5/24/2020

Rajasekhar Reddy kothapalli

BATTERYALAEM

THE SAVER

**BATTERYALAEM**

(THE SAVER)

-BY

K.RAJASEKHAR REDDY.

L19CSE279006.

CSE 2nd year.

(2019-2020).

**Content**

|  |  |  |
| --- | --- | --- |
| S.NO | TOPIC | PAGE.NO |
| 1. | Introduction | 03 |
| 2. | App-Explanation | 04 |
| 3. | App-Interface | 05 |
| 4. | Code for user-interface | 06-07 |
| 5. | Blueprint of User-Interface | 08 |
| 6. | Code for app working | 09-14 |
| 7. | Code for String.xml file | 15 |
| 8. | Code for AndroidManifest.xml | 16 |
| 9. | Code for Styles.xml | 17 |
| 10. | Output | 18 |
| 11. | Advantages,Disadvantages,Conclusion | 19 |

**Introduction:**

At present we are having enough resources like, fuel, electricity.,e.t.c.. But for your future generations there will be not if we waste resources. This made me too think. Now-a-days everyone is using a smartphone. Smartphone is the only thing through which ever person can connect (or) communicate thorough out the world. To run a smartphone, we needPower to charge battery. How many are you charging your smartphone whole night? Some time I also did it. Due to charging our phone for a long time our battery spoil and we are wasting one of the most useful non-renewable resources (I.e. coal for producing electricity).

For to save electricity and batter spoiling issue. I came up with a solution called “Battery Alarm”. Battery Alarm is an application (app) on our smartphone from which we can save electricity. Let`s see how it happened!

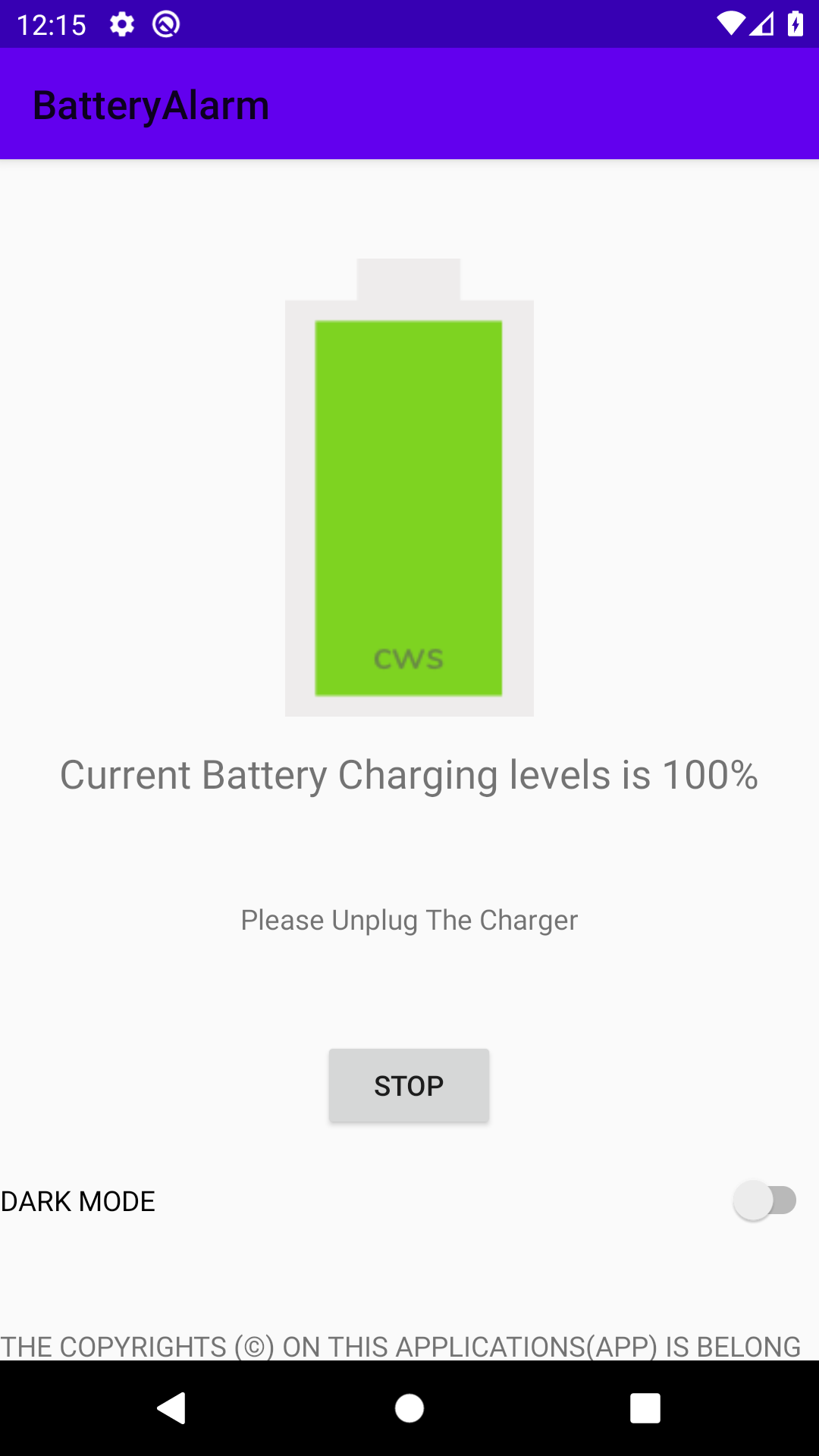
Basically, we are humans we all having a busy schedule. So that we are simply forgot your phone in charging socket throughout the night. To achieve our goal, we need to do is simply ringing our phone up by an alarm. Which gives us remainder to take out phone from the charging socket.

