

浙江师范大学
ZHEJIANG NORMAL UNIVERSITY

移动开发课程期末项目报告

项目名称: 节电管理

专业班级: 计师 171、网络 171

小组编号: 23

成员 1: 韩玮

成员 2: 费俊涛

成员 3: 顾霄鹏

成员 4: 许晓鹏

成绩:

1. 项目实施说明

1.1 项目要求

1.1.1 项目概况：非充电状态下，尤其是低电量模式下，APP 更新频率过高，会加速手机电池老化；通过监测手机电量和充电状态，提示用户对 APP 进行节电设置（即降低后台更新频率）。

- 监测到手机电量过低，自动设置所有 APP 进入节电模式；
- 监测到手机断开电源，列出若干未设置节电模式的 APP 信息，供用户选择是否进入节电模式；
- 监测到电量恢复正常水平，列出已处于节电模式的 APP 信息，询问用户是否取消节电模式，返回正常模式。
- 监测到手机正在充电，自动设置所有 APP 恢复正常模式；

1.1.2 小组合作部分：

- 数据存储模式设计 ~ 数据应至少覆盖以下信息：
 - 请自行编制一份 APP 信息清单（APP 名、节电比、当前模式）；
- 统一风格的布局设计

1.1.3 个人功能点 1：电量过低—统一节电

- 监测手机电量是否过低——发文字提示信息；
- 自动将所有 APP 的模式设置为节电模式，并显示在主界面上；

1.1.4 个人功能 2：断开电源—选择节电

- 监测手机是否断开外接电源 ——使手机震动；
- 显示所有未进入节电模式的 APP 信息，用户可从中选择要设置节电的 APP；

1.1.5 个人功能 3：电量正常—部分恢复正常

- 监测手机电量是否已恢复正常——发文字提示信息；
- 显示所有处于节电模式的 APP 信息，用户从中选择要恢复正常模式的 APP；

1.1.6 个人功能点 4：接上电源—全部恢复正常

- 监测手机是否连接上外接电源——使手机震动；
- 自动将所有 APP 的模式更改为正常模式，并显示在主界面上。

1.2 小组成员分工

表 1-1 任务分工

	负责的功能点	其 他
成员 1	UI 设计、activity 之间跳转	模块组装测试、APP 图标设计及修改
成员 2	数据库存储模式设计	
成员 3	断开电源—选择节电 接上电源—全部恢复正常	广告启动页面
成员 4	电量正常—部分恢复正常 电量过低—统一节电 手机震动功能	

1.3 数据存储模式设计

本项目中使用数据库存储的方式，对数据进行存储。设置了一个表格，用于存放 APP 节电状态信息。

表 1-2 数据库设计

数据库名	数据表
Android.db	APP

APP 表中存放数据对应的 id，图片 id，名称，节电比，节电状态

表 1-3 APP 表

列名	数据类型	长度	备注
Id	Interger	/	主键
Img	Varchar	20	
Name	Varchar	20	
Scale	Varchar	20	
status	Varchar	20	

1.3.1 创建数据库以及初始化

```
public DBUtils(Context context) {
    db = context.openOrCreateDatabase("android.db", context.MODE_PRIVATE, null);
    createTableApp();
}

private void createTableApp() {
    sql="CREATE TABLE IF NOT EXISTS APP (id INTEGER PRIMARY KEY AUTOINCREMENT, img
    VARCHAR2(20), name VARCHAR2(20), scale VARCHAR2(20), status VARCHAR2(20))";
    try{
        db.rawQuery("SELECT id _id, img, name, scale, status FROM APP", null);
    } catch (Exception e) {
        Log.i("Info", "not exit");
        db.execSQL(sql);
        insert("0", "京東", "85", "正常");
        insert("1", "QQ", "98", "正常");
        insert("2", "天猫", "52", "正常");
        insert("3", "UC 浏览器", "75", "正常");
        insert("4", "微信", "65", "正常");
        insert("5", "新浪", "77", "正常");
    }
}
```

1.3.2 查询所有应用，节电状态的应用，未节电的应用

```
public Cursor queryall() {
    sql="SELECT id _id, img, name, scale, status FROM APP";
}
```

```

        c=db.rawQuery(sql,null);
        return c;
    } //查询 app 信息

    public Cursor querynormal() {
        sql="SELECT id _id,img,name,scale,status FROM APP where status=' 正常' ";
        c=db.rawQuery(sql,null);
        return c;
    } //查询正常模式的 app 信息

    public Cursor querysavepower() {
        sql="SELECT id _id,img,name,scale,status FROM APP where status=' 节电' ";
        c=db.rawQuery(sql,null);
        return c;
    } //查询节电模式的 app 信息

    public Integer querysavacount() {
        Integer count=0;
        sql="SELECT count(*) FROM APP where status=' 节电' ";
        Cursor cursor=db.rawQuery(sql,null);
        if(cursor.getCount()>0)
        {
            cursor.moveToFirst();
            do{
                count = cursor.getInt(0);
            }while (cursor.moveToNext());
        }
        return count;
    } //查询节电模式的 app 数量

```

1.3.3 保存，修改数据

//插入数据

```

    public void insert(String app_img,String app_name,String app_scale,String
app_status) {
        ContentValues values=new ContentValues();
        values.put("img", app_img);
        values.put("name", app_name);
        values.put("scale", app_scale);
        values.put("status", app_status);
        db.insert("APP", null, values);
    }

```

//修改指定 APP 的节电状态

```

    public void update(String name,String app_status) {
        ContentValues values = new ContentValues();
        values.put("status", app_status);
        db.update("APP", values, "name=?", new String[] {name});
    }

```

//修改所有 APP 的节电状态

```

    public void updateAll(String app_status) {
        ContentValues values = new ContentValues();
        values.put("status", app_status);
        db.update("APP", values, null, null);
    }

