University of Nevada, Reno

BEE: Bettering Emergency Evacuations

A thesis submitted in partial fulfillment of the
requirements for the degree of Master of Science
in Computer Science and Engineering.

by

Christopher William Ginac

Dr. Sergiu Dascalu, Thesis Advisor

May, 2020

THE GRADUATE SCHOOL

We recommend that the thesis
prepared under our supervision by

CHRISTOPHER W. GINAC

Entitled

BEE: Bettering Emergency Evacuations

be accepted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE

Sergiu M. Dascalu, Ph.D., Co-Advisor

Sushil Louis, Ph.D., Co-Advisor

Pavel Solin, Ph.D., Graduate School Representative

David W. Zeh, Ph.D., Dean, Graduate School

May, 2020

Abstract

Emergency evacuations are often confusing and dangerous. Fires can block roads, floods can wash out infrastructure, and hurricane evacuation zones can be hard to determine. The good news, though, is that the vast majority of people subject to evacuation orders have smartphones in their pockets.

This thesis presents BEE, or Bettering Emergency Evacuations, a system designed to make easier disseminating vital information during an evacuation or disaster. This system is composed of two distinct, yet complimentary, components. The first component is the EMS (Emergency Management System), which is used by Emergency Managers to define the evacuation zones, evacuation routes, and information to send to evacuees. The second component is the BEE iPhone application, which is used by citizens to know that they need to evacuate, what path to evacuate on, and what information they need to know during the evacuation.

This thesis presents the motivation of this project then describes in detail the requirements, architecture, user experience design, and implementation of each system component and the behind the scenes server that manages the data.

Additionally, the thesis contains a survey designed to gather information about what users think about the system's ability to disseminate important information, and it discusses the results obtained. Finally, related works and possible future developments are presented. The data collected in the survey shows that the efforts invested in designing and building BEE were successful, as the respondents rated BEE highly in every aspect questioned in the survey.

Dedication

This Thesis and Project is dedicated to the 85 people who lost their lives in the Camp Fire, the families who lost a loved one, and the nearly 50,000 residents of Northern California who were displaced.

You all lost so much that November; let's hope it doesn't happen to anyone else.

Acknowledgements

I would like to thank my thesis advisors, Dr. Sergiu Dascalu and Dr. Sushil Louis for their time and support throughout my academic career. I would also like to thank Dr. Pavel Solin for being a part of my committee and providing advice and support through this project. Without all of them, I would not have been able to make this project a reality.

I also would like to thank my wife and family. There are too many people to single out in this section, but suffice to say that I have the best support group that anyone could ever ask for.

Table of Contents

List of Tables

List of Figures

Chapter 1 Introduction

The horrifying pictures and videos coming out of the most destructive and deadliest wildfire in California history [1], the Camp Fire, shows that disseminating information about evacuations is vital to saving lives. Between the time the Camp Fire started at 6:29 am on November 8th, 2018 and when it was announced it was fully contained on November 25th, it burned 153,336 acres (almost 240 square miles, or over five times the area of San Francisco) and killed 85 people with more than 1200 missing at one point [2]. The town of Paradise, CA saw the vast majority of the fatalities in this fire, and the entire town (with some exceptions) was burned to the ground within the first hours of the blaze. At the height of the danger, about 52,000 people had evacuated due to the fire racing towards their homes [3]. In total, the fire burned down almost 14,000 residences and over 500 commercial buildings [4].

The town of Paradise had exactly one escape path, and it quickly became jammed with vehicles filled with residents fleeing the flames. With walls of flames bearing down on them, people abandoned their vehicles and ran for their lives. Once the fire swept through, bodies were found within homes, burned out cars, and on foot [5].

What if there was a way that technology could have helped those people survive? Fires will happen, evacuations will happen. How can we do evacuations better?

These questions led to Bettering Emergency Evacuations (BEE) being developed. So many people died and so many families had their world's turned upside down. Could technology help prevent that? This project aimed to find out.

First, it is important to understand the issues that evacuations face. Studying past disasters and evacuations can help with that. It was important to choose a couple of

extremely large disasters that had specific issues with evacuations. Since this project was born out of the stories coming from the Camp Fire, that was the primary disaster explored. The second disaster chosen was Hurricane Irma; with the stories of over six million people evacuating, heavy traffic, and no gas. Finally, a local point of view was chosen: the Caughlin Fire in 2011. The Caughlin Fire was chosen partially because of the closeness to the University of Nevada – Reno, but also a personal connection. I had friends that had to evacuate out of the path of the flames and their lives were at risk due to a lack of communication. Each of the disasters had specific issues and lessons to be learned. This document will go more in depth into those issues and lessons.

Once the issues and lessons from prior disasters are identified, then requirements and use cases can be created. This project would be split into two different (but complimentary) components: An Emergency Management System (EMS) for Emergency Managers to use to define the information, and an iPhone Application that civilians to use to consume and act upon the information sent to them. Requirements and use cases are identified for each component later in this document.

The main requirement is that the entire system will enable turn by turn navigation after an evacuation is ordered; and to facilitate the quick dissemination of information.

The EMS was designed to run on Amazon's Elastic Beanstalk cloud service (which is a part of Amazon Web Services or AWS). It includes the web application that the Emergency Manager can access through a modern browser (designed and tested on Google Chrome). Through this web application, the Emergency Manager can define all the information that they need to disseminate; such as evacuation areas, evacuation routes, additional instructions, and locations. This web application was written in

JavaScript and takes advantage of a mapping framework called Mapbox. The information defined by Emergency Manager is then put into a mySQL database, which is hosted on AWS also. This database is described in detail later in this document.

The iPhone application was designed to run on the latest iOS and latest iPhones released by Apple. The application is written in Swift and allows the users to see all the information that was defined within the EMS. This application also uses the iOS version of the Mapbox Framework for both mapping functions and navigation functions.

To assist in the flow of information from the EMS to the database to the iPhone Application, a server was designed with an API that allows both the EMS and the iPhone Application to send and receive data. This server is what puts the data into the database and retrieves the data from the database. This server was written in Java and is hosted alongside the EMS in the Elastic Beanstalk instance.

This project was designed to work in conjunction and in addition to existing solutions in the evacuation space. Existing solutions include the federally mandated Emergency Alert System (EAS) which come over the radio or over television, and the Wireless Emergency Alerts (WEA) which are sent to every cell phone. EAS and WEA have a very wide reach, but the format does not allow for much information to be given.

Other applications that could be used to assist in evacuations and emergencies include Code Red (a system similar to BEE but doesn't have functionality for navigation), and Google Maps.

To determine if BEE has met its goals, a user study was designed to gather user thoughts and feedback. The user study is a survey hosted on Google Forms. The survey required participants to have evacuated in the past so that they can give an accurate

comparison. The survey then asked the participants to give a summary of their experiences and asked them to gauge how much they agreed to six statements about their evacuation. Then a video showing a mock evacuation in BEE was shown and the participants were asked about the same six statements in relation to the mock evacuation. Then participants were given an opportunity to give feedback, share concerns, or make suggestions about how to make BEE a better application. The results from this user study are discussed later in this document but suffice to say BEE met all of its goals.

If BEE met all of its goals, it would not be a stretch to say that it could save lives in the future. Helping people evacuate quickly, safely, and with as much information as possible is a goal that all Emergency Managers strive for. Hopefully in the future BEE can help.

Chapter 2 Background

2.1 Incidents

The following incidents highlight the need to be able to disseminate vital information quickly and in a format that is easy to understand. The first two incidents were chosen due to their evacuations that affected a wide variety of people. These evacuations had major issues due to vital information not being communicated or due to evacuees not knowing or remembering instructions given to them by emergency managers. The third incident gives a local perspective (to the University) of how this application might be able to help evacuations.

2.1.1 Camp Fire - Butte County, CA 2018

What has become the most destructive and deadliest fire in California history [5], the Camp Fire started at 6:29am on November 8th, 2018 as a small fire on the side of the highway. That morning, a Red Flag Warning was issued for "Critical Fire Weather Conditions through [the following] Morning for Portions of Interior Northern California" [6]. In the final update before the fire began, issued at 5:33am, the National Weather Service in Sacramento warned of low humidity readings of 15 to 30 percent, locally strong winds of up to 45 miles per hour, and specifically calling out that "any fires that develop will likely spread rapidly" [6]. Even knowing that the weather was perfect for a massive wildfire, nobody was expecting what was to come less than an hour after the update from the NWS.

One of the towns closest to the starting point of the fire was Paradise, CA. According to the United States Census Bureau, Paradise was the home to an estimated

26,682 people in 2017 [7]. As residents were waking up, they didn't know that later that morning, they would be running for their lives.

The leaders of Paradise, CA knew that there was danger that something like the Camp Fire could happen. Their town was surrounded by dense brush and forest. They had one main evacuation route out of the city, and knew that they needed to have a plan in place in case of disaster. In June 2016, the town practiced their evacuation plan in a drill. The drill took two hours to complete, and during it, evacuation zone maps were given out to people who lived in Paradise [8].

That evacuation plan failed the morning of November 8th, 2018.

Alerts that were supposed to be sent to residents had failed to be sent out. According to the Butte County Sheriff's Department, the alerts are not sent as soon as the danger is present. The workflow is that firefighters must be on scene, know which way the winds are spreading the fire, and what towns are in danger. Only then can that information be given to the Sheriff's Department, who makes the final decision on what alerts are sent out, and to whom [8].

According to the Mercury News, many residents were unaware of the danger to their homes and lives. This caused what the Mercury News describes as "terrifying and crowded last-minute escapes, even death." Once residents learned of the danger, usually from smelling smoke, or seeing the apocalyptic scene outside, they rushed to evacuate only to get stuck in traffic. Many evacuees were surrounded by flames while hoping the traffic would ease up [8].

Some would eventually give up trying to evacuate in their vehicles and try to flee on foot. One couple, Richard and Zetta Gore, spoke to the Mercury News about their

escape. They said that there were stuck in traffic, barely moving for a whole half hour as the flames descended upon their town. They decided they would rather die trying to escape the fire rather than being burned alive in their vehicle. As they came upon a cliff, they called their son and said "Zach, this is it. We're going to make a run for it on foot. This could be the last time we ever talk to you" [5]. They slid down the side of the cliff, waded through a muddy stream, and then proceeded to evacuate on foot for five miles. They survived, but many didn't.

The final official count wasn't known for weeks, but in the end 85 people perished in the flames [4]. Bodies were found inside homes, inside vehicles that were overcome by flames during the evacuation, and alongside the road [5]. At one point, over 1,000 people were listed as missing [2]. During the height of the danger, over 52,000 people were under evacuation orders [3]. The town of Paradise is basically gone along with over 18,500 buildings [4].

The Camp Fire shows that something to help with evacuations is needed. We may never know how many people could have been saved if they got timely alerts and directions out of town, but it stands to reason that a tool that does timely alerts and directions could have saved lives during the Camp Fire.

2.1.2 Hurricane Irma – Monroe County, FL 2017

In their 2:00 pm EDT update on August 26th, 2017, the National Hurricane Center started tracking a tropical wave that was over western Africa that had the potential to turn into a tropical cyclone [9]. Over the next two weeks, that tropical wave would organize and strengthen into Hurricane Irma. Hurricane Irma became an extremely

strong hurricane that was forecasted to directly hit Southern Florida, so naturally evacuations were ordered.

Monroe County, FL was the first to order evacuations. The Florida Keys were in the direct path of the hurricane and they were at risk of being cut off from the mainland due to the fact there is exactly one highway in and out. Also, the forecast called for five to ten feet of storm surge, hitting both sides of the islands due to the fact that the winds would shift once the eye of the hurricane passed [10]. Considering the forecast and the damage that could be caused to property and life, Monroe County called for a mandatory evacuation of all residents of the county [11]. This order was to go into effect at 7:00 pm on September 6th, 2017, about four and a half days before Hurricane Irma hit.

Other evacuations would be called throughout the Florida Peninsula prior to Hurricane Irma making landfall. In total about 6,000,000 residents in Florida were evacuated [12]. Those evacuations were definitely needed considering the damage that was left behind after Hurricane Irma.

In total, there were seven deaths directly attributable to the hurricane in Florida, including three in Monroe County. According to the National Hurricane Center, "the damage was most severe in the Florida Keys" [12]. The vast majority of buildings were rendered unsafe for occupation on the Keys, and the power was completely knocked out to the chain of islands. The main road in and out of the Keys was closed until October 1st, 2017 [13], and life took much longer to go back to normal for residents of Monroe County.

This disaster is very different than the Camp Fire disaster in Section 2.1.2. There were weeks of warning, and evacuations were ordered almost a week ahead of time.

However, this disaster has taught many lessons on how large scale evacuations should happen given these conditions.

With 6,000,000 residents fleeing such a large storm, one that was forecasted to affect the entire peninsula, the question should be asked, "Where will they all go?" According to one expert, this mass evacuation was successful [14], but there were reports of the evacuation commonly taking 20 to 30 hours. What could cause traffic to be so bad that it would take over a full day of driving just to evacuate? The problem is that almost half of the people who did evacuate didn't need to, they were not under any type of evacuation order.

Large scale hurricane evacuations are rarely issued for the wind associated with hurricanes (with the exception of less than sturdy residences such as mobile homes), they are usually issued for the water. Flooding and storm surge are the killers in hurricanes, and those will only affect small portions of the area that is in the way of the storm. This means that not only were too many people on the road, they were evacuating too far also.

With fuel shortages along evacuations routes severe enough that the Governor of Florida, Rick Scott, ordered the state highway patrol to escort fuel trucks to stations, it's easy to see where improvements can be made [15].

2.1.3 Caughlin Fire – Washoe County, NV 2011

On the night of November 17th, 2011, residents of Reno, NV went to bed under a High Wind Warning [16], not an uncommon occurrence in Fall and Winter in this area. In the warning, the National Weather Service said that there would be gusts of up to 70 to

80 miles per hour in the foothills and that “strong downslope winds may damage trees… power lines and property” [16]. Unfortunately, that prediction came true.

Early in the morning on November 18th, 2011, at 12:22 am, two power lines in the foothills southwest of Reno came within close contact of each other and arced [17], starting a fire that spread quickly due to the high winds and very dry fuels. Not only did the flames itself spread quickly, burning embers were carried by the wind up to a half mile away from active flames.

As a resident of Reno, NV at the time, I remember having a friend who lived up on Skyline, not too far away from where the fire started. I called her up, and she had no idea there was a fire until the power went out while I was on the phone with her and she could see the orange glow. She lived in a suite on the back side of a very large home, but there was no way for firefighters to know she was back there. She received no warning or order to evacuate, and she and her dogs could have died if I did not call and get her to evacuate. Luckily her home was spared from the flames, unlike 37 other structures that were lost in the fire [17].

