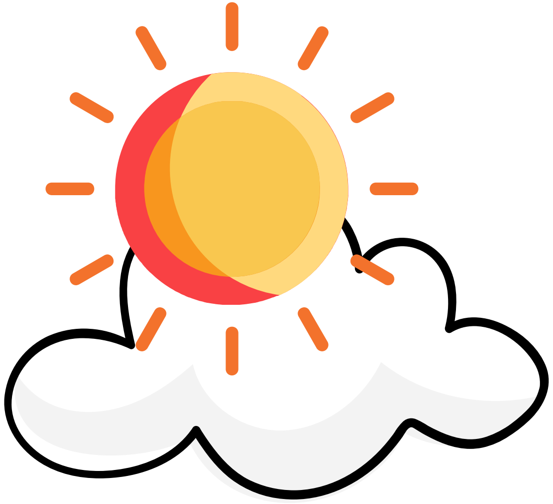
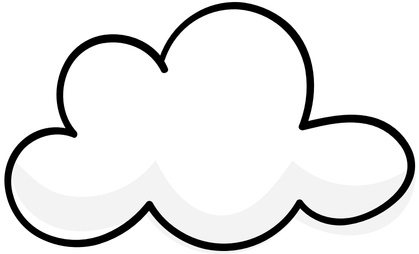
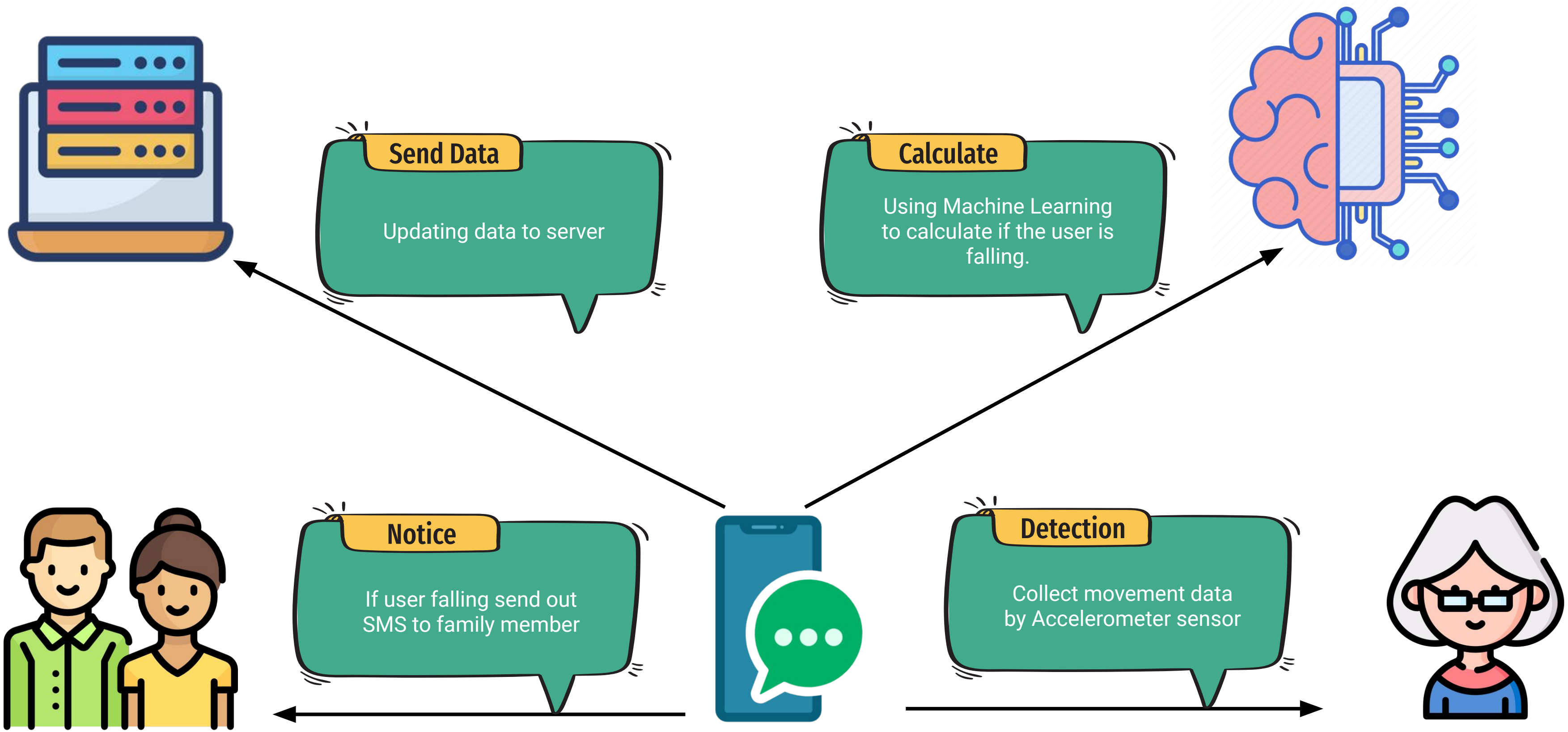


