

**DOCUMENTATION ON MAD PROJECT**



# NAME- Meet Patel

ENROLLMENT- **18012011074**

SUBJECT- **MOBILE APPLICATION DEVELOPMENT**

BATCH- **MAD2**

TITLE OF PROJECT**- PEDOMETER**

 **INRTODUCTION:-**

* The purpose of this project is to implement a Java based pedometer application that will allow users without an internet connection to track the fitness record.

* The development of this project centered on the daily fitness record all about your walking according to it.It calculates your average workout. We can also compare between daily workout.

* + **PURPOSE:**
* This project is to create a pedometer application with a user itself to keep a daily track of workout.
* The main purpose of this project is to keep aware of your daily energy as well as strength.
* This project can play an important role in day-to-day life to keep us aware regarding our fitness.

* + **CODE:-**

# AndroidManifest.xml-

<?**xml version="1.0" encoding="utf-8"**?>

<**manifest**

**package="com.example.pedometer"**

**xmlns:android="http://schemas.android.com/apk/res/android" xmlns:tools="http://schemas.android.com/tools" android:versionCode="1511" android:versionName="1.5.11"**>

<**uses-sdk**

**android:minSdkVersion="19" android:targetSdkVersion="26"**/>

<**uses-feature**

**android:name="android.hardware.sensor.stepcounter"**

**android:required="true"**/>

<**uses-permission android:name="android.permission.RECEIVE\_BOOT\_COMPLETED"**/>

<**uses-permission android:name="android.permission.WRITE\_EXTERNAL\_STORAGE"**/>

<**uses-permission android:name="android.permission.FOREGROUND\_SERVICE"**/>

<**uses-permission android:name="android.permission.WAKE\_LOCK"**/>

<**application**

**android:allowBackup="true" android:icon="@mipmap/ic\_launcher" android:label="@string/app\_name"**

**android:theme="@android:style/Theme.DeviceDefault.Light.DarkActionBar" tools:replace="label"**>

<**activity**

**android:name=".ui.Activity\_Main" android:label="@string/app\_name" android:launchMode="singleTask"**>

<**intent-filter**>

<**action android:name="android.intent.action.MAIN"**/>

<**category android:name="android.intent.category.LAUNCHER"**/>

</**intent-filter**>

</**activity**>

<**receiver android:name=".BootReceiver"**>

<**intent-filter**>

<**action android:name="android.intent.action.BOOT\_COMPLETED"**/>

</**intent-filter**>

</**receiver**>

<**receiver android:name=".AppUpdatedReceiver"**>

<**intent-filter**>

<**action android:name="android.intent.action.MY\_PACKAGE\_REPLACED"**/>

</**intent-filter**>

</**receiver**> <**service**

**android:name=".SensorListener"**/>

<**service**

**android:name=".widget.DashClock"**

**android:permission="com.google.android.apps.dashclock.permission.READ\_EXTENSION\_DATA"**>

<**intent-filter**>

<**action android:name="com.google.android.apps.dashclock.Extension"**/>

</**intent-filter**> <**meta-data**

**android:name="protocolVersion"**

**android:value="2"**/>

<**meta-data**

**android:name="worldReadable" android:value="true"**/>

<**meta-data**

**android:name="description"**

**android:value="Displays the steps taken today"**/>

</**service**>

<**activity**

**android:name=".widget.WidgetConfig"**

**android:theme="@android:style/Theme.Holo.Dialog.NoActionBar"**>

<**intent-filter**>

<**action android:name="android.appwidget.action.APPWIDGET\_CONFIGURE"**/>

</**intent-filter**>

</**activity**>

<**receiver android:name=".widget.Widget"**>

<**intent-filter**>

<**action android:name="android.appwidget.action.APPWIDGET\_UPDATE"**/>

</**intent-filter**> <**meta-data**

**android:name="android.appwidget.provider" android:resource="@xml/widget"**/>

</**receiver**>

<**service android:name=".widget.WidgetUpdateService" android:exported="true" android:permission="android.permission.BIND\_JOB\_SERVICE"**/> </**application**>

</**manifest**>

# JAVA FILE-

#  com.example.pedometer-

#  ui-

# 1) Dialog\_Split.java:-

**package** com.example.pedometer.ui; **import** android.app.Dialog; **import** android.content.Context; **import** android.content.SharedPreferences; **import** android.view.View;

**import** android.view.View.OnClickListener; **import** android.widget.Button; **import** android.widget.TextView; **import** com.example.pedometer.R; **abstract class** Dialog\_Split { **private static boolean** split\_active;

**public static** Dialog getDialog(**final** Context c, **final int** totalSteps) { **final** Dialog d = **new** Dialog(c);

d.setTitle(R.string.**split\_count**);

d.setContentView(R.layout.**dialog\_split**); **final** SharedPreferences prefs =

c.getSharedPreferences(**"pedometer"**, Context.**MODE\_MULTI\_PROCESS**); **long** split\_date = prefs.getLong(**"split\_date"**, -1); **int** split\_steps = prefs.getInt(**"split\_steps"**, totalSteps);

((TextView) d.findViewById(R.id.**steps**))

.setText(Fragment\_Overview.**formatter**.format(totalSteps - split\_steps)); **float** stepsize = prefs.getFloat(**"stepsize\_value"**,

Fragment\_Settings.**DEFAULT\_STEP\_SIZE**);

**float** distance = (totalSteps - split\_steps) \* stepsize; **if** (prefs.getString(**"stepsize\_unit"**,

Fragment\_Settings.**DEFAULT\_STEP\_UNIT**).equals(**"cm"**)) { distance /= 100000;

((TextView) d.findViewById(R.id.**distanceunit**)).setText(**"km"**);

} **else** {

distance /= 5280;

((TextView) d.findViewById(R.id.**distanceunit**)).setText(**"mi"**);

}

((TextView) d.findViewById(R.id.**distance**))

.setText(Fragment\_Overview.**formatter**.format(distance)); ((TextView) d.findViewById(R.id.**date**)).setText(c.getString(R.string.**since**, java.text.DateFormat.getDateTimeInstance().format(split\_date))); **final** View started = d.findViewById(R.id.**started**); **final** View stopped = d.findViewById(R.id.**stopped**);

split\_active = split\_date > 0;

started.setVisibility(split\_active ? View.**VISIBLE** : View.**GONE**); stopped.setVisibility(split\_active ? View.**GONE** : View.**VISIBLE**); **final** Button startstop = (Button) d.findViewById(R.id.**start**); startstop.setText(split\_active ? R.string.**stop** : R.string.**start**); startstop.setOnClickListener(**new** OnClickListener() {

@Override

**public void** onClick(**final** View v) {

**if** (!split\_active) {

prefs.edit().putLong(**"split\_date"**, System.currentTimeMillis())

.putInt(**"split\_steps"**, totalSteps).apply(); split\_active = **true**; d.dismiss();

} **else** {

started.setVisibility(View.**GONE**); stopped.setVisibility(View.**VISIBLE**);

prefs.edit().remove(**"split\_date"**).remove(**"split\_steps"**).apply(); split\_active = **false**;

}

startstop.setText(split\_active ? R.string.**stop** : R.string.**start**);

}

});

d.findViewById(R.id.**close**).setOnClickListener(**new** OnClickListener() { @Override

**public void** onClick(**final** View v) { d.dismiss();

} }); **return** d;

}

}

# 2) Dialog\_Statistics.java:-

**package** com.example.pedometer.ui; **import** android.app.Dialog; **import** android.content.Context; **import** android.util.Pair; **import** android.view.View; **import** android.view.View.OnClickListener; **import** android.view.Window; **import** android.widget.TextView; **import** java.util.Calendar; **import** java.util.Date;

**import** com.example.pedometer.Database; **import** com.example.pedometer.R; **import** com.example.pedometer.util.Util; **abstract class** Dialog\_Statistics {

**public static** Dialog getDialog(**final** Context c, **int** since\_boot) {

**final** Dialog d = **new** Dialog(c);

d.requestWindowFeature(Window.**FEATURE\_NO\_TITLE**);

d.setContentView(R.layout.**statistics**);

d.findViewById(R.id.**close**).setOnClickListener(**new** OnClickListener() { @Override

**public void** onClick(View v) { d.dismiss();

}

});

Database db = Database.getInstance(c);

Pair<Date, Integer> record = db.getRecordData(); Calendar date = Calendar.getInstance();

date.setTimeInMillis(Util.getToday());

**int** daysThisMonth = date.get(Calendar.**DAY\_OF\_MONTH**);

date.add(Calendar.**DATE**, -6);

**int** thisWeek = db.getSteps(date.getTimeInMillis(), System.currentTimeMillis()) + since\_boot; date.setTimeInMillis(Util.getToday()); date.set(Calendar.**DAY\_OF\_MONTH**, 1);

**int** thisMonth = db.getSteps(date.getTimeInMillis(), System.currentTimeMillis()) + since\_boot;

((TextView) d.findViewById(R.id.**record**)).setText(

Fragment\_Overview.**formatter**.format(record.**second**) + **" @ "**

+ java.text.DateFormat.getDateInstance().format(record.**first**));

((TextView)

d.findViewById(R.id.**totalthisweek**)).setText(Fragment\_Overview.**formatter**.format(thisWeek));

((TextView)

d.findViewById(R.id.**totalthismonth**)).setText(Fragment\_Overview.**formatter**.format(thisMonth)

);

((TextView)

d.findViewById(R.id.**averagethisweek**)).setText(Fragment\_Overview.**formatter**.format(thisWee k / 7)); ((TextView)

d.findViewById(R.id.**averagethismonth**)).setText(Fragment\_Overview.**formatter**.format( thisMo nth / daysThisMonth));

db.close(); **return** d;

}

}

# 3) Fragment\_Overview.java:-

**package** com.example.pedometer.ui; **import** android.app.AlertDialog; **import** android.app.Fragment; **import** android.content.Context; **import** android.content.DialogInterface; **import** android.content.Intent; **import** android.content.SharedPreferences; **import** android.graphics.Color; **import** android.hardware.Sensor; **import** android.hardware.SensorEvent; **import** android.hardware.SensorEventListener; **import** android.hardware.SensorManager;

**import** android.os.Build; **import** android.os.Bundle; **import** android.util.Pair; **import** android.view.LayoutInflater; **import** android.view.Menu; **import** android.view.MenuInflater; **import** android.view.MenuItem;

**import** android.view.View; **import** android.view.View.OnClickListener; **import** android.view.ViewGroup; **import** android.widget.TextView; **import** org.eazegraph.lib.charts.BarChart; **import** org.eazegraph.lib.charts.PieChart; **import** org.eazegraph.lib.models.BarModel; **import** org.eazegraph.lib.models.PieModel; **import** java.text.NumberFormat; **import** java.text.SimpleDateFormat;

**import** java.util.Date; **import** java.util.List; **import** java.util.Locale; **import** com.example.pedometer.BuildConfig; **import** com.example.pedometer.Database; **import** com.example.pedometer.R; **import** com.example.pedometer.SensorListener; **import** com.example.pedometer.util.API26Wrapper; **import** com.example.pedometer.util.Logger; **import** com.example.pedometer.util.Util;

**public class** Fragment\_Overview **extends** Fragment **implements** SensorEventListener { **private** TextView **stepsView**, **totalView**, **averageView**; **private** PieModel **sliceGoal**, **sliceCurrent**; **private** PieChart **pg**;

**private int todayOffset**, **total\_start**, **goal**, **since\_boot**, **total\_days**;

**public final static** NumberFormat **formatter** = NumberFormat.getInstance(Locale.getDefault()); **private boolean showSteps** = **true**;

@Override

**public void** onCreate(**final** Bundle savedInstanceState) { **super**.onCreate(savedInstanceState); setHasOptionsMenu(**true**);

**if** (Build.VERSION.**SDK\_INT** >= 26) {

API26Wrapper.startForegroundService(getActivity(), **new** Intent(getActivity(), SensorListener.**class**));

} **else** {

getActivity().startService(**new** Intent(getActivity(), SensorListener.**class**));

}

}

@Override

**public** View onCreateView(**final** LayoutInflater inflater, **final** ViewGroup container, **final** Bundle savedInstanceState) {

**final** View v = inflater.inflate(R.layout.**fragment\_overview**, **null**); **stepsView** = (TextView) v.findViewById(R.id.**steps**); **totalView** = (TextView) v.findViewById(R.id.**total**); **averageView** = (TextView) v.findViewById(R.id.**average**); **pg** = (PieChart) v.findViewById(R.id.**graph**);

**sliceCurrent** = **new** PieModel(**""**, 0, Color.parseColor(**"#99CC00"**)); **pg**.addPieSlice(**sliceCurrent**);

**sliceGoal** = **new** PieModel(**""**, Fragment\_Settings.**DEFAULT\_GOAL**,

Color.parseColor(**"#CC0000"**)); **pg**.addPieSlice(**sliceGoal**);

**pg**.setOnClickListener(**new** OnClickListener() {

@Override

**public void** onClick(**final** View view) { **showSteps** = !**showSteps**;

stepsDistanceChanged();

}

});

**pg**.setDrawValueInPie(**false**); **pg**.setUsePieRotation(**true**); **pg**.startAnimation();

**return** v;

}

@Override **public void** onResume() { **super**.onResume();

getActivity().getActionBar().setDisplayHomeAsUpEnabled(**false**);

Database db = Database.getInstance(getActivity()); **if** (BuildConfig.**DEBUG**) db.logState(); **todayOffset** = db.getSteps(Util.getToday()); SharedPreferences prefs =

getActivity().getSharedPreferences(**"pedometer"**, Context.**MODE\_PRIVATE**); **goal** = prefs.getInt(**"goal"**, Fragment\_Settings.**DEFAULT\_GOAL**);

**since\_boot** = db.getCurrentSteps();

**int** pauseDifference = **since\_boot** - prefs.getInt(**"pauseCount"**, **since\_boot**); SensorManager sm = (SensorManager)

getActivity().getSystemService(Context.**SENSOR\_SERVICE**);

Sensor sensor = sm.getDefaultSensor(Sensor.**TYPE\_STEP\_COUNTER**); **if** (sensor == **null**) {

**new** AlertDialog.Builder(getActivity()).setTitle(R.string.**no\_sensor**)

.setMessage(R.string.**no\_sensor\_explain**)

.setOnDismissListener(**new** DialogInterface.OnDismissListener() {

@Override

**public void** onDismiss(**final** DialogInterface dialogInterface) { getActivity().finish(); }

}).setNeutralButton(android.R.string.**ok**, **new** DialogInterface.OnClickListener() {

@Override

**public void** onClick(**final** DialogInterface dialogInterface, **int** i) { dialogInterface.dismiss();

}

}).create().show();

} **else** {

sm.registerListener(**this**, sensor, SensorManager.**SENSOR\_DELAY\_UI**, 0);

}

**since\_boot** -= pauseDifference; **total\_start** = db.getTotalWithoutToday(); **total\_days** = db.getDays(); db.close();

stepsDistanceChanged();

}

**private void** stepsDistanceChanged() {

**if** (**showSteps**) {

((TextView) getView().findViewById(R.id.**unit**)).setText(getString(R.string.**steps**));

} **else** {

String unit = getActivity().getSharedPreferences(**"pedometer"**,

Context.**MODE\_PRIVATE**).getString(**"stepsize\_unit"**, Fragment\_Settings.**DEFAULT\_STEP\_UNIT**); **if** (unit.equals(**"cm"**)) { unit = **"km"**; } **else** {

unit = **"mi"**;

}

((TextView) getView().findViewById(R.id.**unit**)).setText(unit);

} updatePie();

updateBars();

}

@Override **public void** onPause() {

**super**.onPause();

**try** {

SensorManager sm =

(SensorManager) getActivity().getSystemService(Context.**SENSOR\_SERVICE**); sm.unregisterListener(**this**); } **catch** (Exception e) {

**if** (BuildConfig.**DEBUG**) Logger.log(e);

}

Database db = Database.getInstance(getActivity()); db.saveCurrentSteps(**since\_boot**);

db.close(); }

@Override

**public void** onCreateOptionsMenu(**final** Menu menu, **final** MenuInflater inflater) { inflater.inflate(R.menu.**main**, menu);

}

@Override

**public boolean** onOptionsItemSelected(**final** MenuItem item) { **switch** (item.getItemId()) { **case** R.id.**action\_split\_count**: Dialog\_Split.getDialog(getActivity(),

**total\_start** + Math.max(**todayOffset** + **since\_boot**, 0)).show(); **return true**; **default**:

**return** ((Activity\_Main) getActivity()).optionsItemSelected(item);

}

}

@Override

**public void** onAccuracyChanged(**final** Sensor sensor, **int** accuracy) {

}

@Override

**public void** onSensorChanged(**final** SensorEvent event) { **if** (BuildConfig.**DEBUG**) Logger.log(

**"UI - sensorChanged | todayOffset: "** + **todayOffset** + **" since boot: "** + event.**values**[0]);

**if** (event.**values**[0] > Integer.**MAX\_VALUE** || event.**values**[0] == 0) { **return**;

}

**if** (**todayOffset** == Integer.**MIN\_VALUE**) { **todayOffset** = -(**int**) event.**values**[0];

Database db = Database.getInstance(getActivity()); db.insertNewDay(Util.getToday(), (**int**) event.**values**[0]); db.close();

}

**since\_boot** = (**int**) event.**values**[0];

updatePie();

}

**private void** updatePie() {

**if** (BuildConfig.**DEBUG**) Logger.log(**"UI - update steps: "** + **since\_boot**); **int** steps\_today = Math.max(**todayOffset** + **since\_boot**, 0); **sliceCurrent**.setValue(steps\_today); **if** (**goal** - steps\_today > 0) { **if** (**pg**.getData().size() == 1) { **pg**.addPieSlice(**sliceGoal**);

}

**sliceGoal**.setValue(**goal** - steps\_today);

} **else** { **pg**.clearChart();

**pg**.addPieSlice(**sliceCurrent**);

} **pg**.update(); **if** (**showSteps**) {

**stepsView**.setText(**formatter**.format(steps\_today)); **totalView**.setText(**formatter**.format(**total\_start** + steps\_today));

**averageView**.setText(**formatter**.format((**total\_start** + steps\_today) / **total\_days**));

} **else** {

SharedPreferences prefs =

getActivity().getSharedPreferences(**"pedometer"**, Context.**MODE\_PRIVATE**); **float** stepsize = prefs.getFloat(**"stepsize\_value"**,

Fragment\_Settings.**DEFAULT\_STEP\_SIZE**);

**float** distance\_today = steps\_today \* stepsize; **float** distance\_total = (**total\_start** + steps\_today) \* stepsize;

**if** (prefs.getString(**"stepsize\_unit"**, Fragment\_Settings.**DEFAULT\_STEP\_UNIT**)

.equals(**"cm"**)) { distance\_today /= 100000; distance\_total /= 100000;

} **else** {

distance\_today /= 5280;

distance\_total /= 5280;

}

**stepsView**.setText(**formatter**.format(distance\_today)); **totalView**.setText(**formatter**.format(distance\_total));

**averageView**.setText(**formatter**.format(distance\_total / **total\_days**));

}

}

**private void** updateBars() {

SimpleDateFormat df = **new** SimpleDateFormat(**"E"**, Locale.getDefault()); BarChart barChart = (BarChart) getView().findViewById(R.id.**bargraph**); **if** (barChart.getData().size() > 0) barChart.clearChart(); **int** steps;

**float** distance, stepsize = Fragment\_Settings.**DEFAULT\_STEP\_SIZE**; **boolean** stepsize\_cm = **true**; **if** (!**showSteps**) {

SharedPreferences prefs =

getActivity().getSharedPreferences(**"pedometer"**, Context.**MODE\_PRIVATE**); stepsize = prefs.getFloat(**"stepsize\_value"**, Fragment\_Settings.**DEFAULT\_STEP\_SIZE**); stepsize\_cm = prefs.getString(**"stepsize\_unit"**,

Fragment\_Settings.**DEFAULT\_STEP\_UNIT**)

.equals(**"cm"**);

}

barChart.setShowDecimal(!**showSteps**);

BarModel bm;

Database db = Database.getInstance(getActivity()); List<Pair<Long, Integer>> last = db.getLastEntries(8); db.close(); **for** (**int** i = last.size() - 1; i > 0; i--) { Pair<Long, Integer> current = last.get(i); steps = current.**second**; **if** (steps > 0) {

bm = **new** BarModel(df.format(**new** Date(current.**first**)), 0,

steps > **goal** ? Color.parseColor(**"#99CC00"**) : Color.parseColor(**"#0099cc"**)); **if** (**showSteps**) { bm.setValue(steps);

} **else** {

distance = steps \* stepsize; **if** (stepsize\_cm) { distance /= 100000;

} **else** {

distance /= 5280;

}

distance = Math.round(distance \* 1000) / 1000f; // 3 decimals bm.setValue(distance);

}

barChart.addBar(bm);

}

}

**if** (barChart.getData().size() > 0) {

barChart.setOnClickListener(**new** OnClickListener() {

@Override

**public void** onClick(**final** View v) {

Dialog\_Statistics.getDialog(getActivity(), **since\_boot**).show();

}

});

barChart.startAnimation();

} **else** {

barChart.setVisibility(View.**GONE**);

}

}

}

# 4) Fragment\_Settinge.java:-

**package** com.example.pedometer.ui;

**import** android.Manifest;

**import** android.app.AlertDialog;

**import** android.app.Dialog;

**import** android.app.NotificationManager;

**import** android.content.Context;

**import** android.content.DialogInterface;

**import** android.content.DialogInterface.OnClickListener; **import** android.content.Intent;

**import** android.content.SharedPreferences;

**import** android.content.pm.PackageManager;

**import** android.database.Cursor;

**import** android.os.Build;

**import** android.os.Bundle;

**import** android.os.Environment;

**import** android.preference.Preference;

**import** android.preference.Preference.OnPreferenceClickListener; **import** android.preference.PreferenceFragment;

**import** android.view.Menu;

**import** android.view.MenuInflater;

**import** android.view.MenuItem;

**import** android.view.View;

**import** android.view.WindowManager;

**import** android.widget.EditText;

**import** android.widget.NumberPicker;

**import** android.widget.RadioGroup;

**import** android.widget.Toast;

**import** java.io.BufferedReader;

**import** java.io.BufferedWriter;

**import** java.io.File;

**import** java.io.FileReader;

**import** java.io.FileWriter;

**import** java.io.IOException;

**import** java.util.Locale;

**import** com.example.pedometer.Database;

**import** com.example.pedometer.R;

**import** com.example.pedometer.SensorListener;

**import** com.example.pedometer.util.API23Wrapper; **import** com.example.pedometer.util.API26Wrapper; **import** com.example.pedometer.util.PlaySettingsWrapper;

**public class** Fragment\_Settings **extends** PreferenceFragment **implements** OnPreferenceClickListener {

**final static int DEFAULT\_GOAL** = 10000;

**final static float DEFAULT\_STEP\_SIZE** = Locale.getDefault() == Locale.**US** ? 2.5f : 75f; **final static** String **DEFAULT\_STEP\_UNIT** = Locale.getDefault() == Locale.**US** ? **"ft"** :

**"cm"**;

@Override

**public void** onCreate(**final** Bundle savedInstanceState) { **super**.onCreate(savedInstanceState); addPreferencesFromResource(R.xml.**settings**);

**final** SharedPreferences prefs =

getActivity().getSharedPreferences(**"pedometer"**, Context.**MODE\_PRIVATE**); findPreference(**"import"**).setOnPreferenceClickListener(**this**); findPreference(**"export"**).setOnPreferenceClickListener(**this**);

**if** (Build.VERSION.**SDK\_INT** >= 26) {

findPreference(**"notification"**).setOnPreferenceClickListener(**this**);

} **else** {

findPreference(**"notification"**)

.setOnPreferenceChangeListener(**new** Preference.OnPreferenceChangeListener() {

@Override

**public boolean** onPreferenceChange(**final** Preference preference, **final** Object newValue) {

prefs.edit().putBoolean(**"notification"**, (Boolean) newValue).apply(); NotificationManager manager = (NotificationManager) getActivity() .getSystemService(Context.**NOTIFICATION\_SERVICE**); **if** ((Boolean) newValue) {

manager.notify(SensorListener.**NOTIFICATION\_ID**,

SensorListener.getNotification(getActivity()));

} **else** {

manager.cancel(SensorListener.**NOTIFICATION\_ID**);

}

**return true**;

}

});

}

Preference account = findPreference(**"account"**);

PlaySettingsWrapper

.setupAccountSetting(account, savedInstanceState, (Activity\_Main) getActivity());

Preference goal = findPreference(**"goal"**); goal.setOnPreferenceClickListener(**this**);

goal.setSummary(getString(R.string.**goal\_summary**, prefs.getInt(**"goal"**,

**DEFAULT\_GOAL**)));

Preference stepsize = findPreference(**"stepsize"**); stepsize.setOnPreferenceClickListener(**this**); stepsize.setSummary(getString(R.string.**step\_size\_summary**, prefs.getFloat(**"stepsize\_value"**, **DEFAULT\_STEP\_SIZE**), prefs.getString(**"stepsize\_unit"**, **DEFAULT\_STEP\_UNIT**))); setHasOptionsMenu(**true**);

}

@Override

**public void** onSaveInstanceState(**final** Bundle outState) {

**super**.onSaveInstanceState(outState);

PlaySettingsWrapper.onSavedInstance(outState, (Activity\_Main) getActivity());

}

@Override **public void** onResume() {

**super**.onResume();

getActivity().getActionBar().setDisplayHomeAsUpEnabled(**true**);

**if** (Build.VERSION.**SDK\_INT** >= 26) { // notification settings might have changed API26Wrapper.startForegroundService(getActivity(), **new** Intent(getActivity(), SensorListener.**class**));

}

}

@Override

**public void** onCreateOptionsMenu(**final** Menu menu, **final** MenuInflater inflater) { inflater.inflate(R.menu.**main**, menu);

}

@Override

**public void** onPrepareOptionsMenu(**final** Menu menu) { **super**.onPrepareOptionsMenu(menu); menu.findItem(R.id.**action\_settings**).setVisible(**false**); menu.findItem(R.id.**action\_split\_count**).setVisible(**false**);

}

@Override **public boolean** onOptionsItemSelected(**final** MenuItem item) { **return** ((Activity\_Main) getActivity()).optionsItemSelected(item);

}

@Override

**public boolean** onPreferenceClick(**final** Preference preference) {

AlertDialog.Builder builder; View v;

**final** SharedPreferences prefs =

getActivity().getSharedPreferences(**"pedometer"**, Context.**MODE\_PRIVATE**); **switch** (preference.getTitleRes()) { **case** R.string.**goal**: builder = **new** AlertDialog.Builder(getActivity()); **final** NumberPicker np = **new** NumberPicker(getActivity()); np.setMinValue(1); np.setMaxValue(100000);

np.setValue(prefs.getInt(**"goal"**, 10000));

builder.setView(np);

builder.setTitle(R.string.**set\_goal**);

builder.setPositiveButton(android.R.string.**ok**, **new** OnClickListener() {

@Override

**public void** onClick(DialogInterface dialog, **int** which) { np.clearFocus();

prefs.edit().putInt(**"goal"**, np.getValue()).commit();

preference.setSummary(getString(R.string.**goal\_summary**, np.getValue())); dialog.dismiss();

getActivity().startService(**new** Intent(getActivity(), SensorListener.**class**) .putExtra(**"updateNotificationState"**, **true**));

}

});

builder.setNegativeButton(android.R.string.**cancel**, **new** OnClickListener() {

@Override

**public void** onClick(DialogInterface dialog, **int** which) { dialog.dismiss();

}

});

Dialog dialog = builder.create(); dialog.getWindow().setSoftInputMode(

WindowManager.LayoutParams.**SOFT\_INPUT\_STATE\_ALWAYS\_VISIBLE**); dialog.show(); **break**; **case** R.string.**step\_size**:

builder = **new** AlertDialog.Builder(getActivity());

v = getActivity().getLayoutInflater().inflate(R.layout.**stepsize**, **null**); **final** RadioGroup unit = (RadioGroup) v.findViewById(R.id.**unit**); **final** EditText value = (EditText) v.findViewById(R.id.**value**); unit.check( prefs.getString(**"stepsize\_unit"**, **DEFAULT\_STEP\_UNIT**).equals(**"cm"**) ? R.id.**cm** :

R.id.**ft**);

value.setText(String.valueOf(prefs.getFloat(**"stepsize\_value"**,

**DEFAULT\_STEP\_SIZE**))); builder.setView(v);

builder.setTitle(R.string.**set\_step\_size**);

builder.setPositiveButton(android.R.string.**ok**, **new** OnClickListener() {

@Override

**public void** onClick(DialogInterface dialog, **int** which) { **try** {

prefs.edit().putFloat(**"stepsize\_value"**, Float.valueOf(value.getText().toString()))

.putString(**"stepsize\_unit"**,

unit.getCheckedRadioButtonId() == R.id.**cm** ? **"cm"** : **"ft"**)

.apply();

preference.setSummary(getString(R.string.**step\_size\_summary**, Float.valueOf(value.getText().toString()),

unit.getCheckedRadioButtonId() == R.id.**cm** ? **"cm"** : **"ft"**));

} **catch** (NumberFormatException nfe) {

nfe.printStackTrace();

}

dialog.dismiss();

}

});

builder.setNegativeButton(android.R.string.**cancel**, **new** OnClickListener() {

@Override

**public void** onClick(DialogInterface dialog, **int** which) { dialog.dismiss();

}

});

builder.create().show(); **break**; **case** R.string.**import\_title**: **case** R.string.**export\_title**:

**if** (hasWriteExternalPermission()) {

**if** (preference.getTitleRes() == R.string.**import\_title**) { importCsv(); } **else** { exportCsv();

}

} **else if** (Build.VERSION.**SDK\_INT** >= 23) {

API23Wrapper.requestPermission(getActivity(),

**new** String[]{Manifest.permission.**WRITE\_EXTERNAL\_STORAGE**});

} **else** {

Toast.makeText(getActivity(), R.string.**permission\_external\_storage**,

Toast.**LENGTH\_SHORT**).show();

} **break**; **case** R.string.**notification\_settings**:

API26Wrapper.launchNotificationSettings(getActivity()); **break**;

}

**return false**;

}

**private boolean** hasWriteExternalPermission() { **return** getActivity().getPackageManager()

.checkPermission(Manifest.permission.**WRITE\_EXTERNAL\_STORAGE**, getActivity().getPackageName()) ==

PackageManager.**PERMISSION\_GRANTED**;

}

**private void** exportCsv() { **if** (Environment.getExternalStorageState().equals(Environment.**MEDIA\_MOUNTED**)) { **final** File f = **new** File(Environment.getExternalStorageDirectory(), **"Pedometer.csv"**); **if** (f.exists()) {

**new** AlertDialog.Builder(getActivity()).setMessage(R.string.**file\_already\_exists**)

.setPositiveButton(android.R.string.**ok**, **new** OnClickListener() {

@Override

**public void** onClick(DialogInterface dialog, **int** which) { dialog.dismiss(); writeToFile(f);

}

}).setNegativeButton(android.R.string.**cancel**, **new** OnClickListener() {

@Override

**public void** onClick(DialogInterface dialog, **int** which) { dialog.dismiss();

}

}).create().show();

} **else** {

writeToFile(f);

}

} **else** {

**new** AlertDialog.Builder(getActivity())

.setMessage(R.string.**error\_external\_storage\_not\_available**)

.setPositiveButton(android.R.string.**ok**, **new** OnClickListener() {

@Override

**public void** onClick(DialogInterface dialog, **int** which) { dialog.dismiss();

}

}).create().show();

}

}

**private void** importCsv() { **if** (Environment.getExternalStorageState().equals(Environment.**MEDIA\_MOUNTED**)) { File f = **new** File(Environment.getExternalStorageDirectory(), **"Pedometer.csv"**); **if** (!f.exists() || !f.canRead()) {

**new** AlertDialog.Builder(getActivity())

.setMessage(getString(R.string.**file\_cant\_read**, f.getAbsolutePath()))

.setPositiveButton(android.R.string.**ok**, **new** OnClickListener() {

@Override

**public void** onClick(DialogInterface dialog, **int** which) { dialog.dismiss();

}

}).create().show(); **return**;

}

Database db = Database.getInstance(getActivity());

String line;

String[] data;

**int** ignored = 0, inserted = 0, overwritten = 0;

BufferedReader in; **try** {

in = **new** BufferedReader(**new** FileReader(f)); **while** ((line = in.readLine()) != **null**) { data = line.split(**";"**);

**try** {

**if** (db.insertDayFromBackup(Long.valueOf(data[0]),

Integer.valueOf(data[1]))) {

inserted++; } **else** {

overwritten++;

}

} **catch** (Exception nfe) {

ignored++;

} } in.close();

} **catch** (IOException e) {

**new** AlertDialog.Builder(getActivity())

.setMessage(getString(R.string.**error\_file**, e.getMessage()))

.setPositiveButton(android.R.string.**ok**, **new** OnClickListener() {

@Override

**public void** onClick(DialogInterface dialog, **int** which) { dialog.dismiss();

}

}).create().show(); e.printStackTrace();

**return**; } **finally** { db.close(); }

String message = getString(R.string.**entries\_imported**, inserted + overwritten); **if** (overwritten > 0)

message += **"\n\n"** + getString(R.string.**entries\_overwritten**, overwritten); **if** (ignored > 0) message += **"\n\n"** + getString(R.string.**entries\_ignored**, ignored); **new** AlertDialog.Builder(getActivity()).setMessage(message) .setPositiveButton(android.R.string.**ok**, **new** OnClickListener() {

@Override

**public void** onClick(DialogInterface dialog, **int** which) { dialog.dismiss();

}

}).create().show();

} **else** {

**new** AlertDialog.Builder(getActivity())

.setMessage(R.string.**error\_external\_storage\_not\_available**)

.setPositiveButton(android.R.string.**ok**, **new** OnClickListener() {

@Override

**public void** onClick(DialogInterface dialog, **int** which) { dialog.dismiss();

}

}).create().show();

