globsyn

Taking People To The Next Level

Integrated Walkmeter

Group Members:

Neelabja Roy, Heritage Institute of Technology, Kolkata (Form number- 37240)

Trijit Bhattacharjee, FIEM (Form Number - 35420)

Table of Contents

Acknowledgement

I take this opportunity to express my profound gratitude and deep regards to my faculty Mr.Arnab Chakraborty for his exemplary guidance, monitoring and constant encouragement throughout the course of this project. The blessing, help and guidance given by him/her time to time shall carry me a long way in the journey of life on which I am about to embark.

I am obliged to my project team members for the valuable information provided by them in their respective fields. I am grateful for their cooperation during the period of my assignment.

Neelabja Roy

Trijit Bhattacharjee

Project Objective

Develop and implement an Android based Integrated Pedometer System to help enthusiastic walkers in the pursuit of having a comprehensive understanding of their walking patterns, energy expenditure as well as enable a single-platform solution for other necessary accesses for walk-time requirements, even in a resource constrained android environment system.

The primary project goals consist of:

- *Providing users with dynamic walk-time parameter measurements and presenting a comprehensive and satisfactory representation of their activity the enthusiastic user will be interested in.*
- *Enable users to know about their duration of walk, number of steps, distance covered, walking speed, energy count while on their activity.*
- *Enable users to also have a knowledge of their patterns of their energy expenditure throughout the day and also possibly have recommender systems suggesting activity changes towards goal.*
- *Providing users a single application access to useful walking accessories like location-finding maps for explorers as well and flashlight systems for those who love to take a walk in the dark.*
- *Providing users with an in-application music player system with access to .mp3 and .wav files in their device to listen to their music on the go.*
- *The primary goal being to provide a pedometer application enabled with in-application accessories that makes efficient time-resource utilisation of the environment and agent by not having to download, open and access different applications individually.*
- *Best possible efforts have been made to provide users with the most satisfactory UX and UI experience.*

Project Scope

(This page shall elaborate the boundary conditions of the project)

The broad scope of the Integrated Pedometer System (LMS) project includes:

- *The system will be available to users as a one-stop solution to enthusiastic users with all their walktime needs and without the need of individually downloading, installing, opening, accessing different apps with different app sizes and resource needs to use during walking.*
- *The system will support pedometer features with duration of walk, steps taken, distance covered, walking speed, calories burnt.*
- *It will also be a one-platform access to services of map, nighttime-flashlight, mediaplayer.*
- *The system looks to help in the process of building a healthy community around the exercise of activities by being a resource-efficient application in the process of doing so.*

Future Scope of Improvements

The broad scope of the Integrated Pedometer System (LMS) project includes:

- *Due to constraint with time we couldn't properly implement the statistical visualisation module from the generated data and we will surely look to implement as soon as we have generated enough data.*
- *We look to implement a proper goal-based recommendation and coaching system that reminds and alerts users of their target goals and the necessary action requirements.*
- *We want to improve our map services to make it a tool for providing a wholesome walking experience and also as tool for building a walking based social community where a user can locate other app users in his walking spot (say his morning walk area) and group for a community walk.*
- *We look to sync our application with IoT devices and wearables for a better integrated experience.*
- *We look to drastically improve the user interface and user experience based upon feedback and recommendations.*

Requirement Specification

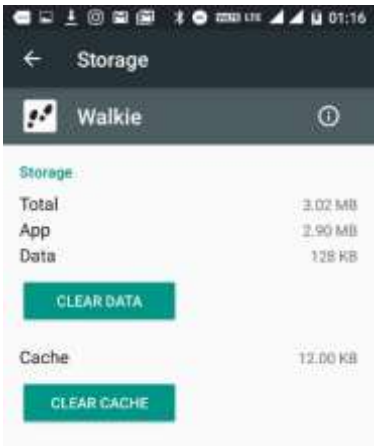
(This section contains the business challenge that the application is solving from the point of view of a user of the system)

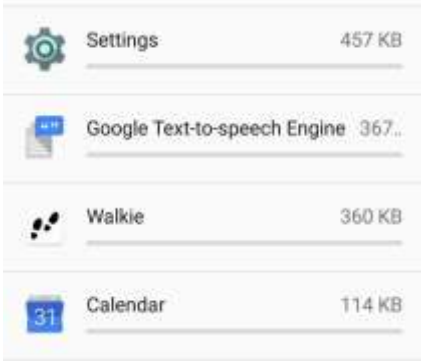
- Domain Description
  - Walking being an activity that has been exercised by many also to address health issues, the possible applications of smartphone sensors to track and measure and analyse the salient features of the data from walking came to be noticed and smartphones can also be thus also used today as a dynamic health regulator device.

- Problem Definition
  - We observed that in the status quo while in any such activity, be it for fitnesss or be it for any random walking, we also like to use other smartphone applications like listening to music in a media player application, using GoogleMaps to know where we are or where we want to go, and/or using another flashlight application just to see our way in the dark.
  - So just for walking we have to individually download, install and access the other applications individually to use them, that too simultaneously, costing significant usages of storage and memory
  - In a resource-intensive platform like android and in a resource constraint like ours these resources of Storage and System memory continue to be a predominant factor for our smartphone usages.
  - This is one of the most unaddressed problem that we attempted to address through our application along with providing the salient features of a simple pedometer.

- Hardware /Software Requirements
  - ○ Hardware requirement
    - Minimum Ram: 512 MB
    - Minimum Space: 20MB
    - Accelerometer (Mandatory)
    - Camera flash (Recommended)
    - Gyroscope (Recommended
  - ○ Software requirement
    - Minimum Req: Android Kitkat (API LEVEL 19)
    - Recommended Req: AndroidLollipop 5.1 (API LEVEL 22)
    - Target Spec: Android Nougat 7 (API LEVEL 25)

