**Software Requirements and Design Document**

**For**

**Group 20**

Version 1.0

**Authors**:

Betty Tannuzzo

Genevieve Larkins

Madison Vandersluis

Melanie Maguire

Nicole Garcia

# **1.** **Overview**

Natural Disaster Relief is a web application that provides services such as providing advice for people preparing for a natural disaster, providing insight with news updates, and allowing users to ask/provide help with tasks or finding a missing person during or after a natural disaster. In order to provide advice, we will have a preparation tips page that will provide useful tips to prepare for different disasters and links to other sites with useful tips as well. For news we will provide NOAA updates on our page. For the missing/found persons aspect there will be a place for people to submit anyone that is either missing or any unidentified person that they have found and there will be a place for people to look through the submitted entries. People requiring or providing help will also have a forum where users can interact with others to complete a task someone may need help with before or after a natural disaster.

# **2.** **Functional Requirements**

**High**

1. Homepage with tabs to each of the corresponding features
2. User sign up and login pages for users to have accounts with the page
3. Database holding the different users and info
4. Database holding missing people and info
5. Database holding post information
6. Missing person page that allows a user to report a missing person or submit an unidentified person for other users to help identify. The page will also display all entries for users to look through
7. Help forum page that allows users to request help or filter the posts in order to provide help

**Medium**

1. Preparation tips page that lists useful tips and information for natural disasters and links to other helpful sources
2. Map of shelters in a nearby area on preparation tips page
3. Map of users that need help

**Low**

1. Search bar to search a specific person/keyword for help
2. Admin should be able to remove posts that are spam/not legitimate

*List the* ***functional requirements*** *in sentences identified by numbers and for each requirement state if it is of high, medium, or low priority. Each functional requirement is something that the system shall do. Include all the details required such that there can be no misinterpretations of the requirements when read. Be very specific about what the system needs to do (not how, just what). You may provide a brief design rationale for any requirement which you feel requires explanation for how and/or why the requirement was derived.*

# **3.** **Non-functional Requirements**

* Security
  + Users PII stored in our databases should not be leaked and should not be seen by other users
  + Users should only be able to log into their accounts and no others and should only be able to modify their own posts
* Performance
  + Access to our site should be quick (everything should load in under 3-5 seconds) in case a user has limited time to use our resources; issues with google maps API loading promptly occured due to the high volume of markers. However, that is going to depend on the user's internet connectivity and operating system.
  + A user logging into an account or creating an account should have quick results and should easily be able to see their own posts in forums.
* Reliability
  + News updates should have the most up to date information to help people prepare
  + Accuracy of our help forum should be accurate so people can get the most up to date information to provide help with a task.
  + Accuracy of our missing people forum should be accurate so people can get the most up to date information to provide help with locating a missing person.
* User Friendly
  + Users should be able to easily navigate our site to whatever feature they desire without any confusion

*List the* ***non-functional requirements*** *of the system (any requirement referring to a property of the system, such as security, safety, software quality, performance, reliability, etc.) You may provide a brief rationale for any requirement which you feel requires explanation as to how and/or why the requirement was derived.*

# **4.** **Use Case Diagram**

# 

# Name: Report Found/Missing

# Participating actors: User

# Entry condition: User navigates to MIA page and clicks the add entry button, Passenger has a person they want to report found/missing

# Exit condition: Missing/Found person has been added to the database and shows up on MIA page

# Flow of events:

# 1. User goes to MIA page

# 2. User clicks add entry

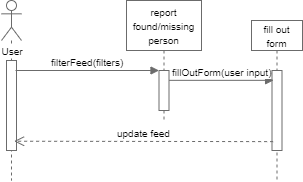
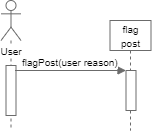
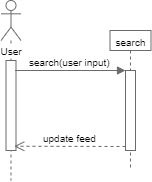
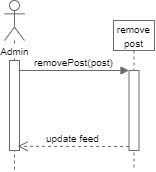
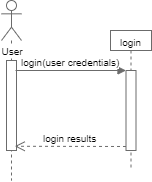
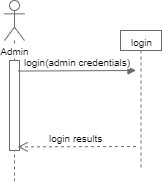
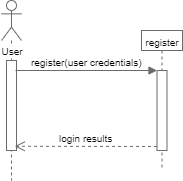
# 3. User fills out/submits form

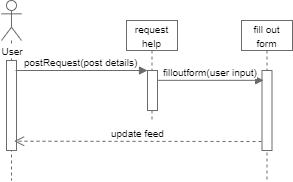
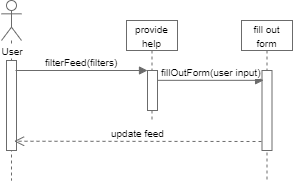
# 4. Entry is added to database/MIA page

# Special requirements: None

# 

# **5.** **Class Diagram and/or Sequence Diagrams**





# 

# 

# **6.** **Operating Environment**

This will be a web application that works both on desktop and mobile. For the best user experience the web application should be used on a desktop. A mobile device screen may be too small and have the webpage act weird. It is compatible with all updated operating systems. The browsers the web app are compatible with are the most recent versions of Chrome, Firefox, Edge (2 most recent), IE (9, 10, 11 (minus “compatibility view” mode) ), IE mobile, Safari, iOS, and Android (Nougat - KitKat). As long as you have access to any of these browsers the user should be able to use the site.

*Describe the environment in which the software will operate, including the hardware platform, operating system and versions, and any other software components or applications with which it must peacefully coexist.*

# **7.** **Assumptions and Dependencies**

* Assumptions
  + If the Google Maps My Map or NOAA Weather Services API changes this may cause issues as it may not display the resources we wish to provide (i.e. shelters). The Google My Maps is updated by a Google user and embedded into the website, therefore, if the user deletes is there could be issues on the site.
  + If the API’s aren’t compatible with our software or don’t achieve the desired effect we want.
  + The size of the creation of the free Firebase database may not be able to hold the amount of users created, which could definitely cause issues.
  + Certain bootstrap layouts don’t work
  + Differences in Angular and Node.js setup on different operating systems
  + Certain HTML templates and code didn’t work with our versions
  + Ways to check information in the database of a specific child that were found were either old or were used for AngularJS/some other language
  + Certain things work for different operating systems, so some solutions have to be scrapped
* Dependencies
  + Google Maps API
  + NOAA Weather Services API
  + Bootstrap templates
  + Firebase
  + Node.js (npm)

*List any assumed factors (as opposed to known facts) that could affect the requirements stated in this document. These could include third-party or commercial components that you plan to use, issues around the development or operating environment, or constraints. The project could be affected if these assumptions are incorrect, are not shared, or change. Also identify any dependencies the project has on external factors, such as software components that you intend to reuse from another project.*