

Jeffrey Sabo

10/11/17

Assignment 3: OSS Project Proposal

Github: https://github.com/saboj2/Assignment3_Sabo

In the past few years, issues regarding allowing the use of gender-neutral bathrooms by transgender and genderqueer individuals have become a topic of debate at the national level. Over time, different institutions and businesses have begun declaring their bathrooms as gender-neutral or unisex bathrooms, meaning they allow for people who don't identify with their assigned gender at birth, or any constructs of gender for that matter, to comfortably use their bathrooms without needing to fit the binary norm. Since these bathrooms are not yet a guarantee for most businesses or institutions, locating them can be somewhat difficult. However, access to bathrooms are a necessity for all people regardless of gender alignment.

My plan is to create an android app to direct users to the nearest gender-neutral bathroom located on TCNJ's campus. My application will also provide resources about TCNJ's gay-straight alliance, PRISM. Since I am making my own app, I chose option 1. Although there is already an open-source project that focuses on locating gender-neutral bathrooms, my application will be specific to TCNJ's campus, and perform some basic mapping of sidewalks on TCNJ's campus. This accuracy in such a small area is important because bathrooms are something that need to be accessed in a very short amount of time. Also, college students may not always have access to their own private bathrooms if they are living in dormitories, so knowing how to get to a nearby gender neutral bathroom will make these students feel more comfortable and welcome on TCNJ's campus. There is also the possibility of expanding this to other college campuses, as well as the ability to leave comments about each bathroom to let other people know whether it is safe and clean to use.

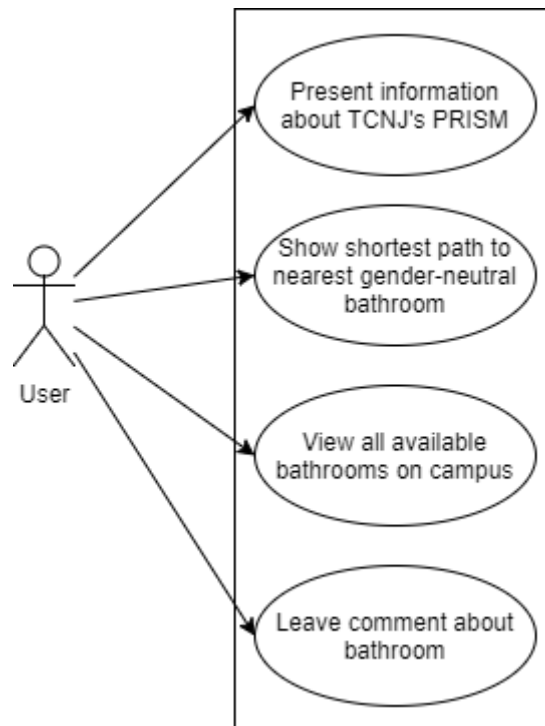
To implement this, I'm planning on using a weighted graph data structure to create distances between different sidewalk junctions. Certain destination vertices will be specified as the usable bathrooms, and the app will determine the shortest path to the nearest bathroom vertex and outline the path by comparing the weighted connections between the vertices. The app will also differentiate between residential buildings and public buildings. Some gender-neutral bathrooms are located within residential buildings that require special access to enter, so it is important to make sure the user can actually find a bathroom they can use.

The algorithm that will be used to find the shortest paths will be Dijkstra's algorithm. This algorithm goes through each node in a graph structure and compares the distances between neighboring nodes until it finally results in the shortest path to a node. Both the algorithm and the data structure should be able to incorporate newly added nodes in the event that new bathrooms or buildings are added to TCNJ's campus.

Java is the main language used for developing android apps. Java, graph data structures, and Dijkstra's algorithm were learned in CSC 250. However, coding android apps is something I have little experience with, so learning the more advanced and specific coding techniques

required for making a working android app will need to be researched. Also, front-end programming techniques will also need to be researched to make features like a user interface and different graphics. The Android Studio IDE will be used to develop the app. The IDE has a lot of documentation to go with it, and the java language is extremely well documented as well. Also resources like Lynda.com can be very helpful for learning the important skills required for undertaking this project.

A UML use-case diagram is provided for better understanding of the overall functionality of this app.



I think it's also important to mention that since there is an open-source project whose aim is to help people locate gender-neutral bathrooms, I should consider contributing what I make towards that project. I can either use stuff that they have already done to help build my project, or specifically design a module to work with their already existing project. It's important to understand that working with an already existing project can be very challenging because reading someone else's code can be very confusing, but sometimes it is made up for by reducing the total amount of work you need to do. This already existing project will be explored further before deciding whether it is worth it to integrate my project with this one.

Licensing:

MIT License:

- Very unrestrictive, allow others to keep their derivative source code closed
- Waives liability from me from any bugs or problems it causes.

- Very short and to the point, may be enticing for others to help contribute or use the project

GNU General Purpose License:

- Forces contributors to adopt the same license for anything they make with your source code
- Prevents people from making money off of it without crediting you
- Can be a lot more confusing to understand in terms of legal analysis, can scare off possible contributors/adopters

Apache:

- Very similar to the MIT license in that there are very little restrictions on what people can do with your code, but uses a lot more words and legal articulation
- Better for settling legal disputes because of how specific it is
- The extra verbiage can scare off possible contributors or users similar to the general purpose license.

The MIT License is the license I'm choosing to use for my project. The object of my app is to provide a comforting service to people in need, and I don't want to have any restrictions on what people can do with it. The most important thing to me is that I am not liable for any harm someone else might enable using my app. However, for the people who want to positively contribute to either my project or to the good of society using what I made, I want to make sure it is easy for them to do so.