Project Hermes

Formerly known as CallForCode

Final Report

Project 40

Advisors

Dr. Diane Rover

Dr. Nicholas Fila

Team

David Boschwitz - Team Lead

Caleb Nash - Lead Frontend

Logan Fladung - Subject Matter Expert

Justin Kaufer - User Research & Test Lead

Austin Keen - Graphic Design

Robert Schedler - Backend & Networking Lead

Contact

callforcode@iastate.edu

sdmay19-40.sd.ece.iastate.edu

Table of Contents

List of Figures

List of Tables

List of Definitions

.NET: The Microsoft utility library used commonly with C#

Xamarin: A C#, cross-platform mobile development toolkit

XAML: C# UI Templates based on/similar to XML/HTML

ECE/ECPE: ISU Department of Electrical, Computer, and Software Engineering

MVVM: Model-View-ViewModel, a frontend design pattern commonly implemented when using XAML and C#

1. Executive Summary

1.1 Acknowledgments

We would like to express our appreciation to Dr. Diane Rover for her valuable and constructive suggestions during every meeting. Her willingness to squeeze us into her already crammed schedule has been very much appreciated. We would also like to thank Dr. Nicholas Fila for his continual work with our team as his knowledge of user research was extremely important to many of our functionalities.

This project was inspired by IBM's Call for Code challenge, which is "a rallying cry to developers to use their skills and mastery of the latest technologies to drive positive and long-lasting change across the world with their code." Unfortunately, the timeline of the class did not allow for our team to be involved, but this challenge helped stem the idea for this project.

1.2 Problem statement

With natural disasters becoming increasingly more common, ways to provide relief to these victims are becoming more and more necessary. The goal of Call for Code project is to design a tool to improve preparedness for natural disasters and relief when they hit in order to safeguard the health and wellbeing of communities, as well as provide these basic services needed without an internet connection by storing extra data offline and leveraging alternative connectivity methods.

1.3 overview

Our solution to the defined problem is creating a tool box of communication features to allow disaster victims to get important survival updates, and allow users to make inquiries to Red Cross staff about any question related to survival that they may have. We have broken down our app into 3 main features: Chat, News and Maps. Each of these three features provides users with knowledge related to their situation that will aid them with the help that they need.

2. Requirement Specification

2.1 Functional Requirements

Below are functional requirements that we believe are vital for our applications success. From our extensive user research in semester 1 and discussions with the American Red Cross, we determined that the requirements defined below are where we believe the Red Cross and improve their interactions with the communities struck with a natural disaster.

FR1. Provide users with locations of relief support, such as medical assistance, food, shelter, etc.

- This focuses on providing help to those that have means of transportation to allow them access to crucial items needed for survival.

FR2. Mark roads that are unsafe and not allow routes through them

- Hazardous roads are another safety measure put in place to ensure our users are not put in a position where they are in danger or have to reroute themselves

FR3. Track users and reroute paths based on traffic along popular routes

- This helps the users efficiently get to their destination which is beneficial for time sensitive situations.

FR4. Allow users to communicate with each other via chat messaging within the application

FR5. Function even when internet and cellular connections are unavailable through an ad hoc network process

- The ad hoc network allows information to ping off of other users to transmit the most up to date data throughout the application

FR6. Work across iOS and Android platforms

- The application shall be designed to function cross platform

FR7. Have user status' for different kinds of users, including relief worker, ordinary user, and admin

- This will allow the functionality and purpose of the app to differentiate based on the user's intent

FR8. Provide a constant news update based off locational settings

2.2 Non-functional Requirements

Below we defined several non-functional requirements that we believe, through our user research, are key to creating a successful app that can be used by anyone anywhere. We want to make sure our application solution can meet the immediate needs of our users. For example we noticed that the worst of natural disasters happen in third world countries, so we need to make sure our application is accessible to all users in the world.

