Project Hermes

Formerly known as CallForCode

Project Plan

Project 40

Client

Diane Rover

Team

David Boschwitz - Team Lead

Caleb Nash - Lead Frontend

Logan Fladung - Subject Matter Expert & Graphics

Justin Kaufer - QA & Test Lead

Austin Worm - Design Lead

Robert Schedler - Backend & Networking Lead

Contact

callforcode@iastate.edu

sdmay19-40.sd.ece.iastate.edu

Revised: 10-26-2018 - V2

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. Introductory Material

1.1 Acknowledgement Caleb

We would like to convey our gratitude towards Dr. Diane Rover, our advisor that will be guiding us through this entire project. Her project management expertise has been monumental to helping us get on track and stay on track. We would also like to thank Dr. Nicholas Fila, who is assisting us on user requirements. His knowledge of the subject has not only helped us tailor our requirements, but also helped us understand his thought process so we can apply this knowledge in the future.

1.2 Problem Statement Caleb

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.

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's involvement in the challenge, but this challenge helped stem the idea for this project.

Our solution to this project is simple. Our first step will be to conduct user research and compile user requirements. This will be done over the course of a few weeks with the help from one of our advisors, Nicholas. After compiling user research and requirements, we will spend some time creating software requirements from the user stories. This will help us streamline what features we want to get done first and when we get into the development process we will be able to complete and test features faster. After developing and testing code, more vigorous code reviews and user tests will help us wrap up and finalize the software.

1.3 Operating Environment Caleb

The operating environment for this product will all be either server-side or on a physical device such as a cell phone. Since we are writing software, the environment doesn't play a part in what we are doing. We will be taking advantage of the EcE servers, which are not only out of our reach, but are very well maintained. Overall, the operating environment is not a factor in this project.

1.4 Intended Users and Intended Uses Caleb

Our intended user is a wide variety. We will have users of different ages, cultures, and mental statuses. We do not know how the disaster will affect each of our users, and we

will focus on helping them out as best as possible. Our main focus, if we had to choose, would most likely be families with children, therefore, we are focusing most on the general natural disaster relief plans that aid families, such as shelter, food, clean water, and medical assistance. Knowing how dangerous hazardous conditions are to children and the elderly will also be a very high priority. We need the elderly to be able to easily handle using our application so that they can get to safety as soon as possible.

Our intended use is pretty self-explanatory, we are creating an application to aid those being affected by natural disasters. This application will be used as a guide for finding the relief stations that will help save our users' lives. Our application will allow users to safely route their efforts to get to the nearest shelters for food, medical assistance, and a place to rest with a roof over their heads.

1.5 Assumptions and Limitations Austin

Assumptions:

- The EcE system will be available 24/7 and will not have any technical problems regarding loss of data or downtime.
- The software will be used in a location that has Google maps data
- The software will be used in a location that has access to NOAA/other government organization disaster evacuation maps.
- The user knows how to use a smartphone and doesn't need assistance in using any of the features.
- The user is proficient in english enough to utilize the software.
- People utilizing specific features (e.g. shelter hosting / energy hosting) are not malicious in their intent during a natural disaster.

Limitations:

- Server limitations. If the EcE system isn't sufficient, then we will have to find a different server solution and possibly receive resources from IBM.
- Network connectivity, especially in times of disaster, will not always be available.
- The system will be a mobile application used on mobile devices that can properly run a mobile application distributed through the play store/iTunes.

1.6 Expected End Product and Other Deliverables Austin

CallForCode Mobile App

> The client will be delivered a mobile application that will use maps with optimized path functions embedded to reduce traffic congestion and speed up travel times during natural disasters. This application will also allow our users to visually see relief sources and have real time updates from news sources to keep them up to date on the current status of their location. This application will fulfill the specified user need statements and functional requirements specified in the team's initial research. This will be delivered with proper documentation for a new

software team to continue development of the project, as well as proper documentation for an administrator to run the app, and proper documentation for a user to use the app. Our application will use the maps functionality to solve the problems of traffic congestion, as well as optimize travel times for users in times. This will be delivered two weeks before the end of Spring semester 2019 (April 26th, 2019)

2 Proposed Approach and Statement of Work

2.1 Objective of the Task Bobby

To develop our natural disaster app, we first need to research and create user stories from natural disasters, and use those stories to create user need statements. Once we created the user needs, the team will create user requirements. From the user requirements the team will create a set of software requirements. These software requirements will create a solution to the user's problems in the form of a cell phone app.