```

1.4 项目界面设计

1.4.1 总体设计

APP 下分两个界面，分别是主界面和应用耗电状态管理界面。应用耗电状态管理根据主界面跳转的请求码的不同，分别加载数据库中不同数据内容。同事返回时刷新主界面中当前节电应用占比。整体界面加载流程如下：

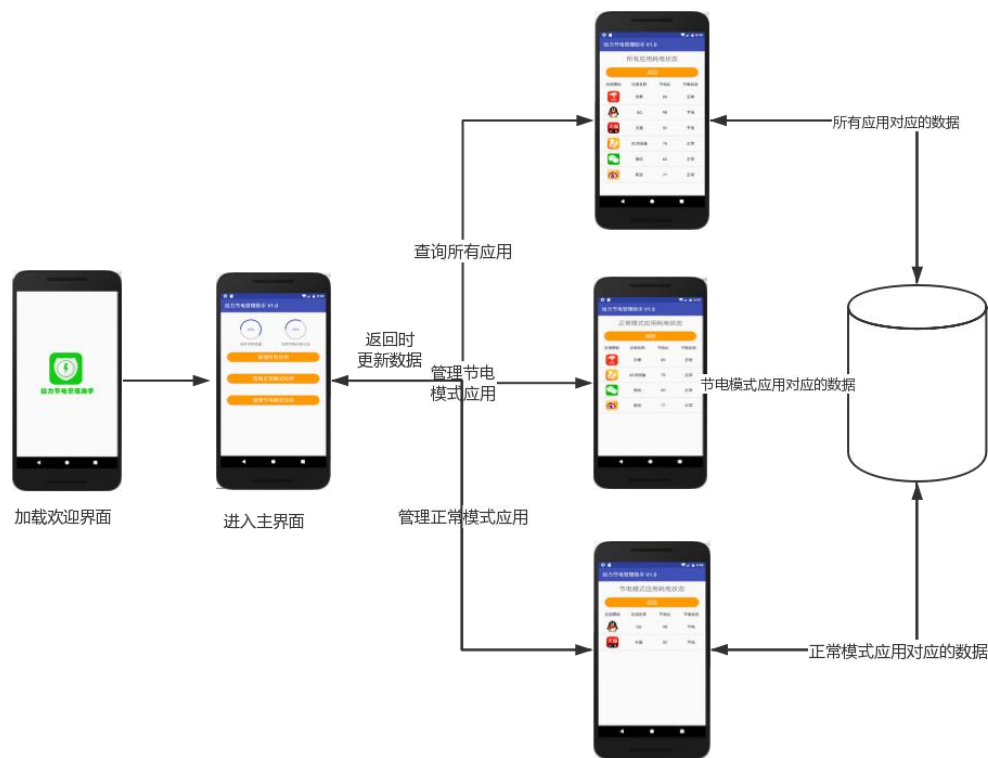


图 1-1 项目界面加载流程图

1.4.2 欢迎界面设计

①界面预览：



图 1-2 欢迎页面设计（预览）

②设计思路：
欢迎界面整体风格简约大方，突出应用的图标和名称，采用了线性布局，是图标能够居中显示，同时为了适应分辨率的屏幕，在图片中设置了 android:adjustViewBounds="true"属性保持宽高比。

1.4.3 主界面设计

①界面预览



图 1-3 主界面设计（预览）

②设计思路
主界面界面采用了线性布局，线性排列。为了生动形象的显示当前手机电量和节电模式应用占比，使用了开源的一个圆形进度条 UI 移植到项目里面，进度条下面设计了三个按钮 Button, 使用 shape 定义圆角矩形的外边框。

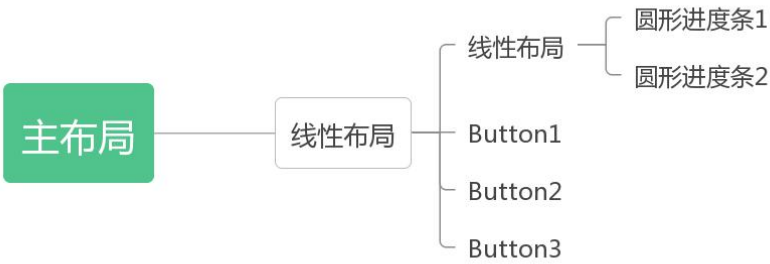


图 1-4 主界面控件结构图

1.4.4 应用状态管理界面设计

①界面预览



图 1-5 应用状态管理界面（预览）

②设计思路

本界面功能是显示数据库中应用程序的耗电状态，并进行节电管理的界面，采用线性布局，垂直排列，顶部是一个 TextView 显示主标题，接着是一个 Button 提供返回操作，下面是四个 TextView 嵌套在一个线性布局里面显示应用信息的标题，底部是一个 ListView 显示数据。

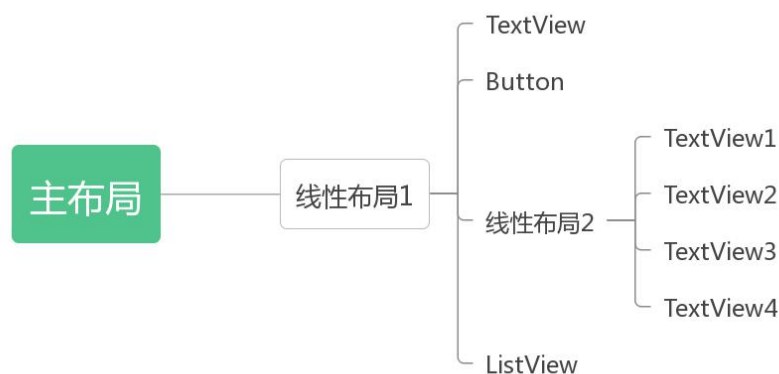


图 1-6 应用状态管理界面控件结构图

1.5 项目总结

在完成这个节电管理项目的整个过程时，起初我们组每人认领了一个老师给的任务点进行编程，在这过程中我们发现了要想实现各个任务点需求并不难并且每个任务点与任务点之间都有许多重复的部分。但是随后我们发现我们还需统一的 UI 与数据存取方法，于是我们开始重新分工，将 UI 设计、数据存取、功能实现、app 优化美化等部分进行了重新分配。

