Instructions: write the Steps. Checkbox when things work. Note only when there are problem spots and why

1.Steps of Task (These are micro)	2. User Action Obvious	3. Change in States	4.Good Feedback	Critique: Will the user know: What to try to do? How to do it? Are there any wrong actions that may look correct
Go to specified address.	Obvious.	Goes to home page.	Yes.	He/she needs to know the address (URL).
Login.	Obvious. cannot see upload button until logged in.	Goes to login page.	Yes.	The user must login to upload a dataset.
Enter user details if existing user or signup.	Obvious.	Goes to home page with upload button	Yes.	The user should have valid username and password.
Click on upload button.	Obvious.	Goes to upload page.	Yes.	User should have a valid file (.csv) to upload. Yes, user know how and what to do it.
If new user click on "signup".	Obvious.	Goes to signup page.	Yes.	Needs to sign up to become a valid user. Yes, user know how and what to do it (more information is provided).
Upload a dataset.	Obvious if he/she wants to enter data.	Go to homepage with new dataset.	Yes	Yes, user know how and what to do it. Should enter valid name and description for dataset then click upload.
Home page.	Obvious if he/she does not want to upload.	Goes to home page.	Yes	Yes, user know how and what to do it.
Logout	Obvious.	Goes to homepage with login.	Yes	Just have to click Logout button.

APPENDIX D: Heuristic Analysis Results

Description of Interface portion Under Evaluation: Welcome page with Login and help.

Heuristic	Brief Description of the Evaluation Decision.	Severity Rating (1-3) 3= Big Problem
1. Visibility of system status.	Clearly mentioned.	0
2. Match between system and the real world	Everything is clear.	0
3. User control and freedom	Since the user did not login, the view is appropriate	0
4. Consistency	Help and login are at bottom of page	0
5. Error prevention	No chance of getting errors in this page	0
6. Recognition rather than recall	User don't have to remember	0
7. Flexibility and efficiency of use	The page have obvious information about icons or buttons	0
8. Aesthetic and minimalist design	Good	0
9. Help users recognize diagnose and recover from errors	No chance to get any error messages	0
10. Help and documentation	Add help regarding usage of page	1

Description of Interface portion Under Evaluation: Login Screen

Heuristic	Brief Description of the Evaluation Decision.	Severity Rating (1-3) 3= Big Problem
1. Visibility of system status.	Visibility looks good.	0
2. Match between system and the real world	Looks good.	0
3. User control and freedom	fine	0
4. Consistency	Home and view are at top	0
5. Error prevention	Entering wrong information popup nothing	2
6. Recognition rather than recall	No need to recall	0
7. Flexibility and efficiency of use	Can go to home or view page which made it flexible and efficient	0
8. Aesthetic and minimalist design	Sufficient information is present	0
9. Help users recognize diagnose and recover from errors	Should show error messages when user enters wrong username/password	2
10. Help and documentation	No need of help/documentation	1

Description of Interface portion Under Evaluation: Welcome page after Login

Heuristic	Brief Description of the Evaluation Decision.	Severity Rating (1-3) 3= Big Problem
1. Visibility of system status.	Fine	0
2. Match between system and the real world	Everything is clear.	0
3. User control and freedom	Provided with upload for the registered users	0
4. Consistency	Can place help button to maintain standards.	2
5. Error prevention	No chance of getting errors.	0
6. Recognition rather than recall	No	0
7. Flexibility and efficiency of use	Looks fine	0
8. Aesthetic and minimalist design	Good	0
9. Help users recognize diagnose and recover from errors	No chance to get any error	0
10. Help and documentation	can have help about what registered users can do	1

Description of Interface Portion Under Evaluation: Upload Data When Logged

in.

Heuristic	Brief Description of the Evaluation Decision.	Severity Rating (1-3) 3= Big Problem
1. Visibility of system status.	Good	0
2. Match between system and the real world	Looks fine	0
3. User control and freedom	Looks good	0
4. Consistency	Good	0
5. Error prevention	Can add error prevention when user don't select file to upload	1
6. Recognition rather than recall	No	0
7. Flexibility and efficiency of use	Interface is flexible to use	0
8. Aesthetic and minimalist design	Good	0
9. Help users recognize diagnose and recover from errors	Yes	0
10. Help and documentation	Yes can show error message when data file is not selected.	1