NFR1. Security

- Secure sign-in of authenticated users cannot be duped
- 100% of messages will be encrypted

NFR2. Performance

- It is important that users experience short response time, within 3 seconds from request
- High availability of our system with above 99% uptime

NFR3. Accessibility

- Ease of usability, the interface is easy to learn and navigate
- Application should work with or without network connectivity
- Support for languages other than english.

NFR4. Scalability

- The ability to maintain performance and operation from not only concentrated local areas but to more distributed geographic regions
- Ability to add new features and deploy rapidly

3. System Design & Development

3.1 Design Plan

We built this application to be very easy to navigate. We expect our users to be a very diverse group of individuals, varying in age, gender, ethnicity, and technology literacy. The application should be universally understandable, so that in the time of crisis these people will be guided toward relief as fast as possible.

To make the app easier to use by people of different cultures, we will utilize Visual Studio's built in resx culture files. These files enable the app to be easily translated. This is achieved by auto-generating a class from the provided key/value strings for each translation, and mapping it to different cultures. For the project completion we will only create the en-us culture, but in the future it would be very easy to create additional cultures as part of a globalization effort.

The app will utilize a modified Model-ViewModel-Model approach, ours differs from MVVM as we use a controller to handle all logic surrounding model files. We will create components for each of our capabilities (ie Chat, News, Maps). The capabilities will contain all of the model classes and a controller that contains all of the logic necessary for the user to interact with the network and database. This creates more modular logic that can be used by all applications. See component diagram (Figure 3.1.1) below for further visualization.

The database will utilize sqlite.net, an implementation which creates tables out of model files. Paired with Newtonsoft.Json.Net as our serializer, we are able to efficiently pass type objects (of classes) between networking and database components to create tables, and serialize and deserialize model files. This enables us to quickly change items as all platforms, including the headless server, use these model files.

This design is strong because of its user focused design. We worked closely with Dr. Nick Fila, a postdoc researcher at ISU focusing on user focused design, to dive deeper into our end-user's situation and create an app that will fulfill their ideal functionality. We used this process to recognize early on that the CallForCode's focus on cloud was flawed, and that we needed to establish communication methods that would be more suitable for victims of large scale disasters that may not have access to infrastructure.

Our connectivity is the most unique part of the app as it does not require a persistent connection. The app is meant to be usable with and without an internet connection, so instead of only getting necessary information from the database, it syncs and stores the entire database locally. This allows for ad-hoc synchronization. For example: a relief worker could sync with the central/cloud server at an aid station using WiFi/Cellular, then go out on rounds and sync with victims via bluetooth, allowing them to send and receive data.

As for systems level, our design focuses on detaching functionality from connectivity. Please see the figure below (2.5.1) We do this by using connectivity only for syncing the database for the app with the database in the cloud, enabling all of the data to be available offline. This fulfills our

non-functional requirement of NFR3 “Accessibility of data to all those who need it”, and “application should work with or without network connectivity”. The rest of the design focuses on ease of enabling new features and capabilities by using abstraction to make interaction generic and reusable. This fulfilled NFR4 “ability to rapidly deploy updates as needed”.