**App-Explanation:**

“Battery Alarm” which will give you remainder by alarm, when phone battery is reached 100%. When user take out the plug the alarm will be stop automatically. Otherwise we have “STOP” Button that can stop alarm reminder manually. To improve user experience, we added battery graphically to determine battery percentage.

Let`s check the User-interface of app.

**App-Interface:**

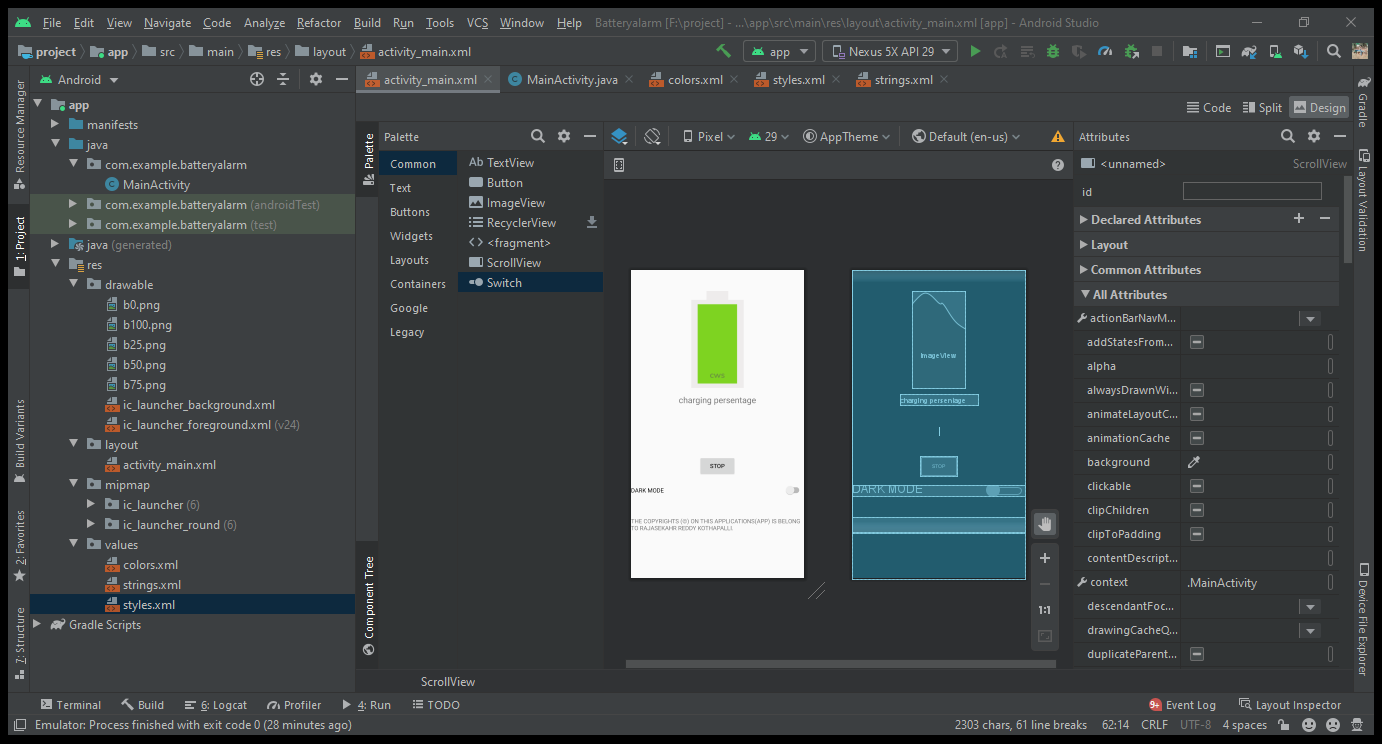


The above picture shows the User-interface of the App. Which consist of an image view to show battery graphically, consist of text view to show battery percentage, a button to stop alarm, and a text view which consist of copyrights info. let`s see code for user-interface.

**Code for user-interface:**

<?xml version="1.0" encoding="utf-8"?>  
<ScrollView  
  
 xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 tools:context=".MainActivity">  
  
 <LinearLayout  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:orientation="vertical">  
 <ImageView  
 android:id="@+id/batteryimage"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:src="@drawable/b100"  
 android:layout\_marginTop="50dp"  
 android:layout\_gravity="center"  
 android:contentDescription="@string/todo" />  
 <TextView  
 android:id="@+id/textView"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_gravity="center"  
 android:layout\_marginTop="15dp"  
 android:textSize="20sp"  
 android:text="@string/charging\_persentage" />  
 <TextView  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:id="@+id/unplug"  
