Ryan Bogutz (VM: csc415-server22.hpc.tcnj.edu)

24 February 2020

CSC 415-02 Assignment 2

Proposal

I. Background

I would like to create a web-based application to address the issue of homelessness, more specifically, the issue of access to homeless shelters in cold weather. The city of Philadelphia declares "Code Blue" during times of extreme cold and uses teams from Homeless Outreach in the Office of Homeless Services (OHS) to transport homeless people to shelters. Presently, members of the public can call OHS to identify a homeless person who may need protection from the cold. I would like to create a web-based application to streamline the process of identifying homeless people in code blue situations. With the proposed application, I hope OHS receives more tips and can allocate resources more efficiently in times of need. This project will also be called "Code Blue" to follow the language used by the city.

Since my proposed project is not among those listed among Option 1 in the assignment, I chose to work under Option 2. This is because the issue of homelessness is not addressed in the preexisting projects. Additionally, I am more interested in creating a protype than contributing to a preexisting project. I do not have a lot of experience with web programming and I would like to use this project as an opportunity to build my skills from the ground up.

I would like to create a web application, as opposed to a mobile application, for accessibility purposes. If a member of the public wants to quickly send a tip to OHS, accessing a website is easier than downloading a mobile application. Consequently, the website needs to be responsive and support both mobile and desktop users. Following the requirements of the project, I plan to use Ruby, Ruby-on-Rails, and PostgreSQL. Additionally, I will use my assigned VM to host the application.

II. Project Description

Code Blue is a web application that allows members of the public to provide detailed descriptions to appropriate authorities in the city of Philadelphia to streamline the process of identifying the locations of homeless persons in life threatening cold-weather situations.

III. Innovation and Social Relevancy

A code blue warning is created in response to inclement weather predictions with wind chill below 20°F or 32°F with precipitation [1]. Outreach teams must patrol the city in order to find homeless people and transport them to shelters to prevent injury or death. This task is manpower intensive. While the current phoneline may help outreach teams in their search, I would like to streamline the process of reporting, viewing, and finding homeless people during code blue warnings.

Phone lines are limited, especially in a municipal office. With a web application, the number of reports made at a time are bound to the capabilities of the web server. This allows more reports to be made in a given period of time with the application as opposed to the phone line. With more reports, outreach teams can spend less time searching and more time transporting homeless people to shelters. Additionally, there does not yet exist a system for OHS to manage cases of found homeless people during code blue warnings from report to pick up, and a web application would allow management of this data in a central repository instead of a variety of cross-platform areas. For all these reasons, I believe my proposed project is both innovative and interesting.

Homelessness is a socially relevant issue in our country. According to the U.S. Department of Housing and Urban Development., there are over 500,000 homeless people in the United States [3]. Additionally, an estimated 700 people who are homeless or at risk of homelessness die from hypothermia each year [2]. These deaths are preventable. We can decrease this number by helping the homeless find shelter in cold weather situations. While it is difficult to completely address homelessness, due to the largely multifaceted nature of the issue, it is possible to make a tangible impact by expanding the abilities of preexisting policy.

IV. Algorithms and Data Structures

The algorithms and data structures used for the front end and back end of the application will differ. Starting with the front end, I plan to create a report object for each new report made by a user, with values for the location, number of people found, user-entered description, and time of report. I may add identifiable data of the user who creates the report as an identifier to discourage false or incorrect reports. My rationale for this data structure is front-end requests are unrelated will be entered in separate instances of the app. They do not need to be stored together until all input is validated.

More complicated algorithms will need to be used to validate user input of location and the temperature in the location. I will need to use a geolocation API to get the user's location and verify that it is within the city limits. I need to ask the user for permission to use their location and have a secondary plan if they choose not to enable location services or if their device cannot provide one. I could try to geolocate based on their IP address, or ask them to enter an approximate location. I would prefer the former method since it is harder to spoof a location with location services than simply entering an address. I am considering mapbox as a location API since it is open source and has a generous free tier. Google Maps is another option since it is the de facto standard, but its free tier is less generous. Regarding how the location will be stored, I need to consider my options of cross streets, coordinates, or some other option.

Additionally, I want to verify that a user is making a report during a code blue warning. I will use a weather API to see if the weather conditions in Philadelphia warrant a warning and block the creation of reports otherwise. There are a few options for APIs, with one good option being OpenWeatherMap. It allows 60 API calls per minute with unlimited access to current weather conditions on its free tier.

The back end of the program will need to use efficient algorithms and data structures to store and access reports. I would like to use a database to store reports, with a table per day. Along with the values for reports that are gathered on the front end, I would like to have an additional Boolean

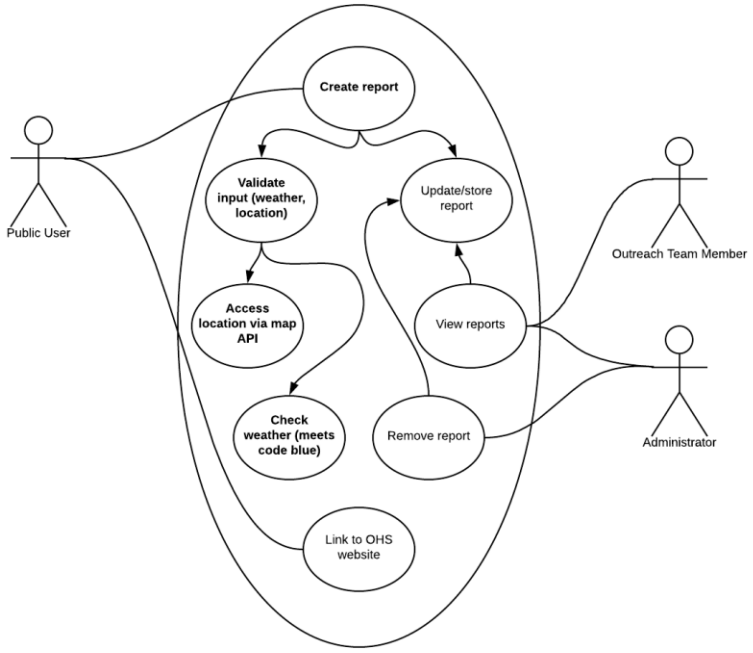
value to indicate whether a report has been addressed. The front end of the program as well as the backend will access a program that manages access to the database, both storage and updating. I will use the library for PostgreSQL to add reports or update their values. This program will also manage the creation and deletion of tables. Depending on the memory capabilities, I would like to delete the older tables after a given period of time, to be determined.

In order to view reports, I would like to make an administrative view that lists pending reports. They will be stored in a queue structure ordered by reporting time, pulled from the database. A queue is appropriate given the ordinal nature of the reports. Additionally, I would like to give administrators the option to update or remove reports, which will be done through an interface to the database program.

V. Software Engineering Concepts

I expect to apply many new concepts in this project. I will learn how to apply my knowledge of the Model, View, Controller paradigm through the development of a web application with Ruby on Rails. Through the planning process and other assignments, I will reinforce the concepts of requirements analysis and modeling. I will learn about proper design of a user interface, how to properly use open source software, how to create secure software, as well as maintenance and refactoring. I am excited to create a large application and apply what I have learned about programming, algorithms, and data structures in a more meaningful way than my previous, smaller programming assignments from other courses.

VI. Use Case Diagram



Note: use cases in bold will be implemented this semester. Other use cases will be implemented if there is time by the end of the semester or in the future.

VII. Proposed Timeline and Resources

Phase 1: Proposal (10 Feb – 26 Feb)

- Research social issues
- Brainstorm project ideas
- Gain feedback from faculty and peers
- Create proposal
- Create project stubs and GitHub

Phase 2: Requirements Gathering and Practice (27 Feb – 26 March)

- Contact Philadelphia OHS for suggestions and restrictions
- Create more detailed plans and models (via Assignment 3)
- Gain familiarity with Rails, PostgreSQL, and location/weather APIs
- Create prototype V1

Phase 3: Prototyping (27 March- 6 April)

- Refactor project based on feedback
- Progress report with stakeholders
- Create Prototype V2

Phase 4: Project Transition (7 April – 7 May)

- Continue refactoring code and adding features as necessary
- Plan future of project based on current progress and past goals
- Submit final prototype by end of semester

Resources:

- Ruby on Rails guide
- Map and weather API guides (pending API approval)
- PostGreSQL guide
- Feedback from peers, faculty, Philadelphia OHS

GitHub Repository

Name: codeblue

URL: https://github.com/rbogutz/codeblue

Bibliography

[1] City of Philadelphia Office of Homeless Services. "Weather – Code Blue." http://philadelphiaofficeofhomelessservices.org/services/weather-code-blue/

[2] Sturgis, Rebecca, et al. "Winter homeless services: bringing our neighbors in from the cold." A Report From National Coalition for the Homeless (2010).

[3] U.S. Department of Housing and Urban Development. "The 2019 Annual Homeless Assessment Report (AHAR) to Congress." (2020).