In Memory and Storage



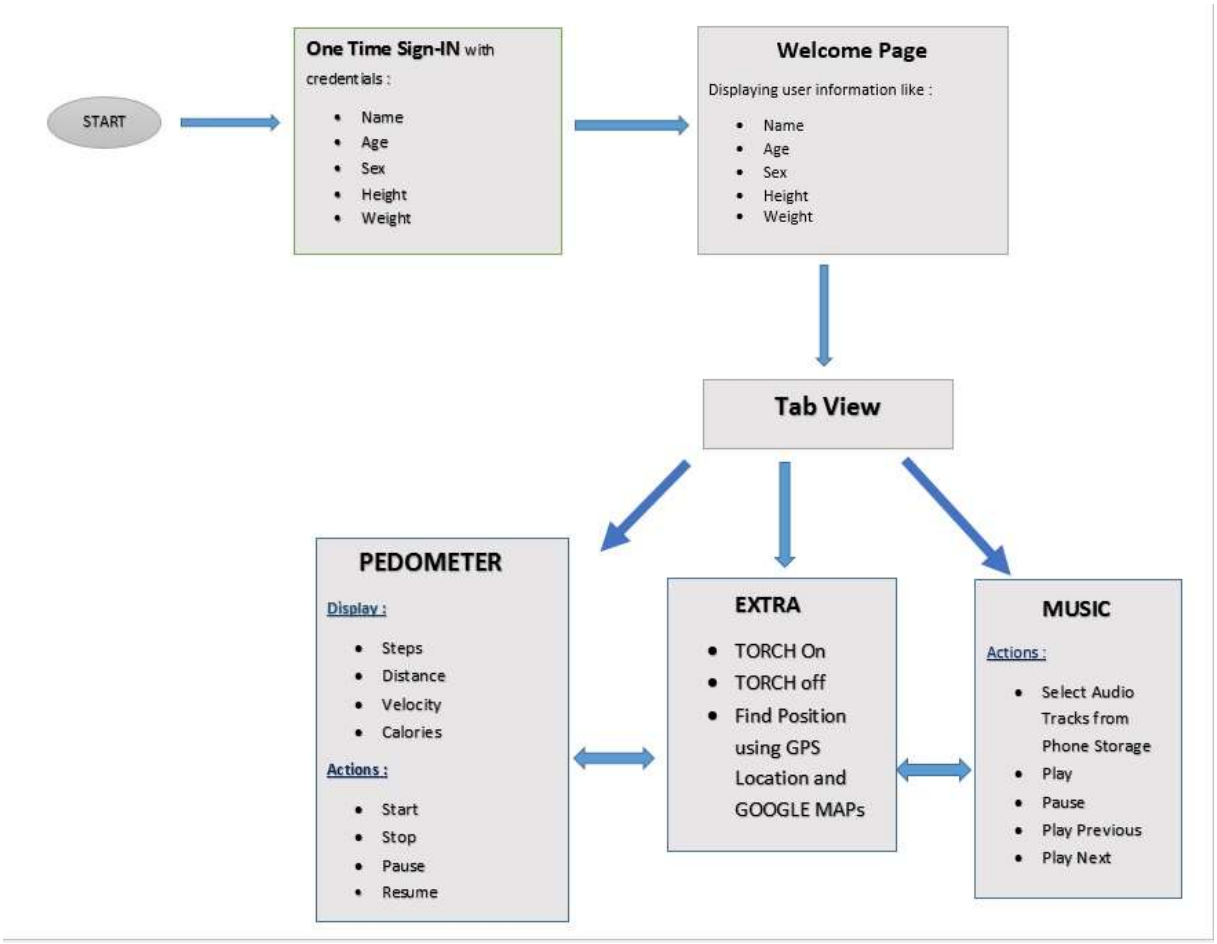


We have noticed that even the most simple and basic pedometer application as available in the market costs us 2.5 MB and above.
Additionally, Flashlight basic apps come at about 3MB, LocationFinder a minimum of 3MBs, and Music Players at a minimum of 5MBs

Here we are providing the same solution for an app that costs us barely 1.5MB that too with data.

Application Work Flow

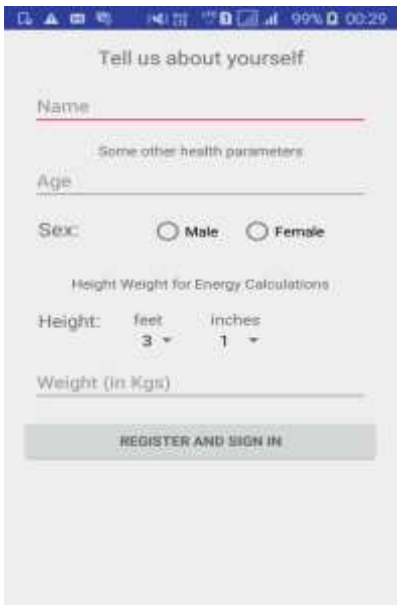
(This section displays the flow of information in the application)



Screenshots

App icon:

one time registration

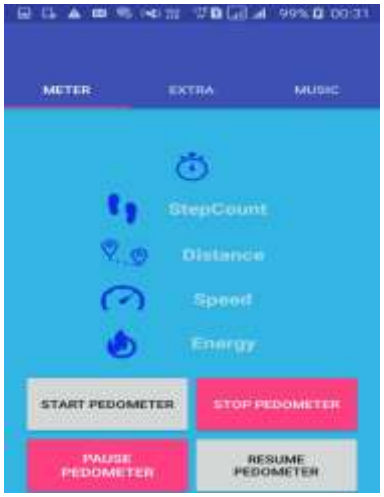


reg fill up

welcome page – on click

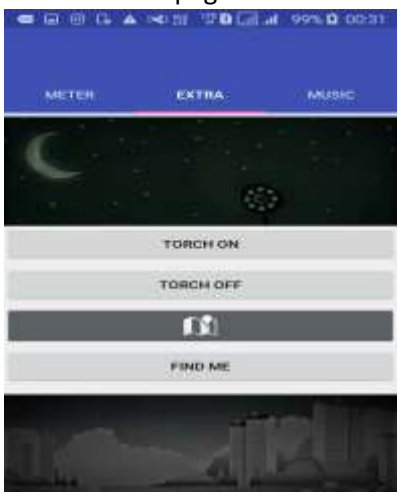
After Proceed from welcome page we have the tabbed activity page with Scale (Pedometer) page, Extras Page and Music Page. Below are the visualisations of the activities.

Before clicking “Start Pedometer”

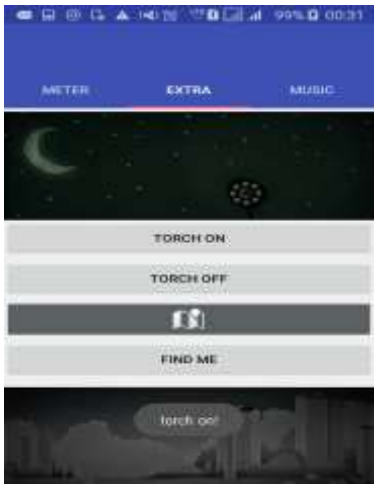


After Clicking Start and then pausing

The “Extra” page is as

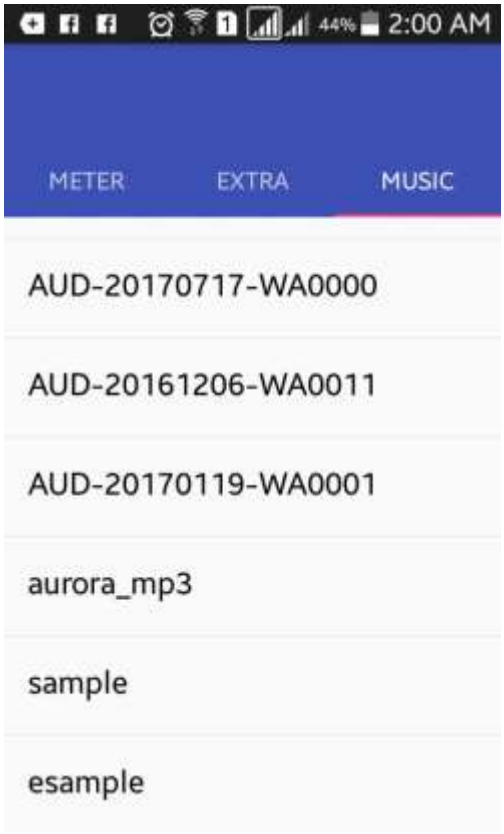


Torch switched on!