2.2 Functional Requirements

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 a mesh network process

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

FR6. Easy navigation for all types of users

- High stress and technological inexperience shall not hinder anyone from effectively using the application

FR7. Work across all mobile devices platforms

- The application shall be designed to function cross platform

FR8. 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

FR9. Allow users to send emergency pings to relief workers

- A call for immediate assistance in the event that there are relief parties nearby or search teams out

FR10. Provide a constant news update based off locational settings

2.3 Constraints Considerations Bobby

User research will be overseen and critiqued by Dr. Nick Fila. We will complete full sprints of user research and documentation to ensure that our product is the correct solution for our user's problems.

We will follow standard .NET styling and procedures for code style and creation. This will give us a common standard that is already built into Visual Studio, and is best suited for our language choice of C#.

Our architecture will utilize a microservice style on the backend, and MVVM on the frontend. This works well with Xamarin and XAML. We decided to use Xamarin because it can easily work for android and iphone, which means easier access for our users who need it.

2.4 Previous Work And Literature David

Other products have been created to do similar things. The most similar app to our goals is the Serval Project [5] an app that uses mesh networking to talk and share critical data between android phones. Other mesh networking apps exist that hold one or more features similar to those we wish to implement. The most well-known of these is Fire Chat [6], which is a popular mesh-networked chat app. While both of these apps utilize much of the core concept that we are seeking to implement, they lack many of the features that we believe will greatly benefit our users. We hope to go beyond what they've done, by focusing on multiple features in our app, where Fire Chat only allows for chat, we will create more extensive features that our users could benefit from such as Maps and News as well.

2.5 Proposed Design David

We are going to build this application to be very easy to navigate and leave no questions for the user to ponder. We expect our user's to be a very diverse group of individuals, varying in age, gender, ethnicity, etc. We will make this application universally understandable, so that in the time of crisis these people will be guided toward relief as fast as possible. Our main page will be a map that will allow users to view different locations of relief, as well as understand what relief is at that location. Each location will be designated with a universally understood image describing the services at that location. For example, medical will be the red cross, shelter will be a house, and so on. Other than the map feature, we will have tabs to show a list of the specific services and what they are providing. We will also be implementing a chat room so that users can communicate with each other to find rides, because many natural disasters affect transportation and carpooling can help those who've lost their vehicles, as well as the traffic congestion that ensues. Our final product will be aesthetically pleasing, but not overtly beautiful, because our main goal is providing the best product to suit the needs of the user.

This design is strong because of its focus on the user. The team is doing in depth design that deviates from the initial requirements specified by our advisors. After working with our advisors to find better ways to dive into previous natural disasters we found that the ability for the app to communicate without infrastructure would be a huge strength. This

The big weakness in our app is that without mass adoption it will fail. Our app relies on having many devices on our network, and without that it could fail.

2.6 Technology Considerations David

Distributing specialized communication to the people who need it during a disaster would be impractical. Most people today have cellphones which they can use to communicate, however cell service is usually required to communicate. We are attempting to leverage the availability of smartphones in combination with the communication technology in our cell phone application to allow people in "dead zones" to still receive important aid information.

2.7 Safety Considerations Bobby

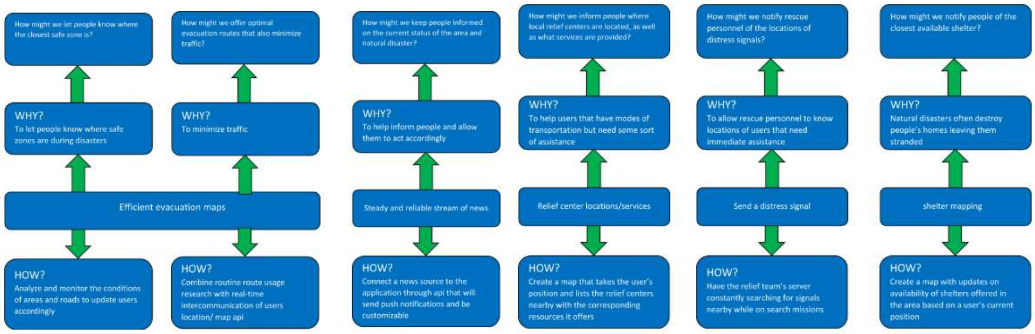
As this is a software based project, safety concerns for the development team are fairly minimal. The development team will implement safety more in the sense of security, to protect user's and their data from other parties with malicious intents.

