# A DATA-DRIVEN AND COLLABORATIVE MOBILE APPLICATION TO ASSIST SENSORS USING ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING

Aaron Tse[1] and Yu Sun[2]

[1]Arcadia High School, 180 Campus Dr, Arcadia, CA 91006
[2]California State Polytechnic University, Pomona,
CA, 91768, Irvine, CA 92620

## *ABSTRACT*

*There are numerous arguments as to what the best GPS software is [1]. However, there are no definitive answers as to which is the best. In this paper, we use multiple GPS applications tracking user locations to determine what GPS is best in terms of tracking users through a mobile app [3]. The app utilizes a GPS as well as a Google Firebase Realtime Database to manage, pinpoint, and track users' locations [2]. The application is explicitly applied to track locations of people that need care, such as the elderly. This will allow concerned caretakers to help keep track and take care of people in need.*

## *KEYWORDS*

*Android, GPS, Flutter, Firebase, Google Maps.*

## 1. INTRODUCTION

A GPS (Global Positioning System) has many uses in the modern age [4]. When it was started by the US Department of Defense in 1973, it was originally designed to assist soldiers and military vehicles in accurately determining their locations world-wide. In the modern day, however, it has taken on a whole new use. Whether it is helping people navigate to a new place or finding your missing phone, the GPS has become a central part of our lives by helping us get around. With 2.5 billion registered Android users and 1.8 billion Apple devices worldwide (all of them having GPS on them), it is imperative that this system is a well functioning part of every device. One of the many useful uses for this system is to locate people in need of help. This includes missing people, kids that aren't coming home on time, etc. One group that would greatly benefit from this app would be the elderly. With approximately 125,000 search and rescue missions where volunteer teams are deployed to find missing Alzheimer's patients every year, people with mental disabilities and their caretakers would greatly benefit from this app as well. Despite the potential dangers that this app poses, it is a necessary system to implement. With over 90,000 people missing in the USA at any given time on average, a GPS locator app will be imperative in aiding tracking and the rescue of people. Whether it be concerned parents or medical personnel utilizing the app, it will be used to save lives. There are many GPS apps on the market, with the two biggest ones being Google Maps and Apple Maps. According to BatuhanKilic and Faith Gulgen, Google Maps is accurate because the positional accuracy of the Google Map is mathematically more significant [5]. However, how does this mathematical accuracy of Google Maps measure up to that of Apple Maps?

David C. Wyld et al. (Eds): SIP, CRBL, WiMNeT, BIOM, EDUPT, NLDM, SOFTFM, COMIT - 2022
pp. 115-124, 2022. CS & IT - CSCP 2022 DOI: 10.5121/csit.2022.121712

Some of the existing technologies that have been proposed to enable or expedite location tracking, such as LifeAlert, have been marketed as being useful to people in need, especially seniors [14]. The primary function of LifeAlert is that it is a necklace with a button on it. Once the button is pressed, a user would be automatically connected with a 911 operator, to whom the person in need is able to speak to. Although the necklace itself doesn't have a GPS system, the 911 operator is able to determine the victim's location through tracing of the call. This slows down response times, taking away precious seconds of life. Although sounding promising, these implementations have largely faded into obscurity for multiple reasons. GPS tracking systems that aren't connected to mobile devices are plagued with issues. The first being battery life. Since they cannot be charged while still in use, this leaves people in need vulnerable during the charging duration. Secondly, most of these devices charge subscription fees in order to function to their full effect, such as LifeAlert with their subscription service price being up to $89.95 a month. Compounded by the fact that most services similar to services like LifeAlert bind people in multi-year contracts, these devices are nonoptimal. Another problem that is prevalent with medical alert devices is that they do not function in low-reception areas. This is not an issue for mobile devices, as they can utilize wifi as well to make calls. Although these medical alert devices have gained some traction, for the large part, they have not broken through the market. An ideal solution would be some device that would be near effortless to operate and not be forgotten when being brought out. Since people always have their mobile device on hand, it would be natural to have some location software on a mobile device [6].

