Family Trail: A GPS Tracking Mobile Application

Aadhavan Anbuchezhian, Mark Folan

May 2017

Group Project

Submitted in part fulfilment for the degree of
B.Sc. Computing in Digital Forensics and Cyber Security
School of Informatics and Engineering,
Institute of Technology Blanchardstown,
Dublin, Ireland

Declaration

We hereby certify that this material, which we now submit for our 3rd year group project leading to the award of Degree in **B.Sc. in Computer Science in Cyber Security and Digital Forensics** in the Institute of Technology Blanchardstown, is entirely our own work except where otherwise stated and has not been submitted for assessment for an academic purpose at this or any other academic institution other than in partial fulfilment of the requirements of that stated above.

Signed: ________________________ Aadhavan Anbuchezhian Dated: 18_/5_/2017_

Signed: ________________________ Mark Folan Dated: 18/5/2017

Acknowledgement

Group Members Aadhavan Anbuchezhian and Mark Folan would like to thank supervisors Oxana Serada and Mark Cummins for their continuous support and guidance through the duration of the year.

Table of Contents

Abstract

This report covers a GPS-based tracking system for mobile and web based platforms . The Goal of the project is two deliver a solution for family and especially parents to track their children. It’s achieved by employing a three-module implementation on android and web platforms. One android and a Web application used by parents and another used on children’s phone.

1 Introduction

The project application focuses on delivering a complete web and mobile based tracking for families. Knowledge on people's locations are quite important especially for a family that consists of people working and studying away from their home. It aims to provide a 24/7 service for families that desire for a simple yet featureful application. The project as a whole is separated into two sets of implementation. The backbone of the project is located on the Android device of the user that is to be tracked (e.g. a child), while the rest of the application is dedicated for the tracker (e.g. parents).

The application framework is specifically developed for Android OS based mobile devices which involves the use of Google Maps Application Programming Interface (API) and Global Positioning System (GPS) service provided by default on mobile networks which can also use the GPS receivers.

2 Literature Review

2.1 General Research

Global Positioning System consists of multiple satellites, a network that forwards highly accurate information regarding the position of an object in the world. The coordinates are then received by GPS receivers for geographical location. There are three different segment to the GPS system: Space Segment, Control Segment and User Segment (Peter, 1999). The Space Segment which consists of all the satellites that send radio signals to the control segment which are located around the world. Tracking Stations that are located around the world receives these signals and send clock data and additional information to the Space Segment. The User Segment (the receivers) then obtains raw information from the Space Segment which it converts to parameters such as positions and time. The GPS System that is currently most utilized all around the globe is the American navigation system which is owned by the American government and operated by the American Department of Defence (DOD). (Peter, 1999). There are other satellite navigation systems that are used around the world such as the Russian based GLONASS and Europe based Galileo.

2.2. Official materials

The Official documentation from Android developers provided the project a head start into the development of 'Location-aware' application. (Developer Android, 2017). It includes tutorials for all levels of coding, from the basics of Android JAVA programming to other distinctive parts of applications. The documentation includes various methods and function as to creating and implementation location based features. The documentation suggested crucial information such as using Google's Play services location API which is preferred over using the Android framework location API. It provided us with examples of code which were helpful in writing several different program sections. It was useful in learning with various code fragments that are present in our application. The functionalities explained in the documentation such as getting the last known location, location updates and adding maps were all integral to our app.

The Android developer's documentation lead us into the Google Official Map API documentation. (Developers Google, 2017). These documentations provided us with functions that make the core of our application. As our application integrates Google Maps API, the guides offered illustrated the activities and classes used by Google Maps API. It introduced objects like MapObjects which holds the map overlay. The configuration of maps inside an application which involved using the Google Developer Console to obtain API key that allowed maps to be used. MapView which is a subclass of Android's View system which allowed the application to perform various actions based on creation (onCreate) and destruction (onDestroy) among others.

Google’s own developer console hub provided important credentials such as the Maps API key. This Maps API key is registered to a Google account which allows for the display and usage of Google Maps. The key is very useful since it allows for confidentiality when using the maps. The map instance used in our application is unique which provided us with some security. (Google Developers Console, 2017).

2.3. Existing implementations

GPS-based Tracking has become an essential function in many people’s lives whether children, family, friends or work related. The need for keeping up to date on individuals’ whereabouts have many uses. GPS applications used for school buses especially became an important requirement in many places around the world. There are many advantages in using GPS that can be very beneficial for both the bus management and child safety. (Sonia, 2015). The tracking system can help organize a safe route for the bus and can allow for quick and easy navigation during crucial hours. It can also facilitate the management of bus schedules making way for easier flow of buses. Efficient routes can also be plotted which can save valuable resources such as fuels.

There are several applications and implementations in the area of the personal tracking based around Android phones. All of these applications aim to provide a solution for basic day-to-day tracking for families and friends. The common goal is achieved mostly with the help of Google Maps API and their location services.

Nick Fox’s tracking application is described as software for developers (Nick, 2014). This implementation consists of one Android application and one web-based tracking service. Various parts of the program include components such as a background service in conjunction with alarm receiver (allows for time management during updates). It also includes a location service function that sends updates to the website periodically using the native async HTTPclient library (Developers Android, 2014).

Another type of implementation from Family Safety Production consists of single app implementation for tracking. (Family Safety P, 2017). The tracking is done through a single interface that tracks other Android phones with the application installed. It is mainly targeted towards the family and children, which allows for a variety of functions. It also includes features such as directions and distance between users for better knowledge on whereabouts. Apps can also sends alerts via text messages with information such as expected time of arrival.

Similar application framework is seen in the Greenalp Communication’s “Real-Time GPS Tracker”. (Greenalp, 2015) Unlike the previous application discussed earlier, this program contains two module framework. The Tracking app records locations in the app and the users are allowed to send locations to others as a link to their website where the location is plotted on a Google map. Users can also use the Android app to add each other to their group, which makes them visible among the group on the map.

Conclusion

In studying the existing solutions for GPS tracking it can be seen that there are several faults and room for improvement on their implementations.

Some implementation can utilize database connection functionalities (Async tasks) which should be replaced with faster technologies such as Android Volley. (Josh, 2013)

In most of applications, there are only one or two modules (mainly app and web). This can cause inefficiency as one application may have to perform all the functions at once. The lack of options is also quite apparent as there is no backup tracking interface if the user is not able to access the Android program. Similarly with the app to web solution if the user is not able to access a website, there is no second option for live tracking.

As mentioned above, the inefficiency due to single implementation can cause high memory and battery usage, having to render maps and perform various background activities such as sending to database.

The current project aims to provide a possible solution to this problem by implementing a three-module framework. This type of functional structure promotes efficiency by separating workloads while maintaining similar feature sets. A lightweight application ("GPSClient") especially for children can provide a silent and efficient tracking while allowing parents to have the full experience with two options for backup and flexibility – Family Trail application and website tracking.

3 Project planning/ Proposal

3.1 Project Introduction

This Project is a GPS Tracking Application. The team involved in the project consists of Aadhavan Anbuchezhian and Mark Folan. It was supervised by Oxana Sereda.

The Application is specifically designed for parents to keep track of their children when they are at venues like school. The Application's primary target audience are parents who want to monitor their young children's locations periodically and especially when they are far away.

What the project is about

The group decided to design an application for parents to monitor their children's location. Our application will make use of the spread of technology, which means for the first time in history it is affordable to give children advanced technology that can make use of the Internet, GPS satellites as well as mobile communications.

Advances in production as well as nano-technology, means it is cheaper than ever to purchase Android-based devices such as phones, tablets, or smartwatches. Many manufacturers have targeted their products for use by children, making them extra durable and colourful.

It was planned to design an inexpensive system that would allow parents install an application on a smartwatch, for example, that will stay in contact with our servers.

The parents can access their accounts through a separate mobile application or via our website. Through these interfaces, they will have access to turn on and off device tracking, and set up a number of options and features such as a Safe Zone, which, if tracking is enabled, upon leaving the zone chosen, the parents or children will be alerted, depending on the option chosen.

The children can also open the application, and using a simplified user interface, press a big red button to contact the parents in the case of an emergency. This again will alert the parents by whichever method they have chosen, SMS messages or emails bring the most common. The parents can call the child straight away, and using the maps and directions provided, give the child directions to a bus stop, get home or they could drive to collect them.

Why this project was chosen

The group chose to design a GPS tracking application as it combines many and varied aspects the group have studied in college. Also, since it is aimed at monitoring children it needs to be a secure application, so unwanted 'actors' do not get access to their locations.

In researching the topic the group came across a number of forums where parents were trying to find out what system is best for monitoring their vulnerable children, for example children with disabilities. There was no definitive answers available, so the group decided to address the problem and provide a secure application that provides a wide range of useful features. The uniqueness of this project stems from the 3-module implementation of the application. The users are provided with options for tracking individuals.

The group decided to add the web interface, so the parents can easily access the program from home or work without needing their phone. This also gives the parents another platform for ease of access and a pseudo backup application.

3.2 Deliverables/Objectives

The Deliverables/objectives that are to be presented for the project are as follows:

- Create a plan for the project.
- Research and write about different parts of the project.
- Produce an analysis and design documentation.
- Test and develop model versions of the application.
- Finalise and testi the application for daily use.
- Include a literature review based on research.

3.3 Tasks Involved

Feasibility – Expectations and Difficulties

Expected Results for the Project

- Creating a Tracking App that works as intended.

- App should be easy to use and looks clean.
- Should be relatively Secure and Bug Free.

Expected Work

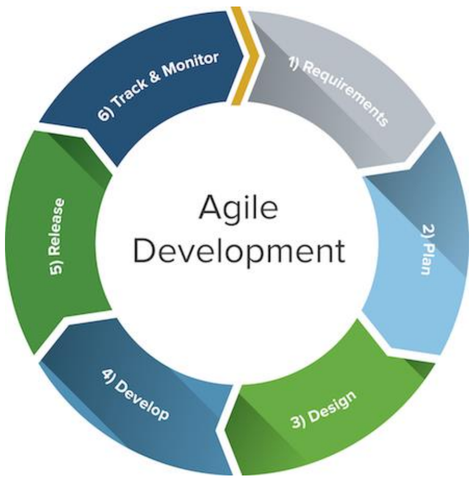
- Being able to use IDE to create an full-fledged application
- Being able to work with Android OS and its specific programming configurations.
- Being able to understand programming languages and use programming languages such as Java and PHP
- Being able to utilize Databases like MySQL and manage a website.

Possible Difficulties

- Depth of Programming related to android OS may be huge.
- Unable to fulfil all the objectives of the application due to technical constraints.
- Difficulties may arise regarding testing as GPS services can be unreliable.

3.4 Methodology – AGILE

Agile Development is the methodology that closely parallels our development cycle. This model consists of six different stages.



Figures 1 – Agile Model

Requirements
The requirements that need to be fulfilled are outlined during this phase. Main objectives of the application and optional functions are decided.

Plan
The team discusses the work schedule and prioritizes features and objectives of the application development. Partitioning work involved in development into smaller sections (renting and managing a website) among members and collaborating on the core part of the project (Core mobile application).

Design
Project members involved in designing various parts of the program start to design early versions of the software. They spend significant time in modelling the program that meet the requirements that were placed during the inception of the project

Development
The models that were created during the design phase are materialized as per the suggestions from objectives. Technical problems and constraints are handled and the application is developed by avoiding such situations without having to compromise the goals of the project.

Testing & Release
The final version of the program is selected and tested against the requirements list. Testing for errors, bugs and security flaws are also performed. The data from user experience are also collected in order to improve any minor areas of the application. The Final product which is ready for use is installed on the platforms it is intended to be used on (mobile devices).

Track and Monitor
This phase is more or less an optional phase in our project. The released app is monitored and tracked continuously during its usage. Monitoring of hosted website to avoid downtimes and responding to possible technical difficulties or attacks on the infrastructure. Adding more features that are were initially not included in the finalized version for more utility.

Advantages and Disadvantages of this model

There are notable benefits when using this model:

- Changes are easily adapted during the development phase, as the name suggests the cycle allows flexibility in work priorities (developing for web and mobile platforms).

- Final product features are not set in stone. As the project is evolving a variety of features can be added or dropped depending on the conditions around the implementation. This prevents restricted and a single path bound development towards end product.

- Highlights good team work, where members in the team are able to work able on different parts of the project without compromising efficiency. Responsibility based around work allotted is clearly present among the team.

As with any other development models there are cons for the agile model:

- Flexibility in project goals can cause massive deviation from the initial vision of the project. Uncertainty around the final result allows for time constraints at end of the development phase which can impact the delivery of the project. This was a central aspect of our project since our ideas for the implementation were constantly changing especially deciding the focus of the project (web and mobile application) and time allocation for prioritized tasks.

- Experience requirement among team members is a crucial part in the model. With a team of two members which are usually standard for agile, individuals need to be highly skilled among the areas allotted for them on the project. This aspect was a challenge for us to tackle. Between learning mobile application development and implementation of Google's application interfaces on two sets of platform (web and mobile).

- Engagement among team members is vital for this model. As a constantly evolving project with features added "on the go", a constant communication and collaboration is required. Constant evaluation of features sets and scenario bound characteristics are an important part of the development. This was an element of development which was hard to tackle during our project. Changes made to the particular parts of code sometimes caused issues with other services on the application which called for quick communication and carrying out a resolution for the issues.

3.5 Tools/Applications Used

The project began with the discussion of the environment that application will be developed on. Android Studio which is the official implementation of an Integrated Development Environment (IDE) for Android OS. From researching its features, it proved to be the best tool for developing Android applications which are written in JAVA. It offers vast variety of functions such as code debugging, building and deploying of code and highly flexible code editor. It also gives a built-in emulator software package which is very useful for testing the application in a quick and efficient way. It allowed testing the GPS functions while being stationary. (Developer android, 2013)

3.6 Plan and Outcomes

The following section explains the features that were implemented as per the project plan and the features that were not added for various reasons.

Features that were included and working successfully were:

- GPSClient sending coordinates from the phone to database table.
- Shutdown/Kill alert message is sent from the Trackee's Phone to Tracker.
- FamilyTrail Application receiving Trackee's coordinates from the database and plotting the markers on the map component.
- Getting the exact location address of the child in the FamilyTrail app.
- Google Maps application linking on Android phone for directions service.
- Website able to host a login system for tracking.
- Tracking page dedicated to show locations of the users on the map on a timely basis.
- Directions obtained between Tracker and Trackee.

Partially Working features

-

Challenges we faced / overcame / could not get by.

Time management. Deciding what aspects of the application are most important, and scheduling our time as to get these done first, in case we run into problems and need to spend more time.

4 System Requirements and Specification

4.1 Functional Requirements

Mobile Applications

The “GPSClient” application contains interface-free/lightweight application for the person to be tracked (e.g. child) that can be run on the background. After starting the application, the app can be minimized in to the background. The application will also start a background service for periodically monitoring the status of the phone/app. If the phone or application is about to shutdown/stop, a text message will be sent to the tracker (e.g. parent) as an alert.

The Tracker application called the “Family Trail” equips a concise tracking interface. In the mobile application, the user is first prompted to log into the application. After successful login, the main tracking interface for the application is presented. The Tracking interface will display a map with two different markers. One marker representing the tracker’s location and one representing the trackee’s location. Exact location of the trackee (e.g. a child) can be retrieved by pressing the get location button on the interface. By tapping the marker of the child, the user will be able to get directions to the child from their current location. The project application is linked to Google Maps app which fetches directions. Google Maps application can show various type of travel routes and modes of transport.

Web Application

Website that implements the web based tracking allows for login of the users. The website is composed of several pages for various functionalities. A dedicated tracking page which displays a map is present for tracking the trackee. The map and positions of users are updated on a timely basis. The tracking page also provides markers for both parties of the system. Directions between the two sets of users are fetched automatically and are updated to periodically on the right side of the tracking page.

Diagrams

The following concept diagram shows the basic mechanisms of the project application

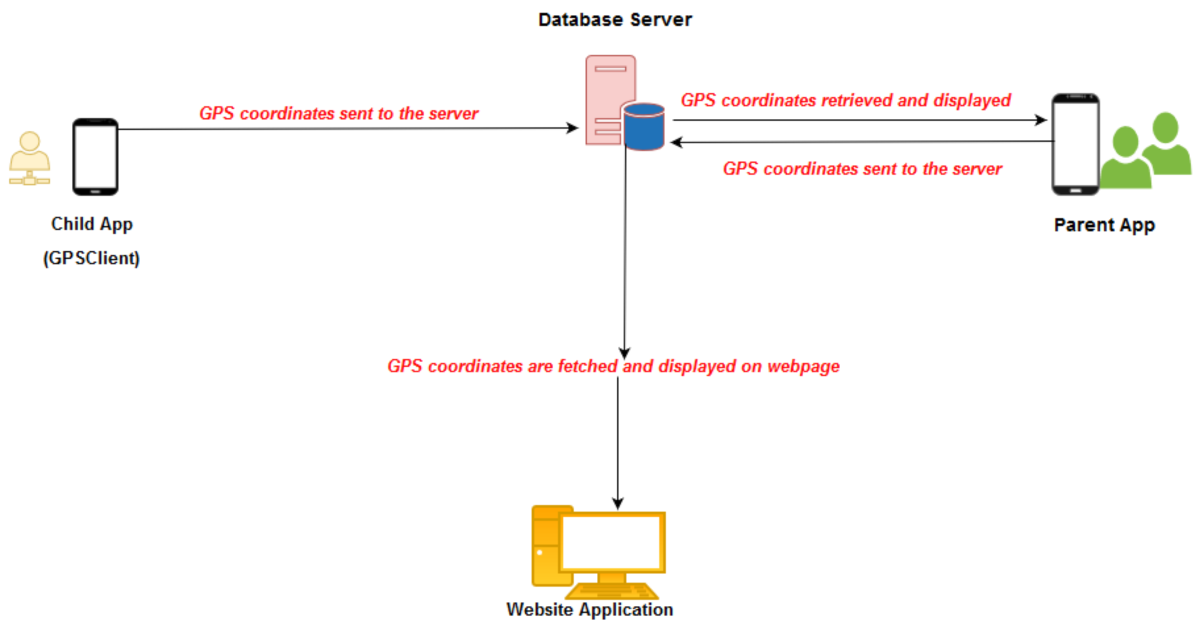

Database Server
GPS coordinates sent to the server
GPS coordinates retrieved and displayed
GPS coordinates sent to the server
Child App
(GPSClient)
Parent App
GPS coordinates are fetched and displayed on webpage
Website Application

Figure 2 – Structure and functions Model

The Client (Trackee) Application Use Case Diagram

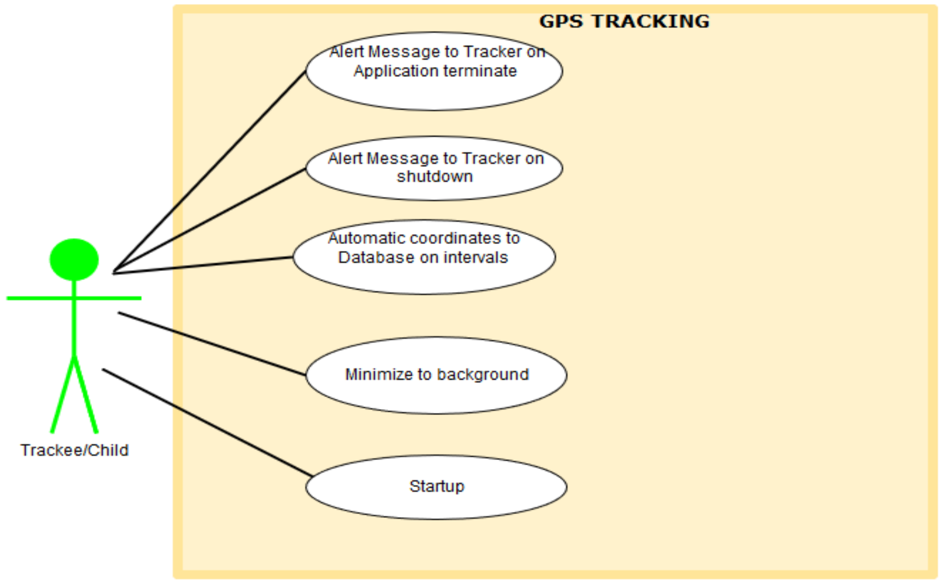

GPS TRACKING
Alert Message to Tracker on Application terminate
Alert Message to Tracker on shutdown
Automatic coordinates to Database on intervals
Minimize to background
Startup
Trackee/Child

Figure 3 – GPS Client Use Case Diagram

The “FamilyTrail” Tracker Application Use case Diagram

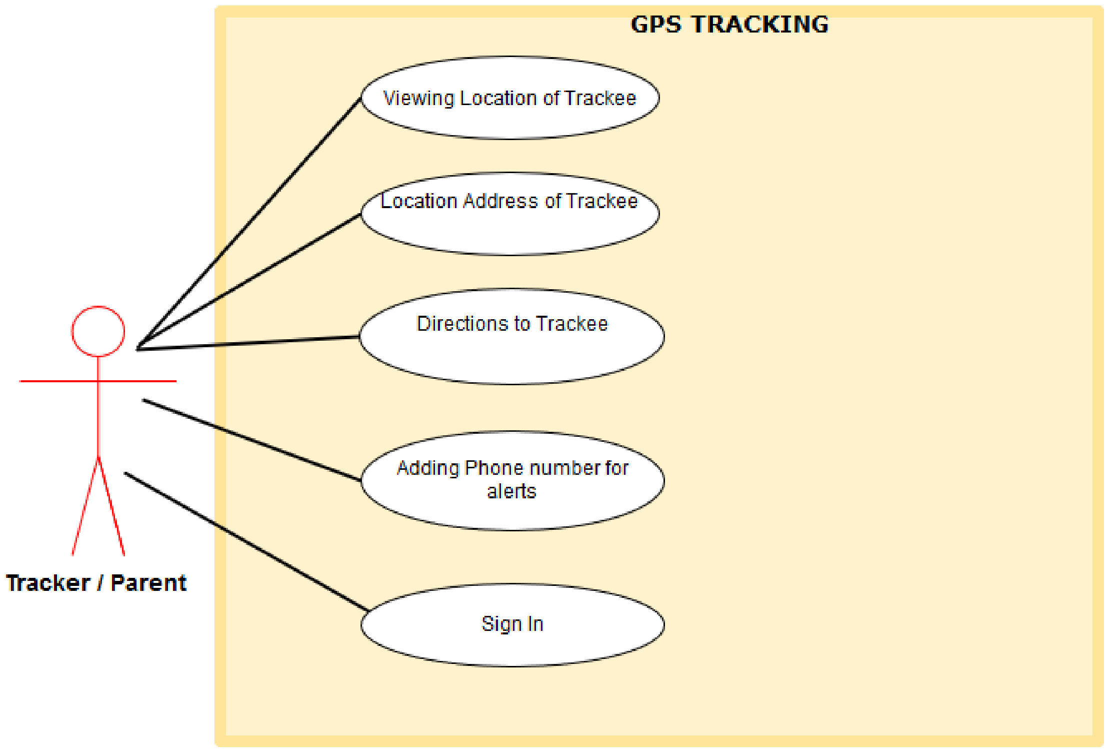

GPS TRACKING
Viewing Location of Trackee
Location Address of Trackee
Directions to Trackee
Adding Phone number for alerts
Sign In
Tracker / Parent

Figure 4 – FamilyTrail Use Case Diagram

The Website Application Use Case Diagram

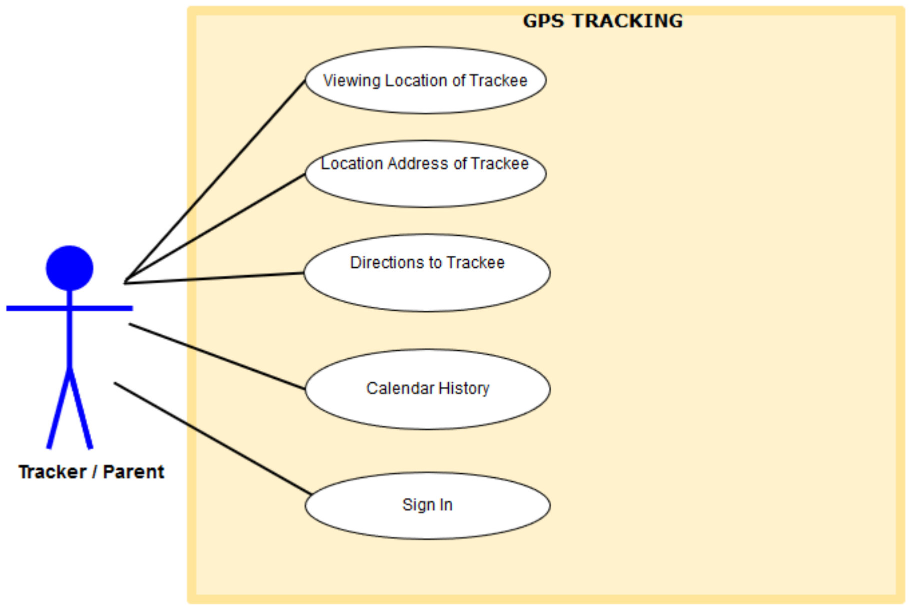


Figure 5 – FamilyTrail-Website Use Case Diagram

4.3 Data Requirements

The MySQL database server contains two different tables (coordinates and coordupdate) inside the GPS schema.

Database table contents/structure is separated into three different columns: id/locno as the primary unique identifier, latitude and longitude as the coordinates to be stored.

Coordinates table (updated by client/trackee)

Column Name	**Description Type**	**Data type**
locno(Primary Key)	an unique id that is present for individual coordinates	int
latitude(only Distinct values)	values for latitude part of the coordinates to be converted	float
longitude(only Distinct values)	values for longitude part of the coordinates to be converted	float

coordupdate table (updated by Parent/Tracker)

Column Name	**Description Type**	**Data type**
id(Primary Key)	a unique id that is present for individual coordinates	int
latitude(only Distinct values)	values for latitude part of the coordinates to be converted	float
Longitude (only Distinct values)	values for longitude part of the coordinates to be converted	float

5 System Design

5.1 Graphical User Interface Design

"GPS Client" Application (Trackee)

This application consist of a lightweight activity/interface. It is simplistic in terms of look and easy to use

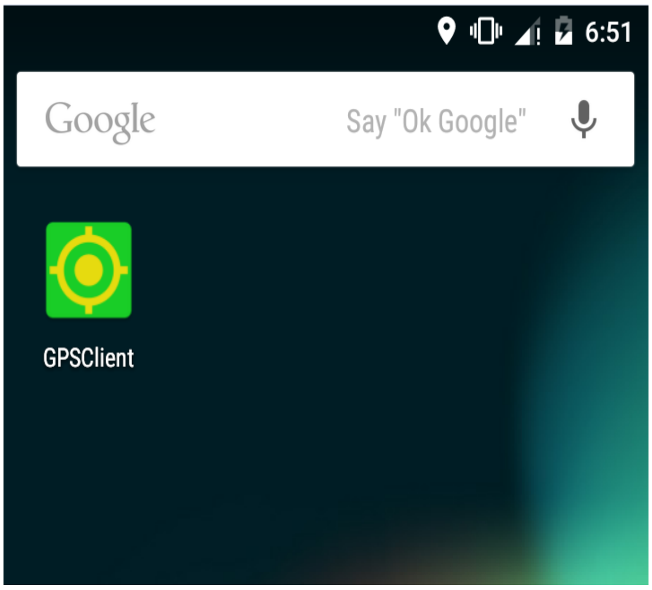
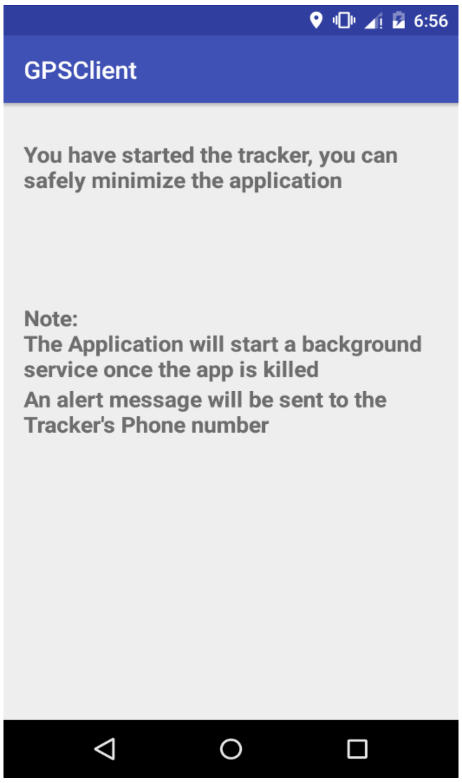


Figure 6 -- App on the Phone's Main Menu

Once the Application is started the Main interface will be displayed which contains information on the app status and notes on activities. As soon as the user sees this activity screen, the application starts sending GPS coordinates to database for tracking.