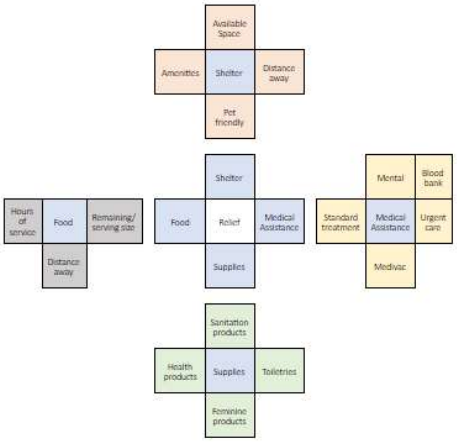
We will not be following any lab standards at this time due to the lack of any lab work.

2.8 Task Approach Austin

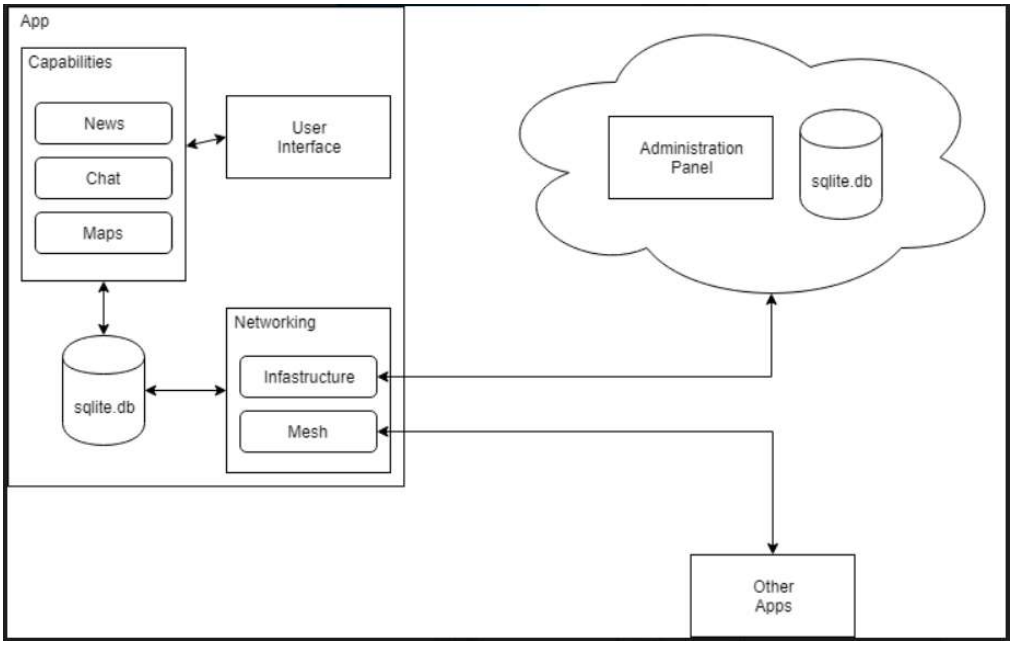
1. Research Natural Disasters

2. Decide User Needs and User Requirements

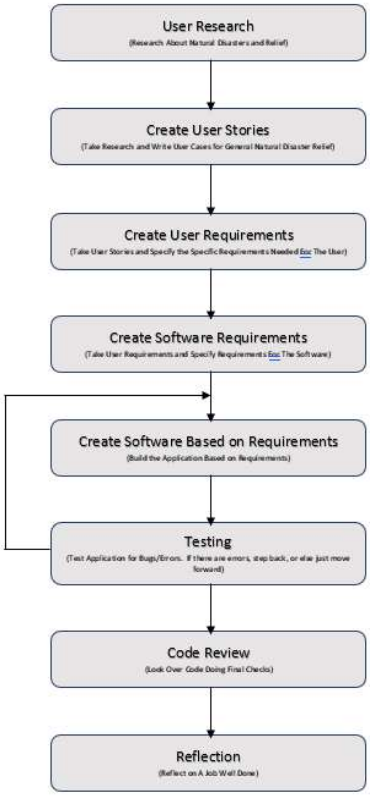




3. Draw Out Application Structure / Components



4. Plan Development Process For Completing Task



Describe any possible methods and/or solutions for approaching the project at hand. You may want to include diagrams such as flowcharts to, block diagrams, or other types to visualize these concepts.

2.9 Possible Risks And Risk Management Austin

Include any concerns or details that may slow or hinder your plan as it is now. These may include anything to do with costs, materials, equipment, knowledge of area, accuracy issues, etc.

1. Equipment
    a. ETG does not have all of the necessary materials necessary to test our application on all platforms
        i. We will manage this by running emulators to test our application for IOS
2. Knowledge of Area

    a. We have completed some research on mesh networks and our subject matter expert has determined that it may not be feasible
        i. We will manage this by tabling the decision to use a mesh network until we gain a better understanding of its possible implementation in our project
    b. We could run into speed bumps with our team not all being very familiar with Xamarin and other concepts
        i. We managed this by creating the position of subject matter expert, and have assigned leads for other positions based on ability to answer questions in their areas
3. Cost
    a. We cannot afford Google Maps API
        i. We will manage this by using another form of maps libraries that is free
4. Accuracy Issues
    a. We will be having a lot more customization to the maps, given that we will not be using Google Maps API due to budget, therefore there will be risks that we will need to address
        i. This is hard to manage, rather we will be vigilant and heavily test our changes to the maps functionality

2.10 Project Proposed Milestones and Evaluation Criteria

Our project is divided into a several key components. Milestones will be reached when features are code and usability requirement complete, and have no major bugs.

User Research

We are diving deep into understanding what our users are going through, and analyzing how natural disasters are formed. We are using our advisor, Dr. Fila, for his expertise on how to approach this using web research, as well as contacting experts in this field who have first hand experience.

Core App