During the fire, the road with the largest capacity was shut down, leaving residential streets as the main evacuation route. Amazingly, the fire was contained at only 1935 acres, which considering the winds is quite impressive. 4,000 people were evacuated due to the fire [17].

This fire was included in the incidents list to give a local perspective on natural disasters and evacuations. While there were no major problems evacuating like the Camp Fire and Hurricane Irma, there is still a lesson to be learned that not everyone will get the news to evacuate.

Chapter 3 User Experience Design

3.1 Emergency Management System

3.1.1 Requirements

The most important aspect of an emergency management system, at least from the emergency manager's point of view, is to be able to get pertinent information out quickly and reliably to as many citizens in the way of a disaster as possible.

The functional requirements of the emergency management system are listed in Table 1, below. Many of the requirements deal with defining what information to send to civilians, both prior and during an emergency. Having predefined evacuation zones, evacuation routes, and instructions to send out to civilians during an evacuation saves time and hopefully lives. FR1 through FR4 deal with information defined before an emergency, FR5 through FR9 deal with information defined during an emergency, and FR12 through FR14 deal with information updated during an emergency. FR10, FR11, and FR15 deal with actually sending out the information to citizens.

Finally, FR16 through FR19 deal with allowing the emergency manager to see information about the civilians under their care. These requirements will help emergency managers know who is missing, who doesn't know about the evacuation, and where to send emergency resources. Knowing this information will not only help keep civilians safe, but keep first responders safe by keeping them out of harm's way when they aren't needed there.

Table 1: Functional Requirements for Emergency Management System

Name	Description
FR1	The user shall be able to define evacuation zones prior to an emergency.
FR2	The user shall be able to define evacuation routes prior to an emergency.
FR3	The user shall be able to define evacuation instructions prior to an emergency.
FR4	The user shall be able to define locations of various resources prior to an emergency.
FR5	The user shall be able to define locations of various disasters or emergencies during said disaster or emergency
FR6	The user shall be able to define evacuation zones during an emergency.
FR7	The user shall be able to define evacuation routes during an emergency.
FR8	The user shall be able to define evacuation instructions during an emergency.
FR9	The user shall be able to define locations of various resources during an emergency.
FR10	The user shall be able to send an evacuation order to citizens with the application.
FR11	The user shall be able to send an evacuation warning to citizens with the application.
FR12	The user shall be able to update evacuation zones as necessary during an emergency.
FR13	The user shall be able to update evacuation routes as necessary during an emergency.
FR14	The user shall be able to update instructions as necessary during an emergency.
FR15	The user shall be able to send out updated information from FR12, FR13 and FR14 to citizens.
FR16	The user shall be able to see data on which citizens were sent the alert.
FR17	The user shall be able to see data on which citizens acknowledged the alert.
FR18	The user shall be able to see locations from citizens on a map during their evacuation.
FR19	The user shall be able to see data on which citizens marked themselves safe after an evacuation.

3.1.2 Workflows

As discussed in Section 3.1.1, the most important aspect of the Emergency Management system is to get information out to the public as quickly as possible. In this section, we will work through the most basic workflow of the Emergency Management System, sending out an evacuation alert.

First, the Emergency Manager will access the Emergency Management System. Then they will define a new evacuation zone. During creation of the zone, they will be able to define the area of the zone, define the instructions related to the zone, and set the severity of the zone.

Then the Emergency Manager will define evacuation routes. During the creation of the route, they will define certain points on the map that will act as checkpoints along the route and, once finished, the system will automatically create a route along those checkpoints. If the generated route is incorrect, the emergency manager will be able to delete the route and try again. If there are more routes to create, then the emergency manager will start the process again.

Finally, the emergency manager will be able to send out information using push notifications. This will start the evacuation and make sure information is in the hands, literally, of civilians.

This workflow is shown in Figure 1.

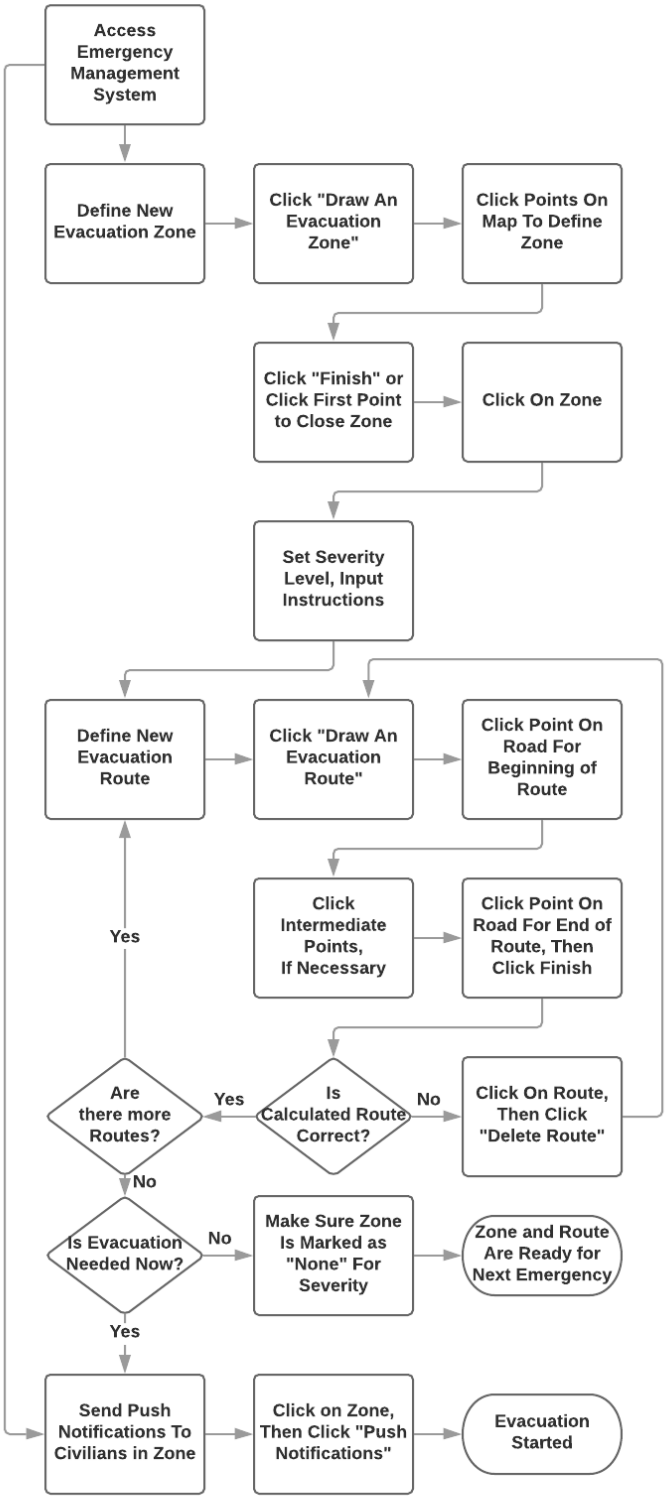


Figure 1: Emergency Manager Workflow

From a system standpoint, the first workflow would be to load the information for display in the Emergency Management System. First the emergency manager will open the Emergency Management System. Then the system will send a request to the BEE Web Server for any previously defined alerts or evacuation zones, including any routes and instructions. The server will gather this information from the BEE Database, package it into a JSON response, and send it back to the Emergency Management System.

Once the alert or zone information is retrieved, then the Emergency Management System will make another request to the BEE Web Server to get any previously defined locations. In much the same way as before the BEE Web Server will get the data from the database and return a JSON formatted response. Finally, the Emergency Management System will display all of the data retrieved on a map so the emergency manager can see all the data quickly and easily.

Figure 2 is a sequence diagram of the above workflow showing the interactions between the user, the Emergency Management System (or BEE Web Application), the Web Server, and the Database. This sequence diagram is based upon the diagrams described in Ian Sommerville's "Software Engineering" [18].

3.1.3 User Interface

When the Emergency Manager service is accessed, the emergency manager will see a map of their local area. On the left hand side, there are buttons and tools. This initial view can be seen in Figure 3.

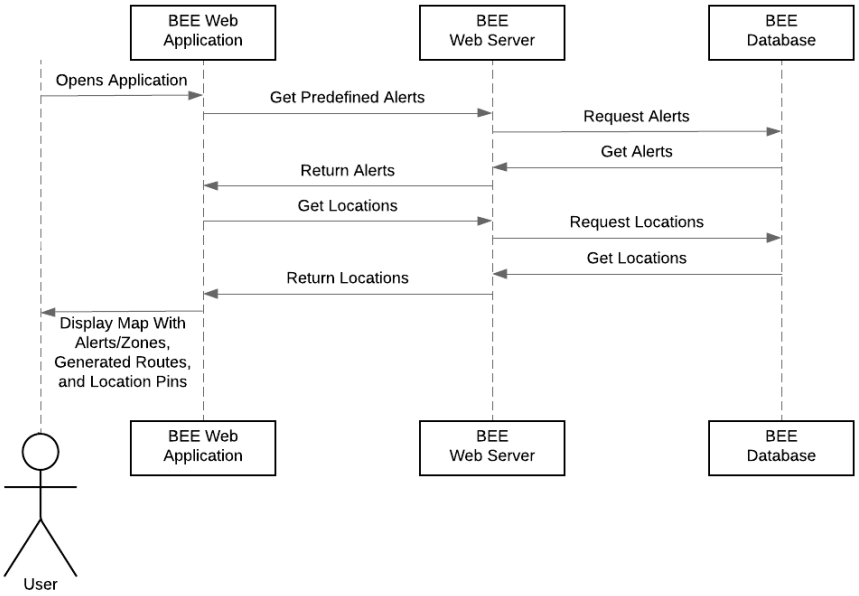


Figure 2: Emergency Manager Start Up Sequence Diagram

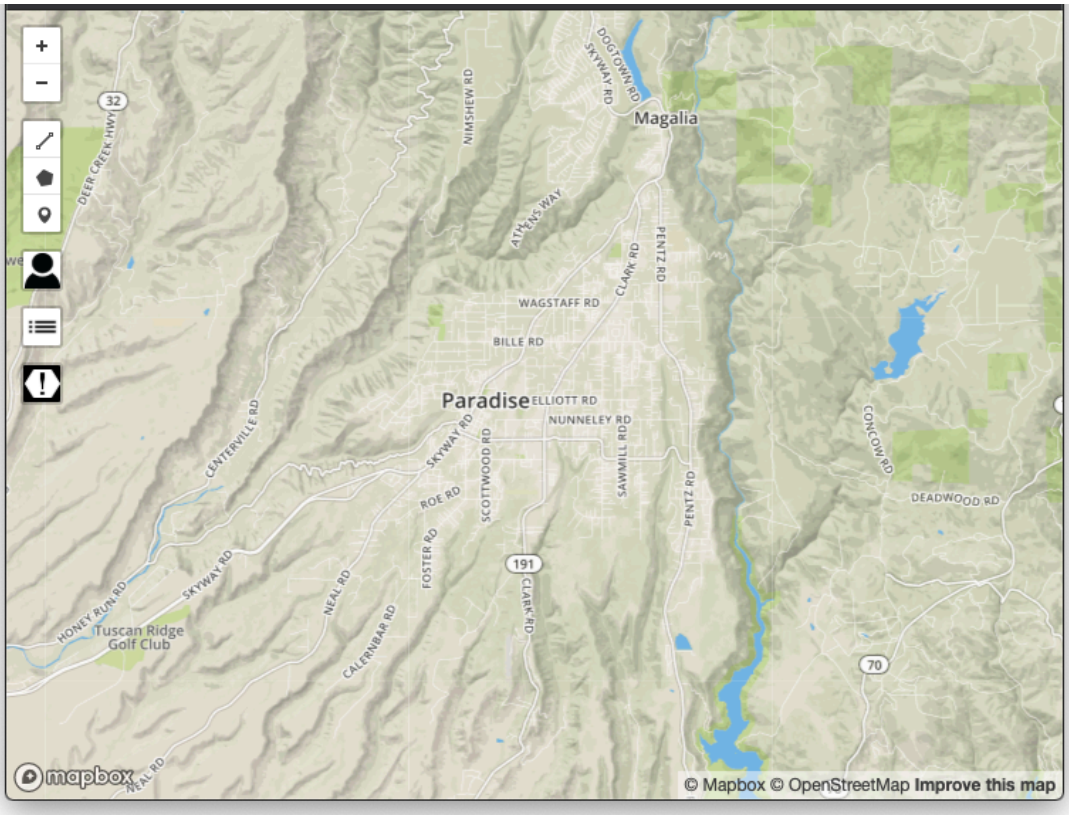


Figure 3: Initial View of Emergency Manager Interface

The tools on the left-hand side are (from top to bottom) “Draw an Evacuation Route”, “Draw an Evacuation Zone”, “Create a Location”, “Show Evacuee Locations”, “Show Evacuee Information”, and “Show Reports”. A closer look at these tools are shown in Figure 4.

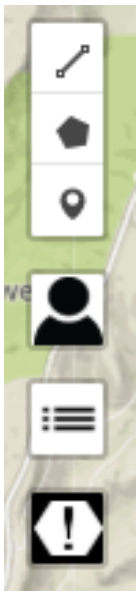


Figure 4: Closer View of Available Tools

The first step in communicating an emergency would be to draw a zone. The Emergency Manager will click the “Draw An Evacuation Zone” button and draw the zone by clicking around the map.

To see information or to change details about a zone on the map, the Emergency Manager will click on the zone and a popup will appear. In this popup, the Emergency Manager can change the severity of the evacuation between “None”, “Warning”, and “Order” which will also change the color of the zone. The Emergency Manager may also

type in instructions to send to evacuees into a text area. Once the information is correct and ready to broadcast, the "Push Notifications" button can be pressed to send out push notifications to users in the affected area (see Section 3.2.3, below). This popup with sample information can be seen in Figure 5.