You have started the tracker, you can safely minimize the application

Note:
The Application will start a background service once the app is killed
An alert message will be sent to the Tracker's Phone number

Figure 7 -- Main Interface for the Application

FamilyTrail Tracker Application

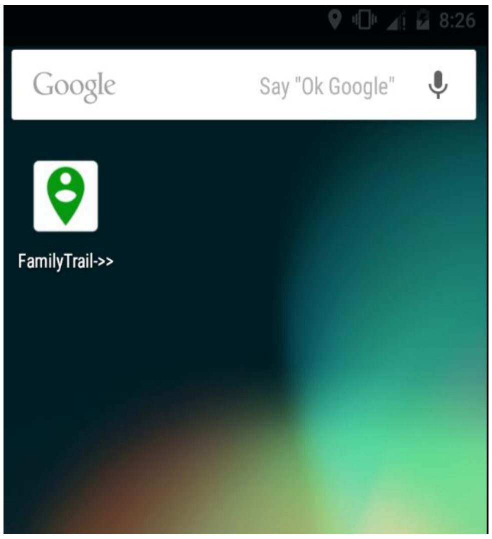
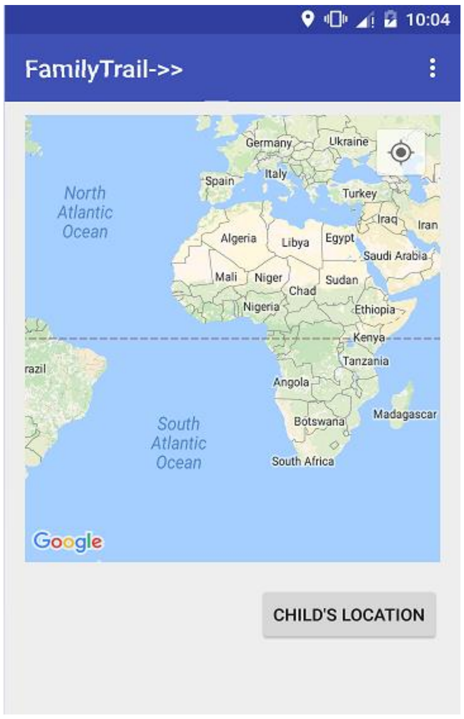


Figure 8 -- Application on the menu screen

When the Application is launched, it displays the tracking map on the main screen. There are several options to use different type of map (satellite/terrain etc.) depending on preference.



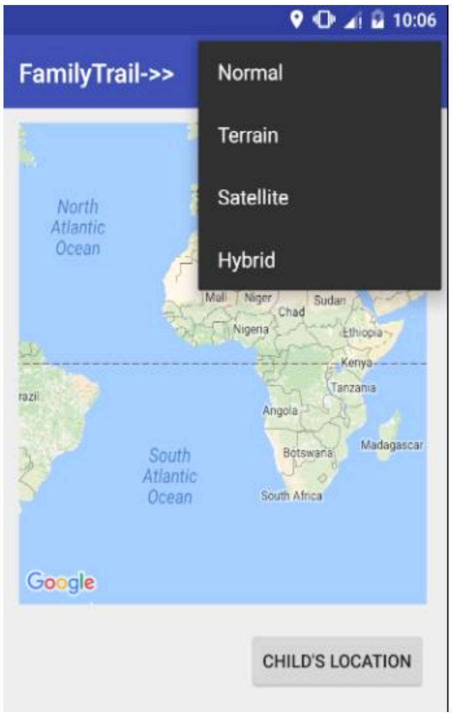
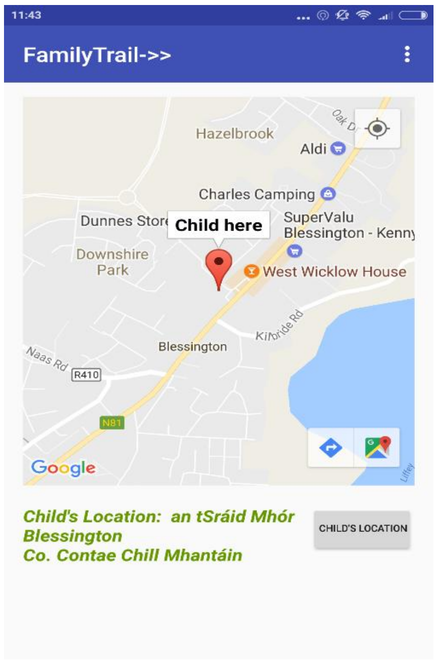


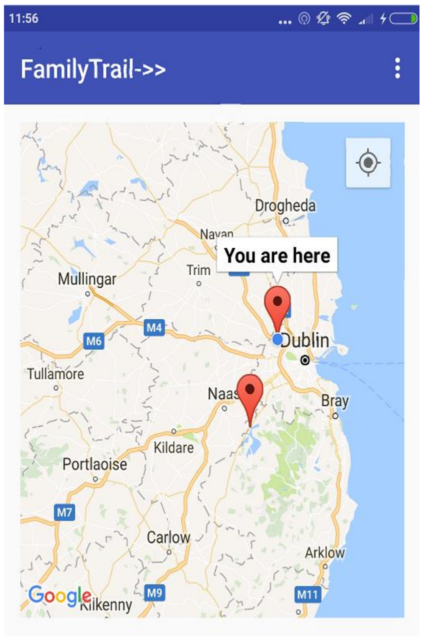
Figure 9 – Main interface of the Map

Figure 10 – Choosing Types of Map

Markers are placed in the map based on the location of the users. Child/Trackee's Location address can also be retrieved by tapping the Child's Location button which will be displayed adjacent to the left of the interface

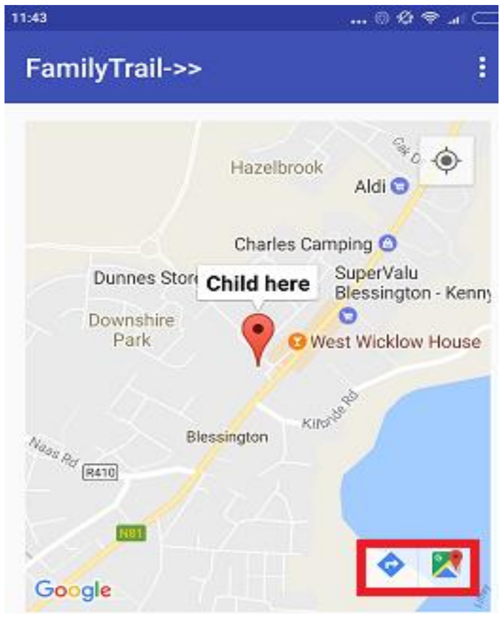


Child's Location: an tSráid Mhór
Blessington
Co. Contae Chill Mhantáin



Figures 11&12 – Markers and Get Location in action

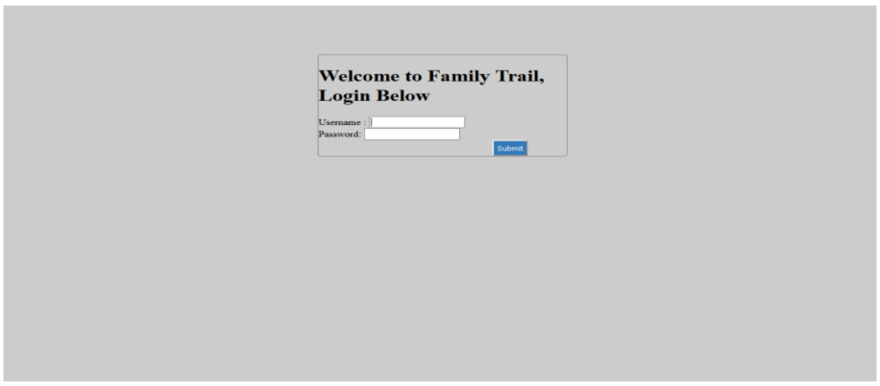
The Application is also linked with Google Maps application which will display directions to the Trackee if needed. (Seen below inside the highlighted rectangle on the lower right corner). The options can either show the exact location on Google Maps application or show the directions to the other user.



Figures 13 – Link to Getting directions

FamilyTrail web Application

When the user visits the website, they are prompted to Login on welcome page.


Welcome to Family Trail, Login Below

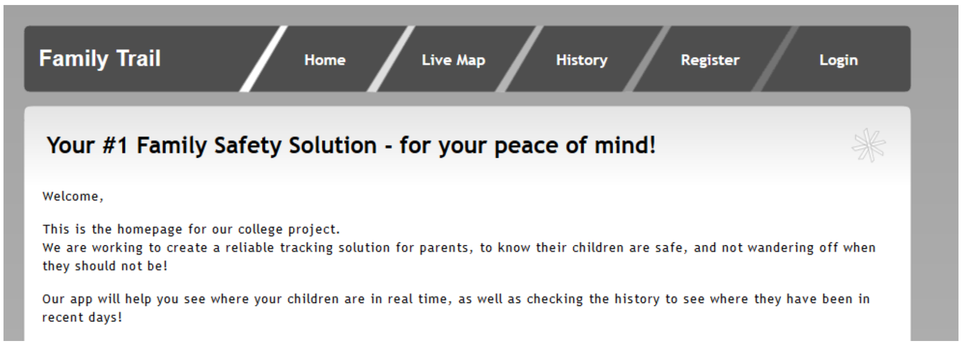
Username :

Password:

Submit

Figure 14 – Link to Getting directions

After the successful login, they are brought to the home page



Your #1 Family Safety Solution - for your peace of mind!

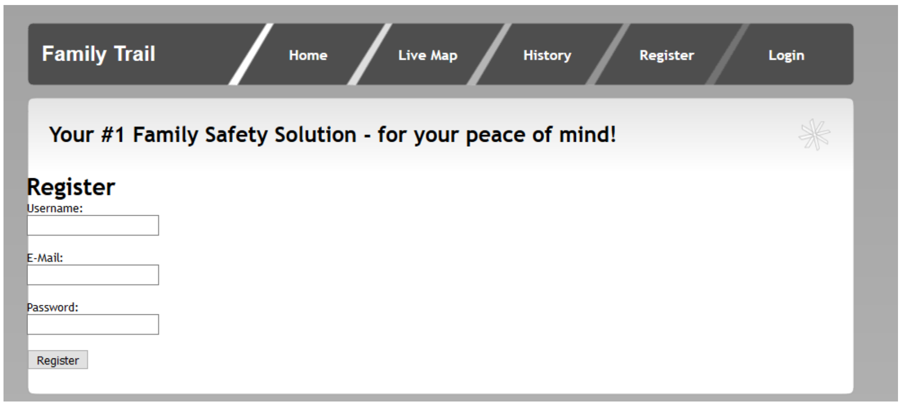
Welcome,

This is the homepage for our college project.
We are working to create a reliable tracking solution for parents, to know their children are safe, and not wandering off when they should not be!

Our app will help you see where your children are in real time, as well as checking the history to see where they have been in recent days!

Figure 15 – Link to Getting directions

There is also a register page where they can register the trackers.



Your #1 Family Safety Solution - for your peace of mind!

Register

Username:

E-Mail:

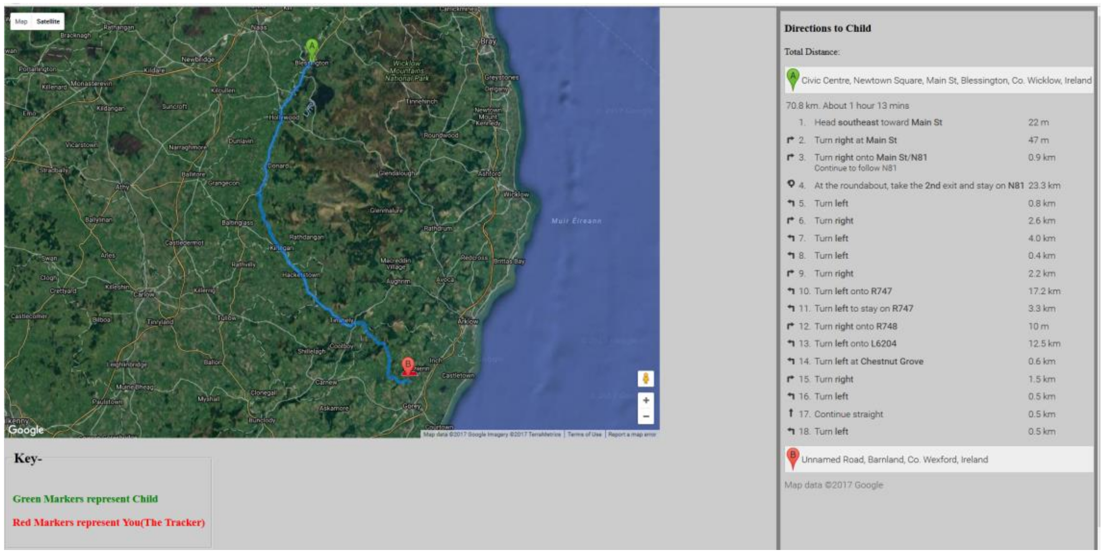
Password:

Register

Figures 16 – Link to Getting directions

There is also a history page where the user can check the location history of the trackee (in testing phase).

The LiveMap link will lead the user to the dedicated tracking page.


Map
Satellite
Google
Key-
Green Markers represent Child
Red Markers represent You(The Tracker)

Directions to Child

Total Distance:

A Civic Centre, Newtown Square, Main St, Blessington, Co. Wicklow, Ireland

70.8 km. About 1 hour 13 mins

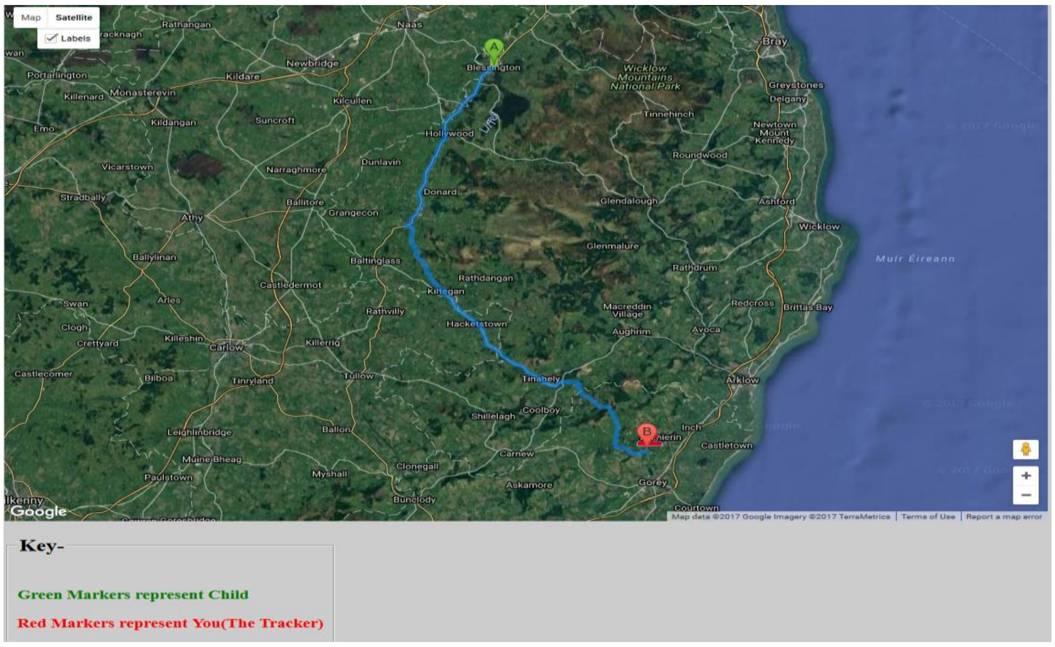
1.	Head **southeast** toward **Main St**	22 m
2.	Turn **right** at **Main St**	47 m
3.	Turn **right** onto **Main St/N81** Continue to follow N81	0.9 km
4.	At the roundabout, take the **2nd** exit and stay on **N81**	23.3 km
5.	Turn **left**	0.8 km
6.	Turn **right**	2.6 km
7.	Turn **left**	4.0 km
8.	Turn **left**	0.4 km
9.	Turn **right**	2.2 km
10.	Turn **left** onto **R747**	17.2 km
11.	Turn **left** to stay on **R747**	3.3 km
12.	Turn **right** onto **R748**	10 m
13.	Turn **left** onto **L6204**	12.5 km
14.	Turn **left** at **Chestnut Grove**	0.6 km
15.	Turn **right**	1.5 km
16.	Turn **left**	0.5 km
17.	Continue straight	0.5 km
18.	Turn **left**	0.5 km

B Unnamed Road, Barnland, Co. Wexford, Ireland

Map data ©2017 Google

Figure 17 – Link to Getting directions

This dedicated tracking page contains a frame of Google Maps which is updated periodically with both sets of users' locations. The Red markers on the map indicate the location of the Tracker and the Green ones indicate the person to be tracked (as clarified by the Key at the bottom of the page).



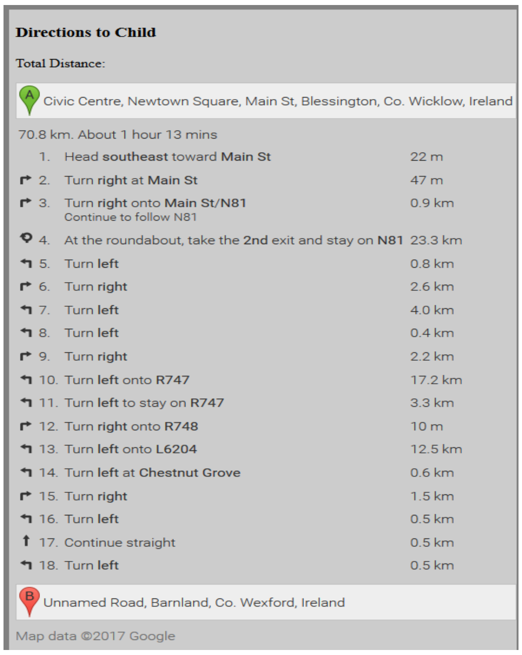
Key-

Green Markers represent Child

Red Markers represent You(The Tracker)

Figure 18 – Map and Markers

On the right side of the web page there is a box frame that updates the directions to the trackee if the locations change.



Directions to Child

Total Distance:

Civic Centre, Newtown Square, Main St, Blessington, Co. Wicklow, Ireland

70.8 km. About 1 hour 13 mins

1.	Head **southeast** toward **Main St**	22 m
↱ 2.	Turn **right** at **Main St**	47 m
↱ 3.	Turn **right** onto **Main St/N81** Continue to follow N81	0.9 km
4.	At the roundabout, take the **2nd** exit and stay on **N81**	23.3 km
↰ 5.	Turn **left**	0.8 km
↱ 6.	Turn **right**	2.6 km
↰ 7.	Turn **left**	4.0 km
↰ 8.	Turn **left**	0.4 km
↱ 9.	Turn **right**	2.2 km
↰ 10.	Turn **left** onto **R747**	17.2 km
↰ 11.	Turn **left** to stay on **R747**	3.3 km
↱ 12.	Turn **right** onto **R748**	10 m
↰ 13.	Turn **left** onto **L6204**	12.5 km
↰ 14.	Turn **left** at **Chestnut Grove**	0.6 km
↱ 15.	Turn **right**	1.5 km
↰ 16.	Turn **left**	0.5 km
↑ 17.	Continue straight	0.5 km
↰ 18.	Turn **left**	0.5 km

Unnamed Road, Barnland, Co. Wexford, Ireland

Map data ©2017 Google

Figure 19 – Directions to the Trackee, updated automatically

5.2 Functional Design and Code Structure Walkthrough

The following section explains the code-based functions that are carried out while the applications are utilized

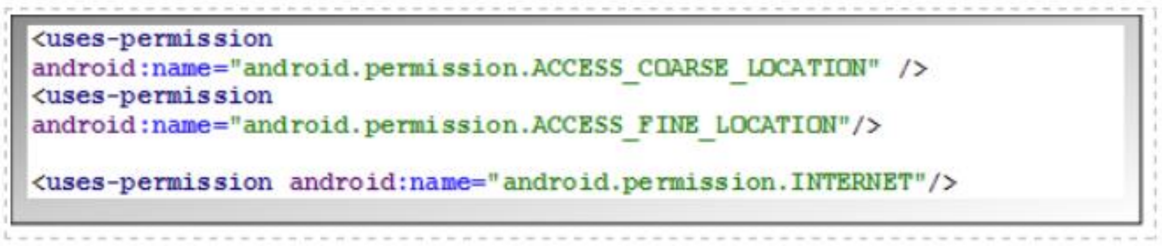
"GPS Client" Application (Trackee)

Requirements for the application:

- Android build API versions 15+(Ice cream Sandwich & above)
- GPS service permissions
- Internet connection

Start-up and the main functions

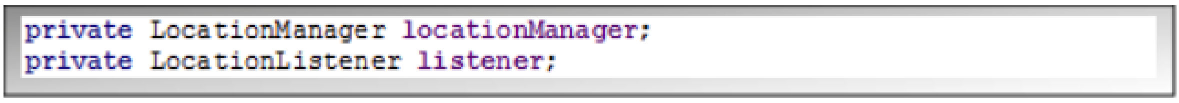
When the application is opened for the first time, the user is required to grant app the permission to use GPS locator service. The code in the Android manifest (an XML file that holds important components for running the app) file allows for this action to be performed. There are two permissions that are present in the file : app's ability to access location info and Internet



```
<uses-permission
android:name="android.permission.ACCESS_COARSE_LOCATION" />
<uses-permission
android:name="android.permission.ACCESS_FINE_LOCATION"/>

<uses-permission android:name="android.permission.INTERNET"/>
```

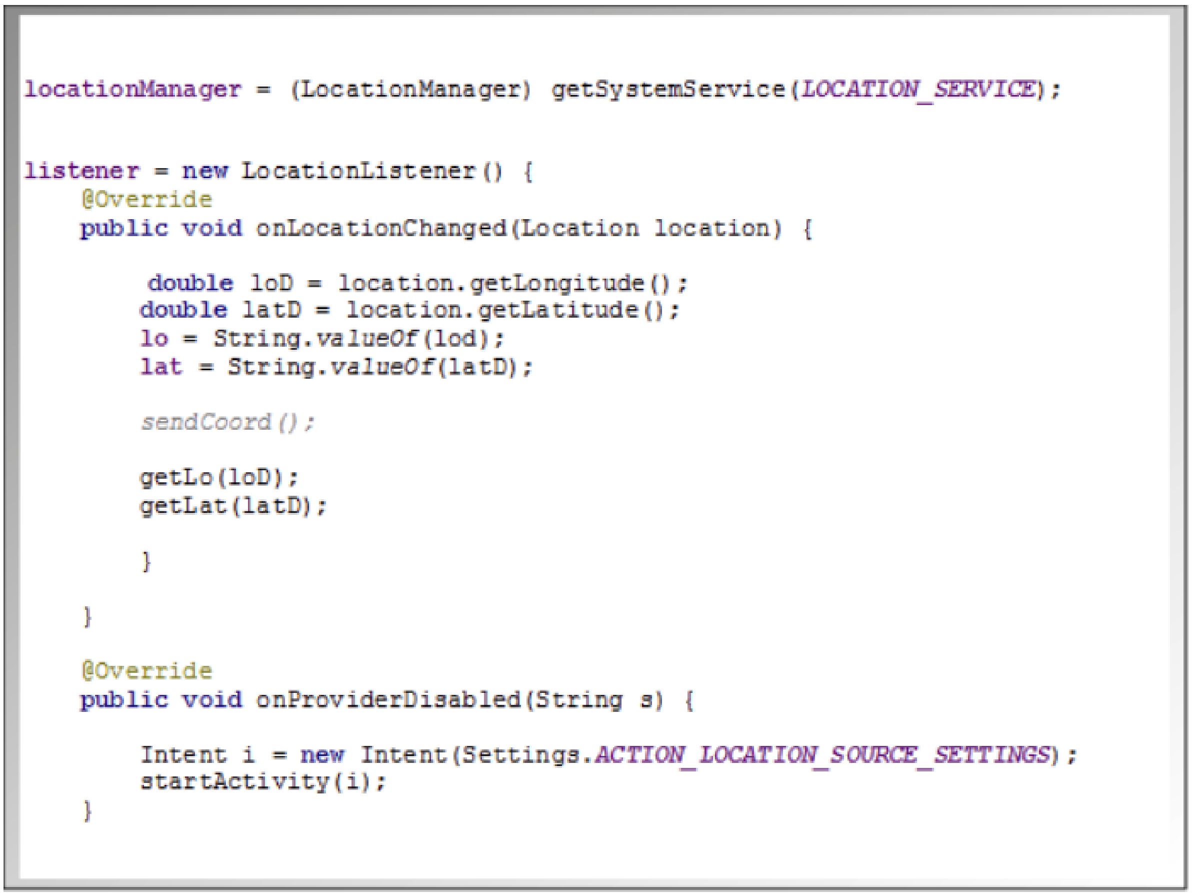
When the User grants the permissions required, the main interface is created using the code in the MainActivity.java file. Basic information such as status of the app and general information are displayed on the main screen as a text.

Two of the important location based functions are assigned : location manager – for gathering location info and location listener for listening to changes.



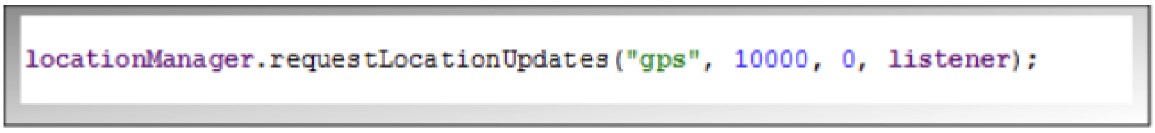
```
private LocationManager locationManager;
private LocationListener listener;
```

The Location Manager is then initialized to start the task of gathering the location info. Two Global Double variables are used for latitude and longitude. (loD, latD). These values are assigned with the current coordinates. The values are converted into string variables to be sent to the database. Sendcoord method is called in the function which will create the format for sending coordinates to the database on a timed basis.



```
locationManager = (LocationManager) getSystemService(LOCATION_SERVICE);


listener = new LocationListener() {
    @Override
    public void onLocationChanged(Location location) {

         double loD = location.getLongitude();
        double latD = location.getLatitude();
        lo = String.valueOf(lod);
        lat = String.valueOf(latD);

        sendCoord();

        getLo(loD);
        getLat(latD);

        }

    }

    @Override
    public void onProviderDisabled(String s) {

        Intent i = new Intent(Settings.ACTION_LOCATION_SOURCE_SETTINGS);
        startActivity(i);
    }
```

The following code, which is a part of the listener function assigned earlier, sends the actual coordinates to the location manager every 10 seconds (in the instance). However this can perform its function if permissions are granted by the user



```
locationManager.requestLocationUpdates("gps", 10000, 0, listener);
```

Database functionalities

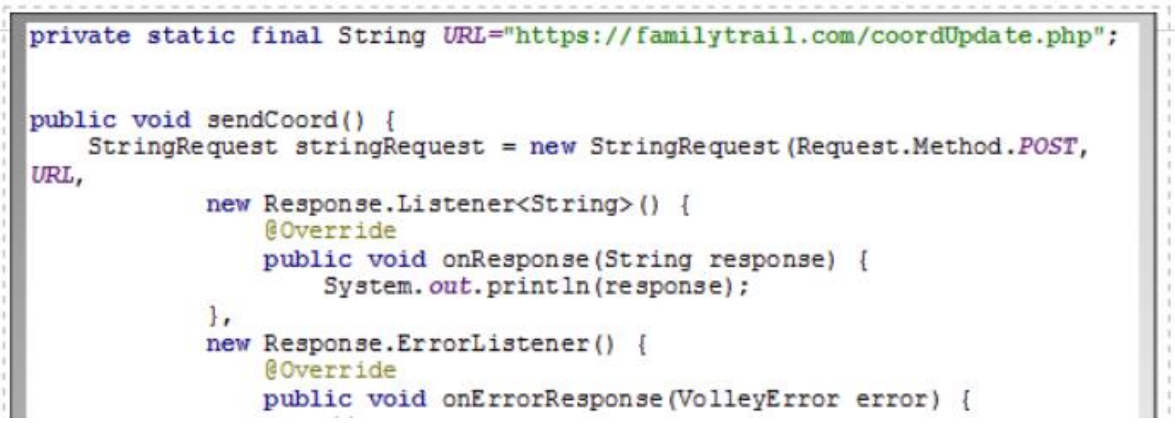
A private string variable is assigned, the URL that receives and handles the values sent from the application. This following sendCoord method uses Android Volley library to send coordinates to the queue. The variables are first received inside the response listener. The received values are put into a Hash Map array list. The array list is then processed into parameters which will be received and sent to the php webpage on the URL variable.