FindMe- house level accurate

Music player tab



Music Item Playingg

Code

Signin code

```
package com.reg.user.reg;

/**
 * Created by USER on 17-07-2017.
 */
import android.app.Activity;
import android.content.Intent;
import android.content.SharedPreferences;
import android.os.Bundle;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.ArrayAdapter;
import android.widget.Button;
import android.widget.EditText;
import android.widget.ImageView;
import android.widget.RadioButton;
import android.widget.RadioGroup;
import android.widget.Spinner;
import android.widget.Toast;

public class Signin extends Activity {

    Button button;
    RadioGroup genderRadioGroup;
    EditText name;
    EditText age;
    EditText weight;
    Spinner ht_feet;
    Spinner ht_inches;
    boolean completed;
    SharedPreferences sharedPrefs;
```

```
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.signin);
        sharedPrefs = getSharedPreferences("sharedprefs", 0);
        completed = sharedPrefs.getBoolean("complete", false);
        if (completed == true) {
            Intent intent = new Intent();
            intent.setClass(Signin.this, hi2.class);
            startActivity(intent);
            finish();
        }


        name = (EditText) findViewById(R.id.name);
        age = (EditText) findViewById(R.id.age);
        weight = (EditText) findViewById(R.id.weight);
        genderRadioGroup = (RadioGroup) findViewById(R.id.gender);
        ht_feet = (Spinner) findViewById(R.id.ft_spinner);
        ht_inches = (Spinner) findViewById(R.id.inch_spinner);
        button=(Button) findViewById(R.id.submit);



       // Spinner ht_feet = (Spinner) findViewById(R.id.ft_spinner);

        ArrayAdapter<CharSequence> adapter1 = ArrayAdapter.createFromResource(this,
R.array.ft_range, android.R.layout.simple_spinner_item);

adapter1.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item);
        ht_feet.setAdapter(adapter1);

        //Spinner ht_inch = (Spinner) findViewById(R.id.inch_spinner);
        ArrayAdapter<CharSequence> adapter = ArrayAdapter.createFromResource(this,
R.array.inch_range, android.R.layout.simple_spinner_item);

adapter.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item);
        ht_inches.setAdapter(adapter);


        button.setOnClickListener(new OnClickListener() {

            @Override
            public void onClick(View v) {
                String name1 = name.getText().toString();
```

```
                String age1 = age.getText().toString();
                String weight1 = weight.getText().toString();
                String feet1 = ht_feet.getSelectedItem().toString();
                String inches1 = ht_inches.getSelectedItem().toString();

                int id = genderRadioGroup.getCheckedRadioButtonId();
                RadioButton radBut = (RadioButton) findViewById(id);
                String gender1= radBut.getText().toString();

                if (name1.equals("") || age1.equals("")
                        || weight1.equals("") || gender1.equals("")
                        || feet1.equals("")|| inches1.equals("")) {
                    Toast.makeText(getApplicationContext(), "Enter all fields",
                            Toast.LENGTH_SHORT).show();
                } else{
                    SharedPreferences.Editor editors = sharedPrefs.edit();
                    editors.putString("name_dt", name1);
                    editors.putString("age_dt", age1);
                    editors.putString("gender_dt", gender1);
                    editors.putString("weight_dt", weight1);
                    editors.putString("feet_dt", feet1);
                    editors.putString("inches_dt", inches1);
                    editors.putBoolean("complete", true);
                    editors.apply();
                    Intent reg_done=new Intent(Signin.this, hi2.class);
                    startActivity(reg_done);
                    finish();
                }
            }
        });
    }
}
```

Welcome code

```
package com.reg.user.reg;

import android.annotation.SuppressLint;
```

```
import android.content.Context;
import android.content.Intent;
import android.content.SharedPreferences;
import android.preference.PreferenceManager;
import android.support.v7.app.ActionBar;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.os.Handler;
import android.view.MotionEvent;
import android.view.View;
import android.widget.TextView;
import android.widget.Toast;

import static android.R.attr.key;

/**
 * An example full-screen activity that shows and hides the system UI (i.e.
 * status bar and navigation/system bar) with user interaction.
 */
public class hi2 extends AppCompatActivity {
    /**
     * Whether or not the system UI should be auto-hidden after
     * {@link #AUTO_HIDE_DELAY_MILLIS} milliseconds.
     */
    private static final boolean AUTO_HIDE = true;

    /**
     * If {@link #AUTO_HIDE} is set, the number of milliseconds to wait after
     * user interaction before hiding the system UI.
     */
    private static final int AUTO_HIDE_DELAY_MILLIS = 3000;

    /**
     * Some older devices needs a small delay between UI widget updates
     * and a change of the status and navigation bar.
     */
    private static final int UI_ANIMATION_DELAY = 300;
    private final Handler mHideHandler = new Handler();
    private View mContentView;
    private final Runnable mHidePart2Runnable = new Runnable() {
        @SuppressLint("InlinedApi")
        @Override
        public void run() {
            // Delayed removal of status and navigation bar
```

```
            // Note that some of these constants are new as of API 16 (Jelly Bean)
            // and API 19 (KitKat). It is safe to use them, as they are inlined
            // at compile-time and do nothing on earlier devices.
            mContentView.setSystemUiVisibility(View.SYSTEM_UI_FLAG_LOW_PROFILE
                    | View.SYSTEM_UI_FLAG_FULLSCREEN
                    | View.SYSTEM_UI_FLAG_LAYOUT_STABLE
                    | View.SYSTEM_UI_FLAG_IMMERSIVE_STICKY
                    | View.SYSTEM_UI_FLAG_LAYOUT_HIDE_NAVIGATION
                    | View.SYSTEM_UI_FLAG_HIDE_NAVIGATION);
        }
    };
    private View mControlsView;
    private final Runnable mShowPart2Runnable = new Runnable() {
        @Override
        public void run() {
            // Delayed display of UI elements
            ActionBar actionBar = getSupportActionBar();
            if (actionBar != null) {
                actionBar.show();
            }
            mControlsView.setVisibility(View.VISIBLE);
        }
    };
    private boolean mVisible;
    private final Runnable mHideRunnable = new Runnable() {
        @Override
        public void run() {
            hide();
        }
    };
    /**
     * Touch listener to use for in-layout UI controls to delay hiding the
     * system UI. This is to prevent the jarring behavior of controls going away
     * while interacting with activity UI.
     */
    private final View.OnTouchListener mDelayHideTouchListener = new
View.OnTouchListener() {
        @Override
        public boolean onTouch(View view, MotionEvent motionEvent) {
            if (AUTO_HIDE) {
                delayedHide(AUTO_HIDE_DELAY_MILLIS);
            }
            return false;
        }
    };
```

```
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

        setContentView(R.layout.activity_hi2);

        mVisible = true;
        mControlsView = findViewById(R.id.fullscreen_content_controls);
        mContentView = findViewById(R.id.fullscreen_content);


        // Set up the user interaction to manually show or hide the system UI.
        mContentView.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                toggle();
            }
        });

        // Upon interacting with UI controls, delay any scheduled hide()
        // operations to prevent the jarring behavior of controls going away
        // while interacting with the UI.

findViewById(R.id.dummy_button).setOnTouchListener(mDelayHideTouchListener);

        //Bundle b = getIntent().getExtras();
        TextView name_get = (TextView) findViewById(R.id.name_val);
        TextView age_get = (TextView) findViewById(R.id.age_val);
        TextView gender_get = (TextView) findViewById(R.id.sex_val);
        TextView weight_get = (TextView) findViewById(R.id.weight_val);
        TextView ht_feet_get = (TextView) findViewById(R.id.ht_ft);
        TextView ht_inches_get = (TextView) findViewById(R.id.ht_in);


        SharedPreferences sharedPrefs = getSharedPreferences("sharedprefs", 0);
        String name= sharedPrefs.getString("name_dt", " ");
        String age= sharedPrefs.getString("age_dt", " ");
        String gender= sharedPrefs.getString("gender_dt", " ");
        String weight= sharedPrefs.getString("weight_dt", " ");
        String ht_feet= sharedPrefs.getString("feet_dt", " ");
        String ht_inches= sharedPrefs.getString("inches_dt", " ");
```

```
        name_get.setText(name);
        age_get.setText(age);
        gender_get.setText(gender);
        weight_get.setText(weight);
        //ht_feet.setSelection(b.getCharSequence("height_feet"));
        //ht_inches.setText(b.getCharSequence("height_inches"));
        ht_feet_get.setText(ht_feet);
        ht_inches_get.setText(ht_inches);
    }

    @Override
    protected void onPostCreate(Bundle savedInstanceState) {
        super.onPostCreate(savedInstanceState);

        // Trigger the initial hide() shortly after the activity has been
        // created, to briefly hint to the user that UI controls
        // are available.
        delayedHide(100);
    }

    private void toggle() {
        if (mVisible) {
            hide();
        } else {
            show();
        }
    }

    private void hide() {
        // Hide UI first
        ActionBar actionBar = getSupportActionBar();
        if (actionBar != null) {
            actionBar.hide();
        }
        mControlsView.setVisibility(View.GONE);
        mVisible = false;

        // Schedule a runnable to remove the status and navigation bar after a
delay
        mHideHandler.removeCallbacks(mShowPart2Runnable);
        mHideHandler.postDelayed(mHidePart2Runnable, UI_ANIMATION_DELAY);
    }

    @SuppressLint("InlinedApi")
    private void show() {
```

```
        // Show the system bar
        mContentView.setSystemUiVisibility(View.SYSTEM_UI_FLAG_LAYOUT_FULLSCREEN
                | View.SYSTEM_UI_FLAG_LAYOUT_HIDE_NAVIGATION);
        mVisible = true;

        // Schedule a runnable to display UI elements after a delay
        mHideHandler.removeCallbacks(mHidePart2Runnable);
        mHideHandler.postDelayed(mShowPart2Runnable, UI_ANIMATION_DELAY);
    }

    /**
     * Schedules a call to hide() in [delay] milliseconds, canceling any
     * previously scheduled calls.
     */
    private void delayedHide(int delayMillis) {
        mHideHandler.removeCallbacks(mHideRunnable);
        mHideHandler.postDelayed(mHideRunnable, delayMillis);
    }

    public void Home(View view){
        Intent home=new Intent(this, Home.class);
        Toast.makeText(this, "Welcome!", Toast.LENGTH_SHORT).show();
        startActivity(home);
    }
}
```

Tabview Home code

```
package com.reg.user.reg;

import android.support.v4.app.Fragment;
import android.support.v4.app.FragmentManager;
import android.support.v4.app.FragmentPagerAdapter;

import android.support.design.widget.TabLayout;
import android.support.design.widget.FloatingActionButton;
import android.support.design.widget.Snackbar;
```

```
import android.support.v7.app.AppCompatActivity;
import android.support.v7.widget.Toolbar;

import android.support.v4.app.Fragment;
import android.support.v4.app.FragmentManager;
import android.support.v4.app.FragmentPagerAdapter;
import android.support.v4.view.ViewPager;
import android.os.Bundle;
import android.util.Log;
import android.view.LayoutInflater;
import android.view.Menu;
import android.view.MenuItem;
import android.view.View;
import android.view.ViewGroup;

import android.widget.TextView;

public class Home extends AppCompatActivity {

    private static final String TAG = "MainActivity";
    private SectionsPageAdapter mSectionsPageAdapter;
    private ViewPager mViewPager;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_home);

        Log.d(TAG, "onCreate:Starting");
        mSectionsPageAdapter= new SectionsPageAdapter(getSupportFragmentManager());

        mViewPager = (ViewPager)findViewById(R.id.container);
        setupViewPager(mViewPager);

        TabLayout tabLayout= (TabLayout) findViewById(R.id.tabs);
        tabLayout.setupWithViewPager(mViewPager);
    }

    private void setupViewPager(ViewPager viewPager){

        SectionsPageAdapter adapter=new
SectionsPageAdapter(getSupportFragmentManager());
        adapter.addFragment(new Tab1Fragment(), "Scales");
        adapter.addFragment(new Tab2Fragment(), "Extra");
        adapter.addFragment(new Tab3Fragment(), "Music");
```

```
        viewPager.setAdapter(adapter);
    }

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        // Inflate the menu; this adds items to the action bar if it is present.
        getMenuInflater().inflate(R.menu.menu_home, menu);
        return true;
    }

    @Override
    public boolean onOptionsItemSelected(MenuItem item) {
        // Handle action bar item clicks here. The action bar will
        // automatically handle clicks on the Home/Up button, so long
        // as you specify a parent activity in AndroidManifest.xml.
        int id = item.getItemId();

        //noinspection SimplifiableIfStatement
        if (id == R.id.action_settings) {
            return true;
        }

        return super.onOptionsItemSelected(item);
    }

    public static class PlaceholderFragment extends Fragment {
        /**
         * The fragment argument representing the section number for this
         * fragment.
         */
        private static final String ARG_SECTION_NUMBER = "section_number";

        public PlaceholderFragment() {
        }

        /**
         * Returns a new instance of this fragment for the given section
         * number.
         */
        public static PlaceholderFragment newInstance(int sectionNumber) {
            PlaceholderFragment fragment = new PlaceholderFragment();
            Bundle args = new Bundle();
            args.putInt(ARG_SECTION_NUMBER, sectionNumber);
            fragment.setArguments(args);
            return fragment;
```

```
        }
    }

    /**
     * A placeholder fragment containing a simple view.
     */
}
```