经过重新分工后我们的工作变得更加的效率，很快的就完成了大部分任务，但是在如何将一系列不同的应用图标显示在 listview 中的 ImageView 中这一事上我们遇到了困难。根据资料查询等方式我们决定先使用将图片解析为二进制的数，然后放入到数据库中的一种 BLOB 类型中这一方法。在经过一番尝试之后我们发现这一方法的实现过于繁琐，并且效果并不好，数据库中无法存储过多的图片，于是我们转用了通过简化图片的 drawable, id 的存储实现不同应用图标的显示，事实也证明通过简化图片的 drawable, id 的存储比将图片解析为二进制数据存入数据库中更加的便捷。

本次项目经历对于我们来说是一次很好的经验积累，也是一次很好的提升自己的机会。通过完成本次项目，我们大家的项目编程能力均得到了提升，学会了分工与合作，也学到了很多在书上没有的知识，并且对 Android Studio 的操作也变得更加得心应手了。而在整个项目的完成过程中，我们每个人也都发现了自己的不足之处，会在今后对其加以弥补和修正。希望以后能获得更多能提升自己的机会，也很感谢陈老师给予了我们这次机会。

2. 功能点 1（完成人：韩玮）

2.1 分析与设计

①UI 设计部分

技术点：常见布局的使用

设计分析：参考 1.4 项目界面设计

②activity 之间跳转

技术点：activity 之间跳转及数据传递、activity 的生命周期

设计分析：

由于 APP 设计的时候考虑到应用程序显示的界面的复用性，创建了一 ShowAPPActivity，根据 MainActivity 跳转的时候 intent 的携带的数据的不同进行相应的数据库查询操作；同时主界面的节电应用占比在改变应用耗电模式的时候需要刷新，发生在 ShowAPPActivity 销毁和后台运行的时候，于是我重写了 ShowAPPActivity 的 onResume() 方法并添加了返回按钮，使用 Intent 回传数据给 MainActivity，告诉主界面节电模式的 APP 数量有变动，MainActivity 中使用 startActivityForResult() 方法跳转，同时重写 onActivityResult() 方法，在接受到数据库变动的信息后，进行相应的刷新主界面的节电应用占比进度条。

③APP 图标设计及修改

技术点：更换 APP 应用图标

设计分析：

作为一款 APP 应用，当然需要一个醒目的 logo，在网上的图标库中查找合适的图标素材并参考网上的教程配置了新的 APP 图标。

④参考资料

Android studio 更换 APP 应用图标：<https://www.cnblogs.com/whycxb/p/9096727.html>

Android 自定义 view 之圆形进度条：<https://www.jianshu.com/p/45c44ad92d50>

引用圆形进度条项目地址：<https://github.com/24Kshign/CustomCircleProgressBar>

2.2 实现代码

2.2.2UI 设计

①主界面 xml 代码

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout 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"
    android:orientation="vertical"
    tools:context="com.example.h.myapplication.MainActivity">

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginLeft="30dp"
        android:layout_marginRight="30dp">
```



```

android:layout_marginTop="30dp"
android:gravity="center"
android:orientation="horizontal">

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_weight="1"
    android:orientation="vertical">

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:gravity="center">

        <com.example.h.myapplication.widget.CustomCircleProgressBar
            android:id="@+id/am_progressbar_one"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            app:direction="left"
            app:max_progress="100"
            app:outside_color="@color/colorPrimary"
            app:outside_radius="40dp"
            app:progress="0"
            app:progress_width="4dp" />
        </LinearLayout>

        <TextView
            android:id="@+id/textView"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_marginTop="10dp"
            android:gravity="center"
            android:text="当前手机电量" />
        </LinearLayout>

        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:layout_weight="1"
            android:orientation="vertical">

            <LinearLayout
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:gravity="center">

                <com.example.h.myapplication.widget.CustomCircleProgressBar
                    android:id="@+id/am_progressbar_two"
                    android:layout_width="wrap_content"
                    android:layout_height="wrap_content"
                    app:direction="left"
                    app:max_progress="100"
                    app:outside_color="@color/colorPrimary"
                    app:outside_radius="40dp"
                    app:progress="0"
                    app:progress_width="4dp" />
                </LinearLayout>

                <TextView
                    android:id="@+id/textView1"

```

```

        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginTop="10dp"
        android:gravity="center"
        android:text="当前节电应用占比" />
    </LinearLayout>

```

```

</LinearLayout>

```

```

<Button
    android:id="@+id/btn_1"
    android:layout_width="match_parent"
    android:layout_height="40dp"
    android:text="管理所有应用"
    android:layout_margin="25dp"
    android:textSize="20sp"
    android:textColor="#ffffff"
    android:background="@drawable/bg_btn1"
    android:layout_marginTop="10dp"/>

```

```

<Button
    android:id="@+id/btn_2"
    android:layout_width="match_parent"
    android:layout_height="40dp"
    android:text="管理正常模式应用"
    android:layout_margin="25dp"
    android:textSize="20sp"
    android:textColor="#ffffff"
    android:background="@drawable/bg_btn1"
    android:layout_marginTop="10dp"/>

```

```

<Button
    android:id="@+id/btn_3"
    android:layout_width="match_parent"
    android:layout_height="40dp"
    android:text="管理节电模式应用"
    android:layout_margin="25dp"
    android:textSize="20sp"
    android:textColor="#ffffff"
    android:background="@drawable/bg_btn1"
    android:layout_marginTop="10dp"/>

```

```

</LinearLayout>

```

②应用程序耗电状态显示界面 xml 代码

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout 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"
    android:orientation="vertical"
    android:padding="5dp"
    tools:context="com.example.h.myapplication.ShowAPPActivity">

```

```

    <TextView
        android:id="@+id/tv_title"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:gravity="center_horizontal"
        android:textSize="25sp"

```

```

        android:layout_margin="10dp"
        android:text="所有程序用电状态" />
<Button
    android:id="@+id/btn_exit"
    android:layout_width="match_parent"
    android:layout_height="40dp"
    android:text="返回"
    android:textSize="20sp"
    android:textColor="#ffffff"
    android:background="@drawable/bg_btn1"
    android:layout_marginLeft="20dp"
    android:layout_marginRight="20dp"
    android:layout_marginTop="10dp"
    android:layout_marginBottom="10dp"/>

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="40dp"
    android:orientation="horizontal" >

    <TextView
        android:layout_width="0dp"
        android:layout_height="match_parent"
        android:layout_weight="2"
        android:gravity="center"
        android:text="应用图标"
        android:textColor="@color/black"
        android:textSize="15sp" />

    <TextView
        android:layout_width="0dp"
        android:layout_height="match_parent"
        android:layout_weight="2"
        android:gravity="center"
        android:text="应用名称"
        android:textColor="@color/black"
        android:textSize="15sp" />

    <TextView
        android:layout_width="0dp"
        android:layout_height="match_parent"
        android:layout_weight="2"
        android:gravity="center"
        android:text="节电比"
        android:textColor="@color/black"
        android:textSize="15sp" />

    <TextView
        android:layout_width="0dp"
        android:layout_height="match_parent"
        android:layout_weight="2"
        android:gravity="center"
        android:text="节电状态"
        android:textColor="@color/black"
        android:textSize="15sp" />
</LinearLayout>

<ListView
    android:id="@+id/lv"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:divider="#d9d9d9"

```

```

        android:dividerHeight="1dp"/>

</LinearLayout>

```

(2) Activity 跳转

MainActivity 中关键代码:

```

@Override
public void onClick(View v) {
    switch (v.getId())
    {
        case R.id.btn_1:
            Intent intent1 = new Intent(MainActivity.this, ShowAPPActivity.class);
            intent1.putExtra("app_select", "01");
            startActivityForResult(intent1, 1);
            break;
        case R.id.btn_2:
            Intent intent2 = new Intent(MainActivity.this, ShowAPPActivity.class);
            intent2.putExtra("app_select", "02");
            startActivityForResult(intent2, 1);
            break;
        case R.id.btn_3:
            Intent intent3 = new Intent(MainActivity.this, ShowAPPActivity.class);
            intent3.putExtra("app_select", "03");
            startActivityForResult(intent3, 1);
            break;
    }
}

@Override
protected void onActivityResult(int requestCode, int resultCode, Intent data) {
    super.onActivityResult(requestCode, resultCode, data);
    if (data != null){
        if (requestCode == 1){
            if (resultCode == 1){
                count_next = (int)(dbUtils.querysavacount()/6.0*100);
                if (count_next!=count_current)
                {
                    progressBarTwo.setProgress(count_next);
                    count_current = count_next;
                }
            }
        }
    }
}

```

ShowAPPActivity 关键代码:

```

button.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        Intent intent1 = new Intent();
        setResult(1,intent1);
        finish();
    }
});
@Override
protected void onResume() {
    super.onResume();
    Intent intent1 = new Intent();
    setResult(1,intent1);
}

```

2.3 测试截图

①根据点击的按钮不同，分别加载不同的压面



图 1-7 点击管理所有按钮页面跳转结果



图 1-8 点击管理正常模式按钮页面跳转结果



图 1-9 点击管理节电模式按钮页面跳转结果

②安装后 APP 图标显示



图 1-10 安装后 APP 图标

2.4 小结与反思

在完成这个项目的时候，因为是分工完成的，刚开始分工之后大家自己完成手头上的任务，然后汇总，在第一次程序汇总的时候发现了许多问题，原因是做任务的时候大家之间没有协商好，一些接口和数据库没有统一，导致模块组装的时候任务模块之间不能相互数据传递。于是我们重新分配任务，经过商讨明确了一些接口和数据库中数据表和字段名，在后续的代码编程中我们组员之间及时沟通，相互帮助，及时发现程序问题，及时解决，最终完成了小组项目，通过这个实验我体会到了小组合作中沟通交流的重要性，通过沟通能够及时的了解其他模块的开发情况，及时协调数据处理的接口匹配，使得项目开发的最终组装顺利完成。作为项目的 UI 设计者，项目可以在背景图片、颜色搭配等方面进一步改进，得到更好的用户体验。

3. 功能点 2（完成人: 费俊涛）

3.1 分析与设计

为了实现数据库的存储模式，需要先进行建表，还要对其里面的数据初始化。同时需要实现数据的增删改查，所以在 DBUtils 的 java 文件先准备好对数据库基本操作的函数。之后在自定义的适配器里增加对应的监听事件。