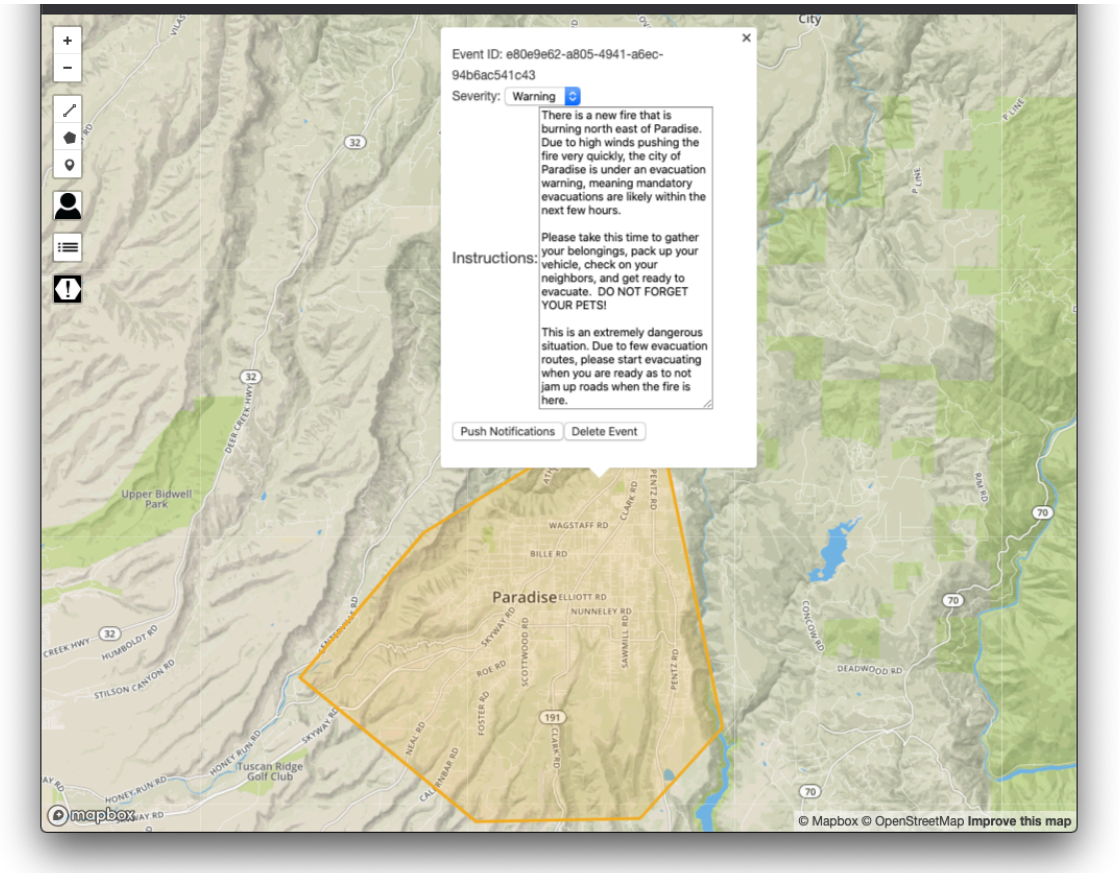


Figure 5: Event/Zone Information Popup

To define an evacuation route, the Emergency Manager will need to press the "Draw An Evacuation Route" button. The Emergency Manager should define only mainline routes, as the app will route the evacuee to the closest point on the fastest evacuation route. Users will click on the road at the beginning of the route, and click on

the road just outside of the evacuation zone to define waypoints. The web app will automatically snap to the shortest route between the waypoints. If this route is incorrect, the user will delete the route by clicking on it and redrawing the route with additional waypoints. A sample route with two waypoints, and the popup if a user clicks on a route is shown in Figure 6.

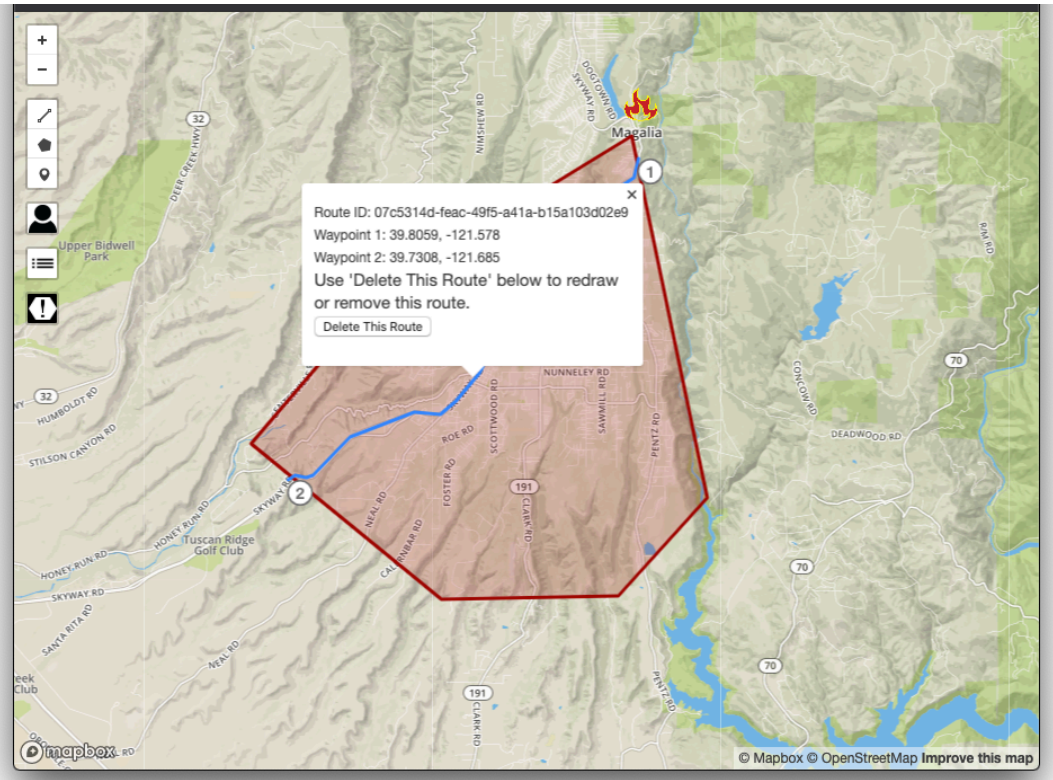


Figure 6: Sample Evacuation Route

To give Evacuees and Civilians more information about the event, the user can create a location by pressing the "Create A Location" button. Once a marker is dropped, the user can click on the marker to define a name, the type of location it is, and write additional information (such as information about a fire, or hospital information) in the

text area. The location types include (but aren't limited to) locations of disasters, shelters, hospitals, and police stations. A sample marker and information popup is shown in Figure 7.

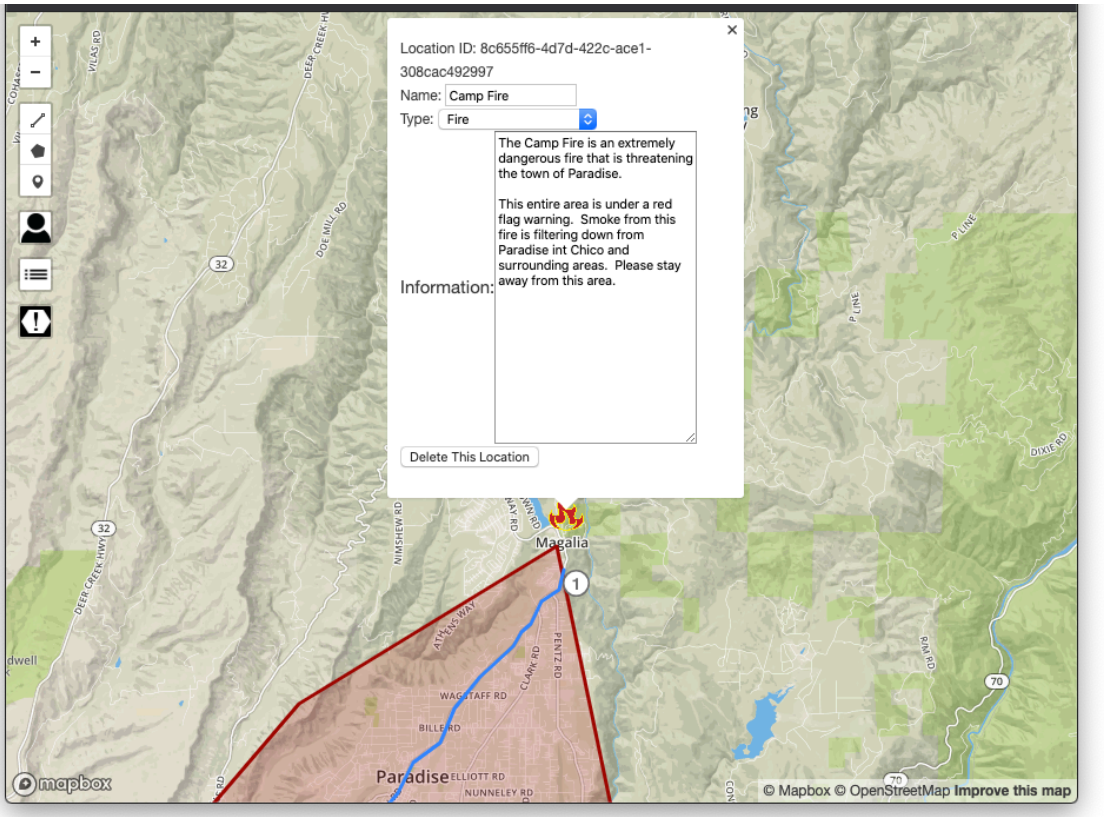
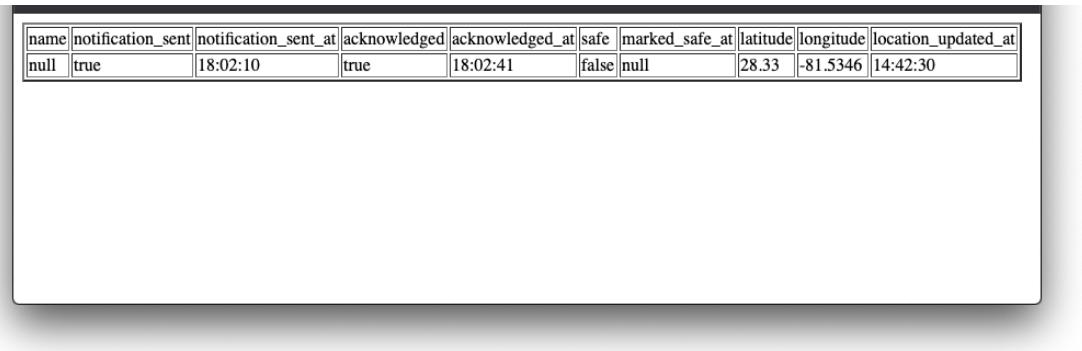


Figure 7: Location Marker Popup

The Emergency Manager can also see a visual representation of where Evacuees are by pressing the "Show Evacuee Locations" button. This will place a small dot at the location of each user of the Civilian application. App users will be depicted using color codes, green denoting that the user has completed the evacuation and is safe, blue denoting that the user is in the process of evacuating, yellow denoting that a push notification has been sent to the user's device but has not started evacuating, and red

denoting that the user's last known location is in the evacuation zone but the user has not acknowledged the order to evacuate in the phone app.

If more detailed information about evacuees is required, the Emergency Manager can press the "Show Evacuee Information" button. This will redirect the browser to a page with information about evacuees. The page includes a table with the evacuee's name (if they have started evacuating, as the application doesn't send names and locations unless the evacuation has started), notification information, acknowledgment information, and location information. A sample list is shown in Figure 8.



name	notification_sent	notification_sent_at	acknowledged	acknowledged_at	safe	marked_safe_at	latitude	longitude	location_updated_at
null	true	18:02:10	true	18:02:41	false	null	28.33	-81.5346	14:42:30

Figure 8: Sample List of Evacuees

While evacuating, users of the iPhone application can report dangers and other information from within the app (see Section 3.2.3 below). The Emergency Manager can see these reports by pressing "Show Reports" button. This will display icons on the map at the location the user was at when they reported the event. The Emergency Manager may then click on the icon to see information about the report and delete the report if necessary. This is shown in Figure 9.

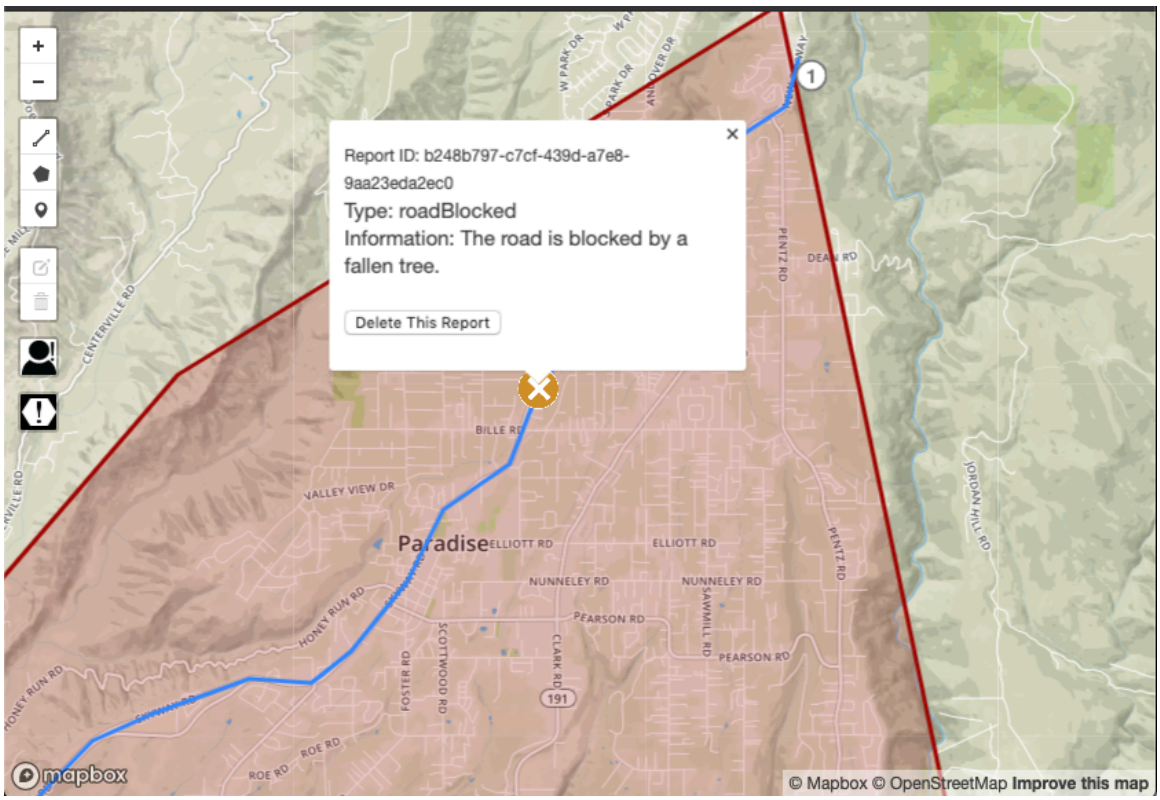


Figure 9: Sample Report and Popup

3.2 Civilian iOS Application

3.2.1 Requirements

As was discussed in Section 3.1.1, the section on requirements for the Emergency Management System, the most important aspect of an emergency management system is to get pertinent information out quickly. The requirements in Section 3.1.1 were written to help the Emergency Manager get information out, and the requirements in this section were written to help civilians get that information.