 android:layout\_gravity="center"  
 android:layout\_marginTop="50dp"/>  
 <Button  
 android:id="@+id/button"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_gravity="center"  
 android:layout\_marginTop="50dp"  
 android:layout\_marginBottom="20dp"  
 android:text="@string/stop\_Button"  
 android:onClick="stopButton"/>  
  
 <Switch  
 android:id="@+id/switch1"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="DARK MODE" />  
 <TextView  
 android:id="@+id/copyrights"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_gravity="bottom"  
 android:layout\_marginTop="50dp"  
 android:text="@string/copyrights"/>  
  
  
 </LinearLayout>  
</ScrollView>

**Blueprint of User-Interface:**



Let`s take a look at main java program.

**Code for app working:**

package com.example.batteryalarm;  
  
import androidx.appcompat.app.AppCompatActivity;  
import androidx.appcompat.app.AppCompatDelegate;  
  
import android.os.Bundle;  
import android.content.BroadcastReceiver;  
import android.content.Context;  
import android.content.Intent;  
import android.content.IntentFilter;  
import android.content.res.Resources;  
import android.media.Ringtone;  
import android.media.RingtoneManager;  
import android.os.BatteryManager;  
import android.os.Bundle;  
import android.view.View;  
import android.widget.CompoundButton;  
import android.widget.ImageView;  
import android.widget.Switch;  
import android.widget.TextView;  
import android.widget.Toast;  
  