Description of Interface portion Under Evaluation: View Data

Heuristic	Brief Description of the Evaluation Decision.	Severity Rating (1-3) 3= Big Problem
11. Visibility of system status.	Good	0
12. Match between system and the real world	Looks fine like twitter display	0
13. User control and freedom	Can play/pause the display	0
14. Consistency	Can maintain consistency by placing buttons in same position	2
15. Error prevention	No chance of getting errors	0
16. Recognition rather than recall	No	0
17. Flexibility and efficiency of use	Can go to any page or can upload	0
18. Aesthetic and minimalist design	Good	0
19. Help users recognize diagnose and recover from errors	No need	0
20. Help and documentation	No need	0

APPENDIX E: Usability Test Results

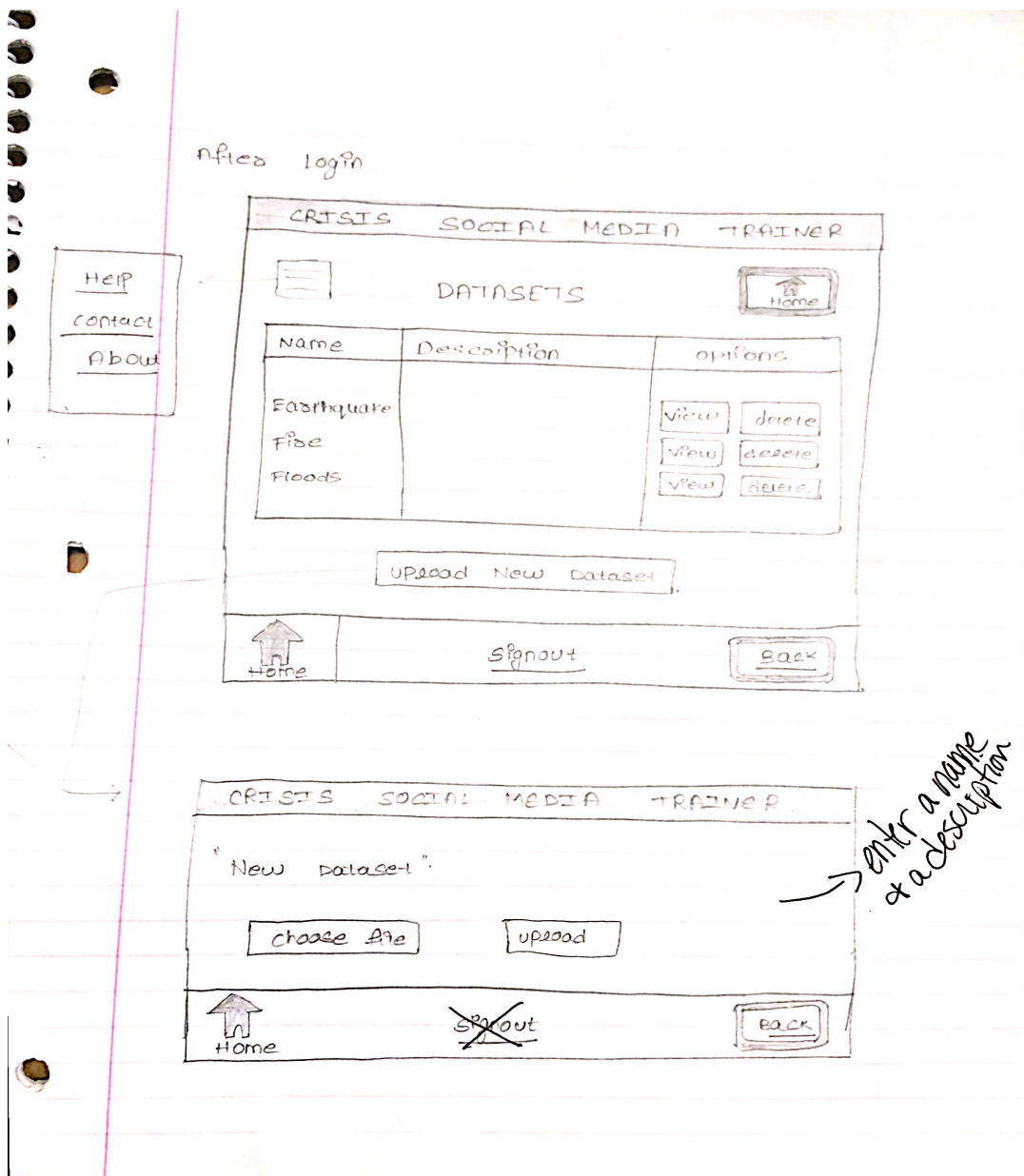
Task	Testing Notes	Improvements
Task 1: Register as a user, providing an email address and password.	<ul style="list-style-type: none"> • Increase size of button. • Page has Login button but task is to register as a new user. 	<ul style="list-style-type: none"> • I have increased the size of login button and information is provided on page to avoid confusion between register and login.
Task 2: Login with your email address and password.	<ul style="list-style-type: none"> • No Comments 	<ul style="list-style-type: none"> • Not Applicable
Task 3: Choose to upload a dataset.	<ul style="list-style-type: none"> • No Comments 	<ul style="list-style-type: none"> • Not Applicable
Task 4: Browse and choose a csv file (dataset) with required format.	<ul style="list-style-type: none"> • Confused about choosing a CSV file. • Did not understand what exactly a dataset should contain and it took a minute to figure out what exactly a CSV file means. 	<ul style="list-style-type: none"> • Emergency Responders did not have any clue about what a CSV file means. So I just made it clear by adding some helpful information on page so that they can just have a look at it and sort out issue with dataset.
Task 5: Enter a name and description for the dataset selected.	<ul style="list-style-type: none"> • What exactly I mean by name and description was their question. • Do they have to give their name and what a description column should have? 	<ul style="list-style-type: none"> • I had to improvise content on page to make it clearer that name means name of dataset that they upload and description has to do something with the content in dataset.
Task 6: Upload dataset.	<ul style="list-style-type: none"> • No Comments 	<ul style="list-style-type: none"> • Not Applicable
Task 7: Select a dataset to view the messages in it.	<ul style="list-style-type: none"> • Font of messages is little small. It would be great if font size is increased 	<ul style="list-style-type: none"> • I had to increase font size of messages to make it readable.
Task 8: Pause the message flow.	<ul style="list-style-type: none"> • No Comments 	<ul style="list-style-type: none"> • Not Applicable
Task 9: Read messages on screen and comment on the usefulness of the messages for emergency response.	<ul style="list-style-type: none"> • Easy to read. • Everything looks good. • It would be more helpful if data provided is current data. 	<ul style="list-style-type: none"> • Added these points to Future improvements.

