# **Project:** Forecast Destination

Location Tracking System, Notice Unpredictable Behavior

## Project Goal

This project aims to solve the problem of component monitoring by providing a software platform. An efficient object tracking system will be implemented for monitoring a targeted object from any location at any time with the help of the Global Positioning System (GPS) which will enable users to locate their targeted object with ease and in a convenient manner. Whenever the targeted object deviates from its optimized path, a notification popup is generated on the screen of the user.

## Unpredictable Behavior:

- The targeted object is out of the safe zone.
- Extra time lag.

## Solution Approach:

1. Get the source and target location on the map.
2. Find out the best or allowed paths from source to destination (the user may have some preference for this).
3. For any selected path, get some number of intermediate points in the route.
4. Draw a polygon by considering the points and some margin of error (e.g., 50 meters above and below the given point).
5. Get user location from the device in real time.
6. Find out if the current location is outside of the polygon corresponding to any allowed path.
7. If it is outside, calculate the time since it went out (to have some tolerance for error).
8. If the user is outside of the selected paths for a certain amount of time, consider this as undesired behavior and notify accordingly.

## Implemented Procedure:

First, we have set the Google Map API key.

We have asked the user to select the following journey details.

- Source and Destination address by searching in the map application.
- A Geofencing threshold, the radius of the safe zone in meters around the source to the

destination path. E.g., if the threshold is 100, then a radius of 100 meter will be set as the safe zone around the path point of the object where the pin is placed on the map.

We have worked on getting geo-coordinates that would contain GPS points (latitude, longitude, altitude) with timestamps of the targeted object and for this, we have made use of the GPS module to capture location, speed and time of last received data in accordance. That means, in real time, we would get to see the location of the targeted object.

The concept of Geofencing has been used in this task. A geofence is a defining virtual boundary around geographic objects or an area so that every time the targeted object enters or leaves the boundary perimeters, actions or notifications can be triggered [1, 2]. We need to have an area that marks the geofence.

Here, if we will use Java/Android Platform, we can get the accurate intermediate waypoints in n intervals (time/distances) [3, 4]. But, as it is a Python implementation demo, we have to get the points manually. We have used some mathematical calculations for this. such as retrieving the coordinates between those two points (source, destination) at every n meter interval starting at the first coordinate and ending at the end coordinate, generating a merged polyline through the point and creating a polygon around that polyline.

Our data is basically in Datum: World Geodetic System 1984, Ellipsoidal 2D CS and CRS: EPSG:4326. For this, we have projected this data in Cartesian 2D, unit in meters. After the mathematical calculations, we have transformed the points vice versa [5, 6].

Using the threshold offset, we have plotted the geofence polygon. We have overlaid the position coordinate points of the targeted object into the geofence polygon area to check whether the points have been inside the polygon or not [7]. Finally, we have animated the points to visualize with track movements. If the points had been outside of the polygon, red colored points should be alarmed to the user.

We have taken some geolocation with some outlier points as a Geopandas Dataframe for testing purposes and checked if the system is working or not.

## Limitations:

The user has to fill the input form manually, not on the map. This Python implemented system can not obtain the intermediate waypoints accurately. This can be done by Java/Android Platform. Also this is not a real time application.

## Future Implementation:

We can calculate the movement of the targeted object in meters per second and predict the time difference variable [8]. This can be implemented using the Haversine [9] or Vincenty [10] distance formula.

This part will include the generation of specific reports for the journey to the user. This report will show basic journey details and google maps representing position coordinates of the targeted object based on GPS location. If the object travels in a vehicle, it also reports a graph showing the speed of the vehicle at every instance of time. It also notifies the alarm if the targeted object position shows any unpredictable behavior.

## Challenges:

Position coordinates are acquired based on the GPS signal. Various uncontrollable and unpredictable factors (e.g., atmospheric disturbances, failure of the GPS antenna, electromagnetic interference, weather change, GPS signal attack, or solar activity) may cause GPS receivers to lose signal occasionally. Depending on these, there may be a time lag in sending notifications.

## References:


[1]. https://en.wikipedia.org/wiki/Geo-fence.
[2]. https://medium.com/@Synerzip/basics-of-geofence-implementation-b467c31f46df.
[3]. Find All the points in a path in Android.
[4]. Android Path to Array.
[5]. source of EPSG: 4326.
[6]. source of EPSG: 3857.
[7]. https://towardsdatascience.com/the-art-of-geofencing-in-python-e6cc237e172d.
[8].https://towardsdatascience.com/how-tracking-apps-analyse-your-gps-data-a-hands-on-tutorial-in-python-756d4db6715d
[9]. https://en.wikipedia.org/wiki/Haversine_formula.
[10]. https://en.wikipedia.org/wiki/Vincenty%27s_formulae.