Metre/Scale tab (fragment1)

```
package com.reg.user.reg;

/**
 * Created by USER on 17-07-2017.
 */
import android.content.Context;
import android.content.SharedPreferences;
import android.hardware.SensorManager;
import android.hardware.Camera;
import android.hardware.Sensor;
import android.hardware.SensorEvent;
import android.support.v7.app.AppCompatActivity;
import android.hardware.SensorEventListener;
import android.hardware.SensorManager;
import android.os.SystemClock;
import android.speech.tts.TextToSpeech;
import android.support.v4.app.Fragment;
import android.os.Bundle;
import android.support.annotation.Nullable;
import android.support.v4.app.FragmentActivity;
import android.support.v7.app.AppCompatActivity;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.Button;
import android.widget.Chronometer;
import android.widget.TextView;
```

```
import android.app.Activity;

import com.reg.user.reg.StepListener.StepListener;

import java.util.Timer;

import static android.content.Context.SENSOR_SERVICE;

/**
 * Created by USER on 16-07-2017.
 */

public class Tab1Fragment extends Fragment implements SensorEventListener,
StepListener, TextToSpeech.OnInitListener {
    private static final String TAG = "Tab1Fragment";

    //added from SimpleStepCounter
    private TextView TvSteps,TvDist, TvSpeed, Tvcal;
    private Button BtnStart,BtnStop,BtnPause,BtnResume;
    private long timeWhenStopped = 0;
    private boolean stopClicked;
    private Chronometer chronometer;

    private double BMR;


    private StepDetector simpleStepDetector;
    private SensorManager sensorManager;
    private Sensor accel;
    private static final String TEXT_NUM_STEPS = "Number of Steps: ";
    private int numSteps;
    private Timer timer;

    private TextToSpeech textToSpeech;

    public View onCreateView(LayoutInflater inflater, @Nullable ViewGroup
container, @Nullable Bundle savedInstanceState){
        View view= inflater.inflate(R.layout.tab1fragment, container, false);

        // Get an instance of the SensorManager

        SharedPreferences sharedPrefs =
this.getActivity().getSharedPreferences("sharedprefs", 0);
        String name= sharedPrefs.getString("name_dt", " ");
        double age= Double.parseDouble(sharedPrefs.getString("age_dt", " "));
```

```
        String gender= sharedPrefs.getString("gender_dt", " ");
        double weight= Double.parseDouble(sharedPrefs.getString("weight_dt", " "));
        double ht_feet= Double.parseDouble(sharedPrefs.getString("feet_dt", " "));
        double ht_inches= Double.parseDouble(sharedPrefs.getString("inches_dt", "
"));


        if(gender.equalsIgnoreCase("Male")){
            double cm=30.48*ht_feet+2.54*ht_inches;
            BMR= (13.75*weight)+(5*cm)-(6.76*age)+66;
        }
        else{
            double cm=30.48*ht_feet+2.54*ht_inches;
            BMR= (9.56*weight)+(1.85*cm)-(4.68*age)+655;
        }

        sensorManager = (SensorManager)
getActivity().getSystemService(Context.SENSOR_SERVICE);

        accel = sensorManager.getDefaultSensor(Sensor.TYPE_ACCELEROMETER);
        simpleStepDetector = new StepDetector();
        simpleStepDetector.registerListener(this);

        TvSteps = (TextView) view.findViewById(R.id.tv_steps);
        TvDist = (TextView) view.findViewById(R.id.tv_dist);
        TvSpeed = (TextView) view.findViewById(R.id.tv_walkspeed);
        Tvcal = (TextView) view.findViewById(R.id.tv_cal);


        BtnStart = (Button) view.findViewById(R.id.btn_start);
        BtnStop = (Button) view.findViewById(R.id.btn_stop);
        BtnPause = (Button) view.findViewById(R.id.btn_pause);
        BtnResume = (Button) view.findViewById(R.id.btn_resume);

        chronometer = (Chronometer) view.findViewById(R.id.chronometer);
        chronometer.setText("");

        textToSpeech= new TextToSpeech(getActivity(), this);



        BtnStart.setOnClickListener(new View.OnClickListener() {

            @Override
            public void onClick(View arg0) {
```

```
                textToSpeech.speak("start walking",0,null);

                numSteps = 0;
                sensorManager.registerListener(Tab1Fragment.this, accel,
SensorManager.SENSOR_DELAY_FASTEST);

                chronometer.setBase(SystemClock.elapsedRealtime() );

                chronometer.start();
                stopClicked = false;



            }
        });


        BtnStop.setOnClickListener(new View.OnClickListener() {

            @Override
            public void onClick(View arg0) {

                textToSpeech.speak("stop walking",0,null);
                sensorManager.unregisterListener(Tab1Fragment.this);
                if (!stopClicked) {

                    chronometer.stop();
                    stopClicked = true;

                }
            }
        });

        BtnPause.setOnClickListener(new View.OnClickListener() {

            @Override
            public void onClick(View arg0) {


                textToSpeech.speak("paused walking",0,null);
                sensorManager.unregisterListener(Tab1Fragment.this);
                if (!stopClicked) {
```

```
                    timeWhenStopped = chronometer.getBase() -
SystemClock.elapsedRealtime();
                    int seconds = (int) timeWhenStopped / 1000;

                    chronometer.stop();
                    stopClicked = true;

                }


            }
        });

        BtnResume.setOnClickListener(new View.OnClickListener() {

            @Override
            public void onClick(View arg0) {

                textToSpeech.speak("resumed walking",0,null);

                sensorManager.registerListener(Tab1Fragment.this, accel,
SensorManager.SENSOR_DELAY_FASTEST);

                chronometer.setBase(SystemClock.elapsedRealtime() +
timeWhenStopped);

                chronometer.start();
                stopClicked = false;

            }



        });


        return view;

    }
    public void onAccuracyChanged(Sensor sensor, int accuracy) {
    }

    @Override
    public void onSensorChanged(SensorEvent event) {
        if (event.sensor.getType() == Sensor.TYPE_ACCELEROMETER) {
```

```
            simpleStepDetector.updateAccel(
                    event.timestamp, event.values[0], event.values[1],
event.values[2]);
        }
    }

    @Override
    public void step(long timeNs) {
        numSteps++;
        TvSteps.setText(numSteps+" Steps");

        TvDist.setText(String.format("%.2f",(double)(0.9*numSteps))+" meters");

        String time= String.valueOf(chronometer.getText());
        int hh,mm,ss;
        if(time.length()==5){
            mm=Integer.parseInt(time.substring(0,2));
            ss=Integer.parseInt(time.substring(3))+60*mm;
        }
        else{
            hh=Integer.parseInt(time.substring(0,2));
            mm=Integer.parseInt(time.substring(3,5));
            ss=Integer.parseInt(time.substring(7))+60*mm+60*60*hh;
        }
      TvSpeed.setText(String.format("%.2f",(double)(0.9*numSteps)/ss)+" m/sec");
        double calorie=0;
        calorie=(BMR/(24*60*60))*4*(ss);
        Tvcal.setText(String.format("%.6f",calorie)+" cals");
    }

    @Override
    public void onInit(int status) {

    }

}
```

Extra Tab Code

```
package com.reg.user.reg;

/**
```

```
 * Created by USER on 17-07-2017.
 */

import android.content.Intent;
import android.hardware.Camera;
import android.speech.tts.TextToSpeech;
import android.support.v4.app.Fragment;
import android.os.Bundle;
import android.support.annotation.Nullable;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.Button;
import android.content.Context;
import android.hardware.Camera;
import android.hardware.Camera.Parameters;
import android.hardware.Sensor;
import android.hardware.SensorEvent;
import android.hardware.SensorEventListener;
import android.hardware.SensorManager;
import android.widget.Toast;

import com.reg.user.reg.StepListener.StepListener;

import java.util.Timer;

/**
 * Created by USER on 16-07-2017.
 */

public class Tab2Fragment extends Fragment implements TextToSpeech.OnInitListener {
    private static final String TAG = "Tab2Fragment";
    private Button BtnStart,BtnStop,BtnPause,BtnFindMe,BtnTorchOn,BtnTorchOff;
    private StepDetector simpleStepDetector;
    private SensorManager sensorManager;
    private Sensor accel;
    private static final String TEXT_NUM_STEPS = "Number of Steps: ";
    private int numSteps;
    private Timer timer;

    private TextToSpeech textToSpeech;

    private Camera camera;
    private Parameters parameters;
```

```
    public View onCreateView(LayoutInflater inflater, @Nullable ViewGroup
container, @Nullable Bundle savedInstanceState) {
        View view = inflater.inflate(R.layout.tab2fragment, container, false);

        BtnTorchOn = (Button) view.findViewById(R.id.btn_torch_on);
        BtnTorchOff = (Button) view.findViewById(R.id.btn_torch_off);
        BtnFindMe = (Button) view.findViewById(R.id.find_me);


        textToSpeech= new TextToSpeech(getActivity(), this );
        camera = Camera.open();
        parameters = camera.getParameters();



        BtnTorchOn.setOnClickListener(new View.OnClickListener() {

            @Override
            public void onClick(View arg0) {
                Toast.makeText(getActivity(), "torch on!",
Toast.LENGTH_SHORT).show();
                textToSpeech.speak("Torch turned on",0,null);
                parameters.setFlashMode(Camera.Parameters.FLASH_MODE_TORCH);
                camera.setParameters(parameters);
                camera.startPreview();

            }



        });

        BtnTorchOff.setOnClickListener(new View.OnClickListener() {

            @Override
            public void onClick(View arg0) {

                textToSpeech.speak("Torch turned off",0,null);

                Toast.makeText(getActivity(), "torch off!",
Toast.LENGTH_SHORT).show();
                parameters.setFlashMode(Camera.Parameters.FLASH_MODE_OFF);
                camera.setParameters(parameters);
                camera.stopPreview();
```

```
            }


        });

        BtnFindMe.setOnClickListener(new View.OnClickListener() {

            @Override
            public void onClick(View arg0) {

                textToSpeech.speak("Finding You",0,null);
                Intent intent= new Intent(getActivity(), Findme.class);
                Toast.makeText(getActivity(), "There you are!",
Toast.LENGTH_SHORT).show();
                startActivity(intent);

            }
        });

        return view;
    }

    public void onInit(int status) {

    }
}
```