	<ul style="list-style-type: none"> • Search for hashtags and go to that page when they find some rumors and able to shut down. 	
Task 10: Resume the message flow.	<ul style="list-style-type: none"> • No Comments 	<ul style="list-style-type: none"> • Not Applicable
Task 11: Restart the flow of messages from the beginning.	<ul style="list-style-type: none"> • Could not catch restart button. 	<ul style="list-style-type: none"> • I had to change restart button to make it look better.
Task 12: Return to the homepage.	<ul style="list-style-type: none"> • No Comments 	<ul style="list-style-type: none"> • Not Applicable
Task 13: Logout	<ul style="list-style-type: none"> • No Comments 	<ul style="list-style-type: none"> • Not Applicable

APPENDIX F: Post Usability Test Question Responses

Question	Testing Notes	Improvements
Question 1: How comfortable do you feel using this application?	<ul style="list-style-type: none"> • Little confusing at first. • Once they figure out easy to browse. • The more they use it the more comfortable it would be. 	<ul style="list-style-type: none"> • It was little time taking for every one when they use new tool to figure out what exactly it does. • Tried to make most of the changes so that it becomes more user-friendly.
Question 2: What do you think about the appearance of the application?	<ul style="list-style-type: none"> • Most of them liked the appearance of the tool. • Some of them had issues with size of buttons and font-size. • Some of them wanted it to look little flashier. 	<ul style="list-style-type: none"> • Increased size of buttons and made font size little bigger. • Changed some of the colors to gain user attention.
Question 3: Do you think this application could be useful for emergency responders? If yes why?	<ul style="list-style-type: none"> • Most of them answered "YES". • Getting real time information would be more helpful. • They felt easy to use it and very less time taking. • It is great as monitoring tool. • It works well for tracking response. 	<ul style="list-style-type: none"> • I added Getting real time information in future work.
Question 4: Do you have any ideas for additional application features?	<ul style="list-style-type: none"> • Most of them felt it is fine. • User 4 wanted to highlight the name of crisis that currently happened. • User 1 felt adding twitter hyperlinks and resource hyperlinks would be helpful. 	<ul style="list-style-type: none"> • Hyperlinks can be implemented once I get real time information as messages.

