Call For Code

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: 2018-09-28/v1

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

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 Nic also helping us understand his thought process so we can apply this knowledge in the future.

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.

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

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

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

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

CallForCode Mobile App

> The client will be delivered a mobile app with that fulfills 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.

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

The object of this senior design project is to research and create user stories from natural disasters, and use those stories to create user need statements. Then, using the user need statements, the team will create user requirements. From the user requirements the team will create a set of software requirements and a solution to the user's problems in the form of a software app.

2.2 Functional Requirements

- The application shall allow users to see locations of relief, such as medical assistance, food, shelter, etc.
    - This is our main focus of the application, and is a huge assistance to those stranded in very hazardous places.
- The application shall mark roads that are unsafe and not allow routes to run through them
    - This is another safety measure we are putting in place, because the last thing our users need is to be put in a position where they have to reroute themselves because the application didn't tell them that their route was unsafe.
- The application shall track users and route other user's paths based on traffic along popular paths
    - This will help the user not be caught in a traffic jam if they need serious medical help
- The application shall allow users to communicate with each other via chat room in case of having no transportation
    - This will allow users who have lost their vehicles to still reach a safe destination
- The application shall work even when internet and cellular connections are unavailable
    - This will make sure that users are always able to get the assistance that they need
- The application shall be easy to navigate for all users without any stress
    - The users are already going to be stressed and our application should serve to relieve stress rather than add more
- This application shall work on all mobile devices
    - For this application to be successful, everyone should be able to use it
- This application shall have user status' for different kinds of users, such as relief worker, ordinary user, and admin
    - This will allow relief workers to display aid and not allow panic in a time of need by a user posting a false relief source

- The application shall allow users to send emergency pings to the relief workers in case of dire situations
    - If someone needs a medivac, they should be able to get one as soon as possible

2.3 Constraints Considerations

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.

2.4 Previous Work And Literature

Other products have been created to do similar things. The most similar app to our goals is the Serval Project (http://www.servalproject.org/) 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 (https://www.opengarden.com/firechat/), 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.

2.5 Proposed Design

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.

2.6 Technology Considerations

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

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.

2.8 Task Approach

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

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.

2.10 Project Proposed Milestones and Evaluation Criteria

Our project is divided into a couple key features. We will reach milestones once a large feature project is closed and no longer has pending cards in the backlog.

2.11 Project Tracking Procedures

Our team will track progress though a service like Trello or Jira. Keeping a close eye on our backlog will assure we are making forward progress. The team will conduct a short stand-up to hold each other accountable at the start of each weekly meeting.

2.12 Expected Results and Validation

Our final product will be a software that can be ran through a host node and allows connections from multiple devices that expands the range of the network. It won’t require a constant connection to the internet as it supports its own network. It will provide those who connect valuable information about areas that have been hit by natural disasters including shelter, dangerous areas, and nearest emergency services.

We will construct a test platform that simulates an area affected by a natural disaster. Here we would test our hardware to ensure new devices can connect and use all the services.

2.13 Test Plan

Our test plan includes extensively testing all committed code to our application. Our test lead will facilitate and enforce individual testing. For all code we will create unit tests to assure code is functioning as the team expects. As the user interface evolves we will create scripts for testing feature functionality.3 Project Timeline, Estimated Resources, and Challenges

3 Estimated Resources and Project Timeline

3.1 Project Timeline

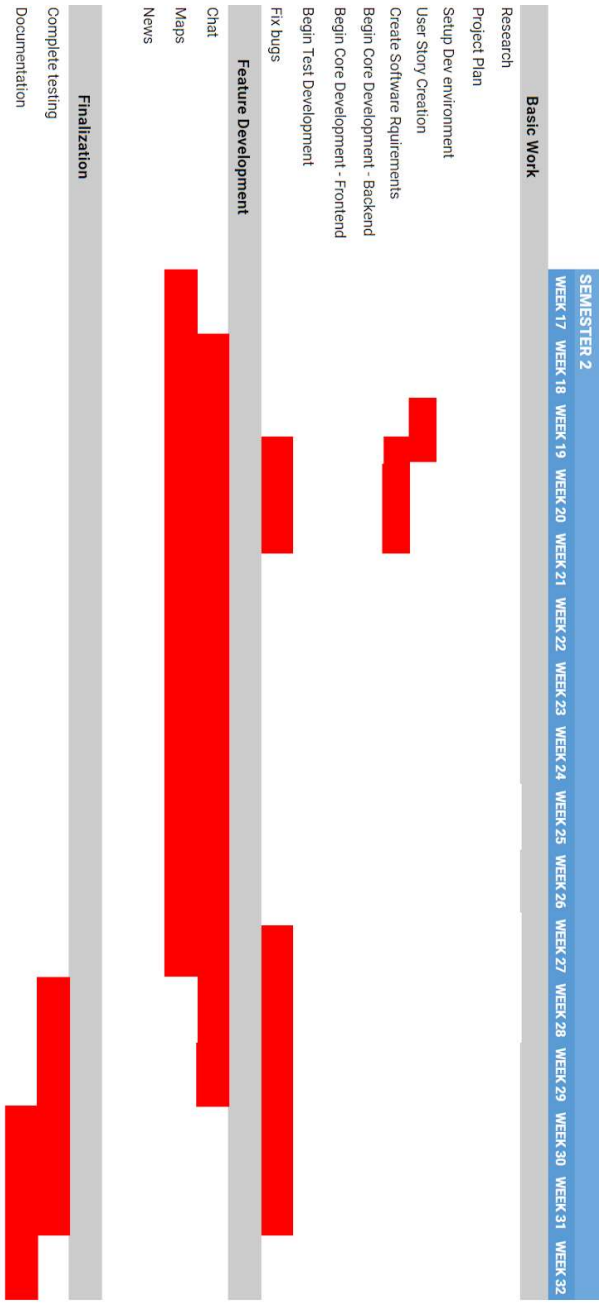

Table 3.1: Gantt Chart

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

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

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:

Task	Description	Estimated Time
Create Back end	Create the interface between the Db and Front end	150 hr.
Set Up Database	Create required tables	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.

Table 3.3 Personnel Effort Breakdown

3.4 Other Resource Requirements

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

We have no relevant financial requirements.

Software	Type of License	Total Cost
Visual Studio Community	Free	$ -
GitLab	Free	$ -
sqllite.db	Open-source	$ -
Autofac C# library	Open-source	$ -
Nunit Testing Library	Open-source	$ -
Xamarin Cross-Platform Development Library	Open-source	$ -
OpenStreetMap Library & API	Open-source	$ -
	TOTAL	$ -

Table 3.5 Financial Requirements Breakdown

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

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

1. "Register for the Challenge." *Call for Code*, callforcode.org/challenge/
2. https://www.greatjourneysofnz.co.nz/blog/kaikoura-earthquake-derails-the-main-north-line/
3. https://www.theguardian.com/world/2016/nov/17/new-zealand-earthquake-first-relief-trucks-sent-to-kaikoura-as-road-opens
4. https://www.nepalhousingreconstruction.org/sites/nuh/files/2017-03/PDNA%20Volume%20A%20Final.pdf
5. http://www.servalproject.org/

4.3 Appendices

Figure 3.4.1 Project Process Flowchart