The core app will be a base that gives access to networking and a local database, under a layer of abstraction that all of the features will utilize. The core app will be built using Xamarin, which will allow it to be developed cross-platform for Android and iOS. This will include infrastructure and ad-hoc mode, allowing the app to either communicate with a server, or peer to peer. The database will use sqlite.

Maps Feature

The maps feature will feature a local area map with all the information the user may need for navigation and wherewithal. It will be offline, updateable over both types of network,

and will keep information on suggested evacuation routes, points of interest (ie. First Aid Station, Running Water, Public Outlets, Supply Drops).

News Feature

This feature will include critical updates from verifiable authoritative sources over both types of networks.

Chat Feature

This feature will allow for messages to be sent over the air using both types of networking. It will also allow different channels to be created for group chats, including custom groups, location/area based groups, and auto generated groups based on need. Privacy precautions should be taken and the current intent is to use public and private keys to encrypt messages.

2.11 Project Tracking Procedures

Our team will track progress through issue boards on GitLab. Anytime an issue is presented it is assigned as collateral to a team member and added to either open or in progress. At weekly meeting the issue boards are addressed and updated accordingly to reflect the progress or completion. All meetings, research, and technical work will require issues to log and maintain progress throughout the project life cycle. When an issue is completed and agreed upon by the team it is then moved to the closed log. This process not only keeps the team organized, but allows members to be assigned tasks with notifications of any comments or relevant changes to requirements.

2.12 Expected Results and Validation

The desired outcome is an application that can be ran on any mobile platform and allows connections from multiple devices that expands the range of the network. It will not require a constant connection to the internet as it supports its own network. It will allow users to connect valuable information in areas that have been struck by natural disasters including shelter, hazardous areas, and nearest emergency services.

The team has agreed to follow a process of simultaneously writing tests while developing, as opposed to after finishing. All validation of requirements will come through our extensive testing listed in the section below.