Nursing Home Falling Detection

Avery Niou 2022



Introduction



Send Data

Updating data to server

Calculate

Using Machine Learning to calculate if the user is falling.

Notice

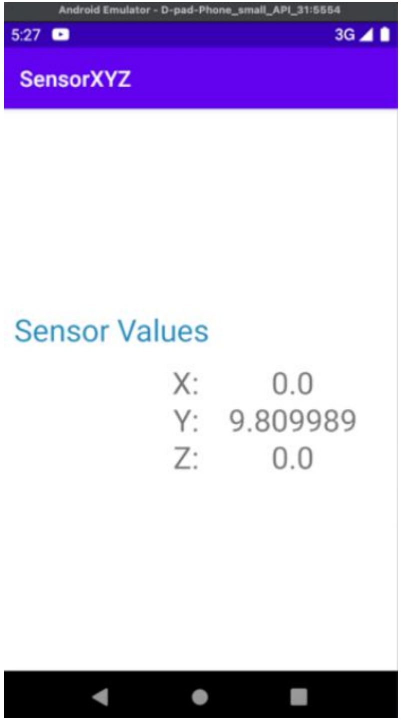
If user falling send out SMS to family member

Detection

Collect movement data by Accelerometer sensor

Get Accelerometer Sensor data

```
private TextView xValue, yValue, zValue;
private SensorManager sensorManager;
void getSensorValue() {
   xValue_a = (TextView) findViewById(R.id.x_value);
   yValue_a = (TextView) findViewById(R.id.y_value);
   zValue_a = (TextView) findViewById(R.id.z_value);
   sensorManager = (SensorManager) getSystemService(Context.SENSOR_SERVICE);
   sensorManager.registerListener(myAccelerometerListener,
           sensorManager.getDefaultSensor(Sensor.TYPE_ACCELEROMETER),
           SensorManager.SENSOR_DELAY_NORMAL);
}
final SensorEventListener myAccelerometerListener = new SensorEventListener(){
   public void onSensorChanged(SensorEvent sensorEvent){
       if(sensorEvent.sensor.getType() == Sensor.TYPE_ACCELEROMETER){
           float X_lateral = sensorEvent.values[0];
           float Y_longitudinal = sensorEvent.values[1];
           float Z_vertical = sensorEvent.values[2];
           xValue_a.setText(String.valueOf(X_lateral));
           yValue_a.setText(String.valueOf(Y_longitudinal));
           zValue_a.setText(String.valueOf(Z_vertical));
           isfalling_KNN(X_lateral,Y_longitudinal,Z_vertical);
       }
   }
   public void onAccuracyChanged(Sensor sensor , int accuracy){
   }
};
```

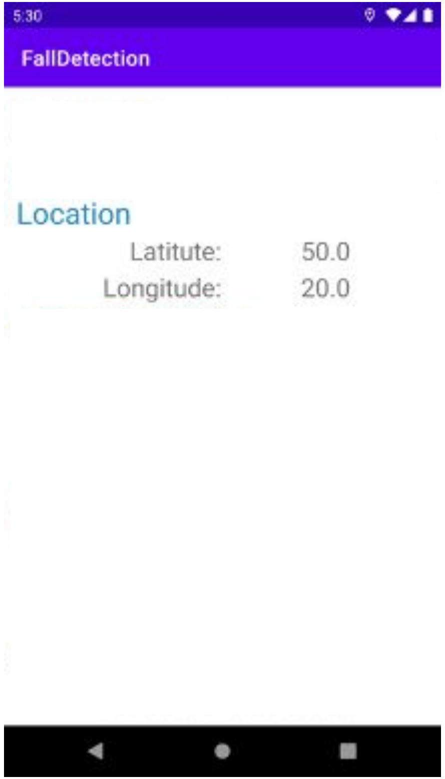


Sensor Values

X: 0.0
Y: 9.809989
Z: 0.0

Get Location

```
private LocationManager locationManager;
private TextView textViewLa, textViewLo;
private String stringLa, stringLo;
void getLocation() {
   textViewLa = (TextView) findViewById(R.id.latitute_textView);
   textViewLo = (TextView) findViewById(R.id.longitude_textView);
   locationManager = (LocationManager) getSystemService(Context.LOCATION_SERVICE);
   if (ContextCompat.checkSelfPermission(MainActivity.this,
Manifest.permission.ACCESS_COARSE_LOCATION) != PackageManager.PERMISSION_GRANTED &&
           ContextCompat.checkSelfPermission(MainActivity.this,
Manifest.permission.ACCESS_FINE_LOCATION) != PackageManager.PERMISSION_GRANTED) {
       ActivityCompat.requestPermissions(MainActivity.this, new
String[]{Manifest.permission.ACCESS_COARSE_LOCATION,
Manifest.permission.ACCESS_FINE_LOCATION}, 1);
   }
   locationManager.requestLocationUpdates(LocationManager.GPS_PROVIDER, 0, 0, new
LocationListener() {
       @Override
       public void onLocationChanged(@NonNull Location location) {
           stringLa = String.valueOf(location.getLatitude());
           stringLo = String.valueOf(location.getLongitude());
           textViewLa.setText(stringLa);
           textViewLo.setText(stringLo);
       }
   });
}
```



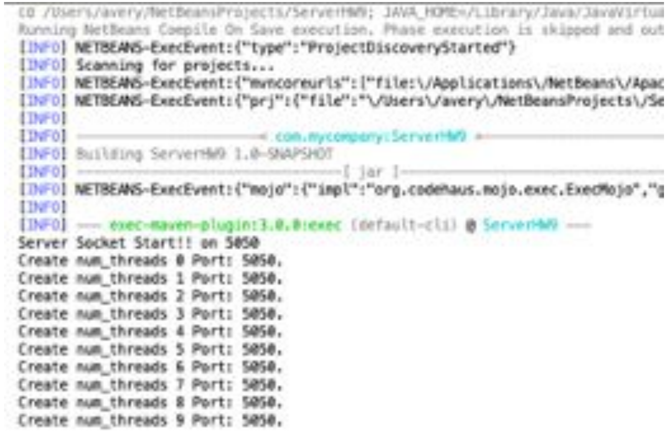
Location

Latitute: 50.0

Longitude: 20.0

Set Server

```
public class MyServer extends Thread{
        ServerSocket theServer;
        static int num_threads = 10;
        public static void main(String[] args) throws IOException {
                try {
                ServerSocket ss = new ServerSocket(5050);
                System.out.println("Server Socket Start!! on 5050");
                for (int i = 0; i < num_threads; i++) {
                                System.out.println("Create num_threads " + i + " Port: 5050.");
                                MyServer myserver = new MyServer(ss);
                                myserver.start();
                }
                } catch (IOException e) { System.err.println(e); }
        }
        public MyServer(ServerSocket ss) { theServer = ss; }
        public void run() {
        while (true) {
        try {
                        Socket connection = theServer.accept();
                        DataOutputStream output = new DataOutputStream(connection.getOutputStream());
                        DataInputStream input = new DataInputStream(connection.getInputStream());
                        System.out.println("Client Connected and Start get I/O!!");
                        System.out.println("==> Input from Client: " + input.readUTF());
                        System.out.println("Output to Client ==> \"Connection successful\"");
                        output.writeUTF( "Connection successful" );
                        output.flush();
                        input.close();
                        connection.close();
        } catch (IOException e) { }
          }
        }
}
```



```
cd /Users/avery/NetBeansProjects/ServerHW9; JAVA_HOME=/Library/Java/JavaVirtua
Running NetBeans Compile On Save execution. Phase execution is skipped and out
[INFO] NETBEANS-ExecEvent:{"type":"ProjectDiscoveryStarted"}
[INFO] Scanning for projects...
[INFO] NETBEANS-ExecEvent:{"mvncoreurls":["file:\/Applications\/NetBeans\/Apac
[INFO] NETBEANS-ExecEvent:{"prj":{"file":"\/Users\/avery\/NetBeansProjects\/Se
[INFO]
[INFO] ------------------------< com.mycompany:ServerHW9 >--------------------
[INFO] Building ServerHW9 1.0-SNAPSHOT
[INFO] --------------------------------[ jar ]---------------------------------
[INFO] NETBEANS-ExecEvent:{"mojo":{"impl":"org.codehaus.mojo.exec.ExecMojo","g
[INFO]
[INFO] --- exec-maven-plugin:3.0.0:exec (default-cli) @ ServerHW9 ---
Server Socket Start!! on 5050
Create num_threads 0 Port: 5050.
Create num_threads 1 Port: 5050.
Create num_threads 2 Port: 5050.
Create num_threads 3 Port: 5050.
Create num_threads 4 Port: 5050.
Create num_threads 5 Port: 5050.
Create num_threads 6 Port: 5050.
Create num_threads 7 Port: 5050.
Create num_threads 8 Port: 5050.
Create num_threads 9 Port: 5050.
```