3.2 实现代码

```
public class DBUtils {

    private SQLiteDatabase db;
    private String sql;
    private Cursor c;

    public DBUtils(Context context) {
        db = context.openOrCreateDatabase("android.db", context.MODE_PRIVATE, null);
        createTableApp();
    }

    private void createTableApp() {
        sql="CREATE TABLE IF NOT EXISTS APP (id INTEGER PRIMARY KEY AUTOINCREMENT, img VARCHAR2(20) ,name VARCHAR2(20) ,scale VARCHAR2(20) ,status VARCHAR2(20))";
        try{
            db.rawQuery("SELECT id _id, img, name, scale, status FROM APP", null);
        } catch (Exception e) {
            Log.i("Info", "not exit");
            db.execSQL(sql);
            insert("0", "京東", "85", "正常");
            insert("1", "QQ", "98", "正常");
            insert("2", "天貓", "52", "正常");
            insert("3", "UC 浏览器", "75", "正常");
            insert("4", "微信", "65", "正常");
            insert("5", "新浪", "77", "正常");
        }
    }

    public Cursor queryall() {
        sql="SELECT id _id, img, name, scale, status FROM APP";
        c=db.rawQuery(sql, null);
        return c;
    } //查询 app 信息

    public Cursor querynormal() {
        sql="SELECT id _id, img, name, scale, status FROM APP where status=' 正常' ";
        c=db.rawQuery(sql, null);
        return c;
    } //查询正常模式的 app 信息

    public Cursor querysavepower() {
        sql="SELECT id _id, img, name, scale, status FROM APP where status=' 节电' ";
        c=db.rawQuery(sql, null);
        return c;
    } //查询节电模式的 app 信息

    public Integer querysavacount() {
        Integer count=0;
        sql="SELECT count(*) FROM APP where status=' 节电' ";
        Cursor cursor=db.rawQuery(sql, null);
        if(cursor.getCount()>0) {
            cursor.moveToFirst();
            do{
                count = cursor.getInt(0);
            }while (cursor.moveToNext());
        }
        return count;
    } //查询节电模式的 app 数量

    public Cursor deleteall() {
        sql="DELETE APP";
        c=db.rawQuery(sql, null);
    }
```

```

        return c;
    } //查询 app 信息

    //保存数据
    public void insert(String app_img,String app_name,String app_scale,String app_status){
        ContentValues values=new ContentValues();
        values.put("img",app_img);
        values.put("name",app_name);
        values.put("scale",app_scale);
        values.put("status",app_status);
        db.insert("APP",null,values);
    }

    //修改数据
    public void update(String name,String app_status){
        ContentValues values = new ContentValues();
        values.put("status",app_status);
        db.update("APP",values,"name=?",new String[]{name});
    }

    public void updateAll(String app_status){
        ContentValues values = new ContentValues();
        values.put("status",app_status);
        db.update("APP",values,null,null);
    }
}

```

3.3 测试截图



图 1-11 测试节电模式是否修改成功

3.4 小结与反思

数据库文件还是好写的，关键在于适配器的调用那边。一开始写的监听事件有点问题，没有调用到关键按钮，调试很久才发现。可能因为有数据库的基础，所以觉得 sql 语句很亲切，但对安卓的调用规则还是无感，以后多多写写。

4. 功能点 3 （完成人: 顾霄鹏）

4.1 分析与设计

读取电池状态，利用 BroadcastReceiver 组件，注册广播接收数据实现的。

启动广告页面通过利用了 android.os.Handler 的消息的延迟发送以及处理来实现 3s 的启动广告再转跳到主界面的功能。

4.2 实现代码

电源管理:

```
private void monitorBatteryState() {
    batteryLevelRcvr = new BroadcastReceiver() {

        @Override
        public void onReceive(Context context, Intent intent) {
            if(intent !=null){
                String action = intent.getAction();
                switch (action){
                    case Intent.ACTION_POWER_CONNECTED:
                        Vibrator vibrator_on=(Vibrator) context.getSystemService(Service.VIBRATOR_SERVICE);
                        vibrator_on.vibrate(new long[]{0,1000}, -1);
                        Toast.makeText(context, "[电源连接]", Toast.LENGTH_SHORT).show();
                        dbUtils.updateAll("正常");
                        progressBarTwo.setProgress(0);
                        break;
                    case Intent.ACTION_POWER_DISCONNECTED:
                        Vibrator vibrator_off=(Vibrator) context.getSystemService(Service.VIBRATOR_SERVICE);
                        vibrator_off.vibrate(new long[]{0,1000}, -1);
                        final AlertDialog.Builder builder = new AlertDialog.Builder(MainActivity.this);
                        //单选框
                        builder.setTitle("温馨提示: ");
                        builder.setMessage("当前电源已断开, 是否进入节电模式管理? ");
                        builder.setNegativeButton("确定", new DialogInterface.OnClickListener() {
                            @Override
                            public void onClick(DialogInterface dialog, int which) {
                                Intent intent = new Intent(MainActivity.this, ShowAPPActivity.class);
                                intent.putExtra("app_select", "02");
                                startActivity(intent);
                            }
                        });
                        builder.setPositiveButton("取消", null);
                        builder.create().show();
                        Toast.makeText(context, "[电源断开]", Toast.LENGTH_SHORT).show();
                        break;
                }
            }
        }
    };
}
```

注册广播:

```
batteryLevelFilter = new IntentFilter();
batteryLevelFilter.addAction(Intent.ACTION_POWER_CONNECTED);
batteryLevelFilter.addAction(Intent.ACTION_POWER_DISCONNECTED);
batteryLevelFilter.addAction(Intent.ACTION_BATTERY_CHANGED);
registerReceiver(batteryLevelRcvr, batteryLevelFilter);
```

启动页面转跳:

```
public class WelcomeActivity extends Activity {

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