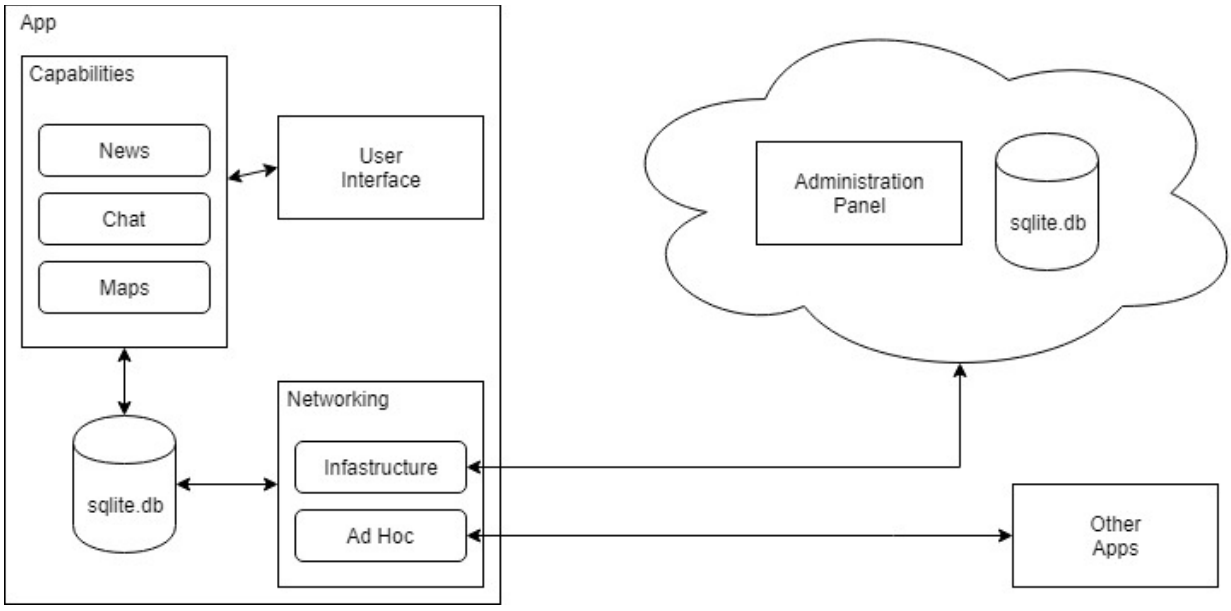

Figure 3.1.1 - Application Structure

3.2 System Constraints

Our user research was overseen and critiqued by Dr. Nick Fila. We completed full sprints of user research and documentation to ensure that our product is the correct solution for our user’s problems.

Our application requires the use of WPAN standards since we use Bluetooth technology. This constraint is necessary but easily overcome by the fact most modern smartphones have Bluetooth capability.

We were mindful to make sure we complied with the recently announced EU data collection laws. Some information users generate using our app could be very personal so we had to we protect them to the best of our ability. With these ethical ideas in mind we also tried to keep our development practices up to the industry standard.

4. Testing, Validation, and Evaluation

4.1 Unit and integration testing

Unit testing allows developers to test their code down to the smallest units possible to assure they are working as expected before trying to implement several pieces together. As a team there will be a standard format for unit testing of arrange, act, and assert to keep tests consistent and easily observed by others. This ensures the correctness and functionality to catch any mistakes before they create a larger effect. All data structures were unit tested to ensure the smallest objects of our code were working as intended. Each capability team then wrote various tests as seen fit to detect defects within the application.

Integration tests were made throughout the development process, along with the unit tests. The purpose of integration testing is to test faults on the interaction between specific functions. Since all of the capabilities are independent it is important that the interactions between them do not affect the integrity of the code. Success is determined by the test correctly executing the function that is designed by the test. The relationships and integration were tested by the team to maintain correct functionality, and when defects were found, the corresponding functionality teams handled them immediately.

4.2 User Acceptance Testing

These tests were not written in code, but were researched on the user experience of the application. Usability by all types of users was extremely crucial to the application. A variety of procedures established by the team or developer responsible for the feature was established and tested by various populations. The test engineer established a survey to ensure that a variety of demographic and skill sets were being tested. All feedback received from usability tests was accounted for and used to help redesign or validate the specific feature. Success was determined by all populations being able to successfully complete the tasks established by the team. (e.g. a user should be able to create a their own user then go in and set their location settings).