In this paper, we aim to solve the aforementioned problems regarding the Google Maps GPS and how accurate it is [7]. My goal is to pinpoint the accuracy of the GPS by going to multiple locations and tracking the accuracy by using the Google provided position and measuring it against the actual position the person is at. There are various different studies that have been conducted regarding this topic. One such study studies the accuracy of the GPS relative to multiple cellular data options, such as 2G, 3G, and 4G. Another study focuses on the potential of Google Maps location history data to characterize individual air pollution studies. My experiment is to solely test the accuracy of the Google Maps GPS relative to your actual location in the real world.

In order to make informed and reasonable predictions on the accuracy of the Google Maps GPS, I have developed an app that tracks users' locations. By making an account and pressing the button to get your location, the app shows your location with a red pin on a map. Additionally, by "friending" other people by inputting their email, you are able to see their locations as well. By comparing where the red pin is relative to my relative location in the real world, I am able to judge the accuracy of the Google Maps GPS. The accuracy will be tested in multiple locations, ranging from places with a high density of people with places with a low density of people to assure that there are no other outside factors that may adversely affect the study. Additionally, I will utilize multiple apps, such as Apple Maps to further verify the legitimacy of the location listed.

The rest of the paper is organized as follows: Section 2 details the multiple challenges that I ran across developing the location tracking app and performing the experiments; Section 3 focuses on the details of my solutions corresponding to the challenges that were encountered in Section 2; Section 4 presents the relevant details obtained as a result of the experiments, with Section 5 presenting related work to this experiment. Finally, Section 6 will give the conclusion on the study, as well as state any future work that may be done.

## 2. CHALLENGES

In order to build the project, a few challenges have been identified as follows.

### 2.1. App Requires Real-Life Unpredictable Variables (Internet Speed)

On top of the accuracy of the map itself, there are a multitude of unpredictable variables in real life that may affect the accuracy of the map even more. The first variable, internet speed, can play a huge factor. If the user isn't connected to the internet either through wi-fi or a cellular data connection, they are unable to update their location in the database, not being able to track their location as well as others'. Another unpredictable variable is the phone itself. As every phone is a little bit different, one wrong setting or outdated operating system can throw off the accuracy of the GPS pinpoint.

### 2.2. Google Services May Not Yield Proper Results For Various Reasons

The app relies on various google services, such as Google Firebase and Google Maps, to function properly. It relies on Google Firebase to read and display the information of various users, such as their user information and their location. It relies on Google Maps to show the locations of "friended" users. If any of these services are rendered inoperable for even a moment, the app will cease to function properly. Another disadvantage to using google services has to do with the Google Play store. Compared to the iOS App Store, the Google Play Store is difficult for new users to find apps. Google Play Store has limited options as to how developers can promote their apps. Additionally, Google is also unable to produce optimal results based on user ratings, popularity, etc.

## 3. SOLUTION

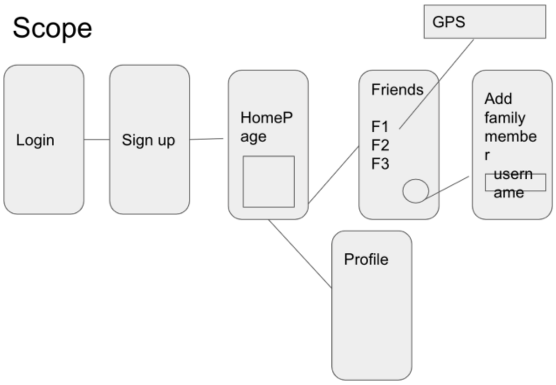


Figure 1. Overview of the solution

Figure 1 displays user navigation paths for the application. This application allows users to track each other's locations through a "friend" system. Once you have "friended" someone else, you are able to see your friend's location on a map, along with their name and their latitude/longitude coordinates. This application utilizes flutter and dart as its coding languages. Additionally, it uses Google Firebase as a database to store and encrypt user information [8]. First, the user will navigate to the "Get Location" page. Once at that page, you can press the "Get Location" button to use Google Map Services to pinpoint your location. Once your location has been pinpointed, you can press the "Push" button to push your latitude and longitude coordinates to the database.