  
public class MainActivity extends AppCompatActivity {  
 private TextView textView;  
 private Ringtone ringtone;  
 private TextView unplug;  
 private Switch darkmode;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.activity\_main);  
 textView = findViewById(R.id.textView);  
 unplug = findViewById(R.id.unplug);  
 darkmode = findViewById(R.id.switch1);  
  
 ringtone = RingtoneManager.getRingtone(getApplicationContext(), RingtoneManager.getDefaultUri(RingtoneManager.TYPE\_RINGTONE));  
 BroadcastReceiver broadcastReceiverBattrery = new BroadcastReceiver() {  
 @Override  
 public void onReceive(Context context, Intent intent) {  
 // for linking  
 ImageView batteryimage = (ImageView) findViewById(R.id.batteryimage);  
 Integer integerBatteryLevel = intent.getIntExtra(BatteryManager.EXTRA\_LEVEL, 0);  
 Integer chargePlug = intent.getIntExtra(BatteryManager.EXTRA\_PLUGGED, -1);  
 Boolean usbCharge = chargePlug == BatteryManager.BATTERY\_PLUGGED\_USB;  
 Boolean acCharge = chargePlug == BatteryManager.BATTERY\_PLUGGED\_AC;  
 Resources res = context.getResources();  
  
 // for images  
 if (integerBatteryLevel >= 90) {  
 batteryimage.setImageDrawable(res.getDrawable(R.drawable.b100));  
 } else if (90 > integerBatteryLevel && integerBatteryLevel >= 65) {  
 batteryimage.setImageDrawable(res.getDrawable(R.drawable.b75));  
 } else if (65 > integerBatteryLevel && integerBatteryLevel >= 40) {  
 batteryimage.setImageDrawable(res.getDrawable(R.drawable.b50));  
 } else if (40 > integerBatteryLevel && integerBatteryLevel >= 15) {  
 batteryimage.setImageDrawable(res.getDrawable(R.drawable.b25));  
 } else {  
 batteryimage.setImageDrawable(res.getDrawable(R.drawable.b0));  
 }  
 textView.setText("Current Battery Charging levels is " + integerBatteryLevel.toString() + "%");  
  
 //for charging full  
  
 if (integerBatteryLevel > 99) {  
 if (acCharge || usbCharge) {  
 ringtone.play();  
 unplug.setText("Please Unplug The Charger");  
 }  
 }  
  
 //for charging remove toast  
 if (acCharge == false && usbCharge == false) {  
 ringtone.stop();  
 unplug.setText("");  
 if(integerBatteryLevel == 100 && acCharge == false && usbCharge == false ){  
 Toast.makeText(getApplicationContext(),"THANK YOU FOR CHOOSING OUR APP.", Toast.LENGTH\_SHORT).show();  
 }  
 }  
  
 //implementing dark mode  
 darkmode.setOnCheckedChangeListener(new CompoundButton.OnCheckedChangeListener() {  
 @Override  
 public void onCheckedChanged(CompoundButton buttonView, boolean isChecked) {  
 if (isChecked) {  
 AppCompatDelegate.setDefaultNightMode(AppCompatDelegate.MODE\_NIGHT\_YES);  
 } else {  
 AppCompatDelegate.setDefaultNightMode(AppCompatDelegate.MODE\_NIGHT\_NO);  
 }  
 }  
 });  
  
 }  
 };  
 registerReceiver(broadcastReceiverBattrery, new IntentFilter(Intent.ACTION\_BATTERY\_CHANGED));  
 }  
 public void stopButton(View view) {  
 ringtone.stop();  
 }  
}

The above program is the main java program to run app.  
Let`s see the string.xml. The file which consist strings used in app.

**Code for String.xml file:**

<resources>  
 <string name="app\_name">BatteryAlarm</string>  
 <string name="charging\_persentage">charging persentage</string>  
 <string name="stop\_Button">STOP</string>  
 <string name="copyrights">  
 THE COPYRIGHTS (©) ON THIS APPLICATIONS(APP) IS BELONG TO RAJASEKAHR REDDY KOTHAPALLI.  
 </string>  
 <string name="todo">TODO</string>  
</resources>

Let`s see the manifest.xml file which determine the icon style ,shape,e.t.c.