APPENDIX G: Paper Prototypes



This is the new start page
 Put welcome script here describe how to use the site

CRISIS SOCIAL MEDIA TRAINER

DATASETS

Name	Description	Options
Earthquake		<input type="button" value="view"/>
Floods		<input type="button" value="view"/>
fire		<input type="button" value="view"/>

CRISIS SOCIAL MEDIA TRAINER

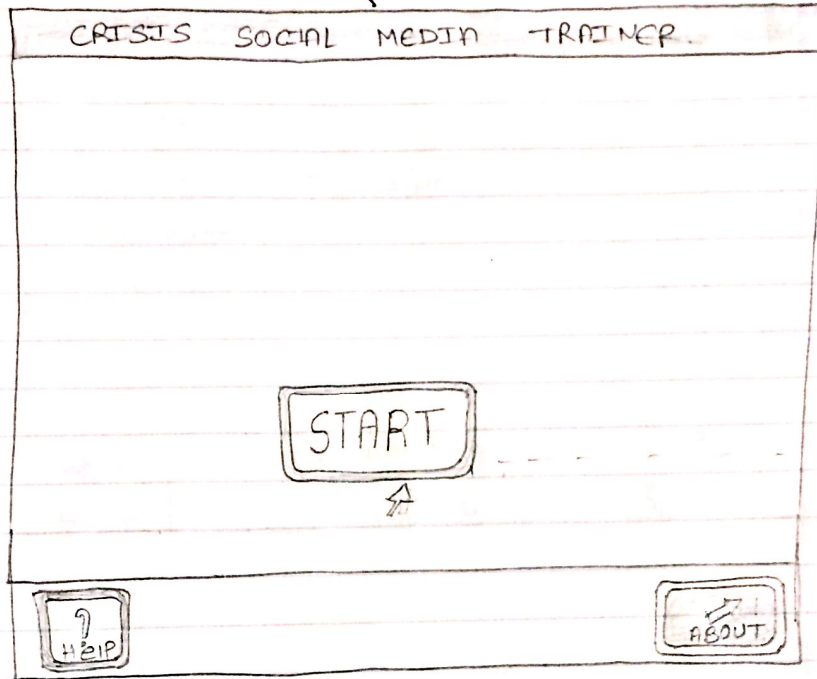
FOR emergency responders only

user name
 password

Help close
 Here we explain how each page works

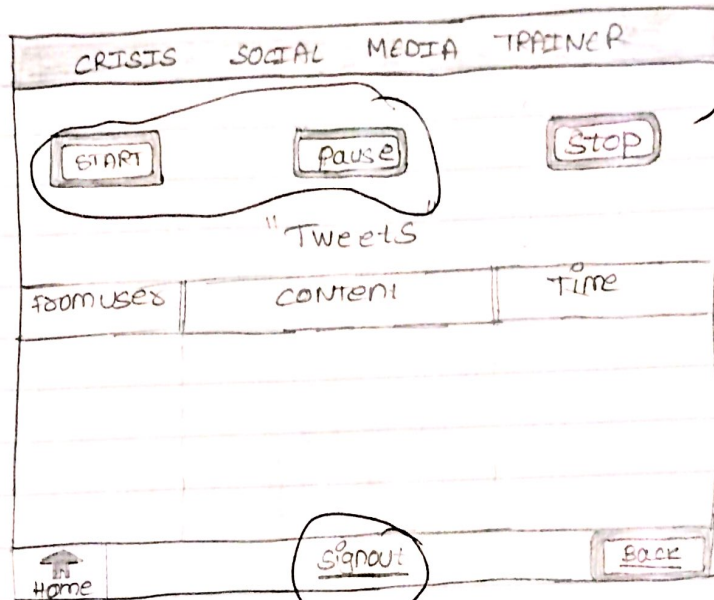
TYPE-3

~~More welcome to next page/screen~~



HELP	close
<p>The ultimate goal of this tool is to help emergency responders understand what kinds of information.</p>	