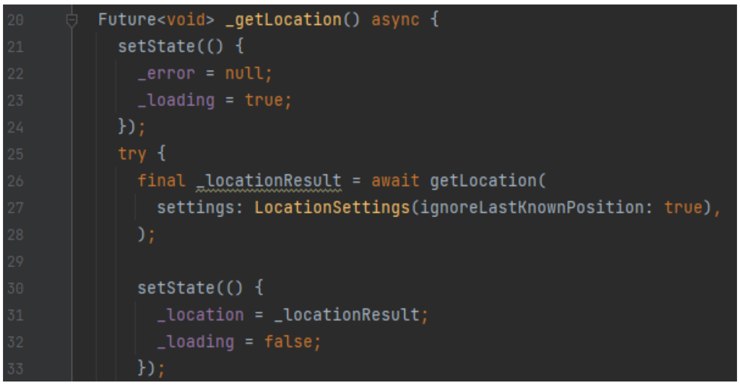
Once your information has been pushed, your coordinates will be shown on screen for you and your friends to see.

The frontend of the app was built using Android Studio, using Google's Open Source UI Software Development Kit, Flutter. Flutter uses the object-oriented programming language, Dart.

The application, targeted towards any person who wants to track the location of others, consists of 3 parts. The first part is the login/sign up page. In order to sign up, you have to provide various pieces of information about yourself, such as your username and email. Following that, you are now able to either change the credentials of your newly created account, view the map where the locations of yourself or your friends are located, update your location, or "friend" people.

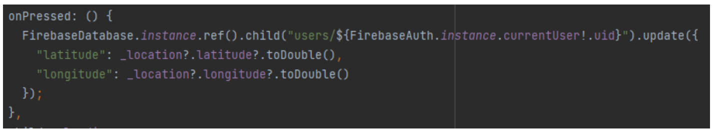
All the functions of this app are made possible with the communication of the front end of the app, as well as the back end, which is a cloud database, made possible by Google Firebase. Google Firebase provides many advantageous features such as user authentication by email, efficient communication between front and back end, and, most importantly, is supported by Flutter.

Get Location:



```
Future<void> _getLocation() async {
  setState(() {
    _error = null;
    _loading = true;
  });
  try {
    final _locationResult = await getLocation(
      settings: LocationSettings(ignoreLastKnownPosition: true),
    );

    setState(() {
      _location = _locationResult;
      _loading = false;
    });
```

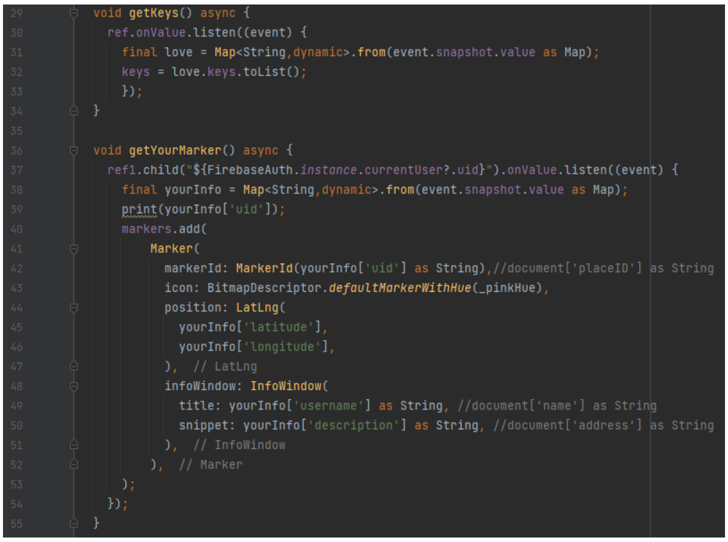
Figure 2. Screenshot of code 1



```
onPressed: () {
  FirebaseDatabase.instance.ref().child("users/${FirebaseAuth.instance.currentUser!.uid}").update({
    "latitude": _location?.latitude?.toDouble(),
    "longitude": _location?.longitude?.toDouble()
  });
},
```

