### [Homework#15](https://bb-csuohio.blackboard.com/webapps/assignment/uploadAssignment?content_id=_6362383_1&course_id=_168060_1&group_id=&mode=view)

The the improved pedometer app with the custom activity detection algorithm has a design issue: the step data are pulled from the SQLite database only at the start of the app. So, if you have run the app for a few minutes with multiple steps, the display the main activity won’t change at all. To see the data collected, you would have to close the app and restart it.

To address this issue, modify the app so that whenever a new batch of steps are recorded, the display on the main activity would be updated. Alternatively, you may add a refresh button on the main activity layout to retrieve the data and display it.

The app actually has another issue: the step count is not as accurate as the step counter. Do some experiments (you will need to carry your phone and walk for say, 100 steps, and compare what is recorded in your app with respect to the actual steps you made) and tune the 3 thresholds used for step counting so that they would work for you – meaning the step count is almost as accurately as the actual number of steps

Solution:

Code display step count:

<TextView  
 android:id="@+id/stepssincereboot"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="Hello World!"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent" />

Code to store and display steps through out the day in list view:

<ListView  
 android:id="@+id/steps\_list"  
 android:layout\_width="fill\_parent"  
 android:layout\_height="0dip"  
 android:layout\_weight="1"  
 android:cacheColorHint="@android:color/transparent"  
 android:dividerHeight=".5dip" />

Java code to count and display steps number:

public class StepCounterActivity extends AppCompatActivity implements SensorEventListener {  
 private SensorManager mSensorManager;  
 private Sensor mSensor;  
 private boolean isSensorPresent;  
 private TextView mStepsSinceReboot;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*stepcounter\_layout*);  
  
 mStepsSinceReboot = (TextView) findViewById(R.id.*stepssincereboot*);  
 mSensorManager = (SensorManager) this.getSystemService(Context.*SENSOR\_SERVICE*);  
 if (mSensorManager.getDefaultSensor(Sensor.*TYPE\_STEP\_COUNTER*) != null) {  
 mSensor = mSensorManager.getDefaultSensor(Sensor.*TYPE\_STEP\_COUNTER*);  
 isSensorPresent = true;  
 } else {  
 isSensorPresent = false;  
 }  
 }  
  
 @Override  
 protected void onResume() {  
 super.onResume();  
 if (isSensorPresent) {  
 mSensorManager.registerListener(this, mSensor, SensorManager.*SENSOR\_DELAY\_NORMAL*);  
 }  
 }  
  
 @Override  
 protected void onPause() {  
 super.onPause();  
 if (isSensorPresent) {  
 mSensorManager.unregisterListener(this);  
 }  
 }  
  
 @Override  
 public void onSensorChanged(SensorEvent event) {  
 mStepsSinceReboot.setText("Steps since reboot:" + String.*valueOf*(event.values[0]));  
 }  
  
 @Override  
 public void onAccuracyChanged(Sensor sensor, int i) {  
  
 }  
}

Screenshots:

A white background with black dots

Description automatically generated A number on a white background

Description automatically generated