SQLiteDB to store labeled movement data

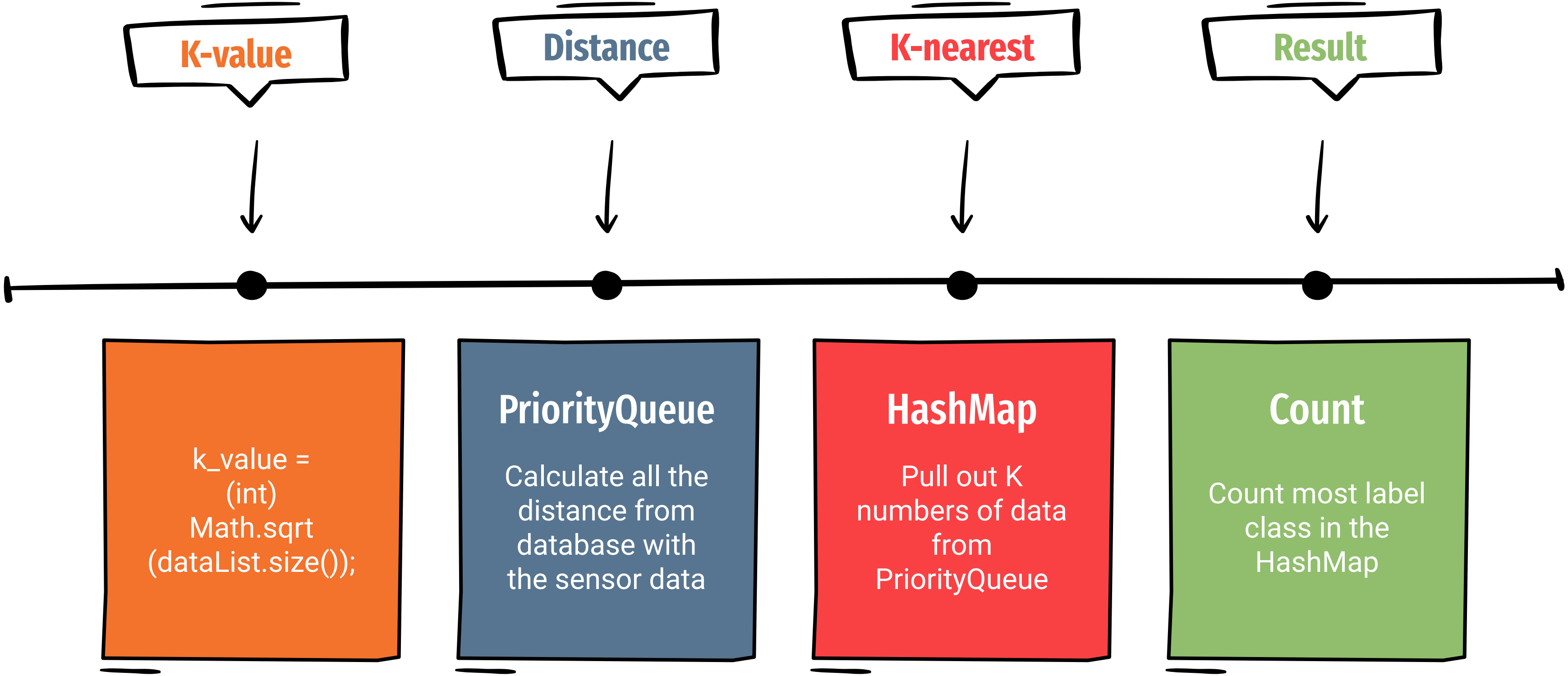
```
public class SQLiteDB extends SQLiteOpenHelper {
   public static final String CREATE_TABLE = "create table " +
MyConstant.TABLE_NAME +
           " (" + MyConstant.COL_ID + " INTEGER, " + MyConstant.COL_X +
" REAL, " + MyConstant.COL_Y +
           " REAL, " + MyConstant.COL_Z + " REAL, " +
MyConstant.COL_CLASS + " varchar(10))";
   private SQLiteDatabase db;
   public SQLiteDB(@Nullable Context context) {
       super(context, MyConstant.DB_NAME, null, 1);
       db = this.getWritableDatabase();
   }
   @Override
   public void onCreate(SQLiteDatabase db) {
       Log.d("DatebaseHelper","Create dataset");
       db.execSQL(CREATE_TABLE);
   }
   @Override
   public void onUpgrade(SQLiteDatabase db, int i, int i1) {
   }
   public long insertRecord(float x, float y, float z, String class_) {
       ContentValues values = new ContentValues();
       values.put(MyConstant.COL_X, x);
       values.put(MyConstant.COL_Y, y);
       values.put(MyConstant.COL_Z, z);
       values.put(MyConstant.COL_CLASS, class_);
       return db.insert(MyConstant.TABLE_NAME,null, values);
   }
```

```
   public List<DataBean> queryRecord() {
       Cursor cursor = db.rawQuery("SELECT * FROM " +
MyConstant.TABLE_NAME, null);
       List<DataBean> recordBeanList = new ArrayList<>();
       if (cursor.moveToFirst()) {
           do {
               recordBeanList.add(new
DataBean(Float.parseFloat(cursor.getString(1)),

Float.parseFloat(cursor.getString(2)),

Float.parseFloat(cursor.getString(3)),
                       cursor.getString(4)));
           } while (cursor.moveToNext());
       }
       cursor.close();
       return recordBeanList;
   }
}
```