Figure 3. Screenshot of code 2

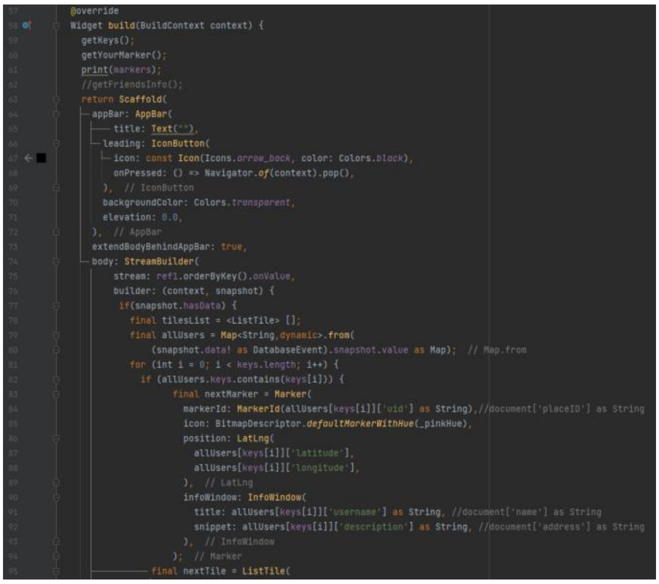
Figure 2 and Figure 3 use the Flutter Package named Location to obtain the most recent location of the user. It grabs the latitude and longitude coordinates from the device, pushes it to the account's Google Firebase database location. In other words, these pieces of code work in tandem with the database to display your latitude and longitude coordinates.



```
void getKeys() async {
  ref.onValue.listen((event) {
    final love = Map<String,dynamic>.from(event.snapshot.value as Map);
    keys = love.keys.toList();
    });
}

void getYourMarker() async {
  ref1.child("${FirebaseAuth.instance.currentUser?.uid}").onValue.listen((event) {
    final yourInfo = Map<String,dynamic>.from(event.snapshot.value as Map);
    print(yourInfo['uid']);
    markers.add(
        Marker(
          markerId: MarkerId(yourInfo['uid'] as String),//document['placeID'] as String
          icon: BitmapDescriptor.defaultMarkerWithHue(_pinkHue),
          position: LatLng(
            yourInfo['latitude'],
            yourInfo['longitude'],
          ),  // LatLng
          infoWindow: InfoWindow(
            title: yourInfo['username'] as String, //document['name'] as String
            snippet: yourInfo['description'] as String, //document['address'] as String
          ),  // InfoWindow
        ),  // Marker
    );
  });
}
```

Figure 4. Screenshot of code 3

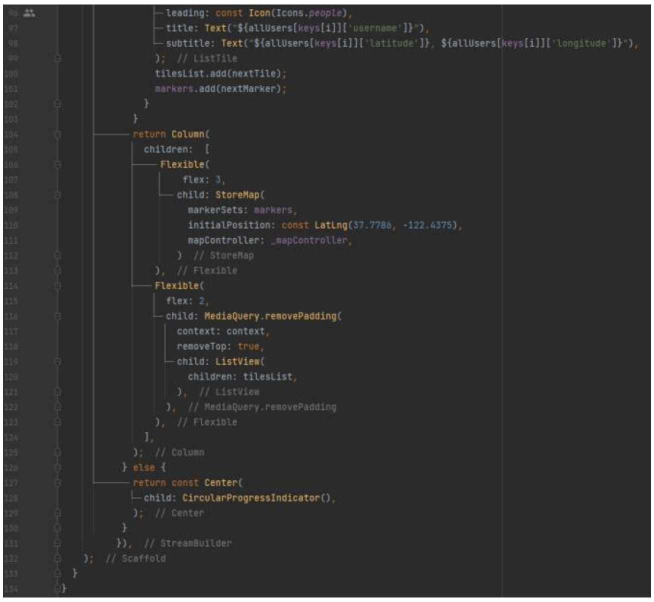
The code in Figure 4 grabs important user information from the database, including their uid, latitude and longitude coordinates, username, and description. It continuously updates the information using a StreamBuilder.



```
57  @override
58  Widget build(BuildContext context) {
59    getKeys();
60    getYourMarker();
61    print(markers);
62    //getFriendsInfo();
63    return Scaffold(
64      appBar: AppBar(
65        title: Text(""),
66        leading: IconButton(
67          icon: const Icon(Icons.arrow_back, color: Colors.black),
68          onPressed: () => Navigator.of(context).pop(),
69        ),  // IconButton
70        backgroundColor: Colors.transparent,
71        elevation: 0.0,
72      ),  // AppBar
73      extendBodyBehindAppBar: true,
74      body: StreamBuilder(
75          stream: ref1.orderByKey().onValue,
76          builder: (context, snapshot) {
77           if(snapshot.hasData) {
78             final tilesList = <ListTile> [];
79             final allUsers = Map<String,dynamic>.from(
80                 (snapshot.data! as DatabaseEvent).snapshot.value as Map);  // Map.from
81             for (int i = 0; i < keys.length; i++) {
82               if (allUsers.keys.contains(keys[i])) {
83                    final nextMarker = Marker(
84                      markerId: MarkerId(allUsers[keys[i]]['uid'] as String),//document['placeID'] as String
85                      icon: BitmapDescriptor.defaultMarkerWithHue(_pinkHue),
86                      position: LatLng(
87                        allUsers[keys[i]]['latitude'],
88                        allUsers[keys[i]]['longitude'],
89                      ),  // LatLng
90                      infoWindow: InfoWindow(
91                        title: allUsers[keys[i]]['username'] as String, //document['name'] as String
92                        snippet: allUsers[keys[i]]['description'] as String, //document['address'] as String
93                      ),  // InfoWindow
94                    );  // Marker
95                 final nextTile = ListTile(
```