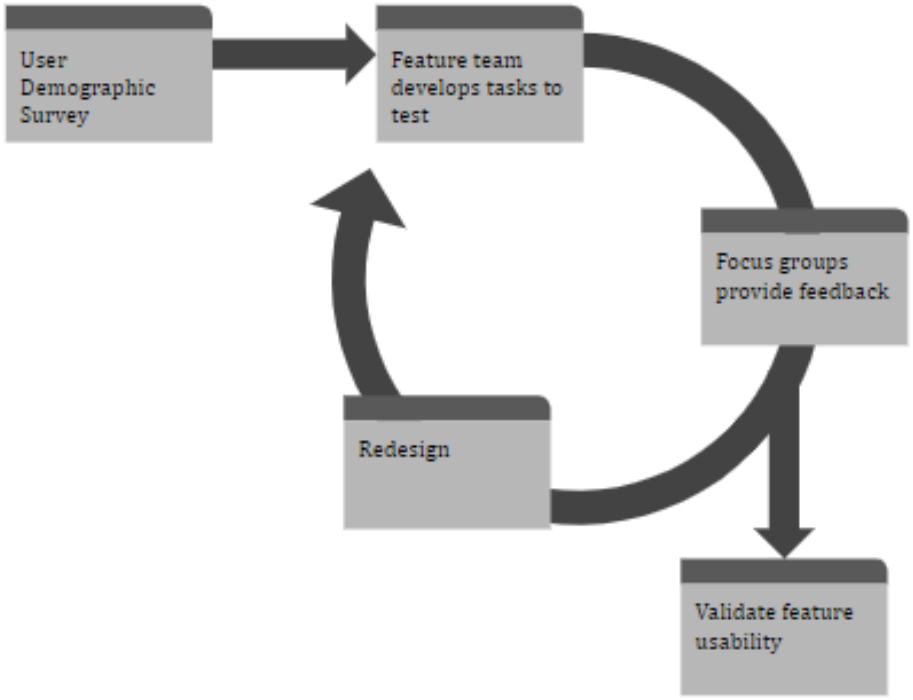

Figure 4.2.1 Iterative User Experience Design Process

A usability testing model was used to first determine the demographic being tested. Then after requesting certain tasks within the application, the users gave specific feedback on their experience performing the function, as well as any feedback not pertaining to the specific task.

5. Project and Risk Management

5.1 Task decomposition & Roles and responsibilities

The roles for each team member were set up in a liaison-style format, where the team member assigned to that lead was responsible for the overarching tasks. Our actual task for this project were split up to members based on features such as: maps, news feed, and messaging.

Team Lead

The Team Lead was in charge of maintaining communication between advisors and team members in regards to overall team production and iterative checkpoints. The Team Lead was also responsible for scheduling/directing meetings with team members and advisors.

Frontend Lead

The Frontend Lead was in charge of understanding/organizing the development of the frontend of our application, as well as communicating with the Graphic Design Lead to build the User Interface. The Frontend Lead was also responsible for communicating the frontend development progress to the team members and advisors.

Subject Matter Expert

The Subject Matter Expert was in charge of researching various options for developing the application, and being the most knowledgeable member on the options our team decided to move forward with. The Subject Matter Expert was also responsible for explaining development decisions in-depth to team members and advisors.

User Research & Test Lead

The User Research & Test Lead was in charge of understanding/organizing the development of test cases for our application. Another responsibility was organizing research on natural disasters, and communicating with Red Cross.. The User Research & Test Lead was also responsible for communicating specifically with the Graphic Design Lead and advisors to best develop a UI that was user-focused.

Graphic Design Lead

The Graphic Design Lead was in charge of understanding/organizing the development of the user interface. Another responsibility was communicating with the User Research & Test Lead, Frontend Lead, and advisors to best plan a user interface that was centered around the user's needs. The Graphic Design Lead was responsible for communicating the UI iterations and changes to the team members and advisors.

Backend & Network Lead

The Backend & Network Lead was in charge of understanding/organizing the development of the backend of the application, as well as communicating to the Frontend Lead to connect the application with the backend. The Backend & Network Lead was also in charge of understanding how the application will be communicating through the backend, and be able to communicate this knowledge to the team members and advisors.

