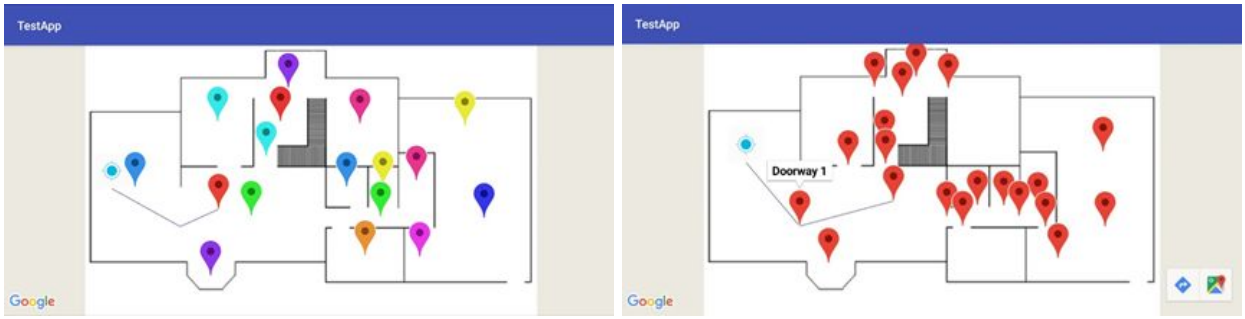
Methods and Materials

For this project, Android Studio was used to create and program the application. Android Studio was used due to it being the official IDE for Android applications and it being free and open-source. Android Studio has a large amount of features that were essential for the production of this application. One feature is the simple transfer of code from computer to an Android device, which was essential for bug fixing and testing the application, as any changes to the code could be transferred to the Android device without hassle. Another feature that Android Studio had was that it would automatically generate files and folders needed for coding an application. This allows for less focus on getting an app running, and more time on coding the app. The final feature that Android Studio had was the easy implementation of external APIs, which is third-party code that is used to do specific features. Android Studio uses Gradle, which is a build-automation system that allows for the import of different APIs without having to manually download the APIs and implement them into the code.

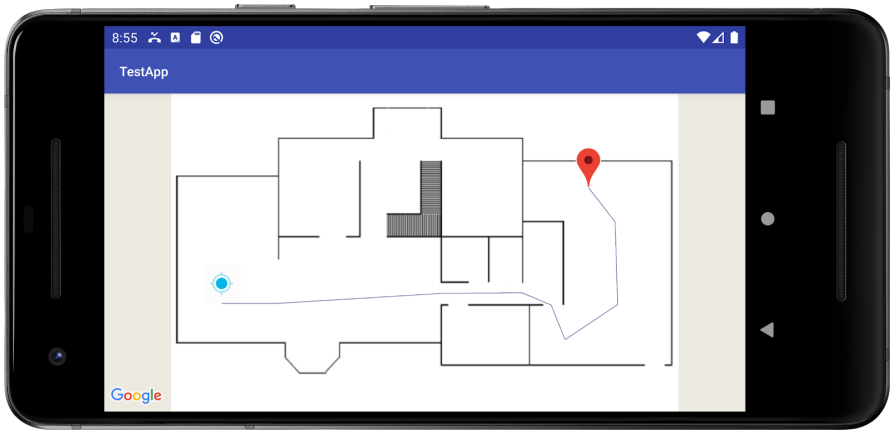
The main API library that was used for this project was the Google Maps API. This adds multiple useful features for creating a navigation app, such as the ability to create a custom map, add markers, create lines that can connect multiple markers, and determine if a marker is in a specific boundary. The prototype that was created used all of these features. The prototype used boundaries to represent rooms; markers to represent doorways, current position, and final position; and lines to show the best path from current to final position.



(Fig. #) Left picture represents rooms, right represents doorways.

In Fig. #, the left picture shows markers at the center of each room. The rooms are created by having a rectangle boundary be created by using the corners of the room. The right picture shows the markers that represents doorways. The code works by determining which rooms contain which doorways and which doorways are connected to each other. The first part is used by the code to determine which room the user is in and which doorways the user can go through, and the second part is used to determine the shortest combination of doorways to get from one point to another. All of those markers are hidden in the background, as shown in Fig. #+1. Using this API comes with some restrictions, however, such as requiring the Google logo to be displayed on

the app (as seen on the bottom left of Fig. # and Fig. #+1).



(Fig. #+1) The app as the user would see.

The final important part of this project were routers or access points with IEEE 802.11mc functionality. IEEE 802 is a family LAN protocols used by routers, access points, and other wireless devices that allows communication between devices. This is one of the most common LAN protocols, used by devices ranging from routers to smartphones. IEEE 802 however, contains specific protocols, with some having unique features. IEEE 802.11mc has a feature called WiFi Round-Trip-Time (WiFi RTT). This feature allows position to be determined in indoor spaces with an accuracy of 1-2 meters. It does this by sending not only signal strength information to a user, but also the time it takes to get to the user and back. This time can be used to calculate the distance the user is from the router because radio waves always go at the speed of light. For this project, I used a Google Nest WiFi router with multiple Google Nest WiFi points. This is because IEEE 802.11mc is a fairly new LAN protocol, and Google is one of the companies leading development in creating routers that have that WiFi RTT capabilities.

To test how many routers were needer, trials were run with 1 router and 2 points, 1 router with 3 points, and 1 router with 4 points. In these trials, the phone would be placed on multiple points, and the position the application thought the phone was compared to the position the phone was in real life. They were compared by distance the application was in real life, which was calculated by scaling the point by a common ratio as the floor displayed on the map was created to be the same shape as the floor plan in real life. These positions were then used again with additional access points, and the process of measuring the distance would repeat.