KNN



K-value

k_value = (int) Math.sqrt (dataList.size());

Distance

PriorityQueue

Calculate all the distance from database with the sensor data

K-nearest

HashMap

Pull out K numbers of data from PriorityQueue

Result

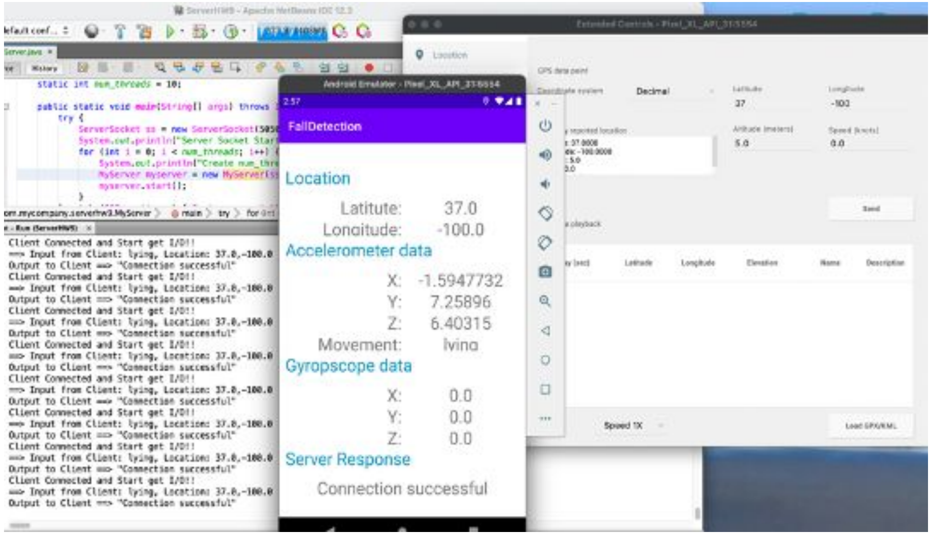
Count

Count most label class in the HashMap

Sent to Server

```
private TextView textIn;
private EditText patientID_editText;

void sentToServer() {
   StrictMode.ThreadPolicy policy = new StrictMode.ThreadPolicy.Builder().permitAll().build();
   StrictMode.setThreadPolicy(policy);
   String P_ID = patientID_editText.getText().toString();
   String msg = "Patient:"+ P_ID + ", " + movementResult +", Location: "+ stringLa + "," +stringLo;
   Socket socket;
     DataOutputStream dataOutputStream;
     DataInputStream dataInputStream;
     try {
         socket = new Socket("192.168.1.72", 5050);
         dataOutputStream = new
  DataOutputStream(socket.getOutputStream());
         dataOutputStream.writeUTF(msg);
         dataInputStream = new
  DataInputStream(socket.getInputStream());
         textIn.setText(dataInputStream.readUTF());
         dataOutputStream.close();
         dataOutputStream.flush();
         socket.close();
     } catch (IOException e) {
         e.printStackTrace();
     }
  }
```



```
static int num_threads = 10;

public static void main(String[] args) throws
    try {
        ServerSocket ss = new ServerSocket(5050
        System.out.println("Server Socket Star
        for (int i = 0; i < num_threads; i++) {
            System.out.println("Create num_thre
            MyServer myserver = new MyServer(
            myserver.start();
        }
```

```
Client Connected and Start get I/O!!
==> Input from Client: lying, Location: 37.0,-100.0
Output to Client ==> "Connection successful"
Client Connected and Start get I/O!!
==> Input from Client: lying, Location: 37.0,-100.0