Background services & status monitor

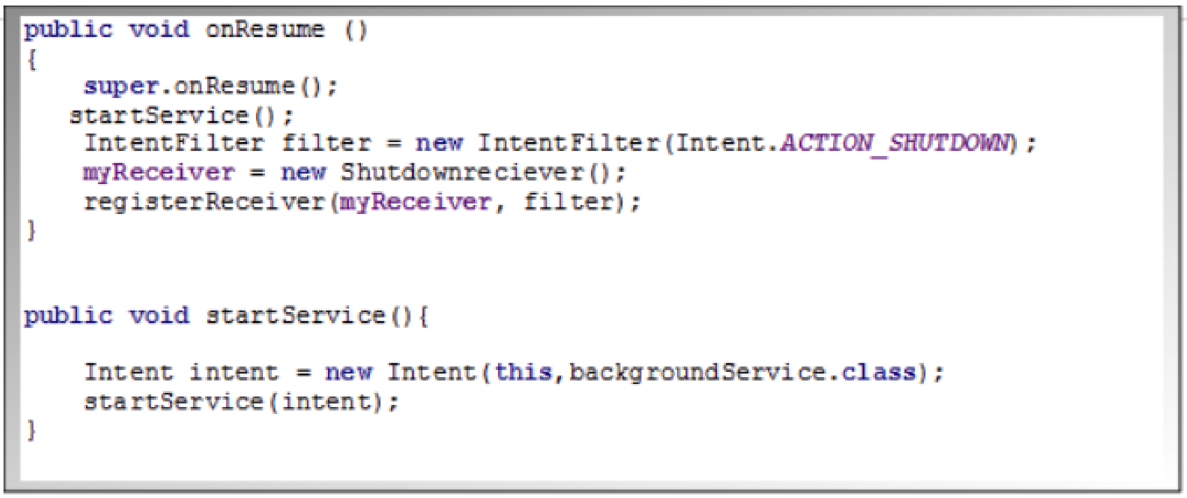
There are two other class components that are used in this application : backgroundService.class and ShutdownReciever.class.

While the app is running background service class is called through a method known as startService() as seen below inside the onResume method.   Shutdown receiver class is also initiated.

BackgroundService.class contains important code for sending an SMS message to the tracker's phone. When the App is closed, the method onTaskremoved is triggered. This method sends a text message to the tracker apps using the parameters such as the phoneNo and message text. Text

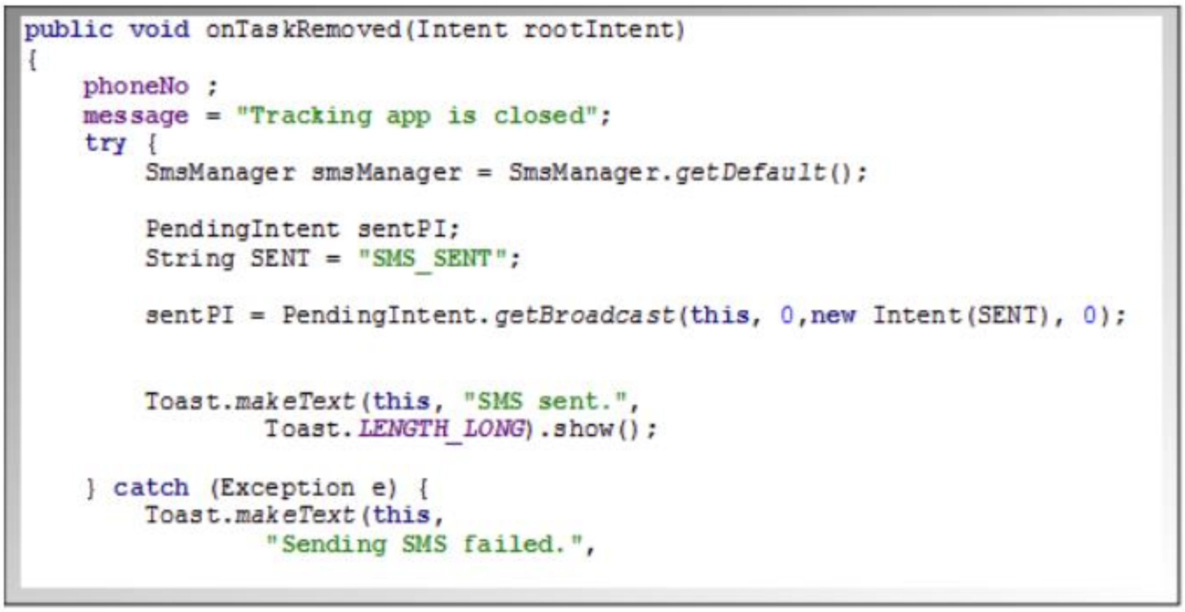


```
private static final String URL="https://familytrail.com/coordUpdate.php";


public void sendCoord() {
    StringRequest stringRequest = new StringRequest(Request.Method.POST,
URL,
            new Response.Listener<String>() {
                @Override
                public void onResponse(String response) {
                    System.out.println(response);
            },
            new Response.ErrorListener() {
                @Override
                public void onErrorResponse(VolleyError error) {
```



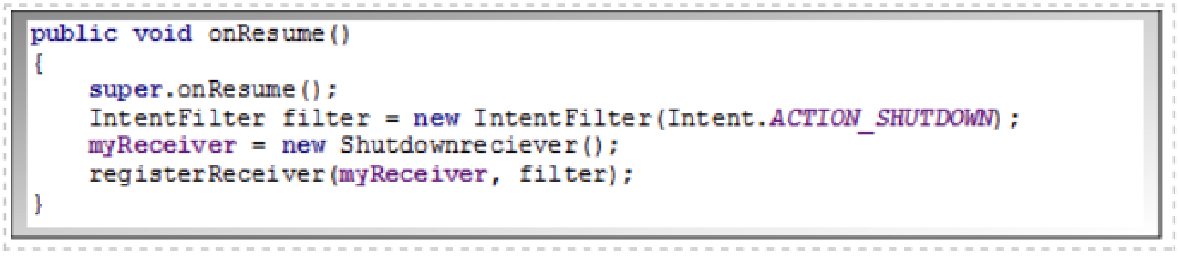
```
public void onResume ()
{
    super.onResume();
   startService();
    IntentFilter filter = new IntentFilter(Intent.ACTION_SHUTDOWN);
    myReceiver = new Shutdownreciever();
    registerReceiver(myReceiver, filter);
}


public void startService(){

    Intent intent = new Intent(this,backgroundService.class);
    startService(intent);
}
```

message will be sent to the tracker through the native SMS manager function (as seen below).



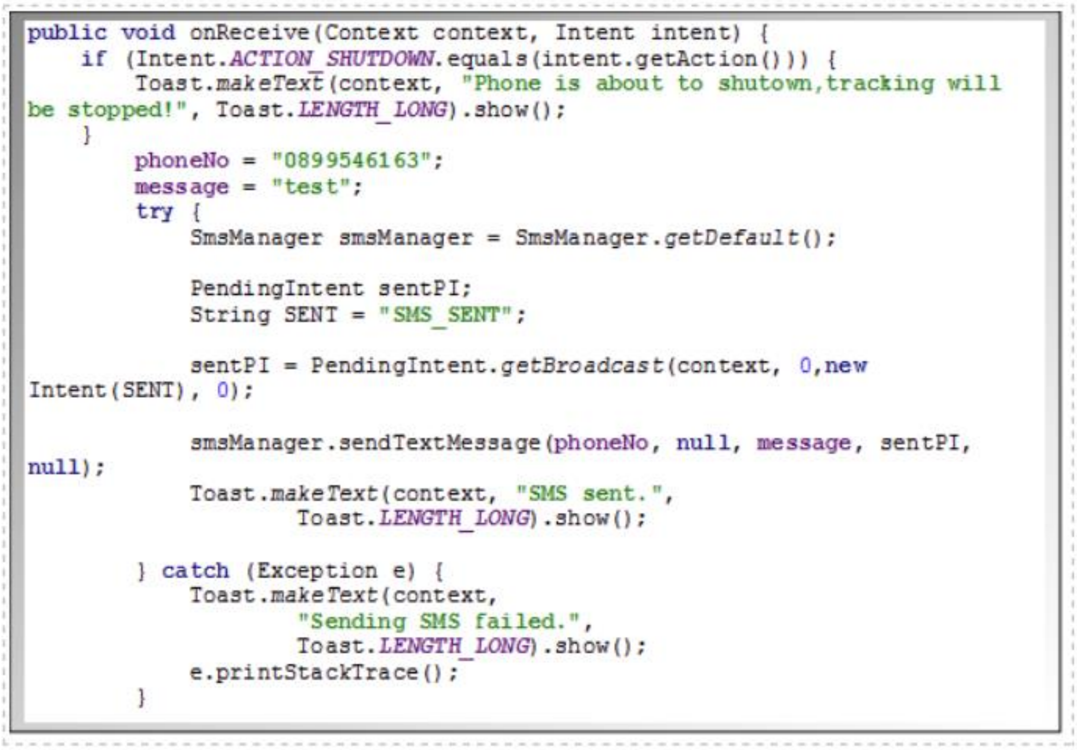
```
public void onTaskRemoved(Intent rootIntent)
{
    phoneNo ;
    message = "Tracking app is closed";
    try {
        SmsManager smsManager = SmsManager.getDefault();

        PendingIntent sentPI;
        String SENT = "SMS_SENT";

        sentPI = PendingIntent.getBroadcast(this, 0,new Intent(SENT), 0);


        Toast.makeText(this, "SMS sent.",
                Toast.LENGTH_LONG).show();

    } catch (Exception e) {
        Toast.makeText(this,
                "Sending SMS failed.",
```

Shutdown receiver class is started when the application registers the receiver in the main activity.



```
public void onResume()
{
    super.onResume();
    IntentFilter filter = new IntentFilter(Intent.ACTION_SHUTDOWN);
    myReceiver = new Shutdownreciever();
    registerReceiver(myReceiver, filter);
}
```

When the phone goes into shutdown, a text message will be sent to the tracker. The method onRecieve monitor phone status while the application is active. OnRecieve constantly checks for “SHUTDOWN” action. If the action from the app is “SHUTDOWN”, the function sends a text message to the tracker alerting with a message about shutdown. Text message will be sent to the tracker through the native SMS manager function (as seen below).

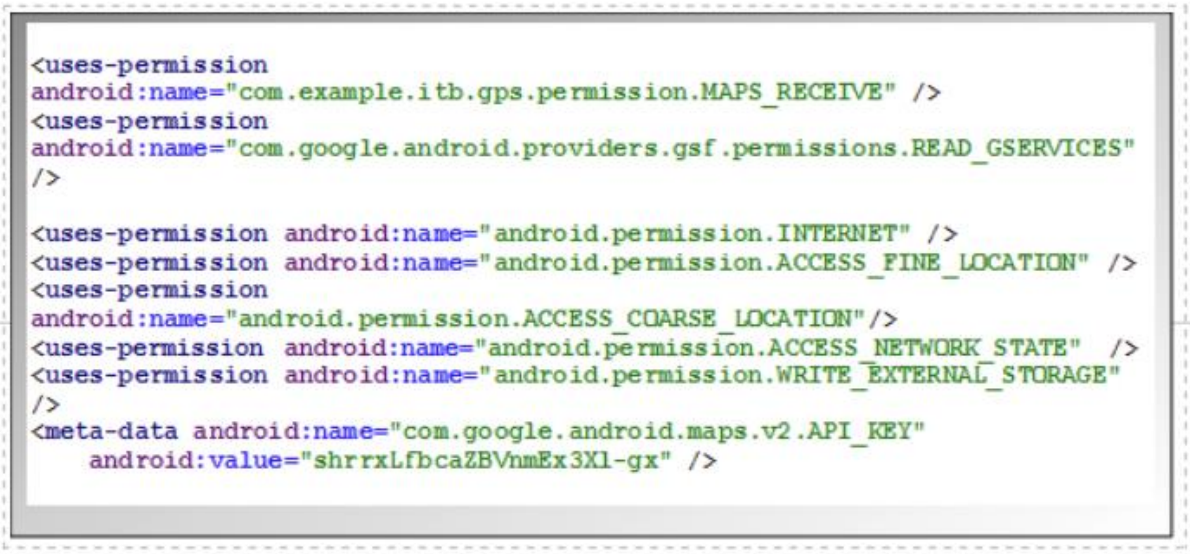


```
public void onReceive(Context context, Intent intent) {
    if (Intent.ACTION_SHUTDOWN.equals(intent.getAction())) {
        Toast.makeText(context, "Phone is about to shutown,tracking will
be stopped!", Toast.LENGTH_LONG).show();
    }
        phoneNo = "0899546163";
        message = "test";
        try {
            SmsManager smsManager = SmsManager.getDefault();

            PendingIntent sentPI;
            String SENT = "SMS_SENT";

            sentPI = PendingIntent.getBroadcast(context, 0,new
Intent(SENT), 0);

            smsManager.sendTextMessage(phoneNo, null, message, sentPI,
null);
            Toast.makeText(context, "SMS sent.",
                    Toast.LENGTH_LONG).show();

        } catch (Exception e) {
            Toast.makeText(context,
                    "Sending SMS failed.",
                    Toast.LENGTH_LONG).show();
            e.printStackTrace();
        }
```

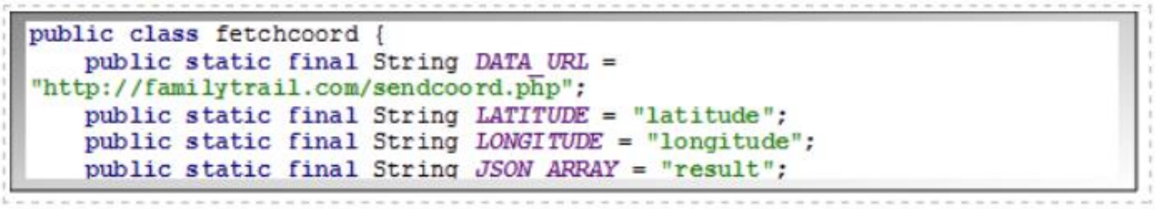
“FamilyTrail” Application (Tracker)

Similar to the GPSClient application, when the tracker is started for the first time the user is required to grant app the permission to use GPS locator service. The code in the Android manifest (an XML file that holds important components for running the app) file allows for this action to be performed. There are two permissions that are present in the file: app’s ability to access location info and Internet

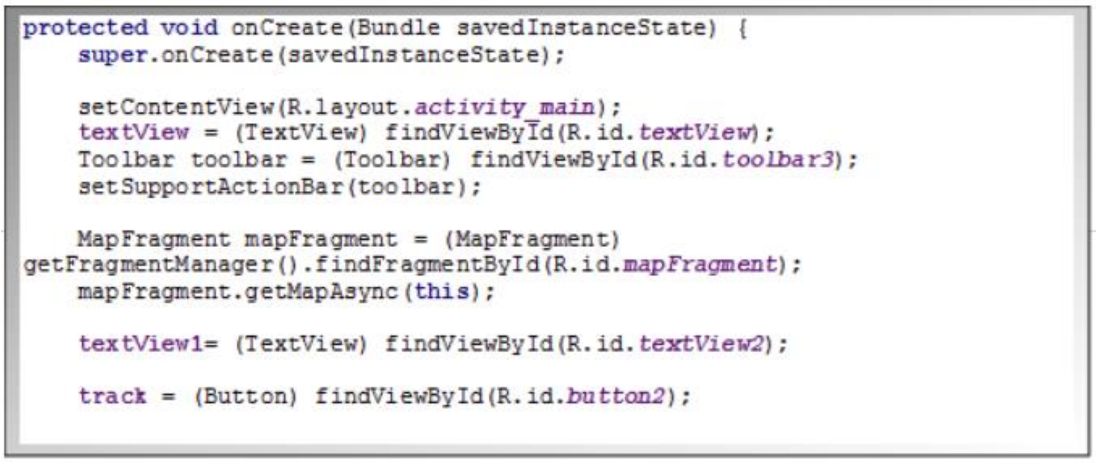
Google Map’s permission and API keys (unique key for Android map app) are also added to support Google Map component.


```
<uses-permission
android:name="com.example.itb.gps.permission.MAPS_RECEIVE" />
<uses-permission
android:name="com.google.android.providers.gsf.permissions.READ_GSERVICES"
/>

<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
<uses-permission
android:name="android.permission.ACCESS_COARSE_LOCATION"/>
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE"  />
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE"
/>
<meta-data android:name="com.google.android.maps.v2.API_KEY"
    android:value="shrrxLfbcaZBVnmEx3Xl-gx" />
```

When the app makes a request for location of the trackee, the fetchcoord.java class is utilized. This class assigns the variables that needs to retrieved from the URL in the form a JSON array.

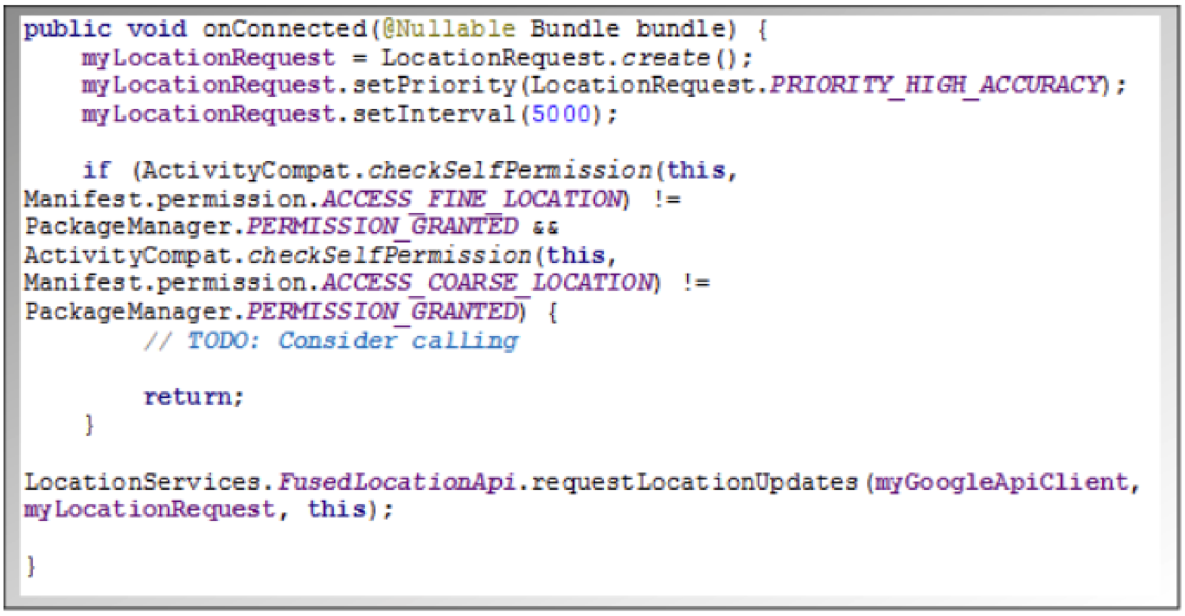

```
public class fetchcoord {
    public static final String DATA_URL =
"http://familytrail.com/sendcoord.php";
    public static final String LATITUDE = "latitude";
    public static final String LONGITUDE = "longitude";
    public static final String JSON_ARRAY = "result";
```

When the application is launched, onCreate method prepares the GUI components for display. As seen below the MAP fragment which holds the map component in the interface is assigned. Other components such as child location button and a textview boxes are also created.



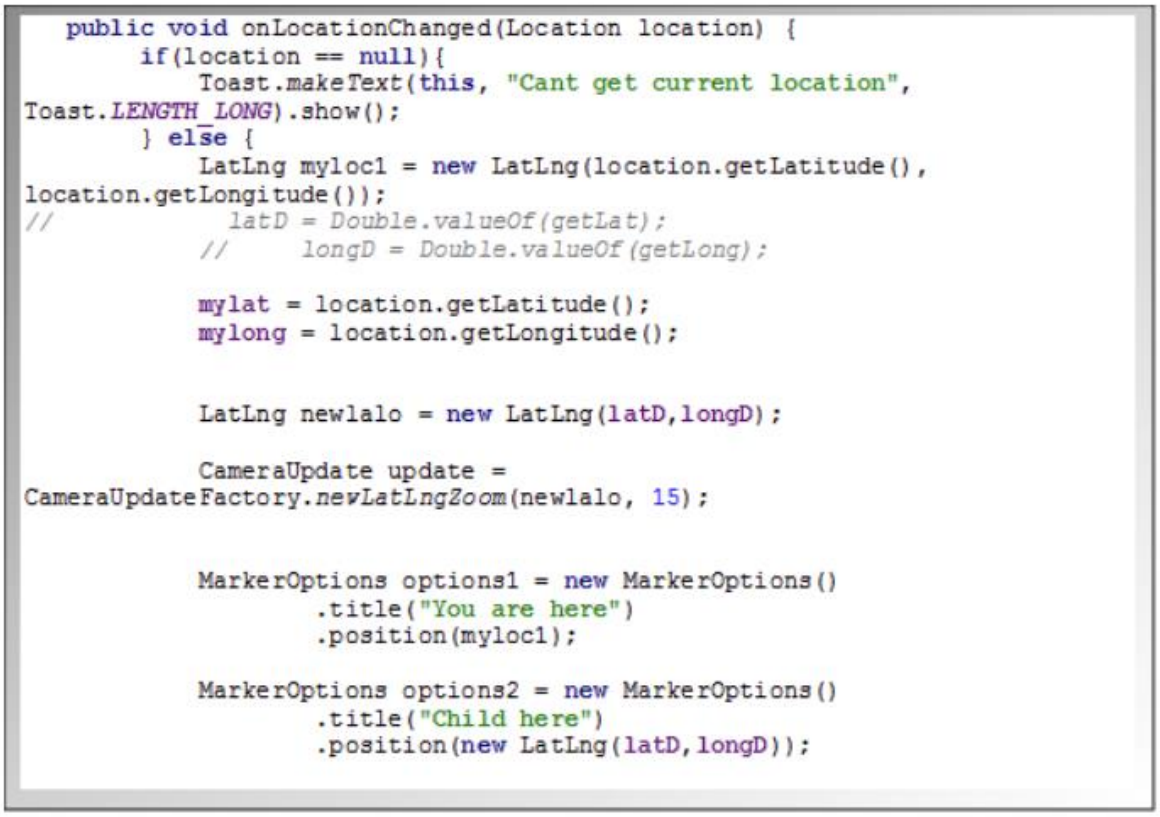
```
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);

    setContentView(R.layout.activity_main);
    textView = (TextView) findViewById(R.id.textView);
    Toolbar toolbar = (Toolbar) findViewById(R.id.toolbar3);
    setSupportActionBar(toolbar);

    MapFragment mapFragment = (MapFragment)
getFragmentManager().findFragmentById(R.id.mapFragment);
    mapFragment.getMapAsync(this);

    textView1= (TextView) findViewById(R.id.textView2);

    track = (Button) findViewById(R.id.button2);
```

When the app searches for the location, it retrieves current location from onConnected method. This method is collects the location information every five seconds (as seen in this instance) and sends it to OnLocationChanged method.



```
public void onConnected(@Nullable Bundle bundle) {
    myLocationRequest = LocationRequest.create();
    myLocationRequest.setPriority(LocationRequest.PRIORITY_HIGH_ACCURACY);
    myLocationRequest.setInterval(5000);

    if (ActivityCompat.checkSelfPermission(this,
Manifest.permission.ACCESS_FINE_LOCATION) !=
PackageManager.PERMISSION_GRANTED &&
ActivityCompat.checkSelfPermission(this,
Manifest.permission.ACCESS_COARSE_LOCATION) !=
PackageManager.PERMISSION_GRANTED) {
        // TODO: Consider calling

        return;
    }

LocationServices.FusedLocationApi.requestLocationUpdates(myGoogleApiClient,
myLocationRequest, this);

}
```

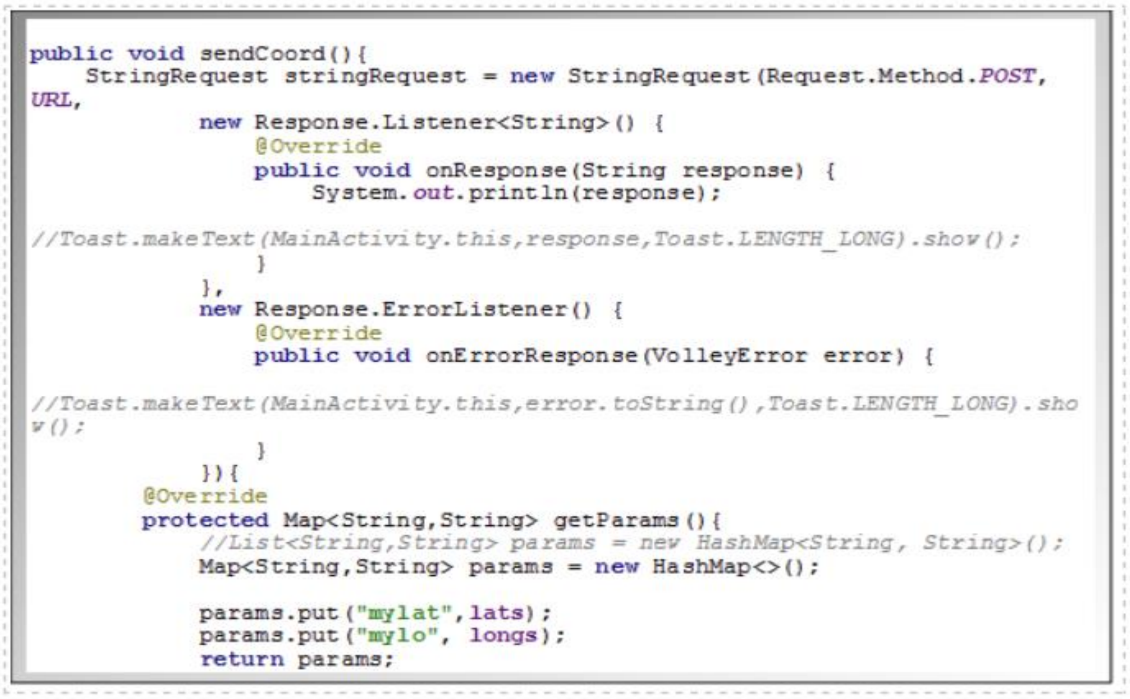
After gathering the location information, the onLocation method assigns the coordinates to a lat and long variables which will be sent to the database. Map camera zoom and focus operations are also defined using CamerUpdate function. The tracker and trackee marker options are also placed are also defined.



```
    public void onLocationChanged(Location location) {
          if(location == null){
               Toast.makeText(this, "Cant get current location",
Toast.LENGTH_LONG).show();
          } else {
               LatLng myloc1 = new LatLng(location.getLatitude(),
location.getLongitude());
//               latD = Double.valueOf(getLat);
               //     longD = Double.valueOf(getLong);

               mylat = location.getLatitude();
               mylong = location.getLongitude();


               LatLng newlalo = new LatLng(latD,longD);

               CameraUpdate update =
CameraUpdateFactory.newLatLngZoom(newlalo, 15);


               MarkerOptions options1 = new MarkerOptions()
                         .title("You are here")
                         .position(myloc1);

               MarkerOptions options2 = new MarkerOptions()
                         .title("Child here")
                         .position(new LatLng(latD,longD));
```

Database Functionalities

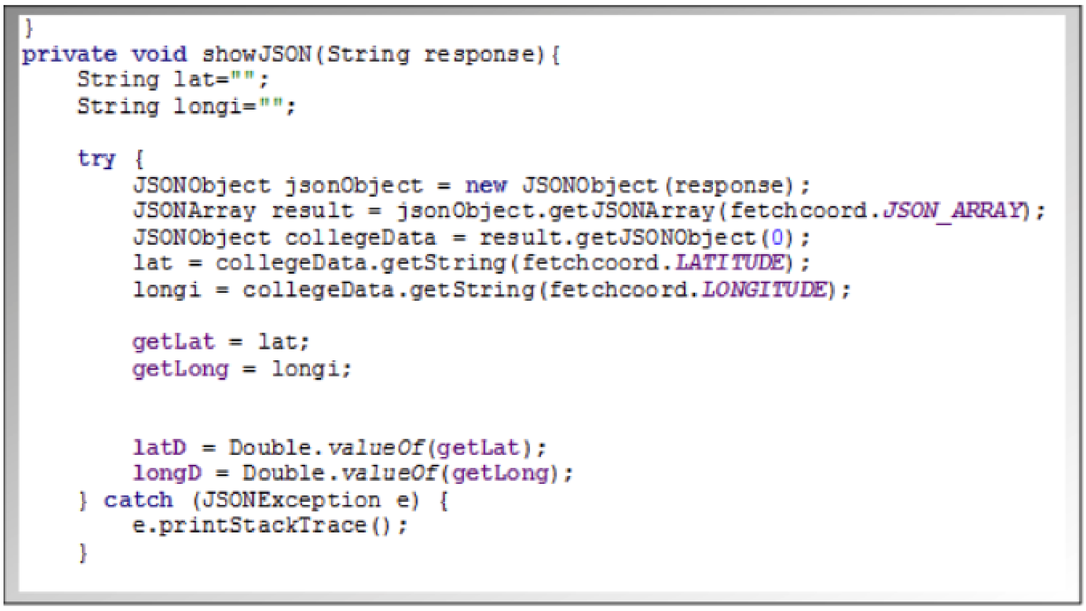
The tracker's coordinates are sent to the database through the sendCoord method. A private string variable is assigned, the URL that receives and handles the values sent from the application. This following sendCoord method uses Android Volley library (Android Developers, 2017) to send coordinates to the queue. The variables are first received inside the response listener. The received values are put into a Hash Map array list. The array list is then processed into parameters which will be received and sent to the php webpage on the URL variable.



```
public void sendCoord(){
    StringRequest stringRequest = new StringRequest(Request.Method.POST,
URL,
            new Response.Listener<String>() {
                @Override
                public void onResponse(String response) {
                    System.out.println(response);
//Toast.makeText(MainActivity.this,response,Toast.LENGTH_LONG).show();
                }
            },
            new Response.ErrorListener() {
                @Override
                public void onErrorResponse(VolleyError error) {
//Toast.makeText(MainActivity.this,error.toString(),Toast.LENGTH_LONG).sho
w();
                }
            }){
        @Override
        protected Map<String,String> getParams(){
            //List<String,String> params = new HashMap<String, String>();
            Map<String,String> params = new HashMap<>();

            params.put("mylat",lats);
            params.put("mylo", longs);
            return params;
```

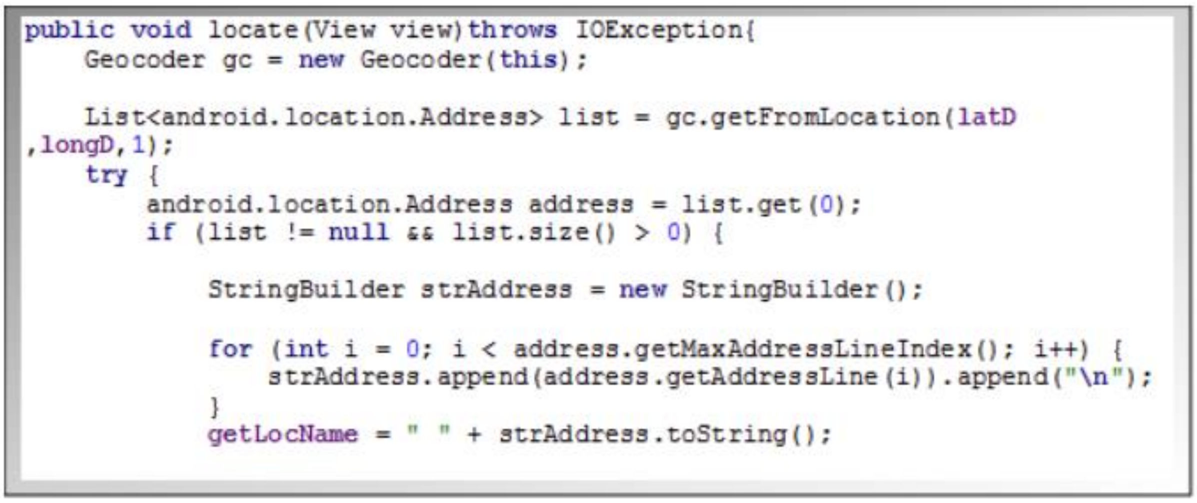
The trackee's location is retrieved using the combination of fetchcoord class and showJSON method.