```

getWindow().setFlags(WindowManager.LayoutParams.FLAG_FULLSCREEN,
    WindowManager.LayoutParams.FLAG_FULLSCREEN);
requestWindowFeature(Window.FEATURE_NO_TITLE);
setContentView(R.layout.activity_welcome);
handler.sendMessageDelayed(0, 2000);
}

private Handler handler = new Handler() {
    @Override
    public void handleMessage(Message msg) {
        getHome();
        super.handleMessage(msg);
    }
};

public void getHome() {
    Intent intent = new Intent(WelcomeActivity.this, MainActivity.class);
    startActivity(intent);
    finish();
}
}

```

4.3 测试截图

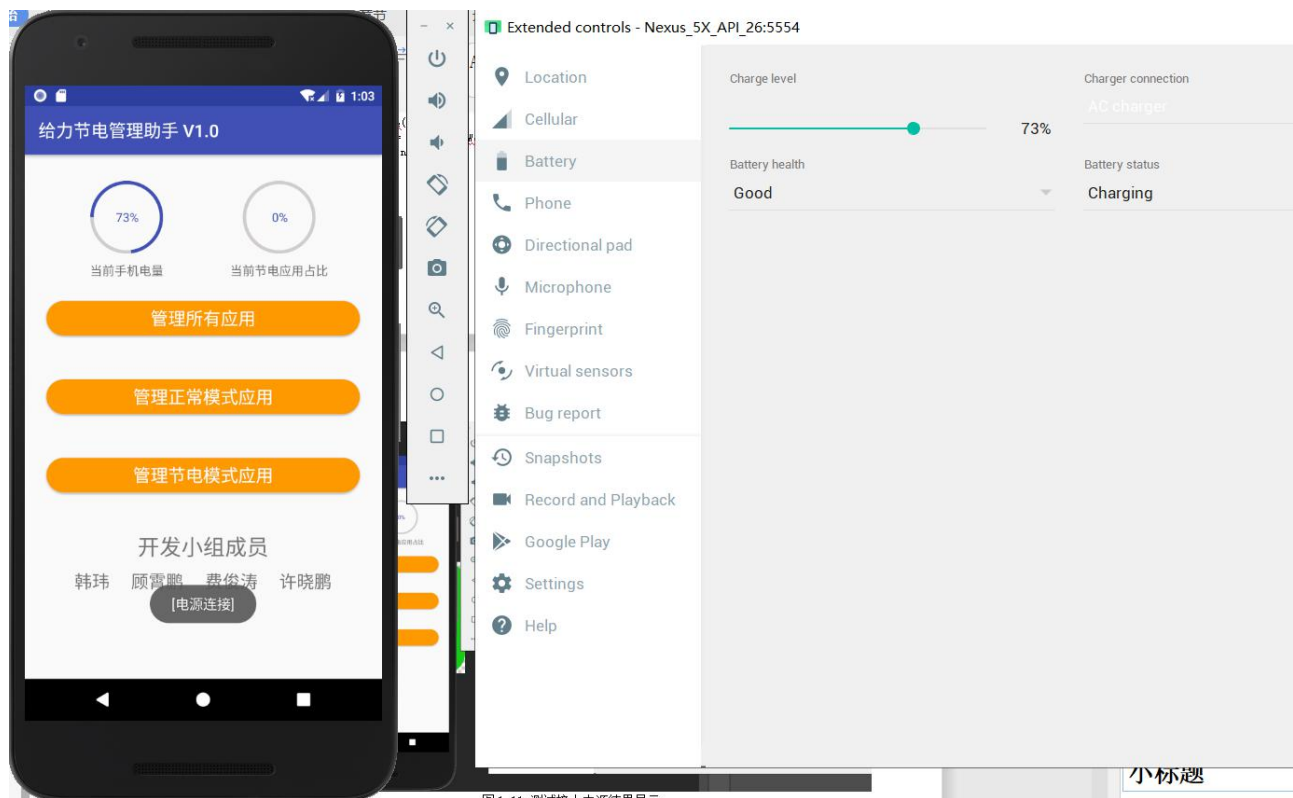


图 1-12 测试接上电源结果显示



图 1-13 测试接上电源后程序的节点状态变化

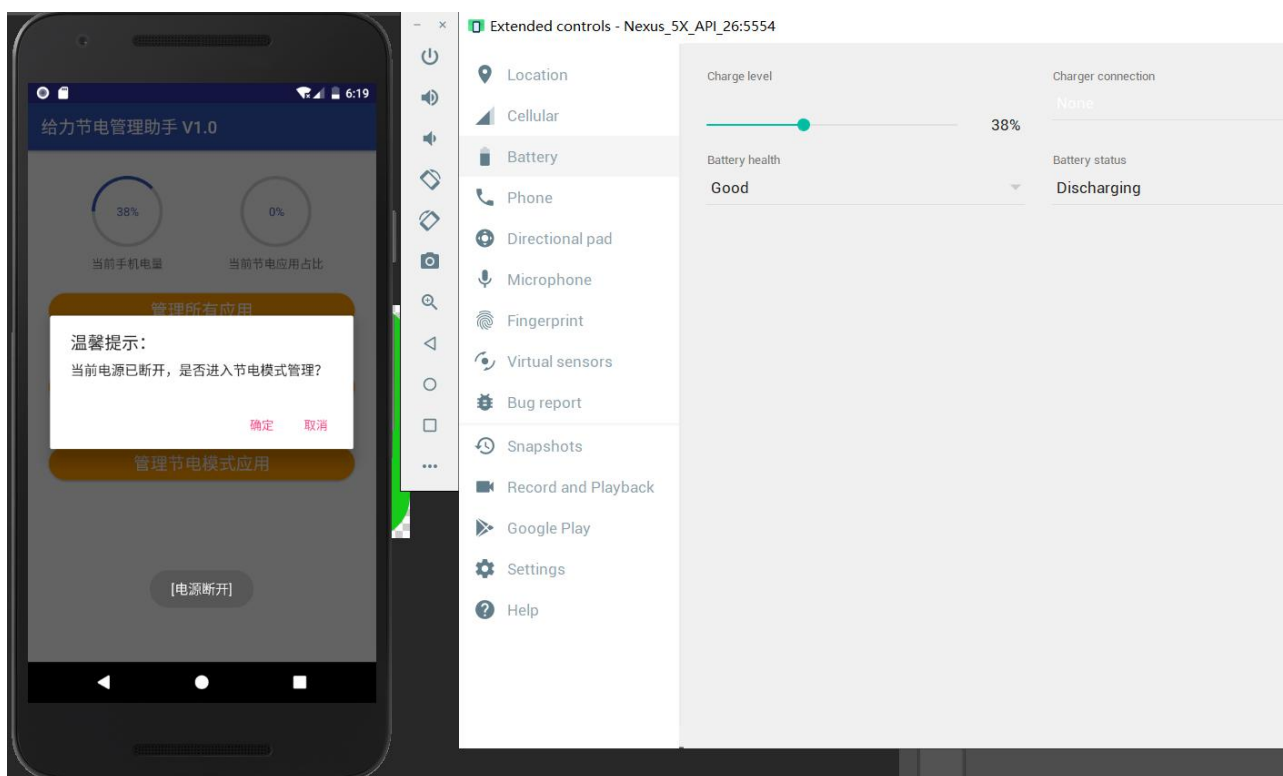


图 1-14 测试断开电源结果显示



图 1-15 延时广告启动页面

4.4 小结与反思

在这个任务点中，学习到了广播的用法，但是在接收者向广播者回传数据的点上一开始遇到了困难本来准备写个接口来实现数据传输但后来选择了更加简单的方法同时学习到了利用延时发送来实现 APP 的延时广告启动页面。

5. 功能点 4（完成人:许晓鹏）

5.1 分析与设计

注册一个 BroadcastReceiver 接收电池电量改变时发出的广播，通过 batteryManager 类中的 BatteryManager.BATTERY_PROPERTY_CAPACITY 获取电量百分比，通过 intent.getIntExtra("status", 0) 获取电池状态判断电池是否处于充电状态，并以此为基准进行电量的提示和数据库的修改操作。

5.2 实现代码

