Project Development Phase

Date	*10 November 2022*
Team ID	*PNT2022TMID52173*
Project Name	*IoT based safety gadget for child monitoring and notification*
Maximum Marks	*8 Marks*

Sprint 3: Setting Geo-Fence

Geo-fencing combines awareness of the user's current location with awareness of the user's proximity to locations that may be of interest. To mark a location of interest, you specify its latitude and longitude. To adjust the proximity for the location, you add a radius. The latitude, longitude, and radius define a Geo-fence, creating a circular area, or fence, around the location of interest.

You can have multiple active Geo-fences, with a limit of 100 per app, per device user. For each Geo-fence, you can ask Location Services to send you entrance and exit events, or you can specify a duration within the geo-fence area to wait, or dwell, before triggering an event. You can limit the duration of any Geo-fence by specifying an expiration duration in milliseconds. After the Geo-fence expires, Location Services automatically removes it. Here the receiver board covers a range and the range covered by the receiver is referred as Geo-fence.

Receiver Code:

```
 #include <Wire.h>
 #include <LiquidCrystal_I2C.h>
 #include <SPI.h>
 #include <nRF24L01.h>
 #include <RF24.h>

 RF24 radio(9, 10); // CE, CSN

 const byte address[6] = "00001";
 const int ENA = 6;
 const int ENB = 5;

 LiquidCrystal_I2C lcd(0x27,20,4);

 void setup()
 {
   pinMode(8, OUTPUT);
   pinMode(7, OUTPUT);
   pinMode(4, OUTPUT);
   pinMode(3, OUTPUT);
```

```
  pinMode (ENA, OUTPUT);
  pinMode (ENB, OUTPUT);

  Serial.begin(9600);

  radio.begin();
  radio.openReadingPipe(0, address);
  radio.setPALevel(RF24_PA_MIN);
  radio.startListening();

  lcd.init();
  lcd.init();
  lcd.backlight();
}
void loop()
{
  if (radio.available())
  {
    char text[32] = "";
    radio.read(&text, sizeof(text));
    Serial.println(text);
  lcd.setCursor(0,0);
  lcd.print("   CONNECTED    ");

  digitalWrite(8,HIGH);
  digitalWrite(7,LOW);
  digitalWrite(4,HIGH);
  digitalWrite(3,LOW);
  analogWrite(ENA,150);
  analogWrite(ENB,150);
}
else
{
        digitalWrite(8,HIGH);
  digitalWrite(7,LOW);
  digitalWrite(4,HIGH);
  digitalWrite(3,LOW);
  analogWrite(ENA,255);
  analogWrite(ENB,255);

  Serial.println("OUT OF RANGE");
```

```
    lcd.setCursor(0,0);
    lcd.print("  CHECKING THE   ");
    lcd.setCursor(0,1);
    lcd.print("    SERVER      ");
  }
}
```