

## **mainActivity.java**

```
Package com.example.pedometer;
Import Public class MainActivity extends AppCompatActivity {
    @Override
    Protected void onCreate (Bundle savedInstanceState){
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    }
    Public void navigateToStepCounterActivity(View v) {
        Intent mIntent =new Intent(packageContext: this,
stepsCounterActivity.class);
        startActivity(mIntent);
    }
    Public void navigateToStepHistoryActivity(View v) {
        Intent mIntent =new Intent(packageContext: this,
stepsHistoryActivity.class);
        startActivity(mIntent);
    }
    Public void navigateToStepAlgoActivity(View v) {
Intent mIntent =new Intent(packageContext:
this,CustomAlgoResultsActivity.class);
        startActivity(mIntent);
    }
}
```

## **stepsCounterActivity**

```
public class StepsCounterActivity extends Activity implements
SensorEventListener{

    private SensorManager mSensorManager;

    private Sensor mSensor;
```

```
private boolean isSensorPresent; private TextView  
StepssinceReboot;
```

```
@Override
```

```
protected void onCreate(Bundle savedInstanceState) {
```

```
    super.onCreate(savedInstanceState);
```

```
    setContentView(R.layout.stepcounter_layout);
```

```
    StepsSinceReboot = (TextView) findViewById(R.id.stepssincereboot);
```

```
    #SensorManager = (SensorManager)
```

```
    this.getSystemService(Context.SENSOR_SERVICE);
```

```
    if (mSensorManager.getDefaultSensor
```

```
        (Sensor.TYPE_STEP_COUNTER) != null) { #Sensor =
```

```
        mSensorManager.getDefaultSensor
```

```
        (Sensor.TYPE_STEP_COUNTER);
```

```
    isSensorPresent = true;
```

```
    } else {
```

```
        isSensorPresent = false;
```

```
    }
```

```
    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 accuracy) {
```

```
    mStepsSinceReboot.setText("Steps since reboot:" +  
        String.valueOf(event.values[1]));
```

```
}
```

```
@Override
```

```
protected void onDestroy() {
```

```
    super.onDestroy();
```

```
    mSensorManager = null;
```

```
}
```

```
}
```

mSensor = null

### **XML:**

```
<?xml version="1.0" encoding="utf-8"?> <LinearLayout  
xmlns:android="http://schemas.android.com/apk/res/android"  
xmlns:tools="http://schemas.android.com/tools"
```

```
    android:layout_width="match_parent"
```

```
    android:layout_height="match_parent"  
    android:paddingLeft="@dimen/activity_horizontal_margin"  
    android:paddingRight="@dimen/activity_horizontal_margin"  
    android:paddingTop="@dimen/activity_vertical_margin"
```

```
    android:paddingBottom="@dimen/activity_vertical_margin"  
    tools:context=".StepsCounterActivity"  
    android:background="@android:color/white"  
    android:orientation="vertical">
```

```
        android:layout_height="wrap_content"
```

```
        <TextView android:layout_width="wrap_content" android:text="Steps  
        since reboot: @" android:textColor="@android:color/black"  
        android:id="@+id/stepssincereboot"  
        android:layout_gravity="center_horizontal"
```

```
    />
```

```
</LinearLayout>
```

**java**

```

public class StepsHistoryActivity extends Activity{
StepsHistoryActivity.java4
StepsCounterActivity.java);

MainActivity.java package com.example.pedometer;

StepsDBHelper mStepsDBHelper;
ListView mSensorListView;
ListAdapter ListAdapter;
ArrayList<DateStepsModel>
mStepCountList;

@Override
protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.pedometerlist_layout);

"SensorListView = (ListView) findViewById(R.id.steps_list);

getDataForList();

mListAdapter = new ListAdapter(mStepCountList, this);

mSensorListView.setAdapter(mListAdapter);

Intent stepsIntent = new Intent(getApplicationContext(),
StepsService.class

startService(stepsIntent);

}

public void getDataForList()

```

```
{
```

```
mStepsDBHelper = new StepsDBHelper(this);
```

```
mStepCountList = mStepsDBHelper.readSteps Entries
```

## **Pedometer.java**

```
StepsCounterA
```

```
}
```

```
public class StepsDBHelper extends SQLiteOpenHelper
```

```
private static final int DATABASE_VERSION 1;
```

```
private static final String DATABASE_NAME="StepsDatabase";
```

```
private static final String TABLE STEPS SUMMARY  
"StepsSummary";
```

```
private static final String ID = "id";
```

```
private static final String STEPS_COUNT"stepscount"; private static  
final String CREATION DATE="Creationdate";
```

```
private static final String CREATE_TABLE_STEPS SUMMARY=  
"CREATE TABLE "
```

```
+ TABLE_STEPS SUMMARY + "("+ID + " INTEGER PRIMARY KEY  
AUTOINCREMENT," + CREATION DATE +" TEXT, "+ STEPS  
COUNT+ INTEGER+"");
```

```

@Override
public void onCreate(SQLiteDatabase db) {
    db.execSQL(CREATE_TABLE_STEPS_SUMMARY);

}

public boolean createStepsEntry() {

    boolean isDateAlreadyPresent = false;

    boolean createSuccessful = false; int currentDateStepCounts = 0;

    Calendar mCalendar = Calendar.getInstance(); String todayDate =
    String.valueOf(mCalendar.get(Calendar.MONTH))+"/"+

```

### **Steps Counter.java**

```

String.valueOf(mCalendar.get(Calendar.DAY_OF_MONTH)+1)+"/"+String.v
alueOf(Calendar.get(Calendar.YEAR))
String selectQuery= "SELECT TABLE STEPS SUMMARY"+STEPS_COUNT +
FROM "+WHERE+ CREATION_DATE"+ todayDate+"*";

SQLiteDatabase db= this.getReadableDatabase();

Cursor c= db.rawQuery(selectQuery, null);

if (c.moveToFirst()) {
    do
    {
        isDateAlreadyPresent= true;
        currentDateStepCounts = c.getInt((c.getColumnIndex(STEPS_COUNT)));
    }
    while (c.moveToNext());

}

```

```
db.close();

catch(Exception e) {

e.printStackTrace();
}
try {

SQLiteDatabase db = this.getWritableDatabase();
ContentValues values = new ContentValues();
values.put(CREATION DATE, todayDate);
(isDateAlreadyPresent) {
values.put(STEPS_COUNT, ++currentDateStepCounts); int row
db.update(TABLE_STEPS SUMMARY, values, CREATION DATE
if(row 1)
{
wwcreateSuccessful =true;

}
```