

Project Development Phase

SPRINT-1

Date	10 November 2022
Team ID	PNT2022TMID00975
Project Name	Project - IOT Based Safety Gadget for Child Safety Monitoring and Notification

Live Location Tracking:

GPS is installed on gadget to track the current location of the child. It can be viewed using an android app and via SMS alert notifications can be sent. Co-ordinates of the live location is stored in the cloud.

Panic Alert Systems:

Panic Alert System is used to notify the parent or guardian with emergency messages. Prestored messages are also added for a more time efficient service

Stay Connected Feature:

Live tracking can be enabled for continuous monitoring of the child. Messages and voice mails can be exchanged to stay in contact with each other.

Health Monitoring System:

Health monitoring system is implemented using heart beat sensor, temperature sensor which is updated to the cloud and also can be monitored via app. The current value of sensors can be obtained using SMS request sent to gadget from parent phone.

Gadget Plugged or Unplugged Monitoring:

Gadget plug or unplugged is monitored using contact switch installed on smart gadget, as soon as the device is unplugged, an alert is provided to parent phone via SMS and it is also updated to cloud for app monitoring.

GEOFENCING CODE:

Basic Example Code:

```
import time
def
    stopwatch(secon
    ds,d,lspoint): start
    = time.time()
    time.clock()
    elapsed
    = 0 flag
    = False
    num = 0
    while elapsed < seconds:
        elapsed          =
        time.time() - start
        print  "%02d"  %
        elapsed

        if elapsed > d[num] and elapsed < d[num+1]
            and flag == False: x = lspoint[num][0]

y    =    lspoint[num][1]
createpoint(x,y) flag=True
```

```

        print "Shot Taken"
    print
    point_in_poly(x,y,polygon)
    if elapsed > d[num+1]:

        print  "Shot
        Taken"  flag
        == False

        num = num+1
        x      =
        lspoint[num]
        [0]    y    =
        lspoint[num]
        [1]
        createpoint(x
        ,y)
        print
        point_in_poly(x,y,polygon)
        time.sleep(1)

```

```

def createpoint(x,y):

```

```

crs="point?crs=epsg:27700&field=id
:integer"

layer =QgsVectorLayer(crs, 'points' ,
"memory")

pr =layer.dataProvider()
pt =QgsFeature()
point1=QgsPoint
(x,y)

pt.setGeometry(QgsGeometry.fromPoint
(point1))pr.addFeatures([pt])

# update extent of the
layer
layer.updateExtents()

# add the second point

    pt= QgsFeature()
    QgsMapLayerRegistry
instance().add
MapLayers([layer])
)

def point_in_poly(x,y,poly):

```

```
n =len(poly)inside=False
```

```
p1x,p1y=poly[0]
```

```
for i in range(n+1):
```

```
    p2x,p2y =poly[i%n]
```

```
    if y>min(p1y,p2y):
```

```
        if y <= max(p1y,p2y):
```

```
            if x<=max(p1x,p2x):
```

```
                if p1y !=p2y:
```

```
                    xints = (y-p1y)*(p2x-p1x)/(p2y-p1y)+p1x
```

```
            if p1x == p2x or x<=xints:
```

```
                inside = not inside
```

```
                p1x,p1y =p2x,p2y
```

```
return inside
```

```
##### define the polygon
```

```
polygon =
```

```
[(512882.78819722467,120811.83924772343),(512960.84437170526,  
120809.7007223952),(512960.84437170526,120809.7007223952),  
(512959.77510904113,120754.09906386107),(512882.78819722467,  
120756.2375891893)]
```

```
###set how long the script will run (70 seconds will getyou  
in and out of geofence)time_seconds = 70
```

```
#### first coordinate=512915
```

```
y = 120728
```

```
#### time intervals, 10 seconds
```

```
between shots / or pointsintervals =
```

```
int(time_seconds / 10)
```

```
lspoint = []
```

```
#### build the list of coordinates to be plotted
```

```
for i inrange(0,intervals+1):
```

```
y1 = y + (i*12.5)
```

```
lspoint.append(x,y1)
```

```
#### to build the blocks of time in intervals, so we know the number  
of intervals (default is 7),
```

```
#### we need a list of time intervals [0,10,20,30 etc] to check  
against the clock this list is d, f is thegap ie 10 seconds, a is starting  
point (0)
```

```
### b is the number of intervals + 1 becuae the code will  
check the the next in the listf = 10
```

```
a = 0
```

```
b = intervals+1
```

```
d = [x * f for x in range(a, b)]
```

```
### Run the stopwatch, or start the program!
```

```
stopwatch(time_seconds,d, lspoint)
```