Music tab code (fragment3)

```
package com.reg.user.reg;

/**
 * Created by USER on 17-07-2017.
 */

import android.content.Intent;
import android.hardware.Camera;
import android.speech.tts.TextToSpeech;
import android.support.v4.app.Fragment;
import android.os.Bundle;
import android.support.annotation.Nullable;
import android.view.LayoutInflater;
```

```
import android.view.View;
import android.view.ViewGroup;
import android.widget.Button;
import android.content.Context;
import android.hardware.Camera;
import android.hardware.Camera.Parameters;
import android.hardware.Sensor;
import android.hardware.SensorEvent;
import android.hardware.SensorEventListener;
import android.hardware.SensorManager;
import android.widget.Toast;

import com.reg.user.reg.StepListener.StepListener;

import java.util.Timer;

/**
 * Created by USER on 16-07-2017.
 */

public class Tab2Fragment extends Fragment implements TextToSpeech.OnInitListener {
    private static final String TAG = "Tab2Fragment";
    private Button BtnStart,BtnStop,BtnPause,BtnFindMe,BtnTorchOn,BtnTorchOff;
    private StepDetector simpleStepDetector;
    private SensorManager sensorManager;
    private Sensor accel;
    private static final String TEXT_NUM_STEPS = "Number of Steps: ";
    private int numSteps;
    private Timer timer;

    private TextToSpeech textToSpeech;

    private Camera camera;
    private Parameters parameters;

    public View onCreateView(LayoutInflater inflater, @Nullable ViewGroup
container, @Nullable Bundle savedInstanceState) {
        View view = inflater.inflate(R.layout.tab2fragment, container, false);

        BtnTorchOn = (Button) view.findViewById(R.id.btn_torch_on);
        BtnTorchOff = (Button) view.findViewById(R.id.btn_torch_off);
        BtnFindMe = (Button) view.findViewById(R.id.find_me);


        textToSpeech= new TextToSpeech(getActivity(), this );
```

```
        camera = Camera.open();
        parameters = camera.getParameters();



        BtnTorchOn.setOnClickListener(new View.OnClickListener() {

            @Override
            public void onClick(View arg0) {
                Toast.makeText(getActivity(), "torch on!",
Toast.LENGTH_SHORT).show();
                textToSpeech.speak("Torch turned on",0,null);
                parameters.setFlashMode(Camera.Parameters.FLASH_MODE_TORCH);
                camera.setParameters(parameters);
                camera.startPreview();

            }



        });

        BtnTorchOff.setOnClickListener(new View.OnClickListener() {

            @Override
            public void onClick(View arg0) {

                textToSpeech.speak("Torch turned off",0,null);

                Toast.makeText(getActivity(), "torch off!",
Toast.LENGTH_SHORT).show();
                parameters.setFlashMode(Camera.Parameters.FLASH_MODE_OFF);
                camera.setParameters(parameters);
                camera.stopPreview();
            }



        });

        BtnFindMe.setOnClickListener(new View.OnClickListener() {

            @Override
            public void onClick(View arg0) {
```

```
                textToSpeech.speak("Finding You",0,null);
                Intent intent= new Intent(getActivity(), Findme.class);
                Toast.makeText(getActivity(), "There you are!",
Toast.LENGTH_SHORT).show();
                startActivity(intent);

            }
        });

        return view;
    }

    public void onInit(int status) {

    }
}
```

Findme map code

```
package com.reg.user.reg;
import android.content.pm.PackageManager;
import android.location.Address;
import android.location.Geocoder;
import android.location.Location;
import android.location.LocationListener;
import android.location.LocationManager;
import android.support.v4.app.ActivityCompat;
import android.support.v4.app.FragmentActivity;
import android.os.Bundle;

import com.google.android.gms.maps.CameraUpdateFactory;
import com.google.android.gms.maps.GoogleMap;
import com.google.android.gms.maps.OnMapReadyCallback;
import com.google.android.gms.maps.SupportMapFragment;
import com.google.android.gms.maps.model.LatLng;
import com.google.android.gms.maps.model.MarkerOptions;

import java.io.IOException;
import java.util.List;

public class Findme extends FragmentActivity implements OnMapReadyCallback {
```

```
    private GoogleMap mMap;

    LocationManager locationManager;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_findme);
        // Obtain the SupportMapFragment and get notified when the map is ready to
be used.
        SupportMapFragment mapFragment = (SupportMapFragment)
getSupportFragmentManager()
                .findFragmentById(R.id.map);
        mapFragment.getMapAsync(this);
        locationManager = (LocationManager) getSystemService(LOCATION_SERVICE);
        if (ActivityCompat.checkSelfPermission(this,
android.Manifest.permission.ACCESS_FINE_LOCATION) !=
PackageManager.PERMISSION_GRANTED && ActivityCompat.checkSelfPermission(this,
android.Manifest.permission.ACCESS_COARSE_LOCATION) !=
PackageManager.PERMISSION_GRANTED) {
            // TODO: Consider calling
            //    ActivityCompat#requestPermissions
            // here to request the missing permissions, and then overriding
            //   public void onRequestPermissionsResult(int requestCode, String[]
permissions,
            //                                          int[] grantResults)
            // to handle the case where the user grants the permission. See the
documentation
            // for ActivityCompat#requestPermissions for more details.
            return;
        }

        if(locationManager.isProviderEnabled(LocationManager.NETWORK_PROVIDER)) {

locationManager.requestLocationUpdates(LocationManager.NETWORK_PROVIDER, 0, 0, new
LocationListener() {
                @Override
                public void onLocationChanged(Location location) {
                    double latitude = location.getLatitude();
                    double longitude = location.getLongitude();
                    LatLng latLng= new LatLng(latitude, longitude);
                    Geocoder geocoder= new Geocoder(getApplicationContext());
                    try {
                        List<Address> addressList=
```

```
geocoder.getFromLocation(latitude, longitude, 1);
                        String str= addressList.get(0).getSubThoroughfare()+",";
                        str += addressList.get(0).getThoroughfare()+",";
                        str += addressList.get(0).getSubLocality()+",";
                        str += addressList.get(0).getLocality()+",";
                        str += addressList.get(0).getAdminArea()+",";
                        str += addressList.get(0).getCountryName()+",";
                        mMap.addMarker(new
MarkerOptions().position(latLng).title(str));
                        mMap.moveCamera(CameraUpdateFactory.newLatLngZoom(latLng,
16f));
                    } catch (IOException e) {
                        e.printStackTrace();
                    }
                }

                @Override
                public void onStatusChanged(String s, int i, Bundle bundle) {

                }

                @Override
                public void onProviderEnabled(String s) {

                }

                @Override
                public void onProviderDisabled(String s) {

                }
            });
        }
        else if(locationManager.isProviderEnabled(LocationManager.GPS_PROVIDER)){
            locationManager.requestLocationUpdates(LocationManager.GPS_PROVIDER, 0,
0, new LocationListener() {
                @Override
                public void onLocationChanged(Location location) {
                    double latitude = location.getLatitude();
                    double longitude = location.getLongitude();
                    LatLng latLng1= new LatLng(latitude, longitude);
                    Geocoder geocoder= new Geocoder(getApplicationContext());
                    try {
                        List<Address> addressList=
geocoder.getFromLocation(latitude, longitude, 1);
                        String str= addressList.get(0).getLocality()+",";
```

```
                        str += addressList.get(0).getCountryName();
                        mMap.addMarker(new
MarkerOptions().position(latLng1).title(str));
                        mMap.moveCamera(CameraUpdateFactory.newLatLngZoom(latLng1,
16f));
                    } catch (IOException e) {
                        e.printStackTrace();
                    }
                }

                @Override
                public void onStatusChanged(String s, int i, Bundle bundle) {

                }

                @Override
                public void onProviderEnabled(String s) {

                }

                @Override
                public void onProviderDisabled(String s) {

                }
            });
        }

    }


    /**
     * Manipulates the map once available.
     * This callback is triggered when the map is ready to be used.
     * This is where we can add markers or lines, add listeners or move the camera.
In this case,
     * we just add a marker near Sydney, Australia.
     * If Google Play services is not installed on the device, the user will be
prompted to install
     * it inside the SupportMapFragment. This method will only be triggered once
the user has
     * installed Google Play services and returned to the app.
     */
    @Override
    public void onMapReady(GoogleMap googleMap) {
        mMap = googleMap;
```

```
        // Add a marker in Sydney and move the camera
        //LatLng sydney = new LatLng(-34, 151);
        //mMap.addMarker(new MarkerOptions().position(sydney).title("Marker in
Sydney"));
        //mMap.moveCamera(CameraUpdateFactory.newLatLngZoom(sydney, 10.2f));
    }
}
```