2.13 Test Plan

Unit 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. Directed by our test engineer, the unit tests will be expected to be written by each developer as they are writing code, not at the end.

This will ensure the correctness and functionality to catch any mistakes before they create a domino effect. All parts of functional requirements shall be extensively unit tested to ensure quality. Success is determined by the output being as expected by the developer that wrote the code.

> An example: If someone creates a new user, does that user get stored in the database.

Integration Testing

These will also be made throughout the development process, along with the unit tests. The difference between these and the unit tests is that these test functionality of code as a group. The purpose of integration testing is to test faults on the interaction between specific functions. The team will use the Xamarin.UITest framework to perform these tests on user interfaces. The format will follow the same setup as unit tests in the order of arrange, act, and assert to maintain uniformity. Success is determined by the output being as expected by the developer that wrote the code.

> An example: If a user wants to navigate to the news feed from the main menu, assuming they click the correct buttons do they reach the desired page.

Usability Testing

These tests will not be written in code, but will be research on the user experience of the application. Usability by all types of users is extremely crucial to the application. A variety of procedures established by the team or developer responsible for the feature will be established and tested by various populations. The test engineer will establish a survey to ensure that a variety of demographic and skill sets are being tested. Any feedback received from usability tests will be accounted for and used to help redesign or validate the specific feature. Success will eventually be determined by all populations being able to successfully complete the tasks established by the team.

> An example: Someone should be able to create a user then go in and set their location settings.

Cross Platform performance

With our application we are implementing a cross platform between IOS and Android. Tests are done to assure that the API's and environments of the application are performing at an equal level between platforms.