```
private void monitorBatteryState() {
    batteryLevelRcvr = new BroadcastReceiver() {
        @Override
        public void onReceive(Context context, Intent intent) {
            if(intent !=null){
                String action = intent.getAction();
                switch (action){
                    case Intent.ACTION_BATTERY_CHANGED:
                        BatteryManager batteryManager = (BatteryManager) getSystemService(BATTERY_SERVICE);
                        battery_next = batteryManager.getIntProperty(BatteryManager.BATTERY_PROPERTY_CAPACITY);
                        int status = intent.getIntExtra("status", 0); //获取电池状态
                        if (battery_next == 15 && status == BatteryManager.BATTERY_STATUS_DISCHARGING) {
                            dbUtils.updateAll("节电");
                            progressBarTwo.setProgress(100);
                            Toast.makeText(context, "[电量过低，请充电]", Toast.LENGTH_SHORT).show();
                        }
                        if(battery_next == 60 && status == BatteryManager.BATTERY_STATUS_CHARGING) {
                            final AlertDialog.Builder builder1 = new AlertDialog.Builder(MainActivity.this);
                            //单选框
                            builder1.setTitle("温馨提示: ");
                            builder1.setMessage("当前电量恢复正常，是否恢复正常模式? ");
                            builder1.setNegativeButton("确定", new DialogInterface.OnClickListener() {
                                @Override
                                public void onClick(DialogInterface dialog, int which) {
                                    Intent intent = new Intent(MainActivity.this, ShowAPPActivity.class);