Music player tab code

```
package com.reg.user.reg;

import android.content.Intent;
import android.media.MediaPlayer;
import android.net.Uri;
import android.os.Bundle;
import android.support.v7.app.AppCompatActivity;
import android.view.View;
import android.widget.Button;
import android.widget.SeekBar;

import java.io.File;
import java.util.ArrayList;

import android.content.Intent;
import android.hardware.SensorEventListener;
import android.media.MediaPlayer;
import android.net.Uri;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.SeekBar;

import java.io.File;
import java.util.ArrayList;

public class Player extends AppCompatActivity implements View.OnClickListener {
    static MediaPlayer mp;
    ArrayList<File> mySongs;
    int position;
```

```
Uri u;
Thread updateSeekBar;

SeekBar sb;
Button btPlay, btNxt, btPrv;
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_player);

    btPlay=(Button) findViewById(R.id.btPlay);
    btPrv=(Button) findViewById(R.id.btPrev);
    btNxt=(Button) findViewById(R.id.btNxt);
    sb= (SeekBar) findViewById(R.id.seekBar);

    btPlay.setOnClickListener(this);
    btPrv.setOnClickListener(this);
    btNxt.setOnClickListener(this);

    updateSeekBar=new Thread(){
        public void run() {
            int totalDuration = mp.getDuration();
            int currentPosition = 0;
            //sb.setMax(totalDuration);
            while (currentPosition < totalDuration) {
                try {
                    sleep(500);
                    currentPosition=mp.getCurrentPosition();
                    sb.setProgress(currentPosition);
                } catch (InterruptedException e) {
                    e.printStackTrace();
                }
            }
        }
    };


    if(mp!=null){
        mp.stop();
        mp.release();
    }

    Intent i= getIntent();
    Bundle b=i.getExtras();
    mySongs= (ArrayList) b.getParcelableArrayList("songlist");
```

```
        position= b.getInt("pos",0);

        u= Uri.parse(mySongs.get(position).toString());
        mp= MediaPlayer.create(getApplicationContext(),u);
        mp.start();
        sb.setMax(mp.getDuration());

        updateSeekBar.start();
        sb.setOnSeekBarChangeListener(new SeekBar.OnSeekBarChangeListener() {
            @Override
            public void onProgressChanged(SeekBar seekBar, int i, boolean b) {

            }

            @Override
            public void onStartTrackingTouch(SeekBar seekBar) {

            }

            @Override
            public void onStopTrackingTouch(SeekBar seekBar) {

                mp.seekTo(seekBar.getProgress());
            }
        });
    }

    public void onClick(View v){
        int id=v.getId();
        switch (id){
            case R.id.btPlay:
                if(mp.isPlaying()){

                    btPlay.setText("Play");
                    mp.pause();
                }
                else {
                    mp.start();
                    btPlay.setText("Pause");
                }
                break;

            case R.id.btNxt:
                mp.stop();
                mp.release();
```

```
                position=(position+1)%mySongs.size();
                u= Uri.parse(mySongs.get(position).toString());
                mp= MediaPlayer.create(getApplicationContext(),u);
                mp.start();
                sb.setMax(mp.getDuration());
                break;

            case R.id.btPrev:
                mp.stop();
                mp.release();
                position=(position-1<0)? mySongs.size()-1: position-1;
                u= Uri.parse(mySongs.get(position).toString());
                mp= MediaPlayer.create(getApplicationContext(),u);
                mp.start();
                sb.setMax(mp.getDuration());
                break;


        }
    }
}
```

Other supporting java codes

```
package com.reg.user.reg;

/**
 * Created by USER on 17-07-2017.
 */

import android.support.v4.app.Fragment;
import android.support.v4.app.FragmentActivity;
import android.support.v4.app.FragmentManager;
import android.support.v4.app.FragmentPagerAdapter;

import java.util.ArrayList;
import java.util.List;

/**
 * Created by USER on 16-07-2017.
 */
```

```
public class SectionsPageAdapter extends FragmentPagerAdapter {


    private final List<Fragment> mFragmentList = new ArrayList<>();
    private final List<String> mFragmentTitleList = new ArrayList<>();

    public void addFragment(Fragment fragment, String title) {

        mFragmentList.add(fragment);
        mFragmentTitleList.add(title);

    }

    public SectionsPageAdapter(FragmentManager fm) {
        super(fm);
    }

    @Override
    public CharSequence getPageTitle(int position) {
        return mFragmentTitleList.get(position);
    }

    @Override
    public Fragment getItem(int position){
        return mFragmentList.get(position);
    }

    @Override
    public int getCount() {
        return mFragmentList.size();
    }
}
```

simplesteplistener

```
package com.reg.user.reg;

import android.hardware.Camera;
import android.hardware.Sensor;
import android.hardware.SensorEvent;
import android.hardware.SensorEventListener;
```

```
import android.hardware.SensorManager;
import android.os.Bundle;
import android.os.SystemClock;
import android.speech.tts.TextToSpeech;
import android.support.v7.app.AppCompatActivity;
import android.view.View;
import android.widget.Button;
import android.widget.Chronometer;
import android.widget.TextView;

import java.util.Timer;


import android.content.Context;
import android.hardware.Camera;
import android.hardware.Camera.Parameters;
import android.hardware.Sensor;
import android.hardware.SensorEvent;
import android.hardware.SensorEventListener;
import android.hardware.SensorManager;
import android.os.SystemClock;
import android.speech.tts.TextToSpeech;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.Button;

import android.widget.Chronometer;
import android.widget.TextView;

import com.reg.user.reg.StepListener.StepListener;

import java.util.Timer;

public class SimpleStepListener extends AppCompatActivity implements
SensorEventListener, StepListener, TextToSpeech.OnInitListener {
    private TextView TvSteps,TvDist;
    private Button BtnStart,BtnStop,BtnPause,BtnResume,BtnTorchOn,BtnTorchOff;
    private long timeWhenStopped = 0;
    private boolean stopClicked;
    private Chronometer chronometer;
```

```
private StepDetector simpleStepDetector;
private SensorManager sensorManager;
private Sensor accel;
private static final String TEXT_NUM_STEPS = "Number of Steps: ";
private int numSteps;
private Timer timer;

private TextToSpeech textToSpeech;

private Camera camera;
private Parameters parameters;

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_simple_step_listener);


    // Get an instance of the SensorManager
    sensorManager = (SensorManager) getSystemService(SENSOR_SERVICE);
    accel = sensorManager.getDefaultSensor(Sensor.TYPE_ACCELEROMETER);
    simpleStepDetector = new StepDetector();
    simpleStepDetector.registerListener(this);

    TvSteps = (TextView) findViewById(R.id.tv_steps);
    TvDist = (TextView) findViewById(R.id.tv_dist);


    BtnStart = (Button) findViewById(R.id.btn_start);
    BtnStop = (Button) findViewById(R.id.btn_stop);
    BtnPause = (Button) findViewById(R.id.btn_pause);
    BtnResume = (Button) findViewById(R.id.btn_resume);
    BtnTorchOn = (Button) findViewById(R.id.btn_torch_on);
    BtnTorchOff = (Button) findViewById(R.id.btn_torch_off);

    chronometer = (Chronometer) findViewById(R.id.chronometer);
    chronometer.setText("");

    textToSpeech= new TextToSpeech(this, this);

    camera = Camera.open();
    parameters = camera.getParameters();


    BtnStart.setOnClickListener(new View.OnClickListener() {
```

```
            @Override
            public void onClick(View arg0) {

                textToSpeech.speak("start walking",0,null);

                numSteps = 0;
                sensorManager.registerListener(SimpleStepListener.this, accel,
SensorManager.SENSOR_DELAY_FASTEST);

                chronometer.setBase(SystemClock.elapsedRealtime() );

                chronometer.start();
                stopClicked = false;




            }
        });


        BtnStop.setOnClickListener(new View.OnClickListener() {

            @Override
            public void onClick(View arg0) {

                textToSpeech.speak("stop walking",0,null);
                sensorManager.unregisterListener(SimpleStepListener.this);
                if (!stopClicked) {

                    chronometer.stop();
                    stopClicked = true;

                }
            }
        });

        BtnPause.setOnClickListener(new View.OnClickListener() {

            @Override
            public void onClick(View arg0) {


                textToSpeech.speak("paused walking",0,null);
```

```
                sensorManager.unregisterListener(SimpleStepListener.this);
                if (!stopClicked) {

                    timeWhenStopped = chronometer.getBase() -
SystemClock.elapsedRealtime();
                    int seconds = (int) timeWhenStopped / 1000;

                    chronometer.stop();
                    stopClicked = true;

                }


            }
        });

        BtnResume.setOnClickListener(new View.OnClickListener() {

            @Override
            public void onClick(View arg0) {

                textToSpeech.speak("resumed walking",0,null);

                sensorManager.registerListener(SimpleStepListener.this, accel,
SensorManager.SENSOR_DELAY_FASTEST);

                chronometer.setBase(SystemClock.elapsedRealtime() +
timeWhenStopped);

                chronometer.start();
                stopClicked = false;

            }



        });

        BtnTorchOn.setOnClickListener(new View.OnClickListener() {

            @Override
            public void onClick(View arg0) {

                textToSpeech.speak("Torch turned on",0,null);
```

```
                parameters.setFlashMode(Parameters.FLASH_MODE_TORCH);
                camera.setParameters(parameters);
                camera.startPreview();



            }


        });

        BtnTorchOff.setOnClickListener(new View.OnClickListener() {

            @Override
            public void onClick(View arg0) {

                textToSpeech.speak("Torch off",0,null);


                parameters.setFlashMode(Parameters.FLASH_MODE_OFF);
                camera.setParameters(parameters);
                camera.stopPreview();
            }


        });



    }


    @Override
    public void onAccuracyChanged(Sensor sensor, int accuracy) {
    }

    @Override
    public void onSensorChanged(SensorEvent event) {
        if (event.sensor.getType() == Sensor.TYPE_ACCELEROMETER) {
```

```
            simpleStepDetector.updateAccel(
                    event.timestamp, event.values[0], event.values[1],
event.values[2]);
        }
    }

    @Override
    public void step(long timeNs) {
        numSteps++;
        TvSteps.setText("Steps : " +numSteps);
        TvDist.setText("Distance : "+((int) (0.9*numSteps)));

    }

    @Override
    public void onInit(int status) {

    }
}
```