The most important aspect of an application that helps civilians evacuate is to make the information they received easy to understand, and to make sure they received it. FR2 through FR7 deal with displaying and conveying the information sent to them by the emergency manager in an easy to understand and visual way. FR5 in particular is the most important aspect of the BEE application, as BEE is the only application that has turn by turn navigation for evacuation routes. FR1 deals with receiving the alert. The functional requirements of the BEE iPhone application are in Table 2.

In addition to functional requirements, there are a few non-functional requirements that the BEE iPhone application must meet. These requirements are self-explanatory, and they are listed in Table 3.

Table 2: Functional Requirements for Civilian Phone App

Name	Description
FR1	The user shall be alerted when a new evacuation warning or order is issued for their area.
FR2	The user shall be able to see a visual representation of active evacuation orders or warnings for their area.
FR3	The user shall be able to acknowledge receipt of an evacuation warning or order.
FR4	The user shall be able to view any instructions provided by the authorities.
FR5	The user shall be able to view turn by turn navigation out of the evacuation area if an evacuation order is active for their area.
FR6	The user shall be able to see a visual representation of evacuation routes available to them prior to navigation.
FR7	The user shall be able to see a visual representation of locations of hazards, shelters, police stations, hospitals, and other locations on the map.

Table 3: Non-Functional Requirements for Civilian Phone App

Name	Description
NR1	The application shall run on the latest version of iOS available.
NR2	The application shall support push notifications.
NR3	The application shall know the device's location.

3.2.2 Workflows

The first action that the user of the BEE iPhone Application would be to install and set up the application. First the user will be asked to enter their name, only to be used during emergencies to protect privacy. Then the user will be asked to set location tracking to always to receive alerts, and also to allow notifications. Each of these configuration options are important to the overall workflow of getting information from emergency managers to civilians. This workflow is shown in Figure 10.

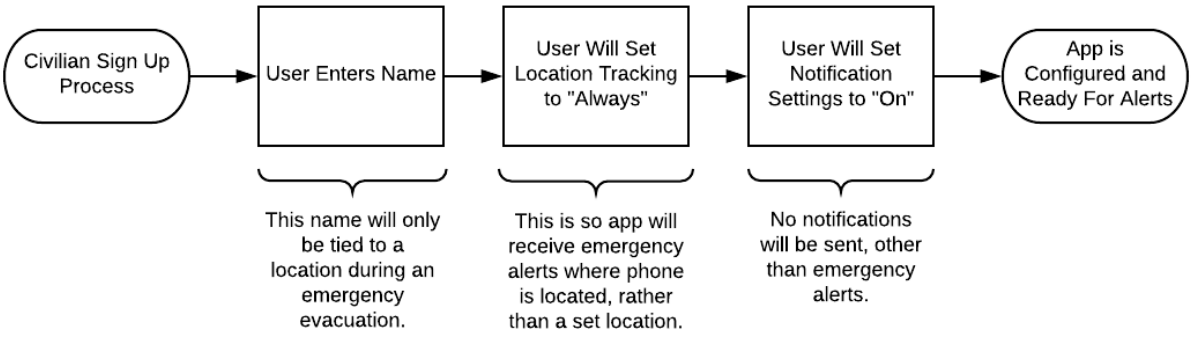


Figure 10: Civilian Set Up Workflow

However, the most important workflow in the civilian iPhone application is to actually evacuate. This workflow is not triggered by the user, it is triggered by getting an alert pushed to their phone by the emergency manager.

Once the alert is pushed to a user's device, their device will display the notification and allow the user to open the application with a single click on the notification.

Once the app is opened after an alert is pushed, it will display the alert information provided by the emergency manager. The user will then acknowledge receipt of the alert and that information is sent to the server to store, showing that the user knows they should evacuate. Once the alert is acknowledged, the application will display a map that shows information on active alerts, evacuation routes, and defined locations. Then the user can start navigation to evacuate out of the evacuation area. The application will determine the best route, and lead the user out of harm's way. When the navigation is finished, the user will be marked safe in the database.

This workflow is shown in Figure 11.

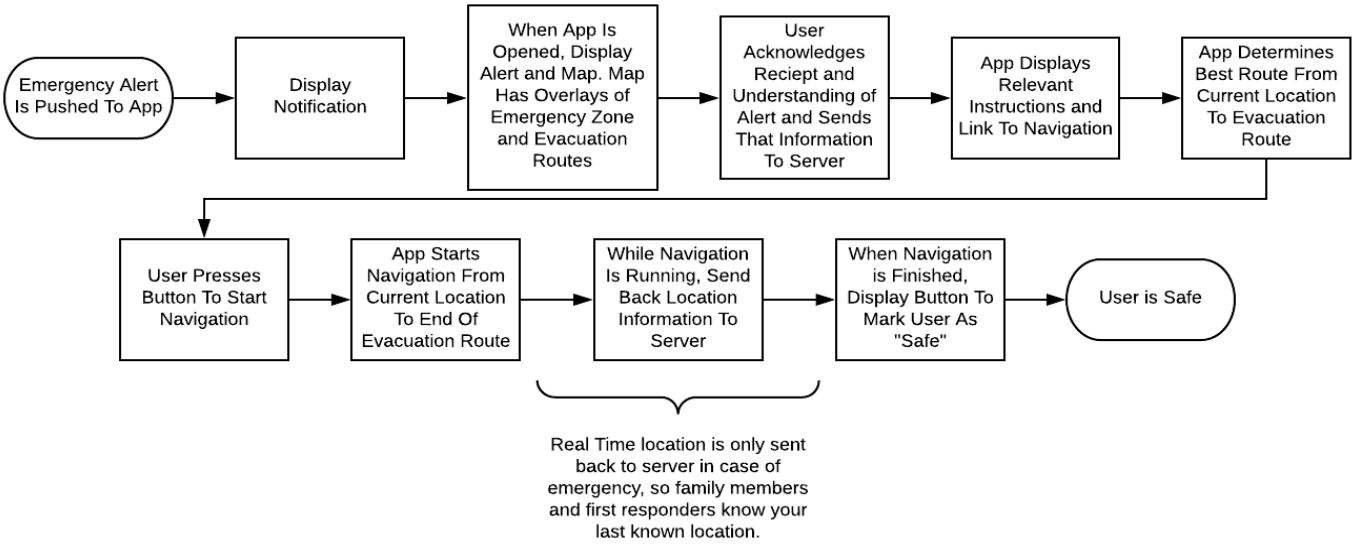


Figure 11: Civilian Emergency Workflow

This workflow is also shown in a sequence diagram in Figure 12. This sequence diagram is more technical and has each call to the BEE Web Server and related requests to the BEE Database.

First the user will open the application, which then will request the server to update the user's location in the database. Then the application will make a request to get any alerts for the user's area. The server will query the database for all active alerts for the area then filter out the alerts that aren't close to the user. The application will then display the alerts to the user.

After the alerts are shown to the user, the application will make a request of the server to get any defined locations in the immediate area around the user. Again, the server will query the database and then filter out locations that are too far away. Once this information is sent back to the application, it will display the locations on the map.

If the user acknowledges the alert, the application will request the server to update the acknowledgement flag for the current user. While the user is navigating, the application will repeatedly update the user's location, until they are finished with navigation. The application will then request the server to update the safe flag for the current user.

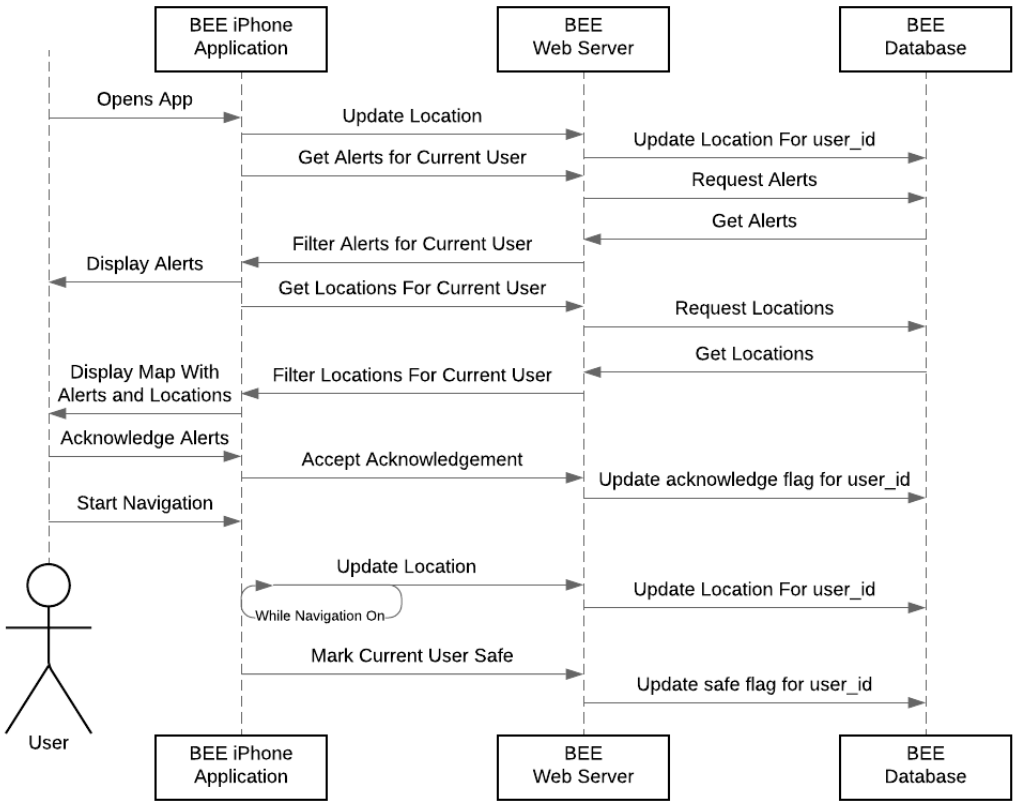


Figure 12: Evacuation Sequence Diagram

The only other thing in the application that the user can do other than evacuate is to report a hazard to the emergency manager. First the user reports the hazard, with information. The application will send a request with the hazard and associated

information to the server which will then store it in the database. This sequence is shown in Figure 13.

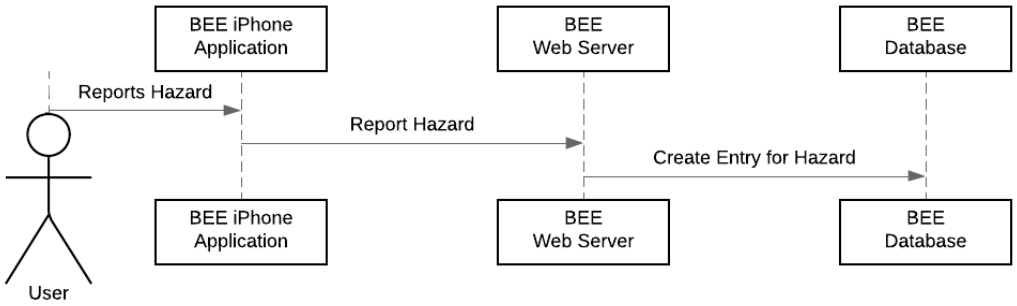


Figure 13: Report Hazard Sequence Diagram

3.2.3 User Interface

When the application launches for the first time after installation, it will first ask if it can track the user's location. There are three levels of location tracking that are offered by the built in API for location in the iOS operating system. First is "Only While Using the App", second is "Always Allow", and third is "Don't Allow".

As the application is location based, if "Don't Allow" is selected, then the app will not work.

If "Always Allow" is selected, then the application can send periodic location updates to the server so the user can get notifications about evacuations at their current location. If "Only While Using the App" is selected, the user will only receive notifications about evacuations for the location they were at the last time the app was opened on their phone.

This location permissions modal is shown below in Figure 14.

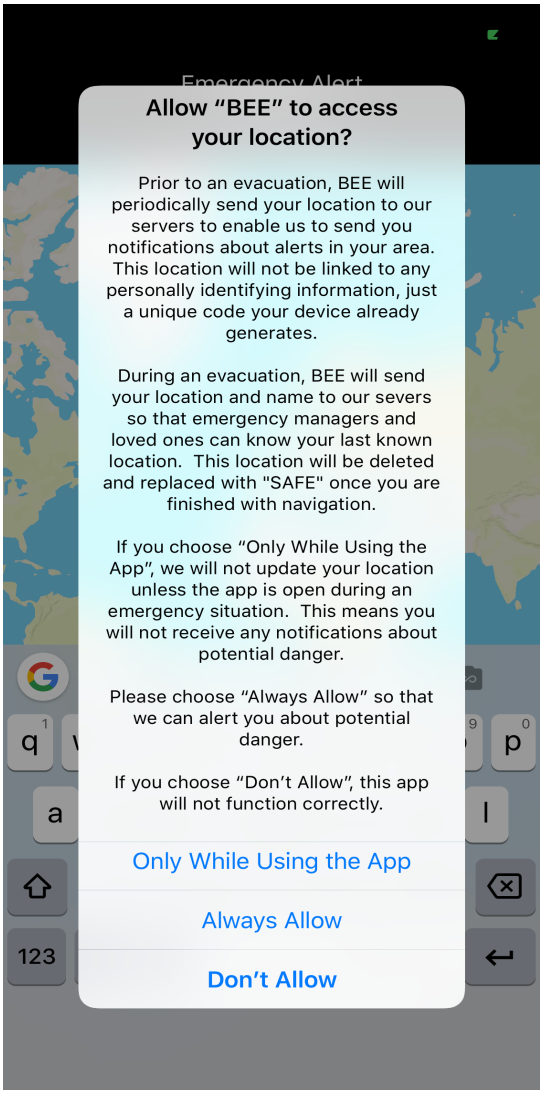


Figure 14: Asking for Location Permissions

Once the application gets location permissions, it will ask for permission to send Push Notifications to the user, as seen in Figure 15.

Next, the application will ask the user to input their name. The user's name is stored locally and is not sent to the server during normal operation. The only time the user's name is stored in relation to their location is when there is an evacuation ordered

for the user. This will allow loved ones and emergency managers to determine if someone is missing, and their last known location.