ABOUT	close
Version	1.2
Privacy policy	→
Terms of use	→
End user license agreement	→



does
make more
sense
at the top

signout

if they are logged in
it says "signout" but
if they are not logged in
it says "Login"

when login is clicked

CRISIS SOCIAL MEDIA TRAINER ✓

For Emergency Responders only

username

password

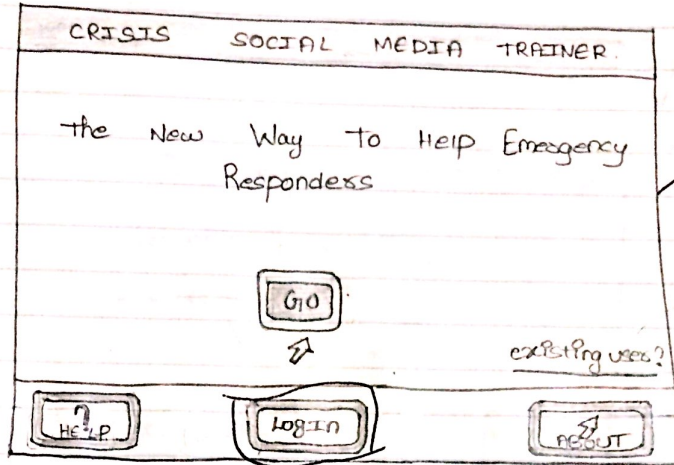
once logged in (if he/she is admin)

CRISIS SOCIAL MEDIA TRAINER

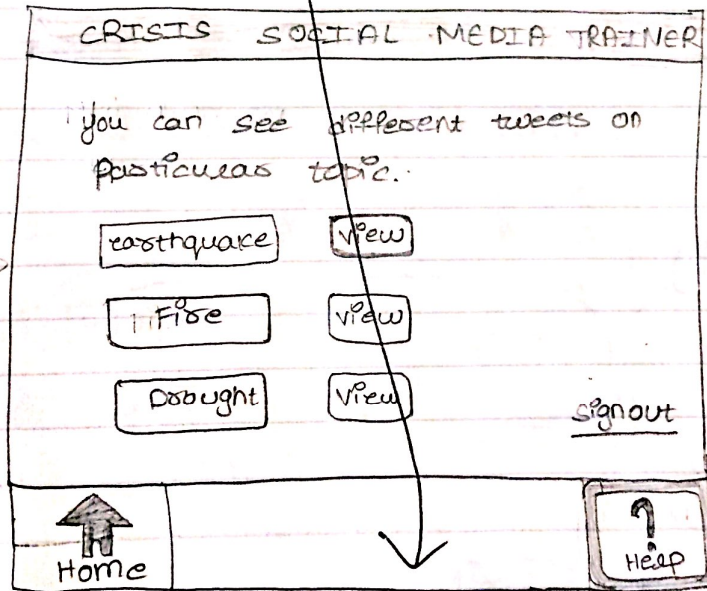
- [Home](#)
- [About](#)
- [View](#)
- [Help](#)
- [Contact](#)

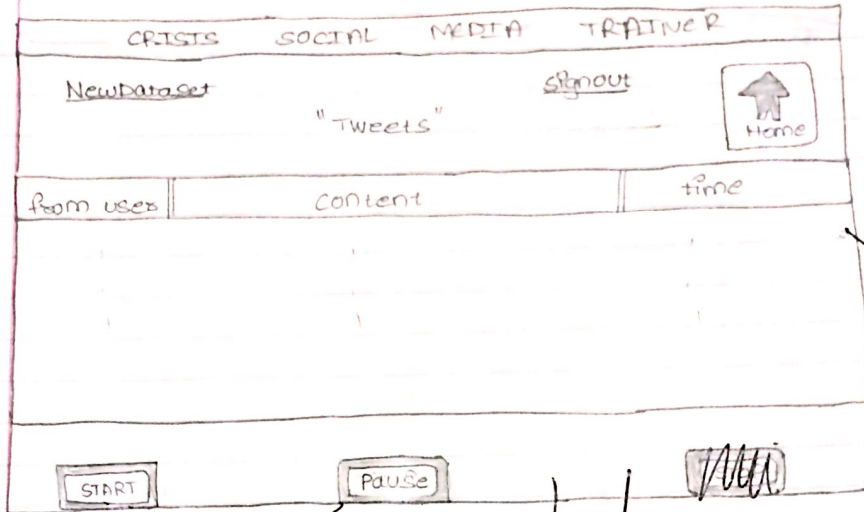
Change to be made we discussed earlier

Type-2



If 'Go' is clicked, then goes to page down have, cannot upload data, can just view it





only need 1 button - text ~~is~~ changes depending on the state

Do we need a way to show the user that it's done?

what do these entries look like? can they look more like the way Twitter displays them

may want a restart option

also a "done" option - maybe home & done are the same?