StepDetector

```
package com.reg.user.reg;

/**
 * Created by USER on 16-07-2017.
 */
import com.reg.user.reg.StepListener.StepListener;

/**
 * Created by HP on 7/12/2017.
 */

public class StepDetector {

    private static final int ACCEL_RING_SIZE = 50;
    private static final int VEL_RING_SIZE = 10;

    // change this threshold according to your sensitivity preferences
    private static final float STEP_THRESHOLD = 50f;

    private static final int STEP_DELAY_NS = 250000000;
```

```
    private int accelRingCounter = 0;
    private float[] accelRingX = new float[ACCEL_RING_SIZE];
    private float[] accelRingY = new float[ACCEL_RING_SIZE];
    private float[] accelRingZ = new float[ACCEL_RING_SIZE];
    private int velRingCounter = 0;
    private float[] velRing = new float[VEL_RING_SIZE];
    private long lastStepTimeNs = 0;
    private float oldVelocityEstimate = 0;

    private StepListener listener;

    public void registerListener(StepListener listener) {
        this.listener = listener;
    }


    public void updateAccel(long timeNs, float x, float y, float z) {
        float[] currentAccel = new float[3];
        currentAccel[0] = x;
        currentAccel[1] = y;
        currentAccel[2] = z;

        // First step is to update our guess of where the global z vector is.
        accelRingCounter++;
        accelRingX[accelRingCounter % ACCEL_RING_SIZE] = currentAccel[0];
        accelRingY[accelRingCounter % ACCEL_RING_SIZE] = currentAccel[1];
        accelRingZ[accelRingCounter % ACCEL_RING_SIZE] = currentAccel[2];

        float[] worldZ = new float[3];
        worldZ[0] = SensorFilter.sum(accelRingX) / Math.min(accelRingCounter,
ACCEL_RING_SIZE);
        worldZ[1] = SensorFilter.sum(accelRingY) / Math.min(accelRingCounter,
ACCEL_RING_SIZE);
        worldZ[2] = SensorFilter.sum(accelRingZ) / Math.min(accelRingCounter,
ACCEL_RING_SIZE);

        float normalization_factor = SensorFilter.norm(worldZ);

        worldZ[0] = worldZ[0] / normalization_factor;
        worldZ[1] = worldZ[1] / normalization_factor;
        worldZ[2] = worldZ[2] / normalization_factor;

        float currentZ = SensorFilter.dot(worldZ, currentAccel) -
normalization_factor;
        velRingCounter++;
```

```
        velRing[velRingCounter % VEL_RING_SIZE] = currentZ;

        float velocityEstimate = SensorFilter.sum(velRing);

        if (velocityEstimate > STEP_THRESHOLD && oldVelocityEstimate <=
STEP_THRESHOLD
                && (timeNs - lastStepTimeNs > STEP_DELAY_NS)) {
            listener.step(timeNs);
            lastStepTimeNs = timeNs;
        }
        oldVelocityEstimate = velocityEstimate;
    }
}
```

SensorFilter

```
package com.reg.user.reg;

/**
 * Created by USER on 16-07-2017.
 */

public class SensorFilter {

    private SensorFilter() {
    }

    public static float sum(float[] array) {
        float retval = 0;
        for (int i = 0; i < array.length; i++) {
            retval += array[i];
        }
        return retval;
    }

    public static float[] cross(float[] arrayA, float[] arrayB) {
        float[] retArray = new float[3];
        retArray[0] = arrayA[1] * arrayB[2] - arrayA[2] * arrayB[1];
        retArray[1] = arrayA[2] * arrayB[0] - arrayA[0] * arrayB[2];
        retArray[2] = arrayA[0] * arrayB[1] - arrayA[1] * arrayB[0];
        return retArray;
    }
```

```
    public static float norm(float[] array) {
        float retval = 0;
        for (int i = 0; i < array.length; i++) {
            retval += array[i] * array[i];
        }
        return (float) Math.sqrt(retval);
    }


    public static float dot(float[] a, float[] b) {
        float retval = a[0] * b[0] + a[1] * b[1] + a[2] * b[2];
        return retval;
    }

    public static float[] normalize(float[] a) {
        float[] retval = new float[a.length];
        float norm = norm(a);
        for (int i = 0; i < a.length; i++) {
            retval[i] = a[i] / norm;
        }
        return retval;
    }

}
```

Certificate

This is to certify that Mr/Ms [*fill in your name*] of [*fill in the name of the institution where you are studying*], registration number: [*fill in your university registration number*], has successfully completed a project on [*fill your project title*] using [*mention the technology on which your project was developed i.e., JEE/.Net/Java/Oracle Forms/ PHP/ Android/ Networking*] under the guidance of Mr/Ms/Mrs [*mention your faculty's name*].

--- ----------

-

[*Name of your faculty*]

Globsyn Finishing School

Certificate

This is to certify that Mr/Ms [*fill in your name*] of [*fill in the name of the institution where you are studying*], registration number: [*fill in your university registration number*], has successfully completed a project on [*fill your project title*] using [*mention the technology on which your project was developed i.e., JEE/.Net/Java/Oracle Forms/ PHP/ Android/ Networking*] under the guidance of Mr/Ms/Mrs [*mention your faculty's name*].

--- ----------

-

[*Name of your faculty*]

Globsyn Finishing School