The modal asking for the user's name is in Figure 16.

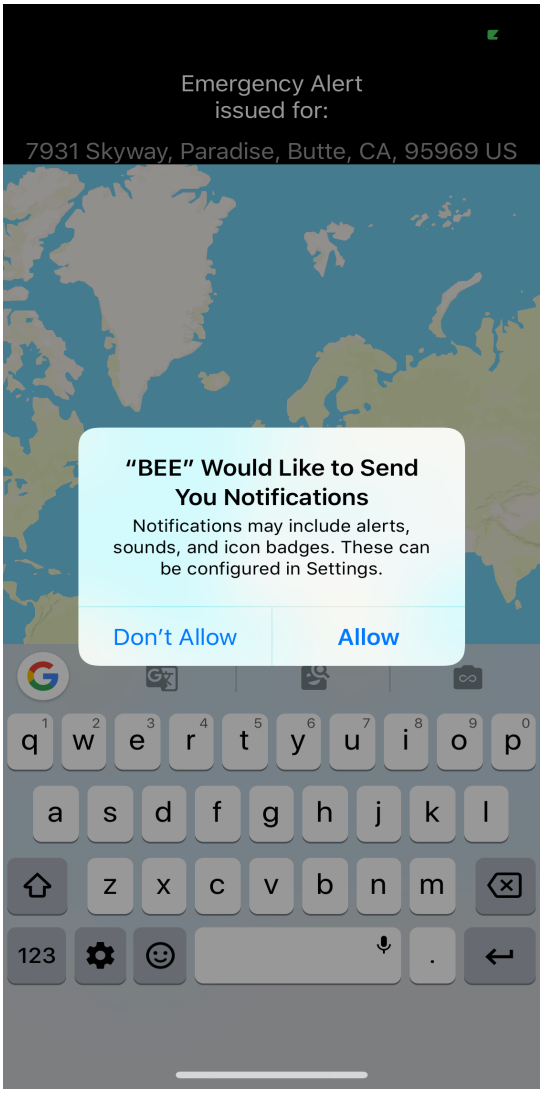


Figure 15: Asking for Notification Permissions

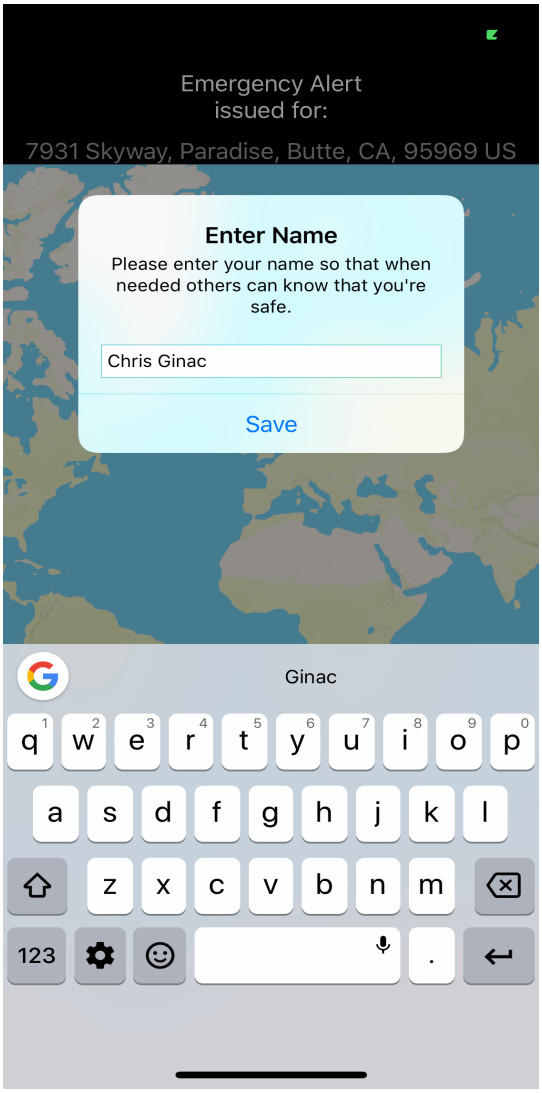


Figure 16: Name Input

In the event there is an evacuation warning sent by an emergency manager, a push notification can be sent to every device that is located in the evacuation warning zone.

Once the user taps this notification, it will launch the app. An example of this notification is seeing in Figure 17.

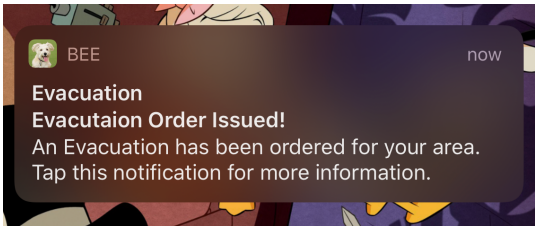


Figure 17: Push Notification

Once the application launches, then any instructions given by the Emergency Manager will be displayed in a modal, as see in Figure 18.

Behind the scenes, the application will update the server to say that the user received the warning. Once the user clicks on "Acknowledge Order", the application will update the server to say that they understand the warning and is taking steps to follow the instructions.

Once the warning is acknowledged, then a map of the user's area will be displayed with the outline of the warning, and any locations that the Emergency Manager has put on the map. In this example, the Florida Keys are under an evacuation warning due to a hurricane just south of Florida. See Figure 19.

If conditions get worse, then an Evacuation Order will be issued. Again, a push notification can be sent out by the Emergency Manager. Just like with the warning, instructions will be displayed awaiting acknowledgement of the order, as seen in Figure 20.

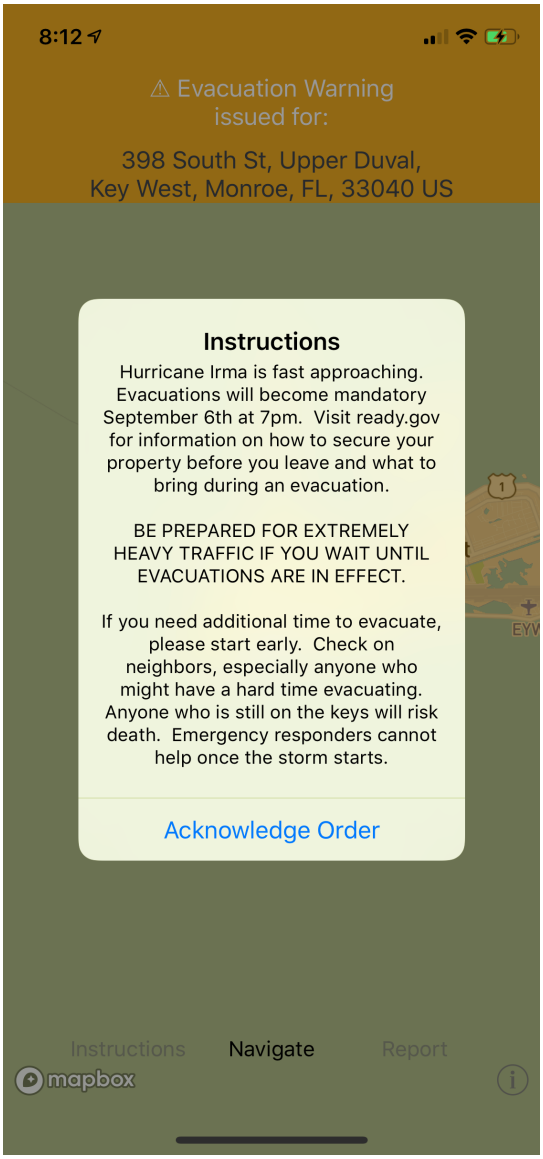


Figure 18: Example Evacuation Warning

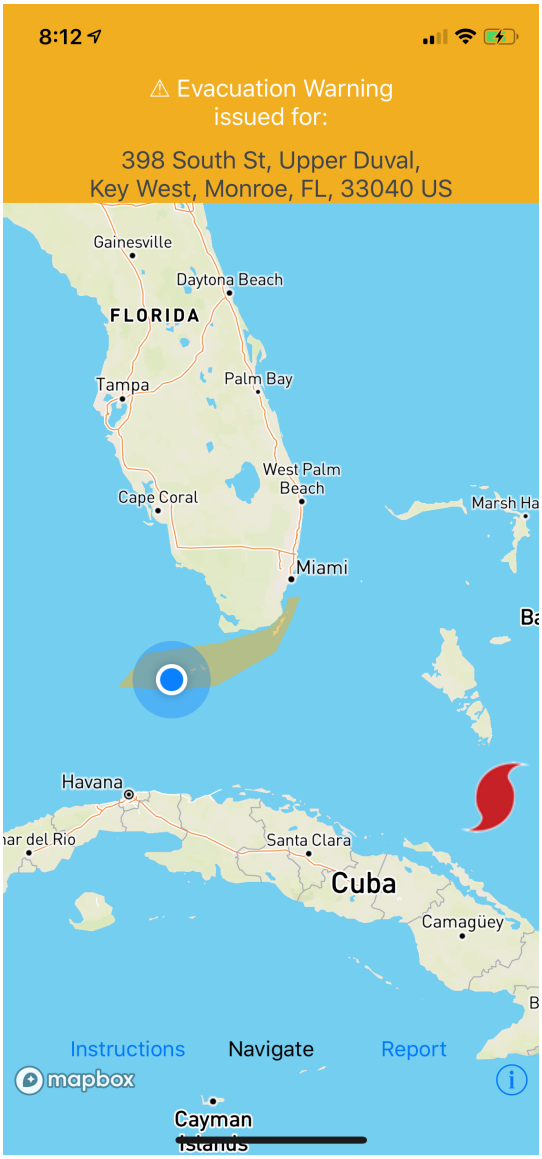


Figure 19: Example Hurricane Situation

Once the order is acknowledged, the map with overlays will be displayed as it was with the warning, however, now it will be displayed with an evacuation route. This route has been designated by the Emergency Manager as one of the best routes out of the evacuation zone. The application will determine the closest point on the route, and in the case where multiple routes are defined, the application will determine which route will be

the fastest and automatically route the user along that route. This search is similar to an A* search [19]. This is shown in Figure 21.

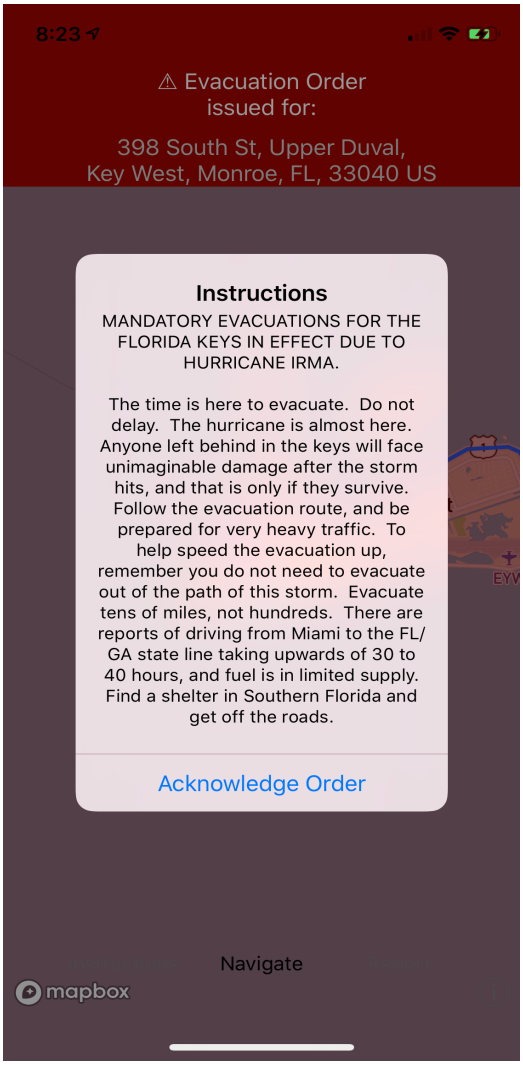


Figure 20: Example Evacuation Order

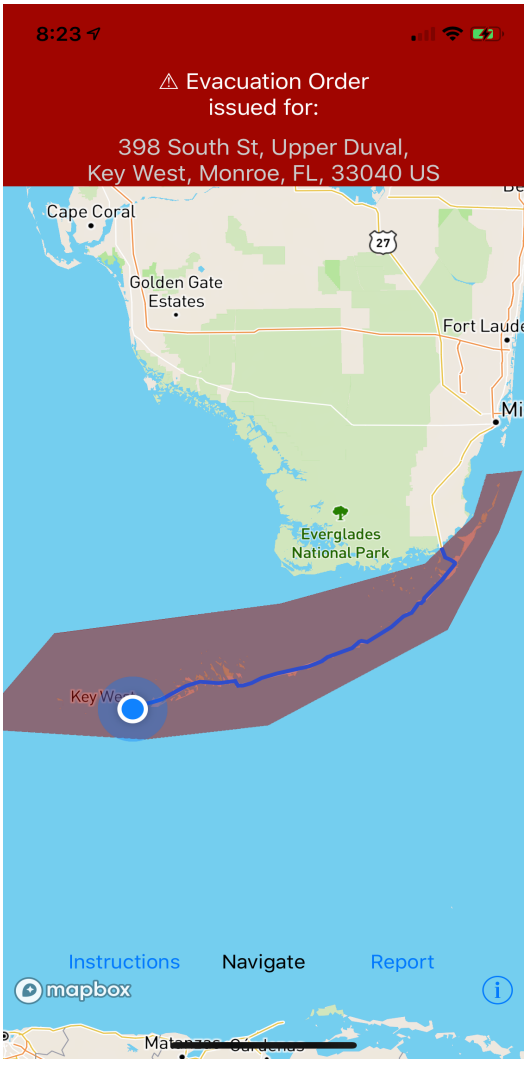


Figure 21: Example Evacuation Order Map

Since the entire point of this application is to make evacuations safer, faster, and less confusing; the user can click on "Navigate" and be given turn by turn navigation to the end of the evacuation route. This is shown in Figure 22.

Finally, once the user is safely out of the Evacuation Zone, their location on the server is changed to “SAFE” so their privacy is protected. If the user cancels the navigation, they will be asked if they should be marked safe. This is shown in Figure 23.

The user is marked safe immediately after leaving the Evacuation Zone since they are no longer in danger. Also, navigation is turned off once the user reaches the end of the evacuation route due to many evacuees evacuating too far and clogging roads.

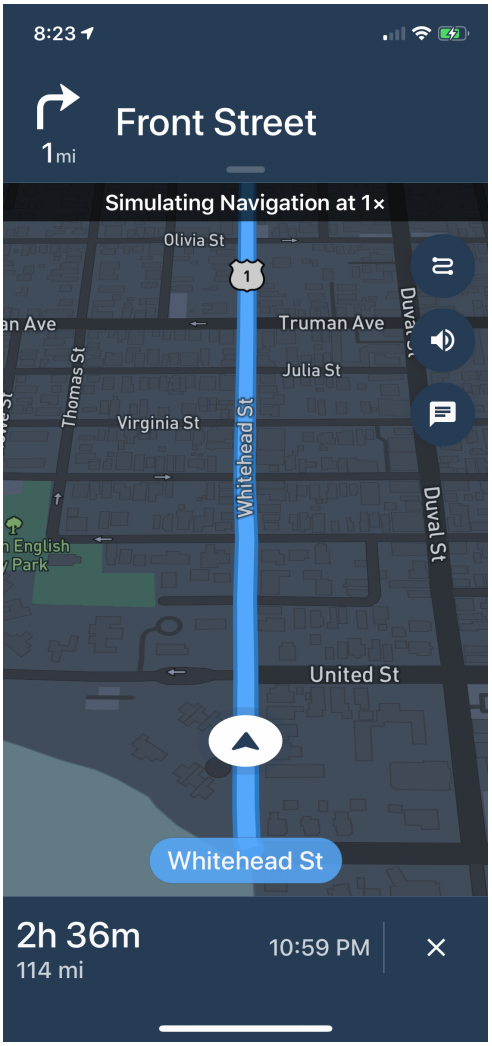


Figure 22: Turn by Turn Navigation

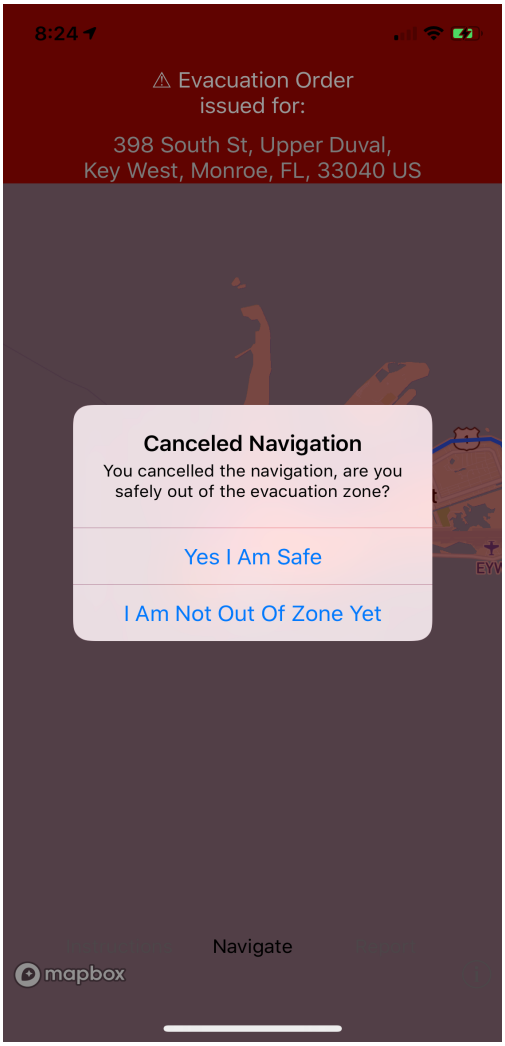


Figure 23: Canceled Evacuation Navigation

Chapter 4 Implementation

4.1 Project Architecture

BEE is broken up into two main user facing components, the Emergency Management System and the Civilian iPhone Application. To communicate information between the Emergency Management System and the iPhone application, a web server was created to process and send information on request. To store all of the information, the BEE Web Server uses a SQL Database. A diagram showing the overall architecture of the project is shown in Figure 24.

Each component is described in more detail in the following sections of this chapter.

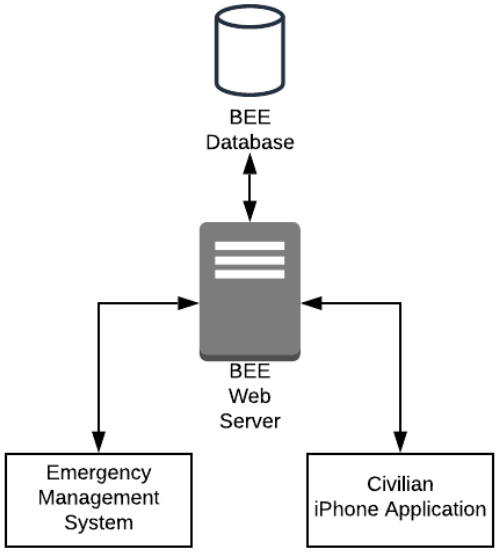


Figure 24: Server Interaction Diagram

4.2 Database

The BEE Database is a MySQL Database showed on Amazon Web Service's RDS platform. It is composed of seven different tables; which hold information about events (used interchangeably with evacuations), users (also called evacuees), and map related data.

The tables and relationships are shown below in Figure 25.

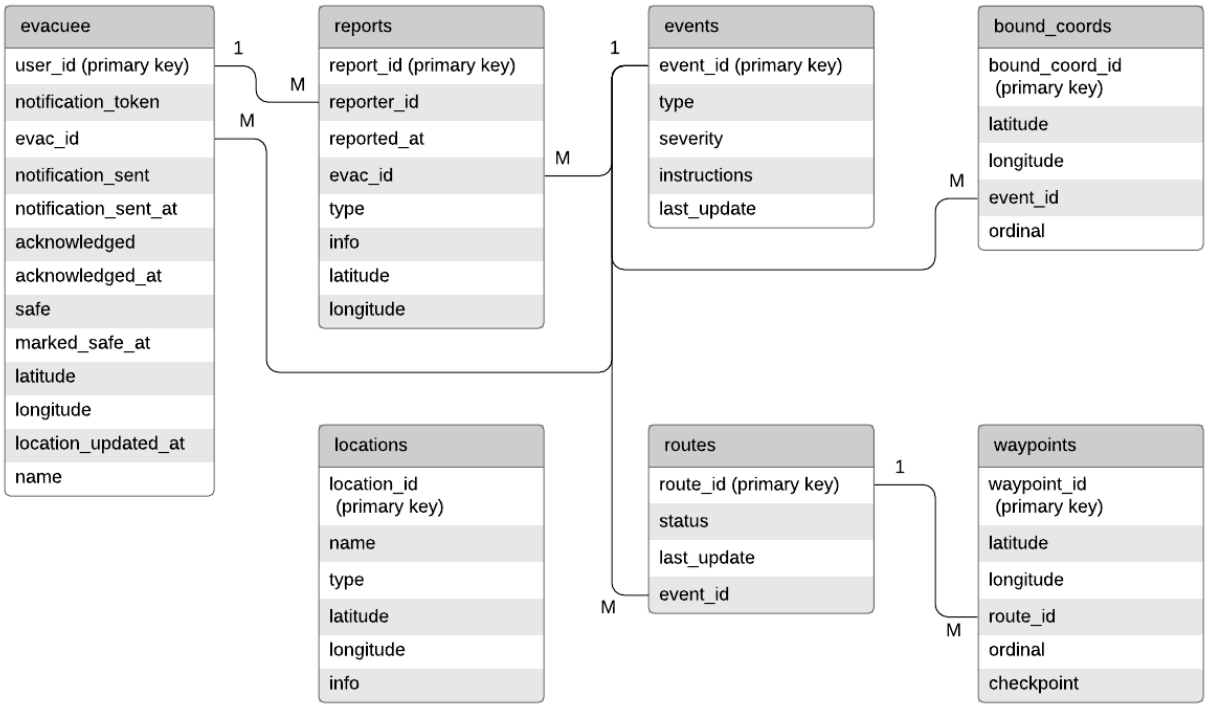


Figure 25: Database Architecture

4.2.1 evacuee Table

The evacuee table is filled with information about the civilian users of the iPhone application. Every evacuee is assigned a random unique user_id. If the user has decided to allow notifications when they installed the application, a device token (generated by their device specifically for this application) is stored in notification_token. If they are

included in the area assigned to an event, that event (or evac) identification number is assigned to the evacuee to link the evacuee to the event.

There are six columns in the evacuee table for evacuation related information, of which two are updated by the Emergency Management System and the rest are updated by the iPhone application. If a push notification is sent by the Emergency Manager and the Apple Push Notification (APN) service accepts the command to send the notification, notification_sent would be updated to true and the notification_sent_at field would be updated with the current time. If the evacuee opens the application during an event/evacuation, they will be shown instructions and must acknowledge they received the alert. Once the evacuee acknowledges the alert, acknowledged will be set to true and acknowledged at will be updated with the current time. Finally, if the evacuee completes navigation out of the evacuation zone, they will be marked safe and marked_safe_at will be updated with the current time.

There are three columns in the evacuee table for current location, and the last time their location was updated. Finally, there is the "name" column that will populate with the name the evacuee gave when installing the application, but only during evacuations.

4.2.2 reports Table

To store reports from evacuees, the reports table was created. This table holds a random unique identifier (report_id), the evacuee's user_id (held in reporter_id), the time the report was made, the evac_id assigned to the evacuee, the type of report (active flames, blocked road, etc.), information the evacuee provided when making the report, and the location of the report.

4.2.3 events Table

The table that holds the most important information is the events table. The events table holds a random unique identifier (event_id), the severity of the event (whether it is an evacuation order, evacuation warning, or none), any instructions the Emergency Manager has given, and the last time the event was updated. Information about the area the event is related to is stored in the bound_coords table (Section 4.2.4), and evacuation routes are stored in the routes table (Section 4.2.5).

4.2.4 bound_coords Table

The bound_coords table holds information about boundary points for the evacuation zone. Each vertex of the boundary of the evacuation zone will have an entry in this table. Each will have a random unique identification number (bound_coord_id), their location, the event_id they are related to, and the order they were defined (stored in ordinal). In practice, this will give points on a map to connect to make an outline of the evacuation zone.

4.2.5 routes Table

Each evacuation may have many evacuation routes, and this is reflected in the design of this table. Each route is stored in this table, which will hold a random unique identifier (route_id), the status of the route (open, congested, closed), the last time the route was updated, and the evacuation it is tied to (event_id). The exact location of the route is held in the waypoints table (Section 4.2.6).

4.2.6 waypoints Table

Two types of location points are held in the waypoints table. The first type is what is called a "checkpoint". These "checkpoints" are locations defined on the map by the Emergency Manager. The other points, called "waypoints" are intermediate locations returned by the Mapbox API when requesting routing between the checkpoints. The "checkpoints" will have true in the checkpoint column, and the "waypoints" will have false.

The other data held in each row of this table will be a random unique identifier (waypoint_id), the location of the waypoint/checkpoint, the position in the route the waypoint appears (ordinal) and the route_id of the route this waypoint is a part of.

4.2.7 locations Table

The only table that is a standalone table (no relationships to other tables) is the locations table as locations are not tied to events or users. This table holds data for each location in its own row. Each row contains a random unique identifier (location_id), the name of the location, the type of location (police station, hospital, etc.), the location of the location, and any information the Emergency Manager wanted to include about this location.

4.3 Server

The server is a Java process running in an Amazon Elastic Beanstalk environment. The Elastic Beanstalk environment allows the web service to get configuration information about the Amazon RDS database quickly and easily.

The server is composed of two main parts: the database module and the web service. In all, the server is comprised of over 1700 lines of code split between the database module, the server module, and data types shared between the two modules.

4.3.1 Database Module

The database module is where all of the interactions with the RDS database, described in Section 4.2 above, happens. This module first creates a remote connection to the RDS database by getting the configuration from the Elastic Beanstalk environment. Once the connection is created and verified, it will check to see if the tables described in Section 4.2 are present. If the tables are not present, it will generate the SQL statements required to create the tables and will execute them on the RDS database.

This module also holds generic functions to update (or add) and delete an entry in a table. These functions take in two parameters, the table name in a String format and a Map of String (column name) to Object (value).

Finally, this module holds specific functions to retrieve data from the database including events, locations, routes, evacuees, and much more.

4.3.2 Web Service

The web service module is where the logic for interfacing with the two user endpoints. This module contains the API to get and set information from the database.

Functions that can be called by the Emergency Management System include getting a list of evacuees; get, add, and remove events; get, add, and remove routes; get, add, and remove locations; get and remove reports; and push notifications to evacuees.

Functions that can be called by the BEE iPhone Application include get events for current user, get locations near current user, acknowledge evacuation, update location before and during evacuation, mark current user safe, and report a hazard.

4.4 Civilian iPhone Application

The civilian iPhone application is an application written in Swift for the iOS operating system. This application will work on any current iPhone and iPad with the latest version of iOS installed.

The application is functionally split into three parts: navigation, server interactions, and map interactions.

The navigation portion takes the information returned from the API in the server interactions module and passes it to the Mapbox API (more on this framework in Section 4.4.1). Once it gets a navigable route back from the API, it is handed over to the Mapbox framework to be ready for the user to click “Navigate”.

The server interactions module interacts with the server (which is described in the previous section). This means that this module will send updates to location data and user data, send reports, and receive data from the Emergency Management System about evacuations.

The map interactions module takes all of the data from the server interactions module and displays it on a map. This will include evacuation zones, routes, and emergency locations. It will also use the built in iOS alert to display information sent directly from the emergency manager in the form of instructions.

In all, the BEE iPhone application is composed of about 1100 lines of code written in Swift (the programming language designed by Apple for iOS development), split between 8 files. It also includes 11 different icons that can be displayed on the map to identify different locations.

4.4.1 Mapping and Navigation Framework – Mapbox for iOS

The first thing that was decided on was which mapping framework to use in the iPhone application. Naturally Apple Maps was the first choice, as it is Apple's own implementation. Apple Maps has tighter integration with other Apple applications, such as the calendar [18], and it can be argued that its UI is less obtrusive. Unfortunately, one of the requirements of this application is that it must do turn-by-turn navigation within the application itself and Apple Maps does not allow for that, it only allows for another application to launch navigation within the native Apple Maps application itself.

The next framework to be considered was the Google Maps framework. This is the natural second choice, as Google Maps has by far the most market share with 67% [19] (compared with second place, Waze, with 12%). The larger market share would mean more users would be comfortable with and know how to use the navigation. Unfortunately, Google Maps also doesn't allow for in application turn-by-turn navigation.

The final framework to be considered, and ultimately chosen, was Mapbox. Mapbox is a free to use, customizable mapping framework. According to their website, "[Mapbox] provide[s] building blocks to add location features like maps, search, and navigation to any experience" [20].

This framework met the requirement that the application itself could launch turn-by-turn navigation. It also meets the requirements, like the other mapping frameworks did, to allow the application to draw locations, zones, routes, and other information on the map.

The framework easily allows a route to be passed to it to draw on the map and to easily start navigation. It also allows the application to draw and display information on the map with only a few configuration items. In short, the Mapbox for iOS framework was exactly what this application needed.

4.5 Emergency Management System

The Emergency Management System is a web application written in JavaScript to run on Google Chrome and other similar browsers. The user interface is described in Section 3.1.3. This section will focus on the technical aspects of the Emergency Management System.

The Emergency Management System is composed of two parts: The Mapbox Framework (described in Section 4.5.1 below), and the business logic. The business logic aspect of the Emergency Management System handles all the API calls to the BEE Server (see Section 4.3.2).

It also stores all of the data it needs locally in data types that mirror the database tables. These datatypes were written in the Constructor Pattern, a classic object oriented pattern for JavaScript [21]. This pattern generates objects within JavaScript, mimicking classes and objects in other object oriented languages (this pattern still sees quite a bit of use even though JavaScript added classes in 2015).

For example, the event datatype has event_id, type, severity, instructions and last_update. It also stores an array of bound_coords, to mimic the relationship between the tables, and an array of routes. Finally, each datatype object has two functions: one to transmit data to the API and one to request the server to remove the object from the database before it deletes itself.

All of the data that gets stored into these special objects comes from either the API/Database or from the Mapbox Framework for JavaScript. In all, the Emergency Management system is composed of almost 1400 lines of code in 12 files. It additionally includes 25 waypoint icons, 11 location icons, 10 button icons, and 4 evacuee icons.

4.5.1 Mapping Framework – Mapbox for JavaScript

The mapping framework, Mapbox for JavaScript, was chosen due to the civilian iPhone application using the Mapbox for iOS framework. It was decided that it would be easiest to stay with the same provider for multiple reasons, including ease to learn (as there was experience already with Mapbox APIs) and same format of results from API calls.

The Mapbox for JavaScript framework includes functionality built in for drawing polygons, which was used for creating evacuation zones; drawing polylines, which was used for creating evacuation routes; and creating pinned locations, which was used for creating emergency management designated locations.

The data returned from this functionality was used to create the datatypes that was discussed in Section 4.5 above. After the datatypes were created and sent to the BEE

Server API, then it was added to the map. When the emergency manager clicks on an object on the map, an easy to configure popup is automatically shown.

In addition to the tools that were built in, the Mapbox framework makes it easy to add new tools to the toolbar that is right on the map. In this project, a button to show evacuees on the map and a button to show evacuee reports were added. All it takes to add a new tool is an icon configuration object and a function to attach to the onClick configuration item.

The choice of Mapbox for JavaScript as the mapping framework for the Emergency Management System was a good one, it allowed for easy implementation and allowed for the focus to be on the functionality of the EMS rather than on how to implement a mapping framework.

Chapter 5 Comparison with Related Works

5.1 CodeRED

CodeRED is the emergency alert system that Butte County, CA uses, and had issues with the morning of November 8th, 2018 when the Camp Fire ripped through Paradise, CA and killed 85 people.

According to OnSolve, the organization that created and maintains CodeRED, this system can disseminate information through "landline, cellular, email, text, social media, mobile applications, IPAWS, RSS feeds, and more" [21].

From an emergency management point of view, CodeRED is a more capable system than BEE is in its current form. CodeRED allows for severe weather alerts, missing person alerts, and shelter-in-place alerts in addition to evacuation alerts.

However, from a civilian point of view BEE is an easier to use system that provides more information. With CodeRED, you must sign up for an account, providing your location, your phone number, your email address, and sign up for alert subscriptions. With BEE, all you have to do is install the application, allow notifications and location tracking, and enter your name. There is no lengthy sign up process, and you get alerts based on your current location rather than your location at sign up. In addition to the easier set up process, CodeRED does not send out evacuation routes and it does not provide turn-by-turn navigation. CodeRed currently has a 2.6 out of 5-star rating on the Apple App Store [22].

5.2 Emergency Alert System (EAS) and Wireless Emergency Alert (WEA) System

The only federally mandated system for evacuation and emergency alerts, the EAS and WEA systems are available to everyone, regardless of device or application install. The EAS is the emergency system that plays alerts on many different mediums, such as radio and television. The WEA is a slimmed down version of the EAS that sends alerts to cell phones.

There are many ways for an emergency manager to send out information through the EAS and WEA, including CodeRED using the IPAWS system or contacting their local National Weather Service office.

The EAS can send out information about many different emergencies, including severe weather and Amber Alerts. In fact, there are about 80 different alert codes that are present in the EAS standard [23].

There are, however, many draw backs to these systems. In the case of EAS, there is no visual component, with the exception of text crawling on the bottom of a television screen. This is unhelpful during an evacuation, as any information provided would have to be remembered. In the case of WEA, only certain alerts get sent to cell phones. Alerts include Tsunami Warnings, Tornado Warnings or Emergencies, Extreme Wind Warnings, Hurricane or Typhoon Warnings, Storm Surge Warnings, Snow Squall Warnings, Flash Flood Warnings, or Dust Storm Warnings [24].

5.3 Google Maps

In Sumer 2019, Google Maps will roll out an update to its "SOS Alerts" feature that includes features similar to BEE; including emergency updates during navigation, visual representations of emergencies on the map, and user reporting of road obstructions. [25]

Since these updates have not rolled out at the time of publication, it can be hard to compare these updates to BEE. However, there are some main differences between the two applications. First, Google Maps does not automatically route you out of the evacuation zone. This is an important part of BEE; the automatic navigation reduces the stress on the evacuee by removing the guess work of figuring out where to go. Second, Google Maps does not prominently show evacuation zones or evacuation instructions like BEE does. This is also an important feature of BEE because the focus of BEE is to get information to people impacted by the evacuation as fast and as clear as possible.

Google Maps has other main functionality that is useful, though. It will have better native navigation, as the navigation computations is handled by Google itself rather than through a third-party framework.

Given that Google Maps is installed on virtually every mobile device, its market share is unimaginable by a small-time developer. In fact, the best-case outcome for BEE (discussed more thoroughly in the Future Work section) would be for the features to be incorporated into an integrated maps application like Google Maps or Apple Maps for the sole reason of market share.

Chapter 6 User Study

6.1 Institutional Review Board Approval

This user study was reviewed by the University of Nevada, Reno Institutional Review Board (IRB) and was approved under the project title "[1379379-2] BEE: Bettering Emergency Evacuations". This means this user study meets the requirements of the Code of Federal Regulations on the Protection of Human Subjects (45 CFR 46.101).

6.2 Experiment Setup

To collect thoughts and suggestions from potential users of BEE, an online survey was created. This survey would require the taker to have evacuated in the past, so that an accurate comparison between current methods of disseminating information and BEE could be made.

Once the taker has verified that they had evacuated in the past, they would be asked to give a description of their evacuation, focusing on how they were told to leave. Then a set of six statements would be presented to the participant to rank how much they agree or disagree with (1 being "Strongly Disagree", 5 being "Strongly Agree", and an option for "N/A" if it didn't apply to their situation). The six statements were:

- I was notified in a timely manner.
- I knew what the emergency was.
- I knew where the emergency was.
- I was given useful information about what to do.

- I knew how to get where I was supposed to evacuate to.
- I knew for sure I was located in the evacuation zone.

After the section about the participant's evacuation, a page describing BEE was shown with a video of a mock evacuation within the application. This mock evacuation was based upon the Camp Fire, but the page asked the participant to pretend it was for their emergency. The survey then showed the same six statements (worded slightly differently from past tense to future tense) and asked the user to once again rate how much they agreed with each. Finally, the survey allowed the participants to give general feedback or suggestions about the application.

6.3 Results

Responses for the survey were collected over a 24-day period in December 2019 and January 2020. In the end 13 people participated in the study (15 including responses from participants that did not evacuate and did not qualify to take the survey). The following sections will go into detail about the breakdown of responses.

6.3.1 Participant's Evacuation

This section focuses on the participant's responses to the statements in relation to their evacuation in the past. Responses referenced common evacuations due to hurricanes (Irma in 2017 and Florence in 2018), and fires (Camp Fire in 2018, Cascade Fire in 2017, Napa Fires in 2018). There was also a response that was from a person who had to evacuate Yosemite during "The 100-Year Flood" in which the infrastructure was damaged so badly the respondent had to leave for over three months. This 1997 flood

was notable for how high the Merced River rose, it was at “a record of 23.4 feet, well above its 10-foot flood stage” [28].

A breakdown of responses is shown in Figure 26 (some questions had responses of “N/A” which is not shown in the chart).

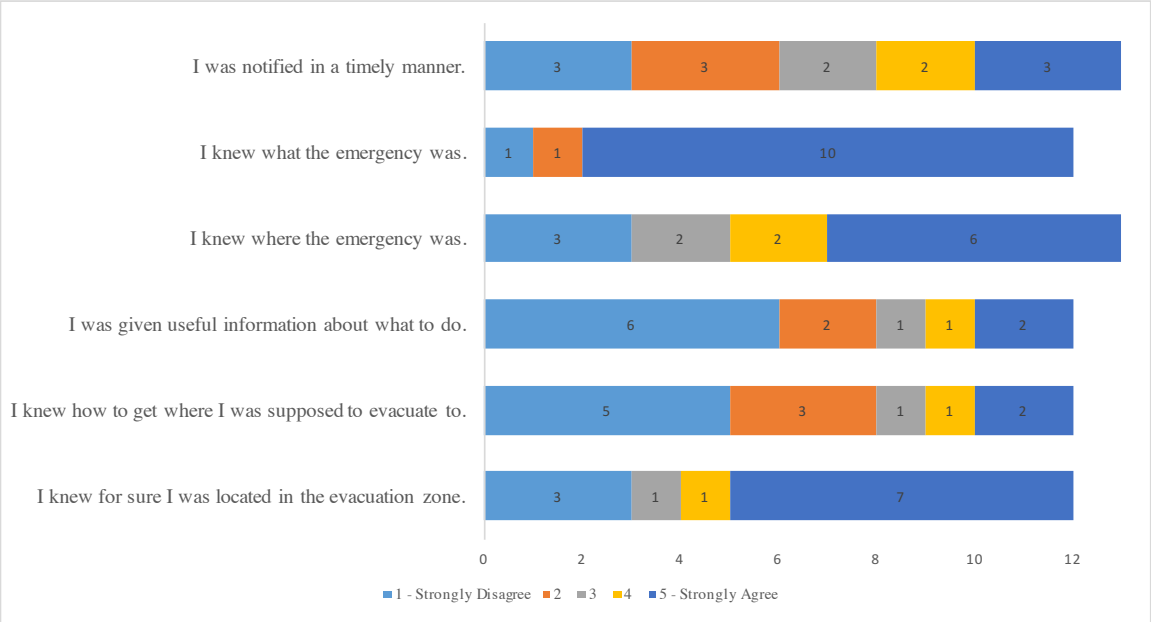


Figure 26: Visual Breakdown of Responses - Participant's Evacuation

Table 4: Statistics of Responses - Participant's Evacuation

Statement	Count	Average	Variance
I was notified in a timely manner.	13	3.23	2.36
I knew what the emergency was.	13	4.77	0.69
I knew where the emergency was.	13	3.92	2.24
I was given useful information about what to do.	12	2.50	2.64
I knew how to get where I was supposed to evacuate to.	12	2.58	2.45
I knew for sure I was located in the evacuation zone.	13	4.08	2.24

In Table 4, the aggregated statistics of the responses are shown. As the table shows, respondents slightly agreed on average on if they were notified in a timely manner (average of 3.23), they slightly disagreed on if they were given useful information and how to get to their evacuation point (average of 2.50 and 2.58, respectively), and strongly agreed that they knew what the emergency was, where the emergency was, and that they were in the evacuation zone (average of 4.77, 3.92, and 4.08, respectively). However, the variance of the responses was fairly high, meaning that the responses were all over the place; there was very little agreement between the respondents.

6.3.2 Mock Evacuation in BEE

This section focuses on the responses in relation to the mock evacuation in BEE that was shown to participants in the survey. A breakdown of these responses is shown in Figure 27 (again, some questions had responses of "N/A" which is not shown in the chart).

As seen in Figure 27, the majority of responses to the statements were positive (fours and fives, denoted by yellow and blue respectively). In Table 5, the aggregated statistics of the responses are shown. Participants, on average, strongly agreed with all statements. In addition to the average score given by the participants, the variance is fairly low, meaning that the participants gave similar responses to each other and the agreement was fairly universal.

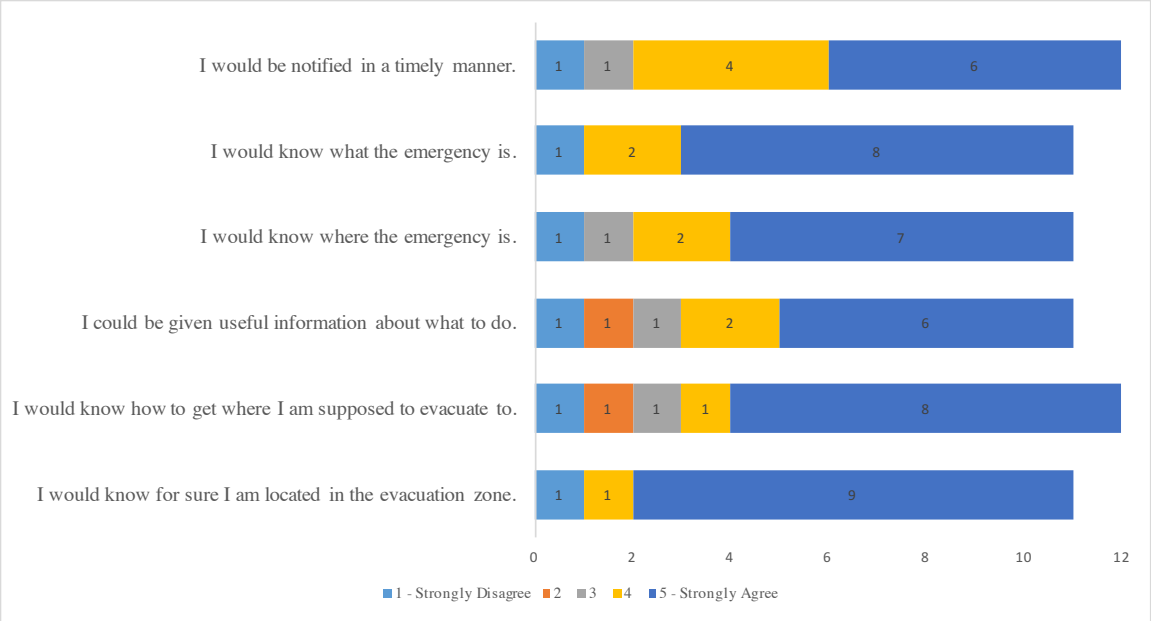


Figure 27: Visual Breakdown of Responses - Mock Evacuation

Table 5: Statistics of Responses - Mock Evacuation

Statement	Count	Average	Variance
I would be notified in a timely manner.	12	4.50	0.45
I would know what the emergency is.	12	4.83	0.15
I would know where the emergency is.	12	4.67	0.42
I could be given useful information about what to do.	12	4.42	0.99
I would know how to get where I am supposed to evacuate to.	12	4.50	1.00
I would know for sure I am located in the evacuation zone.	12	4.92	0.08

6.4 Discussion of Results

Looking at the results, it might seem clear that BEE scored higher than the existing methods of disseminating emergency evacuation information, however, the results might not be statistically significant. Since the survey used a Likert scale, a non-parametric test must be run to determine if the differences are statistically significant. Since the data is "within-subjects," meaning that the same person answered the same question about two different scenarios, and there are only two scenarios; a Wilcoxon Signed-Rank test was run [30].