Figure 5. Screenshot of code 4

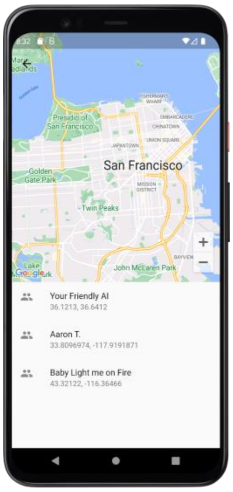
Shown in Figure 5, the StreamBuilder starting from Line 74 streams all the information of the people on the app. After, an “if “statement is made to filter out all the people that aren’t your friends. The StreamBuilder provides a constant stream of constantly updating friend information.



```
                   leading: const Icon(Icons.people),
                   title: Text("${allUsers[keys[i]]['username']}"),
                   subtitle: Text("${allUsers[keys[i]]['latitude']}, ${allUsers[keys[i]]['longitude']}"),
                 );  // ListTile
                 tilesList.add(nextTile);
                 markers.add(nextMarker);
               }
             }
             return Column(
               children:  [
                 Flexible(
                   flex: 3,
                   child: StoreMap(
                     markerSets: markers,
                     initialPosition: const LatLng(37.7786, -122.4375),
                     mapController: _mapController,
                   )  // StoreMap
                 ),  // Flexible
                 Flexible(
                   flex: 2,
                   child: MediaQuery.removePadding(
                     context: context,
                     removeTop: true,
                     child: ListView(
                       children: tilesList,
                     ),  // ListView
                   ),  // MediaQuery.removePadding
                 ),  // Flexible
               ],
             );  // Column
           } else {
             return const Center(
               child: CircularProgressIndicator(),
             );  // Center
           }
          }),  // StreamBuilder
    );  // Scaffold
  }
}
```

Figure 6. Screenshot of code 5

Figure 6 displays the StreamBuilder's constant stream of updating information displayed on the bottom of the Google Map. This information includes information about their name and their latitude and longitude as shown in Figure 7.