**Code for AndroidManifest.xml:**

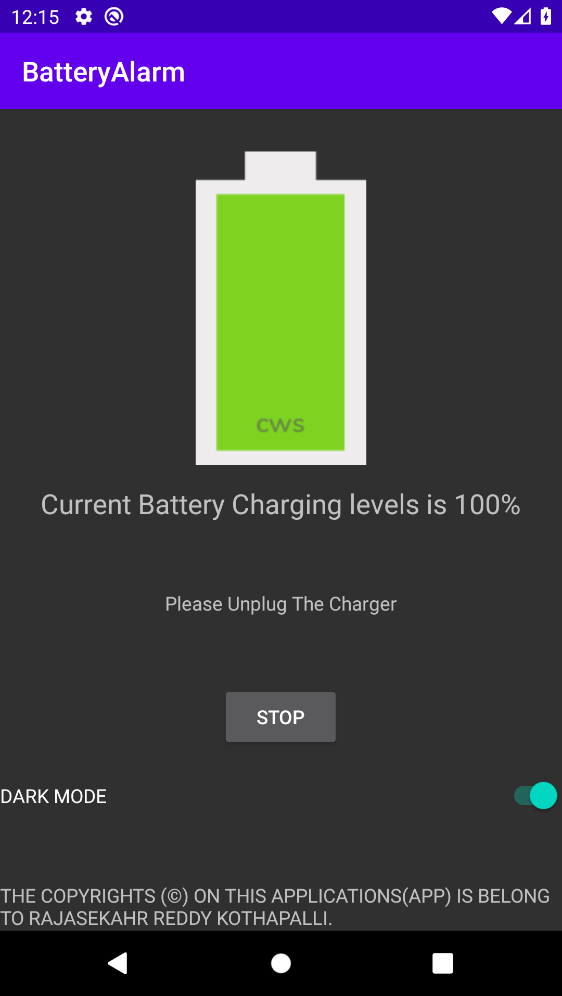
<?xml version="1.0" encoding="utf-8"?>  
<manifest xmlns:android="http://schemas.android.com/apk/res/android"  
 package="com.example.batteryalarm">  
 <supports-screens  
 android:largeScreens="true"  
 android:normalScreens="true"  
 android:smallScreens="true"  
 android:xlargeScreens="true"/>  
 <application  
 android:allowBackup="true"  
 android:icon="@mipmap/ic\_launcher"  
 android:label="@string/app\_name"  
 android:roundIcon="@mipmap/ic\_launcher\_round"  
 android:supportsRtl="true"  
 android:theme="@style/AppTheme">  
 <activity android:name=".MainActivity">  
 <intent-filter>  
 <action android:name="android.intent.action.MAIN" />  
 <category android:name="android.intent.category.LAUNCHER" />  
 </intent-filter>  
 </activity>  
 </application>  
</manifest>