> Success: There is fluency between all mobile platforms

3 Estimated Resources and Project Timeline

3.1 Project Timeline

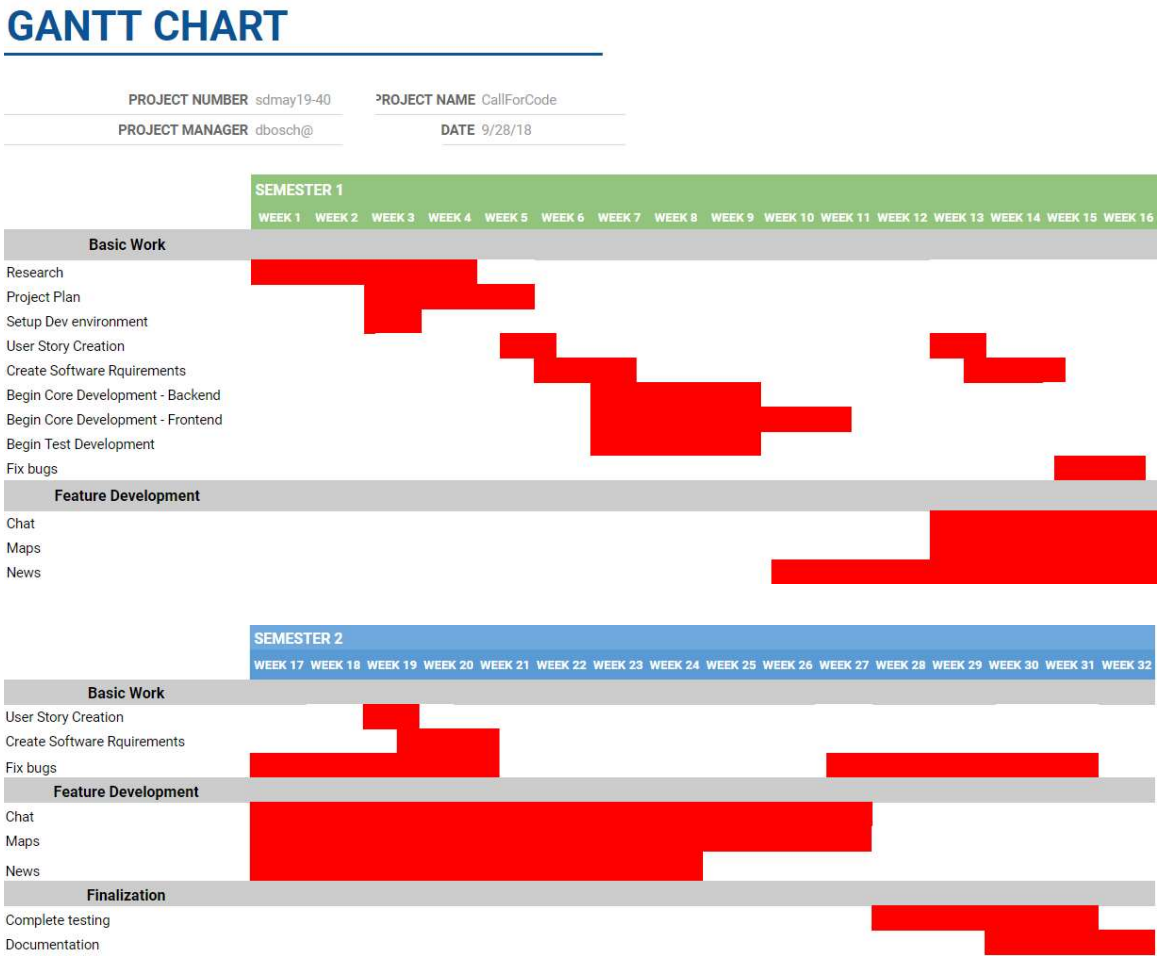


Table 3.1: Gantt Chart

Description of Table 3.1 Gantt Chart

Within the team's first semester the team will mainly focus on user research. The process we have established with our advisors will consist of initial research by each member that will evolve into user needs. From there the user needs will create functional requirements. After successfully establishing our first round of functional requirements we will convert them into software requirements, and develop a full plan on how we are going to integrate them into our application. Later in the semester we plan to start developing the key features to our application in feature teams, as well as continually test the code we produce.

During the second semester our team plans to continually work on our features as well as develop advanced user features to expand the functionality of the individual features. This semester we will really focus on finalizing and integrating our features to mesh well in the overall project. Towards the end of the semester we will finish testing the functionality and unit tests of our code and create documentation and final reports for our entire project.

3.2 Feasibility Assessment Logan

The feasibility of our project is relatively high based on our prediction of using various API's that are available to aid us in our development. The main challenges that we foresee would be the functionality of the application without internet or cellular availability, as well as the application not rerouting with damaged roads. Otherwise, our team feels we have the resources and skill set to accomplish our goal and provide an application that can help save lives in the terrible instances of natural disasters.

3.3 Personnel Effort Requirements Logan

All task required to create the application are broken down below. The front end and the back end will require a lot of time. The peer to peer network will also use lot of time, and require a lot a research.

Tasks:

Table 3.3 Personnel Effort Breakdown

Task	Description	Estimated Time
Create Back End	Create the interface between the DB and front end	150 hr.
Set Up Database	Create required tables and stored procedures	25 hr.
Create Front End	Create the interface between the user and back end	200 hr.
Create Peer to Peer Network	Allow users to communicate without network infrastructure	200 hr.

3.4 Other Resource Requirements Logan

Our other resource requirements include credible articles that will be used to initially develop user needs that will eventually become our functional requirements.

3.5 Financial Requirements Logan

We have no relevant financial requirements.

Table 3.5 Financial Requirements Breakdown

Software	Type of License	Total Cost
Visual Studio Community	Free	$0
Gitlab	Free	$0
sqlite.db	Open-Source	$0
Autofac C# Library	Open-Source	$0
Nunit Testing Library	Open-Source	$0
Xamarin Cross-Platform Development Libr	Open-Source	$0
OpenStreetMap Library & API	Open-Source	$0
	Total	$0

Above is a break down of the list of software we are using to create out application. All of the software is either open source or free.

4 Closure Materials

4.1 Conclusion Bobby

Our goal for this project is to create an application that utilizes mesh-networking technology to provide a disconnected network that supplies those affected by natural disasters with valuable information. We have created a timeline that lays out our development process.

First we need to research the issues victims and aid providers experience when entering an area afflicted with disasters. Then we will research the best techniques to implement our mesh network and give easy access to our users. We will create our requirements for our product and test cases after this. Once we've decided these, we'll make the application and use extensive testing to ensure a smooth and polished experience.

4.2 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.

4.3 Appendices

Figure 4.3.1 Project Process Flowchart