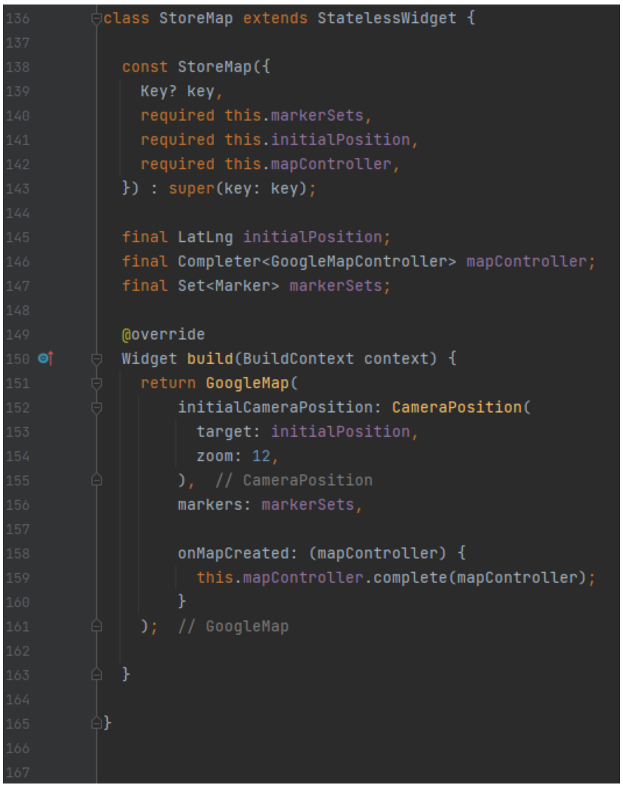


Your Friendly AI
36.1213, 36.6412

Aaron T.
33.8096974, -117.9191871

Baby Light me on Fire
43.32122, -116.36466

Figure 7. Screenshot of name, latitude, and longitude



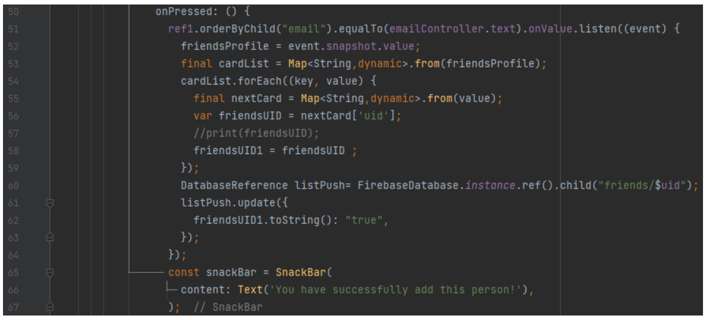
```
class StoreMap extends StatelessWidget {

  const StoreMap({
    Key? key,
    required this.markerSets,
    required this.initialPosition,
    required this.mapController,
  }) : super(key: key);

  final LatLng initialPosition;
  final Completer<GoogleMapController> mapController;
  final Set<Marker> markerSets;

  @override
  Widget build(BuildContext context) {
    return GoogleMap(
        initialCameraPosition: CameraPosition(
          target: initialPosition,
          zoom: 12,
        ),  // CameraPosition
        markers: markerSets,

        onMapCreated: (mapController) {
          this.mapController.complete(mapController);
        }
    );  // GoogleMap

  }

}
```

Figure 8. Screenshot of code 6

Figure 8 displays the code that starts by setting up the map with its initial position (where the map initially shows when first launched). Then, it places markers on the map with the information from the StreamBuilder.

In conclusion, the location is grabbed by the device sending its own location to the Google Firebase database [10]. This database then constantly updates and sends information to the app, which displays this information on the bottom of the map.

## Friend Users:



```
onPressed: () {
  ref1.orderByChild("email").equalTo(emailController.text).onValue.listen((event) {
    friendsProfile = event.snapshot.value;
    final cardList = Map<String,dynamic>.from(friendsProfile);
    cardList.forEach((key, value) {
      final nextCard = Map<String,dynamic>.from(value);
      var friendsUID = nextCard['uid'];
      //print(friendsUID);
      friendsUID1 = friendsUID ;
    });
    DatabaseReference listPush= FirebaseDatabase.instance.ref().child("friends/$uid");
    listPush.update({
      friendsUID1.toString(): "true",
    });
  });
  const snackBar = SnackBar(
    content: Text('You have successfully add this person!'),
  );  // SnackBar
```

