Christopher Pham

01651641

Mobile App Programming I

Term Paper – Simple Navigation

Simple Navigation

I sought out to create a navigation app that would help hikers find out where they are quickly. When creating this app, I was looking to make hiking on trails easier for those who already use current navigation apps on their phone and those who may be looking to find an easy way to know where they are going.

The conception of the app began when a few of my friends and I were hiking up Mt. Cannon, a hike that we considered long and grueling. Hiking in New Hampshire appeals to us because unlike the city of Lowell, it is a calming and peaceful atmosphere where we can experience the beauty of nature. Like many amateur hikers, we relied mostly on our phones for navigation to find what turns to take along the trail and orient ourselves. Of the four of us, three of us used Google Maps while I chose to use the map feature on Snapchat. The reason I avoided using Google Maps like my friends was that in my past experiences with the app, I found it too cluttered with unnecessary information that made it difficult to use. My friends agreed with me and added that it was difficult to use such a cluttered app especially while trying to manipulate the map when fatigued and sweaty. This sparked the idea of creating an app that was simpler and intuitive to use for tired hikers.

One of the issues I see in many of today’s navigation apps, such as Guthooks, All Trails, and Google Maps, is the use of unnecessary features that make the app more complex and less intuitive to use. Due to features such as search bars, insignificant details on the map, and complex map views, the user has a harder time figuring where they are. As a result of being cluttered with unnecessary information and features, the user is forced to spend more time navigating the app rather than focusing on the hike at hand. By streamlining my own app to deliver solely what is needed by someone talking a walk or hike, we can improve the hiking experience for those wishing to spend more time enjoying nature and less time on the phone. Ultimately, the goal of the app I would build would be to create something that a person hiking could look at for a few seconds, find their location, find where they are going, and be on their way quickly.

When I first started to create this app, I began with using the pre-implemented Google Maps activity in Android Studio. What I wanted to do was have a map that showed a highlighted path between the user and their destination. As I continued to use this, it occurred to me how difficult it was to develop over the Google Maps main screen. Little did I know that this would be due to a problem with Android Studio I would discover later on. I attempted to create this interface that would lie inside Google Maps. However, this would prove harder than I anticipated. Upon searching for tutorials to do so, I found less than I hoped for and finally came to the conclusion that it was not feasible to create an interface in Google Maps I desired, a decision I would later reverse. I came to this conclusion because my initial interface for the map required drawing a path from where the user was to where the user wanted to go, on a trail that may not exist in Google Maps. Thus, for the time being, I could no longer rely on adding an interface inside of Google Maps. In an effort to solve this problem, I thought I could create an interface that would layer over the Google Maps page rather than inside of it. Rather than manipulating the map and creating significant points inside the map, I would layer what I needed to display on top of the map and find a way to correlate the map layer and interface layer together seamlessly. To explain how this would work, say the user is at a certain location and I wanted to put a marker on said location. I would take in the location data point where the user was, then take the portion of the map that was showing on the screen, and then find where I needed to put the point on the screen so it would align with the user’s location on the map. Although this seems like many steps to mark the user’s location, it was the only option I had at the time given the problems I was having and issues I discovered later on in the process. Upon trying to implement this idea, I found it incredibly difficult to align the map with the phone screen because I would need to take in the data of where Google Maps was displaying and have that data tell me where to put a marker on the other layer of the interface. As I tested this on multiple devices, however, each device showed a slightly different location due to the map and interface not aligning together with the different number of pixels. The result of this method was an inaccurate and unreliable display of a certain location. Eventually, I cut this first attempt short due to this error and non-compiling issues.

As I began the second iteration of the app, I discovered an issue that resulted in my first attempt at the app taking longer than expected. The issue I was facing was that my code would not compile most of the time and I would get inconsistent “Unresolved Reference” errors despite trying everything I could imagine. When I referenced a class in another file, Android Studio would not recognize the reference despite my code being written correctly. Sometimes the code would compile without this issue and other times it would not. This resulted in me retyping the same code over and over again multiple times in multiple cases in hopes that it would work. Through trial and error, I realized that when I wrote the code without using Android Studio’s “auto fill” code, the class reference would not be recognized. Thus, many hours were spent rewriting code until I figured out that I had to select the reference from Android Studio’s auto fill code to get it to work. I cannot stress how much of a problem this was to me, which is why I’ve dedicated an entire paragraph to this issue. This error has certainly delayed my project by weeks. If this is a characteristic of Android Studio and not just and issue on my machine, I would have to blame the negligence of Android Studio’s developers in this regard. If this problem exists in all Android IDE’s I would call it unacceptable and as one can tell, I am very upset. Too much time was wasted retyping grammatically correct code.

After learning how to fix this issue and taking from my previous experience with Google Maps in the last attempt, I reconsidered what a good interface could be. I decided that rather than drawing the path that the user wanted to take on a trail Google Maps had not yet properly “mapped”, I could just label two points; where the user was and where the user wanted to go. This would allow me to put the interface directly on Google Maps and maintain the idea of a super simple interface. I figured that the user does not actually need to see a highlighted path of where they are going, as they can already see the trail outline on the map and where they are. This would enable the user to quickly find where they are going in likely the same amount of time. Additionally, to make up for the lack of a highlighted path, I’ve decided to add a simple compass that would align with the map. This would allow the user to quickly orient themselves on the trail in relation to the map’s position. Additionally, for those who hike as much as I do, a trick to quickly determine the direction you are going in is to align yourself with a compass and where the sun is. This little trick helps experienced hikers figure out basic left or right turns.

The resulting app does cover the basic needs of a simple navigation app. It consists of a Google Maps home screen, a compass, and the ability to create, label, and save location points of where the user wants to go. Although it is very minimal and not as aesthetically pleasing as the apps it is meant to compete against, I believe it has done what it needs to accomplish. With less clutter and more simplicity, I could see how this kind of app could save milliseconds to seconds on the trail when the user wants to know his location in relation to their destination and path. A hiker could easily use this app to navigate wherever they are going quickly.

Sources

Youtube tutorial for Google Maps: (this is a series a videos made by this Youtube creator): <https://www.youtube.com/watch?v=5dBnunaRkq0&t=529s&ab_channel=RahulPandey>

Youtube tutorial for Compass:

<https://www.youtube.com/watch?v=972tRIzQ5iI&ab_channel=R%C3%B3bertKiszeli>

<https://developers.google.com/maps/documentation/android-sdk/marker>

<https://developers.google.com/maps> (For Google Maps API)

Other tutorials:

<https://developer.android.com/kotlin/parcelize>

<https://github.com/albertopeam/kotlin-maps> (learned Google Maps manipulation through this code)

<https://stackoverflow.com/questions/55586613/how-to-draw-shortest-path-on-google-map-in-kotlin-with-live-navigation>

<https://developer.android.com/guide/topics/ui/layout/recyclerview>

<https://github.com/android/views-widgets-samples/tree/main/RecyclerViewKotlin/>

<https://stackoverflow.com/questions/60465435/how-to-add-marker-on-google-map-with-kotlin>