5.2 Project Schedule

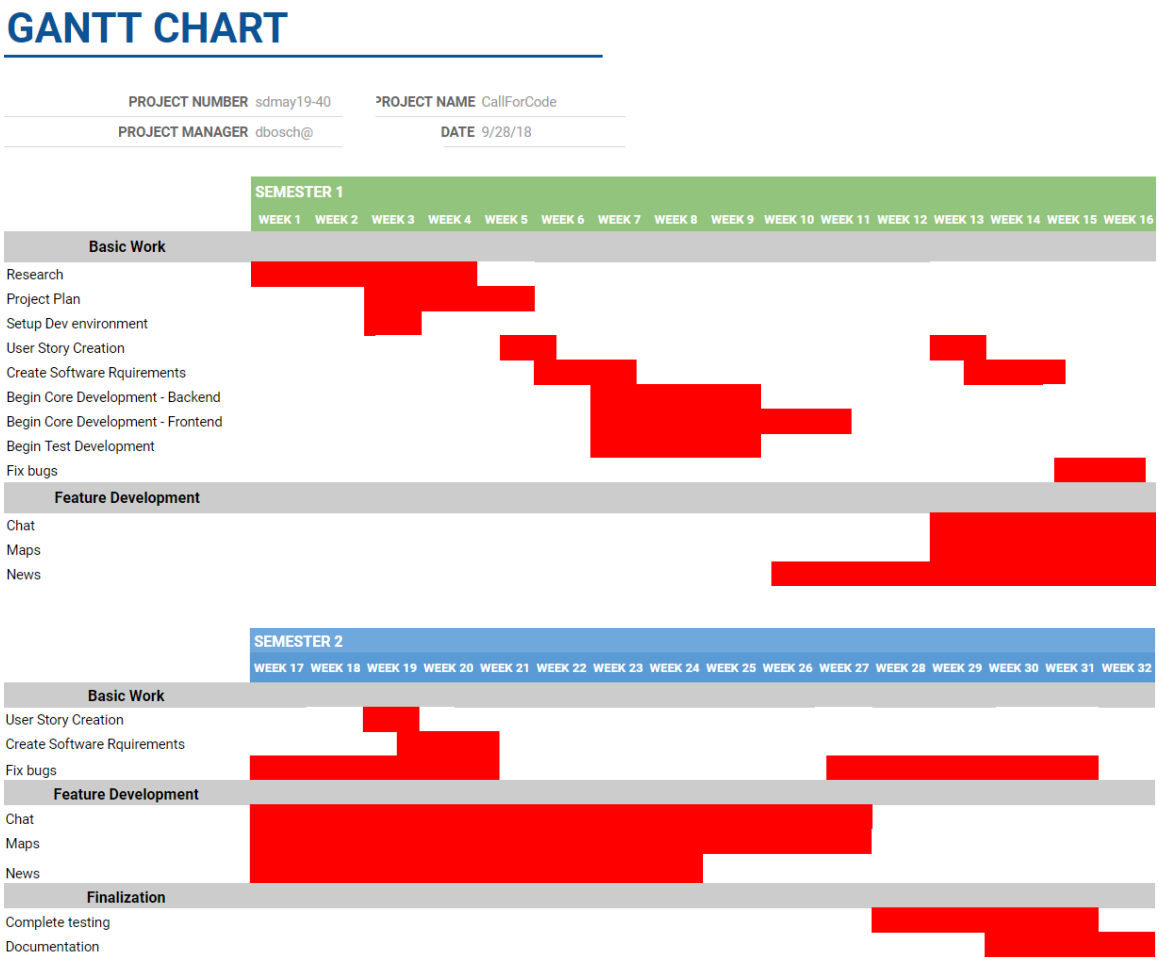

Table 5.2: Gantt Chart

Description of Table 5.2 Gantt Chart

During our first semester, we focused mainly on user research. The process we established with our advisors consisted of initial research by each member that evolved into user needs. From there the user needs helped generate functional requirements. After successfully establishing our first round of functional requirements, we converted them into software requirements, and developed a full plan on how we were going to integrate them into our application. Later in the semester we

planned to start developing the key features to our application in feature teams, as well as continually testing the code we produced.

During the second semester, our team continuously worked on our features as well as develop advanced user features to expand the functionality of the individual features. We focused on finalizing and integrating our features to mesh well in the overall project. Towards the end of the semester we finished testing the functionality and unit tests of our code and created documentation and final reports for our entire project.

5.3 Risks and mitigation

1. **Equipment**

Testing our application on real products and hardware will always be a better way to test an application when compared to testing on an emulator. The ETG provided our group with Android devices for the people who did not own one to begin with. Other than that, we didn't have any other equipment that was required.

2. **Knowledge of Area**

Firstly, we had to table the decision to use an ad hoc network as a form of communication as it was possibly unfeasible. With the time allotted we found that it was not feasible to implement an ad hoc network and that working on other functionalities would prove much more fruitful.

Secondly, we took a little more time than expected to get used to and fully learn Xamarin to the point where we were competent. To mitigate this we assigned two people to work on each part of the functionality. With two people working on different parts of a single function, it was a lot easier to get help and collaborate and work around issues than if you were working solo.

3. **Cost**

Using Google Maps, cost was initially a concern. After looking through their API and deciding what features we are using exactly, we found out that a lot of the calls we are making to the Google Maps API are free of charge and there are only a couple that cost money.

Luckily, Google gives you $300 worth of credit for free and the calls only cost fractions of a cent so we have enough to work with our app while we are developing.

5.1 Lessons learned

The biggest lesson that our entire team got out of this project was the user-driven development, or user-centered design, that we used so heavily. User-driven development is a framework of processes in which usability goals, user characteristics, environment, tasks and workflow of a product are given extensive attention at each stage of the design process. Dr. Nicholas Fila, as previously mentioned, was monumental in directing us through this user-driven development with weekly meetings with our UI lead and team lead. We have all had the pleasure of working

with Dr. Fila and the whole experience has really opened our eyes to just how important using some of these design thinking concepts can be towards a project.

Another lesson that was humbling to the team had to do with some incorrect assumptions about what capabilities were entirely possible within an application. When you first start a project it seems as if the possibilities are endless. Our team really wanted to implement mesh networking, which is a local network topology in which the infrastructure nodes (i.e. bridges, switches and other infrastructure devices such as phones) connect directly, dynamically and non-hierarchically to all the other nodes to work together to efficiently route data. This seemed as if it would be an amazing addition to the list of functional and non-functional requirements we were amassing, but unfortunately we assumed that it was entirely feasible which unfortunately it was not.

Incorrect assumptions really did put a damper on some of the things that the team was trying to work on throughout the entirety of the project development process. Not only was it discouraging when we realized that mesh networking wasn't feasible within the time frame we were given, but we also had some other incorrect assumptions about how easy some frameworks were to learn that took longer than initially projected as well. We had to have extra meetings and plan around these road bumps, which helped in the end but it was interesting to learn and see how our team went about these problems.

Lastly, it helped that we utilized agile software development techniques as it allowed for the things stated above to not harm as much as they should have. Running into a road bump due to an incorrect assumption was not as detrimental as it would normally be as our team was running fluidly using agile development techniques that allowed for changes to be made on the fly without hindering everyone else that was working on the project.

6. Closure Materials

6.1 Conclusion

With natural disasters becoming increasingly more common, ways to provide relief to these victims are becoming more and more necessary. The goal of Call for Code project is to design a tool to improve preparedness for natural disasters and relief when they hit in order to safeguard the health and wellbeing of communities, as well as provide these basic services needed without an internet connection.

The main focus of our design process was user research. Having this main focus and making sure that we based all of our design around user research was important. Having weekly meetings with our advisor, Dr. Nick Fila, helped shape both our functional and nonfunctional requirements. Making sure that we had another set of eyes on our project to help steer us in the right direction was extremely helpful, and we would like to thank Dr. Nick Fila for his help in this regard.