Figure 9. Screenshot of code 7

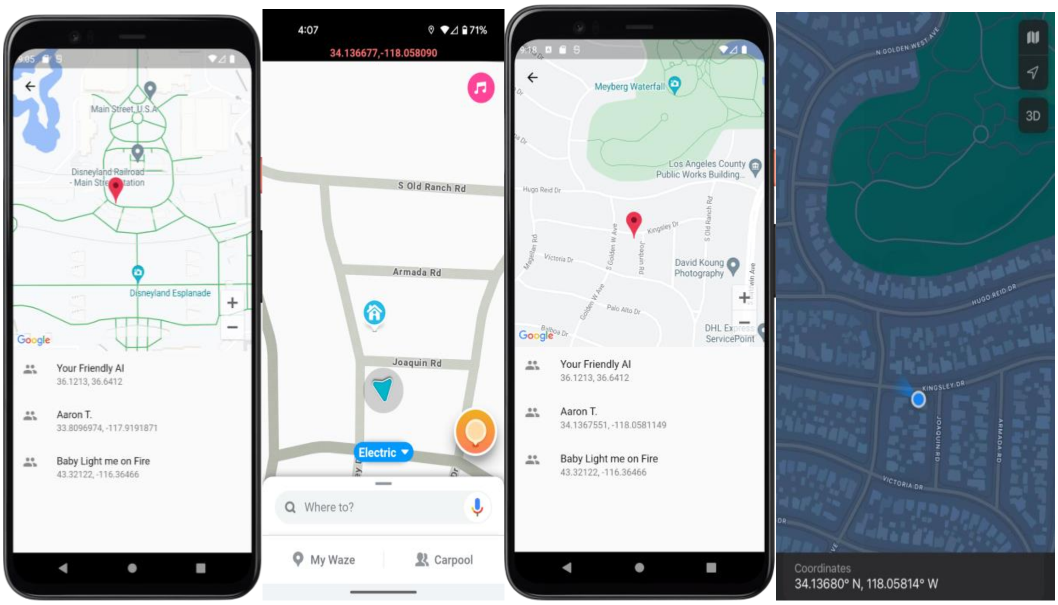
The code displayed in Figure 9 allows users to type the information of their friend’s name and email to make them their friend. After the request is sent, it adds that friend in the user’s Google Firebase database.

# 4. EXPERIMENT

## 4.1. Experiment 1

This solution solves the problem of locating your loved ones by accurately pinpointing their position. This experiment aims to determine how accurate this position is displayed compared to other mapping programs such as Apple Maps and Waze [9].

This experiment was carried across 3 different devices: An LG G5, a Google Pixel 4a, and an Iphone XS Max as shown in Figure 10. Through the process of testing the app on multiple phones, many points of feedback were received, all of which are to be implemented in the future.



Your Friendly AI
36.1213, 36.6412

Aaron T.
33.8096974, -117.9191871

Baby Light me on Fire
43.32122, -116.36466

34.136677,-118.058090

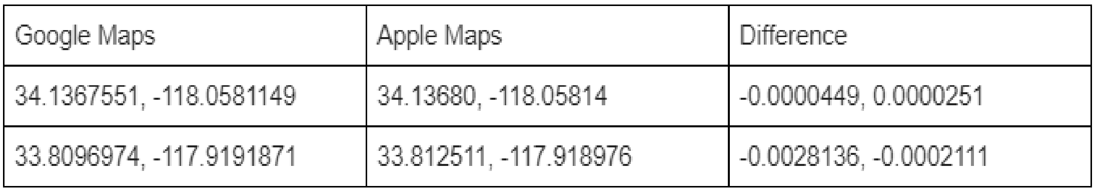
Your Friendly AI
36.1213, 36.6412

Aaron T.
34.1367551, -118.0581149

Baby Light me on Fire
43.32122, -116.36466

Coordinates
34.13680° N, 118.05814° W

Figure 10. Screenshots of of locations



| Google Maps | Apple Maps | Difference |
|---|---|---|
| 34.1367551, -118.0581149 | 34.13680, -118.05814 | -0.0000449, 0.0000251 |
| 33.8096974, -117.9191871 | 33.812511, -117.918976 | -0.0028136, -0.0002111 |

Figure 11. Table of different location between Google Maps and Apple Maps

From Figure 11, it is clear that the latitudes and the longitude measurements of Google and Apple Maps do not differ by much.

In the beginning of the experiment, it was hypothesized that the coordinates of the Google Maps and Apple Maps services would be almost identical, and that hypothesis was somewhat true. With the deviations between the latitudes and longitudes measured never going above 0.003 (due to an outlier), it is safe to say that the two map services are accurate when compared to one another. The results are proven to be legitimate by the corroboration of the multiple sources, as well as evidence from other papers backing up this claim as well.

## 5. RELATED WORK

In the paper Zandbergen, Paul A., and Sean J. Barbeau. "Positional accuracy of assisted GPS data from high-sensitivity GPS-enabled mobile phones." presents an empirical analysis of the positional accuracy of location data gathered using a high-sensitivity GPS-enabled mobile phone [11]. The performance of the mobile phone is compared to that of regular recreational grade GPS receivers.