}

}

**private void** writeToFile(**final** File f) {

BufferedWriter out; **try** {

f.createNewFile(); out = **new** BufferedWriter(**new** FileWriter(f));

} **catch** (IOException e) {

**new** AlertDialog.Builder(getActivity())

.setMessage(getString(R.string.**error\_file**, e.getMessage()))

.setPositiveButton(android.R.string.**ok**, **new** OnClickListener() {

@Override

**public void** onClick(DialogInterface dialog, **int** which) { dialog.dismiss();

}

}).create().show(); e.printStackTrace();

**return**;

}

Database db = Database.getInstance(getActivity());

Cursor c =

db.query(**new** String[]{**"date"**, **"steps"**}, **"date > 0"**, **null**, **null**, **null**, **"date"**, **null**); **try** {

**if** (c != **null** && c.moveToFirst()) { **while** (!c.isAfterLast()) {

out.append(c.getString(0)).append(**";"**)

.append(String.valueOf(Math.max(0, c.getInt(1)))).append(**"\n"**); c.moveToNext();

} } out.flush();

out.close();

} **catch** (IOException e) {

**new** AlertDialog.Builder(getActivity())

.setMessage(getString(R.string.**error\_file**, e.getMessage()))

.setPositiveButton(android.R.string.**ok**, **new** OnClickListener() {

@Override

**public void** onClick(DialogInterface dialog, **int** which) { dialog.dismiss();

}

}).create().show(); e.printStackTrace();

**return**; } **finally** {

**if** (c != **null**) c.close(); db.close();

}

**new** AlertDialog.Builder(getActivity())

.setMessage(getString(R.string.**data\_saved**, f.getAbsolutePath()))

.setPositiveButton(android.R.string.ok, **new** OnClickListener() {

@Override

**public void** onClick(DialogInterface dialog, **int** which) { dialog.dismiss();

}

}).create().show();

}

}

#  util-

# 1) API23Wrapper.java:-

**package** com.example.pedometer.util; **import** android.annotation.TargetApi; **import** android.app.Activity; **import** android.app.AlarmManager; **import** android.app.PendingIntent; **import** android.os.Build;

@TargetApi(Build.VERSION\_CODES.**M**)

**public class** API23Wrapper {

**public static void** requestPermission(**final** Activity a, **final** String[] permissions) { a.requestPermissions(permissions, 42);

}

**public static void** setAlarmWhileIdle(AlarmManager am, **int** type, **long** time,

PendingIntent intent) {

am.setAndAllowWhileIdle(type, time, intent);

}

}

# 2) API26Wrapper.java:-

**package** com.example.pedometer.util; **import** android.annotation.TargetApi; **import** android.app.Notification; **import** android.app.NotificationChannel; **import** android.app.NotificationManager; **import** android.content.ActivityNotFoundException; **import** android.content.Context; **import** android.content.Intent; **import** android.os.Build; **import** android.provider.Settings;

**import** android.widget.Toast;

@TargetApi(Build.VERSION\_CODES.**O**)

**public class** API26Wrapper { **public final static** String **NOTIFICATION\_CHANNEL\_ID** = **"Notification"**; **public static void** startForegroundService(**final** Context context, **final** Intent intent) { context.startForegroundService(intent);

}

**public static** Notification.Builder getNotificationBuilder(**final** Context context) {

NotificationManager manager =

(NotificationManager)

context.getSystemService(Context.**NOTIFICATION\_SERVICE**);

NotificationChannel channel =

**new** NotificationChannel(**NOTIFICATION\_CHANNEL\_ID**,

**NOTIFICATION\_CHANNEL\_ID**,

NotificationManager.**IMPORTANCE\_NONE**); channel.setImportance(NotificationManager.**IMPORTANCE\_MIN**); channel.enableLights(**false**); channel.enableVibration(**false**); channel.setBypassDnd(**false**); channel.setSound(**null**, **null**);

manager.createNotificationChannel(channel);

Notification.Builder builder = **new** Notification.Builder(context,

**NOTIFICATION\_CHANNEL\_ID**);

**return** builder;

}

**public static void** launchNotificationSettings(**final** Context context) {

Intent intent = **new** Intent(Settings.**ACTION\_CHANNEL\_NOTIFICATION\_SETTINGS**); intent.putExtra(Settings.**EXTRA\_CHANNEL\_ID**, **NOTIFICATION\_CHANNEL\_ID**); intent.putExtra(Settings.**EXTRA\_APP\_PACKAGE**, context.getPackageName()); **try** {

context.startActivity(intent);

} **catch** (ActivityNotFoundException e) {

Toast.makeText(context,

**"Settings not found - please search for the notification settings in the Android settings manually"**,

Toast.**LENGTH\_LONG**).show();

}

}

}

# 3) Logger.java:-

**package** com.example.pedometer.util; **import** android.database.Cursor; **import** android.os.Environment;

**import** java.io.File; **import** java.io.FileWriter; **import** java.io.IOException; **import** java.util.Date;

**import** com.example.pedometer.BuildConfig;

**public abstract class** Logger { **private static** FileWriter fw; **private static final** Date **date** = **new** Date(); **private final static** String **APP** = **"Pedometer"**; **public static void** log(Throwable ex) {

log(ex.getMessage());

**for** (StackTraceElement ste : ex.getStackTrace()) { log(ste.toString());

}

}

**public static void** log(**final** Cursor c) { **if** (!BuildConfig.**DEBUG**) **return**; c.moveToFirst(); String title = **""**; **for** (**int** i = 0; i < c.getColumnCount(); i++)

title += c.getColumnName(i) + **"\t| "**; log(title);

**while** (!c.isAfterLast()) {

title = **""**;

**for** (**int** i = 0; i < c.getColumnCount(); i++)

title += c.getString(i) + **"\t| "**; log(title);

c.moveToNext();

}

}

@SuppressWarnings(**"deprecation"**) **public static void** log(String msg) { **if** (!BuildConfig.**DEBUG**) **return**;

android.util.Log.d(**APP**, msg); **try** {

**if** (fw == **null**) {

fw = **new** FileWriter(**new** File(

Environment.getExternalStorageDirectory().toString() + **"/"** + **APP** + **".txt"**), **true**);

}

**date**.setTime(System.currentTimeMillis()); fw.write(**date**.toLocaleString() + **" - "** + msg + **"\n"**); fw.flush();

} **catch** (IOException e) { e.printStackTrace();

}

}

}

**protected void** finalize() **throws** Throwable { **try** {

**if** (fw != **null**) fw.close();

} **finally** { **super**.finalize();

}

}

}

# 4) Util.java:-

**package** com.example.pedometer.util;

**import** java.util.Calendar; **public abstract class** Util { **public static long** getToday() { Calendar c = Calendar.getInstance();

c.setTimeInMillis(System.currentTimeMillis());

c.set(Calendar.**HOUR\_OF\_DAY**, 0);

c.set(Calendar.**MINUTE**, 0);

c.set(Calendar.**SECOND**, 0);

c.set(Calendar.**MILLISECOND**, 0);

**return** c.getTimeInMillis();

}

**public static long** getTomorrow() { Calendar c = Calendar.getInstance();

c.setTimeInMillis(System.currentTimeMillis());

c.set(Calendar.**HOUR\_OF\_DAY**, 0);

c.set(Calendar.**MINUTE**, 0);

c.set(Calendar.**SECOND**, 1);

c.set(Calendar.**MILLISECOND**, 0);

c.add(Calendar.**DATE**, 1);

**return** c.getTimeInMillis();

}

}

#  Widget-

# 1) DashClock.java:-

**package** com.example.pedometer.widget; **import** android.content.Intent; **import** com.google.android.apps.dashclock.api.DashClockExtension; **import** com.google.android.apps.dashclock.api.ExtensionData; **import** com.example.pedometer.Database; **import** com.example.pedometer.R; **import** com.example.pedometer.ui.Activity\_Main; **import** com.example.pedometer.ui.Fragment\_Overview; **import** com.example.pedometer.util.Util;

**public class** DashClock **extends** DashClockExtension {

@Override **protected void** onUpdateData(**int** reason) { ExtensionData data = **new** ExtensionData(); Database db = Database.getInstance(**this**); **int** steps = Math.max(db.getCurrentSteps() + db.getSteps(Util.getToday()), 0); data.visible(**true**).status(Fragment\_Overview.**formatter**.format(steps))

.icon(R.drawable.**ic\_dashclock**)

.clickIntent(**new** Intent(DashClock.**this**, Activity\_Main.**class**)); db.close(); publishUpdate(data);

}

}

# 2) Widget.java:-

**package** com.example.pedometer.widget; **import** android.app.PendingIntent; **import** android.appwidget.AppWidgetManager; **import** android.appwidget.AppWidgetProvider; **import** android.content.Context; **import** android.content.Intent;

**import** android.content.SharedPreferences; **import** android.graphics.Color; **import** android.widget.RemoteViews; **import** com.example.pedometer.R; **import** com.example.pedometer.ui.Activity\_Main; **public class** Widget **extends** AppWidgetProvider {

@Override

**public void** onUpdate(Context context, AppWidgetManager appWidgetManager, **int**[] appWidgetIds) {

WidgetUpdateService.enqueueUpdate(context);

}

**static** RemoteViews updateWidget(**final int** appWidgetId, **final** Context context, **final int** steps) {

**final** SharedPreferences prefs =

context.getSharedPreferences(**"Widgets"**, Context.**MODE\_PRIVATE**);

**final** PendingIntent pendingIntent = PendingIntent

.getActivity(context, appWidgetId, **new** Intent(context, Activity\_Main.**class**), 0); **final** RemoteViews views = **new** RemoteViews(context.getPackageName(), R.layout.**widget**); views.setOnClickPendingIntent(R.id.**widget**, pendingIntent);

views.setTextColor(R.id.**widgetsteps**, prefs.getInt(**"color\_"** + appWidgetId, Color.**WHITE**));

views.setTextViewText(R.id.**widgetsteps**, String.valueOf(steps)); views.setTextColor(R.id.**widgettext**, prefs.getInt(**"color\_"** + appWidgetId, Color.**WHITE**));

views.setInt(R.id.**widget**, **"setBackgroundColor"**,

prefs.getInt(**"background\_"** + appWidgetId, Color.**TRANSPARENT**)); **return** views;

}

}

# 3) WidgetConfig:-

**package** com.example.pedometer.widget;

**import** android.app.Activity;

**import** android.appwidget.AppWidgetManager;

**import** android.content.Context; **import** android.content.Intent; **import** android.graphics.Color; **import** android.os.Bundle; **import** android.view.View; **import** android.view.View.OnClickListener; **import** com.example.lib.colorpicker.ColorPickerDialog; **import** com.example.lib.colorpicker.ColorPreviewButton; **import** com.example.pedometer.R;

**public class** WidgetConfig **extends** Activity **implements** OnClickListener { **private static int** widgetId;

@Override

**protected void** onPause() { **super**.onPause();

WidgetUpdateService.enqueueUpdate(**this**);

}

@Override

**protected void** onCreate(**final** Bundle savedInstanceState) { **super**.onCreate(savedInstanceState); **final** Intent intent = getIntent(); **final** Bundle extras = intent.getExtras(); **if** (extras != **null**) {

setContentView(R.layout.**widgetconfig**);

ColorPreviewButton textcolor = (ColorPreviewButton) findViewById(R.id.**textcolor**);

textcolor.setOnClickListener(**this**); textcolor.setColor(Color.**WHITE**);

ColorPreviewButton bgcolor = (ColorPreviewButton) findViewById(R.id.**bgcolor**); bgcolor.setOnClickListener(**this**); bgcolor.setColor(Color.**TRANSPARENT**);

widgetId = extras.getInt(AppWidgetManager.**EXTRA\_APPWIDGET\_ID**,

AppWidgetManager.**INVALID\_APPWIDGET\_ID**); **final** Intent resultValue = **new** Intent();

resultValue.putExtra(AppWidgetManager.**EXTRA\_APPWIDGET\_ID**, widgetId); setResult(**RESULT\_OK**, resultValue);

} **else** { finish();

}

}

@Override

**public void** onClick(**final** View v) {

ColorPickerDialog dialog = **new** ColorPickerDialog(**this**, (findViewById(v.getId()).getTag() != **null**) ?

(Integer) findViewById(v.getId()).getTag() : -1);

dialog.setHexValueEnabled(**true**); dialog.setAlphaSliderVisible(**true**);

dialog.setOnColorChangedListener(**new** ColorPickerDialog.OnColorChangedListener() {

@Override **public void** onColorChanged(**int** color) { ((ColorPreviewButton) v).setColor(color); v.setTag(color);

getSharedPreferences(**"Widgets"**, Context.**MODE\_PRIVATE**).edit()

.putInt((v.getId() == R.id.**bgcolor** ? **"background\_"** : **"color\_"**) + widgetId, color).apply();

}

});

dialog.show();

}

}

# 4) WidgetUpdateService.java:-

**package** com.example.pedometer.widget; **import** android.appwidget.AppWidgetManager; **import** android.content.ComponentName; **import** android.content.Context; **import** android.content.Intent; **import** android.support.annotation.NonNull; **import** android.support.v4.app.JobIntentService; **import** com.example.pedometer.Database; **import** com.example.pedometer.util.Util;

**public class** WidgetUpdateService **extends** JobIntentService {

**private static final int JOB\_ID** = 42; **public static void** enqueueUpdate(Context context) {

enqueueWork(context, WidgetUpdateService.**class**, **JOB\_ID**, **new** Intent());

}

@Override

**protected void** onHandleWork(@NonNull Intent intent) { Database db = Database.getInstance(**this**);

**int** steps = Math.max(db.getCurrentSteps() + db.getSteps(Util.getToday()), 0); db.close();

**final** AppWidgetManager appWidgetManager = AppWidgetManager.getInstance(**this**); **int**[] appWidgetIds =

appWidgetManager.getAppWidgetIds(**new** ComponentName(**this**, Widget.**class**)); **for** (**int** appWidgetId : appWidgetIds) {

appWidgetManager

.updateAppWidget(appWidgetId, Widget.updateWidget(appWidgetId, **this**, steps));

}

}

}

#  AppUpdatedReceiver.java:-

**package** com.example.pedometer; **import** android.content.BroadcastReceiver; **import** android.content.Context; **import** android.content.Intent; **import** android.os.Build; **import** com.example.pedometer.util.API26Wrapper; **import** com.example.pedometer.util.Logger;

**public class** AppUpdatedReceiver **extends** BroadcastReceiver {

@Override **public void** onReceive(**final** Context context, **final** Intent intent) { **if** (BuildConfig.**DEBUG**) Logger.log(**"app updated"**);

**if** (Build.VERSION.**SDK\_INT** >= 26) {

API26Wrapper.startForegroundService(context, **new** Intent(context,

SensorListener.**class**));

} **else** {

context.startService(**new** Intent(context, SensorListener.**class**));

}

}

}

#  BootReceiver:-

**package** com.example.pedometer; **import** android.content.BroadcastReceiver; **import** android.content.Context; **import** android.content.Intent; **import** android.content.SharedPreferences; **import** android.os.Build; **import** com.example.pedometer.util.API26Wrapper; **import** com.example.pedometer.util.Logger;

**public class** BootReceiver **extends** BroadcastReceiver {

@Override

**public void** onReceive(**final** Context context, **final** Intent intent) {

**if** (BuildConfig.**DEBUG**) Logger.log(**"booted"**);

SharedPreferences prefs = context.getSharedPreferences(**"pedometer"**,

Context.**MODE\_PRIVATE**);

Database db = Database.getInstance(context); **if** (!prefs.getBoolean(**"correctShutdown"**, **false**)) { **if** (BuildConfig.**DEBUG**) Logger.log(**"Incorrect shutdown"**);

**int** steps = Math.max(0, db.getCurrentSteps());

**if** (BuildConfig.**DEBUG**) Logger.log(**"Trying to recover "** + steps + **" steps"**); db.addToLastEntry(steps);

}

db.removeNegativeEntries(); db.saveCurrentSteps(0);

db.close();

prefs.edit().remove(**"correctShutdown"**).apply();

**if** (Build.VERSION.**SDK\_INT** >= 26) {

API26Wrapper.startForegroundService(context, **new** Intent(context,

SensorListener.**class**));

} **else** {

context.startService(**new** Intent(context, SensorListener.**class**));

}

}

}

#  Database:-

**package** com.example.pedometer; **import** android.content.ContentValues; **import** android.content.Context; **import** android.database.Cursor; **import** android.database.sqlite.SQLiteDatabase; **import** android.database.sqlite.SQLiteOpenHelper;

**import** android.util.Pair; **import** java.util.ArrayList; **import** java.util.Date; **import** java.util.List; **import** java.util.concurrent.atomic.AtomicInteger; **import** com.example.pedometer.util.Logger; **import** com.example.pedometer.util.Util; **public class** Database **extends** SQLiteOpenHelper { **private final static** String **DB\_NAME** = **"steps"**; **private final static int DB\_VERSION** = 2;

**private static** Database instance;

**private static final** AtomicInteger **openCounter** = **new** AtomicInteger(); **private** Database(**final** Context context) {

**super**(context, **DB\_NAME**, **null**, **DB\_VERSION**);

}

**public static synchronized** Database getInstance(**final** Context c) { **if** (instance == **null**) {

instance = **new** Database(c.getApplicationContext());

}

**openCounter**.incrementAndGet();

**return** instance;

}

@Override **public void** close() {

**if** (**openCounter**.decrementAndGet() == 0) { **super**.close();

}

}

@Override

**public void** onCreate(**final** SQLiteDatabase db) {

db.execSQL(**"CREATE TABLE "** + **DB\_NAME** + **" (date INTEGER, steps**

**INTEGER)"**);

}

@Override

**public void** onUpgrade(**final** SQLiteDatabase db, **int** oldVersion, **int** newVersion) { **if** (oldVersion == 1) {

db.execSQL(**"CREATE TABLE "** + **DB\_NAME** + **"2 (date INTEGER, steps INTEGER)"**);

db.execSQL(**"INSERT INTO "** + **DB\_NAME** + **"2 (date, steps) SELECT date, steps FROM "** +

**DB\_NAME**);

db.execSQL(**"DROP TABLE "** + **DB\_NAME**);

db.execSQL(**"ALTER TABLE "** + **DB\_NAME** + **"2 RENAME TO "** + **DB\_NAME** +

**""**);

}

}

**public** Cursor query(**final** String[] columns, **final** String selection,

**final** String[] selectionArgs, **final** String groupBy, **final** String having,

**final** String orderBy, **final** String limit) {

**return** getReadableDatabase()

.query(**DB\_NAME**, columns, selection, selectionArgs, groupBy, having, orderBy, limit);

}

**public void** insertNewDay(**long** date, **int** steps) {

getWritableDatabase().beginTransaction(); **try** {

Cursor c = getReadableDatabase().query(**DB\_NAME**, **new** String[]{**"date"**}, **"date =**

**?"**,

**new** String[]{String.valueOf(date)}, **null**, **null**, **null**); **if** (c.getCount() == 0 && steps >= 0) {

addToLastEntry(steps);

ContentValues values = **new** ContentValues();

values.put(**"date"**, date); values.put(**"steps"**, -steps);

getWritableDatabase().insert(**DB\_NAME**, **null**, values);

}

c.close();

**if** (BuildConfig.**DEBUG**) {

Logger.log(**"insertDay "** + date + **" / "** + steps); logState();

}

getWritableDatabase().setTransactionSuccessful();

} **finally** {

getWritableDatabase().endTransaction();

}

}

**public void** addToLastEntry(**int** steps) {

getWritableDatabase().execSQL(**"UPDATE "** + **DB\_NAME** + **" SET steps = steps + "** + steps +

**" WHERE date = (SELECT MAX(date) FROM "** + **DB\_NAME** + **")"**);

}

**public boolean** insertDayFromBackup(**long** date, **int** steps) { getWritableDatabase().beginTransaction(); **boolean** newEntryCreated = **false**;

**try** {

ContentValues values = **new** ContentValues(); values.put(**"steps"**, steps);

**int** updatedRows = getWritableDatabase()

.update(**DB\_NAME**, values, **"date = ?"**, **new** String[]{String.valueOf(date)}); **if** (updatedRows == 0) { values.put(**"date"**, date);

getWritableDatabase().insert(**DB\_NAME**, **null**, values);

newEntryCreated = **true**;

}

getWritableDatabase().setTransactionSuccessful();

} **finally** {

getWritableDatabase().endTransaction();

}

**return** newEntryCreated;

}

**public void** logState() { **if** (BuildConfig.**DEBUG**) { Cursor c = getReadableDatabase()

.query(**DB\_NAME**, **null**, **null**, **null**, **null**, **null**, **"date DESC"**, **"5"**); Logger.log(c); c.close();

}

}

**public int** getTotalWithoutToday() {

Cursor c = getReadableDatabase()

.query(**DB\_NAME**, **new** String[]{**"SUM(steps)"**}, **"steps > 0 AND date > 0 AND date < ?"**,

**new** String[]{String.valueOf(Util.getToday())}, **null**, **null**, **null**); c.moveToFirst(); **int** re = c.getInt(0); c.close(); **return** re;

}

**public int** getRecord() {

Cursor c = getReadableDatabase()

.query(**DB\_NAME**, **new** String[]{**"MAX(steps)"**}, **"date > 0"**, **null**, **null**, **null**, **null**); c.moveToFirst(); **int** re = c.getInt(0); c.close(); **return** re;

}

**public** Pair<Date, Integer> getRecordData() {

Cursor c = getReadableDatabase()

.query(**DB\_NAME**, **new** String[]{**"date, steps"**}, **"date > 0"**, **null**, **null**, **null**, **"steps DESC"**, **"1"**); c.moveToFirst();

Pair<Date, Integer> p = **new** Pair<Date, Integer>(**new** Date(c.getLong(0)), c.getInt(1)); c.close(); **return** p;

}

**public int** getSteps(**final long** date) {

Cursor c = getReadableDatabase().query(**DB\_NAME**, **new** String[]{**"steps"**}, **"date = ?"**, **new** String[]{String.valueOf(date)}, **null**, **null**, **null**); c.moveToFirst();

**int** re;

**if** (c.getCount() == 0) re = Integer.**MIN\_VALUE**; **else** re = c.getInt(0); c.close(); **return** re;

}

**public** List<Pair<Long, Integer>> getLastEntries(**int** num) {

Cursor c = getReadableDatabase()

.query(**DB\_NAME**, **new** String[]{**"date"**, **"steps"**}, **"date > 0"**, **null**, **null**, **null**,

**"date DESC"**, String.valueOf(num));

**int** max = c.getCount();

List<Pair<Long, Integer>> result = **new** ArrayList<>(max); **if** (c.moveToFirst()) {

**do** {

result.add(**new** Pair<>(c.getLong(0), c.getInt(1)));

} **while** (c.moveToNext());

}

**return** result;

}

**public int** getSteps(**final long** start, **final long** end) {

Cursor c = getReadableDatabase()

.query(**DB\_NAME**, **new** String[]{**"SUM(steps)"**}, **"date >= ? AND date <= ?"**, **new** String[]{String.valueOf(start), String.valueOf(end)}, **null**, **null**, **null**); **int** re;

**if** (c.getCount() == 0) {

re = 0;

} **else** {

c.moveToFirst(); re = c.getInt(0);

}

c.close(); **return** re;

}

**void** removeNegativeEntries() {

getWritableDatabase().delete(**DB\_NAME**, **"steps < ?"**, **new** String[]{**"0"**});

}

**public void** removeInvalidEntries() {

getWritableDatabase().delete(**DB\_NAME**, **"steps >= ?"**, **new** String[]{**"200000"**});

}

**public int** getDaysWithoutToday() {

Cursor c = getReadableDatabase()

.query(**DB\_NAME**, **new** String[]{**"COUNT(\*)"**}, **"steps > ? AND date < ? AND date > 0"**,

**new** String[]{String.valueOf(0), String.valueOf(Util.getToday())}, **null**, **null**, **null**); c.moveToFirst(); **int** re = c.getInt(0); c.close();

**return** re < 0 ? 0 : re;

}

**public int** getDays() { // todays is not counted yet

**int** re = **this**.getDaysWithoutToday() + 1; **return** re;

}

**public void** saveCurrentSteps(**int** steps) { ContentValues values = **new** ContentValues();

values.put(**"steps"**, steps);

**if** (getWritableDatabase().update(**DB\_NAME**, values, **"date = -1"**, **null**) == 0) { values.put(**"date"**, -1);

getWritableDatabase().insert(**DB\_NAME**, **null**, values);

}

**if** (BuildConfig.**DEBUG**) {

Logger.log(**"saving steps in db: "** + steps);

}

}

**public int** getCurrentSteps() {

**int** re = getSteps(-1);

**return** re == Integer.**MIN\_VALUE** ? 0 : re;

}

}

#  SensorListener:-

**package** com.example.pedometer; **import** android.app.AlarmManager; **import** android.app.Notification; **import** android.app.NotificationManager; **import** android.app.PendingIntent; **import** android.app.Service;

**import** android.content.BroadcastReceiver; **import** android.content.Context; **import** android.content.Intent; **import** android.content.IntentFilter; **import** android.content.SharedPreferences; **import** android.hardware.Sensor; **import** android.hardware.SensorEvent; **import** android.hardware.SensorEventListener; **import** android.hardware.SensorManager;

**import** android.os.Build; **import** android.os.IBinder; **import** java.text.NumberFormat; **import** java.util.Date; **import** java.util.Locale; **import** com.example.pedometer.ui.Activity\_Main; **import** com.example.pedometer.util.API23Wrapper; **import** com.example.pedometer.util.API26Wrapper; **import** com.example.pedometer.util.Logger; **import** com.example.pedometer.util.Util;

**import** com.example.pedometer.widget.WidgetUpdateService;

**public class** SensorListener **extends** Service **implements** SensorEventListener { **public final static int NOTIFICATION\_ID** = 1; **private final static long MICROSECONDS\_IN\_ONE\_MINUTE** = 60000000; **private final static long SAVE\_OFFSET\_TIME** = AlarmManager.**INTERVAL\_HOUR**;

**private final static int SAVE\_OFFSET\_STEPS** = 500; **private static int** steps; **private static int** lastSaveSteps; **private static long** lastSaveTime;

**private final** BroadcastReceiver **shutdownReceiver** = **new** ShutdownRecevier();

@Override

**public void** onAccuracyChanged(**final** Sensor sensor, **int** accuracy) { **if** (BuildConfig.**DEBUG**) Logger.log(sensor.getName() + **" accuracy changed: "** + accuracy);

}

@Override **public void** onSensorChanged(**final** SensorEvent event) { **if** (event.**values**[0] > Integer.**MAX\_VALUE**) {

**if** (BuildConfig.**DEBUG**) Logger.log(**"probably not a real value: "** + event.**values**[0]); **return**; } **else** {

steps = (**int**) event.**values**[0];

updateIfNecessary();

}

}

**private boolean** updateIfNecessary() {

**if** (steps > lastSaveSteps + **SAVE\_OFFSET\_STEPS** ||

(steps > 0 && System.currentTimeMillis() > lastSaveTime +

**SAVE\_OFFSET\_TIME**)) {

**if** (BuildConfig.**DEBUG**) Logger.log(

**"saving steps: steps="** + steps + **" lastSave="** + lastSaveSteps +

**" lastSaveTime="** + **new** Date(lastSaveTime));

Database db = Database.getInstance(**this**);

**if** (db.getSteps(Util.getToday()) == Integer.**MIN\_VALUE**) {

**int** pauseDifference = steps -

getSharedPreferences(**"pedometer"**, Context.**MODE\_PRIVATE**)

.getInt(**"pauseCount"**, steps);

db.insertNewDay(Util.getToday(), steps - pauseDifference); **if** (pauseDifference > 0) {

getSharedPreferences(**"pedometer"**,

Context.**MODE\_PRIVATE**).edit()

.putInt(**"pauseCount"**, steps).commit();

}

}

db.saveCurrentSteps(steps);

db.close(); lastSaveSteps = steps;

lastSaveTime = System.currentTimeMillis(); showNotification();

WidgetUpdateService.enqueueUpdate(**this**); **return true**; } **else** { **return false**;

}

}

**private void** showNotification() { **if** (Build.VERSION.**SDK\_INT** >= 26) {

startForeground(**NOTIFICATION\_ID**, getNotification(**this**)); } **else if** (getSharedPreferences(**"pedometer"**, Context.**MODE\_PRIVATE**)

.getBoolean(**"notification"**, **true**)) {

((NotificationManager) getSystemService(Context.**NOTIFICATION\_SERVICE**))

.notify(**NOTIFICATION\_ID**, getNotification(**this**));

}

}

@Override

**public** IBinder onBind(**final** Intent intent) {

**return null**;

}

@Override

**public int** onStartCommand(**final** Intent intent, **int** flags, **int** startId) { reRegisterSensor(); registerBroadcastReceiver(); **if** (!updateIfNecessary()) {

showNotification();

}

**long** nextUpdate = Math.min(Util.getTomorrow(),

System.currentTimeMillis() + AlarmManager.**INTERVAL\_HOUR**); **if** (BuildConfig.**DEBUG**) Logger.log(**"next update: "** + **new**

Date(nextUpdate).toLocaleString());

AlarmManager am = (AlarmManager)

getApplicationContext().getSystemService(Context.**ALARM\_SERVICE**);

PendingIntent pi = PendingIntent

.getService(getApplicationContext(), 2, **new** Intent(**this**, SensorListener.**class**),

PendingIntent.**FLAG\_UPDATE\_CURRENT**);

**if** (Build.VERSION.**SDK\_INT** >= 23) {

API23Wrapper.setAlarmWhileIdle(am, AlarmManager.**RTC**, nextUpdate, pi);

} **else** {

am.set(AlarmManager.**RTC**, nextUpdate, pi);

}

**return START\_STICKY**;

}

@Override

**public void** onCreate() { **super**.onCreate();

**if** (BuildConfig.**DEBUG**) Logger.log(**"SensorListener onCreate"**);

}

@Override

**public void** onTaskRemoved(**final** Intent rootIntent) { **super**.onTaskRemoved(rootIntent); **if** (BuildConfig.**DEBUG**) Logger.log(**"sensor service task removed"**); ((AlarmManager) getSystemService(Context.**ALARM\_SERVICE**))

.set(AlarmManager.**RTC**, System.currentTimeMillis() + 500, PendingIntent

.getService(**this**, 3, **new** Intent(**this**, SensorListener.**class**), 0)); }

@Override **public void** onDestroy() { **super**.onDestroy();

**if** (BuildConfig.**DEBUG**) Logger.log(**"SensorListener onDestroy"**); **try** {

SensorManager sm = (SensorManager) getSystemService(**SENSOR\_SERVICE**); sm.unregisterListener(**this**); } **catch** (Exception e) {

**if** (BuildConfig.**DEBUG**) Logger.log(e); e.printStackTrace();

}

}

**public static** Notification getNotification(**final** Context context) { **if** (BuildConfig.**DEBUG**) Logger.log(**"getNotification"**);

SharedPreferences prefs = context.getSharedPreferences(**"pedometer"**,

Context.**MODE\_PRIVATE**);

**int** goal = prefs.getInt(**"goal"**, 10000); Database db = Database.getInstance(context); **int** today\_offset = db.getSteps(Util.getToday()); **if** (steps == 0)

steps = db.getCurrentSteps();

db.close();

Notification.Builder notificationBuilder =

Build.VERSION.**SDK\_INT** >= 26 ? API26Wrapper.getNotificationBuilder(context) : **new** Notification.Builder(context); **if** (steps > 0) {

**if** (today\_offset == Integer.**MIN\_VALUE**) today\_offset = -steps; NumberFormat format = NumberFormat.getInstance(Locale.getDefault()); notificationBuilder.setProgress(goal, today\_offset + steps, **false**).setContentText( today\_offset + steps >= goal ?

context.getString(R.string.**goal\_reached\_notification**, format.format((today\_offset + steps))) : context.getString(R.string.**notification\_text**,

format.format((goal - today\_offset - steps)))).setContentTitle( format.format(today\_offset + steps) + **" "** + context.getString(R.string.**steps**));

} **else** {

notificationBuilder.setContentText(

context.getString(R.string.**your\_progress\_will\_be\_shown\_here\_soon**)) .setContentTitle(context.getString(R.string.**notification\_title**));

}

notificationBuilder.setPriority(Notification.**PRIORITY\_MIN**).setShowWhen(**false**)

.setContentIntent(PendingIntent

.getActivity(context, 0, **new** Intent(context, Activity\_Main.**class**),

PendingIntent.**FLAG\_UPDATE\_CURRENT**)) .setSmallIcon(R.drawable.**ic\_notification**).setOngoing(**true**); **return** notificationBuilder.build();

}

**private void** registerBroadcastReceiver() {

**if** (BuildConfig.**DEBUG**) Logger.log(**"register broadcastreceiver"**); IntentFilter filter = **new** IntentFilter();

filter.addAction(Intent.**ACTION\_SHUTDOWN**); registerReceiver(**shutdownReceiver**, filter);

}

**private void** reRegisterSensor() {

**if** (BuildConfig.**DEBUG**) Logger.log(**"re-register sensor listener"**);

SensorManager sm = (SensorManager) getSystemService(**SENSOR\_SERVICE**); **try** {

sm.unregisterListener(**this**); } **catch** (Exception e) {

**if** (BuildConfig.**DEBUG**) Logger.log(e); e.printStackTrace();

}

**if** (BuildConfig.**DEBUG**) { Logger.log(**"step sensors: "** +

sm.getSensorList(Sensor.**TYPE\_STEP\_COUNTER**).size());

**if** (sm.getSensorList(Sensor.**TYPE\_STEP\_COUNTER**).size() < 1) **return**; // emulator Logger.log(**"default: "** +

sm.getDefaultSensor(Sensor.**TYPE\_STEP\_COUNTER**).getName());

}

sm.registerListener(**this**, sm.getDefaultSensor(Sensor.**TYPE\_STEP\_COUNTER**),