                                    intent.putExtra("app_select", "03");
                                    startActivity(intent);
                                }
                            });
                            builder1.setPositiveButton("取消", null);
                            builder1.create().show();
                            Toast.makeText(context, "[当前电量恢复正常]", Toast.LENGTH_SHORT).show();
                        }
                        if (battery_next != battery_current) {
                            progressBarOne.setProgress(battery_next);
                            battery_current = battery_next;
                        }
                        break;
                    }
            }
        }
    };
    batteryLevelFilter = new IntentFilter(); //注册广播
    batteryLevelFilter.addAction(Intent.ACTION_BATTERY_CHANGED);
    registerReceiver(batteryLevelRcvr, batteryLevelFilter);
}
```

清单文件中添加震动器权限<uses-permission android:name="android.permission.VIBRATE" />

震动器调用 Vibrator vibrator_on=(Vibrator) context.getSystemService(Service.VIBRATOR_SERVICE);
 vibrator_on.vibrate(new long[]{0,1000}, -1);

5.3 测试截图

5.3.1 测试功能点：电量过低—统一节电

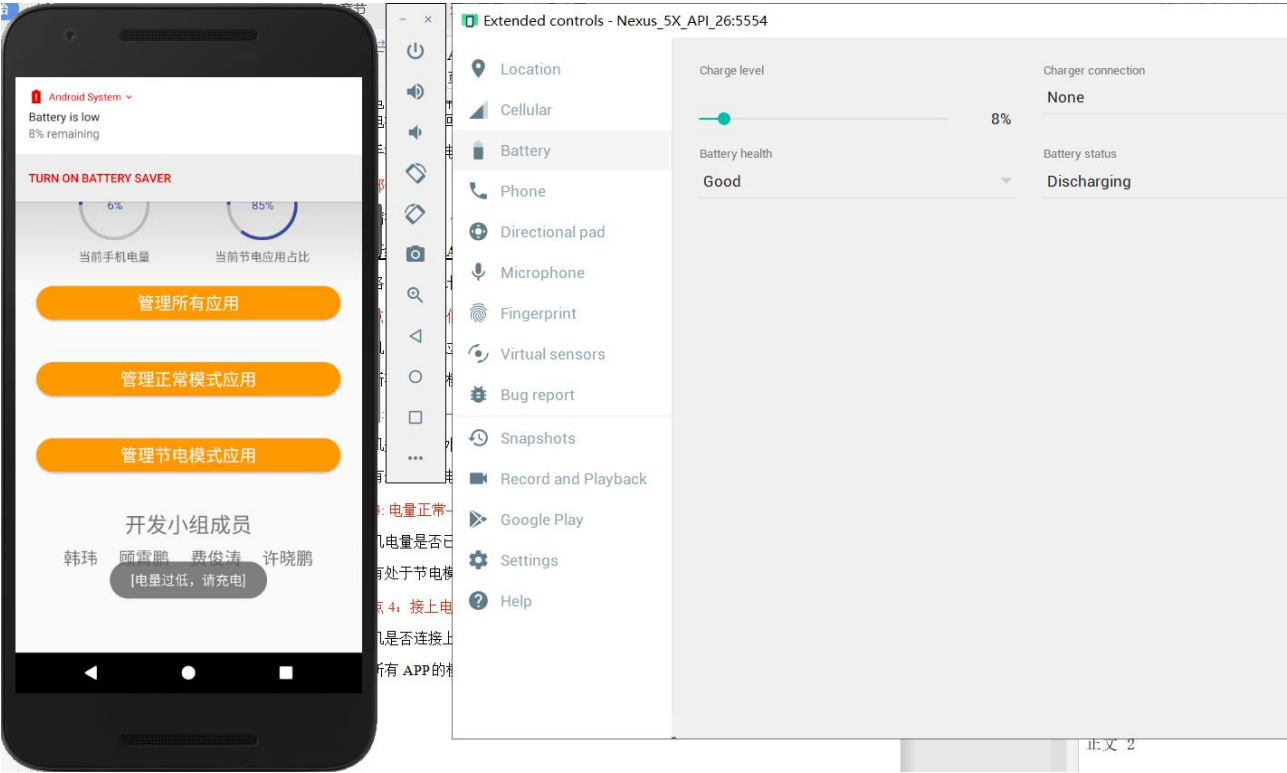


图 1-16 测试电量过低的时候发消息提醒



图 1-17 测试电量过低的时候 APP 节电状态变化

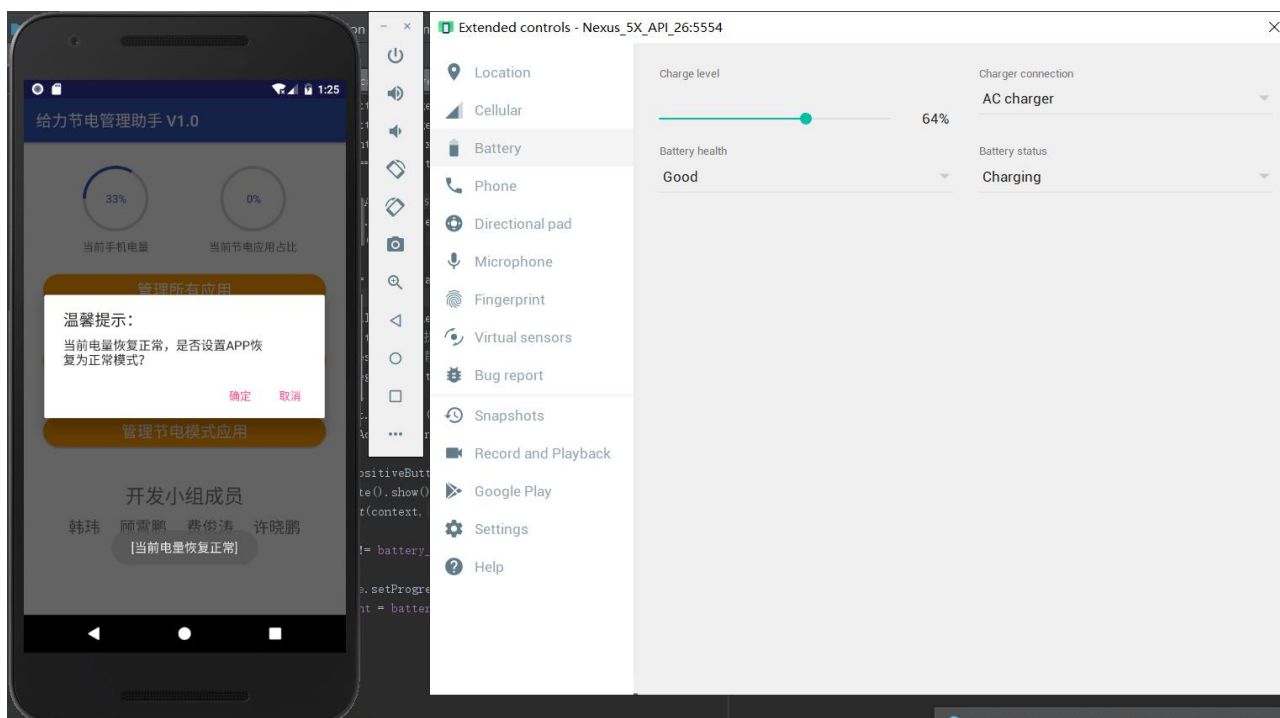


图 1-18 测试电量恢复正常的时候发消息提醒

5.4 小结与反思

`intent.getExtras().getInt("level")`和`intent.getExtras().getInt("scale")`;这两个方法是我开始构思的读取电量百分比的方式，它们分别读取电池当前电量和总电量，但在整合的时候发现 UI 的电量进度条的显示无法用此方法进行，所以使用了 `BatteryManager.BATTERY_PROPERTY_CAPACITY` 获取电量百分比。震动器由于一开始忘记添加权限，所以导致震动器不可用，有关于是否需要权限这一问题以后还得多加留意。