A JSON object is initiated for the response (from the database). The response array is passed to an array of values that will be retrieved from database. The values in the array fetched include latitude and longitude values which are converted to double to plot the point on the map.



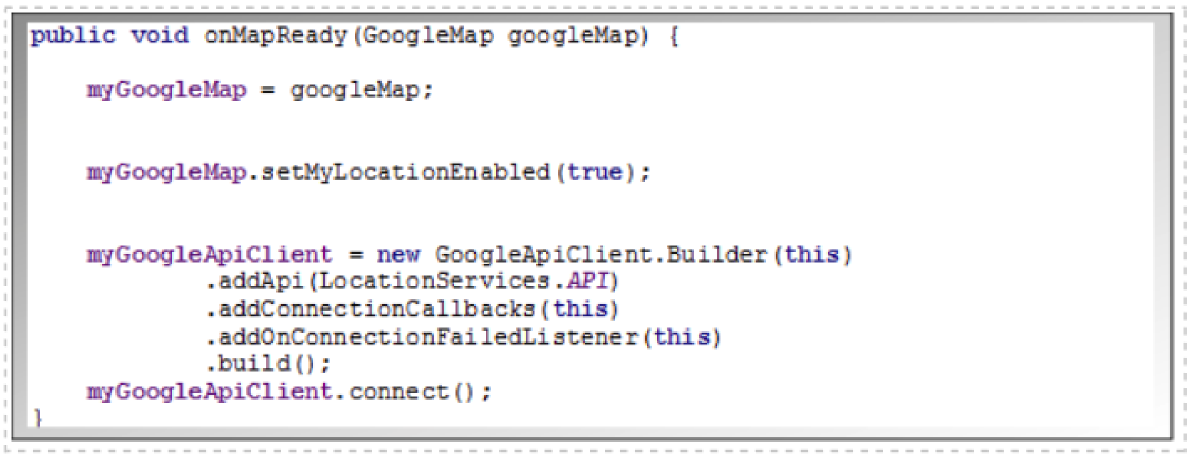
```
}
private void showJSON(String response){
    String lat="";
    String longi="";

    try {
        JSONObject jsonObject = new JSONObject(response);
        JSONArray result = jsonObject.getJSONArray(fetchcoord.JSON_ARRAY);
        JSONObject collegeData = result.getJSONObject(0);
        lat = collegeData.getString(fetchcoord.LATITUDE);
        longi = collegeData.getString(fetchcoord.LONGITUDE);

        getLat = lat;
        getLong = longi;


        latD = Double.valueOf(getLat);
        longD = Double.valueOf(getLong);
    } catch (JSONException e) {
        e.printStackTrace();
    }
```

For getting the actual name of the location of the trackee, the locate method is utilized. Geocoder API is used to translate the coordinates of the trackee to real geographic location. The coordinates are put into a list array. The list array is then fed into the list of the addresses in the for loop for translation.



```
public void locate(View view)throws IOException{
    Geocoder gc = new Geocoder(this);

    List<android.location.Address> list = gc.getFromLocation(latD
,longD,1);
    try {
        android.location.Address address = list.get(0);
        if (list != null && list.size() > 0) {

            StringBuilder strAddress = new StringBuilder();

            for (int i = 0; i < address.getMaxAddressLineIndex(); i++) {
                strAddress.append(address.getAddressLine(i)).append("\n");
            }
            getLocName = " " + strAddress.toString();
```

When the map options are set the onMapReady method is called. This method initiates the map object with the previously configured options such as the markers. The GooglemapAPI client function adds things such as locationservices that are needed for the map.



```
public void onMapReady(GoogleMap googleMap) {

    myGoogleMap = googleMap;


    myGoogleMap.setMyLocationEnabled(true);


    myGoogleApiClient = new GoogleApiClient.Builder(this)
            .addApi(LocationServices.API)
            .addConnectionCallbacks(this)
            .addOnConnectionFailedListener(this)
            .build();
    myGoogleApiClient.connect();
}
```

Website

A web interface was chosen as the primary interface for parents to access all features included in this mobile app. A simple single-page template was used: this included a standard grey colour scheme, a set of images, and all back-end functions were also implemented.

The main features of the app include the real-time map positioning, the ready availability of historical locations from a calendar interface, and the ability to set a number of options such as the safe-zone area, and turning on and off the tracking.

Free webhosting that included MySQL databases were obtained from DigiWeb. Two databases were needed, one to handle the uploading of GPS co-ordinates, the other to manage user accounts and keep track of user preferences.

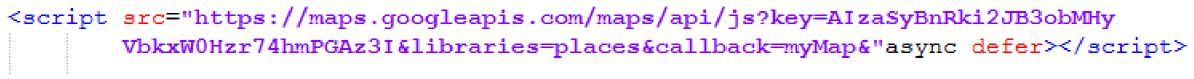
The same design was retained for when the user was logged in or logged out, but if not logged in, a PHP script identifies that the session variable for ‘user’ has not been set, and will redirect to the log-in screen. If the same variable has been set, it will be used to greet the user in the text on the page.

```php
session_start();
if (isset($_SESSION['user'])) {
  $user = $_SESSION['user'];
} else {
  header("Location: login.php");
}
```

PHP code on the index and home page was added to check that the site has been accessed using the secure HTTP protocol instead of port 80.

```php
if(empty($_SERVER['HTTPS']) || $_SERVER['HTTPS'] == "off"){
    $redirect = 'https://' . $_SERVER['HTTP_HOST'] . $_SERVER['REQUEST_URI'];
    header('HTTP/1.1 301 Moved Permanently');
    header('Location: ' . $redirect);
    exit();
}
```

Googles Maps API (application programming interface) was chosen to display the GPS locations uploaded to the database from the child’s phone. This was because Google Maps is the most comprehensive mapping application available without charge on the internet. It has grown very appreciably over the last decade, using precise mapping information supplied from many sources, from local councils, to forest services, to Geological Surveys. It has also acquired high-quality satellite images in recent years from NASA. Also, individual towns consider it in their best interest to keep the information both precise and up to date as it is such a popular resource, so over the last few years it has developed into the most powerful mapping application available on the internet. APIs are available for developers to integrate the maps into their apps and websites. For this application a java API was used which can be accessed by a simple URL using the SCRIPT tag in HTML.



```
<script src="https://maps.googleapis.com/maps/api/js?key=AIzaSyBnRki2JB3obMHy
        VbkxW0Hzr74hmPGAz3I&libraries=places&callback=myMap&"async defer></script>
```

In designing this app it was necessary to sign up to Google Developers Website to get a unique API key that is linked to our Google account. This unique ID will allow Google to keep track of how many and what type of requests are being made by our application. If the app were to be launched commercially on the Google Play market, it would be necessary to sign up for a Premium Plan to legally use the API. The prices for this are quite affordable for most uses of the API. The standard plan is free for most of the Maps functions up to a certain amount of requests a day. For example, an Android app can have unlimited requests using the Maps API, although if the developer wants to use the Maps for assets or people then they must get in contact with Google to discuss a price for their Premium Plan. To use the Directions API, as well as most web services, Google allow up to 2,500 requests for free per API key a day, and an extra $0.50 per 1,000 additional requests up to 100,000 per day. The Premium Plans pricing is based on the number of requests required per day. For the Premium Plan with an asset-tracking license, an annual contract is arranged providing the developer access to 24-hour customer support and a guaranteed ad-free experience, as well as enhanced API features such as unlimited requests per day and access to speed limit data.

For the development of this app, use was made of the main Maps API as well as the Directions API. Using the MySQL database as a filing system, the coordinates can be uploaded from the phone, and downloaded and displayed on the Maps API in both the website and android application.

To implement the Safe-Zone feature, the user would select a region on the map, and be notified if the child leaves this region. To implement this, code was available on Googles Developers Site for "User-Editable Shapes". It was decided to restrict zone selection to a rectangle shape, and the code was modified to save the coordinates to the database in the current user's preferences, as well as displaying the new coordinates. The database was updated by updating four hidden text fields using JavaScript's getElementById function, and passing a global variable made to store the location from Google's function.

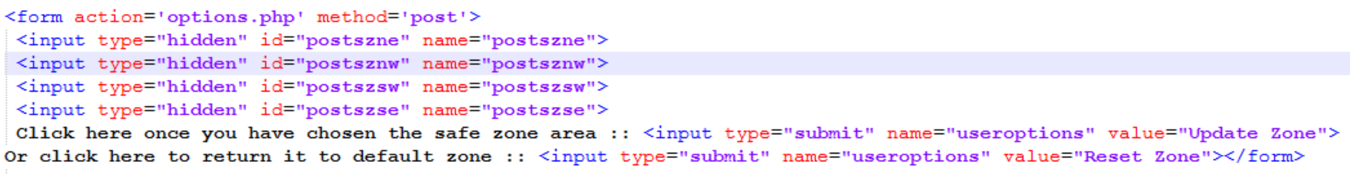
The zoom level on the map was set to hover over Dublin, and before the map loads, the current Safe Zone is pulled from the database. So, when a new area is selected, when the page is reloaded, that area will still show on the map, instead of a static example zone.

```
// Set variables to prepare to send to PHP and update DB
szne = ne.lat();
sznw = ne.lng();
szsw = sw.lat();
szse = sw.lng();
//update txt fields
updatetext();
```

```
function updatetext() {
    document.getElementById("postszne").value = szne;
    document.getElementById("postsznw").value = sznw;
    document.getElementById("postszsw").value = szsw;
    document.getElementById("postszse").value = szse;
}
```

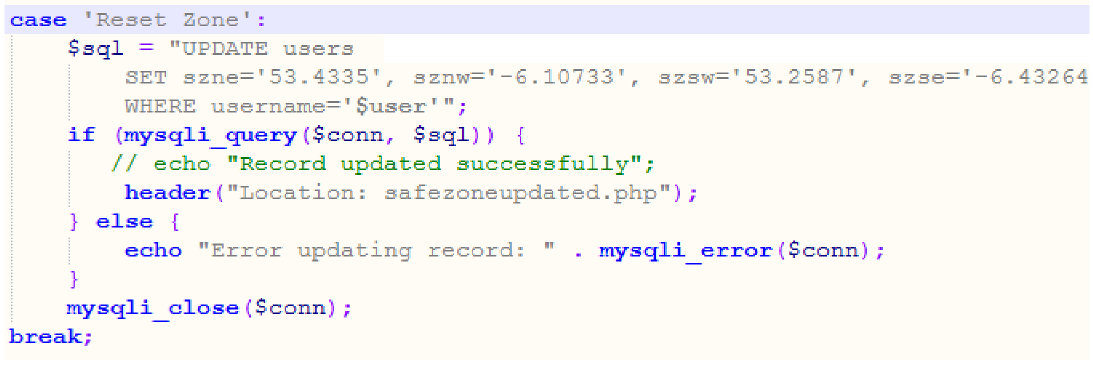
```
case 'Update':
    $interv = $_POST['interval'];
    $interv=mysqli_real_escape_string($conn,$interv);
    $sql = "UPDATE users SET intervalmins='$interv' WHERE username='$user'";
    if (mysqli_query($conn, $sql)) {
      // echo "Record updated successfully";
        header("Location: home.php");
    } else {
        echo "Error updating record: " . mysqli_error($conn);
    }
    mysqli_close($conn);
break;
```

```
case 'Turn On':
    $sql = "UPDATE users SET tracking=1 WHERE username='$user'";
    if (mysqli_query($conn, $sql)) {
        header("Location: home.php");
    } else {
        echo "Error updating record: " . mysqli_error($conn);
    }
    mysqli_close($conn);
break;

case 'Turn Off':
    $sql = "UPDATE users SET tracking=0 WHERE username='$user'";
    if (mysqli_query($conn, $sql)) {
        header("Location: home.php");
    } else {
        echo "Error updating record: " . mysqli_error($conn);
    }
    mysqli_close($conn);
break;
```



```
<form action='options.php' method='post'>
 <input type="hidden" id="postszne" name="postszne">
 <input type="hidden" id="postsznw" name="postsznw">
 <input type="hidden" id="postszsw" name="postszsw">
 <input type="hidden" id="postszse" name="postszse">
 Click here once you have chosen the safe zone area :: <input type="submit" name="useroptions" value="Update Zone">
Or click here to return it to default zone :: <input type="submit" name="useroptions" value="Reset Zone"></form>
```

The text fields were sent on to a specially written PHP script which uses a case statement to distinguish which option column to update in the database, in this case using the identifier from the buttons name ‘Update Zone’. The strings were escaped to remove any characters that could cause malice, in case any actor tried to pass unwanted data by editing the post parameters with a proxy.



```
case 'Reset Zone':
    $sql = "UPDATE users
        SET szne='53.4335', sznw='-6.10733', szsw='53.2587', szse='-6.43264
        WHERE username='$user'";
    if (mysqli_query($conn, $sql)) {
       // echo "Record updated successfully";
        header("Location: safezoneupdated.php");
    } else {
        echo "Error updating record: " . mysqli_error($conn);
    }
    mysqli_close($conn);
break;
```

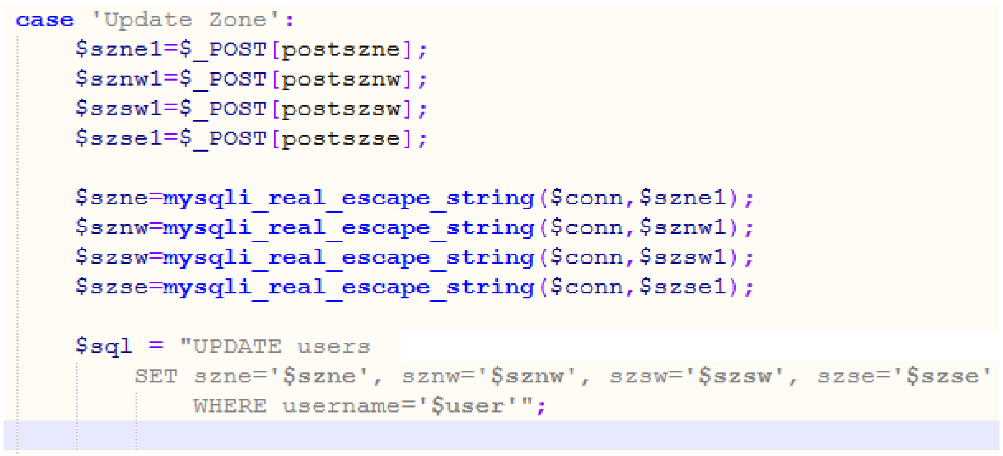
The PHP options script also handles the other options on the home page, such as the following:

Updating time interval between client GPS updates:

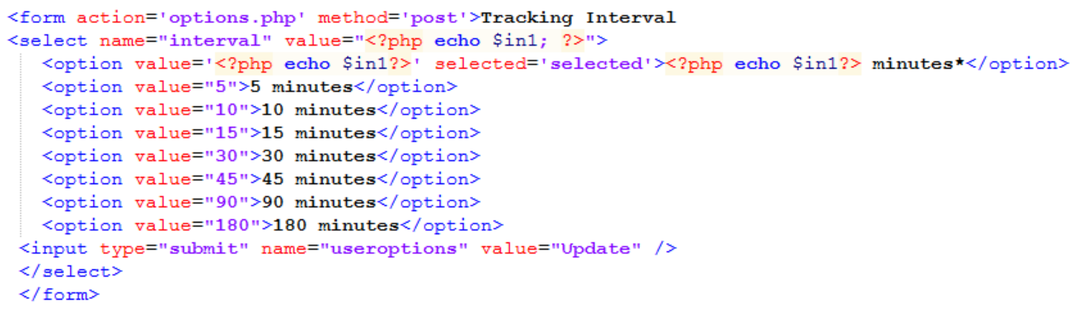
Back-end code to toggle between tracking on and off, by press of single button on homepage:

A reset button was also added, so if the user lost track of the Safe Zone, or in order that a new account can get the coordinates reset to a static value, covering Dublin was chosen.

This Update Zone code will take in the new coordinates set on the Safe Zone map by the user, and sanitise the input before updating the database.

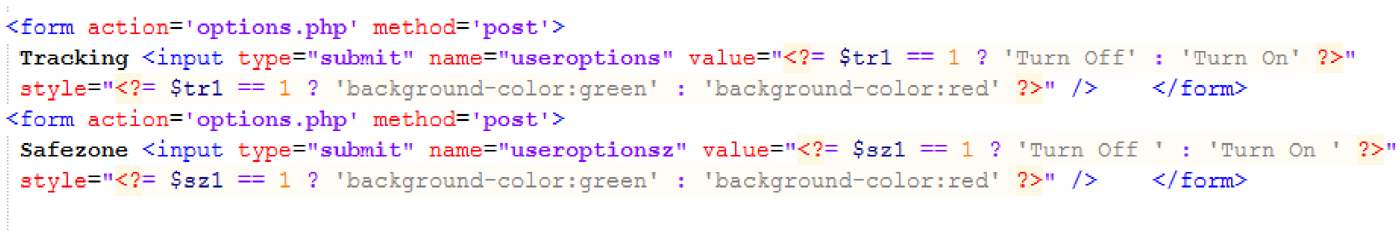


```
case 'Update Zone':
    $szne1=$_POST[postszne];
    $sznw1=$_POST[postsznw];
    $szsw1=$_POST[postszsw];
    $szse1=$_POST[postszse];

    $szne=mysqli_real_escape_string($conn,$szne1);
    $sznw=mysqli_real_escape_string($conn,$sznw1);
    $szsw=mysqli_real_escape_string($conn,$szsw1);
    $szse=mysqli_real_escape_string($conn,$szse1);

    $sql = "UPDATE users
        SET szne='$szne', sznw='$sznw', szsw='$szsw', szse='$szse'
            WHERE username='$user'";
```



```
<form action='options.php' method='post'>Tracking Interval
<select name="interval" value="<?php echo $in1; ?>">
   <option value='<?php echo $in1?>' selected='selected'><?php echo $in1?> minutes*</option>
   <option value="5">5 minutes</option>
   <option value="10">10 minutes</option>
   <option value="15">15 minutes</option>
   <option value="30">30 minutes</option>
   <option value="45">45 minutes</option>
   <option value="90">90 minutes</option>
   <option value="180">180 minutes</option>
 <input type="submit" name="useroptions" value="Update" />
 </select>
 </form>
```

On the home page, the options are ready to be set. The Tracking and Safe Zone features are ready to be activated. When they are deactivated the button is coloured red and says ‘Turn On’, while when they are green, they are activated and the button says ‘Turn Off’.



```
<form action='options.php' method='post'>
 Tracking <input type="submit" name="useroptions" value="<?= $tr1 == 1 ? 'Turn Off' : 'Turn On' ?>"
 style="<?= $tr1 == 1 ? 'background-color:green' : 'background-color:red' ?>" />    </form>
<form action='options.php' method='post'>
 Safezone <input type="submit" name="useroptionsz" value="<?= $sz1 == 1 ? 'Turn Off ' : 'Turn On ' ?>"
 style="<?= $sz1 == 1 ? 'background-color:green' : 'background-color:red' ?>" />    </form>
```

The interval setting is accessed through a drop-down menu, providing seven options between five and 180 minutes. The value that is saved in the database is shown as the first option, and will update if the user decides to choose a new

6 System Implementation

This Project is made up of a configuration that can be described as a Three-module implementation. Two modules of Android applications and one module of Website based application. The Main Tracking Application ("FamilyTrail") and the Website application serve as options for tracking the trackee with the help of the client application. FamilyTrail and the Website offer similar functionalities for users which makes them a backup option for one another.

The options for tracking (app and web) are the core for the motive of the project. Flexibility and Ease of Access among the options were top priorities during the implementation phase of the project

6.1 Android Applications Implementation

The First part of the Project is implemented as two parts of Android-based applications. The "GPSClient" application is implemented as the carrier module for the Users. This Trackee/Child carries the client on their phones which will be updating coordinates periodically. This part of the tracking service is lightweight and offering precise information on whereabouts of the carrier on a set intervals of time.

Efficiency in tracking stems from strong API options for High tracking with relatively moderate power usage. As the App can be run on the background, the implementation is silent and hidden away preventing obstruction for the phone user.

A Background service is also implemented in this module of the service. This service is used mainly for monitoring the Trackee/Child's phone/app and alerting the Parents/Tracker via SMS message in case of emergency.

The Second half of the Android-based implementation is the "FamilyTrail" application. This application is implemented as the core module of the project for tracking service. The Parents/Tracker allows for highly precise tracking of the Child/Trackee's phone through map's object.

The Map window can be tweaked to show different types of map such as terrain etc. Markers are placed based on the location of the users. The markers are interactive to revealing information on whereabouts. Application is also linked with Google Maps app to provide extensive directions support. Exact location of the child being tracked can also retrieved with the implementation of a geocoder service.

6.2 Website

To find out information about the app, users may visit the website, which describes the functionality of the app, and has a free registration form. They just need to provide their username, password, email address and phone number to begin. Upon logging in they have access to the full site, but before they can start tracking they must download the Android APK from the Google Play market, it is called ‘GPSclient’. This application is designed to run ideally on a smartwatch, but is compatible with any Android device that has internet and GPS. Upon installing the APK, the user can log in with the same details they registered on the website with, and this will link the child’s device to the parents account!

On the website upon logging in, the home page provides a number of options to be configured. First the parents will decide the interval between location updates, there are choices ranging from every five minutes to 180 minutes. Next they can click into the Safe Zone feature which has been outlined previously. This brings them to a new page which has a Google Maps interface, that had a square which is pre-set over Dublin city. The user is able to move this box around, making bigger or smaller to suit their requirements. Once they have decided the boundaries which the child should not leave while tracking is enables, they click the ‘Update Zone’ button. This will save the location to the database, where a back-end script is ready to monitor that any uploaded GPS coordinates do not reside outside the chosen area. The page is reloaded and the user may choose how they would like to be alerted if the child does leave the area, with the most popular option being an SMS message or email. They may also choose to set an alarm on the child’s device so they realise they are not supposed to be venturing so far away!

Returning to the previous homepage the parents can now choose to activate the tracking, which means the child’s device will upload GPS coordinates as often as the chosen interval dictates. They may also active the Safe Zone setting, which will activate a server-side monitoring script which will be enabled. If that is the case, a message will be sent to the parents to inform them that the child’s device is out of contact, and could be caused by battery running out, the child intentionally closing the application or turning off the device, the device losing coverage, or simply running out of credit. Once tracking is enabled the user can navigate to the Live Map page, which will show the location of the child in the most recent upload of coordinates. Here the parents can get directions to the child from their current location, which is uploaded through an Android application which the parents can also use to access most features available on the web interface.

After a few days tracking the user may make use of the history page, which displays a calendar, and any dates where tracking has been enabled provides a link to a Google Map, which displays all coordinates uploaded on that day so they can keep records of where the child was on certain dates. The next page is a Help page, which includes a frequently-asked-questions section, as well as a live chat, to get in contact with a member of the development team, who will provide any assistance necessary to ensure the customer gets the best experience possible from the product. The only link left on the website is to logout when the parents are leaving their computer, or more importantly if they are leaving a public computer, so the children’s location will not be accessed by unwanted actors.

6.3 Technologies Utilized

The main technologies implemented in the development of the project are:

- JAVA
- PHP
- JAVASCRIPT
- HTML
- CSS
- JQUERY
- JSON
- MYSQL

The main applications which were used during development are:

- Android studio
- PHPSTORM
- Notepad++
- MySQL server
- OneDrive
- GitHub
- PHPmyadmin

7 Testing and Evaluation

7.1 Functional Testing

GPS Client" Application (Trackee)

Requirements for the application:

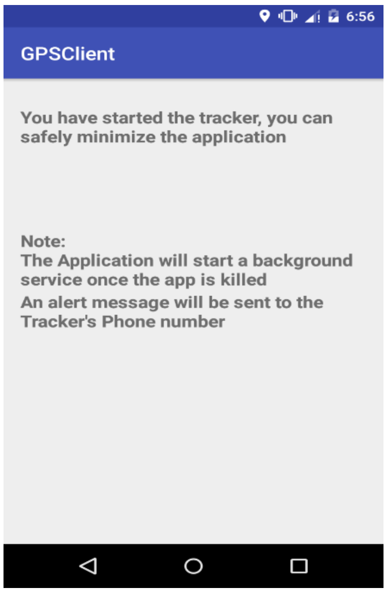
- Android build API versions 15+(Ice cream Sandwich & above)
- GPS service permissions
- Internet connection
- Google maps Application(pre-installed app on most phones)

The basic tasks that are tested in this section are:

- Application starting without errors.
- Main interface being displayed.
- Any warning or errors tested
- Coordinates sent to the Database using the GPS service and the app working in the background.
- Validity of the coordinates sent to the Database from the phone.
- Background service starting as soon as the application closes.
- SMS messages sent to Tracker phone before a shutdown or a quit through the background service.

Application starting without errors & Main Interface being displayed

Application is working without errors when the application is started and the Main Screen is displayed without any issues



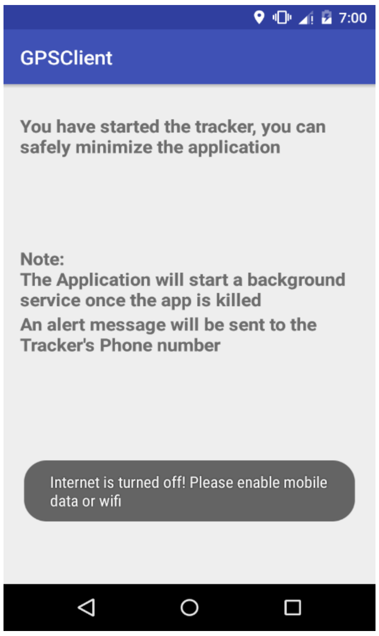
You have started the tracker, you can safely minimize the application

Note:
The Application will start a background service once the app is killed
An alert message will be sent to the Tracker's Phone number

Figure 20 – Starting application

Any warning or errors tested

When the internet is turned off, the error message alerting the user is displayed as intended.



GPSClient

You have started the tracker, you can safely minimize the application

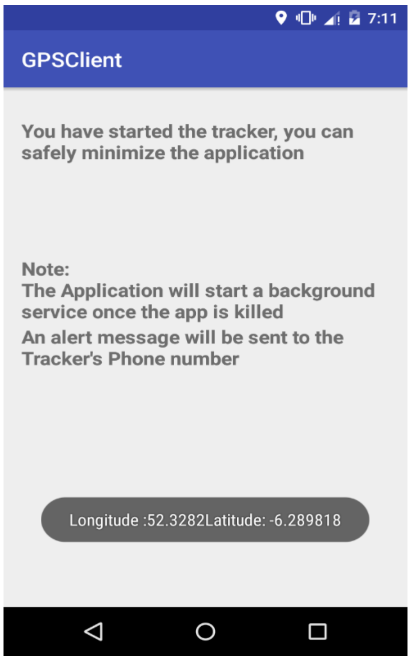
Note:
The Application will start a background service once the app is killed
An alert message will be sent to the Tracker's Phone number

Internet is turned off! Please enable mobile data or wifi

Figure 21 – Warning Message

Coordinates sent to the Database using the GPS service and the app working in the background.

As show in the popup Toast message below the latitude and longitude are updated as the app is running, in the background.



GPSClient

You have started the tracker, you can safely minimize the application

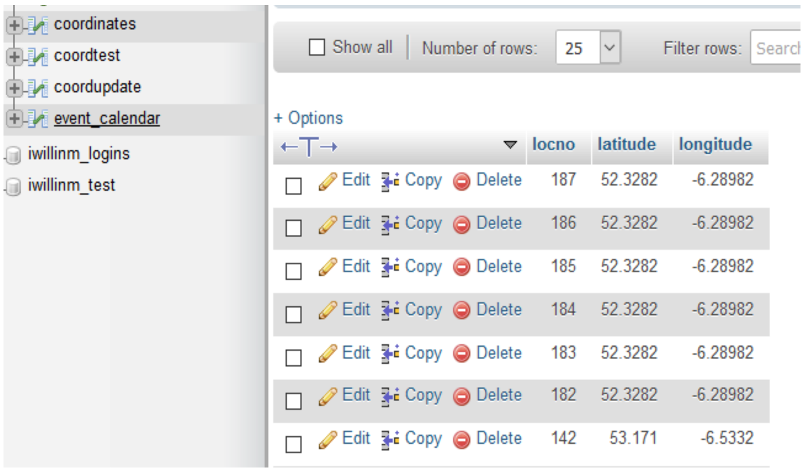
Note:
The Application will start a background service once the app is killed
An alert message will be sent to the Tracker's Phone number

Longitude :52.3282Latitude: -6.289818

Figure 22 – Warning Message

Validity of the coordinates sent to the Database from the phone.

The values updated as seen in Figure 23 are added correctly to the table coordinates in the database as intended.



coordinates
coordtest
coordupdate
event_calendar
iwillinm_logins
iwillinm_test

Show all | Number of rows: 25 | Filter rows: Searc

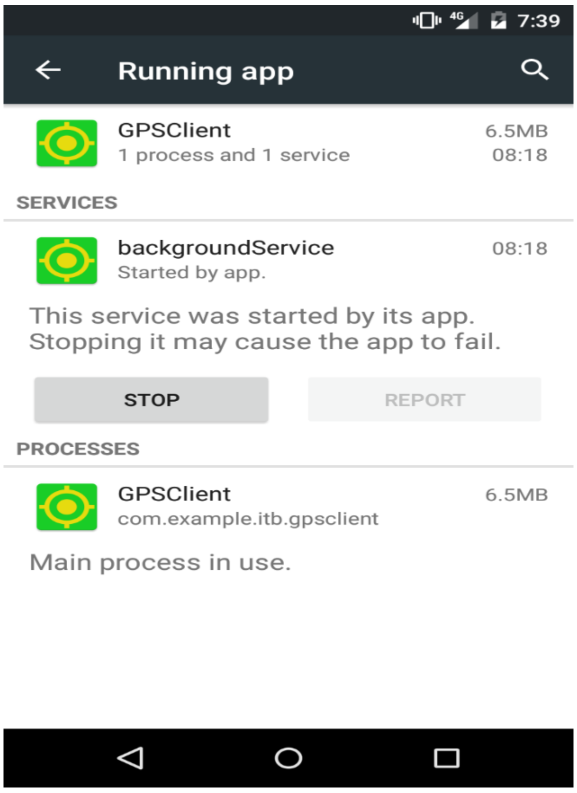
+ Options

	locno	latitude	longitude
Edit Copy Delete	187	52.3282	-6.28982
Edit Copy Delete	186	52.3282	-6.28982
Edit Copy Delete	185	52.3282	-6.28982
Edit Copy Delete	184	52.3282	-6.28982
Edit Copy Delete	183	52.3282	-6.28982
Edit Copy Delete	182	52.3282	-6.28982
Edit Copy Delete	142	53.171	-6.5332

Figure 23 – Warning Message

Background service starting as soon as the application closes.

After the app closing, the background service is started. From the running apps section in the settings, the "backgroundService" (a child process of GPSClient) is seen to be running as show below.



7:39

Running app

GPSClient 6.5MB
1 process and 1 service 08:18

SERVICES

backgroundService 08:18
Started by app.

This service was started by its app.
Stopping it may cause the app to fail.

STOP REPORT

PROCESSES

GPSClient 6.5MB
com.example.itb.gpsclient

Main process in use.

Figure 24 – Background service started

SMS messages sent to Tracker phone before a shutdown or a quit through the background service.

As soon as the application closes an “SMS Sent” and “Tracking removed” is shown in a toast message. In the messages sent folder, the sent message “ Tracking is Closed!!” is seen.(Figure 26)

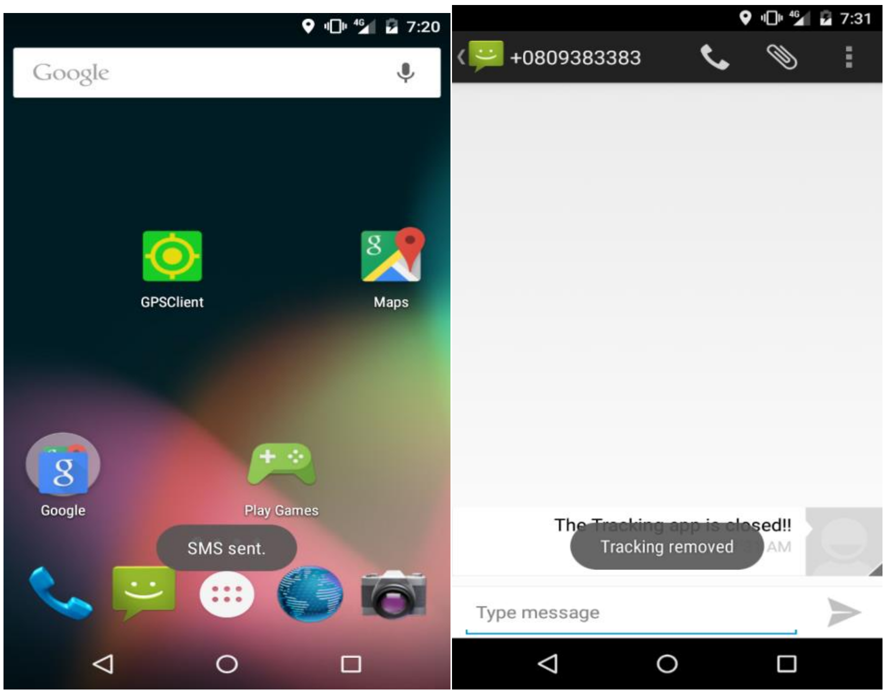


Figure 25 – Close alert Messages

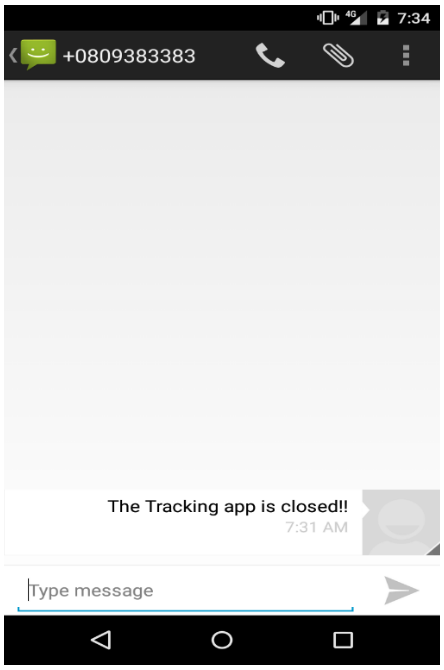


Figure 26– SMS Message sent after application closing

FamilyTrail" Application (Tracker)

Requirements for the application:

- Android build API versions 15+(Ice cream Sandwich & above)
- GPS service permissions
- Internet connection
- Google maps Application(pre-installed app on most phones)

The basic functions that are tested in this section are:

- Application starting without errors.
- Main interface displaying with Map Fragments and other GUI Components.
- Any warning or errors tested
- Coordinates sent to the database using the GPS service.
- Validity of the coordinates sent to Database and retrieved from the database.
- Markers placed on the correct set of locations for both the tracker and trackee.
- Link to the Google maps for Directions between users.
- Trackee's location retrieval based on the current location.

Application starting without errors & Main interface displaying with Map Fragments and other GUI Components.

When the Application is launched, it can be seen to start without any errors. All graphical user interface components such as the Maps are displayed as intended. (Figure 27)

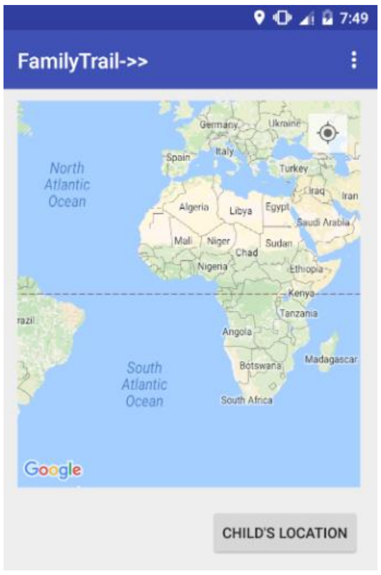
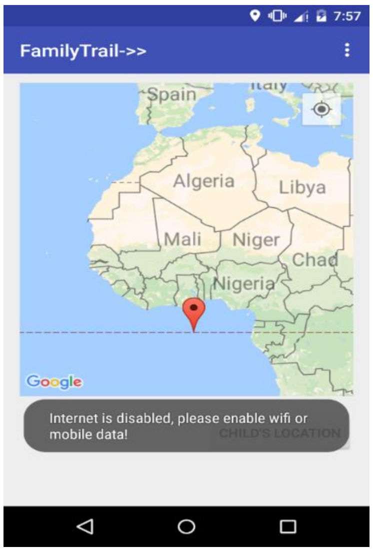


Figure 27 – App starting and displaying U

Any warning or errors tested

Similar to the GPSClient Application, when this program starts it searches for internet connection. If the internet connection is not found a small toast message is popped to alert the user to turn on internet.

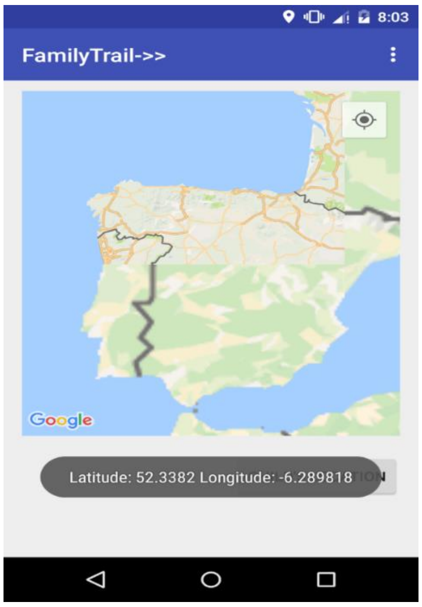


Internet is disabled, please enable wifi or mobile data!

Figure 28 – Warning message about connection

Coordinates sent to the database using the GPS service.

From the Figure 29, it can be seen that Latitude and Longitude sent to the Database are shown as a toast message.

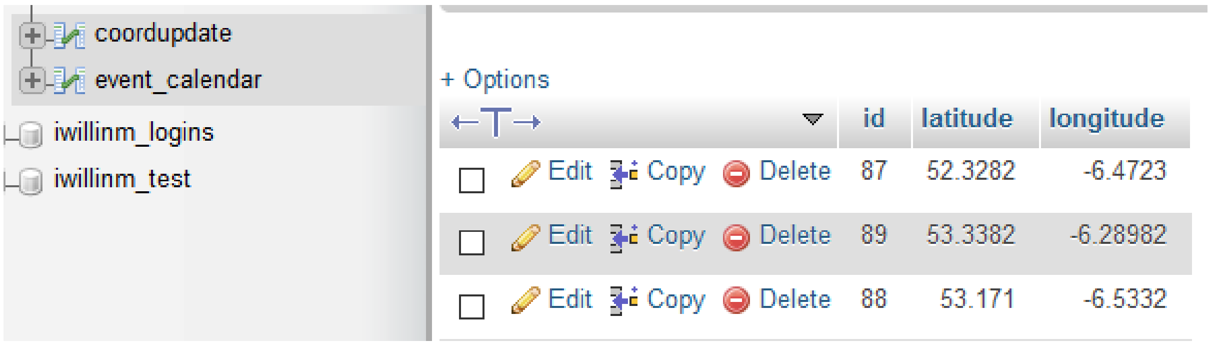


Latitude: 52.3382 Longitude: -6.289818

Figure 29 – Latitude and Longitude popup

Validity of the coordinates sent to Database

The GPS coordinate values sent from the application are shown to be matching with the latest id row added to the coordupdate table.

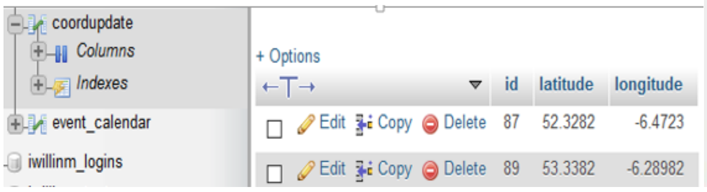


	id	latitude	longitude
Edit Copy Delete	87	52.3282	-6.4723
Edit Copy Delete	89	53.3382	-6.28982
Edit Copy Delete	88	53.171	-6.5332

Figure 30 – Values from coordupdate table

Markers placed on the correct set of location for the tracker.

The coordinates for both the tracker and trackee are fetched from DB and plotted onto the map. In the test below it can be seen that database values match the marker locations on the application.(Figure 31,32)



	id	latitude	longitude
Edit Copy Delete	87	52.3282	-6.4723
Edit Copy Delete	89	53.3382	-6.28982

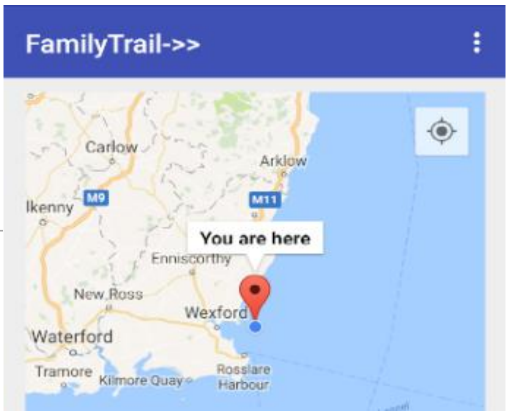


Figure 31 – Values from coordupdate table Figure 32 – marker shown in the app

Finding the location of the exact coordinates show on the app, reveals that markers working as intended.

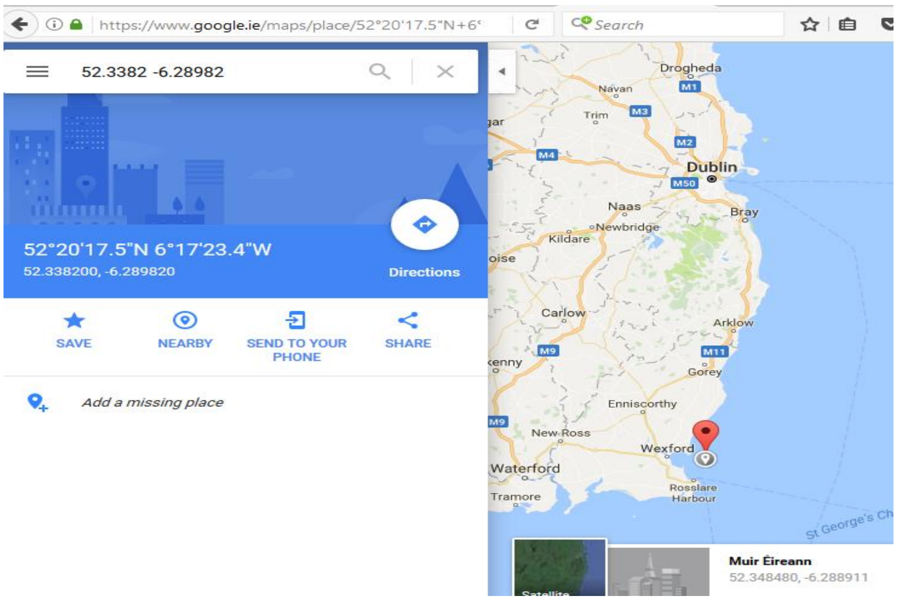
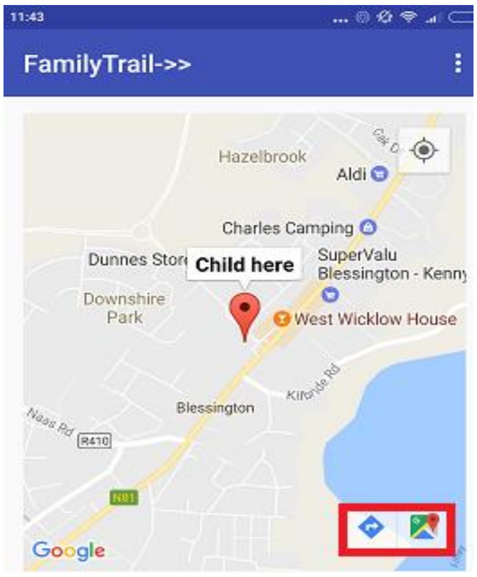


Figure 33 – Location shown on google maps

Link to the Google Maps for Directions between users.

When the user taps one of the buttons on the red rectangle shown on figure 34, they are redirected to the official Google Maps application. The locations of tracker and the trackee are provided by the FamilyTrail app. The Google Map application will then plot a course from the tracker to the child/trackee.



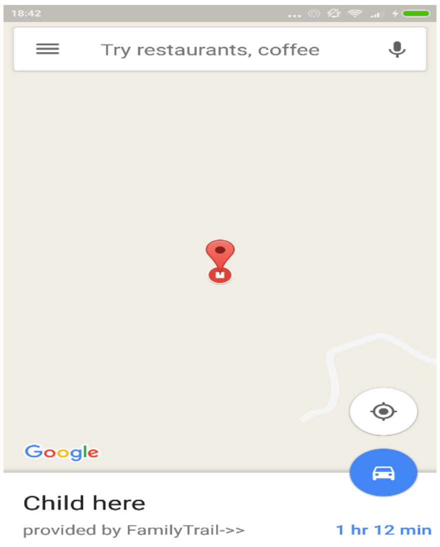


Figure 34 – Google Maps' link

Figure 35 – Location provided to Google Maps' application

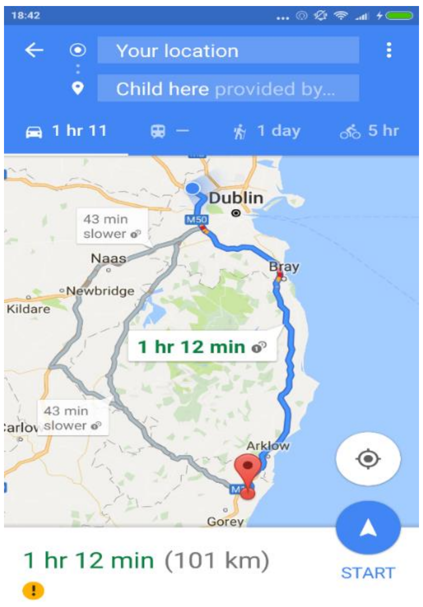
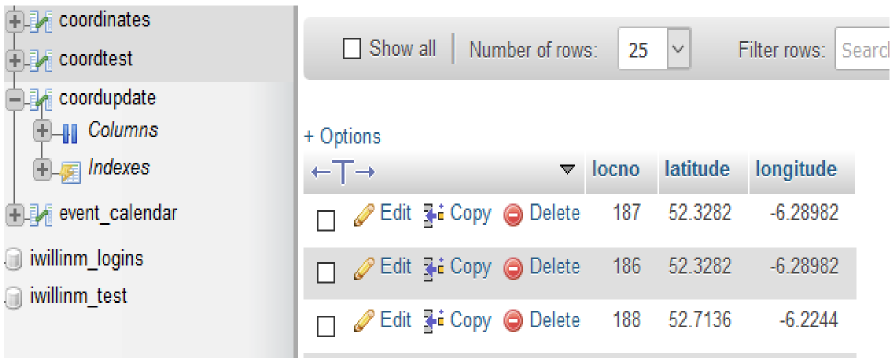


Figure 36 – Directions are plotted between the users in the Google Maps

Trackee's location name retrieval based on the current location and their marker placement on the map

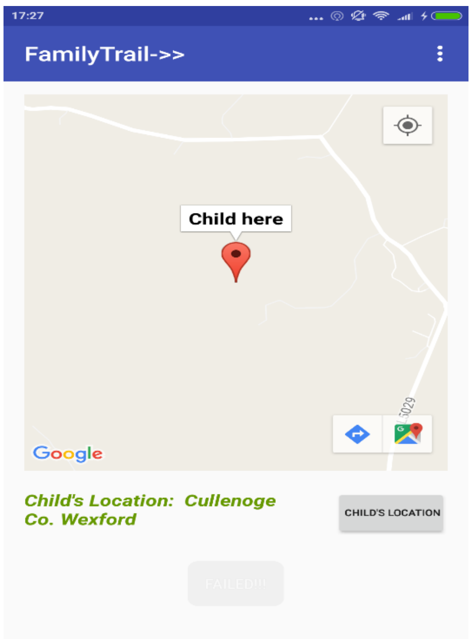
The trackee's location is retrieved from the coordinates table. The latest row contains the last known location of the trackee.



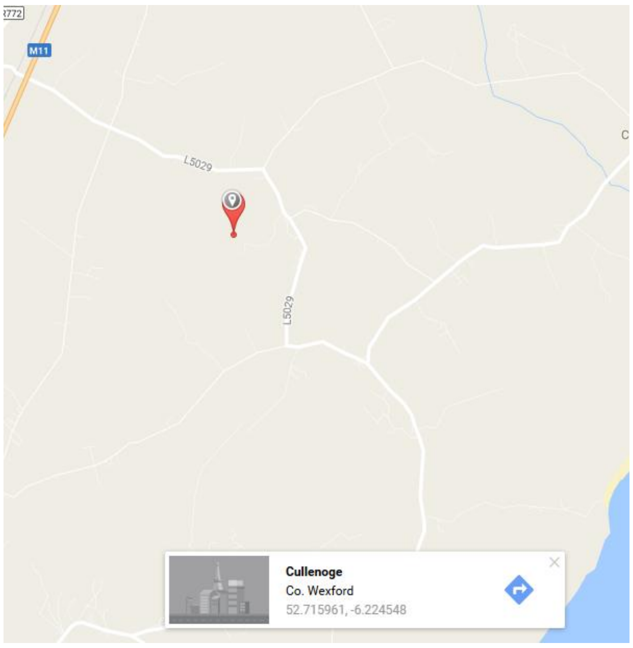
		locno	latitude	longitude
☐	Edit Copy Delete	187	52.3282	-6.28982
☐	Edit Copy Delete	186	52.3282	-6.28982
☐	Edit Copy Delete	188	52.7136	-6.2244

Figure 37 – Values from coordinates table

In Google's official map website, it can be seen that location shown on the app is found in wex'ford (Figure 39) which matches the displayed address. This confirms that valid details are retrieved from the database.



Child's Location: Cullenoge Co. Wexford



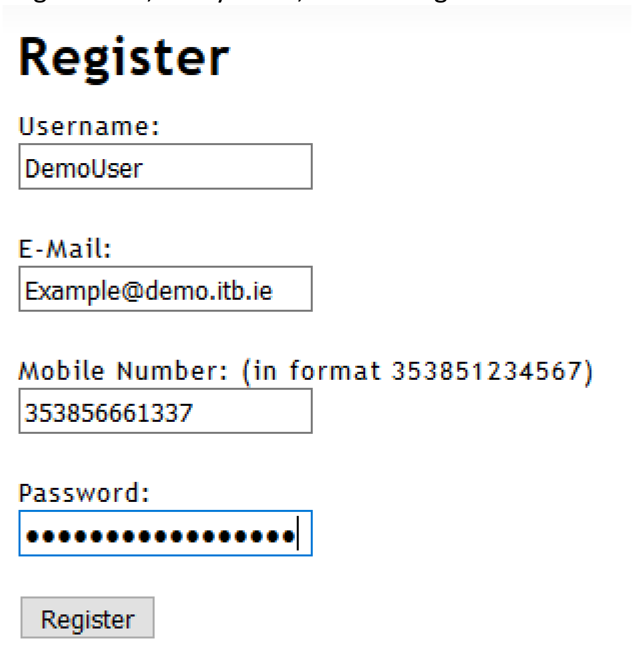
Cullenoge
Co. Wexford
52.715961,-6.224548

Figure 38 – location name displayed Figure 39 – Google Maps location based on coordinates

Website – Functionality

Items to test:

- Registration, verify in DB, and can log in with



Register

Username:
DemoUser

E-Mail:
Example@demo.itb.ie

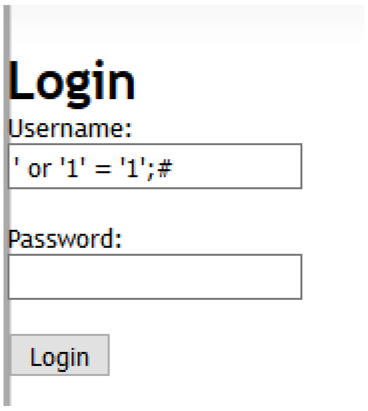
Mobile Number: (in format 353851234567)
353856661337

Password:
•••••••••••••••••

- Edit safe zone, Verify in DB, new coordinates show up on safe zone page load

- Change options Activate Tracking, Safe Zone and Change Interval
- Tracking

Website – Security testing



Login

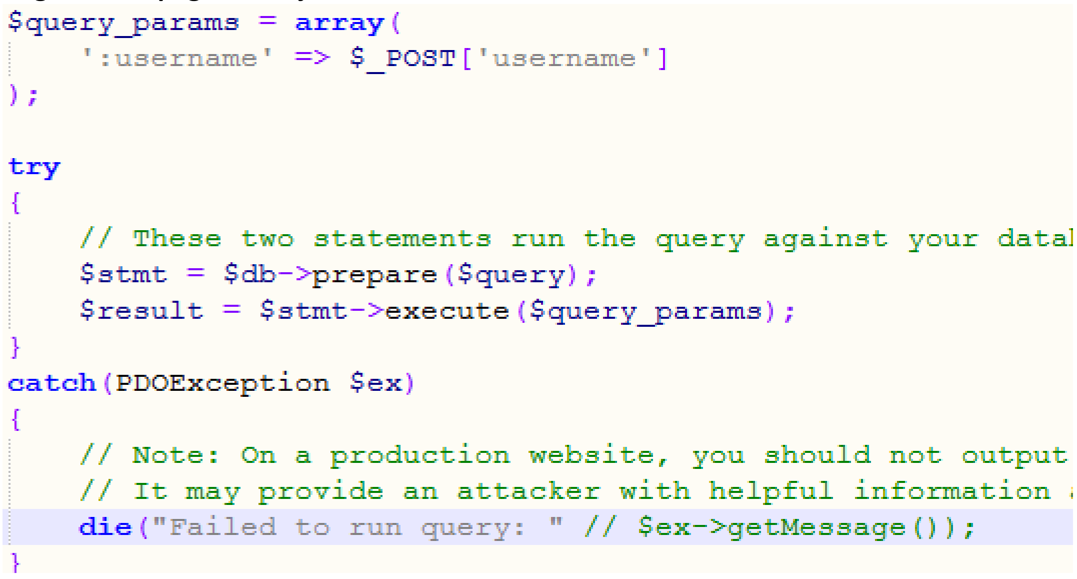
Username: ' or '1' = '1';#

Password:

Login

Log in page MySQL Injection::
This type of attack will not work as the malicious characters are filtered. See below in registration page to see the prepared statements used.

Registration page SQL injection::



```
$query_params = array(
    ':username' => $_POST['username']
);

try
{
    // These two statements run the query against your datal
    $stmt = $db->prepare($query);
    $result = $stmt->execute($query_params);
}
catch(PDOException $ex)
{
    // Note: On a production website, you should not output
    // It may provide an attacker with helpful information
    die("Failed to run query: " // $ex->getMessage());
}
```

Similarly, on the registration page, all input fields have been sanitised using prepared statements to ensure no malicious characters can cause damage or run unwanted commands in the database. As shown here the “$ex->getMessage()” has been commented out, which means the warning message will not be displayed as that could give an attacker valuable information on the systems they would like to attack.