← prototype
2 views
- normal user
- admin user

Crisis Social Media Trainers

upload data through csv files. can upload different data using different buttons provided.

earthquake	view	upload
drought	view	upload
fire	view	upload

Home ? HELP BACK

select a data set or upload a new one

Admin can delete & upload

Help close

Here we explain how does this work

Back :- this will take back to previous page.

Name	Description	Options
earthquake 1	description	use delete

Different views for Admin + regular user

refresh

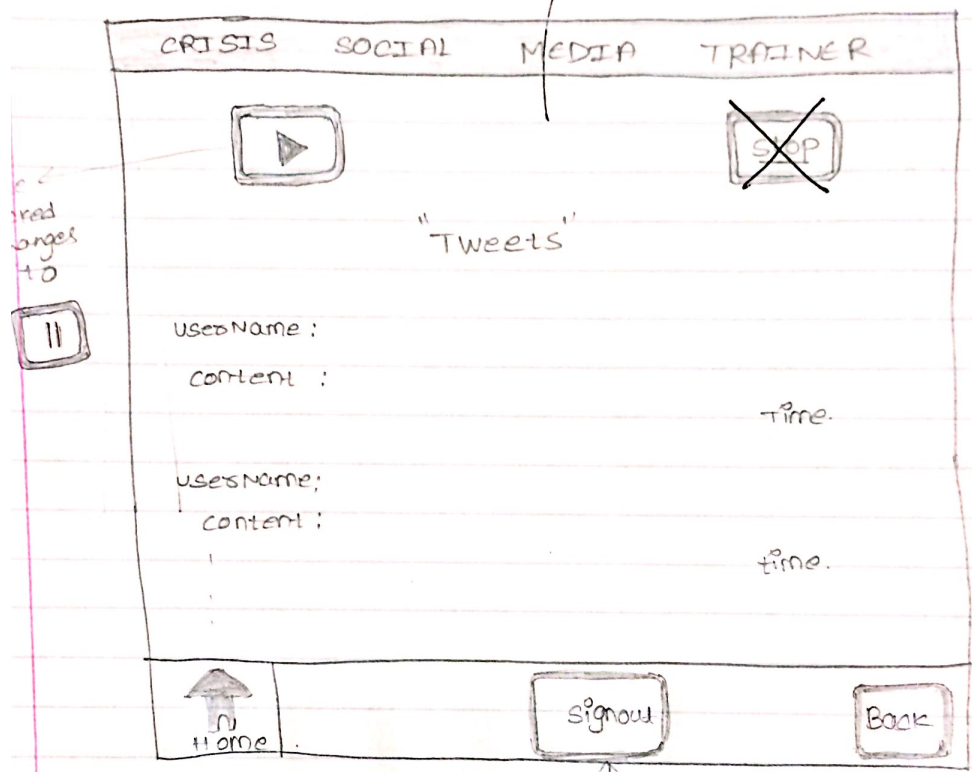
Upload new data set

pop up that collect the name, description, + filename

Restart



Notes

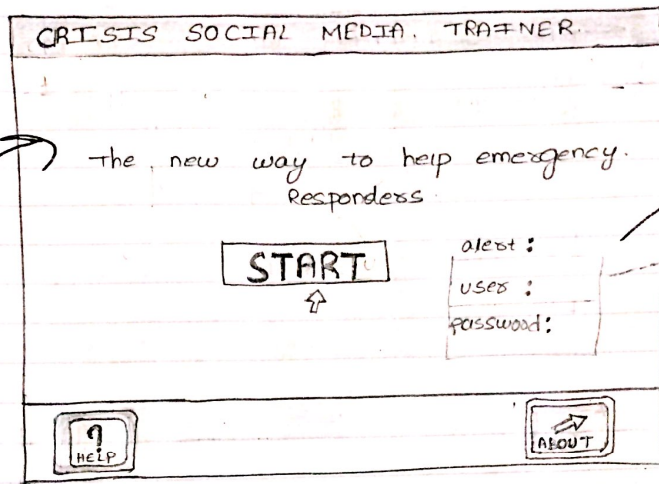


red
changes
to



if signed in
else
signin option appears

Type-1



do we need users?

uses authentication.

Should be more descriptive one sentence

help :- This will explain what happens when user clicks button start. It takes to next page, where user can either upload data or start viewing the existing data.

about ? This explains all about the tool. what is the ultimate goal of this tool is explained here.

Help	close
the ultimate goal of this tool is to help emergency responders understand what kinds of information they might see during an event and allow them	

About	close
version	1.1
privacy policy	→
Terms of use	→
End user License Agreement	→