In the paper Payne, Rap. "Developing in Flutter." Beginning App Development with Flutter [12]. Apress, Berkeley. The paper discusses how flutter enables us to create apps that run on the Web, on desktop computers, and on mobile devices (which seems to be the main draw). This paper is very useful for the developers who are new to the flutter framework

In the paper Khawas, Chunnu, and Pritam Shah. "Application of firebase in android app development-a study" [13]. They show the application of Firebase with Android and aims at familiarizing its concepts, related terminologies, advantages and limitations. tries to demonstrate some of the features of Firebase by developing an Android app.

## 6. CONCLUSIONS

The safety of our loved ones, especially our elderly, is never a given. Loved Ones Locator provides a way to know that they are safe, or in the event that they are not, be able to locate them. The experiment tested whether the location shown on the device is accurate or not. This accuracy was measured by comparing the latitude and longitude of the user measured on the Google Map to those measured on other apps, including Apple Maps and Waze. Google Maps, Apple Maps, and Waze were used in the experiment because they are the most-used GPS apps [15].

Since the app shares very personal information such as name, email, and location information, it is of the utmost importance to ensure that this information cannot be obtained easily by the wrong people. Additionally, the way the information is transferred between the frontend and the database can be improved, making it more optimized. This will result in faster communication between the frontend of the app and the backend Firebase Database. As well as the above changes, there are some Quality of Life changes to be made as well. This will make the user experience more enjoyable, as well as simpler to navigate the menus.

To ensure the safety of the users' information, the method of data transfer and storage between the app and the database needs to be encrypted. To optimize the data transfer between the frontend and backend, redundancies can be eliminated from the code. For the quality of life improvements, a function could be added that lets the users edit their account data, such as their username and a confirmation pop-up confirming that users have created an account.

### REFERENCES

[1] Tolman, Brian W., et al. "The GPS toolkit… open source GPS software." Proceedings of the 17th International Technical Meeting of the Satellite Division of the Institute of Navigation (ION GNSS 2004). 2004.

[2] Moroney, Laurence. "The firebase realtime database." The Definitive Guide to Firebase. Apress, Berkeley, CA, 2017. 51-71.

[3] Rajendran, G., M. Arthanari, and M. Sivakumar. "GPS Tracking Simulation by Path Replaying." International journal of innovative technology & creative engineering 1.1 (2011): 20-26.

[4] McNeff, Jules G. "The global positioning system." IEEE Transactions on Microwave theory and techniques 50.3 (2002): 645-652.

[5] Zamir, Amir Roshan, and Mubarak Shah. “Accurate image localization based on google maps street view.” European Conference on Computer Vision. Springer, Berlin, Heidelberg, 2010.

[6] Punja, Shafik G., and Richard P. Mislan. "Mobile device analysis." Small scale digital device forensics journal 2.1 (2008): 1-16.

[7] Li, He, and Lai Zhijian. “The study and implementation of mobile GPS navigation system based on Google Maps.” 2010 International Conference on Computer and Information Application. IEEE, 2010.

[8] Chatterjee, Nilanjan, et al. "Real-time communication application based on android using Google firebase." Int. J. Adv. Res. Comput. Sci. Manag. Stud 6.4 (2018).

[9] Sathiyanarayanan, Mithileysh, and Tobias Mulling. “Map navigation using hand gesture recognition: A case study using myo connector on apple maps.” Procedia computer science 58 (2015): 50-57.

[10] Moroney, Laurence. "An Introduction to Firebase." The Definitive Guide to Firebase. Apress, Berkeley, CA, 2017. 1-24.

[11] Zandbergen, Paul A., and Sean J. Barbeau. “Positional accuracy of assisted GPS data from high-sensitivity GPS-enabled mobile phones.” The Journal of Navigation 64.3 (2011): 381-399.

[12] Singh, Mohit, and G. Shobha. "Comparative Analysis of Hybrid Mobile App Development Frameworks."

[13] Khawas, Chunnu, and Pritam Shah. “Application of firebase in android app development-a study.” International Journal of Computer Applications 179.46 (2018): 49-53.

[14] Ahn, BoHyun. "Life Alert System for vulnerable adults who live alone." (2016).

[15] Pack, Michael, and Nikola Ivanov. “Are you gonna go my Waze?.” Institute of Transportation Engineers. ITE Journal 87.2 (2017): 28.