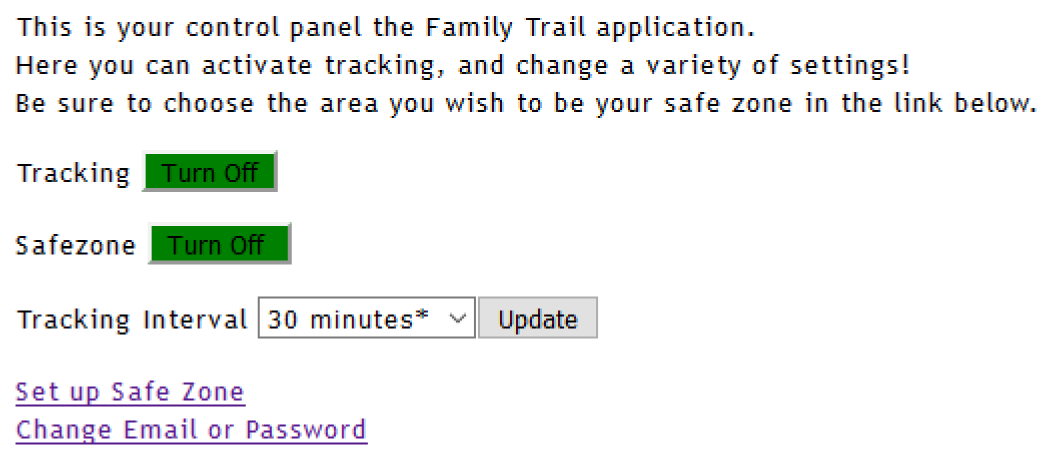
```php
<?php
if(empty($_SERVER['HTTPS']) || $_SERVER['HTTPS'] == "off"){
    $redirect = 'https://' . $_SERVER['HTTP_HOST'] . $_SERVER['REQUEST_URI'];
    header('HTTP/1.1 301 Moved Permanently');
    header('Location: ' . $redirect);
    exit();
}
?>
```

Session based access control

```php
<?php
    session_start();
    if (isset($_SESSION['user'])) {
    $username = $_SESSION['user'];
    } else {
    header("Location: login.php");
    }

    require_once './dbinfo.inc.php';
```

If a user tries to gain direct access to a restricted area of the site without logging in, this snippet of PHP code will start the session function. It will then check that the 'user' variable has been set, and if it has the value, which is the current users log in name, will be set to the variable $username, for future use. If the Session 'user' variable as not been initiated, then the user will be redirected to the login page.

Using HTTPS protocol::
To ensure the sensitive data is not being recorded on public WiFi, for example by an attacker using a packet sniffer, a PHP script will detect if the user has loaded the site through HTTP protocol and it will redirect them to the HTTPS port, and all subsequent links will use the secure protocol.

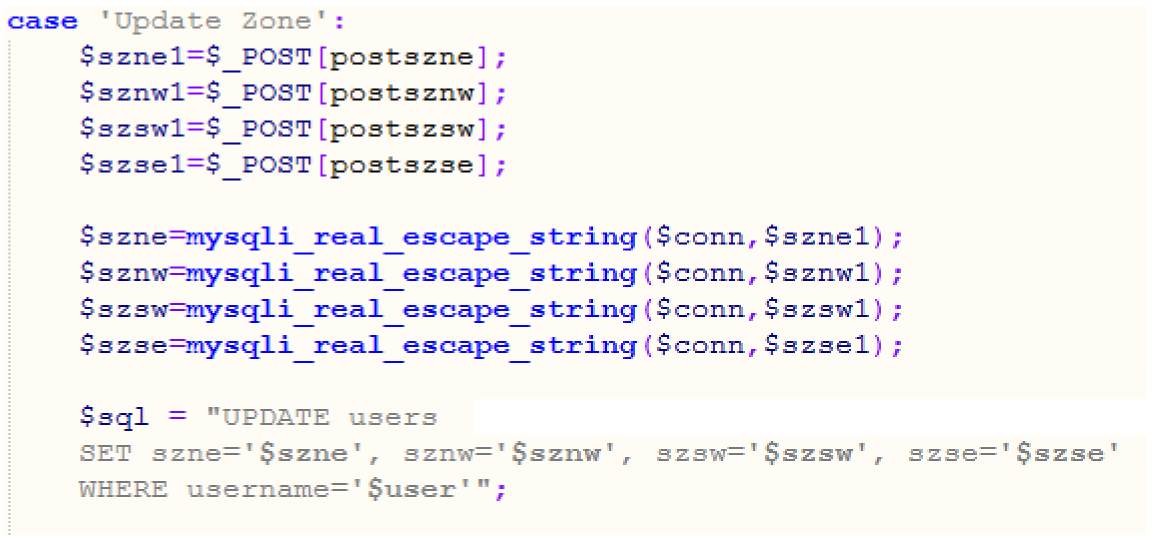

This is your control panel the Family Trail application.
Here you can activate tracking, and change a variety of settings!
Be sure to choose the area you wish to be your safe zone in the link below.

Tracking Turn Off

Safezone Turn Off

Tracking Interval 30 minutes* Update

Set up Safe Zone
Change Email or Password


```
case 'Update Zone':
    $szne1=$_POST[postszne];
    $sznw1=$_POST[postsznw];
    $szsw1=$_POST[postszsw];
    $szse1=$_POST[postszse];

    $szne=mysqli_real_escape_string($conn,$szne1);
    $sznw=mysqli_real_escape_string($conn,$sznw1);
    $szsw=mysqli_real_escape_string($conn,$szsw1);
    $szse=mysqli_real_escape_string($conn,$szse1);

    $sql = "UPDATE users
    SET szne='$szne', sznw='$sznw', szsw='$szsw', szse='$szse'
    WHERE username='$user'";
```

Safe Zone Co-ordinates update, filtering

In order to implement the Safe Zone feature, forms were used to pass POST data to the PHP script which updates the MySQL database. The text fields in the form were updated client-side by JavaScript which leaves them vulnerable to being changed by use of a proxy, and being replaced by malicious code. In order to defend the site from such attacks, the POST date received by the PHP script sanitizes the data before allowing it to interact with the database. MySQL has a function “real escape string” which will remove characters that could be used maliciously.

Implementation

To find out information about the app, users may visit the website, which describes the functionality of the app, and has a free registration form. They just need to provide their username, password, email address and phone number to begin. Upon logging in they have access to the full site, but before they can start tracking they must download the Android APK called ‘GPSclient’ from the Google Play market. This application is designed to run ideally on a smartwatch, but is compatible with any Android device that has internet and GPS. Upon installing the APK, the user can log in with the same details they registered on the websit, and this will link the child’s device to the parents account.

On the website upon logging in the home page provides a number of options to be configured. First the parents will decide the interval between location updates; there are choices ranging from every five minutes to 180 minutes. Next they can click into the Safe Zone feature which has been described previously. This brings them to a new page which has a Google Maps interface, that had a square which is pre-set over Dublin city. The user is able to move this box around, making it bigger or smaller to suit their requirements. Once they have decided the boundaries which the child should not leave while tracking is enabled, they click the 'Update Zone' button. This will save the location to the database, where a back end script is ready to monitor that any uploaded GPS coordinates do not reside outside the chosen area. The page is reloaded and the user may choose how they would like to be alerted if the child does leave the area, with the most popular option being an SMS message or email. They may also choose to set an alarm on the child's device so they realise they are not supposed to be venturing so far away.

Returning to the previous homepage the parents can now choose to activate the tracking, which means the child's device will upload GPS coordinates as often as the chosen interval dictates. They may also active the Safe Zone setting, which will activate a server side monitoring script which will be enabled. If that is the case, a message will be sent to the parents to inform them that the child's device is out of contact, and could be caused by battery running out, the child intentionally closing the application or turning off the device, the device losing coverage, or simply running out of credit. Once tracking is enabled the user can navigate to the Live Map page, which will show the location of the child in the most recent upload of coordinates. Here the parents can get directions to the child from their current location, which is uploaded through an Android application which the parents can also use to access most features available on the web interface.

After a few days tracking, the user may make use of the History Page, which displays a calendar, and any dates where tracking has been enabled provides a link to a Google Map, which displays all coordinates uploaded on that day so they can keep records of where the child was on certain dates. The next page is a Help Page, which includes a frequently-asked-questions section, as well as a live chat, to get in contact with a member of the development team, where any assistance necessary to ensure the customer gets the best experience possible from the app will be provided. The only link left on the website is to logout when the parents are leaving their computer, or more importantly if they are leaving a public computer, so the children's location will not be accessed by unwanted persons.

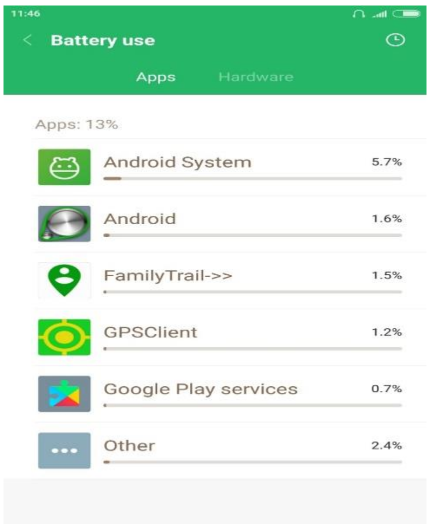
7.2 Conditions Testing

GPS Client & Family Trail

Battery Consumption and efficiency

The applications are tested under select conditions that resembles a real-life usage scenario. The conditions observed are used to calculate results for two hours of usage.

The client app is seen to be using 1.2%-1.5% of battery power for 10 mins of usage(Figure 30). Calculating the app usage for two hours puts the app at around 5% of utilization. Accumulating the results of both applications running at the same time for 10%-15% for the given time.



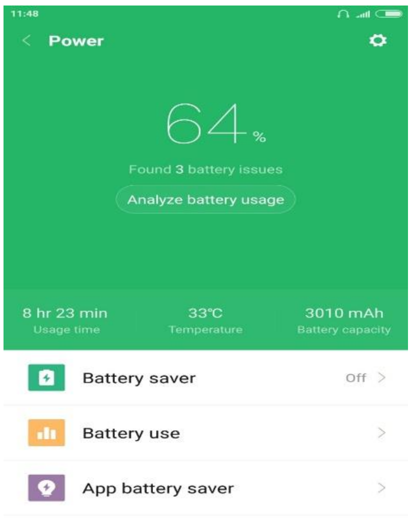
11:46

Battery use

Apps Hardware

Apps: 13%

Android System	5.7%
Android	1.6%
FamilyTrail->>	1.5%
GPSClient	1.2%
Google Play services	0.7%
Other	2.4%



11:48

Power

64%

Found 3 battery issues

Analyze battery usage

8 hr 23 min
Usage time

33℃
Temperature

3010 mAh
Battery capacity

Battery saver Off

Battery use

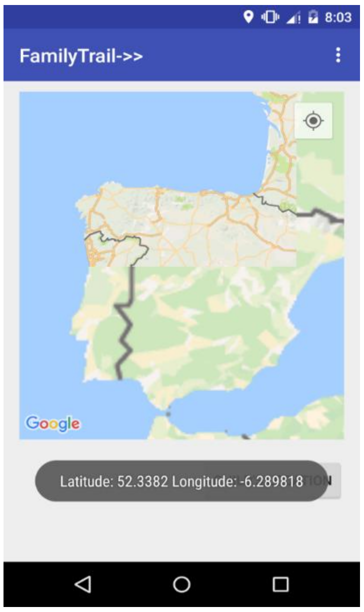
App battery saver

Figure 40 – applications battery usage Figure 39 – applications battery /temperature

GPS Tracking Accuracy

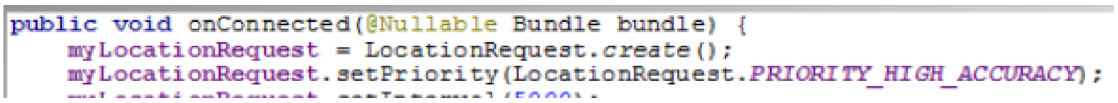
For Both applications tracking accuracy can vary depending on the environment they are tested in. In certain cases the GPS transmissions are affected inside closed houses due to thickness of the walls and interference. With bigger buildings and large areas of structures such as a school area or corporate offices, the GPS service is fairly accurate.

In the figure 40 the GPS coordinates values gathered by the service can be seen. By default Android's location service retrieves latitude and longitude values as double values with six decimal places for precise coordinates. However, for sending to database, the values are trimmed to 4-6 decimal points. The accuracy can vary immensely if the code is configured to use other options (Figure 42) such as Low power which reduces the accuracy of the coordinates greatly. (location settings) (Developers Android , 2017).



Latitude: 52.3382 Longitude: -6.289818

Figure 41 – Latitude values and Longitude values



```
public void onConnected(@Nullable Bundle bundle) {
    myLocationRequest = LocationRequest.create();
    myLocationRequest.setPriority(LocationRequest.PRIORITY_HIGH_ACCURACY);
```

Figure 42 – GPS Accuracy Options

8 Conclusion and Further Work

In completing this project and implementing the solutions for users in of need tracking, the group has learned a lot of technologies and how they perform in real world scenarios. By implementing three-modules, the group believes that the project can provide an unique and flexible with many options.

This Project can also be ported and extended to other devices such as tablets and smartwatche. The group also decided to target Android as the client application, as parents could get an Android Smartwatch for relatively low price. Many Smartwatches available are aimed at kids, and are very colourful, which would entice them to carry it around everywhere, as a phone could be easily discarded and thrown about.

Appendix A – Project Diary

The project diary is divided into two parts for each month during the development of the applications.

- **February** (First half) :
  - Group is gathered to discuss the Project components and ideas
  - Team roles are also assigned to each member.

- **February** (Second half) :
  - Aadhavan – Researching ideas for structure of the applications.
  - Mark – Research into the implementation of actual components

- **March** (First half) :
  - Aadhavan – Getting started on concepts of Android coding
  - Mark – Getting familiar with Google Maps API and web browser services.

- **March** (Second half) :

  - Aadhavan – Developing the prototype of the tracking application, getting into Android specific functions

- Mark – Website domain purchase, setting up the base of the web application using PHP and HTML

- **April** (First half) :
  - Aadhavan – Researching similar literary work of implementations found on the Google Play store, Implementation of google maps for the FamilyTrail application and retrieving values from database.
  - Mark -- Creating the Database and tables for retaining coordinates sent from the phone applications

- **April** (Second half) :
  - Aadhavan – Establishing connections from applications to the database, sending information to database from both the client and tracker application.
  - Mark – Finalizing the Website's map Implementation and adding necessary features

- **May** (First half) :
  - Aadhavan – Finalizing the Application's features and working with webserver to display locations based on time.
  - Mark – Cleaning up website features such as logins. Adding security mechanism on the website to prevent attacks such as SQL injections.

- **May** (Second half) :
  - Aadhavan – Final write-up on the Design of the applications and testing of the applications for various functions and conditions.
  - Mark – Testing the website for validity of the features implemented, security testing and write-up.

Appendix B – Code Listing

GPSClient Application

Main Activity.java

```
package com.example.itb.gpsclient;

import android.app.Activity;
import android.content.IntentFilter;
import android.os.AsyncTask;
import android.renderscript.Double2;
import android.support.v4.content.ContextCompat;
import android.support.v4.util.Pair;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.Manifest;
import android.content.Intent;
import android.content.pm.PackageManager;
import android.location.Location;
import android.location.LocationListener;
import android.location.LocationManager;
import android.os.Build;
import android.provider.Settings;
import android.support.annotation.NonNull;
import android.support.annotation.Nullable;
import android.support.v4.app.ActivityCompat;
import android.telephony.SmsManager;
import android.view.View;
import android.widget.Button;
import android.widget.TextView;
import android.widget.Toast;

import android.content.BroadcastReceiver;

import com.android.volley.Request;
import com.android.volley.RequestQueue;
import com.android.volley.Response;
import com.android.volley.VolleyError;
import com.android.volley.toolbox.StringRequest;
import com.android.volley.toolbox.Volley;

import java.util.ArrayList;
import java.util.HashMap;
import java.util.List;
import java.util.Map;

public class MainActivity extends AppCompatActivity {

    //  private Button b;
    // private TextView t;
    private LocationManager locationManager;
    private LocationListener listener;




    private BroadcastReceiver broadcastReceiver;

    private String coord = "hello";
    //  private static float lo;
    //  private static float lat;
    public String lo;
    public String lat;
    private BroadcastReceiver myReceiver;
    private static final String URL = "http://iwillinmy.com/coordUpdate.php";

    private static final String URL2 = "http://iwillinmy.com/testCoord.php";


    final String latAsString = String.valueOf(lat);
    final String loAsString = String.valueOf(lo);

    public static final String lati = "la1";
    public static final String longi = "lo1";

    final String lo1 = "-6.2422709";
    final String la1 = "53.3243201";


    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

      // sendSMSMessage();

        locationManager = (LocationManager) getSystemService(LOCATION_SERVICE);


        listener = new LocationListener() {
            @Override
            public void onLocationChanged(Location location) {
                double loD = location.getLongitude();
                double latD = location.getLatitude();

                double newlod = ((int) Math.round(loD * 1E6)) / 1E6;
                double newlatD = ((int) Math.round(latD * 1E6)) / 1E6;


                lo = String.valueOf(newlod);
                lat = String.valueOf(newlatD);


                sendCoord();
               // sendtorecords();

                Toast.makeText(getApplicationContext(),"Longitude :"+lat+"Latitude:
"+lo, Toast.LENGTH_LONG).show();

                getLo(loD);
                getLat(latD);

                if(location.getSpeed() <= 0){

                }

            }

            @Override
            public void onStatusChanged(String s, int i, Bundle bundle) {

            }

            @Override
            public void onProviderEnabled(String s) {
            }

            @Override
            public void onProviderDisabled(String s) {

                Intent i = new Intent(Settings.ACTION_LOCATION_SOURCE_SETTINGS);
                startActivity(i);
            }

        };

        if (ActivityCompat.checkSelfPermission(this,
Manifest.permission.ACCESS_FINE_LOCATION) != PackageManager.PERMISSION_GRANTED &&
ActivityCompat.checkSelfPermission(this,
Manifest.permission.ACCESS_COARSE_LOCATION) != PackageManager.PERMISSION_GRANTED) {
            return;
        }
        locationManager.requestLocationUpdates("gps", 5000, 0, listener);

        configure_button();
        // Toast.makeText(this,coord, Toast.LENGTH_LONG).show();
    }



    public void getLo(Double L) {
        String loAstring = String.valueOf(L);

    }

    public void getLat(Double L) {
        //L = lat;
        String latAsString = String.valueOf(L);

    }

    public void onResume()
    {
        super.onResume();
       startService();
        IntentFilter filter = new IntentFilter(Intent.ACTION_SHUTDOWN);
        myReceiver = new Shutdownreciever();
        registerReceiver(myReceiver, filter);
    }

    public void onPause()
    {
        super.onPause();

        this.unregisterReceiver(myReceiver);
    }

    public void onDestroy() {

        super.onDestroy();

    }

    public void startService(){

        Intent intent = new Intent(this,backgroundService.class);
        startService(intent);
    }
    public void sendCoord() {
        StringRequest stringRequest = new StringRequest(Request.Method.POST, URL,
                new Response.Listener<String>() {
                    @Override
                    public void onResponse(String response) {
                        System.out.println(response);
                        //
Toast.makeText(MainActivity.this,response,Toast.LENGTH_LONG).show();
                    }
                },
                new Response.ErrorListener() {
                    @Override
                    public void onErrorResponse(VolleyError error) {
                         Toast.makeText(MainActivity.this,"Internet is turned off!
Please enable mobile data or wifi",Toast.LENGTH_LONG).show();
                    }
                }) {
            @Override
            protected Map<String, String> getParams() {
                //List<String,String> params = new HashMap<String, String>();
                Map<String, String> params = new HashMap<String, String>();

                params.put("la1", lat);
                params.put("lo1", lo);
                return params;
            }

        };
        RequestQueue requestQueue = Volley.newRequestQueue(this);
        requestQueue.add(stringRequest);

    }

    public void sendtorecords() {
        StringRequest stringRequest = new StringRequest(Request.Method.POST, URL2,
                new Response.Listener<String>() {
                    @Override
                    public void onResponse(String response) {
                        System.out.println(response);
                        //
Toast.makeText(MainActivity.this,response,Toast.LENGTH_LONG).show();
                    }
                },
                new Response.ErrorListener() {
                    @Override
                    public void onErrorResponse(VolleyError error) {
                         Toast.makeText(MainActivity.this,"Internet is turned off!
Please enable mobile data or wifi",Toast.LENGTH_LONG).show();
                    }
                }) {
            @Override
            protected Map<String, String> getParams() {
                //List<String,String> params = new HashMap<String, String>();
                Map<String, String> params = new HashMap<String, String>();

                params.put("lattest", lat);
                params.put("longtest", lo);
                return params;
            }

        };
        RequestQueue requestQueue = Volley.newRequestQueue(this);
        requestQueue.add(stringRequest);

    }
    // @Override
    public void onRequestPermissionsResult(int requestCode, @NonNull String[]
permissions, @NonNull int[] grantResults) {
        switch (requestCode) {
            case 10:
                configure_button();
                break;
            default:
                break;
        }
    }

    void configure_button() {
        // checking for permissions
        if (ActivityCompat.checkSelfPermission(this,
Manifest.permission.ACCESS_FINE_LOCATION) != PackageManager.PERMISSION_GRANTED &&
ActivityCompat.checkSelfPermission(this,
Manifest.permission.ACCESS_COARSE_LOCATION) != PackageManager.PERMISSION_GRANTED) {
            if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.M) {
                requestPermissions(new
String[]{Manifest.permission.ACCESS_COARSE_LOCATION,
Manifest.permission.ACCESS_FINE_LOCATION, Manifest.permission.INTERNET}
                        , 10);
            }
            return;
        }
    }



}
```

backgroundSerivice.java

```
package com.example.itb.gpsclient;


import android.app.PendingIntent;
import android.app.Service;
import android.content.Intent;
import android.os.IBinder;
import android.support.annotation.Nullable;
import android.telephony.SmsManager;
import android.widget.Toast;

public class backgroundService extends Service {
    String phoneNo = "0809383383";
    String message = "Hello";
    @Override
    public void onCreate() {
        super.onCreate();
    }
```

```java
    @Override
    public void onTaskRemoved(Intent rootIntent)
    {
        phoneNo = "+0809383383";
        message = "The Tracking app is closed!!";
        try {
            SmsManager smsManager = SmsManager.getDefault();

            PendingIntent sentPI;
            String SENT = "SMS_SENT";

            sentPI = PendingIntent.getBroadcast(this, 0,new Intent(SENT), 0);

            smsManager.sendTextMessage(phoneNo, null, message, sentPI, null);
            Toast.makeText(this, "SMS sent.",
                    Toast.LENGTH_LONG).show();

        } catch (Exception e) {
            Toast.makeText(this,
                    "Sending SMS failed.",
                    Toast.LENGTH_LONG).show();
            e.printStackTrace();
        }
        Toast.makeText(this,"Tracking removed",Toast.LENGTH_LONG).show();

    }

    @Override
    public int onStartCommand(Intent intent, int flags, int startId) {

        Toast.makeText(this,"Service started",Toast.LENGTH_LONG).show();

       return START_STICKY;
    }

    @Override
    public void onDestroy() {
        super.onDestroy();


    }

    @Override
    public IBinder onBind(Intent intent) {
        return null;
    }
}
```

Shutdownreciever.java

```java
package com.example.itb.gpsclient;


import android.app.PendingIntent;
import android.content.BroadcastReceiver;
import android.content.Context;
import android.content.Intent;
import android.telephony.SmsManager;
import android.widget.Toast;

public class Shutdownreciever extends BroadcastReceiver {
    String phoneNo;
    String message;
    @Override
    public void onReceive(Context context, Intent intent) {
        if (Intent.ACTION_SHUTDOWN.equals(intent.getAction())) {
            Toast.makeText(context, "Phone is about to shutown,tracking will be
stopped!", Toast.LENGTH_LONG).show();
        }
            phoneNo = "0809383383";
            message = "test";
            try {
                SmsManager smsManager = SmsManager.getDefault();

                PendingIntent sentPI;
                String SENT = "SMS_SENT";

                sentPI = PendingIntent.getBroadcast(context, 0,new Intent(SENT),
0);

                smsManager.sendTextMessage(phoneNo, null, message, sentPI, null);
                Toast.makeText(context, "SMS sent.",
                        Toast.LENGTH_LONG).show();

            } catch (Exception e) {
                Toast.makeText(context,
                        "Sending SMS failed.",
                        Toast.LENGTH_LONG).show();
                e.printStackTrace();
            }

        }



}
```

Activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/activity_main"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    tools:context="com.example.itb.gpsclient.MainActivity">

    <ScrollView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_below="@+id/button"
        android:layout_alignParentRight="true"
        android:layout_alignParentEnd="true"
        android:id="@+id/scrollView2">

    </ScrollView>
    <TextView
        android:text="You have started the tracker, you can safely minimize the
application"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="14dp"
        android:id="@+id/textView"
        android:textStyle="normal|bold"
        android:textSize="18sp"
        android:layout_below="@+id/scrollView2"
        android:layout_alignParentLeft="true"
        android:layout_alignParentStart="true" />

    <TextView
        android:text="Note:
The Application will start a background service once the app is killed "
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@+id/textView"
        android:layout_marginTop="90dp"
        android:id="@+id/textView2"
        android:textSize="18sp"
        android:textStyle="normal|bold"
        android:layout_alignParentLeft="true"
        android:layout_alignParentStart="true" />

    <TextView
        android:text="An alert message will be sent to the Tracker's Phone number"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_centerVertical="true"
        android:layout_alignParentLeft="true"
        android:layout_alignParentStart="true"
        android:id="@+id/textView3"
        android:textSize="18sp"
        android:textStyle="normal|bold" />


</RelativeLayout>
```

Build.gradle

```
apply plugin: 'com.android.application'

android {
    compileSdkVersion 25
    buildToolsVersion "25.0.2"
    defaultConfig {
        applicationId "com.example.itb.gpsclient"
        minSdkVersion 15
        targetSdkVersion 25
        versionCode 1
        versionName "1.0"
        testInstrumentationRunner "android.support.test.runner.AndroidJUnitRunner"
    }
    buildTypes {
        release {
            minifyEnabled false
            proguardFiles getDefaultProguardFile('proguard-android.txt'),
'proguard-rules.pro'
        }
    }
}

dependencies {
    compile fileTree(dir: 'libs', include: ['*.jar'])
    androidTestCompile('com.android.support.test.espresso:espresso-core:2.2.2', {
        exclude group: 'com.android.support', module: 'support-annotations'
    })
    compile 'com.android.support:appcompat-v7:25.1.0'
    compile 'com.mcxiaoke.volley:library:1.0.19'
    testCompile 'junit:junit:4.12'
}
```

AndroidManifest.xml

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
package="com.example.itb.gpsclient">

    <uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION" />
    <uses-permission android:name="android.permission.ACCESS_FINE_LOCATION"/>
    <uses-permission android:name="android.permission.RECEIVE_BOOT_COMPLETED" />

    <uses-permission android:name="android.permission.SEND_SMS" />
    <uses-permission android:name="android.permission.WRITE_SMS" />

    <uses-permission android:name="android.permission.INTERNET"/>

    <uses-permission android:name="android.permission.RECEIVE_BOOT_COMPLETED" />

<application

    android:allowBackup="true"
    android:icon="@mipmap/clienticon"
    android:label="@string/app_name"
    android:supportsRtl="true"
    android:theme="@style/AppTheme">
    <activity android:name=".MainActivity">
        <intent-filter>
            <action android:name="android.intent.action.MAIN" />

            <category android:name="android.intent.category.LAUNCHER" />
        </intent-filter>
    </activity>


    <receiver
        android:enabled="true"
        android:exported="true"
        android:name=".Shutdownreciever"
        android:permission="android.permission.RECEIVE_BOOT_COMPLETED">

        <intent-filter>
            <action android:name="android.intent.action.ACTION_SHUTDOWN" />
            <action android:name="android.intent.action.BOOT_COMPLETED" />
            <action android:name="android.intent.action.QUICKBOOT_POWEROFF" />
            <action android:name="android.intent.action.QUICKBOOT_POWERON" />
            <category android:name="android.intent.category.DEFAULT" />
        </intent-filter>
    </receiver>

    <service android:name=".backgroundService"
        android:exported="false">
    </service>

</application>

</manifest>
```