SensorManager.**SENSOR\_DELAY\_NORMAL**, (**int**) (5 \*

**MICROSECONDS\_IN\_ONE\_MINUTE**));

}

}

#  ShutDownReceiver:-

**package** com.example.pedometer; **import** android.content.BroadcastReceiver; **import** android.content.Context; **import** android.content.Intent; **import** com.example.pedometer.util.Logger; **import** com.example.pedometer.util.Util;

**public class** ShutdownRecevier **extends** BroadcastReceiver {

@Override

**public void** onReceive(**final** Context context, **final** Intent intent) { **if** (BuildConfig.**DEBUG**) Logger.log(**"shutting down"**); context.startService(**new** Intent(context, SensorListener.**class**));

context.getSharedPreferences(**"pedometer"**, Context.**MODE\_PRIVATE**).edit()

.putBoolean(**"correctShutdown"**, **true**).commit();

Database db = Database.getInstance(context);

**if** (db.getSteps(Util.getToday()) == Integer.**MIN\_VALUE**) {

**int** steps = db.getCurrentSteps(); db.insertNewDay(Util.getToday(), steps);

} **else** {

db.addToLastEntry(db.getCurrentSteps());

}

db.close();

}

}

#  Com.example.pedometer(fdroid)-

#  ui-

# 1) Activity\_Main.java:-

**package** com.example.pedometer.ui; **import** android.Manifest; **import** android.app.AlertDialog; **import** android.app.Fragment; **import** android.app.FragmentTransaction; **import** android.content.DialogInterface; **import** android.content.Intent;

**import** android.content.pm.PackageManager.NameNotFoundException; **import** android.net.Uri; **import** android.os.Build; **import** android.os.Bundle; **import** android.support.v4.app.FragmentActivity; **import** android.support.v4.content.PermissionChecker; **import** android.text.method.LinkMovementMethod; **import** android.view.MenuItem; **import** android.widget.TextView; **import** com.example.pedometer.BuildConfig; **import** com.example.pedometer.R; **import** com.example.pedometer.SensorListener; **public class** Activity\_Main **extends** FragmentActivity {

@Override

**protected void** onCreate(**final** Bundle b) {

**super**.onCreate(b);

startService(**new** Intent(**this**, SensorListener.**class**)); **if** (b == **null**) {

Fragment newFragment = **new** Fragment\_Overview();

FragmentTransaction transaction = getFragmentManager().beginTransaction(); transaction.replace(android.R.id.**content**, newFragment); transaction.commit();

}

**if** (BuildConfig.**DEBUG** && Build.VERSION.**SDK\_INT** >= 23 && PermissionChecker .checkSelfPermission(**this**, Manifest.permission.**WRITE\_EXTERNAL\_STORAGE**)

!= PermissionChecker.**PERMISSION\_GRANTED**) {

requestPermissions(**new**

String[]{Manifest.permission.**WRITE\_EXTERNAL\_STORAGE**}, 0);

}

}

@Override

**public void** onBackPressed() {

**if** (getFragmentManager().getBackStackEntryCount() > 0) { getFragmentManager().popBackStackImmediate();

} **else** { finish();

}

}

**public boolean** optionsItemSelected(**final** MenuItem item) { **switch** (item.getItemId()) { **case** android.R.id.**home**:

getFragmentManager().popBackStackImmediate(); **break**; **case** R.id.**action\_settings**: getFragmentManager().beginTransaction()

.replace(android.R.id.**content**, **new** Fragment\_Settings()).addToBackStack(**null**)

.commit(); **break**;

}

**return true**;

}

}

#  util-

# 1) PlaySettingsWrapper:-

**package** com.example.pedometer.util; **import** android.os.Bundle; **import** android.preference.Preference; **import** com.example.pedometer.ui.Activity\_Main; **public class** PlaySettingsWrapper {

**public static void** setupAccountSetting(**final** Preference account,**final** Bundle savedInstanceState, **final** Activity\_Main main) {

account.setSummary(**"This feature is not available on the F-Droid version of the app"**); account.setEnabled(**false**);

}

**public static void** onSavedInstance(**final** Bundle outState, **final** Activity\_Main main) {

}

}

**XML FILE-**

 **layout-**

 **dialog\_split.xml:-**

<?**xml version="1.0" encoding="utf-8"**?>

<**LinearLayout xmlns:android="http://schemas.android.com/apk/res/android" android:layout\_width="match\_parent" android:layout\_height="match\_parent" android:orientation="vertical"** >

<**FrameLayout**

**android:layout\_width="match\_parent" android:layout\_height="0dip" android:layout\_weight="1" android:padding="10dp"** >

<**TextView**

**android:id="@+id/stopped" android:layout\_width="match\_parent" android:layout\_height="match\_parent"**

**android:gravity="center"**

**android:text="@string/no\_split\_active"** />

<**RelativeLayout**

**android:id="@+id/started" android:layout\_width="match\_parent" android:layout\_height="match\_parent"** >

<**TextView**

**android:id="@+id/date" android:layout\_width="wrap\_content" android:layout\_height="wrap\_content" android:layout\_alignParentLeft="true" android:layout\_below="@+id/distance" android:layout\_marginTop="15dp"** />

<**TextView**

**android:id="@+id/distanceunit"**

**android:layout\_width="wrap\_content" android:layout\_height="wrap\_content" android:layout\_alignBaseline="@+id/distance" android:layout\_alignBottom="@+id/distance" android:layout\_alignParentRight="true"** />

<**TextView**

**android:id="@+id/steps" android:layout\_width="wrap\_content" android:layout\_height="wrap\_content" android:layout\_alignParentTop="true" android:layout\_centerHorizontal="true"**

**android:textSize="50sp"** />

<**TextView**

**android:id="@+id/distance" android:layout\_width="wrap\_content" android:layout\_height="wrap\_content" android:layout\_below="@+id/steps" android:layout\_centerHorizontal="true"**

**android:textSize="50sp"** />

<**TextView**

**android:id="@+id/textView2" android:layout\_width="wrap\_content" android:layout\_height="wrap\_content" android:layout\_alignBaseline="@id/steps" android:layout\_alignParentRight="true" android:text="@string/steps"** />

</**RelativeLayout**>

</**FrameLayout**> <**View android:layout\_width="fill\_parent" android:layout\_height="1dp" android:layout\_marginBottom="0dp"**

**android:background="?android:attr/dividerVertical"** />

<**LinearLayout style="?android:attr/buttonBarStyle" android:layout\_width="match\_parent" android:layout\_height="wrap\_content"** >

<**Button**

**android:id="@+id/start"**

**style="?android:attr/buttonBarButtonStyle" android:layout\_width="0dip" android:layout\_height="wrap\_content" android:layout\_weight="1"** />

<**Button**

**android:id="@+id/close"**

**style="?android:attr/buttonBarButtonStyle" android:layout\_width="0dip" android:layout\_height="wrap\_content" android:layout\_weight="1" android:text="@string/close"** />

</**LinearLayout**>

</**LinearLayout**>

 **fragment\_overview-**

**1) fragment\_overview.xml:-**

<**RelativeLayout 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:padding="5dp" tools:context=".ui.Activity\_Main"**> <**org.eazegraph.lib.charts.PieChart xmlns:eaze="http://schemas.android.com/apk/res-auto" android:id="@+id/graph" android:layout\_width="match\_parent" android:layout\_height="200dp" android:layout\_marginTop="10dp" eaze:egLegendHeight="0dp" eaze:egInnerPadding="75" eaze:egHighlightStrength="1"**/>

<**TextView android:id="@+id/steps" android:layout\_width="match\_parent" android:layout\_height="wrap\_content" android:layout\_marginTop="75dp" android:gravity="center" android:textSize="45sp" android:text="10.000"**/> <**TextView android:id="@+id/unit" android:layout\_width="wrap\_content" android:layout\_height="wrap\_content" android:layout\_below="@+id/steps" android:layout\_centerHorizontal="true" android:text="@string/steps" android:textSize="20sp"**/> <**LinearLayout android:id="@+id/averageandtotal" android:layout\_width="match\_parent" android:layout\_height="wrap\_content" android:layout\_alignParentLeft="true" android:layout\_below="@+id/graph" android:layout\_marginTop="20dp"**>

<**TextView android:id="@+id/average" android:layout\_width="0dp" android:layout\_height="wrap\_content" android:layout\_weight="1" android:gravity="center\_horizontal" android:textSize="20sp" android:textStyle="bold"**/>

<**TextView android:id="@+id/total" android:layout\_width="0dp" android:layout\_height="wrap\_content" android:layout\_weight="1" android:gravity="center\_horizontal" android:textSize="20sp" android:textStyle="bold"**/>

</**LinearLayout**> <**LinearLayout android:id="@+id/averageandtotaltext" android:layout\_width="match\_parent" android:layout\_height="wrap\_content" android:layout\_below="@id/averageandtotal"**>

<**TextView android:layout\_width="0dp" android:layout\_height="wrap\_content" android:layout\_weight="1" android:gravity="center\_horizontal" android:text="@string/average"**/>

<**TextView**

**android:layout\_width="0dp" android:layout\_height="wrap\_content" android:layout\_weight="1" android:gravity="center\_horizontal" android:text="@string/total"**/>

</**LinearLayout**>

<**org.eazegraph.lib.charts.BarChart**

**xmlns:eaze="http://schemas.android.com/apk/res-auto" android:id="@+id/bargraph" android:layout\_width="match\_parent" android:layout\_height="150dp"**

**android:layout\_below="@+id/averageandtotaltext" android:layout\_marginTop="50dp" eaze:egLegendHeight="35dp" eaze:egShowValues="true"**/>

</**RelativeLayout**>

**2) fragment\_overview.xml(land):-**

<**RelativeLayout 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:padding="5dp" tools:context=".ui.Activity\_Main"**> <**org.eazegraph.lib.charts.PieChart**

**xmlns:eaze="http://schemas.android.com/apk/res-auto" android:id="@+id/graph" android:layout\_width="200dp" android:layout\_height="200dp" android:layout\_marginTop="10dp" android:layout\_alignParentLeft="true" android:layout\_alignParentTop="true" eaze:egLegendHeight="0dp" eaze:egInnerPadding="75" eaze:egHighlightStrength="1"**/>

<**TextView android:id="@+id/steps" android:text="10.000" android:layout\_width="200dp" android:layout\_height="wrap\_content" android:layout\_marginTop="75dp" android:gravity="center\_horizontal" android:textSize="45sp"**/>

<**TextView android:id="@+id/unit" android:layout\_width="200dp" android:layout\_height="wrap\_content" android:layout\_below="@+id/steps" android:gravity="center\_horizontal" android:text="@string/steps" android:textSize="20sp"**/>

<**LinearLayout**

**android:id="@+id/averageandtotal"**

**android:layout\_width="match\_parent"**

**android:layout\_height="wrap\_content" android:layout\_alignLeft="@+id/bargraph" android:layout\_alignParentRight="true" android:layout\_alignTop="@+id/graph"**>

<**TextView android:id="@+id/average" android:layout\_width="0dp" android:layout\_height="wrap\_content" android:layout\_weight="1" android:gravity="center\_horizontal" android:textSize="20sp" android:textStyle="bold"**/>

<**TextView android:id="@+id/total" android:layout\_width="0dp" android:layout\_height="wrap\_content" android:layout\_weight="1" android:gravity="center\_horizontal" android:textSize="20sp" android:textStyle="bold"**/>