From our research we concluded that the three main features we wanted to implement were: maps and routes, news, and messaging. Looking at user stories from natural disasters we decided these were the most important features that would prove useful to individuals who are impacted by natural disasters.

Overall, based on the above specifications that we focused on, our application has navigation, communication, and news updates. Our map shows pins for relief, supplies, and medical help, our messaging platform allows users to message family members and connect with relief workers, and our news updates show relevant articles that relief employees select based on the necessity towards the current natural disaster.

6.2 Future work

Looking towards the future, there were some features that we would have liked to implement if we had more time. The first is mesh networking. This was something that we would have loved to figure out and get working, but proved too challenging. We ended up implementing a bluetooth connection that will allow users to message without an internet connection, but this implementation is not a full mesh network like we were hoping to implement.

The mesh networking was the biggest thing that we would love to work on more if we had time, but we also think that forms, group chats and life alert could all prove useful future extensions for this project. Forms for filling out request to different relief sources, like a red cross member for example. Group chats could prove useful for location and region based chat rooms. Lastly, a life alert type system would be a great implementation that would work around a lot of the features that we already have. Something with an emergency SOS beacon that could be sent directly to first responders would fit nicely within the overall scope of our project.

Some other functionalities we were hoping to implement, but proved to be out of scope, were adding blocked routes to maps, routes based on traffic, and a connection between maps and messaging to allow users to suggest pins. These additions would allow users to best handle very common issues with natural disasters, undrivable road conditions. The blocked routes functionality would allow the application to best route users to the relief sources. The routes based on traffic functionality would allow the application to handle the common issue of traffic surplus on routes to relief sources. The last functionality, connection between maps and messages, would allow users to help red cross relief workers catch things they may have missed and allow for full coverage of map updates.

6.3 References

"Register for the Challenge." *Call for Code*, callforcode.org/challenge/.

"Rebuilding Rail after the Earthquake." *Great Journeys of New Zealand*, www.greatjourneysofnz.co.nz/blog/kaikoura-earthquake-derails-the-main-north-line/.

Roy, Eleanor Ainge. "New Zealand Earthquake: First Relief Trucks Sent to Kaikoura as Road Opens." *The Guardian*, Guardian News and Media, 17 Nov. 2016, www.theguardian.com/world/2016/nov/17/new-zealand-earthquake-first-relief-trucks-sent-to-kaikoura-as-road-opens.

"The Serval Project." *Serval*, www.servalproject.org/.

Pokharel, Govind Raj. "Nepal Earthquake 2015 Post Disaster Needs Assessment." *Www.nepalhousingreconstruction.org*, Government of Nepal, 2015, www.nepalhousingreconstruction.org/sites/nuh/files/2017-03/PDNA%20Volume%20A%20Final.pdf.

Lokesh, Likitha. "Mobile Application Testing with Microsoft's Xamarin Test Cloud Services (Part 1)." *Medium*, Slalom Engineering, 15 Mar. 2018, medium.com/slalom-engineering/mobile-app-testing-with-microsoft-test-cloud-services-part-1-c0b40b7c19d0.

Collins, Jennifer. Ersing, Robin. Polen, Amy. "Evacuation Behavior Measured at Time of Expected Hurricane Landfall: An Assessment of the Effects of Social Networks" (2015), https://hazards.colorado.edu/quick-response-report/evacuation-behavior-measured-at-time-of-expected-hurricane-landfall-an-assessment-of-the-effects-of-social-networks

Ashar, Faisal. Amaratunga, Dilanthi. Haigh, Richard. "Tsunami Evacuation Routes Using Network Analysis: A case study in Padang." (2018), https://www.sciencedirect.com/science/article/pii/S1877705818300249

Patel, Dhaval. Shah, Kushal. Yagnik, Patel. Avadh, Radadiya. "Literature Review on Tsunami Evacuation System on Coast of Diu.: (2017). https://www.researchgate.net/publication/321422087_Literature_Review_on_Tsunami_Evacuation_System_on_Coast_of_Diu