FamilyTrail-Tracker Application

Mainactivity.java

```
package com.example.itb.gps;

import android.Manifest;
import android.app.Dialog;
import android.content.BroadcastReceiver;
import android.content.Context;
import android.content.Intent;
import android.content.IntentFilter;
import android.content.pm.PackageManager;
import android.location.Geocoder;
import android.location.Location;
import android.location.LocationListener;
import android.os.Build;
import android.os.Bundle;
import android.support.annotation.NonNull;
import android.support.annotation.Nullable;
import android.support.v4.app.ActivityCompat;
import android.support.v4.content.ContextCompat;
import android.support.v7.app.AppCompatActivity;
import android.support.v7.widget.Toolbar;
import android.text.TextUtils;
import android.view.Menu;
import android.view.MenuItem;
import android.view.View;
import android.widget.Button;
import android.widget.TextView;
import android.widget.Toast;

import com.android.volley.Request;
import com.android.volley.RequestQueue;
import com.android.volley.Response;
import com.android.volley.VolleyError;
import com.android.volley.toolbox.StringRequest;
import com.android.volley.toolbox.Volley;
import com.google.android.gms.common.ConnectionResult;
import com.google.android.gms.common.GoogleApiAvailability;
import com.google.android.gms.common.api.GoogleApiClient;
import com.google.android.gms.identity.intents.Address;
import com.google.android.gms.location.LocationRequest;
import com.google.android.gms.location.LocationServices;
import com.google.android.gms.maps.CameraUpdate;
import com.google.android.gms.maps.CameraUpdateFactory;
import com.google.android.gms.maps.GoogleMap;
import com.google.android.gms.maps.MapFragment;
import com.google.android.gms.maps.SupportMapFragment;
import com.google.android.gms.maps.OnMapReadyCallback;
import com.google.android.gms.maps.model.LatLng;
import org.json.JSONArray;
import org.json.JSONException;
import org.json.JSONObject;
import com.google.android.gms.location.LocationRequest;
import com.google.android.gms.location.LocationServices;
import com.google.android.gms.maps.model.MarkerOptions;


import java.io.IOException;
import java.util.HashMap;
import java.util.List;
import java.util.Map;


public class MainActivity extends AppCompatActivity implements OnMapReadyCallback,
GoogleApiClient.ConnectionCallbacks, GoogleApiClient.OnConnectionFailedListener,
com.google.android.gms.location.LocationListener {

    private Button track, stoptrack;
    private TextView textView;
    private BroadcastReceiver broadcastReceiver;
    private TextView textView1 ;
    private TextView textView3;

    public String getLocName;

    public String getLat;
    public String getLong;

    public double DGetlat;
    public double DGetlong;

    public double latD;
    public double longD;

    public double mylat;
    public double mylong;
    public String lats;
    public String longs;

    GoogleMap myGoogleMap;
    GoogleApiClient myGoogleApiClient;

    private static final String URL =
"http://iwillinmy.com/trackercoordUpdate.php";

    Button addDB;

    @Override
    protected void onResume() {
        super.onResume();
        if (broadcastReceiver == null) {
            broadcastReceiver = new BroadcastReceiver() {
                @Override
                public void onReceive(Context context, Intent intent) {

                    textView.append("\n" + intent.getExtras().get("coordinates"));
                    String getCoord = textView.getText().toString();

                }
            };
        }
        registerReceiver(broadcastReceiver, new IntentFilter("location_update"));
    }
```

```java
    @Override
    protected void onDestroy() {
        super.onDestroy();
        if (broadcastReceiver != null) {
            unregisterReceiver(broadcastReceiver);
        }
    }

    public void onConnected() {


    }

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

        setContentView(R.layout.activity_main);
        textView = (TextView) findViewById(R.id.textView);
        Toolbar toolbar = (Toolbar) findViewById(R.id.toolbar3);
        setSupportActionBar(toolbar);

        MapFragment mapFragment = (MapFragment)
getFragmentManager().findFragmentById(R.id.mapFragment);
        mapFragment.getMapAsync(this);

        textView1= (TextView) findViewById(R.id.textView2);

        track = (Button) findViewById(R.id.button2);

        getData();

        enable_buttons();

        if (!runtime_permissions())
            enable_buttons();


    }

    public boolean googleServicesAvailable() {
        GoogleApiAvailability api = GoogleApiAvailability.getInstance();
        int isAvailable = api.isGooglePlayServicesAvailable(this);
        if (isAvailable == ConnectionResult.SUCCESS) {
            return true;
        } else if (api.isUserResolvableError(isAvailable)) {
            Dialog dialog = api.getErrorDialog(this, isAvailable, 0);
            dialog.show();
        } else {
            Toast.makeText(this, "Cant connect to play services",
Toast.LENGTH_LONG).show();
        }
        return false;
    }


    private void enable_buttons() {

        track.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {

                try {
                    locate(null);
                } catch (IOException e) {
                    e.printStackTrace();
                }
            }
        });
    }

    private void getData(){
        String id = textView1.getText().toString().trim();
        if (id.equals("")) {
           // Toast.makeText(this, "Please enter an id", Toast.LENGTH_LONG).show();
            return;
        }


        String url = fetchcoord.DATA_URL;

        StringRequest stringRequest = new StringRequest(url, new
Response.Listener<String>() {
            @Override
            public void onResponse(String response) {
                showJSON(response);
            }
        },
                new Response.ErrorListener() {
                    @Override
                    public void onErrorResponse(VolleyError error) {
                       //
Toast.makeText(MainActivity.this,"",Toast.LENGTH_LONG).show();
                    }
                });

        RequestQueue requestQueue = Volley.newRequestQueue(this);
        requestQueue.add(stringRequest);



    }
    private void showJSON(String response){
        String lat="";
        String longi="";

        try {
            JSONObject jsonObject = new JSONObject(response);
            JSONArray result = jsonObject.getJSONArray(fetchcoord.JSON_ARRAY);
            JSONObject collegeData = result.getJSONObject(0);
            lat = collegeData.getString(fetchcoord.LATITUDE);
            longi = collegeData.getString(fetchcoord.LONGITUDE);

            getLat = lat;
            getLong = longi;


            latD = Double.valueOf(getLat);
            longD = Double.valueOf(getLong);
        } catch (JSONException e) {
            e.printStackTrace();
        }
        textView1.setText("Child's Location: "+""+getLocName);

    }

    private boolean runtime_permissions() {
        if (Build.VERSION.SDK_INT >= 23 && ContextCompat.checkSelfPermission(this,
Manifest.permission.ACCESS_FINE_LOCATION) != PackageManager.PERMISSION_GRANTED &&
ContextCompat.checkSelfPermission(this, Manifest.permission.ACCESS_COARSE_LOCATION)
```

```
!= PackageManager.PERMISSION_GRANTED) {

            requestPermissions(new
String[]{Manifest.permission.ACCESS_FINE_LOCATION,
Manifest.permission.ACCESS_COARSE_LOCATION}, 100);

            return true;
        }
        return false;
    }


    @Override
    public void onRequestPermissionsResult(int requestCode, @NonNull String[]
permissions, @NonNull int[] grantResults) {
        super.onRequestPermissionsResult(requestCode, permissions, grantResults);
        if (requestCode == 100) {
            if (grantResults[0] == PackageManager.PERMISSION_GRANTED &&
grantResults[1] == PackageManager.PERMISSION_GRANTED) {

            } else {
                runtime_permissions();
            }
        }
    }

    @Override
    public void onMapReady(GoogleMap googleMap) {

        myGoogleMap = googleMap;

        if (ActivityCompat.checkSelfPermission(this,
Manifest.permission.ACCESS_FINE_LOCATION) != PackageManager.PERMISSION_GRANTED &&
ActivityCompat.checkSelfPermission(this,
Manifest.permission.ACCESS_COARSE_LOCATION) != PackageManager.PERMISSION_GRANTED) {
            // TODO: Consider calling

            return;
        }
        myGoogleMap.setMyLocationEnabled(true);


        myGoogleApiClient = new GoogleApiClient.Builder(this)
                .addApi(LocationServices.API)
                .addConnectionCallbacks(this)
                .addOnConnectionFailedListener(this)
                .build();
        myGoogleApiClient.connect();
    }



    private void goToLocation(double lat, double lng) {
        LatLng ll = new LatLng(lat, lng);
        CameraUpdate update = CameraUpdateFactory.newLatLng(ll);
        myGoogleMap.moveCamera(update);
    }


    private void goTomyLoc(double lat, double lng, float zoom) {
        LatLng ll = new LatLng(lat, lng);
        CameraUpdate update = CameraUpdateFactory.newLatLngZoom(ll, zoom);
        myGoogleMap.moveCamera(update);
        //Toast.makeText(this, "?????", Toast.LENGTH_LONG).show();
    }

    public boolean onCreateOptionsMenu(Menu menu) {
        getMenuInflater().inflate(R.menu.menu_main, menu);
        return super.onCreateOptionsMenu(menu);
    }

    public boolean onOptionsItemSelected(MenuItem item) {
        switch (item.getItemId()) {
            case R.id.mapTypeNormal:
                myGoogleMap.setMapType(GoogleMap.MAP_TYPE_NORMAL);
                break;
            case R.id.mapTypeSatellite:
                myGoogleMap.setMapType(GoogleMap.MAP_TYPE_SATELLITE);
                break;
            case R.id.mapTypeTerrain:
                myGoogleMap.setMapType(GoogleMap.MAP_TYPE_TERRAIN);
                break;
            case R.id.mapTypeHybrid:
                myGoogleMap.setMapType(GoogleMap.MAP_TYPE_HYBRID);
                break;

            default:
                break;
        }

        return super.onOptionsItemSelected(item);
    }

    LocationRequest myLocationRequest;

    @Override
    public void onConnected(@Nullable Bundle bundle) {
        myLocationRequest = LocationRequest.create();
        myLocationRequest.setPriority(LocationRequest.PRIORITY_HIGH_ACCURACY);
        myLocationRequest.setInterval(5000);

        if (ActivityCompat.checkSelfPermission(this,
Manifest.permission.ACCESS_FINE_LOCATION) != PackageManager.PERMISSION_GRANTED &&
ActivityCompat.checkSelfPermission(this,
Manifest.permission.ACCESS_COARSE_LOCATION) != PackageManager.PERMISSION_GRANTED) {
            // TODO: Consider calling

            return;
        }
        LocationServices.FusedLocationApi.requestLocationUpdates(myGoogleApiClient,
myLocationRequest, this);

    }

    @Override
    public void onConnectionSuspended(int i) {

    }

    @Override
    public void onConnectionFailed(@NonNull ConnectionResult connectionResult) {

    }

    @Override
    public void onLocationChanged(Location location) {
        if(location == null){
            Toast.makeText(this, "Cant get current location",
Toast.LENGTH_LONG).show();
```

```
        } else {
            LatLng myloc1 = new LatLng(location.getLatitude(),
location.getLongitude());
//              latD = Double.valueOf(getLat);
            //     longD = Double.valueOf(getLong);

            mylat = location.getLatitude();
            mylong = location.getLongitude();
            double myLongD = ((int) Math.round(mylong * 1E6)) / 1E6;
            double myLatD = ((int) Math.round(mylat * 1E6)) / 1E6;
            longs = String.valueOf(myLongD);
            lats = String.valueOf(myLatD);


            getData();
            LatLng newlalo = new LatLng(latD,longD);

            CameraUpdate update = CameraUpdateFactory.newLatLngZoom(newlalo, 15);


            MarkerOptions options1 = new MarkerOptions()
                    .title("You are here")
                    .position(myloc1);

            MarkerOptions options2 = new MarkerOptions()
                    .title("Child here")
                    .position(new LatLng(latD,longD));

            Toast.makeText(MainActivity.this,"Latitude: "+lats+" Longitude:
"+longs,Toast.LENGTH_LONG).show();


            myGoogleMap.addMarker(options1);
            myGoogleMap.addMarker(options2);

            myGoogleMap.animateCamera(update);


            enable_buttons();

            sendCoord();

        }

    }
    public void locate(View view)throws IOException{
        Geocoder gc = new Geocoder(this);
        String location = "Paris";

        List<android.location.Address> list = gc.getFromLocation(latD ,longD,1);
        try {
            android.location.Address address = list.get(0);
            if (list != null && list.size() > 0) {

                StringBuilder strAddress = new StringBuilder();

                // getLocName = address.getLocality();

                for (int i = 0; i < address.getMaxAddressLineIndex(); i++) {
                    strAddress.append(address.getAddressLine(i)).append("\n");
                }
                getLocName = " " + strAddress.toString();

            } else {
            }
        }
        catch(Exception e){


        }

    }

    public void sendCoord(){
        StringRequest stringRequest = new StringRequest(Request.Method.POST, URL,
                new Response.Listener<String>() {
                    @Override
                    public void onResponse(String response) {
                        System.out.println(response);

                    }
                },
                new Response.ErrorListener() {
                    @Override
                    public void onErrorResponse(VolleyError error) {
                        Toast.makeText(MainActivity.this,"Internet is disabled,
please enable wifi or mobile data!",Toast.LENGTH_LONG).show();
                    }
                }){
            @Override
            protected Map<String,String> getParams(){
                //List<String,String> params = new HashMap<String, String>();
                Map<String,String> params = new HashMap<>();

                params.put("mylat",lats);
                params.put("mylo", longs);
                return params;
            }

        };
        RequestQueue requestQueue = Volley.newRequestQueue(this);
        requestQueue.add(stringRequest);

    }

    public void onStatusChanged(String provider, int status, Bundle extras) {

    }

    public void onProviderEnabled(String provider) {

    }

    public void onProviderDisabled(String provider) {

    }
}
```

fetchcoord.java

```
package com.example.itb.gps;
```

```
public class fetchcoord {
    public static final String DATA_URL = "http://iwillinmy.com/sendcoord.php";
    public static final String LATITUDE = "latitude";
    public static final String LONGITUDE = "longitude";
    public static final String JSON_ARRAY = "result";
}
```

ContentMain.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/content_main"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    app:layout_behavior="@string/appbar_scrolling_view_behavior"
    tools:context="com.example.itb.gps.MainActivity"
    tools:showIn="@layout/activity_main">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/textView"
        android:textStyle="normal|bold|italic"
        android:textColor="@android:color/holo_green_dark" />

    <fragment
        android:layout_width="360dp"
        android:layout_height="370dp"
        android:id="@+id/mapFragment"
        android:name="com.google.android.gms.maps.MapFragment"
        android:layout_below="@+id/textView"
        android:layout_marginTop="32dp" />

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignTop="@+id/button2"
        android:layout_alignParentLeft="true"
        android:layout_alignParentStart="true"
        android:id="@+id/textView2"

        android:textSize="14sp"
        android:textColor="@android:color/holo_green_dark"
        android:textStyle="normal|bold|italic" />

    <Button
        android:text="Child's Location"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/button2"
        android:layout_marginTop="19dp"
        android:layout_below="@+id/mapFragment"
        android:layout_alignParentRight="true"
        android:layout_alignParentEnd="true" />

</RelativeLayout>
```

Activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.design.widget.CoordinatorLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:fitsSystemWindows="true"
    tools:context="com.example.itb.gps.MainActivity">

    <android.support.v7.widget.Toolbar
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:background="?attr/colorPrimary"
        android:theme="?attr/actionBarTheme"
        android:minHeight="?attr/actionBarSize"
        android:id="@+id/toolbar3" />

    <include layout="@layout/content_main" />

</android.support.design.widget.CoordinatorLayout>
```

AndoidManifest.xml

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.itb.gps">

    <!-- setting permissions for accessing location -->

    <uses-permission android:name="android.permission.INTERNET" />
    <uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
    <uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION"/>
    <uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
  <uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />


  <permission
        android:name="com.example.itb.gps.permission.MAPS_RECEIVE"
        android:protectionLevel="signature"/>

    <uses-permission android:name="com.example.itb.gps.permission.MAPS_RECEIVE" />
    <uses-permission
android:name="com.google.android.providers.gsf.permissions.READ_GSERVICES" />
```

```
    <uses-feature
        android:glEsVersion="0x00020000"
        android:required="true"
        />


    <!-- setting permissions usage for accessing location -->



    <application
        android:allowBackup="true"
        android:icon="@mipmap/trakcericon
         android:label="@string/app_name"
        android:supportsRtl="true"
        android:theme="@style/AppTheme">
        <activity
            android:name=".MainActivity"
            android:label="@string/app_name"
            android:theme="@style/AppTheme.NoActionBar">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>

        <service android:name=".gpsTrack"/>

        <meta-data android:name="com.google.android.maps.v2.API_KEY"
            android:value="AIzaSyCshrrxBVsfsEx3Xl-gxa0SkgwDgo" />

        <meta-data android:name="com.google.android.gms.version"
            android:value="@integer/google_play_services_version" />

    </application>

</manifest>
```

Build.gradle

```
apply plugin: 'com.android.application'

android {
    compileSdkVersion 25
    buildToolsVersion "25.0.2"
    defaultConfig {
        applicationId "com.example.itb.gps"
        minSdkVersion 15
        targetSdkVersion 25
        versionCode 1
        versionName "1.0"
        testInstrumentationRunner "android.support.test.runner.AndroidJUnitRunner"
    }
    buildTypes {
        release {
            minifyEnabled false
            proguardFiles getDefaultProguardFile('proguard-android.txt'),
'proguard-rules.pro'
        }
    }
}

dependencies {
    compile fileTree(dir: 'libs', include: ['*.jar'])
    androidTestCompile('com.android.support.test.espresso:espresso-core:2.2.2', {
        exclude group: 'com.android.support', module: 'support-annotations'
    })
    compile 'com.android.support:appcompat-v7:25.1.0'
    compile 'com.android.support:design:25.1.0'
    compile 'com.google.firebase:firebase-database:10.0.1'
    testCompile 'junit:junit:4.12'
    compile 'com.google.android.gms:play-services:10.0.1'
    compile 'com.mcxiaoke.volley:library:1.0.19'
}
```

Web Application

Tracking.php

```
<?php
    session_start();

    require 'login.php';


    $lat;
    $long;
    $trackerlat;
    $trackerlong;
   // $latA;
   // $longB;


    $response = array();



    $host = "iwillinmy.com";
    $userDname = "iwillinm_gpsuser";
    $Dpassword = "password";
    $db_name = "iwillinm_gps";
    $table_name = "coordinates";
    $table_name2 ="coordupdate";

    $conn2 = mysqli_connect($host, $userDname, $Dpassword, $db_name);
    // mysqli_select_db($conn,$db_name);

    $query2 = "SELECT * FROM $table_name ORDER BY locno ASC";
    $result2= mysqli_query($conn2, $query2) or
die("FAILED!!!".mysqli_error($conn2));

    while($row = mysqli_fetch_array($result2)){

        $lat = $row['latitude'];
        $long = $row['longitude'];

    }

    $conn3 = mysqli_connect($host, $userDname, $Dpassword, $db_name);
    $query6 = "SELECT * FROM $table_name2 ORDER BY id ASC ";
    $result6= mysqli_query($conn3, $query6) or die("");

    if(!$result6){
        echo '';
    }
    else{

    }
    while($row2 = mysqli_fetch_array($result6)){

    $trackerlat = $row2['latitude'];
    $trackerlong = $row2['longitude'];

    }




?>
    <html>
    <section>
        <div id="contents">
           <?php //echo 'Hello,'.$_SESSION['username']; ?>
        </div>

        <div id="one"><div id="map"
style="width:1140px;height:750px;background:white"></div>
            <div id="key">
                <form>
                    <fieldset>
                        <legend><h2>Key-</h2></legend>
                        <h3 style="color:green"> Green Markers represent Child</h3>
                        <h3 style="color:red">Red Markers represent You(The
Tracker)</h3>
                    </fieldset>

                </form>
            </div>
        </div>
        <div id="two">
             <div id="right-panel">
                 <h3>Directions to Child</h3>
                <p>Total Distance: <span id="total"></span></p>
            </div>
        </div>
        <link rel="stylesheet" href="style">
        <body>


        <script>
            function myMap() {

                var directionsService = new google.maps.DirectionsService;

                var mylocat = new google.maps.LatLng();

                var coordla = '<?php echo $lat;?>';
                var comma = ',';
                var coordlo = '<?php echo $long;?>';
                var coords = coordla.concat(comma, coordlo);
                var b = coords.split(",");
                //document.write(b);

                var trackerla = '<?php echo $trackerlat;?>';
                var trackerlo = '<?php echo $trackerlong;?>';
                var trackercoords = trackerla.concat(comma, trackerlo);
                var trackb = trackercoords.split(",");


                var coordsN = new google.maps.LatLng(parseFloat(b[0]),
parseFloat(b[1]));
                var TrcoordsN = new google.maps.LatLng(parseFloat(trackb[0]),
parseFloat(trackb[1]));
```

```
                //document.write(coords);
                //document.write(coordsN);

                Chcoords = new google.maps.LatLng(parseFloat(b[0]),
parseFloat(b[1]));
                Trcoords = new google.maps.LatLng(parseFloat(trackb[0]),
parseFloat(trackb[1]));

                var mapOptions = {
                    center: new google.maps.LatLng(parseFloat(b[0]),
parseFloat(b[1])),
                    zoom: 2,
                    mapTypeId: google.maps.MapTypeId.HYBRID
                };
                var map = new google.maps.Map(document.getElementById("map"),
mapOptions);


                // mylocationg = new google.maps.LatLng(53.3282, -6.28982);

                document.write(mylocat + "LOC HERE!!");

                infowindow = new google.maps.InfoWindow();



                 var marker = new google.maps.Marker({
                 position: (coordsN),
                 title: 'Child here!!!'

                 });
                 marker.setIcon('child.png');



                 google.maps.event.addListener(marker, 'click', function() {

                 infowindow.setContent("Child here");
                 infowindow.open(map, this);
                 });

                var marker2 = new google.maps.Marker({
                    position: (TrcoordsN),
                    title: 'You are here'

                });
                marker2.setIcon('parent.png');

                google.maps.event.addListener(marker2, 'click', function() {

                    infowindow.setContent("You are here");
                    infowindow.open(map, this);
                });

                /*


                 */

                var transitLayer = new google.maps.TransitLayer();
                transitLayer.setMap(map);

                marker.setMap(map);
                marker2.setMap(map);
                var directionsDisplay = new google.maps.DirectionsRenderer({
                    draggable: true,
                    map: map,
                    panel: document.getElementById('right-panel')

                });


                directionsDisplay.setMap(map);

                directionsDisplay.addListener('directions_changed', function () {
                    computeTotalDistance(directionsDisplay.getDirections());
                });


                displayRoute(Chcoords, Trcoords, directionsService,
directionsDisplay);




            function displayRoute(origin, destination, service, display) {
                service.route({
                    origin: origin,
                    destination: destination,
                   // waypoints: [{location: 'Dublin, IE'}, {location: 'Cork,
IE'}],

                    travelMode: 'DRIVING',
                    avoidTolls: true
                }, function(response, status) {
                    if (status === 'OK') {
                        display.setDirections(response);
                    } else {
                        alert('Could not display directions due to: ' + status);
                    }
                });
         }





        </script>

        <script
src="https://maps.googleapis.com/maps/api/js?key=AIzaSyBnRki2JB3obMHyVbkxW0Hzr74hmP
GAz3I&libraries=places&callback=myMap&" async defer></script>

        <script type="text/javascript" src="jquery.js"> </script>
        <script type="text/javascript">
            $(document).ready(function () {
                setInterval(function () {
                    $('#map').load('tracking.php')
                }, 97000);

            });

        </script>
        </body>
        </section>
    </html>
<?php
```

Help.php

```
<?php
    session_start();
    if (isset($_SESSION['user'])) {
      $username = $_SESSION['username'];
    } else {
      header("Location: login.php");
    }

?>


<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"
"http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<link rel="stylesheet" type="text/css" href="css/style.css" />
<title>Help - Family Trail</title>
<link rel="apple-touch-icon" sizes="180x180" href="/images/favicon/apple-touch-
icon.png">
<link rel="icon" type="image/png" sizes="32x32" href="/images/favicon/favicon-
32x32.png">
<link rel="icon" type="image/png" sizes="16x16" href="/images/favicon/favicon-
16x16.png">
<link rel="manifest" href="/images/favicon/manifest.json">
<link rel="shortcut icon" href="/images/favicon/favicon.ico">
<meta name="msapplication-config" content="/images/favicon/browserconfig.xml">
<meta name="theme-color" content="#ffffff">
</head>

<body>
    <div id="page">

        <div id="header">
           <h1>Family Trail</h1>
            <ul>
                      <li><a href="home.php">Home</a></li>
                   <li><a href="livemap.php">Live Map</a></li>
                <li><a href="history.php"> History</a></li>
                <li><a href="help.php">Get Help</a></li>
                <li><a href="logout.php">Logout</a></li>
          </ul>
      </div>

        <div id="main">

           <div class="main_top">
               <h1>Your #1 Family Safety Solution - for your peace of mind!<span
class="links" style="font-weight: bold; font-size: 24px; text-align:
center;"></span></h1>
            </div>

               <div class="main_body">
                <p>Welcome <?php echo $_SESSION['user']; ?>,</p>
                <p>Here will contain a webchat, FAQ and contact information, for
timely customer support! ;)</p>
                <p><span style="text-align: center"><img
src="images/livechat_icon.png" alt="" width="472" height="236"
```

```
align="middle"/></span>
            </div>

               <div class="main_bottom"></div>

        </div>



        <div id="footer">
        <p>
        <a href="http://www.aszx.net">web development</a> by <a
href="http://www.bryantsmith.com">bryant smith</a>
        </p>
        </div>

        </div>
</body>
</html>
```

Index.php

```
<?php
if(empty($_SERVER['HTTPS']) || $_SERVER['HTTPS'] == "off"){
    $redirect = 'https://' . $_SERVER['HTTP_HOST'] . $_SERVER['REQUEST_URI'];
    header('HTTP/1.1 301 Moved Permanently');
    header('Location: ' . $redirect);
    exit();
}
?>


<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"
"http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<link rel="stylesheet" type="text/css" href="css/style.css" />
<title>Family Trail</title>
<link rel="apple-touch-icon" sizes="180x180" href="/images/favicon/apple-touch-
icon.png">
<link rel="icon" type="image/png" sizes="32x32" href="/images/favicon/favicon-
32x32.png">
<link rel="icon" type="image/png" sizes="16x16" href="/images/favicon/favicon-
16x16.png">
<link rel="manifest" href="/images/favicon/manifest.json">
<link rel="shortcut icon" href="/images/favicon/favicon.ico">
<meta name="msapplication-config" content="/images/favicon/browserconfig.xml">
<meta name="theme-color" content="#ffffff">
</head>

<body>
    <div id="page">

        <div id="header">
           <h1>Family Trail</h1>
            <ul>
                      <li><a href="index.php">Home</a></li>
                   <li><a href="livemap.php">Live Map</a></li>
                <li><a href="history.php"> History</a></li>
                <li><a href="register.php">Register</a></li>
                <li><a href="login.php">Login</a></li>
            </ul>
        </div>

        <div id="main">

            <div class="main_top">
                <h1>Your #1 Family Safety Solution - for your peace of mind!<span 
class="links" style="font-weight: bold; font-size: 24px; text-align: 
center;"></span></h1>
            </div>

                <div class="main_body">
                 <p>Welcome, </p>
                 <p>This is the homepage for our college project. <br />
                 We are working to create a reliable tracking solution for parents, 
to know  their children are safe, and not wandering off when they should not 
be!</p>
                 <p>Our app will help you see where your children are in real time, 
as well as checking the history to see where they have been in recent days!</p>
                 <p>We have included a feature to allow you to set a safe zone, and 
if they child leaves the safe zone, you can choose from multiple options how the 
program should react. For example, you could get a phonecall/email or the childs 
phone could buzz to warn them to stay in bounds!</p>
                 <p>We also have a safety button on the childs phone so they can 
press this to get in contact with you if they are lost, and we will give you a 
number of ways for your child to get home, or simply their location so you can 
drive and pick them up!</p>
            </div>

                <div class="main_bottom"></div>

        </div>



        <div id="footer">
        <p>
        <a href="http://www.aszx.net">web development</a> by <a 
href="http://www.bryantsmith.com">bryant smith</a>
        </p>
        </div>

        </div>
</body>
</html>
```

Login.php

```
<?php 

    // First we execute our common code to connection to the database and start the 
session 
    require("common.php"); 

    // This variable will be used to re-display the user's username to them in the 
    // login form if they fail to enter the correct password.  It is initialized 
here 
    // to an empty value, which will be shown if the user has not submitted the 
form. 
    $submitted_username = ''; 
    // This if statement checks to determine whether the login form has been
submitted
    // If it has, then the login code is run, otherwise the form is displayed
    if(!empty($_POST))
    {
        // This query retreives the user's information from the database using
        // their username.
        $query = "SELECT id, username, password, salt, email FROM users WHERE
username = :username";

        // The parameter values
        $query_params = array(':username' => $_POST['username']);

        try
        {
            // Execute the query against the database
            $stmt = $db->prepare($query);
            $result = $stmt->execute($query_params);
        }
        catch(PDOException $ex)
        {
            // Note: On a production website, you should not output $ex-
>getMessage().
            // It may provide an attacker with helpful information about your code.
            die("Failed to run query: " . $ex->getMessage());
        }

        // This variable tells us whether the user has successfully logged in or
not.
        // We initialize it to false, assuming they have not.
        // If we determine that they have entered the right details, then we switch
it to true.
        $login_ok = false;