The output of the Wilcoxon Signed-Rank test showed that the differences between scenarios for the following generic statements (worded slightly differently based upon which scenario was being judged, see Table 4 and Table 5 for specific wording) were statistically significant:

- "Get notification in a timely manner." ($z = -2.521$, $p < .05$)
- "Given useful information." ($z = -2.239$, $p < .05$)
- "Know how to evacuate." ($z = -2.687$, $p < .01$)

The differences between existing methods and BEE were not statistically significant; meaning that while there may still be differences between the methods, it cannot be said that the differences in the responses is due to the differences in methods itself.

6.5 User Comments

There were many different stories of evacuations that were submitted in the survey. Some were terrifying. Some showed the lack of communication. All showed

hardship. It is easy to forget the ultimate reason for a project, especially one for a class or a degree. Reading through the comments made me remember why I started this project and why I believe in it. Below are some excerpts from the responses (some are edited to fix typos and misspellings).

The following responses describe evacuations from Hurricane Irma in 2017:

> *In September of 2017, I was living in St. Petersburg, FL. […] Because I was aware that this was potentially a very serious situation (models show potential 25+ feet of storm surge into Pinellas and Hillsborough counties depending on the circumstances), I started contemplating evacuation possibilities. While I was looking for options, I was notified by my apartment complex that Evacuation Zones A and B in my county had been put under a mandatory evacuation order. […] I was given no instructions other than news of the mandatory evac order and address of shelters.*

> *We evacuated Hurricane Irma from Zone A in St. Petersburg FL, mandated evacuation. Not a lot of instructions other than to leave and that shelters were opening locally. We evacuated to Georgia, which ended up not being the smartest option. Gas was out across the area and we thought we left early enough. But we did make it to our destination south of Atlanta. Unfortunately, the ride home was harrowing with a rescue dog, and gas nowhere to be found, stores shut down, nothingness.*

The following reponses describe evacuations from the Camp Fire in 2018 and the Cascade Fire in 2017:

> *I evacuated from my home in Paradise, CA from a fire. I was trying to evacuate to Chico with no knowledge of how to get out. My mother called me early in the morning to tell me all of Paradise was in flames. I had to follow the people in front of me in hopes we were going the correct way.*

> *I was notified via a reverse 911 call. However, we were not given the exact location of the fire or the seriousness of it (like mentioned before, this wasn't our first evacuation notice because of a fire) or where to go/where a safe route out was. We finally left when a neighbor came back for her grandma and said a school was on fire. We went down Pentz Road towards Butte College and did not come into contact with traffic until we were out of the evacuation zone. However, my neighbor went towards Pearson Rd to meet with Skyway, and wound up having to abandon her car to escape flames. If she had gone the way me and my family did, she would not had to have left her car to burn.*

> *I evacuated from the Cascade Fire in October 2017. We only knew about the fire because my neighbor saw it and called to alert my family and I. There was no official evacuation until hours later, and by that time my house was already engulfed in flames. There was no official evacuation area until well into the next morning (the fire started around 10 in the evening).*

These responses show how frantic, terrifying, and dangerous evacuations can be. There are other responses that said the participants of the survey were paying attention to the news, or in contact with schools because of their children, or they received the evacuation order in a timely manner. The current ways of disseminating information are designed to work; and in many cases, they work flawlessly. However, when they don't provide all the information needed or they break down, it's easy to see the danger that is caused.

Every person that took the survey also provided some comments about the application itself. Some responses are below:

> *Something to consider - in my case and in many other emergencies, people who live in rural areas with poor phone and internet service (many times none) and the elderly who may not be on the internet get the word last. A program like this is important but still would not have saved my life.*

This comment is something to consider, however there is nothing an application can do about this. However, since an Emergency Manager can put anything in the instructions of the evacuation, they can choose to mention that people who receive the alert should make sure their neighbors know also.

> *It's one thing to know the routes on a map, it's another to try and navigate them during a crisis. If there was a live update somehow of traffic at the same time, that may help people determine which routes are safer/quicker. Like mentioned in my other entry, I went a route that went around the disaster, whereas my neighbor chose a different route (one that we used to use all the time to get to Chico), but this wound up putting her into a life-threatening situation.*

This situation is something that I thought of when developing this application. There is no indication of this from the application user point of view, but the Emergency Manager can define multiple evacuation routes, and the application will automatically choose which one to use based upon the time it takes to get to the end (taking into account traffic). The Emergency Manager can also mark certain routes closed or congested which tells the application to use other routes, if available.

> *You have a whole lot of text, really important text, when you open the app. Lots of people are just going to skip over that, especially in a stressful situation. I'm not sure how you could break that out into byte sized pieces, but I would really rather you try for something like that.*

The point made in this comment is completely valid. In the future instructions could be split up into different chunks as mentioned in the comment. Possibilities could include having instructions for preparing to leave pop up when you open the application, then pop up instructions that relate to actually leaving (such as driving instructions), and then instructions of what to do when you are finally out of the danger zone.

Chapter 7 Conclusion and Future Work

7.1 Conclusions

Bettering Emergency Evacuations achieves its goal, to provide easy information transfer from emergency managers to civilians in harm's way. It displays relevant information in an easy to understand format and tells people evacuating exactly how to get out of the path of the disaster and exactly what to remember while evacuating. Not only can it help save lives, but it can help people remember their pets, their medications, their important documents, or whatever else emergency managers remind them to remember.

As discussed in Chapter 5, there are many other ways to get this information. Navigation can be done within Google Maps, but it doesn't tell you where to navigate to or how to get out of the way of the disaster. CodeRED and WEA can get information out, but they are limited. WEA can only say short messages. CodeRED can send out user defined messages, but it doesn't allow for easy navigation like BEE does.

However, Google Maps and WEA have a large reach (especially WEA since every single modern cell phone can get WEA alerts). CodeRED serves over 10,000 communities [21]. The fact that these options have such a large reach means any information that gets sent out will be read by more people. Hopefully these options will add BEE's helpful features to their products in the future.

7.2 Future Work

Bettering Emergency Evacuations doesn't have to stop at evacuations, even though it is part of the catchy abbreviation. It could change to BEST, "Bettering Emergency Situations… Today!"

All joking aside, there are many more emergency situations that getting complete information out is important. BEE can be updated to handle shelter-in-place orders, or severe weather warnings with instructions on what to do (rather than "Check media." like WEA messages suggest).

Evacuations were focused on because of the urgency of getting a large amount of people out of harm's way. Severe weather doesn't usually endanger as many people all at once (with the exception of Hurricanes, which would have days of warning, not minutes).

However, in its current form, BEE isn't a commercially viable product. That would require a sales staff, an engineering team, and much more. Currently BEE is more of a proof of concept or a prototype, albeit a very fleshed out and functioning prototype. BEE is exactly what it was designed to do, create a dialogue and show that evacuations can be done better. Best thing is for the major smartphone operating system makers (Google and Apple) to pick this up and pull it into their native mapping applications. We need every single person to have access to this information, and it will save lives.

Bibliography

[1] K. Alexander, S. Ravani and E. Allday, "Camp Fire is most destructive wildfire in California history: 9 dead, 6,713 structures incinerated," *San Francisco Chronicle,* 10 November 2018.

[2] E. Sernoffsky, "Camp Fire: Death toll grows to 71, missing list grows to 1,011," 16 November 2018. [Online]. Available: https://www.sfgate.com/california-wildfires/article/Camp-Fire-Grows-to-146-000-acres-50-percent-13400404.php?t=7d3541f7c6. [Accessed 17 November 2018].

[3] A. McBride, M. Gutierrez and N. Asimov, "29 dead in Camp Fire as firefighters make gains on Butte County blaze," 13 November 2018. [Online]. Available: https://www.sfgate.com/california-wildfires/article/Butte-County-fire-only-grows-slightly-as-13381900.php. [Accessed 17 November 2018].

[4] Cal Fire, "Camp Fire Incident Information," 2018. [Online]. Available: http://www.fire.ca.gov/current_incidents/incidentdetails/Index/2277. [Accessed 17 November 2018].

[5] J. P. Sulek and A. Sciacca, "'This is when I die' . . . unforgettable tales of escape from the Camp Fire," *Mercury News,* 10 November 2018.

[6] "Red Flag Warning Issued By NWS Sacramento," 9 November 2018. [Online]. Available: https://mesonet.agron.iastate.edu/vtec/#2018-O-NEW-KSTO-FW-W-0013/USCOMP-N0Q-201811072120. [Accessed 17 November 2018].

[7] "Annual Estimates of the Resident Population: April 1, 2010 to July 1, 2017," May 2018. [Online]. Available: https://factfinder.census.gov/faces/tableservices/jsf/pages/productview.xhtml?src=bkmk. [Accessed 17 November 2018].

[8] L. M. Krieger and D. Debolt, "Camp Fire: Paradise residents say they received no mass cellphone alerts to evacuate, or to warn of fires," *Mercury News,* 13 November 2018.

[9] "NHC Graphical Outlook Archive," 26 August 2017. [Online]. Available: https://www.nhc.noaa.gov/archive/xgtwo/gtwo_archive.php?current_issuance=2017082623 39&basin=atl&fdays=5. [Accessed 18 November 2018].

[10] "Irma Slams the Keys; Florida's Southwest Coast at Risk of Major Storm Surge," Weather Underground, 10 September 2017. [Online]. Available: https://www.wunderground.com/cat6/irma-slams-keys-floridas-southwest-coast-risk-major-storm-surge. [Accessed 18 November 2018].

[11] "Monroe County Emergency Directive," 5 September 2017. [Online]. Available: https://www.monroecounty-fl.gov/DocumentCenter/View/14743/2017-Hurricane-Irma-Mandatory-Evacuation-for-Residents-of-Monroe-County-. [Accessed 18 November 2018].

[12] J. P. Cangialosi, A. S. Latto and R. Berg, "National Hurricane Center Tropical Cyclone Report: Hurricane Irma," 30 June 2018. [Online]. Available:

https://www.nhc.noaa.gov/data/tcr/AL112017_Irma.pdf. [Accessed 18 November 2018].

[13] N. Madigan, "How the Florida Keys Are Coming Back to Life," *New York Times,* 16 October 2017.

[14] "Lessons From Hurricane Irma: When To Evacuate And When To Shelter In Place," 1 June 2018. [Online]. Available: https://www.npr.org/2018/06/01/615293318/lessons-from-hurricane-irma-when-to-evacuate-and-when-to-shelter-in-place. [Accessed 18 November 2018].

[15] L. Lazo, "Florida Gov. Rick Scott orders highway patrol escorts for fuel trucks to get gas to stations," *Washington Post,* 7 September 2017.

[16] "High Wind Warning Issued By NWS Reno," 17 November 2011. [Online]. Available: https://mesonet.agron.iastate.edu/vtec/#2011-O-NEW-KREV-HW-W-0006/USCOMP-N0Q-201111180600. [Accessed 19 November 2018].

[17] "Revisiting the Caughlin Fire," October 2012. [Online]. Available: http://www.livingwithfire.info/wp-content/uploads/2012/10/Caughlin-Fire-2011.pdf. [Accessed 19 November 2018].

[18] I. Sommerville, Software Engineering, 10th Edition ed., Pearson Education, 2016.

[19] S. Russell and P. Norvig, Atrificial Intelligence: A Modern Approach, Third ed., Boston: Pearson, 2016.

[20] D. Oragui, "Apple Maps vs. Google Maps: Which Is Better?," 12 September 2018. [Online]. Available: https://themanifest.com/app-development/apple-maps-vs-google-maps-which-better. [Accessed 11 November 2019].

[21] R. Panko, "The Popularity of Google Maps: Trends in Navigation Apps in 2018," 10 July 2018. [Online]. Available: https://themanifest.com/app-development/popularity-google-maps-trends-navigation-apps-2018. [Accessed 11 November 2019].

[22] "About Mapbox," [Online]. Available: mapbox.com/about/company. [Accessed 11 November 2019].

[23] A. Osmani, "JavaScript Design Patterns," in *Learning JavaScript Design Patterns*, Sebastopol, CA: O'Reilly Media, Inc, 2012.

[24] "Complete Emergency Notification and Alerting with CodeRED," [Online]. Available: https://www.onsolve.com/solutions/products/codered/. [Accessed 11 November 2019].

[25] "CodeRED Mobile Alert," Apple, [Online]. Available: https://apps.apple.com/us/app/codered-mobile-alert/id468404991. [Accessed 11 November 2019].

[26] "Specific Area Message Encoding," [Online]. Available: https://en.wikipedia.org/wiki/Specific_Area_Message_Encoding. [Accessed 11 November 2019].

[27] "Current NWS WEA Messages," [Online]. Available: https://www.weather.gov/images/wrn/2019_WEA_Messages_Originated_by_NWS.png. [Accessed 11 November 2019].

[28] Google Maps, "Updates in Maps and Search to help during times of crisis," 6 June 2019. [Online]. Available: https://www.blog.google/products/maps/updates-maps-and-search-help-during-times-crisis/. [Accessed 11 June 2019].

[29] K. Alexander, "Memories of disastrous 1997 flood haunt Yosemite," *San Francisco Chronicle,* 6 January 2017.

[30] I. S. Mackenzie, Human-Computer Interaction: An Empirical Research Perspective, Waltham, MA: Morgan Kaufmann, 2013.