Output to Client ==> "Connection successful"
Client Connected and Start get I/O!!
==> Input from Client: lying, Location: 37.0,-100.0
Output to Client ==> "Connection successful"
Client Connected and Start get I/O!!
==> Input from Client: lying, Location: 37.0,-100.0
Output to Client ==> "Connection successful"
Client Connected and Start get I/O!!
==> Input from Client: lying, Location: 37.0,-100.0
Output to Client ==> "Connection successful"
Client Connected and Start get I/O!!
==> Input from Client: lying, Location: 37.0,-100.0
Output to Client ==> "Connection successful"
Client Connected and Start get I/O!!
==> Input from Client: lying, Location: 37.0,-100.0
Output to Client ==> "Connection successful"
Client Connected and Start get I/O!!
==> Input from Client: lying, Location: 37.0,-100.0
Output to Client ==> "Connection successful"
```

FallDetection

Location

Latitute: 37.0
Longitude: -100.0

Accelerometer data

X: -1.5947732
Y: 7.25896
Z: 6.40315
Movement: lying

Gyropscope data

X: 0.0
Y: 0.0
Z: 0.0

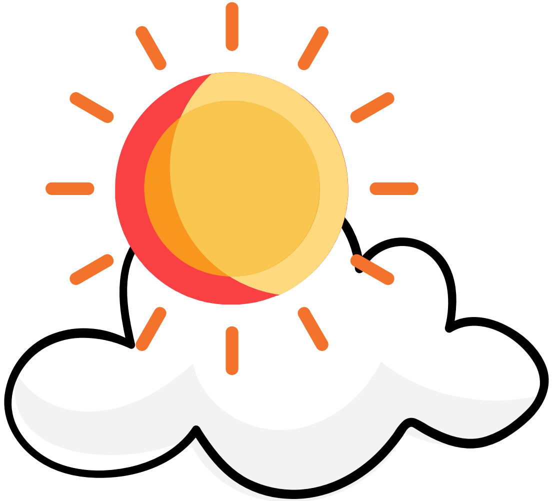
Server Response

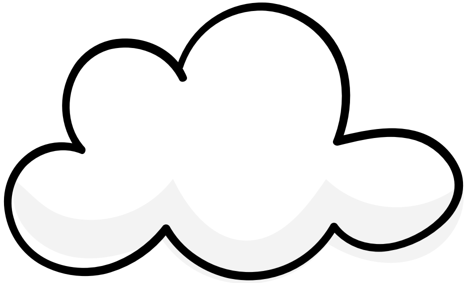
Connection successful

Sent SMS

```
private EditText phone_Edittext;
private SmsManager smsManager;

if (movementResult.equals("falling")) {
   sentSMS();
}
void sentSMS() {
   String P_ID = patientID_editText.getText().toString();
   String phoneNumber = phone_Edittext.getText().toString();
   String sms_msg = "Patient:"+ P_ID + ", " + movementResult +", Location: "+
stringLa + "," +stringLo;
   try {
       smsManager = SmsManager.getDefault();
       smsManager.sendTextMessage(phoneNumber, null, sms_msg, null, null);
       Log.i("Send SMS", "");
       Toast.makeText(getApplicationContext(), "SMS sent.",
Toast.LENGTH_LONG).show();
   } catch (Exception e) {
       Toast.makeText(getApplicationContext(), "SMS faild, please try again.",
Toast.LENGTH_LONG).show();
   }
}
```





Any Questions?