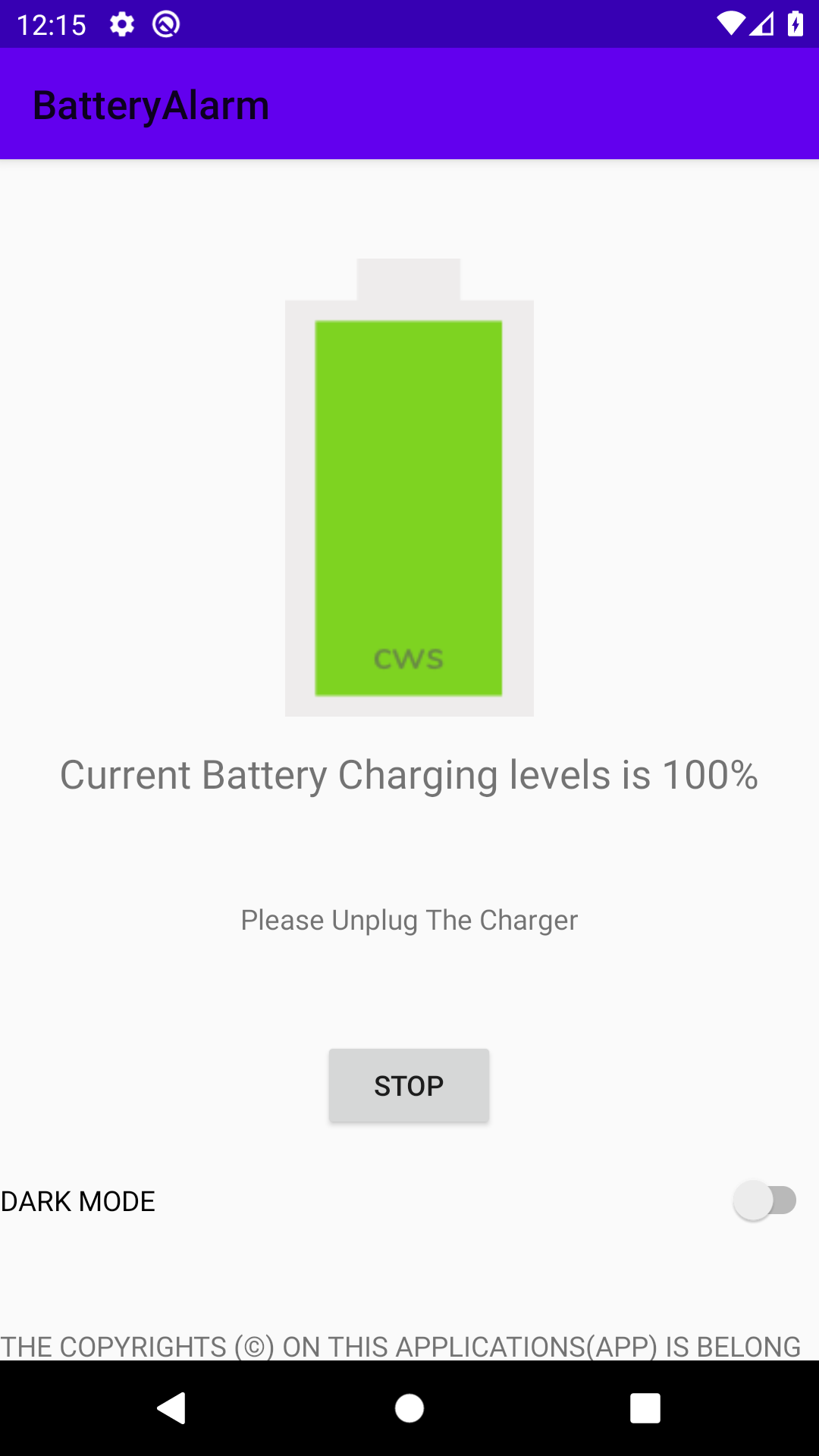
**Code for Styles.xml:**

<resources>  
 <!-- Base application theme. -->  
 <style name="AppTheme" parent="Theme.AppCompat.DayNight">  
 <!-- Customize your theme here. -->  
 <item name="colorPrimary">@color/colorPrimary</item>  
 <item name="colorPrimaryDark">@color/colorPrimaryDark</item>  
 <item name="colorAccent">@color/colorAccent</item>  
 </style>  
  
</resources>

**Output:**

Here is the finally out of the application (app).





**Advantages:**

1. We can save electricity.
2. We can prevent battery from spoil issues.
3. We can prevent phone from burn out issues.
4. The app is small in size which consume small memory.
5. By using dark mode we can save battery power.

**Disadvantages:**

1. The app should be in memory (or) User need to open the app before charging.

**Conclusion:**

Battery Alarm app helps to save power and battery. It is small in size at the end of the day memory will also be saved.