</**LinearLayout**> <**LinearLayout android:id="@+id/averageandtotaltext" android:layout\_width="match\_parent" android:layout\_height="wrap\_content" android:layout\_alignLeft="@+id/averageandtotal" android:layout\_below="@id/averageandtotal"**>

<**TextView android:id="@+id/textView2" android:layout\_width="0dp" android:layout\_height="wrap\_content" android:layout\_weight="1" android:gravity="center\_horizontal" android:text="@string/average"**/>

<**TextView android:id="@+id/textView4" android:layout\_width="0dp" android:layout\_height="wrap\_content" android:layout\_weight="1" android:gravity="center\_horizontal" android:text="@string/total"**/>

</**LinearLayout**>

<**org.eazegraph.lib.charts.BarChart**

**xmlns:eaze="http://schemas.android.com/apk/res-auto" android:id="@+id/bargraph" android:layout\_width="match\_parent" android:layout\_height="wrap\_content" android:layout\_below="@+id/averageandtotaltext" android:layout\_alignParentBottom="true" android:layout\_alignParentRight="true" android:layout\_toRightOf="@+id/graph" eaze:egLegendHeight="35dp" eaze:egShowValues="true"**/>

</**RelativeLayout**>

 **signin.xml:-**

<?**xml version="1.0" encoding="utf-8"**?>

<**LinearLayout xmlns:android="http://schemas.android.com/apk/res/android" android:layout\_width="match\_parent" android:layout\_height="match\_parent" android:orientation="vertical" android:padding="5dp"** >

<**TextView android:id="@+id/textView1" android:layout\_width="wrap\_content" android:layout\_height="wrap\_content" android:paddingBottom="10dp" android:text="@string/sign\_in"** />

<**TextView**

**android:id="@+id/signedin" android:layout\_width="wrap\_content" android:layout\_height="wrap\_content" android:paddingBottom="10dp"** /> <**com.google.android.gms.common.SignInButton android:id="@+id/sign\_in\_button" android:layout\_width="wrap\_content" android:layout\_height="wrap\_content" android:layout\_gravity="center\_horizontal" android:paddingBottom="10dp"** />

</**LinearLayout**>

 **statistics-**

**1) statistics.xml:-**

<?**xml version="1.0" encoding="utf-8"**?>

<**LinearLayout xmlns:android="http://schemas.android.com/apk/res/android" android:layout\_width="match\_parent" android:layout\_height="match\_parent" android:orientation="vertical" android:padding="10dp"** >

<**TextView**

**android:id="@+id/record" android:layout\_width="wrap\_content" android:layout\_height="wrap\_content"**

**android:textSize="20sp" android:textStyle="bold"** />

<**TextView android:layout\_width="wrap\_content" android:layout\_height="wrap\_content" android:layout\_marginBottom="20dp" android:text="@string/record"** />

<**TextView android:id="@+id/totalthisweek" android:layout\_width="wrap\_content" android:layout\_height="wrap\_content"**

**android:textSize="20sp" android:textStyle="bold"** />

<**TextView**

**android:layout\_width="wrap\_content" android:layout\_height="wrap\_content" android:layout\_marginBottom="20dp" android:text="@string/total\_last\_7\_days"** />

<**TextView**

**android:id="@+id/averagethisweek" android:layout\_width="wrap\_content" android:layout\_height="wrap\_content"**

**android:textSize="20sp" android:textStyle="bold"** />

<**TextView**

**android:layout\_width="wrap\_content"**

**android:layout\_height="wrap\_content" android:layout\_marginBottom="20dp" android:text="@string/average\_last\_7\_days"** />

<**TextView**

**android:id="@+id/totalthismonth" android:layout\_width="wrap\_content" android:layout\_height="wrap\_content"**

**android:textSize="20sp" android:textStyle="bold"** />

<**TextView android:layout\_width="wrap\_content" android:layout\_height="wrap\_content" android:layout\_marginBottom="20dp" android:text="@string/total\_this\_month"** />

<**TextView android:id="@+id/averagethismonth" android:layout\_width="wrap\_content" android:layout\_height="wrap\_content"**

**android:textSize="20sp" android:textStyle="bold"** />

<**TextView android:layout\_width="wrap\_content" android:layout\_height="wrap\_content" android:layout\_marginBottom="20dp" android:text="@string/average\_this\_month"** />

<**View android:layout\_width="fill\_parent" android:layout\_height="1dip"**

**android:background="?android:attr/dividerHorizontal"** />

<**Button**

**android:id="@+id/close" style="?android:attr/buttonBarButtonStyle" android:layout\_width="match\_parent" android:layout\_height="wrap\_content" android:layout\_gravity="center\_horizontal" android:text="@android:string/ok"** />

</**LinearLayout**>

**2) statistics.xml(land):-**

<?**xml version="1.0" encoding="utf-8"**?>

<**RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"**

**android:layout\_width="match\_parent" android:layout\_height="match\_parent" android:orientation="vertical" android:padding="10dp"**>

<**TextView**

**android:id="@+id/record" android:layout\_width="wrap\_content" android:layout\_height="wrap\_content"**

**android:textSize="20sp" android:textStyle="bold"**/>

<**TextView**

**android:id="@+id/recordtext" android:layout\_below="@id/record" android:layout\_width="wrap\_content" android:layout\_height="wrap\_content" android:layout\_marginBottom="20dp" android:text="@string/record"**/>

<**TextView**

**android:id="@+id/averagethisweektext" android:layout\_below="@id/recordtext" android:layout\_alignParentRight="true" android:layout\_width="wrap\_content" android:layout\_height="wrap\_content" android:text="@string/average\_last\_7\_days"**/>

<**TextView**

**android:id="@+id/totalthisweek" android:layout\_below="@id/averagethisweektext" android:layout\_width="wrap\_content" android:layout\_height="wrap\_content" android:textSize="20sp" android:textStyle="bold"**/>

<**TextView android:id="@+id/averagethisweek" android:layout\_alignBaseline="@id/totalthisweek" android:layout\_alignParentRight="true" android:layout\_width="wrap\_content" android:layout\_height="wrap\_content" android:textSize="20sp" android:textStyle="bold"**/>

<**TextView android:id="@+id/totalthisweektext" android:layout\_below="@id/totalthisweek" android:layout\_width="wrap\_content" android:layout\_height="wrap\_content" android:layout\_marginBottom="20dp" android:text="@string/total\_last\_7\_days"**/>

<**TextView android:id="@+id/averagethismonthtext" android:layout\_alignParentRight="true" android:layout\_below="@id/totalthisweektext" android:layout\_width="wrap\_content" android:layout\_height="wrap\_content" android:text="@string/average\_this\_month"**/>

<**TextView**

**android:id="@+id/totalthismonth" android:layout\_below="@id/averagethismonthtext" android:layout\_width="wrap\_content" android:layout\_height="wrap\_content" android:textSize="20sp" android:textStyle="bold"**/>

<**TextView android:id="@+id/averagethismonth" android:layout\_alignBaseline="@id/totalthismonth" android:layout\_alignParentRight="true" android:layout\_width="wrap\_content" android:layout\_height="wrap\_content" android:textSize="20sp" android:textStyle="bold"**/>

<**TextView**

**android:layout\_below="@id/totalthismonth" android:layout\_width="wrap\_content" android:layout\_height="wrap\_content" android:layout\_marginBottom="20dp" android:text="@string/total\_this\_month"**/>

<**View android:layout\_width="fill\_parent" android:layout\_height="1dip" android:layout\_above="@id/close" android:background="?android:attr/dividerHorizontal"**/>

<**Button**

**android:layout\_alignParentBottom="true" android:id="@+id/close" style="?android:attr/buttonBarButtonStyle" android:layout\_width="match\_parent" android:layout\_height="wrap\_content" android:layout\_gravity="center\_horizontal" android:text="@android:string/ok"**/>

</**RelativeLayout**>

 **stepsize.xml:-**

<?**xml version="1.0" encoding="utf-8"**?>

<**LinearLayout xmlns:android="http://schemas.android.com/apk/res/android" android:layout\_width="match\_parent" android:layout\_height="match\_parent" android:gravity="center" android:orientation="horizontal"** >

<**EditText**

**android:id="@+id/value" android:layout\_width="wrap\_content" android:layout\_height="wrap\_content" android:ems="4"**

**android:inputType="numberDecimal"** >

<**requestFocus** />

</**EditText**> <**RadioGroup android:id="@+id/unit" android:layout\_width="wrap\_content" android:layout\_height="wrap\_content" android:orientation="horizontal"** >

<**RadioButton android:id="@+id/cm" android:layout\_width="wrap\_content" android:layout\_height="wrap\_content" android:checked="true" android:text="cm"** /> <**RadioButton android:id="@+id/ft" android:layout\_width="wrap\_content" android:layout\_height="wrap\_content"**

**android:text="ft"** />

</**RadioGroup**>

</**LinearLayout**>

 **widget.xml:-**

<?**xml version="1.0" encoding="utf-8"**?>

<**LinearLayout xmlns:android="http://schemas.android.com/apk/res/android" android:id="@+id/widget" android:layout\_width="match\_parent" android:layout\_height="match\_parent" android:gravity="center\_vertical|center\_horizontal" android:orientation="vertical"** >

<**TextView**

**android:id="@+id/widgetsteps" android:layout\_width="wrap\_content" android:layout\_height="wrap\_content" android:gravity="center\_horizontal" android:shadowColor="#000" android:shadowDx="0"**

**android:shadowDy="2" android:shadowRadius="3" android:textSize="20sp"** />

<**TextView android:id="@+id/widgettext" android:layout\_width="wrap\_content" android:layout\_height="wrap\_content" android:gravity="center\_horizontal" android:shadowColor="#000" android:shadowDx="0" android:shadowDy="2" android:shadowRadius="3" android:text="@string/steps" android:textSize="12sp"** />

</**LinearLayout**>

 **widgetconfig.xml:-**

<?**xml version="1.0" encoding="utf-8"**?>

<**RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"**

**android:layout\_width="match\_parent" android:layout\_height="match\_parent" android:padding="15dp"**>

<**TextView**

**android:id="@+id/texttext" android:layout\_width="wrap\_content" android:layout\_height="wrap\_content" android:layout\_alignParentLeft="true" android:layout\_alignParentTop="true" android:text="@string/text\_color"**/>

<**TextView**

**android:id="@+id/bgtext" android:layout\_width="wrap\_content" android:layout\_height="wrap\_content" android:layout\_alignParentLeft="true" android:layout\_below="@+id/texttext" android:layout\_marginTop="15dp" android:text="@string/background\_color"**/> <**com.example.lib.colorpicker.ColorPreviewButton android:id="@+id/bgcolor" android:layout\_width="20dp" android:layout\_height="20dp" android:layout\_alignTop="@+id/bgtext" android:layout\_toRightOf="@+id/bgtext" android:layout\_marginLeft="15dp"**/> <**com.example.lib.colorpicker.ColorPreviewButton**

**android:id="@+id/textcolor" android:layout\_width="20dp" android:layout\_height="20dp" android:layout\_alignTop="@+id/texttext" android:layout\_alignLeft="@+id/bgcolor"**/>

</**RelativeLayout**>

* **menu-**

**1) main.xml:-**

<**menu xmlns:android="http://schemas.android.com/apk/res/android"** >

<**item android:id="@+id/action\_settings" android:orderInCategory="100" android:title="@string/settings"**/> <**item**

**android:id="@+id/action\_split\_count" android:orderInCategory="100" android:title="@string/split\_count"**/>

</**menu**>

* **OUTPUT-**