        // Retrieve the user data from the database.  If $row is false, then the
username
        // they entered is not registered.
        $row = $stmt->fetch();
        if($row)
        {
            // Using the password submitted by the user and the salt stored in the
database,
            // we now check to see whether the passwords match by hashing the
submitted password
            // and comparing it to the hashed version already stored in the
database.
            $check_password = hash('sha256', $_POST['password'] . $row['salt']);
            for($round = 0; $round < 65536; $round++)
            {
                $check_password = hash('sha256', $check_password . $row['salt']);
            }

            if($check_password === $row['password'])
            {
                // If they do, then we flip this to true
                $login_ok = true;
            }
        }

        // If the user logged in successfully, then we send them to the private
members-only page
        // Otherwise, we display a login failed message and show the login form
again
        if($login_ok)
        {
            // Here I am preparing to store the $row array into the $_SESSION by
            // removing the salt and password values from it.  Although $_SESSION
is
            // stored on the server-side, there is no reason to store sensitive
values
            // in it unless you have to.  Thus, it is best practice to remove these
            // sensitive values first.
            unset($row['salt']);
            unset($row['password']);

            // This stores the user's data into the session at the index 'user'.
            // We will check this index on the private members-only page to
determine whether
            // or not the user is logged in.  We can also use it to retrieve
            // the user's details.
            $_SESSION['user'] = $row['username'];
            //$_SESSION['user'] = $user_name;

            // Redirect the user to the private members-only page.
            header("Location: home.php");
            die("Redirecting to: home.php");
        }
        else
        {
            // Tell the user they failed
            print("Login Failed.");

            // Show them their username again so all they have to do is enter a new
            // password.  The use of htmlentities prevents XSS attacks.  You should
            // always use htmlentities on user submitted values before displaying
them
            // to any users (including the user that submitted them).  For more
information:
            // http://en.wikipedia.org/wiki/XSS_attack
            $submitted_username = htmlentities($_POST['username'], ENT_QUOTES,
'UTF-8');
        }
    }

?>



<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"
"http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<link rel="stylesheet" type="text/css" href="css/style.css" />
<title>Log In - Family Trail</title>
<link rel="apple-touch-icon" sizes="180x180" href="/images/favicon/apple-touch-
icon.png">
<link rel="icon" type="image/png" sizes="32x32" href="/images/favicon/favicon-
32x32.png">
<link rel="icon" type="image/png" sizes="16x16" href="/images/favicon/favicon-
16x16.png">
<link rel="manifest" href="/images/favicon/manifest.json">
<link rel="shortcut icon" href="/images/favicon/favicon.ico">
<meta name="msapplication-config" content="/images/favicon/browserconfig.xml">
<meta name="theme-color" content="#ffffff">
</head>

<body>
    <div id="page">
```

```
        <div id="header">
           <h1>Family Trail</h1>
            <ul>
                      <li><a href="index.php">Home</a></li>
                   <li><a href="tracking.php">Live Map</a></li>
                <li><a href="history.php"> History</a></li>
                <li><a href="register.php">Register</a></li>
                <li><a href="login.php">Login</a></li>
          </ul>
      </div>

        <div id="main">

           <div class="main_top">
               <h1>Your #1 Family Safety Solution - for your peace of mind!<span
class="links" style="font-weight: bold; font-size: 24px; text-align:
center;"></span></h1>
            </div>

               <div class="main_body">

            <h1><p>Login</p></h1>
            <form action="login.php" method="post">
               Username:<br />
               <input type="text" name="username" value="<?php echo
$submitted_username; ?>" />
               <br /><br />
               Password:<br />
               <input type="password" name="password" value="" />
               <br /><br />
               <input type="submit" value="Login" />
            </form>
            </div>

               <div class="main_bottom"></div>

        </div>



        <div id="footer">
        <p>
        <a href="http://www.aszx.net">web development</a> by <a
href="http://www.bryantsmith.com">bryant smith</a>
        </p>
        </div>

        </div>
</body>
</html>
```

Logout.php

```
<?php
   session_start();
   unset($_SESSION["name"]);  // where $_SESSION["nome"] is your own variable. if
you do not have one use only this as follow **session_unset();**
   unset($_SESSION['user']);
   session_destroy();
```

```
    header("Location: index.php");
?>
```

Register.php

```
<?php

    // First we execute our common code to connection to the database and start the
session
    require("common.php");

    // This if statement checks to determine whether the registration form has been
submitted
    // If it has, then the registration code is run, otherwise the form is
displayed
    if(!empty($_POST))
    {
        // Ensure that the user has entered a non-empty username
        if(empty($_POST['username']))
        {
            // Note that die() is generally a terrible way of handling user errors
            // like this.  It is much better to display the error with the form
            // and allow the user to correct their mistake.  However, that is an
            // exercise for you to implement yourself.
            die("Please enter a username.");
        }

        // Ensure that the user has entered a non-empty password
        if(empty($_POST['password']))
        {
            die("Please enter a password.");
        }

        // Make sure the user entered a valid E-Mail address
        // filter_var is a useful PHP function for validating form input, see:
        // http://us.php.net/manual/en/function.filter-var.php
        // http://us.php.net/manual/en/filter.filters.php
        if(!filter_var($_POST['email'], FILTER_VALIDATE_EMAIL))
        {
            die("Invalid E-Mail Address");
        }

        // We will use this SQL query to see whether the username entered by the
        // user is already in use.  A SELECT query is used to retrieve data from
the database.
        // :username is a special token, we will substitute a real value in its
place when
        // we execute the query.
        $query = "
                SELECT
                    1
            FROM users
            WHERE
                username = :username
        ";

        // This contains the definitions for any special tokens that we place in
        // our SQL query.  In this case, we are defining a value for the token
        // :username.  It is possible to insert $_POST['username'] directly into
        // your $query string; however doing so is very insecure and opens your
        // code up to SQL injection exploits.  Using tokens prevents this.
        // For more information on SQL injections, see Wikipedia:
```

```
        // http://en.wikipedia.org/wiki/SQL_Injection
        $query_params = array(
            ':username' => $_POST['username']
        );

        try
        {
            // These two statements run the query against your database table.
            $stmt = $db->prepare($query);
            $result = $stmt->execute($query_params);
        }
        catch(PDOException $ex)
        {
            // Note: On a production website, you should not output $ex-
>getMessage().
            // It may provide an attacker with helpful information about your code.
            die("Failed to run query: " . $ex->getMessage());
        }

        // The fetch() method returns an array representing the "next" row from
        // the selected results, or false if there are no more rows to fetch.
        $row = $stmt->fetch();

        // If a row was returned, then we know a matching username was found in
        // the database already and we should not allow the user to continue.
        if($row)
        {
            die("This username is already in use");
        }

        // Now we perform the same type of check for the email address, in order
        // to ensure that it is unique.
        $query = "
            SELECT
                1
            FROM users
            WHERE
                email = :email
        ";

        $query_params = array(
            ':email' => $_POST['email']
        );

        try
        {
            $stmt = $db->prepare($query);
            $result = $stmt->execute($query_params);
        }
        catch(PDOException $ex)
        {
            die("Failed to run query: " . $ex->getMessage());
        }

        $row = $stmt->fetch();

        if($row)
        {
            die("This email address is already registered");
        }

        // An INSERT query is used to add new rows to a database table.
        // Again, we are using special tokens (technically called parameters) to
        // protect against SQL injection attacks.
        $query = "
```

```
            INSERT INTO users (
                username,
                password,
                salt,
                email,
            mobile
            ) VALUES (
                :username,
                :password,
                :salt,
                :email,
            :mobile
            )
        ";

        // A salt is randomly generated here to protect again brute force attacks
        // and rainbow table attacks.  The following statement generates a hex
        // representation of an 8 byte salt.  Representing this in hex provides
        // no additional security, but makes it easier for humans to read.
        // For more information:
        // http://en.wikipedia.org/wiki/Salt_%28cryptography%29
        // http://en.wikipedia.org/wiki/Brute-force_attack
        // http://en.wikipedia.org/wiki/Rainbow_table
        $salt = dechex(mt_rand(0, 2147483647)) . dechex(mt_rand(0, 2147483647));

        // This hashes the password with the salt so that it can be stored securely
        // in your database.  The output of this next statement is a 64 byte hex
        // string representing the 32 byte sha256 hash of the password.  The
original
        // password cannot be recovered from the hash.  For more information:
        // http://en.wikipedia.org/wiki/Cryptographic_hash_function
        $password = hash('sha256', $_POST['password'] . $salt);

        // Next we hash the hash value 65536 more times.  The purpose of this is to
        // protect against brute force attacks.  Now an attacker must compute the
hash 65537
        // times for each guess they make against a password, whereas if the
password
        // were hashed only once the attacker would have been able to make 65537
different
        // guesses in the same amount of time instead of only one.
        for($round = 0; $round < 65536; $round++)
        {
            $password = hash('sha256', $password . $salt);
        }

        // Here we prepare our tokens for insertion into the SQL query.  We do not
        // store the original password; only the hashed version of it.  We do store
        // the salt (in its plaintext form; this is not a security risk).
        $query_params = array(
            ':username' => $_POST['username'],
            ':password' => $password,
            ':salt' => $salt,
            ':email' => $_POST['email']
         ':mobile' => $_POST['mobile']
        );

        try
        {
            // Execute the query to create the user
            $stmt = $db->prepare($query);
            $result = $stmt->execute($query_params);
        }
        catch(PDOException $ex)
        {
            // Note: On a production website, you should not output $ex-
>getMessage().
            // It may provide an attacker with helpful information about your code.
            die("Failed to run query: " . $ex->getMessage());
        }

        // This redirects the user back to the login page after they register
        header("Location: login.php");

        // Calling die or exit after performing a redirect using the header
function
        // is critical.  The rest of your PHP script will continue to execute and
        // will be sent to the user if you do not die or exit.
        die("Redirecting to login.php");
    }

?>
 <!--
Design by Bryant Smith
http://www.bryantsmith.com
http://www.aszx.net
email: templates [-at-] bryantsmith [-dot-] com
Released under Creative Commons Attribution 2.5 Generic.  In other words, do with
it what you please; but please leave the link if you'd be so kind :)

Name       : The Slant
Version    : 1.0
Released   : 2009-07-25
-->

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"
"http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<link rel="stylesheet" type="text/css" href="css/style.css" />
<title>Family Trail</title>
</head>

<body>
    <div id="page">

        <div id="header">
           <h1>Family Trail</h1>
            <ul>
                      <li><a href="index.php">Home</a></li>
                   <li><a href="tracking.php">Live Map</a></li>
                <li><a href="history.php"> History</a></li>
                <li><a href="register.php">Register</a></li>
                <li><a href="login.php">Login</a></li>
          </ul>
      </div>

        <div id="main">

           <div class="main_top">
               <h1>Your #1 Family Safety Solution - for your peace of mind!<span
class="links" style="font-weight: bold; font-size: 24px; text-align:
center;"></span></h1>
            </div>

               <div class="main_body">

         <h1><p>Register</p></h1>
         <form action="register.php" method="post">
```

```
            Username:<br />
            <input type="text" name="username" value="" />
            <br /><br />
            E-Mail:<br />
            <input type="text" name="email" value="" />
            <br /><br />
            Mobile Number: (in format 353851234567)<br />
            <input type="number" name="mobile" value="" />
            <br /><br />
            Password:<br />
            <input type="password" name="password" value="" />
            <br /><br />
            <input type="submit" value="Register" />
         </form>

            </div>
               <div class="main_bottom"></div>

        </div>



        <div id="footer">
        <p>
        <a href="http://www.aszx.net">web development</a> by <a
href="http://www.bryantsmith.com">bryant smith</a>
        </p>
        </div>

        </div>
</body>
</html>
```

trackercoordUpdate.php

```
<?php

    $host = "iwillinmy.com";
    $userDname = "iwillinm_gpsuser";
    $Dpassword = "c7BcKQF,X-!X";
    $db_name = "iwillinm_gps";
    $table_name = "coordupdate";

    $conn5 = mysqli_connect($host, $userDname, $Dpassword, $db_name);


    $mylatN = $_POST["mylat"];
    $mylongN = $_POST["mylo"];

    $query5 = "INSERT INTO $table_name(latitude,longitude) VALUES
('$mylatN','$mylongN')";

    $result5= mysqli_query($conn5, $query5) or die();

    if(!$result4){
        echo "trackers location added";
    }else{
        echo "Not added trackers location";
    }
```

SendCoord.php

```php
<?php

    $host = "iwillinmy.com";
    $userDname = "iwillinm_gpsuser";
    $Dpassword = "c7BcKQF,X-!X";
    $db_name = "iwillinm_gps";
    $table_name = "coordinates";

    $conn4 = mysqli_connect($host, $userDname, $Dpassword, $db_name);
    // mysqli_select_db($conn,$db_name);

    $query4 = "SELECT * FROM $table_name ORDER BY locno DESC";
    $result4= mysqli_query($conn4, $query4) or die();

    $res = mysqli_fetch_array($result4);

    $resultA = array();

    array_push($resultA,array(
            "latitude"=>$res['latitude'],
            "longitude"=>$res['longitude'],
        )
    );

    echo json_encode(array("result"=>$resultA));

    while($row3 = mysqli_fetch_array($result4)){

        $lat = $row3['latitude'];
        $long = $row3['longitude'];
       // echo $lat . ': ' . $long . '<br/>';
    }
```

Livemap.php

```php
<?php

    session_start();
    if (isset($_SESSION['user'])) {
      $username = $_SESSION['username'];
    } else {
      header("Location: login.php");
    }

   require_once './dbinfo.inc.php';

    $lat;
    $long;
    $response = array();
   $conn2 = mysqli_connect($host, $userDname, $Dpassword, $db_name);

    $query2 = "SELECT * FROM $table_name";
    $result2= mysqli_query($conn2, $query2) or
die("FAILED!!!".mysqli_error($conn2));

    while($row = mysqli_fetch_array($result2)){
        $lat = $row['latitude'];
        $long = $row['longitude'];
      //  echo $lat . ': ' . $long . '<br/>';
    }

?>
   <!--

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"
"http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<link rel="stylesheet" type="text/css" href="css/style.css" />
<title>Live - Family Trail</title>
<link rel="apple-touch-icon" sizes="180x180" href="/images/favicon/apple-touch-
icon.png">
<link rel="icon" type="image/png" sizes="32x32" href="/images/favicon/favicon-
32x32.png">
<link rel="icon" type="image/png" sizes="16x16" href="/images/favicon/favicon-
16x16.png">
<link rel="manifest" href="/images/favicon/manifest.json">
<link rel="shortcut icon" href="/images/favicon/favicon.ico">
<meta name="msapplication-config" content="/images/favicon/browserconfig.xml">
<meta name="theme-color" content="#ffffff">
</head>

<body>
    <div id="page">

        <div id="header">
           <h1>Family Trail</h1>
            <ul>
                      <li><a href="home.php">Home</a></li>
                   <li><a href="livemap.php">Live Map</a></li>
                <li><a href="history.php"> History</a></li>
                <li><a href="help.php">Get Help</a></li>
                <li><a href="logout.php">Logout</a></li>
          </ul>
      </div>

        <div id="main">

           <div class="main_top">
               <h1>Your #1 Family Safety Solution - for your peace of mind!<span
class="links" style="font-weight: bold; font-size: 24px; text-align:
center;"></span></h1>
            </div>

               <div class="main_body">
                <p>Welcome <?php echo $_SESSION['user']; ?>,</p>
                <p>Below is your childs location on &lt;php.get.last.date&gt;</p>
                        <div id="map"
style="width:915px;height:550px;background:white"></div>
        <link rel="stylesheet" href="style">
                <p>
                <div id="right-panel">
            <p>Total Distance: <span id="total"></span></p>
        </div>

        <script>
            function myMap() {

                var directionsService = new google.maps.DirectionsService;
                var mylocat = new google.maps.LatLng();

                var coordla = '<?php echo $lat ;?>';
                var comma = ','
                var coordlo = '<?php echo $long ;?>';
                var coords = coordla.concat(comma,coordlo);

                var b=coords.split(",");
                document.write(b);
                var coordsN=new google.maps.LatLng(parseFloat(b[0]),
parseFloat(b[1]));
                 //document.write(coords);
                //document.write(coordsN);

                Chcoords = new google.maps.LatLng(parseFloat(b[0]),
parseFloat(b[1]));

                var mapOptions = {
                    center: new google.maps.LatLng(parseFloat(b[0]),
parseFloat(b[1])),
                    zoom: 15,
                    mapTypeId: google.maps.MapTypeId.HYBRID
                };
                var map = new google.maps.Map(document.getElementById("map"),
mapOptions);



                document.write(mylocat+"LOC HERE!!");

                //displayRoute(Chcoords, mylocat, directionsService,
                  //  directionsDisplay);


                var marker = new google.maps.Marker({
                    position: (coordsN),
                    title: 'Child here!!!'

                });

                var transitLayer = new google.maps.TransitLayer();
                transitLayer.setMap(map);

                marker.setMap(map);

                //document.write(coordsN);


        </script>

        <script
src="https://maps.googleapis.com/maps/api/js?key=AIzaSyBnRki2JB3obMHyVbkxW0Hzr74hmP
GAz3I&callback=myMap"async defer></script>

        <script type="text/javascript" src="jquery.js"> </script>
        <script type="text/javascript">
            $(document).ready(function () {
                setInterval(function () {
                    $('#map').load('tracking.php')
                }, 97000);

            });
        </script> </p>
               </div>

               <div class="main_bottom"></div>

        </div>



        <div id="footer">
        <p>
        <a href="http://www.aszx.net">web development</a> by <a
href="http://www.bryantsmith.com">bryant smith</a>
        </p>
        </div>

        </div>
</body>
</html>
```

History.php

```
<?php
   session_start();
   if (isset($_SESSION['user'])) {
   $username = $_SESSION['user'];
   } else {
   header("Location: login.php");
   }

   require_once './dbinfo.inc.php';

   if (!isset($install) or $install != '1') {
   $connection = mysql_connect($host, $userDname, $Dpassword) or die ('Unable to
connect to MySQL server.<br ><br >Please make sure your MySQL login details are
correct.');
   $db = mysql_select_db($db_name, $connection) or die ('request "Unable to select
database."');
};
?>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"
"http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<link rel="stylesheet" type="text/css" href="css/style.css" />
<link href="css/calendar.css" rel="stylesheet" type="text/css" />

<title>History - Family Trail</title>
<link rel="apple-touch-icon" sizes="180x180" href="/images/favicon/apple-touch-
icon.png">
<link rel="icon" type="image/png" sizes="32x32" href="/images/favicon/favicon-
32x32.png">
<link rel="icon" type="image/png" sizes="16x16" href="/images/favicon/favicon-
16x16.png">
<link rel="manifest" href="/images/favicon/manifest.json">
<link rel="shortcut icon" href="/images/favicon/favicon.ico">
<meta name="msapplication-config" content="/images/favicon/browserconfig.xml">
<meta name="theme-color" content="#ffffff">
</head>
```

```
<body>
    <div id="page">

        <div id="header">
           <h1>Family Trail</h1>
            <ul>
                      <li><a href="home.php">Home</a></li>
                   <li><a href="livemap.php">Live Map</a></li>
                <li><a href="history.php"> History</a></li>
                <li><a href="help.php">Get Help</a></li>
                <li><a href="logout.php">Logout</a></li>
          </ul>
      </div>

        <div id="main">

           <div class="main_top">
               <h1>Your #1 Family Safety Solution - for your peace of mind!<span
class="links" style="font-weight: bold; font-size: 24px; text-align:
center;"></span></h1>
            </div>

               <div class="main_body">
                <p>Welcome <?php echo $_SESSION['user']; ?>,</p>
                <p>Here will be a calendar lets hope! =)</p>
                <p><div id="Calendar"> </div>
            <div id="Events"> </div>
            <script language="javascript" src="calendar.js"></script></p>
            </div>

               <div class="main_bottom"></div>

        </div>



        <div id="footer">
        <p>
        <a href="http://www.aszx.net">web development</a> by <a
href="http://www.bryantsmith.com">bryant smith</a>
        </p>
        </div>

        </div>
</body>
</html>
```

Safezone.php

```
<?php
    session_start();
    if (isset($_SESSION['user'])) {
      $user = $_SESSION['user'];
    } else {
      header("Location: login.php");
    }

   /*error_reporting(E_ALL);
   ini_set('display_errors', 1);
   var_dump($_POST);*/

   require_once './dbinfo.inc.php';
   // Create connection
   $conn = new mysqli($host, $userDname, $Dpassword, $db_name);
   // Check connection
   if ($conn->connect_error) {
      die("Connection failed: " . $conn->connect_error);
   }

   $sql = "SELECT sznw, szne, szse, szsw FROM users WHERE username = '$user'";
   //$sql = "SELECT sznw, szne, szse, szsw FROM users WHERE username = 'Mark'";

   $result = $conn->query($sql);
   $c1 = 0; $c2=0; $c3 = 0; $c4=0;

   if ($result->num_rows > 0) {
      // output data of each row
      while($row = $result->fetch_assoc()) {
      // echo "szsw: " . $row["szsw"]. " - szse: " . $row["szse"]. " - sznw: " .
$row["sznw"]. " - szne: " . $row["szne"]. "<br>";
         $c1= $row["szsw"]; $c2= $row["szse"]; $c3= $row["sznw"]; $c4=
$row["szne"];
      }
   } else {
      echo "0 results";
   }
   $conn->close();

?>



<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"
"http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta name="viewport" http-equiv="Content-Type" content="text/html; charset=utf-8,
initial-scale=1.0, user-scalable=no" />
<link rel="stylesheet" type="text/css" href="css/style.css" />
<title>Select Safe-Zone - Family Trail</title>
<link rel="apple-touch-icon" sizes="180x180" href="/images/favicon/apple-touch-
icon.png">
<link rel="icon" type="image/png" sizes="32x32" href="/images/favicon/favicon-
32x32.png">
<link rel="icon" type="image/png" sizes="16x16" href="/images/favicon/favicon-
16x16.png">
<link rel="manifest" href="/images/favicon/manifest.json">
<link rel="shortcut icon" href="/images/favicon/favicon.ico">
<meta name="msapplication-config" content="/images/favicon/browserconfig.xml">
<meta name="theme-color" content="#ffffff">
</head>

<body>
    <div id="page">

        <div id="header">
           <h1>Family Trail</h1>
            <ul>
                      <li><a href="home.php">Home</a></li>
                   <li><a href="livemap.php">Live Map</a></li>
                <li><a href="history.php"> History</a></li>
                <li><a href="help.php">Get Help</a></li>
                <li><a href="logout.php">Logout</a></li>
          </ul>
      </div>

        <div id="main">
```

```
           <div class="main_top">
               <h1>Your #1 Family Safety Solution - for your peace of mind!<span
class="links" style="font-weight: bold; font-size: 24px; text-align:
center;"></span></h1>
            </div>

               <div class="main_body">
                <p>Welcome <?php echo $_SESSION['user']; ?>,</p>
                <p>Here you can select the safezone. This means if your child
leaves this area you will be notified and an automatic action can be taken, such as
ringing your phone, setting off an alarm on the childs phone or anything else we
can think of! ;p &lt;EDIT&gt;</p>
                <p>
                <p>What happens if trackee(?? better word, so isnt only kids?)
leaves the safe zone? &lt;Email, Alarm, Phonecall, Alert Trackee&gt;<br />
                  &lt;options menu here underneath map&gt;

               <div id="map" style="width:915px;height:400px;"></div>
              <p>
       <script>
         var rectangle;

         var map;
         var infoWindow;
         var szne;
         var sznw;
         var szsw;
         var szse;

         // This example adds a user-editable rectangle to the map.
         function initMap() {
              map = new google.maps.Map(document.getElementById('map'), {
              center: {lat: 53.4033, lng: -6.3771},
              streetViewControl: false,
              zoom: 9
            });

            var bounds = {
               north: <?php echo $c4 ?>,
               south: <?php echo $c1 ?>,
              east: <?php echo $c3 ?>,
              west: <?php echo $c2 ?>,
            };

            // Define a rectangle and set its editable property to true.
            rectangle = new google.maps.Rectangle({
               bounds: bounds,
               editable: true,
               draggable: true
            });
            rectangle.setMap(map);

            // Add an event listener on the rectangle.
            rectangle.addListener('bounds_changed', showNewRect);

            // Define an info window on the map.
            infoWindow = new google.maps.InfoWindow();
         }
         // Show the new coordinates for the rectangle in an info window.

         /** @this {google.maps.Rectangle} */
         function showNewRect(event) {
            var ne = rectangle.getBounds().getNorthEast();
            var sw = rectangle.getBounds().getSouthWest();
```

```
             var contentString = '<b>Rectangle moved.</b><br>' +
            'New north-east corner: ' + ne.lat() + ', ' + ne.lng() + '<br>' +
            'New south-west corner: ' + sw.lat() + ', ' + sw.lng();

            // Set the info window's content and position.
            infoWindow.setContent(contentString);
            infoWindow.setPosition(ne);

            infoWindow.open(map);


            // Set variables to prepare to send to PHP and update DB
            szne = ne.lat();
            sznw = ne.lng();
            szsw = sw.lat();
            szse = sw.lng();
            //update txt fields
            updatetext();
         }


         function updatetext() {
            document.getElementById("postszne").value = szne;
            document.getElementById("postsznw").value = sznw;
            document.getElementById("postszsw").value = szsw;
            document.getElementById("postszse").value = szse;
         }

         var butn1 = document.getElementById("txtUpdate");
         butn1.addEventListener("click", updatetext);
         var butn2 = document.getElementById("txtDocwrite");
         butn2.addEventListener("click", docwrite);
         </script>

         <script async defer
src="https://maps.googleapis.com/maps/api/js?key=AIzaSyBnRki2JB3obMHyVbkxW0Hzr74hmP
GAz3I&callback=initMap">
         </script>

          </p>
         <p>
         <form action='options.php' method='post'>
          <input type="hidden" id="postszne" name="postszne">
          <input type="hidden" id="postsznw" name="postsznw">
          <input type="hidden" id="postszsw" name="postszsw">
          <input type="hidden" id="postszse" name="postszse">
          Click here once you have chosen the safe zone area :: <input
type="submit" name="useroptions" value="Update Zone">
         Or click here to return it to default zone :: <input type="submit"
name="useroptions" value="Reset Zone"></form>

            <!--<button>Click Here to update your safe zone </button></p>-->
         </div>
               <div class="main_bottom"></div>
        </div>



        <div id="footer">
        <p>
        <a href="http://www.aszx.net">web development</a> by <a
href="http://www.bryantsmith.com">bryant smith</a>
        </p>
        </div
```

Appendix C - Bibliography

Peter H. Dana . (1999). Colorado.edu [online]. Available from
http://www.colorado.edu/geography/gcraft/notes/gps/gps_f.html [accessed 20 April 2017].
Garmin. (2017). Garmin [online]. Available from https://www8.garmin.com/aboutGPS/
[accessed 3 march 2017].

Developers. Android. (2013) Official IDE for Android. Andriod Studio [online]. Available from
https://developer.android.com/studio/index.html [accessed 3 March 2017].

Developers. Android. (2017). developers.andriod [online]. Available from
https://developer.android.com/training/location/index.html [accessed 3 March 2017].

Developers. Google. (2017) Maps Android API. [online]. Available from
https://developers.google.com/maps/documentation/android-api/map [accessed 3 March 2017].

Google Developers Console. (2017) Console for google developers. [online]. Available from
https://console.developers.google.com [accessed 3 March 2017].

Family Safety Production. (2017) GPS Phone Tracker. Google Play Store [online]. Available from
https://play.google.com/store/apps/details?id=com.fsp.android.c&hl=en [accessed 10 April 2017].

Nick Fox. (2015) Gps Tracker. Google Play Store[online]. Available from
https://play.google.com/store/apps/details?id=com.websmithing.gpstracker [accessed 10 April 2017].

Greenalp Communication. (2016) Real-Time GPS Tracker. Google Play Store [online]. Available from
https://play.google.com/store/apps/details?id=com.greenalp.RealtimeTracker [accessed 10 April 2017].

Josh Ruesch. (2013) volley-vs-retrofit. Instructure github [online]. Available from
http://instructure.github.io/blog/2013/12/09/volley-vs-retrofit/ [accessed 10 April 2017].

Sonia Mastros. (2015) 4-reasons-why-gps-vehicle-tracking-software-has-become-essential. BusBoss [online]. Available from https://www.busboss.com/blog/4-reasons-why-gps-vehicle-tracking-software-has-become-essential [accessed 29 March 2017]

Smartsheet community. (2016)
What's the Difference? Agile vs Scrum vs Waterfall vs Kanban. smartsheet [online]. Available from https://www.smartsheet.com/agile-vs-scrum-vs-waterfall-vs-kanban [accessed date].

Developers. Android. (2013) Transmitting Network Data Using Volley. Developers. Android [online]. Available from https://developer.android.com/training/volley/index.html [accessed 5 April 2017].

Developers android. (2017) Changing Location Settings. developer.android[online]. Available from https://developer.android.com/training/location/change-location-settings.html [accessed May 4 2017].
Bryant Smith. (2017) Website CSS Styling. [Online] http://www.bryantsmith.com/