ASSIGNMENT 1

Anudeep Nallamothu
nan4@umbc.edu

Venkata Rami Reddy, Bujunuru
bo26494@umbc.edu

Approach 1:

Algorithm to process Sleeping, Sitting and Walking data collected from Sensors:
We made some assumptions, they are as follows:

- When the phone face is stable and parallel to the ground, it is in sleeping position.
- When the phone is perpendicular to the ground, it is in sitting position.
- When there is motion then it is considered as Walking.

Data Collection:

- We collect data from accelerometer and calculations are done on those values.

Implementation/Logic:

- The Main UI thread creates a BoundService on StartUp.
- On Clicking the "Start Activity" button, the sensorListener is registered for the accelerometer.
- A timer thread is created to trigger processing of raw sensor data and *ProcessDataForActivity* is the TimerTask that is triggered at the 2 minute interval.
- processRawAccelerometerData() method does all the processing on the raw data.
- For a given sample in the List of samples collected during the interval, the activity is calculated (sleeping or sitting or walking) and is added to aData list.
- ProcessActivityCollection() iterates over the aData and decides that, the most performed activity during interval as the activity for the interval.

Process to collect data:
We implemented activity as a Bound service.This service collects data from Sensor manager.

Algorithm to detect activity:

AccX: X axis component of accelerometer data
AccY: Y axis component of accelerometer data
AccZ: Z axis component of accelerometer data

Sleeping : In position, the phone is kept parallel to the ground. Hence AccY(inclusive of gravity) has the least value of all three components.
AccY < Min(AccX, Accz)
Sitting: In this position, the phone is kept perpendicular to the ground . Hence AccY has the highest value of all three components.
AccY > Max(AccX,AccZ)
Walking:
First the gravity factor is divided based the orientation of the phone.

This is done by multiplying the respective axis accelerometer value with 9.8 and divided by the normalization factor.

double normalization = “RMS of accelerometer values”

gravity[0] = accelerometer_values[0]*GRAVITY_CONSTANT/normalization;

gravity[1] = accelerometer_values[1]*GRAVITY_CONSTANT/normalization;

gravity[2] = accelerometer_values[2]*GRAVITY_CONSTANT/normalization;

- Then we subtract the gravityfactors from the respective accelerometer_values to get the absolute acceleration values(linear_acceleration).
- We then find the Root Mean Square(RMS) value of this linear_acceleration and compare it to a threshold(We assumed this value as **0.5** after a few test runs of the application.)

If the RMS value is greater than the magnitude, we detect the sample to represent the walking activity

Storing in External Storage:

We are polling the bound service every 2 minutes to fetch the list of activities and storing in the external storage.

It is stored in the file called “ActivityTracker.txt”. It is demonstrated in the video

Screenshot of Activity UI displaying activities: